

东南大学计算机学院

计算机系统组成

主讲教师： 徐造林

第5章 存储系统

- **存储器**是存放（指令）**程序和数据**的部件，是计算机系统的重要组成部分。
- 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**由各类存储设备及有关的软件所构成。

5.1 存储器概述

◆ 存储器的基本概念

- **存储介质**：能表示二进制1和0的物理部件；
- **存储元**：存储1位二进制代码信息的器件；
- **存储单元**：若干个存储元的集合，它可以存放一个字或一个字节；
- **存储体**：若干个存储单元的集合；
- **地址**：存储单元的编号；
- **存储容量**：能存放的二进制位数或字节数；
- **存储器的速度**：可用访问时间、存储周期或频宽来描述。

5.1.1 存储器分类

1. 按存储介质分类

- 存储元件必须具有两个截然不同的物理状态，才能被用来表示二进制代码“0和1”。
- 半导体存储器
- 磁性材料存储器
- 光介质存储器

2. 按存取方式分类

(1) 顺序存取存储器（SAM）

- 信息顺序存放或读出，其存取时间取决于信息存放位置；
- 以记录块为单位编址；
- 磁带存储器就是一种顺序存储器，它存储容量大，但存取速度慢。

(2) 随机存取存储器 (RAM)

- CPU或I/O设备在任一时刻都可按地址访问其任何一个存储单元，且访问时间与地址无关，都是一个存取周期。
- 半导体存储器一般属于这类存储器。

(3) 直接存取存储器 (DAM)

- 存取方式介于RAM和SAM之间，先选取需要存取信息所在的区域，然后用顺序方式存取；
- 磁盘属于直接存取存储器，它的容量也比较大，速度则介于SAM和RAM之中，主要用作辅存。

(4) 只读存储器 (**ROM**)

- 在正常读写操作下，这类存储器的内容只能读出而不能写入。
- 有的**ROM**位于主存中特定区域（如**IBM-PC**机中**ROM BIOS**）其访问方式和**RAM**一样按地址访问；
- 也有的**ROM**用作辅存，采用顺序访问方式，例如**CDROM**。

3. 按存储器在计算机中的功能分类

(1) 高速缓冲存储器（Cache）

- 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。
- 其存取速度能接近CPU的工作速度，用来临时存放指令和数据。

(2) 主存储器

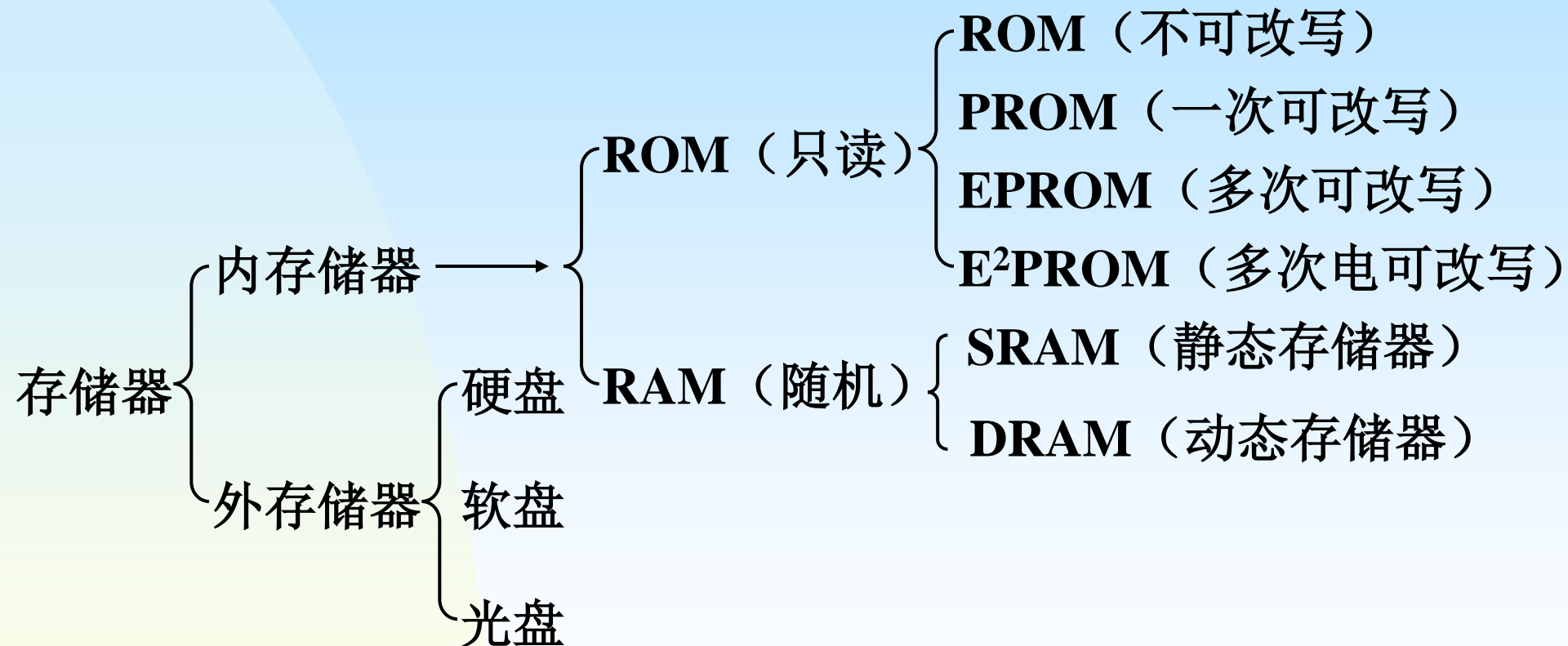
- 用来存放计算机运行时的大量程序和数据，主存储器目前一般用CMOS半导体存储器构成。

- CPU能够直接访问的存储器称**内存储器**，**高速缓存**和**主存**都是内存储器。

(3) 辅助存储器（外存储器）

- **外存储器**主要由磁表面存储器组成；光存储器渐成为一种重要的辅助存储器。
- 外存储器的内容需要调入主存后才能被CPU访问。
- 外存储器容量大，可存放大量的程序和数据。

- 存储器的分类简图



5.1.2 存储器的主要性能指标

1) 容量

- 存储器的**容量**指它能存放的二进制位数或字（字节）数；
- 单位有B（字节Byte），KB（千字节，K为 2^{10} ），MB（兆字节，M为 2^{20} ），GB（千兆字节，G为 2^{30} ），TB（兆兆字节，T为 2^{40} ）等。

2) 速度

- ▲ 存储器的速度可用访问时间、存储周期或频宽来描述；
- ▲ 访问时间：用读出时间 T_A 及写入时间 T_W 来描述；
 - T_A 是从存储器接到读命令以后至信息被送到数据总线上所需的时间；
 - T_W 是将一个字写入存储器所需的时间。

- ▲ **存取周期** (T_M) 是存储器进行一次完整的读写操作所需要的全部时间;
 - 常用存储器进行连续读写操作的最短间隔时间;
 - T_M 直接关系到计算机的运算速度;
一般有 $T_M > T_A$ 、 $T_M > T_W$ ，单位用微秒或毫微秒。
- ▲ **存储器的频宽B**: 表示存储器被连续访问时，提供的数据传送速率;
 - 常用每秒钟传送信息的位数（或字节数）来衡量。

3) 价格

- ▲ 存储器的价格：可用总价格C或每位价格c来表示，若存储器按位计算的容量为S；
 - 则： $c = C/S$

5.1.3 存储器结构

1. 存储系统的层次结构

- ▲ 存储系统的层次结构是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；
- 程序和数据按不同的层次存放在各级存储器中，使整个存储系统具有较好综合性能指标。

- ▲ 由二类存储器构成的存储系统层次结构

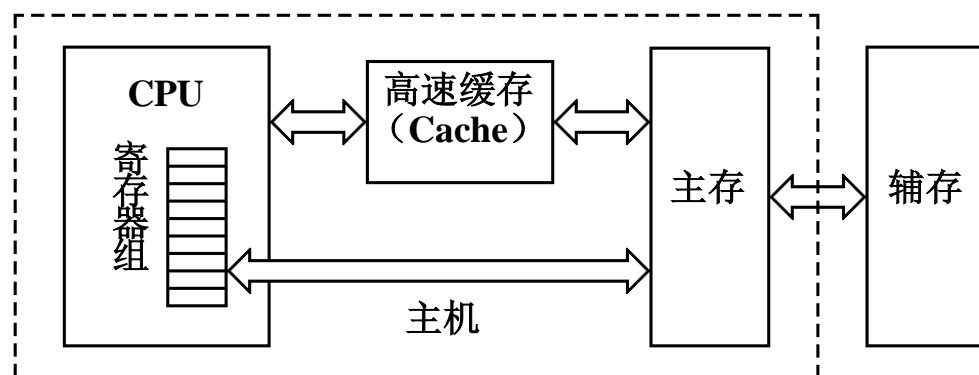


图5.1 存储器系统的层次结构

(1) “高速缓存—主存”层次

- 这个层次主要解决存储器的**速度**问题；
- 在CPU与主存之间增设一级存储器，称**高速缓冲存储器（Cache）**；
- **Cache**速度可与CPU相匹配，但容量较小，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，若在**Cache**中找到相应内容，称访问“**命中**”，信息就从**Cache**中读取；
- 否则CPU从主存中读取（称访问“**不命中**”）；此时一般要进行**Cache**和主存的信息交换。

◆ 提高存储器速度的措施

- ① 采用高速器件；
- ② 采用**cache**（高速缓冲存储器）；
- ③ 采用多体交叉存储器；
- ④ 采用双端口存储器；
- ⑤ 采用相联存储器，加长存储器的字长。

◆ 多体交叉存取

- ▲ 把主存分为若干容量相同、能独立地由CPU进行存取的存储体。
- 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频宽。

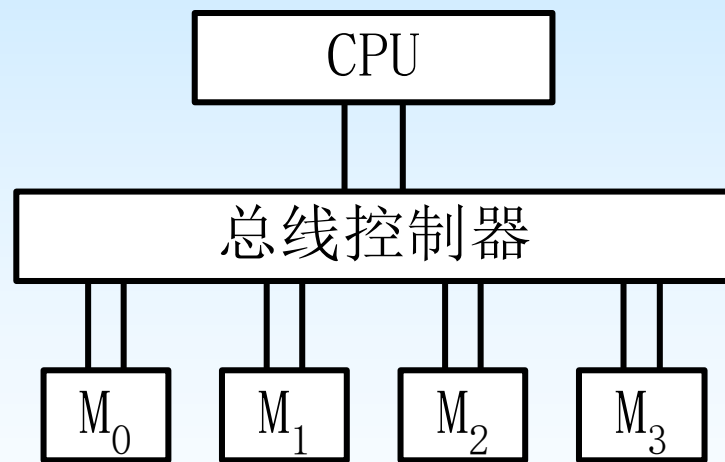


图5.2 多体交叉存储系统

(2) “主存—辅存” 层次

- 这个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，其余信息存放在容量大、但速度较慢的辅存中。
- “主存—辅存”层次是一个既具有主存的存取速度又具有辅存的大容量低成本特点的一个存储器总体。
- 虚拟存储技术面对程序员的是一个具有辅存的容量、主存的速度存储器；解决了主存容量不足的问题。

5.1.4 主存储器的编址和与CPU的连接

◆ 存储单元及其编址

- ▲ 赋予存储单元惟一的编号，以二进制数表示；称为**地址**或地址码。
- ▲ 能访问的存储单元数目，称为**地址空间**；由地址码的位数决定。
- ▲ 目前计算机主存的编址大多**按字节编址**。

◆ 主存与CPU的连接

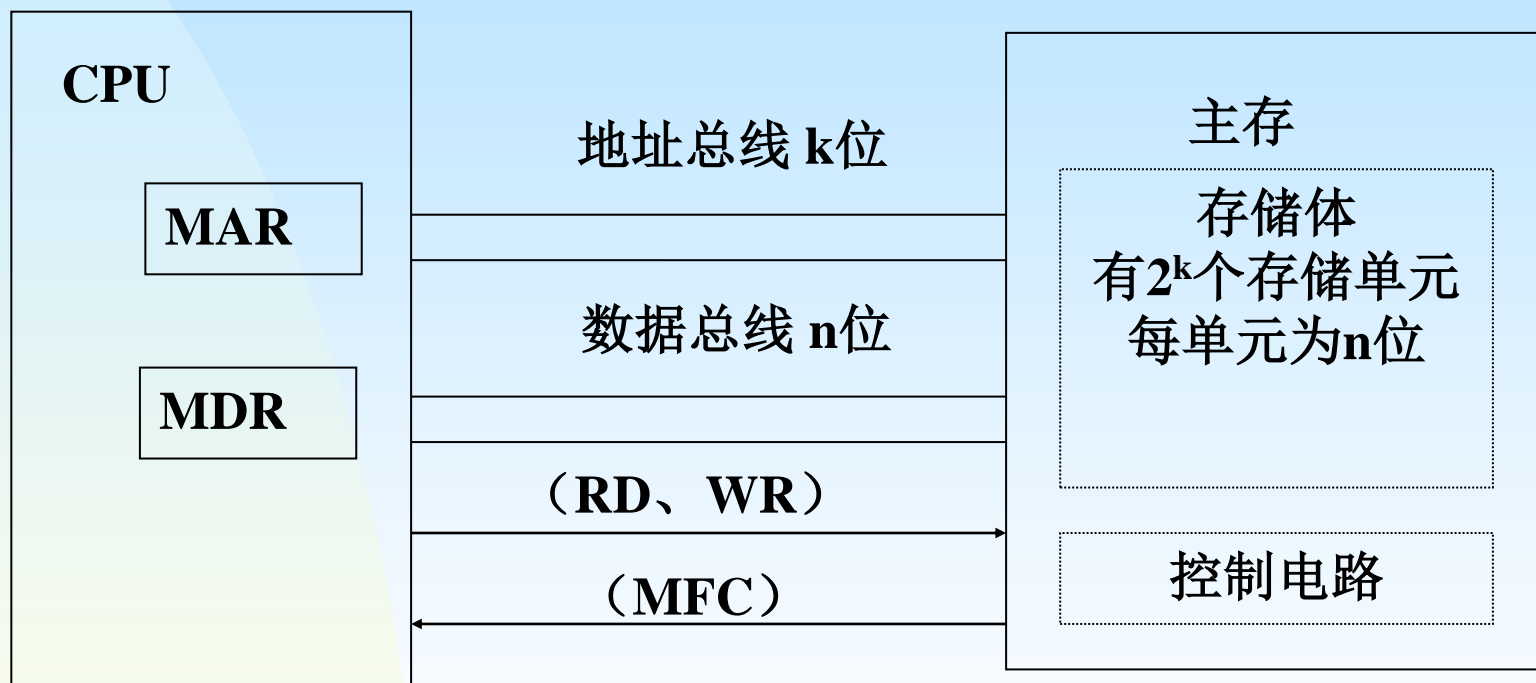


图5.3 主存与CPU的连接

5.2 半导体随机存储器



5.2.1 静态存储器 (SRAM)

1. 静态存储单元

(1) 保持状态

- 字选线低电位， T_3 与 T_4 截止，触发器与外界隔离。

- 保持“1”态：

→ A高→ T_2 导通→B低→ T_1 截止→

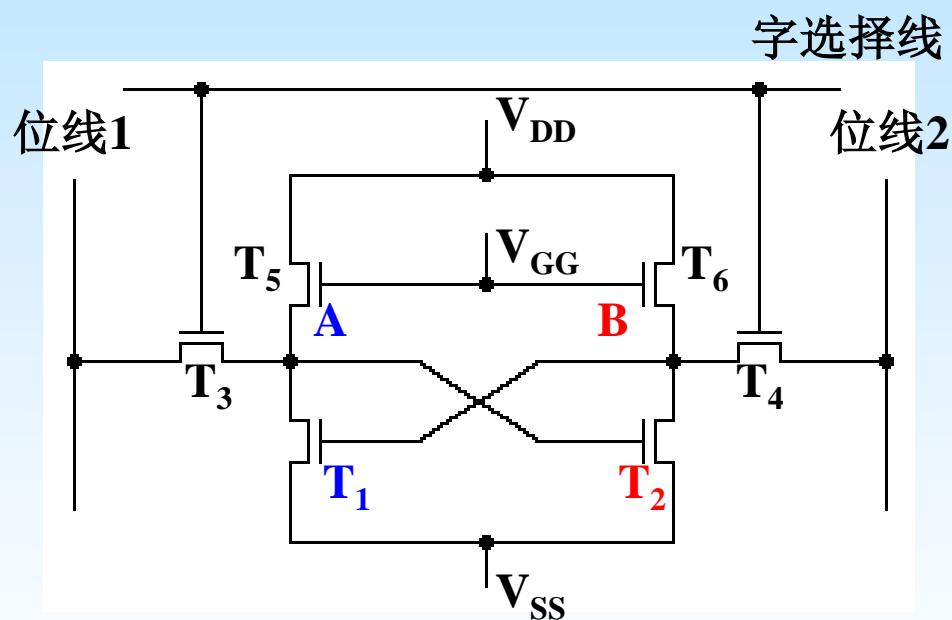


图5.4 六管MOS静态存储器的存储单元

(2) 读出

- 字选线加高电位， T_3 与 T_4 开启；使电路读出A、B信息。

(3) 写入

- 字线上加高电位， T_3 与 T_4 开启；若要写1，在位线2上加低电位。
- 若要写0，在位线1上加低电位。

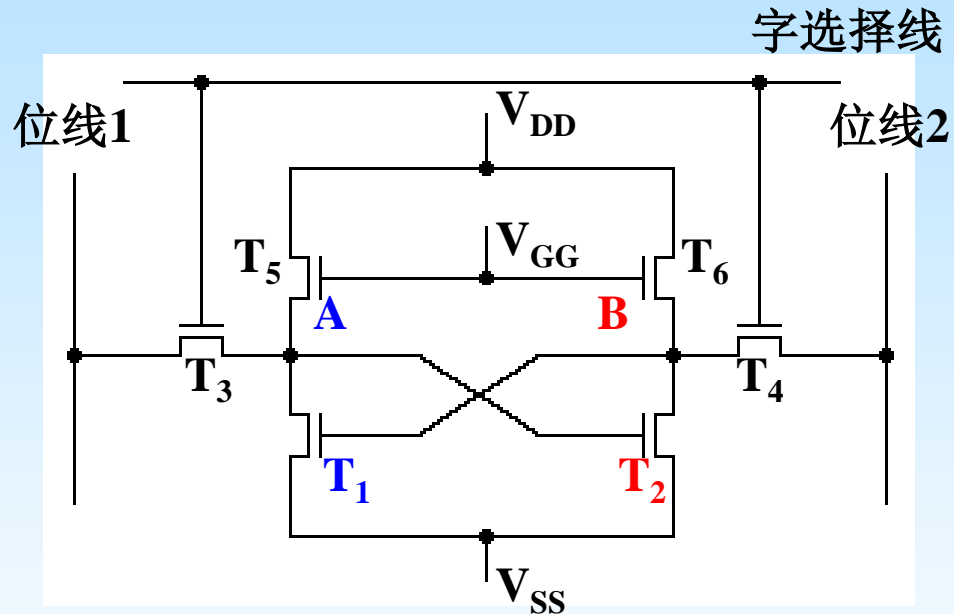


图5.4 六管MOS静态存储器的存储单元

2. 静态MOS存储器

(1) 静态MOS存储器组成

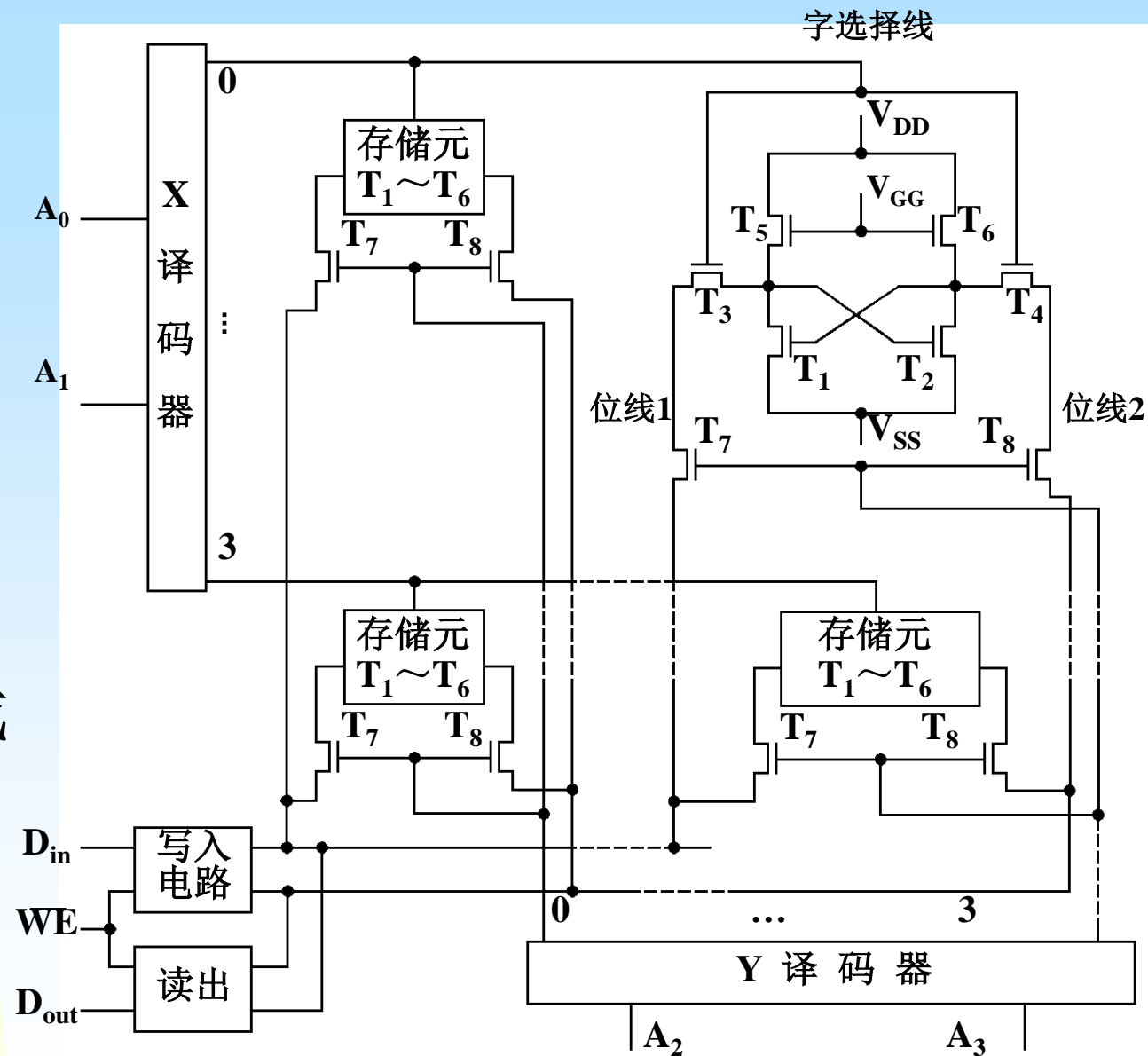


图5.5 MOS静态存储器结构图

(2) 静态MOS存储器芯片

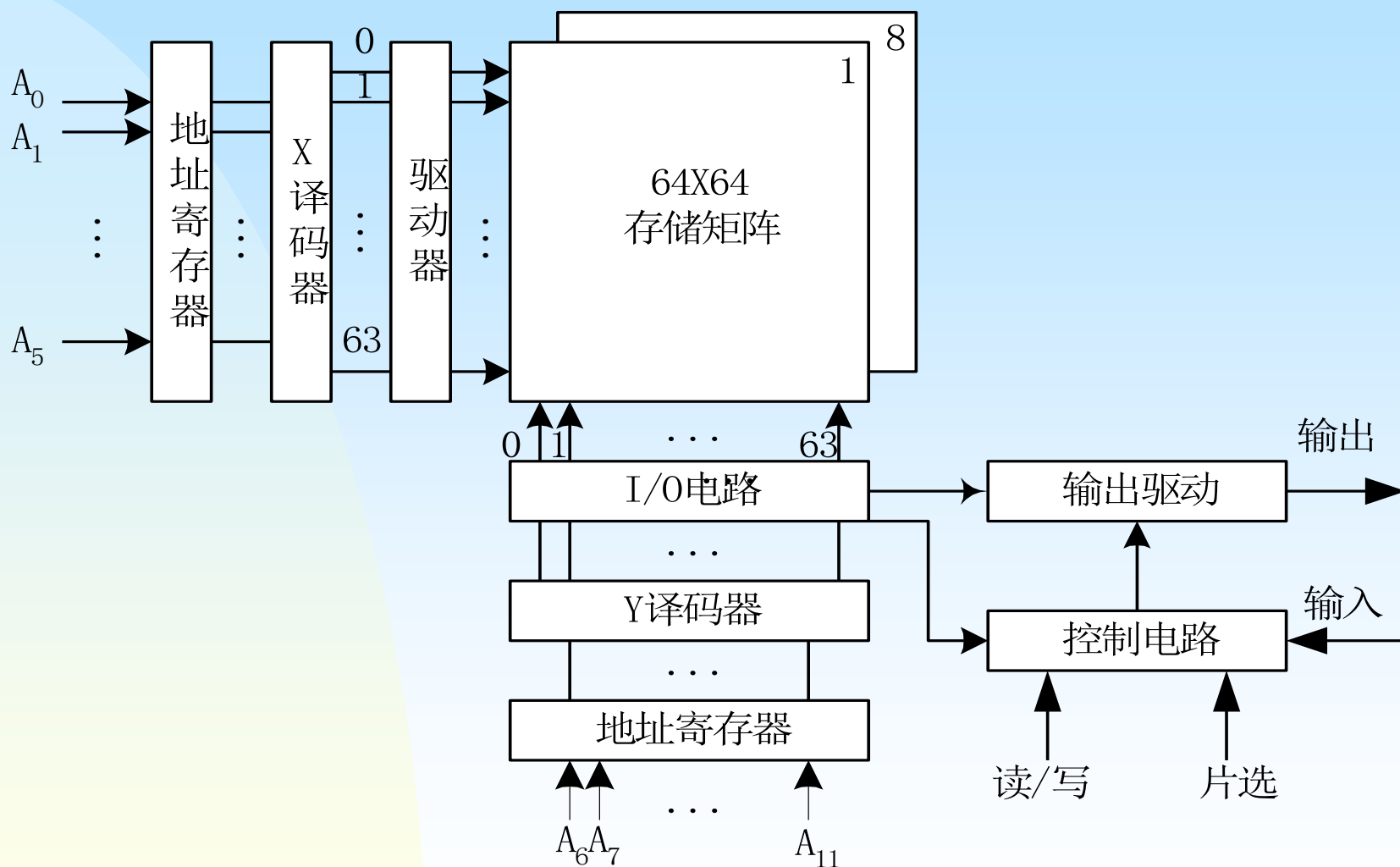


图5.6 静态MOS RAM芯片结构图

① 存储体（存储矩阵）

- **存储体**是存储单元的集合。在容量较大的存储器中往往把**各个字的同一位**组织在一个集成片中；
- 图5.6中的芯片是 4096×1 位，由这样的8个芯片可组成4096字节的存储器。
- 4096个存储元排成 64×64 的矩阵。由X选择线(行选择线)和Y选择线(列选择线)来选择所需用的单元。

② 地址译码器

- **地址译码器**把用二进制表示的地址转换为**译码输入线上的高电位**，以便驱动相应的读写电路。

▲ 两种地址译码方式：

- 一种是**单译码方式**，适用于小容量存储器；
- ⊙ 地址译码器只有一个，其输出叫字选线，**选择某个字的所有位**。
- ⊙ 地址输入线 $n=5$ ，经地址译码器译码后，产生32个字选线，分别对应32个地址。

- 另一种是**双译码方式**，适用于容量较大的存储器
- ⊙ 地址译码器分为X和Y两个译码器。每一个译码器有 $n/2$ 个输入端，可以**译出 $2^{n/2}$ 个状态**，两译码器交叉译码的结果，可**产生 $2^{n/2} \times 2^{n/2}$ 个输出状态**；
- ⊙ 图5.7是采用双译码结构的 **4096×1 位**的存储元矩阵；对**4096**个单元选址，需要**12**根地址线： $A_0 \sim A_{11}$ 。

