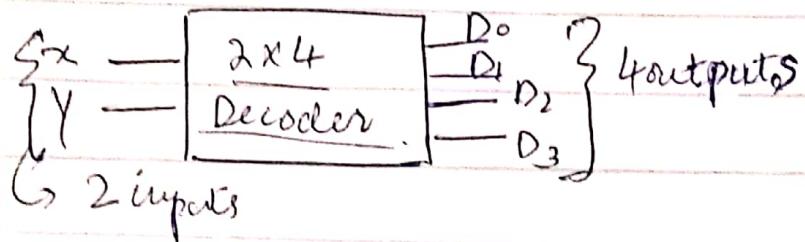


## PLDs:-

### Decoder:-

A decoder is a combinational circuit that converts binary information from  $n$  input lines to maximum of  $2^n$  unique outputs.



### Truth Table:-

$X$	$Y$	$D_0$	$D_1$	$D_2$	$D_3$
0	0	0	1	0	0
0	1	1	0	0	0
1	0	2	0	0	1
1	1	3	0	0	1

$$D_0 = (\bar{x} \bar{y})$$

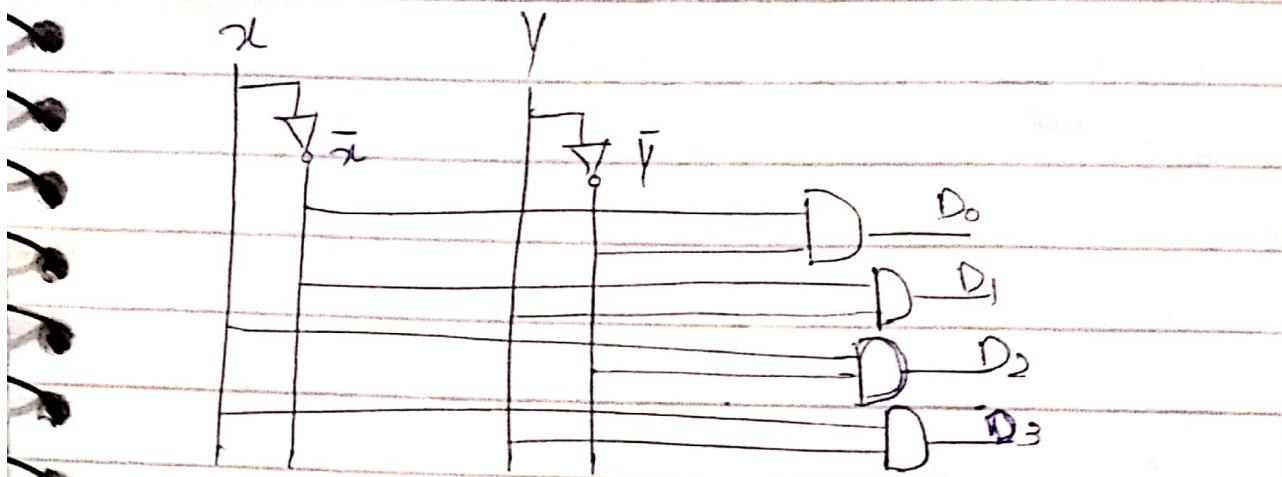
$$D_1 = (\bar{x} y)$$

$$D_2 = (x \bar{y})$$

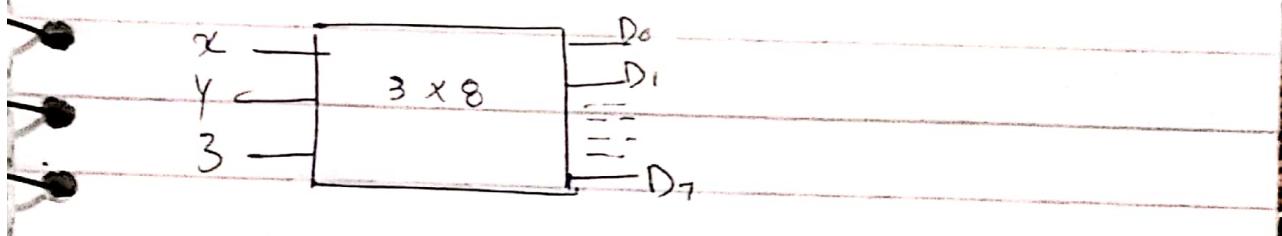
$$D_3 = (xy)$$

functions  
minterms

Circuit Diagram :-



Design 3x8 Decoder:-



$x$	$y$	$z$	$D_0$	$D_1$	$D_2$	$D_3$	$D_4$	$D_5$	$D_6$	$D_7$
0	0	0	1	0	0	0	0	0	0	0
0	0	1	0	1	0	0	0	0	0	0
0	1	0	0	0	1	0	0	0	0	0
0	1	1	0	0	0	1	0	0	0	0
1	0	0	0	0	0	0	0	1	0	0
1	0	1	0	0	0	0	0	0	1	0
1	1	0	0	0	0	0	0	0	1	0
1	1	1	0	0	0	0	0	0	0	1

$$D_0 = \bar{x}\bar{y}\bar{z}$$

$$D_1 = \bar{x}\bar{y}z$$

$$D_2 = \bar{x}yz$$

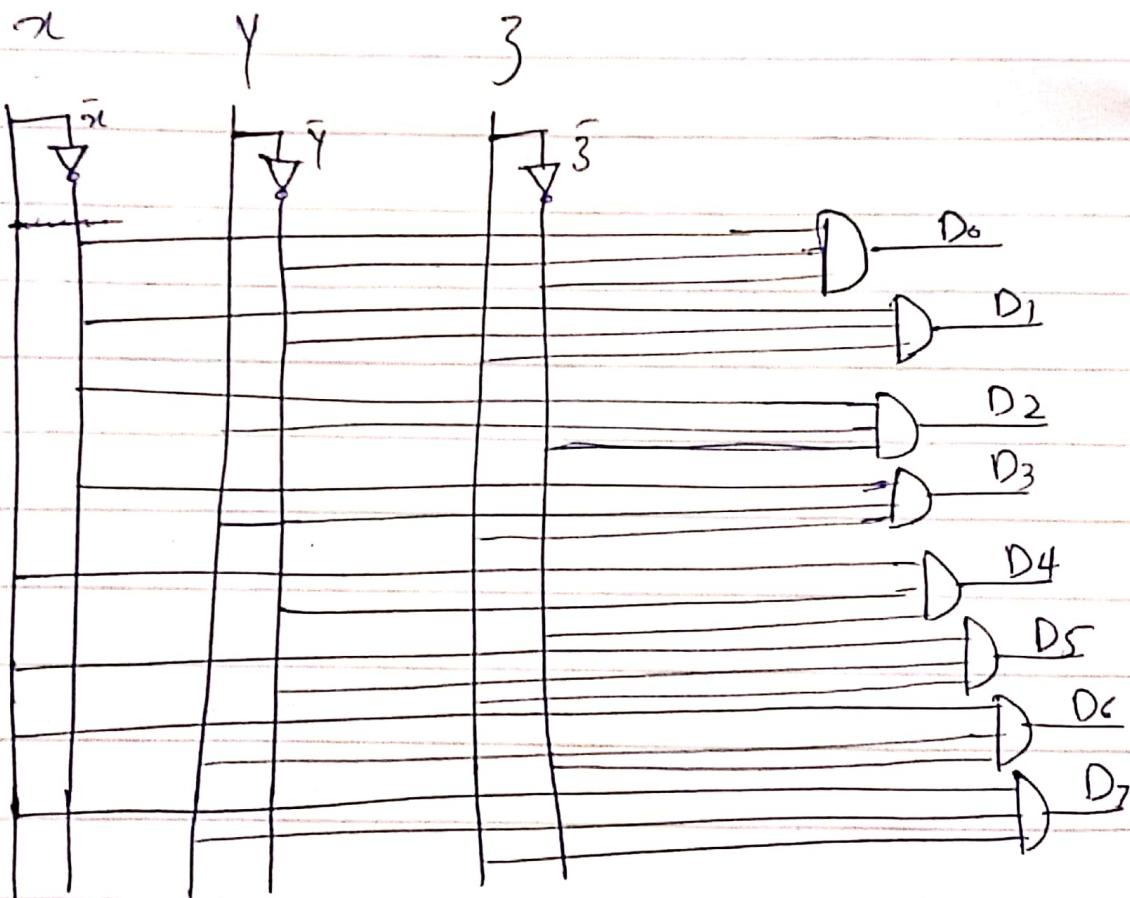
$$D_3 = \bar{x}yz$$

$$D_4 = xy\bar{z}$$

$$D_5 = xy\bar{z}$$

$$D_6 = xyz$$

$$D_7 = xyz$$



~~then don't perform~~ | when I not perform

Enable input to control the outputs, if we  
don't want any function then we use  
enable. [Enable =  $\overline{01}$ , Disable = 0]  
low  $\rightarrow 0$  

E	x	y	D <sub>0</sub>	D <sub>1</sub>	D <sub>2</sub>	D <sub>3</sub>	
0	X	X	Don't care	0	0	0	0
1	0	0	1	0	0	0	✓
1	0	1	0	1	0	0	
1	1	0	0	0	1	0	
1	1	1	0	0	0	1	

$$D_0 = E \bar{x} \bar{y}$$

$$D_1 = E \bar{x} y$$

$$D_2 = E x \bar{y}$$

$$D_3 = E x y$$

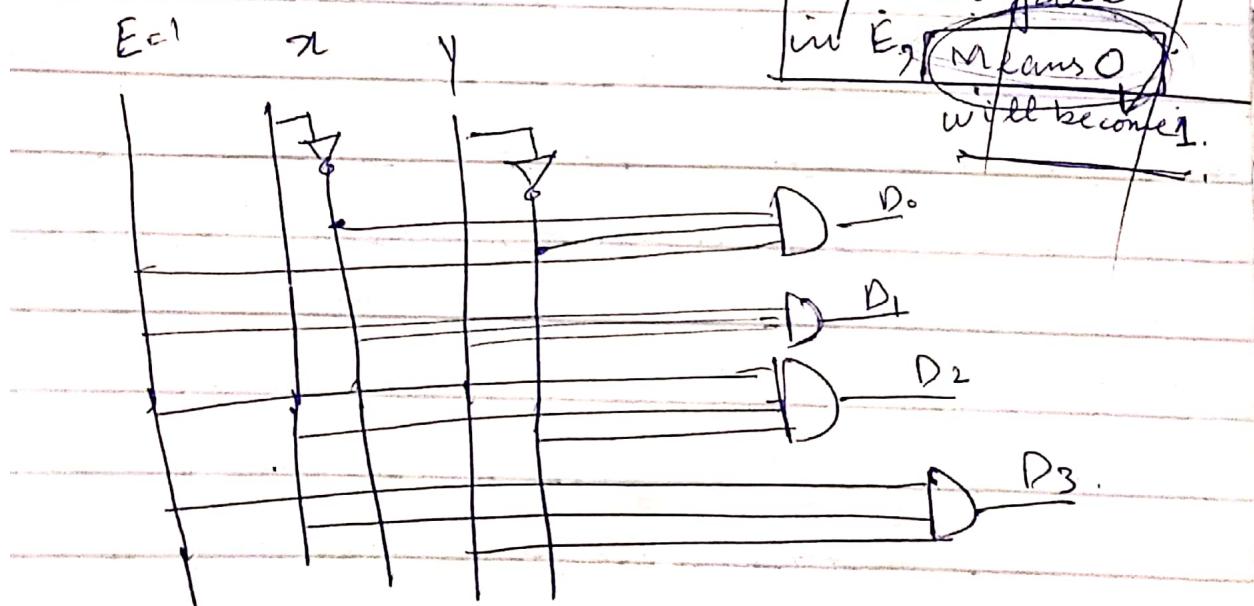
Enable input active high means '1'

Active low

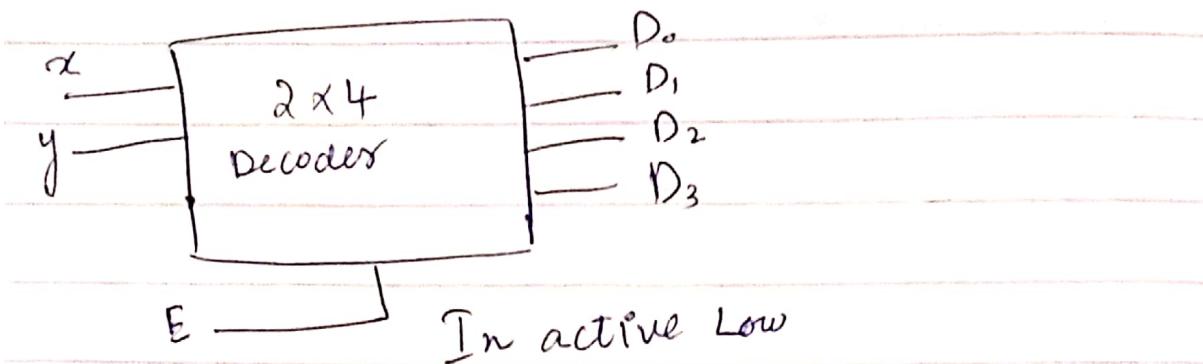
then not gate

in E,  $\overline{01}$  means 0

will become 1.

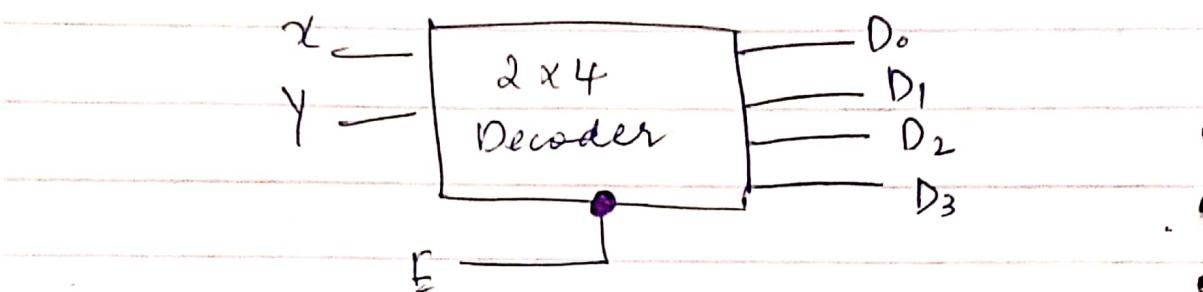


Active high



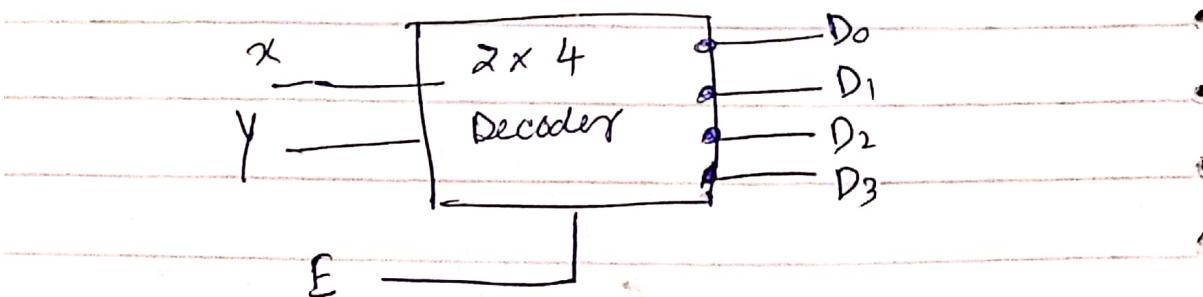
In active Low

then :-



\* If we were asked to solve decoder using NAND gate then we use NAND instead of AND and  $0 \rightarrow 1$  and  $1 \rightarrow 0$ , output will change.

NAND gate & diagram:-  
block

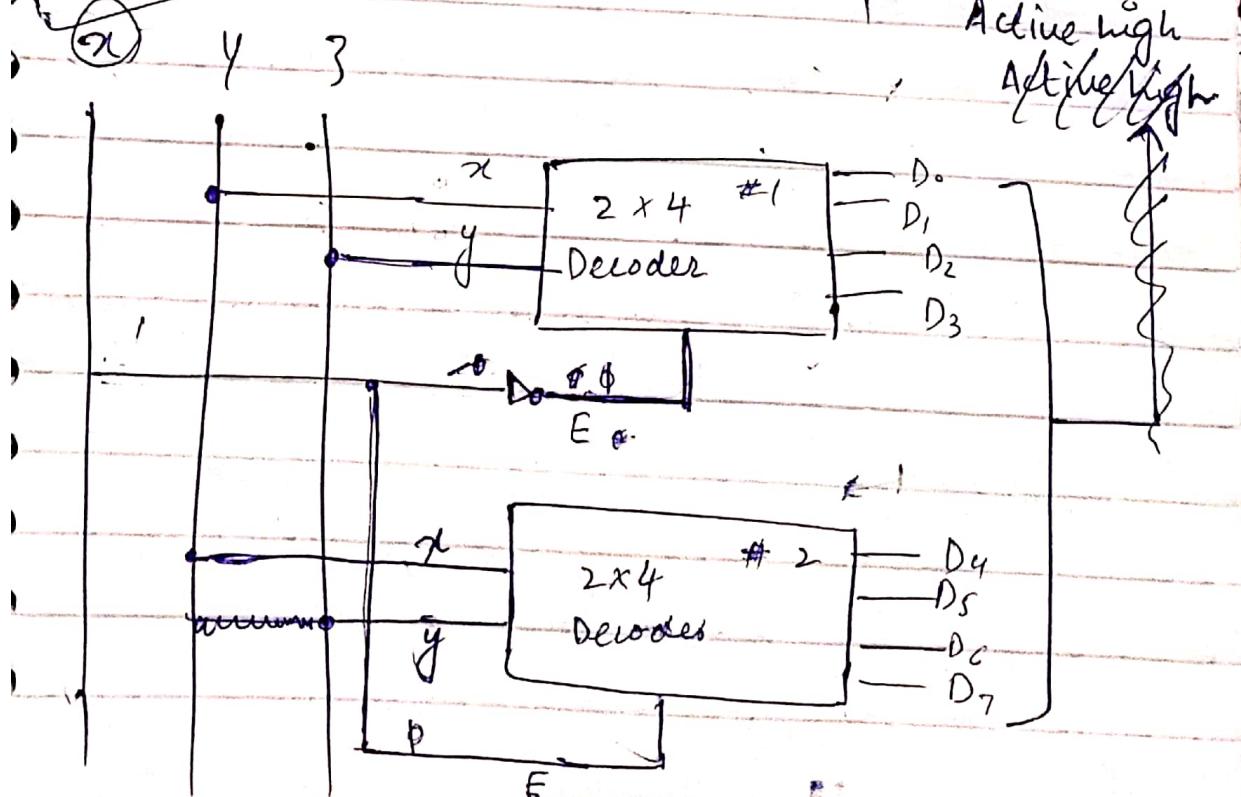


Design  $3 \times 8$  Decoder using  $2 \times 4$  Decoder and external circuitry.

