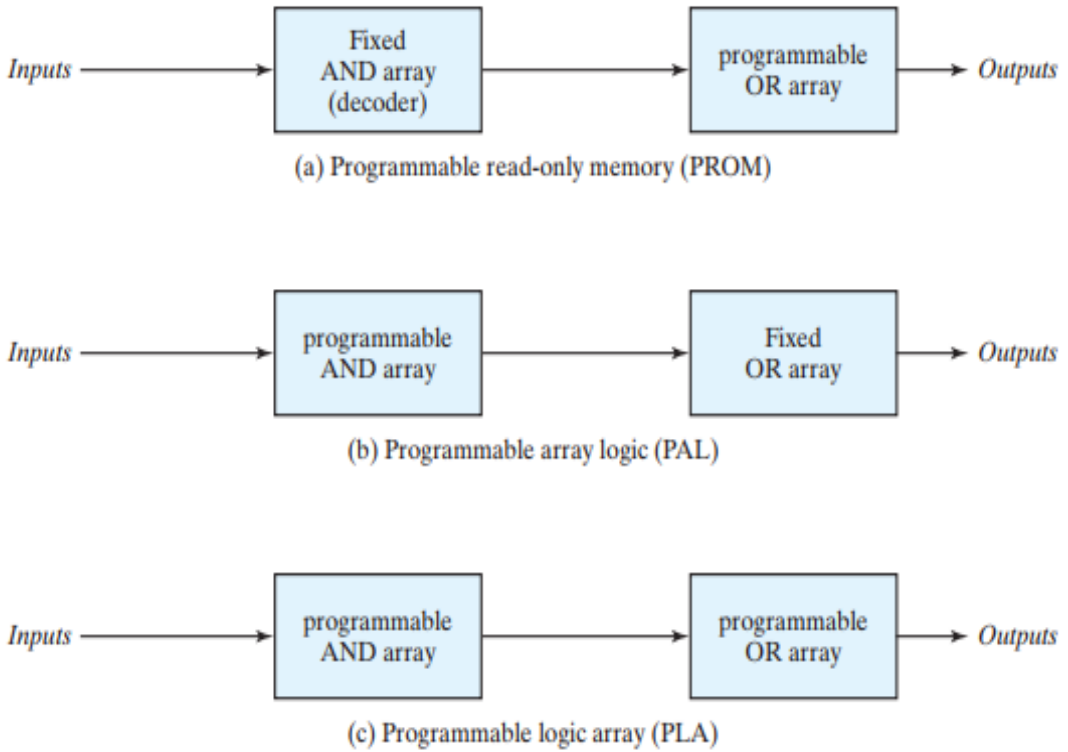
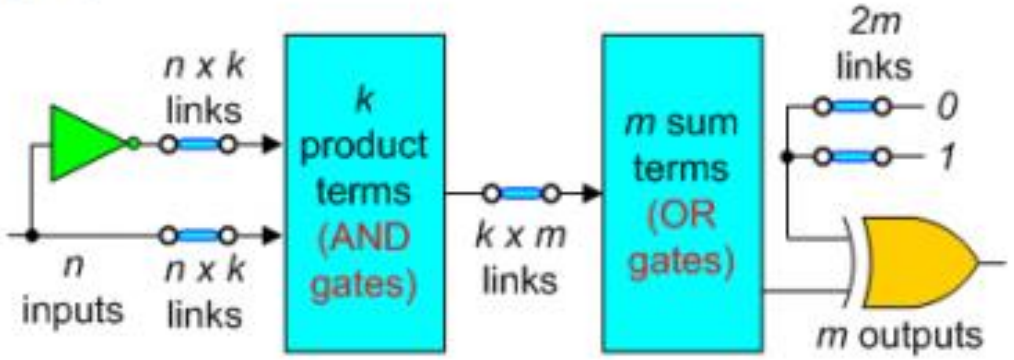
Programmable Logic Array(PLA)

- 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 .



- 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 *2n* inputs.

