$Y_2 = BC + AC.$

(1) Programmable Logic Array:- [Both AND and OR gates are programmable]

Implement the following functions using PLA.

$F_1 = \Sigma m(1, 2, 4, 6)$, $F_2 = \Sigma m(0, 1, 6, 7)$,

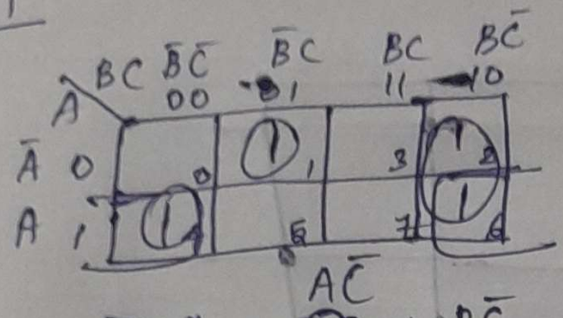
$F_3 = \Sigma m(2, 6)$;

Step 1 → Truth table:

A	B	C	F_1	F_2	F_3
0	0	0	0	1	0
0	0	1	1	1	0
0	1	0	1	0	1
0	1	1	0	0	0
1	0	0	1	0	0
1	0	1	0	0	0
1	1	0	1	1	1
1	1	1	0	1	0

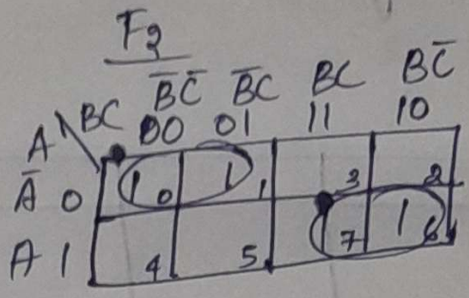
Step 2 → K-map

F1



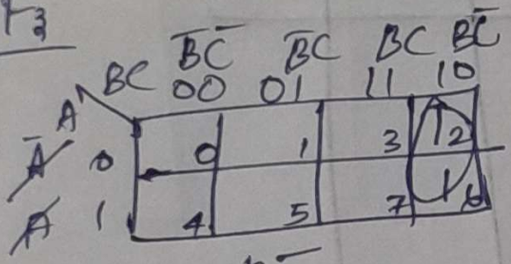
$F_1 = \bar{A}B\bar{C} + A\bar{C} + B\bar{C}$

