Chapter 7

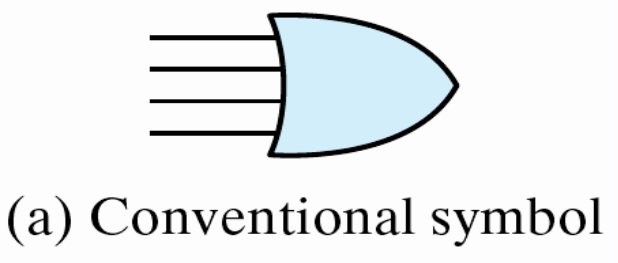
Memory and Programmable Logic

Introduction

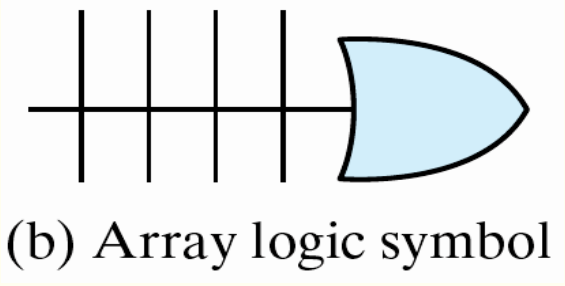
- Memory
  - information storage
  - a collection of cells storing binary information
- RAM – Random Access Memory
  - read operation
  - write operation
- ROM – Read Only Memory
  - read operation only
  - a programmable logic device

Programmable Logic Device (PLD)

- ROM
- PLA – programmable logic array
- PAL – programmable array logic
- FPGA – field-programmable gate array
  - programmable logic blocks
  - programmable interconnects



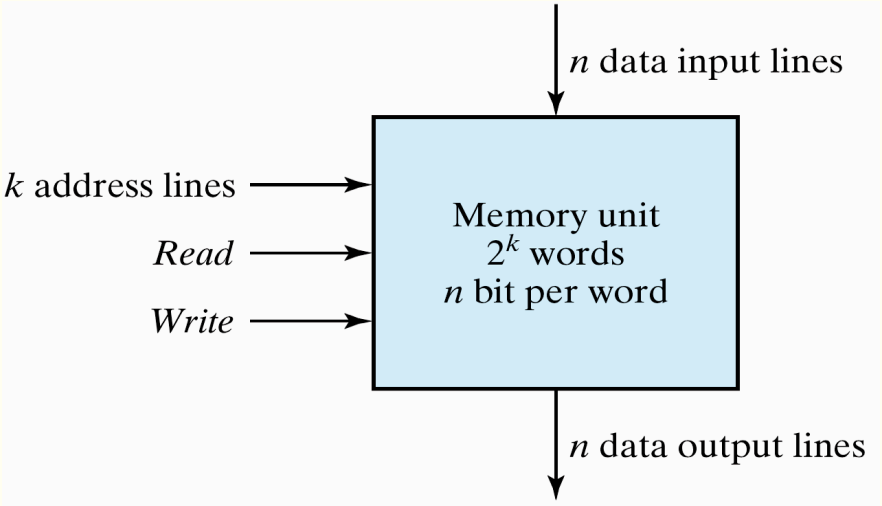
(a) Conventional symbol



(b) Array logic symbol

Random-Access Memory

- A memory unit
  - stores binary information in groups of bits (words)
  - 8 bits (a byte), 2 bytes, 4 bytes
- Block diagram



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

Fig. 7.3 Contents of a 1024 × 16 memory

Write and Read Operations

- Write operation
  - Apply the binary address to the address lines
  - Apply the data bits to the data input lines
  - Activate the *write* input
- Read operation
  - Apply the binary address to the address lines
  - Activate the *read* input