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

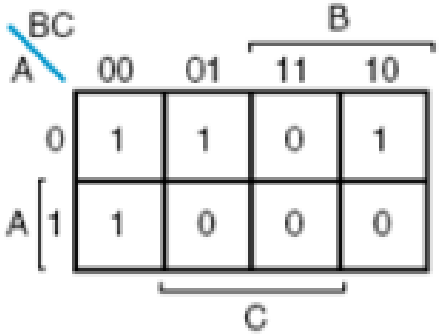
Source: Digital Fundamentals, Floyd

- 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 $x\ XOR\ 1 = \bar{x}$).
- 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.

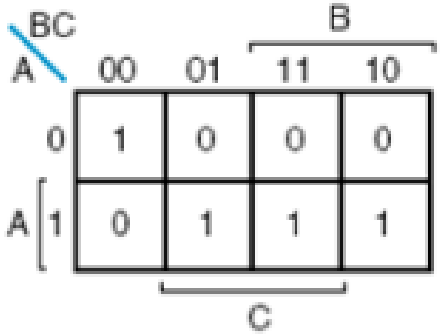
A	B	C	F1	f2
0	0	0	1	1
0	0	1	1	0
0	1	0	1	0
0	1	1	0	0
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	0	1

Step 1: K-maps for simplification:



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

$\overline{F}_1 = AB + AC + BC$

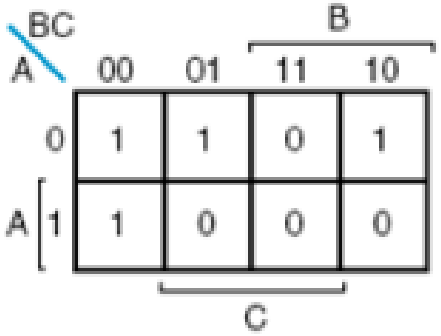


$F_2 = AB + AC + \overline{A}\overline{B}\overline{C}$

$\overline{F}_2 = \overline{A}\overline{C} + \overline{A}\overline{B} + A\overline{B}\overline{C}$

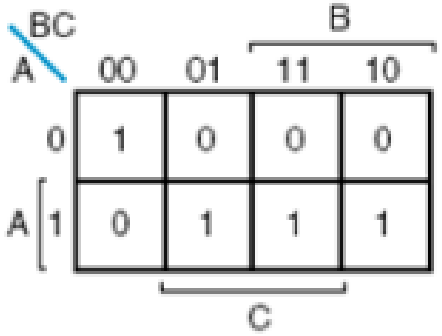
- Designing using a PLA, a careful investigation must be taken in order to reduce the distinct product terms.
- Both the true and complement forms of each function should be simplified to see which one can be expressed with fewer product terms and which one provides product terms that are common to other functions.

- K-maps for simplification:



