IAT-1 SOLUTION ADE

Answer COMPULSARILY

Simplify the following expression using Quine –McCluskey's method

F(A,B,C,D)= $\Sigma m(0,1,2,3,10,11,12,13,14,15)$.

Solution : Step 1 : List all minterms in the binary form as shown in table (column (a)).

Step 2 : Arrange the minterms according to no. of 1s, as shown in table (column (b)).

Minterms	Binary representation	Minterms	Binary representation
m_0	0000	m_0	0000 ✓
m_1	0001	m_1	0001 ✓
m_2	0010	m_2	0010 ✓
m_3	0011	m_3	0011 ✓
m_{10}	1010	m_{10}	1010 ✓
m_{11}	1011	m_{12}	1100 ✓
m_{12}	1100	m_{11}	1011 ✓
m_{13}	1101	m_{13}	1101 ✓
m_{14}	1110	m_{14}	1110 ✓
m_{15}	1111	m_{15}	1111 ✓

Step 3 : Compare each binary no. with every term in the adjacent next high
category and if they differ only by one position, put a check mark and copy the term
the next column with '–' in the position that they differ.

Step 4 : Apply the same process described in step 3 for the resultant column ar
continue these cycles until a single pass through cycle yields us further elimination
literals.

Minterms	Binary representation		Minterms	Binary representation
0, 1	0 0 0 –	✓	0, 1, 2, 3	0 0 – –
0, 2	0 0 – 0	✓	2, 3, 10, 11	– 0 1 –
1, 3	0 0 – 1	✓	10, 11, 14, 15	1 – 1 –
2, 3	0 0 1 –	✓	12, 13, 14, 15	1 1 – –
2, 10	– 0 1 0	✓		
3, 11	– 0 1 1	✓		
10, 11	1 0 1 –	✓		
10, 14	1 – 1 0	✓		
12, 13	1 1 0 –	✓		
12, 14	1 1 – 0	✓		
11, 15	1 – 1 1	✓		
13, 15	1 1 – 1	✓		
14, 15	1 1 1 –	✓		

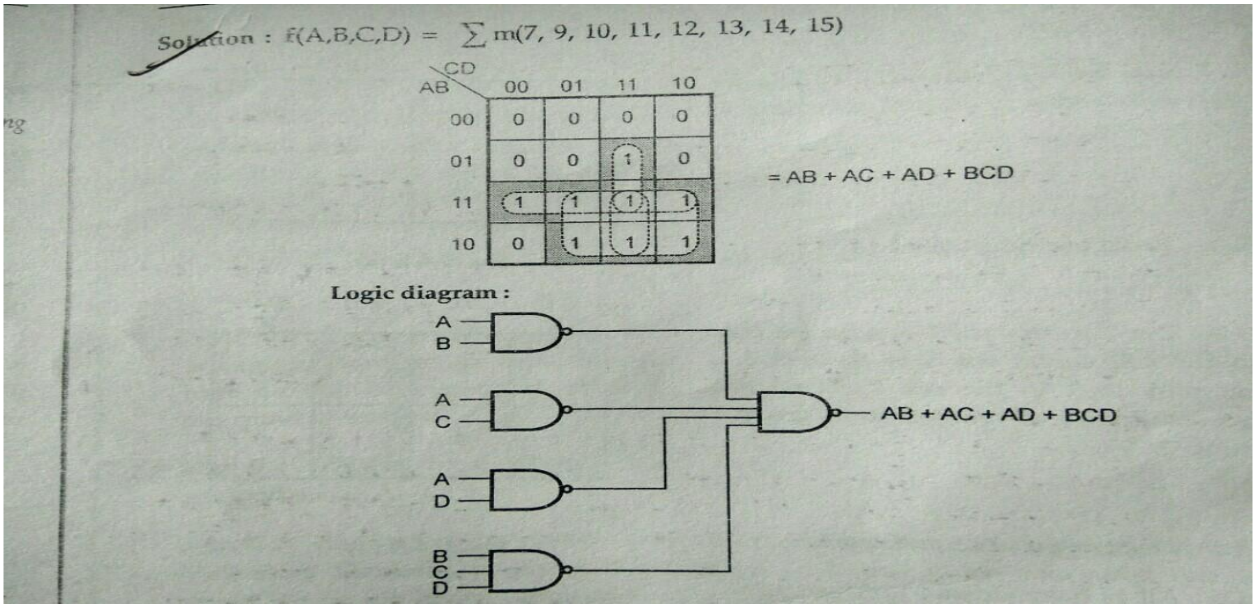
Step 5 : Select the minimum number of prime implicants which must cover all th
minterms.

