

Ques 1. a) $BC5_{16} - A3B_{16}$

$$\begin{array}{r} & \text{B} \\ & | \\ B & 9 & 5 \\ - A & 3 & B \\ \hline 1 & 8 & A \\ \hline \end{array} \quad F5 = 21$$

⇒ Excess - 3 representation for 353

$$\begin{array}{rcl} 3 & 5 & 3 \\ = & 0011 & 0101 & 0011 \\ + & 0011 & 0011 & 0011 \\ \hline 0 & 1 & 1 & 0 & 1 & 000 & 011 & 0 \end{array}$$

⇒ $11011011 - 10111$ using 2's compliment method.

→ II. $00010111 \rightarrow 1\text{'s compliment}$

$$\begin{array}{rcl} & = 11101000 & \text{For} \\ & + 1 & \rightarrow 2\text{'s complement} \\ \hline & 11101001 & \end{array}$$

→ I. 10111 to 8 bits

$$= 00010111 = 23$$

→ III. Then, Add 11011011

$$\begin{array}{rcl} & + 11101001 & \\ \hline & 11000100 & (\text{carry out}) \end{array}$$

$$11011011_2 = 219$$

$$00010111_2 = 23$$

$$\begin{array}{r} 219 \\ - 23 \\ \hline 196 \end{array}$$

$$= 11000100_2 = 196$$

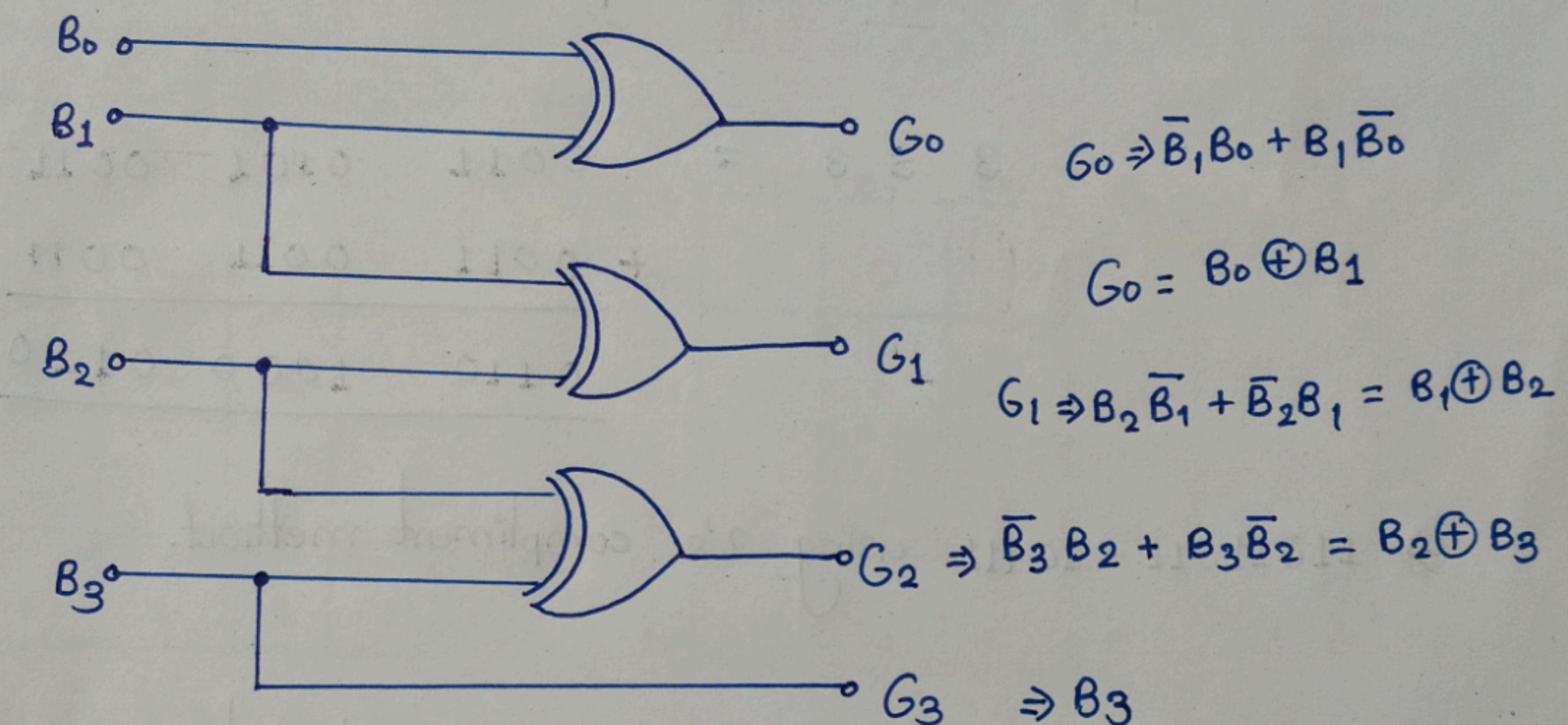
→ IV. Therefore $= 11011011 - 10111$

$$= 11000100$$

$$\text{d)} (A6BF5)_{16} = (?)_2$$

$$= \begin{array}{ccccc} A & & 6 & & B \\ (1010) & & (0110) & & (1011) \\ & & & & (1111) \\ & & & & (0101) \end{array}_2$$

2. Design a 4-bit gray to binary code converter. (Derive equations with the help of K-map).



Truth Table

	B_3	B_2	B_1	B_0	G_3	G_2	G_1	G_0
0	0	0	0	0	0	0	0	0
1	0	0	0	1	0	0	0	1
2	0	0	1	0	0	0	1	1
3	0	0	1	1	0	0	1	0
4	0	1	0	0	0	1	1	0
5	0	1	0	1	0	1	1	1
6	0	1	1	0	0	1	0	1
7	0	1	1	1	0	1	0	0
8	1	0	0	0	1	1	0	0
9	1	0	0	1	1	1	0	1
10	1	0	1	0	1	1	1	1
11	1	0	1	1	1	1	1	0

