

计算机组成原理

2017年修订

西南交通大学信息科学与技术学院
唐慧佳 hjtang@home.swjtu.edu.cn



第5章 存储系统和结构

§ 5.1 存储系统的组成

§ 5.2 主存储器的组织

§ 5.3 半导体随机存储器和只读存储器

§ 5.4 主存储器的连接与控制

§ 5.5 提高主存读写速度的技术

§ 5.6 并行存储器和相联存储器

§ 5.7 高速缓冲存储器

§ 5.8 虚拟存储器

§ 5.9 磁表面存储原理和光记录原理



第5章 存储系统和结构

本章要点:

1. 存储器的分类方法和存储系统的层次, 半导体随机存储器（静态RAM和动态RAM）、各种类型ROM的基本存储原理, Cache存储系统和虚拟存储器的概念;
2. 主存储器的基本结构、存储单元和主存储器的主要技术指标, 动态RAM的刷新, RAM芯片的基本结构;
3. 数据在主存中的存放方法; 主存储器的工作原理、容量的各种扩展方法; 主存储器和CPU的连接。

第5章 存储系统和结构

存储器用途：存放程序和数据。

要求容量大、速度高、成本低。

（但在同样技术条件下三者往往相互矛盾）

存储系统：由几个容量、速度和价格各不相同的存储器构成的系统。（以提高整体性能）

§ 5.1 存储系统的组成

5.1.1 存储器的分类

1. 按存储介质分类

- (1) 半导体存储器：TTL、MOS
- (2) 磁表面存储器：磁盘、磁带
- (3) 磁芯存储器：已被半导体存储器取代
- (4) 光存储器：光盘等



§ 5.1 存储系统的组成

5.1.1 存储器的分类

2. 按存储方式分类

(1) 存取时间与物理地址无关（随机访问）

随机存储器RAM：在程序的执行过程中可读可写

只读存储器ROM：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）

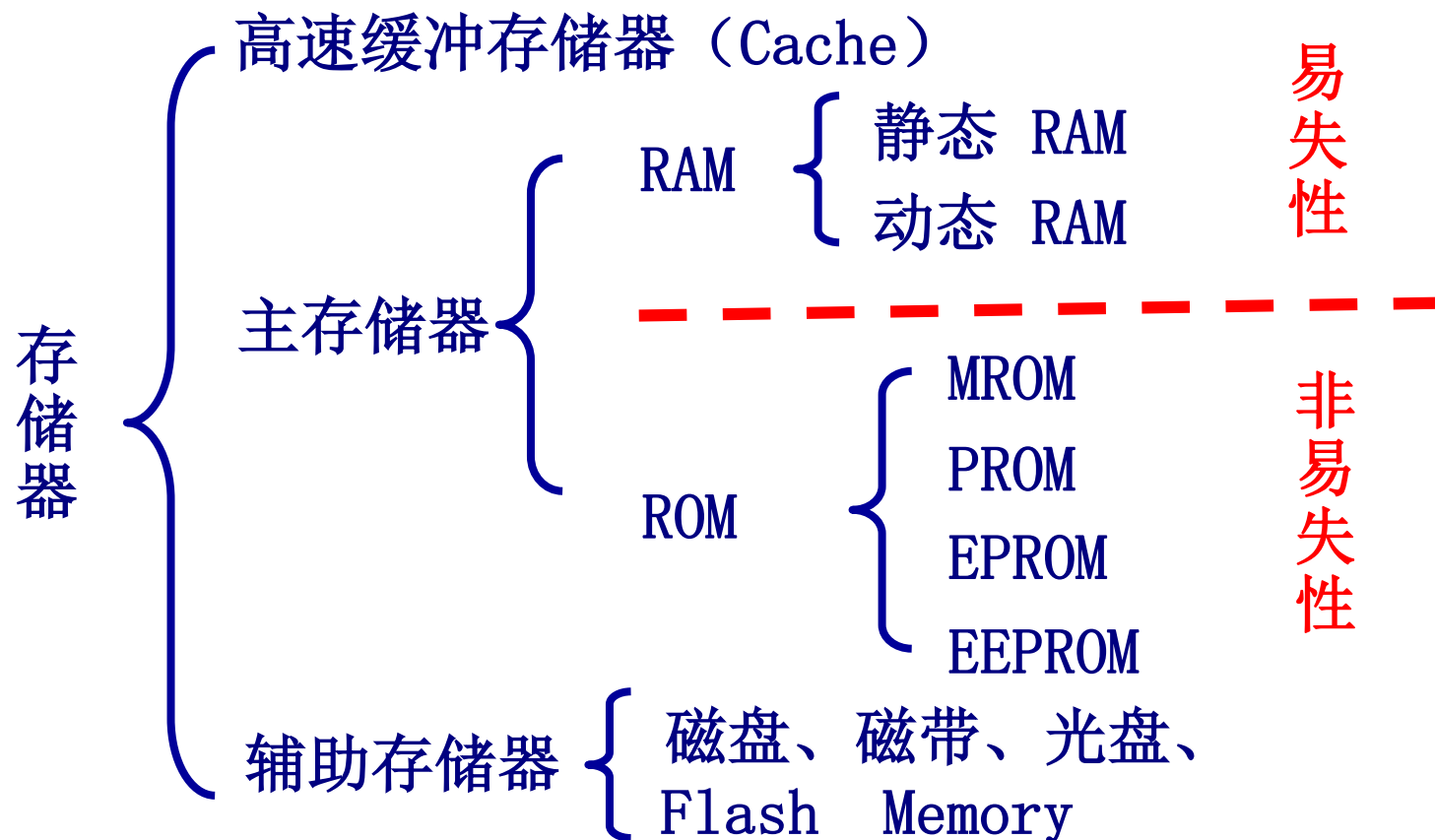
顺序存取存储器：磁带

直接存取存储器：磁盘

§ 5.1 存储系统的组成

5.1.1 存储器的分类

3. 按在计算机中的作用分类



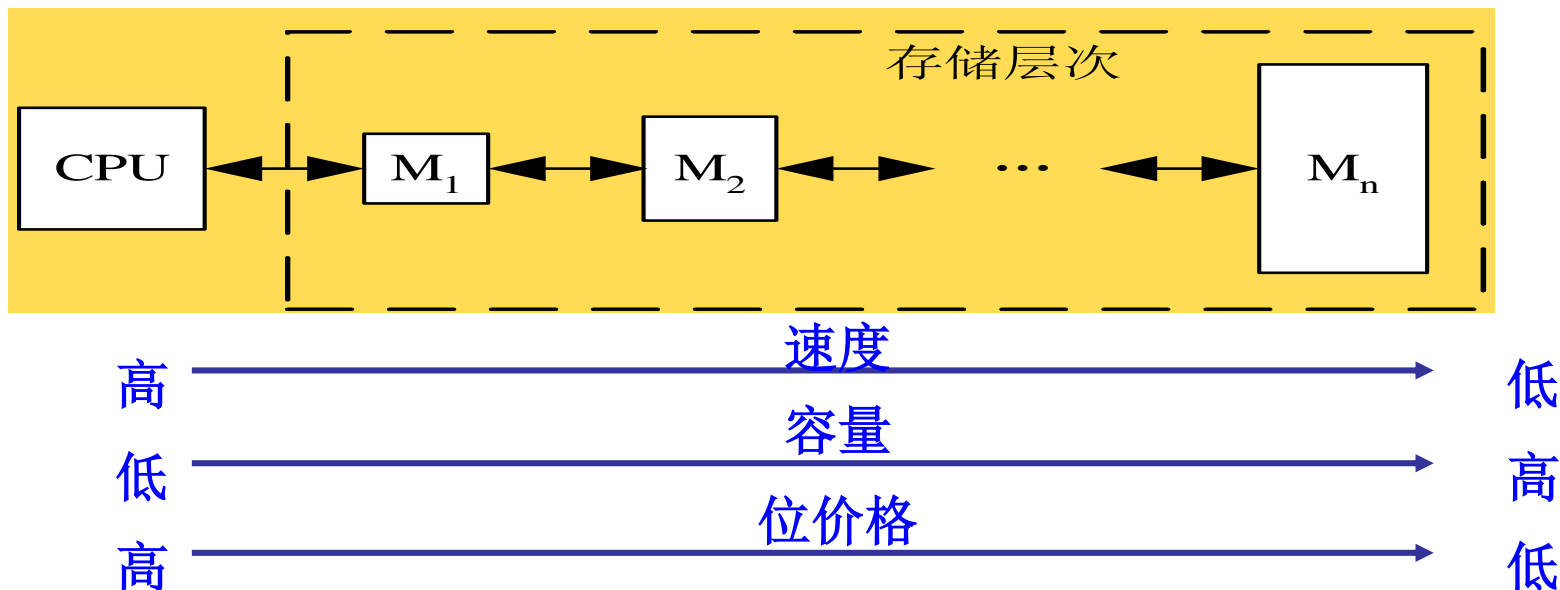
§ 5.1 存储系统的组成

5.1.2 存储器的层次结构

目的：解决存储容量、存取速度和价格之间的矛盾。

方法：采用多级存储层次，以提高存储系统的整体性能。

1. 多级存储层次

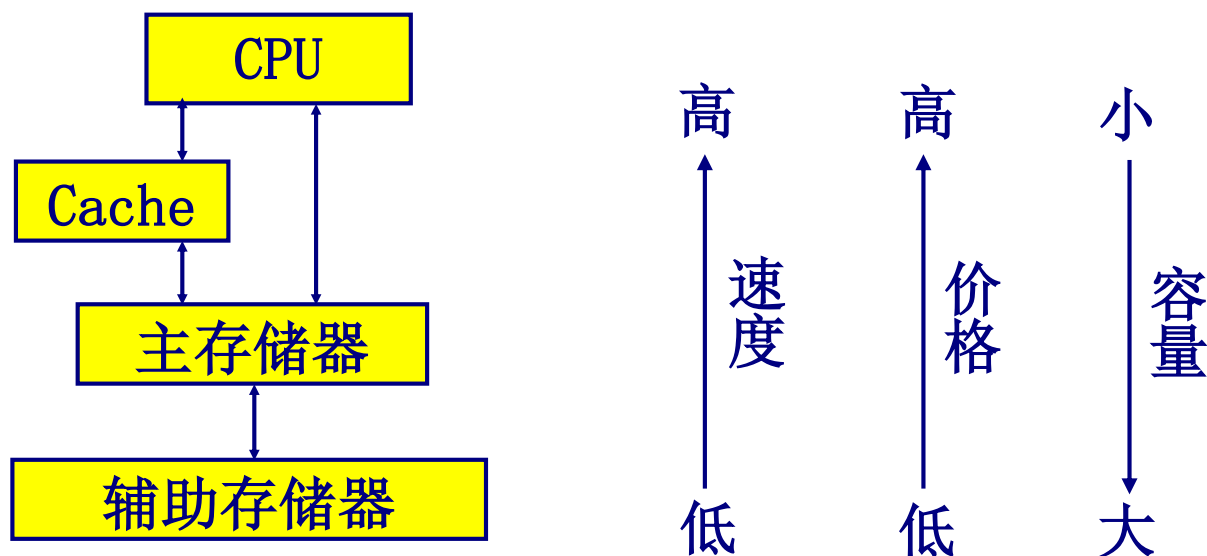


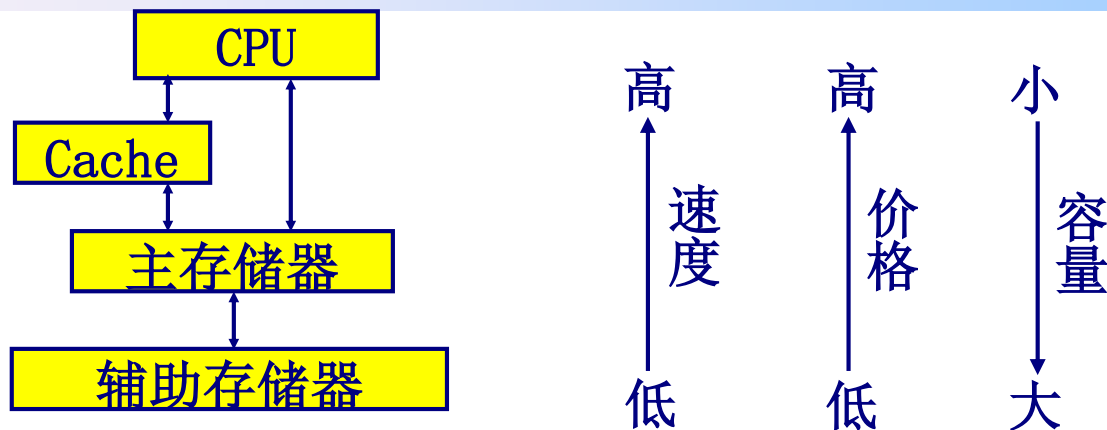
§ 5.1 存储系统的组成

5.1.2 存储器的层次结构

1. 多级存储层次

最典型的三级物理存储体系：“Cache—主存—外存”





(1) Cache

存放少量内存数据的副本，速度很快，可与CPU速度匹配。
(与主存之间有数据映射算法、淘汰算法)

(2) 主存

能由CPU直接编程访问。运行的程序及数据要放在主存中。

(3) 辅存

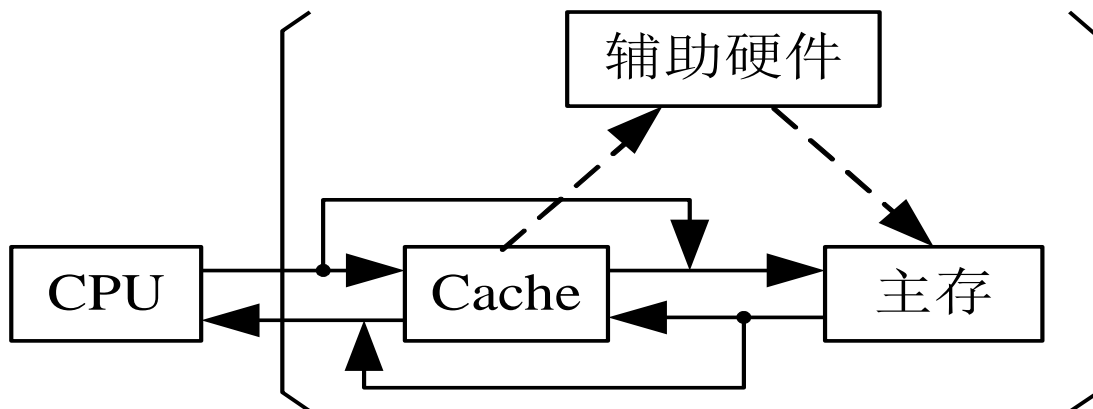
存放需联机保存但暂不使用的程序与数据。

当要运行其中的程序时，先把传到内存再运行。

5.1.2 存储器的层次结构

2. Cache—主存存储层次（Cache存储系统）

Cache存储系统是为解决主存速度不足而提出来的。从CPU看，速度接近Cache的速度，容量是主存的容量，每位价格接近于主存的价格。

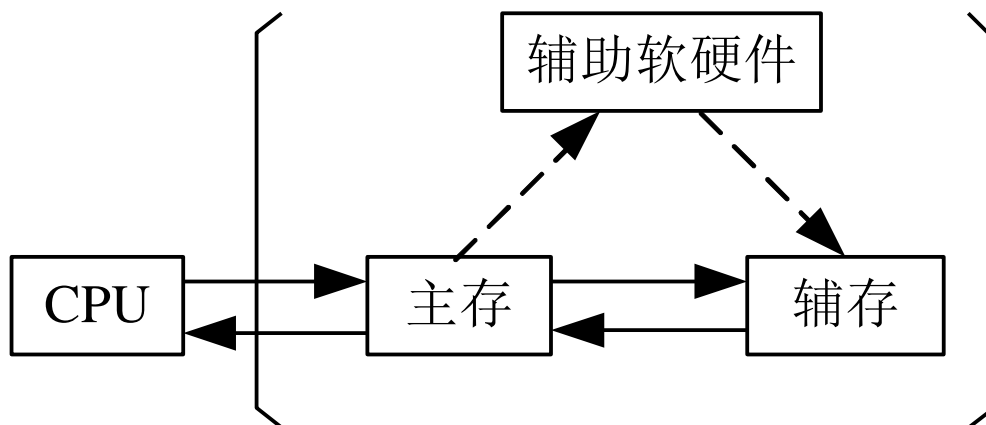


由于Cache存储系统全部用硬件来调度，因此它对系统程序员和系统程序员都是透明的。

5.1.2 存储器的层次结构

3. 主存-辅存存储层次（虚拟存储系统）

虚拟存储系统是为**解决主存容量不足**而提出来的。从CPU看，速度接近主存的速度，容量是虚拟的地址空间，每位价格是接近于辅存的价格。



由于虚拟存储系统需要通过操作系统来调度，因此对系统程序员是不透明的，但对应用程序员是透明的。

5.1.2 存储器的层次结构

3. 主存-辅存存储层次（虚拟存储系统）

虚拟内存（虚拟地址空间）：

用户可使用的内存的逻辑地址空间（地址的范围）。

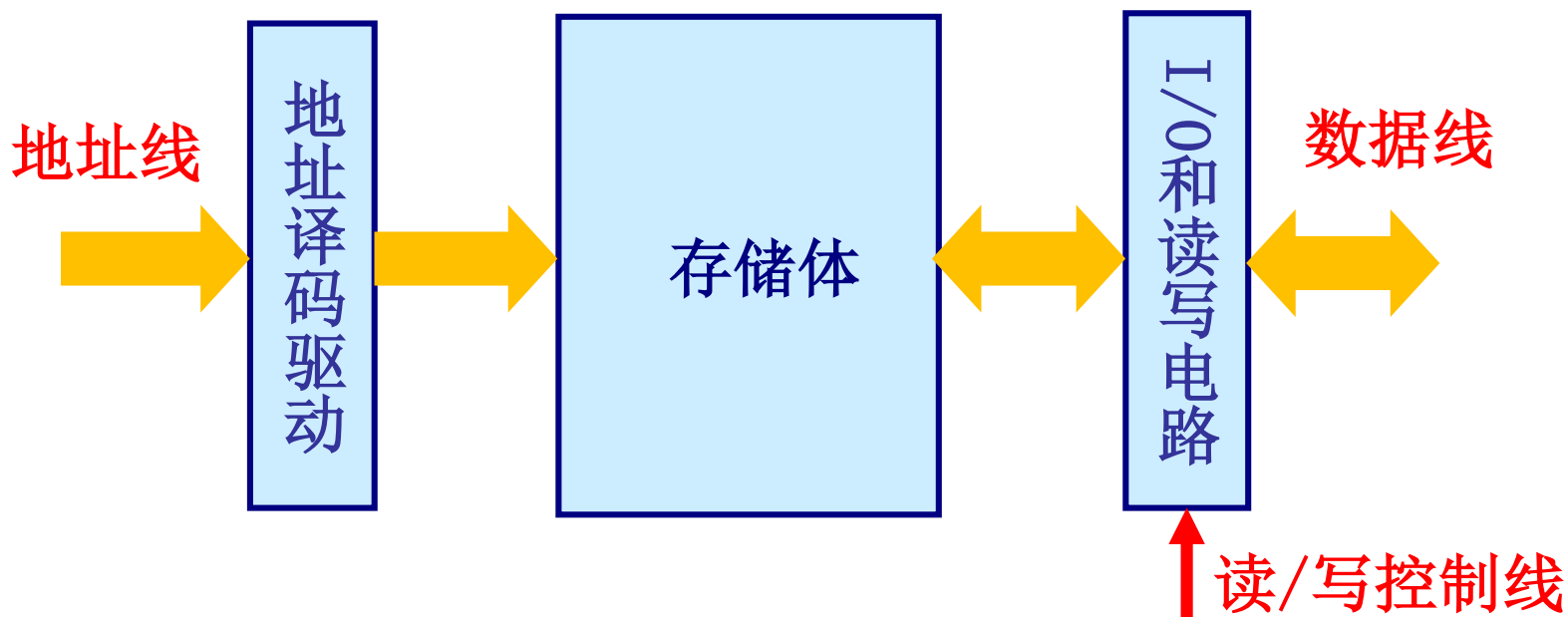
被执行的代码和被访问的数据必须赋予物理内存（映射到物理内存）。

在Windows XP系统中，Pagefile.sys文件就是用作虚拟内存的磁盘文件。

§ 5.2 主存储器的组织

5.2.1 主存储器的基本结构

主存通常由存储体、地址译码驱动电路、I/O和读写电路组成。



5.2.2 主存储器的存储单元

1. 基本概念

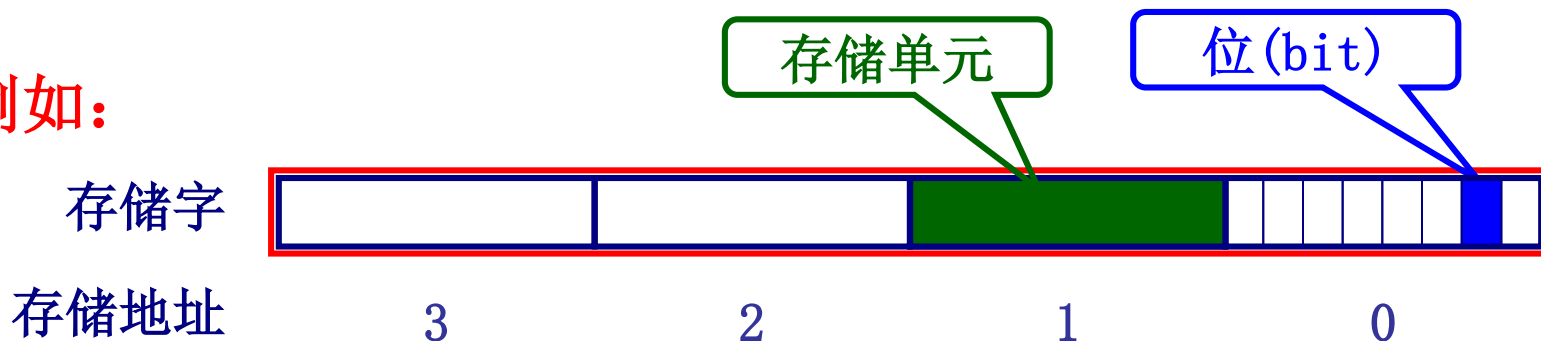
位(bit): 是存储器存储信息的最小单位。

存储字: 可作为一个整体存入或取出的二进制位数。

存储单元: 存放存储字/字节的主存空间,
是CPU对主存可访问操作的最小单位。

存储地址: 每个存储单元的编号称为（存储）地址。

例如:



5.2.2 主存储器的存储单元

2. 地址编排方案

内存的常见编址单位有：

- 1) 按字编址：编址单位=计算机字长
- 2) 按字节编址：编址单位=1个字节
- 3) 按位编址：编址单位=1bit

5.2.2 主存储器的存储单元

3. 按字节编址时的字地址

1) 大端方案

字地址等于最高有效字节地址。（高字节在前）

例如：IBM 370机，字长32位，按字节编址，字地址总是等于4的整数倍，用地址码的最末两位来区分同一个字的4个字节。

字地址	字节地址			
	0(A3)	1(A2)	2(A1)	3(A0)
0	4	5	6	7
4	8	9	10	11

5.2.2 主存储器的存储单元

3. 按字节编址时的字地址

2) 小端方案

字地址等于最低有效字节地址。（低字节在前）

例如：PDP-11机，字长为16位，按字节编址，字地址等于2的整数倍。用地址码的最末1位来区分同一个字的两个字节。

字地址	字节地址	
0	0(A0)	1(A1)
2	2	3
4	4	5

5.2.3 数据在主存中的存放

按字节编址时，数据在主存中有3种不同存放方法：

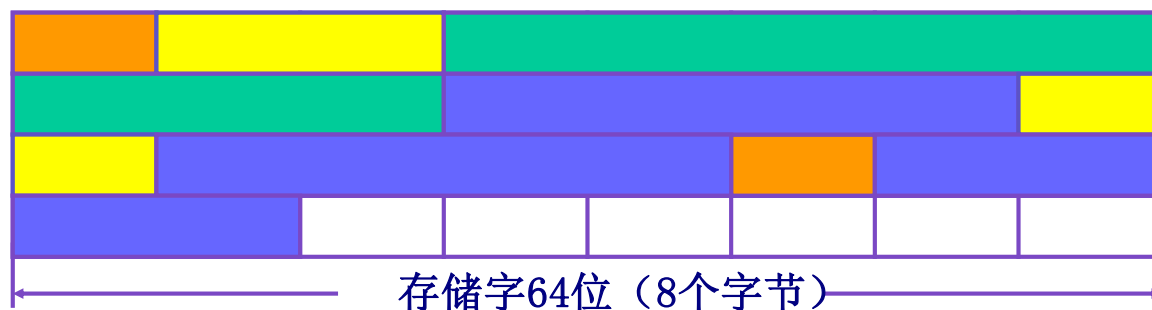
- 1) 紧缩存放
- 2) 存储字字边界存放
- 3) 整数边界存放

5.2.3 数据在主存中的存放

【例】设存储字长为64位，读写的数据有以下4种不同长度



1) 紧缩存放 4种不同长度的数据一个紧接着一个存放。



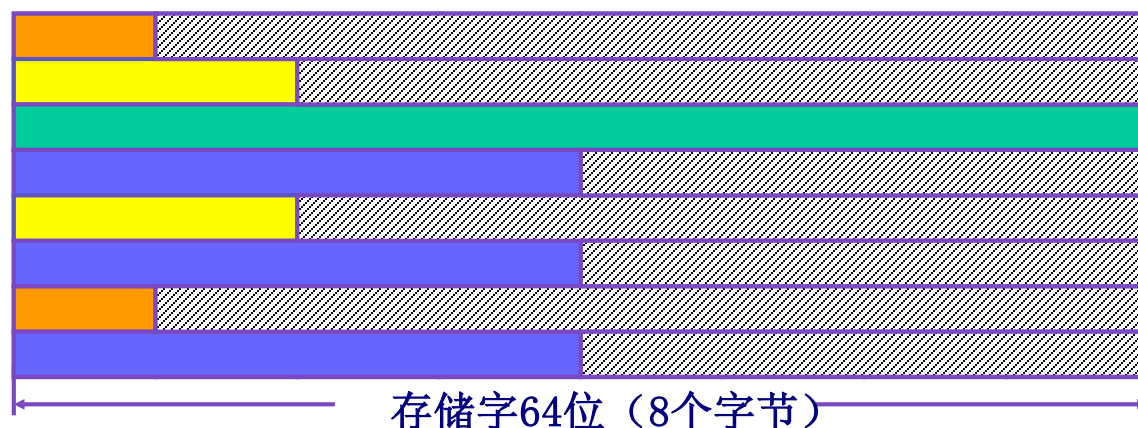
特点：不浪费主存资源，但当一个数据跨一个存储字存放时，要增加访存次数，且读写控制复杂。

5.2.3 数据在主存中的存放

【例】设存储字长为64位，读写的数据有以下4种不同长度



2) 从存储字的起始位置开始存放



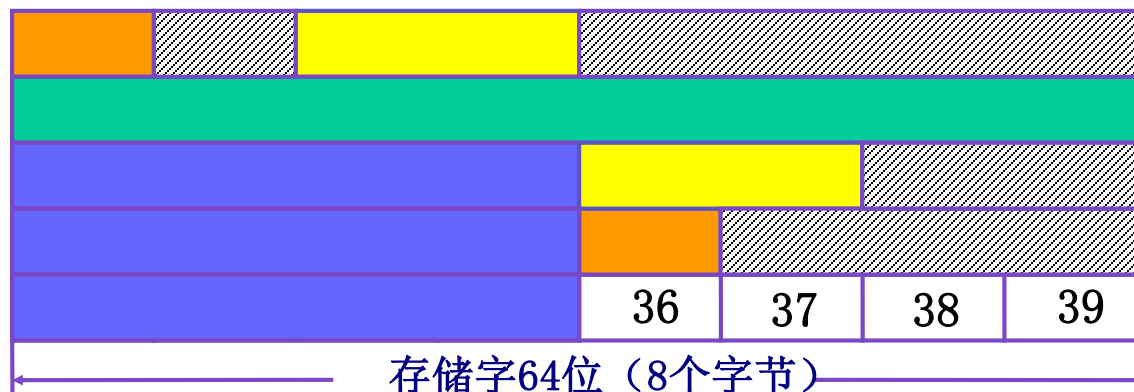
特点：读写速度快、控制比较简单；浪费了存储器资源。

5.2.3 数据在主存中的存放

【例】设存储字长为64位，读写的数据有以下4种不同长度



3) 整数边界存放：单字节数据的地址可任意，
 二字节数据的地址其最末一位只能是 0_2
 四字节数据的地址其最末两位只能是 00_2



5.2.4 存储器的主要技术指标

1. 存储容量

以字节编址的计算机，通常以字节为单位。如：2GB

以字编址的计算机，以字数与其字长的乘积表示容量。

例：512K×16 （容量等于1024 KB）



注意：通常情况下，1MB代表1024KB，即 2^{10} KB或 2^{20} Byte。

但在表述硬盘、U盘的存储容量时，目前习惯上1MB指1000KB。

5.2.4 存储器的主要技术指标

2. 存取速度

(1) 存取时间 T_a （访问时间或读写时间）

从启动一次存储器操作到完成该操作所经历的时间。
 T_a 越小，存取速度越快。

(2) 存取周期 T_m （访问周期或读写周期）

存储器进行一次完整的读写操作所需的全部时间。即两次连续访问存储器(读或写)操作之间所需的最小时间间隔。
一般 $T_m > T_a$ 。

(3) 主存带宽 B_m （数据传输率）

每秒从主存进出信息的最大数量。位/秒。

5.2.4 存储器的主要技术指标

1. 存储容量

2. 存取速度

3. 可靠性

指在规定时间内，存储器无故障的概率。

4. 功耗

反映存储器耗电的多少和发热的程度。

5. 性价比

是衡量存储器经济性能好坏的综合性指标。

§ 5.3 半导体随机存储器和只读存储器

半导体存储器RAM { 双极型存储器(静态)
MOS存储器 { 静态存储器SRAM
动态存储器DRAM

记忆单元： 存放一个二进制位的物理器件。
(是存储器的最基本构件)

