數位系統 Digital Systems

Department of Computer Science and Information Engineering,
Chaoyang University of Technology

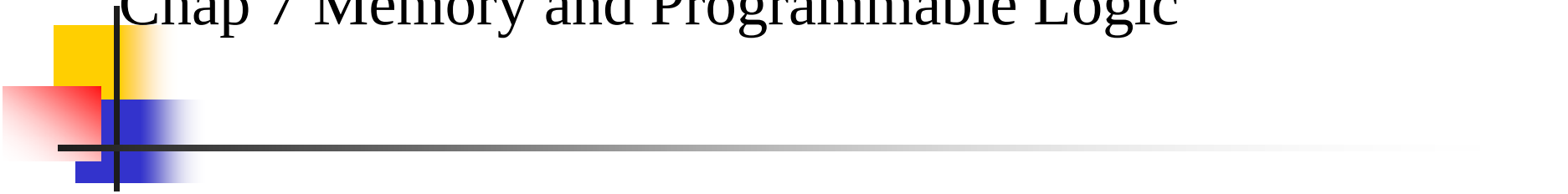
朝陽科技大學資工系

Speaker: Fuw-Yi Yang 楊伏夷

伏夷非征番,
道德經 察政章(Chapter 58) **伏**者潛藏也
道紀章(Chapter 14) 道無形象, 視之不可見者曰**夷**

Text Book: Digital Design 4th Ed.

Chap 7 Memory and Programmable Logic



7.1 Introduction
7.2 Random-Access Memory
7.3 Memory Decoding
7.4 Error Detection and Correction
7.5 Read-Only memory
7.6 Programmable Logic Array
7.7 Programmable Array Logic
7.8 Sequential Programmable Devices

Text Book: Digital Design 4th Ed.

Chap 7.1 Introduction

A **memory unit** is a device to which binary information is transferred for storage and from which information is retrieved when needed for processing.

There are two types of memories that are used in digital systems: **random-access memory** (RAM) and **read-only memory** (ROM).

RAM can perform both **Write** and **Read** operations.

ROM can perform only the **Read** operation.

Text Book: Digital Design 4th Ed.

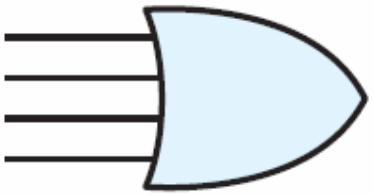
Chap 7.1 Introduction

ROM is a **programmable logic device** (PLD). The binary information that is stored within such a device is specified in some fashion and then embedded within the hardware in a process is referred to as programming the device.

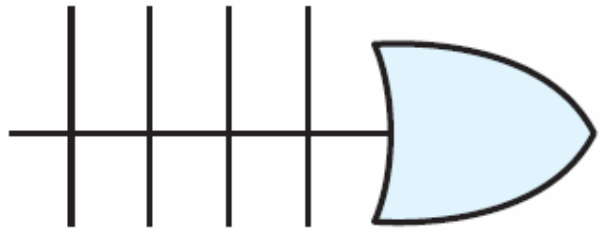
ROM is one example of a PLD. Other such units are the **programmable logic array** (PLA), **programmable array logic** (PAL), and the **field-programmable gate array** (FPGA).

Text Book: Digital Design 4th Ed.

Chap 7.1 Introduction – OR gate



(a) Conventional symbol



(b) Array logic symbol

Text Book: Digital Design 4th Ed.

Chap 7.2 Random-Access Memory

A memory unit is a collection of storage cells, together with associated circuits needed to transfer into and out of device. The architecture of memory is such that information can be selectively retrieved from any of its internal locations. The time it takes to **transfer information to or from any desired random location** is always the same – hence the name **random access memory**.

A memory unit stores binary information in groups of bits called **words.** A group of 8 bits is called a **byte**.

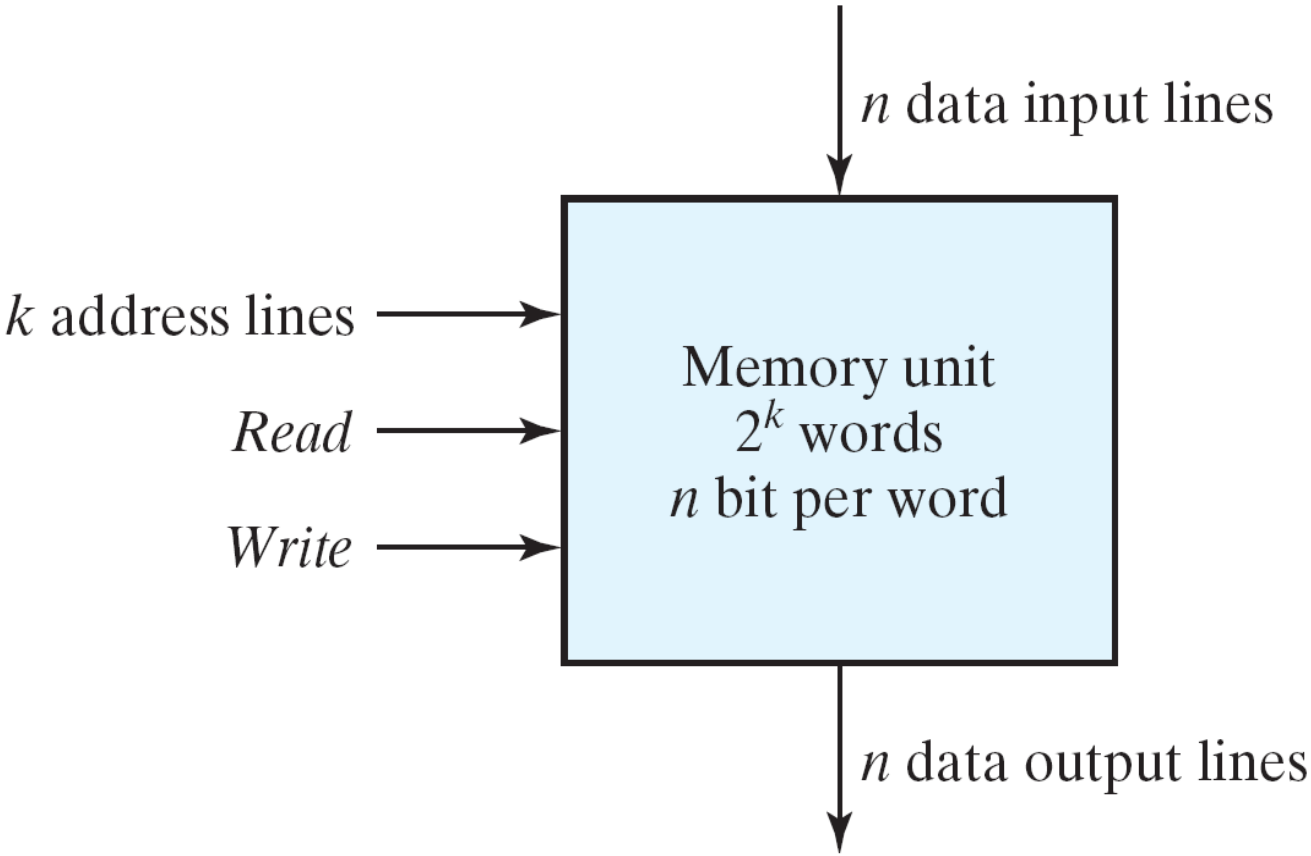
Text Book: Digital Design 4th Ed.

Chap 7.2 Random-Access Memory

Integrated circuit RAM units are available in two operating modes: **static** and **dynamic**.

Static RAM (SRAM) consists essentially of internal **latches** that store the binary information. Dynamic RAM (DRAM) stores the binary information in the form of electric charges on **capacitors** provided inside the chip by MOS transistors (require refreshing).

Text Book: Digital Design 4th Ed.
Chap 7.2 Random-Access Memory



The **write** input causes binary data to be transferred **into** the memory, and the **read** input causes binary data to be transferred **out of** 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

