



Vietnam National University HCMC  
International University

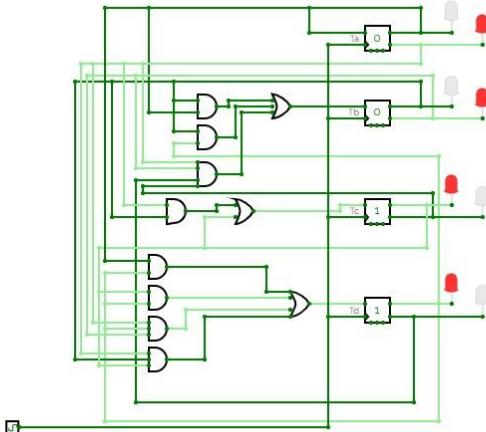


School of  
Electrical Engineering

EE053IU

Digital Logic Design

Lecture 11: Data Storage



INSTRUCTOR: Dr. Vuong Quoc Bao

# 1. Semiconductor Memory Basics

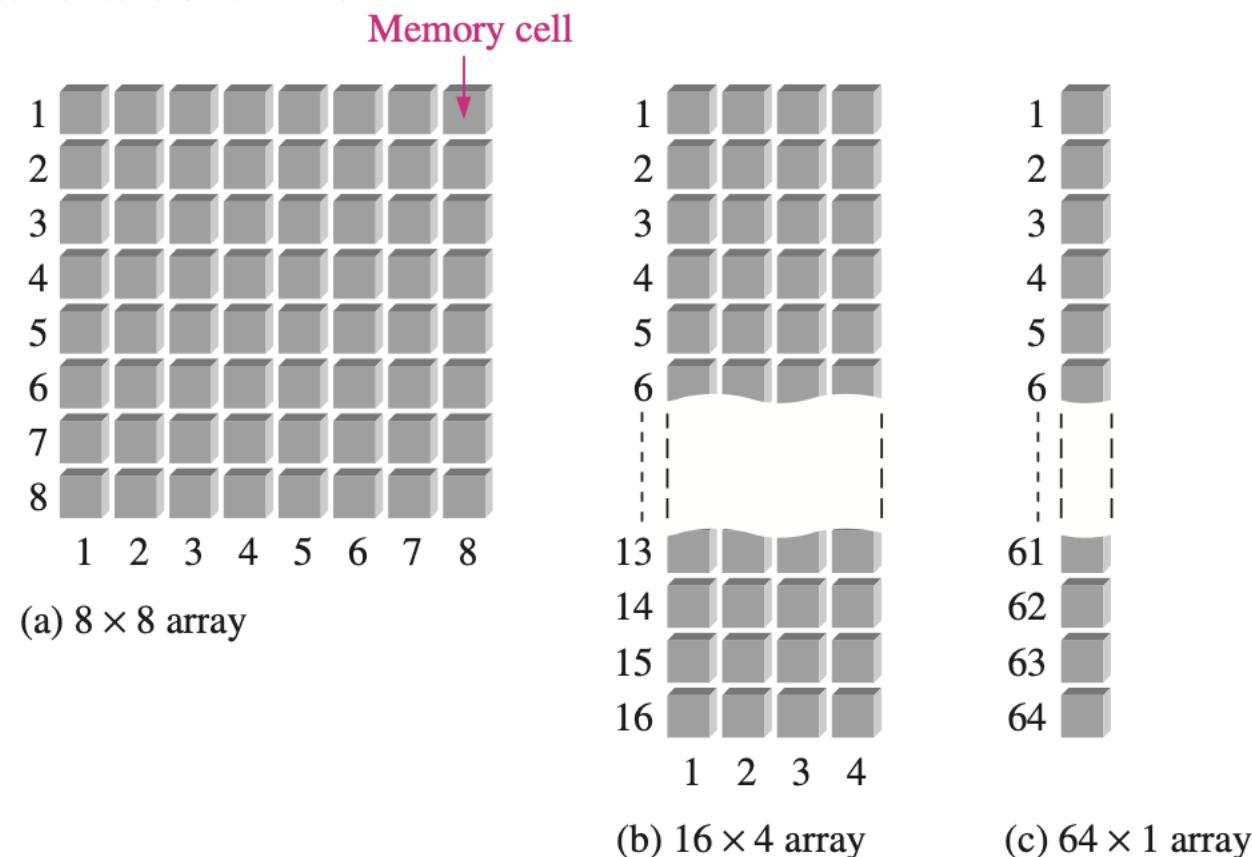
- Memory is the portion of a computer or other system that stores binary data. In a computer, memory is accessed millions of times per second.

## Units of Binary Data: Bits, Bytes, Nibbles, and Words

- As a rule, memories store data in units that have from one to eight bits. The smallest unit of binary data, as you know, is the bit.
- In many applications, data are handled in an 8-bit unit called a byte or in multiples of 8-bit units. The byte can be split into two 4-bit units that are called nibbles. Bytes can also be grouped into words.
- The term word can have two meanings in computer terminology. In memories, it is defined as a group of bits or bytes that acts as a single entity that can be stored in one memory location. In assembly language, a word is specifically defined as two bytes.

# The Basic Memory Array

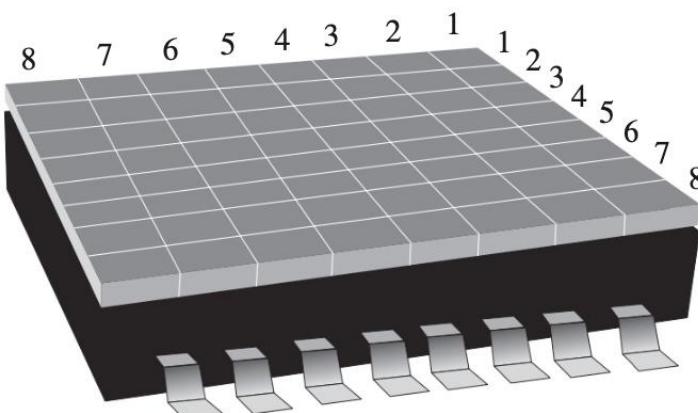
- Each storage element in a memory can retain either a 1 or a 0 and is called a cell.
- Each block in the memory array represents one storage cell, and its location can be identified by specifying a row and a column.



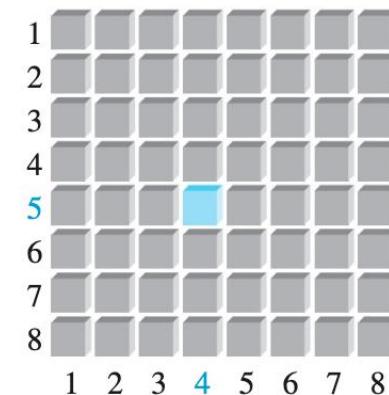
**FIGURE 11-1** A 64-cell memory array organized in three different ways.

# Memory Address and Capacity

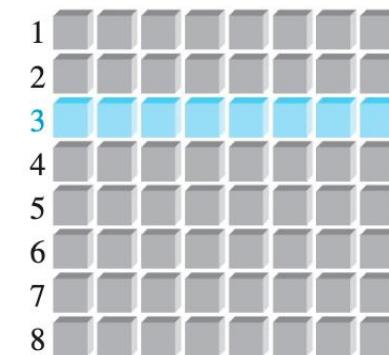
- The location of a unit of data in a memory array is called its address.
- Personal computers have random-access memories organized in bytes. This means that the smallest group of bits that can be addressed is eight.
- The capacity of a memory is the total number of data units that can be stored.



(a) Physical structure of 64-bit memory.

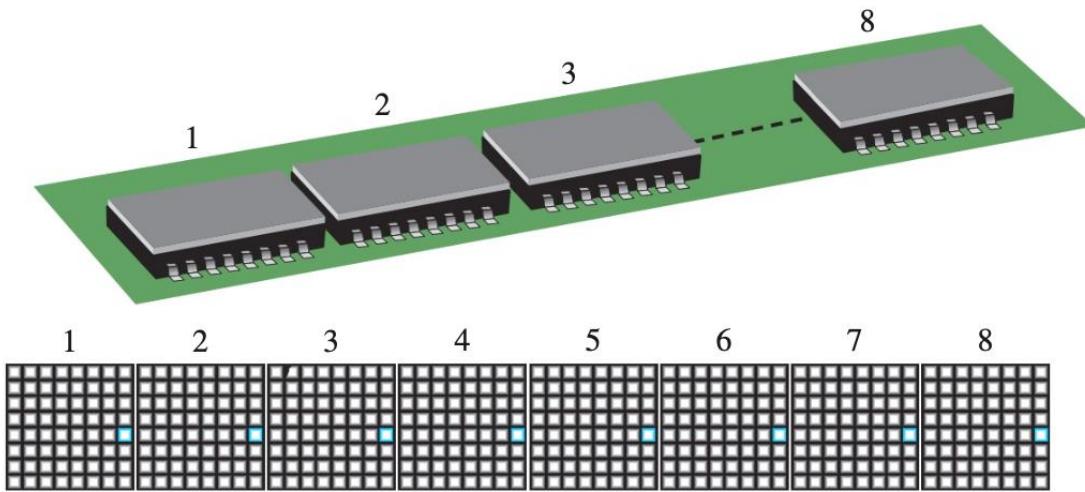


(b) The address of the blue bit  
is row 5, column 4.

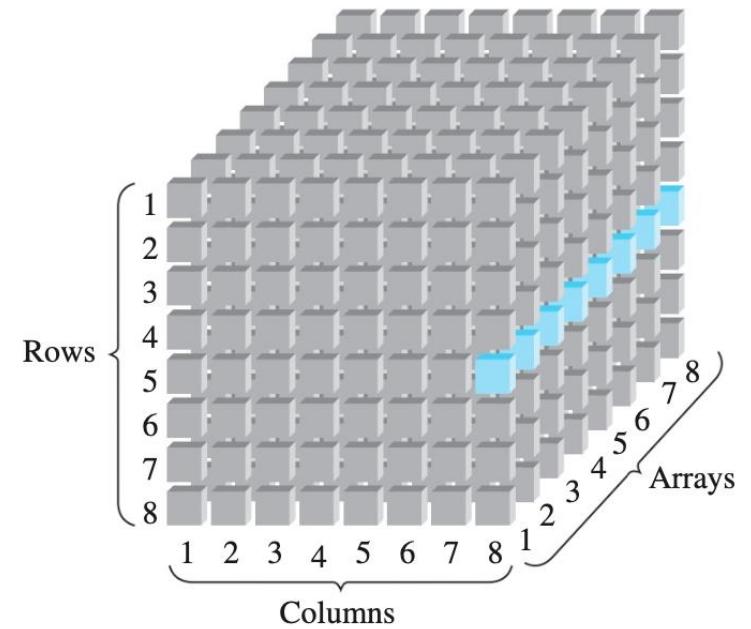


(c) The address of the blue byte  
is row 3.

**FIGURE 11-2** Examples of memory address in a 2-dimensional memory array.



(a) The  $8 \times 8$  bit array expanded to a  $64 \times 8$  bit array. This array forms a memory module.

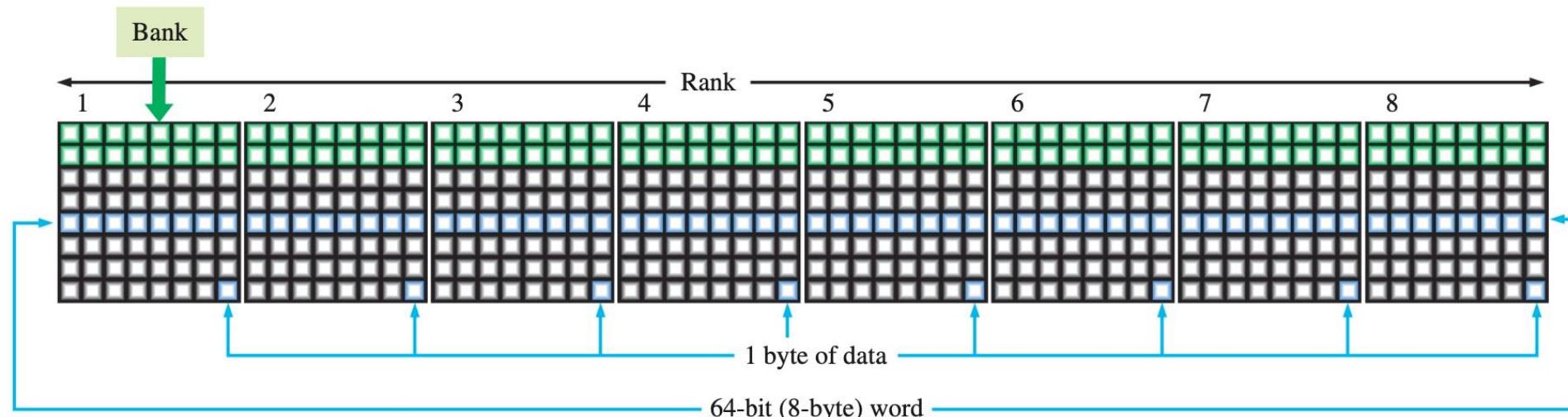


(b) The address of the blue byte is row 5, column 8.

**FIGURE 11-3** Example of memory address in an expanded (multiple) array.

## Memory Banks And Ranks

- A **bank** is a section of memory within a single memory array (chip). A memory chip may have one or more banks. Memory banks can be used for storing frequently used information. Easier and faster access can be achieved by knowing the section of memory in which the data are stored.
- A **rank** is a group of chips that make up a memory module that stores data in units such as words or bytes.

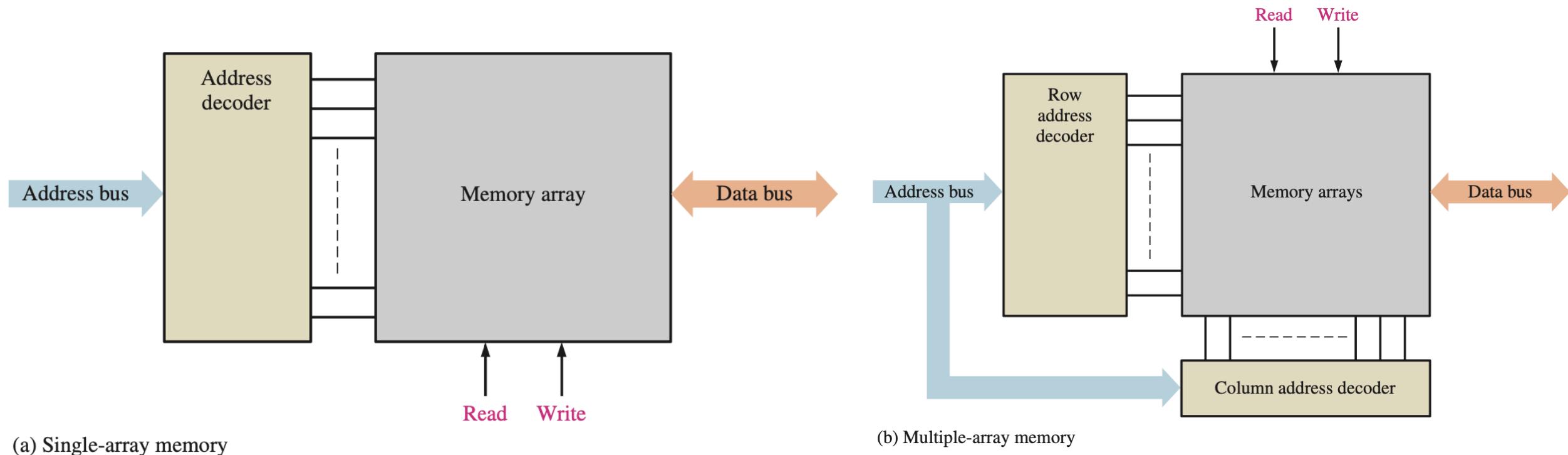


**FIGURE 11-4** Simple illustration of memory bank and memory rank.

# Basic Memory Operations

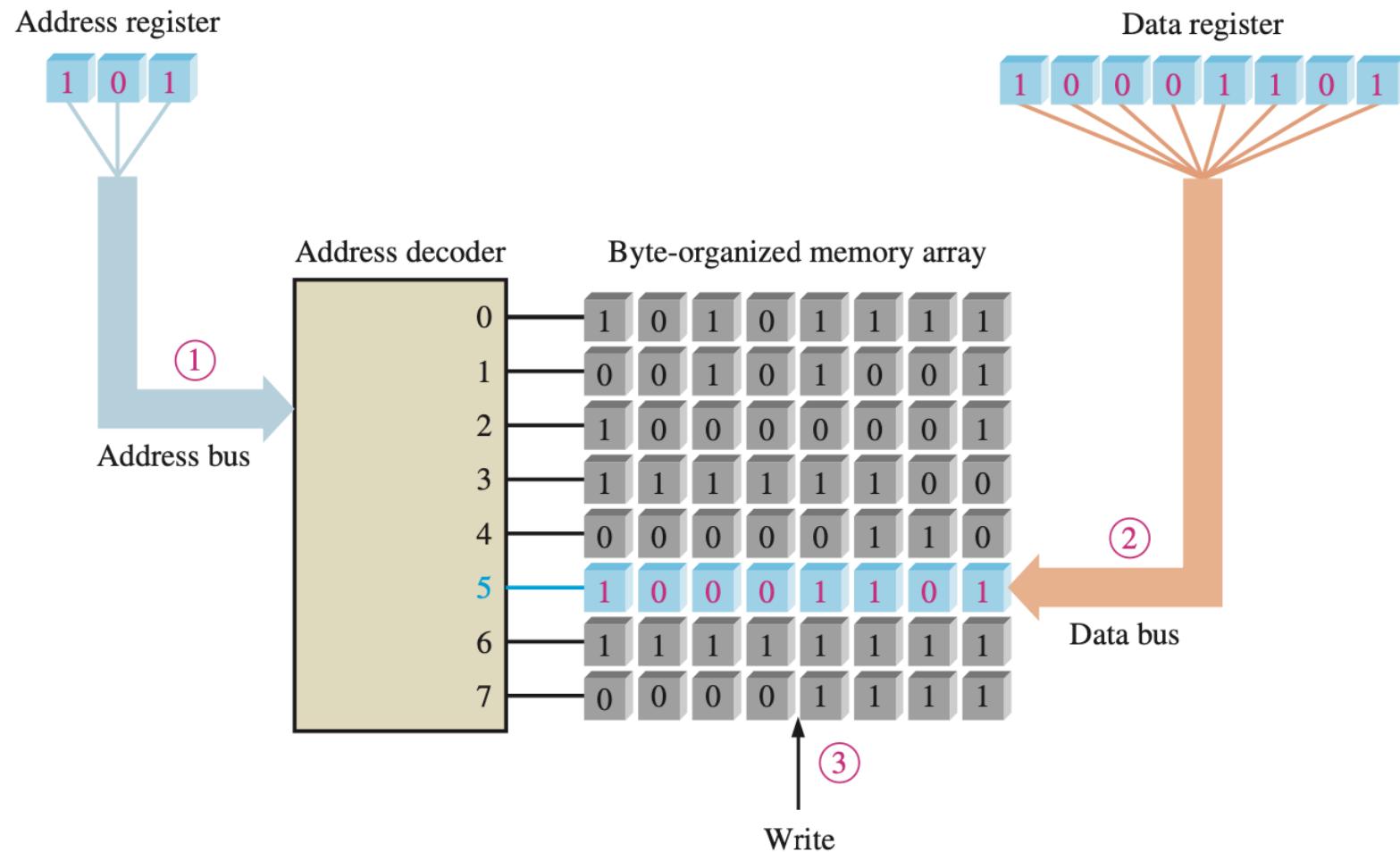
- Addressing is the process of accessing a specified location in memory. Since a memory stores binary data, data must be put into the memory and data must be copied from the memory when needed.
- *The write* operation puts data into a specified address in the memory, and *the read* operation copies data out of a specified address in the memory.
- The addressing operation, which is part of both the write and the read operations, selects the specified memory address.
- Data units go into the memory during a write operation and come out of the memory during a read operation on a set of lines called the *data bus*.