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

$\overline{F}_1 = AB + AC + BC$



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

$\overline{F}_2 = \overline{A}C + \overline{A}B + A\overline{B}\,\overline{C}$

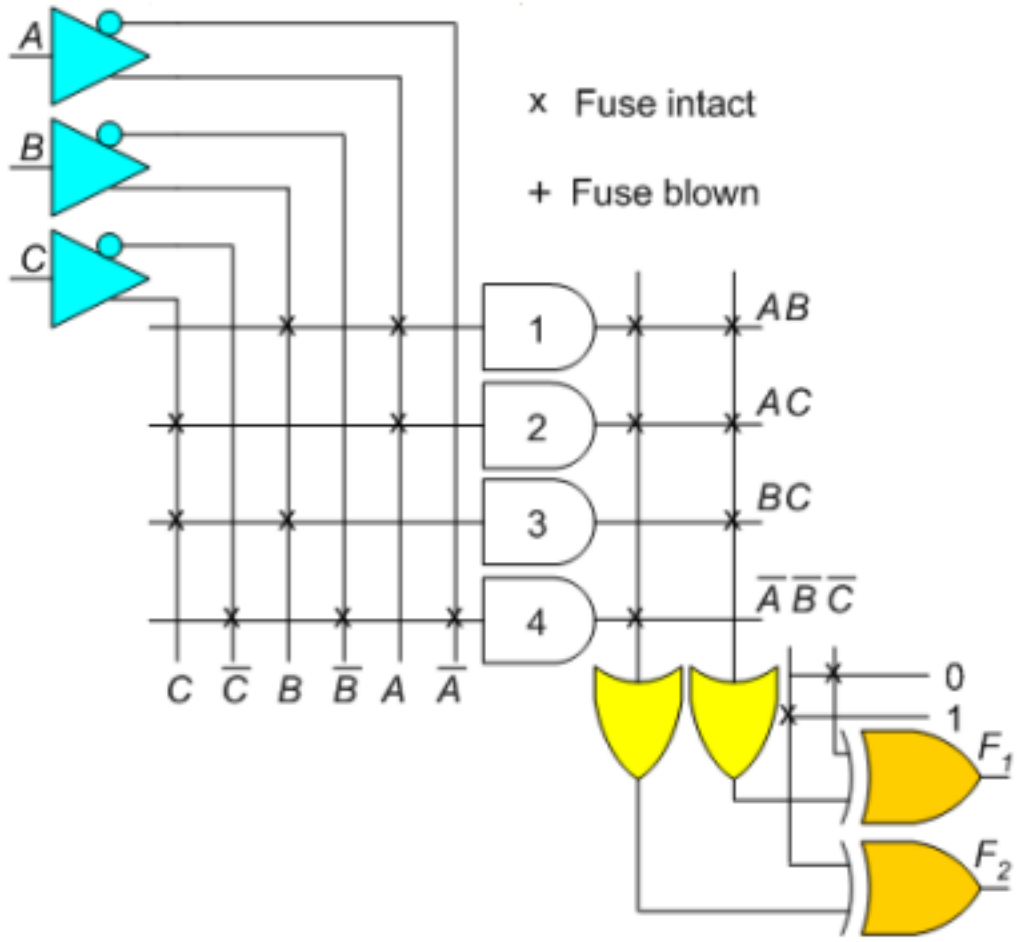
The combination that gives a minimum number of product terms is:

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

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

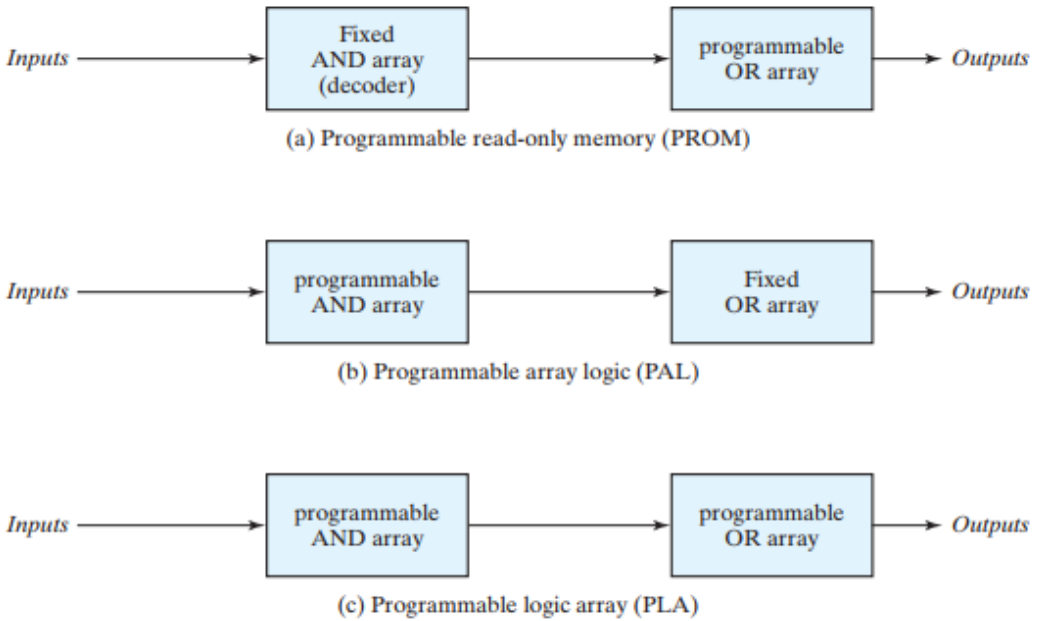
This gives only 4 distinct product terms: AB, AC, BC, and $A'B'C'$.