→  $3 \times 8$  Table.

$x$	$y$	$z$	$D_0$	$D_1$	$D_2$	$D_3$	$D_4$	$D_5$	$D_6$	$D_7$
0	0	0	1	0	0	0	0	0	0	0
0	0	1	0	1	0	0	0	0	0	0
0	1	0	0	0	1	0	0	0	0	0
0	1	1	0	0	0	1	0	0	0	0
1	0	0	0	0	0	0	1	0	0	0
1	0	1	0	0	0	0	0	1	0	0
1	1	0	0	0	0	0	0	0	1	0
1	1	1	0	0	0	0	0	0	0	1

act as Enable.



### Example #01

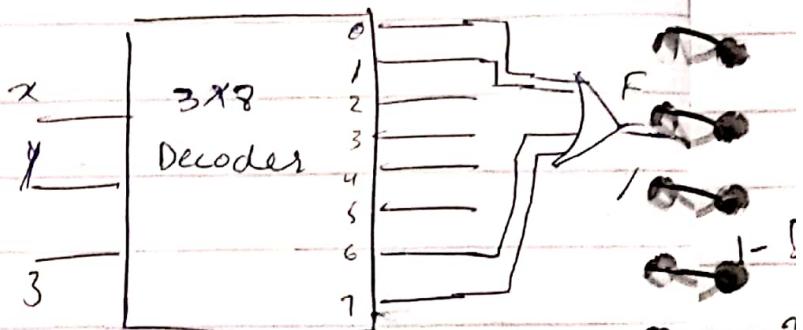
Design the given function using decoder and OR gate.

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

$$= m_0 + m_1 + m_6 + m_7$$

$$= E(0, 1, 6, 7)$$

x	y	z	F
0	0	0	1
0	0	1	1
0	1	0	0
0	1	1	0
1	0	0	0
1	0	1	0
1	1	0	1
1	1	1	1



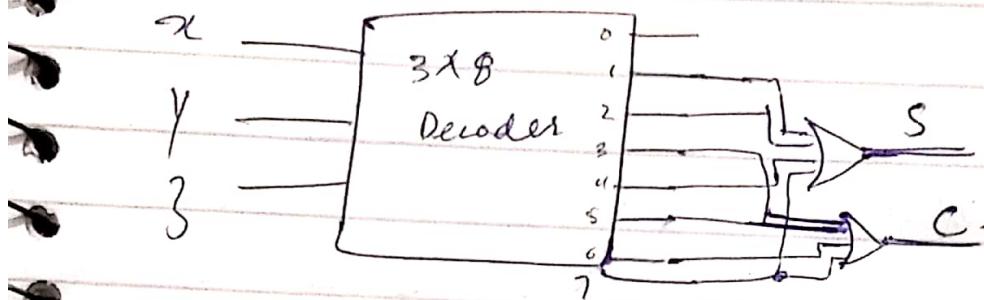
### Example #02

Design Full Adder circuit using decoder  
and external circuit. (OR gate).

x	y	z	C	S
0	0	0	0	0
0	0	1	0	1
0	1	0	0	1
0	1	1	1	0
1	0	0	0	1
1	0	1	1	0
1	1	0	1	0
1	1	1	1	1

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

$$C(x, y, z) = \Sigma(3, 5, 6, 7)$$

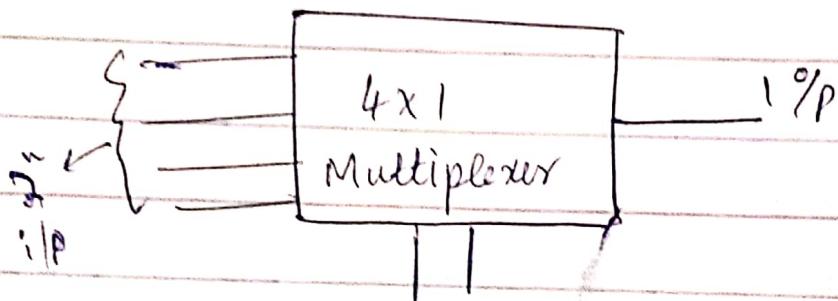


- 1- Design  $4 \times 16$  decoder using appropriate number of  $2 \times 4$  decoder and some external circuitry (if required).
- 2- Design BCD to decimal decoder
- 3- Design Full subtractor circuit using decoders and external OR gates

Decoder Expansion:-

Multiplexers:-

$2^n$  inputs  $\rightarrow$  single output

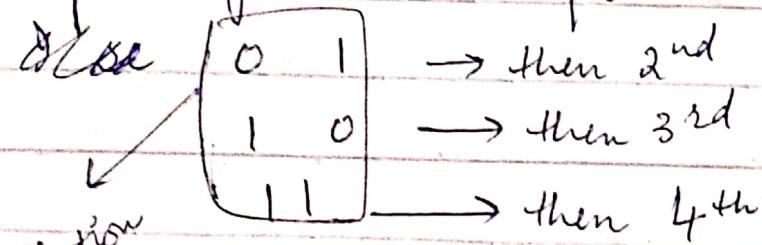


→ Select one input for output

Select lines, control lines

For  $4 \rightarrow 2$ ,  $8 \rightarrow 3$   
input  $2^{\text{nd}}$   $6^{\text{th}}$   $\rightarrow$  lines to go.  
selection

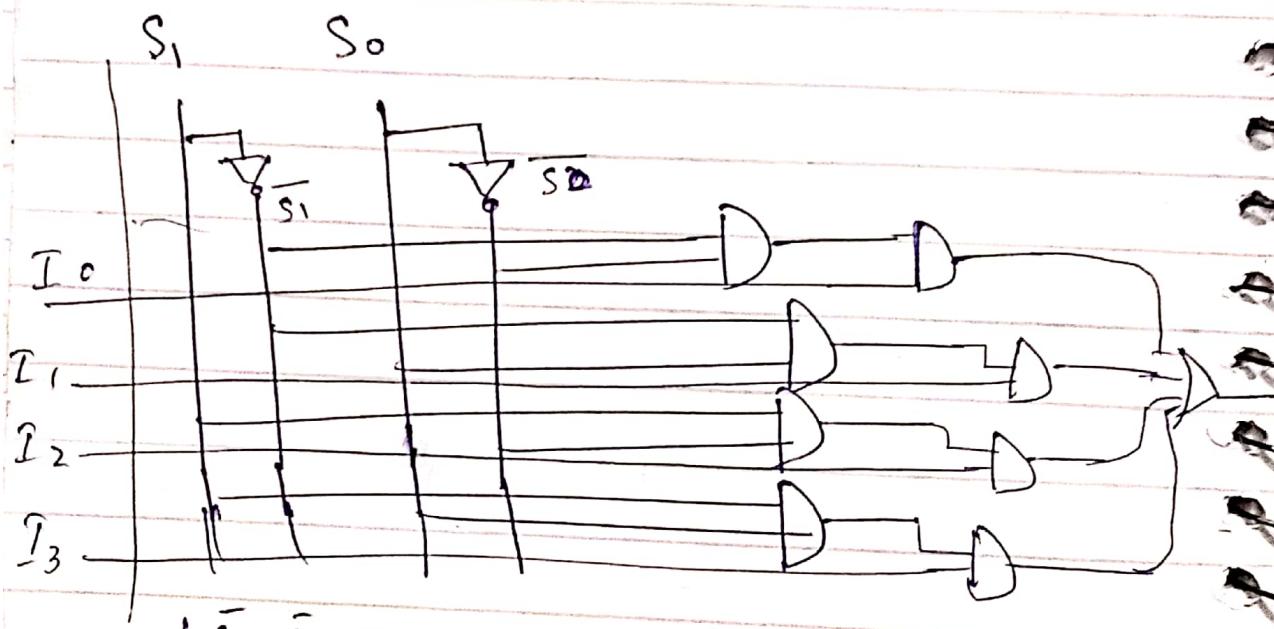
If selection line is 00 the 1<sup>st</sup> input goes in output.



Selection lines:

Selection line

$S_1$	$S_0$	O/P	E	A	B	C
0	0	1 <sup>st</sup> $I_0$	0	x	x	00
0	1	2 <sup>nd</sup> $I_1$				
1	0	3 <sup>rd</sup> $I_2$				
1	1	4 <sup>th.</sup> $I_3$				

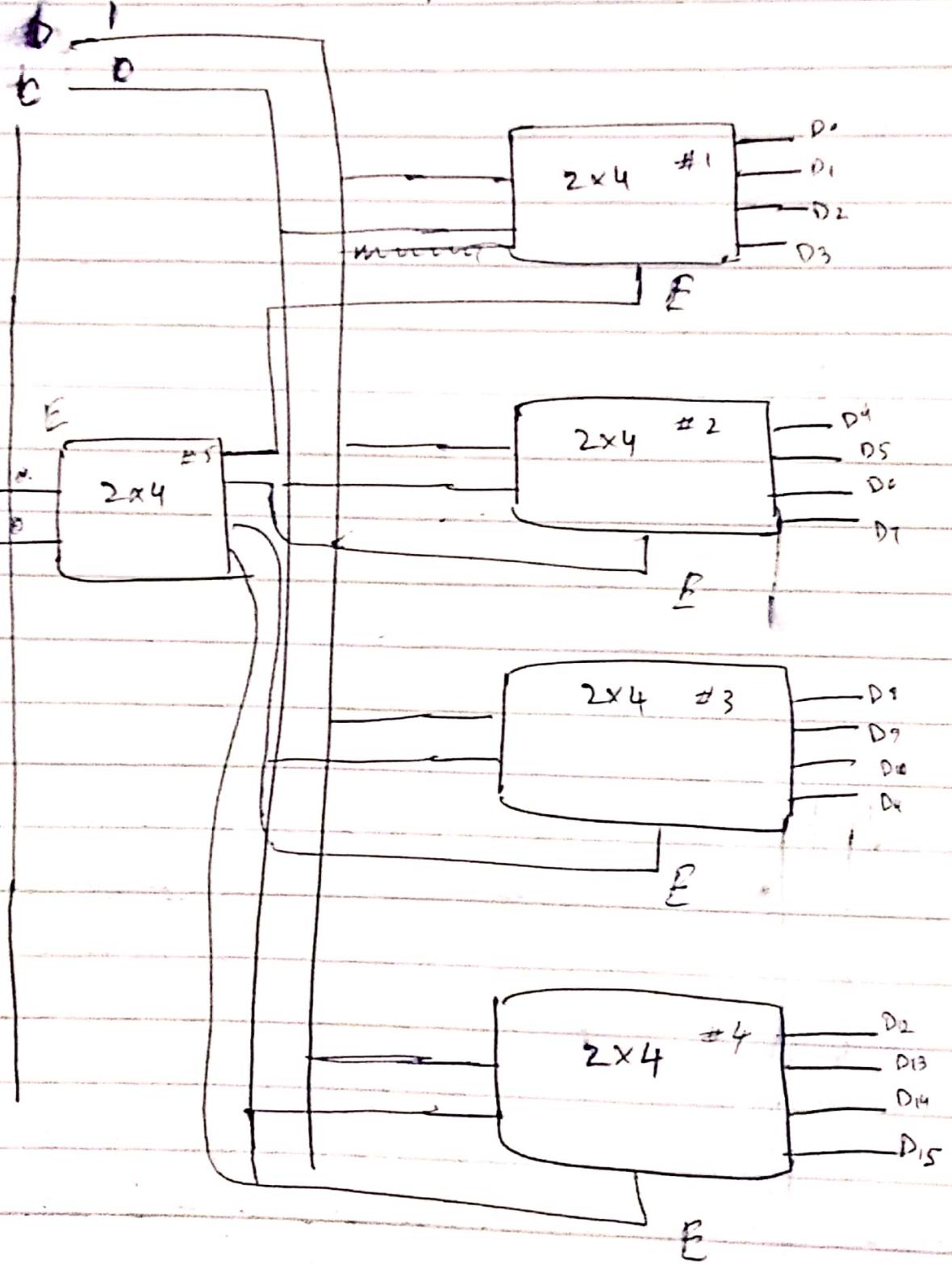


$$(\bar{S}_1, \bar{S}_0, I_0) + (\bar{S}_1, S_0, I_1) + (S_1, \bar{S}_0, I_2) + \\ (S_1, S_0, I_3)$$

4x16 decoder

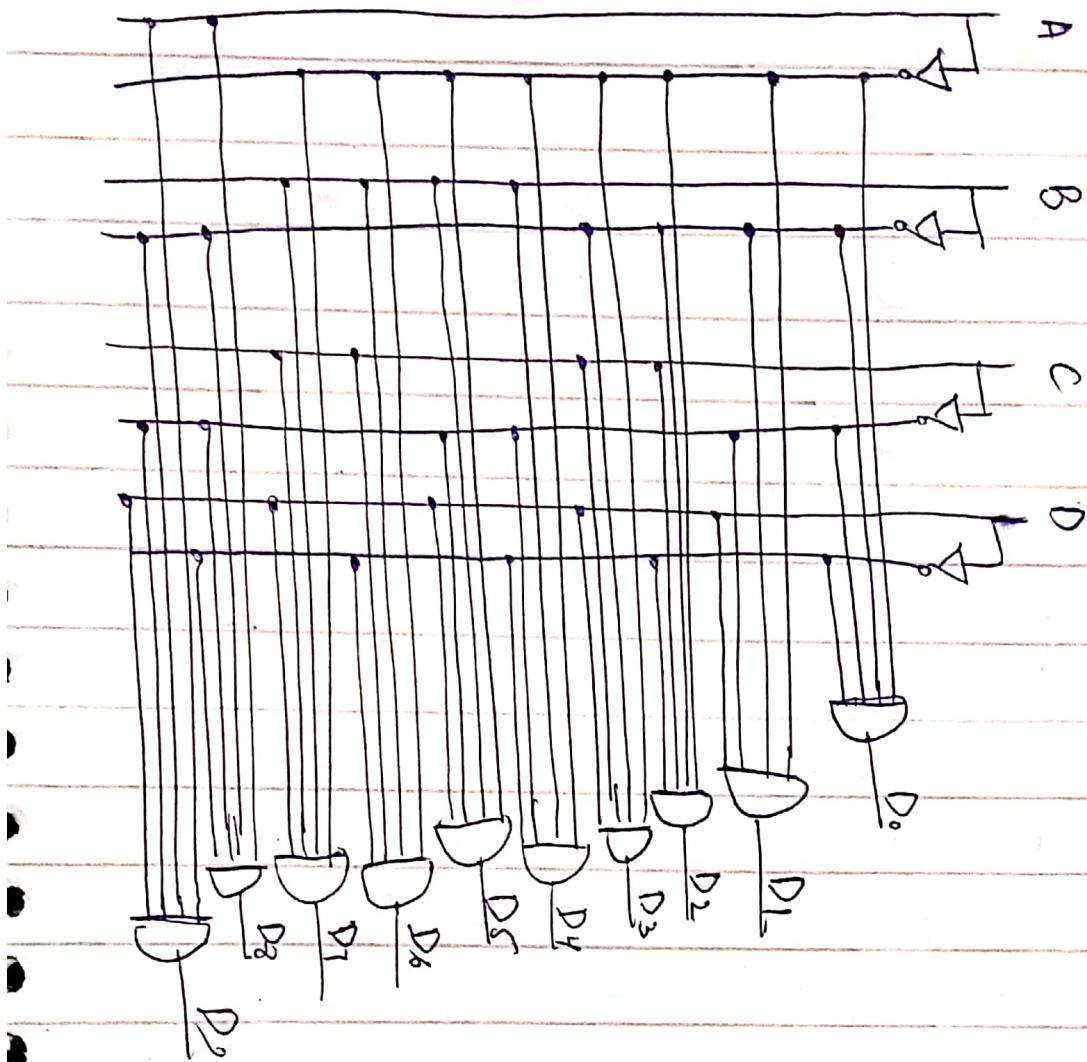
\* A B C D D<sub>0</sub> D<sub>1</sub> D<sub>2</sub> D<sub>3</sub> D<sub>4</sub> D<sub>5</sub> D<sub>6</sub> D<sub>7</sub> D<sub>8</sub> D<sub>9</sub> D<sub>10</sub> D<sub>11</sub> D<sub>12</sub> D<sub>13</sub> D<sub>14</sub> D<sub>15</sub>

A	B	C	D	D <sub>0</sub>	D <sub>1</sub>	D <sub>2</sub>	D <sub>3</sub>	D <sub>4</sub>	D <sub>5</sub>	D <sub>6</sub>	D <sub>7</sub>	D <sub>8</sub>	D <sub>9</sub>	D <sub>10</sub>	D <sub>11</sub>	D <sub>12</sub>	D <sub>13</sub>	D <sub>14</sub>	D <sub>15</sub>
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	1	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	1	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0
0	0	1	1	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0
0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0



## BCD to Decimal Decoder.

### Logic Diagram:-



### Logic expression :-

$$D_0 = \overline{A} \overline{B} \overline{C} \overline{D}$$

$$D_1 = \overline{A} \overline{B} \overline{C} D$$

$$D_2 = A \overline{B} \overline{C} \overline{D}$$

$$D_3 = A \overline{B} C D$$

$$D_4 = \overline{A} B \overline{C} \overline{D}$$

$$D_5 = A B \overline{C} \overline{D}$$

$$D_6 = \overline{A} B C \overline{D}$$

$$D_7 = A B C D$$

### $(3 \times 8)$ Decoder

Design full subtractor using decoder OR and external OR gates.