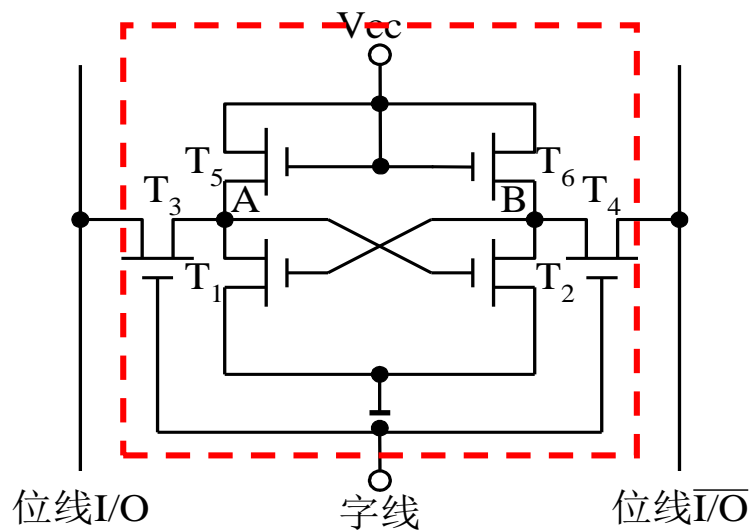
地址码相同的多个记忆单元构成一个**存储单元**。

§ 5.3 半导体随机存储器和只读存储器

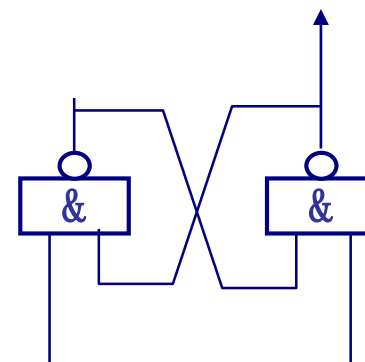
5.3.1 RAM记忆单元电路

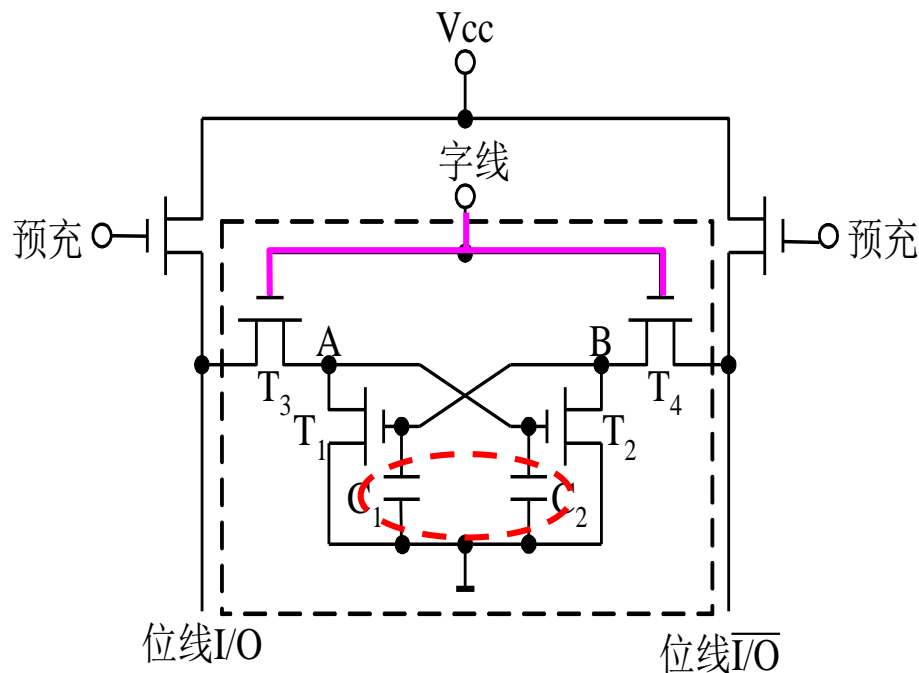
1. 6管SRAM记忆单元电路

用双稳态触发器来存储一位二进制信息0或1。



对照：





2. 4管DRAM记忆单元电路

靠MOS电路中的栅极电容C1、C2来存储信息的。

② 写入操作

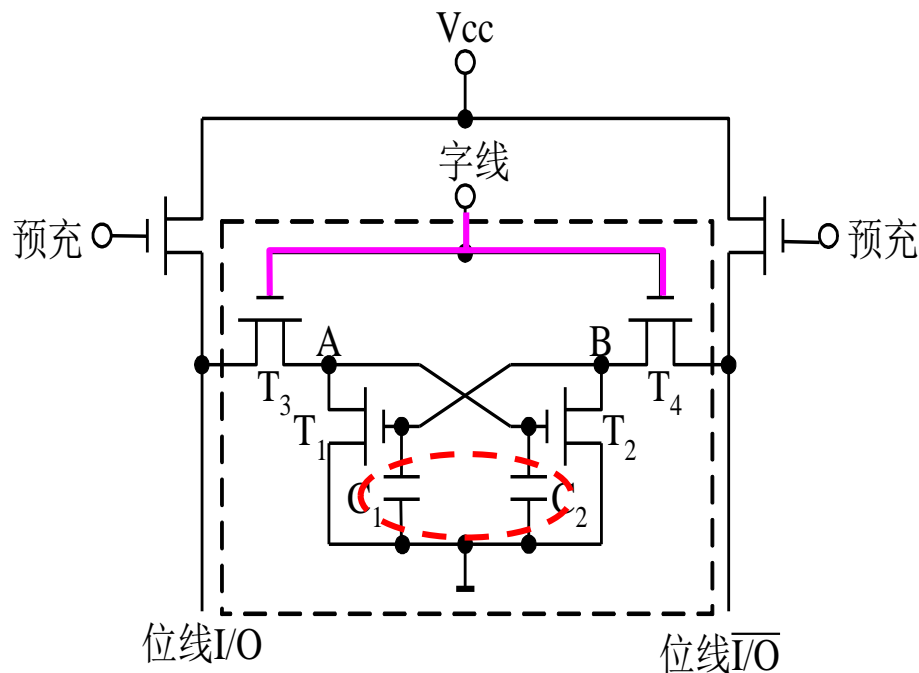
字线为高；

写“1”时, I/O为高电位,

$\overline{I/O}$ 为低电位;

写“0”时, $\overline{I/O}$ 为高电位,

I/O为低电位。



3. 单管DRAM记忆单元

用电容C存储电荷来表示信息。

① 保存信息：字线为低

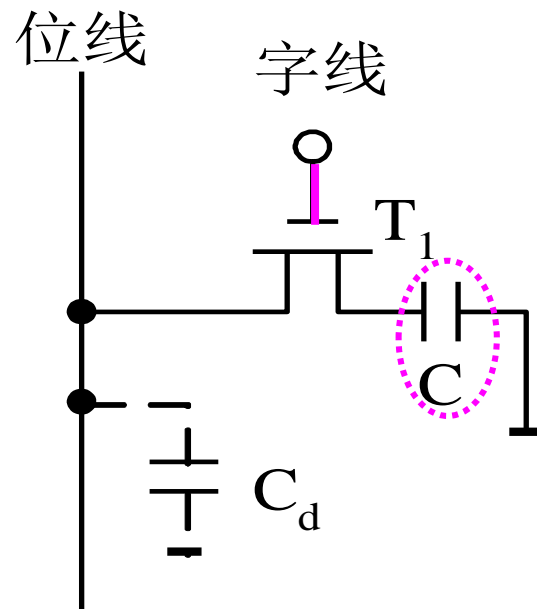
由于存在漏电流，需周期性给电容补充电荷，即刷新。

② 写入操作：字线为高电平

当位线为高电平时，C被充电，写入“1”；当位线为低电平，C被放电，写入“0”

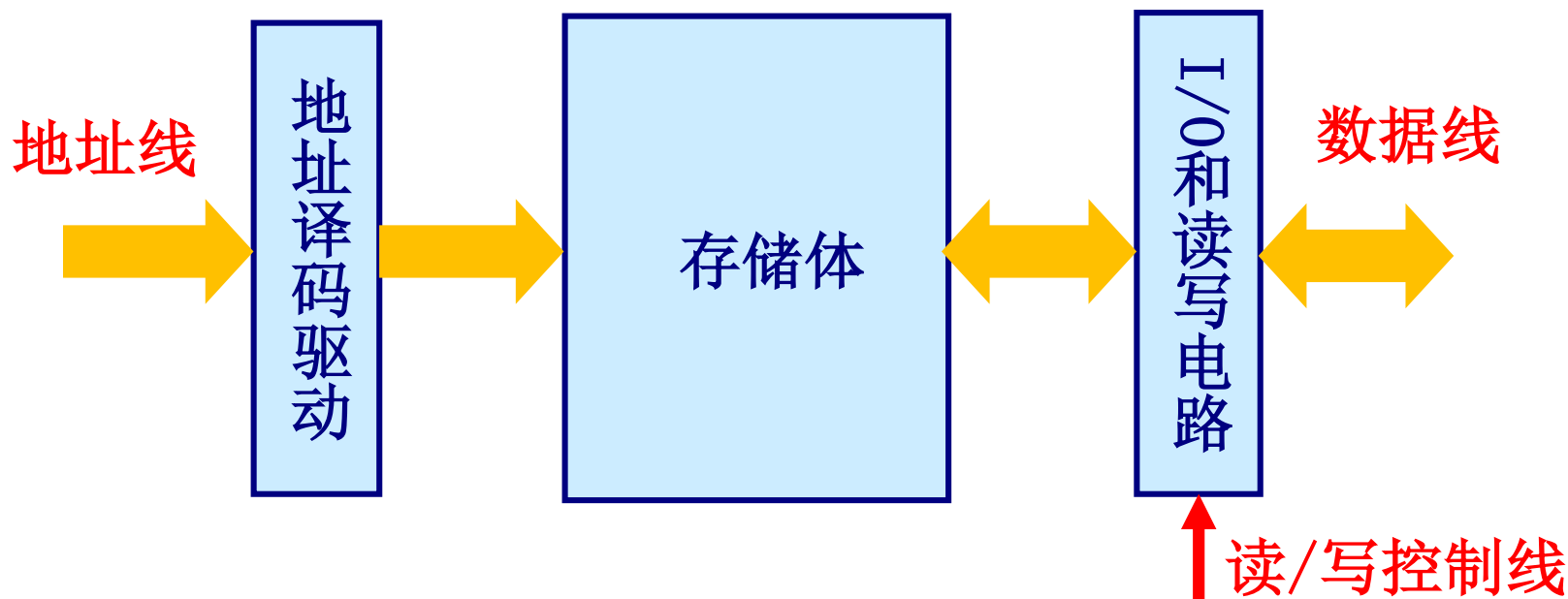
③ 读出操作：字线为高电平（T1管导通）

若存储的是“1”，C上有电荷，位线上产生读电流；
若存储的是“0”，C上无电荷，位线上不产生读电流。



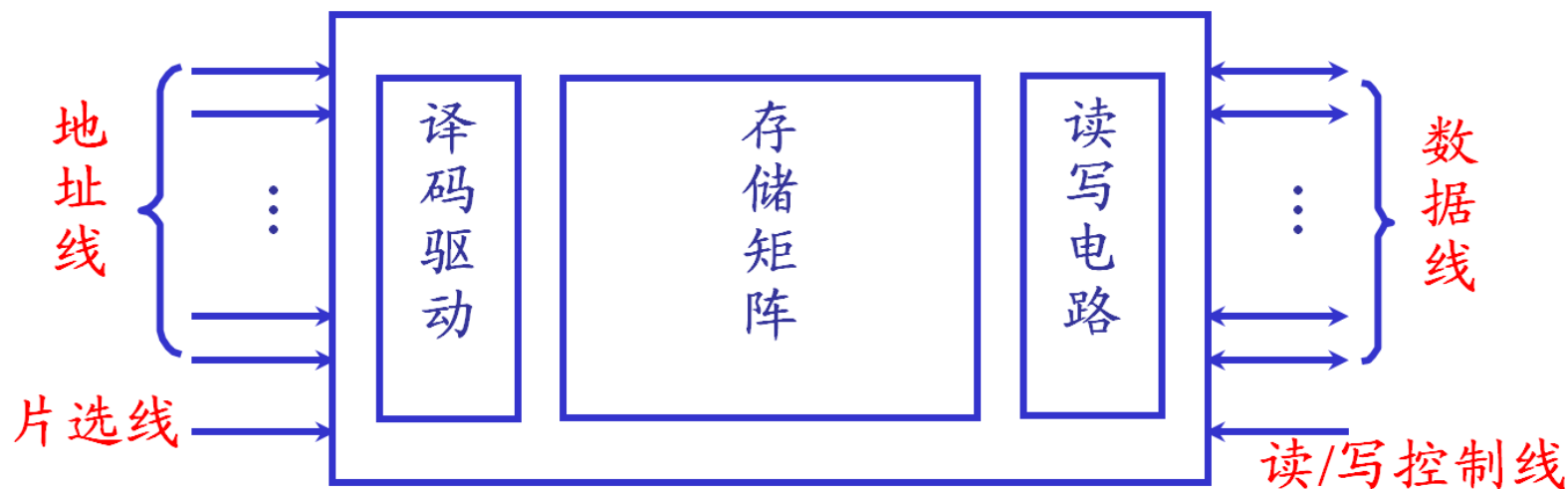
§ 5.3 半导体随机存储器和只读存储器

5.3.2 RAM芯片分析



5.3.2 RAM芯片分析

1. RAM芯片



地址线(单向)

10

14

13

数据线(双向)

4

1

8

芯片容量

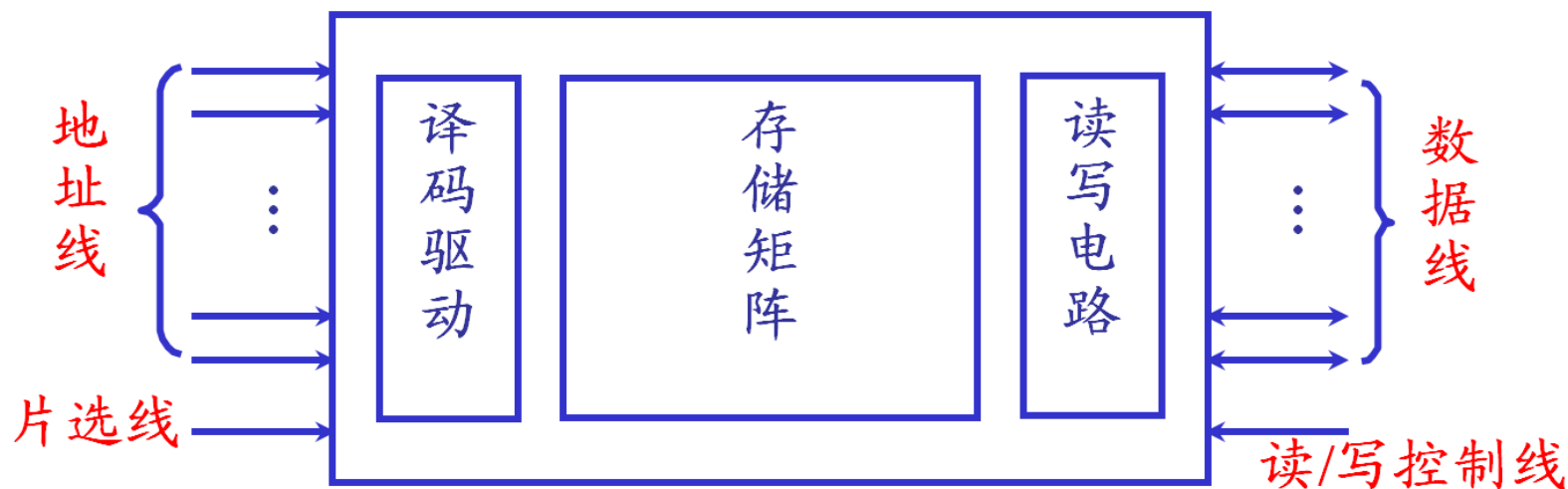
1K×4位

16K×1位

8K×8位

5.3.2 RAM芯片分析

1. RAM芯片



片选线: \overline{CS} 或 \overline{CE} 或 \overline{RAS} 和 \overline{CAS}

读/写控制线: \overline{WE} (低电平写、高电平读)
或 \overline{WR} (写)和 \overline{RD} (读)
 \overline{OE} (允许数据输出)

5.3.2 RAM芯片分析

2. 地址译码方式

(1) 单译码方式(一维地址译码方式，字选法)

只有一个地址译码器，译码器的输出为字选通线，选择某个字的所有位。

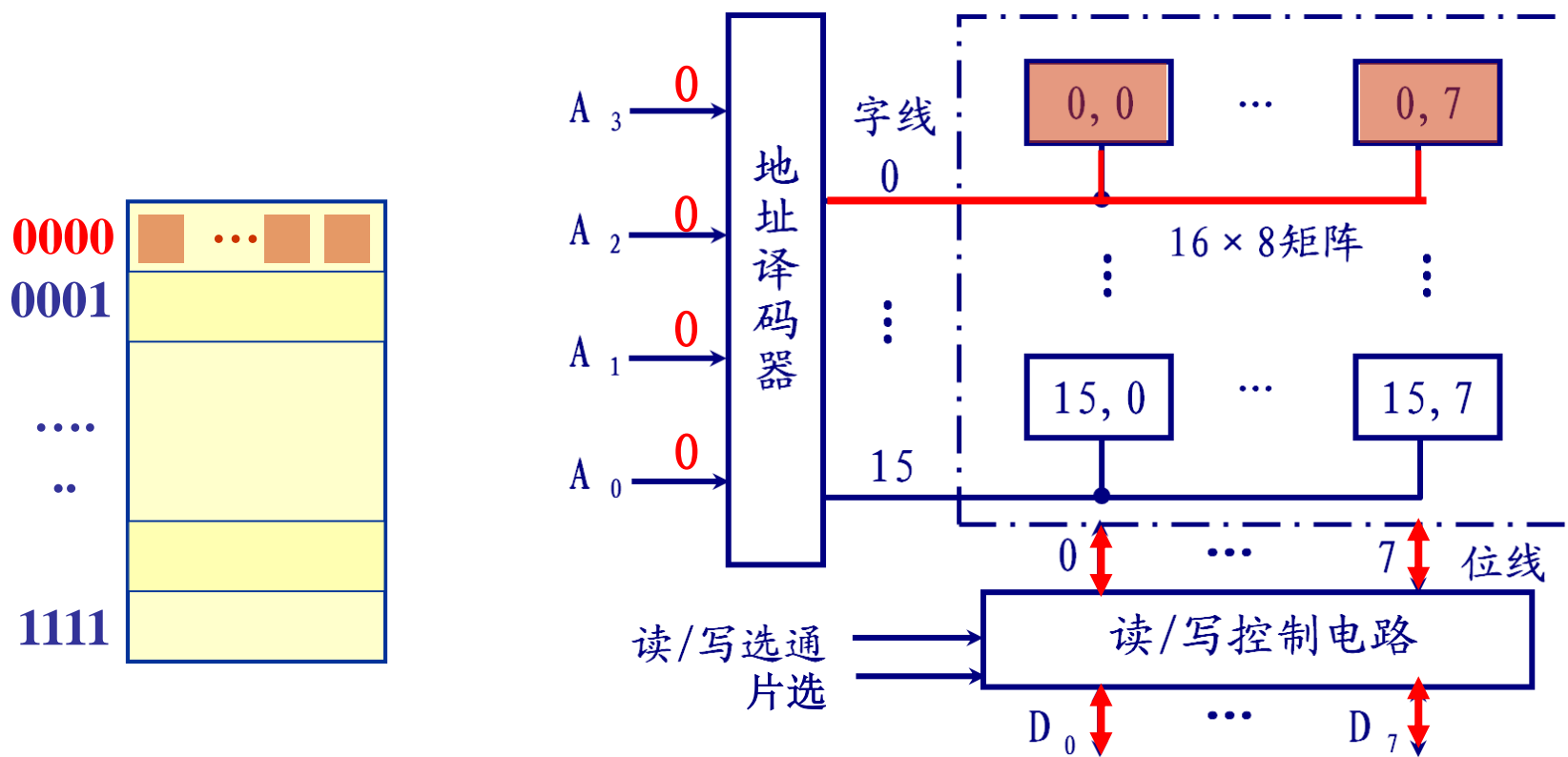
设容量为 $W \times b$ 的存储器（即有 W 个字，每个字为 b 位），排列成 W 行 $\times b$ 列的矩阵。

特点：结构简单，但译码器的输出线太多，集成度低。

5.3.2 RAM芯片分析

2. 地址译码方式

(1) 单译码方式(一维地址译码方式, 字选法)



5.3.2 RAM芯片分析

2. 地址译码方式

(2) 双译码方式(二维地址译码方式、重合法)

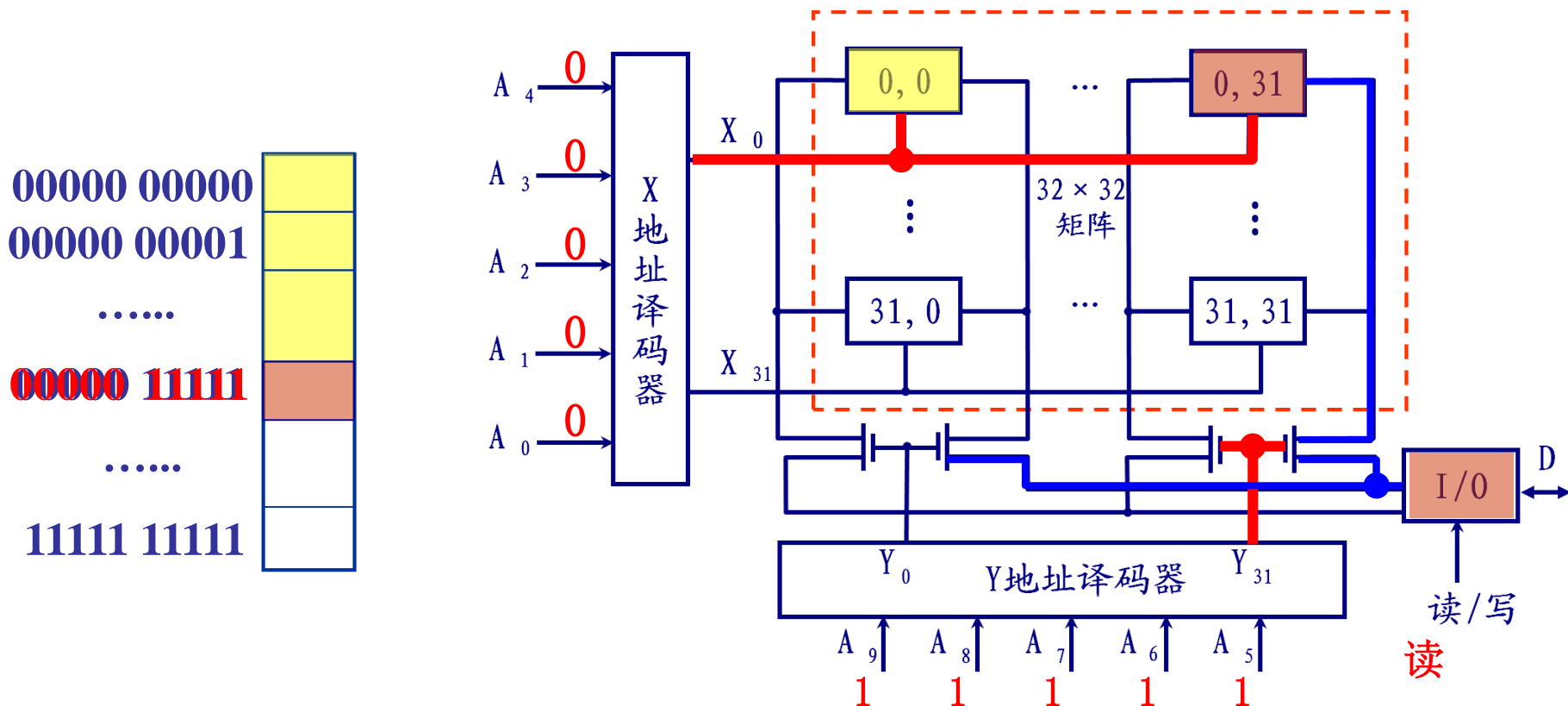
采用位结构的双译码方式，把 n 位地址信号按行和列分为相等的两部分，分别产生字选通信号（水平方向）和位选通信号（垂直方向），二者的交叉点就是被选中的单元。

特点：地址译码器结构简单，连线少；
每个存储芯片只能提供存储单元的一位。

5.3.2 RAM芯片分析

2. 地址译码方式

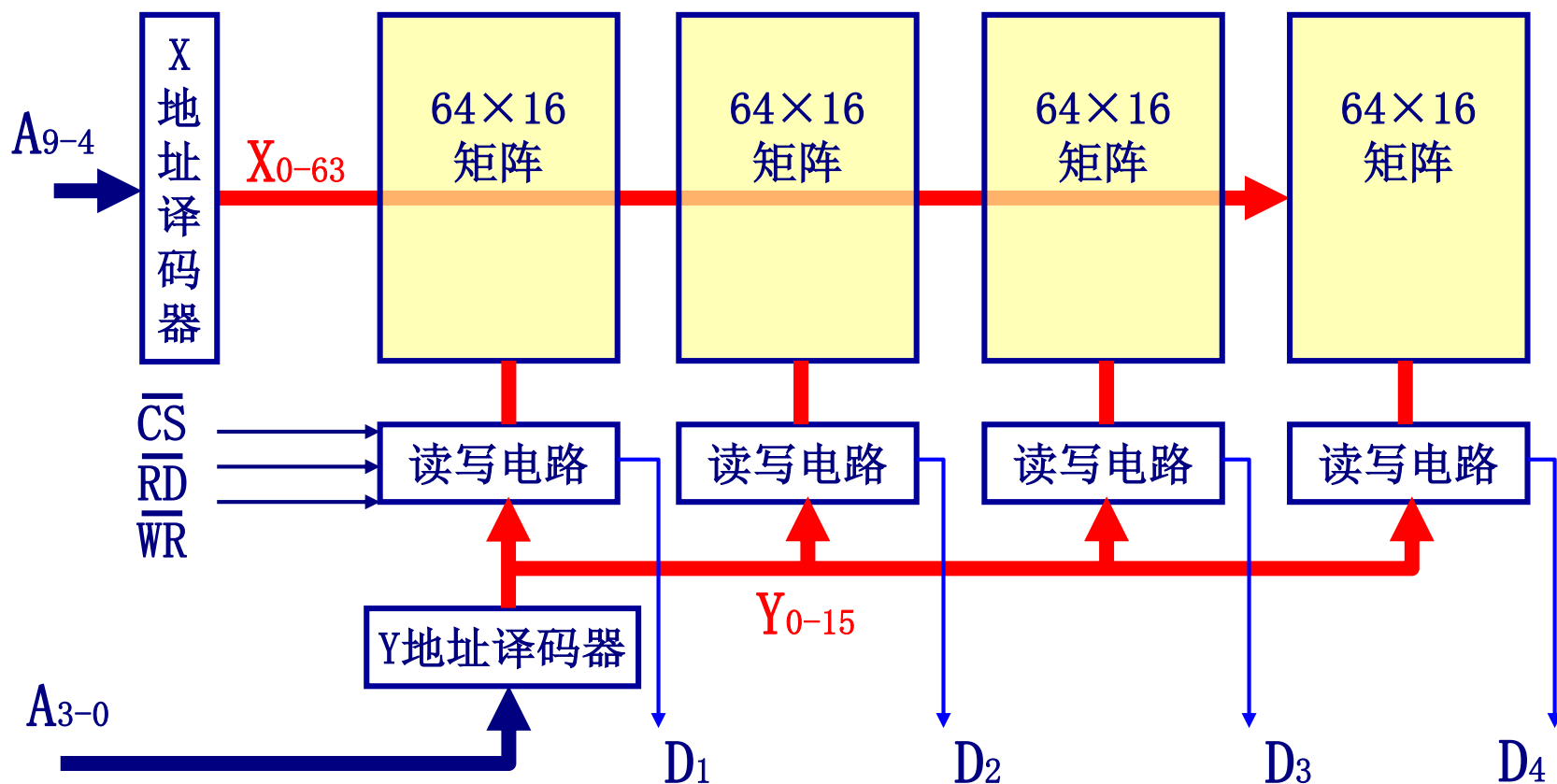
(2) 双译码方式(二维地址译码方式、重合法)