Step 2: Logic Design

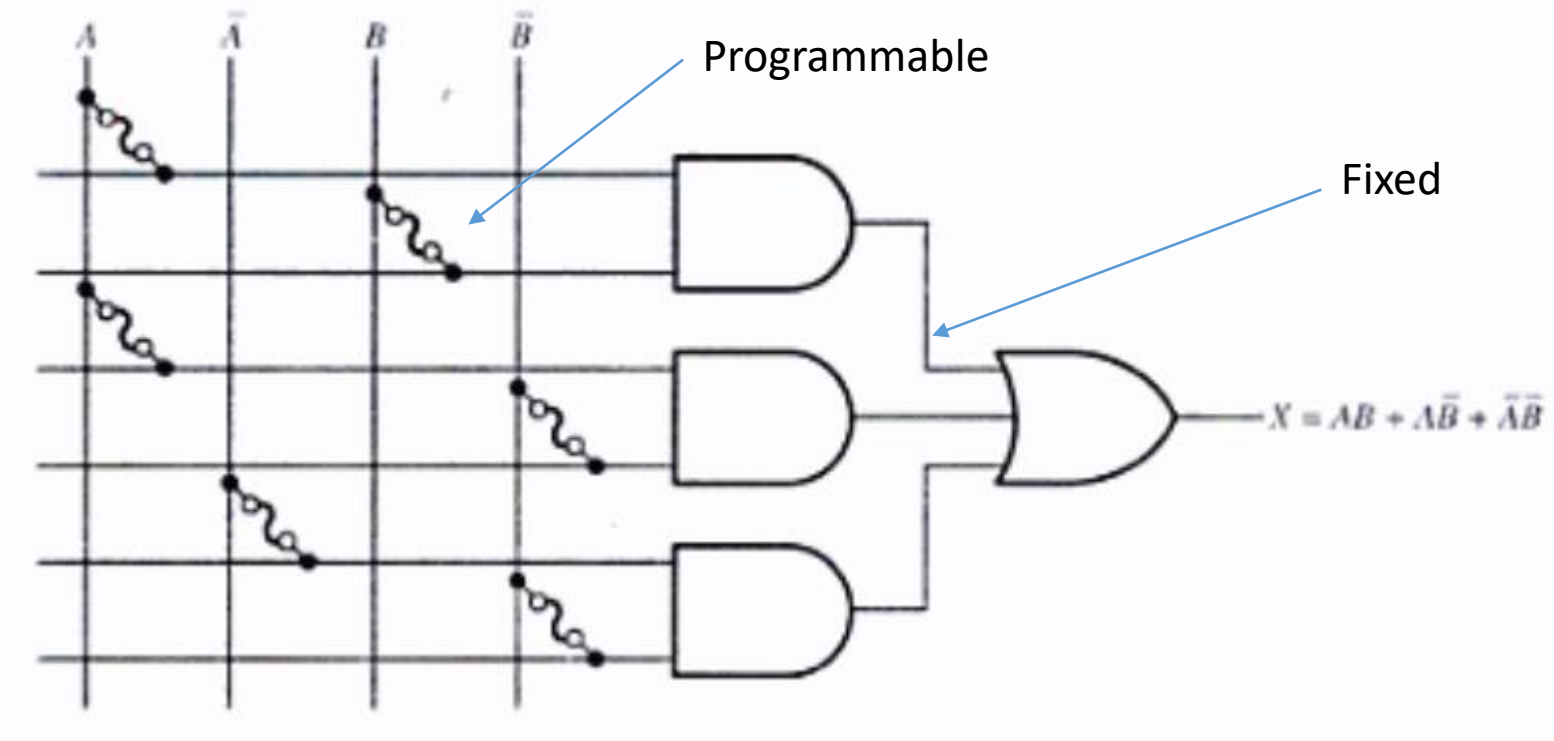


Programmable Array Logic (PAL)

