# 计算机组成原理

翁睿

哈尔滨工业大学

# 4.3 高速缓冲存储器

## 一、概述

### 1. 问题的提出

减小 CPU 和主存（DRAM）的速度差异

避免 CPU “空等” 现象

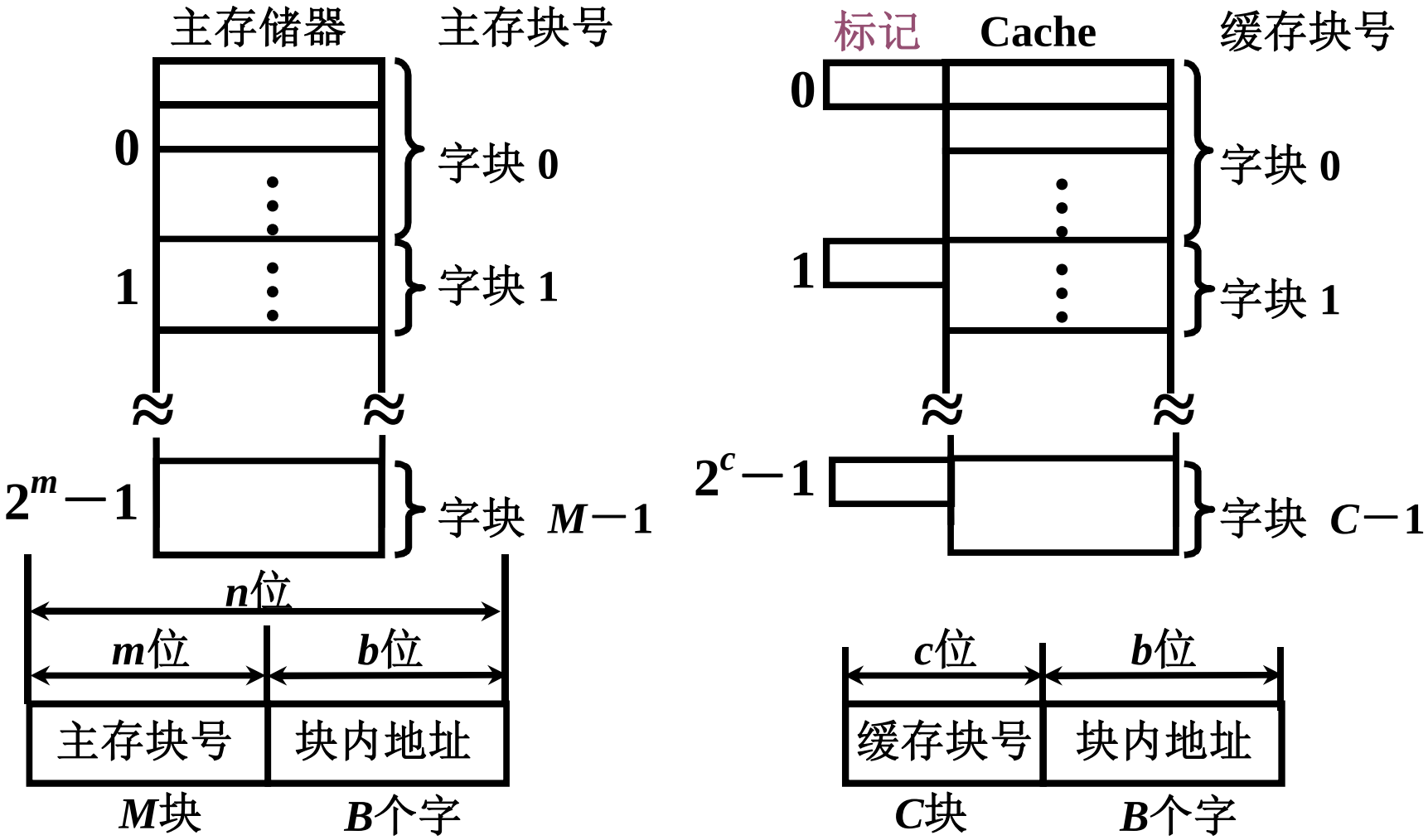
解决方法：利用程序访问的局部性原理，
将常用的信息放入缓存中。

- 时间局部性：程序马上将要用到的信息很可能就是现在正在使用的信息。
- 空间局部性：程序马上将要用到的信息很可能与现在正在使用的信息在存储空间上是相邻的。

# 2. Cache 的工作原理



## (1) 主存和缓存的编址




主存和缓存按块存储　　块的大小相同

# (2) 命中与未命中



缓存共有 $C$ 块

主存共有 $M$ 块　$M >> C$

命中　主存块 已调入 缓存

主存块与缓存块 建立 了对应关系

未命中　主存块 未调入 缓存

主存块与缓存块 未建立 对应关系

用 标记 记录与某缓存块建立了对应关系的 主存块块号

