7.8 PLA (PROGRAMMING LOGIC ARRAY)

It is a programmable device used to implement combinational logic circuits. PLA is most flexible device in the family of PLDs. The internal organization of a PLA is different form that of the ROM. The decoder of the ROM is replaced with an AND array that perform product terms of the input variables. The AND array is followed by an OR array which OR's together the product terms needed to form the output functions. Both the AND and OR arrays are programmable giving a lot of flexibility for implementing logic design.

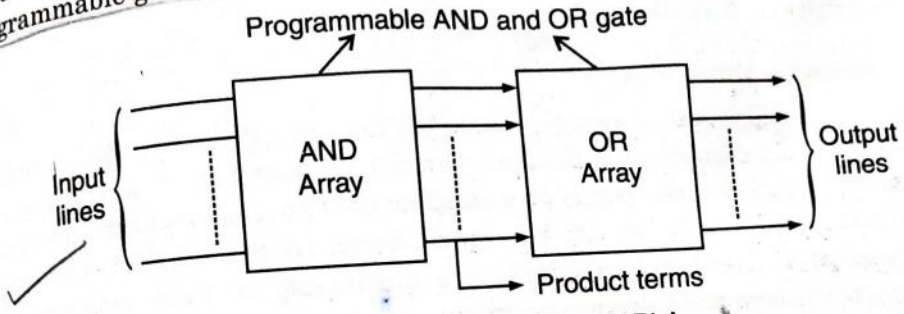


Fig. 7.19. Logic diagram of PLA.

The size of a PLA is specified by the number of inputs, the number of product terms and the number of outputs. The number of sum terms is equal to the number of outputs. It is denoted as $n \times p \times m$ PLA where

n denotes number of inputs
p denotes number of product terms
m denotes the number of output terms.

A comparison between ROM and PLA can be made to show how reduction in number of gates is possible in PLA. Consider a typical example of implementation of a combinational circuit of 16 inputs, 8 outputs and no more than 48 product terms. A $16 \times 48 \times 8$ PLA can serve the purpose which consists of 48 product terms. To implement the same combinational circuit, a $2^{16} \times 8$ ROM is needed which consists of $2^{16} = 65536$ minterms or product terms. So there is a drastic reduction in number of AND gates with in the chip, thus reducing the fabrication time and cost. In PLA both complemented and uncomplemented inputs, *i.e.*, 2^n number of inputs appear at each AND gates providing maximum flexibility in product term generation.

Advantage of PLA

Following are the advantages of PLA:

(1) As both AND and OR array are programmable. It gives as lot of flexibility for implementing logic design.
(2) Efficient in terms of area needed for implementation on a IC chip ($P < 2^N$).
(3) It is included as a part of larger chips such as microprocessors.
(4) Power requirement is less than ROM (as less product terms $p < 2^N$).
(5) Fabrication time decreases as number of AND gates drastically reduces (as compared to ROM).
(6) Cost is also less (as hardware reduces as compare to ROM)

Disadvantage of PLA

In ROM based design, since all minterms are generated. Hence realization of a set of boolean function is based on minterms. It is not necessary to minimize expressions prior to realization with ROM.

Yet in **PLA based design** product terms generated are not necessarily all minterms. There is always a specific limit on use of AND gates and it is necessary to obtain a set of reduced expression in a way that number of product terms doesnot exceed the number of AND gates in PLA. Hence, **simplification of boolean function is needed.**

7.8.1. Designing Using PLA

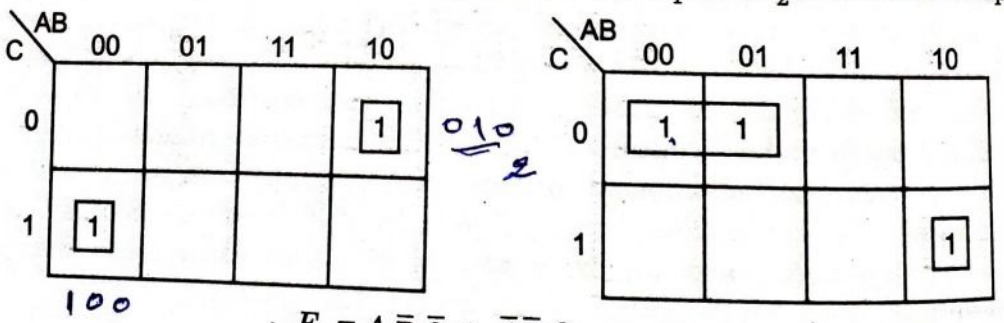
In case of ROM based design, since all the minterms are generated in a ROM, the realization of a set of Boolean functions is based on minterms canonical expressions. It is never necessary to minimize the expressions prior to obtaining the realization with a ROM. On the other hand, in case of PLA, the product terms generated are not necessarily the minterms, as these product terms depend upon the how the fuses are programmed. As a consequence the realization using PLA is based on the SOP expressions. It is necessary to obtain a set of expressions in such a way that number of product terms does not exceed the number of AND gates in the PLA. Therefore, simplification of Boolean functions is needed.