Contents of a 1024 x 16 memory

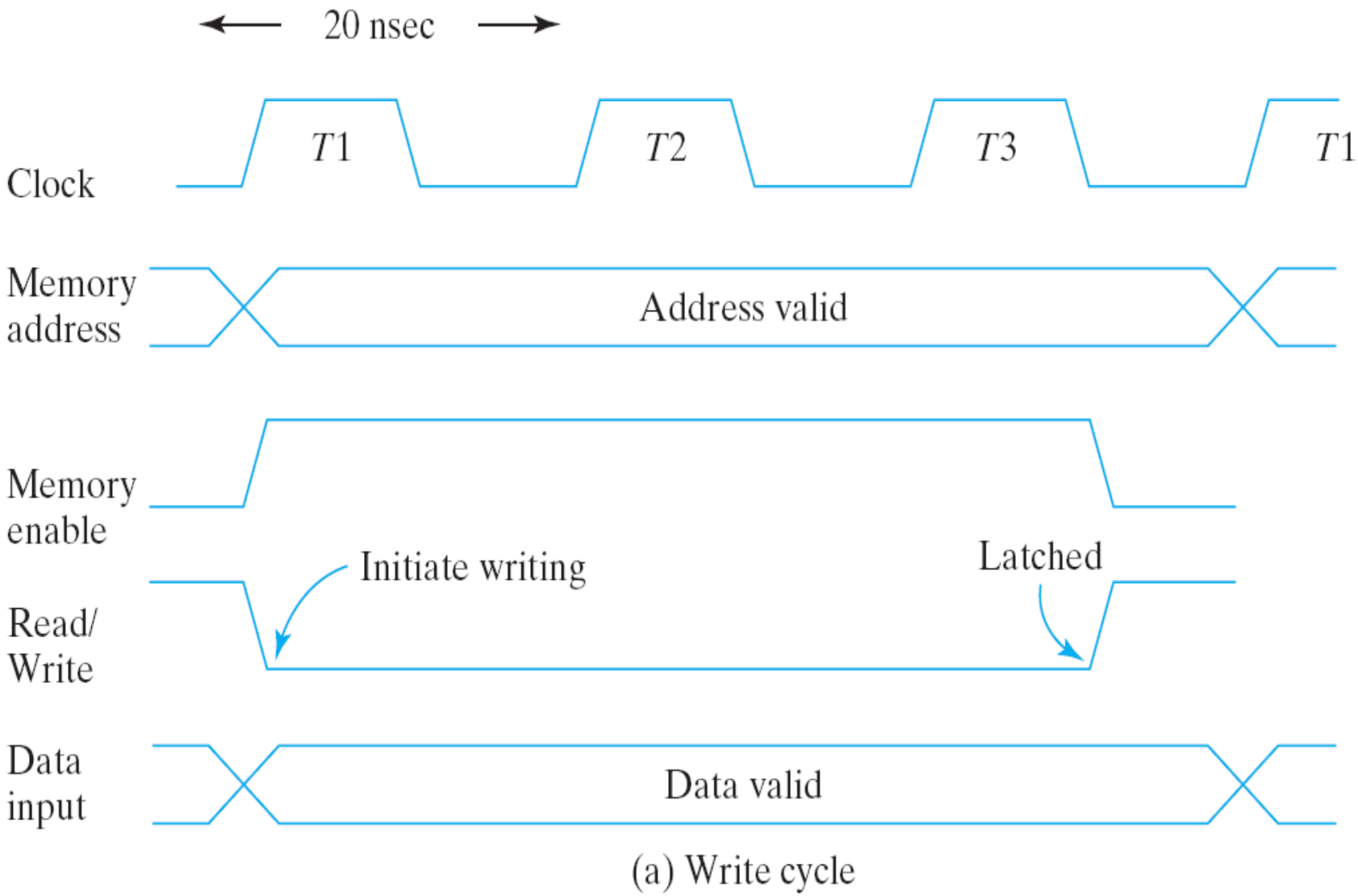
Text Book: Digital Design 4th Ed.

Chap 7.2 Random-Access Memory

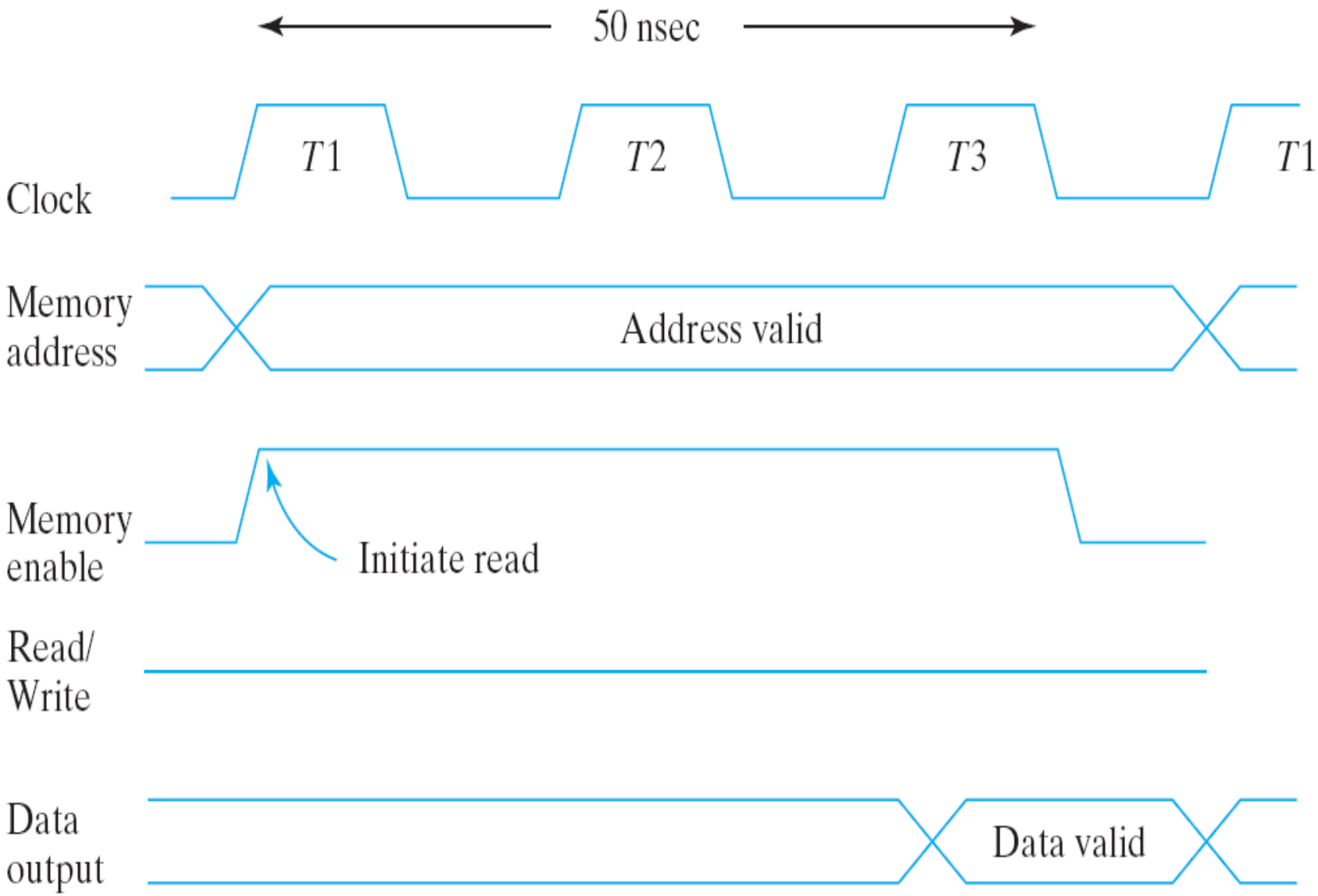
Table 7.1

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



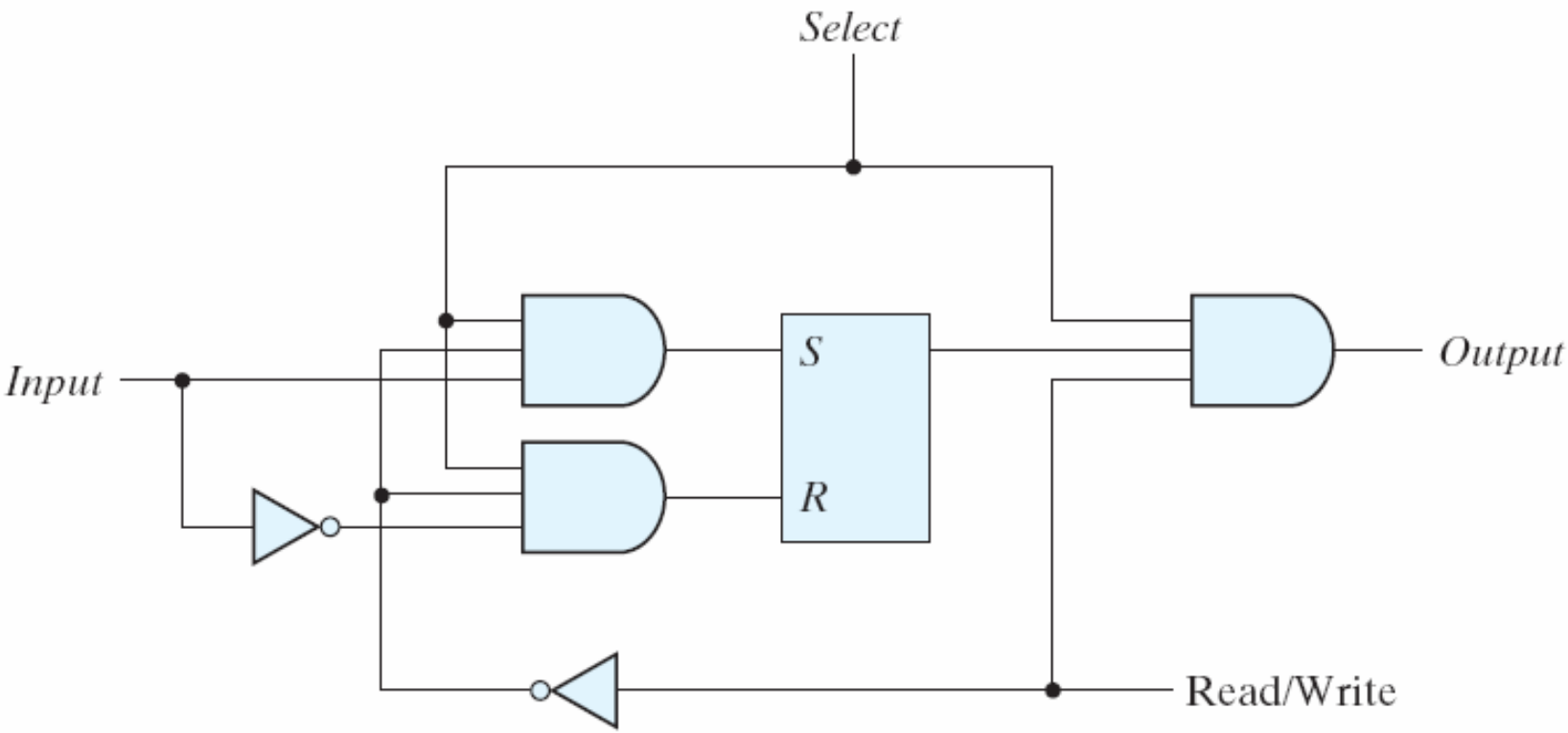
(a) Write cycle


50 nsec
T1
T2
T3
T1
Clock
Memory address
Address valid
Memory enable
Initiate read
Read/ Write
Data output
Data valid

(b) Read cycle

Text Book: Digital Design 4th Ed.

