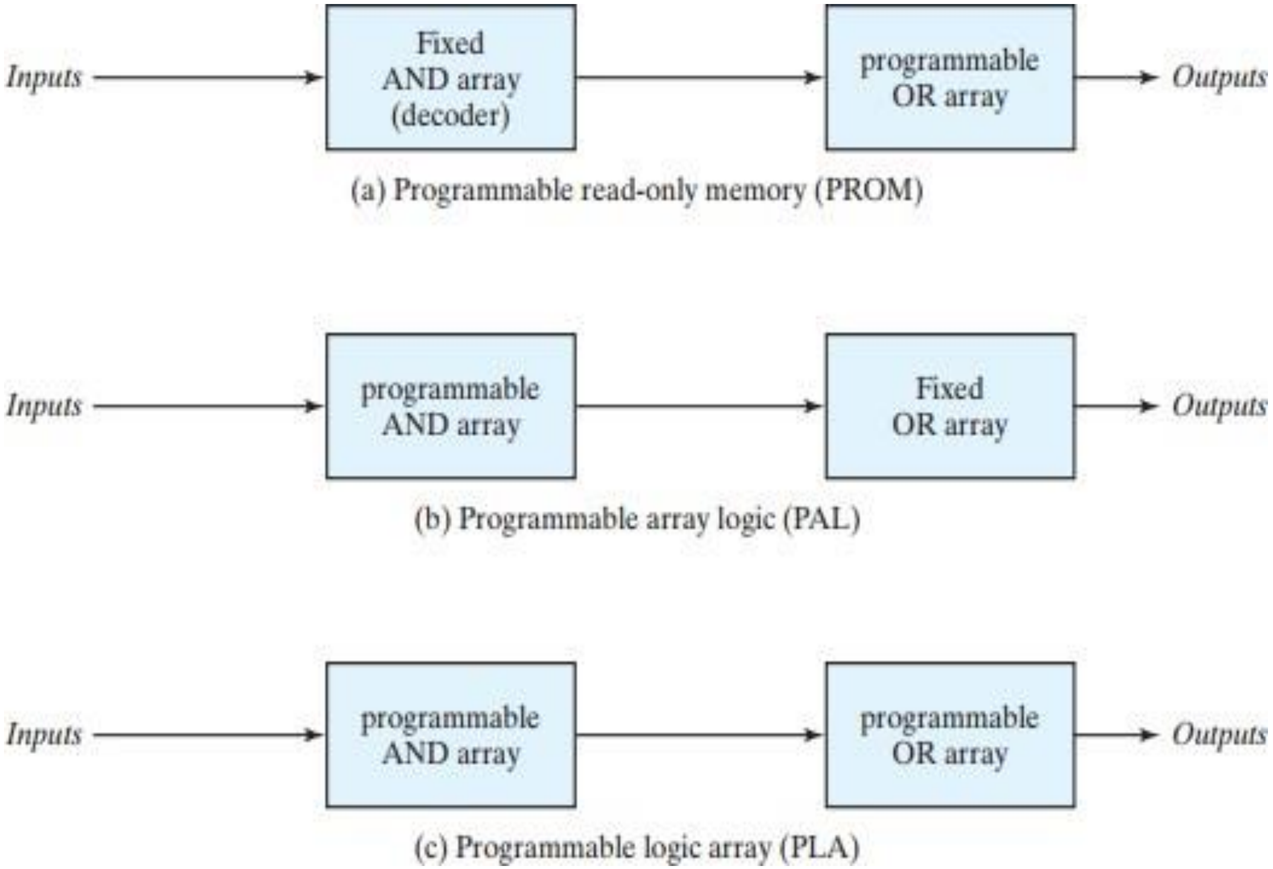
UNIT VI: ECE 213
PLA and PAL

Programmable Logic Array(PLA)

- It has programmable AND array and programmable OR array.
- Because both arrays are programmable, it is flexible.
- 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 .

