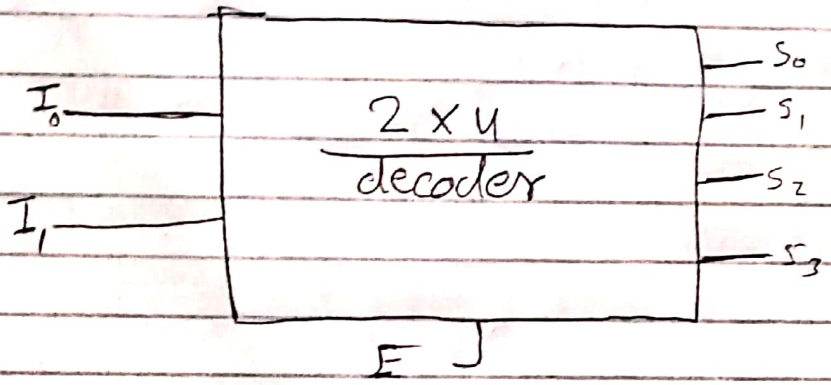
# Note of DLD for final term

decoder is combinational circuit which Taking max n inputs and produce $(2^n)$ outputs

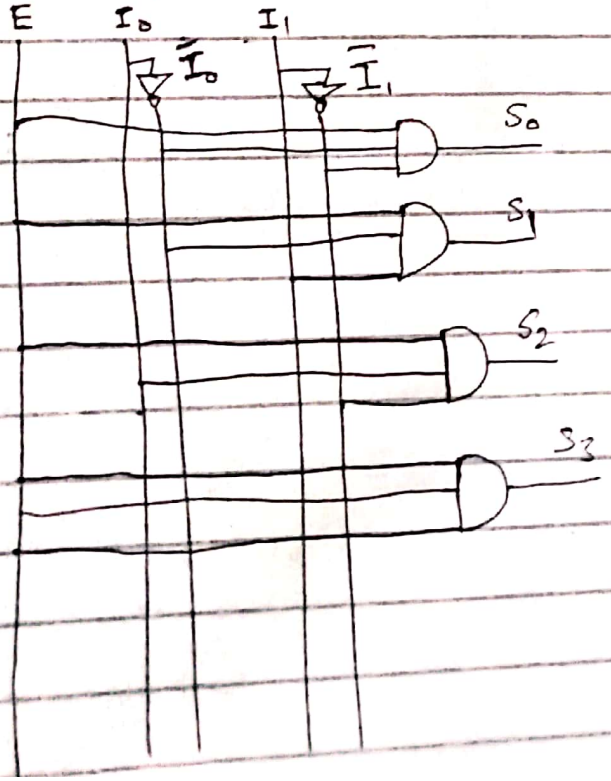
design 2x4 decoder

blaek diagarrrr.



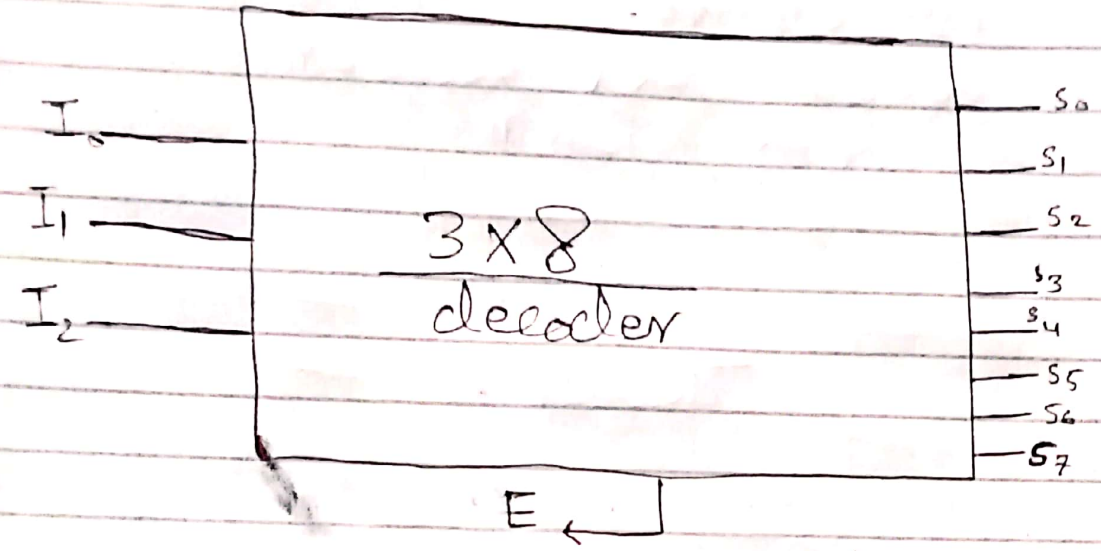

| E | $I_0$ | $I_1$ | $S_0$ | $S_1$ | $S_2$ | $S_3$ |
|---|---|---|---|---|---|---|
| 0 | x | x | 0 | 0 | 0 | 0 |
| 1 | 0 | 0 | 1 | 0 | 0 | 0 |
| 1 | 0 | 1 | 0 | 1 | 0 | 0 |
| 1 | 1 | 0 | 0 | 0 | 1 | 0 |
| 1 | 1 | 1 | 0 | 0 | 0 | 1 |

$S_0 = E\bar{I}_0\bar{I}_1 \quad S_1 = E\bar{I}_0 I_1 \quad S_2 = E I_0 \bar{I}_1 \quad S_3 = E I_0 I_1$



design 3X8 decoder

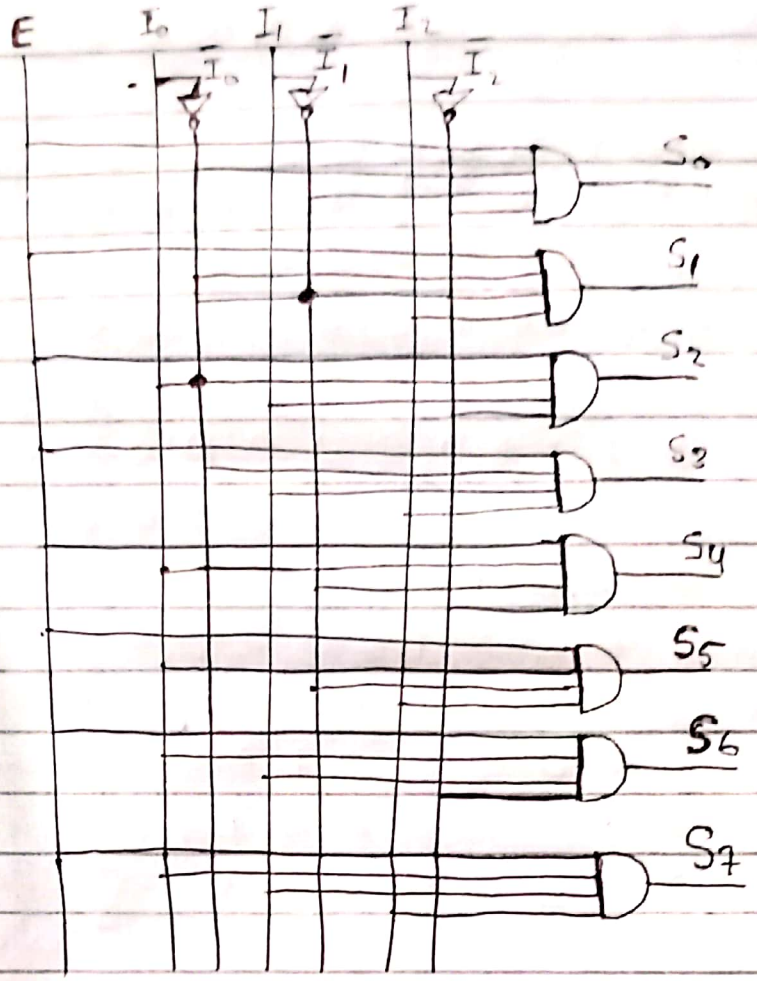
black diagarm:




| E | $I_0$ | $I_1$ | $I_2$ | $S_0$ | $S_1$ | $S_2$ | $S_3$ | $S_4$ | $S_5$ | $S_6$ | $S_7$ |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | X | X | X | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 1 | 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 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 |
| 1 | 0 | 1 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 |
| 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 |
| 1 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 |
| 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 |
| 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |

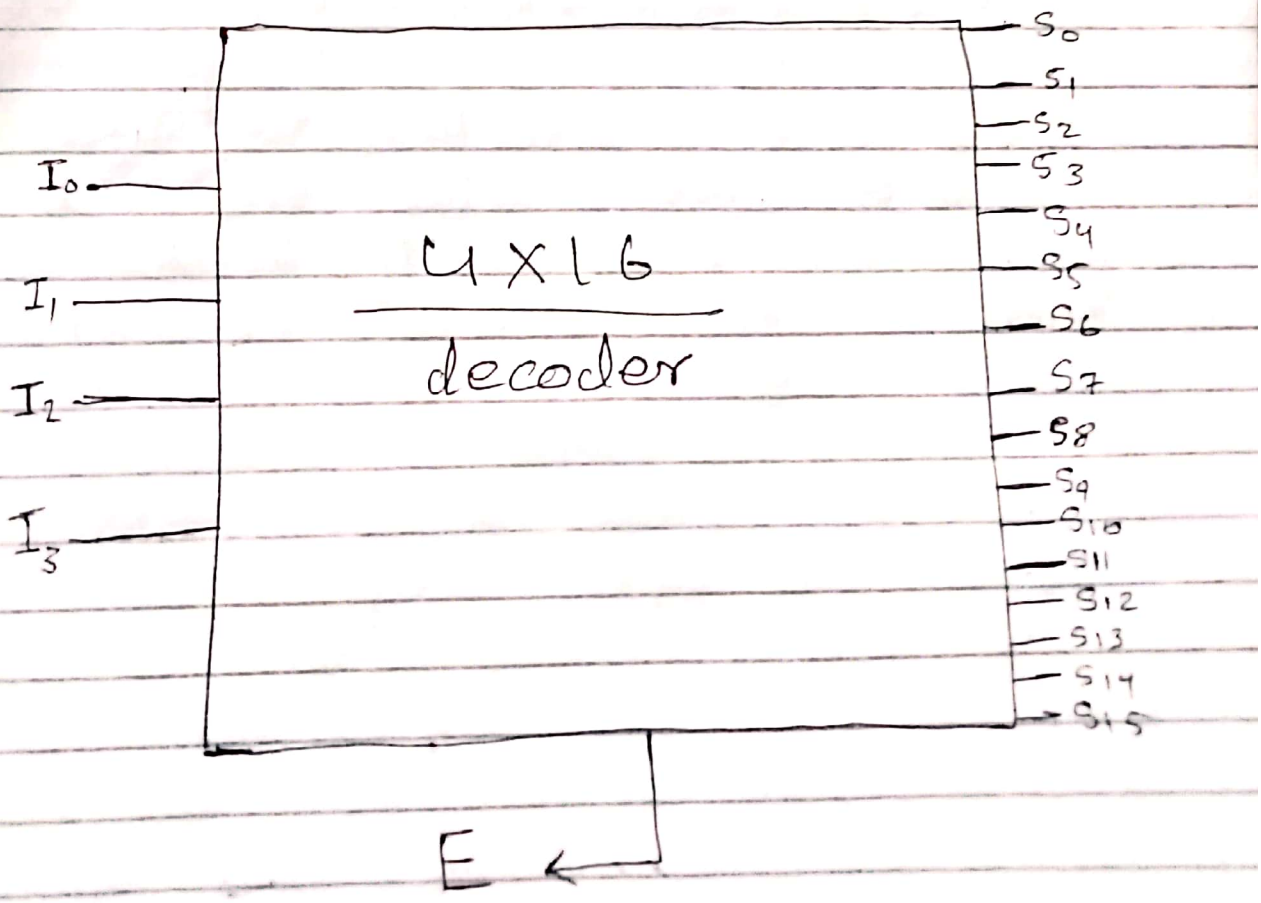
$S_0 = E\bar{I}_0\bar{I}_1\bar{I}_2$ $S_1 = E\bar{I}_0\bar{I}_1 I_2$ $S_2 = E\bar{I}_0 I_1\bar{I}_2$ $S_3 = E\bar{I}_0 I_1 I_2$ $S_4 = EI_0\bar{I}_1\bar{I}_2$

$S_5 = EI_0\bar{I}_1 I_2$ $S_6 = EI_0 I_1\bar{I}_2$ $S_7 = EI_0 I_1 I_2$




So This is 3×8 decoder

Now design 4×16 decoder



# Truth Table

| E | $I_0$ | $I_1$ | $I_2$ | $I_3$ | $S_0$ | $S_1$ | $S_2$ | $S_3$ | $S_4$ | $S_5$ | $S_6$ | $S_7$ | $S_8$ | $S_9$ | $S_{10}$ | $S_{11}$ | $S_{12}$ | $S_{13}$ | $S_{14}$ | $S_{15}$ |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | x | x | x | x | 0 | 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 | 0 |
| 1 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 1 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 1 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 1 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 1 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 1 | 0 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 1 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 |
| 1 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 |
| 1 | 1 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 |
| 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 |
| 1 | 1 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 |
| 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 |
| 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |

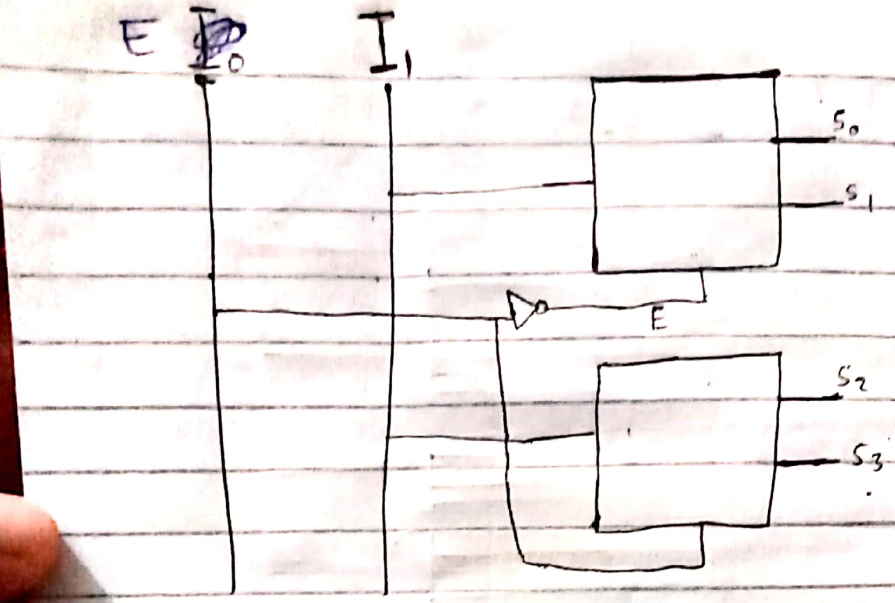
$S_0 = E\bar{I}_0\bar{I}_1\bar{I}_2\bar{I}_3$ $S_1 = E\bar{I}_0\bar{I}_1\bar{I}_2 I_3$ $S_2 = E\bar{I}_0\bar{I}_1 I_2\bar{I}_3$ $S_3 = E\bar{I}_0\bar{I}_1 I_2 I_3$

$S_4 = E\bar{I}_0 I_1\bar{I}_2\bar{I}_3$ $S_5 = E\bar{I}_0 I_1\bar{I}_2 I_3$ $S_6 = E\bar{I}_0 I_1 I_2\bar{I}_3$ $S_7 = E\bar{I}_0 I_1 I_2 I_3$

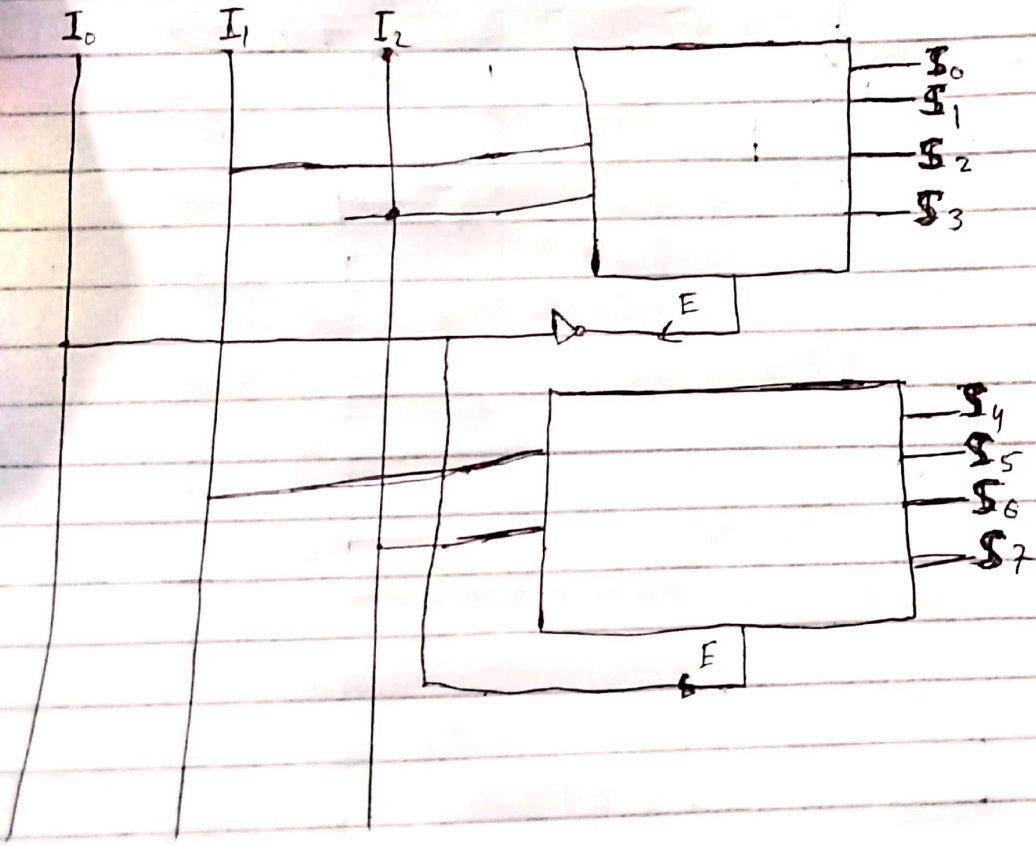
$S_8 = EI_0\bar{I}_1\bar{I}_2\bar{I}_3$ $S_9 = E\bar{I}_0 I_1 I_2\bar{I}_3$ $S_{10} = EI_0\bar{I}_1 I_2\bar{I}_3$ $S_{11} = EI_0\bar{I}_1 I_2 I_3$