Prime Implicants		m_0 (col 1)	m_1 (col 2)	m_2 (col 3)	m_3 (col 4)	m_{10} (col 5)	m_{11} (col 6)	m_{12} (col 7)	m_{13} (col 8)	m_{14} (col 9)	m_{16} (col 10)
AB	12, 13, 14, 15 ✓							⊙	⊙	⊙	⊙
AC	10, 11, 14, 15 ✓					⊙	⊙			⊙	⊙
$\overline{B}C$	2, 3, 10, 11			•	•	•	•				
$\overline{A}\,\overline{B}$	0, 1, 2, 3 ✓	⊙	⊙	⊙	⊙						

$$F(A, B, C, D) = \overline{A}\,\overline{B} + AC + AB$$

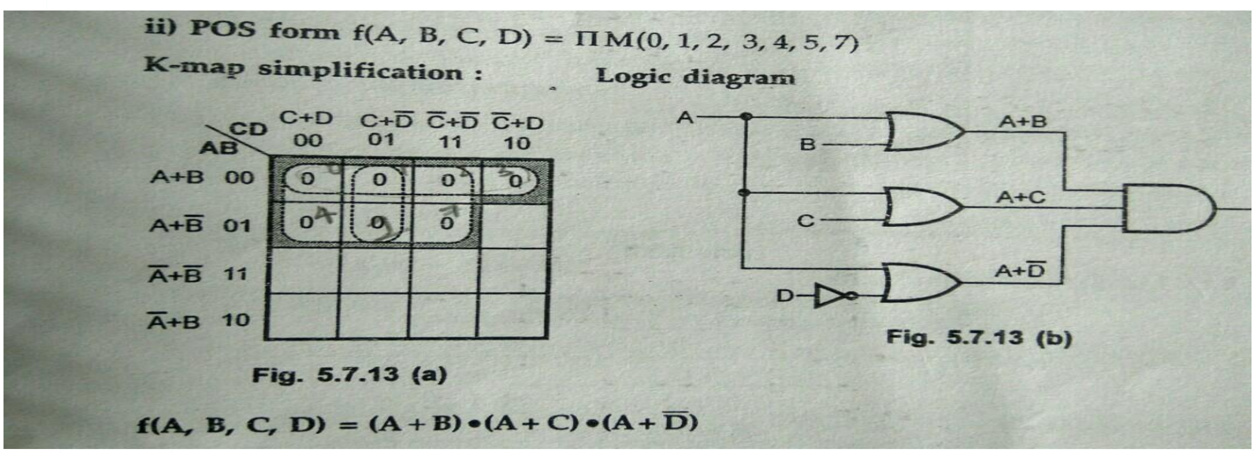
2 A) Obtain the simplified Boolean equation by using K-map method and express it in SOP form. Realize logic circuit by using NAND gates only

F(A,B,C,D)=Σm(7,9,10,11,12,13,14,15).

