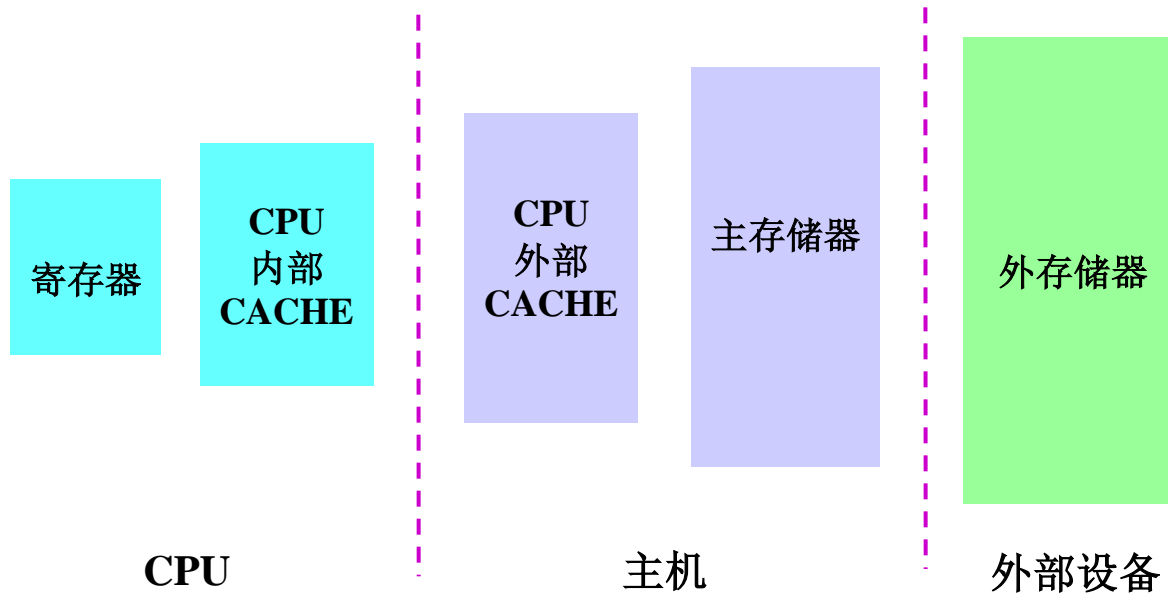
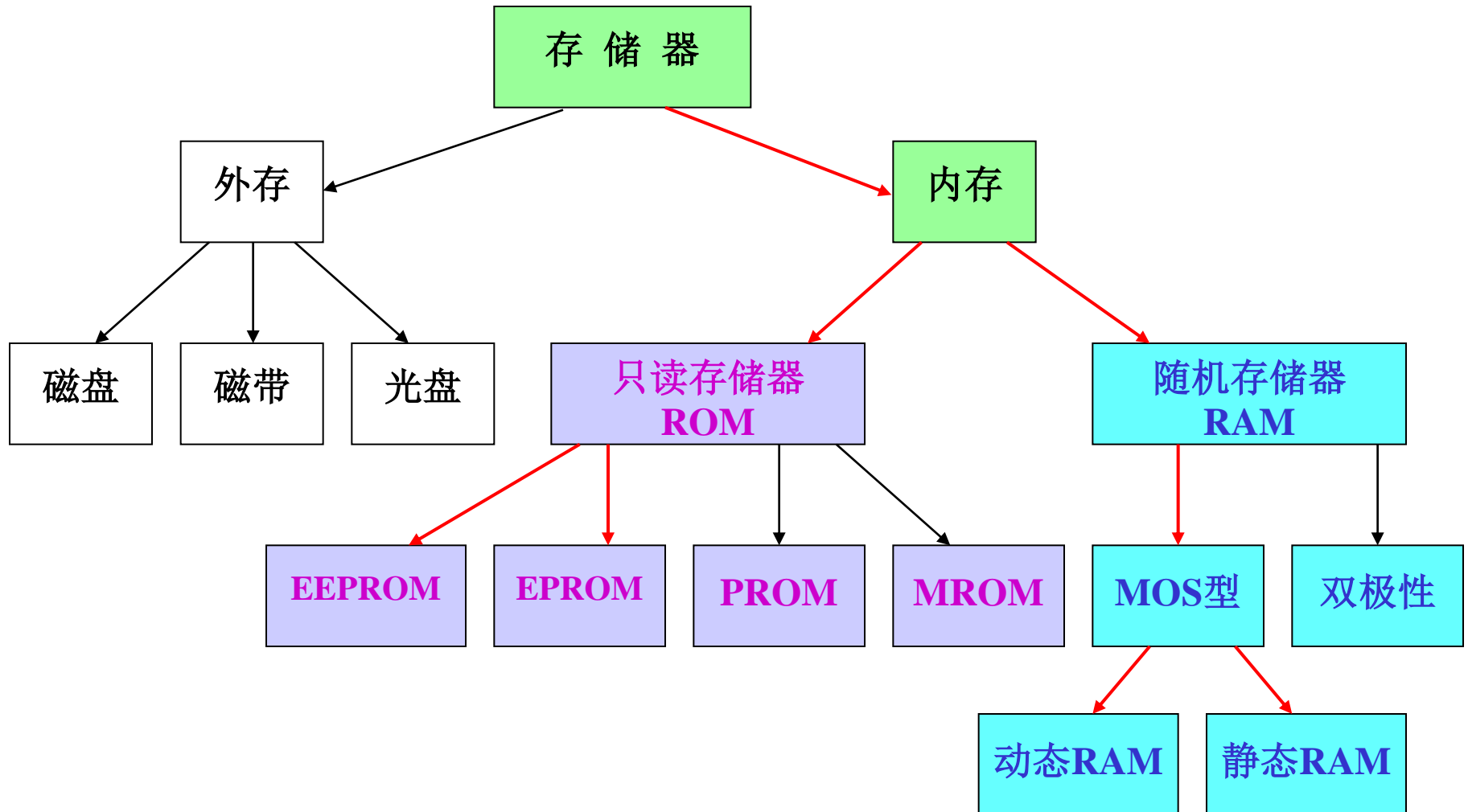


第4章 微型计算机的内存储器

- 4.1 概述
- 微型计算机存储系统采用层次结构：高速缓存技术、虚拟存储技术。
- 从连接角度：靠近CPU的存储体速度快、成本高、存储容量小。
- 从CPU角度：访问速度接近最靠近的存储体，访问空间接近最远的存储体空间范围。



存储器分类



4.2 存储器器件

- 4.2.1 随机存储器（RAM）
- 1. 静态随机存储器（SRAM）
- SRAM特点：**用**触发器**存储信息，电源掉电时信息丢失，属易失性半导体存储器，不需要刷新，速度快，功耗大。
- Intel 6264：**SRAM，容量8K×8位，28条引线，包括13根地址线、8根数据线、4根控制信号线。

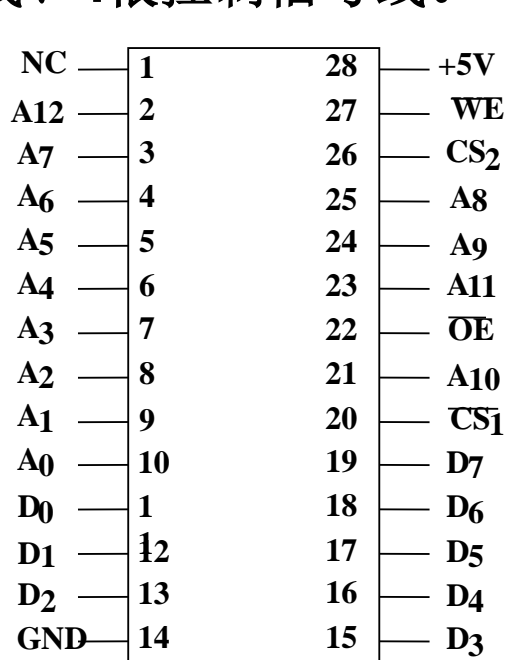
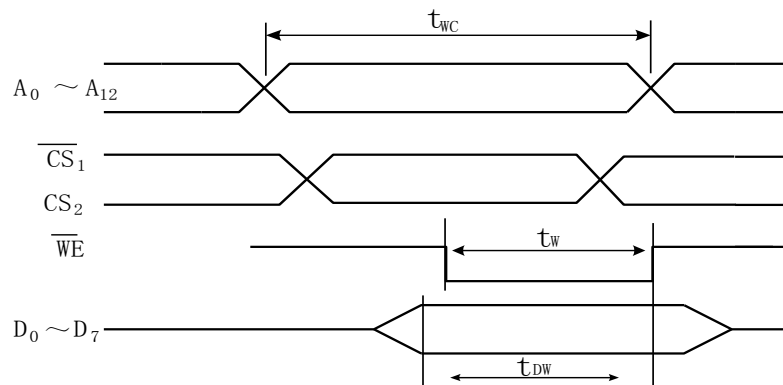


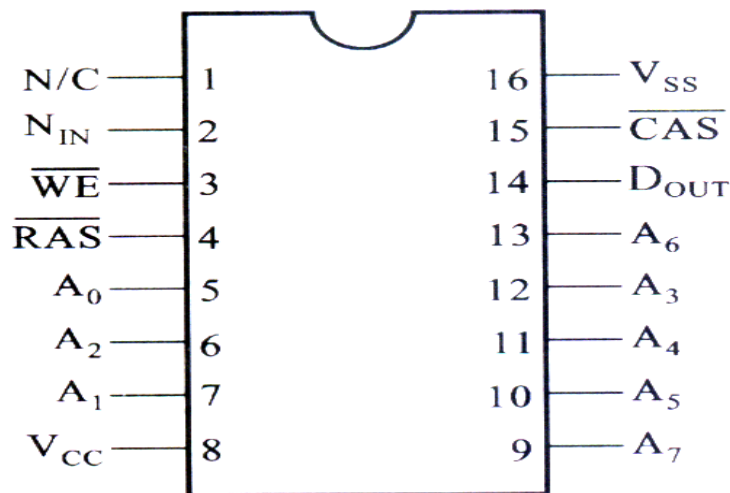
图4.3 SRAM 6264外部引脚



| \overline{WE} | $\overline{CS_1}$ | CS_2 | \overline{OE} | $D_0 \sim D_7$ |
|-----------------|-------------------|--------|-----------------|----------------|
| 0 | 0 | 1 | × | 写入 |
| 1 | 0 | 1 | 0 | 读出 |
| × | 0 | 0 | × | 三态 (高阻) |
| × | 1 | 1 | × | |
| × | 1 | 0 | × | |