$S_{12} = EI_0 I_1\bar{I}_2\bar{I}_3$ $S_{13} = EI_0 I_1\bar{I}_2 I_3$ $S_{14} = EI_0 I_1 I_2\bar{I}_3$ $S_{15} = EI_0 I_1 I_2 I_3$

design 2x4 decoder using 1x2 decoder

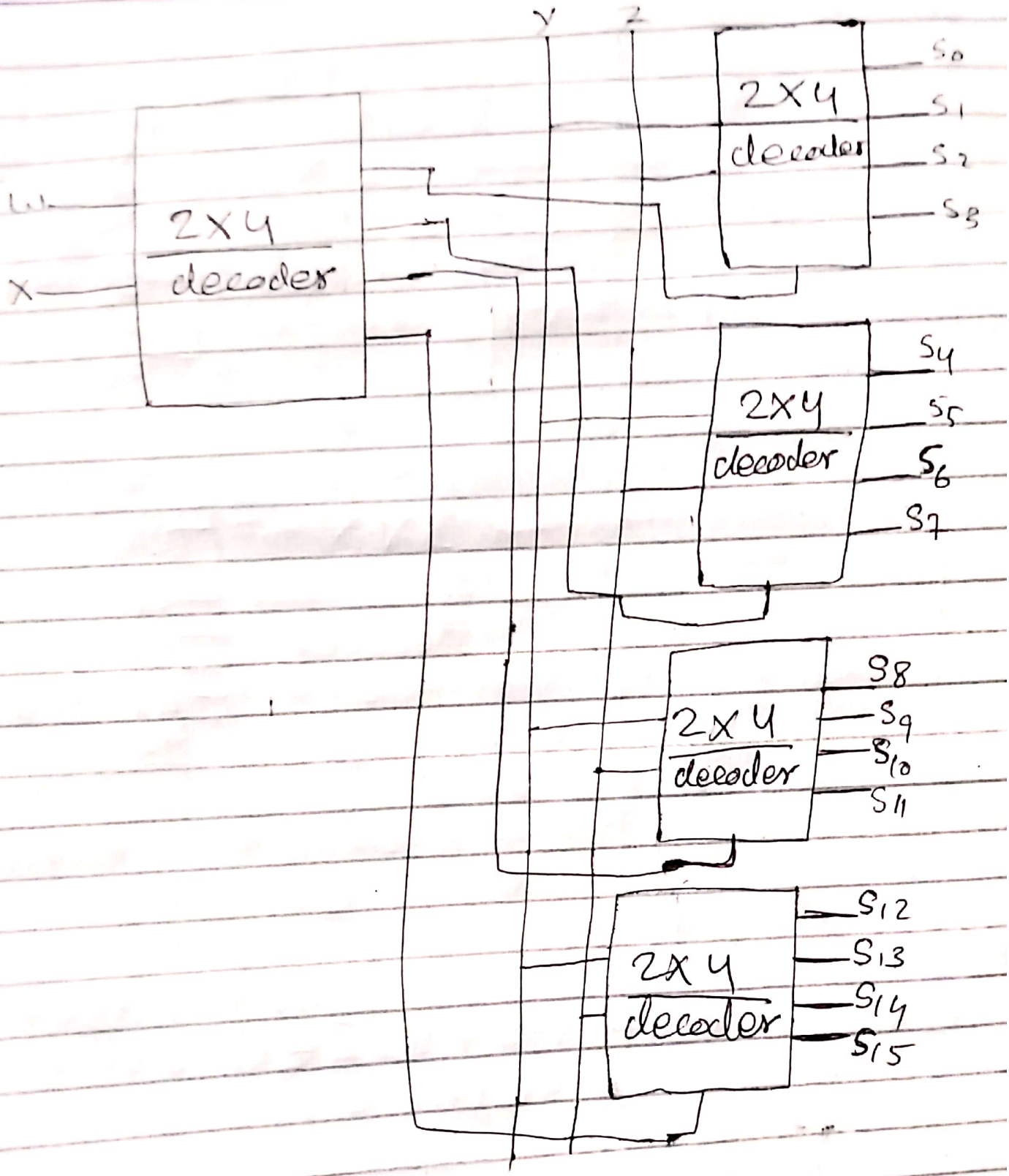



design 3x8 decoder using 2x4 decoder

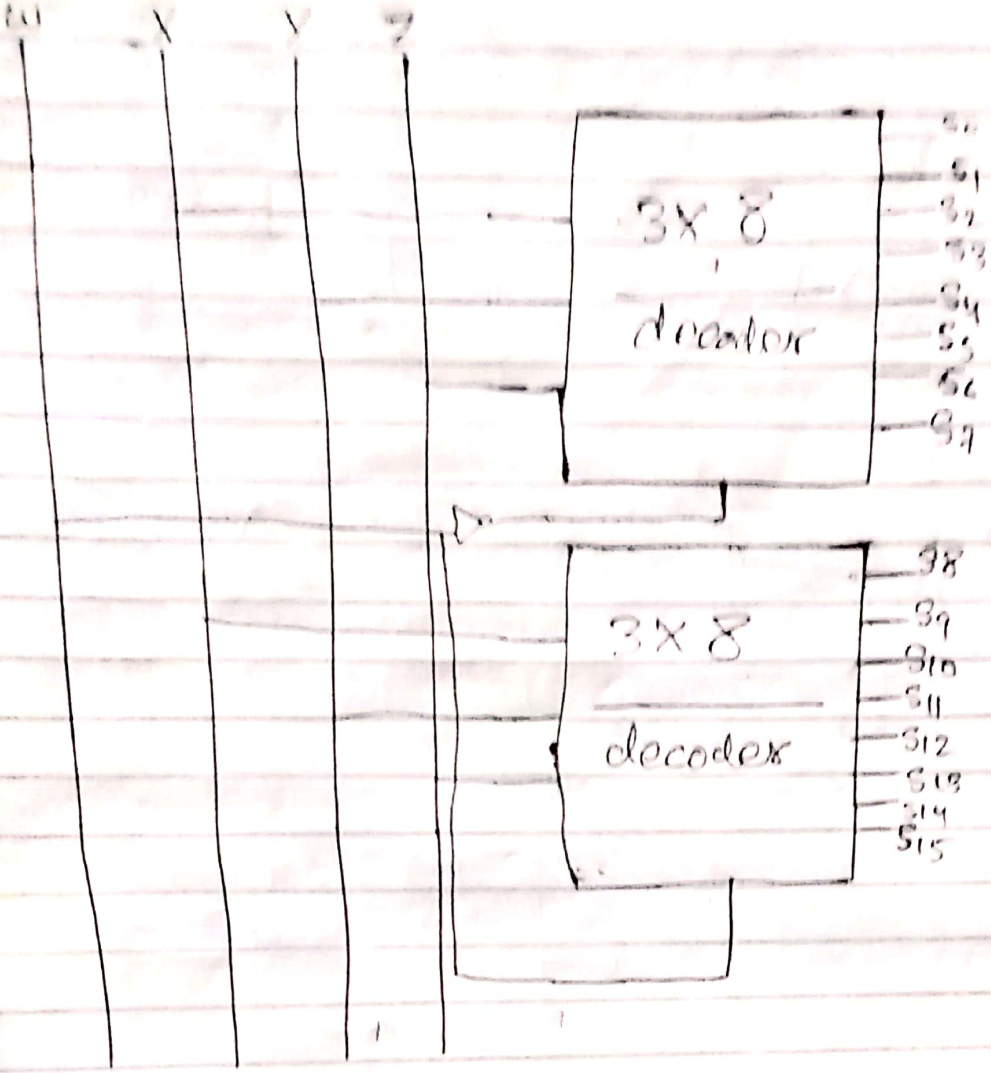



so this is 3x8 decoder using 2x4 decoder

design 4X16 decoder using 2X4

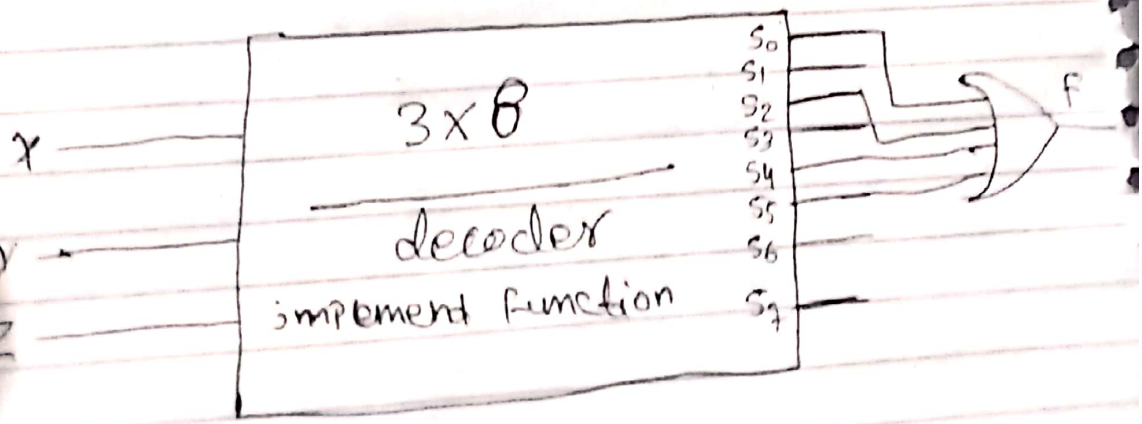


design 4x16 decoder using 3x8 decoder



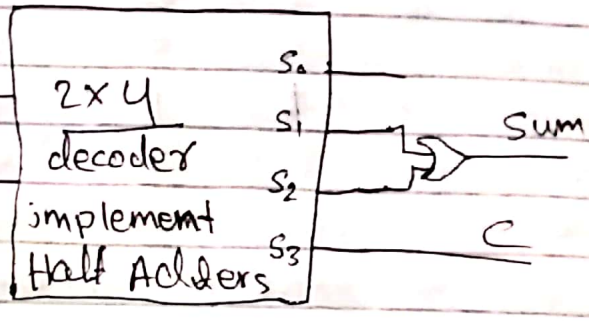

$$F(x,y,z) = \bar{x}\bar{y}\bar{z} + x\bar{y}z + \bar{x}yz + x\bar{y}\bar{z}$$

$$F(x,y,z) = \Sigma(0,5,2,4)$$



## Implement Half Adders using decoder

| | A | B | S | C |
|---|---|---|---|---|
| $m_0$ | 0 | 0 | 0 | 0 |
| $m_1$ | 0 | 1 | 1 | 0 |
| $m_2$ | 1 | 0 | 1 | 0 |
| $m_3$ | 1 | 1 | 0 | 1 |




$S = \bar{A}B + A\bar{B} = S = \Sigma(1,2)$

$C = AB = C = \Sigma(3)$

## implementation of full Adders using decoders

| | A | B | C | S | C |
|---|---|---|---|---|---|
| $m_0$ | 0 | 0 | 0 | 0 | 0 |
| $m_1$ | 0 | 0 | 1 | 1 | 0 |
| $m_2$ | 0 | 1 | 0 | 1 | 0 |
| $m_3$ | 0 | 1 | 1 | 0 | 1 |
| $m_4$ | 1 | 0 | 0 | 1 | 0 |
| $m_5$ | 1 | 0 | 1 | 0 | 1 |
| $m_6$ | 1 | 1 | 0 | 0 | 1 |
| $m_7$ | 1 | 1 | 1 | 1 | 1 |

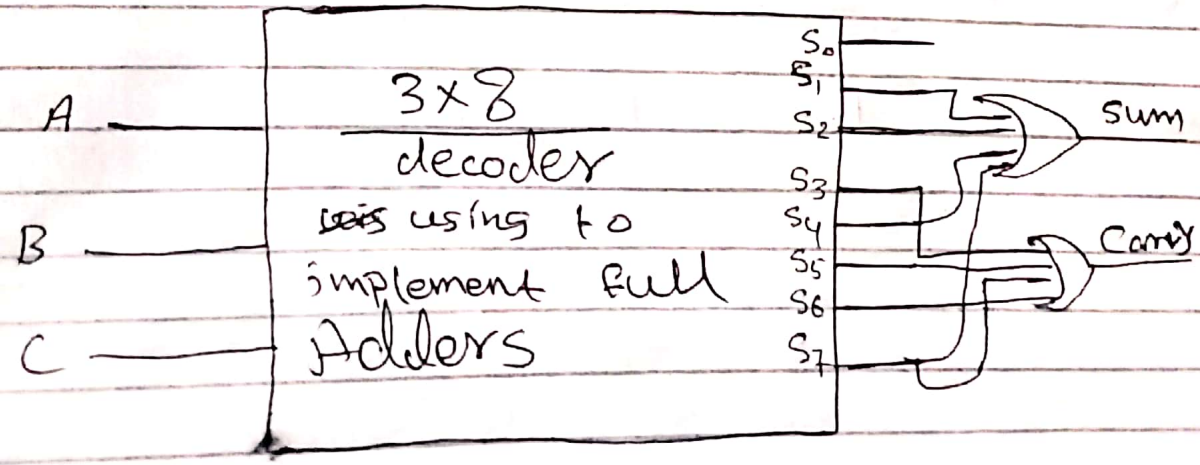
$S = \bar{A}\bar{B}C + \bar{A}B\bar{C} + A\bar{B}\bar{C} + ABC$

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

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

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

So in full Adder we use Theree variable
So we should use 3×8 decoder




So This The implementation of full Adders using decoders

## implementation of Half subtractor using decoder

| | A | B | D | $B_{in}$ |
|---|---|---|---|---|
| $m_0$ | 0 | 0 | 0 | 0 |
| $m_1$ | 0 | 1 | 1 | 1 |
| $m_2$ | 1 | 0 | 1 | 0 |
| $m_3$ | 1 | 1 | 0 | 0 |

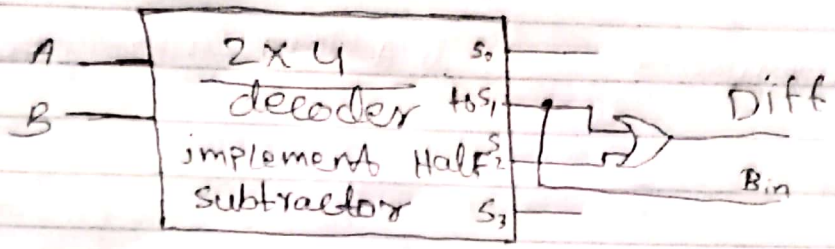
$D = \bar{A}B + A\bar{B}$

$D = \Sigma(1,2)$

$B_{in} = \bar{A}B$

$B_{in} = \Sigma(1)$

So we want to implement Half subtractor so we should use 2 x 4 decoder for it




## implementation of full subtractor using decoder

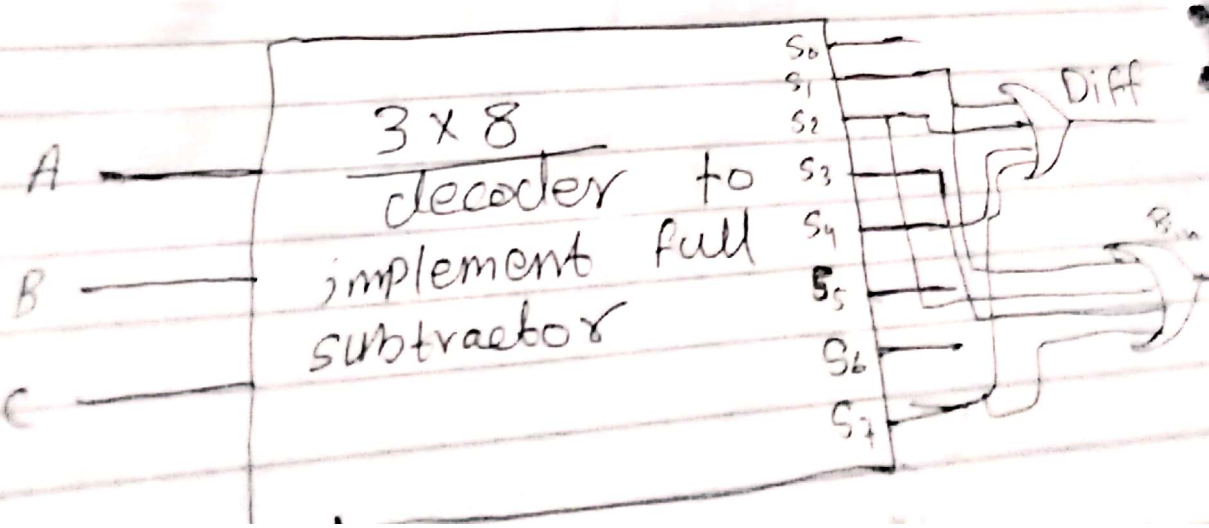
| | A | B | C | Diff | $B_{in}$ |
|---|---|---|---|---|---|
| $m_0$ | 0 | 0 | 0 | 0 | 0 |
| $m_1$ | 0 | 0 | 1 | 1 | 1 |
| $m_2$ | 0 | 1 | 0 | 1 | 1 |
| $m_3$ | 0 | 1 | 1 | 0 | 1 |
| $m_4$ | 1 | 0 | 0 | 1 | 0 |
| $m_5$ | 1 | 0 | 1 | 0 | 0 |
| $m_6$ | 1 | 1 | 0 | 0 | 0 |
| $m_7$ | 1 | 1 | 1 | 1 | 1 |

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

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

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

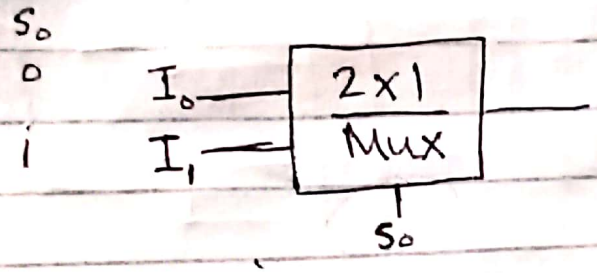
$B_{in} = \Sigma(1,2,3,7)$



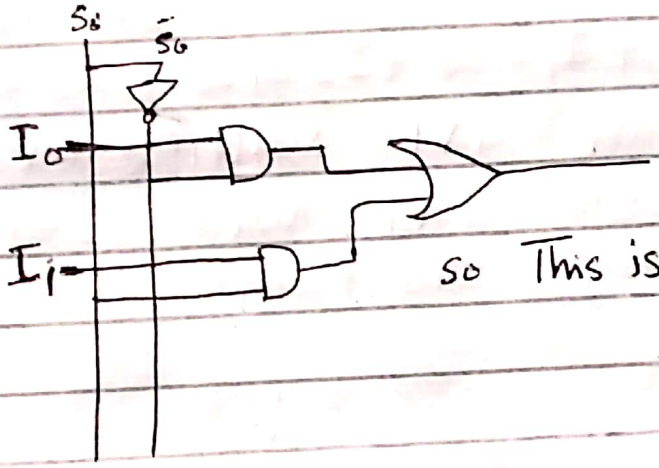
# Note of Multipelexer

Multiplexer is a combinational circuit which Take n input and produce one output and it has n-1 selcabion lines

design Two by one 2x1 Multiplexer

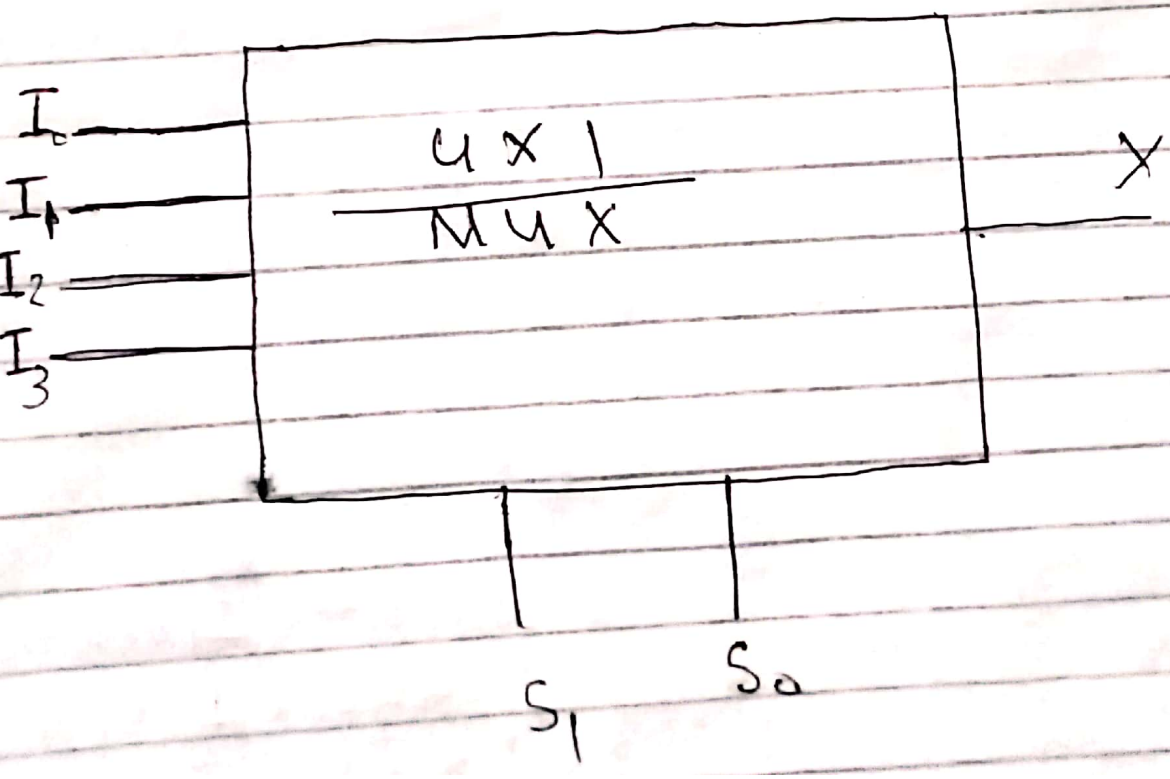



$\bar{S}_0 I_0 + S_0 I_1$



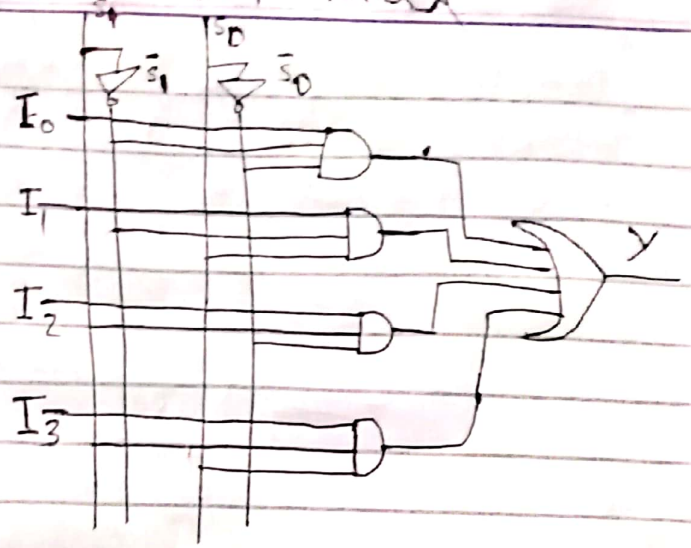

so This is 2x1 MUX

design 4x1 Multiplexer



# Turth Table for 4x1 MUX

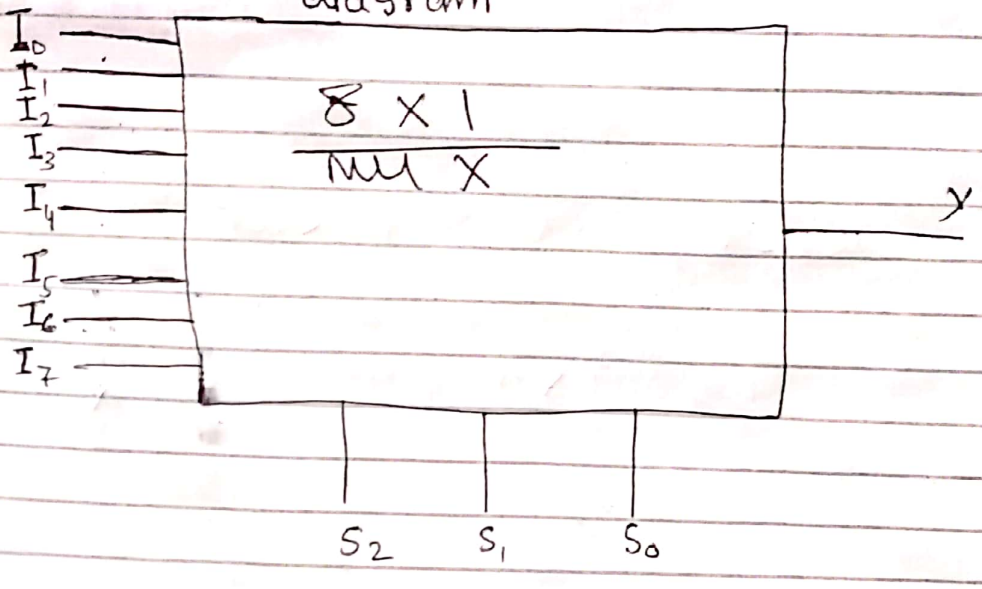
| $S_1$ | $S_0$ | y |
|---|---|---|
| 0 | 0 | $I_0$ |
| 0 | 1 | $I_1$ |
| 1 | 0 | $I_2$ |
| 1 | 1 | $I_3$ |




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

This is an 4x1 multiplexer.

