

# 第 5 章 存储系统

本章学习内容：

- ( 1 ) 主存储器的组成与结构。
- ( 2 ) 存储系统的三层结构：Cache 主存和辅存
- ( 3 ) Cache 及虚拟存储器的工作原理。
- ( 4 ) 辅存 ( 磁表面存储器、光盘 ) 的存储原理与读 / 写过程。

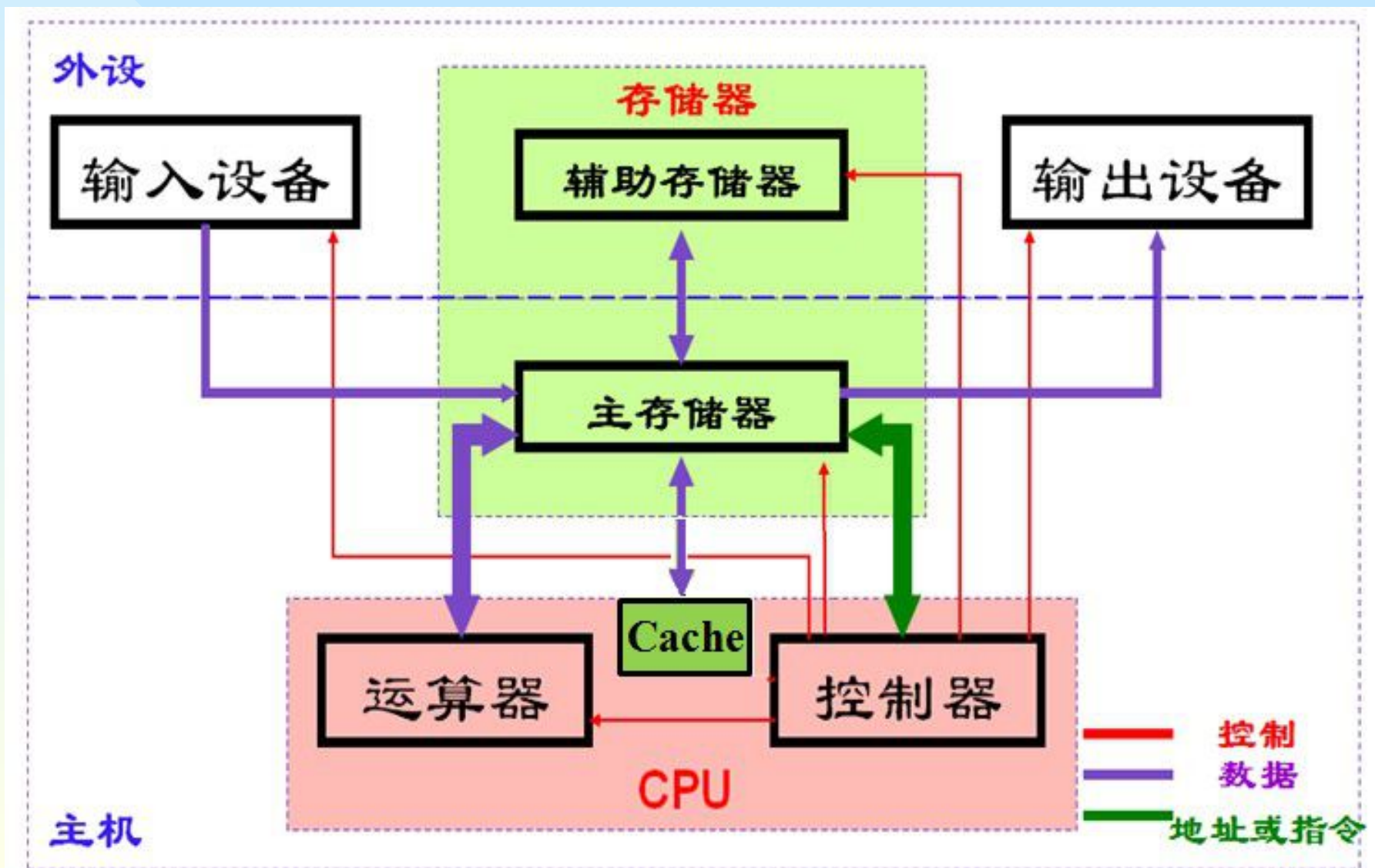
概念解释：

**存储器**：计算机系统的重要组成部分，用来存放程序和数据。

有了存储器，计算机就具有**记忆能力**，因而能自动地进行操作。

**存储系统**：由存放程序和数据的各类**存储设备**及管理存储器的有关**软件**所构成。

# 存储器



# 存储器

在现代计算机中，存储器处于全机**中心地位**，其原因是：

**(1)** 当前计算机正在执行的程序和数据（除了暂存于CPU寄存器的）均存放在存储器中。

**(2)** 计算机系统中输入输出设备数量增多，数据传送速度加快，因此采用了DMA技术和I/O通道技术等，在存储器与输入输出系统之间**直接**传送数据。

CPU  $\longleftrightarrow$  主存  $\longleftrightarrow$  外设

# 存储器

(3) 共享存储器的**多处理机**的出现，利用存储器存放**共享数据**，并实现处理机之间的通信，更加强了存储器作为全机中心的地位。

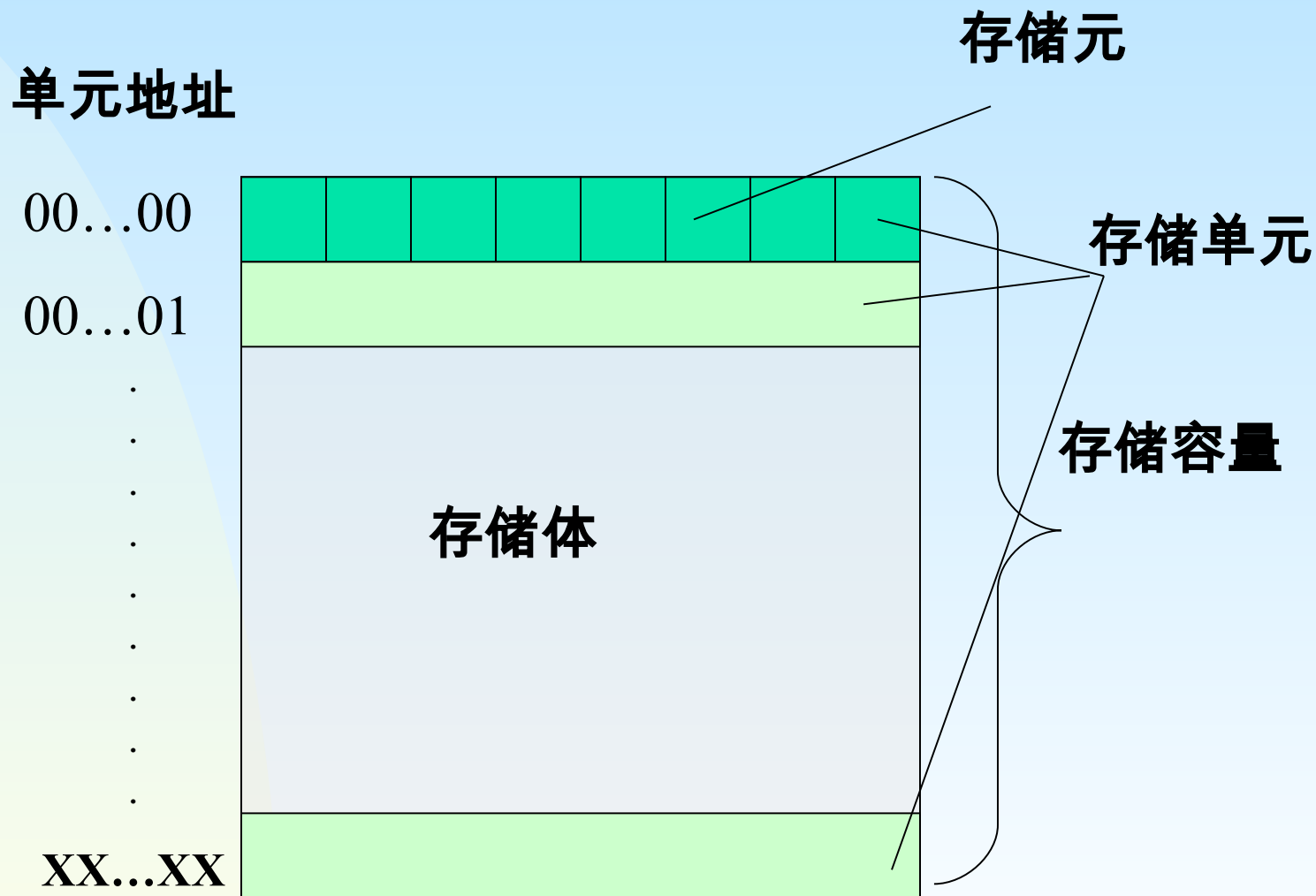
(4) 由于 CPU 都是由高速器件组成，不少指令的执行速度基本上取决于主存储器的速度。

与主存储器的技术发展密切相关：

计算机解题能力的提高；系统软件的日益丰富；

应用软件等。

# 5.1 存储系统的组成



## 几个基本概念：

**存储元**：存储器的**最小组成单位**，用以存储 1 位二进制代码。

**可编址的最小单位**

**存储单元**：是 CPU 访问存储器的**基本单位**，由若干个具有相同操作属性的存储元组成。

**单元地址**：在存储器中用以标识存储单元的**唯一编号**，CPU 通过该编号访问相应的存储单元。



## 5.1.1 存储器分类

1. 按存储介质分类
2. 按存取方式分类
3. 按存储器在计算机系统中的功能分类
4. 按信息的可保存性分类



# 1. 按存储介质分类

存储介质：

- 1) 明显具有并能保持两种稳定状态；
- 2) 能够方便的与电信号进行转换（读出与写入）

分类：

- 1) 半导体存储器——主存普遍采用
- 2) 磁性材料存储器
- 3) 光介质存储器

(1) 半导体存储器

TTL 、 MOS

易失

(2) 磁表面存储器

磁头、载磁体

(3) 磁芯存储器

硬磁材料、环状元件

(4) 光盘存储器

激光、磁光材料

非  
易  
失



**常用概念：**

**非易失性存储器 / 不挥发性存储器：**断电后能保存信息。

**破坏性读出：**读出信号后，破坏了原来的状态。

**再生 / 重写：**将读出的信息重新写回存储介质，恢复读出前的状态。

## ① 半导体存储器——主存普遍采用

**静态存储器**：需要电源才能工作，断电信息将会丢失，是挥发性存储器（或易失性）。

速度快、功耗大，集成度低；适于作小容量快速存储器，如 Cache 或寄存器组；

**动态存储器**：需要动态刷新。

- 1) 依靠**电容**上的**存储电荷**暂存信息，时间一长电荷会**泄露**，需要**定时刷新**内容。
- 2) 破坏性读出，需要重写。
- 3) 集成度最高，适于作大容量主存。

## ② 磁性材料存储器

**存储体结构**：在金属（硬盘）或塑料（软盘）基体上，涂一层很薄的**磁性材料**作为记录载体。如磁盘，磁带，磁卡等。

**写入原理**：写‘1’和‘0’时分别在读写线圈中通入不同方向的电流，磁头相对磁层运动时在磁层表面留下不同方式的**磁化单元**，分别代表‘1’和‘0’。

**读出原理**：磁头相对磁层运动时，磁层表面不同方式的磁化单元在读写线圈感应出不同的电动势，分别为‘1’和‘0’。

**特点**：

- 1) 容量大，每位价格低；
- 2) 不挥发性，非破坏性读出；
- 3) 存取速度低（读写时介质作机械运动）；
- 4) 广泛用作外存，为虚拟存储提供物理基础。

### ③ 光介质存储器

**读信息：**聚焦激光束，光反射原理。  
信息都以坑点形式分布。

**特点：**

- 1) 记录密度高、存储容量大；
- 2) 信息保存寿命长、工作稳定可靠；
- 3) 环境要求低等。

## 2. 按存取方式分类

### ① 存取时间与物理地址无关（随机访问）

- 随机存储器      在程序的执行过程中 可读 可 写
- 只读存储器      在程序的执行过程中 只 读

### ② 存取时间与物理地址有关（串行访问）

- 顺序存取存储器      磁带
- 直接存取存储器      磁盘



## (1) 顺序存取存储器 (SAM-Sequential Access Memory)

**特点：**信息只能按某种顺序存放或读出，存取时间取决于信息的存放位置。

**磁带存储器：**是一种顺序存储器，磁带正向或反向走带，顺序寻找并顺序读出或写入，时间视距离而定。

**优缺点：**存储容量大；每位价格低；存取速度慢。  
仅适用于外存。



## (2) 随机存取存储器 (RAM-Random Access Memory)

### 特点：

- 1) CPU 或 I/O 设备在任一时刻都可按地址**随机**地访问任一存储单元；
- 2) 访问各存储单元所需的读写时间相同，与地址无关。

### 说明：

- 1) 半导体存储器一般属于 RAM 存储器；
- 2) 主存与 Cache 采取随机访问的存取方式，CPU 可直接编程访问；
- 3) 分为 SRAM、DRAM 两种。

### (3) 直接存取存储器 (DAM-Direct Access Memory)

**特点：**

- 1) 先选择存取信息的**较小局部区域**，然后用**顺序方式**存取；
- 2) 存取方式介于 RAM 和 SAM 之间。

**例：**磁盘，先寻找信息所在的扇区，然后再顺序存取信息。

**说明：**此类存储器，

- 1) 容量比较大；
- 2) 速度则介于 SAM 和 RAM 之间；
- 3) 主要用作辅存。

#### (4) 只读存储器 (ROM-Read Only Memory)

**特点：** 存储器的内容只能读出而不能写入。

**应用：**

- 1) 有的 ROM 位于主存中特定区域，固化系统软件的核心部分，如 IBM-PC 机中 BIOS, 中断向量表等。
- 2) CPU 中用 ROM 固化微程序；
- 3) 汉字字库——根据字符编码输出字符形状的点阵代码，如汉卡、汉字设备等；
- 4) 访问方式和 RAM 一样按地址访问；

**说明：** 也有的 ROM 用作辅存，采用直接访问方式，如 CDROM。

# 半导体只读存储器

根据只读存储器的工艺，可分为：

- 1) **MROM (Masked ROM)**：固定掩模型 ROM，存储的信息由生产厂家在加工过程中“写入”，用户不能修改。
- 2) **PROM**：一次编程写入型 ROM，允许用户用特定的编程器向 ROM 中写入数据，写入后，不能修改。
- 3) **EPROM**：（紫外线）可擦除可编程型 ROM。
- 4) **EEPROM (E<sup>2</sup>PROM - Electrically Erasable and Programmable Rom)**：电擦除可编程型 ROM。

# 闪存存储器 (Flash Memory)

一种高密度、非易失性的读 / 写半导体存储器，突破了传统的存储器体系，改善了现有存储器的特性。

## 特点：

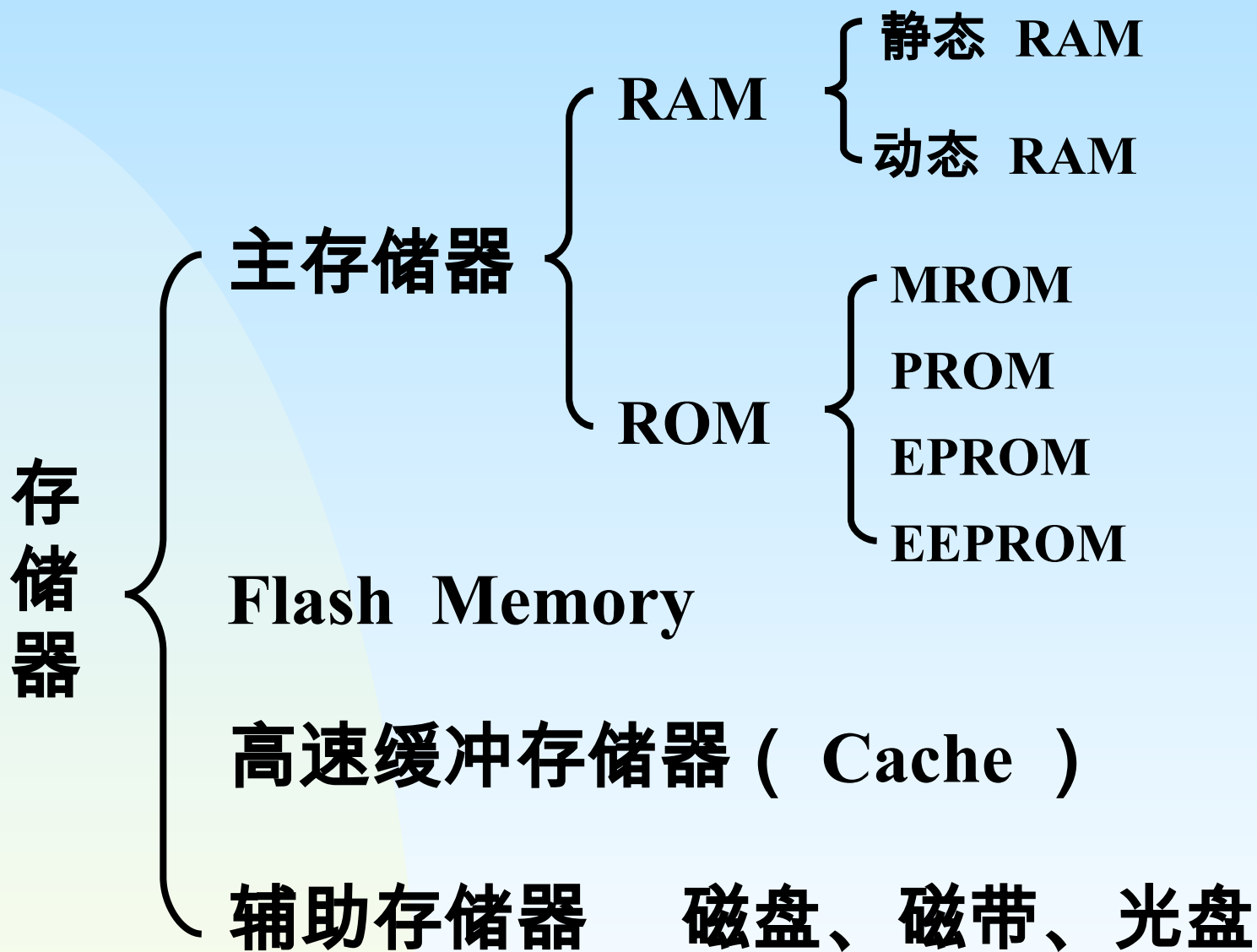
- (1) 固有的非易失性
- (2) 廉价的高密度
- (3) 可直接执行

## 电擦除和重新编程能力：

闪存存储器是在 EPROM 功能基础上增加了电路的电擦除和重新编程能力。

### 3. 按存储器在计算机系统中的作用分类

- (1) 高速缓冲存储器 (Cache)
- (2) 主存储器
- (3) 辅助存储器 ( 外存储器 )
- (4) 控制存储器 (CM)



## (1) 高速缓冲存储器 (Cache)

- 1) 目前由**双极型半导体**组成，构成计算机系统中的一个高速小容量存储器；
- 2) 存取速度接近 CPU 的工作速度，用来临时存放指令和数据。

## (2) 主存储器

- 1) 用来存放计算机**运行时**的大量程序和数据；
- 2) 一般用 **MOS 半导体存储器**构成。

**说明：** CPU 能直接访问的存储器称**内存**；  
**高速缓存**和**主存**都是内存。



### (3) 辅助存储器 ( 外存储器 )

1) 外存储器主要由磁表面存储器组成；

3) 外存储器的内容需调入主存后才能被 CPU 访问

特点：容量大，可存放大量的程序和数据。

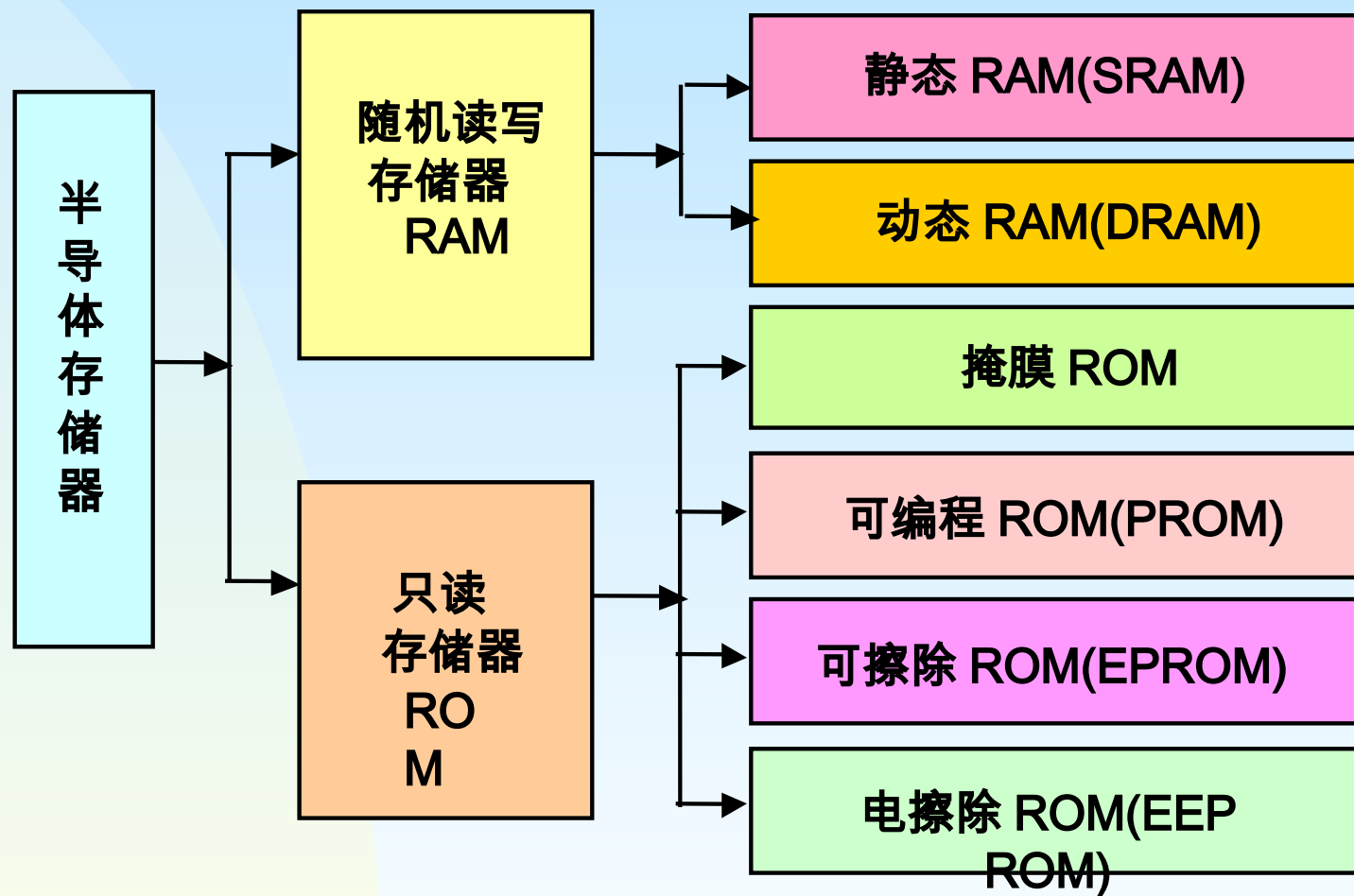
### (4) 控制存储器 (CM)