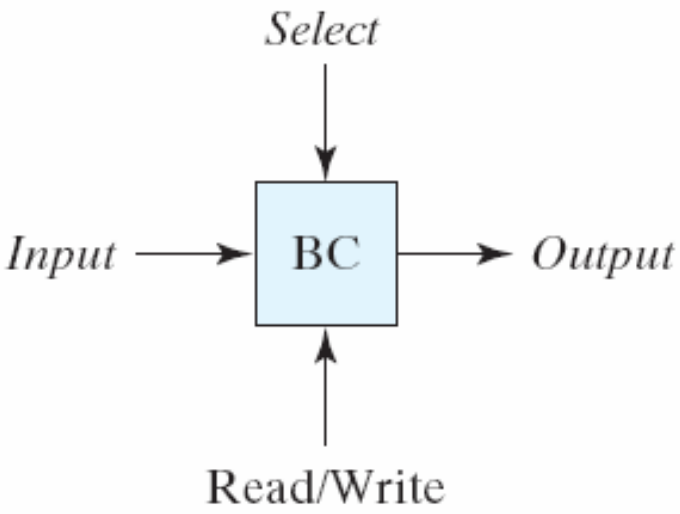
Chap 7.3 Memory Decoding



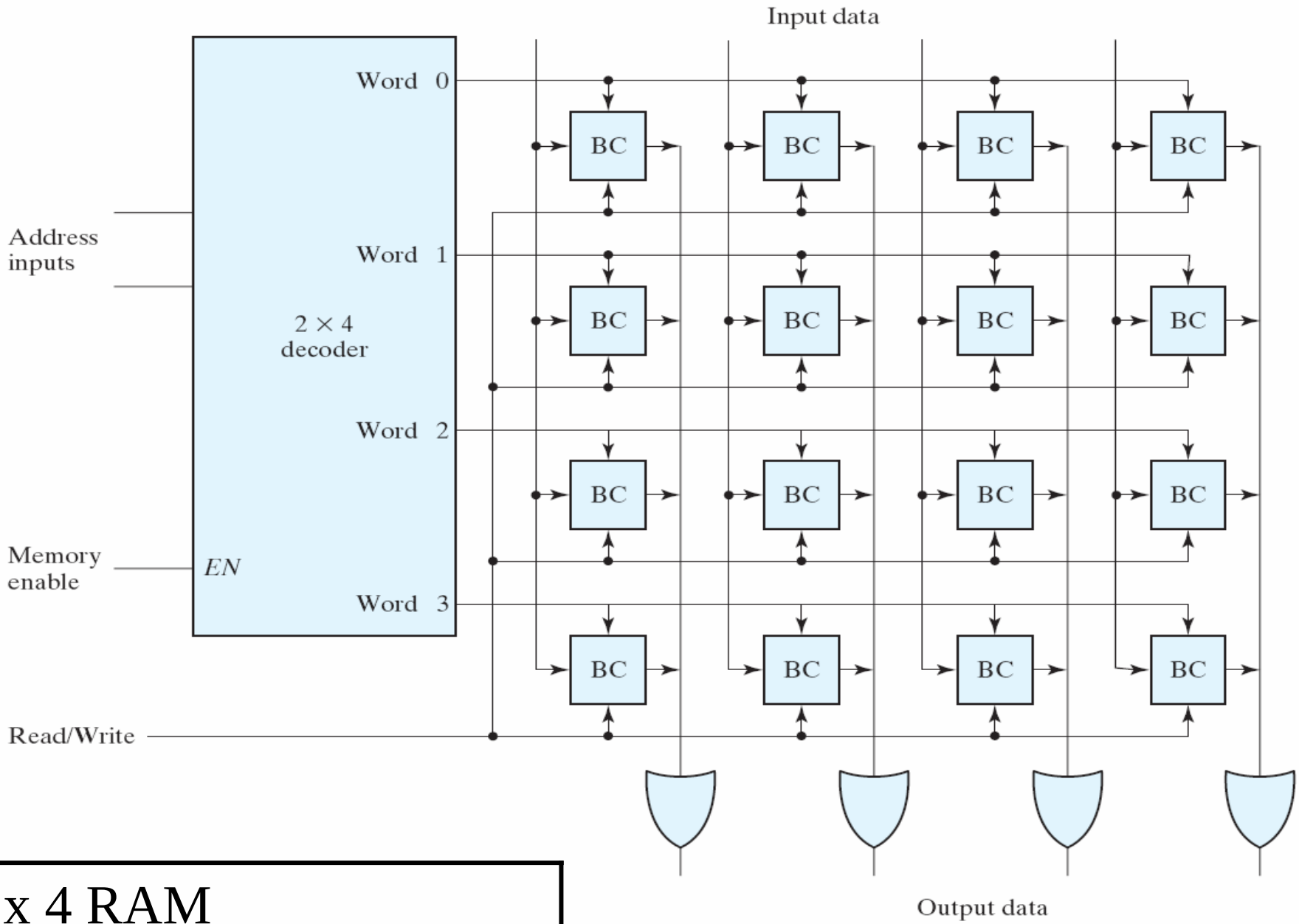
(a) Logic diagram

Text Book: Digital Design 4th Ed.

Chap 7.3 Memory Decoding



(b) Block diagram


Input data
Word 0
Word 1
Word 2
Word 3
Address inputs
2 × 4 decoder
Memory enable
EN
Read/Write
BC
Output data

4 x 4 RAM

Text Book: Digital Design 4th Ed.

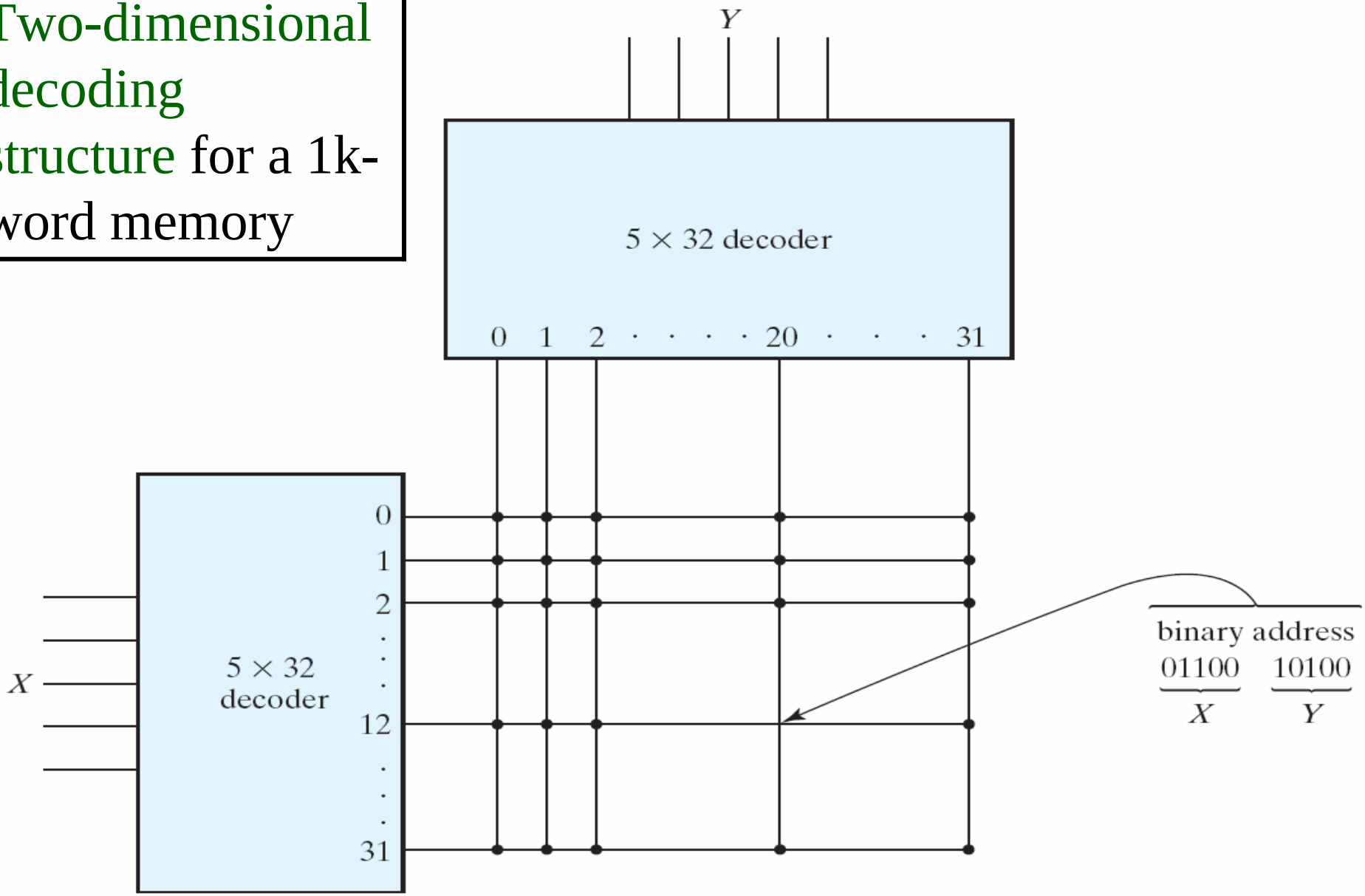
Chap 7.3 Memory Decoding – Coincident Decoding

