CỨU RỖI CHÚNG SINH KHỎI KIẾP NẠN TRẦN THẾ

ĐẠI SỐ BOOLEAN

$$0 + 0 = 0$$
$$0 + 1 = 1$$
$$1 + 0 = 1$$
$$1 + 1 = 1$$

$$0 \,.\, 0 = 0$$
$$0 \,.\, 1 = 0$$
$$1 \,.\, 0 = 0$$
$$1 \,.\, 1 = 1$$

$$\overline{0} = 1$$
$$\overline{1} = 0$$

ĐỊNH LÝ DEMORGAN

$$\overline{A + B + C + \ldots} = \overline{A}.\overline{B}.\overline{C}\ldots.$$
$$\overline{A.B.C\ldots.} = \overline{A} + \overline{B} + \overline{C} + \ldots$$

Giao hoán $A + B = B + A$

$A.B = B.A$

Phân phối $(A + B).C = A.C + B.C$

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

Kết hợp $(A + B) + C = A + (B + C)$

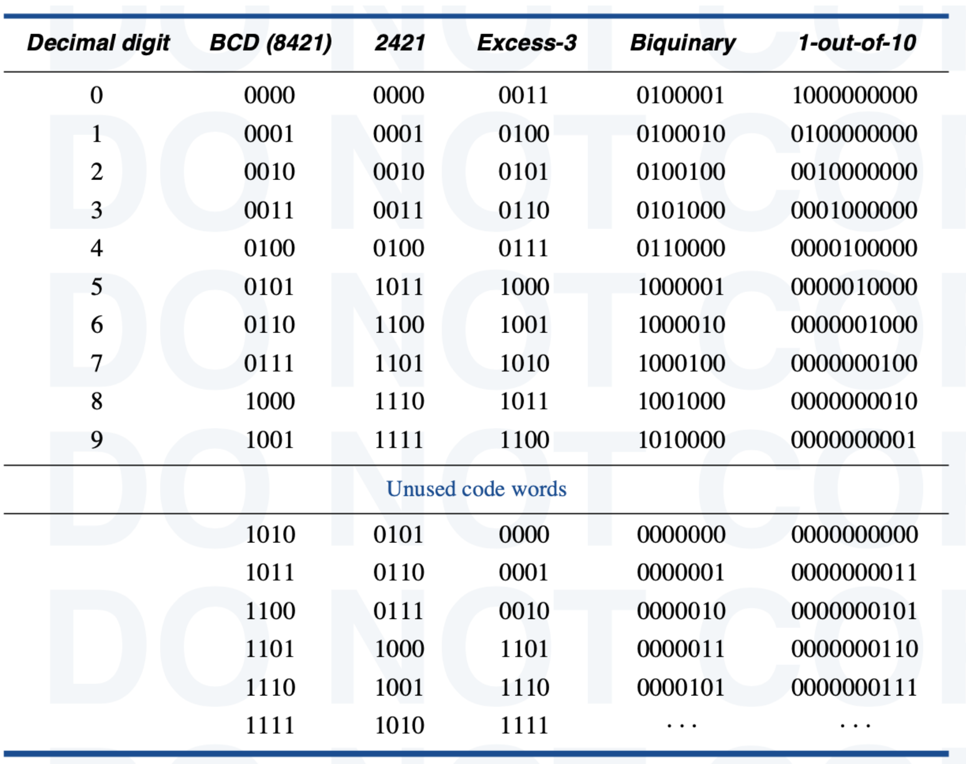
$(A.B).C = A.(B.C)$

$$A + 0 = A \qquad A.0 = 0 \qquad \overline{\overline{A}} = A$$
$$A + 1 = 1 \qquad A.1 = A$$
$$A + A = A \qquad A.A = A$$
$$A + \overline{A} = 1 \qquad A.\overline{A} = 0$$

Một số đẳng thức:

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

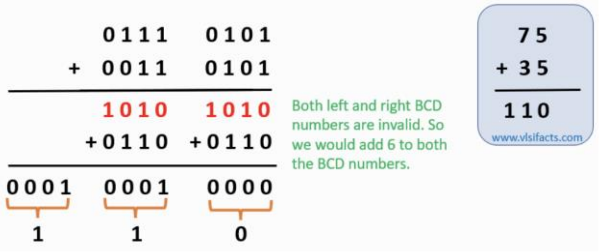
BCD CODE



Decimal digit	BCD (8421)	2421	Excess-3	Biquinary	1-out-of-10
0	0000	0000	0011	0100001	1000000000
1	0001	0001	0100	0100010	0100000000
2	0010	0010	0101	0100100	0010000000
3	0011	0011	0110	0101000	0001000000
4	0100	0100	0111	0110000	0000100000
5	0101	1011	1000	1000001	0000010000
6	0110	1100	1001	1000010	0000001000
7	0111	1101	1010	1000100	0000000100
8	1000	1110	1011	1001000	0000000010
9	1001	1111	1100	1010000	0000000001
		Unused code words			
	1010	0101	0000	0000000	0000000000
	1011	0110	0001	0000001	0000000011
	1100	0111	0010	0000010	0000000101
	1101	1000	1101	0000011	0000000110
	1110	1001	1110	0000101	0000000111
	1111	1010	1111	. . .	. . .

85_{10} = 1000 0101 (BCD)

572_{10} = 0101 0111 0010 (BCD)



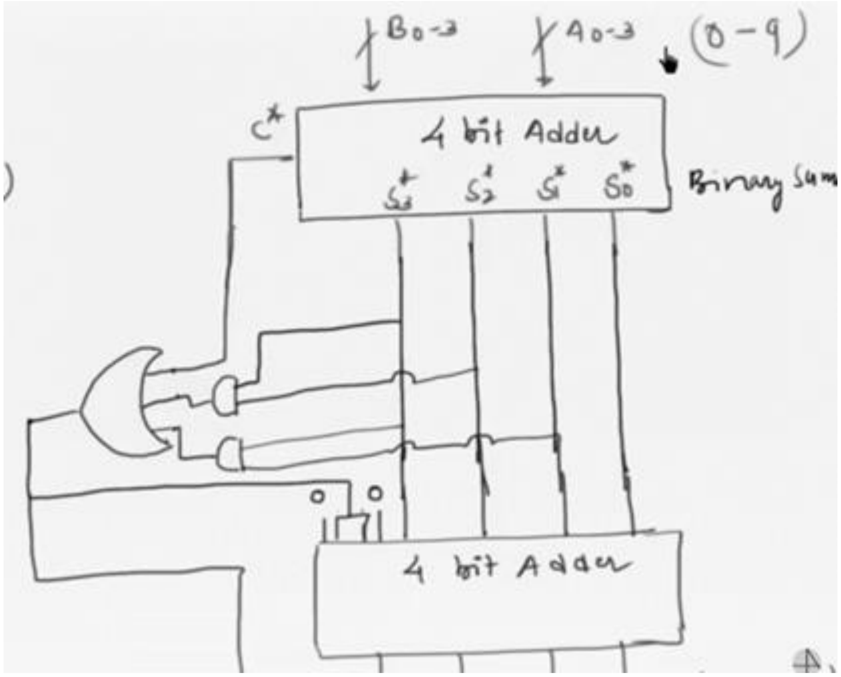
```
         0111   0101
       + 0011   0101
  ------------------
         1010   1010
       +0110  +0110
  ------------------
  0001   0001   0000
    1      1      0
```

Both left and right BCD numbers are invalid. So we would add 6 to both the BCD numbers.

```
  75
+ 35
----
 110
```

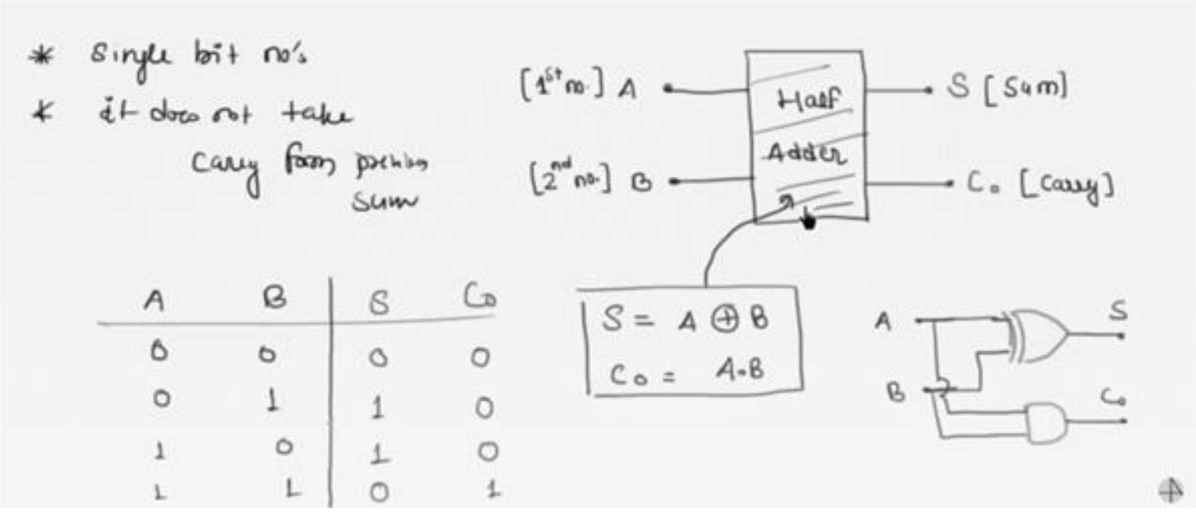
www.vlsifacts.com

BCD ADDER



ADDER

HALF ADDER



* Single bit no's

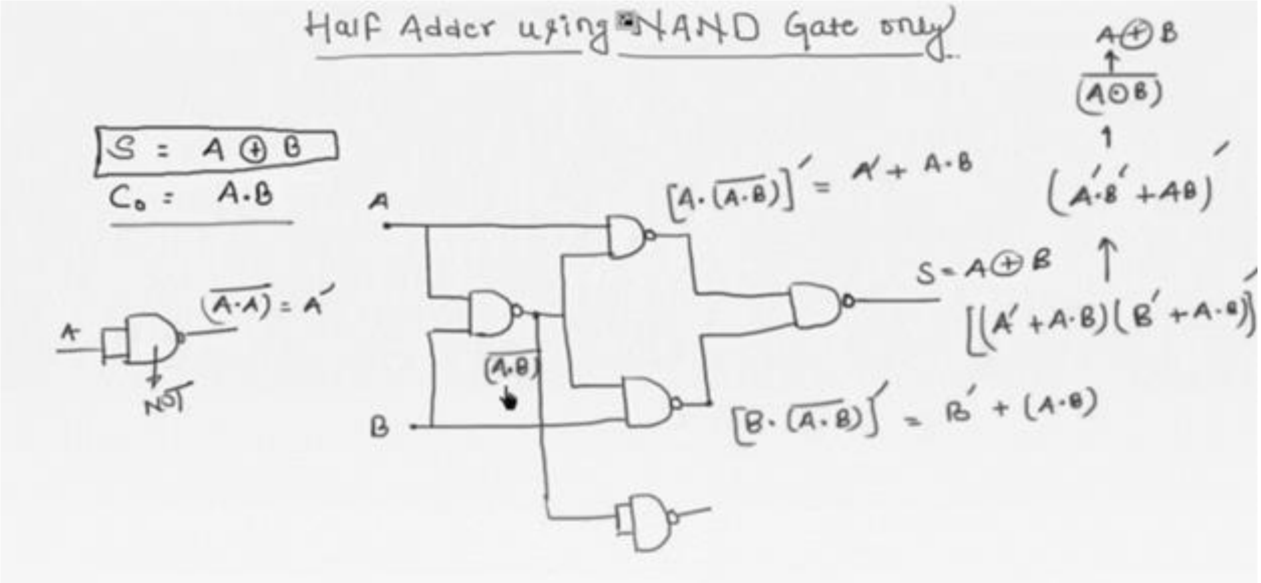
* it does not take carry from previous sum

A	B	S	Co
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

$S = A \oplus B$

$C_o = A \cdot B$

USING NAND, NOR

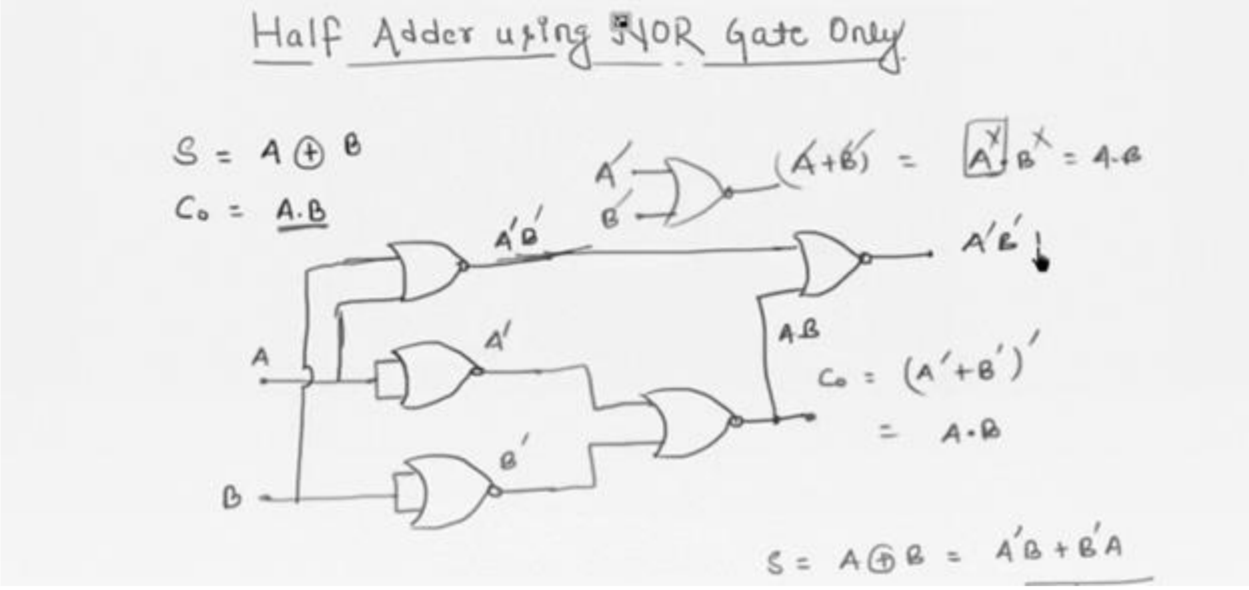


Half Adder using NAND Gate only

$S = A \oplus B$

$C_o = A \cdot B$

A
$(\overline{A \cdot A}) = A'$
NOT
A
B
$(\overline{A \cdot B})$
$[A \cdot (\overline{A \cdot B})]' = A' + A \cdot B$
$[B \cdot (\overline{A \cdot B})]' = B' + (A \cdot B)$
$S = A \oplus B$
$[(A' + A \cdot B)(B' + A \cdot B)]'$
$(A' \cdot B' + AB)'$
$(\overline{A \odot B})$
$A \oplus B$



Half Adder using NOR Gate Only

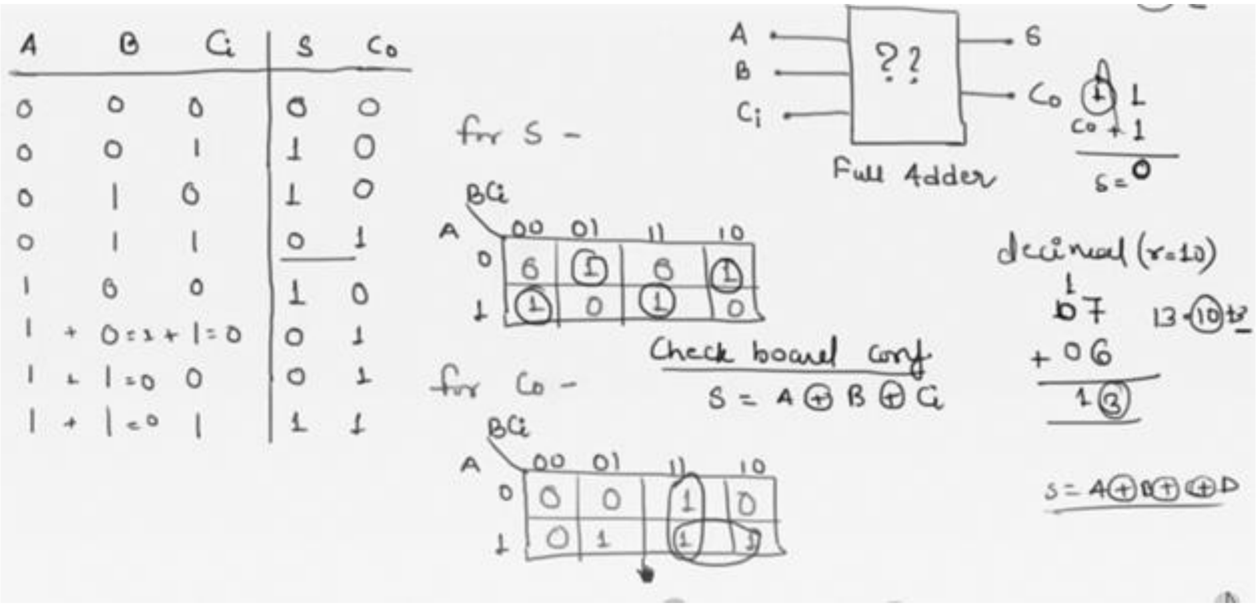
$S = A \oplus B$

$C_o = A.B$

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

$S = A \oplus B = A'B + B'A$

FULL ADDER



A	B	Ci	S	Co
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1 +	0=1 +	1=0	0	1
1 +	1=0	0	0	1
1 +	1=0	1	1	1

for S -

A \ BCi	00	01	11	10
0	0	1	0	1
1	1	0	1	0

Check board conf

$S = A \oplus B \oplus C_i$

for Co -

A \ BCi	00	01	11	10
0	0	0	1	0
1	0	1	1	1

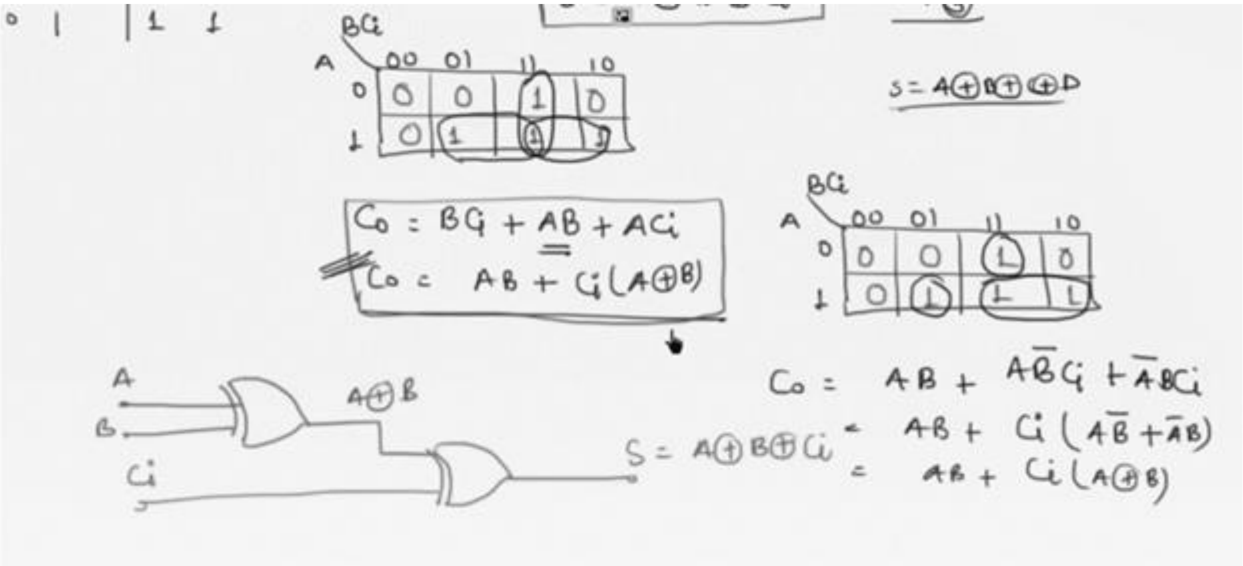
$$\begin{array}{r} C_o\ 1\ 1 \\ C_o + 1 \\ \hline s = 0 \end{array}$$

decimal (r=10)

$$\begin{array}{r} 1 \\ 07 \\ +06 \\ \hline 13 \end{array}$$

13-10=3

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



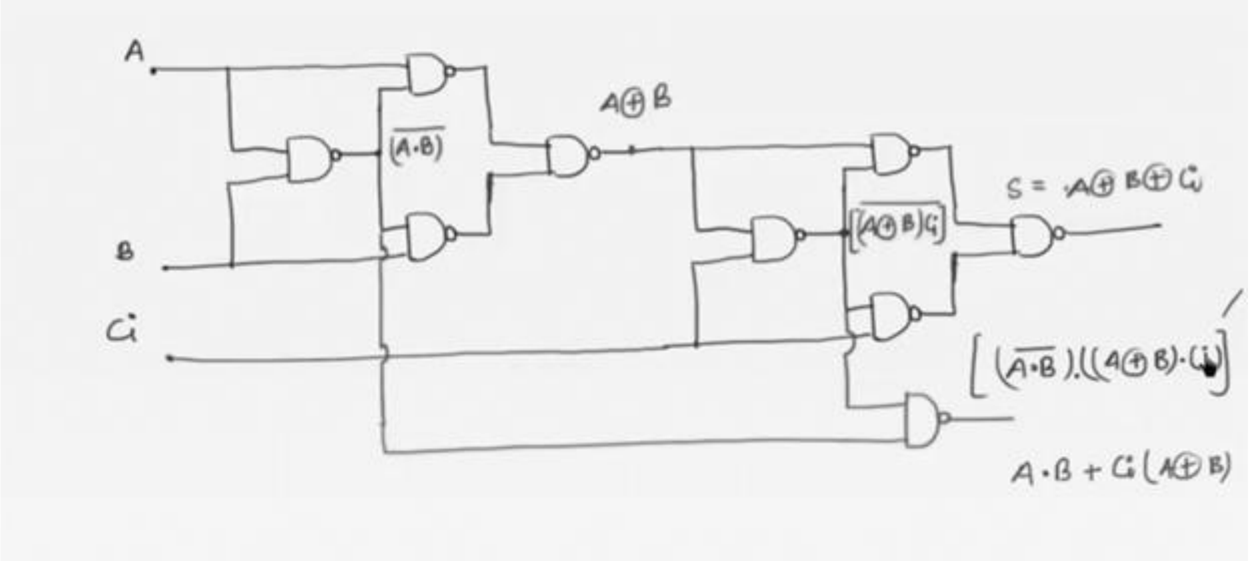
A \ BCi	00	01	11	10
0	0	0	1	0
1	0	1	1	1

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

$$\boxed{\begin{aligned} C_o &= BC_i + AB + AC_i \\ C_o &= AB + C_i(A \oplus B) \end{aligned}}$$

A \ BCi	00	01	11	10
0	0	0	1	0
1	0	1	1	1

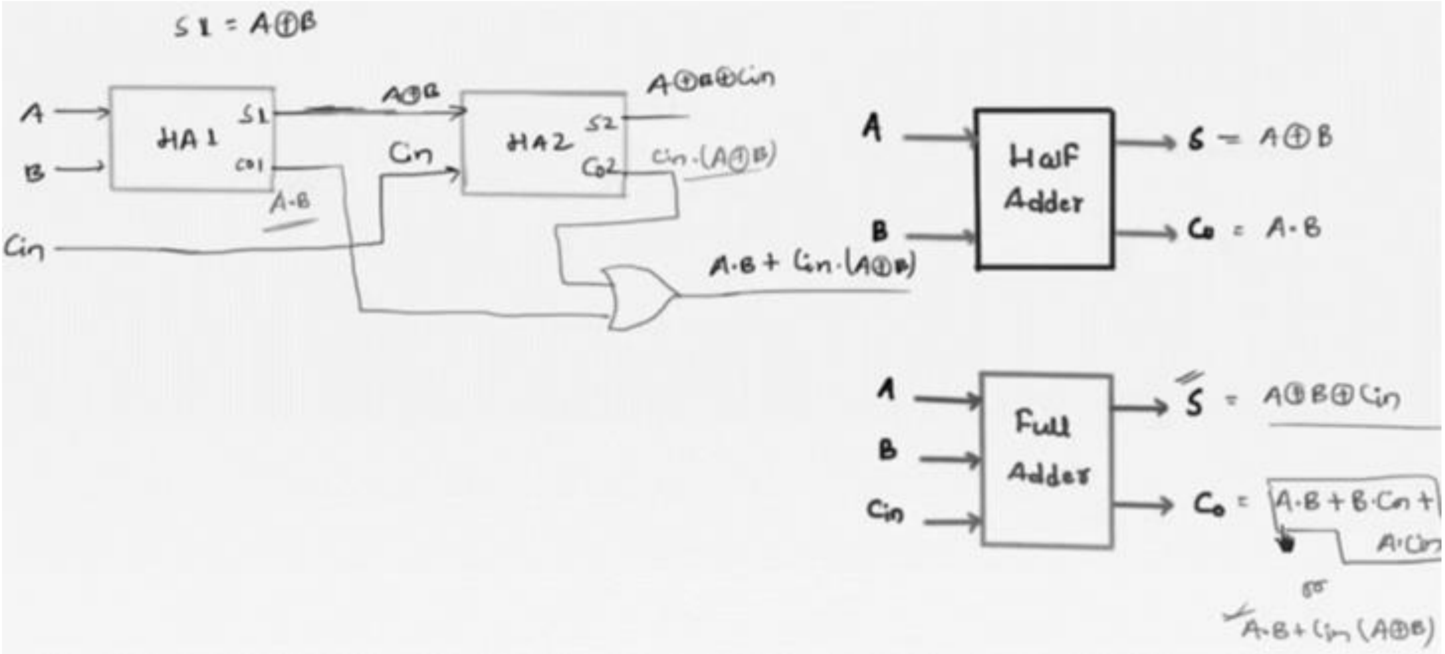
$$\begin{aligned} C_o &= AB + A\overline{B}C_i + \overline{A}BC_i \\ &= AB + C_i(A\overline{B} + \overline{A}B) \\ &= AB + C_i(A \oplus B) \end{aligned}$$

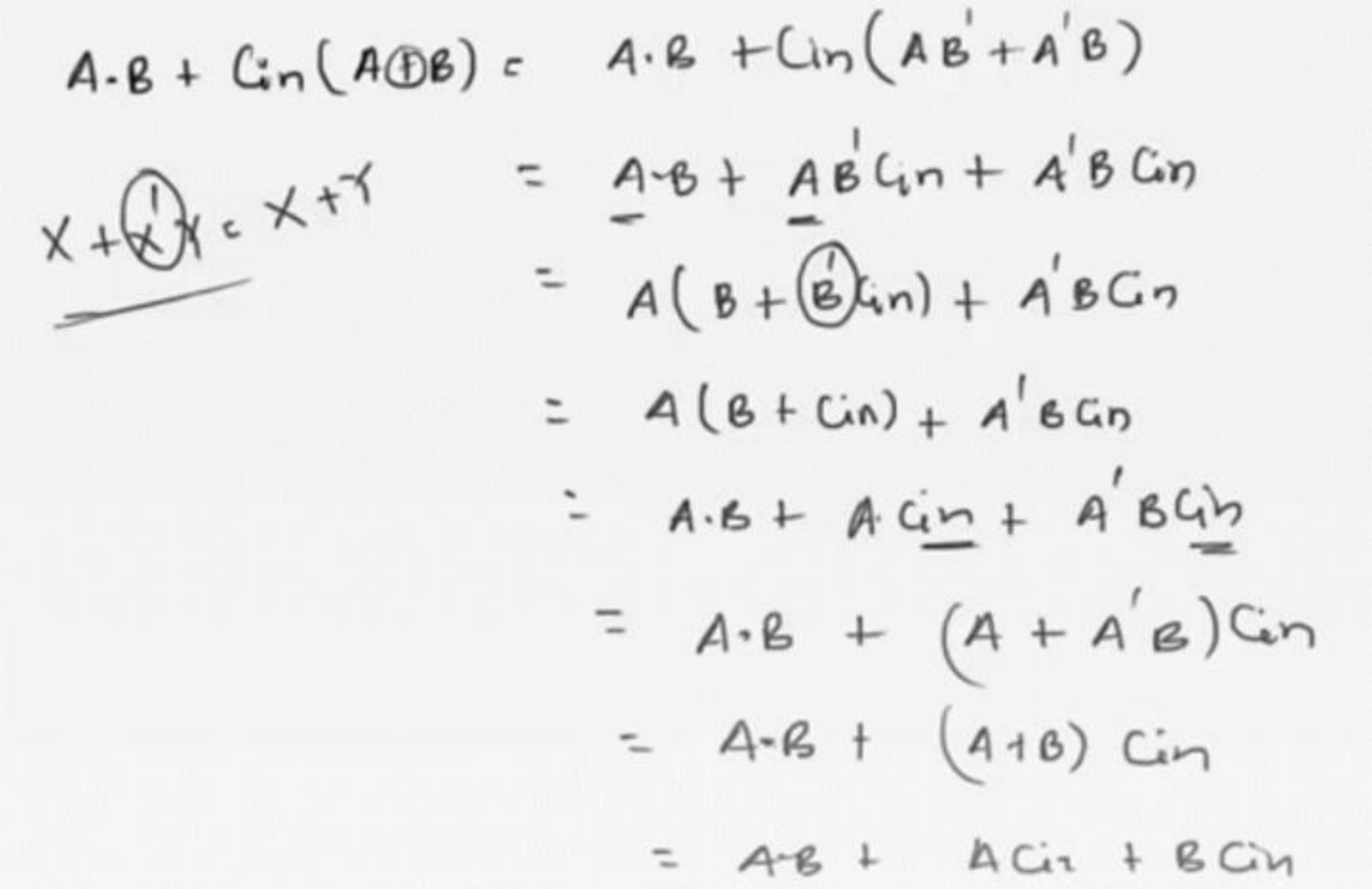


$\left[\overline{(\overline{A \cdot B}) . (\overline{(A \oplus B) \cdot C_i})}\right]$

