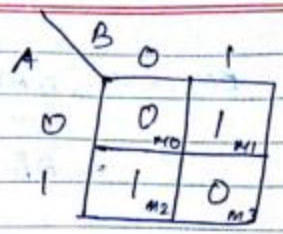
SECTION-D

Binary Subtractor

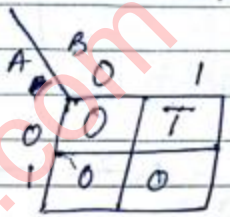
- half
- Full Subtractor
- half Subtractor

A	B	Diff	Borrow
0	0	m0 0	0
0	1	m1 1	1
1	0	m2 1	0
1	1	m3 0	0

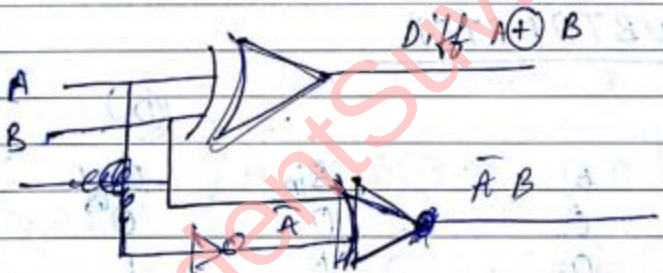


$Diff = A\bar{B} + \bar{A}B$

$= A \oplus B$



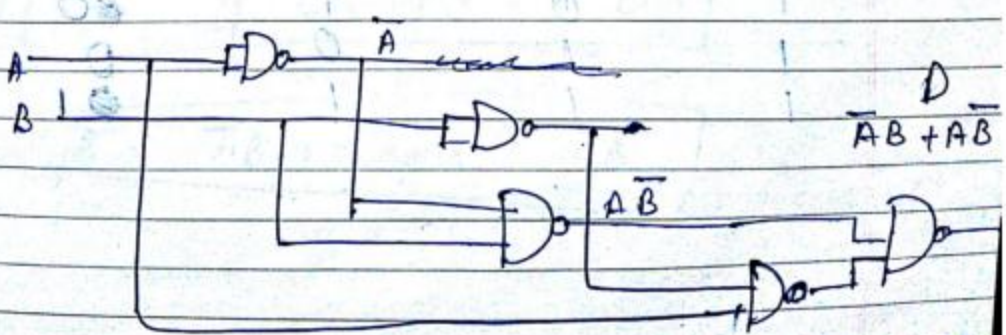
$Borrow = B_o = \bar{A}B$



NAND

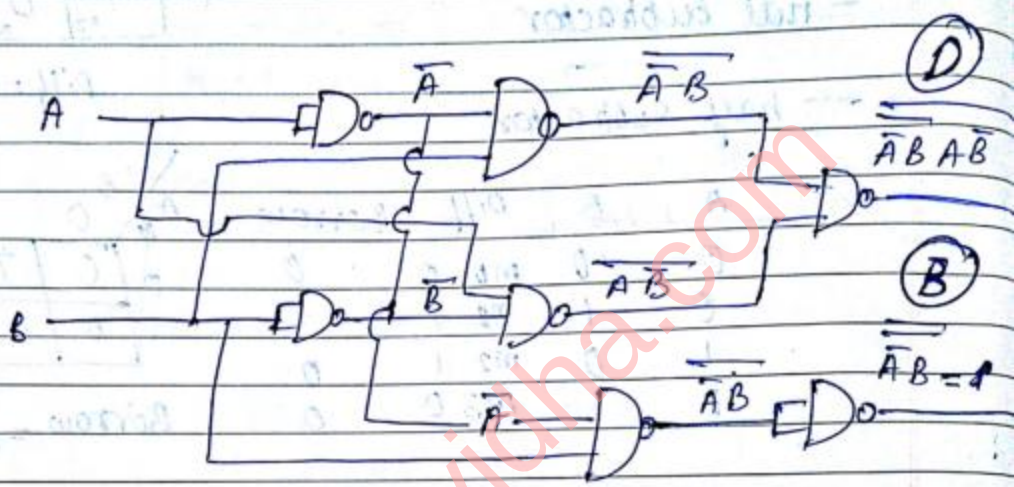
Difference $\bar{A}B + A\bar{B} = D$

$\overline{\bar{A}B} \cdot \overline{A\bar{B}} = D$



Borrow NAND

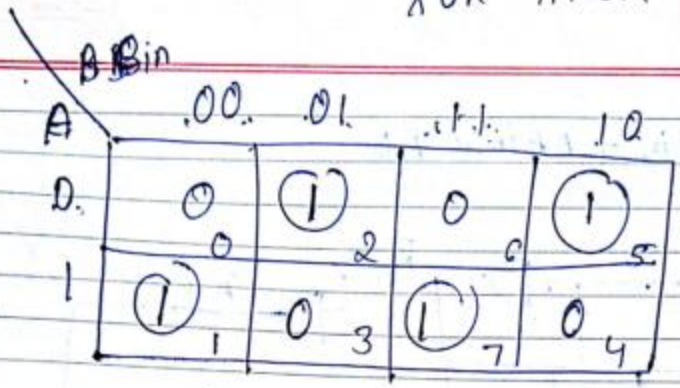
$= \overline{\overline{A}B}$



FULL SUBTRACTOR

A	B	Bin	D Diff	B Borrow
0	0	0	0	0
0	0	1	1	1
0	1	0	1	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

XOR - XNOR



$$D = \bar{A}B\,Bin + AB\bar{Bin} + A\bar{B}\,Bin + AB\,Bin$$

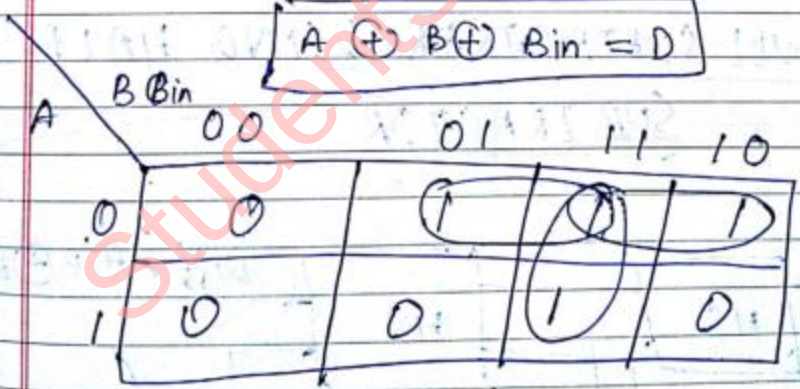
$$D = \bar{A}(\underbrace{B\,Bin + \bar{B}\,\bar{Bin}}_{\overline{XOR}}) + A(\underbrace{\bar{B}\,\bar{Bin} + B.\,Bin}_{XNOR})$$

$$\bar{A}(B \oplus Bin) + A(\overline{B \oplus Bin})$$

Assume $B \oplus Bin = X$

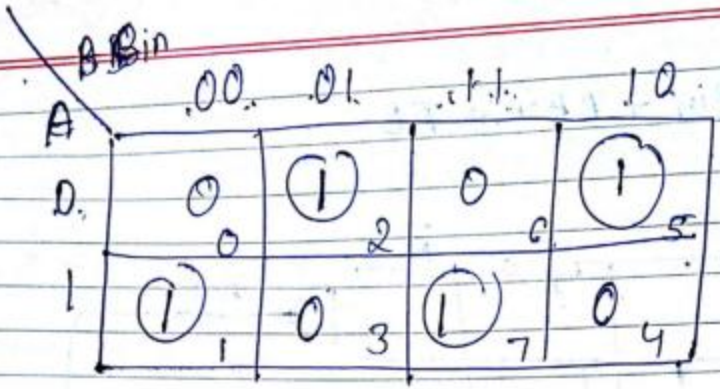
$$\bar{A}X + A\bar{X} = A \oplus X$$

$$A \oplus B \oplus Bin = D$$



$$B = \bar{A}Bin + Bin\,B + \bar{A}B$$

XOR - XNOR



$$D = \bar{A}B\bar{Bin} + AB\bar{Bin} + A\bar{B}\bar{Bin} + ABBin$$

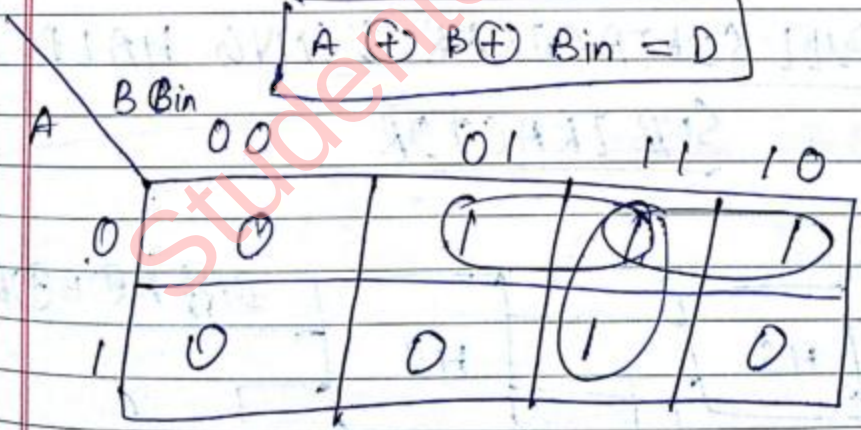
$$D = \bar{A}\underbrace{(B\bar{Bin} + \bar{B}Bin)}_{XOR} + A\underbrace{(B\bar{Bin} + BBin)}_{XNOR}$$