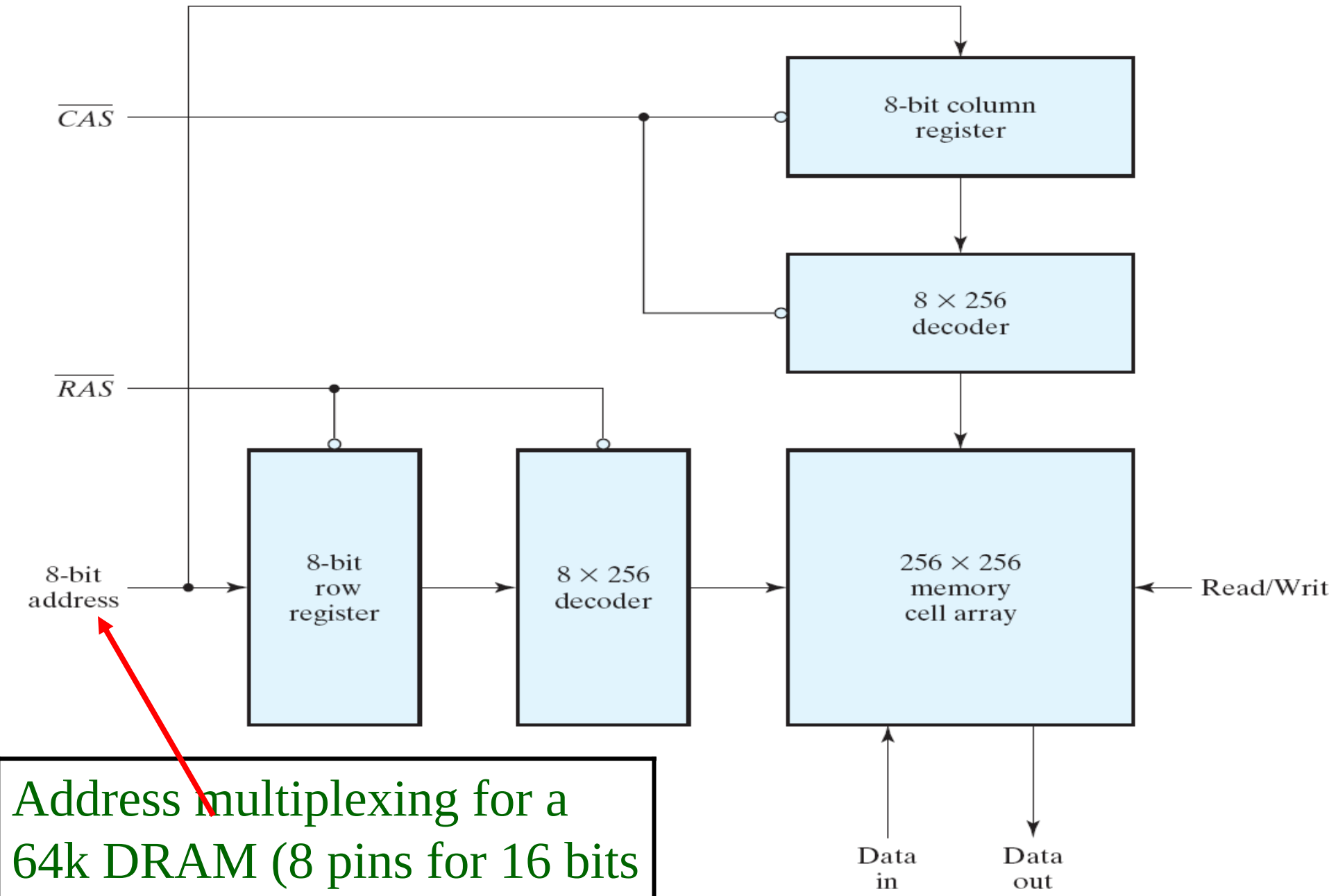
A decoder with ***k* inputs** and **2^k outputs** requires **2^k AND** gates with *k* inputs per gate.

The total number of gates and the number of inputs per gate can be reduced by employing **two decoders in a two-dimensional selection** scheme.

In this configuration, two ***k*/2-input** decoder are used instead of one ***k*-input** decoder. (next page).

Two-dimensional decoding structure for a 1k-word memory





Address multiplexing for a 64k DRAM (8 pins for 16 bits address)

Text Book: Digital Design 4th Ed.

Chap 7.4 Error detection and correction

Table 7.2

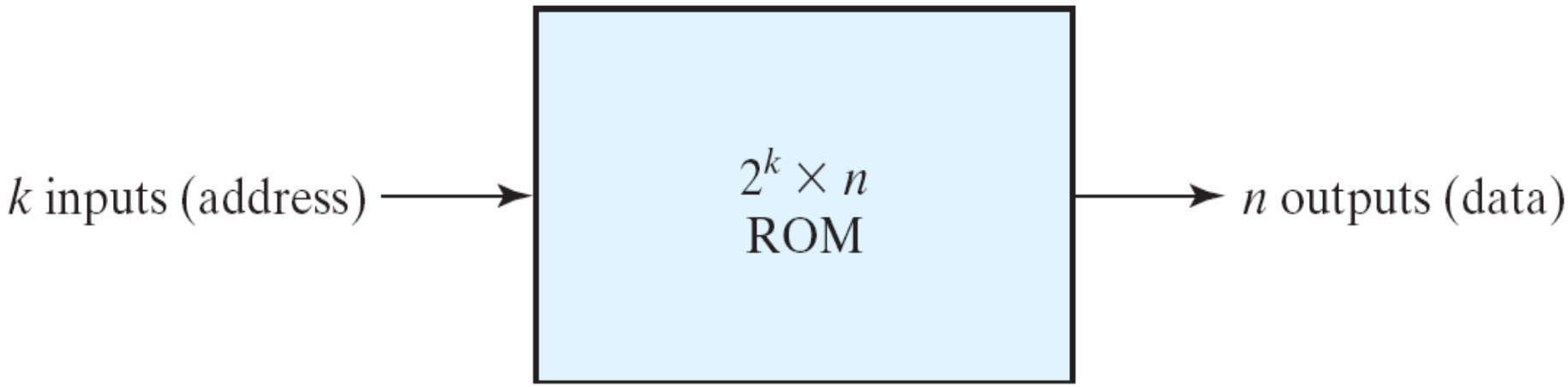
Range of Data Bits for k Check Bits

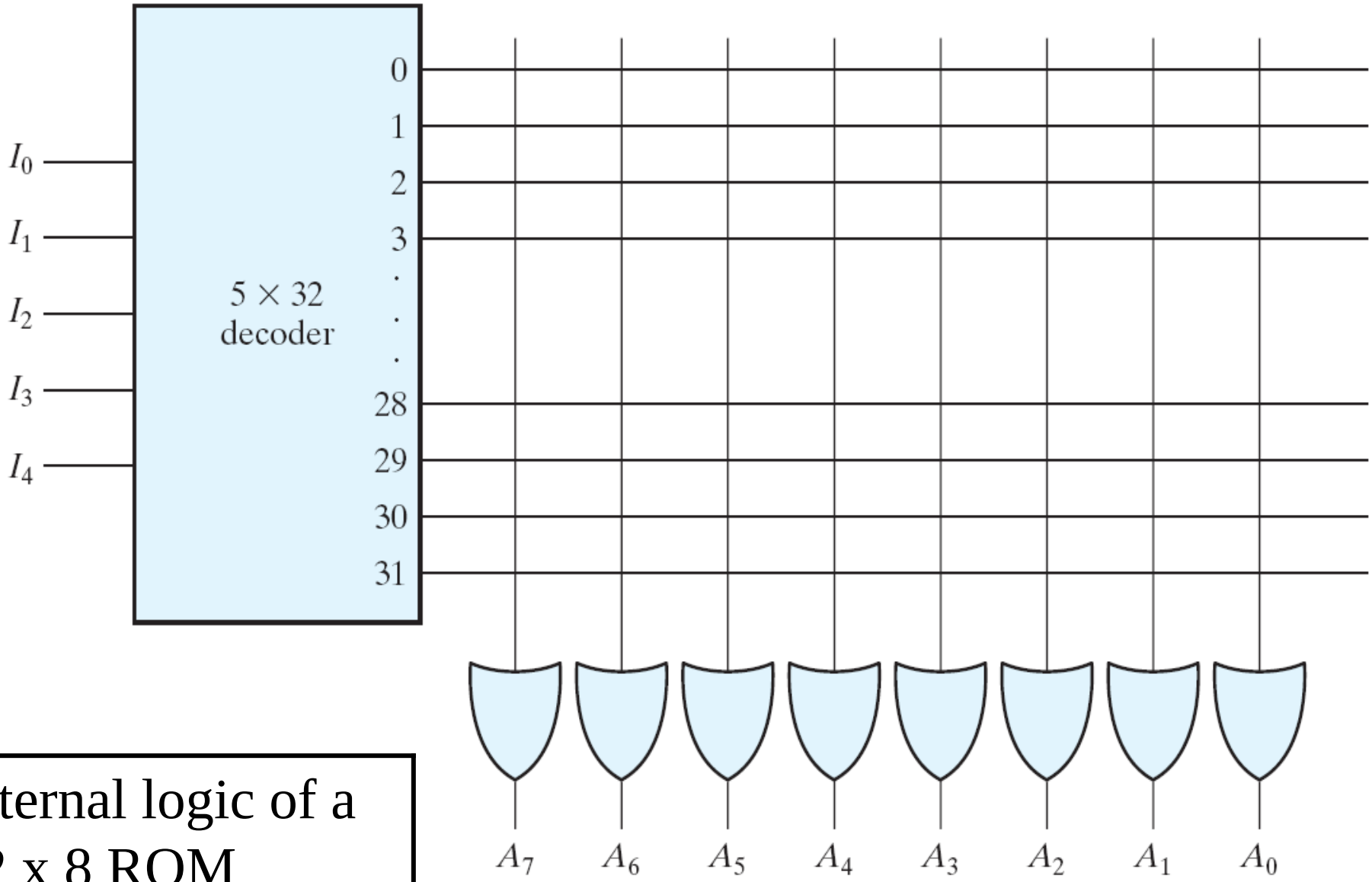
Number of Check Bits, *k*	Range of Data Bits, *n*
3	2–4
4	5–11
5	12–26
6	27–57
7	58–120

Text Book: Digital Design 4th Ed.