$A \cdot B + C_i(A \oplus B)$

FULL ADDER USING HALF ADDER





$$A\cdot B + C_{in}(A \oplus B) = A\cdot B + C_{in}(AB' + A'B)$$

$$X + \boxed{X'}Y = X + Y$$

$$= \underline{A}\cdot B + \underline{A}B'C_{in} + A'BC_{in}$$

$$= A(B + \boxed{B'}C_{in}) + A'BC_{in}$$

$$= A(B + C_{in}) + A'BC_{in}$$

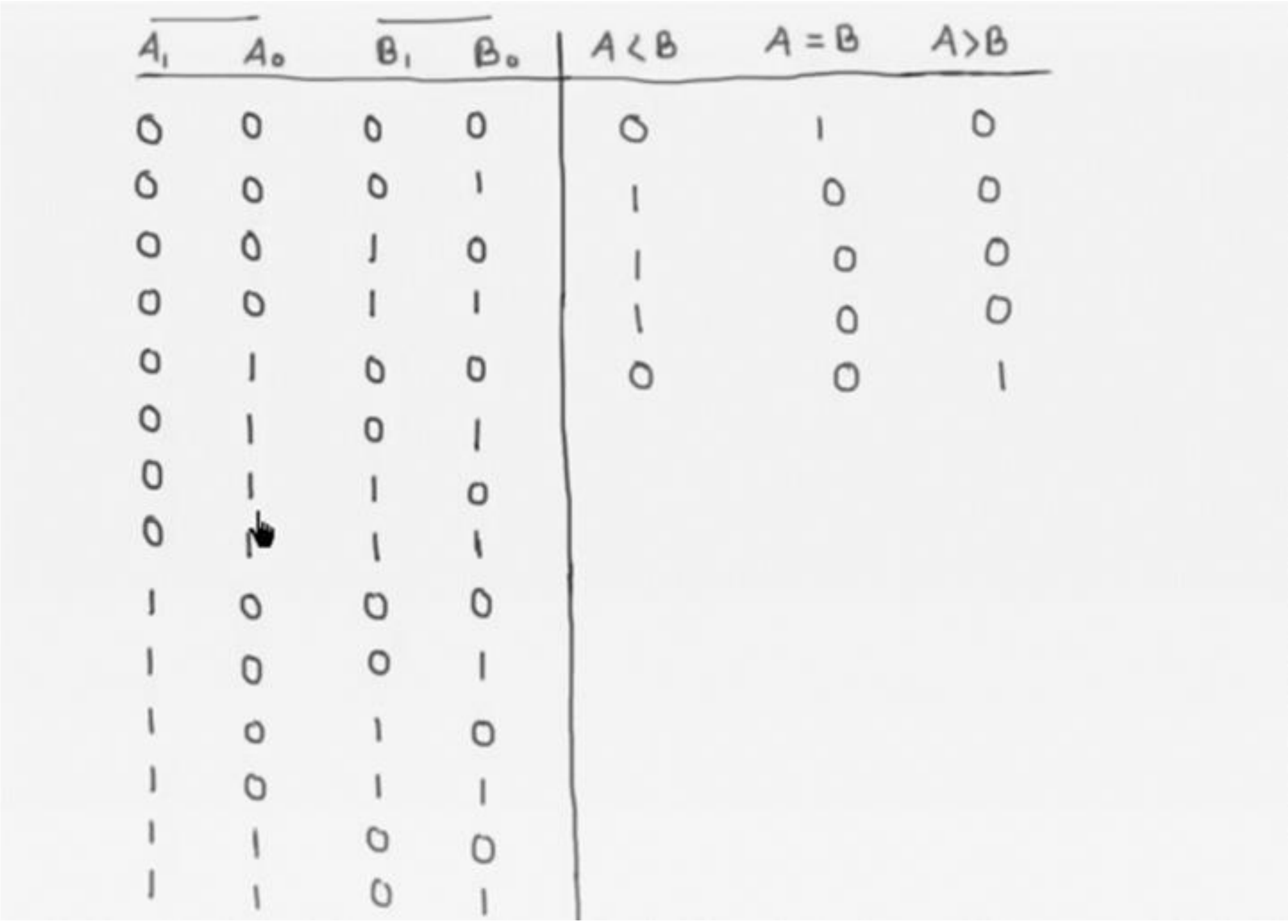
$$= A\cdot B + A\,\underline{C_{in}} + A'B\underline{C_{in}}$$

$$= A\cdot B + (A + A'B)C_{in}$$

$$= A\cdot B + (A + B)\,C_{in}$$

$$= A\cdot B + AC_{in} + BC_{in}$$

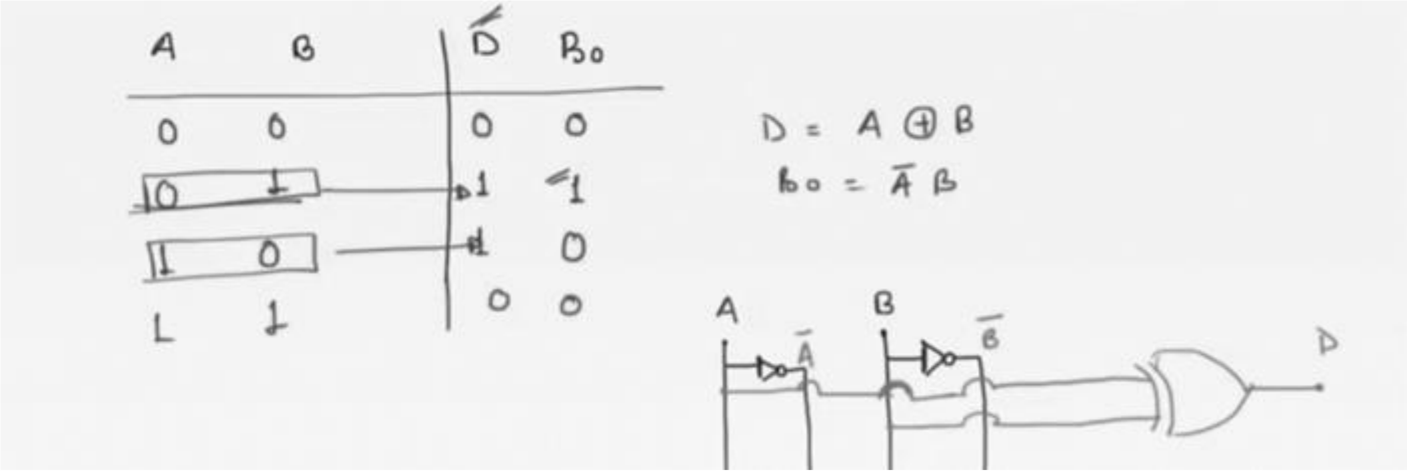
2 BIT COMPARATOR



A_1	A_0	B_1	B_0	A<B	A=B	A>B
0	0	0	0	0	1	0
0	0	0	1	1	0	0
0	0	1	0	1	0	0
0	0	1	1	1	0	0
0	1	0	0	0	0	1
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			

SUBTRACTOR

HALF SUBTRACTOR

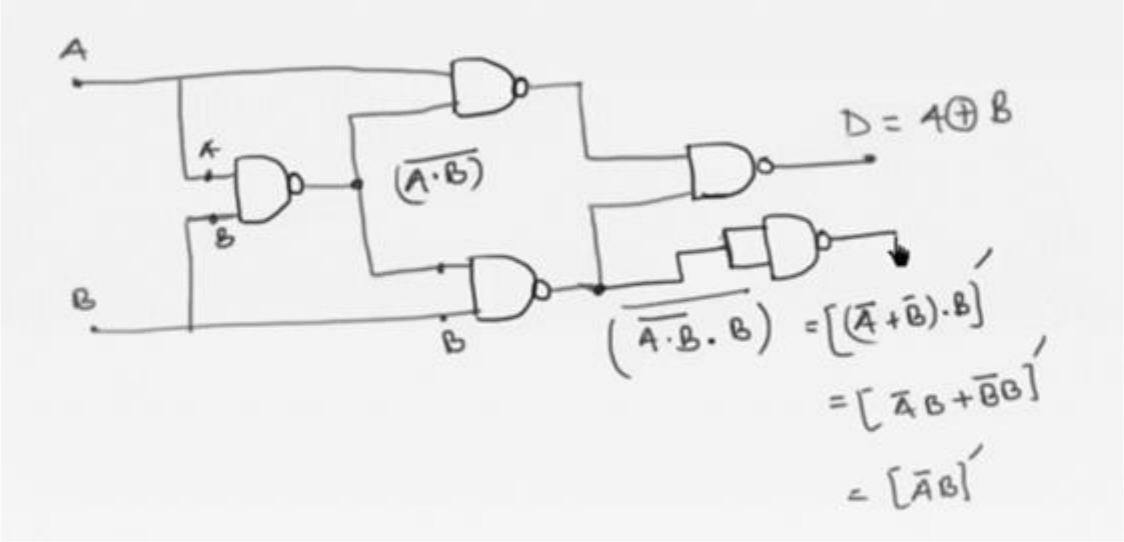


A	B	D	Bo
0	0	0	0
0	1	1	1
1	0	1	0
L	1	0	0

$D = A \oplus B$

$Bo = \bar{A} B$

HS USING NAND



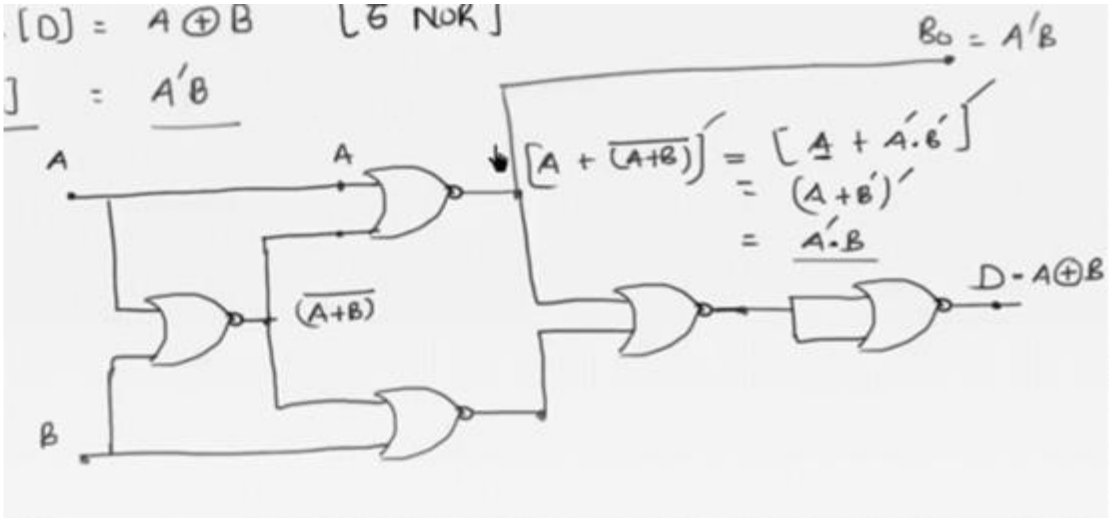
$D = A \oplus B$

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

$= [\bar{A}B + \bar{B}B]'$

$= [\bar{A}B]'$

HA USING NOR



$[D] = A \oplus B$ [5 NOR]

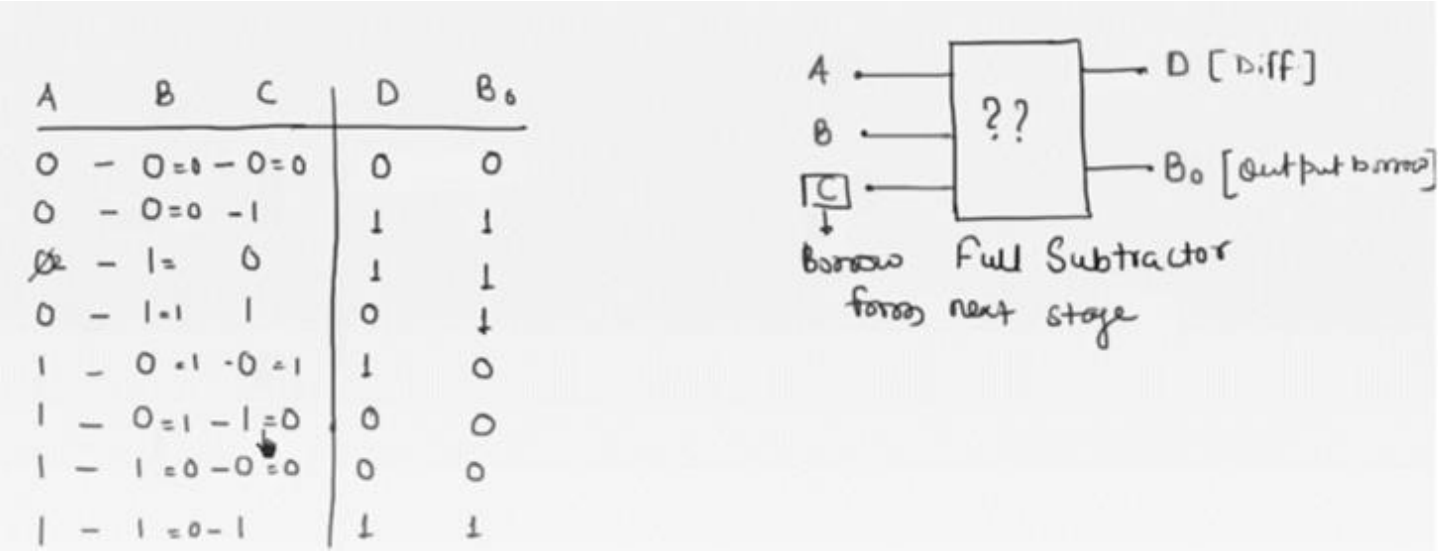
$] = A'B$

$[A + \overline{(A+B)}]' = [A + A'.B']'$

$= (A+B')'$

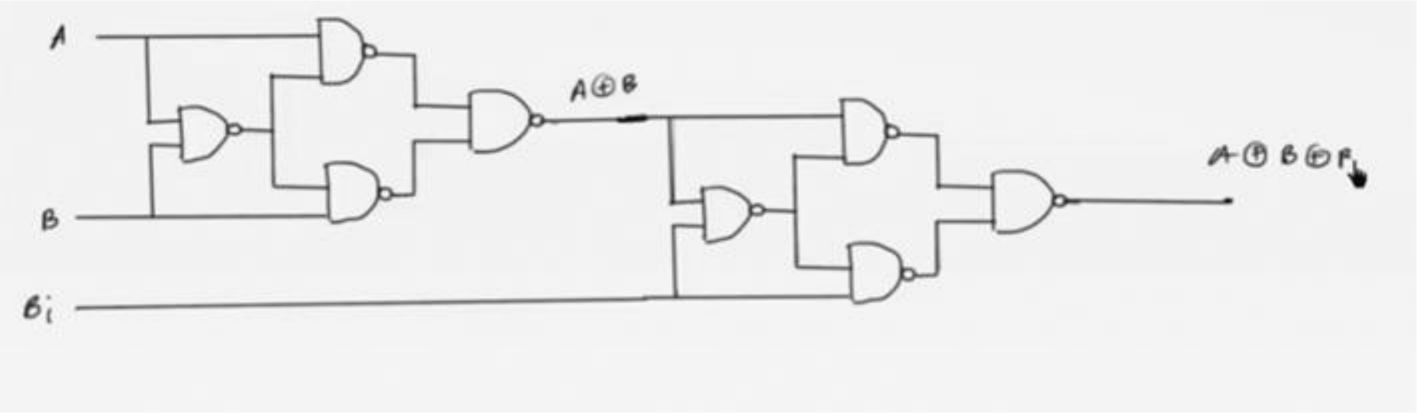
$= A'.B$

FULL SUBTRACTOR

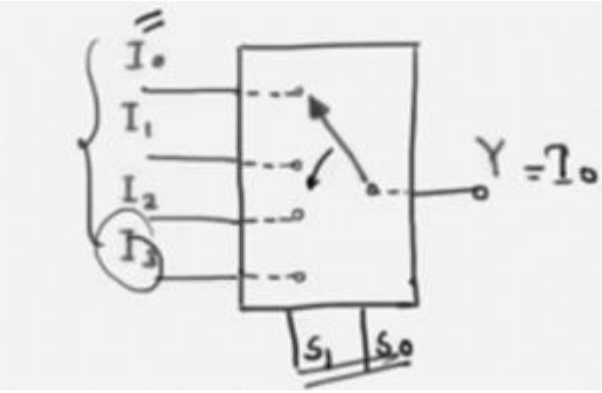


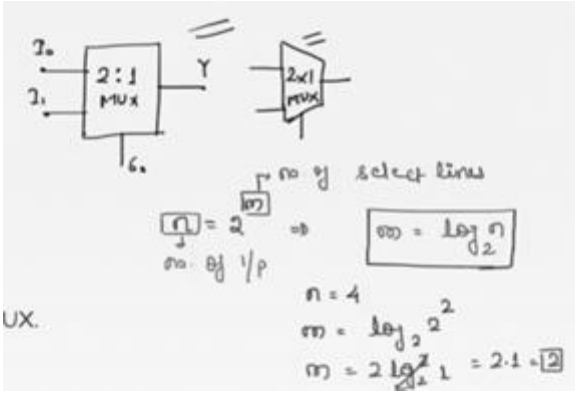
A	B	C	D	B_o
0	− 0=0	− 0=0	0	0
0	− 0=0	−1	1	1
~~0~~2	− 1=	0	1	1
0	− 1=1	1	0	1
1	− 0 =1	−0 =1	1	0
1	− 0=1	− 1 =0	0	0
1	− 1 =0	−0 =0	0	0
1	− 1 =0	− 1	1	1

FS USING NAND



MUX





$n = 2^{m}$ → no of select lines; n → no. of I/P

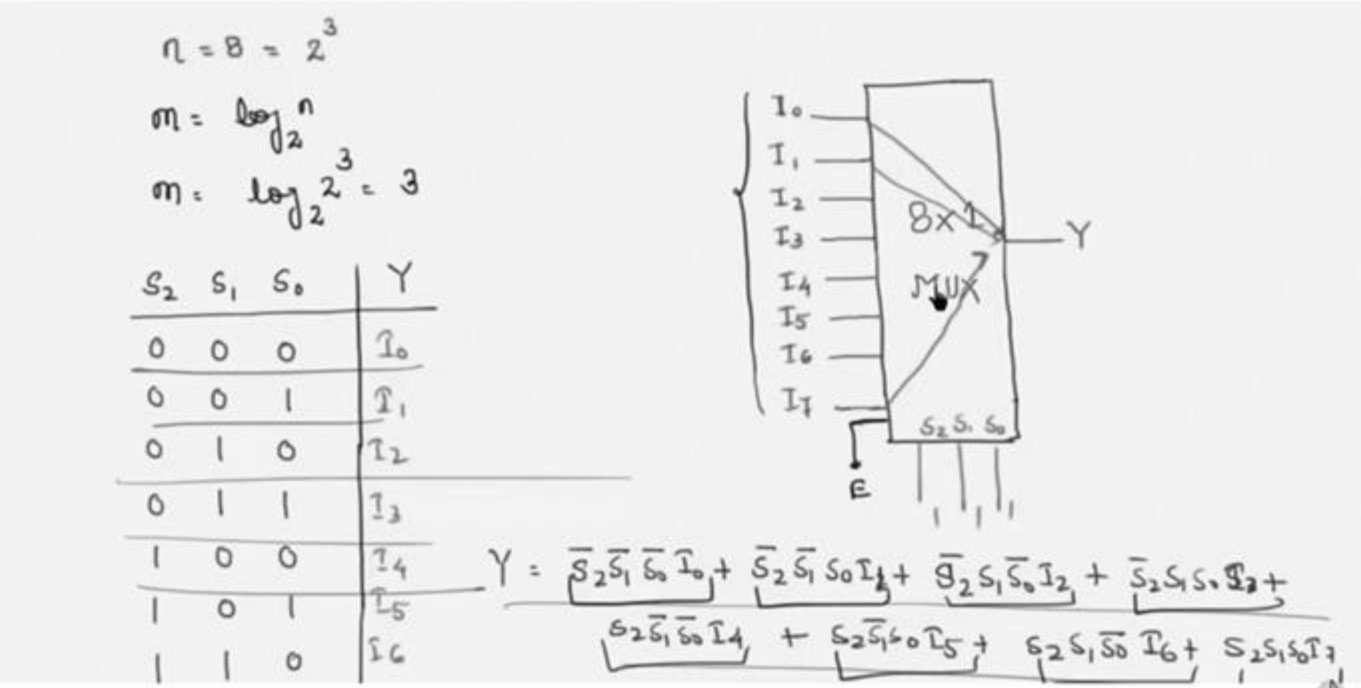
$\Rightarrow \boxed{m = \log_2 n}$

$n = 4$

$m = \log_2 2^2$

$m = 2\log_2 2 = 2 \cdot 1 = \boxed{2}$

8X1 MUX



$n = 8 = 2^3$

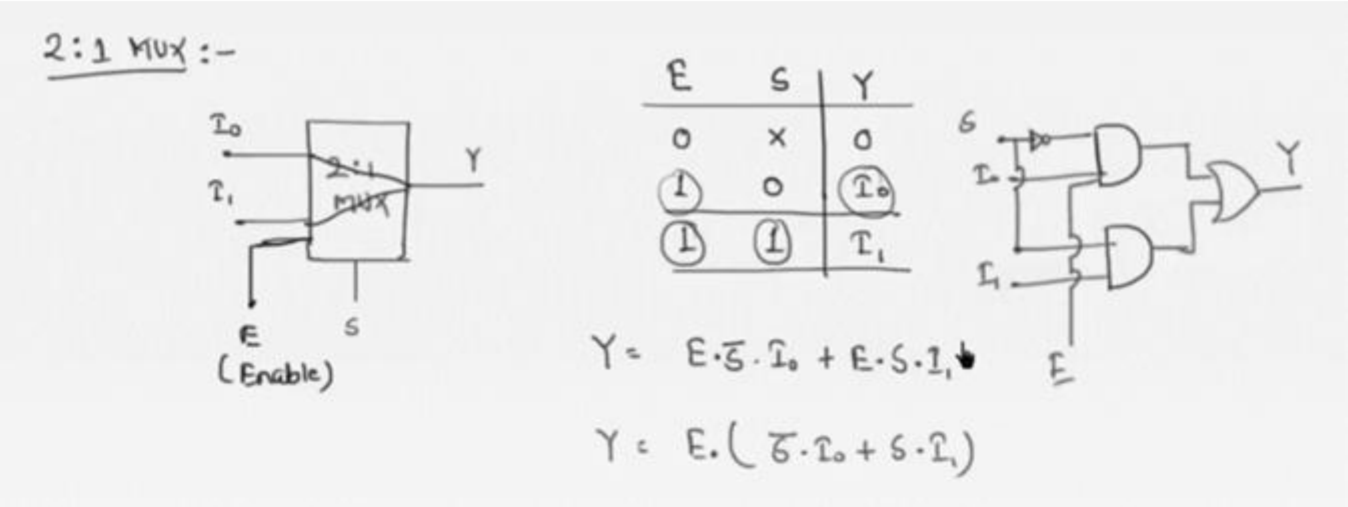
$m = \log_2 n$

$m = \log_2 2^3 = 3$

S_2	S_1	S_0	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

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

2X1



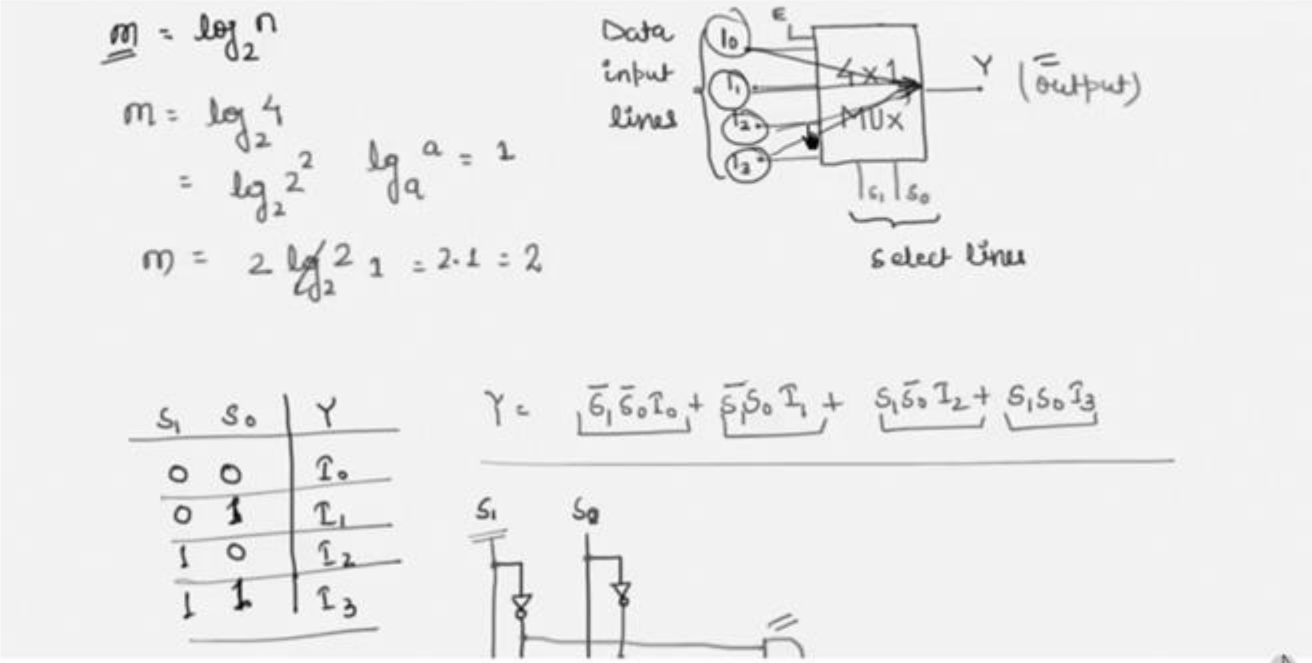
2:1 MUX :-

E	S	Y
0	×	0
1	0	I_0
1	1	I_1

$Y = E \cdot \bar{S} \cdot I_0 + E \cdot S \cdot I_1$

$Y = E \cdot (\bar{S} \cdot I_0 + S \cdot I_1)$

4X1



$m = \log_2 n$

$m = \log_2 4$

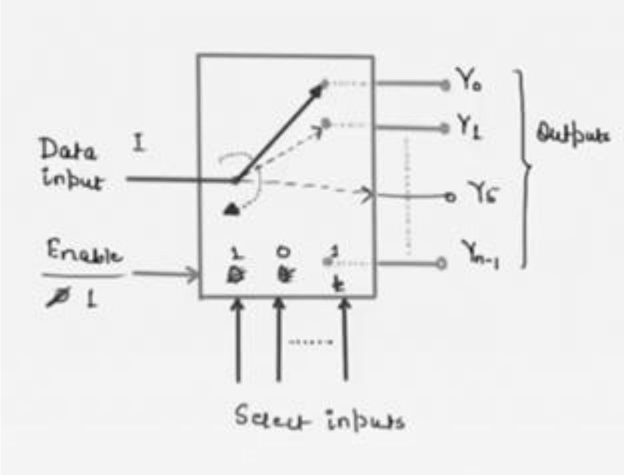
$= \log_2 2^2 \quad \log_a a = 1$

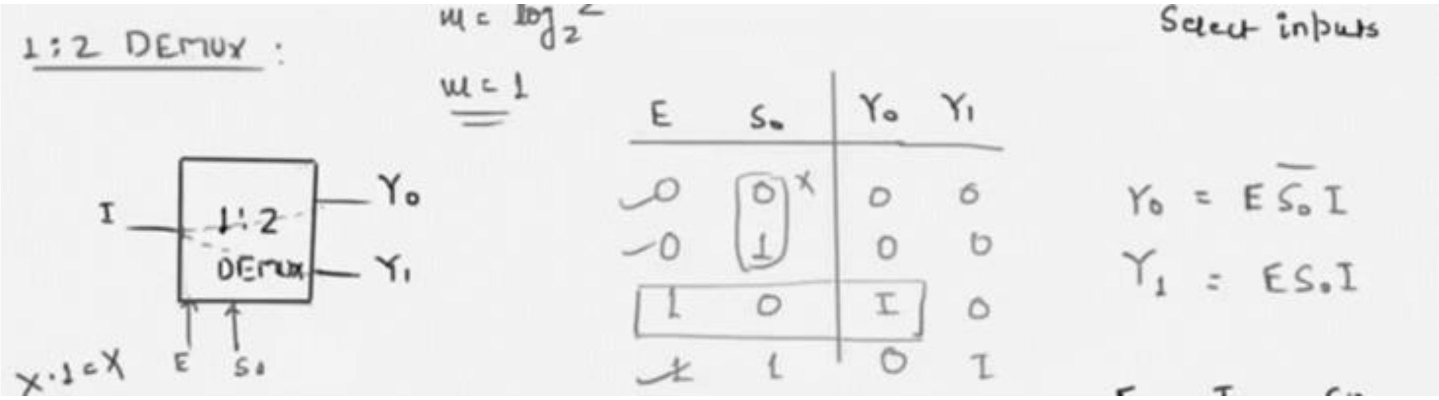
$m = 2 \cancel{\log_2 2}\, 1 = 2 \cdot 1 = 2$

S_1	S_0	Y
0	0	I_0
0	1	I_1
1	0	I_2
1	1	I_3

$Y = \underbrace{\bar{S}_1 \bar{S}_0 I_0} + \underbrace{\bar{S}_1 S_0 I_1} + \underbrace{S_1 \bar{S}_0 I_2} + \underbrace{S_1 S_0 I_3}$

DEMUX





1:2 DEMUX :

$m = \log_2 2$

$m = 1$

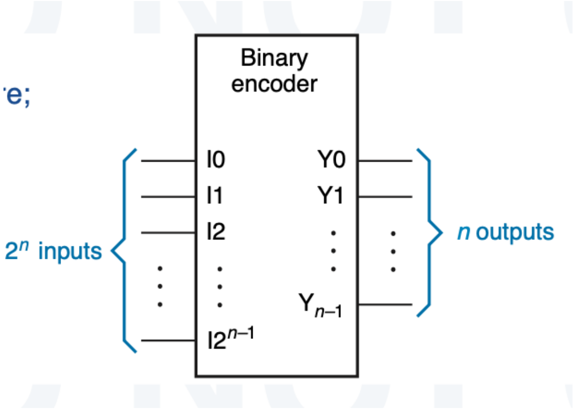
$x \cdot 1 = x$

E	S_0	Y_0	Y_1
0	0	0	0
0	1	0	0
1	0	I	0
1	1	0	I

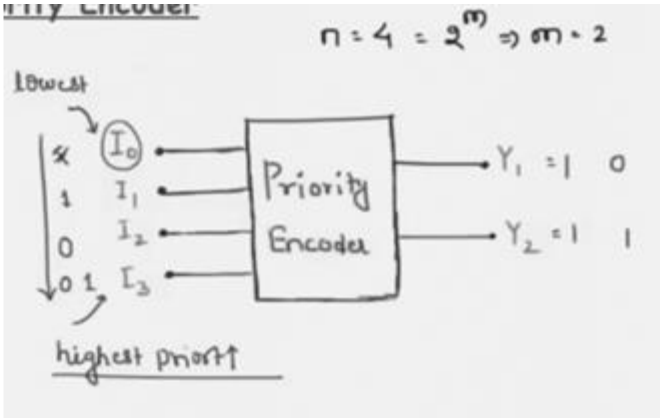
$Y_0 = E \bar{S}_0 I$

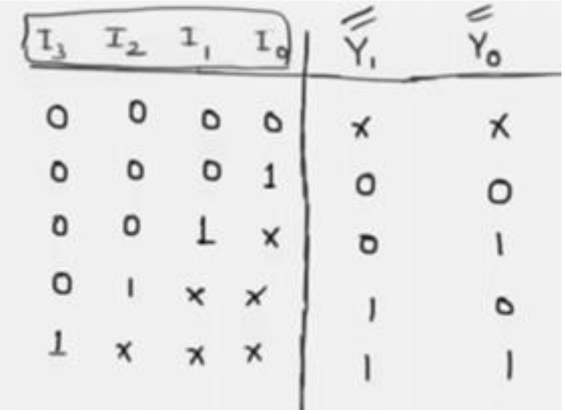
$Y_1 = E S_0 I$

ENCODER

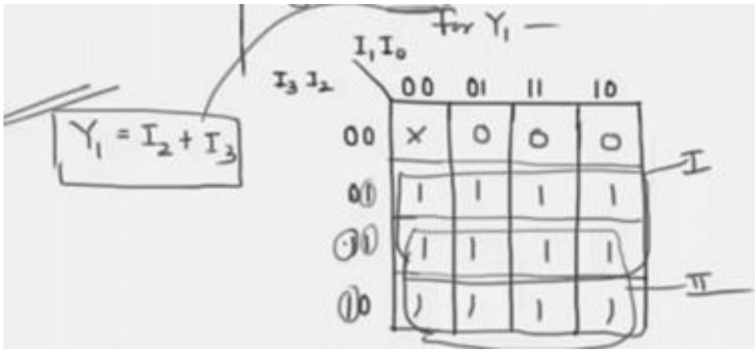


PRIORITY ENCODER





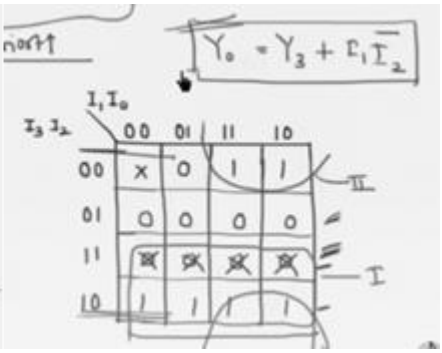
I_3	I_2	I_1	I_0	Y_1	Y_0
0	0	0	0	x	x
0	0	0	1	0	0
0	0	1	x	0	1
0	1	x	x	1	0
1	x	x	x	1	1



for Y_1 —

$$\boxed{Y_1 = I_2 + I_3}$$

I_3I_2 \ I_1I_0	00	01	11	10
00	x	0	0	0
01	1	1	1	1
11	1	1	1	1
10	1	1	1	1

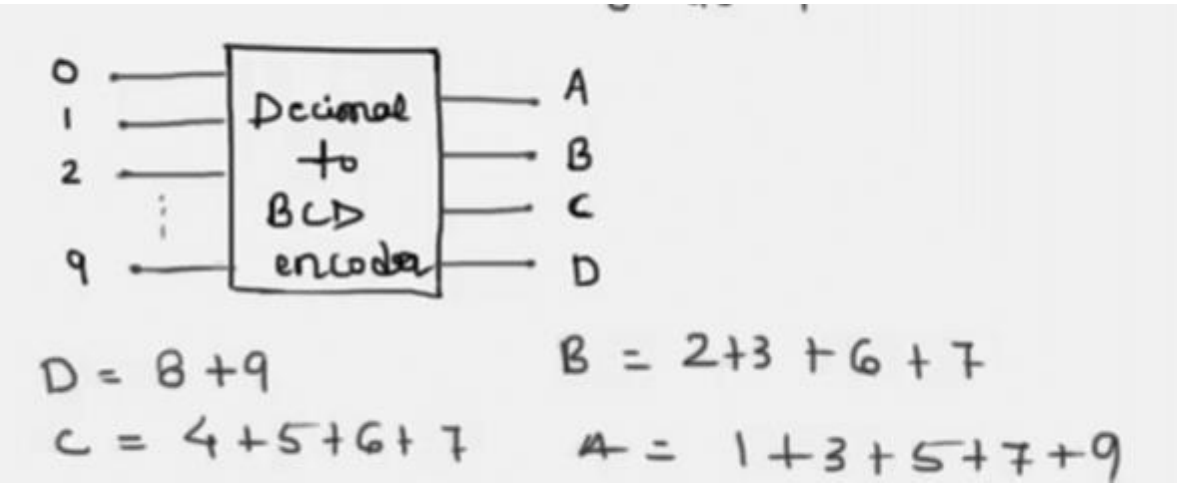


priorit

$$\boxed{Y_0 = Y_3 + I_1\bar{I}_2}$$

I_3I_2 \ I_1I_0	00	01	11	10
00	x	0	1	1
01	0	0	0	0
11	X	X	X	X
10	1	1	1	1

DECIMAL TO BCD ENCODER

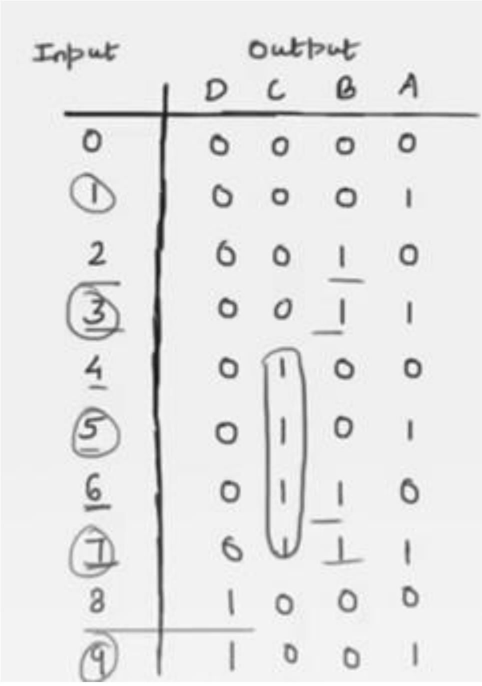


$D = 8 + 9$

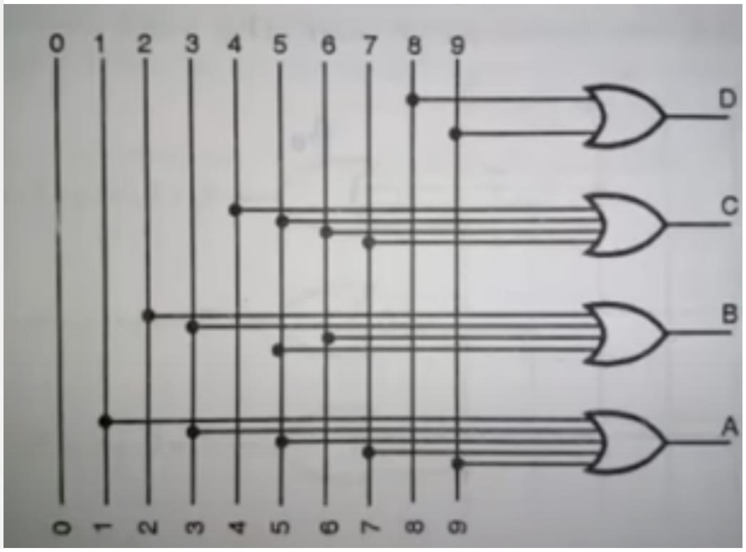
$B = 2 + 3 + 6 + 7$

$C = 4 + 5 + 6 + 7$

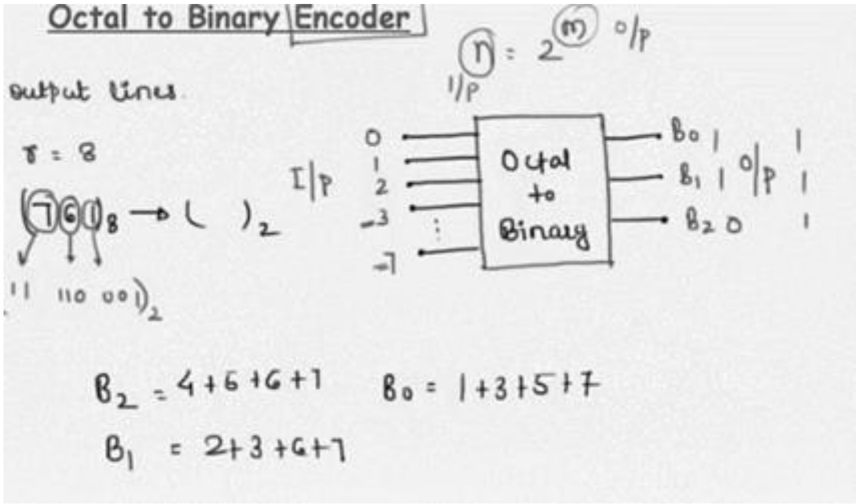
$A = 1 + 3 + 5 + 7 + 9$



Input	D	C	B	A
0	0	0	0	0
1	0	0	0	1
2	0	0	1	0
3	0	0	1	1
4	0	1	0	0
5	0	1	0	1
6	0	1	1	0
7	0	1	1	1
8	1	0	0	0
9	1	0	0	1