2. 动态随机存储器DRAM

- 2. 动态随机存储器 (DRAM)
- DRAM特点:** 用**电容**存储电荷来保存信息, 电源掉电时信息丢失, 属易失性半导体存储器, 由于电容漏电, 所以要进行刷新, 存储容量高。
- 刷新:** 将存储单元的内容重新按原值设置一遍。读/写操作实际上也进行了刷新, 但由于读/写操作本身是随机的, 所以必须采用定时刷新的方法, 定时的时间叫做**刷新周期**。
- Intel 2164:** DRAM, 容量 $64K \times 1$ 位, 16脚封装, 16条地址线分为行地址与列地址两部分。刷新周期 $2ms$, 每次刷新512个存储单元, $2ms$ 内需有128个刷新周期。



4.2.2 只读存储器

- 只读存储器**ROM**: 非易失性半导体存储器, 掉电后信息不丢失。
- 1. 掩膜**ROM**
- 掩膜**ROM**: 在**ROM**的制作阶段, 通过“掩膜”工序将信息做到芯片里。
- 2. 可编程的只读存储器 (**PROM**)
- **PROM**: 一次可编程的只读存储器。
- 3. 可擦除、可编程的只读存储器 (**EPROM**)
- **EPROM**: 可由用户进行编程、并可用紫外光擦除的**ROM**芯片。
- 典型的**EPROM**芯片: 27系列。

2716 2K×8位

2732 4K×8位

2764 8K×8位

27128 16K×8位

27256 32K×8位

27512 64K×8位

27010 128K×8位

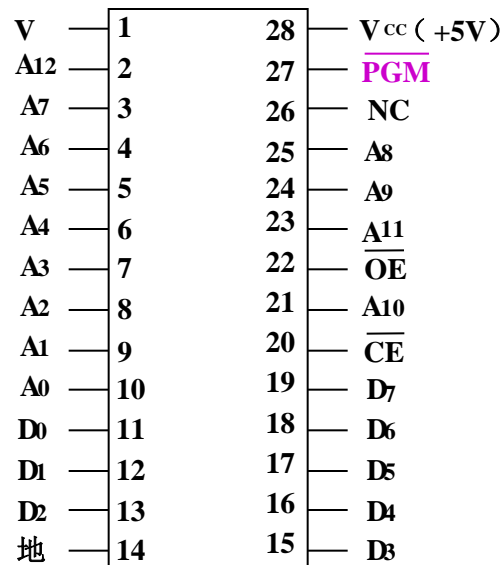
27020 256K×8位

27040 512K×8位

27080 1M×8位

EPROM 2764

- 2764的容量为 $8K \times 8$ 位，采用NMOS工艺制造，读出时间 $200 \sim 450ns$ ，13根地址线 $A_{12} \sim A_0$ ，8条数据线 $D_7 \sim D_0$ ，电源 V_{cc} ，编程电源 V_{pp} ，一个编程控制端。
- 编程时，引脚需加 $50ms$ 宽的负脉冲；正常读出时，该引脚应无效。另外，它还有一个片选端和一个输出允许控制端。
- **8种工作方式：**读出、读出禁止、待用、读Intel标识符、标准编程、Intel编程、编程校验、编程禁止。



4.电可擦除的可编程只读存储器（EEPROM）

- **EEPROM**: 可用加电的方法在线擦除和编程, 擦写次数大于1万次, 数据可保存10年以上。
- EEPROM芯片提供不同的擦写手段: 字节擦除、整片擦除、状态查询、页擦写等。
- 典型EEPROM芯片: 28系列, 如2816、2817、2864。

表4.2 2864的功能表

| 工作方式 | $\overline{\text{CE}}$ | $\overline{\text{OE}}$ | $\overline{\text{WE}}$ | D7~D0 |
|------|------------------------|------------------------|------------------------|-------|
| 维持 | 1 | x | x | 高阻 |
| 读出 | 0 | 0 | 1 | 输出 |
| 写入 | 0 | 1 | 负脉冲 | 输入 |
| 数据查询 | 0 | 0 | 1 | 输出 |

| | | | |
|-----|----|----|------------------------|
| V | 1 | 28 | V _{CC} (+5V) |
| A12 | 2 | 27 | $\overline{\text{WE}}$ |
| A7 | 3 | 26 | NC |
| A6 | 4 | 25 | A8 |
| A5 | 5 | 24 | A9 |
| A4 | 6 | 23 | A11 |
| A3 | 7 | 22 | $\overline{\text{OE}}$ |
| A2 | 8 | 21 | A10 |
| A1 | 9 | 20 | $\overline{\text{CE}}$ |
| A0 | 10 | 19 | D7 |
| D0 | 11 | 18 | D6 |
| D1 | 12 | 17 | D5 |
| D2 | 13 | 16 | D4 |
| 地 | 14 | 15 | D3 |

5. 快速擦写存储器 (Flash Memory)

- **Flash Memory:** 简称Flash, 快速擦写存储器, 又叫闪速存储器、快擦型存储器。
- **特点:** Flash兼顾EPROM和EEPROM的特点, 沿用了EPROM的简单结构和编程写入方式, 兼备了EEPROM的电擦除特性, 既有EPROM的价格便宜、集成度高的优点, 又有EEPROM在线擦除、改写的特性, 是一种新型的非易失性存储器, 具有更好的性价比, 可靠性高, 整片擦除, 擦除、重写速度快。
- **应用场合:** 高密度、非易失性、长期反复使用的大容量数据采集和存储, 如IC卡、便携式计算机、工控系统、单片机系统和其他电子设备中。大容量的Flash也可以“固态硬盘”的形式代替软盘或硬盘作为海量存储器用。

4.3 CPU与存储器的连接

- 4.3.1 存储器芯片与CPU连接时应处理的问题
 - 1. 总线驱动能力
 - CPU的总线驱动能力是指可以直接驱动的标准门电路器件数量，通常CPU的直流负载是一个TTL器件，现在多为MOS器件，为容性负载，直流分量较小，可带20个。但若系统较大，需要驱动AB、DB、CB。
 - 2. 时序配合
 - CPU时序与存储器存取速度配合，可插入等待状态Tw。
 - 3. 数据线的连接
 - 字节编址，存储体为8位，CPU数据线宽度不同，连接的存储体的个数也不同。

4.3.1 存储器芯片与CPU连接时应处理的问题

- 4. 地址线的连接
- 问题：多个芯片连接，需要选择。
- CPU地址信号与存储器连接的选择有三种：
- (1) 高位地址信号
- CPU高位地址选择存储器芯片，即片选，有三种方法，即全译码、部分译码、线选。
 - 1) 全译码法：高位地址线全部参加译码，地址连续。
 - 2) 部分译码法：高位地址线部分参加译码，地址不连续。
 - 3) 线选法：高位地址线单根地址线进行片选，地址不连续。
- (2) 低位地址信号
- CPU低位地址用于片内译码，即字选。
- (3) 最低位地址信号
- CPU最低位地址选择存储器体，即体选。每个存储体数据宽度一个字节，8位CPU连1个存储体，依次，16位→2个体，32位→4个体，64位→8个体。