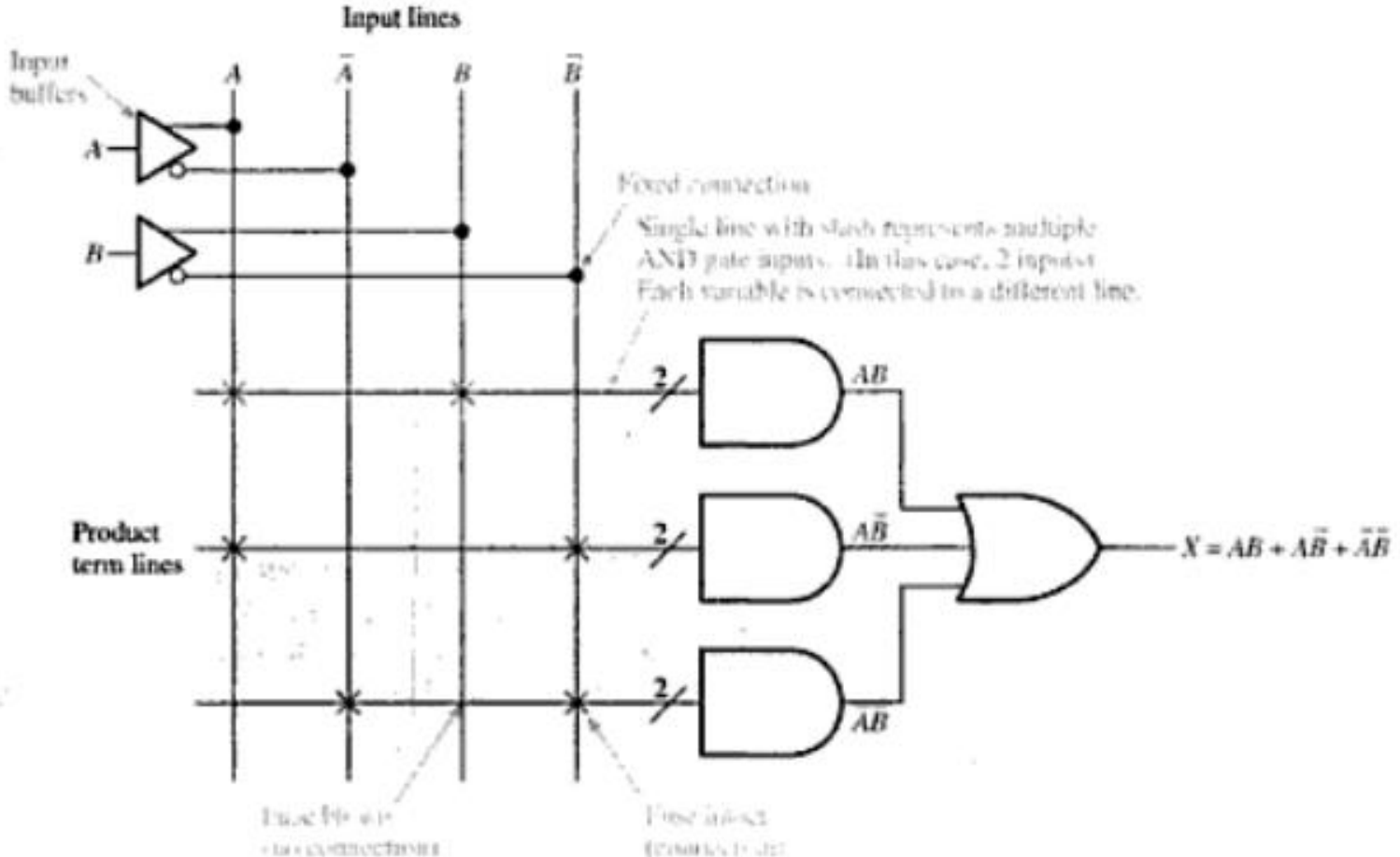
- It has programmable AND array and fixed OR array.
- Because only the AND array is programmable, it is easier to use
- However, it is not flexible as compared to Programmable Logic Array (PLA).



