SRM INSTITUTE OF SCIENCE AND TECHNOLOGY
TIRUCHIRAPPALLI CAMPUS

E. G. Pradeep
RAa a11 00A0500a6

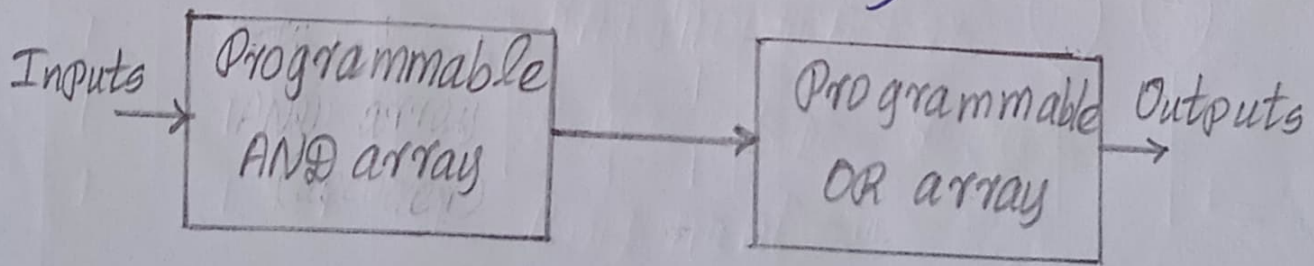
Digital Logic Devices
Assignment - III
B. TECH. ECE - II - Year

