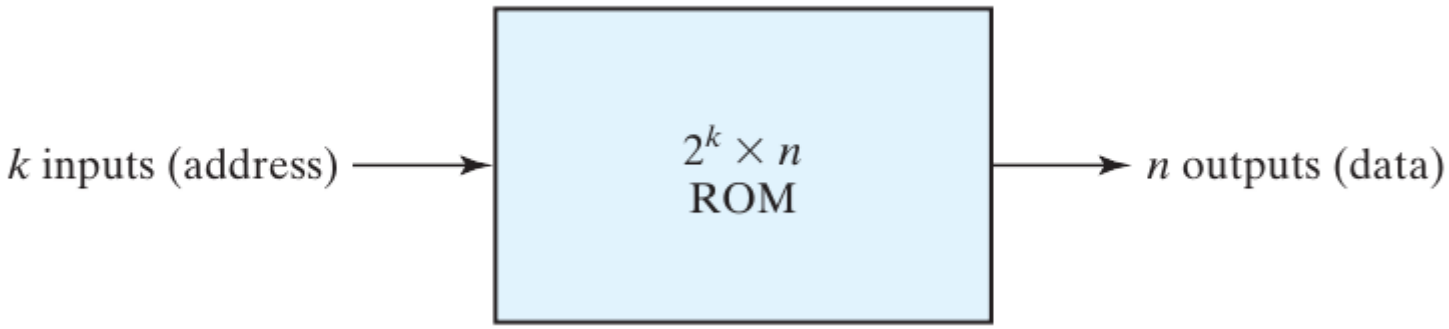
Chap 7 (3)
Read Only Memory

Hanwool Jeong
hwjeong@kw.ac.kr

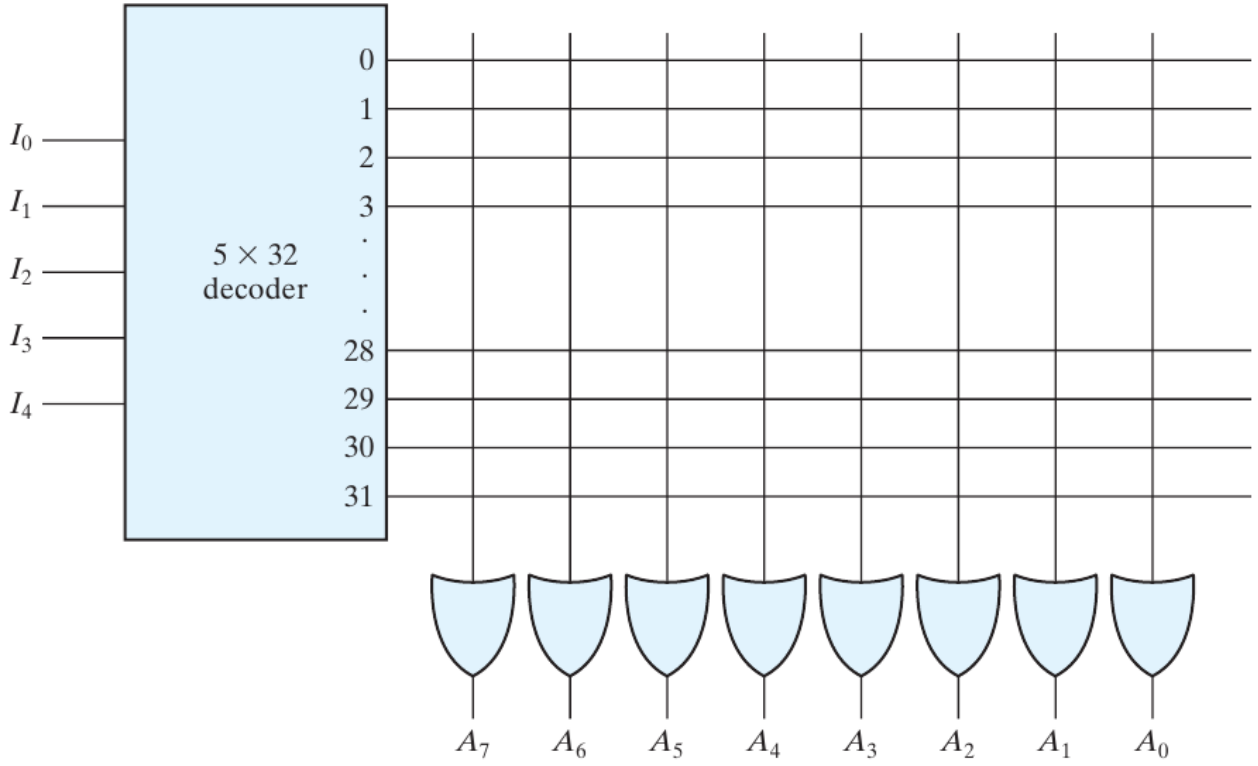
Read Only Memory

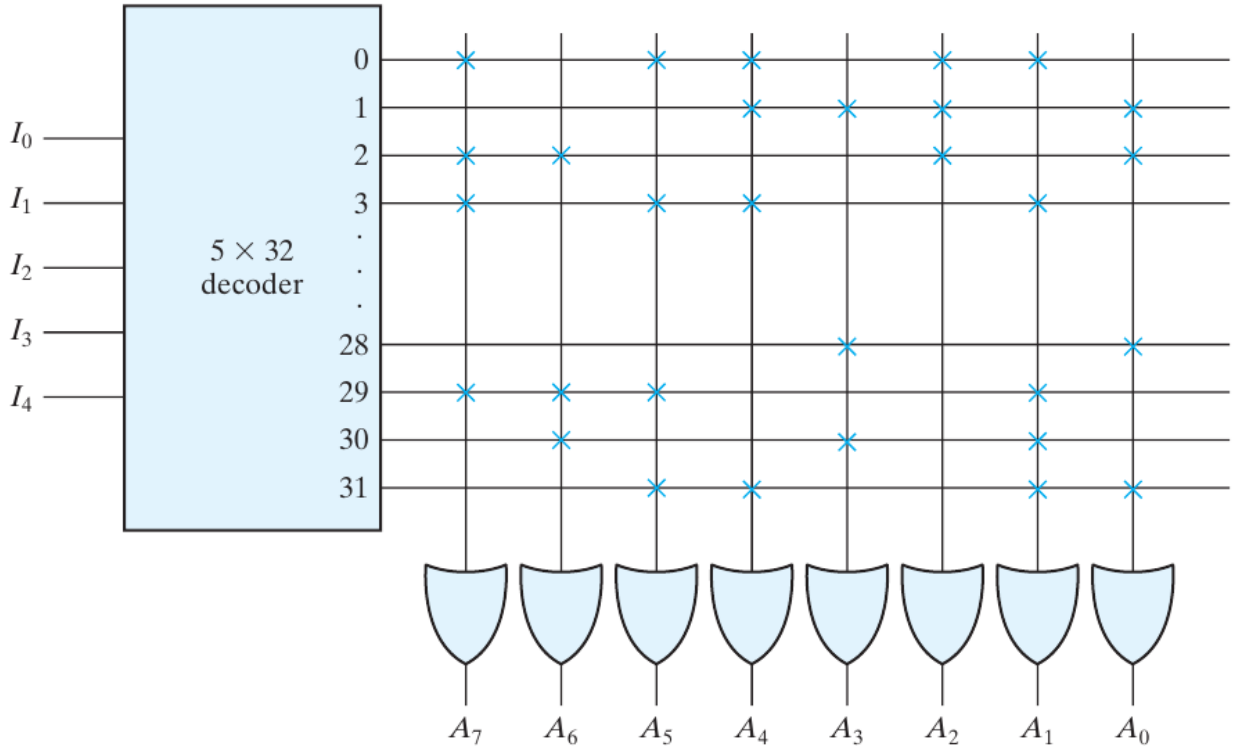
- A read-only memory (ROM) is essentially a memory device in which permanent binary information is stored.
- A block diagram of a ROM consisting of k inputs and n outputs: 2^k words are stored and n bits for each word
- Note that it has no data input. Why?



32x8 ROM

- 256 intersections, called crosspoints, are programmable
- Various physical devices are used to implement crosspoint switches.
- One of the simplest technologies employs a fuse that normally connects the two points, but is opened or “blown” by the application of a high-voltage pulse into the fuse.





