Lecture – 9

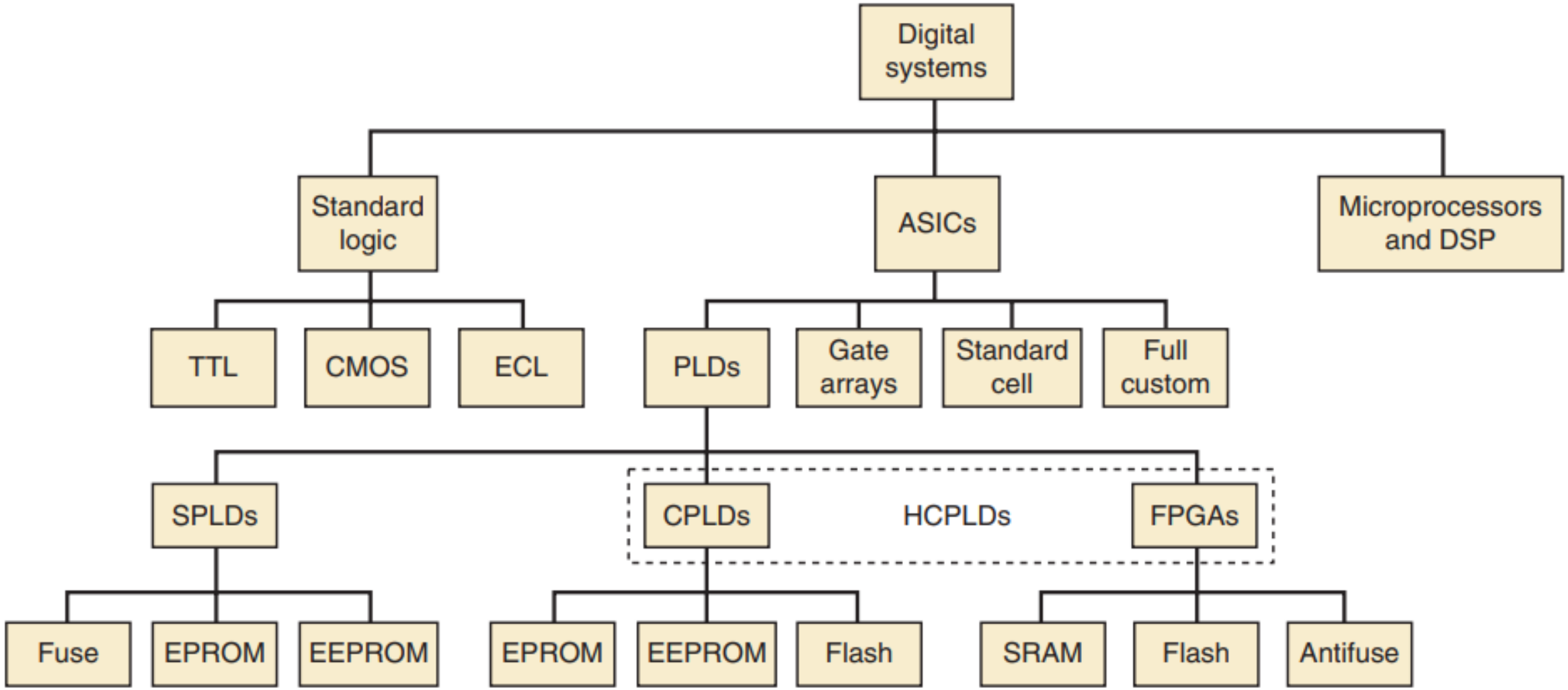
Programmable Logic Devices

Lesson Outcomes

After completing this lecture, students will be able to

- *Describe the difference between an PAL and GAL*
- *Explain the principle of operation of PAL and PLA*
- *Explain the difference of SPLD and CPLD*
- *Apply PAL and PLA for CPLDs, for example, PROM and FPGA*

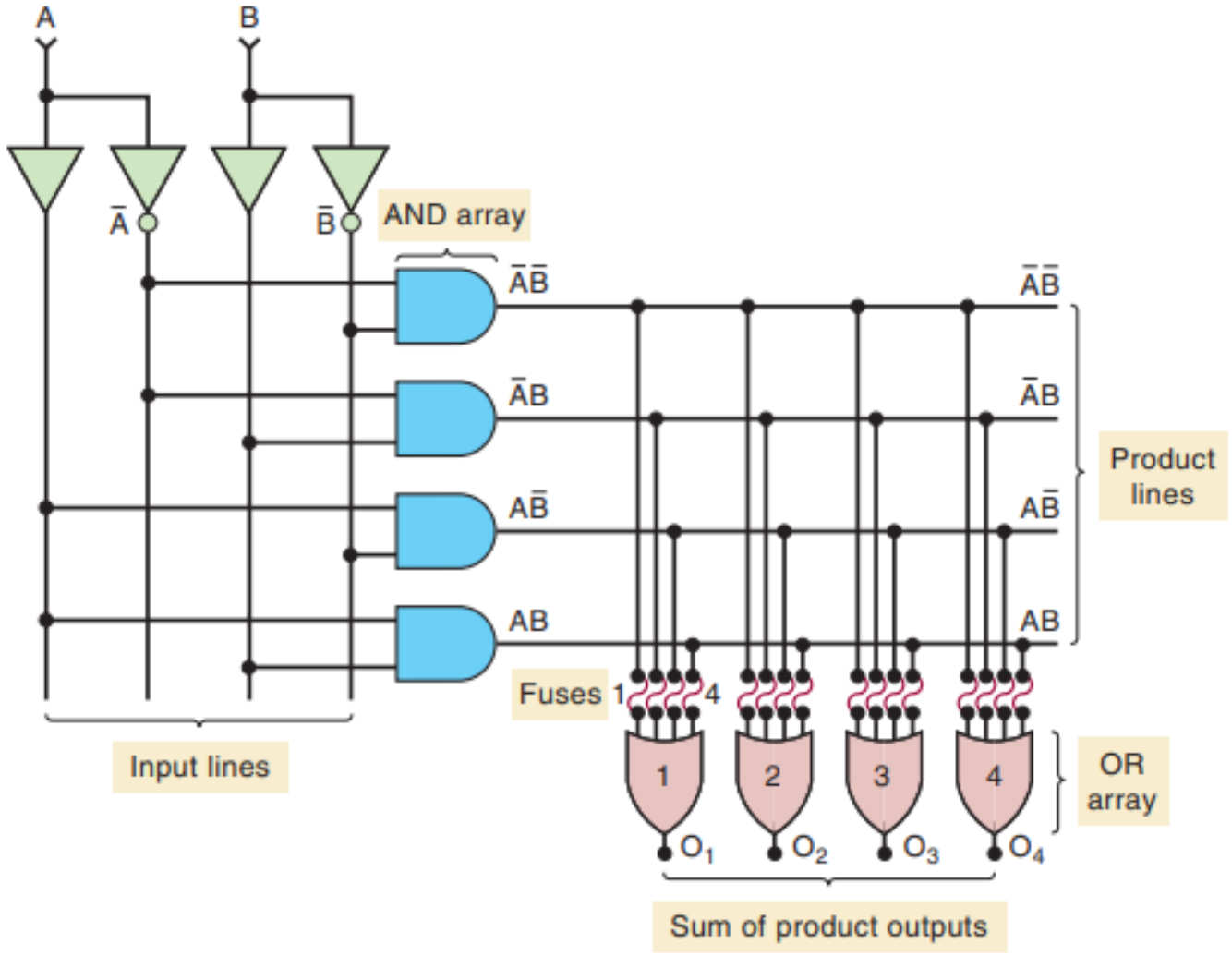
Digital Systems Family Tree



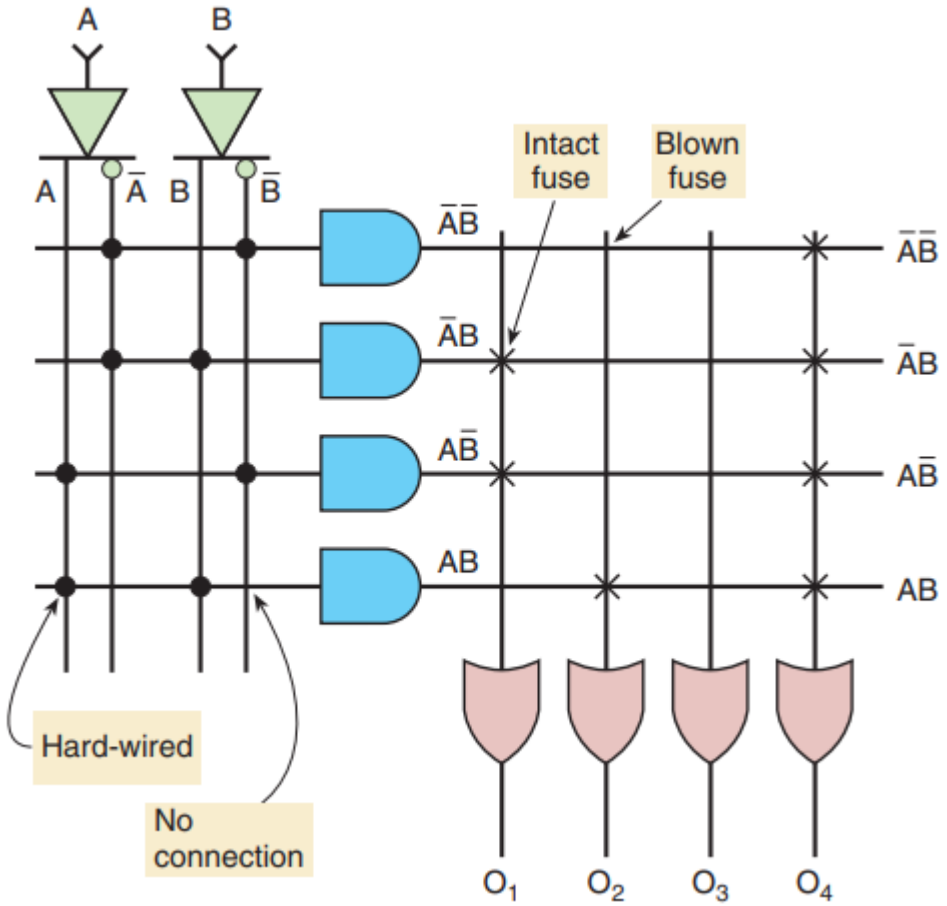
Digital Systems Family Tree

TTL : Transistor Transistor Logic
CMOS: Complementary Metal-Oxide Semiconductor
ECL : Emitter Coupled Logic
SPLD : Simple Programmable Logic Device
CPLD : Complex Programmable Logic Device
HCPLD: High Capacity Programmable Logic Device
PAL : Programmable Array Logic
GAL : Generic Array Logic
PLA : Programable Logic Array
FPGA : Field Programable Gate Array
EPROM: Erasable Programmable Read Only Memory
EEPROM: Electrically Erasable Programmable Read Only Memory
SRAM: Static Random Access Memory

