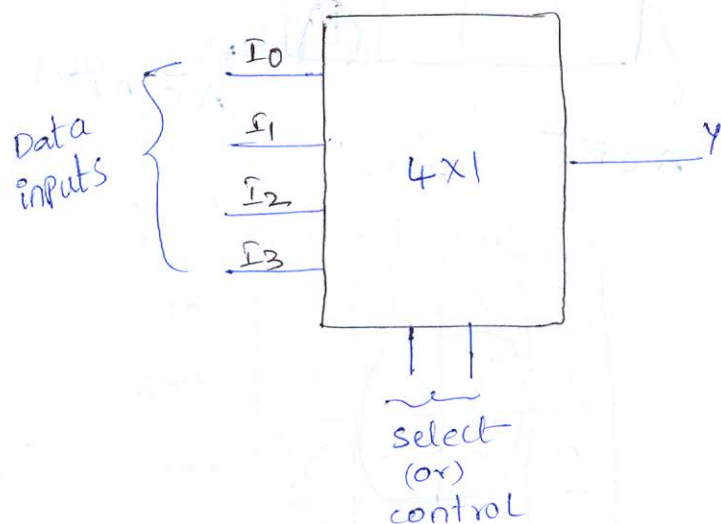


23/06/11.

MULTIPLEXER

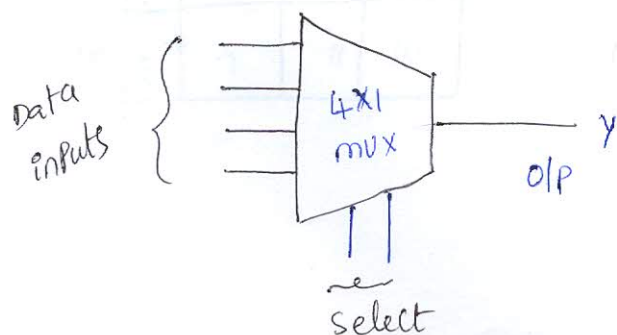
→ Combination ckt.

many i/p - one o/p.



→ mux is a combinational ckt. which have many i/p's and one o/p. depending on control or select input one of the data input is transferred to the output hence it is known as "many to one ckt", "data selector" (or) "universal logic gate" (or) "parallel to serial data converter".

→

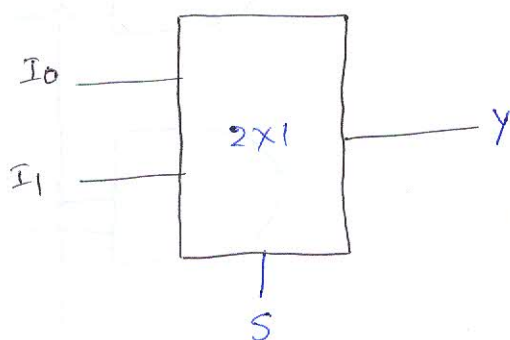


→ let m is no. of data inputs, n - no. of select lines then relation between them is

