南京理工大学
NANJING UNIVERSITY OF SCIENCE & TECHNOLOGY

第 3 章　存储层次与系统

主讲：张功萱

本章学习内容

- 存储器的分类及主要技术指标
- 存储系统的层次结构
- 半导体存储器的工作原理
- 存储器与CPU的连接
- 非易失性存储器的特点
- 并行存储系统
- Cache的工作原理
- 虚拟存储器的工作原理

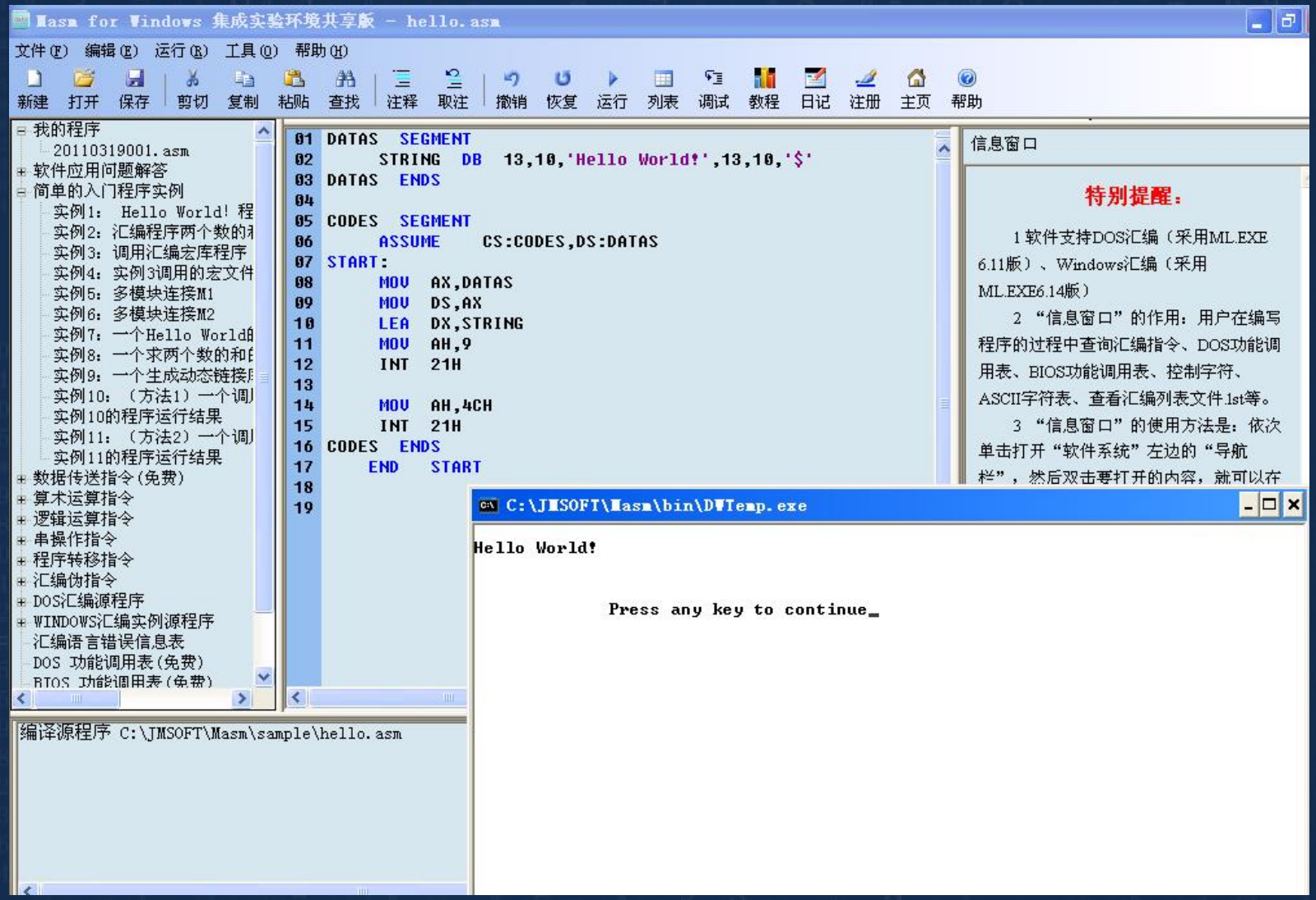
存储：
数据（各类型数据，第2章）和
程序（机器指令，第4章）

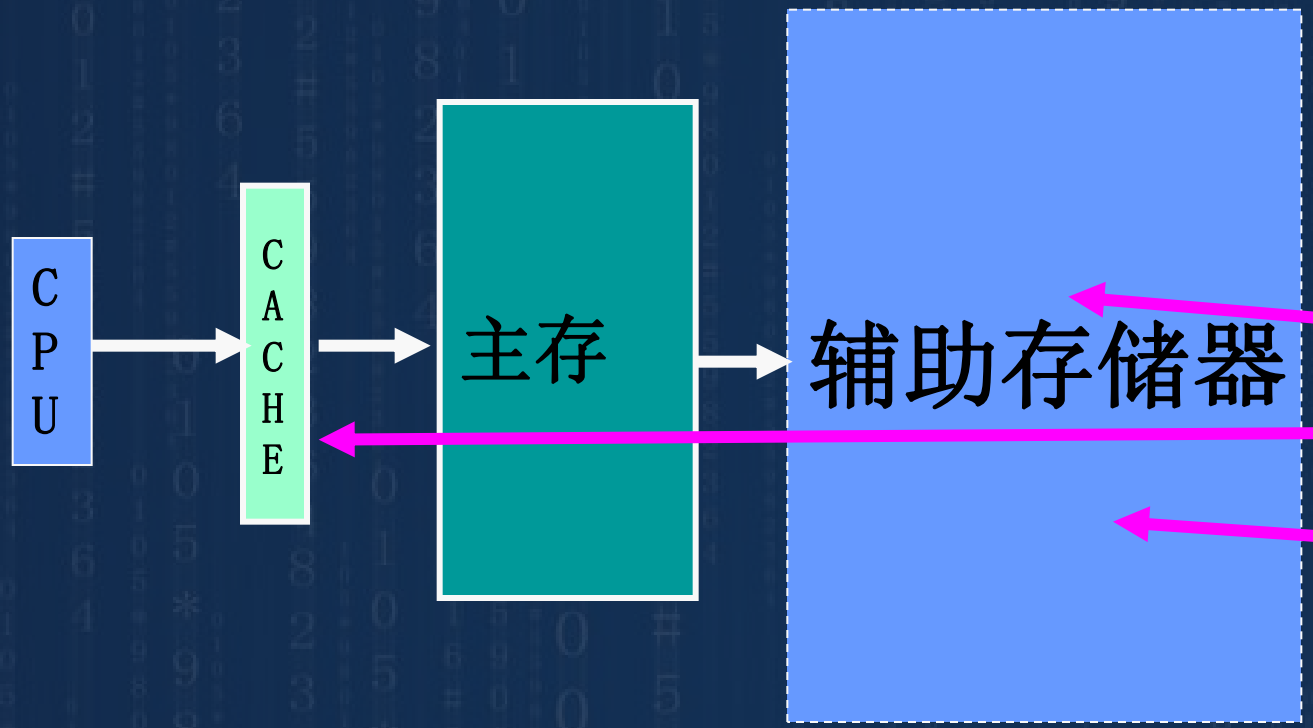
存储系统概述

第3.1节

1. 存储器有哪些类型?
2. 存储器性能指标有哪些?
3. 什么是存储层次结构?

