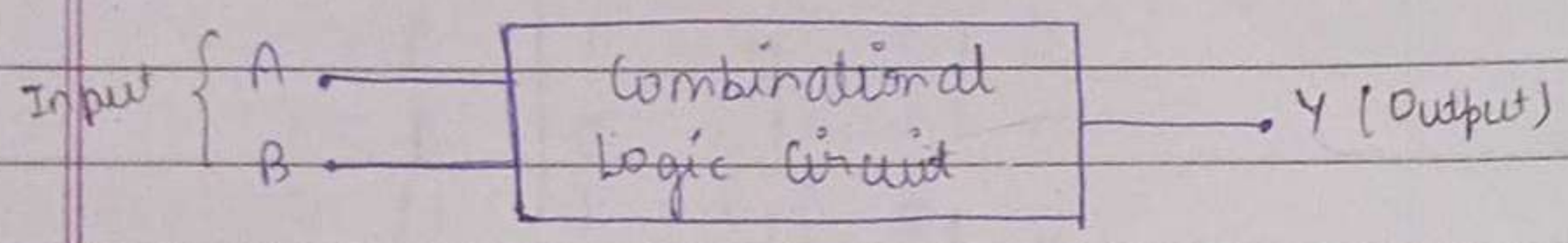
26/10/21

UNIT - 02

COMBINATIONAL LOGIC CIRCUIT

Combinational logic circuit –

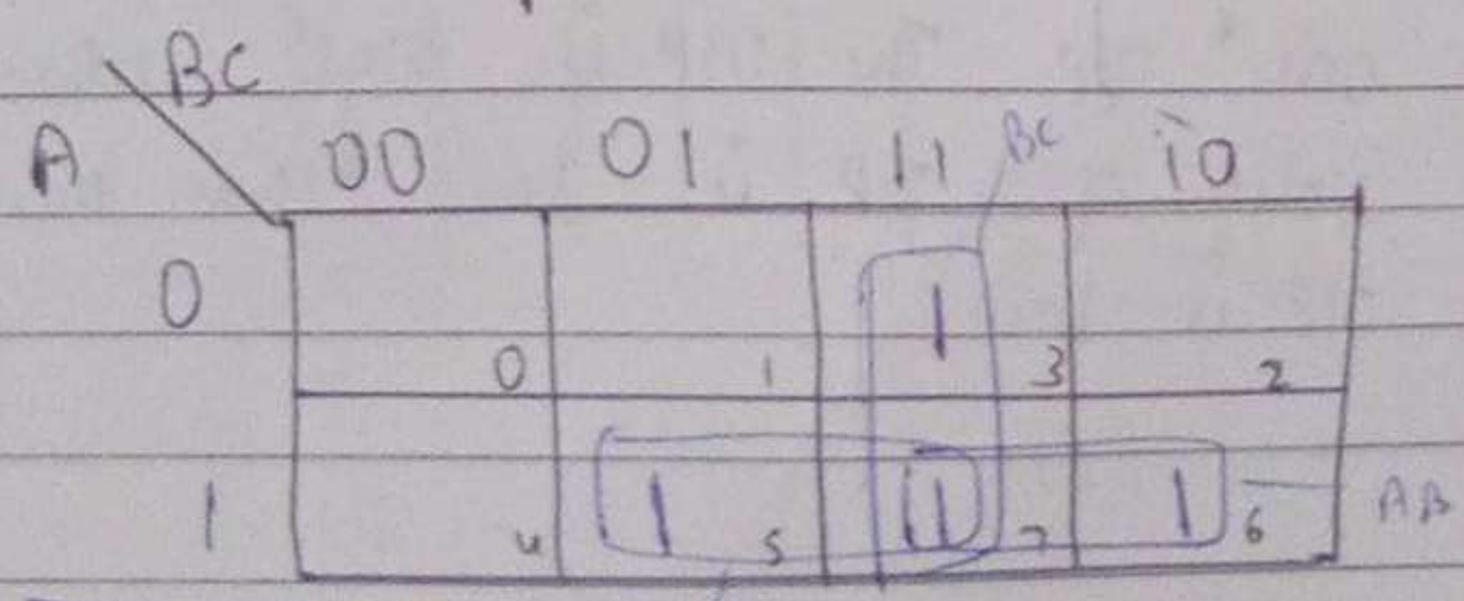


Combinational logic circuit is the circuit in which the O/P depends only on the present I/P and ~~NOT~~ not ~~and~~ on the Past O/P. No feedback is present.

Ex :- Adder, Substracter, Multiplexer, Encoder, Decoder, Demultiplexer etc.

Q Design a combinational logic circuit which has 3 input and one O/P. The O/P is high when the no. of input ~~are~~ ~~one~~ as 1 is two or more than two.

Ans:- Step II : K-map –

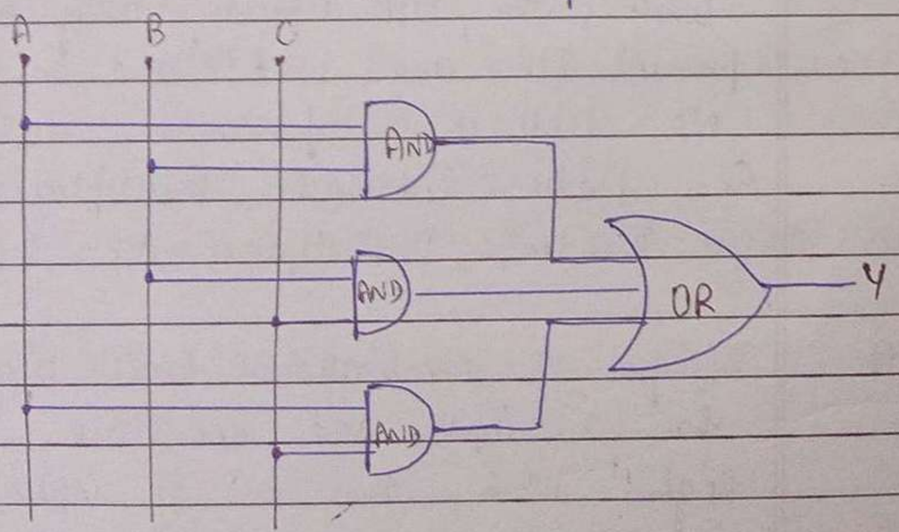


$$\boxed{Y = AB + BC + AC}$$

Step①: Input O/P

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

Step③: Circuit design: Using Basic Gates -



Que-2 Design a combinational logic circuit with 3 I/P & one O/P. The O/P is high when the Binary value of I/P is less than three. Otherwise the O/P is low.

Ans Step①:

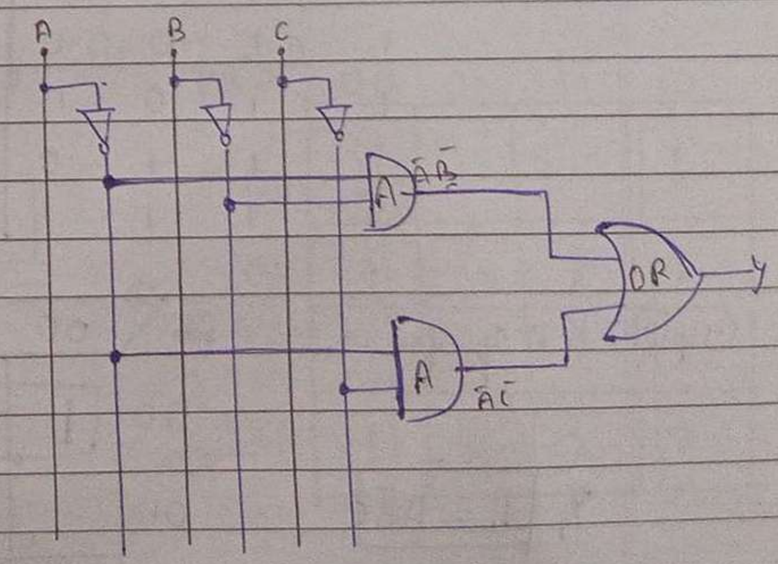
	A	B	C	Y	
0 =	0	0	0	0	1
1 =	0	0	1	1	1
2 =	0	1	0	1	1
3 =	0	1	1	1	0
4 =	1	0	0	1	0
5 =	1	0	1	1	0
6 =	1	1	0	1	0
7 =	1	1	1	0	0

Step②:

A \ BC	00	01	11	10
0	1 (0)	1 (1)	(3)	1 (2)
1	(4)	(5)	(7)	(6)

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

Step③:



③ Design a combinational logic circuit that generates 9's complement of a BCD I/P.

Sol^n: Step①:

I/P A (8)	B (4)	C (2)	D (1)	O/P BCD (9-Binary) f_1	f_2	f_3	f_4
0	0	0	0	1	0	0	1
0	0	0	1	1	0	0	0
0	0	1	0	0	1	1	1
0	0	1	1	0	1	1	0
0	1	0	0	0	1	0	1
0	1	0	1	0	1	0	0
0	1	1	0	0	0	1	1
0	1	1	1	0	0	1	0
1	0	0	0	0	0	0	1
1	0	0	1	0	0	0	0
1	0	1	0	X	X	X	X
1	0	1	1	X	X	X	X
1	1	0	0	X	X	X	X
1	1	0	1	X	X	X	X
1	1	1	0	X	X	X	X
1	1	1	1	X	X	X	X

Step②: K-map for f_1 —:

AB \ CD	00	01	11	10
00	1 (0)	1 (1)	(3)	(2)
01	(4)	(5)	(7)	(6)
11	X (12)	X (13)	X (15)	X (14)
10	(8)	(9)	X (11)	X (10)

$f_1 = \bar{A}\bar{B}$

K-map for f_2 —:

AB \ CD	00	01	11	10
00	(0)	(1)	1 (3)	1 (2)
01	1 (4)	1 (5)	(7)	(6)
11 ($B\bar{C}$)	X (12)	X (13)	X (15)	X (14)
10	(8)	(9)	X (11)	X (10)

$f_2 = B\bar{C} + \bar{B}C$

K-map for f_3 —:

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

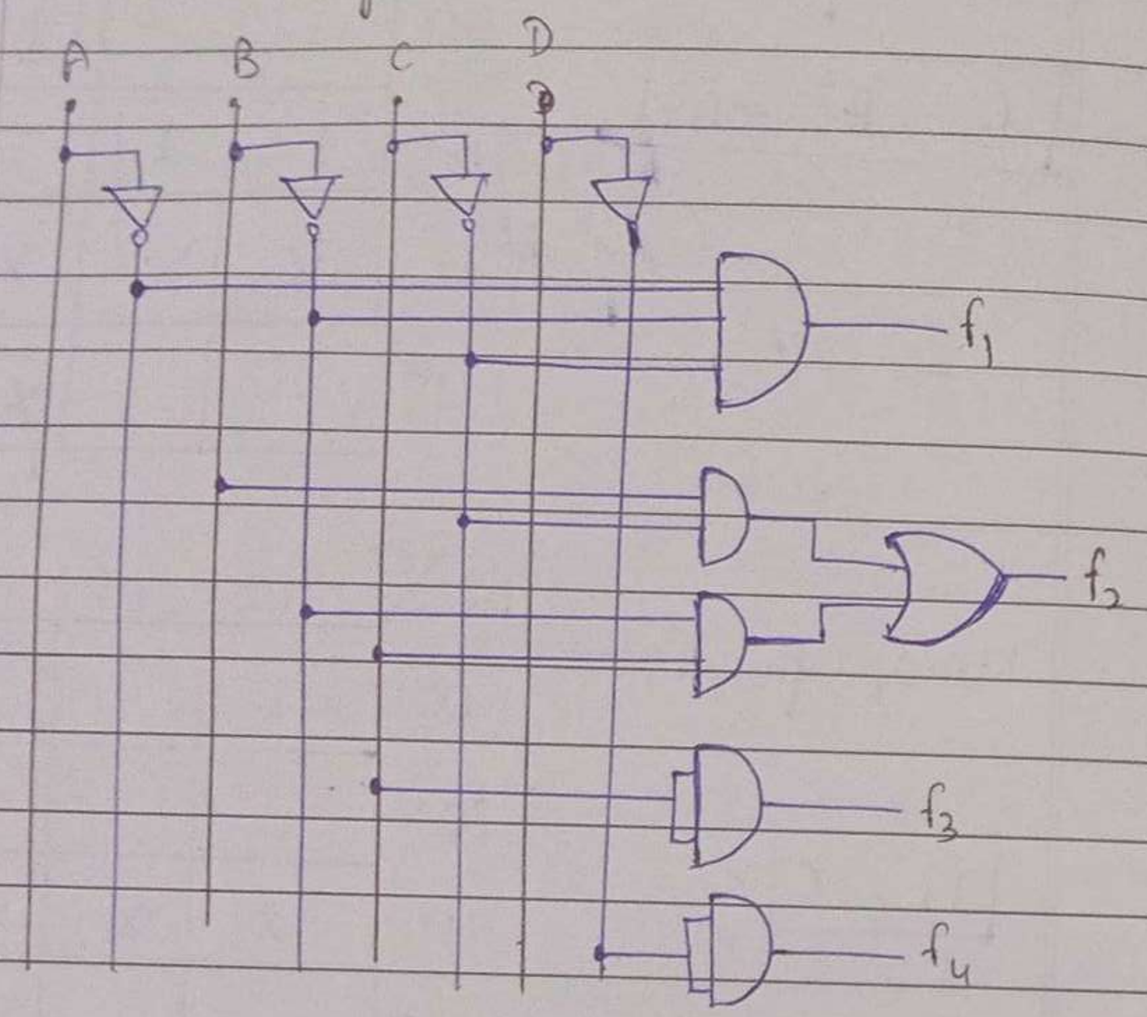
$f_3 = C$

K-map for f_4 :—

AB \ CD	00	01	11	10
00	1			1
01	1			1
11 ($\bar{D}$)	X	X	X	X
10	1	X	X	X

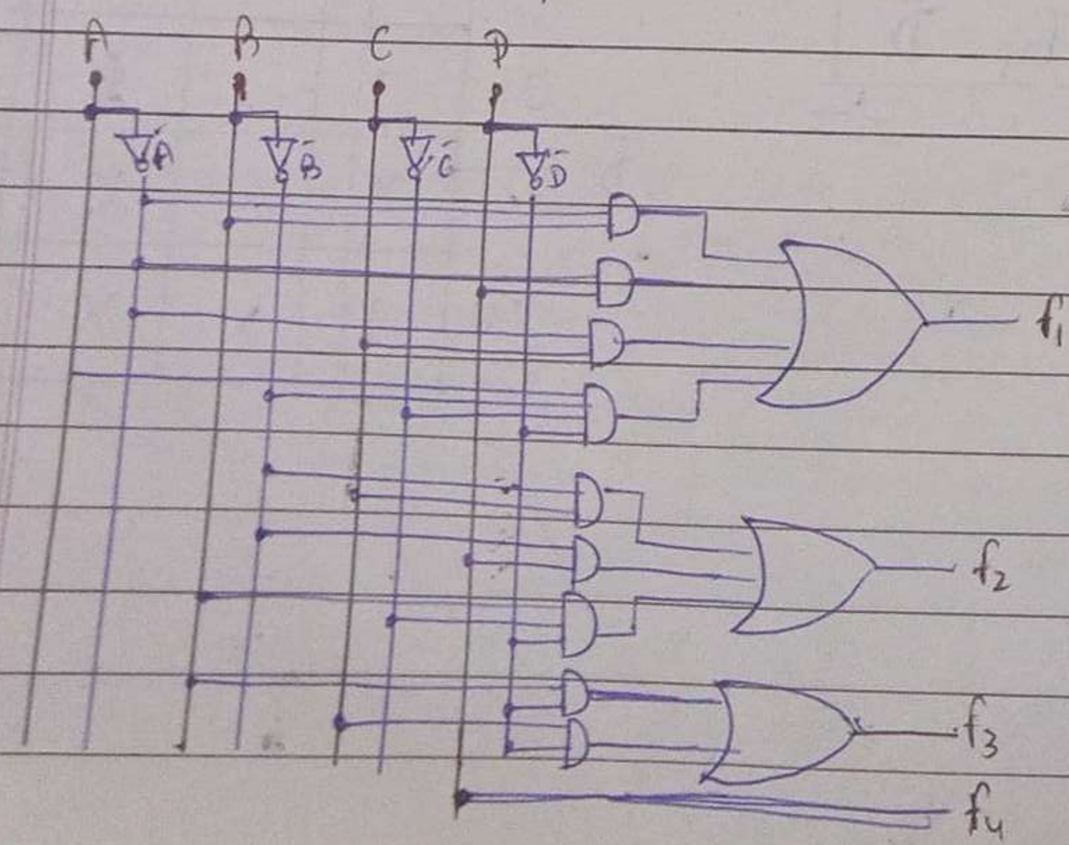
$f_4 = \bar{D}$

Circuit Design —



④ Design a circuit of four Bit I/P which gives 2's combinate of I/P.

Sol: Step②: Circuit diagram —



Step①:

Binary I/P				2's comb. O/P			
A	B	C	D	f_1	f_2	f_3	f_4
0	0	0	0	0	0	0	0
0	0	0	1	1	1	1	1
0	0	1	0	1	1	1	0
0	0	1	1	1	1	0	1
0	1	0	0	1	1	0	0
0	1	0	1	1	0	1	1
0	1	1	0	1	0	1	0
0	1	1	1	1	0	0	1
1	0	0	0	1	0	0	0
1	0	0	1	0	1	1	1
1	0	1	0	0	1	1	0
1	0	1	1	0	1	0	1
1	1	0	0	0	1	0	0
1	1	0	1	0	0	1	1
1	1	1	0	0	0	1	0
1	1	1	1	0	0	0	1

Step②: K-map for f_1

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

$$f_1 = \bar{A}B + \bar{A}D + \bar{A}C + A\bar{B}\bar{C}\bar{D}$$

27/10/21

K-map for f_2 —

$$\boxed{f_2 = \bar{B}D + \bar{B}C + B\bar{C}\bar{D}}$$

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

K-map for f_3 —

$$\boxed{f_3 = \bar{C}D + C\bar{D}}$$

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

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

$\overline{(A\bar{B})} \rightarrow \bar{A} + B$

$\boxed{Y = \overline{AB}} \rightarrow (\bar{A} + \bar{B})$

NAND G

K-map for f_4 —

$$\boxed{f_4 = D}$$

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

$A\bar{A} = 0$

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

Adder

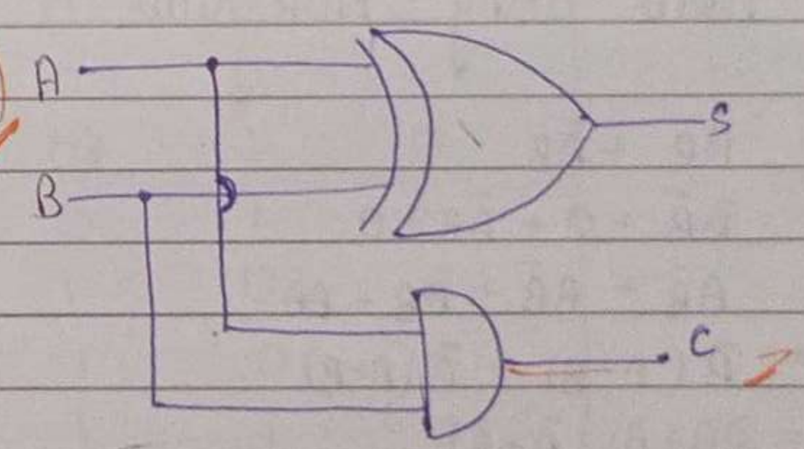
Adder — Half Adder (2 Bit)
— Full Adder (3 Bit)

Half Adder —

A	B	Sum(S)	Carry(C)
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

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

$\boxed{S = A \oplus B}$ $\boxed{C = AB}$



Half Adder using NAND Gate —

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

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

$A(\overline{AB}) + B(\overline{AB})$

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

$= A\bar{B} + 0 + \bar{A}B + 0$

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

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

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

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

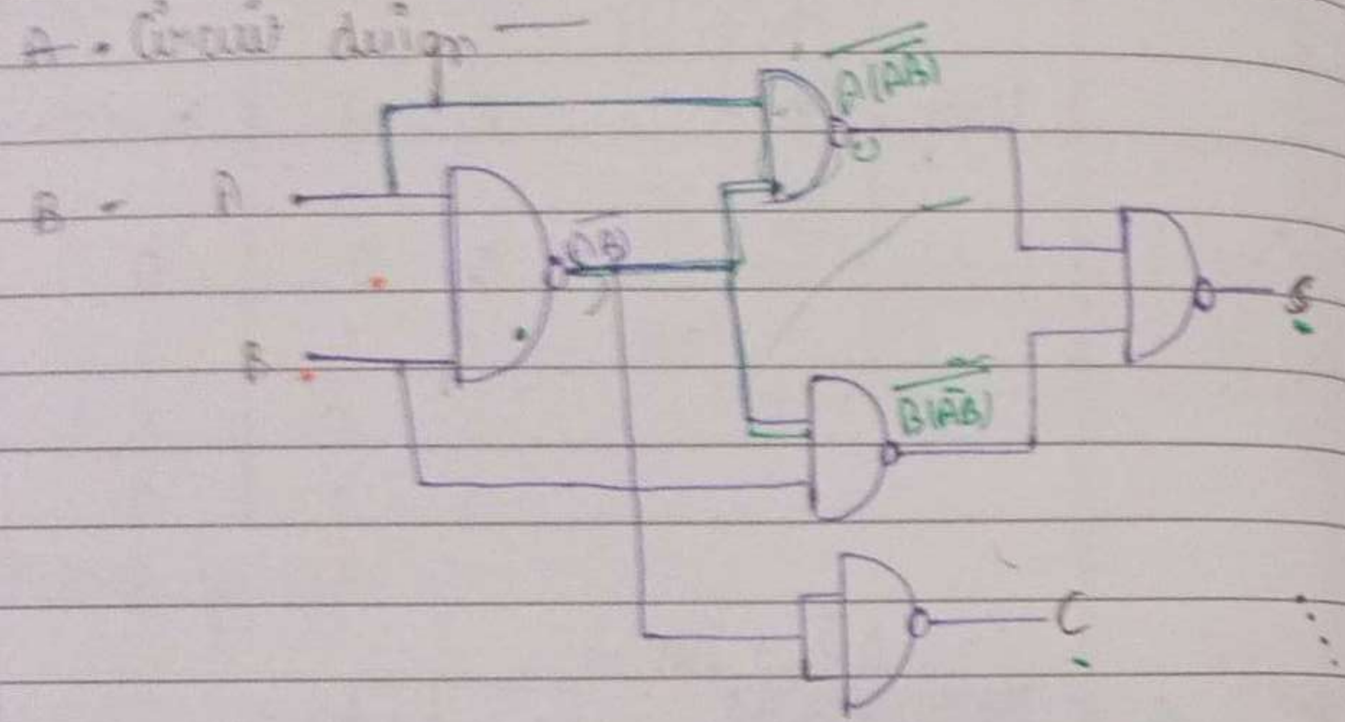
$S = \overline{\overline{B(\overline{AB})} + \overline{A(\overline{AB})}}$

$S = \overline{\overline{B(\overline{AB})} \cdot \overline{A(\overline{AB})}}$

$C = \overline{\overline{AB}}$

$C = \overline{\overline{AB}}$

A. Circuit design —



Half Adder using NOR Gate —

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

$S = A\bar{B} + 0 + \bar{A}B + 0$

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

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

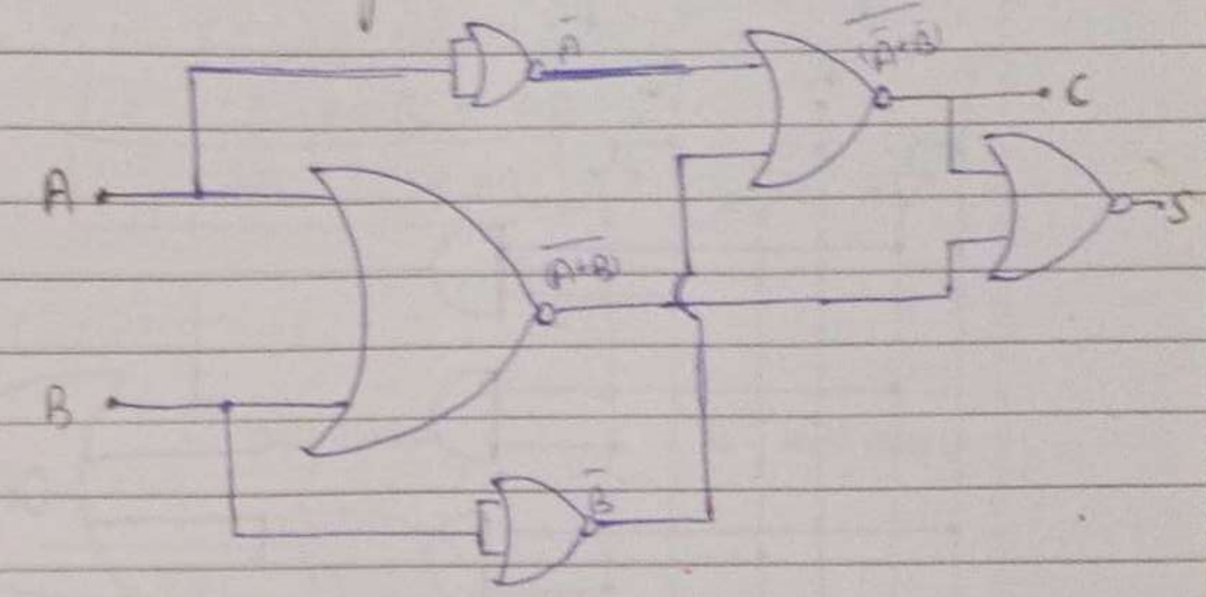
$S = \overline{\overline{(A+B)(\bar{A}+\bar{B})}}$

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

$C = \overline{\overline{AB}}$

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

Circuit design —



Full Adder —

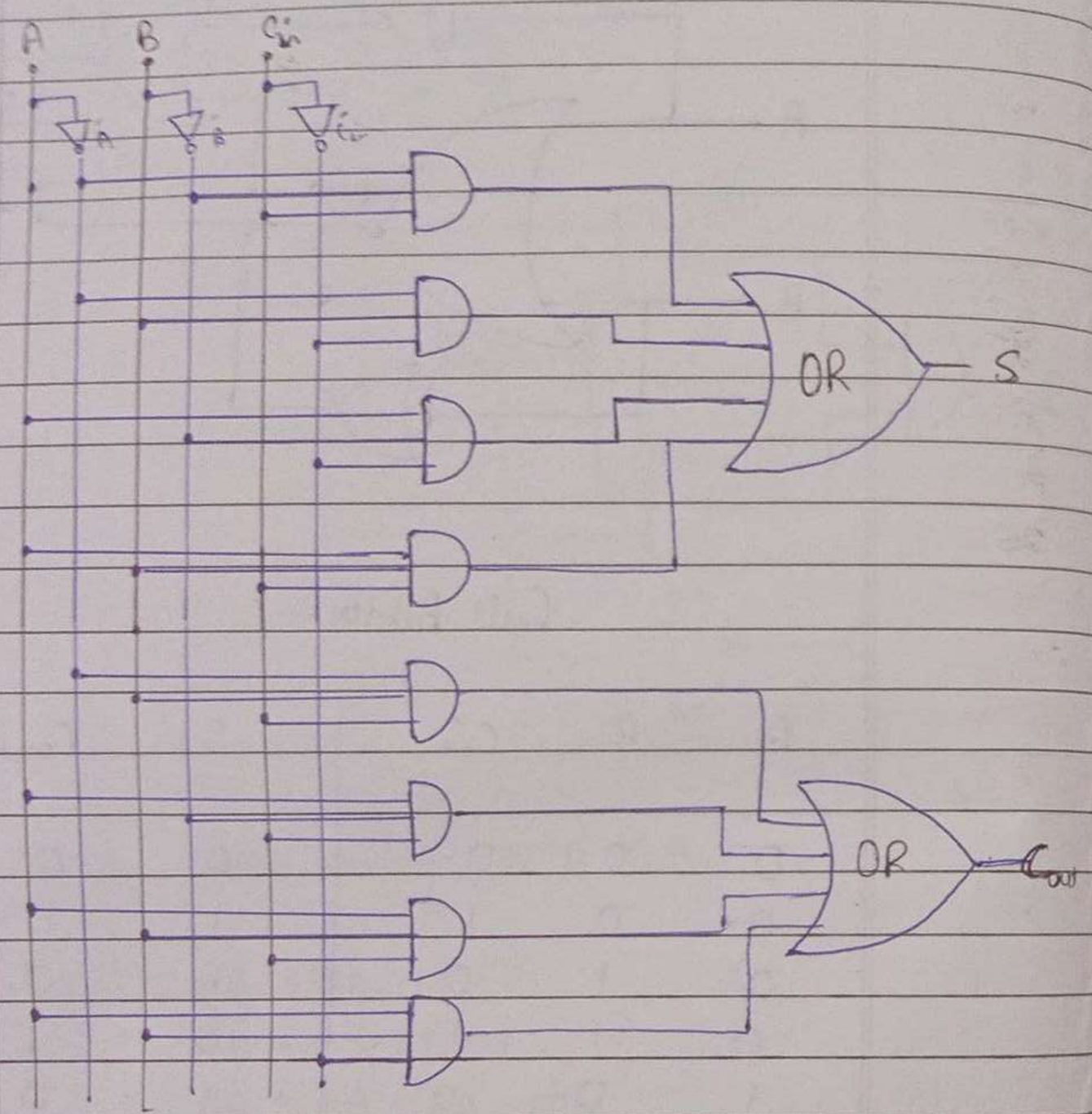
A	B	C_{in}	S	C_{out}
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

$S = \bar{A}\bar{B}C_{in} + \bar{A}B\bar{C}_{in} + A\bar{B}\bar{C}_{in} + ABC_{in}$

$C_{out} = \bar{A}BC_{in} + A\bar{B}\bar{C}_{in} + AB\bar{C}_{in} + ABC_{in}$

Date
Page

Using Basic gates —



Full adder using Two Half Adder —

$S = \bar{A}\bar{B}C_{in} + \bar{A}B\bar{C}_{in} + A\bar{B}\bar{C}_{in} + ABC_{in}$

$S = C_{in}(\bar{A}\bar{B} + AB) + \bar{C}_{in}(\bar{A}B + A\bar{B})$

$S = C_{in}(A \odot B) + \bar{C}_{in}(A \oplus B)$

$S = C_{in}\,\overline{(A \oplus B)} + \bar{C}_{in}(A \oplus B)$

let $A \oplus B = Y$

$\therefore S = \bar{Y}C_{in} + Y\bar{C}_{in}$

$S = (Y \oplus C_{in})$ $\boxed{S = A \oplus B \oplus C_{in}}$

Date
Page

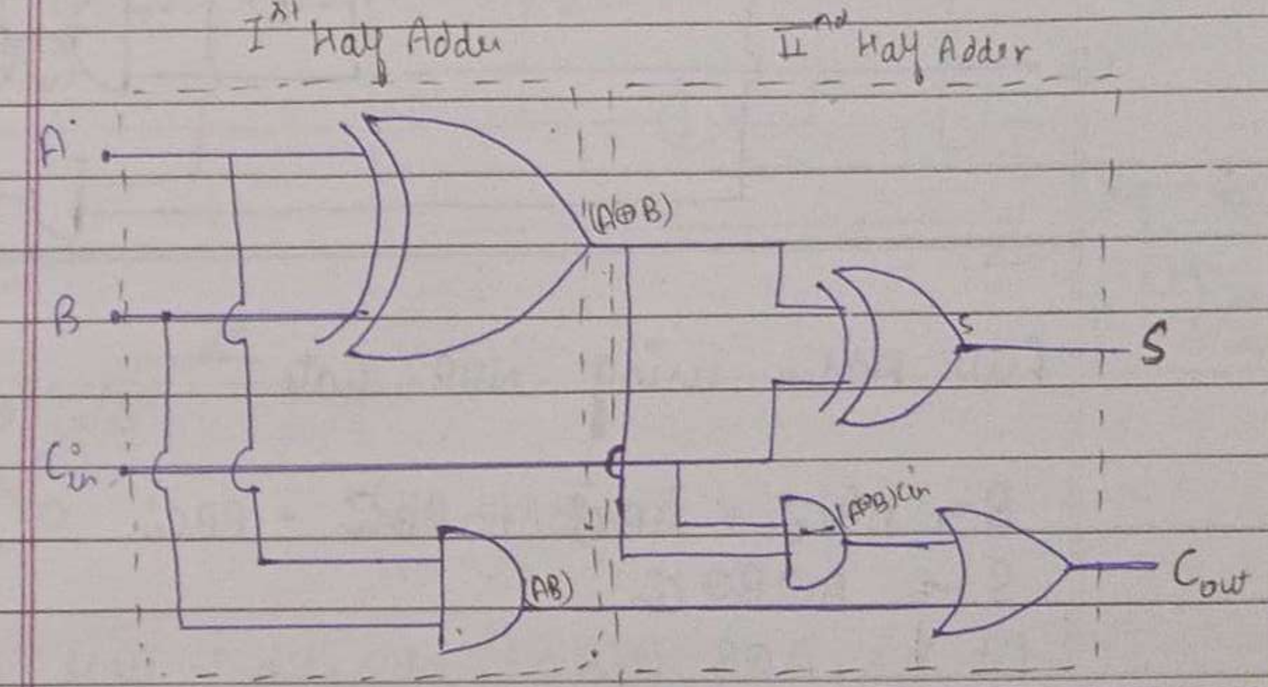
$C = \bar{A}BC_{in} + A\bar{B}C_{in} + AB\bar{C}_{in} + ABC_{in}$

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

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

$\boxed{C = AB + C_{in}(A \oplus B)}$

Circuit —



Full Adder using NAND Gate —

$S = \bar{A}\bar{B}C_{in} + \bar{A}B\bar{C}_{in} + A\bar{B}\bar{C}_{in} + ABC_{in}$

$S = A \oplus B \oplus C_{in}$ — ①

Expression of S using NAND —

$S =$ let $X = A \oplus B$

$X = \overline{A \cdot \overline{(A \cdot B)} \cdot B\overline{(AB)}}$

$S = X \oplus C_{in}$ (from ①)

$S = \overline{X\overline{(X \cdot C_{in})} \cdot C_{in}\overline{(X \cdot C_{in})}}$

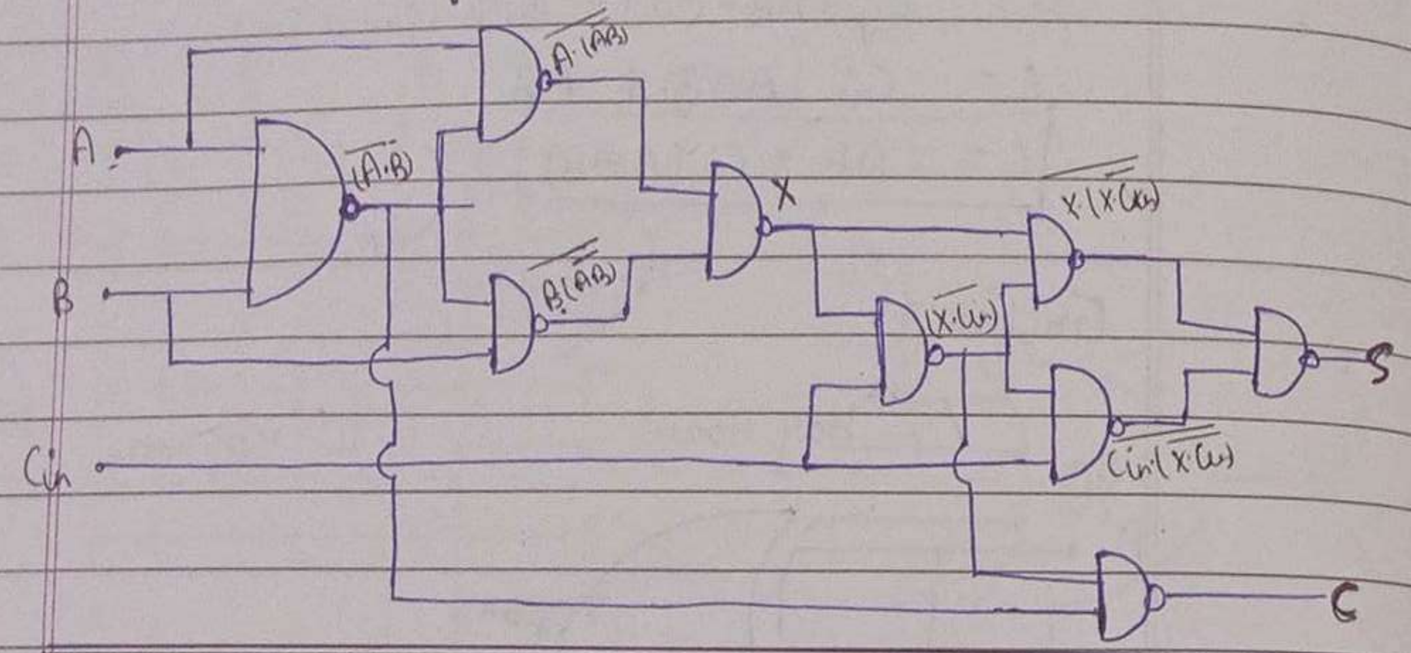
$C_{out} = \bar{A}BC_{in} + A\bar{B}C_{in} + AB\bar{C}_{in} + ABC_{in}$

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

$C_{out} = \overline{\overline{AB} + C_{in}X}$ $C_{out} = \overline{\overline{AB} \cdot \overline{(C_{in}X)}}$

Date
Page

Circuit Design —



Full Adder using NOR Gate —

$S = \bar{A}\bar{B}C_{in} + \bar{A}B\bar{C}_{in} + A\bar{B}\bar{C}_{in} + ABC_{in}$

$S = A \oplus B \oplus C_{in}$

Let $X = A \oplus B$

& $X = \overline{(\overline{A+B}) + (\overline{\bar{A}+\bar{B}})}$

$S = X \oplus C_{in}$

$S = \overline{(\overline{X + C_{in}}) + (\overline{\bar{X} + \bar{C}_{in}})}$

$C_{out} = \bar{A}BC_{in} + A\bar{B}C_{in} + AB\bar{C}_{in} + ABC_{in}$

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

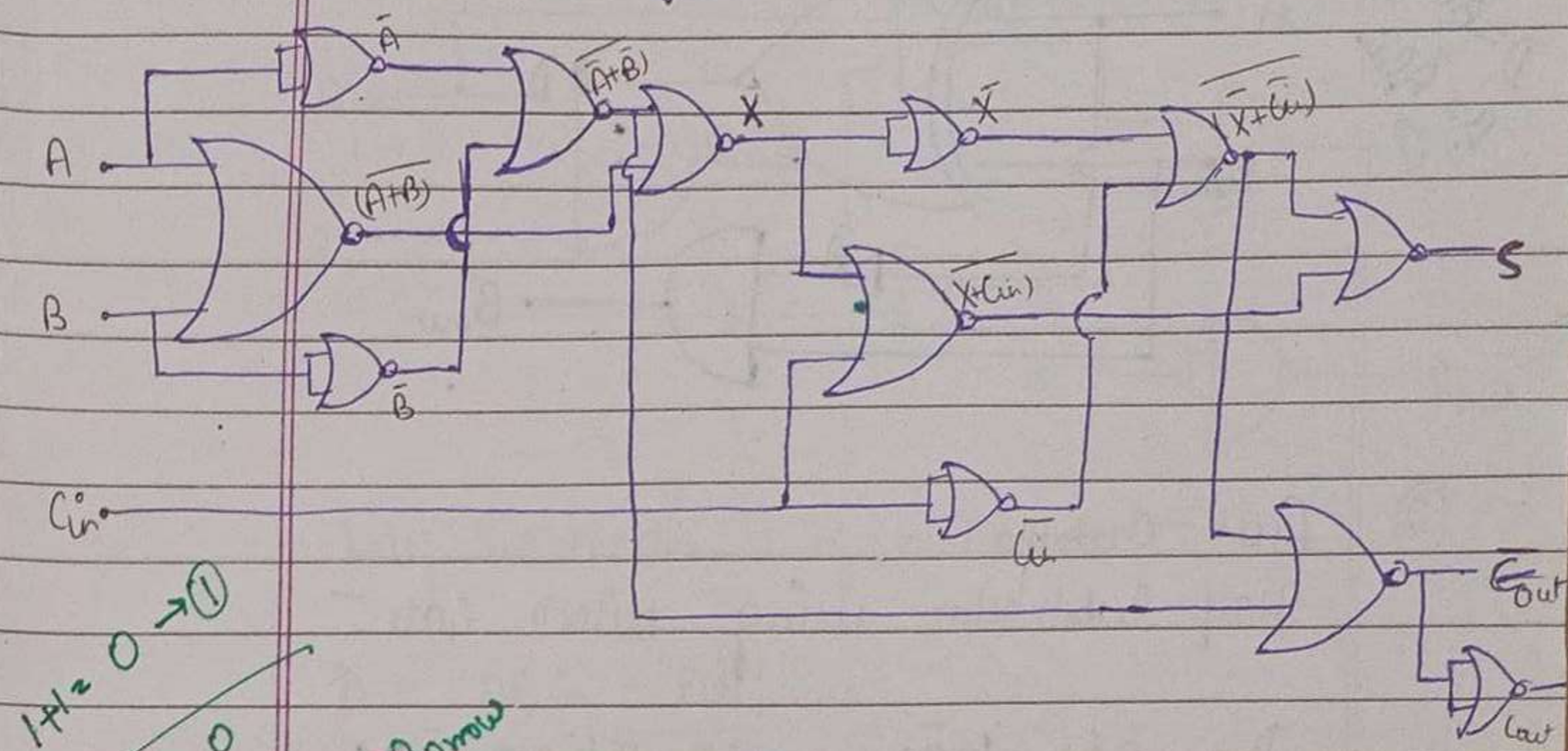
$C_{out} = AB + C_{in}X$

$C_{out} = \overline{\bar{A}\bar{B}} + \overline{\bar{C}_{in}\bar{X}}$

$C_{out} = \overline{(\bar{A}+\bar{B})} + \overline{(\bar{C}_{in}+\bar{X})}$

Date
Page

Circuit Design —



$1+1 = 0 \rightarrow$ ①

$0-0=0$

$0-1=1$ with B=1

$1-0=1$

$1-1=0$

$0-0=0$

$1-0=1$

$0-1=1$

$1-1=0$

① Borrow

Substractor

① Half Substractor (2 Bit)

