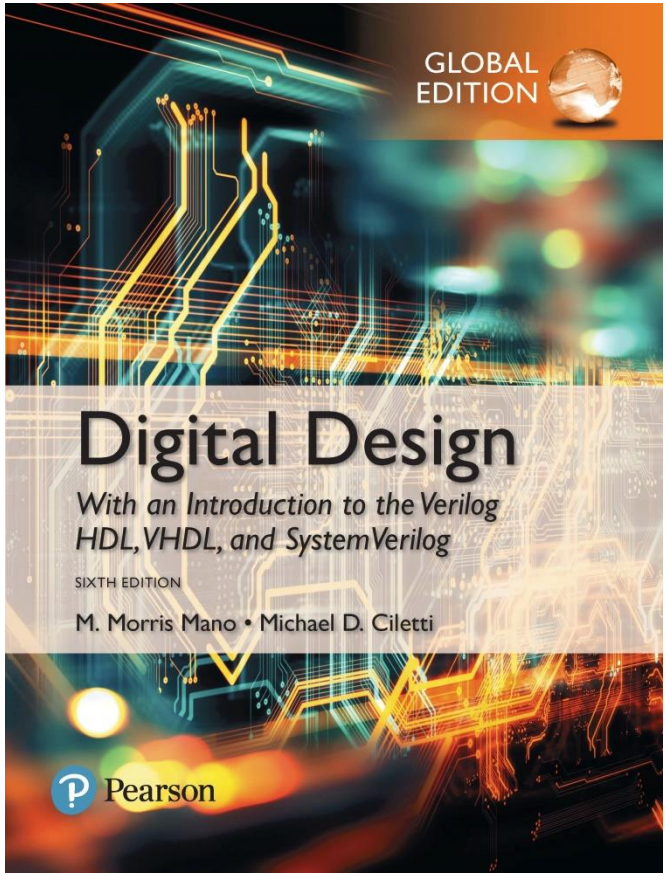
Digital Design

With an Introduction to the Verilog HDL, VHDL, and SystemVerilog

6th Edition, Global Edition



Chapter 07

Memory and Programmable Logic

Pearson

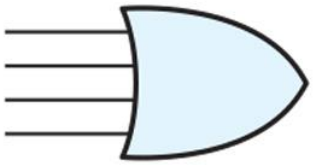
Introduction

Memory

- **RAM**(random access memory): read and write operation
- **ROM**(read only memory) : only read operation

Volatile?

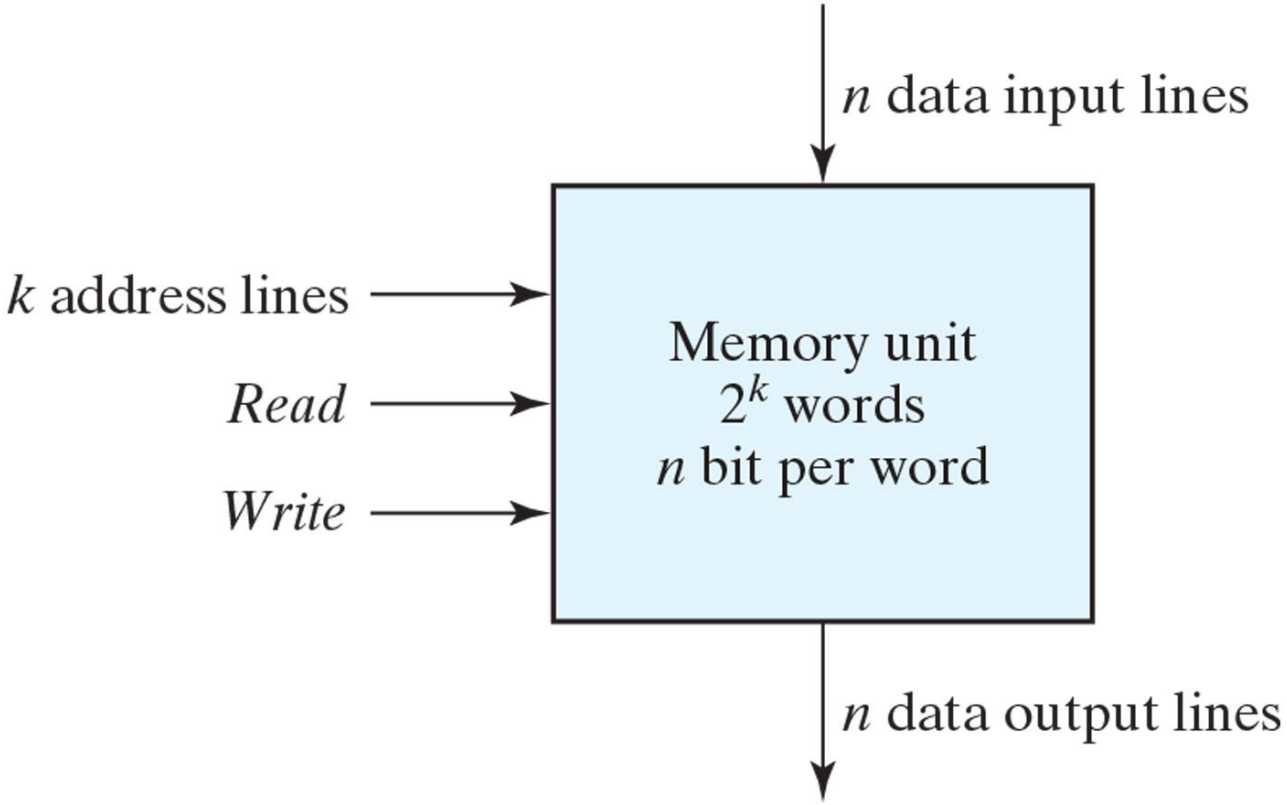
Conventional and array logic diagrams for OR gate.



(a) Conventional symbol

(b) Array logic symbol

Block diagram of a memory unit.



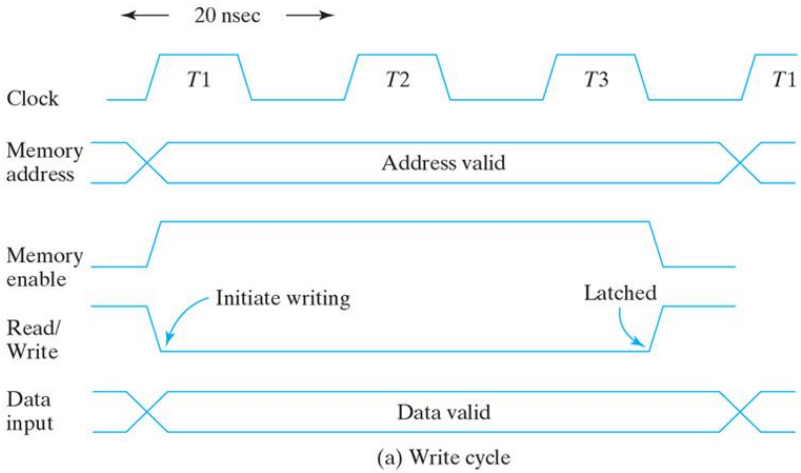
Contents of a 1024 × 16 memory.

Memory address		Memory content
Binary	Decimal	
0000000000	0	1011010101011101
0000000001	1	1010101110001001
0000000010	2	0000110101000110
⋮	⋮	⋮
1111111101	1021	1001110100010100
1111111110	1022	0000110100011110
1111111111	1023	1101111000100101

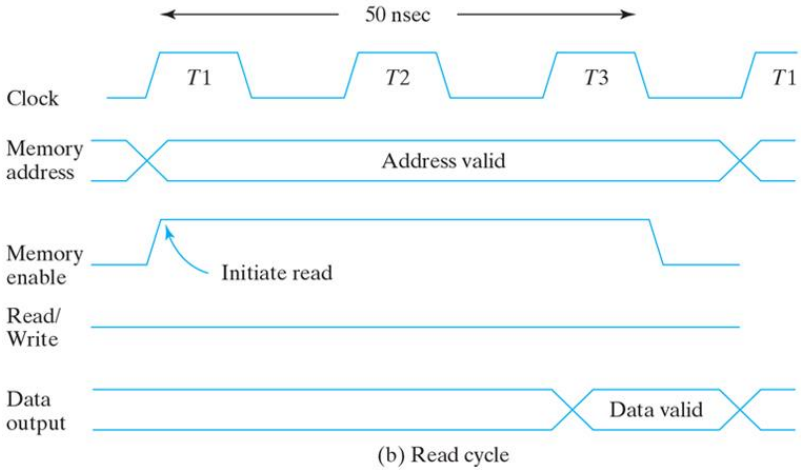
Control Inputs to Memory Chip.