EXAMPLE 7.5. Design the following function using PLA.

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

$$F_2 (A, B, C) = \Sigma m\ (0, 2, 5)$$

Solution: In order to realize the output functions F_1 and F_2 solve the K-map.



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

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

Therefore the functions solved by K-map shows that it requires 4 AND gates and 2 OR gate. The implementation using PLA is given as follows :

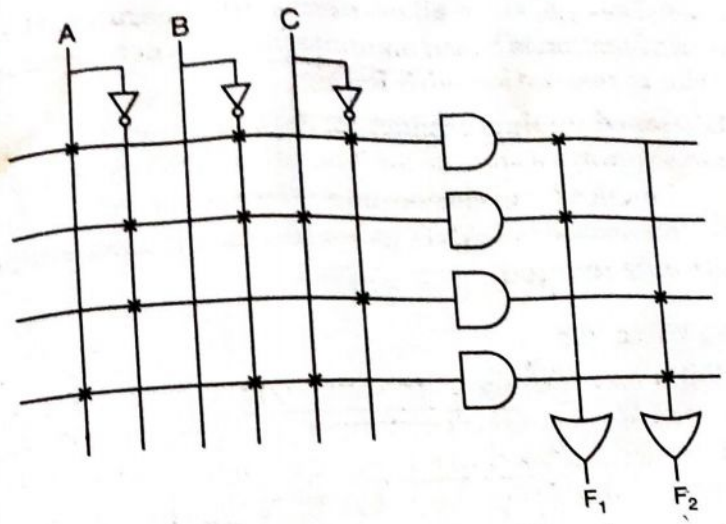


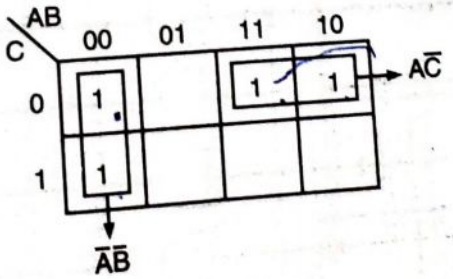
Fig. 7.20.

EXAMPLE 7.6. Solve the given functions using PLA

$$F_0 = \Sigma m\ (0, 1, 4, 6)$$
$$F_1 = \Sigma m\ (2, 3, 4, 6, 7)$$
$$F_2 = \Sigma m\ (0, 1, 2, 6)$$
$$F_3 = \Sigma m\ (2, 3, 5, 6, 7)$$

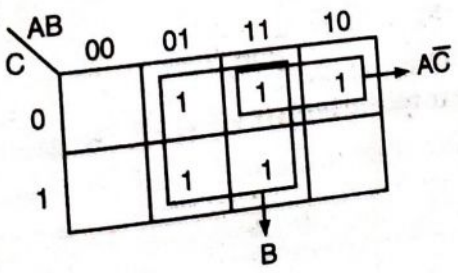
Solution: Solve the K-map for the F_0, F_1, F_2, F_3

For F_0 :



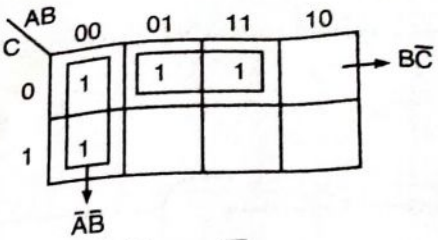
$$F_0 = \overline{A}\,\overline{B} + A\overline{C}$$

For F_1 :



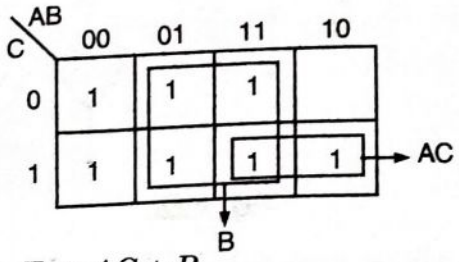
$$F_1 = B + A\overline{C}$$

For F_2 :



$$F_2 = \overline{A}\,\overline{B} + B\overline{C}$$

For F_3 :



$$F_3 = AC + B$$

In order to design these function 5 AND and 4 OR gates are required.