OCTAL TO BINARY ENCODER



Octal to Binary Encoder

$n = 2^m$ o/p

I/P

output lines

8 = 8

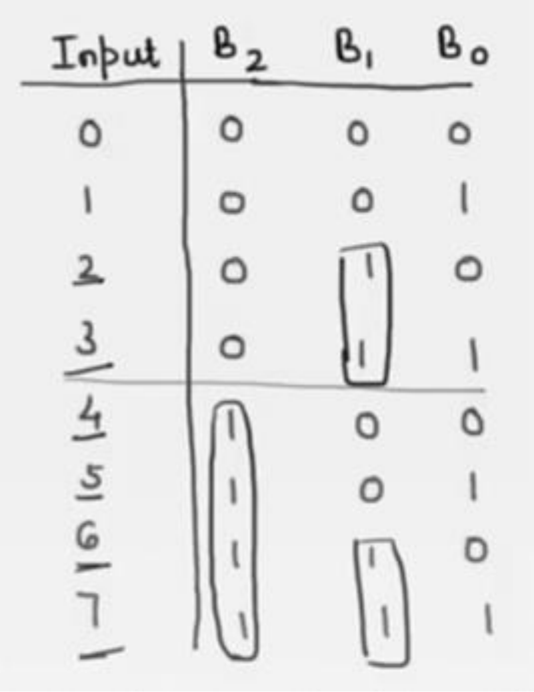
$(760)_8 \rightarrow (\quad)_2$

$11\ 110\ 00)_2$

0
1
2
3
:
7
I/P
Octal to Binary
B0 1 1
B1 1 O/P 1
B2 0 1

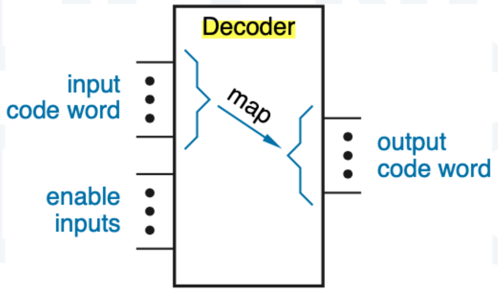
$B_2 = 4+6+6+7$ $B_0 = 1+3+5+7$

$B_1 = 2+3+6+7$

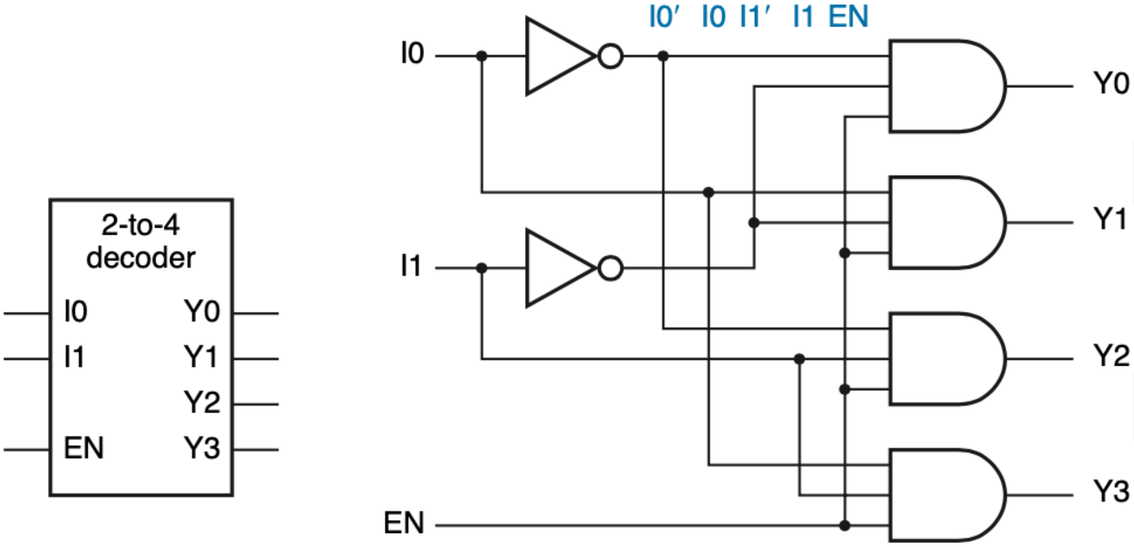


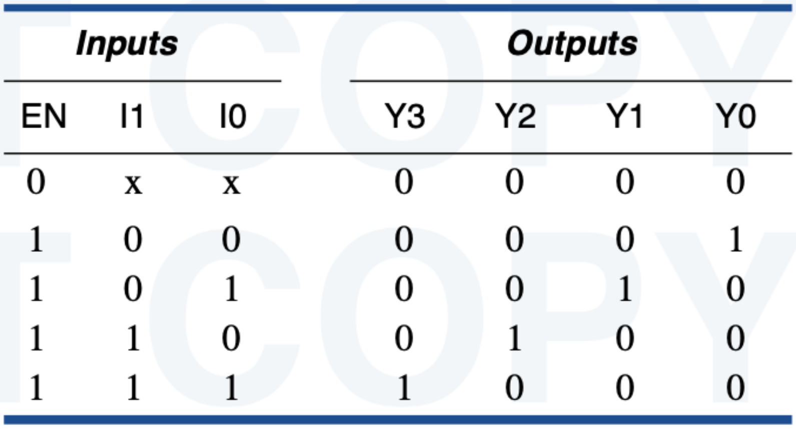
Input	B_2	B_1	B_0
0	0	0	0
1	0	0	1
2	0	1	0
3	0	1	1
4	1	0	0
5	1	0	1
6	1	1	0
7	1	1	1

DECODER



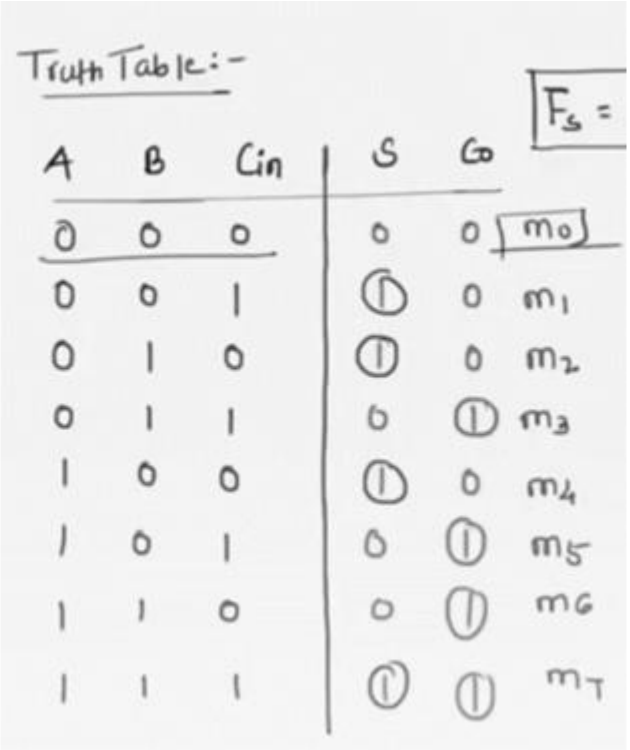
2X4 DECODER





Inputs			*Outputs*			
EN	I1	I0	Y3	Y2	Y1	Y0
0	x	x	0	0	0	0
1	0	0	0	0	0	1
1	0	1	0	0	1	0
1	1	0	0	1	0	0
1	1	1	1	0	0	0

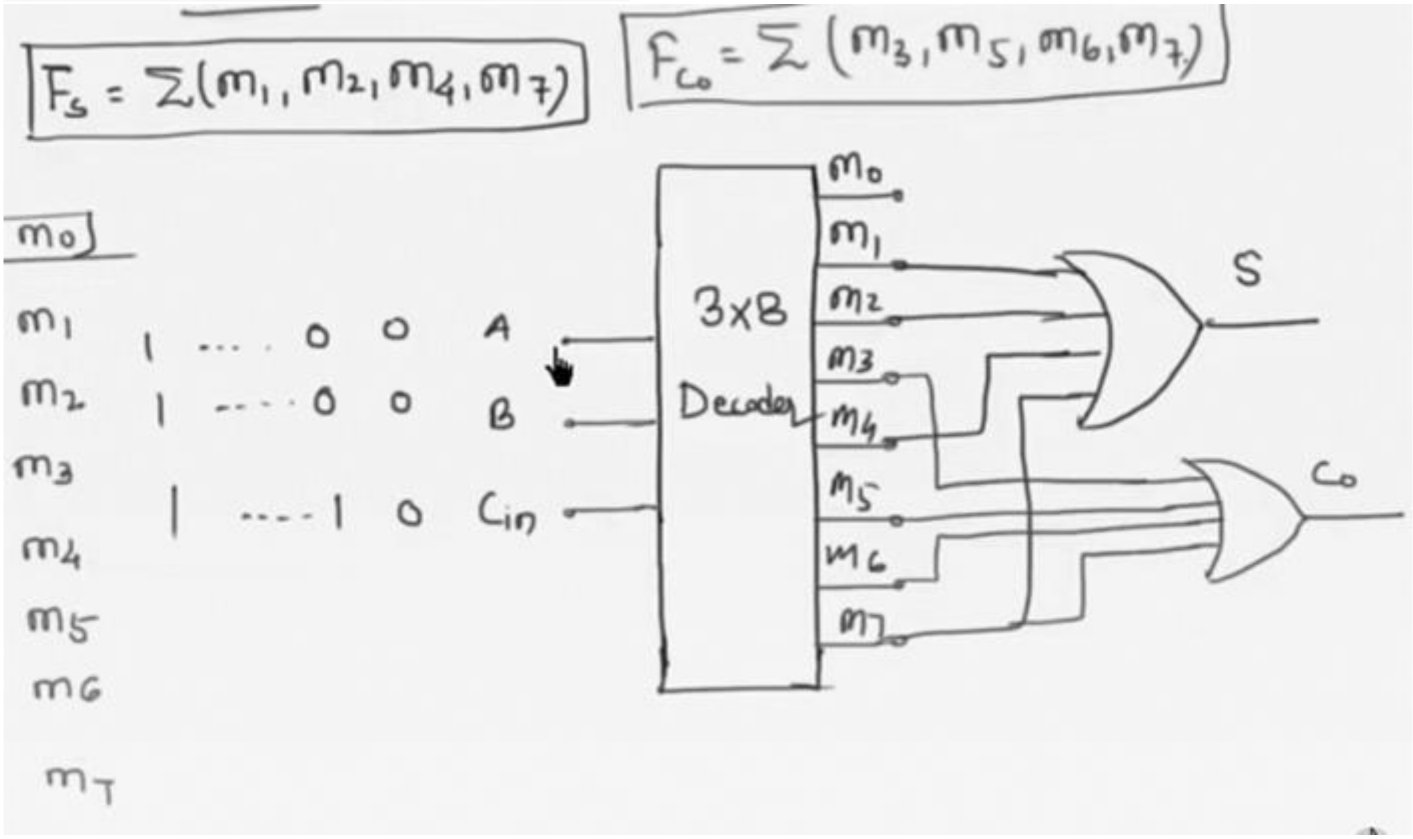
COMBIANTIONAL LOGIC USING DECODER



Truth Table:-

A	B	Cin	S	Co	
0	0	0	0	0	m0
0	0	1	1	0	m1
0	1	0	1	0	m2
0	1	1	0	1	m3
1	0	0	1	0	m4
1	0	1	0	1	m5
1	1	0	0	1	m6
1	1	1	1	1	mT

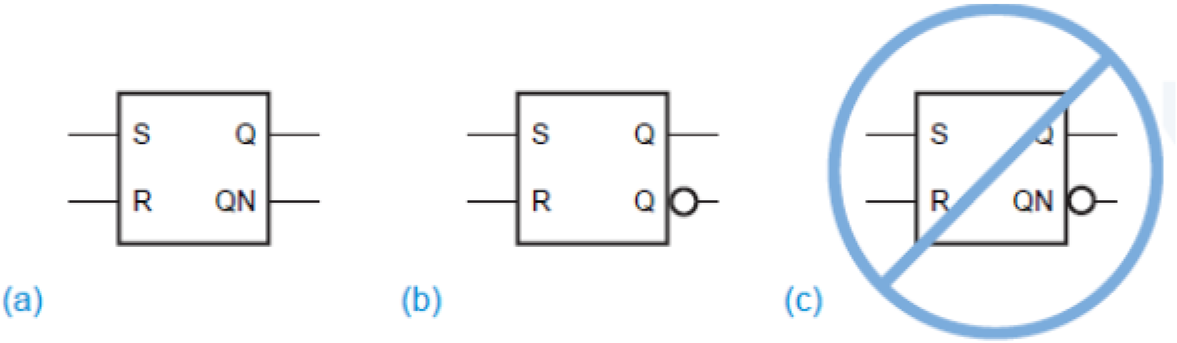
$F_s =$



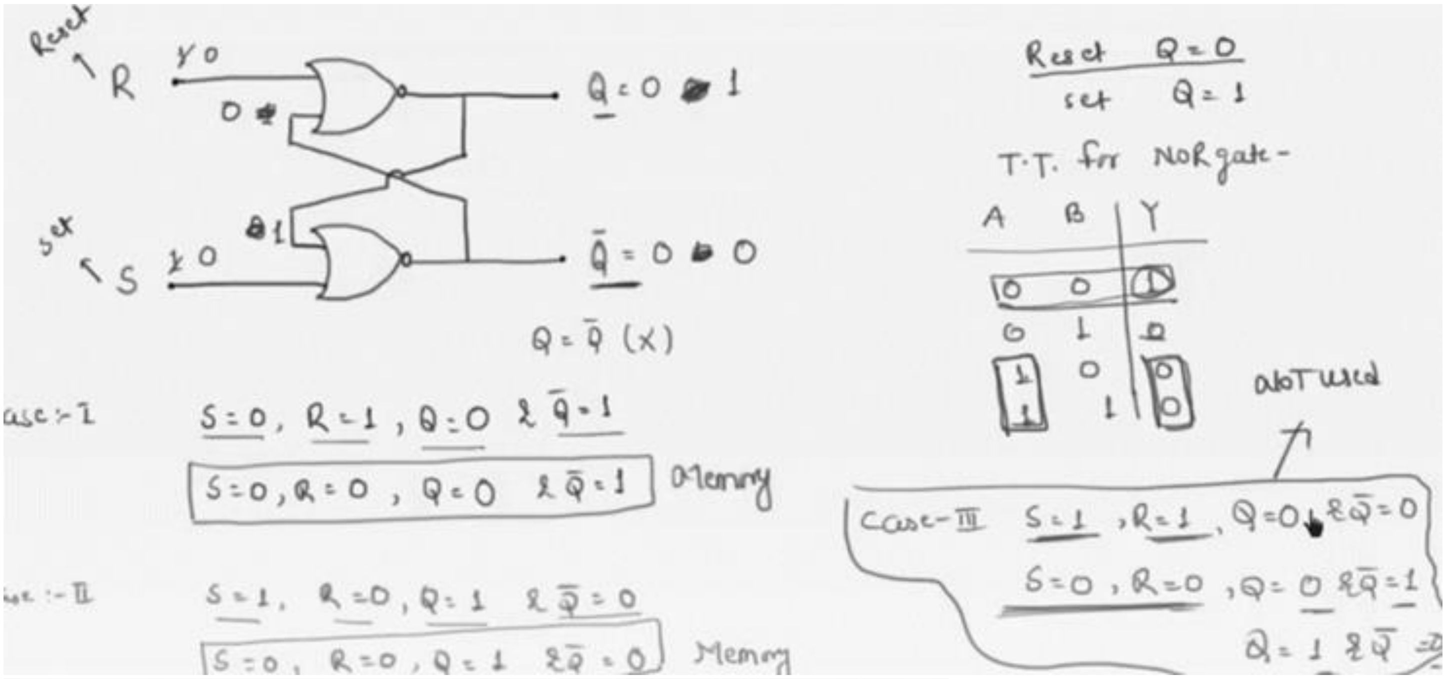
$$\boxed{F_s = \Sigma(m_1, m_2, m_4, m_7)}$$

$$\boxed{F_{Co} = \Sigma(m_3, m_5, m_6, m_7)}$$

SR LATCH



NOR LATCH



$Q = \bar{Q}$ (x)

Reset $Q = 0$

set $Q = 1$

T.T. for NOR gate -

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

not used

case :- I $S = 0$, $R = 1$, $Q = 0$ & $\bar{Q} = 1$

$\boxed{S = 0, R = 0, Q = 0 \ \& \ \bar{Q} = 1}$ Memory

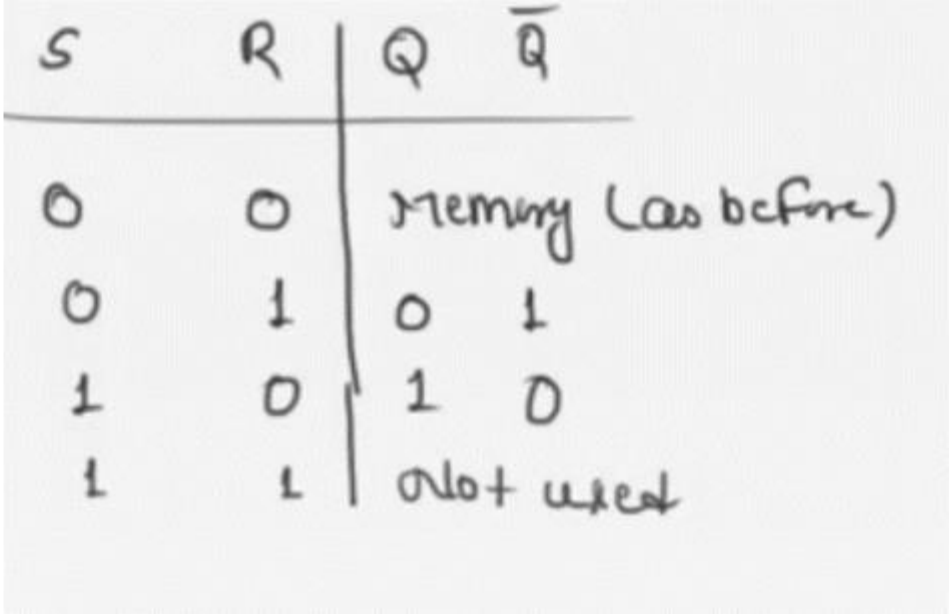
case :- II $S = 1$, $R = 0$, $Q = 1$ & $\bar{Q} = 0$

$\boxed{S = 0, R = 0, Q = 1 \ \& \ \bar{Q} = 0}$ Memory

case - III $S = 1$, $R = 1$, $Q = 0$ & $\bar{Q} = 0$

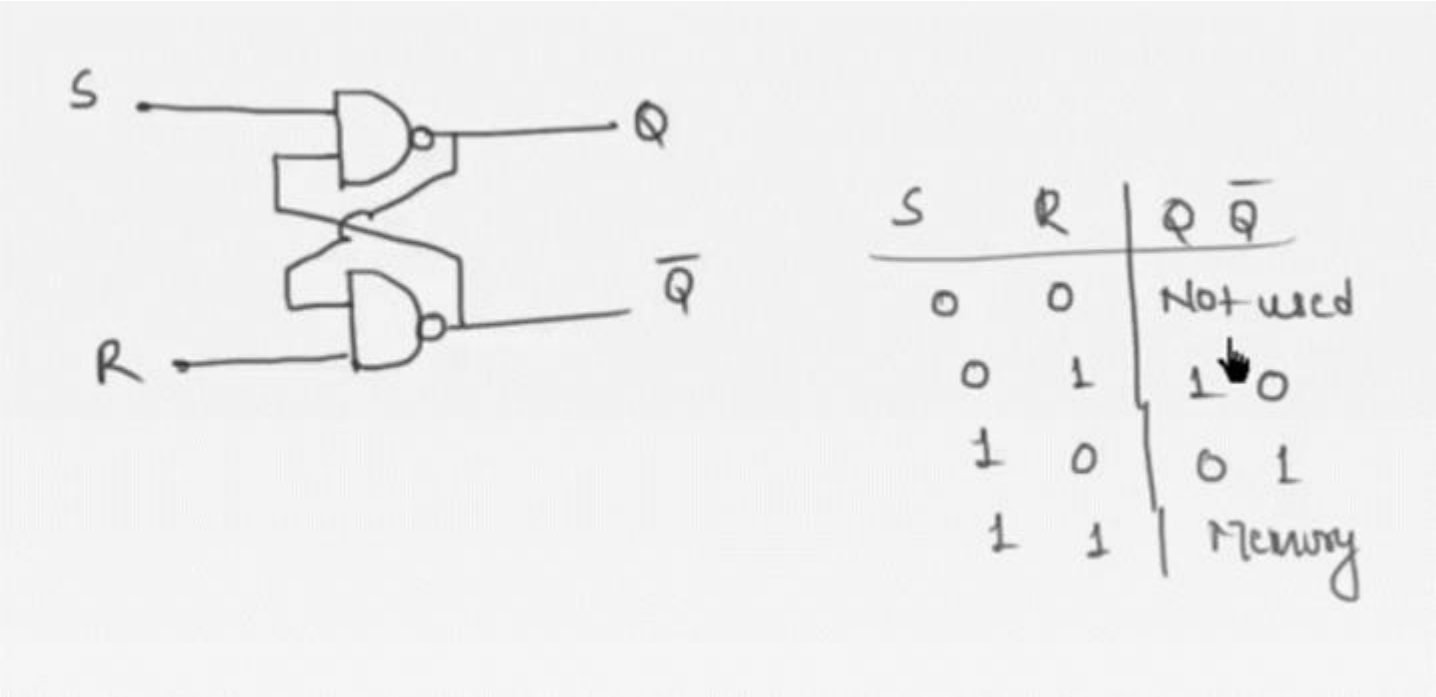
$S = 0$, $R = 0$, $Q = 0$ & $\bar{Q} = 1$

$Q = 1$ & $\bar{Q}$ =



S	R	Q	$\bar{Q}$
0	0	Memory (as before)	
0	1	0	1
1	0	1	0
1	1	Not used	

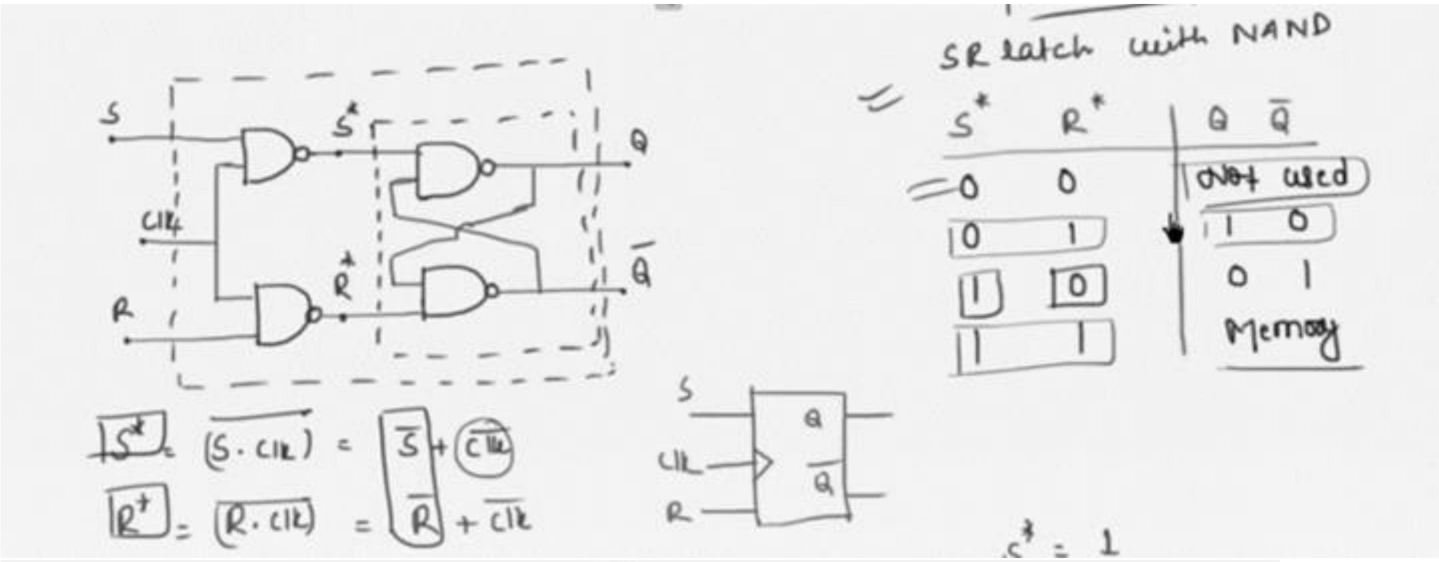
NAND LATCH



S	R	Q	$\bar{Q}$
0	0	Not used	
0	1	1	0
1	0	0	1
1	1	Memory	

FLIPFLOP

SR FF



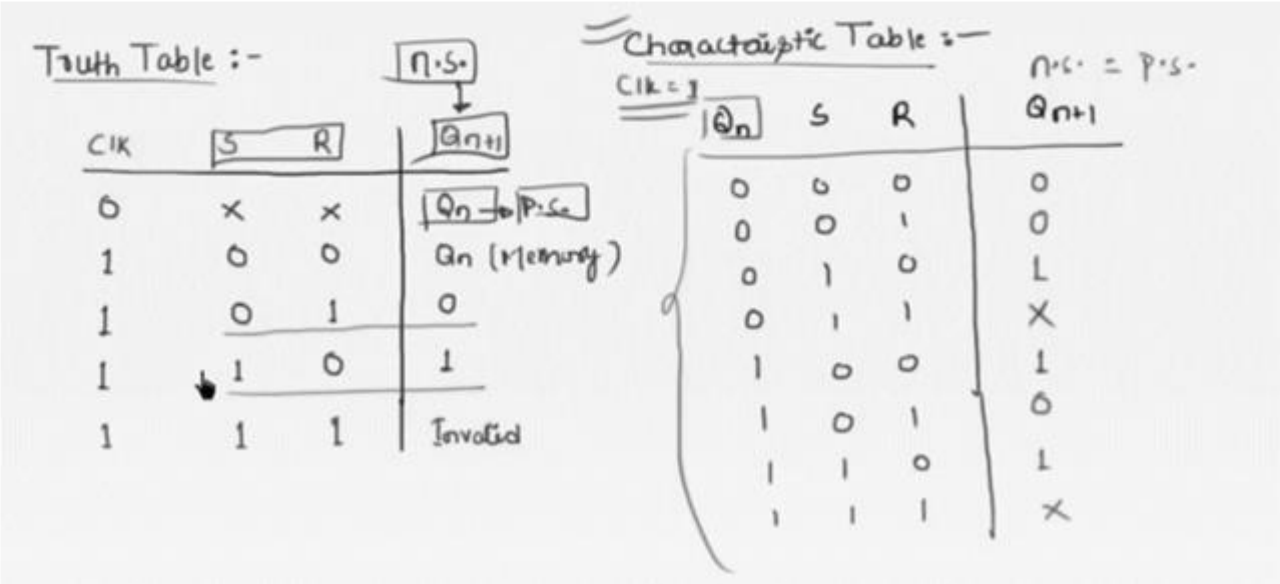
$\boxed{S^*} = \overline{(S\cdot Clk)} = \boxed{\overline{S}} + \overline{Clk}$

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

SR latch with NAND

S^*	R^*	Q	$\overline{Q}$
0	0	not used	
0	1	1	0
1	0	0	1
1	1	Memory	

$S^* = 1$



Truth Table :-

n.s. → Q_{n+1}

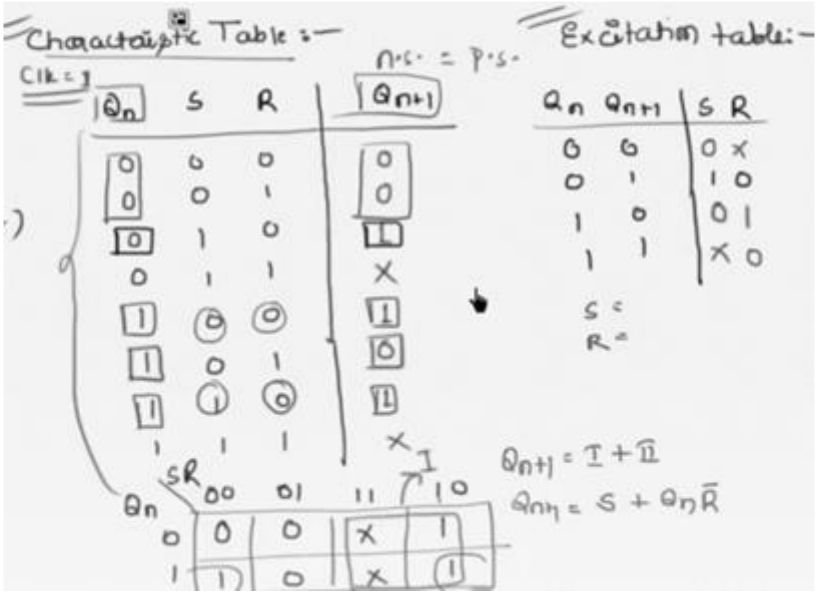
Clk	S	R	Q_{n+1}
0	x	x	Q_n to P.S.
1	0	0	Q_n (Memory)
1	0	1	0
1	1	0	1
1	1	1	Invalid

Characteristic Table :-

n.s. = p.s.