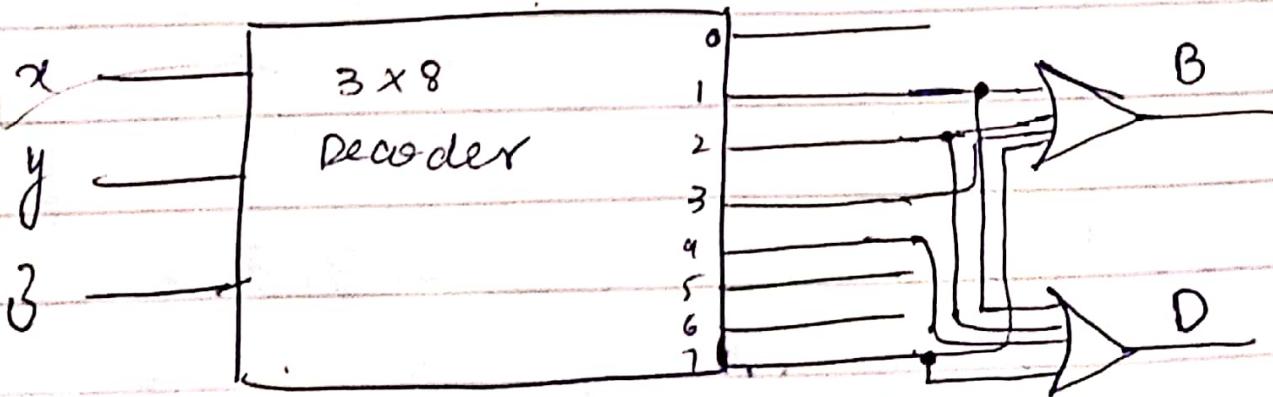
x	y	z	B	D
0	0	0	0	0
0	0	1	1	1
0	1	0	1	1
0	1	1	1	0
1	0	0	0	1
1	0	1	0	0
1	1	0	0	0
1	1	1	1	1

$$\text{E}(\overline{x}\overline{y}\overline{z}) \cup (\overline{x}yz) \cup (x\overline{y}z) \cup (xy\overline{z}) \in \{1, 2, 3, 7\}$$

$$\text{Borrow} = \overline{m_1 + m_2 + m_3 + m_7}$$

$$\text{Diff} = \overline{m_1 + m_2 + m_4 + m_7} \in \{1, 2, 4, 7\}$$

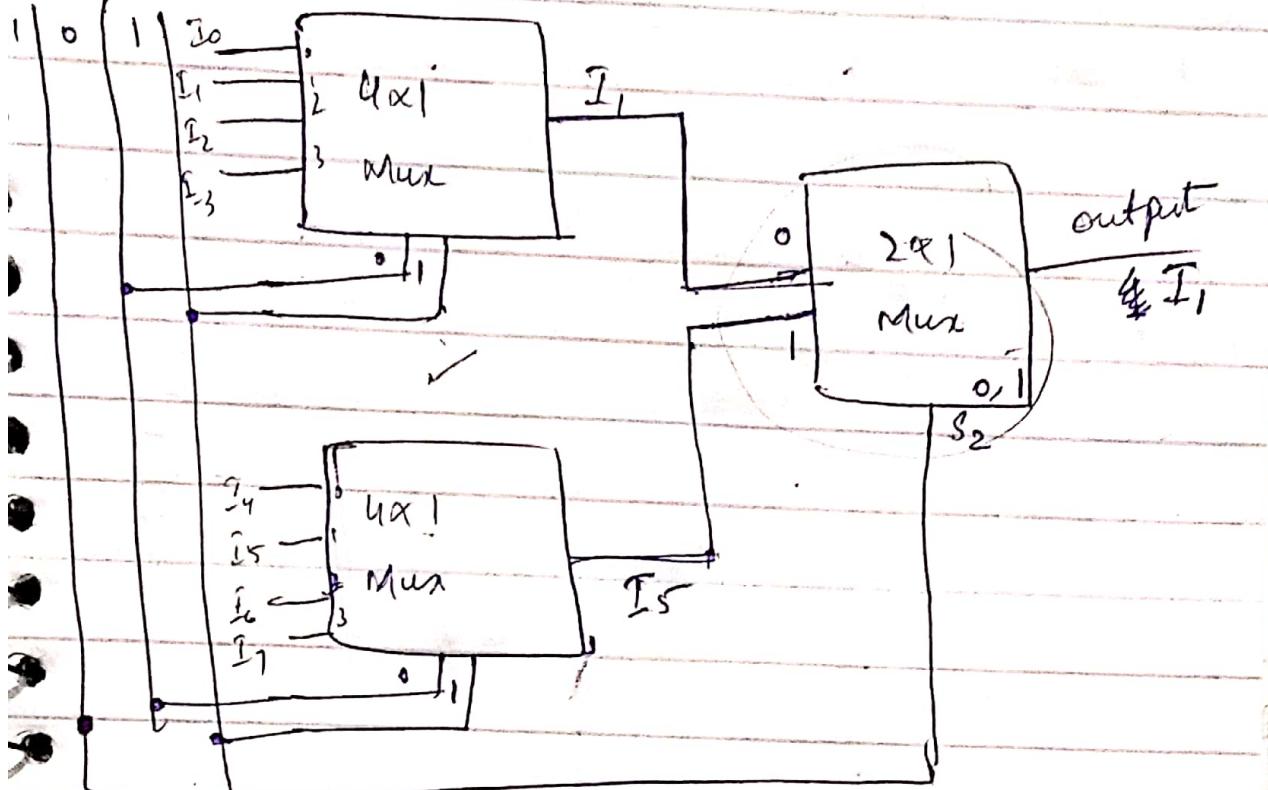
$\in \{E, F, G, H\}$



Design  $8 \times 1$  Mux using Two  $4 \times 1$  and one

$2 \times 1$  Mux.

$I_0$   
 $S_2 S_1 S_0$



$S_2 \quad S_1 \quad S_0 \quad \text{output}$

0 0 0  $I_0$

0 0 1  $I_1$

0 1 0  $I_2$

0 1 1  $I_3$

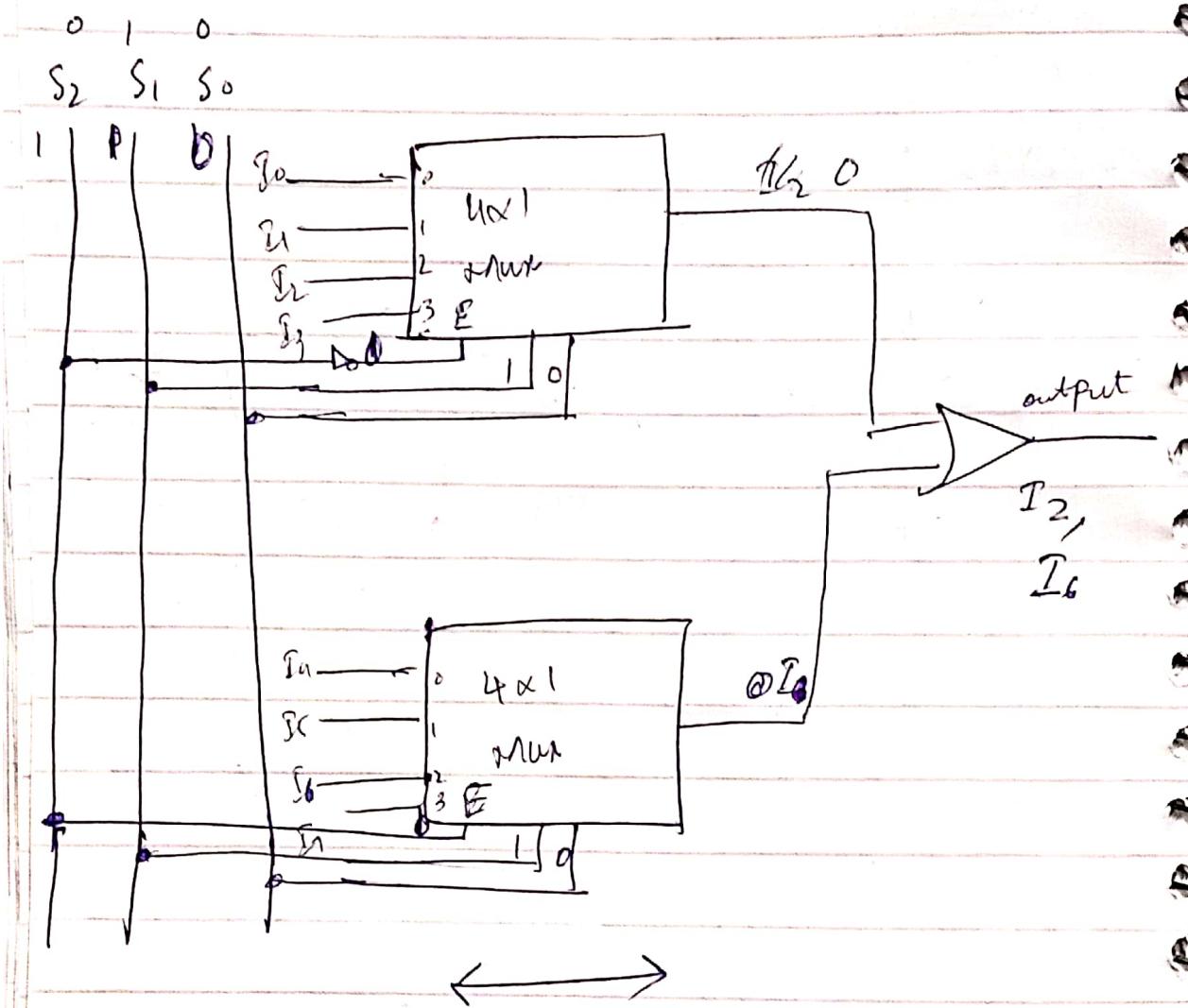
1 0 0  $I_4$

1 0 1  $I_5$

1 1 0  $I_6$

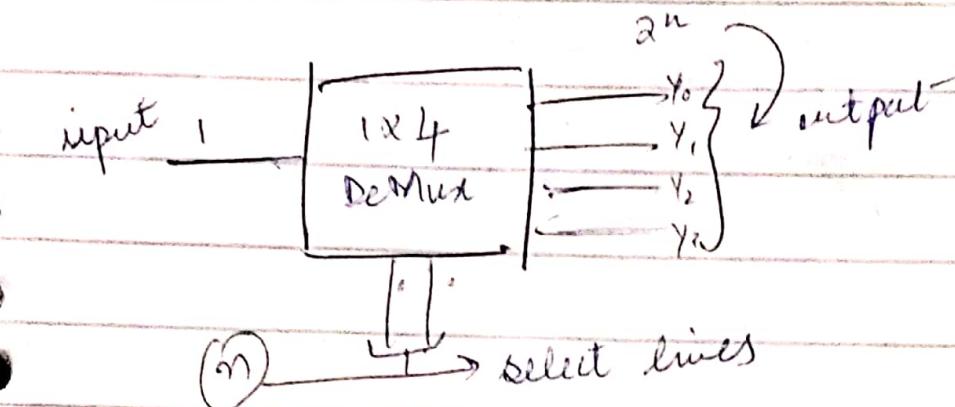
1 1 1  $I_7$

Design  $8 \times 1$  Mux using two  $4 \times 1$  Mux and one OR gate



For n variable function take n-1 as selection lines. leaving value determine input.

De Multiplexer :- opposite to Multiplexer



① Multiplexer  $\rightarrow$  Data Selector

② De Multiplexer  $\rightarrow$  Data distributor

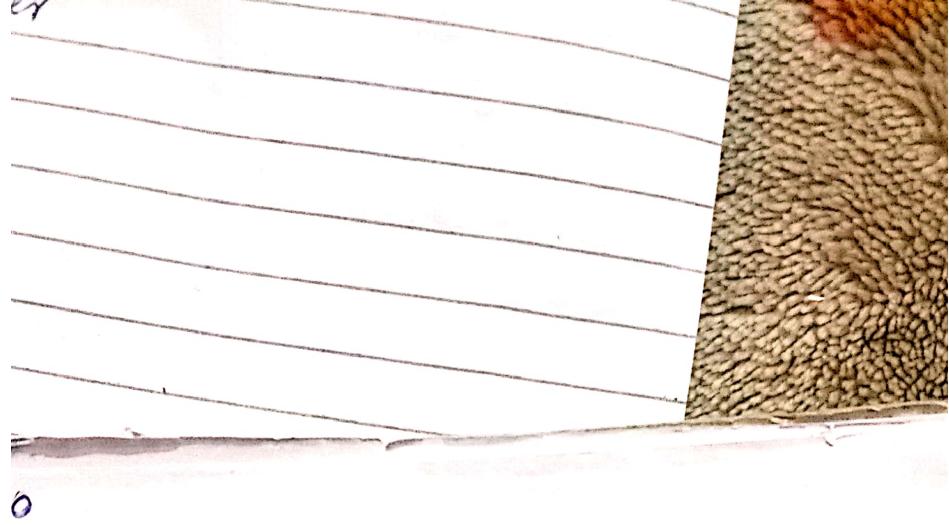
$S_1 S_0$	$I$	O/P
0 0	0	$y_0$
0 1	1	$y_1$
1 0	0	$y_2$
1 1	1	$y_3$

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

$$y_1 = \bar{S}_1 S_0 I$$

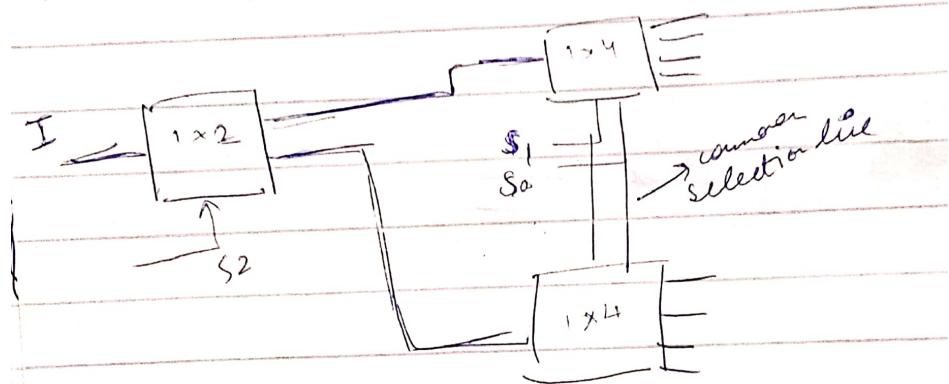
$$y_2 = S_1 \bar{S}_0 I$$

$$y_3 = S_1 S_0 I$$

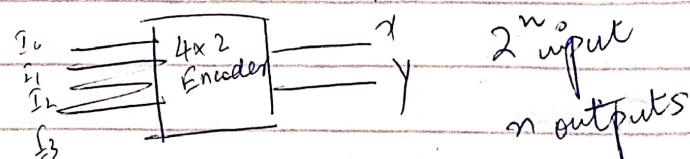
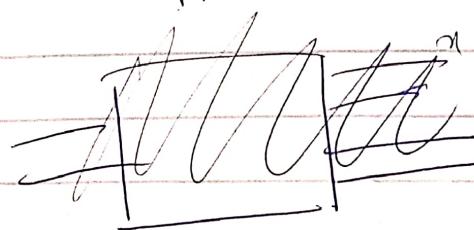


0

$1 \times 8$  using small Demux



Encoder:- opposite to decoder



decoder

MUX

MUX  $\frac{N}{4}$

## Flip Flops

Binary storage device

every F/F has two outputs.  $Q$  and  $\bar{Q}$

States :-

Has two states.

Set States:-

\*  $Q = 1$  and  $\bar{Q} = 0 \rightarrow$  Flip/Flop is ~~set~~ in set state.

\*  $Q = 0$  and  $\bar{Q} = 1 \rightarrow$  [reset state]

→ 1 Flip flop stores 1 bit

→ Registers :-

$\hookrightarrow n = 4$

range

$\hookrightarrow$  signed  
 $\hookrightarrow$  unsigned

range for unsigned :-  $0 \rightarrow 2^n - 1$

MSB is taken as signed bit.

negative numbers are stored in 2's complement

\* sign bit = 0 +ve

\* sign bit = 1 -ve

$$-\frac{2^n}{2} \rightarrow \frac{2^n - 1}{2}$$

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

for signed integer

DLD :-

### Flip Flop:- (NOR gate)

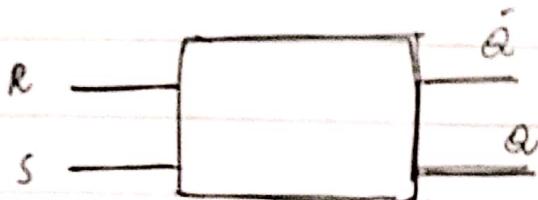
→ Binary storage device that store the one bit of data in it.

Two outputs  $Q$  and  $\bar{Q}$

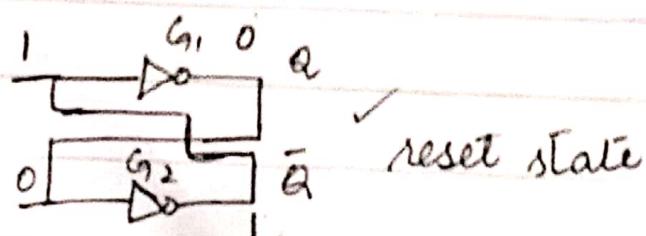
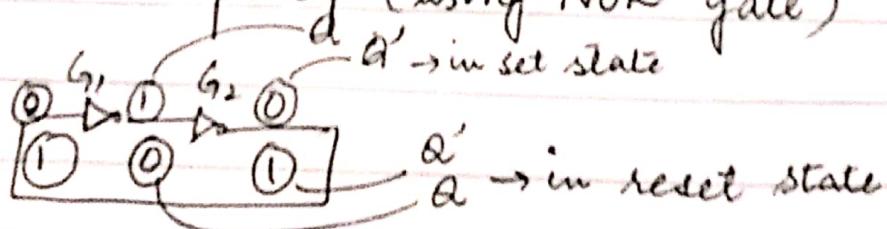
- $Q$  represents normal (output) value of bit.
- $\bar{Q}$  represents complement of that bit value.

→  $Q=1$  and  $\bar{Q}=0$  in set state

→  $Q=0$  and  $\bar{Q}=1$  in reset state



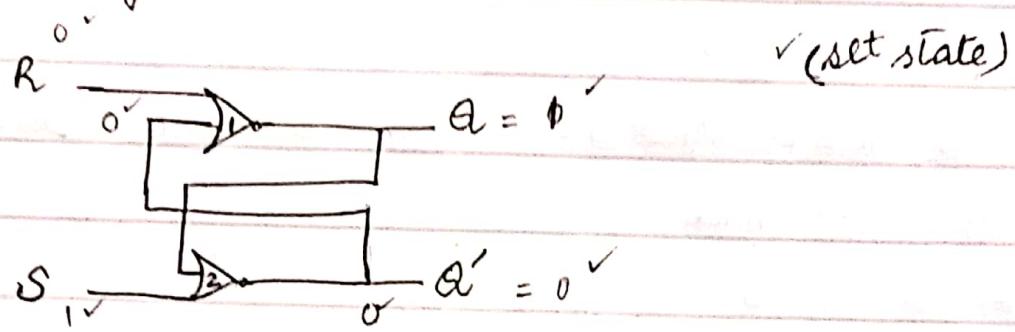
### Basic SR Flip-Flop (using NOR gate)



Two gate back to back connected

single input NOT gate is a NOR gate

To control circuit (set and reset state)  
then NOR gate is used.