Memory Enable	Read/Write	Memory Operation
0	X	None
1	0	Write to selected word
1	1	Read from selected word

Memory cycle timing waveforms.

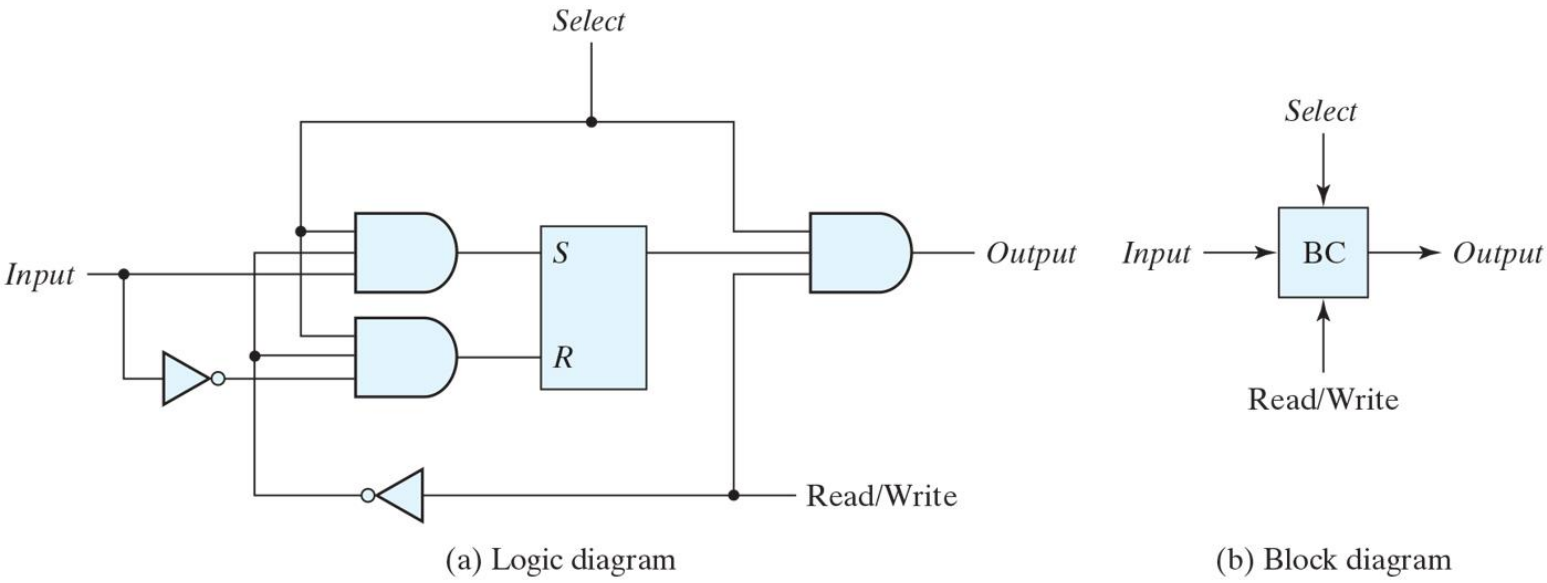


(a) Write cycle



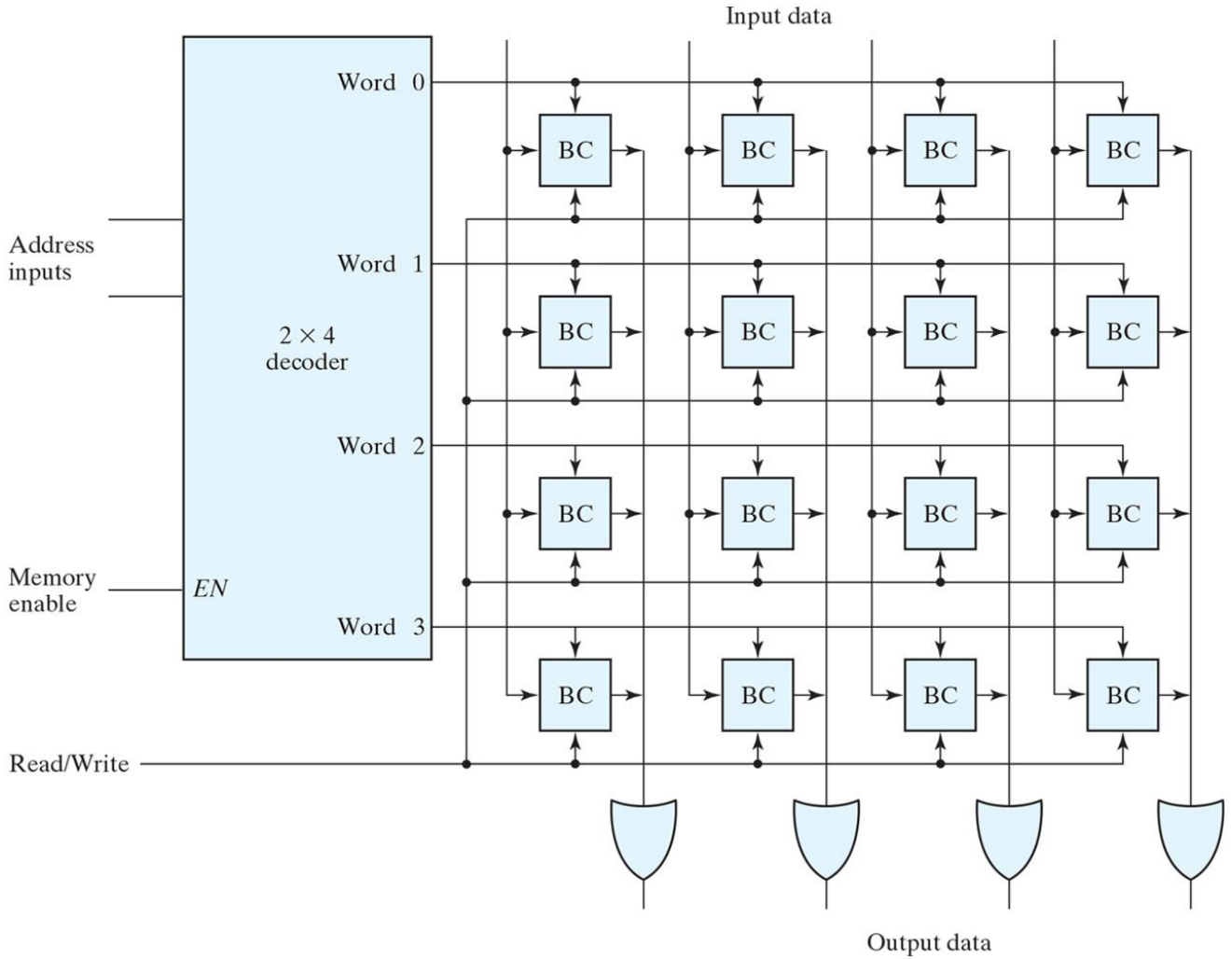
(b) Read cycle

Memory cell.