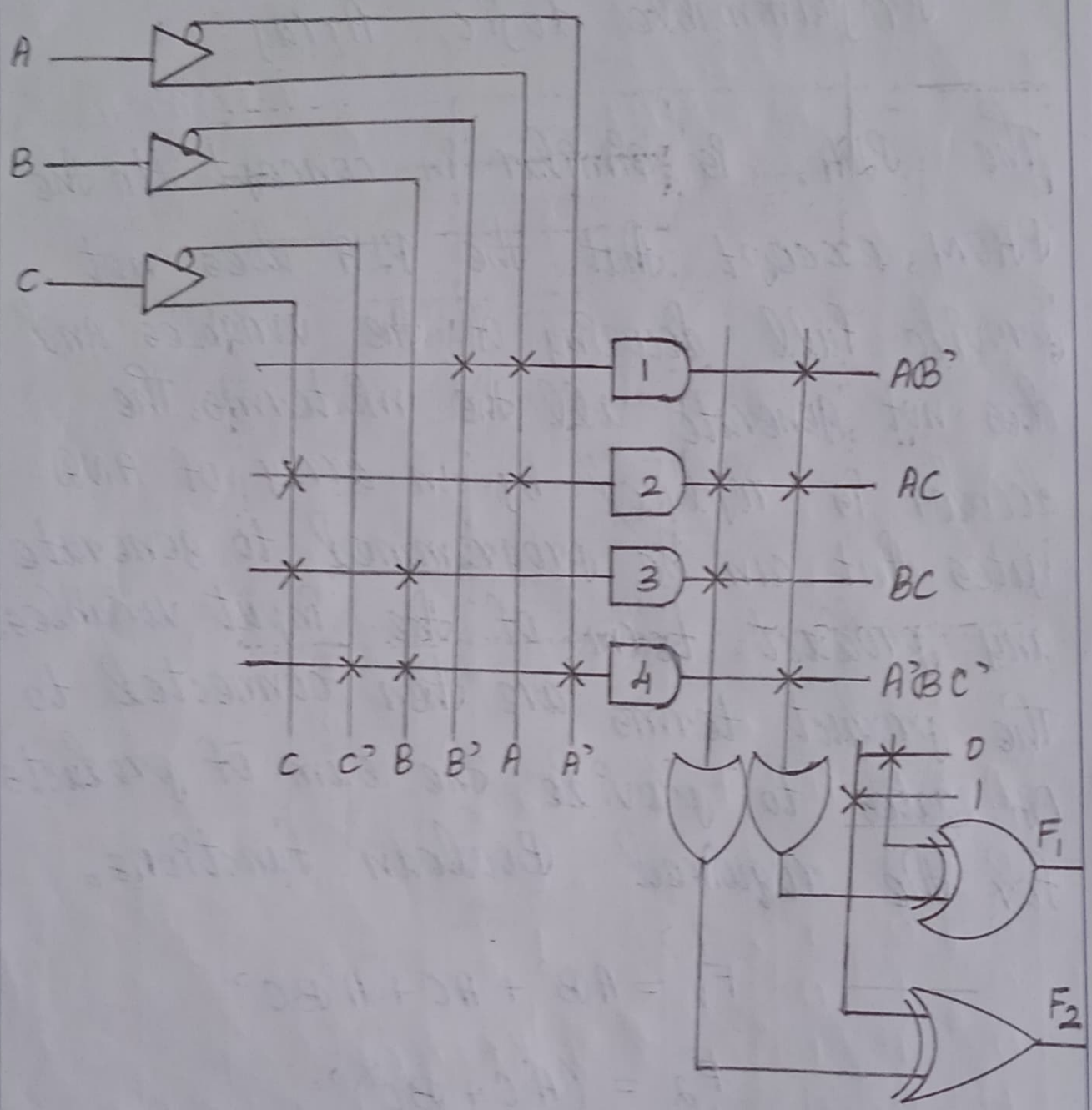
Programmable Logic Array

The PLA is similar in concept to the PROM, except that the PLA does not provide full decoding of the variables and does not generate all the minterms. The decoder is replaced by an array of AND gates that can be programmed to generate any product term of the input variables. The product terms are then connected to OR gates to provide the sum of products for the required Boolean functions.

$$F_1 = AB' + AC + A'BC'$$

$$F_2 = (AC + BC)'$$




A
B
C
1
2
3
4
AB'
AC
BC
A'B C'
C C' B B' A A'
D
1
F1
F2

PLA Programming Table

	Product Terms	Inputs			Outputs (T)	Outputs (C)
		A	B	C	F_1	F_2
AB'	1	1	0	–	1	–
AC	2	1	–	1	1	1
BC	3	–	1	1	–	1
$A'BC'$	4	0	1	0	1	–

Example:

Implement the following two boolean functions with a PLA:

$$F_1(A, B, C) = \Sigma(0, 1, 2, 4)$$

$$F_2(A, B, C) = \Sigma(0, 5, 6, 7)$$

The combination that gives the minimum number of product terms is

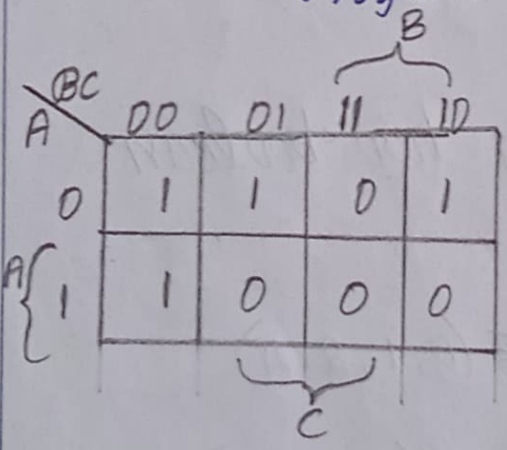
$$F_1 = (AB + AC + BC)'$$

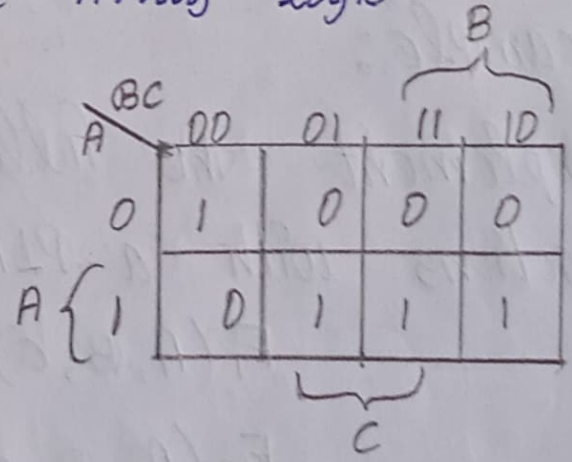
$$F_2 = AB + AC + A'B'C'$$

PLA Programming table

	Product term	Inputs A B C	Outputs (C) (F) F_1	F_2
AB'	1	1 0 –	1	–
AC	2	1 – 1	1	1
BC	3	– 1 1	–	1
$A'BC'$	4	0 1 0	1	–