# The Write Operation



**FIGURE 11–5** Block diagram of a single-array memory and a multiple-array memory showing address bus, address decoder(s), bidirectional data bus, and read/write inputs.

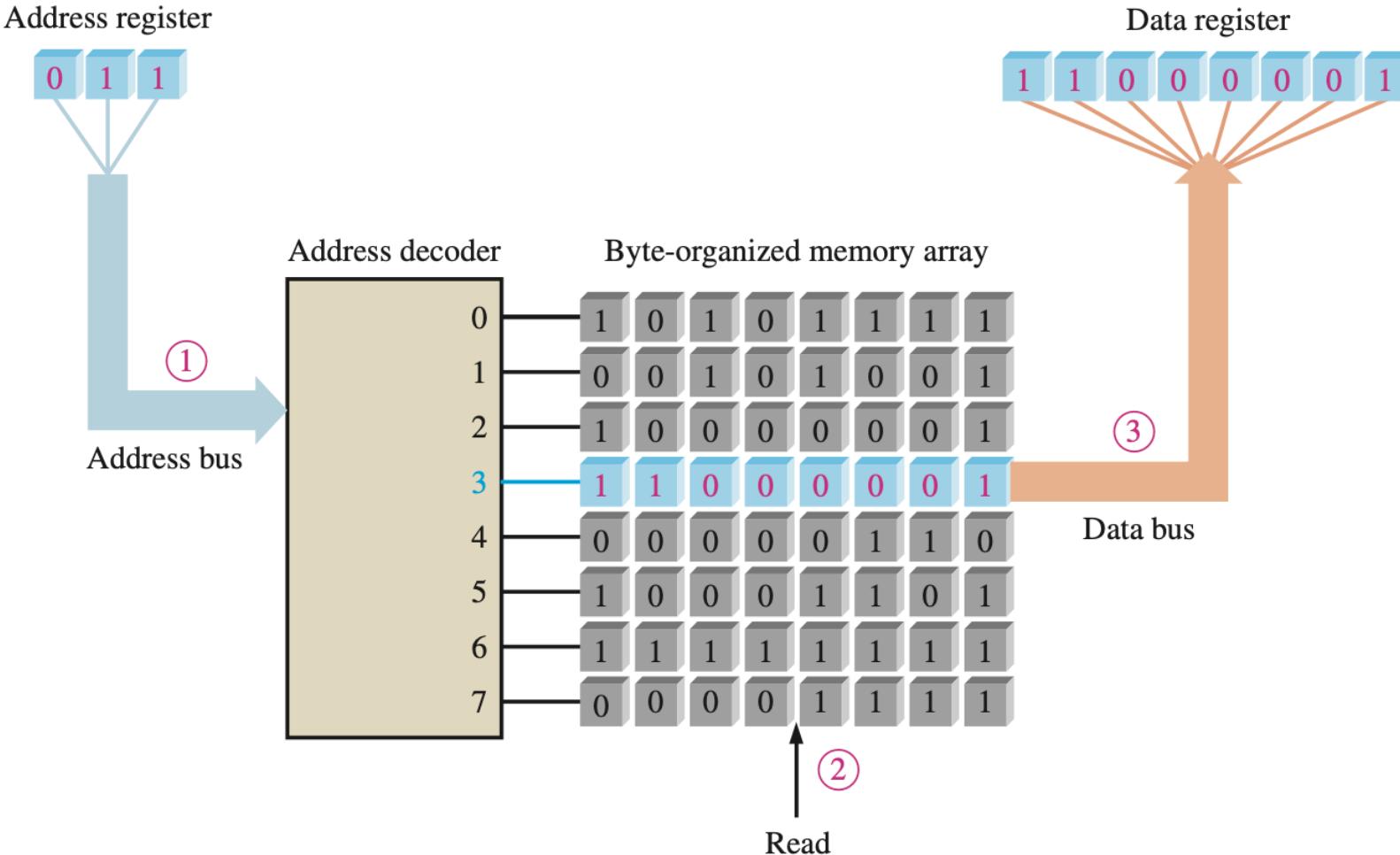
# The Write Operation



- ① Address code 101 is placed on the address bus and address 5 is selected.
- ② Data byte is placed on the data bus.
- ③ Write command causes the data byte to be stored in address 5, replacing previous data.

**FIGURE 11-6** Illustration of the write operation.

# The Read Operation



- ① Address code 011 is placed on the address bus and address 3 is selected.
- ② Read command is applied.
- ③ The contents of address 3 is placed on the data bus and shifted into data register.  
The contents of address 3 is not erased by the read operation.

**FIGURE 11-7** Illustration of the read operation.

## RAMs and ROMs

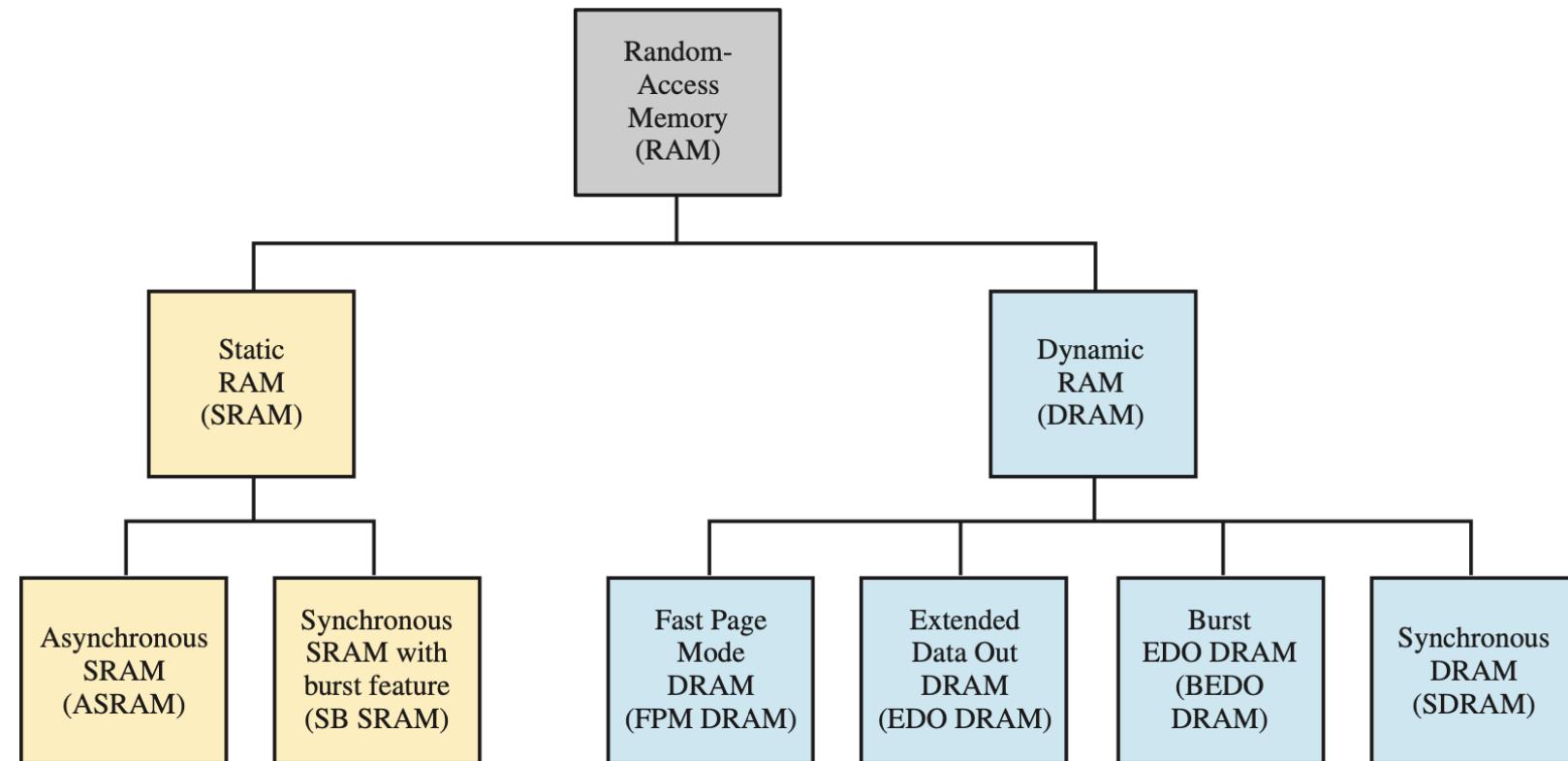
- The two major categories of semiconductor memories are the RAM and the ROM. **RAM** (*random-access memory*) is a type of memory in which all addresses are accessible in an equal amount of time and can be selected in any order for a read or write operation. All RAMs have both read and write capability. Because RAMs lose stored data when the power is turned off, they are volatile memories.
- **ROM (read-only memory)** is a type of memory in which data are stored permanently or semipermanently. Data can be read from a ROM, but there is no write operation as in the RAM. The ROM, like the RAM, is a random-access memory but the term RAM traditionally means a random-access read/write memory. Because ROMs retain stored data even if power is turned off, they are nonvolatile memories.

## 2. The Random-Access Memory (RAM)

### The RAM Family

- The two major categories of RAM are the static RAM (SRAM) and the dynamic RAM (DRAM).
- *SRAMs* generally use latches as storage elements and can therefore store data indefinitely as long as dc power is applied.
- *DRAMs* use capacitors as storage elements and cannot retain data very long without the capacitors being recharged by a process called refreshing.
- Both SRAMs and DRAMs will lose stored data when dc power is removed and, therefore, are classified as volatile memories.

# The RAM Family

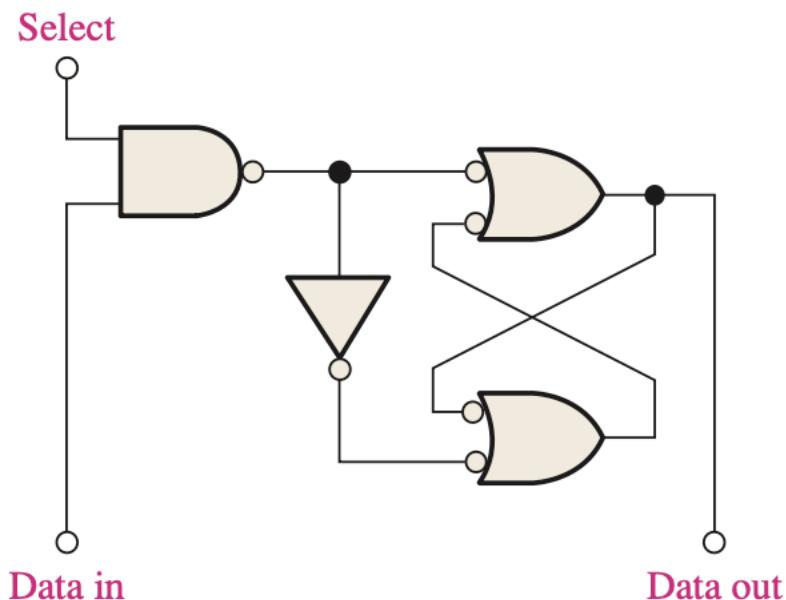


**FIGURE 11–8** The RAM family.

# Static RAMs (SRAMs)

## Memory Cell

- All SRAMs are characterized by latch memory cells. As long as dc power is applied to a static memory cell, it can retain a 1 or 0 state indefinitely. If power is removed, the stored data bit is lost.



**FIGURE 11–9** A typical SRAM latch memory cell.

# State Memory Cell Array

- The memory cells in a SRAM are organized in rows and columns. All the cells in a row share the same Row Select line.
- Each set of Data in and Data out lines go to each cell in a given column and are connected to a single data line that serves as both an input and output (Data I/O) through the data input and data output buffers.

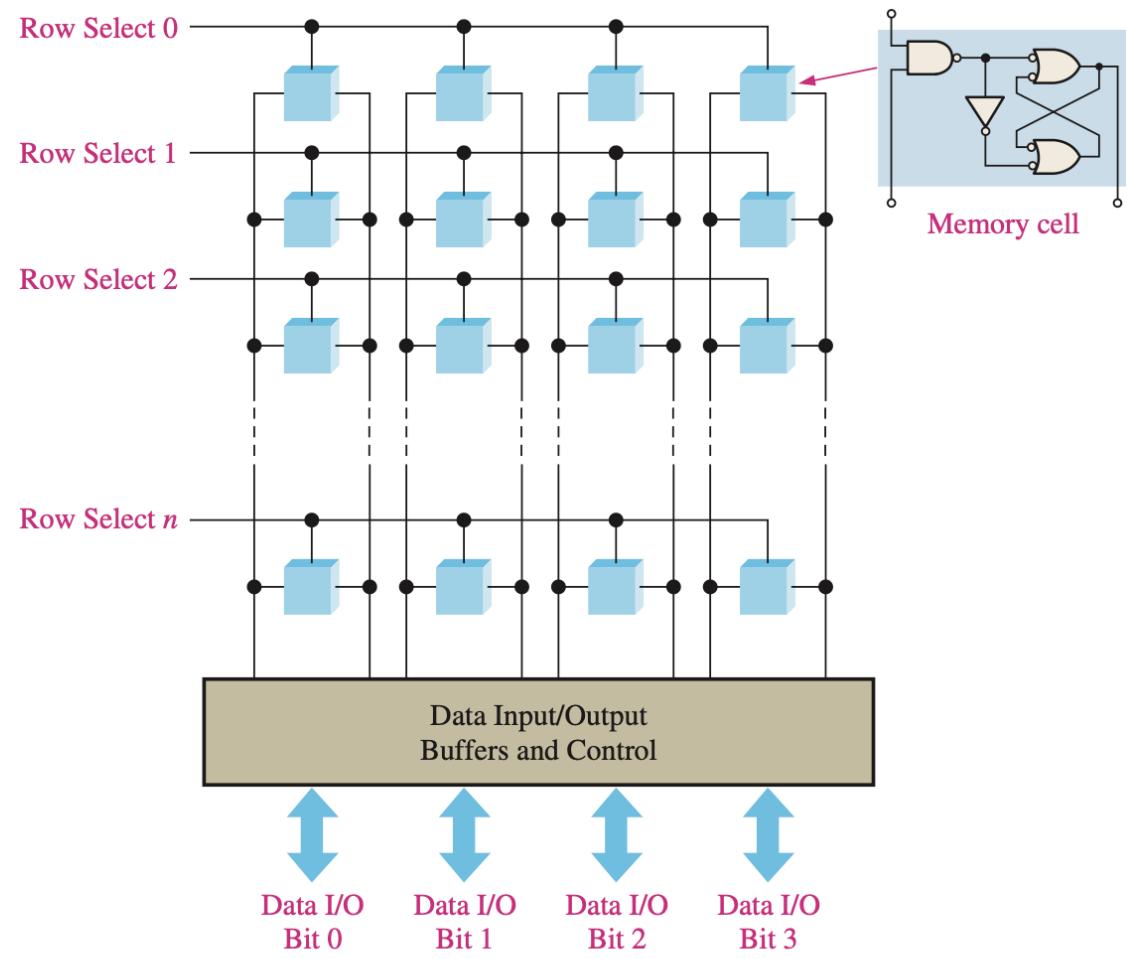
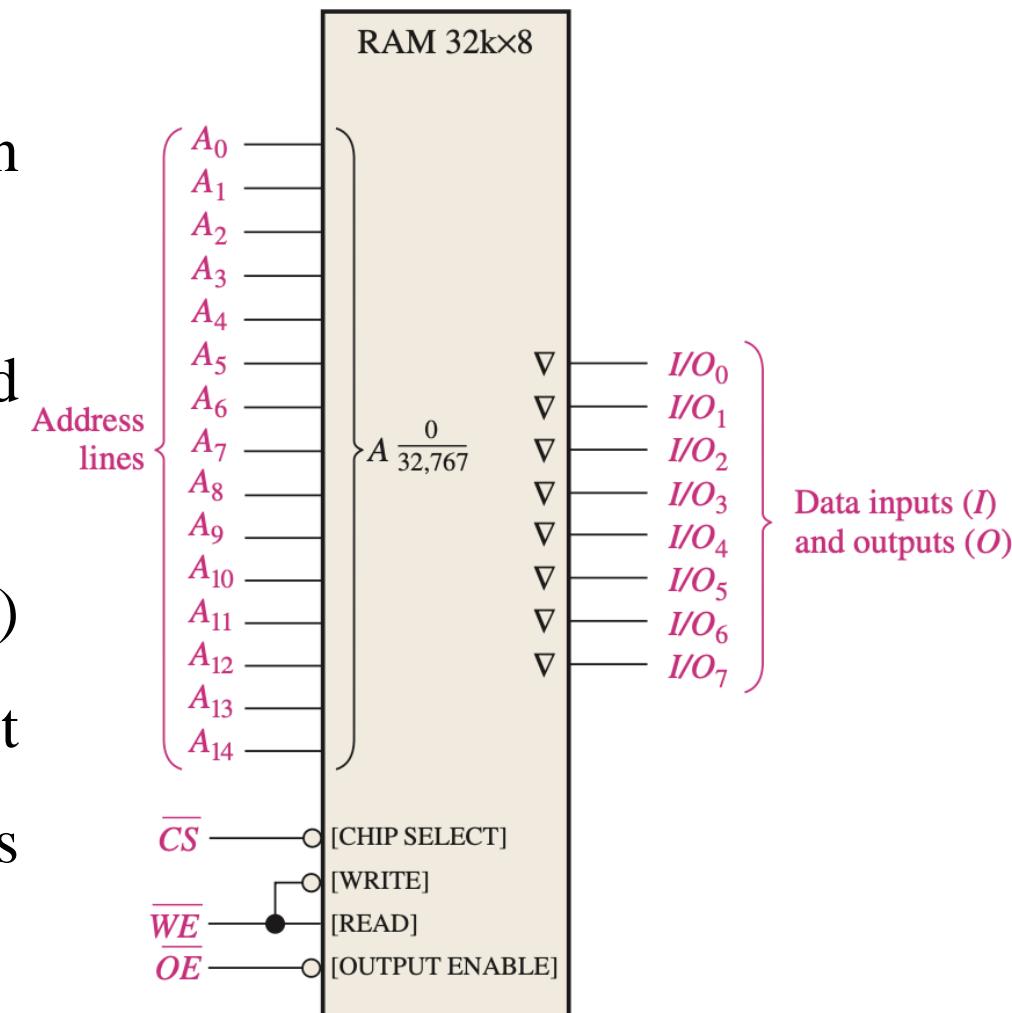


FIGURE 11-10 Basic SRAM array.