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



Source: Digital Fundamentals, Floyd



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:

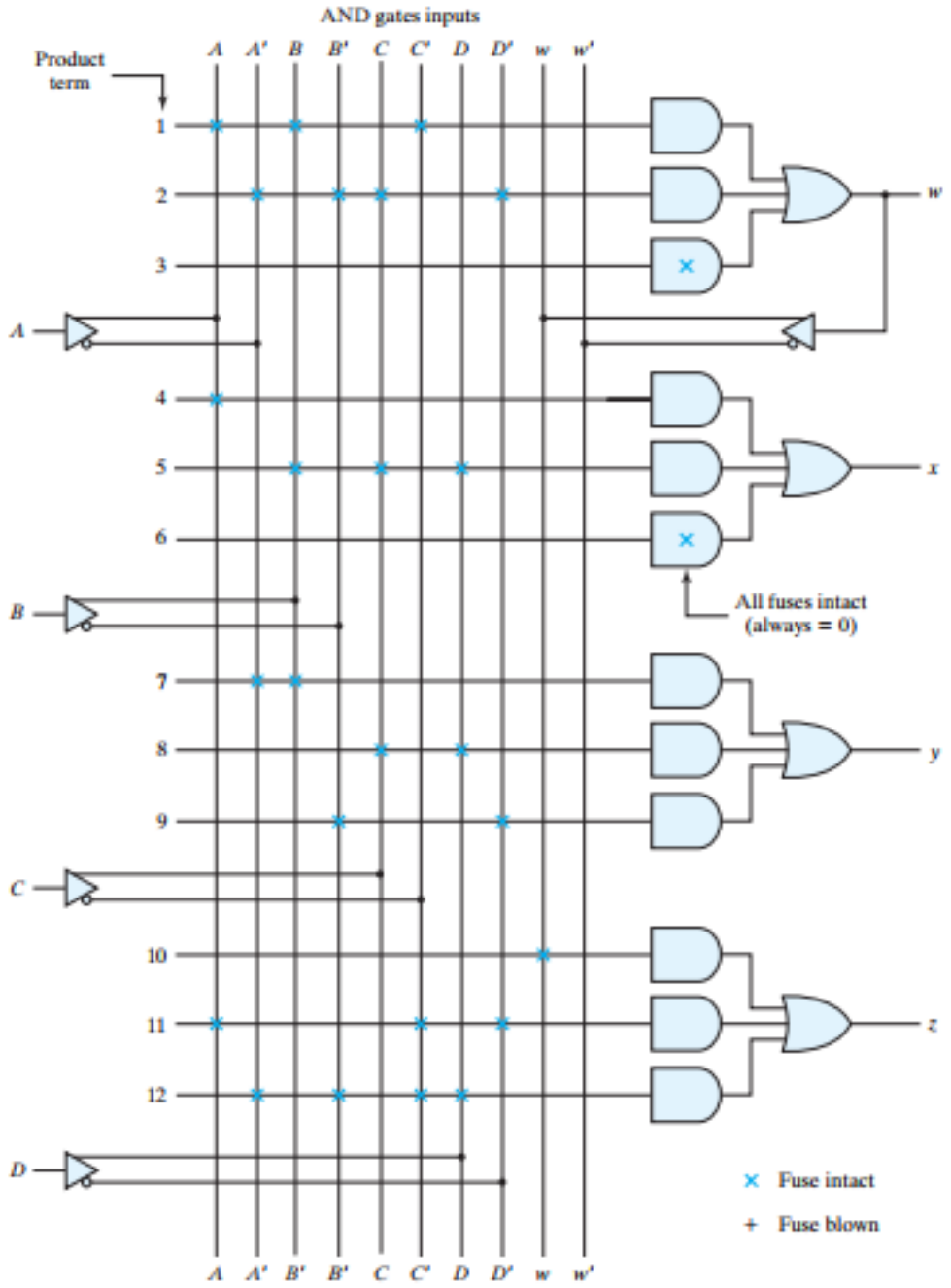
Use K-map for Simplification

$$w(A, B, C, D) = \sum(2, 12, 13)$$
$$x(A, B, C, D) = \sum(7, 8, 9, 10, 11, 12, 13, 14, 15)$$
$$y(A, B, C, D) = \sum(0, 2, 3, 4, 5, 6, 7, 8, 10, 11, 15)$$
$$z(A, B, C, D) = \sum(1, 2, 8, 12, 13)$$