存储器芯片与CPU连接时应处理的问题

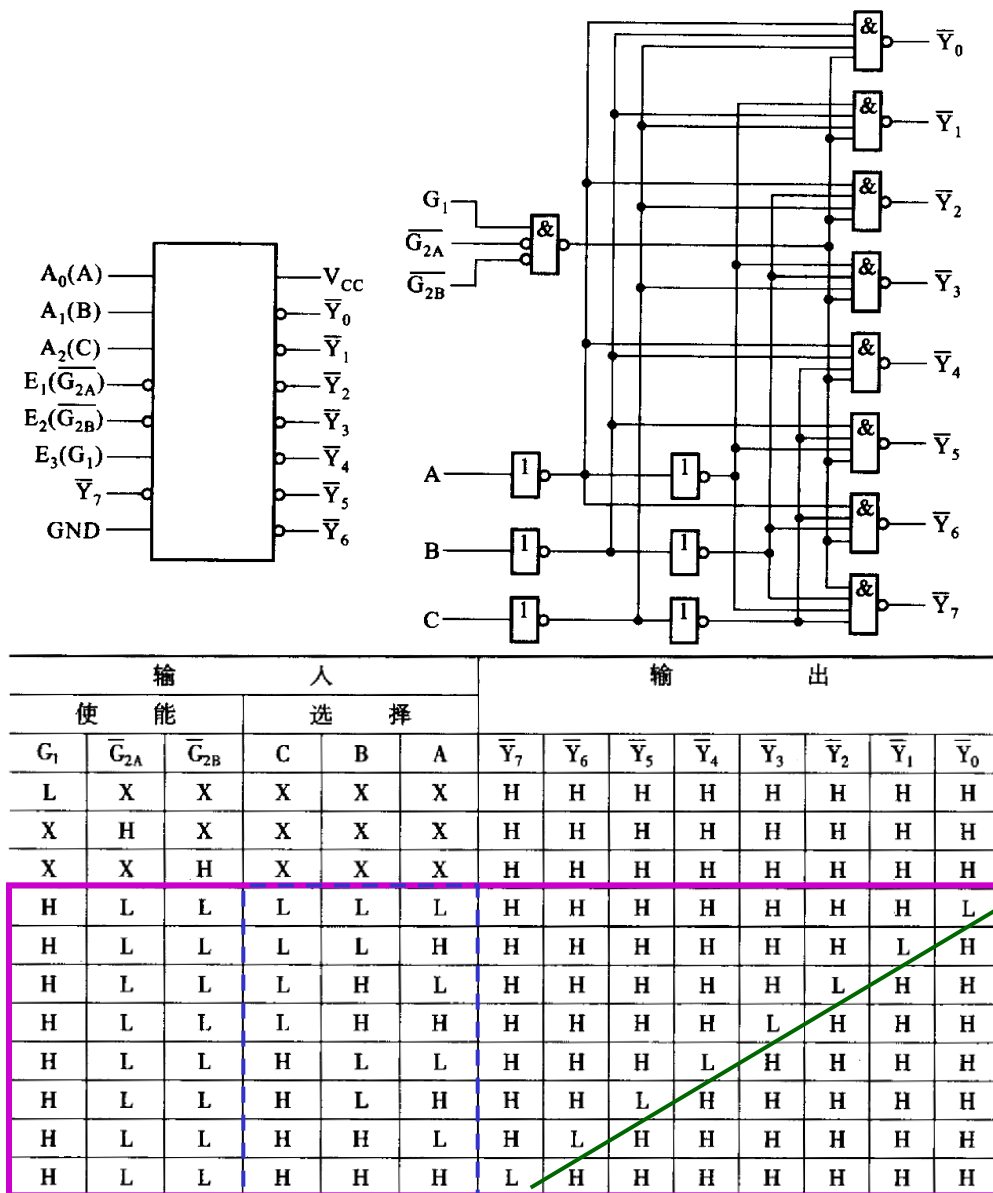
- 5. 读写控制线的连接
- 读写控制信号用于控制对存储器的读/写操作。当存储芯片的工作速度与CPU不匹配时，存储器芯片的接口电路就必须具有向CPU发等待命令的控制信号，以使CPU根据需要插入1个或几个等待周期。
- 6. ROM与RAM在存储器中的地址分配
- RAM存放临时数据和当前的应用软件，非易失的ROM存放核心系统软件。80X86微处理机复位后从物理地址高端开始运行，所以总是在物理地址空间的高地址位置使用只读存储器ROM。
- 7. 对多种宽度数据访问的支持
- 存储器编址的最小单元称为编址基本单元，80x86微处理器的编制基本单元为字节。64位外部数据线的Pentium访问存储器时，要求可以一次读写一个字节、或任何地址开始的连续两个字节（字）、或任何地址开始的连续四字节（双字）、或任何地址开始的连续八字节（四倍字），即支持多种数据宽度。存储系统设计时必须支持CPU的这种访问方式。

4.3.2 地址译码器

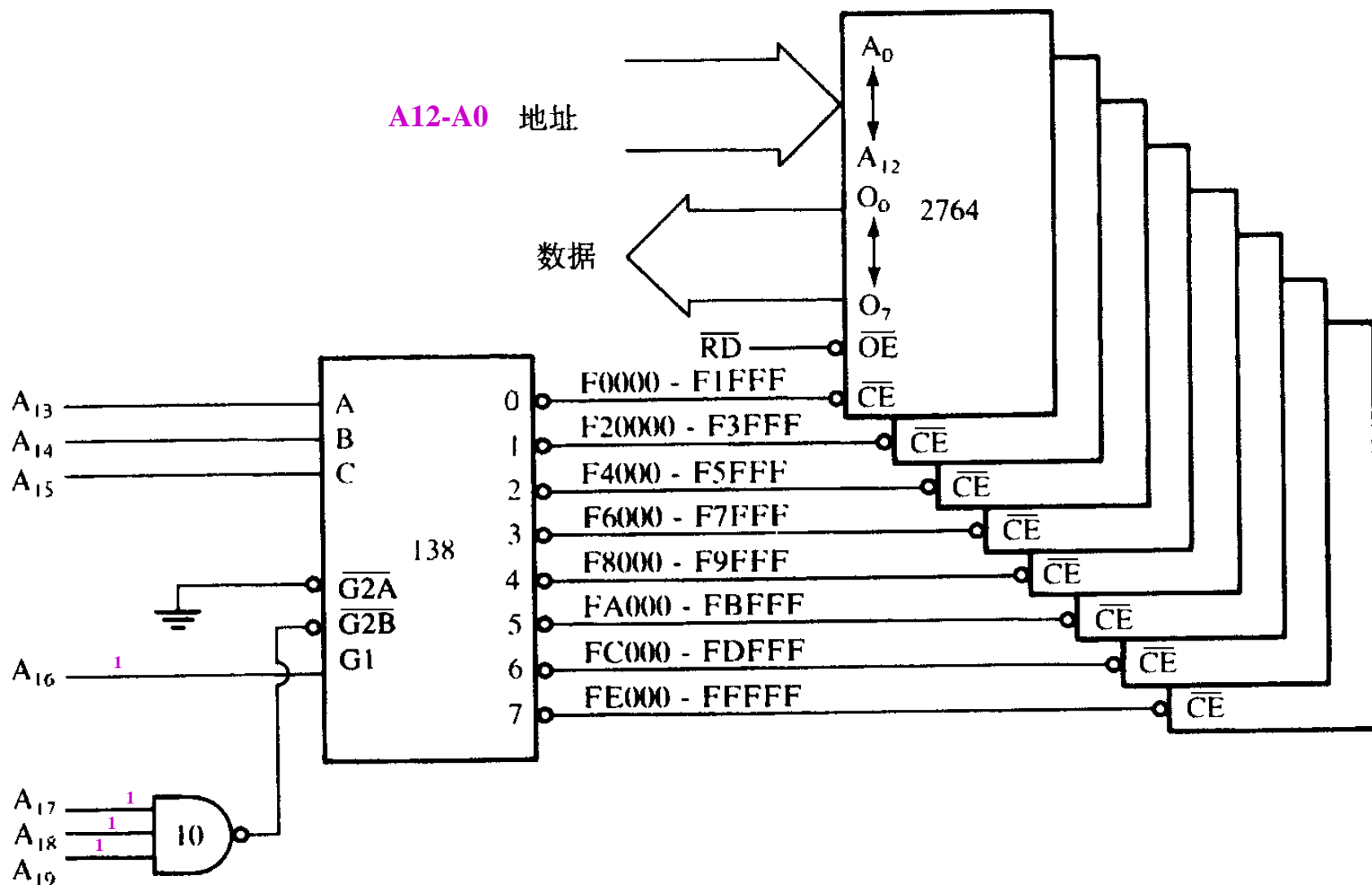
1. 3/8译码器（74LS138）

- 在微型计算机系统中，CPU与存储器连接时一般采用译码器芯片作为高位地址产生片选信号的地址译码器。

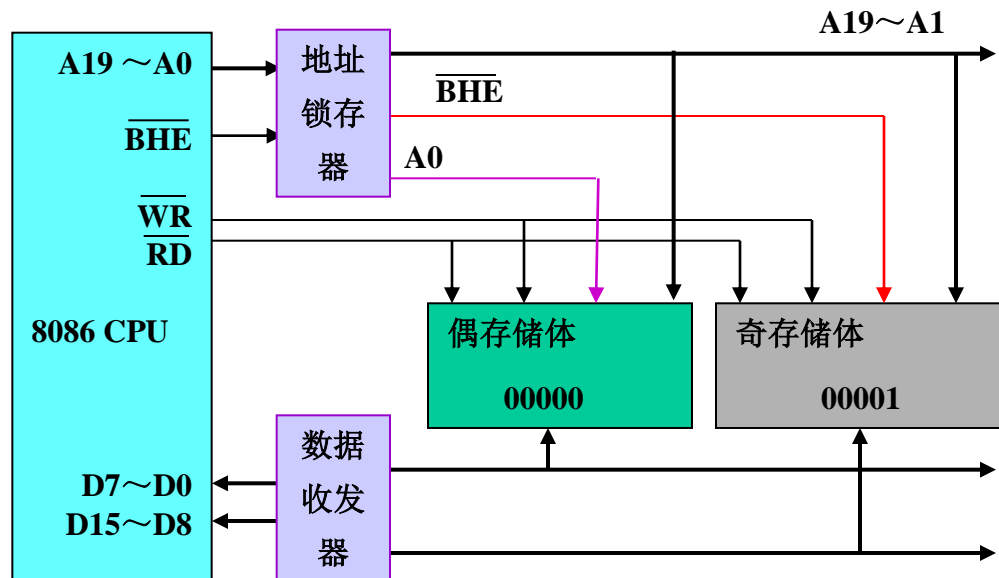
- 常用的译码器之一是74LS138，这是一个3/8译码器芯片，有三个选择输入端，三个控制端，八个输出端。



2. 74LS138译码器的应用



4.3.3 16位微型计算机系统的主存储器接口



接口关键:

(1) 16位/B=2个存储体, 偶体和奇体。

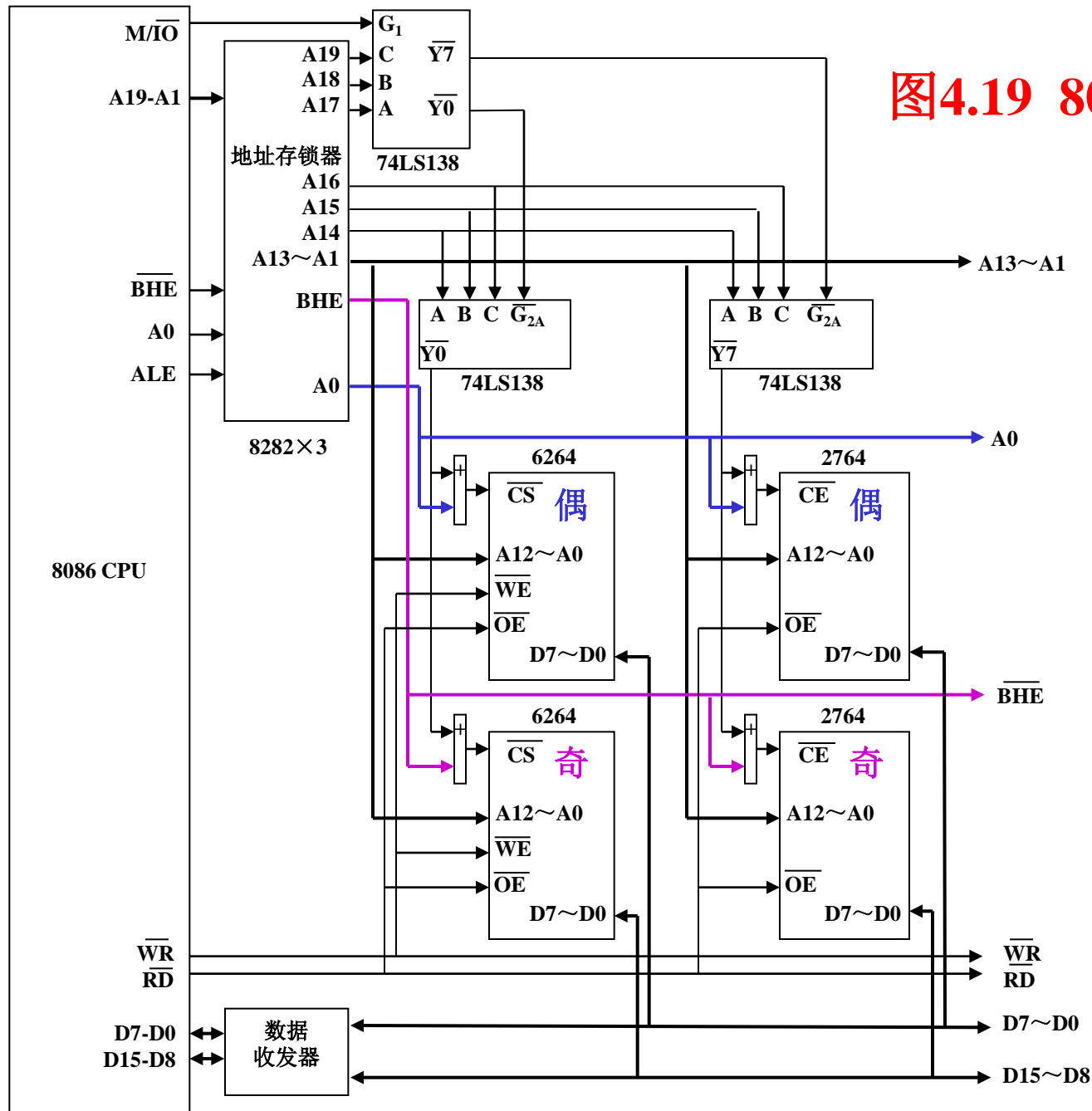
(2) 偶体D7-D0, 由A0=0选择。

(3) 奇体D15-D8, 由 \overline{BHE} =0选择。

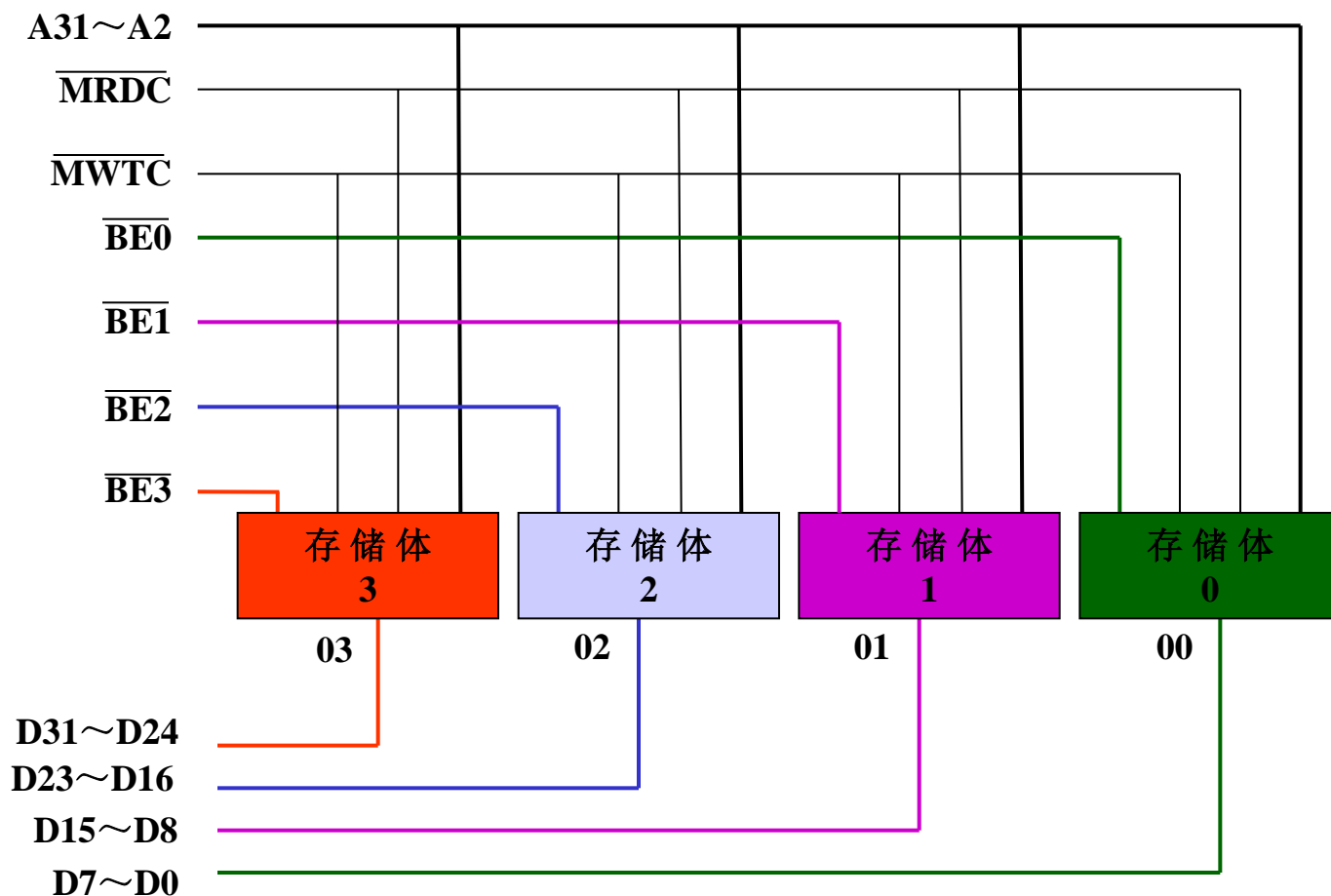
表4.5 存储体选择编码表

| \overline{BHE} | A0 | 传送的字节 | 所用的数据引脚 |
|------------------|----|---------------|---------|
| 0 | 0 | 从偶地址开始两个字节读/写 | D15~D0 |
| 0 | 1 | 奇地址的一个字节读/写 | D15~D8 |
| 1 | 0 | 偶地址的一个字节读/写 | D7~D0 |
| 1 | 1 | 不传送 | |