$$\bar{A}(B \oplus Bin) + A\overline{(B \oplus Bin)}$$

Assume $B \oplus Bin = X$

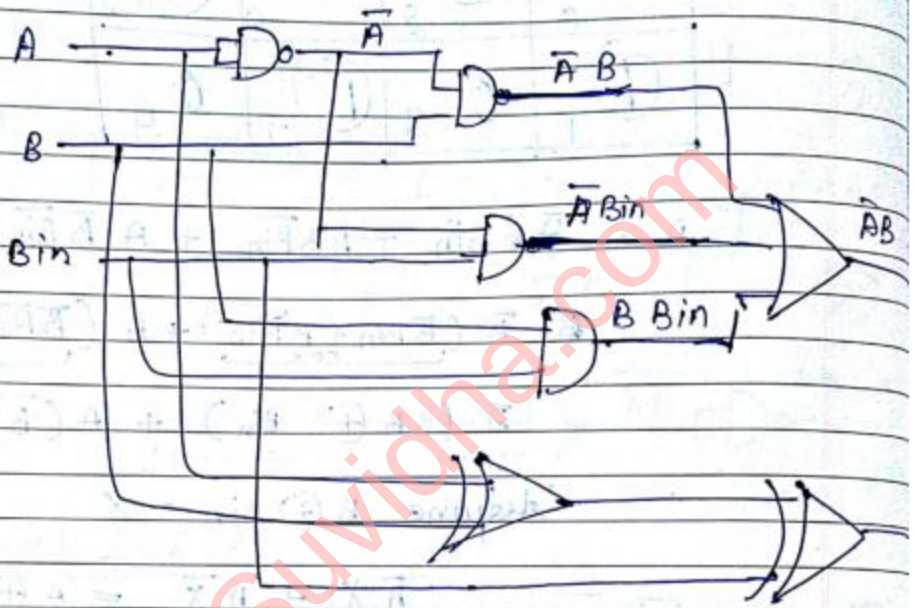
$$\bar{A}X + A\bar{X} = A \oplus X$$

$$A \oplus B \oplus Bin = D$$

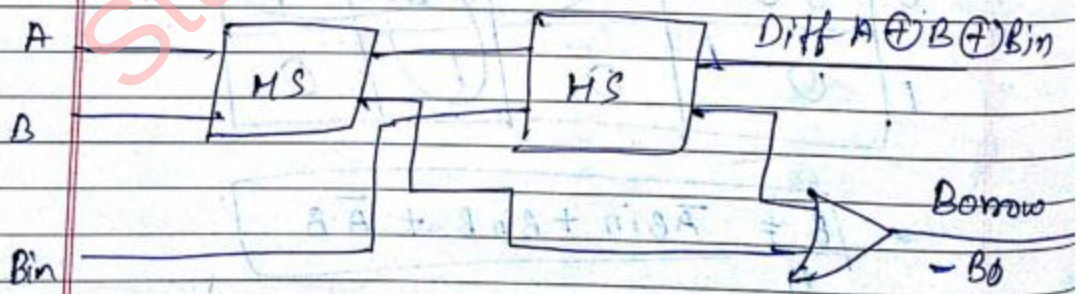


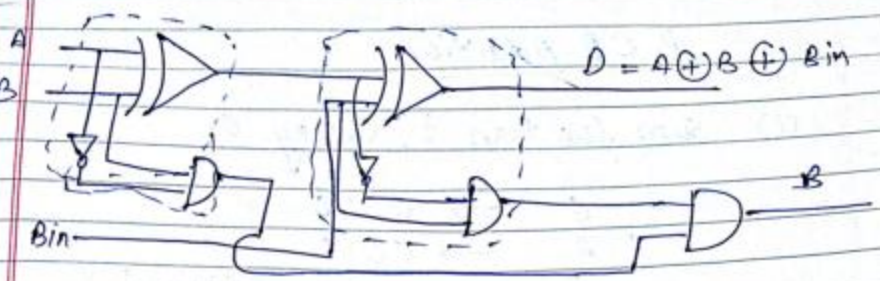
$$B = \bar{A}Bin + Bin\,B + \bar{A}B$$

(B) $\bar{A}B_{in} + BB_{in} + \bar{A}B$



FULL SUBTRACTRAR USING HALF SUBTRACTOR





(B)

HBin + $\bar{A}$B

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

$(\bar{A}\bar{B} + AB)Bin + \bar{A}B$

(D)

A⊕B⊕Bin

$\bar{A}\bar{B}Bin + ABBin + \bar{A}B(1 + Bin)$

$\bar{A}\bar{B}Bin + ABBin + \bar{A}B + \bar{A}BBin$

$\bar{A}Bin(\bar{B} + B) + ABBin + \bar{A}B$

$\bar{A}Bin + ABBin + \bar{A}B(1 + Bin)$

$\bar{A}Bin + ABBin + \bar{A}B + \bar{A}BBin$

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

BCD Addition

(1) Sum less than 9, carry 0

$$\begin{array}{rcl} 6 & \rightarrow & 0110 \\ +\ 2 & \rightarrow & 0010 \\ \hline 8 & & 1000 \end{array}$$ (Binary)

↓

BCD

(2) Sum greater than or equal to 9, carry 0

$$\begin{array}{rcl} 9 & & 1001 \\ +4 & & 0100 \\ \hline 13 & & 1101 \end{array} \rightarrow \text{Binary addition}$$

$+0110 \rightarrow$ Add 6

$0001 \quad 0011$

1 3 → BCD

(3) Sum greater than 9 but carry is equivalent to

$$\begin{array}{rcl} 9 & & 1001 \\ +8 & + & 1000 \\ \hline 17 & & 10001 \end{array}$$ Binary addition

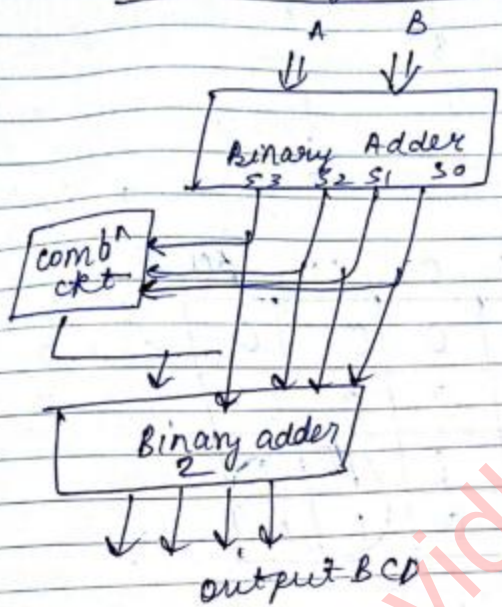
$+0110$

$0001 \quad 0111$

1 7 BCD

(Decimal Adder)