Table 7.1
Control Inputs to Memory Chip

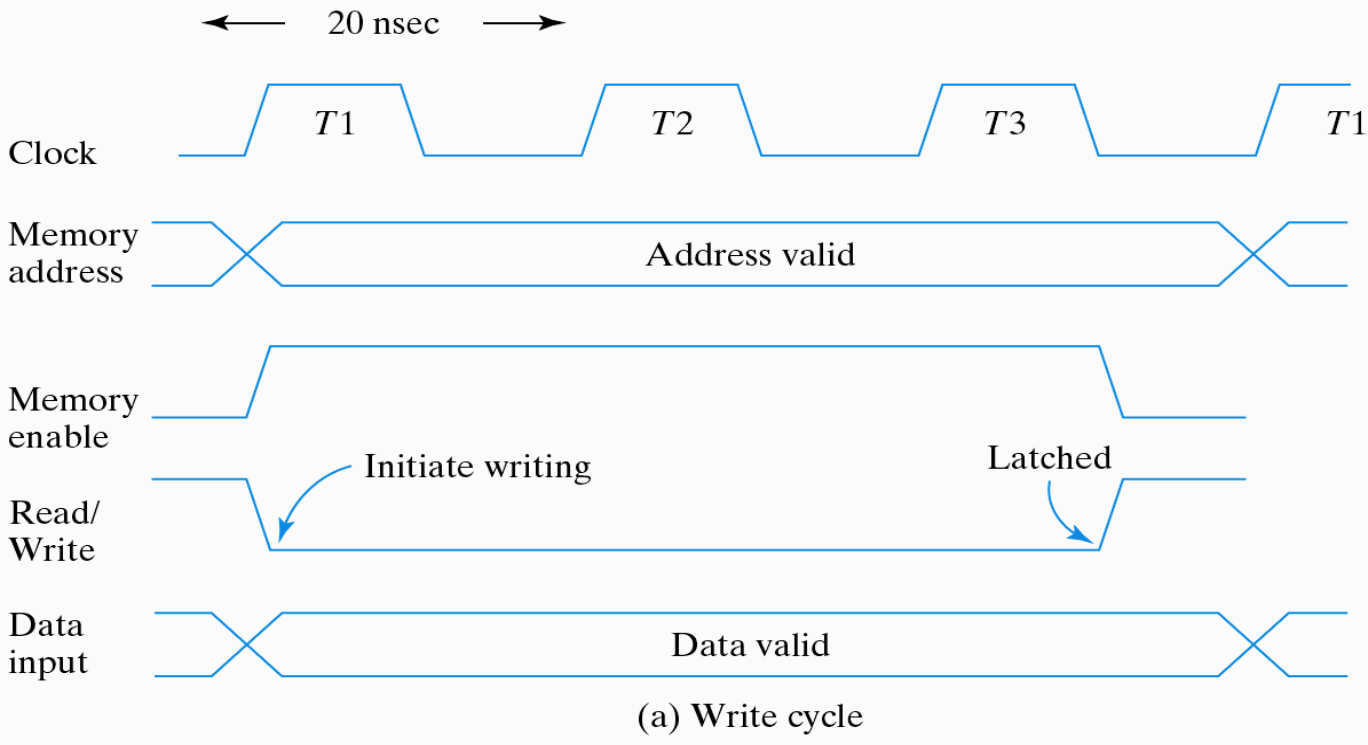
Memory Enable	Read/Write	Memory Operation
0	X	None
1	0	Write to selected word
1	1	Read from selected word

Timing Waveforms

- The operation of the memory unit is controlled by an external device
- The access time
  - the time required to select a word and read it
- The cycle time
  - the time required to complete a write operation
- Read and write operations must synchronized with an external clock