$$m = 2^n$$

$$\rightarrow n = \log_2 m$$

⇒ 2x1 mux :-



⇒ Truth Table :-

select input (S)	O/P (Y)
0	I_0
1	I_1

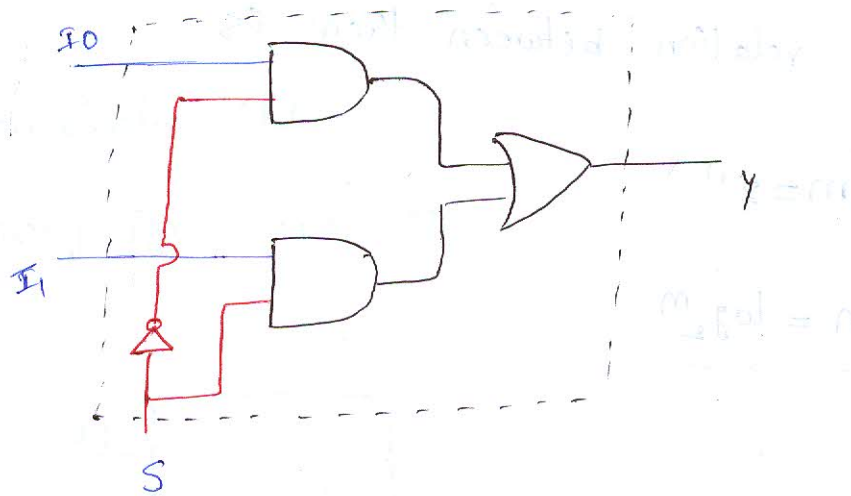
⇒ Logical Exp :-

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

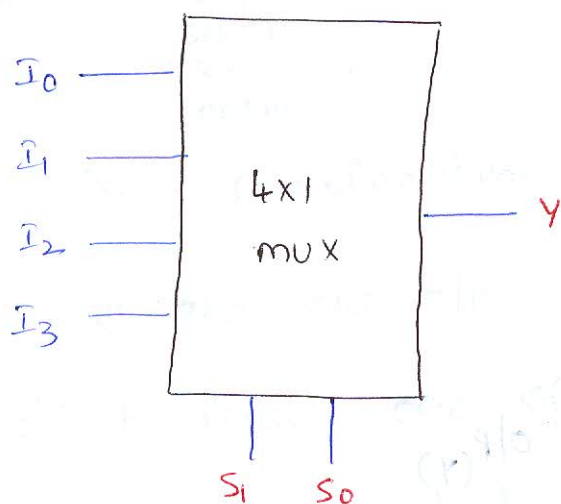
↓
L.S.B

↓
M.S.B

S	I ₀	I ₁	Y
0	0	0	0
0	1	0	1
1	0	1	1
1	1	1	1



⇒ 4x1 mux :-

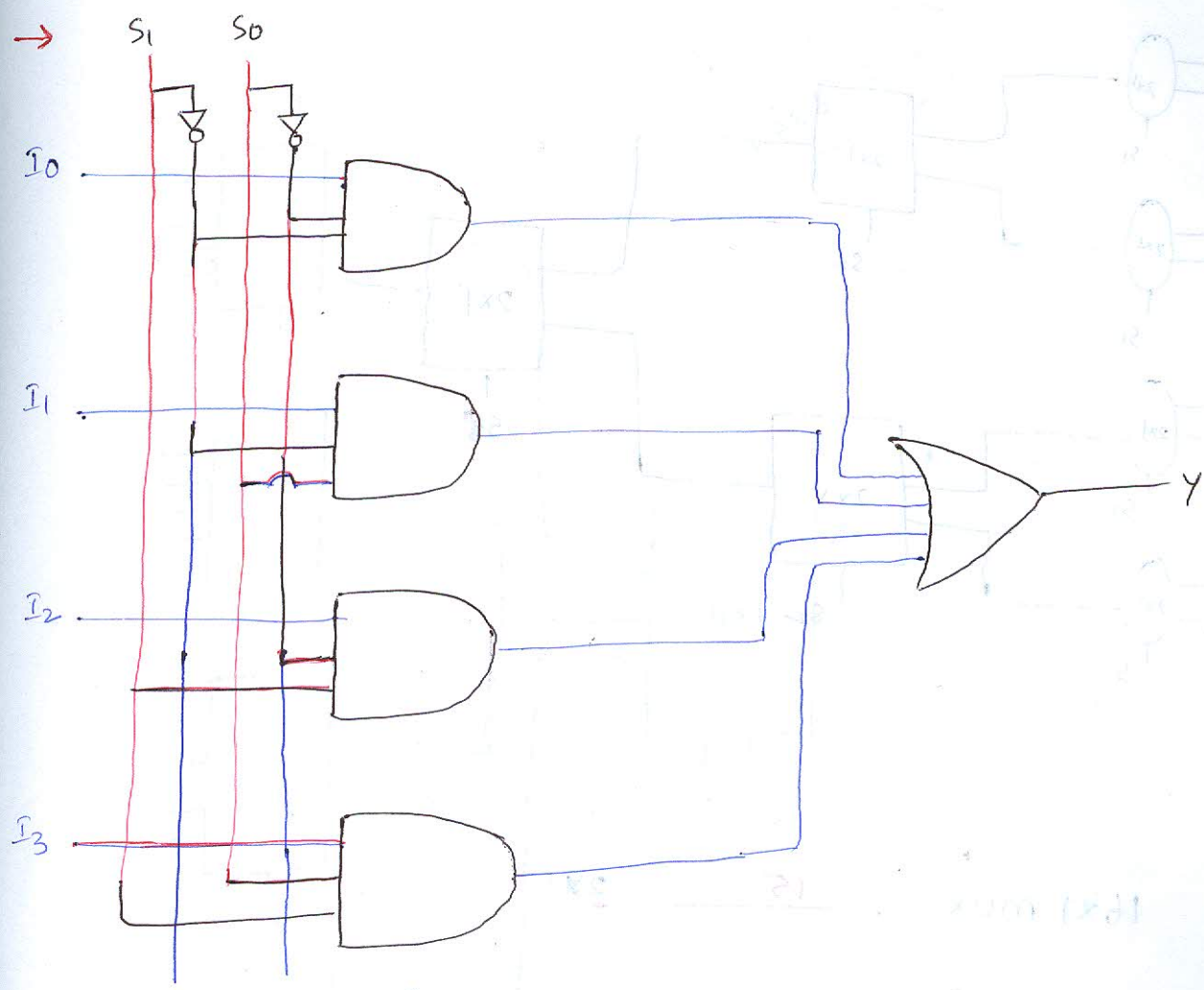


⇒ Truth Table :-

<u>S_1</u>	<u>S_0</u>	<u>Y</u>
0	0	I_0
0	1	I_1
1	0	I_2
1	1	I_3

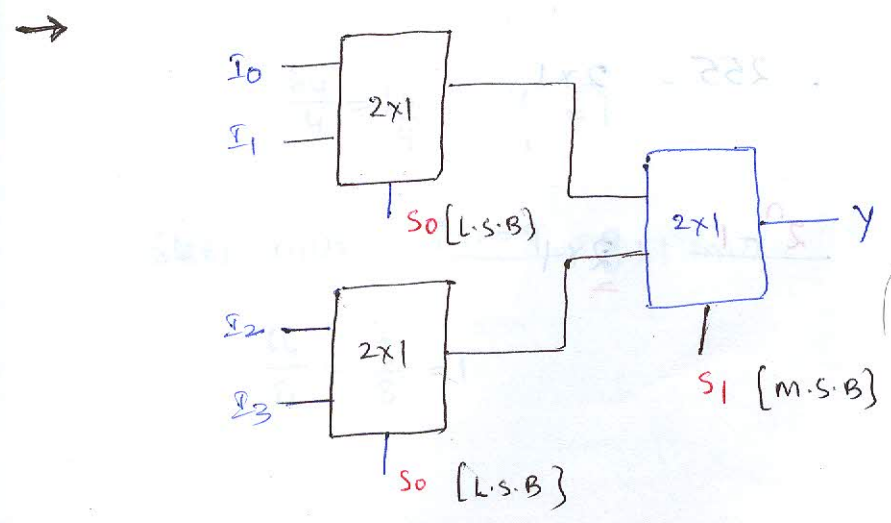
$$Y = I_0 \bar{S}_1 \bar{S}_0 + I_1 \bar{S}_1 S_0 + I_2 S_1 \bar{S}_0 + I_3 S_1 S_0$$

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



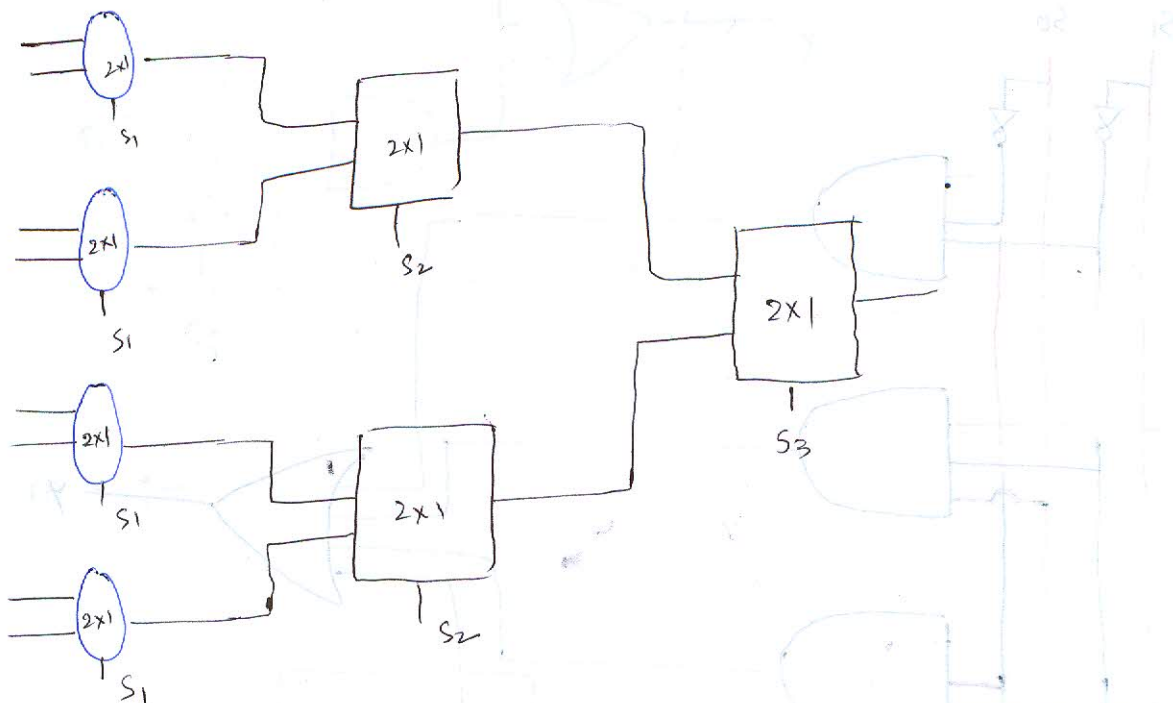
⇒ Implementation of higher order mux using lower order :-

1) 4x1 mux ——— 2x1



→ 8x1 mux $\frac{4+2+1=7}{2} = 2x1$

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



→ 16x1 mux $\frac{15}{2} = 2x1$

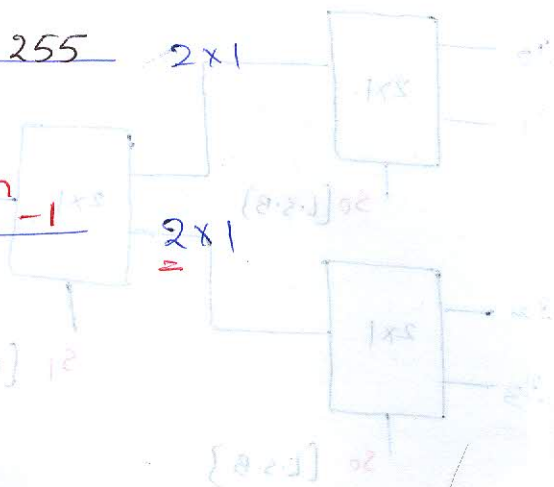
$$\frac{16}{2} = 8 \Rightarrow \frac{8}{2} = 4 \Rightarrow \frac{4}{2} = 2 \Rightarrow \frac{2}{2} = 1$$

