# Chapter 7. Memory and Programmable Logic

7.1 Introduction

- Memory unit:
  - A device to which binary information is transferred for storage and from which information is retrieved when needed for processing
  - A collection of cells capable of storing a large quantity of binary information

- RAM
  - Random access memory
  - Write/read operations

- ROM
  - Read only memory
  - Programmable logic device (PLD), programmable logic array (PLA), programmable array logic (PAL), field-programmable gate array (FPGA)
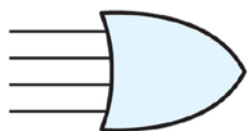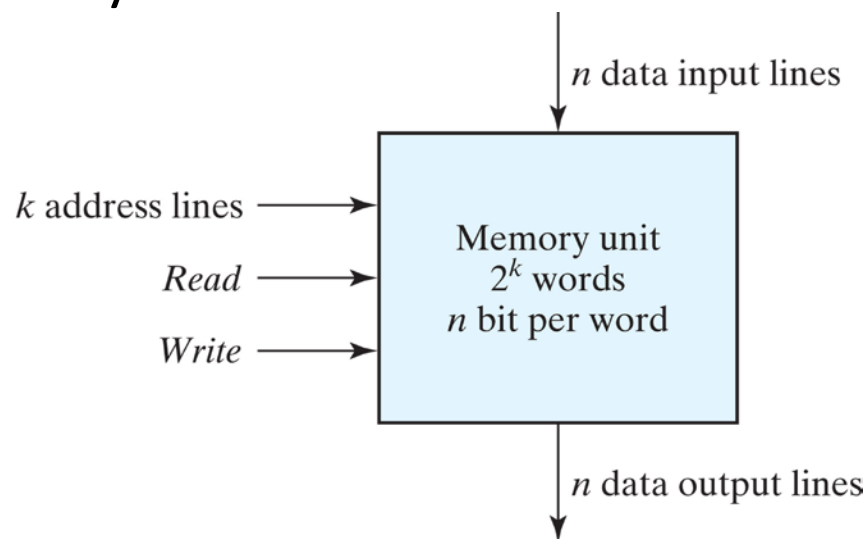
(a) Conventional symbol

(b) Array logic symbol

FIGURE 7.1 Conventional and array logic diagrams for OR gate

7.2 Random-Access Memory

- Time to transfer information to or from any desired random location is same

- Words:
    - Binary information in groups of bits
    - Byte: A group of 8 bits
    - 16-bit word, 32-bit word

- Data input and output lines

- Address selection lines
    - Identification number for selecting one particular word

- Control lines specify the direction of transfer

- K(kilo=2^{10}), M(mega=2^{20}), G(giga=2^{30})

n data input lines

k address lines ⟶

Memory unit
2^k words
n bit per word

Read ⟶

Write ⟶

n data output lines

Copyright ©2013 Pearson Education, publishing as Prentice Hall

3 **FIGURE 7.2**   Block diagram of a memory unit

Memory address

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

Copyright ©2013 Pearson Education, publishing as Prentice Hall

FIGURE 7.3 Contents of a 1024 × 16 memory

Write and Read Operations

- Write operation
    1. Apply the binary address of the desired word to the address lines
    2. Apply the data bits that must be stored in memory to the data input lines
    3. Activate the *write* input

- Read operation
    1. Apply the binary address of the desired word to the address lines
    2. Activate the read input