12	1 1 0 0	1 0 1 0
13	1 1 0 1	1 0 1 1
14	1 1 1 0	1 0 0 1
15	1 1 1 1	1 0 0 0

K - MAP

		$G_1 G_0$				
		$G_3 G_2$	00	01	11	10
$G_3 G_2$	00	0	1	0	1	
01	1	0	1	0		
11	0	1	0	1		
10	1	0	1	0		

		$G_1 G_0$				
		$G_3 G_2$	00	01	11	10
$G_3 G_2$	00	0	0	1	1	
01	1	1	0	0		
11	0	0	1	1		
10	1	1	0	0		

$$B_0 = \bar{G}_3 \bar{G}_2 G_0 + \bar{G}_3 \bar{G}_2 G_1 + \bar{G}_3 G_2 + G_2 G_1 G_0 +$$

$$G_3 G_2 G_0 + G_3 G_2 G_1 + G_3 \bar{G}_2 + G_3 \bar{G}_2 G_1 G_0$$

$$= G_3 \oplus G_2 \oplus G_1 \oplus G_0 \quad B_3 =$$

		$G_1 G_0$				
		$G_3 G_2$	00	01	11	10
$G_3 G_2$	00	0	0	0	0	
01	1	1	1	1		
11	0	0	0	0		
10	1	1	1	1		

		$G_1 G_0$				
		$G_3 G_2$	00	01	11	10
$G_3 G_2$	00	0	0	0	0	
01	0	0	0	0		
11	1	1	1	1		
10	1	1	1	1		

$$B_2 = \bar{G}_3 G_2 + G_3 \bar{G}_2 = G_3 \oplus G_2$$

$$B_3 = G_3$$

3. Simplify the expression $\Pi(0, 2, 3, 4, 5, 6)$ using k-map and implement it in AND logic as well as NAND logic

		BC				
		A	00	01	11	10
BC	00	0	1	0	0	
01	0	0	0	1	1	
11	0	0	0	0	1	
10	1	1	1	0	0	

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

NAND only Implementation:-

$$I. \quad F = \bar{A} \bar{B} C + A B C$$

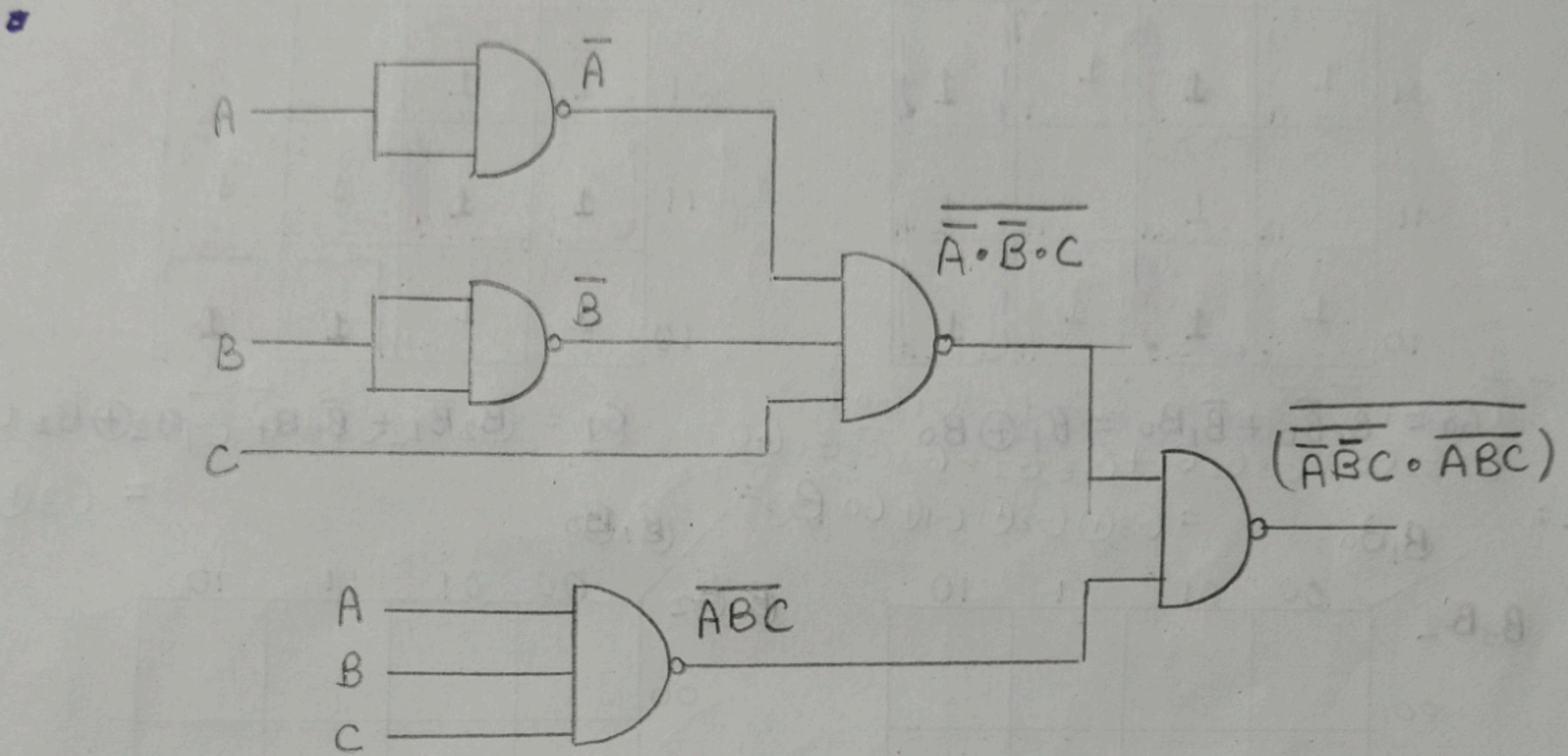
((Complement using De Morgan's)
[$(\overline{x+y}) = x+y$])

$$F = \bar{A} \bar{B} C + \overline{\bar{A} \bar{B} C}$$

$$F = (\overline{\bar{A} \cdot \bar{B} \cdot C}) + (\overline{\bar{A} \bar{B} C}) \rightarrow (\text{Using } \overline{\overline{xyz}} = \bar{x} + \bar{y} + \bar{z})$$