Clk = 1

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



Characteristic Table :-

n.s. = p.s.

Clk = 1

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

Excitation table:-

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

S =

R =

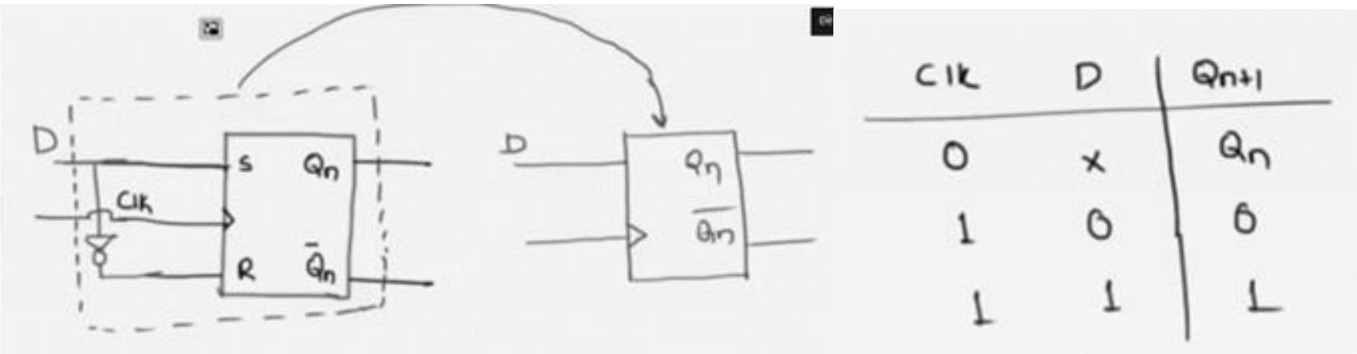
$$\begin{array}{c|cccc} Q_n \backslash SR & 00 & 01 & 11 & 10 \\ \hline 0 & 0 & 0 & X & 1 \\ 1 & 1 & 0 & X & 1 \end{array}$$

I

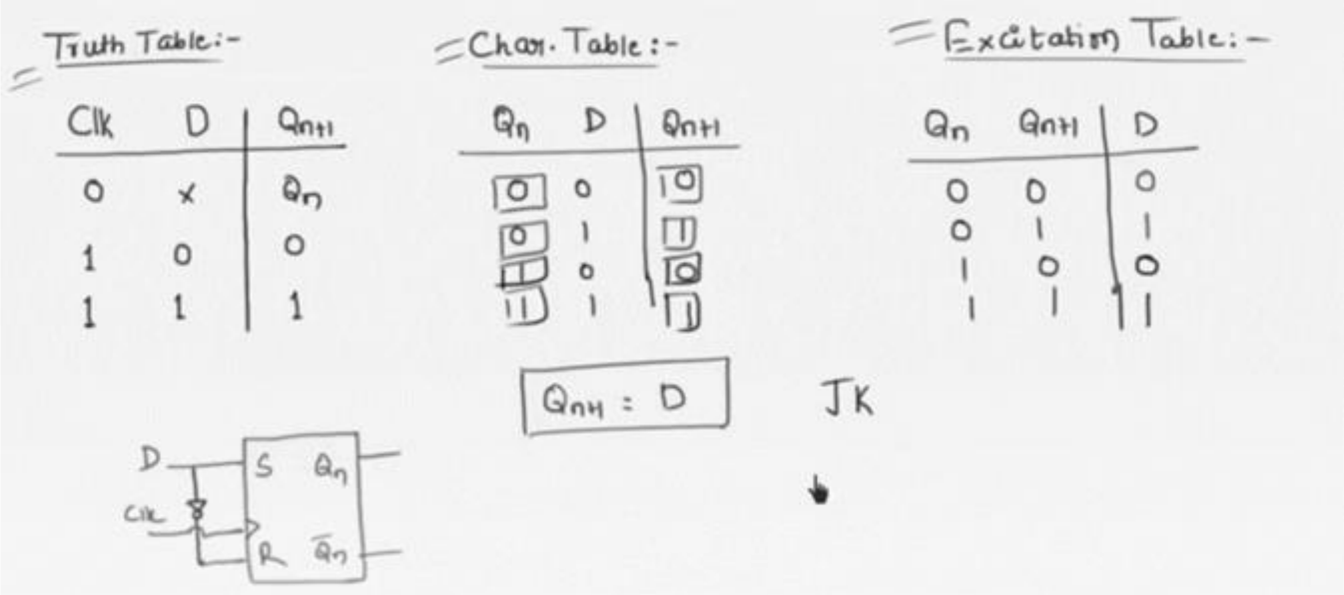
$Q_{n+1} = \text{I} + \text{II}$

$Q_{n+1} = S + Q_n\overline{R}$

D FF



Clk	D	Q_{n+1}
0	x	Q_n
1	0	0
1	1	1



Truth Table:-

Clk	D	Q_{n+1}
0	x	Q_n
1	0	0
1	1	1

Char. Table:-

Q_n	D	Q_{n+1}
0	0	0
0	1	1
1	0	0
1	1	1

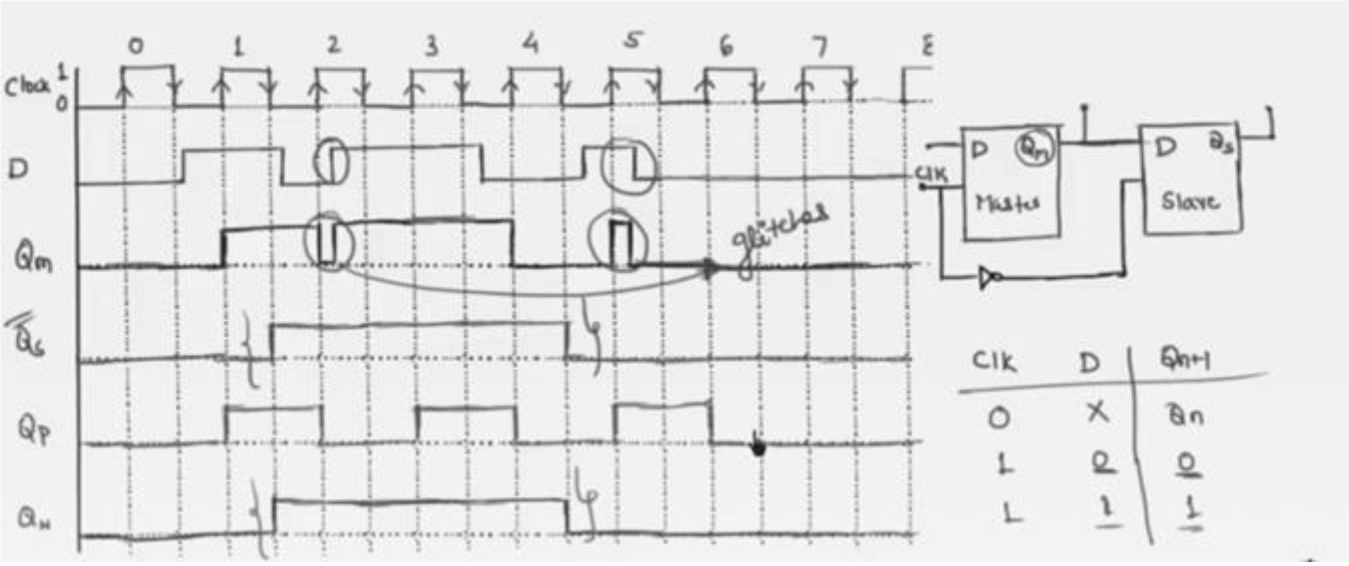
Excitation Table:-

Q_n	Q_{n+1}	D
0	0	0
0	1	1
1	0	0
1	1	1

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

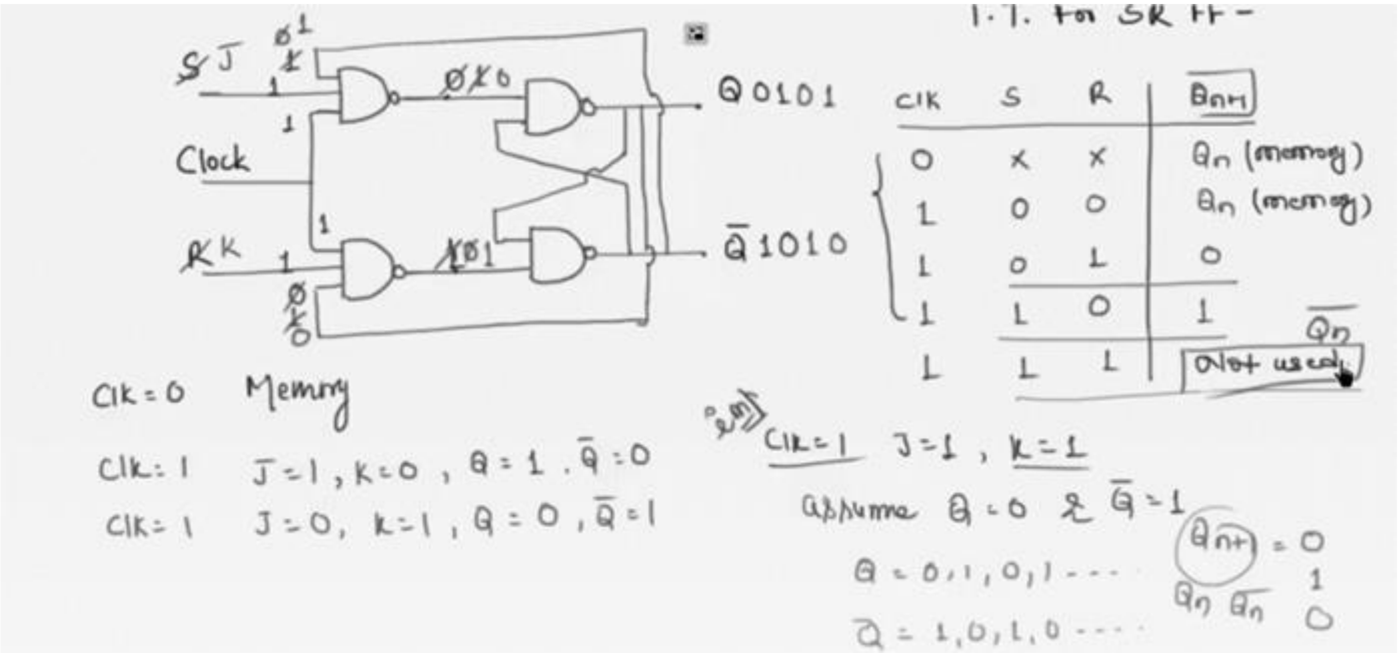
JK

BEHAVIOUR OF MASTER SLAVER D FF



Clk	D	Q_{n+1}
0	X	Q_n
1	0	0
1	1	1

JK FF



1.1. for SR FF –

Clk	S	R	Q_{n+1}
0	x	x	Q_n (memory)
1	0	0	Q_n (memory)
1	0	1	0
1	1	0	1
1	1	1	Not used $\overline{Q_n}$

Clk = 0 Memory

Clk = 1 J = 1 , K = 0 , Q = 1 . $\bar{Q}$ = 0

Clk = 1 J = 0, K = 1 , Q = 0 , $\bar{Q}$ = 1

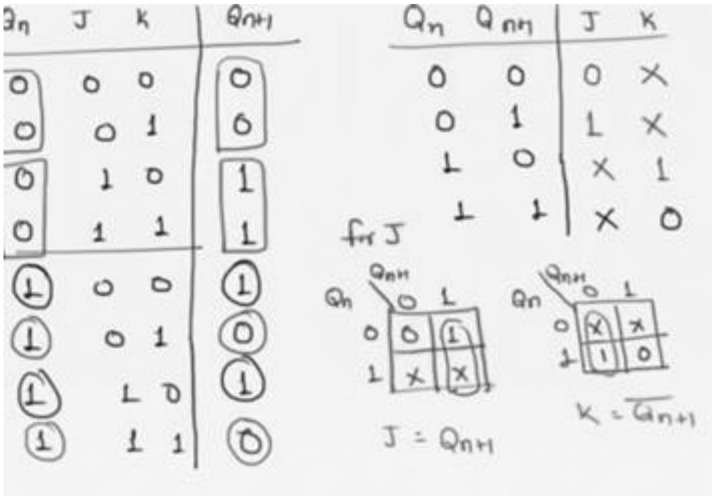
Clk = 1 J = 1 , K = 1

assume Q = 0 & $\bar{Q}$ = 1

Q = 0, 1, 0, 1 - - -

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

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



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

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

for J

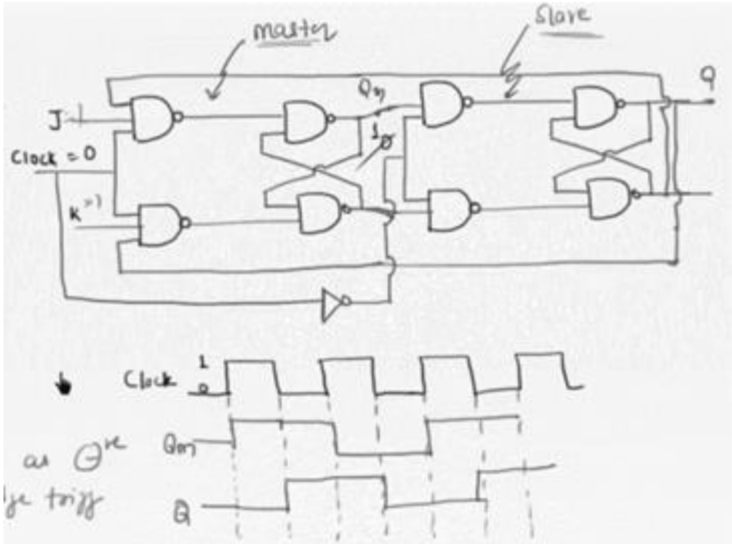
Q_n \ Q_{n+1}	0	1
0	0	1
1	x	x

J = Q_{n+1}

Q_n \ Q_{n+1}	0	1
0	x	x
1	1	0

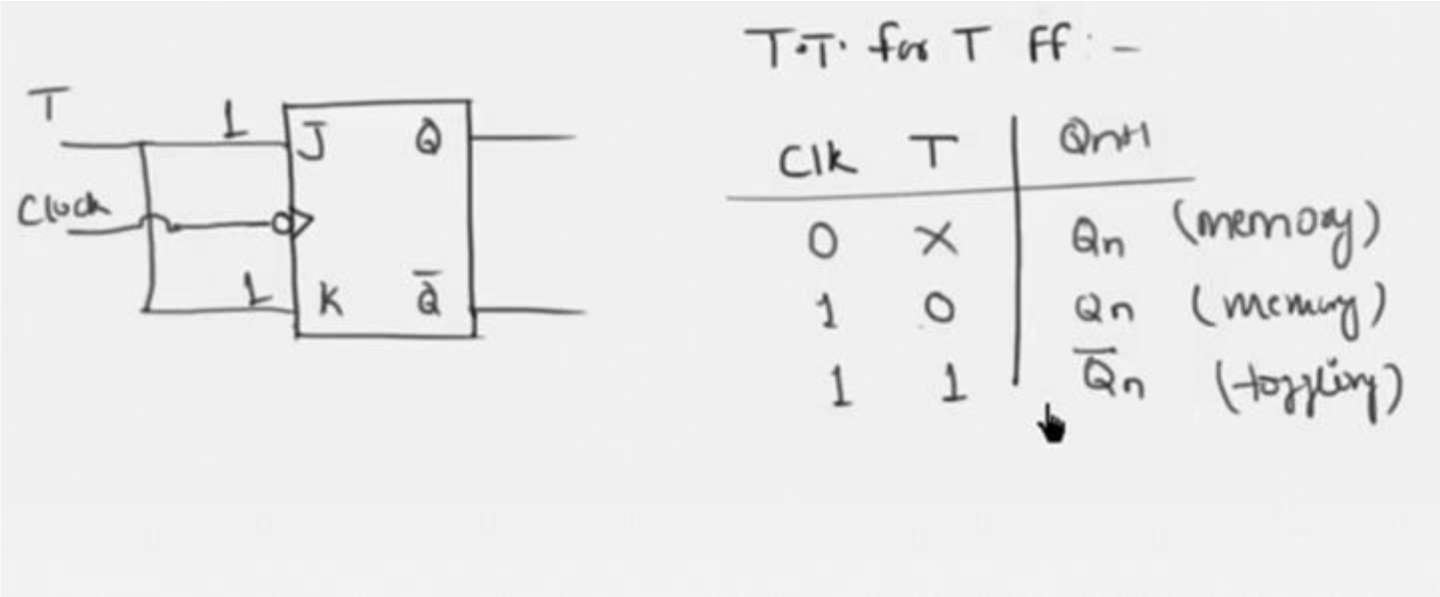
K = $\overline{Q_{n+1}}$

MASTER SLAVE JK FF



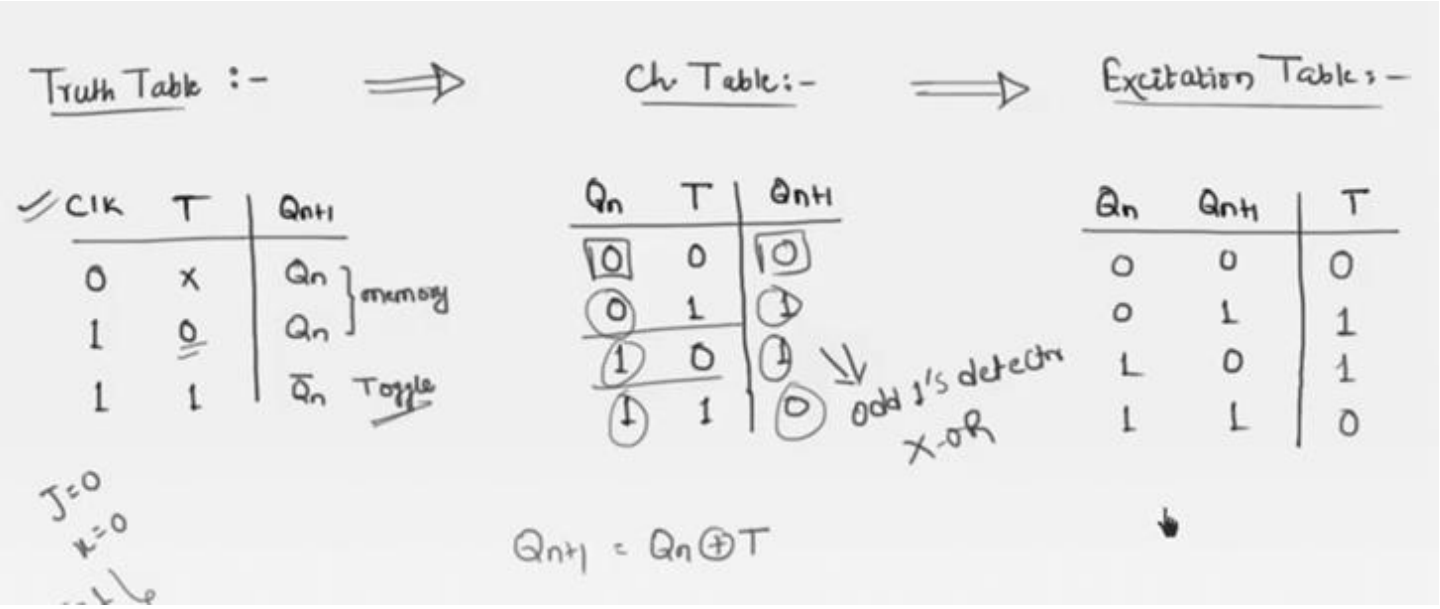
Master slave JK FF is the same as negative edge trigg

T FF



T·T· for T FF :-

Clk	T	Q_{n+1}
0	x	Q_n (memory)
1	0	Q_n (memory)
1	1	$\bar{Q}_n$ (toggling)



Truth Table :- ⟹ Ch. Table :- ⟹ Excitation Table :-

Clk	T	Q_{n+1}
0	x	Q_n } memory
1	0	Q_n
1	1	$\bar{Q}_n$ Toggle

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

odd 1's detector

X-OR

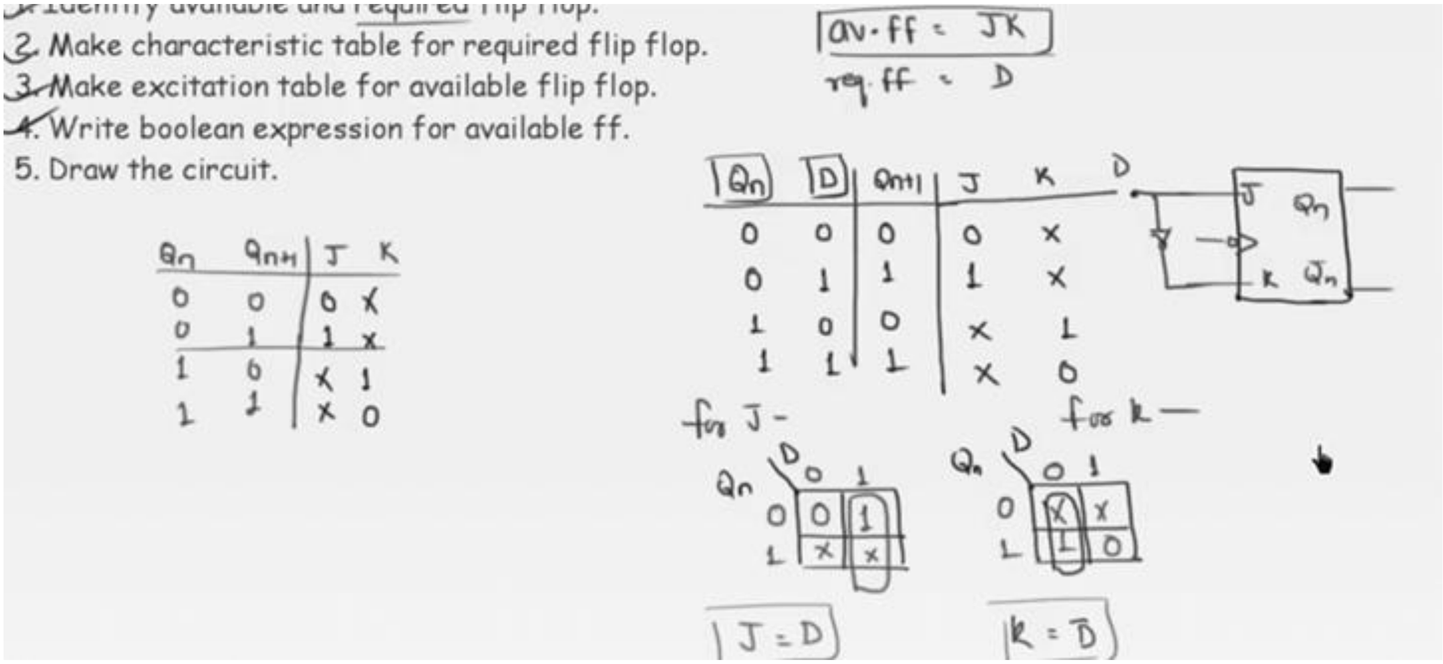
Q_n	Q_{n+1}	T
0	0	0
0	1	1
1	0	1
1	1	0

J=0
K=0

$Q_{n+1} = Q_n \oplus T$

FLIPFLOP CONVERSION

JK TO D



~~1.~~ Identify available and required flip flop.

~~2.~~ Make characteristic table for required flip flop.

~~3.~~ Make excitation table for available flip flop.

~~4.~~ Write boolean expression for available ff.

5. Draw the circuit.

av.ff = JK

req. ff = D

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

Q_n	D	Q_{n+1}	J	K
0	0	0	0	x
0	1	1	1	x
1	0	0	x	1
1	1	1	x	0

for J -

Q_n \ D	0	1
0	0	1
1	x	x

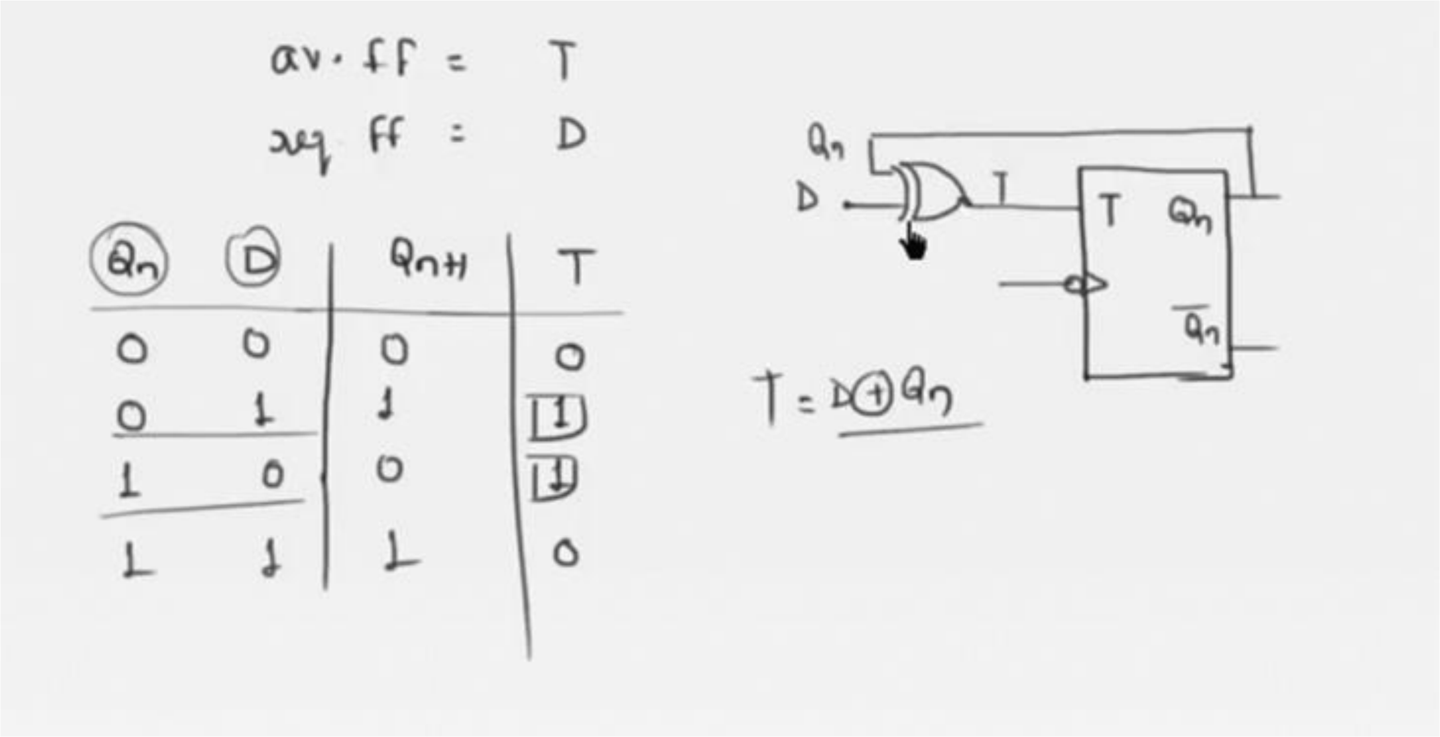
$J = D$

for K —

Q_n \ D	0	1
0	X	x
1	1	0

$K = \bar{D}$

T TO D



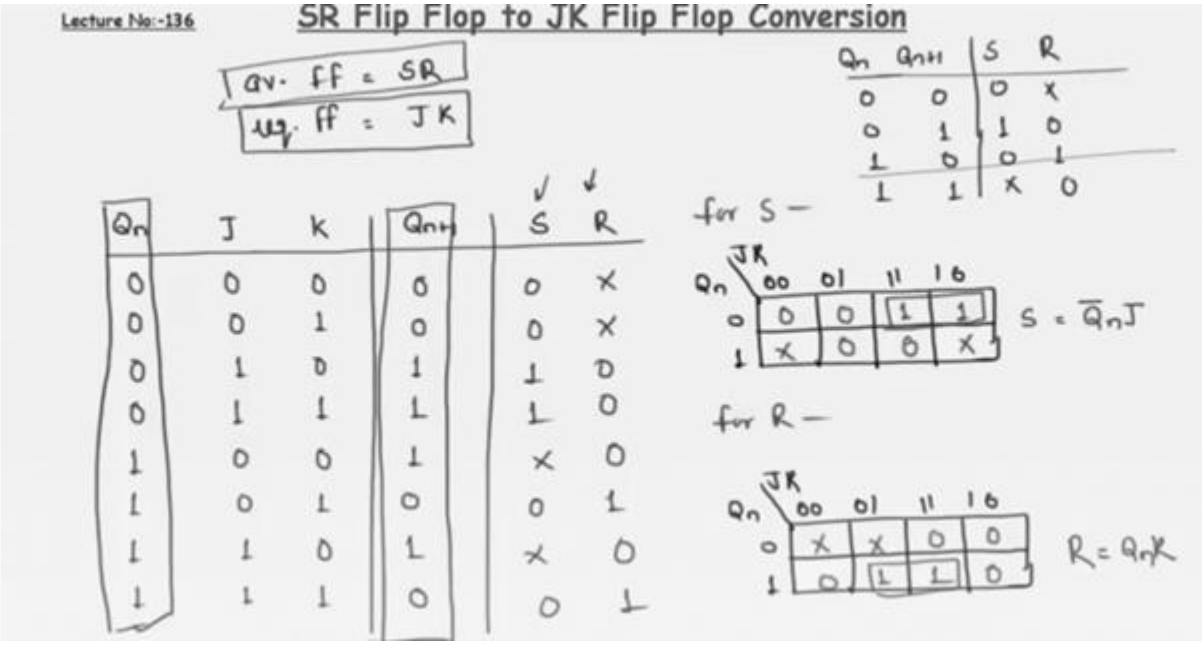
av.ff = T

req ff : D

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

$T = D \oplus Q_n$

SR TO JK



Lecture No:-136

SR Flip Flop to JK Flip Flop Conversion

av. ff = SR

rq. ff = JK

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

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

for S —

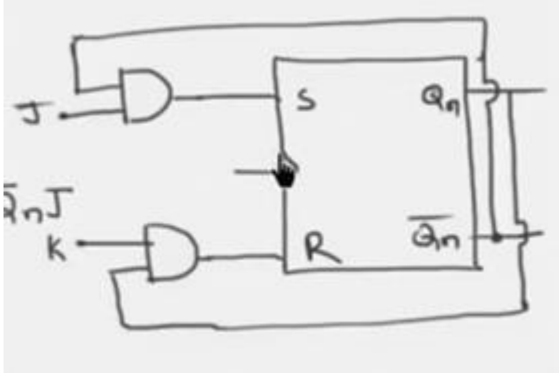
Q_n \ JK	00	01	11	10
0	0	0	1	1
1	X	0	0	X

$S = \overline{Q}_n J$

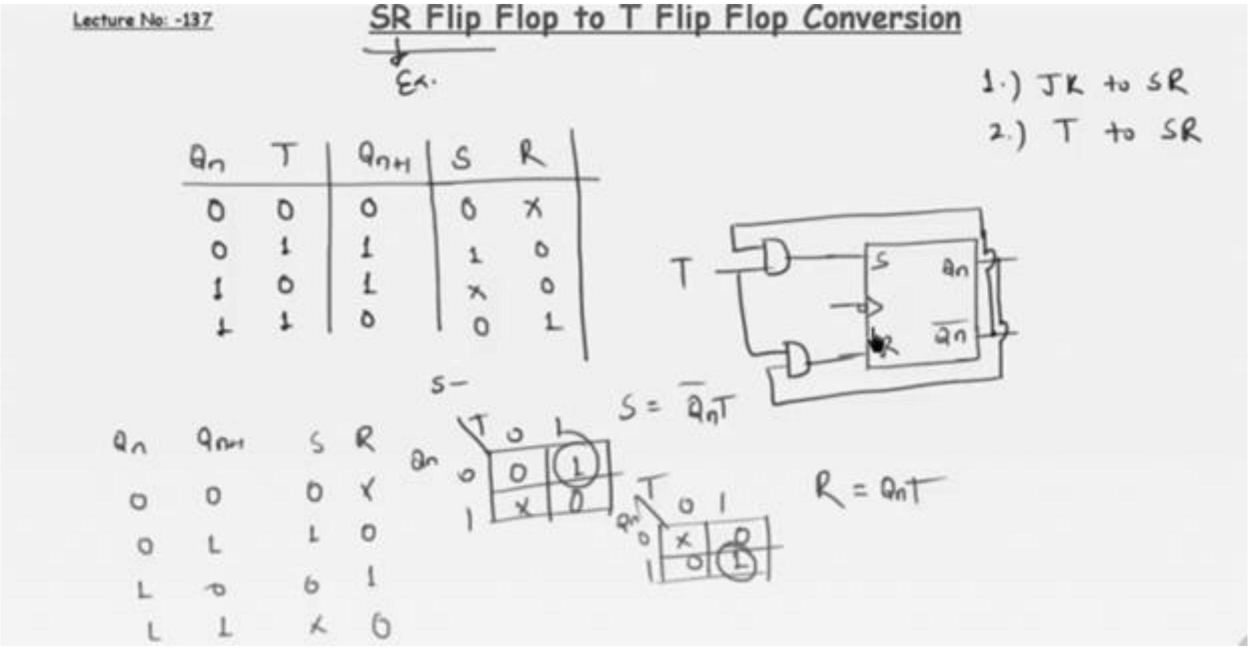
for R —

Q_n \ JK	00	01	11	10
0	X	X	0	0
1	0	1	1	0

$R = Q_n K$



SR TO T



Lecture No: -137

SR Flip Flop to T Flip Flop Conversion

Ex.