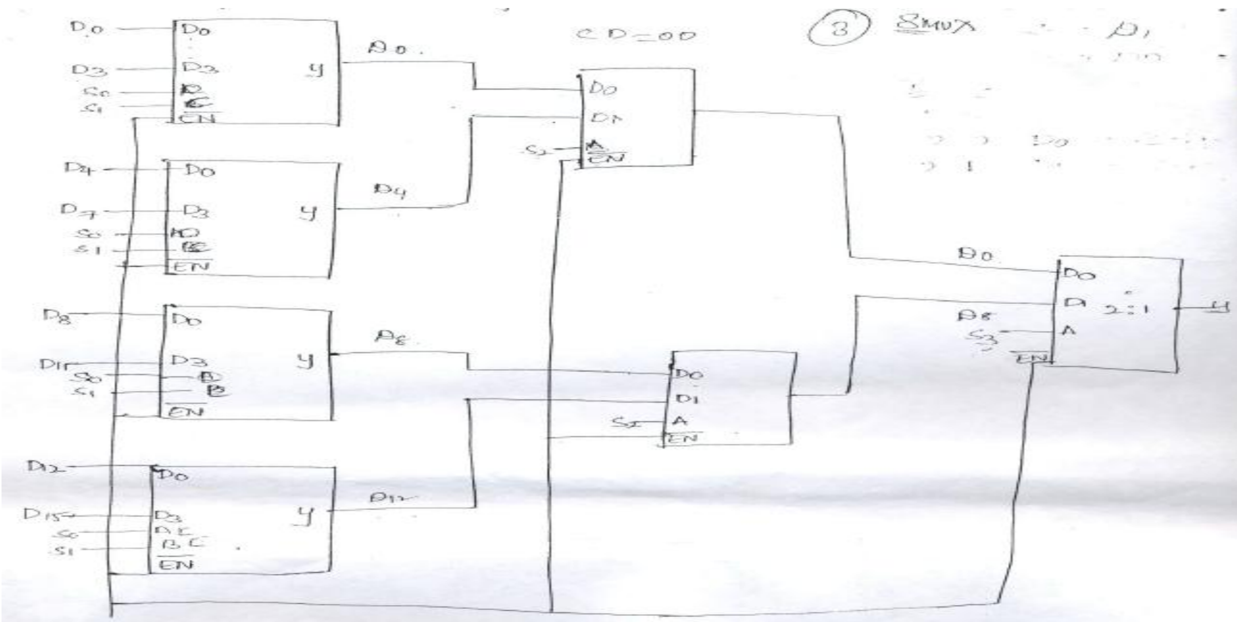


2 B) Find the minimal product using K-map for the Boolean function

F(A,B,C,D)= Σm(6,7,9,10,13)+d(1,4,5,11).



4 (a) Construct a 16:1 multiplexer using 4 to 1 and 2 to 1multiplexers.



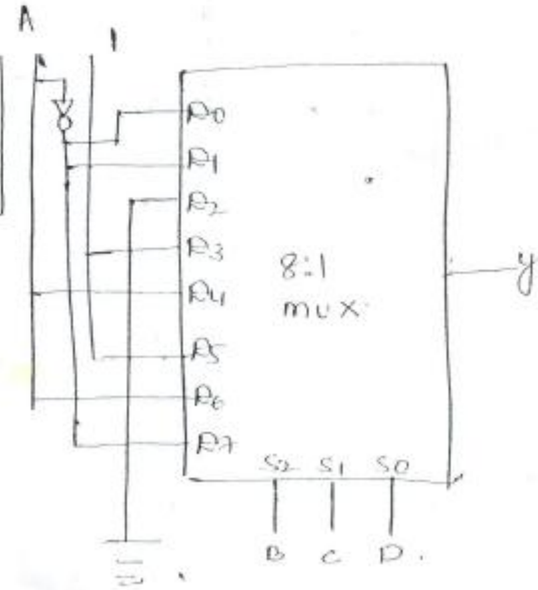
(b) Implement the given Boolean function by using 8:1 multiplexer.