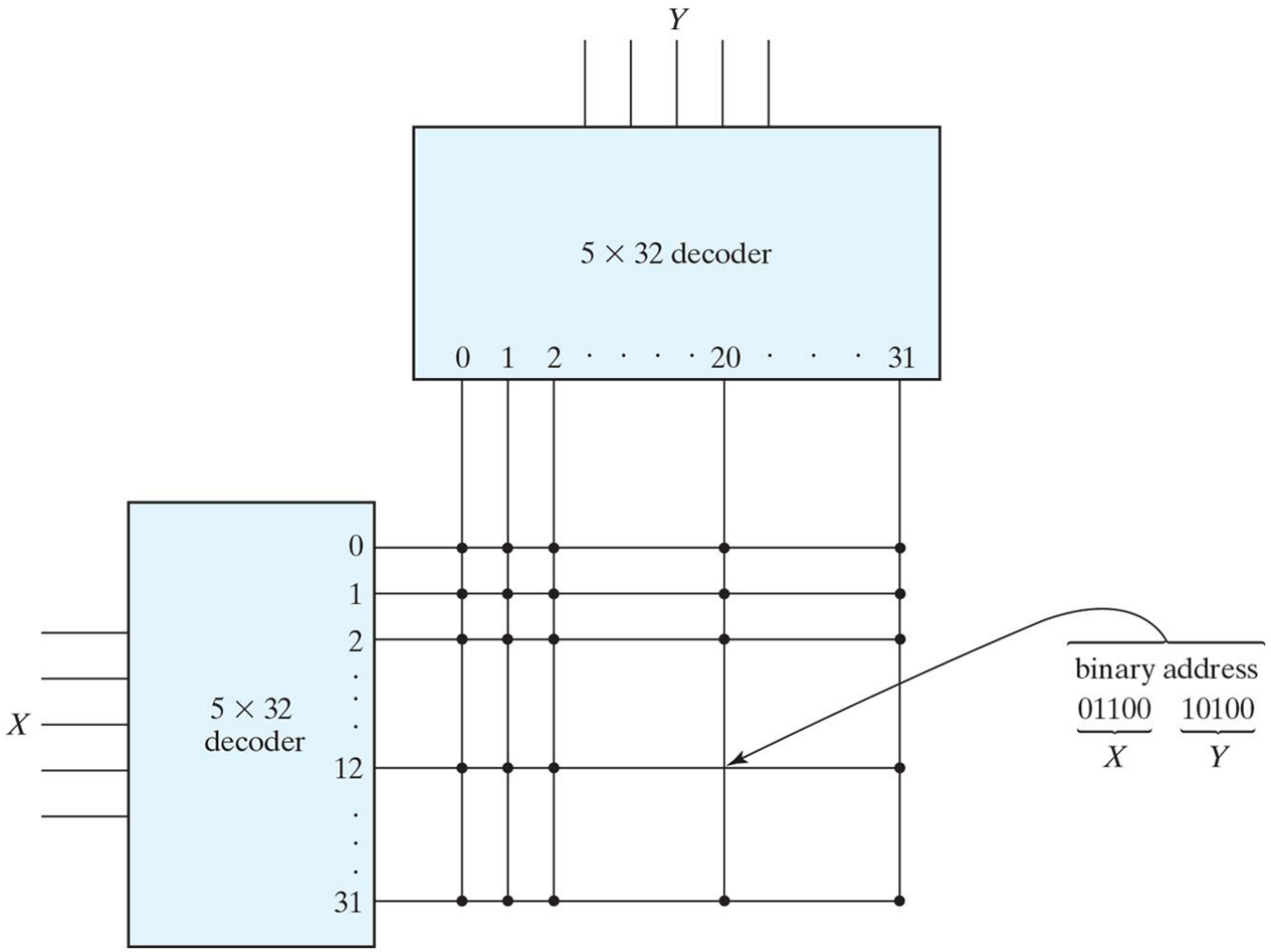


(a) Logic diagram (b) Block diagram

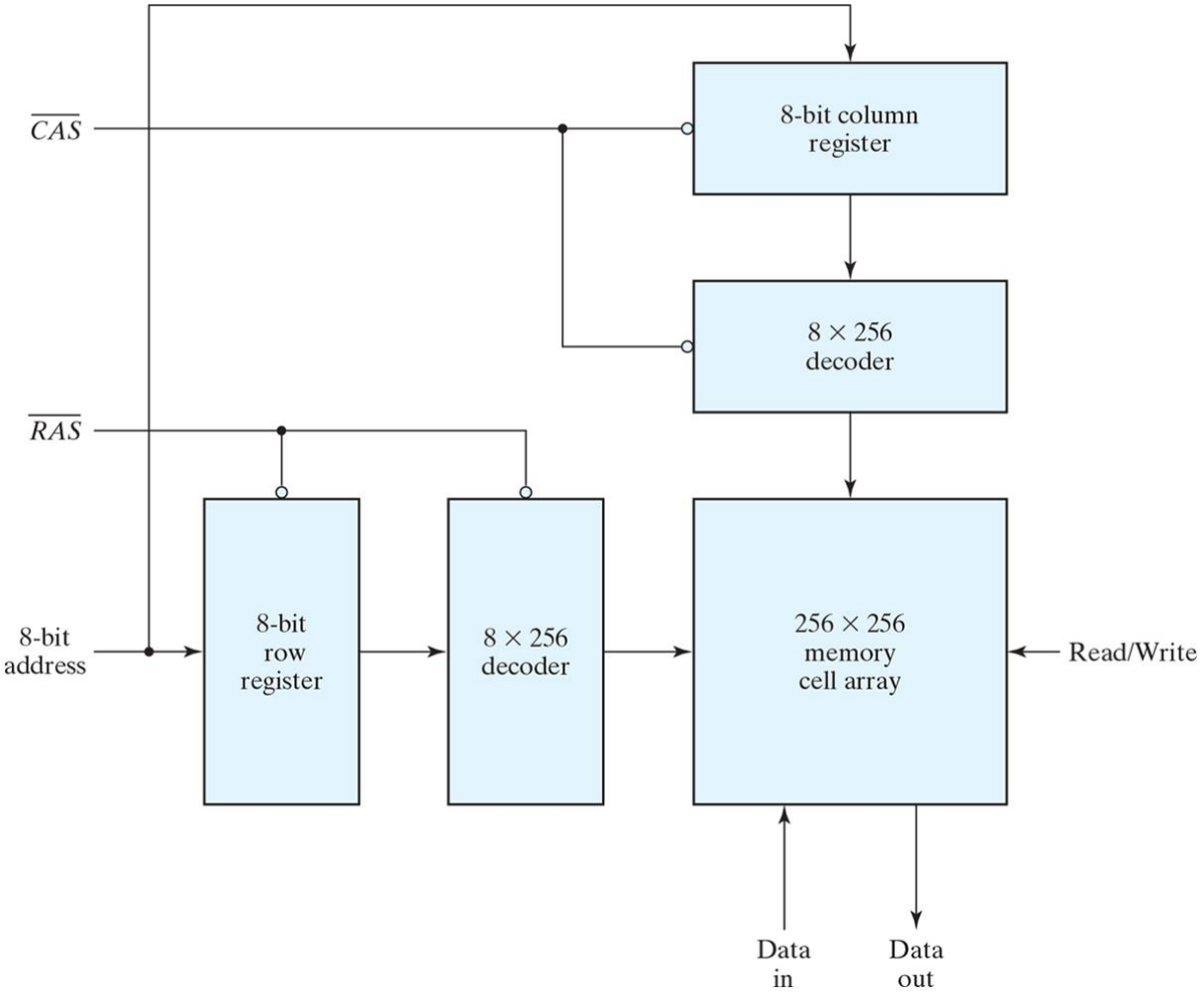
Diagram of a 4 × 4 RAM.



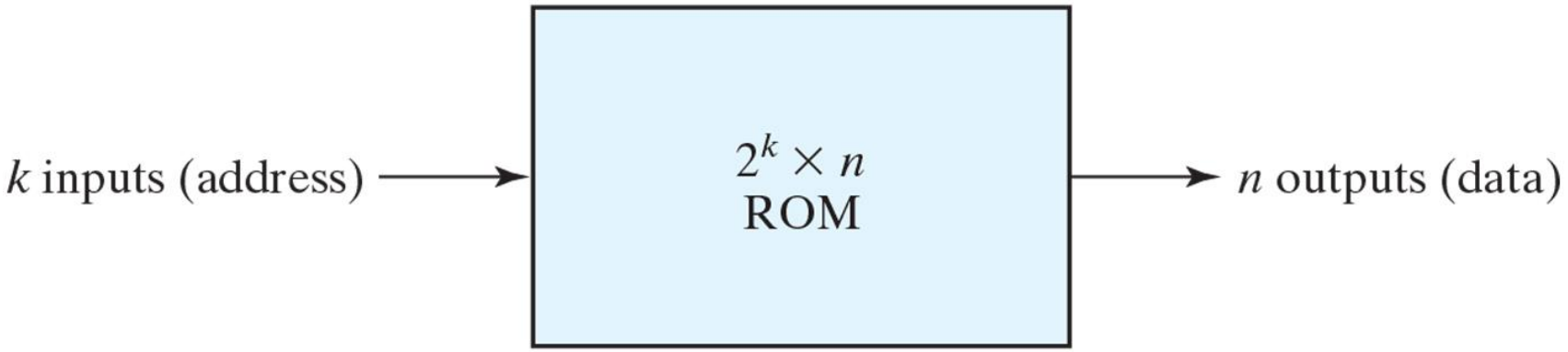
Two-dimensional decoding structure for a 1K-word memory.