Block Diagram of BCD Adder



Design Combinational Ckt of BCD Adder

(1) if Sub > 9, Carry

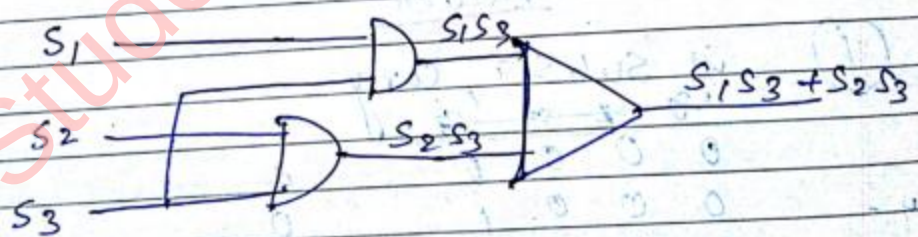
S3	S2	S1	S0	Y
0	0	0	0	0
0	0	0	1	0
0	0	1	0	0
0	0	1	1	0
0	1	0	0	0
0	1	0	1	0
0	1	0	0	0
0	1	0	1	0
1	0	1	0	0
1	0	1	1	0
1	1	0	0	0
1	1	0	1	0
1	1	1	0	0
1	1	1	1	0

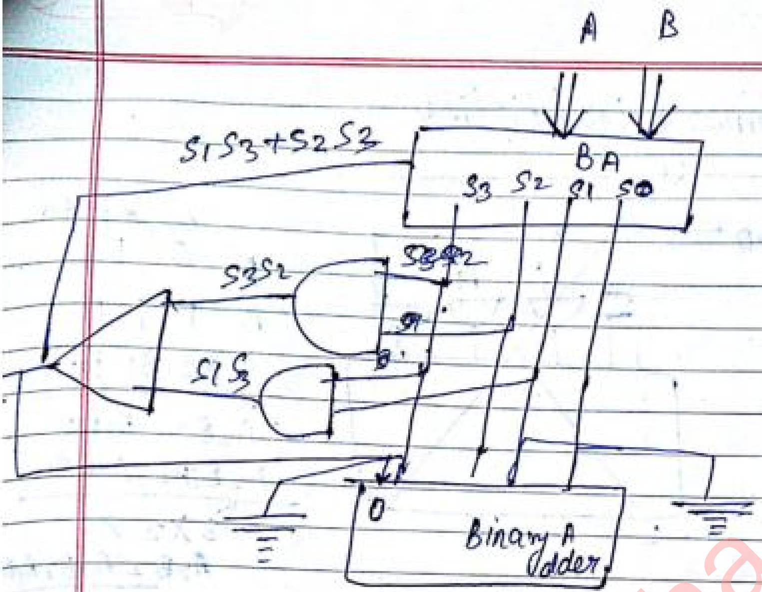
				Y
1	0	0	0	1
1	0	0	1	1
1	1	1	0	1
1	1	1	1	1
1	1	0	0	1
1	1	0	1	1

$S_3 = A$
$S_2 = B$
$S_1 = C$
$S_0 = D$

S_3S_2 \ S_1S_0	00	01	11	10
00	0	0	0	0
01	0	0	0	0
11	1	1	1	1
10	0	0	1	1

$Y = S_2S_3 + S_3S_1$





BCD SUBTRACTOR

10's Complement
↓
9's Complement + 1

4 no.

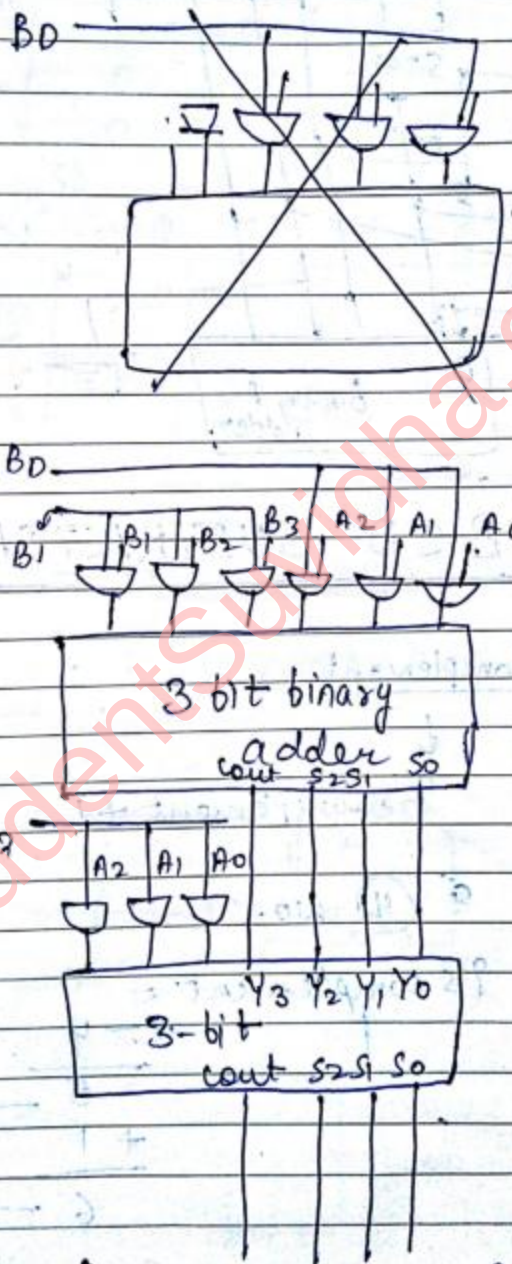
9's complement = 9 − 4 = 5 ← 9's
+ 1
6 ← 10's

8
− 4
(4)

8 + 6 = 14

1000
−0110
1110 +6
0110

Binary Multiplier



$A_2\ A_1\ A_0$

A 0 1 1

B 1 1 1

$B_2\ B_1\ B_0$

A_2B_0 A_1B_0

A_2B_1 A_1 A_0B_1

$X_3\ X_2\ X_1$

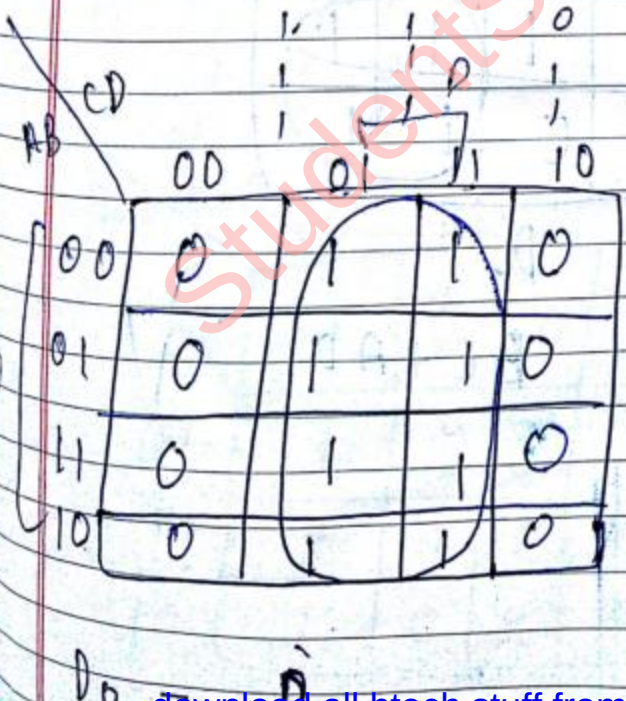
A_2B_2 A_1B_2 A_0

X

(Block diagram of Binary Model)

Binary to BCD Conversion

A	B	C	D	BCD D_4	D_3	D_2	D_1	D_0
0	0	0	0	0	0	0	0	0
0	0	0	1	0	0	0	0	1
0	0	1	0	0	0	0	1	0
0	0	1	1	0	0	0	1	1
0	1	0	0	0	0	1	0	0
0	1	0	1	0	0	1	0	1
0	1	1	0	0	0	1	1	0
0	1	1	1	0	0	1	1	1
1	0	0	0	0	1	0	0	0
1	0	0	1	0	1	0	0	1
1	0	1	0	0	1	1	0	0
1	0	1	1	0	1	1	0	1
1	1	0	0	0	1	1	1	0
1	1	0	1	0	1	1	1	1
1	1	1	0	1	0	0	0	0
1	1	1	1	1	0	0	0	1