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

Timing Waveforms

- ## Access time:
  - Time required to select a word and read it

- ## Cycle time:
  - Time required to complete a write operation

- ## Example
  - CPU 50MHz clock (20ns)
  - 50 ns maximum cycle time


(a) Write cycle

(b) Read cycle
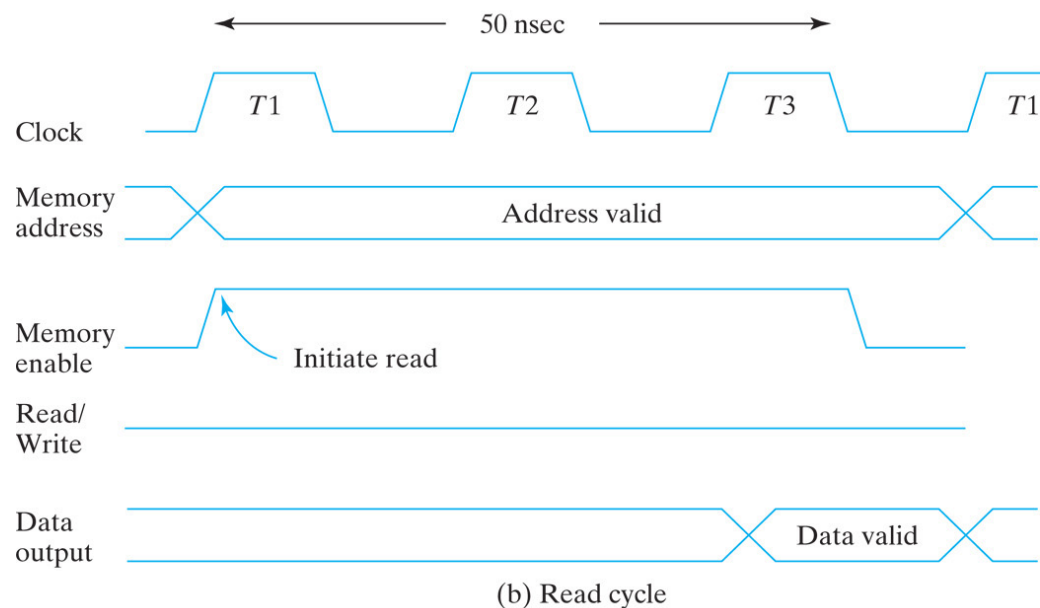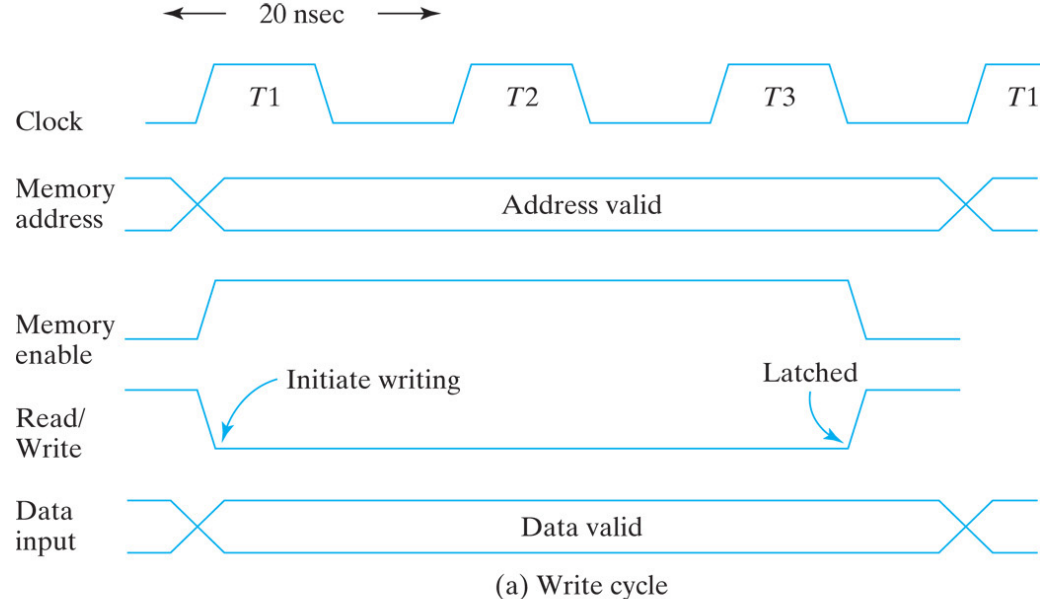
Types of Memories

- Access time
  - Random-access memory: same access time
  - Sequential-access memory: magnetic disc or tape, variable access time

- Operating modes
  - Static RAM(SRAM): internal latches
  - Dynamic RAM(DRAM): MOS transistors for storing binary information as electric charges on the capacitors; require refreshing the dynamic memory (recharge)

- Volatile
  - Loss stored information when power down (CMOS integrated circuit RAM ; SRAM, DRAM)

- Nonvolatile
  - Retain the stored information after the removal of power(Magnetic disk)

7.3 Memory Decoding

- RAM of m words and n bits per word: m X n binary storage cells
- Associated decoding circuits for selecting individual words
- SR latch
- To pack as many cells as possible in the small area