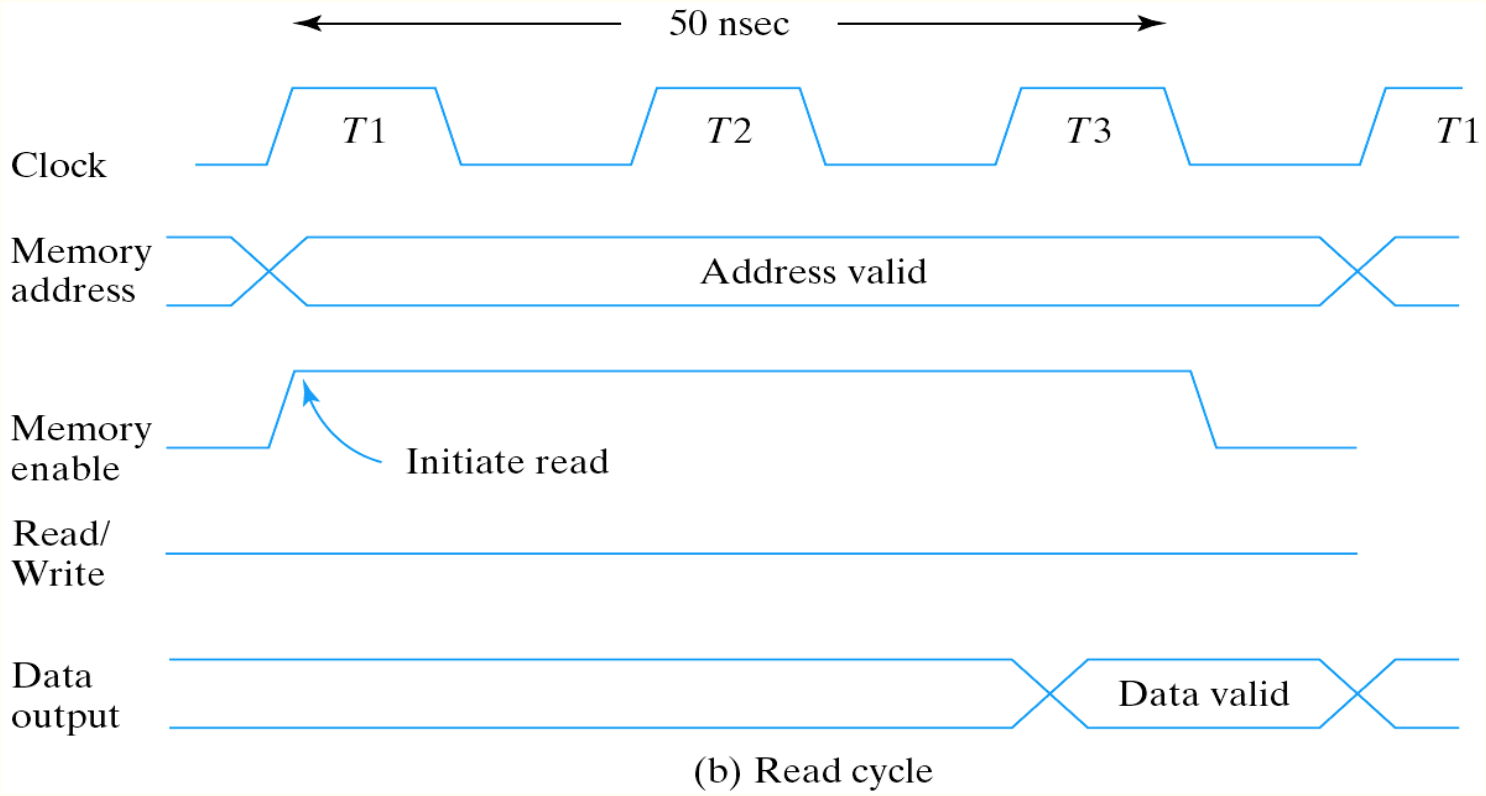
A Write Cycle

- CPU clock – 50 MHz, access/cycle time < 50 ns



(a) Write cycle

A read cycle



(b) Read cycle

Types of RAM

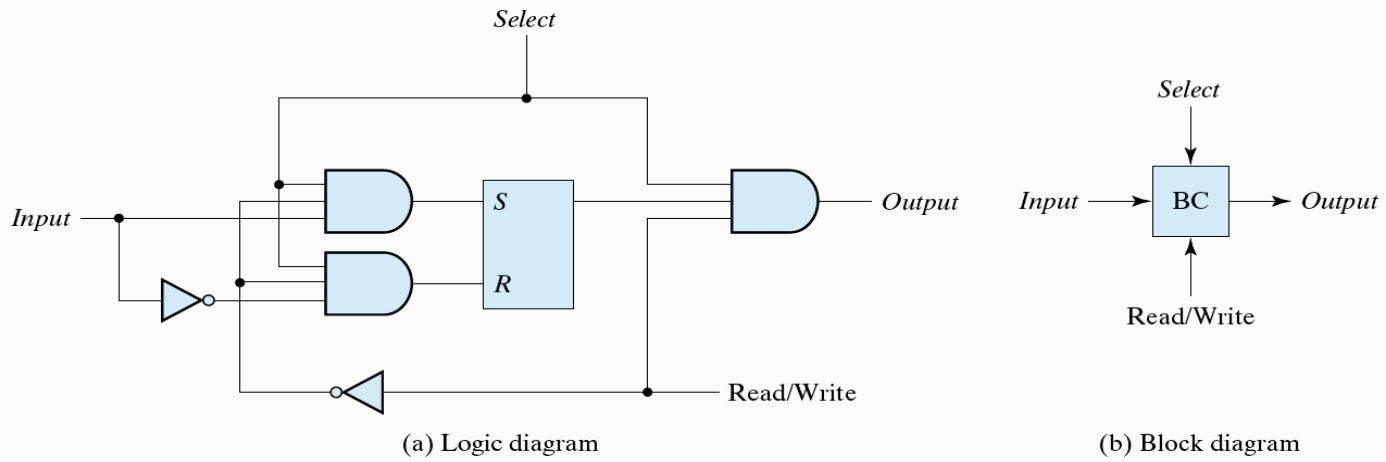
- Static RAM (cache memory)
  - Information are stored in latches
  - remains valid as long as power is applied
  - short read/write cycle
- Dynamic RAM
  - Information are stored in the form of charges on capacitors