PLDs (Programmable Logic Devices) – symbology

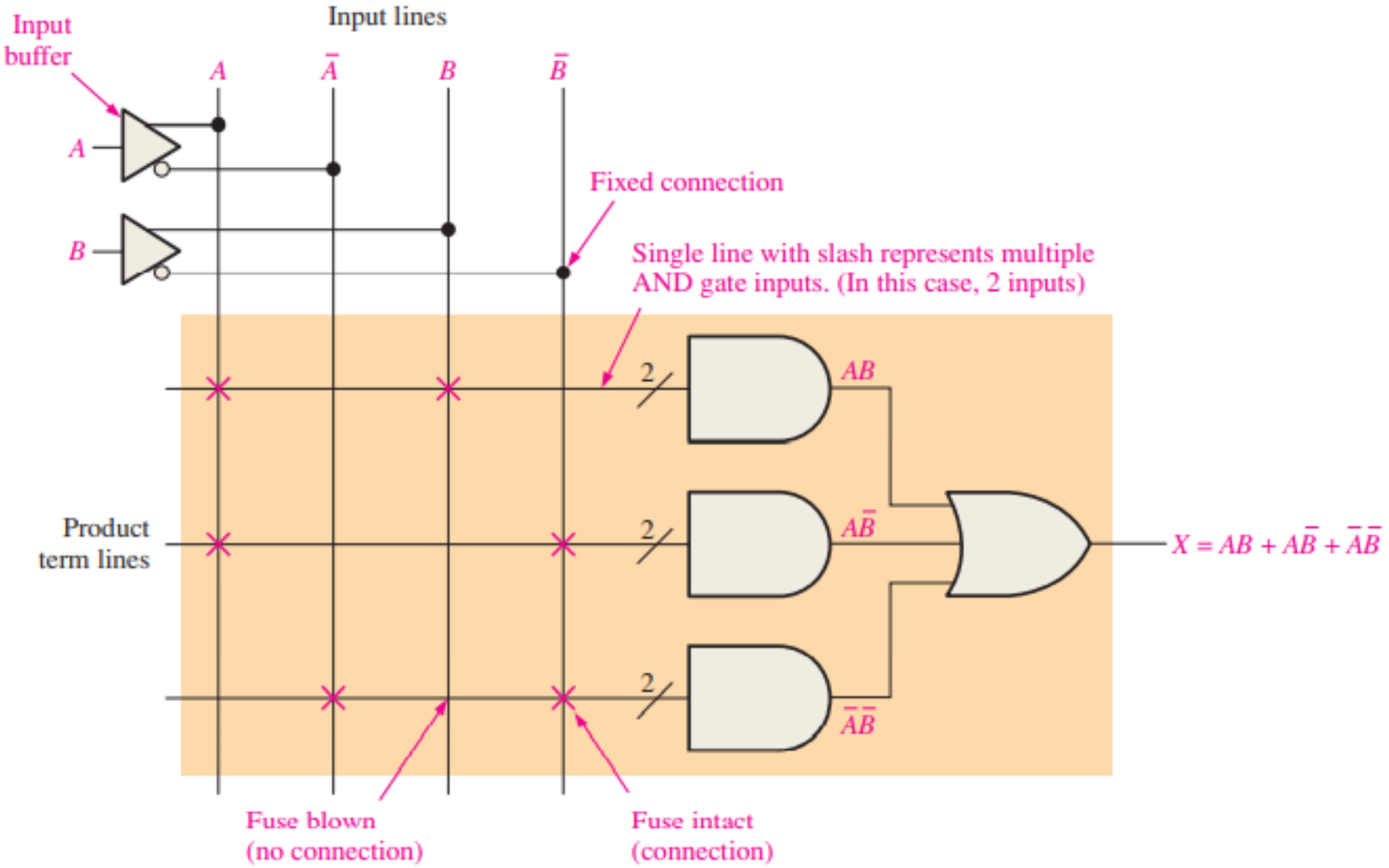


Programmable Logic Devices - example



Simplified PLD Symbology

PLAs – Programmable Logic Arrays

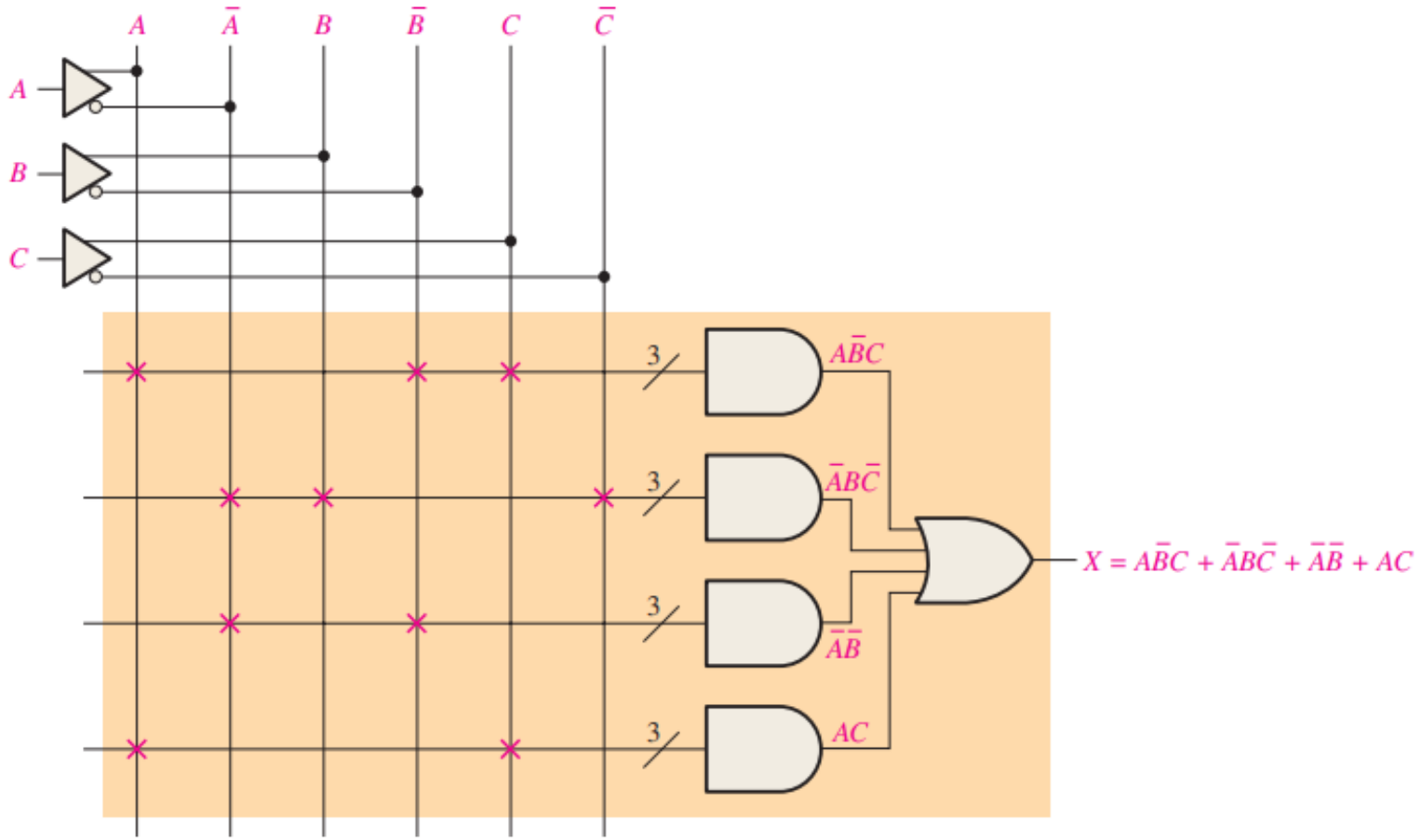


PLAs – Example

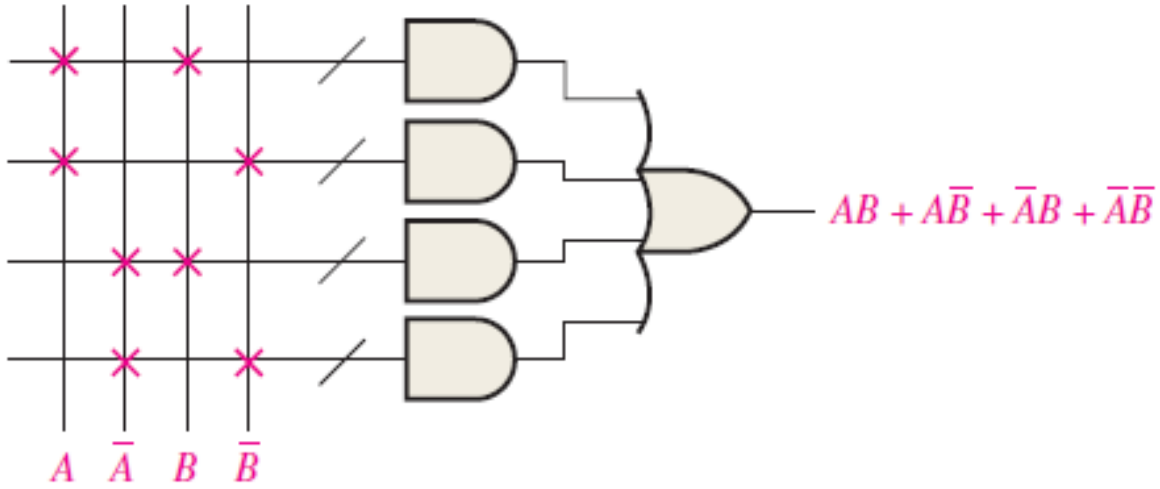
EXAMPLE 10–1

Show how a PAL is programmed for the following 3-variable logic function: $X = A\overline{B}C + \overline{A}B\overline{C} + \overline{A}\,\overline{B} + AC$