5.3.2 RAM芯片分析

2. 地址译码方式

(3) 字段结构双译码方式（以1K×4的SRAM为例）

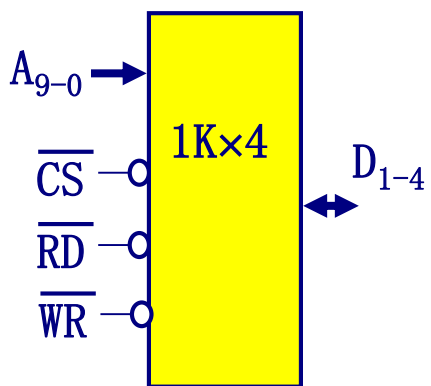


5.3.2 RAM芯片分析

2. 地址译码方式

(3) 字段结构双译码方式（以1K×4的SRAM为例）

有 1K 个单元地址，每个单元可同时读/写 4 位。



地址线： 10 条， A_9-A_0 ($2^{10}=1K$)

数据线： 4 条双向， D_{1-4}

片选信号： \overline{CS} ，低电平时芯片才能读写

读写控制： \overline{RD} 为低时读出； \overline{WR} 为低时写入

(有的芯片用 \overline{WE} 控制读写 (0-写，1-读))

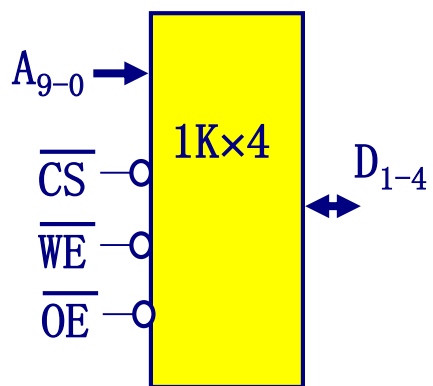
芯片内部通常采用二维地址译码。

5.3.2 RAM芯片分析

2. 地址译码方式

(3) 字段结构双译码方式（以1K×4的SRAM为例）

有 1K 个单元地址，每个单元可同时读/写 4 位。



地址线： 10 条， A_9-A_0 ($2^{10}=1K$)

数据线： 4 条双向， D_{1-4}

片选信号： \overline{CS} ，低电平时芯片才能读写

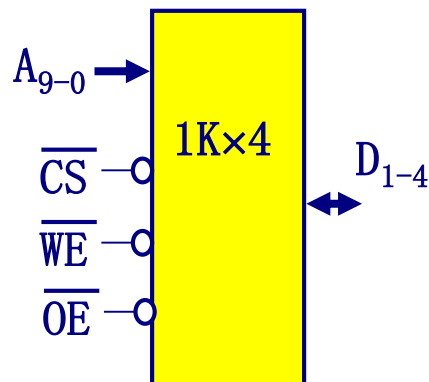
读写控制： \overline{WE} ，0-写，1-读

输出控制： \overline{OE} ，0-可输出，1-浮空

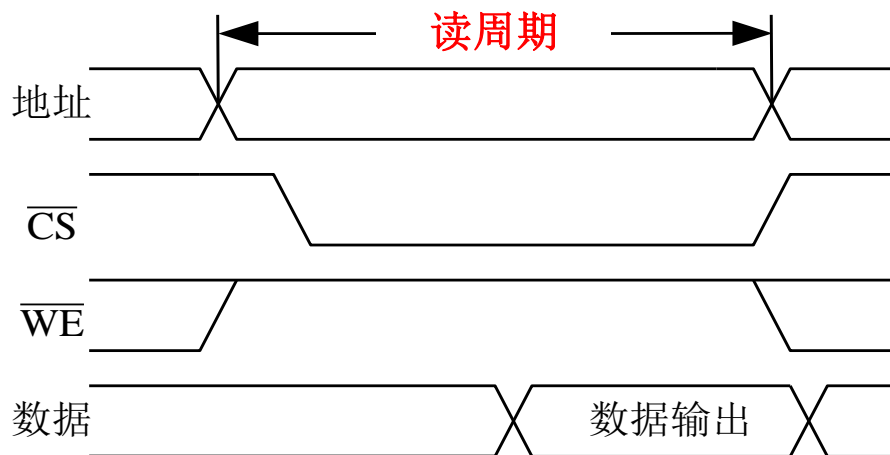
芯片内部通常采用二维地址译码。

5.3.2 RAM芯片分析

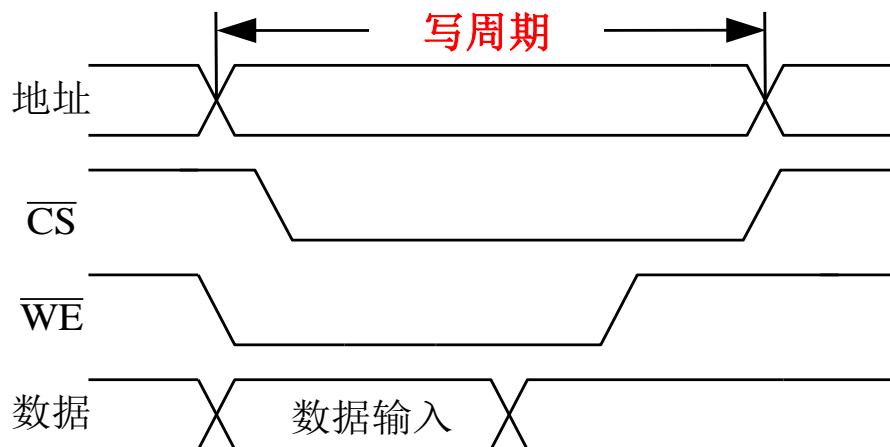
3. 静态存储器（SRAM）的读写时序



读:



写:



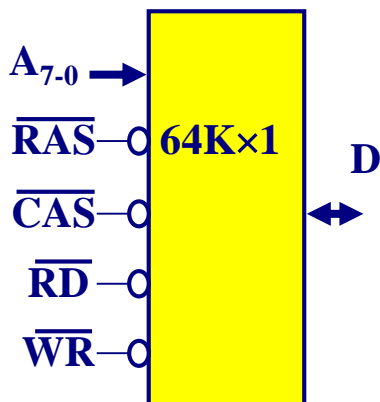
5.3.2 RAM芯片分析

4. 动态存储器（DRAM）芯片及其读写时序

以 $64\text{K} \times 1$ 的芯片为例（需 16 位地址 ($2^{16}=64\text{K}$)）

采用二维地址译码

用 8 根地址线（分时复用）：



$\overline{\text{RAS}}$ 下降沿时从地址线送入行地址 A_{15-8}

$\overline{\text{CAS}}$ 下降沿时从地址线送入列地址 A_{7-0}

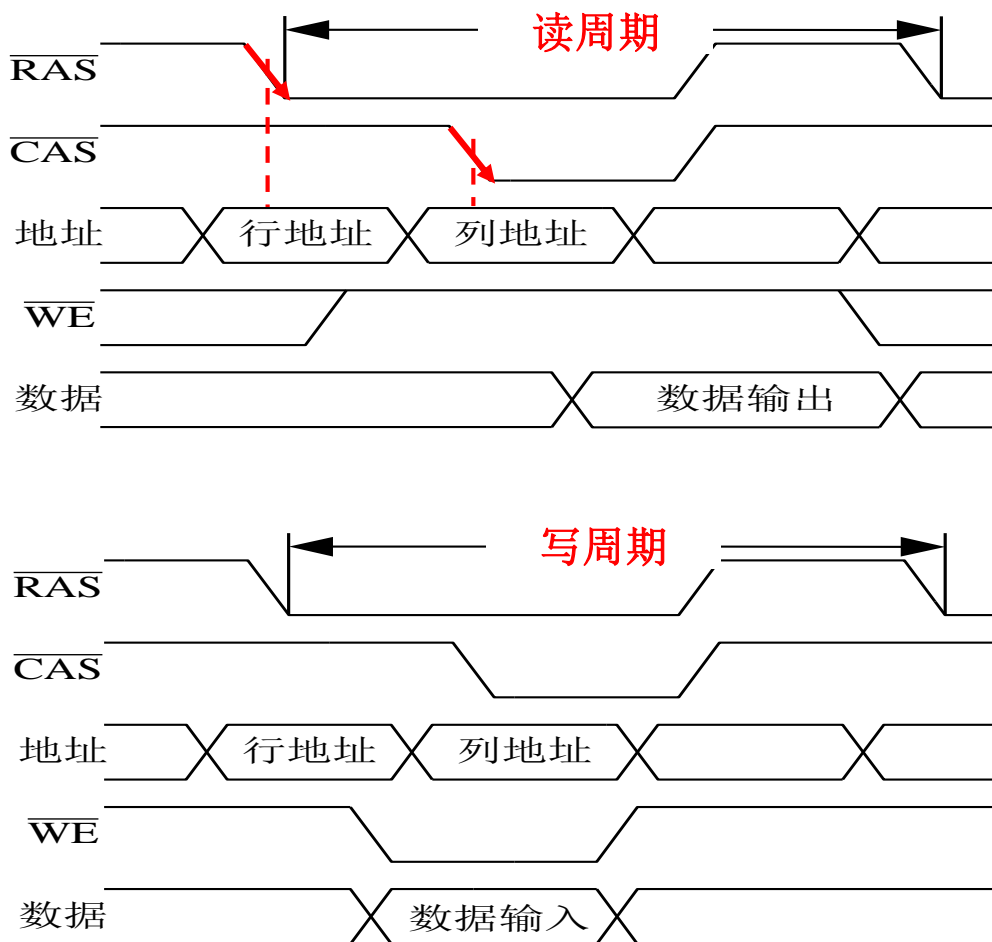
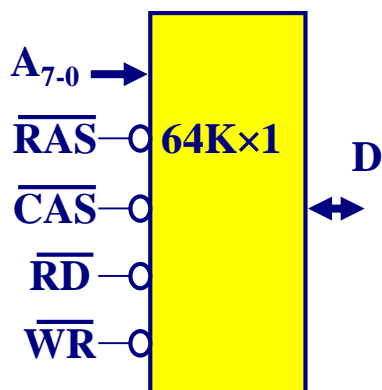
数据线：1 条双向线

（有的芯片用 2 条单向线 D_{in} , D_{out} ）

读写控制： $\overline{\text{RD}}$ 为低时读出； $\overline{\text{WR}}$ 为低时写入
（有的芯片用 $\overline{\text{WE}}$ 控制读写（0-写，1-读））

5.3.2 RAM芯片分析

4. 动态存储器（DRAM）芯片及其读写时序



5.3.3 动态存储器的刷新

刷新： DRAM是用电容存储信息，经过一定时间后电容上的电荷可能被泄放掉，因此每隔一定时间必须向存有电荷的电容补充一次电荷，称为“刷新”。

1. 刷新闻隔

DRAM记忆单元中电容信息可保持的时间决定了两次刷新操作的时间间隔，在这段时间内必须将所有存储单元都刷新一遍。目前一般芯片的最大刷新闻隔为2mS。

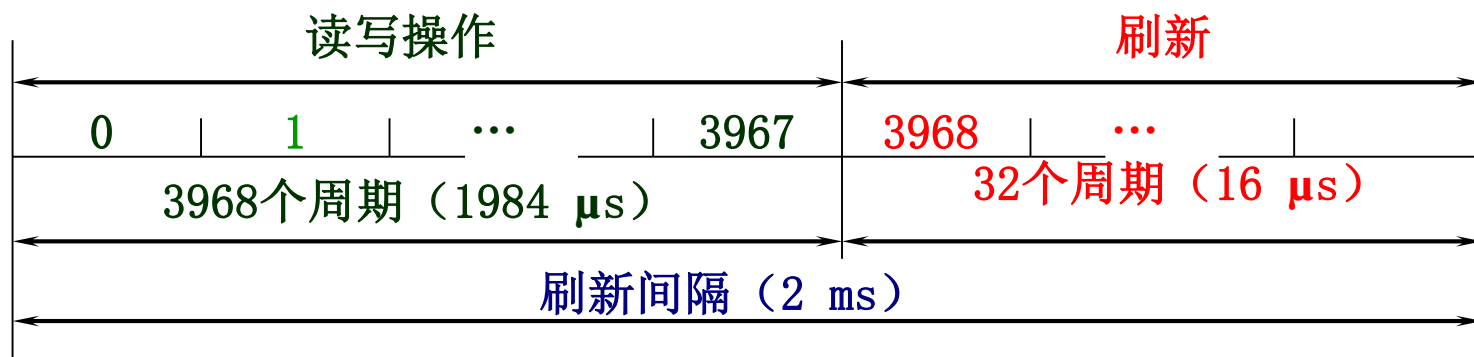
2. 刷新方式

(1) 集中刷新方式

在允许的最大刷新间隔内，按照存储芯片容量的大小集中安排若干个刷新周期，刷新时停止读写操作。

刷新时间=存储矩阵行数×刷新周期

注：这里刷新周期是指刷新一行所需要的时间，由于刷新过程就是“假读”的过程，所以刷新周期等于存取周期。

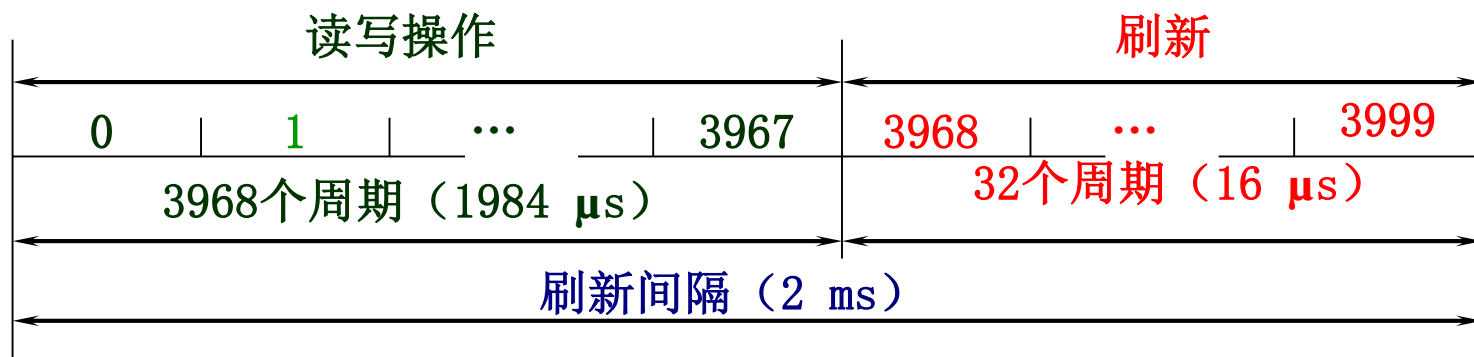


2. 刷新方式

(1) 集中刷新方式

在允许的最大刷新间隔内，按照存储芯片容量的大小集中安排若干个刷新周期，刷新时停止读写操作。

例：设存储器芯片的存取周期为 500ns ($0.5\mu\text{s}$)，芯片有 1024 个存储单元，排列成 32×32 的存储矩阵。刷新方式如下。



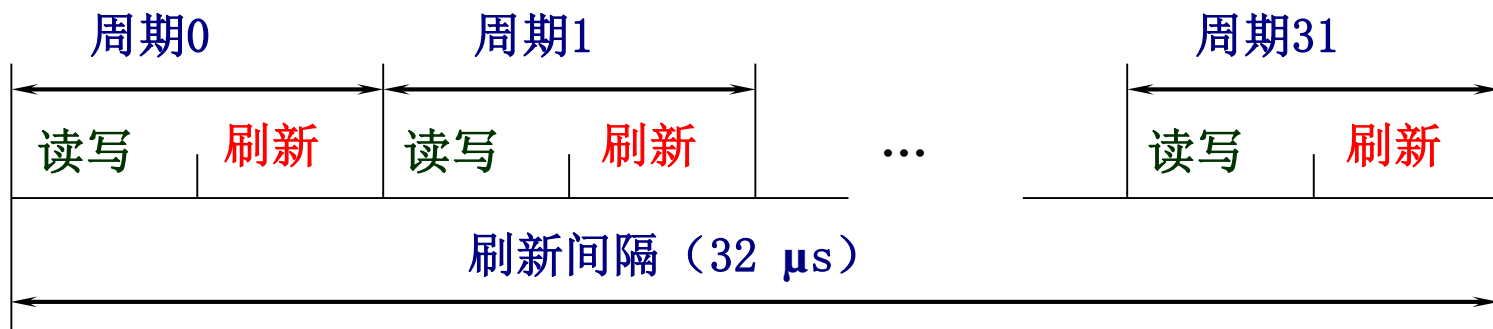
特点：读写速度快；

在读写时间存在死区，且容量越大死区越长。

2. 刷新方式

(2) 分散刷新方式

把刷新操作分散到每个存取周期内进行，在一个系统存取周期内刷新存储矩阵中的一行。



特点：没有死区；

加长了系统存取周期，降低了整机速度；

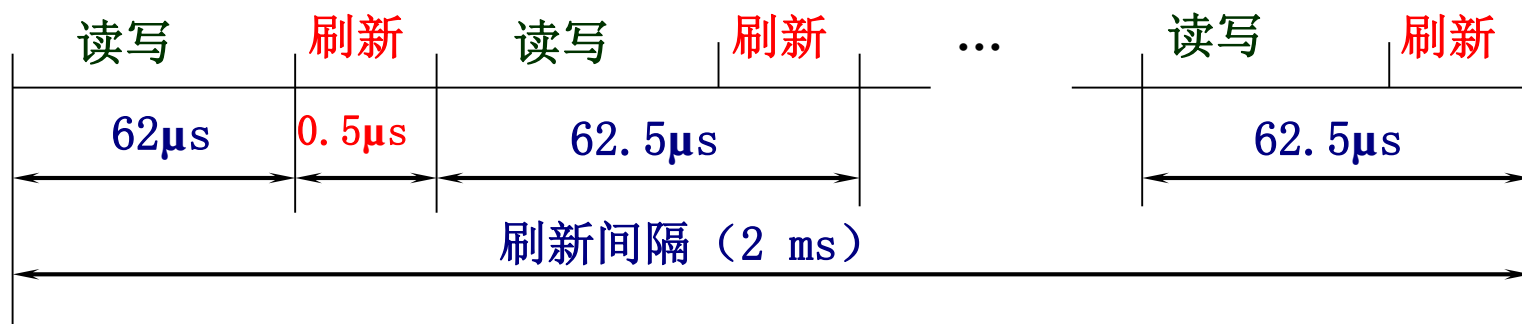
刷新很频繁，没有充分利用所允许的最大刷新闻隔。

2. 刷新方式

(3) 异步刷新方式

是前两种方式的结合，充分利用了最大刷新间隔时间，把刷新操作平均分配到整个最大刷新间隔时间内进行。

相邻两行的刷新间隔 = 最大刷新间隔时间 ÷ 行数



特点：虽然也有死区，但比集中刷新方式的死区小得多，且减少了刷新次数。

2. 刷新方式

(4) 其它刷新方式

如不定期刷新，把刷新安排在CPU不访存的空闲时间里，不会出现死区，也不会降低存储器存取速度，但控制复杂。

3. 刷新控制

- 刷新控制电路的主要任务是解决刷新和CPU访问存储器之间的矛盾。当刷新请求和访存请求同时发生时，应优先进行刷新操作。
- 刷新对CPU是透明的。
- 刷新以行为单位进行，每一行中各记忆单元同时被刷新，故刷新操作时仅需要行地址，不需要列地址。
- 刷新操作类似于读出操作，但不需要信息输出。刷新时不需要加片选信号。
- 考虑刷新问题时，只需从单个芯片的存储容量着手，而不是从整个存储器的容量着手。

【例1】 64K×1 的芯片，共有8位行地址，在2ms 内至少应有256个刷新周期？

【解】 芯片共有8位行地址，译码后产生256个行，所以，在2ms 内至少应有256个刷新周期。

【例2】 某机主存容量1MB，用16K×1/片的DRAM芯片构成，芯片最大刷新周期为2ms。问：(2) 刷新一遍需要多少次刷新？主存不能提供读写服务的百分比占到多少？

【解】 16K×1的DRAM芯片有 $14/2=7$ 位行地址，即共有128个刷新行地址。所以，需要128次的刷新才能刷遍所有的单元。

若每次刷新所需的时间为 T_s ，则主存不能提供读写服务的百分比为： $((T_s \times 128) / 2\text{ms}) \times 100\%$

思考题：P184 1-10, 23, 24, 26

习题：P185 11, 12, 13, 16, 17, 18, 25, 27

思考题：P164 1, 2, 5-10, 23, 24, 26

习题：P164 3, 4, 11, 12, 13, 16, 17, 18, 25, 27

5.3.4 半导体只读存储器（ROM）

- **特点：**非易失性存储器，造价比RAM低，集成度高，组成结构比RAM简单。
- **用途：**存放软件；存放微程序；存放特殊编码

1. ROM类型

(1) 掩模式ROM (MROM)

厂家制造芯片时把数据用光刻掩摸写入芯片，不能改。

特点：可靠性高，集成度高，批量生产价格便宜，但用户对厂家依赖性大，灵活性差。

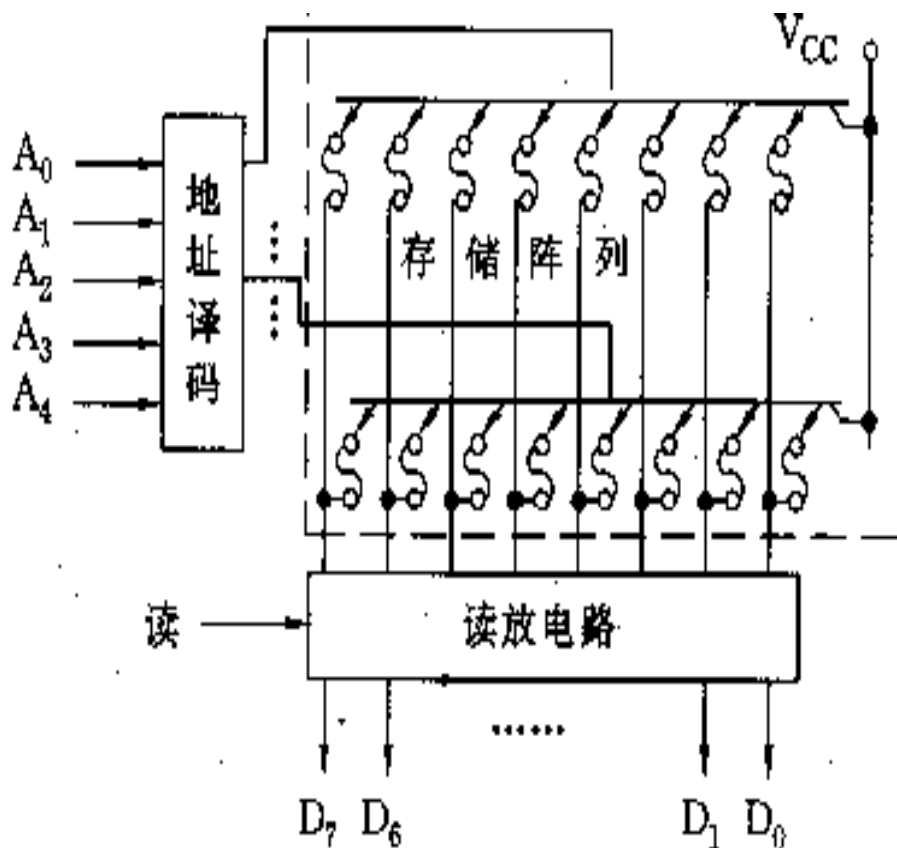
5.3.4 半导体只读存储器 (ROM)

1. ROM类型

(2) 一次可编程ROM (PROM)

用户可用专门的编程器或写入器，加过载电压来写入信息，但只能写入一次。

存储单元可分为熔丝型、二极管型等。

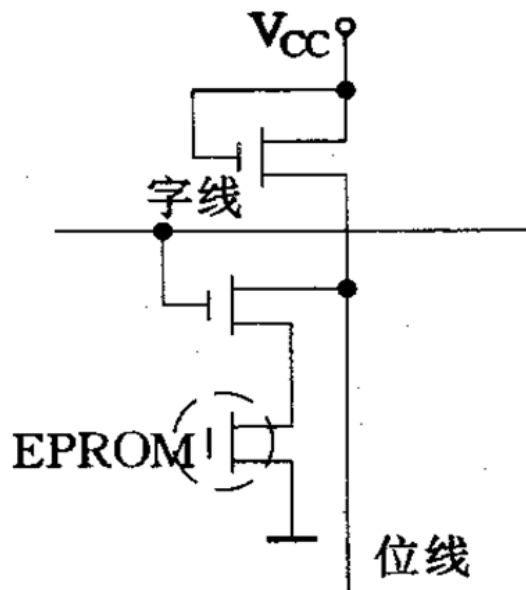


5.3.4 半导体只读存储器 (ROM)

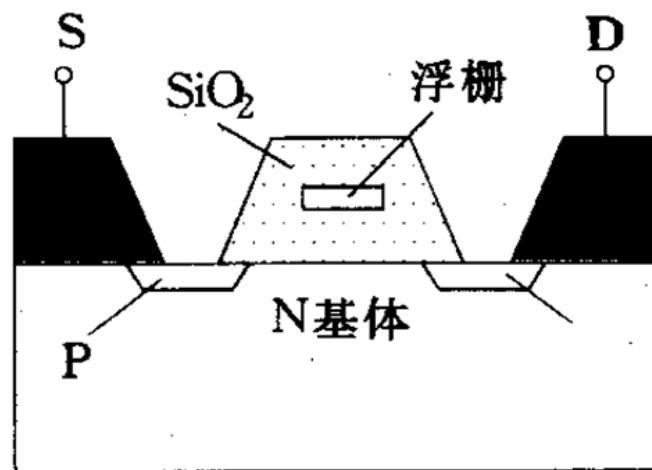
1. ROM类型

(3) 可擦除可编程ROM (EPROM)

■ UVEPROM (简称 EPROM)



(a) 基本存储电路



(b) P沟道FAMOS管结构

5.3.4 半导体只读存储器（ROM）

1. ROM类型

(3) 可擦除可编程ROM (EPROM)

■ UVEPROM（简称 EPROM）

需用紫外线灯制作的擦抹器照射存储器芯片上的透明窗口，擦除芯片中信息，只能对整个芯片擦除，不能对芯片中个别存储单元单独擦除。

■ EEPROM

采用电气方法进行擦除。在联机条件下，既可以用**字擦除**方式擦除，也可以用**数据块擦除**方式擦除。

5.3.4 半导体只读存储器（ROM）

1. ROM类型

(4) 闪速存储器（Flash Memory）

存储结构与EEPROM类似，具备RAM与ROM的所有功能，但擦除、重写速度比EEPROM快很多。

可联机工作，在计算机内进行擦除和重写。擦写次数在10万次以上，读取速度小于90ns。具有密度高、可靠性高、体积小、功耗低等特点。

- 闪存可分为：
- 1) **NOR型** 读速度较快，擦除和写很慢
接口简单 （常用于存程序）
 - 2) **NAND型** 擦除和写速度快，集成度高
接口复杂 （常用于存数据）

5.3.4 半导体只读存储器 (ROM)

2. ROM芯片

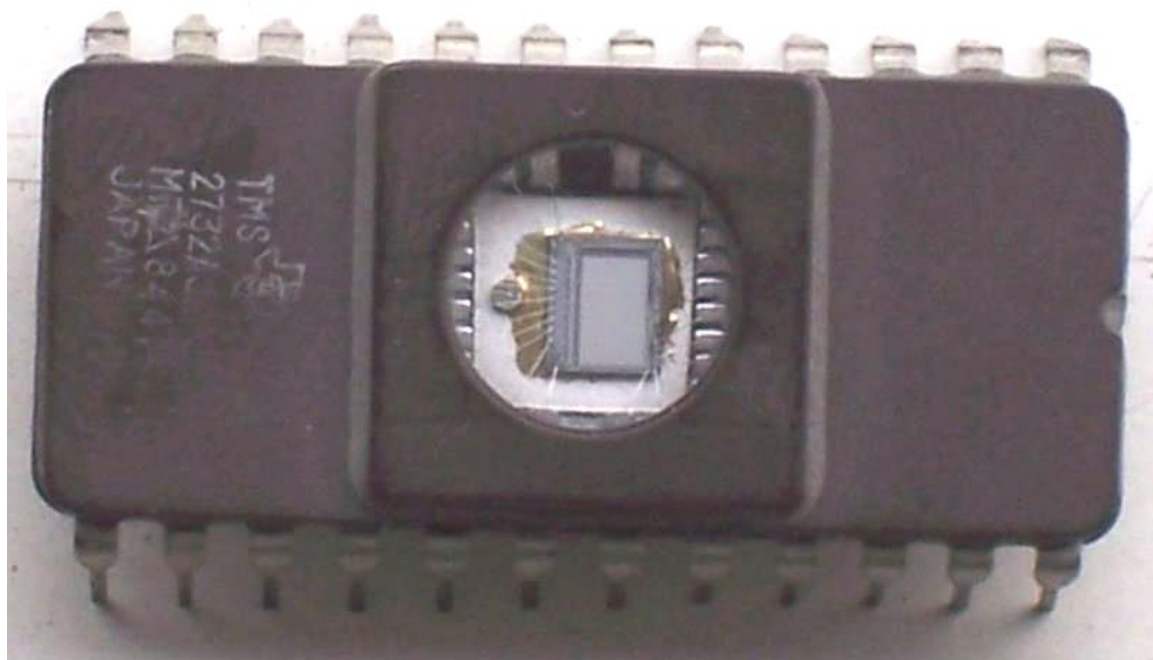
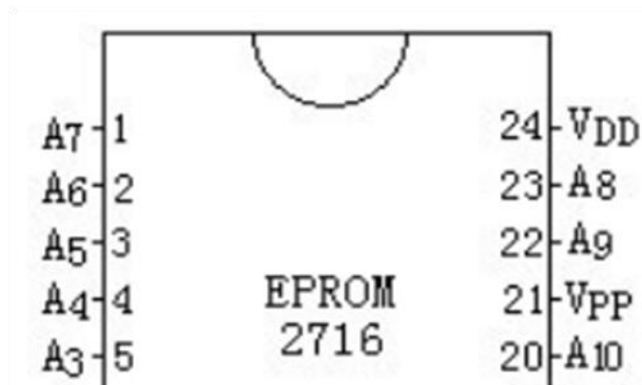
地址线