Chap 7.5 Read-Only Memory

A ROM is essentially a memory device in which permanent binary information is stored.





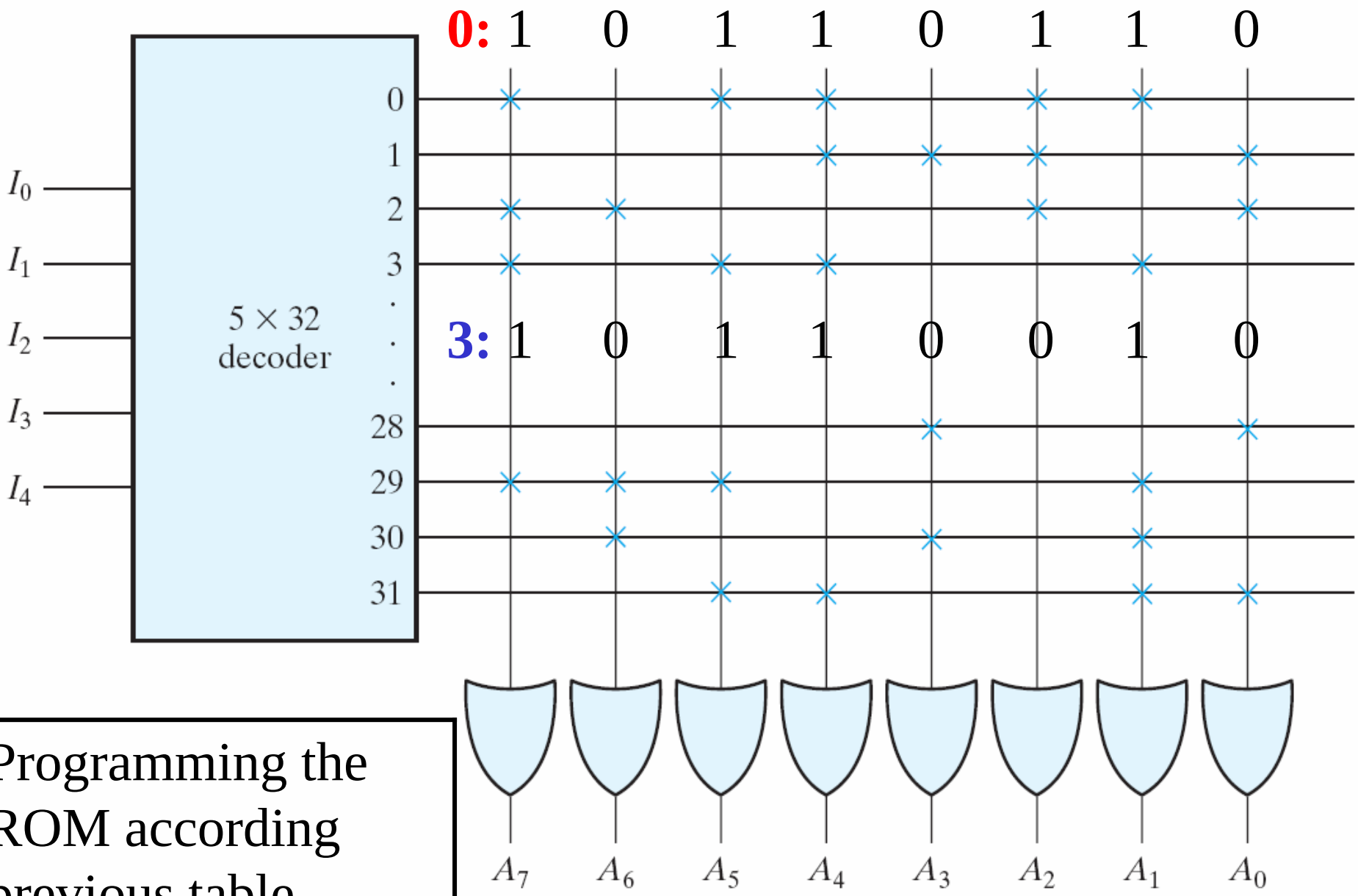
Internal logic of a 32 x 8 ROM

Text Book: Digital Design 4th Ed.

Chap 7.5 Read-Only Memory

Table 7.3
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 previous table

Text Book: Digital Design 4th Ed.

Chap 7.5 Read-Only Memory

Example 7.1 Design a combinational circuit using a ROM. The circuit accepts a three-bit number and outputs a binary number equal to the square of the input number.

Text Book: Digital Design 4th Ed.

Chap 7.5 Read-Only Memory – Example 7.1

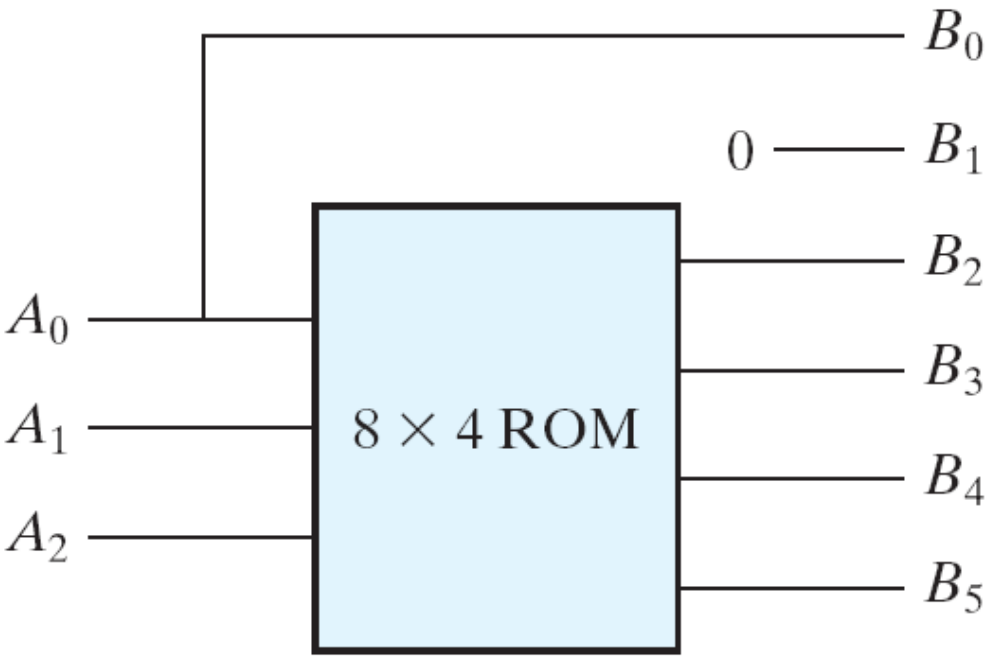
Table 7.4

Truth Table for Circuit of Example 7.1

Inputs			Outputs						
A_2	A_1	A_0	B_5	B_4	B_3	B_2	B_1	B_0	Decimal
0	0	0	0	0	0	0	0	0	0
0	0	1	0	0	0	0	0	1	1
0	1	0	0	0	0	1	0	0	4
0	1	1	0	0	1	0	0	1	9
1	0	0	0	1	0	0	0	0	16
1	0	1	0	1	1	0	0	1	25
1	1	0	1	0	0	1	0	0	36
1	1	1	1	1	0	0	0	1	49

Text Book: Digital Design 4th Ed.

Chap 7.5 Read-Only Memory – Example 7.1



(a) Block diagram

A_2	A_1	A_0	B_5	B_4	B_3	B_2
0	0	0	0	0	0	0
0	0	1	0	0	0	0
0	1	0	0	0	0	1
0	1	1	0	0	1	0
1	0	0	0	1	0	0
1	0	1	0	1	1	0
1	1	0	1	0	0	1
1	1	1	1	1	0	0

(b) ROM truth table

Text Book: Digital Design 4th Ed.

Chap 7.5 Read-Only Memory—Types of ROMs

The required paths in a ROM may be **programmed in four different ways**.

1. Mask programming
2. Programmable read-only memory (PROM)
3. Erasable (EPROM)
4. Electrically Erasable (EEPROM)

Flash memory devices are similar to EEPROMs, but have additional built-in circuitry to selectively program and erase the device in-circuit, eithout the need for a special programmer.

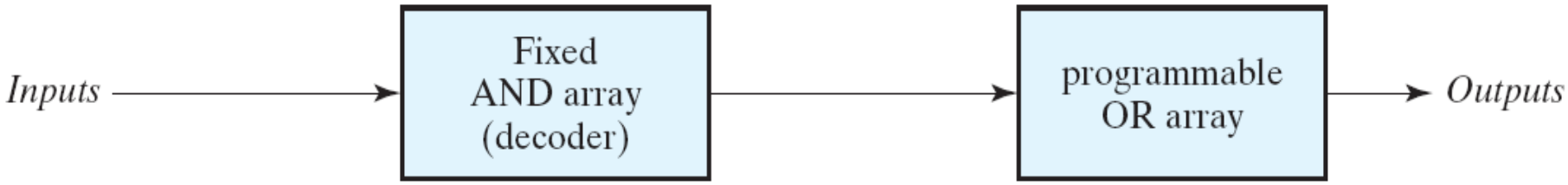
Text Book: Digital Design 4th Ed.