图4.19 8086存储器系统例

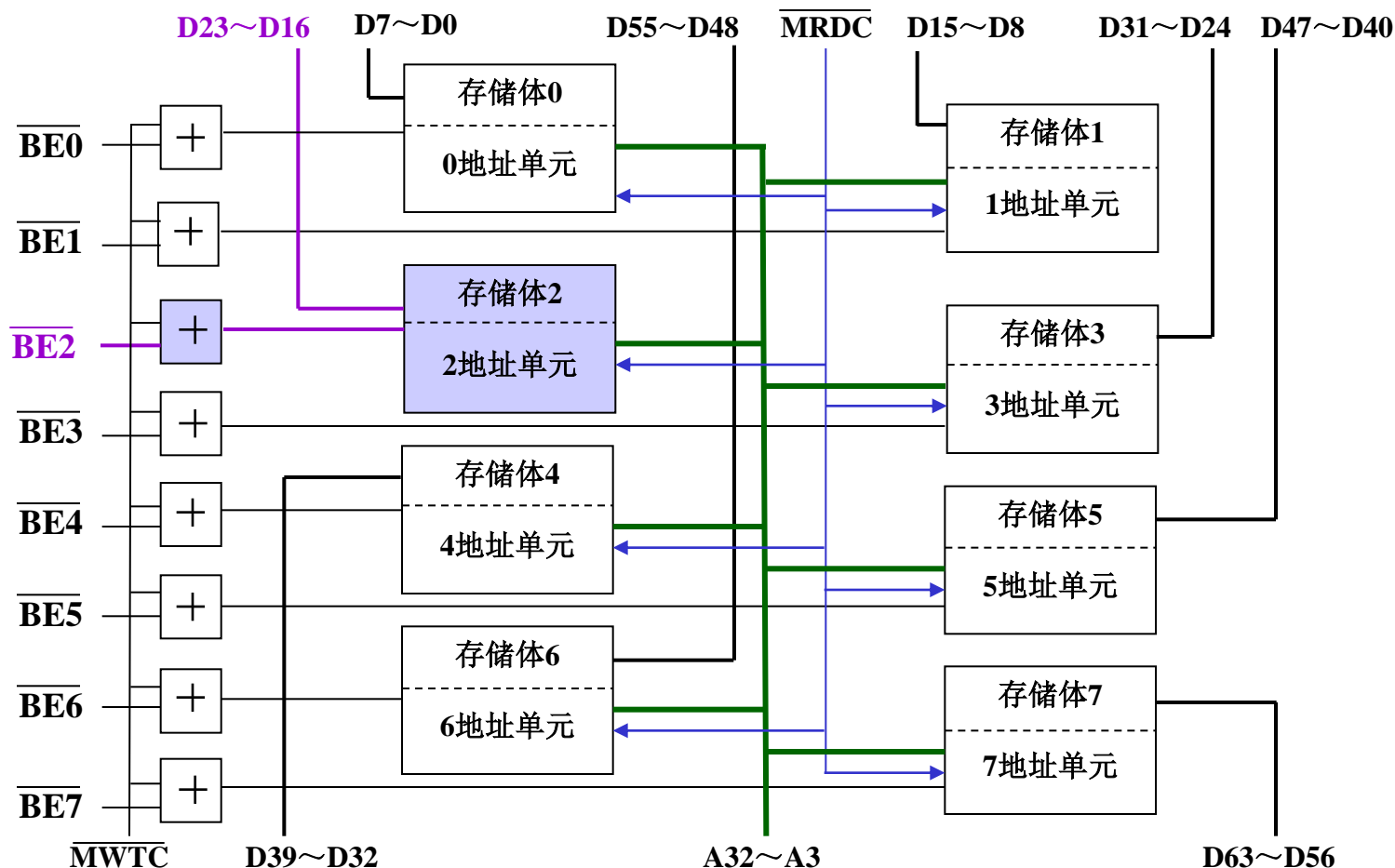


4.3.4 32位微型计算机系统的主存储器接口



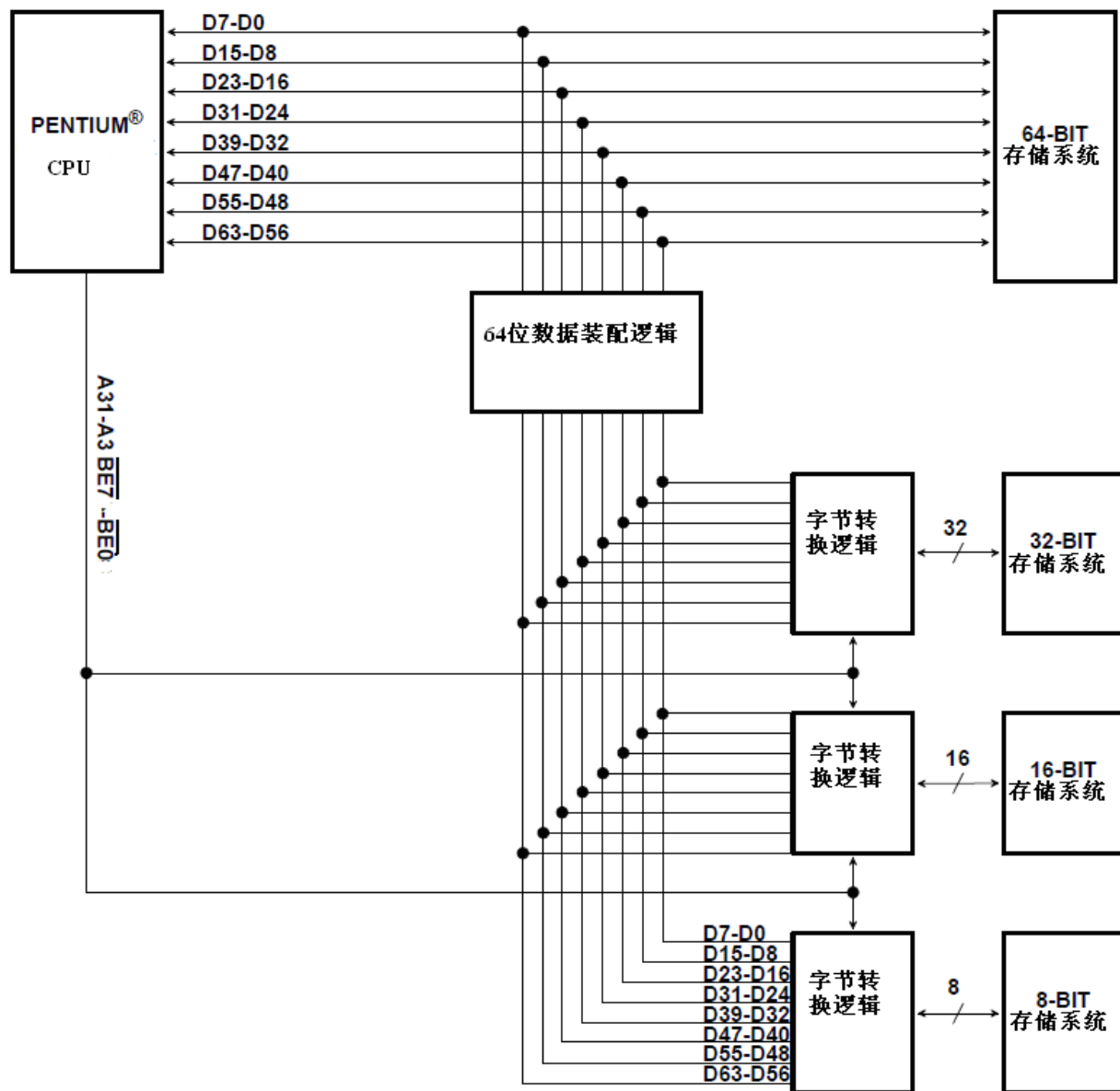
- 接口关键:
- (1) 4个体, 每体1G
 - (2) 每体分别连相应的8位数据线
 - (3) 每个体用BE2-0对应选通

4.3.5 64位微型计算机系统的主存储器接口



- 接口关键:
- (1) 8个体, 每体512M
 - (2) 每体分别连相应的8位数据线
 - (3) 每个体用 $\overline{BE}7-0$ 对应选通, 独立写选通

Pentium与存储器的接口逻辑



4.4 高速缓冲存储器Cache

- 1. **Cache**: 弥补主存速度, 在CPU与主存之间设置的**高速、小容量的存储器**, 构成Cache-主存存储层次, 速度是Cache的, 容量是主存的。
- 2. **解决CPU与主存速度的问题, 3种方法:**
 - (1) 在CPU中增设**寄存器**。
 - (2) 采用**多体交叉并行存储器**。
 - (3) 采用**Cache存储器**。
- 主存速度不够→引出Cache, 主存容量不够→引出虚拟存储器。
- 3. **程序具有局部性** (各存储层次构成的主要依据)
 - (1) **时间上的局部性**: 最近的未来要用到的信息可能是现在正在使用的信息, 因为程序存在循环。
 - (2) **空间上的局部性**: 最近的未来要用到的信息可能与现在使用的信息在程序空间上是邻近的, 因为指令按顺序存放。
- 4. **命中率**:
$$H = \text{命中次数} / \text{总访问次数}$$