CPU 中固化微程序的存储器。

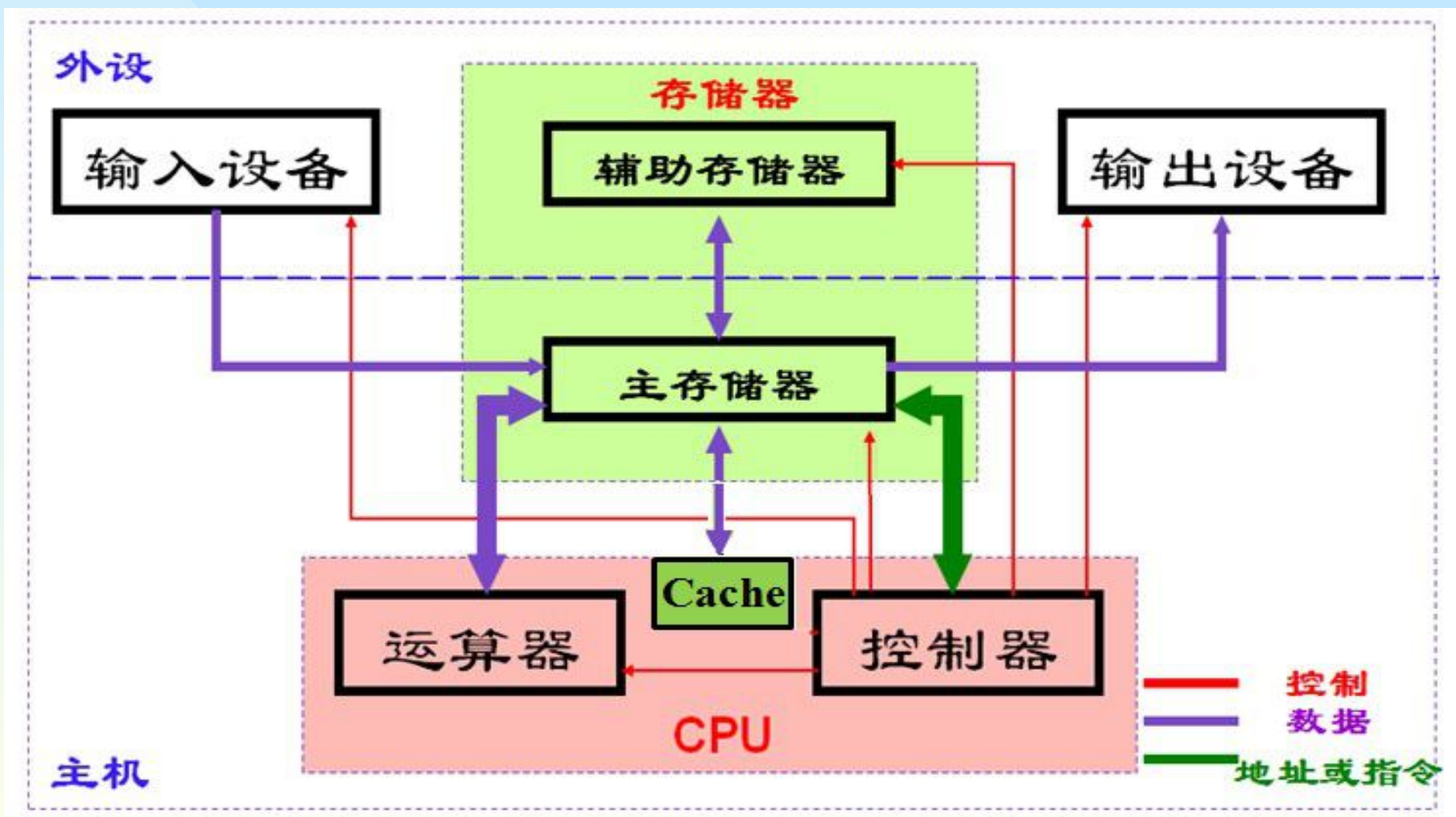
## 4. 按信息的可保存性分类

- (1) **挥发性 / 易失性存储器**：断电后信息即消失的存储器。
- (2) **非挥发性 / 非易失性存储器**：断电后仍能保存信息的存储器。

# 半导体存储器

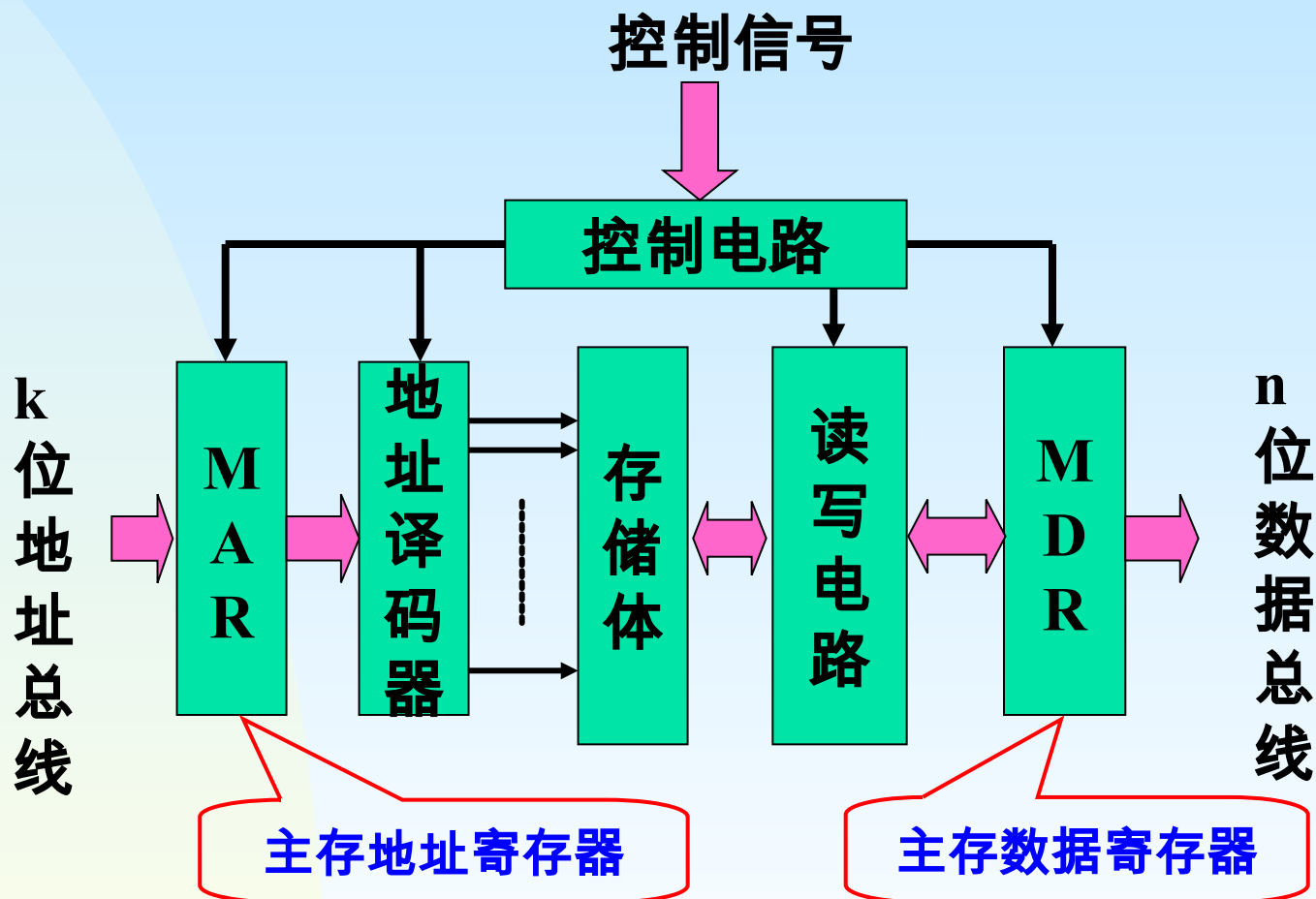


## 5.1.2 存储器系统层次结构



## 5.2 主存储器

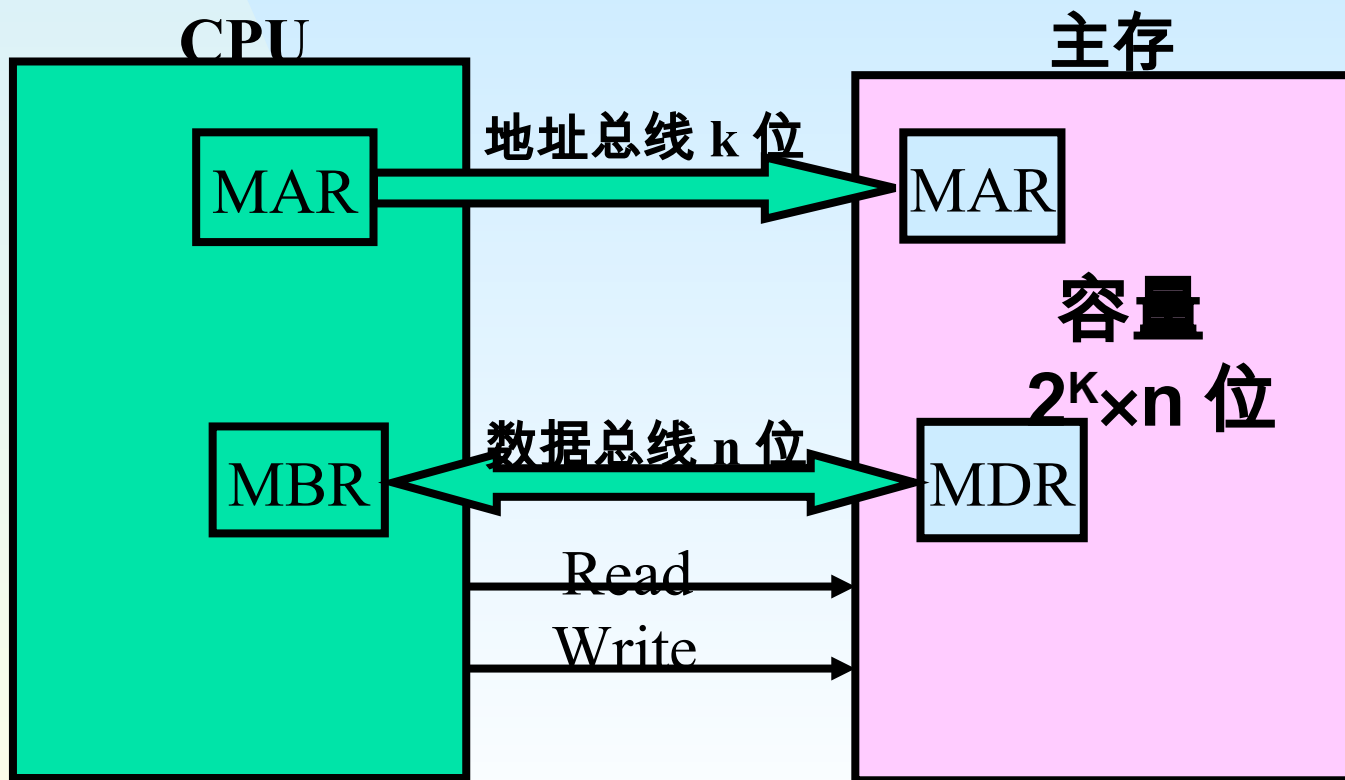
### 主存储器的基本结构：



# 主存储器的基本操作：

主存储器与 CPU 的关系密切：存储 CPU 正在执行的指令和使用的数据；

CPU 通过使用 **MAR**( 主存地址寄存器 ) 和 **MBR**( 主存缓冲寄存器 ) 经由总线与主存进行数据传送。



寄存器与寄存器由物理总线直接相连  
—— 接口



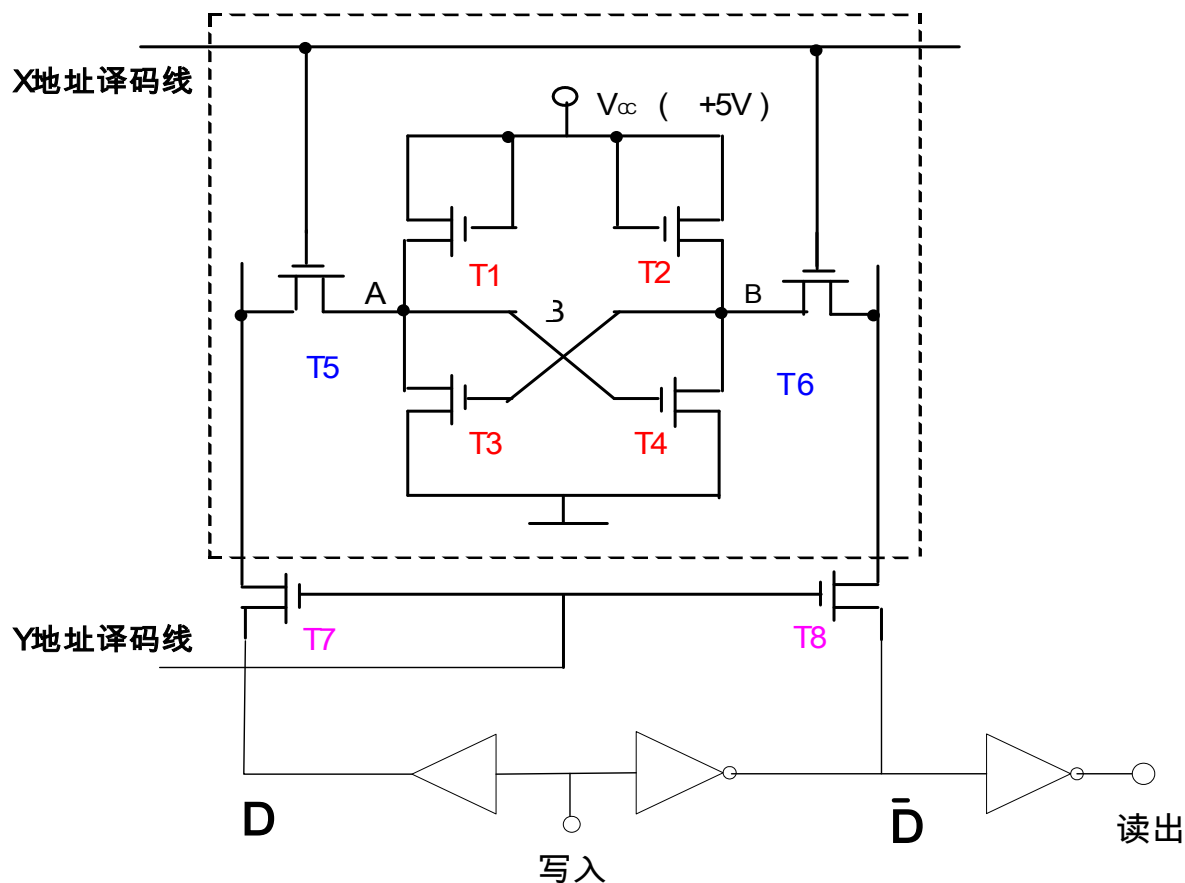
# 主存储器的主要性能指标

主要性能指标有：

- 1) **存储容量**：能存放的二进制位数或字节数；
- 2) **速度**：存取时间、存储周期和存储器带宽；
- 3) **价格**：每位价格；
- 4) **可靠性**：通常用**平均无故障时间** MTBF(Mean Time Between Failures) 来表征。
- 5) **功耗**：包括**维持功耗**和**操作功耗**。
- 6) **集成度等**。指在一片数平方毫米的芯片上能集成多少个存储元，每个存储元存储一个二进制位；  
集成度常表示为**位 / 片**。

## 5.3 半导体 RAM 芯片

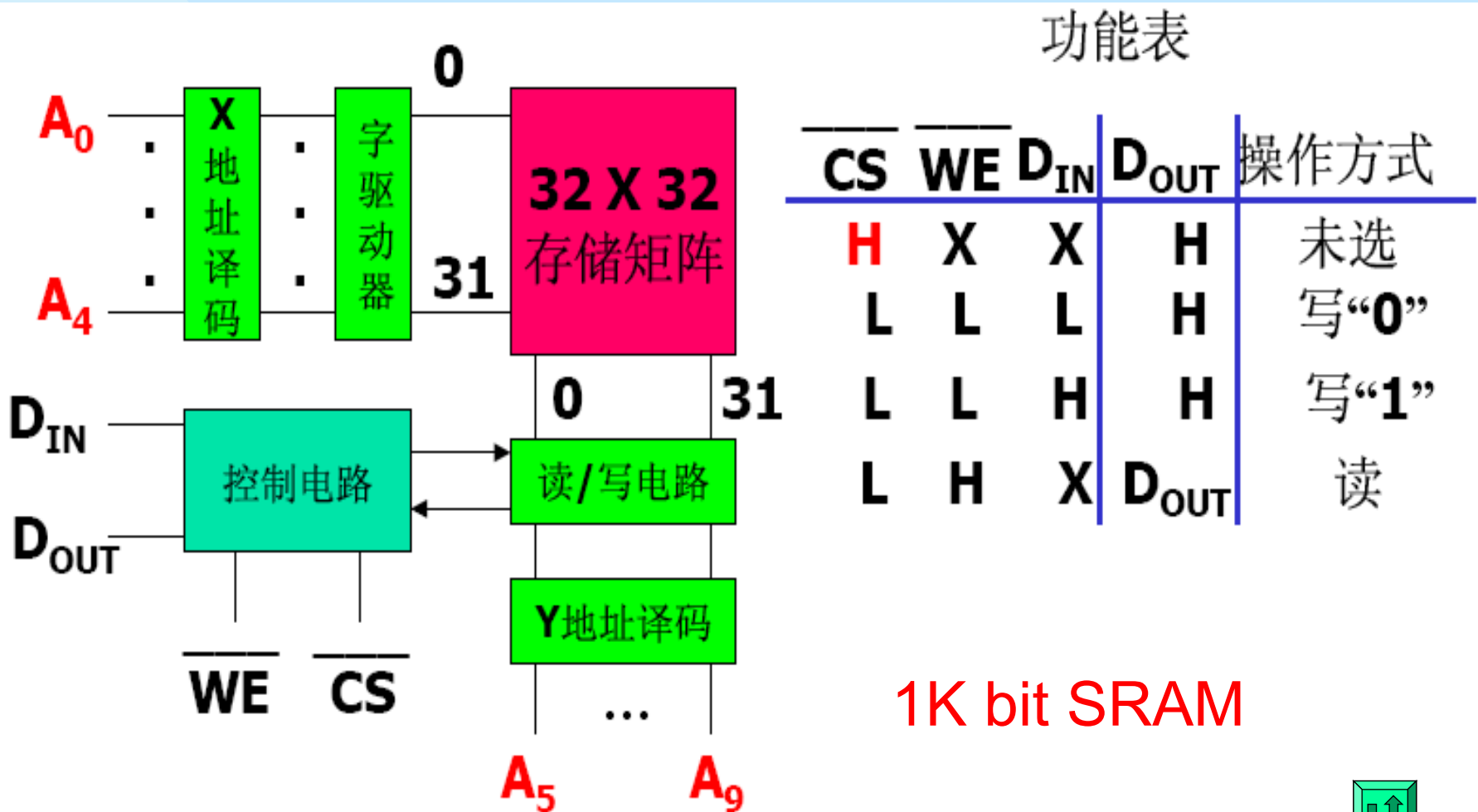
**基本存储元**：存储器的最基本的存储元件，它用来存储一位二进制信息。如六管静态存储单元、动态 MOS 存储单元等。



六管基本存储单元电路



**存储矩阵**：一个**基本存储元**存放一位二进制信息，  
存储器中将许多**基本存储元**按一定的顺序排列成阵列形式，称为**存储矩阵**。



## 地址译码器

二进制表示的地址→为译码输出线上的高电位，选择相应的存储单元。

### 地址译码方式：

**单译码方式**：适用于小容量存储器，只有一个译码器。

**双译码方式**，适用于容量较大的存储器，也称矩阵译码器。**可有效减少选择线的数目。**

分为 X 和 Y 两个译码器。每一个译码器有 **n** 个输入端，可以译出  **$2^n$**  个状态，两译码器交叉译码的结果，可产生  $2^n \times 2^n = \mathbf{2^{2n}}$  个输出状态。