F(A,B,C,D) = $\Sigma m(0,1,3,5,7,11,12,13,14)$

$f(A,B,C,D) = \Sigma(0,1,3,5,7,11,12,13,14)$

using 8:1 mux

	D_0	D_1	D_2	D_3	D_4	D_5	D_6	D_7
$\bar{A}$	0	1	2	3	4	5	6	7
A	8	9	10	11	12	13	14	15
	$\bar{A}$	$\bar{A}$	0	1	A	1	A	$\bar{A}$



5 Design a 4 bit priority Encoder.

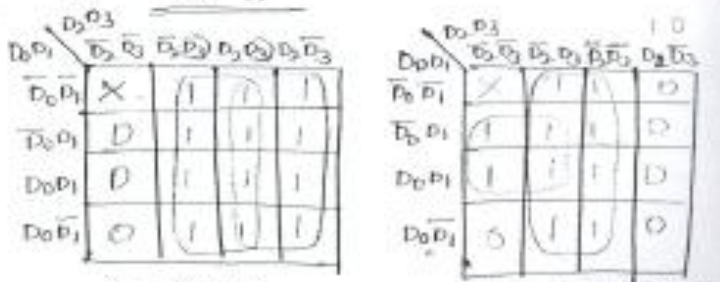
Priority Encoders:- The encoders will operate correctly provided that one & only one decimal i/p is High at any given time.

- Priority Encoder is a logic ckt that responds to just one i/p in accordance with some priority s/m, among all those that may simultaneously go high.

4 i/p priority Encoder

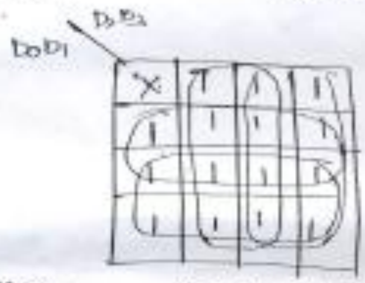
Inputs				o/p's		
D_0	D_1	D_2	D_3	A	B	V
0	0	0	0	x	x	0
1	0	0	0	0	0	1
x	1	0	0	0	1	1
x	x	1	0	1	0	1
x	x	x	1	1	1	1

For A



$A = D_3 + D_2$

$B = D_3 + D_1\overline{D_2}$



$Y = D_0 + D_1 + D_2 + D_3$

From the T.T we get

$A = D_3 + D_2$

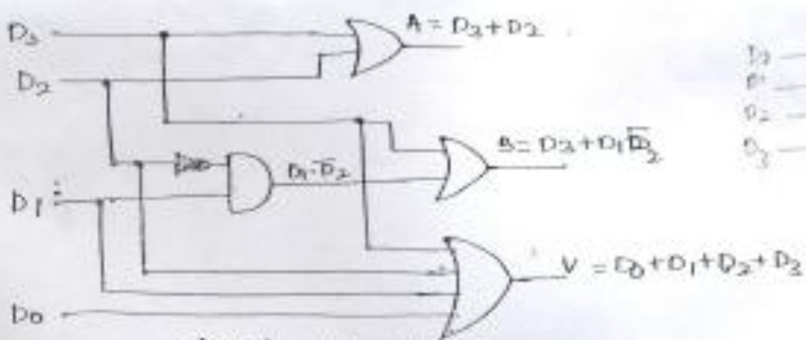
$B = D_3 + D_1\overline{D_2}\,\overline{D_3}$

$Y = D_0 + D_1 + D_2 + D_3$

There are 16 possible i/p combination.

$A = \Sigma m(1,2,3,5,6,7,9,10,11,13,14,15)$

$B = \Sigma m(1,3,4,5,7,9,11,12,13,15)$



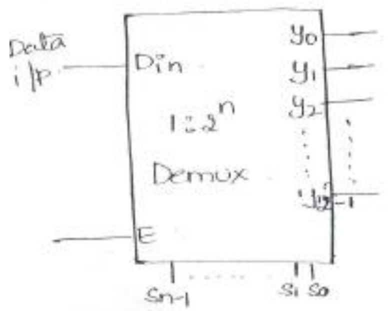
4 bit priority Encoder

$4 = 2^{2}$ → o/p lines of Encoder

6 (a) Explain the working of 1:4 demultiplexers.

Demultiplexers :- Is a ckt that receives a single i/p & distributes it over several o/p's. (i.e) 2^n possible o/p lines. (1 to N).

- The selection of specific o/p line is controlled by the values of n selection lines.