Now design 8x1 multiplexer frist black diagram

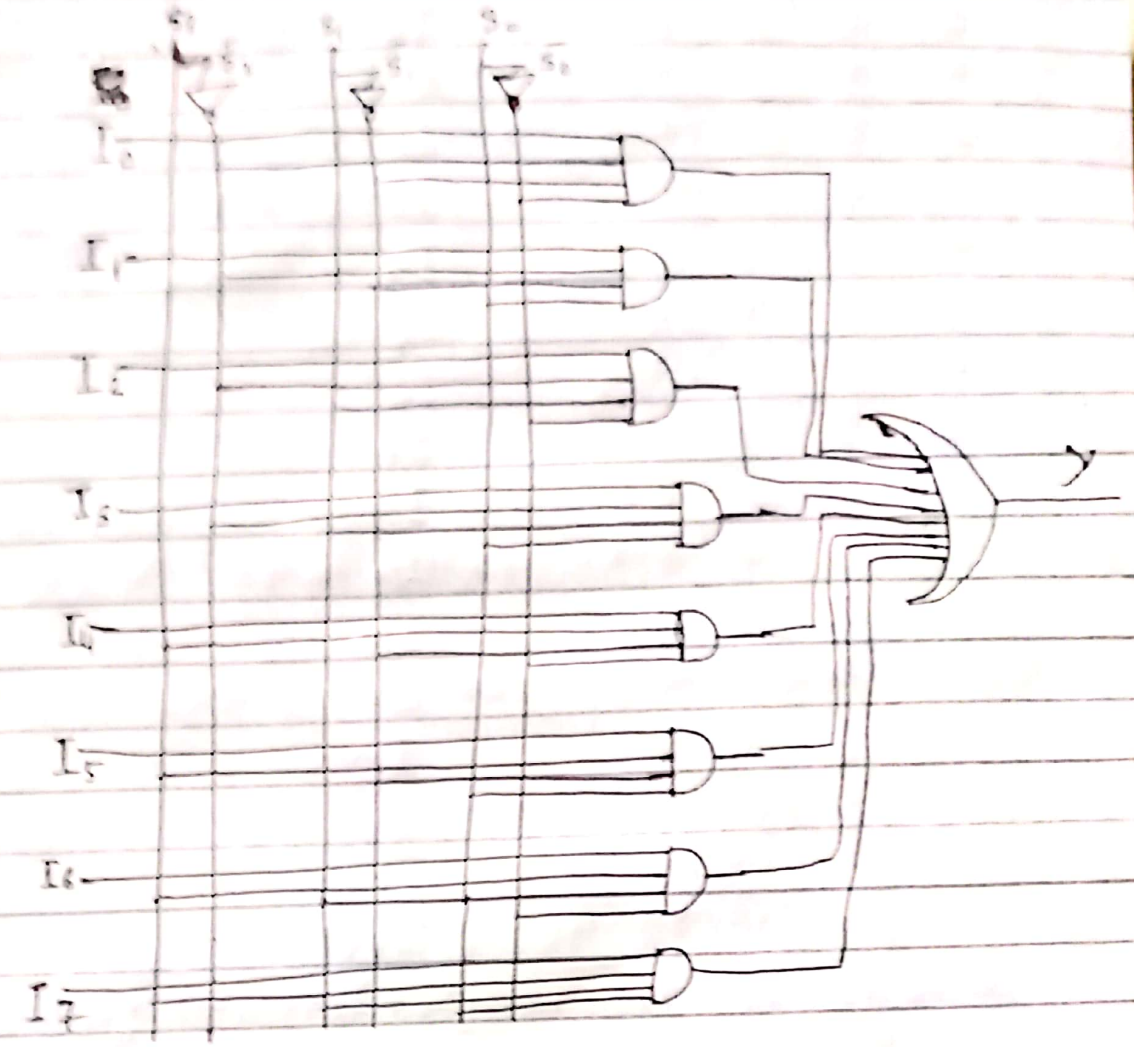



| $S_2$ | $S_1$ | $S_0$ | y |
|---|---|---|---|
| 0 | 0 | 0 | $I_0$ |
| 0 | 0 | 1 | $I_1$ |
| 0 | 1 | 0 | $I_2$ |
| 0 | 1 | 1 | $I_3$ |
| 1 | 0 | 0 | $I_4$ |
| 1 | 0 | 1 | $I_5$ |
| 1 | 1 | 0 | $I_6$ |
| 1 | 1 | 1 | $I_7$ |

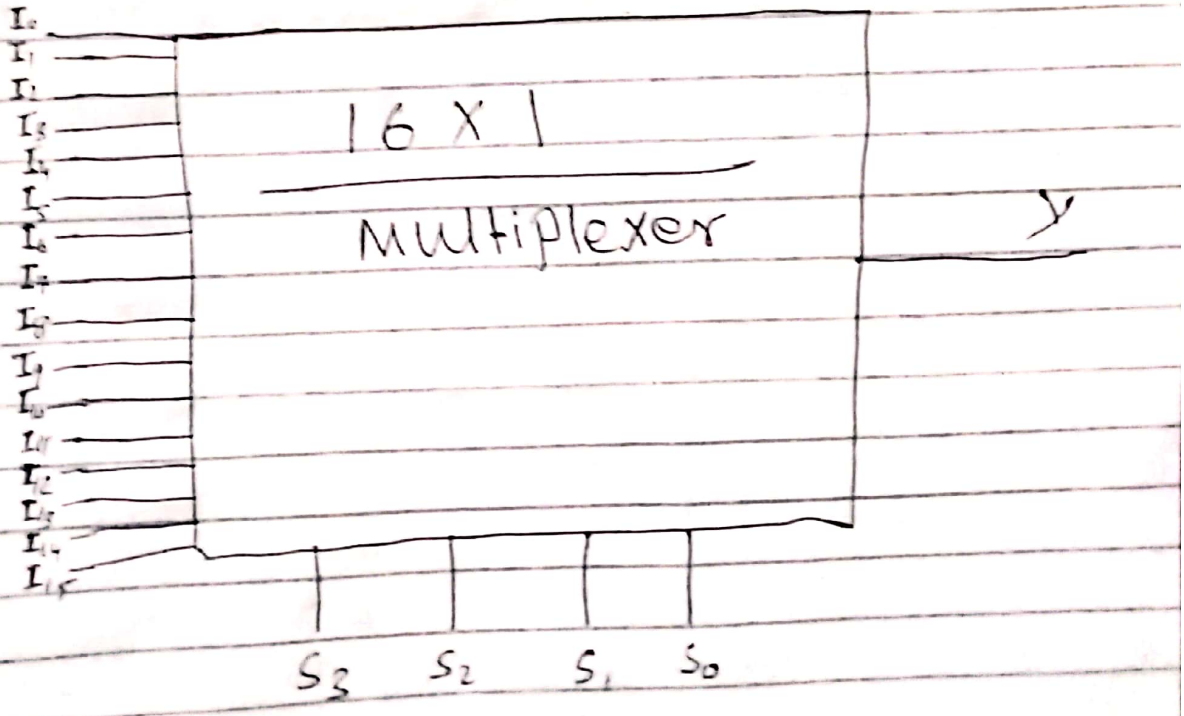
$$y = I_0 \bar{S}_2 \bar{S}_1 \bar{S}_0 + I_1 \bar{S}_2 \bar{S}_1 S_0 + I_2 \bar{S}_2 S_1 \bar{S}_0 + I_3 \bar{S}_2 S_1 S_0 + I_4 S_2 \bar{S}_1 \bar{S}_0 + I_5 S_2 \bar{S}_1 S_0 + I_6 S_2 S_1 \bar{S}_0 + I_7 S_2 S_1 S_0$$

This function for 8x1 MUX

diagram of 8x1 MUX



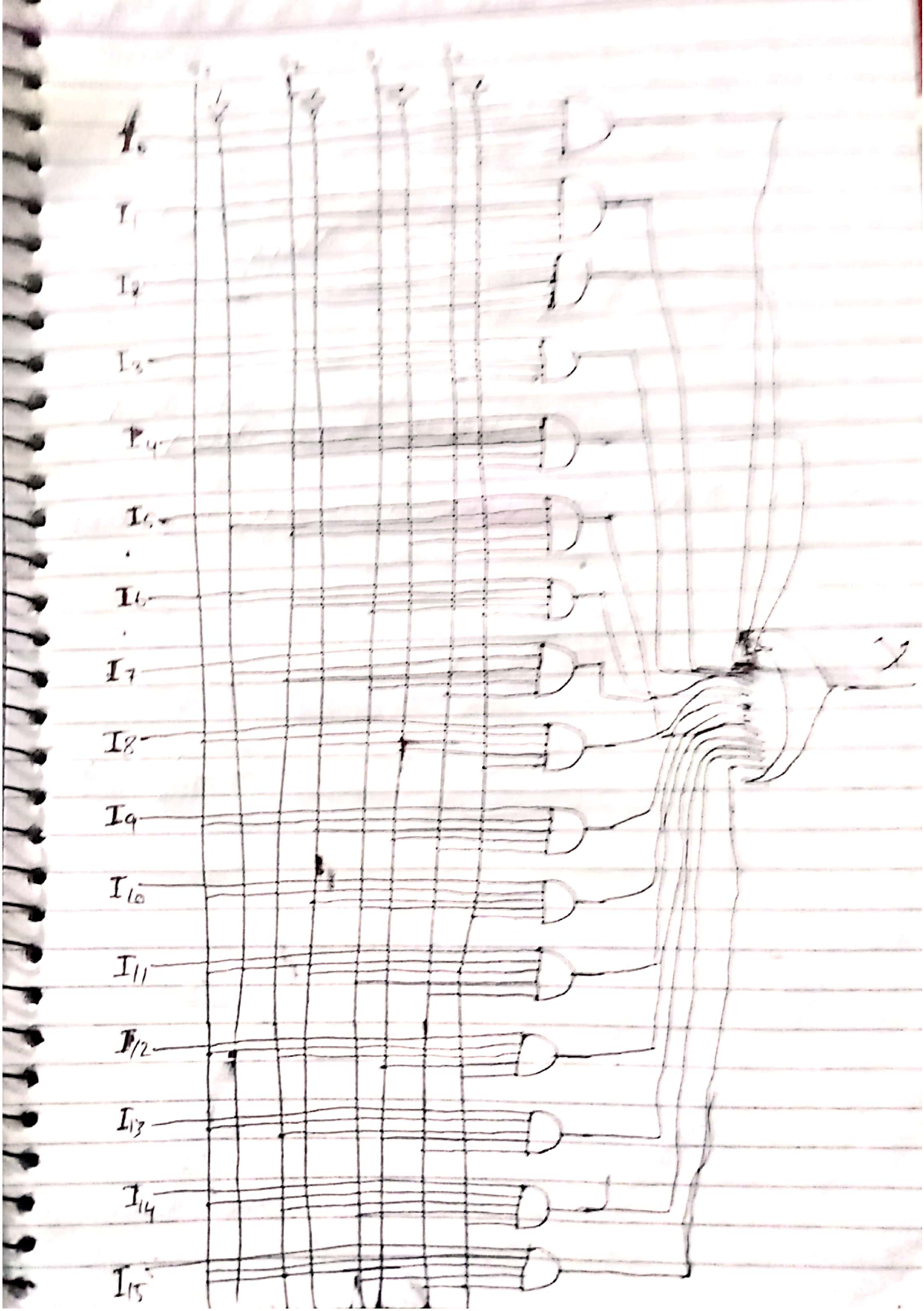

Design 16x1 multiplexer frist balck diagram



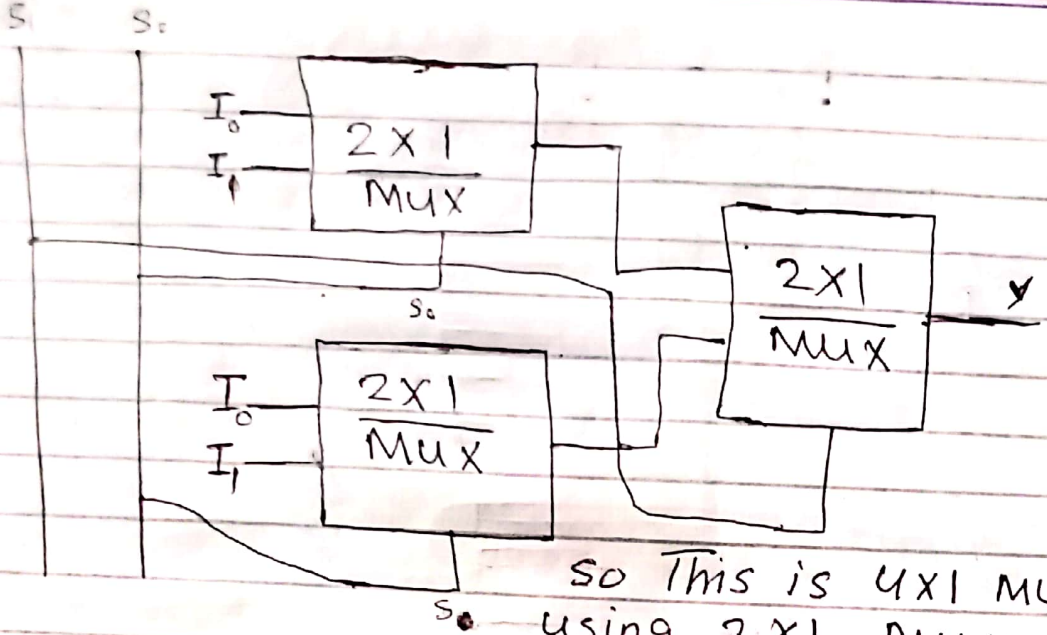
Now Turth Table for 16x1 Multiplexer

| $S_3$ | $S_2$ | $S_1$ | $S_0$ | y |
|---|---|---|---|---|
| 0 | 0 | 0 | 0 | $I_0$ |
| 0 | 0 | 0 | 1 | $I_1$ |
| 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 | $I_{15}$ |

$$y = I_0 \bar{S}_3 \bar{S}_2 \bar{S}_1 \bar{S}_0 + I_1 \bar{S}_3 \bar{S}_2 \bar{S}_1 S_0 + I_2 \bar{S}_3 \bar{S}_2 S_1 \bar{S}_0 + I_3 \bar{S}_3 \bar{S}_2 S_1 S_0 + I_4 \bar{S}_3 S_2 \bar{S}_1 \bar{S}_0 + I_5 \bar{S}_3 S_2 \bar{S}_1 S_0 + I_6 \bar{S}_3 S_2 S_1 \bar{S}_0 + I_7 \bar{S}_3 S_2 S_1 S_0 + I_8 S_3 \bar{S}_2 \bar{S}_1 \bar{S}_0 + I_9 S_3 \bar{S}_2 \bar{S}_1 S_0 + I_{10} S_3 \bar{S}_2 S_1 \bar{S}_0 + I_{11} S_3 \bar{S}_2 S_1 S_0 + I_{12} S_3 S_2 \bar{S}_1 \bar{S}_0 + I_{13} S_3 S_2 \bar{S}_1 S_0 + I_{14} S_3 S_2 S_1 \bar{S}_0 + I_{15} S_3 S_2 S_1 S_0$$