## 驱动器

驱动挂在一条行选择线上的所有存储元电路（负载较大）。

## 读 / 写电路

控制被选中的单元读出或写入，并具有放大信息的作用。

## 片选控制

在地址选择时，首先要选片，只有当片选信号有效时，地址线才有效。

## 控制电路

根据 CPU 给出的信号是读命令还是写命令，控制被选中存储单元的读写。



半导体存储器芯片容量表示：字数×位数

**字**：编址单元，有独立统一的地址；

**位数**：一个字里的二进制的位数。

**例**：一片 62256(SRAM) 容量为  $32K \times 8$ ，表示，

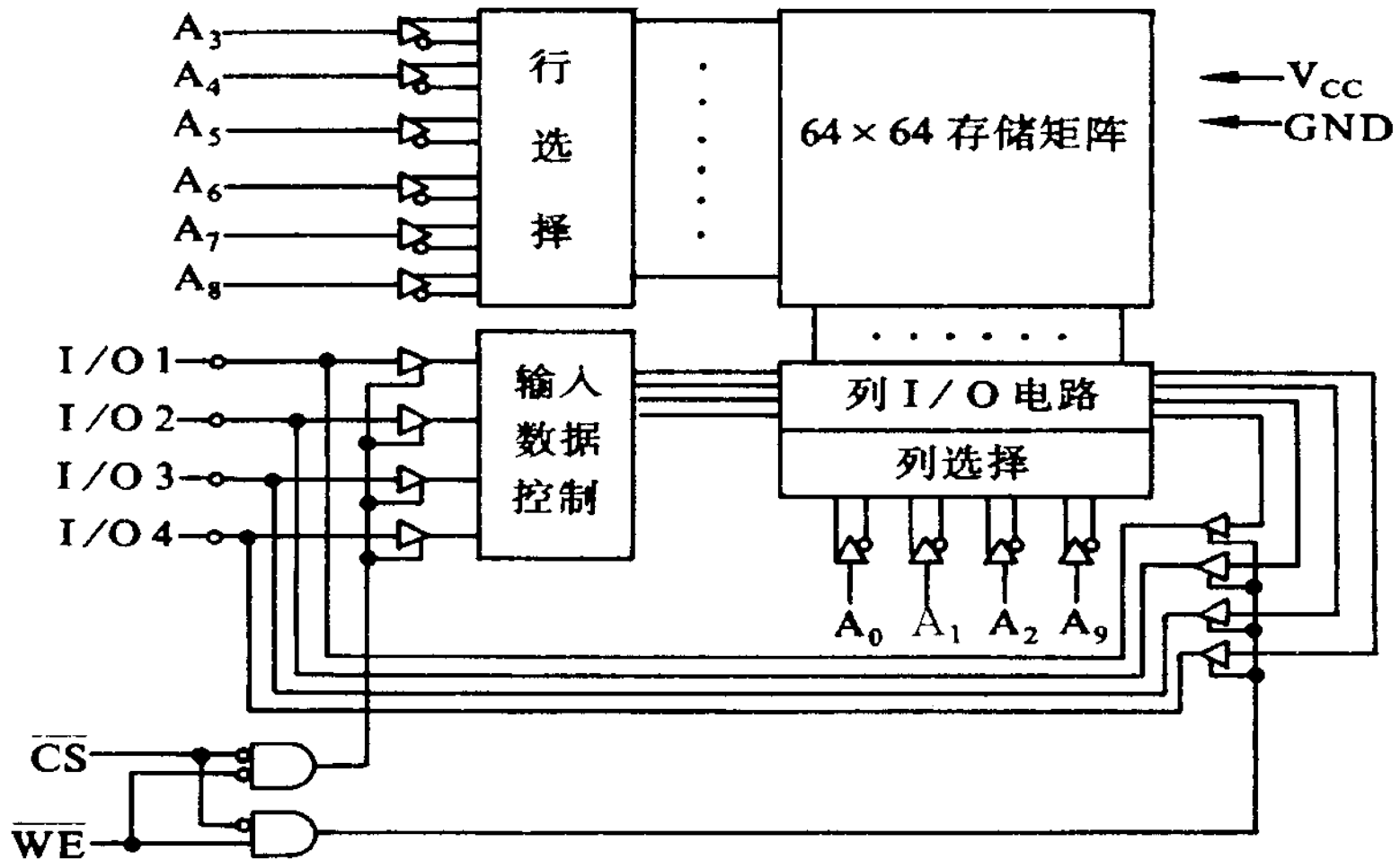
- 1) 可存放  $2^{15}$  个 字节的信息；
- 2) 其内部有地址线15根，数据线8根。

## 存储器芯片实例

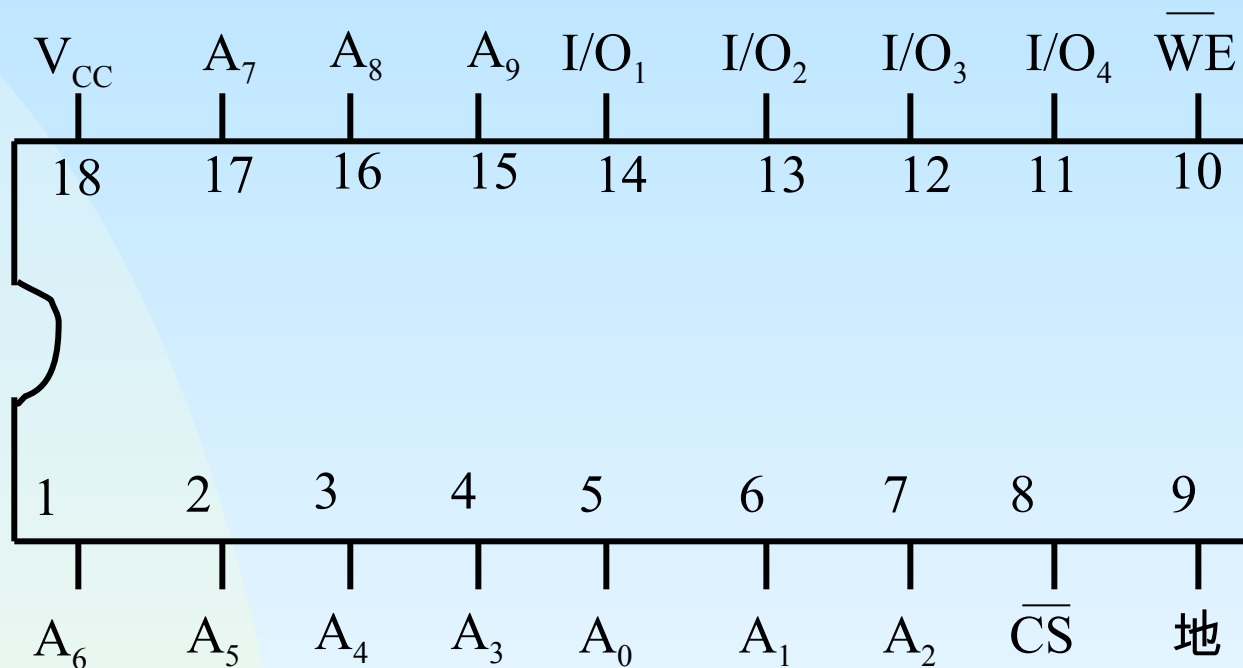
Intel 2114 —— 1K×4 存储器 ( SRAM ) :

- 1) 需\_\_\_\_根地址线寻址；
- 2) 数据线需\_\_\_\_根。

# Intel 2114——1K×4 SRAM



## Intel2114 引脚及功能：



$\overline{CS}$  : 片选，低电平选中；

$\overline{WE}$  : 写使能；

$A_9 \sim A_0$  : 地址线；

$I/O_4 \sim I/O_1$  : 双向数据线。

# 存储容量的扩展

存储器容量：字数×位数

1. 位扩展：字数不变，扩充位数。

如：1K×4 的芯片构成 1K×8 的存储器。

2. 字扩展：位数不变，扩充字数。

如：1K×4 的芯片构成 4K×4 的存储器。



## 1. 位扩展

用若干片**位数较少**的存储器芯片构成**位数较多**的存储器。

方法：

- 1) 由  $m \times n_1$  的存储芯片组成  $m \times n_2$  的存储器，需  $n_2/n_1$  片芯片；
- 2) 各片**数据线**相拼，**地址线**与**片选信号线**相同。

拼成  $n_2$

位

对任一地址，同时选中  
各片相应的存储单

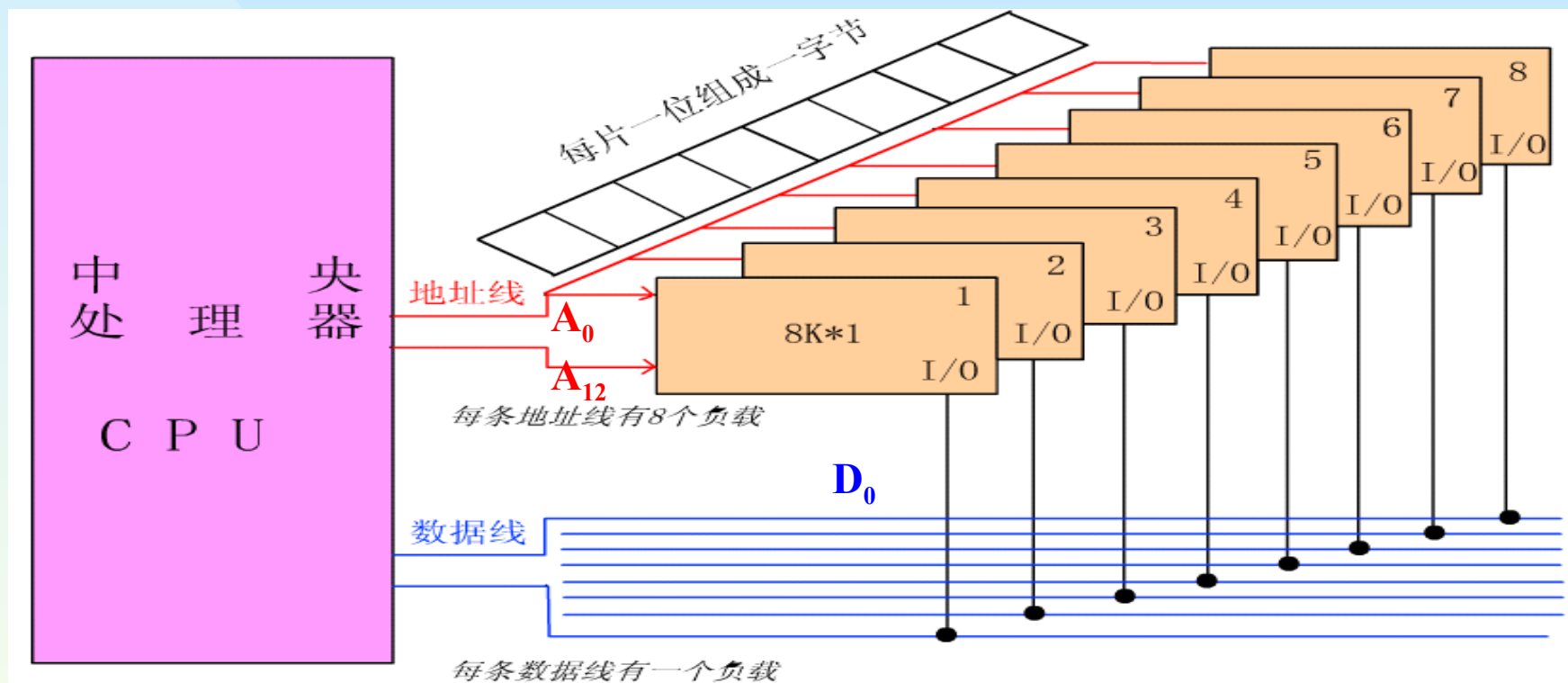
元



**例**. 用  $8k \times 1$  的片子组成  $8k \times 8$  的存储器需\_\_个芯片。

地址线——**13**\_\_根      数据线——**8**\_\_根

控制线——  $\overline{WE}$  接存储器的  $\overline{WE}$



只加大字长，**存储器的字数与芯片字数一致。**



## 2. 字扩展

字扩展是字数的扩充，位数不变。

方法：

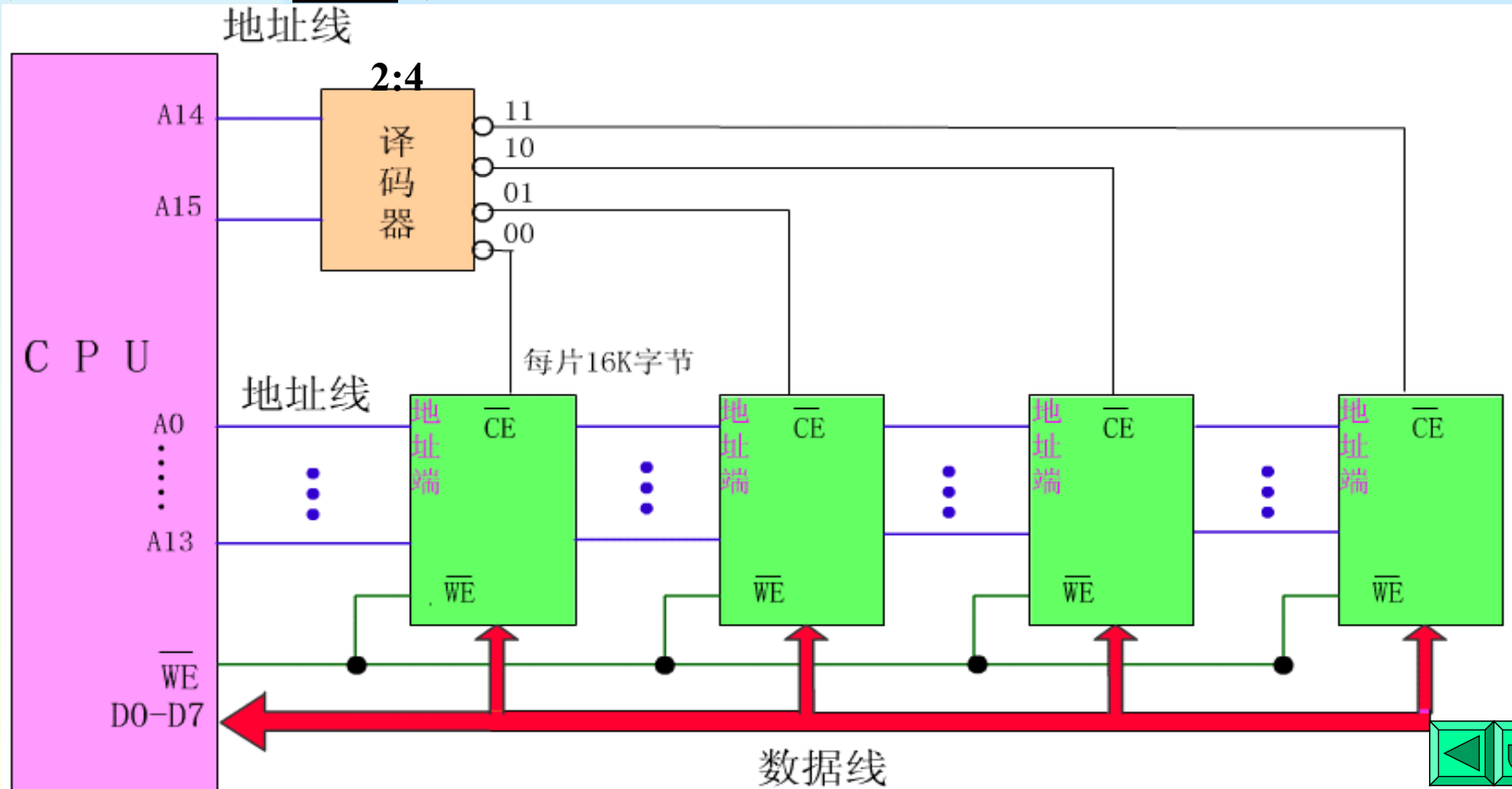
- 1) 由  $m_1 \times n$  的存储芯片组成  $m_2 \times n$  的存储器，需  $m_2/m_1$  片芯片；
- 2) 低位地址直接送往各芯片，选择片内单元；高位地址译码，产生片选信号；
- 3) 数据线同时按位连到数据总线。



**例**. 用  $16\text{K} \times 8$  的芯片组成  $64\text{K} \times 8$  的存储器需\_\_\_个芯片  
地址线——共需\_\_\_根；

片内\_\_\_根； 选片\_\_\_根

数据线——\_\_\_根 控制线——  $\overline{\text{WE}}$





## 地址空间分配表

地址 片号	选片 $A_{15} A_{14}$	片内 $A_{13} A_{12} \dots\dots\dots A_1 A_0$	总地址	说明
1	00 00	00,0000,0000,0000 11,1111,1111,1111	0000 3FFF	最低地址 最高地址
2	01 01	00,0000,0000,0000 11,1111,1111,1111	4000 7FFF	最低地址 最高地址
3	10 10	00,0000,0000,0000 11,1111,1111,1111	8000 BFFF	最低地址 最高地址
4	11 11	00,0000,0000,0000 11,1111,1111,1111	C000 FFFF	最低地址 最高地址

# 主存储器的基本组织

1. 半导体存储器的逻辑结构与设计：如何用存储芯片如 SRAM、DRAM、EPROM 等组成一个实际的存储器；
2. 主存与 CPU 的连接：主要是地址线、控制线、数据线的连接。
3. 主存校验，以确保存取信息的正确性。



# 1. 半导体存储器的逻辑结构与设计

## 步骤：

1. 明确存储器要求的总容量，即**字数 × 位数**；  
**字数**，编址单元数；  
**位数**，每个编址单元的位数；
2. 确定选用的存储芯片：类型，型号，每片容量等；  
若每片容量低于总容量，则需用多块芯片组成。  
存在**位数**与**字数**的扩展问题；
3. 半导体存储器的**寻址逻辑**设计——如何按地址选择芯片和片内单元。

**例**，某半导体存储器容量  $4k \times 8$  位，其中固化区  $2K$  B，选用 EPROM 芯片 2716( $2k \times 8$ )；工作区  $2K$  B，选用 RAM 芯片 2114( $1k \times 4$ )。

地址总线  $A_{11} \sim A_0$ ，双向数据总线  $D_7 \sim D_0$ 。

**解** . 1) 存储空间分配，确定芯片数，

既有**位扩展**，又有**字扩展**，

芯片数：

2716**1**：\_\_\_片；

2114 **(2/1) × (8/4) = 4** \_\_\_片。

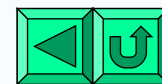
2k×8	
1k×4	1k×4
1k×4	1k×4



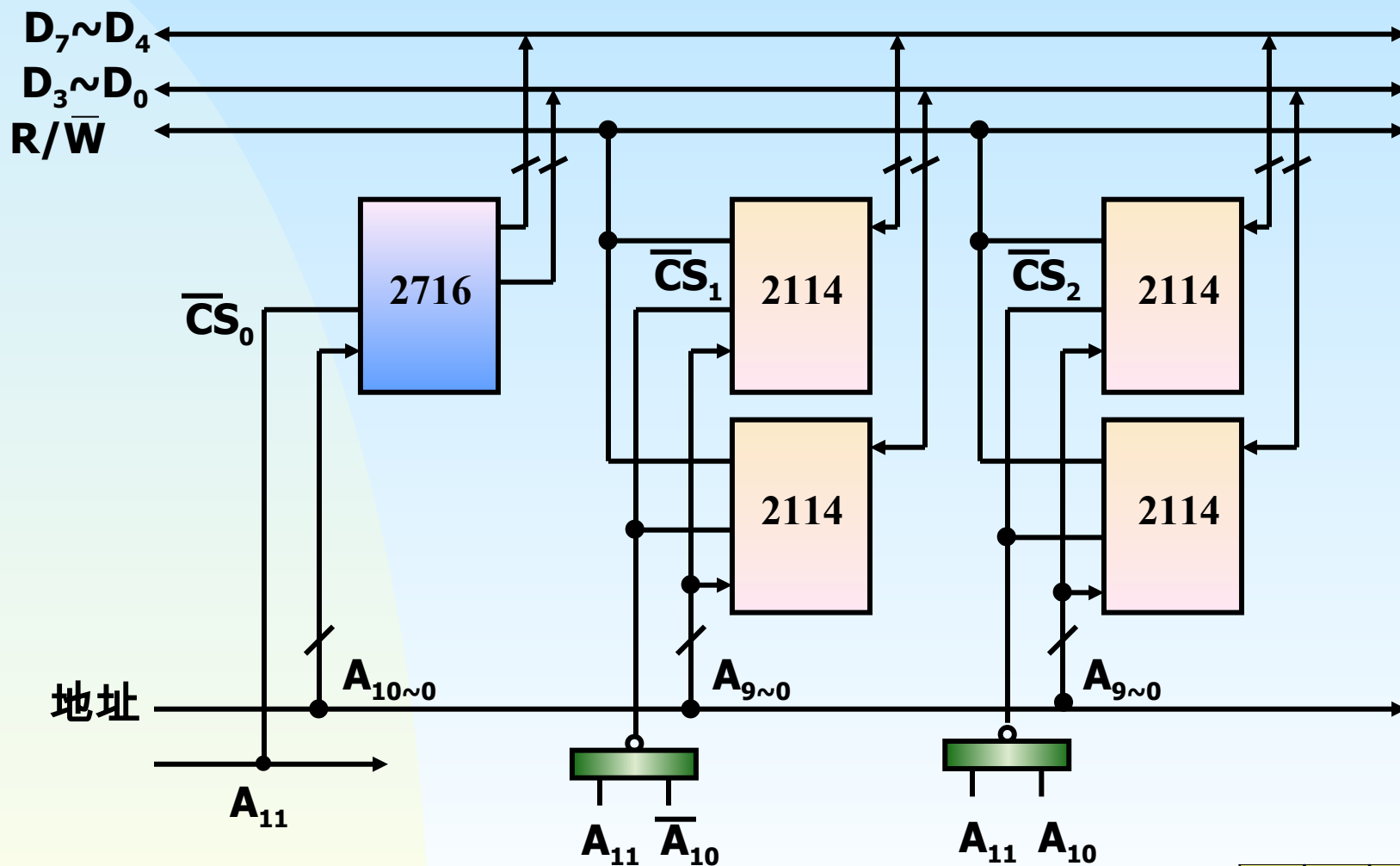


## 2) 地址分配与片选逻辑

芯片容量	芯片地址	片选信号	片选逻辑	地址范围
2k	$A_{10} \sim A_0$	$CS_0$	$A_{11} \text{ —}$	000H~7FFH
1k	$A_9 \sim A_0$	$CS_1$	$A_{11} A_{10} \text{ —}$	800H~BFFH
1k	$A_9 \sim A_0$	$CS_2$	$A_{11} A_{10}$	C00H~FFFH