1.) JK to SR

2.) T to SR

Q_n	T	Q_{n+1}	S	R
0	0	0	0	X
0	1	1	1	0
1	0	1	X	0
1	1	0	0	1

S —

Q_n \ T	0	1
0	0	1
1	X	0

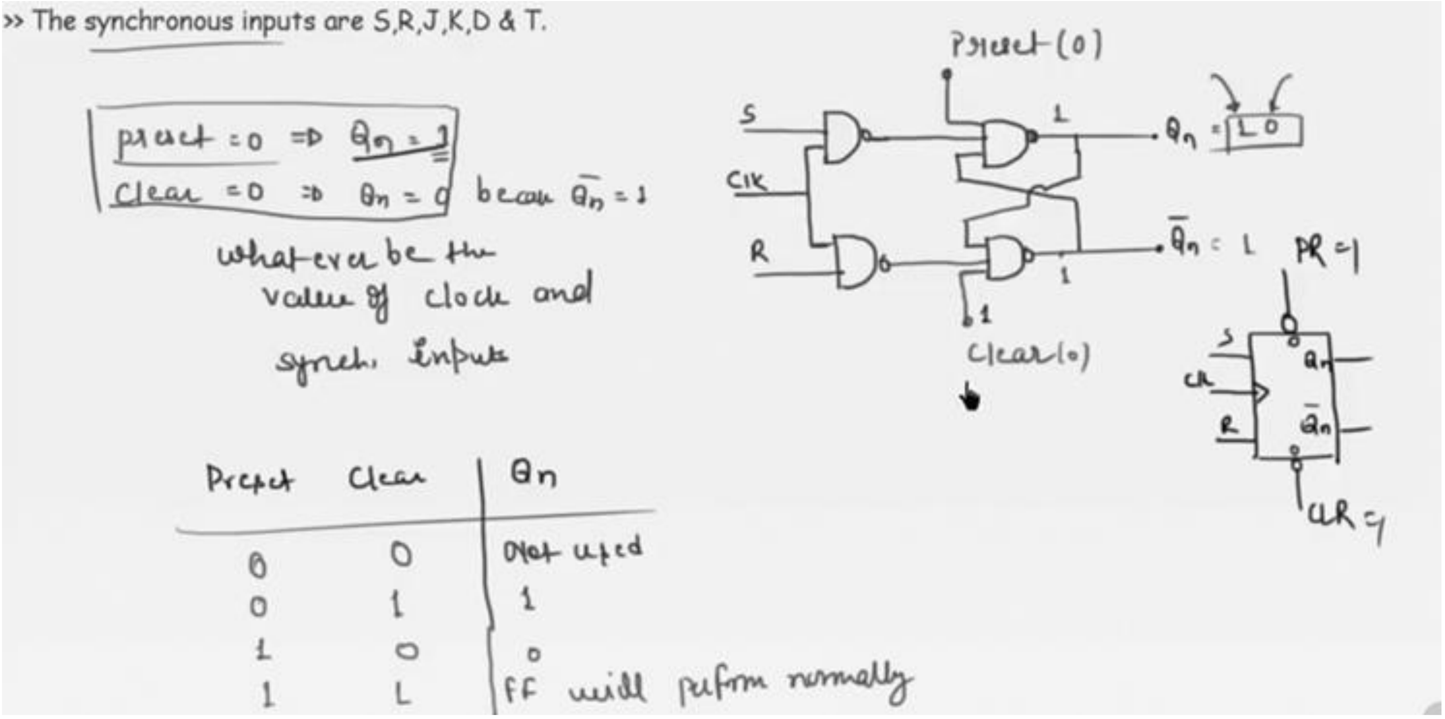
$S = \overline{Q}_n T$

Q_n \ T	0	1
0	X	0
1	0	1

$R = Q_n T$

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

SET AND CLR INPUT



>> The synchronous inputs are S,R,J,K,D & T.

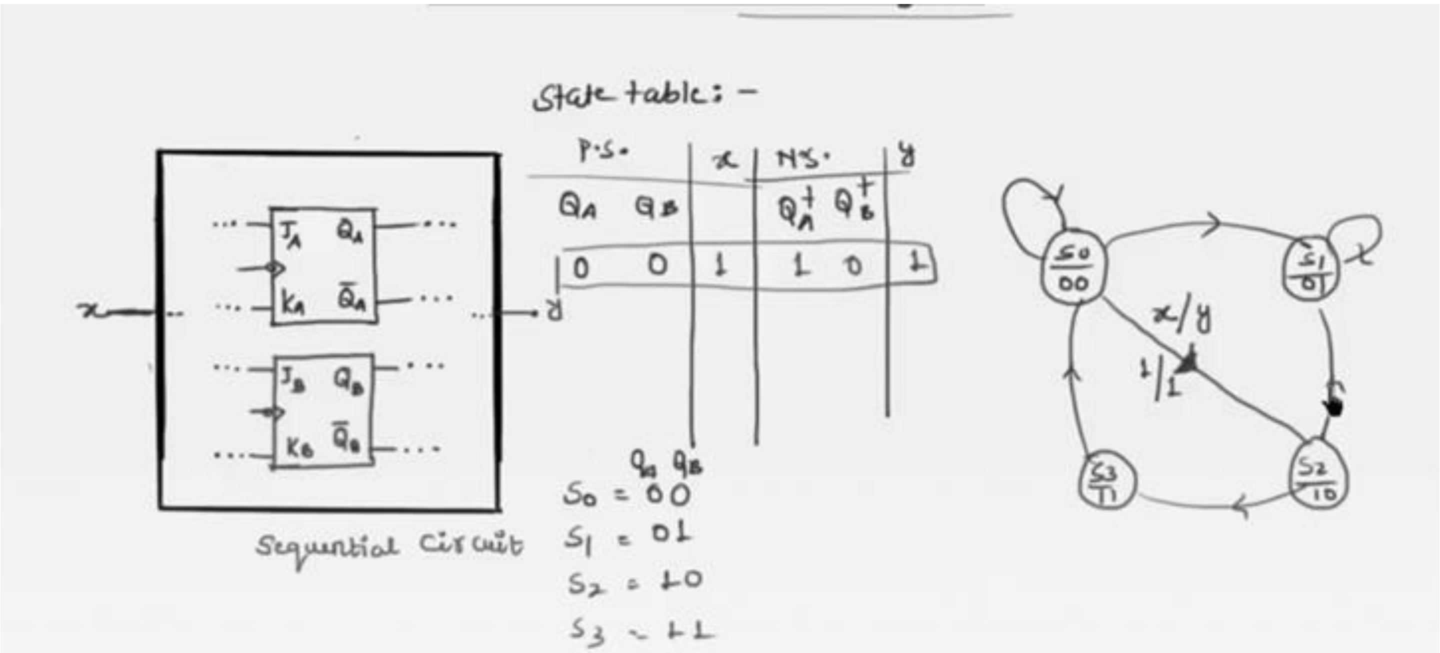
preset = 0 ⇒ $Q_n = 1$

Clear = 0 ⇒ $Q_n = 0$ becau $\bar{Q}_n = 1$

whatever be the value of clock and synch. inputs

Preset	Clear	Q_n
0	0	Not used
0	1	1
1	0	0
1	1	FF will pefrom normally

STATE DIAGRAM



State table: -

P.S.		x	N.S.		y
Q_A	Q_B		Q_A^+	Q_B^+	
0	0	1	1	0	1

Q_A Q_B

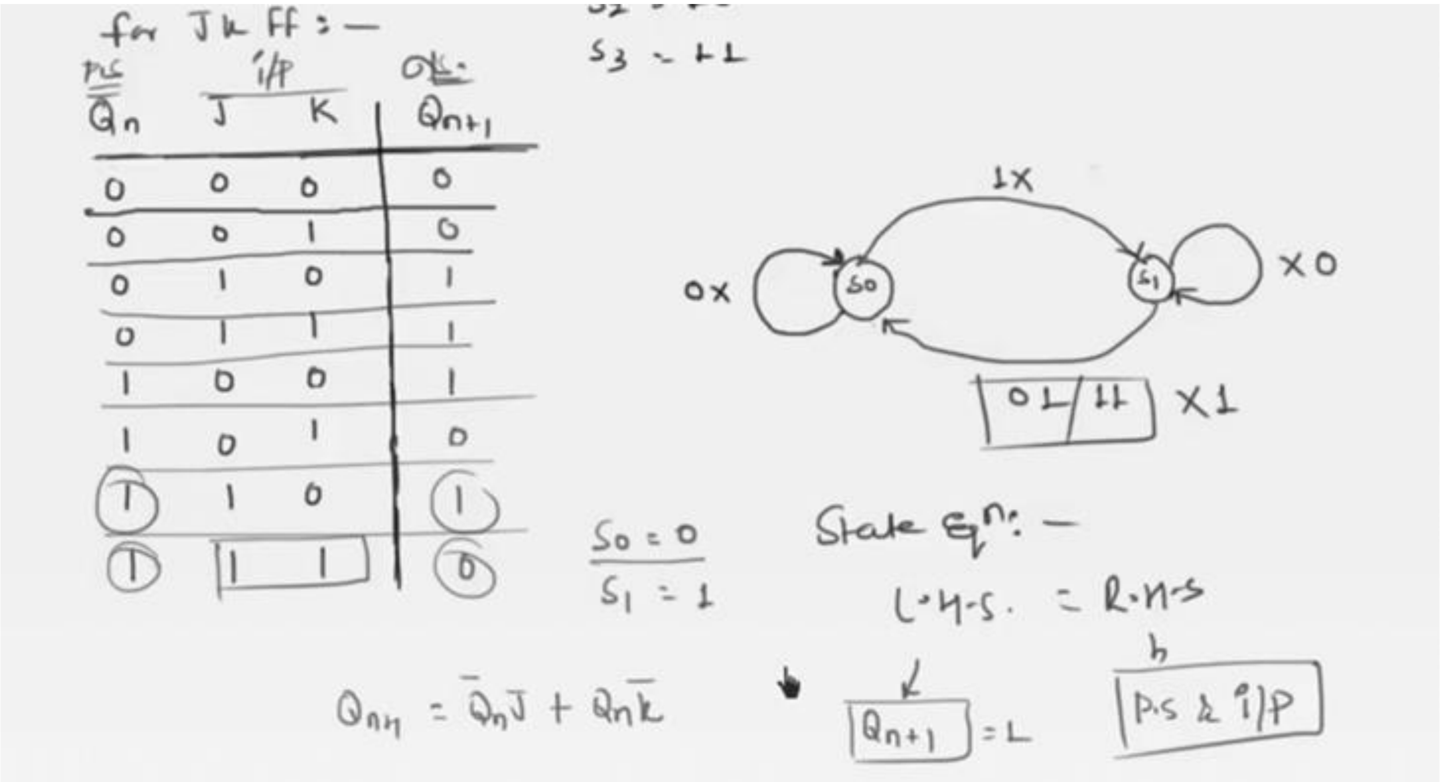
S_0 = 00

S_1 = 01

S_2 = 10

S_3 = 11

EX JK FF



S_3 = 11

for JK FF :-

p.s	i/p		o/p
$\overline{Q}_n$	J	K	Q_{n+1}
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	1
1	0	0	1
1	0	1	0
1	1	0	1
1	1	1	0

$S_0 = 0$

$S_1 = 1$

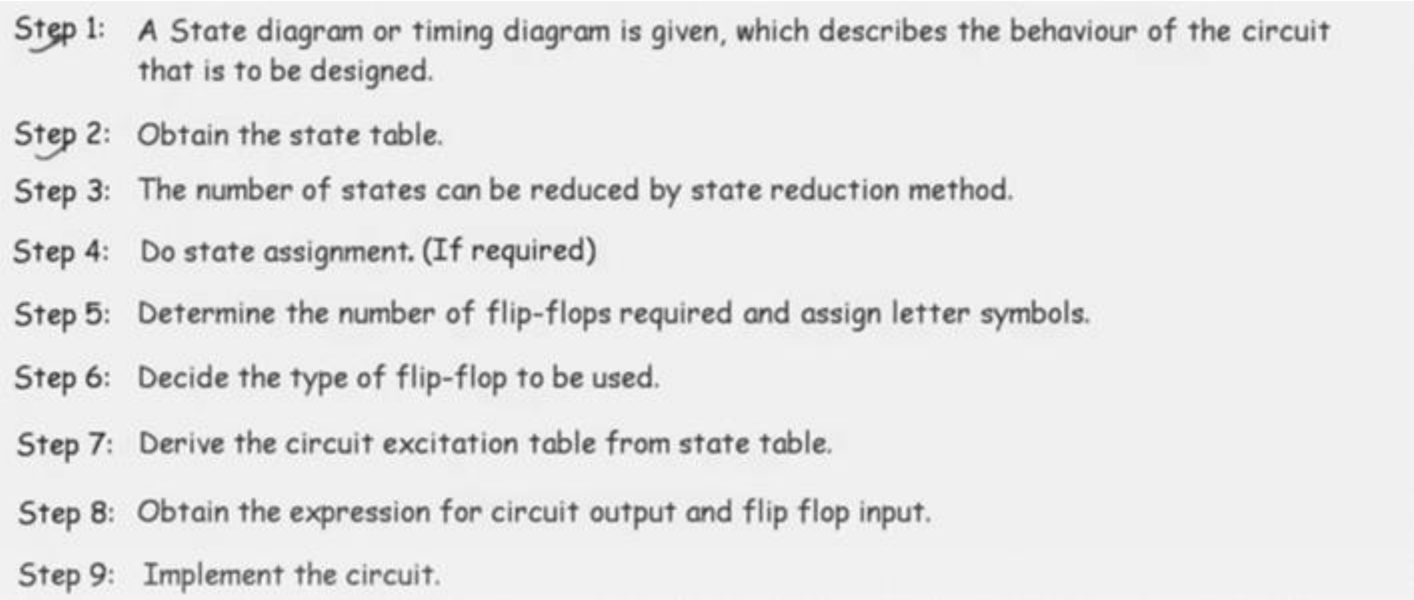
State eqn: -

L.H.S. = R.H.S

$\boxed{Q_{n+1}}$ = L $\boxed{\text{P.S \& i/p}}$

$$Q_{n+1} = \overline{Q}_n J + Q_n \overline{K}$$

Design Procedure for Clocked Sequential Circuits



Step 1: A State diagram or timing diagram is given, which describes the behaviour of the circuit that is to be designed.

Step 2: Obtain the state table.

Step 3: The number of states can be reduced by state reduction method.

Step 4: Do state assignment. (If required)

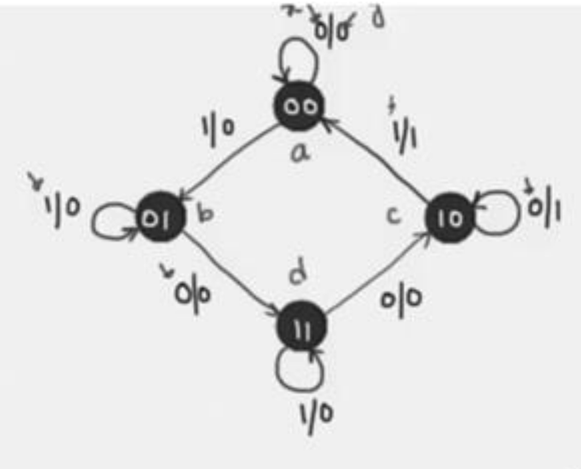
Step 5: Determine the number of flip-flops required and assign letter symbols.

Step 6: Decide the type of flip-flop to be used.

Step 7: Derive the circuit excitation table from state table.

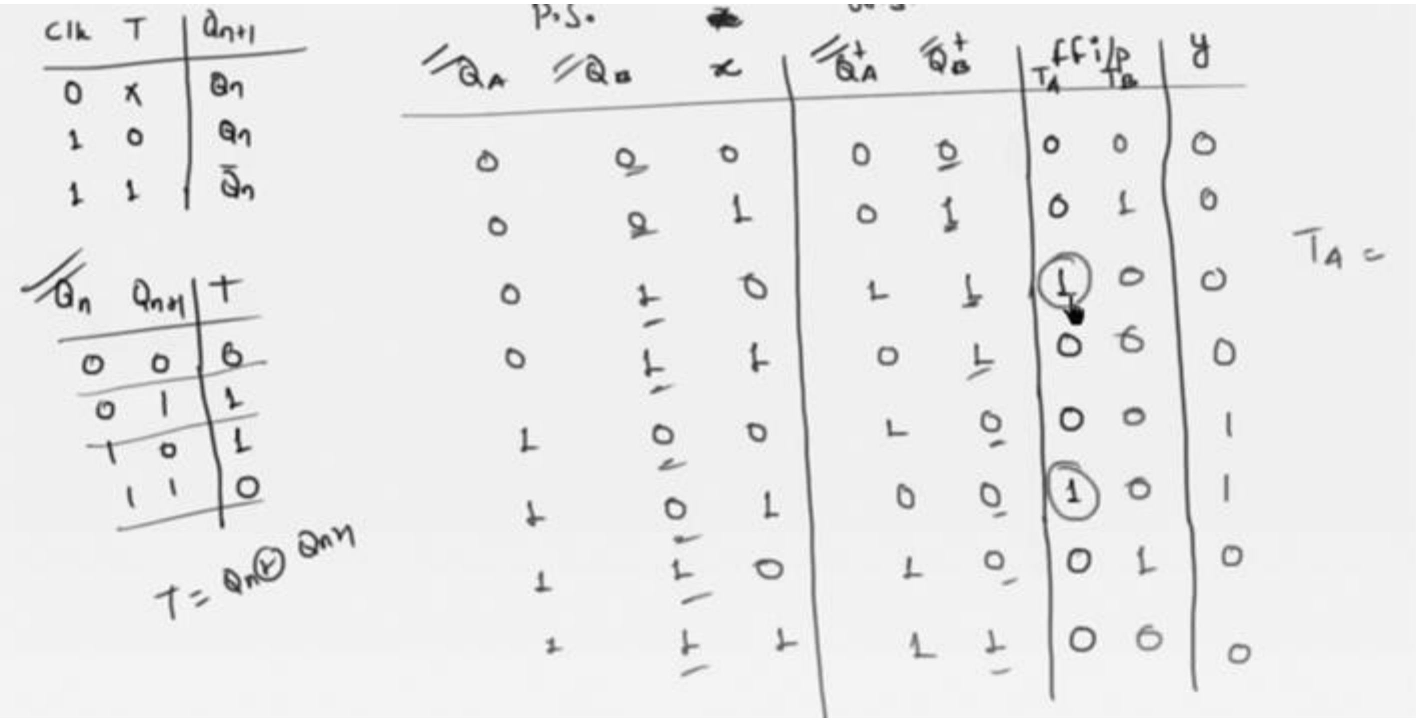
Step 8: Obtain the expression for circuit output and flip flop input.

Step 9: Implement the circuit.



State table : –

P.S.		N.S.				Output (y)	
		$x=0$		$x=1$		$x=0$	$x=1$
Q_A	Q_B	Q_A^+	Q_B^+	Q_A^+	Q_B^+		
0	0	0	0	0	1	0	0
0	1	1	1	0	1	0	0
1	0	1	0	0	0	1	1
1	1	1	0	1	1	0	0



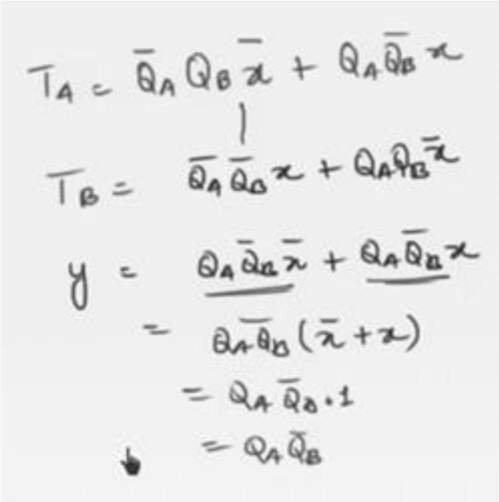
Clk	T	Q_{n+1}
0	x	Q_n
1	0	Q_n
1	1	$\bar{Q}_n$

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

$T = Q_n \oplus Q_{n+1}$

P.S.					ff i/p		
Q_A	Q_B	x	Q_A^+	Q_B^+	T_A	T_B	y
0	0	0	0	0	0	0	0
0	0	1	0	1	0	1	0
0	1	0	1	1	1	0	0
0	1	1	0	1	0	0	0
1	0	0	1	0	0	0	1
1	0	1	0	0	1	0	1
1	1	0	1	0	0	1	0
1	1	1	1	1	0	0	0

$T_A =$



$$T_A = \bar{Q}_A Q_B \bar{x} + Q_A \bar{Q}_B x$$

$$T_B = \bar{Q}_A \bar{Q}_B x + Q_A Q_B \bar{x}$$

$$y = \underline{Q_A \bar{Q}_B \bar{x}} + \underline{Q_A \bar{Q}_B x}$$

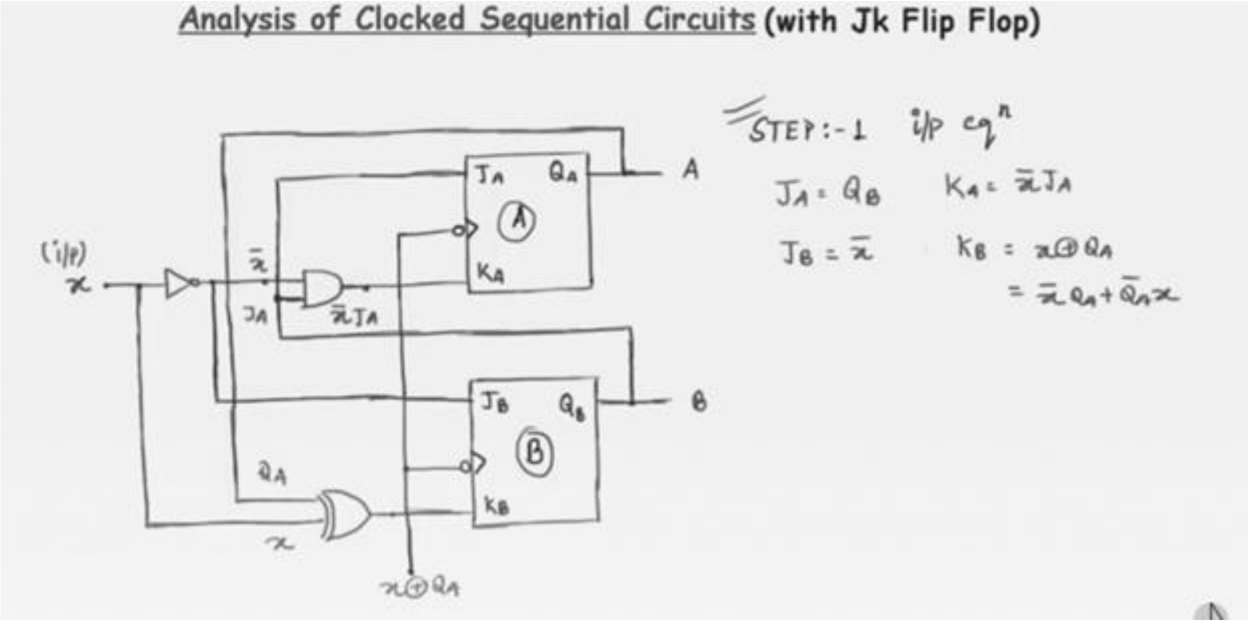
$$= Q_A \bar{Q}_B (\bar{x} + x)$$

$$= Q_A \bar{Q}_B \cdot 1$$

$$= Q_A \bar{Q}_B$$

Analysis of Clocked Sequential Circuits

JK FF



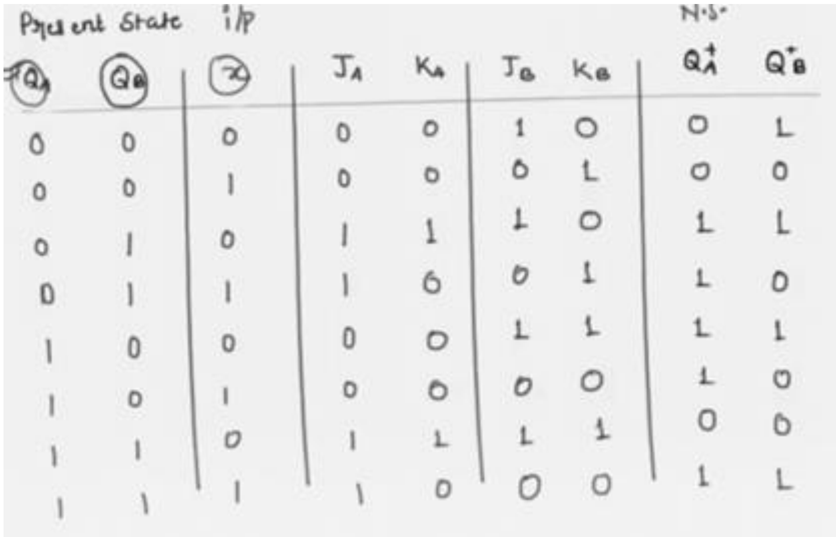
Analysis of Clocked Sequential Circuits (with Jk Flip Flop)

STEP:-1 i/p eq^n

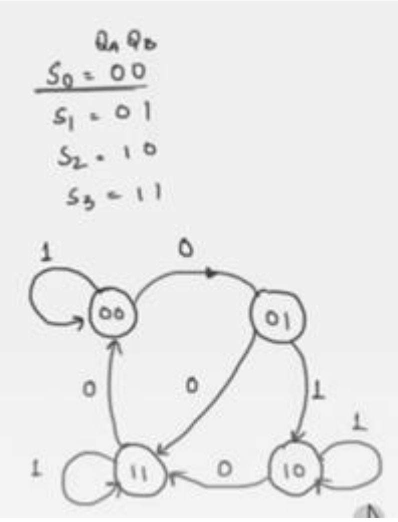
$J_A = Q_B \qquad K_A = \bar{x} J_A$

$J_B = \bar{x} \qquad K_B = x \oplus Q_A$

$= \bar{x} Q_A + \bar{Q}_A x$



Present State		i/p					N.S.	
Q_A	Q_B	x	J_A	K_A	J_B	K_B	Q_A^+	Q_B^+
0	0	0	0	0	1	0	0	1
0	0	1	0	0	0	1	0	0
0	1	0	1	1	1	0	1	1
0	1	1	1	0	0	1	1	0
1	0	0	0	0	1	1	1	1
1	0	1	0	0	0	0	1	0
1	1	0	1	1	1	1	0	0
1	1	1	1	0	0	0	1	1



$Q_A Q_B$

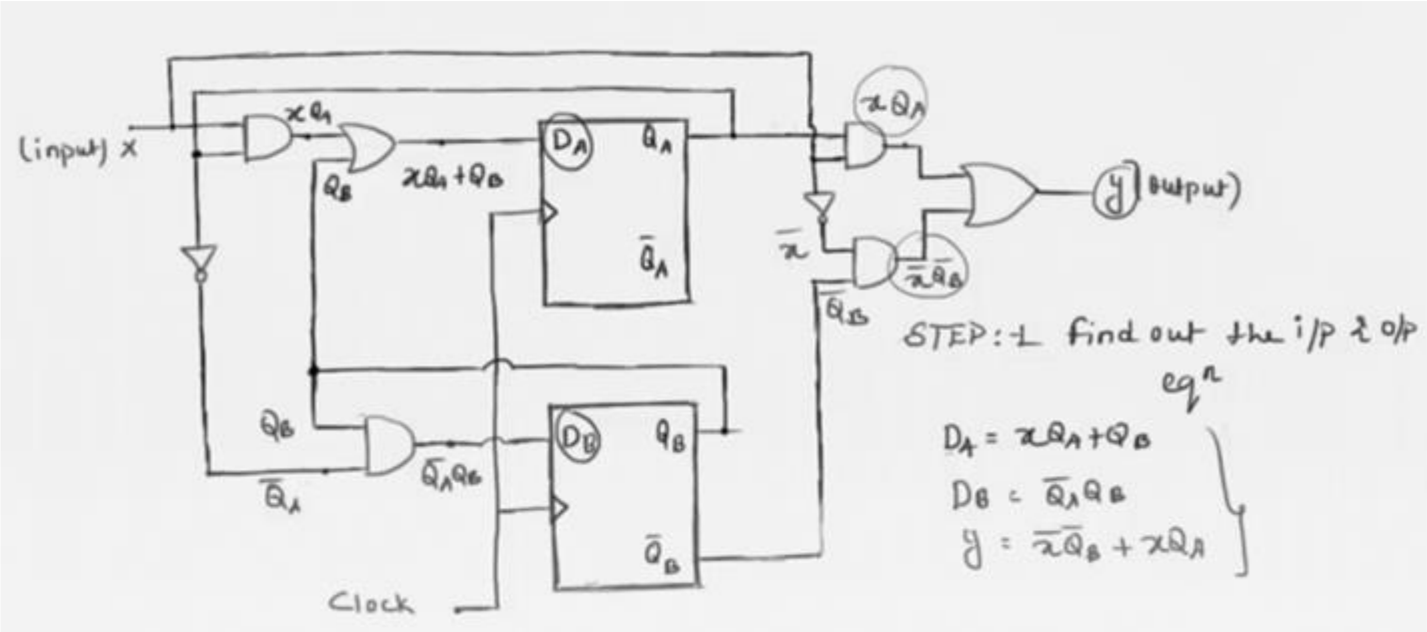
$S_0 = 00$

$S_1 = 01$

$S_2 = 10$

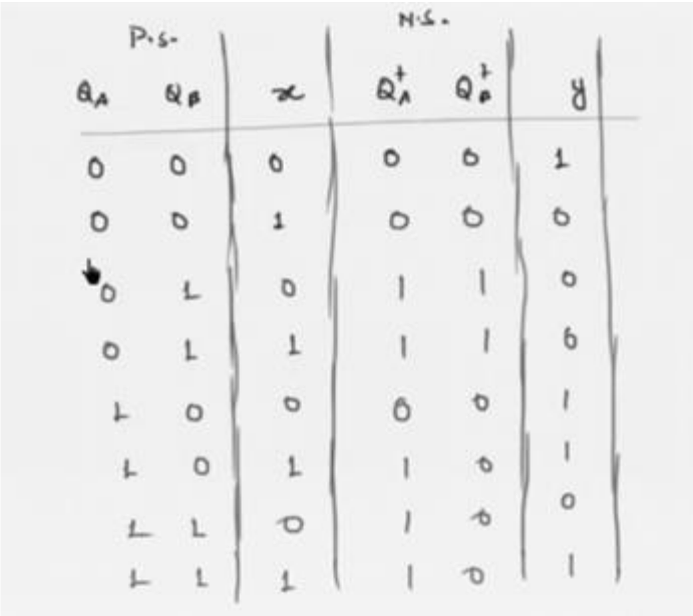
$S_3 = 11$

D FF



STEP:1 find out the i/p & o/p eq^n

$$\left.\begin{aligned} D_A &= xQ_A+Q_B \\ D_B &= \bar{Q}_AQ_B \\ y &= \bar{x}\bar{Q}_B+xQ_A \end{aligned}\right\}$$

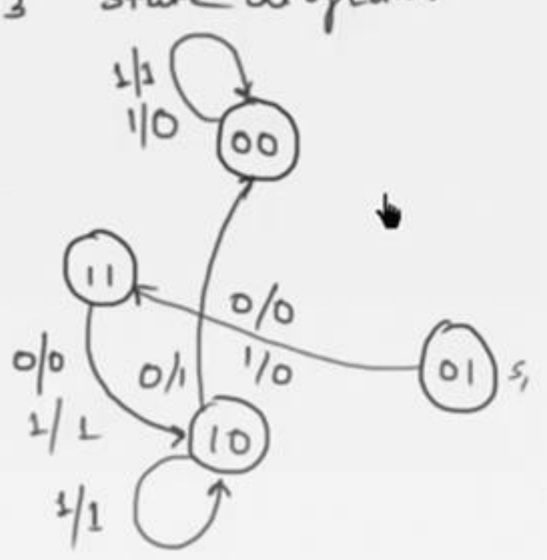


P.S.			N.S.		
Q_A	Q_B	x	Q_A^+	Q_B^+	y
0	0	0	0	0	1
0	0	1	0	0	0
0	1	0	1	1	0
0	1	1	1	1	0
1	0	0	0	0	1
1	0	1	1	0	1
1	1	0	1	0	0
1	1	1	1	0	1