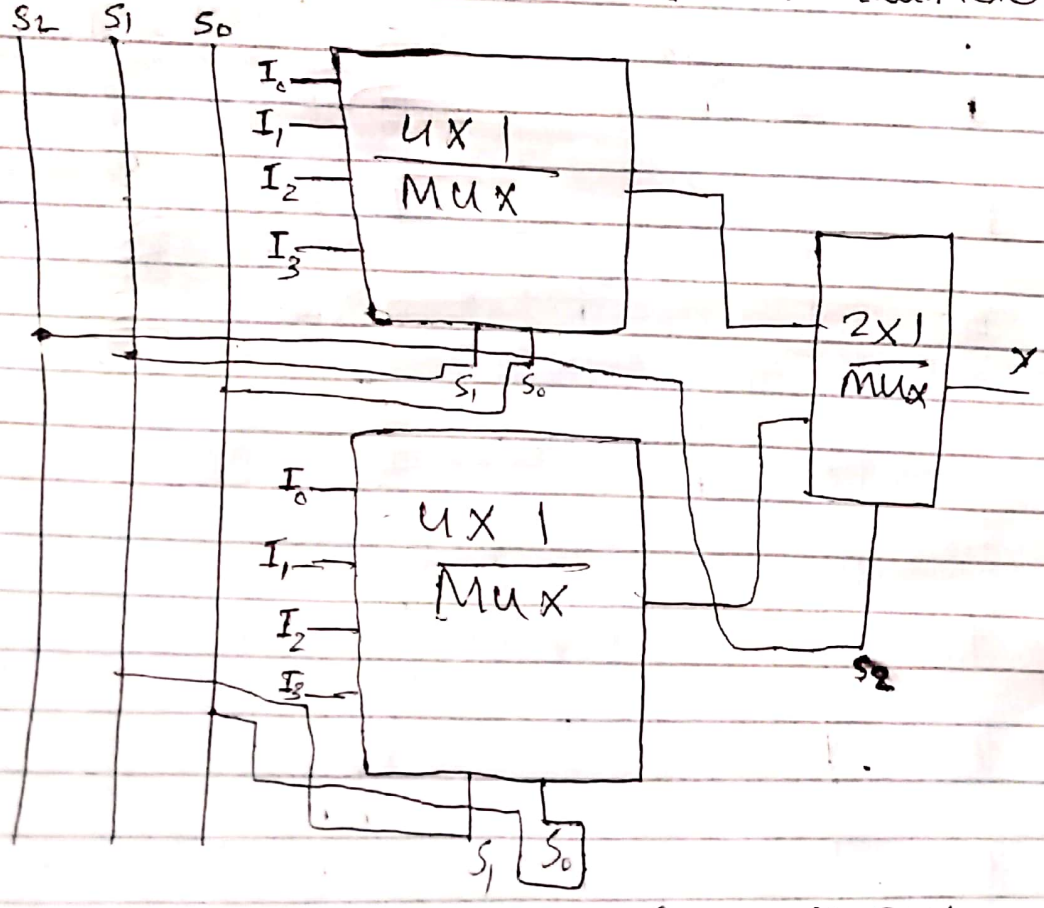


Design 4x1 Multiplexer using 2x1 Mux



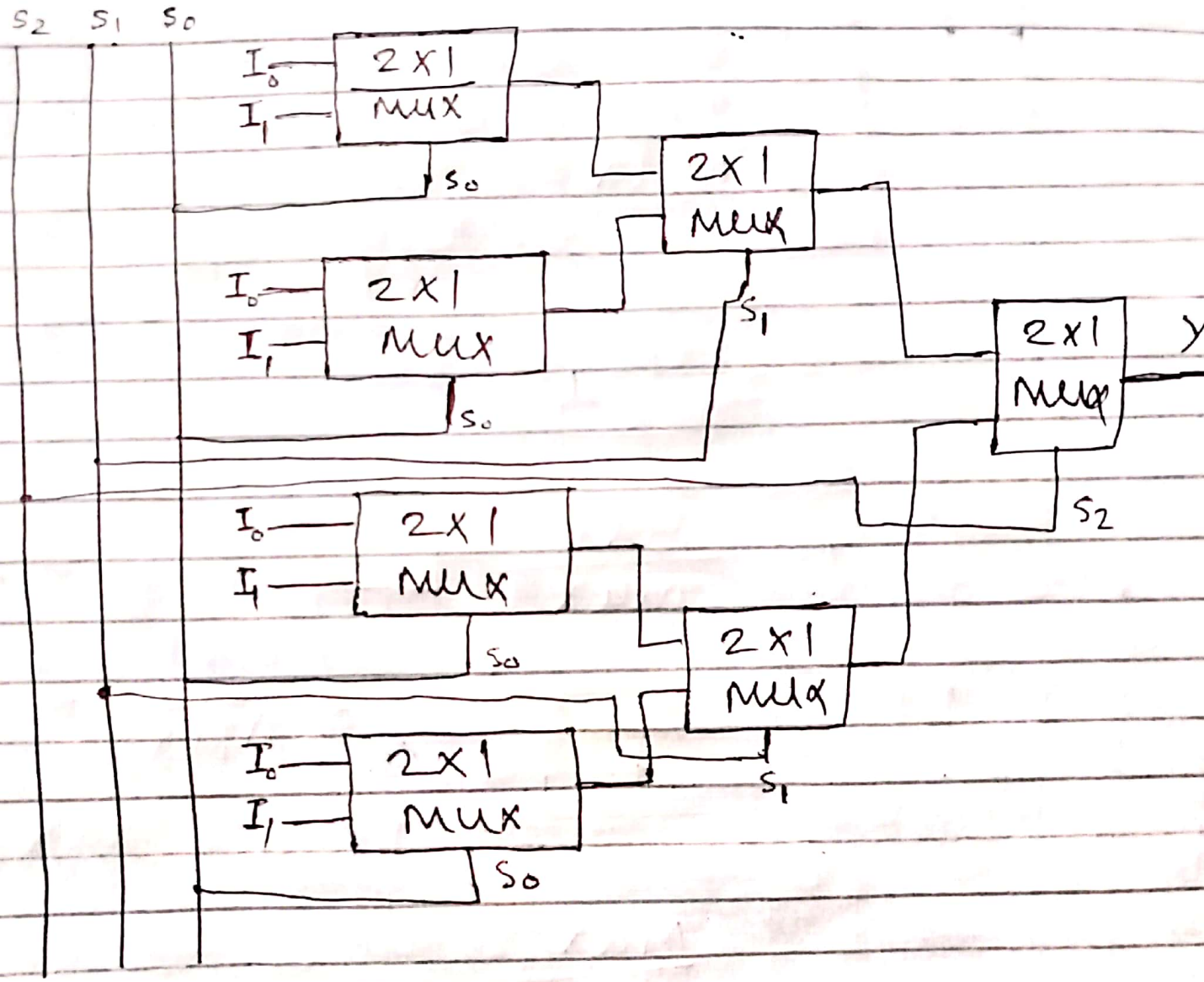

So This is 4x1 Mux using 2x1 Mux

Design 8x1 Multiplexer using 4x1 Multiplexer

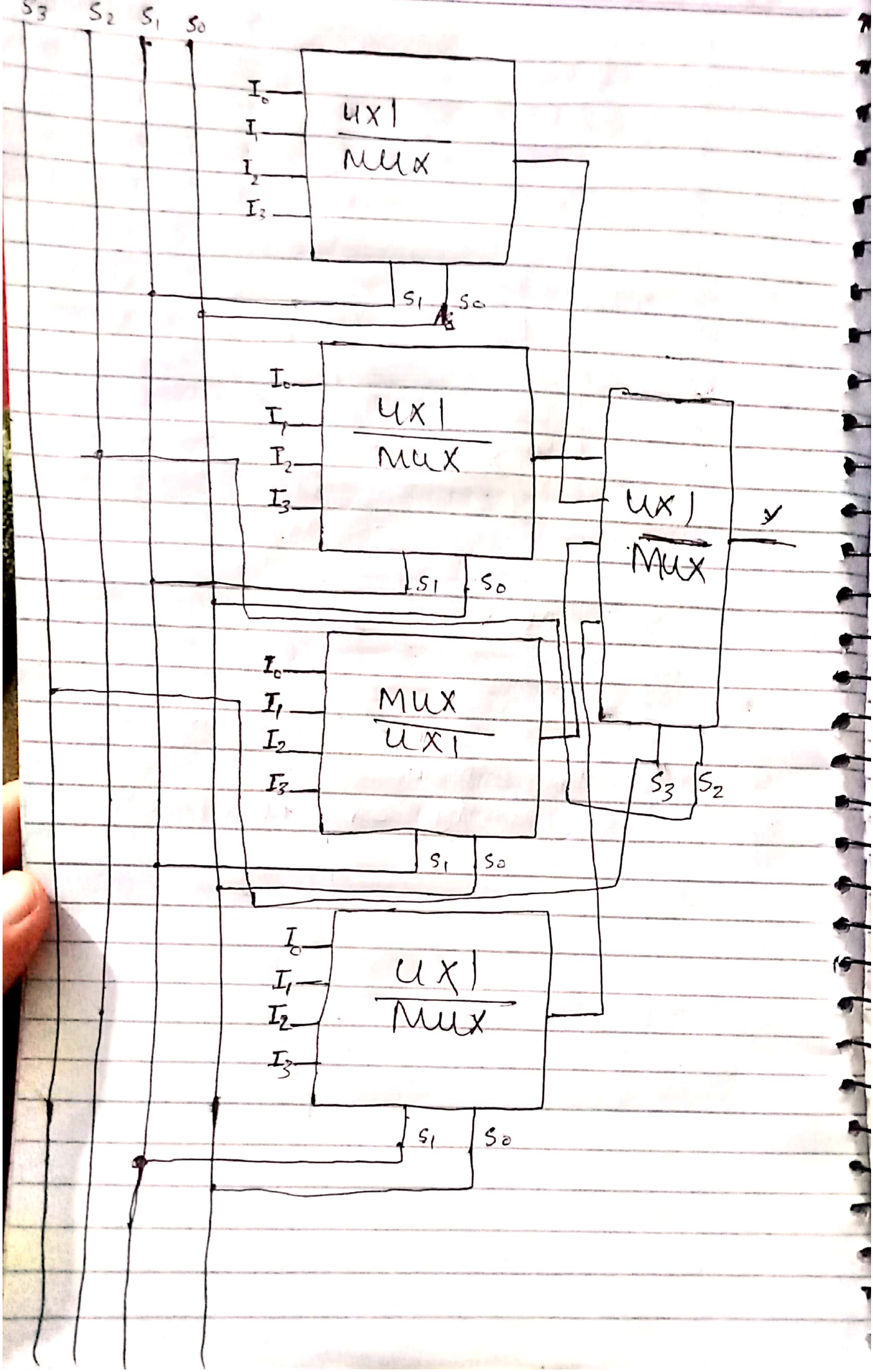



So This is implementation of 8x1 multiplexer using Two 4x1 Multiplexer and on 2x1 Multiplexer

# Design 8x1 using 2x1 Multiplexer




So This The implementation of 8x1 Multiplexer using 2x1 multiplexer.



## implementation of Half Adders using MUX

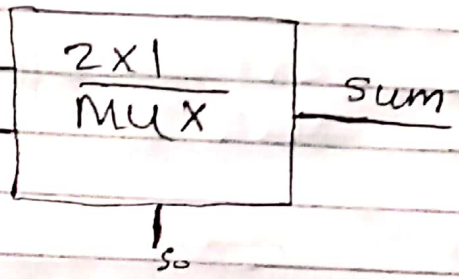
| | A | B | S | C | |
|---|---|---|---|---|---|
| $m_0$ | 0 | 0 | 0 | 0 | $S = \bar{A}B + A\bar{B}$ |
| $m_1$ | 0 | 1 | 1 | 0 | $S = \Sigma(1,2)$ |
| $m_2$ | 1 | 0 | 1 | 0 | $C = AB$ |
| $m_3$ | 1 | 1 | 0 | 1 | $C = \Sigma(3)$ |

$\bar{A}$ 0 (1)

A (2) 3

A $\bar{A}$

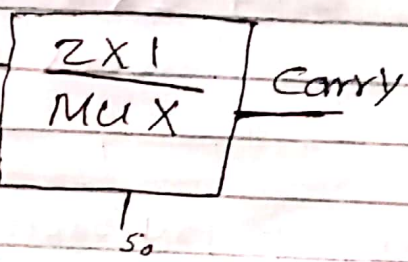
Sum implementation




$\bar{A}$ 0 1

A 2 (3)

0 A




## implementation of Full Adders using Multiplexer

| | A | B | C | S | C | |
|---|---|---|---|---|---|---|
| $m_0$ | 0 | 0 | 0 | 0 | 0 | $Sum = \bar{A}\bar{B}C + \bar{A}B\bar{C} + A\bar{B}\bar{C} + ABC$ |
| $m_1$ | 0 | 0 | 1 | 1 | 0 | $Sum = \Sigma(1,2,4,7)$ |
| $m_2$ | 0 | 1 | 0 | 1 | 0 | $Carry = \bar{A}BC + A\bar{B}C + AB\bar{C} + ABC$ |
| $m_3$ | 0 | 1 | 1 | 0 | 1 | $Carry = \Sigma(3,5,6,7)$ |
| $m_4$ | 1 | 0 | 0 | 1 | 0 | |
| $m_5$ | 1 | 0 | 1 | 0 | 1 | |
| $m_6$ | 1 | 1 | 0 | 0 | 1 | |
| $m_7$ | 1 | 1 | 1 | 1 | 1 | |

$\bar{A}$ 0 (1) (2) 3 Sum

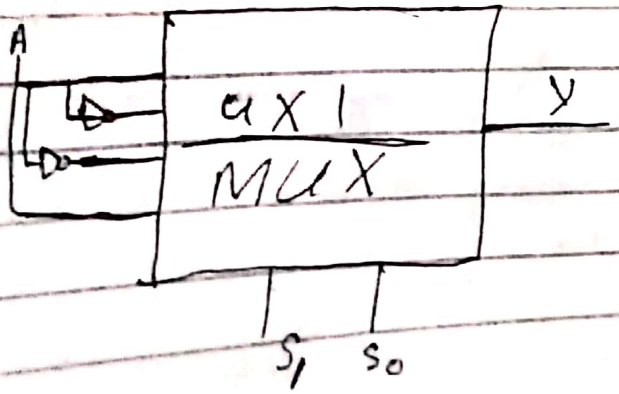
A (4) 5 6 (7)

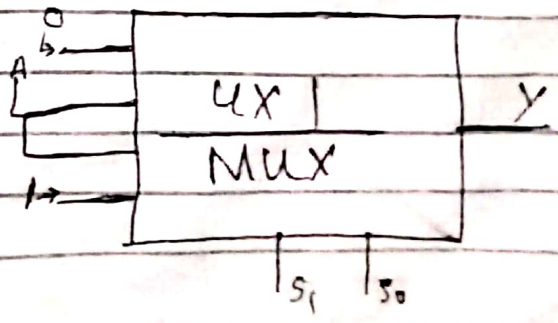
A $\bar{A}$ $\bar{A}$ A

$\bar{A}$ 0 1 2 (3) Carry

A 4 (5) (6) (7)

0 A A 1



## Implementation of Half subtractor using multiplexer

| | A | B | D | Bin |
|---|---|---|---|---|
| $m_0$ | 0 | 0 | 0 | 0 |
| $m_1$ | 0 | 1 | 1 | 1 |
| $m_2$ | 1 | 0 | 1 | 0 |
| $m_3$ | 1 | 1 | 0 | 0 |

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

$D = \Sigma(1, 2)$

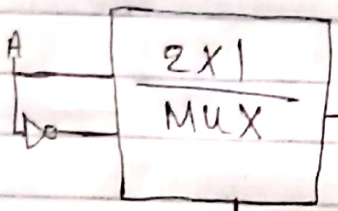
$B_{in} = \bar{A}B$

$B_{in} = \Sigma(1)$

$\bar{A}$ 0 (1)

A (2) 3

A $\bar{A}$

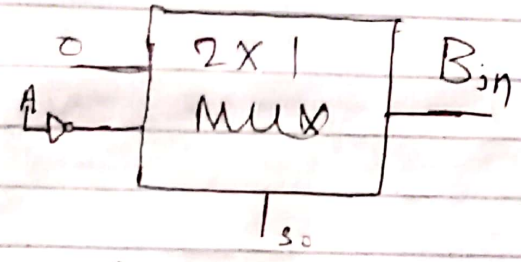



This is Difference

$\bar{A}$ 0 (1)

A 2 3

0 $\bar{A}$




## implementation of Full subtractor using MUX

| | A | B | C | D | Bin |
|---|---|---|---|---|---|
| $m_0$ | 0 | 0 | 0 | 0 | 0 |
| $m_1$ | 0 | 0 | 1 | 1 | 1 |
| $m_2$ | 0 | 1 | 0 | 1 | 1 |
| $m_3$ | 0 | 1 | 1 | 0 | 1 |
| $m_4$ | 1 | 0 | 0 | 1 | 0 |
| $m_5$ | 1 | 0 | 1 | 0 | 0 |
| $m_6$ | 1 | 1 | 0 | 0 | 0 |
| $m_7$ | 1 | 1 | 1 | 1 | 1 |

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

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

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

$B_{in} = \Sigma(1, 2, 3, 7)$

$\bar{A}$ 0 (1) (2) 3 sum

A (4) 5 6 (7)

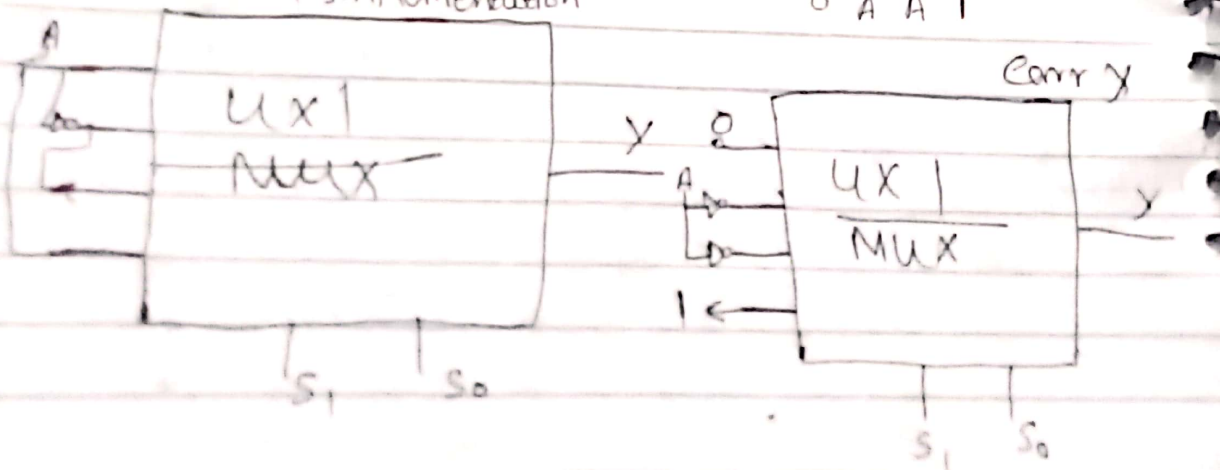
A $\bar{A}$ $\bar{A}$ A

$\bar{A}$ 0 (1) (2) (3) carry

A 4 5 6 (7)

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

Sum implementation

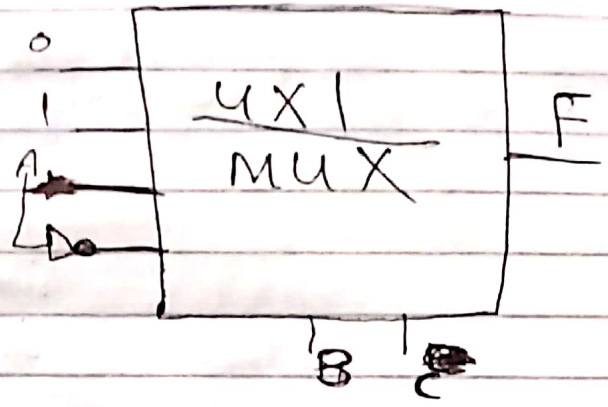


## Implementation of Boolean function using MUX

$F(A,B,C) = \Sigma(1,3,5,6)$ This is three variable Function

| | A | B | C | F |
|---|---|---|---|---|
| $m_0$ | 0 | 0 | 0 | 0 |
| $m_1$ | 0 | 0 | 1 | 1 |
| $m_2$ | 0 | 1 | 0 | 0 |
| $m_3$ | 0 | 1 | 1 | 1 |
| $m_4$ | 1 | 0 | 0 | 0 |
| $m_5$ | 1 | 0 | 1 | 1 |
| $m_6$ | 1 | 1 | 0 | 1 |
| $m_7$ | 1 | 1 | 1 | 0 |

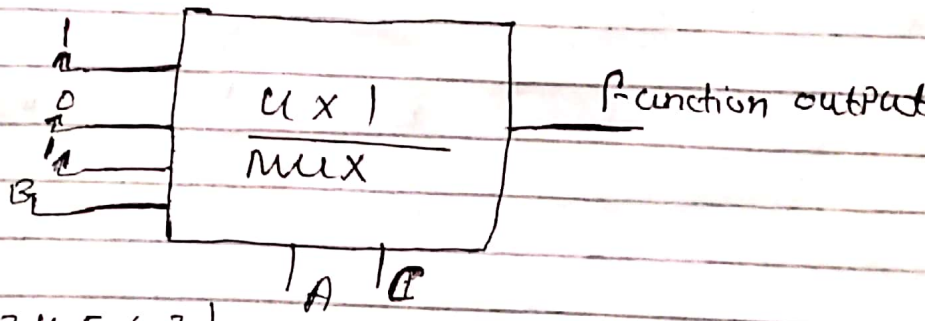
| $\bar{A}$ | 0 | (1) | 2 | (3) |
|---|---|---|---|---|
| A | 4 | (5) | (6) | 7 |
| | 0 | 1 | A | $\bar{A}$ |




$F(A,B,C) = \Sigma(0,2,4,6,7)$

| | A | B | C | F |
|---|---|---|---|---|
| $m_0$ | 0 | 0 | 0 | 1 |
| $m_1$ | 0 | 0 | 1 | 0 |
| $m_2$ | 0 | 1 | 0 | 1 |
| $m_3$ | 0 | 1 | 1 | 0 |
| $m_4$ | 1 | 0 | 0 | 1 |
| $m_5$ | 1 | 0 | 1 | 0 |
| $m_6$ | 1 | 1 | 0 | 1 |
| $m_7$ | 1 | 1 | 1 | 1 |

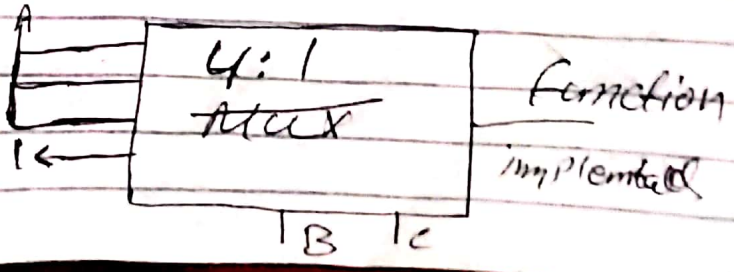
| $\bar{B}$ | (0) | 1 | (4) | 5 |
|---|---|---|---|---|
| B | (2) | 3 | (6) | (7) |
| | 1 | 0 | 1 | B |




$F(A,B,C) = \Sigma(3,4,5,6,7)$

| | A | B | C | F |
|---|---|---|---|---|
| $m_0$ | 0 | 0 | 0 | 0 |
| $m_1$ | 0 | 0 | 1 | 0 |
| $m_2$ | 0 | 1 | 0 | 0 |
| $m_3$ | 0 | 1 | 1 | 1 |
| $m_4$ | 1 | 0 | 0 | 1 |
| $m_5$ | 1 | 0 | 1 | 1 |
| $m_6$ | 1 | 1 | 0 | 1 |
| $m_7$ | 1 | 1 | 1 | 1 |

| $\bar{A}$ | 0 | 1 | 2 | (3) |
|---|---|---|---|---|
| A | (4) | (5) | (6) | (7) |
| | A | A | A | 1 |



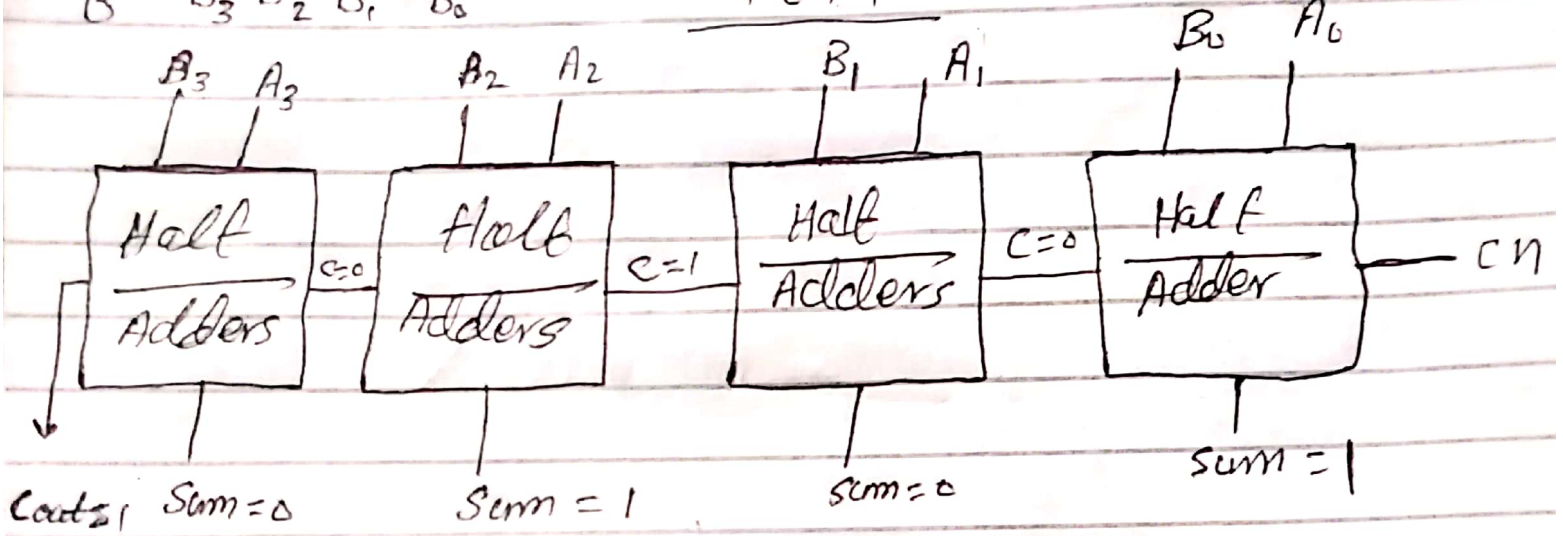
Adders we have already done with Full and Half Adders Now we want to design 4-bit binary Adders