F2



$F_2 = \bar{A}\bar{B} + AB$

F3



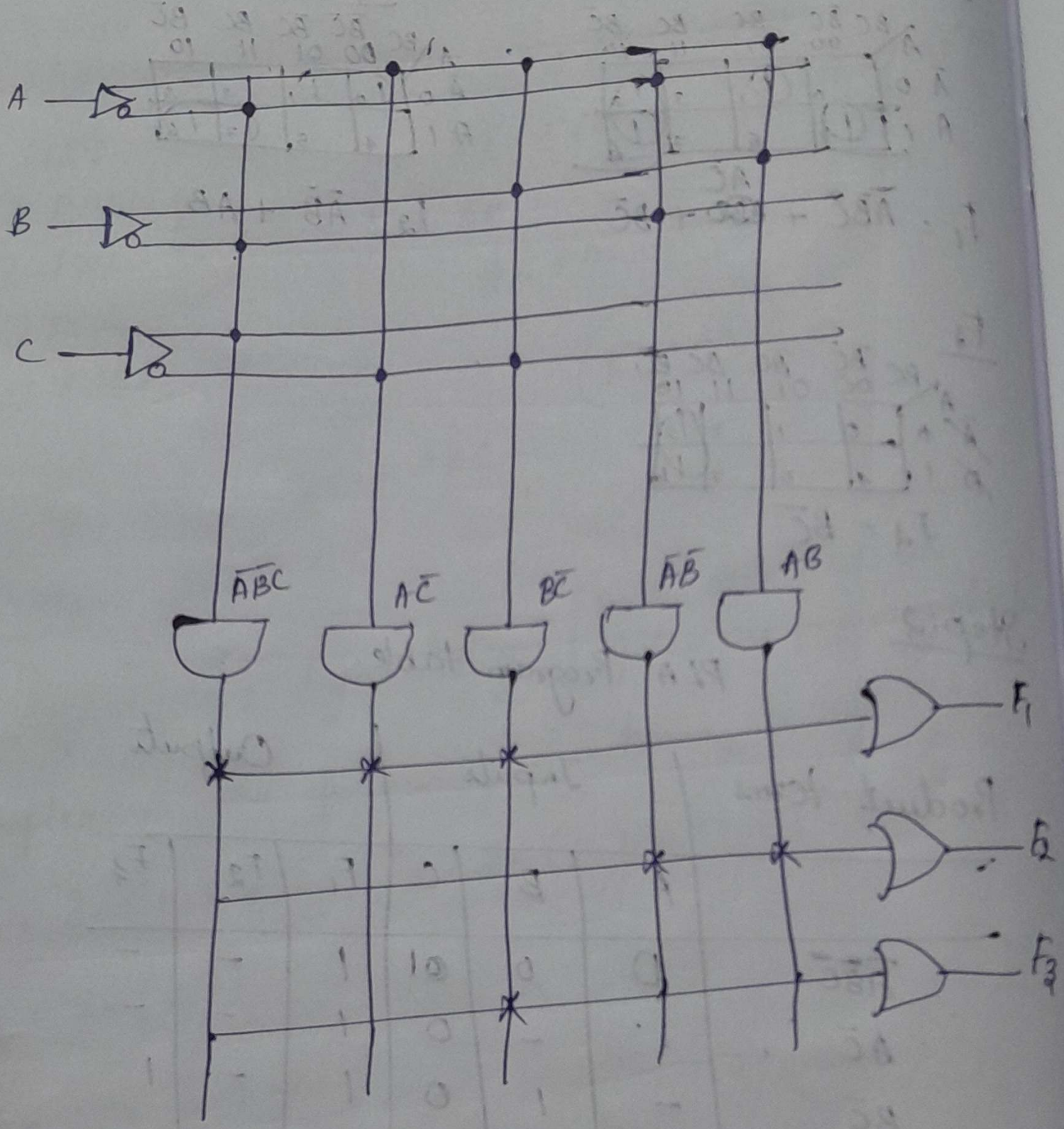
$F_3 = B\bar{C}$

Step: 2

PLA Program table.

Product terms	Inputs A	Inputs B	Inputs C	Outputs F_1	Outputs F_2	Outputs F_3
$\bar{A}B\bar{C}$	0	0	1	1	–	–
$A\bar{C}$	1	–	0	1	–	1
$B\bar{C}$	–	1	0	1	–	–
$\bar{A}\bar{B}$	0	0	–	–	1	–
AB	1	1	–	–	1	–

Logic diagram:



(2) Programmable Array Logic: [AND → Programmable, OR → fixed]

Implement the following function using PAL.

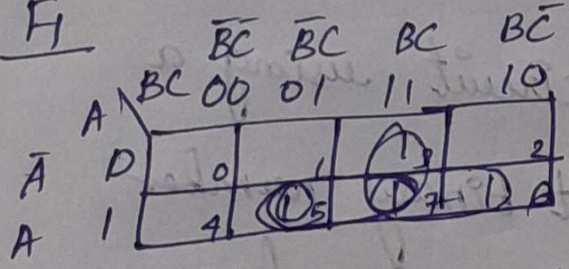
$F_1 = \Sigma m(3, 5, 6, 7)$, $F_2 = \Sigma m(0, 1, 6, 7)$

Truth table:

Inputs			Outputs	
A	B	C	F_1	F_2
0	0	0	0	1
0	0	1	0	1
0	1	0	0	0
0	1	1	1	0
1	0	0	0	0
1	0	1	1	0
1	1	0	1	1
1	1	1	1	1

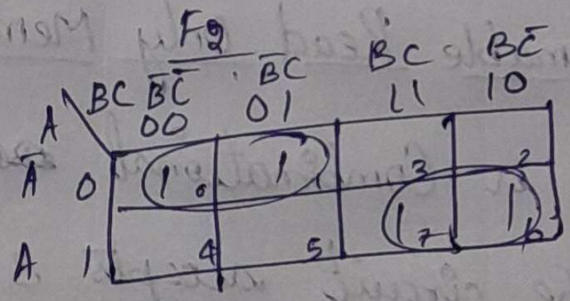
K-map:

F_1



$F_1 = AC + BC + A\bar{B}$

F_2

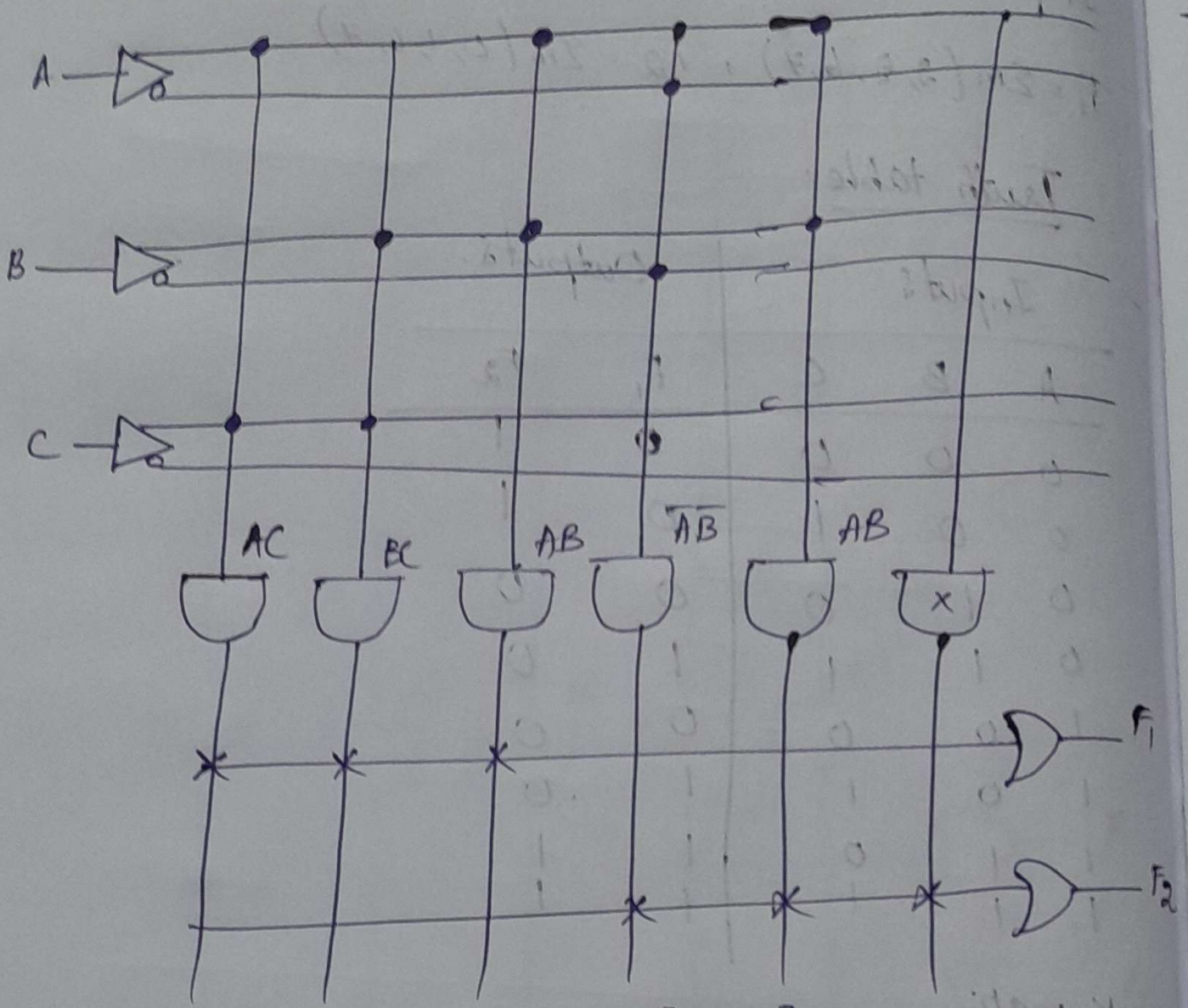


$F_2 = \bar{A}\bar{B} + AB$

Program Table:

Product terms	Inputs			Outputs	
	A	B	C	F_1	F_2
	1	–	1	1	–
AC	–	1	1	1	–
BC	1	1	–	1	1
AB	1	1	–	1	1
$\bar{A}\bar{B}$	0	0	–	0	1

Logic diagram:



③ Programmable Read Only Memory [PROM]

Design a Combinational ~~ckt~~ circuit using a PROM, the circuit accepts 3 bit binary number and generates its equivalent Excess-3-code

Truth table:

Inputs			Outputs			
B_2	B_1	B_0	E_3	E_2	E_1	E_0
0	0	0	0	0	1	1
0	0	1	0	1	0	0
0	1	0	0	1	0	1
0	1	1	0	1	1	0
1	0	0	0	1	1	1
1	0	1	1	0	0	0
1	1	0	1	0	0	1
1	1	1	1	0	1	0

(B) $E_0 = \Sigma m(0, 2, 4, 6)$

$E_1 = \Sigma m(0, 3, 4, 7)$

$E_2 = \Sigma m(1, 2, 3, 4)$

$E_3 = \Sigma m(5, 6, 7)$

Logic diagram:-