$$8+4+2+1=15$$

→ 64x1 mux $\frac{63}{2} = 2x1$

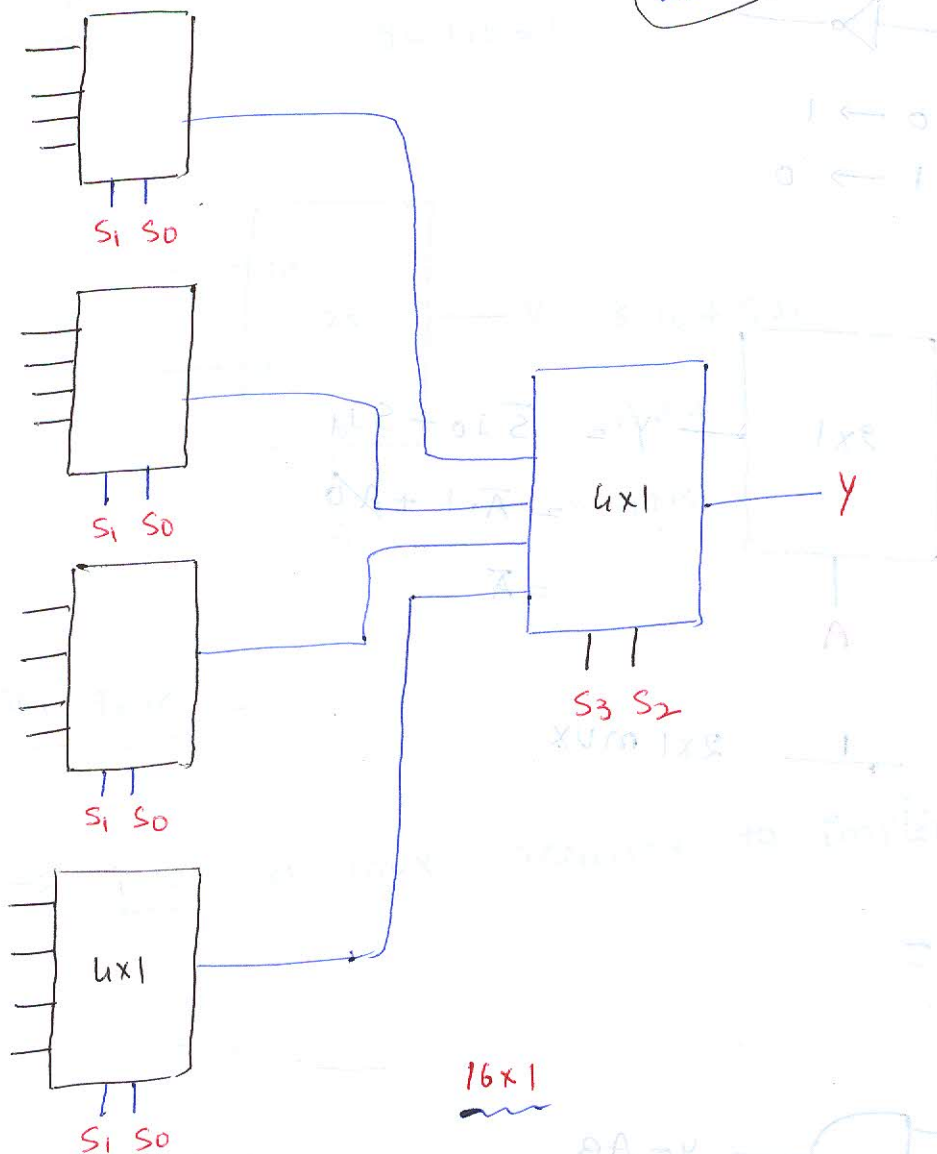
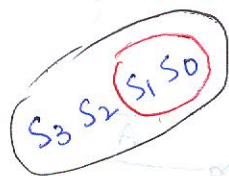
→ 256x1 mux $\frac{255}{2} = 2x1$

→ $2^n \times 1$ mux $\frac{2^n - 1}{2} = 2x1$



⇒ OR 16x1 mux $\frac{4+1=5}{4}$ 4x1 mux

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



→ 64x1 mux $\frac{21}{4}$ 4x1 mux

$$\frac{64}{4} = \frac{16}{4} = \frac{4}{4} = 1$$

→ 64x1 mux $\frac{8+1=9}{8}$ 8x1 mux

$$\frac{64}{8} = \frac{8}{8} = 1$$



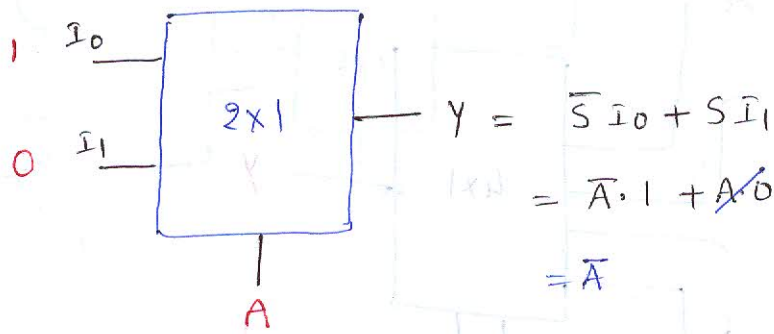
⇒ mux as universal ckt :-

⇒ Not gate :-



0 → 1
1 → 0

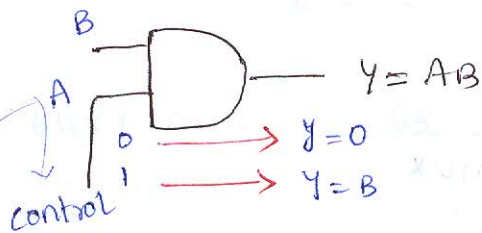
→



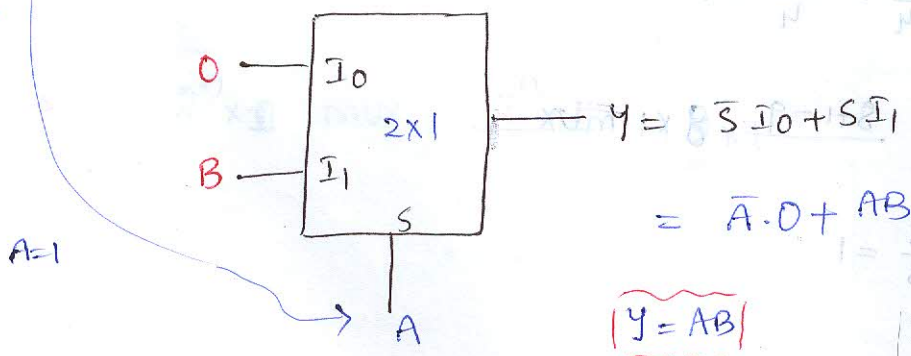
→

NOT ——— 1 ——— 2x1 mux

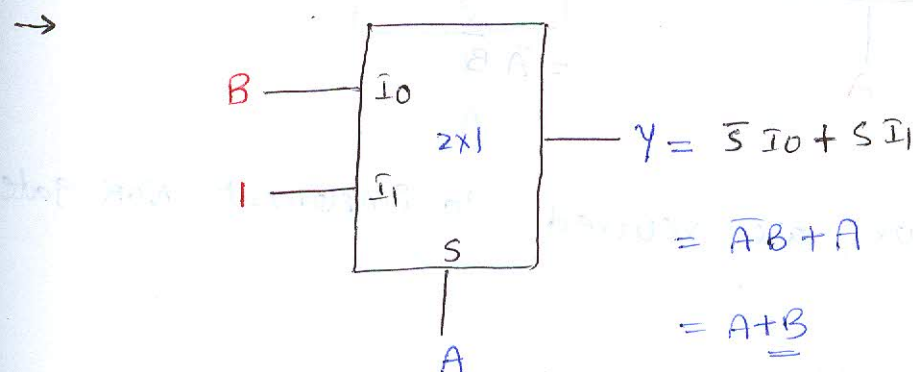
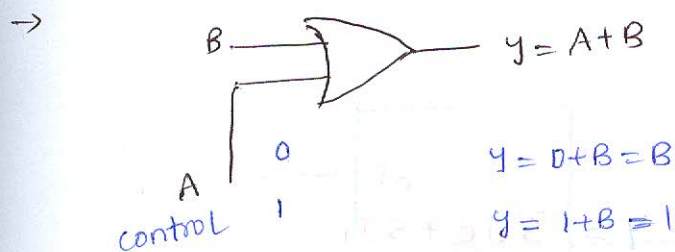
⇒ And gate :-