  - the stored charge tends to discharge with time
  - need to be refreshed (read and write back)
  - reduced power consumption
  - Larger memory capacity

Memory

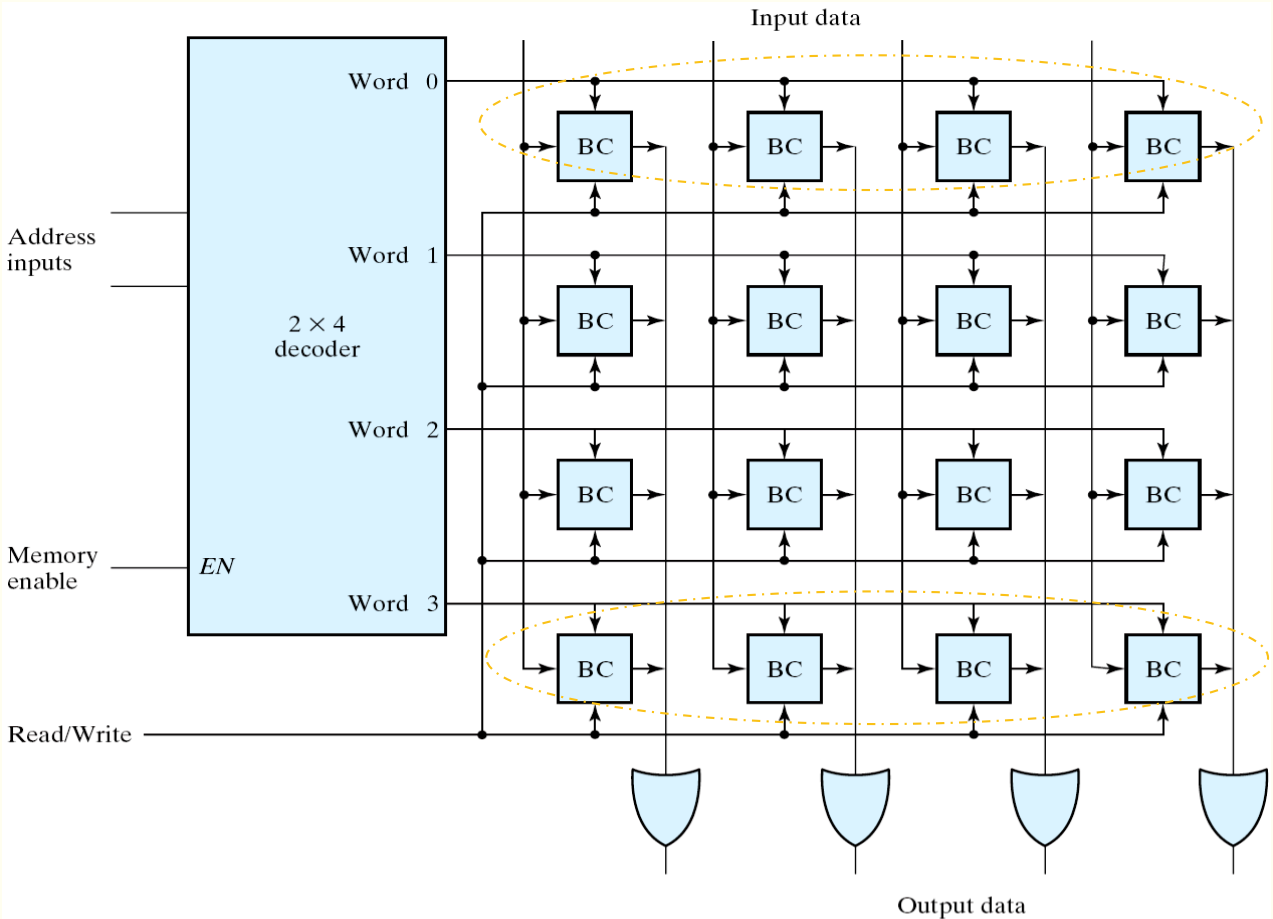
- Volatile
  - lose stored information when power is turned off
  - SRAM, DRAM
- Non-volatile
  - Retains its stored information after the removal of power
  - ROM
  - EPROM, EEPROM
  - Flash memory