- **存储器**：计算机的存储部件，用于存放程序（**instructions**）和 数据（**Data**）。





计算机发展的重要问题之一，就是如何设计容量大、速度快、价格低的存储器。

效果(左向右看)：Cache的速度，辅存的容量和价格

3.1.1 存储器的分类

- 1. 按与CPU的连接和功能分类
- (1) 主存储器

CPU能够直接访问的存储器。用于存放当前运行的程序和数据。

主存储器设在主机内部，所以又称内存储器。简称内存或主存。

(2) 辅助存储器

- 为解决主存容量不足而设置的存储器，用于存放当前不参加运行的程序和数据。当需要运行程序和数据时，将它们成批调入内存供CPU使用。CPU不能直接访问辅助存储器。
- 辅助存储器属于外部设备，所以又称为外存储器，简称外存或辅存。

(3) 高速缓冲存储器（Cache）

- Cache是一种介于主存与CPU之间用于解决CPU与主存间速度匹配问题的高速小容量的存储器。
- Cache用于存放CPU立即要运行或刚使用过的程序和数据。

2. 按存取方式分类

- (1) 随机存取存储器(RAM)
- RAM存储器中任何单元的内容均可按其地址随机地读取或写入，且存取时间与单元的物理位置无关。
- RAM主要用于组成主存。

- (2) 只读存储器(ROM)
- ROM存储器中任何单元的内容只能随机地读出而不能随便写入和修改。
- ROM可以作为主存的一部分，用于存放不变的程序和数据，与RAM分享相同的主存空间。ROM还可以用作其它固定存储器，如存放微程序的控制存储器、存放字符点阵图案的字符发生器等。

- (3) 顺序存取存储器(SAM)
- SAM存储器所存信息的排列、寻址和读写操作均是按顺序进行的，并且存取时间与信息在存储器中的物理位置有关。如磁带存储器，信息通常是以文件或数据块形式按顺序存放，信息在载体上没有唯一对应的地址，完全按顺序存放或读取。

- (4) 直接存取存储器（DAM）
- DAM是介于RAM和SAM之间的存储器。也称半顺序存储器。典型的DAM就是磁盘。当对磁盘进行信息存取时，先进行寻道，属于随机方式，然后在磁道中寻找扇区，属于顺序方式。

3．按存储介质分类

- 存储介质：具有两个稳定物理状态，可用来记忆二进制代码的物质或物理器件。
- 目前，构成存储器的存储介质主要是半导体器件和磁性材料。

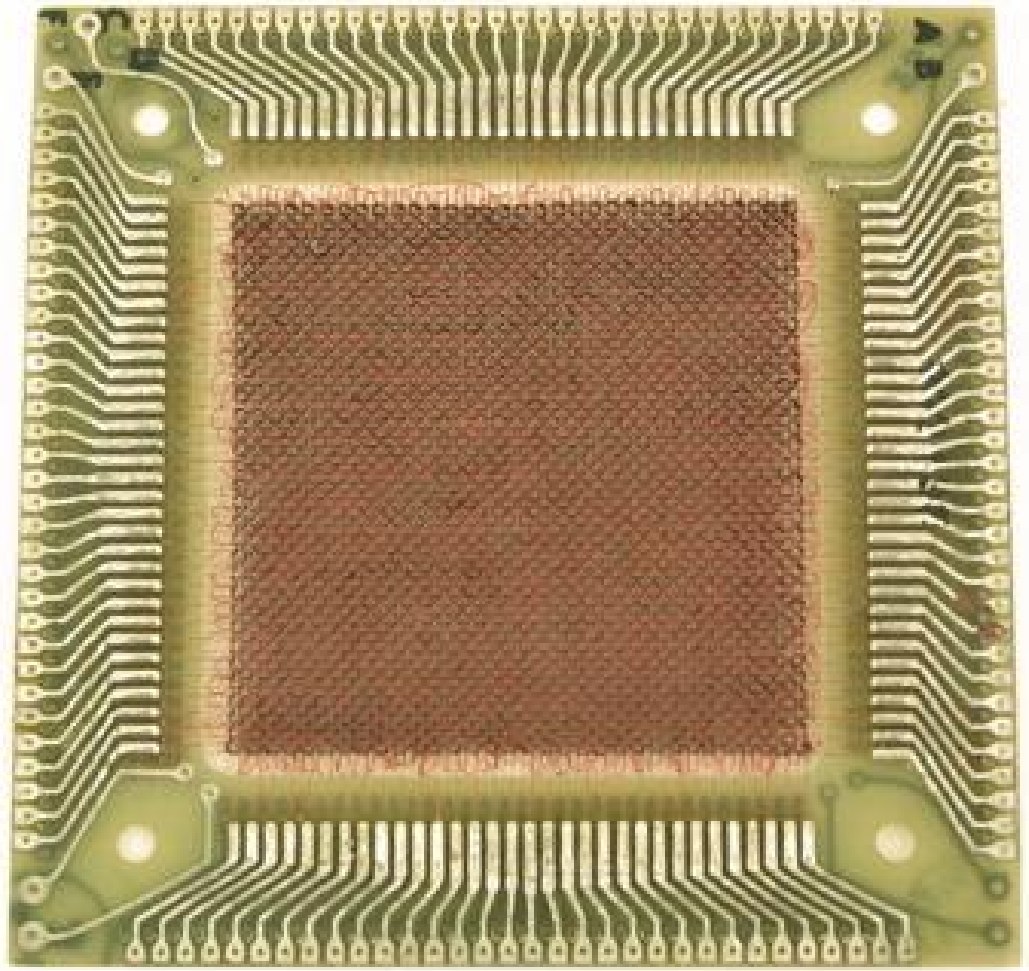
(1)半导体存储器

- 半导体存储器是用半导体器件组成的存储器。
- 根据制造工艺不同，可分为双极型和MOS型。

U盘

(2)磁表面存储器

- 磁存储器就是采用磁性材料制成的存储器。
- 磁存储器是利用磁性材料的的两个不同剩磁状态存放二进制代码“0”和“1”。早期有磁芯存储器，现多为磁表面存储器，如磁盘、磁带等。