Programmable Array logic





Programmable logic device

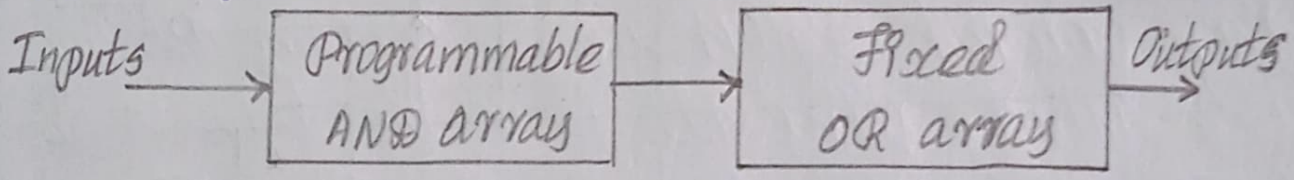
The PAL is a programmable logic devices with a fixed OR array and a programmable AND array. Because only the AND gates are programmable, the PAL is easier to program than, but is not as flexible as, the PLA. The logic configuration of a typical PAL with four inputs and four outputs. Each input has a buffer - Inverter gate, and each output is generated by a fixed OR gate. There are four sections in the unit, each composed of an AND-OR array that is three wide, the term used to indicate that there are three programmable AND gates in each section and one fixed OR gate. Each AND gate has 10 programmable input connections, The horizontal line symbolizes the multiple-input configuration of the AND gate.

One of the outputs is connected to a buffer-inverted gate and then fed back into two inputs of the AND gates.

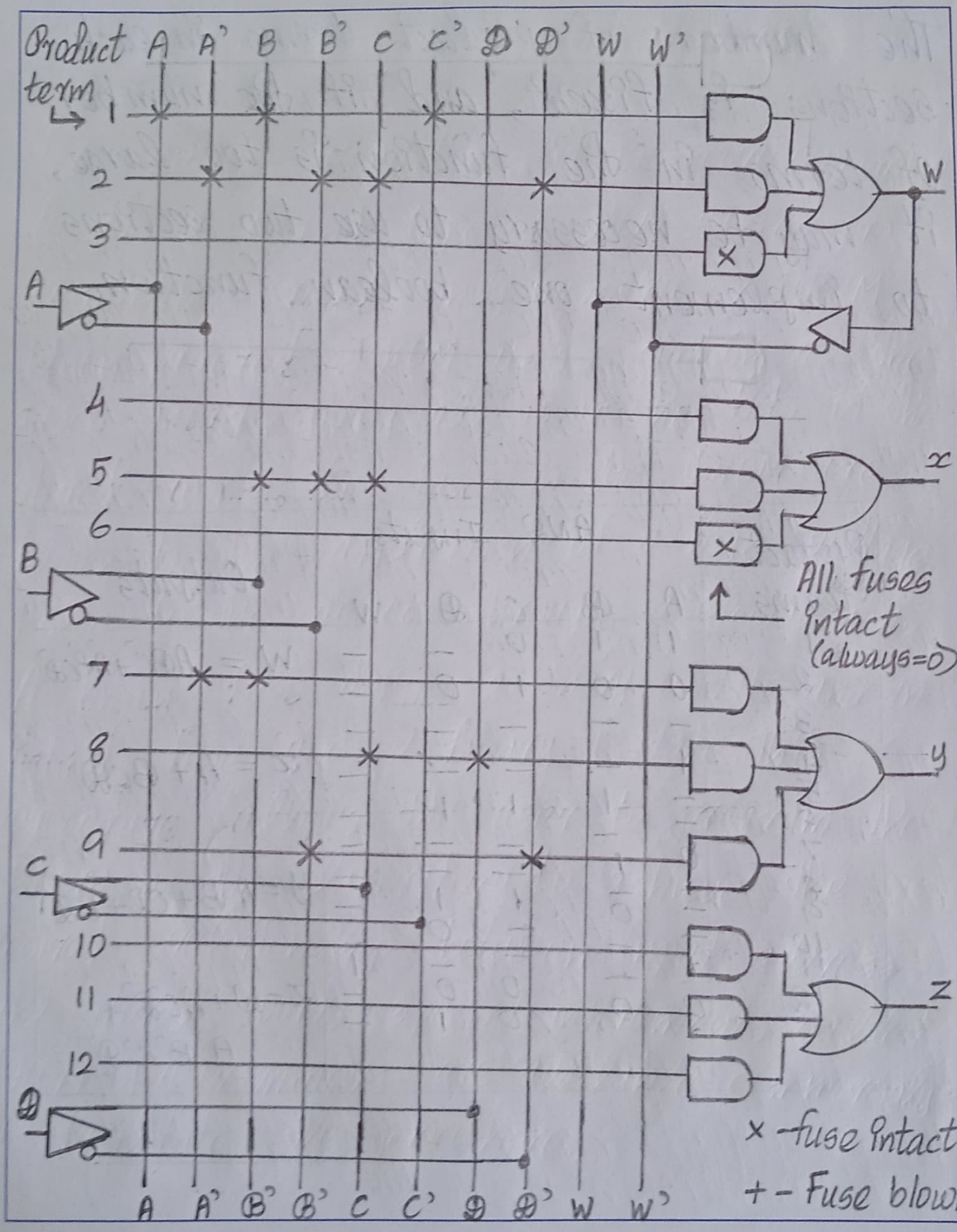
Commercial PAL devices contain more gates than the one. A typical PAL integrated circuit may have eight inputs, eight outputs, and eight sections, each consisting of an eight-wide AND-OR array. The output terminals are sometimes driven by three-state buffers or inverters.