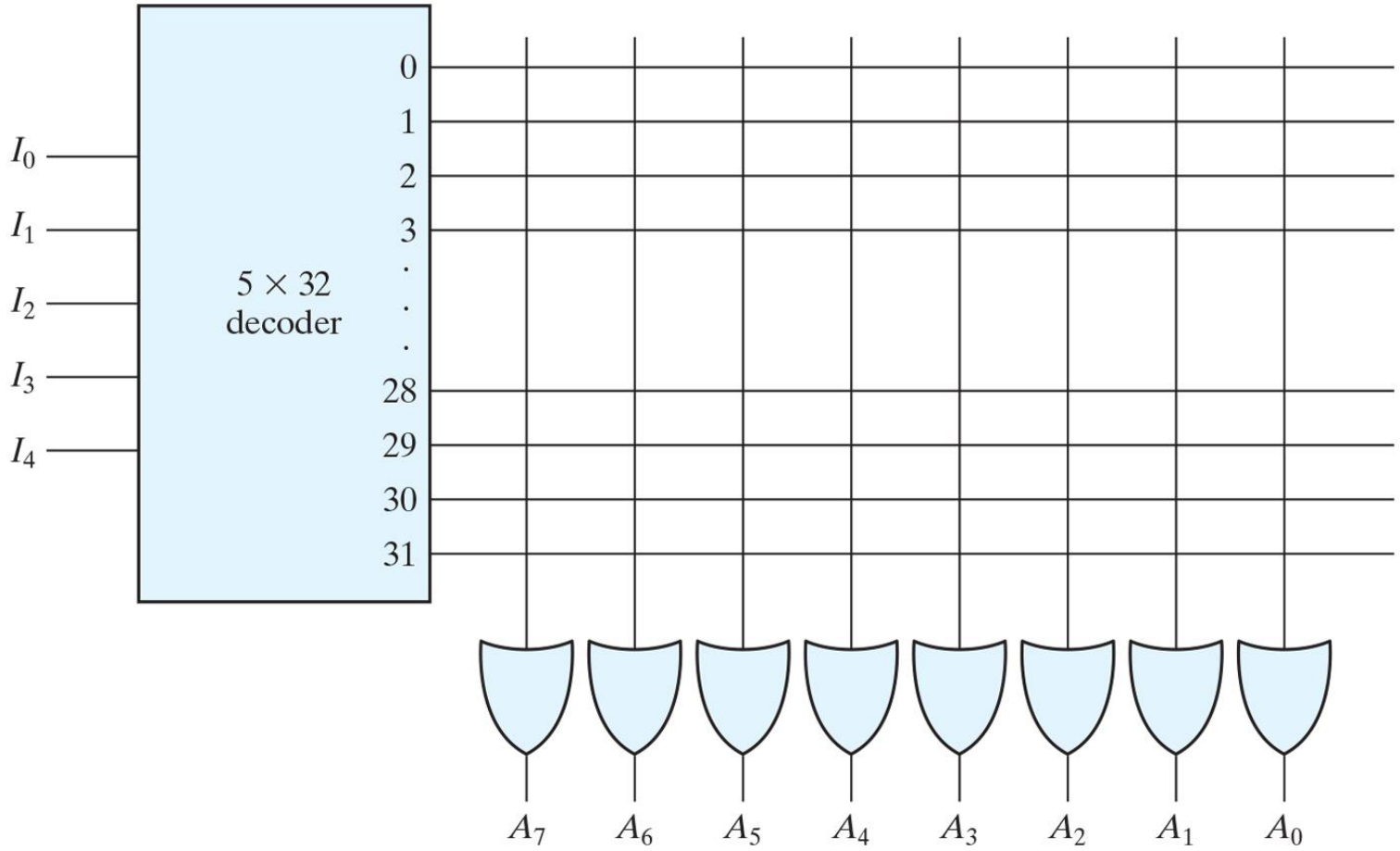
Address multiplexing for a 64K DRAM.



ROM block diagram.



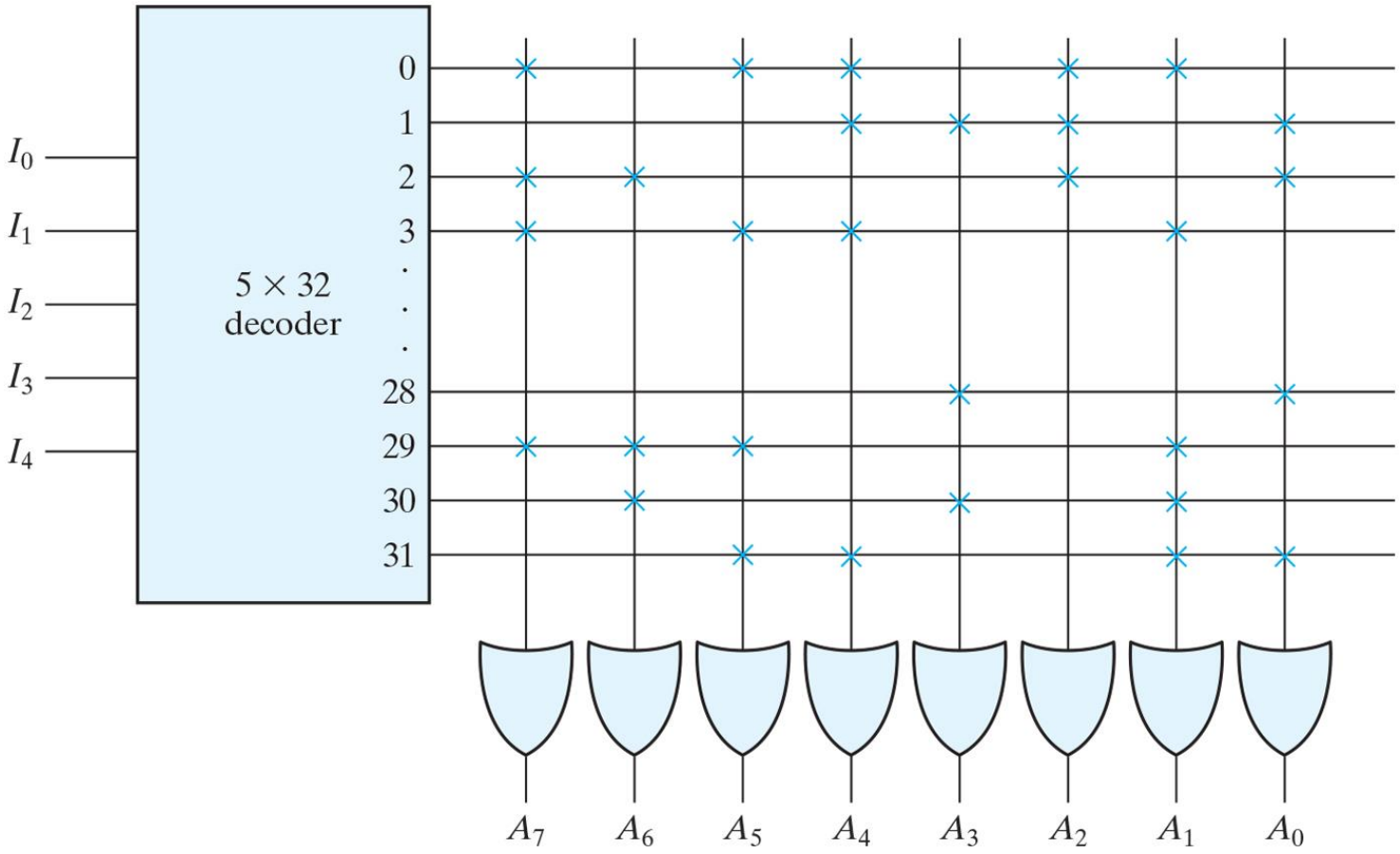
Internal logic of a 32 × 8 ROM.