Chap 7.5 Read-Only Memory—Combinational PLDs

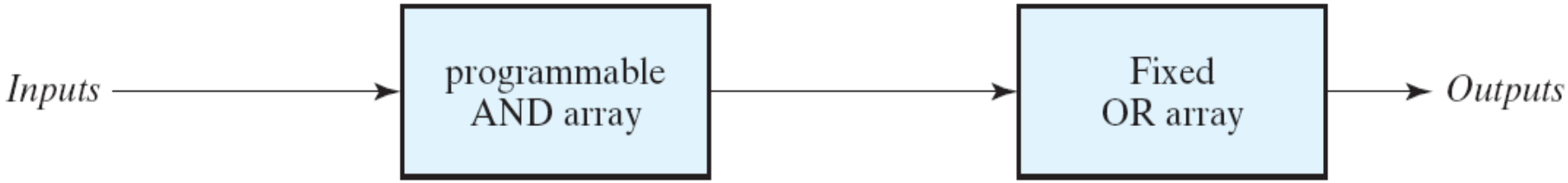
The PROM is a **combinational programmable logic device** (PLD) – an integrated circuit with programmable gates divided into an **AND array** and an **OR array** to provide an **AND-OR sum-of-product** implementation.

There are three major types of combinational PLDs, differing in **the placement of the programmable connections** in the AND-OR arrays.

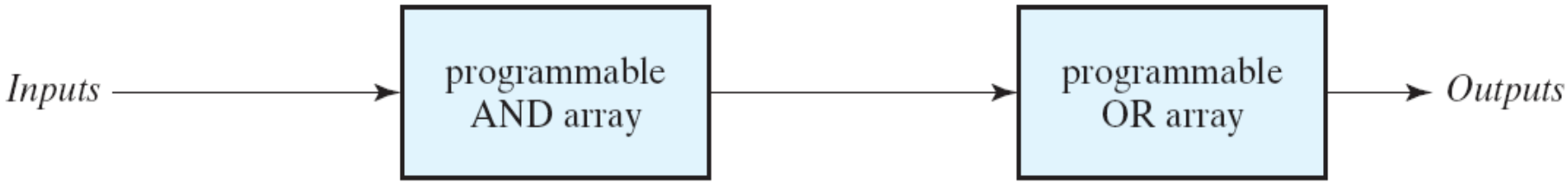
1. PROM next page
2. PAL
3. PLA



(a) Programmable read-only memory (PROM)



(b) Programmable array logic (PAL)



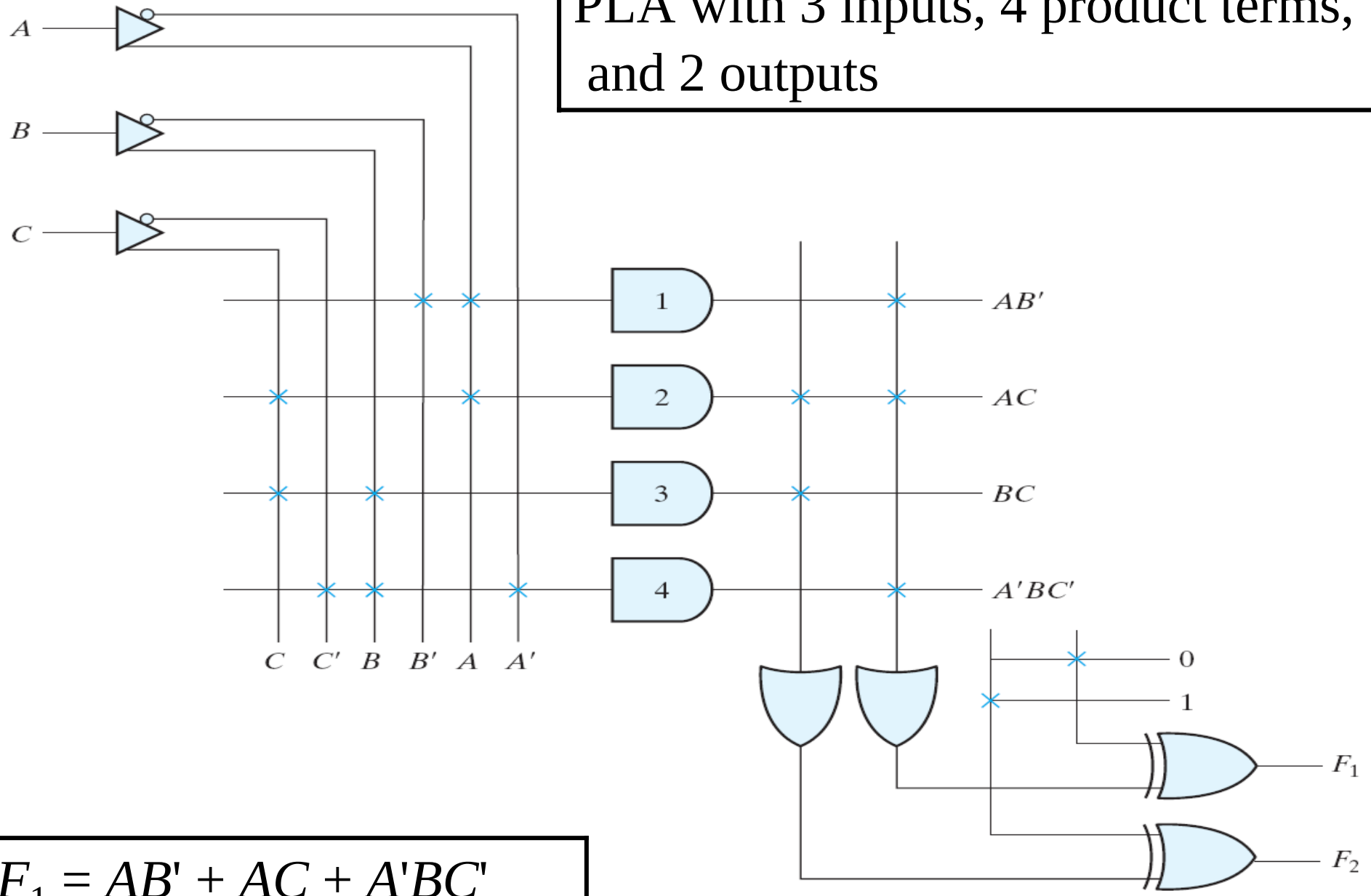
(c) Programmable logic array (PLA)

Text Book: Digital Design 4th Ed.

Chap 7.6 Programmable Logic Array

The PLA is similar in concept to the PROM, except that the PLA does not provide fully decoding of the variables and does not generate all the minterms.

PLA with 3 inputs, 4 product terms, and 2 outputs



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

Table 7.5
PLA Programming Table

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

		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	—

Note: See text for meanings of dashes.

Fuse map of F_1 and F_2 specified in a tabular form

Example 7.2 Implement the following Boolean functions with a PLA: $F_2(A, B, C) = \Sigma(0, 5, 6, 7)$

$F_1(A, B, C) = \Sigma(0, 1, 2, 4) = (AB + AC + BC)'$

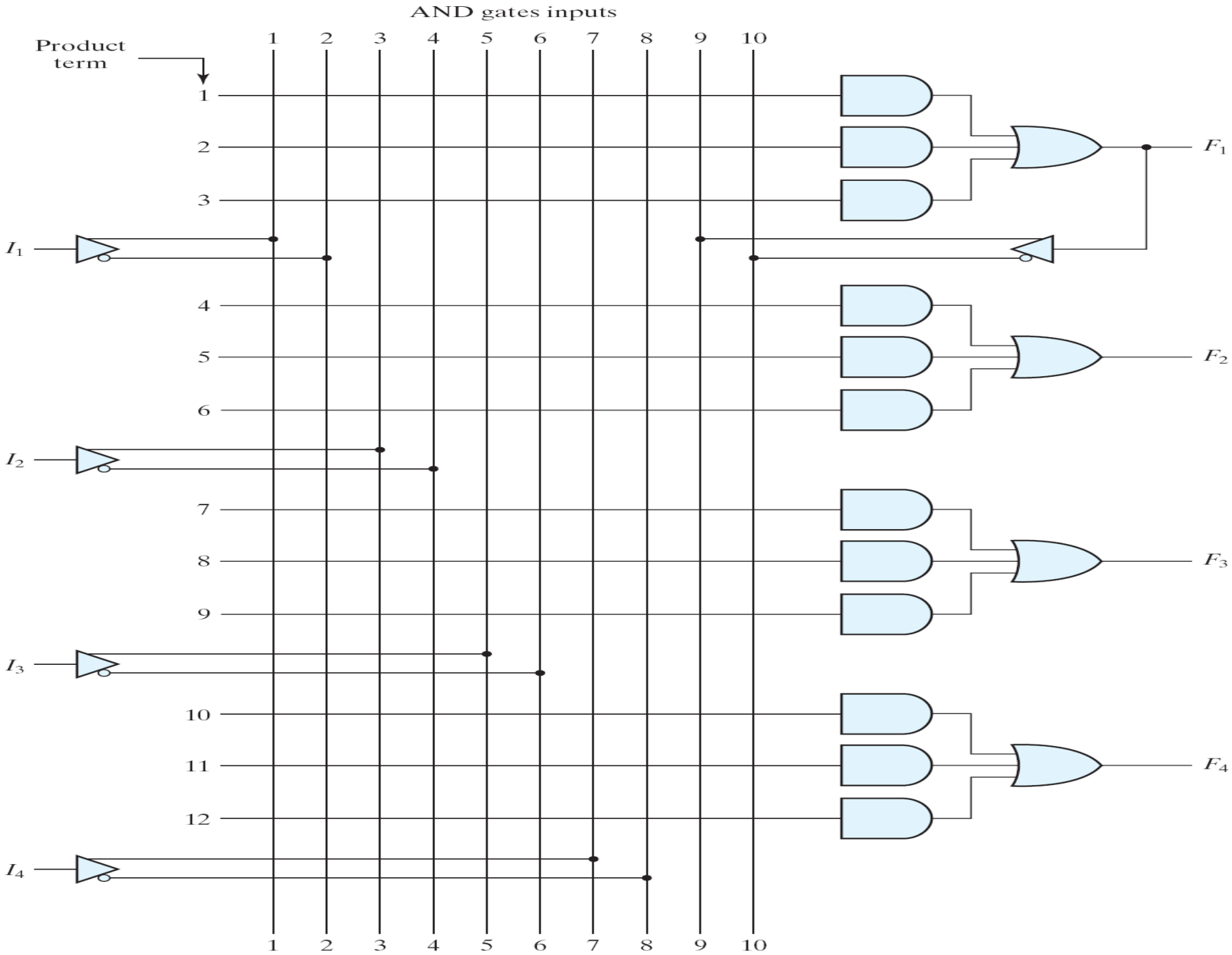
PLA programming table

	Product term	Inputs			Outputs (C)	Outputs (T)
		A	B	C	F_1	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

Text Book: Digital Design 4th Ed.