Memory Decoding

- A memory unit
  - the storage components
  - the decoding circuits to select the memory word
- A memory cell



A 4 x 4 RAM


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

Coincident Decoding

- A two-dimensional selection scheme
  - reduce the complexity of the decoding circuits

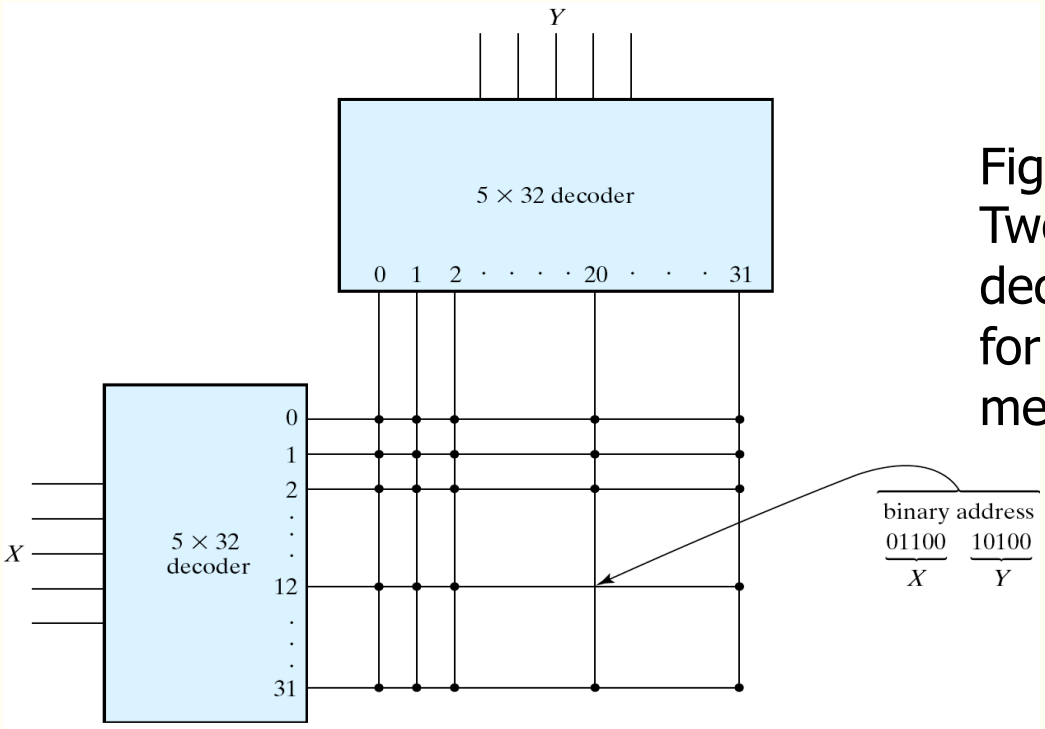


Fig. 7.7
Two-dimensional decoding structure for a 1K-word memory

Error Detection And Correction

- Improve the reliability of a memory unit
- A simple error detection scheme
  - **a parity bit (Sec. 3-9)**
  - **a single bit error can be detected, but cannot be corrected**
- An error-correction code
  - **generates multiple parity check bits**
  - **the check bits generate a unique pattern, called a syndrome**
  - **the specific bit in error can be identified**