## (3) Cache 的命中率和平均访存时间 4.3

命中率： $H=\dfrac{N_1}{N_1+N_2}$

$N_1$ —— 访问Cache的次数

$N_2$ —— 访问主存的次数

不命中率： $F=1-H$

设 Cache 命中率 为 $h$，访问 Cache 的时间为 $t_c$，

访问 主存 的时间为 $t_m$

平均访存时间 $t_a = h \times t_c + (1-h) \times t_m$

# (4) Cache –主存系统的效率



效率 $e$ 与 命中率 有关

$$e = \frac{\text{访问 Cache 的时间}}{\text{平均访问时间}} \times 100\%$$

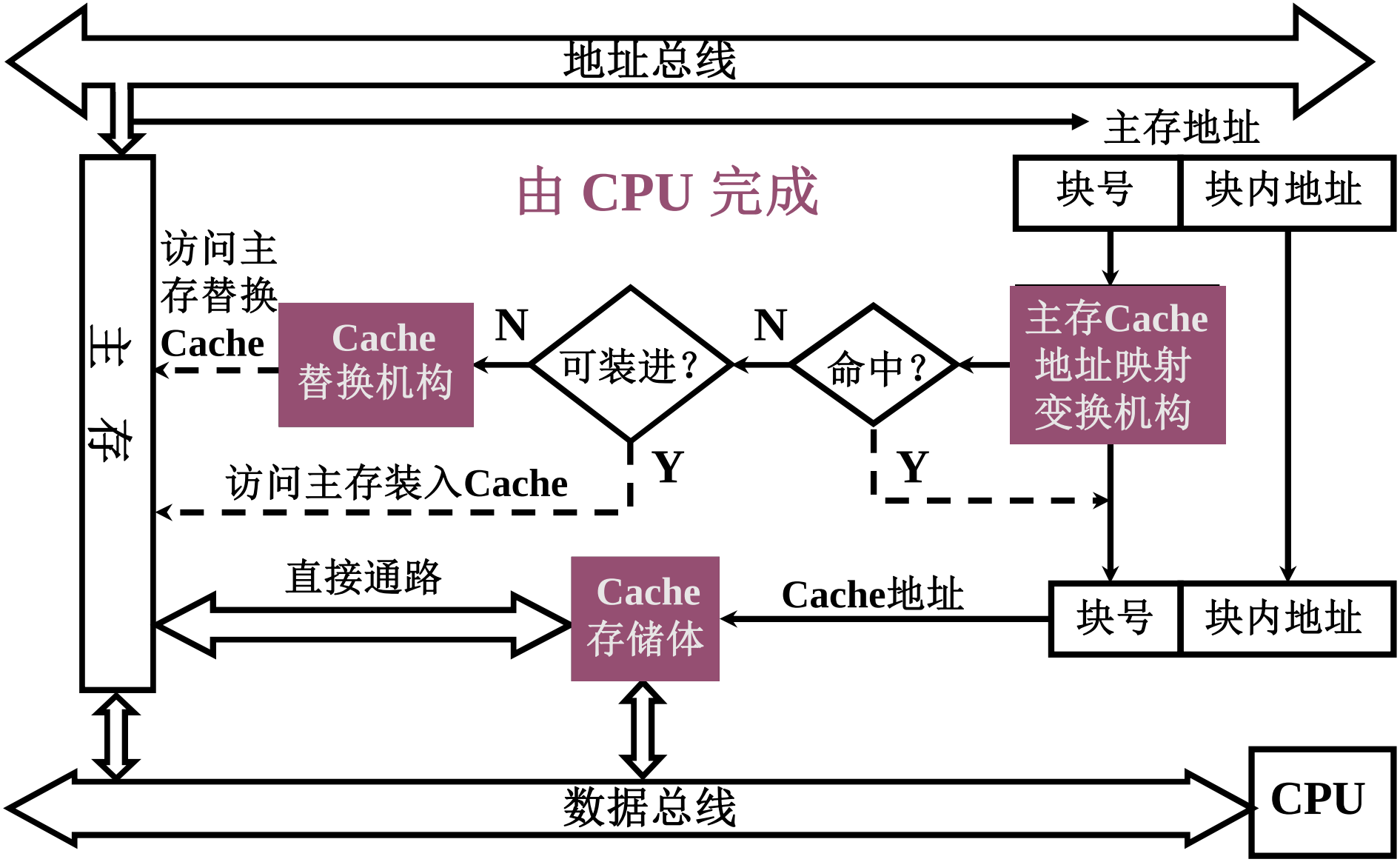
则 $$e = \frac{t_c}{h \times t_c + (1-h) \times t_m} \times 100\%$$