(a) Logic diagram

(b) Block diagram
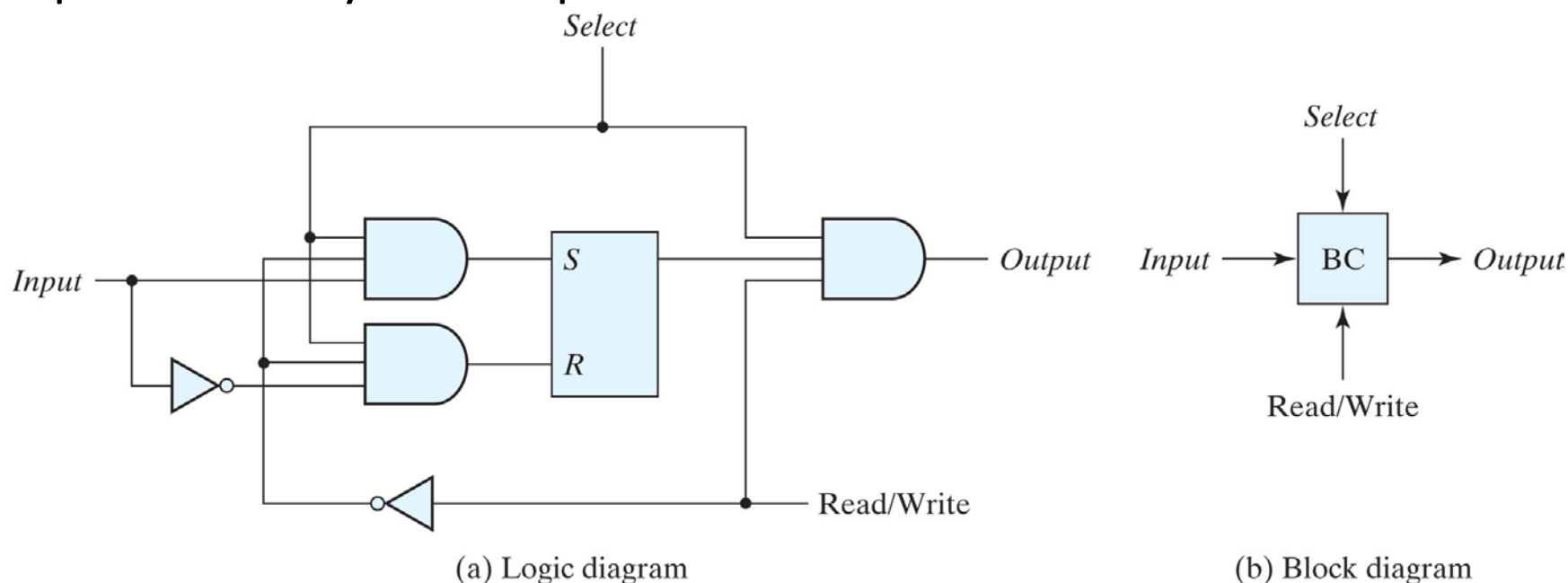
Copyright ©2013 Pearson Education, publishing as Prentice Hall

FIGURE 7.5 Memory cell

8

Logical Construction of RAM

- 4 words of 4 bits each

- 2^k words of n bits per word

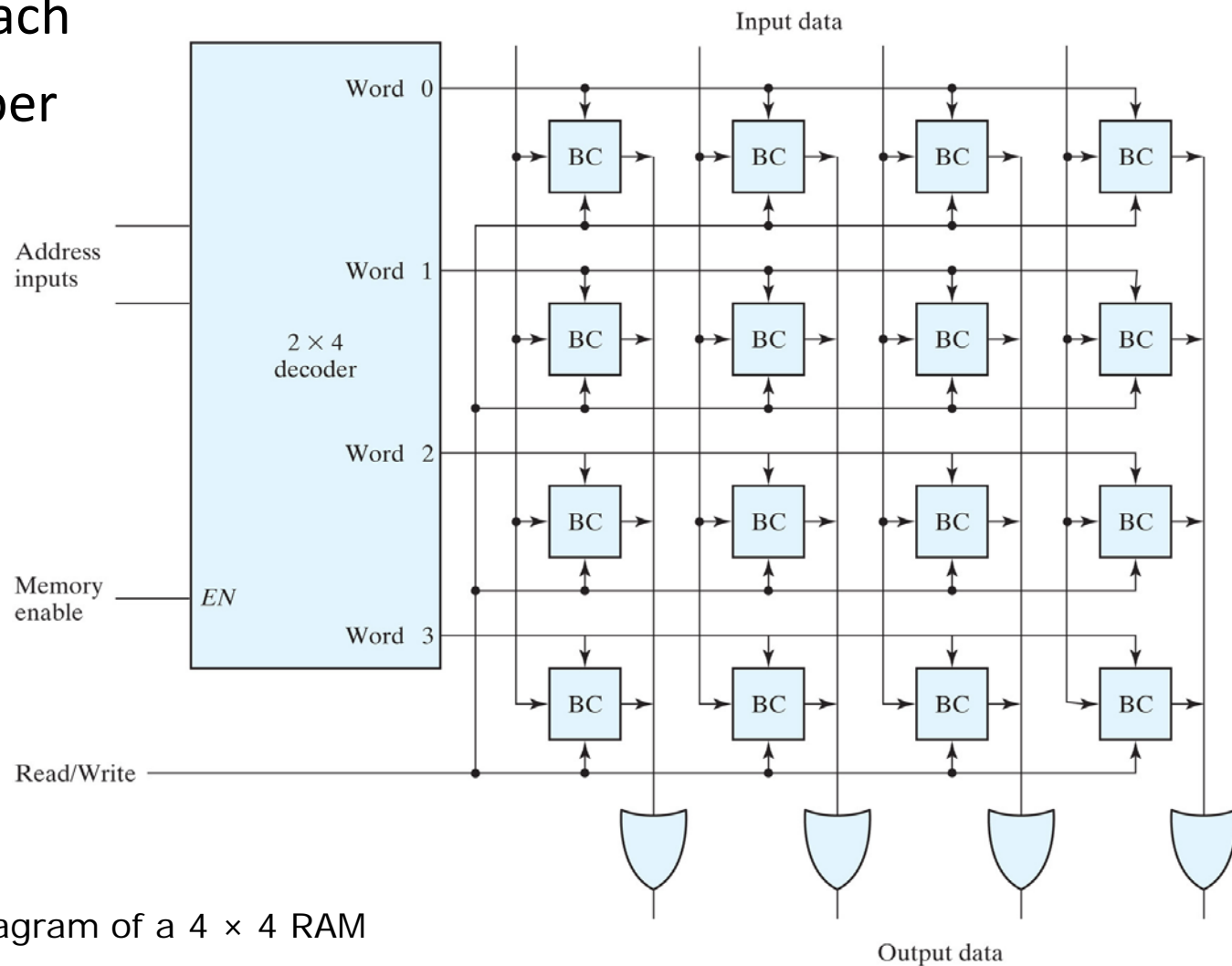- Require k address lines and k x 2^k decoder