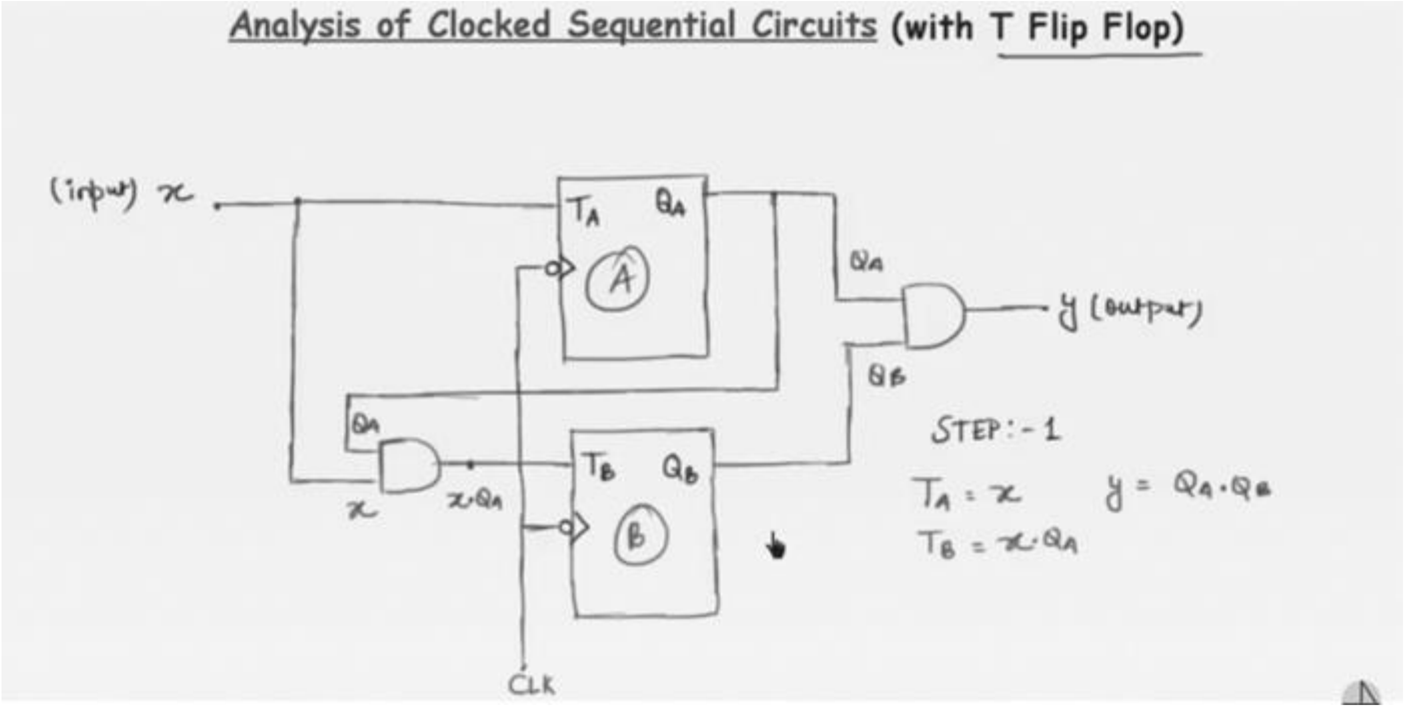
10

.11

STEP:-3 State diagram



T FF

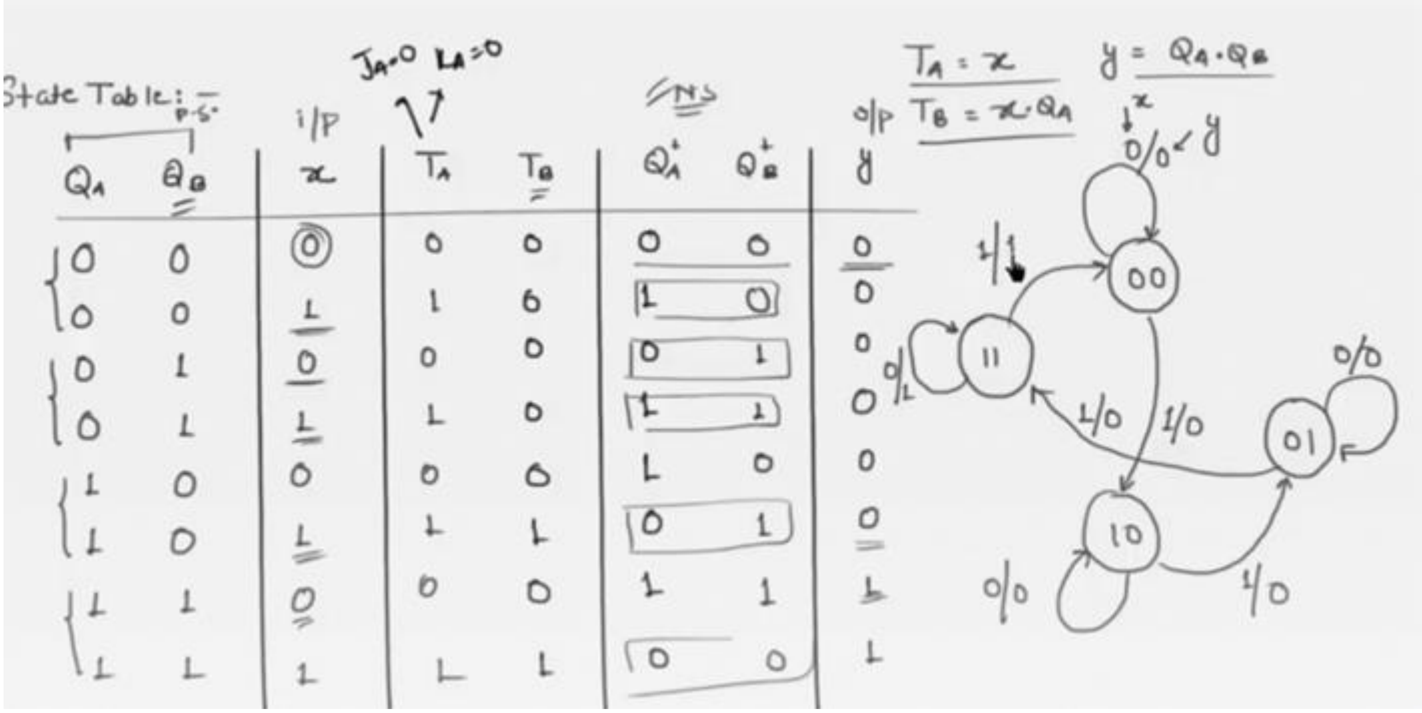


Analysis of Clocked Sequential Circuits (with T Flip Flop)

STEP :- 1

$T_A = x \qquad y = Q_A \cdot Q_B$

$T_B = x \cdot Q_A$



State Table :-

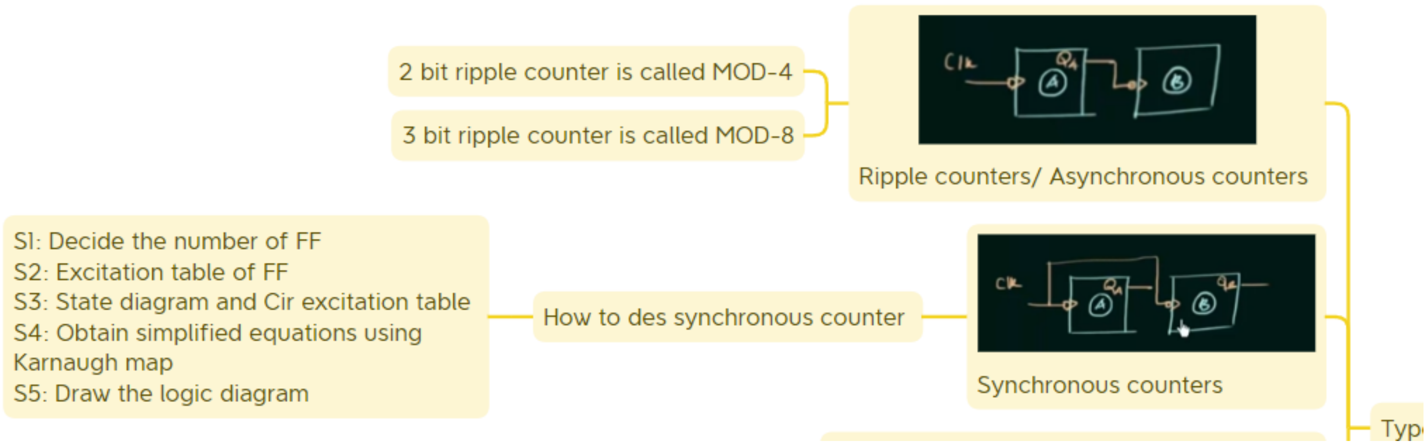
$J_A = 0 \quad K_A = 0$

P.S.		i/P			NS		o/P
Q_A	Q_B	x	T_A	T_B	Q_A^+	Q_B^+	y
0	0	0	0	0	0	0	0
0	0	1	1	0	1	0	0
0	1	0	0	0	0	1	0
0	1	1	1	0	1	1	0
1	0	0	0	0	1	0	0
1	0	1	1	1	0	1	0
1	1	0	0	0	1	1	1
1	1	1	1	1	0	0	1

$T_A = x \qquad y = Q_A \cdot Q_B$

$T_B = x \cdot Q_A$

COUNTER



2 bit ripple counter is called MOD-4

3 bit ripple counter is called MOD-8

Ripple counters/ Asynchronous counters

Sl: Decide the number of FF
S2: Excitation table of FF
S3: State diagram and Cir excitation table
S4: Obtain simplified equations using Karnaugh map
S5: Draw the logic diagram

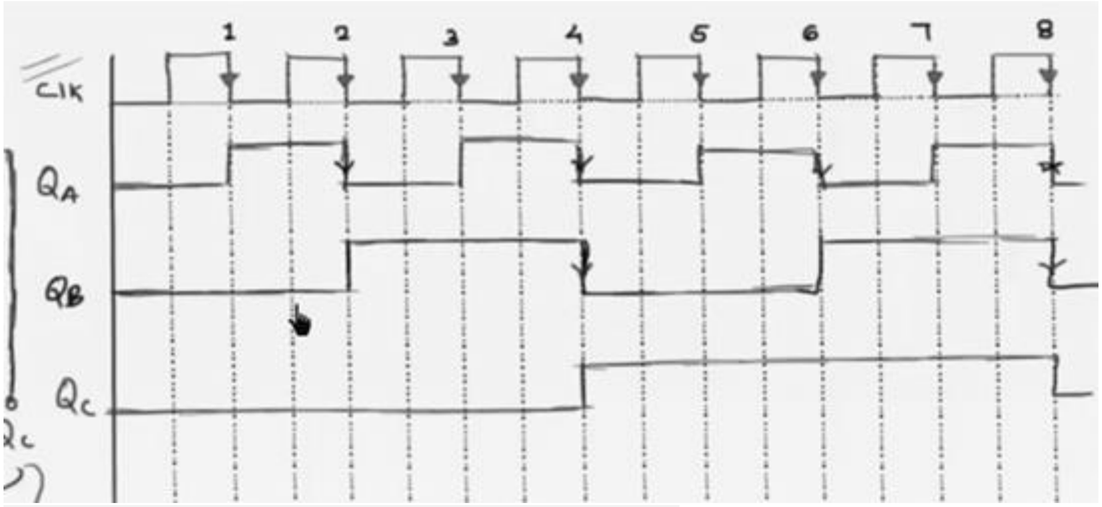
How to des synchronous counter

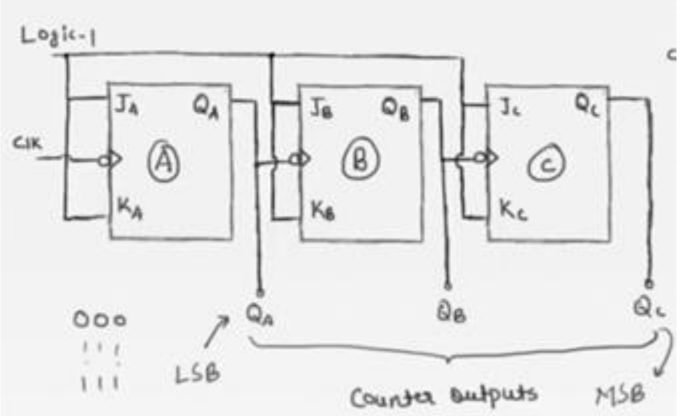
Synchronous counters

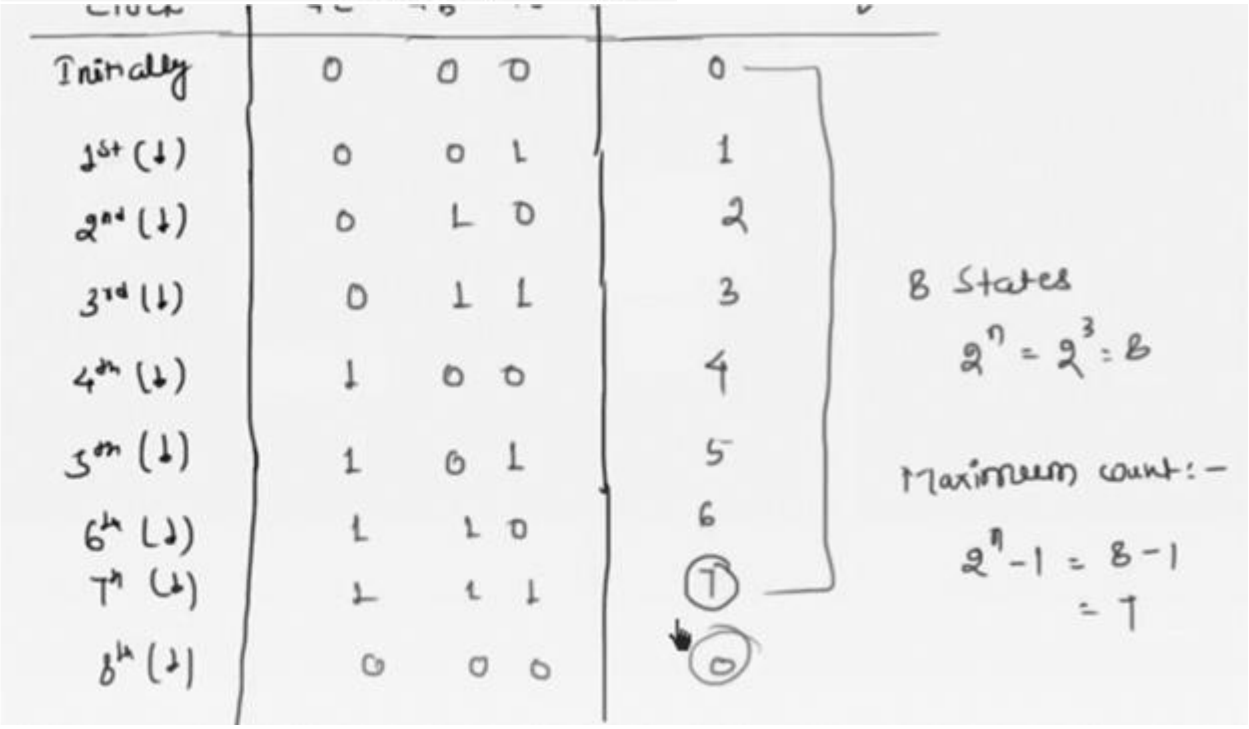
Typ

UP COUNTER

3 BIT SYNCHRONOUS/RIPPLE COUNTER







Initially	0	0	0	0
1st (↓)	0	0	1	1
2nd (↓)	0	1	0	2
3rd (↓)	0	1	1	3
4th (↓)	1	0	0	4
5th (↓)	1	0	1	5
6th (↓)	1	1	0	6
7th (↓)	1	1	1	7
8th (↓)	0	0	0	0

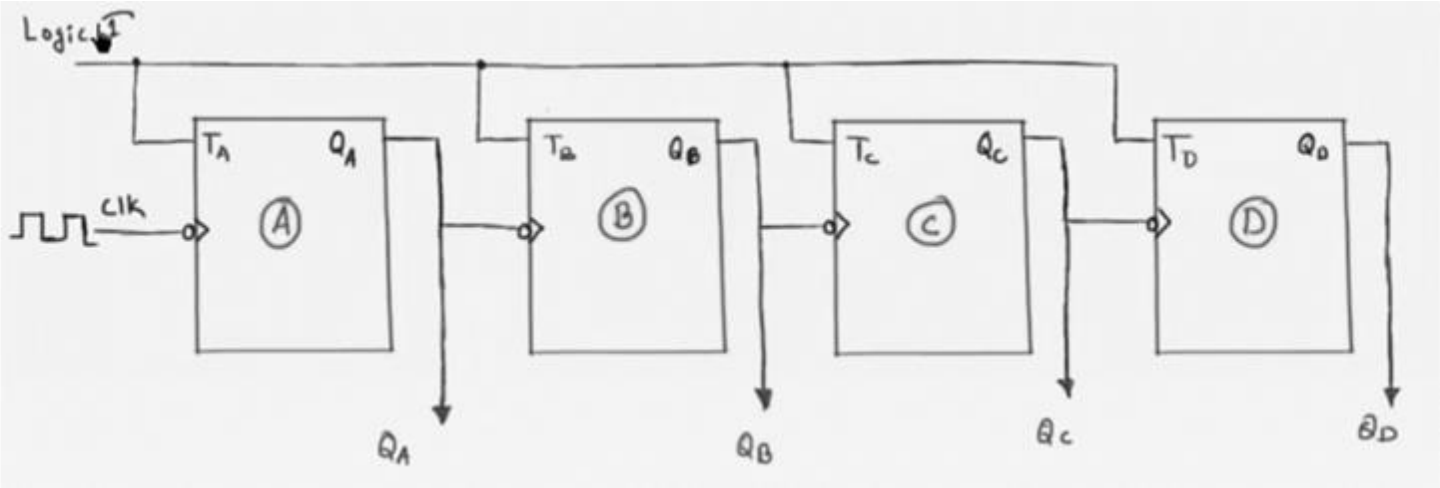
8 States

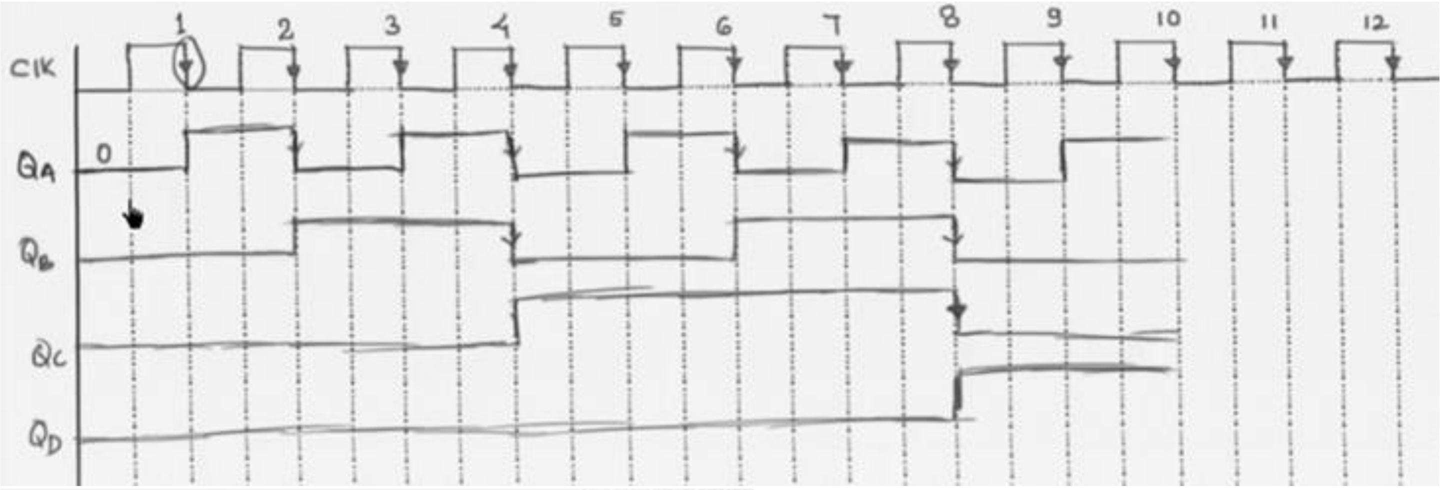
$2^n = 2^3 = 8$

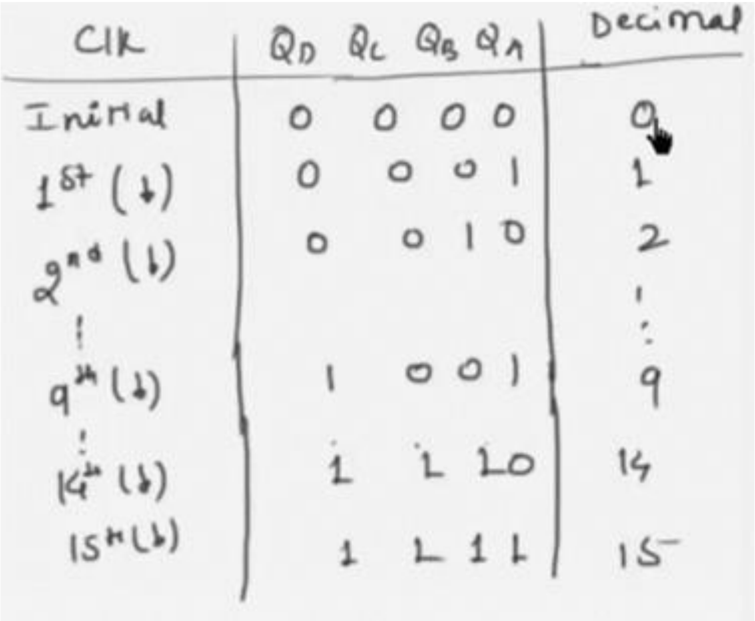
Maximum count:-

$2^n - 1 = 8 - 1$
$= 7$

4 BIT RIPPLE COUNTER

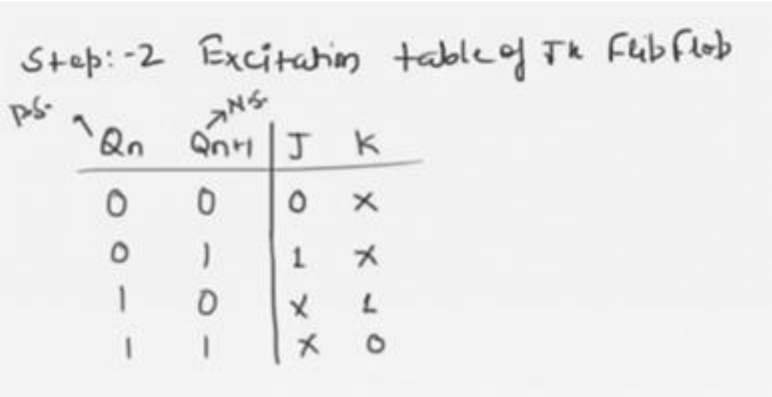






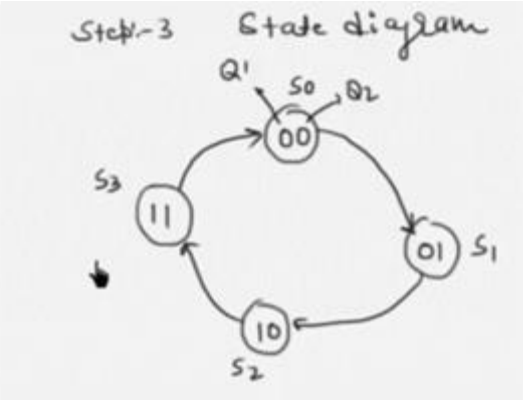
Clk	Q_D	Q_C	Q_B	Q_A	Decimal
Initial	0	0	0	0	0
1st (↓)	0	0	0	1	1
2nd (↓)	0	0	1	0	2
⋮					⋮
9th (↓)	1	0	0	1	9
⋮					
14th (↓)	1	1	1	0	14
15th (↓)	1	1	1	1	15

2 BIT SYNCHRONOUS COUNTER

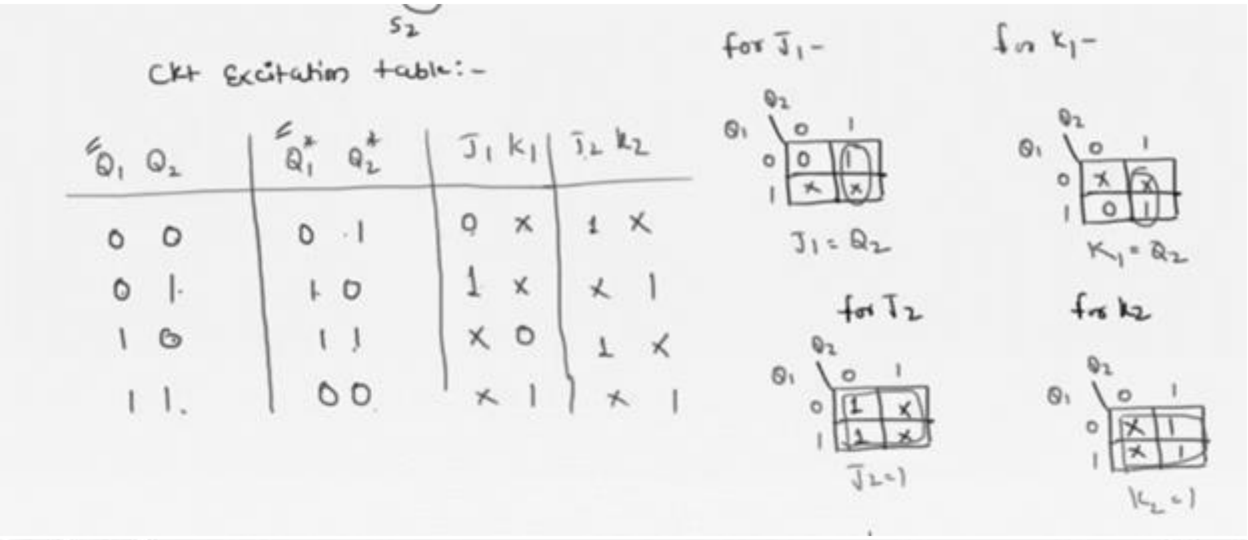


Step:-2 Excitation table of JK FlipFlop

PS Q_n	NS Q_{n+1}	J	K
0	0	0	X
0	1	1	X
1	0	X	1
1	1	X	0



Step:-3 State diagram



Ckt 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	0	1	1	X	0	1	X
1	1	0	0	X	1	X	1

for J_1-

$Q_1 \backslash Q_2$	0	1
0	0	1
1	X	X

$J_1 = Q_2$

for K_1-

$Q_1 \backslash Q_2$	0	1
0	X	X
1	0	1

$K_1 = Q_2$

for J_2

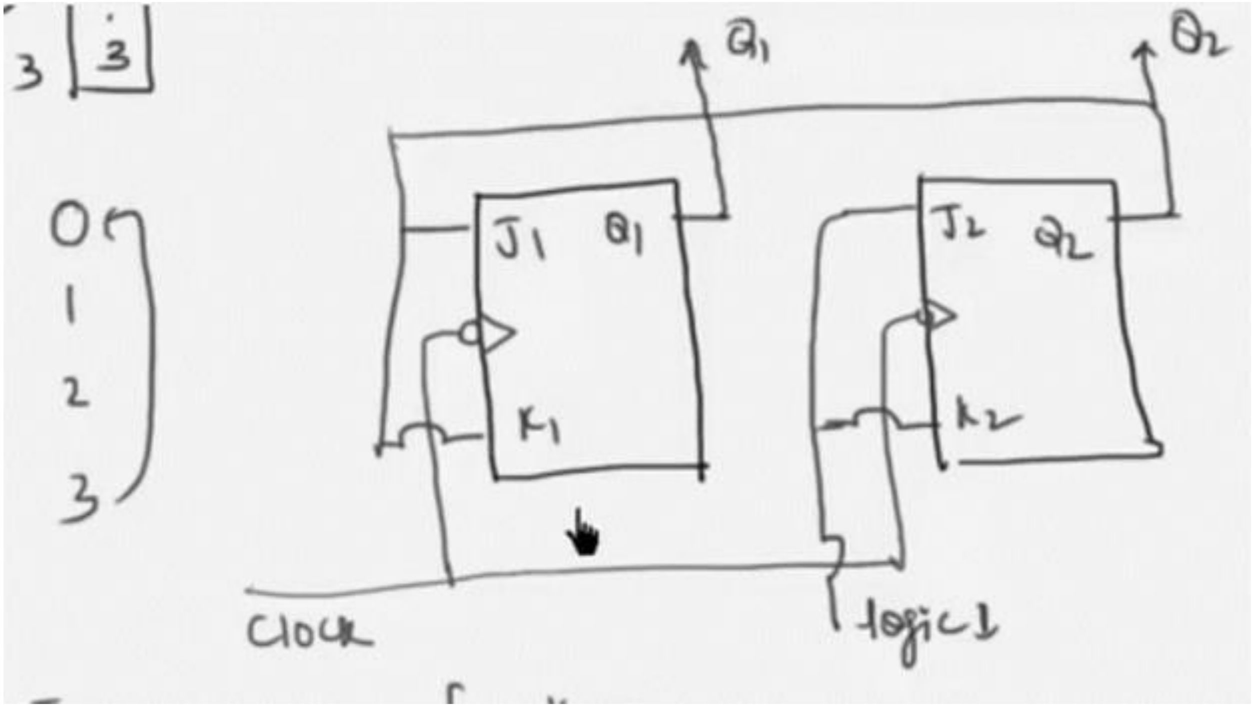
$Q_1 \backslash Q_2$	0	1
0	1	X
1	1	X

$J_2 = 1$

for K_2

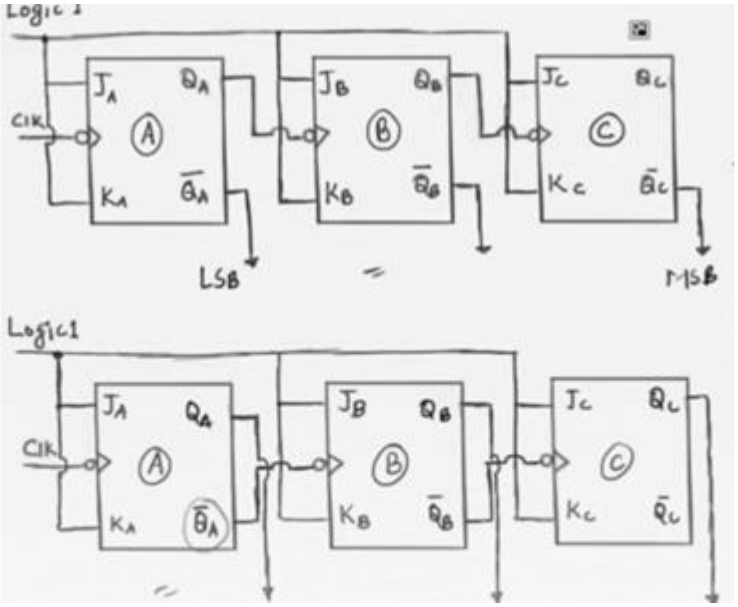
$Q_1 \backslash Q_2$	0	1
0	X	1
1	X	1

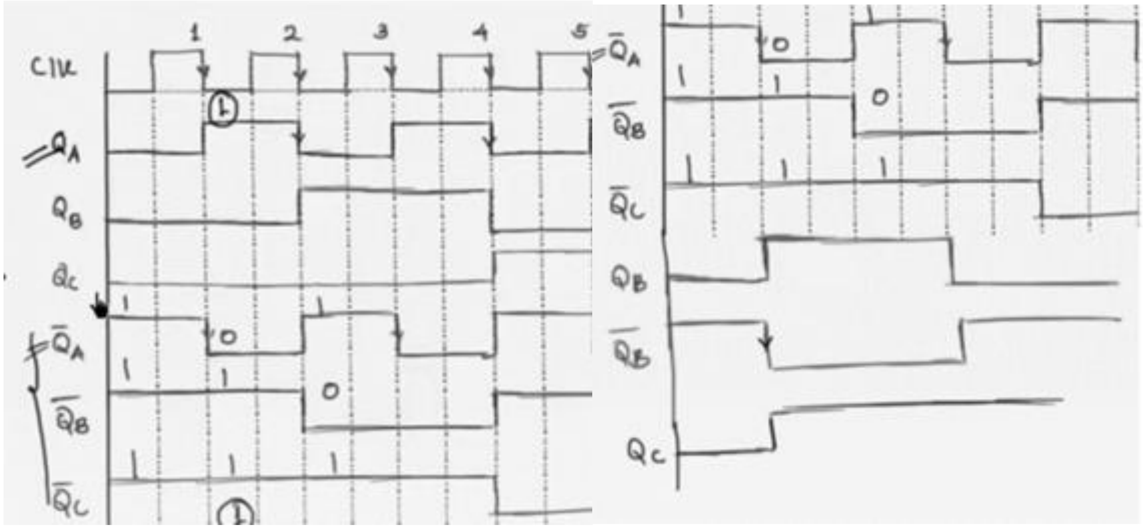
$K_2 = 1$

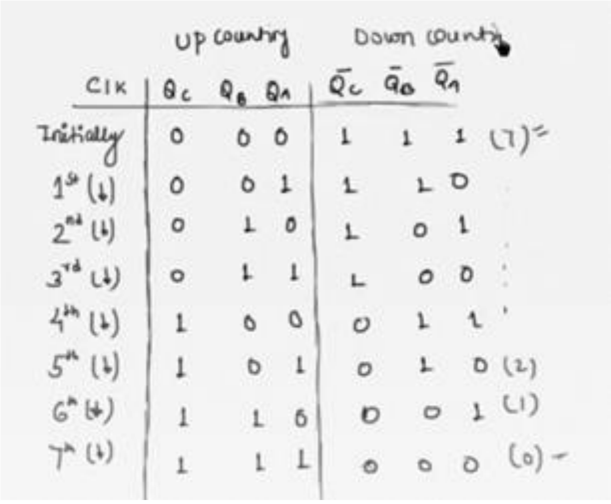


DOWN COUNTER

3 BIT RIPPLE DOWN COUNTER



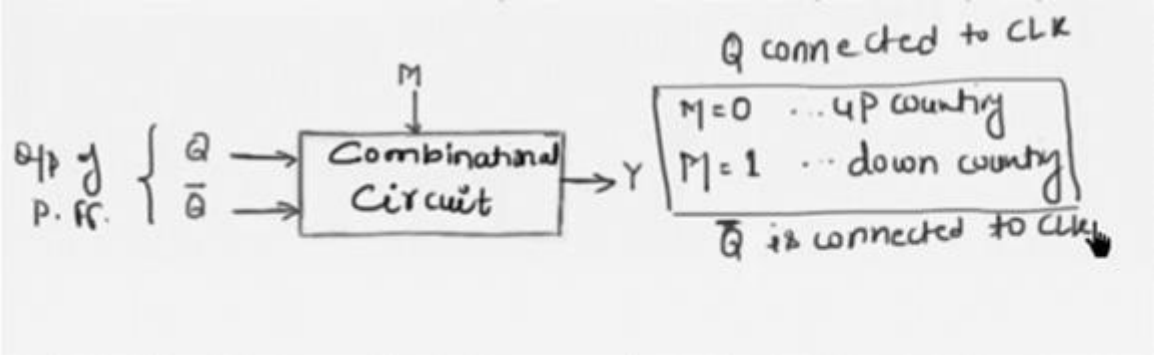


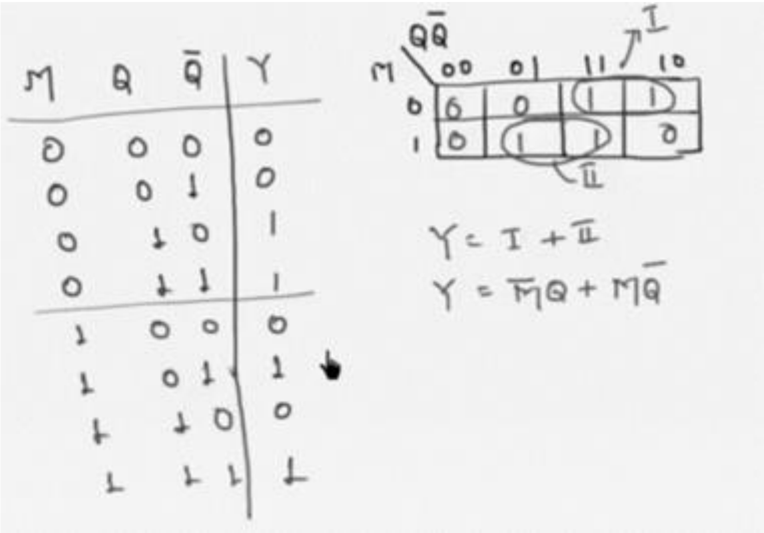


	Up counting			Down counting			
CLK	Q_C	Q_B	Q_A	$\bar{Q}_C$	$\bar{Q}_B$	$\bar{Q}_A$	
Initially	0	0	0	1	1	1	(7)
1st (↓)	0	0	1	1	1	0	
2nd (↓)	0	1	0	1	0	1	
3rd (↓)	0	1	1	1	0	0	
4th (↓)	1	0	0	0	1	1	
5th (↓)	1	0	1	0	1	0	(2)
6th (↓)	1	1	0	0	0	1	(1)
7th (↓)	1	1	1	0	0	0	(0)