磁芯存储器

3.5英寸软盘

硬盘

(3) 光存储器

- 利用光学原理制成的存储器，它是通过能量高度集中的激光束照在基体表面引起物理的或化学的变化，记忆二进制信息。如光盘存储器。

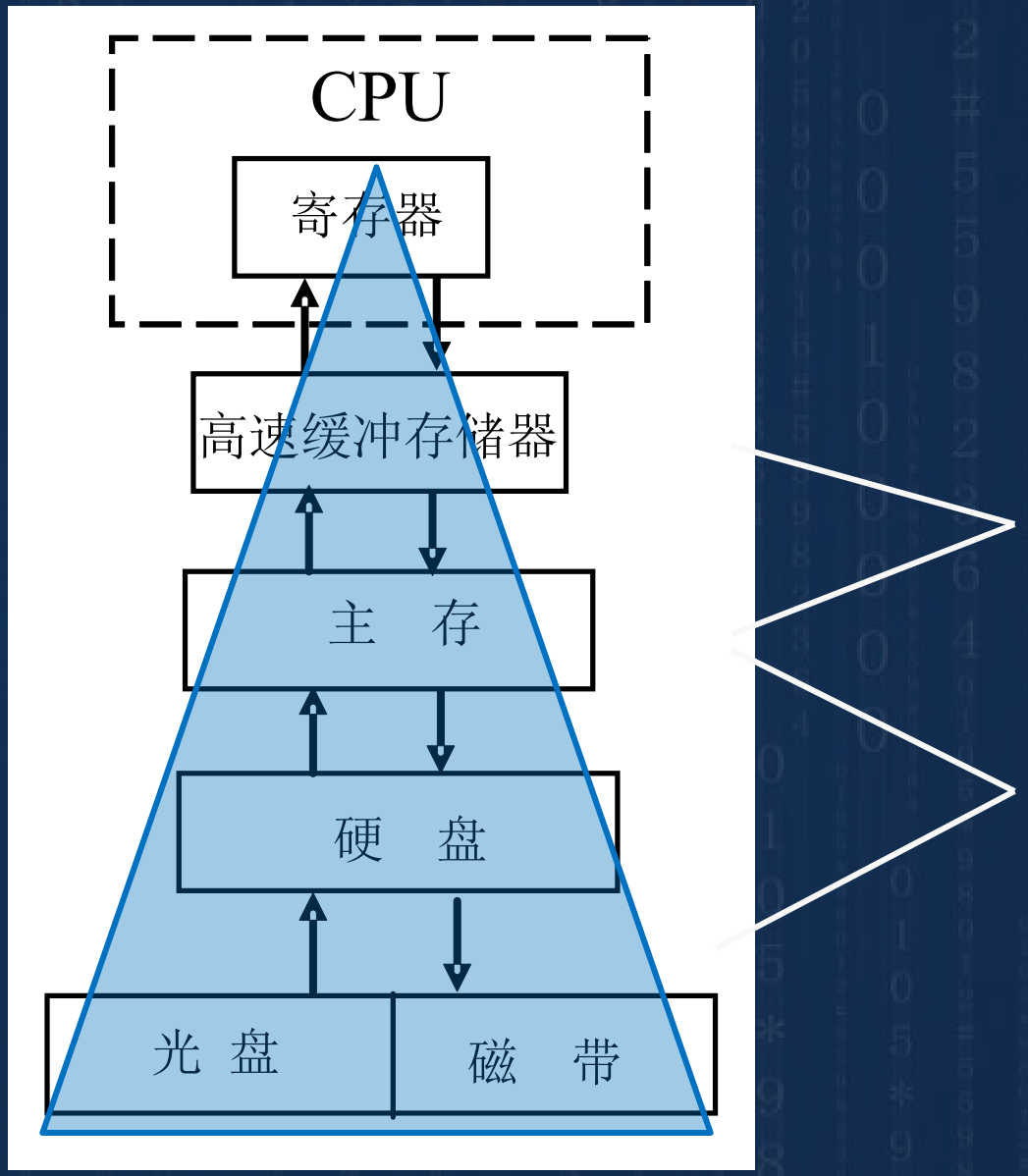
光盘和光驱

3.1.2 存储器系统的层次结构

- 不管主存储器的容量有多大，它总是无法满足人们的期望。其主要原因是，随着技术的进步，人们开始希望存放以前完全属于科学幻想领域的信息，存储器存储能力的扩大永远无法赶上需要它存放的信息的膨胀。

1. 存储系统的结构层次

- 最上层是CPU中的寄存器，其存取速度可以满足CPU的要求。下面一层是高速缓存，再往下是主存储器，然后是磁盘存储器，这是当前用于永久存放数据的主要存储介质。最后，还有用于后备存储的磁带、光盘存储器以及基于网络的各种文件系统。



We take advantage of the principle of locality by implementing the memory of a computer as a **memory hierarchy**. A memory hierarchy consists of multiple levels of memory with different speeds and sizes. The faster memories are more expensive per bit than the slower memories and thus are smaller.

Figure 5.1 shows the faster memory is close to the processor and the slower, less expensive memory is below it. The goal is to present the user with as much memory as is available in the cheapest technology, while providing access at the speed offered by the fastest memory.

The data are similarly hierarchical: a level closer to the processor is generally a subset of any level further away, and all the data are stored at the lowest level. By analogy, the books on your desk form a subset of the library you are working in, which is in turn a subset of all the libraries on campus. Furthermore, as we move away from the processor, the levels take progressively longer to access, just as we might encounter in a hierarchy of campus libraries.

A memory hierarchy can consist of multiple levels, but data are copied between only two adjacent levels at a time, so we can focus our attention on just two levels.