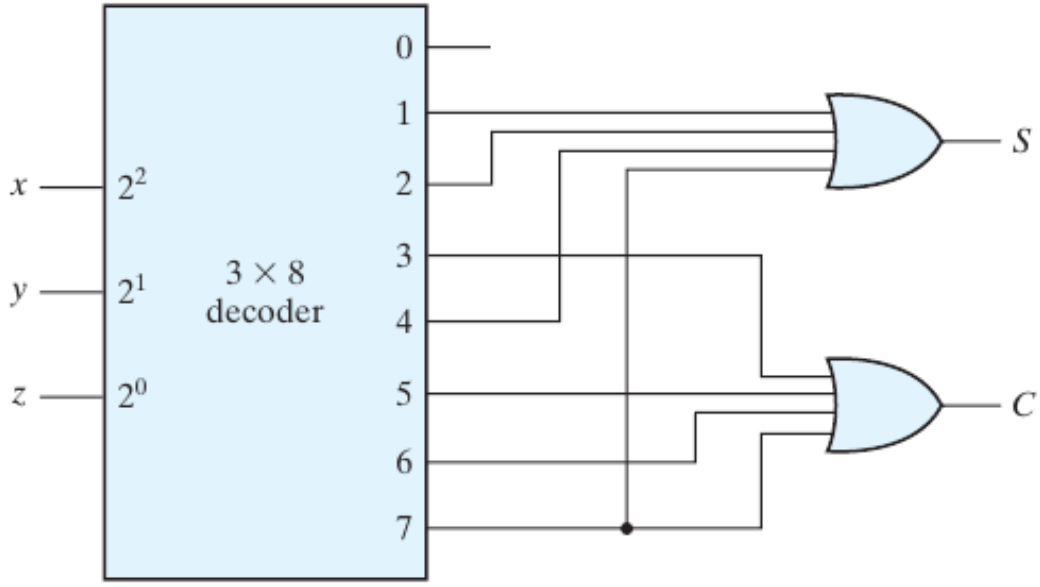
Inputs					Outputs							
I_4	I_3	I_2	I_1	I_0	A_7	A_6	A_5	A_4	A_3	A_2	A_1	A_0
0	0	0	0	0	1	0	1	1	0	1	1	0
0	0	0	0	1	0	0	0	1	1	1	0	1
0	0	0	1	0	1	1	0	0	0	1	0	1
0	0	0	1	1	1	0	1	1	0	0	1	0
		⋮							⋮			
1	1	1	0	0	0	0	0	0	1	0	0	1
1	1	1	0	1	1	1	1	0	0	0	1	0
1	1	1	1	0	0	1	0	0	1	0	1	0
1	1	1	1	1	0	0	1	1	0	0	1	1

Revisit Minterm Generation by Decoder

- With a decoder, by inserting OR gates to sum the minterms of Boolean functions, we were able to generate any desired combinational circuit.
- EX) Implementation of FA with a decoder