Hamming Code

- k parity bits are added to an n-bit data word
  - **$(2^k - 1 \geq n + k)$**
  - **The bit positions are numbered in sequence from 1 to $n + k$**
  - **Those positions numbered as a power of 2 are reserved for the parity bits**
  - **The remaining bits are the data bits**

Hamming Code

- 8-bit data word 11000100
  - **Include 4 parity bits and the 8-bit word ⇒ 12 bits**

    $2^k - 1 \geq n + k,\ n = 8 \Rightarrow k = 4$

    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
  - **Calculate the parity bits: even parity — assumption**

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

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

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

    P_8 = XOR of bits (9, 10, 11, 12) = $0 \oplus 1 \oplus 0 \oplus 0 = 1$
  - **Store the 12-bit composite word in memory.**

    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

Hamming Code

- **When the 12 bits are read from the memory**
  - **Check bits are calculated**
    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)

  - **If no error has occurred**

Bit position:	1	2	3	4	5	6	7	8	9	1011 12
	0	0	1	1	1	0	0	1	0	1	0	0

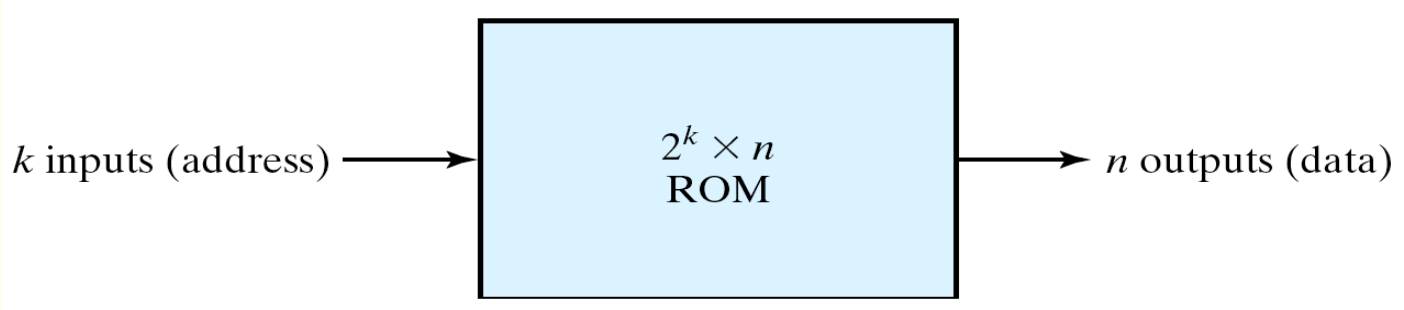
$\Rightarrow C = C_8C_4C_2C_1 = 0000$

Hamming Code

- **One-bit error**
  - **error in bit 1**
    - C_1 = XOR of bits (1, 3, 5, 7, 9, 11) = 1
    - C_2 = XOR of bits (2, 3, 6, 7, 10, 11) = 0
    - C_4 = XOR of bits (4, 5, 6, 7, 12) = 0
    - C_8 = XOR of bits (8, 9, 10, 11, 12) = 0
    - $C_8C_4C_2C_1$ = 0001
  - **error in bit 5**
    - $C_8C_4C_2C_1$ = 0101