NOR gate :-  $x \quad y \quad F$

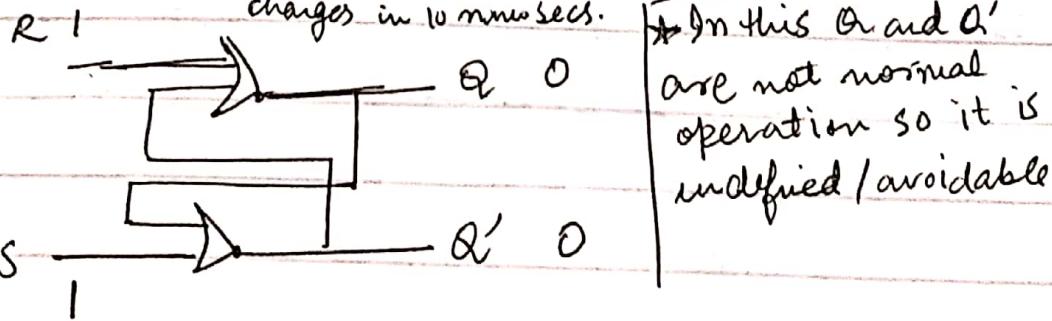
0 0 1

single one  
then O/P will  
be zero.

$\begin{cases} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 1 & 1 & 0 \end{cases}$

If  $R=1$  and  $S=1$  (If  $R=0$  and  $S=0$ ) then also  
because we cannot determine the state it  
is avoidable

changes in 10 nusecs.



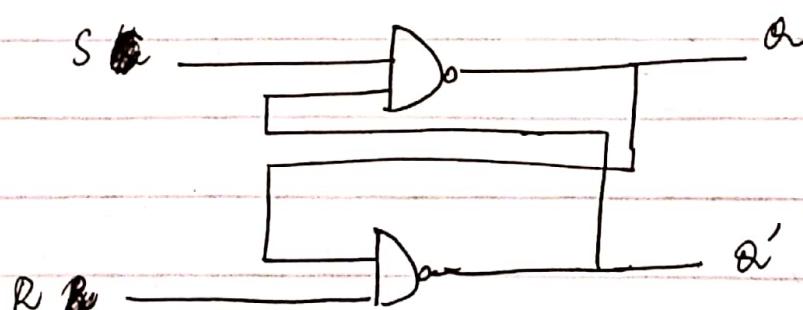
for normal operations of flip-flop both Q and Q' must be complement to each other.

- By applying  $R=0$  and  $S=0$  flip-flop does not change its state (set remains set and reset remains reset) No change
- If  $R=1$  and  $S=0$  then in Reset state in NOR gate.
- If  $R=0$  and  $S=1$  then in Set state in NOR gate.

~~1/1 of R & S~~

### Flip Flop (using NAND gate) :-

- single input NAND gate also act as NOT gate.

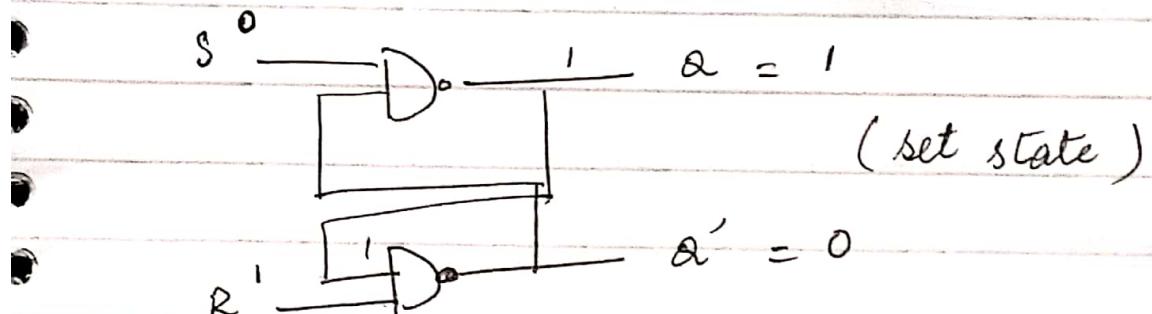


x	y	F
0	0	1
0	1	1
1	0	1
1	1	0

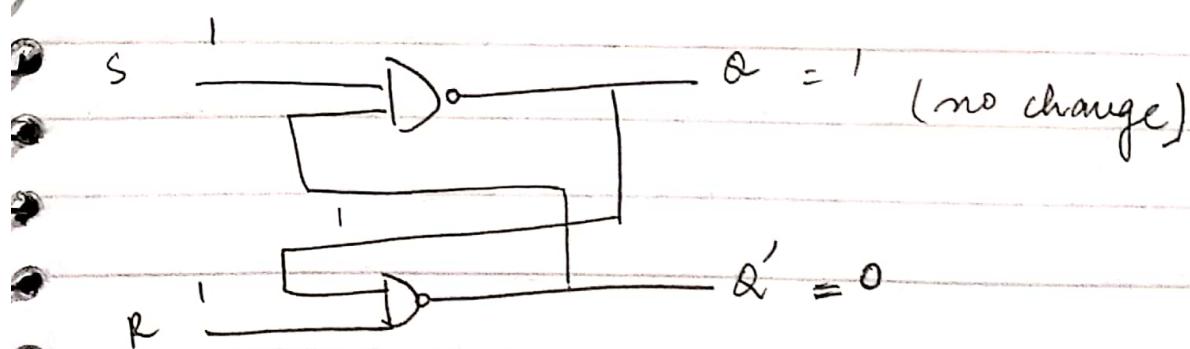
{ if one input 0 then O/P is 1 }

- \* In NOR gate upper input is R and lower is S.
- \* In NAND gate upper input is S and lower is R.

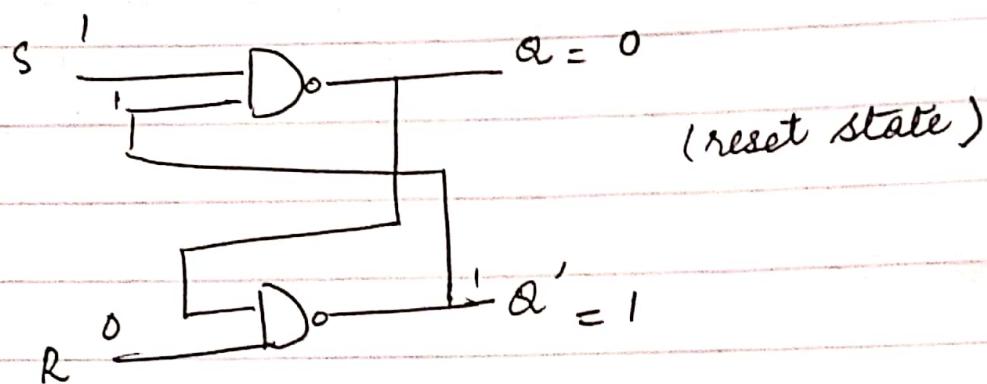
case 1 :-  $S = 0$        $R = 1$



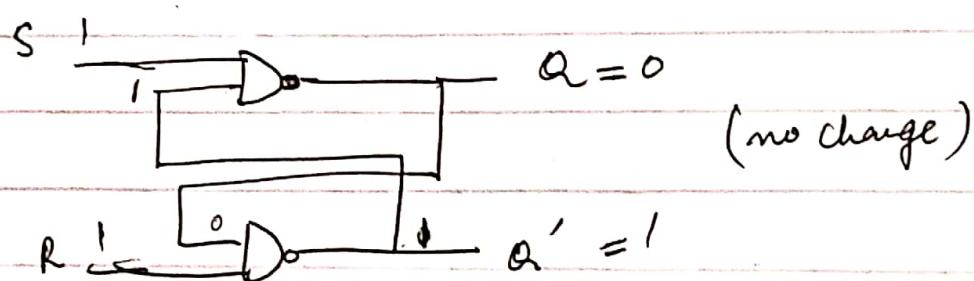
case 2 :-  $S = 1$  and  $R = 1$  (in set state)



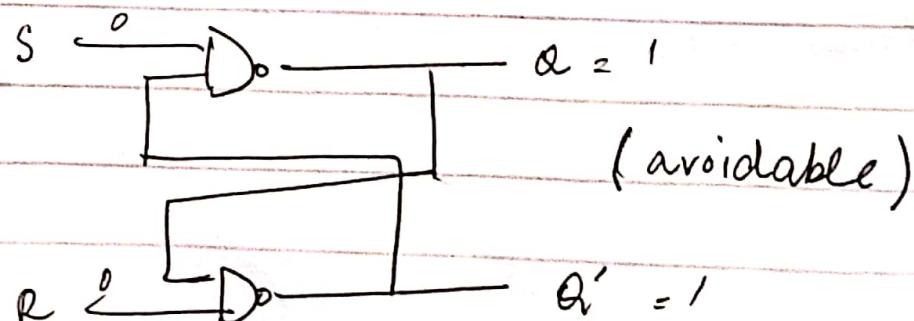
case # 3  $S = 1$  and  $R = 0$



case # 4  $S = 1$  and  $R = 1$  (in reset state)

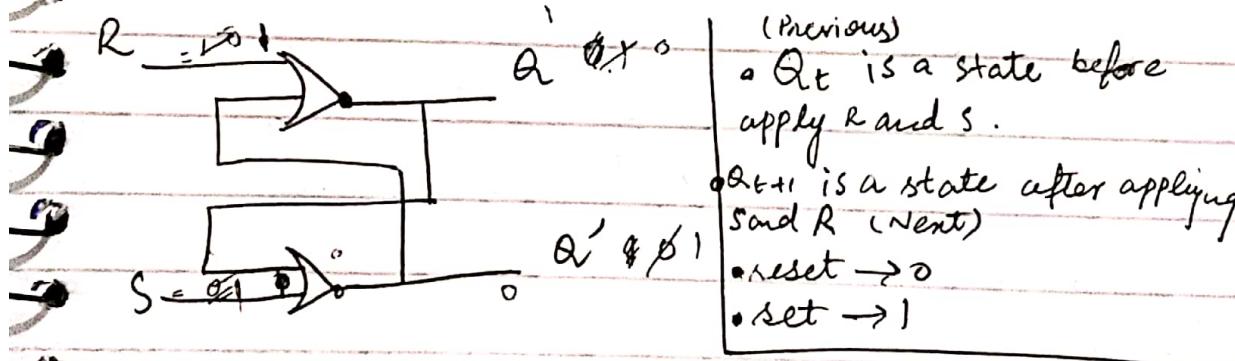


case # 5  $S = 0$  and  $R = 0$



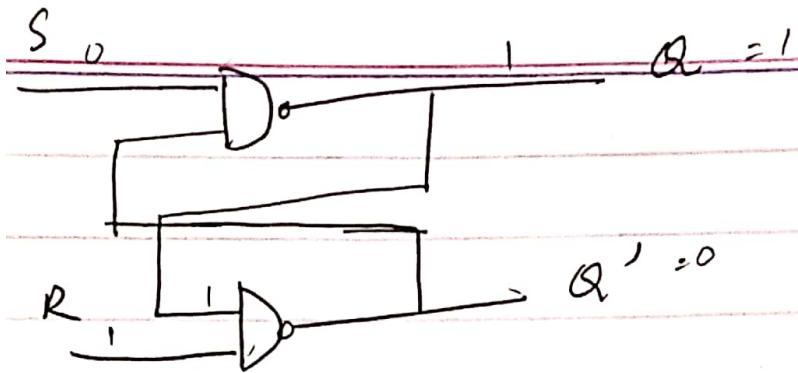
- If  $R=1$  and  $S=0$  (set state), if set and  $R=1$  and  $S=1$  then no change in it.
- If  $R=0$  and  $S=1$  (reset state), if reset and  $R=1$  and  $S=1$  then no change in it.
- If  $R=0$  and  $S=0$  using NAND gate then it is avoidable.

Comparison between Basic SR Flip/Flop using NOR gates and NAND gates.



$Q_t$	$S$	$R$	$Q_{t+1}$	State		
0	0	0	0	No change		
0 ✓	<table border="1" style="margin-left: auto; margin-right: auto;"><tr><td>0</td><td>1</td></tr></table>		0	1	0	reset
0	1					
0 ✓	<table border="1" style="margin-left: auto; margin-right: auto;"><tr><td>1</td><td>0</td></tr></table>		1	0	1	set
1	0					
0	1	1	X	unspecified		
1	0	0	1	No change		
1 ✓	<table border="1" style="margin-left: auto; margin-right: auto;"><tr><td>0</td><td>1</td></tr></table>		0	1	0	reset
0	1					
1	1	0	1	reset $\rightarrow 0$		
1	1	1	X	Set $\rightarrow 1$		

## NAND



$Q_t$	<del>S</del>	R	$Q_{t+1}$	State
0	0	0	X	prohibited
0	0	1	1	set
0	1	0	0	reset
0	1	1	0	No change
1	0	0	X	prohibited
1	0	1	1	set
1	1	0	0	reset
1	1	1	1	No change

Summary:-

NOR gate :-

$S=1$  and  $R=0$  (set)

$S=0$  and  $R=1$  (reset)

$S=0$  and  $R=0$  (no change)

$S=1$  and  $R=1$  (prohibited)

NAND gate :-

$S=0$  and  $R=1$  (set)

$S=1$  and  $R=0$  (reset)

$S=1$  and  $R=1$  (no change)

$S=0$  and  $R=0$  (prohibited)

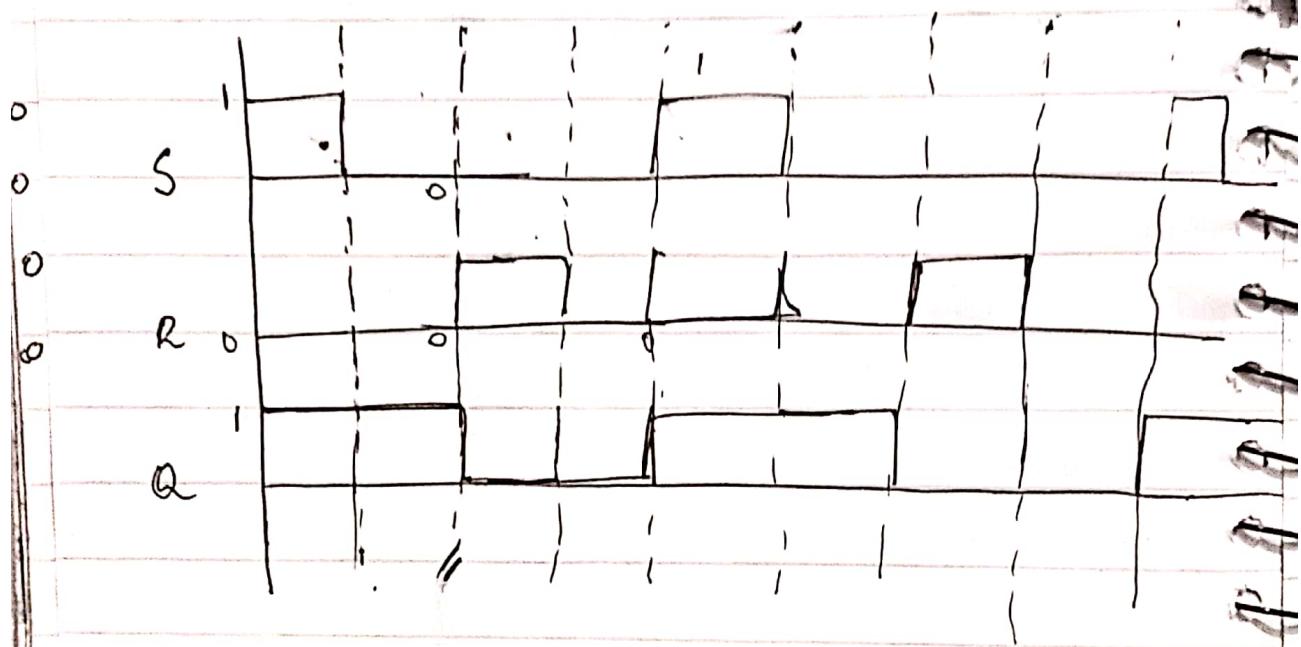
$$\begin{array}{l} S=1 \\ R=0 \end{array} \rightarrow Q=1$$