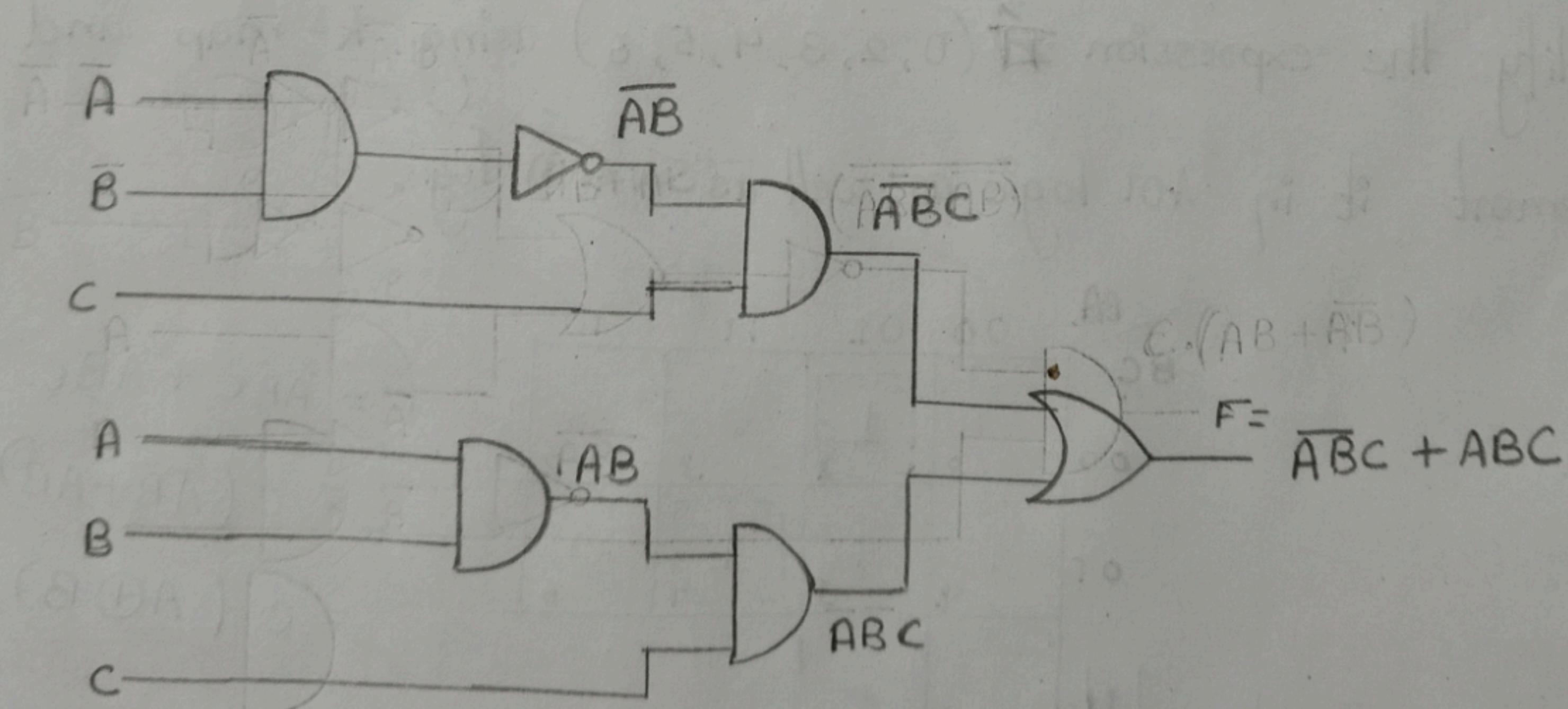
To convert the OR gate into NAND gate

$$F = \left((\overline{A}\overline{B}C) \circ (\overline{A}B\overline{C}) \right) \rightarrow \left(\text{using } x + y = (\overline{x}\overline{y}) \right)$$



Algebraic logic of expression:-

$$\begin{aligned} F &= \overline{A}\overline{B}C + A\overline{B}C \\ &= C(\overline{A}\overline{B} + AB) \\ &= C(A \oplus B) \end{aligned}$$