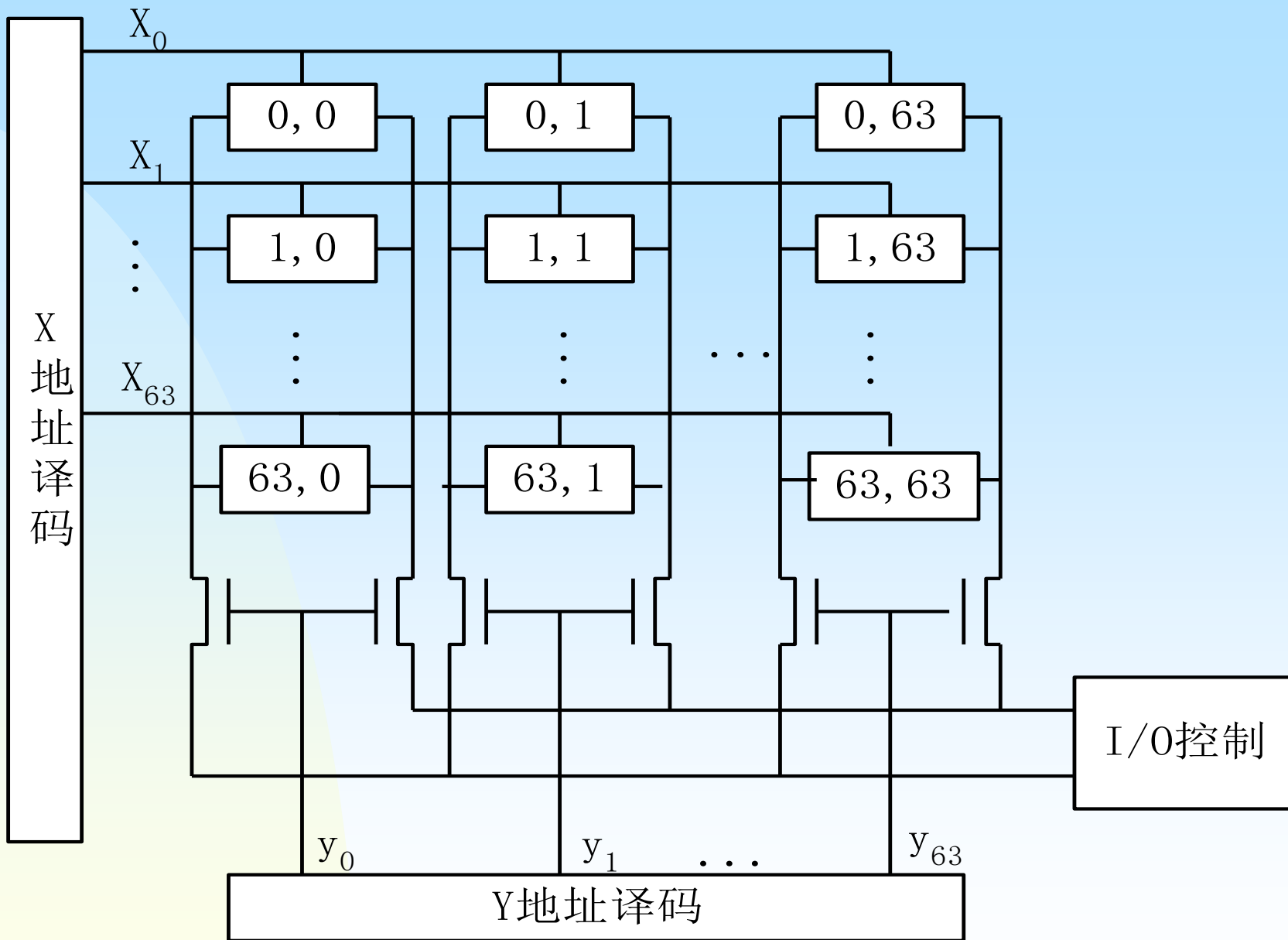


图5.7 双地址译码存储结构

③ 驱动器

- 一条X方向的选择线要控制在其上的各个存储单元的字选线，负载较大，要在译码器输出后加驱动器。

④ I/O控制

- 它处于数据总线和被选用的单元之间，用以控制被选中的单元读出或写入。

⑤ 片选控制

- 芯片外的地址译码器产生片选控制信号，选中要访问的存储字所在的芯片。

⑥ 读/写控制

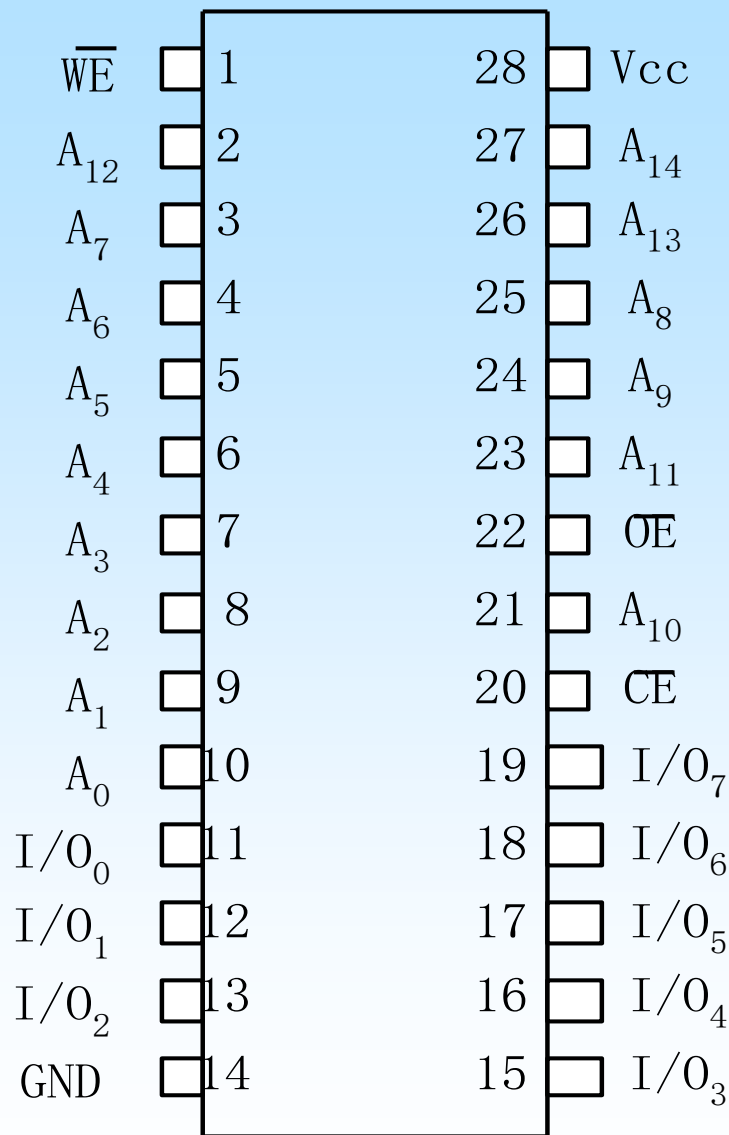
- 根据CPU给出的信号是读命令还是写命令，控制被选中存储单元的读写。

(3) 静态MOS存储器芯片 实例(62256 SRAM)

▲ 该芯片容量为 $32K \times 8B$

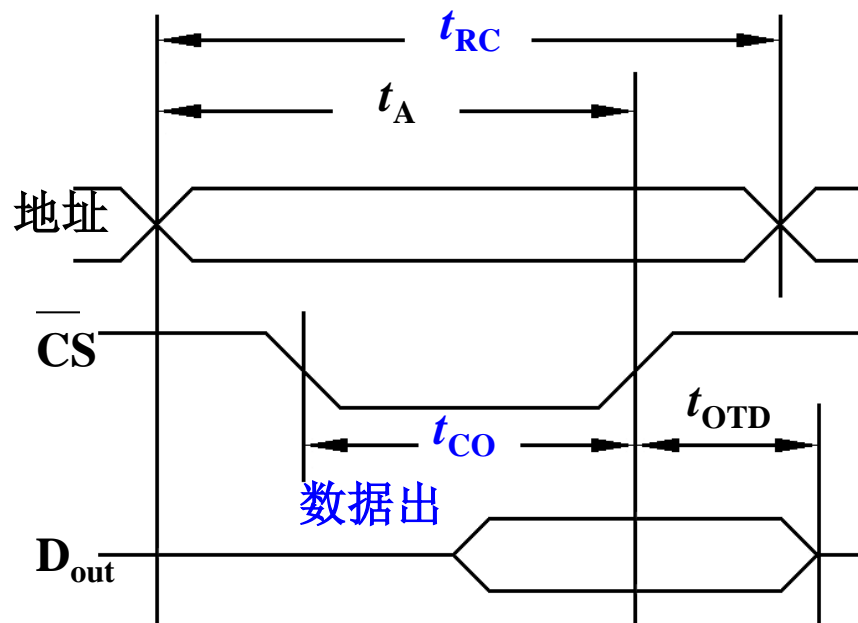
▲ 62256 SRAM芯片引脚

- 地址引脚: $A_0—A_{14}$
- 数据引脚: $I/O_0—I/O_7$
- 片选: \overline{CE} 低电平有效
- 读/写控制: \overline{WE} , 低电平时为写入控制; 高电平时为读出控制。

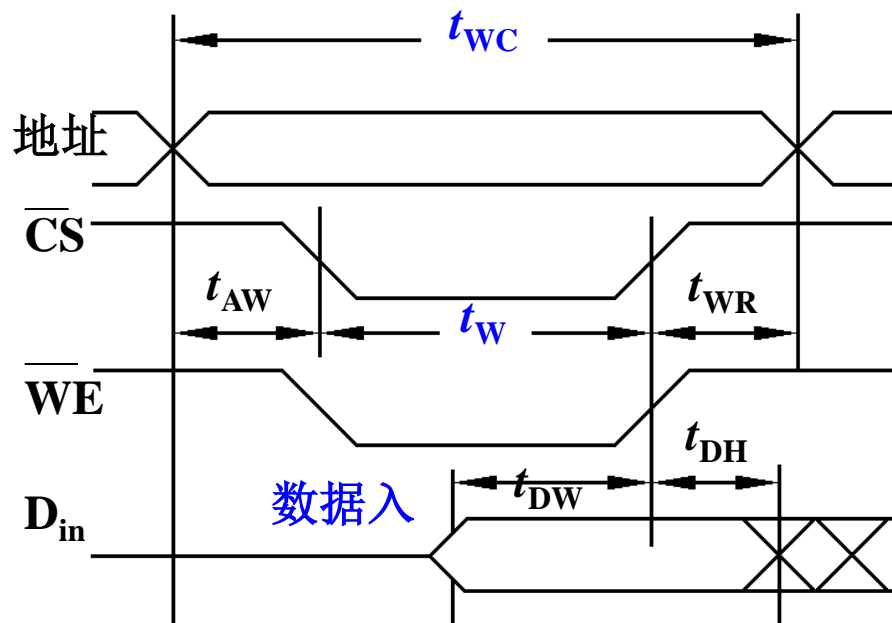


| 图5.8 M62256

(4) 静态存储芯片的读/写周期



(a) 读周期



(b) 写周期

图5.9 静态RAM芯片的读、写周期

5.2.2 存储器的基本组织

◆ 存储器容量的扩展：位扩展、字扩展和字位扩展

1. 位扩展

- 用若干片位数较少的存储器芯片，构成具有给定字长的存储器，而存储器的字数与芯片上的字数相同。
- 由 $m \times n_1$ 位存储器芯片组成 $m \times n_2$ 位的存储器，需要 (n_2/n_1) 片的 $m \times n_1$ 位存储器芯片。
- 用8片 4096×1 位的芯片构成4KB的存储器；如图5.10所示。

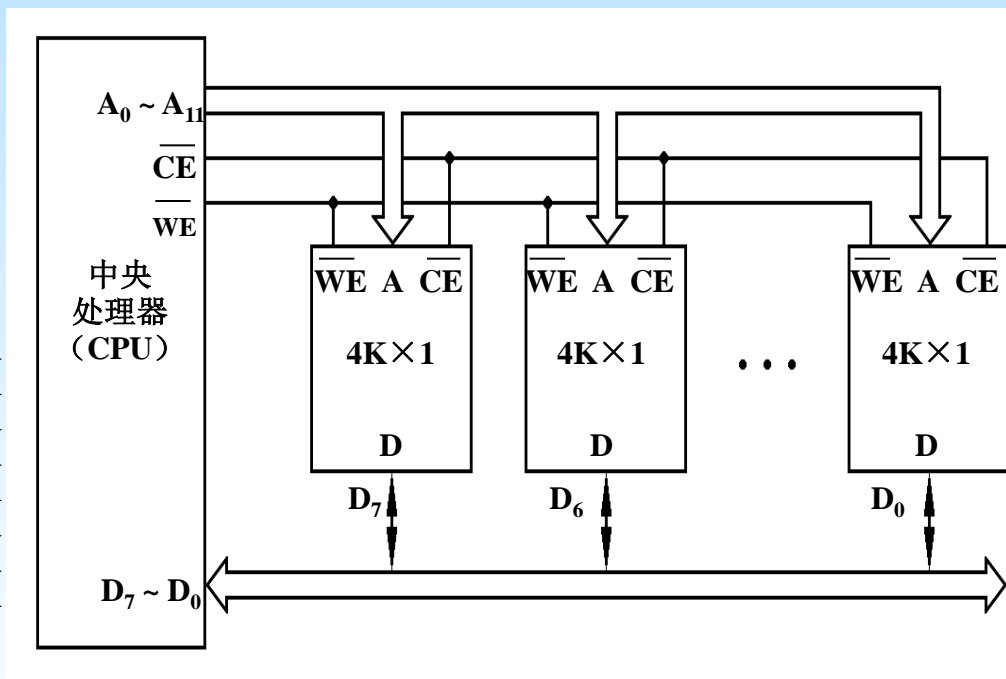
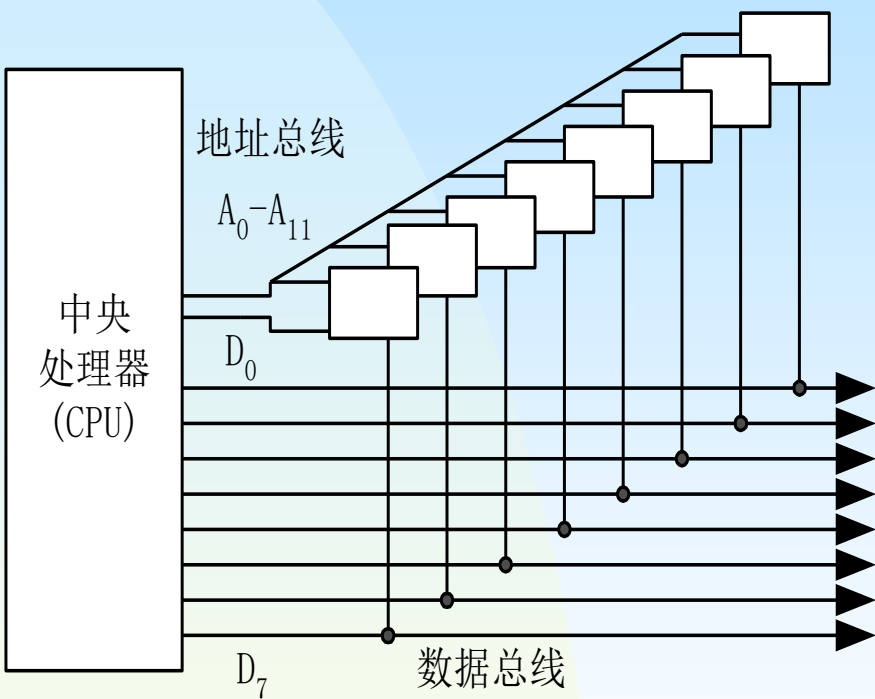


图5.10 位扩展构成的存储器

2. 字扩展

- **字扩展**是容量的扩充（地址线增加），**位数不变**。
- 由 $m_1 \times n$ 位存储器芯片组成 $m_2 \times n$ 位的存储器，需要 (m_2/m_1) 片的 $m_1 \times n$ 位存储器芯片。
- 用4片 $16K \times 8$ 位的存储器构成 $64K \times 8$ 的存储器 (图5.11)。

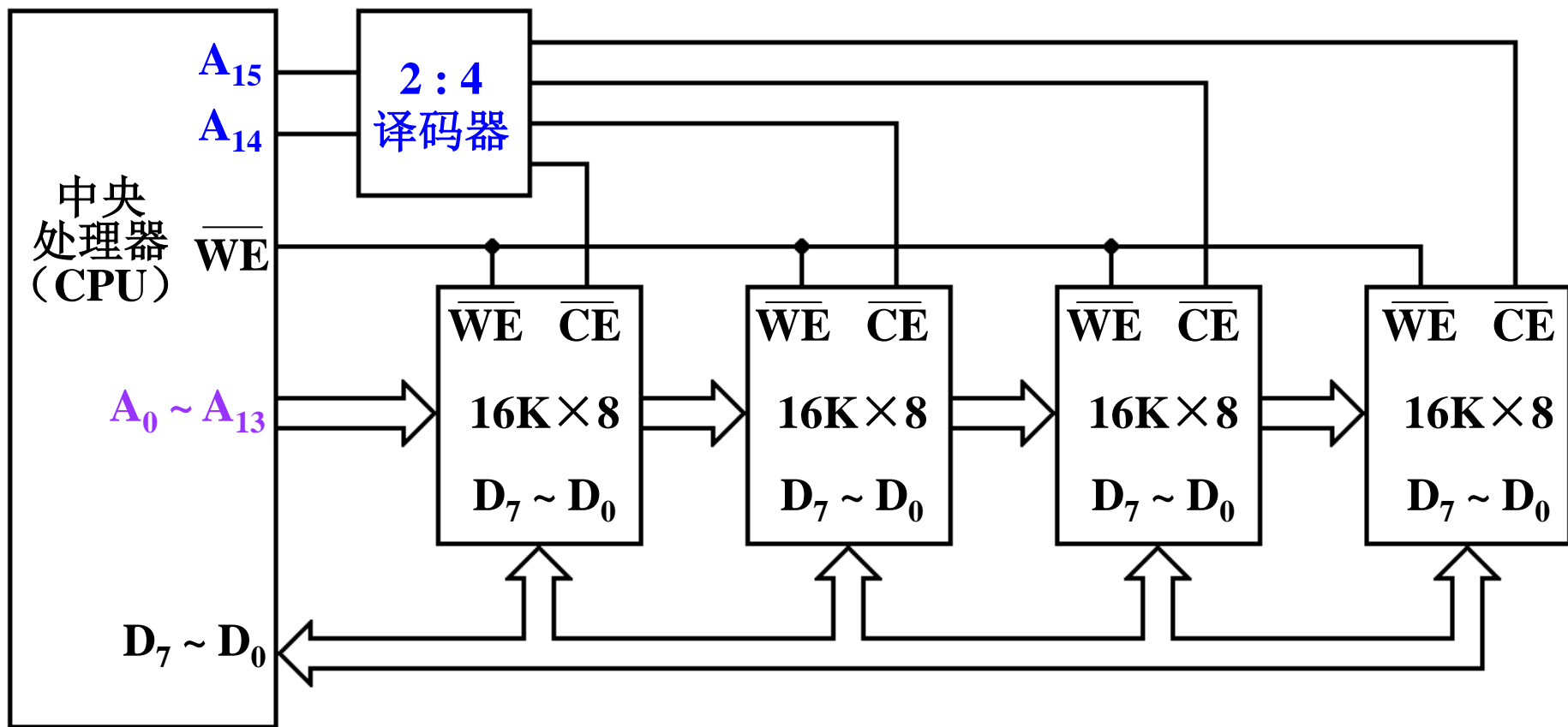


图5.11 字扩展构成的存储器

3. 字位同时扩展

- 由 $m_1 \times n_1$ 位存储器芯片组成 $m_2 \times n_2$ 位的存储器，需要 $(m_2/m_1) \times (n_2/n_1)$ 片的 $m_1 \times n_1$ 位存储器芯片。

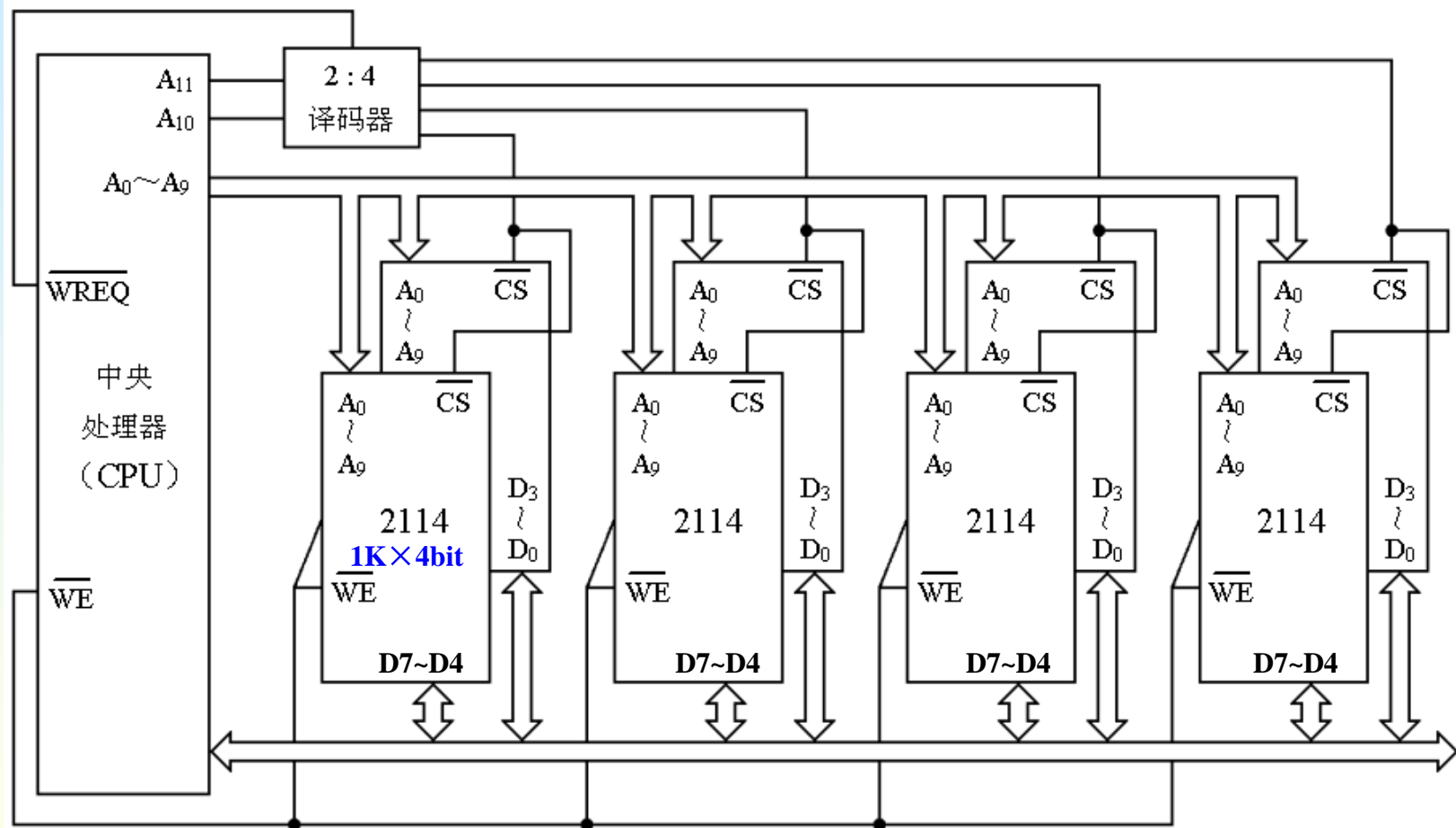


图5.12 字位扩展构成的存储器

4. 静态RAM芯片与CPU连接

- ① 地址线的连接，包括内部地址线和芯片选择线的连接；
- ② 数据线的连接，数据线对应相接；
- ③ 控制线的连接，控制线主要有读/写控制线 \overline{WE} 和存储器访问线 \overline{MREQ} 。

例1 某存储器数据总线宽度为32 bit，存取周期为250 ns。试问该存储器的带宽是多少？

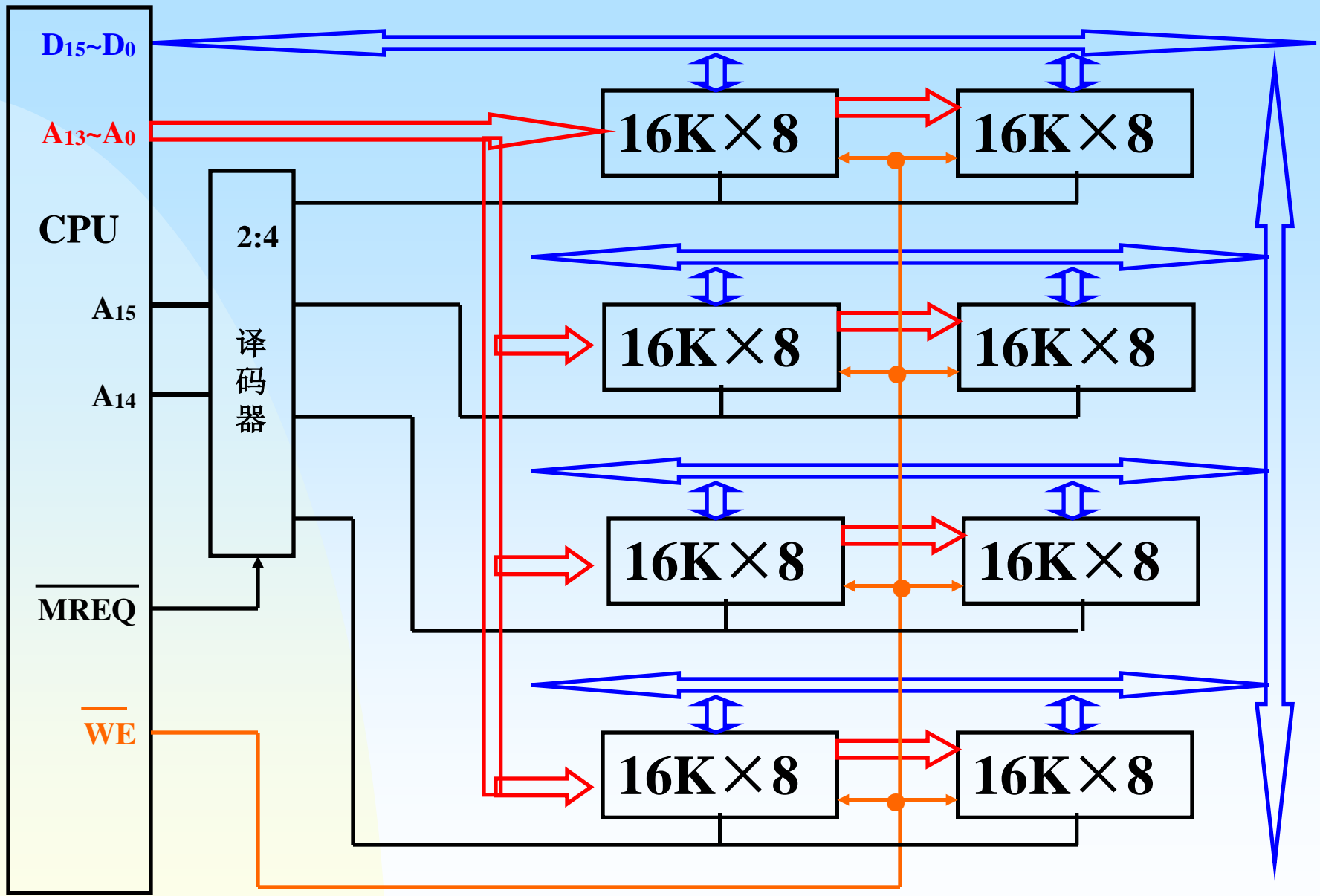
解 $T_m=250 \text{ ns}$ ， $W=32 \text{ bit}$

$$B_m = \frac{W}{T_m} = \frac{32 \text{ bit}}{250 \text{ ns}} = 128 * 10^6 \text{ bit/s} = 16 * 10^6 \text{ B/s}$$

例2. 用 $16\text{K} \times 8$ 位的SRAM芯片构成 $64\text{K} \times 16$ 位的存储器，试画出该存储器的组成逻辑框图。

解：

- 存储器容量为 $64\text{K} \times 16$ 位，其地址线为16位($A_{15} \sim A_0$)，数据线也为16位($D_{15} \sim D_0$)；
- 组成存储器时需要字位同时扩展，字扩展采用2:4译码器，共4个模块；位扩展采用两片并接。



- 各组的字地址的范围？

▲ 连接时要考虑和解决的几个问题：

- CPU的负载能力：

- ⊙ 当存储芯片较多时，在CPU与存储芯片之间，要增加必要的缓冲和驱动电路。

- 速度匹配问题：

- ⊙ 存储器与CPU的速度相比，还是有很大差距；

- 多片存储芯片的选通：

- ⊙ 增加外部译码电路，产生片选信号；

- 读/写控制信号：

- CPU的读/写控制信号不一定与存储芯片引脚定义的控制信号相符，有时要增加某些附加线路来实现正确的控制。

5. 内存条及其类型

- DIMM内存条封装的标准；共有168根镀金接线，一次能够传送64位数据，容量已达到256MB或以上。

5.2.2 动态存储器 (DRAM)

- **动态RAM**利用MOS管的栅极电容来保存信息，在“信息保持”状态下，存储单元中没有电流流动。

1. 动态存储单元

(1) MOS管栅极电容保存信息的原理

- 在 $\phi = +E$ 时，由 V_I （反相）写入信息到 C_s 。

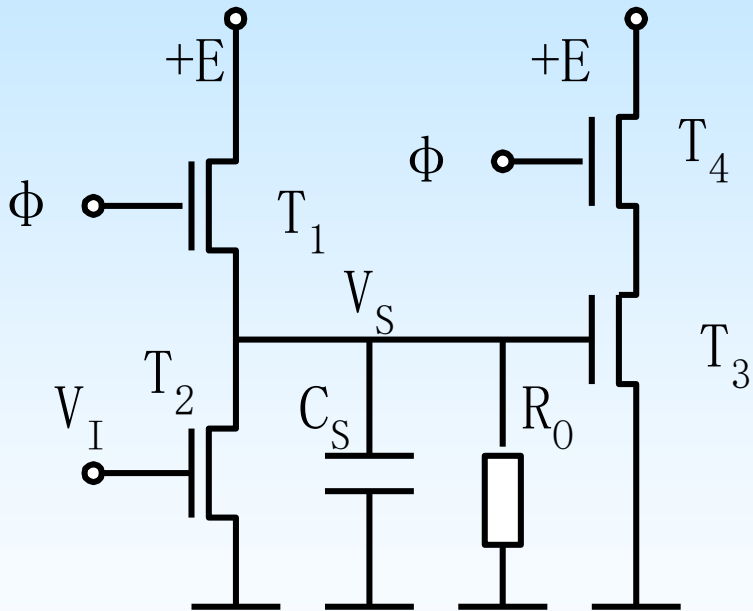


图5.13 MOS管栅极电容保存信息的原理

- 当 ϕ 变为 $0V$ ， $T_1 \sim T_4$ 均不导通， C_S 保持原有电平。
- 实际上 T_3 的输入阻抗(R_0)不可能为无穷大， C_S 上的电荷会缓慢泄放；

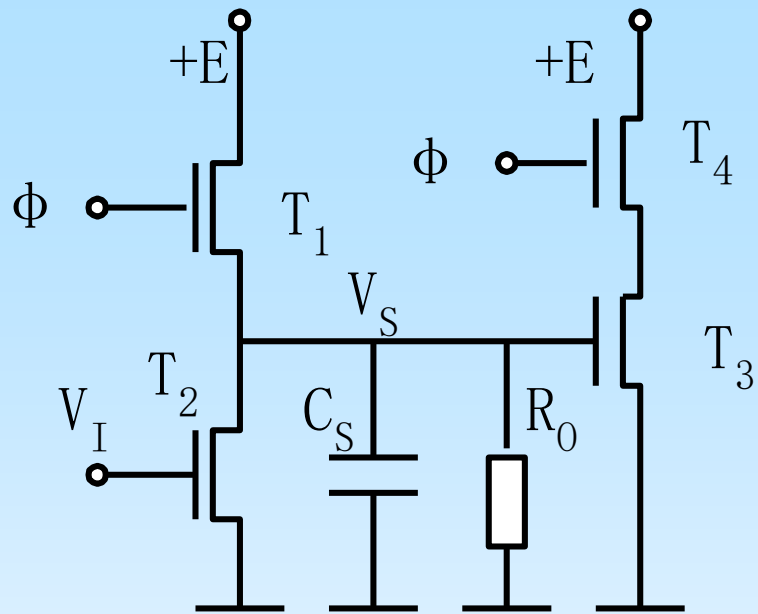


图5.14 MOS管栅极电容保存信息的原理

- 假定在 $t = t_R$ 时 V_S 会下降到 “1” 电平的临界值 V_1 以下，从而使保存的信息丢失；
- 保存时间 t_R 约为 $2ms \sim 3.3ms$ 。

(2) 四管动态RAM单元

① 信息保持:

- 行选信号无效时， T_3 、 T_4 ， T_1 、 T_2 均不导通。
- 假定 C_{S2} 上充有电荷， C_{S1} 未充电的状态为“存1”状态。

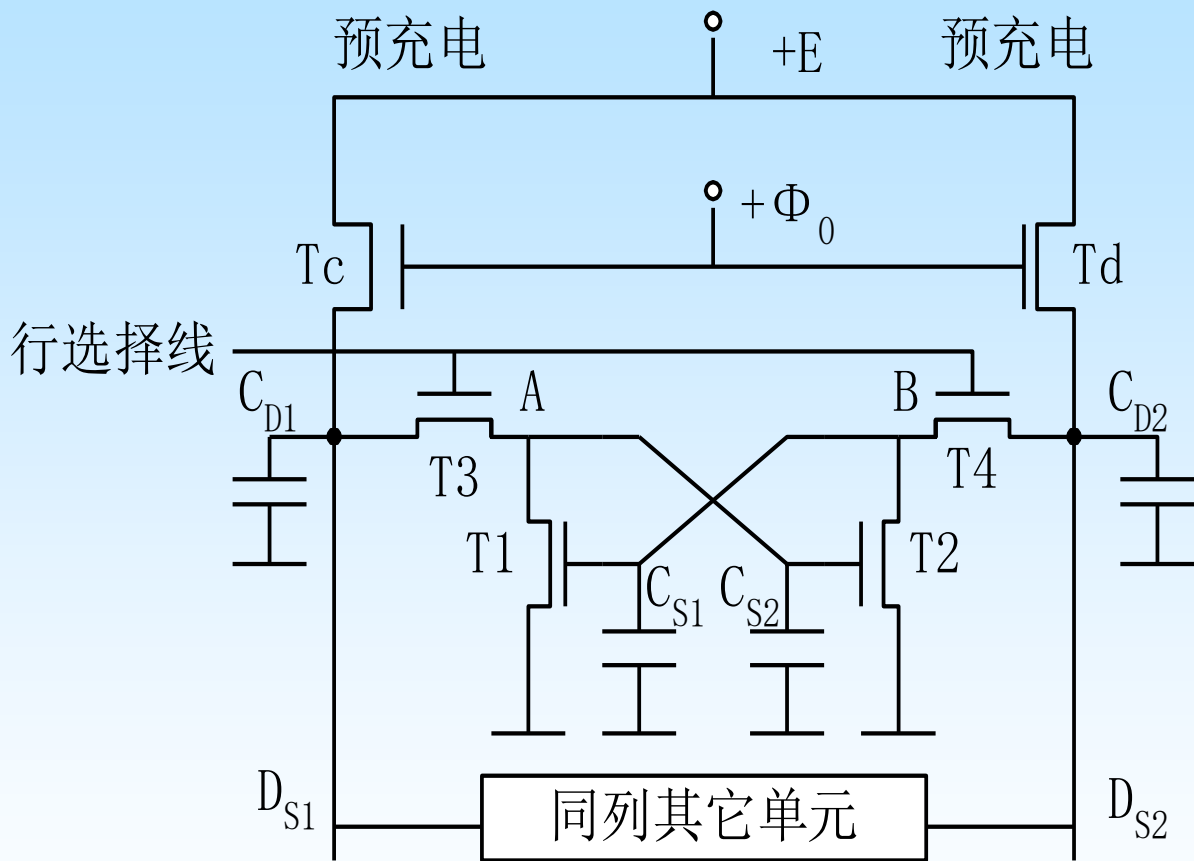


图5.15 四管动态MOS单元电路

② 读操作：

- 预充电：发 $+\Phi_0$ 脉冲，使 C_{D1} 和 C_{D2} 预充电。
- 令行选择线有效 行选择线， T_3 和 T_4 导通。
- D_{S1} 和 D_{S2} 的电位差使I/O电路产生读“1”或“0”信号。
- 以上过程读出是非破坏性的。

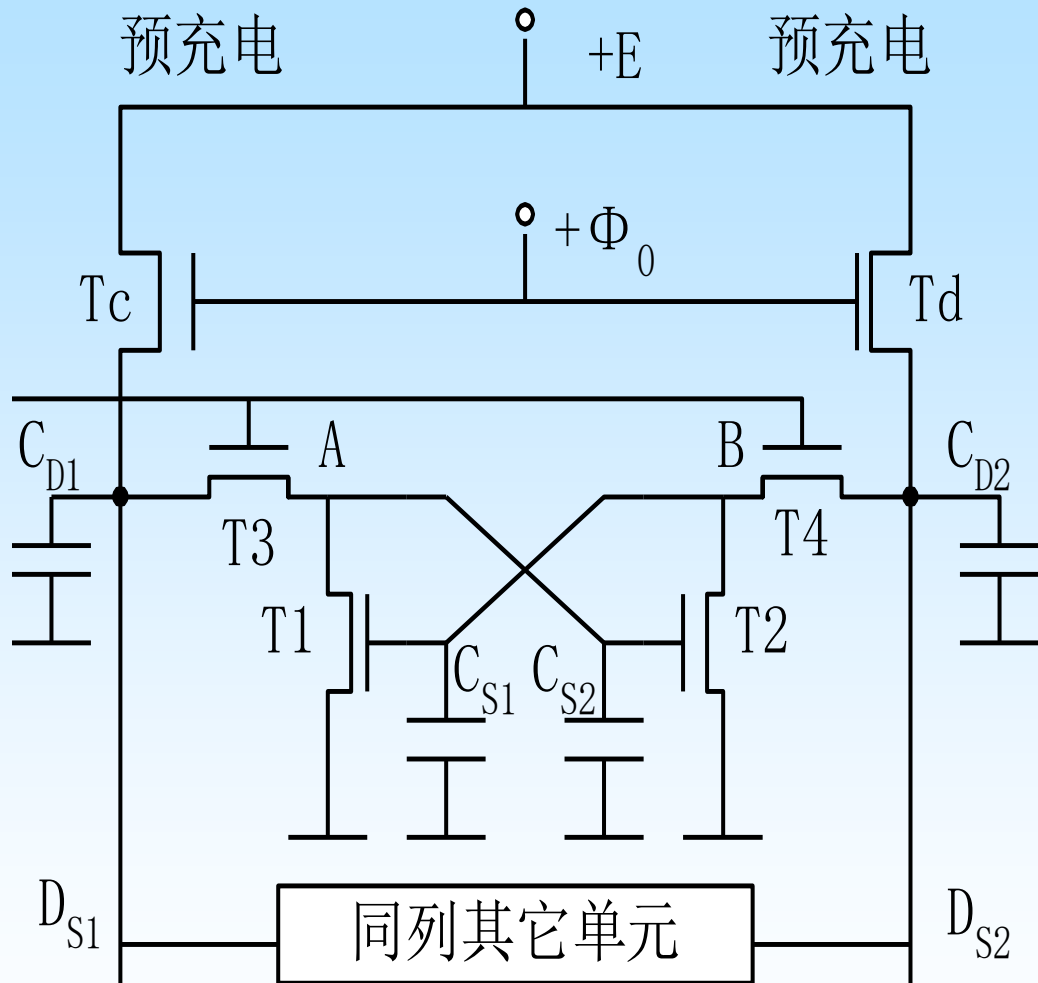


图5.16 四管动态MOS单元电路

③ 写操作:

- 先后送 $+\Phi_0$ 和行选择脉冲；
行选择线
- 若要写“1”，在 DS_1 上加高电位，在 DS_2 上加低电位。
- 若写“0”，则在 DS_1 线上加低电位， DS_2 线上加高电位。

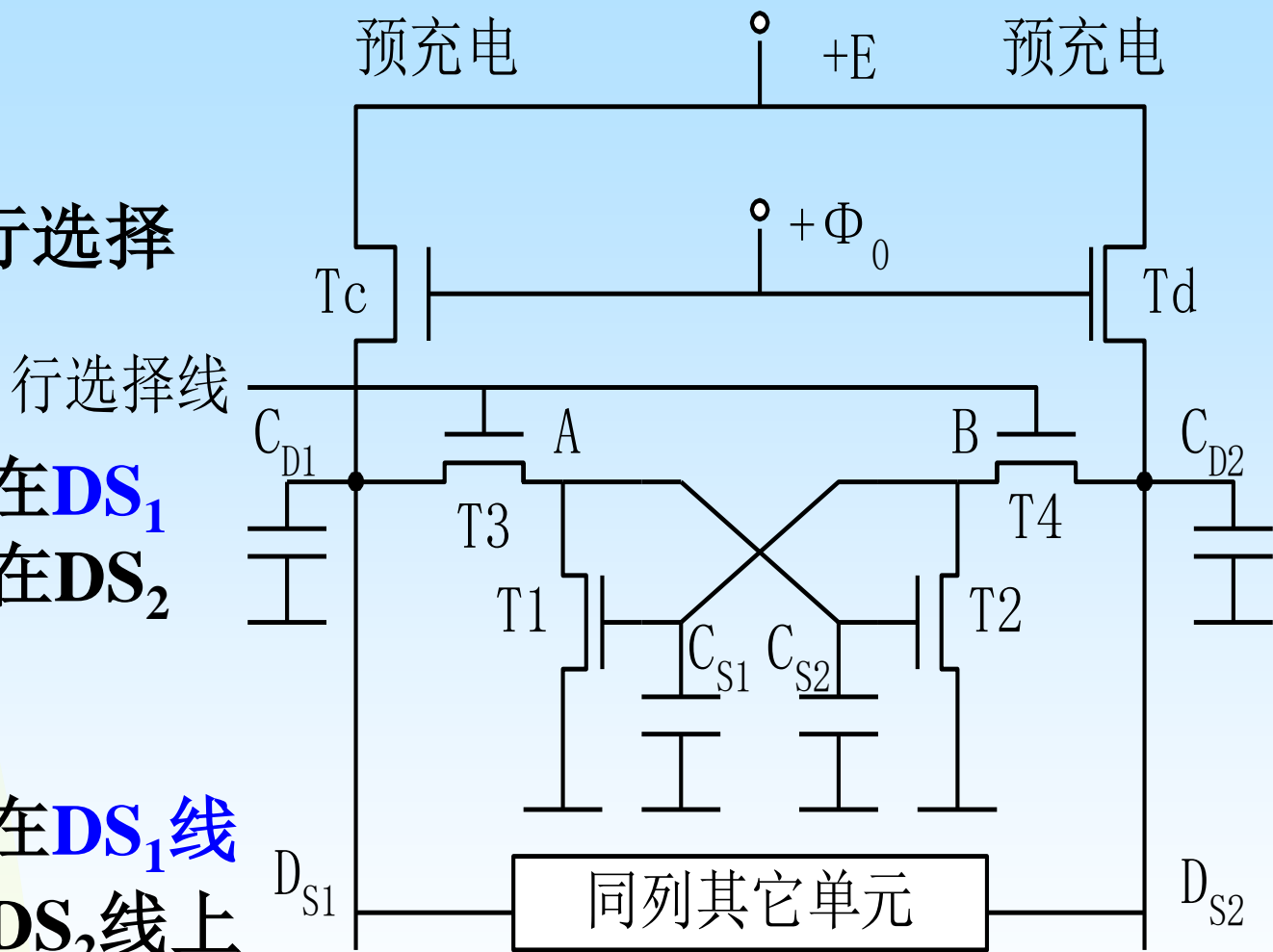


图5.17 四管动态MOS单元电路

(3) 单管动态RAM电路

- ① **读出**: 若原存“1”, 则 C_s 上电荷通过T管向数据线泄放, 形成读“1”信号。
- ② **写入**: 要写1, 在数据线上加高电位, 经T管对CS充电。
- ③ **刷新(Refresh)操作**: 定时给栅容补充充电, 这一过程称为“刷新”。

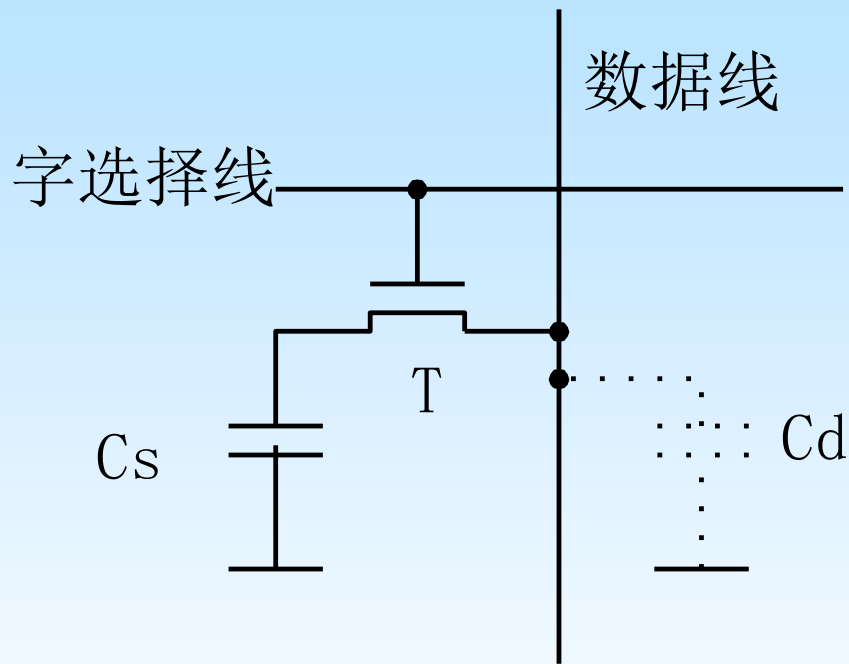


图5.18 单管动态MOS RAM 单元电路

2. MOS管动态存储器

(1) 动态RAM特点

- 容量较大，大多数产品都采用一位输入输出，如： $256\text{K} \times 1\text{b}$ 、 $1\text{M} \times 1\text{b}$ 、 $4\text{M} \times 1\text{b}$ 等。
- 它的行地址和列地址通过相同的管脚分先后两次输入，这样地址引脚数可减少一半。
- 当 $\overline{\text{RAS}}$ 低电平时输入行地址， $\overline{\text{CAS}}$ 低电平时输入列地址。

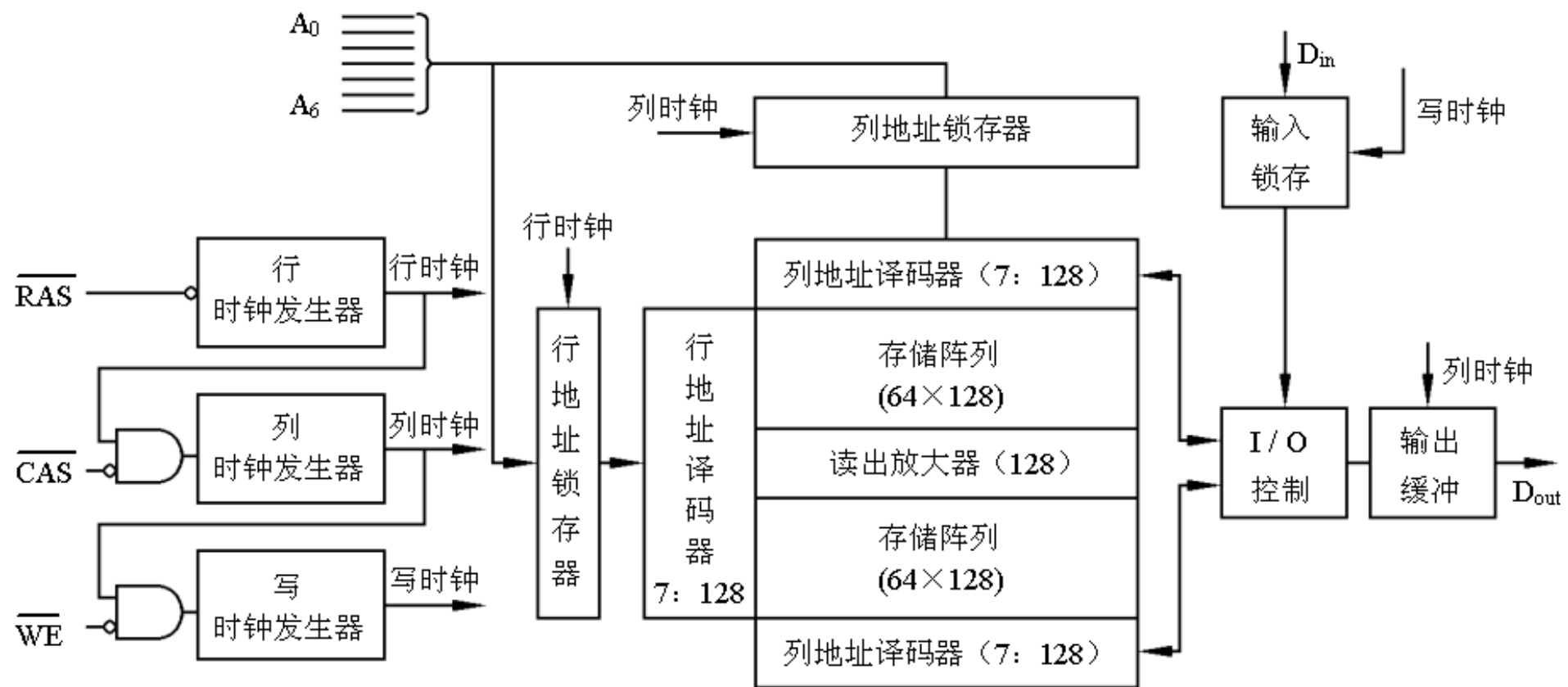


图5.19 16K×1位动态存储器框图

(2) 动态RAM的再生(刷新)

▲ 只送 $\overline{\text{RAS}}$ 信号的一种“刷新”方法

- 芯片中一行的所有元素被选中并进行“读出”操作。根据读出内容对各单元进行“重写”；完成补充充电。
- 由于没有列地址和CAS信号，各单元的数据读写彼此隔离，并且不会送到读出电路。
- 对 256×256 的存储体，256次刷新操作可刷新整个存储体。也可分为4个 128×128 并行连接，只需128次刷新。

(3) 刷新方式

▲ **刷新周期**：一次刷新的时间间隔，一般为2ms；

▲ 常用的刷新方式有四种：

- 集中式刷新、分散式刷新、异步刷新和透明刷新。

① 集中式刷新(图5.20(a))

- 整个刷新闻隔内，前一段时间用于正常的读/写操作。而在后一段时间逐行进行刷新。
- 若将 128×128 存储体刷新一遍，读写周期为 $0.5\mu\text{s}$ ，刷新间隔为2ms；前3872个周期用来进行正常的读/写操作，而后128个读写周期用来进行刷新操作。
- 该方式会出现读/写操作死区（128个周期）。

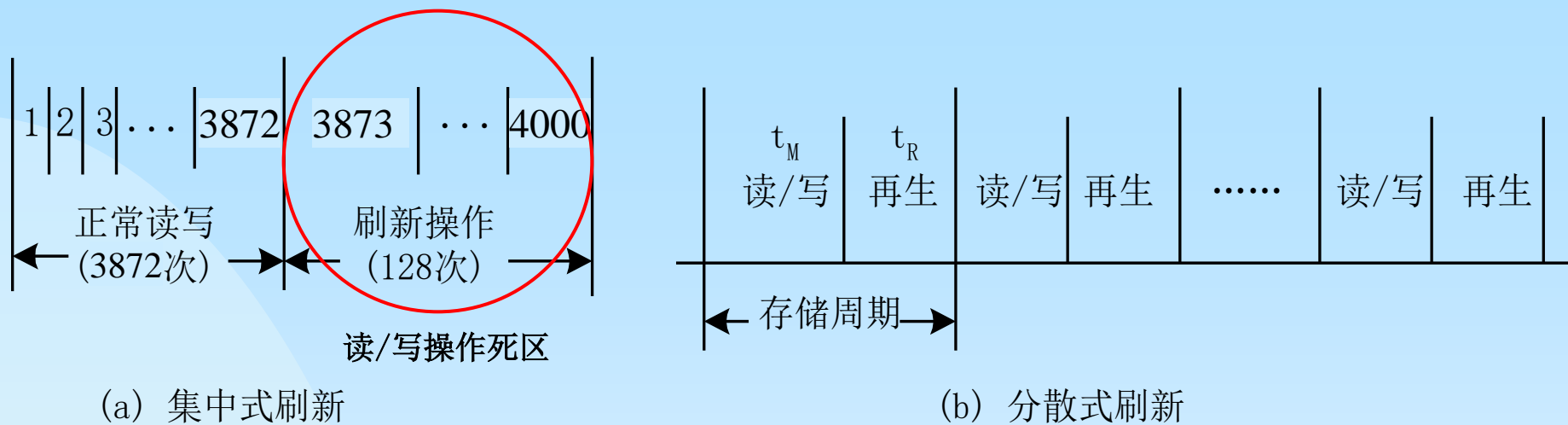


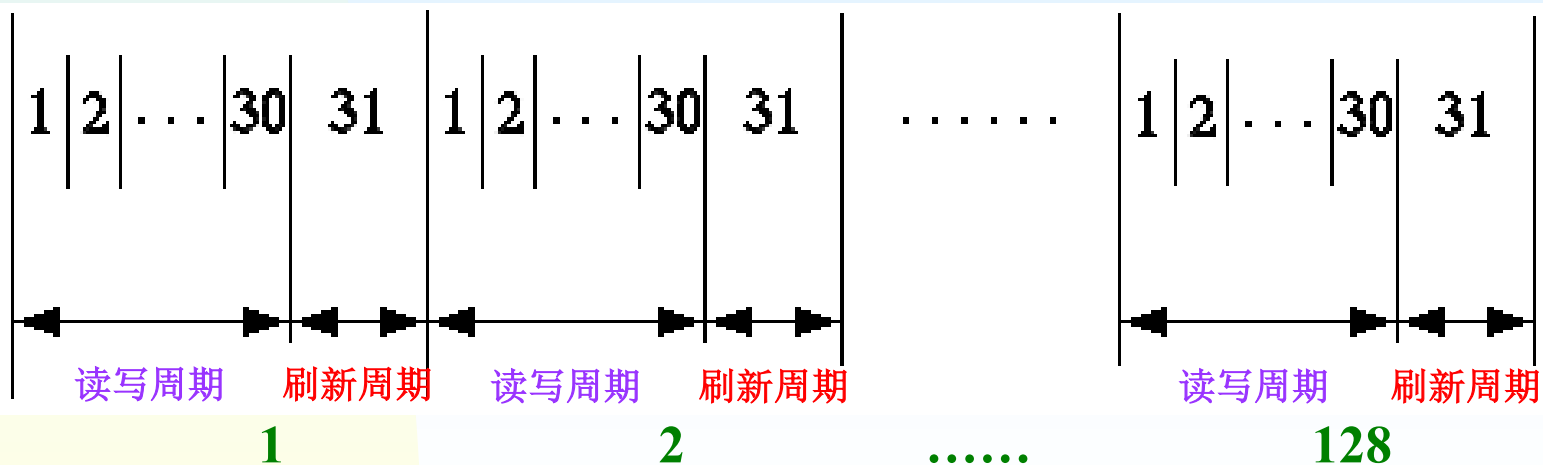
图5.20 两种刷新方式

② 分散式刷新(图5.20 (b))

- 一个存储周期的时间分为两段，前一段时间 t_M 用于正常的读/写操作，后一段时间 t_R 用于刷新操作。
- 假定读/写操作和刷新操作的时间都为 $0.5\mu s$ ，则一个存储周期为 $1\mu s$ 。在 $2ms$ 时间内进行2000次刷新操作，只能进行2000次读/写操作。

③ 异步刷新

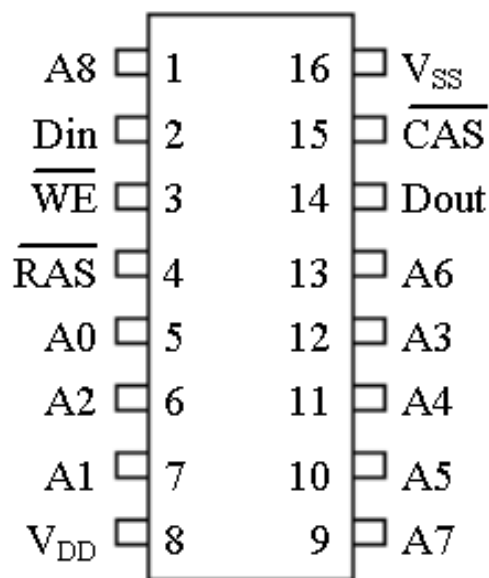
- 上述两种方式结合起来构成异步刷新。
- 以128行为例，在2ms时间内必须轮流对每一行刷新一次($2000/128=15.625$)，即每隔 $15.5\mu\text{s}$ 刷新一行。
- 前 $15\mu\text{s}$ 可以进行CPU的读/写操作，而最后 $0.5\mu\text{s}$ 完成刷新操作($30 \times 128 + 128 = 3968$)。目前用得较多。