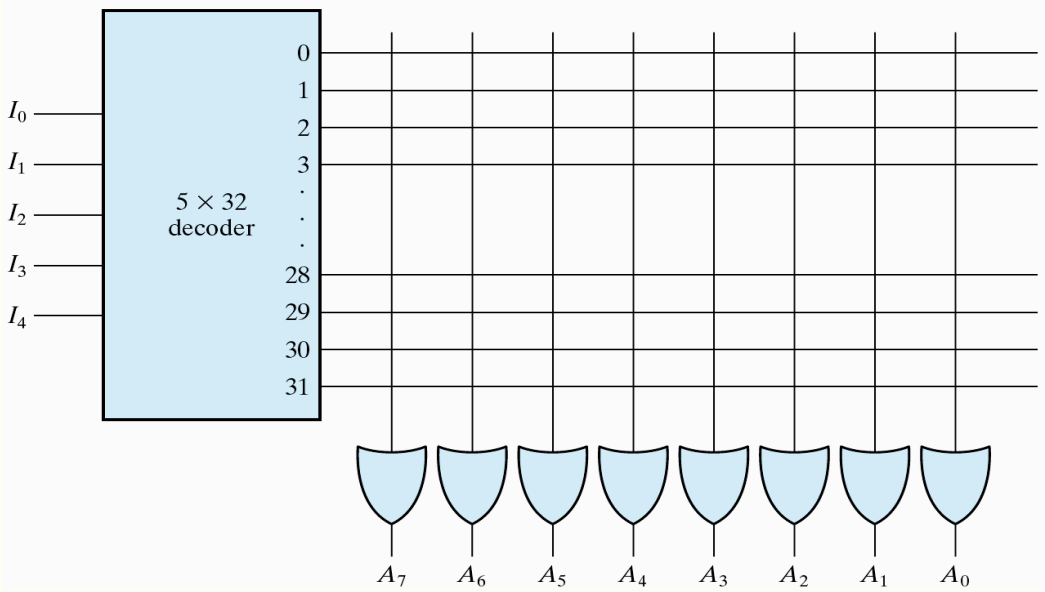
Read-Only Memory

- Store permanent binary information
- 2^k x n ROM
  - **k address input lines**
  - **enable input(s)**
  - **three-state outputs**



32 x 8 ROM

- 5-to-32 decoder
- 8 OR gates
  - each has 32 inputs
- 32x8 internal programmable connections

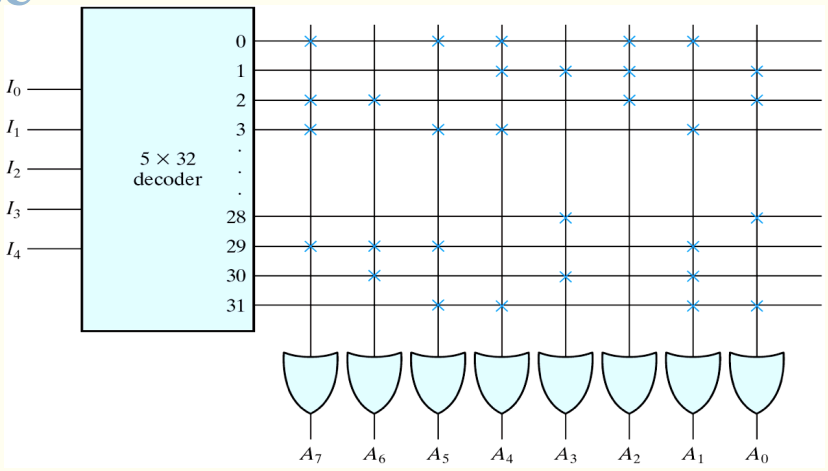


Programmable Interconnections32

- close (two lines are connected)
- or open
- A fuse that can be blown by applying a high voltage pulse

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

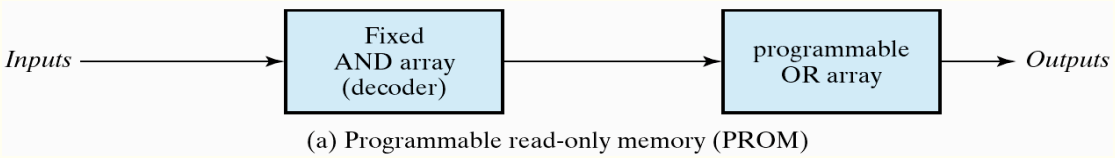


Types of ROM

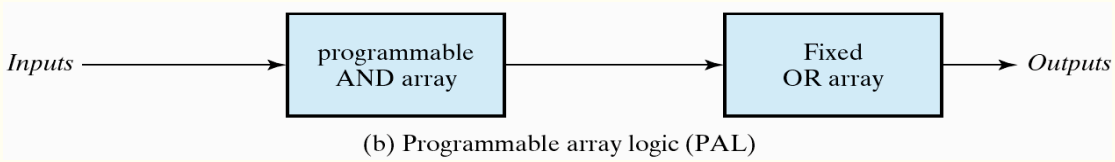
- **Types of ROM**
  - **mask programming ROM**
    - **IC manufacturers**
    - **is economical only if large quantities**
  - **PROM: Programmable ROM**
    - **fuses**
    - **universal programmer**
  - **EPROM: erasable PROM**
    - **floating gate**
    - **ultraviolet light erasable**
  - **EEPROM: electrically erasable PROM (E^2PROM)**
    - **longer time is needed to write**
    - **limited times of write operations**
  - **Flash (like EEPROM)**