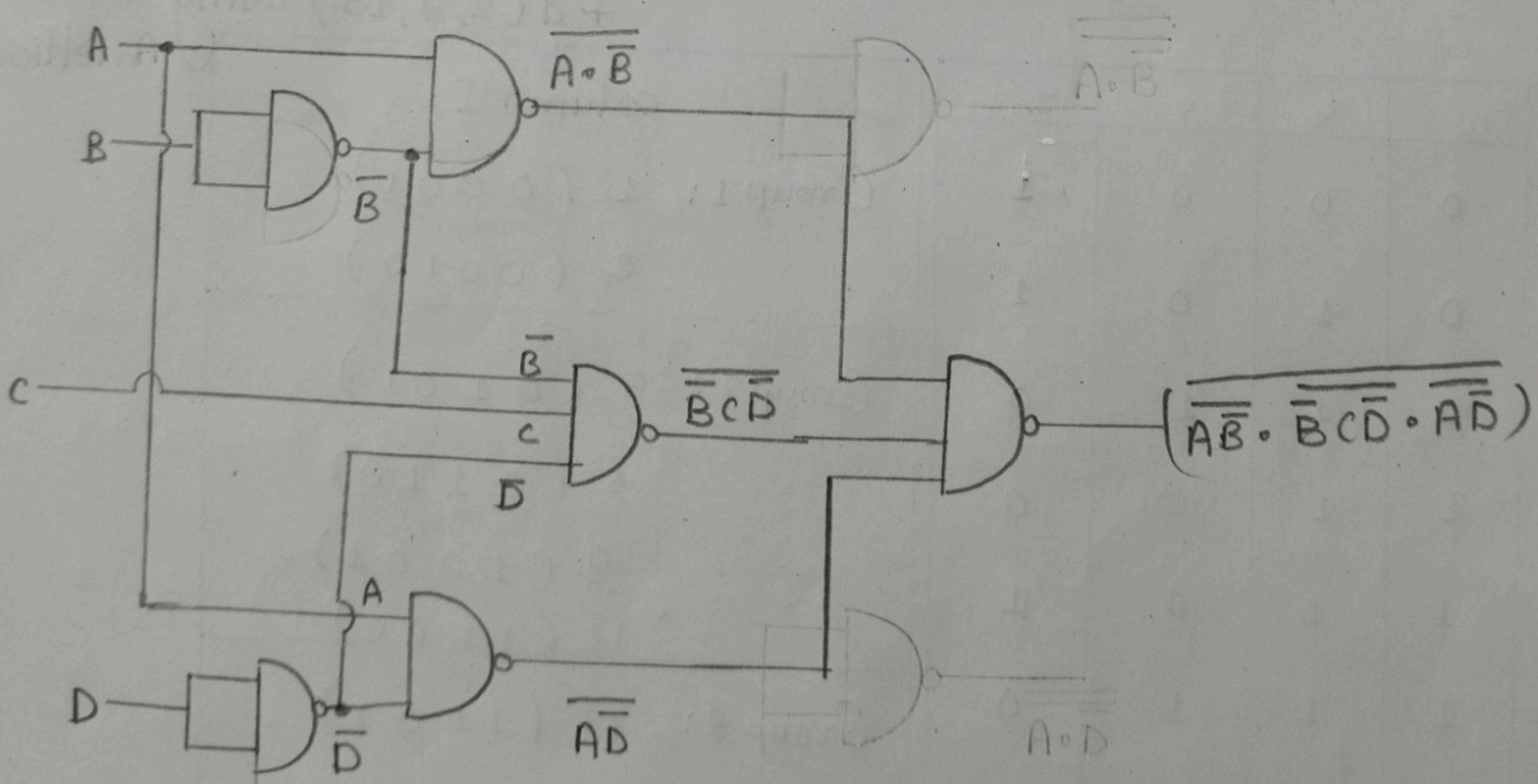
4. Reduce the expression $\pi(2, 8, 9, 10, 11, 12, 14)$ using mapping and implement it in universal logic

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

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

For Product of SUM

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

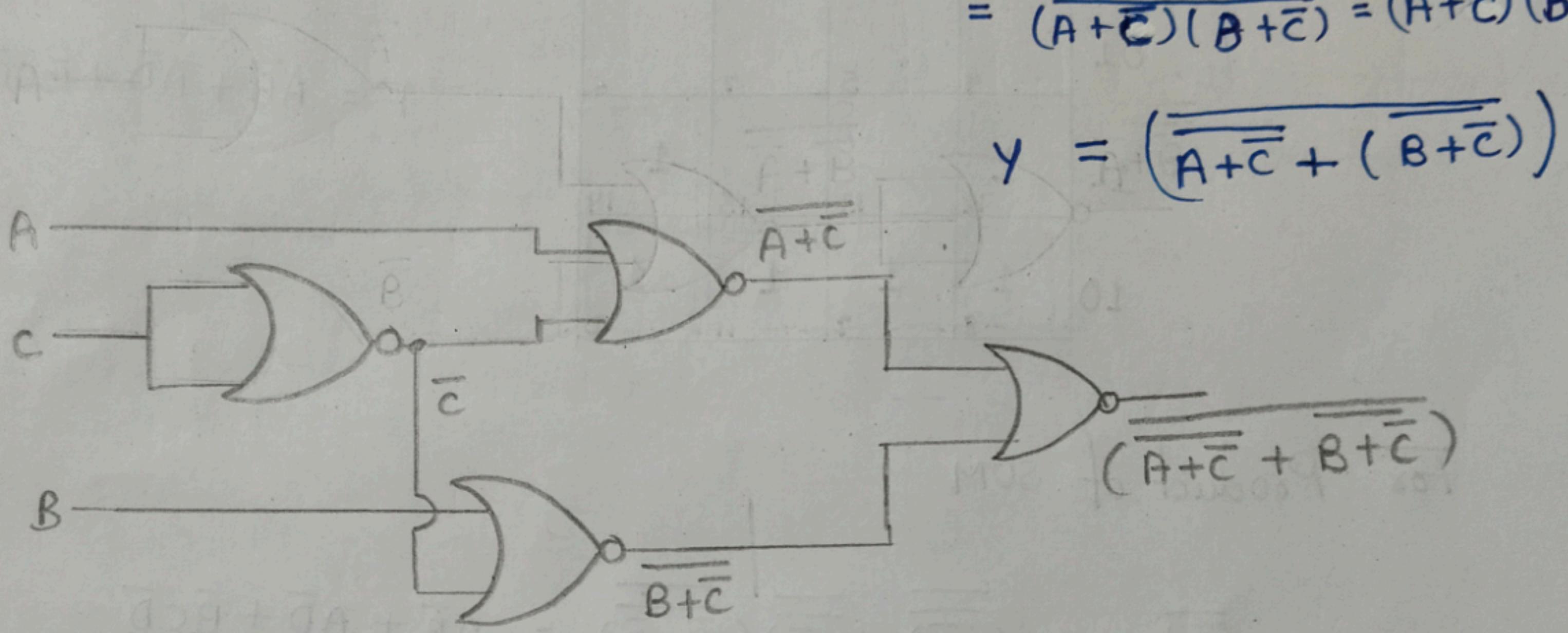


5. Implement $AB + \overline{C}$ using NOR gates only.

$$AB + \overline{C} = (\cancel{\text{NOR gate}}) \quad (\text{using 2 inputs})$$

$$= (\cancel{\text{NOR gate}}) + \cancel{\text{NOR gate}} \quad (\cancel{\text{NOR gate}}) = \cancel{\text{NOR gate}}$$

$$y = (A + \overline{C})(B + \overline{C}) \quad (\because x + yz = (x+y)(x+z)) \\ = \overline{(A + \overline{C})(B + \overline{C})} = \overline{\overline{A + \overline{C}}} \cdot \overline{\overline{B + \overline{C}}}$$