A $A_3$ $A_2$ $A_1$ $A_0$

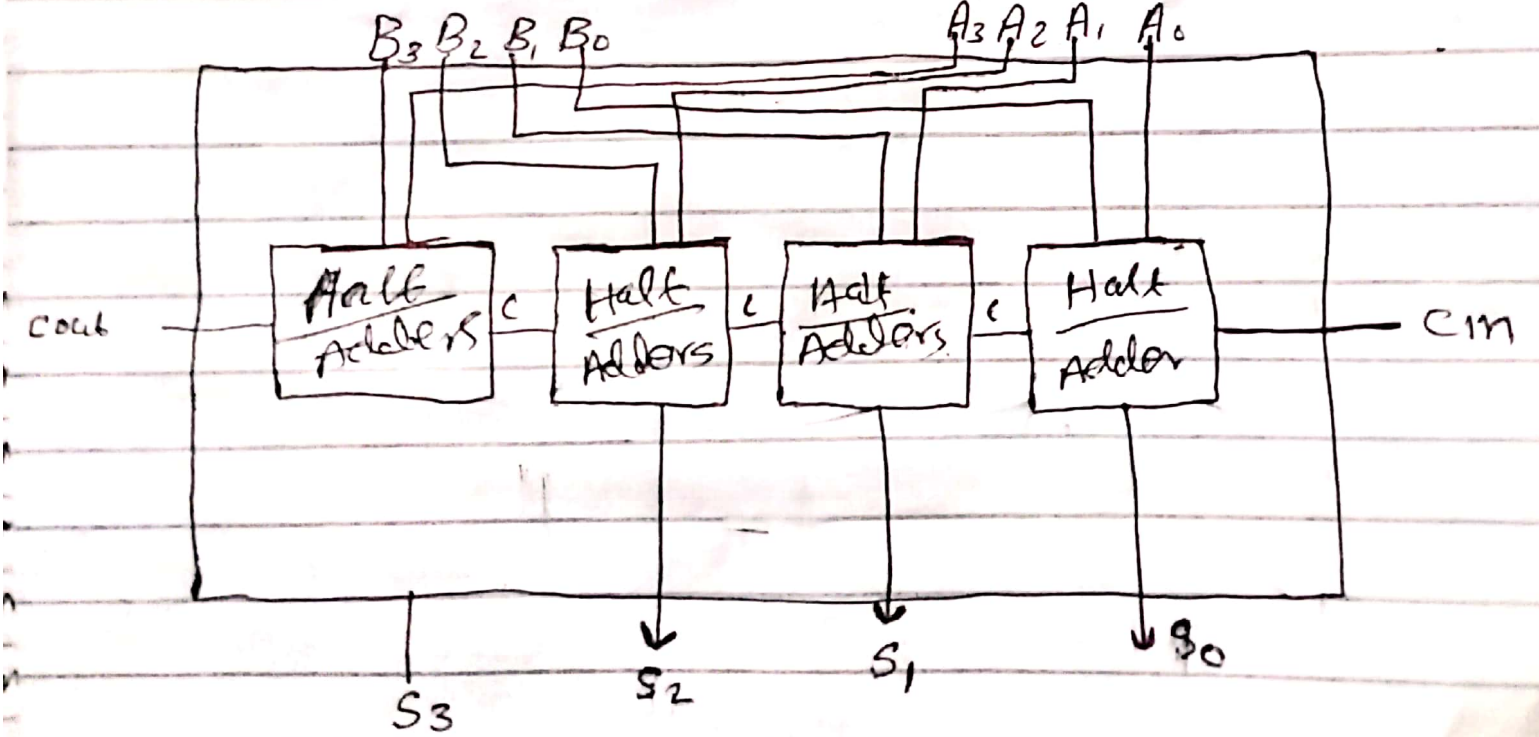
B $B_3$ $B_2$ $B_1$ $B_0$

$A_3$ $A_2$ $A_1$ $A_0$
0 1 0

$B_3$ $B_2$ $B_1$ $B_0$
0 1 1




we can do This in another way



Design 16-bit Adders using 4-bit Adders

Design 4-bit Multiplier using 4-bit binary Adders

A

B

$A_3\ A_2\ A_1\ A_0$

$B_3\ B_2\ B_1\ B_0$

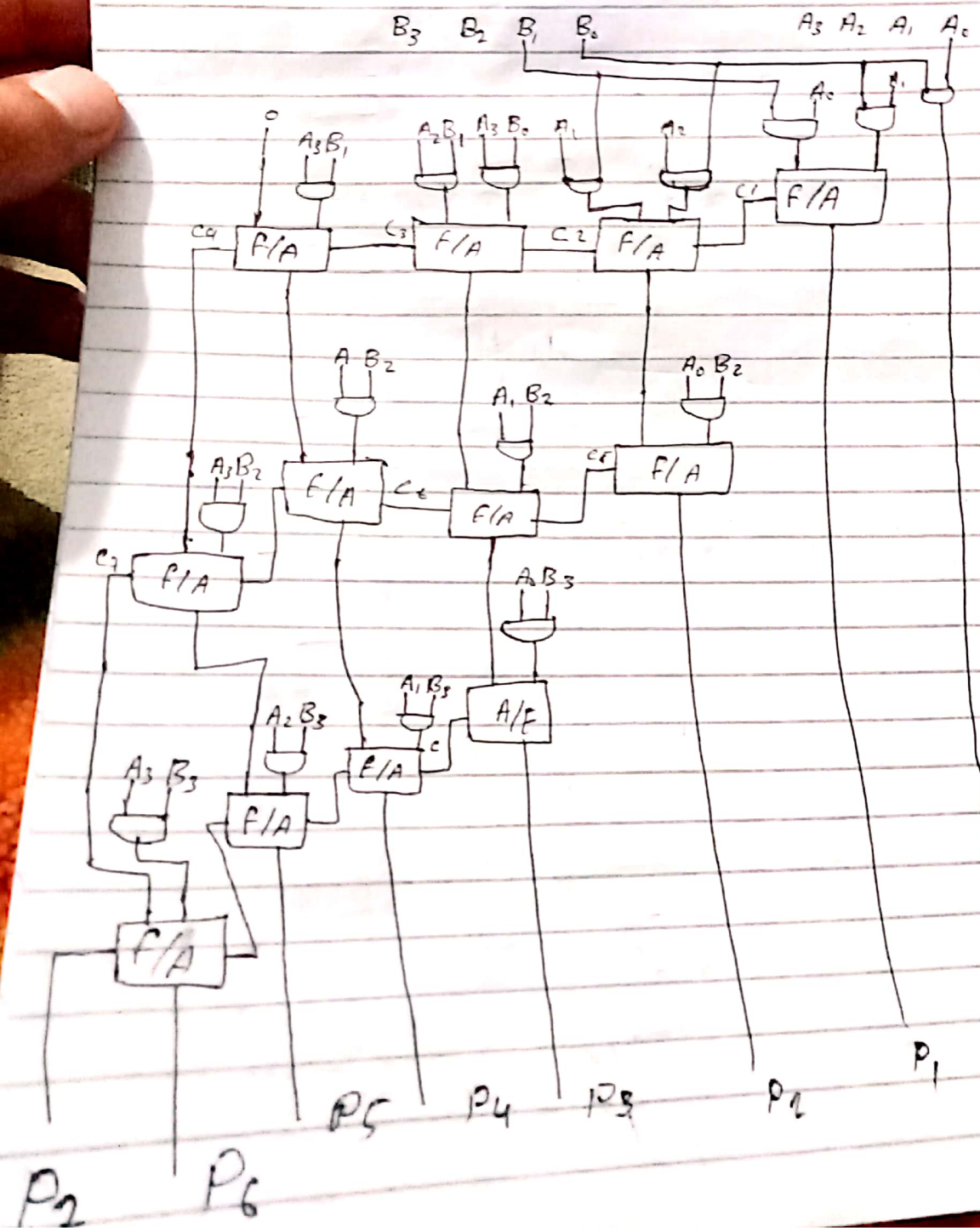
$A_3B_0\ \ A_2B_0\ \ A_1B_0\ \ A_0B_0$

$A_3B_1\ \ A_2B_1\ \ A_1B_1\ \ A_0B_1\ \ X$

$A_3B_2\ \ A_2B_2\ \ A_1B_2\ \ A_0B_2\ \ X\ \ X$

$A_3B_3\ \ A_2B_3\ \ A_1B_3\ \ A_0B_3\ \ X\ \ X\ \ X$

$P_7\ P_6\ P_5\ P_4\ P_3\ P_2\ P_1\ P_0$




Design

Note

1-bit

The

# Design 4-bit subtractor using 4-bit Adders

A $A_3\ A_2\ A_1\ A_0$

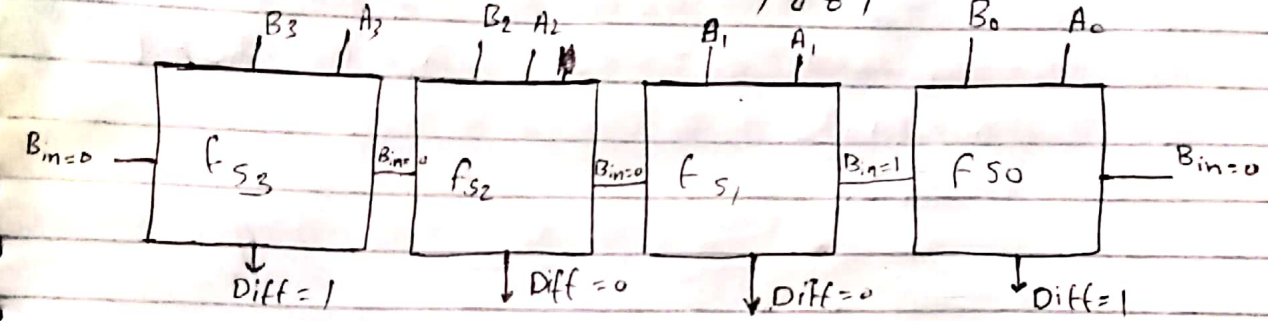
B $B_3\ B_2\ B_1\ B_0$

$D_3\ D_2\ D_1\ D_0$

A $A_3\ A_2\ A_1\ A_0$ = 1 1 1 0

B $B_3\ B_2\ B_1\ B_0$ = − 0 1 0 1

1 0 0 1



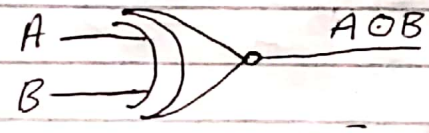

Note of magnitude comparator

1-bit magnitude comparator

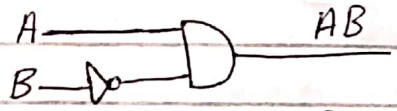
Truth Table

| 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 |

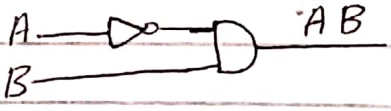
$A == B$



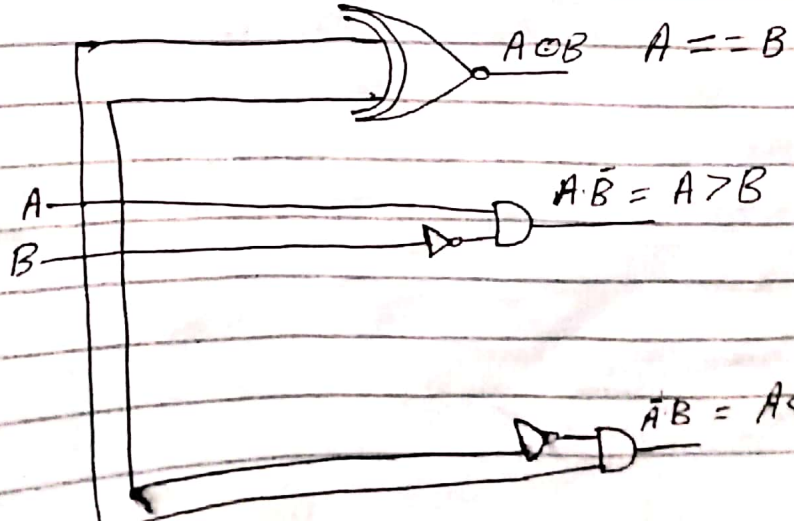

$A > B$




$A < B$




The overall circuit for 1-bit comparator



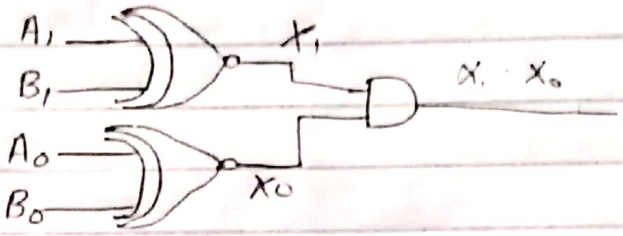
2-bit magnitude comparator $A == B$

$A == B$ A $A_0$ $A_1$

B $B_0$ $B_1$

if $A_1 == B_1$ and $A_0 == B_0$ $X_1 = 1$ $X_0 = 1$

$(X_1 \cdot X_0)$

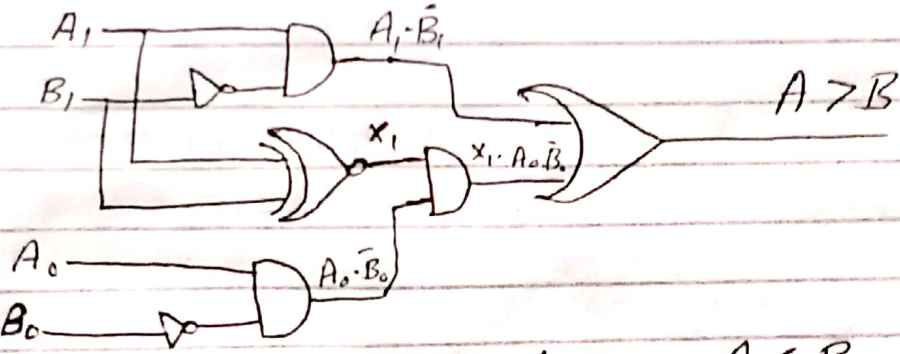



2-bit magnitude comparator $A > B$

A $A_0$ $A_1$

B $B_0$ $B_1$

if $A_1 == 1$ and $B_1 == 0$ or $A_1 == B_1$ and $A_0 == 1$ and $B_0 == 0$ $(A_1 \bar{B}_1)$ or $(X_1 \cdot A_0 \cdot \bar{B}_0)$

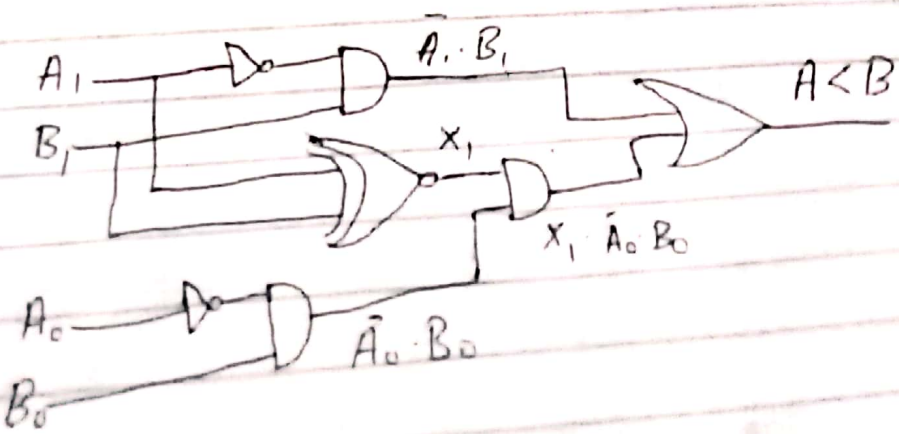



2-bit magnitude comparator $A < B$

A $A_0$ $A_1$

B $B_0$ $B_1$

if $A_1 == 0$ and $B_1 == 1$ or $A_1 == B_1$ and $A_0 == 0$ and $B_0 == 1$ $(\bar{A}_1 \cdot B_1)$ or $(X_1 \cdot \bar{A}_0 \cdot B_0)$



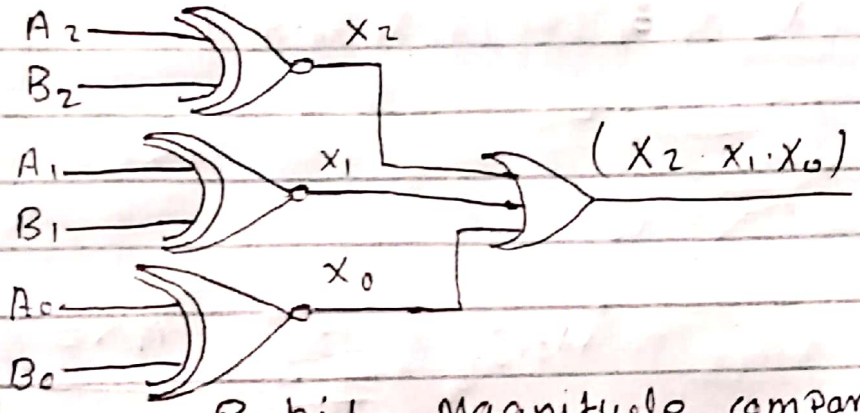
3-bit magnitude comparator $A == B$

A $A_0\ A_1\ A_2$

B $B_0\ B_1\ B_2$

if $A_2 == B_2$ and $A == B_1$ and $A_0 == B_0$

$(X_2 = 1 \cdot X_1 = 1 \quad X_0 = 1) \Rightarrow (X_2 \cdot X_1 \cdot X_0)$



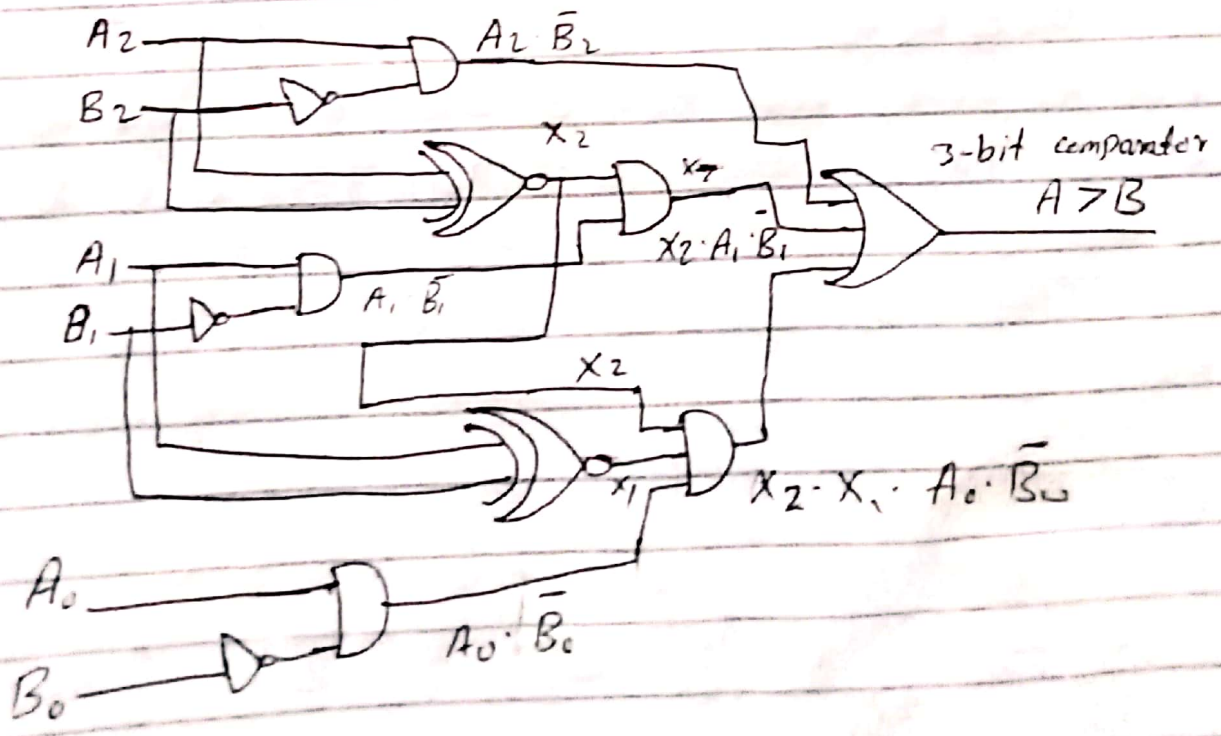

3-bit Magnitude comparator $A \not> B$

A $A_0\ A_1\ A_2$

B $B_0\ B_1\ B_2$

if $A_2 == \phi$ and $B_2 == 0$ or $A_2 == B_2$ and $A_1 == 1$ and $B_1 = 0$ or $A_1 == B_1$ and $A_0 == 1$ and $B_0 == 0$

$(A_2 \cdot \bar{B_2})$ or $(X_2 \cdot A_1 \cdot \bar{B_2})$ or $(X_2 \cdot X_1 \cdot A_0 \cdot \bar{B_0})$