数据线

片选线

读出/编程输入线

电源线等



5.3.4 半导体只读存储器 (ROM)

2. ROM芯片

- 闪速存储器 (Flash Memory) : AT29C040A

读写时间120ns

**4-Megabit
(512K x 8)
5-volt Only
256-Byte Sector
Flash Memory**

Pin Configurations

Pin Name	Function
A0 - A18	Addresses
\overline{CE}	Chip Enable
\overline{OE}	Output Enable
\overline{WE}	Write Enable
I/O0 - I/O7	Data Inputs/Outputs
NC	No Connect

DIP Top View



5.3.5 半导体存储器的封装

1. DIP芯片 (Dual In-line Package)

双列直插封装的存储芯片。

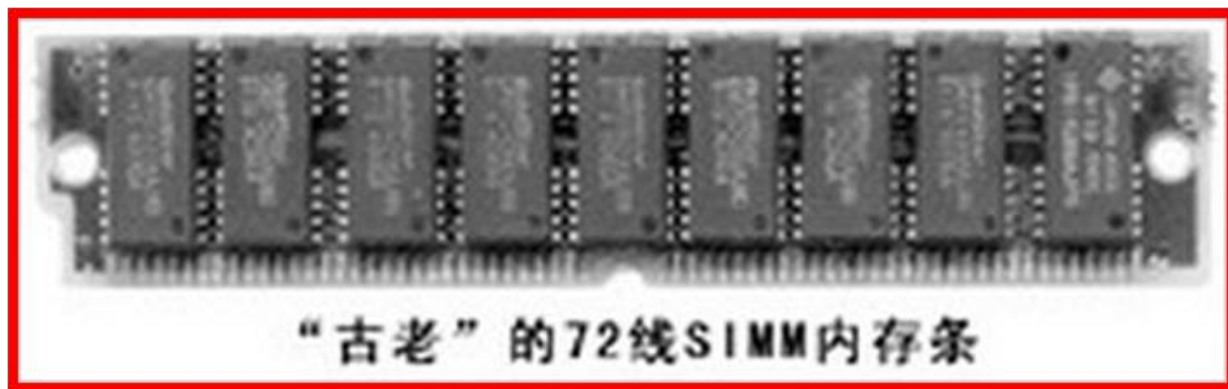
容量一般不可能很大，如 $64K \times 1$ 或 $256K \times 1$ 的芯片。

2. 内存条

是一条焊有多片存储芯片的印刷电路板，插在主板内存插槽中。内存条主要有单列直插存储模块 (SIMM)、双列直插存储模块 (DIMM) 和Rambus直插存储模块 (RIMM)。

5.3.5 半导体存储器的封装

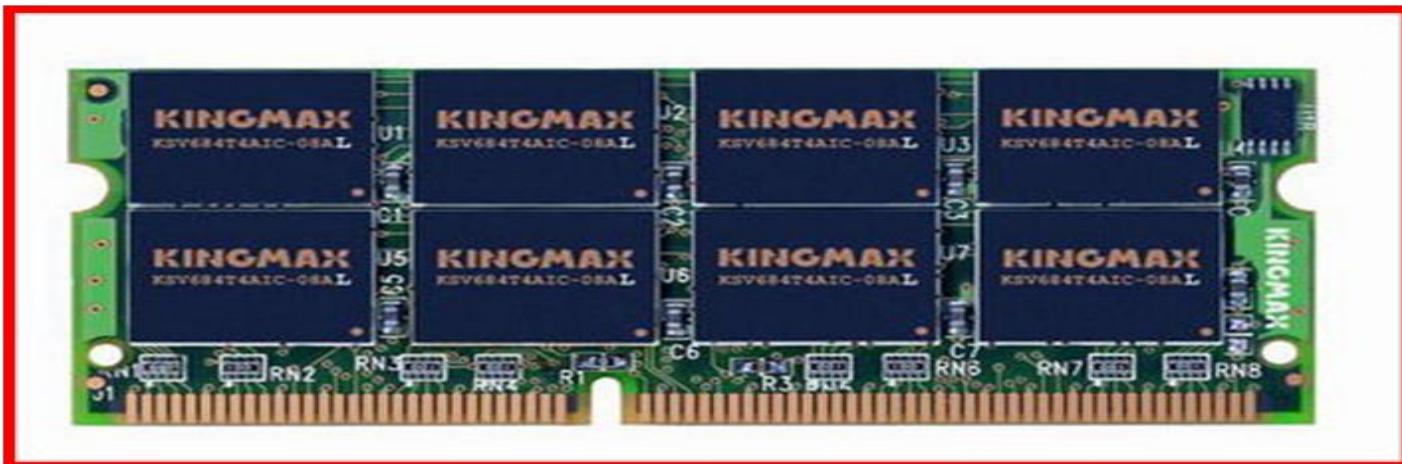
- SIMM (single in-line memory module) 单内置内存模型



168针SIMM插槽

5.3.5 半导体存储器的封装

- DIMM (dual in-line memory module) 双内置内存模型



184针DIMM插槽



计算机主板

§ 5.4 主存储器的连接与控制

5.4.1 主存容量的扩展

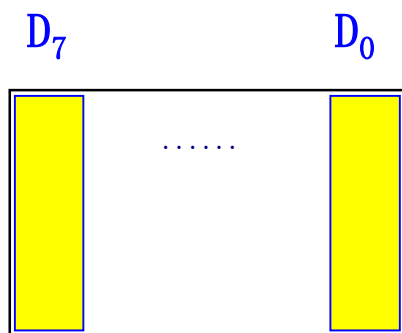
当一个存储器芯片的容量规格不能满足主存系统容量规格的要求时，需要用多个芯片来构成主存系统，即对存储体进行扩展。

容量扩展方式：

- ① 位扩展（位并连法）
- ② 字扩展（地址串联法）
- ③ 字位同时扩展

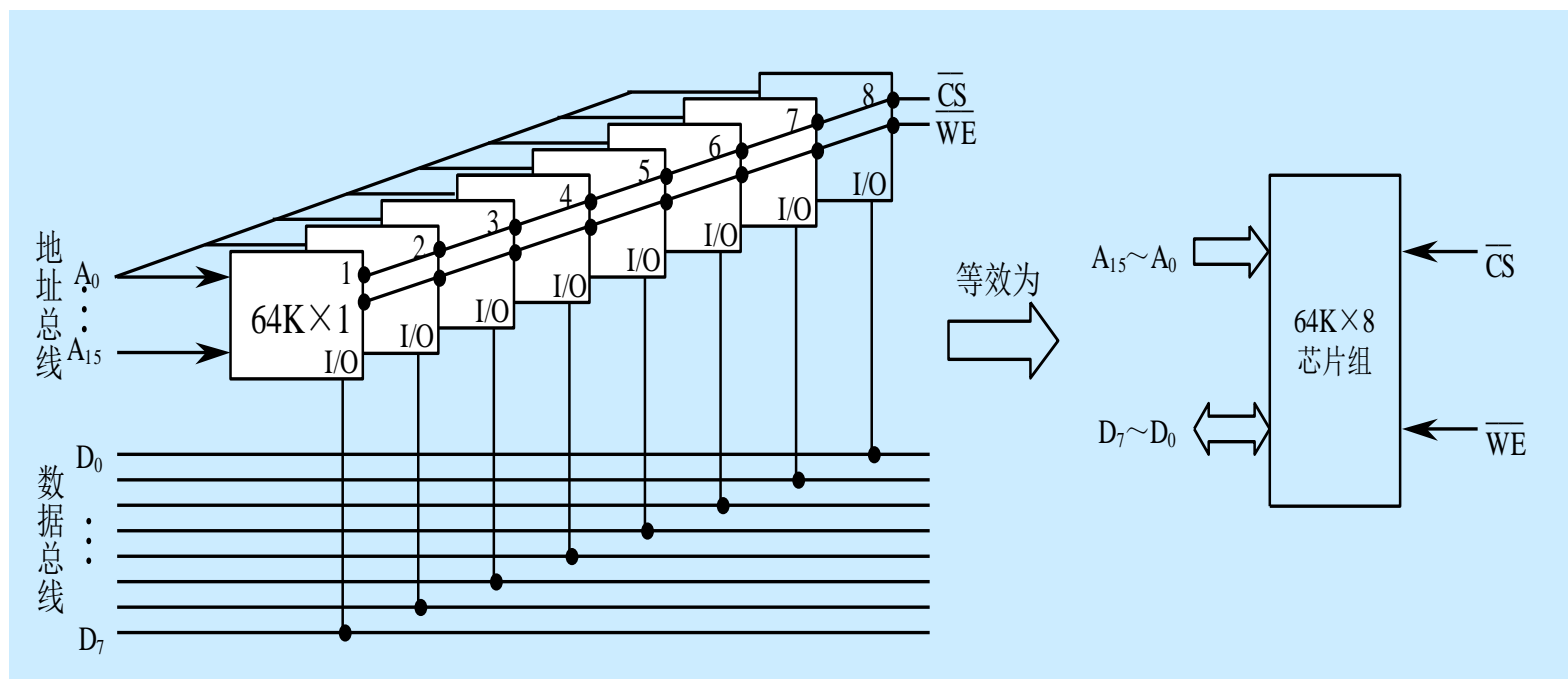
1. 位扩展

只在位数方向扩展（加大字长），而芯片的字数和存储器的字数是一致的。



连接方式：各存储芯片的地址线、片选线和读/写线并联
各芯片的数据线单独列出。

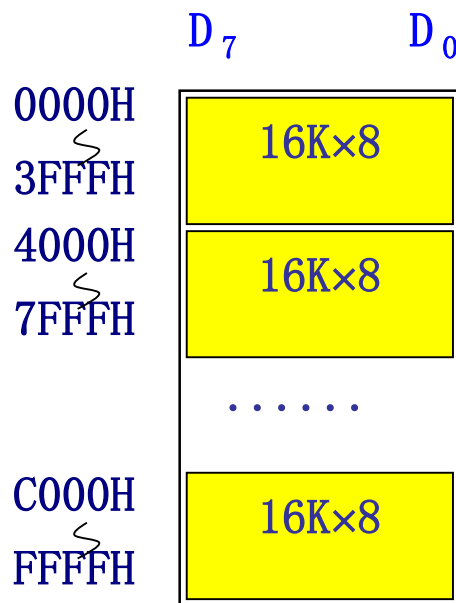
【例】用 $64\text{K} \times 1$ 的SRAM芯片组成 $64\text{K} \times 8$ 的存储器



扩展条件： 设目标容量为 M 字 $\times N$ 位，存储器芯片容量为 m 字 $\times n$ 位，
 $M=m$ ， $N>n$ ， 则需要的存储器芯片数= N/n 。

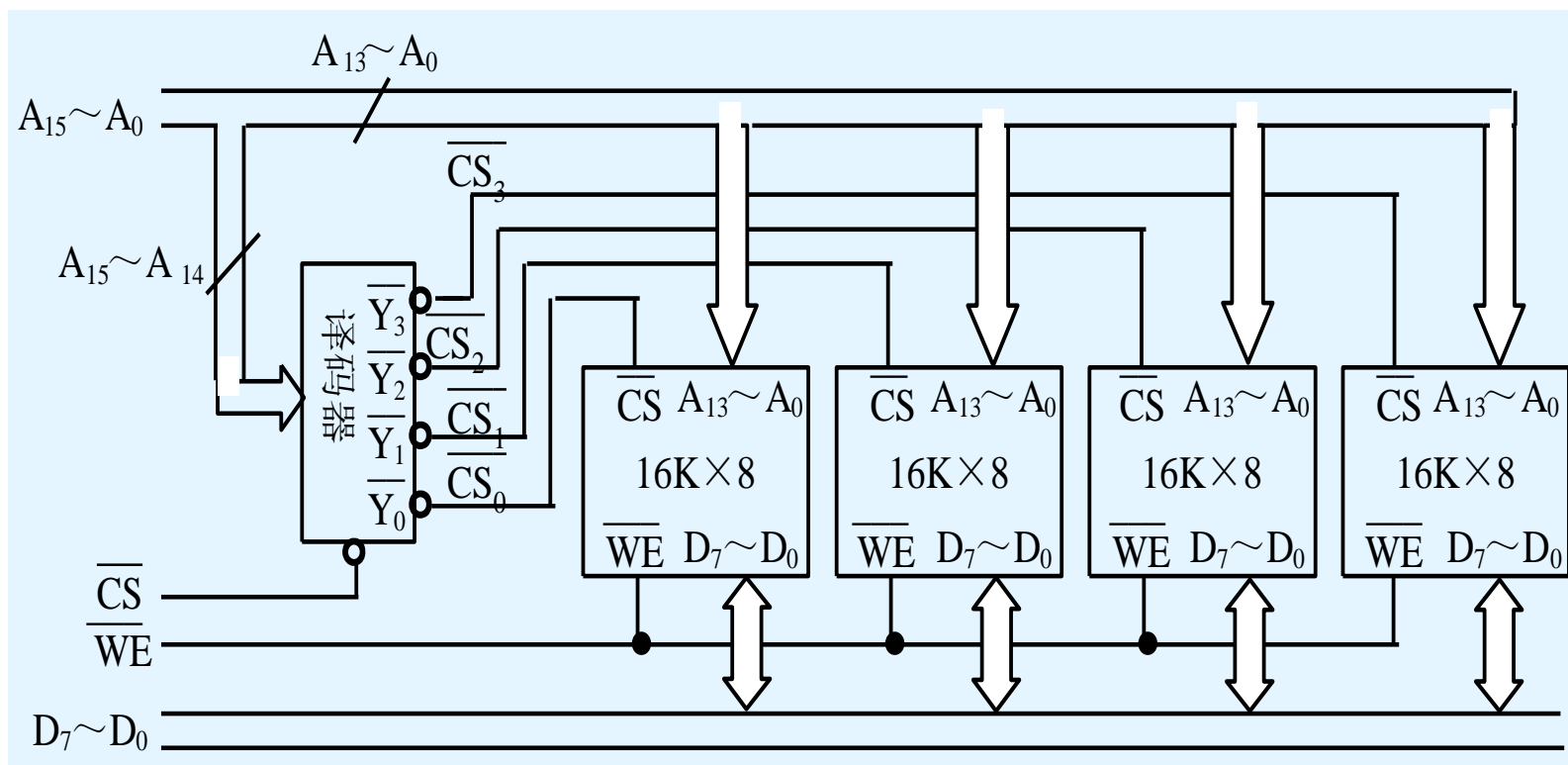
2. 字扩展

仅在字数方向扩展，位数不变。



连接方式：各芯片的地址线、数据线、读/写线并联
由片选信号来区分各个芯片。

【例】用 $16\text{K} \times 8$ 的SRAM芯片组成 $64\text{K} \times 8$ 存储器



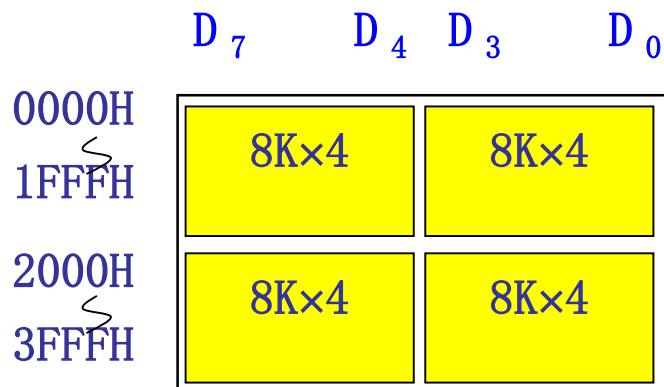
在同一时间内4个芯片中最多只有一个芯片被选中。

这4片SRAM芯片的地址分配为:

芯片编号	$A_{15} A_{14}$	$A_{13} A_{12} \dots A_0$	地址范围
SRAM芯片 #0	0 0	$\begin{array}{ccc} 0 & 0 & \dots & 0 \\ & & \{ & \\ 1 & 1 & \dots & 1 \end{array}$	0000H-3FFFH
SRAM芯片 #1	0 1	$\begin{array}{ccc} 0 & 0 & \dots & 0 \\ & & \{ & \\ 1 & 1 & \dots & 1 \end{array}$	4000H-7FFFH
SRAM芯片 #2	1 0	$\begin{array}{ccc} 0 & 0 & \dots & 0 \\ & & \{ & \\ 1 & 1 & \dots & 1 \end{array}$	8000H-BFFFH
SRAM芯片 #3	1 1	$\begin{array}{ccc} 0 & 0 & \dots & 0 \\ & & \{ & \\ 1 & 1 & \dots & 1 \end{array}$	C000H-FFFFH

3. 字位同时扩展

例：用 $8K \times 4$ 芯片组成 $16K \times 8$ 存储器



扩展条件：

目标容量为 M 字 $\times N$ 位，

存储器芯片容量为 m 字 $\times n$ 位， $M > m$ ， $N > n$ ，

则需要的存储器芯片数 = $(M/m) \times (N/n)$

5.4.2 存储芯片的地址分配和片选

片选——选择存储器芯片，由CPU送出的高位地址决定。


字选——从存储器芯片中选择相应的存储单元，由CPU送出的N条低位地址线完成， $N = \log_2 M$

片选信号的产生方法常见的有：

- ①线选法
- ②全地址译码
- ③部分地址译码。

1. 线选法

将高位地址线直接（或经过反相器）分别连接至各芯片的片选端，当某地址线为“0”时，就选中与之对应的芯片（假设片选信号是低电平有效）。

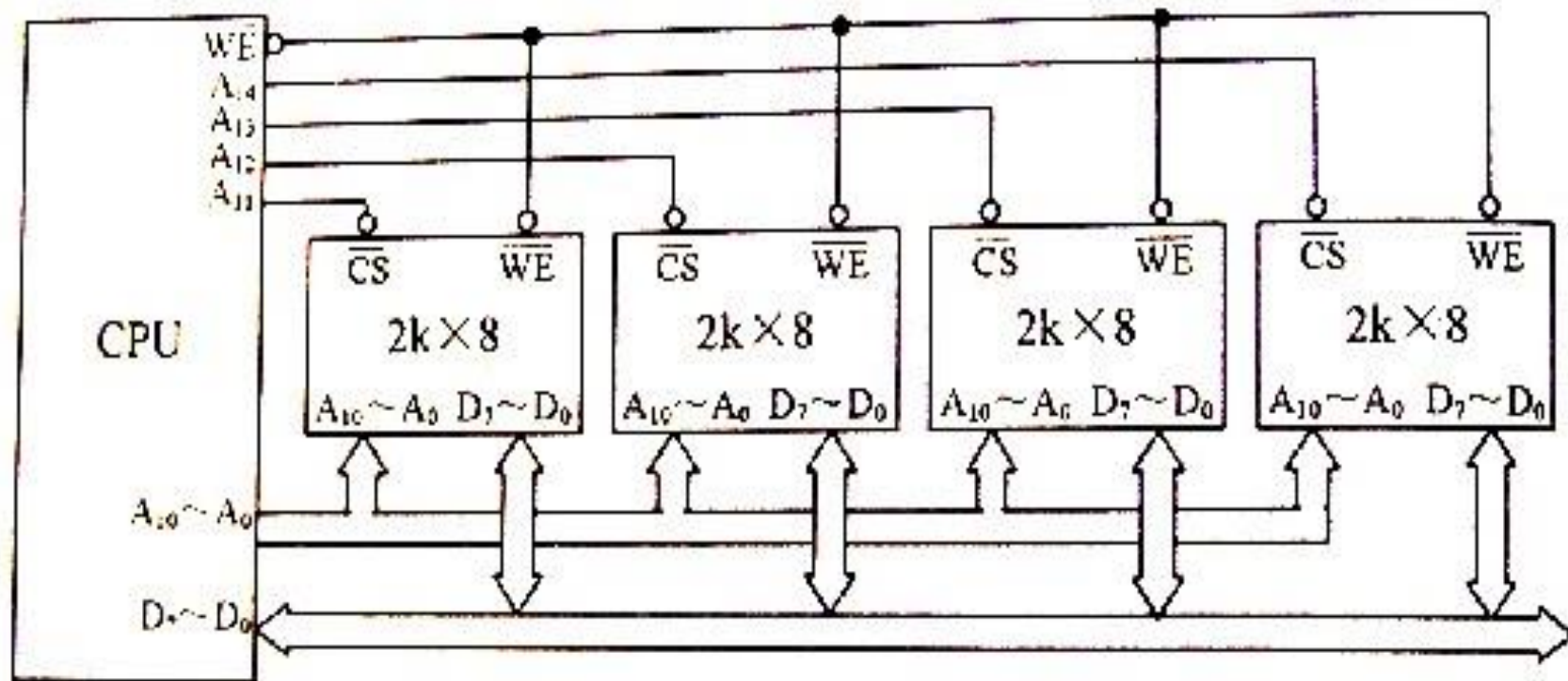
 **特点：**选择芯片不需外加逻辑电路，线路简单，但仅适合芯片较少的场合，且不能充分利用系统的存储器空间。

【例】用 $2K \times 8$ 的存储器芯片构成 $8K \times 8$ 的存储器系统，假设地址总线有20位。

解：进行字扩展，需要4片芯片。

低位地址： $A_{10} \sim A_0$ 作为片内寻址线；

高位地址： $A_{19} \sim A_{11}$ 作为片选地址。



芯片	$A_{19} \sim A_{15}$ (不用)	$A_{14} \sim A_{11}$ (片选地址)	$A_{10} \sim A_0$ (片内地址)	地址范围 (空间)
0#	00000	1 1 1 <u>0</u>	0000000000 1111111111	07000~077FFH
1#	00000	1 1 <u>0</u> 1	0000000000 1111111111	06800~06FFFH
2#	00000	1 <u>0</u> 1 1	0000000000 1111111111	05800~05FFFH
3#	00000	<u>0</u> 1 1 1	0000000000 1111111111	03800~03FFFH

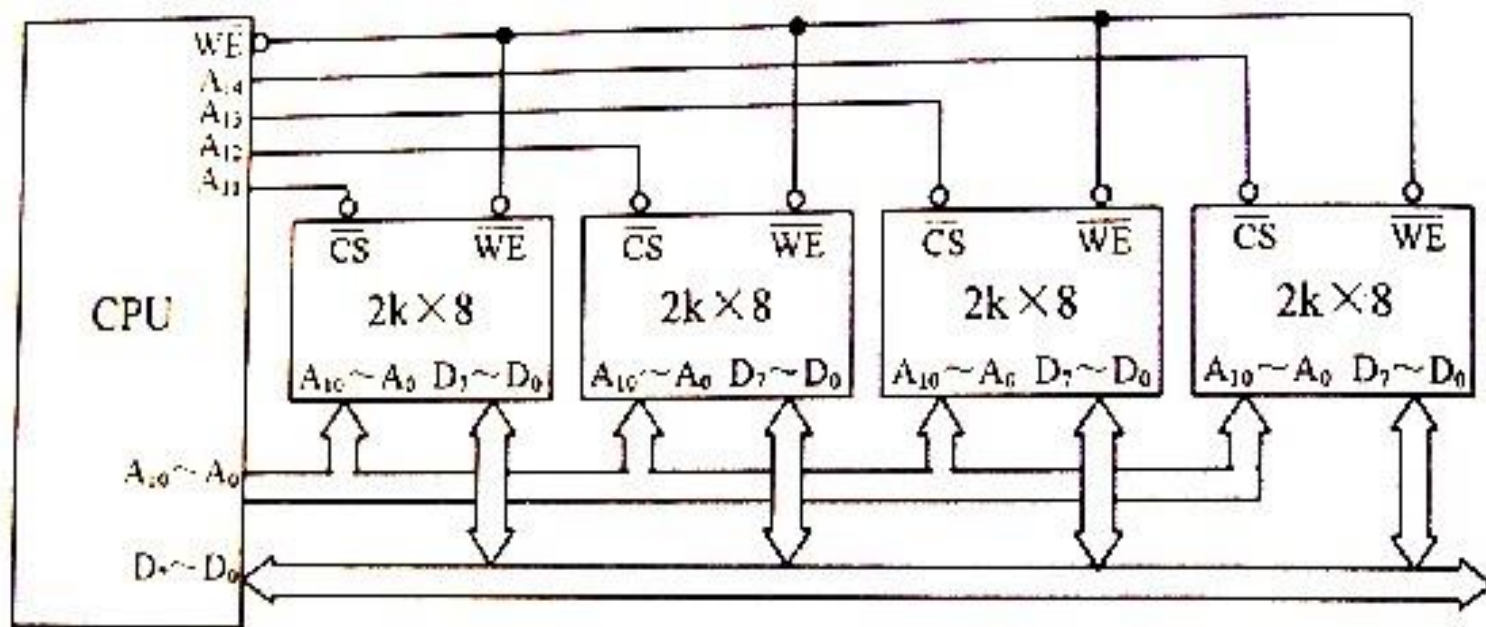
注意：片选地址线只能有一位有效，不能同时多位有效。

? 思考：

上例中，芯片可否工作在00000~007FFH的地址范围？为什么？

【例】用 $2K \times 8$ 的存储器芯片构成 $8K \times 8$ 的存储器系统，假设地址总线有20位。

片选采用线选法。



2. 全译码法

外部地址的除片内寻址外的所有高位地址都作为译码器的输入，产生各芯片的片选信号。

【例】用 $2K \times 8$ 的存储器芯片构成 $8K \times 8$ 的存储器系统，假设地址总线为 $A_{19} \sim A_0$

芯片	$A_{19} \sim A_{11}$ (片选用)	$A_{10} \sim A_0$ (片内地址)	地址范围(空间)
0#	000000000	00000000000 i i i i 1 1 1 1 1 1 1	00000~007FFH (0~2K-1)
1#	000000001	00000000000 i i i i 1 1 1 1 1 1 1	00800~00FFFH (2K~4K-1)
2#	000000010	00000000000 i i i i 1 1 1 1 1 1 1	01000~017FFH (4K~6K-1)
3#	000000011	00000000000 i i i i 1 1 1 1 1 1 1	01800~01FFFH (6K~8K-1)

【例】用 $2K \times 8$ 的存储器芯片构成 $8K \times 8$ 的存储器系统，
假设地址总线为 $A_{19} \sim A_0$

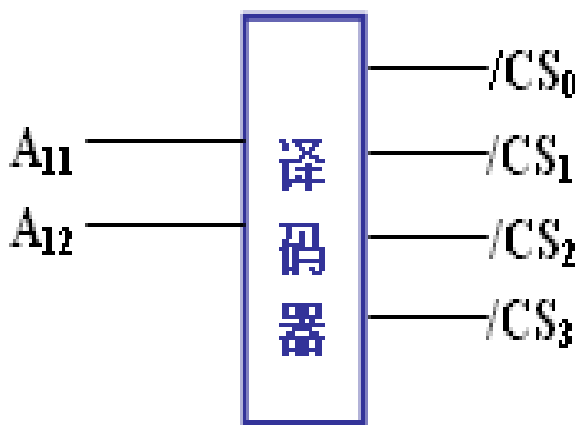


全译码法特点：每片芯片的地址范围是唯一确定且连续的，不会产生地址重叠的存储器，对译码电路要求较高。

3. 部分译码法

将除片内寻址外的全部高位地址线的一部分作为译码器的输入来产生片选信号。

【例】用4片 $2K \times 8$ 的存储器芯片构成 $8K \times 8$ 的存储器系统，地址总线为： $A_{15} \sim A_0$ ，采用部分译码法。



地址线 $A_{10} \sim A_0$ 作为片内寻址线，4个片选信号只需要2位地址线(A_{11} 、 A_{12})参与译码，即仅在8K范围内译码。

芯片	$A_{19} \sim A_{13}$ (不用) $A_{19} \sim A_{13}$ (参与译码)	$A_{10} \sim A_0$ (片内地址)	地址范围 (空间)
0#	0000000 00	00000000000 i i i i 1 1 1 1 1 1 1 1	00000~007FFH (0~2K-1)
1#	0000000 01	00000000000 i i i i 1 1 1 1 1 1 1 1	00800~00FFFH (2K~4K-1)
2#	0000000 10	00000000000 i i i i 1 1 1 1 1 1 1 1	01000~017FFH (4K~6K-1)
3#	0000000 11	00000000000 i i i i 1 1 1 1 1 1 1 1	01800~01FFFH (6K~8K-1)