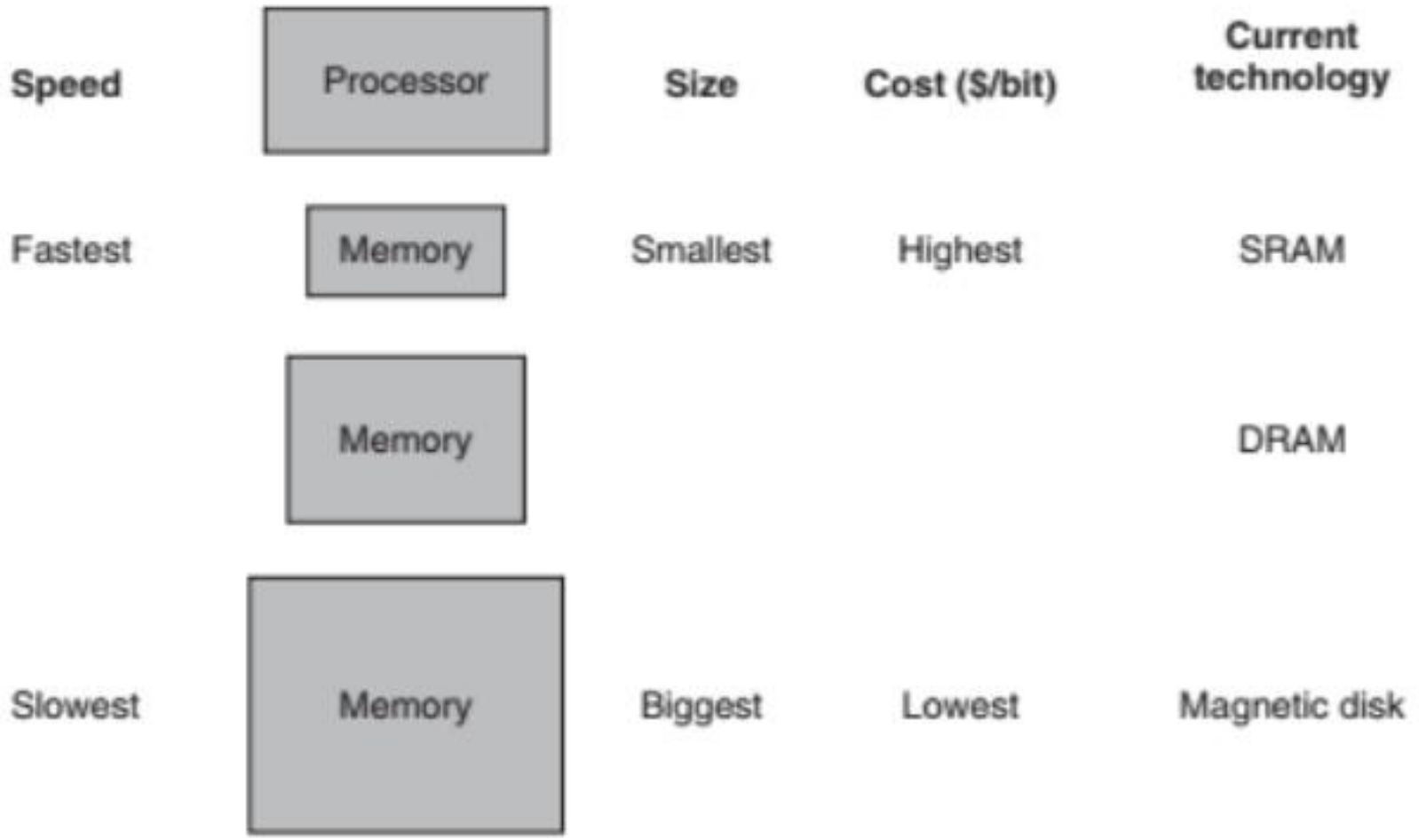


FIGURE 5.1 The basic structure of a memory hierarchy. By implementing the memory system as a hierarchy, the user has the illusion of a memory that is as large as the largest level of the hierarchy, but can be accessed as if it were all built from the fastest memory. Flash memory has replaced disks in many personal mobile devices, and may lead to a new level in the storage hierarchy for desktop and server computers; see Section 5.2.

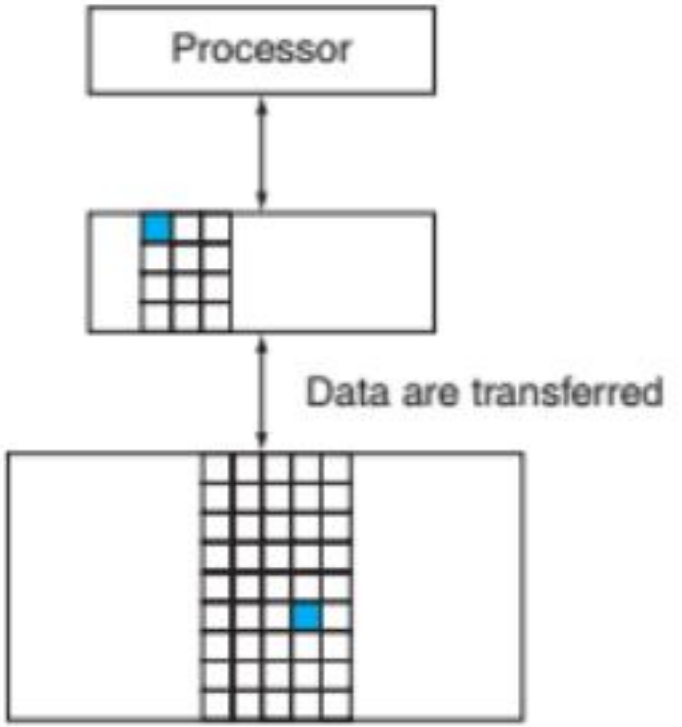


FIGURE 5.2 Every pair of levels in the memory hierarchy can be thought of as having an upper and lower level. Within each level, the unit of information that is present or not is called a *block* or a *line*. Usually we transfer an entire block when we copy something between levels.

The upper level—the one closer to the processor—is smaller and faster than the lower level, since the upper level uses technology that is more expensive. Figure 5.2 shows that the minimum unit of information that can be either present or not present in the two-level hierarchy is called a **block** or a **line**; in our library analogy, a block of information is one book.