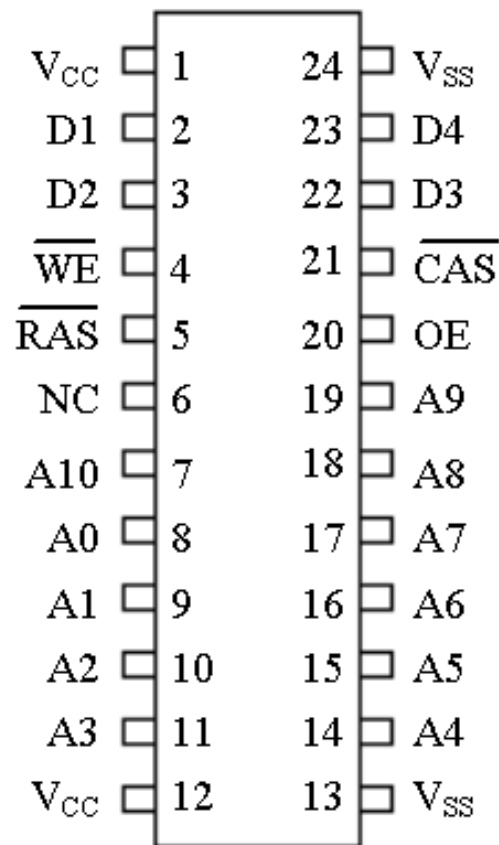
④ 透明刷新（或称稳含式刷新）

- CPU在取指周期后的译码时间内，插入刷新操作。
- 有单独的刷新控制器，刷新由单独的时钟、行计数与译码独立完成；高档微机中大部分采用这种方式。

3. 动态RAM芯片



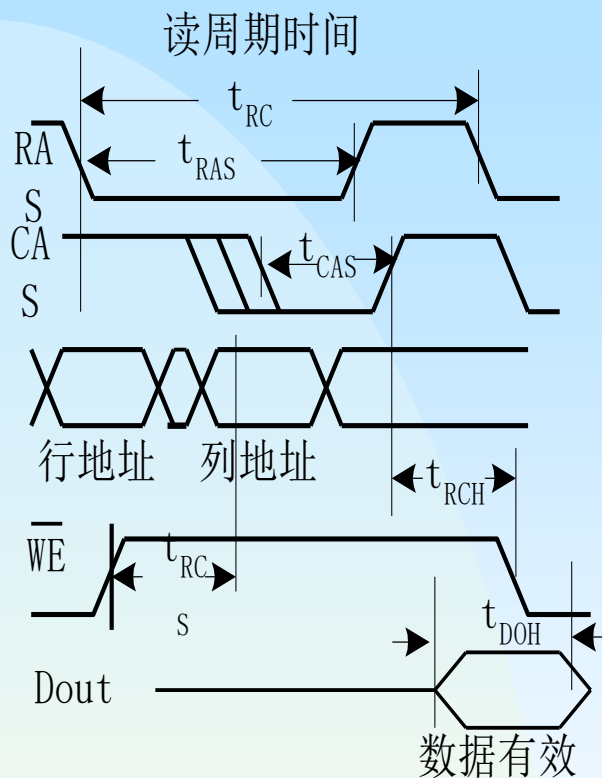
(a) 256Kx1 DRAM 芯片



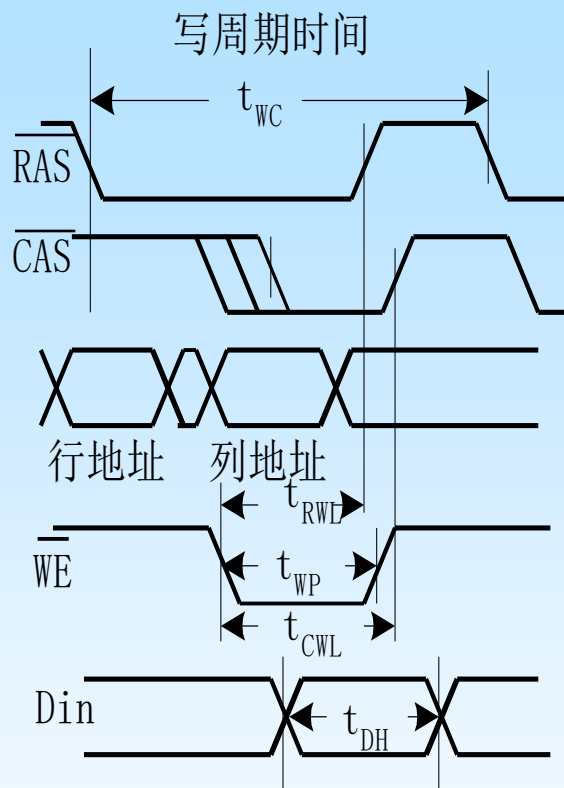
(b) 2Mx4 DRAM 芯片

图5.21 两种DRAM芯片

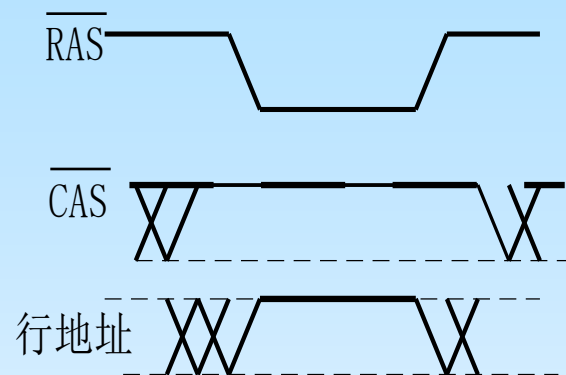
▲ 动态RAM芯片的读写和再生（刷新）时序



(a) 读周期



(b) 写周期



(c) 只用 \overline{RAS} 的刷新周期

图5.22 DRAM 时序

- 写数据必须在 \overline{CAS} 有效之前出现在Din端；
- “刷新”可以采用“读出”的方法进行。

4. DRAM控制器

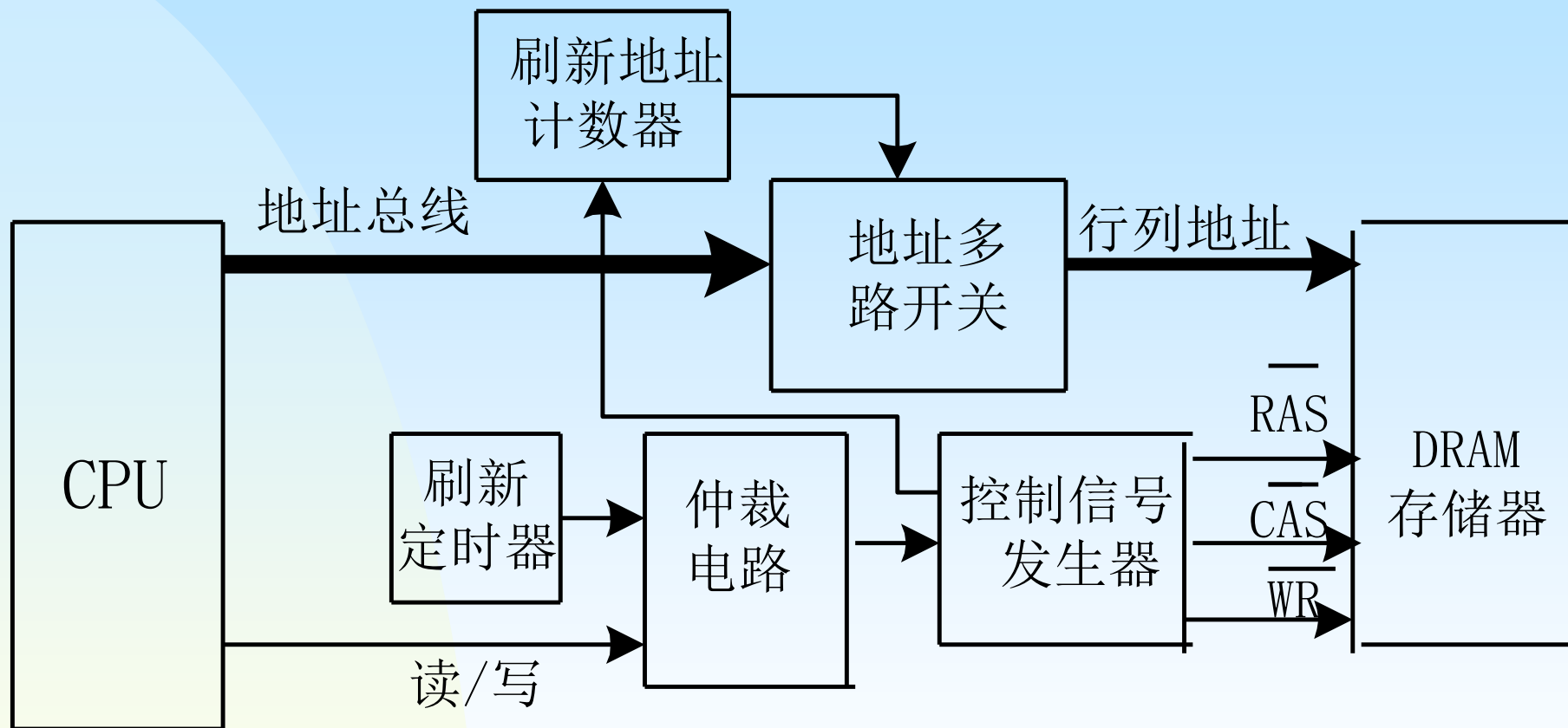


图5.23 DRAM控制器结构框图

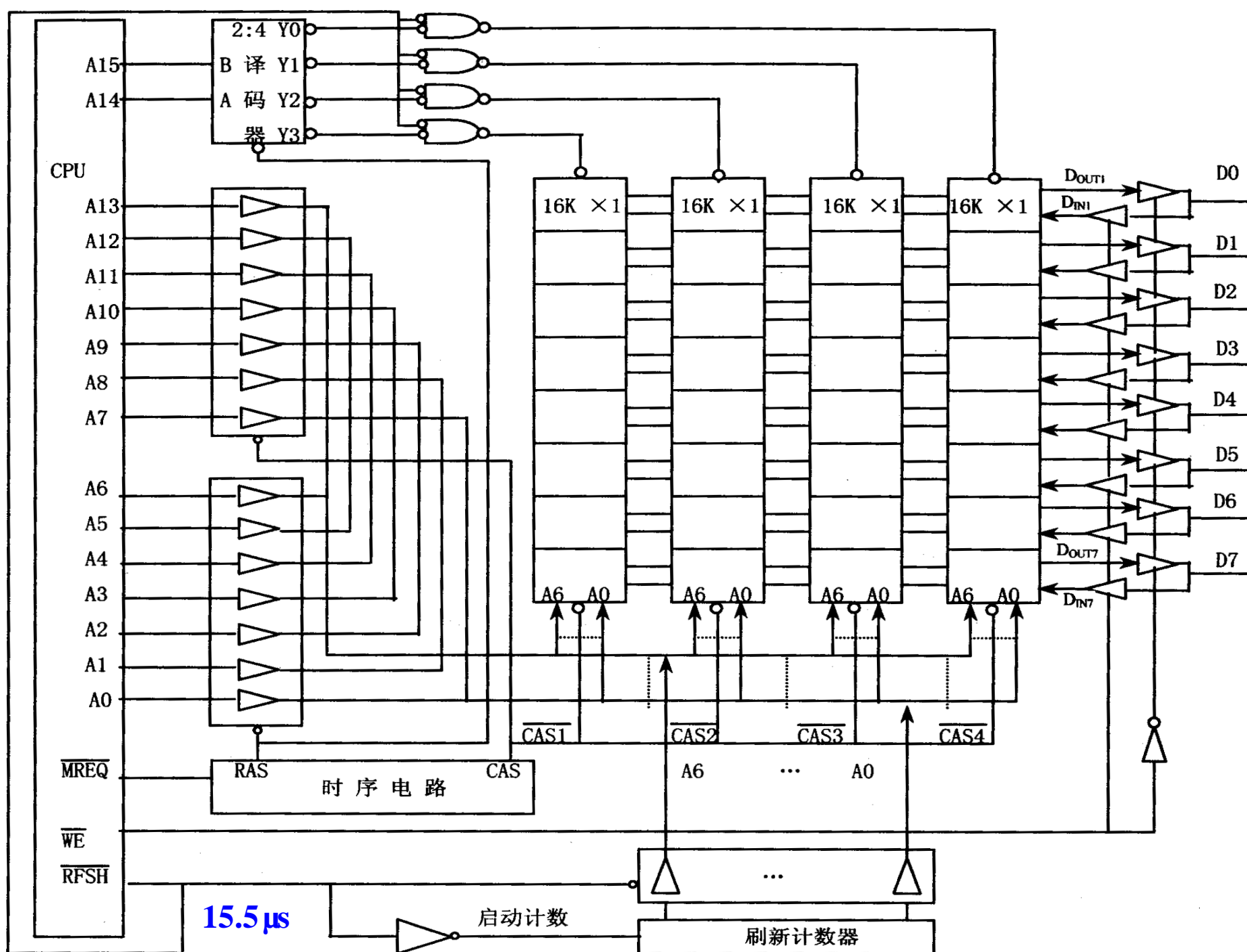
例. 用 $16\text{K} \times 1$ 位的DRAM芯片（由 128×128 矩阵存储元组成）构成 $64\text{K} \times 8$ 位的存储器。要求：

（1）画出该存储器的组成逻辑框图。

（2）设存储器的读/写周期均为 $0.5\mu\text{s}$ ，CPU在 $1\mu\text{s}$ 内至少访问一次。试问采用哪种刷新方式较合理？两次刷新的最大时间间隔是多少？对全部存储单元刷新一遍，所需实际刷新时间是多少？

解：

（1）总容量为 64KB ，故地址线共需16位；需要 $16\text{K} \times 1$ 位的DRAM芯片32片。芯片本身地址线占14位，所以要使用一片2:4译码器。



(2) CPU在 $1\mu\text{s}$ 内至少访问一次，所以采用异步刷新方式比较合理；

- 对DRAM来说，两次刷新的最大时间间隔是 2ms ；刷新时只对128行进行异步刷新。
- 刷新间隔为 $2\text{ms}/128=15.5\mu\text{s}$ ，前 $15\mu\text{s}$ 可以进行CPU的读/写操作，而后 $0.5\mu\text{s}$ 完成刷新操作。
- DRAM刷新一遍实际所需时间为 $128\times 0.5\mu\text{s}$ 。

5. 新型DRAM芯片

- **DRAM**具有存储密度大、功耗小、成本低等优点，由于它需要定时刷新，对其工作速度有较大影响。
- ◆ **EDRAM (Enhanced DRAM)**(图5.24)
 - **EDRAM** 在动态芯片上集成了一个容量小SRAM缓冲器，可以存放前一次读出的一整行元素内容，共2048位（512个4位组）。
 - **刷新可以和读写并行操作**，使芯片不能读写的时间减至最低程度。此外，从SRAM缓冲器读出数据的通路
与写入数据的通路各自独立。

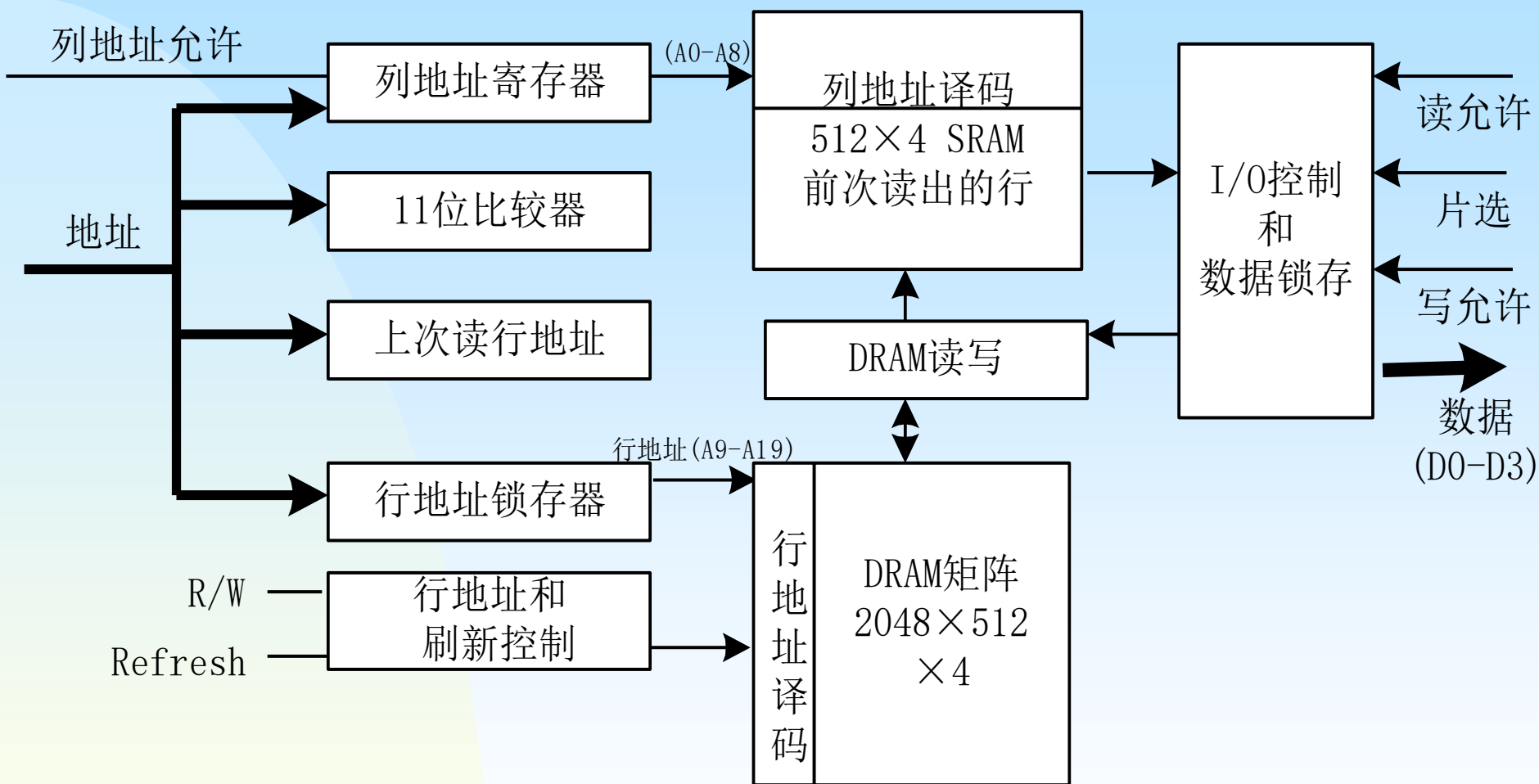


图5.24 EDRAM结构框图

◆ SDRAM (Synchronous DRAM)

- 锁存处理机或其它主设备发出的地址和控制信息；
- SDRAM内部采用双存储体结构，极大地改善了片内存取的并行性。

◆ DDR SDRAM (Double-Data-Rate SDRAM)

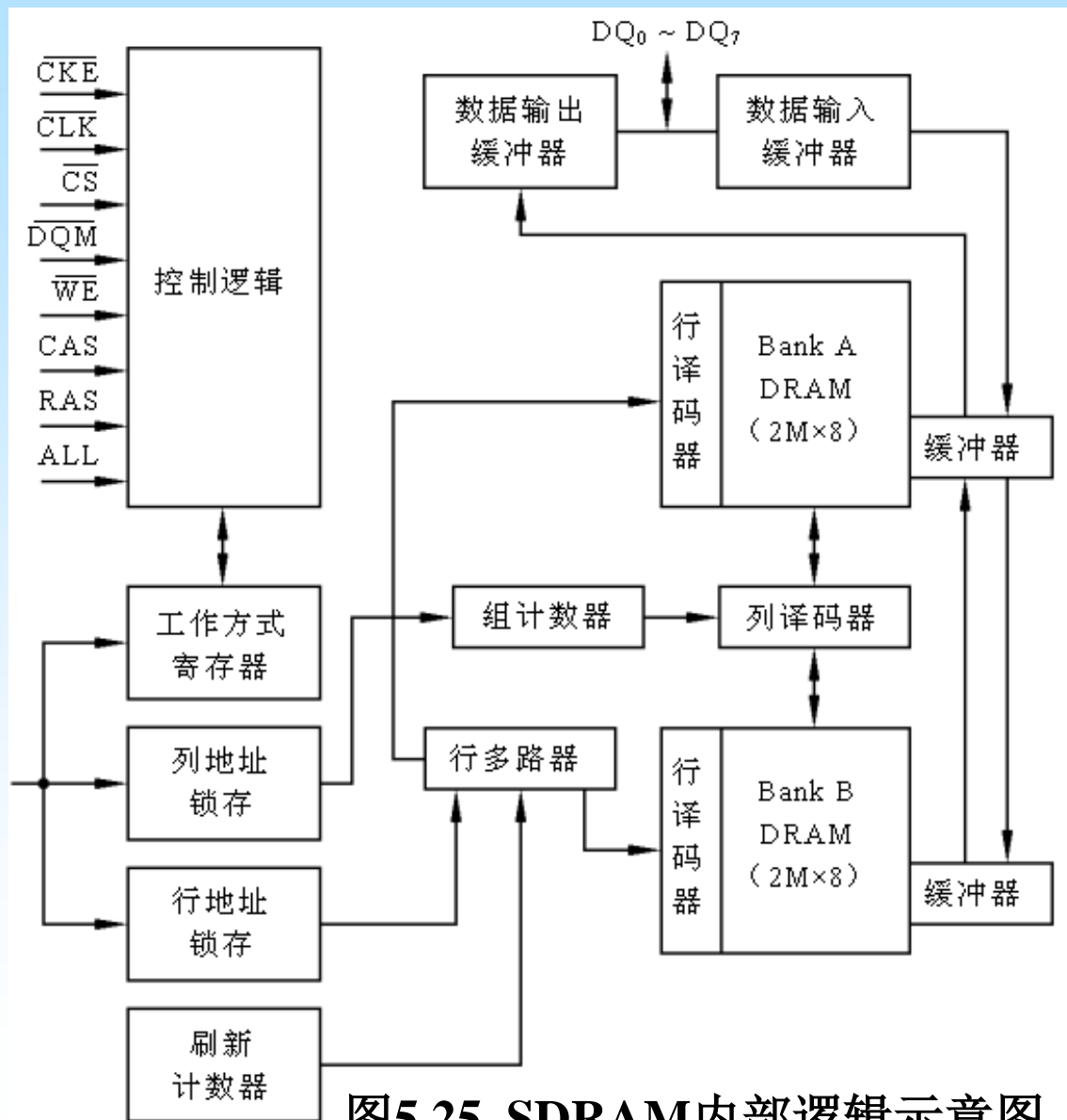


图5.25 SDRAM内部逻辑示意图

5.3 半导体只读存储器

- 根据只读存储器的工艺，可分为：

ROM

PROM

EPROM

EEPROM (E²PROM)

5.3.1 掩膜只读存储器 (masked ROM)

- 掩膜ROM存储的信息由生产厂家在掩膜工艺过程中“写入”，用户不能修改。

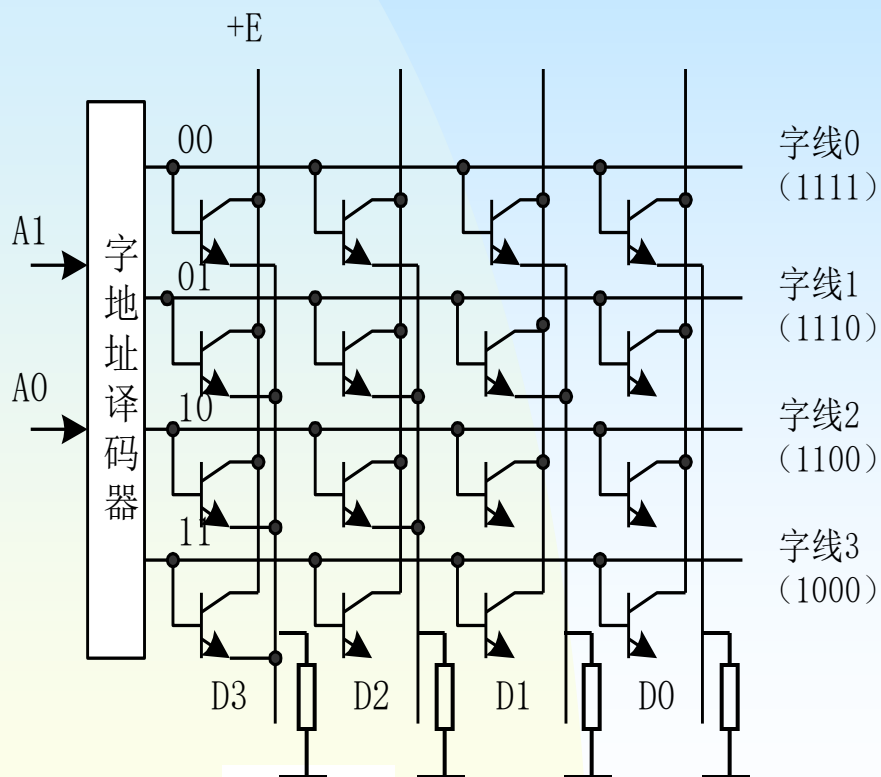


图5.26 4×4 双极型掩膜ROM

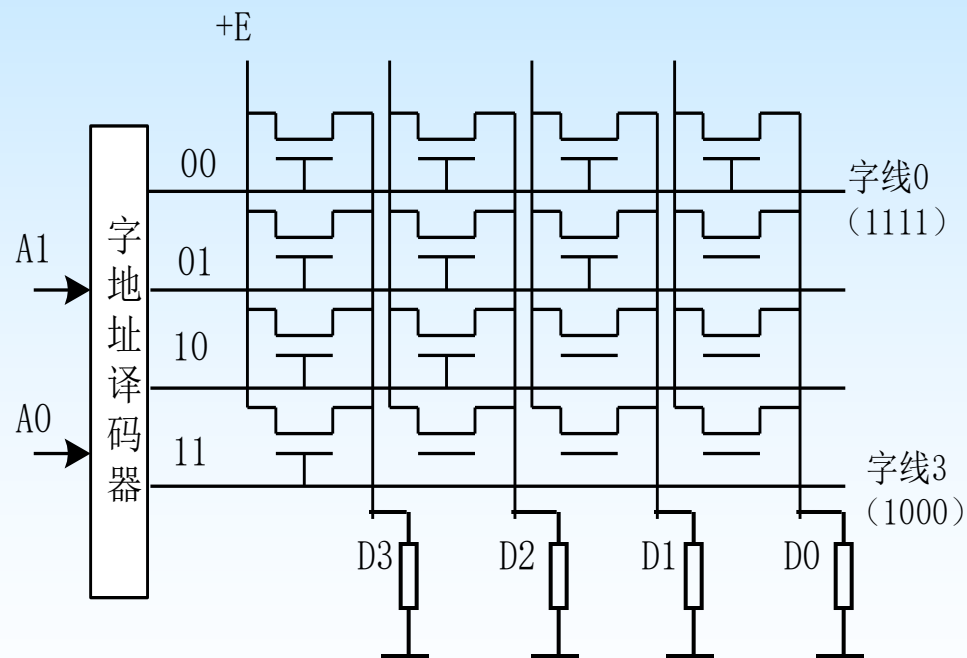


图5.27 4×4 MOS型掩膜ROM

5.3.2 可编程ROM (PROM)

- ◆ 这类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；
- ◆ 有P-N结破坏型和熔丝烧断型两种。

1. P—N结破坏型PROM(图5.28)

- 写入时，字线加电压E。若写“1”，则位线D加负压，将反向偏置的二极管击穿；若写“0”，位线上不加负压，P—N结不烧穿。

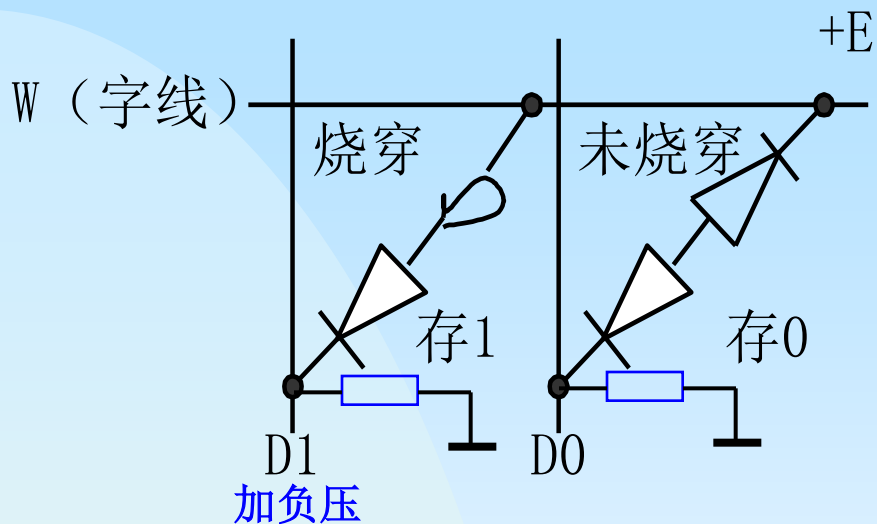


图5.28 P-N结破坏型PROM

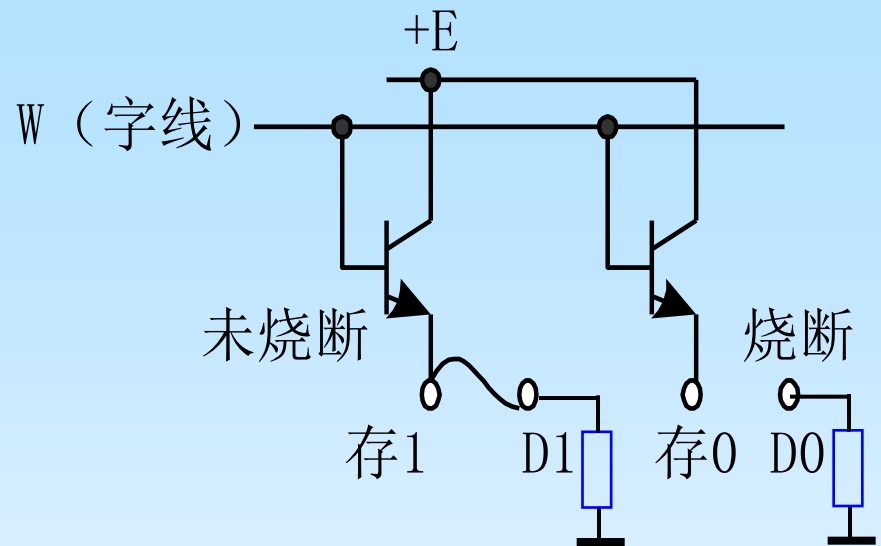


图5.29 熔丝烧断型PROM

2. 熔丝烧断型(图5.29)

- 写入时，在字线和位线上加高电压，使熔丝烧断，烧断存0，不烧断存1。

5.3.3 可擦除和编程的ROM (EPROM)

- ▲ 漏极上加正高压(20~30V)，浮动栅上带有足够多的正电荷，使FAMOS管处于导通状态；写入完毕后，撤消D极上的高压，保持导通状态。

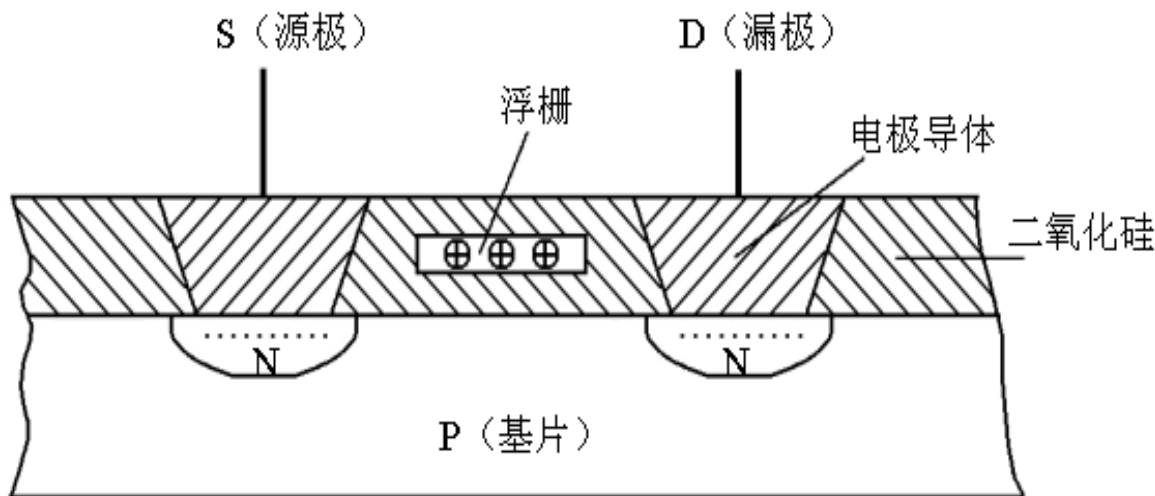


图5.30 N沟道FAMOS存储单元结构

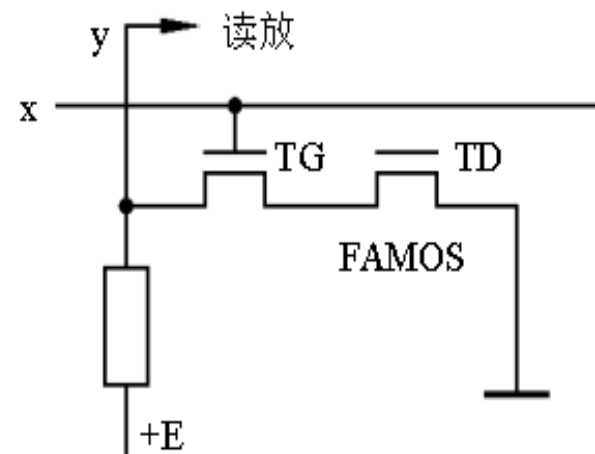


图5.31 FAMOS存储单元电路

◆ EPROM芯片27256
(32K×8)

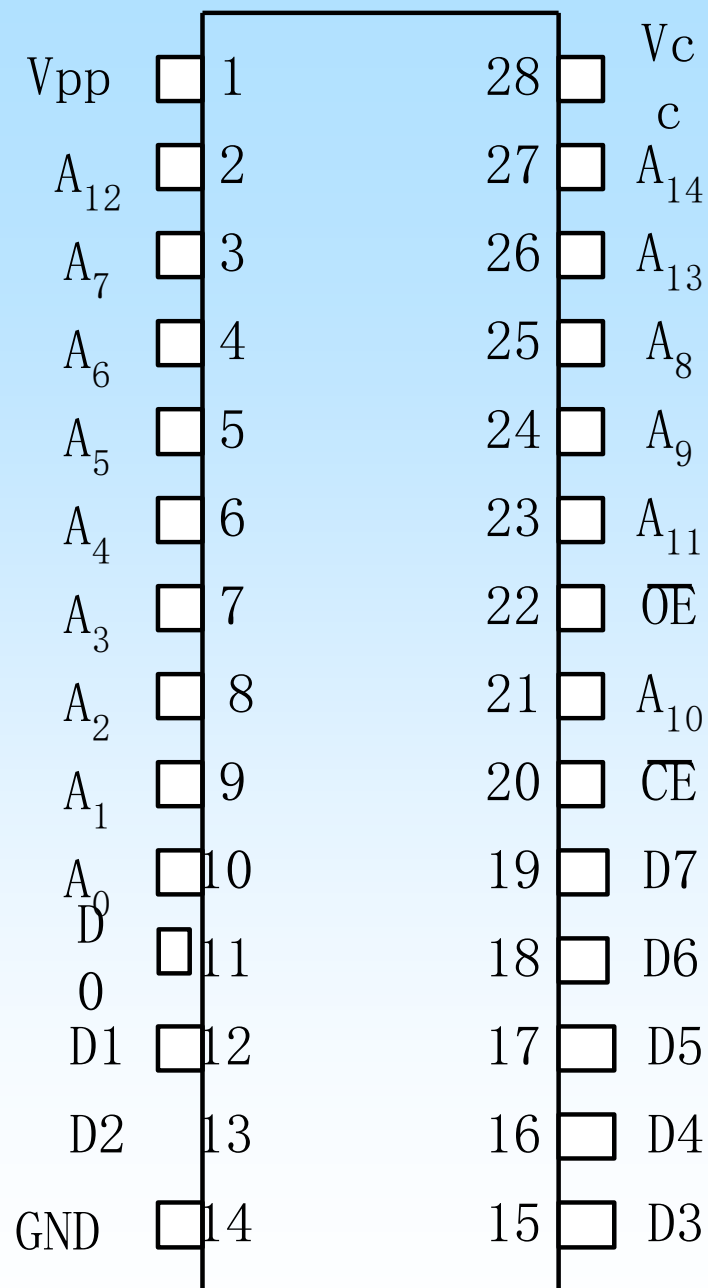


图5.32 27256 引脚图

5.3.4 电擦除电改写只读存储器（EEPROM）

- ◆ 电擦除电改写只读存储器又叫EEPROM或E²PROM（Electrically Erasable and Programmable Rom）。
- 在读数据的方式上与EPROM完全一样，优点是可以用电来擦除和重编程。
- E²PROM在每次写入操作时执行一个自动擦除，因此比RAM的写操作慢的多。 E²PROM存放的数据至少可维持10年。

5.3.5 闪速存储器（flash memory）

- ◆ 闪速存储器（闪存）是在EPROM和E²PROM的制造技术基础上发展起来的一种新型的电可擦除非易失性存储器件。
- 存储单元结构与E²PROM的类似，闪速存储单元的氧化层较薄，具有更好的电可擦除性能。
- 新型闪速存储器则可擦除一块数据，因而更适于存储文件方面的应用。
- 闪存速度快、功耗低、体积小，可用于代替ROM、磁盘；进行数据采集。

5.4 多体交叉存储器

5.4.1 编址方式

- ◆ 多模块存储器可以实现重叠与交叉存取。
- 设存储器包括M个存储模块, 每个存储模块的容量为L;
- 第i存储模块的地址编号为 $M \times j + i$
其中: $i=0, 1, 2, \dots, M-1$; $j=0, 1, 2, \dots, L-1$ 。

表5.1 模4交叉编址

存储模块	地址编址系列	对应二进制地址最低两位
M_0	0, 4, 8, 12, ..., $4j+0$, ...	0 0
M_1	1, 5, 9, 13, ..., $4j+1$, ...	0 1
M_2	2, 6, 10, 14, ..., $4j+2$, ...	1 0
M_3	3, 7, 11, 15, ..., $4j+3$, ...	1 1