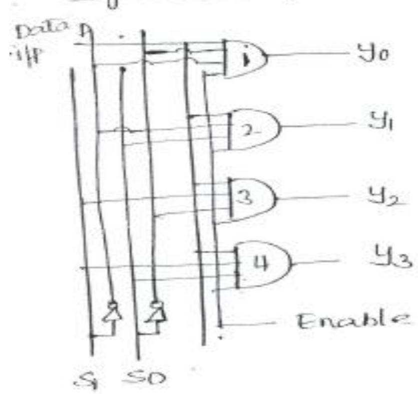
Block diagram.

→ Types of Demultiplexer

i] 1:4 Demux :- The single i/p variable Din has a path to all four o/p's, but the i/p info is directed to only one of the o/p lines depending on select i/p's.

Enable	S_1	S_0	Din	y_0	y_1	y_2	y_3
0	X	X	X	0	0	0	0
1	0	0	0	0	0	0	0
1	0	0	1	1	0	0	0
1	0	1	0	0	0	0	0
1	0	1	1	0	1	0	0
1	1	0	0	0	0	0	0
1	1	0	1	0	0	1	0
1	1	1	0	0	0	0	0
1	1	1	1	0	0	0	1

logic diagram



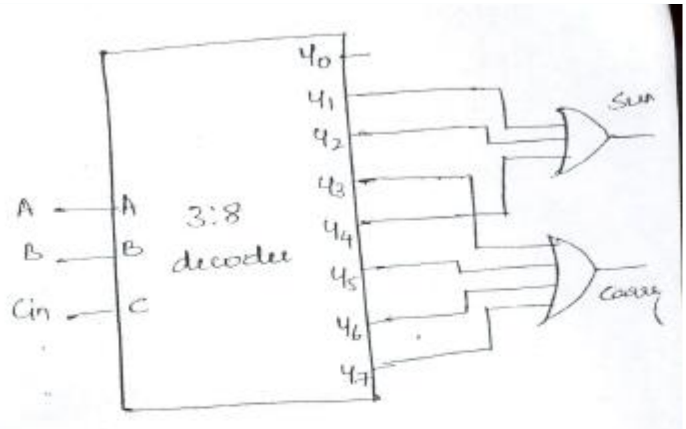
Select lines		o/p			
S_1	S_0	y_0	y_1	y_2	y_3
0	0	1	0	0	0
0	1	0	1	0	0
1	0	0	0	1	0
1	1	0	0	0	1

func table 4:1 MUX

(b) Implement a full adder using 3 to 8 decoders

Soln

I/p's			O/p's	
A	B	Cin	Carry	Sum
0	0	0	0	0
0	0	1	0	1
0	1	0	0	1
0	1	1	1	0
1	0	0	0	1
1	0	1	1	0
1	1	0	1	0
1	1	1	1	1



3 (a) Design a 2 bit magnitude comparator using gates.

2 bit Magnitude Comparator

Let the two 2bit no's be $A = A_1A_0$ & $B = B_1B_0$

1. If $A_1 = 1$ & $B_1 = 0$, then $A > B$ (or)
2. If A_1 & B_1 coincides & $A_0 = 1$ & $B_0 = 0$ then $A > B$.

$A > B : G = A_1\overline{B_1} + (A_1 \odot B_1) A_0 \cdot \overline{B_0}$.

3. If $A_1 = 0$ & $B_1 = 1$ then $A < B$ (or).

If A_1 & B_1 coincides & $A_0 = 0$ & $B_0 = 1$ then $A < B$.

4. If A_1 & B_1 coincides & if A_0 & B_0 coincides $A = B$:

$E = (A_1 \odot B_1)(A_0 \odot B_0)$.