In designing with a PAL, the boolean functions must be simplified to fit into each section. Unlike the situation with a PLA, a product term cannot be shared among two or more OR gates. Therefore, each function can be simplified by itself, without regard to common terms.

The number of product terms in each sections is fixed, and if the number of terms in the function is too large, it may be necessary to use two sections to implement one boolean function.



Product terms	AND Inputs A	B	C	D	W	Outputs
1	1	1	0	—	—	$W = ABC' + A'B'CD'$
2	0	0	1	0	—	
3	—	—	—	—	—	
4	1	—	—	—	—	$x = A + BCD$
5	—	1	1	1	—	
6	—	—	—	—	—	
7	0	1	—	—	—	
8	—	—	1	1	—	$y = A'B + CD + B'D'$
9	—	0	—	0	—	
10	—	—	—	—	1	
11	1	—	0	0	—	$z = W + AC'D' + A'B'C'D$
12	0	0	0	1	—	

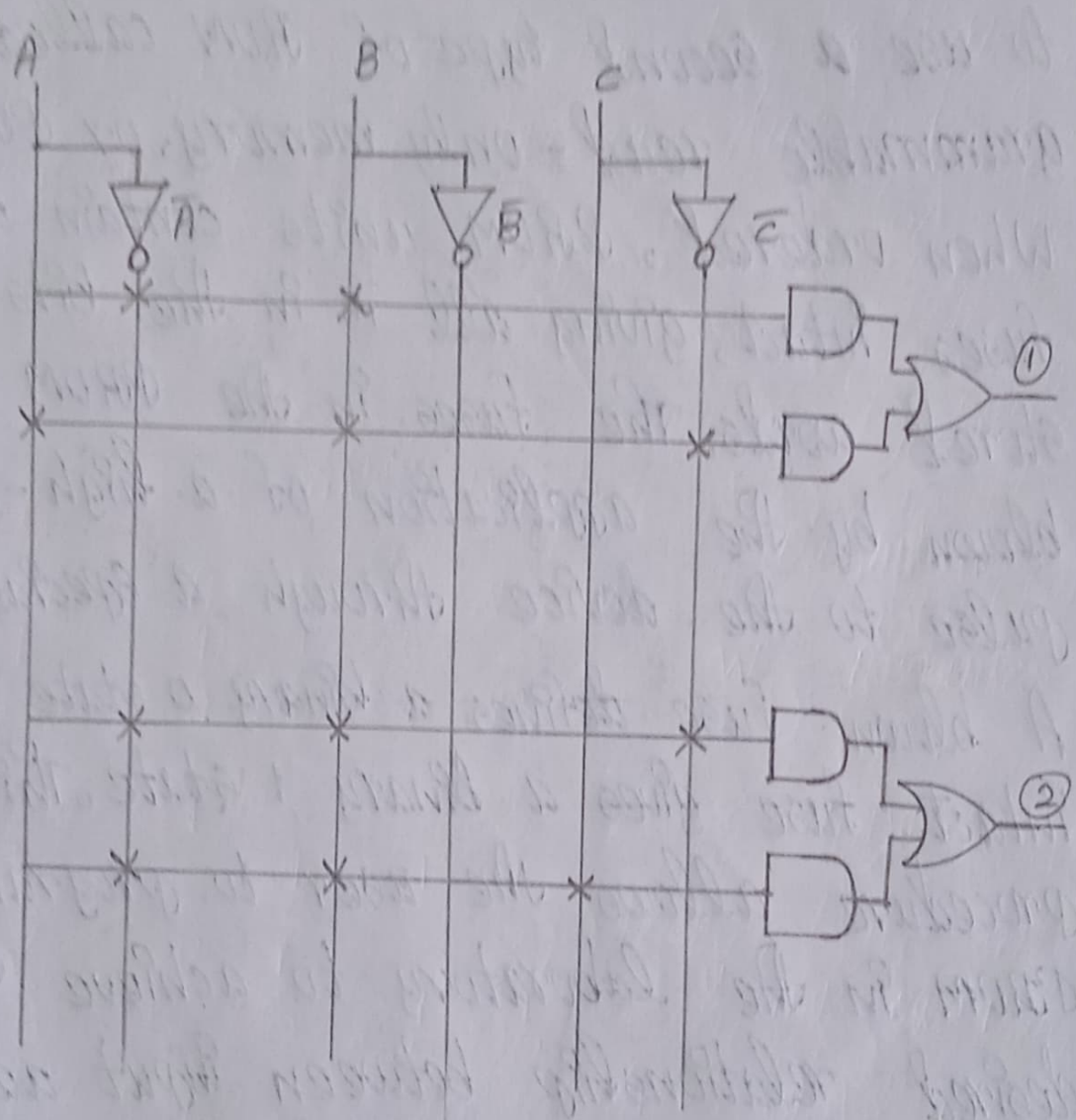

Product term
A A' B B' C C' D D' W W'
1
2
3
4
5
6
7
8
9
10
11
12
W
x
y
z
All fuses Intact (always=0)
x - fuse Intact