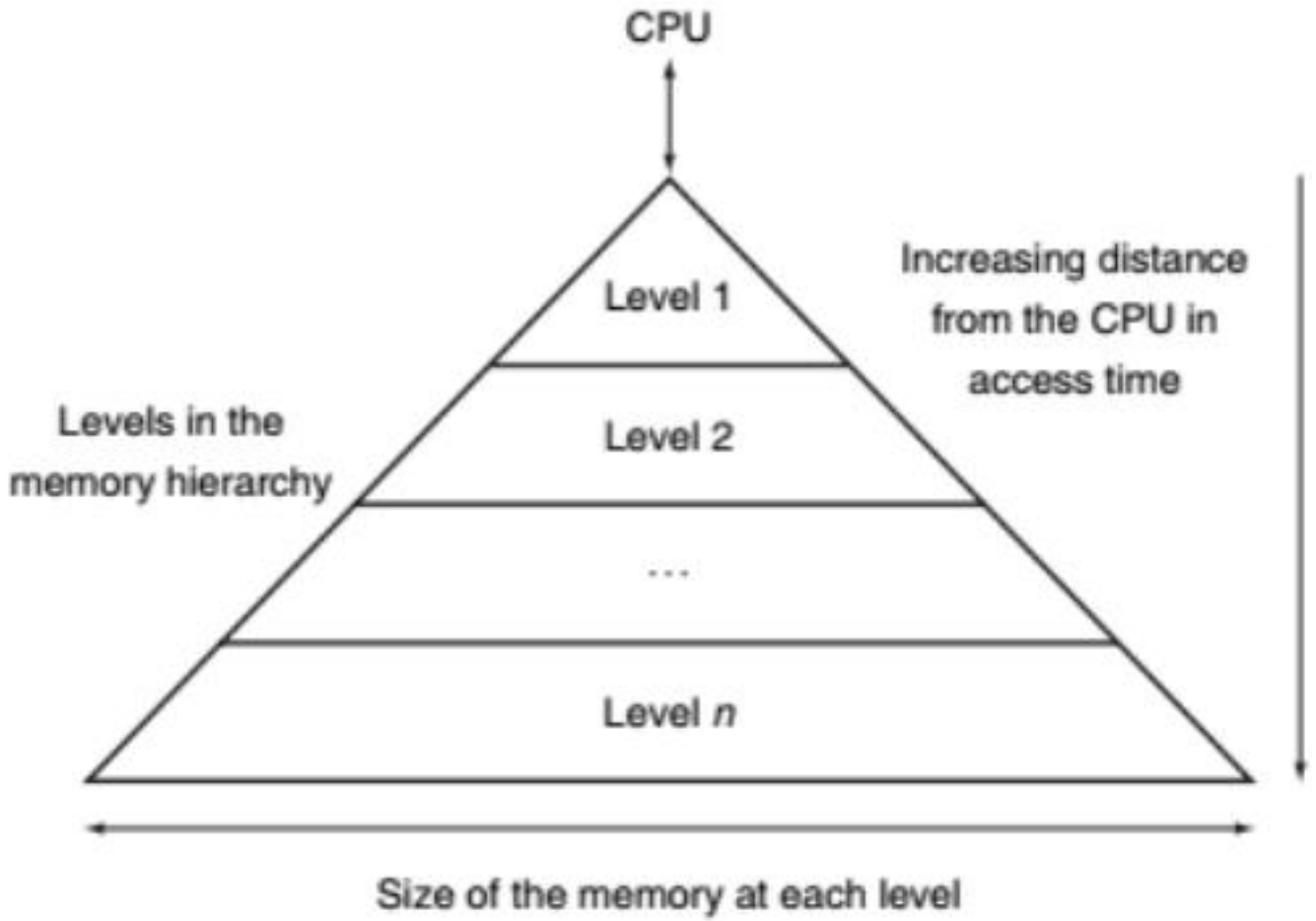


FIGURE 5.3 This diagram shows the structure of a memory hierarchy: as the distance from the processor increases, so does the size. This structure, with the appropriate operating mechanisms, allows the processor to have an access time that is determined primarily by level 1 of the hierarchy and yet have a memory as large as level *n*. Maintaining this illusion is the subject of this chapter. Although the local disk is normally the bottom of the hierarchy, some systems use tape or a file server over a local area network as the next levels of the hierarchy.

Programs exhibit both temporal locality, the tendency to reuse recently accessed data items, and spatial locality, the tendency to reference data items that are close to other recently accessed items. Memory hierarchies take advantage of temporal locality by keeping more recently accessed data items closer to the processor. Memory hierarchies take advantage of spatial locality by moving blocks consisting of multiple contiguous words in memory to upper levels of the hierarchy.

Figure 5.3 shows that a memory hierarchy uses smaller and faster memory technologies close to the processor. Thus, accesses that hit in the highest level of the hierarchy can be processed quickly. Accesses that miss go to lower levels of the hierarchy, which are larger but slower. If the hit rate is high enough, the memory hierarchy has an effective access time close to that of the highest (and fastest) level and a size equal to that of the lowest (and largest) level.

In most systems, the memory is a true hierarchy, meaning that data cannot be present in level i unless they are also present in level $i + 1$.

Which of the following statements are generally true?

1. Memory hierarchies take advantage of temporal locality.
2. On a read, the value returned depends on which blocks are in the cache.
3. Most of the cost of the memory hierarchy is at the highest level.
4. Most of the capacity of the memory hierarchy is at the lowest level.

存储器层次结构自上而下的特点

- (1) 访问时间逐渐增长
- 寄存器的访问时间是几个纳秒
- 高速缓存的访问时间是寄存器访问时间的几倍
- 主存储器的访问时间是几十个纳秒
- 磁盘的访问时间最少10ms以上
- 磁带和光盘的访问时间以秒来计量。

- (2) 存储容量逐渐增大
- 寄存器的容量约几到几百字节
- Cache约几百KB到若干MB
- 主存通常为若干GB
- 磁盘的容量为几百GB到若干TB
- 磁带和光盘一般脱机存放，其容量只受限于用户的预算。

- (3) 存储器每位的价格逐渐降低
- 例如
- 主存的价格约每兆字节几角
- 磁盘的价格是每兆字节几分或更低
- 磁带的价格是每G字节几元或更低

2.传统的三级存储结构

- Cache —— 主存层次
- 主要解决速度问题
- 通过辅助硬件，把主存和Cache构成统一整体，有接近Cache的速度、主存的容量和接近于主存的平均价格。
- 主存 —— 辅存层次
- 主要解决容量问题
- 大量的信息存放在大容量的辅助存储器中，当需要使用这些信息时，借助辅助软、硬件，自动地以页或段为单位成批调入主存中。

