PLA

Sparsh Mittal

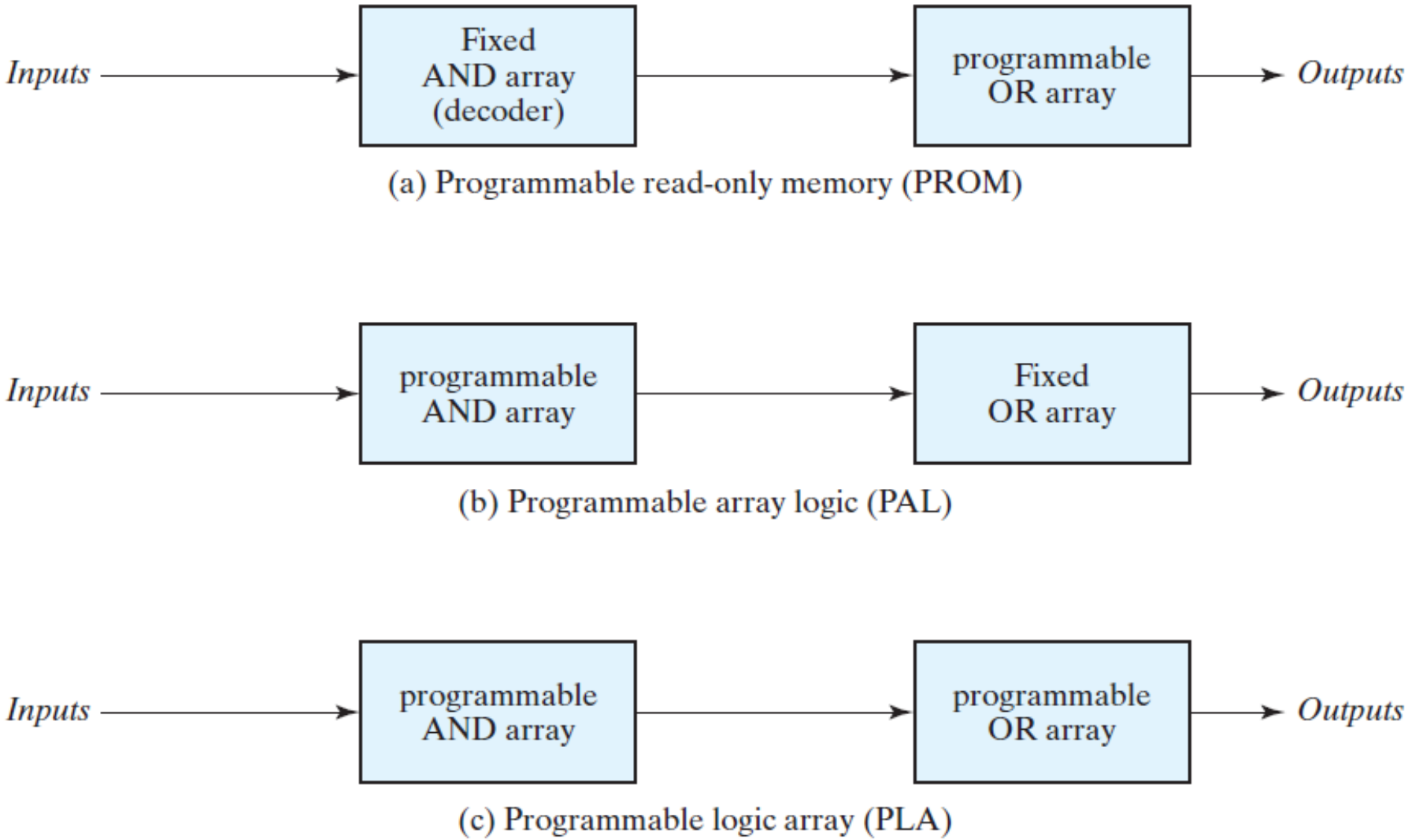


FIGURE 7.13
Basic configuration of three PLDs

PLA: Intro

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.