② Full Substractor (3 Bit)

Half Substractor —

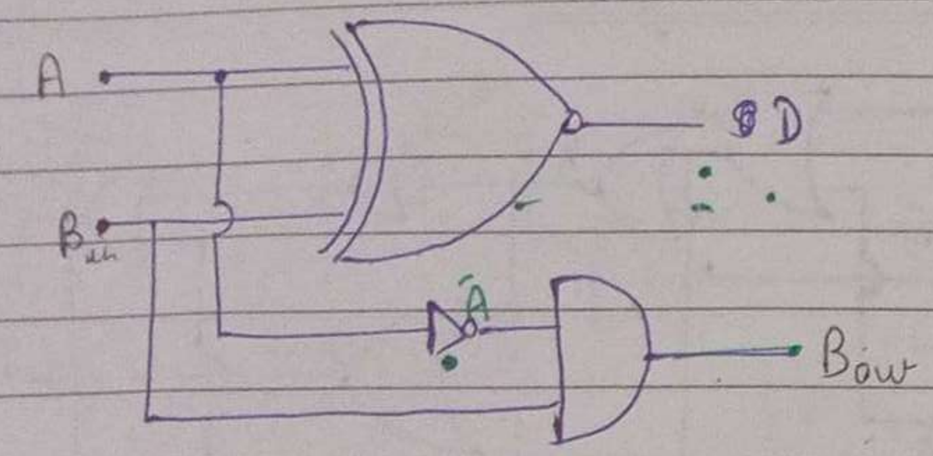
A	B_{in}	Difference (D)	Borrow (B_{out})
0	0	0	0
0	1	1	1
1	0	1	0
1	1	0	0

$D = \bar{A}B_{in} + A\bar{B}_{in} = A \oplus B_{in}$

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

$D = \bar{A}B_{in} + A\bar{B}_{in} = A \oplus B_{in}$

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



② ~~Full Subtra~~

Half Subtractor using NAND Gate –

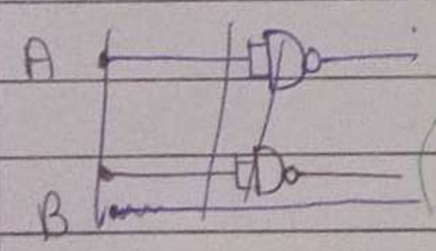
$D = \overline{\bar{A}B_{in} + A\bar{B}_{in}}$

$D = \overline{\bar{A}B_{in} + A\bar{B}_{in}}$

$D = \overline{(\bar{A}B_{in}) \cdot (A\bar{B}_{in})}$

$B_{ow} = \overline{\bar{A}B_{in}}$

$B_{ow} = \overline{\bar{A}B_{in}}$



$D = \bar{A}B_{in} + A\bar{B}_{in}$

$= \bar{A}B_{in} + 0 + A\bar{B}_{in} + 0$

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

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

$= \overline{(A + B_{in})}\,\overline{(\bar{A} + \bar{B}_{in})}$

$= (A + B_{in})\,\overline{(A \cdot B_{in})}$

$D = (A + B_{in}) \cdot (A \cdot B$

$D = \bar{A}(\bar{A} + \bar{B}_{in}) + \bar{B}_{in}(\bar{A} + \bar{B}_{in})$

$= \bar{A}\,\overline{(A \cdot B_{in})} + \bar{B}_{in}\,\overline{(A \cdot B_{in})}$

$D = \overline{\bar{A}\,\overline{(A \cdot B_{in})} \cdot \bar{B}_{in}\,\overline{(A \cdot B_{in})}}$

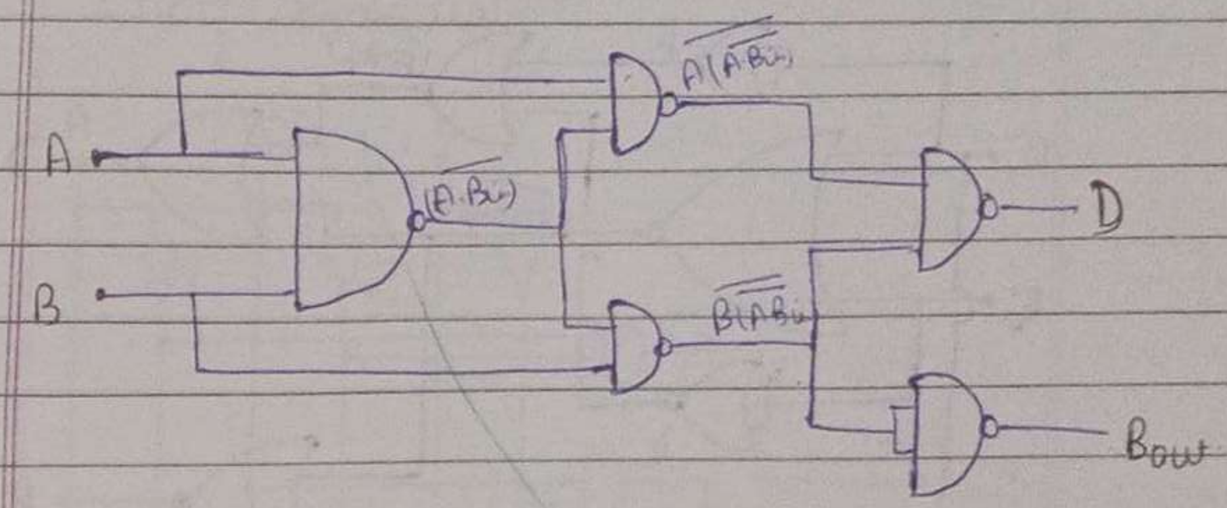
$B_{ow} = \bar{A}B_{in}$

$= \bar{A}B_{in} + 0 = \bar{A}B_{in} + B_{in}\bar{B}_{in} = B_{in}(\bar{A} + \bar{B}_{in})$

$D = B_{in}\,\overline{(A \cdot B_{in})}$

$\bar{A}B_{in} + A\bar{A}$

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



Half Subtractor using NOR Gate –

$D = \bar{A}B_{in} + A\bar{B}_{in}$

$D = \bar{A}B_{in} + A\bar{A} + A\bar{B}_{in} + B\bar{B}_{in}$

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

$= \overline{(\bar{A} + \bar{B}_{in}) + (A + B_{in})}$

$D = \overline{(\bar{A} + \bar{B}_{in}) + \overline{(A + B_{in})}}$

$B_{ow} = \bar{A}B_{in}$ $\quad \bar{A}B_{in} + 0 = \cancel{\bar{A}B_{in} + B_{in}B}$

$= \overline{\bar{A}B_{in}} = \overline{A + \bar{B}_{in}}$

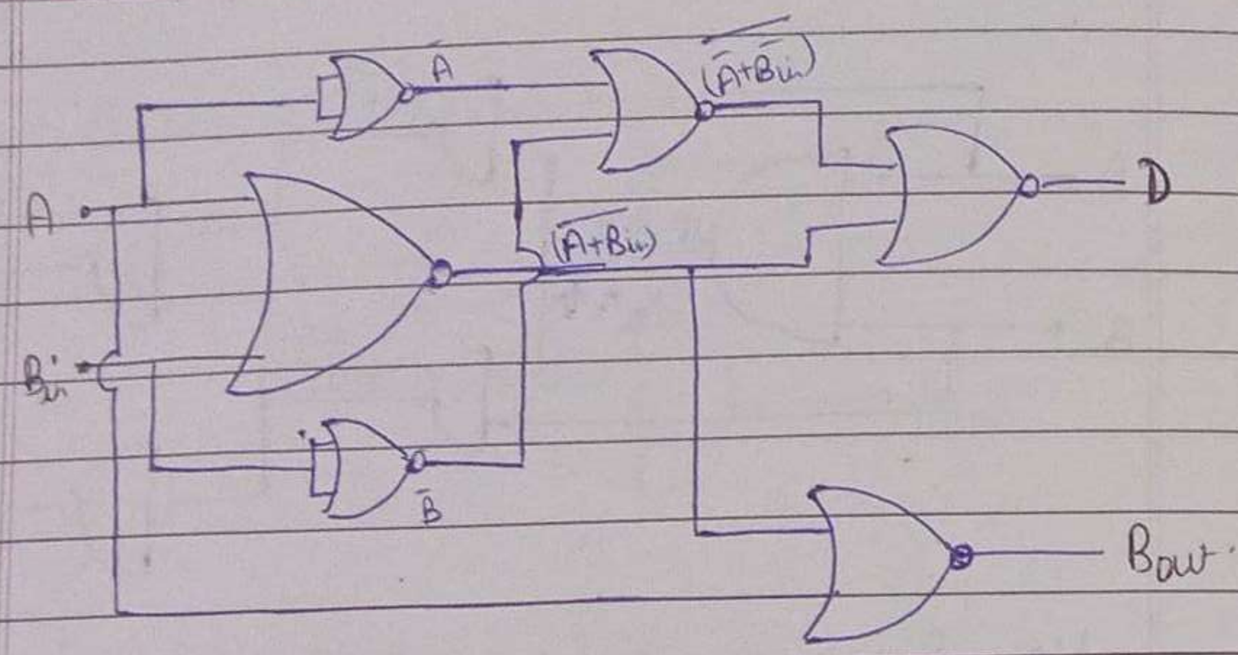
$\bar{A}B_{in} + 0 = \bar{A}B_{in} + B\bar{B}_{in}$

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

$B_{ow} = \bar{A}B_{in} + 0$

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

$= \bar{A}(A + B_{in}) = \overline{A + \overline{(A + B_{in})}}$



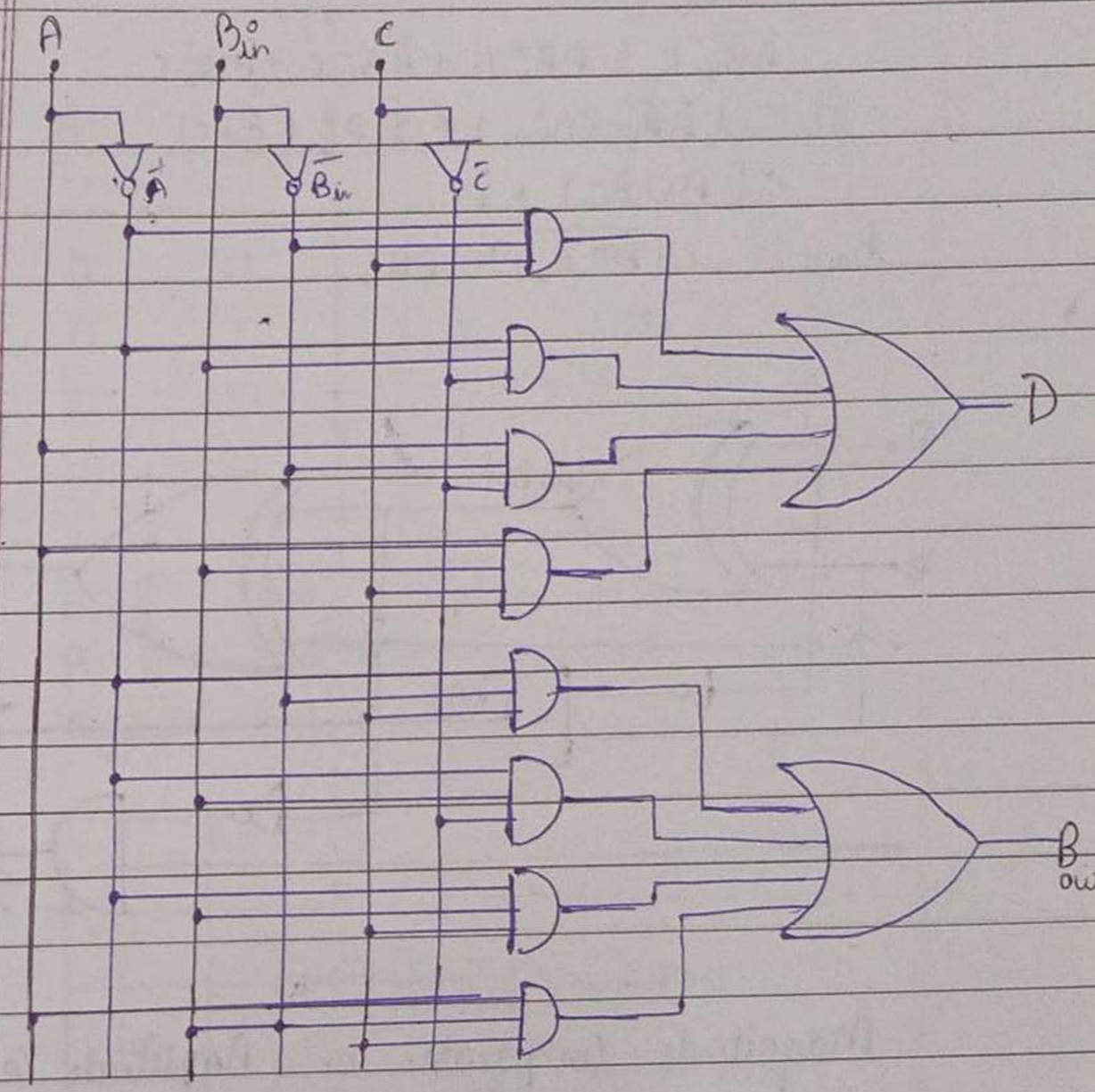
Full Substractor

A	B_{in}	C	D	B_{ow}
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

$D = \bar{A}\bar{B}_{in}C + \bar{A}B_{in}\bar{C} + A\bar{B}_{in}\bar{C} + AB_{in}C$

$B_{ow} = \bar{A}\bar{B}_{in}C + \bar{A}B_{in}\bar{C} + \bar{A}B_{in}C + AB_{in}C$

Using Basic Gate —



Full Subtractor using Two Half Subtractor —

$D = \bar{A}\bar{B}_{in}C + \bar{A}B_{in}\bar{C} + A\bar{B}_{in}\bar{C} + AB_{in}C$

$D = \bar{A}(\bar{B}_{in}C + B_{in}\bar{C}) + A(\bar{B}_{in}\bar{C} + B_{in}C)$

$D = \bar{A}(B_{in} \oplus C) + A(\overline{B_{in} \oplus C})$

$D = \bar{A}(B_{in} \oplus C) + A(\overline{B_{in} \oplus C})$

Let $B_{in} \oplus C = X$

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

$D = A \oplus X$

$D = A \oplus B_{in} \oplus C$

$B_{out} = \bar{A}\bar{B_i}C + \bar{A}B_i\bar{C} + \bar{A}B_iC + \overline{AB_i}C$

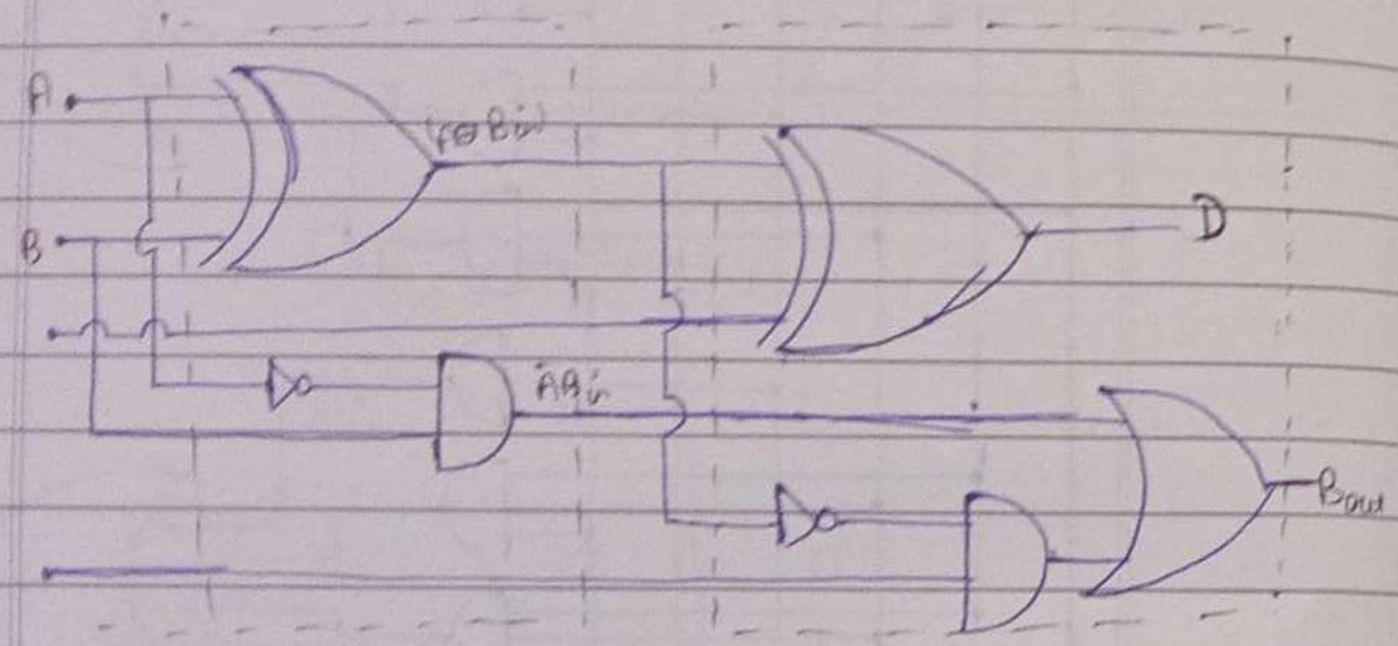
$= \cancel{\bar{A}\bar{B_i} + C +}$

$\bar{A}\bar{B_i}C + \bar{A}B_iC + \bar{A}B_i\bar{C} + \bar{A}B_iC$

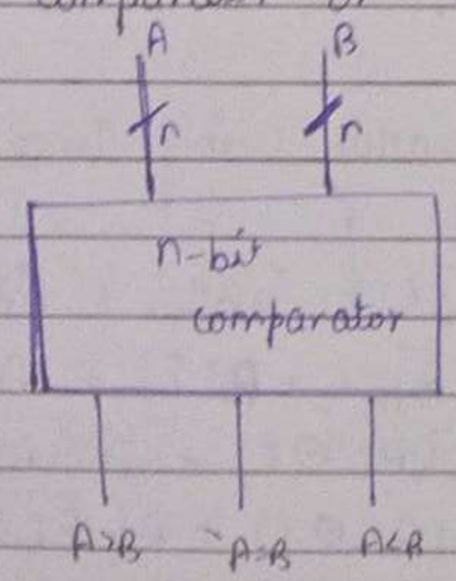
$= C(\bar{A}\bar{B_i} + \bar{A}B_i) + \bar{A}B_i(\bar{C} + C)$

$C(A \odot B_i) + \bar{A}B_i$

$B_{out} = C(\overline{A \odot B_i}) + \bar{A}B_i$



Magnitude Comparator or Amplitude Comparator –



A magnitude comp is a combinational logic circuit which is used to compare the magnitude of input lines and give the corresponding O/P.

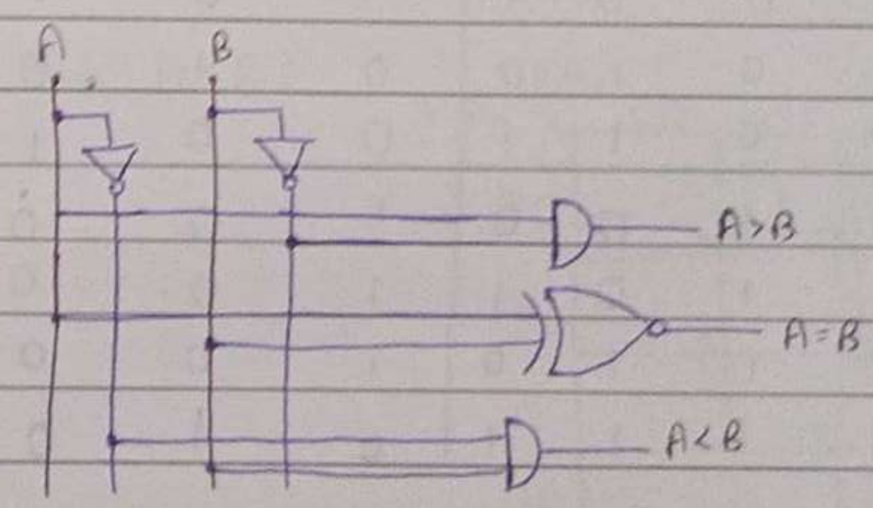
① 1-Bit Comparator –

I/P		O/P		
A	B	A>B	A=B	A<B
0	0	0	1	0
0	1	0	0	1
1	0	1	0	0
1	1	0	1	0

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

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

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

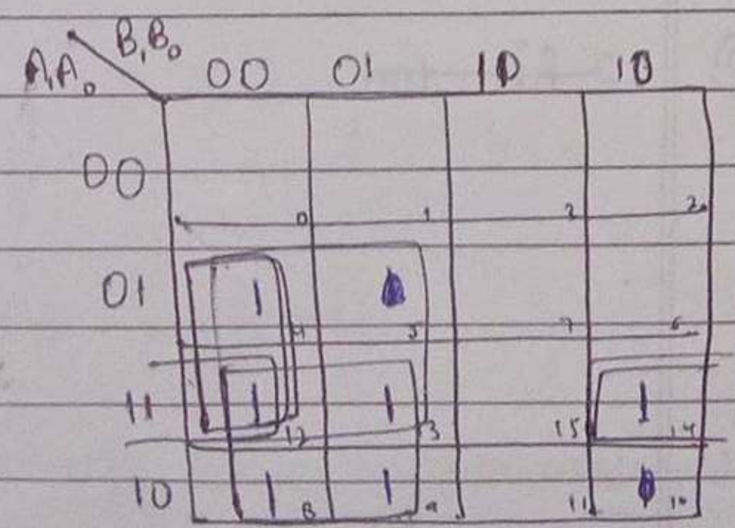


② ~~2-Bit Com~~

② 2-Bit Comparator —

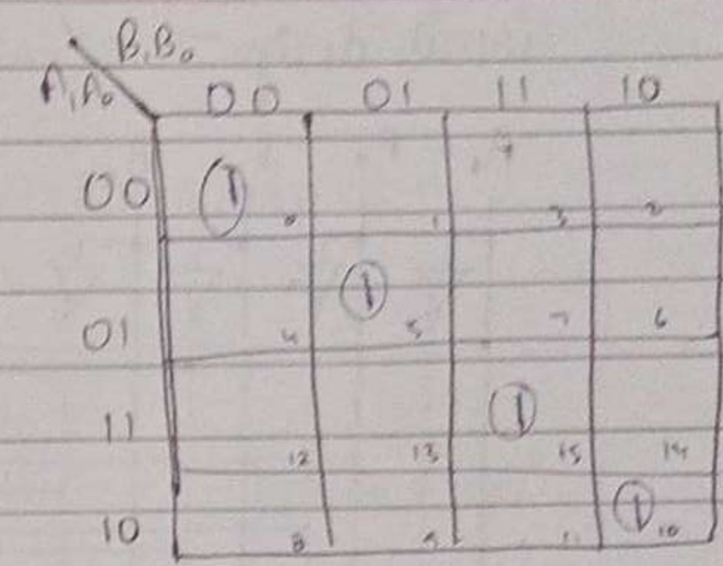
I/P				O/P		
A_1	A_0	B_1	B_0	$A>B$	$A=B$	$A<B$
0	0	0	0	0	1	0
0	0	0	1	0	0	1
0	0	1	0	0	0	1
0	0	1	1	0	0	1
0	1	0	0	1	0	0
0	1	0	1	0	1	0
0	1	1	0	0	0	1
0	1	1	1	0	0	1
1	0	0	0	1	0	0
1	0	0	1	1	0	0
1	0	1	0	0	1	0
1	0	1	1	0	0	1
1	1	0	0	1	0	0
1	1	0	1	1	0	0
1	1	1	0	1	0	0
1	1	1	1	0	1	0

for $A>B$:



$A>B = A_1\bar{B}_1 + A_0\bar{B}_1\bar{B}_0 + A_1A_0\bar{B}_0$

for $A=B$:



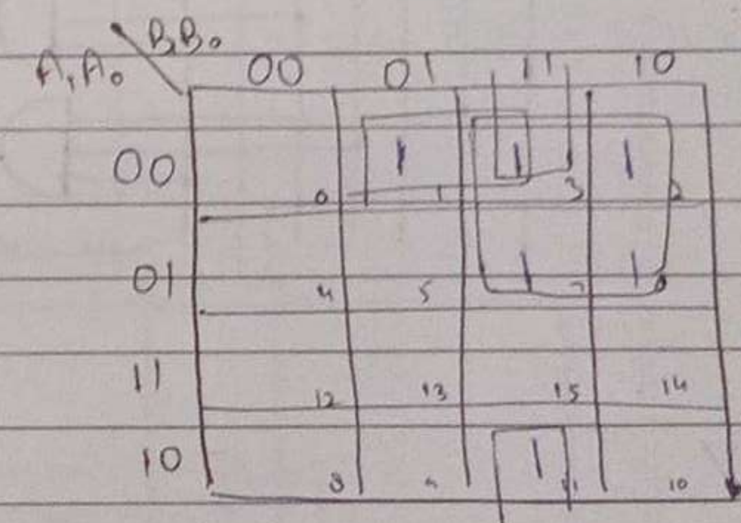
$A=B \Rightarrow \bar{A}_1\bar{A}_0\bar{B}_1\bar{B}_0 + \bar{A}_1A_0\bar{B}_1B_0 + A_1A_0B_1B_0 + A_1\bar{A}_0B_1\bar{B}_0$

$A=B \Rightarrow \bar{A}_0\bar{B}_0(\bar{A}_1\bar{B}_1 + A_1B_1) + A_0B_0(\bar{A}_1\bar{B}_1 + A_1B_1)$

$= (\bar{A}_1\bar{B}_1 + A_1B_1)(\bar{A}_0\bar{B}_0 + A_0B_0)$

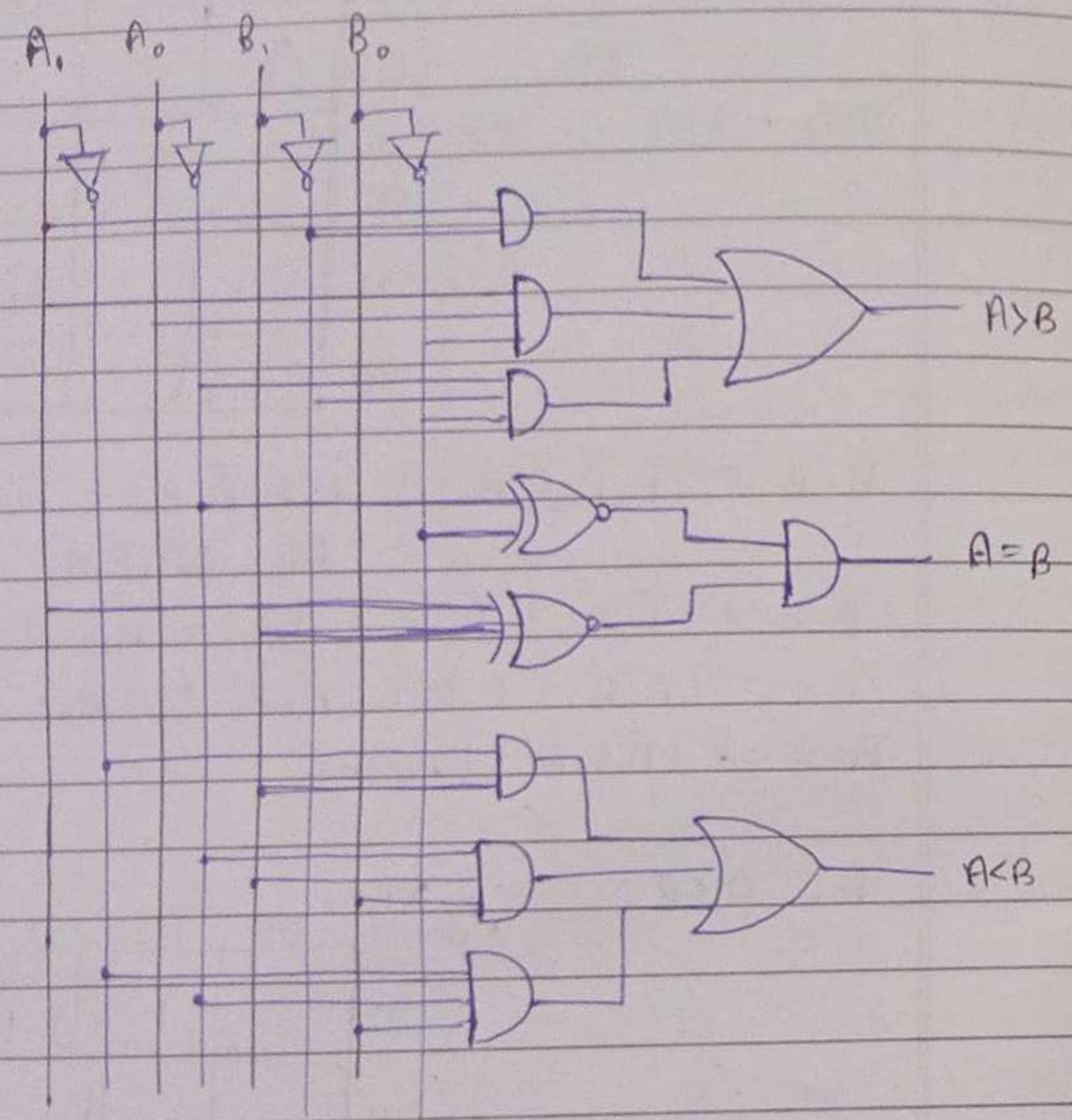
$A=B \Rightarrow (A_1 \odot B_1)(A_0 \odot B_0)$.

for $A<B$:



$A<B = \bar{A}_1B_1 + \bar{A}_0B_1B_0 + \bar{A}_1\bar{A}_0B_0$

Circuit design —



③ 4-Bit Comparator —

Consider two 4-Bit numbers.

$A = A_3A_2A_1A_0$

$B = B_3B_2B_1B_0$

Equivalent function —

$x_i = (A_iB_i + \bar{A}_i\bar{B}_i)' = (\bar{A}_iB_i + A_i\bar{B}_i)'$

$i = 0, 1, 2, 3$

$x_3 = (\bar{A}_3B_3 + A_3\bar{B}_3)' = (A\ (a+b)'$

$x_2 = (\bar{A}_2B_2 + A_2\bar{B}_2)'$

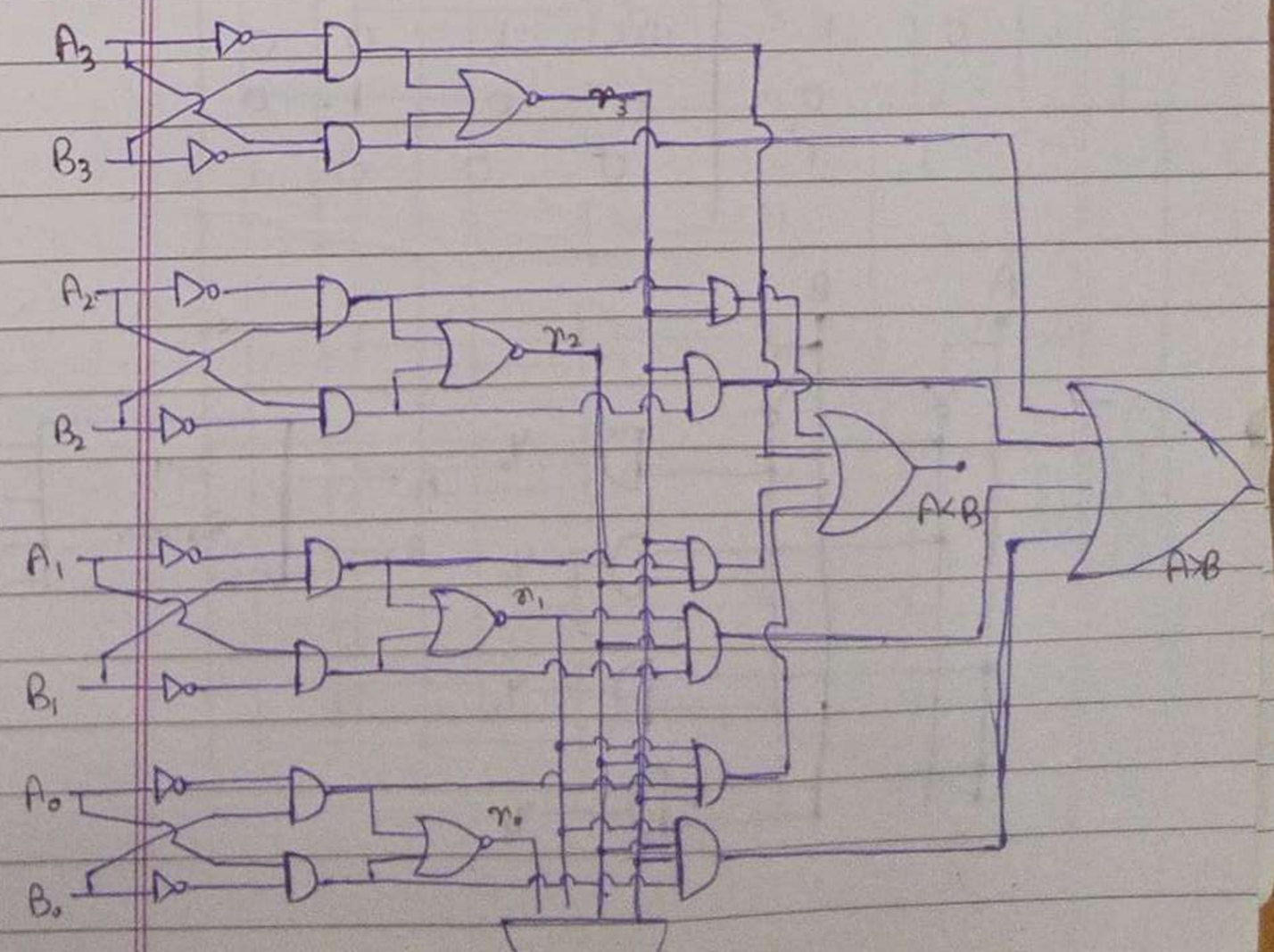
$x_1 = (\bar{A}_1B_1 + A_1\bar{B}_1)'$

$x_0 = (\bar{A}_0B_0 + A_0\bar{B}_0)'$

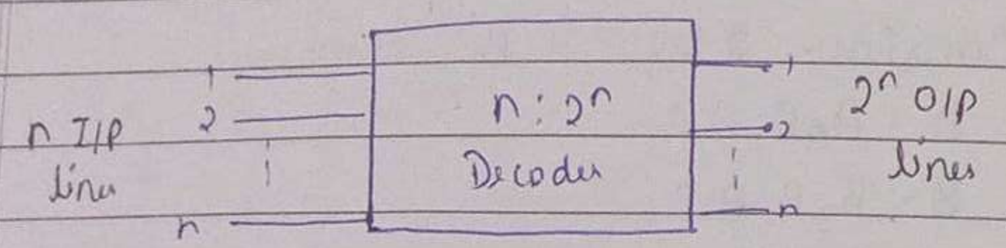
$A=B \Rightarrow \cancel{A_3B_3 +}\ x_3x_2x_1x_0$

$A>B \Rightarrow A_3\bar{B}_3 + x_3A_2\bar{B}_2 + x_3x_2A_1\bar{B}_1 + x_3x_2x_1A_0\bar{B}_0$

$A<B \Rightarrow \bar{A}_3B_3 + x_3\bar{A}_2B_2 + x_3x_2\bar{A}_1B_1 + x_3x_2x_1\bar{A}_0B_0$



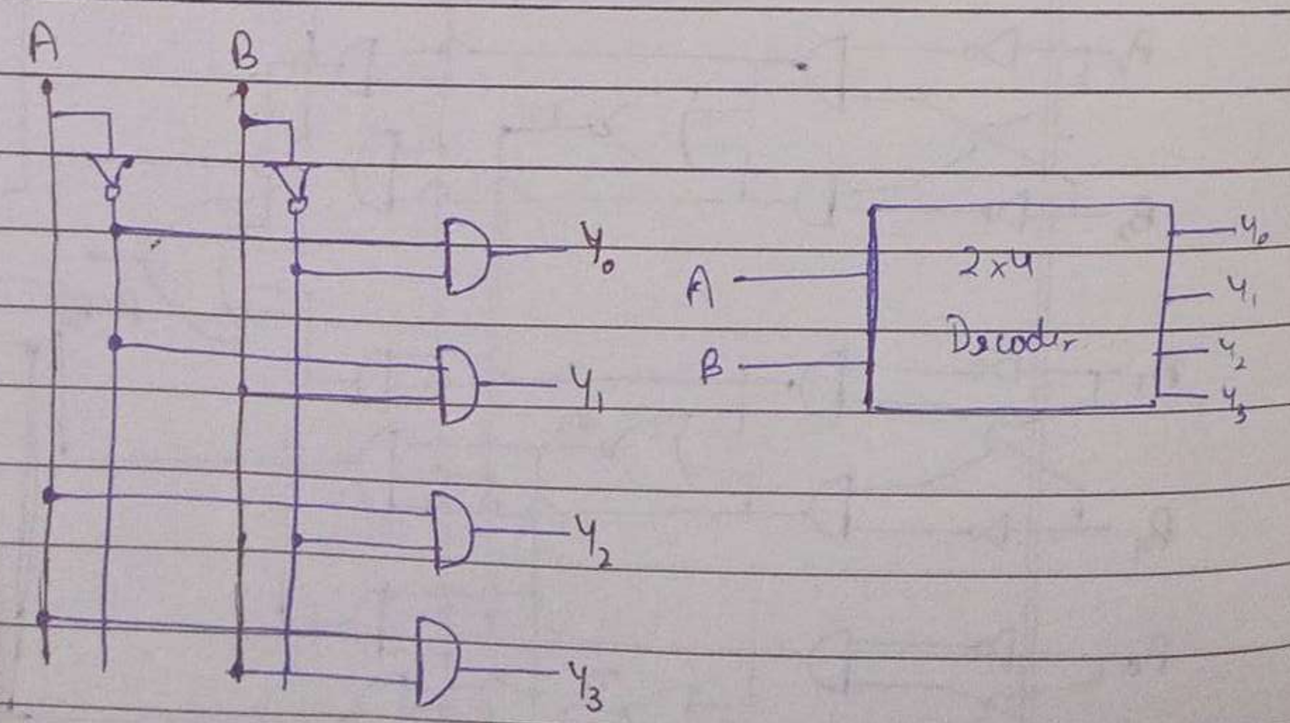
Decoder



A decoder is a combinational logic ckt that converts small 'n' binary I/P into 2^n O/P lines such that only one O/P line is activated for each one of the possible combinations of input.