4.1.3 主存储器的组成和基本操作

- **1.**主存储器的组成

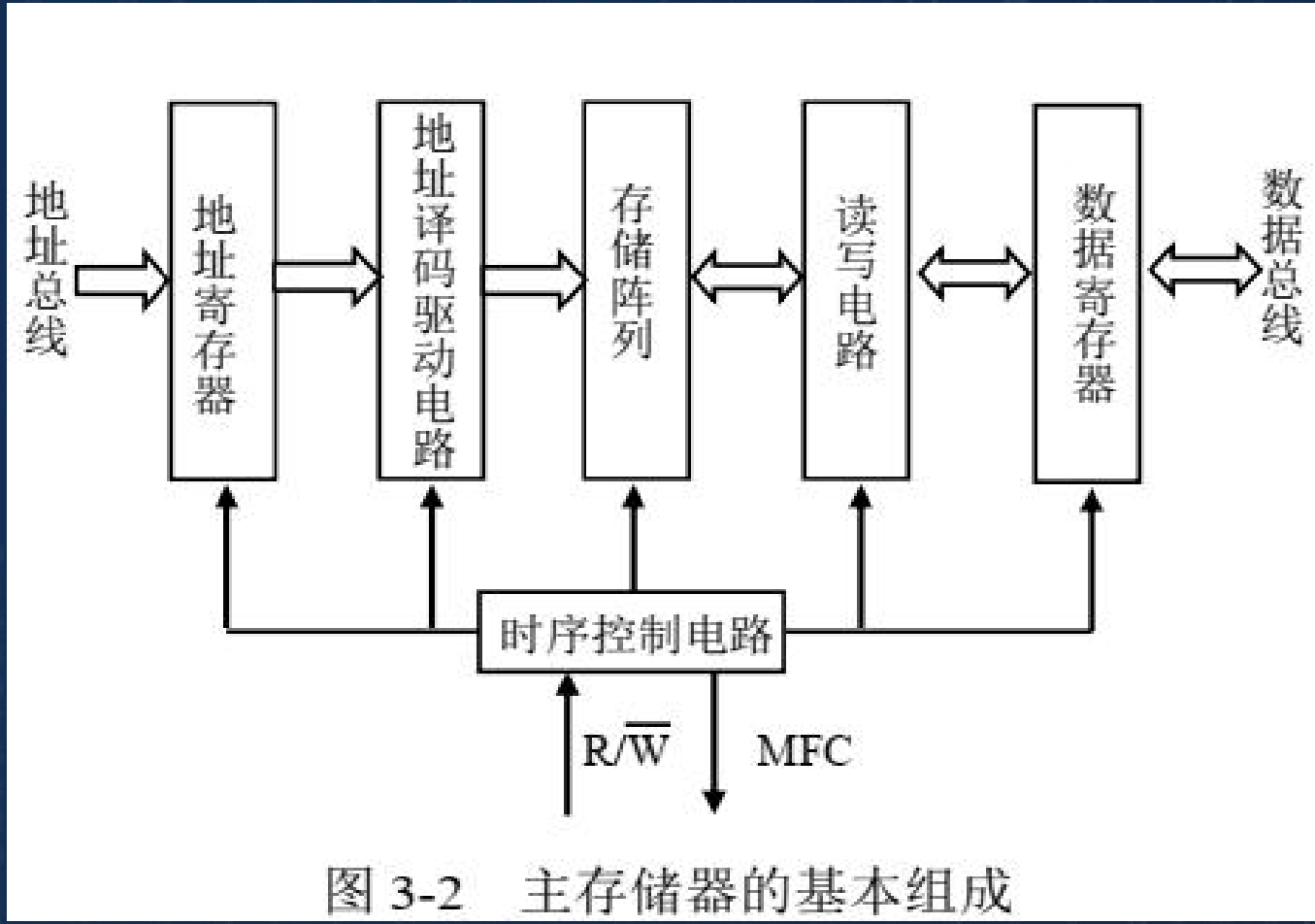


图 3-2 主存储器的基本组成

(1) 存储阵列（存储体）

- 存储阵列是存储器的核心部分，它是存储二进制信息的主体，也称为存储体。
- 存储阵列是由大量存储单元电路按一定的阵列形式排列起来构成的。
- 为了区分存储阵列中的各个存储单元，需要对它们进行统一编号，这个编号称为存储单元的地址。
- 因为地址采用二进制进行编码，所以又称为地址码。

存储单元的地址

- 存储单元的地址是存储体中每个存储单元被赋予的唯一的编号。
- 存储单元的地址用于区别不同的存储单元。
- 要对某一存储单元进行存取操作，必须首先给出被访问的存储单元的地址。

存储单元的编址

- 编址单位：存储器中可寻址的最小单位。
- ① 按字节编址：相邻的两个单元是两个字节。
- ② 按字编址：相邻的两个单元是两个机器字。

- 目前多数计算机是按字节编址的，即最小可寻址单位是一个字节。

- 例如一个32位字长的按字节寻址计算机，一个机器字中包含四个可单独寻址的字节单元。
- 当需要访问一个字，即需要同时访问4个字节时，可以按地址的整数边界进行存取。这时每个字的编址中最低2位的二进制数必须是“00”，这样可以由地址的低两位来区分不同的字节。

地址	11	10	01	00
0000	3	2	1	0
0100	7	6	5	4
1000	11	10	9	8
1100	15	14	13	12

- (2)地址寄存器：用于存放所要访问的存储单元的地址。要对某一单元进行存取操作，首先应通过地址总线将被访问单元地址存放到地址寄存器中。

- (3) 地址译码与驱动电路：用于对地址寄存器中的地址进行译码，通过对应的地址选择线到存储阵列中找到所要访问的存储单元，并提供驱动信号驱动其完成指定的存取操作。

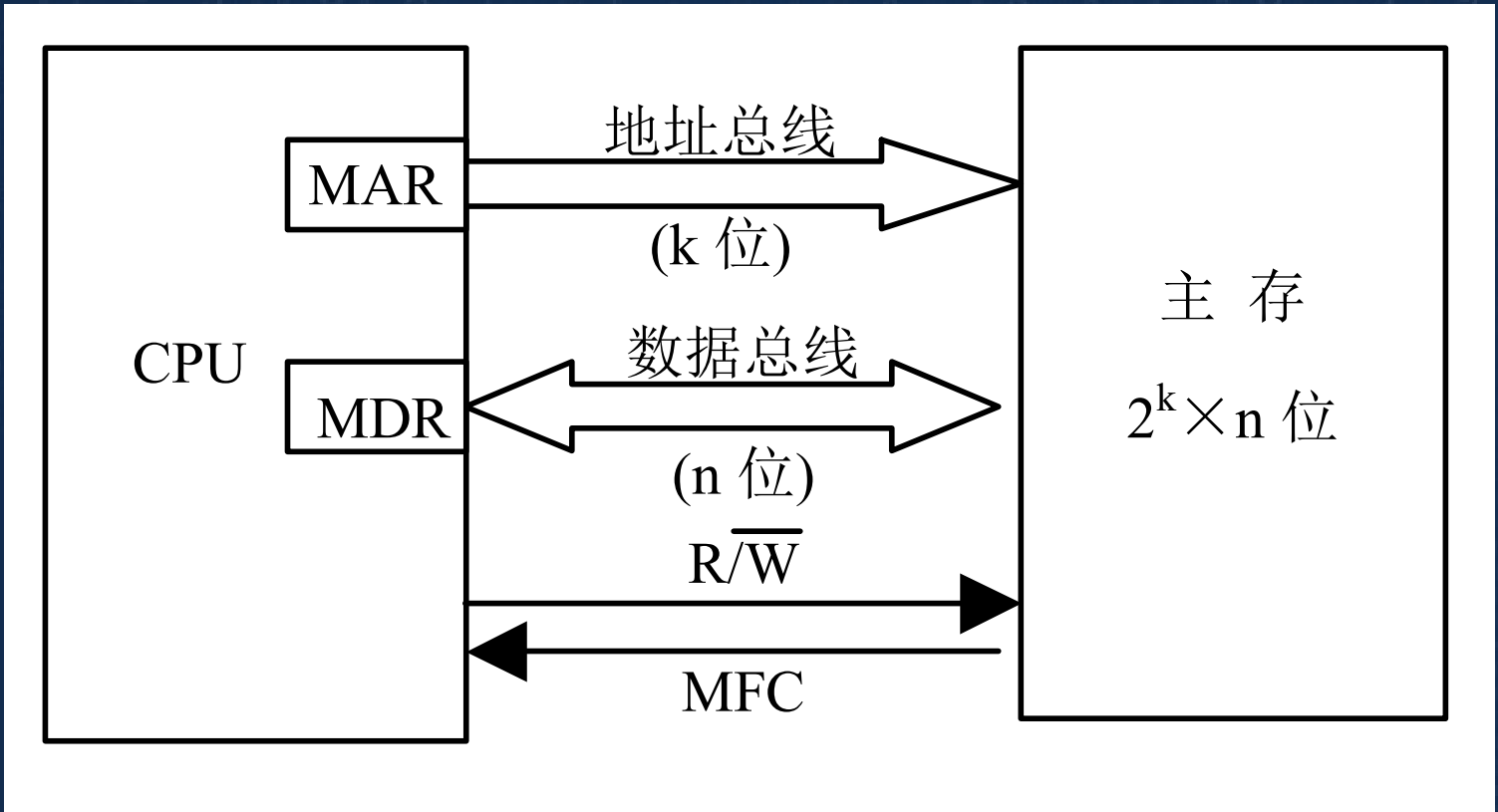
- (4)读写电路：根据CPU发出的读写控制命令，控制对存储单元的读写。

- (5) 数据寄存器：暂存需要写入或读出的数据。数据寄存器是存储器与计算机其它功能部件联系的桥梁。

- (6) 时序控制电路：用于接收来自CPU的读写控制信号，产生存储器操作所需的各种时序控制信号，控制存储器完成指定的操作。如果存储器采用异步控制方式，当一个存取操作完成后，该控制电路还应给出存储器操作完成（MFC）信号。