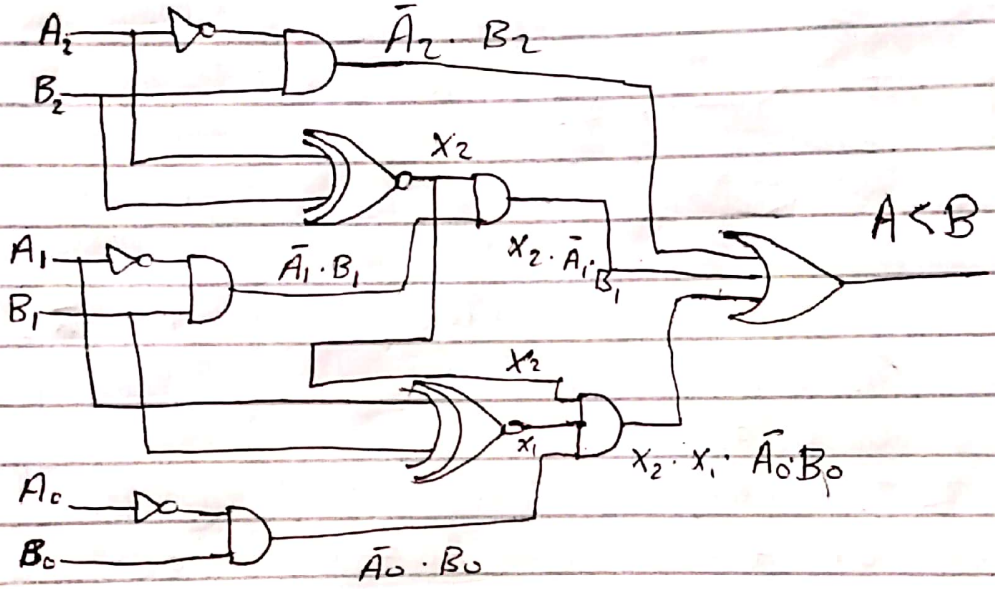
# 3-bit magnitude comparator $A<B$

A $A_0\ A_1\ A_2$

B $B_0\ B_1\ B_2$

if $A_2==0$ and $B_2==1$ or $A_2==B_2$ and $A_1==0$ and $B_1==1$ or $A_1==B_1$ and $A_0==0$ and $B_0==1$

$(\bar{A}_2 \cdot B_2)$ or $(x_2 \cdot \bar{A}_1 \cdot B_1)$ or $(x_2 \cdot x_1 \cdot \bar{A}_0 \cdot A_1)$



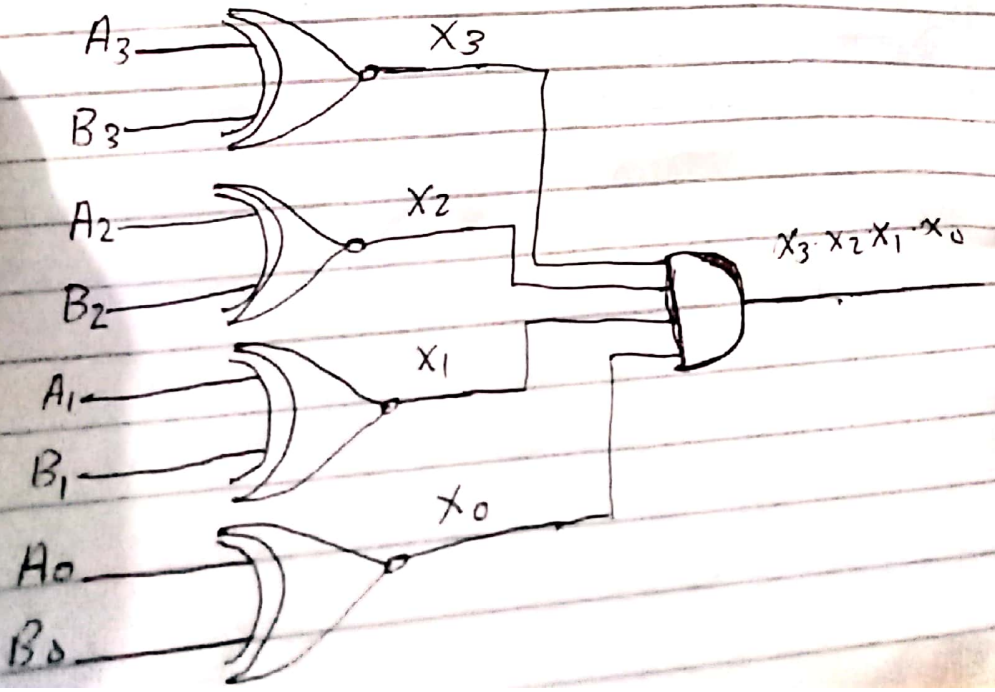

# 4-bit magnitude comparator $A==B$

A $A_0\ A_1\ A_2\ A_3$

B $B_0\ B_1\ B_2\ B_3$

IF $A_3==B_3$ and $A_2==B_2$ and $A_1==B_1$ and $A_0==B_0$

so it's $(x_3=1,\ x_2=1,\ x_1=1,\ x_0=1,) \Rightarrow (x_3 \cdot x_2 \cdot x_1 \cdot x_0)$



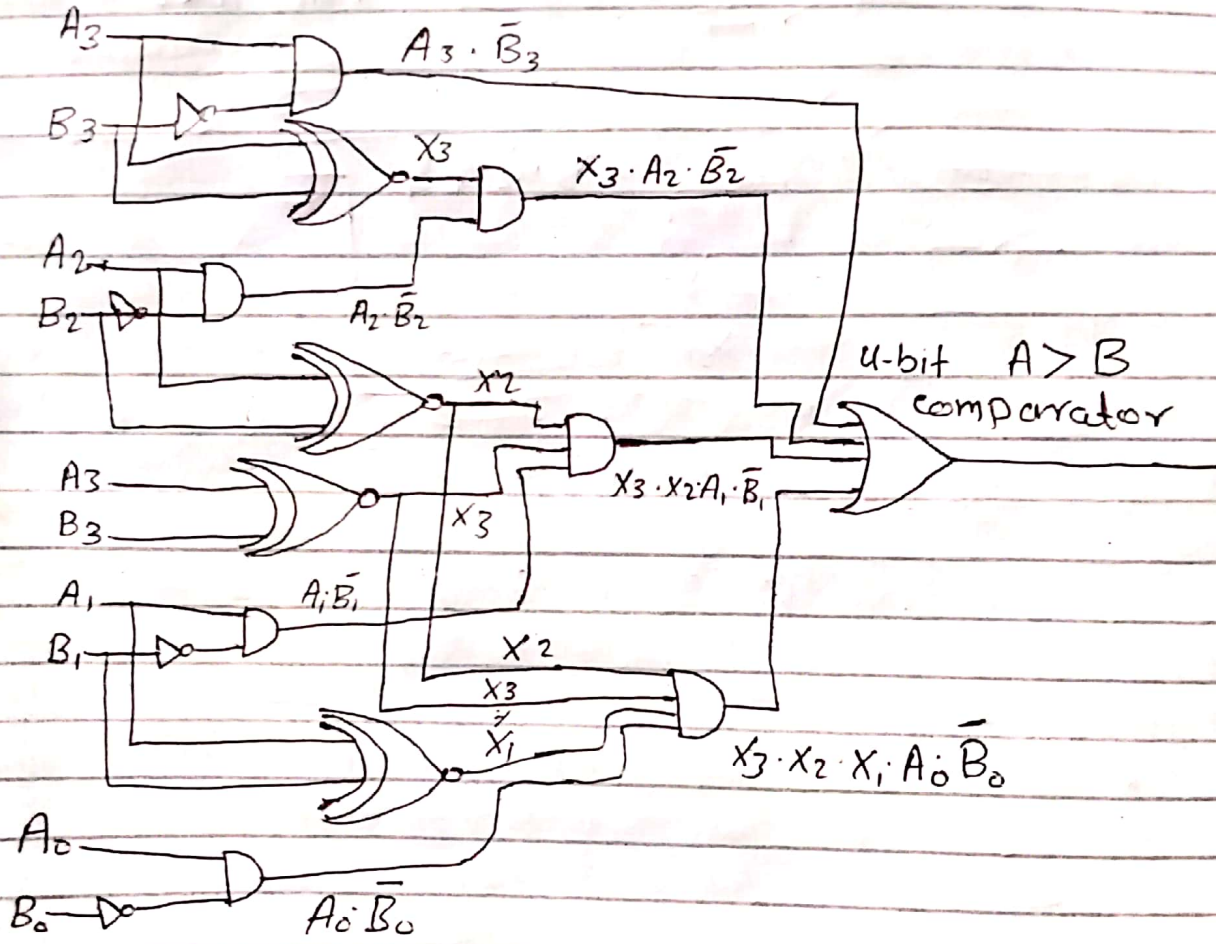
4-bit comparator $A < B$

A $A_0\ A_1\ A_2\ A_3$

B $B_0\ B_1\ B_2\ B_3$

if $A_3 = 1$ and $B_3 = 0$ or $A_3 == B_3$ and $A_2 = 1$ and $B_2 = 0$ or $A_2 == B_2$ and $A_1 = 1$ and $B_1 = 0$ or $A_1 == B_1$ and $A_0 == 1$ and $B_0 = 0$

$(A_3 \cdot \bar{B_3})$ or $(x_3 \cdot A_2 \cdot \bar{B_2})$ or $(x_3 \cdot x_2 \cdot A_1 \cdot \bar{B_1})$ or $(x_3 \cdot x_2 x_1 A_0 \cdot \bar{B_0})$




4-bit magnitude comparator $A < B$

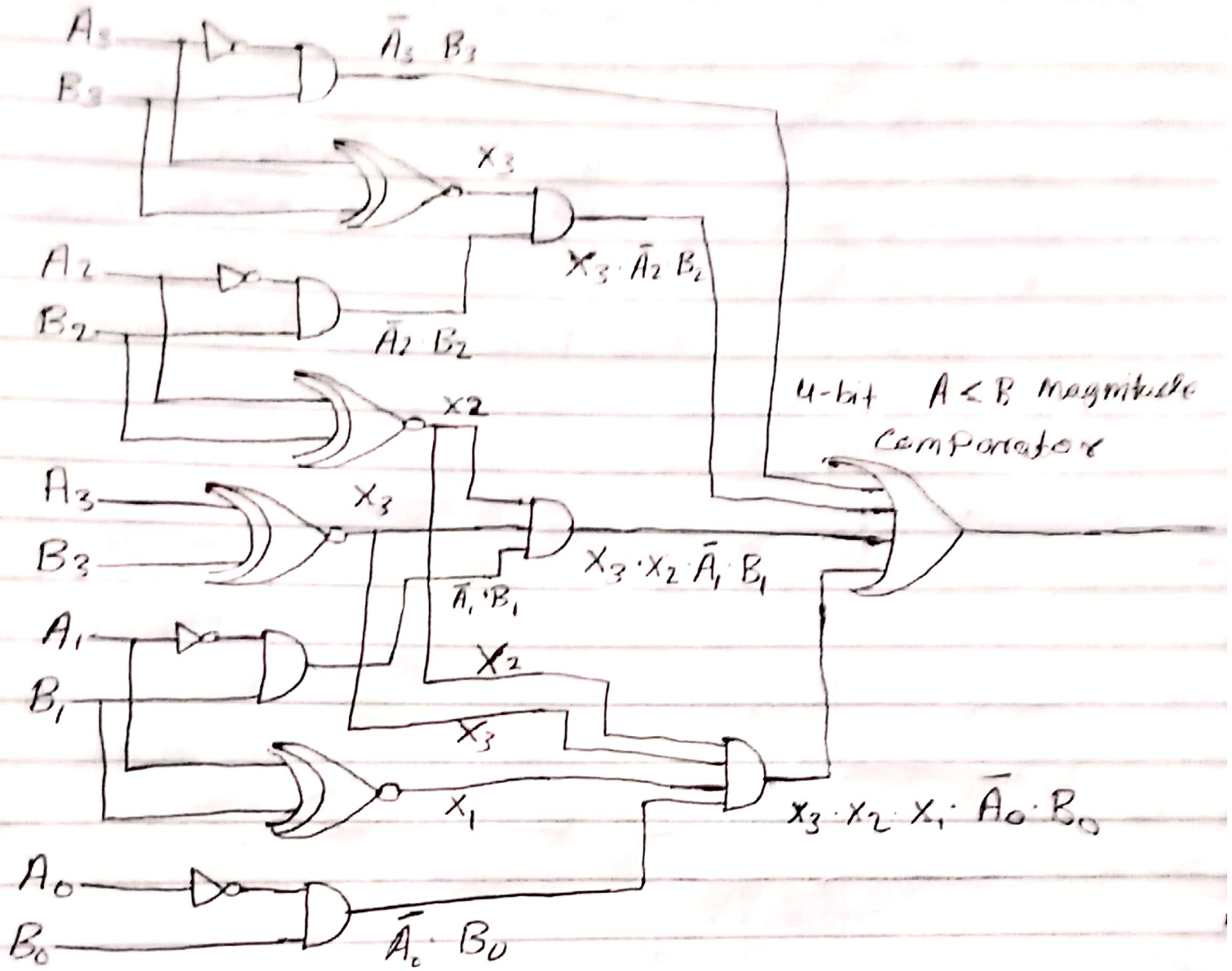
A $A_0\ A_1\ A_2\ A_3$

B $B_0\ B_1\ B_2\ B_3$

if $A_3 = 0$ and $B_3 == 1$ or $A_3 == B_3$ and $A_2 == 0$ and $B_2 == 1$ or $A_2 == B_2$ and $A_1 == 0$ and $B_1 == 1$ or $A_1 == B_1$ and $A_0 == 0$ and $B_0 == 1$

$(\bar{A_3} \cdot B_3)$ or $(x_3 \cdot \bar{A_2} \cdot B_2)$ or $(x_3 x_2 \cdot \bar{A_1} \cdot B_1)$ or $(x_3 x_2 x_1 \cdot \bar{A_0} \cdot B_0)$

The diagram for A<B magnitude comparator

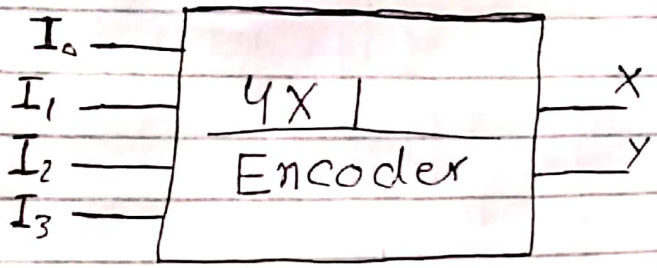


# Encoder

design 4 x 2 Encoder

So Encoder a combinational circuit Take n and ↓ only Two output at time.

Produce balck diagram



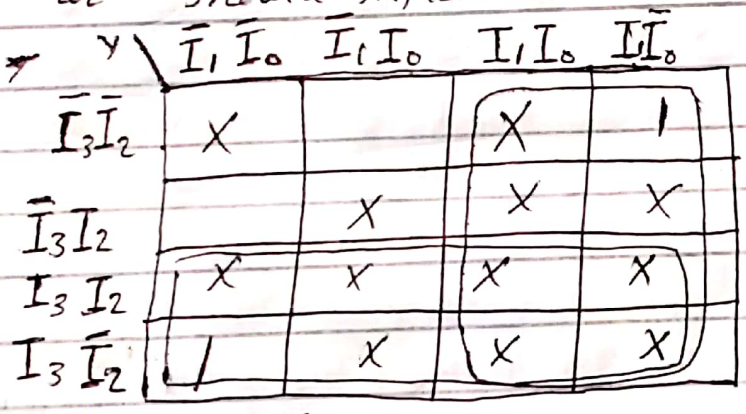

| | $I_3$ | $I_2$ | $I_1$ | $I_0$ | X | Y |
|---|---|---|---|---|---|---|
| $m_1$ | 0 | 0 | 0 | 1 | 0 | 0 |
| $m_2$ | 0 | 0 | 1 | 0 | 0 | 1 |
| $m_4$ | 0 | 1 | 0 | 0 | 1 | 0 |
| $m_8$ | 1 | 0 | 0 | 0 | 1 | 1 |

$y = \Sigma(2,8)$

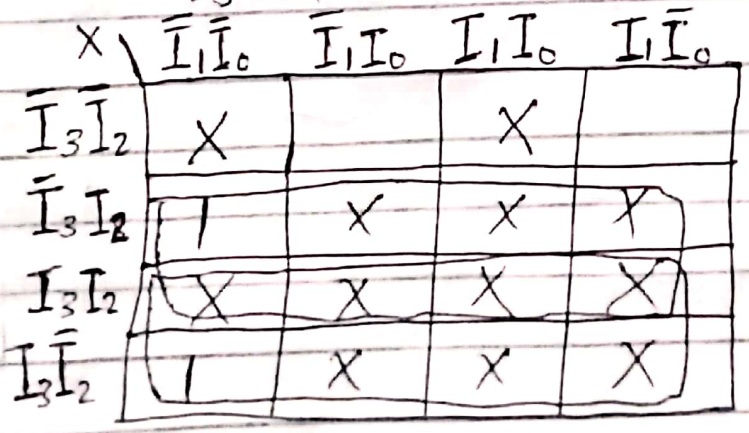
$x = \Sigma(4,8)$

$d = \Sigma(0,3,5,6,7,9,10,11,12,13,14,15)$

So we should implement Them in K-map

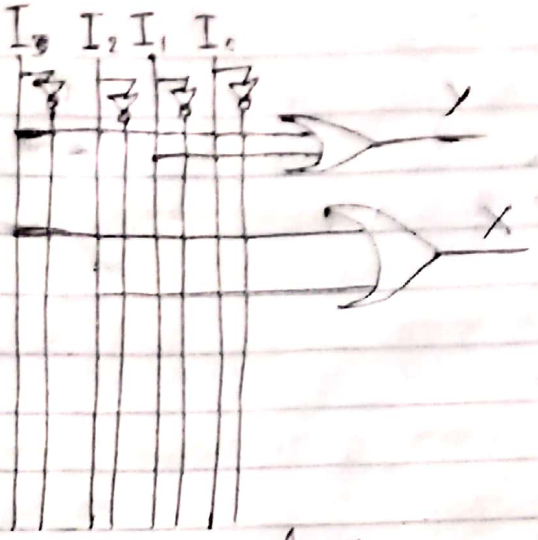



$I_3 + I_1$




$I_2 + I_3$

diagram for 4x2 Encoder



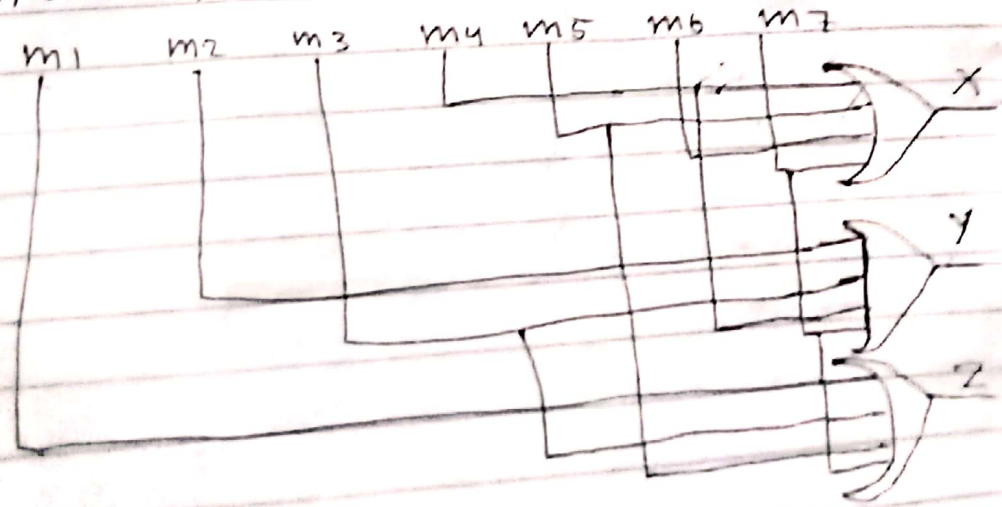

Design 8 x 3 Encoder

| | $I_0$ | $I_1$ | $I_2$ | $I_3$ | $I_4$ | $I_5$ | $I_6$ | $I_7$ | X | Y | Z |
|---|---|---|---|---|---|---|---|---|---|---|---|
| $m_0$ | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| $m_1$ | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |
| $m_2$ | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 |
| $m_3$ | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 1 |
| $m_4$ | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 |
| $m_5$ | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 1 |
| $m_6$ | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 1 | 0 |
| $m_7$ | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 |

$X = \Sigma(4, 5, 6, 7) \Rightarrow m_4 + m_5 + m_6 + m_7$

$Y = \Sigma(2, 3, 6, 7) \Rightarrow m_2 + m_3 + m_6 + m_7$

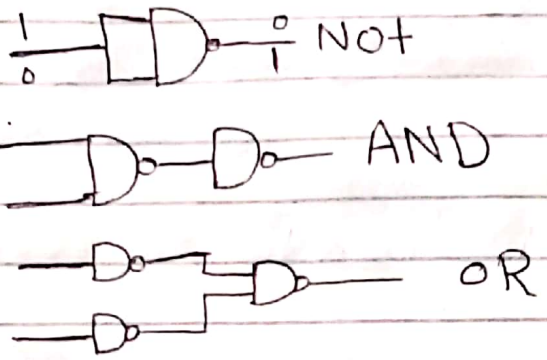
$Z = \Sigma(1, 3, 5, 7) \Rightarrow m_1 + m_3 + m_5 + m_7$



Flip-flops

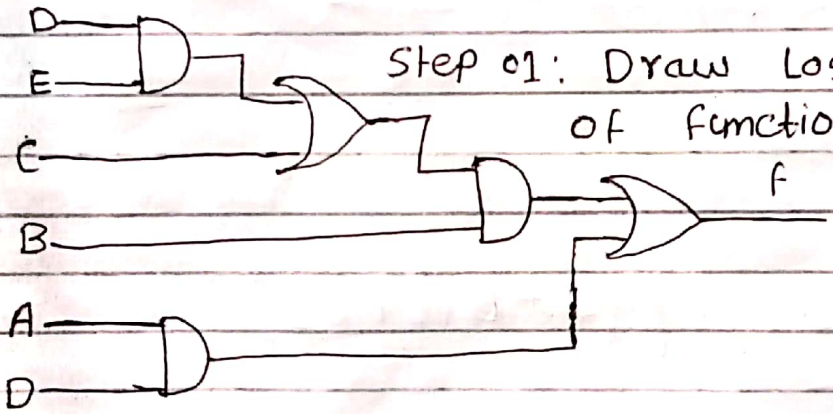
Implementation of function using NAND and NOR Gates