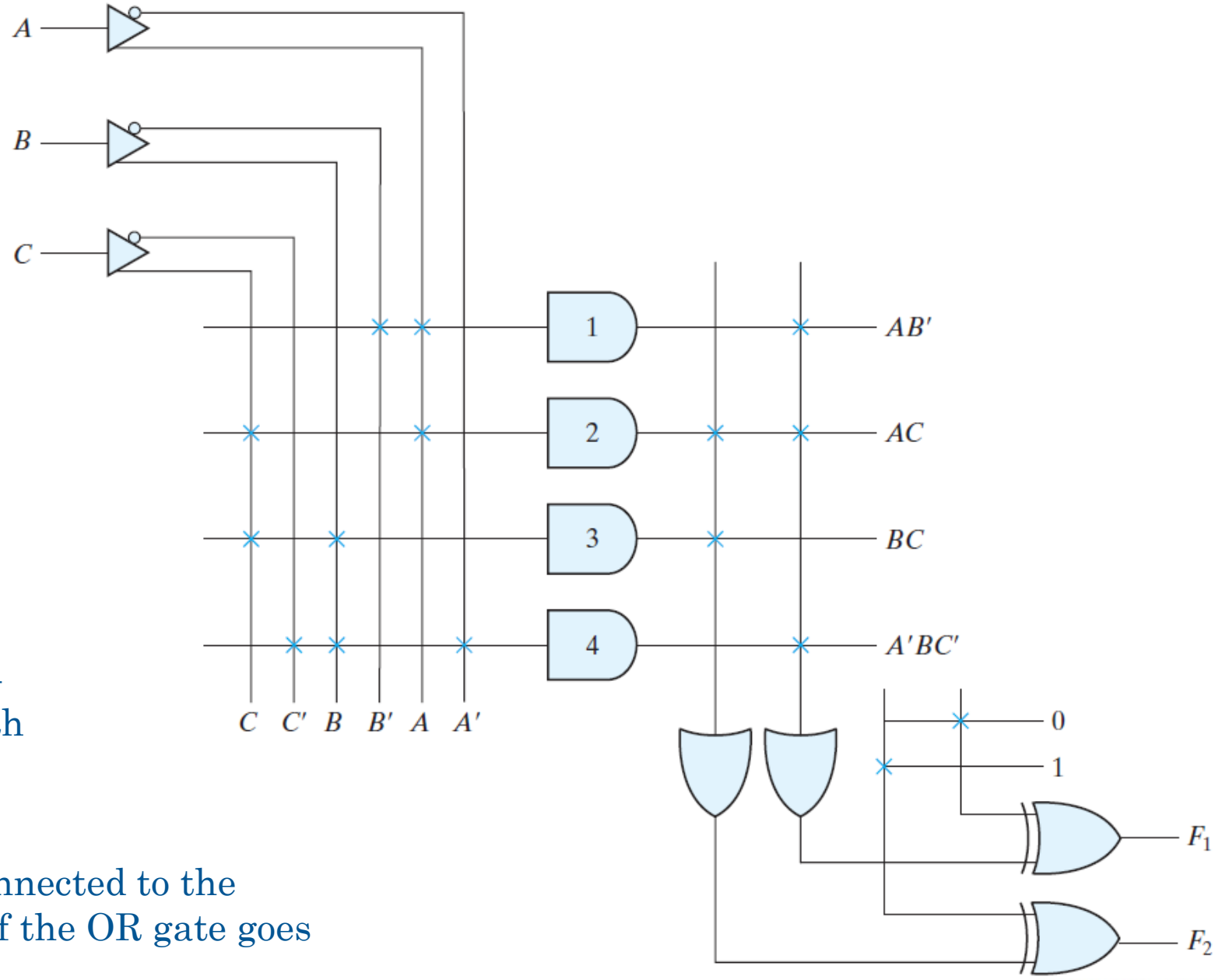
The product terms are then connected to OR gates to provide the sum of products for the required Boolean functions.

Example: a PLA with three inputs and two outputs

We want to implement:

$$F_1 = AB' + AC + A'BC'$$
$$F_2 = (AC + BC)'$$

$$F_1 = AB' + AC + A'BC'$$
$$F_2 = (AC + BC)'$$



PLA with three inputs, four product terms, and two outputs

Each input goes through a buffer–inverter combination, that has both the true and complement outputs.