FIGURE 7.6 Diagram of a 4 × 4 RAM

Coincident Decording

- Two k/2-input decoders used instead of k-input decoder

- Selected by the coincidence of one X line and one Y line

- Each intersection represents a word

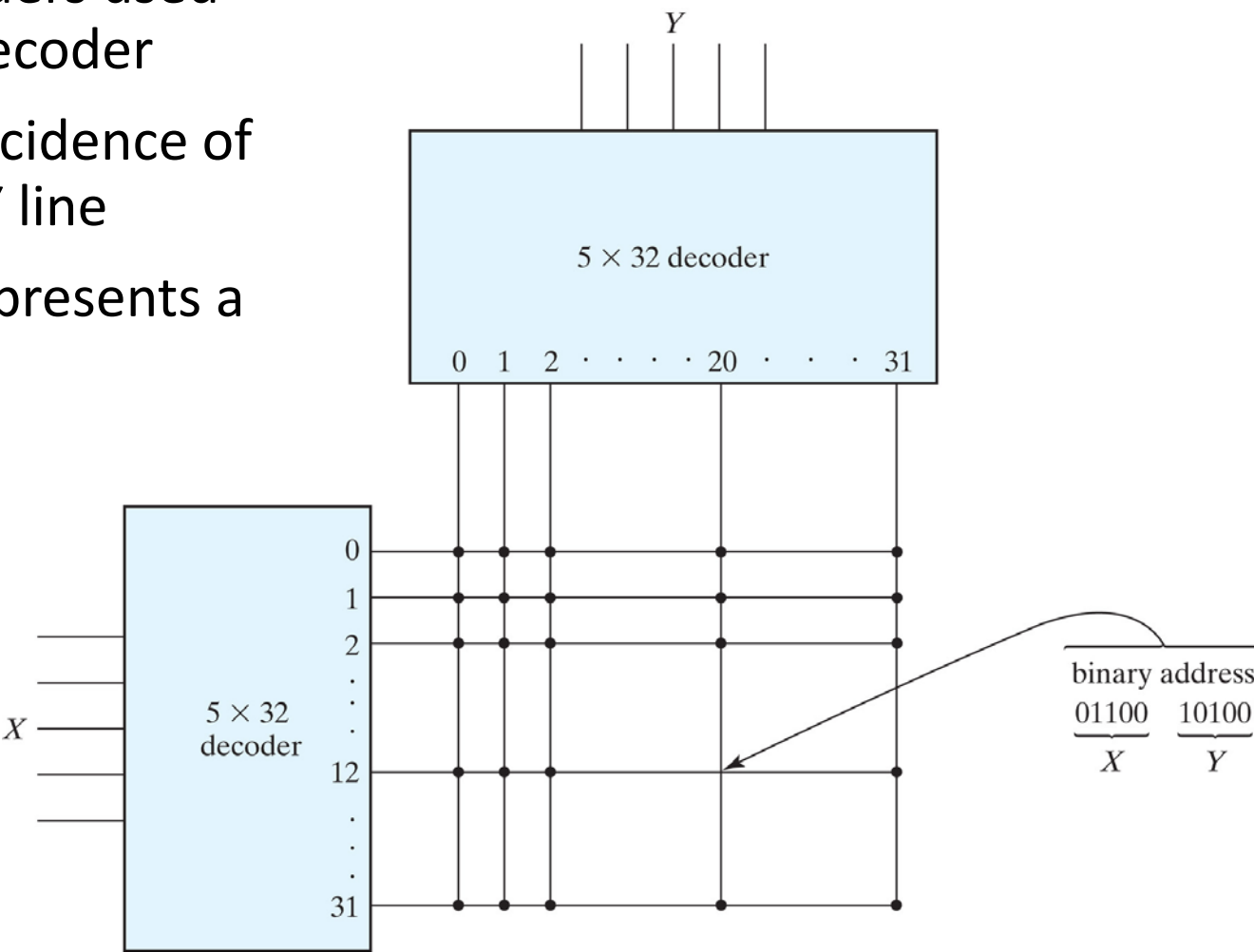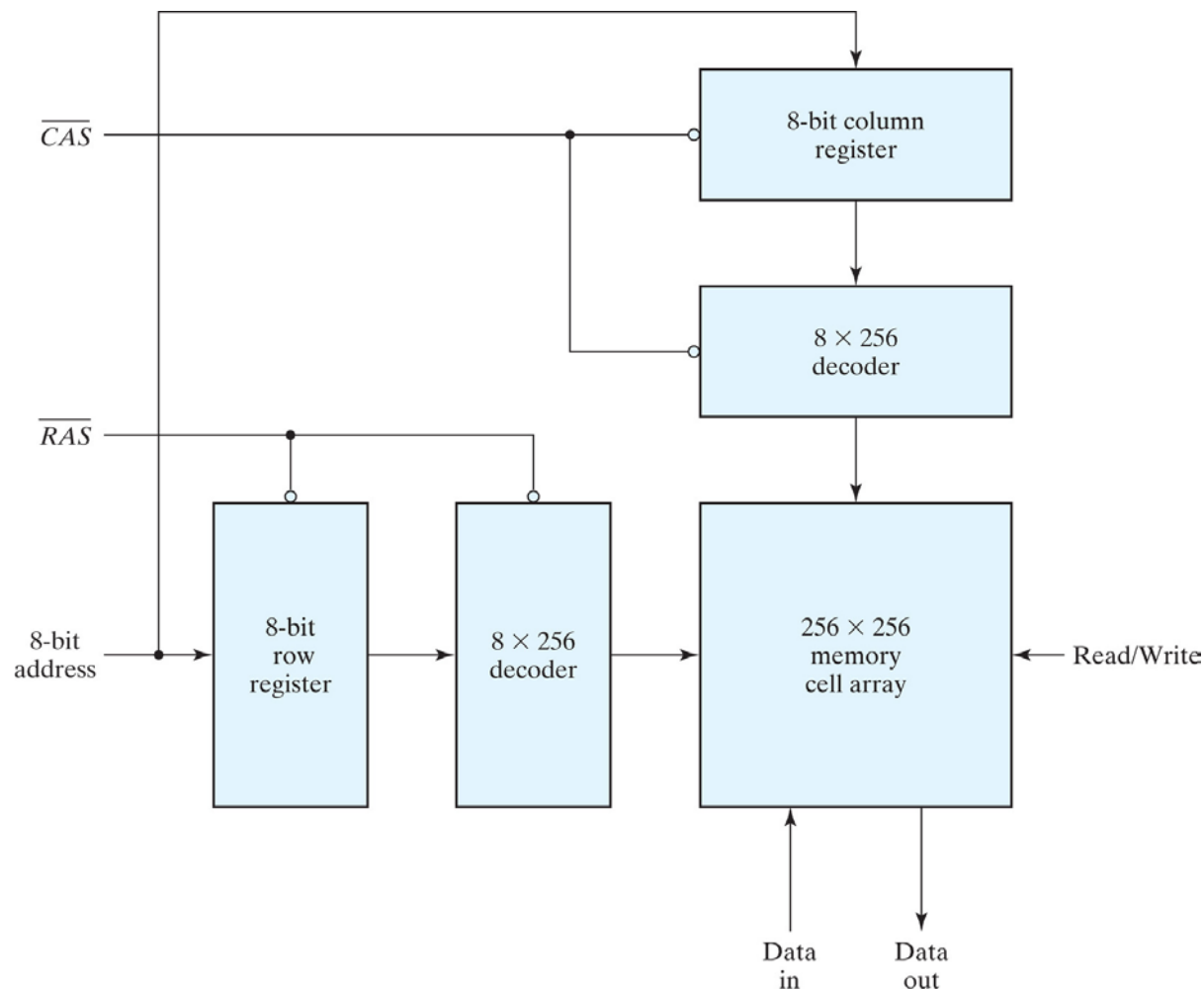FIGURE 7.7 Two-dimensional decoding structure for a 1K-word memory