# 存储层次的四个问题 4.3

1. 当把一个块调入高一层(靠近CPU)存储器时，可以放在哪些位置上?
   (映射规则　调入块可以放在哪些位置)
2. 当所要访问的块在高一层存储器中时，如何找到该块?
   (查找算法　如何在映射规则 规定的候选位置查找)
3. 当发生失效时，应替换哪一块?
   (替换算法　规定的候选位置均被别的块占用时)
4. 当进行写访问时，应进行哪些操作?
   (写策略　如何处理写操作, 保证 Cache 和主存的一致性)

# 4. Cache 的基本结构





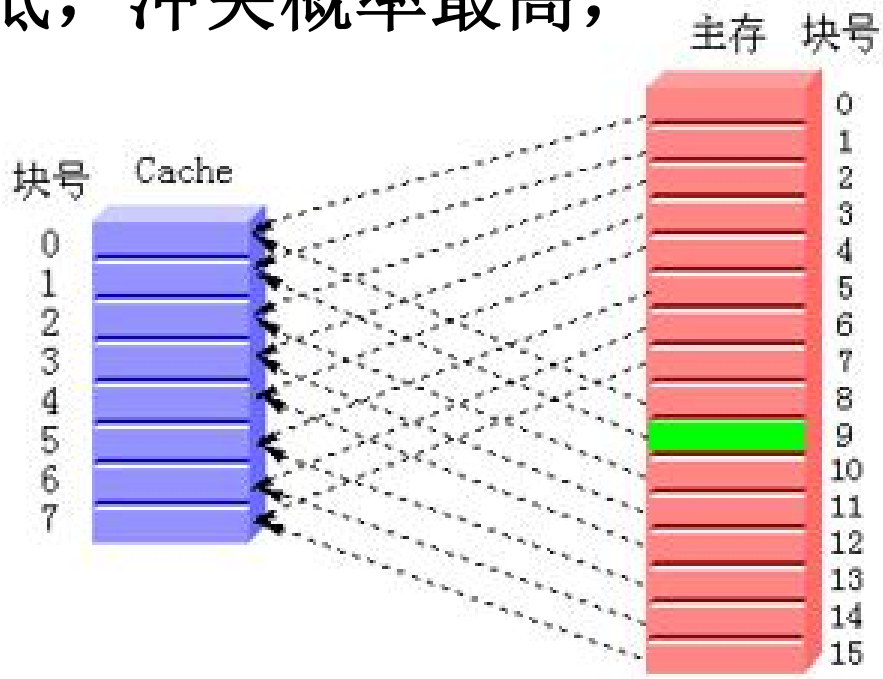