### 3) 逻辑图



# 高速存储器

## 背景：

- 由于 CPU 和主存储器在**速度上不匹配**，限制了高速计算。
- 为了避免 CPU 因为等待存储器读写操作而无事可做，可以采取一些措施提高存储器的速度。

# 高速存储器

## 1. 芯片技术

研究开发高性能芯片技术。

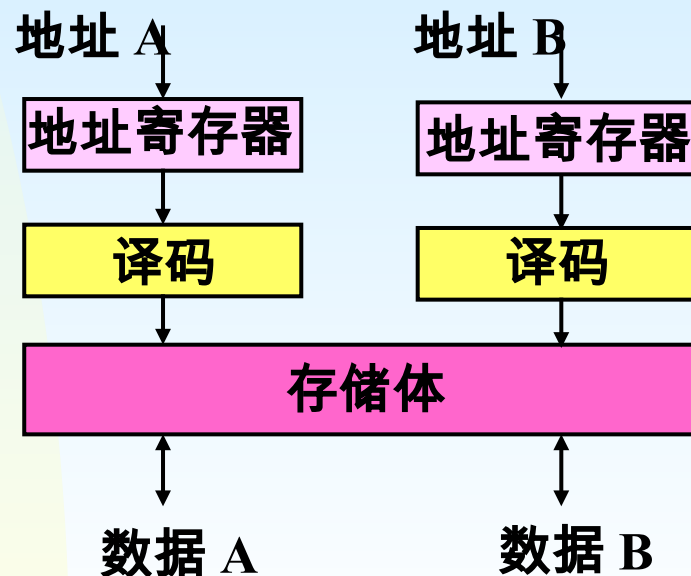
## 2. 结构技术

- 1) 采用**并行操作方式** —— **双端口存储器**
- 2) 采用并行主存储器，提高读出并行性  
—— **多体交叉存储器**
- 3) 主存储器采用更高速的技术来缩短存储器的读出时间  
—— **相联存储器**

# 双端口存储器

**单端口存储器**：每次只接收一个地址，访问一个编址单元。影响工作速度；

**双端口存储器**：有两个彼此独立的读写口，每个读写口都有一套独立的**地址寄存器**和**译码电路**，可并行的独立工作。



# 并行主存系统

为解决 CPU 与主存的速度匹配问题，可在一个存取周期中**并行存取**多个字。分为：

1. **单体多字并行主存系统**
2. **多体交叉存取方式的并行主存系统**

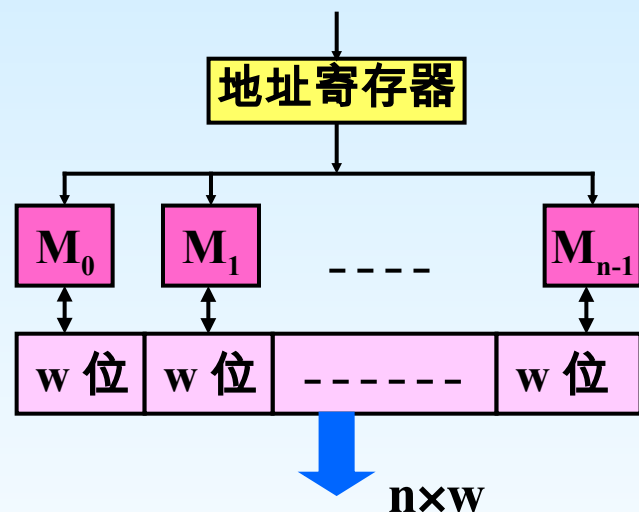
## 1. 单体多字并行主存系统

同一个地址码并行地访问多个存储器的对应单元。

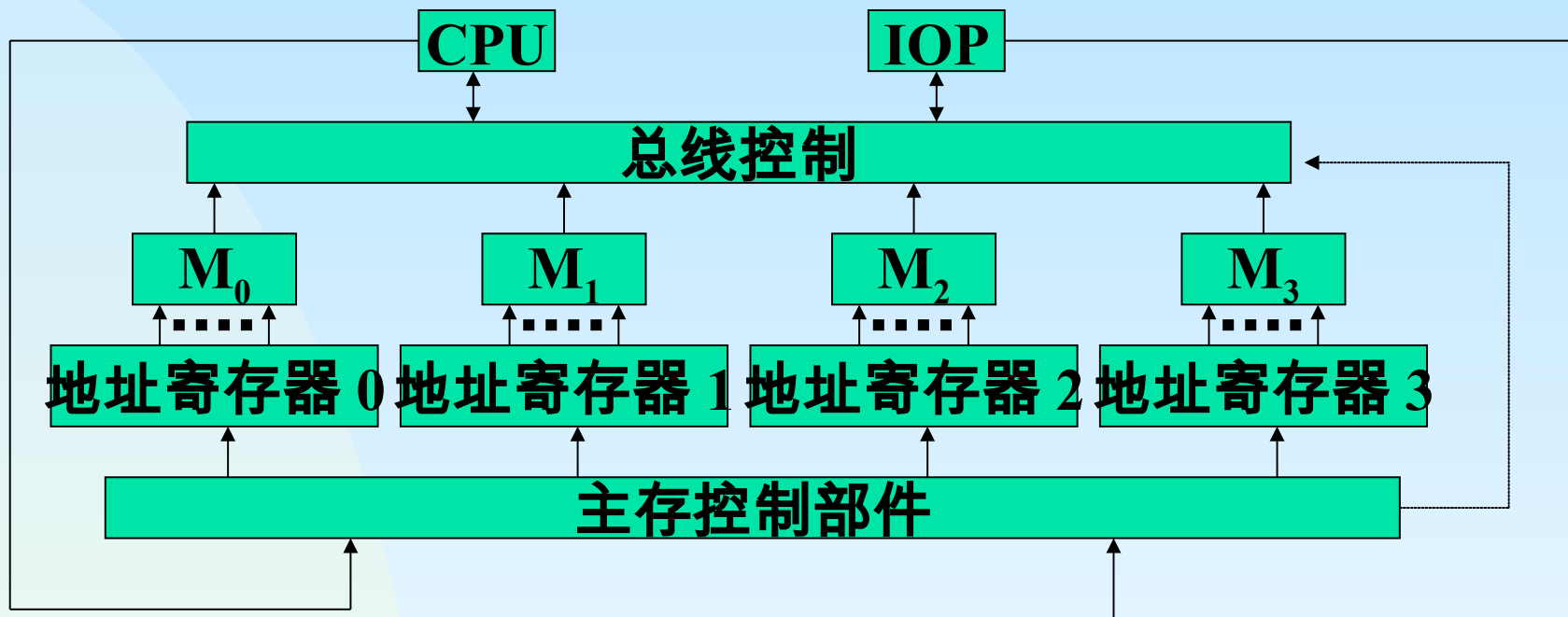
**单体**：只有一套地址寄存器和地址译码器。

**多字**：有多个容量相同的存储模块。

在一次读写周期中，同时对  $n$  个模块中的地址相同的单元进行读写，可同时读出  $n \times w$  位。



## 2. 多体交叉存取方式的并行主存系统



- 1)  $n$  个**独立的存储体** (存储模块) 组成，每个模块都有自己的读写线路、地址寄存器、译码器和数据寄存器；
- 2) 采用以  $n$  为模的交叉编址、交叉访问方式；
- 3) 一个存取周期可访存  $n$  次，提高了存取速度。

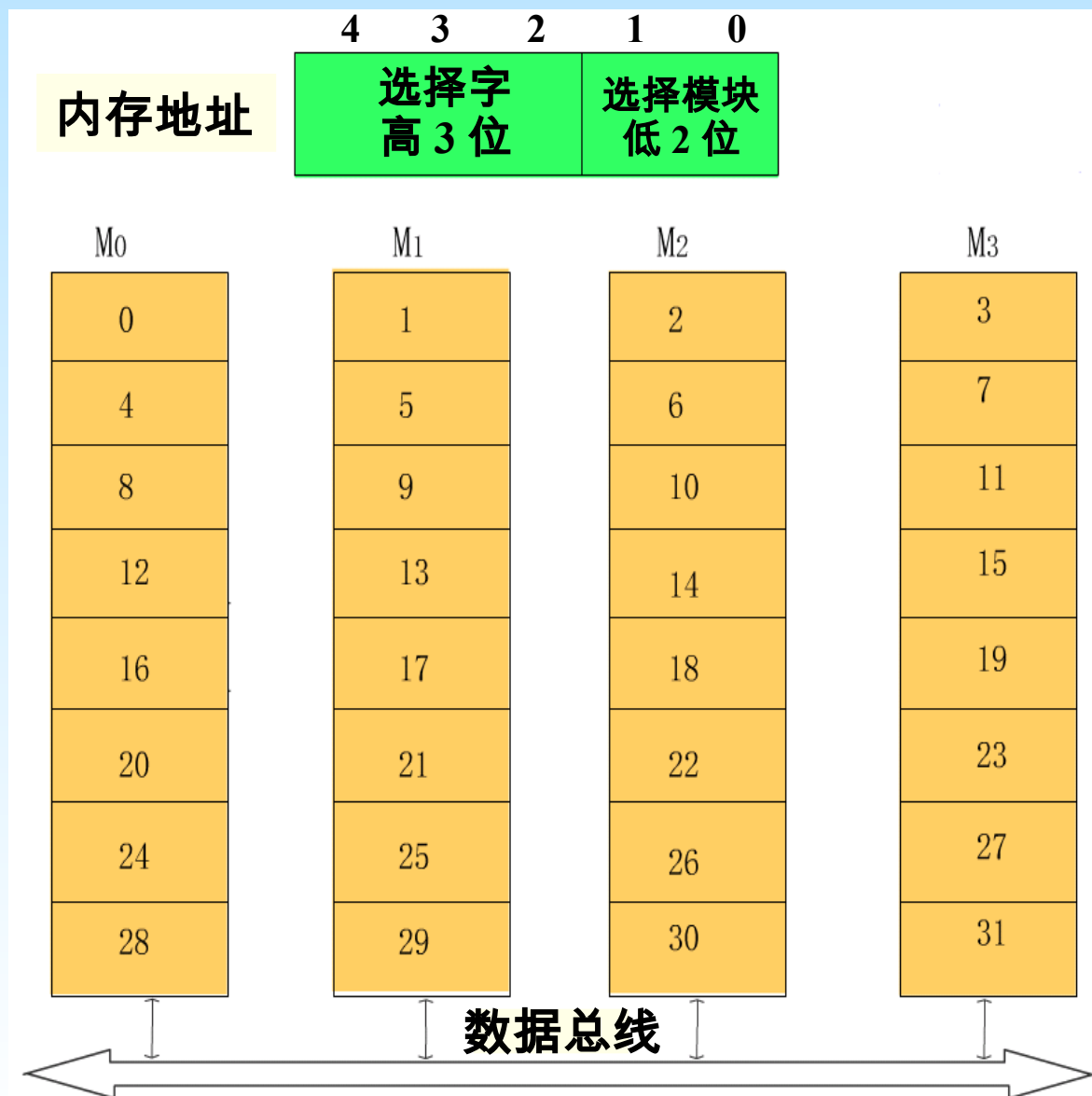




# 交叉编址 设有 $8 \times 4$ 个存储单元：

**问题：**与字扩展的区别？

- 1) 连续地址分布在相邻的不同模块内；
- 2) 同一个模块内的地址都是不连续的；
- 3) 通常情况下，程序段和数据块都连续的在主存中存放和读取。



## 模四交叉各模块的编址序列

片选

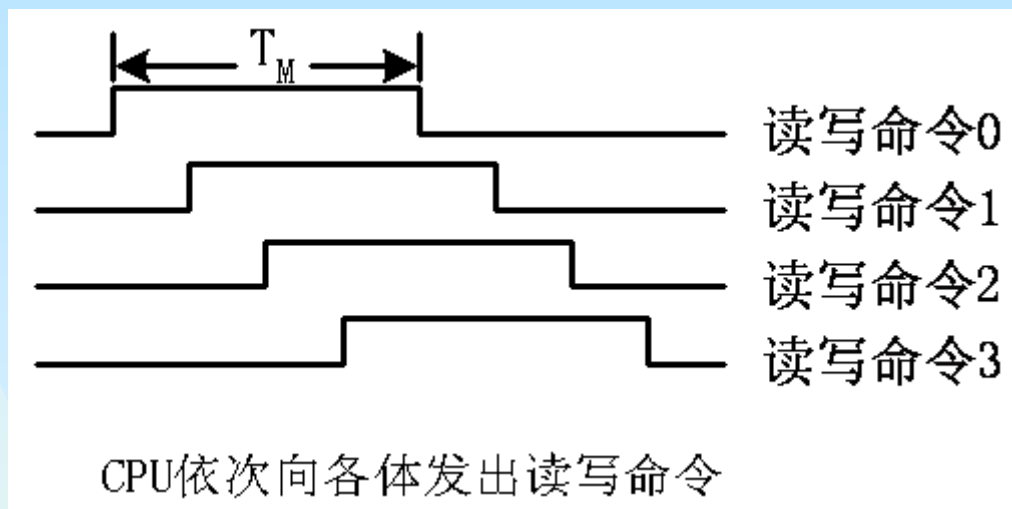
模体	地址编址序列	对应二进制地址最低二位
$M_0$	0,4,8,12, ..., $4 \cdot j + 0$ , ...	0 0
$M_1$	1,5,9,13, ..., $4 \cdot j + 1$ , ...	0 1
$M_2$	2,6,10,14, ..., $4 \cdot j + 2$ , ...	1 0
$M_3$	3,7,11,15, ..., $4 \cdot j + 3$ , ...	1 1

问题：高位（体内）地址？00...0 ~ 11...1

选择片内单元



## CPU 交叉访问 $M_0 \sim M_3$ 的时间安排：



### 说明：

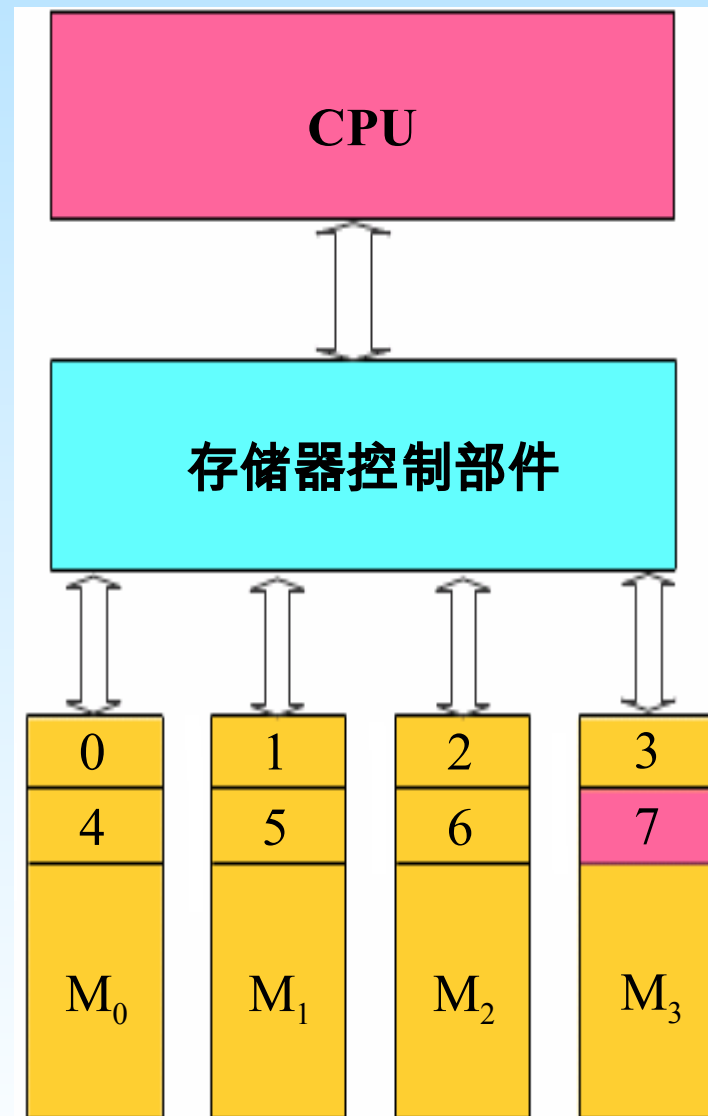
1. 尽管每个存储体的存储周期为  $T_M$ ，但 CPU 每隔  $1/4T_M$  就可以读出或写入一个数据。
2. 在不必提高各存储体存取速度的前提下，通过 CPU 与各存储体的**并行交叉存取操作**，提高整个主存储器的带宽。



# 多体交叉存储器的基本结构



- 1) 每个模块各自以等同的方式与 CPU 传送信息；
- 2) CPU 同时（轮流）访问四个模块，由存储器控制部件控制它们**分时使用数据总线**进行信息传递；
- 3) 对每一个模块来说，从 CPU 给出访存命令直到读出信息仍然使用了一个存取周期时间；
- 4) 对 CPU 来说，它可以在一个存取周期中连续访问 4 个模块；
- 5) 各模块的读写过程**重叠**进行，所以这是一种**并行**存储器结构。



# 相联存储器

相联存储器不是按地址访问的存储器，而是按**内容**寻址的存储器。

**基本原理**：把存储单元所存内容的某一部分作为检索项（即关键字项），去检索该存储器，并将存储器中与该检索项符合的存储单元内容进行读出或写入。



硬件实现符合比较

**主要应用**：虚拟存储器中存放段表、页表和快表。因为需要**快速查找**。

## 2. 主存与 CPU 连接

### 1. 连接模式。

- 1) 最小系统模式
- 2) 较大系统模式
- 3) 专用存储总线模式

### 2. 速度匹配与时序控制。

### 3. 主存与数据总线间宽度匹配问题。

# 连接模式

存储器同 CPU 的连接要完成：

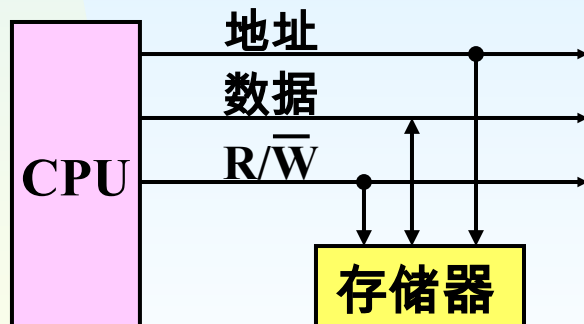
**地址线**的连接，包括内部地址线和芯片选择线；

**数据线**的连接，数据线对应相连；

**控制线**的连接，主要有读 / 写线  $\overline{WE}$  和存储器访问线  $\overline{MREQ}$  等。

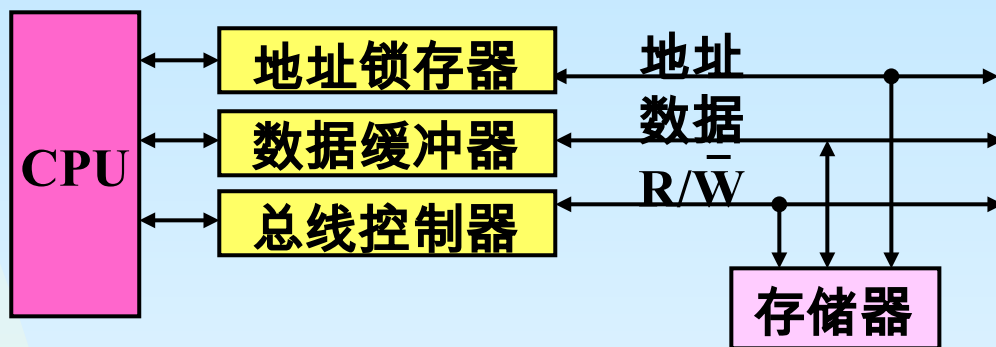
**类型：**

1) **最小系统模式**：CPU 与主存直接相连。

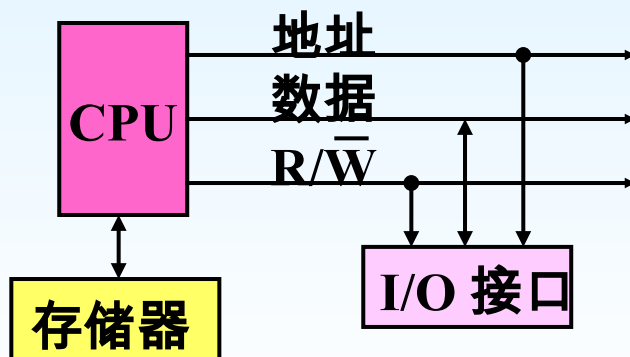


**实例**

- 2) **较大系统模式**：CPU 通过**数据缓冲器**、**地址锁存器**、**总线控制器**等接口芯片形成系统总线。主存作为独立的存储器模块挂在系统总线上。