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

$A>B$

$A_1A_0 \backslash B_1B_0$	00	01	11	10
00	0	0	0	0
01	1	0	0	0
11	1	1	0	1
10	1	1	0	0

$A>B = A_1\overline{B_1} + A_0\overline{B_1}\,\overline{B_0} +$
$A_1A_0\overline{B_0}$

$A=B$

$A_1A_0 \backslash B_1B_0$	00	01	11	10
00	1	0	0	0
01	0	1	0	0
11	0	0	1	0
10	0	0	0	1

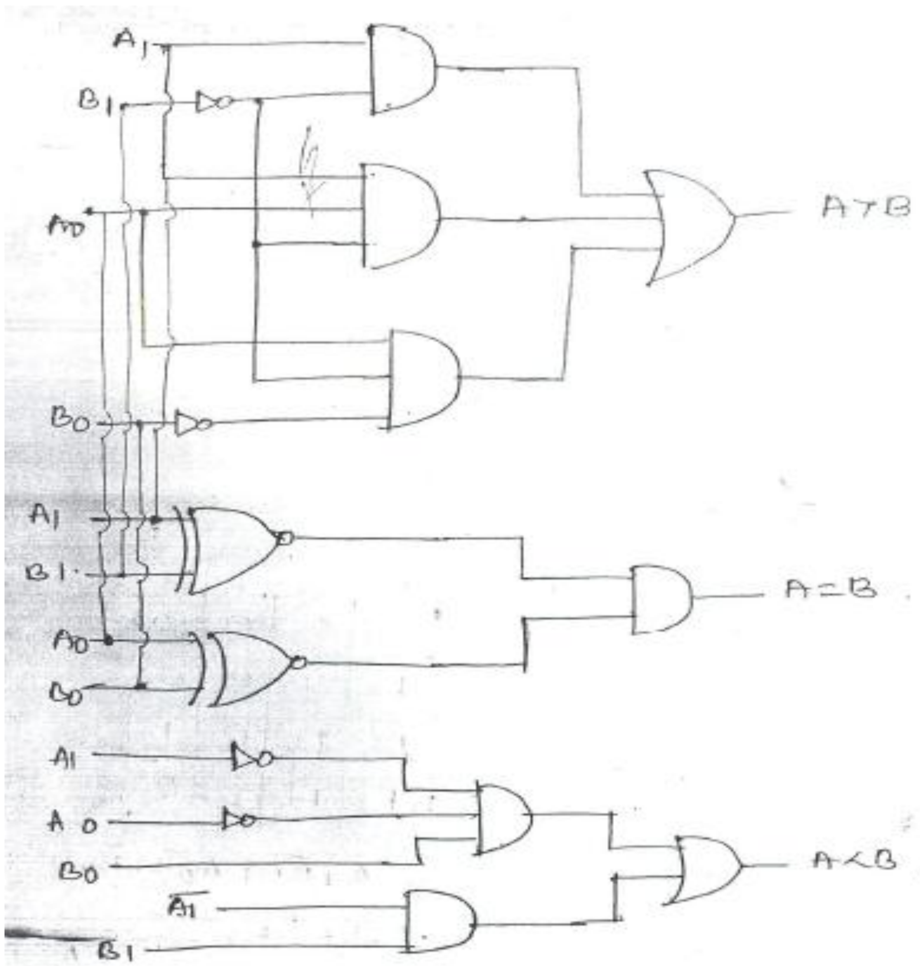
$A=B : \overline{A_1}\,\overline{A_0}\,\overline{B_1}\,\overline{B_0} + \overline{A_1}A_0\overline{B_1}$
$+ A_1A_0B_1B_0 + A_1\overline{A_0}B_1\overline{B_0}$
$= \overline{A_1}\cdot\overline{B_1}(\overline{A_0}\,\overline{B_0} + A_0B_0) +$
$A_1\cdot B_1(A_0B_0 + \overline{A_0}\cdot\overline{B_0})$
$= (A_0 \odot B_0)(A_1 \odot B_1)$

$A<B$

$A_1A_0 \backslash B_1B_0$	00	01	11	10
00	0	1	1	1
01	0	0	1	1
11	0	0	0	0
10	0	0	1	0

A_1 A_0
A_3 A
1 1

$A<B = \overline{A_1}B_1 + \overline{A_0}B_1B_0 + \overline{A_1}\,\overline{A_0}B_0$



7. Implement a PLA circuit having 3 inputs, 3 product terms and two outputs for the given Boolean functions f1=Σm(1,3,5) and f2=Σm(5,6,7).