Ques 6. Reduce the expression $F(w, x, y, z) = \sum(1, 5, 6, 12, 13, 14)$

+ d(2, 9, 15) using tabular method.

	w	x	y	z	column I
M ₁	0	0	0	1	Group 1: 1 (0 0 0 1) 2 (0 0 1 0)
M ₅	0	1	0	1	
M ₆	0	1	1	0	Group 2: 5 (0 1 0 1) 6 (0 1 1 0)
M ₁₂	1	1	0	0	
M ₁₃	1	1	0	1	9 (1 0 0 1) 12 (1 1 0 0)
M ₁₄	1	1	1	0	
D ₂	0	0	1	0	Group 3: 13 (1 1 0 1) 14 (1 1 1 0)
D ₉	1	0	0	1	
D ₁₅	1	1	1	1	Group 4: 15 (1 1 1 1)

COLUMN II

$$\text{Group - I :- } (1, 5) \rightarrow 0 - 0 1$$

$$(1, 9) \rightarrow - 0 0 1$$

$$(2, 6) \rightarrow 0 - 1 0 \checkmark$$

$$\text{Group - II :- } (5, 13) \rightarrow - 1 0 1$$

$$(6, 14) \rightarrow - 1 1 0 \checkmark$$

$$(9, 13) \rightarrow 1 - 0 1$$

$$(12, 13) \rightarrow 1 1 0 -$$

$$(12, 14) \rightarrow 1 1 - 0$$

$$\text{Group - III :- } (13, 15) \rightarrow 1 1 - 1$$

$$(14, 15) \rightarrow 1 1 1 -$$

COLUMN III

$$(1, 5, 9, 13) \rightarrow - - 0 1 \checkmark$$

$$(1, 9, 5, 13) \rightarrow - - 0 1$$

$$(12, 14, 13, 15) \rightarrow 1 1 - - \checkmark$$

$$(12, 13, 14, 15) \rightarrow 1 1 - -$$

List of Prime Implicants -

$$\bar{w}yz$$

$$xy\bar{z}$$

$$\bar{y}z$$

$$wx$$

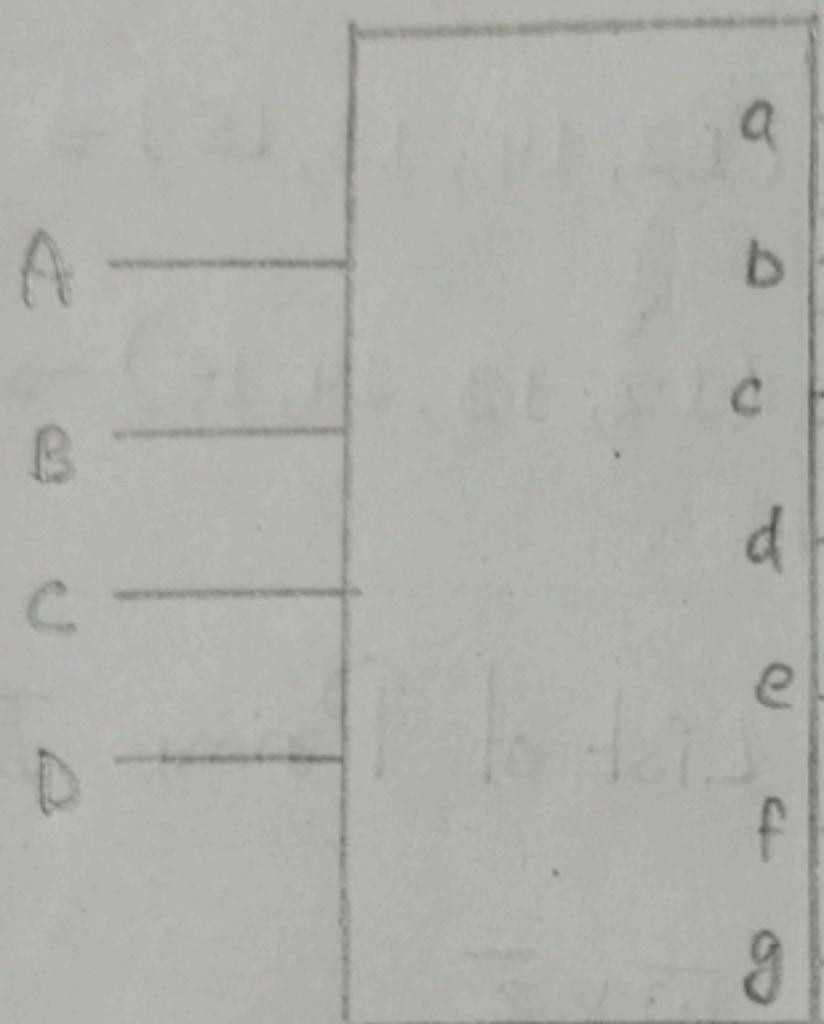