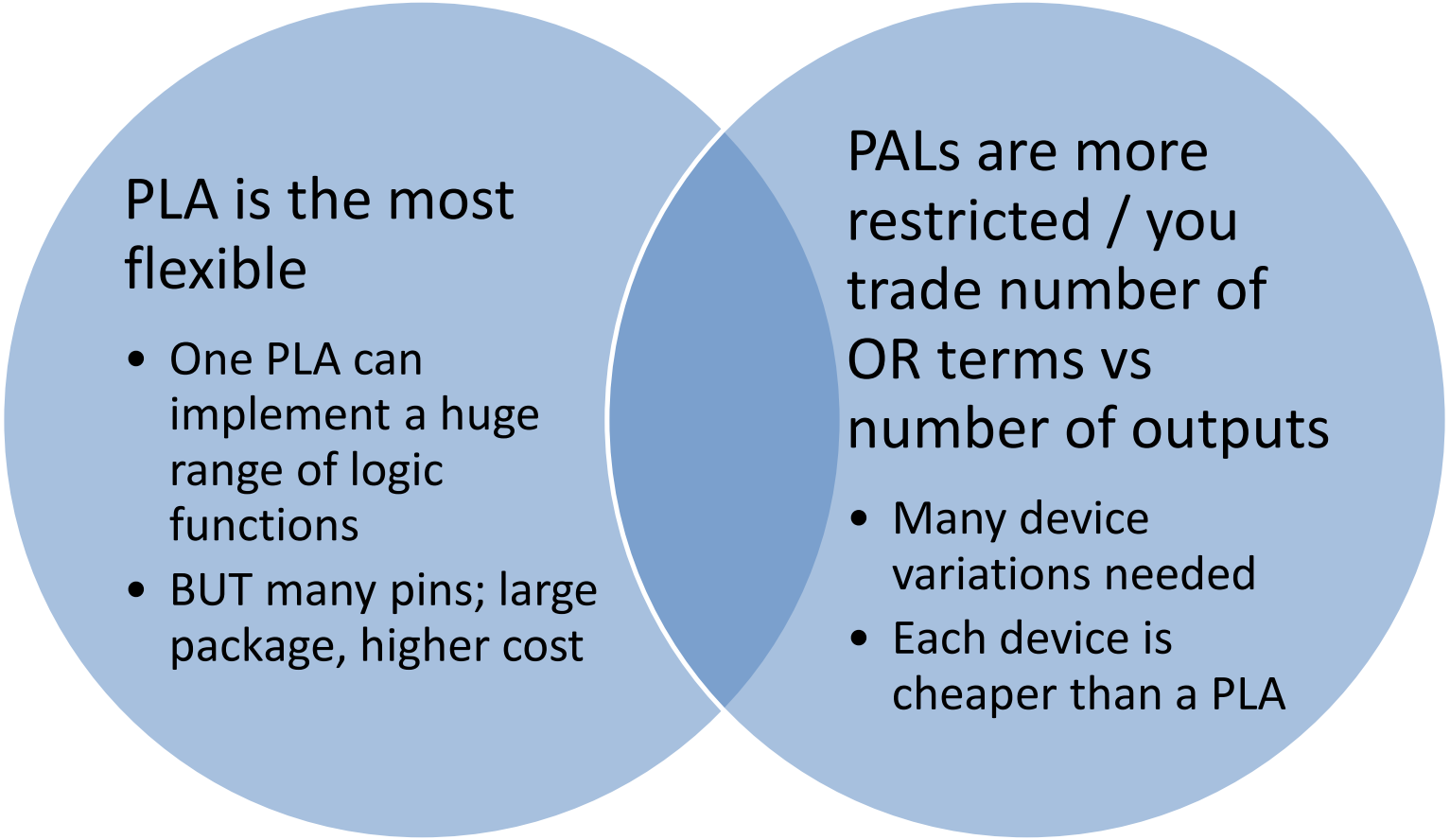
Simplifying the four functions to a minimum number of terms results in the following Boolean functions:

$$w = ABC' + A'B'CD'$$
$$x = A + BCD$$
$$y = A'B + CD + B'D'$$
$$z = ABC' + A'B'CD' + AC'D' + A'B'C'D$$
$$= w + AC'D' + A'B'C'D$$