- 3) **专用存储总线模式**：CPU 与主存间建立专门的**高速存储总线**，CPU 通过**存储总线**访存，通过**系统总线**访问外设。





## 5.5 主存储器

### 1. 主存中存储单元地址的分配

高位字节 地址为字地址

低位字节 地址为字地址

字地址	字节地址			
0	0	1	2	3
4	4	5	6	7
8	8	9	10	11

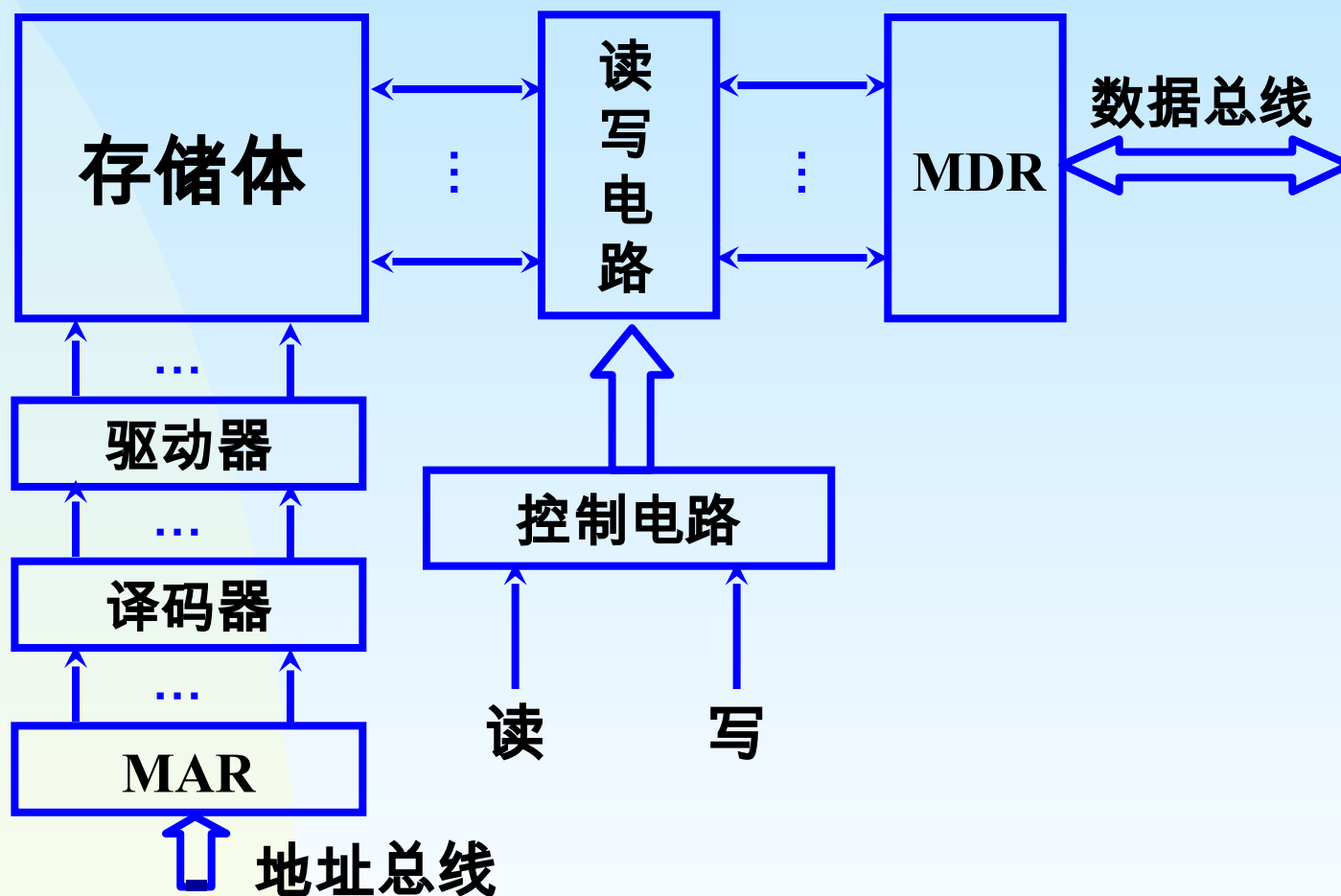
字地址	字节地址	
0	1	0
2	3	2
4	5	4

设地址线 24 根      按 字节 寻址  $2^{24} = 16 \text{ M}$

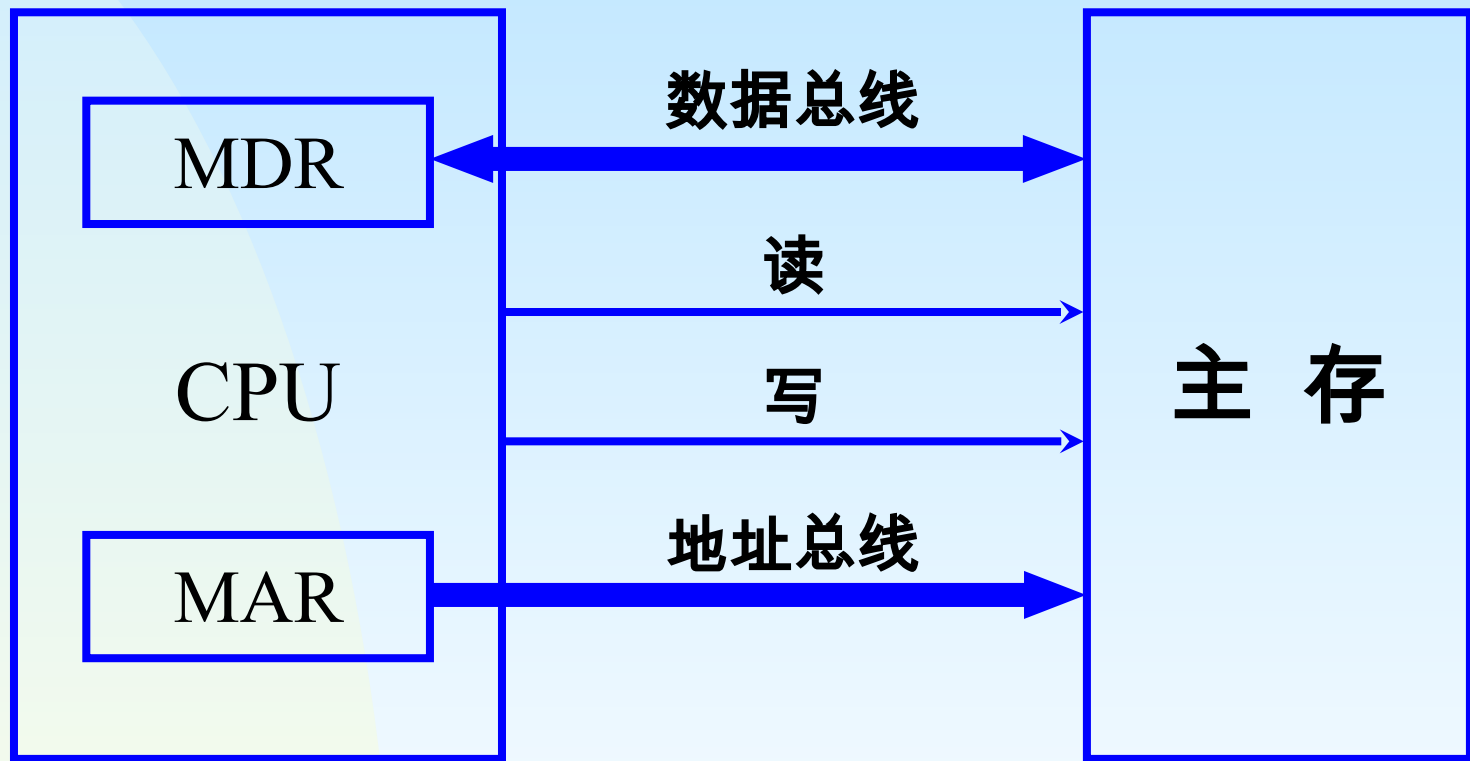
若字长为 16 位      按 字 寻址      8 M

若字长为 32 位      按 字 寻址      4 M

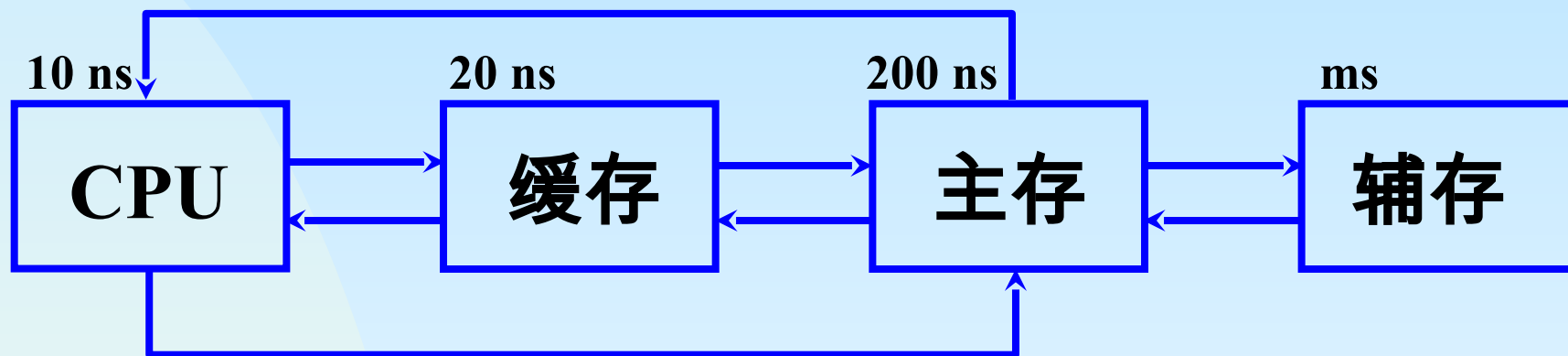
## 2. 主存的基本组成



### 3. 主存和 CPU 的联系



#### 4. 缓存 主存层次和主存 辅存层次



( 速度 )      ( 容量 )  
缓存 — 主存      主存 — 辅存

主存储器

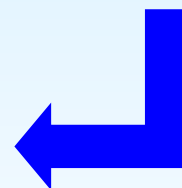
虚拟存储器

实地址

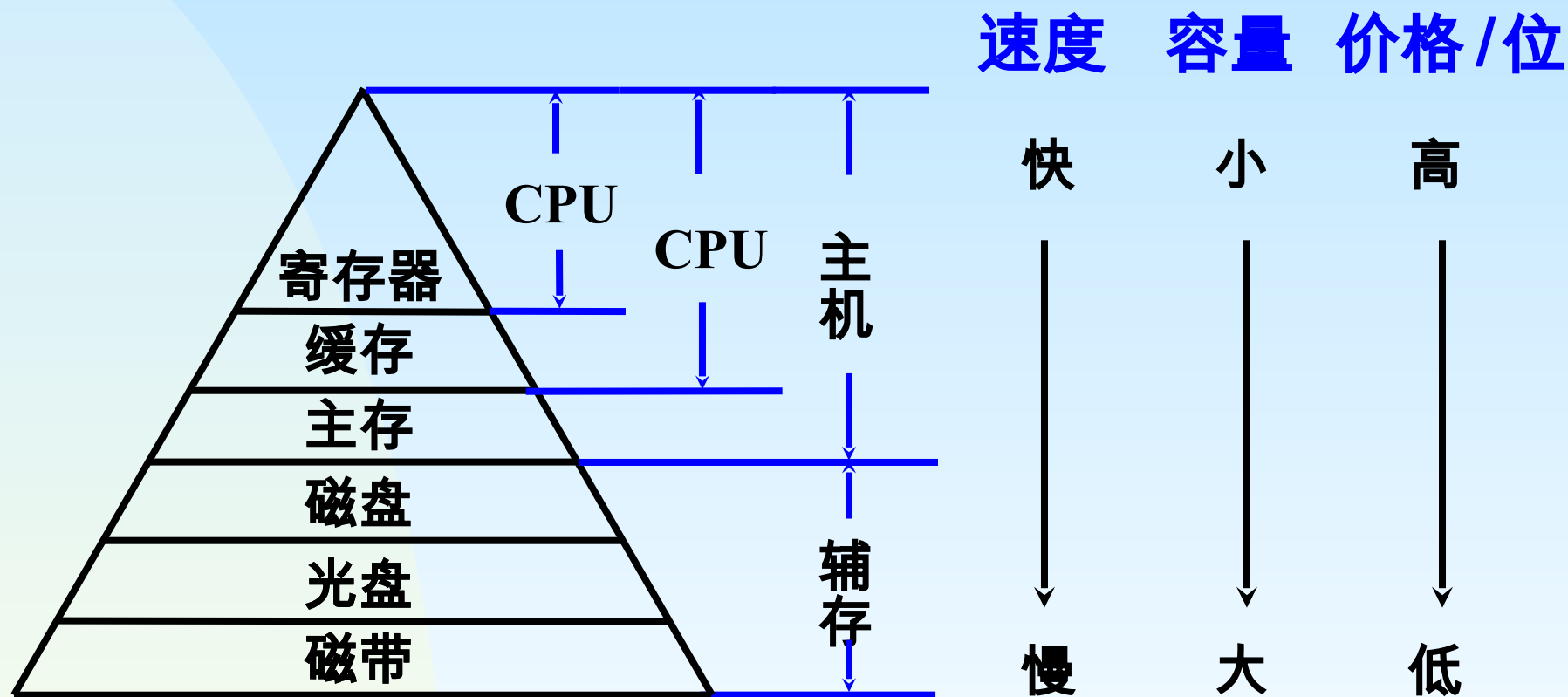
虚地址

物理地址

逻辑地址



## 5. 存储器三个主要特性的关系



## 存储器的速度：

指 标	含 义	表 现	单 位
存储容量	在一个存储器中可以容纳的存储单元总数	存储空间的大小	字数，字节数
存取时间	启动到完成一次存储器操作所经历的时间	主存的速度	n s
存储周期	连续启动两次操作所需间隔的最小时间	主存的速度	n s
存储器带宽	单位时间里存储器所存取的信息量	数据传输速率	位 / 秒，字节 / 秒



# 存储系统的层次结构

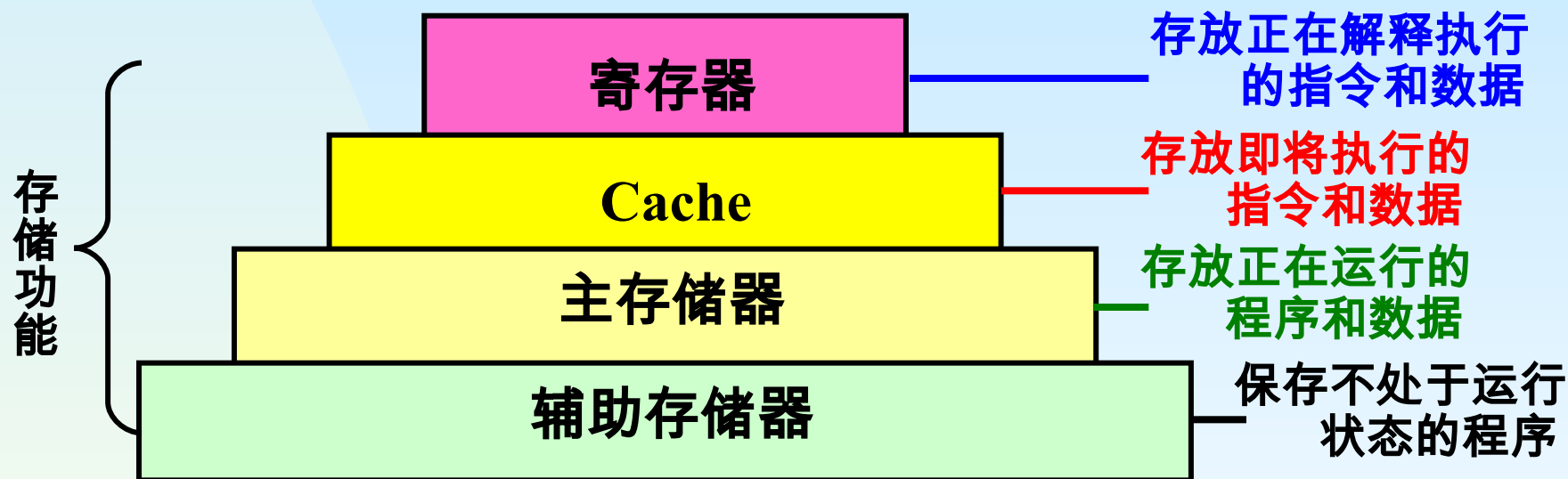
对存储器的**基本要求**：

- 1) **容量大**：伴随计算机的发展，处理的程序量与数据量越来越大；
- 2) **速度高**：提高计算机的工作速度，存储器的速度是关键；
- 3) **成本低**。

各类存储器各具特点：

- 1) 半导体存储器速度快、成本较高；
- 2) 磁表面存储器容量大、成本低，但速度慢。

为解决速度、容量、价格三者之间的矛盾，提高存储器的性能，目前通常采用**多级存储器体系结构**。



举例说明。



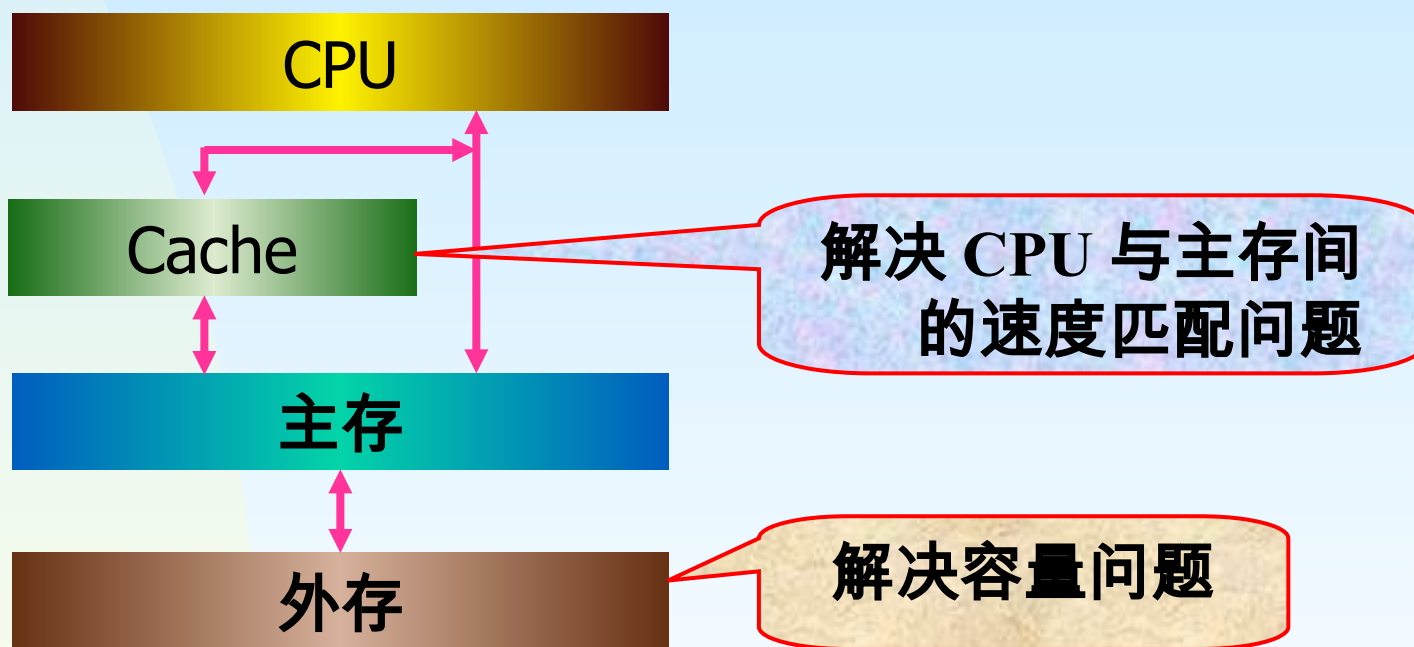


## 存储系统的层次结构：

- 1) 把各种**不同容量**和**不同存取速度**的存储器按一定的结构**有机**地组织在一起；
- 2) 程序和数据按不同的层次存放在各级存储器中；
- 3) 整个存储系统具有较好的**速度**、**容量**和**价格**等方面的综合性能指标。



# 三级存储体系结构：高速缓冲存储器—主存储器—外存储器



## 主存储器

- 1) 能由 CPU **直接编程访问**（根据地址），存放当前正在运行的程序和处理的数据；
- 2) 对主存的基本要求：
  - a. 随机访问；
  - b. 速度快（与 CPU 匹配）；
  - c. 一定的存储容量（切换使效率下降）。

## 外存储器

- 1) 容量大，作为主存的补充与后援；
- 2) CPU **不能直接访问**。存放需要联机保存但暂不使用的程序与数据（如操作系统、编译程序、调试程序等），需要时须调入内存后才能处理；
- 3) 按文件名调用，按数据块存取。



## 5.7 高速缓冲存储器

### Cache 工作原理——程序访问的局部性

**程序访问的局部性**：在一个较短的时间间隔内，CPU 对**局部范围**内的存储器地址频繁访问，而对此范围之外的地址访问很少。

- 1) 指令分布的**连续性**和循环程序及子程序的多次执行。
- 2) 对数组的访问可使存储地址相对集中。

# 高速缓存

- 1) 主存与 CPU 的工作速度存在大约一个数量级的差距；
- 2) 高速缓存可快速存取，存放最近使用的程序和数据（**程序访问的局部性**）；
- 3) CPU 访问内存时，同时访问 Cache：
  - a. 访问**命中**：访问区间内容已经复制在 Cache 中，直接从 Cache 中快速读取信息；
  - b. 访问**未命中**：访问区间内容不在 Cache 中，以较慢的速度从主存读取信息，并更新 Cache 内容。



## (1) “高速缓存—主存”层次

主要解决 CPU 和主存之间的**速度匹配**问题。

在 CPU 与主存之间再增设一级存储器，称**高速缓冲存储器** (Cache)。

Cache：

- 1) 采用和 **CPU 工艺** 相类似的半导体器件构成的存储装置，其**速度**可与 CPU 相匹配；
- 2) 容量较小，只能存放一小段程序和数据。

CPU 访问内存时，将**地址码**同时送到 Cache 和主存，**“命中”**与**“不命中”**，不命中时要进行**数据交换**。



地址映象以及和主存数据交换全由硬件实现，并对程序员透明。

访问 Cache 的时间一般为访问主存时间的  $1/4—1/10$

。高档微机中为获得更高的效率，不仅设置了独立的指令 Cache 和数据 Cache，还设置二级或三级 Cache

如，

- 1) Pentium 微处理器在芯片内集成了一个代码 Cache 和一个数据 Cache( 各为 8KB) ；
- 2) Power PC 微处理器在芯片内集成了一个指令 Cache 和一个数据 Cache( 各为 32KB)