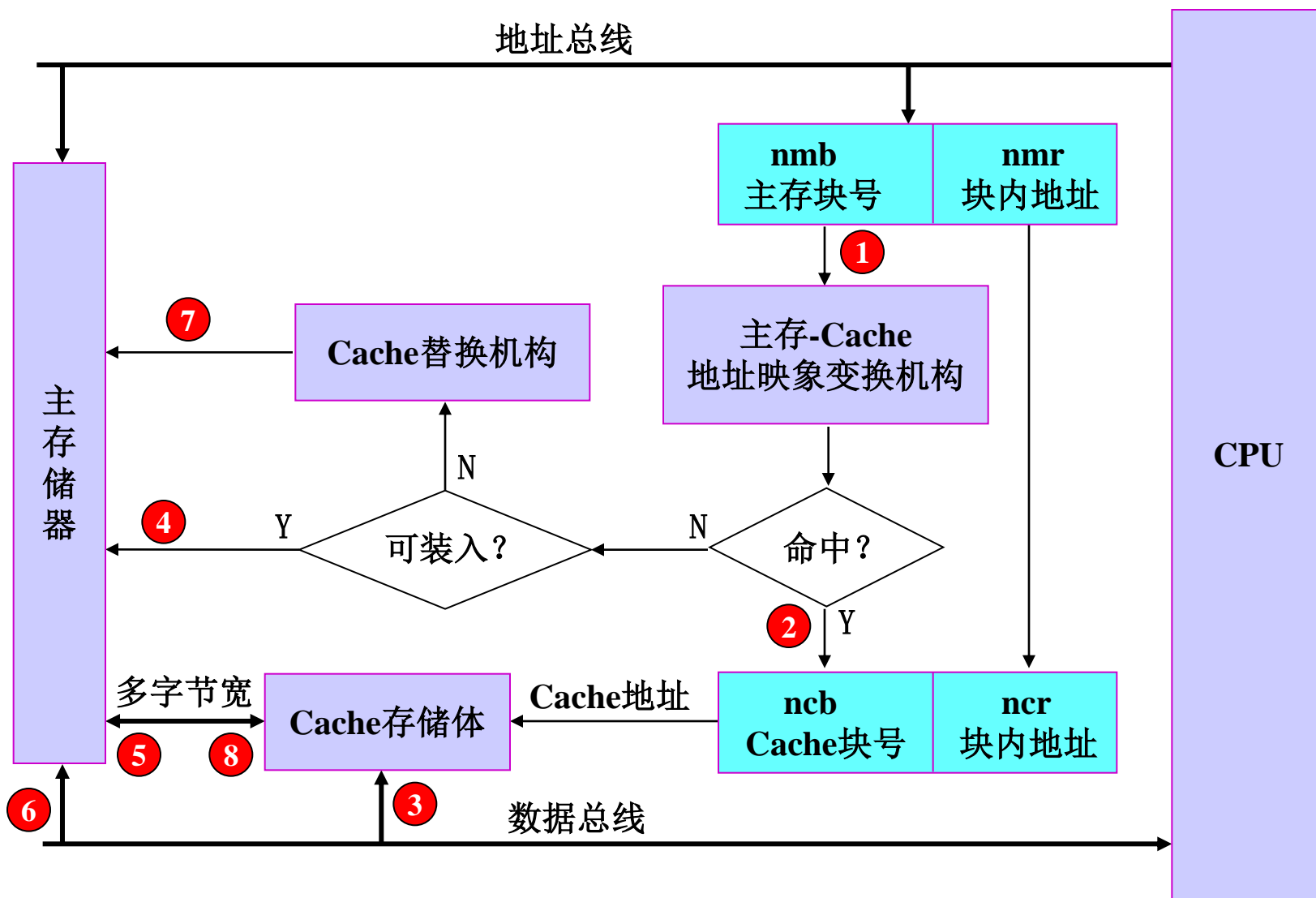
4.4.1 Cache的工作原理简介

- 1. Cache及主存地址的组成
- Cache和主存等分成容量相同的块，每块由4、16、32等字节组成，并且把块有序地编号。
- 2. Cache的基本结构
- **Cache基本结构**：Cache存储体、地址映象变换机构、替换机构。
- **Cache存储体**：Cache存储信息的主体。
- **地址映像**：将每个主存块按什么规则装入Cache中。
- **地址变换**：将主存地址变换成Cache地址。
- **块冲突**：主存块要进入Cache中的位置已被其他主存块占用，要用替换算法。
- **替换机构**：发生块冲突时，根据替换算法完成Cache块替换。
- Cache操作由硬件完成，对用户透明。
- 在进行地址变换、替换时，都是以块为单位进行调度。

| | | |
|-------------|----------|------------|
| nmb 主存块号 | nmr 块内地址 | nm 主存地址 |
| ncb Cache块号 | ncr 块内地址 | nc Cache地址 |

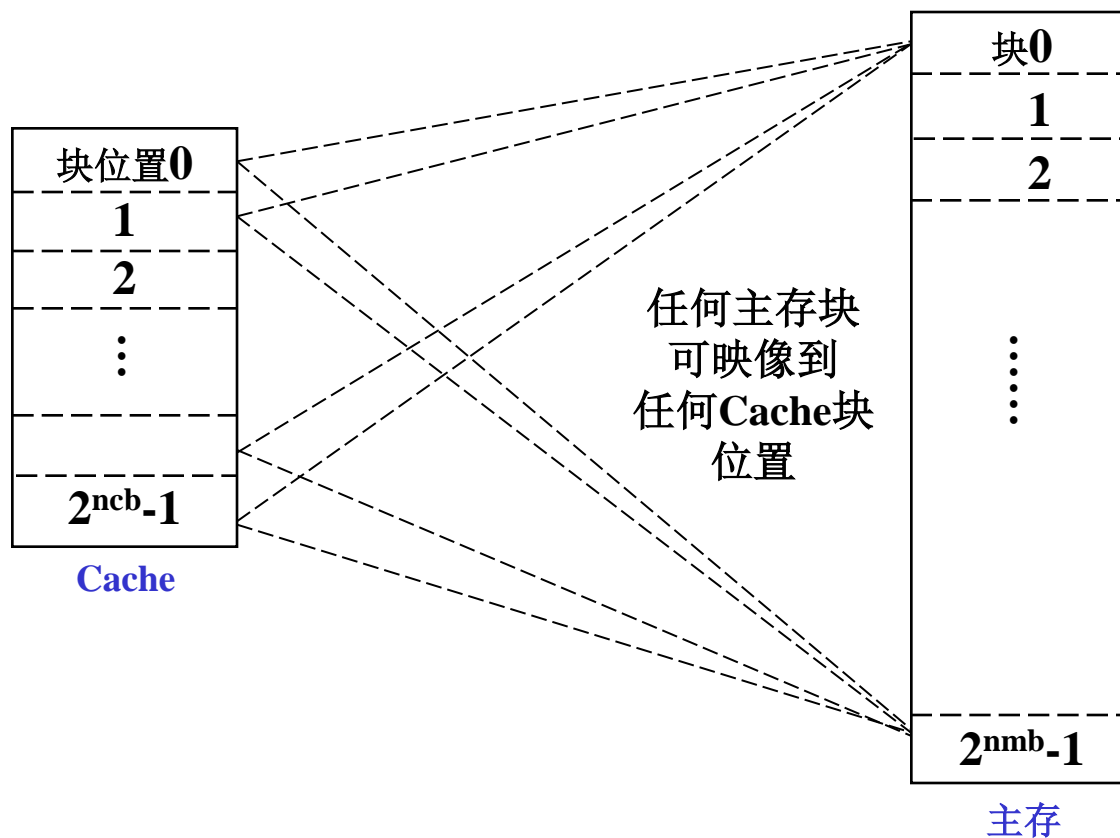
nmr = ncr

3. Cache的工作过程



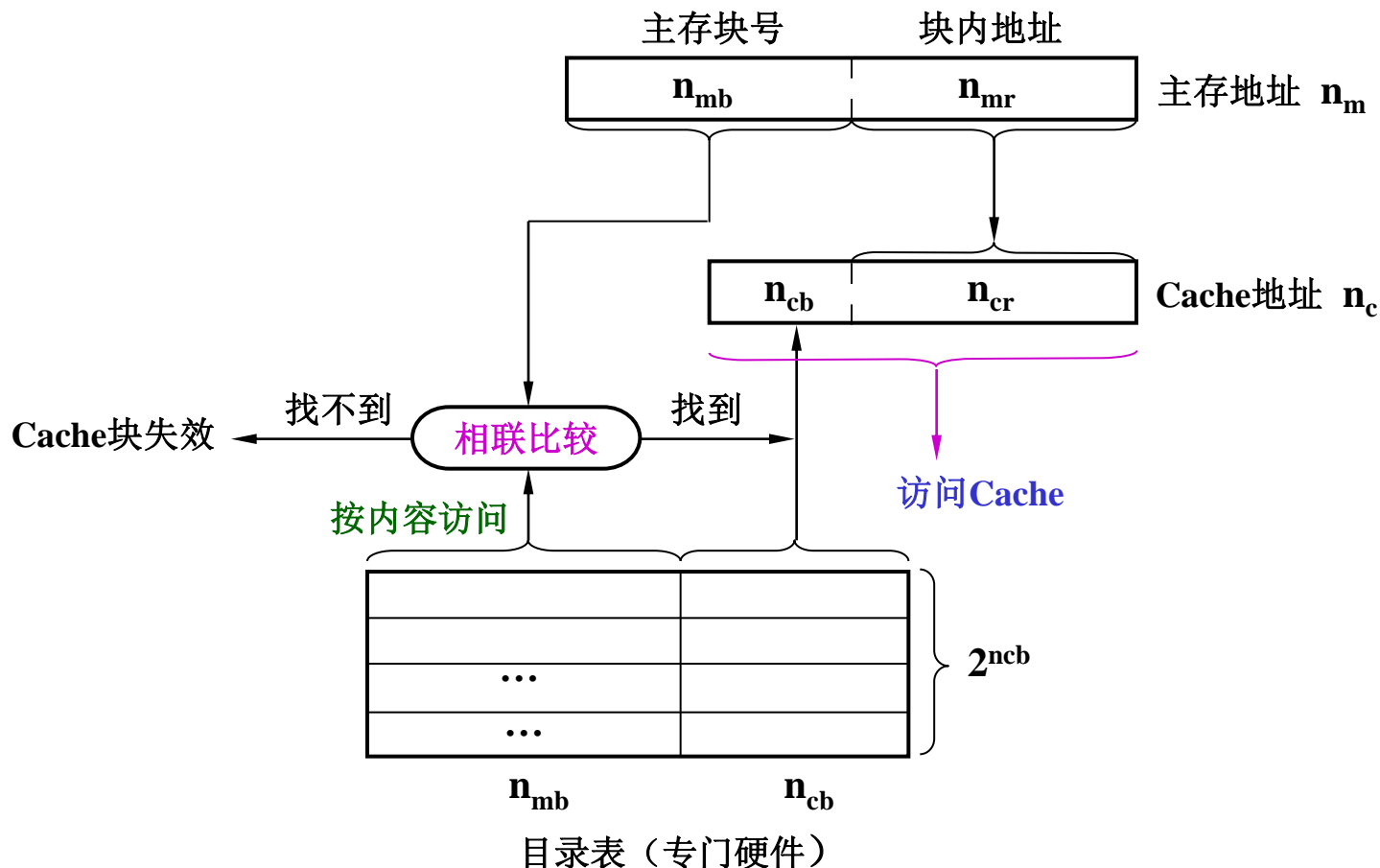
4. 地址映像 - 全相联映像

- **地址映像：**主存中任意一块都可映像装入到Cache中任意一块位置。
- **优点：**灵活，块冲突率低，只有在Cache中的块全部装满后才会出现冲突，Cache利用率高。