$$S(x, y, z) = \Sigma(1, 2, 4, 7)$$

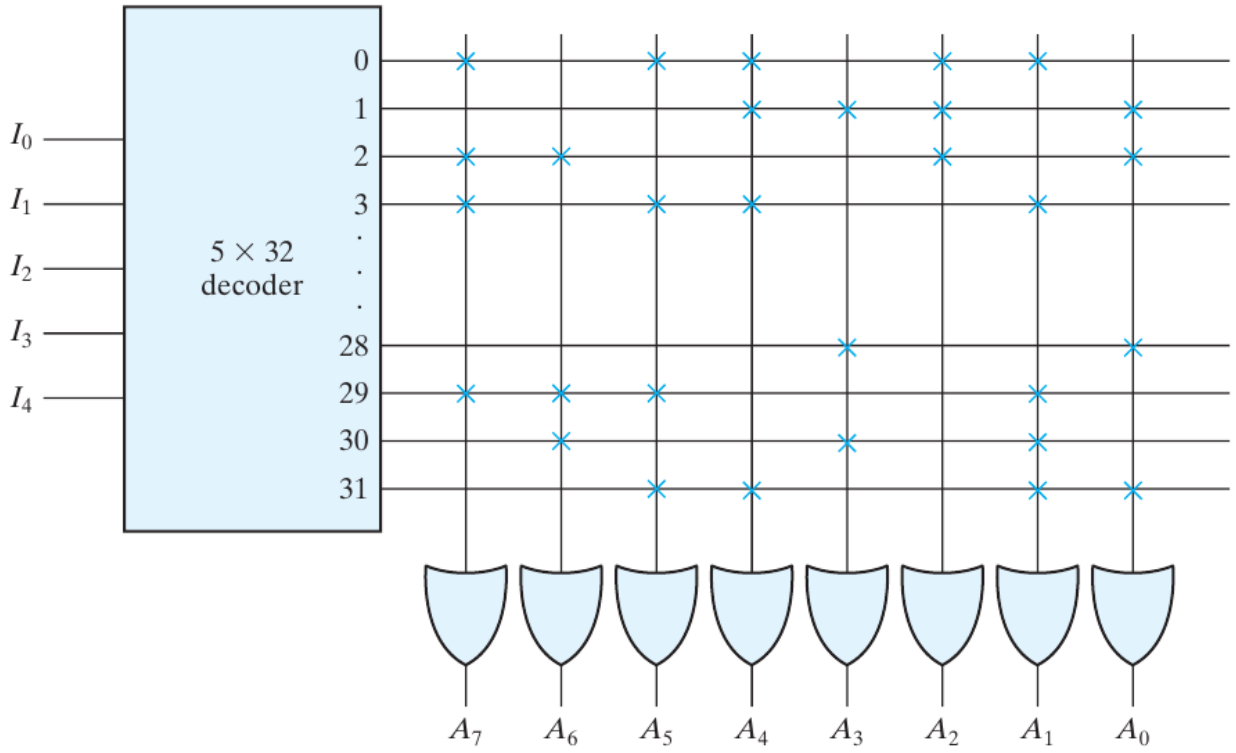
$$C(x, y, z) = \Sigma(3, 5, 6, 7)$$



Interpretation of ROM operation

- ROM can be considered as a combinational circuit implementation.
- For example, A_7 can be expressed in sum of minterms as

$$A_7(I_4, I_3, I_2, I_1, I_0) = \Sigma(0, 2, 3, \ldots, 29)$$



Example 7.1.

- Using a ROM, design the circuit that accepts a three-bit number and outputs a binary number equal to the square of the input number.

Types of ROM

- Mask programming
- Programmable Read only memory (PROM)
  - The fuses in the PROM are blown by the application of a high-voltage pulse to the device through a special pin.
  - The hardware procedure for programming ROMs or PROMs is irreversible, and once programmed, the fixed pattern is permanent and cannot be altered.
- Erasable PROM (EPROM)
  - When EPROM is placed under a special UV light, the shortwave radiation discharges the internal floating gates that serve as the programmed connections.
- Electrically erasable PROM (EEPROM or E^2PROM).
  - This device is like the EPROM, except that the previously programmed connections can be erased with an electrical signal instead of ultraviolet light.
- Flash?

Types of Combinational Programmable Logic Devices (PLD)

- Combinational programmable logic device (PLD)?
  ➔ An integrated circuit with programmable gates divided into AND array and OR array to provide an AND–OR sum-of-product implementation.
- Three major types of combinational PLDs



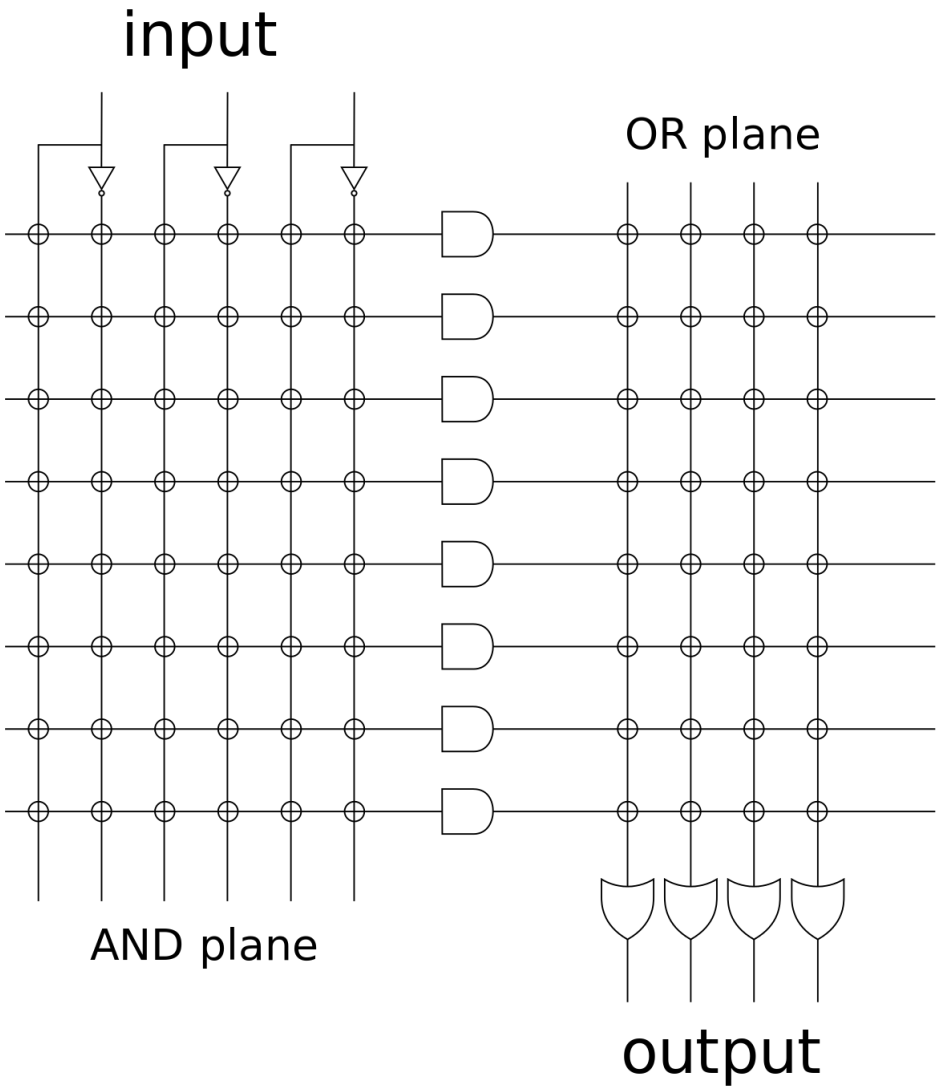
(a) Programmable read-only memory (PROM)