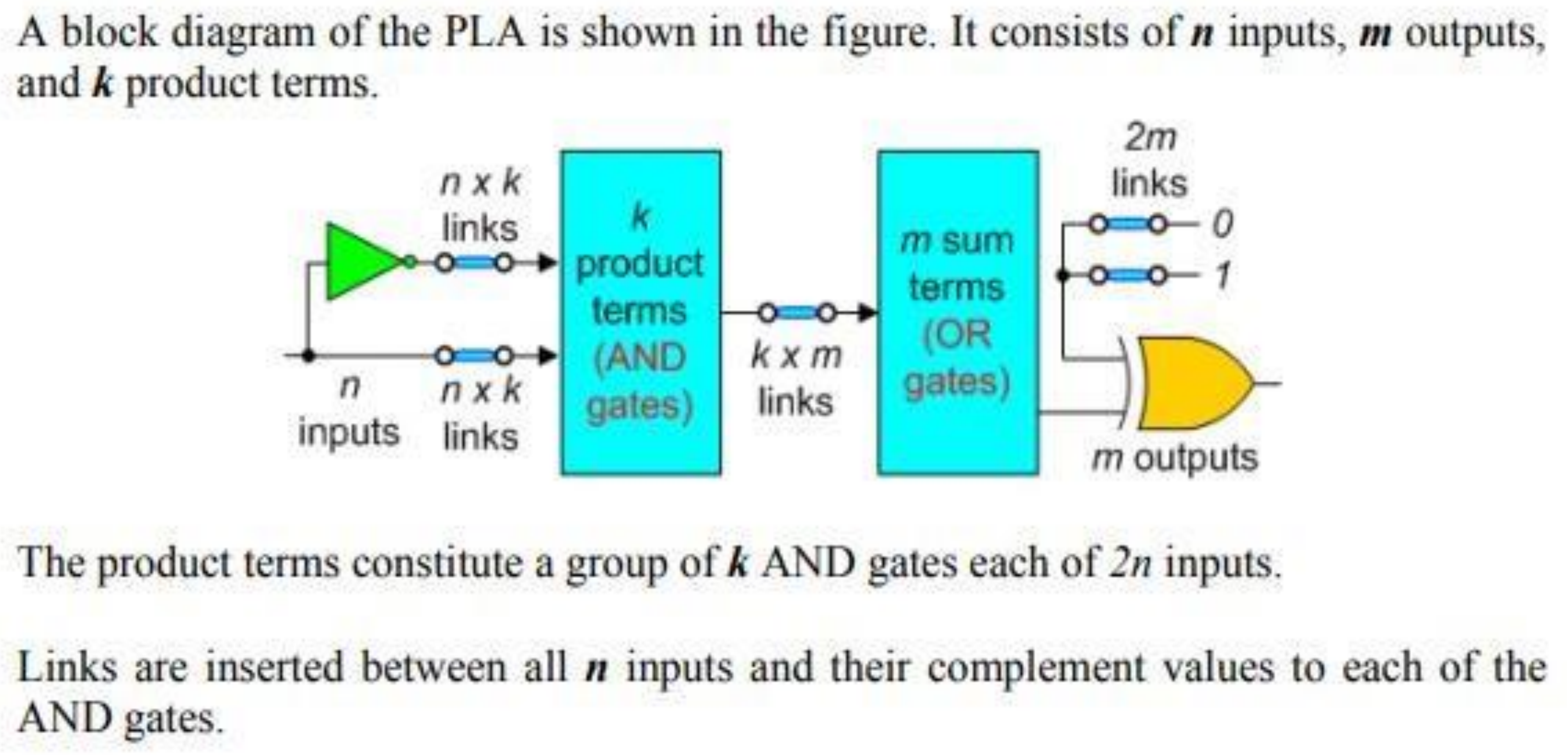


(a) Programmable read-only memory (PROM)

(b) Programmable array logic (PAL)

(c) Programmable logic array (PLA)

- In PLAs, instead of using a decoder as in PROMs, a number (k) of AND gates is used where k < 2n , (n is the number of inputs).



A block diagram of the PLA is shown in the figure. It consists of ***n*** inputs, ***m*** outputs, and ***k*** product terms.

The product terms constitute a group of ***k*** AND gates each of 2*n* inputs.

Links are inserted between all ***n*** inputs and their complement values to each of the AND gates.