⇒

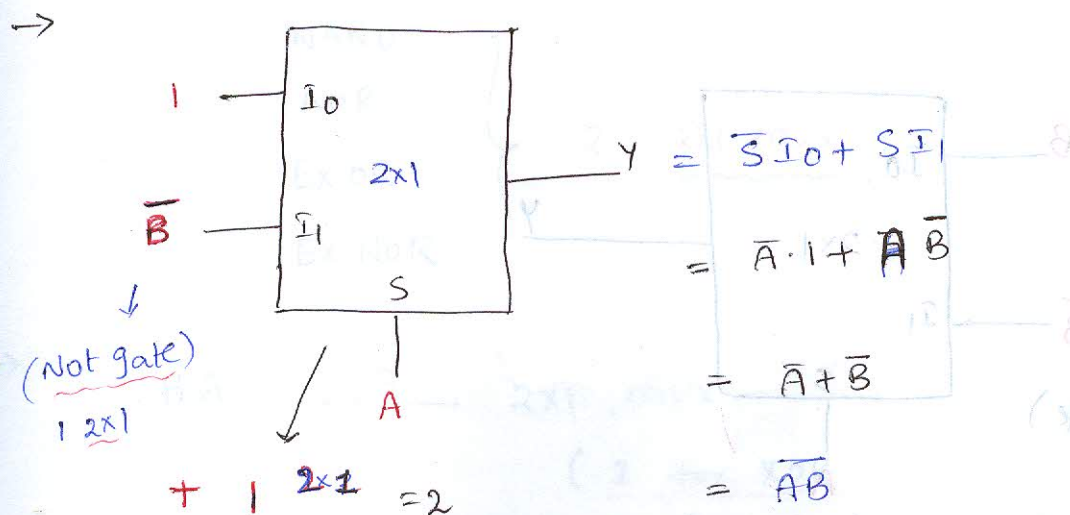


⇒ OR gate :-

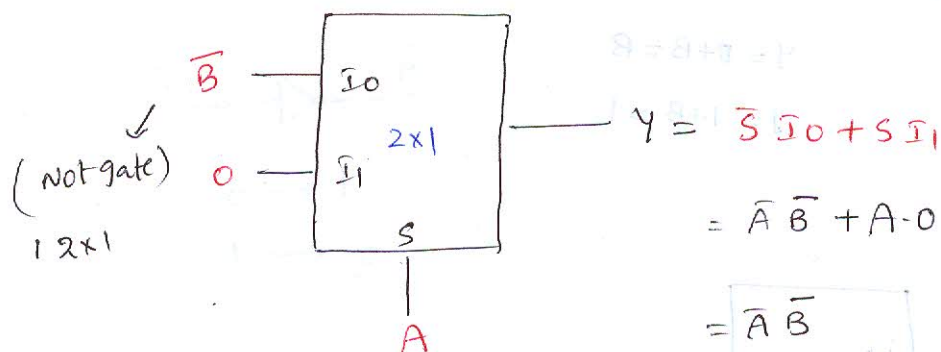


⇒ NAND gate :-

→ Two 2x1 mux required to implement NAND gate

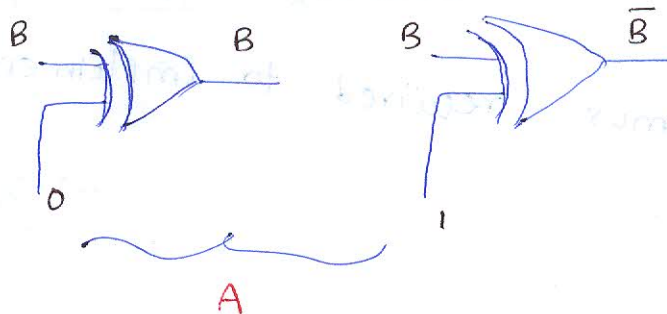


⇒ NOR gate :-

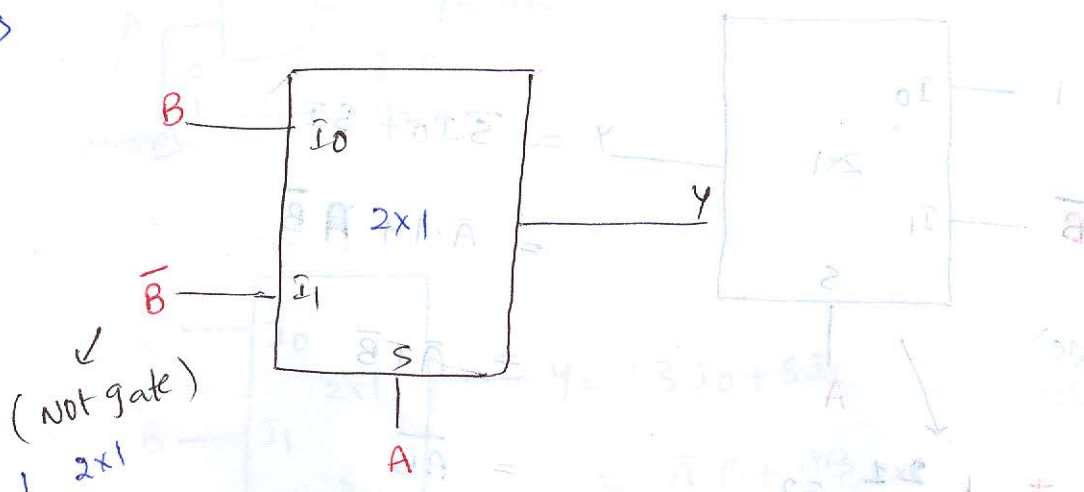


→ Two 2x1 mux are required to implement NOR gate.

XOR gate :-

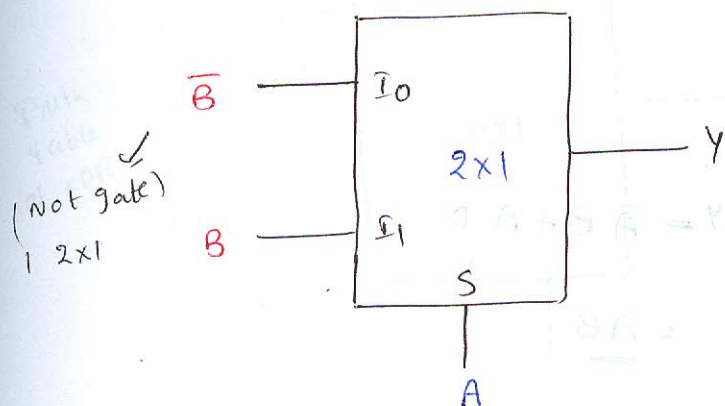


→



→ Two 2x1 mux are required to implement XOR

⇒ XNOR gate :-



Q) TO implement 2 input XOR gate and AND min No. of 2x1 mux are required 2, 1

NOT

AND

OR

1 2x1 mux

NAND

NOR

Ex OR

Ex NOR

2 2x1 mux

HA

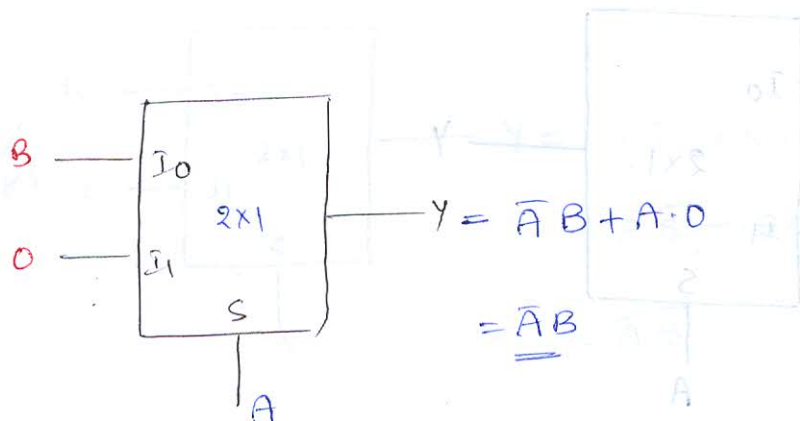
3

2x1 mux

(2 for XOR
1 for AND)

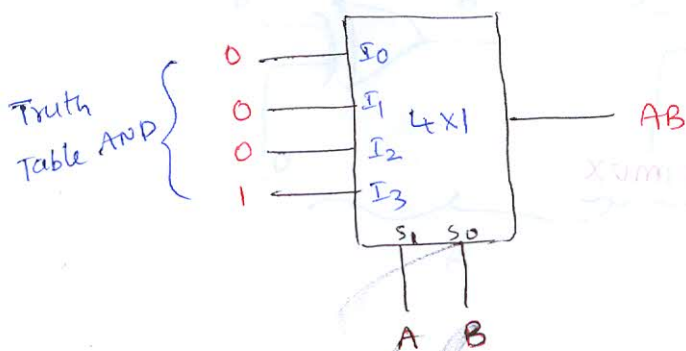
→ HS 3 2x1 MUX

$$\begin{aligned} \text{Sum} &= A \oplus B \rightarrow 2 \\ \text{Borrow} &= \overline{A}B \rightarrow 1 \end{aligned} \quad \left. \vphantom{\begin{aligned} \text{Sum} \\ \text{Borrow} \end{aligned}} \right\} \textcircled{3}$$