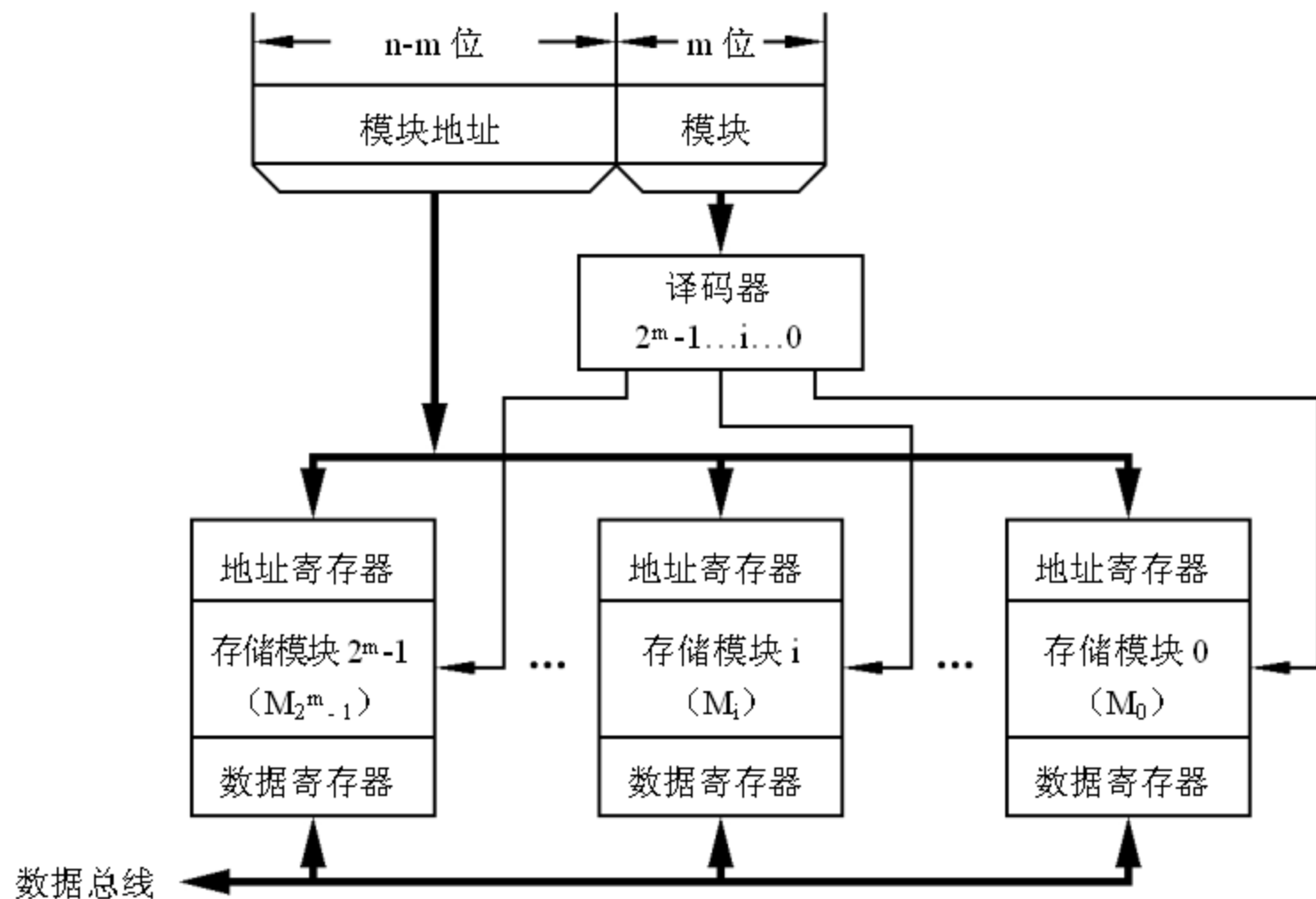


图5.33 多体交叉编址方式

5.4.2 重叠与交叉存取控制

- ◆ 多体交叉存储模块有两种不同的访问方式：“同时访问”和“交叉访问”

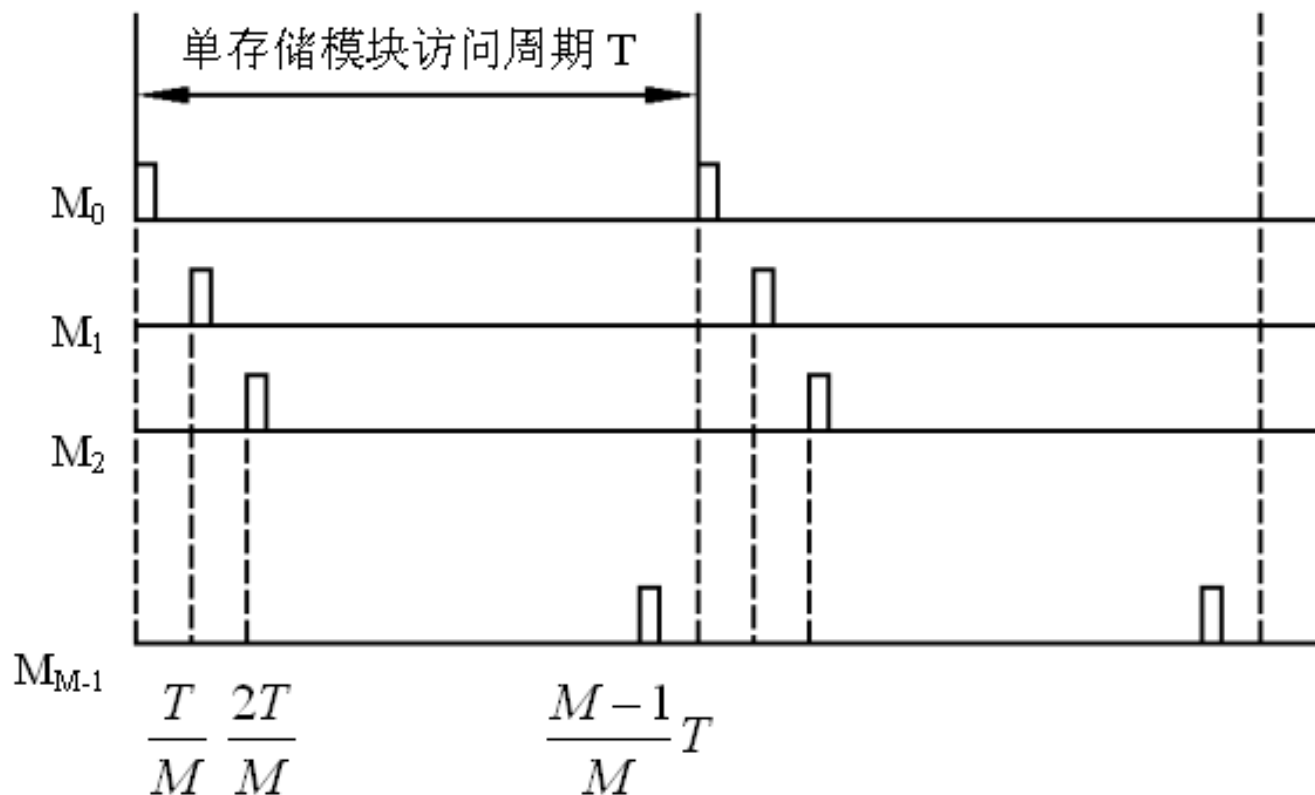


图5.34 交叉访问的存储器工作时间图

5.5 高速缓冲存储器

5.5.1 工作原理

◆ Cache的功能

- 设置Cache是为了解决CPU和主存之间的速度匹配问题，理论依据是程序访存的局部性规律。
- 高速缓存通常由双极型半导体存储器或SRAM组成。
- 地址映象以及和主存数据交换机构全由硬件实现，并对程序员透明。
- 访问Cache的时间一般为访问主存时间的 $1/5 \sim 1/10$ ；Cache已在大、中、小及微型机上普通采用。

◆ Cache的基本结构和工作原理

- 主存和Cache均是模块化的（以块为单位），并且两者之间交换数据以块为单位进行。

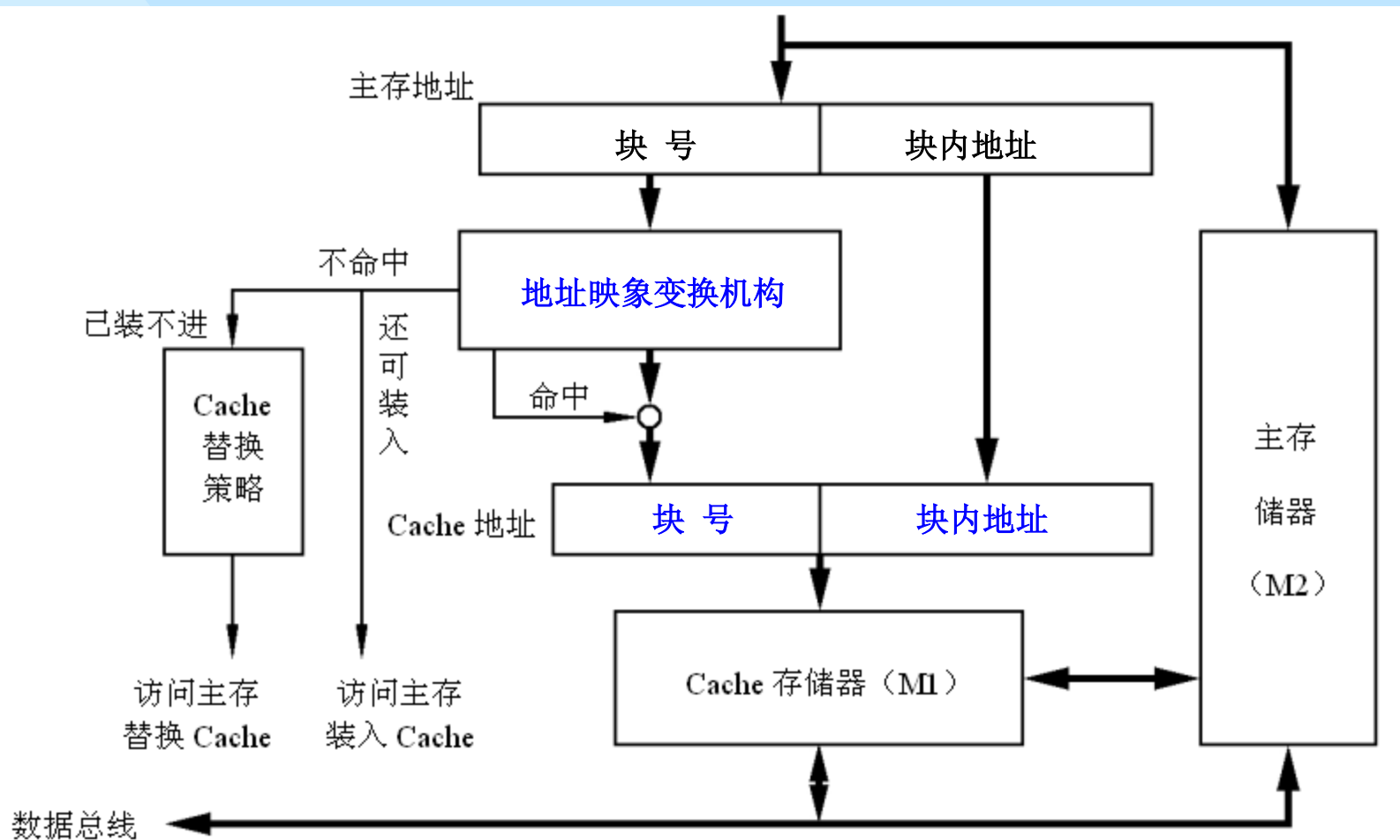


图5.35 Cache的结构原理

◆ Cache的命中率

- 在程序执行期间，设 N_c 表示Cache完成存取的总次数， N_m 表示主存完成存取的总次数；

- Cache的命中率： $P=N_c / (N_c+N_m)$ ；

- Cache/主存系统平均访问时间：

$$T_a = PT_c + (1-P)T_m$$

- Cache/主存系统访问效率：

$$\begin{aligned} e &= T_c / T_a = T_c / [PT_c + (1-P)T_m] \\ &= 1 / [P + (1-P)r] = 1 / [r + (1-r)P] \end{aligned}$$

- $r=T_m/T_c$ 表示主存慢于Cache的倍率，取5~10。

- 命中率P与程序的行为、Cache的容量、组织方式、块的大小有关。
- 例：CPU执行一段程序时，Cache完成存取的次数为1900次，主存完成存取的次数为100次，已知Cache存取周期为50ns，主存存取周期为250ns，求Cache/主存系统的效率和平均访问时间。

解：Cache的命中率： $P = N_c / (N_c + N_m)$
 $= 1900 / (1900 + 100) = 0.95$

$$r = T_m / T_c = 250\text{ns} / 50\text{ns} = 5$$

系统访问效率： $e = 1 / [r + (1 - r)P]$
 $= 1 / [5 + (1 - 5)0.95] = 83.3\%$

系统平均访问时间： $T_a = T_c / e = 50\text{ns} / 0.833 = 60\text{ns}$

5.5.2 映像函数 (Mapping)

◆ 映像函数

- 为了便于根据CPU送来的地址信息到Cache中去读取数据，必须有某种函数把主存地址映像成Cache地址。实现这种映像的函数叫映像函数。
- ◆ Cache空间的分配以及数据交换都以块为单位进行。
 - 假定主存空间被分为 2^m 个块（块号为0、1、2..... 2^m-1 ），每块大小为 2^b 个字。
 - Cache存储空间分为 2^c 块，每块也是 2^b 个字（当然 $m>c$ ）。

◆ 常用的地址映象方式

- 有直接映象、全相联映象和组相联映象。
- 选取何种映象方法，取决于在给定地址映象和变换的硬件条件下，能否达到**高速度**，以及能否使**块冲突的概率小**。
- **块冲突**：要调一个主存块到Cache中，而该主存块要进入的Cache块已被其它主存块占据。

1. 直接映象法

▲ 主存和Cache块号的对应关系

- 直接映象函数为 $i = j \bmod 2^c$ ，其中 i 是Cache块号， j 是主存块号， 2^c 为Cache总块数。
- 将主存分为若干个区，每区与Cache容量相等；每区相同块号的块映象到Cache对应块。

主存块号

$$i = j \bmod 2^c$$

1个Cache块对映多个主存块，t为这些主存块标记。

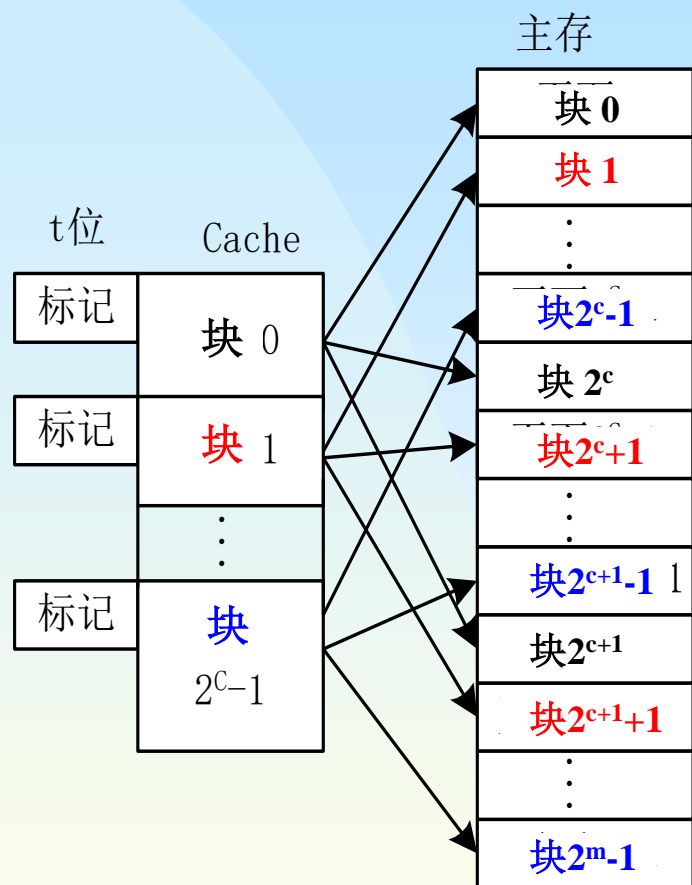


图5.36 直接映像方式

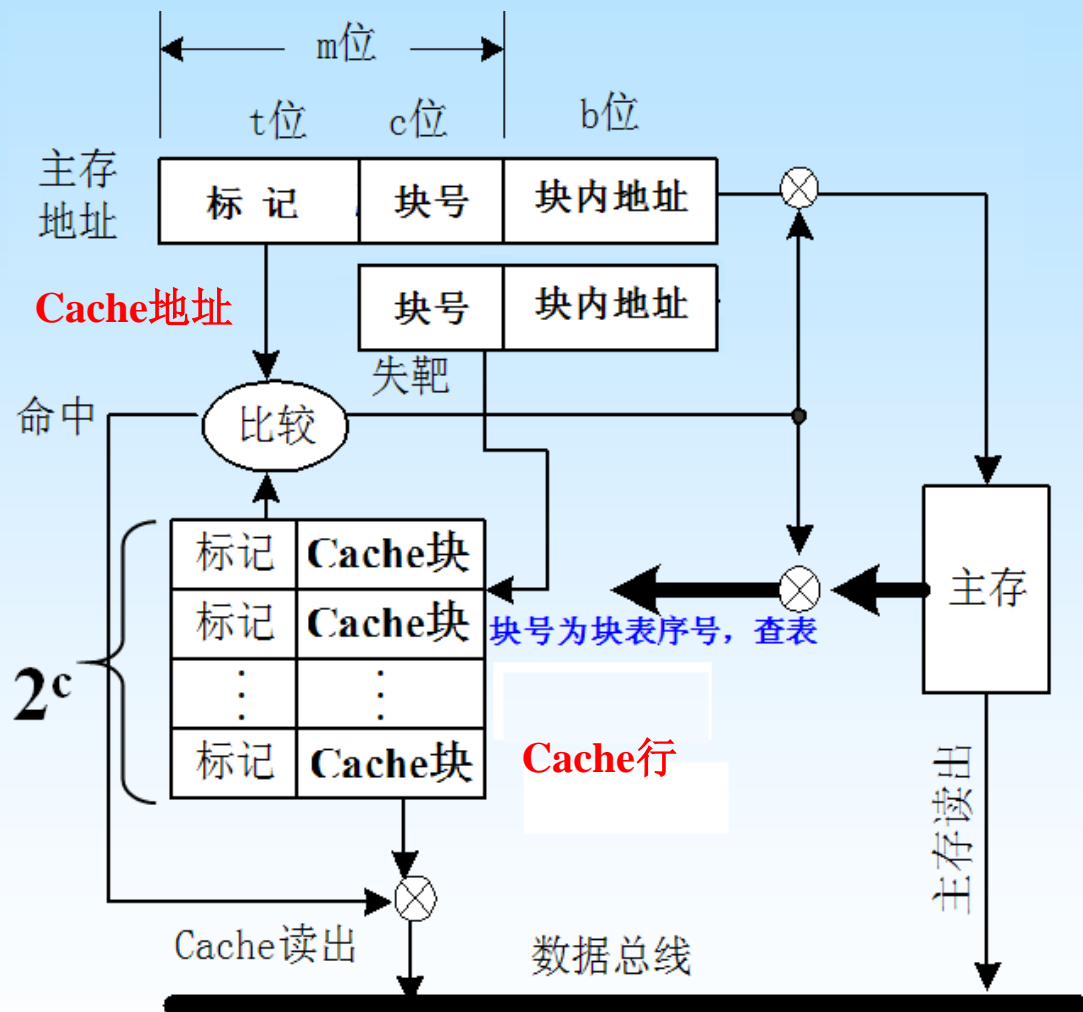


图5.37 直接映像方式下，主存和Cache读出过程

2. 全相联映象法

▲ 对应关系

- 主存中任一页可装入**Cache**内任一块的位置。
- 采用存放于**相联存储器中的目录表**来实现地址映象；以加快“主存—**Cache**”地址变换速度。
- 优点是**块冲突概率最低**；但查表速度难以提高。几乎没有单纯采用全相联映象法。

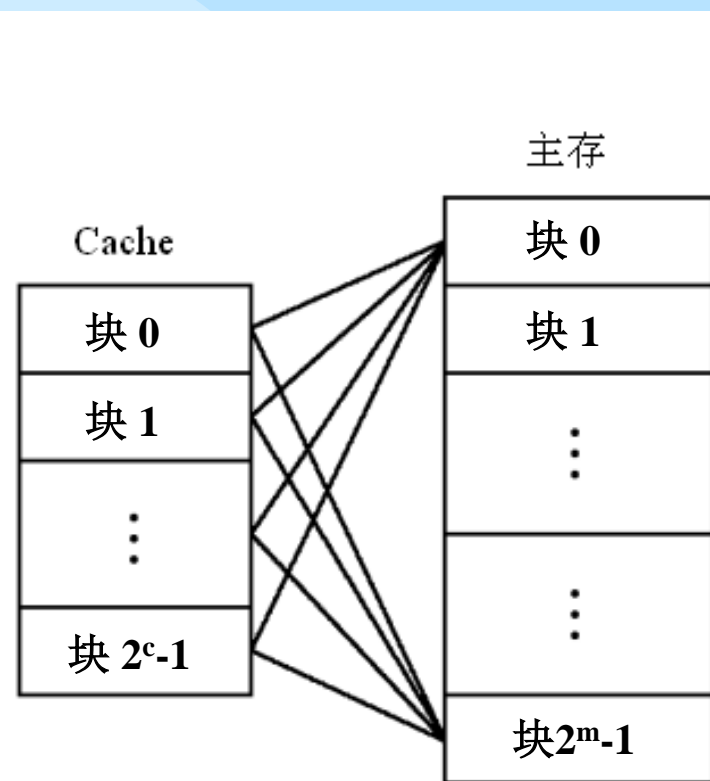


图5.38 全相联映象方式

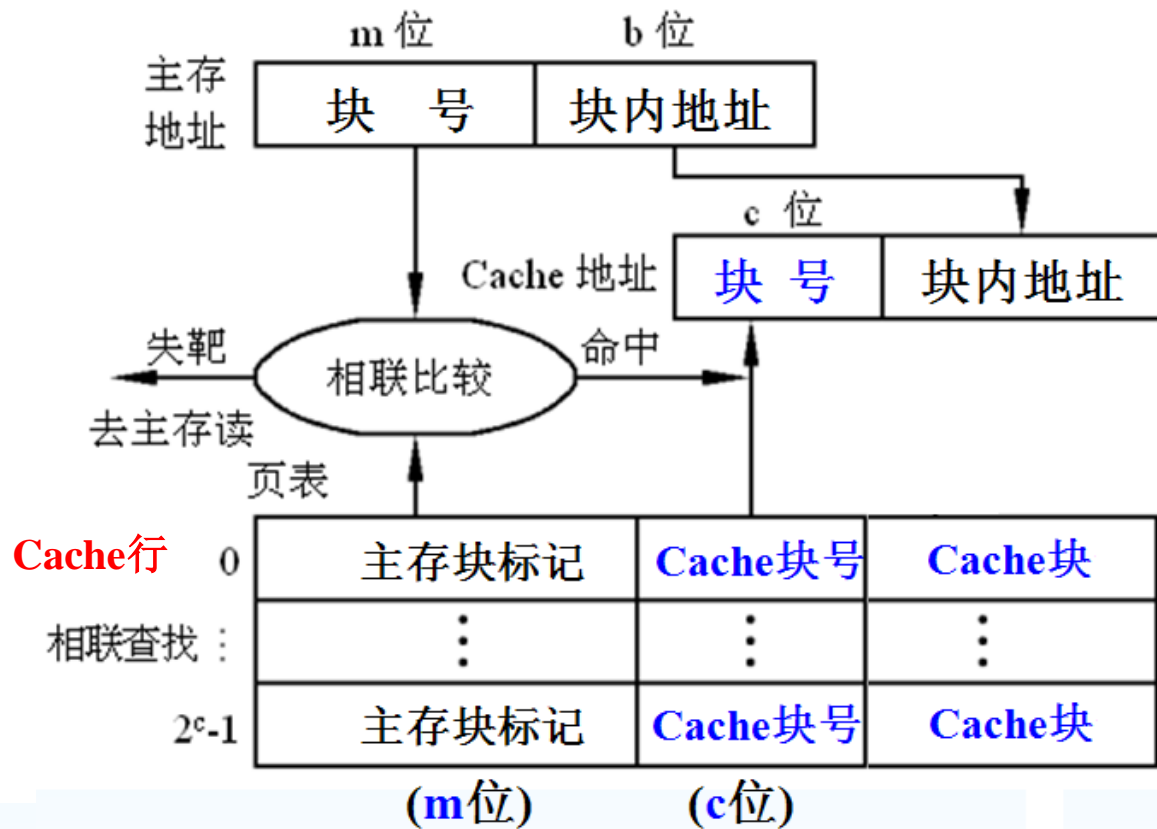
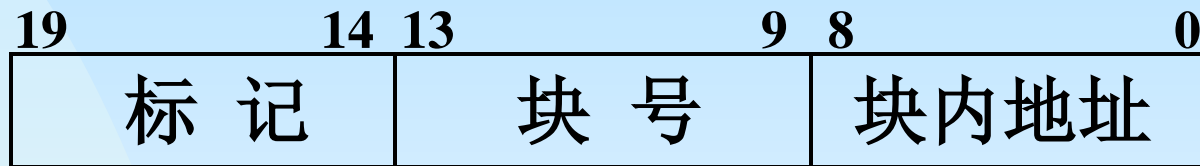


图5.39 全相联映象地址变换

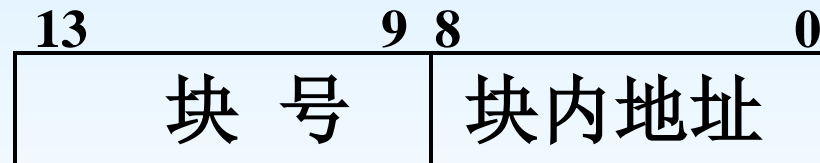
例1. 设主存容量为1MB，高速缓存容量为16KB，块的大小为512字节。采用直接地址映像法。

- (1) 写出主存地址格式。
- (2) 写出Cache地址格式。
- (3) 块表的容量是多大。
- (4) 画出直接方式地址映像及变换示意图。

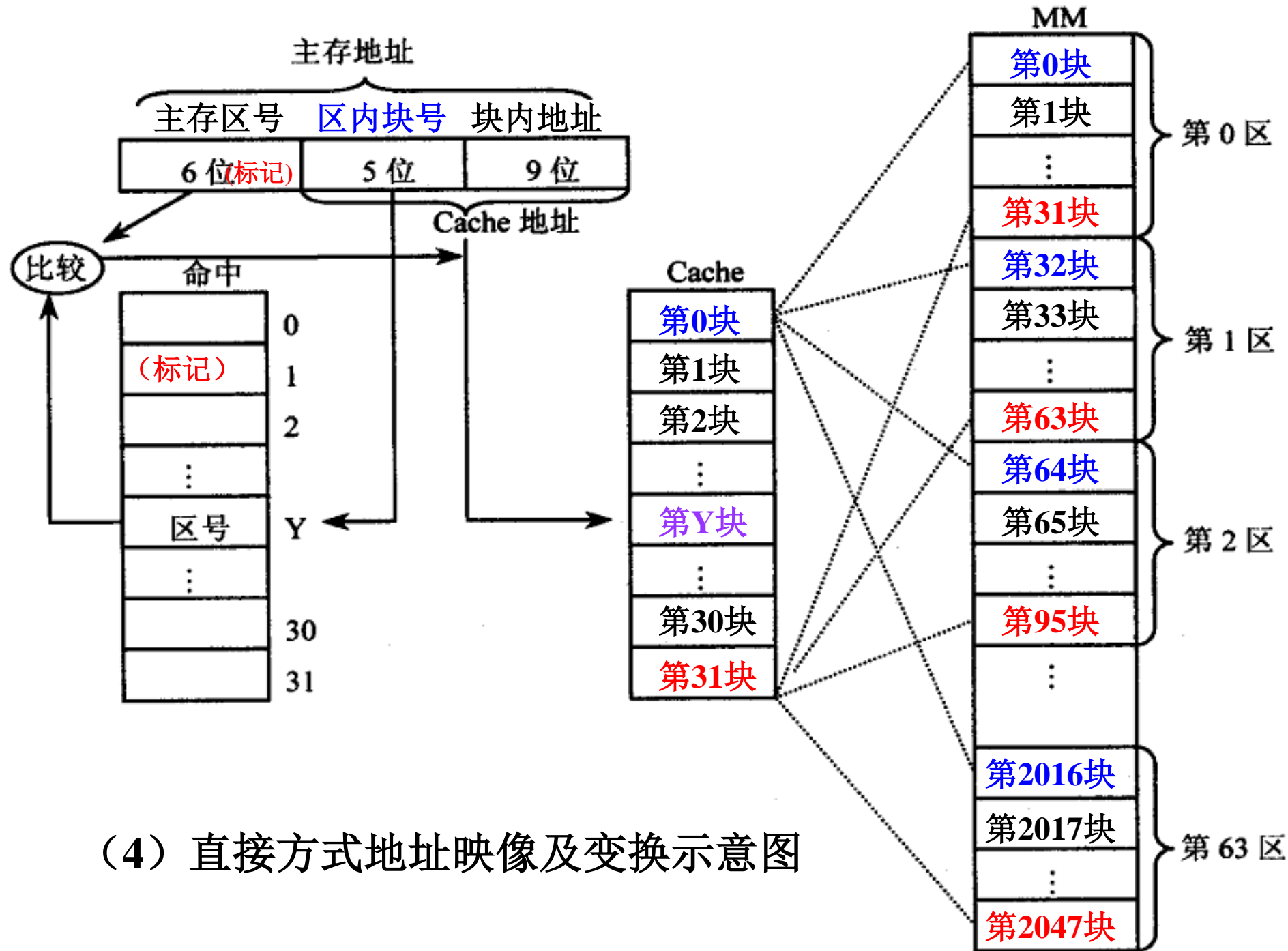
(1) 主存地址格式为:



(2) Cache地址格式为:



(3) 块表的容量为 32×6 位。



(4) 直接方式地址映像及变换示意图

例2. 一个具有16KB直接相联映射Cache的32位微处理器，假定Cache的块为4个32位的字，问主存字节地址为ABCDE8F8H的单元在Cache中的位置是什么？

解：

**Cache容量为 2^{14} 字节， 2^{12} （4096）个字，
对应Cache块地址（块号）：28FH；**

块内字节地址：8H；

3. 组相联映象法

- ▲ 全相联映象法和直接映象法结合起来，就产生了组相联映象法。
- ▲ 组相联映象法的映象规则
 - 将Cache空间分成组，每组 2^s 块（称为 2^s 路相联），Cache有 2^q 组。
 - 主存分成 2^m 个区，每区共有 2^q 块。主存某区的块允许映射到Cache固定组内的任意块。