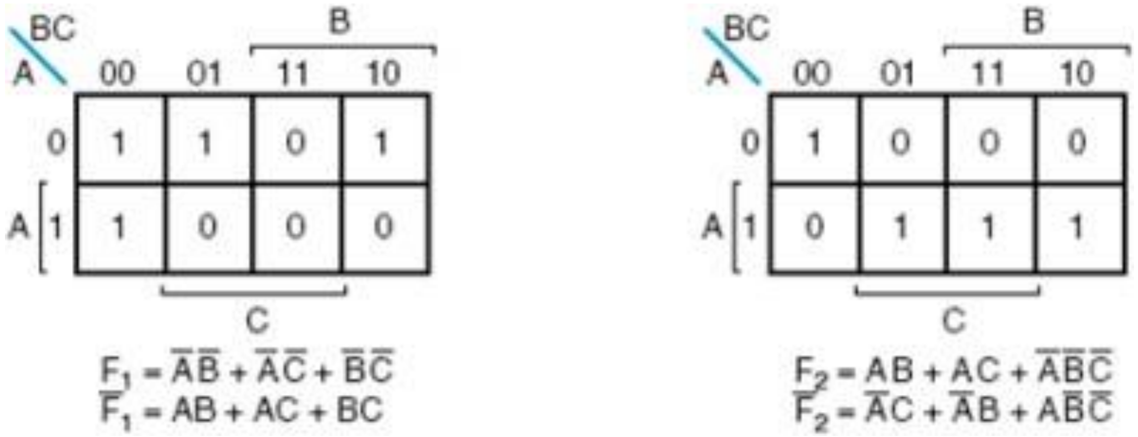
Source: Digital Fundamentals, Floyd

- The output of the OR gate goes to an XOR gate, where the other input can be programmed to receive a signal equal to either logic 1 or logic 0.
- The output is inverted when the XOR input is connected to 1 (since $x\ XOR\ 1 = \bar{\ }$).
- The output does not change when the XOR input is connected to 0 (since $x\ XOR\ 0 = x$).

Example 1: Implement the following table using PLA.

A	B	C	F1	f2
0	0	0	1	1
0	0	1	1	0
0	1	0	1	0
0	1	1	0	0
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	0	1

Step 1: K-maps for simplification:



A \ BC	00	01	11	10
0	1	1	0	1
1	1	0	0	0

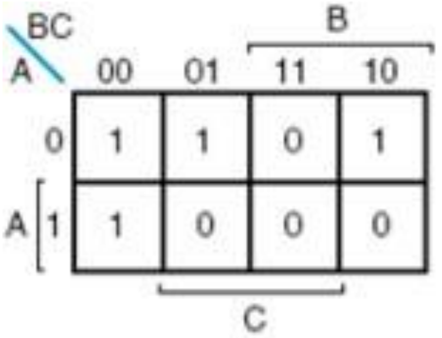
$F_1 = \overline{A}\,\overline{B} + \overline{A}\,\overline{C} + \overline{B}\,\overline{C}$
$\overline{F_1} = AB + AC + BC$

A \ BC	00	01	11	10
0	1	0	0	0
1	0	1	1	1

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

- Designing using a PLA, a careful investigation must be taken in order to reduce the distinct product terms.
- Both the true and complement forms 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.

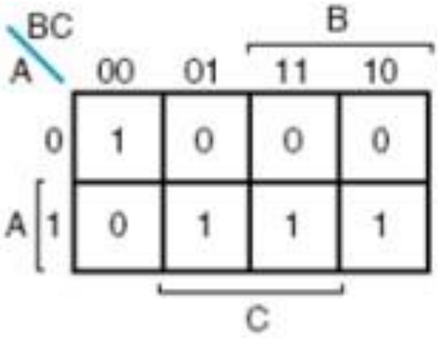
- K-maps for simplification:



A \ BC	00	01	11	10
0	1	1	0	1
1	1	0	0	0

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

$\overline{F_1} = AB + AC + BC$



A \ BC	00	01	11	10
0	1	0	0	0
1	0	1	1	1

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

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

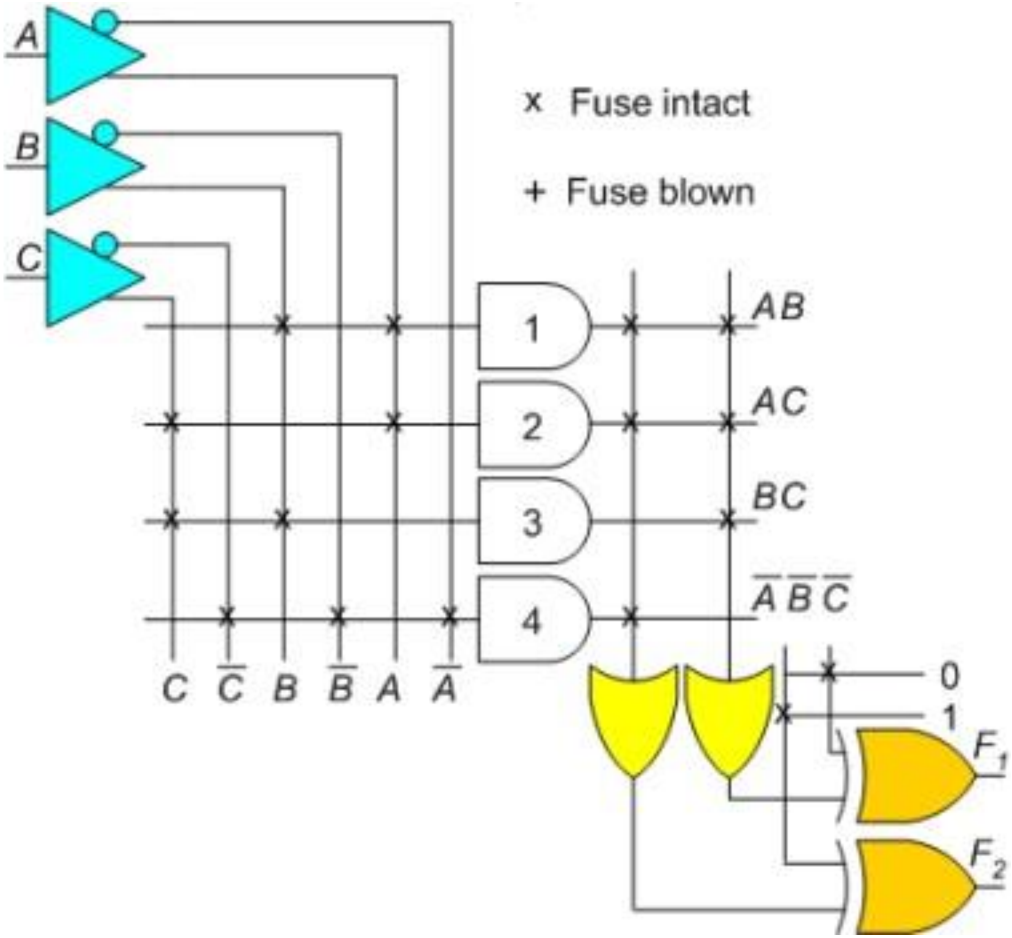
The combination that gives a minimum number of product terms is:

$F_1{}' = AB + AC + BC$ or $F_1 = (AB + AC + BC)'$

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

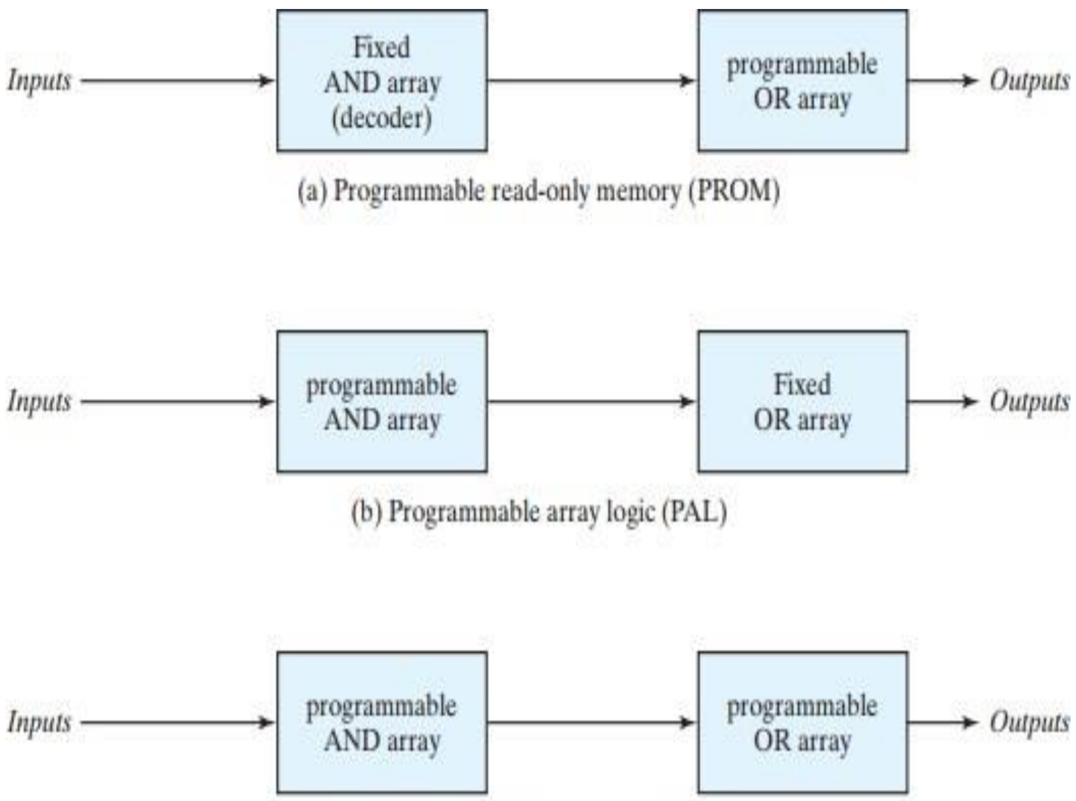
This gives only 4 distinct product terms: *AB*, *AC*, *BC*, and *A'B'C'*.