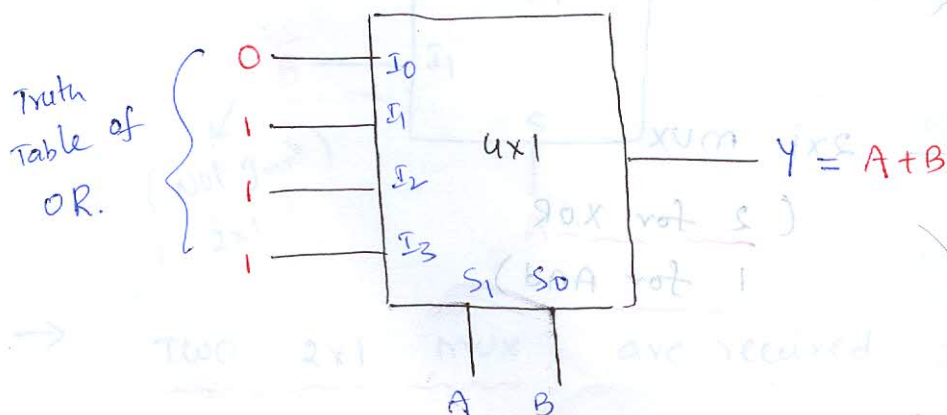


⇒ 4x1 mux :-

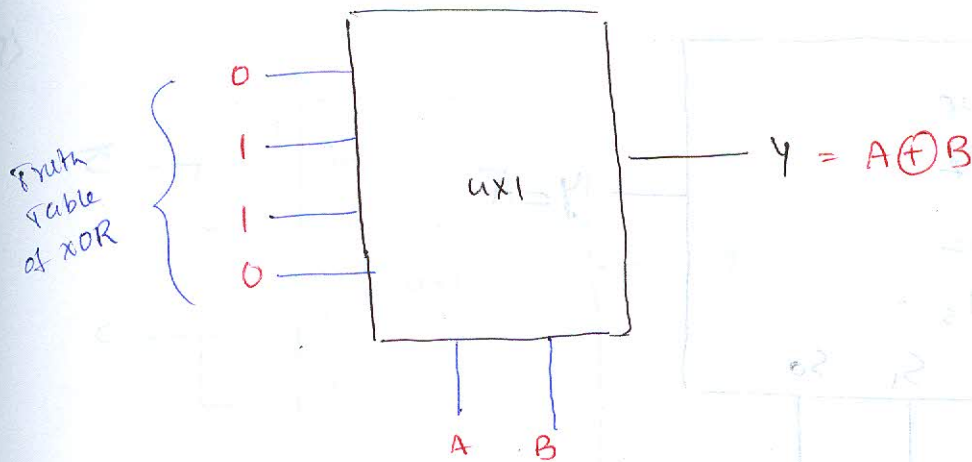
⇒ And :-



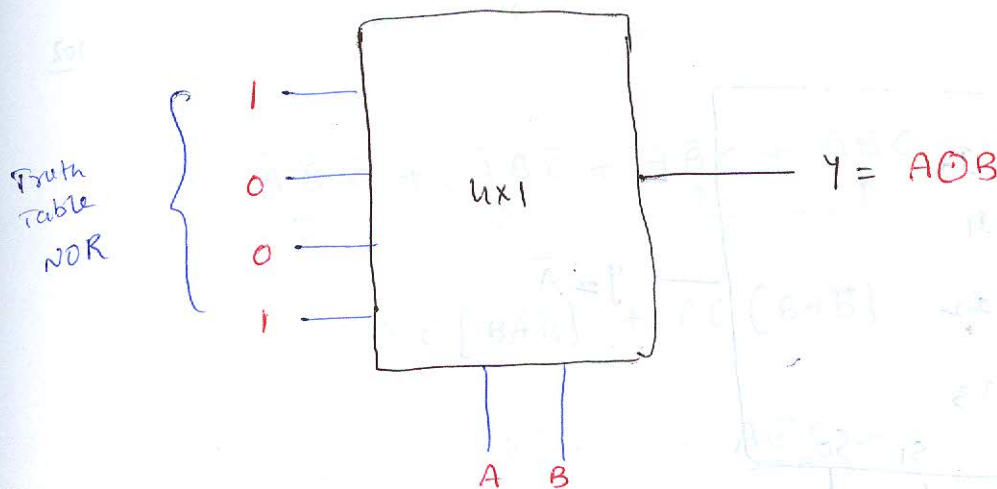
⇒ OR :-



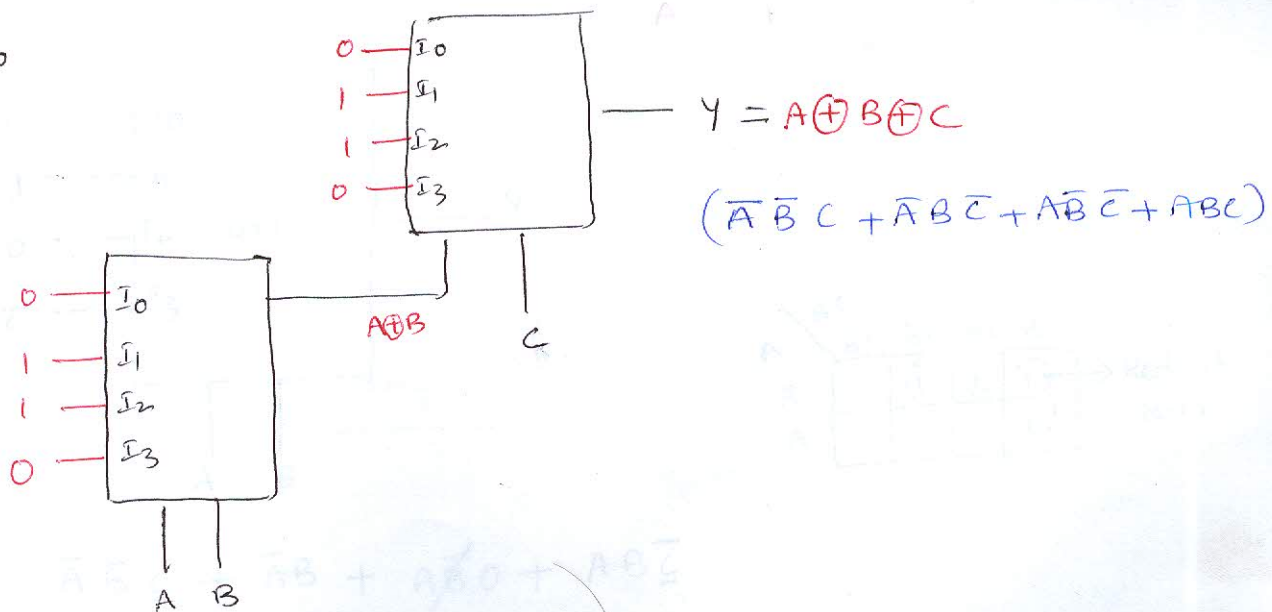
Ex-OR :-



Ex NOR :-

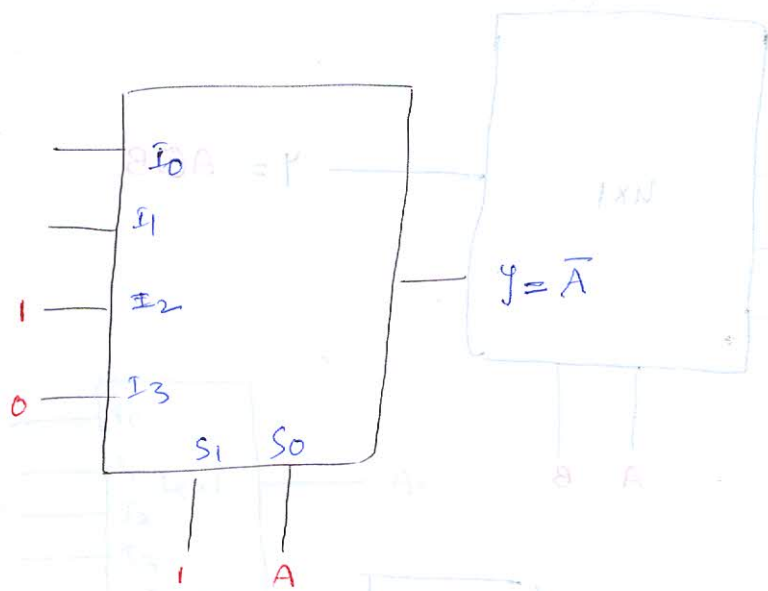
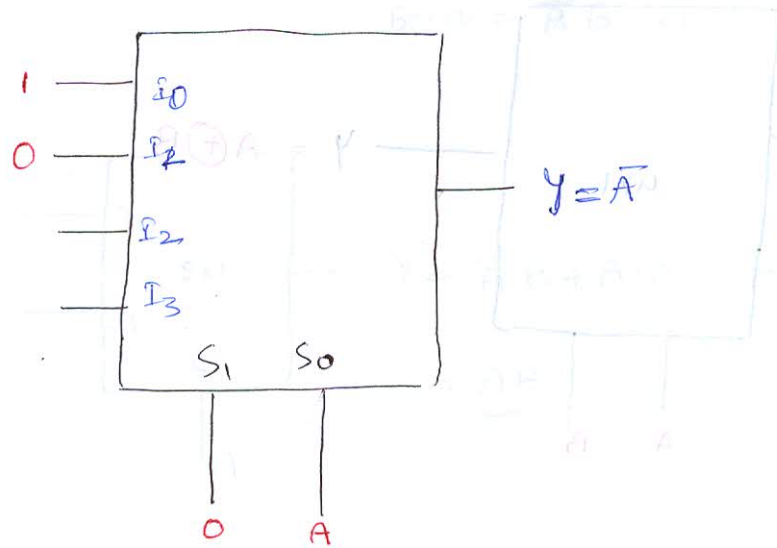