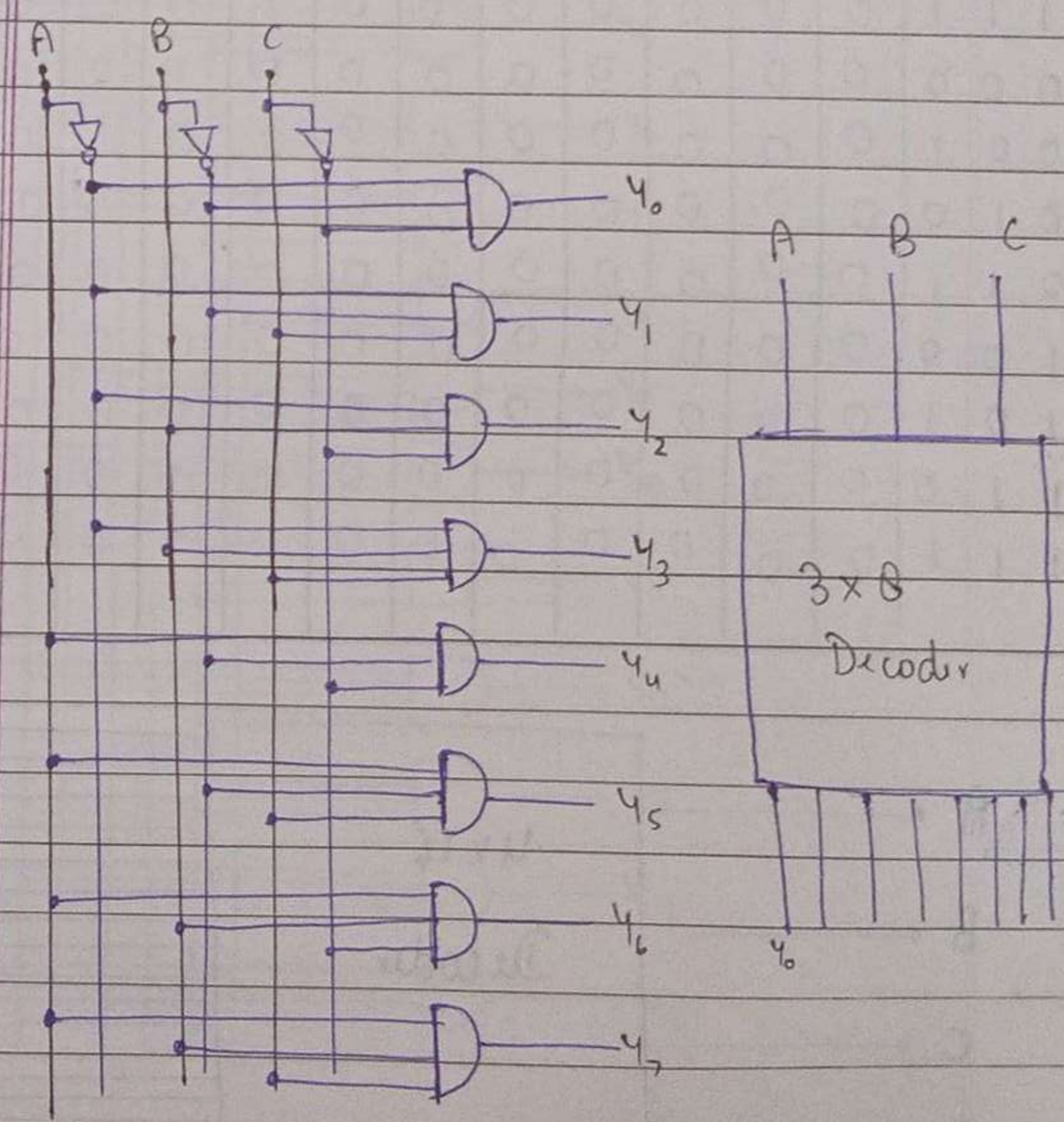
① 2x4 Decoder—

I/P		O/P			
A	B	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



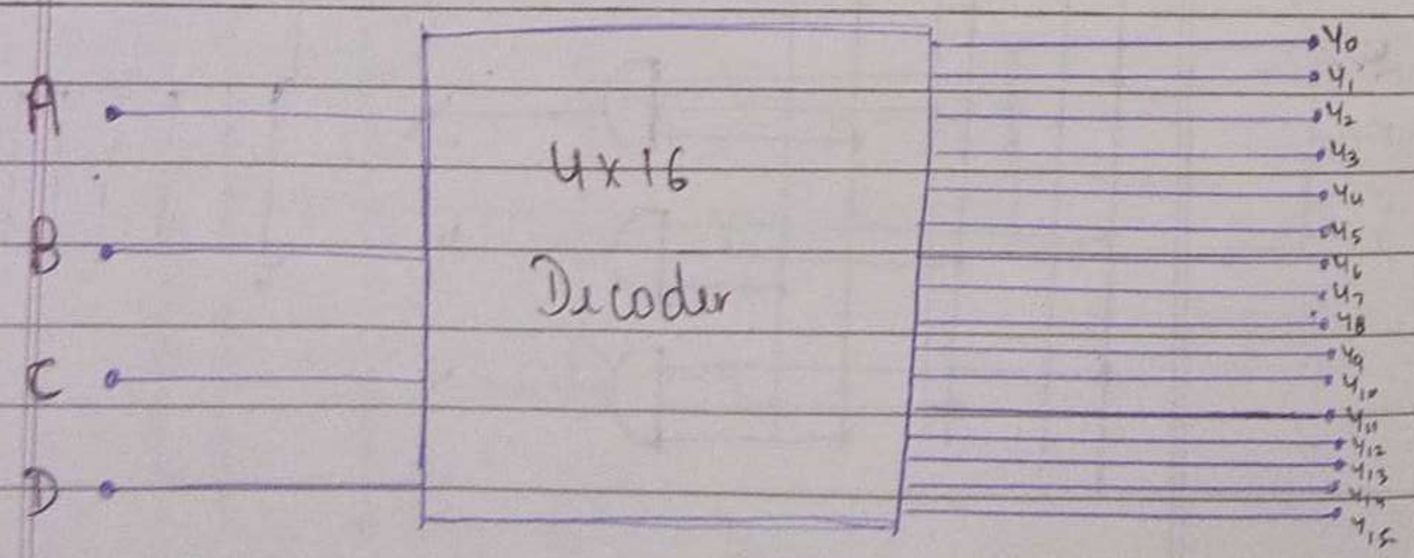
② 3x8 Decoder—

I/P			O/P							
A	B	C	Y_0	Y_1	Y_2	Y_3	Y_4	Y_5	Y_6	Y_7
0	0	0	1	0	0	0	0	0	0	0
0	0	1	0	1	0	0	0	0	0	0
0	1	0	0	0	1	0	0	0	0	0
0	1	1	0	0	0	1	0	0	0	0
1	0	0	0	0	0	0	1	0	0	0
1	0	1	0	0	0	0	0	1	0	0
1	1	0	0	0	0	0	0	0	1	0
1	1	1	0	0	0	0	0	0	0	1

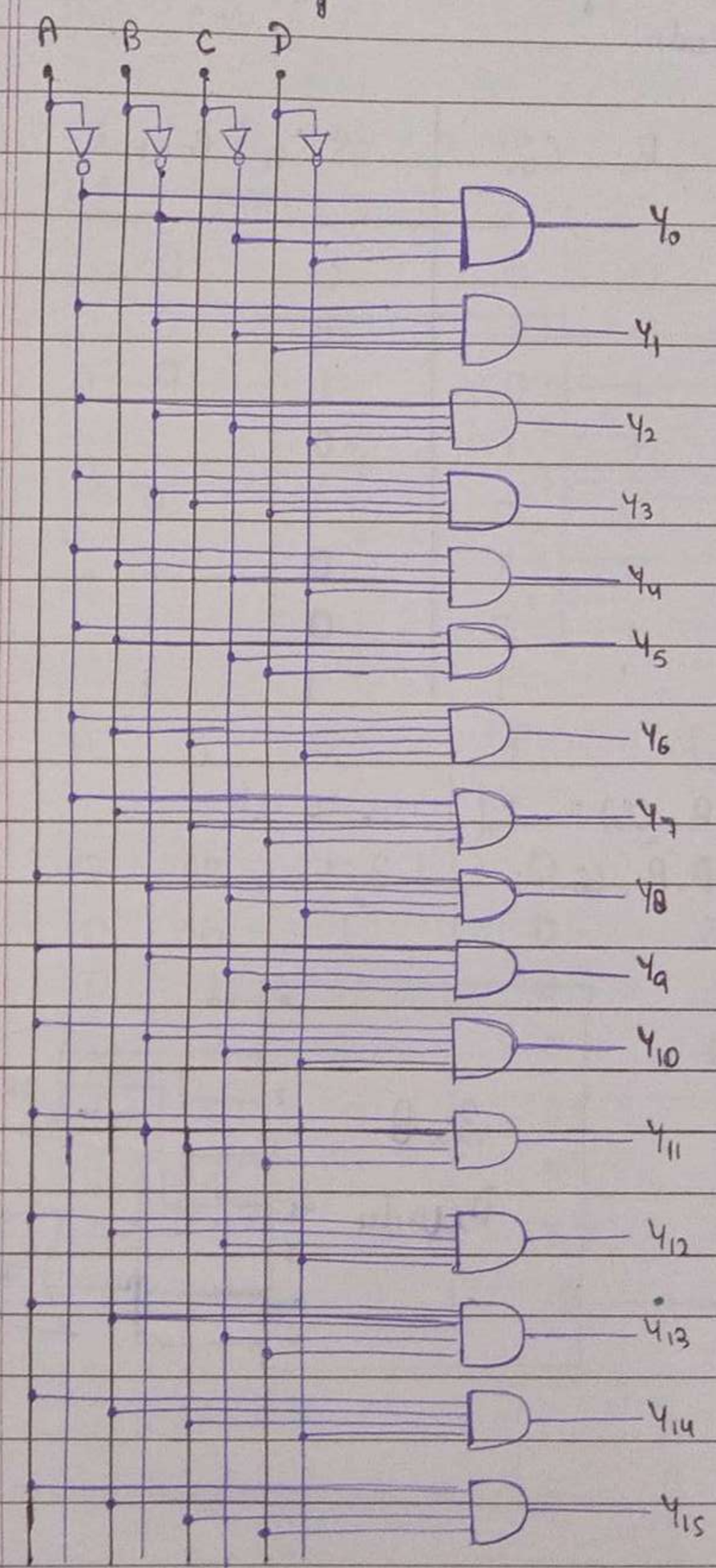


③ 4x16 Decoder —

A	B	C	D	Y_0	Y_1	Y_2	Y_3	Y_4	Y_5	Y_6	Y_7	Y_8	Y_9	Y_{10}	Y_{11}	Y_{12}	Y_{13}	Y_{14}	Y_{15}
0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	1	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	1	1	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0
0	1	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0
0	1	0	1	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0
0	1	1	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0
0	1	1	1	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0
1	0	0	1	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0
1	0	1	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0
1	0	1	1	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0
1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0
1	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0
1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0
1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1



Circuit diagram —



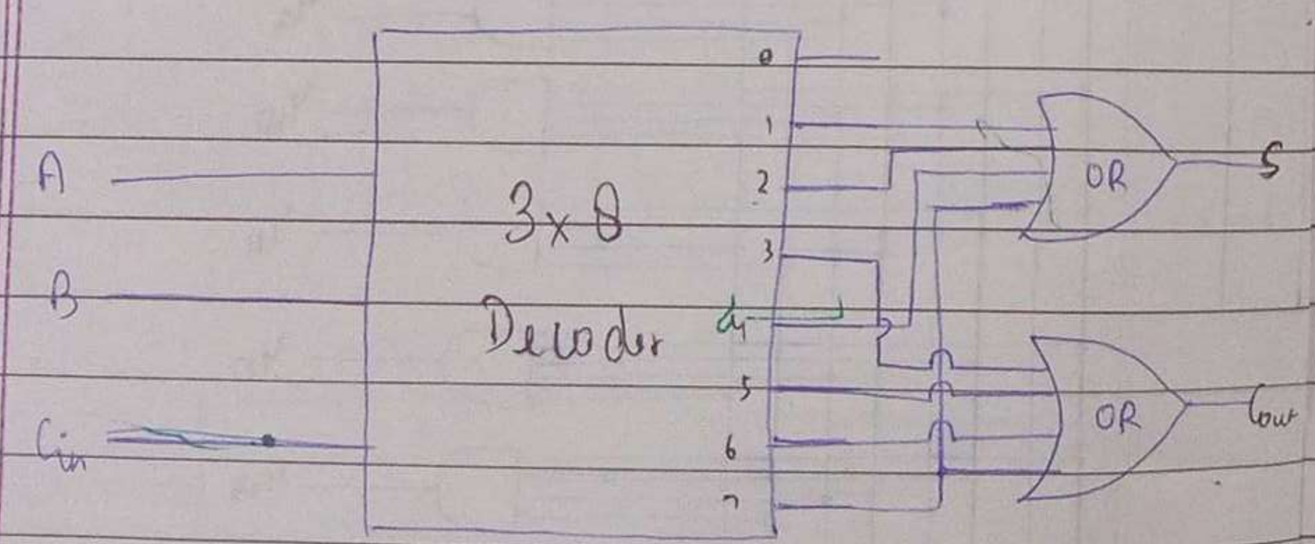
Q Implement full adder circuit with the help of Decoder.

Ans:-

	A	B	Cin	S	Cout
0	0	0	0	0	0
1	0	0	1	1	0
2	0	1	0	1	0
3	0	1	1	0	1
4	1	0	0	1	0
5	1	0	1	0	1
6	1	1	0	0	1
7	1	1	1	1	1

$S(A, B, C_{in}) = \Sigma_m(1, 2, 4, 7)$

$C_{out}(A, B, C_{in}) = \Sigma_m(3, 5, 6, 7)$

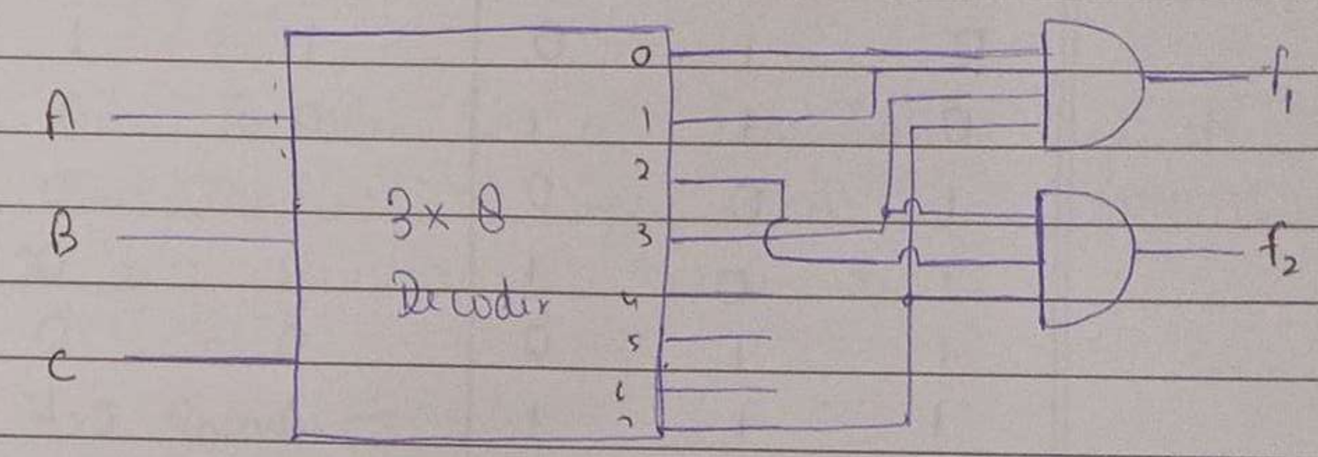


Q Implement the following f^n using Decoder AND gate -

① $f_1 = \Pi_m(0, 1, 3, 7)$

② $f_2 = \Pi m(2, 3, 7)$

Ans:-



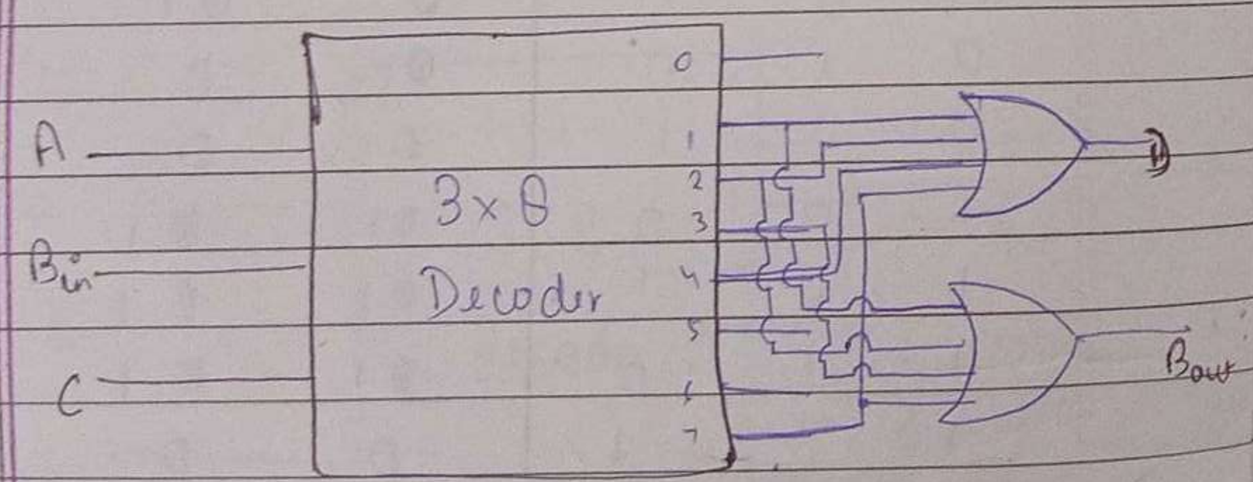
A	B	C	f_1	f_2
0	0	0	0	1
0	0	1	0	1
0	1	0	1	0
0	1	1	0	0
1	0	0	1	1
1	0	1	1	1
1	1	0	1	1
1	1	1	0	0

Q Implement Full Subtractor using Decoder—

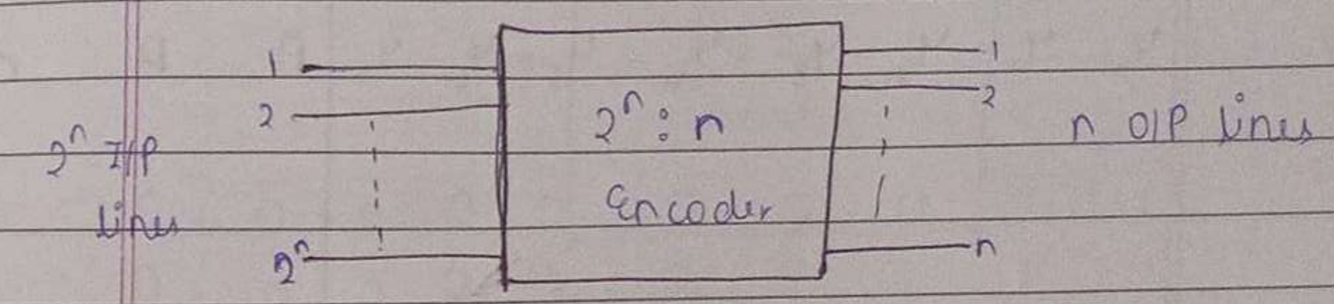
A	An-	A	B_{in}	C	D	B_{out}
0		0	0	0	0	0
0		0	0	1	1	1
0		0	1	0	1	1
0		0	1	1	0	1
0		1	0	0	1	0
0		1	0	1	0	0
0		1	1	0	0	0
0		1	1	1	1	1
1						
1						
1						
1						
1						
1						
1						
1						
1						

$S(A, B_{in}, C) = \Sigma_m(1, 2, 4, 7)$

$B_{out}(A, B_{in}, C) = \Sigma_m(1, 2, 3, 7)$



Encoder



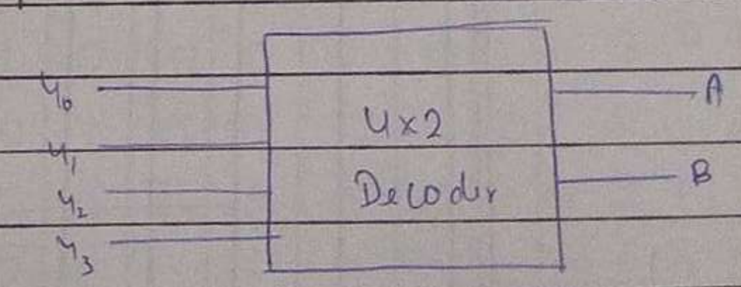
An Encoder is a combination logic ckt that convert 2^n input lines into n O/P lines. It is reverse of Decoder f^n.

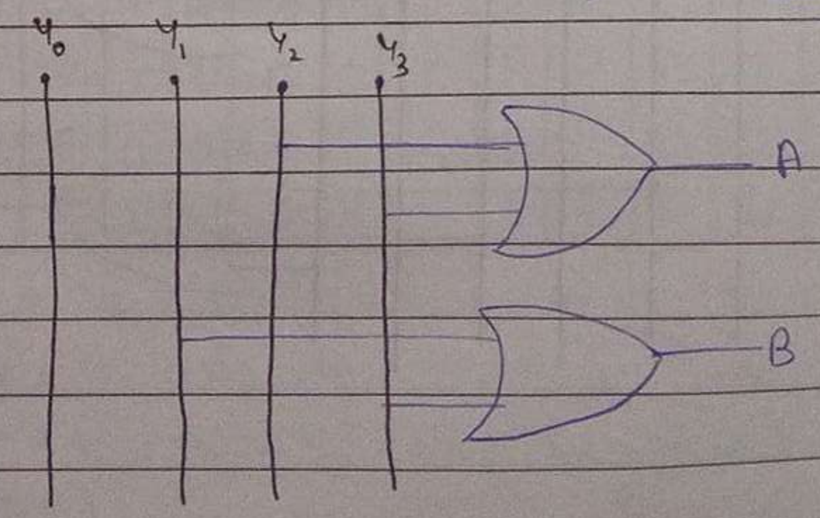
① 4×2 Encoder—

Y_0	Y_1	Y_2	Y_3	A	B
1	0	0	0	0	0
0	1	0	0	0	1
0	0	1	0	1	0
0	0	0	1	1	1

$A = Y_2 + Y_3$

$B = Y_1 + Y_3$





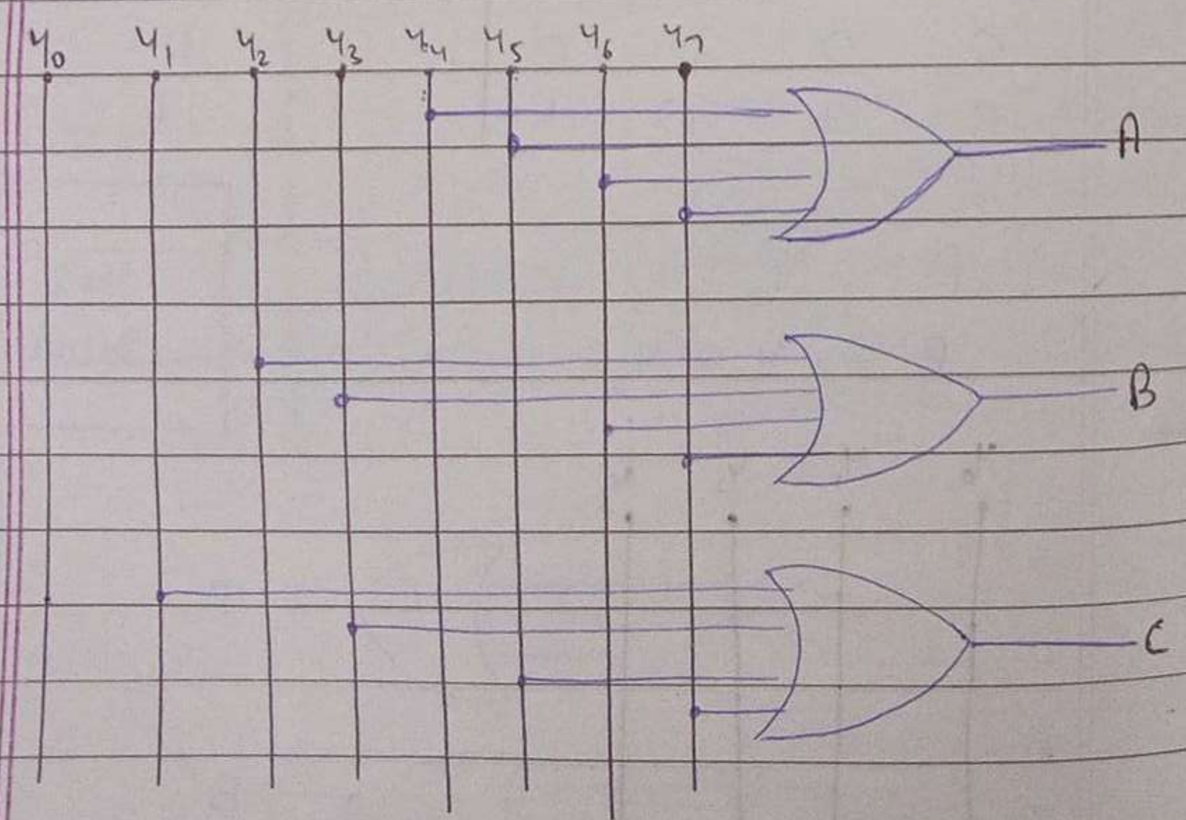
② 8x3 Decoder —

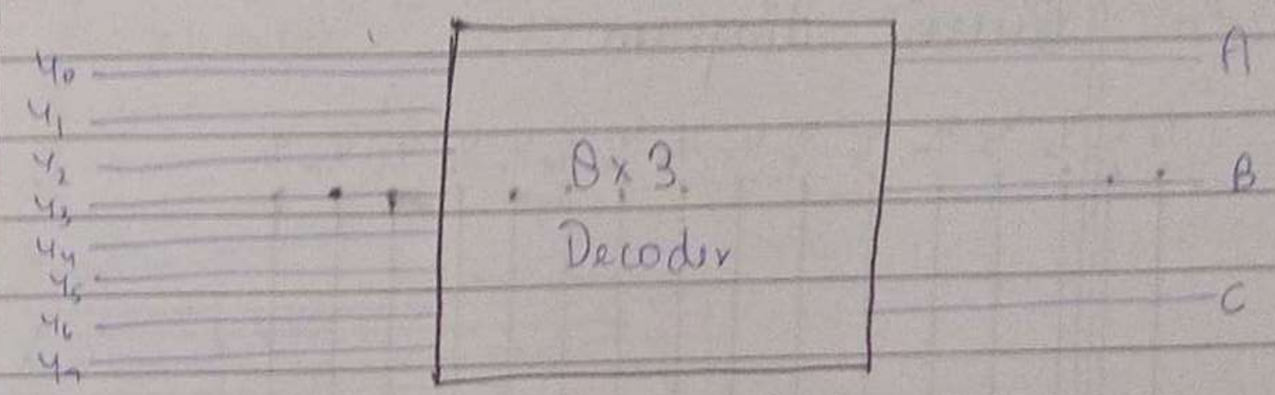
Y_0	Y_1	Y_2	Y_3	Y_4	Y_5	Y_6	Y_7	A	B	C
1	0	0	0	0	0	0	0	0	0	0
0	1	0	0	0	0	0	0	0	0	1
0	0	1	0	0	0	0	0	0	1	0
0	0	0	1	0	0	0	0	0	1	1
0	0	0	0	1	0	0	0	1	0	0
0	0	0	0	0	1	0	0	1	0	1
0	0	0	0	0	0	1	0	1	1	0
0	0	0	0	0	0	0	1	1	1	1

$A = Y_4 + Y_5 + Y_6 + Y_7$

$B = Y_2 + Y_3 + Y_6 + Y_7$

$C = Y_1 + Y_3 + Y_5 + Y_7$





③ 16x4 Encoder

Y_0	Y_1	Y_2	Y_3	Y_4	Y_5	Y_6	Y_7	Y_8	Y_9	Y_{10}	Y_{11}	Y_{12}	Y_{13}	Y_{14}	Y_{15}	A	B	C	D
1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0
0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1
0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0
0	0	0	0	0	1	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	1	1	0
0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	1	1	1
0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1	0	0	0
0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	1	0	0	1
0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	1	0	1	0
0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	1	0	1	1
0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	1	1	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1	1	0	1
0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	1	1	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1

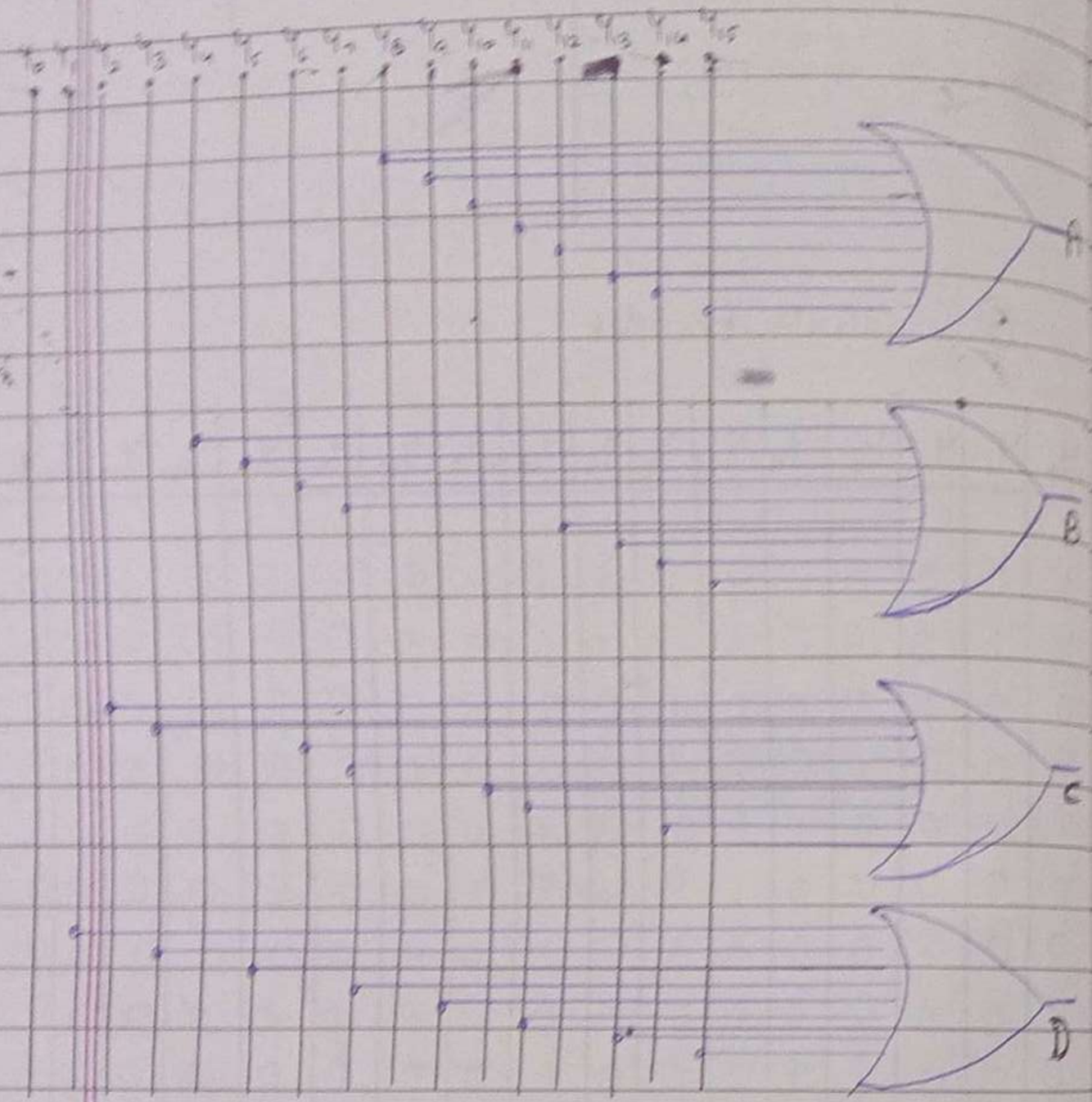
$A = Y_8 + Y_9 + Y_{10} + Y_{11} + Y_{12} + Y_{13} + Y_{14} + Y_{15}$

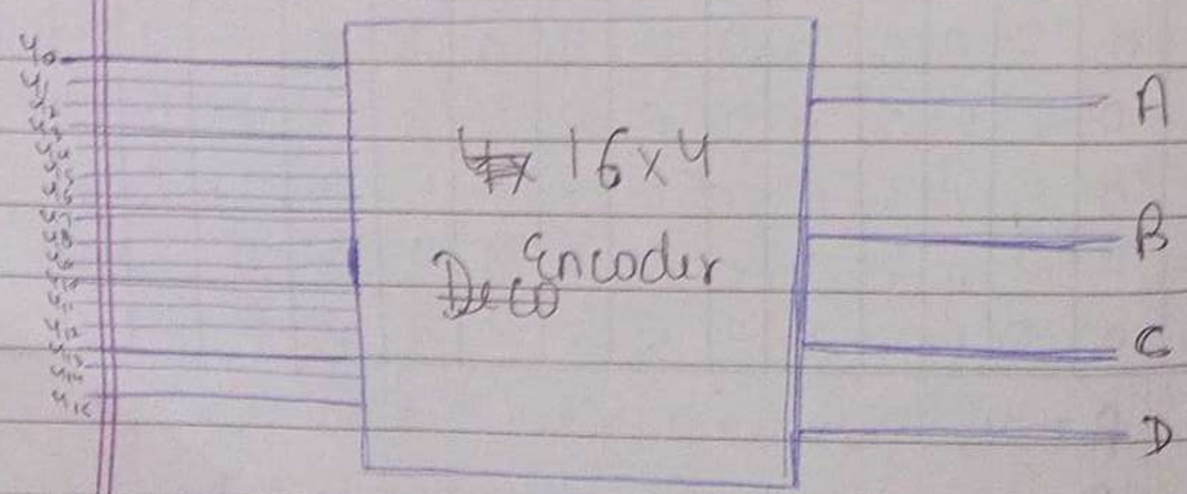
$B = Y_4 + Y_5 + Y_6 + Y_7 + Y_{12} + Y_{13} + Y_{14} + Y_{15}$

$C = Y_2 + Y_3 + Y_6 + Y_7 + Y_{10} + Y_{11} + Y_{14} + Y_{15}$

$D = Y_1 + Y_3 + Y_5 + Y_7 + Y_9 + Y_{11} + Y_{13} + Y_{15}$

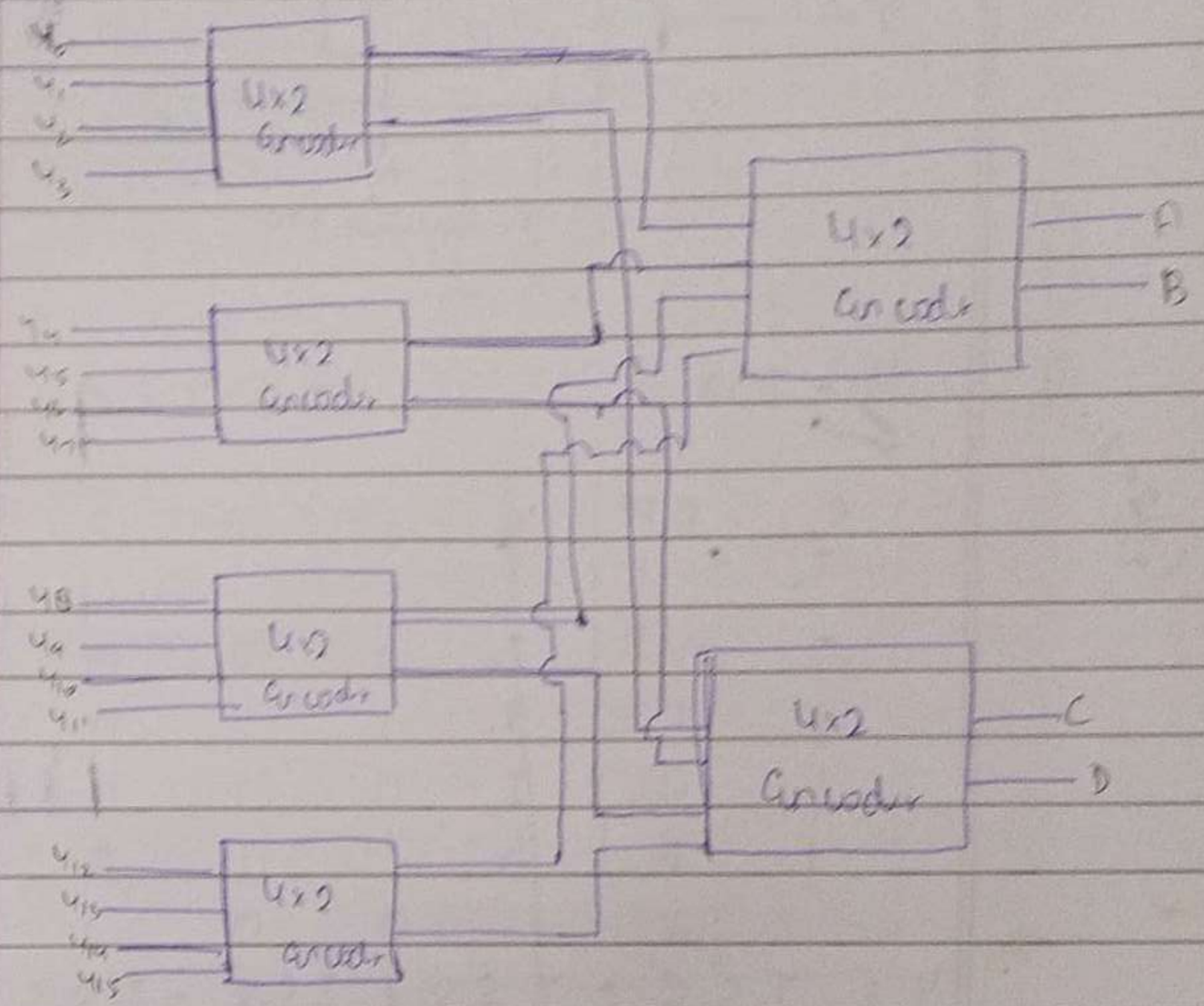
Circuit diagram :-





Q Implement 16x4 Encoder with the help of 4x2 Encoder.