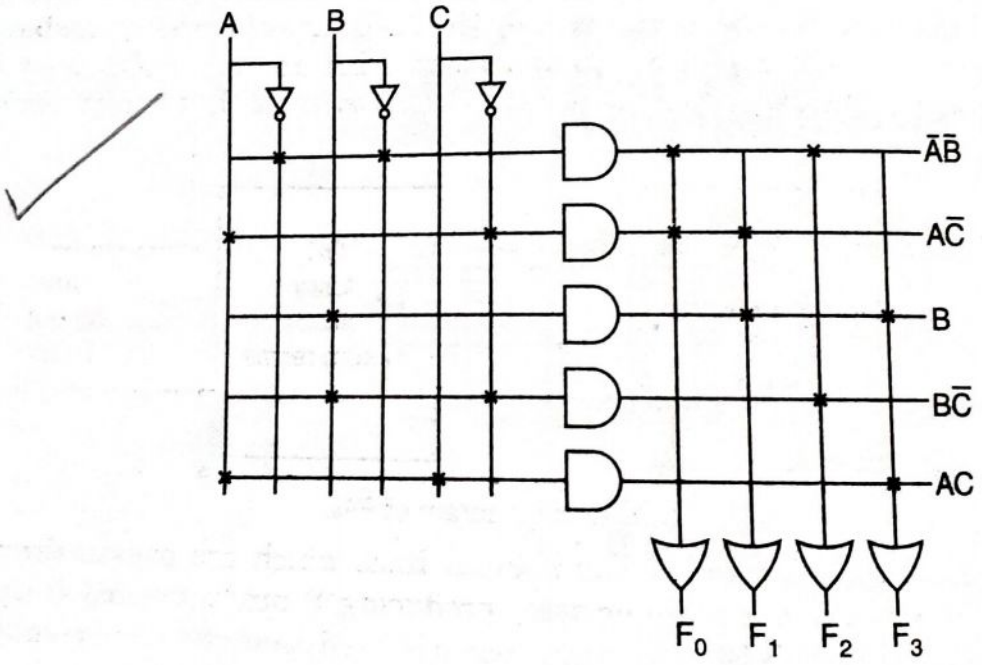


Fig. 7.21.

EXAMPLE 7.7. Design the circuit of Half Subtractor using PLA.

Solution: The truth table for half subtractor is given as :

A	*B*	*D (Difference)*	B_n *(Borrow)*
0	0	0	0
0	1	1	1
1	0	1	0
1	1	0	0

$$D = \overline{A}B + A\overline{B}$$

$$B_n = \bar{A}\,B$$

The given equations required two AND gate and 2 OR gates.

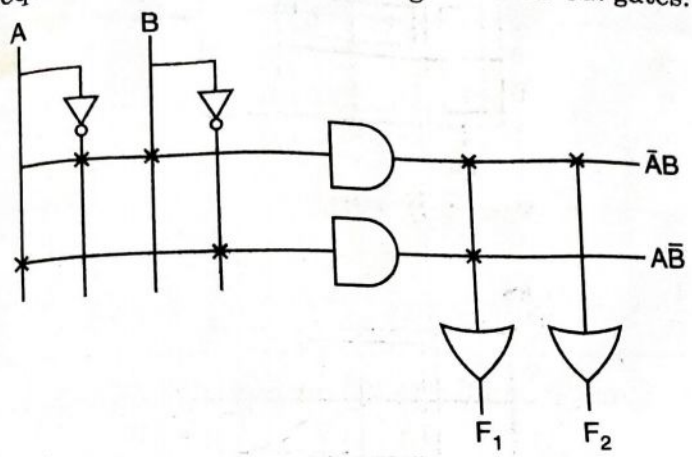


Fig. 7.22.

7.9 PAL (PROGRAMMABLE ARRAY LOGIC)

It is a special case of PLA which has a programmable AND array and a fixed OR array. It is cheap compared to PLA as only the AND array is programmable. It is also easy to program a PAL compared to PLA as only AND must be programmed. On the other hand, since the OR array is fixed, it is less flexible than PLA device.