10

Address Multiplexing

- Large capacity

- Reduce the size

- Row address strobe(RAS)

- Column address strobe(CAS)

FIGURE 7.8 Address multiplexing for a 64K DRAM

7.4 Error Detection and Correction

- Occasional errors in storing and retrieving the binary information
- Employing error-detecting and error-correcting codes
- Parity bit check (detect, but cannot correct)
- Syndrome: generate a unique pattern
- Hamming code
  - Add K parity bits to an n-bit data word
  - Positions numbers as a power of 2 for parity bits

Bit position:	1	2	3	4	5	6	7	8	9	10	11	12
	P_1	P_2	1	P_4	1	0	0	P_8	0	1	0	0

$P_1 = $ XOR of bits $(3, 5, 7, 611) = 1 \oplus 1 \oplus 0 \oplus 0 \oplus 0 = 0$ $C_1 = $ XOR of bits $(1, 3, 5, 7, 9, 11)$

$P_2 = $ XOR of bits $(3, 5, 7, 10, 11) = 1 \oplus 0 \oplus 0 \oplus 1 \oplus 0 = 0$ $C_2 = $ XOR of bits $(2, 3, 6, 7, 10, 11)$

$P_4 = $ XOR of bits $(5, 6, 7, 12) = 1 \oplus 0 \oplus 0 \oplus 0 = 1$ $C_4 = $ XOR of bits $(4, 5, 6, 7, 12)$

$P_8 = $ XOR of bits $(9, 10, 11, 12) = 0 \oplus 1 \oplus 0 \oplus 0 = 1$ $C_8 = $ XOR of bits $(8, 9, 10, 11, 12)$

Bit position:	1	2	3	4	5	6	7	8	9	10	11	12	
	0	0	1	1	1	0	0	1	0	1	0	0	No error
	1	0	1	1	1	0	0	1	0	1	0	0	Error in bit 1
	0	0	1	1	0	0	0	1	0	1	0	0	Error in bit 5

$C_1 = $ XOR of bits $(1, 3, 5, 7, 9, 11)$

$C_2 = $ XOR of bits $(2, 3, 6, 7, 10, 11)$

$C_4 = $ XOR of bits $(4, 5, 6, 7, 12)$

$C_8 = $ XOR of bits $(8, 9, 10, 11, 12)$

	C_8	C_4	C_2	C_1
For no error:	0	0	0	0
With error in bit 1:	0	0	0	1
With error in bit 5:	0	1	0	1

- Give the position of the bit in error

- Uniquely describe a bit in error $(2^k - 1 \geq n + k)$

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

13

Single-Error Correction, Double-Error Detection

- Hamming code: detect and correct only a single error
- By adding another parity bit to the coded word: correct a single error and detect double errors
- Ex 001110010100(P13)

If $C = 0$ and $P = 0$, no error occurred.

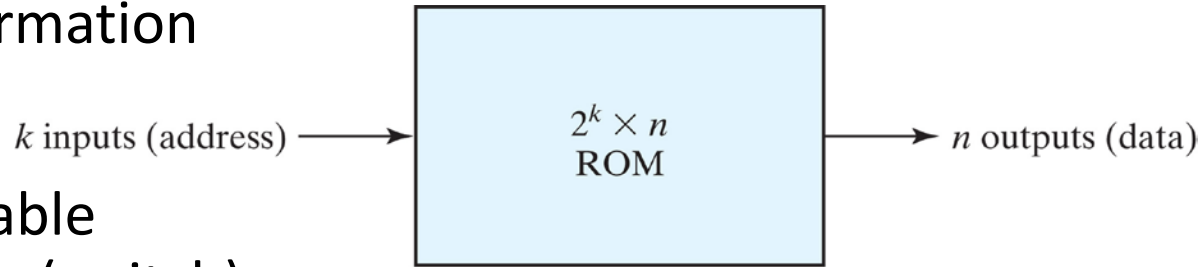If $C \neq 0$ and $P = 1$, a single error occurred that can be corrected.

If $C \neq 0$ and $P = 0$, a double error occurred that is detected, but that cannot be corrected.