# Basic Asynchronous SRAM Organization

- An asynchronous SRAM is one in which the operation is not synchronized with a system clock.
- In the READ mode, the eight data bits that are stored in a selected address appear on the data output lines.
- In the WRITE mode, the eight data bits that are applied to the data input lines are stored at a selected address.
- The data input and data output lines ( $I/O_0$  through  $I/O_7$ ) share the same lines. During READ, they act as output lines ( $O_0$  through  $O_7$ ) and during WRITE they act as input lines ( $I_0$  through  $I_7$ ).

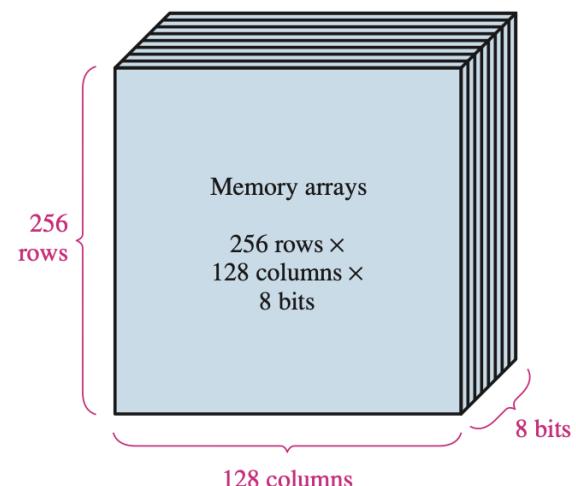


**FIGURE 11-11** Logic diagram for an asynchronous  $32k \times 8$  SRAM.

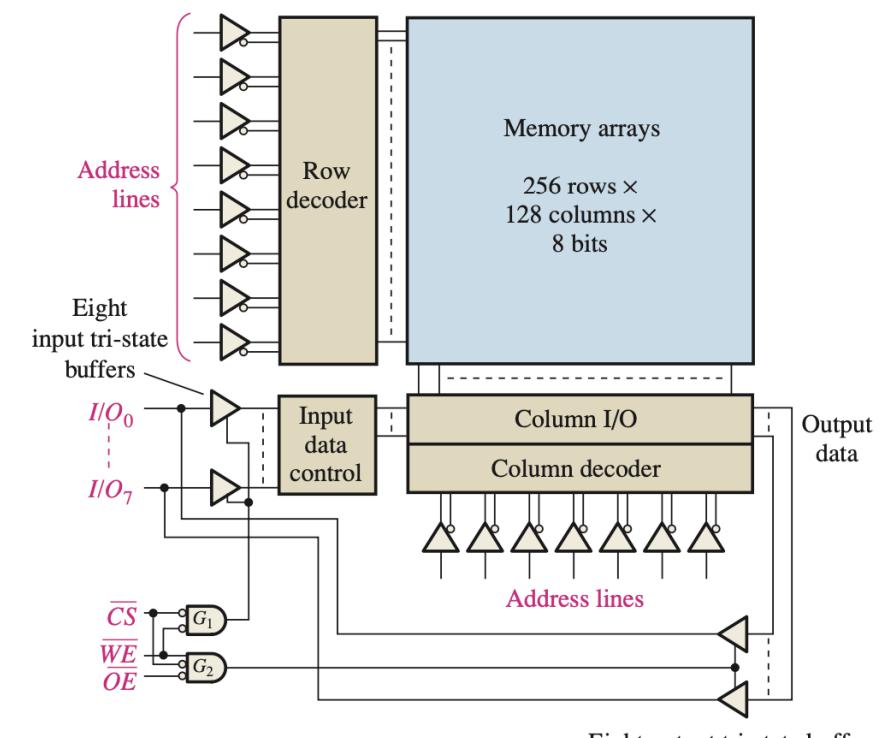
## Tri-state Outputs and Buses

- Tri-state buffers in a memory allow the data lines to act as either input or output lines and connect the memory to the data bus in a computer.
- A bus is one or more conductive paths that serve to interconnect two or more functional components of a system or several diverse systems.

## Memory Array



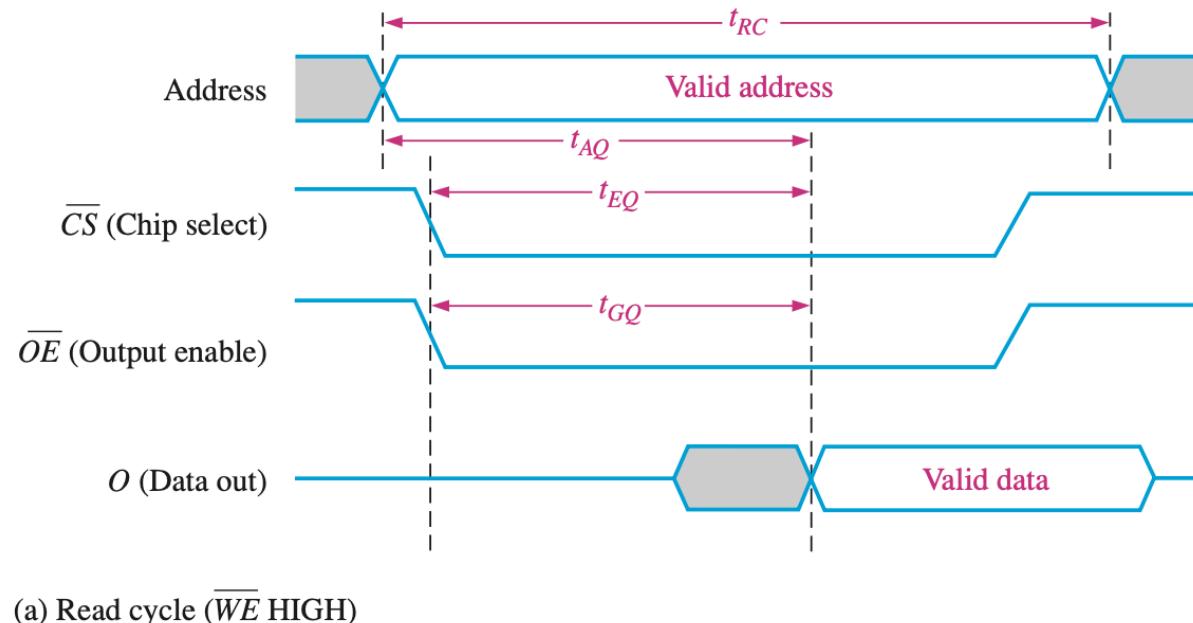
(a) Memory array configuration



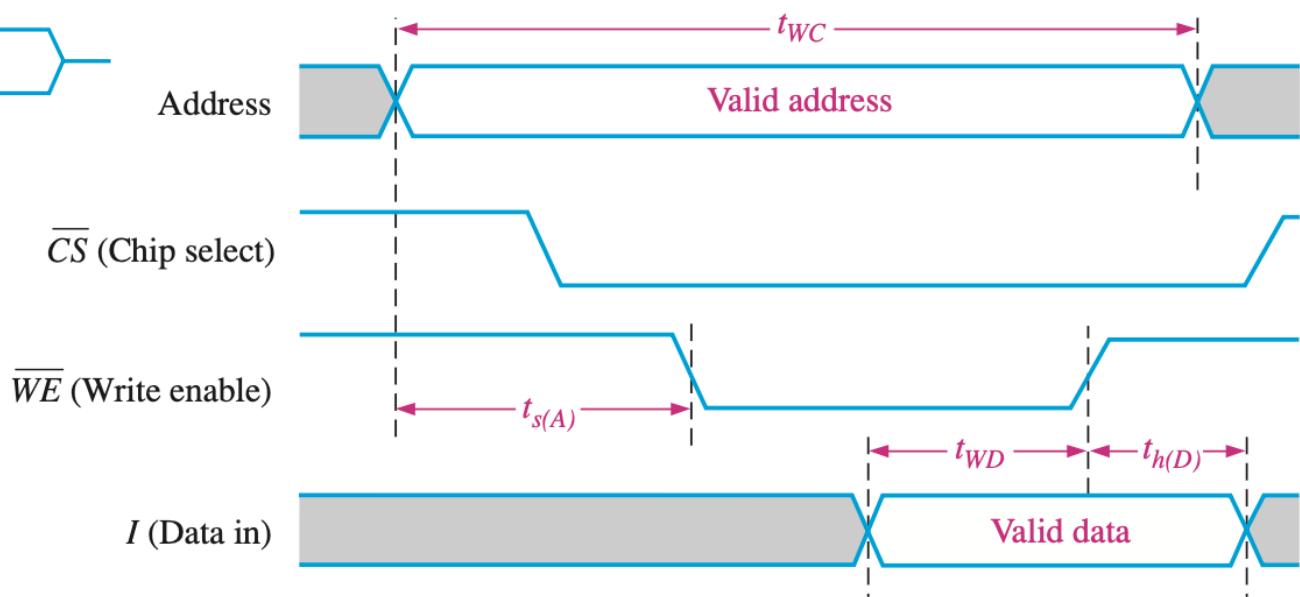
(b) Memory block diagram

FIGURE 11-12 Basic organization of an asynchronous 32k × 8 SRAM.

# Read and Write Cycles



(a) Read cycle ( $\overline{WE}$  HIGH)



(b) Write cycle ( $\overline{WE}$  LOW)

**FIGURE 11-13** Timing diagrams for typical read and write cycles for the SRAM in Figure 11-12.

# Synchronous SRAM with Burst Feature

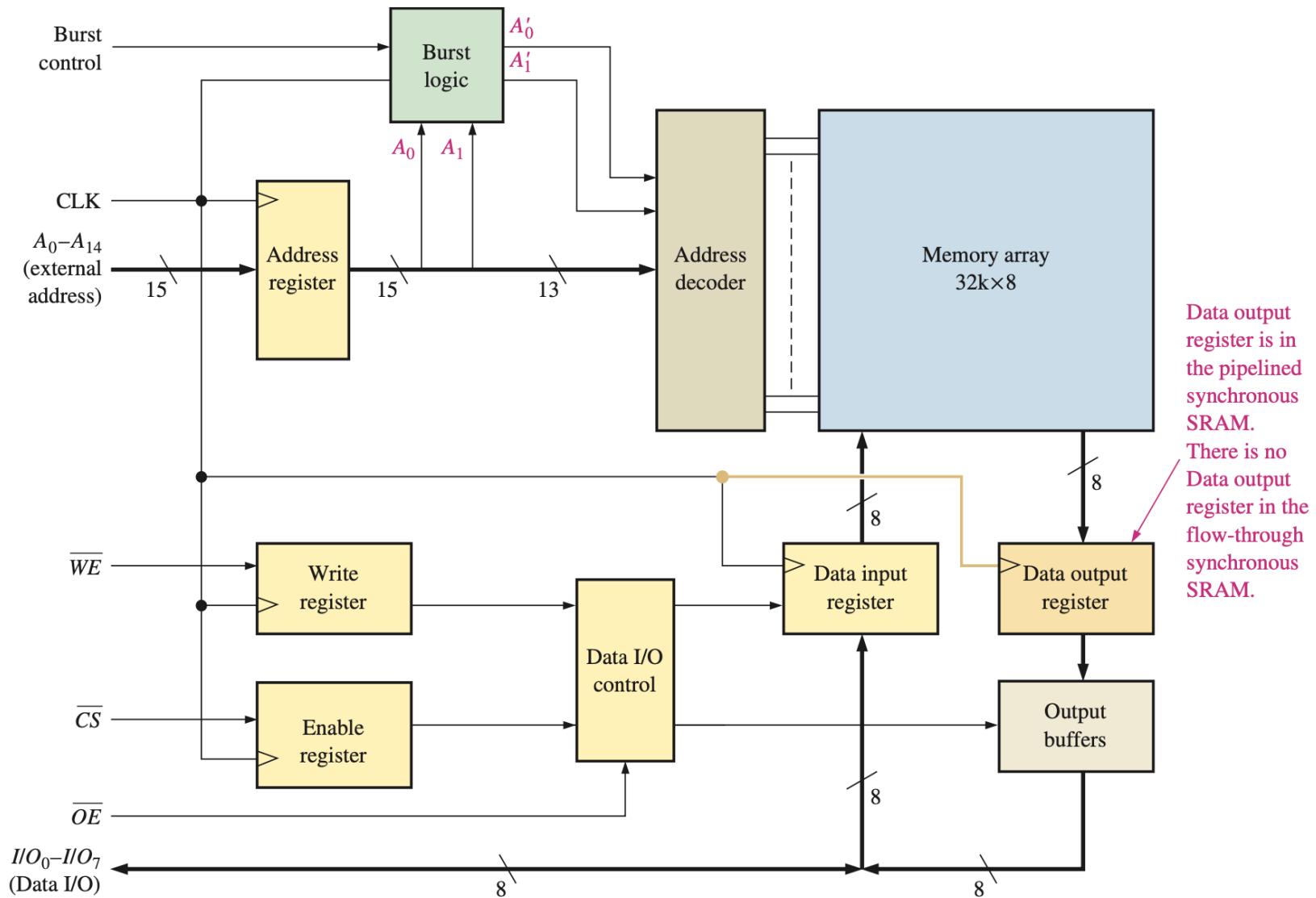
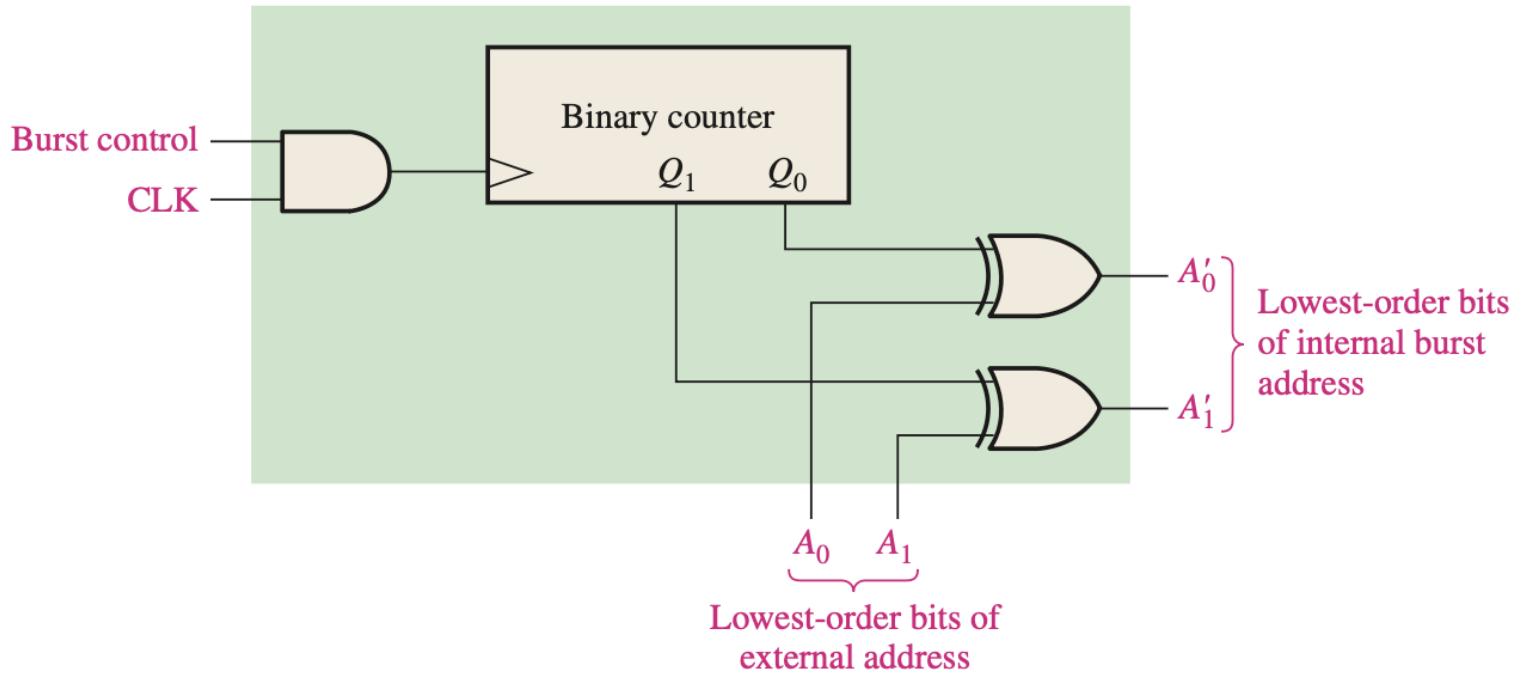


FIGURE 11-14 A basic block diagram of a synchronous SRAM with burst feature.