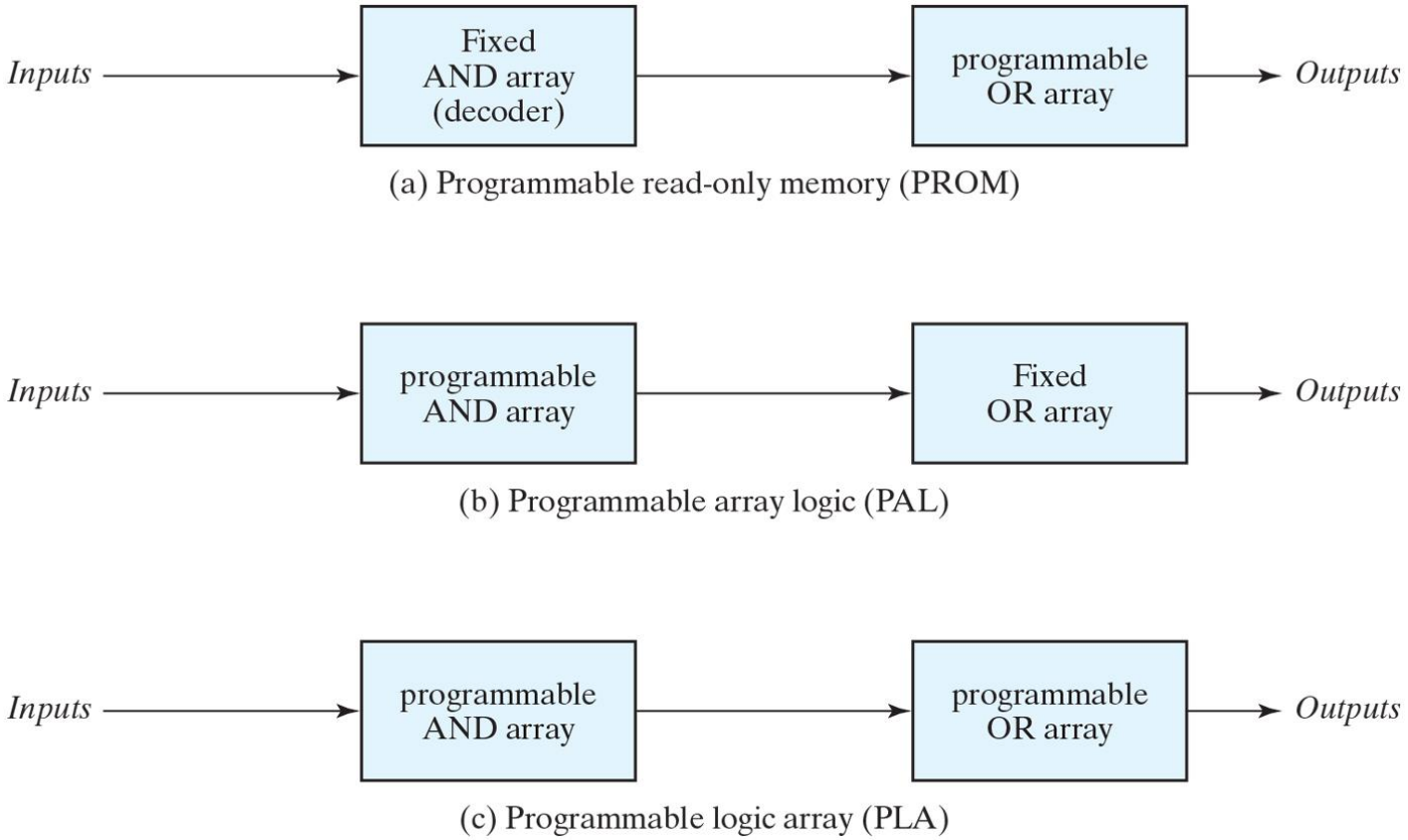
ROM Truth Table (Partial).

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

Programming the ROM according to the previous Table

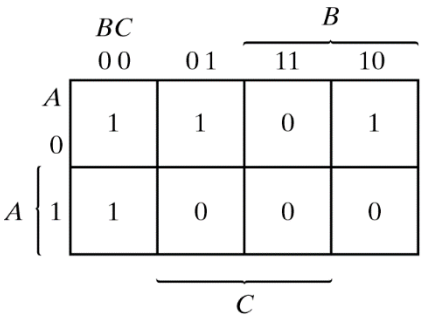


Basic configuration of three PLDs.



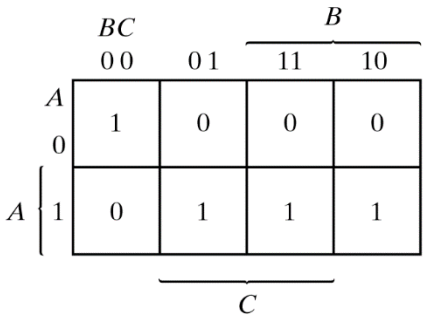
PLA Programming

$$F_1(A,B,C) = \sum(0,1,2,4)$$
$$F_2(A,B,C) = \sum(0,5,6,7)$$
$$F_1 = (AB + AC + BC)'$$
$$F_2 = AB + AC + A'B'C'$$