UP/DOWN COUNTER

3 BIT RIPPLE UD COUNTER



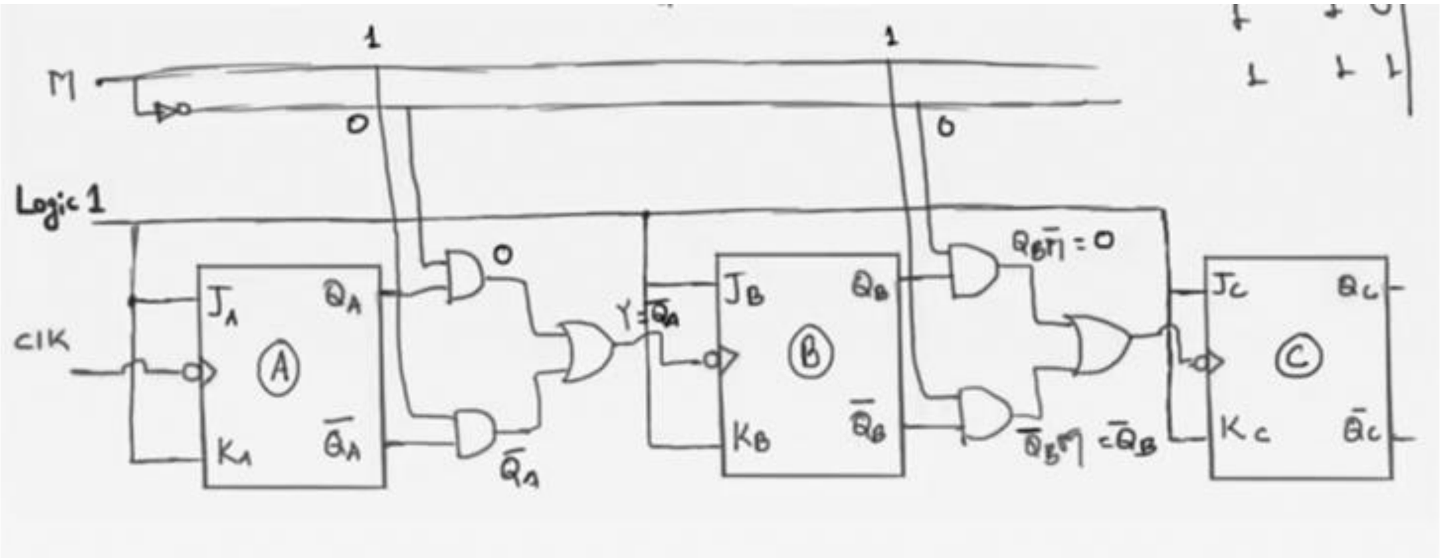


M	Q	$\bar{Q}$	Y
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	0
1	1	1	1

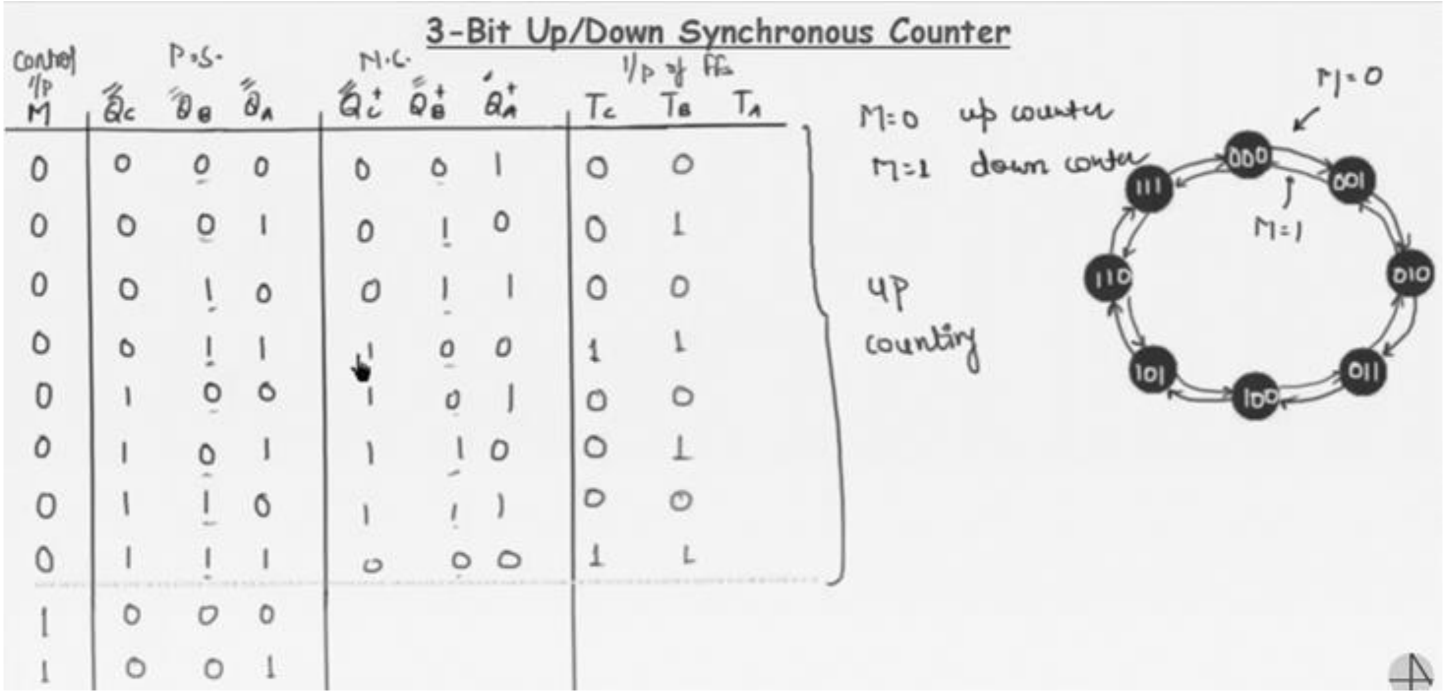
M \ $Q\bar{Q}$	00	01	11	10
0	0	0	1	1
1	0	1	1	0

$Y = I + II$

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



3 BIT UD SYNCHRONOUS COUNTER



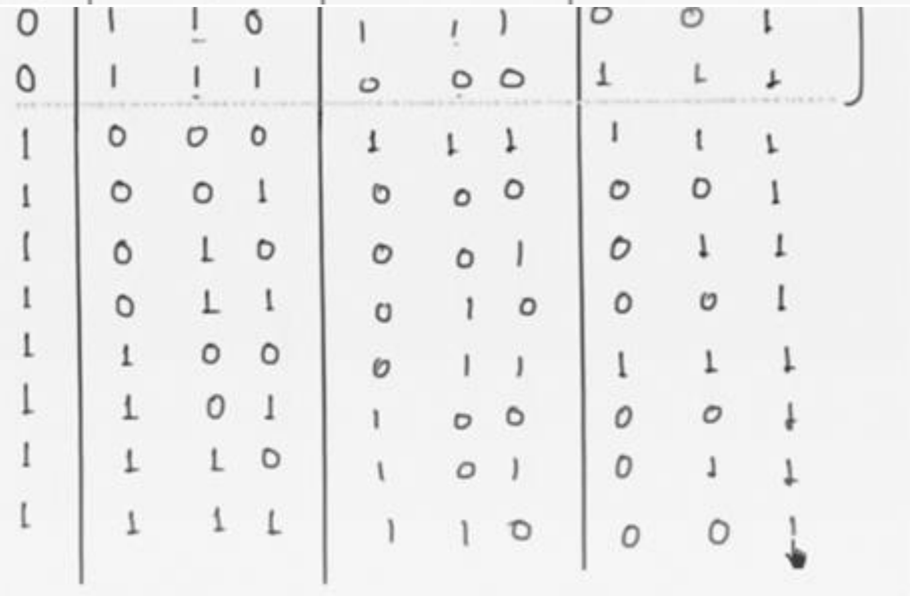
3-Bit Up/Down Synchronous Counter

Control I/P	P.S.			N.C.			I/P of FFs		
M	Q_C	Q_B	Q_A	Q_C^+	Q_B^+	Q_A^+	T_C	T_B	T_A
0	0	0	0	0	0	1	0	0	
0	0	0	1	0	1	0	0	1	
0	0	1	0	0	1	1	0	0	
0	0	1	1	1	0	0	1	1	
0	1	0	0	1	0	1	0	0	
0	1	0	1	1	1	0	0	1	
0	1	1	0	1	1	1	0	0	
0	1	1	1	0	0	0	1	1	
1	0	0	0						
1	0	0	1						

M=0 up counter

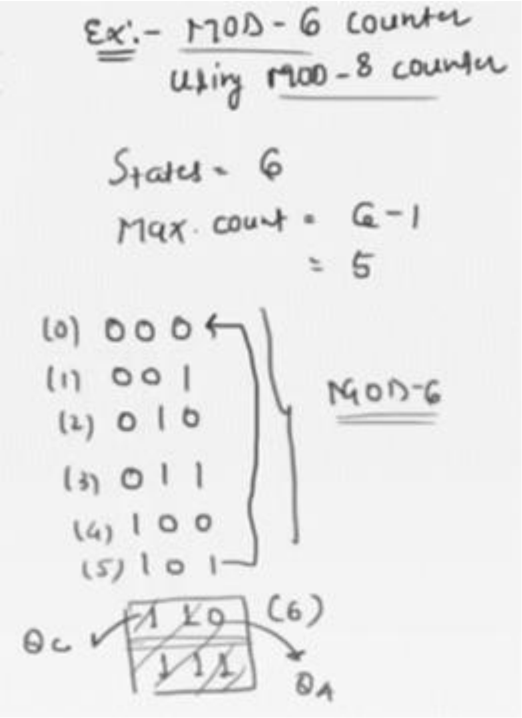
M=1 down counter

up counting



M	Q_C	Q_B	Q_A	Q_C^+	Q_B^+	Q_A^+	T_C	T_B	T_A
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	1	1
1	0	0	1	0	0	0	0	0	1
1	0	1	0	0	0	1	0	1	1
1	0	1	1	0	1	0	0	0	1
1	1	0	0	0	1	1	1	1	1
1	1	0	1	1	0	0	0	0	1
1	1	1	0	1	0	1	0	1	1
1	1	1	1	1	1	0	0	0	1

Modulus of the Counter & Counting up to Particular Value



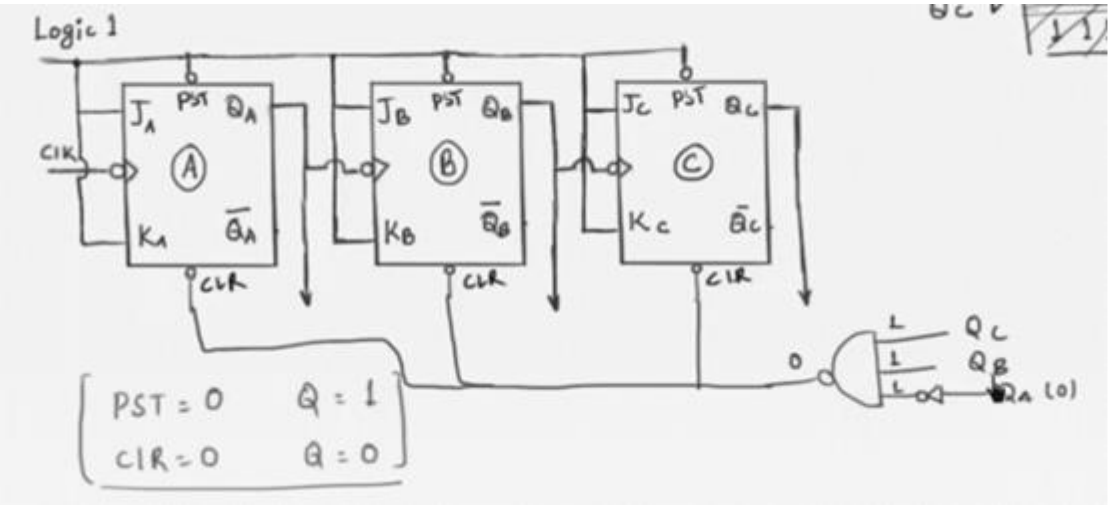
Ex:- MOD-6 counter
using MOD-8 counter

States = 6
Max. count = 6 - 1
= 5

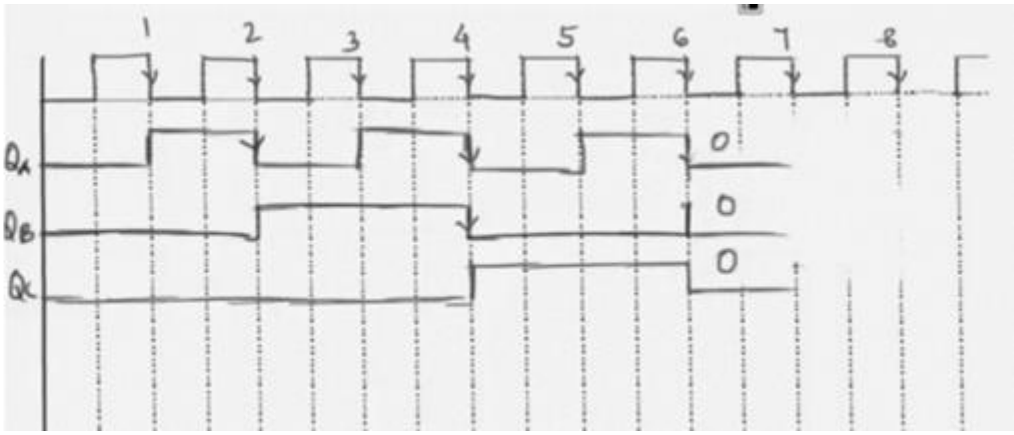
(0) 0 0 0
(1) 0 0 1
(2) 0 1 0
(3) 0 1 1
(4) 1 0 0
(5) 1 0 1

MOD-6

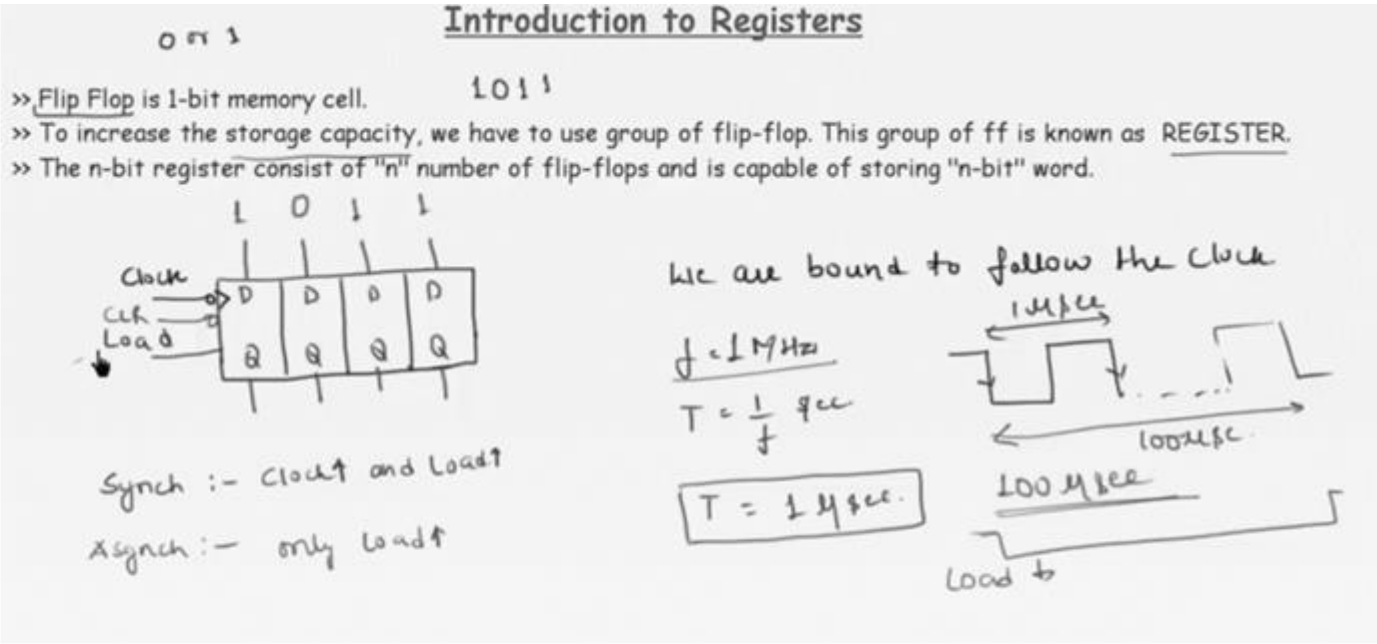
Qc ~~1 1 0~~ (6) QA
~~1 1 1~~



PST = 0 Q = 1
CLR = 0 Q = 0



RESIGTER



Introduction to Registers

0 or 1

>> Flip Flop is 1-bit memory cell. 1011

>> To increase the storage capacity, we have to use group of flip-flop. This group of ff is known as REGISTER.

>> The n-bit register consist of "n" number of flip-flops and is capable of storing "n-bit" word.

We are bound to follow the Clock

$f = 1\,MHz$

$T = \frac{1}{f}$ sec

$\boxed{T = 1\,\mu sec.}$

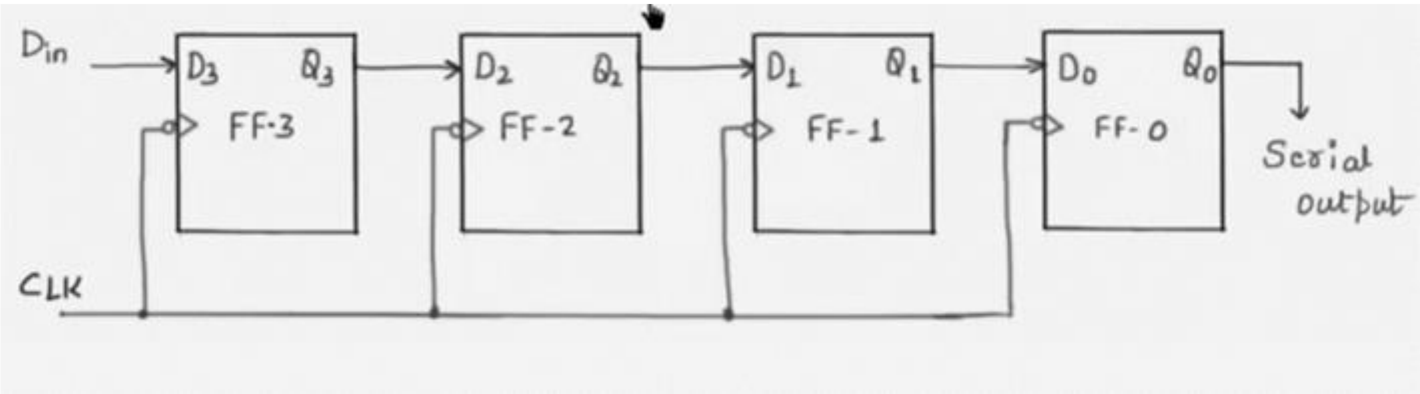
100 μsec

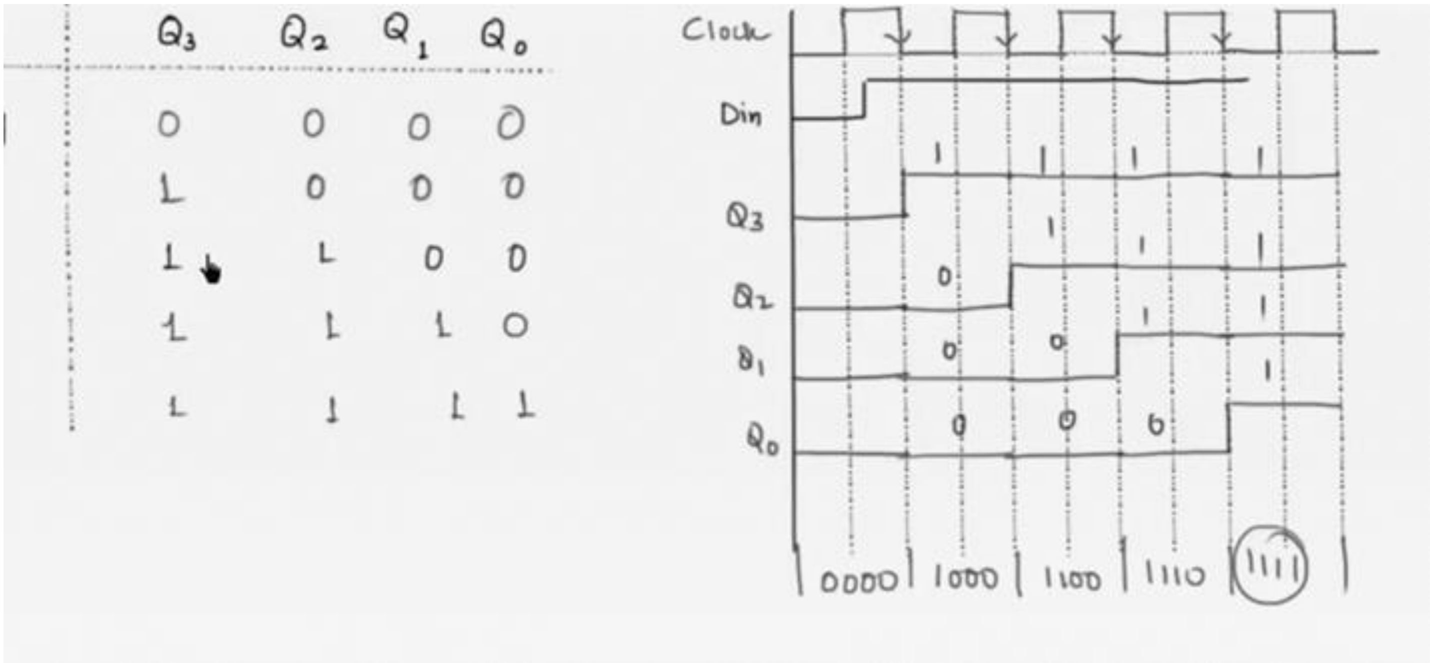
Load

Synch :- Clock↑ and Load↑

Asynch :- only Load↑

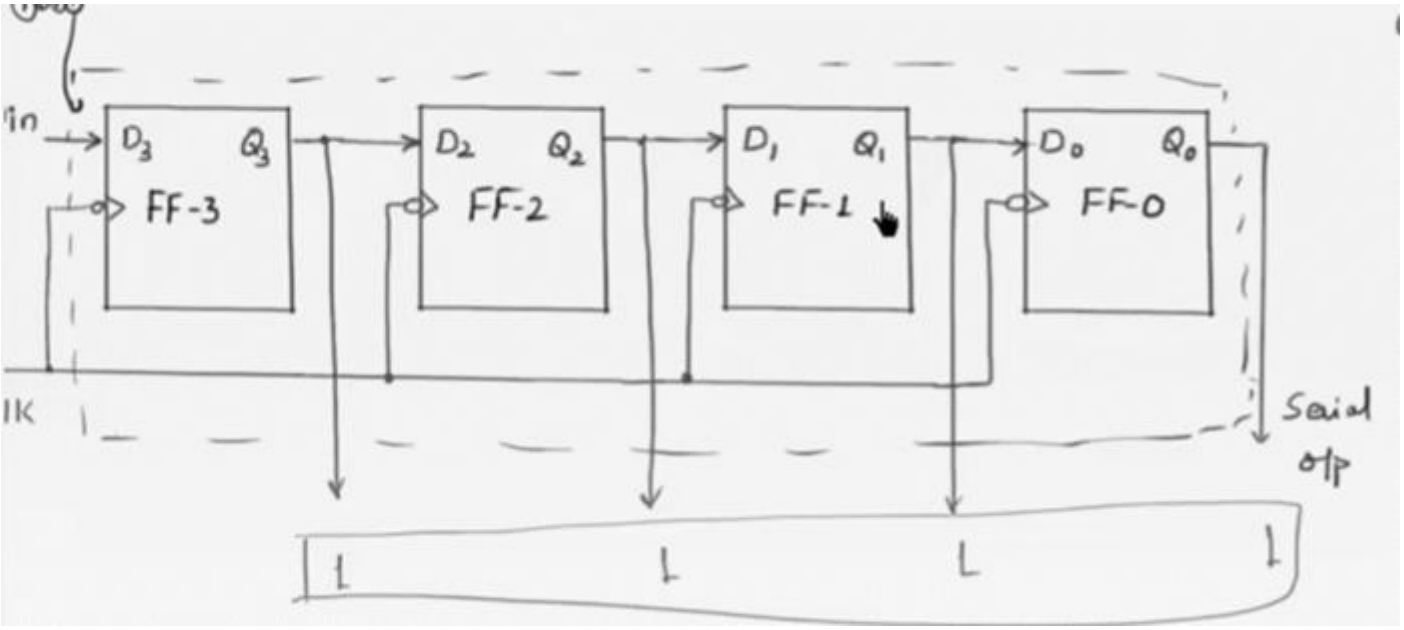
SISO





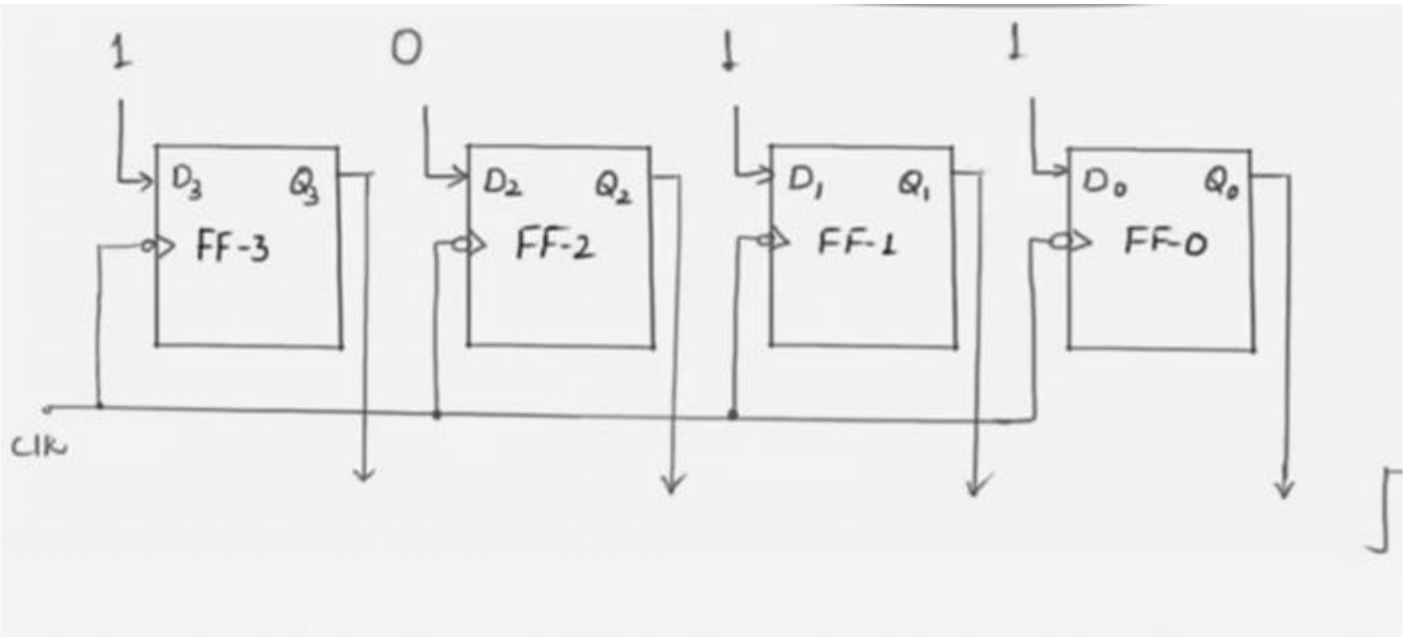
Q_3	Q_2	Q_1	Q_0
0	0	0	0
1	0	0	0
1	1	0	0
1	1	1	0
1	1	1	1