# The Burst Feature

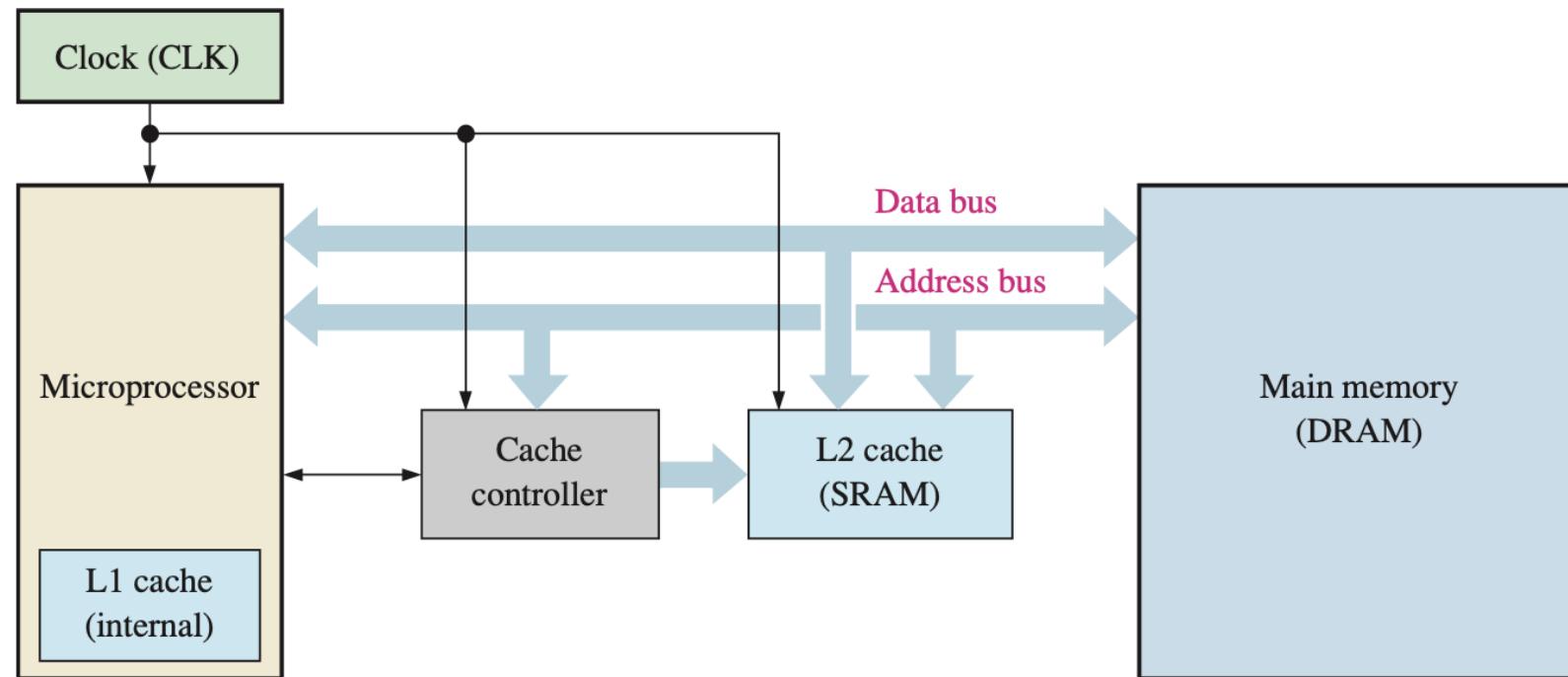


**FIGURE 11-15** Address burst logic.

# Cache Memory

**Cache memory** is a relatively small, high-speed memory that stores the most recently used instructions or data from the larger but slower main memory.

## L1 And L2 Caches

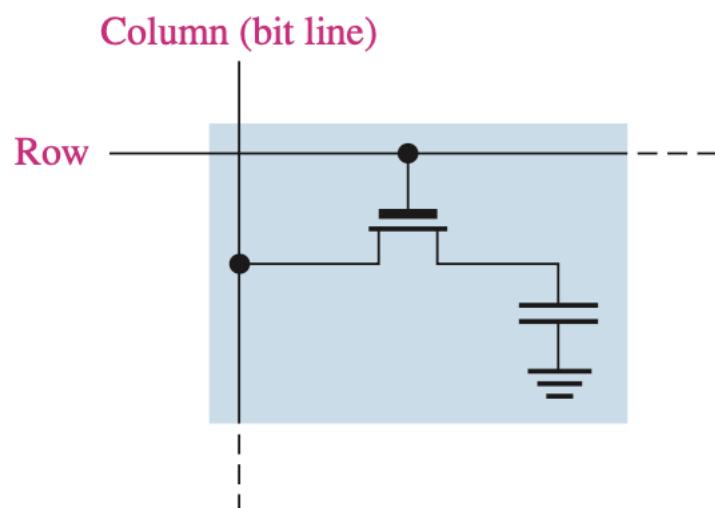


**FIGURE 11-16** Block diagram showing L1 and L2 cache memories in a computer system.

## Dynamic RAM (DRAM) Memory Cells

**Dynamic memory** cells store a data bit in a small capacitor rather than in a latch. The advantage of this type of cell is that it is very simple, thus allowing very large memory arrays to be constructed on a chip at a lower cost per bit. The disadvantage is that the storage capacitor cannot hold its charge over an extended period of time and will lose the stored data bit unless its charge is refreshed periodically. To refresh requires additional memory circuitry and complicates the operation of the DRAM.

These are the *Fast Page Mode (FPM) DRAM*, the *Extended Data Out (EDO) DRAM*, the *Burst Extended Data Out (BEDO) DRAM*, and the *Synchronous (S) DRAM*.



**FIGURE 11-17** A MOS DRAM cell.

# DRAM Organization

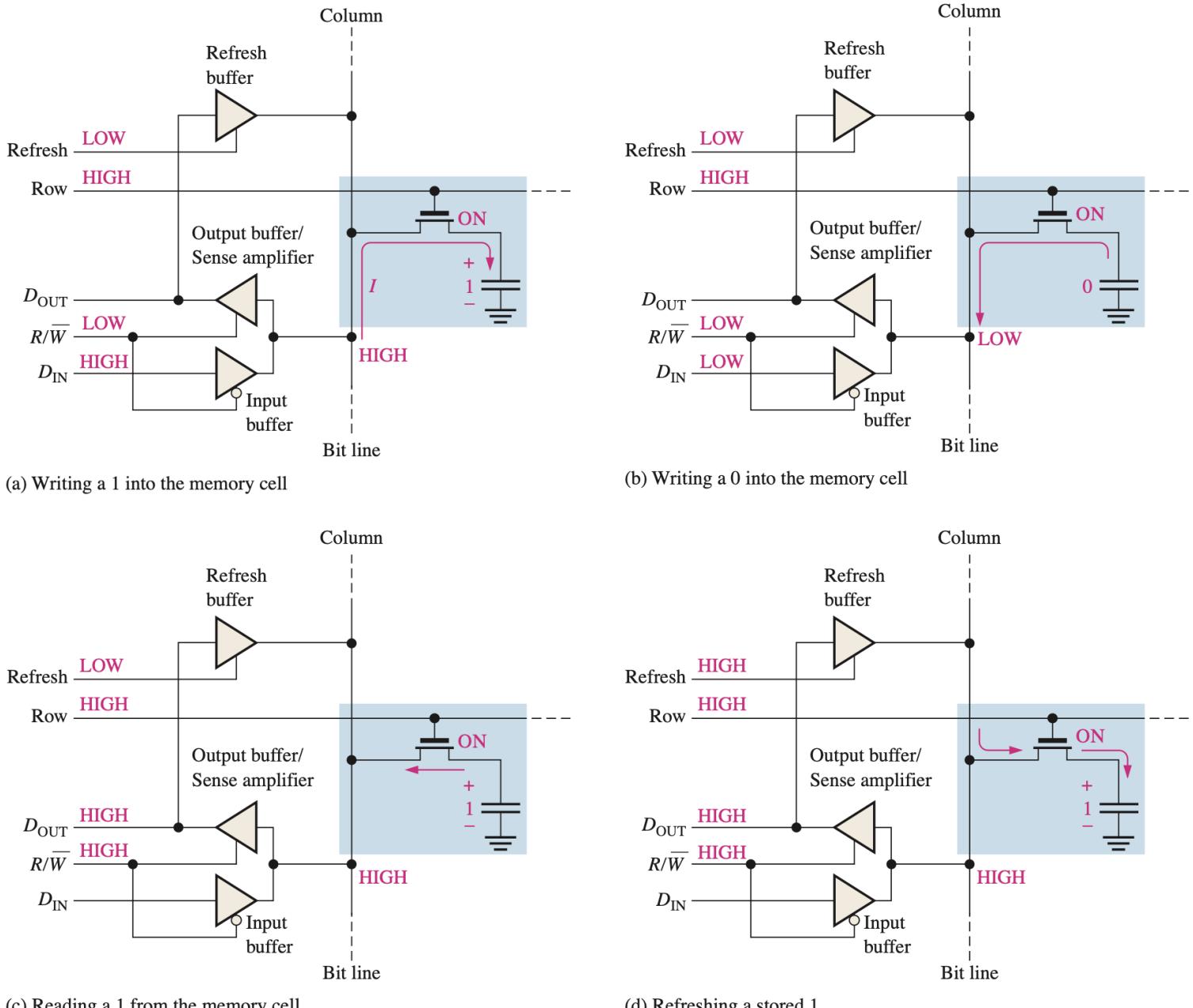
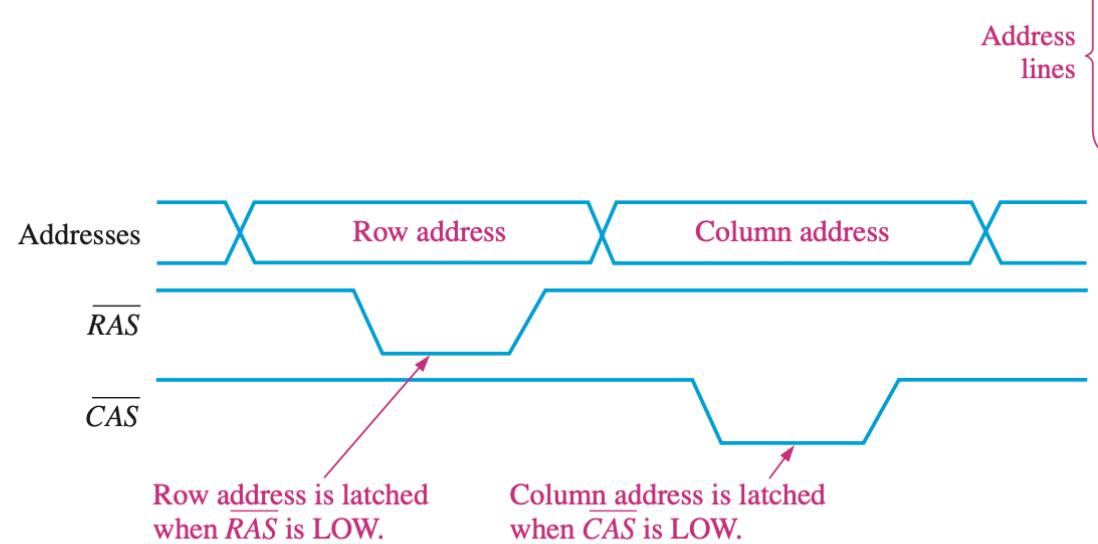
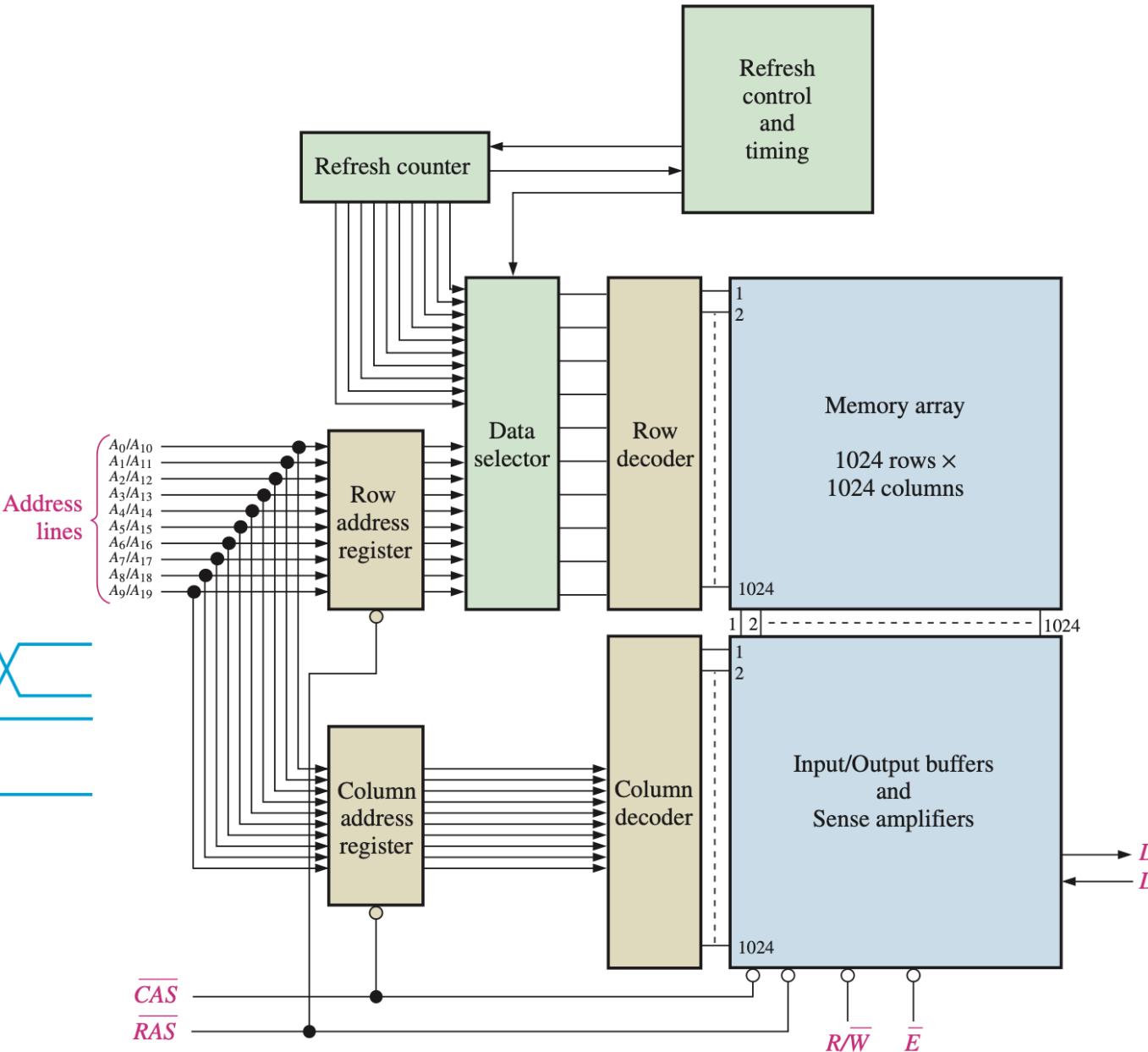


FIGURE 11-18 Basic operation of a DRAM cell.

# Address Multiplexing

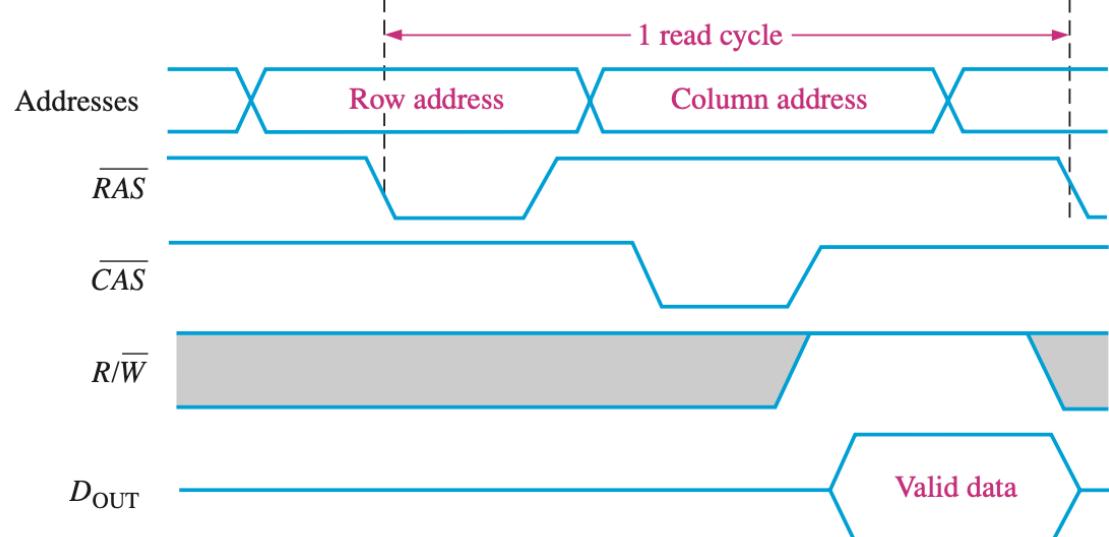


**FIGURE 11-20** Basic timing for address multiplexing.



**FIGURE 11-19** Simplified block diagram of a  $1M \times 1$  DRAM.

# Read and Write Cycles



(a) Read cycle

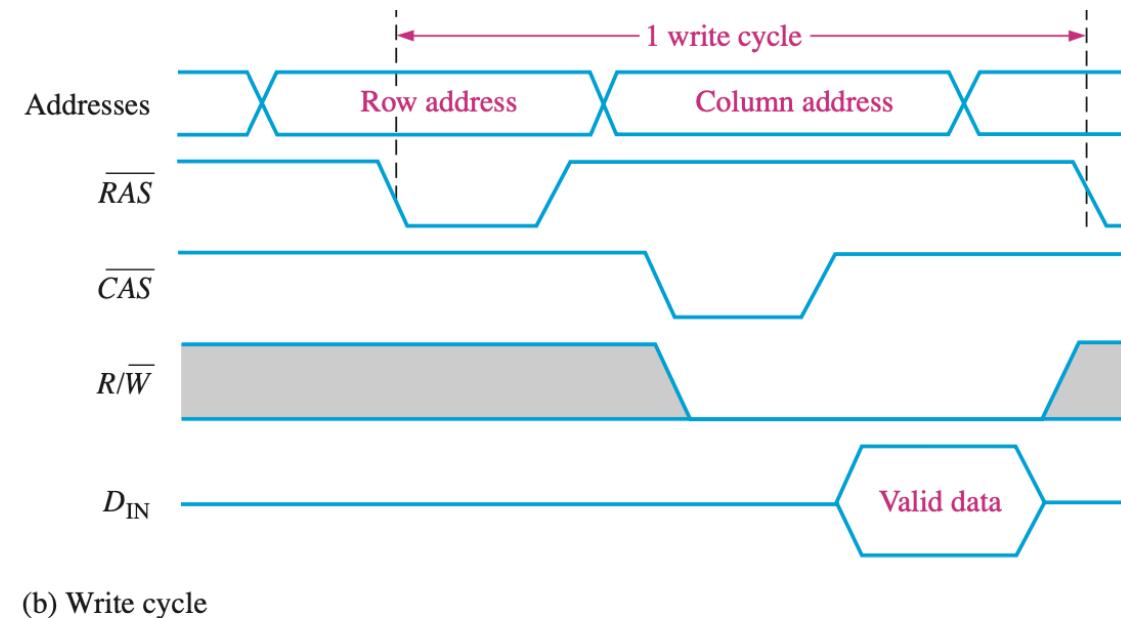


FIGURE 11-21 Timing diagrams for normal read and write cycles.