Ans

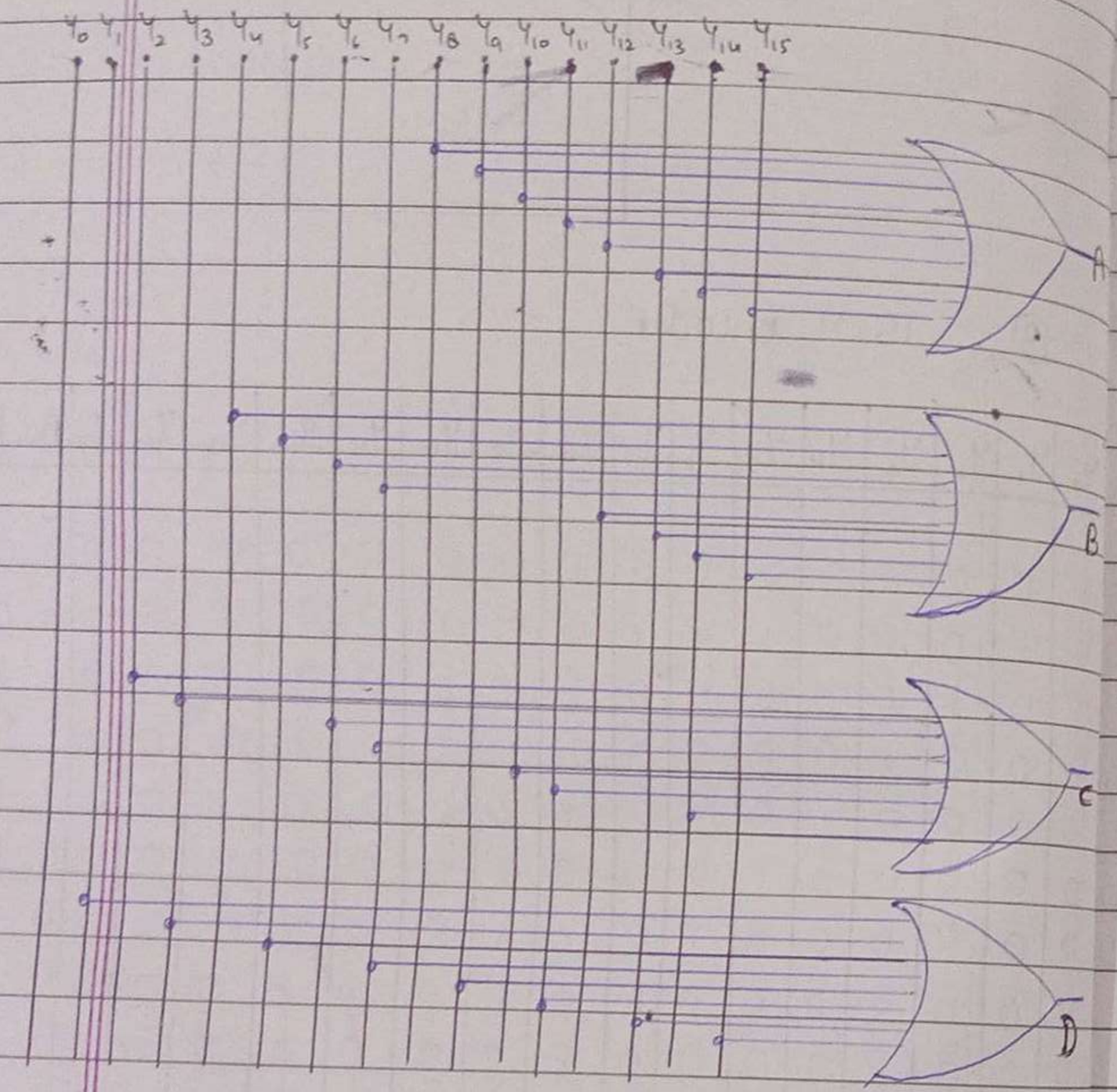


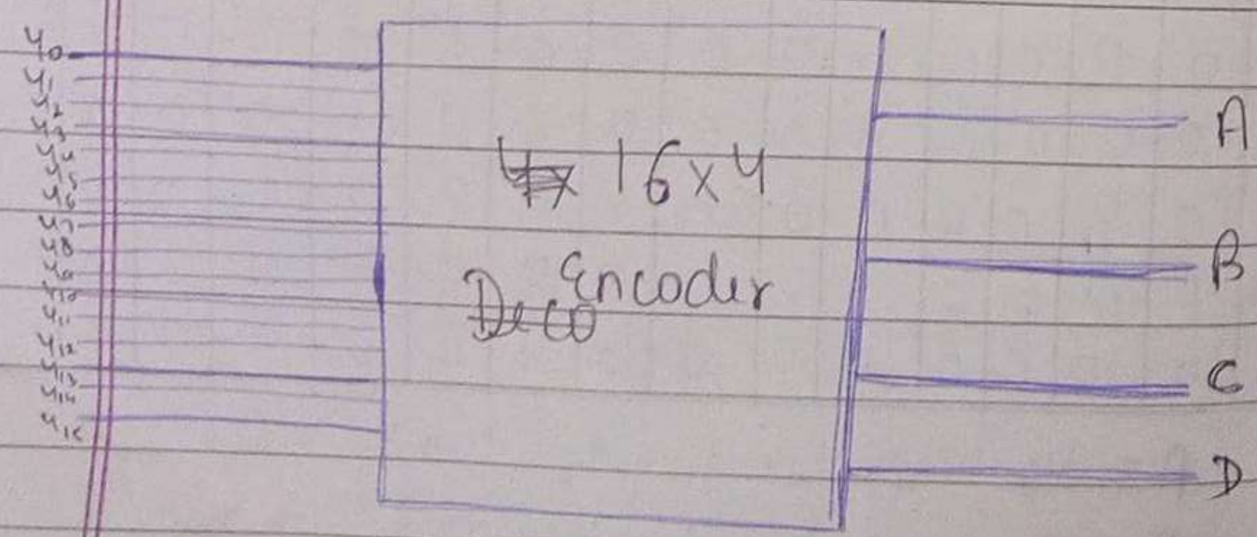
Multiplexer (MUX)

A Multiplexer is also known as Data Selector which provide a number of inputs one by one parallelingin at a single o/p and which i/p will be at the o/p, depends upon the states of selection lines at that instance. It is also known as Parallel to Serial Converter.

A
0
0
0
0
0
0
0
0
1
1
1
1
1
1
1
1

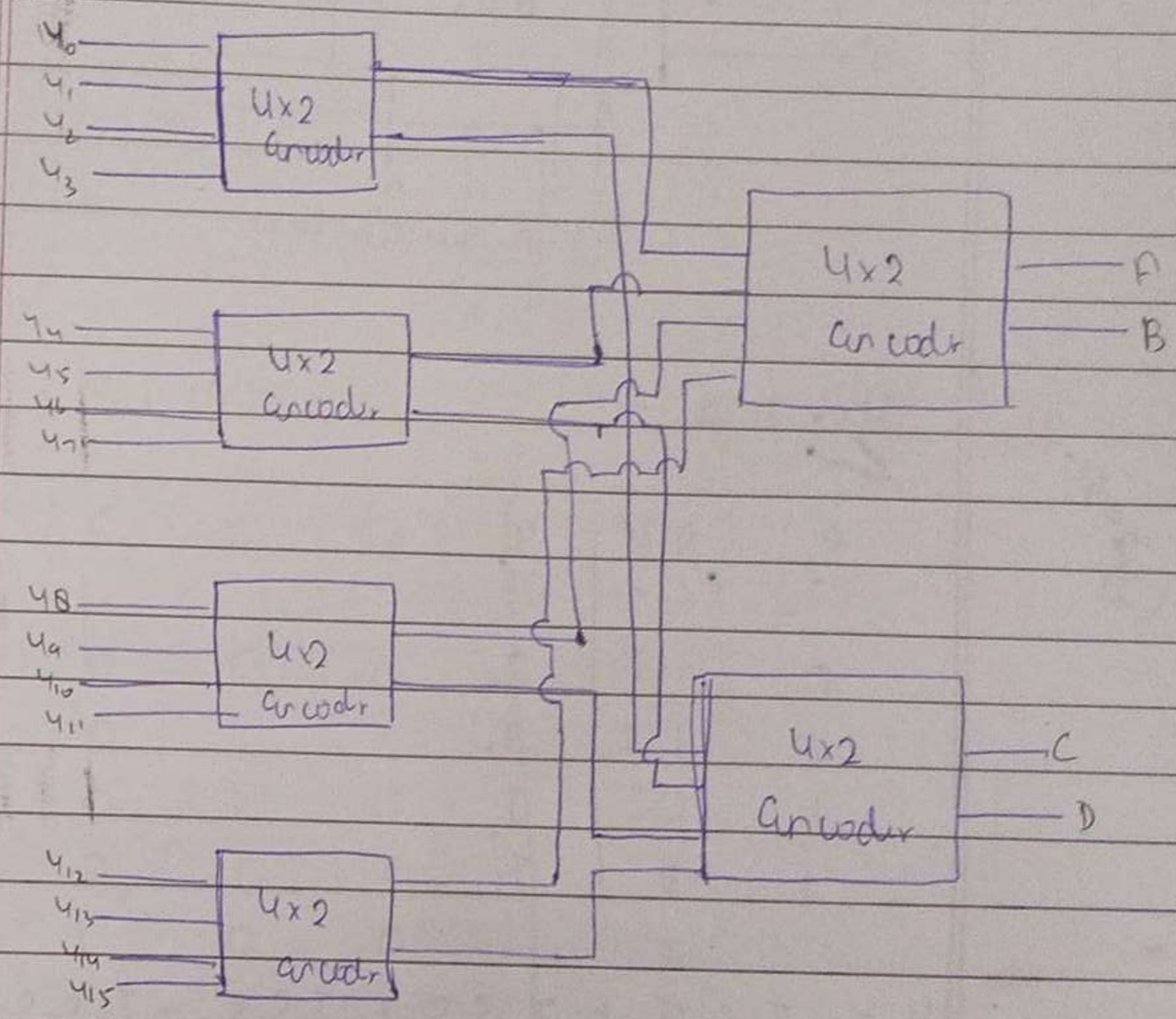
Circuit diagram :-





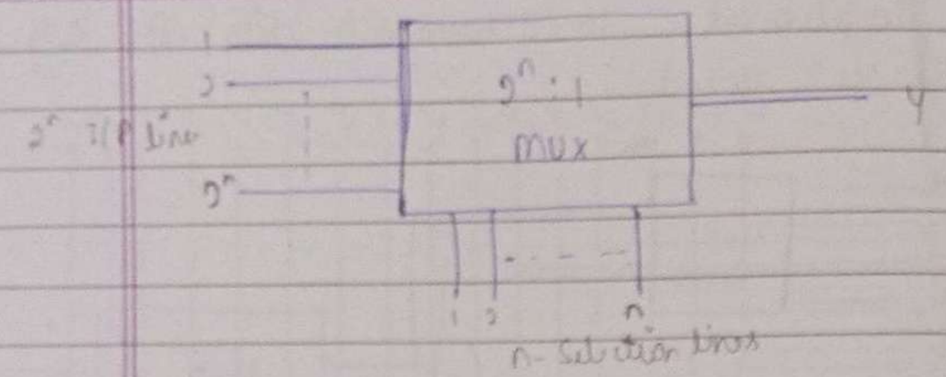
Q Implement 16x4 encoder with the help of 4x2 encoder.

Ans:



Multiplexer (MUX)

A Multiplexer is also known as Data Selector which provides a number of inputs one by one parallelingto at a single o/p and which i/p will be at the o/p, depends upon the status of selection lines at that instance. It is also known as Parallel to Serial Converter.

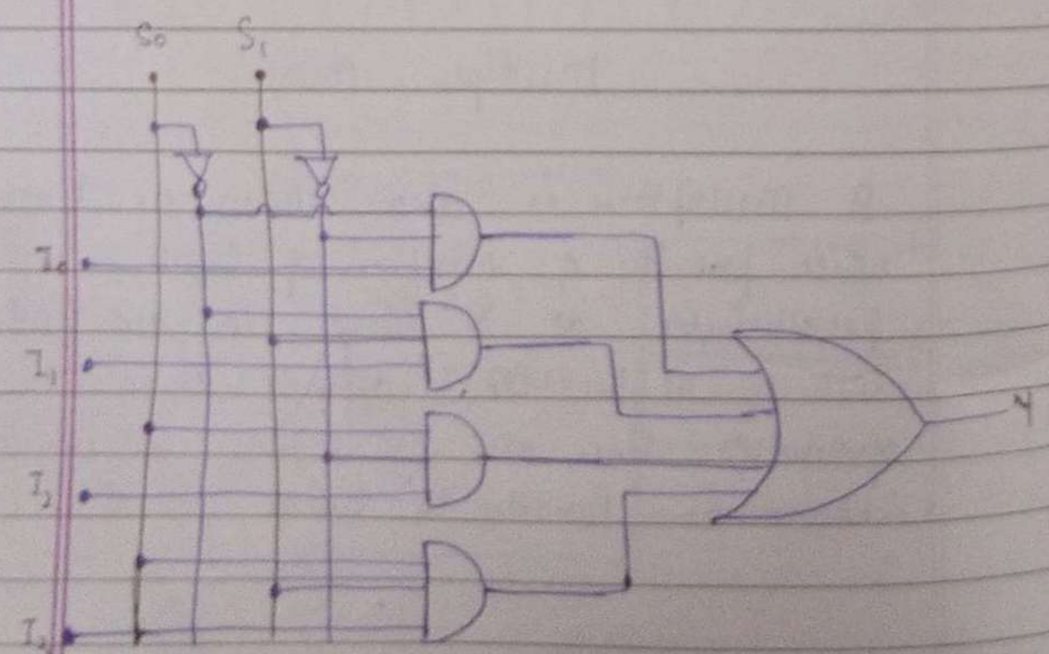


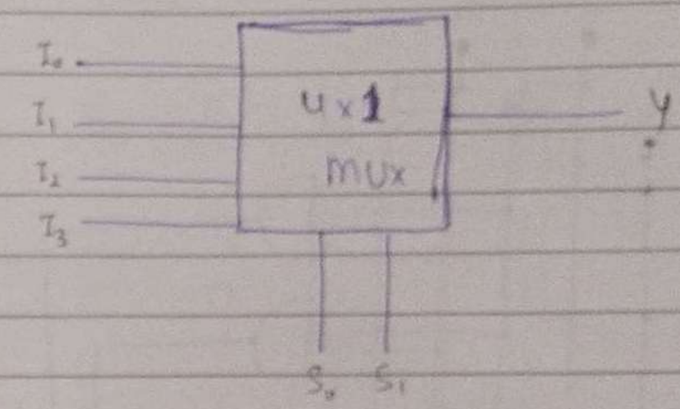
$(2)^2 = 4L$ ①

4x1 mux —

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

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

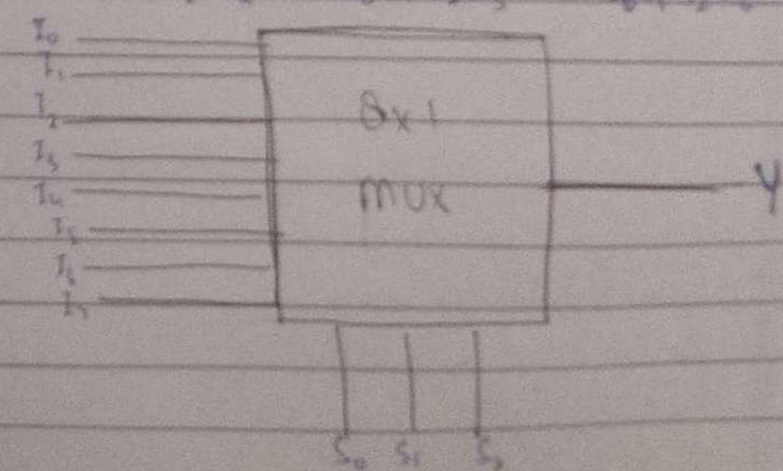


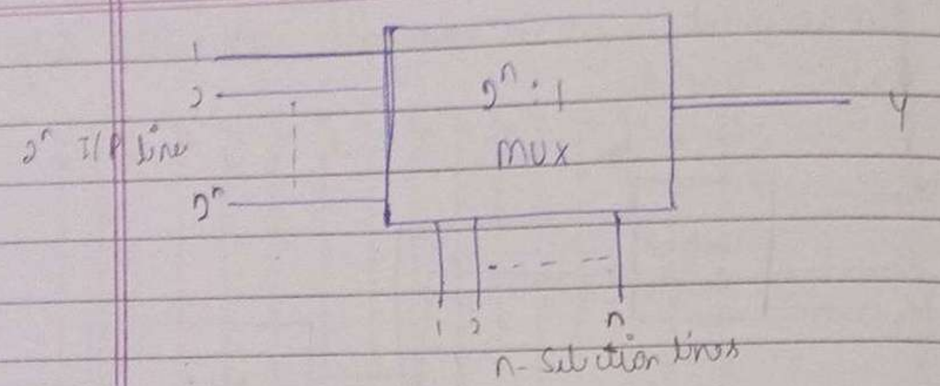


② 8x1 mux —

S_0	S_1	S_2	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

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



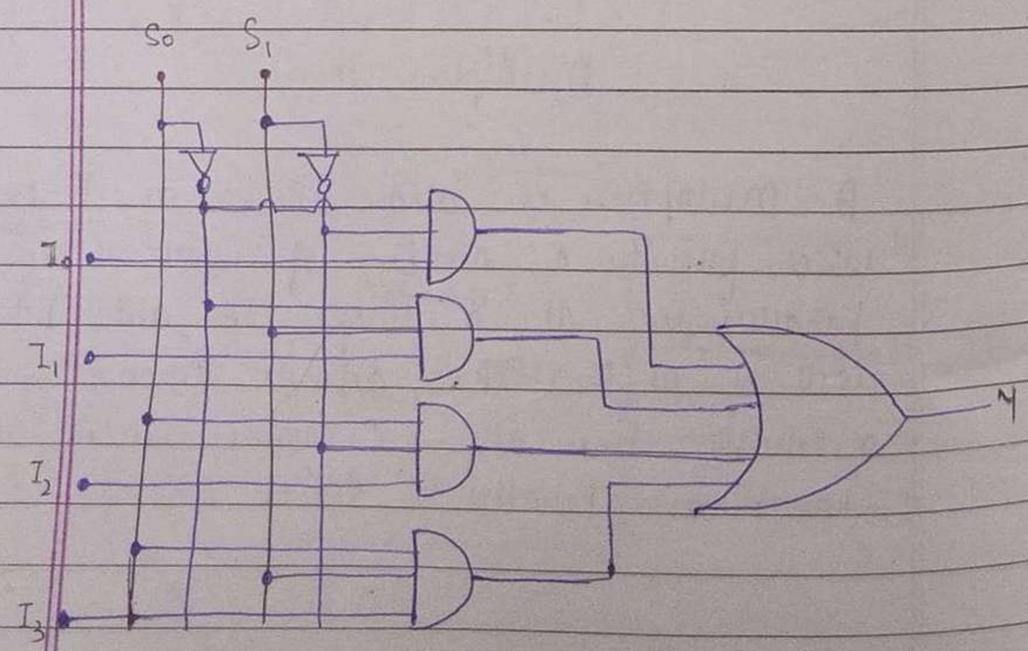


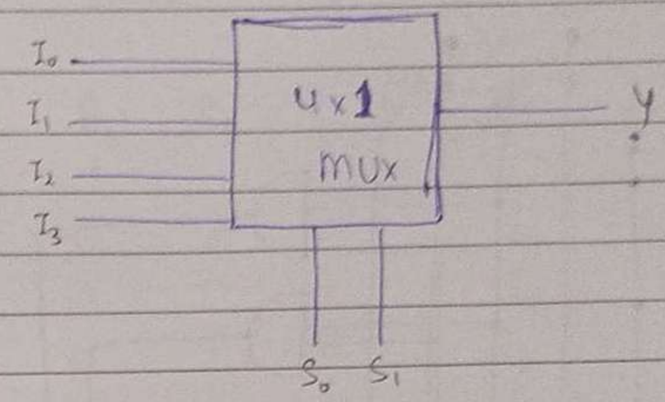
① 4x1 mux —

$(2)^2 = 4$ L.

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

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

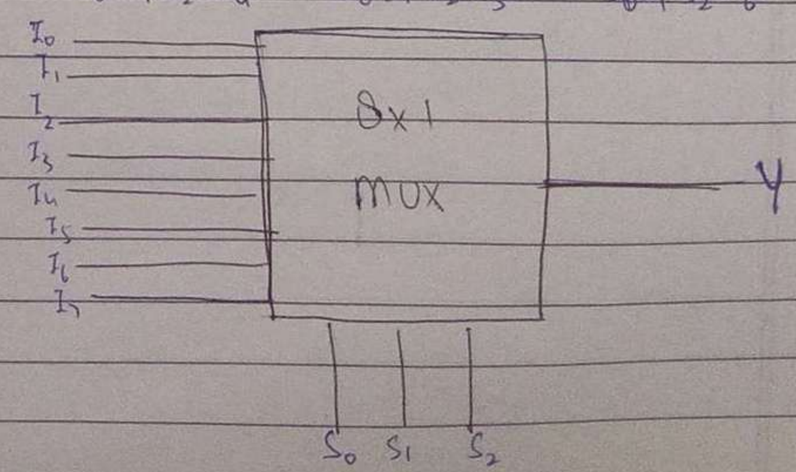


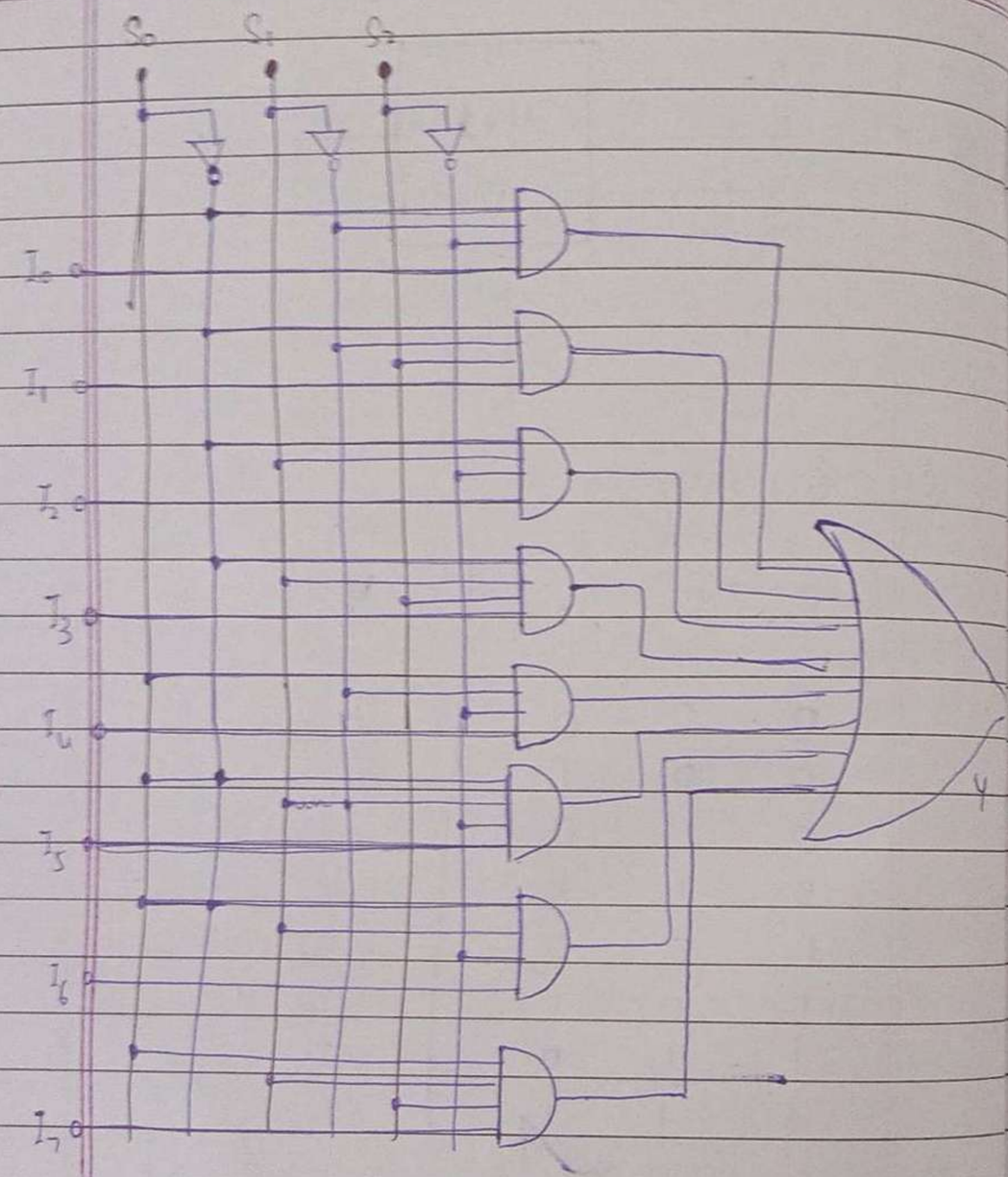


② 8x1 mux —

S_0	S_1	S_2	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

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



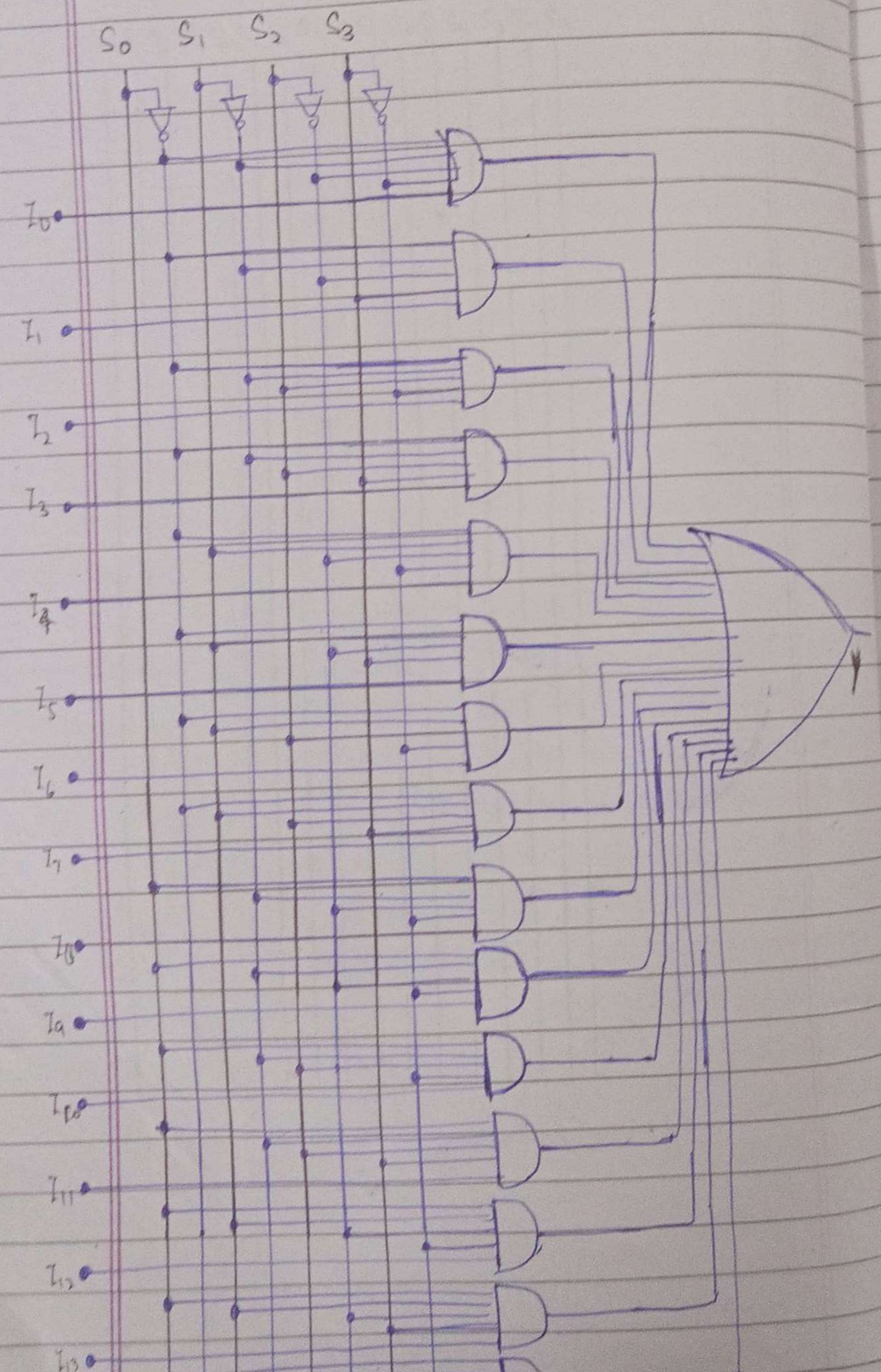


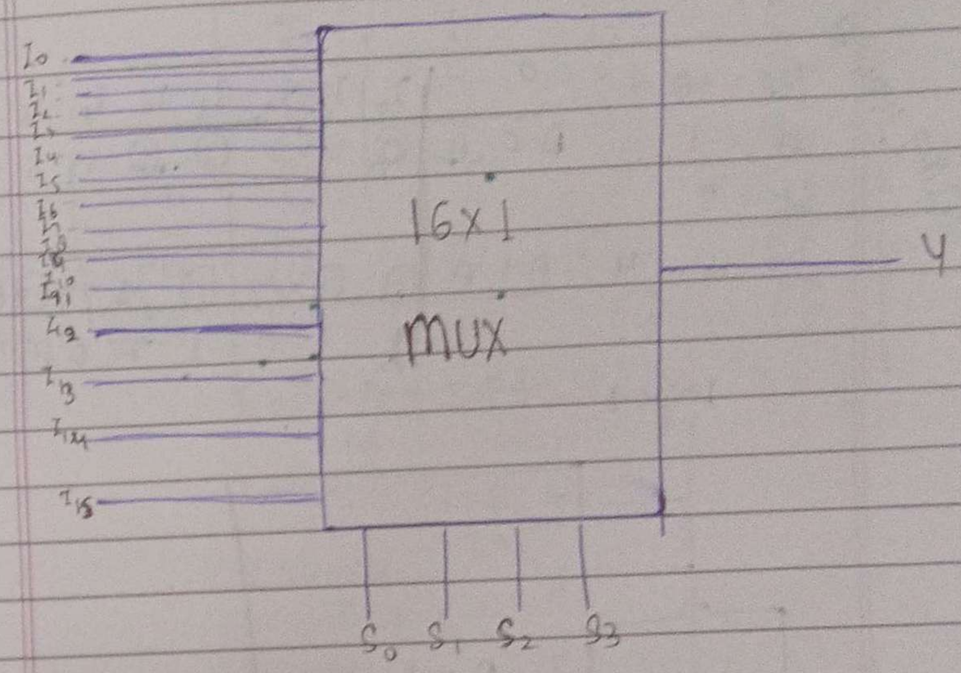
② 16x1 MUX —

S_0	S_1	S_2	S_3	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}

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

Circuit Diagram —





Q Implement the following function using —

(1) 16x1 MUX

(2) 8x1 mux →

(3) 4x1 MUX

(4) 2x1 MUX

S.L = Var

$f(A, B, C, D) = \Sigma_m(0, 1, 3, 4, 7, 8, 9, 11, 14, 15)$

(2)^{4} = 32

Sol: (1) Using 16x1 MUX —

Logic 1

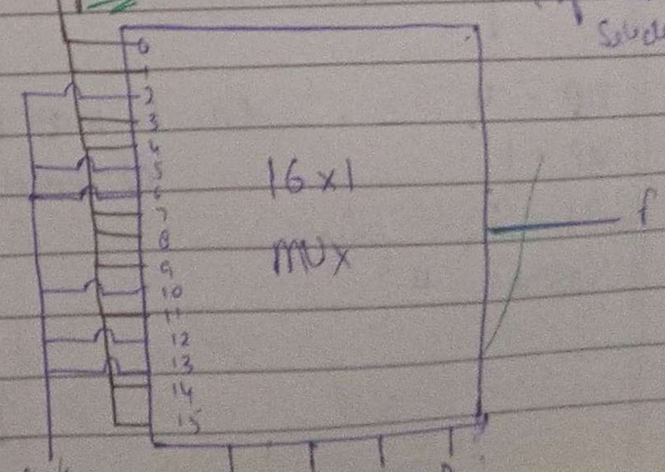
(If I/P variable & Selection lines are equal)

I/P = 16

O/P = 1

S.L = 4

Variable = 4.



$\bar{A}+A=1$

$A\bar{A}=1$

② Using 8x1 mux —

I/P = 8

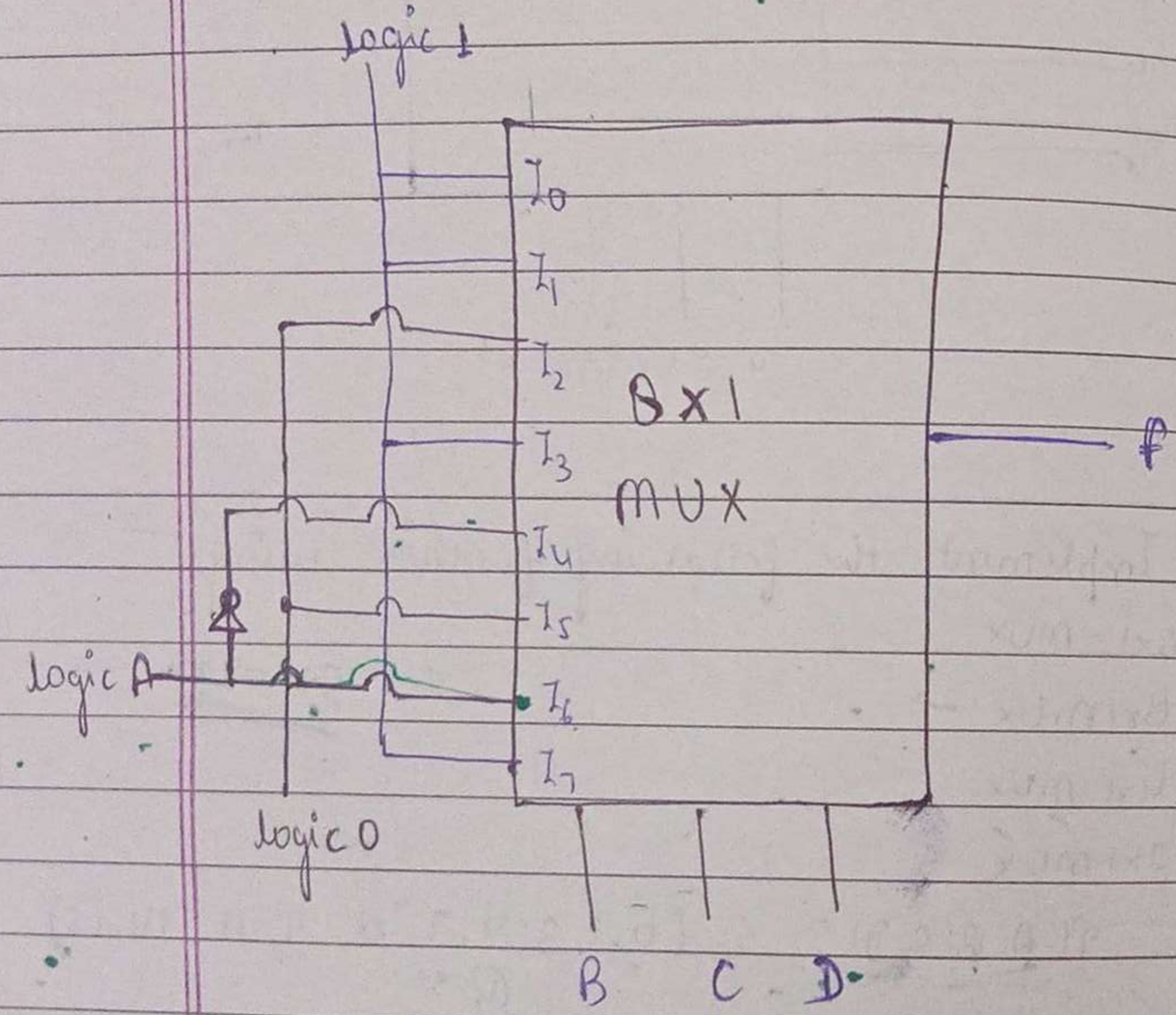
O/P = 1

SL = 3

variable = 4

$f = \Sigma m(0,1,3,4,7,8,9,11,14,15)$

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



③ Using 4x1 mux —

I/P = 4

O/P = 1

S.L = 2

Variable = 4

Let A and B are using as selection line. I/P.

	I_0	I_1	I_2	I_3
$\bar{A}\bar{B}=00$, $\bar{A}\bar{B}$	0	1	2	3
$\bar{A}B=01$, $\bar{A}B$	4	5	6	7
$A\bar{B}=10$, $A\bar{B}$	8	9	10	11
$AB=11$, AB	12	13	14	15

$I_0 = \bar{A}\bar{B}+\bar{A}B+A\bar{B}$

$I_0 = \bar{A}(\bar{B}+B)+A\bar{B}$

$I_0 = \bar{A}+AB$

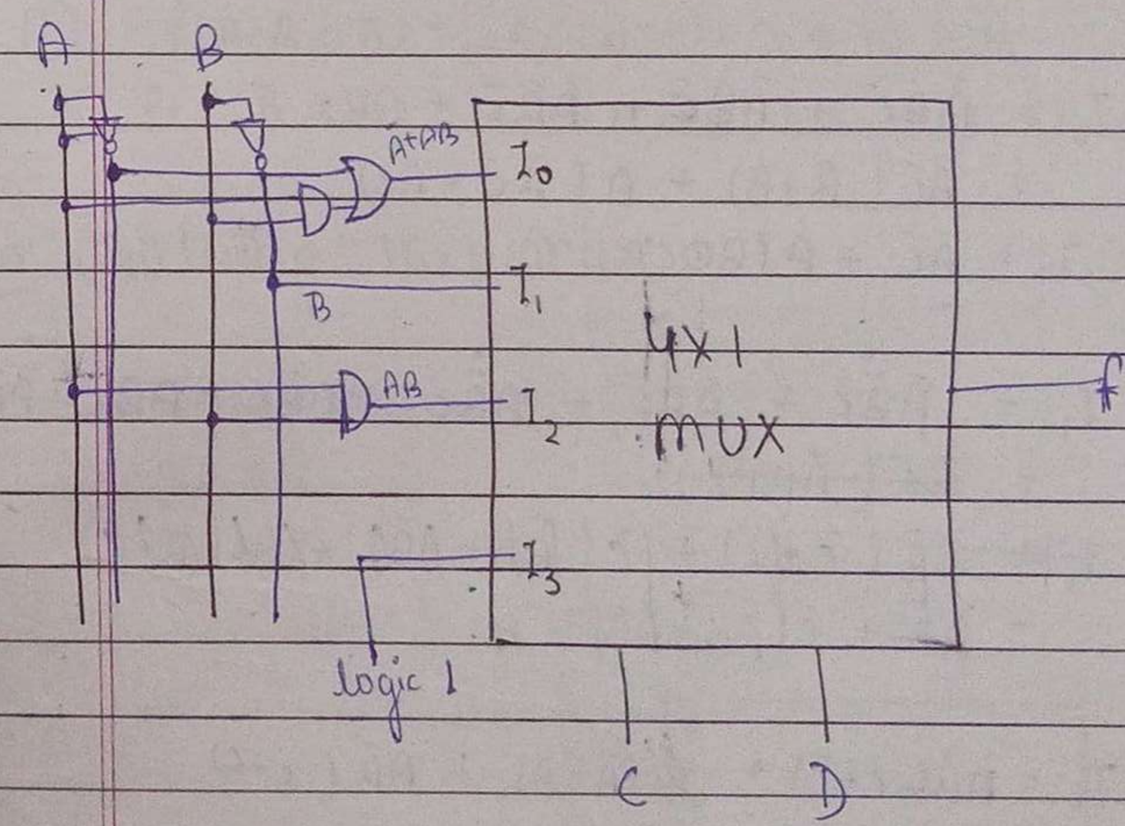
$I_1 = \bar{A}\bar{B}+A\bar{B}$

$I_1 = \bar{B}(A+\bar{A})$

$I_1 = \bar{B}$

$I_2 = AB$

$I_3 = 1$



④ 2x1 mux —

I/P = 2

O/P = 1

S.L = 1

Variable = 4

Using A, B, C as input.