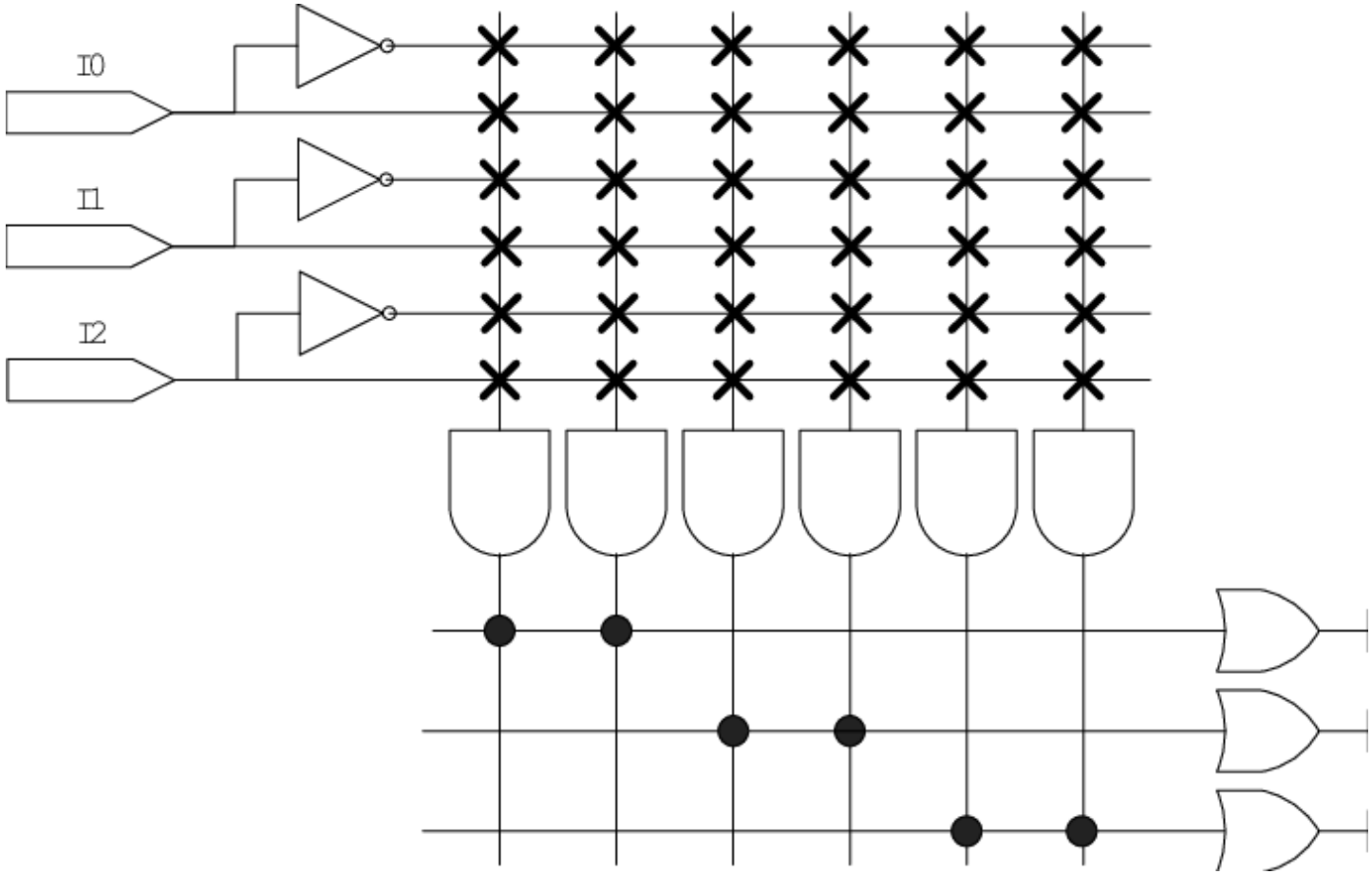
(b) Programmable array logic (PAL)