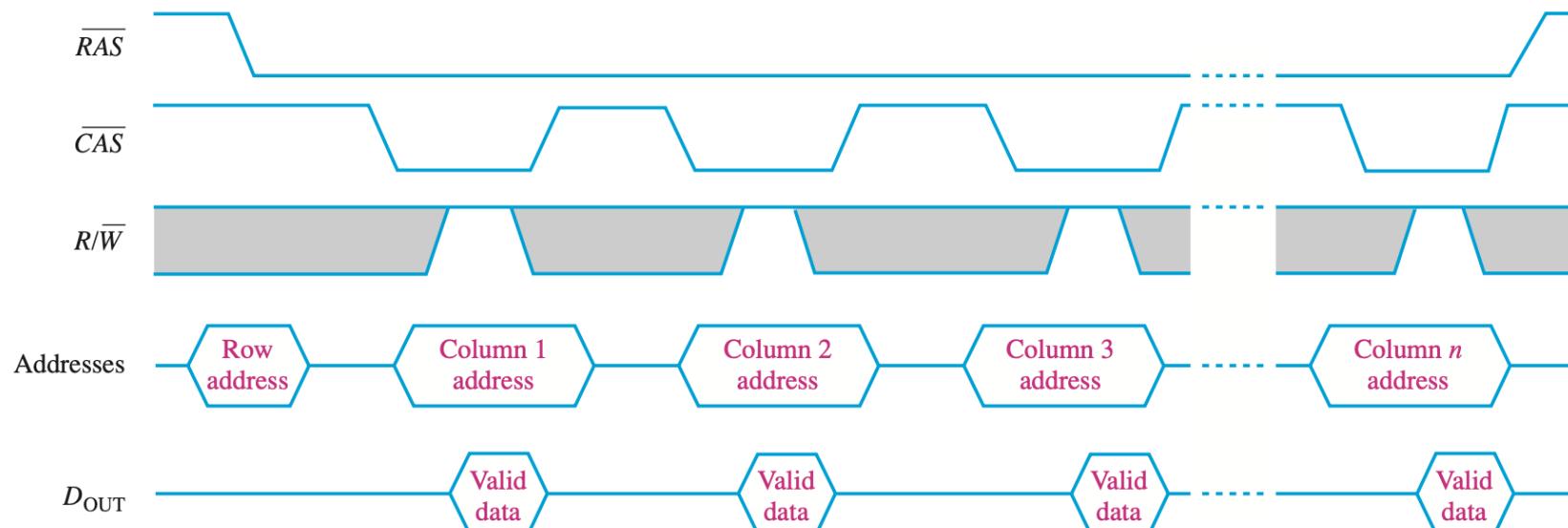
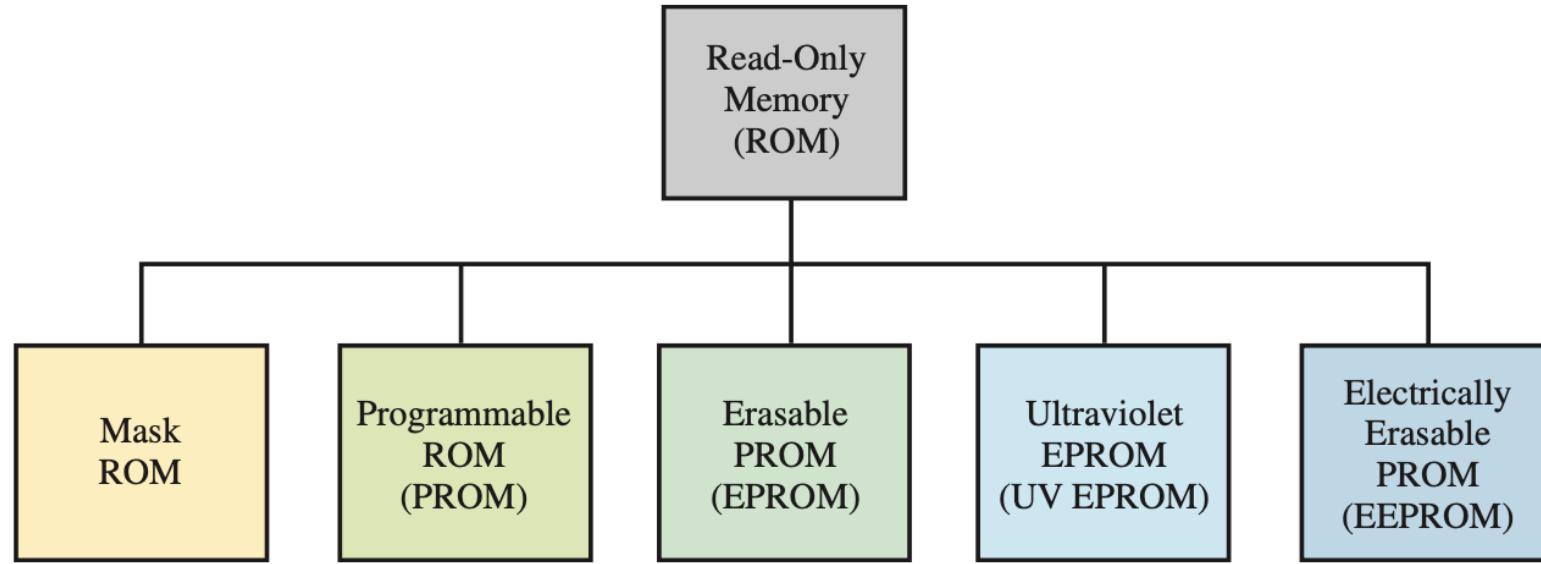


FIGURE 11-22 Fast page mode timing for a read operation.

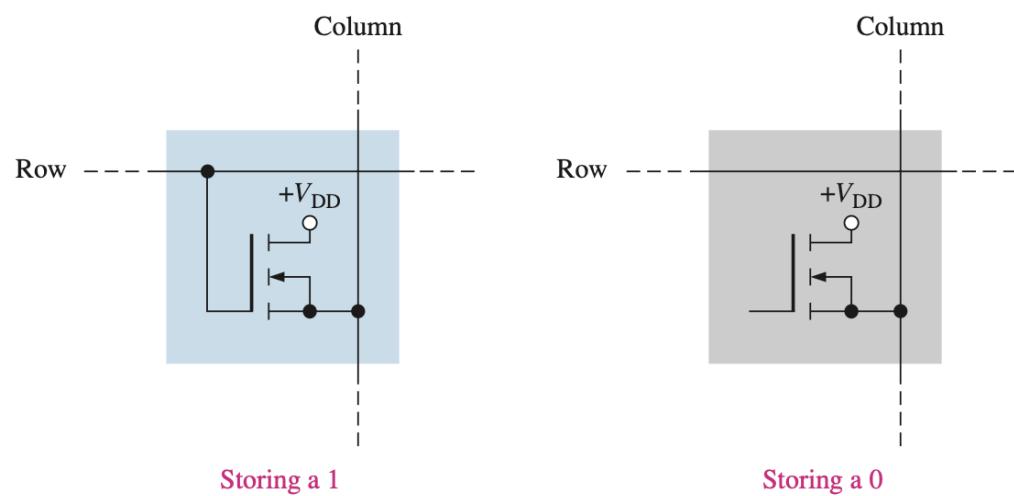
### 3. The Read-Only Memory (ROM)

#### The ROM Family

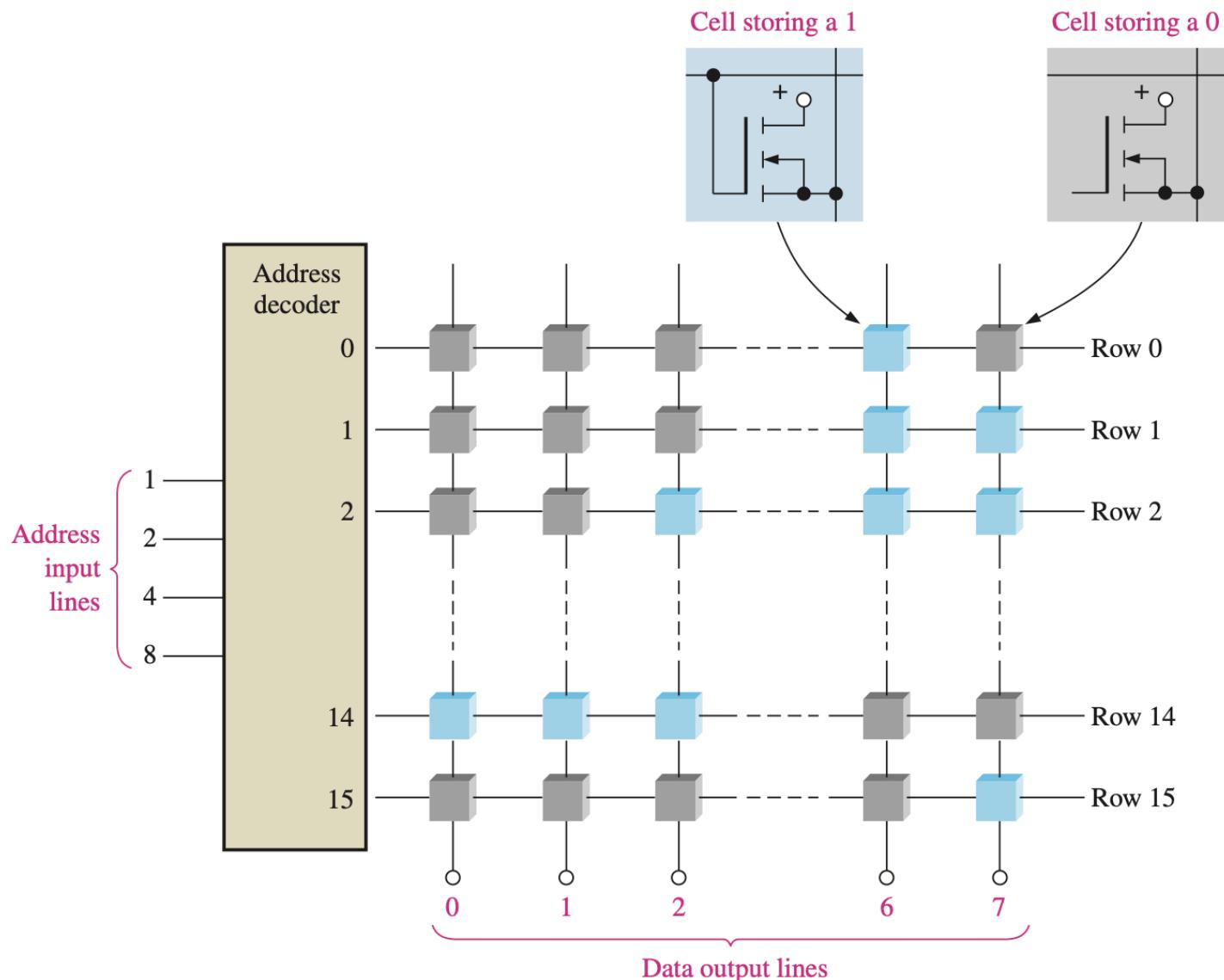


**FIGURE 11-23** The ROM family.

# The Mask ROM

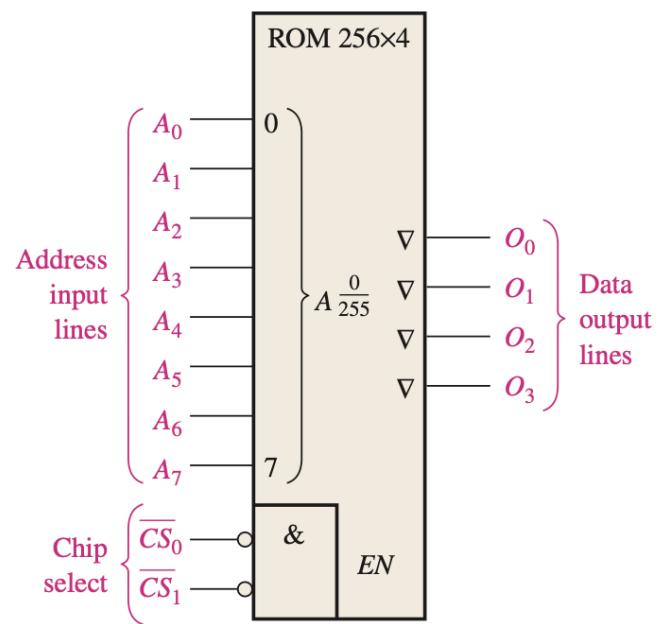


**FIGURE 11-24** ROM cells.

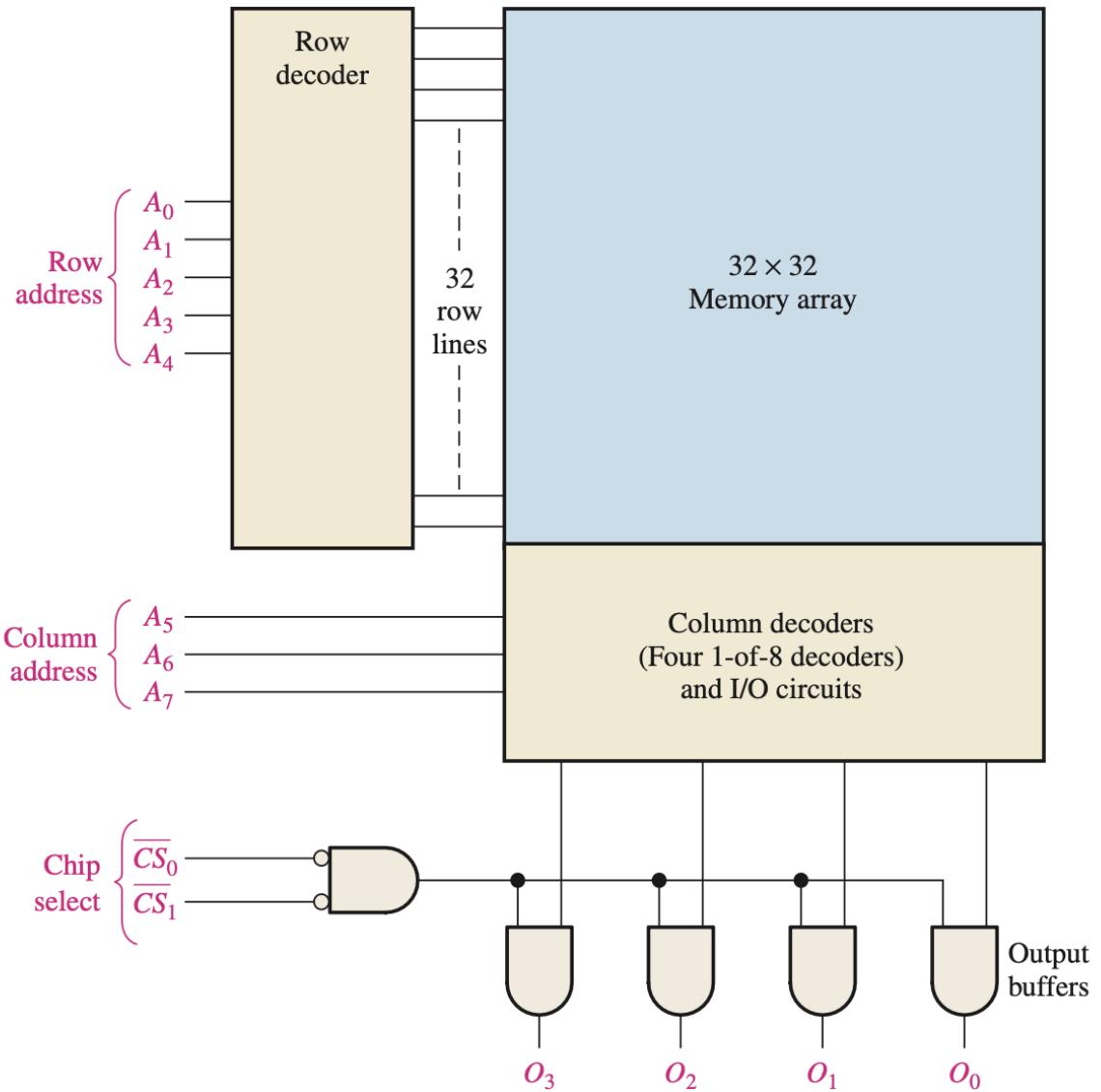


**FIGURE 11-25** A representation of a  $16 \times 8$ -bit ROM array.

# Internal ROM Organization



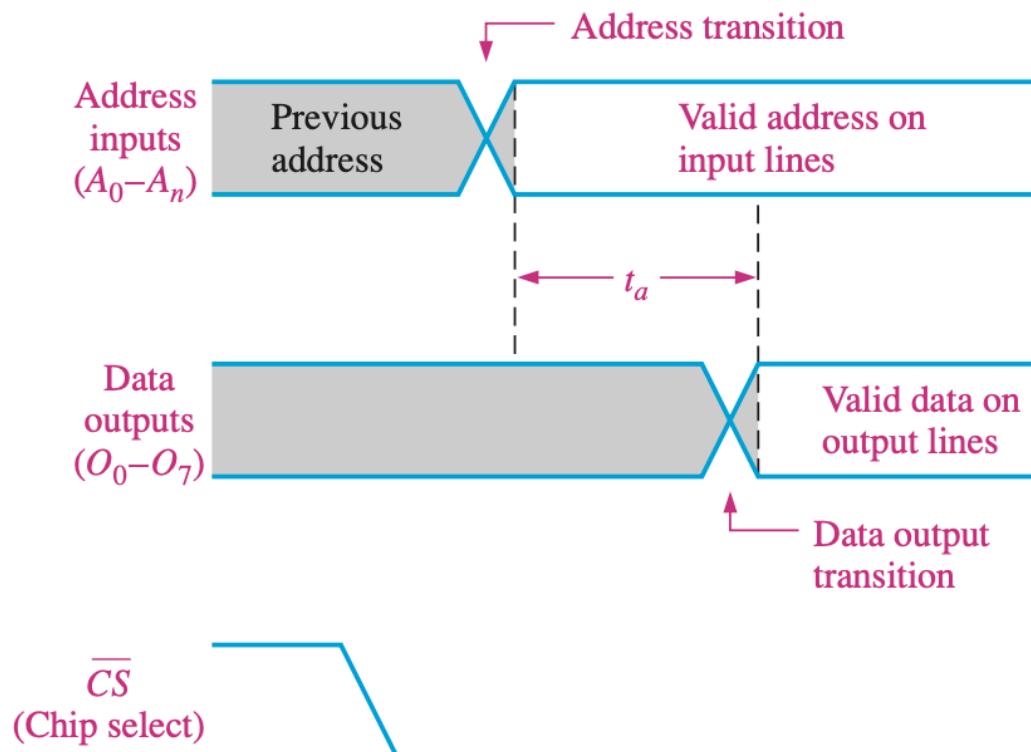
**FIGURE 11-27** A  $256 \times 4$  ROM logic symbol. The  $A_{255}$  designator means that the 8-bit address code selects addresses 0 through 255.