The outputs of the AND gates are connected to the inputs of each OR gate. The output of the OR gate goes to an XOR gate

Fuse map of PLA is shown in tabular form

The table consists of three sections.
1. list the product terms numerically.
2. specify the required paths between inputs and AND gates.
3. specify the paths between the AND and OR gates.

For each output variable, we may have a T (for true) or C (for complement) for programming the XOR gate.
The product terms listed on the left are not part of the table; they are included for reference only.

$$F_1 = AB' + AC + A'BC'$$
$$F_2 = (AC + BC)'$$

Table 7.5
PLA Programming Table

		Inputs			Outputs (T)	(C)
	Product Term	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	—

1, 0 or dash

For each product term, the inputs are marked with 1, 0, or — (dash).

If a variable in the product term appears in the form in which it is true, the corresponding input variable is marked with a 1.

If it appears complemented, the corresponding input variable is marked with a 0.

If the variable is absent from the product term, it is marked with a dash.

General guidelines

In implementing a combinational circuit with a PLA, careful investigation must be undertaken to reduce the number of distinct product terms, since a PLA has a finite number of AND gates.

This can be done by simplifying each Boolean function to a minimum number of terms.

Both the true value and the complement of each function should be simplified to see which one can be expressed with fewer product terms and which one provides product terms that are common to other functions.

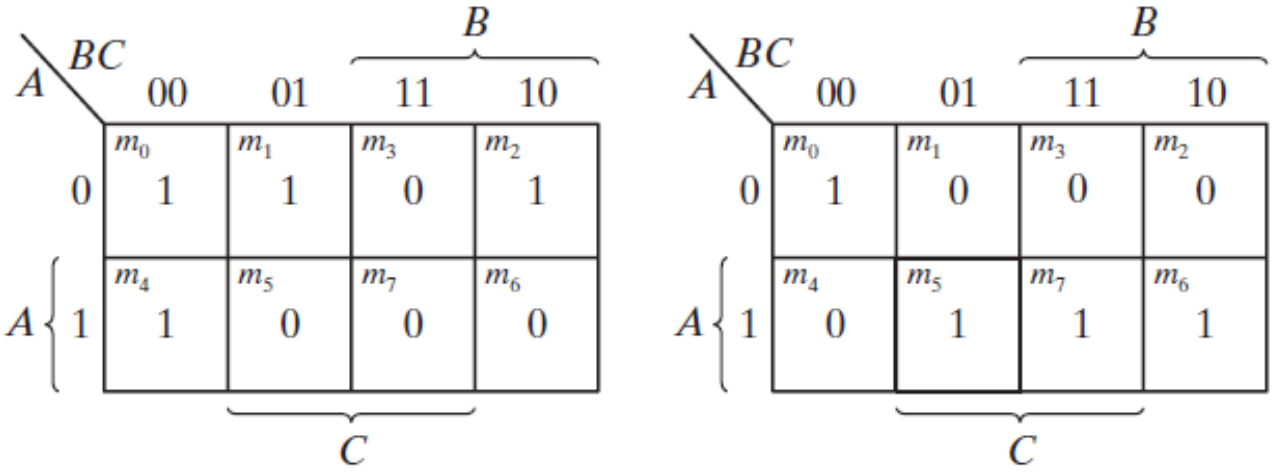
Example 7.2

Implement the following two Boolean functions with a PLA:

$F1(A, B, C) = \sum(0, 1, 2, 4)$

$F2(A, B, C) = \sum(0, 5, 6, 7)$

Solve it as follows:



Example 7.2

Both the true value and the complement of the functions are simplified into sum-of-products form.

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_1	(T) F_2
AB	1	1	1	–	1	1
AC	2	1	–	1	1	1
BC	3	–	1	1	1	–
$A'B'C'$	4	0	0	0	–	1

This combination gives four distinct product terms: *AB*, *AC*, *BC*, and *A'B'C'*.

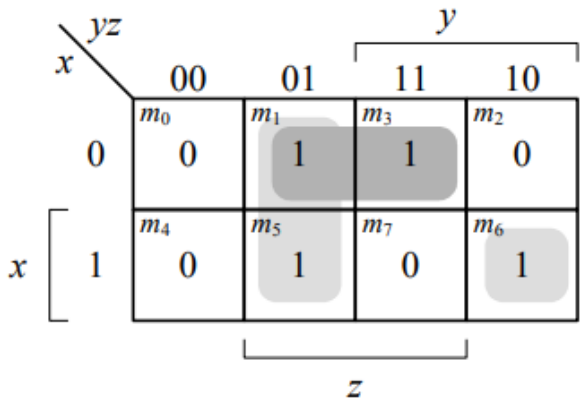
The PLA programming table for the combination is shown in the figure.