D_0

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

D_2

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

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

D_3 $\bar{A}B\bar{C}$

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

D_4

$A\bar{C} + AD$

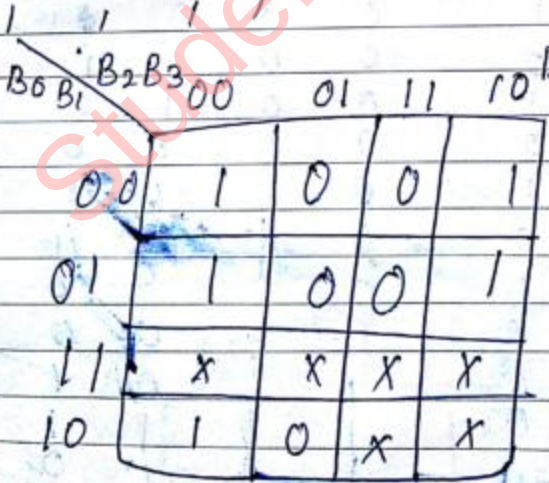
B. A

B

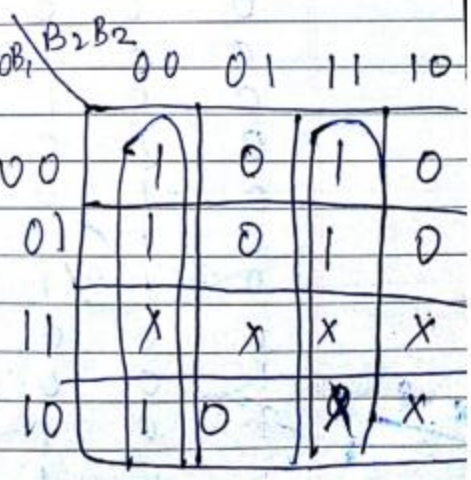
BCD to Excess 3 Convertor

A	B	C	D	BCD B_3	B_2	B_1	B_0	Excess 3 E_3	E_2	E_1	E_0
0	0	0	0	0	0	0	0	0	0	1	1
0	0	0	1	0	0	0	1	0	1	0	0
0	0	1	0	0	0	1	0	0	1	0	1
0	0	1	1	0	0	1	1	0	1	1	0
0	1	0	0	0	1	0	0	0	1	1	1
0	1	0	1	0	1	0	1	1	0	0	0
0	1	1	0	0	1	1	0	1	0	0	1
0	1	1	1	0	1	1	1	1	0	1	0
1	0	0	0	1	0	0	0	1	0	1	1
1	0	0	1	1	0	0	1	1	1	0	0
1	0	1	0								
1	0	1	1								
1	1	0	0								
1	1	0	1								
1	1	1	0								
1	1	1	1								

X



E_0 $\overline{B_3}$



E_1 $B_2\overline{B} + B_1B_3$

B_0B_1 \ B_2B_1	00	01	11	10
00	0	1	1	1
01	1	0	0	0
11	X	X	X	X
10	0	1	X	X

B_0B_1 \ B_2B_3	00	01	11	10
00	0	0	0	0
01	0	1	1	1
11	X	X	X	X
10	1	1	~~0~~	~~0~~

Binary to Gray Code

$G_3 = B_3$

$G_2 = B_2 \oplus B_2$

$G_1 = B_1 \oplus B_2$

$G_0 = B_0 \oplus B_1$

B_3	B_2	B_1	B_0	$G_3=B_3$	G_2	G_1	G_0
0	0	0	0	0	0	0	0
0	0	0	1	0	0	0	1
0	0	1	0	0	0	1	1
0	0	1	1	0	0	1	0
0	1	0	0	0	1	1	0
0	1	0	1	0	1	1	1
0	1	1	0	0	1	0	1
0	1	1	1	0	1	0	0
1	0	0	0	1	1	0	0
1	0	0	1	1	1	0	1
1	0	1	0	1	1	1	1
1	0	1	1	1	1	1	0
1	1	0	0	1	0	1	0
1	1	0	1	1	0	1	1

Gray to Binary

$B_3 = G_3$

$B_2 = G_2 \oplus G_3$

$B_1 = G_1 \oplus G_2 \oplus G_3$

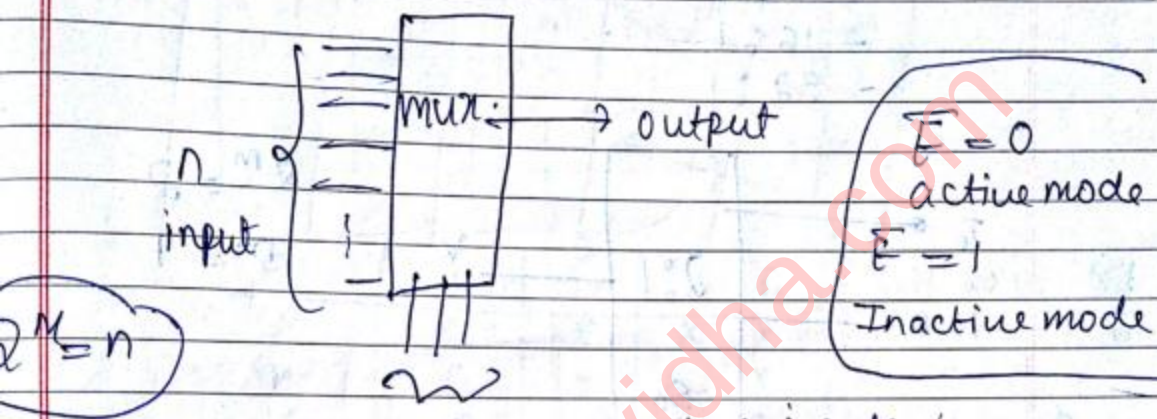
$B_0 = G_0 \oplus G_1 \oplus G_2 \oplus G_3$

G_3	G_2	G_1	G_0	B_3	B_2	B_1	B_0
0	0	0	0	0	0	0	0
0	0	0	1	0	0	0	1
0	0	1	0	0	0	1	1
0	0	1	1	0	0	1	0
0	1	0	0	0	1	1	0
0	1	0	1	0	1	1	1
0	1	1	0	0	1	0	1
0	1	1	1	0	1	0	0
1	0	0	0	1	1	0	0
1	0	0	1	1	1	0	1
1	0	1	0	1	1	1	1
1	0	1	1	1	1	1	0
1	1	0	0	1	0	1	0
1	1	0	1	1	0	1	1
1	1	1	0	1	0	0	1
1	1	1	1	1	0	0	0

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

Multiplexer

is a combinational ~~circuit~~ ~~ckt~~ ~~cct~~ circuit, it has n input and 1 output and n data select inputs.



N – Selected → n no. of select inputs.
n → inputs.

(1) A mux. is a digital circuit which select one of n data inputs and routes it to the output.

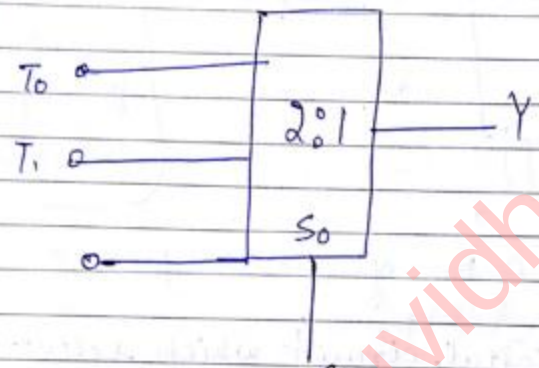
(2) $\overline{E}$ is called as Enable input which is useful for cascading. It is generally an active low terminal that means it will the required operation when it is low.

Advantage of MUX

- It reduce no. of wires.
- It simplify logic design.
- It reduce circuit complexity & cost.