令未用到的高位地址全为0，这样确定的存储器地址称为**基本地址**。本例中的基本地址为：0000H~01FFH。

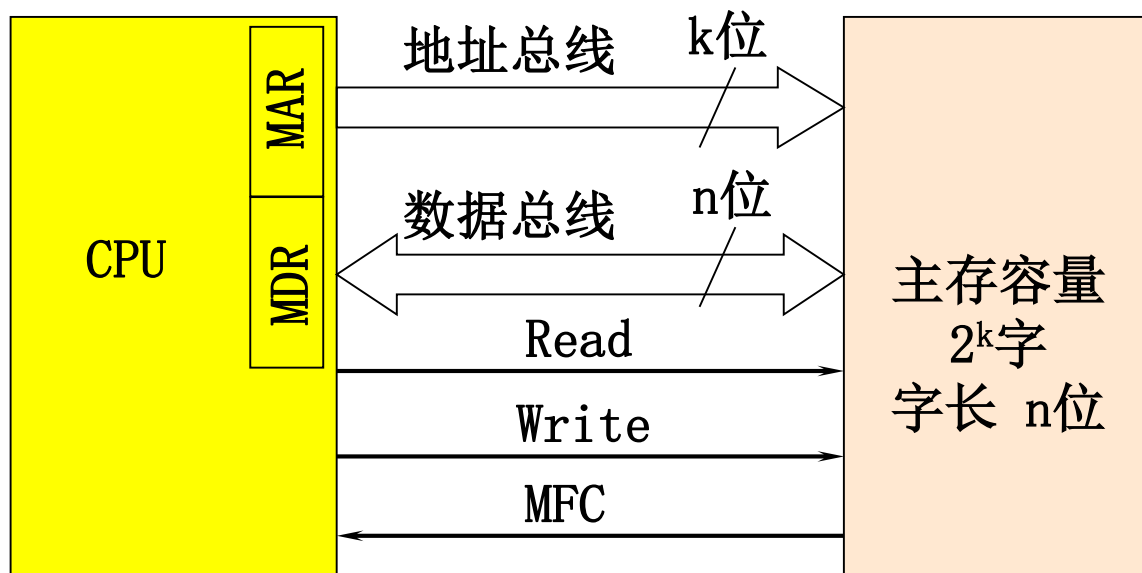
特点：较全译码法简单，但存在地址重叠区。

5.4.3 主存储器和CPU的连接

1. 主存与CPU连接需要考虑的问题

- 1) 寻址逻辑 即，如何按地址来产生存储芯片的片选信号及片内地址；（上节）
- 2) 主存与CPU读写控制信号的连接、工作时序的配合；
- 3) 主存与CPU速度的匹配
（ \because CPU的速度 \gg 主存速度）；
- 4) 使用DRAM芯片时的刷新问题；
- 5) 系统总线的负载能力。

2. 主存和CPU之间的硬连接

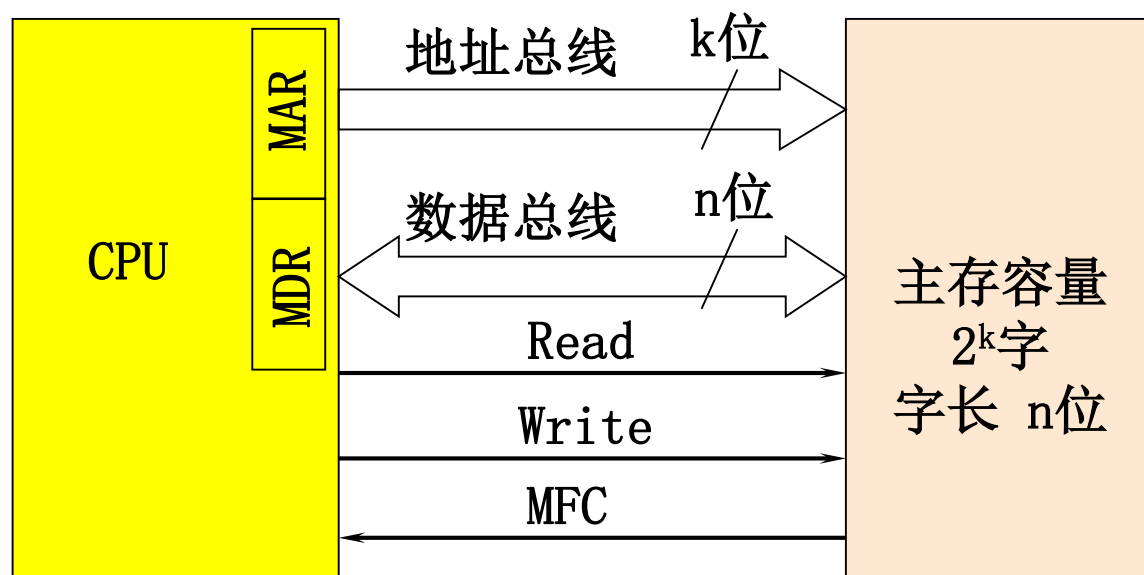


MAR: 存储器的地址寄存器;

MDR: 存储器的数据寄存器;

MFC: 主存的工作完成信号。

2. 主存和CPU之间的硬连接



MAR和MDR从功能上看属于主存，但在小微型机中常放在CPU内。

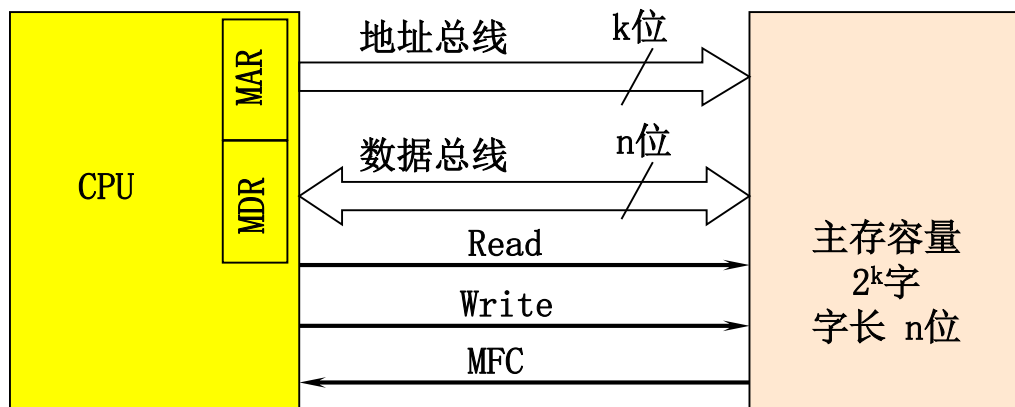
不同的计算机提供的主存控制信号可能各不相同，但都必须提供读/写控制信号。

3. CPU对主存的基本操作

(1) 读操作

从CPU送来的地址所指定的存储单元中取出信息，再送给CPU，其操作过程是：

- 地址→MAR→AB CPU将地址信号送至地址总线；
- Read CPU发读命令；
- Wait for MFC 等待存储器工作完成信号；
- $M(MAR) \rightarrow DB \rightarrow MDR$ 读出信息经数据总线送至CPU

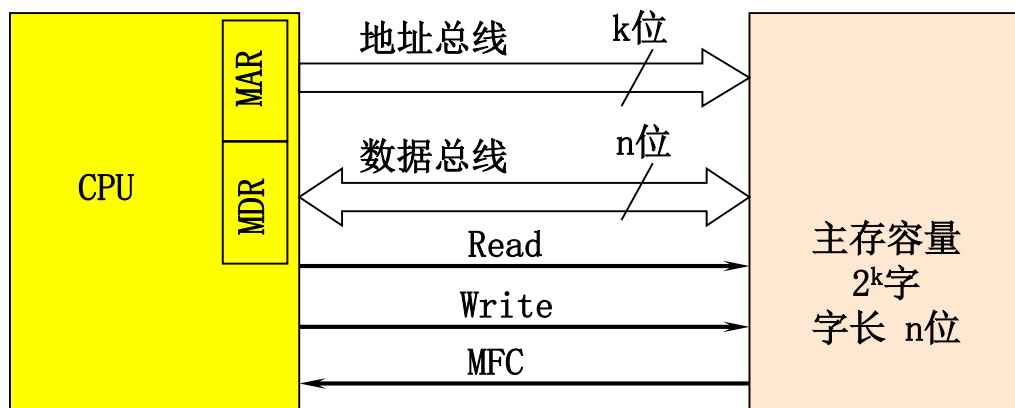


3. CPU对主存的基本操作

(2) 写操作

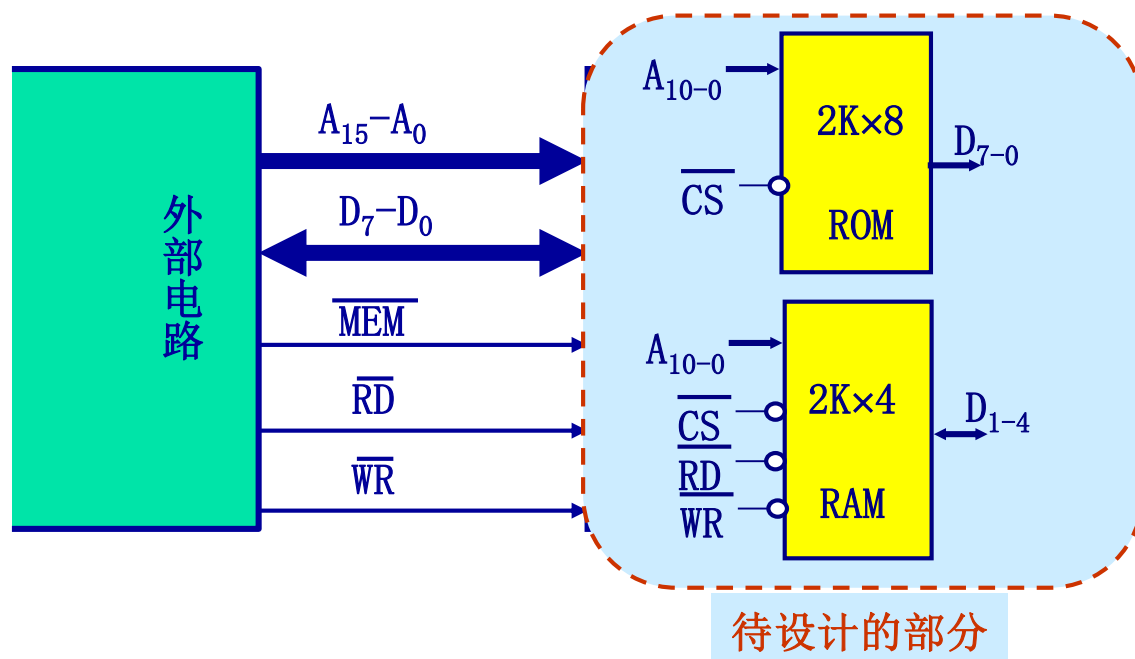
指将要写入的信息存入CPU所指定的存储单元中：

- 地址→MAR→AB CPU将地址信号送至地址总线；
- 数据→MDR→DB CPU将要写入的数据送至数据总线；
- Write CPU发写命令；
- Wait for MFC 等待存储器工作完成信号。



4. 8位静态存储器设计举例

【例】用 $2K \times 8$ /片的ROM芯片、 $2K \times 4$ /片的SRAM芯片，组成 $8K \times 8$ 的内存储器（其中2KB ROM，6KB RAM）。系统提供的控制信号有 \overline{RD} （为低时读）、 \overline{WR} （为低时写）、 \overline{MEM} （为低时表示访问内存）。



【解】

使用ROM芯片1片、SRAM芯片6片，其中 6片RAM按每两片（U1和U2，U3和U4，U5和U6）组成位扩展，分别接入数据线；各芯片的片内地址线接外部提供的地址线的低11位 A_{10-0} ；芯片的片选信号为：

	A_{15}	A_{10}	A_0		D_7	D_4	D_3	D_0
$\overline{CS0}$	0 0 0 0	0 0 0 0	0 0 0 0 0 0 0 0	0000H	2K×8 ROM U0			
	0 0 0 0	0 1 1 1	1 1 1 1 1 1 1 1	07FFH				
$\overline{CS1}$	0 0 0 0	1 0 0 0	0 0 0 0 0 0 0 0	0800H	2K×4 U2		2K×4 U1	
	0 0 0 0	1 1 1 1	1 1 1 1 1 1 1 1	0FFFH				
$\overline{CS2}$	0 0 0 1	0 0 0 0	0 0 0 0 0 0 0 0	1000H	2K×4 U4		2K×4 U3	
	0 0 0 1	0 1 1 1	1 1 1 1 1 1 1 1	17FFH				
$\overline{CS3}$	0 0 0 1	1 0 0 0	0 0 0 0 0 0 0 0	1800H	2K×4 U6		2K×4 U5	
	0 0 0 1	1 1 1 1	1 1 1 1 1 1 1 1	1FFFH				

	A ₁₅	A ₁₀	A ₀		D ₇	D ₄	D ₃	D ₀
$\overline{\text{CS0}}$	0 0 0 0	0 0 0 0	0 0 0 0 0 0 0 0	0000H	2K×8 ROM U0			
	0 0 0 0	0 1 1 1	1 1 1 1 1 1 1 1	07FFH				
$\overline{\text{CS1}}$	0 0 0 0	1 0 0 0	0 0 0 0 0 0 0 0	0800H	2K×4 U2		2K×4 U1	
	0 0 0 0	1 1 1 1	1 1 1 1 1 1 1 1	0FFFH				
$\overline{\text{CS2}}$	0 0 0 1	0 0 0 0	0 0 0 0 0 0 0 0	1000H	2K×4 U4		2K×4 U3	
	0 0 0 1	0 1 1 1	1 1 1 1 1 1 1 1	17FFH				
$\overline{\text{CS3}}$	0 0 0 1	1 0 0 0	0 0 0 0 0 0 0 0	1800H	2K×4 U6		2K×4 U5	
	0 0 0 1	1 1 1 1	1 1 1 1 1 1 1 1	1FFFH				

各芯片的片内地址线应接到外部提供的地址线的低11位A₁₀₋₀，
芯片的片选逻辑(部分译码)为：

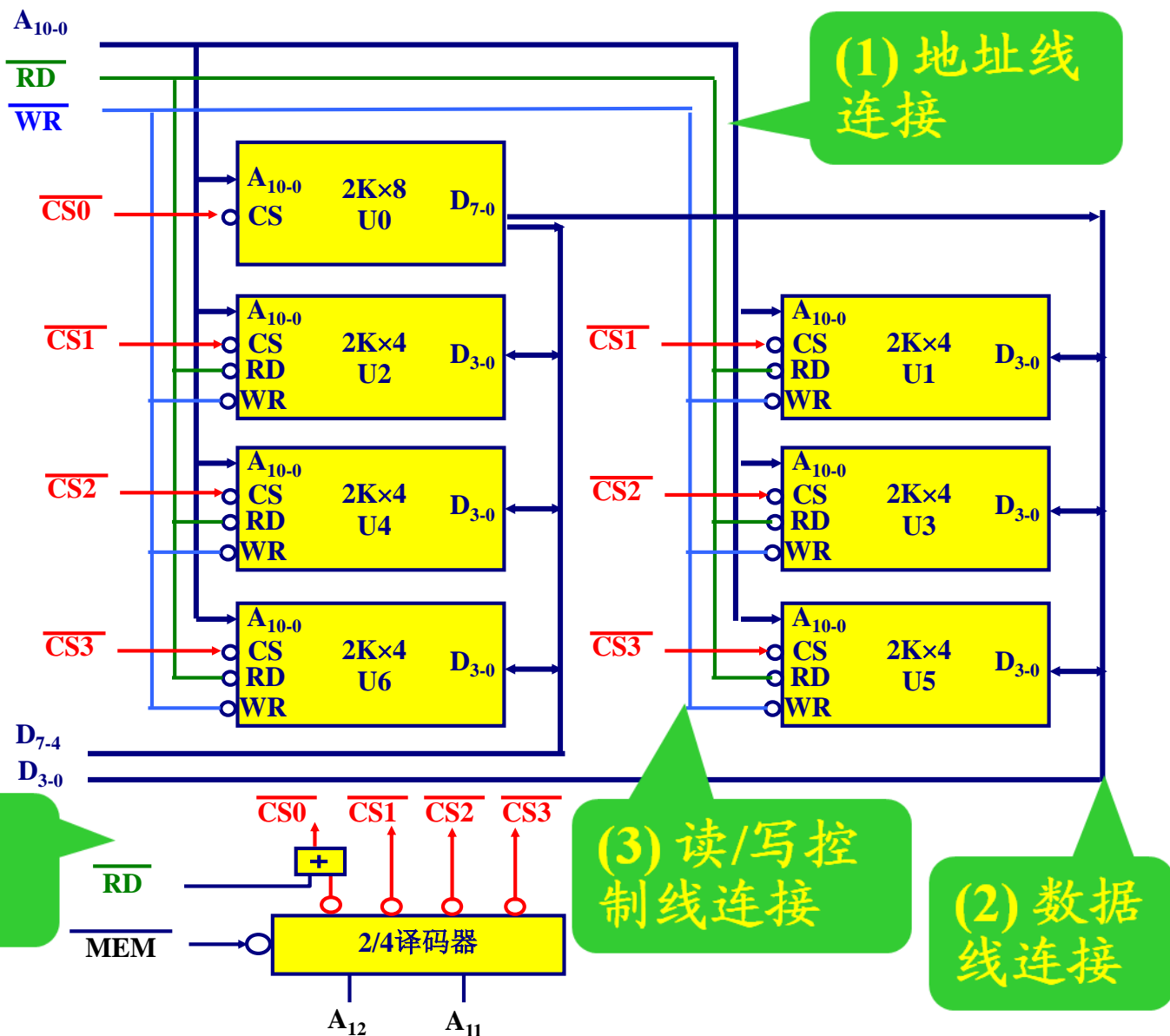
$$\overline{\text{CS0}} = \overline{\overline{\text{A}_{12}} \overline{\text{A}_{11}} / \overline{\text{MEM}} / \overline{\text{RD}}}$$

$$\overline{\text{CS1}} = \overline{\overline{\text{A}_{12}} \overline{\text{A}_{11}} / \overline{\text{MEM}}}$$

$$\overline{\text{CS2}} = \overline{\overline{\text{A}_{12}} \overline{\text{A}_{11}} / \overline{\text{MEM}}}$$

$$\overline{\text{CS3}} = \overline{\overline{\text{A}_{12}} \overline{\text{A}_{11}} / \overline{\text{MEM}}}$$

假定
芯片的读
写控制也
采用两条
线(/RD和
/WR), 则
存储器的
组成逻辑
图为:



思考题：P184 1-10, 23, 24, 26

习题：P185 11, 12, 13, 16, 17, 18, 25, 27

思考题：P164 1, 2, 5-10, 23, 24, 26

习题：P164 3, 4, 11, 12, 13, 16, 17, 18, 25, 27



5.4.4 主存的校验

1. 主存的奇偶校验

每个存储单元共**存储9位信息**（其中8位数据， 1位奇偶校验位），信息中**“1”的个数是奇数**。

写入数据时：奇偶校验电路先计算字节的奇偶校验位值，并存储在主存中。

读出数据时：如果9位数据里“1”的个数为奇数时，表示数据正确；否则数据出错。

奇偶校验仅能检测出奇数个位数错误，不能修正错误！

5.4.4 主存的校验

2. 错误检验与校正（ECC）

ECC主存用一组附加数据位来存储“校验和”。产生ECC码所需的位数取决于系统所用的二进制字长。从主存中读取数据时，将取到的实际数据与其ECC码比较。如果匹配，则数据正确；否则，ECC码能够将出错的一位（或几位）鉴别出来，然后改正错误，再将数据传给CPU。

ECC不仅能检测错误，还能在不打扰计算机工作的情况下修正错误。

现代PC机中主存的容错能力被分为无奇偶校验、奇偶校验和ECC三级。

5.4.5 PC系列微机的存储器接口

1. 8位存储器接口

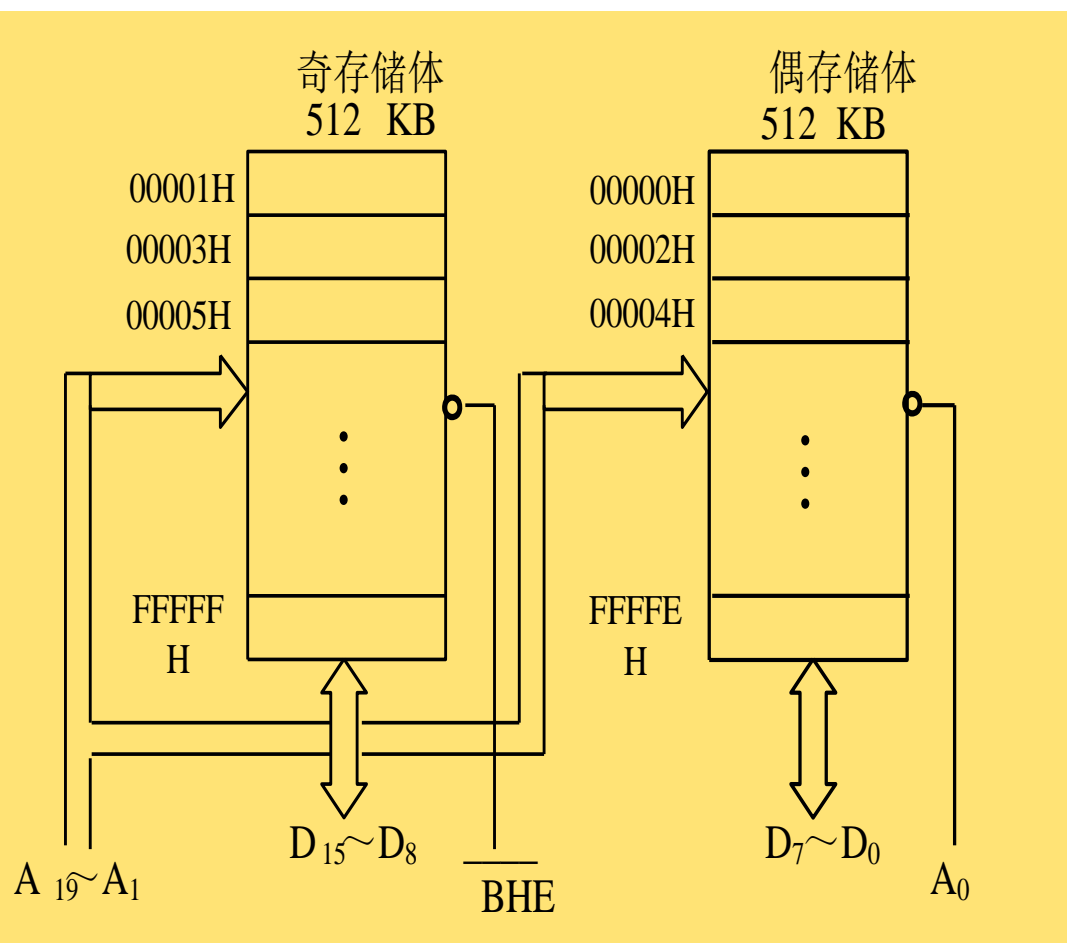
如果数据总线为8位，主存按字节编址，总线周期就等于存取周期，一个总线周期可读写8位。

2. 16位存储器接口

对于16位的微处理器，在一个总线周期内可以传送一个字节数据，也可读写一个从偶地址开始的字（规则字）。如果读写的是非规则字，则需要安排两个总线周期才能实现。

5.4.5 PC系列微机的存储器接口

2. 16位存储器接口



\overline{BHE}	A_0	特征
0	0	全字 (规则字) 传送
0	1	在数据总线高 8 位传送字节
1	0	在数据总线低 8 位传送字节
1	1	备用

5.4.5 PC系列微机的存储器接口

3. 32位存储器接口

80386/80486微处理器的数据总线和地址总线为32位，为与8086等微处理器兼容，要求设计存储器系统时必须满足单字节、双字节和四字节等不同访问。

4. 64位存储器接口

Pentium微处理器的数据总线为64位，地址总线为32位。64位存储器系统由8个存储体组成，每个存储体的能提供8位数据的读写。

§ 5.5 提高主存读写速度的技术

5.5.1 主存与CPU速度的匹配

CPU需频繁和主存交换数据，主存速度将直接影响到系统的性能。

提高主存速度的途径：

- 1) 提高存储芯片的数据存取速度
- 2) 改进存储体系结构

5.5.2 几种提高主存读写速度的技术

1. 快速页模式动态随机存储器（FPM DRAM）

连续读写时可保持行地址不变而只改变列地址，可对给定行的所有数据进行更快的访问。

- 标准的FPM DRAM支持5-3-3-3的突发模式周期。
- 内存条主要采用72线的SIMM封装，其存取速度一般在60~100ns左右。

5.5.2 几种提高主存读写速度的技术

2. 扩展数据输出 (EDO DRAM)

采用一种特殊的主存**读出控制**逻辑，不必等待当前的读写周期完成即可启动下一个读写周期，节省了重选地址的时间，提高了读写速度。

- 可获得5-2-2-2的突发模式周期，性能与FPM DRAM相比改善了22%，而制造成本与FPM DRAM相近。
- 内存条目前主要采用72线的SIMM形式封装，也有少部分采用168线的DIMM封装，存取时间约为50~70ns。

5.5.2 几种提高主存读写速度的技术

3. 同步动态随机存储器（SDRAM）

是一种与主存总线运行**同步**的DRAM, 基本原理是将CPU和RAM通过一个相同的时钟锁在一起, 使RAM和CPU能够共享一个时钟周期, 以相同的速度同步工作。

- 在同步脉冲控制下工作, 取消了主存等待时间, 减少数据传送延迟时间, 加快系统速度。
- SDRAM突发模式可达到5-1-1-1, 比EDO快将近20%。
- SDRAM采用新的双存储体结构, 内含两个交错的存储矩阵, 允许两个主存页面同时打开。
- SDRAM普遍采用168线的DIMM封装, 目前SDRAM的工作频率已达100MHz、133MHz, 能与当前的CPU同步运行。

5.5.2 几种提高主存读写速度的技术

4. 双数据传输同步动态随机存储器（DDR SDRAM）

不仅能在时钟脉冲的上升沿读出数据，还能在下降沿读出数据，不需要提高时钟频率就能加倍提高SDRAM的速度。

- DDR2:每次读写数据的位数是DDR的2倍, 工作电压1.8V
- DDR3:每次读写数据的位数是DDR的4倍。1.8V/1.5V

如：DDR3 800/1066/1333/1600

核心频率100/133/166/200MHz

5.5.2 几种提高主存读写速度的技术

5. Rambus DRAM

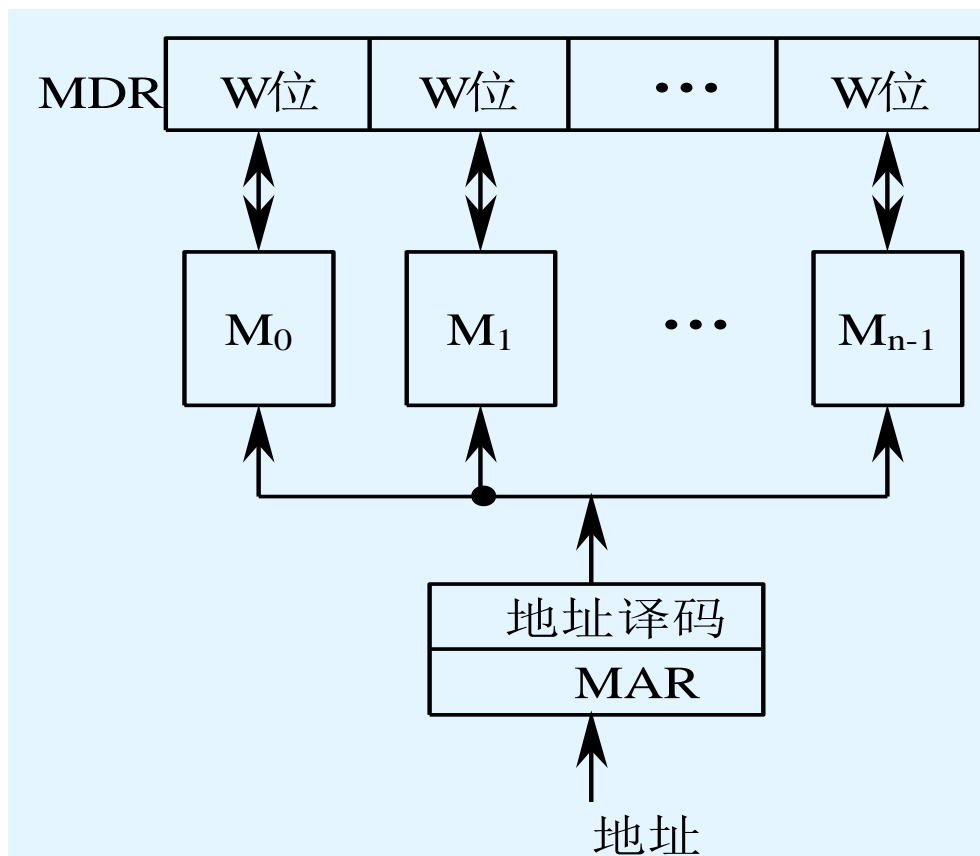
是一种窄通道系统，一次只传输16位数据，速度较宽通道系统快的多。

- 引入了RISC技术，依靠其极高的工作频率，通过减少每个周期的数据量来简化操作。
- Rambus结构的带宽视Rambus通路的个数而定。
- 采用全新设计，需要用RIMM插槽与芯片组配合。
- 行地址与列地址的寻址总线是各自分离的独立总线。