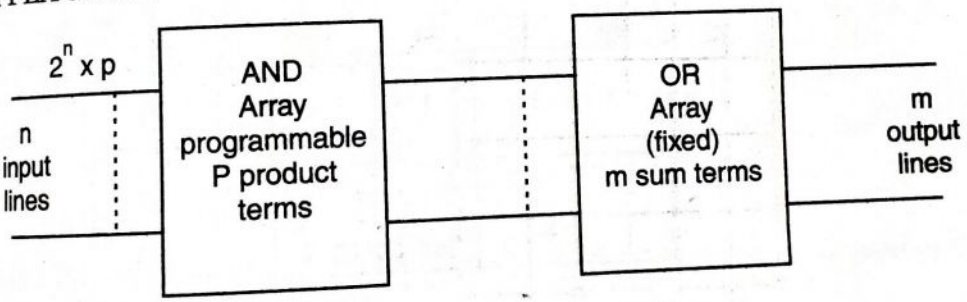


Fig. 7.23. Block diagram of PAL.

From the block diagram, it has n input lines which are passed through AND array which is programmable array producing 'P' product terms. Outputs of AND gates are then fed to the OR gate which is fixed connection and producing m output lines.

Advantages

(1) It is easier to program than PLA.
(2) It is less expensive than PLA.

Disadvantages

It is less flexible as compared to PLA because OR gates are fixed only AND gates are programmable.

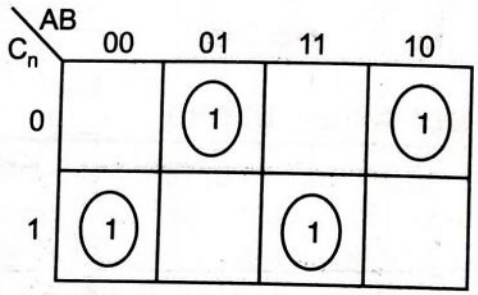
EXAMPLE 7.8. Implement the circuit of full Adder using PAL

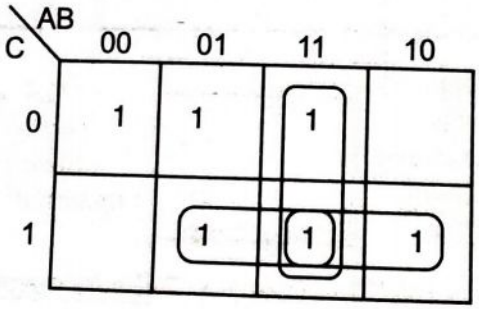
Solution: In order to realize the circuit of full adder, the truth table is designed and then K-map is solved.

The truth table of full Adder is given as follows :

A	B	Cn	S	C
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

The K-map for Sum (S) and Carry (C) output is given as :





$$\text{Sum } (S) = \bar{A}\,B\,\overline{C_n} + A\,\bar{B}\,\overline{C_n} + \bar{A}\,\bar{B}\,C_n + ABC$$

$$\text{Carry } (C) = AB + BC_n + C_nA$$

The given output equations are realized by using ($2^n = 2^3 = 8$) 8 AND gates and two OR gates. The designing is as follows using PAL :

The input which have dots are forwarded through OR gate.

Programmable Lo

EXAMPL
PAL and
Solution:

Decimal No.
0
1
2
3
4
5
6
7
8
9

Desi

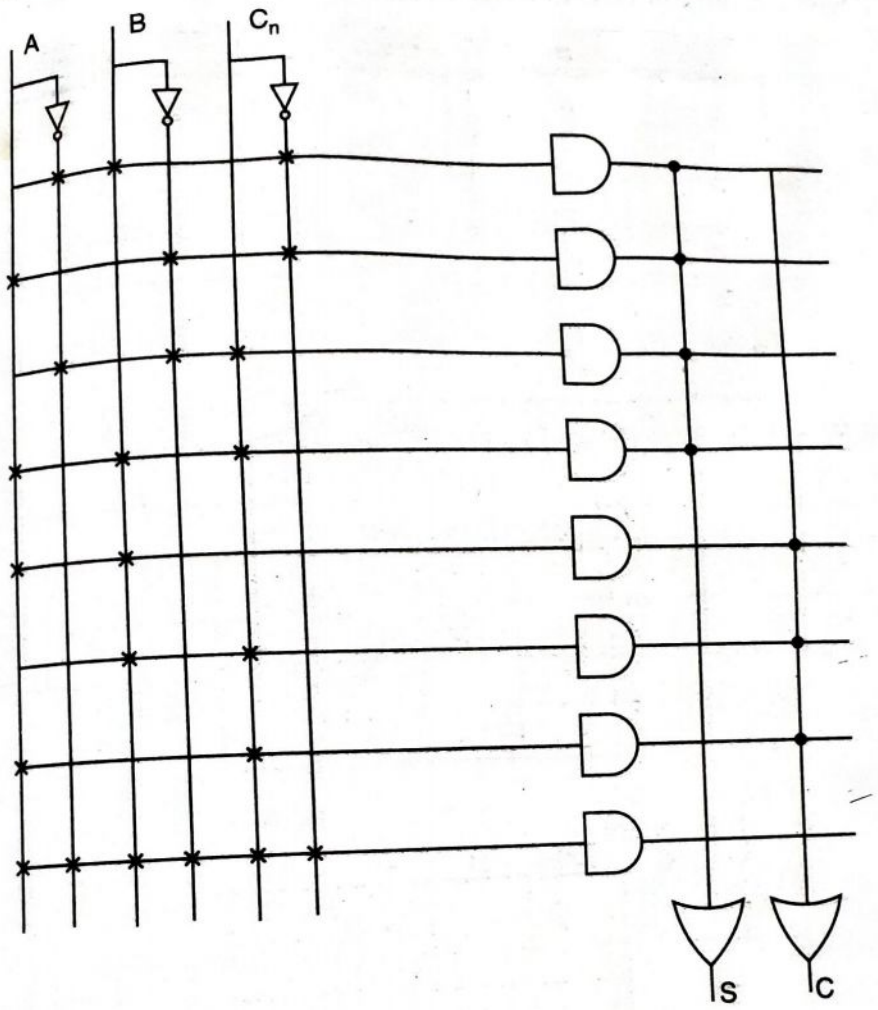


Fig. 7.24.

EXAMPLE 7.9. Design a BCD to Excess-3 Code converter Counter using PAL and PLA:

Solution: The truth table for BCD to Excess-3 Code is given as follows :

Decimal No.	*BCD Code*					*Excess-3 Code*			
	A	*B*	*C*	*D*		*W*	*X*	*Y*	*Z*
0	0	0	0	0	———	0	0	1	1
1	0	0	0	1	———	0	1	0	0
2	0	0	1	0	———	0	1	0	1
3	0	0	1	1	———	0	1	1	0
4	0	1	0	0	———	0	1	1	1
5	0	1	0	1	———	1	0	0	0
6	0	1	1	0	———	1	0	0	1
7	0	1	1	1	———	1	0	1	0
8	1	0	0	0	———	1	0	1	1
9	1	0	0	1	———	1	1	0	0

Designing of the output W, X, Y, Z by using K-map is given as follows :