mux types :-

- 2 : 1
- 4 : 1
- 8 : 1
- 16 : 1
- 32 : 1



$2^m = 2$

$m = 1$

S0

T0	T1	S0
0	0	0
0	1	

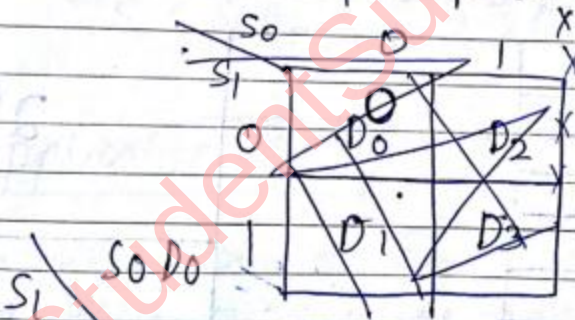
E	S0	Y
0	x	x
1	0	D0
1	1	D1

$2^M = 4$

E	S_0	S_1	D_0	D_1	Y
0	X	X	X	X	X
1	0	0	0	X	0
1	0	0	1	X	1
1	1	0	X	0	0
1	1	1	X	1	1

E	S_0	S_1	Y	A
0	X	X	X	X
1	0	0	D_0	X
1	0	1	D_1	0
1	1	0	D_2	1
1	1	1	D_3	X

S_1	S_0	D_0	X
0	0	0	0
0	0	1	1
0	1	X	0
0	1	X	1
1	0	X	0
1	0	X	1
1	1	X	0
1	1	X	1



0	1	X	X
001	011	010	
X	X	X	X
100	101	111	110

$\bar{S_1}\bar{S_0}X$

D_1

S_1 \ S_0	00	01	11	10
0	X	X	X	1
1	X	X	X	X

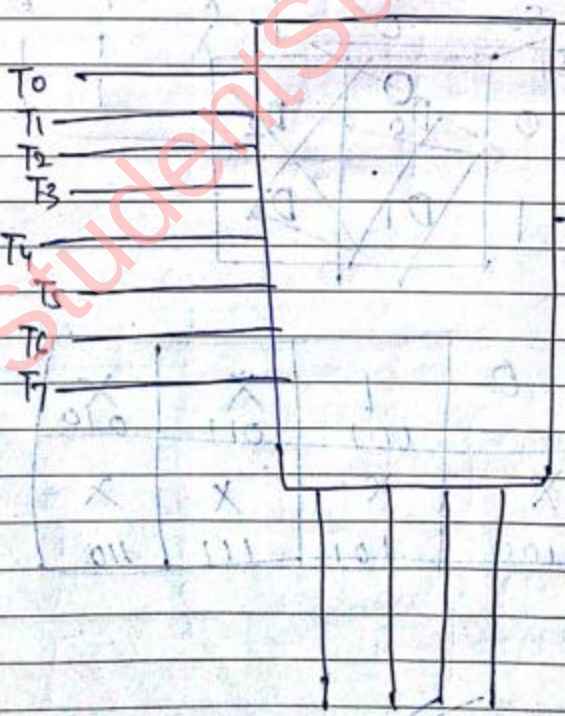
$D_1 = \bar{S_1} S_3 \bar{X}$

$D_2 = S_1 \bar{S_0} X$

$D_3 = S_1 S_0 \bar{X}$

$$Y = \bar{S_1} S_0 X + \bar{S_1} S_0 \bar{X} + S_1 \bar{S_0} X + S_1 S_0 \bar{X}$$

Design 8:1 Mux



$2^M = 8$

Truth Table

E	S_2	S_1	S_0	Y
0	X	X	X	0
1	0	0	0	D_0
1	0	0	1	D_1
1	0	1	0	D_2
1	0	1	1	D_3
1	1	0	0	D_4
1	1	0	1	D_5
1	1	1	0	D_6
1	1	1	1	D_7

$$E = \overline{S_2}\,\overline{S_1}\,\overline{S_0} D_0 + \overline{S_2}\,\overline{S_1} S_0 D_1 + \overline{S_2} S_1 \overline{S_0} D_2 + \overline{S_2} S_1 S_0 D_3$$
$$+ S_2 \overline{S_1}\,\overline{S_0} D_4 + S_2 \overline{S_1} S_0 D_5 + S_2 S_1 \overline{S_0} D_6$$

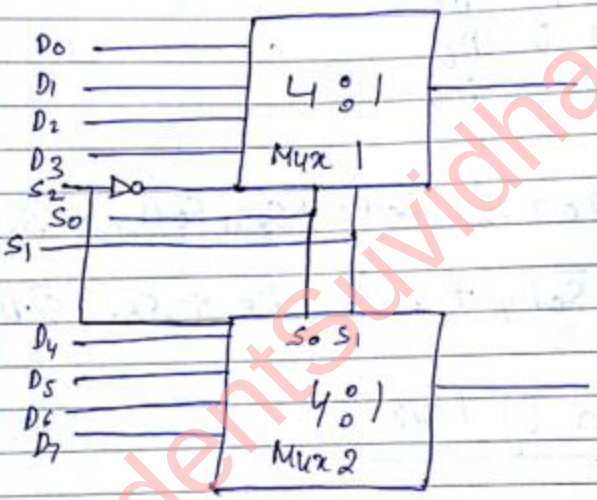
Application Of Mux

1. another name of Mux is data selector

Mux Tree

The mux having more no. of inputs can be ~~mux~~ obtained by cascading two or more multiplexers with less no. of input is called mux. Tree.

Ques Obtain 8:1 mux using ~~2:1~~ two 4:1 muxes.

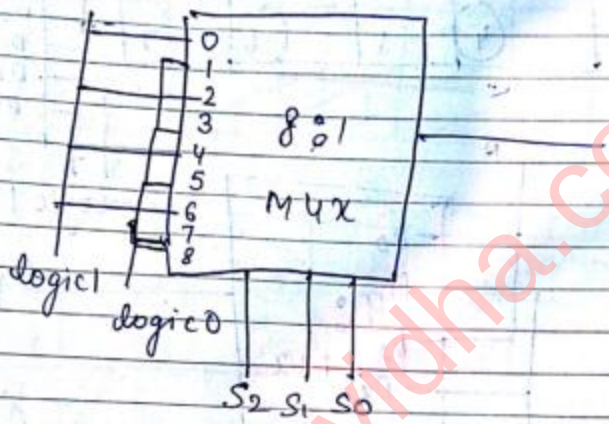


Truth Table

S_2	S_1	S_0	Y	
0	0	0	D_0	Mux 1 enable
0	0	1	D_1	
0	1	0	D_2	
0	1	1	D_3	
1	0	0	D_4	Mux 2 enable
1	0	1	D_5	
1	1	0	D_6	
1	1	1	D_7	

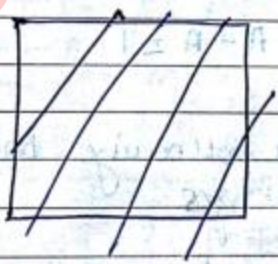
Que Implement the following expr^n using MUXs.

$F(A,B,C) = \Sigma m(0,2,4,6)$ = logic 1
or $\pi(1,3,5,7)$ = logic 0

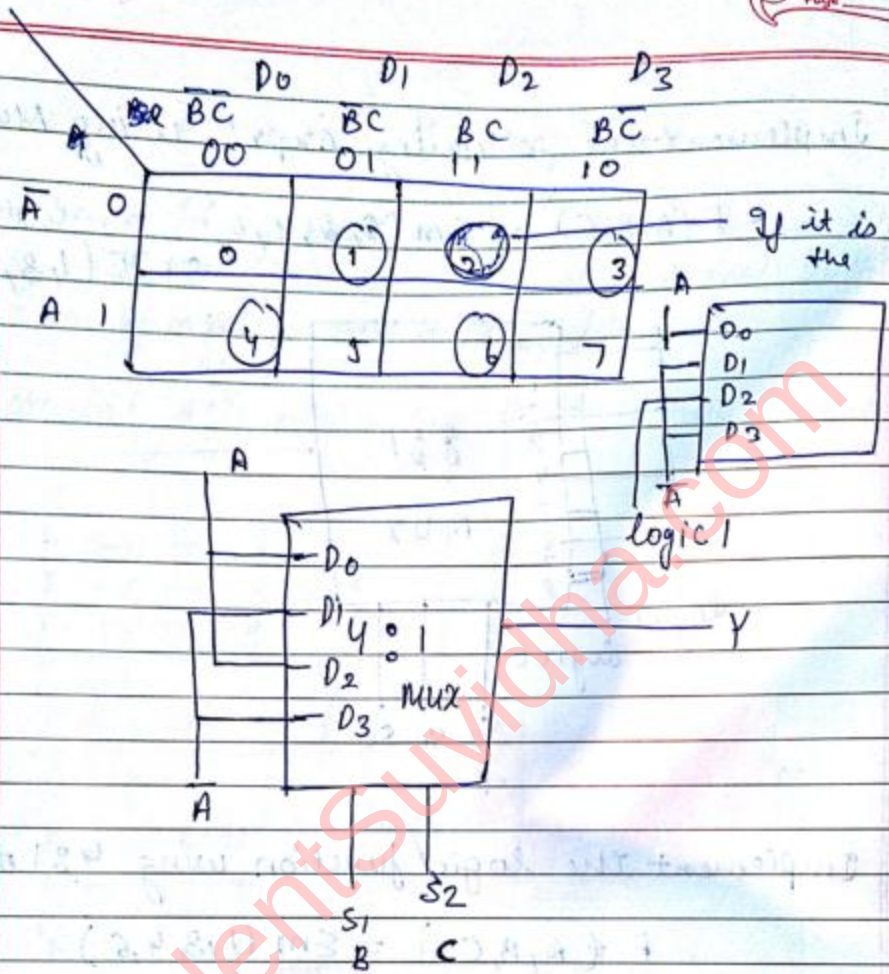


Que Implement the logic function using 4:1 mux.