Chap 7.7 Programmable Array Logic PAL

The PAL is a programmable logic device with a **fixed OR array** and **programmable AND array**.


AND gates inputs
1 2 3 4 5 6 7 8 9 10
Product term
1
2
3
4
5
6
7
8
9
10
11
12
I_1
I_2
I_3
I_4
F_1
F_2
F_3
F_4
1 2 3 4 5 6 7 8 9 10

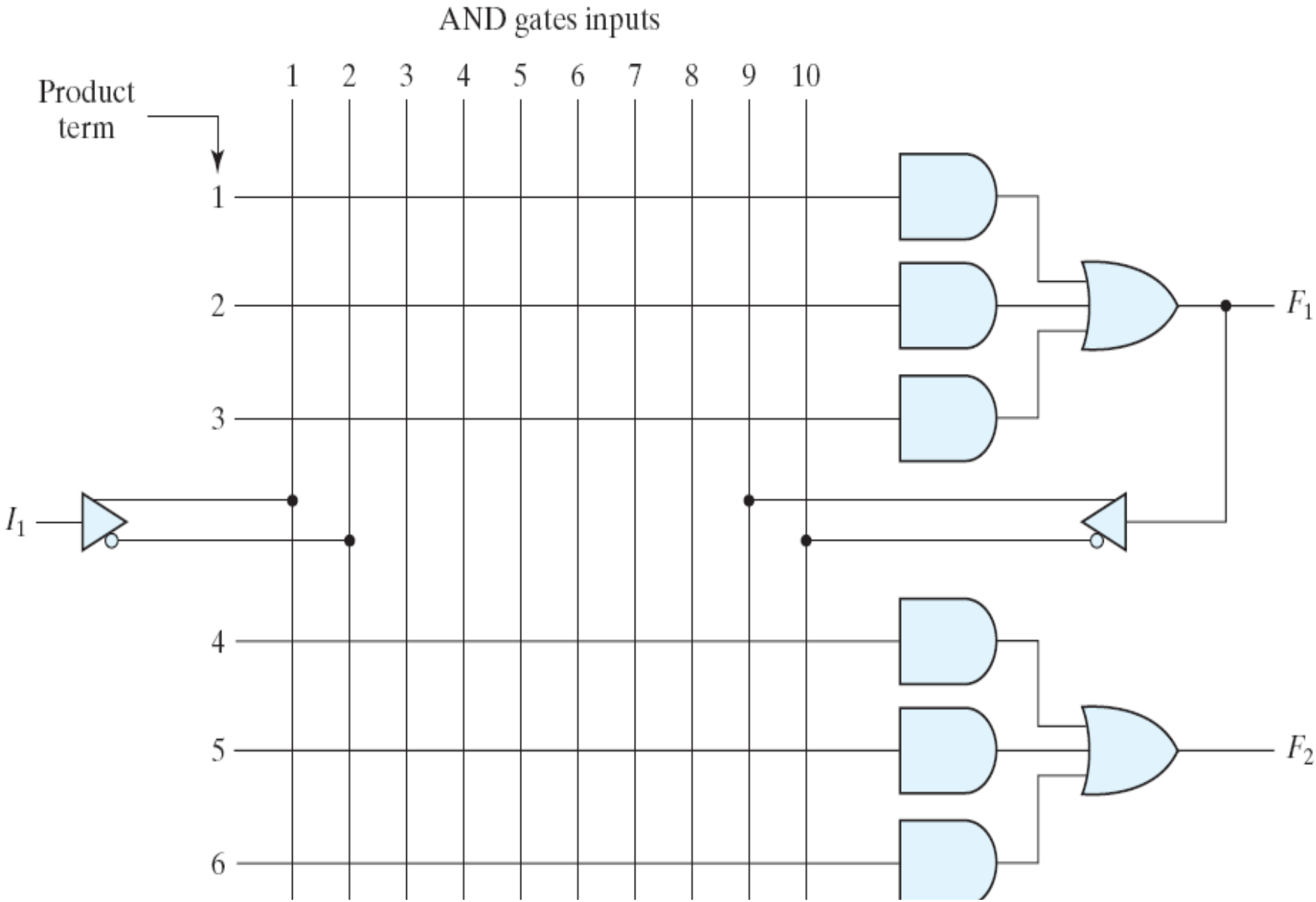

AND gates inputs
1 2 3 4 5 6 7 8 9 10
Product term
1
2
3
4
5
6
I_1
F_1
F_2

Table 7.6
PAL Programming Table

Product Term	AND Inputs 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	—	

Text Book: Digital Design 4th Ed.

Chap 7.8 Sequential Programmable Devices

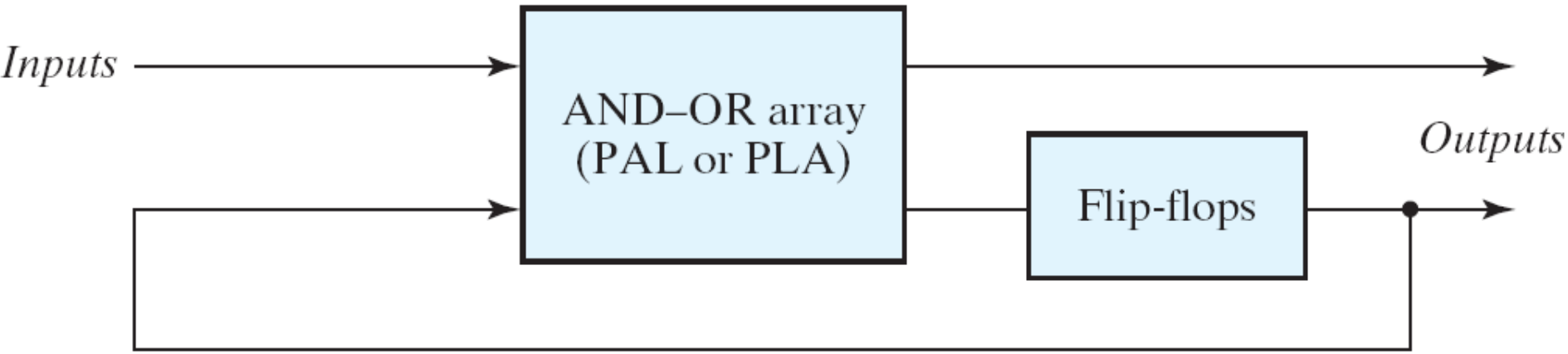
Digital systems are designed with flip-flops and gates. Since the PLD consists of only gates, it is necessary to

Sequential programmable devices include both flip-flops and gates. There are three major types of them.

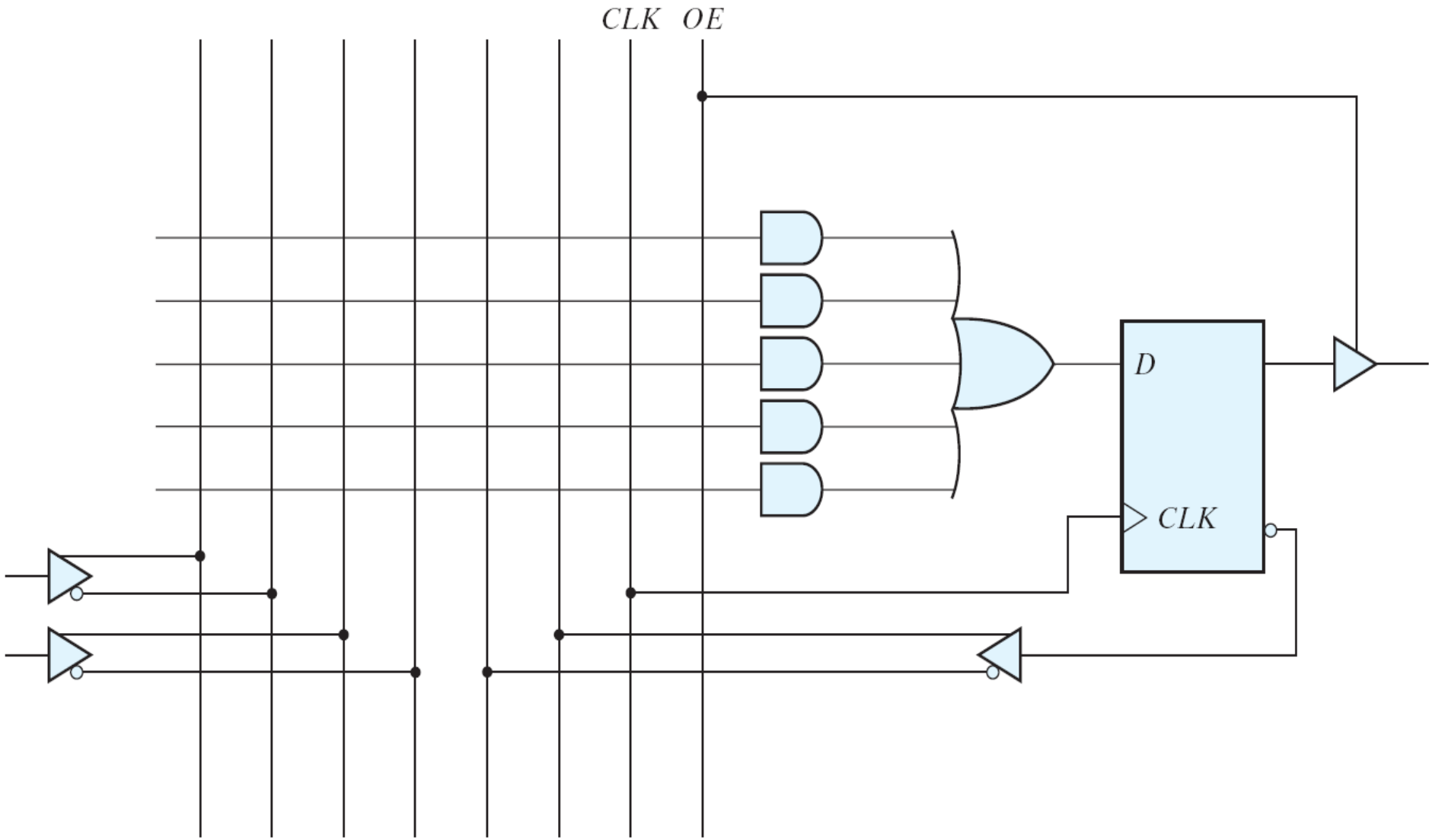
1. Sequential (simple) programmable logic device (SPLD)
2. Complex programmable logic device (CPLD)
3. Field-programmable gate array (FPGA)