+ - Fuse blown

Programmable Read-only Memory

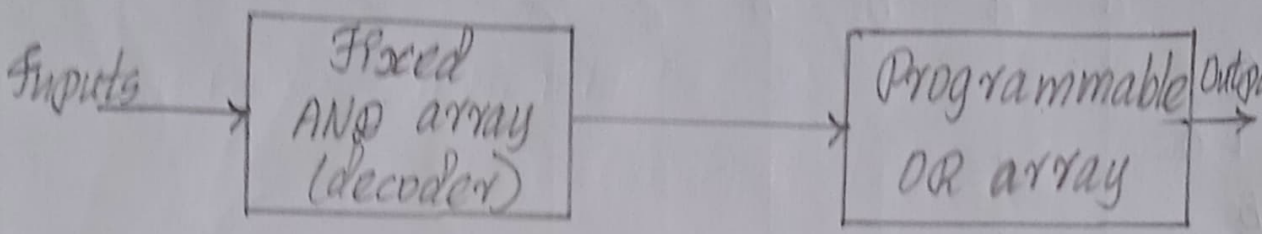
For small quantities, it is more economical to use a second type of ROM called programmable read-only memory, or PROM. When ordered, PROM units contain all the fuses intact, giving all 1's in the bits of the stored words. The fuses in the PROM are blown by the application of a high-voltage pulse to the device through a special pin. A blown fuse defines a binary 0 state and an intact fuse gives a binary 1 state. This procedure allows the user to program the PROM in the laboratory to achieve the desired relationship between input address and stored words. Special instruments called PROM programmers are available commercially to facilitate the procedure.

Example:

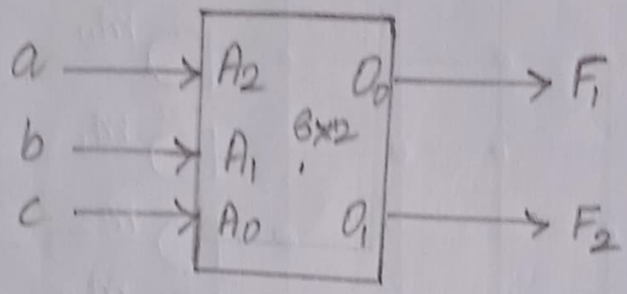
1) $Y = \overline{A}B + AB\overline{C}\,,\ \overline{A}B\overline{C} + \overline{A}BC$