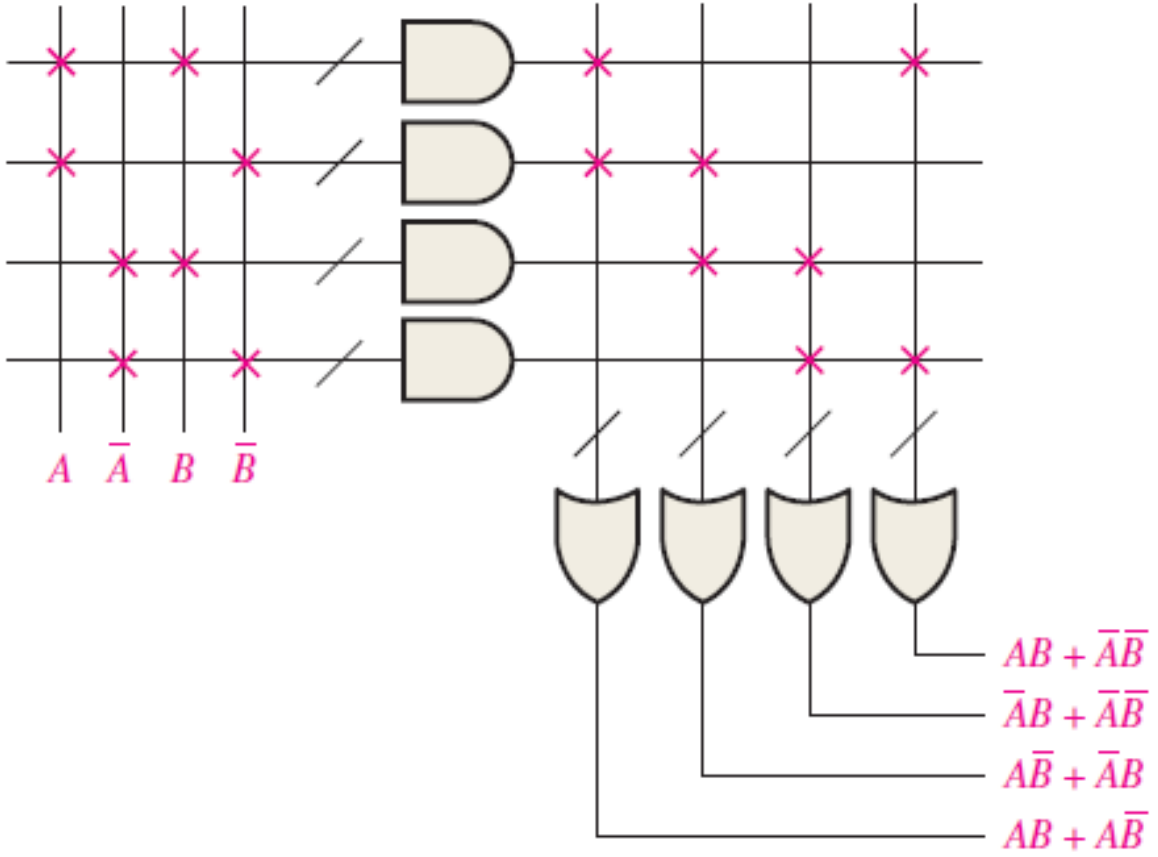
Solution



PAL AND PLA: basic difference



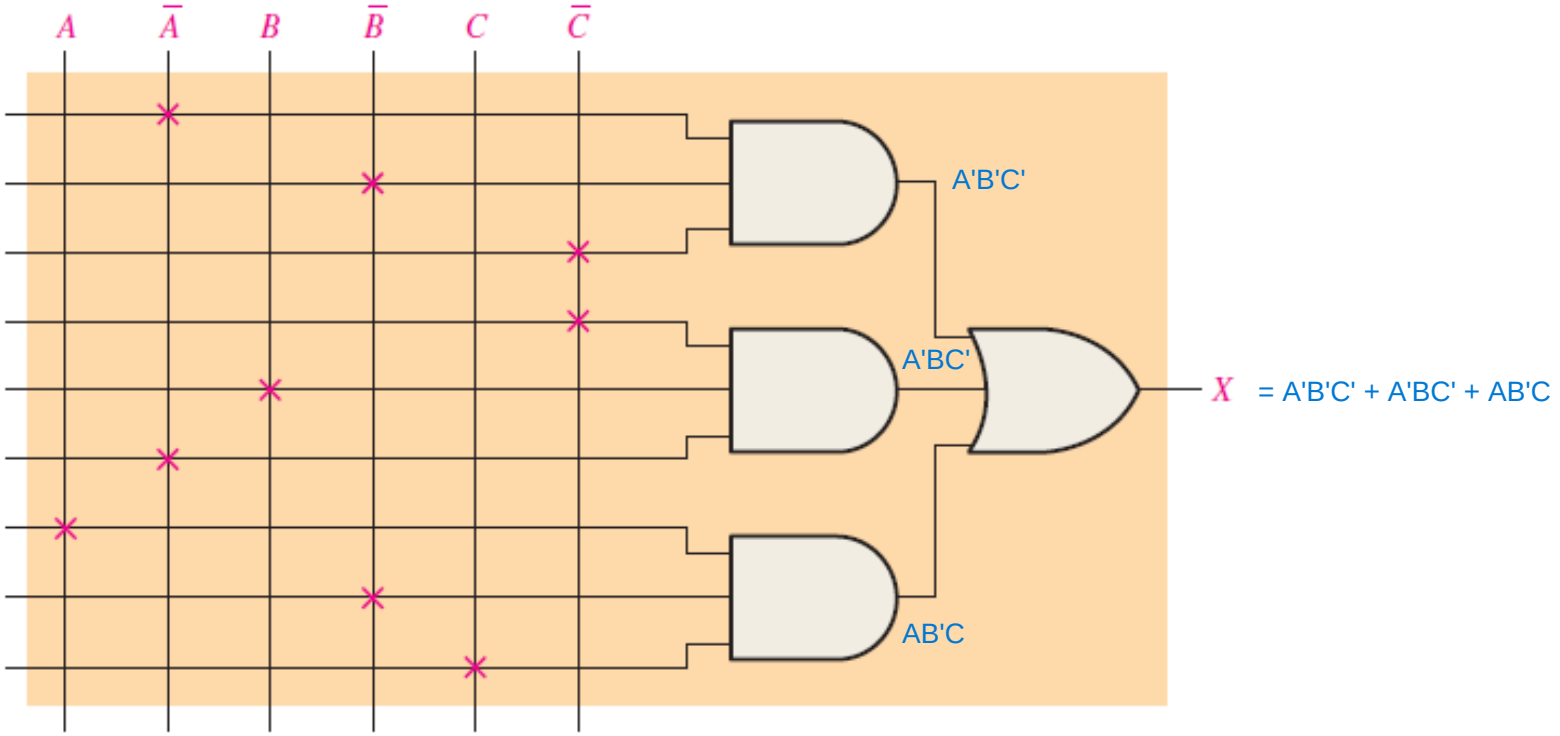
(a) PAL-type array



(b) PLA-type array

Example

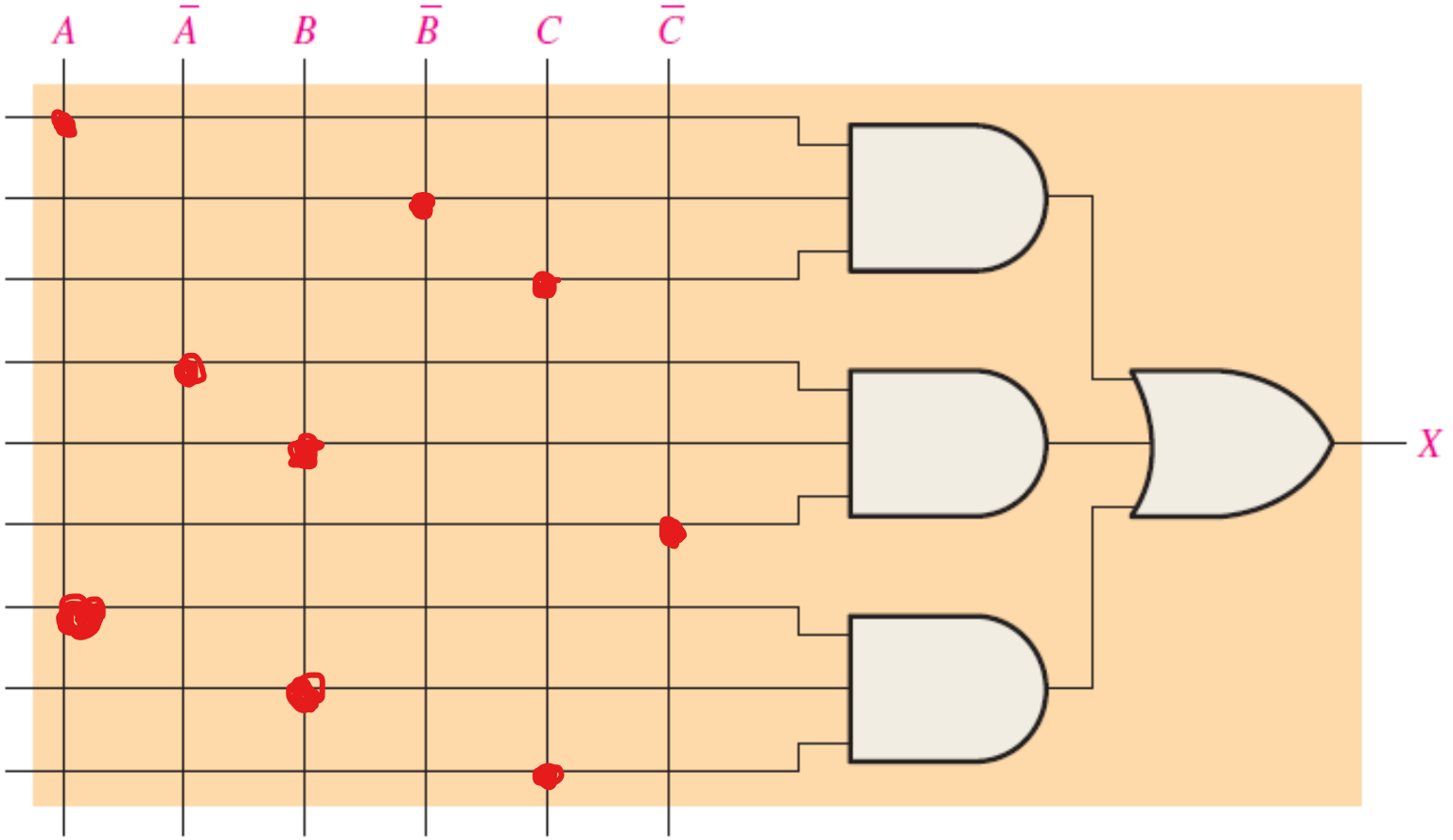
PROBLEM. Determine the Boolean output expression for the simple PAL array shown in Figure. The Xs represent connected links.



Example

PROBLEM. Show how the PAL-type array in Figure should be programmed to implement each of the following SOP expressions. Use an X to indicate a connected link.

(a) $Y = A\overline{B}C + \overline{A}B\overline{C} + ABC$
(b) $Y = A\overline{B}C + \overline{A}\,\overline{B}C + \overline{A}BC$

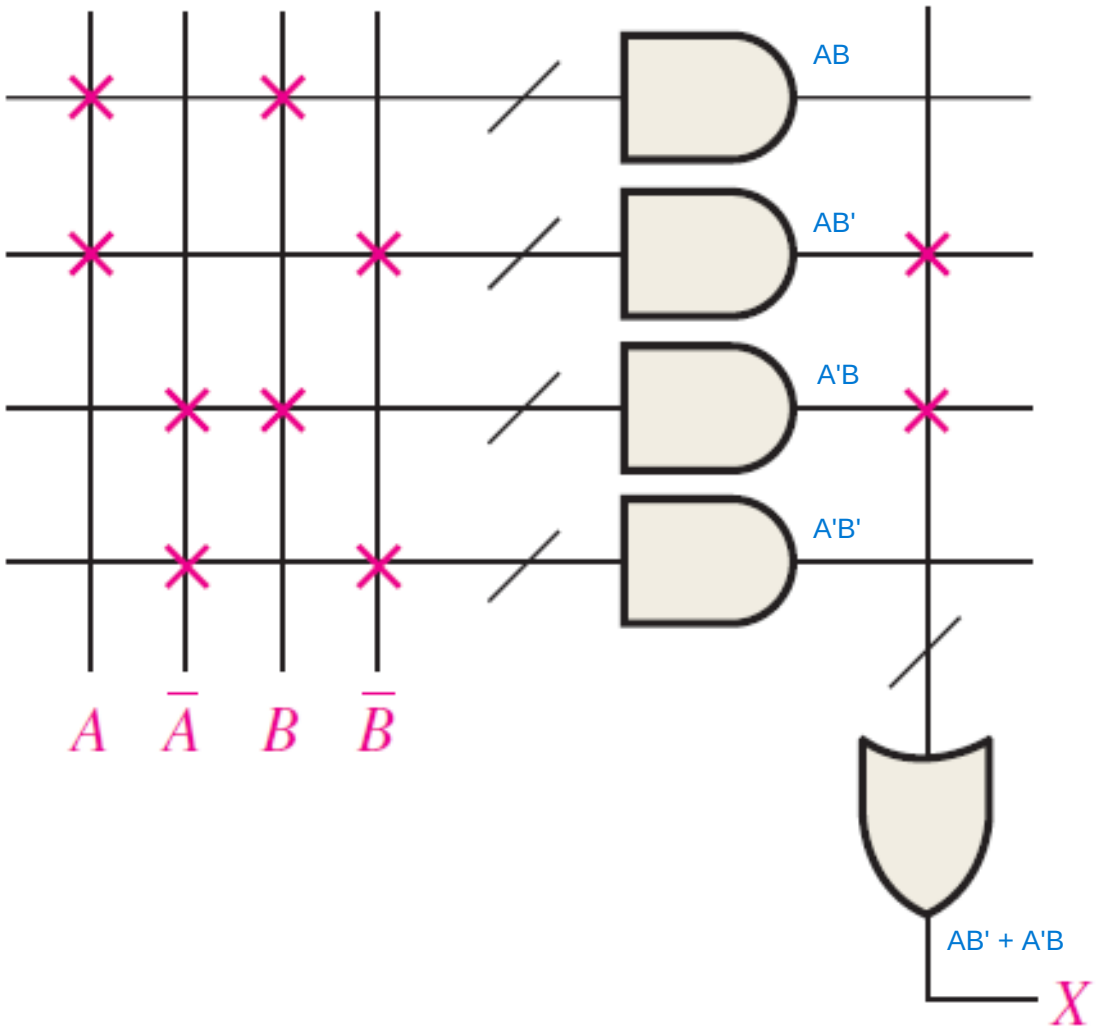


Example

PROBLEM. Determine the output of the array in Figure. The Xs represent connected links.

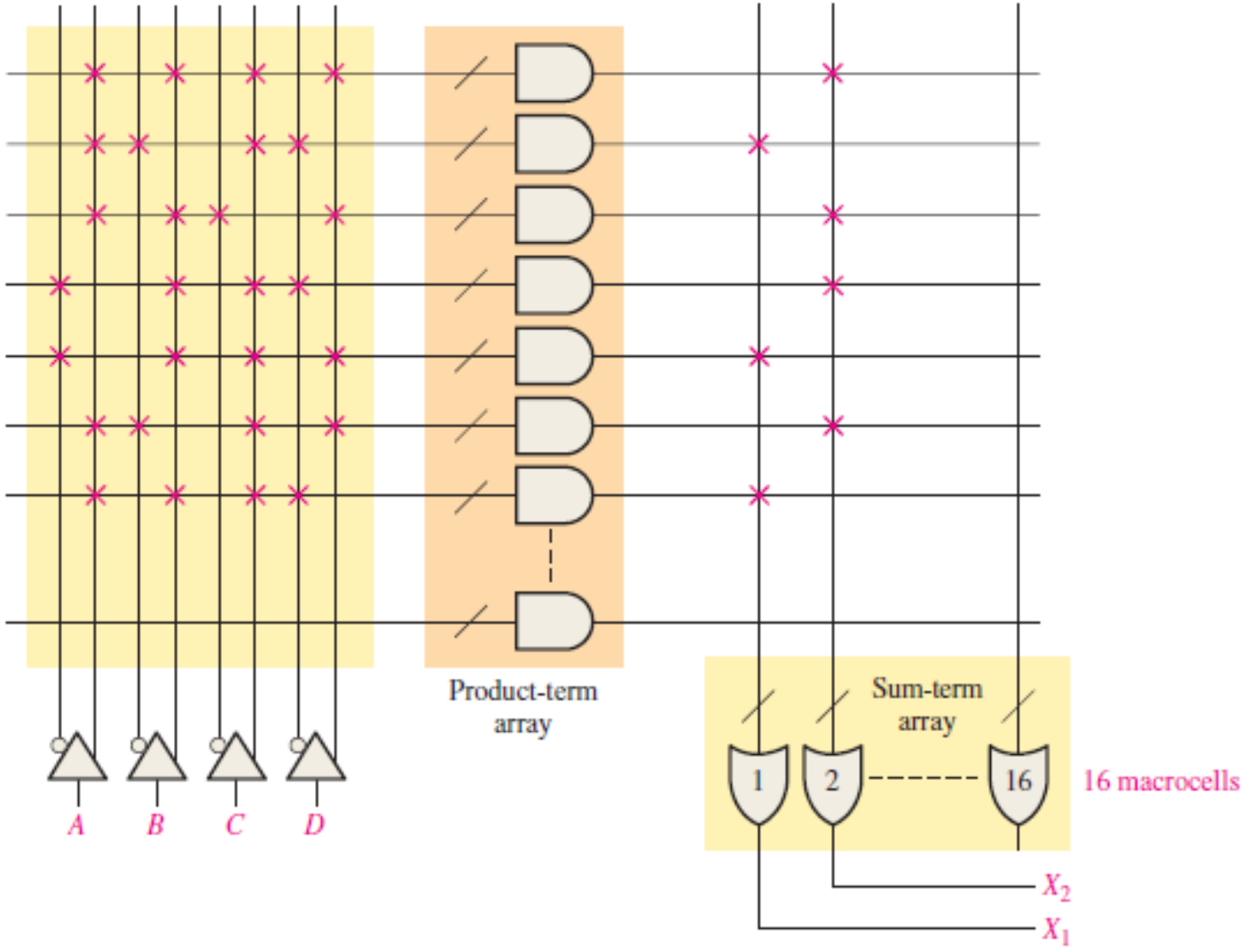
PROBLEM. Modify the array in Figure to produce an output

$$X = \overline{A}\,\overline{B}\,\overline{C} + \overline{A}\,B\,\overline{C} + ABC + A\,\overline{B}\,C$$

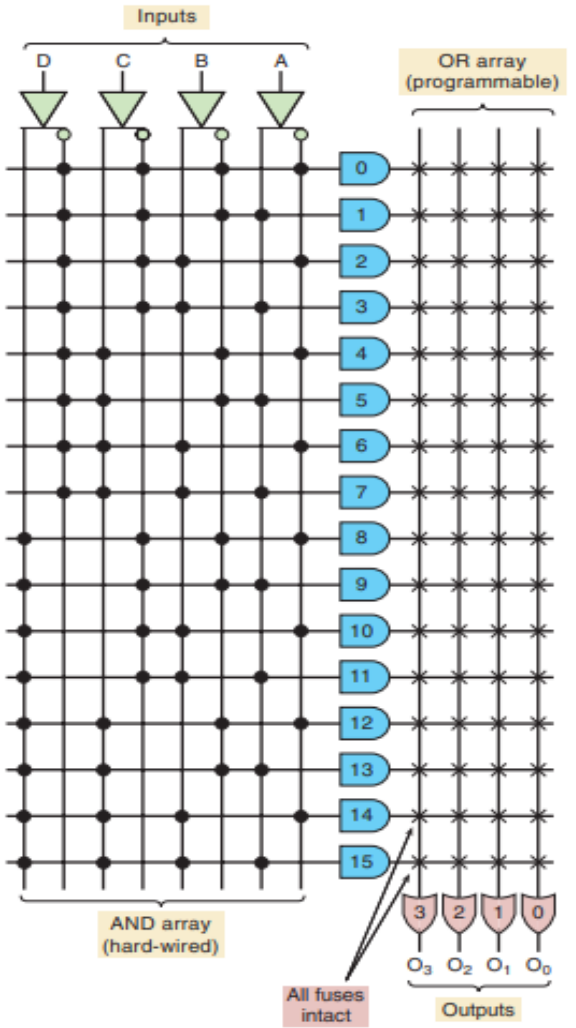


Example

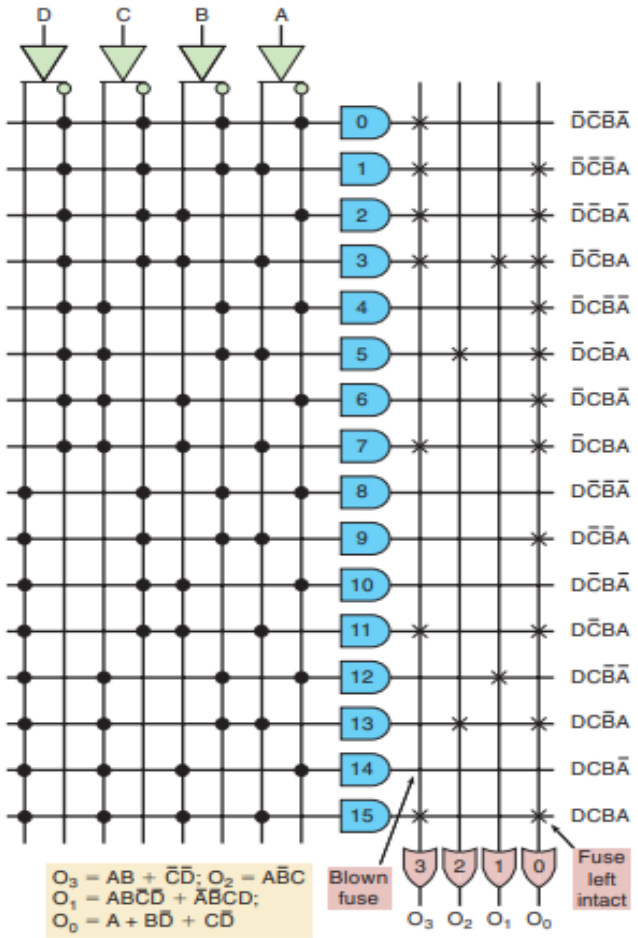
PROBLEM. Determine the output expressions for X_1 and X_2 from macrocells 1 and 2 in Figure.