$$\begin{array}{l} S=0 \\ R=1 \end{array} \rightarrow Q=0$$

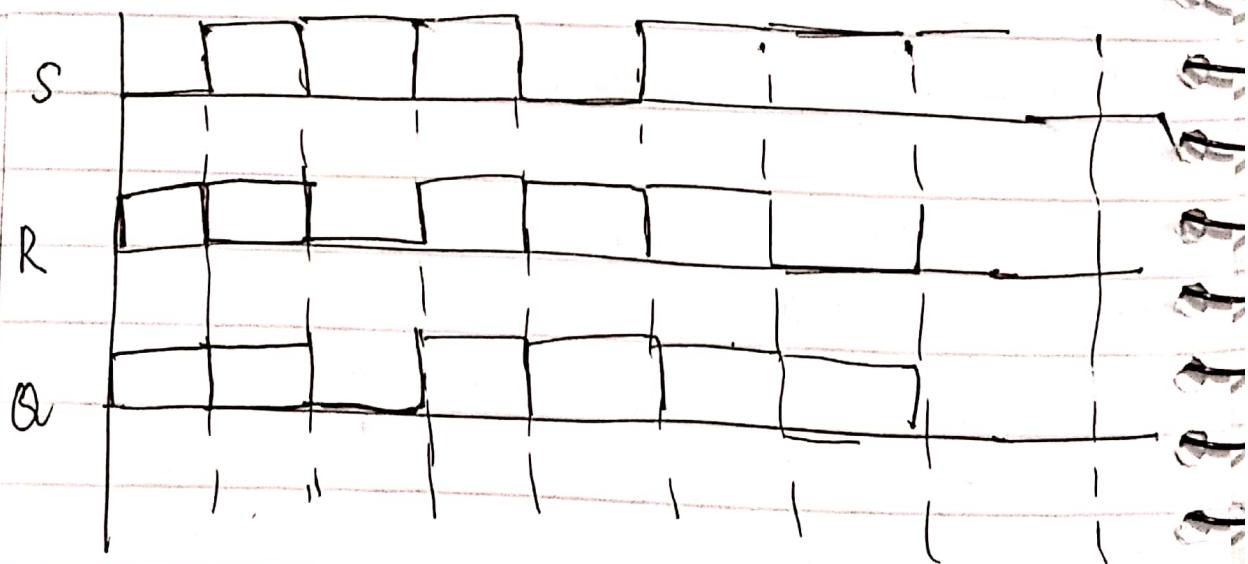
DLD:-

SR Latch Timing Diagram :-

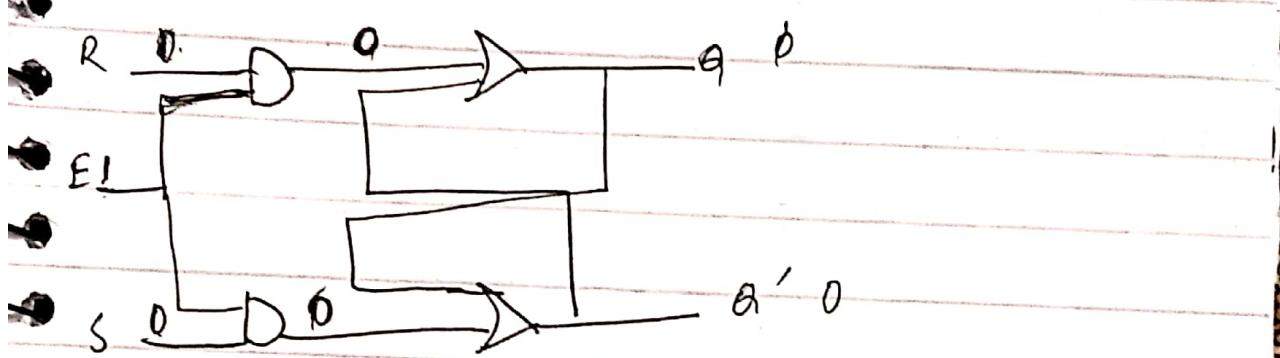
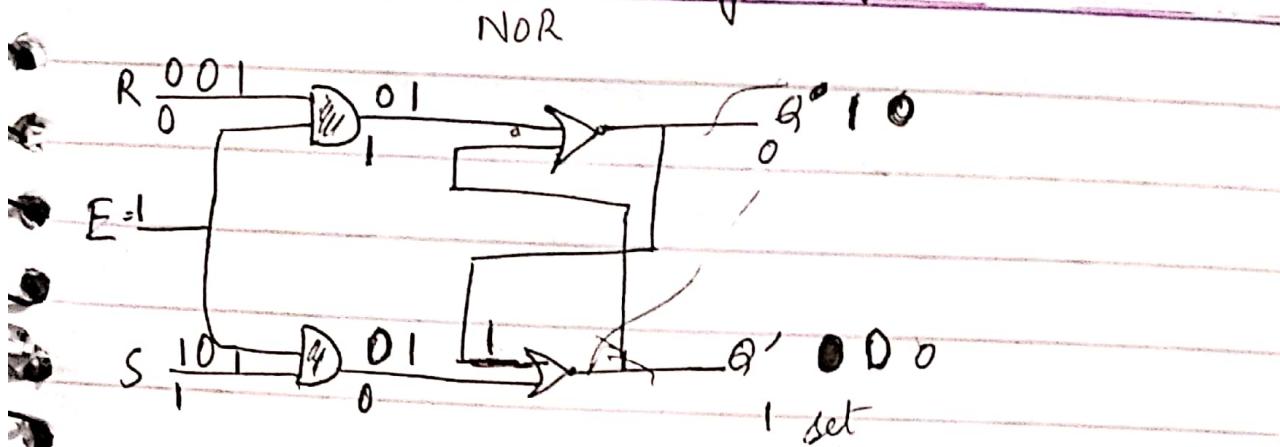
SR Latch using NOR gates -



SR Latch using NAND gates

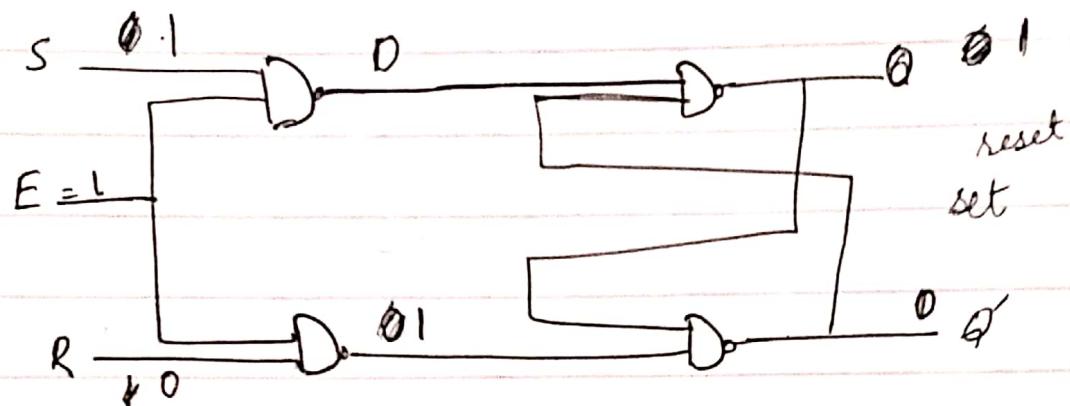


## Gated SR Latch - Timing Diagram:-



$Q_t$	S	R	$Q_{t+1}$	State
0	0	0	Q	
0	0	1	Q	
0	1	0	Q	
0	1	1	Q	
1	0	0	Q	
1	0	1	Q	
1	1	0	1	
1	1	1	Avoidable	

### NAND



Gated SR Latch summary:-

$$S = 1 \quad R = 0 \text{ (set)}$$

$$S = 0 \quad R = 1 \text{ (reset)}$$

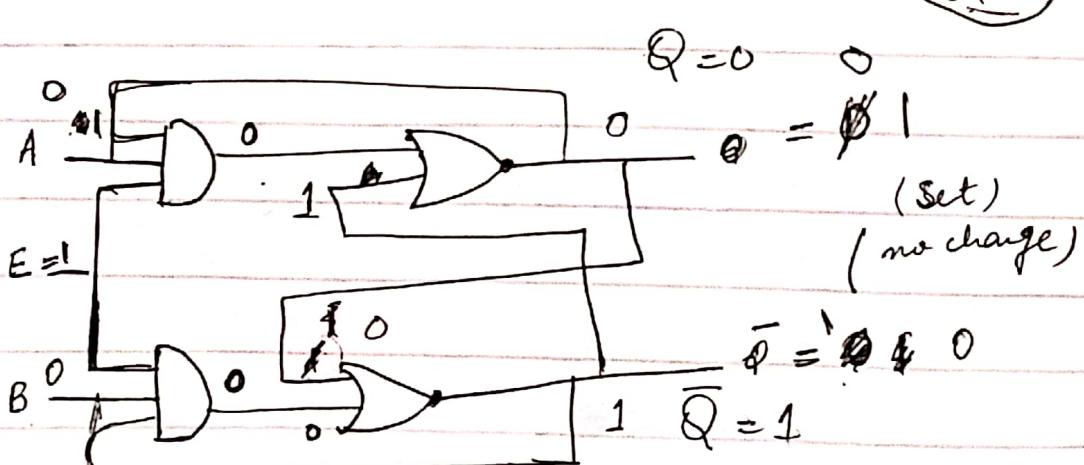
$$S = 0 \quad R = 0 \text{ (No change)}$$

$$S = 1 \quad R = 1 \text{ (avoidable)}$$

Gated SR Latch

$D \ 0 \ 0$

$1 \ 0 \ 1$   
 $\underline{\circ} \ 1 \ 0$



# DLD

- Implementation using NAND and NOR gates.
- Adders
- Subtractors
- Comparators
- Encoders
- Decoders
- Multiplexer
- De Multiplexer
- Flip Flop
- Registers

## Adders :-

### Half Adder:-

X	Y	C	S
0	0	0	0
0	1	0	1
1	0	0	1
1	1	1	0

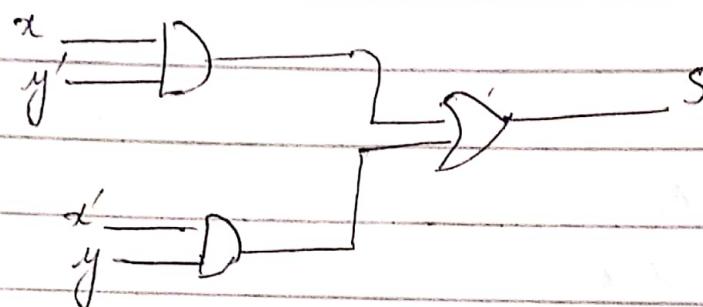
$$C = xy$$

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

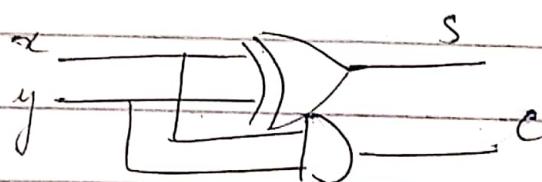
$C =$



$S =$



Sum & Carry :-



$$S = x \oplus y$$

$$C = xy$$

Full Adder:-

$x$	$y$	$S$	$C$	$S$
0	0	0	0	0
0	0	1	0	1
0	1	0	0	1
0	1	1	1	0
1	0	0	0	1
1	0	1	1	0
1	1	0	1	0
1	1	1	1	1

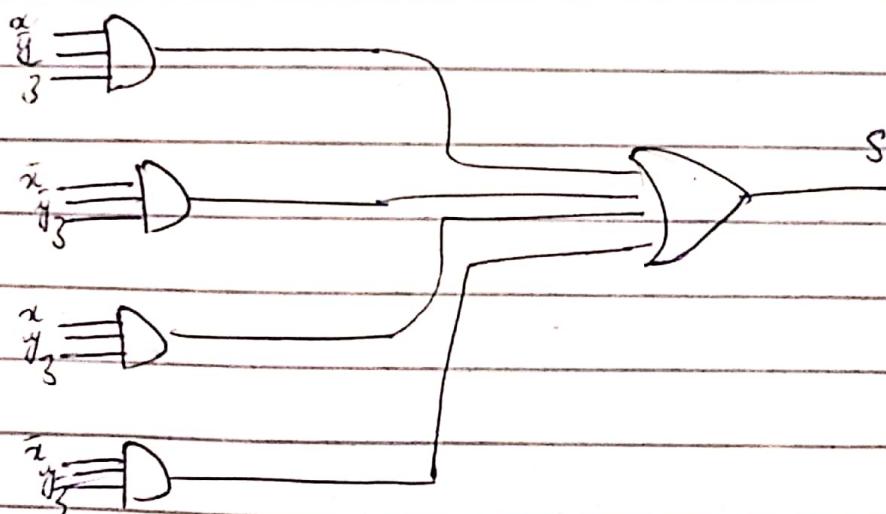
$$S = \begin{array}{c} \bar{y}_3 \quad \bar{y}_3 \quad y_3 \quad y_3 \\ \hline \bar{x} \quad | \quad . \quad | \quad 1, \quad | \quad 3 \quad | \quad 1, \\ x \quad | \quad 1, \quad | \quad 5 \quad | \quad 1, \quad | \quad 6 \end{array}$$

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

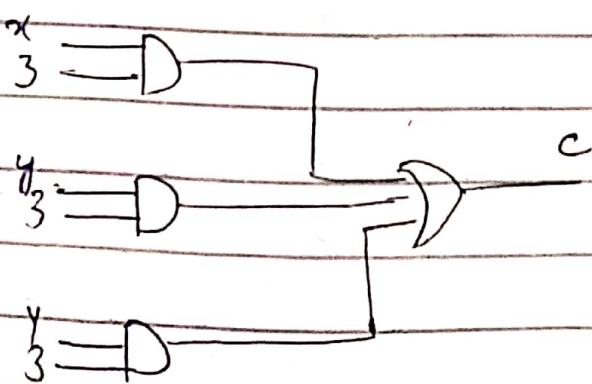
$$C = \begin{array}{c} \bar{y}_3 \quad \bar{y}_3 \quad y_3 \quad y_3 \\ \hline \bar{x} \quad | \quad . \quad | \quad 1, \quad | \quad 1, \quad | \quad 2 \\ x \quad | \quad 1, \quad | \quad 1, \quad | \quad 1, \quad | \quad 1, \quad | \quad 6 \end{array}$$

$$= xz + xy + \cancel{y}z$$

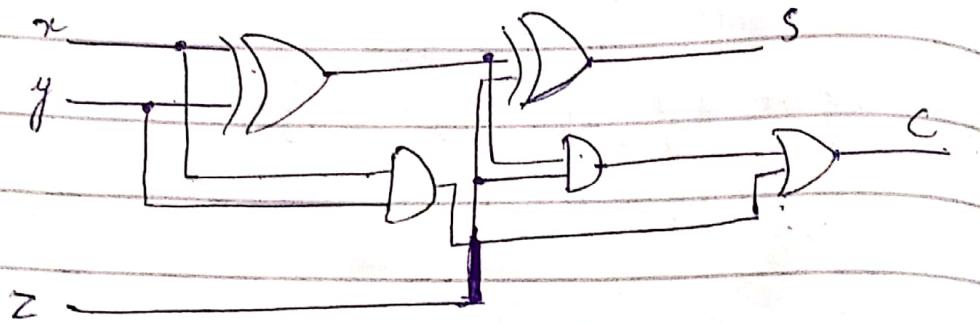
$$S =$$



$$C =$$



Sum & Carry:-



Implementation of full adder using two half adder and an OR gate.

$$S = z \oplus (x \oplus y)$$

$$C = z(xy' + x'y) + xy$$

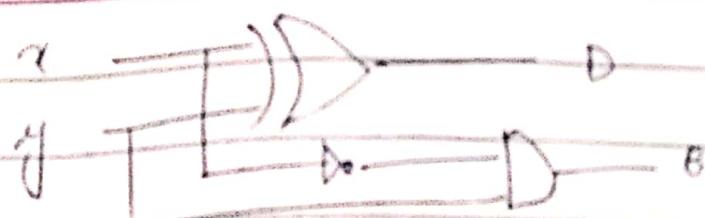
Subtractors:-

Half Subtractor:-

x	y	B	D
0	0	0	0
0	1	1	1
1	0	0	1
1	1	0	0

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

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



## Full Subtractor:-

A	B	C/Bin.	D	B.
0	0	0	0	0
0	0	1	1	1
0	1	0	01	1
0	1	1	0	1
1	0	0	1	0
1	0	1	0	0
1	1	0	0	0
1	1	1	1	1

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

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

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

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

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

7485 IC

## Magnitude Comparators:-

1-bit comparators

Truth Table :-

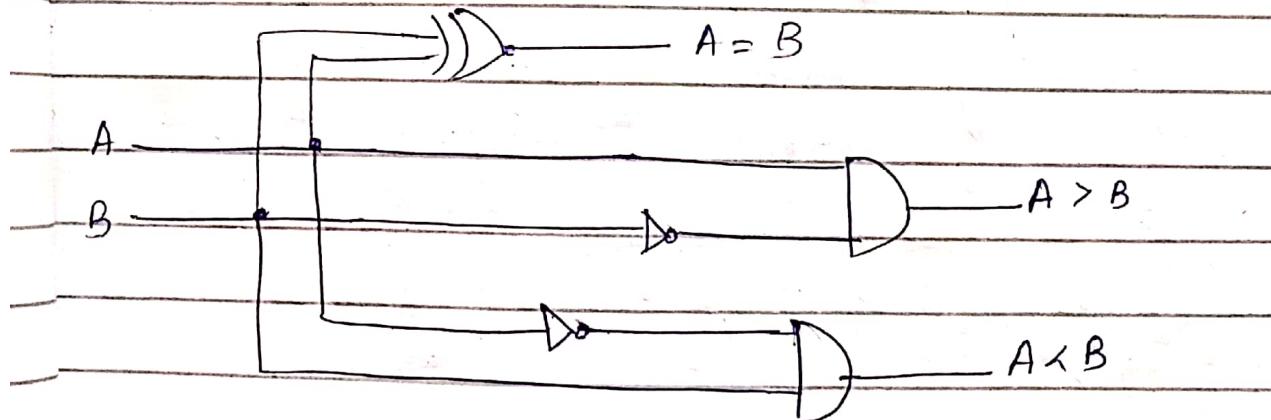
A	B	$A = B$	$A > B$	$A < B$
0	0	1	0	0
0	1	0	0	1
1	0	0	1	0
1	1	1	0	0

$$A = B = A \odot B$$

①  $\rightarrow$  XNOR.

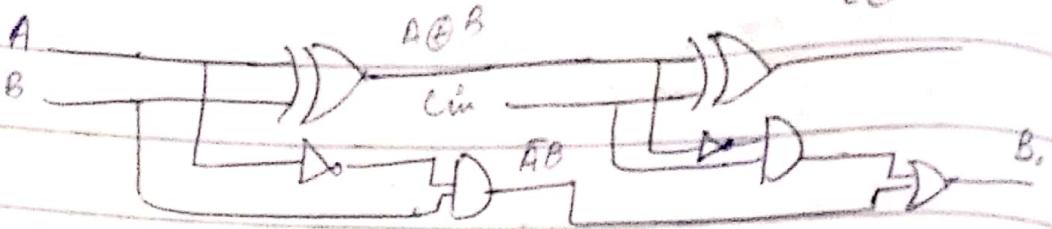
$$A > B = \bar{A} \bar{B}$$

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



$\angle \rightarrow \geq$

$> \rightarrow \leq$



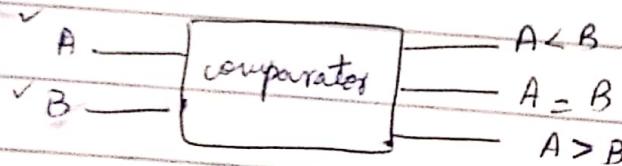
Comparator:- 2-bit comparator

[For 2 bit comparator we have 4 inputs.]

A digital comparator is a combinational circuit designed to compare two n-bit binary. [Each number consist of two bits].