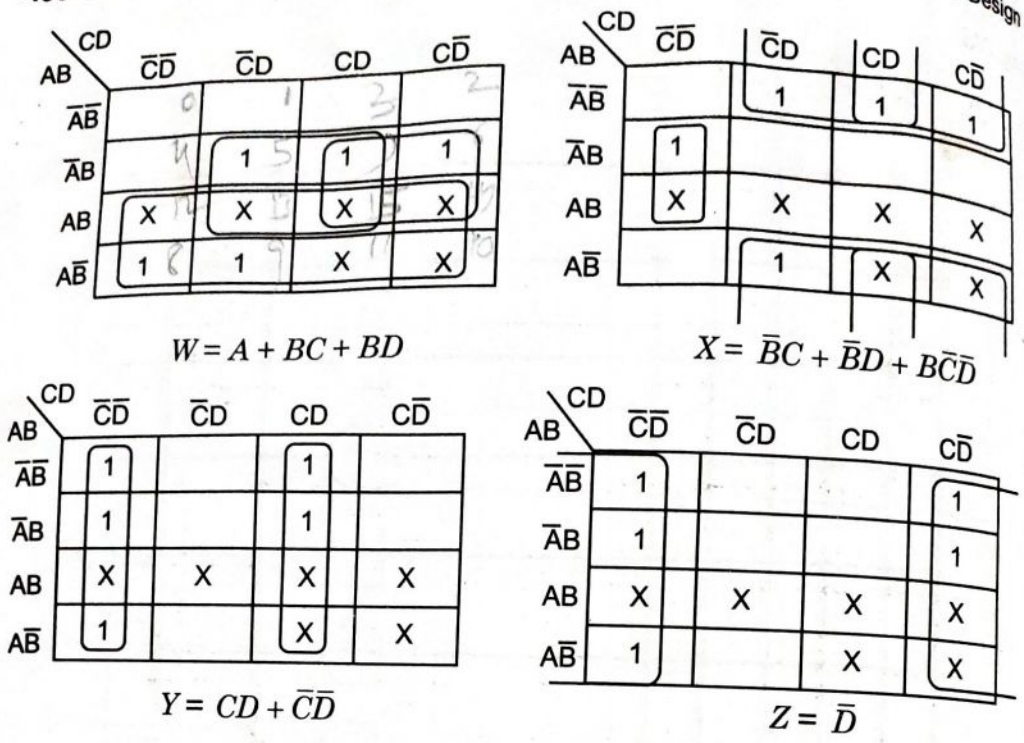


Fig. 7.25.

Using PLA

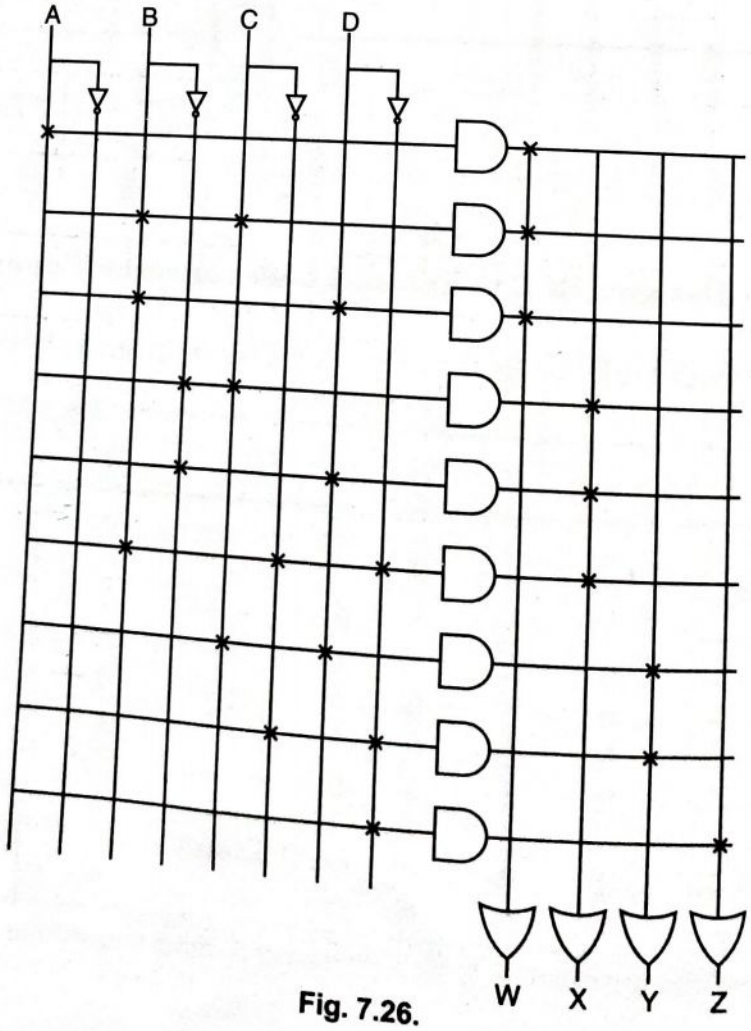


Fig. 7.26.