Gate 2007
Q3 IES 2010



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

⇒ Not gate :-



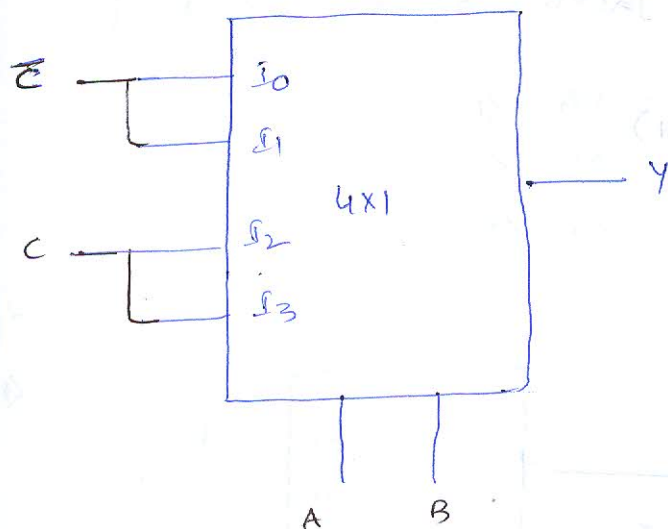
$$(1 \cdot 1 \cdot \bar{A} + 0 \cdot \bar{A} + 1 \cdot \bar{A} + 0 \cdot A) = \bar{A}$$



III

Determine minimized o/p logical exp :-

Q)



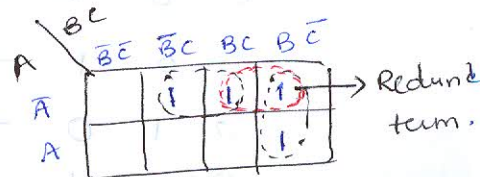
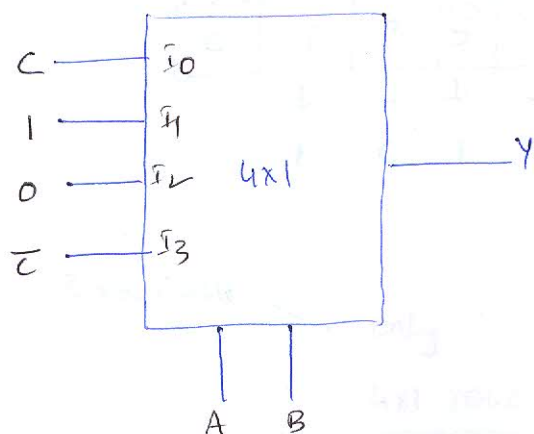
Sol

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

$$\bar{A}\bar{C}[B/\bar{B}] + AC[B/\bar{B}]$$

$$\bar{A}\bar{C} + AC = \underline{A \oplus C}$$

Q)



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

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

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

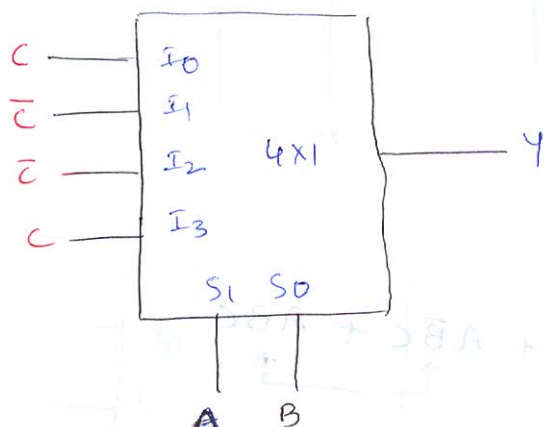
$$= \bar{A}[\bar{B}/B] + B\bar{C}[A/\bar{A}] = \bar{A}\bar{C} + B\bar{C}$$

(iv) Design (or) implementation of logical expression:-

Q) implement $\bar{A}\bar{B}C + \bar{A}B\bar{C} + A\bar{B}\bar{C} + ABC$ using 1 4x1 mux and additional logical gates.

Sol:-

$$f(A, B, C) = \sum m(1, 2, 4, 7)$$



→

	I_0	I_1	I_2	I_3
$\bar{C}$	0	2	4	6
C	1	3	5	7
	$\bar{C}$	$\bar{C}$	$\bar{C}$	$\bar{C}$

→

A B C (L.S.B)

0	0	0	→ $\bar{C}$
0	0	1	→ C
0	1	0	→ $\bar{C}$
0	1	1	→ C
1	0	0	→ $\bar{C}$
1	0	1	→ C
1	1	0	→ $\bar{C}$
1	1	1	→ C

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

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

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

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

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

$$= 1$$

→ 3 variables $\begin{cases} \text{one } 4 \times 1 \text{ mux} \\ \text{one not gate} \end{cases}$

Q) $f(A, B, C) = \sum m(0, 1, 3, 5, 4, 7)$ using 4x1 mux

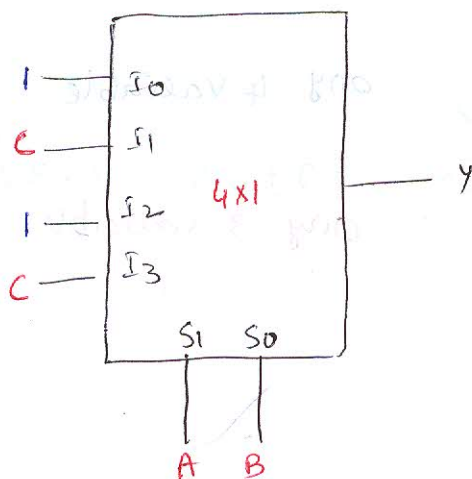
1) AB as select

2) AC "

3) BC "

Sol:-

1)



	I_0	I_1	I_2	I_3
$\bar{C}$	0	2	4	6
C	1	3	5	7
	↓	↓	↓	↓
	1	C	1	C

→ 3 variable

→ only 4x1 mux

⇒ Note :-

using one 4x1 mux we can implement two

variable functions and some of three variable functions

↓
Some

→

One 4x1 mux

&

one NOT

any 2 variable
any 3 variable

→

One 8x1 mux

any 3 variable
Some of 4 variables

→

one 8x1 mux

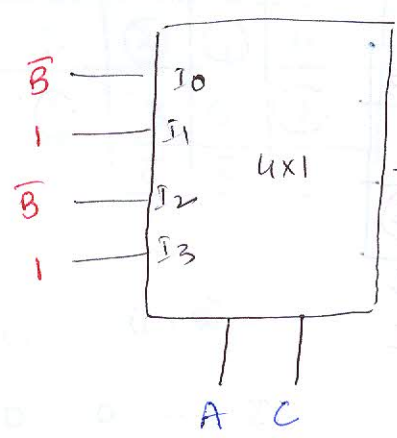
&

one NOT

any 4 variable
any 3 variable

②

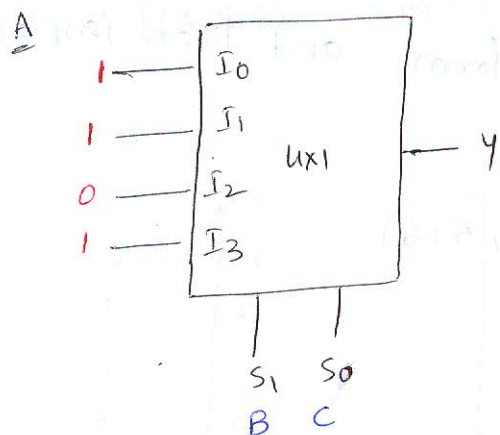
Ac as select line :-



	I_0	I_1	I_2	I_3
$\bar{B}$	0	1	4	5
B	2	3	6	7
	$\bar{B}$	1	$\bar{B}$	1

<u>A</u>	<u>B</u>	<u>C</u>	
0	0	0	→ $\bar{B}$
0	0	1	→ $\bar{B}$
0	1	0	→ $\bar{B}$
0	1	1	→ $\bar{B}$
1	0	0	→ $\bar{B}$
1	0	1	→ $\bar{B}$
1	1	0	→ B
1	1	1	→ B

③ BC as select line :-



Q) $f(A, B, C) = A\bar{B} + B\bar{C}$ using 4x1 MUX

Sol:-

⇒ Note:- mux always receive canonical form

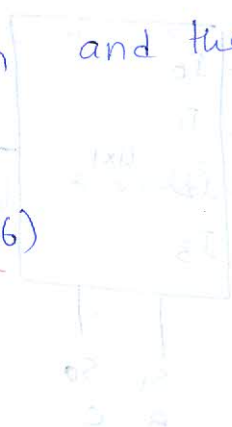
$\bar{A}$	0	2	4	6	8	10	12	14
A	1	3	5	7	9	11	13	15