**FIGURE 11-28** A 1024-bit ROM with a  $256 \times 4$  organization based on a  $32 \times 32$  array.

# ROM Access Time

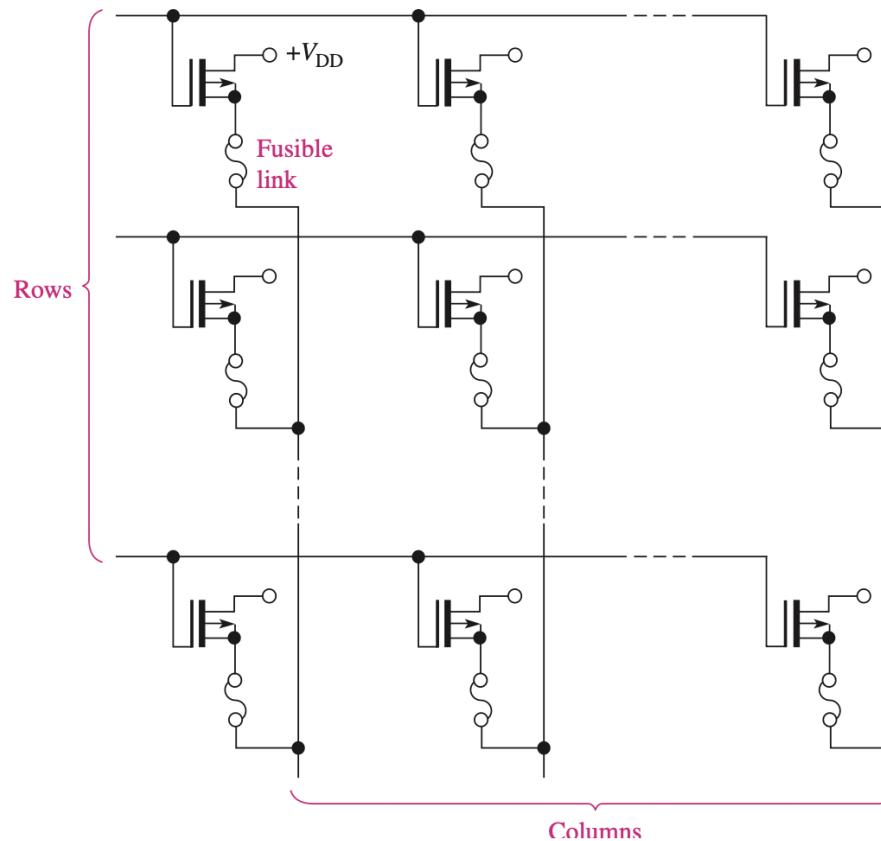
The **access time**,  $t_a$ , of a ROM is the time from the application of a valid address code on the input lines until the appearance of valid output data.



**FIGURE 11-29** ROM access time ( $t_a$ ) from address change to data output with chip select already active.

## 4. Programmable ROMs PROMs

- A PROM uses some type of fusing process to store bits, in which a memory link is burned open or left intact to represent a 0 or a 1. The fusing process is irreversible; once a PROM is programmed, it cannot be changed.



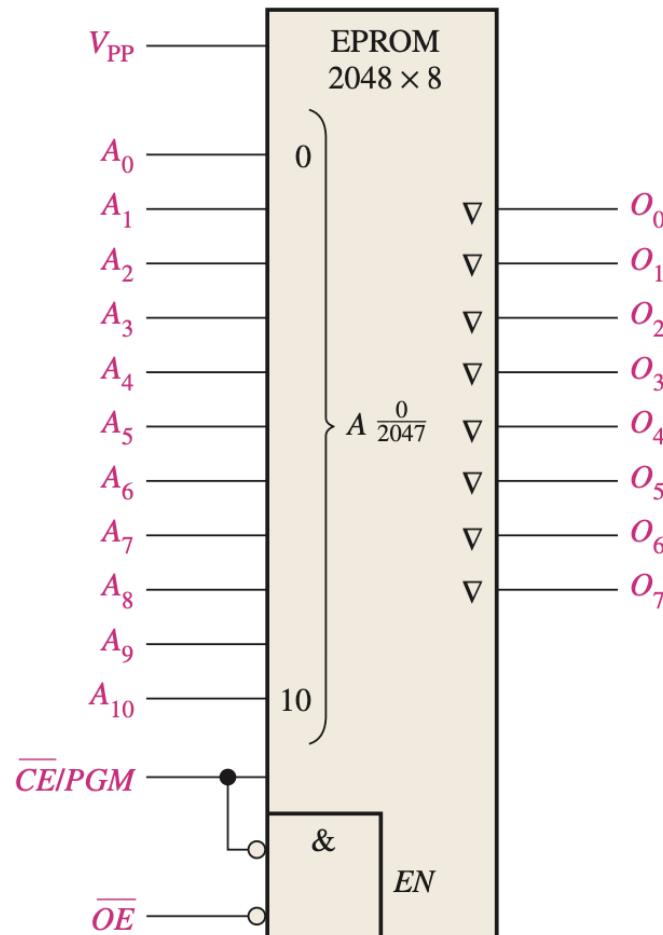
**FIGURE 11-30** MOS PROM array with fusible links. (All drains are commonly connected to  $V_{DD}$ .)

Three basic fuse technologies used in PROMs are metal links, silicon links, and pn junctions. A brief description of each of these follows.

- Metal links are made of a material such as nichrome. Each bit in the memory array is represented by a separate link. During programming, the link is either “blown” open or left intact. This is done basically by first addressing a given cell and then forcing a sufficient amount of current through the link to cause it to open.
- Silicon links are formed by narrow, notched strips of polycrystalline silicon. Programming of these fuses requires melting of the links by passing a sufficient amount of current through them. This amount of current causes a high temperature at the fuse location that oxidizes the silicon and forms an insulation around the now-open link.
- Shorted junction, or avalanche-induced migration, technology consists basically of two pn junctions arranged back-to-back. During programming, one of the diode junctions is avalanche, and the resulting voltage and heat cause aluminum ions to migrate and short the junction. The remaining junction is then used as a forward-biased diode to represent a data bit.

# EPROMs

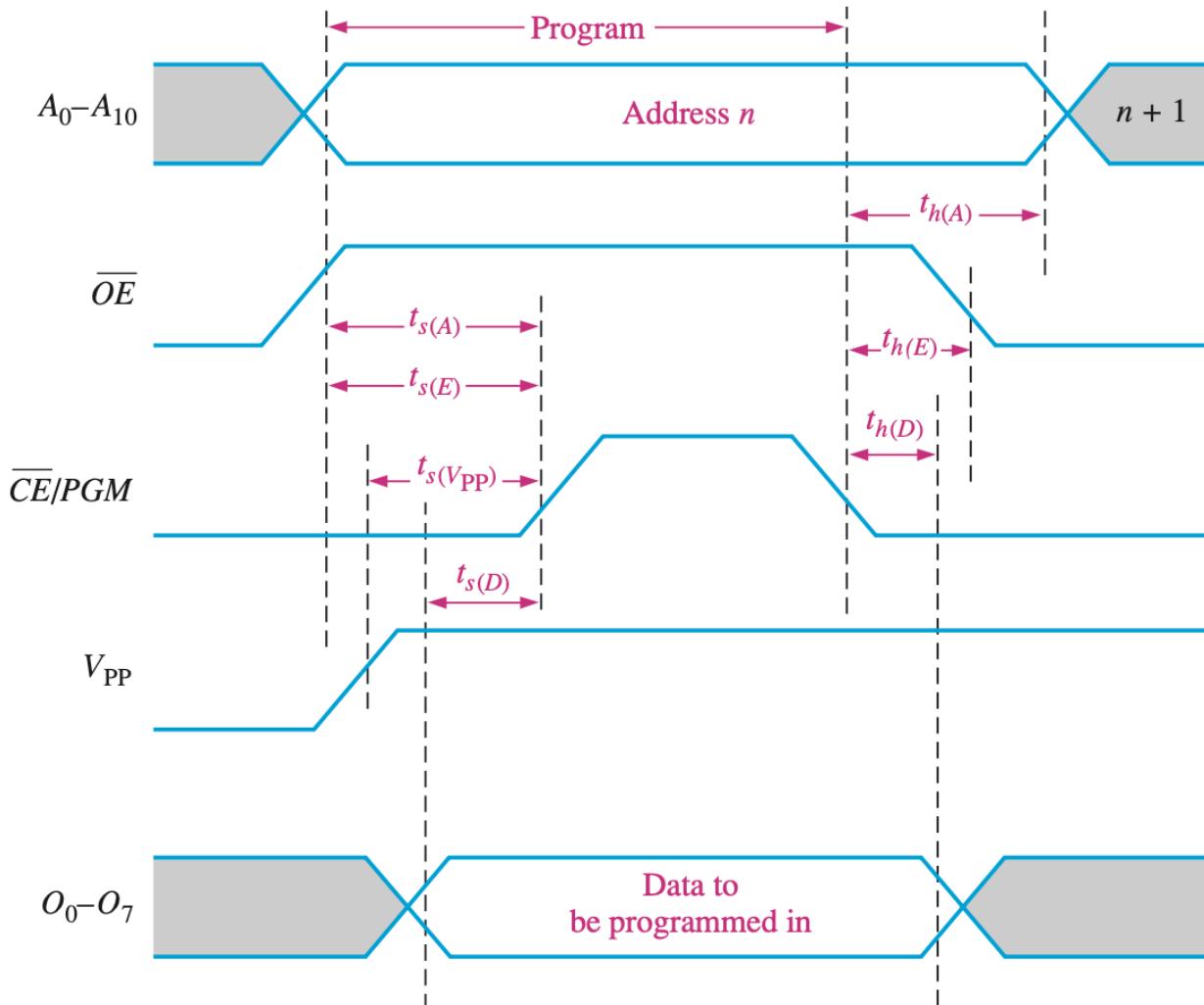
- An EPROM is an erasable PROM. Unlike an ordinary PROM, an EPROM can be reprogrammed if an existing program in the memory array is erased first.
- An EPROM uses an NMOSFET array with an isolated-gate structure.



**FIGURE 11–31** The logic symbol for a 2048 × 8 EPROM.

# EPROMs

- A timing diagram for the programming is:



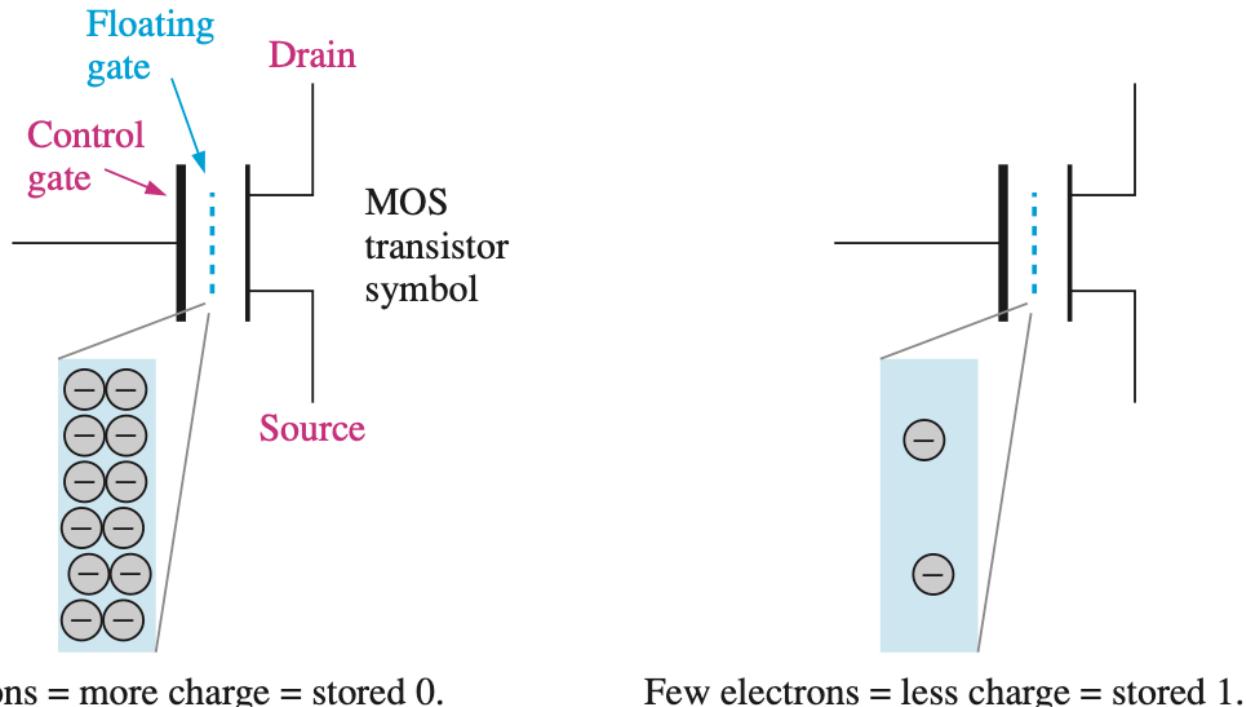
**FIGURE 11-32** Timing diagram for a 2048  $\times$  8 EPROM programming cycle, with critical setup times ( $t_s$ ) and hold times ( $t_h$ ) indicated.

## 5. The Flash Memory

- Flash memories are high-density read/write memories (high-density translates into large bit storage capacity) that are nonvolatile, which means that data can be stored indefinitely without power.
- High-density means that a large number of cells can be packed into a given surface area on a chip; that is, the higher the density, the more bits that can be stored on a given size chip.
- This high density is achieved in flash memories with a storage cell that consists of a single floating-gate MOS transistor.
- A data bit is stored as charge or the absence of charge on the floating gate depending if a 0 or a 1 is stored.

# Flash Memory Cell

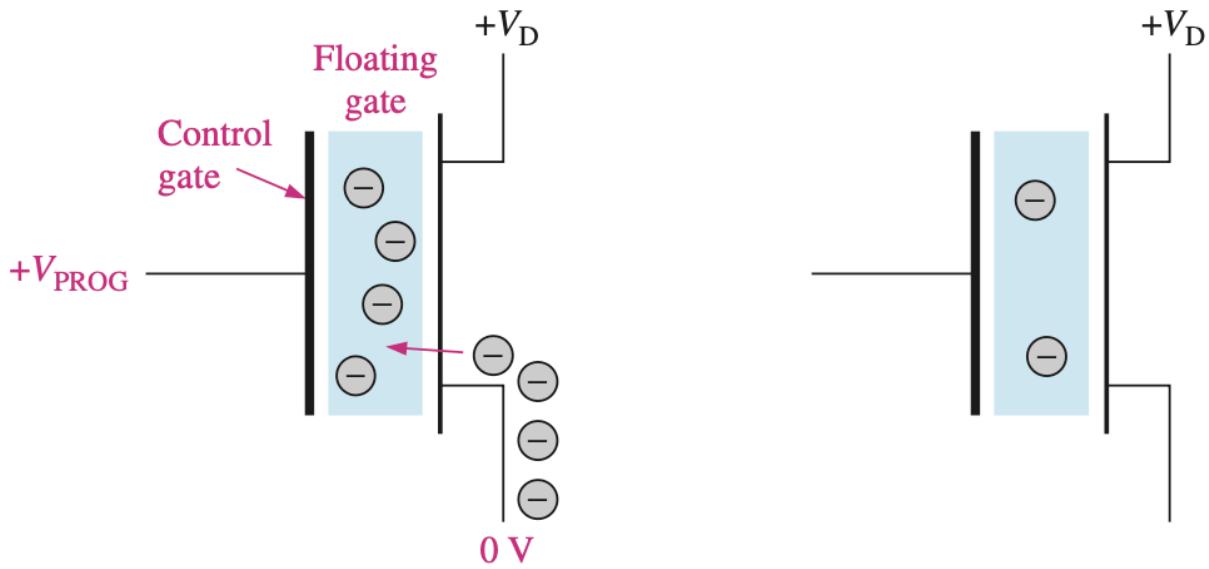
- The floating gate stores electrons (charge) because of a sufficient voltage applied to the control gate.
- A 0 is stored when there is more charge and a 1 is stored when there is less or no charge.
- The amount of charge present on the floating gate determines if the transistor will turn on and conduct current from the drain to the source when a control voltage is applied during a read operation.



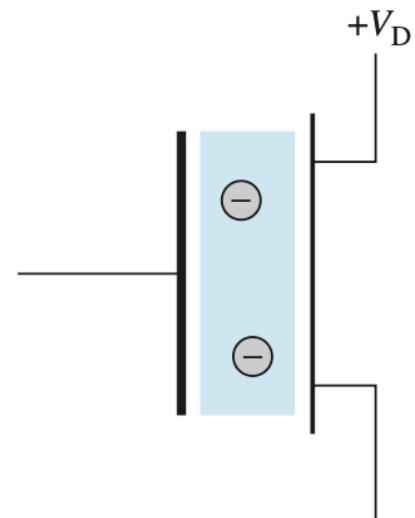
**FIGURE 11-33** The storage cell in a flash memory.

# Basic Flash Memory Operation

## Programming



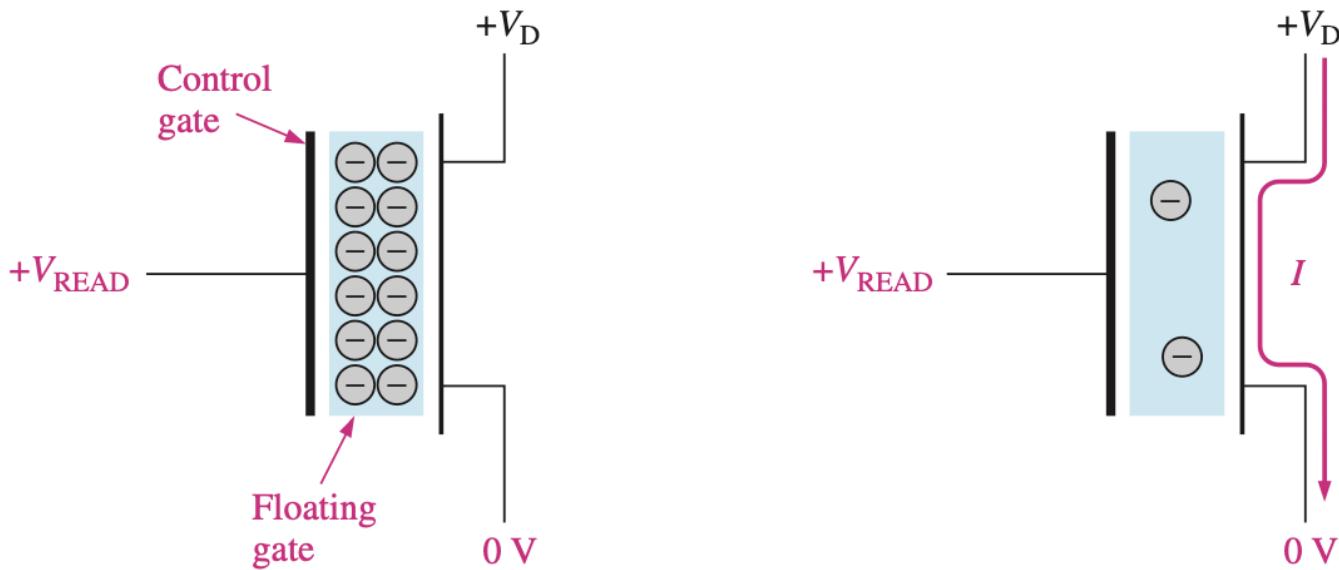
To store a 0, a sufficient positive voltage is applied to the control gate with respect to the source to add charge to the floating gate during programming.