Combinational PLDs

- **Programmable two-level logic**
  - **an AND array and an OR array**



(a) Programmable read-only memory (PROM)



(b) Programmable array logic (PAL)

(c) Programmable logic array (PLA)

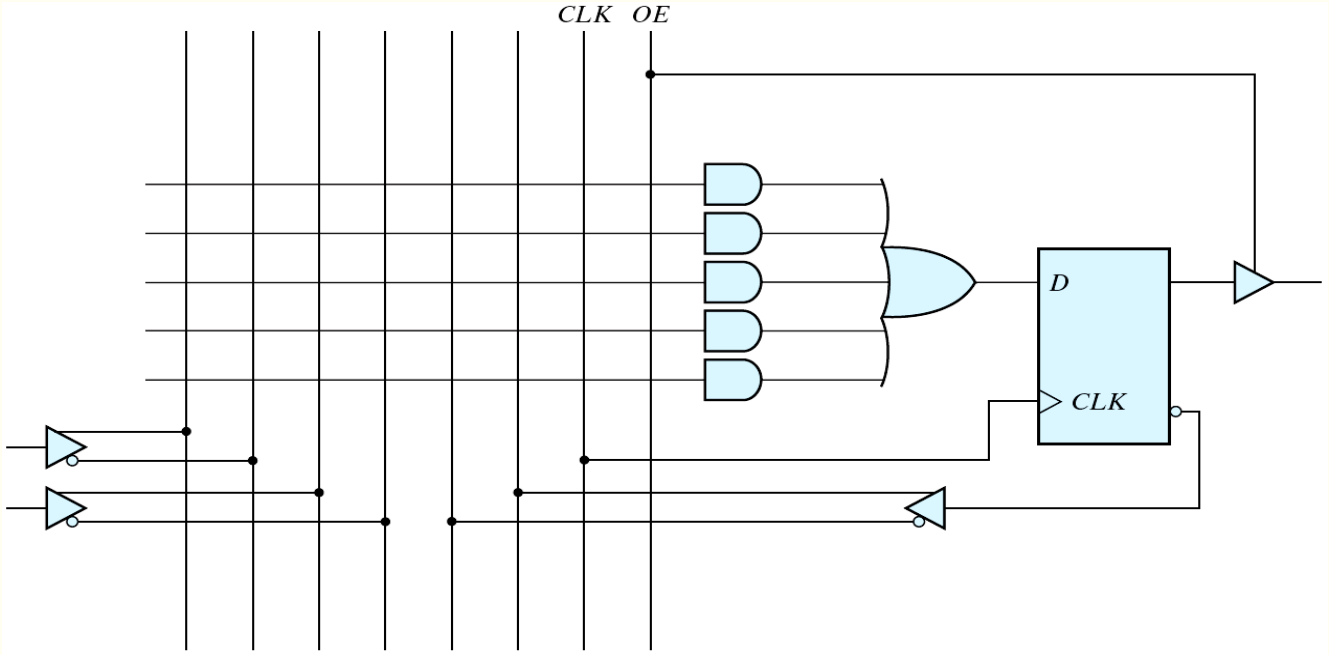
Fig. 7.13
Basic configuration of three PLDs

Sequential PLDs

Three types of sequential PLDs

- Simple programmable logic device (SPLD)
- Complex programmable logic device (CPLD)
- Field programmable gate array (FPGA)

SPLD
A typical SPLD contains 8-10 macrocells



Field-Programmable Gate Arrays

- Logic blocks
  - To implement combinational and sequential logic
- Interconnect
  - Wires to connect inputs and outputs to logic blocks
- I/O blocks
  - Special logic blocks at periphery of device for external connections
- Key questions:
  - How to make logic blocks programmable?
  - How to connect the wires?
  - *After the chip has been fabbed*