Frist NAND Gate:



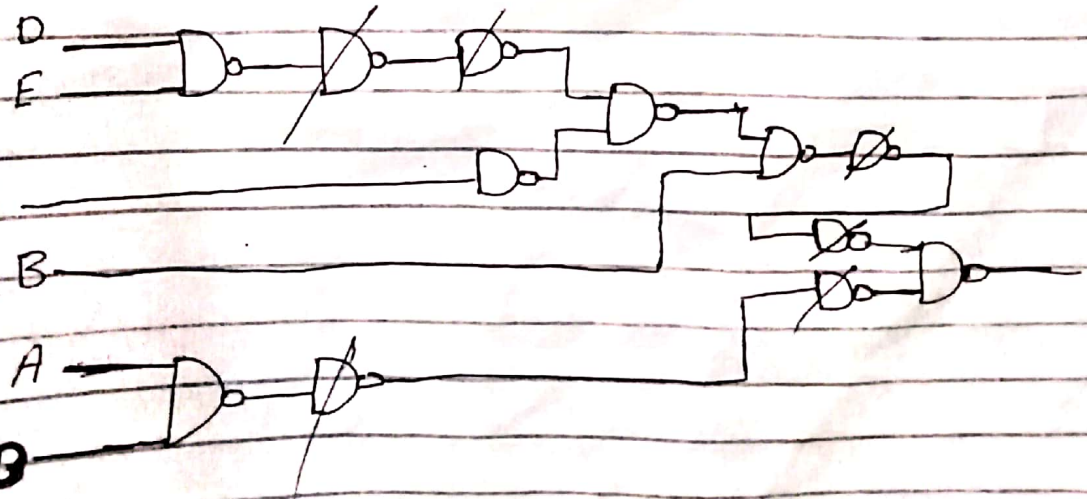

Q: Design the function:

$F = AD + B(C + DE)$

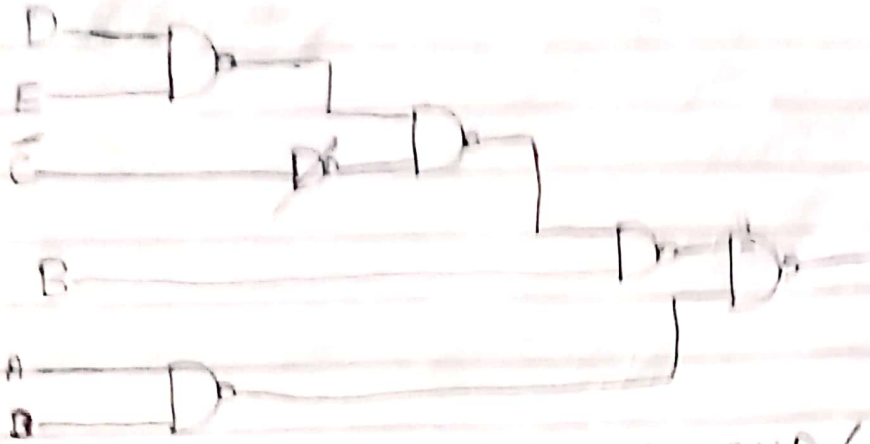



Step 01: Draw Logic diagram of function

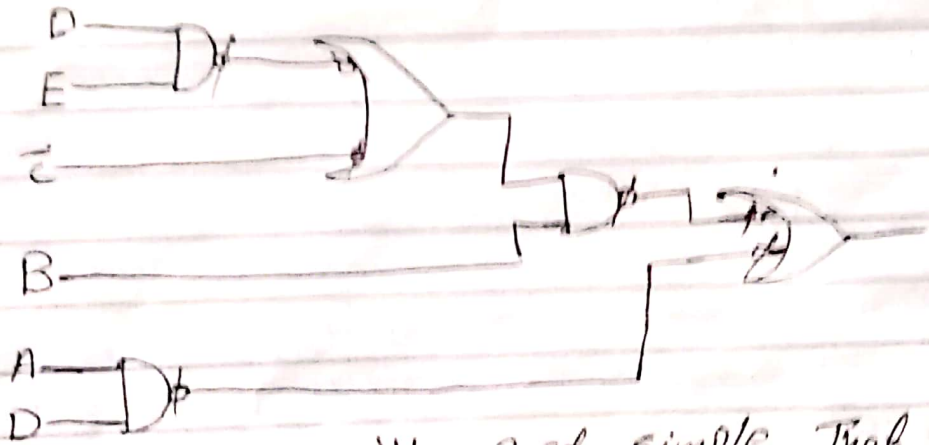
Step 02: Replace equvalent gates by NAND



Step 03 Simplification

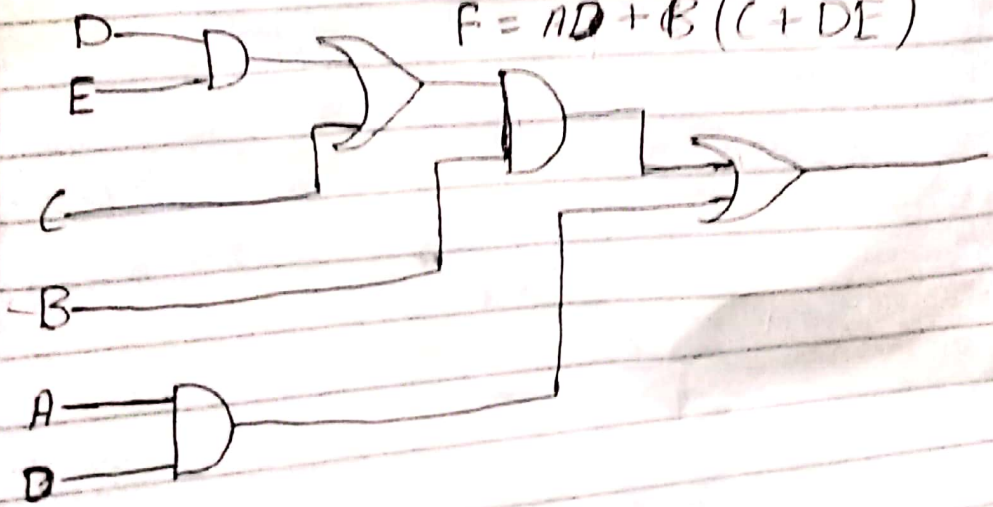



Reversing NAND to AND/OR/NOT



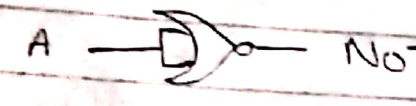

So will get simple That we start form

$F = AD + B(C + DE)$

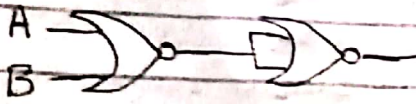


# Implementation of function using NOR gate

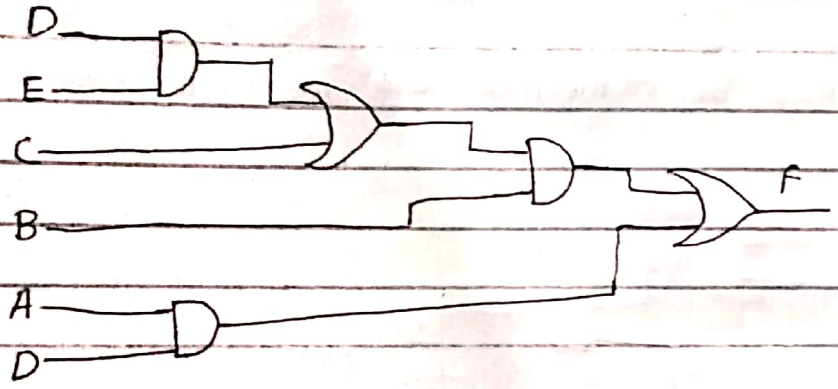
Q: $F = AB + B(C + DE)$



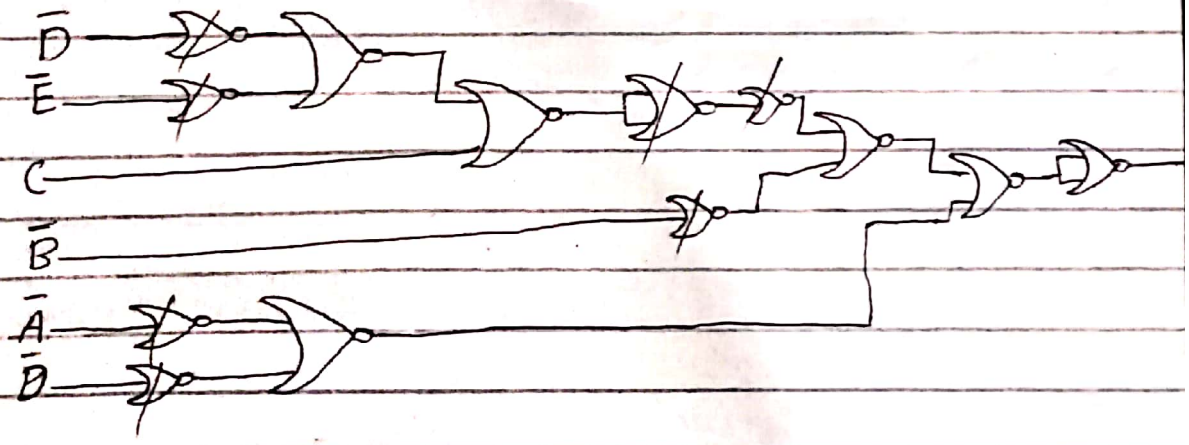




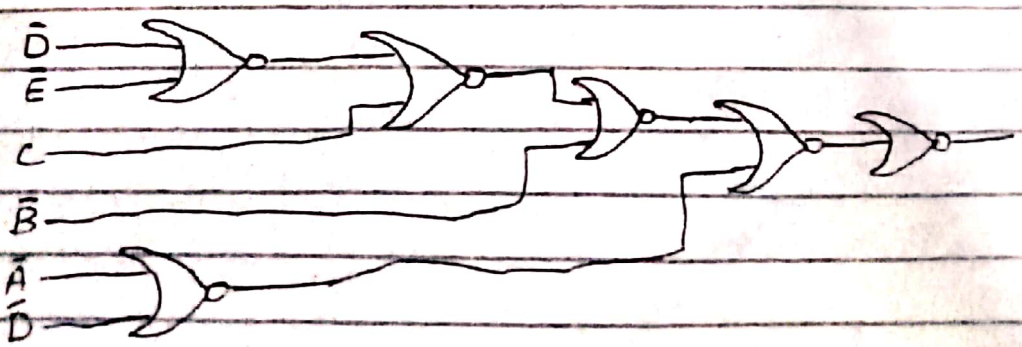

Step1: Draw logic diagram for the function




Step2: Replace Gate by NOR Gate.




simplification:

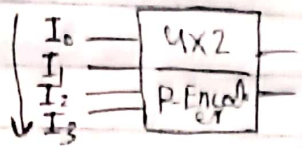


# Priority Encoder

Design 4 x 2 priority Encoder.

| $I_3$ | $I_2$ | $I_1$ | $I_0$ | X | Y | V |
|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 1 | 0 | 0 | 1 |
| 0 | 0 | 1 | X | 0 | 1 | 1 |
| 0 | 1 | X | X | 1 | 0 | 1 |
| 1 | X | X | X | 1 | 1 | 1 |

block diagram




| X | $\bar{I}_1\bar{I}_0$ | $\bar{I}_1 I_0$ | $I_1 I_0$ | $I_1\bar{I}_0$ |
|---|---|---|---|---|
| $\bar{I}_3\bar{I}_2$ | | | | |
| $\bar{I}_3 I_2$ | 1 | 1 | 1 | 1 |
| $I_3 I_2$ | 1 | 1 | 1 | 1 |
| $I_3\bar{I}_2$ | 1 | 1 | 1 | 1 |

$X = I_3 + I_2$

$I_0 < I_1 < I_2 < I_3$

So The Priority of input in Priority Encoder

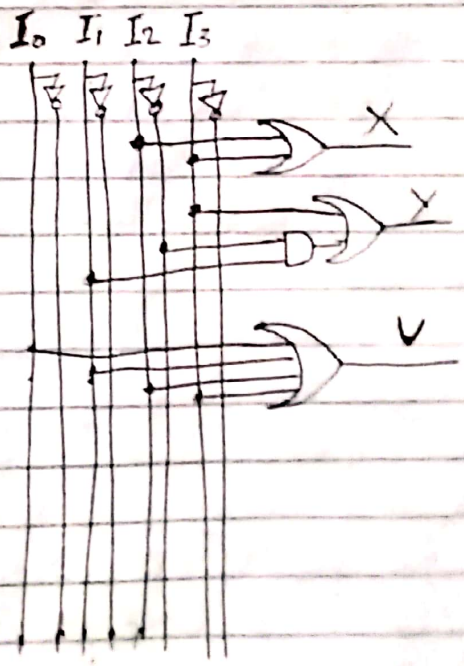
| Y | $\bar{I}_1\bar{I}_0$ | $\bar{I}_1 I_0$ | $I_1 I_0$ | $I_1\bar{I}_0$ |
|---|---|---|---|---|
| $\bar{I}_3\bar{I}_2$ | | | 1 | 1 |
| $\bar{I}_3 I_2$ | | | | |
| $I_3 I_2$ | 1 | 1 | 1 | 1 |
| $I_3\bar{I}_2$ | 1 | 1 | 1 | 1 |

$Y = I_3 + \bar{I}_2 I_1$

| V | $\bar{I}_1\bar{I}_0$ | $\bar{I}_1 I_0$ | $I_1 I_0$ | $I_1\bar{I}_0$ |
|---|---|---|---|---|
| $\bar{I}_3\bar{I}_2$ | 1 | 1 | 1 | 1 |
| $\bar{I}_3 I_2$ | 1 | 1 | 1 | 1 |
| $I_3 I_2$ | 1 | 1 | 1 | 1 |
| $I_3\bar{I}_2$ | 1 | 1 | 1 | 1 |

$I_3 + I_2 + I_0 + I_1$

$V = I_0 + I_1 + I_2 + I_3$



Q2 Draw the turth table for 8x3 priority Encoder

| $I_7$ | $I_6$ | $I_5$ | $I_4$ | $I_3$ | $I_2$ | $I_1$ | $I_0$ | X | Y | Z | V |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 |
| 0 | 0 | 0 | 0 | 0 | 0 | 1 | X | 0 | 0 | 1 | 1 |
| 0 | 0 | 0 | 0 | 0 | 1 | X | X | 0 | 1 | 0 | 1 |
| 0 | 0 | 0 | 0 | 1 | X | X | X | 0 | 1 | 1 | 1 |
| 0 | 0 | 0 | 1 | X | X | X | X | 1 | 0 | 0 | 1 |
| 0 | 0 | 1 | X | X | X | X | X | 1 | 0 | 1 | 1 |
| 0 | 1 | X | X | X | X | X | X | 1 | 1 | 0 | 1 |
| 1 | X | X | X | X | X | X | X | 1 | 1 | 1 | 1 |

$X = \bar{I}_7 \bar{I}_6 \bar{I}_5 I_4 + \bar{I}_7 \bar{I}_6 I_5 + \bar{I}_7 I_6 + I_7$

$X = \bar{I}_6 \bar{I}_5 + I_4 + \bar{I}_6 I_5 + I_6 + I_7$

$X = \bar{I}_5 + I_4 + I_5 + I_6 + I_7$

$X = + I_4 + I_5 + I_6 + I_7$

$y = \bar{I}_7 \bar{I}_6 \bar{I}_5 \bar{I}_4 \bar{I}_3 I_2 + \bar{I}_7 \bar{I}_6 \bar{I}_5 \bar{I}_4 I_3 + \bar{I}_7 I_6 + I_7$

$y = \bar{I}_6 \bar{I}_5 \bar{I}_4 \bar{I}_3 I_2 + \bar{I}_6 \bar{I}_5 \bar{I}_4 I_3 + I_6 + I_7$

$y = \bar{I}_5 \bar{I}_4 \bar{I}_3 I_2 + \bar{I}_5 \bar{I}_4 I_3 + I_6 + I_7$

$y = \bar{I}_5 \bar{I}_4 (\bar{I}_3 I_2 + I_3) + I_6 + I_7$

$y = \bar{I}_5 \bar{I}_4 (I_2 + I_3) + I_6 + I_7$

$Z = \bar{I}_7 \bar{I}_6 \bar{I}_5 \bar{I}_4 \bar{I}_3 \bar{I}_2 I_1 + \bar{I}_7 \bar{I}_6 \bar{I}_5 \bar{I}_4 I_3 + \bar{I}_7 \bar{I}_6 I_5 + I_7$

$Z = \bar{I}_6 \bar{I}_5 \bar{I}_4 \bar{I}_3 \bar{I}_2 I_1 + \bar{I}_6 \bar{I}_5 \bar{I}_4 I_3 + \bar{I}_6 I_5 + I_7$

$Z = \bar{I}_6 (\bar{I}_5 \bar{I}_4 \bar{I}_3 \bar{I}_2 I_1 + \bar{I}_5 \bar{I}_4 I_3 + I_5) + I_7$

$Z = \bar{I}_6 (\bar{I}_4 \bar{I}_3 \bar{I}_2 I_1 + \bar{I}_4 I_3 + I_5) + I_7$

$Z = \bar{I}_6 (\bar{I}_4 (\bar{I}_3 \bar{I}_2 I_1 + I_3) + I_5) + I_7$

$Z = \bar{I}_6 (\bar{I}_4 (\bar{I}_2 I_1 + I_3) + I_5) + I_7$

# FLIP FLOPS

1: A FLIP FLOPS is a binary storage device That can store one bit of data in it.

2: A FLIP FLOPS has Two outputs namely Q (representing The normal value of bit stored in it and $\bar{Q}$ (representing the complement of the value stored in it

3: A FLIP FLOPS also has two state namely set and Reset states.

4. A FLIP-FLOPS is said to be in the set state when $Q = 1$ and $\bar{Q} = 0$.

5: AP FLIP-FLOPS is said to be in the Reset State when $Q = 0$ and $\bar{Q} = 1$.

6: The state of the FLIP-FLOPS is determind from Q output of the FLIP_FLOPS.

7: Further, a FLIP_FLOPS can maintain it's state indefinitely until we change the input data.

8: For the normal operations of FLIP_FLOPS, both Q and $\bar{Q}$ must be complement to each other.

9: Binary information can be stored through different ways in FLIP_FLOPS giving rise to different types of flip-flops.

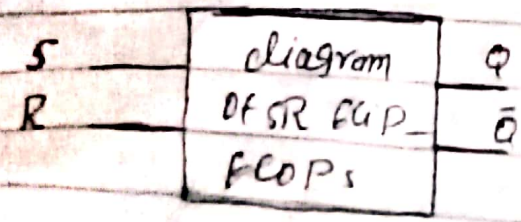
10: The commonly used FLIP_FLOPS are:

- → SR FLIP_FLOPS
- → D FLIP_FLOPS
- → JK FLIP_FLOPS
- → T FLIP_FLOPS

bleack diagrams

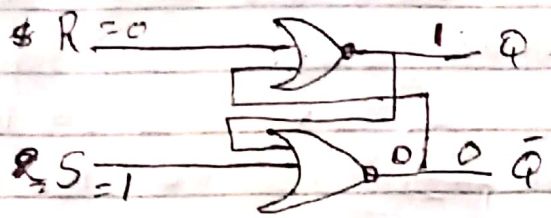
Turth Table

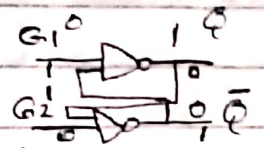
| R | S | O/P |
|---|---|---|
| 0 | 0 | |
| 0 | 1 | 1 |
| 1 | 0 | 0 |
| 1 | 1 | 0 |



# Basic SR Flip-Flops using NOR Gate

A basic Flip-Flops can be constructed using either Two NOR gate or Two NAND gate construction of a basic RS Flip-Flops using Two NOR gates is shown below

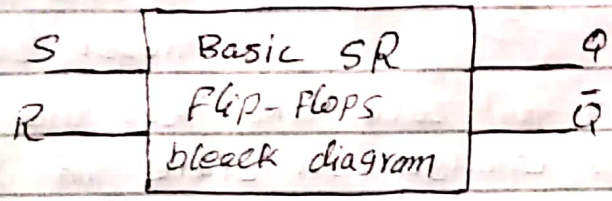



if we give zero to it. It will give us a set state

if we give one to it It will give us a Reset state

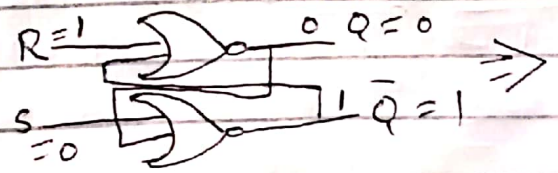
bleack diagram




case 1: Assume R=0 and S=1
so it will give one in Q=1 and $\bar{Q}$=0
when the output of Q is 1 it is called set state

case2: Assume R=1 and S=0
so it will give one in $\bar{Q}$=1 and Q=0 will be 0
when the output of $\bar{Q}$ is one ($\bar{Q}$=1) and Q is zero (Q=0) it is called Reset State

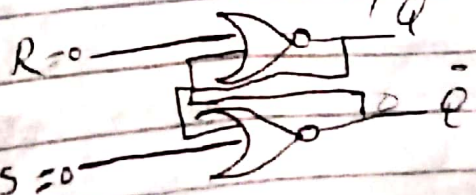
diagram for case 2




⇒ Q=0 and $\bar{Q}$=1
This is Reset state

Case 3: Assume R=0 and S=0
so it will give one in Q=1 and $\bar{Q}$=0
so it will show No change in state
This is for case 1:

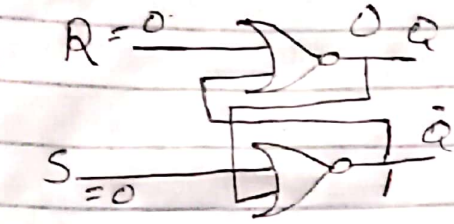
diagram




So it shows No change in Flip flops as same as case1 set state

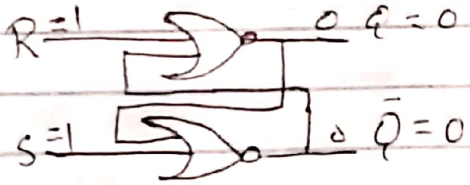
Case 4: Assume That R=0 and S=0
So it will give one in $\bar{Q}$ ($\bar{Q}=1$) and Q=0
So it will also shows No change in state
This one for case 2
diagram:

In case 2 $\bar{Q}$ was one so it will stay one




so it shows No change in flip-flops as same as case 2 Reset state

if you apply zero and zero like R=0 and S=0 So it will show No change in state if There were a set state after it or Reset after it
case 5: Assume That R=1 and S=1
in this situation, one of the input of both gates is "1", so both Q and $\bar{Q}$ will be "0" zero violating The fact that both Q and $\bar{Q}$ must be complement to each other. This state of The flip-flop when both Q and $\bar{Q}$ become same is avoidable/undefined state.