$F[A,B,C] = \Sigma m(1,3,4,6)$



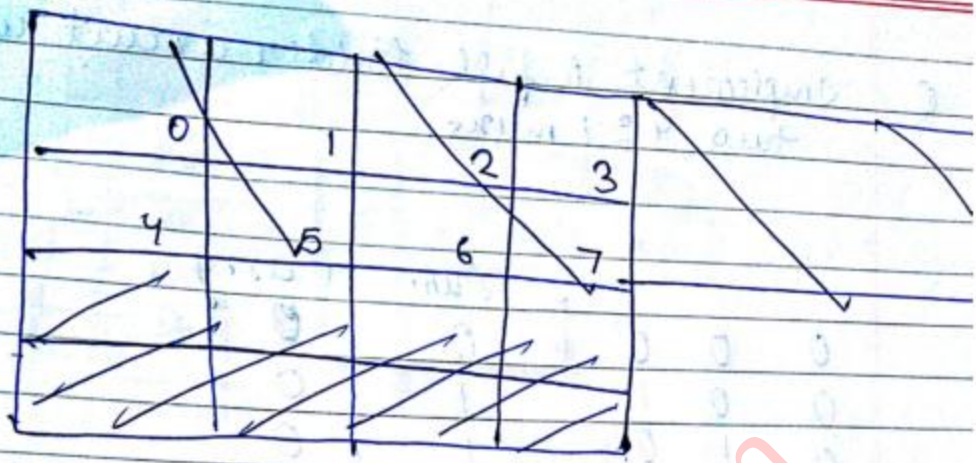
folding Table

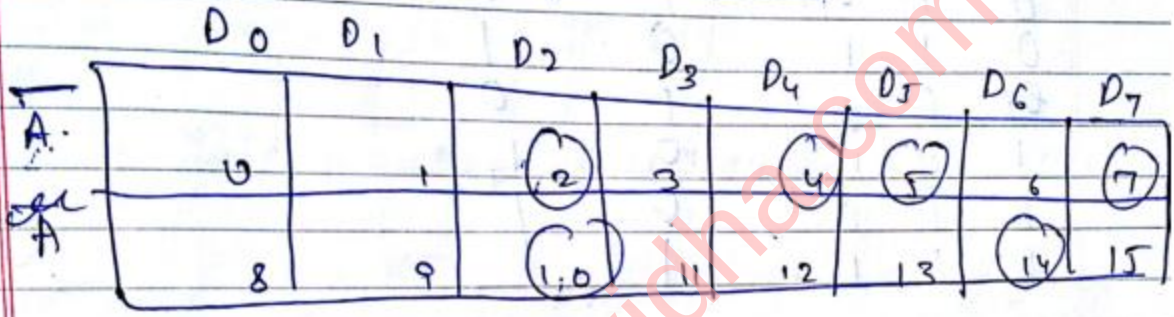


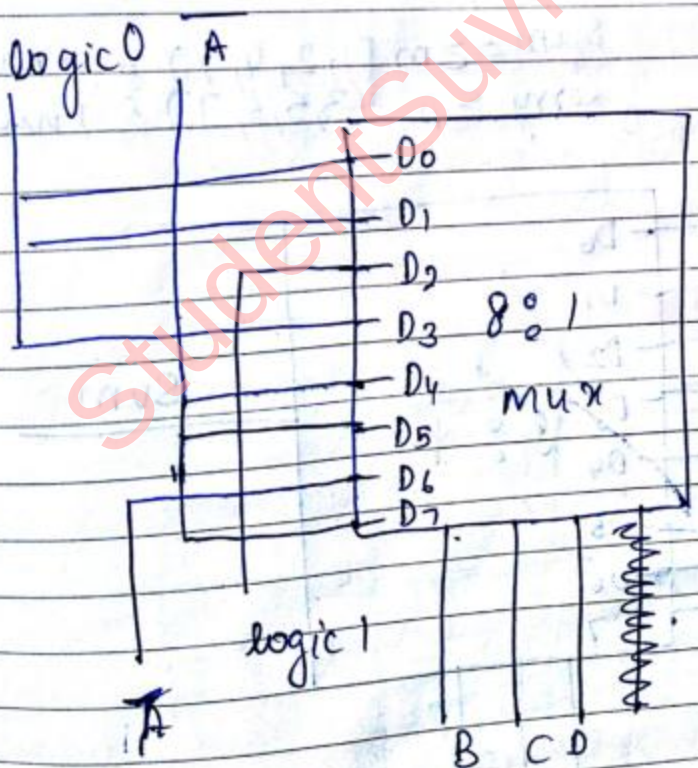
$A + \bar{A} = 1$

Ques Implement the following Boolean function 8:1 mux

$F(A, B, C, D) = \Sigma M(2, 4, 7, 5, 10, 14)$





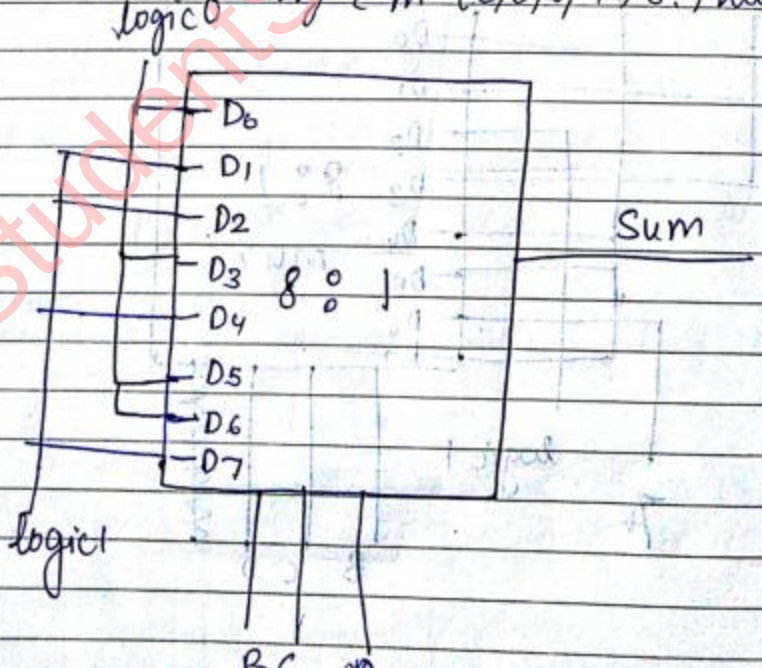


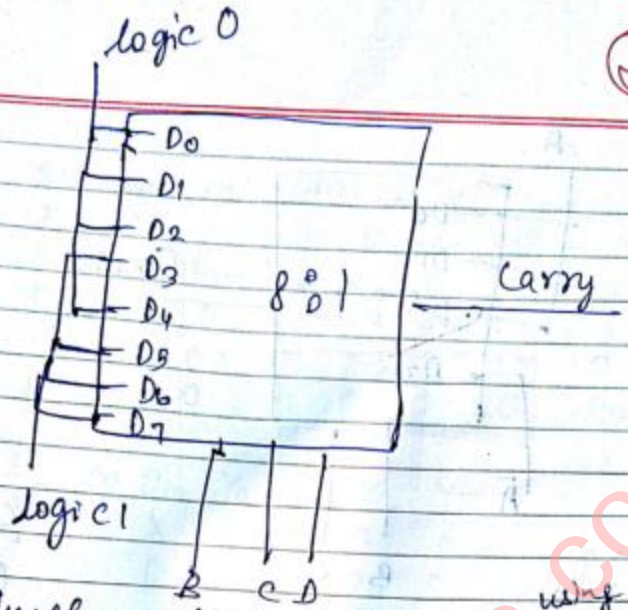
Q Implement a full adder circuit using two 4:1 muxs.