$F_1 = A'B' + A'C' + B'C'$

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



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

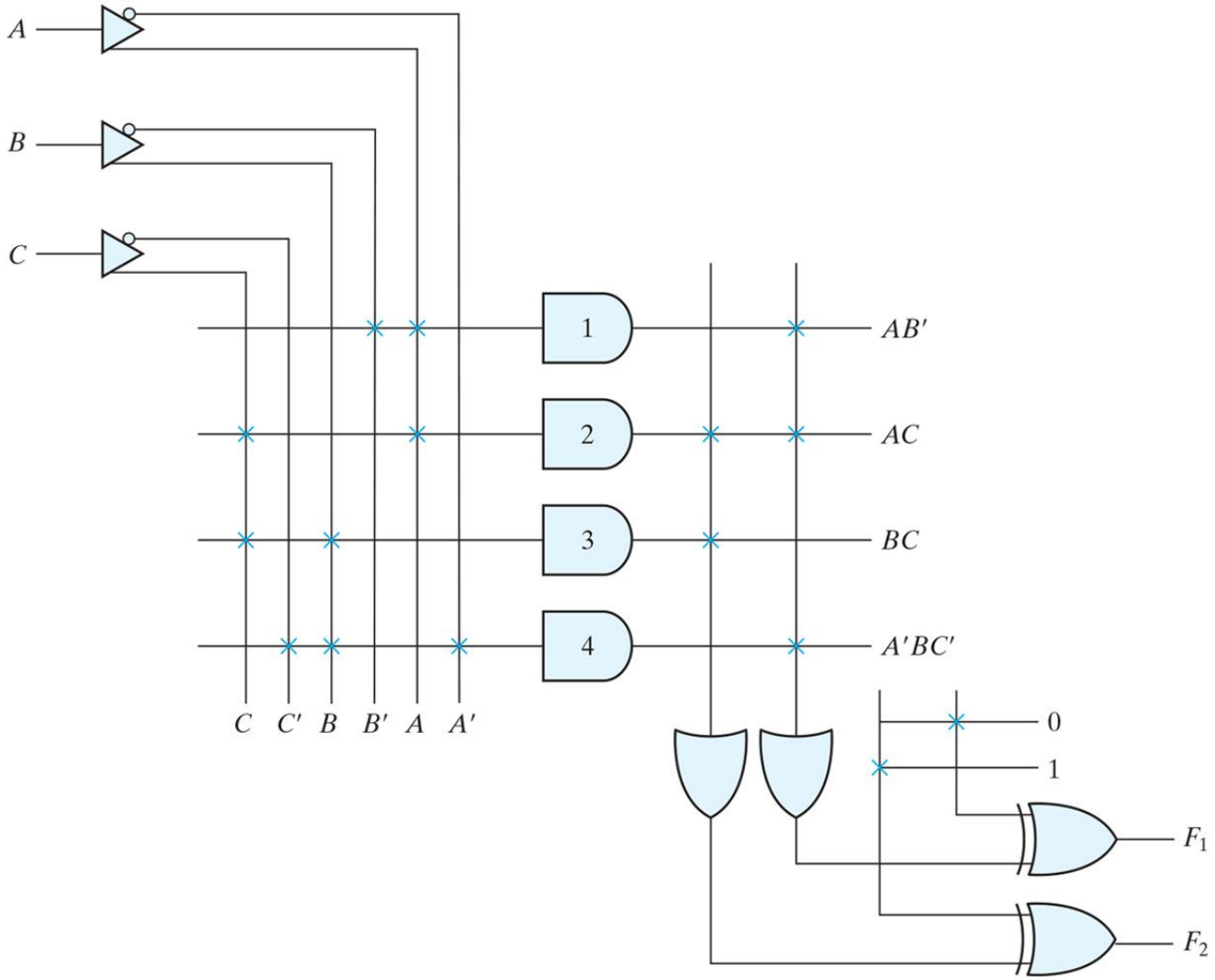
$F_2 = (A'C + A'B + AB'C')'$

PLA programming table

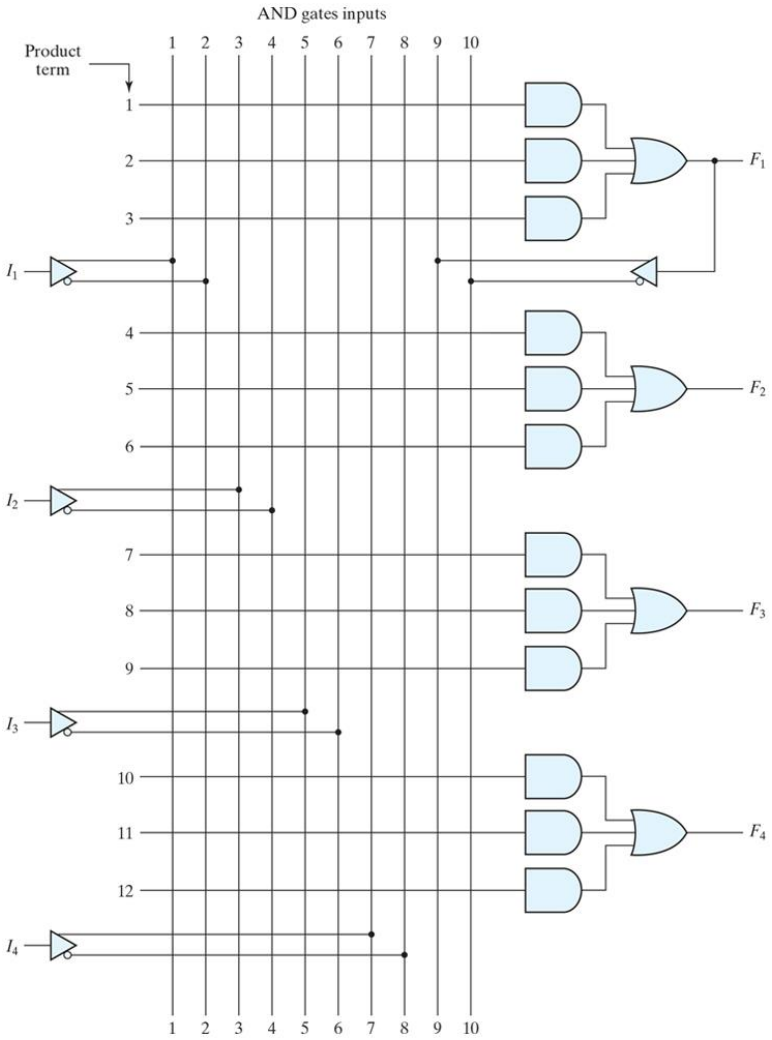
	Product term	Inputs A	B	C	Outputs (C) F_1	(T) F_2
AB	1	1	1	–	1	1
AC	2	1	–	1	1	1
BC	3	–	1	1	1	–
$A'B'C'$	4	0	0	0	–	1

Fig. 7-15 Solution to Example 7-2

PLA with three inputs, four product terms, and two outputs.



PAL with four inputs, four outputs, and a three-wide AND–OR structure.



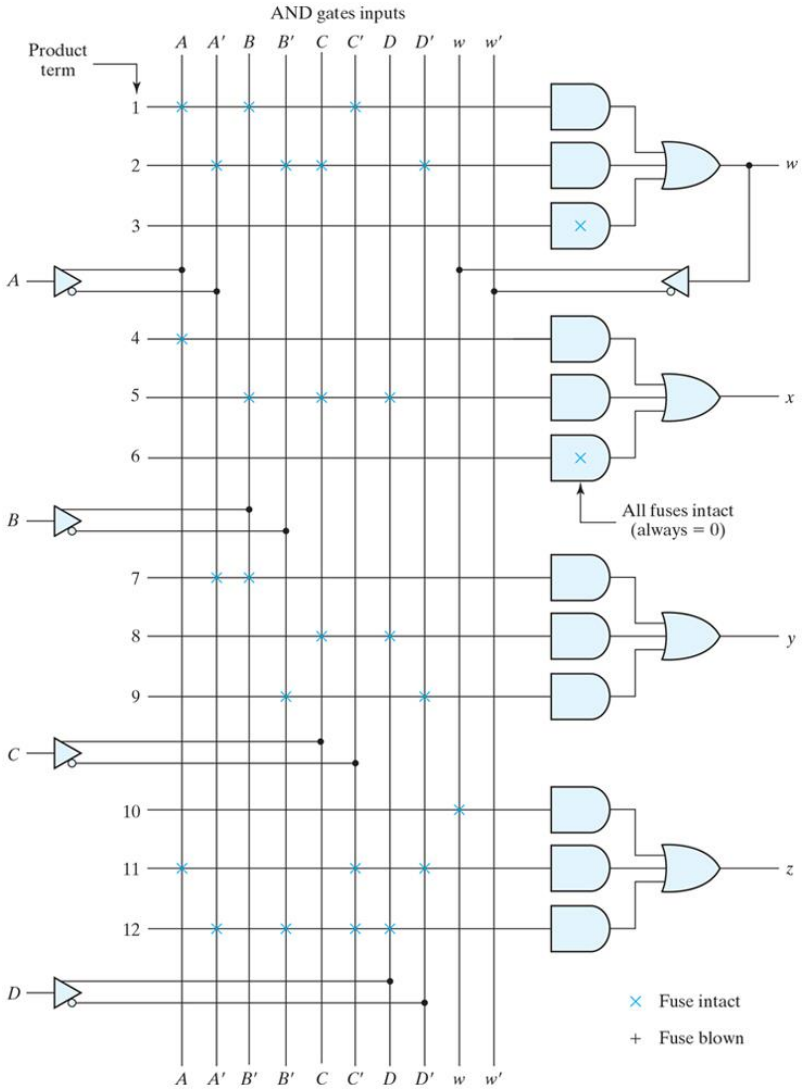
PAL Programming Table.