(c) Programmable logic array (PLA)

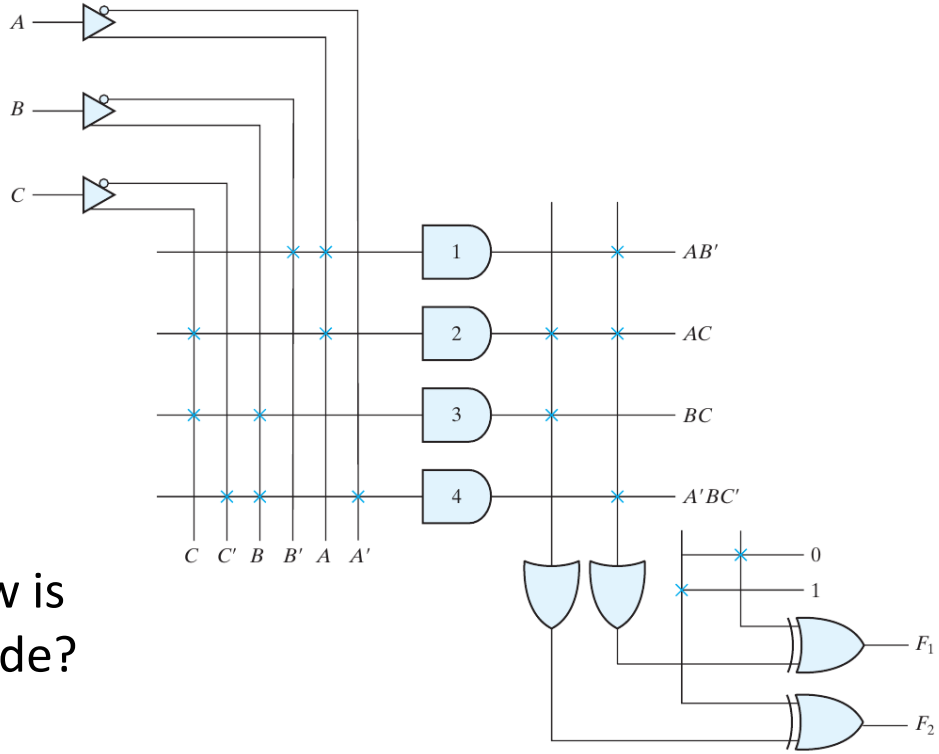

input
OR plane
AND plane
output

PAL



PLA Example

- PLA does not generate all possible minterms as in PROM
  ➔ The decoder is replaced with a programmable AND array
- For **n** inputs, **k** product terms, and **m** outputs, the internal logic of the PLA consists of **n** buffer–inverter gates, **k** AND gates, **m** OR gates, and m XOR gates.



How is open node?

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

Fuse Map of PLA or Programming Table

- A dash specifies a blown fuse in both the input variable and its complement.
- A T (true) output dictates that the other input of the corresponding XOR gate be connected to 0, and a C (complement) specifies a connection to 1.

		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	—

Example 7.2

Implement the following two Boolean functions with a PLA:

$$F_1(A, B, C) = \sum(0, 1, 2, 4)$$

$$F_2(A, B, C) = \sum(0, 5, 6, 7)$$

PAL

- Because only the AND gates are programmable, the PAL is easier to program than, but is not as flexible as, the PLA.
- A typical PAL integrated circuit may have eight inputs, eight outputs, and eight sections, each consisting of an eight-wide AND–OR array. The output terminals are sometimes driven by three-state buffers or inverters.
- In designing with a PAL, the Boolean functions must be simplified to fit into each section. Unlike the situation with a PLA, a product term cannot be shared among two or more OR gates.