Step 2: Logic Design


A
B
C
x Fuse intact
+ Fuse blown
1
2
3
4
AB
AC
BC
$\overline{A}\,\overline{B}\,\overline{C}$
C $\overline{C}$ B $\overline{B}$ A $\overline{A}$
0
1
F_1
F_2

Programmable Array Logic (PAL)

- It has programmable AND array and fixed OR array.
- Because only the AND array is programmable, it is easier to use
- However, it is not flexible as compared to Programmable Logic Array (PLA).

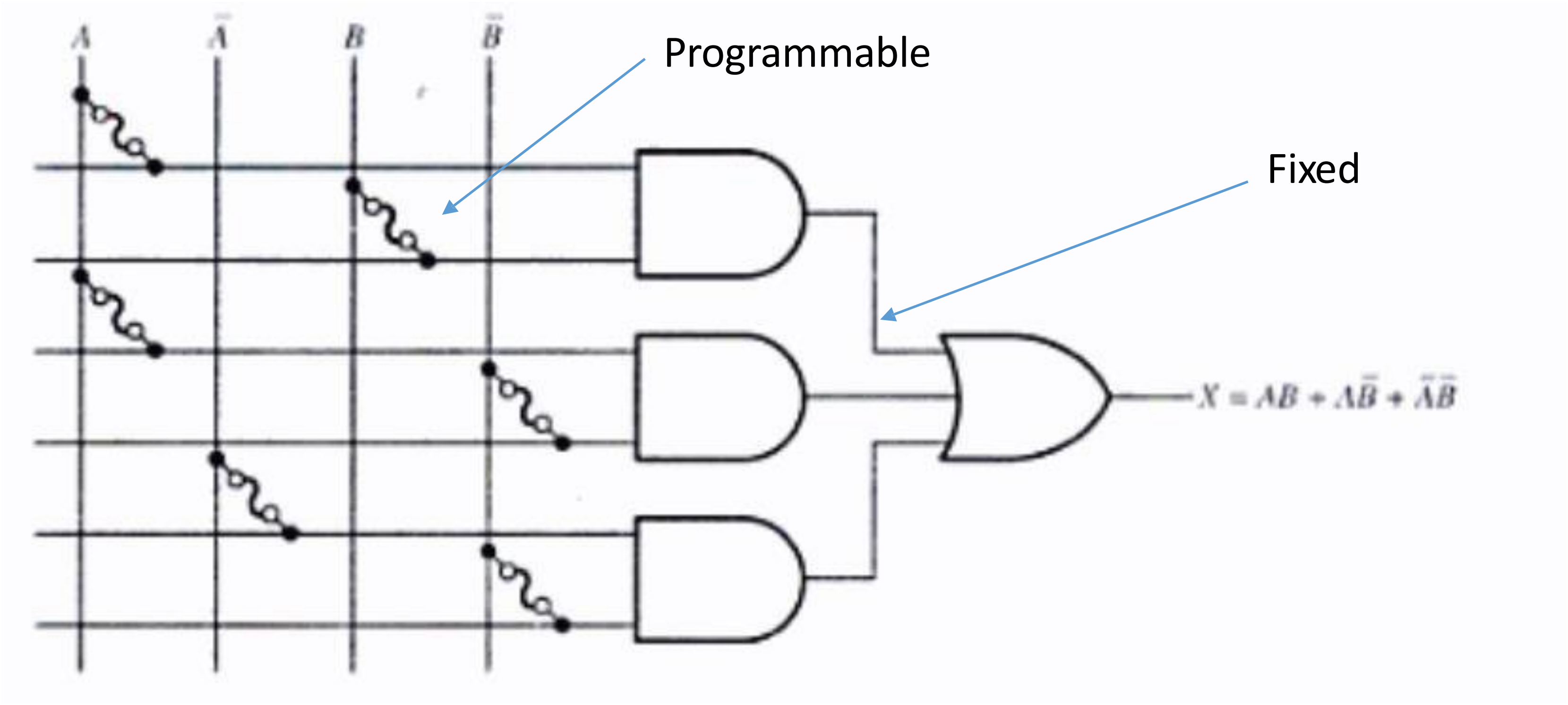


(a) Programmable read-only memory (PROM)

(b) Programmable array logic (PAL)

(c) Programmable logic array (PLA)