A comparator has three outputs.

more than single bit



for  $A = B$  :-

$\bar{A}, A_0$	$B_1, B_0$	$\bar{B}_1, \bar{B}_0$	$B_1, B_0$	$\bar{B}_1, \bar{B}_0$
1	0	1	0	1
0	1	0	1	0
1	1	1	0	0
0	0	0	0	0

for  $A > B$

$\bar{B}_1, \bar{B}_0$	$B_1, B_0$	$\bar{B}_1, \bar{B}_0$	$B_1, B_0$	$\bar{B}_1, \bar{B}_0$
0	0	1	1	0
0	1	0	0	1
1	1	0	0	0
1	0	1	0	1

$$A = B = (A_1 \oplus B_1) (A_0 \oplus B_0)$$

$$= \bar{A}_1 \bar{A}_0 \bar{B}_1 \bar{B}_0 + \bar{A}_1 A_0 \bar{B}_1 B_0 + A_1 A_0 B_1 B_0 \\ + A_1 \bar{A}_0 B_1 \bar{B}_0$$

$$= A_0 \bar{B}_0 + A_1 \bar{B}_1 \bar{B}_0 + A_2 A_1 \bar{B}_0 \\ = A_1 \bar{B}_1 + A_1 A_0 \bar{B}_0 + A_0 \bar{B}_1 \bar{B}_0 \\ = (A_1 \bar{B}_1) + (A_1 \oplus B_1) (A_0 \bar{B}_0)$$

<u>2<sup>nd</sup></u>	<u>1<sup>st</sup></u>	$A_1 \ A_0 \ B_1 \ B_0$	$A < B$	$A = B$	$A > B$
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	0
0	1	1 0	1	0	0
0	1	1 1	1	0	0
1	0	0 0	0	0	1
1	0	0 1	0	0	1
1	0	1 0	0	1	0
1	0	1 1	1	0	0
1	1	0 0	0	0	1
1	1	0 1	0	0	1
1	1	1 0	0	0	1
1	1	1 1	0	1	0

$\bar{B}_1 \ \bar{B}_0$	$\bar{A}_1 \ \bar{A}_0$	$\bar{B}_1 \ B_0$	$\bar{B}_1 \ B_0$	$B_1 \ B_0$	$B_1 \ \bar{B}_0$
1	0 0	1 1	1 1	1 1	1 0
1	0 1	1 0	1 0	1 1	1 1
1	1 1	0 1	0 1	0 0	0 1
1	1 0	1 1	0 1	0 0	1 0
1	0	0	1	1	0

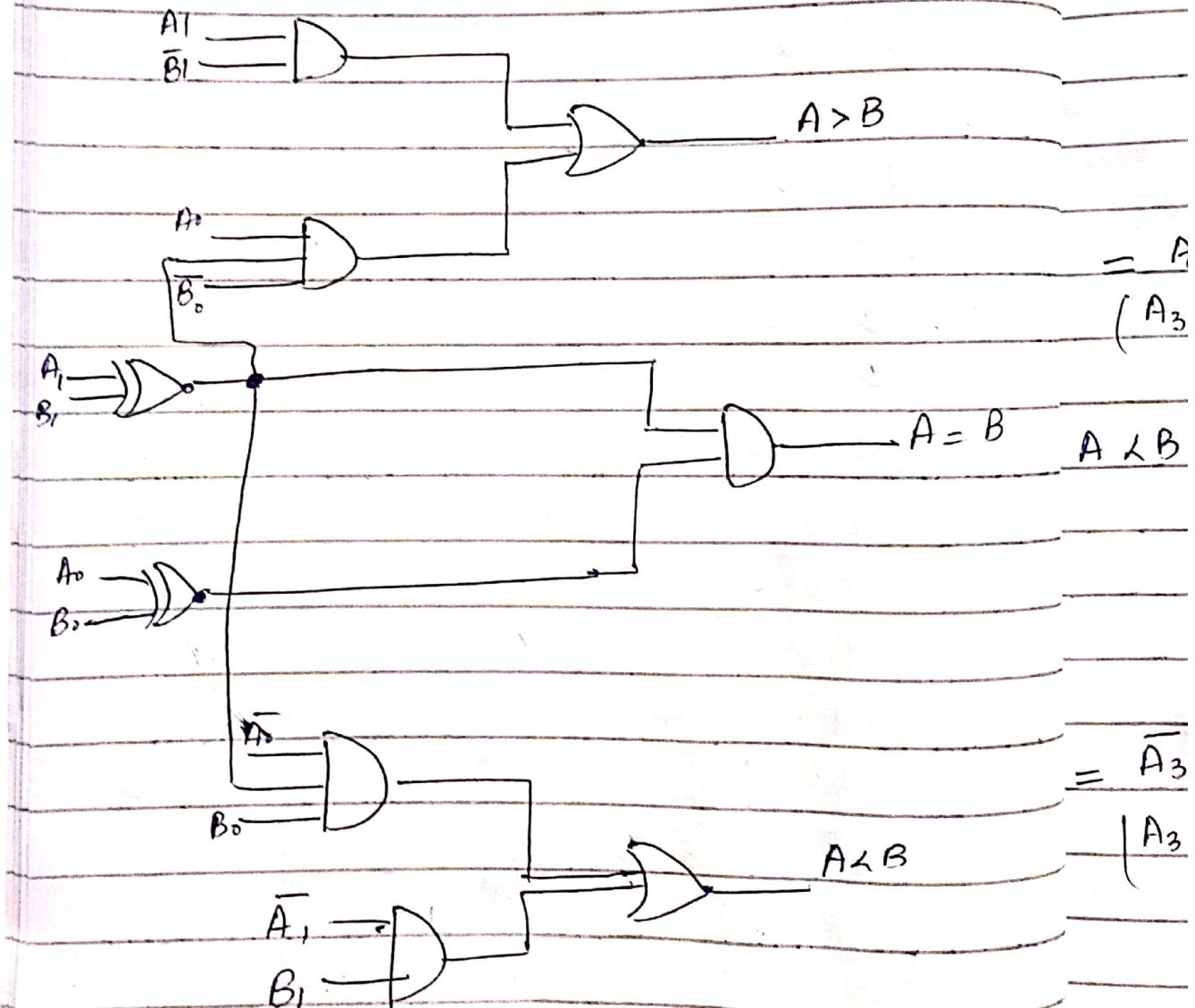
for  $A < B$ :

$$= \bar{A}_1 B_1 + \bar{A}_1 \bar{A}_0 B_0 + \bar{A}_0 B_1 B_0$$

$$= \bar{A}_1 B_1 + (A_1 \odot B_1) (B_0 \bar{A}_0)$$

- 4-1
- 1-bit comparator  $\rightarrow$  2 variables  $\rightarrow$  4 rows
- 2-bit comparator  $\rightarrow$  4 variables  $\rightarrow$  16 rows
- $n$ -bit comparator  $\rightarrow$   $2n$  variables  $\rightarrow$   $2^n$  rows
- $A = A$
- $A \neq B$

Circuits -



## 4-bit comparator

es

es

es

$$A = A_3 A_2 A_1 A_0$$

$$B = B_3 B_2 B_1 B_0$$

$$A = B = \overline{A_3 \oplus B_3} (A_3 \odot B_3) (A_2 \odot B_2) (A_1 \odot B_1) \\ (A_0 \odot B_0)$$

starting  $\rightarrow$  MSB

$$A > B \rightarrow A_3 > B_3 \text{ or}$$

$$A_3 = B_3 \text{ and } A_2 > B_2 \text{ or}$$

$$A_3 = B_3 \text{ and } A_2 = B_2 \text{ and } A_1 > B_1 \text{ or}$$

$$A_3 = B_3 \text{ and } A_2 = B_2 \text{ and } A_1 = B_1 \text{ or } A_0 > B_0 \text{ and}$$

$$= A_3 \bar{B}_3 + (A_3 \odot B_3) A_2 \bar{B}_2 + (A_3 \odot B_3) (A_2 \odot B_2) A_1 \bar{B}_1 + \\ (A_3 \odot B_3) (A_2 \odot B_2) (A_1 \odot B_1) A_0 \bar{B}_0$$

3

$$A < B \rightarrow A_3 < B_3 \text{ or}$$

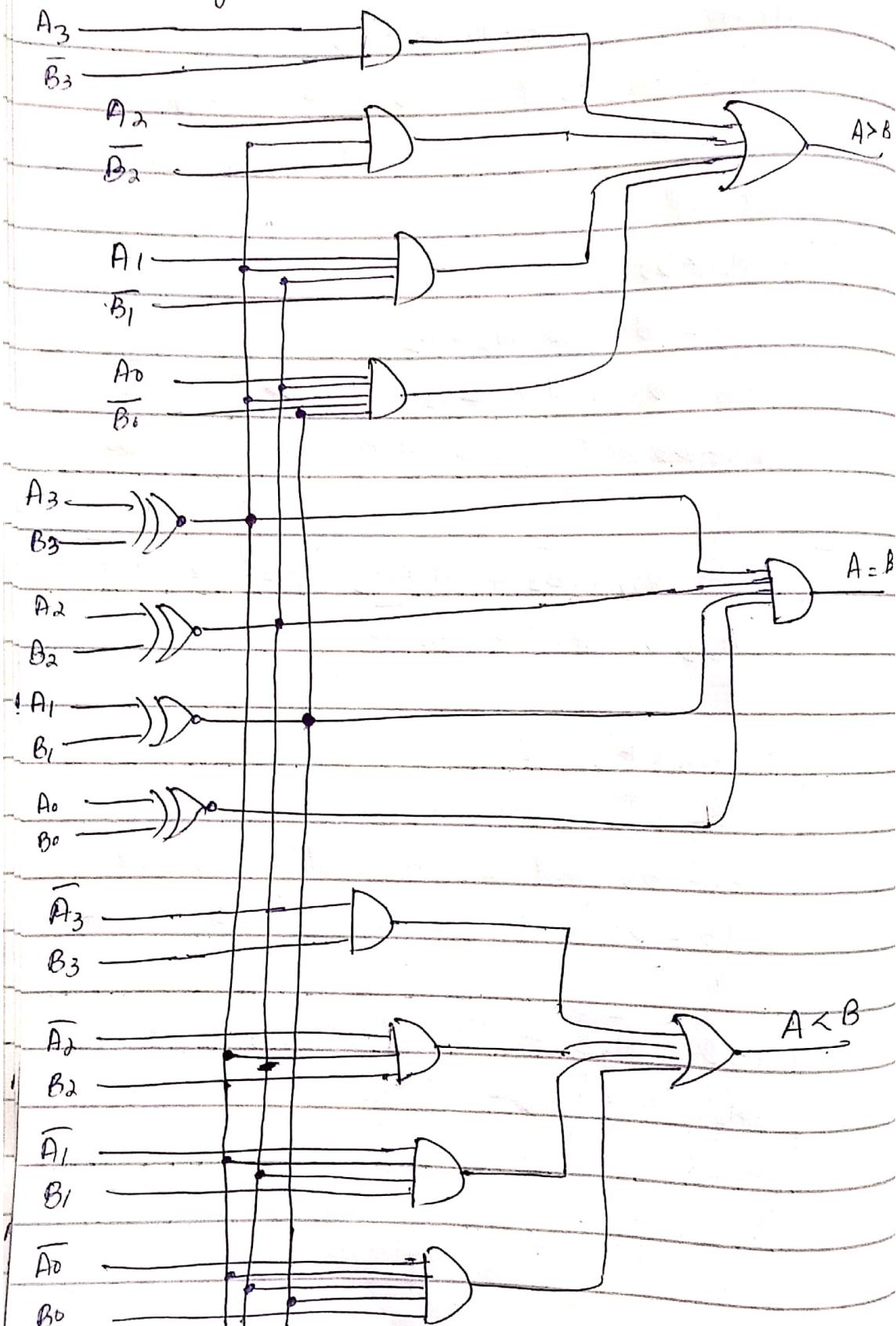
$$A_3 = B_3 \text{ and } A_2 < B_2 \text{ or}$$

$$A_3 = B_3 \text{ and } A_2 = B_2 \text{ and } A_1 < B_1 \text{ or}$$

$$A_3 = B_3 \text{ and } A_2 = B_2 \text{ and } A_1 = B_1 \text{ or } A_0 < B_0$$

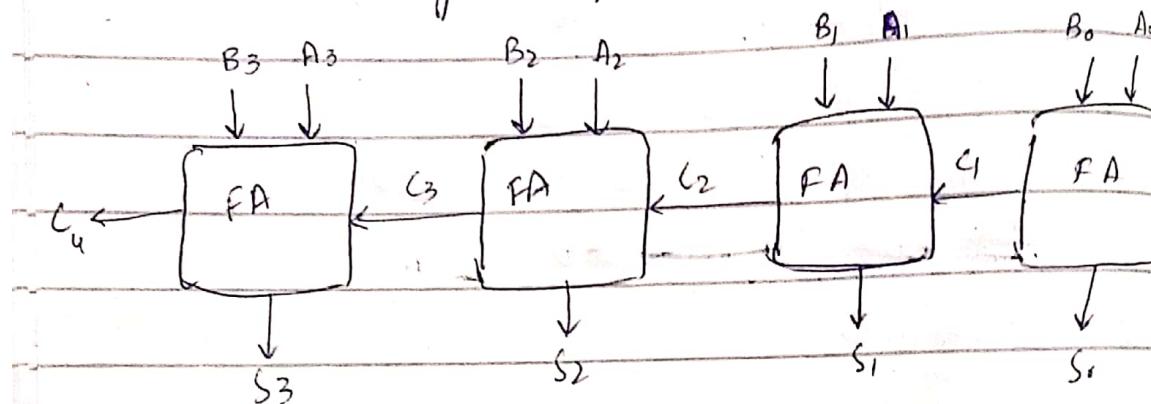
$$= \bar{A}_3 B_3 + (A_3 \odot B_3) \bar{A}_2 B_2 + (A_3 \odot B_3) (A_2 \odot B_2) \bar{A}_1 B_1 + \\ (A_3 \odot B_3) (A_2 \odot B_2) (A_1 \odot B_1) A_0 B_0$$

## Circuit diagrams -

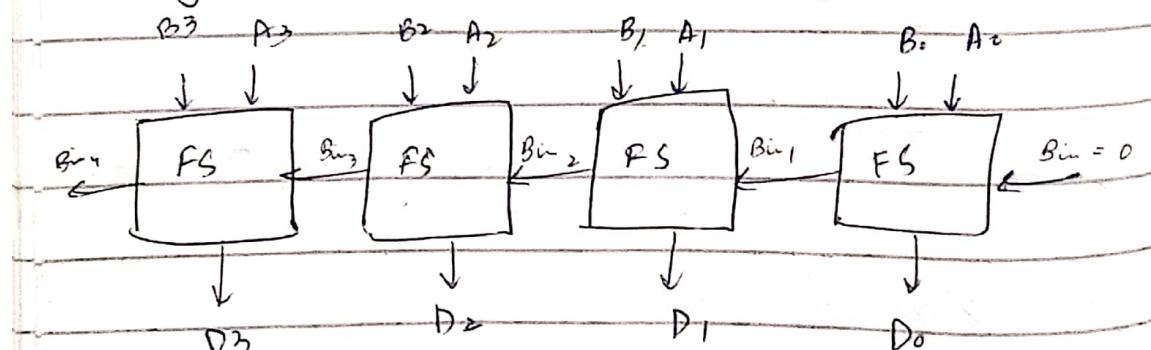


## 4-bit Adder:-

By cascading 4 such full adder blocks, we can easily design 4-bit adder.

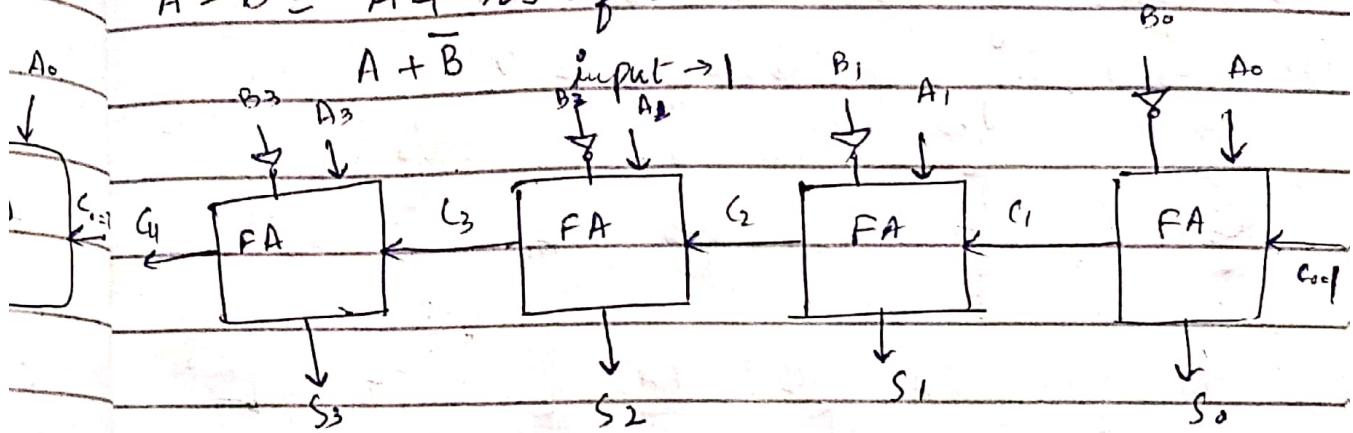


By cascading in such full subtractor we can easily design n-bit subtractor.



How to perform subtraction using adder circuit.

$$A - B = A + \text{1's complement of } B$$



$C_0 = 1 \rightarrow$  adding 1 to 1's complement of No.