Q) $f(A,B,C) = \prod M(0,1,4,7)$

Sol:- above expression in POS form so first

Convert it into SOP form and then implement

$f(A,B,C) = \sum m(2,3,5,6)$



$f(A,B,C) = \sum m(2,3,5,6)$

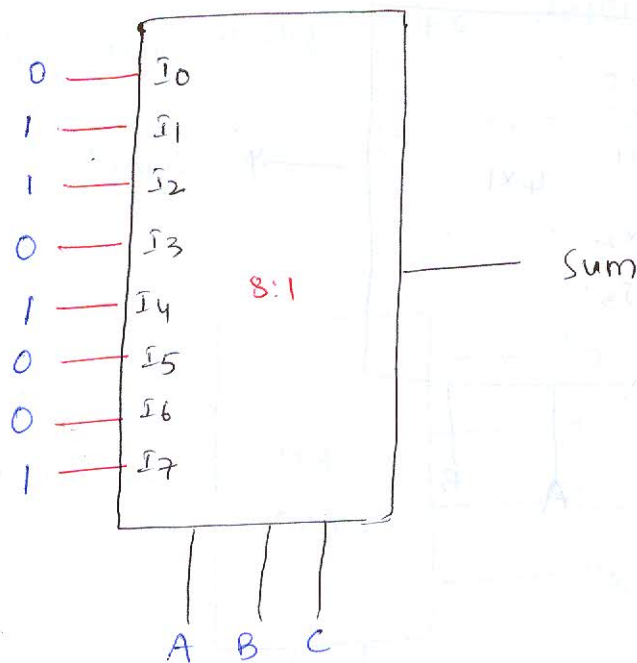


	0	1	2	3
0	0	0	0	0
1	0	1	1	0
2	0	1	0	1
3	0	0	1	1
4	0	0	0	0
5	0	1	1	0
6	0	1	0	1
7	0	0	1	1

Q) implement full adder sum expression using 8:1 mux.

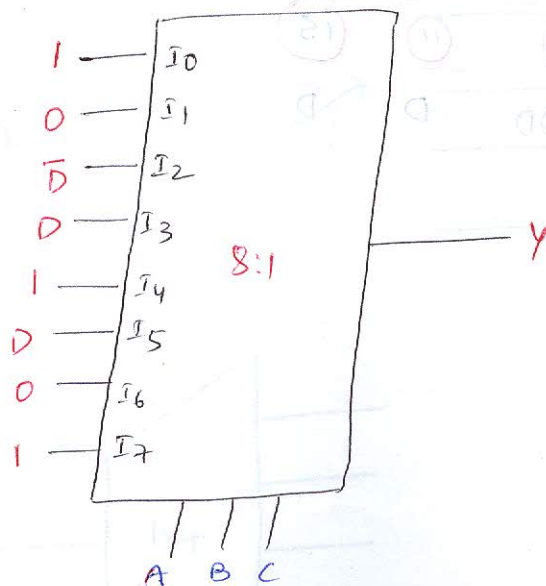
Sol:-

$$f(A, B, C) = \sum m(1, 2, 4, 7)$$



Q) implement $f(A, B, C, D) = \sum m(0, 1, 4, 7, 8, 9, 11, 14, 15)$ using 8:1 mux.

Sol:-

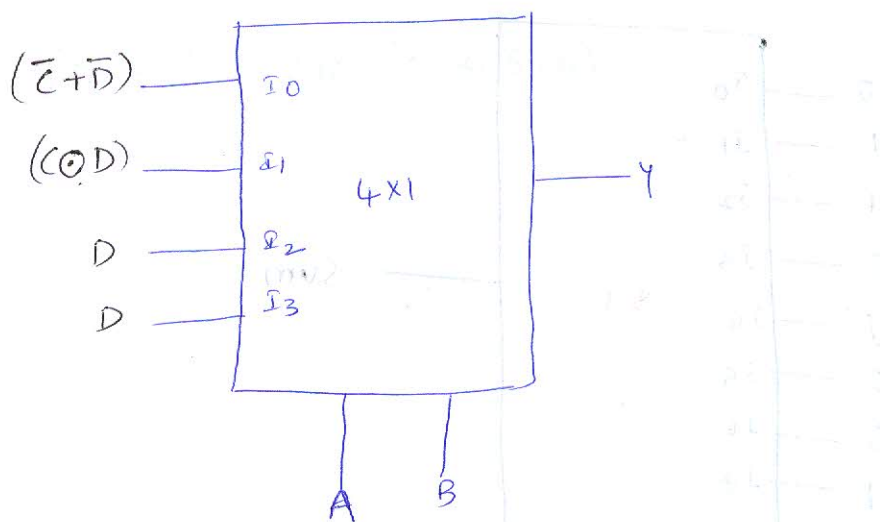


	I_0	I_1	I_2	I_3	I_4	I_5	I_6	I_7
$\bar{D}$	0	2	4	6	8	10	12	14
D	1	3	5	7	9	11	13	15

Q) $f(A, B, C, D) = \sum m(0, 1, 2, 4, 7, 9, 11, 13, 15)$ using 4x1 mux.

with $A, B(C, D)$ as selection lines.

↓ ↓
Selection i/p s
lines

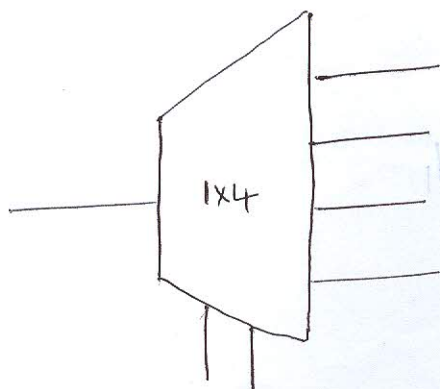
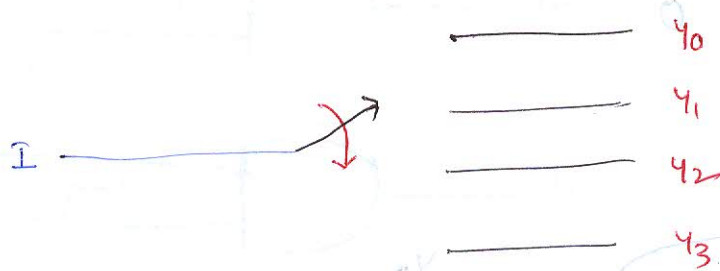
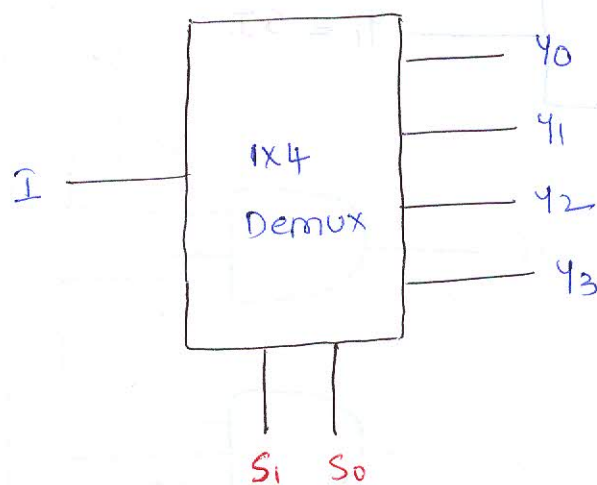
501

	I_0	I_1	I_2	I_3
$\bar{C}\bar{D}$	0	4	8	12
$\bar{C}D$	1	5	9	13
$C\bar{D}$	2	6	10	14
CD	3	7	11	15

24/06/11.

⇒ Demux :-

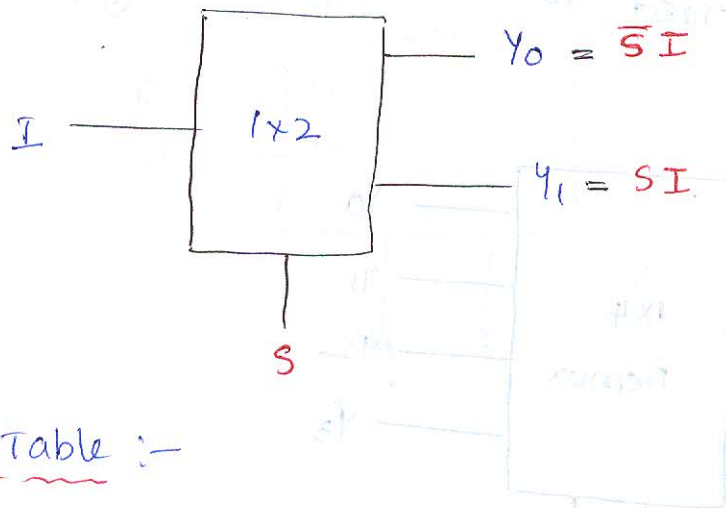
→ Demux is a Combinational ckt which have one data i/p and many o/p's. depending on select i/p data is transferred to one of the o/p line.