(4) A Combinational ckt is defined by the func.

$F_1 = \Sigma m(1, 3, 5)$

$F_2 = \Sigma m(5, 6, 7)$

Implement the ckt with a PLA having $3 \times 3 \times 2$ o/p'

Sol^n Kmap simplification.

For F1

A \ BC	$\bar{B}\bar{C}$	$\bar{B}C$	BC	$B\bar{C}$
$\bar{A}$		1	1	
A		1		

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

For F2

A \ BC	$\bar{B}\bar{C}$	$\bar{B}C$	BC	$B\bar{C}$
$\bar{A}$				
A		1	1	1

$f_2 = AC + AB$.

∴ we get 4 product terms it cannot be used.

examine grouping the maxterms

For F1

A \ BC	$\bar{B}\bar{C}$	$\bar{B}C$	BC	$B\bar{C}$
$\bar{A}$	0			0
A	0		0	0

$\bar{f_1} = \bar{C} + AB$

$\bar{f_1} = \boxed{\bar{B}\bar{C}} + B\bar{C} + AB$

A \ BC	$\bar{B}\bar{C}$	$\bar{B}C$	BC	$B\bar{C}$
$\bar{A}$	0	0	0	0
A	0			

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

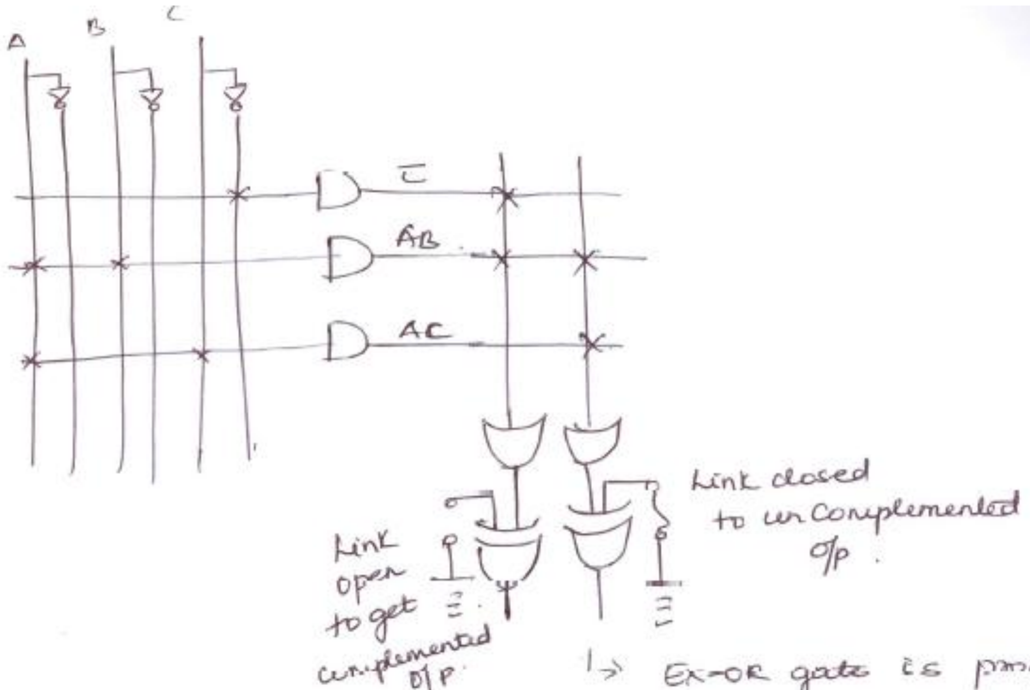
product term.

Fl = ___

funcs $\bar{F_1}$ & F_2 have one common product term.

having 3 product terms.

P.T	i/p's A	B	C	o/p's F_1	F_2
$\bar{C}$	—	—	0	1	—
AB	1	1	—	1	1
AC	1	—	1	—	1
				C	T



→ Ex-OR gate is programmed to invert the fun^c