Implementation table —

	I_0	I_1
$ABC = 000$, $\bar{A}\bar{B}\bar{C}$	0	1
$ABC = 001$, $\bar{A}\bar{B}C$	2	3
$ABC = 010$, $\bar{A}B\bar{C}$	4	5
$ABC = 011$, $\bar{A}BC$	6	7
$ABC = 100$, $A\bar{B}\bar{C}$	8	9
$ABC = 101$, $A\bar{B}C$	10	11
$ABC = 110$, $AB\bar{C}$	12	13
$ABC = 111$, ABC	14	15

$I_0 = \bar{A}\bar{B}\bar{C} + \bar{A}B\bar{C} + A\bar{B}\bar{C} + ABC$

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

$I_0 = \bar{A}\bar{C} + A(B \odot C)$

$I_1 = \bar{A}\bar{B}\bar{C} + \bar{A}B\bar{C} + A\bar{B}\bar{C} + A\bar{B}C + ABC + A\bar{B}C$

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

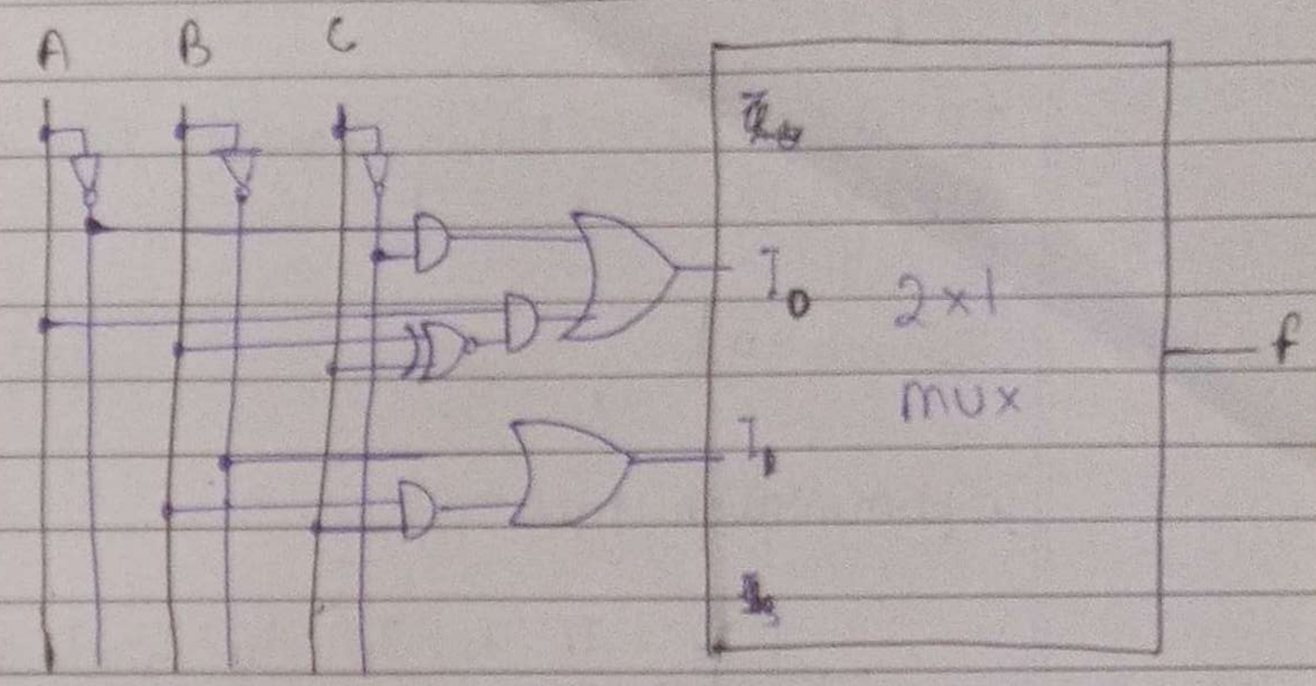
$\cancel{I_1 = \bar{A}\bar{B}(\bar{C}+C) + \bar{C}(\bar{A}B + A\bar{B}) + AC(B+\bar{B})}$

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

$I_1 = \bar{A}\bar{B}(C+\bar{C}) + B\bar{C}(\bar{A}+A) + A\bar{B}(\bar{C}+C)$

$I_1 = \bar{A}\bar{B} + B\bar{C} + A\bar{B}$

$I_1 = \bar{B} + BC.$



② $f(A,B,C,D) = \Sigma_m(0,1,4,7,9,10,11,14)$

③ $f(A,B,C,D) = \Sigma_m(1,2,6,7,8,11,12,15)$

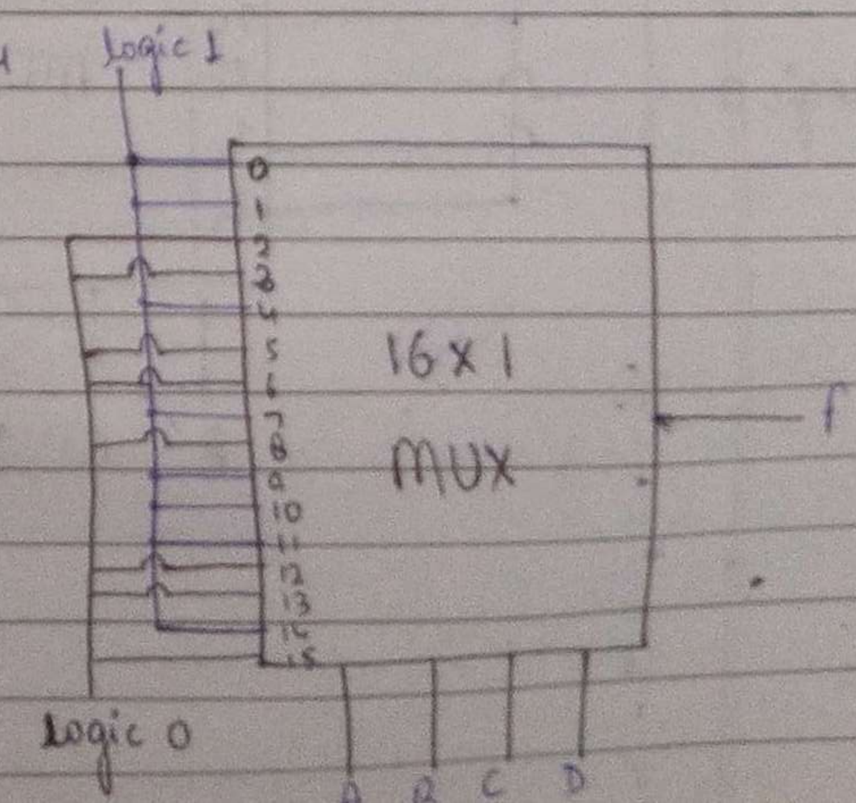
Sol' ②(i) Using 16×1 MUX.

Input = 16

O/P = 1

S.L = 4

Variables = 4

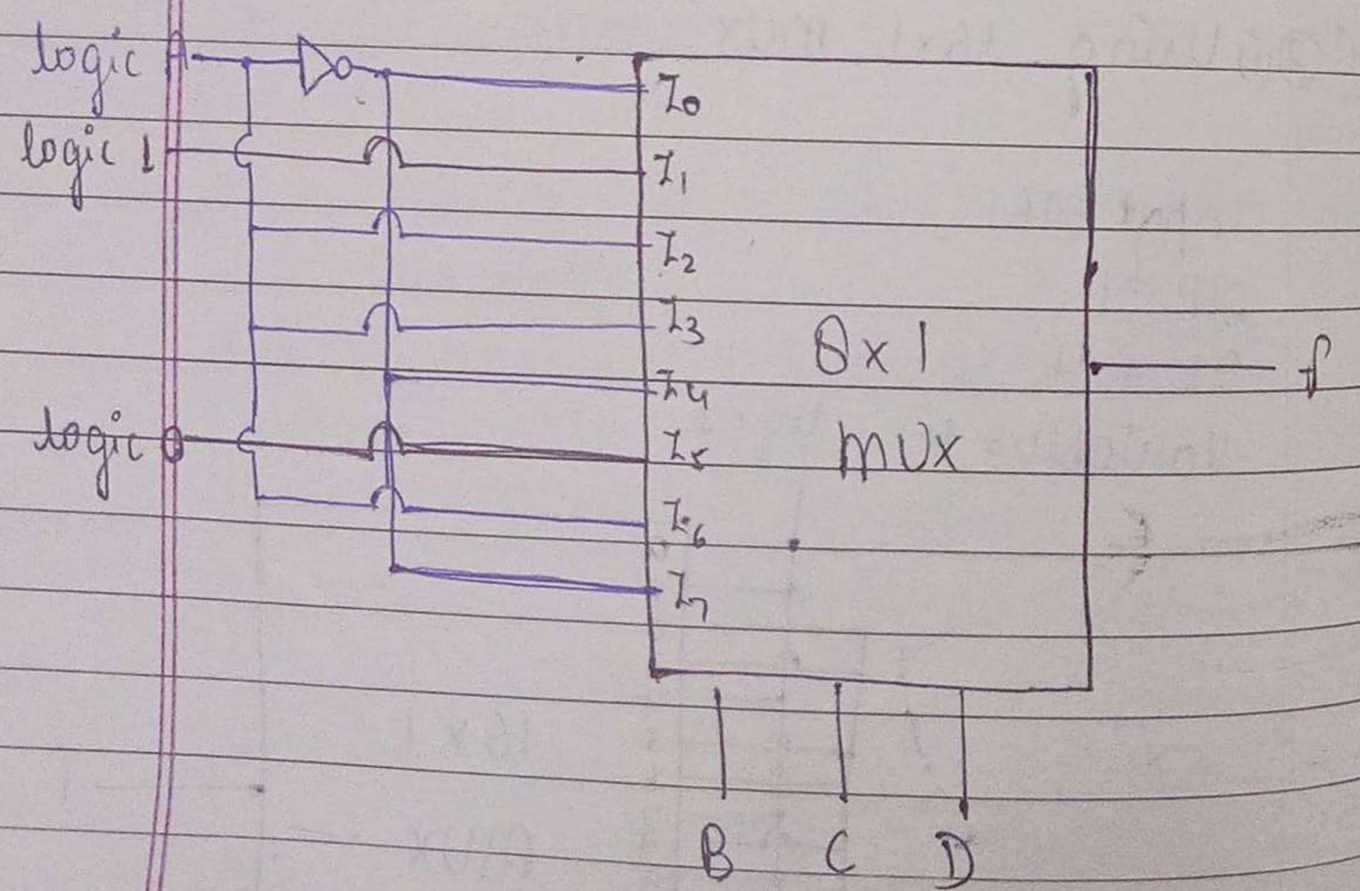


(ii) Using 8x1 MUX –

I/P = 8 Using Variable A as input
O/P = 1
S.L = 3
Variable = 4

Implementation table –

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

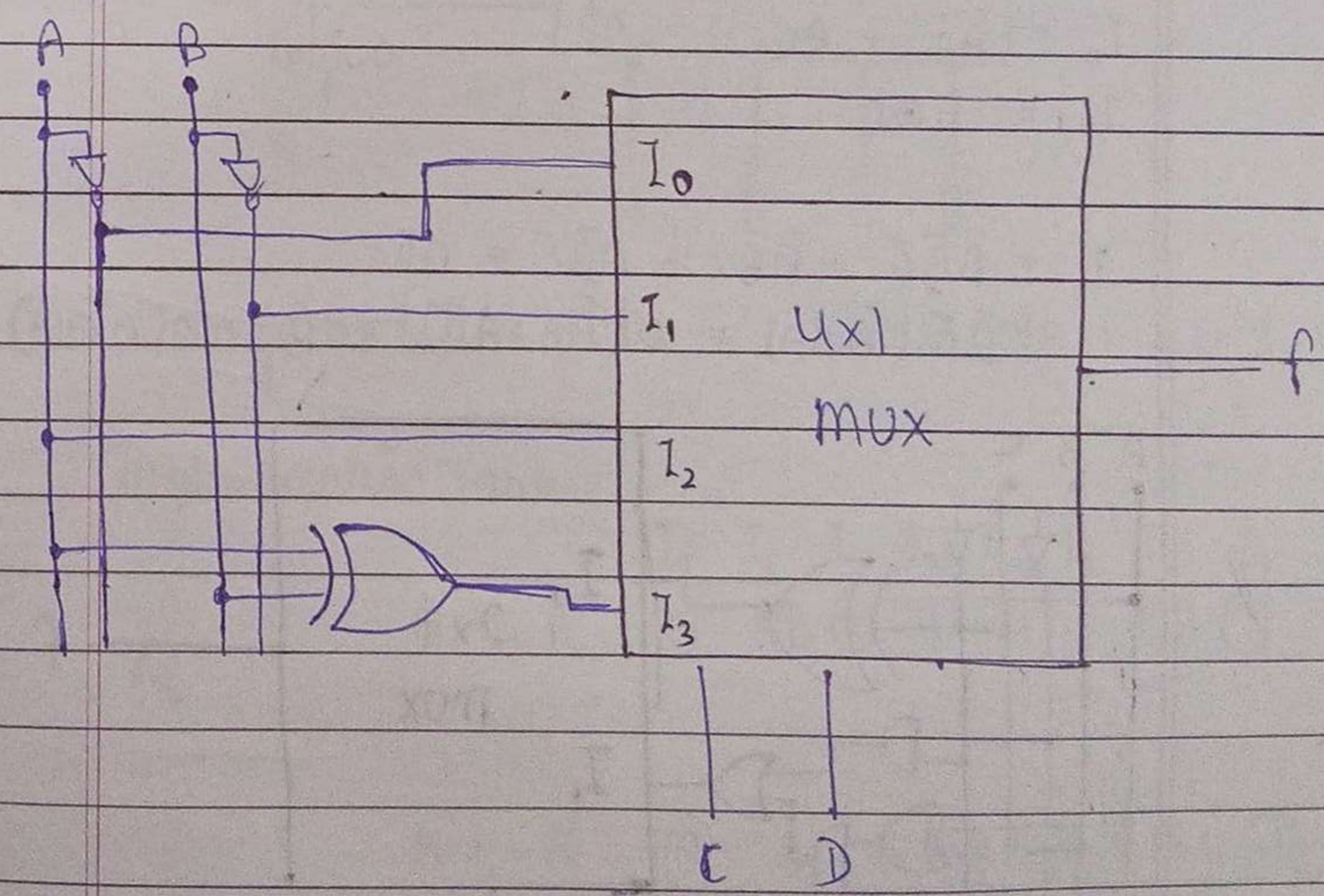


(iii) Using 4x1 MUX –

I/P = 4, O/P = 1, S.L = 2, Variable = 4
Let A and B are using at Input.

Implementation Table –

	I_0	I_1	I_2	I_3
AB=00, $\bar{A}\bar{B}$	0	1	2	3
AB=01, $\bar{A}B$	4	5	6	7
AB=10, $A\bar{B}$	8	9	10	11
AB=11, AB	12	13	14	15
	$= \bar{A}\bar{B}+\bar{A}B$	$I_1 = \bar{A}\bar{B}+A\bar{B}$	$I_2 = A\bar{B}+AB$	$I_3 = \bar{A}B+A\bar{B}$
	$=\bar{A}(\bar{B}+B)$	$=\bar{B}(\bar{A}+A)$	$=A(\bar{B}+B)$	
	$I_0 = \bar{A}$	$I_1 = \bar{B}$	$I_2 = A$	$I_3 = A \oplus B$



(iv) Using 2x1 MUX —

Implementation Table — I/N = 2, O/P = 1, S.L = 1, Var = 4

	I_0	I_1
$ABC = 000$, $\bar{A}\bar{B}\bar{C}$	0	1
$ABC = 001$, $\bar{A}\bar{B}C$	2	3
$ABC = 010$, $\bar{A}B\bar{C}$	4	5
$ABC = 011$, $\bar{A}BC$	6	7
$ABC = 100$, $A\bar{B}\bar{C}$	8	9
$ABC = 101$, $A\bar{B}C$	10	11
$ABC = 110$, $AB\bar{C}$	12	13
$ABC = 111$, ABC	14	15

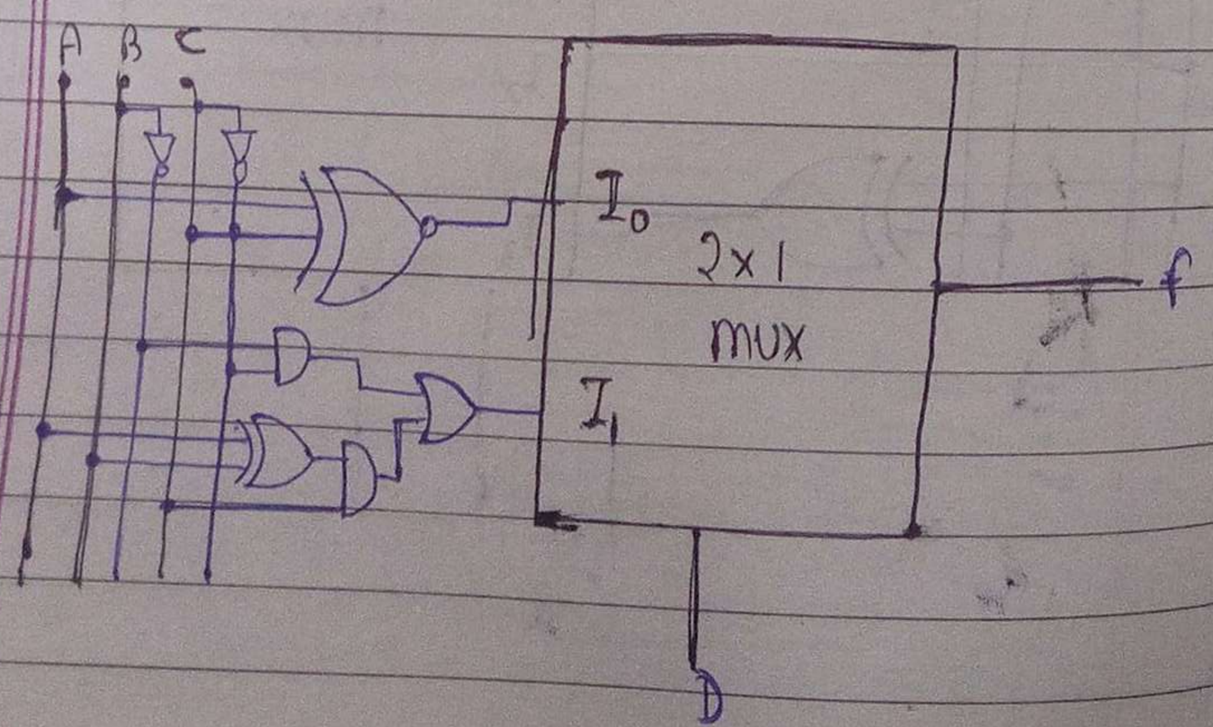
$I_0 = \bar{A}\bar{B}\bar{C} + \bar{A}B\bar{C} + A\bar{B}C + ABC$

$I_0 = \bar{A}\bar{C} + AC$

$I_0 = A \odot C$

$I_1 = \bar{A}\bar{B}\bar{C} + \bar{A}BC + A\bar{B}\bar{C} + A\bar{B}C$

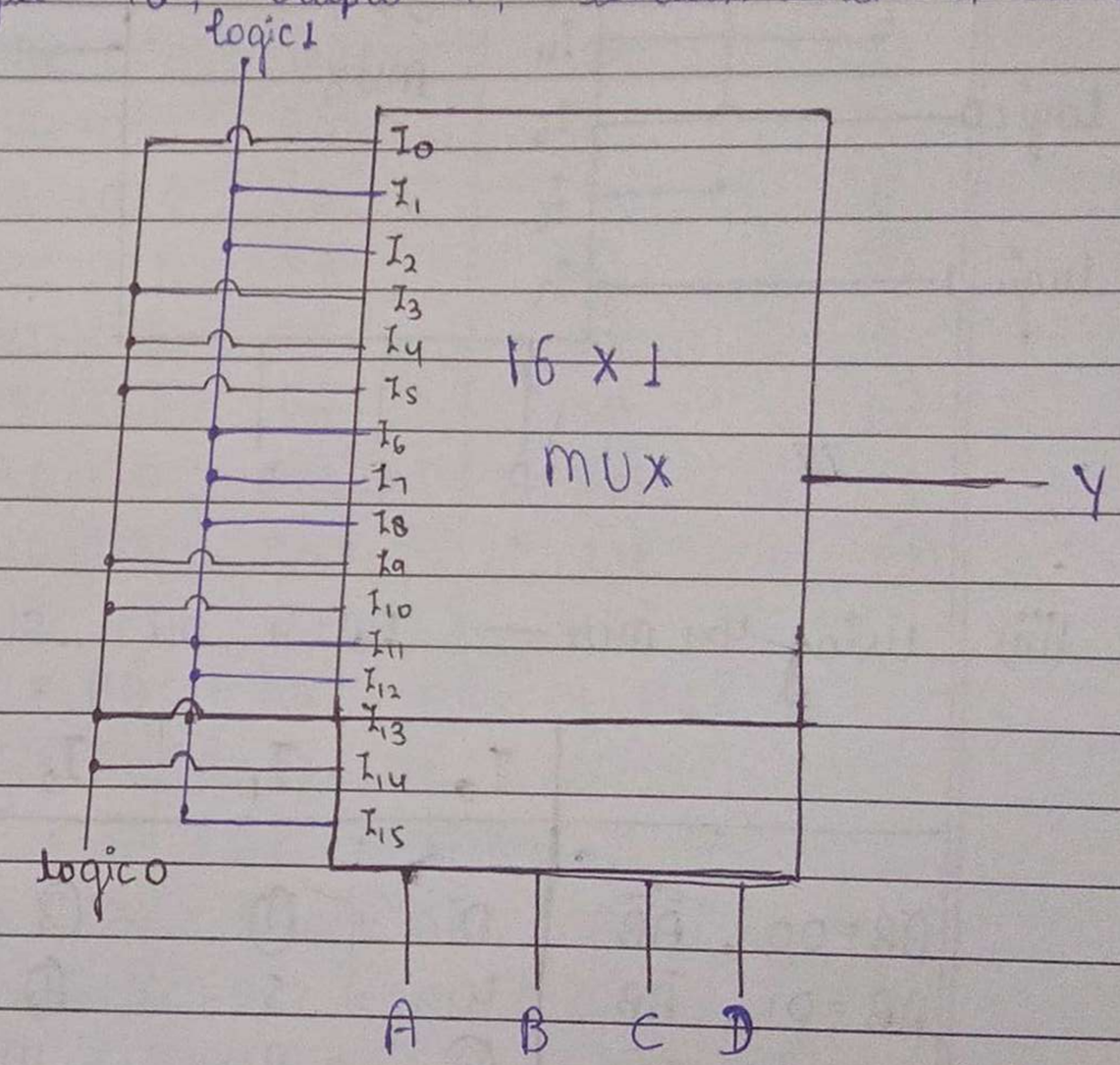
$= \bar{B}\bar{C}(\bar{A}+A) + C(\bar{A}B + A\bar{B}) = \bar{B}\bar{C} + C(A \oplus B)$



(2) $f(A,B,C,D) = \Sigma_m(1,2,6,7,8,11,12,15)$

(i) Using 16x1 mux —

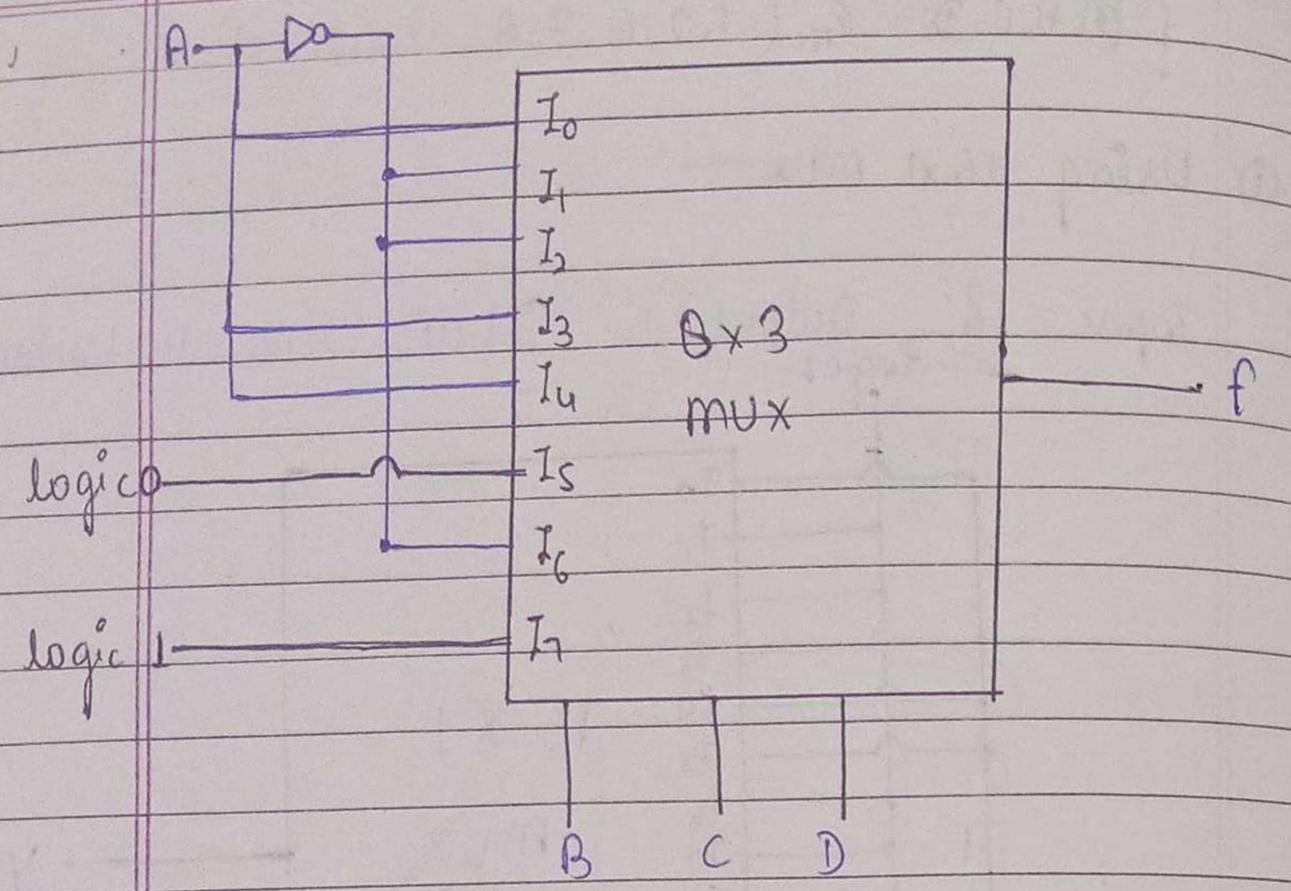
Input = 16, Output = 1, Selection lines = 4, Variables = 4



(ii) Using ~~8x3~~ 8x1 mux — I/N = 8, O/P = 1, S.L = 3, Var = 4

Implementation Table —

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



(iii) Using 4x1 mux — I/P = 4, O/P = 1, S.L = 2, Var = 4

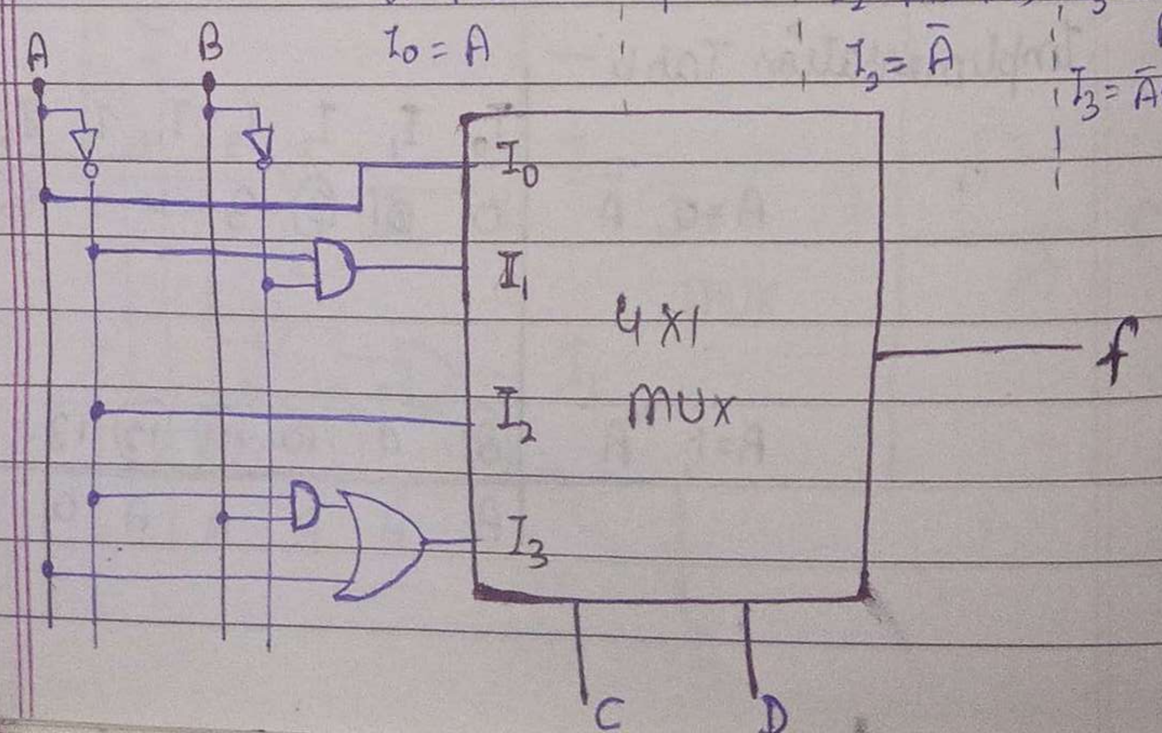
	I_0	I_1	I_2	I_3
$AB=00$, $\bar{A}\bar{B}$	0	1	2	3
$AB=01$, $\bar{A}B$	4	5	6	7
$AB=10$, $A\bar{B}$	8	9	10	11
$AB=11$, AB	12	13	14	15

$I_0 = A\bar{B}+AB$; $I_0 = A$

$I_1 = \bar{A}\bar{B}$

$I_2 = \bar{A}\bar{B}+\bar{A}B$; $I_2 = \bar{A}$

$I_3 = \bar{A}B+A\bar{B}+AB$; $I_3 = \bar{A}B+A$



(iv) Using 2x1 mux — I/P = 2, O/P = 1, S.L = 1, Var = 4

Implementation Table —

	I_0	I_1
$ABC=000$, $\bar{A}\bar{B}\bar{C}$	0	1
$ABC=001$, $\bar{A}\bar{B}C$	2	3
$ABC=010$, $\bar{A}B\bar{C}$	4	5
$ABC=011$, $\bar{A}BC$	6	7
$ABC=100$, $A\bar{B}\bar{C}$	8	9
$ABC=101$, $A\bar{B}C$	10	11
$ABC=110$, $AB\bar{C}$	12	13
$ABC=111$, ABC	14	15

$I_0 = \bar{A}\bar{B}C + \bar{A}BC + A\bar{B}\bar{C} + AB\bar{C}$

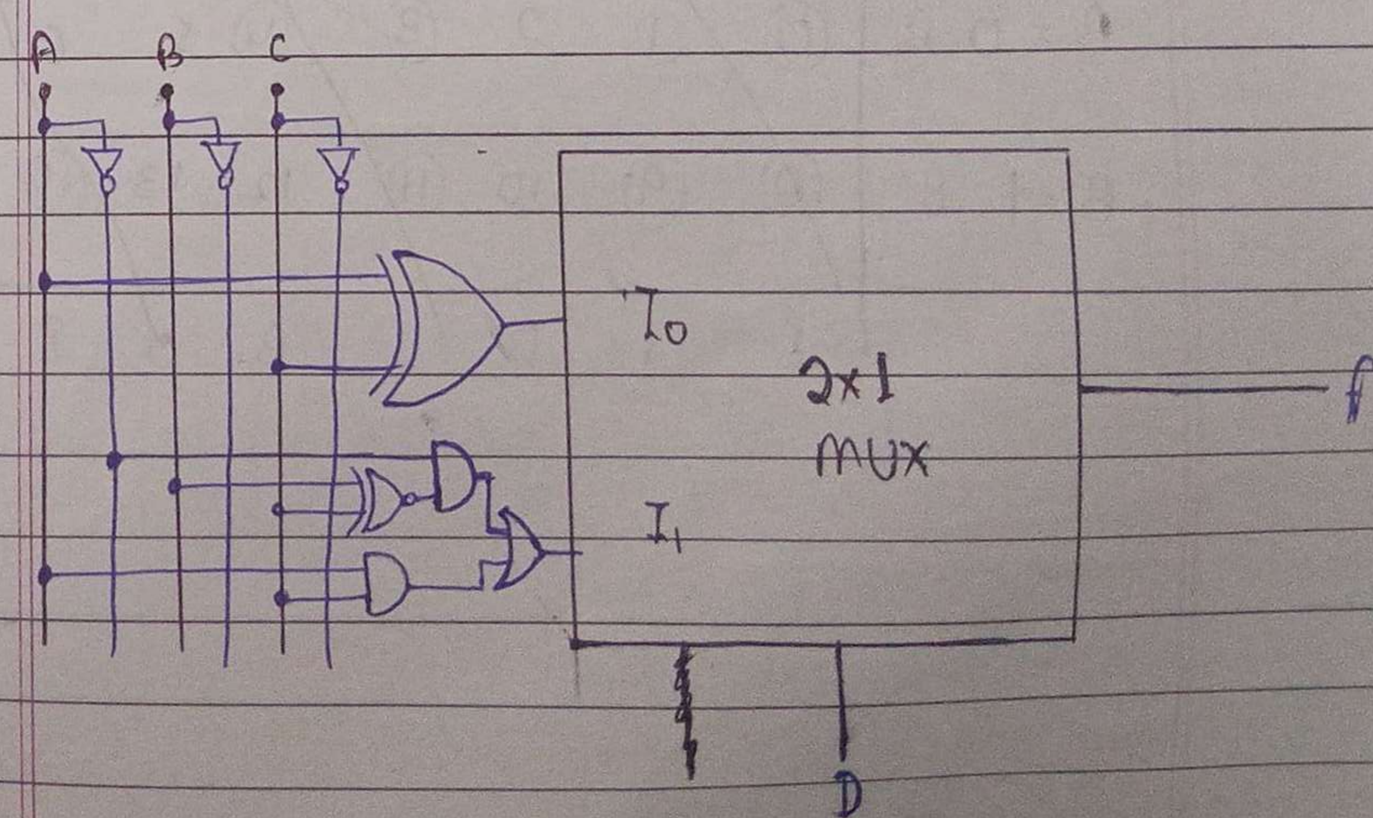
$= \bar{A}C(\bar{B}+B) + A\bar{C}(\bar{B}+B)$

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

$I_1 = \bar{A}\bar{B}\bar{C} + \bar{A}BC + A\bar{B}C + ABC$

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

$I_1 = \bar{A}(B \odot C) + AC$

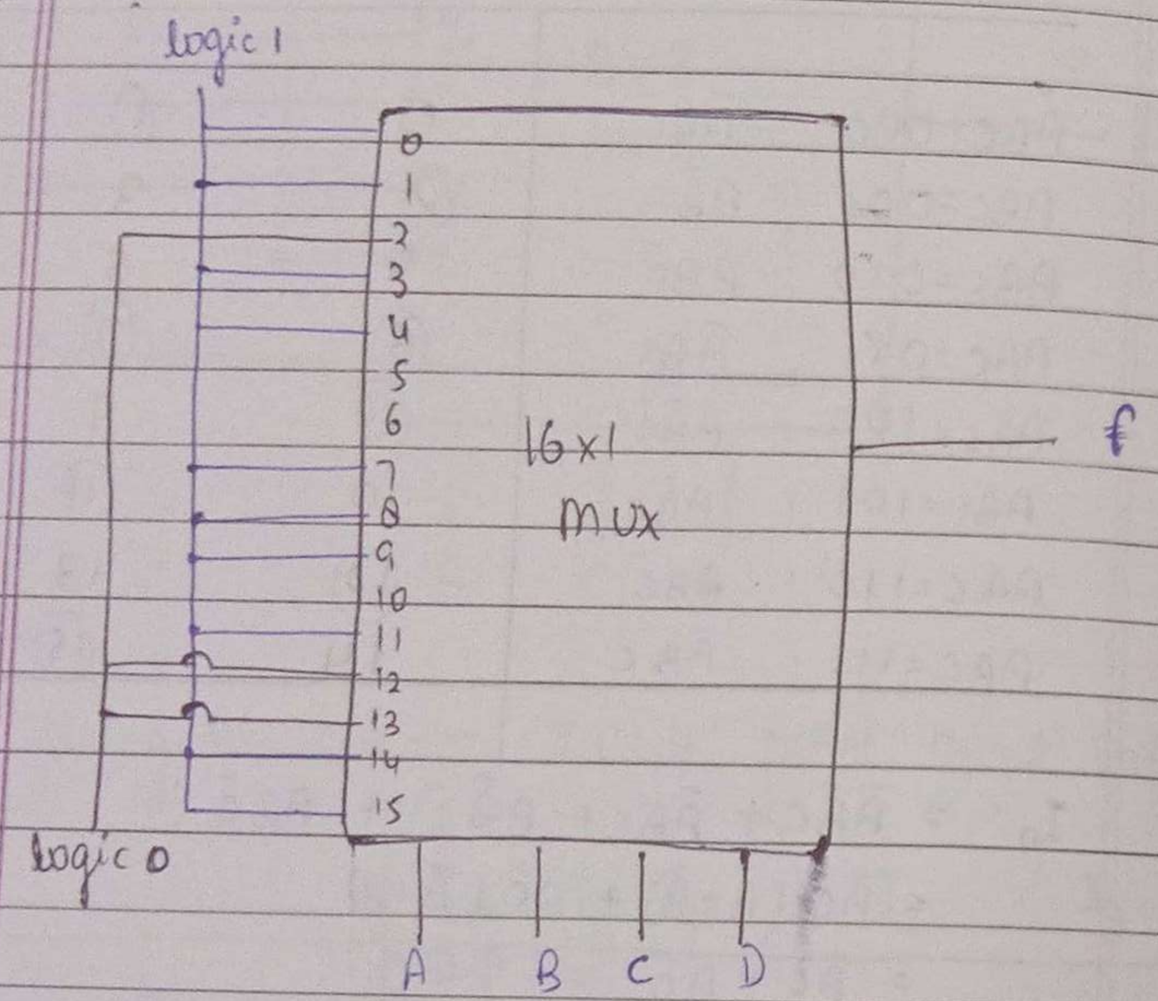


Date
Page

Implement the circuit —

Q $f(A, B, C, D) = \Sigma_m(0, 1, 3, 4, 7, 8, 9, 11, 14, 15)$
Using variable B as input.