If $C = 0$ and $P = 1$, an error occurred in the P_{13} bit.

7.5 Read-Only Memory (ROM)

- Stored permanent binary information

- Programmable Intersection (switch) = crosspoint

- X to denote a temporary connection

- Memory unit

- Combination circuit

FIGURE 7.9   ROM block diagram

k inputs (address) ⟶ $2^k \times n$ ROM ⟶ n outputs (data)

Copyright ©2013 Pearson Education, publishing as Prentice Hall

I_0, I_1, I_2, I_3, I_4 — 5 × 32 decoder — 0, 1, 2, 3, . . . 28, 29, 30, 31
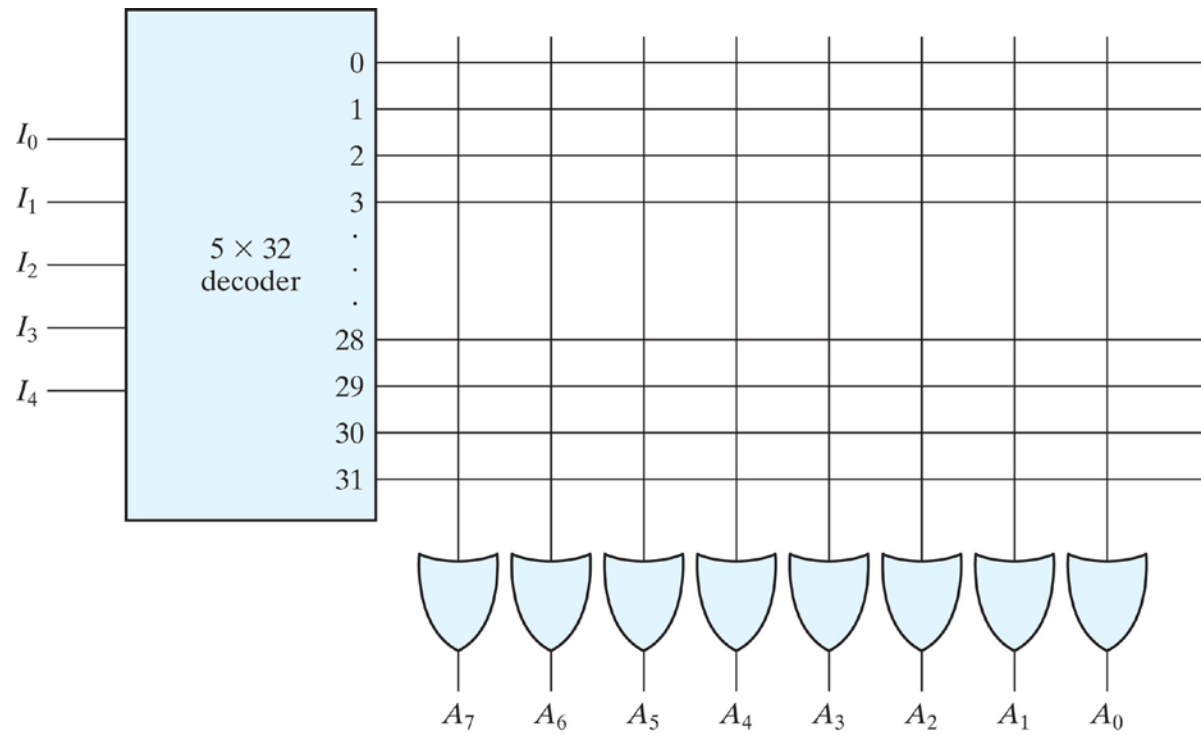
A_7 A_6 A_5 A_4 A_3 A_2 A_1 A_0

FIGURE 7.10   Internal logic of a 32 × 8 ROM

Copyright ©2013 Pearson Education, publishing as Prentice Hall

Digital Design, Kyung Hee Univ.

Table 7.3
ROM Truth Table (Partial)