			Sum	Carry
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

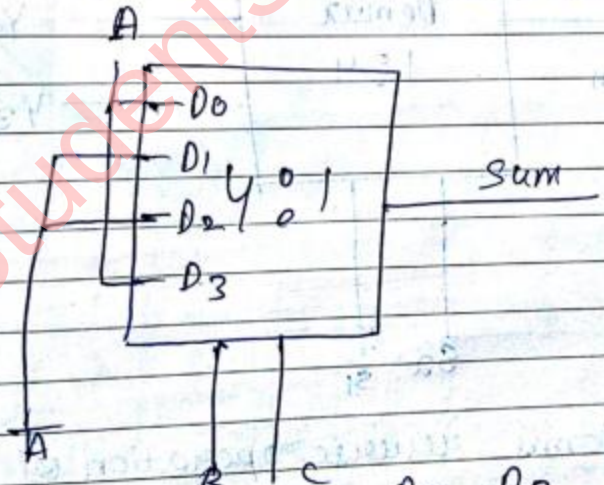
Sum = $\Sigma m(1,2,4,7)$ 8:1 mux

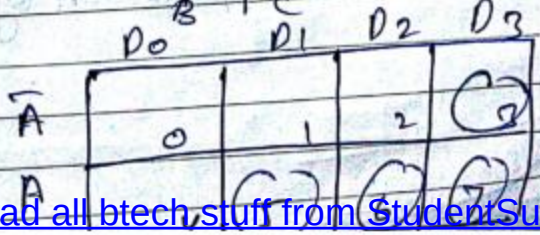
Carry $\Sigma m(3,5,6,7)$ 8:1 mux

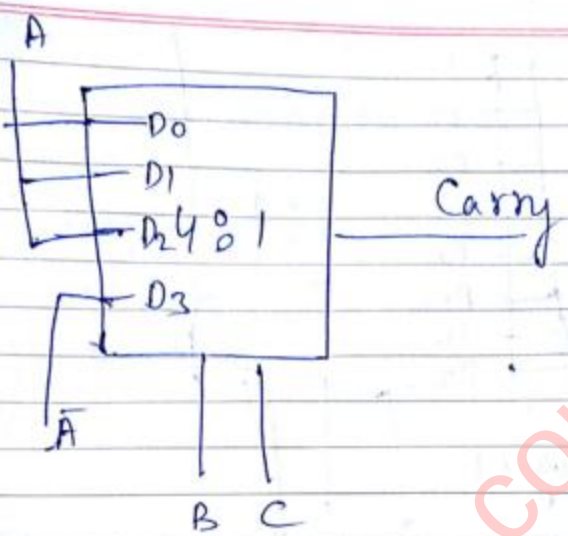




Implementing full adder using 4:1 mux.

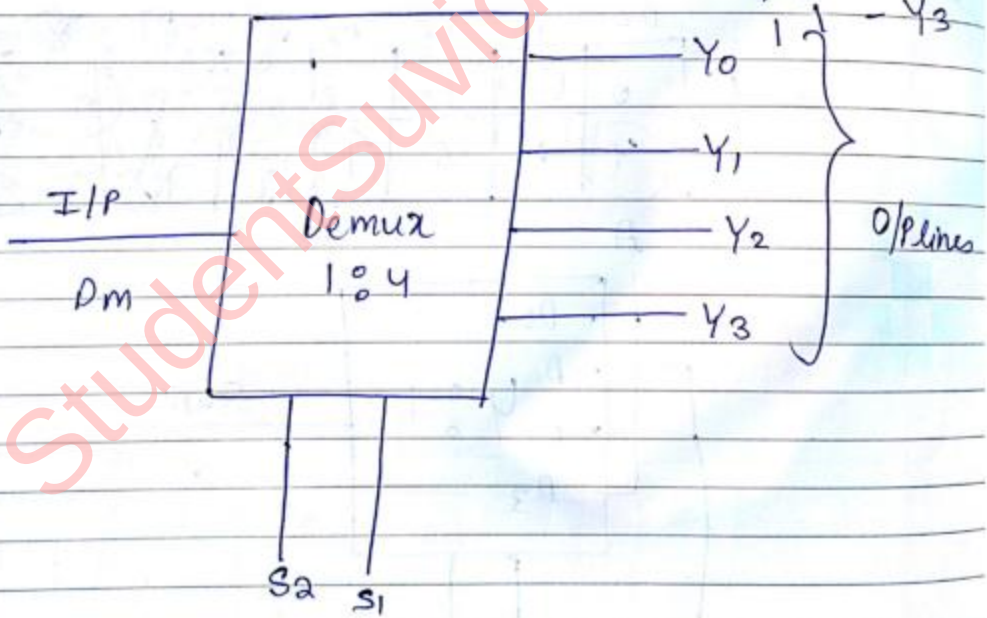






S_0	S_1	
0	0	Y_0
0	1	Y_1
1	0	Y_2
1	1	Y_3

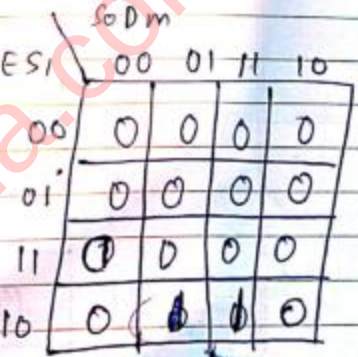
Demux



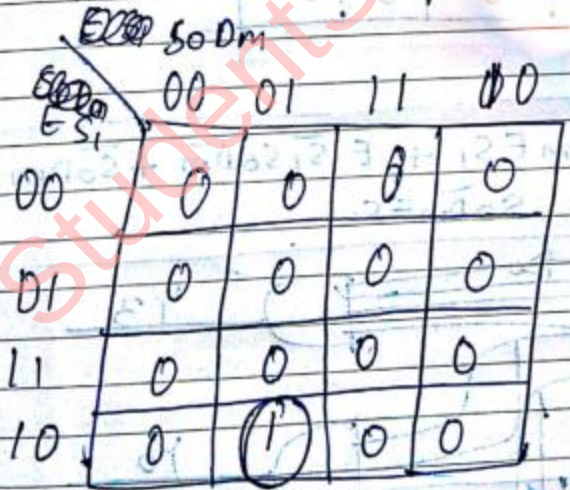
It performs reverse operation of mux.

E	S1	S0	Y0	Dm	Y0	Y1	Y2	Y3
0	X	X	0	X	0	0	0	0
1	0	0	Dm	0	0	0	0	0
1	0	1	0	1	1	Dm	0	0
1	1	0	0	0	0	0	Dm	0
1	1	1	0	1	0	0	0	Dm

	E	S1	S0	Dm	Y0	Y1	Y2	Y3
8	0	X	X	X	0	0	0	0
9	1	0	0	0	0	0	0	0
10	1	0	0	1	1	0	0	0
11	1	0	1	0	0	0	0	0
12	1	0	1	1	0	0	0	0
13	1	1	0	0	0	0	0	0
14	1	1	0	1	0	0	1	0
15	1	1	1	0	0	0	0	0
16	1	1	1	1	0	0	0	1



$Y_1 = \overline{E} S_1 S_0 \overline{Dm}$



$Y_0 = \overline{S_0} Dm \overline{E} \overline{S_1}$

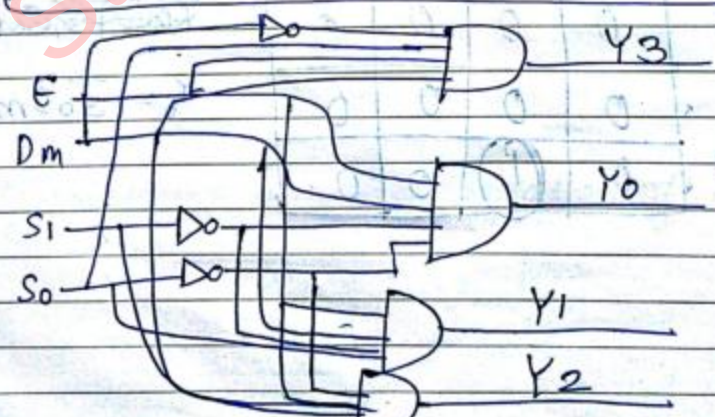
ES_1 \ S_0Dm	00	01	11	10
00	0	0	0	0
01	0	0	0	0
11	0	1	0	0
10	0	0	0	0