PROM



(a) PROM architecture makes it suitable for PLDs

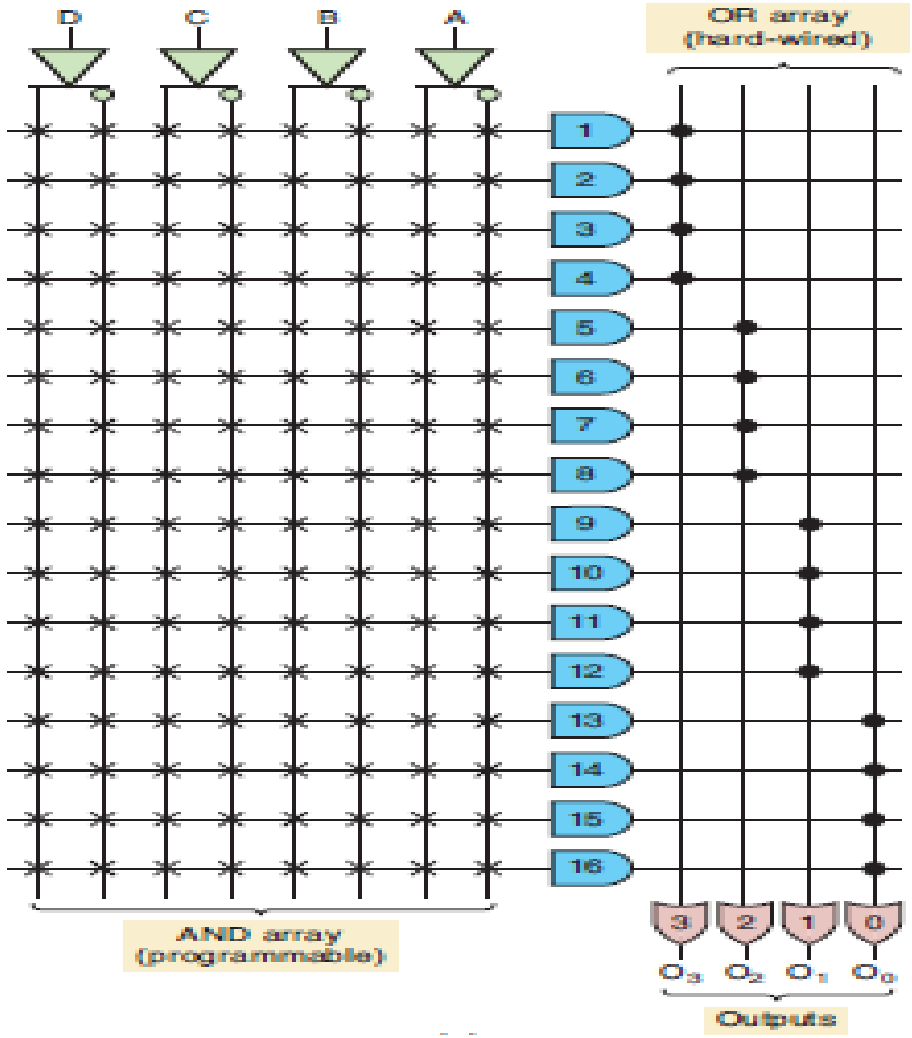


(b) fuses are blown to program outputs for given functions

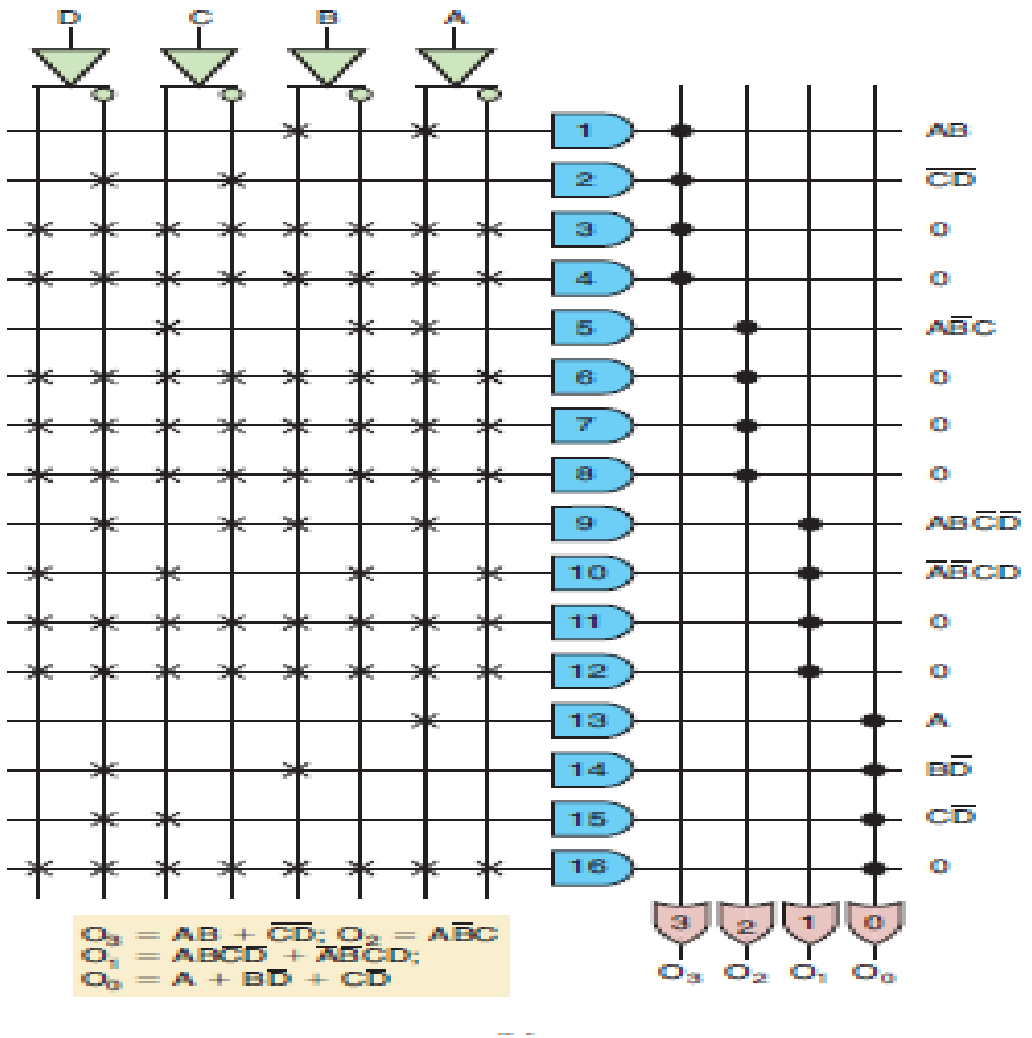
D	C	B	A	O_3		
0	0	0	0	1	→	$\bar{D}\bar{C}\bar{B}\bar{A}$
0	0	0	1	1	→	$\bar{D}\bar{C}\bar{B}A$
0	0	1	0	1	→	$\bar{D}\bar{C}B\bar{A}$
0	0	1	1	1	→	$\bar{D}\bar{C}BA$
0	1	0	0	0		
0	1	0	1	0		
0	1	1	0	0		
0	1	1	1	1	→	$\bar{D}CBA$
1	0	0	0	0		
1	0	0	1	0		
1	0	1	0	0		
1	0	1	1	1	→	$D\bar{C}BA$
1	1	0	0	0		
1	1	0	1	0		
1	1	1	0	0		
1	1	1	1	1	→	$DCBA$

A PROM used to generate a logic function on O3.

PROM – Programmable Logic Array

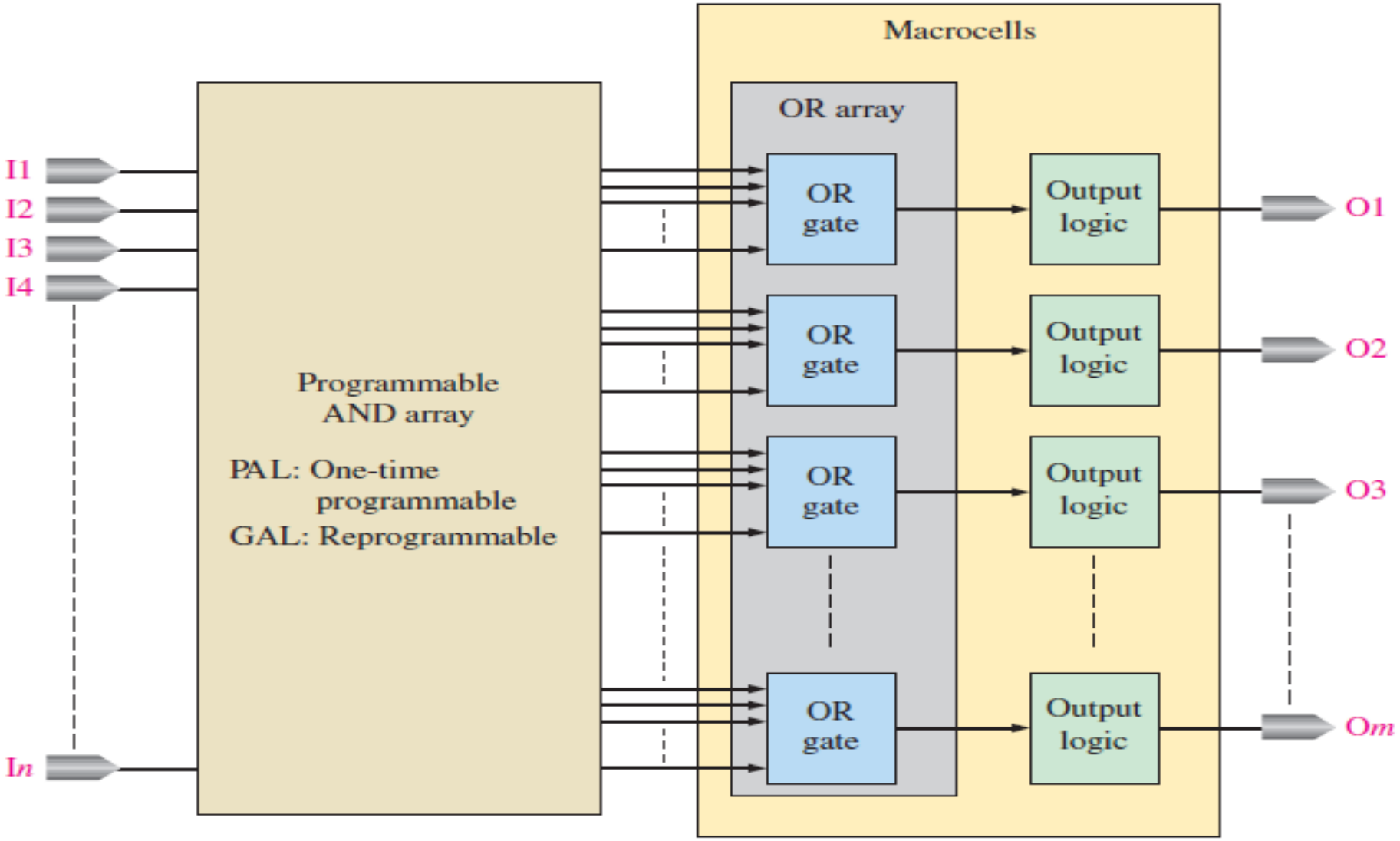


(a) Typical PAL architecture;

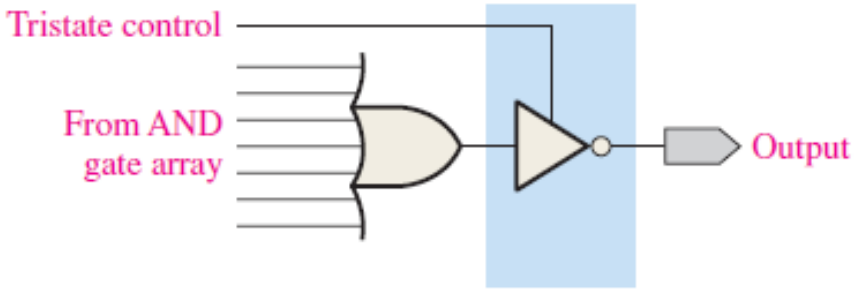


(b) the same PAL programmed for the given functions.