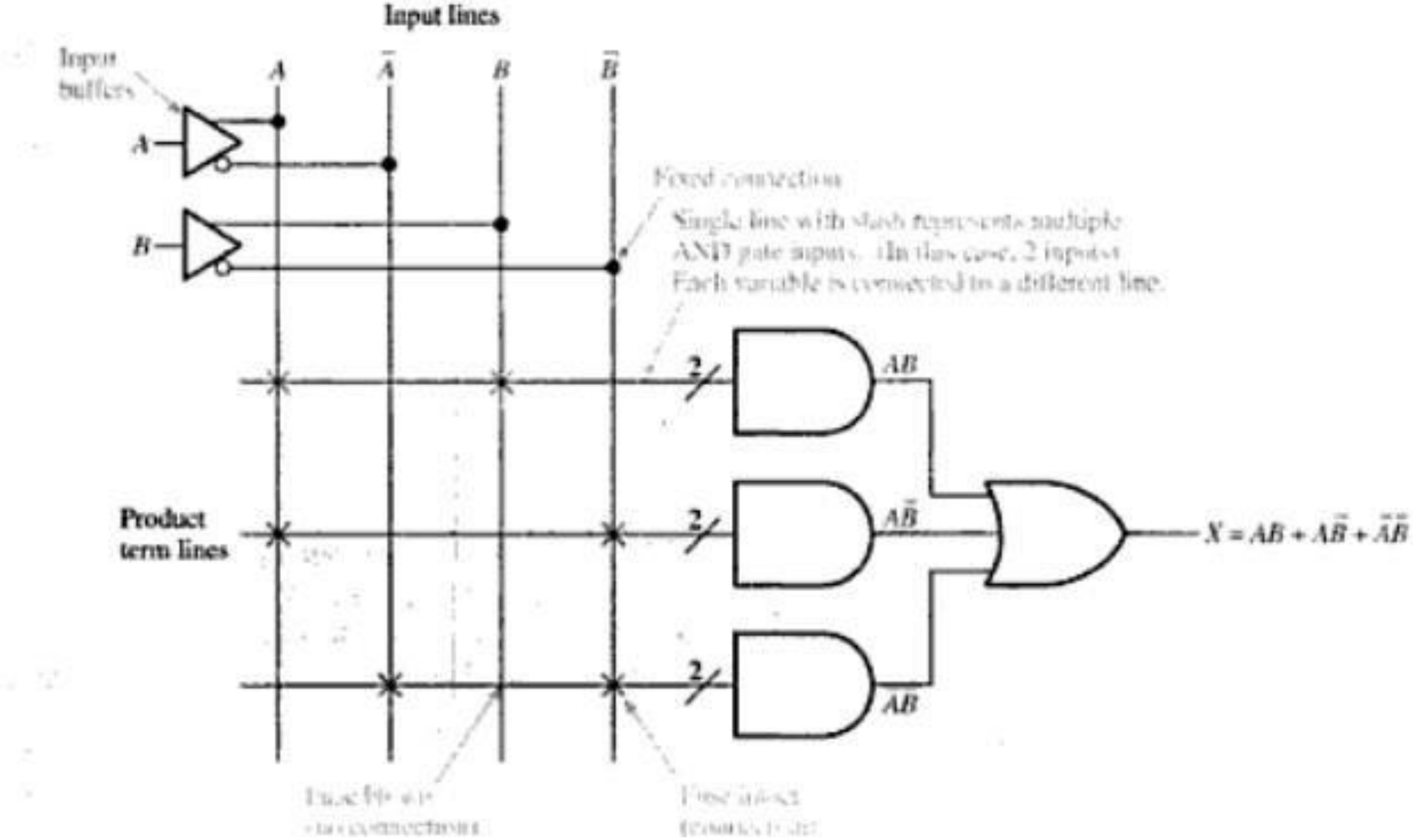
Example 1: Implement $\boldsymbol{AB} + \boldsymbol{A\bar{B}} + \boldsymbol{\bar{A}\bar{B}}$ using PAL?