Programmable L

Using PAL

EXAMPL
and PAL
Solution

Using PAL

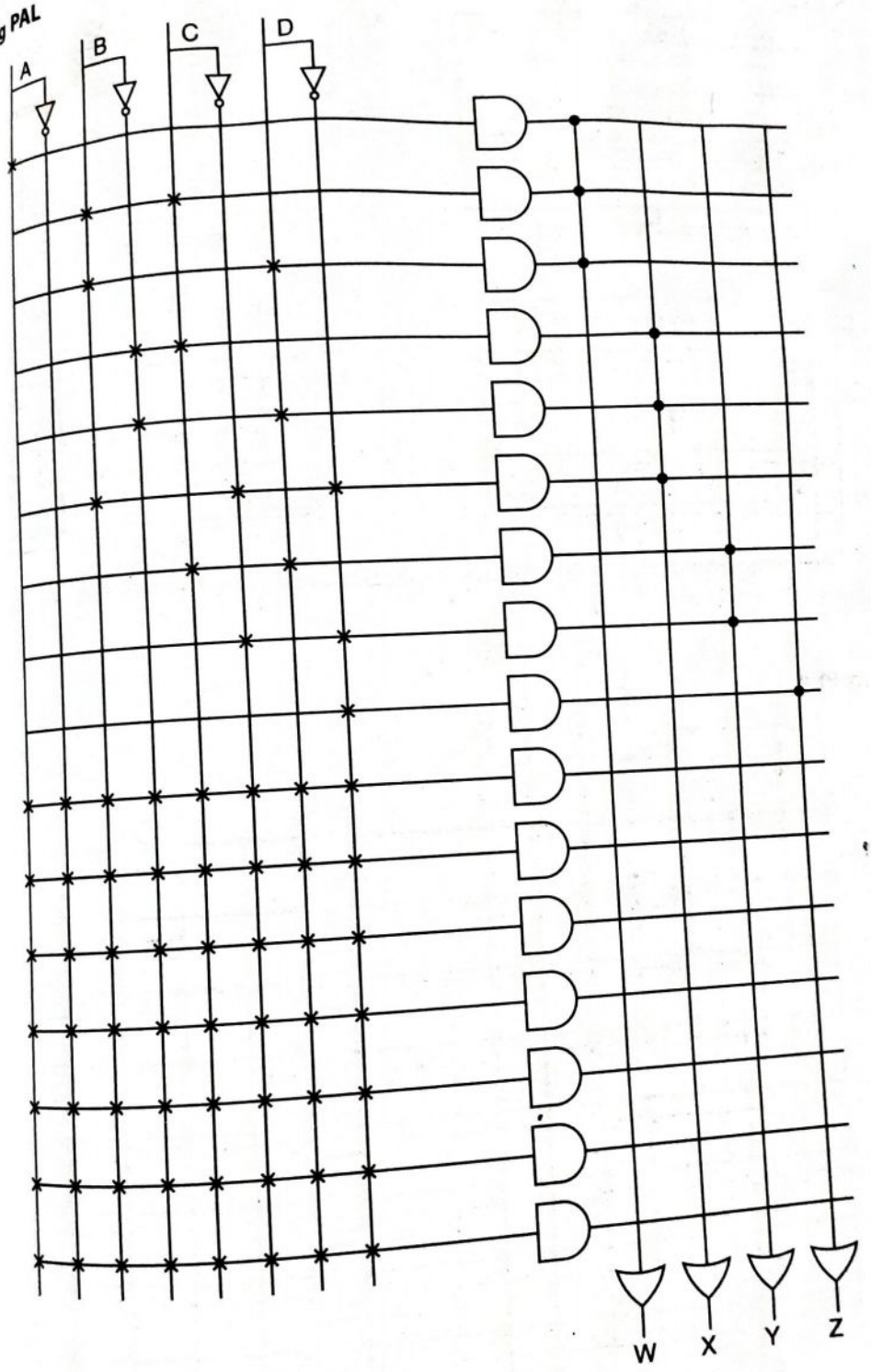
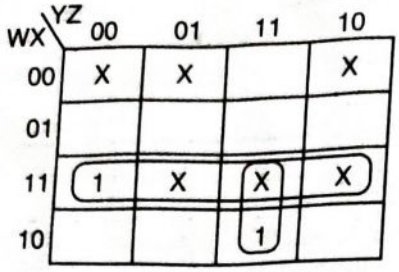


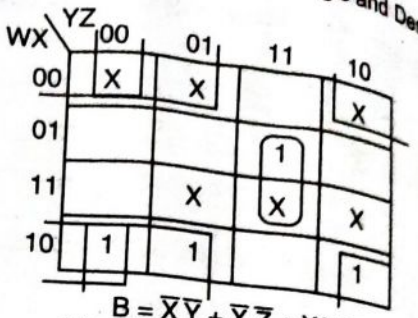
Fig. 7.27.

EXAMPLE 7.10. Design Excess-3 to BCD Code. Converter using PLA and PAL.