Sol: Using 16×1 MUX

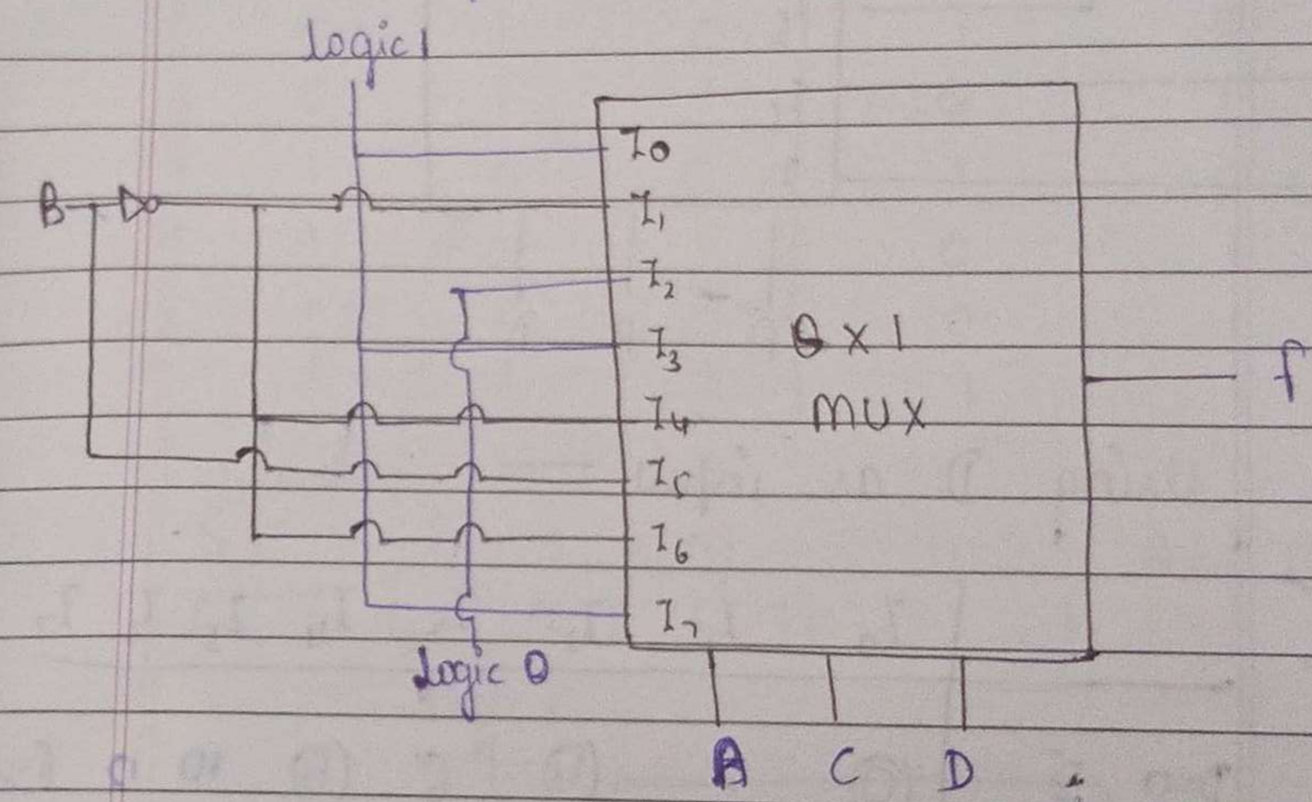


Using 8×1 mux —

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

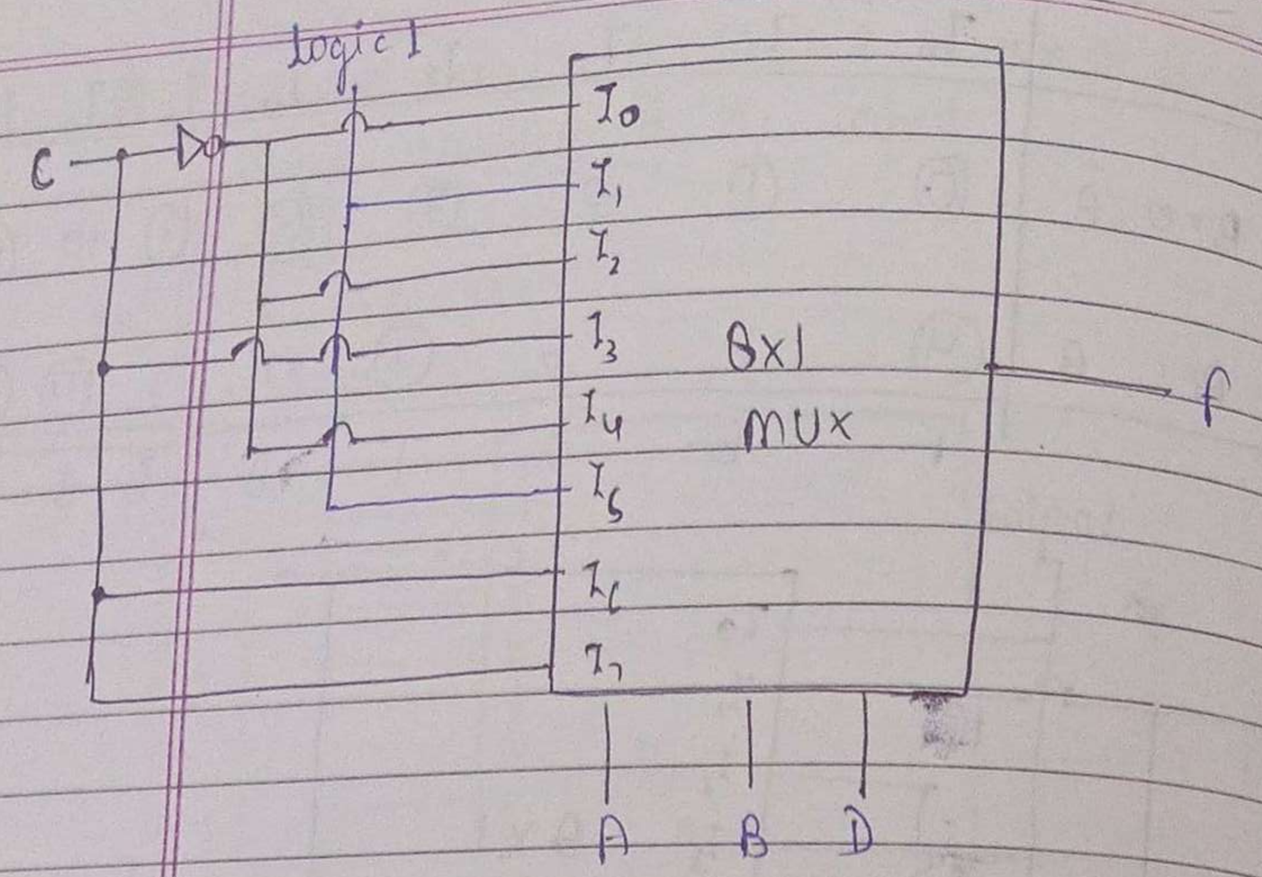
Date
Page

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



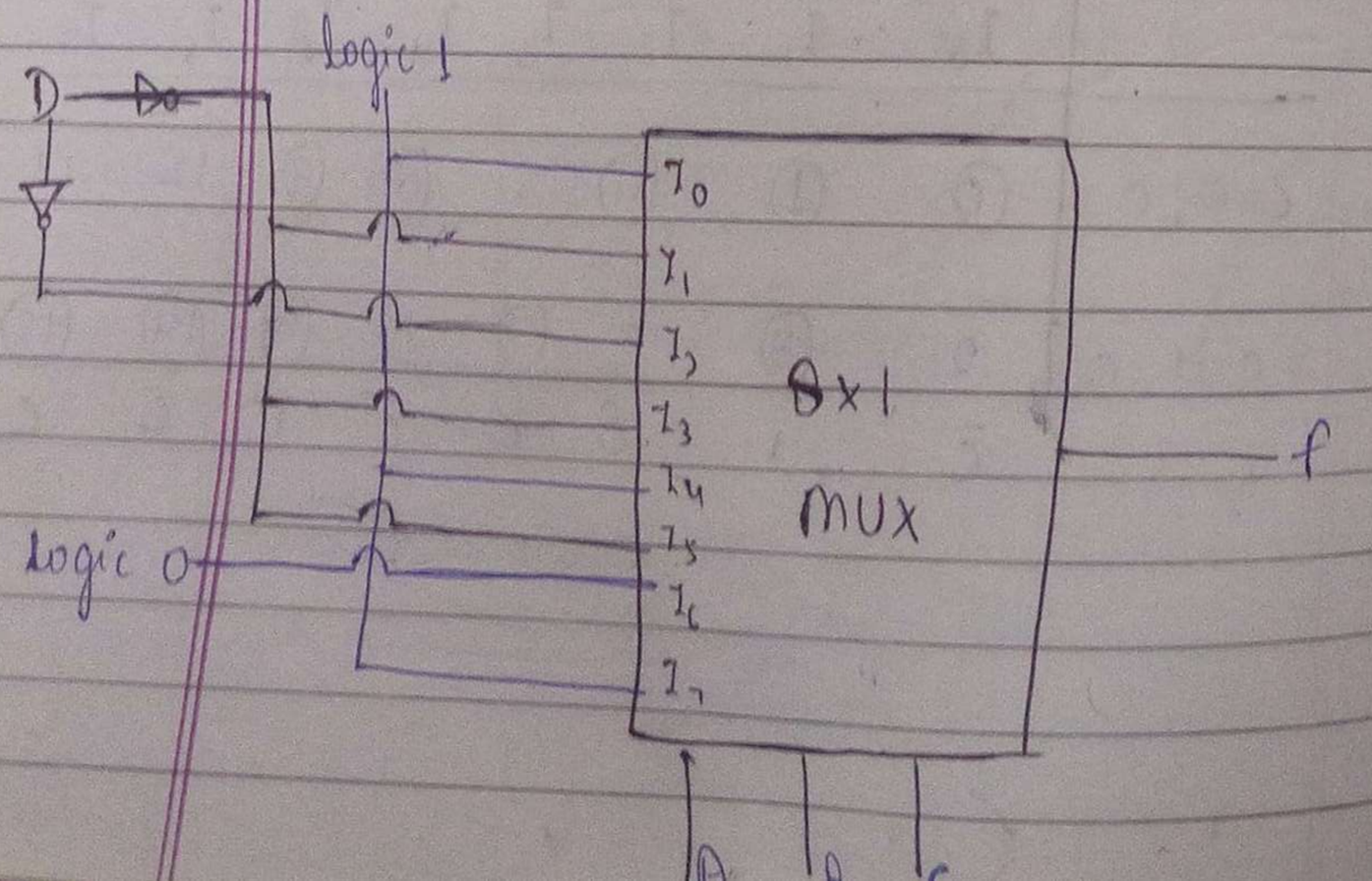
② Using C as input —

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



Using D as input —

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



Date
Page

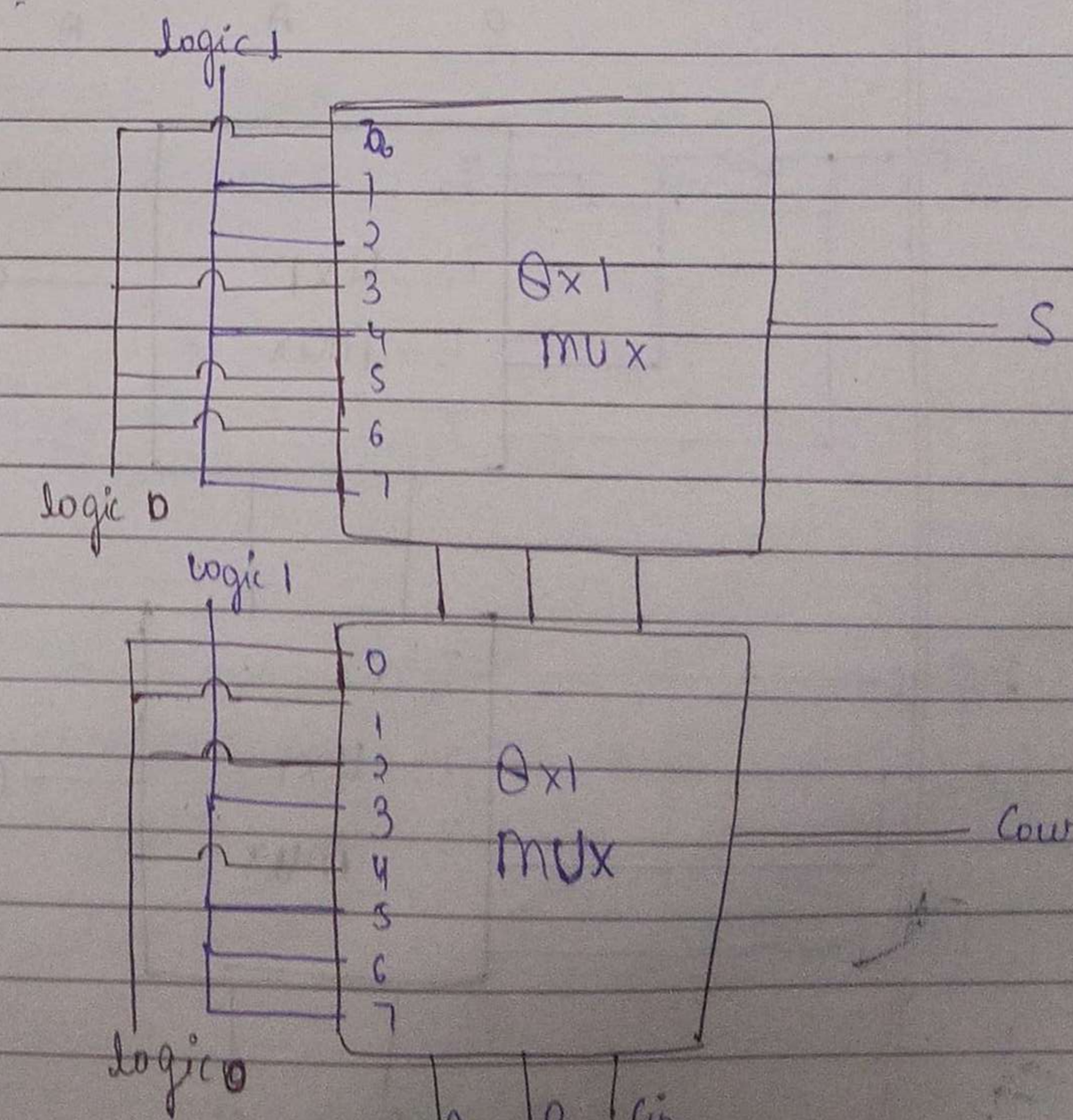
Q Implement Full Adder using Multiplexer —

Sol:

	A	B	C_{in}	S	C_{out}
0	0	0	0	0	0
1	0	0	1	1	0
2	0	1	0	1	0
3	0	1	1	0	1
4	1	0	0	1	0
5	1	0	1	0	1
6	1	1	0	0	1
7	1	1	1	1	1

$S^{(A,B,C_i)} = \Sigma_m(1,2,4,7)$ Using 8x1 MUX

$C_{out}^{(A,B,C_i)} = \Sigma_m(3,5,6,7)$

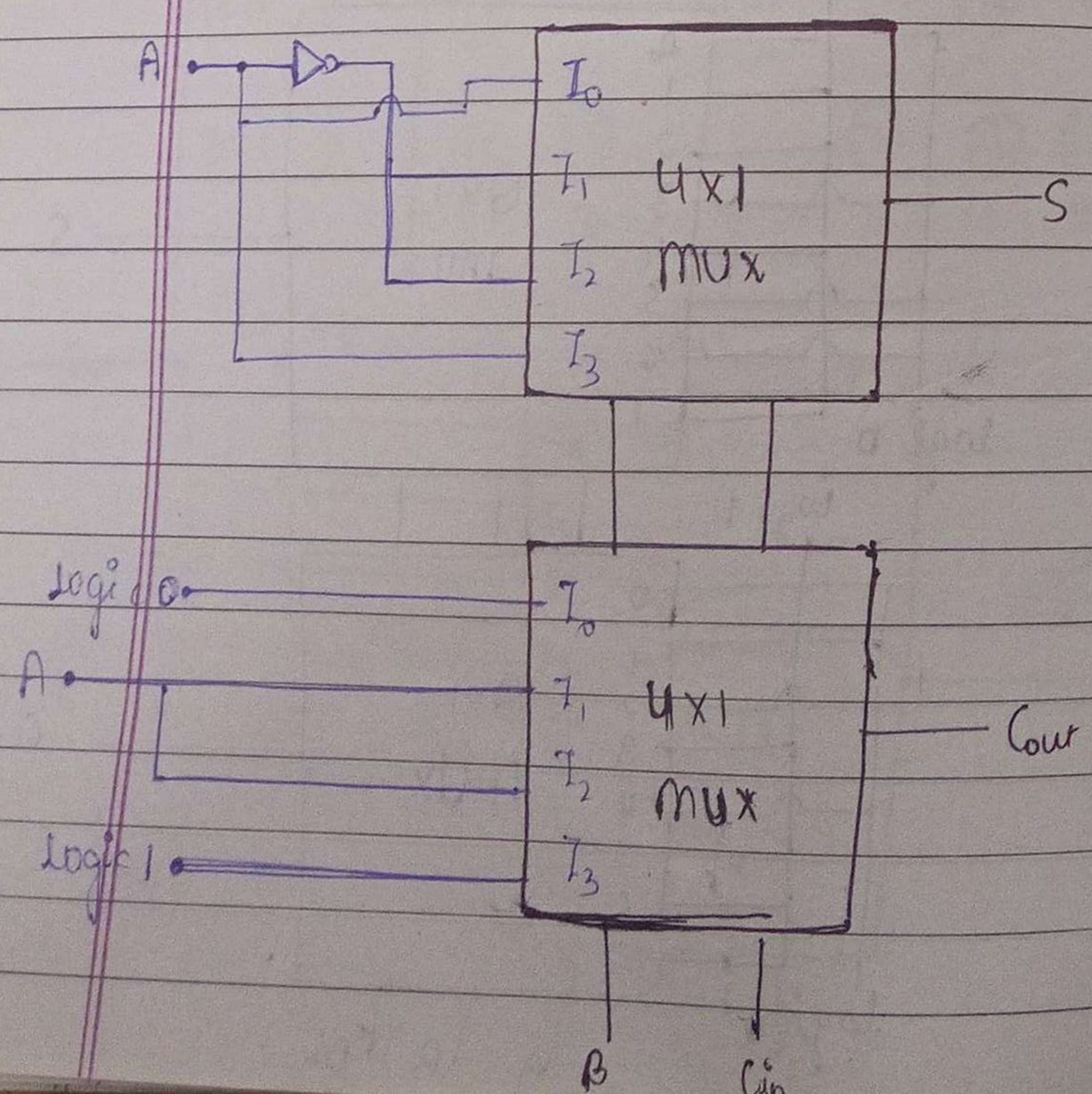


Using 4x1 mux — I/N = 4, O/P = 1, S.L = 2
Variable = 4

Sol: Implementation table —

for S —	I_0	I_1	I_2	I_3
A = 0, $\bar{A}$	0	1	2	3
A = 1, A	4	5	6	7
	$\bar{A}$	$\bar{A}$	$\bar{A}$	A

for C_{in} —	I_0	I_1	I_2	I_3
A = 0, $\bar{A}$	0	1	2	3
A = 1, A	4	5	6	7
	0	A	A	1



Using 2x1 mux — I/N = 2, O/P = 1, S.L = 1
Variable = 4

Sol: Implementation table —

for S —

	I_0	I_1
AB = 00, $\bar{A}\bar{B}$	0	1
AB = 01, $\bar{A}B$	2	3
AB = 10, $A\bar{B}$	4	5
AB = 11, AB	6	7

for C_{in} —

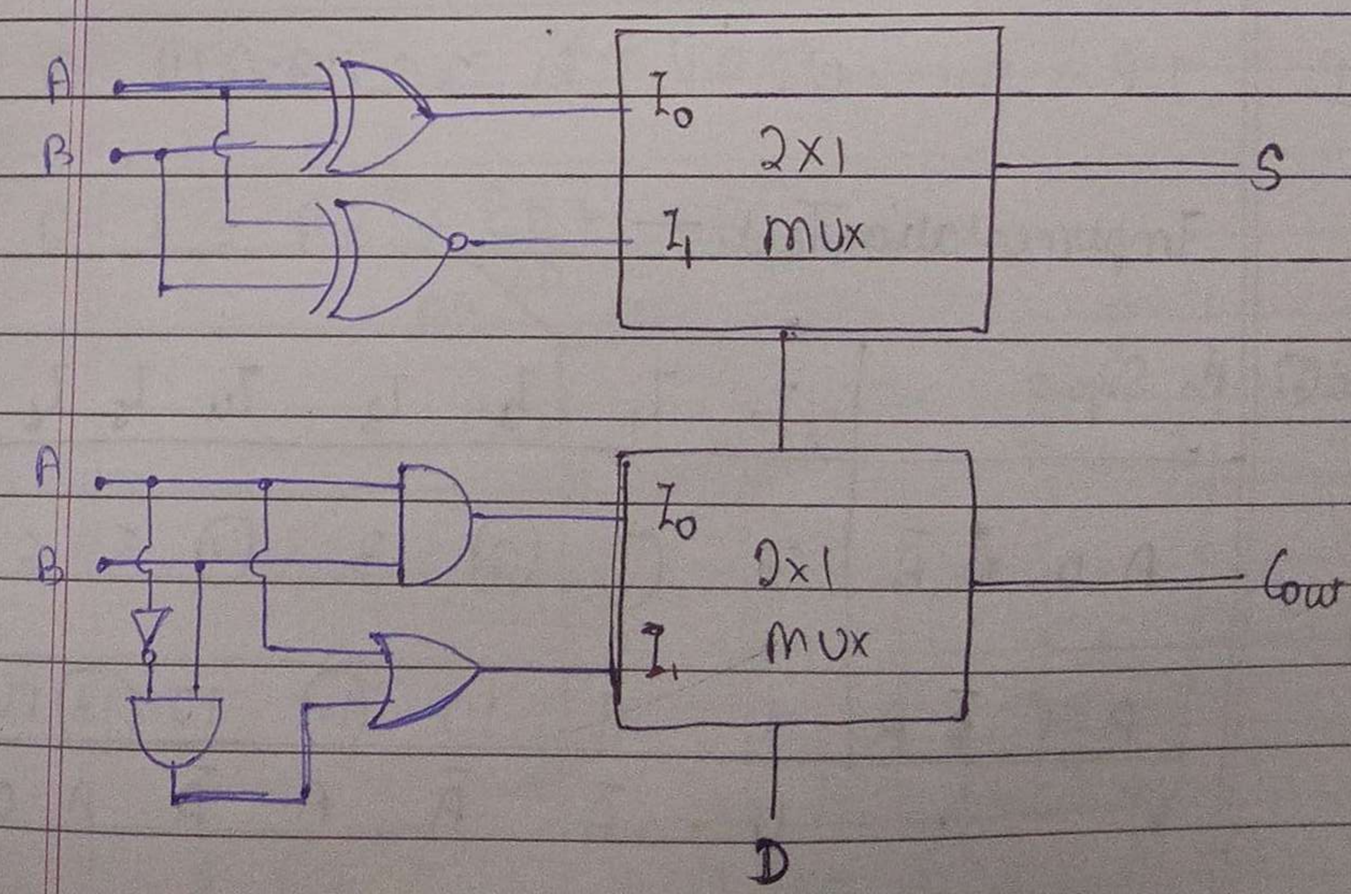
	I_0	I_1
AB = 00, $\bar{A}\bar{B}$	0	1
AB = 01, $\bar{A}B$	2	3
AB = 10, $A\bar{B}$	4	5
AB = 11, AB	6	7

$I_0 = \bar{A}B + A\bar{B} = A \oplus B$

$I_1 = \bar{A}\bar{B} + AB = A \odot B$

$I_0 = AB$

$I_1 = \bar{A}B + A\bar{B} + AB = \bar{A}B + A$



Ques ① Implement full subtractor using 8x1, 4x1 and 2x1 mux.

Ques ② $f(G, H, I, J) = \Sigma_m(1, 2, 4, 5, 6, 9, 10, 11, 15)$ using 16x1 mux, 8x1, 4x1, 2x1 mux
using I, using H, J, using G, I, J

Ques 3 $f(A, B, C, D) = \Sigma_m(0, 1, 3, 4, 8, 9, 15)$ using
16x1 mux
8x1 mux
4x1 mux
2x1

Ques 4 Implement Half Adder using 2x1 mux.
Ques 5 Implement Half subtractor using 2x1 mux.

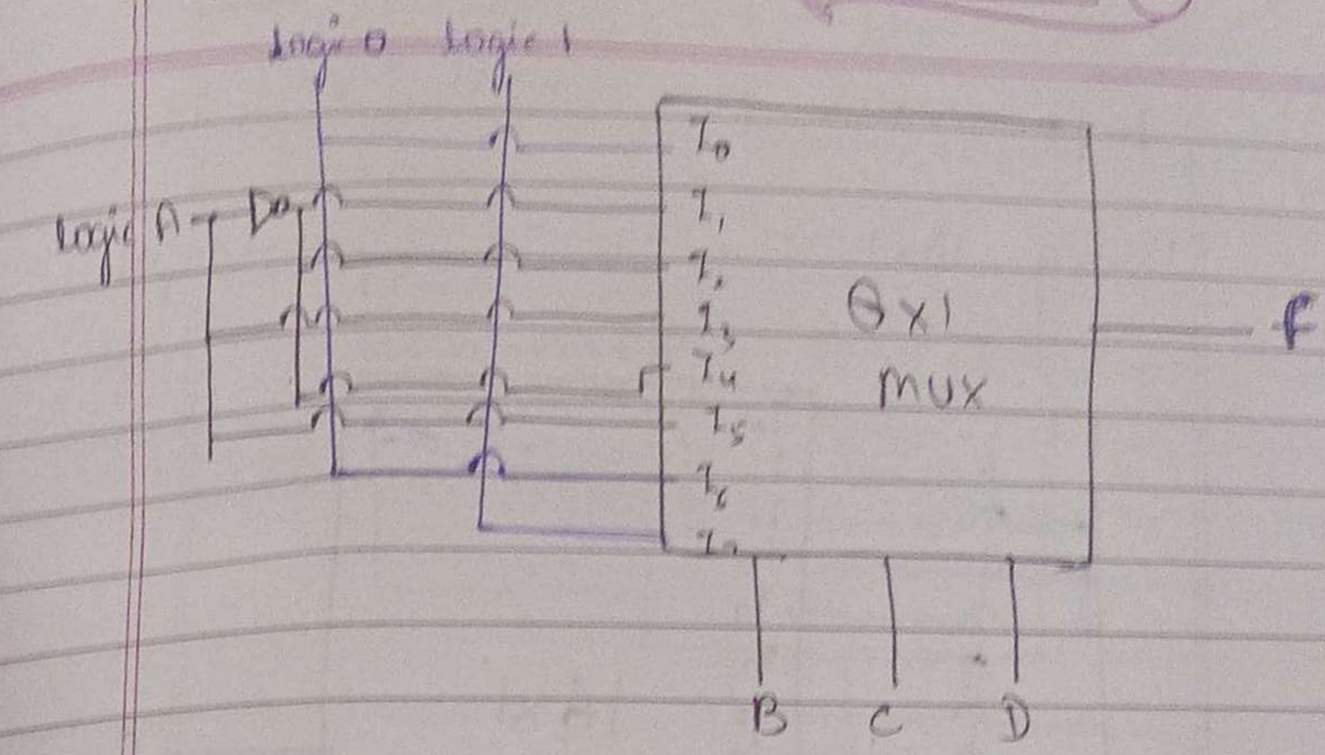
Q $f(A, B, C, D) = \Sigma m(0, 3, 5, 6, 8, 9, 10, 12, 14)$ using 8x1 mux.

Sol: I/P = 8, O/P = 1, SL = 3, Var = 4.

Implementation Table —

Method ①: As Sigma

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



Method ②: Using A

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

Q ~~$f(A, B, C, D) = \Sigma m(0, 3,$~~

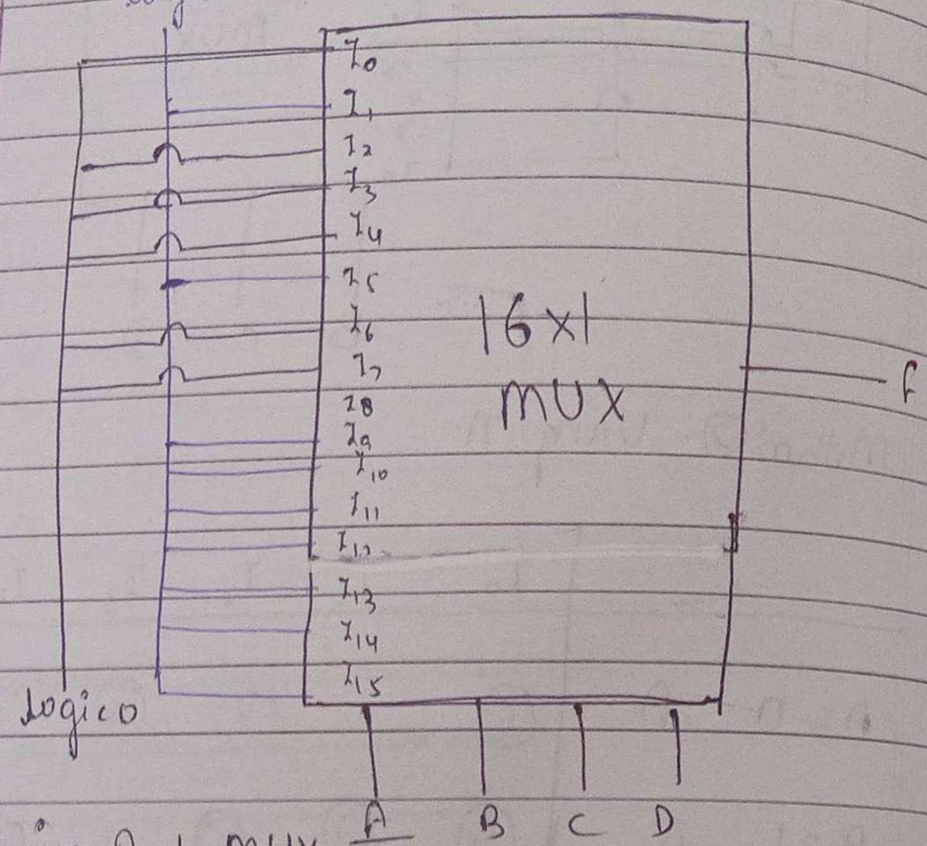
Q: $f = AB + \bar{C}D + A\bar{B}\bar{C}$

AB \ CD	00	01	11	10
00	1 (0)	1 (1)	1 (3)	1 (2)
01	(4)	1 (5)	(7)	(6)
11	1 (12)	1 (13)	1 (15)	1 (14)
10	(8)	1 (9)	1 (11)	1 (10)

$f = \Sigma_m(1, 5, 9, 10, 11, 12, 13, 14, 15)$

~~Implementation I~~

Using 16x1 —

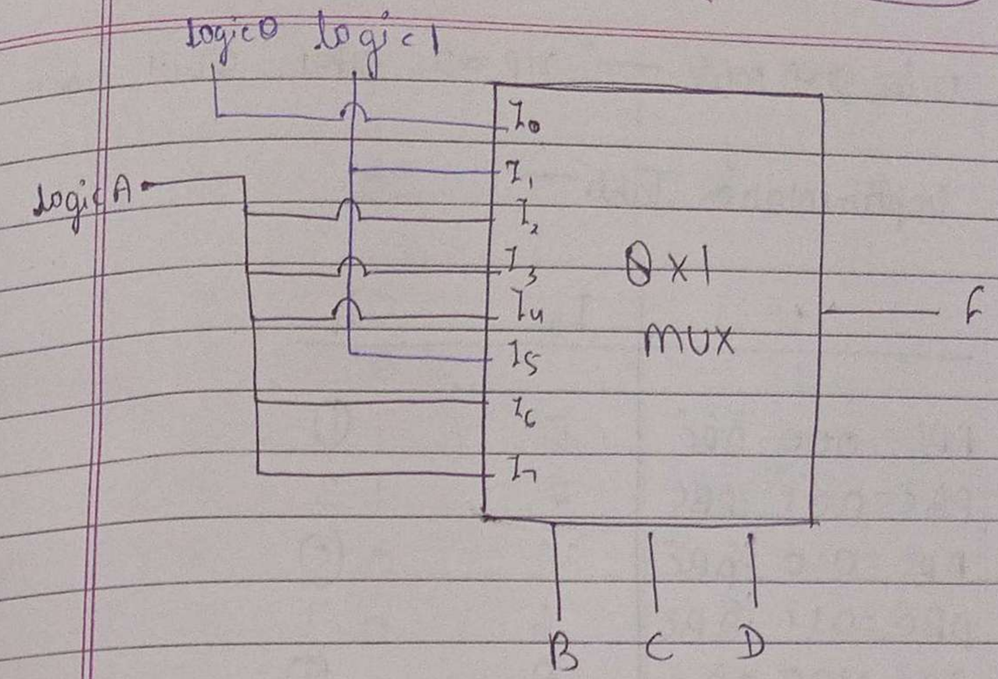


Using 8x1 MUX —

I/P = 8, O/P = 1, SL = 3, Var. = 4

Implementation Table —

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



Using ~~16x~~ 4x1 MUX —

Implementation Table — I/P = 4, O/P = 1, SL = 2, Var = 4

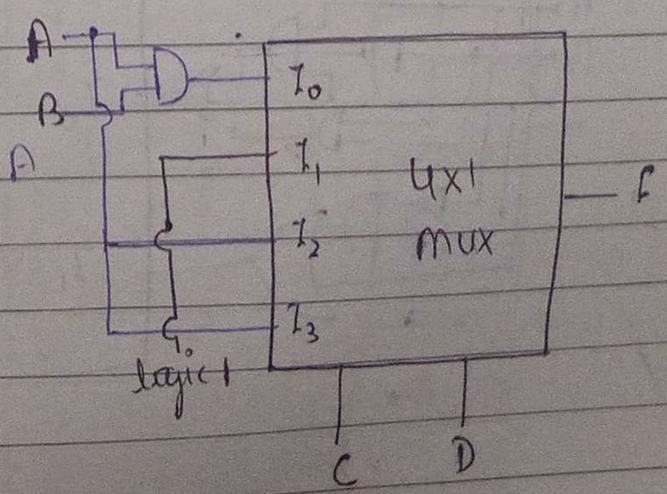
	I_0	I_1	I_2	I_3
AB = 00, $\bar{A}\bar{B}$	0	1	2	3
AB = 01 $\bar{A}B$	4	5	6	7
AB = 10 $A\bar{B}$	8	9	10	11
AB = 11 AB	12	13	14	15

$I_0 = AB$

$I_1 = 1$

$I_2 = A\bar{B} + AB = A$

$I_3 = A$



Using 2x1 MUX — I/P = 2 O/P = 1, S.L = 1, Var = 4

Implimentation Table —

	I_0	I_1
$ABC = 000$, $\bar{A}\bar{B}\bar{C}$	0	1
$ABC = 001$, $\bar{A}\bar{B}C$	2	3
$ABC = 010$, $\bar{A}B\bar{C}$	4	5
$ABC = 011$, $\bar{A}BC$	6	7
$ABC = 100$, $A\bar{B}\bar{C}$	8	9
$ABC = 101$, $A\bar{B}C$	10	11
$ABC = 110$, $AB\bar{C}$	12	13
$ABC = 111$, ABC	14	15

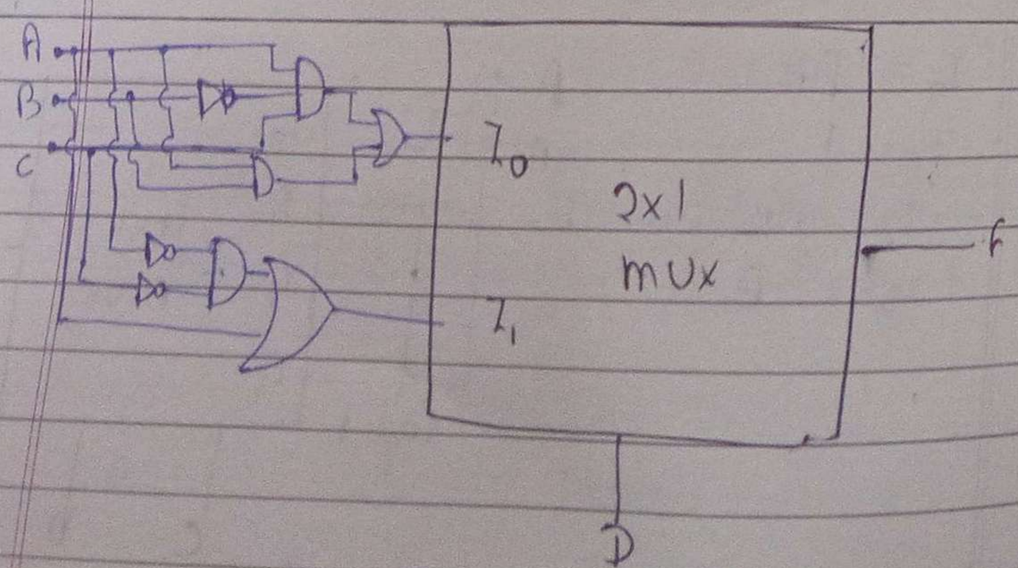
$I_0 = A\bar{B}C + A\bar{B}\bar{C} + ABC = A\bar{B}C + AB$

$I_1 = \bar{A}\bar{B}\bar{C} + \bar{A}B\bar{C} + A\bar{B}\bar{C} + A\bar{B}C + AB\bar{C} + ABC$

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

$I_1 = \bar{A}\bar{C} + A\bar{C} + AC$

$I_1 = \bar{A}\bar{C} + A$



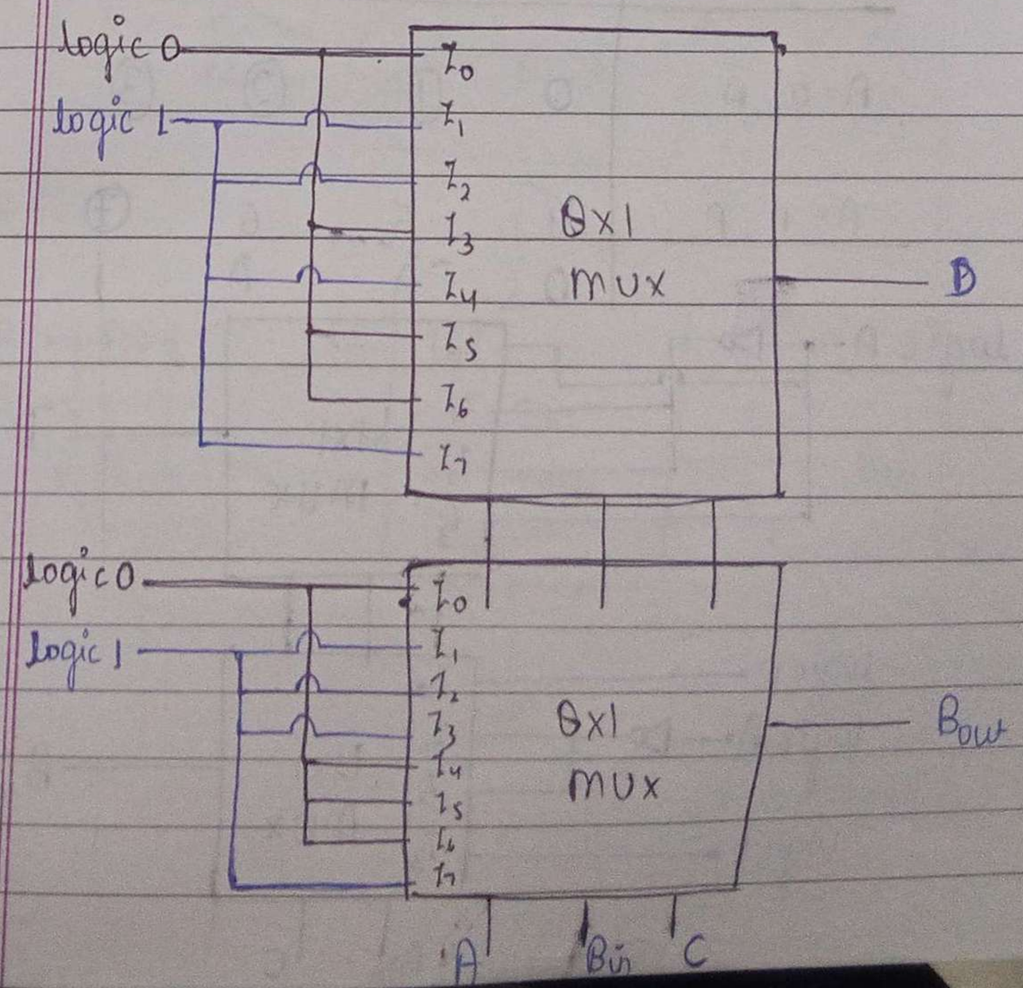
Revision

① Full Subtractor using 8x1 MUX —

A	B_{in}	C	D	B_{ow}
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

(i) Using 8x1 MUX — I/N = 8, O/P = 1

S.L = 3, Variables = 3



A
0
0
0
0
0
0
0
1
1
1
1
1
1
1

(ii) Using 4x1 mux — I/P = 4, S.L = 2
O/P = 1, Variables = 3

Using A as input —

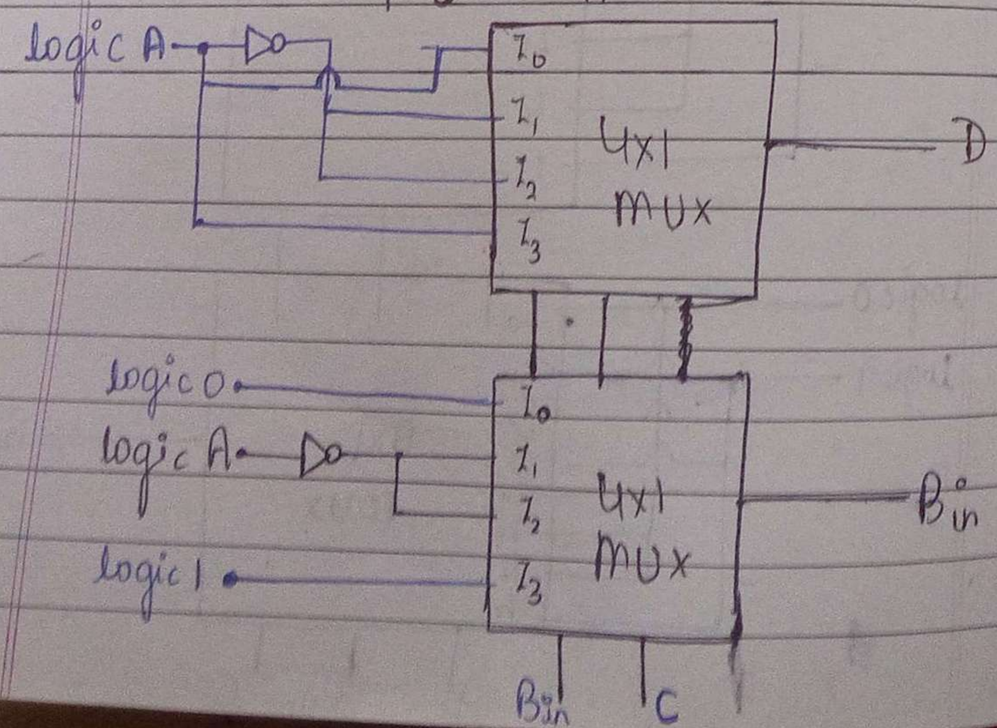
Implementation table —

for ~~B_{in}~~ D —

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

for B_{in} —

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



(iii) Using 2x1 — I/P = 2, S.L = 1
O/P = 1, Variables = 3

for D

	I_0	I_1
$AB_{in} = 00, \bar{A}\bar{B}_{in}$	0	1
$AB_{in} = 01, \bar{A}B_{in}$	2	3
$AB_{in} = 10, A\bar{B}_{in}$	4	5
$AB_{in} = 11, AB_{in}$	6	7