Four inputs
Four outputs
Three-wide AND–OR structure

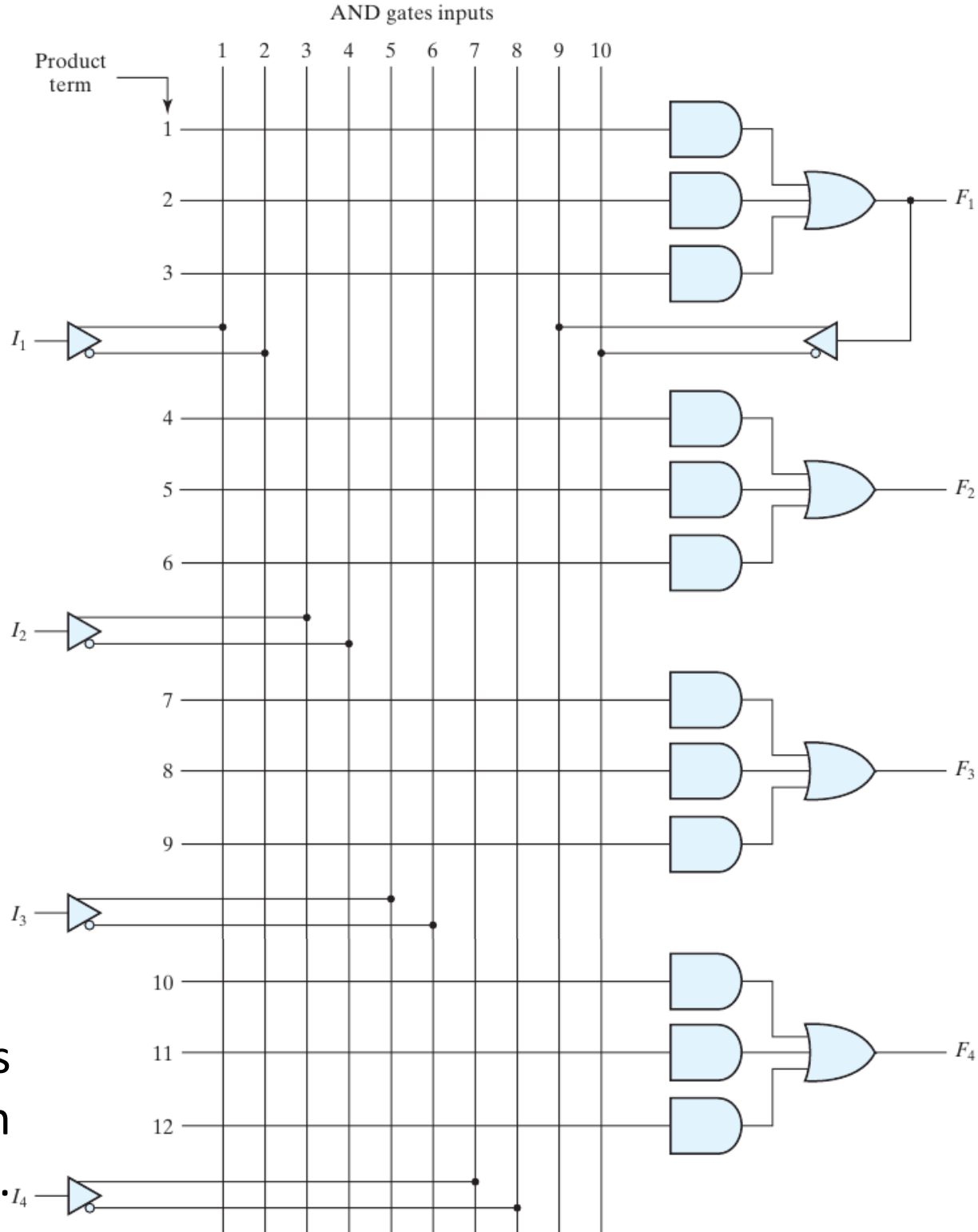
$$w = ABC' + A'B'CD'$$

$$x = A + BCD$$

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

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

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



Commercial PAL devices contain more gates than the one shown.