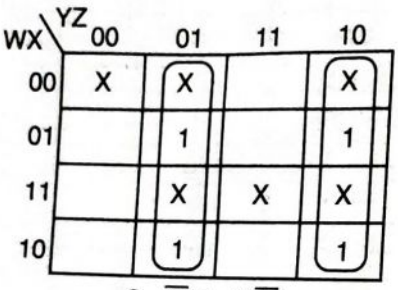
Solution: K-map for Excess-3 to BCD Code is given as :



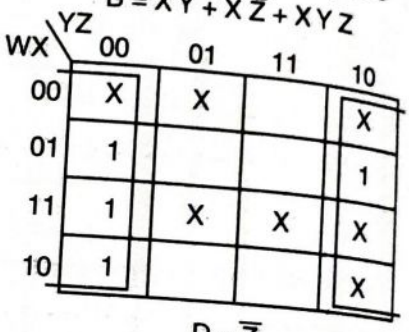
$A = WX + WYZ$



$B = \overline{X}\,\overline{Y} + \overline{X}\,\overline{Z} + XYZ$



$C = \overline{Y}Z + Y\overline{Z}$



$D = \overline{Z}$

The designing of the given equations A, B, C and D by using **PLA** is given as follows :

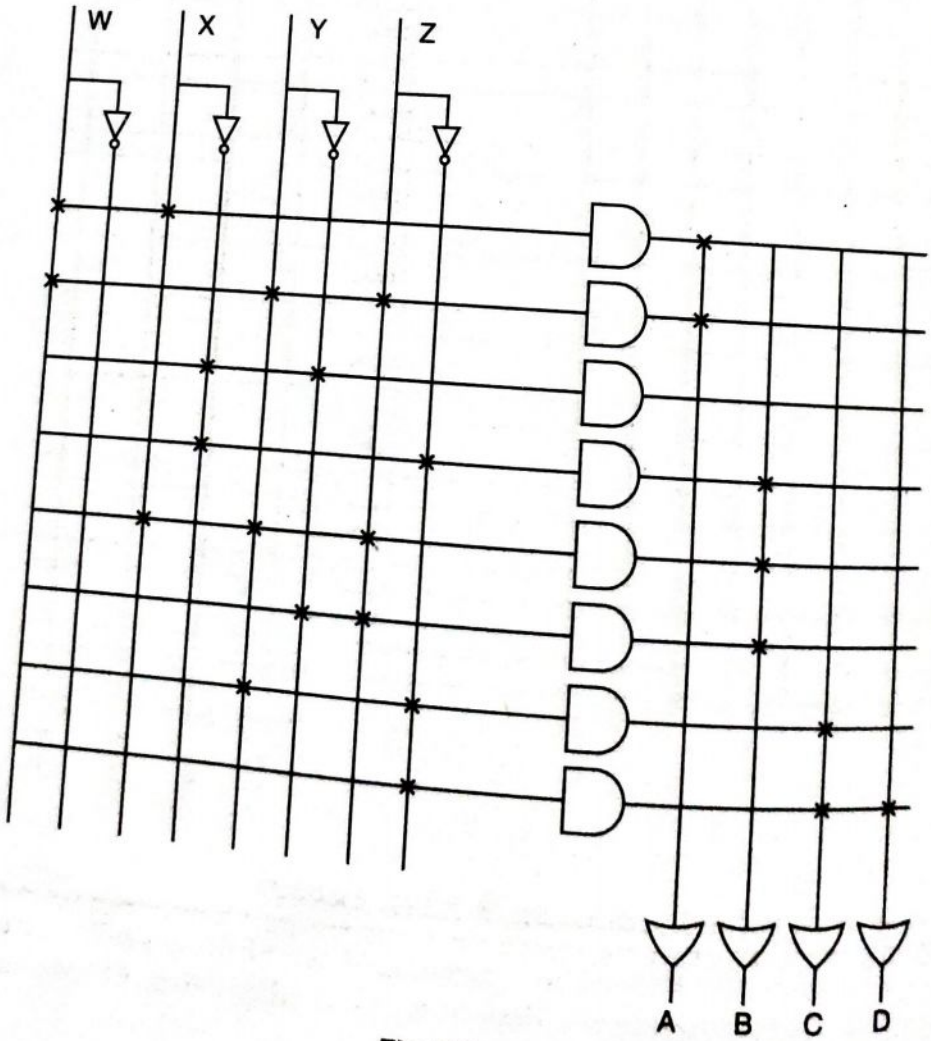


Fig. 7.28.