Can

$$\begin{array}{r}
 1100 \\
 - 1 \\
 \hline
 0111
 \end{array}
 \rightarrow
 \begin{array}{r}
 1100 \\
 - 1001 \\
 \hline
 0101
 \end{array}
 \quad \text{Ignore end carry}$$

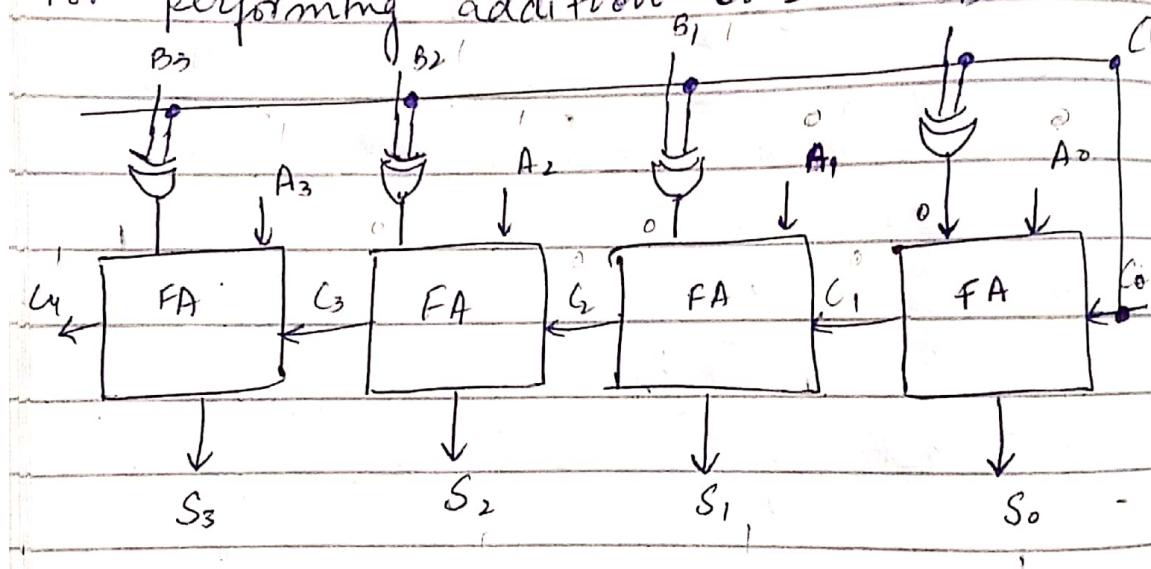
$$\begin{array}{r}
 1100 \\
 + 1001 \\
 \hline
 0101
 \end{array}$$

Ignore end carry

Performing addition and subtraction using  
Some circuits

For performing subtraction  $C_0 = 0$

For performing addition  $C_0 = 0$



e.g. - for addition

$$\begin{array}{r} 12 \rightarrow 01100 \\ + 7 \rightarrow 0111 \\ \hline 19 \end{array}$$

• XOR same then 0  
diff 1 output.

for subtraction :-

$$\begin{array}{r} 12 \quad 01100 \\ - 7 \quad 0111 \xrightarrow{\text{B/S}} 11.00 \\ \hline 5 \quad 0101 \end{array}$$

ignore

ing

when  $A \times B =$

$$10 \rightarrow 1010$$

$$-12 \rightarrow 1100$$

$$1010$$

$$0100$$

$$1110$$

$$0001$$

CTR

$$0001$$

$$1001$$

$$1001$$

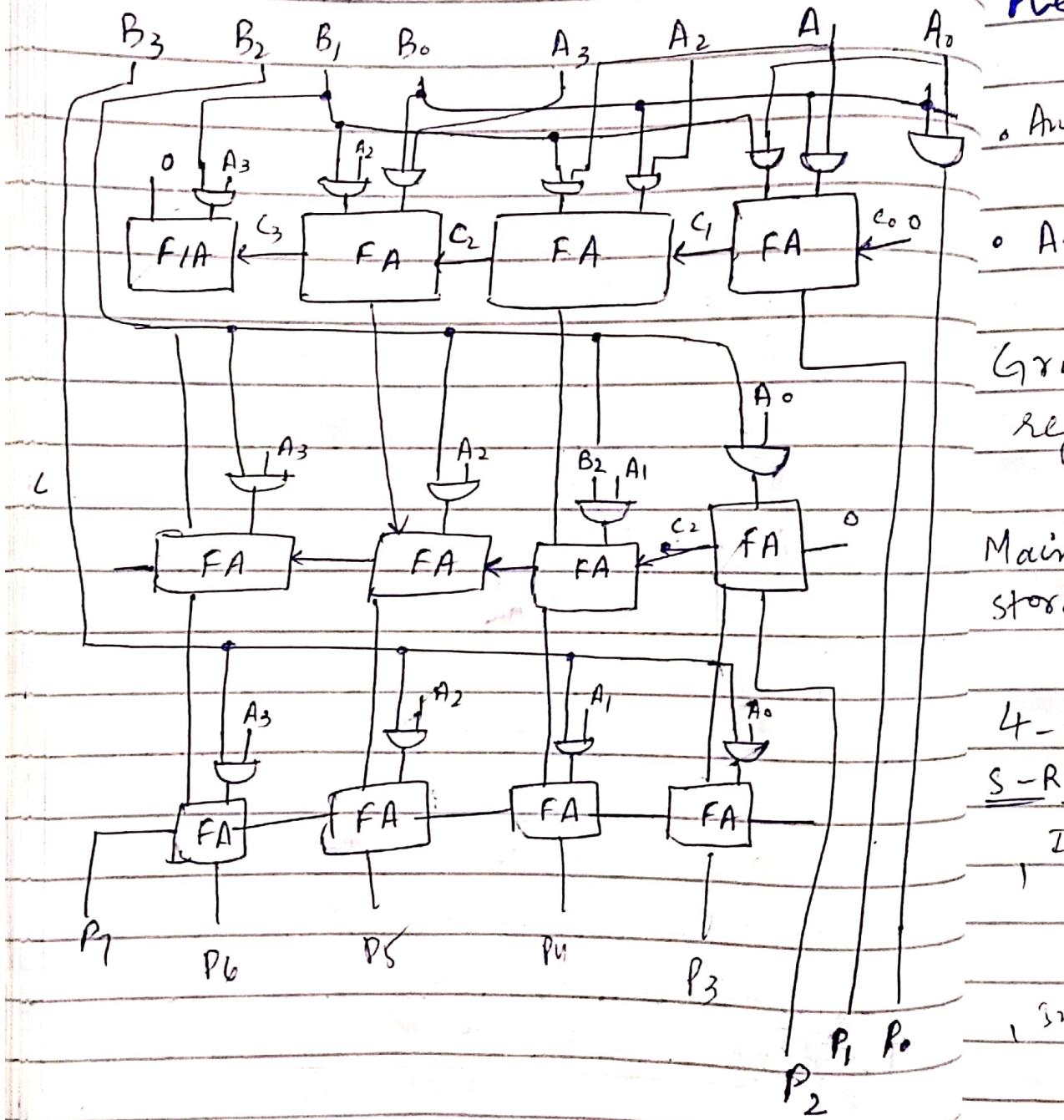
$$1010$$

$C_4$  Same circuit  $C_0$



## 4-bit Multiplier -

$$\begin{array}{r} & A_3 & A_2 & A_1 & A_0 \\ \text{when } 0. & \times & B_3 & B_2 & B_1 & B_0 \\ & & & & & \\ & & A_3B_0 & A_2B_0 & A_1B_0 & A_0B_0 \\ & & A_3B_1 & A_2B_1 & A_1B_1 & A_0B_1 & \times \\ & & A_3B_2 & A_2B_2 & A_1B_2 & A_0B_2 & \times \\ & + & A_3B_3 & A_2B_3 & A_1B_3 & A_0B_3 & \times & \times \\ \hline & P_7 & P_6 & P_5 & P_4 & P_3 & P_2 & P_1 & P_0 \end{array}$$



## Registers:- (Storage device)

(Mostly D Flip Flops are used)

Any flip flop can store only one bit.

An n-bit register can store n-flip-flop.

Group of flip flop that constitute one register.

Main function of a register are data storage and data movement.

### 4-bit Register:-

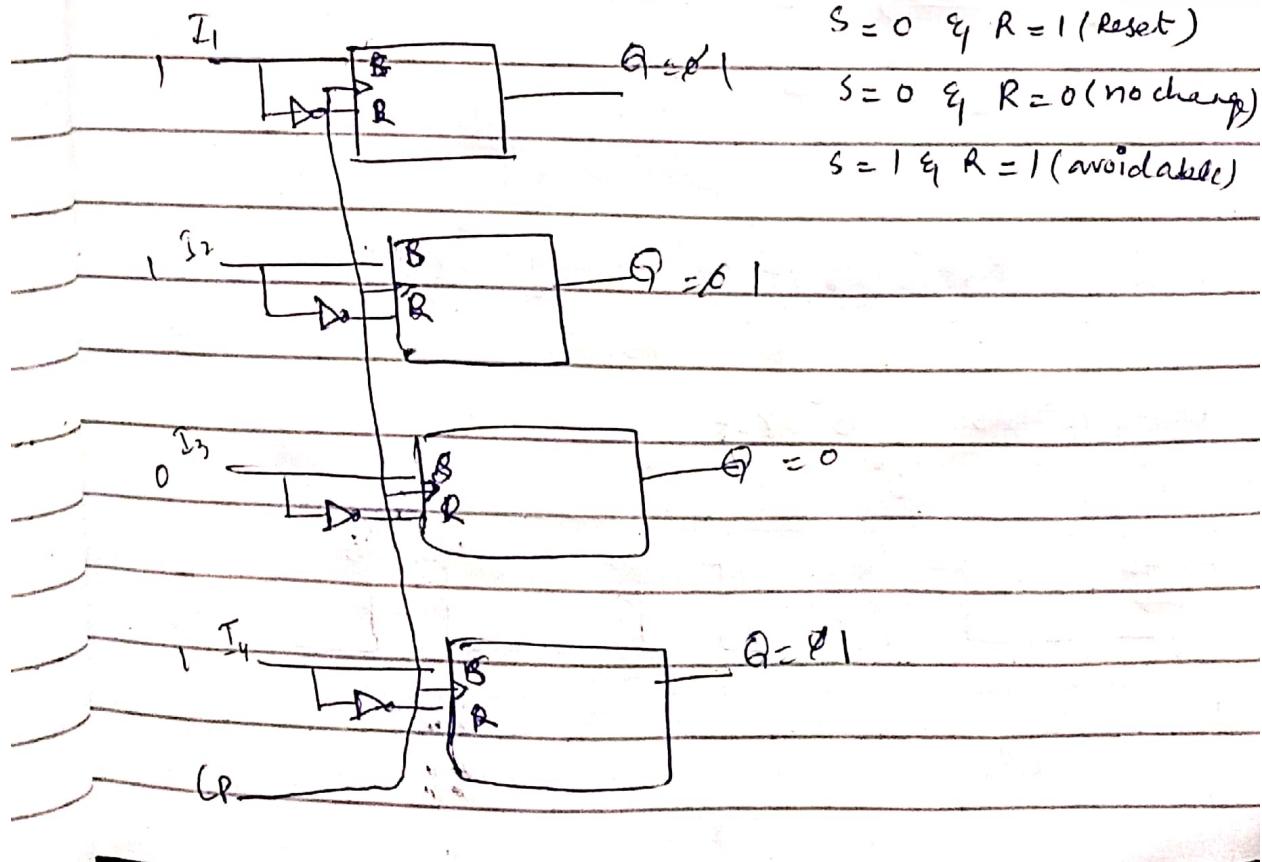
S-R

$S = 1 \& R = 0$  (set)

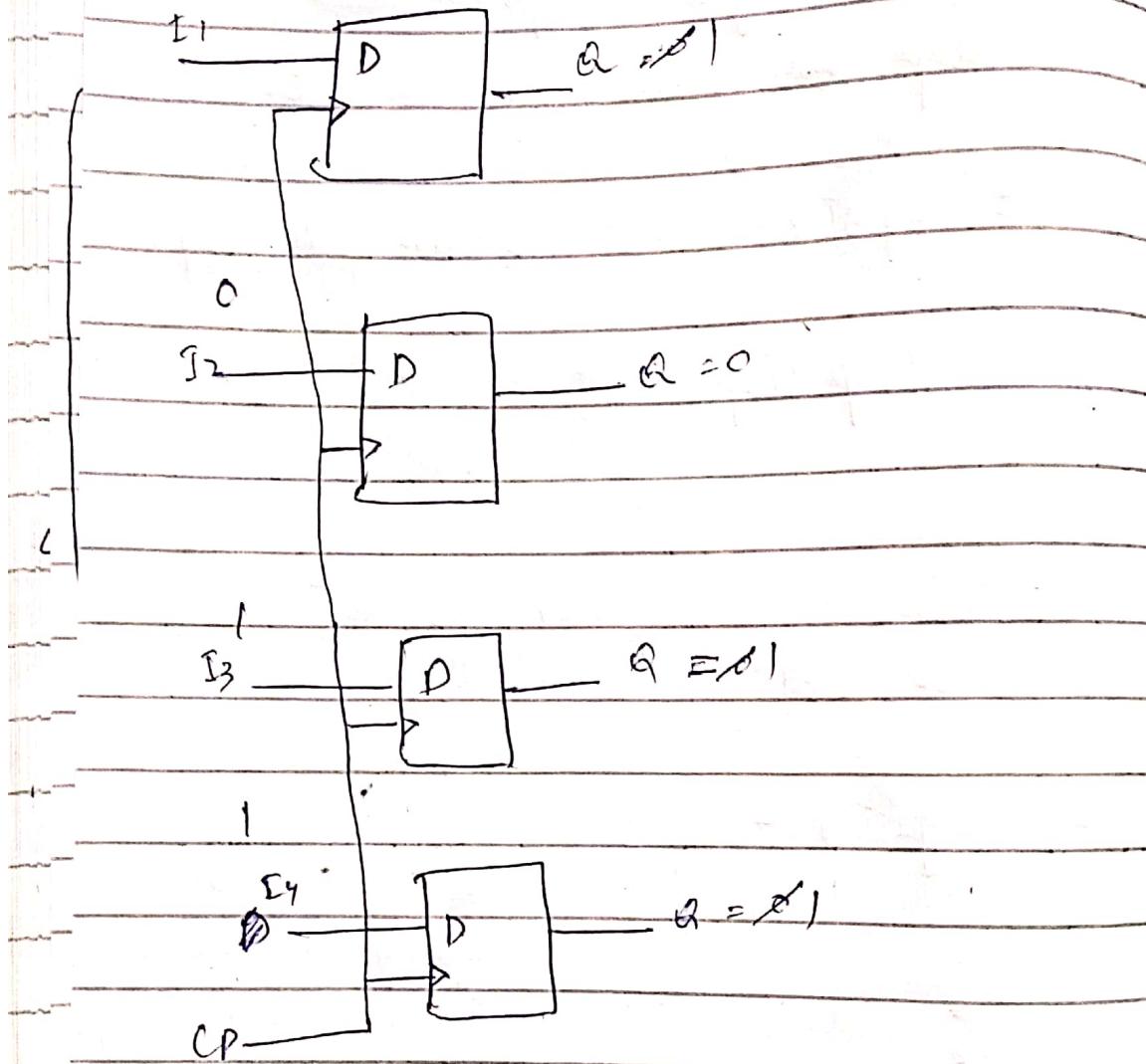
$S = 0 \& R = 1$  (reset)

$S = 0 \& R = 0$  (no change)

$S = 1 \& R = 1$  (avoidable)

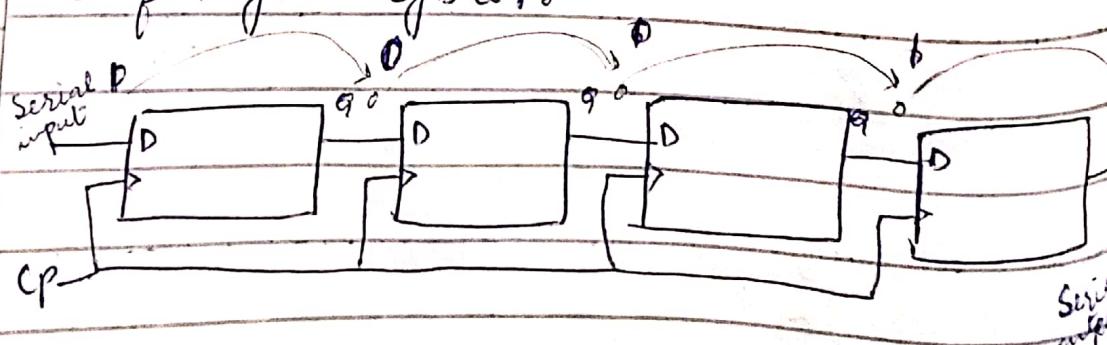


## D flip flops -

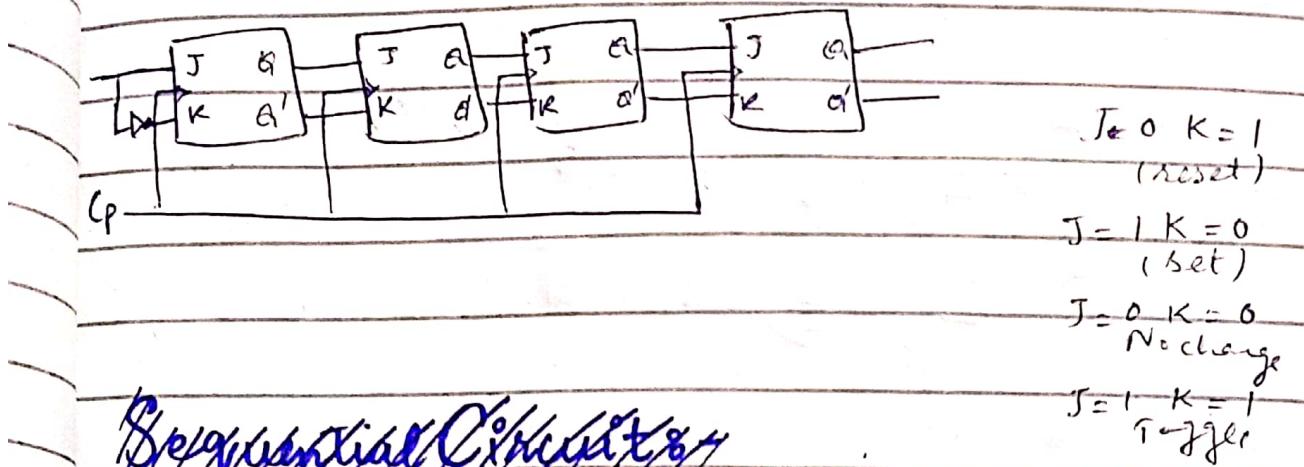


## Shift Register :- (SISO Register)

### Shift Right Register:-



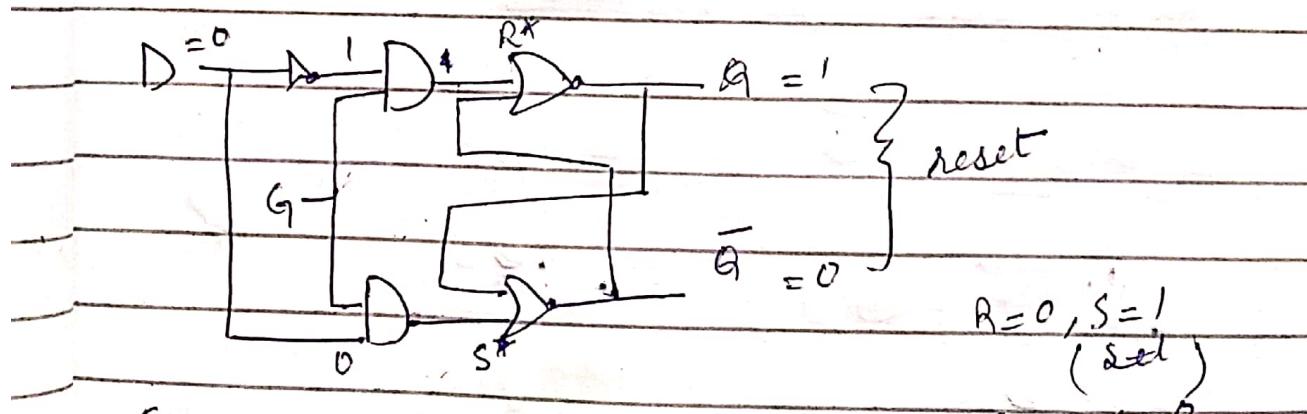
## JK Flip Flop



## Gated D-Latch -

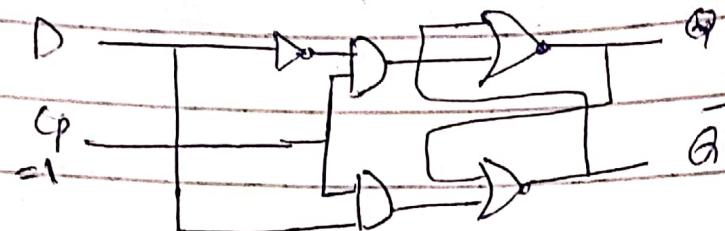
- D latch is a single input version of SR Latch.

D Latch using NOR gate.