→ Demux is also called as one to many (or) data distributor.

⇒ 1x2 De mux :-

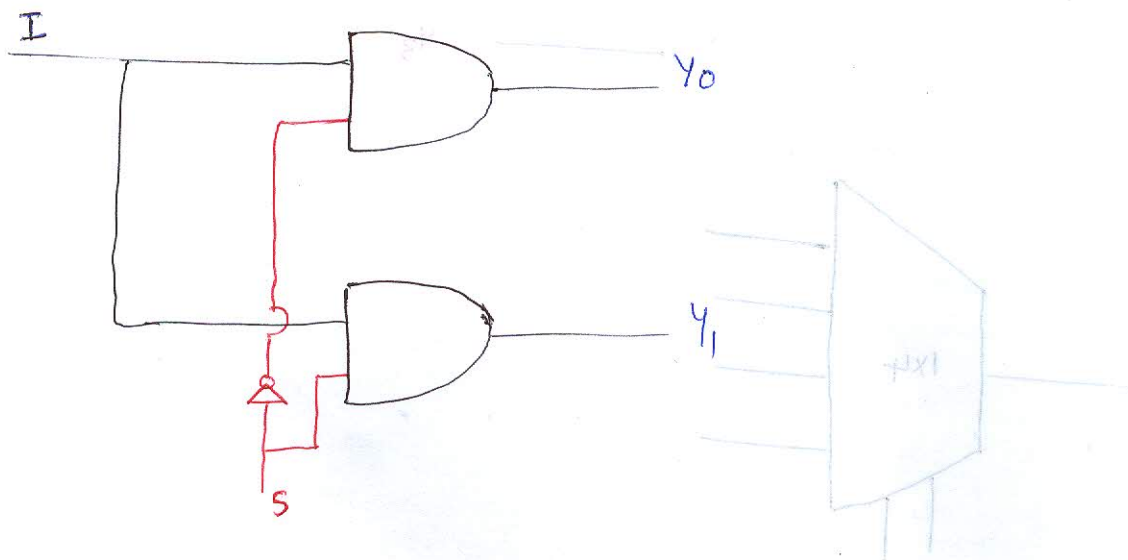
→



⇒ Truth Table :-

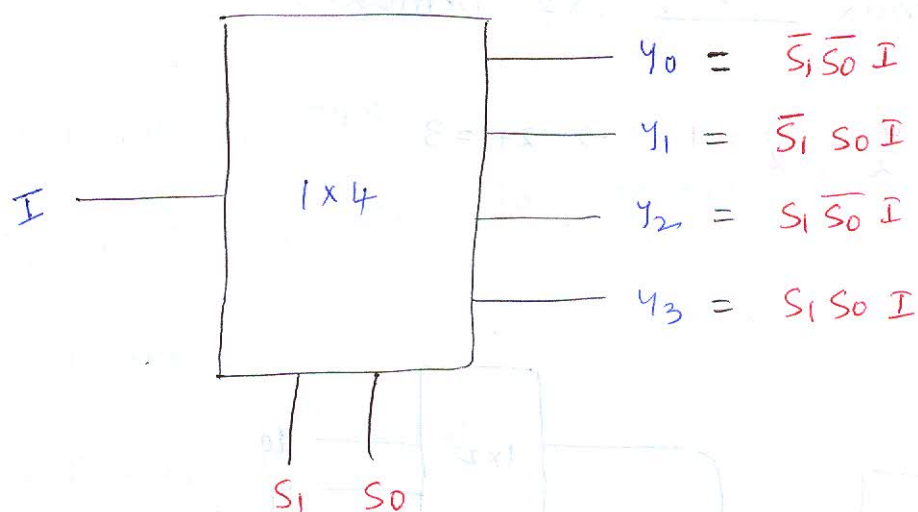
<u>S</u>	<u>Y₁</u>	<u>Y₀</u>
0	0	I
1	I	0

→

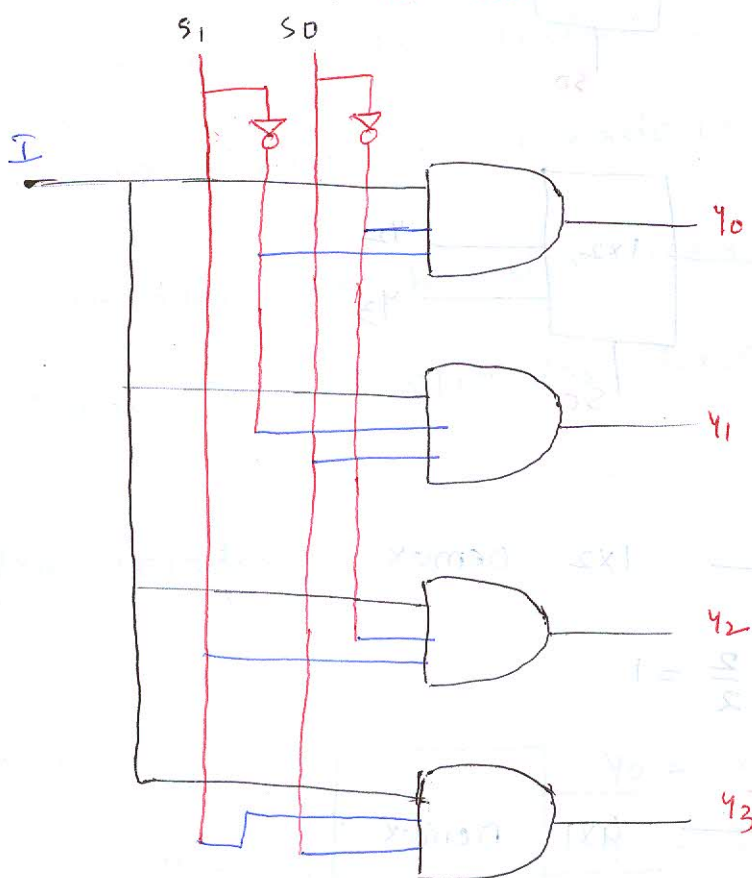


⇒ 1x4 Demux :-

→



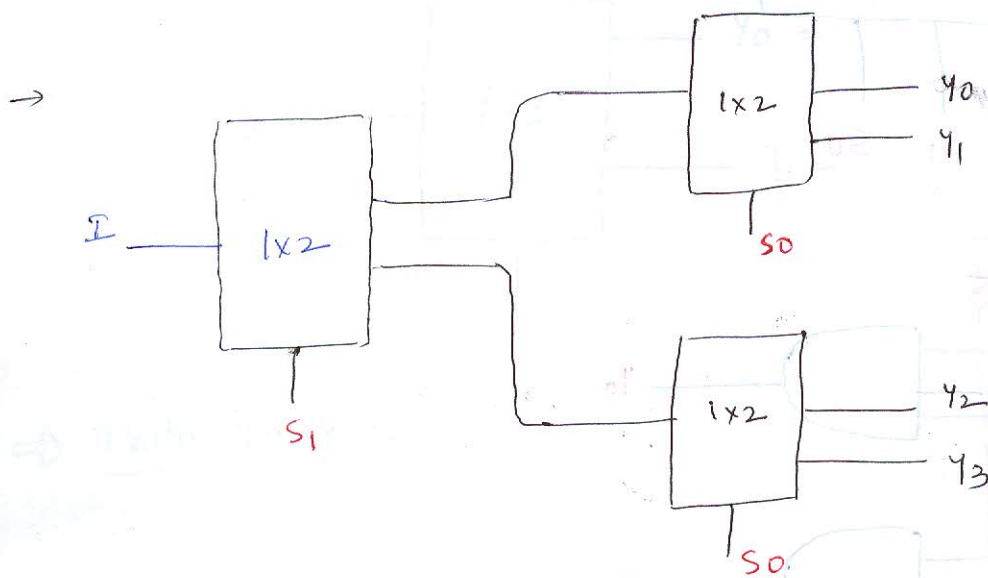
→



⇒ Higher order Demux ————— using lower Order :-

→ 1x4 Demux 3 1x2 Demux

$$\frac{4}{2} = \frac{2}{2} = 1 \rightarrow 2+1=3.$$



→ 1x8 Demux 7 1x2 Demux

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

→ 1x16 Demux 5 4x1 Demux

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

→ 1x64 Demux 1+4+16=21 1x4 Demux

→ 1x256 Demux 17 1x16