Programmable

Fixed

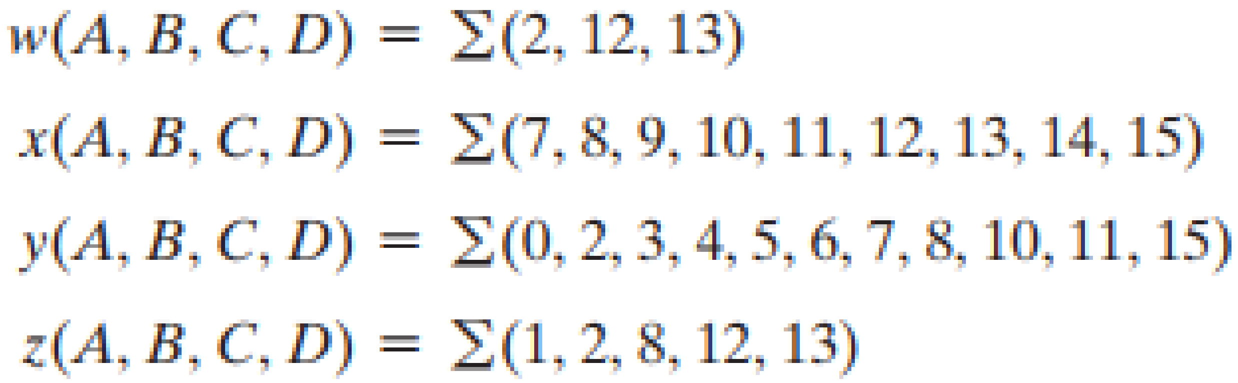
Source: Digital Fundamentals, Floyd


Input lines
Input buffers
A
$\bar{A}$
B
$\bar{B}$
A
B
Fixed connection
Single line with slash represents multiple AND gate inputs. (In this case, 2 inputs) Each variable is connected to a different line.
Product term lines
2
2
2
AB
$A\bar{B}$
AB
$X = AB + A\bar{B} + \bar{A}\bar{B}$
Fuse blown (no connection)
Fuse intact (connected)

Simplified Diagram

Source: Digital Fundamentals, Floyd

Example 2:
Implement following function using PAL?



$$w(A, B, C, D) = \sum(2, 12, 13)$$

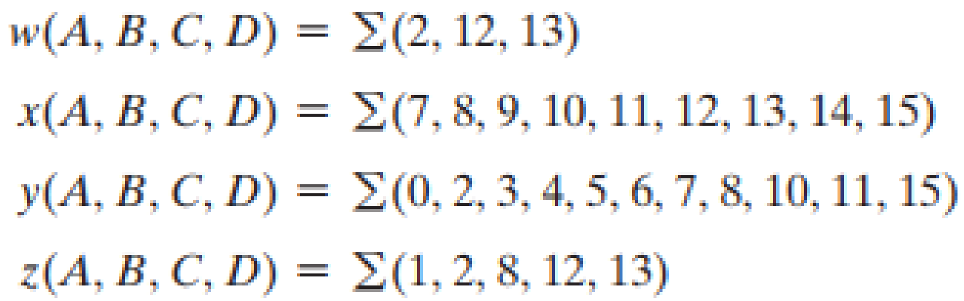
$$x(A, B, C, D) = \sum(7, 8, 9, 10, 11, 12, 13, 14, 15)$$

$$y(A, B, C, D) = \sum(0, 2, 3, 4, 5, 6, 7, 8, 10, 11, 15)$$

$$z(A, B, C, D) = \sum(1, 2, 8, 12, 13)$$

Sol:

Use K-map for Simplification

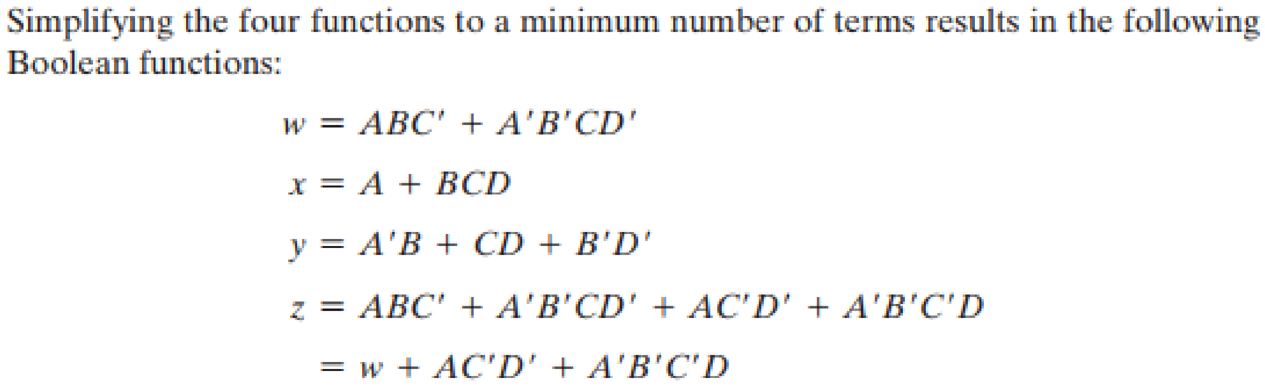


$$w(A, B, C, D) = \Sigma(2, 12, 13)$$

$$x(A, B, C, D) = \Sigma(7, 8, 9, 10, 11, 12, 13, 14, 15)$$

$$y(A, B, C, D) = \Sigma(0, 2, 3, 4, 5, 6, 7, 8, 10, 11, 15)$$

$$z(A, B, C, D) = \Sigma(1, 2, 8, 12, 13)$$



Simplifying the four functions to a minimum number of terms results in the following Boolean functions:

$$w = ABC' + A'B'CD'$$

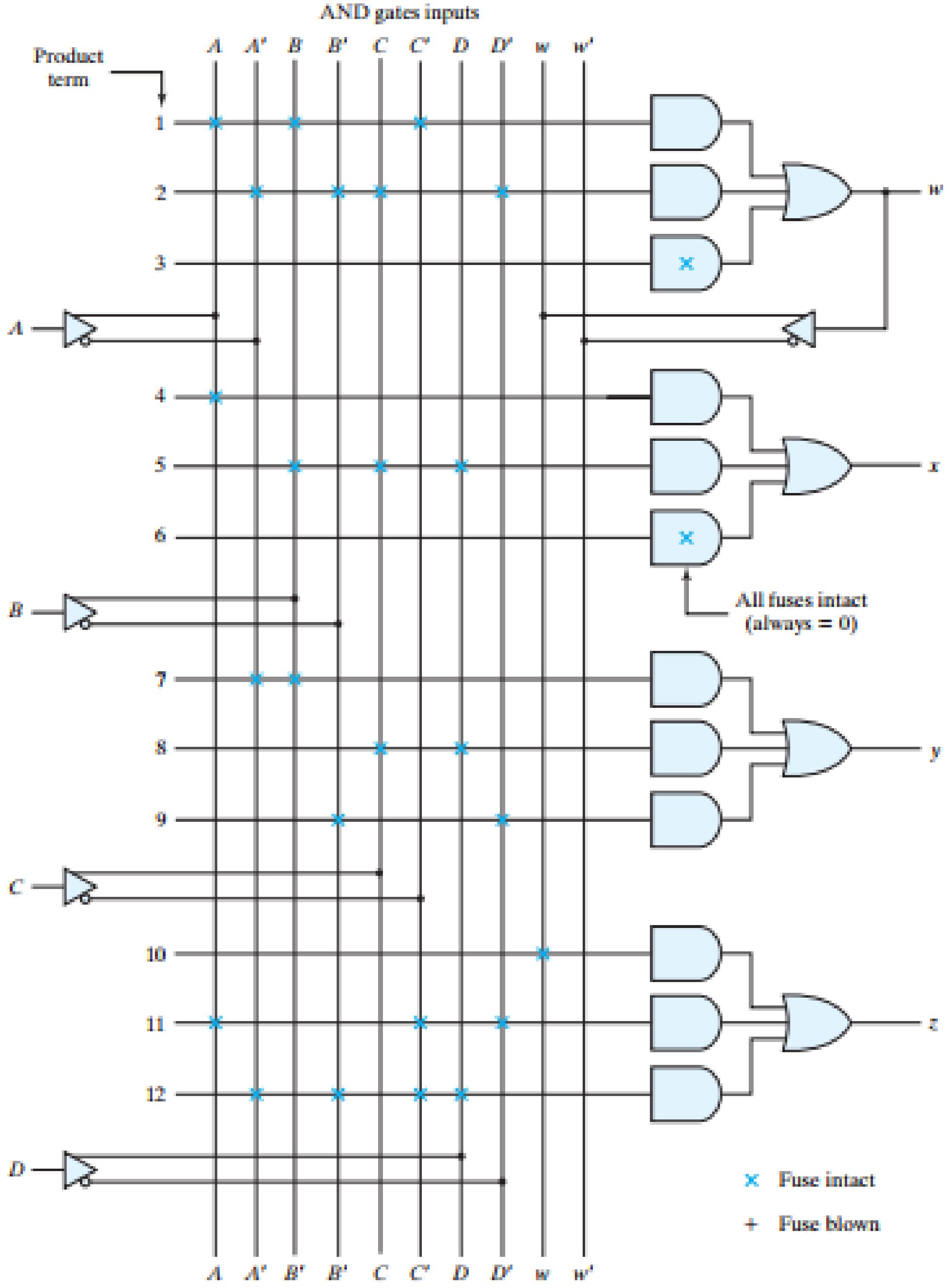
$$x = A + BCD$$

$$y = A'B + CD + B'D'$$

$$z = ABC' + A'B'CD' + AC'D' + A'B'C'D$$

$$= w + AC'D' + A'B'C'D$$

LOGICAL DIAGRAM for PAL:



MCQ

PLAs, CPLDs, and FPGAs are all which type of device?

A. SLD
B. PLD
C. EPROM
D. SRAM

MCQ

PAL refers to

a) Programmable Array Loaded

b) Programmable Logic Array

c) Programmable Array Logic

d) None of the Mentioned

MCQ

PLD contains a large number of

a) Flip-flops

b) Gates

c) Registers

d) All of the Mentioned