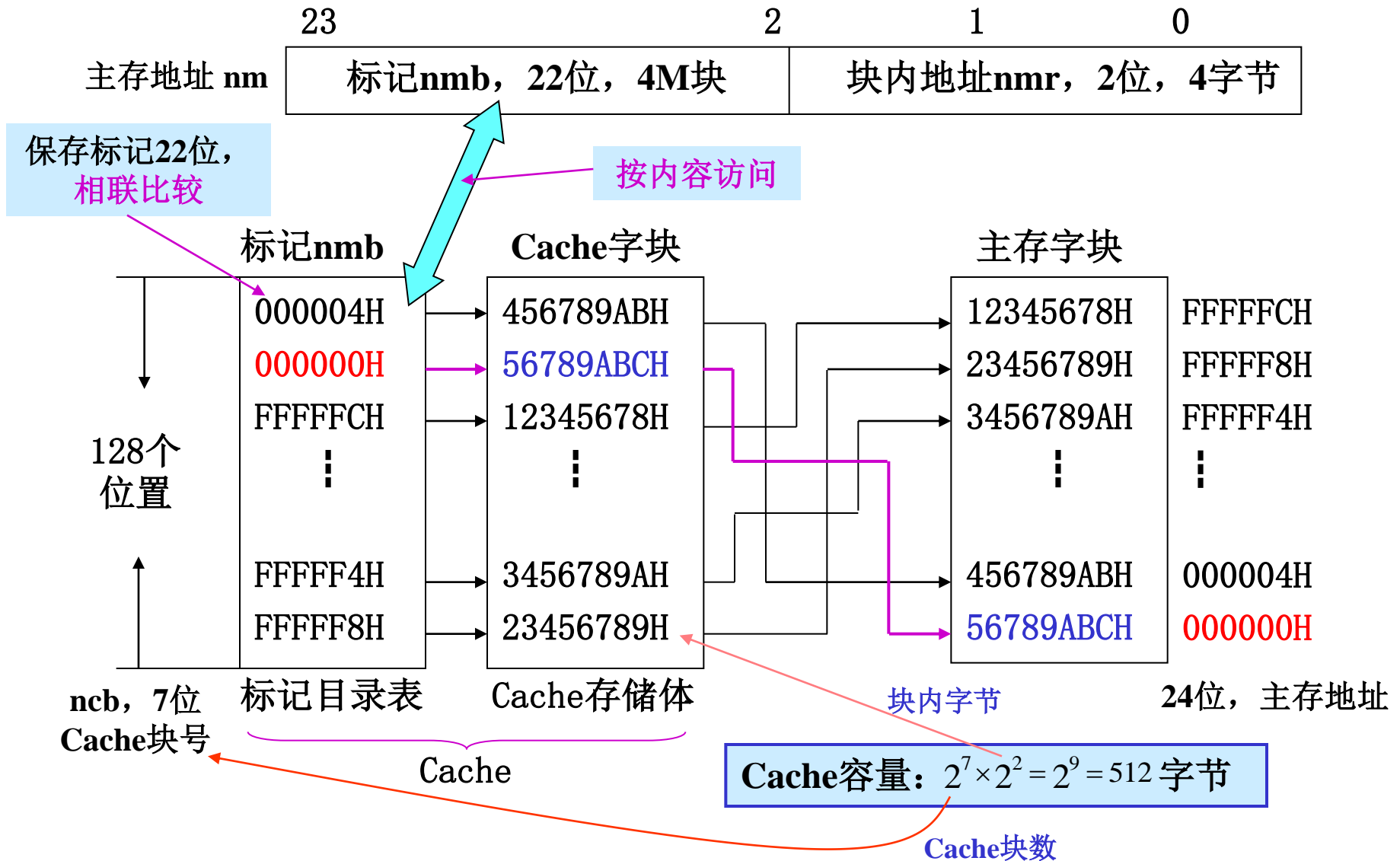


全相联映像地址变换

- 地址变换：硬件实现。目录表，相联比较。
- 缺点：地址变换机构复杂，成本高。

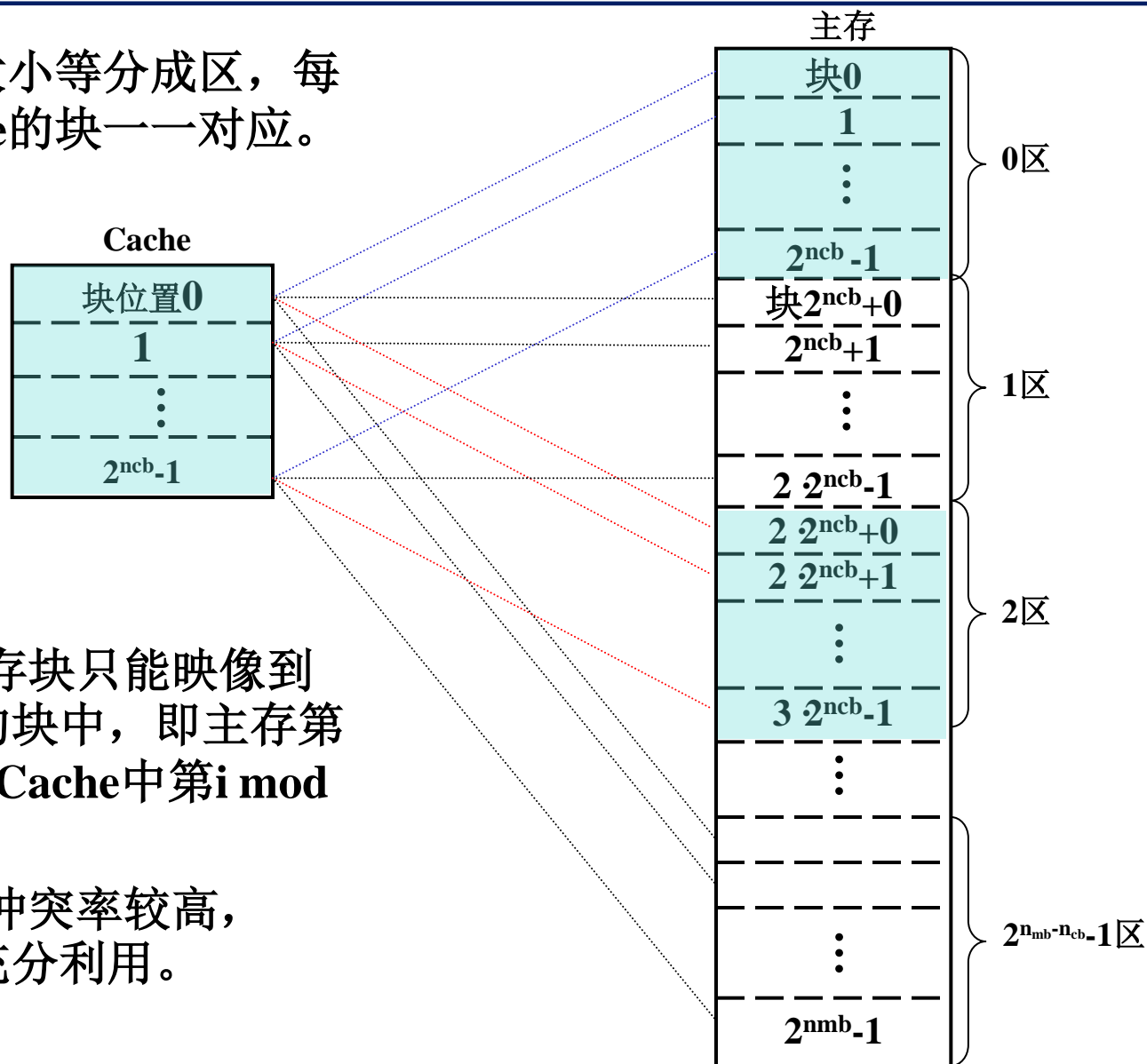


全相联映像示意图



4. 地址映像 - 直接映像

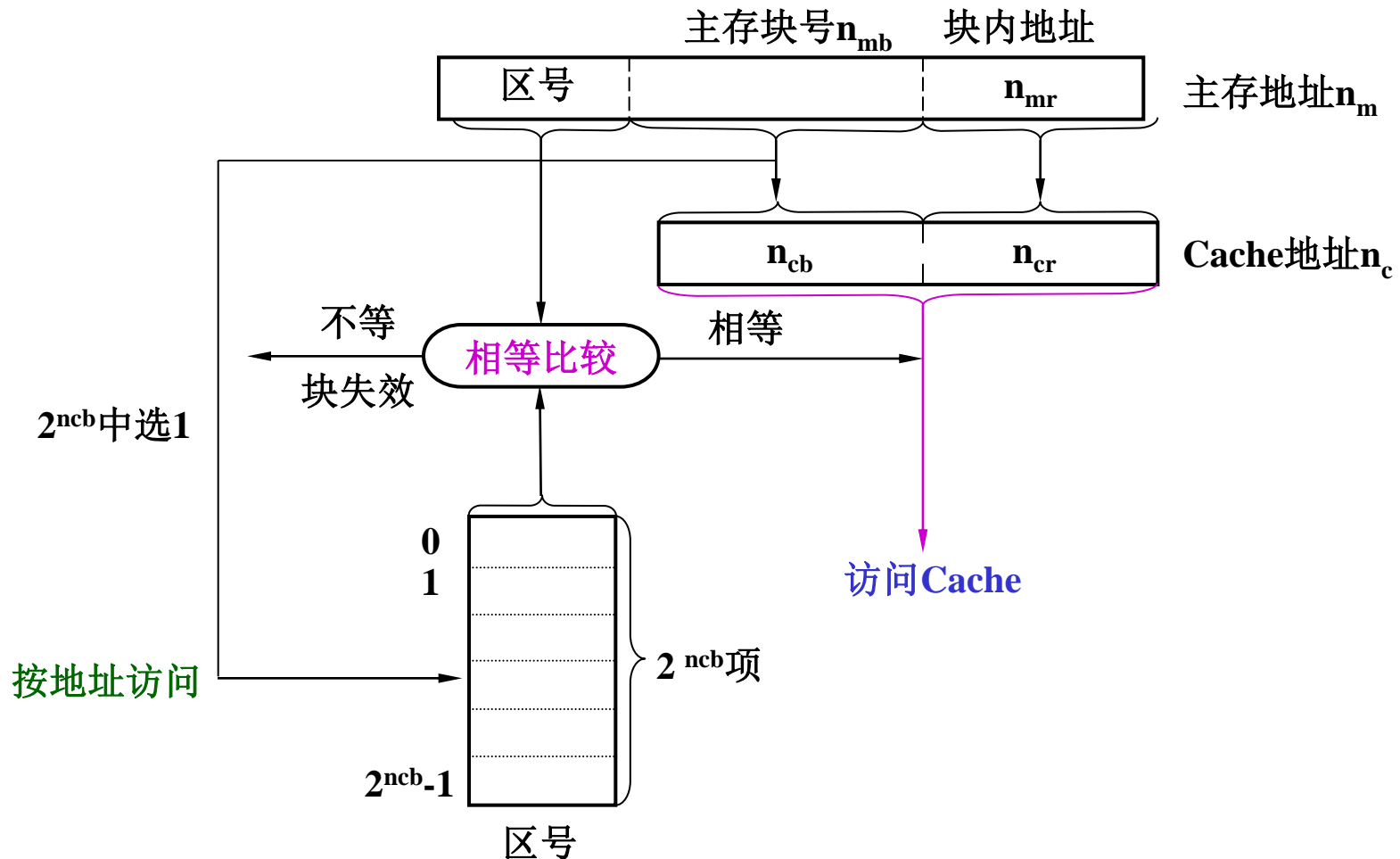
- 主存空间按Cache大小等分成区，每区内的各块与Cache的块一一对应。



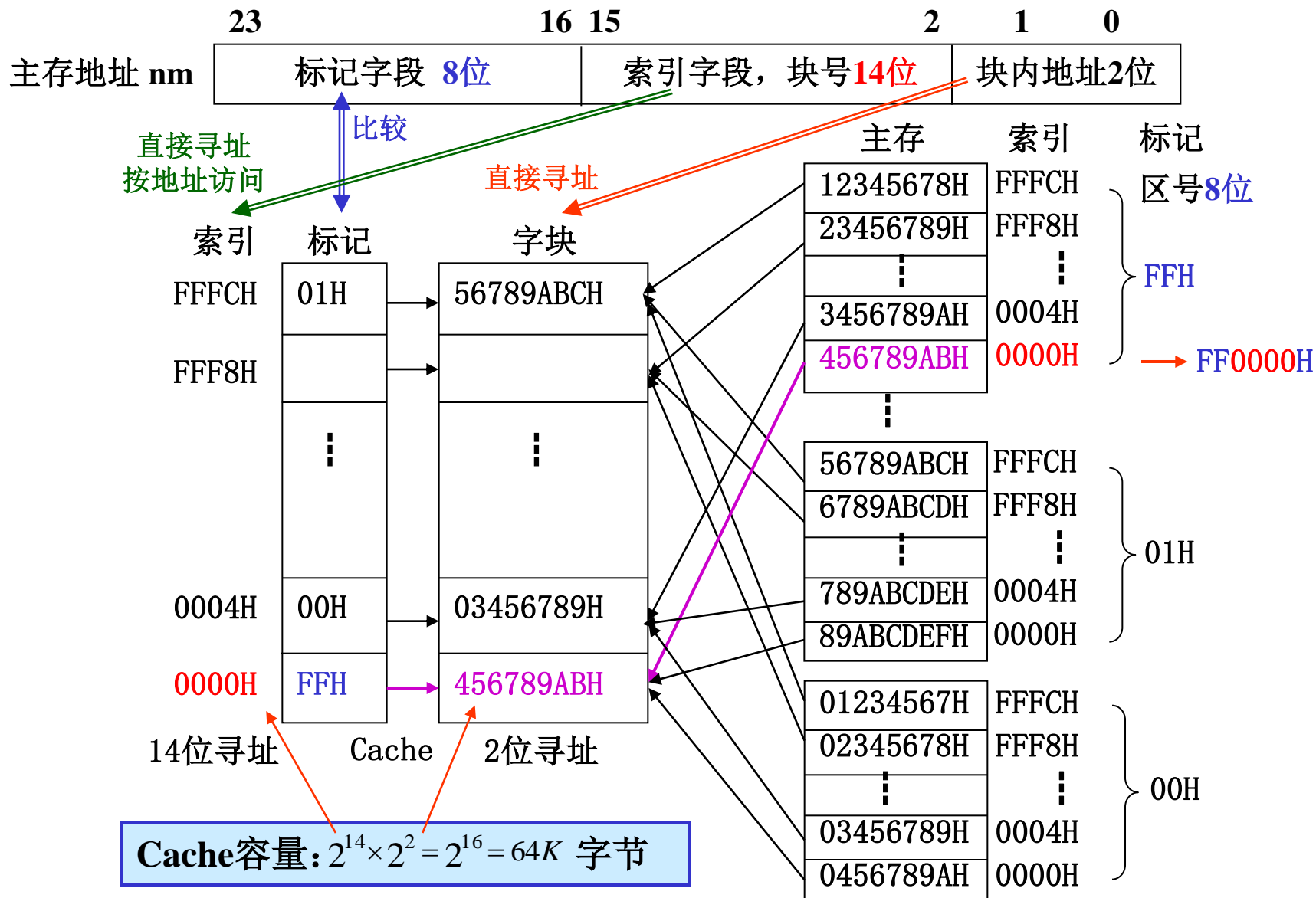
- 地址映像:** 一个主存块只能映像到Cache中唯一指定的块中，即主存第 i 块只能唯一映像到Cache中第 $i \bmod 2^{n_{cb}}$ 块位置上。