$I_0 = \bar{A}B_{in} + A\bar{B}_{in} = A \oplus B_{in}$

$I_1 = \bar{A}\bar{B}_{in} + AB_{in} = A \odot B_{in}$

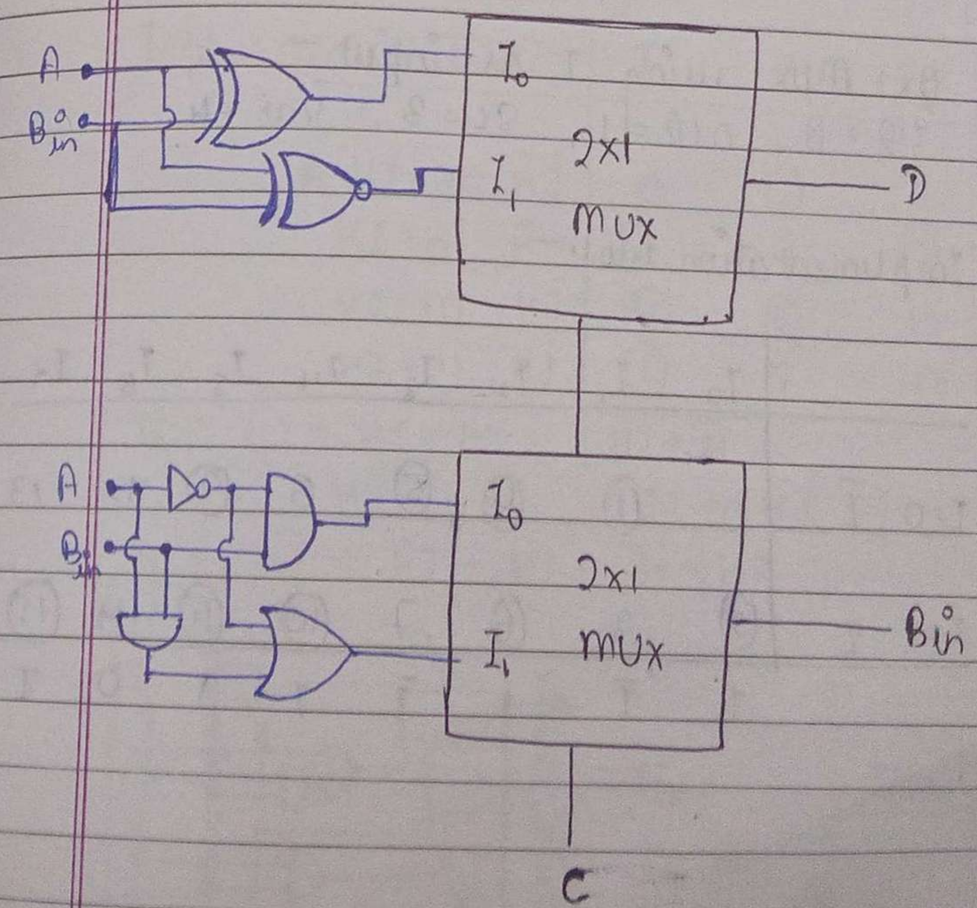
for B_{in} —

I_0	I_1
0	1
2	3
4	5
6	7

$I_0 = \bar{A}B_{in}$

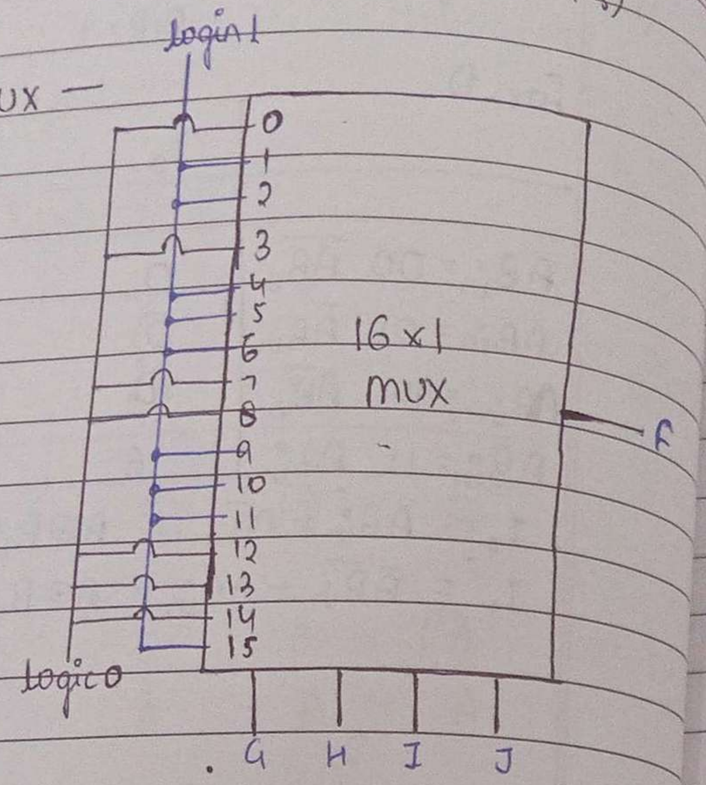
$I_1 = \bar{A}\bar{B}_{in} + \bar{A}B_{in} + AB_{in}$

$I_1 = \bar{A} + AB_{in}$



② $f(G,H,I,J) = \Sigma_m(1,2,4,5,6,9,10,11,15)$

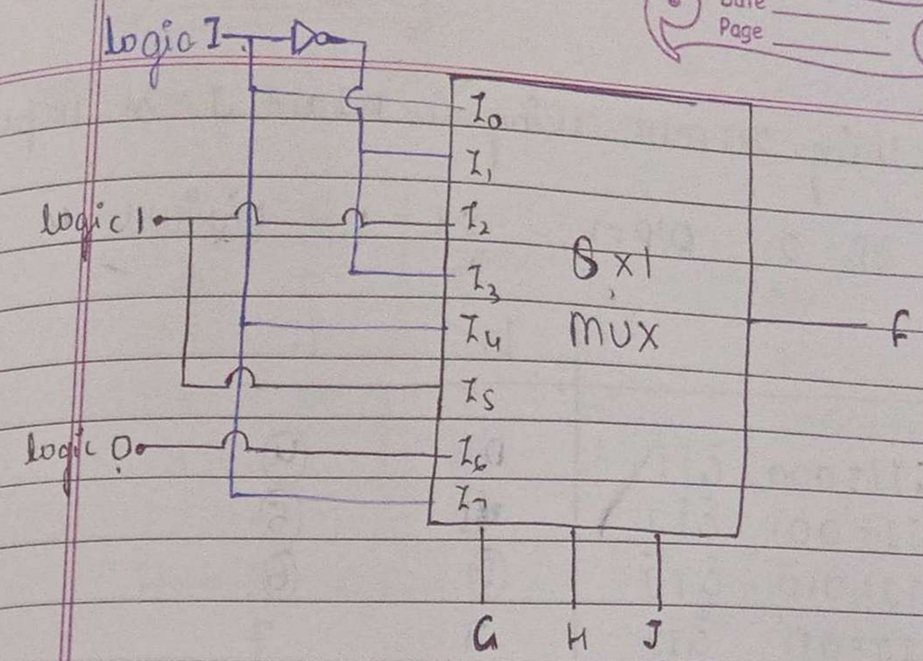
(i) Using 16×1 mux —



(ii) 8×1 mux using I as input —
I/P = 8, O/P = 1, SL = 3, Var = 4

Implementation table —

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



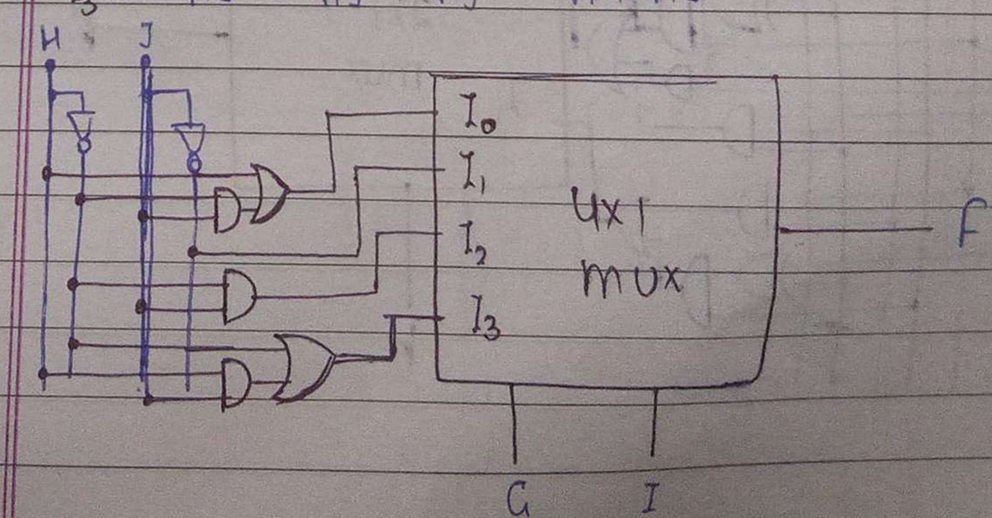
(iii) 4×1 mux using H and J at input —
I/P = 4, O/P = 1, SL = 2, Variable = 4

Implementation Table —	I_0	I_1	I_2	I_3
$HJ=00, \bar{H}\bar{J}$	0	2	8	10
$HJ=01, \bar{H}J$	1	3	9	11
$HJ=10, H\bar{J}$	4	6	12	14
$HJ=11, HJ$	5	7	13	15

$I_0 = \bar{H}\bar{J} + H\bar{J} + HJ = \bar{H}\bar{J} + H$

$I_1 = \bar{H}\bar{J} + H\bar{J} = \bar{J}$, $I_2 = \bar{H}J$

$I_3 = \bar{H}\bar{J} + \bar{H}J + HJ = \bar{H} + HJ$



(iv) Using 2x1 mux using G, I and J at input -

Sol: I/P = 2, O/P = 1, S.L = 1, Variable = 4

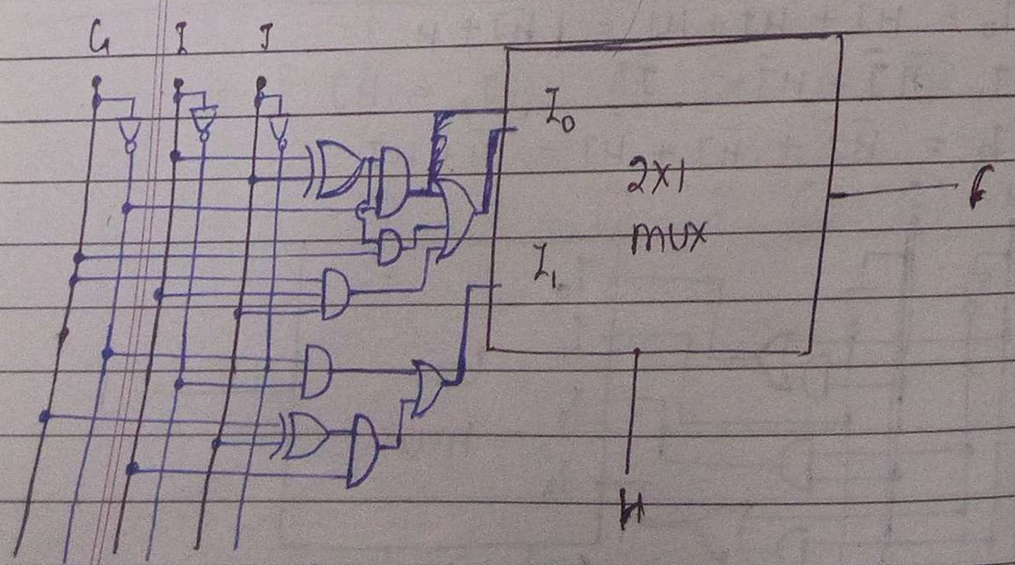
	I_0	I_1
$GIJ = 000$, $\bar{G}\bar{I}\bar{J}$	0	4
$GIJ = 001$, $\bar{G}\bar{I}J$	1	5
$GIJ = 010$, $\bar{G}I\bar{J}$	2	6
$GIJ = 011$, $\bar{G}IJ$	3	7
$GIJ = 100$, $G\bar{I}\bar{J}$	8	12
$GIJ = 101$, $G\bar{I}J$	9	13
$GIJ = 110$, $GI\bar{J}$	10	14
$GIJ = 111$, GIJ	11	15

$I_0 = \bar{G}\bar{I}\bar{J} + \bar{G}I\bar{J} + G\bar{I}J + GI\bar{J} + GIJ$

$I_0 = \bar{G}(I \oplus J) + G(I \oplus J) + GIJ$

$I_1 = \bar{G}\bar{I}\bar{J} + \bar{G}\bar{I}J + \bar{G}I\bar{J} + GIJ$

$I_1 = \bar{G}\bar{I} + I(G \oplus J)$



(3) Half Adder using 2x1 mux —

A	B	S	C
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

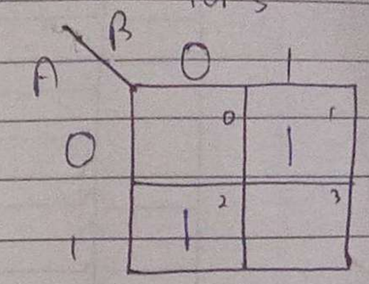
I/P = 2
O/P = 1
S.L = 1
Variable = 2

~~I/N = 2, O/P = 1.~~

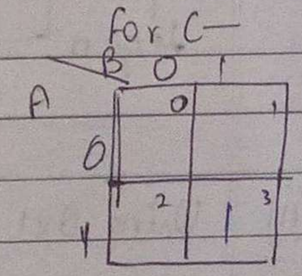
$S = \Sigma_m(1, 2)$

$C = \Sigma_m(3)$

for S —



for C —

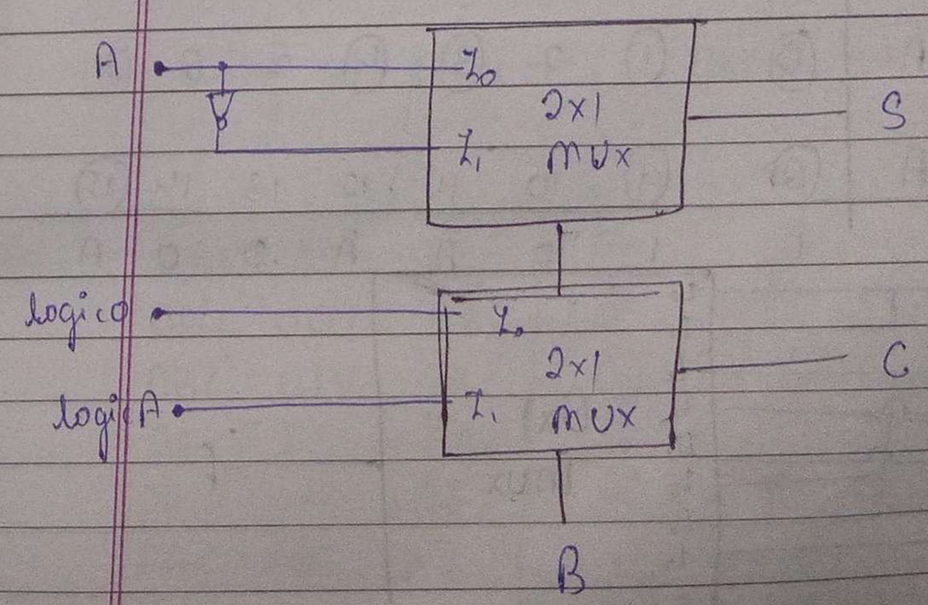


Implementation Table —

for S —	I_0	I_1		I_0	I_1
$A = 0, \bar{A}$	0	1		0	1
$A = 1, A$	2	3		2	3

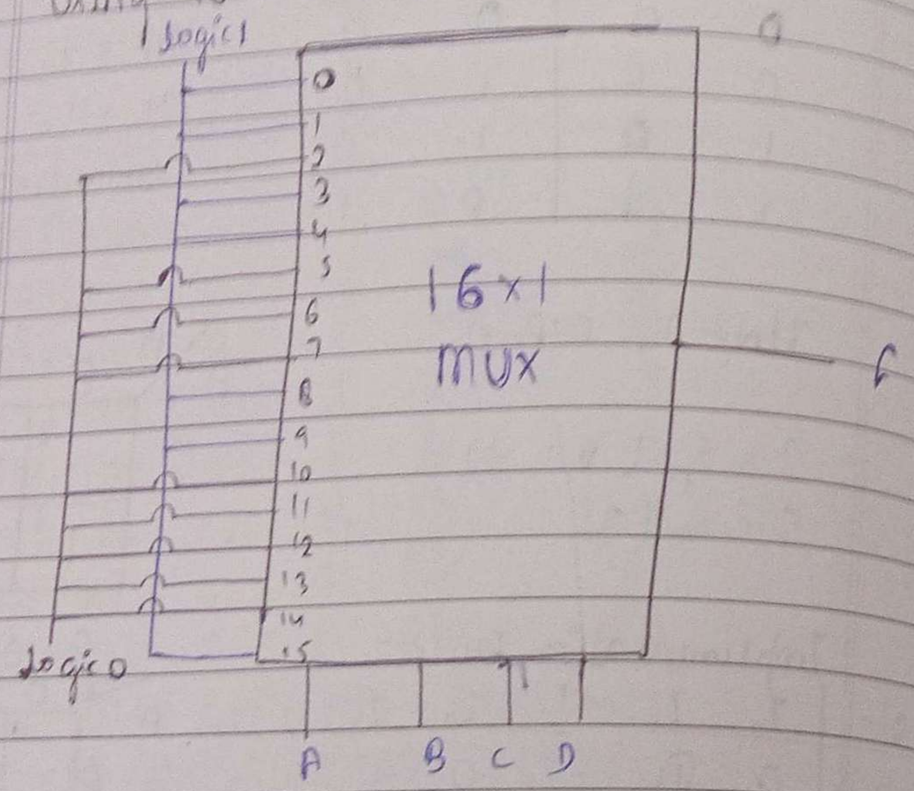
~~$S = A + \bar{A}$~~

$I_0 = A$, $I_1 = \bar{A}$ | $I_0 = 0$, $I_1 = A$



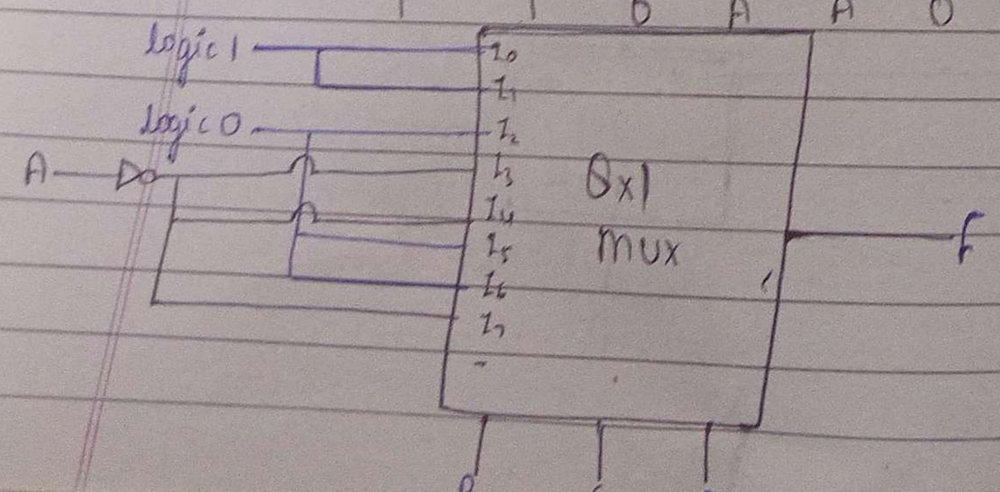
(4) $f(A,B,C,D) = \Sigma_m(0,1,3,4,8,9,15)$

(i) Using 16×1 mux —



(ii) Using 8×1 mux — I/P = 8, O/P = 1, S.L = 3, Var = 4

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



(iii) Using 4×1 mux — I/P = 4, O/P = 1, S.L = 2, Var = 4

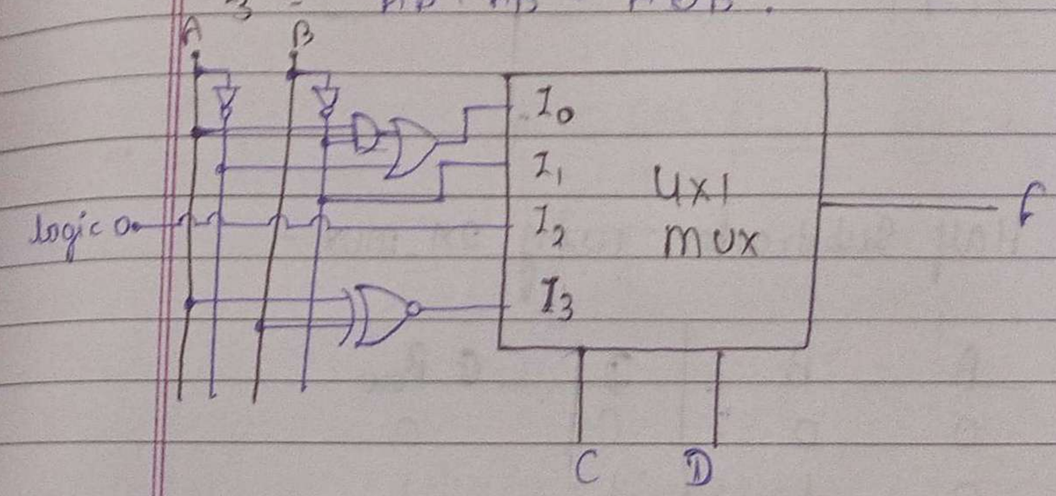
	I_0	I_1	I_2	I_3
$AB = 00$, $\bar{A}\bar{B}$	0	1	2	3
$AB = 01$, $\bar{A}B$	4	5	6	7
$AB = 10$, $A\bar{B}$	8	9	10	11
$AB = 11$, AB	12	13	14	15

$I_0 = \bar{A}\bar{B} + \bar{A}B + A\bar{B} = \bar{A} + A\bar{B}$

$I_1 = \bar{A}B + A\bar{B} = \bar{B}$

$I_2 = 0$

$I_3 = \bar{A}\bar{B} + AB = A \odot B$.

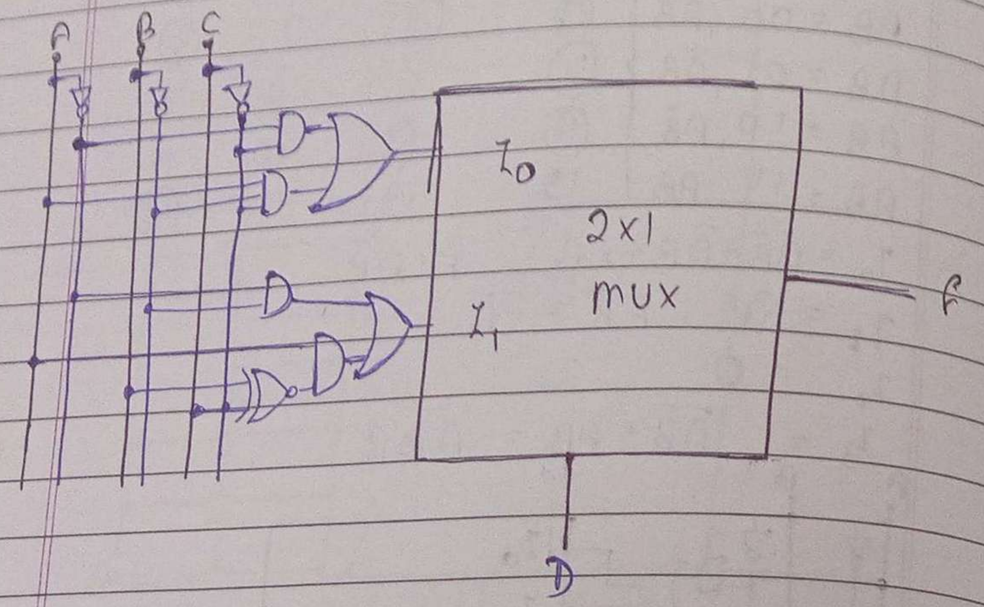


(iv) Using 2×1 mux — I/N = 2, O/P = 1, S.L = 1,

	I_0	I_1
$ABC = 000$, $\bar{A}\bar{B}\bar{C}$	0	1
$ABC = 001$, $\bar{A}\bar{B}C$	2	3
$ABC = 010$, $\bar{A}B\bar{C}$	4	5
$ABC = 011$, $\bar{A}BC$	6	7
$ABC = 100$, $A\bar{B}\bar{C}$	8	9
$ABC = 101$, $A\bar{B}C$	10	11
$ABC = 110$, $AB\bar{C}$	12	13
$ABC = 111$, ABC	14	15

$I_0 = \bar{A}\bar{B}\bar{C} + \bar{A}B\bar{C} + A\bar{B}\bar{C}$
$= \bar{A}\bar{C} + A\bar{B}\bar{C}$

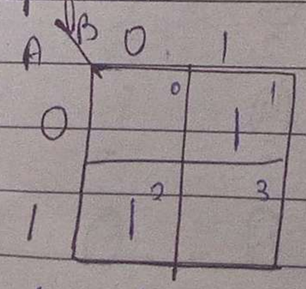
$I_1 = \bar{A}\bar{B}C + \bar{A}BC + A\bar{B}\bar{C} + ABC.$
$= \bar{A}\bar{B} + A(B \odot C).$



⑤ Half Subtractor using 2x1 mux -

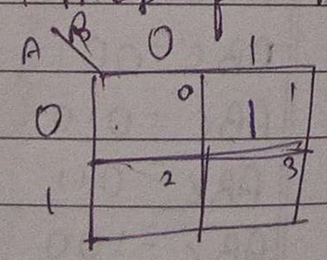
A	B	D	~~C~~ B_{ow}
0	0	0	0
0	1	1	1
1	0	1	0
1	1	0	0

K-map for D —



$D = \Sigma_m(1, 2)$

K-map for B_{ow} -

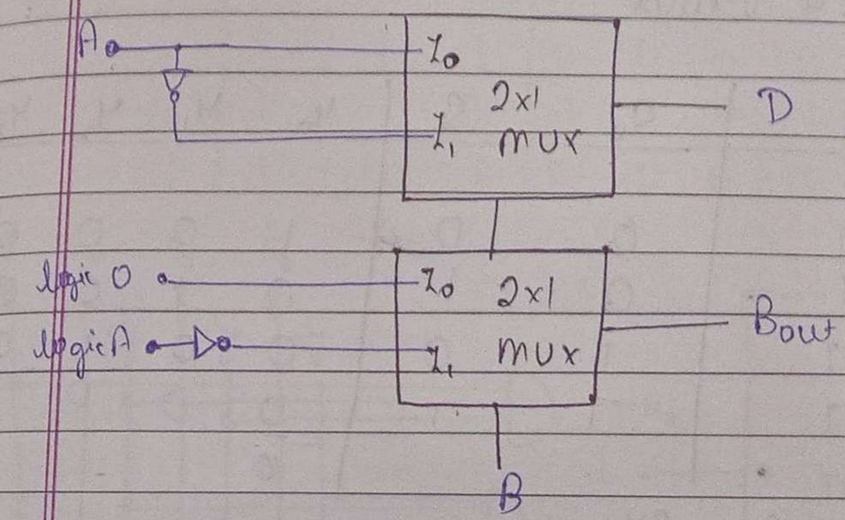


$B_{ow} = \Sigma_m(1)$

Implementation Table —

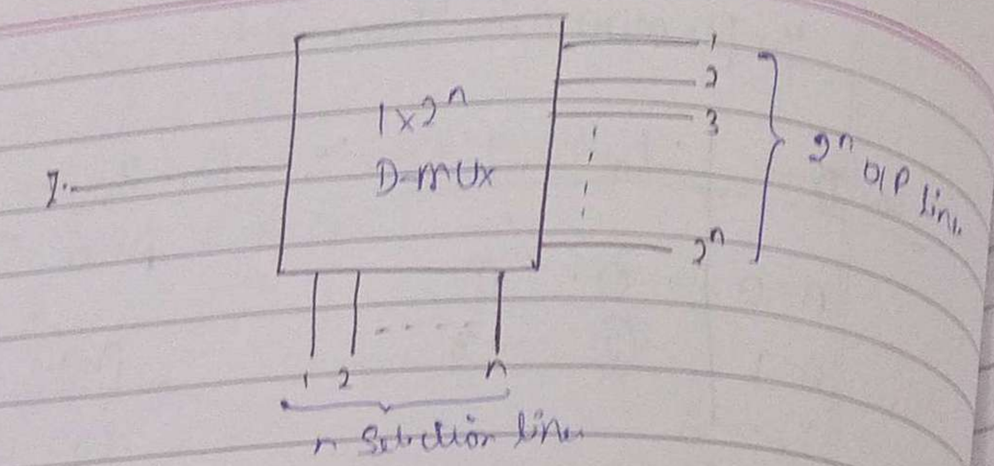
for D —	I_0	I_1
A = 0	0	①
A = 1	②	3
$I_0 = A$		
$I_1 = \bar{A}$		

for B_{ow} —	I_0	I_1
A = 0	0	①
A = 1	2	3
$I_0 = 0$	or	
$I_1 = \bar{A}$		



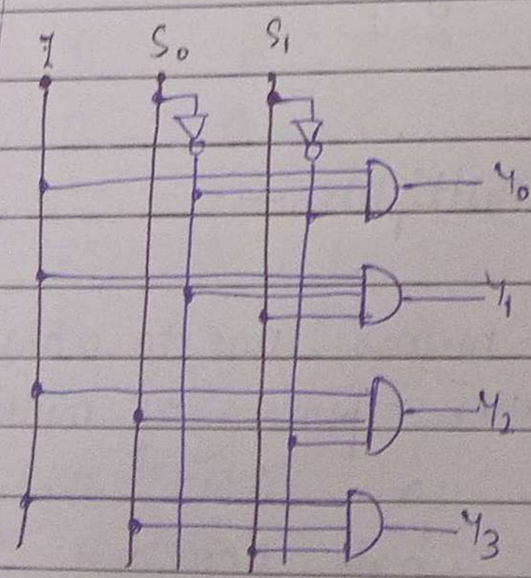
De-multiplexer —

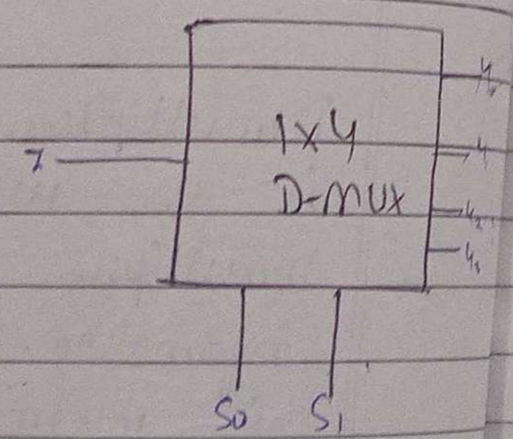
Demultiplexer means one to many. It is the reverse process of multiplexer. Demultiplexer is a circuit that require information of single line and transmit this information on one of the 2^n possible O/P lines. The selection of specific O/P line is control by the value of n selection lines.



① 1×4 D-mux —

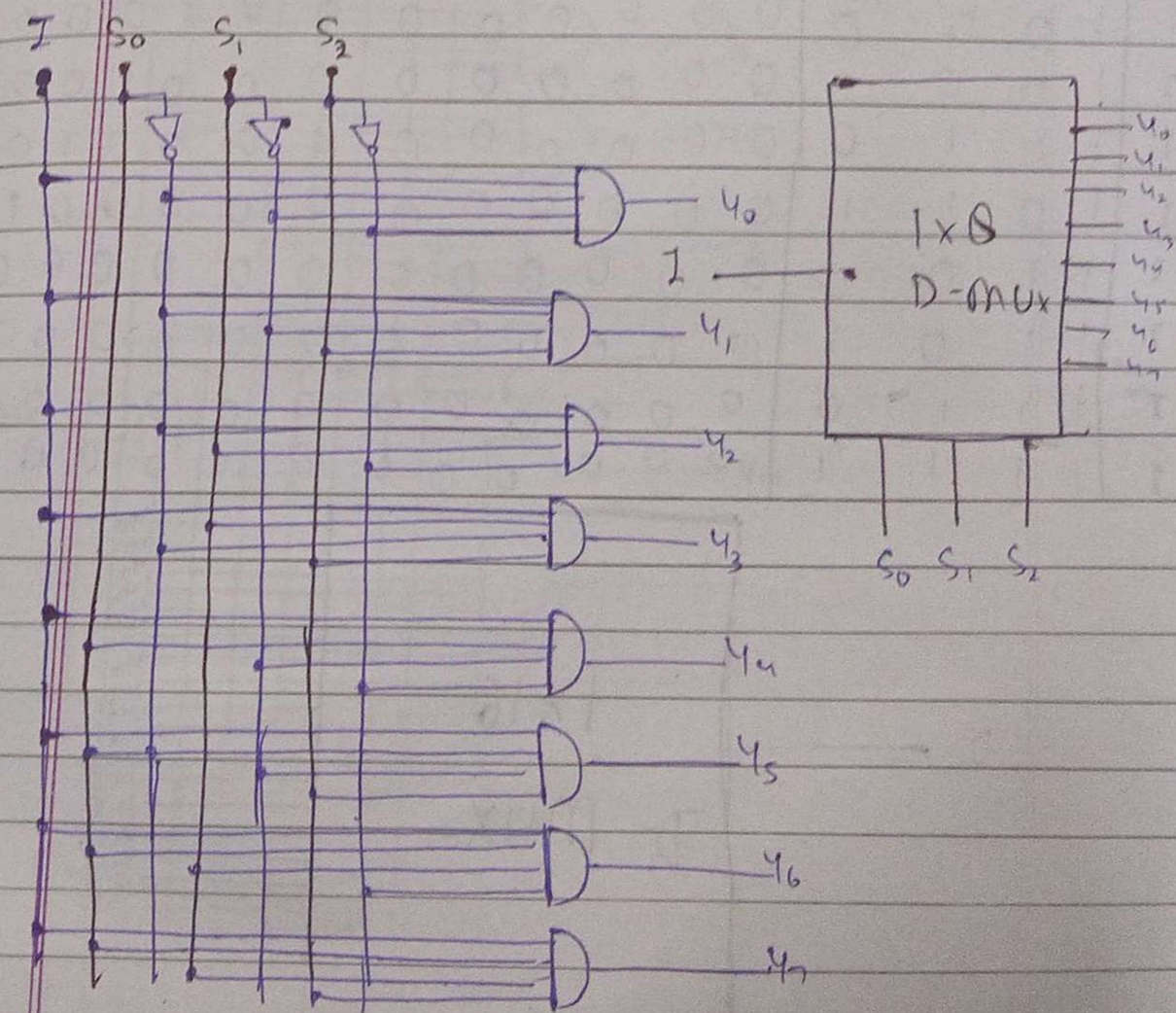
I	S_0	S_1	Y_0	Y_1	Y_2	Y_3
I	0	0	1	0	0	0
I	0	1	0	1	0	0
I	1	0	0	0	1	0
I	1	1	0	0	0	1





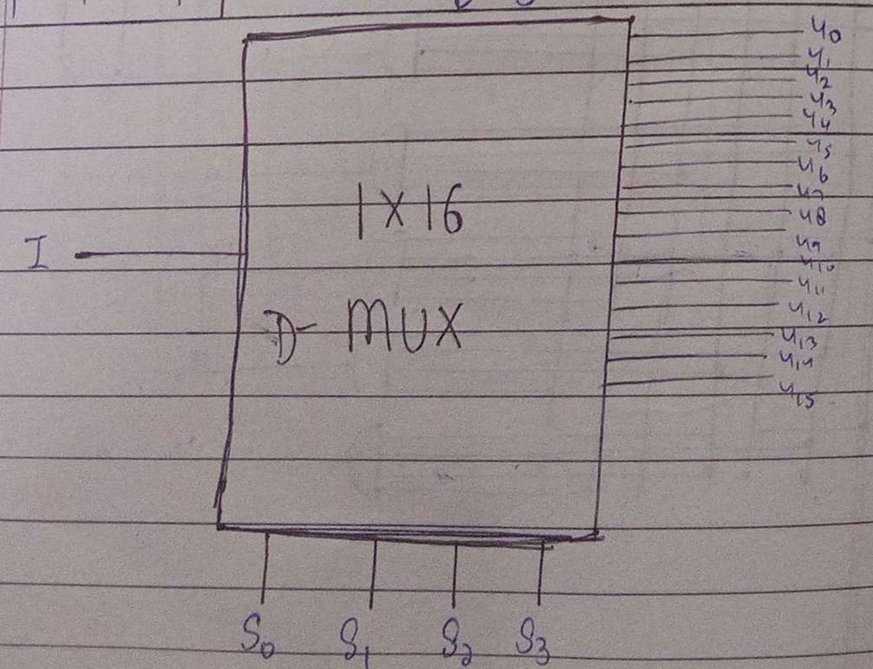
② 1×8 De-mux —

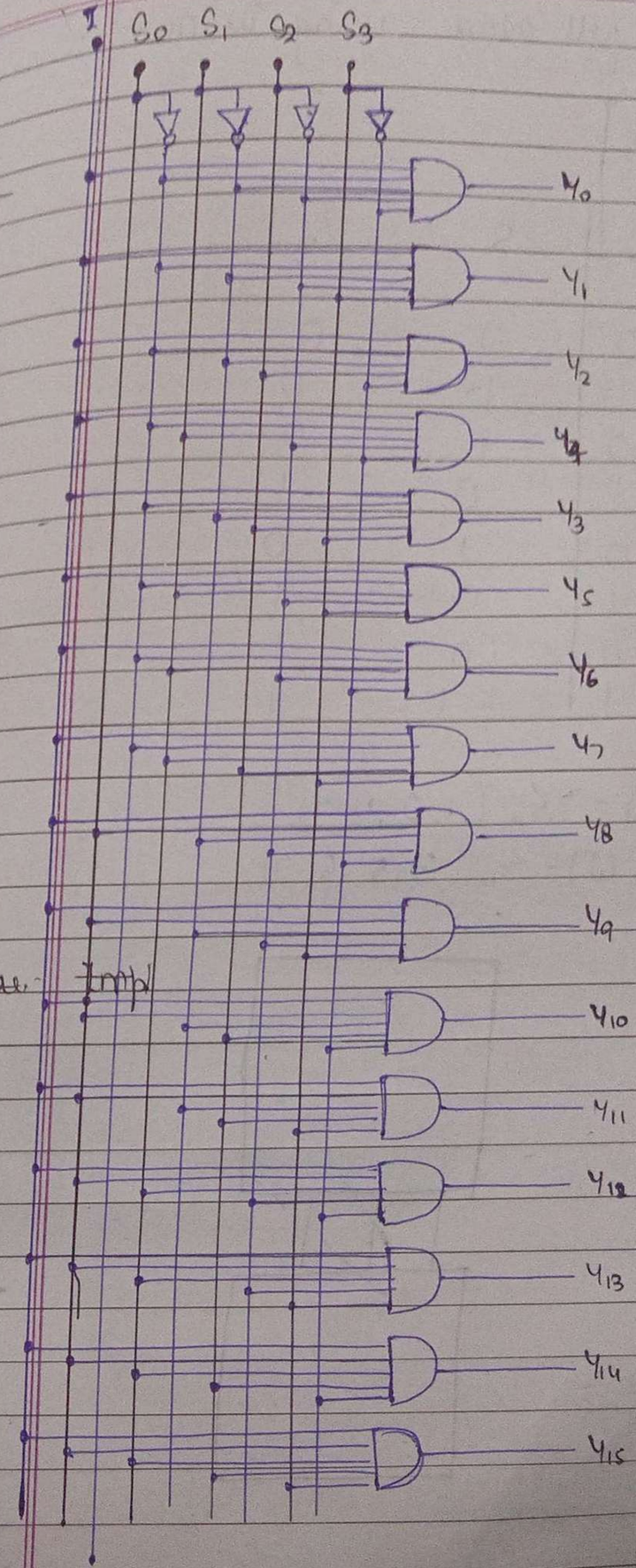
I	S_0	S_1	S_2	Y_0	Y_1	Y_2	Y_3	Y_4	Y_5	Y_6	Y_7
I	0	0	0	1	0	0	0	0	0	0	0
I	0	0	1	0	1	0	0	0	0	0	0
I	0	1	0	0	0	1	0	0	0	0	0
I	0	1	1	0	0	0	1	0	0	0	0
I	1	0	0	0	0	0	0	1	0	0	0
I	1	0	1	0	0	0	0	0	1	0	0
I	1	1	0	0	0	0	0	0	0	1	0
I	1	1	1	0	0	0	0	0	0	0	1



③ 1x16 D-mux

I	S_0	S_1	S_2	S_3	Y_0	Y_1	Y_2	Y_3	Y_4	Y_5	Y_6	Y_7	Y_8	Y_9	Y_{10}	Y_{11}	Y_{12}	Y_{13}	Y_{14}	Y_{15}
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



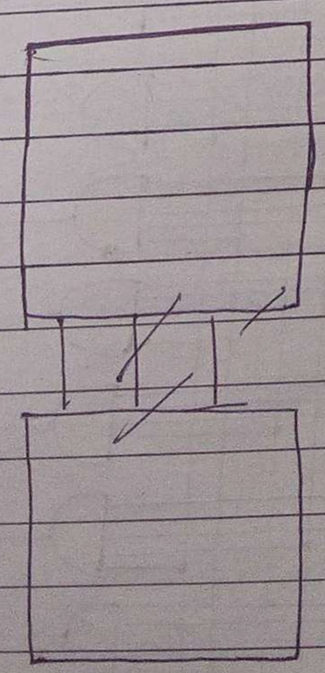


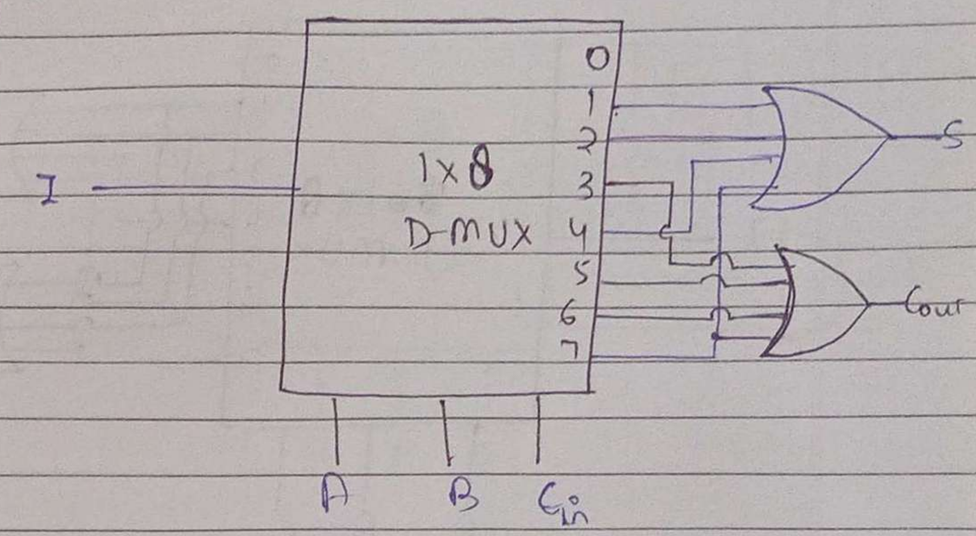
Implement full adder using De-mux —

S_0	S_1	S_2	S_3
A	B	C_{in}	S	C_{out}
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