§ 5.6 并行存储器和相联存储器

从存储结构方面来提高内存速度。

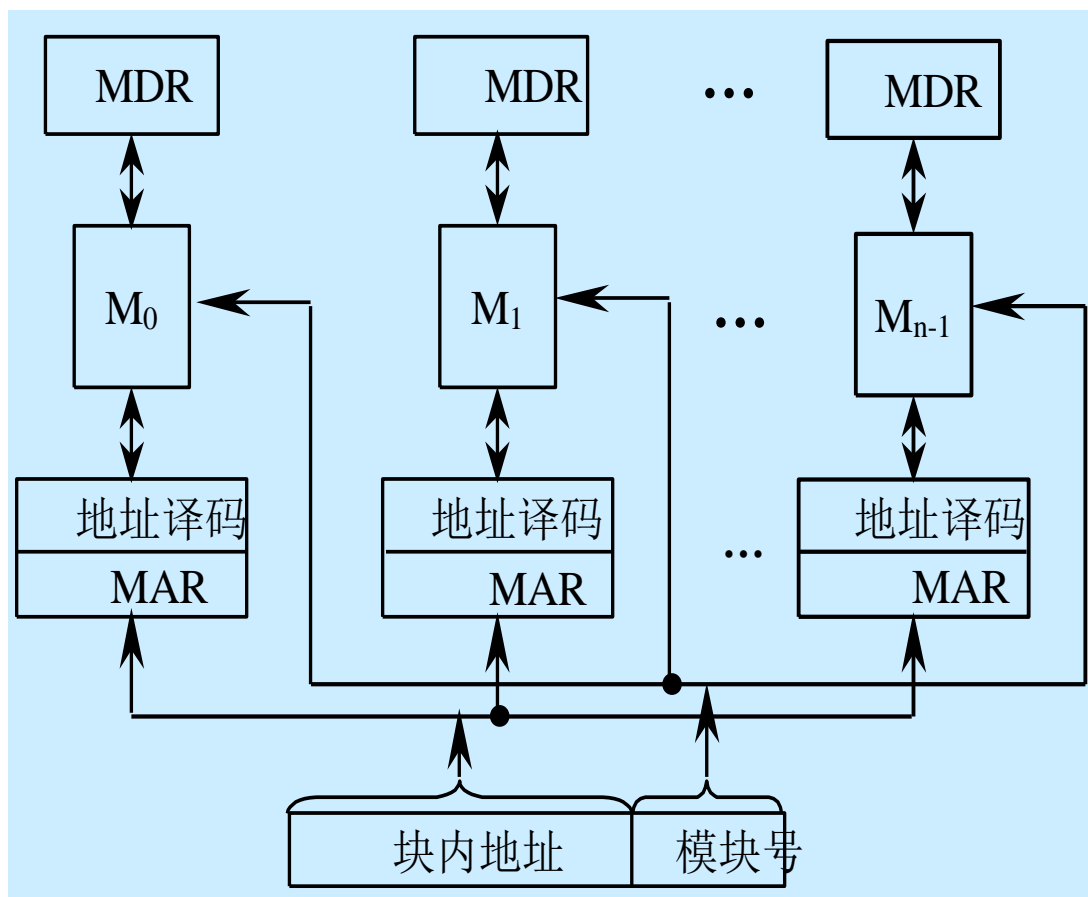
5.6.1 单体多字同时访问方式



单体指只有一套地址寄存器和地址译码器；

多字指有多个容量相同的存储模块。即多个并行工作的存储器**共用一套**地址寄存器和地址译码电路，按同一地址并行地访问各自对应的单元。

5.6.2 多体单字交叉访问方式



多体单字:

指存储体内有多个容量相同的存储模块，且各存储体有自己独立的地址寄存器、译码电路和数据寄存器，任何时间允许对多个存储体独立访问。

5.6.2 多体单字交叉访问方式

以4体为例，其地址编排如下：

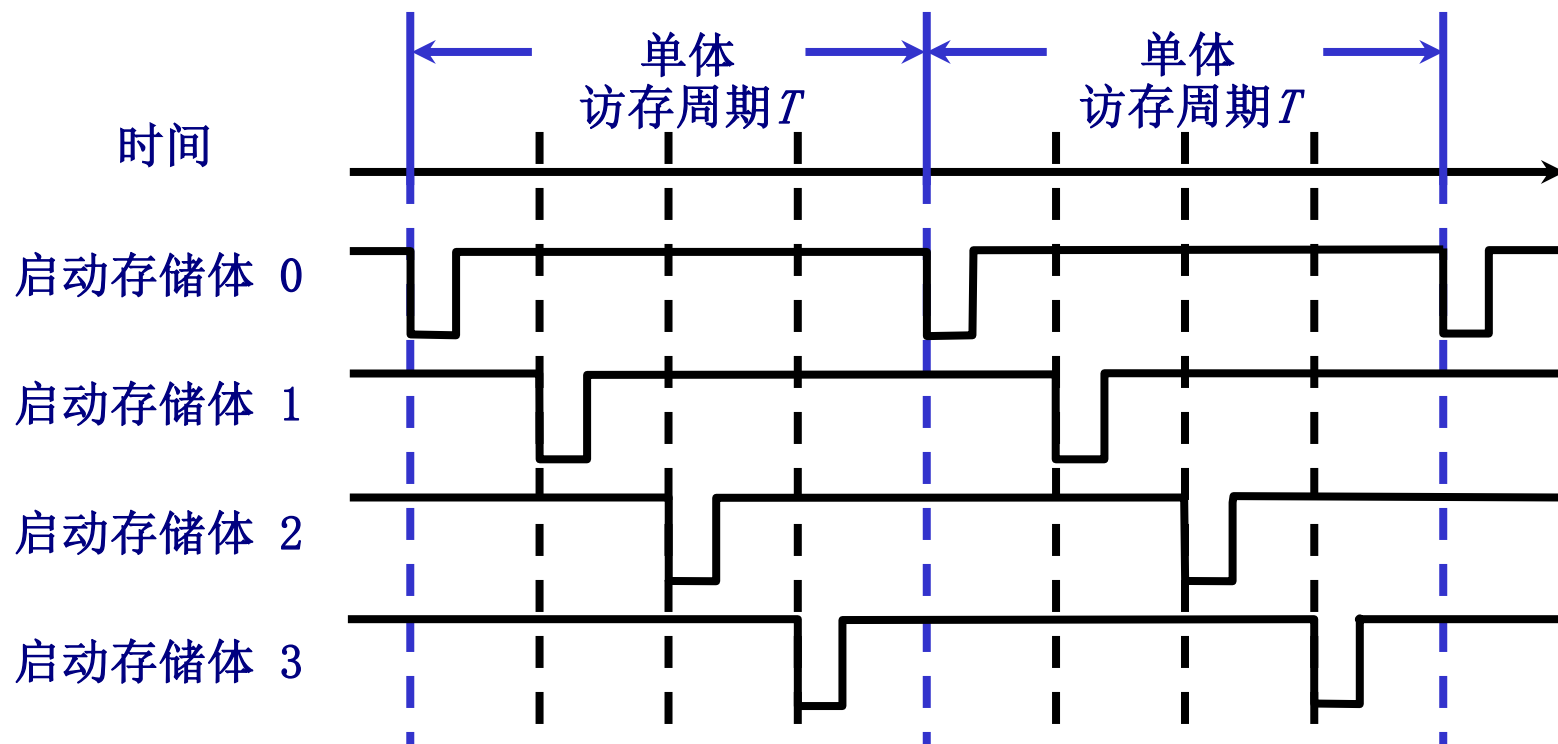
模块号	地址编址序列
M0	0, 4, 8, 12, ..., $4i+0$, ...
M1	1, 5, 9, 13, ..., $4i+1$, ...
M2	地址高位选块内地址, ...
M3	低位选模块号 ...

交叉存取：

指各个模块的存储单元的地址是交叉编排。

5.6.2 多体单字交叉访问方式

分时控制：通常在一个存储器周期 T 内， n 个存储体必须分时启动，则各个存储体的启动间隔为 $t = T/n$



5.6.3 相联存储器

根据内容(或部分内容)，查找其地址及与之相关的内容。

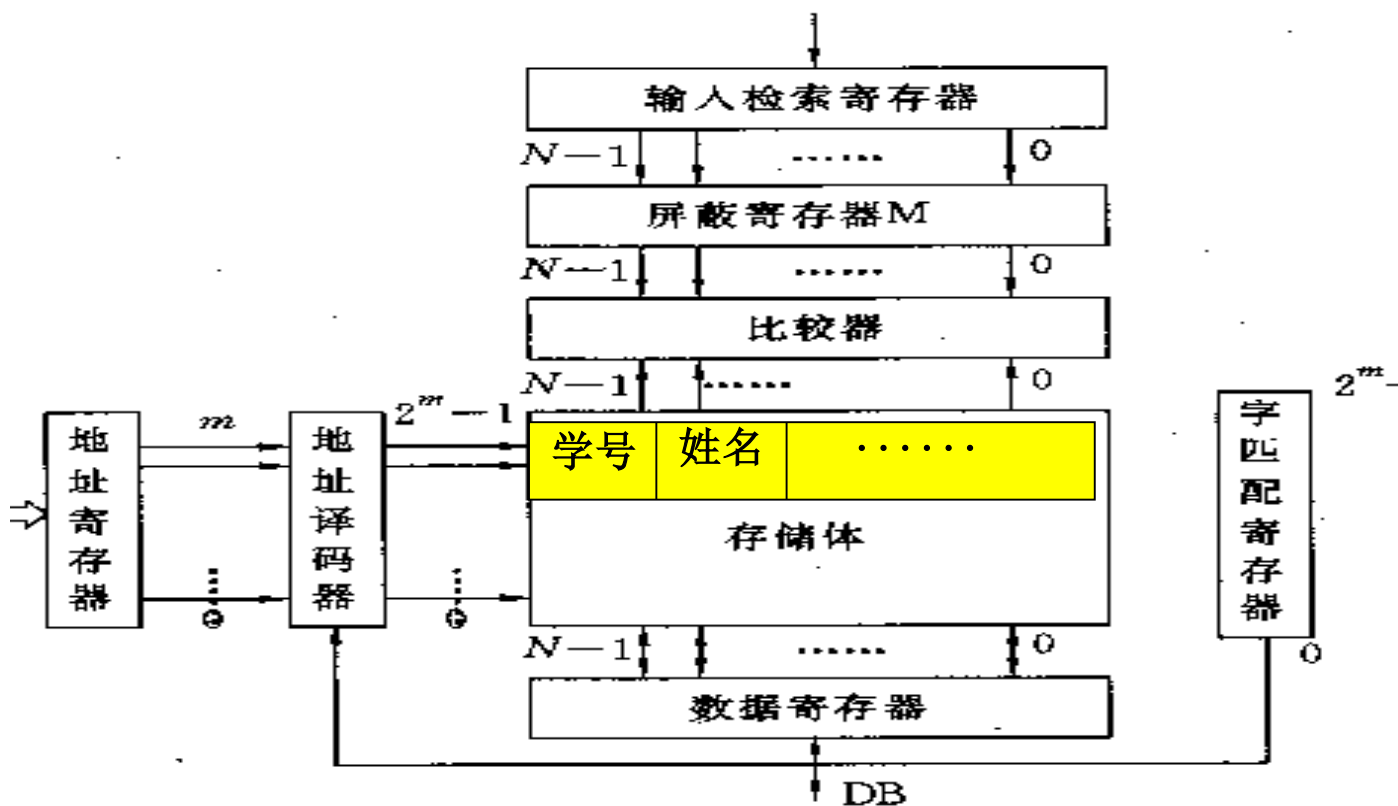
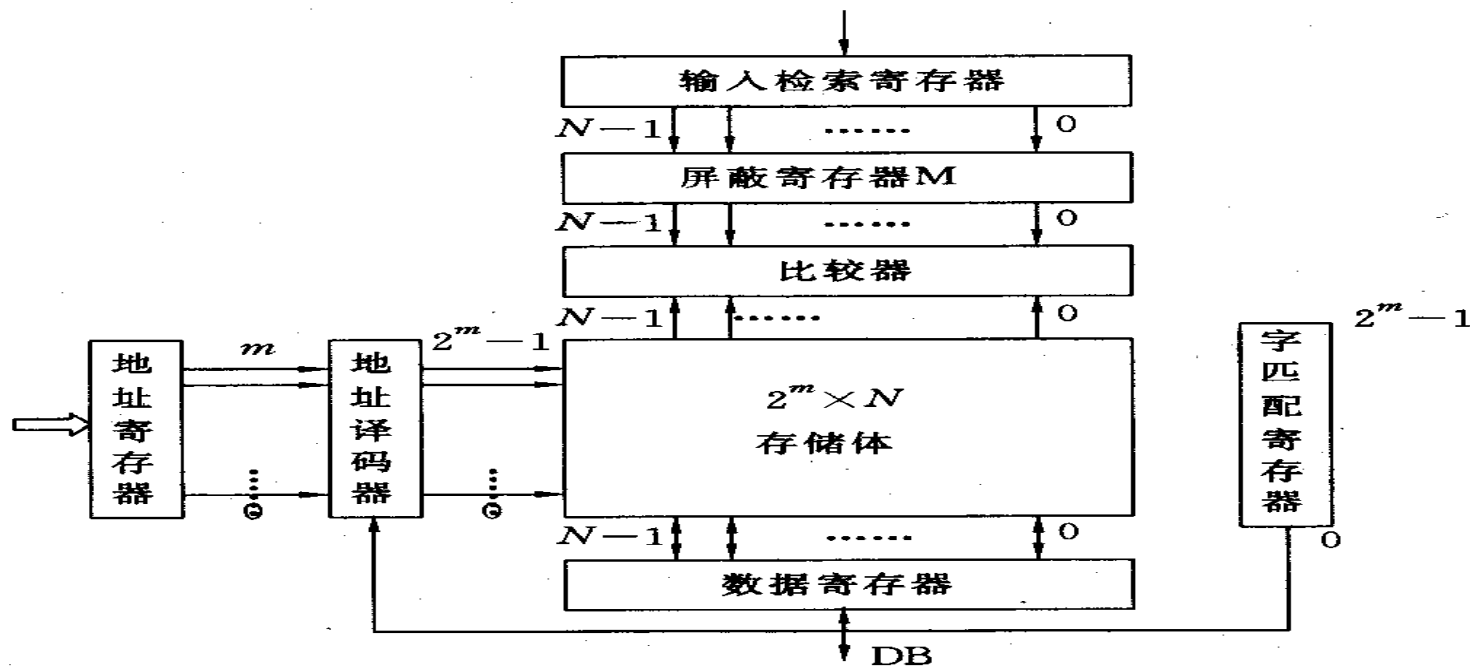
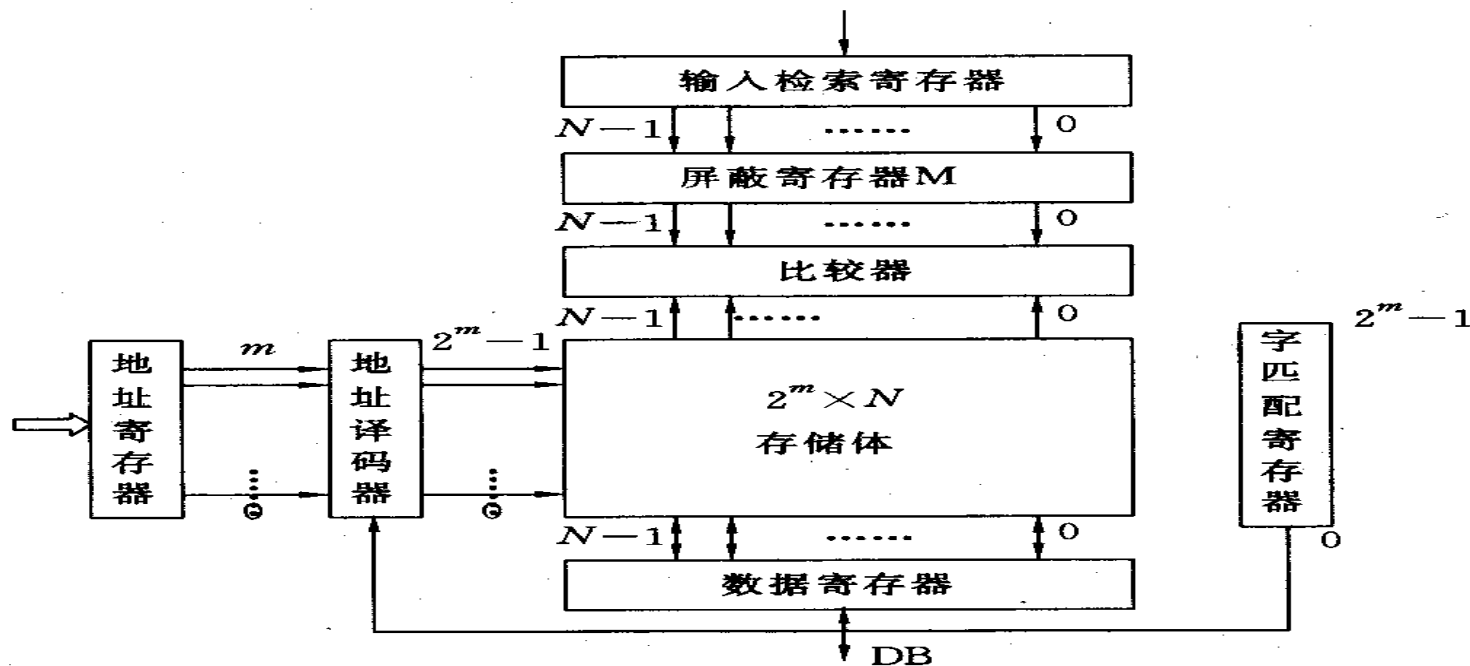


图 2.28 相联存储器的结构图



- 主要包括：
- ① 存储体 2^m 个单元 $\times N$ 位
 - ② 输入检索寄存器
 - ③ 屏蔽寄存器
 - ④ 比较器（与所有单元比较）



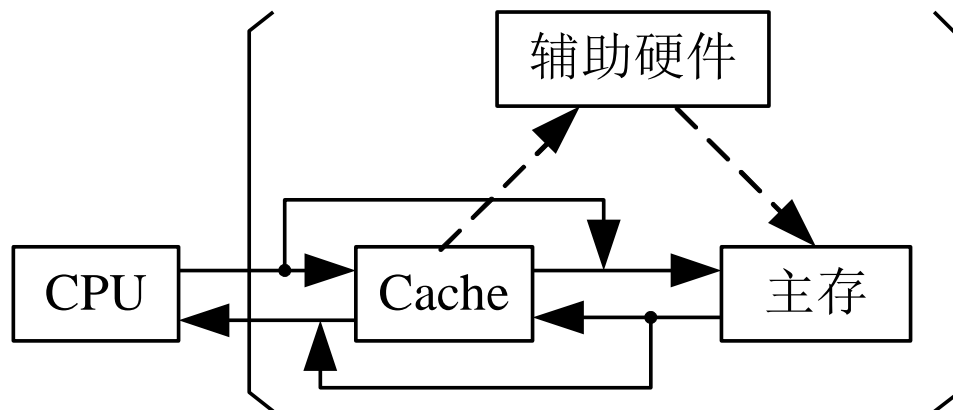
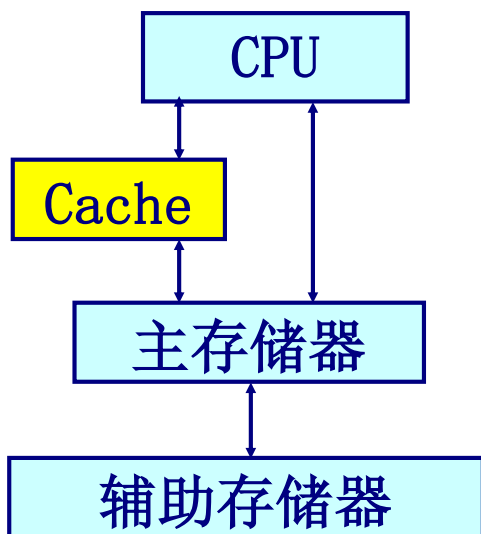
主要包括：

- ⑤ 字匹配寄存器 (2^m 位)
- ⑥ 数据寄存器MDR
- ⑦ 地址寄存器MAR及地址译码

可用于Cache和虚拟存储器的辅助硬设备。

§ 5.7 高速缓冲存储器

高速缓存 (Cache) 保存内存中活跃部分的拷贝。



§ 5.7 高速缓冲存储器

高速缓存(Cache)保存内存中活跃部分的拷贝。

5.7.1 高速缓存工作原理

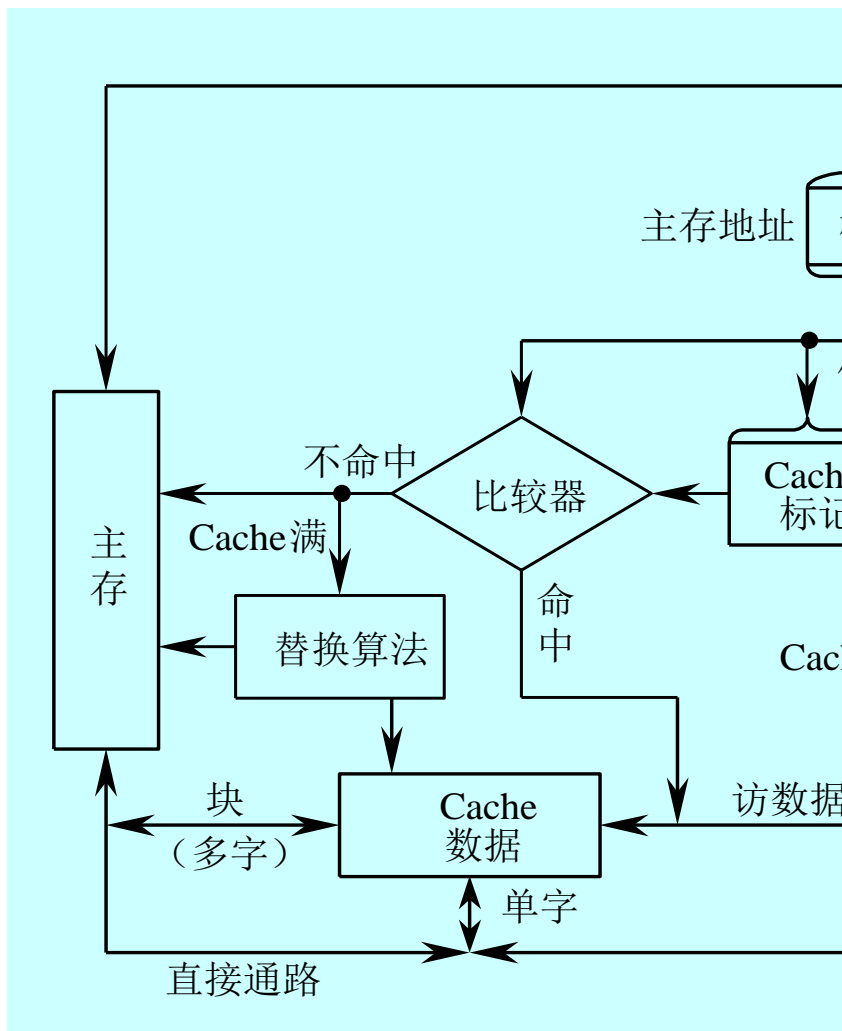
1. 程序的局部性原理

👉 **时间局部性**：如果一个存储单元被访问，则这个存储单元会再次被访问的概率很高。这是由于循环程序的执行，相应的数据要重复访问。

👉 **空间局部性**：如果一个存储单元被访问，则这个存储单元及其相邻单元被访问的概率较高。这是由于程序的顺序执行时，一条指令和下一条指令在存储器中的位置是相邻或相近的。

5.7.1 高速缓存工作原理

2. Cache的基本结构



👉 **基本术语**

● **命中 (Hit):**

CPU发出访问主存操作请求后，所访问的内容已经位于Cache中。

● **失效 (Miss, 不命中):**

CPU发出访问主存操作请求后，如要访问的内容不在Cache中，称为不命中。

5.7.2 Cache的读写操作

1. Cache的读操作

CPU发出读请求后：

i) 读Cache命中

直接对Cache进行读操作，与主存无关。

ii) 读Cache不命中

读主存并把该块信息从主存调入Cache；若Cache已满，则须根据某种替换算法，用这个块替换掉Cache中原来的某块信息。

5.7.2 Cache的读写操作

2. Cache的写操作与更新策略

写操作存在Cache与主存中内容是否一致的问题。

i) 写Cache命中

➤ 写直达法:

CPU写操作时，把数据同时写入主存和Cache。

➤ 写回法:

CPU写操作时，只把数据暂时写入Cache，并用标志将该块注明，等需要将该块替换回到主存时，才写回主存。

5.7.2 Cache的读写操作

2. Cache的写操作与更新策略

写操作存在Cache与主存中内容是否一致的问题。

ii) 写Cache不命中

- 不按写分配法:

只把要写的信息写入主存。

- 按写分配法:

把要写信息写入主存，并把该块从主存中读入Cache。

5.7.3 地址映像和变换

地址映像：把主存地址空间映像到Cache地址空间，即按某种规则把主存的块复制到Cache中。

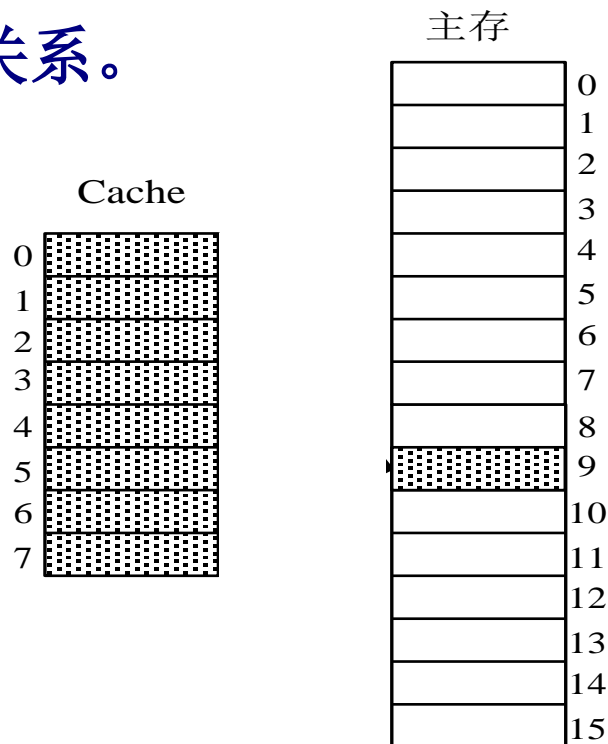
措施：Cache和主存都被分成若干个大小相等的块，每块由若干个字节组成，主存和Cache的数据交换是以块为单位，需要考虑二者地址的逻辑关系。