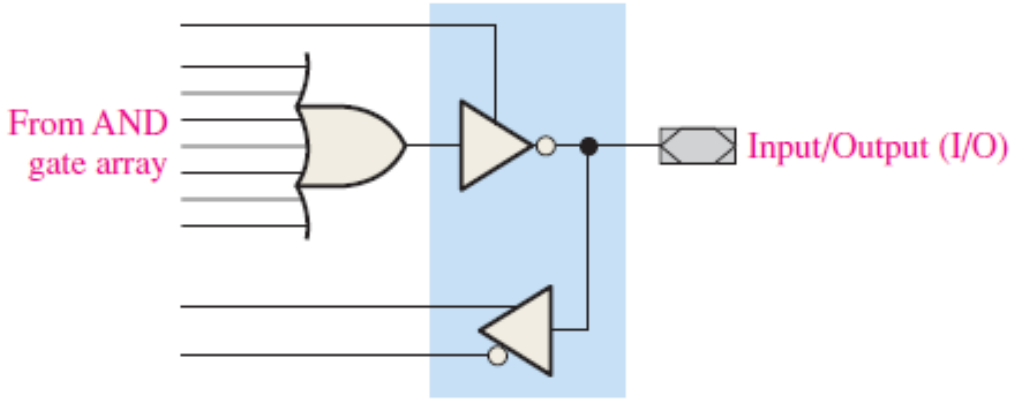
PAL / GAL – General Block Diagram



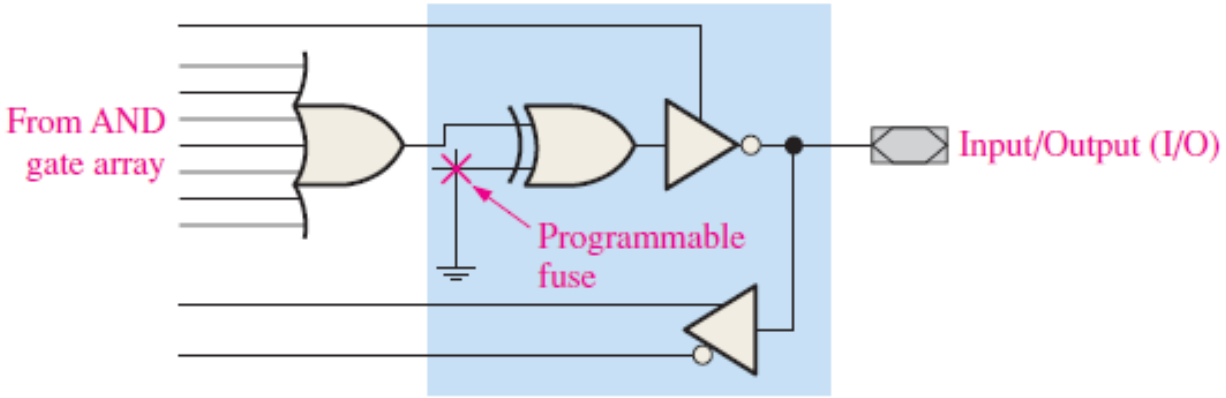
Basic types of PAL / GAL macrocells



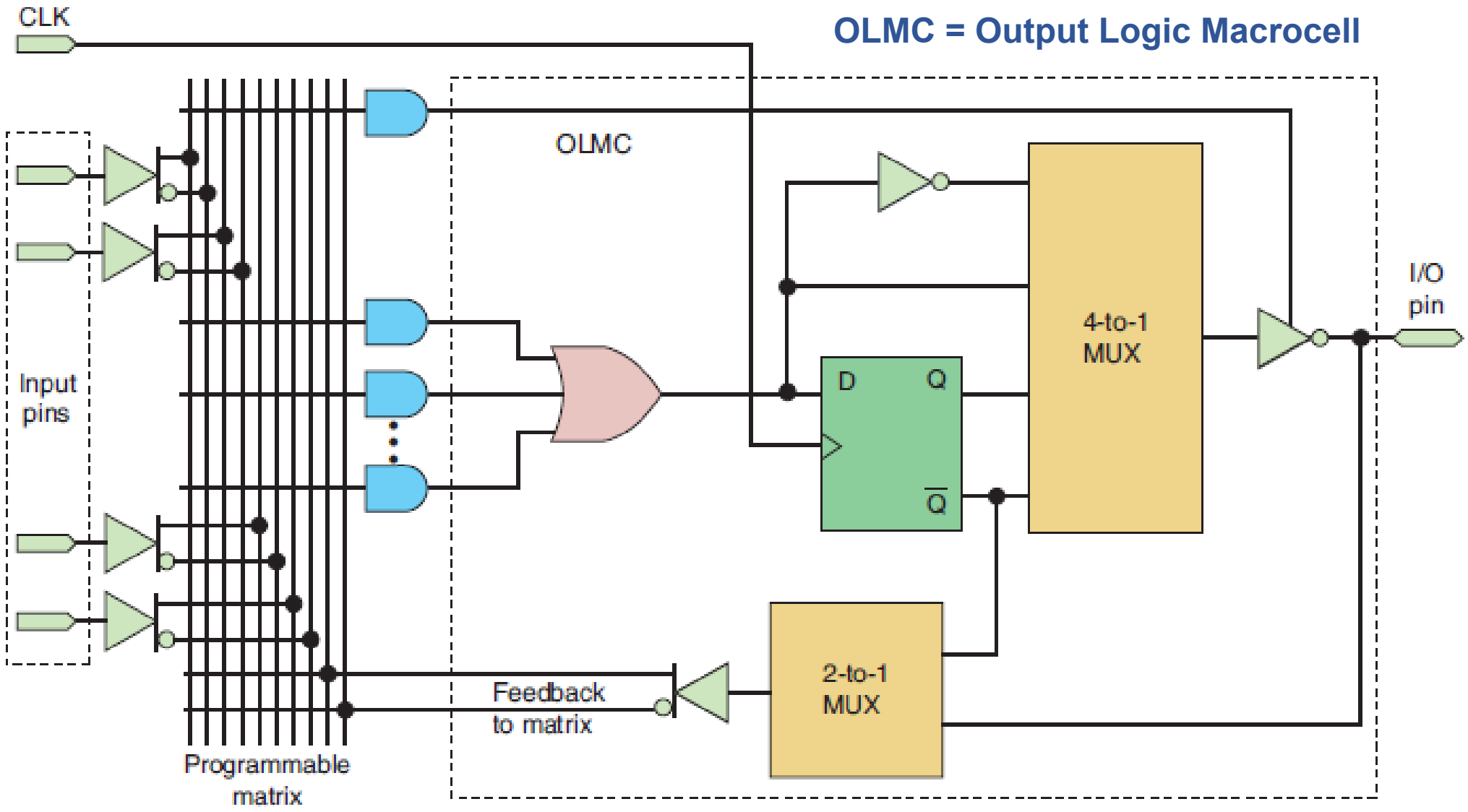
(a) Combinational output (active-LOW). An active-HIGH output would be shown without the bubble on the tristate gate symbol.



(b) Combinational input/output (active-LOW)



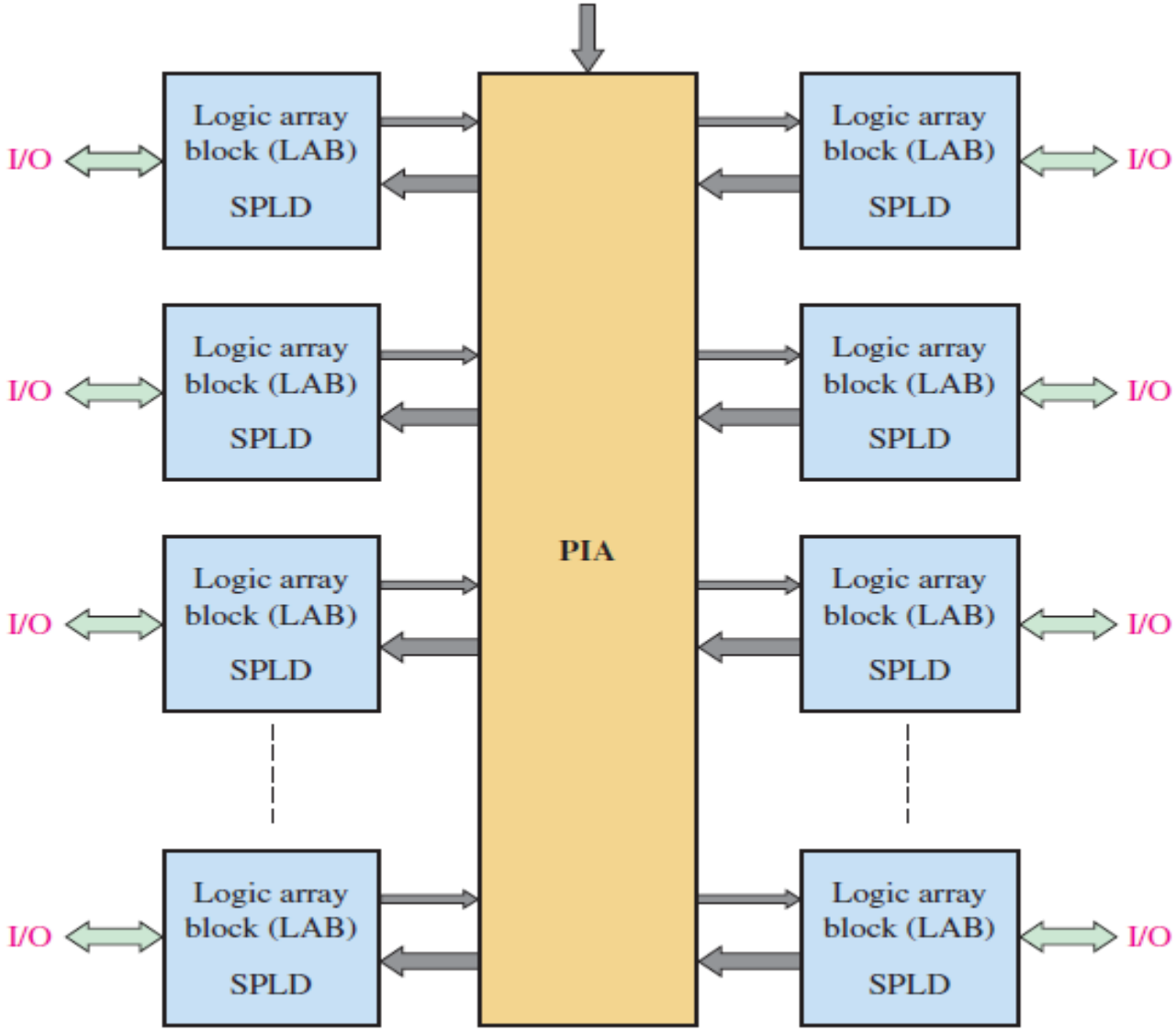
Programmable AND matrix and OLMC in GAL devices



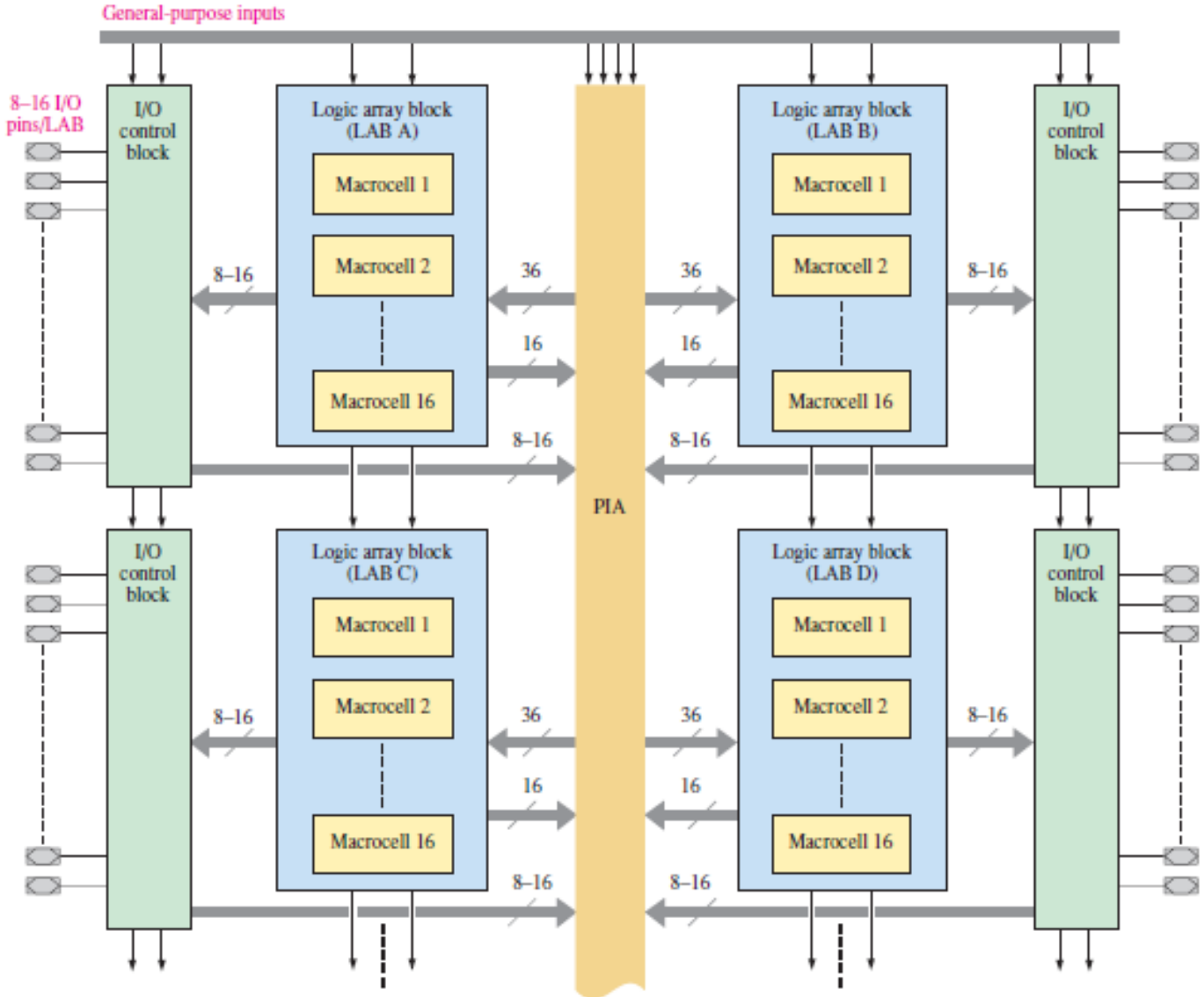
CPLD – Complex Programmable Logic Device

- A CPLD (complex programmable logic device) consists basically of multiple SPLD arrays with programmable interconnections.