$S(A, B, C_{in}) = \Sigma_m(1, 2, 4, 7)$

$C_{out}(A, B, C_{in}) = \Sigma_m(3, 5, 6, 7)$



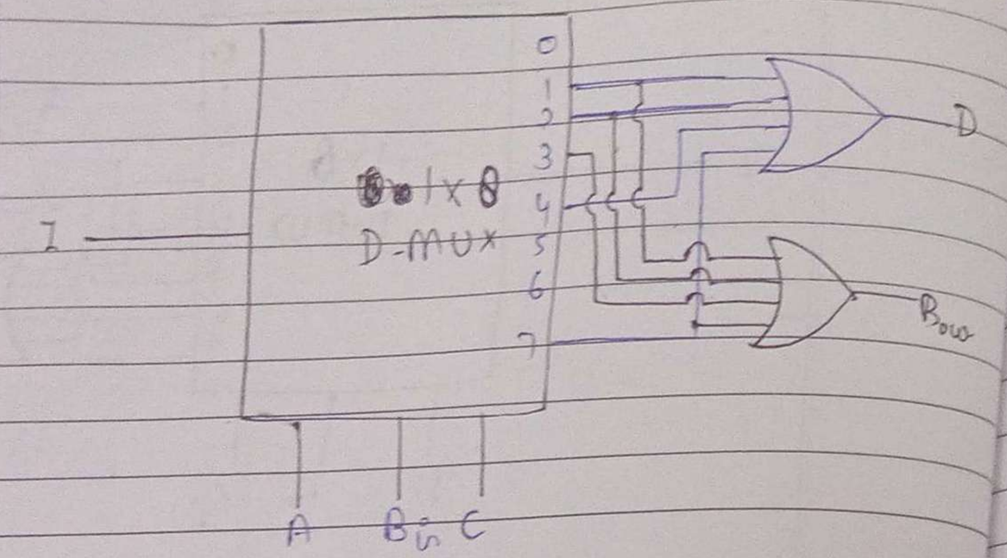


Q ② Full Subtractor using D-mux —

A	B_{in}	C	D	B_{out}
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

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

$B_{out} = \Sigma_m(1, 2, 3, 7)$

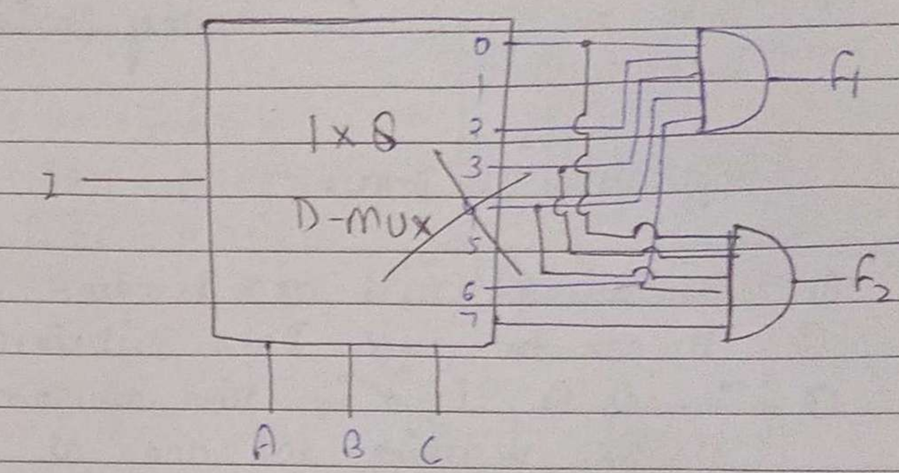


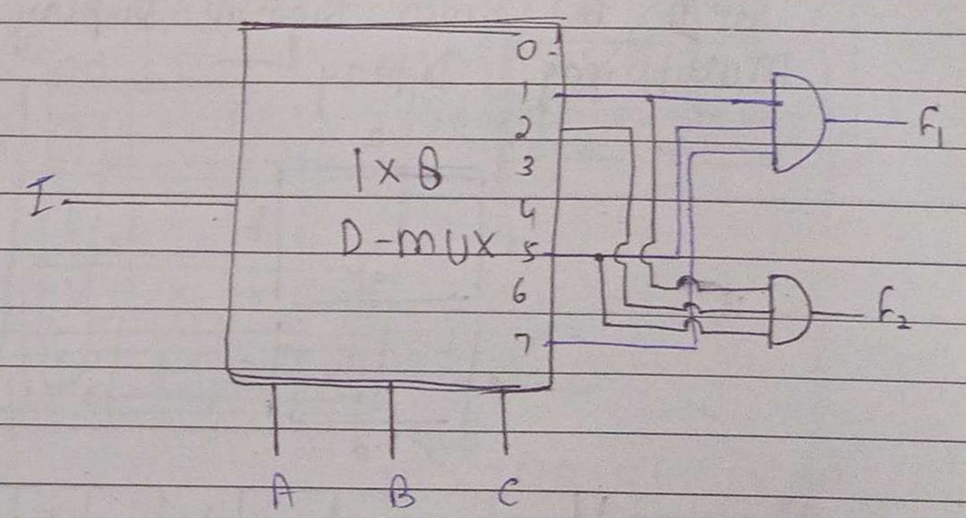
Q3 (i) $F_1(A, B, C) = \Sigma m(1, 5, 7)$
(ii) $F_2(A, B, C) = \Sigma m(1, 2, 5)$

~~Implementation table~~

Truth Table —

A	B	C	F_1	F_2
0	0	0	1	1
0	0	1	0	0
0	1	0	1	0
0	1	1	1	1
1	0	0	1	1
1	0	1	0	0
1	1	0	1	1
1	1	1	0	1





~~(ii)~~ ~~A B C F_1 F_2~~

⇒ Applications of Multiplexer:-

① Used as data selector to select one out of many data i/p.

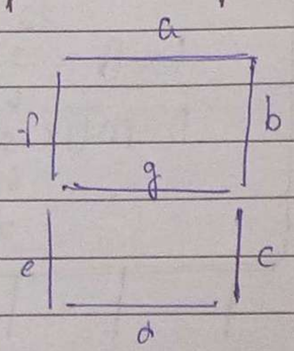
② Used in time division MUX and frequency division multiplexing at the centre end.

④ They are used in Analog to Digital and D/A converter.

Application of D-MUX —

① It can be used as Decoder.
② It can be used Data Distributor.
③ It can be used in time division multiplexing at the receiving end and as data separator.

BCD to Seven Segment Display Decoder/ Multiplexed Display —



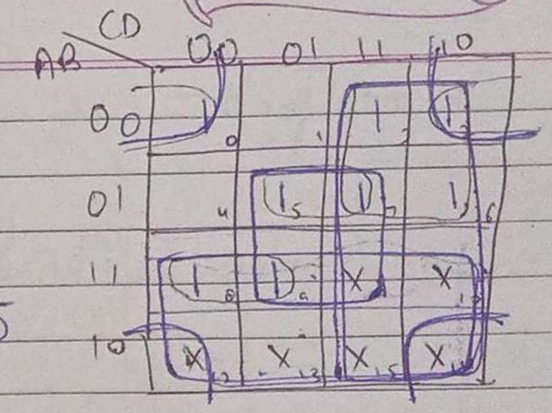
Decimal No.	Segment Activated	BCD NO. A B C D	a	b	c	d	e	f	g
0	a,b,c,d,e,f	0 0 0 0	1	1	1	1	1	1	0
1	b,c	0 0 0 1	0	1	1	0	0	0	0
2	a,b,d,e,g	0 0 1 0	1	1	0	1	1	0	1
3	a,b,c,d,g	0 0 1 1	1	1	1	1	0	0	1
4	b,c,g,f	0 1 0 0	0	1	1	0	0	1	1
5	a,c,d,f,g	0 1 0 1	1	0	1	1	0	1	1
6	a,c,d,e,f,g	0 1 1 0	1	0	1	1	1	1	1
7	a,b,c,f	0 1 1 1	1	1	1	0	0	1	0
8	a,b,c,d,e,f,g	1 0 0 0	1	1	1	1	1	1	1
9	a,b,c,d,f,g	1 0 0 1	1	1	1	1	0	1	1

K-map for a —

$a = \bar{A}C + \bar{A}\bar{B}\bar{D} + \bar{A}BD + AB\bar{C}$

$a = A + C + BD + \bar{B}\bar{D}$

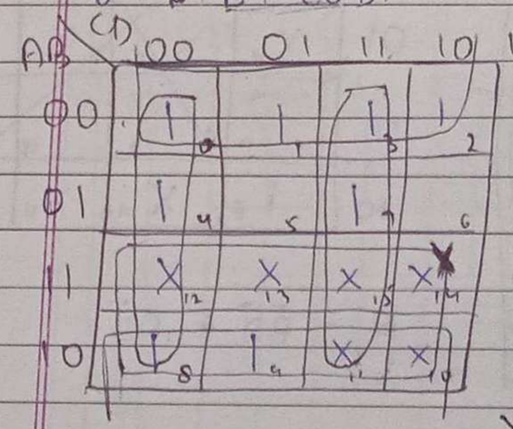
$a = A + C + B \odot D$

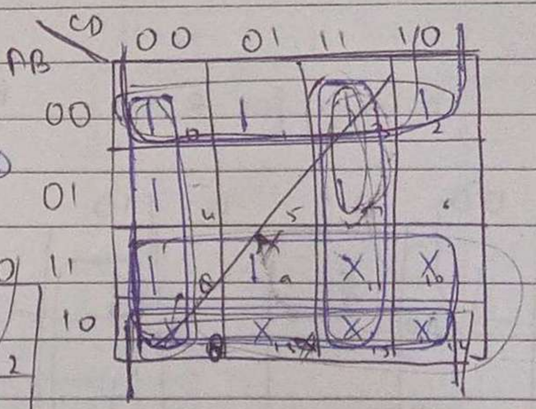


K-map for b —

$b = A + \bar{B} + \bar{C}\bar{D} + CD$

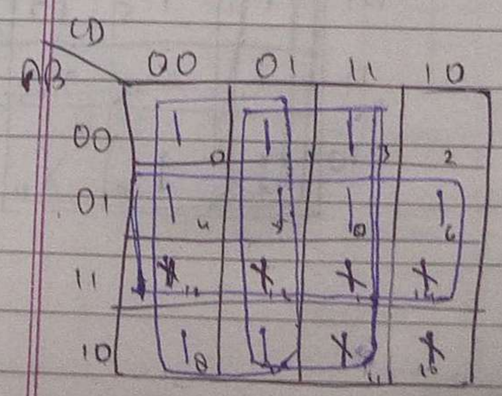
$b = A + \bar{B} + C \odot D$

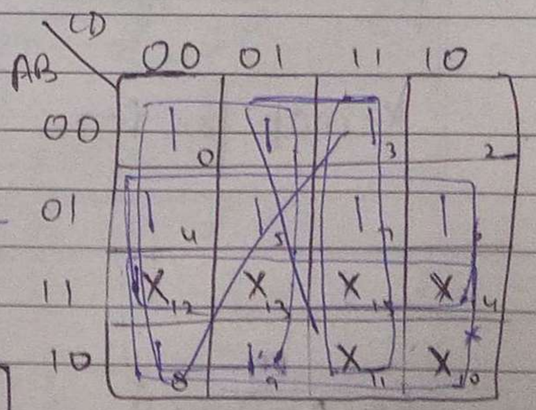




K-map for c —

~~$c = A + B + \bar{C} + \bar{C}\bar{D}$~~





$c = \bar{C} + D + B$

K-map for d —

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

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

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

K-map for e —

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

$e = \bar{B}\bar{D} + C\bar{D}$

K-map for f —

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

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

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

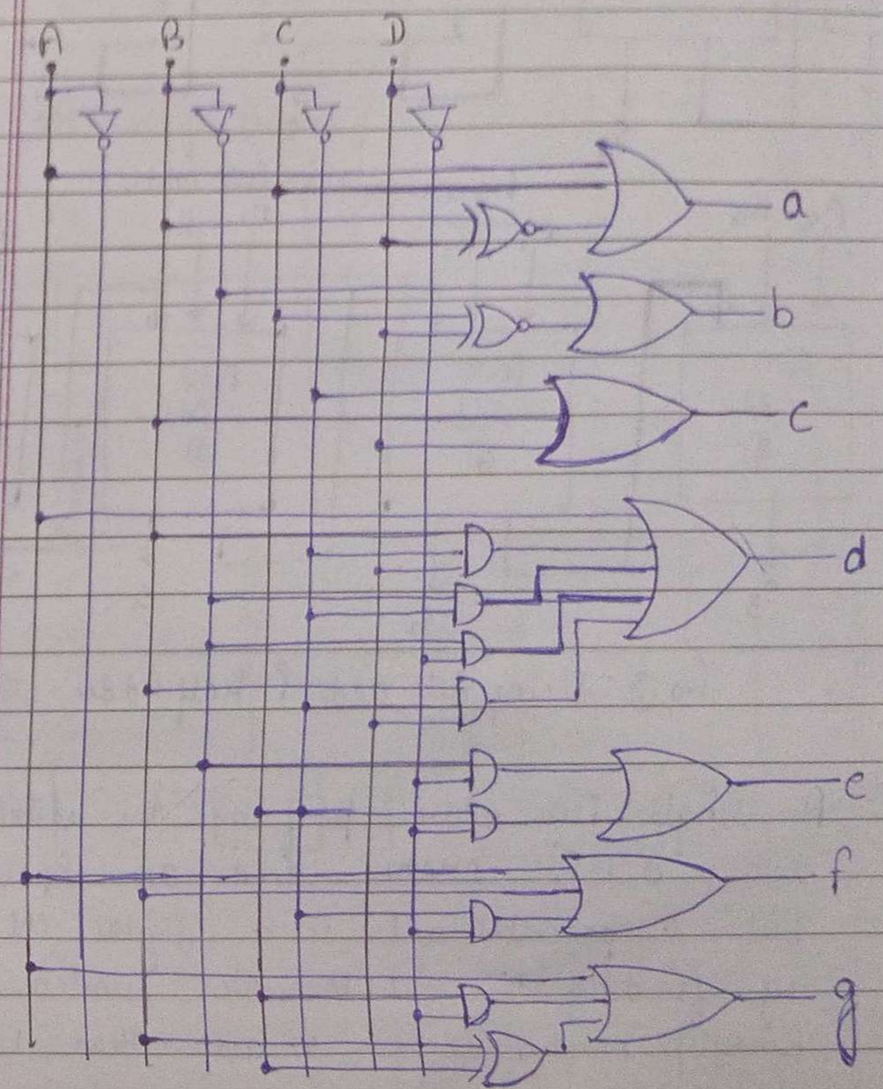
K-map for g —

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

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

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

Circuit design —



$A+B \rightarrow$ Sum + Carry

Date
Page

Parallel Adder / Binary Adder —

$$\begin{array}{rrrrl} & A_3 & A_2 & A_1 & A_0 \leftarrow \text{Augend Bit} \\ + & B_3 & B_2 & B_1 & B_0 \leftarrow \text{Addend Bits} \\ \hline C_{out} & S_3 & S_2 & S_1 & S_0 \end{array}$$

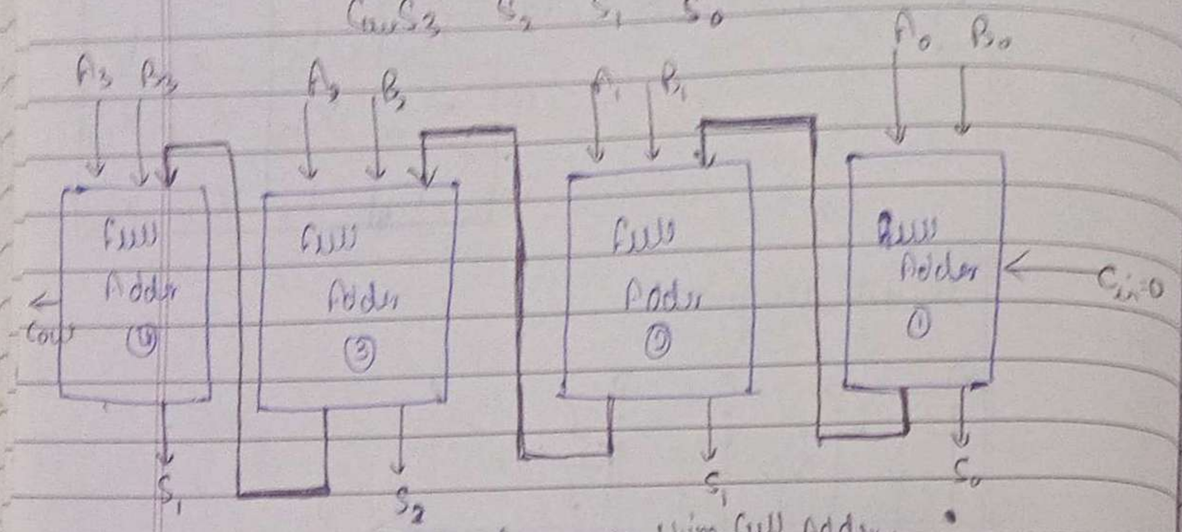


figure ① Using full adder.

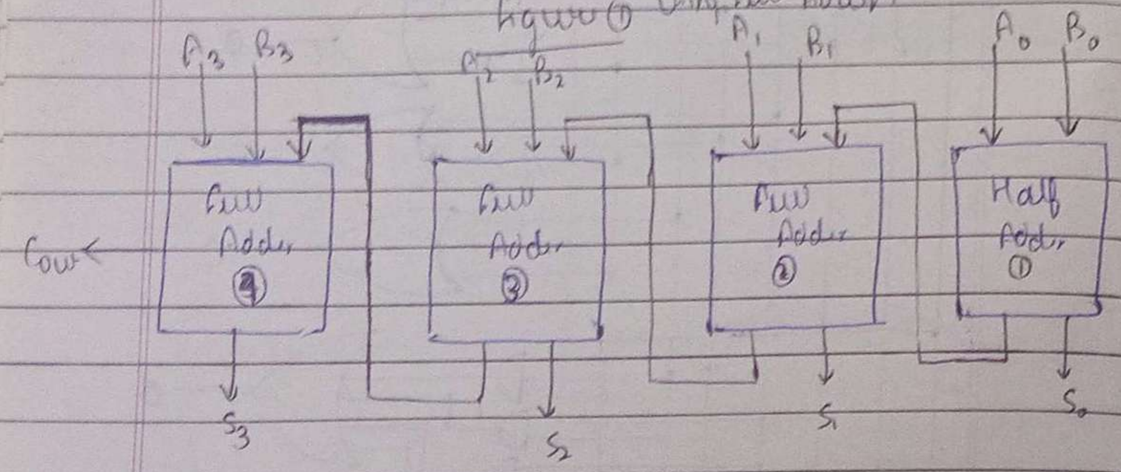
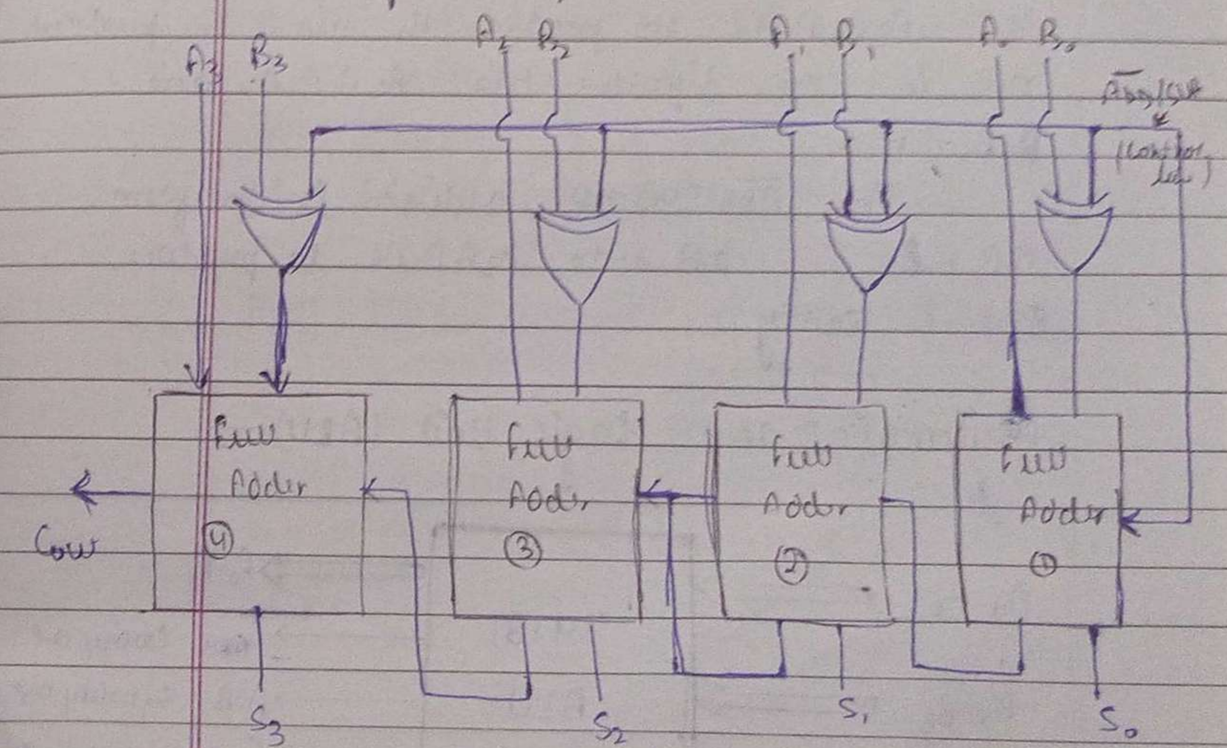


fig ②: Using full adder & Half Adder.

A single full adder performs the addition of two 0.1 Bit number and an input but a parallel adder is a digital ckt suitable of finding the arithmetic sum of two Binary numbers i.e. greater than 1 Bit in length by operating on corresponding pair of Bit in parallel.

A 4 bit Binary adder can be constructed by using four full adder as shown in fig ① or by using three full adders and 1 half adder as shown in fig ②. The four 4 full adder or 1 half and 3 full adder are connected in par i.e. the carry o/p of each adder is connected to the carry i/p of next higher order adder. n Bit binary adder or parallel adder can be constructed by using n no. of adders.

Binary Adder / Subtractor —

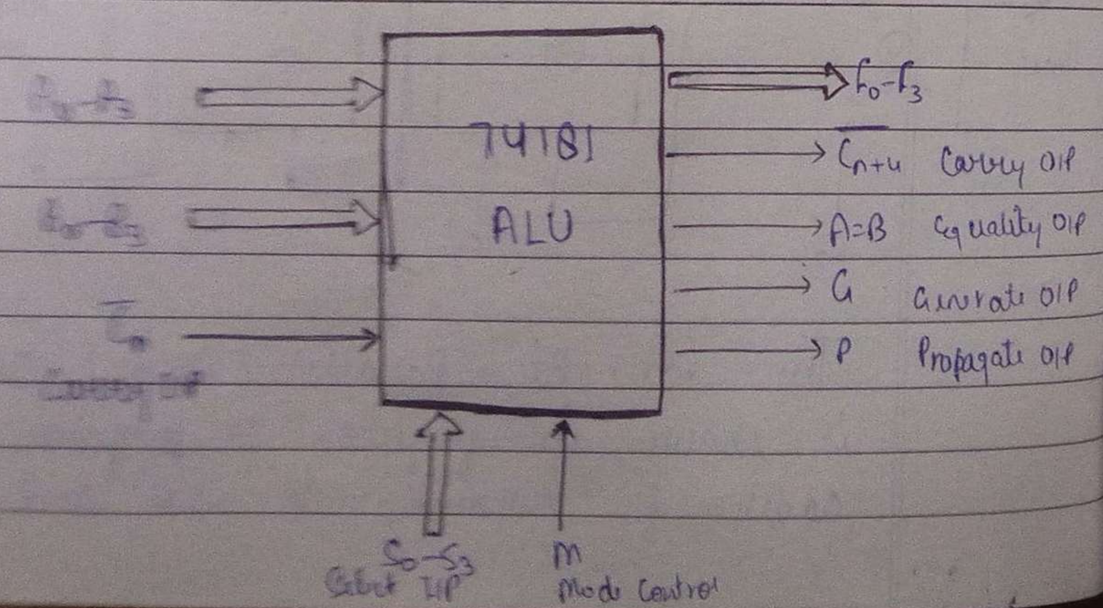


The four Bit Parallel Adder/ Subtractor is shown in figure. It performs both addition & subtract. It has two 4 Bit

input $A_3 A_2 A_1 A_0$ and $B_3 B_2 B_1 B_0$. The two operations cannot be performed simultaneously. Therefore a control signal is needed to perform this operation. $\overline{ADD}$/SUB control line connected with carry input of LSB of Full Adder is used to perform the operaⁿ of Addiⁿ & Subtraction. X-OR Gate are used as control inverter. When $\overline{ADD}$/SUB = 1, X-OR Gate produces 1's compliment of $B_3 B_2 B_1 B_0$. Since, one is given to the carry input of LSB of the adder, it is added to the compliment O/P of X-OR Gate which is equal to 2's compliment of $B_3 B_2 B_1 B_0$. 2's complimented $B_3 B_2 B_1 B_0$ is added to $A_3 A_2 A_1 A_0$ to produce the sum. The produce O/P is the difference b/w $A_3 A_2 A_1 A_0$ and $B_3 B_2 B_1 B_0$.

The $\overline{ADD}$/SUB = 0, addition is perform. $B_3 B_2 B_1 B_0$ is added to $A_3 A_2 A_1 A_0$ to produce sum & carry.

Arithmatic and Logic Unit (ALU) —



ALU is the heart of any microprocessor which is capable of performing arithmatic as well as logical operation. The f^n of various pin is given below—

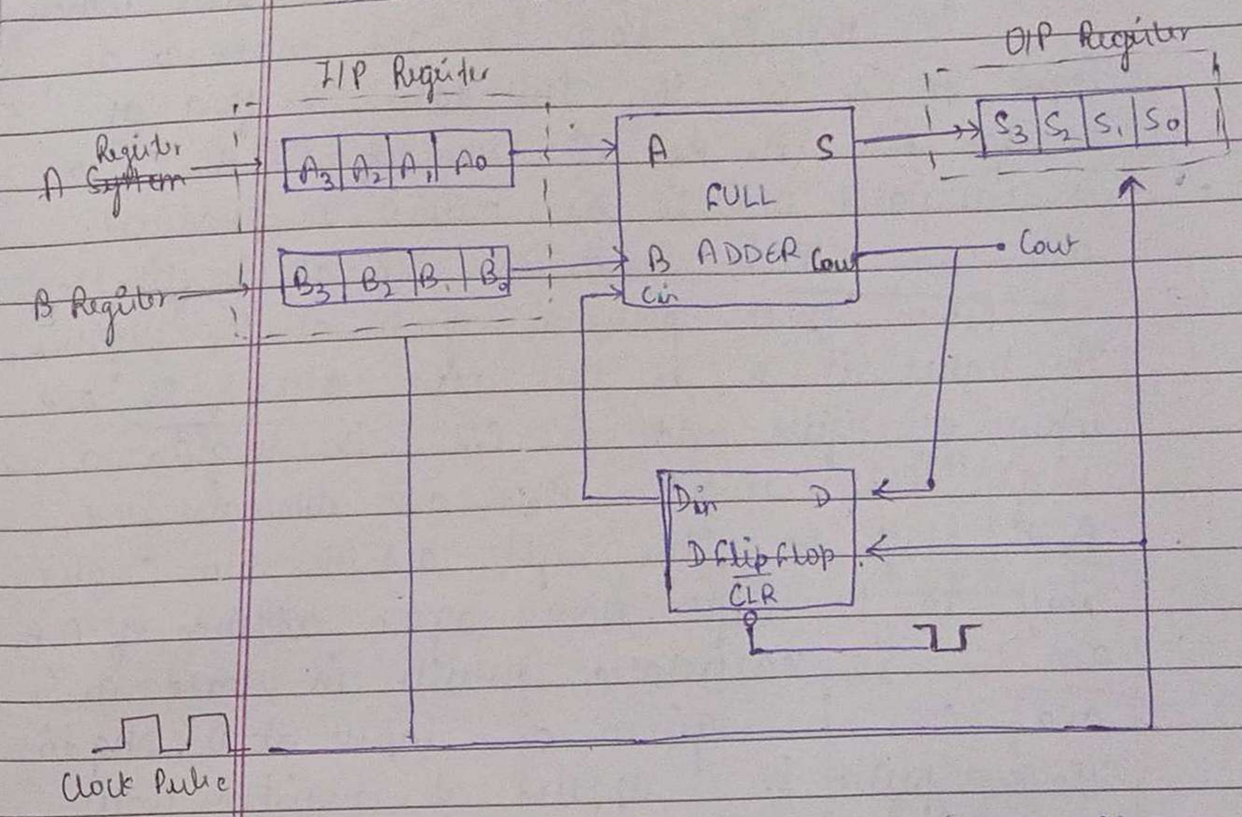
A & B —: 4-Bit Binary data input.
$\overline{C_n}$ —: Carry input (Active low)
F —: 4-Bit binary data output
$\overline{C_{n+4}}$ —: Carry output.
For subtracⁿ operation, it indicates the sign of the O/P. Logic 0 indicates the positive result and logic 1 indicates the negative result.
A=B —: Logic 1 on this line indicates A=B.
G —: Carry generated output.
$\overline{P}$ —: Carry propagated output.
Select inputs —: Selects the operation shown in the table below.
M —: Mode control, at M=0, Performs arithmatic operations.
at M=1, Perform logical operaⁿ.

Function table of 74181 ALU —

Active high data

Selections	M=1	M=0; Arithmetic operation	
	Logic fn	$\bar{C}_n = 1$ (no carry)	$\bar{C}_n = 0$ (with carry)
0 0 0 0	$F = \bar{A}$	$F = A$	$F = A$ PLUS 1
0 0 0 1	$\overline{A+B}$	$F = A+B$	$F = (A+B)$ PLUS 1
0 0 1 0	$F = \bar{A} \cdot B$	$F = A+B$	$F = (A+\bar{B})$ PLUS 1
0 0 1 1	$F = 0$	$F =$ MINUS 1 (2's comp)	$F =$ ZERO
0 1 0 0	$F = \overline{AB}$	$F = A$ PLUS $A\bar{B}$	$F = A$ PLUS $A\bar{B}$ PLUS 1
0 1 0 1	$F = \bar{B}$	$F = (A+B)$ PLUS $A\bar{B}$	$F = (A+B) + A\bar{B}$ PLUS 1
0 1 1 0	$F = A \oplus B$	$F = A$ MINUS B MIN 1	$F = A$ MINUS B
0 1 1 1	$F = A\bar{B}$	$F = A$ ~~PLUS~~ $A\bar{B} - 1$	$F = A\bar{B}$
1 0 0 0	$F = \bar{A}+B$	$F = A$ PLUS AB	$F = A$ PLUS AB PLUS 1
1 0 0 1	$F = \overline{A \oplus B}$	$F = A$ PLUS B	$F = A$ PLUS B PLUS 1
1 0 1 0	$F = B$	$F = (A+\bar{B})$ PLUS AB	$F = (A+\bar{B})$ PLUS AB PLUS 1
1 0 1 1	$F = AB$	$F = AB$ MINUS 1	$F = AB$
1 1 0 0	$F = 1$	$F = A$ PLUS A^*	$F = A$ PLUS A PLUS 1
1 1 0 1	$F = A+\bar{B}$	$F = (A+B)$ PLUS A	$F = (A+B)$ PLUS A PLUS 1
1 1 1 0	$F = A+B$	$F = (A+\bar{B})$ PLUS A	$F = (A+\bar{B})$ PLUS A PLUS 1
1 1 1 1	$F = A$	$F = A$ MINUS 1	$F = A$

Serial Adder —



Parallel Adder performs addition at higher speed but disadvantage is that it requires a large amount of logic circuits.

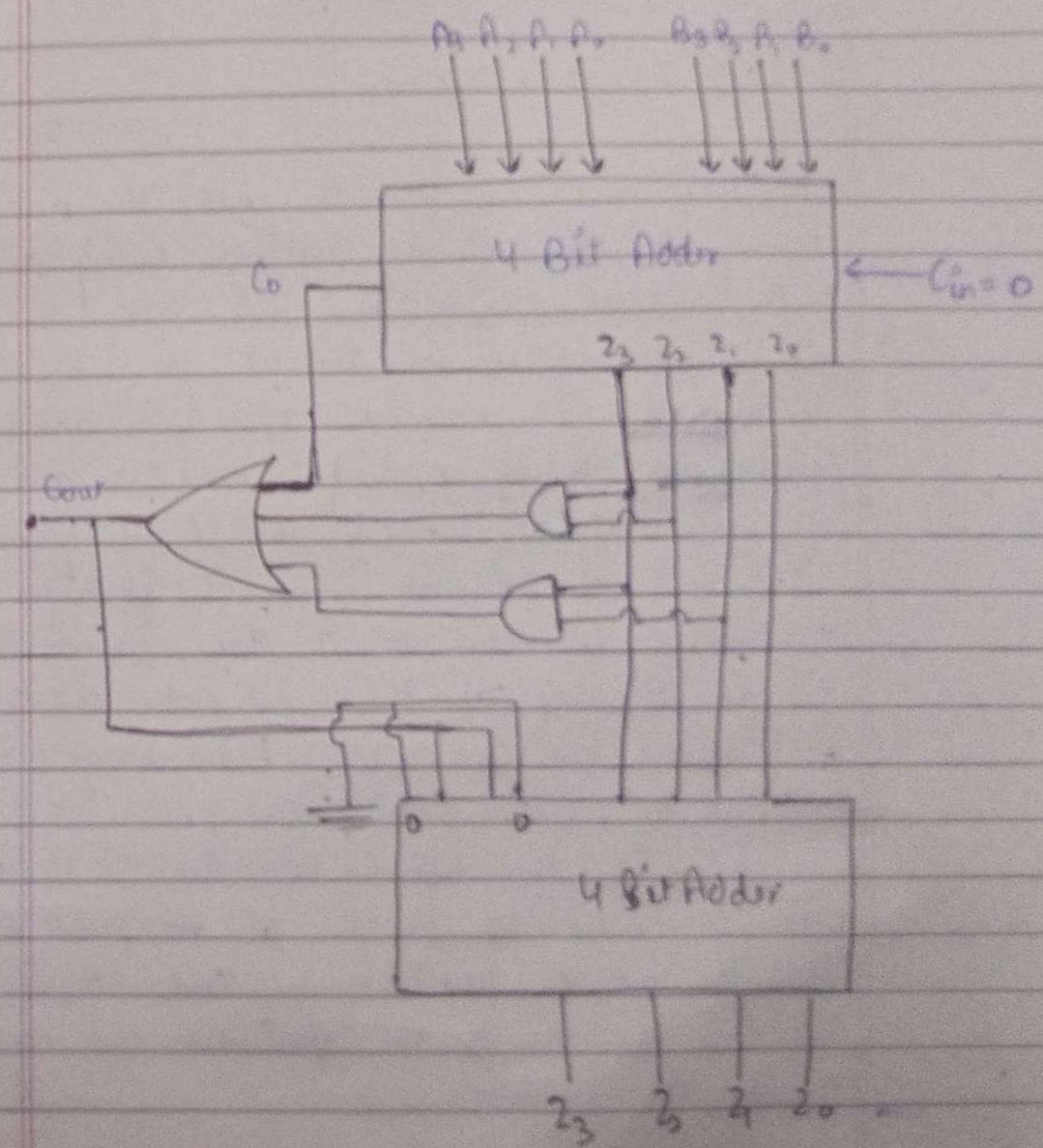
Serial adder performs Bit by Bit addition. So it requires simple circuitry than parallel adder but results in a low speed of operation.

For a Serial Adder, we need a full adder. Three registers are used for input of the Augend and Addend bit and one for sum. Right shift registers are used. Register A stores A_3, A_2, A_1, A_0 and register B stores B_3, B_2, B_1, B_0.

Working — As the flip flop is on reset mode, the D_n holds D and works as an input to C_{in} at the full adder. The first of the four bits (A_0, B_0), the sum of A_0, B_0, C_{in} (0) is calculated and S_0 as output is produced and C_{out} O/P is used as I/P at D flip flop. Now, clock pulse is given at the register. The value at D is the new value of Q_n and input at full adder is C_{in}. The register a right shift register A_0 B_0 are eliminated and A_1 B_1 shift to the right and the sum S_0 also shift to the right. Now again addition of A_1 B_1 and C_{in} is calculated results in stored as S_1 and C_{out} is given as input at D. Again clock pulse is applied at register value of A_2 B_2 now shift to right and S_1 also shift to the right in the register. O/P of D-flip flop Q_n is (new value of D) input given at C_{in}. The addition is again performed on A_2 B_2 and C_{in} and sum is stored as S_2 and C_{out} is given input at D. Again clock pulse is applied, the values in the registers shift to their right. O/P of D-flip flop Q_n (new value of D) is input given at C_{in}. The addition is again performed on A_3 B_3 and C_{in}. The addition is given and the sum is stored at S_3 and C_{out} is produced. As the sum is generated, no clock pulse is applied now. Hence, the value of sum and C_{out} is calculated. In this way, the serial adder performs the addition of n bit and produces Sum and Cout as O/P.

BCD Adder —:

$$\begin{array}{r} A_3 A_2 A_1 A_0 \\ + \; B_3 B_2 B_1 B_0 \\ \hline C_0 Z_3 Z_2 Z_1 Z_0 \end{array}$$



K-map for Invalid BCD nos —

z_3z_2 \ z_1z_0	00	01	11	10
00				
01				
11	1	1	1	1
10			1	1

$C_{out} = z_3 z_2 + z_3 z_1 + C_0$

A BCD Adder is used to find the sum and carry of the binary numbers using a two 4-Bit adders, $A_3A_2A_1A_0$ and $B_3B_2B_1B_0$ and $C_{in} = 0$ are given as I/P at the 4 Bit Adder. It gives the O/P sum as $z_3z_2z_1z_0$ and C_0 as carry out. N of z_3, z_2 (z_3 & z_2) and AND of z_3 and z_1 as ($z_3 \cdot z_1$) and C_0 is given as a I/P at the OR Gate. C_{out} is given as O/P. If C_{out} is calculated as 0 then, 0000 is added to the $z_3z_2z_1z_0$ at the 4 Bit adder and the O/P is produced as C_{out} $z_3z_2z_1z_0$ (no change occurs if the nos are valid BCD numbers).

If the C_{out} is calculated as 1, then 0110 is added to $z_3z_2z_1z_0$ at the 4 Bit Adder and the O/P is produced as C_{out} $z_3z_2z_1z_0$. This C_{out} $z_3z_2z_1z_0$ is the final sum and carry of the nos provided.

Derivation of BCD Adder —

Binary sum					BCD sum					Decimal no
C_0	z_3	z_2	z_1	z_0	C_{out}	z_3	z_2	z_1	z_0	
0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	1	0	0	0	0	1	1
0	0	0	1	0	0	0	0	1	0	2
0	0	0	1	1	0	0	0	1	1	3
0	0	1	0	0	0	0	1	0	0	4
0	0	1	0	1	0	0	1	0	1	5
0	0	1	1	0	0	0	1	1	0	6
0	0	1	1	1	0	0	1	1	1	7
0	1	0	0	0	0	1	0	0	0	8
0	1	0	0	1	0	1	0	0	1	9
0	1	0	1	0	1	0	0	1	0	10
0	1	0	1	1	1	0	0	1	1	11
0	1	1	0	0	1	0	1	0	0	12
0	1	1	0	1	1	0	1	0	1	13
0	1	1	1	0	1	0	1	0	0	14
0	1	1	1	1	1	0	1	0	1	15
1	0	0	0	0	1	0	1	1	0	16
1	0	0	0	1	1	0	1	1	1	17
1	0	0	1	0	1	1	0	0	0	18
1	0	0	1	1	1	1	0	0	1	19