Verifying using k-Map

$$\leq (1, 5, 6, 12, 13, 14) + d(2, 9, 15)$$

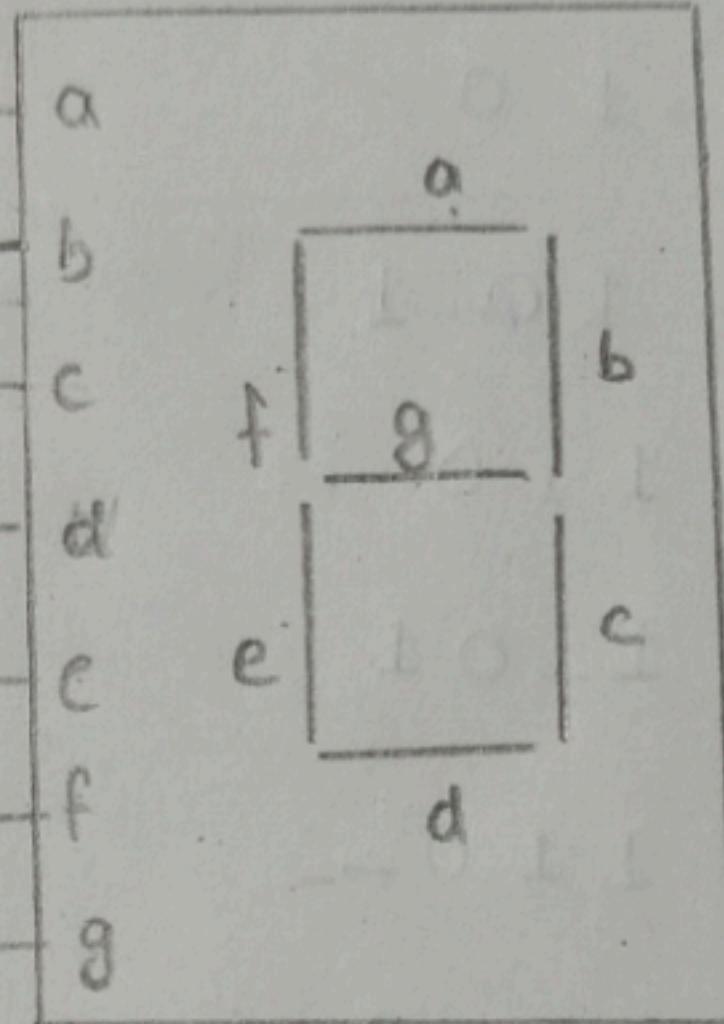
	wx	yz		
wx	00	01	11	10
00	0	1	3	2
01	4	5	7	6
11	1	1	X	1
10	8	X	11	10

$$xy\bar{z} + \bar{w}yz + \bar{y}z + wx$$

7. Draw the diagram of BCD to 7 - segment decoder and write its truth table.



BCD to 7 Segment
Decoder



7-Segment
LED Display

Truth Table

Inputs				Display	O/P Decoder	7-Segment code					
A	B	C	D		a	b	c	d	e	f	g
0	0	0	0	□	1	1	1	1	1	1	0
0	0	0	1		0	1	1	0	0	0	0
0	0	1	0	匚	1	1	0	1	1	0	1
0	0	1	1	匚	1	1	1	1	0	0	1
0	1	0	0	𠂔	0	1	1	0	0	1	1
0	1	0	1	𠂊	1	0	1	1	0	1	1
0	1	1	0	𠂉	1	0	1	1	1	1	1
0	1	1	1	𠂊	1	1	1	0	0	0	0
1	0	0	0	𠂔	1	1	1	1	1	1	1
1	0	0	1	𢈕	1	1	1	1	0	1	1

Que 8. Implement $f(a, b, c) = ab + b'c$ using 4×1 MUX 9.

$$F(a, b, c) = ab + b'c$$

We need 4-data Inputs I_0, I_1, I_2, I_3

2: Select lines: S_1 and S_0

1 output y

A	B	C	AB	$B'C$	$AB + B'C$
0	0	0	0	0	0
0	0	1	0	1	1
0	1	0	0	0	0
0	1	1	0	0	1
1	0	0	0	0	0
1	0	1	0	1	1
1	1	0	1	0	1
1	1	1	1	0	1

$$S_1 = b \text{ (MSB)}$$

$$I_0 = 0 \text{ (for } b=0, c=0\text{)}$$

$$S_0 = c \text{ (LSB)}$$

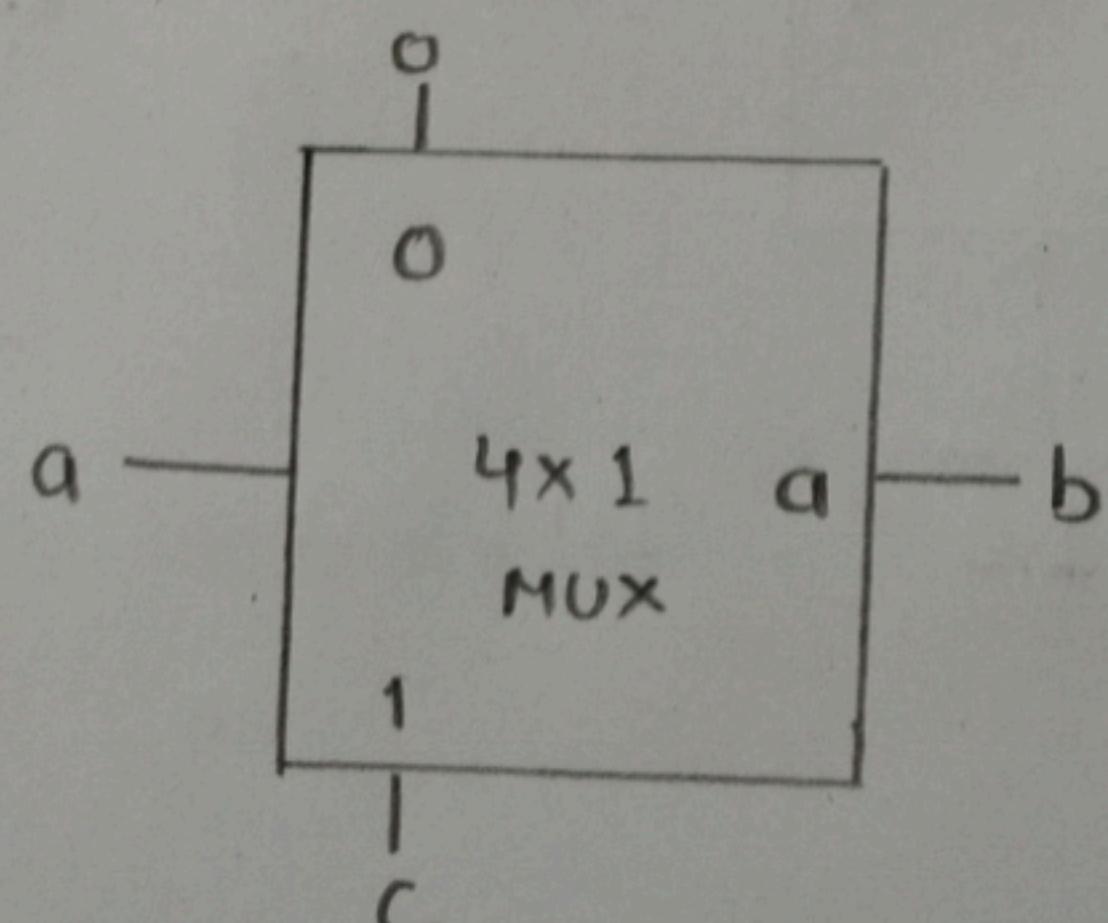
$$I_1 = 1 \text{ (for } b=0, c=1\text{)}$$