LOGICAL DIAGRAM for PAL:



PALs and PLAs



Field Programmable Gate Arrays (FPGAs)

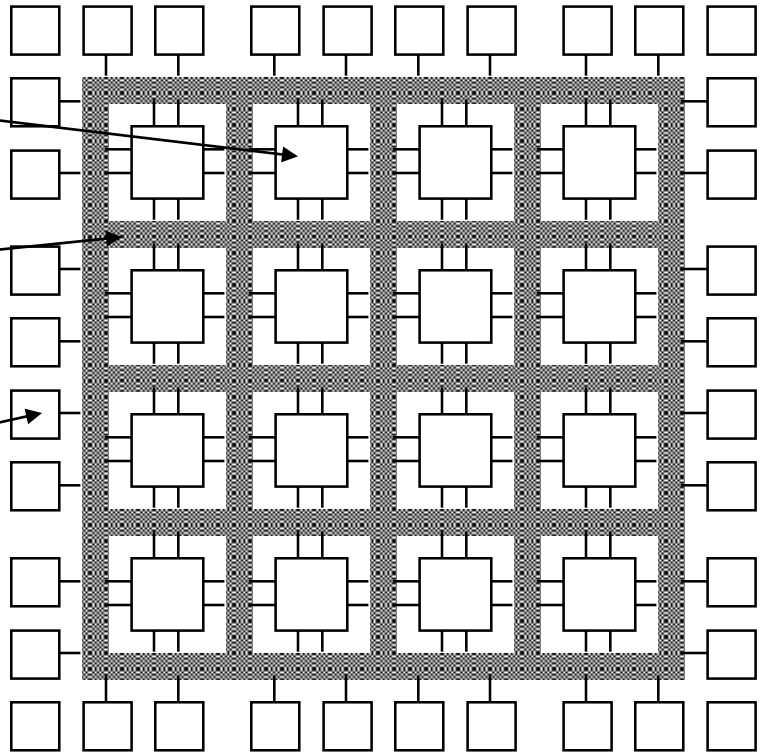
- FPGAs have much more logic than CPLDs
  - 2K to >10M equivalent gates
  - Requires different architecture
  - FPGAs can be RAM-based or Flash-based
    - RAM FPGAs must be programmed at power-on
      - External memory needed for programming data
      - May be dynamically reconfigured
    - Flash FPGAs store program data in non-volatile memory
      - Reprogramming is more difficult
      - Holds configuration when power is off

FPGA Structure

- Typical organization in 2-D array
  - Configurable logic blocks (CLBs) contain functional logic (could be similar to PAL22V10)
    - Combinational functions plus FFs
    - Complexity varies by device
  - CLB interconnect is either local or long line
    - CLBs have connections to local neighbors
    - Horizontal and vertical channels use for long distance
    - Channel intersections have switch matrix
  - IOBs (I/O logic Blocks) connect to pins
    - Usually have some additional C.L./FF in block

Field-Programmable Gate Arrays structure

- Logic blocks
  - To implement combinational and sequential logic
- Interconnect
  - Wires to connect inputs and outputs to logic blocks
- I/O blocks
  - Special logic blocks at periphery of device for external connections



Quick Quiz

- The PLD which is more flexible

A. PLA
B. ROM
C. PAL
D. Fixed logic

Quick Quiz

- The size of the PLA is specified by the

A. Number of product terms
B. Number of inputs
C. Number of the outputs
D. All of the above

Quick Quiz

- PAL consists of following matrix

A. Fixed AND matrix and a programmable OR matrix
B. Programmable AND matrix and a fixed OR matrix
C. Both AND and OR matrix are fixed
D. Both AND and OR matrix are programmable