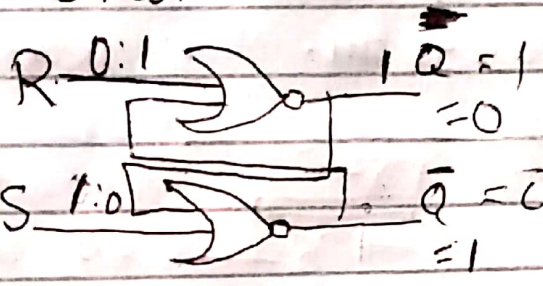

so This is avoidable or undefined state

case 6: Assume R=0 and S=0
However if case 5 occurs and then S and R both are made 0 simultaneously, a problem will occur. The output of the flip-flops is not determined If R=0 and S=0 after case 5 so we will not defend That The flip-flops is in which state so it will go to set state or Reset we don't defend it. we can defend

it in one case ~~then~~ it will detend That which input is Remain coger one. if R is Remain coger one so it will give as a set state

if S is Remain coger one so it will give as a set state.

diagram.

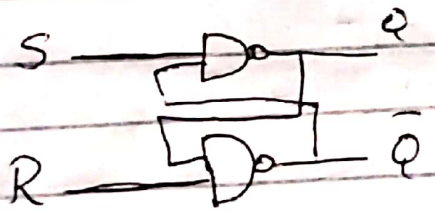


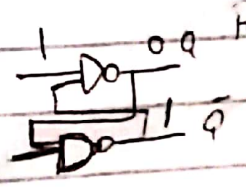

if R equal to zero frist so it will give as set state

if s equal to zero frist so it will give as a Reset state

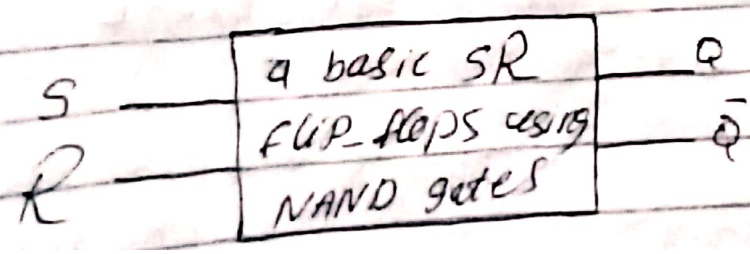
Basic SR FLIP-flops using NAND gates

A basic FLIP-FLOPS can be constructed using either two NOR gates or two NAND gates The construction of a basic RS FLIP-FLOP using Two NAND gates is shown below:




bleack diagram



- Let us analyze the operation of the basic NAND flip flops shown in the one. we know That the output of a NAND gate is (1) if any input is 0 and the output is 0 only when all it's inputs are (1).

Truth Table

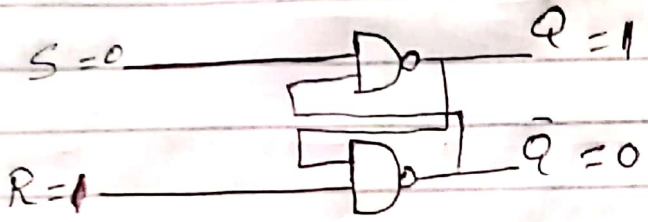
| I/P | | O/P |
|---|---|---|
| 0 | 0 | 1 |
| 0 | 1 | 1 |
| 1 | 0 | 1 |
| 1 | 1 | 0 |

Case 1. Assume $S=0$ and $R=1$

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

So This is on set state.

diagram.




So when $Q=1$ and $\bar{Q}=0$ it's mean That it's set state

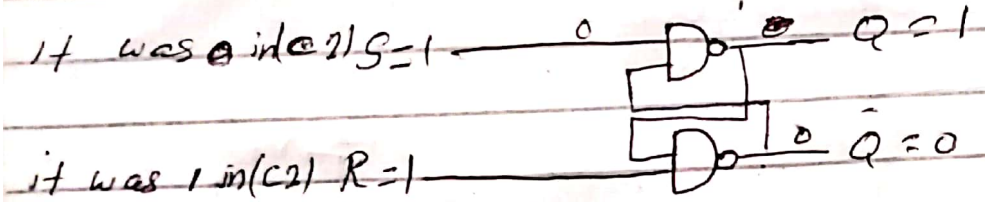
case2: Assum $S=1$ and $R=1$

when we look to the case it was $S=0$ and $R=1$ so it was set state new when we apply $S=1$ and $R=1$ so it give same as in case 1.

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

so it called no change in state

diagerm.



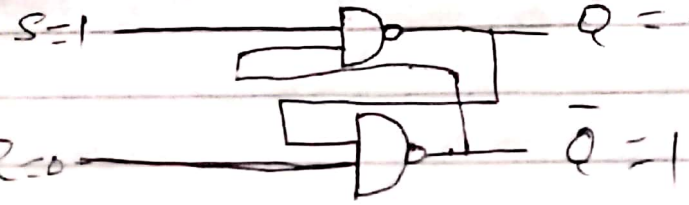

so it give as set state No change will happen to The state

case 3: Assum That $S=1$ and $R=0$

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

so it will show The Reset state

diagram.



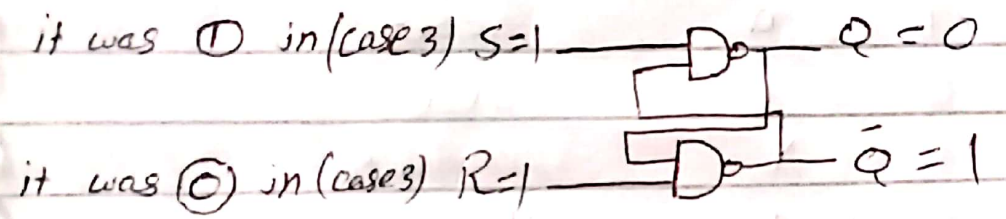

So This on Reset state $Q=0$ and $\bar{Q}=1$ No change state

case 4 Assume $S=1$ and $R=1$

when we look to the case 3 it was $S=1$ and $R=0$ so it was Reset state. When we apply $S=1$ and $R=1$ so it will still Reset state due to case 3.

$Q=0$ and $\bar{Q}=1$ it's called Reset state

diagram.

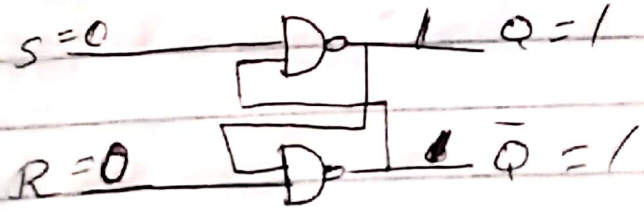



when $Q=0$ and $\bar{Q}=1$ it is a Reset state due to case 3 No change state

case 5: Assume $S=0$ and $R=0$

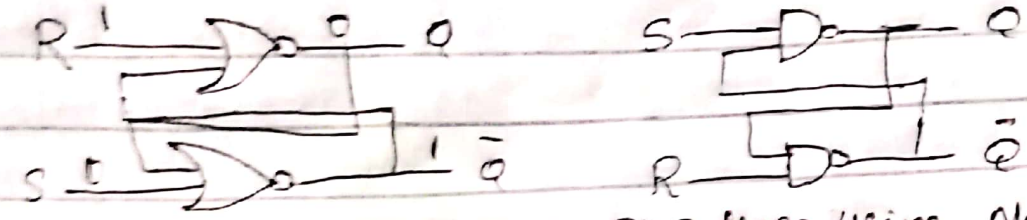
in this situation both of the gate will give one so we said That The output should complement of each other so we give $S=0$ and $R=0$ it will give as $Q=1$ and $\bar{Q}=1$ so this avoidable or undefined state.

diagram.




when $Q=1$ and $\bar{Q}=1$ it is Avoidable state

Comparison between basic SR flip flops using NOR gates and NAND Gates

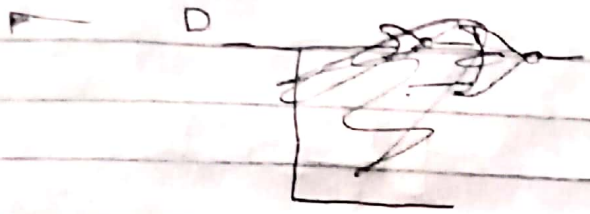



Implementation of ~~[illegible]~~ flip-flops using Nor gates

| | $Q_T$ | S | R | $Q_{T+1}$ | State |
|---|---|---|---|---|---|
| | 0 | 0 | 0 | 0 | No change |
| → | 0 | 0 | 1 | 0 | Reset |
| | 0 | 1 | 0 | 1 | set |
| | 0 | 1 | 1 | X | Avoidable |
| | 1 | 0 | 0 | 1 | No change |
| | 1 | 0 | 1 | 0 | Reset |
| | 1 | 1 | 0 | 1 | set |
| | 1 | 1 | 1 | X | Avoidable |

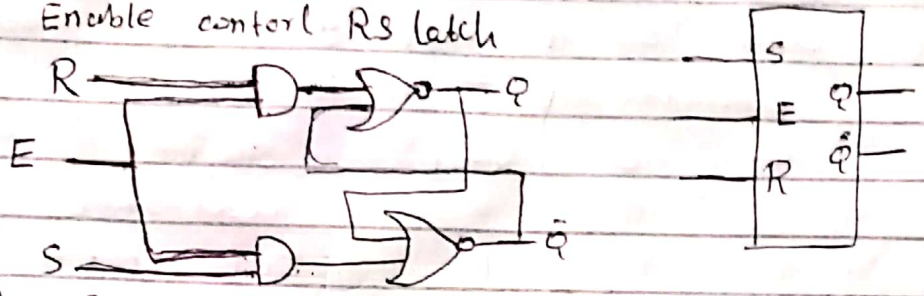
implementation of flip-flops using NAND gates

| $Q_T$ | S | R | $Q_{T+1}$ | State |
|---|---|---|---|---|
| 0 | 0 | 0 | X | Avoidable |
| 0 | 0 | 1 | 1 | set |
| 0 | 1 | 0 | 0 | Reset |
| 0 | 1 | 1 | 0 | No change |
| 1 | 0 | 0 | X | Avoidable |
| 1 | 0 | 1 | 1 | set |
| 1 | 1 | 0 | 0 | Reset |
| 1 | 1 | 1 | 1 | No change |




Gated SR latch

Enable control RS latch

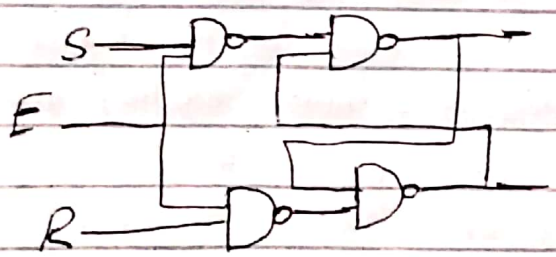



Gated SR latch using Nor Gate

E — charatristic Tunth table

| $G_T$ | S | R | $Q_{T+1}$ | State |
|---|---|---|---|---|
| 0 | 0 | 0 | $Q_T$ | |
| 0 | 0 | 1 | $Q_T$ | |
| 0 | 1 | 0 | $Q_T$ | |
| 0 | 1 | 1 | $Q_T$ | |
| 1 | 0 | 0 | $Q_T$ | |
| 1 | 0 | 1 | 0 | Reset |
| 1 | 1 | 0 | 1 | set |
| 1 | 1 | 1 | X | AVoidable |

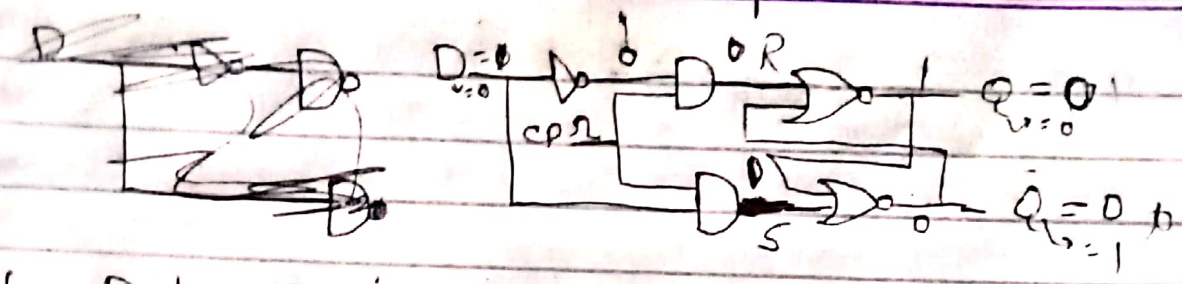
~~SR latch Gated RS latch using NAND gate~~




When we are doing Gated SR latch wethere it is NOR Gate or NAND Gate The output is same; S=1 and R=0 = SEt state R=0 and S=0 No change

S=0 and R=1 = Reset state R=1 and S=0 Avoidable

# D Flip Flops



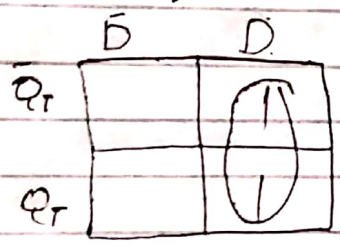

if D=1 So it will give us set state
if D=0 so it will give us a Reset state

charateristic Turth table

| | $Q_T$ | D | $Q_{T+1}$ | State |
|---|---|---|---|---|
| $m_0$ | 0 | 0 | 0 | Reset state |
| $m_1$ | 0 | 1 | 1 | set state |
| $m_2$ | 1 | 0 | 0 | Rest state |
| $m_3$ | 1 | 1 | 1 | set state |

charateristic equation

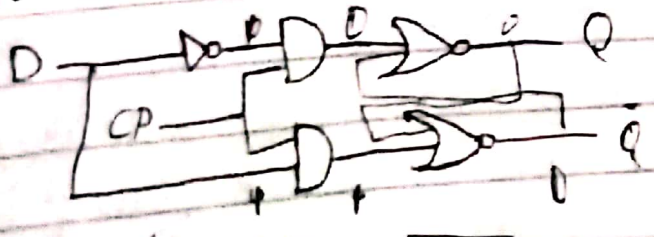
charateristic equation = D




$Q_{T+1} = D$

## Gated D latch

- SR latch is in the prohibited stated when both R = S = 1.
- D (data input) latch is single input version of SR latch
- NoR gates #2 and #2 form the basic Flip-flop and AND gates on A and B are used for the control input.
- D input goes directly to s input (gate #b) and it's complement is applied at R input (gate #a).
- The circuit diagram for D latch.

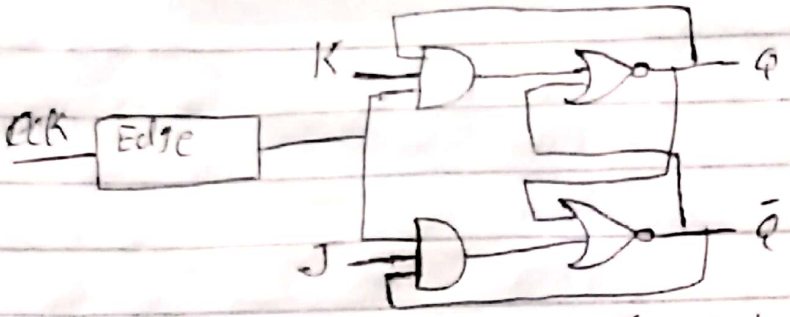



| E | D | $Q_T$ | $Q_{T+1}$ |
|---|---|---|---|
| 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 1 |
| 0 | 1 | 0 | 0 |
| 0 | 1 | 1 | 1 |
| 1 | 0 | 0 | 0 |
| 1 | 0 | 1 | 1 |
| 1 | 1 | 0 | 1 |
| 1 | 1 | 1 | 1 |

Block diagram

# Edge Triggered JK Flip flop

A JK flip-flop is modified form SR flip-flop in order to allow the indeterminate state of SR flip-flop. JK flip-flop is shown in fig below:




In general we can summarize the behavior of JK flip-flop as

J=0 and K=0 → No change in the state of flip-flop

J=1 and K=0 → This is set state

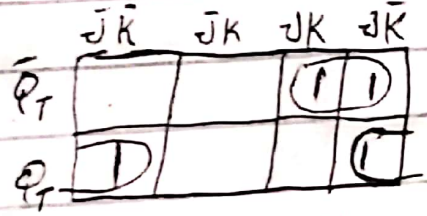
J=0 and K=1 → This is Reset State

J=1 and K=1 → This is complement it's state like if stat is set it's become Reset and Reset become set state.

The characteristic table and characteristic equation for the JK flip are shown below.

| | $Q_T$ | J | K | $Q_{T+1}$ |
|---|---|---|---|---|
| $m_0$ | 0 | 0 | 0 | 0 |
| $m_1$ | 0 | 0 | 1 | 0 |
| $m_2$ | 0 | 1 | 0 | 1 |
| $m_3$ | 0 | 1 | 1 | 1 |
| $m_4$ | 1 | 0 | 0 | 1 |
| $m_5$ | 1 | 0 | 1 | 0 |
| $m_6$ | 1 | 1 | 0 | 1 |
| $m_7$ | 1 | 1 | 1 | 0 |

Characteristic equation

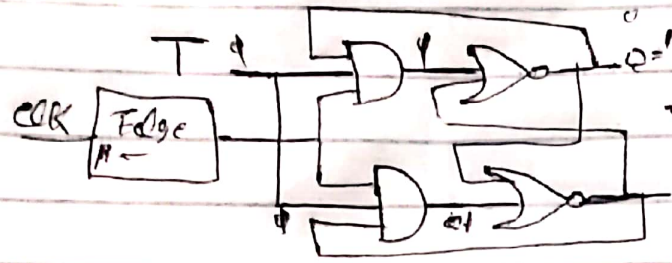



$Q_{T+1} = \bar{Q}_T J + Q_T \bar{K}$

# Edge Triggered T flip-flop

- T flip-flop is slight modification of JK flip-flop with J and K inputs are tied together and named as T

The circuit diagram of T flip-flop is shown in the fig. below:

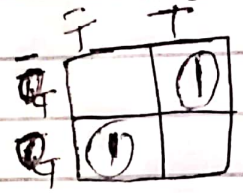



In JK flip-flop
J=0 and K=0 → No change
J=0 and K=1 → Reset
J=1 and K=0 → set
J=1 and K=1 → Toggle

The characteristic table and the characteristic equation are shown below:

| | $Q_T$ | T | $Q_{T+1}$ |
|---|---|---|---|
| $m_0$ | 0 | 0 | 0 |
| $m_1$ | 0 | 1 | 1 |
| $m_2$ | 1 | 0 | 1 |
| $m_3$ | 1 | 1 | 0 |

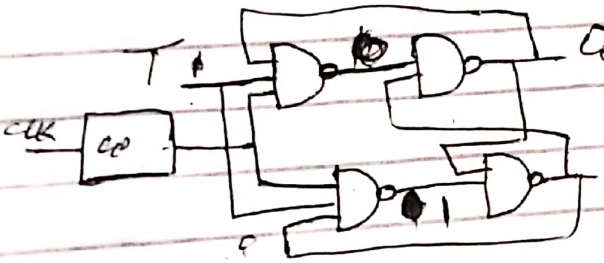
Characteristic equation




$$Q_{T+1} = Q_T\bar{T} + \bar{Q}_T T$$

## Edge Triggered T flip flop using NAND Gate

The circuit diagram of T flip-flop is shown in the figure below:

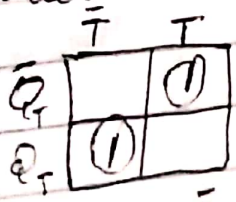



The same behavior as T flip-flop using NOR Gate

The characteristic table and the characteristic equation are shown below:

| | $Q_T$ | T | $Q_{T+1}$ |
|---|---|---|---|
| $m_0$ | 0 | 0 | 0 |
| $m_1$ | 0 | 1 | 1 |
| $m_2$ | 1 | 0 | 1 |
| | 1 | 1 | 0 |

characteristic equation




$$Q_{T+1} = Q_T\bar{T} + \bar{Q}_T T$$