常用的地址映像有：

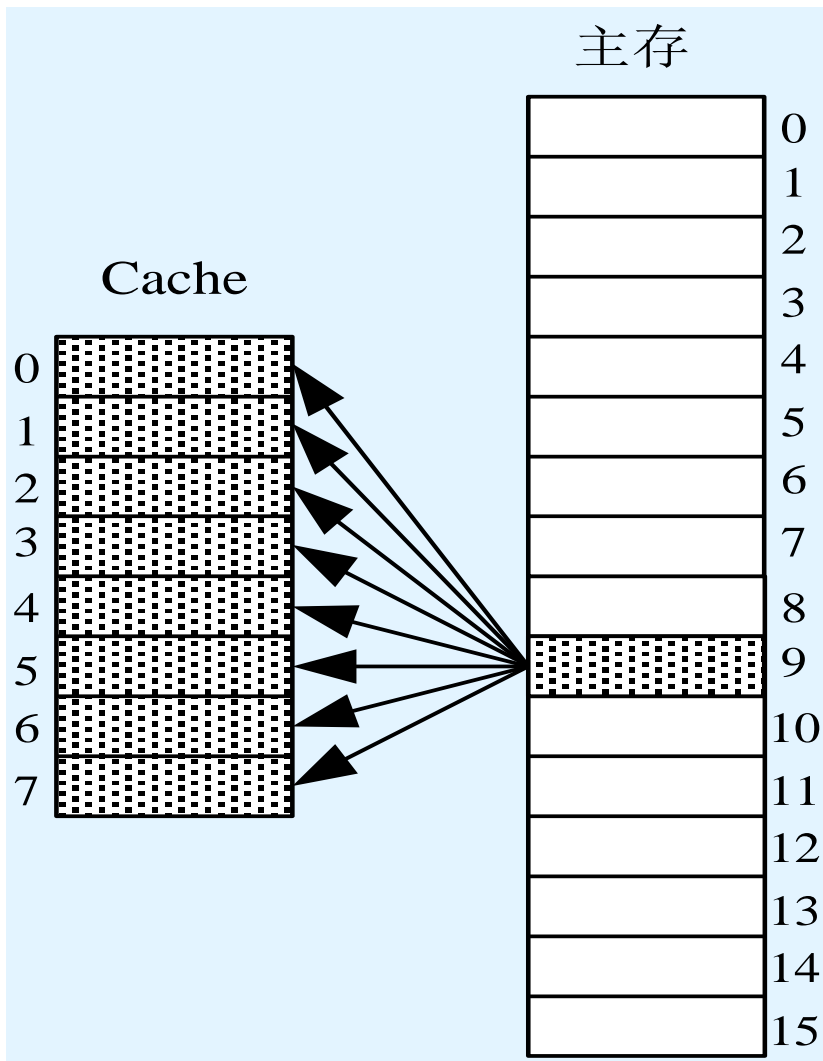
全相联映像

直接映像

组相联映像



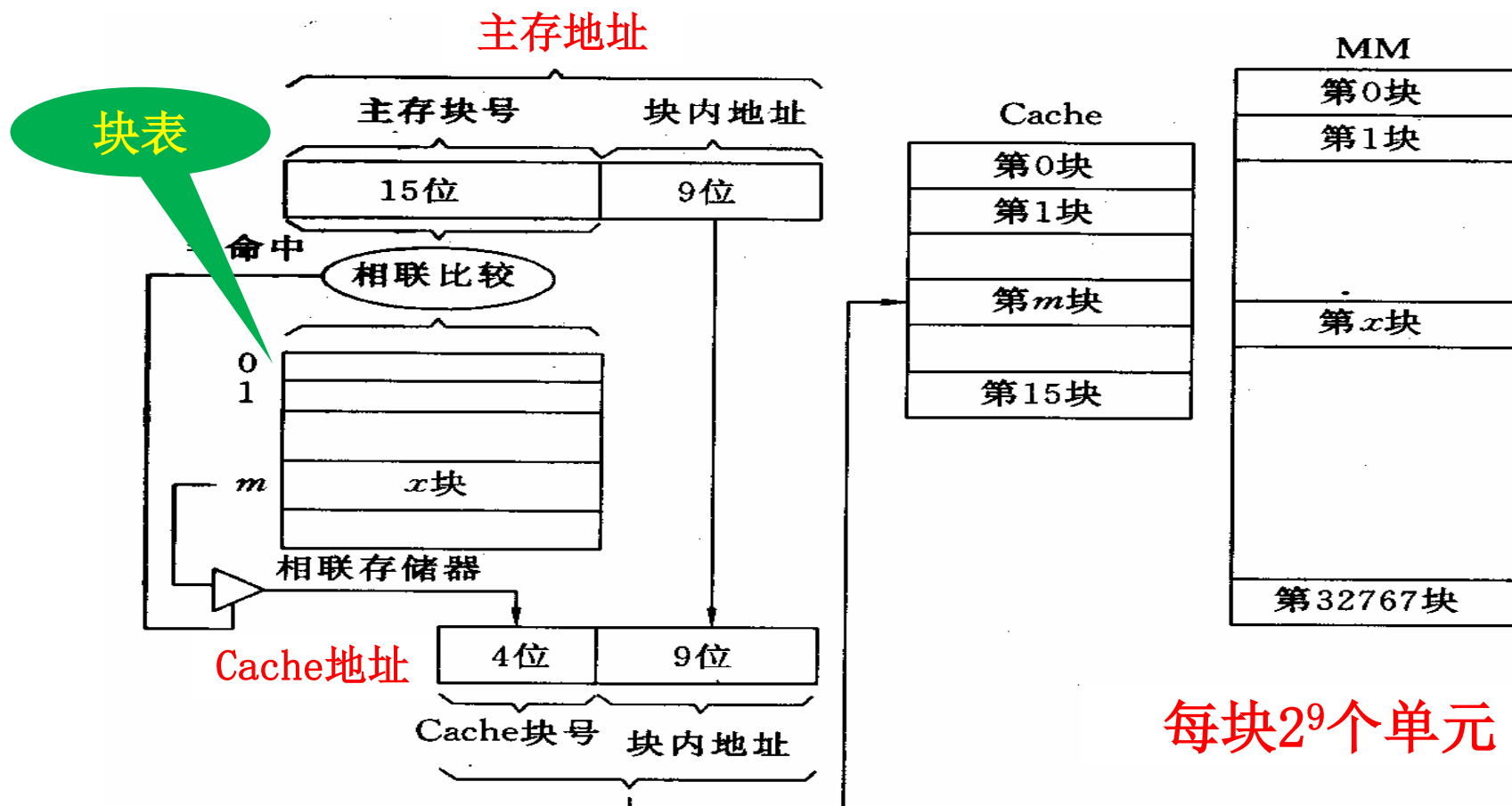
1. 全相联映像



主存中任何一个块均
可以映像装入到Cache中任
何一个块的位置上。

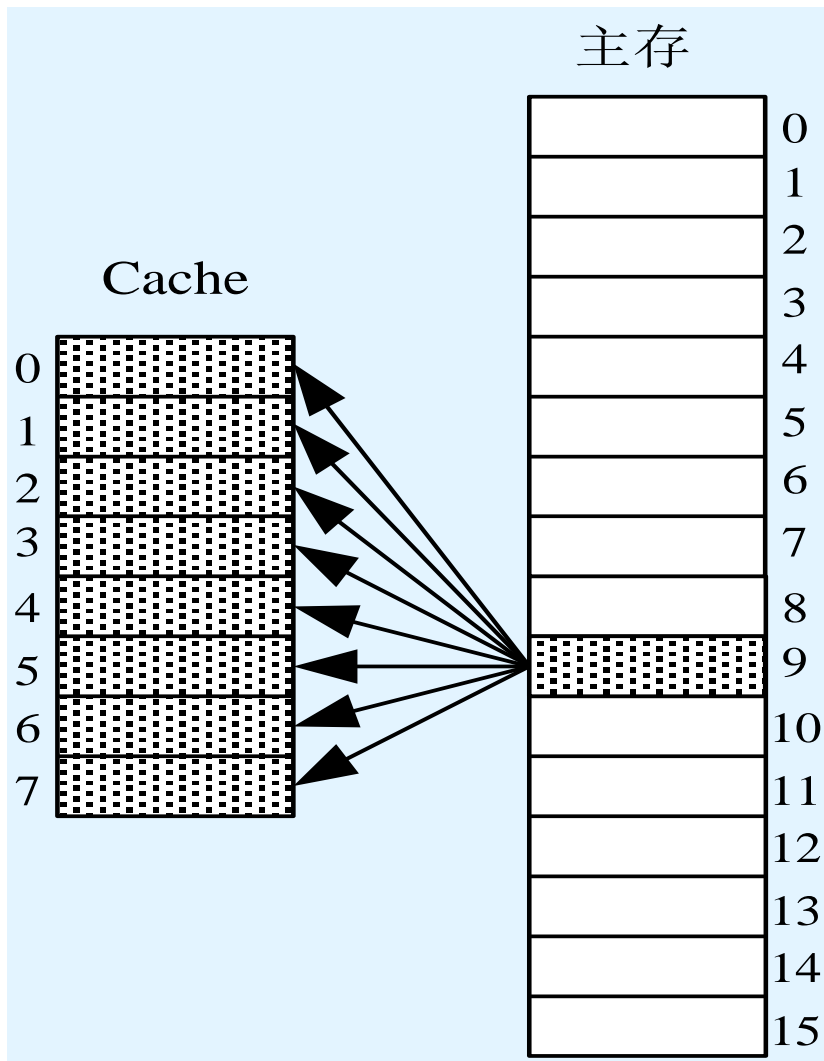
- 主存地址分为两部分：
块号、块内地址。
- Cache地址也分两部分：
块号、块内地址。

1. 全相联映像



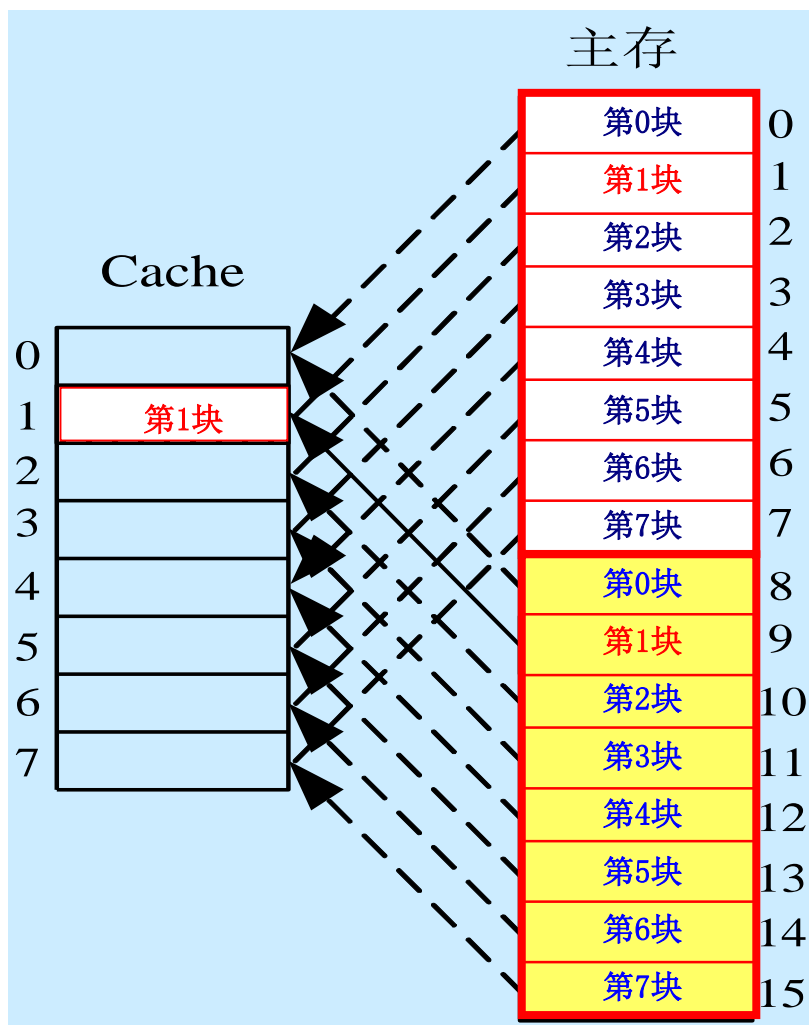
全相联方式地址映像及变换

1. 全相联映像



特点：灵活，块冲突率低，只有在Cache中的块全部装满后才会出现块冲突，Cache利用率高。但地址变换机构复杂，地址变换速度慢，成本高。

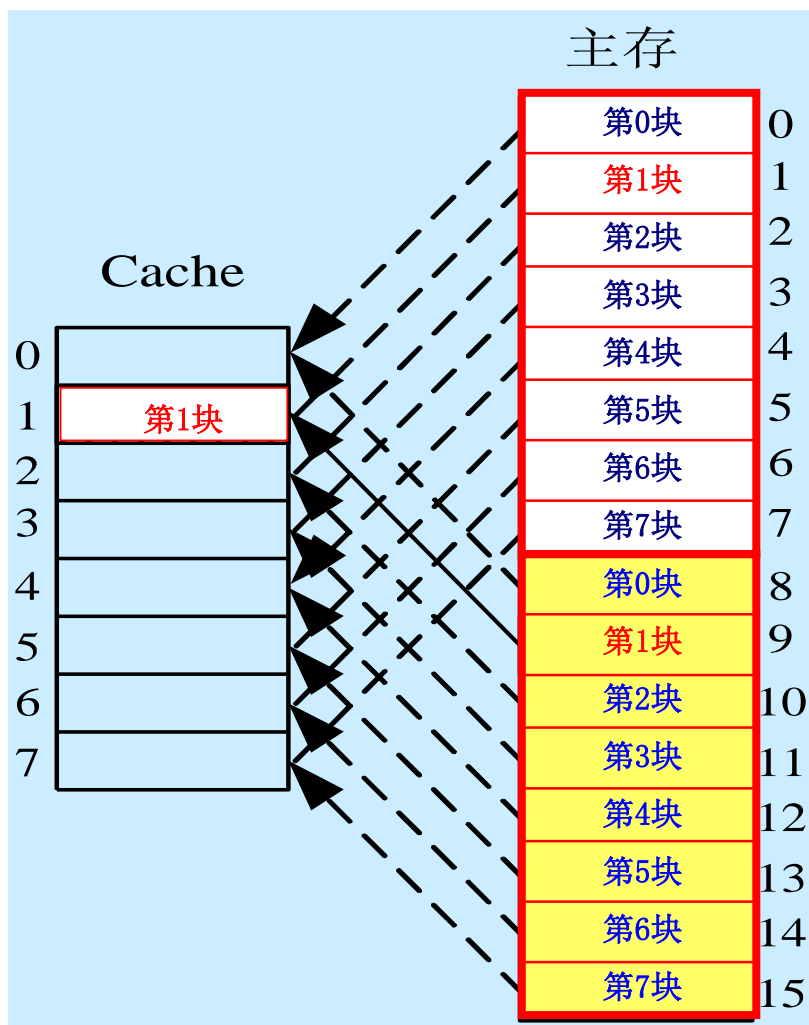
2. 直接映像



把主存分成若干个区，
每个区与Cache大小相同。
区内再分块，其块的个数与
Cache中块的个数相等。

- 主存地址分为三部分：
区号、块号、块内地址。
- Cache地址分为两部分：
块号、块内地址。

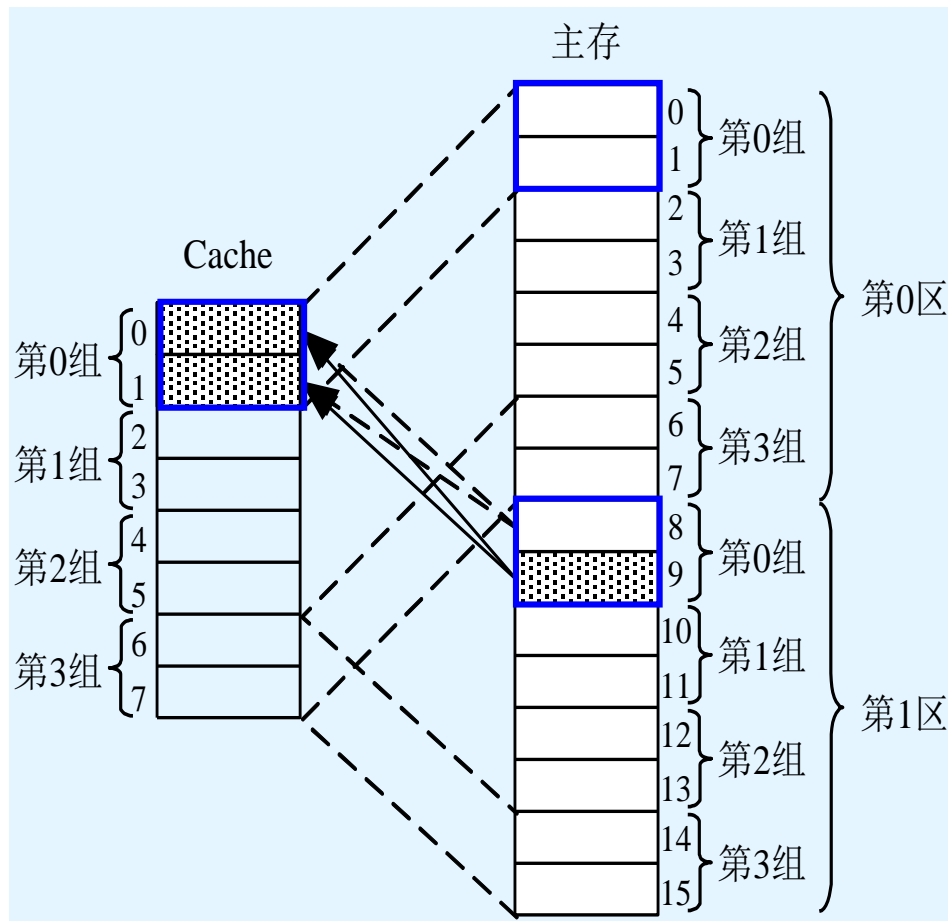
2. 直接映像



特点：地址变换简单、速度快，可直接由主存地址提取出Cache地址。但不灵活，块冲突率较高，Cache空间得不到充分利用。

直接映象方式下，数据块只能映象到Cache中唯一指定的位置，故不存在替换算法的问题。

3. 组相联映像

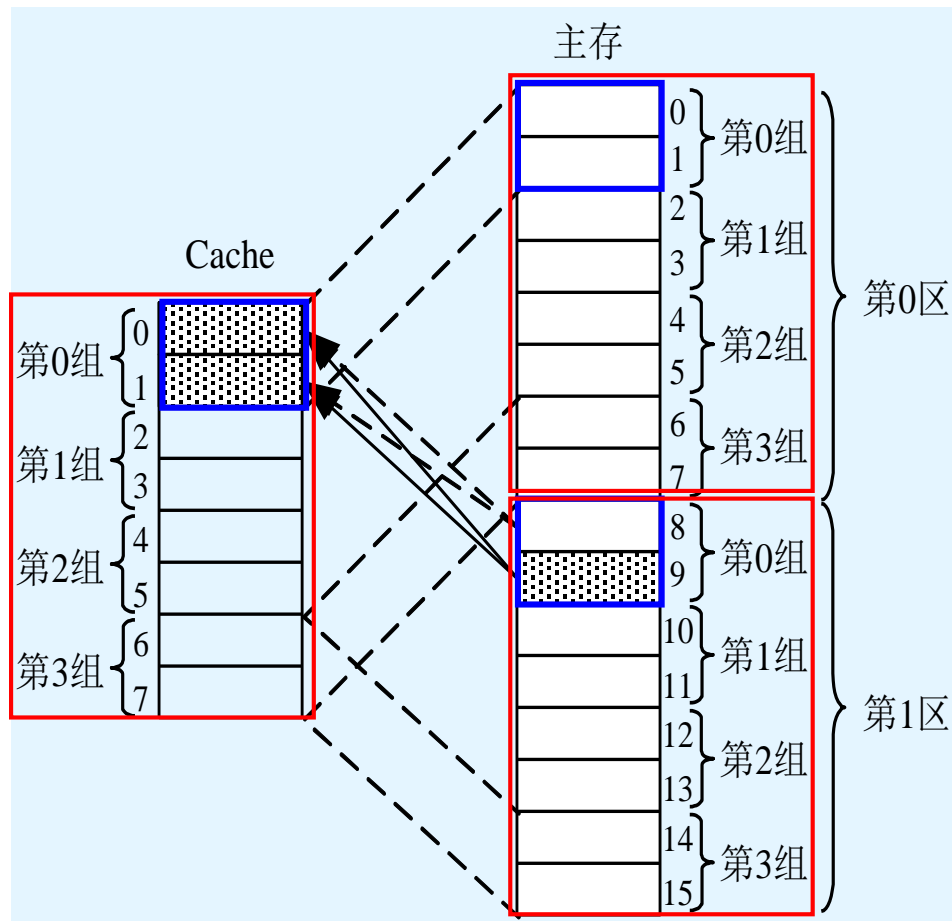


主存按Cache容量分区，每个区分为若干组，每组包含若干块。

Cache也进行同样的分组和分块。

主存中一个组内的块数与Cache中一个组内的块数相等。组间采用直接方式，组内采用全相联方式。

3. 组相联映像



主存按Cache容量分区，每个区分为若干组，每组包含若干块。

- 主存地址分为四部分：
区号、组号、
组内块号、块内地址
- Cache地址分为三部分：
组号、组内块号、
块内地址。

4. 主存地址和Cache地址的相关计算

- ① 主存地址的位数 L ：由主存的存储单元数 N 决定 (设容量 $=N \times B$)
 $L = \log_2 N = \text{区号位数} + \text{块号位数} + \text{块内地址位数}$
- ② Cache地址的位数 P ：由Cache存储单元数 H 决定 (设容量 $=H \times B$)
 $P = \log_2 H = \text{块号位数} + \text{块内地址位数}$
- ③ 区号位数：区号位数 $=$ 主存地址位数 $-$ Cache地址位数
- ④ 块号位数：主存地址的块号和Cache块号的位数相同，位数 K 取决于Cache中能容纳的块的个数 J ， $K = \log_2 J$
- ⑤ 块内地址位数：主存的块内地址和Cache的块内地址长度相同，位数 M 取决于块内的存储单位数 Q ， $M = \log_2 Q$

4. 主存地址和Cache地址的相关计算

【例】设有一个Cache的容量为2K字，每块为16字，求：

- (1) 该Cache可容纳多少个块
- (2) 如果主存容量为256K，则有多少个块？
- (3) 主存的地址有多少位？Cache地址有多少位？
- (4) 直接方式下，主存地址分为哪几部分？每部分有多少位？
主存中第I块映像到Cache中的哪一块？

解：(1) Cache中有 $2048/16=128$ 块
(2) 主存容量为256K，则有 $256K/16=16384$ 块
(3) 主存地址为18位，Cache地址为11位。
(4) 主存地址分为：区号、块号和块内地址。
块内字地址为4位；块号为 $\log_2 128=7$ 位；
区号为 $18-7-4=7$ 位。
主存中第I块映像到Cache中的第 $(I \bmod 128)$ 块中。

课堂练习:

设一个组相联方式的Cache由64个块构成，每组包含4个块。主存包含4096个块，每块由128个字组成。访存地址为字地址。求：

(1) 主存地址有多少位？Cache地址有多少位？

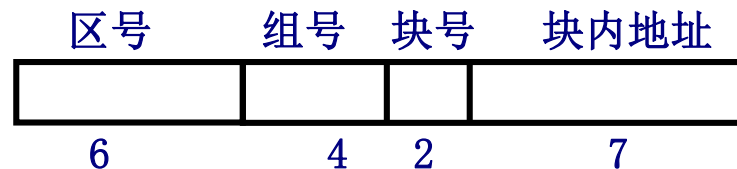
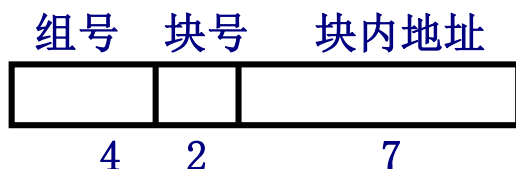
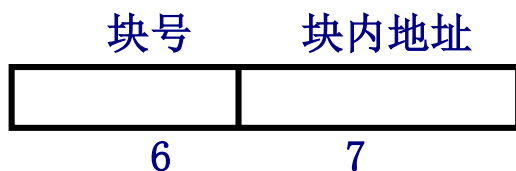
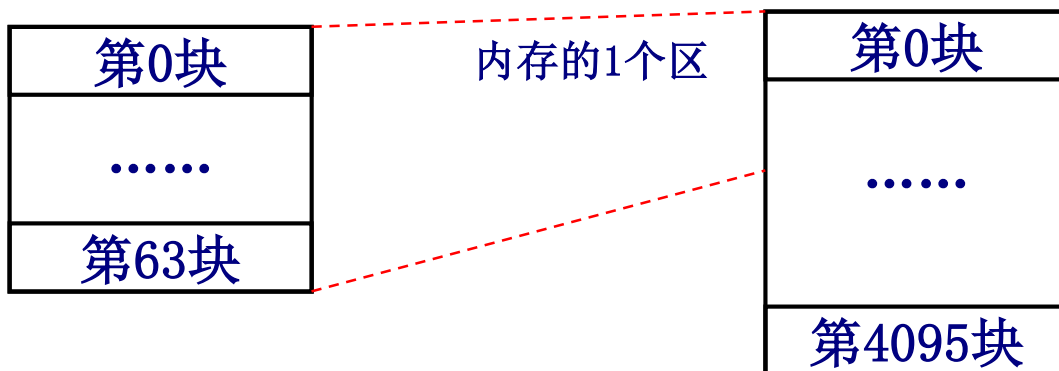
(2) 主存地址格式中，区号、组号、块号和块内地址的位数分别为多少？

课堂练习:

组相联，4块为1组，每块由128个字（按字访存）

Cache: 64块

内存: 4096块



5.7.4 替换算法

(1) 随机算法 (RAND)

随机确定替换单元。可用随机数产生器来产生一个随机的替换单元号。简单，由于没有根据程序访存局部性原理，不能提高系统的命中率。

(2) 先进先出算法 (FIFO)

对进入Cache的块按先后顺序排队，需要替换时，先淘汰最早进入的块。简单，易于实现，也没根据访存局部性原理，命中率较低。

5. 替换算法

(3) 近期最少使用算法 (LRU-Least Recently Used)
将最近使用最少的块替换出去。

能比较正确地利用访存局部性原理，命中率较高，但算法较复杂，系统开销较大。

增加Cache的容量会提高命中率，但两者之间并非成正比关系。块命中率不仅与替换算法有关，还与块容量、块的数量等有关。

5.7.5 PC机中的Cache技术的实现

1. 单一缓存和多级缓存

单一缓存：在CPU和主存之间只设一个Cache。

多级缓存：在CPU芯片内、CPU芯片外扩展以及在主板上设置多级Cache。

➤ 一级Cache (L1 Cache) :

Cache与CPU集成在一个芯片中，以CPU的核心速度运行，速度最快，容量最小。

➤ 二级Cache (L2 Cache):

安装在主板上的Cache，以主板速度运算，容量较大。

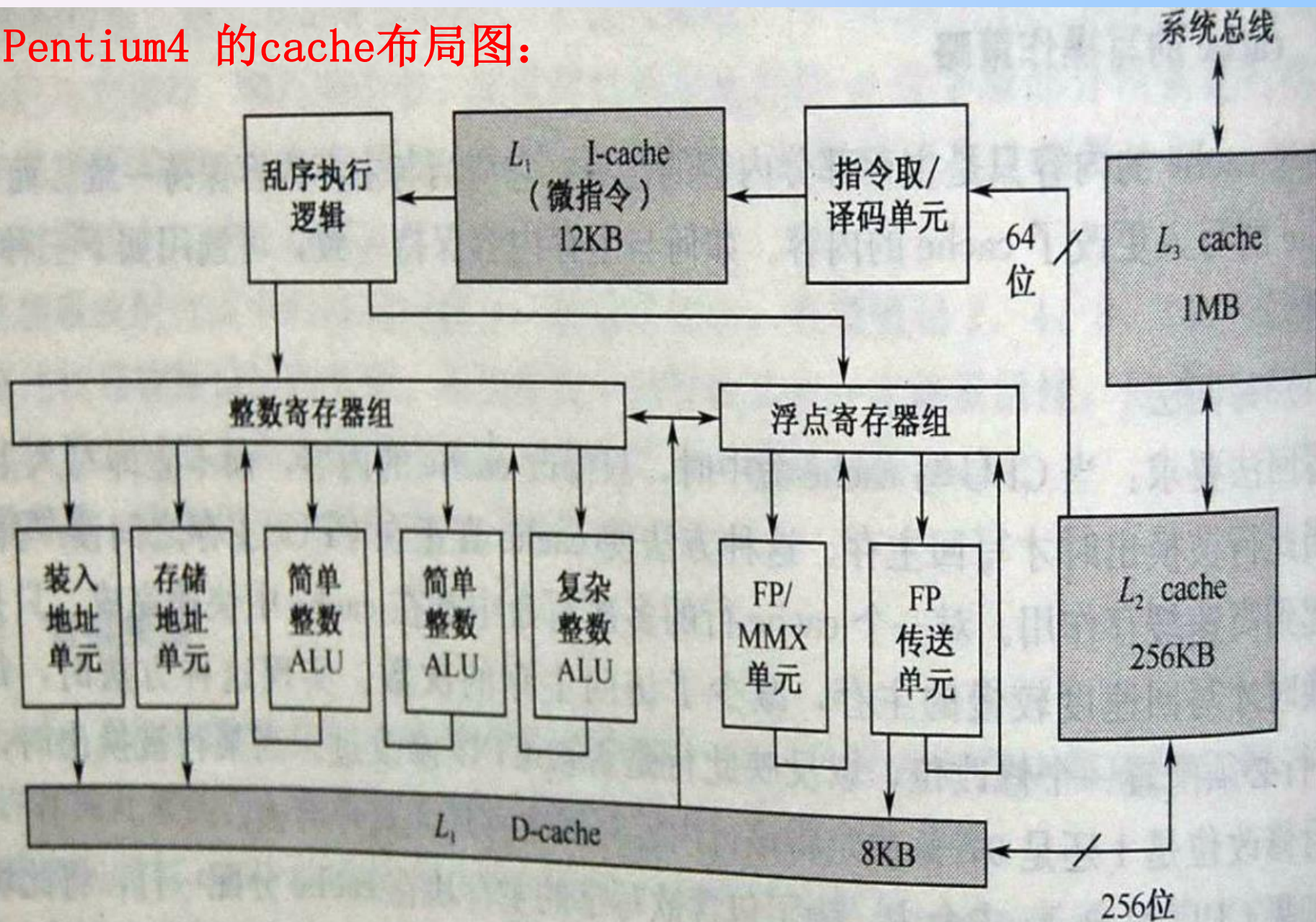
5.7.5 PC机中的Cache技术的实现

2. 统一缓存和分开缓存

统一缓存：指令和数据都存放在同一个Cache中。

分开缓存：指令和数据分别存放在两个Cache中。一个叫指令Cache，一个叫数据Cache

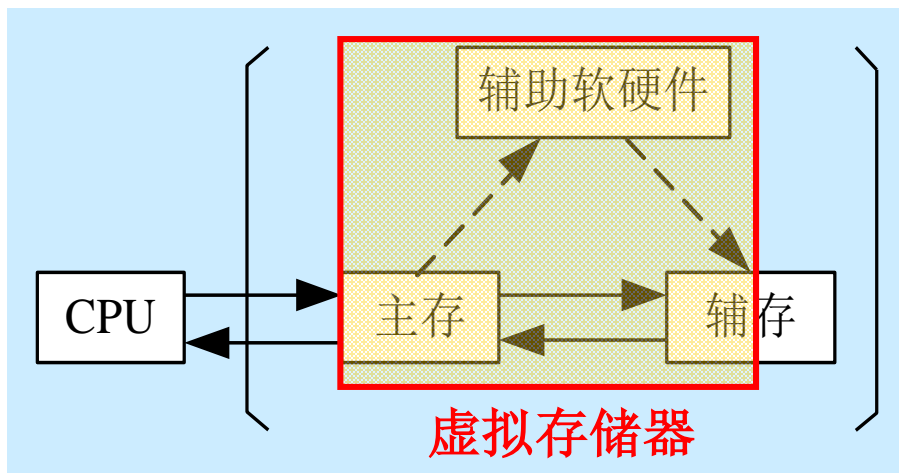
Pentium4 的cache布局图:



§ 5.8 虚拟存储器

5.8.1 虚拟存储器的基本概念

将主存和部分辅存地址空间**统一编址**，形成一个庞大的存储空间。在这个空间里，用户可自由编程，不必考虑程序在主存中的实际存放位置，程序可**像访问内存一样**访问这部分辅存空间。



虚拟存储器是建立在主—辅存层次上，由附加硬件装置和操作系统的存储管理软件组成的存储体系。

§ 5.8 虚拟存储器

5.8.1 虚拟存储器的基本概念

将主存和部分辅存地址空间**统一编址**，形成一个庞大的存储空间。在这个空间里，用户可自由编程，不必考虑程序在主存中的实际存放位置，程序可**像访问内存一样**访问这部分辅存空间。

虚拟地址(逻辑地址)：用户编程的地址。

物理地址(实际地址)：实际主存单元地址。

虚拟地址空间：虚拟地址的范围(程序员可看到的地址空间)。它比实际主存单元数大得多。

5.8.2 虚拟存储器的地址映像

虚拟存储器需要提供动态的地址映像机制，将逻辑地址转换为对应的物理地址。

常见的有：页式虚拟存储器
段式虚拟存储器
段页式虚拟存储器

5.8.2 虚拟存储器的地址映像

1. 页式虚拟存储器

将主存和辅存空间都分成大小相同的页。

虚地址 = 虚页号 + 页内地址；

实地址 = 实页号 + 页内地址

虚地址和实地址中的页内地址相同。

不考虑程序的逻辑功能，面向存储器物理结构。

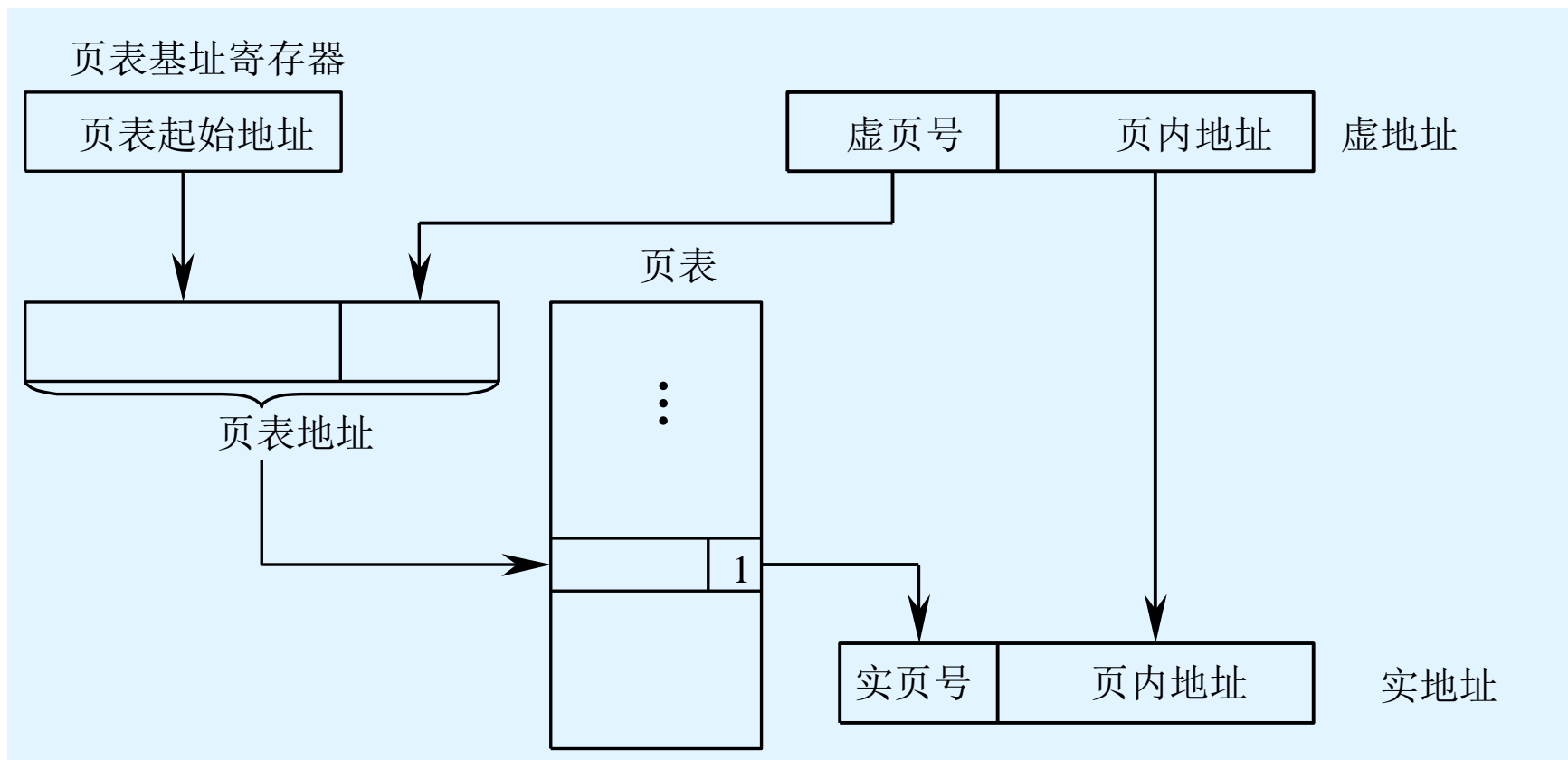
特点

- 页长固定，页表的建立方便，页的调入调出容易实现；
- 当存储空间较大时，页表占的空间将很大，效率降低；
- 页不是逻辑上独立的实体，使程序的处理、保护和共享较困难。

5.8.2 虚拟存储器的地址映像

1. 页式虚拟存储器

虚地址与实地址的映像：



5.8.2 虚拟存储器的地址映像

1. 页式虚拟存储器

【例1】 一个有32位程序地址空间，页面容量为1KB，主存容量为8MB的存储系统，采用页式管理，问：虚页号字段有多少位？页表有多少行？

解：

因为页面容量为1KB，故页内地址字段为10位。

虚页号字段 = $32 - 10 = 22$ 位，页表的长度为 $2^{22} = 4\text{M}$ 行。

5.8.2 虚拟存储器的地址映像

1. 页式虚拟存储器

【思考题】某计算机的页式虚拟管理中采用长度为32字的页面，页表内容如表所示，求按下列二进制虚字地址访存时产生的实际字地址。（1）00001101 （2）10000000 （3）00101000

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

5.8.2 虚拟存储器的地址映像

2. 段式虚拟存储器

将程序按逻辑功能分**段**，各段大小不等，逻辑地址均从0开始，装入时按段分别装入主存，运行时按段进行虚—实地址转换。

虚地址 = 虚段号 + 段内地址

实地址 = 实存段首地址 + 段内地址

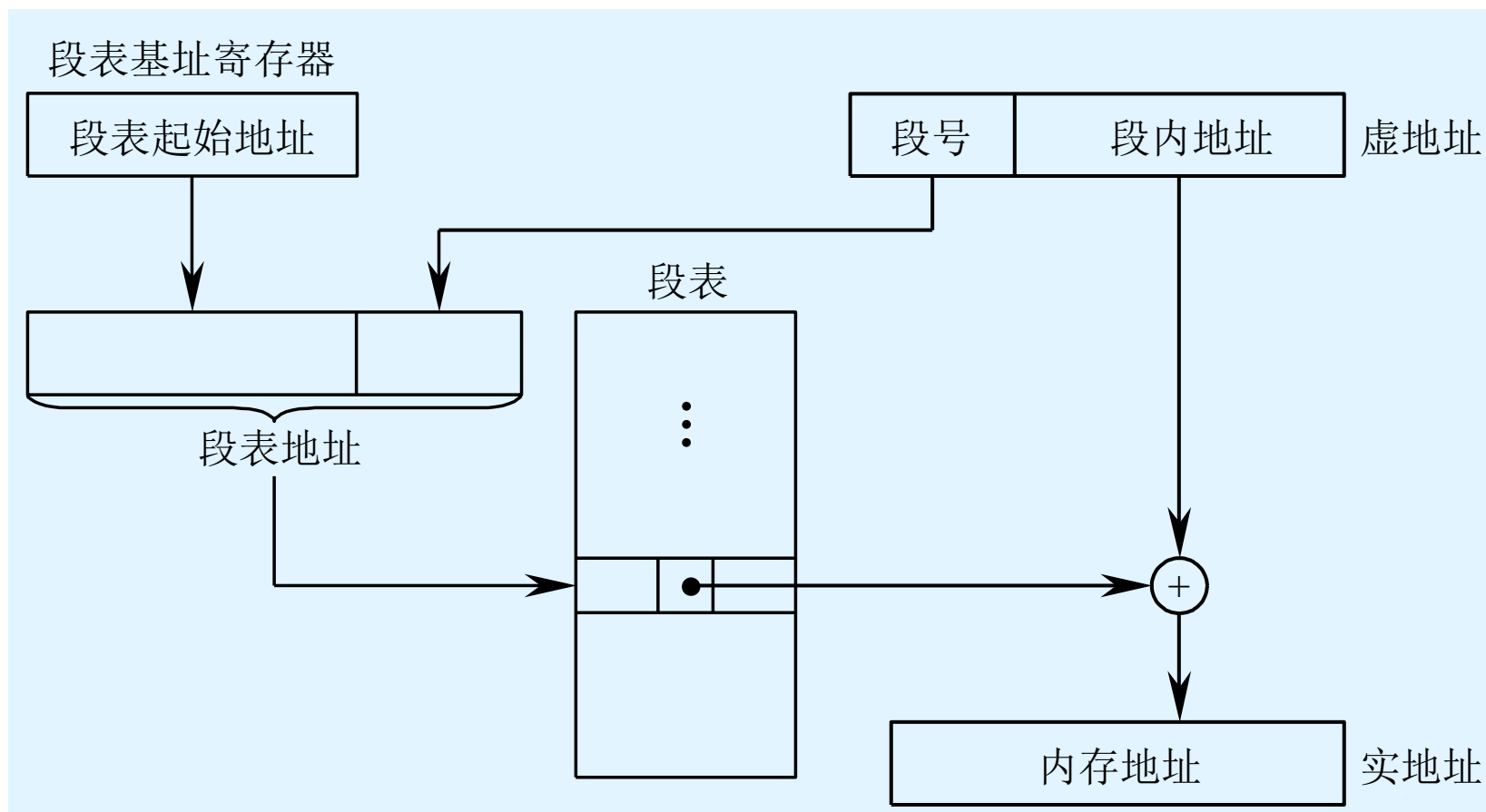


特点

- 用户地址空间分离，段表占用空间少，管理简单；
- 段具有逻辑独立性，易于实现程序的编译、管理和保护，也便于多道程序共享。
- 随着程序的不断运行，会在主存空间中形成较多碎片。

5.8.2 虚拟存储器的地址映像

2. 段式虚拟存储器



5.8.2 虚拟存储器的地址映像

3. 段页式虚拟存储器

是段式管理和页式管理的结合。将存储空间先按逻辑功能分成段，每段又分成若干页。

虚地址 = 虚段号 + 段内虚页号 + 页内地址；

实地址 = 实页号 + 页内地址。

虚地址和实地址中的页内地址相同。



特点

- 兼有段式管理和页式管理二者的优点，但地址变换的速度较慢。目前大多数计算机系统都采用段页式管理；
- 段的起点不能是任意的，必须是主存中**某页的起点**；
- 页表的个数与段数相同，即每个段都有自己的页表。

5.8.2 虚拟存储器的地址映像

3. 段页式虚拟存储器

地址映像的实现：

- (1) 首先由段表基址寄存器中取出段表首地址；
- (2) 根据段表首地址访问段表，得到虚段号对应的页表在内存中的首地址；
- (3) 根据页表首地址访问页表，判断段内虚页号对应页是否已经装入内存；
- (4) 如该页已经装入内存，则从页表中取出对应的实页号，和虚地址中的页内地址相加，得到实地址。

5.8.3 快表与慢表

根据程序局部性规律，一段时间内对页表的访问局限在少数几个存储器字内，故可将页表分为快表和慢表两种。

- **快表：**将当前最常用的页表信息存放在一个小容量的高速存储器中，称为快表。
- **慢表：**存放在主存中的页表。

快表是慢表的一个副本，二者构成了一个由两级存储器组成的存储系统，其访问速度接近于快表，存储容量接近于慢表。

5.8.4 虚拟存储器与Cache管理方式上的区别

- Cache的替换策略是由**硬件**控制的；
虚拟存储器的替换策略是由**操作系统**控制的。
- Cache的存在及其所有操作对程序员都是**透明**的；
虚拟存储器中页面对系统程序员是不透明，对用户是透明的；段对用户可透明也可不透明。

思考题: P184 1-10, 23, 24, 26

习题: P185 11, 12, 13, 16, 17, 18, 25, 27

思考题: P164 1, 2, 5-10, 23, 24, 26

习题: P164 3, 4, 11, 12, 13, 16, 17, 18, 25, 27



§ 5.9 磁表面存储原理和光记录原理

5.9.1 磁表面记录的技术参数

道密度：垂直于磁道方向上单位长度中的磁道数目。

“道/mm”或“道/英寸”

位密度：磁道方向上单位长度上所能记录的二进制位数。

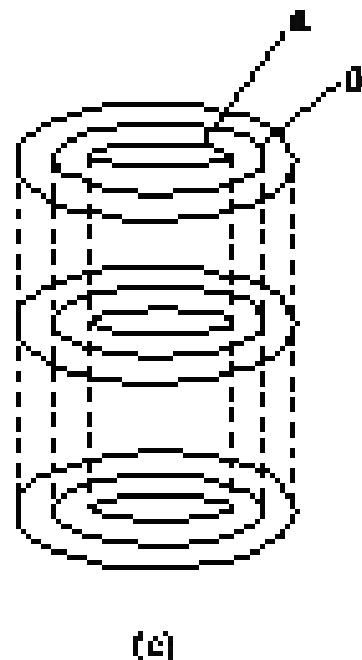
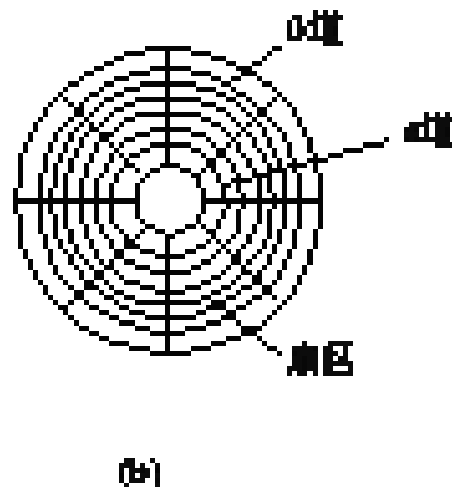
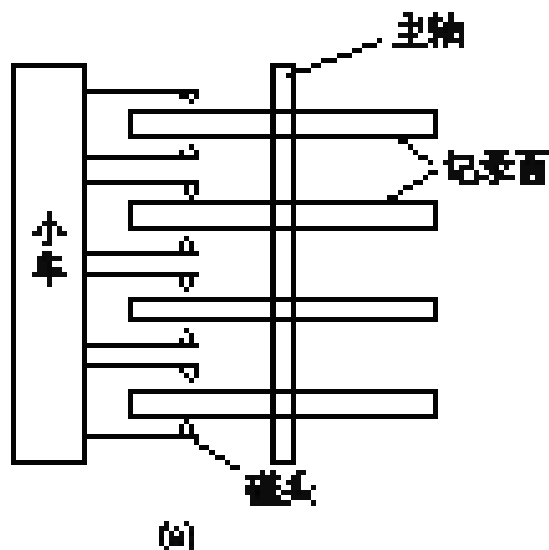
“位/mm”或“位/英寸”

面密度：等于道密度和位密度的乘积

§ 5.9 磁表面存储原理和光记录原理

5.9.2 硬盘的组成结构和信息分布

硬盘存储设备由磁记录介质、磁盘控制器和磁盘驱动器组成。磁盘地址一般表示为：驱动器号、圆柱面号、记录面号、扇区号。



5.9.2 硬盘的组成结构和信息分布

硬盘技术参数:

硬盘容量;

主轴转速: 理论上转速越快, 存取速度越快;

平均存取时间: 约为平均寻道时间+平均等待时间;

缓存大小: 用于硬盘内部与接口数据之间速度的匹配;

数据传输率: 分为内部数据传输率和外部数据传输率;
道密度。

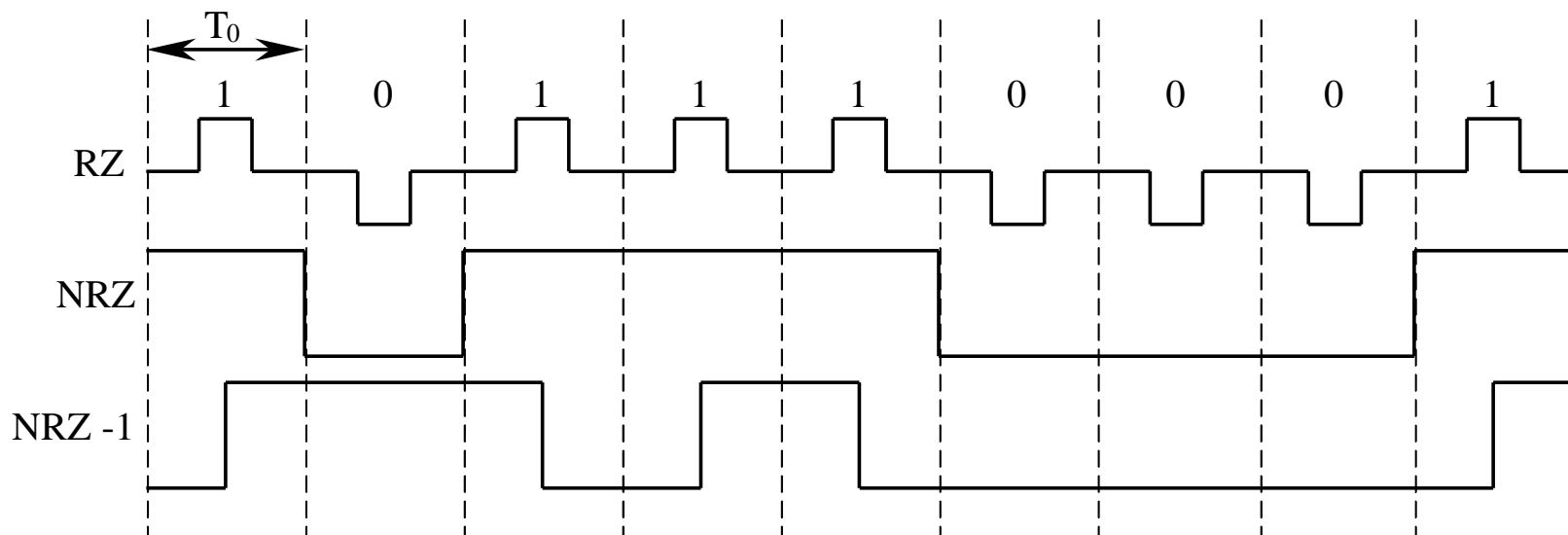
5.9.3 数字磁记录方式

1. 直接记录方式

(1) 归零制 (RZ)

(2) 不归零制 (NRZ)

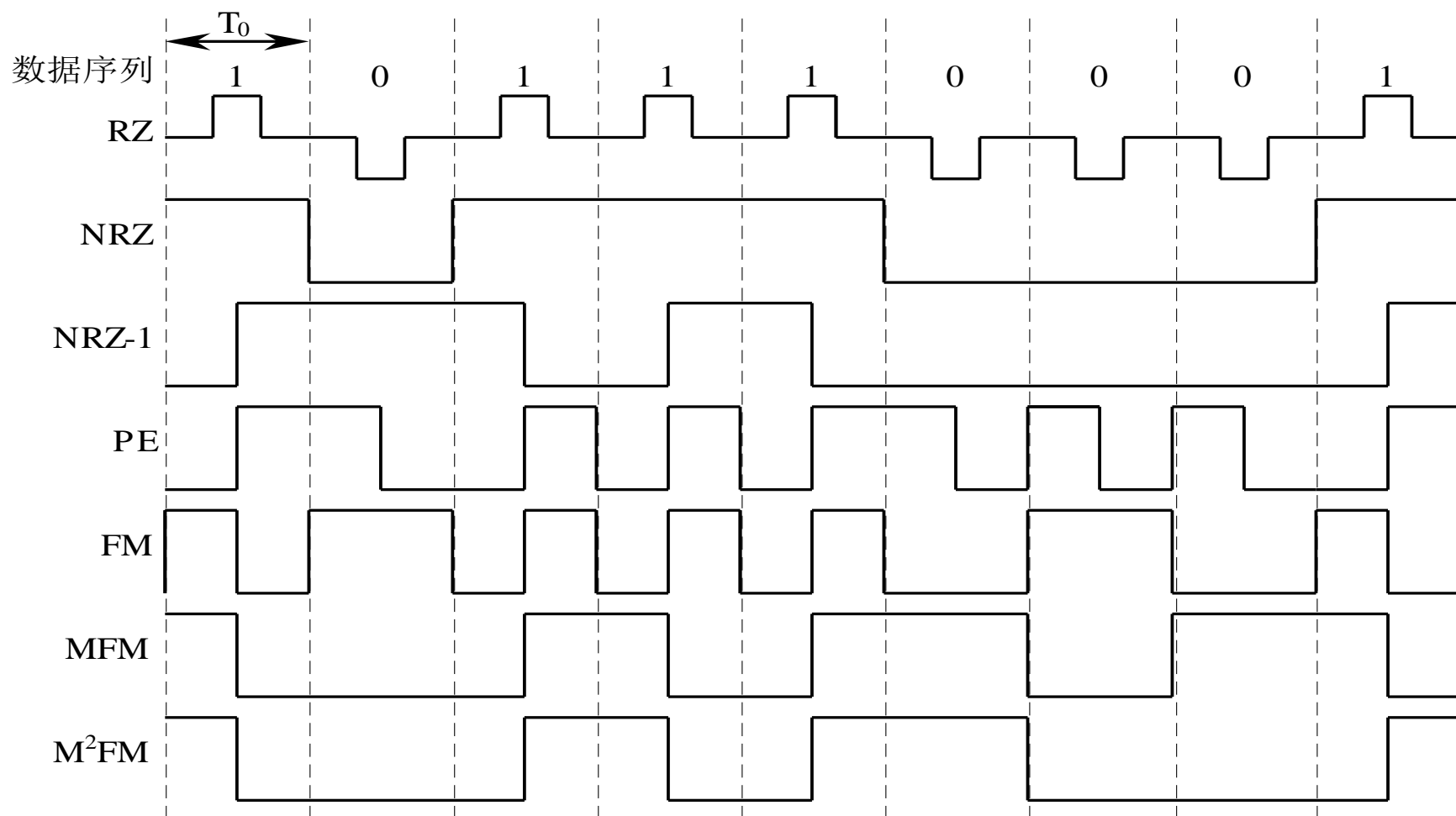
(3) 不归零-1制 (NRZ-1) 见1就翻



5.9.3 数字磁记录方式

2. 按位编码记录方式

- (1) 调相制 (PE)
- (2) 调频制 (FM)
- (3) 改进的调频制 (MFM)
- (4) 改进的改进型调频制 (M^2FM)



MFM: 记录“1”时，写电流在位周期中间改变方向；记录一个“0”时，写电流不改变方向；记录连续的两个“0”时，写电流在位周期边界改变方向。磁通翻转间距有3种： T_0 、 $1.5T_0$ 、 $2T_0$ 。

M²FM: 记录“1”时，写电流在位周期中间改变方向；记录一个“0”时，写电流不改变方向；记录连续的两个“0”时，写电流在第二个“0”起始的位周期边界处改变方向；记录连续多个“0”时，写电流在前两个“0”的位周期边界处改变方向；以后每隔两个“0”在位周期边界处写电流再改变一次方向。磁通翻转间距有4种： T_0 、 $1.5T_0$ 、 $2T_0$ 、 $2.5T_0$

5.9.3 数字磁记录方式

3. 成组编码记录方式

包括群码制（GCR）、三位调制码（3PM）和游程长度受限码（RLL）等。

将数据序列中的数据位几位分成一组，然后按一定的变换规则变换成对应的记录码，再采用NRZ-1制写入记录介质，以提高记录密度。

硬盘中最流行的编码方式为RLL码，其记录密度是调频制的3倍。

游程长度受限源于两个主要特性：两个实际的磁通转换之间允许的最小转换单元数目（游程长度）和最大的转换单元数目（游程受限）。

5.9.4 光记录原理

光盘存储器由光盘控制器、光盘驱动器及接口组成。轨迹是螺旋形（磁盘的磁道是多个同心圆），线速度恒定。

CD-ROM靠盘面上有无凹痕的不同反射率来读出数据。

CD-R光盘的写入是利用聚焦成 $1\mu\text{m}$ 左右的激光束的热能，使记录介质表面的形状发生永久性变化而完成的。只能写入一次，不能抹除和改写。

CD-RW利用激光照射引起记录介质的可逆性物理变化来进行读写。

DVD每面可以有二层用来刻录数据。

思考题： P184 1-10, 23, 24, 26

习题： P185 11, 12, 13, 16, 17, 18, 25, 27

思考题： P164 1, 2, 5-10, 23, 24, 26

习题： P164 3, 4, 11, 12, 13, 16, 17, 18, 25, 27



一、单选题

1. 计算机的存储器采用分级存储体系的主要目的是_____。
A. 便于读写数据 B. 减小机箱的体积
C. 便于系统升级 D. 解决存储容量、价格和存取速度的矛盾
2. 在多级存储体系中,“cache——主存”结构的作用是解决_____的问题。
A. 主存容量不足 B. 主存与辅存速度不匹配
C. 辅存与CPU速度不匹配 D. 主存与CPU速度不匹配
3. 和外存储器相比,内存储器的特点是_____。
A. 容量大、速度快、成本低 B. 容量大、速度慢、成本高
C. 容量小、速度快、成本高 D. 容量小、速度快、成本低
4. 下列器件中存取速度最快的是_____。
A. cache B. 寄存器 C. 内存 D. 外存
5. 静态半导体存储器SRAM_____。
A. 不需要动态刷新 B. 芯片内部已有自动刷新逻辑
C. 断电后仍能保存内容不变 D. 在工作过程中,存储内容静止不变

6. ROM与RAM的主要区别是_____。
- A. 断电后, ROM内保存的信息会丢失, RAM则可以长期保存不会丢失
 - B. 断电后, RAM内保存的信息会丢失, ROM则可以长期保存不会丢失
 - C. ROM是外存储器, RAM是内存储器
 - D. ROM是内存储器, RAM是外存储器
7. 用RAM芯片作字扩展时可以_____。
- A. 增加存储器字长
 - B. 增加存储单元数量
 - C. 提高存储器的速度
 - D. 降低存储器的平均价格
8. 组成 $2\text{M}\times 8\text{bit}$ 的内存, 可以使用_____进行位扩展。
- A. $1\text{M}\times 8\text{bit}$ 的芯片
 - B. $1\text{M}\times 4\text{bit}$ 的芯片
 - C. $1\text{M}\times 1\text{bit}$ 的芯片
 - D. $2\text{M}\times 4\text{bit}$ 的芯片
9. 在程序的执行过程中, Cache与主存的地址映射是由_____。
- A. 操作系统来管理
 - B. 程序员来管理
 - C. 硬件来自动完成
 - D. 软硬件共同完成
10. 主存中的每一块只能复制到某一固定的CACHE块中, 这种映象关系称为_____。
- A. 地址映象
 - B. 直接映象
 - C. 全相联映象
 - D. 组相联映象

二、判断题，如果判断是错误的，请给出正确的描述：

1. 随机存取存储器是指既可以读出也可以写入的存储器。
2. 动态RAM和静态RAM都是易失性半导体存储器。
3. 动态半导体存储器“动态”的含义是指需进行刷新。