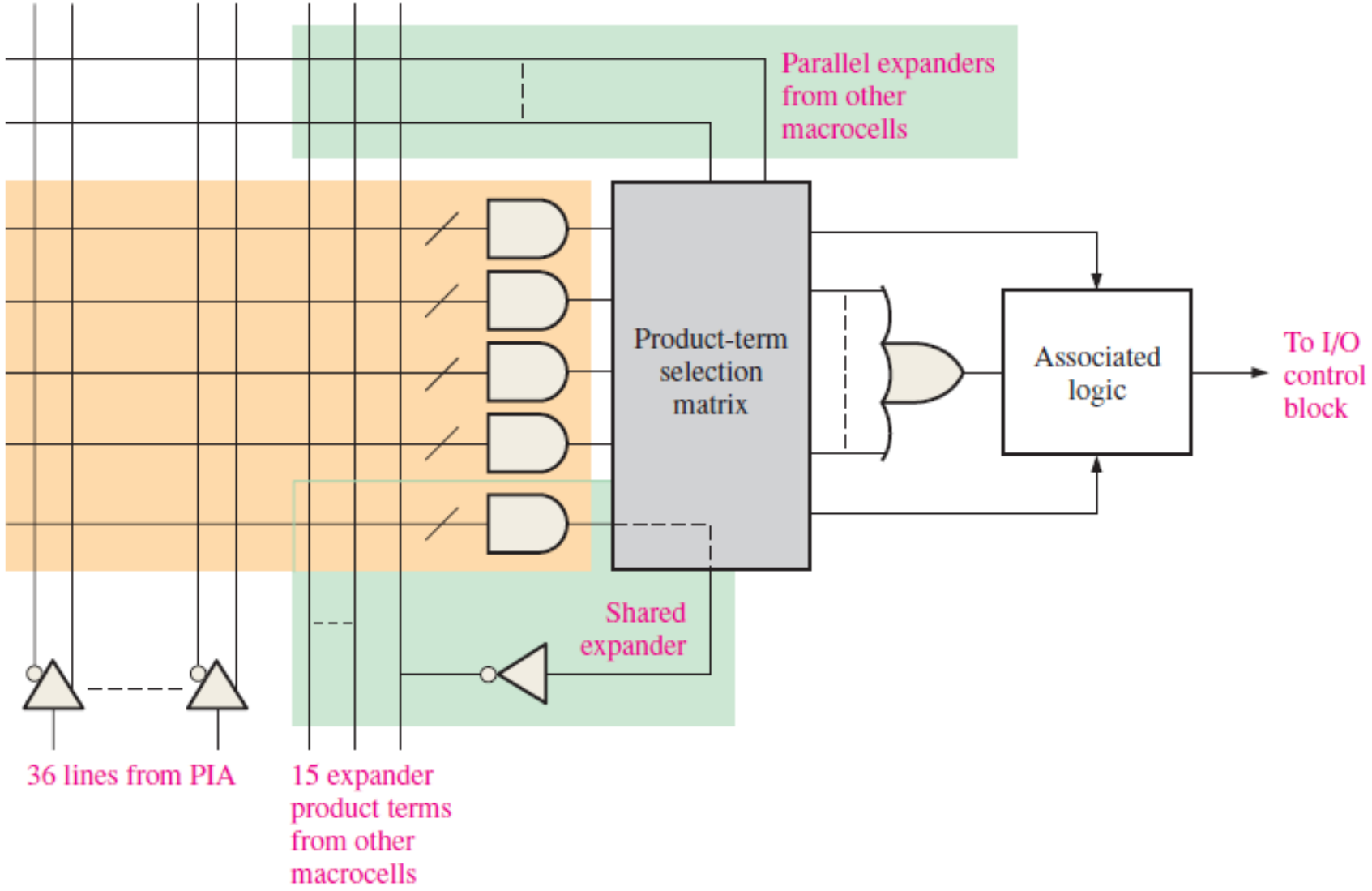
PIA = programmable interconnect array



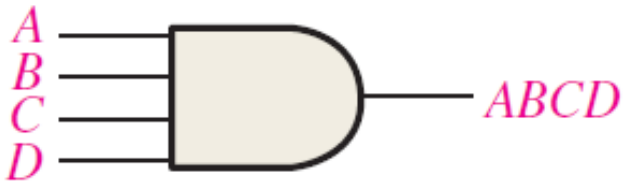
Block diagram of a typical CPLD



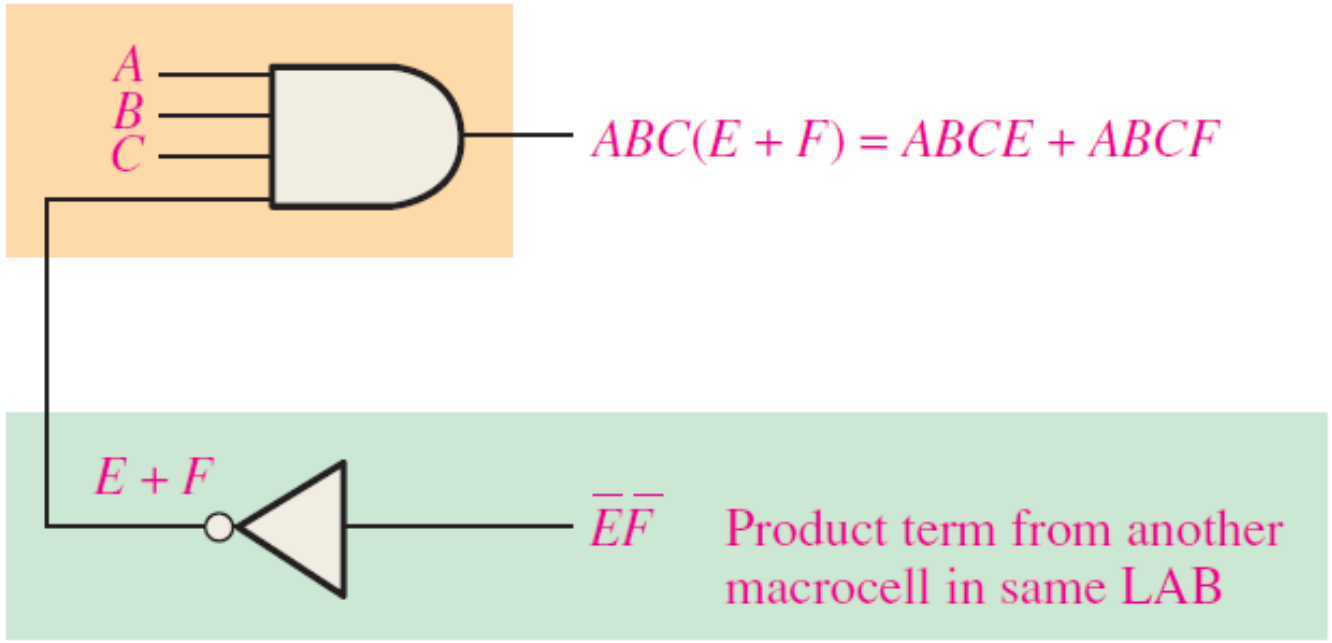
Simplified block diagram of a typical CPLD



Shared expander



(a) A 4-input AND array gate can produce one 4-variable product term.

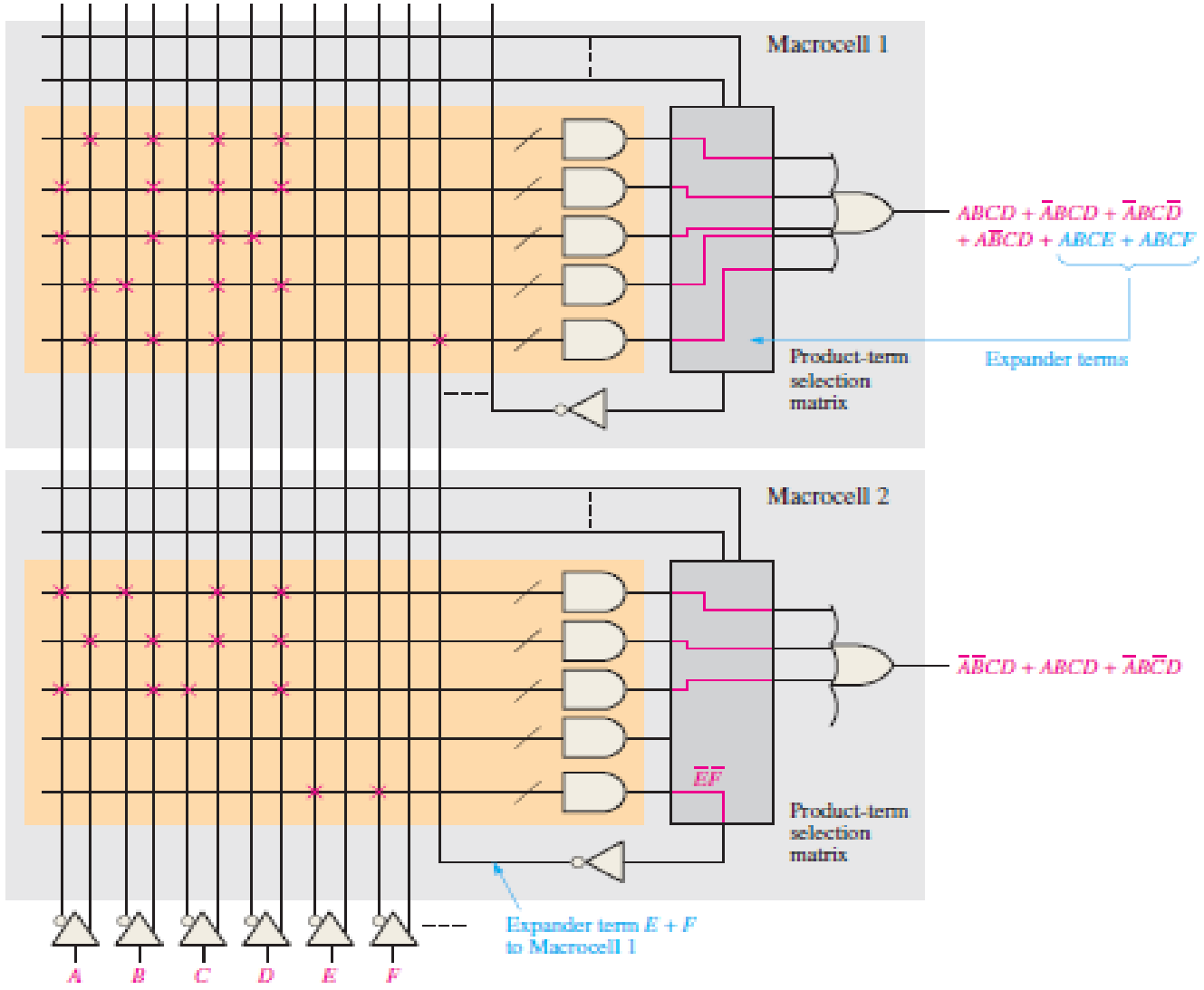


(b) AND gate is expanded to produce two product terms.

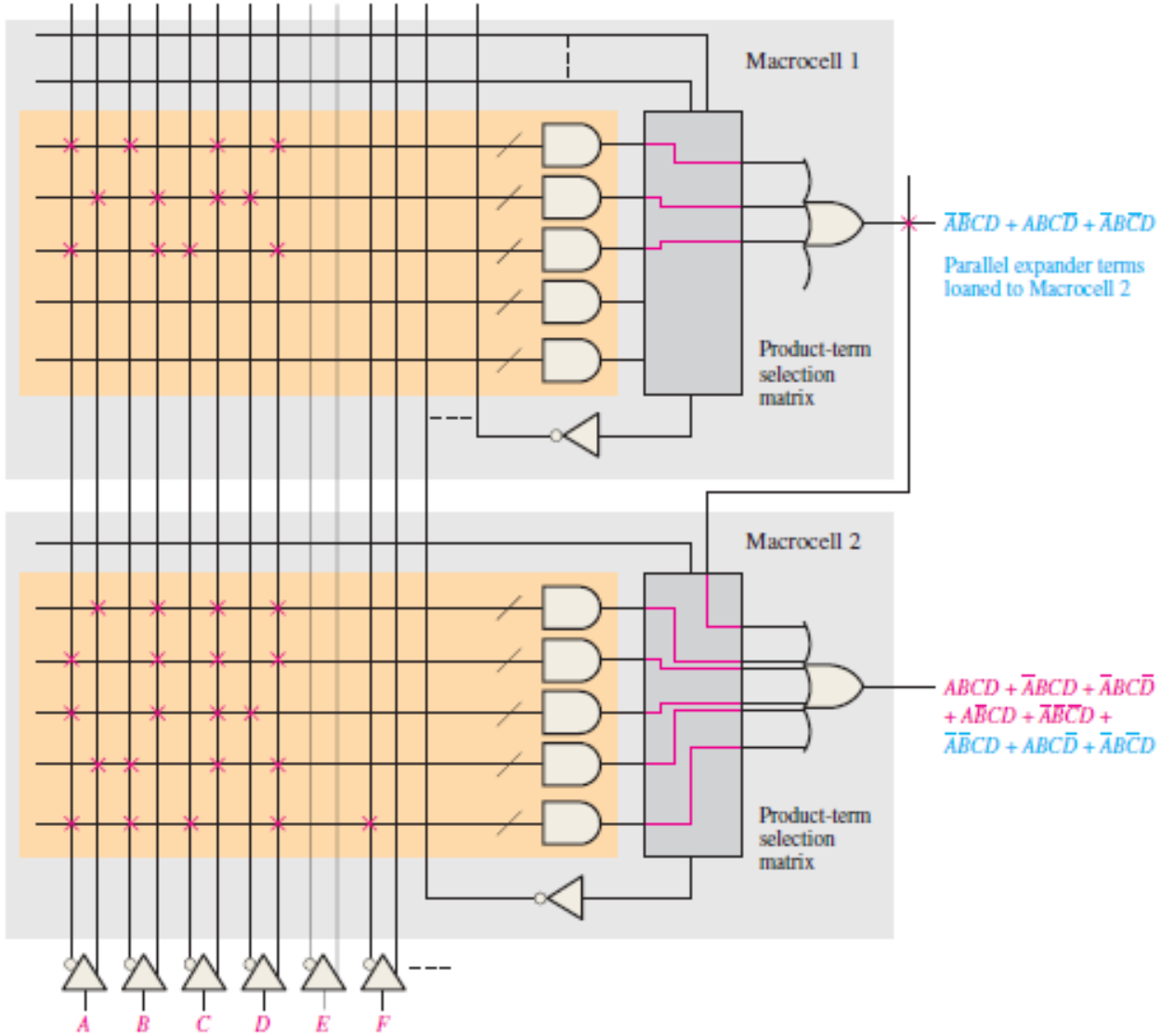
FIGURE. Example of how a shared expander can be used in a microcell to increase the number of product terms