To store a 1, no charge is added and the cell is left in the erased condition.

**FIGURE 11–34** Simplified illustration of storing a 0 or a 1 in a flash cell during the programming operation.

# Read

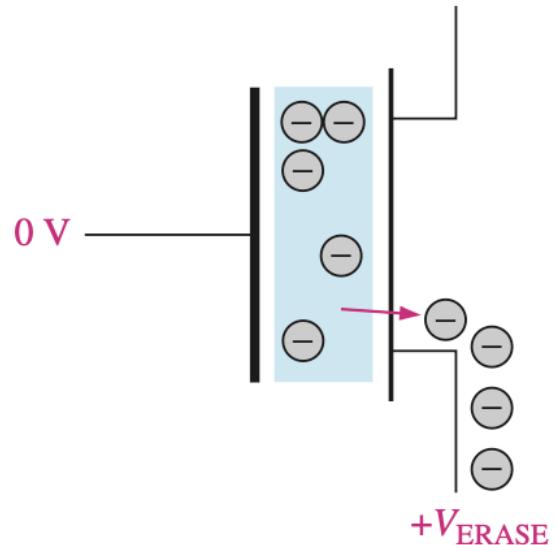


When a 0 is read, the transistor remains off because the charge on the floating gate prevents the read voltage from exceeding the turn-on threshold.

When a 1 is read, the transistor turns on because the absence of charge on the floating gate allows the read voltage to exceed the turn-on threshold.

**FIGURE 11–35** The read operation of a flash cell in an array.

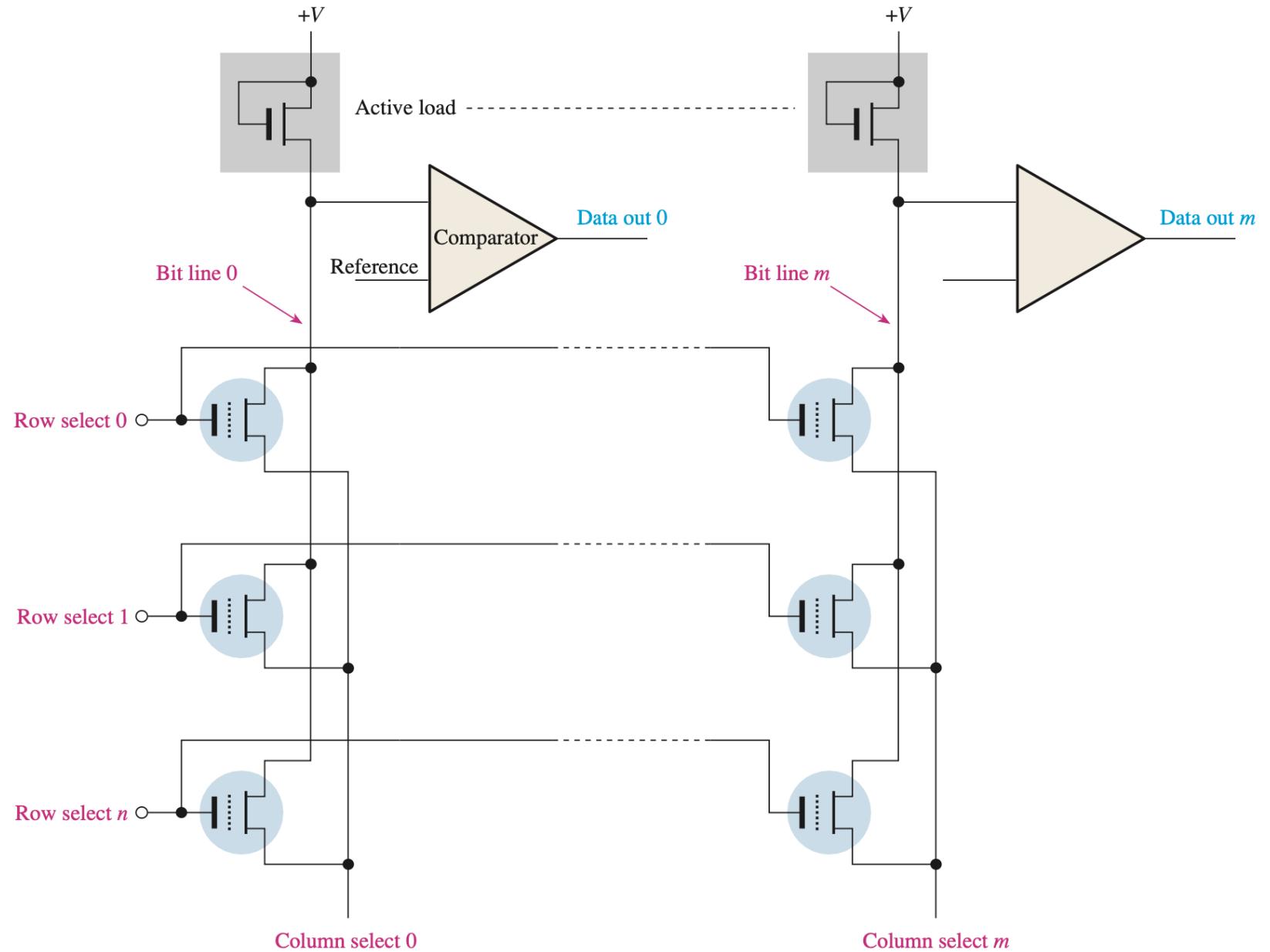
# Erase



To erase a cell, a sufficient positive voltage is applied to the source with respect to the control gate to remove charge from the floating gate during the erase operation.

**FIGURE 11–36** Simplified illustration of removing charge from a cell during erase.

# Flash Memory Array

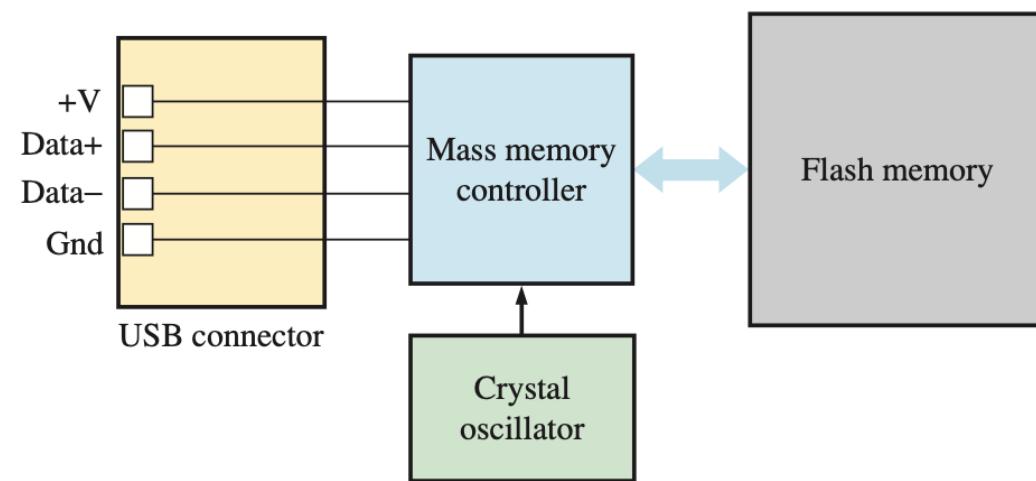


**FIGURE 11-37** Basic flash memory array.

# USB Flash Drive

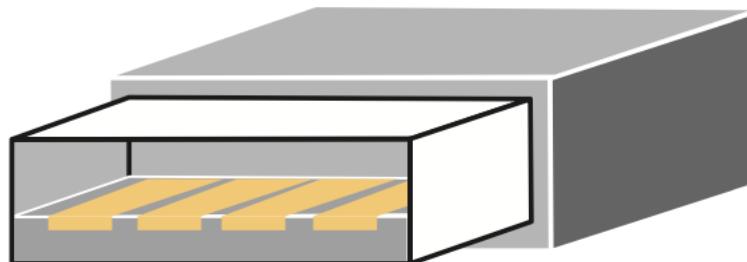


(a) Typical USB flash drive

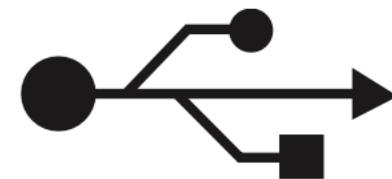


(b) Basic block diagram

**FIGURE 11-38** The USB flash drive.



(a) Type A USB connector

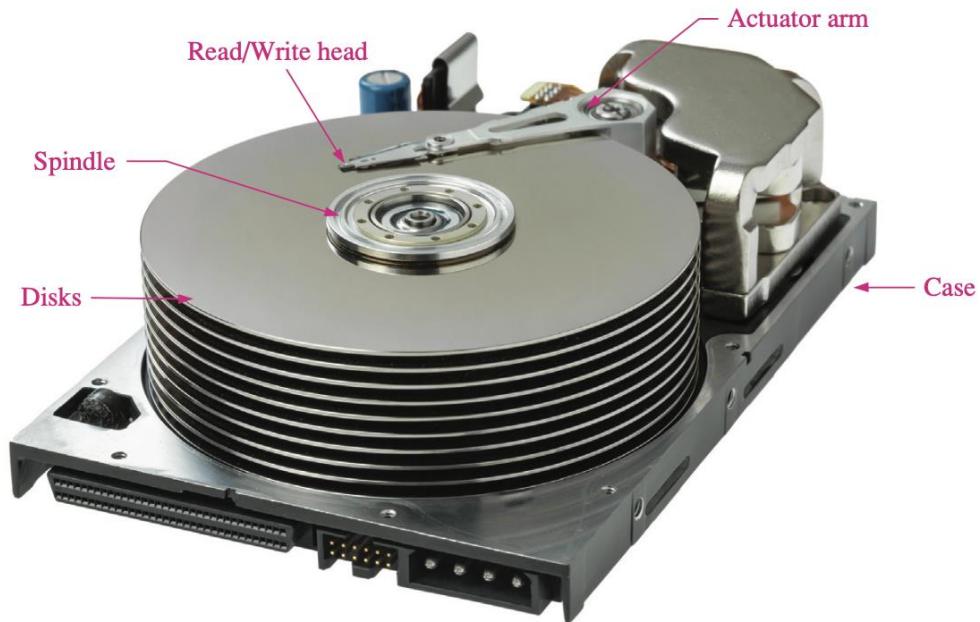


(b) USB icon

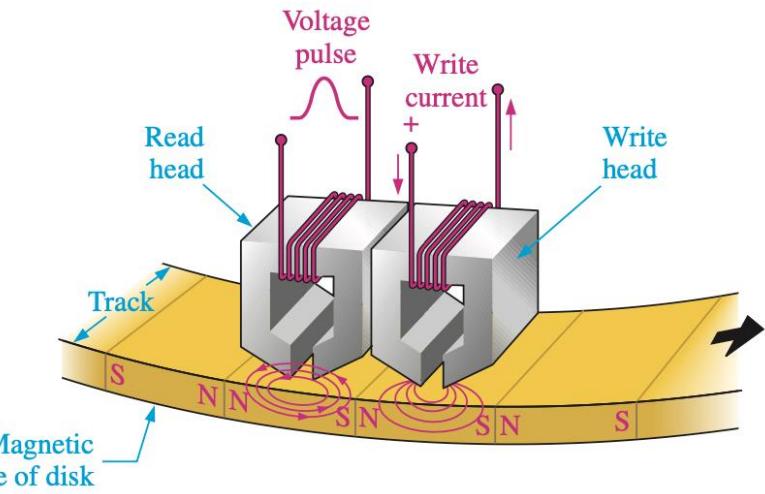
**FIGURE 11-39** Connector and symbol.

# 6. Magnetic and Optical Storage

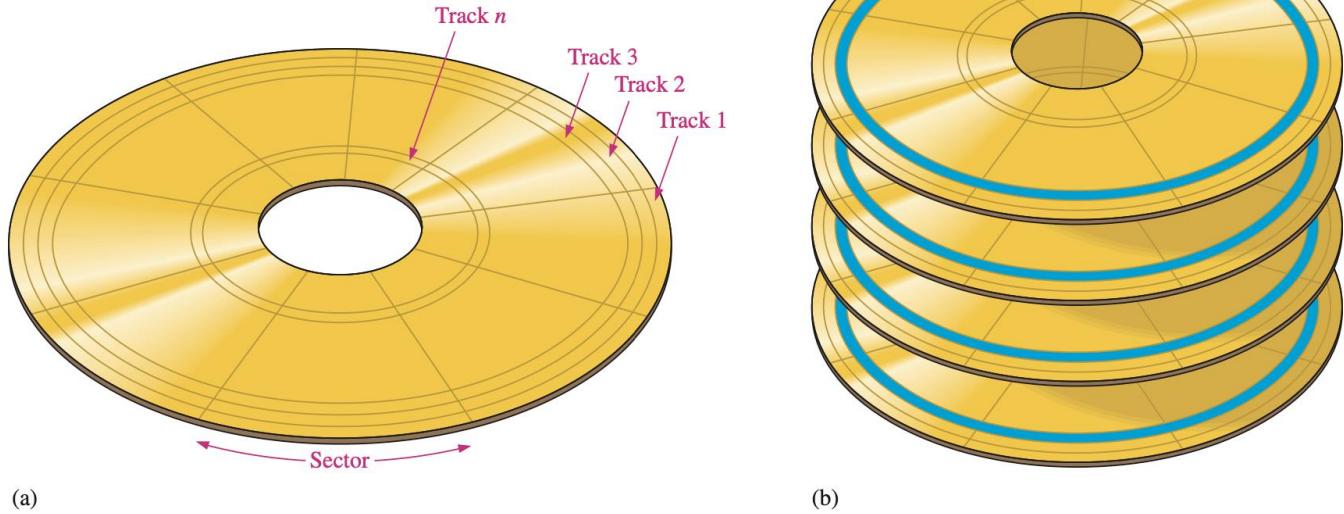
## Magnetic Storage



**FIGURE 11-59** A hard disk drive. FrameAngel/Shutterstock

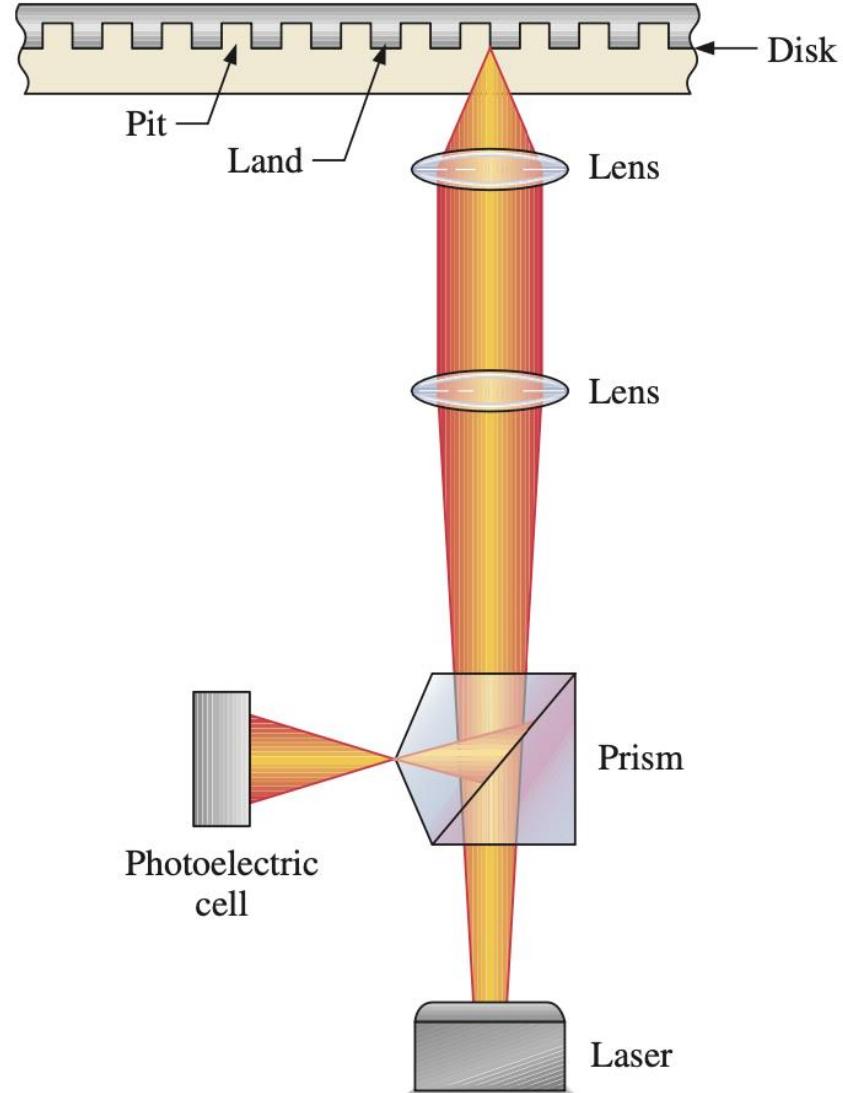


**FIGURE 11-60** Simplified read/write head operation.



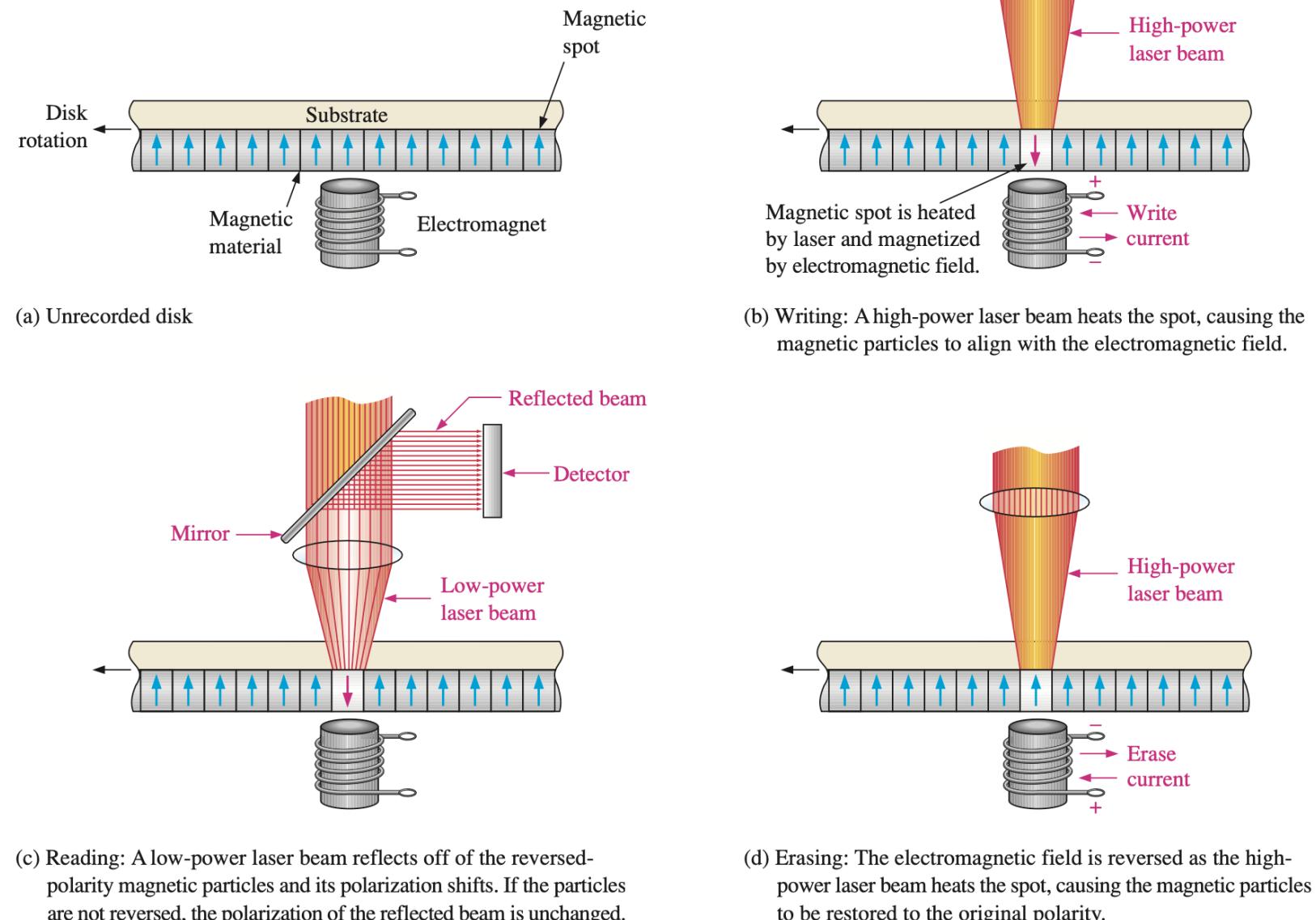
**FIGURE 11-61** Hard disk organization and formatting.