Design Example Using PAL

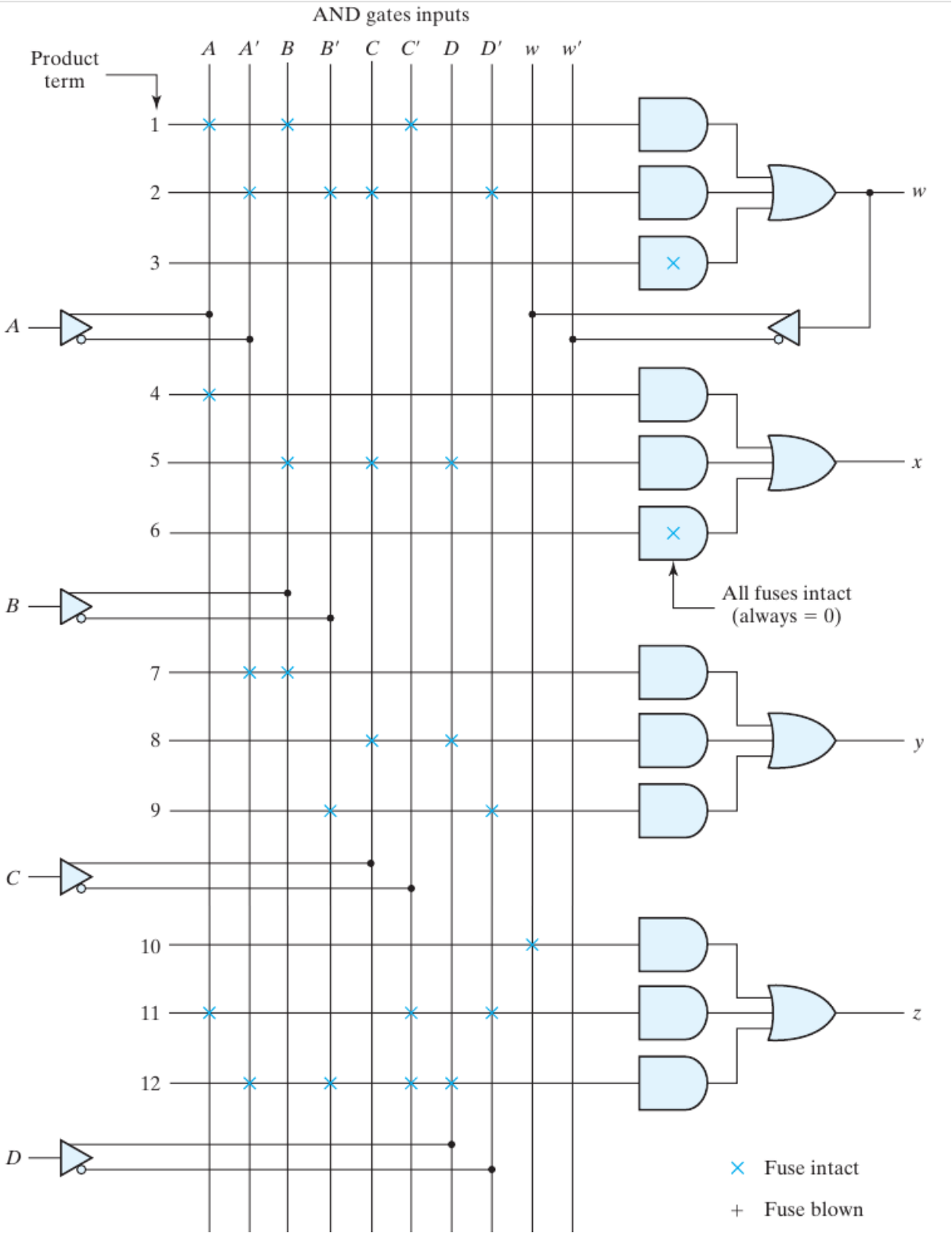
- Given,

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

- We can obtain

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

Product Term	AND Inputs					Outputs
	A	*B*	*C*	*D*	*w*	
1	1	1	0	—	—	$w = ABC' + A'B'CD'$
2	0	0	1	0	—	
3	—	—	—	—	—	
4	1	—	—	—	—	$x = A + BCD$
5	—	1	1	1	—	
6	—	—	—	—	—	
7	0	1	—	—	—	$y = A'B + CD + B'D'$
8	—	—	1	1	—	
9	—	0	—	0	—	
10	—	—	—	—	1	$z = w + AC'D' + A'B'C'D$
11	1	—	0	0	—	
12	0	0	0	1	—	


AND gates inputs
Product term
A
A′
B
B′
C
C′
D
D′
w
w′
1
2
3
4
5
6
7
8
9
10
11
12
w
x
y
z
All fuses intact (always = 0)
Fuse intact
Fuse blown