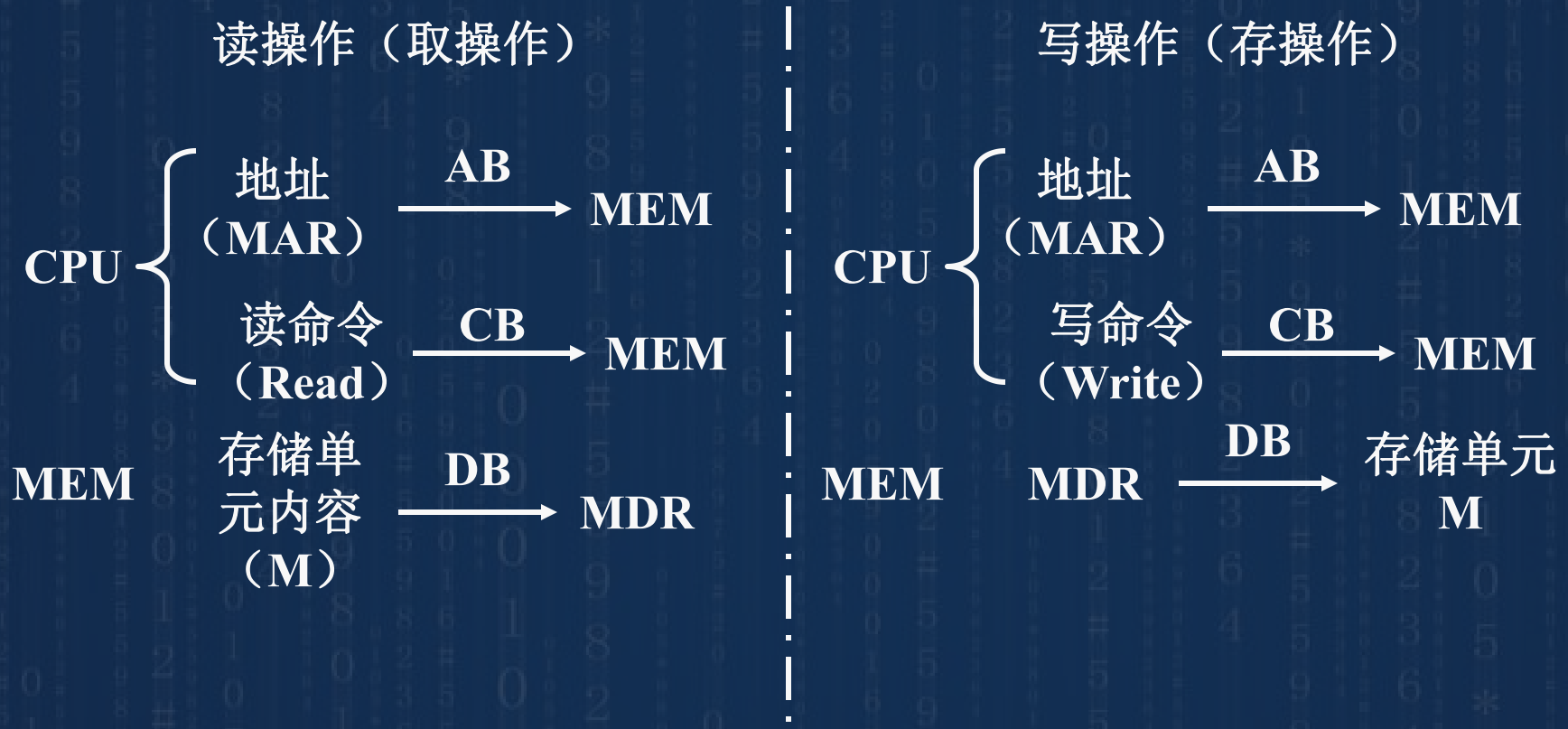
2. 存储器的基本操作

- 主存储器用于存放CPU正在运行的程序和数据。主存与CPU之间通过总线进行连接。



主存的操作过程

- MAR：地址寄存器　　MDR：数据寄存器



CPU与主存之间的数据传送控制方式

- 同步控制方式：数据传送在固定的时间间隔内完成，即在一个存取周期内完成。
- 异步控制方式：数据传送的时间不固定，存储器在完成读/写操作后，需向CPU回送“存储器功能完成”信号（MFC），表示一次数据传送完成。
- 目前多数计算机采用同步方式控制CPU与主存之间的数据传送。
- 异步传送方式允许选用具有不同存取速度的存储器作为主存 。