Text Book: Digital Design 4th Ed.

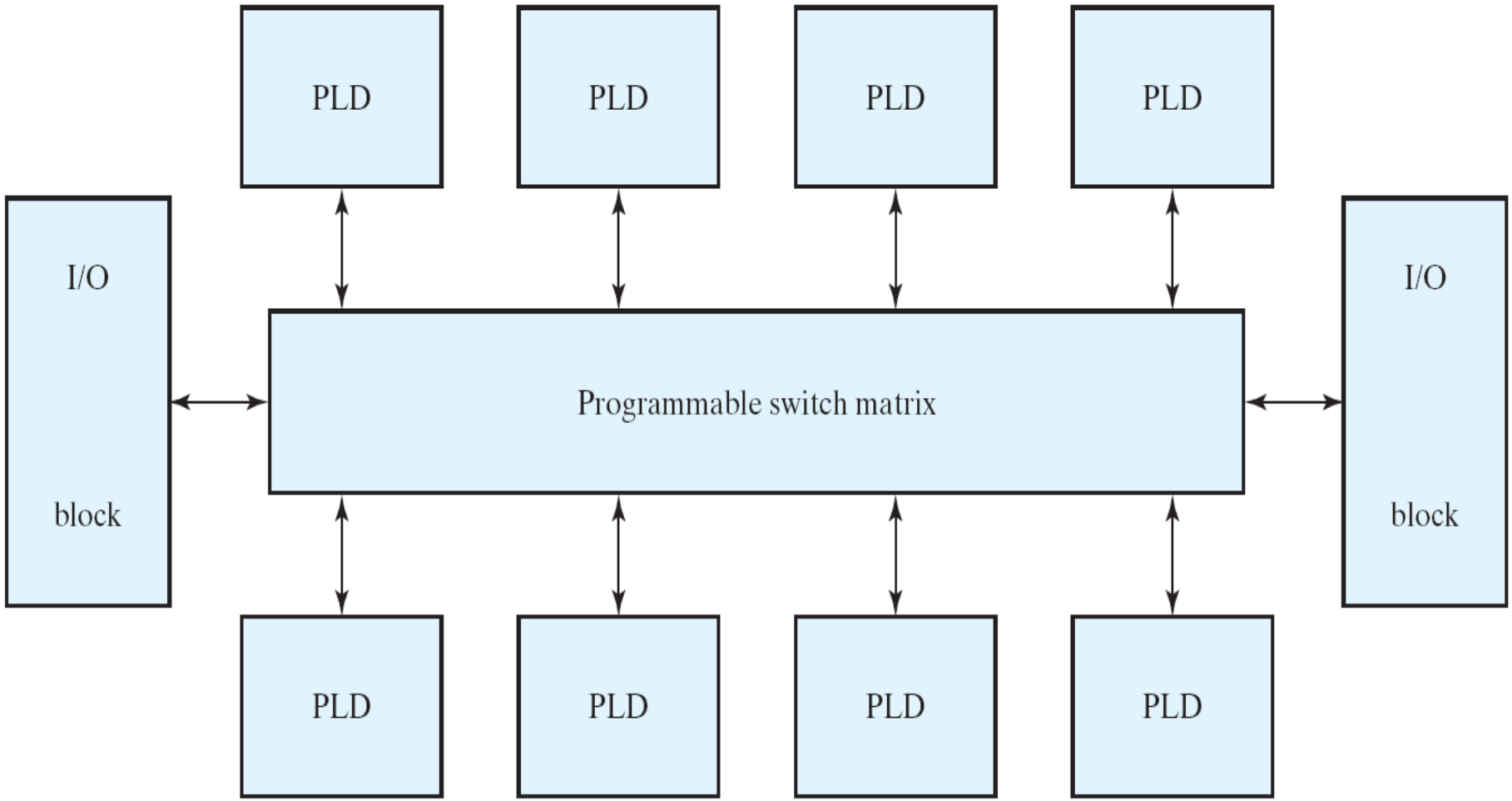
Chap 7.8 Sequential Programmable Devices



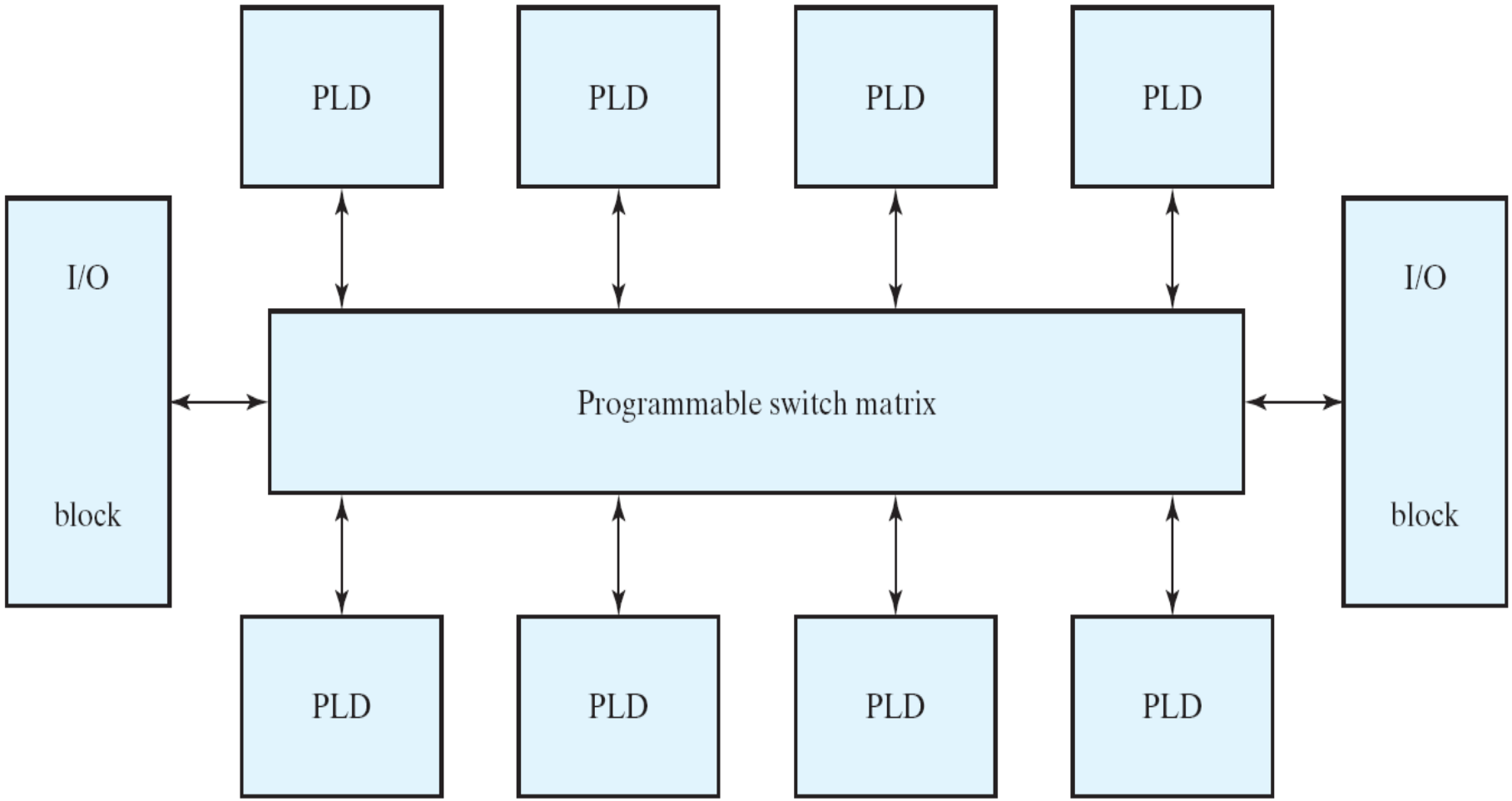
Sequential Programmable logic device



Basic macrocell logic, a typical SPLD has from 8~10 macrocells within one IC package



General CPLD configuration



General CPLD configuration