$$Y_2 = \bar{S}_0 Dm ES_1$$

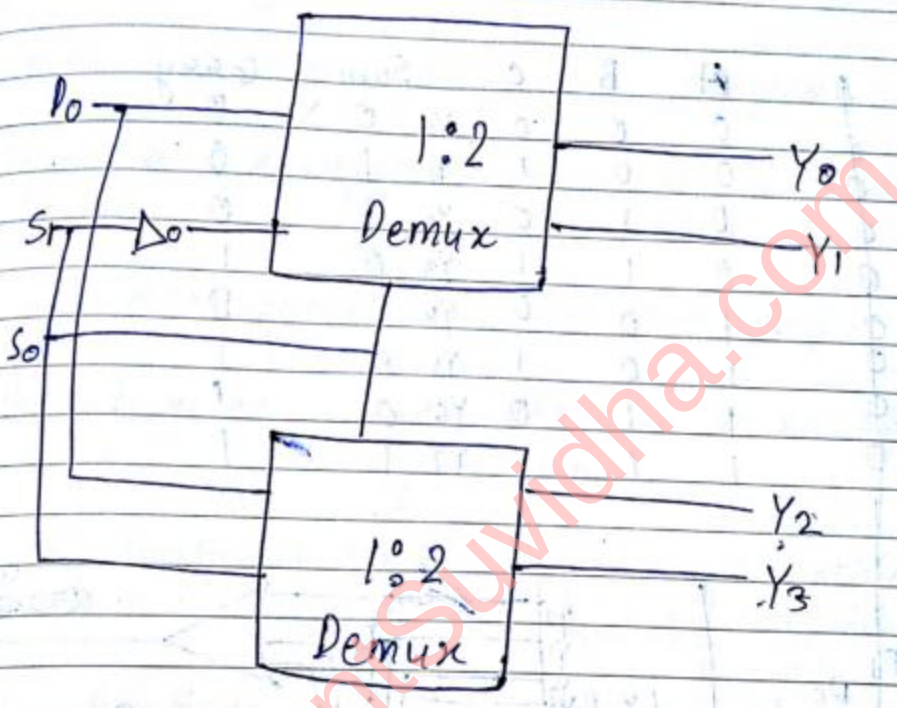
ES_1 \ S_0Dm	00	01	11	10
00	0	0	0	0
01	0	0	0	0
11	0	1	0	1
10	0	0	0	0

$$Y_3 = \bar{S}_0 Dm ES_1$$

$$Y = \bar{S}_0 Dm ES_1 + E\bar{S}_1 S_0 \bar{Dm} + \bar{S}_0 Dm ES_1 + S_0 \bar{Dm} ES_1$$

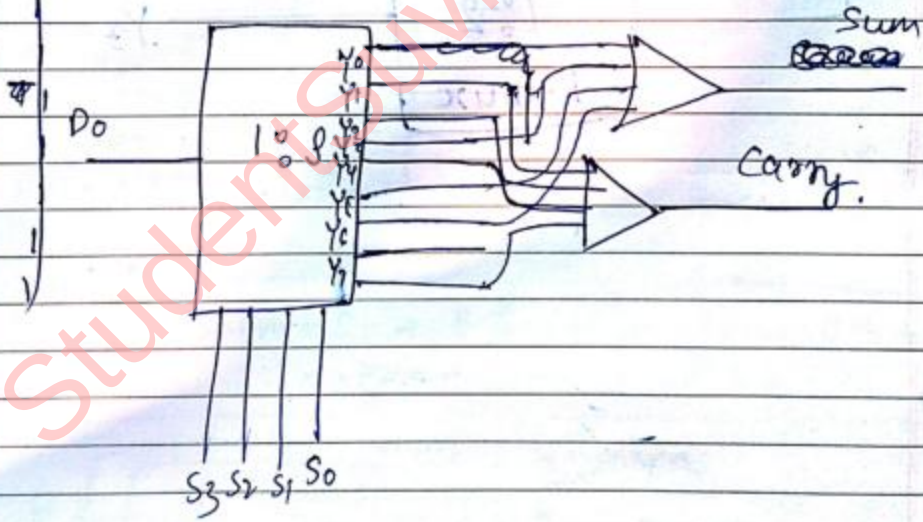


Demux Tree



Implement full adder using Demux

A	B	C		Sum	Carry
0	0	0	Y_0	0	0
0	0	1	Y_1	1	0
0	1	0	Y_2	1	0
0	1	1	Y_3	0	1
1	0	0	Y_4	1	0
1	0	1	Y_5	0	1
1	1	0	Y_6	0	1
1	1	1	Y_7	1	1



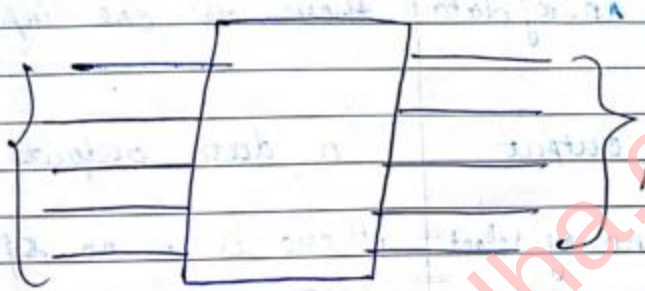
~~Encoder~~

Comparison of mux and demux

mux	demux
there is n no. of data input	there is one input.
one data output	n data outputs
there is m no. of select lines.	there is m no. of select lines.
combinational logic circuit	It is also a combinational logic circuit.
the relation b/w input & output & select lines is $n = 2^m$	the relation b/w input, output & select line is $n = 2^m$
Its operating principle is many to one.	Its operating principle is one to many.
It is also known as data selector	It is known as data distributor.
mux is used in communication system	demux is also used in communication system.

Encoder

is a combinational circuit which is design to perform inverse operation of the decoder.

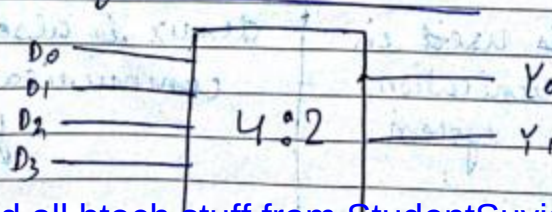


there is no selector in encoder and decoder.

Types of Encoder

- Priority Encoder
- Decimal to BCD "
- Octal to Binary "
- hexa decimal to "

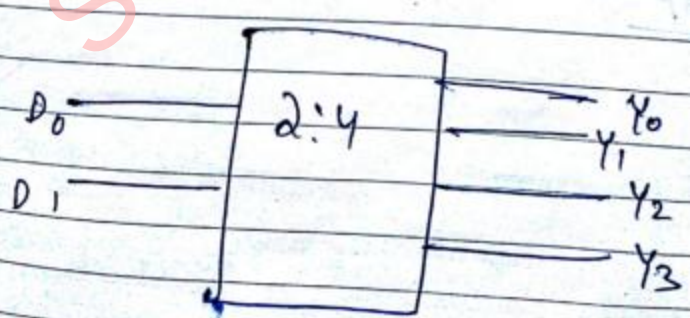
Priority Encoder



Truth Table

D_3	D_2	D_1	D_0	Y_0	Y_1
0	0	0	1	0	0
0	0	1	0	0	1
0	1	X	X	1	0
1	X	X	X	1	1

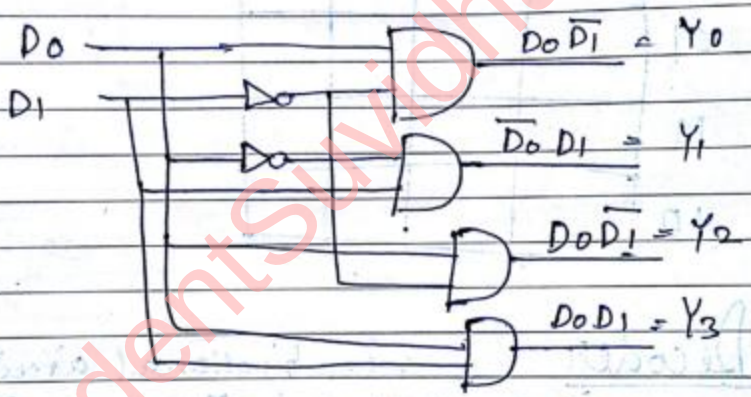
Decoder is a combinational circuit it has n no. of input & 2^n output



RAM, ROM, PLA, PAL -

D_0	D_1	Y_0	Y_1	Y_2	Y_3
0	0	1	0	0	0
0	1	0	1	0	0
1	0	0	0	1	0
1	1	0	0	0	1
		$D_0\bar{D_1}$	$\bar{D_0}D_1$	$D_0\bar{D_1}$	D_0D_1

$$Y = D_0\bar{D_1} + \bar{D_0}D_1 + D_0\bar{D_1} + D_0D_1$$



K-Map

$F(A,B,C,D) = \Sigma m(1,5,6,12,13,14) + d(2,4)$

Step 1

grp	Minterm	Representation A	B	C	D
1	1	0	0	0	1
2.	* 5	0	1	0	1
	* 6	0	1	1	0
	12	1	1	0	0
3.	13	1	1	0	1
	14	1	1	1	0

0
1