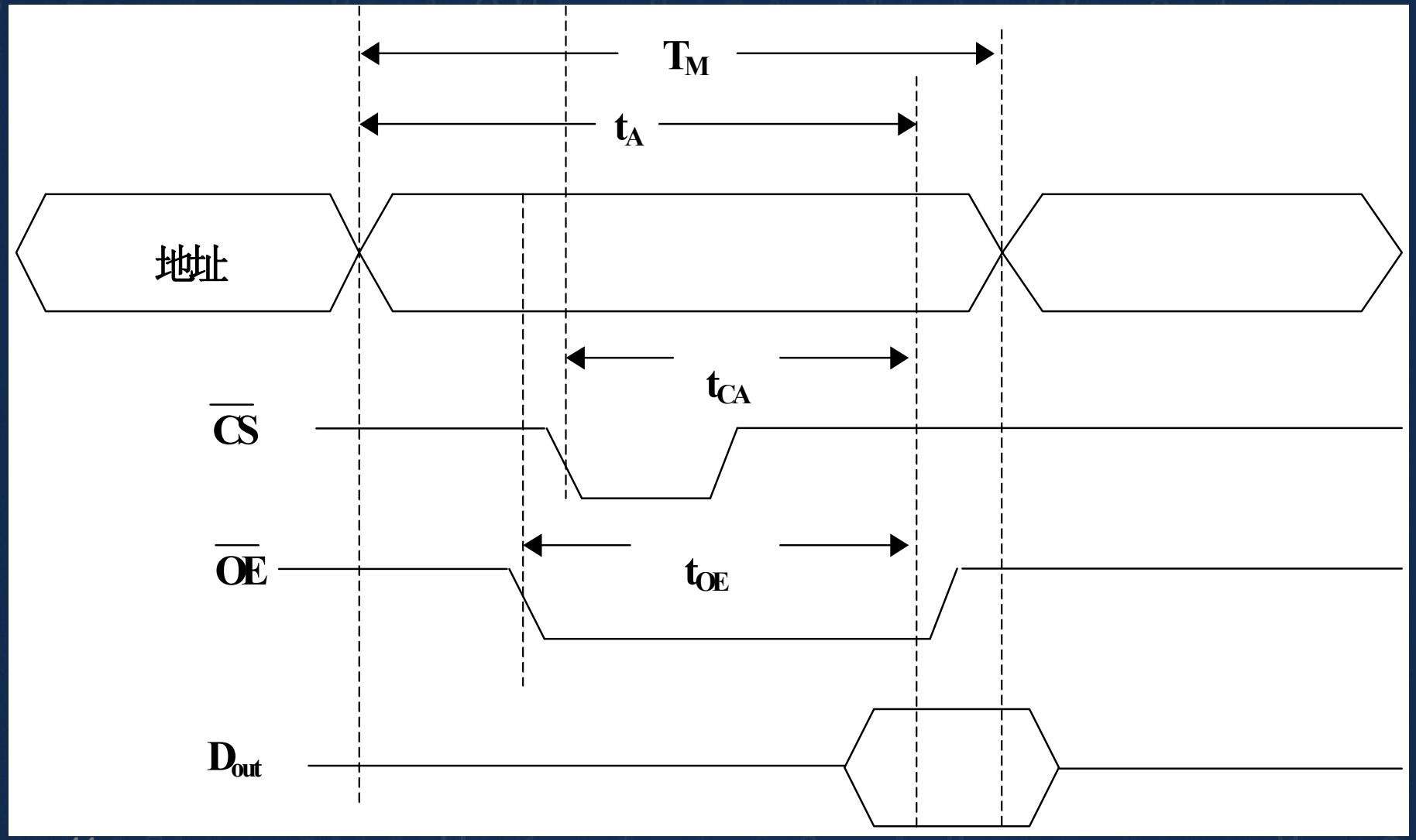
4.1.4 存储器的主要技术指标

- 衡量半导体存储器的主要技术指标：
- 1．存储容量：
- 半导体存储芯片所能存储的二进制信息的位数。
- 存储容量的表示：
- ① 存储芯片通常采用“位”来表示其容量。
  如：256Mbit。
- 有时也用存储单元数与每个单元的位数的乘积表示。如：512K×16位，表示存储芯片有512K个单元，每个单元为16位，共有8388608=2^{23}bit。
- ② 讨论计算机系统存储器的容量时，常用字节表示存储容量，例如4MB、16MB分别表示存储器中可容纳4兆和16兆个字节信息。

2. 速度

- 速度是存储芯片的一项重要技术指标。
- 由于存储芯片的工作速度慢于CPU的工作速度，所以存储芯片的工作速度直接影响着CPU执行指令的速度。
- (1) 访问时间（取数时间 t_A）
- 从启动一次存储器存取操作到完成该操作所经历的时间。
- 即从存储器接到CPU发出的读/写命令和地址信号到数据读入MDR/从MDR写入MEM所需的时间。
- 读出时间：从存储器接到有效地址开始到产生有效输出所需的时间。
- 写入时间：从存储器接到有效地址开始到数据写入被选中单元为止所需的时间。

- 与t_A相关的参数：
- t_{CA}：指从加载到存储器芯片上的（$\overline{CS}$）引脚上的选片信号有效开始，直到读取的数据或指令在存储器芯片的数据引脚上可以使用为止的时间间隔。
- t_{OE}：对于某些ROM芯片，指从读信号（$\overline{OE}$）有效开始，直到读取的数据或指令在存储器芯片的数据引脚上可以使用为止的时间间隔。


T_M
t_A
地址
t_{CA}
$\overline{CS}$
$\overline{OE}$
t_{OE}
D_{out}

- (2) 存取周期（存储周期、读写周期　T_M）
- 对存储器连续进行两次存取操作所需要的最小时间间隔。
- 由于存储器进行一次存取操作后，需有一定的恢复时间，所以通常存储周期T_M大于访问时间t_A。
- 半导体存储器的存取周期T_M
- $T_M = t_A$＋一定的恢复时间
- MOS型存储器的T_M约100ns
- 双极型TTL存储器的T_M约10ns

3. 存储器总线带宽

- 带宽是指存储器单位时间内所存取的二进制信息的位数。
- 带宽也称存储器数据传输率、频宽B_m
- 带宽的单位：位/秒、字节/秒、兆字节/秒

带宽的计算

- 1. 带宽＝每个存取周期访问的位数/存取周期。
- 例如，存取周期为500ns，每个存取周期可访问16位二进制数据，则它的带宽为32Mb/s

- 2. 带宽＝存储器总线宽度/存取周期。

$$B_m = \frac{W}{T_M}$$

- W：存储器总线的宽度，对于单体存储器，W就是数据总线的根数。

提高存储器速度的途径

- ① 采用高速器件
- ② 减少存取周期T_M，如引入Cache。
- ③ 提高总线宽度 W，如采用多体交叉存储方式。
- ④ 采用双端口存储器。
- ⑤ 加长存储器字长。

4. 价格

- 存储器的价格常用每位的价格来衡量。
- 设存储器容量为S位，总价格为$C_{总}$，每位价格为c
- $$c = C_{总}/S$$
- $C_{总}$不仅包含存储器组件本身的价格，也包括为该存储器操作服务的外围电路的价格。
- 存储器的总价格与存储容量成正比，与存储周期成反比。
- 除上述几个指标外，影响存储器性能的因素还有功耗、可靠性等。

*Memory Technologies

There are four primary technologies used today in memory hierarchies. Main memory is implemented from DRAM (*dynamic random access memory*), while levels closer to the processor (caches) use SRAM (*static random access memory*). DRAM is less costly per bit than SRAM, although it is substantially slower. The price difference arises because DRAM uses significantly less area per bit of memory, and DRAMs thus have larger capacity for the same amount of silicon; the speed difference arises from several factors described in Section A.9 of Appendix A. The third technology is flash memory. This nonvolatile memory is the secondary memory in Personal Mobile Devices. The fourth technology, used to implement the largest and slowest level in the hierarchy in servers, is magnetic disk. The access time and price per bit vary widely among these technologies, as the table below shows, using typical values for 2012.

Memory technology	Typical access time	$ per GiB in 2012
SRAM semiconductor memory	0.5–2.5 ns	$500–$1000
DRAM semiconductor memory	50–70 ns	$10–$20
Flash semiconductor memory	5,000–50,000 ns	$0.75–$1.00
Magnetic disk	5,000,000–20,000,000 ns	$0.05–$0.10

见Pp371