SIPO



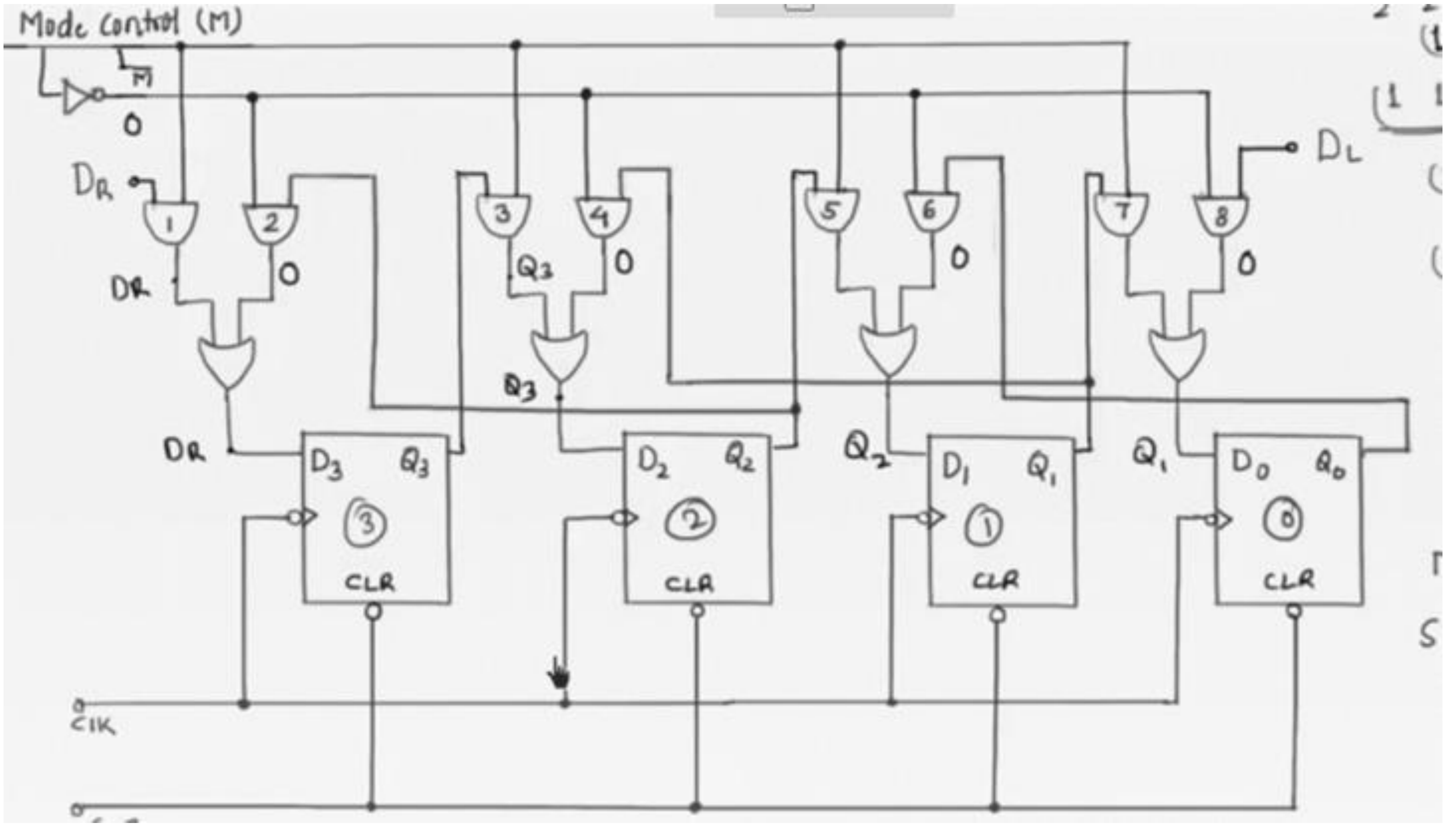
in
D3
Q3
FF-3
D2
Q2
FF-2
D1
Q1
FF-1
D0
Q0
FF-0
lK
Serial
o/p
1
1
1
1

PIPO

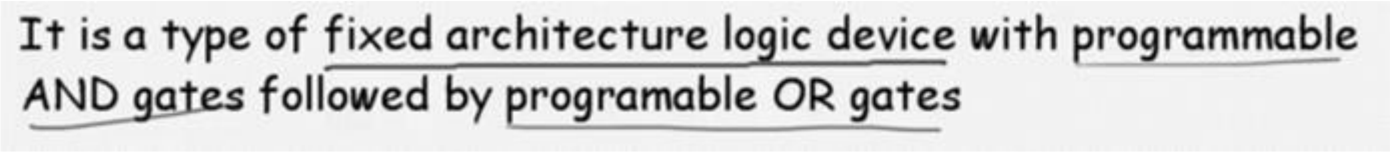


1
0
1
1
D3
Q3
FF-3
D2
Q2
FF-2
D1
Q1
FF-1
D0
Q0
FF-0
Clk

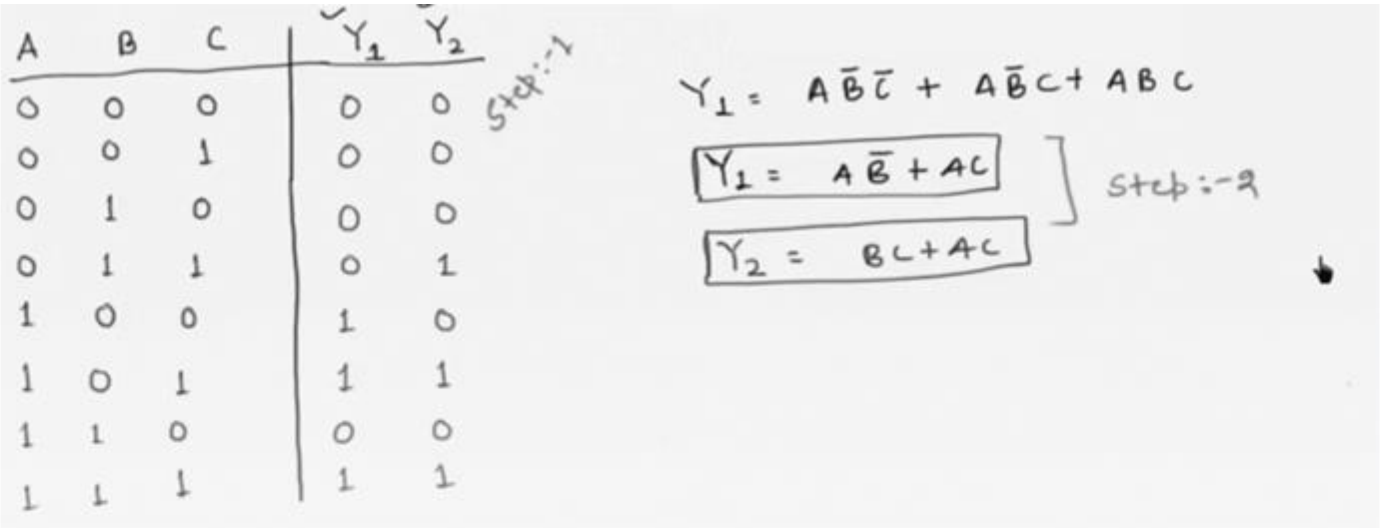
Bidirectional Shift Register



Programmable Logic Array (PLA)



It is a type of fixed architecture logic device with programmable AND gates followed by programable OR gates



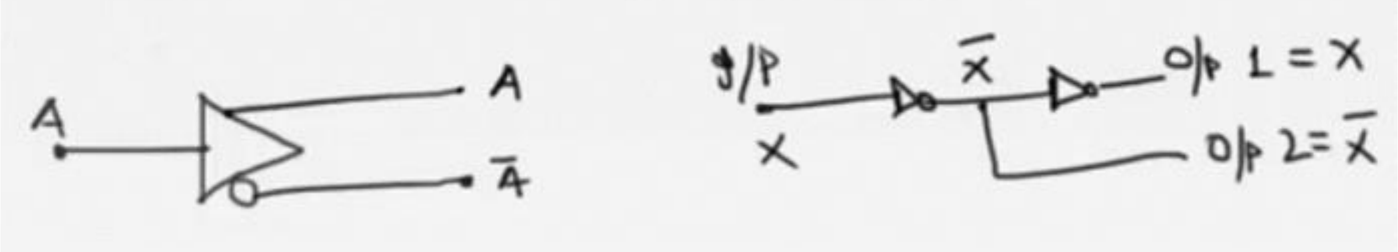
A	B	C	Y_1	Y_2
0	0	0	0	0
0	0	1	0	0
0	1	0	0	0
0	1	1	0	1
1	0	0	1	0
1	0	1	1	1
1	1	0	0	0
1	1	1	1	1

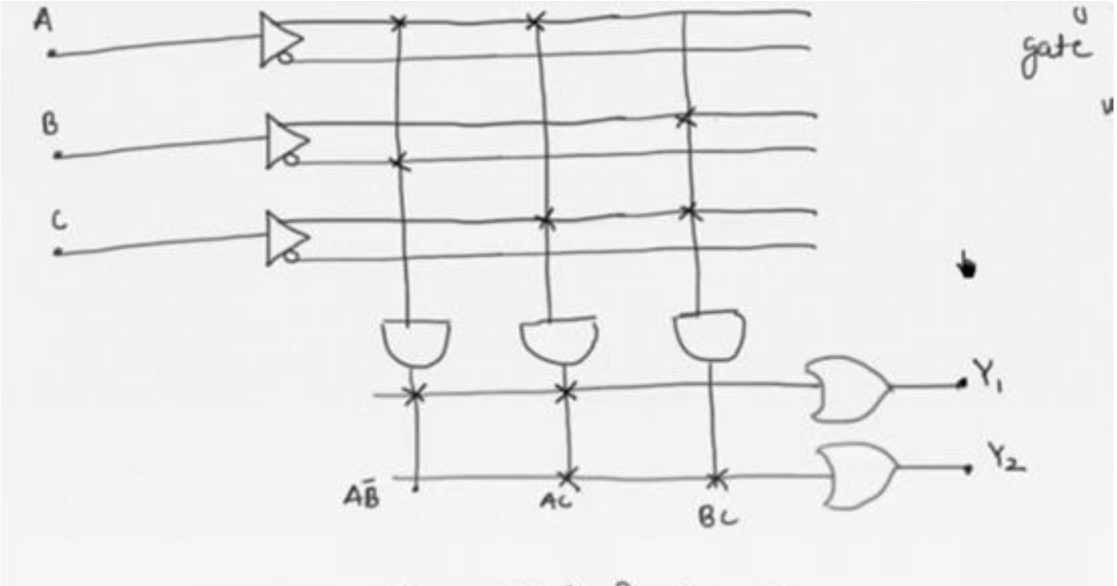
Step:-1

$Y_1 = A\bar{B}\bar{C} + A\bar{B}C + ABC$

$$\left.\begin{array}{l}\boxed{Y_1 = A\bar{B} + AC}\\ \boxed{Y_2 = BC + AC}\end{array}\right]$$ Step:-2

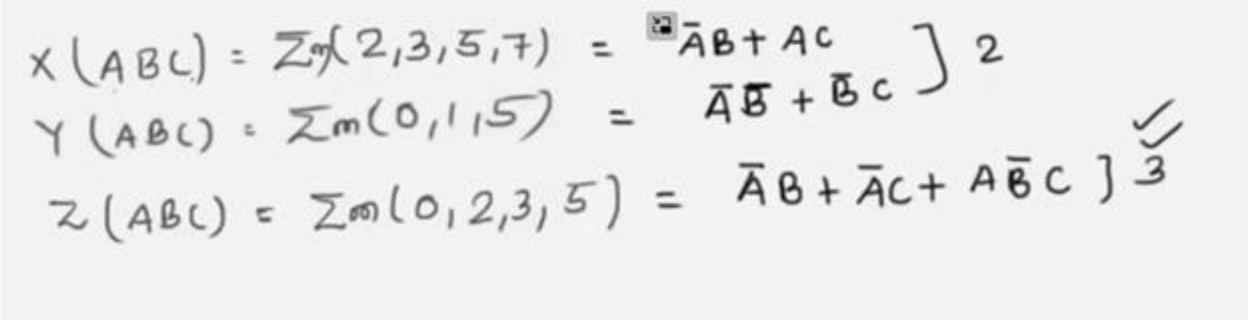
Step:-3 No. of $\boxed{\text{i/p buffer}}$





prog. OR = No. of function = 2

Programmable Array Logic (PAL)

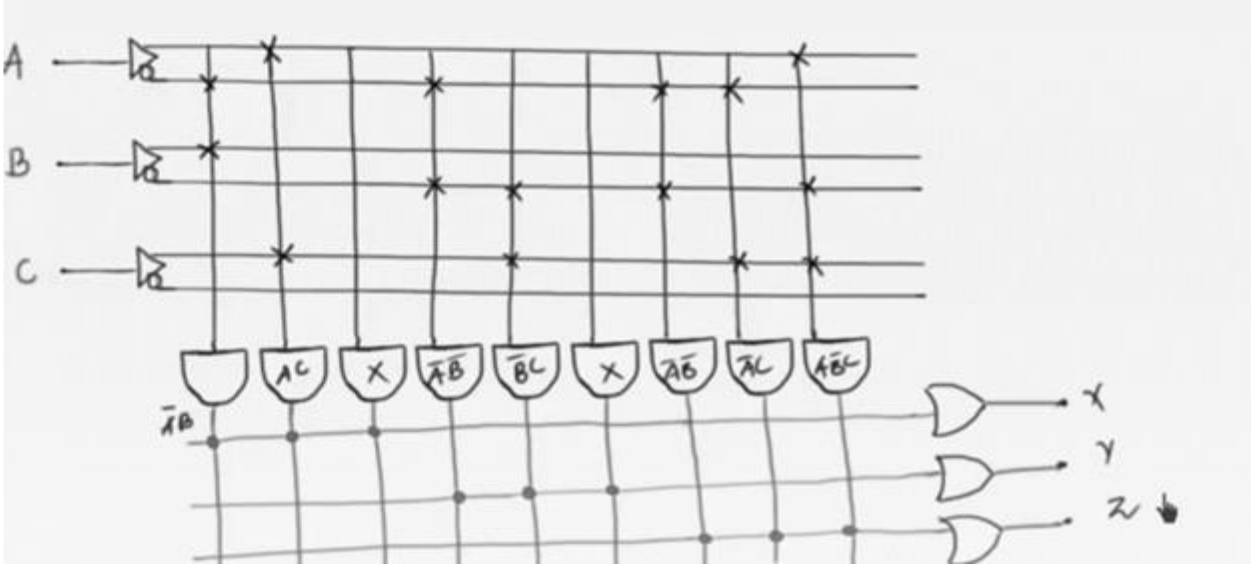


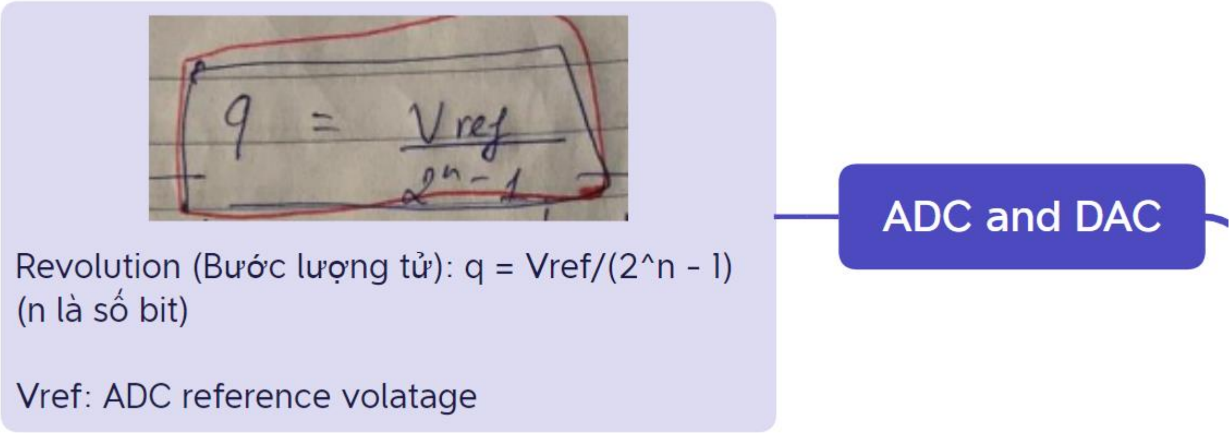
$X(ABC) = \Sigma m(2,3,5,7) = \bar{A}B + AC$] 2

$Y(ABC) = \Sigma m(0,1,5) = \bar{A}\bar{B} + \bar{B}C$

$Z(ABC) = \Sigma m(0,2,3,5) = \bar{A}B + \bar{A}C + A\bar{B}C$] 3

3 MINTERM



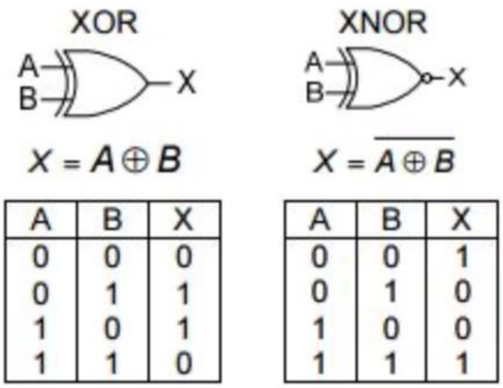


$$q = \frac{V_{ref}}{2^n - 1}$$

Revolution (Bước lượng tử): q = Vref/(2^n - 1) (n là số bit)

Vref: ADC reference volatage

LOGIC GATE



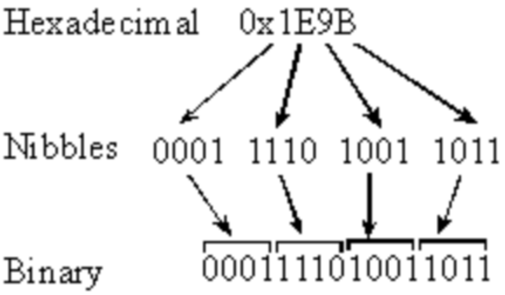
$X = A \oplus B$

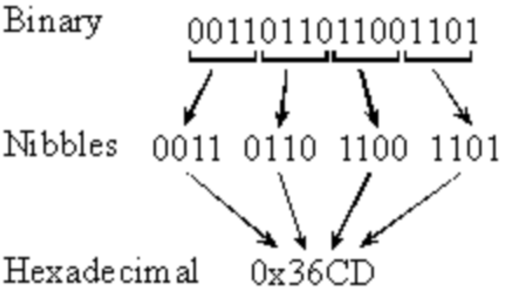
A	B	X
0	0	0
0	1	1
1	0	1
1	1	0

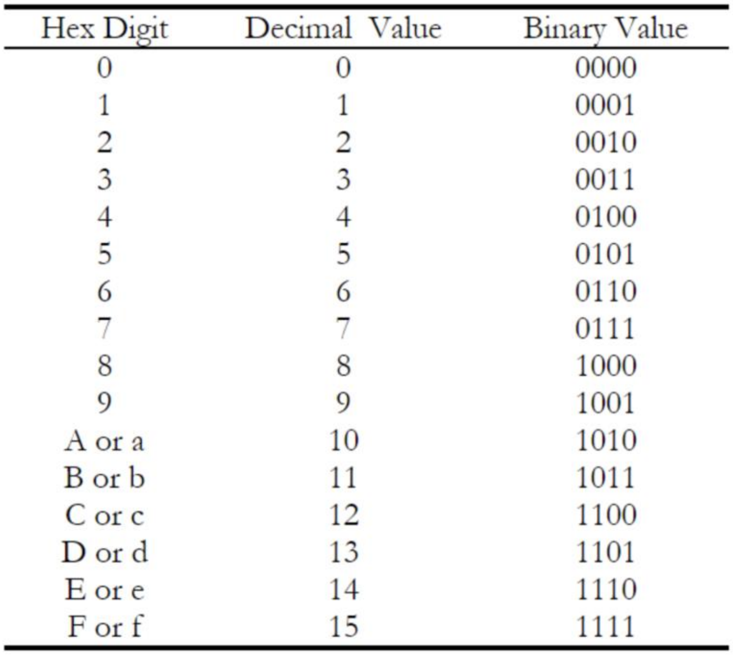
$X = \overline{A \oplus B}$

A	B	X
0	0	1
0	1	0
1	0	0
1	1	1

BINARY CONVERT



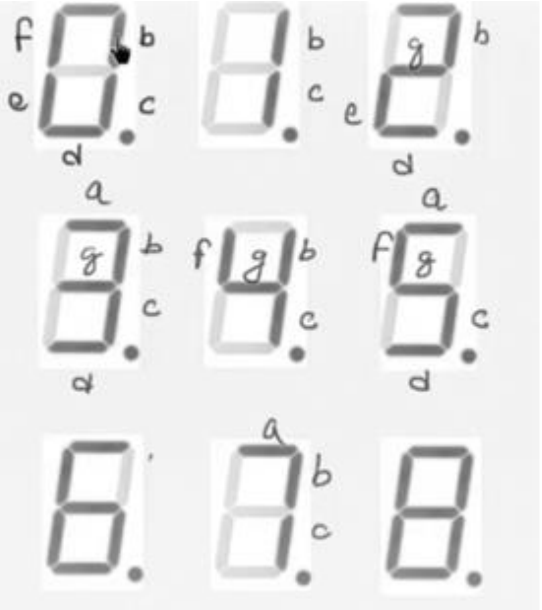


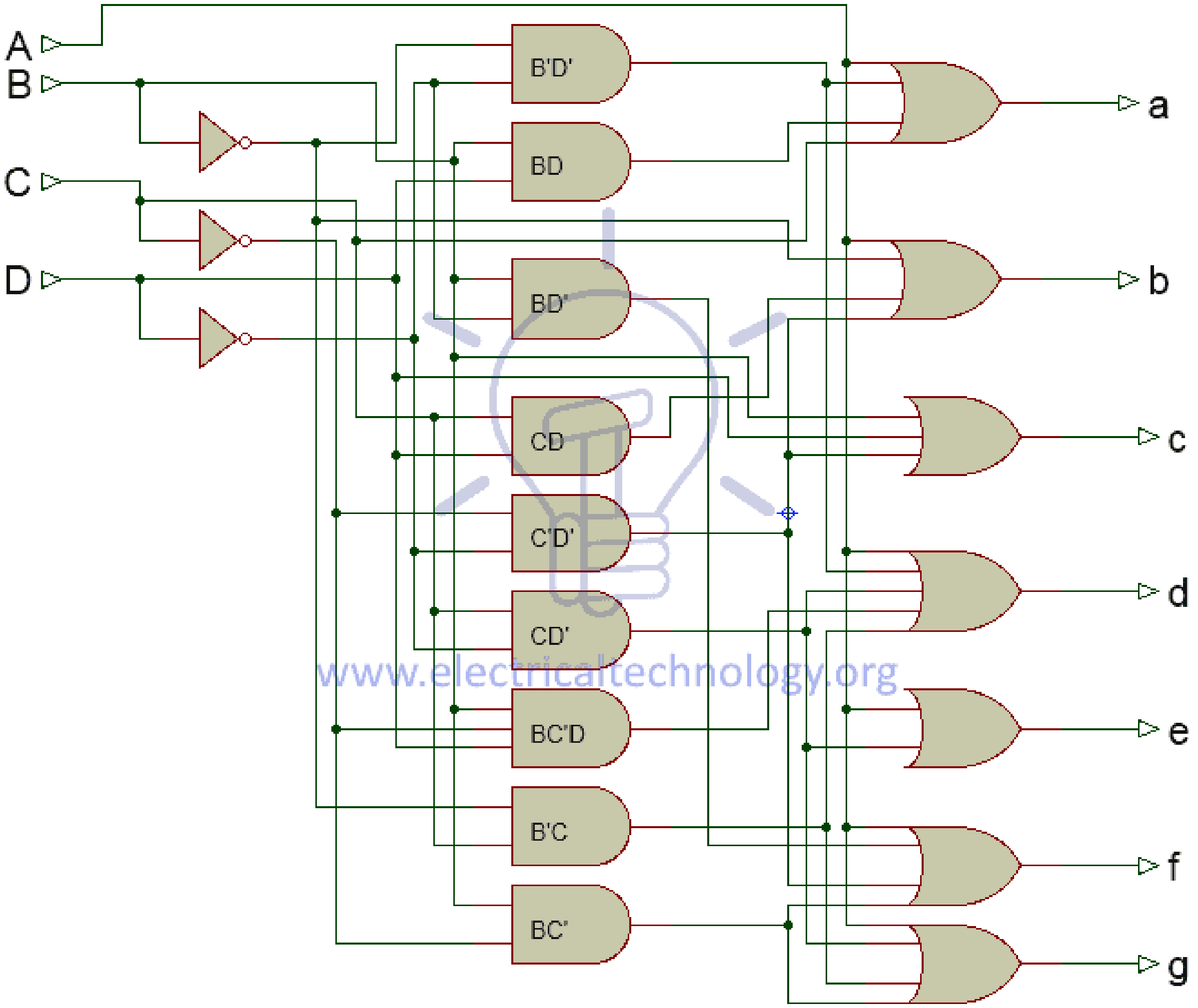


Hex Digit	Decimal Value	Binary Value
0	0	0000
1	1	0001
2	2	0010
3	3	0011
4	4	0100
5	5	0101
6	6	0110
7	7	0111
8	8	1000
9	9	1001
A or a	10	1010
B or b	11	1011
C or c	12	1100
D or d	13	1101
E or e	14	1110
F or f	15	1111

7 SEG DECODER

b_3	b_2	b_1	b_0	a	b	c	d	e	f	g
0	0	0	0 (0)	1	1	1	1	1	1	0
0	0	0	1 (1)	0	1	1	0	0	0	0
0	0	1	0 (2)	1	1	0	1	1	0	1
0	0	1	1 (3)	1	1	1	1	0	0	1
0	1	0	0 (4)	0	1	1	0	0	1	1
0	1	0	1 (5)	1	0	1	1	0	1	1
0	1	1	0 (6)	1	0	1	1	1	1	1
0	1	1	1 (7)	1	1	1	0	0	0	0
1	0	0	0 (8)	1	1	1	1	1	1	1
1	0	0	1 (9)	1	1	1	1	0	1	1





Schematic of BCD to 7-Segment Decoder

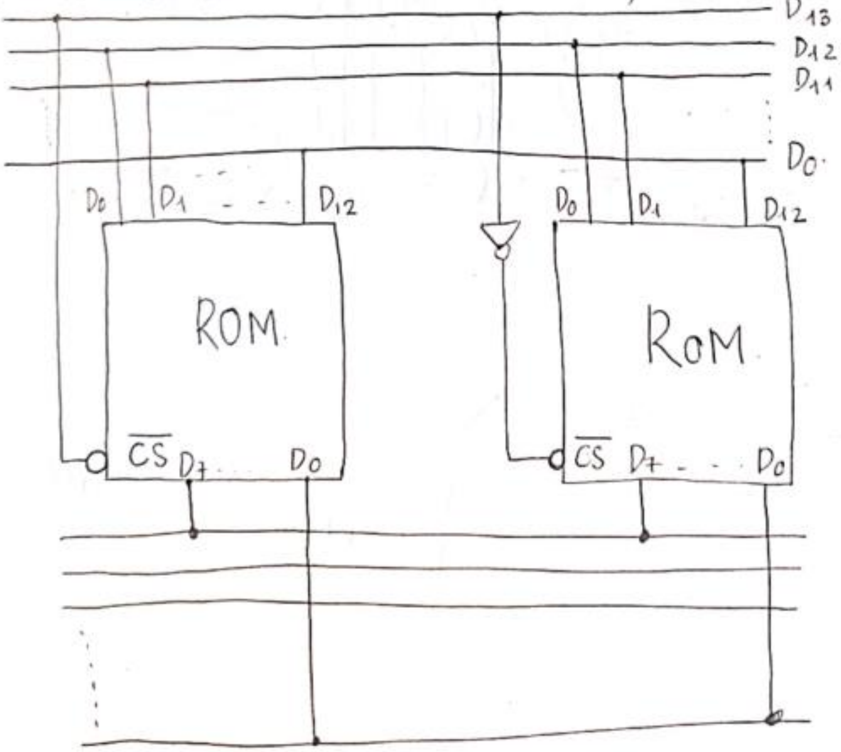
Câu 5 (1 điểm)

Thành lập bộ nhớ có kích thước 16K x 8 bit từ các bộ nhớ có kích thước 8K x 8 bit và các phần tử AND, OR, NOT

Câu 6 (1 điểm)

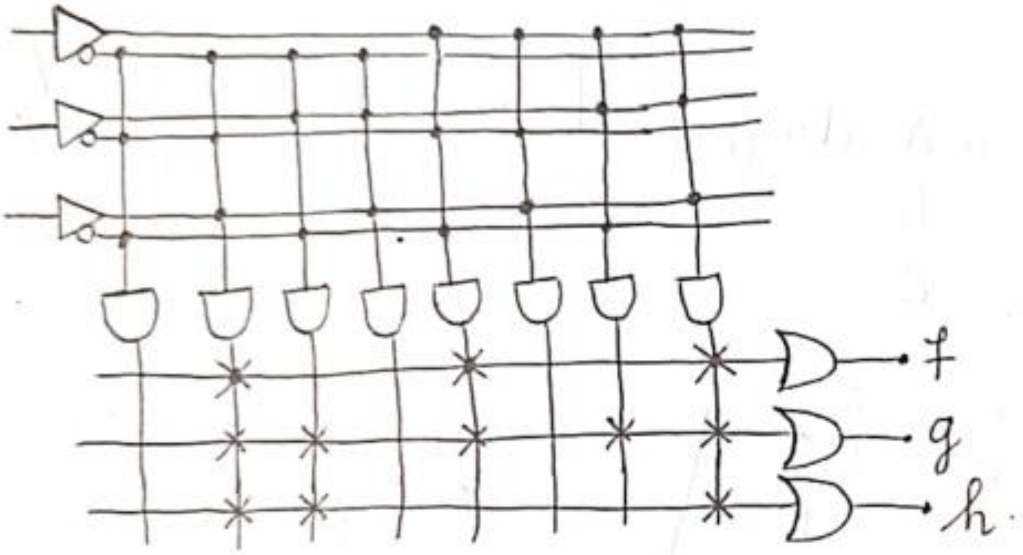
Cho các biểu thức hàm logic sau (a = LSB, c = MSB):
f(a, b, c) = Σ (1,4,7);
g(a, b, c) = $\prod$ (0,3,5);
h(a, b, c) = Σ (1,2,7);
Hãy sử dụng ROM 64 x 8 bit để thực hiện các hàm này.

Câu 5. $16K \times 8\,bit = 2^4 . 2^{10} \times 8bit = 2^{14} \times 8bit$, $8k \times 8bit = 2^{13} \times 8bit$

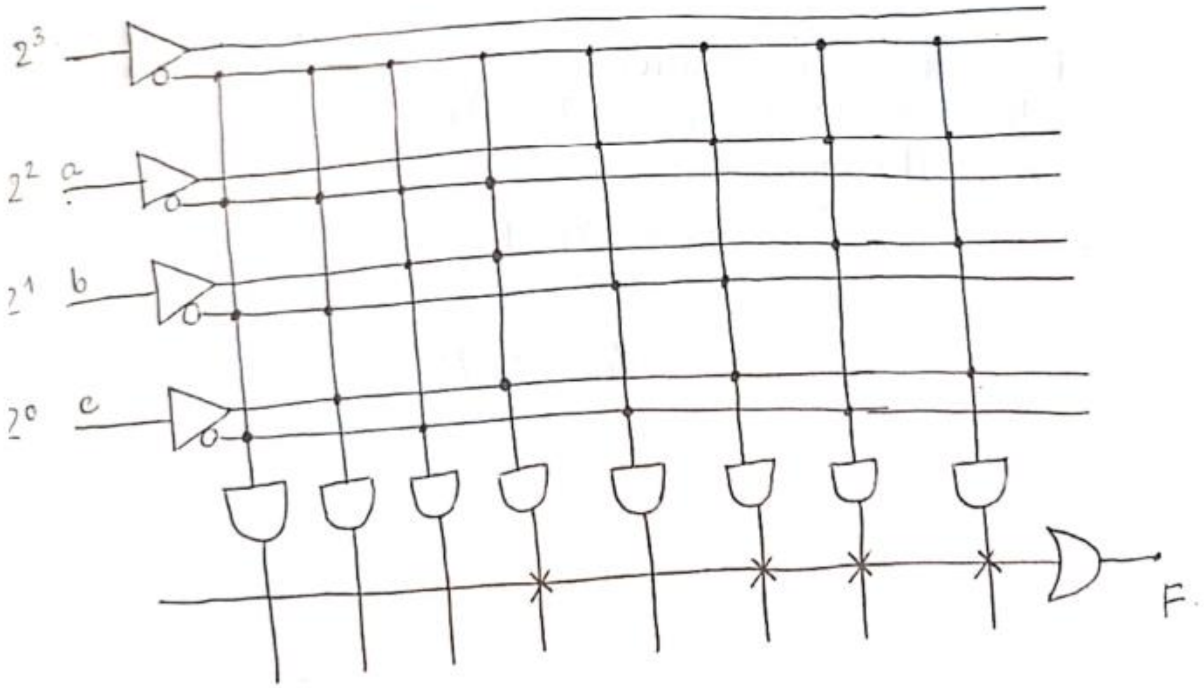


Câu 6:

$$f(a,b,c) = \Sigma(1,4,7)$$
$$g(a,b,c) = \Pi(0,3,5) = \Sigma(1,2,4,6,7)$$
$$h(a,b,c) = \Sigma(1,2,7).$$



d) PROM 4 địa chỉ, mỗi ô nhớ chứa 1 bit dữ liệu


2^3
2^2 a
2^1 b
2^0 c
F