$$I_2 = a \text{ (for } b=1, c=0\text{)}$$

$$I_3 = a \text{ (for } b=1, c=1\text{)}$$

Where $b=0, b=b'c=c$

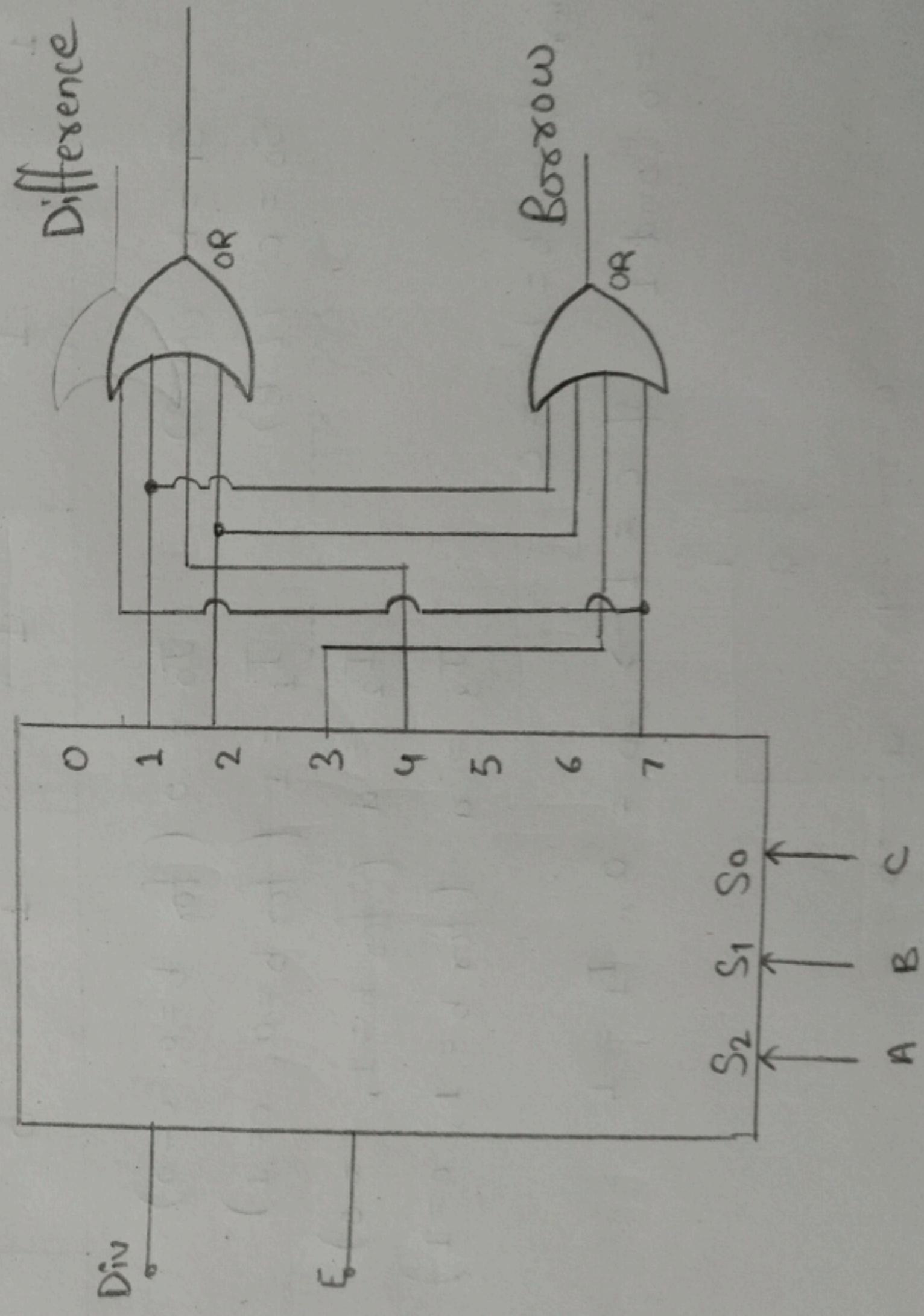
if $c=0$ and 1 if $c=1 \rightarrow I_0 = 0, I_1 = 1$



9. Implement Full Subtractor using 1:8 DEMUX.

Input			Output		
A	B	C	Difference	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	

$$\begin{aligned} D &= \sum_m (1, 2, 4, 7) \\ B_0 &= \sum_m (1, 2, 3, 7) \end{aligned}$$



Que 10. Explain addition of 2 negative No's in 2's Complement form
with the help of an example.