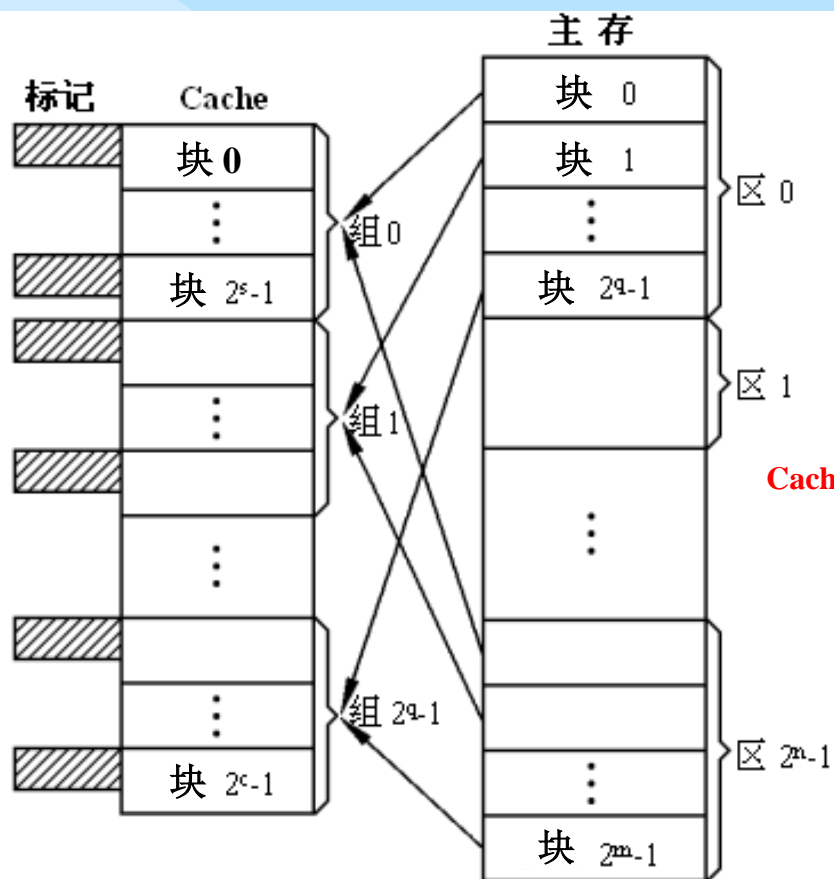


图5.40 组相联映像方式

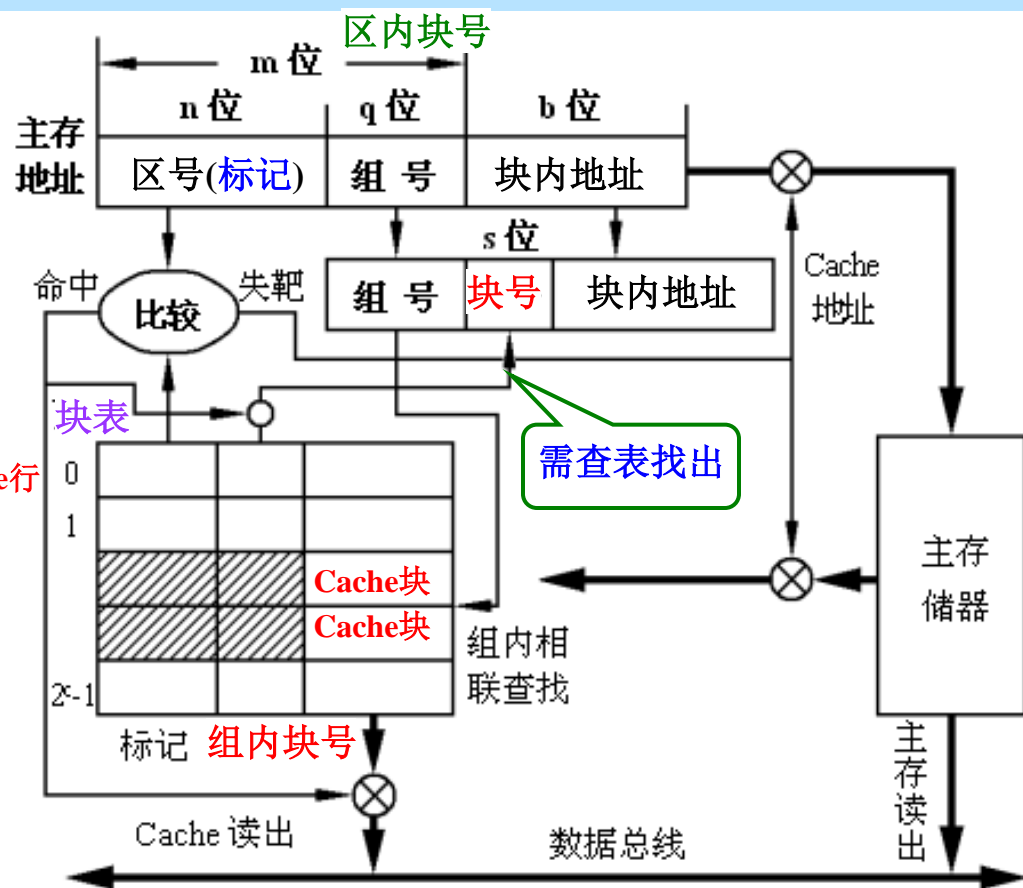


图5.41 组相联映像的地址变换方法

- ▲ 在组相联映象中，主存某区的第*i*块可以调入Cache第*i*组中的任意一块中；
- 组相联映象在各组间用直接映象，组内各块则是全相联映象；
- S （Cache组的大小为 2^s ）的选取决定了块冲突的概率和地址映象的复杂性。 S 字段越大，则Cache块冲突越低，而相联映象表也越大。

例. 一个**组相联映像Cache**由64个存储块组成， 每组包含4个存储块。主存包含4096个存储块， 每块由128字组成。访存地址为字地址。

- (1) 写出主存地址位数和地址格式;
- (2) 写出Cache地址位数和地址格式;
- (3) 画出组相联映像方式示意图,
- (4) 主存字地址为7B568H单元映射到Cache哪个组?

(1) Cache的组数=64个存储块/每组4个存储块=16组,
Cache的容量=64×128字=2¹³字

Cache地址格式为:

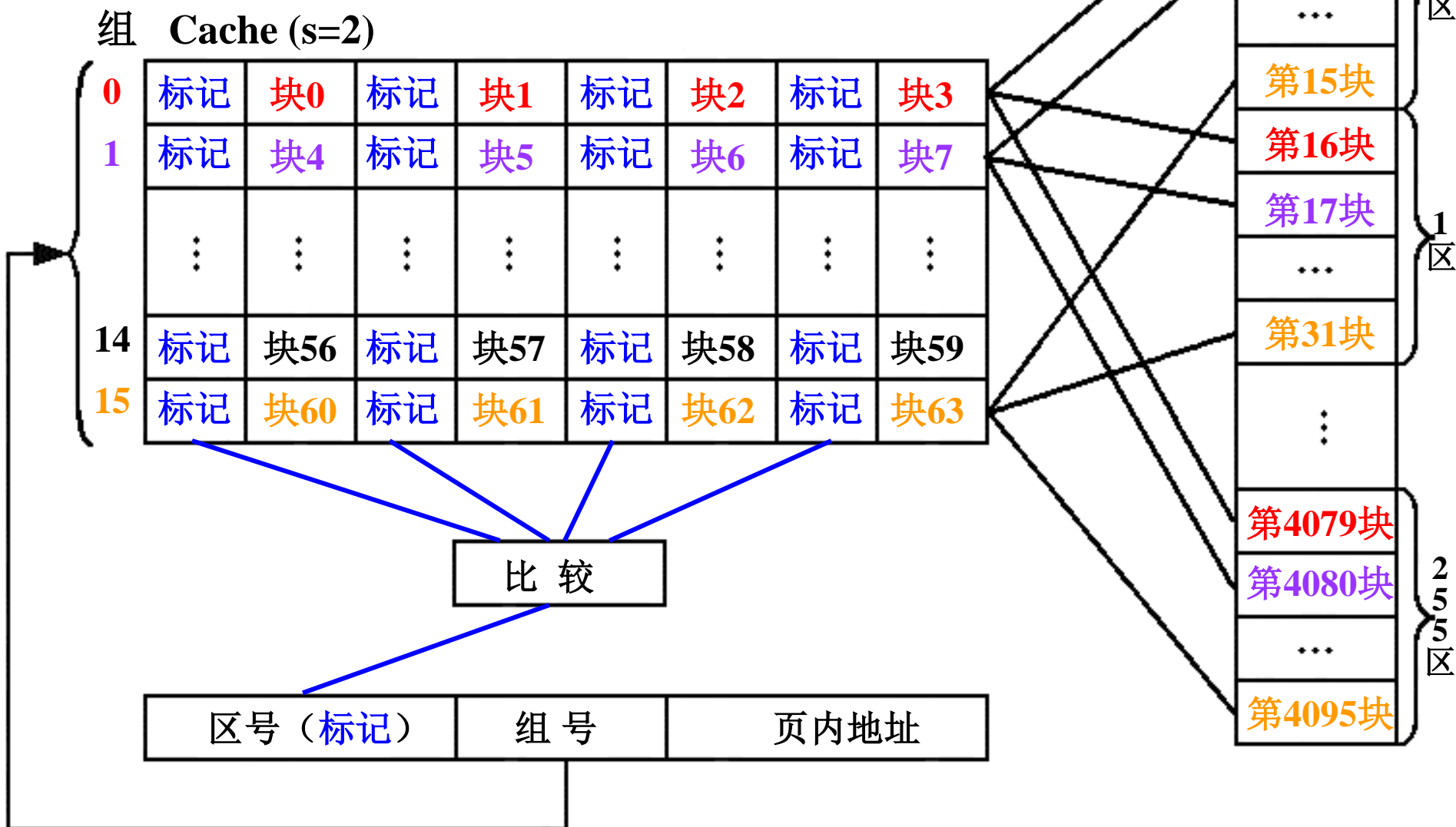
4	2	7
组 号	组内块号	块内地址

(2) 主存每区为16块, 主存容量=4096×128字=2¹⁹字

主存地址格式为:

8	4	7
区 号 (标记)	区内块号(组号)	块内地址

(3) 组相联映像方式示意图



(4) 主存字地址为7B568H单元映射到Cache哪个组? (1010)

5.5.3 替换算法

◆ Cache块失靶处理

- 访存出现Cache块失靶，需将主存块按所采用的映象规则装入Cache。
- 如果此时出现块冲突，就必须按某种策略将Cache块替换出来。

◆ 替换策略的选取

- 从实现的难易和命中率的高低来考察。

1. 先进先出法（FIFO）

- 选择最早装入的Cache块为被替换的页，采用这种方法，有可能产生较大的块失效率。

2. “近期最少使用”算法（LRU）

- 选择近期最少使用访问的Cache块为调出块，这种算法能比较正确地反映程序的局部性；
- 具体实现比FIFO算法要复杂一些，替换算法只能全部用硬件方法实现。

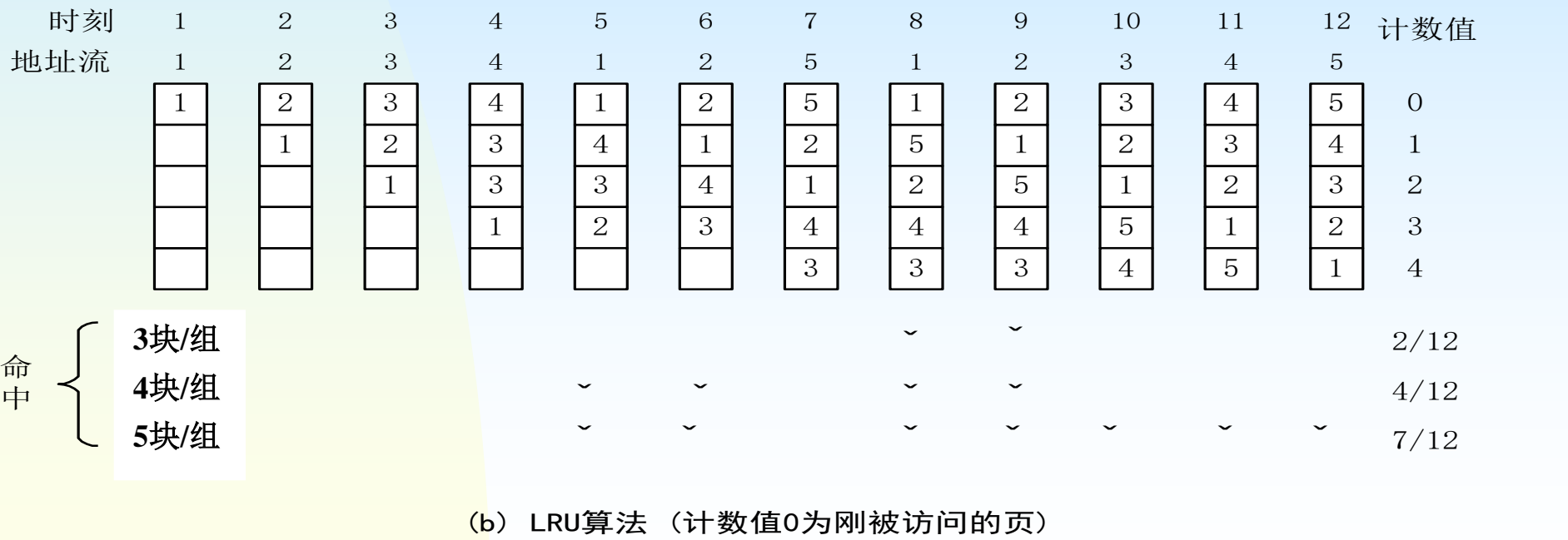
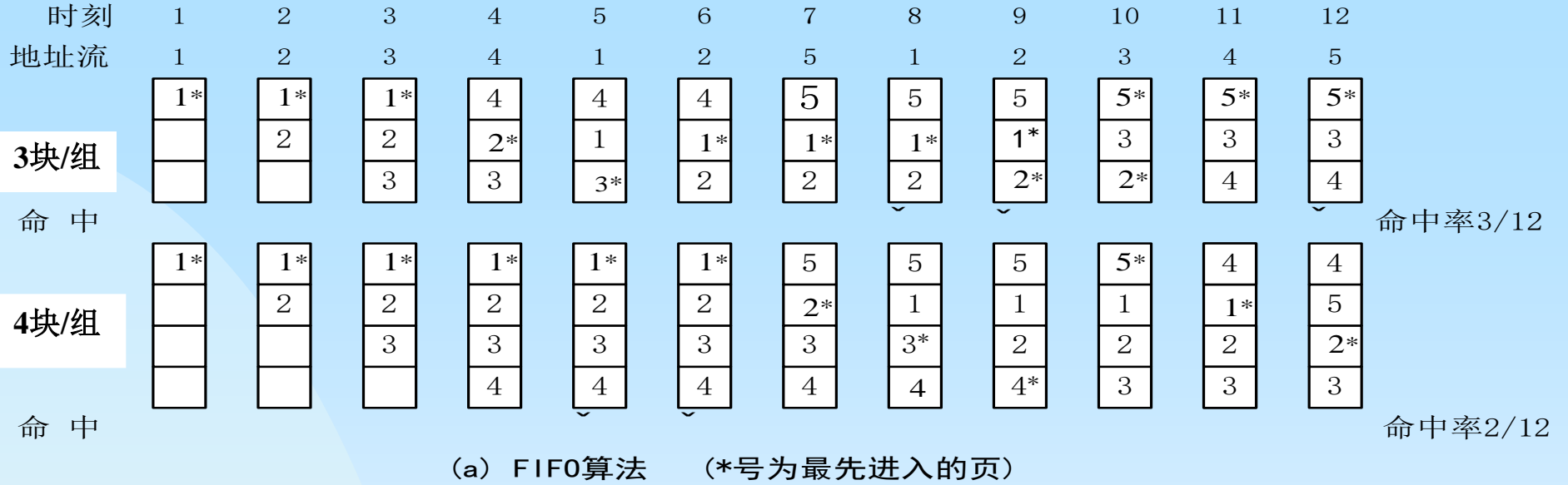


图5.42 Cache两种替换算法的替换和命中率情况

5.5.4 Cache—主存内容的一致性问题

- ◆ CPU执行写操作时，要写的内容恰在Cache中，则Cache内容被更改，但该单元对应的主存内容尚没有改变，这就产生了Cache和主存内容不一致的情况。
- ◆ 解决问题的关键是选择更新主存内容的算法；采用两种算法。
- ▲ ‘写回法’（Write back）
 - 处理机执行写操作时，信息只写入Cache，当Cache块被替换时，将该块内容写回主存后，再调入新块。

▲ 写直达法又称存直达法，

- 处理机进行写操作时，利用“Cache—主存”层次中存在于处理机和主存之间的通路将信息也写回主存。
- 在块替换时，就不必将被替换的Cache块内容写回，可以直接调入新块。

◆ 采用两种算法比较

- 写回法的开销是在块替换时的回写时间；
- 写直达则在每次写入时，都要附加一个比写Cache长得多的写主存时间；写直达法时间的开销大一些，但其一致性保持要好一些。

5.5.5 Pentium Cache结构

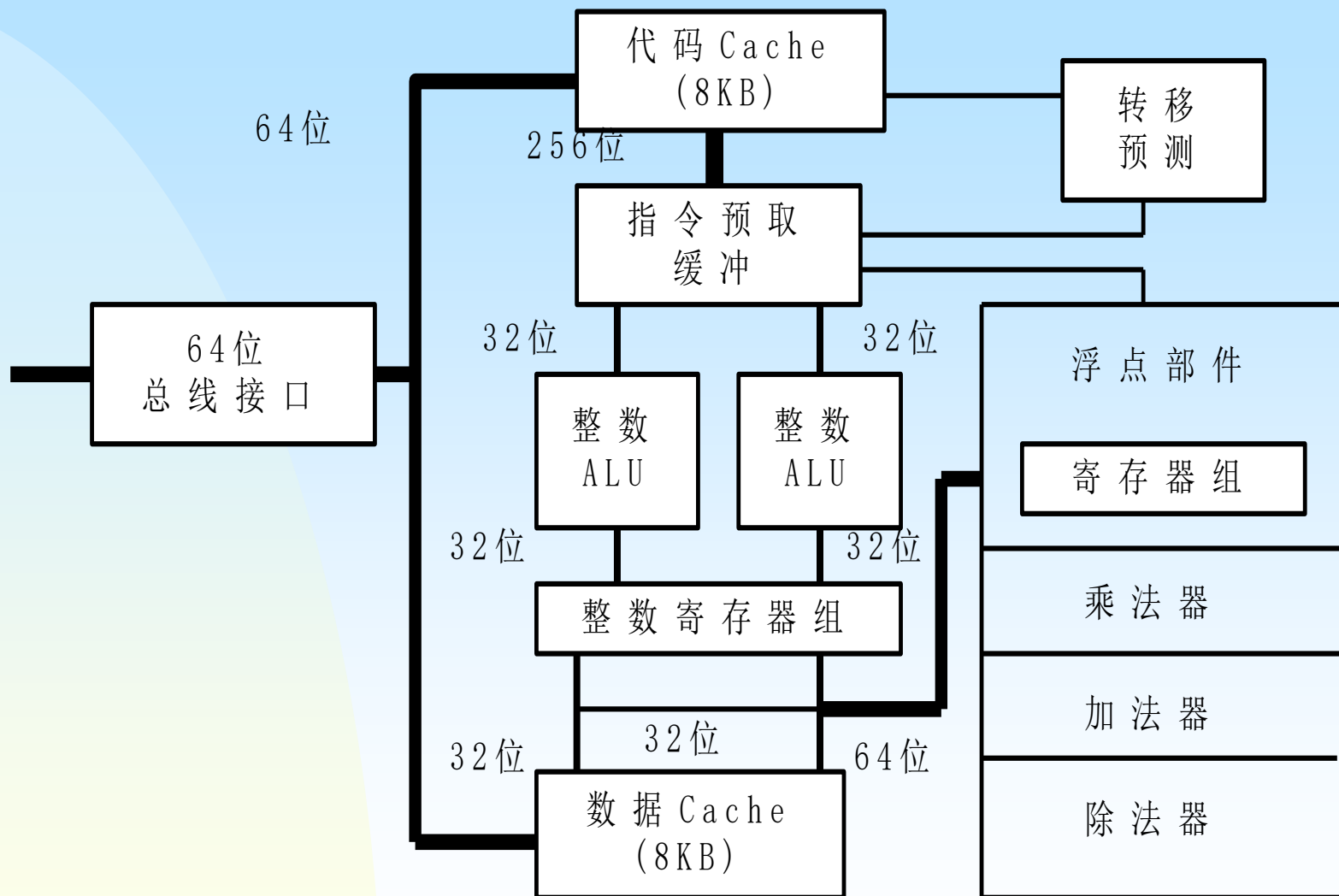


图 5.43 Pentium 片内结构框图

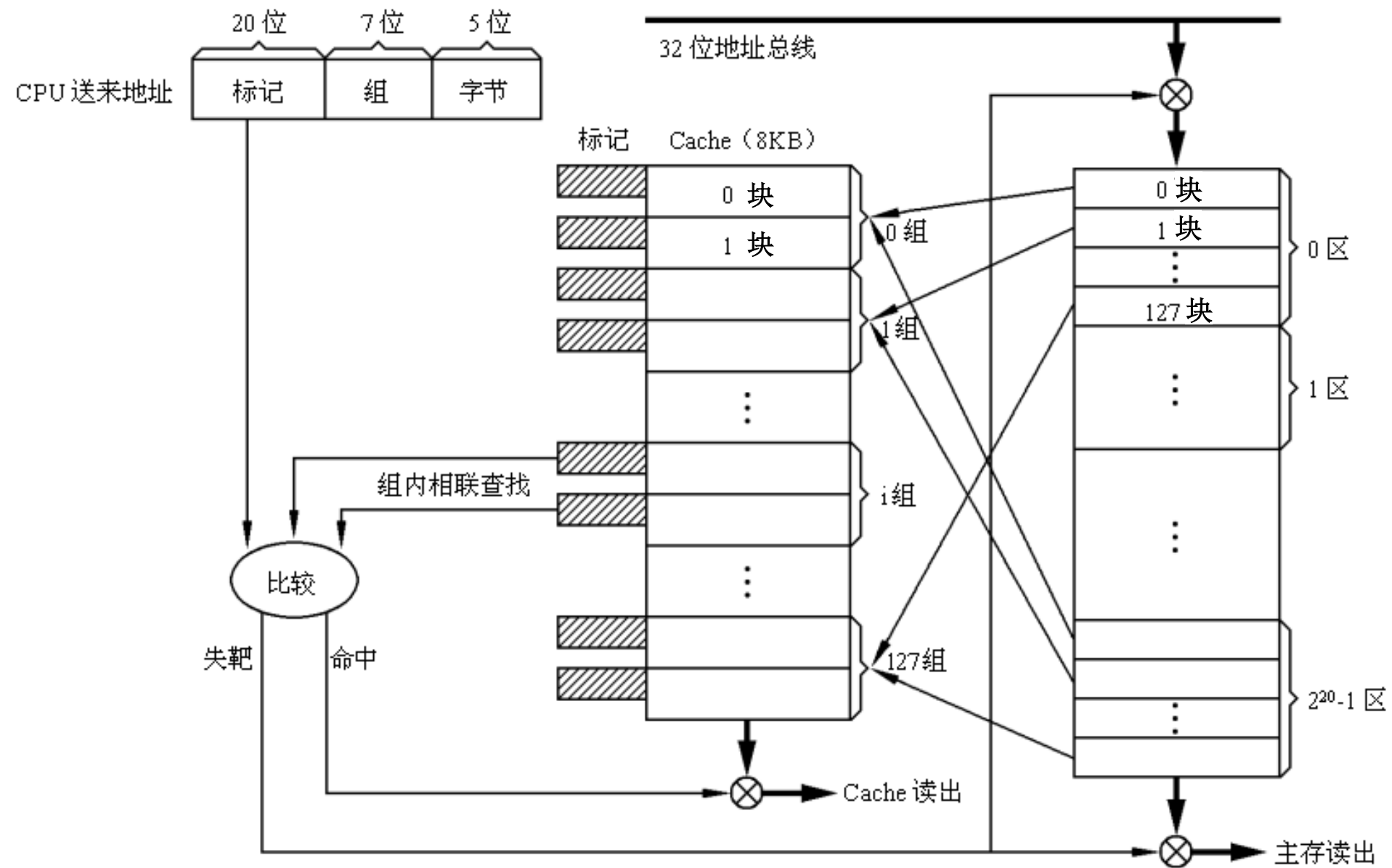


图5.44 Pentium Cache组相联映像

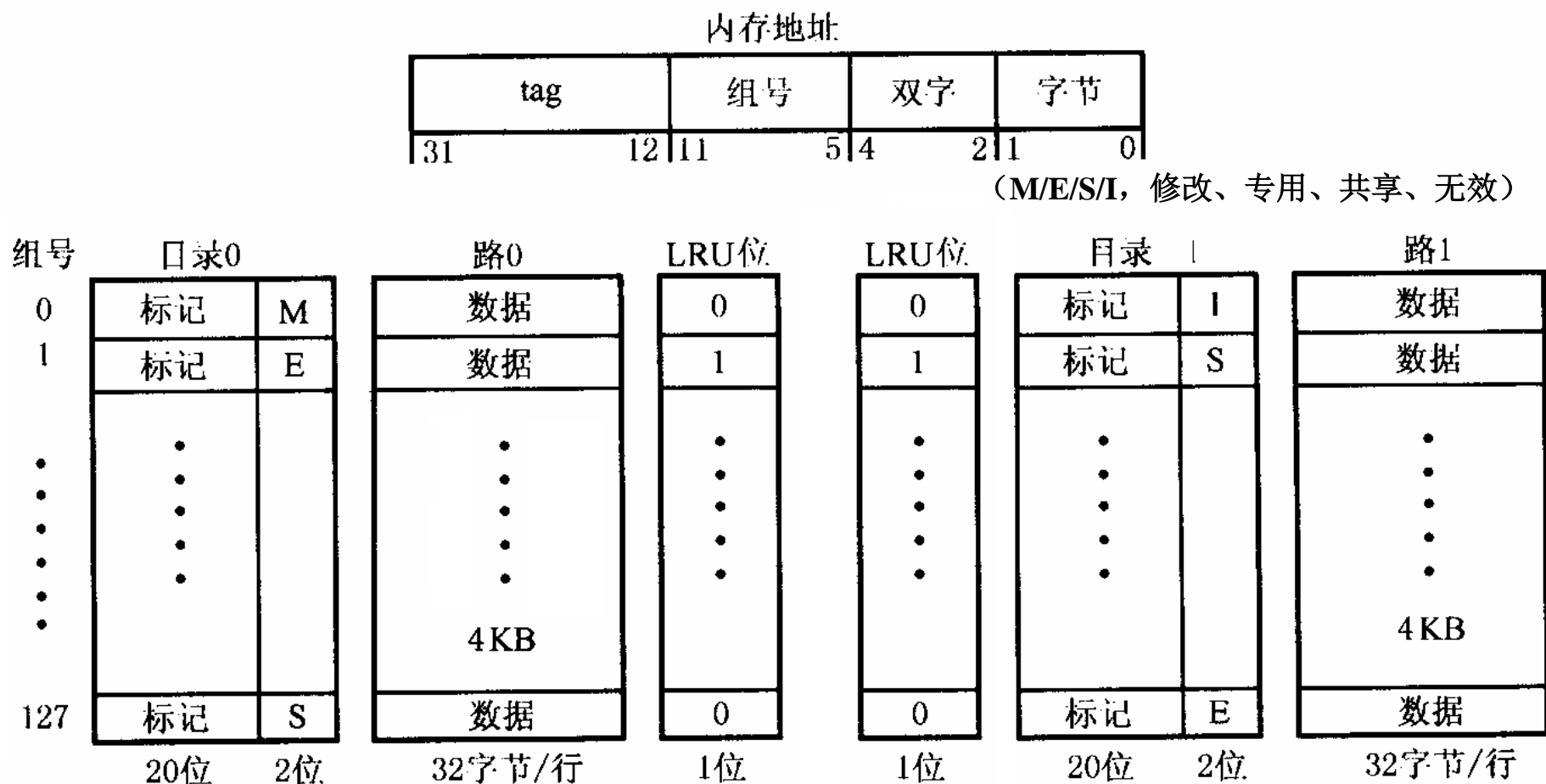


图5.45 Pentium片内Cache目录表和LRU位

5.6 虚拟存储器

5.6.1 虚拟存储器概述

◆ 问题的提出：

- 系统程序和应用程序要求主存容量越来越大，

◆ 虚拟存储技术：

- 将一部分（或全部）辅存和主存结合，把两者的地址空间统一编址，形成比实际主存空间大得多的**逻辑（虚拟）地址空间**。
- 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）。

◆ 虚拟存储器的功能

- 是一种解决**存储容量**和**存取速度**矛盾的一种有效措施，是管理存储设备的有效方法。
- 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。
- 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。

- ◆ “主存—辅存”层次和“**Cache—主存**”层次采用的地址变换及映象方法和替换策略，在原理上是类似的
- **Cache**的替换算法和地址映象方式**完全由硬件实现**，在虚拟存储系统中，用**软、硬件结合的方法来实现**。

5.6.2 虚拟存储器的基本管理方法

- ◆ 虚拟存储器的管理方式有段式、页式或段页式三种。

(1) 段式虚拟存储器管理

- 按照程序的**逻辑结构划分**成多个相对独立部分(过程、子程序、数据表、阵列);
- 用**段表**来指明各段在主存中的位置; 有段名(用户名或数据结构名或段号)、段起点、段长等。

▲ 段式管理系统的优点和不足

- 段的分界与程序的自然分界相对应；段的逻辑独立性使它易于编译、管理、修改和保护，也便于多道程序共享；
- 某些类型的段（堆栈、队列）具有动态可变长度，允许自由调度以便有效利用主存空间。
- 由于段的长度各不相同，段的起点和终点不定，给主存空间分配带来麻烦，且容易在实存中留下许多空白的零碎存储空间不好利用，造成浪费。

(2) 页式管理系统

- 基本信息传送单位是定长的页，主存的物理空间也被划分为等长的固定区域，称为页面。
- 比段式管理系统中段外空间的浪费要小的多。
- 由于页不是逻辑上独立的实体，所以处理、保护和共享都不及段式来得方便。