## (2) “主存—辅存”层次

主要解决存储器的**容量**问题。

把正在被 CPU 使用的**活动**的程序和数据放在主存中，其余信息则存放在容量大、但速度较慢的辅存中。

含“**主存—辅存**”层次的存储系统，具有：

- 1) 接近于主存的存取速度；
- 2) 辅存的大容量；
- 3) 低成本。





## 5.8 虚拟存储器

### 问题的提出：

1. 系统程序和应用程序要求主存容量越来越大；
2. **地址字**能直接访问的存储空间，比主存的实际存储空间大得多。程序员用这种较长的地址字编程序，会感到主存容量不够用（如  $P_{2\sim4}$ ：地址总线 36 位）

### 虚拟存储技术：

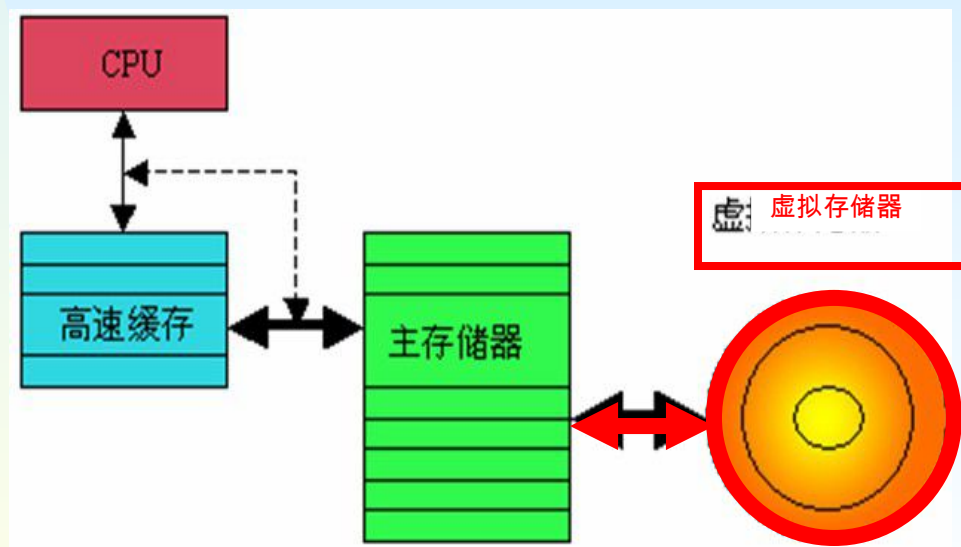
1. 将部分（或全部）辅存和主存结合，把两者的地址空间**统一编址**，依靠操作系统存储管理软件的支持，使用户访存的**编址范围**远比真实主存空间
2. 访存时，用软、硬件结合的方法，将**逻辑地址**（**虚地址**）转化为**物理地址**（**实地址**）。



# 虚拟存储器概述

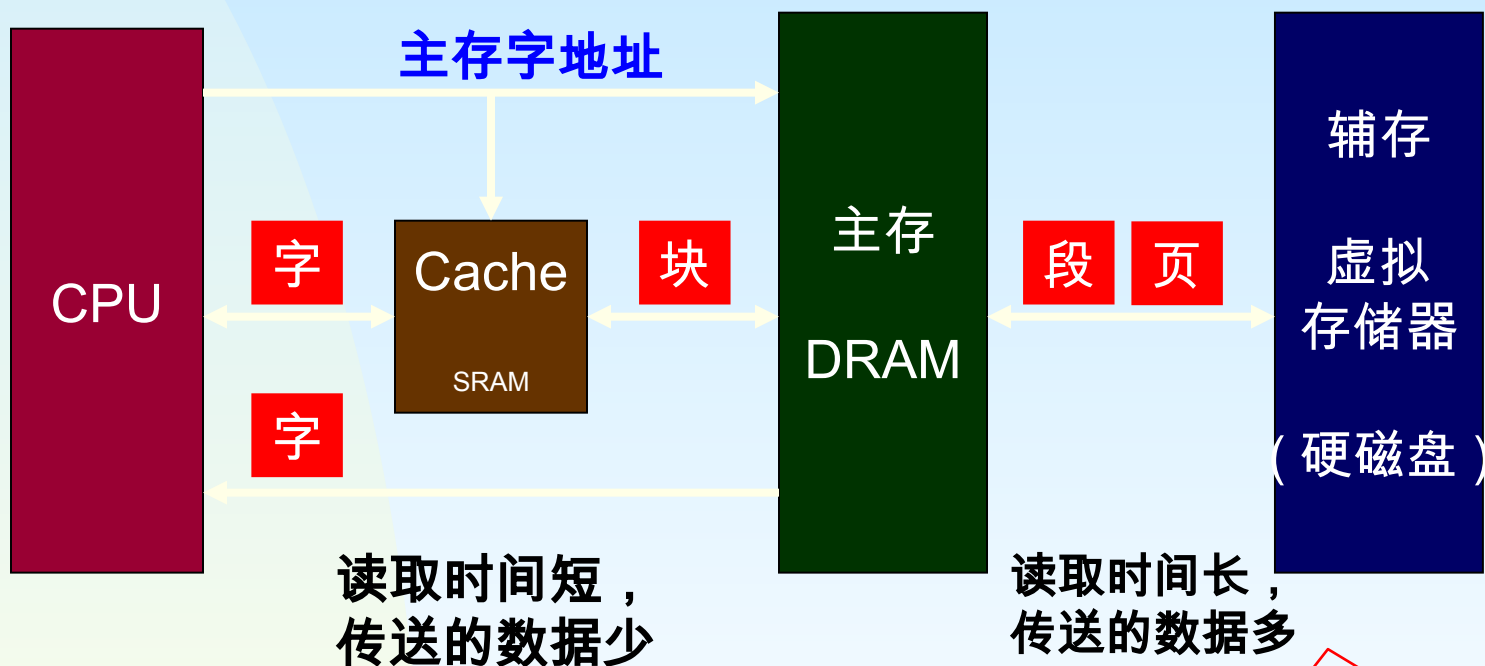
**问题**：主存容量不够大，如何在计算机内运行比主存容量大，甚至大得多的程序？

**解决方案**：虚拟存储器技术



# 虚拟存储器概述

主存 ---- 辅存层次的基本数据传送单位可采用几种不同的方案：  
**段、页、段页。**



由操作系统和硬件共同完成，  
对应用程序员都是透明的但系  
统程序员不透明

# 虚拟存储器概述

## 虚拟存储器的基本概念

**虚拟存储器**是指**主存 --- 辅存**层次存取系统，  
借助于**硬磁盘等辅助存储器**来**扩大主存容量**，  
因为 CPU 只能执行调入主存的程序，所以称“虚拟存储器”。

**虚拟存储器**是一个大容量的存储**逻辑模型**，不是实际的物理存储器。

**物理地址**：由 **CPU 地址引脚**送出的，用于**访问主存**的地址。

**逻辑地址**：由**编译程序生成**，是访问程序的**逻辑地址**，  
其地址空间大小受到**辅助存储器容量**的限制。

**Cache --- 主存层次**与**主存 --- 辅存层次**的  
**地址变换映射方式**和**替换策略**都**相同**，都是基于程序局部性原理。

# 虚拟存储器概述

**两个存储层次遵循的共同原则是：**

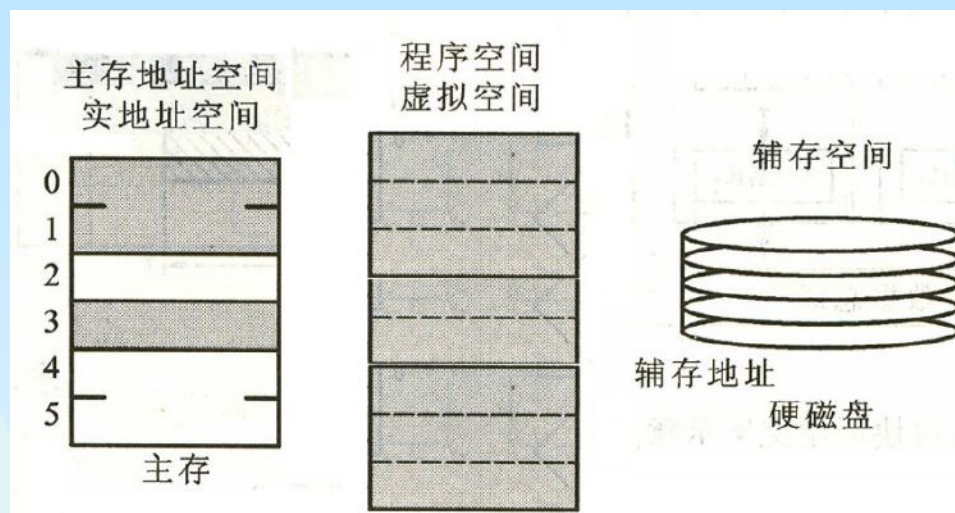
把程序最近、最常用的部分留在高速存储器中，变得不常用了，再送回到低速存储器中，使存储系统的性能接近高速，价格接近低速。

**两个存储层次的区别是：**

- 1 ) 主存 -Cache 主要用来弥补主存和 CPU 之间的速度差距，主存 - 辅存主要用来弥补主存的容量不足。
  - 2 ) 主存的读写时间是 Cache 读写时间的 5--10 倍，而硬盘的读写时间是主存的千倍，所以主存未命中，系统相对性能损失大。
  - 3 ) CPU 与主存、Cache 之间有直接访问通路，与辅存没有。
  - 4 ) 主存 -Cache 之间地址变换、数据替换全部由硬件实现，对程序员完全透明。
- 主存 - 辅存之间地址变换、数据替换由操作系统中的存储管理软件辅助一些硬件共同实现，对系统程序员可见。  
( 所以，这部分内容也放在操作系统中讲。 )

# 虚拟存储器概述

## 主存—辅存地址空间



**实地址空间**：实际存在的存储空间，指主存地址由 CPU 地址引脚送出。

**虚存空间**：比主存空间大得多的存储空间，其大小取决于虚地址的长度。不是主存，也不是辅存，实际并不存在。由编译程序生成是程序的逻辑地址。

**辅存空间**：机器中的辅存储器地址空间

虚地址结构与主存的存储管理策略有关  
辅存实地址格式与采用的存储设备有关

# 虚拟存储器概述

## 页式管理

虚地址

虚页号	页内地址
-----	------

实地址

实页号	页内地址
-----	------

## 磁盘辅存:

辅存地址:

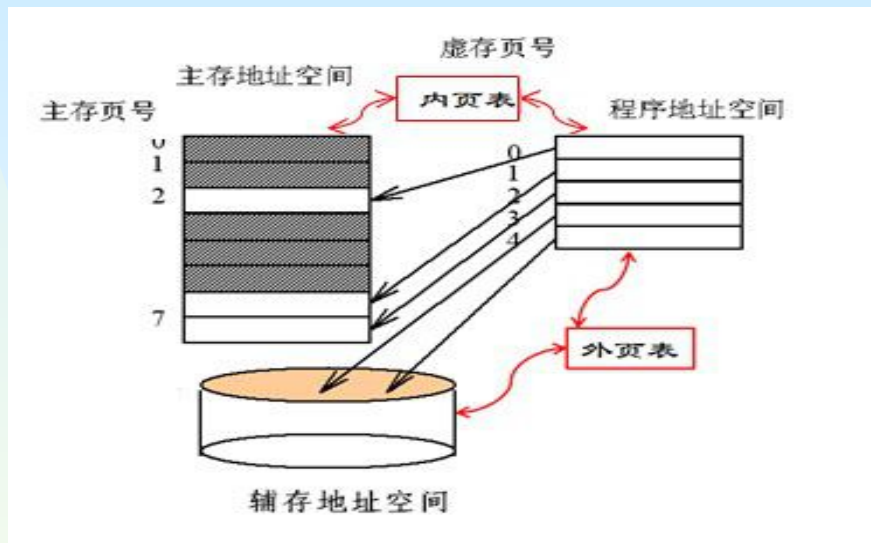
磁盘机号	磁头号	磁道号	扇区号	扇区内地址
------	-----	-----	-----	-------

辅存实页号

辅存页内地址

# 虚拟存储器概述

**虚存技术**就是实现两种虚实变换：若命中，要将虚地址（虚拟存储器存储单元的地址）变为主存地址；若不命中，需将虚地址变为虚单元所在信息块的辅存地址（称为辅存实地址）



**方法：**

**内页表：**把虚存地址变换成主存页号的对照表

**外页表：**把虚拟页号转换成一个辅助存储器上物理地址的对照表



# 存储系统和结构

## ■ 虚拟存储器

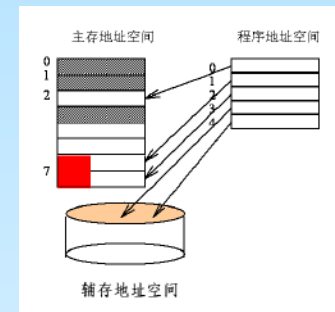
1. 虚拟存储器概述
2. 虚拟存储器管理方式
3. 替换算法
4. 快表 T L B
5. 层次存储器工作全过程

# 页式虚拟存储器

**页式虚拟存储器**：以**页**为基本单位与主存交换数据。

主存空间也分成**同样大小**的**页**。

主存分成的页为**实页**，虚拟存储器分成的页为**虚页**，



**程序虚地址**分为两个字段：高位字段为虚页号，低位字段为页内地址。

虚页号

页内地址

**程序实地址**分为两个字段：高位字段为实页号，低位字段为页内地址。

实页号

页内地址

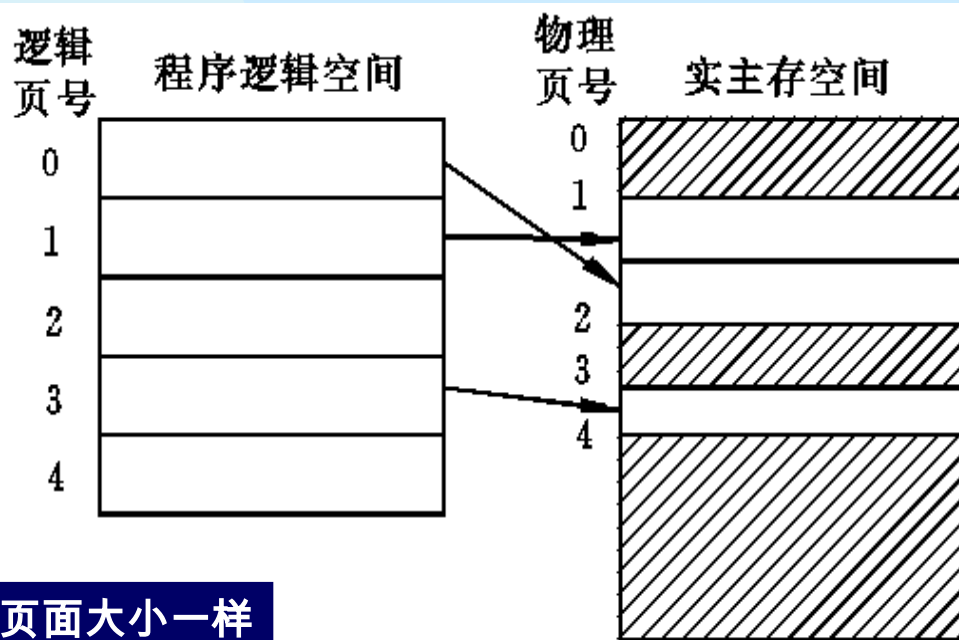
由于虚页与实页大小一样，所以**页内地址是相等**的。

**虚页号**与**实页号**之间的变换是通过查找**主存中的页表**来实现的。

# 页式虚拟存储器

**优点：**页面的起点和终点地址是**固定的**，方便造页表，新页调入主存也很容易掌握，页外空间浪费少。

**缺点：**处理、保护、共享都不方便。



页面大小一样

页表		
逻辑页号	实主存页号	装入位
0	2	1
1	1	1
2		0
3	4	1
4		0

页表在主存中，增加了访问主存次数，即使命中也要先访问主存中的页表，再访问主存单元，所以把页表最活跃部分放在 Cache 中组成**快表**。**快表由硬件构成**，减少了访问主存的时间开销。

# 页式虚拟存储器

页表基址寄存器

页表起始地址

虚页号 页内地址 虚地址

页表起始地址 虚页号

页表地址 (主存中)

	装入标志	实页号
0	1	2
1	1	6
2	1	7

实页号 页内地址 实地址

按照页表起始地址  
+ 虚页号查找页表，  
得到实页号。

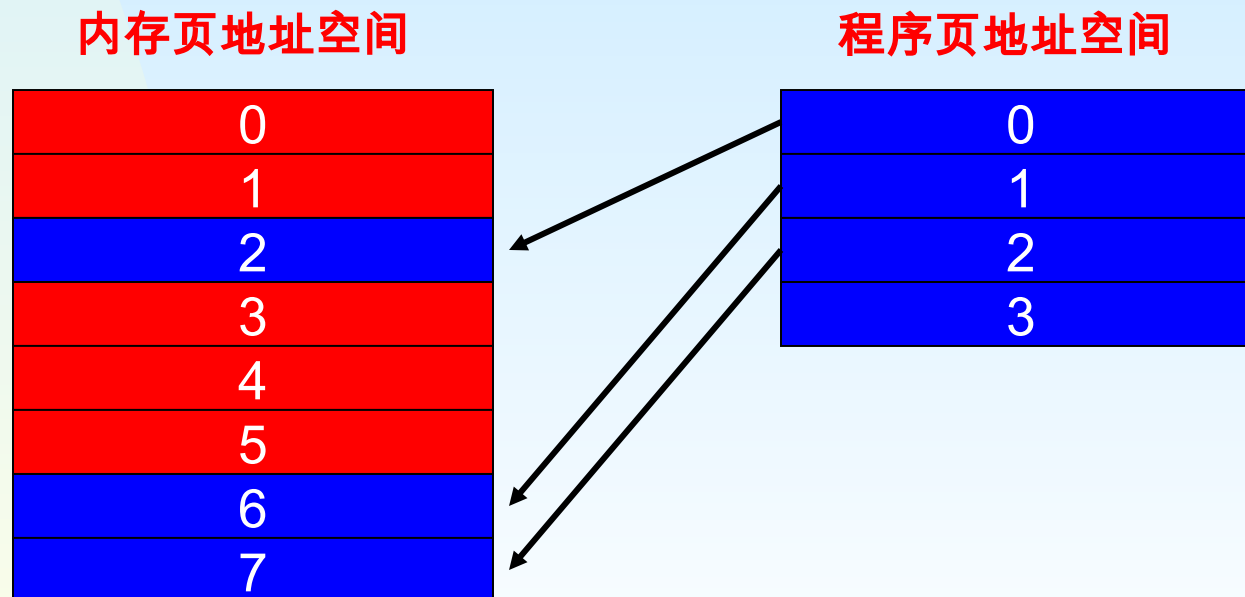
实页号 + 页内地址  
拼接得内存实地址。

# 页式虚拟存储器

**例 1** : 在一个采用页式管理的虚拟存储器中，假设程序的地址空间由 4 个

页面组成，请画出它们之间的映像方式。

第 0 页映象到第 2 个主存页中，第 1 页映象到第 6 个主存页中，  
第 2 页映象到第 7 个主存页中，第 3 页映象到辅存。

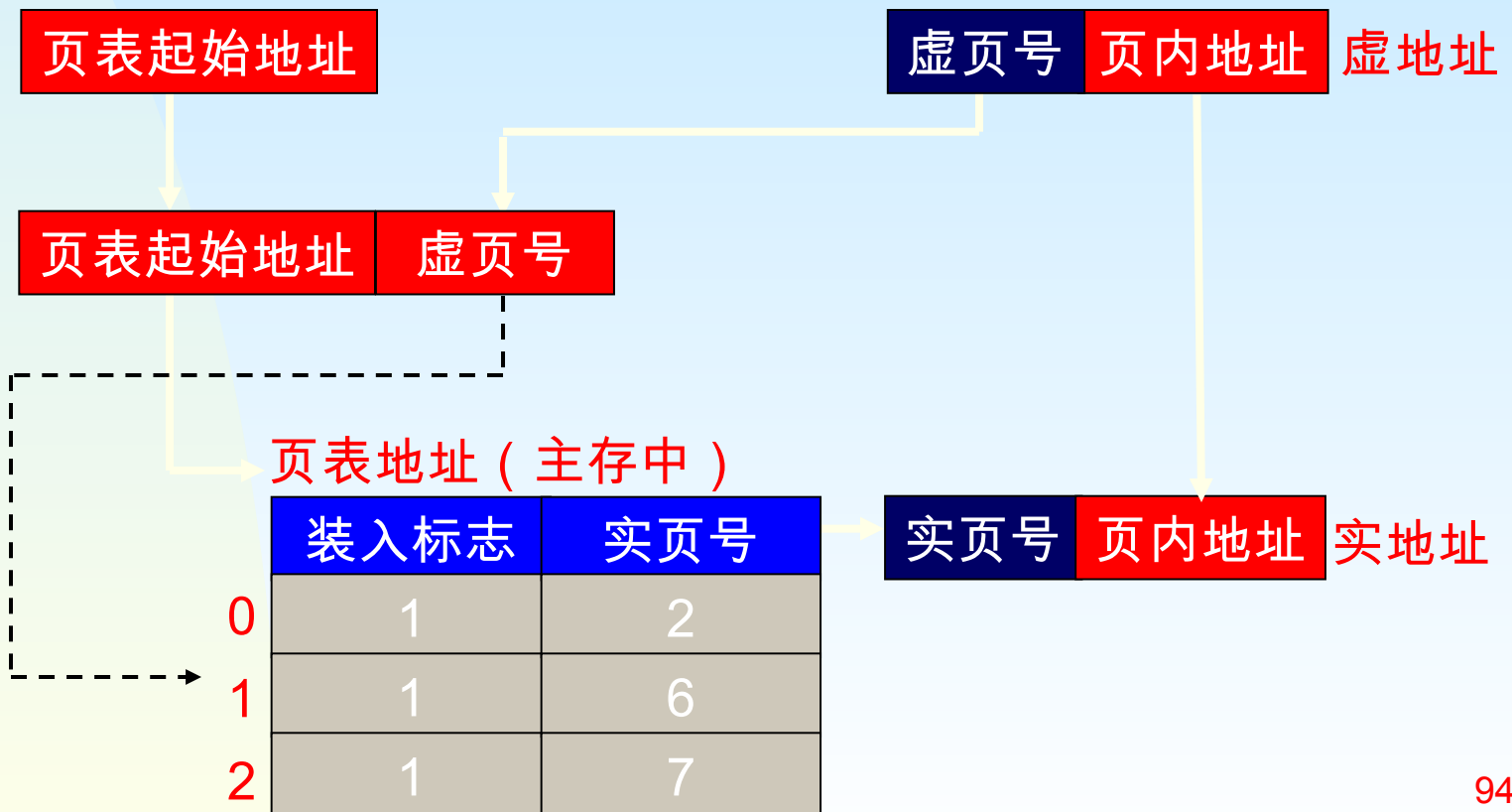


# 页式虚拟存储器

**例 1**：在一个采用页式管理的虚拟存储器中，假设程序的地址空间由 4 个

页面组成，请画出它们之间的映像方式。

第 0 页映像到第 2 个主存页中，第 1 页映像到第 6 个主存页中，  
第 2 页映像到第 7 个主存页中，第 3 页映像到辅存。



# 页式虚拟存储器

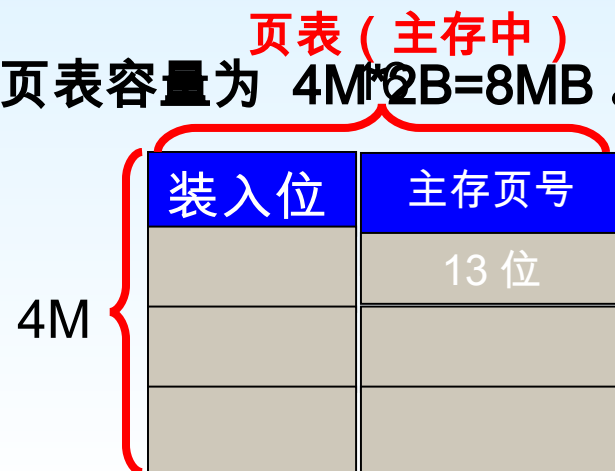
- 例 2**、一个存储系统有 32 位程序地址空间（**虚地址**），  
一页容量为 1KB，主存容量为 8MB，问：
- 1 ) 虚页号字段多少位？页表将有多少行？
  - 2 ) 页表每行多少位？页表的容量为多少字节？