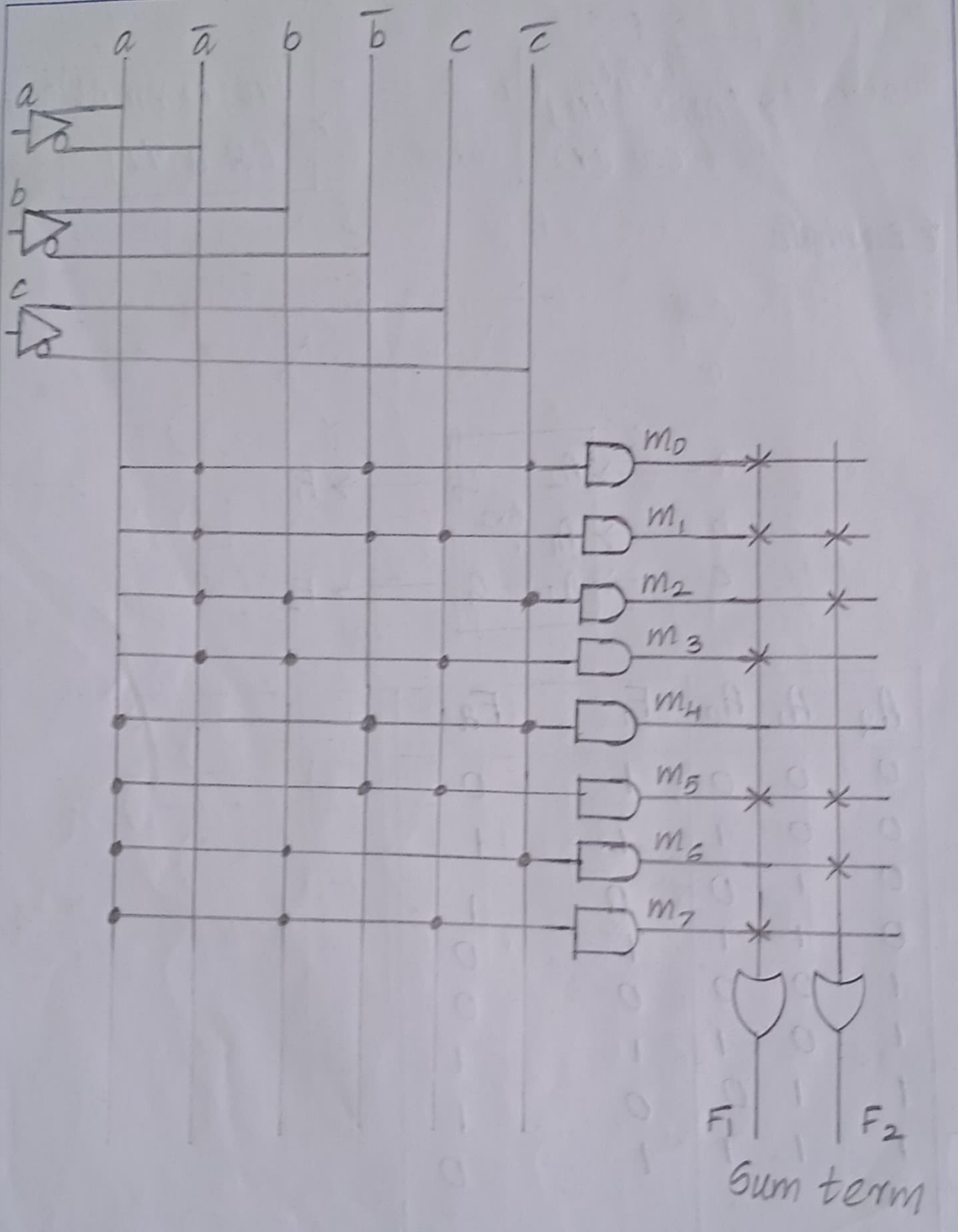
$Y = \overline{A}B + AB\overline{C}\,,\ \overline{A}B\overline{C} + \overline{A}BC$



Example



A_2	A_1	A_0	F_1	F_2
0	0	0	1	0
0	0	1	1	1
0	1	0	0	1
0	1	1	1	0
1	0	0	0	0
1	0	1	1	1
1	1	0	0	1
1	1	1	1	0


a
$\bar{a}$
b
$\bar{b}$
c
$\bar{c}$
a
b
c
m_0
m_1
m_2
m_3
m_4
m_5
m_6
m_7
F_1
F_2
Sum term