# Optical Storage



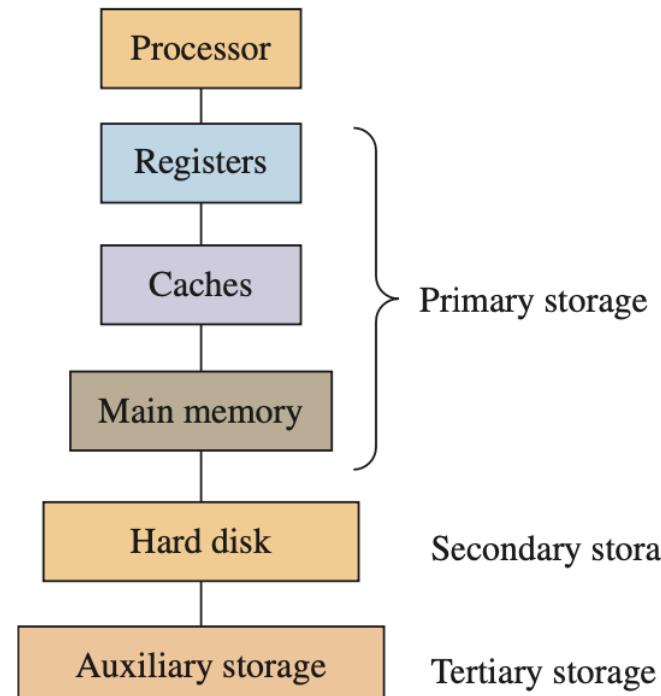
**FIGURE 11-64** Basic operation of reading data from a CD-ROM.

# Magneto-Optical Storage

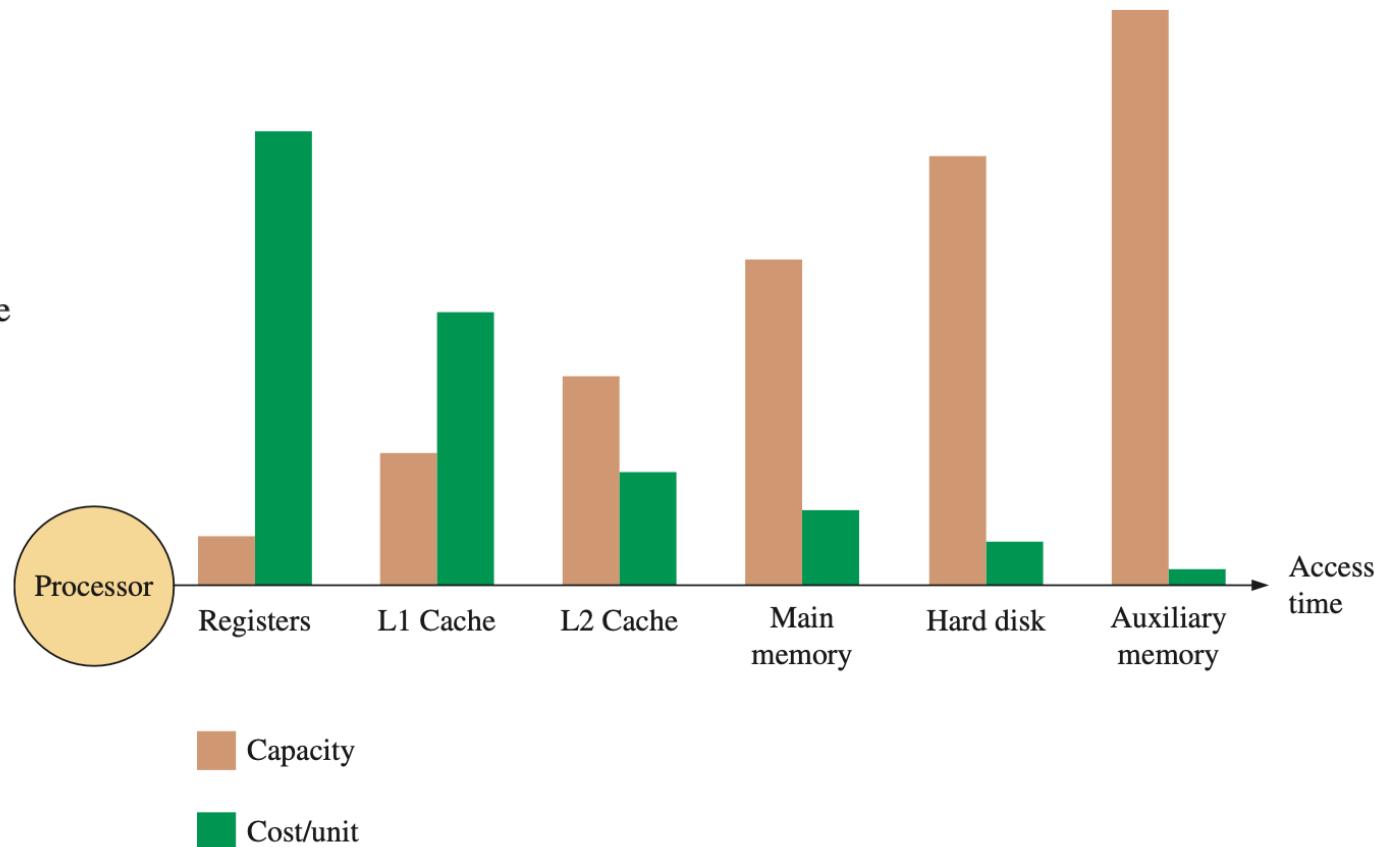


**FIGURE 11-63** Basic principle of a magneto-optical disk.

# 7. Memory Hierarchy

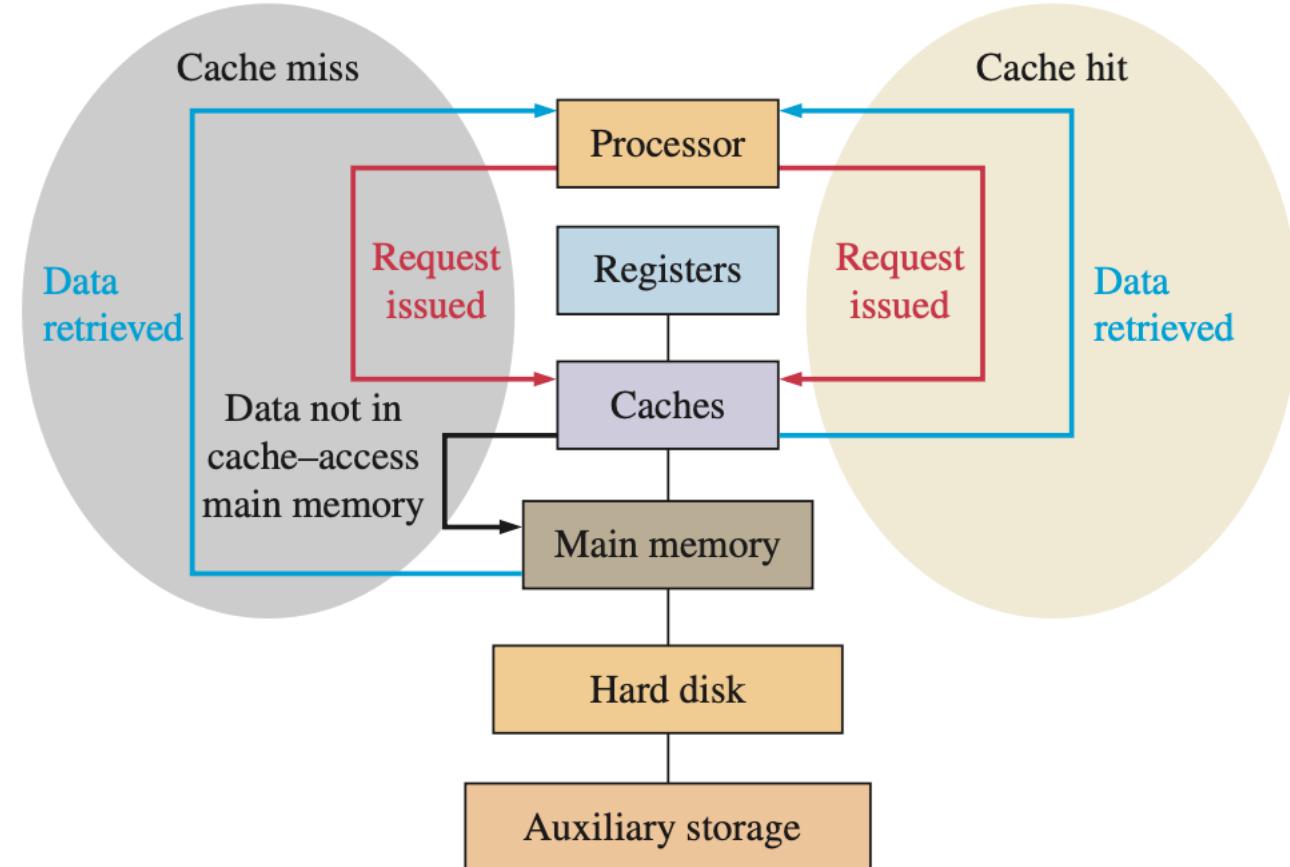


**FIGURE 11-65** Typical memory hierarchy.



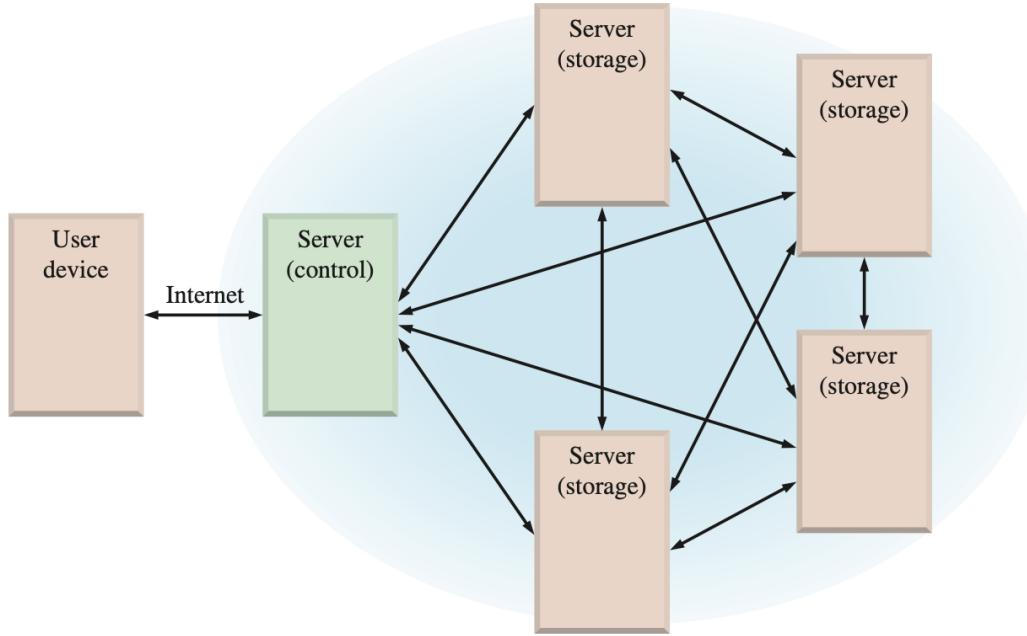
**FIGURE 11-66** Changes in memory capacity and cost per unit of data as latency (access time) increases.

# 7. Memory Hierarchy

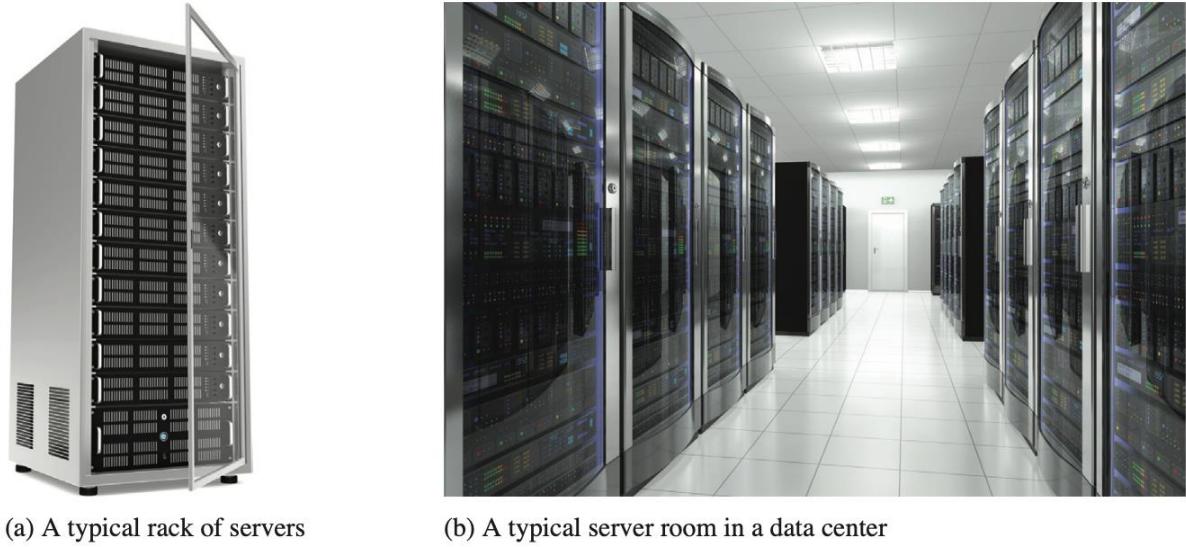


**FIGURE 11-67** Illustration of a cache hit and a miss.

# 8. Cloud Storage



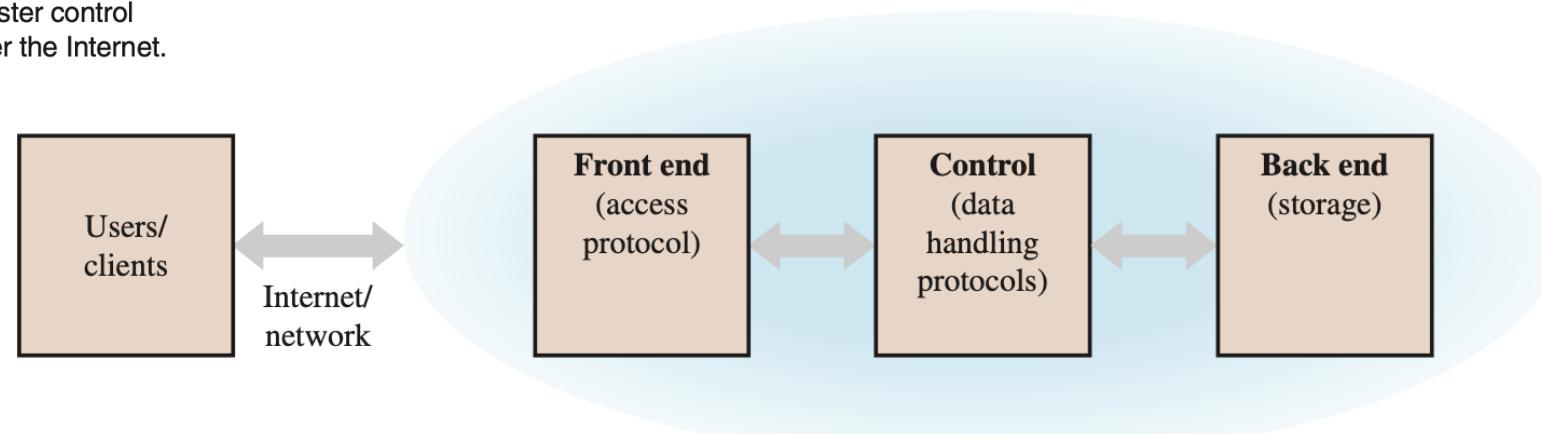
**FIGURE 11-68** A typical cloud storage system architecture consists of a master control server and several storage servers that can be accessed by a user device over the Internet.



(a) A typical rack of servers

(b) A typical server room in a data center

**FIGURE 11-69** Cloud servers. (a) Jojje/Shutterstock (b) Oleksiy Mark/Shutterstock



**FIGURE 11-71** Generic architecture of a cloud storage system.



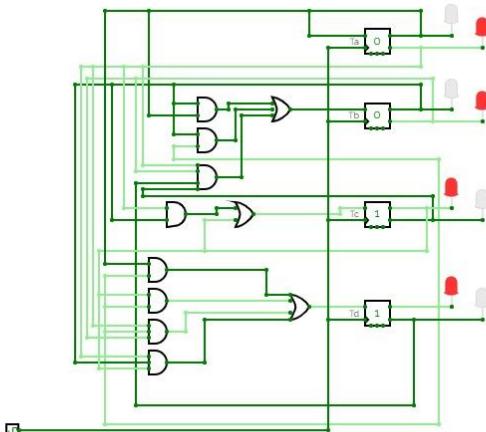
Vietnam National University HCMC  
International University



School of  
Electrical Engineering

THE END

## Lecture 11: Data Storage



INSTRUCTOR: Dr. Vuong Quoc Bao