(3) 采用分段和分页结合的段页式存储管理系统

- 程序按模块分段，段内再分页，进入主存仍以页为基本信息传送单位；
- 用段表和页表（每段一个页表）进行两级定位管理。

1. 页式虚拟存储器

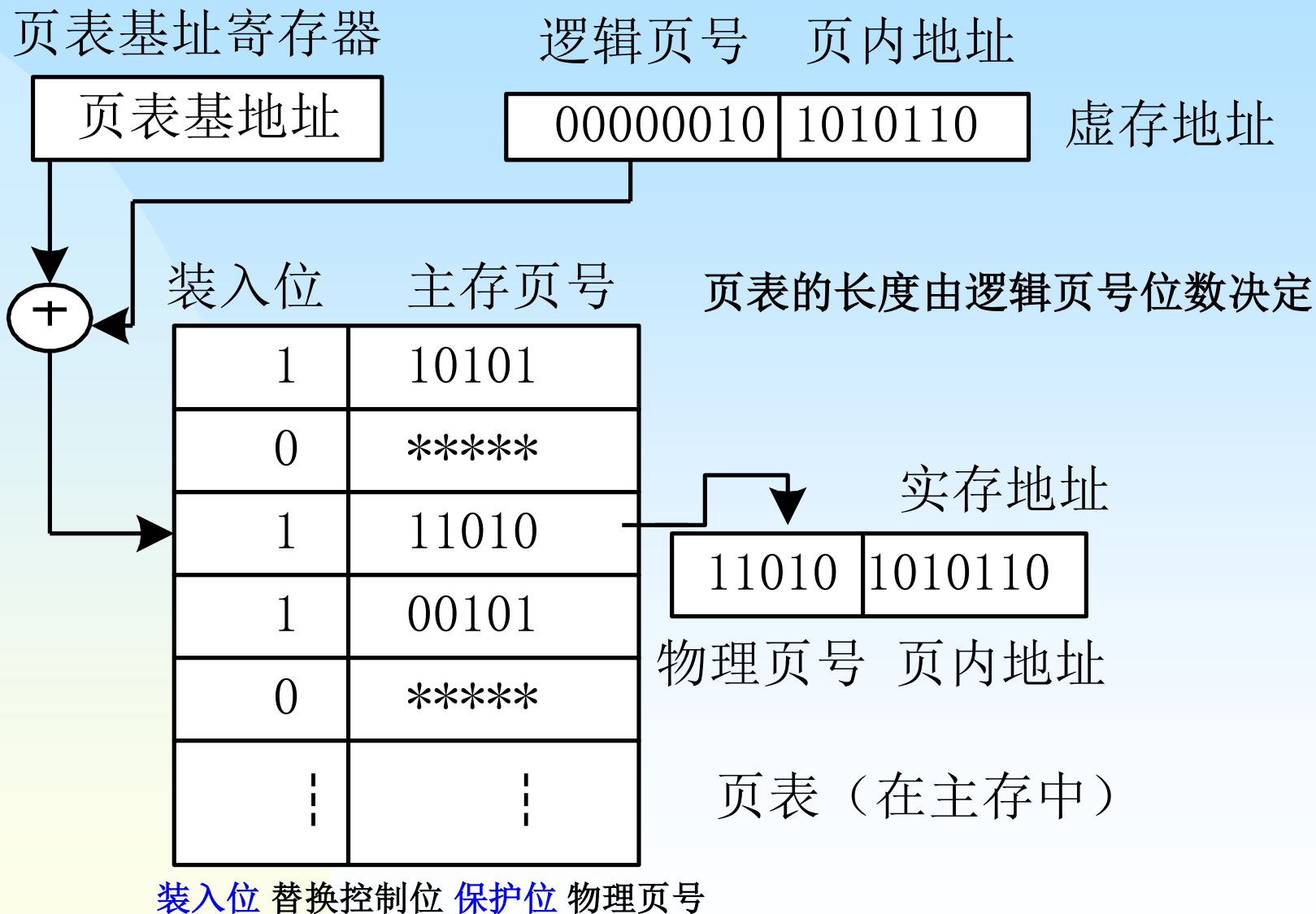


图5.46 页式虚拟存储器地址变换

▲ 经快表与慢表实现地址变换的方式

- 快表由硬件组成，比页表小得多；由逻辑页号同时去查快表和慢表，如果在快表中查不到，要花费一个访主存时间查慢表。

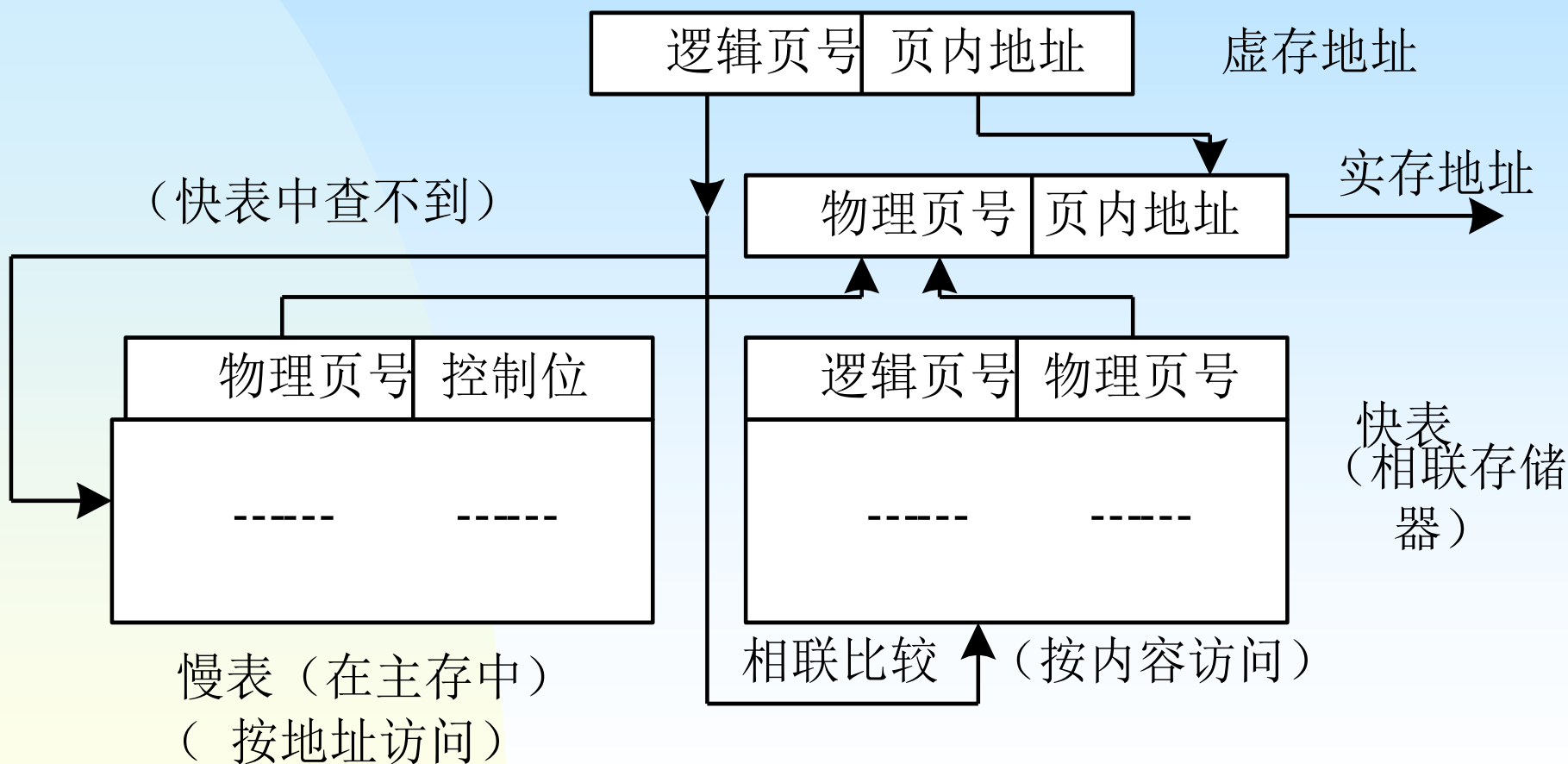


图5.47 经快表和慢表进行地址变换

2. 段式虚拟存储器

▲ 虚拟地址由段号和段内地址组成，通过**段表**把虚拟地址变换成实存地址。

• 段表格式

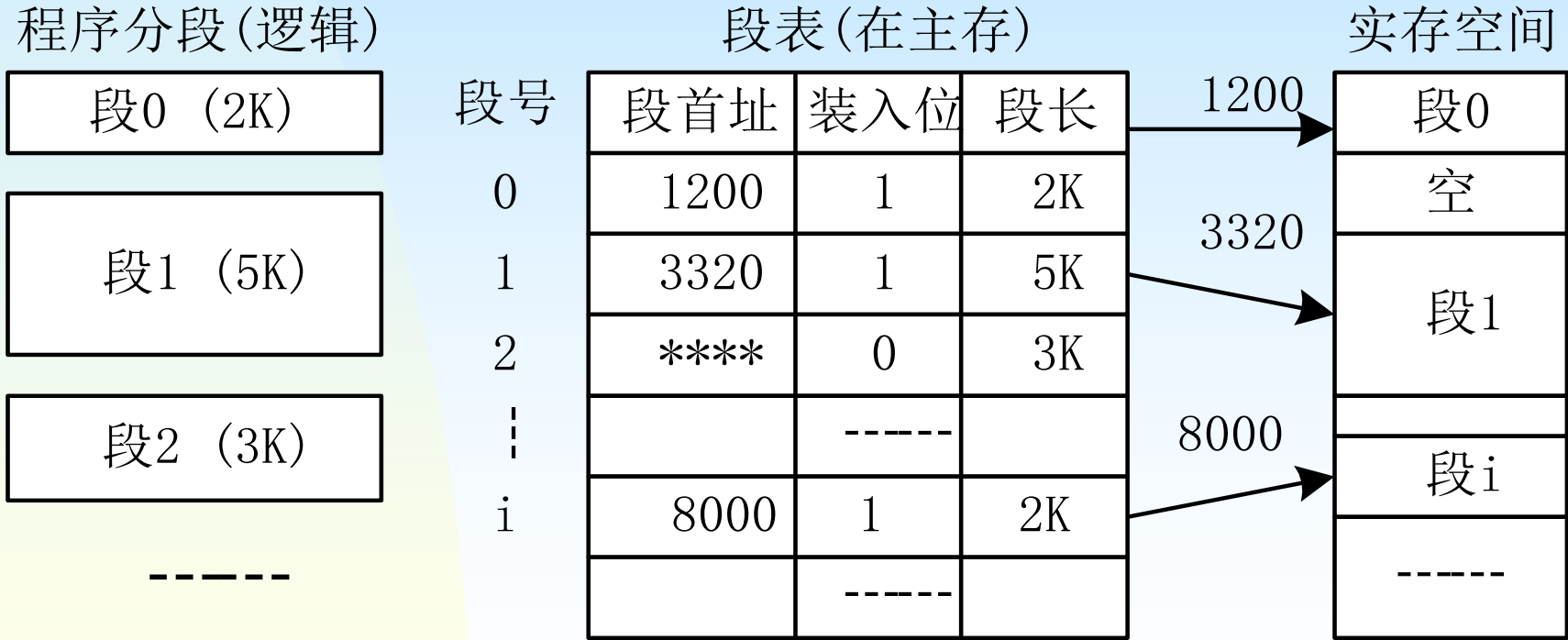


图5.48 段表示意图

- 在访问某段时，如果段内地址值超过段的长度，则发生地址越界中断。
- 虚拟地址向实存地址的变换过程

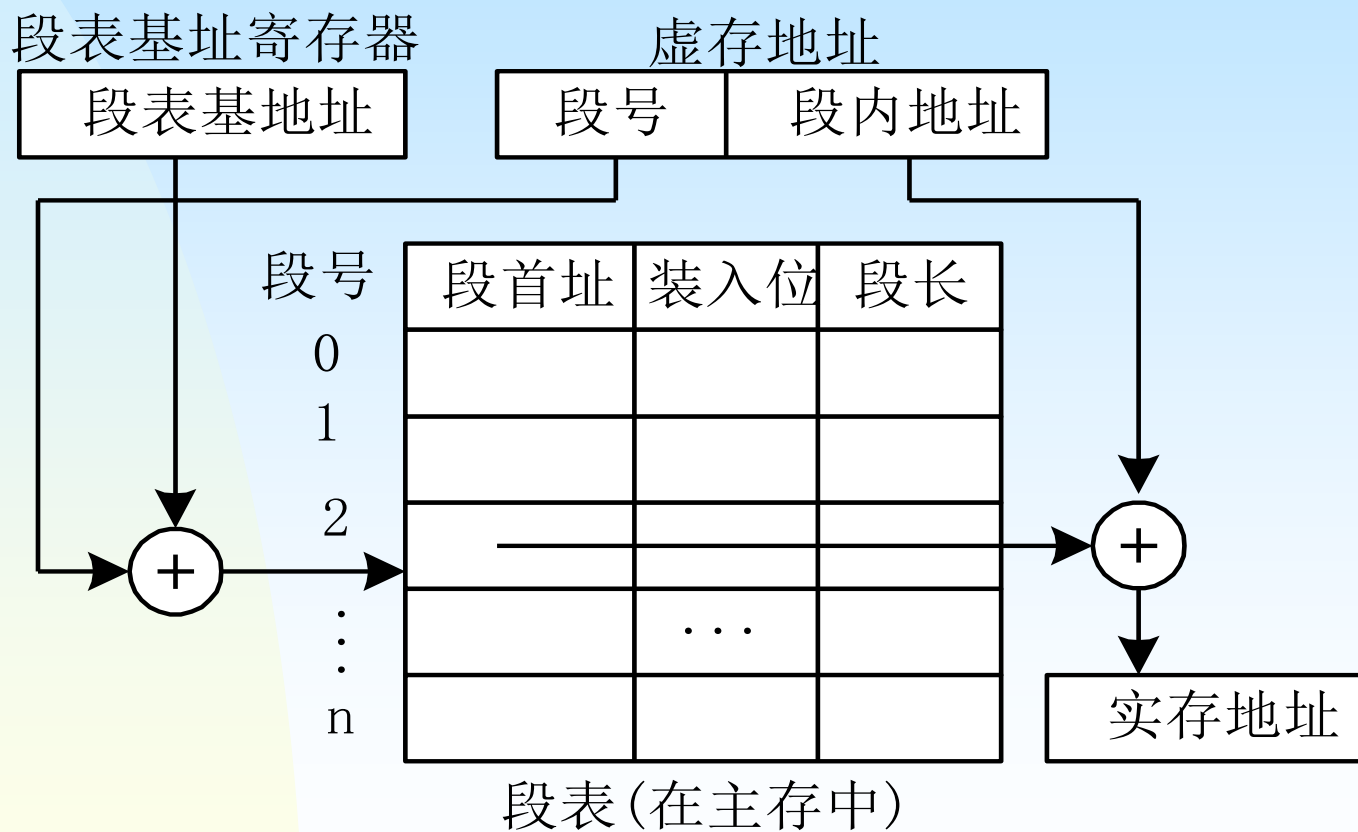


图5.49 段式虚拟存储器地址变换

3. 段页式虚拟存储器

- ▲ 把程序按逻辑单位分段后，再把每段分成固定大小的页；程序对主存的调入调出是**按页面进行的**，按段实现共享和维护。
 - 它兼有页式和段式的优点。缺点是在地址映象过程中**需要多次查表**。
- ▲ 每道程序是通过**一个段表**和**一组页表**来进行定位
 - **段表中**每个表项有一个指向该段的**页表起始地址的指针**和该段的**控制保护信息**。
 - **页表**指明该段各页在**主存中的位置**以及是否已装入、修改等**状态信息**。
 - 目前，大、中型机一般都采用这种段页式存储管理方式。

▲ 多道程序虚拟地址

- 用基号（用户标志号）指明该道程序的段表起始地址（存放在基址寄存器中）。
- 多道程序虚拟地址格式：

基号	段号	页号	页内地址
----	----	----	------

- 若只有一个基址寄存器，基号可以不要，在多道程序切换时，由操作系统修改基址寄存器的内容来实现。
- 例：有三道程序（用户标志号为P1，P2，P3），其基址寄存器内容分别为B1，B2，B3；

- 逻辑地址到物理地址的变换

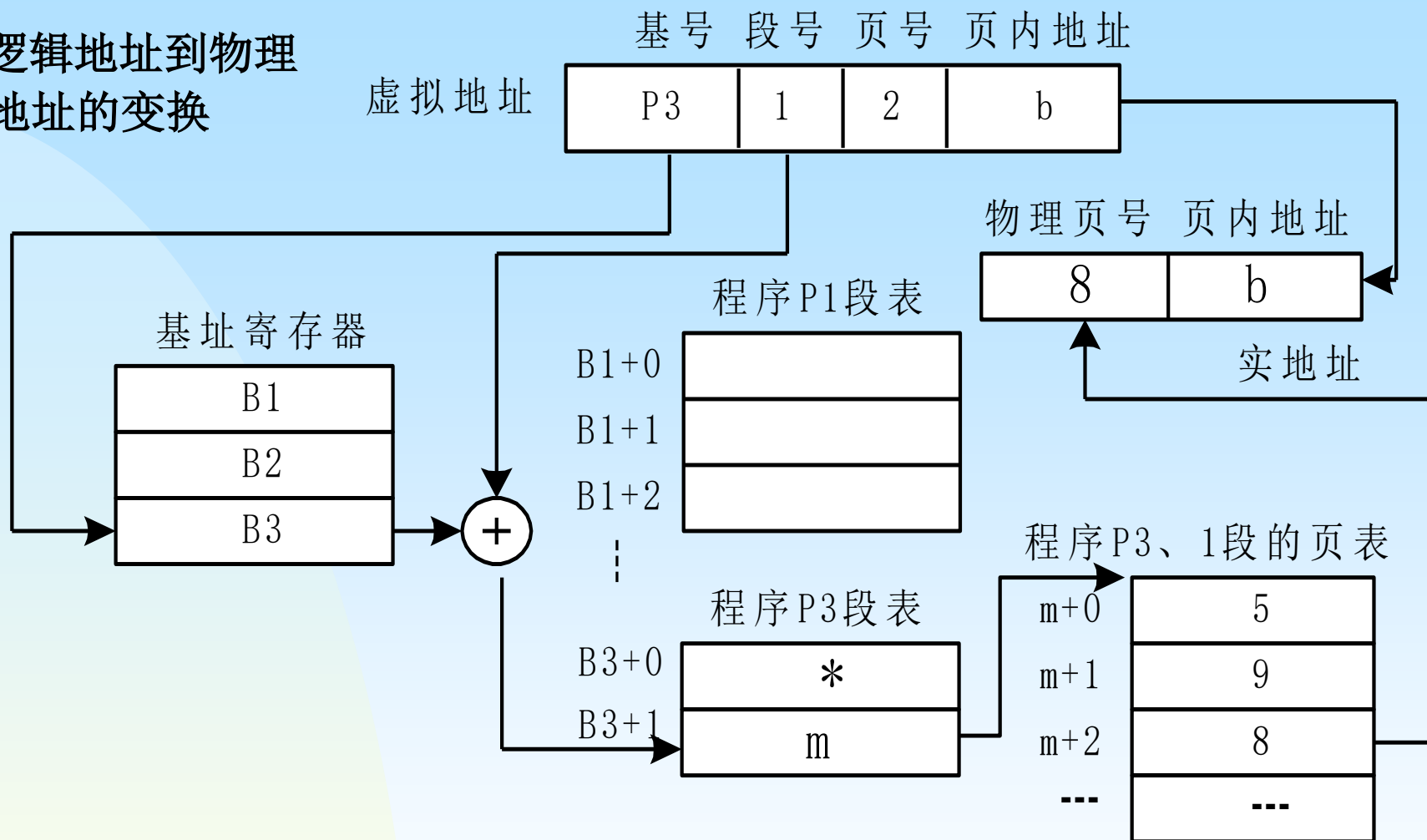


图5.50 段页式虚拟存储器地址变换

- 上述每一张表的每一行都要设置一个有效位；若有效位均为“0”，访问失败，则发中断请求操作系统建表。

4. 替换算法

- ▲ 页式管理中，产生页面失效时，要从外存调进包含有这条指令或数据的页面。
- ▲ 假如主存页面已全部被占满，通常采用LRU算法，把“近期最少使用的页”替换出去。
- ▲ 主存中的内容与外存保持一致
 - 页表的每一行设置一修改位，当该页刚调入主存时，此位为“0”，当对该页内任一地址进行写入时，就把该位修改为“1”(主存中的内容与外存不一致)。
 - 在页被替换时，检查其修改位，如为“1”，先将该页内容从主存写入外存，然后再从外存接收新的一页。

◆ Pentium存储器的地址转换方式

逻辑地址（48位）

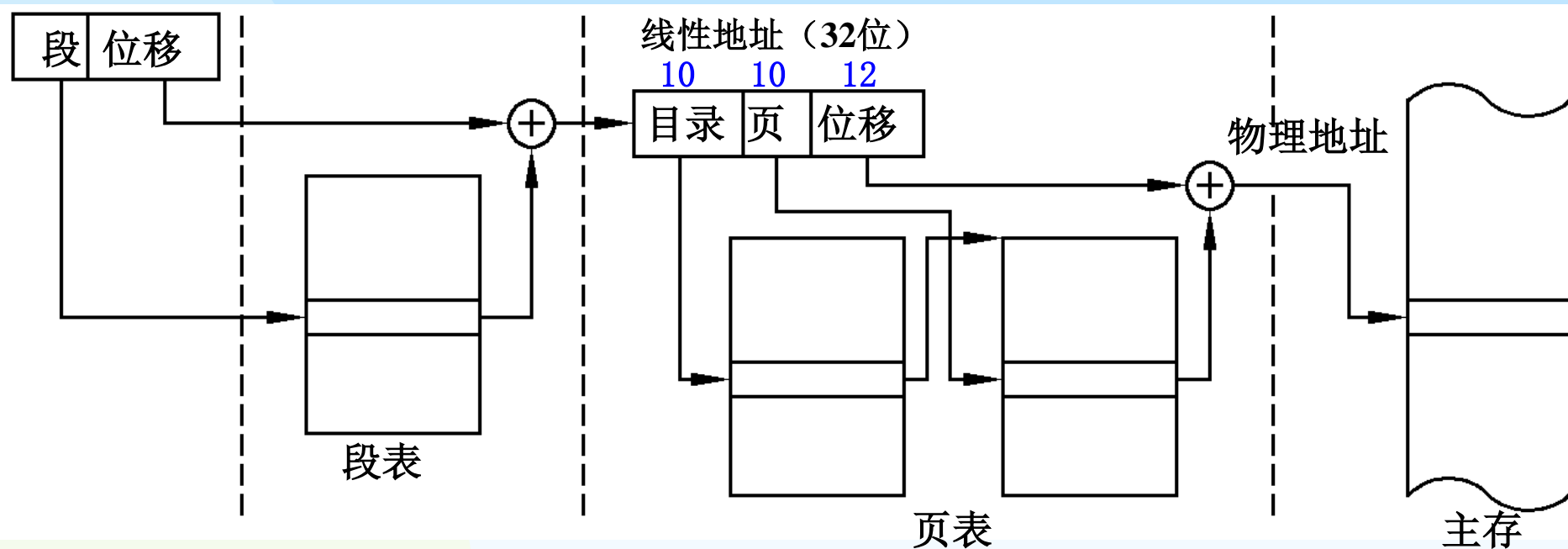
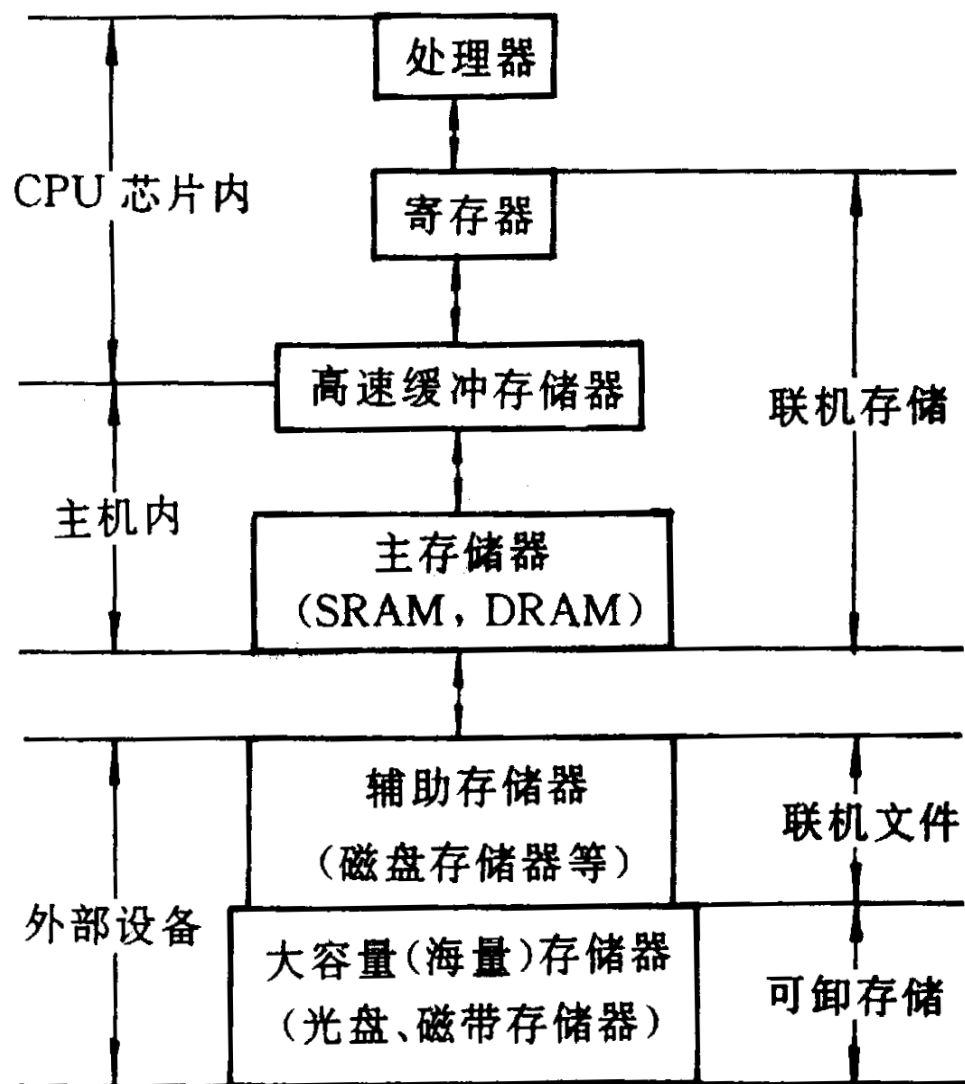
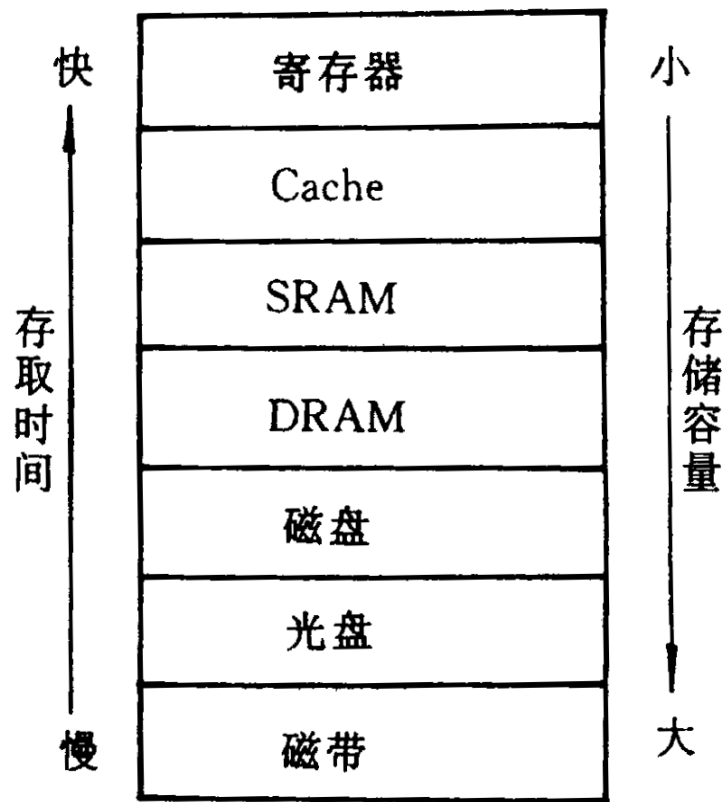


图5.51 Pentium存储器地址转换过程

◆ Pentium存储器结构根据其段表和页表是否设置有4种组合情况

- 1) 无段表和无页表的存储器，属于非虚拟存储器，其逻辑地址即为物理地址；
- 2) 无段表和有页表的存储器，页式虚拟存储器，存储器的管理和保护是通过页面转换实现的；
- 3) 有段表和无页表的存储器，段式虚拟存储器；
- 4) 有段表和有页表的存储器，段页式虚拟存储器。

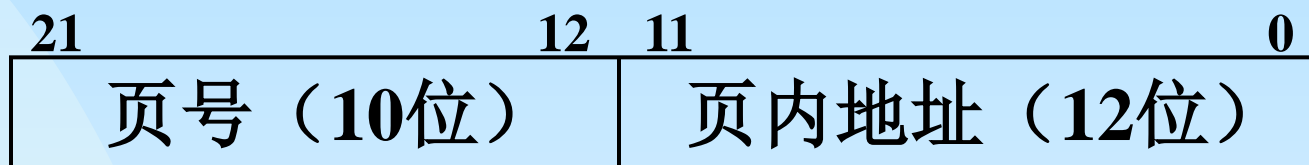
◆ 计算机存储系统的层次结构



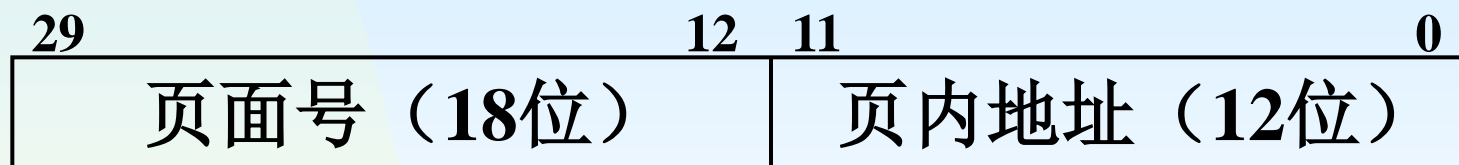
例. 设主存容量4MB，虚存容量1GB，页面大小4KB。采用页式虚拟存储器。

- (1) 写出主存地址格式。
- (2) 写出虚拟地址格式。
- (3) 页表长度为多少？
- (4) 画出虚实地址转换图。

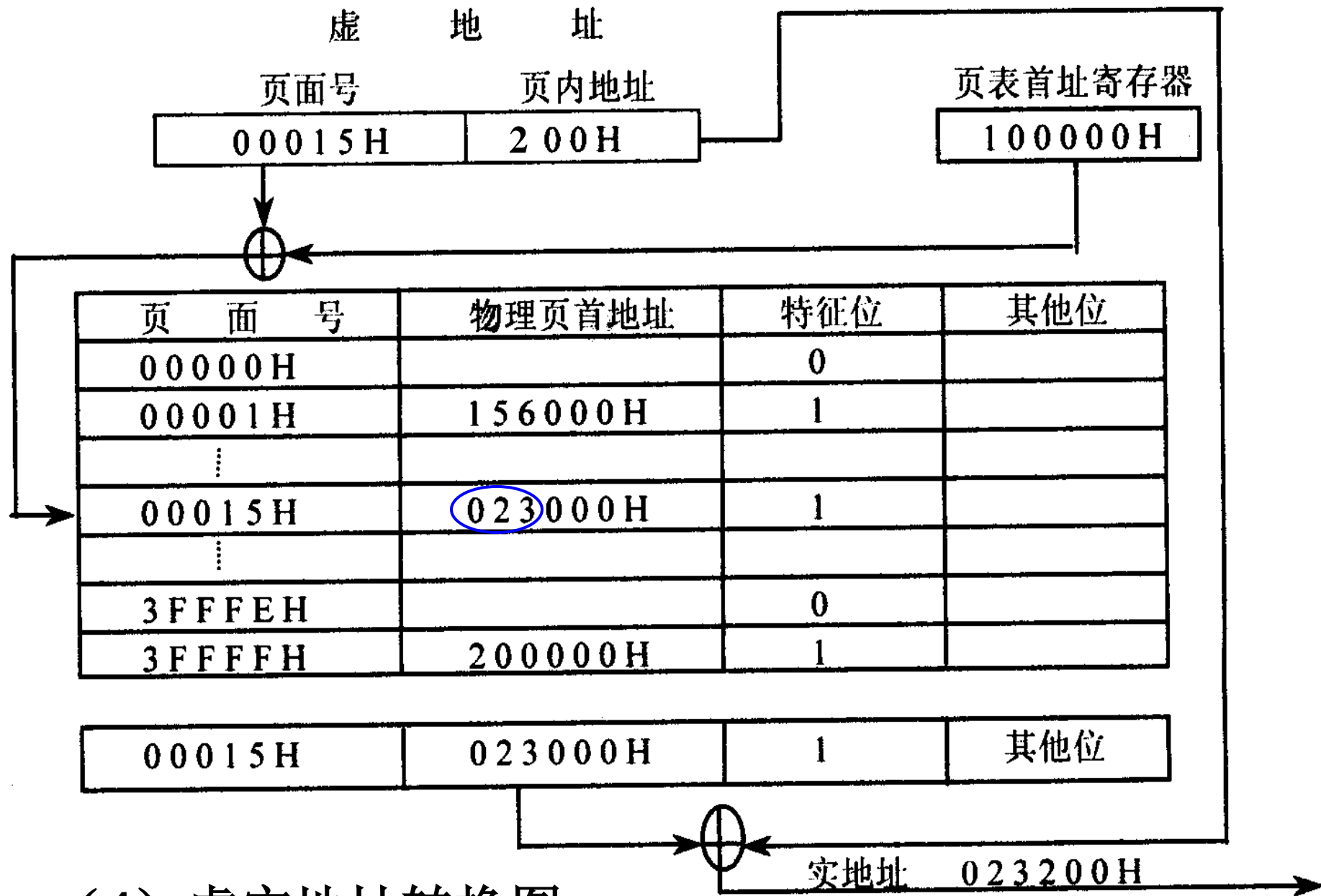
(1) 主存地址格式为:



(2) 虚拟地址格为:



(3) 页表长度为 $2^{18}=256K$



(4) 虚实地址转换图

5.6.3 存储保护

- ◆ 多个用户对主存的共享，就有多个用户程序和系统软件存于主存中。

1) 存储区保护

- ◆ 在虚拟存储器系统中，一个用户程序的各页能离散地分布于主存空间，不能使用界限寄存器保护方式；

(1) 段、页表保护

- 每个程序都有自己的页表和段表，段表和页表本身都有自己的保护功能。
- 段表、页表保护是在没形成主存地址前的保护。

(2) 键保护方式

- 键保护方式的基本思想是为主存的每一页配一个键，称为**存储键**；由操作系统赋予。
- 打开存储键的钥匙，称为**访问键**。访问键赋予每道程序，保存在该道程序的状态寄存器中。
- 当数据写入主存的某一页时，访问键要与存储键相比较；若两键相符，则允许访问该页，否则拒绝访问。

(3) 环保护方式

- **环状保护方式**可以做到对**正在执行的程序**本身进行保护。

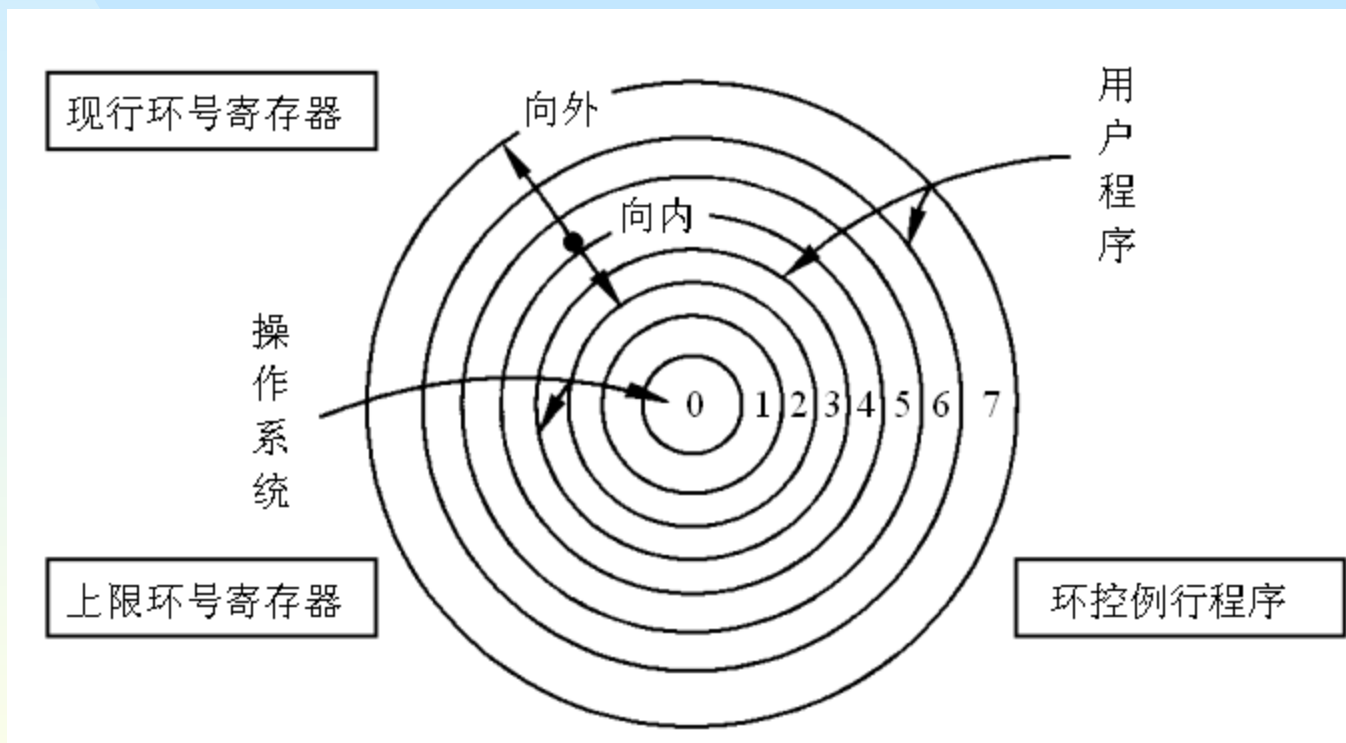


图5.52 环保护方式

2) 访问方式保护

- 对主存信息的使用有三种方式：读（R）、写（W）和执行（E）；对这三种分别设置访问方式保护；
- 访问方式保护是结合上述保护进行的，即在各保护方式中设置访问方式位；
- 在界限寄存器中加一位访问方式位；
- 采用环式保护和页式保护时将访问方式位放在页表和段表中，使得同一环内或同一段内的各页可以有不同的访问方式，从而增强了保护的灵活性。

5.7 辅助存储器

5.7.1 磁表面存储器

- ◆ 磁表面存储器包括磁鼓、磁带、磁盘和磁卡片等。
目前，在计算机系统中以**磁盘和磁带**为主。
- 磁表面存储器主要用作辅存，可存储大量的程序和数据，需要时，调入主存供CPU访问。属I/O设备。
- ◆ 本节主要介绍磁表面存储器的存储原理、数据的记录方式以及**磁盘结构及寻址方式**。

1. 磁表面存储原理

(1) 磁层和磁头

- ▲ 磁层是存放信息的介质，由非矩形剩磁特性的导磁材料（氧化铁、镍钴合金等）构成。
 - 磁材料制成的胶涂敷或镀在载磁体上，其厚度通常为 $0.1\sim 5\text{ }\mu\text{m}$ ，以记录信息。
- ▲ 磁头是实现“磁—电”和“电—磁”转换的元件。
 - 磁头中的缝隙形状和尺寸将直接影响记录密度和读出幅度。

(2) 磁表面存储器的读写过程

- ▲ 磁头固定，磁层作高速回转或匀速直线运动；在相对运动中，通过磁头缝隙对磁层进行信息存取。

① 信息写入过程

- 写线圈中通以写电流脉冲，磁头缝隙处的磁场穿过磁层中一微小区域，使该区域磁层以一定方向磁化，且保持该方向的剩磁（ $+\mathbf{BR}$ 或 $-\mathbf{BR}$ ）。

② 信息读出过程

- 磁头与磁层作相对运动，当磁层中记录单元运动到磁头缝隙下面时，在磁头中产生较大的磁通变化，读线圈两端产生较大的感应电势 E ，经整形、放大后成为读出信号。

(3) 磁表面存储器的性能指标

① 记录密度

- 记录密度可用**道密度**和**位密度**来表示。
- 在沿磁道分布方向上，单位长度内的磁道数目，叫道密度。单位为**TPI**（每英寸磁道数）。
- 在磁道中，单位长度内存放的二进制信息的数目叫位密度。位密度的单位为**bPI**（每英寸二进制位数）。软盘位密度约每英寸**4000~10000**位。

② 存储容量

- 指整个存储器所能存放的二进制字数或字节数。它与磁表面大小和记录密度密切相关。
- 磁鼓容量最小，为几兆字节。磁带容量最大，为若干亿字节；磁盘介于两者之间。

③ 平均存取时间

- 存取时间应包括**定位和等待时间**。
- 磁盘的平均寻道时间一般为10~20ms；等待时间一般取磁盘旋转一周所需时间的一半。
- 磁带的定位时间与走带速度和带长有关，带愈长，带速愈慢，其平均存取时间愈长。

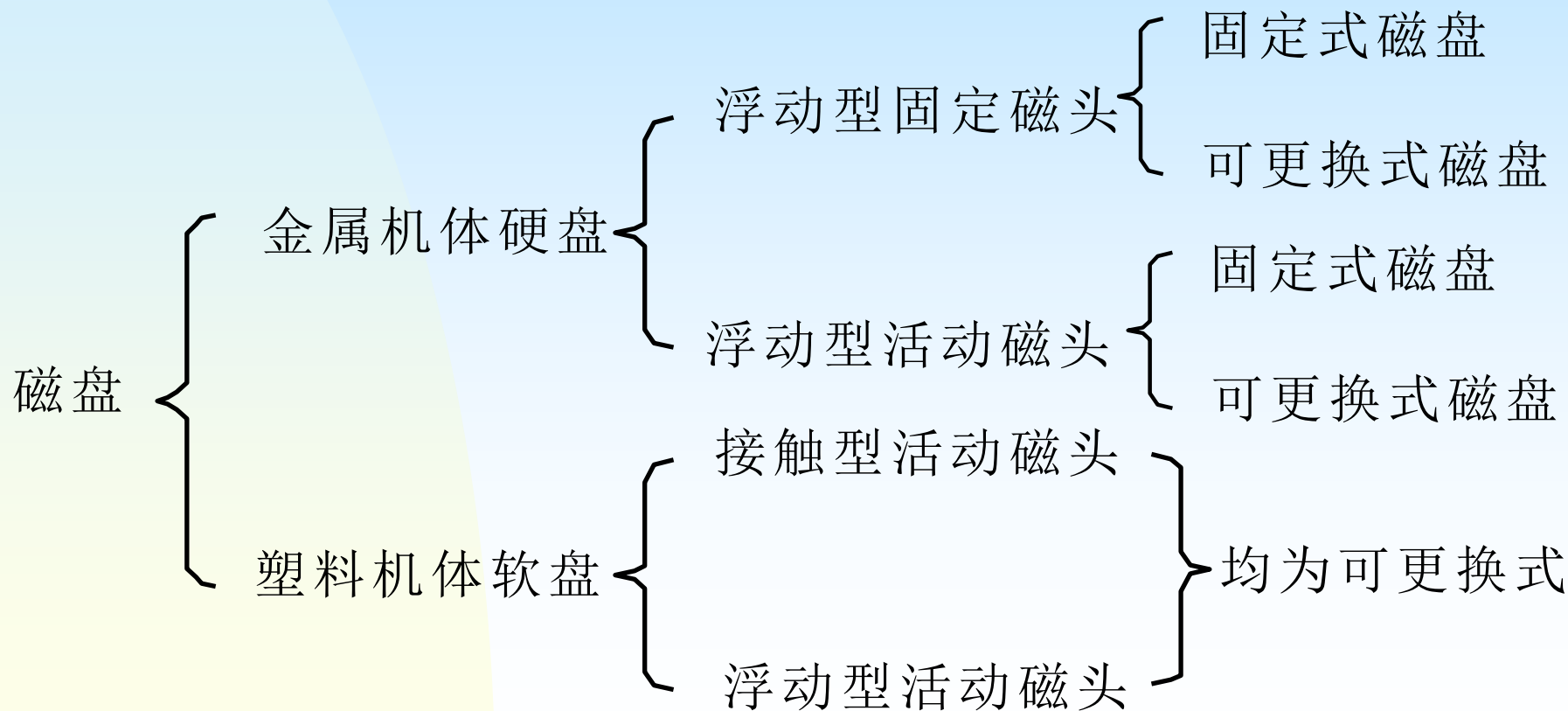
④ 数据传送速率

- 指磁表面存储器完成**定位和等待操作以后**，单位时间内与主机交换数据的二进制信息量，以位/秒或字节/秒表示。
- 由于磁表面存储器和主存之间的数据交换常以**批量方式**进行，故传送速率较大。如软磁盘传送一个字节约为几个微秒，而硬盘每秒可传送几兆字节。

2. 磁盘存储器

▲ 以磁盘或磁盘组作为存储介质的记录装置。

▲ 根据盘片的构成及其可更换性分类：



(1) 硬盘存储器

① 硬盘存储器的基本组成

- 主要由磁记录介质、磁盘驱动器、磁盘控制器三大部分组成。

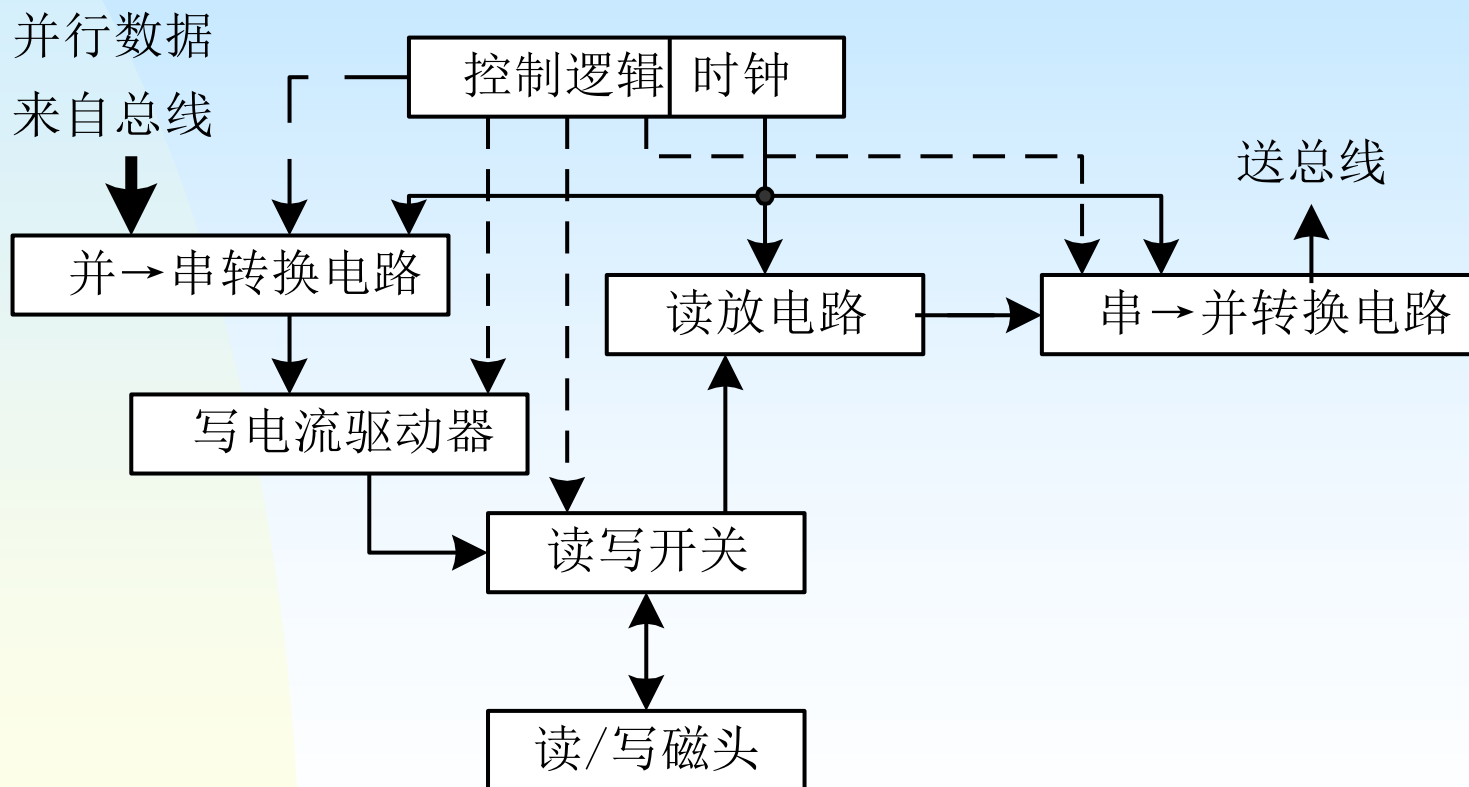


图5.59 硬盘存储器的逻辑结构

② 温彻斯特磁盘存储器

- 可移动磁头转动盘片的磁盘存储器，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。
- 温盘的盘片直径有14吋、8吋、5.25吋和3.5吋等几种，用于IBM PC 系列机的温盘一般是3.5吋的，现在容量可达20GB以上。

③ 硬盘驱动器

- 硬盘驱动器是精密的机电装置，由定位驱动系统、主轴系统和数据转换系统组成。

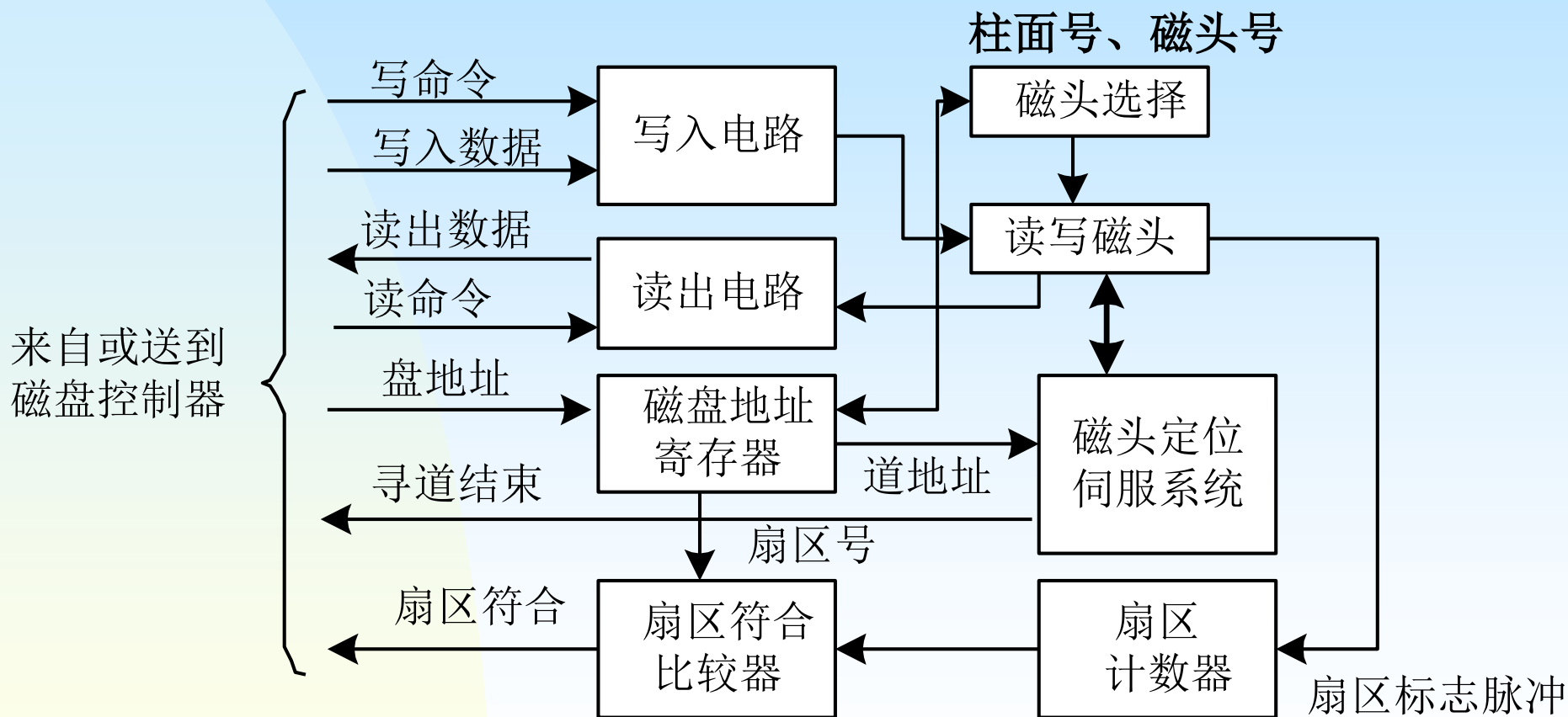


图5.60 磁盘驱动器逻辑框图

④ 磁盘控制器

- 磁盘控制器是主机与磁盘驱动器之间的接口；与和主机之间采用成批数据交换方式。

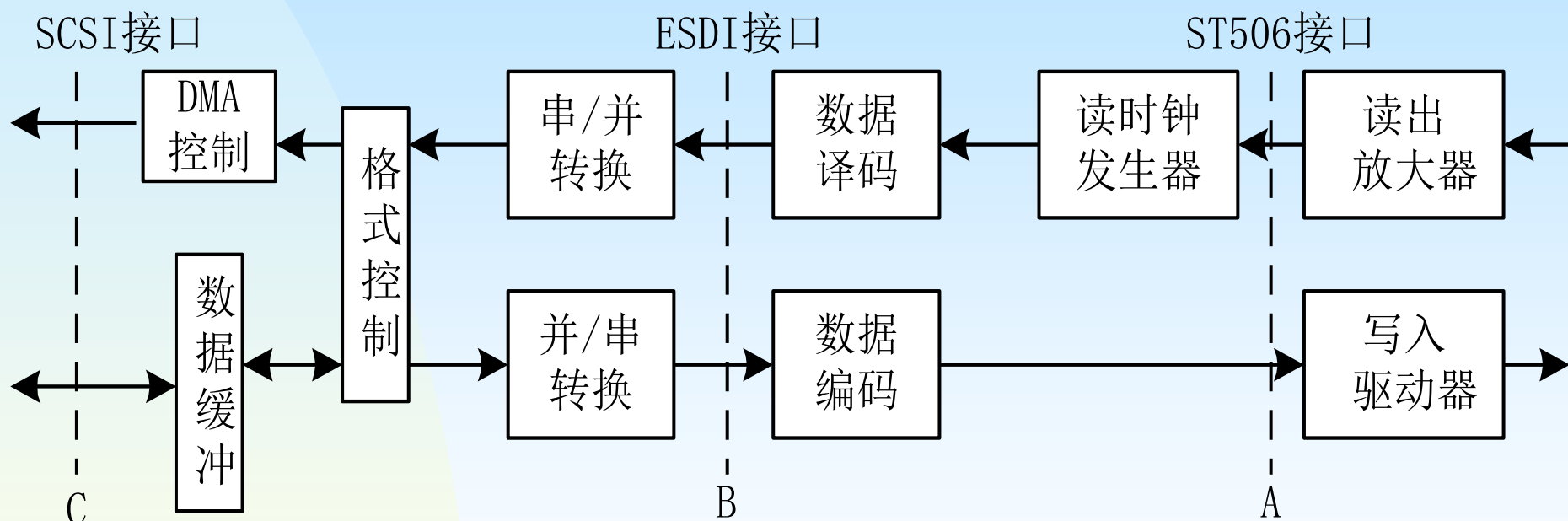


图5.61) 磁盘控制器接口逻辑

- 磁盘控制器与磁盘驱动器之间并没有明确的界线，

⑤ 磁盘上的信息分布

- 磁盘表面称为记录面，记录面上一系列同心圆称为磁道（**track**），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。
- 磁盘组上直径相同的磁道构成一个柱面(**Cylinder**)，柱面从外向里编址。

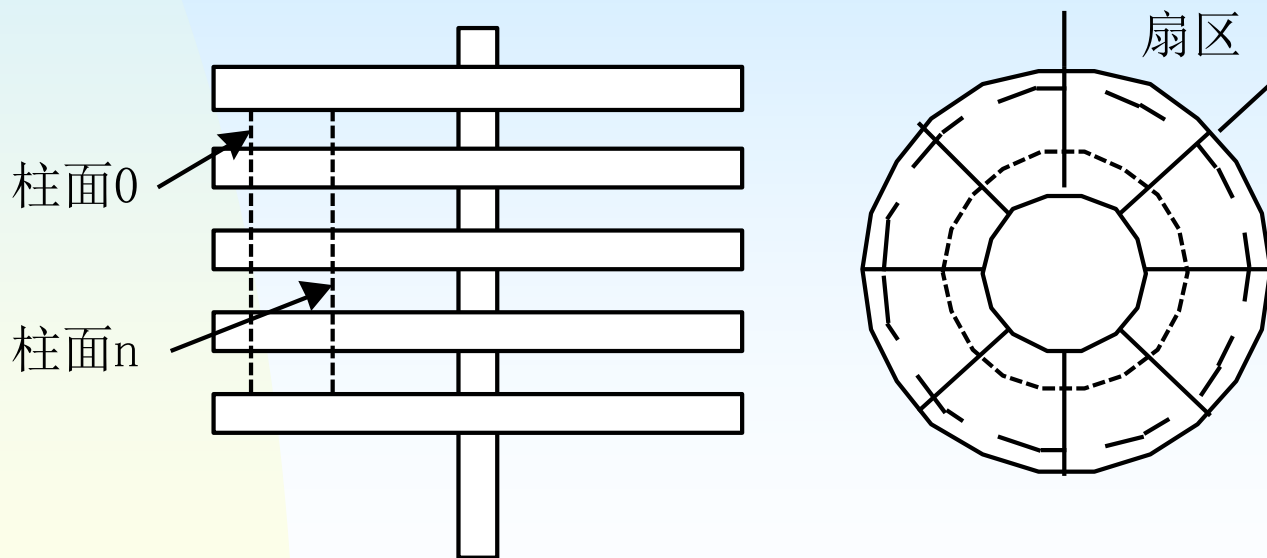
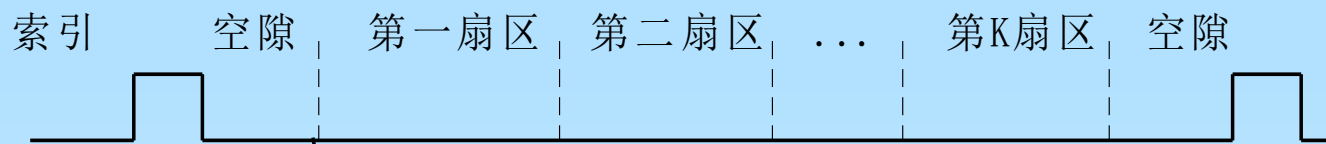


图3.62 磁盘的柱面和扇区示意



(a) 磁道上信息的分布

信息名		头空	ID域				间隙	数据域				尾空
			同步字符	地址标志	地址	CRC		同步字符	数据标志	数据	CRC	
FM制	字节数	40	6	1	4	2	11	6	1	128 256 512	2	27
	代码	FF	00	FE			FF	00	F8			FF
MFM制	字节数	80	12	4	4	2	22	12	1	128 256 512	2	54
	代码	FF	00	3个A1 1个FE			4E	00	F8 或 FB			FF

(b) IBM格式的记录块组成

图5.63 IBM PC机所用软盘的记录格式

- PC机软盘的记录格式，硬盘的记录格式与之类似

⑥ 硬盘容量和数据传输率的计算

(a) 未格式化容量的计算

- 未格式化容量=记录面数×理论柱面数×内圆周长×位密度 (bit)
- 理论上的柱面数应该等于 (磁盘有效记录区外径-有效记录区内径) ÷ 2 × 道密度。
- 磁道能记录的二进制信息位的理论值等于内圆周长×位密度；内圆的记录密度最大。

(b) 格式化后容量（实际容量）的计算

- **实际容量** = 记录面数 × 实际柱面数 × 每道扇区数 × 每扇区的字节数（字节容量）

(c) 数据传输率

- 数据传输率是单位时间内磁盘与主机交换数据的二进制信息量；
- 数据传输率 = (每分钟转速 ÷ 60) × 内圆周长 × 位密度 (bit/S)

3.7.2 光盘存储器

- ◆ 光盘存储器是一种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。
- 具有记录密度高、存储容量大、信息保存寿命长、工作稳定可靠、环境要求低等特点，广泛应用于文件、声音和图象等，各种数字化信息的存储。

1. 光盘存储器的分类

▲ 只读型光盘

- 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LV、CD、CD-ROM，已相当普及。

▲ 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。

▲ 可重写型光盘是读/写型光盘，第一代是改写型，第二代是重写型。

- 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。

2. 光盘存储器的基本工作原理

(1) 只读型光盘的信息存储机理

- 所有只读型光盘上的信息都以坑点形式分布，一系列的坑点（信息元）形成信息记录道。

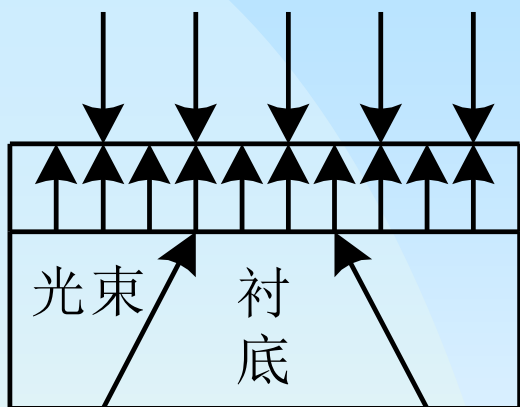
(2) 一次写入型光盘的信息存储机理

- 记录数据以直接可读的方式录刻在光盘上。一旦录制完毕，存储数据就不能再行改写。
- 用强激光束照射薄膜，可在膜层上烧蚀或蒸发出凹坑点（信息元）。

3. 磁光盘存储器

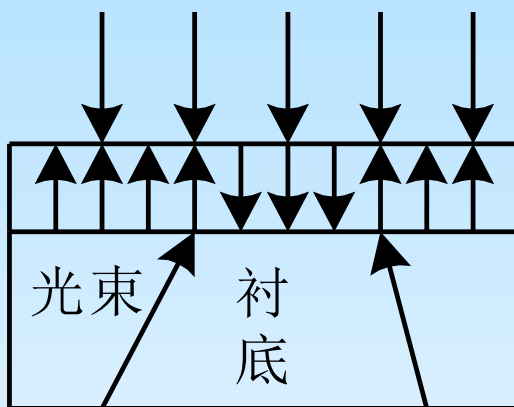
- ▲ 磁光存储技术采用热磁光记录方法来完成写擦功能。又称热磁写入。
- 磁光介质由铁磁材料薄膜组成，在垂直于表面的方向上十分容易磁化。

▲ 磁光写入（擦除）原理



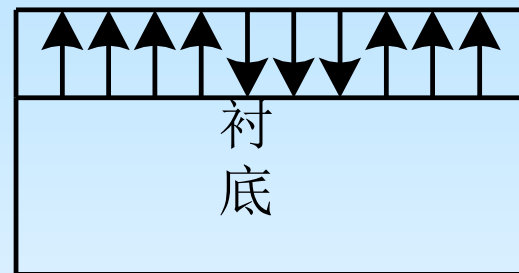
$T < \text{室温}$
 $H_u < H_c(\text{室温})$

(a) 初始状态



$T = T_c$
 $H_c = 0$

(b) 中间状态



$T = \text{室温}$
 $H_u = 0$

(b) 最终状态

图5.66 磁光记录原理示意图

作业一： P143—5、 8、 12、 14

作业二： P144—17、 19、 20、 22