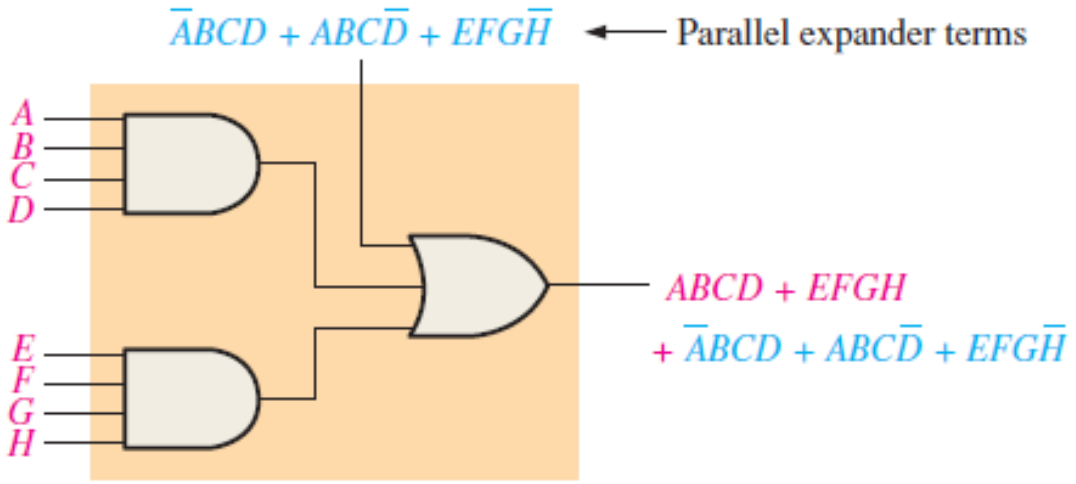
CPLD producing SOP using shared expander

FIGURE. Simplified illustration of using a shared expander term from another macrocell to increase an SOP expression. The red Xs and lines represent the connections produced in the hardware by the software compiler running the programmed design



CPLD producing SOP using parallel expander

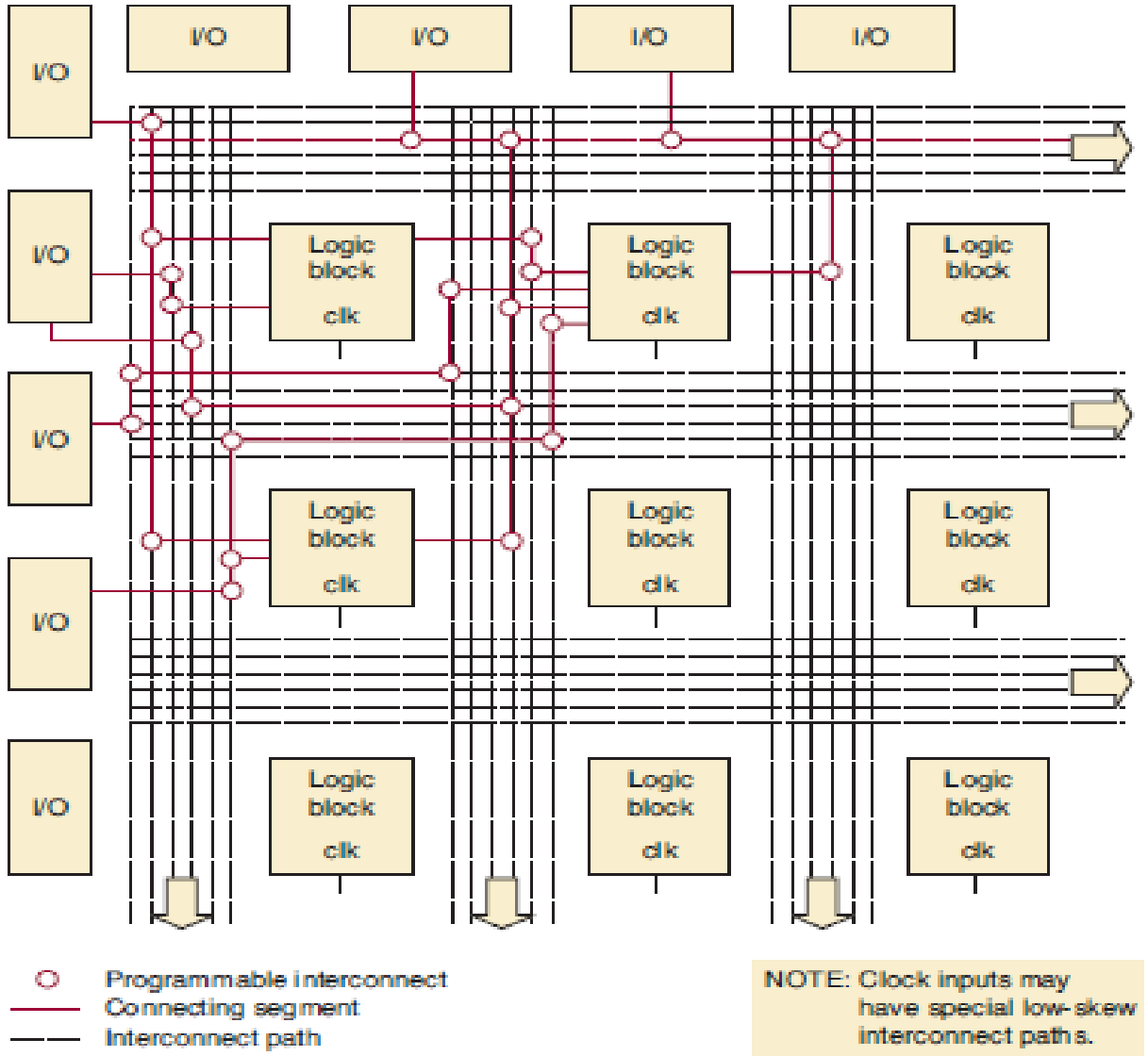




Basic concept of the parallel expander

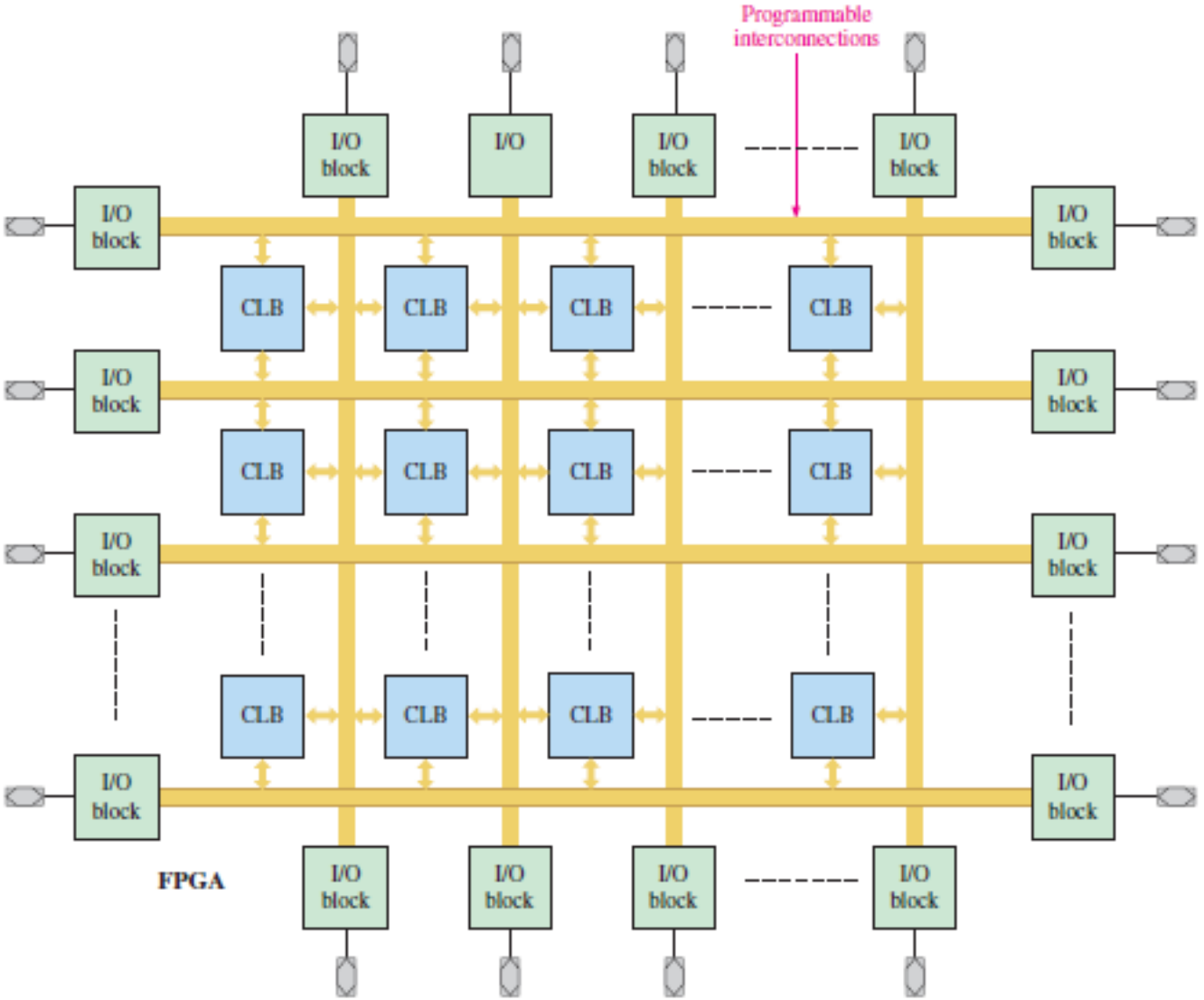
FPGA – Field Programmable Gate Array

- An FPGA (field-programmable gate array) differs in architecture, does not use PAL/PLA type arrays, and has much greater densities than CPLDs.
- A typical FPGA has many times more equivalent gates than a typical CPLD.
- The logic-producing elements in FPGAs are generally much smaller than in CPLDs, and there are many more of them.

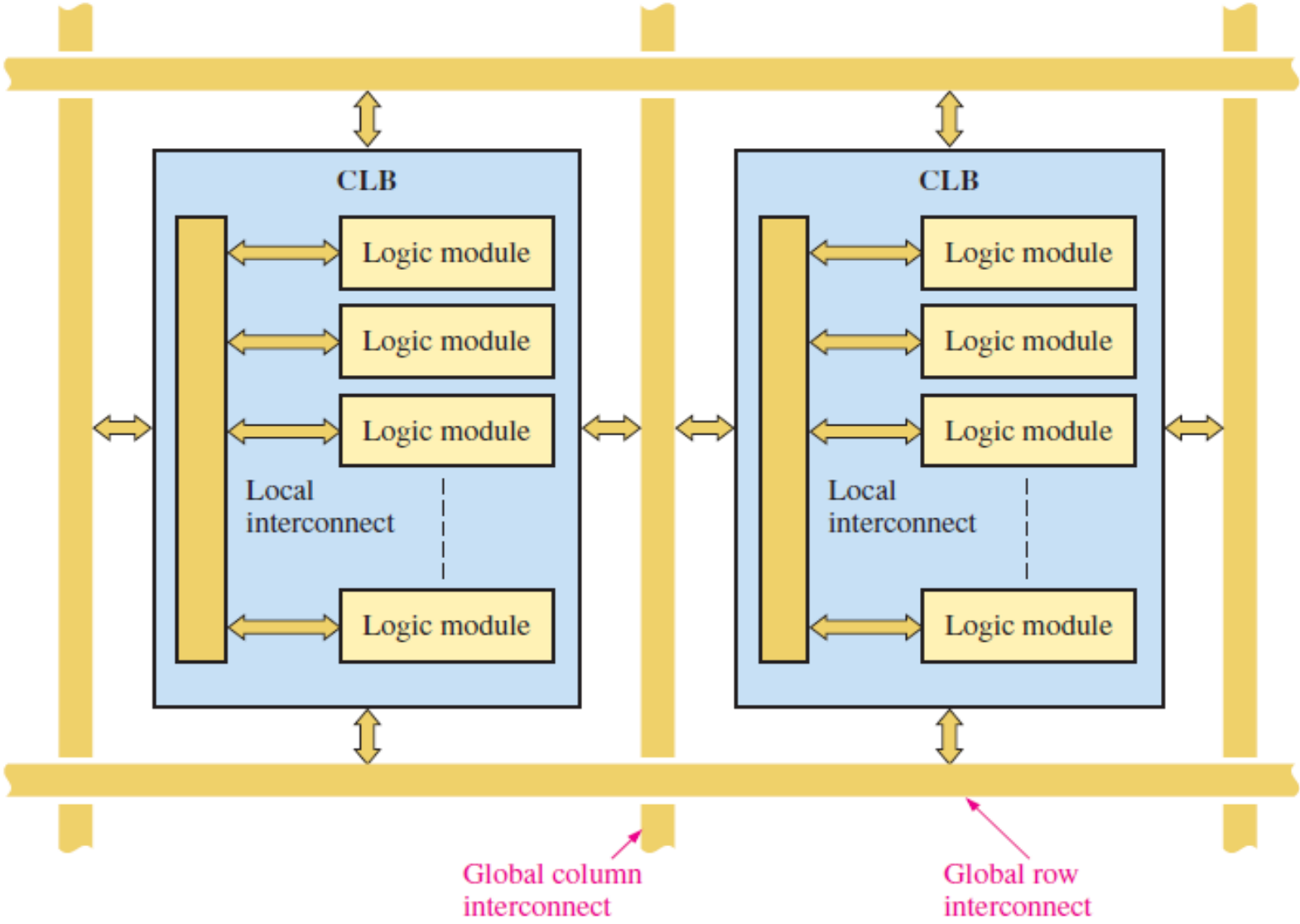


FPGA – Field Programmable Gate Array

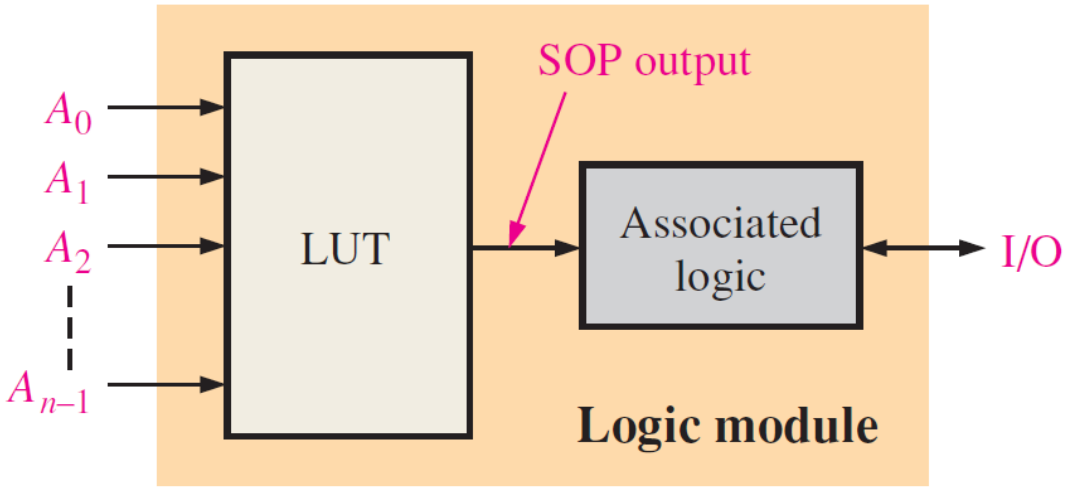
- The three basic elements in an FPGA are
  - ✓ the configurable logic block (CLB),
  - ✓ the interconnections, and
  - ✓ the input/output (I/O) blocks