- 缺点:** 不灵活，块冲突率较高，Cache空间得不到充分利用。

直接映像地址变换

- **地址变换：**硬件实现。主存地址中直接产生Cache地址，标志表中比较区号，标志表存储器按地址访问。
- **优点：**地址变换简单、速度快，节省硬件。



直接映像示意图



4. 地址映像 - N路组相联映像

地址映像：路内直接映像，路间全相联映像

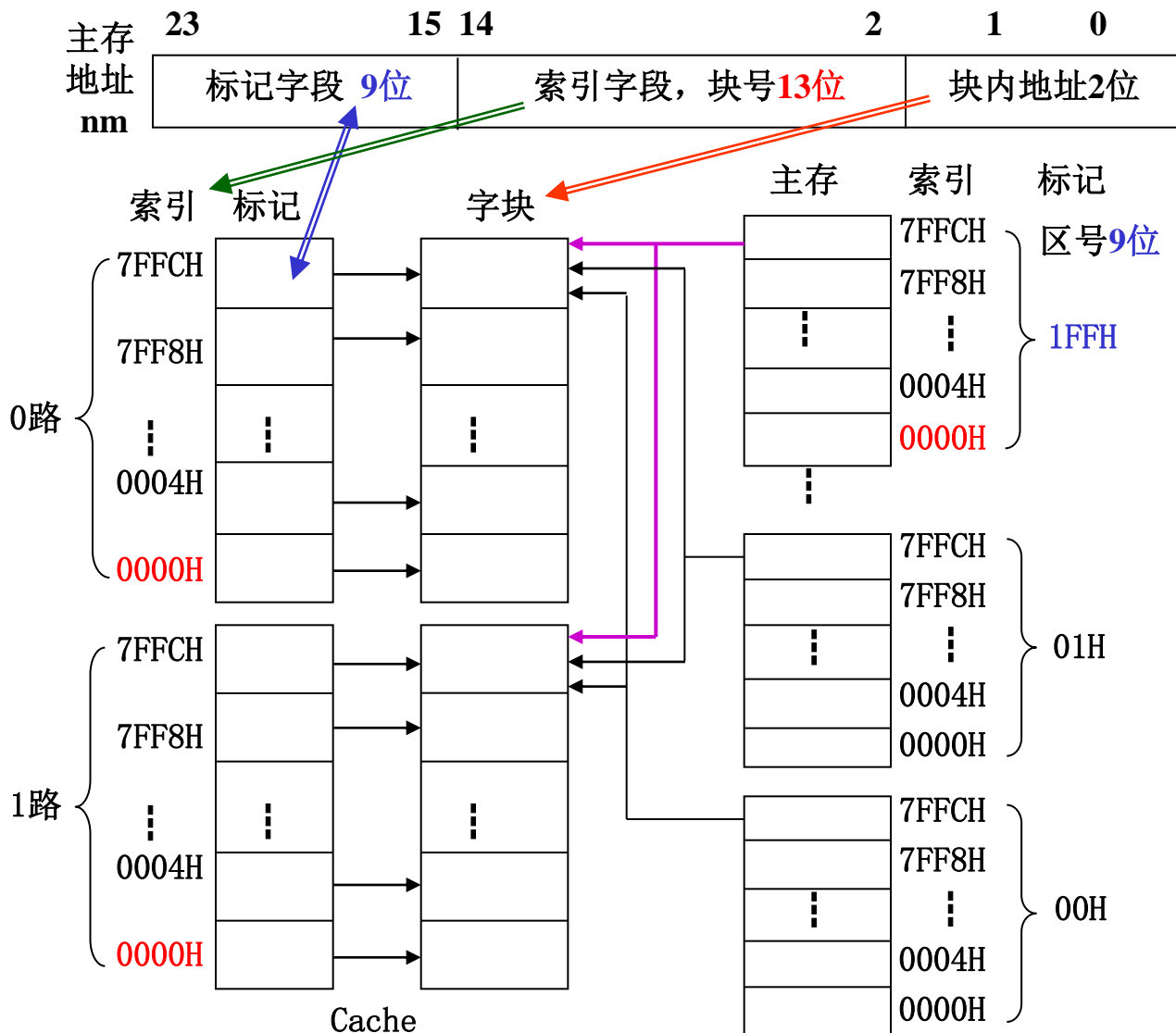
地址变换：主存地址中直接产生Cache地址（2路对应地址相同），同时比较各路对应块标记

特点：是直接映像与全相联映像的折中方案，若路数为1，则为直接映像，若路内块为1，则为全相联映像

每一路容量：

$$2^{13} \times 2^2 = 2^{15} = 32K$$

2路容量为64K字节



两路地址相同，内容靠标记区分

第4章 结 束