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
		Inputs						Outputs				
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
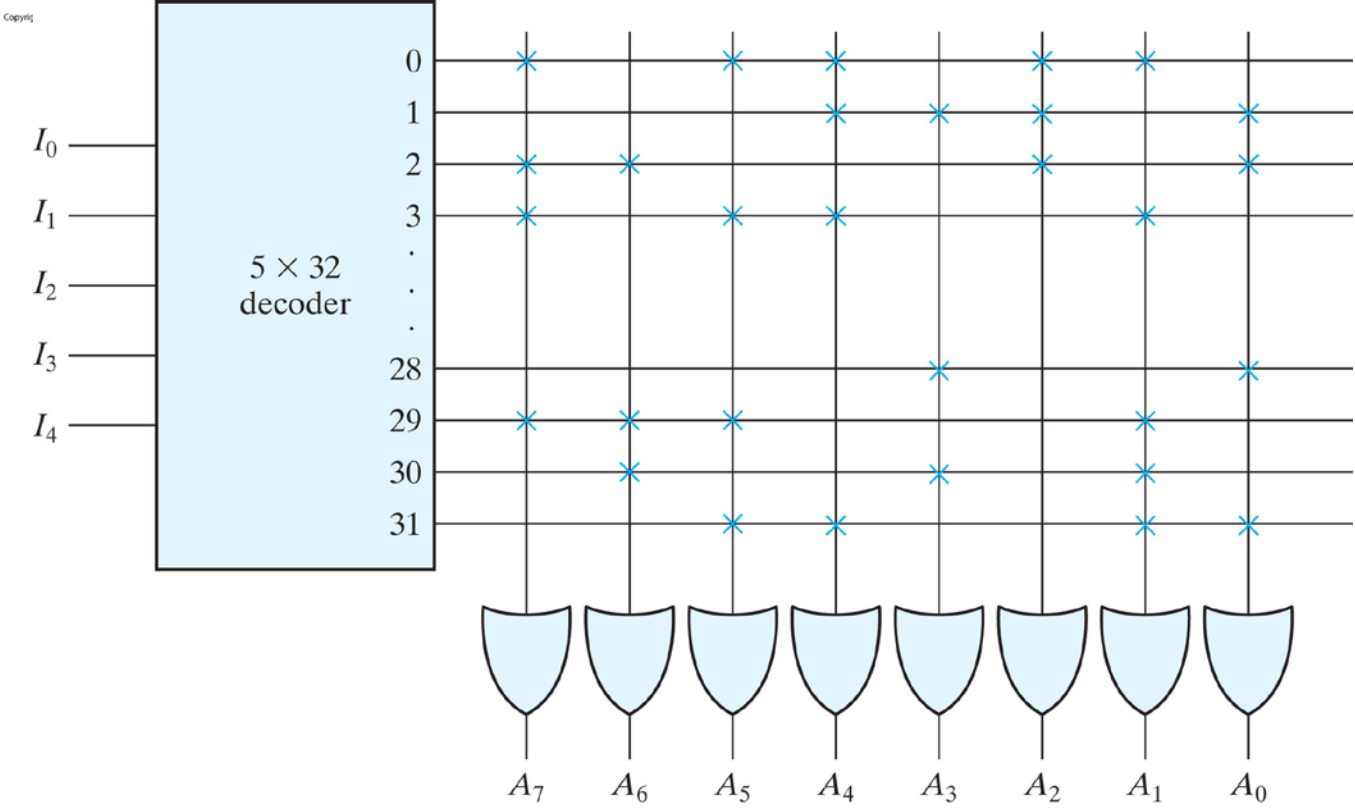
Example 7.1

Table 7.4
Truth Table for Circuit of Example 7.1

Inputs			Outputs						Decimal
A_2	A_1	A_0	B_5	B_4	B_3	B_2	B_1	B_0	
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
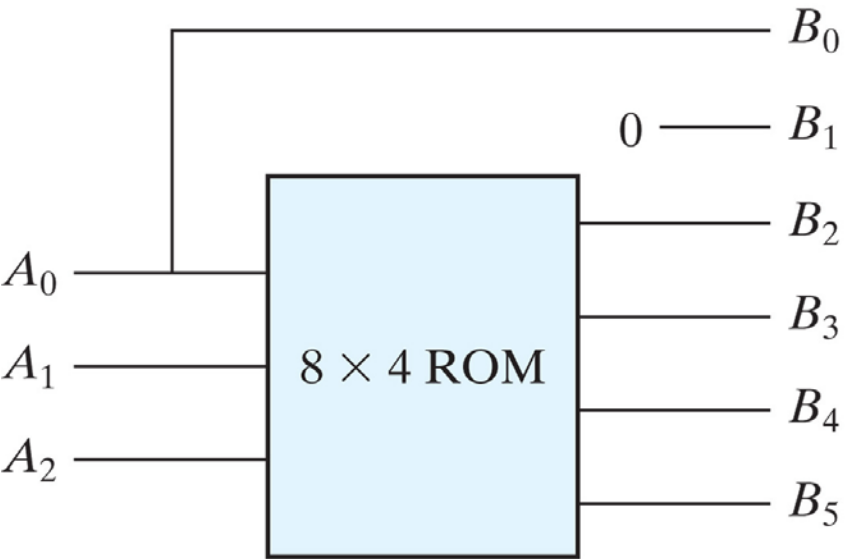
(a) Block diagram

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