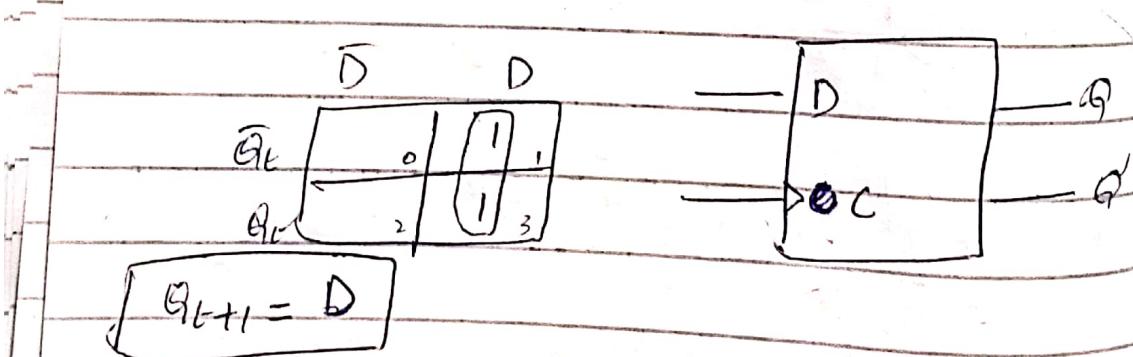
$E_n$	$D$	$Q_t$	$Q_{t+1}$	$R = 1, S = 0$ (reset)
0	0	0 ✓	0 ✓	When $E$ is 0 then
0	0	1 ✓	1 ✓	$Q_{t+1}$ will same as
0	1	0 ✓	0 ✓	$Q_t$ .
1	0 ✗	1 ✓	1 ✓	When $E$ is 1 then
1	0 ✗	0	0	$Q_{t+1}$ will D.
1 ✗	0		1	
1 ✗	1 ✗	1	1	

## Edge Triggered Flip Flops



	$Q_t$	D	$Q_{t+1}$
1	0	0	0
2	0	1	1
3	1	0	0
4	1	1	1

Characteristic equation :-



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

Characteristic equations -

$$\begin{array}{cccc}
 \bar{S}R & \bar{S}R & SR & S\bar{R} \\
 \overline{Q_L} & \boxed{\begin{matrix} 0 \\ \overline{1} \end{matrix}} & \boxed{\begin{matrix} X_3 \\ X_1 \end{matrix}} & \boxed{\begin{matrix} 1 \\ \overline{1} \end{matrix}} \\
 Q_t & \boxed{\begin{matrix} 1 \\ 0 \end{matrix}} & \boxed{\begin{matrix} X_3 \\ X_1 \end{matrix}} & \boxed{\begin{matrix} \overline{1} \\ 0 \end{matrix}}
 \end{array}$$

$$= S + Q_t \bar{R}$$

$$\therefore S \cdot R = 0$$

- Gated SR Latch when active then new state will change according to SR.
- Edge Triggered SR flipflop, +ve edge ~~goes~~ then change, in duration there is no +ve at that it will not effect SR.

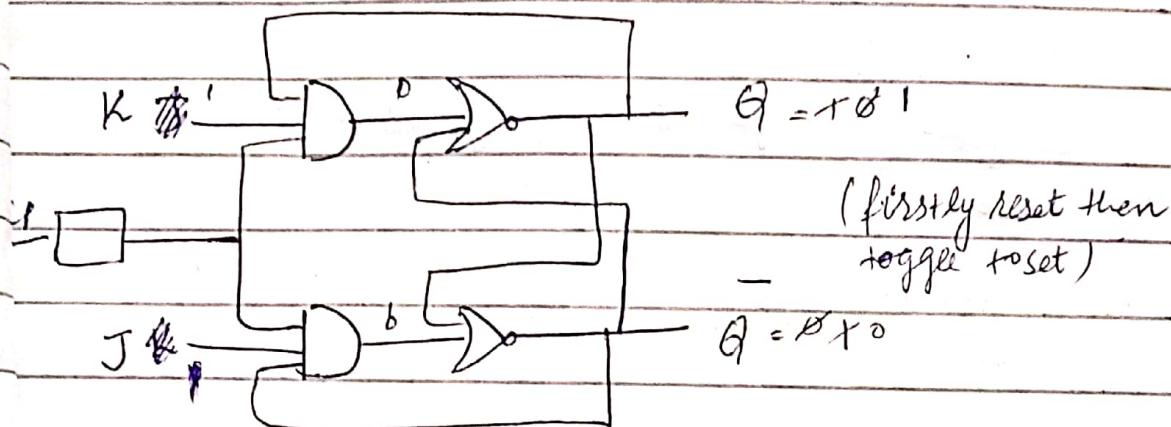
# Edge Triggered JK Flip Flop

$J = 0$  and  $K = 0$  (no change)

$J = 1$  and  $K = 0$  (set)

$J = 0$  and  $K = 1$  (reset)

$J = 1$  and  $K = 1$  (Toggle → Means if set change to reset, if reset change to set),



Characteristic Tables-

$Q_t$	$J$	$K$	$Q_{t+1}$	
0	0	0	0	no change
0	0	1	0	reset
0	1	0	1	set
0	1	1	1	Toggle
1	0	0	1	no change
1	0	1	0	<del>no change</del> reset
1	1	0	1	set
1	1	1	0	Toggle

Characteristic equations-

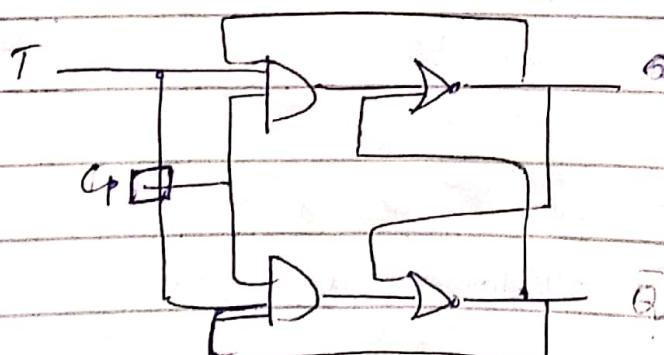
$\bar{J}\bar{K}$	$\bar{J}K$	$J\bar{K}$	$J\bar{K}$
$\bar{Q}_{t+1}$			
$Q_t$	1	1	0

$$Q_{t+1} = \bar{Q}_t J + Q_t \bar{K}$$

- ~~In negative edge the change will according to pulse.~~
- ~~In positive edge the change will according to JK.~~

### Edge Triggered T-flip flop.

Single input version of JK FlipFlop.



$T = 0 \rightarrow \text{no change}$

$T = 1 \rightarrow \text{Toggle}$

• diff betw -ve and  
+ve edge

## Characteristic Tables-

$Q_t$	T	$Q_{t+1}$	
0	0	0	→ no change
0	1	1	→ Toggle
1	0	1	→ no change
1	1	0	→ Toggle

dig		T	T	
g to	$\bar{Q}_t$	1		
	Q <sub>t</sub>	1		
	Q <sub>t</sub>			

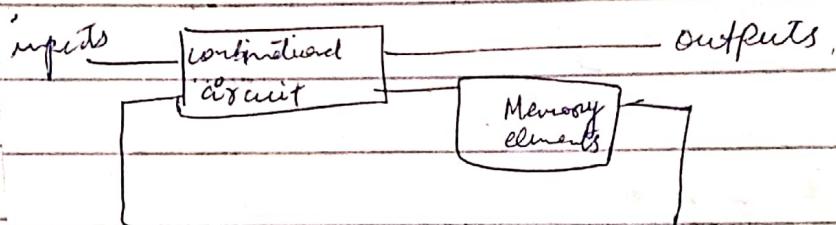
$$Q_{t+1} = \bar{Q}_t T + Q_t \bar{T}$$

## Implementation Tables

## Combinational

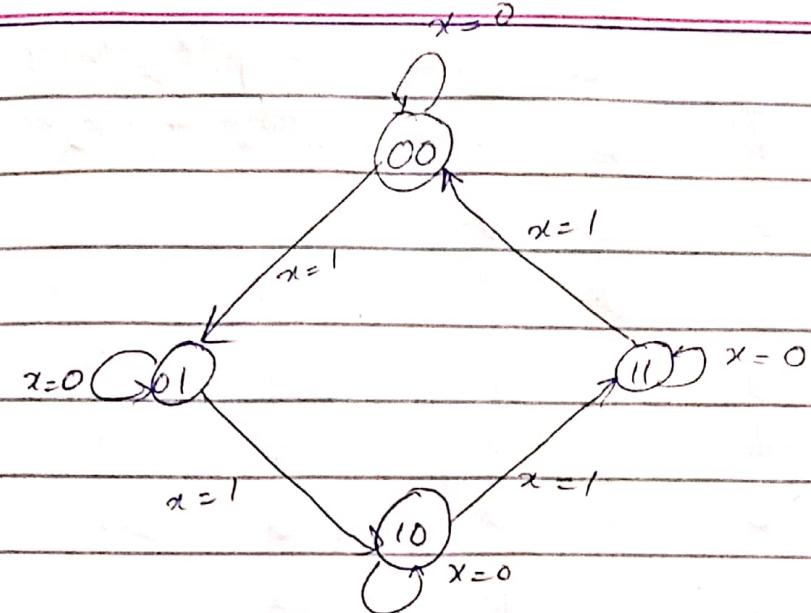
## Sequential Circuits:-

- It not only depend on current output but also depend on previous outputs.
- The Memory elements are the circuits capable to store binary information in them.



## Design Sequential Circuit:-

Design a sequential circuit that goes through a sequence of repeated binary states 00, 01, 10, 11 when external input  $x=1$ . The state of the circuit remains unchanged when  $x=0$ . Use JK flip-flops for the design.



steps:-

- state diagram
- obtain state table
- The number of state may be reduced by state reduction method.
- Assign binary value to each state
- Determine no. of flipflops needed.
- derive ~~Karnaugh~~ circuit output functions and the flip flop input functions
- Draw logic diagram.

at

nals

ipflops

Present state	input	Next state	Flip Flop up				
$A_{Q_t}$	$B_{Q_t}$	$A_{Q_{t+1}}$	$B_{Q_{t+1}}$	$J_A$	$K_A$	$J_B$	$K_B$
0	0	0	0	0	X	0	,
0	0	1	0	1	0	X	1
0	1	0	0	1	0	X	0
0	1	1	1	0	1	X	1
1	0	0	1	0	X	0	0
1	0	1	1	1	X	0	1
1	1	0	1	1	X	0	X
1	1	1	0	0	X	1	X

o Next state will design with the help of diag  
when  $x = 0$  then no change when  $x = 1$  the  
next state.

### Excitation table of JK (F/F)

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

o  $J_A K_A$  and  $J_B K_B$  will be filled with  
help of excitation table. Present state  
and Next state of A for  $J_A K_A$ . Present st

Flop inputs

$J_B \checkmark K_B \checkmark$

$0 \quad x$

$1 \quad x$

$\cancel{X} \quad 0$

$B$  and next state of  $B$  for  $J_B K_B$ -

$x \quad 1$

$J_A \quad \bar{B_2} \quad \bar{B_2}x \quad B_2x \quad B_2\bar{x}$

$K_A \quad \bar{B_2} \quad \bar{B_2} \quad B_2 \quad B_2\bar{x}$

$0 \quad x$

$\bar{A}$	.	.	(1)	2
A	X	X	(X)	X

$\bar{A}$	y	x	(x)	x
A			1	

$1 \quad x$

$x \quad 0$

$$J_A = B_2x$$

$$K_A = B_2\bar{x}$$

$\cancel{X} \quad 1$

$J_B \quad \bar{B_2} \quad \bar{B_2} \quad B_2 \quad B_2\bar{x}$

$K_B \quad \bar{B_2} \quad \bar{B_2} \quad B_2 \quad B_2\bar{x}$

$\bar{A}$	.	(1)	x	x
A	1	x	x	

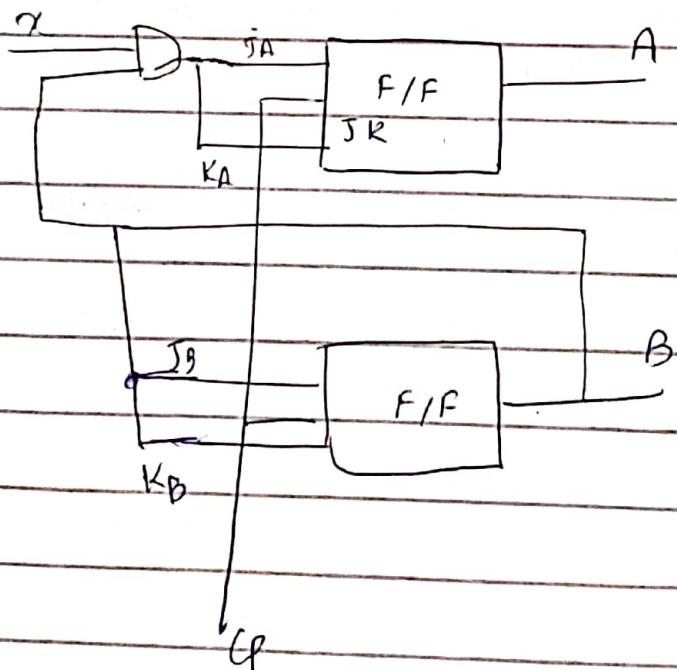
$\bar{A}$	x	x	1
A	x	x	1

diagram

$$J_B = x$$

$$K_B = x$$

then



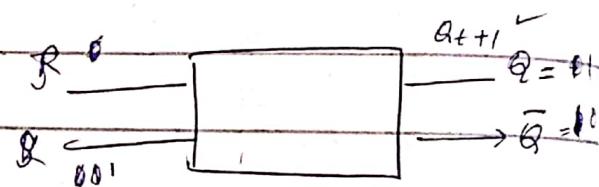
of A

state of

## Excitation Table:-

### RS Flip Flop:-

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



### JK Flip Flops-

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

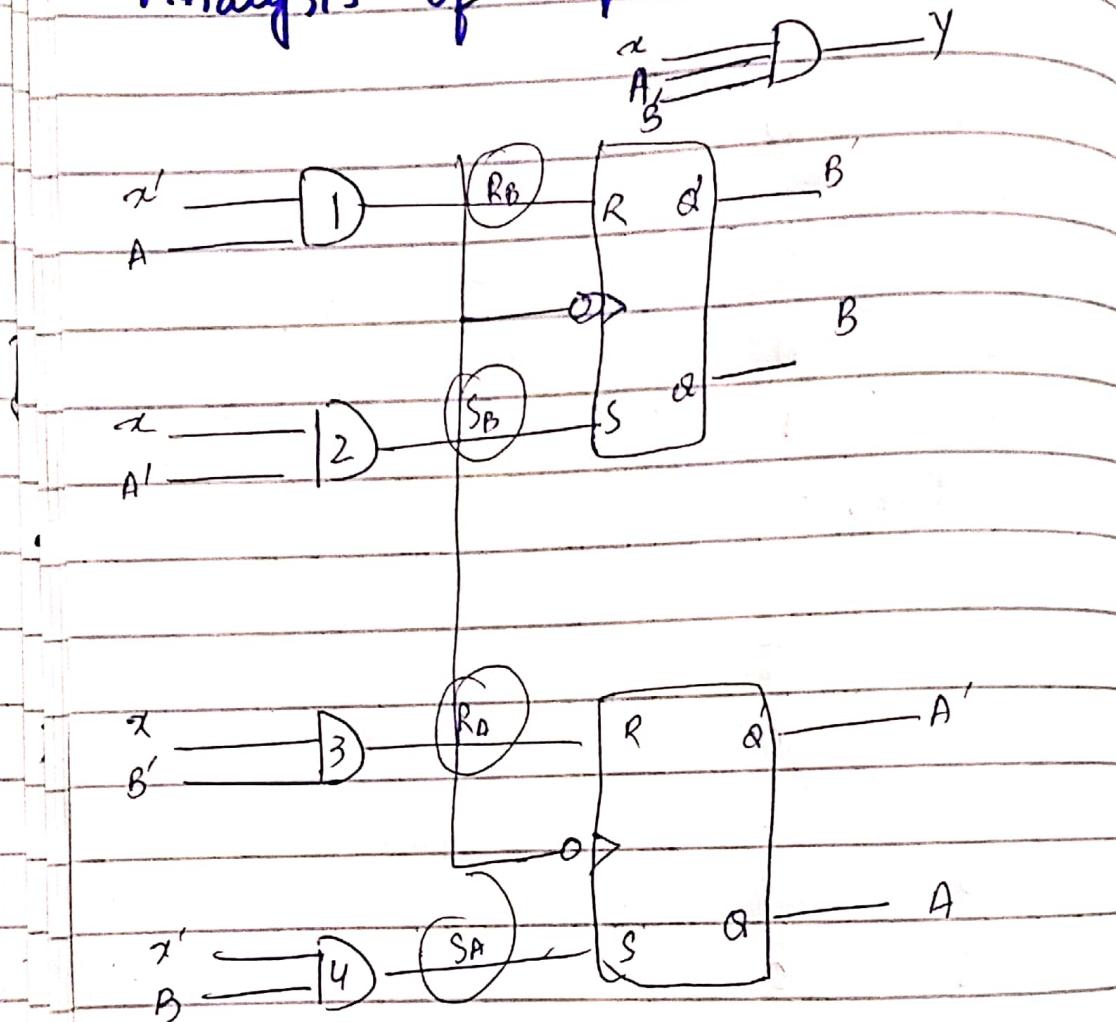
## D Flip Flop :-

$a_t$	$Q_{t+1}$	D
0	0	0
0	1	1
1	0	0
1	1	1

## T Flip Flop :-

$a_{t+1}$	$Q_t$	$Q_{t+1}$	T
$\overline{Q} = 0$	0	0	0
$\overline{Q} = 1$	0	1	1
1	0	1	
1	1	0	

## Analysis of Sequential Circuit :-



$$R_A = \overline{B}x$$

$$S_A = B\bar{x}$$

$$R_B = A\bar{x}$$

$$S_B = \bar{A}x$$

$$Y = \begin{matrix} A & \overline{B}x \\ 1 & 0 & 1 \end{matrix}$$

✓

Present state		Input	Next state		Flip Flop inputs			Y
A	B	x	A	B	R <sub>A</sub>	S <sub>A</sub>	R <sub>B</sub>	S <sub>B</sub>
0	0	0	0	0	0	0	0	0
0✓	0✓	1✓	0	1	1	0	0	1✓0
0	1	0←	1	0	0	1✓	0	0
0✓	1	1✓	0	1	0	0	0	1✓0
1✓	0	0←	0	1	0	0	1✓0	0
1	0✓	1✓	0	0	0	1	0	0
1✓	1	0←	0	1	0	0	1✓	0
1	1	1	1	1	0	0	0	0

state diagram:-