Basic configurable logic blocks (CLBs)

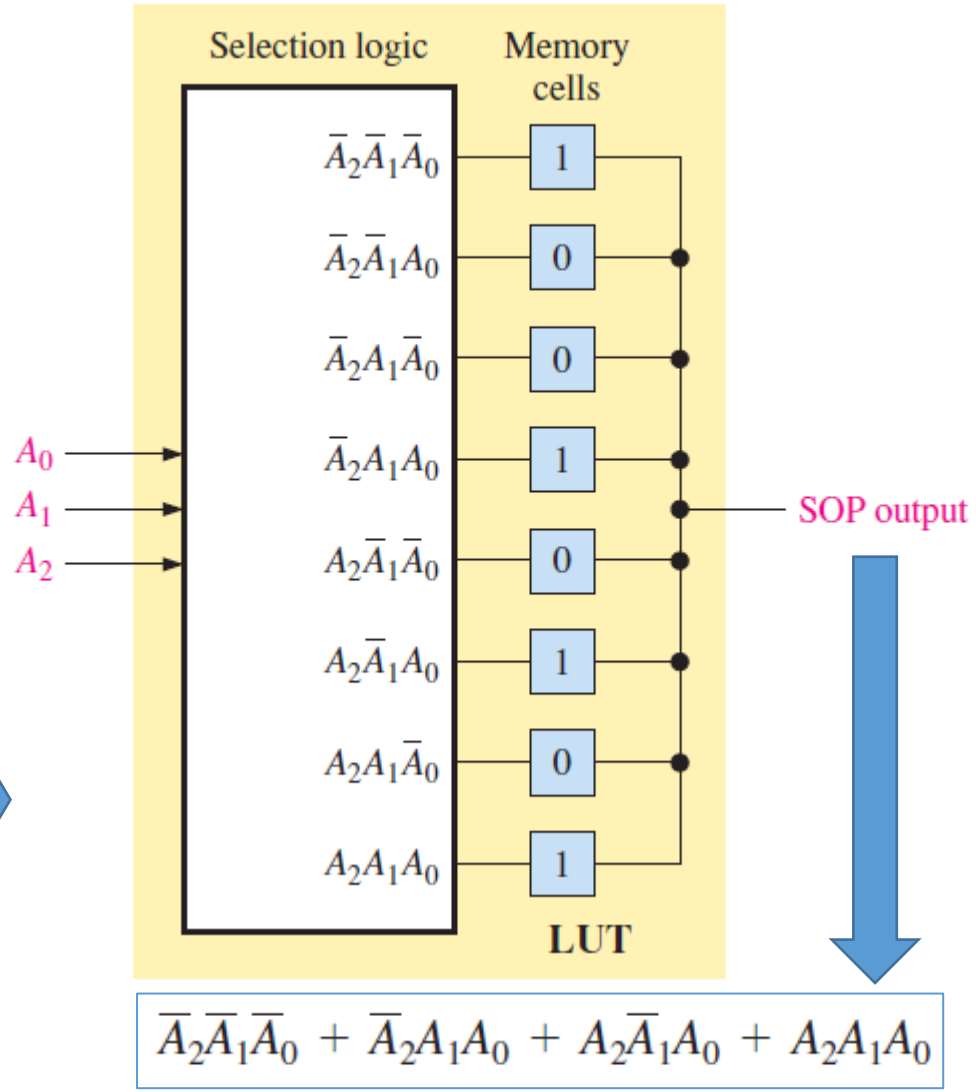


FPGA logic module and look-up table (LUT)



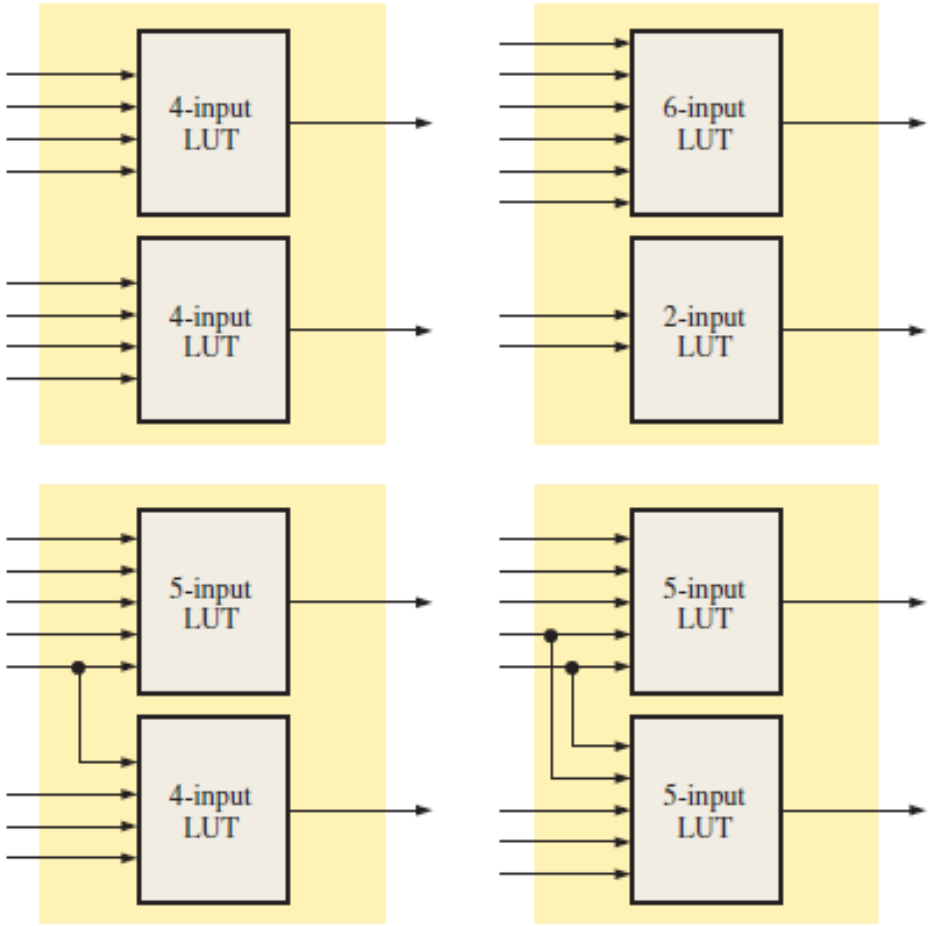
FPGA logic module

number of memory cells equal to 2*n*, where *n* is the number of input variables



$$\bar{A}_2\bar{A}_1\bar{A}_0 + \bar{A}_2A_1A_0 + A_2\bar{A}_1A_0 + A_2A_1A_0$$

Possible look-up table (LUT) configuration in a logic module



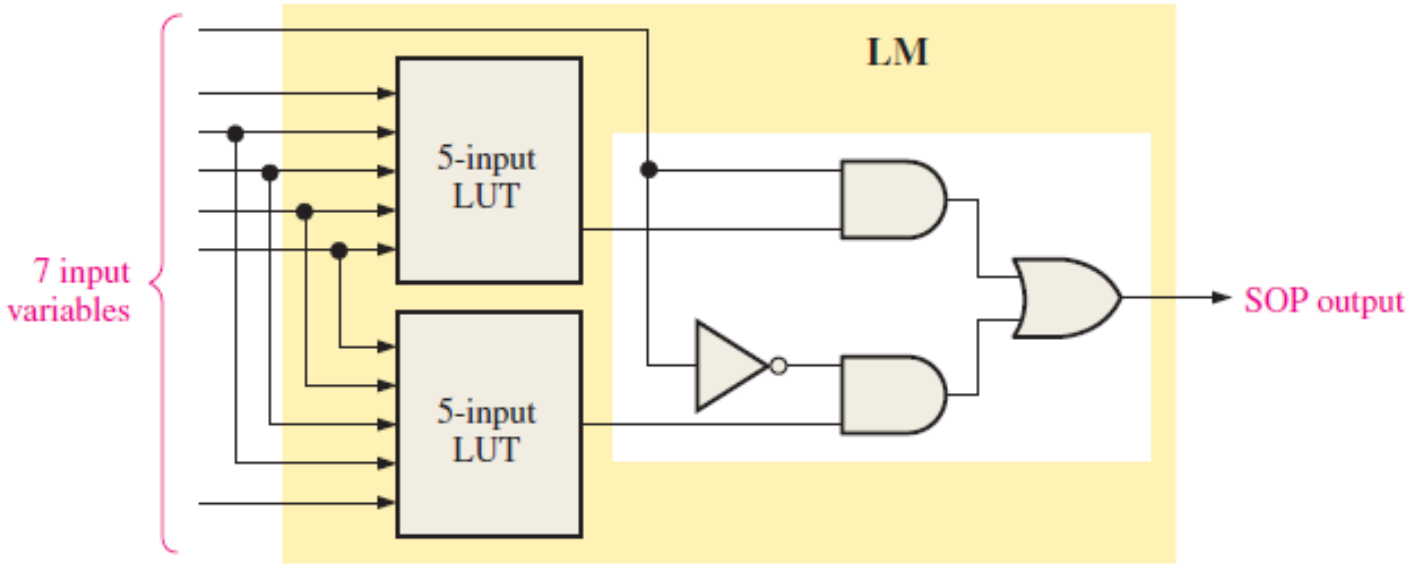
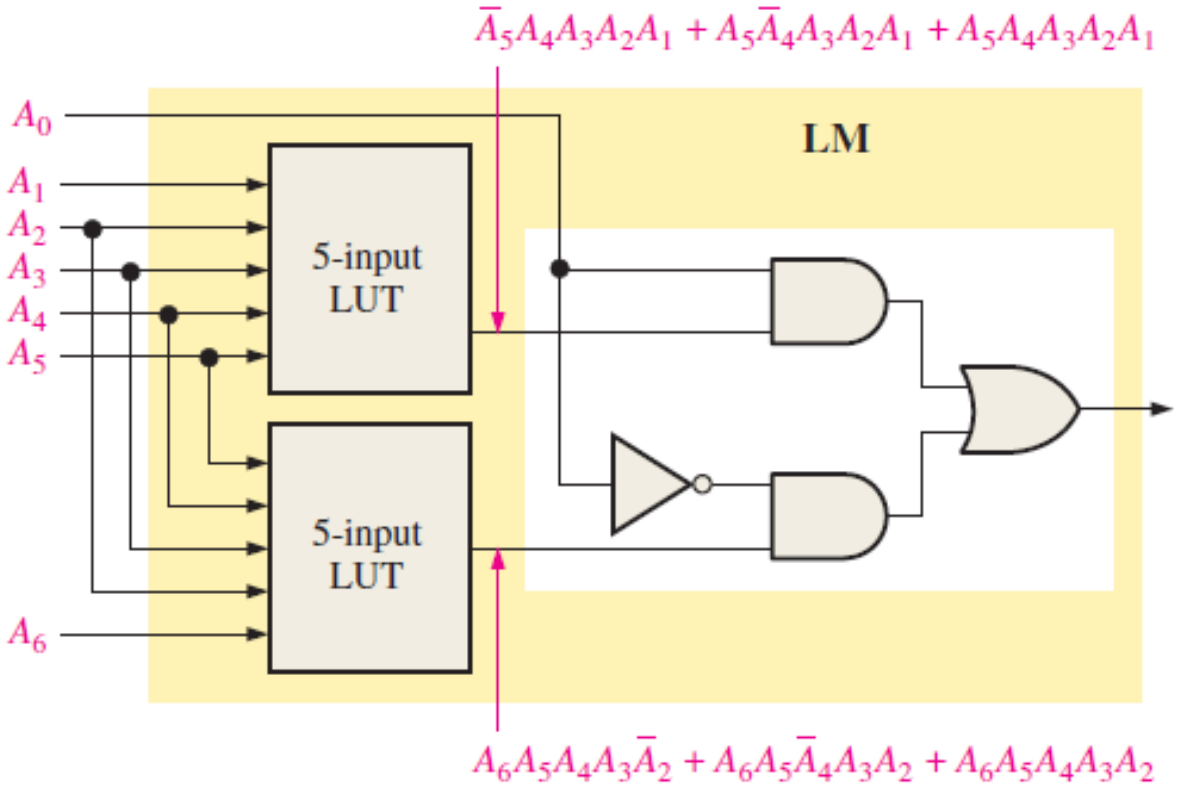


FIGURE. Expansion of a logic module (LM) to produce a 7-variable SOP function in the extended LUT mode

Example

PROBLEM. A logic module is configured in the extended LUT mode, as shown in Figure 10–30. For the specific LUT outputs shown, determine the final SOP output



Solution

The SOP output expression is as follows:

$$\overline{A}_5A_4A_3A_2A_1A_0 + A_5\overline{A}_4A_3A_2A_1A_0 + A_5A_4A_3A_2A_1A_0 + A_6A_5A_4A_3\overline{A}_2\overline{A}_0 + A_6A_5\overline{A}_4A_3A_2\overline{A}_0 + A_6A_5A_4A_3A_2\overline{A}_0$$

References

1. ***Digital Fundamentals*** by Thomas Floyd, Pearson International Edition, 11th Edition, Chapter 9, Page 561-627.

2. ***Digital Systems: Principles and Applications*** by Ronald Tocci, Neal Widmer and Greg Moss, Pearson International Edition, 12th Edition, Chapter 7, Page 941-961.

Next ?

Final Exam