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

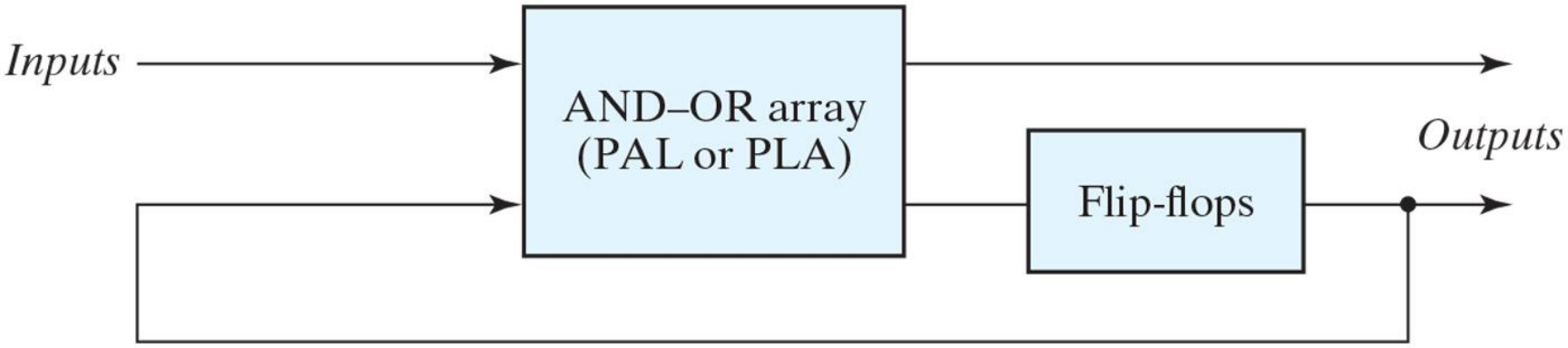
$$w = ABC' + A'B'CD'$$
$$x = A + BCD$$
$$y = A'B + CD + B'D'$$
$$z = ABC' + A'B'CD' + AC'D' + A'B'C'D$$
$$= w + AC'D' + A'B'C'D$$

	AND Inputs					
Product Term	***A***	***B***	***C***	***D***	***w***	**Outputs**
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	—	

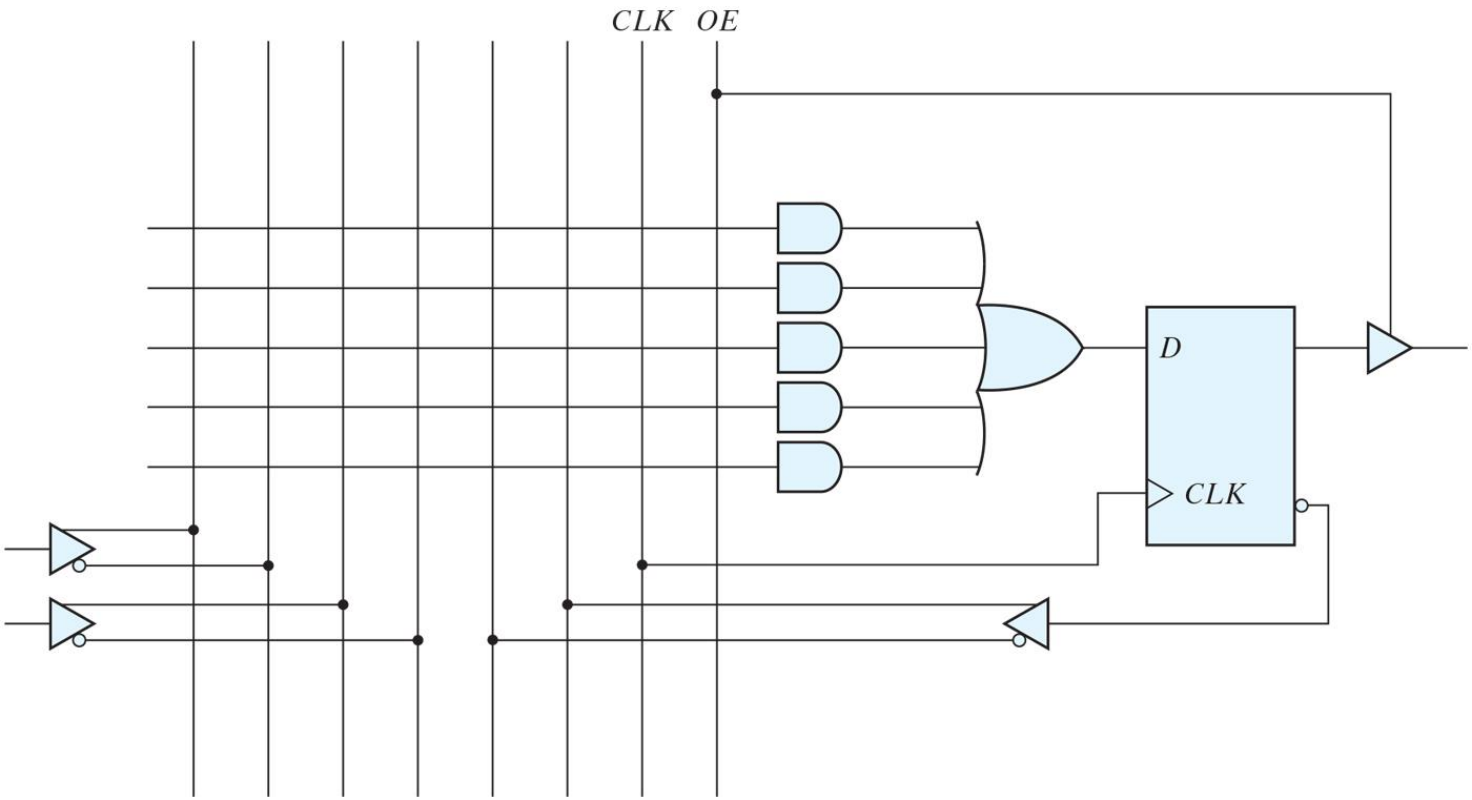
Fuse map for PAL



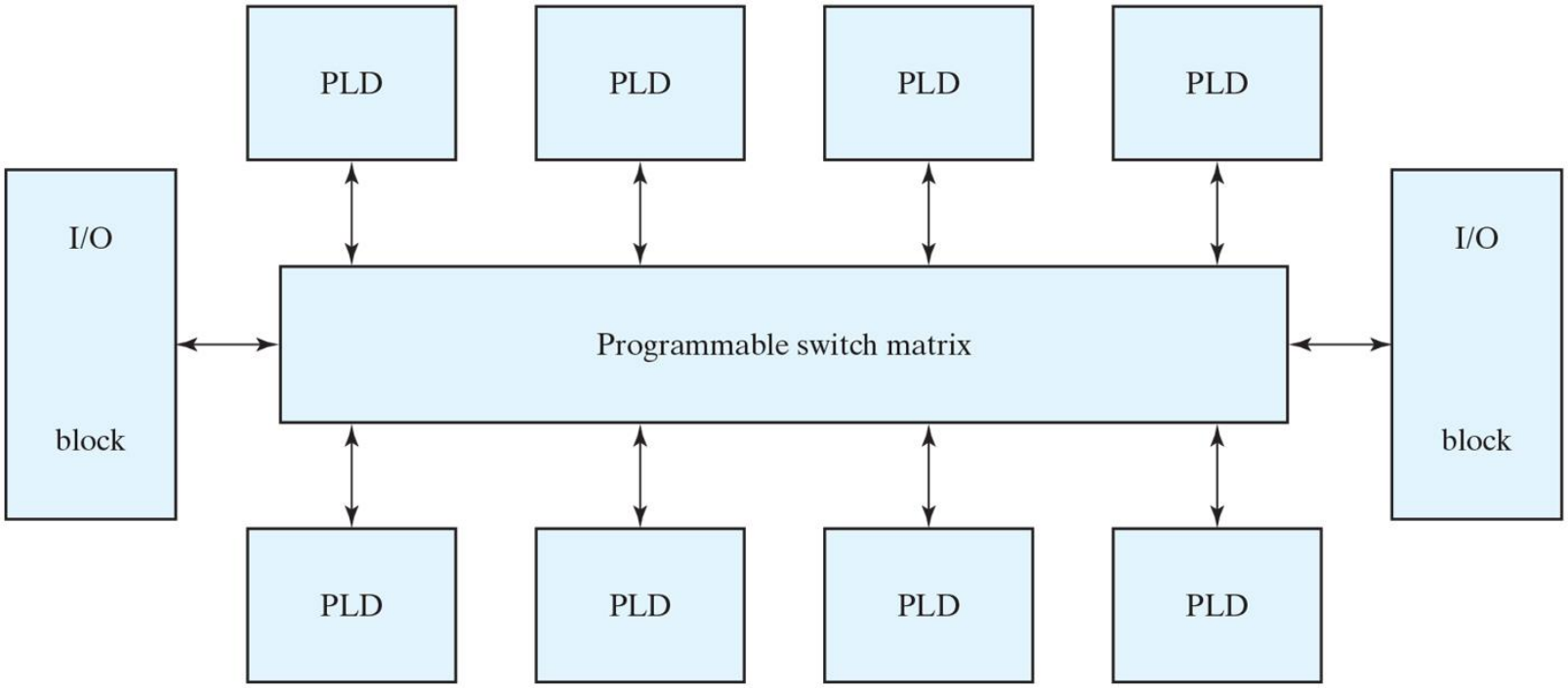
Sequential programmable logic device.