Problem – 7.19

Tabulate the PLA programming table for the four Boolean functions listed below. Minimize the numbers of product terms.

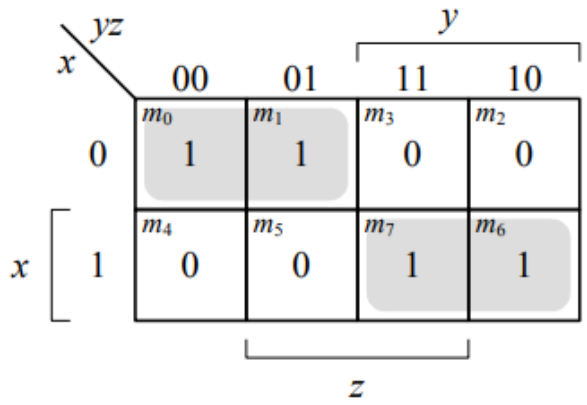
$$A(x, y, z) = (1, 3, 5, 6)$$
$$B(x, y, z) = (0, 1, 6, 7)$$
$$C(x, y, z) = (3, 5)$$
$$D(x, y, z) = (1, 2, 4, 5, 7)$$

Solution – 7.19



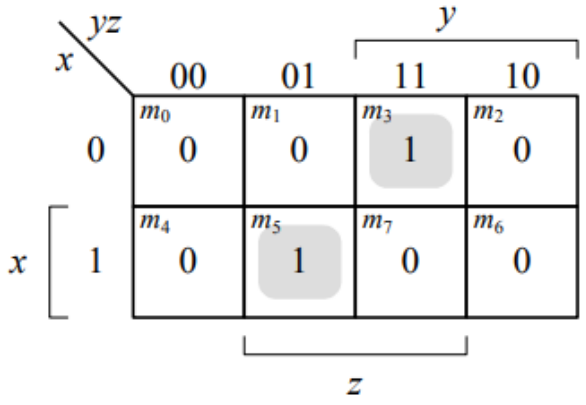
$A = y'z + x'z + xyz'$

$A' = y'z' + x'z' + xyz$



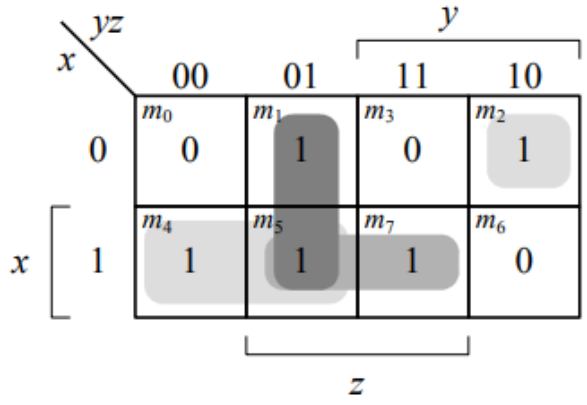
$B = xy + x'y'$

$B' = xy' + x'y$



$C = x'yz + xy'z$

$C' = z' + x'y' + xy$



$D = xy' + xz + y'z + x'yz'$

$D' = x'y'z' + x'yz + xyz'$

Solution – 7.19 – PLA Programming Table

	Product term	Inputs x y z	(T) A	(T) B	(T) C	(T) D
$y'z$	1	- 0 1	1	-	-	1
$x'z$	2	0 - 1	1	-	-	-
xyz'	3	1 1 0	1	-	-	-
xy	4	1 1 -	-	1	-	-
$x'y'$	5	0 0 -	-	1	-	-
$x'yz$	6	0 1 1	-	-	1	-
$xy'z$	7	1 0 1	-	-	1	-
xy'	8	1 0 -	-	-	-	1
Xz	9	1 – 1	-	-	-	1
$y'z$	10	- 0 1	-	-	-	1
$x'yz'$	11	0 1 0	-	-	-	1