**解：** 1 ) 页容量为  $1\text{KB}=2^{10}\text{B}$ ，所以页内地址为 10 位，  
虚页号为  $32-10=22$  位，页表长度为  $2^{22}=4\text{M}$  行。

2 ) 主存容量为  $8\text{MB}=2^{23}\text{B}$ ，主存中的页数有： $2^{23}/2^{10}=2^{13}$  个。

即主存中实页号为 13 位，再加上装入位和访问控制等信号

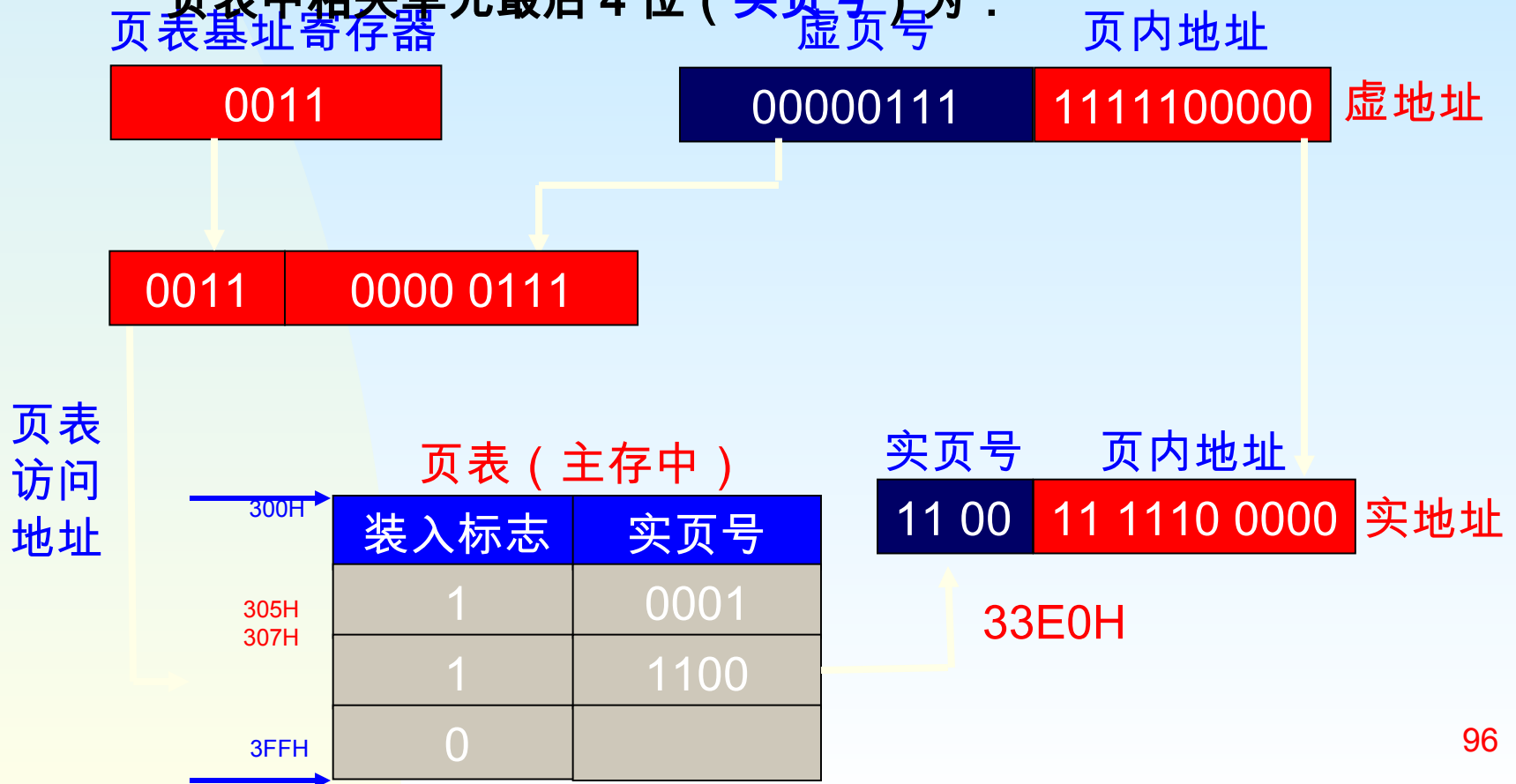
，  
假如页表的每一记录为 16 位，页表容量为  $4\text{M} \times 2\text{B}=8\text{MB}$ 。



# 页式虚拟存储器

**例 3**、采用页式虚拟存储器，页表索引地址由页表基址寄存器和虚页号拼接而成，已知某程序中一条指令的虚地址是 00 0001 1111 1110 0000，页表起始地址为 0011，页面大小 1K，

页表中相关单元最后 4 位 (实页号) 为：





# 页式虚拟存储器

**例 4**、在一个页式管理虚拟存储器中，虚拟存储器有 8 页，主存有 4 页。

1 ) 假设程序的第 0 个页面映象到主存的第 3 页，程序的第 2 个页面映象到主存的第 1 页，程序的第 3 个页面映象到主存的第 0 页

虚页号	装入标志	实页号
000	1	11
001	0	
010	1	01
011	1	00
100	0	
101	0	
110	1	10
111	0	

。试画出地址映象方式

# 页式虚拟存储器

2) 如果要访问的程序第 4 个页面映象到外存，那么如果将第 4 个页面调入主存，如何改变页表供 CPU 访问呢？

解：根据最近最少使用算法 LRU 算法，假如主存的第 3 个页框架是近期最少使用的，则第 3 个页框分配给第 4 个页面，在缺页中断下，启动输入输出系统，将虚地址指示的虚页整页调入主存，同时将页表内容改动如下：

调出 ←	虚页号	装入标志	实页号	调入 →	虚页号	装入标志	实页号
	000	1	11		000	0	
	001	0			001	0	
	010	1	01		010	1	01
	011	1	00		011	1	00
		0			100	1	11
	101	0			101	0	
	110	1	10		110	1	10
	111	0			111	0	

# 段式虚拟存储器

由于程序都具有一定的模块性，一个复杂的大程序可划分成多个逻辑上相对独立的模块，模块之间的界面和调用关系是可以清楚定义的。这些模块可以是子程序、过程、也可以是某类元素的集合。

在**段式虚拟存储器**中，**段是按照程序的逻辑结构划分的**，各段的长度因程序各异而不同。（因此，分段管理是按用户的要求提出来的。）

编程使用的**虚地址**包含两部分：高位是段号，低位是段内地址。

虚地址

段号

段内地址

为了把虚地址变换成主存实地址，需要一个**段表**；

**段表**也是一个段，驻留在**主存中**。

**段表**指明各段在主存中的位置：每段的**名称**、**段起点**、**段长度**等。

# 段式虚拟存储器

## 段式管理

程序逻辑空间 长度

段1 1K

段2 2K

段3 3K

段4 1K

段5 2K

实主存空间 地址



段大小不一样

(a)

段表

1	0	1	1K
2		0	
3	5K	1	3K
4		0	
5	1K	1	2K

段号

段起点

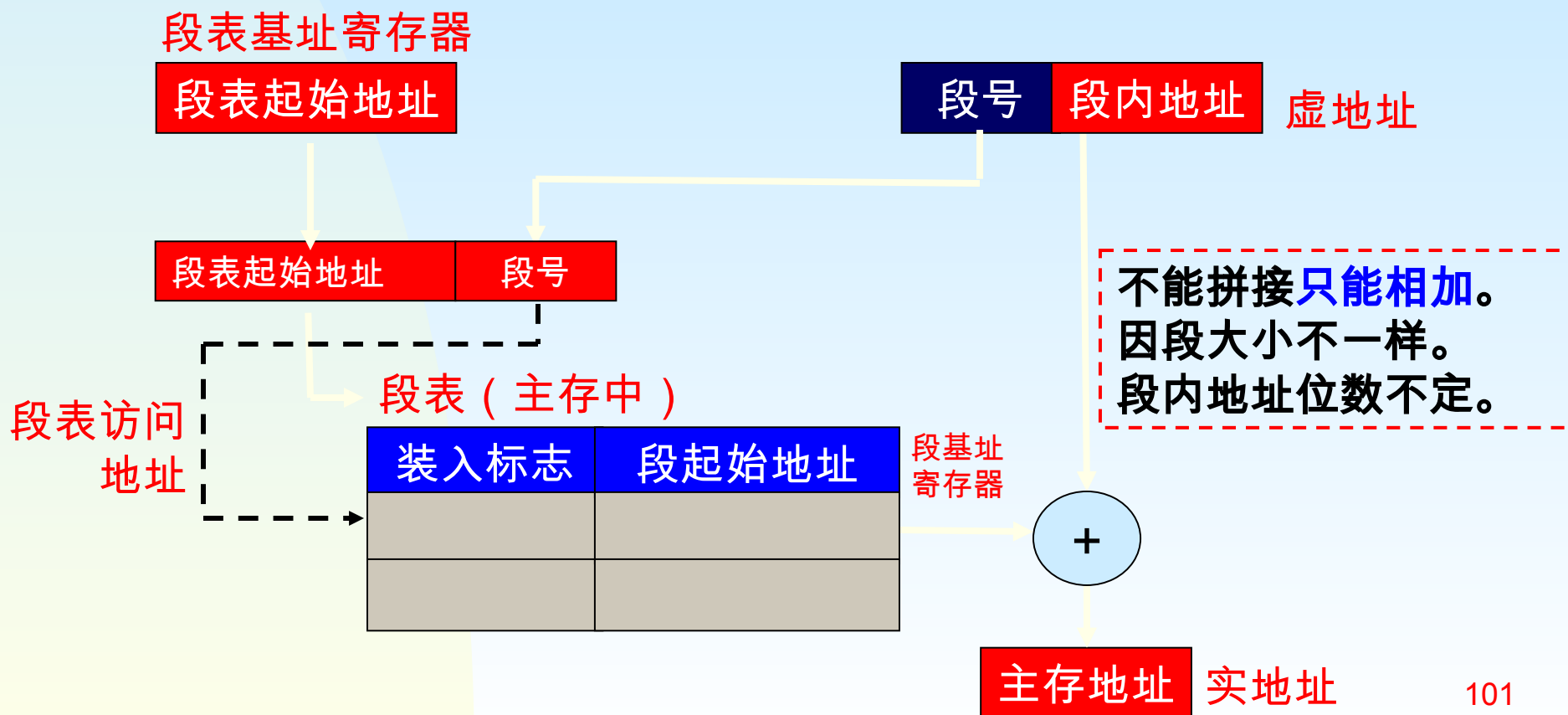
装入位

段长

(b)

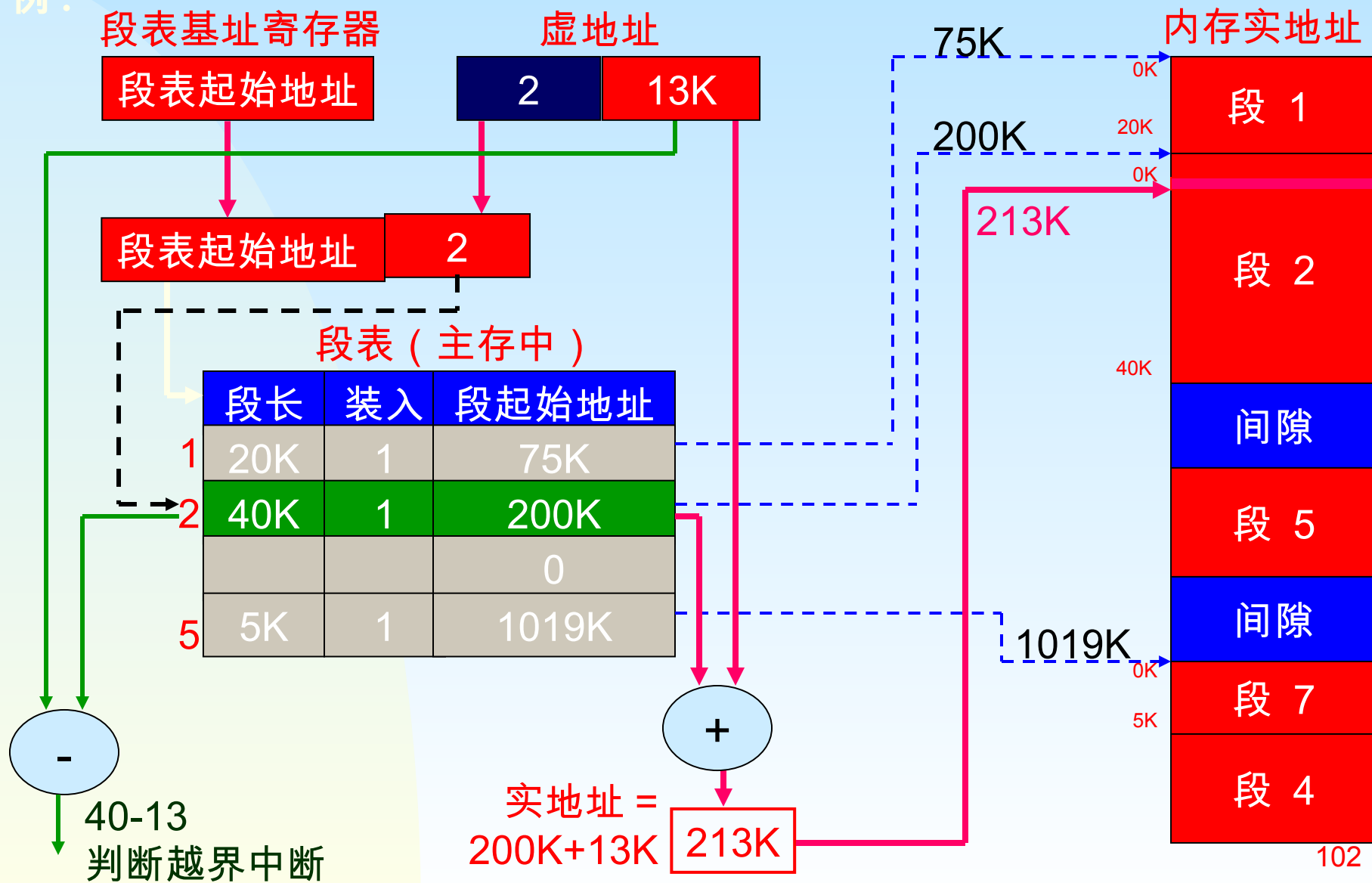
# 段式虚拟存储器

CPU 根据**虚地址访问**，**段表的起始地址 + 段号**，形成访问段表对应存储单元的地址，然后根据表内装入位判断该段是否已经调入主存，如果已经调入，从段表中读出该段在主存中的起始地址，与段内地址相加，得到对应的主存实地址。



# 段式虚拟存储器

例：



# 段式虚拟存储器

## 段式管理

**优点：**段的分界与程序的自然分界相对应，所以具有**逻辑独立性**，易于程序的编译、管理、修改和保护，也便于多道程序共享。某些类型的段具有动态可变长度，允许自由调度以便有效利用主存空间。

**缺点：**因为段的长度各不同，起点和终点不定，给主存空间分配带来麻烦，容易在段间留下**许多零碎的存储空余空间**，造成浪费。

页式存储器，碎片存在页内。（一个大程序可能分 4.6 个页面）  
段式存储器，碎片存在段外。（一个大程序分不同大小的段）

段外的碎片操作系统可以收集，页内碎片不好收集。

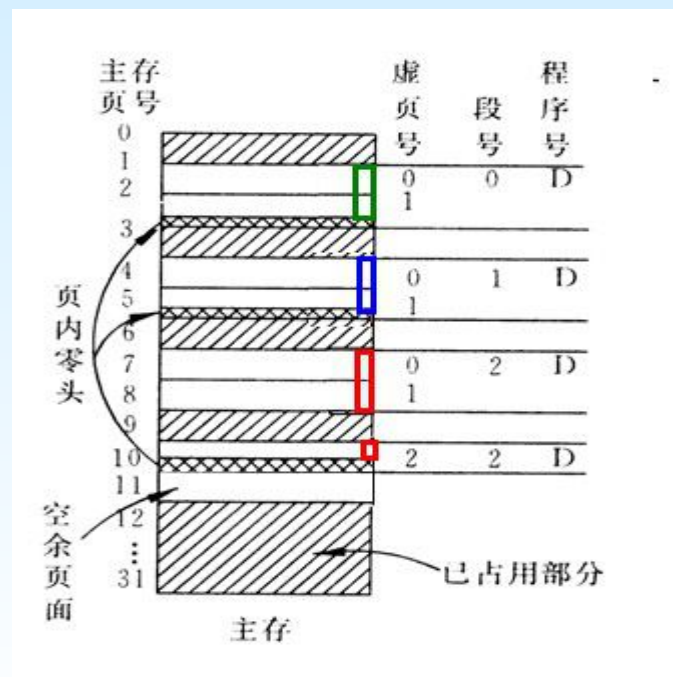
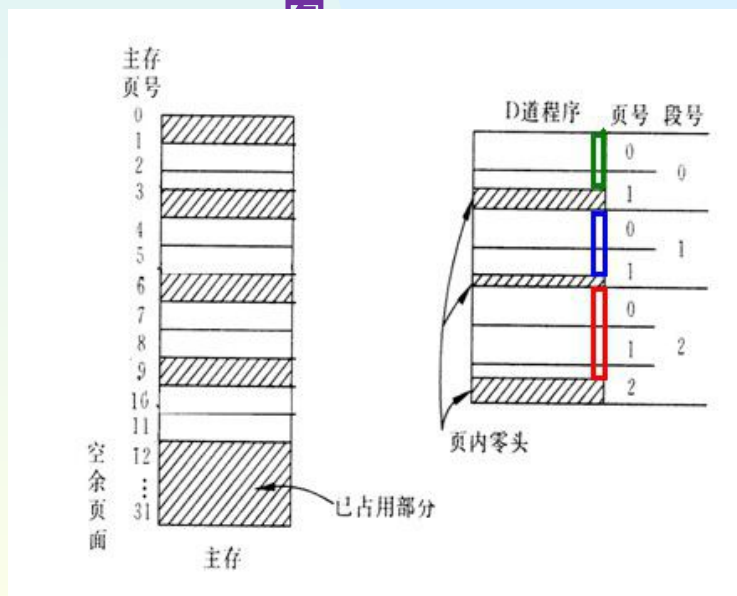
# 段页式虚拟存储器