Take $(-7) + (-5)$

I. -7 in binary

$$= 0111$$

II. Invert the bit's (1's complement)

$$= 1000$$

III. Convert it to 2's complement

by adding 1 to LSB

$$\begin{array}{r} 1000 \\ + 1 \\ \hline \end{array}$$

$$\begin{array}{r} 1001 \\ \hline \end{array}$$

\rightarrow (2's complement)

Do the same to (-5)

I. -5 to binary = 0101

II. 1's complement = 1010

III. 2's complement = 1010

$$\begin{array}{r} + 1 \\ \hline 1011 \\ \hline \end{array}$$

Now add both the No's.

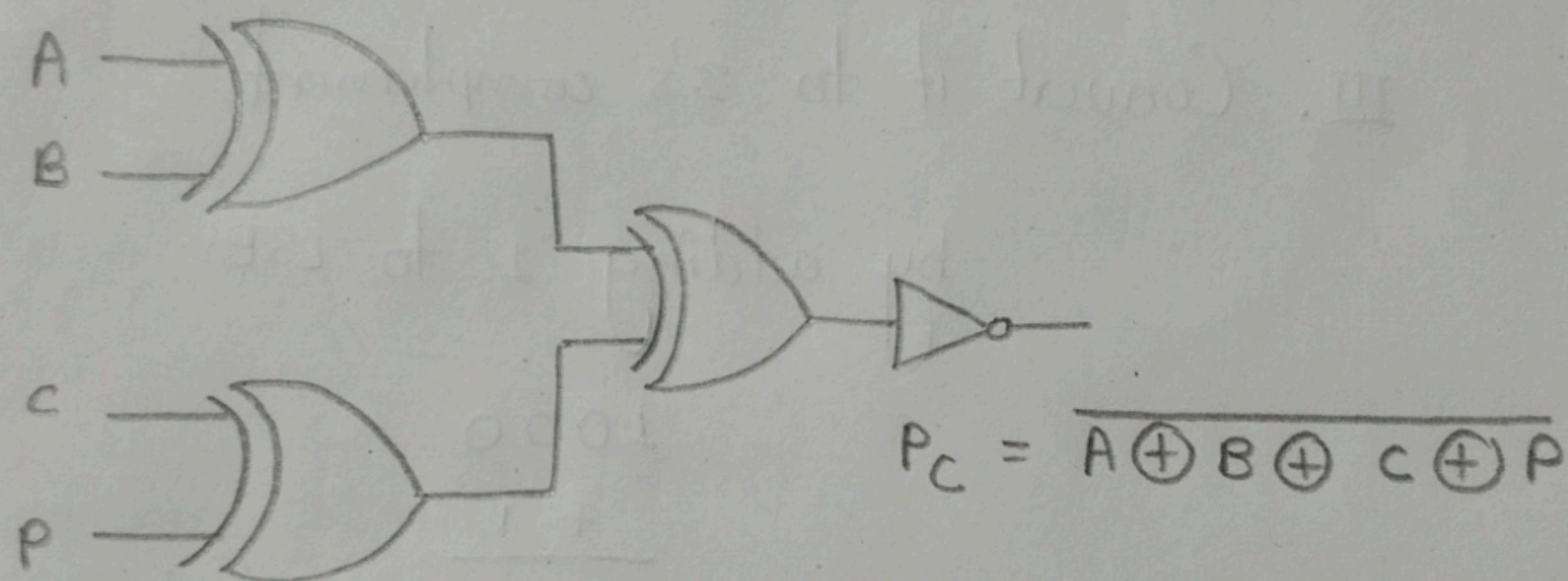
$$\begin{array}{r} 1001 \\ 1011 \\ \hline 10100 \end{array} = 10100 = 0100 \text{ (Ignore the carry)}$$

= 1011 (1's complement)

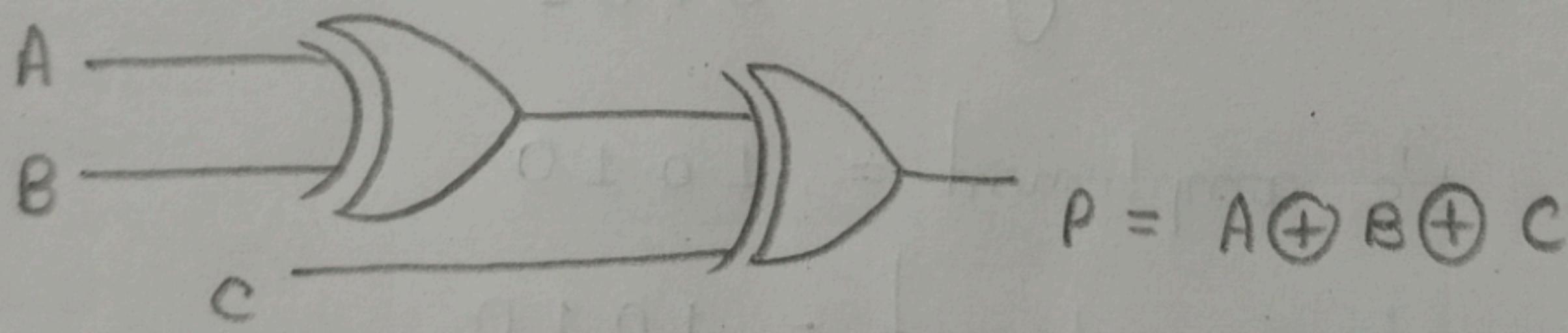
+ 1 (2's complement)

Ques 11. Explain how a parity checker can also behave as parity generator circuit.

A parity checker behaves as a parity generator by essentially mirroring the basic of a generator bit operations on the received data and the received parity bit



Parity checker



Parity generator

Ques 12. Differentiate between multiplexers and encoders. Design a 16×1 MUX using 4×1 MUX's

A Multiplexer also known as data selector, is a combinational logic circuit that selects digital input signal and forwards the selected input signal into a single output line.

An Encoder is a combinational logic circuit that converts information from one format or code to another.

Design of 16 MUX from 4×1 MUX's