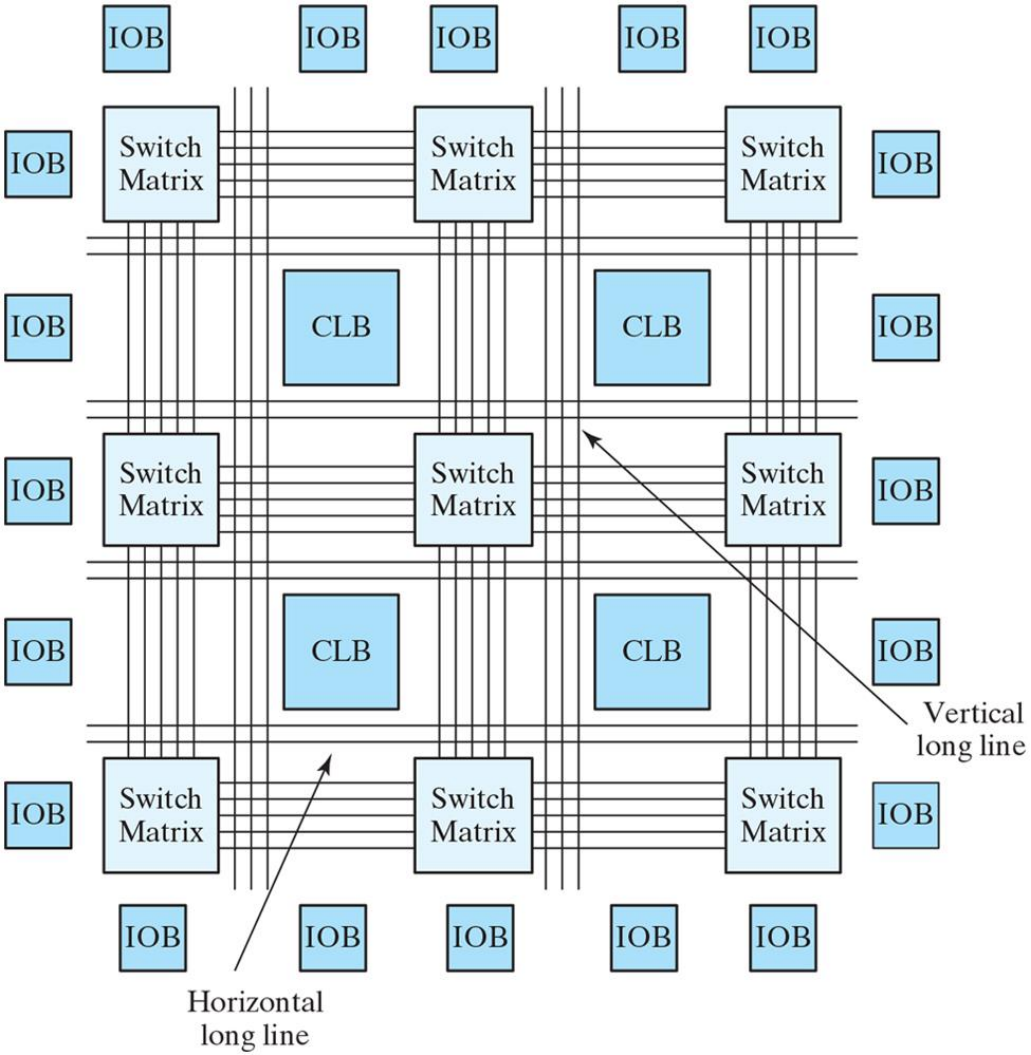
Basic macrocell logic.



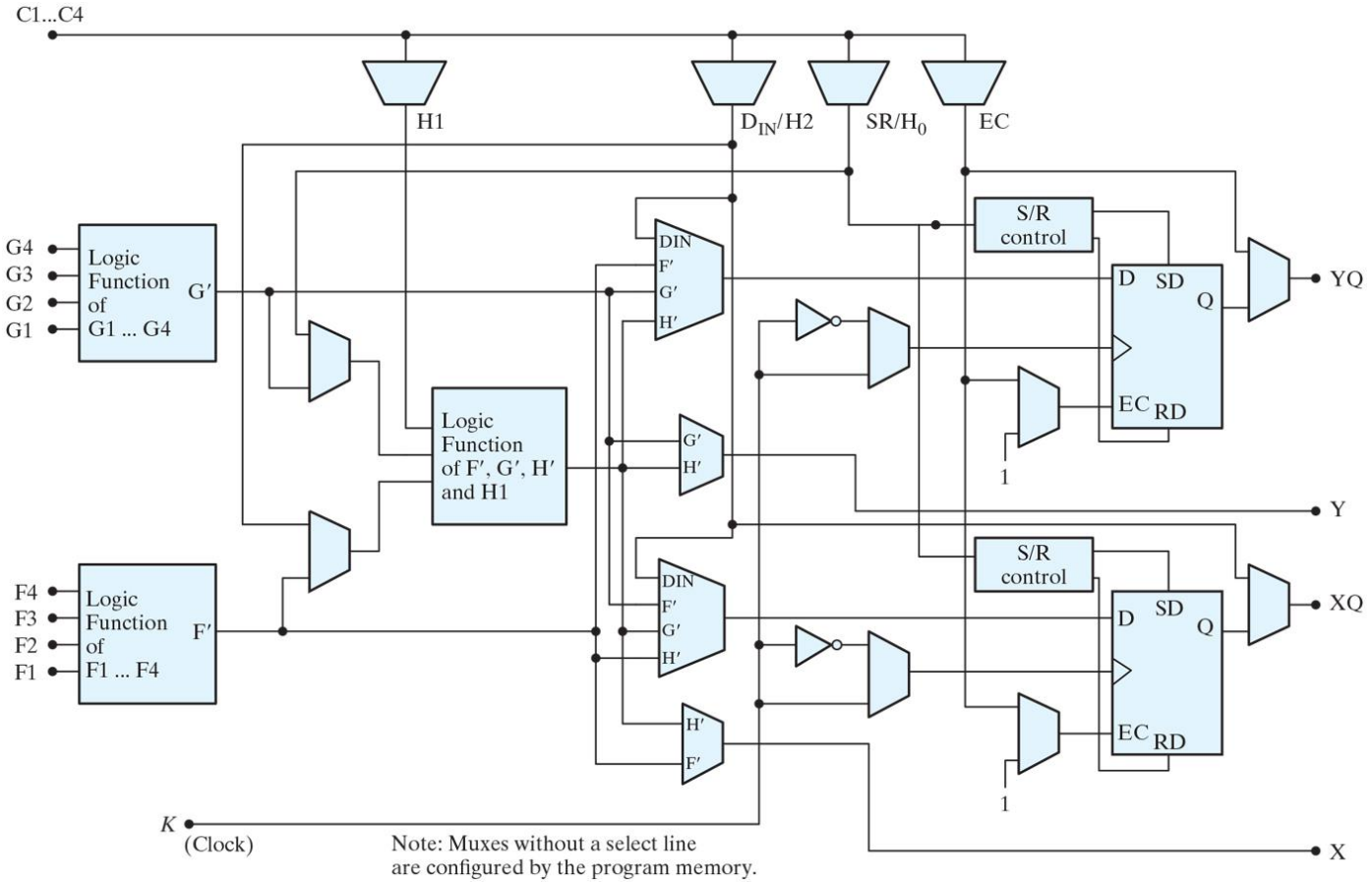
General CPLD configuration.



Basic architecture of Xilinx Spartan and predecessor devices.



CLB architecture.



Xilinx Spartan II architecture.