**段页式管理**：结合**分段**和**分页**方法。

在**分段管理**方法中，存储器与硬盘交换数据是以**可变长的段**为单位。

对于容量比较大的段来讲，一次分配整个一段的主存空间不太灵活，因为这一大段程序中（现在执行的和没有执行的）都调入内存。

如：主存空间及 D 道程序示意



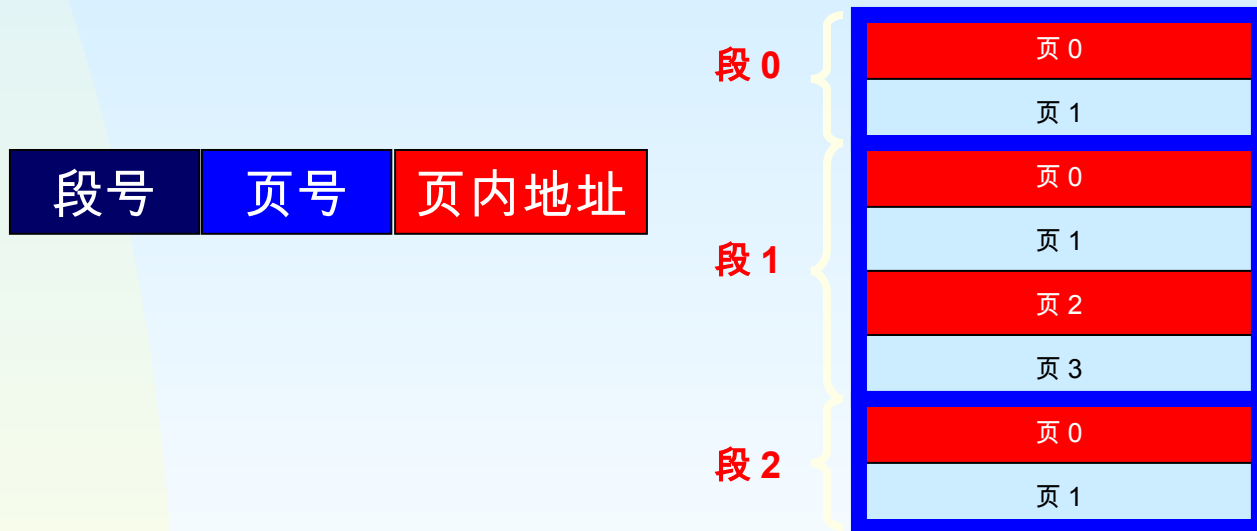


# 段页式虚拟存储器

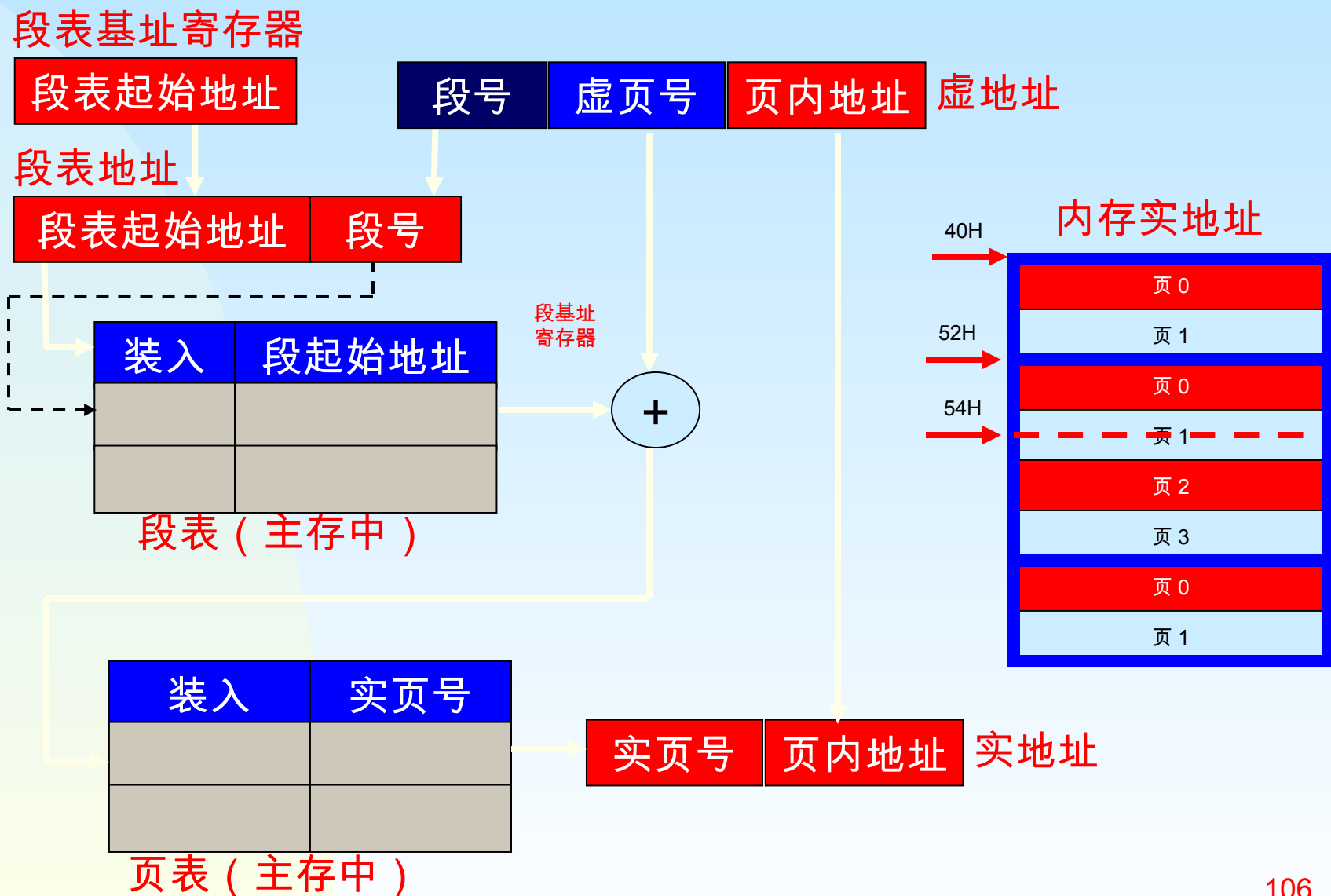
因此我们可以把程序按模块分段，段内再分页，进入主存以页为单位。

用段表和页表进行两级定位管理。

虚拟存储器地址分成 3 部分：  
段号，页号，页内地址。



# 段页式虚拟存储器



**物理存储器 ( 实存 )** : 真正在物理上存在的主存储器。

**物理地址 ( 实地址 )** : 访问主存的真实地址。

**虚拟存储器** : 在软件编程上可使用的存储器。

**虚拟存储空间** : 虚拟存储器的存储容量。

**逻辑地址 ( 虚拟地址 )** : 面向虚拟存储器的编程地址

。

**虚拟存储器不是磁盘存储器与物理存储器的简单拼合**  
:

**硬件支持** : 磁盘存储器, 存放暂时不用的信息;

**软件支持** : 操作系统实现主存与磁盘间的信息更换。  
更换过程对用户**透明**。

# 虚拟存储器的功能

1. 一种解决**存储容量**和**存取速度**矛盾的有效措施，是管理存储设备的有效方法。
2. 采用虚拟存储器，用户编制程序时就无需考虑所编程序在主存中**是否放得下**以及**放在什么位置**等问题。给软件编程提供了极大的方便。
3. 虚拟存储器使计算机具有辅存的容量，接近于主存的速度和辅存的位成本。
4. “主存—辅存”层次和“Cache—主存”层次采用的**地址变换和替换策略**，在原理上是类似的。

### (3) 对照

虚拟存储器和主存—Cache 存储器是两个不同存储层次的存储体系。

相同点：

- 1) 都把程序划分成一个个数据块（页、段等）；
- 2) 运行时自动地把数据块从慢速存储器向快速存储器调度；
- 3) 采用一定的替换策略，新的替换掉最不活跃的旧的，提高命中率（最近最少使用）。



## 不同点：

- 1) 基本信息的传递单位——数据块的**长度**不同；
- 2) Cache 弥补主存与 CPU 间的**速度差距**；虚拟存储器弥补了**主存容量**的不足；
- 2) CPU 与 Cache、主存都有**直接访问的通路**；辅存与 CPU 间**无直接通路**；
- 2) 主存—Cache 存储器中，存取信息、地址变换、内容替换等过程都由**硬件**实现，对**各类程序员**均是**透明**的；

虚拟存储器中，由存储管理软件辅助硬件进行信息块的划分、调度、地址变换等，对设计存储管理软件的**系统程序员**是**不透明**的，对**应用程序员**是**透明**的。



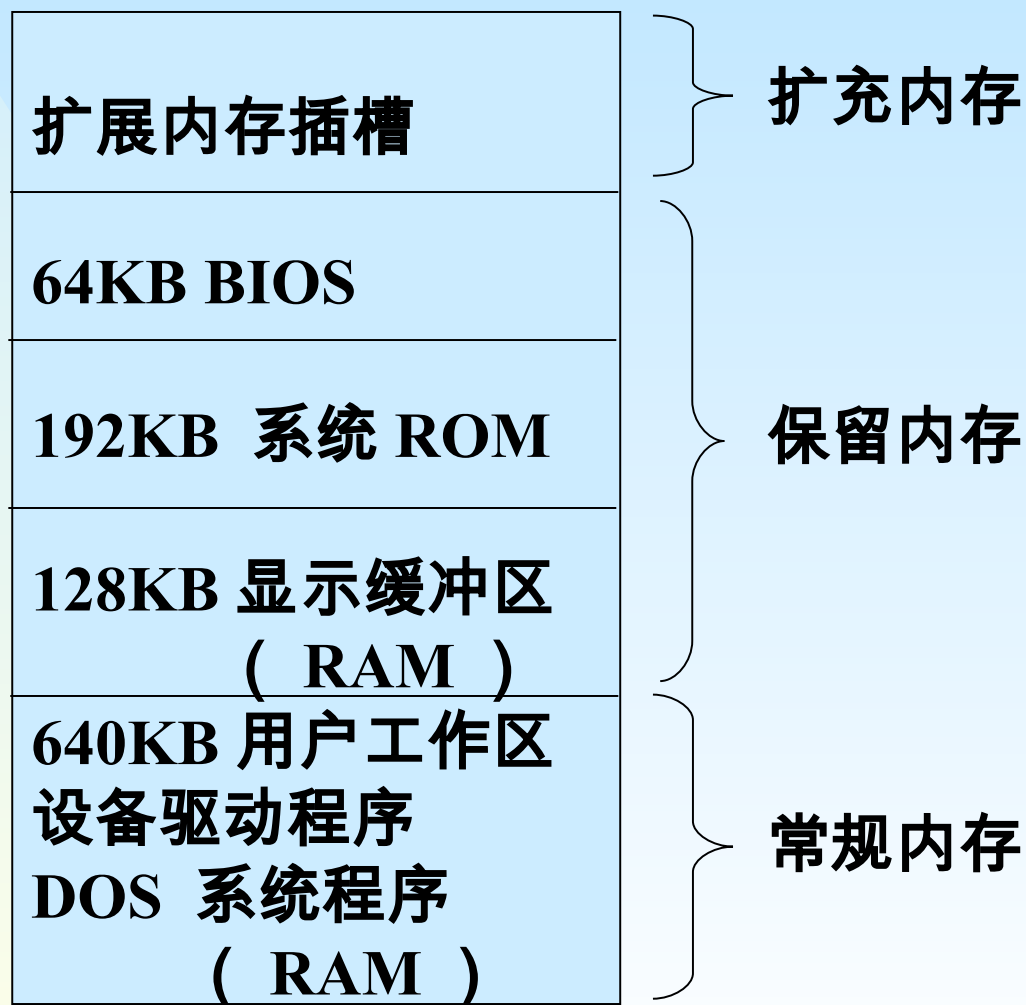
**THANKS**

北京科技大学

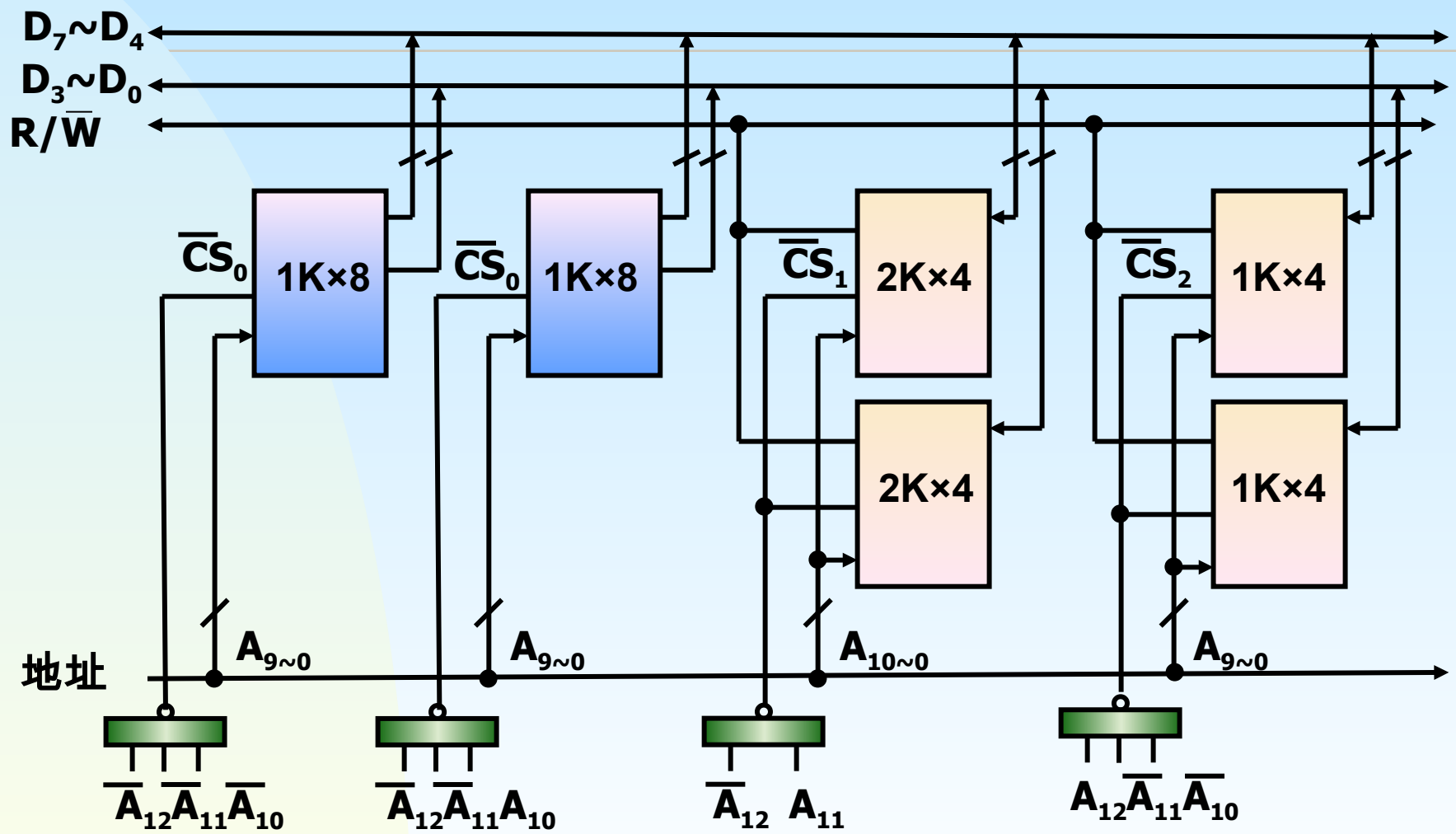
A wide-angle photograph of the Beihang University (Beihang University) campus. In the center, a large, multi-story, light-colored building with many windows is visible. In front of the building, there is a large, white, classical-style statue of a person standing on a pedestal. The statue is surrounded by a circular garden with various plants and flowers. The foreground is filled with lush green trees and bushes. The sky is blue with some white clouds. The overall scene is a well-maintained university campus.

# 微机中的内存管理

## DOS 方式下的内存管理







# 磁盘存储器的组成

- 1) 盘片 ( 存储体 );
- 2) 磁盘驱动器 ( 软盘、硬盘、温盘 );
- 3) 磁盘控制器与接口——串并转换、数据缓冲等。

磁盘 { 金属基体硬盘  
塑料基体软盘

## 温彻斯特磁盘存储器——温盘

- 1) 盘片固定、磁头可移动的磁盘存储器，用密封组合式结构，具有防尘性好、可靠性高等优点，应用广泛。
- 2) 用于 IBM PC 系列机的温盘一般是 3.5 吋的，现在容量可达 120GB 以上。

### 硬盘驱动器：

- 1) 精密的机电装置，由定位驱动系统、主轴系统和数据转换系统组成；
- 2) 加工和安装有严格的技术要求，在超净环境下组装；
- 3) 磁盘驱动器的逻辑操作归纳为寻址、读盘和写盘。

# 磁盘结构

**盘组**：为提供更大的存储容量，将多个盘片组装在同一主轴上，同轴旋转。

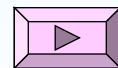
**记录面**：盘片表面，有双面和单面之分。

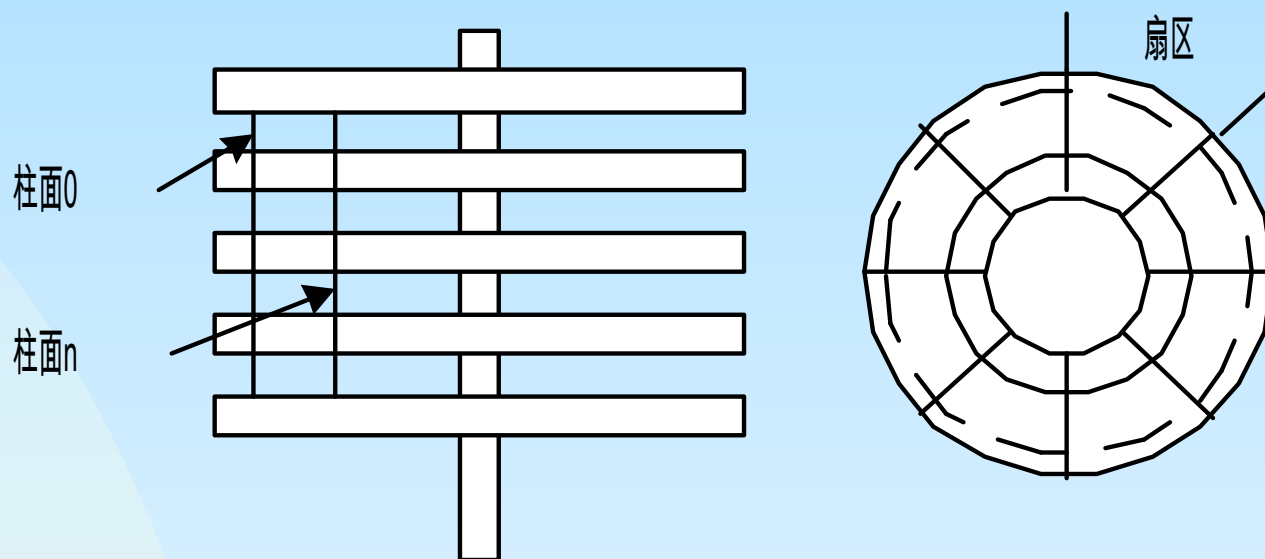
每个记录面上都有一个**磁头**，在寻道时同步运动。

**磁道** (track)：记录面上一系列同心圆。**由外向内**从0开始编号。

**柱面** (Cylinder)：盘组所有记录面上**序号相同**的磁道构成的圆柱面。**柱面号**与**磁道号**相同；

**扇区** (Sector)：一个磁道沿圆周划分为若干扇区，存放固定长度的数据块；





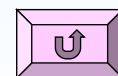
磁盘的柱面和扇区示意图

## 磁盘地址格式：

驱动器号	磁道号 (柱面号)	记录面号 (磁头号)	扇区号
------	-----------	------------	-----

## 定长记录格式：

每个扇区记录一个数据块，每个记录块记录的字节数固定。如 IBM PC 中每记录块记录 128、256 或 512 字节。



# 磁表面存储器的性能指标

1. 记录密度
2. 存储容量
3. 工作速度

# 1. 记录密度

可用**道密度**和**位密度**来表示。

**磁道**：盘片旋转时被磁头扫过的轨迹。一个磁表面会有许多磁道。

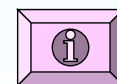
**道密度**：沿半径方向上，单位长度内的磁道数目。

**单位**：TPI( 每英寸磁道数 )。

**位密度**：在磁道中，单位长度内存放的二进制信息的位数。

**单位**：bPI( 每英寸二进制位数 )。

定长记录格式，位密度不固定。



## 2. 存储容量

**未格式化容量** = 记录面数 × 理论柱面数 × 内圆周长 × 位密度

**格式化容量** = 记录面数 × 每面磁道数 × 每道扇区数 × 每扇区字节数。

**例**：某软盘，

$$2 \text{ 面} \times 80 \text{ 道} \times 15 \text{ 扇区} \times 512\text{B} = 1.2\text{MB}$$



### 3. 工作速度

#### 磁盘工作顺序：

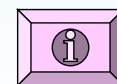
启动——寻道——寻找起始扇区——磁头读写

存取时间与信息所在磁道扇区的位置有关：

平均寻道时间（磁头运动）；

平均旋转延迟时间（盘片旋转）；

数据传输率——磁盘驱动器的读写速度。



# 软磁盘存储器与硬磁盘存储器的差别

- 1) 硬盘（温盘）转速高，磁头与磁层**不接触**；而软盘转速低，磁头与磁层**接触**。
- 2) 大多数硬盘采用固定**盘组**，软盘**单片**使用。
- 3) 硬盘系统价格高，存储量大，存取速度快；软盘价廉，存储量小，存取速度较慢。
- 4) 硬盘盘片**不可拆卸**，一般不能互换，软盘是**可拆卸**的且可互换。

# 光盘存储器

采用**聚焦激光束**在盘形介质上**高密度**地记录信息的存储装置。

## 1. 只读型光盘

由生产厂家预先用激光在盘片上蚀刻而成，信息不能改写。如 CD、CD-ROM，已相当普及。

## 2. 一次写入型光盘

由用户一次写入、可多次读出但不能擦除的光盘。

3. **可重写型光盘**：读 / 写型光盘。按记录介质不同可分为磁光型、相变型两大类。

**磁光型**记录技术：既有光记录信息的**高密度**，又有磁记录介质的**可擦除重写**特点，在可重写型光盘中占据重要地位。