# 二、Cache – 主存的地址映射 4.3

## 1. 直接映射

直接映射：主存中的每一块只能被放置到Cache中唯一的一个位置。（循环分配）

对比：阅览室位置 --只有一个位置可以坐

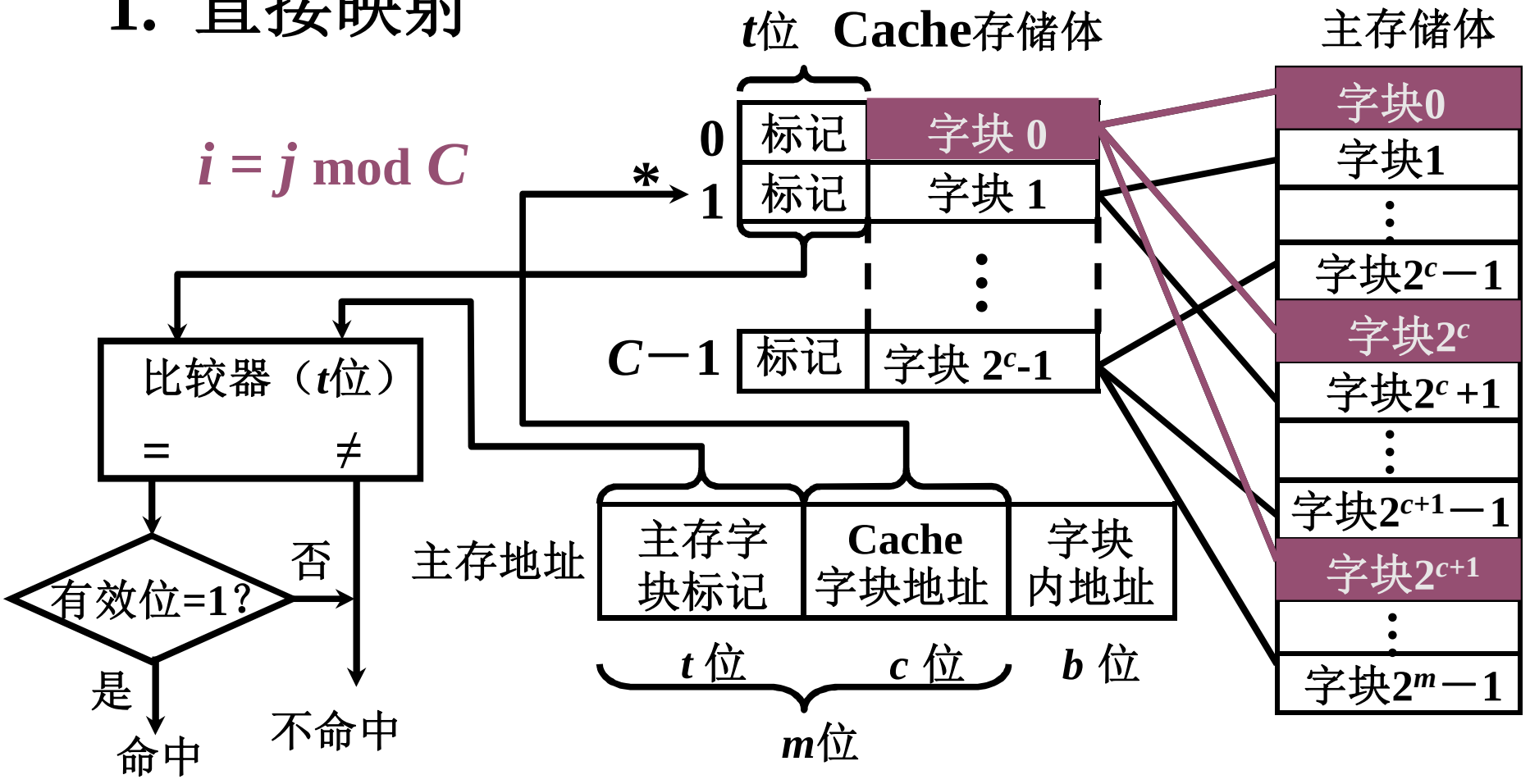
特点：空间利用率最低，冲突概率最高，实现最简单。



# 二、Cache —主存的地址映射



## 1. 直接映射




每个缓存块 *i* 可以和 若干 个 主存块 对应

每个主存块 *j* 只能和 一 个 缓存块 对应

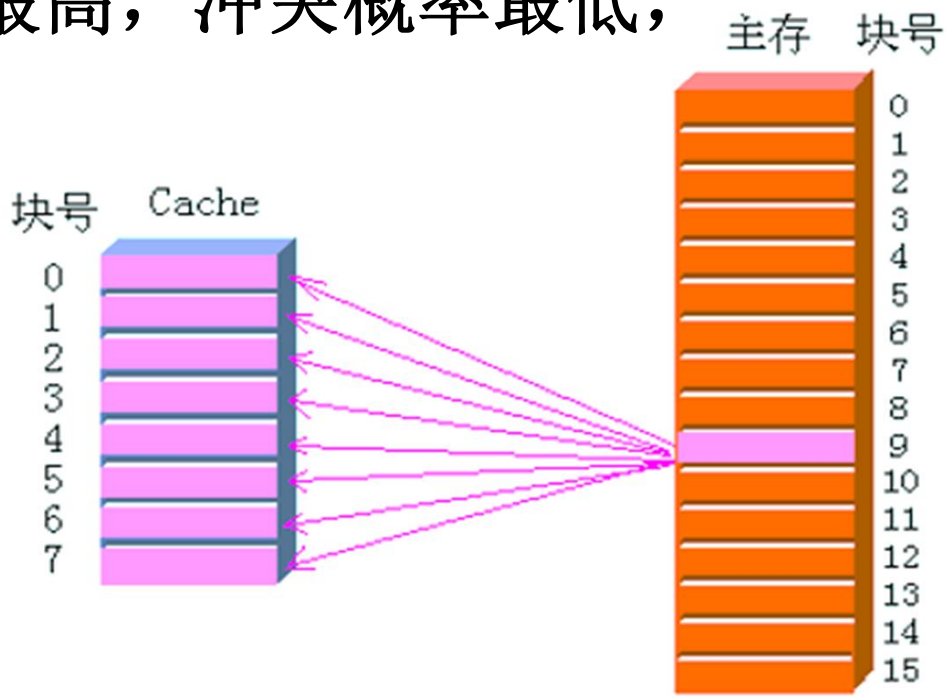
# 2. 全相联映射



全相联：主存中的任一块可以被放置到Cache中的任意一个位置。

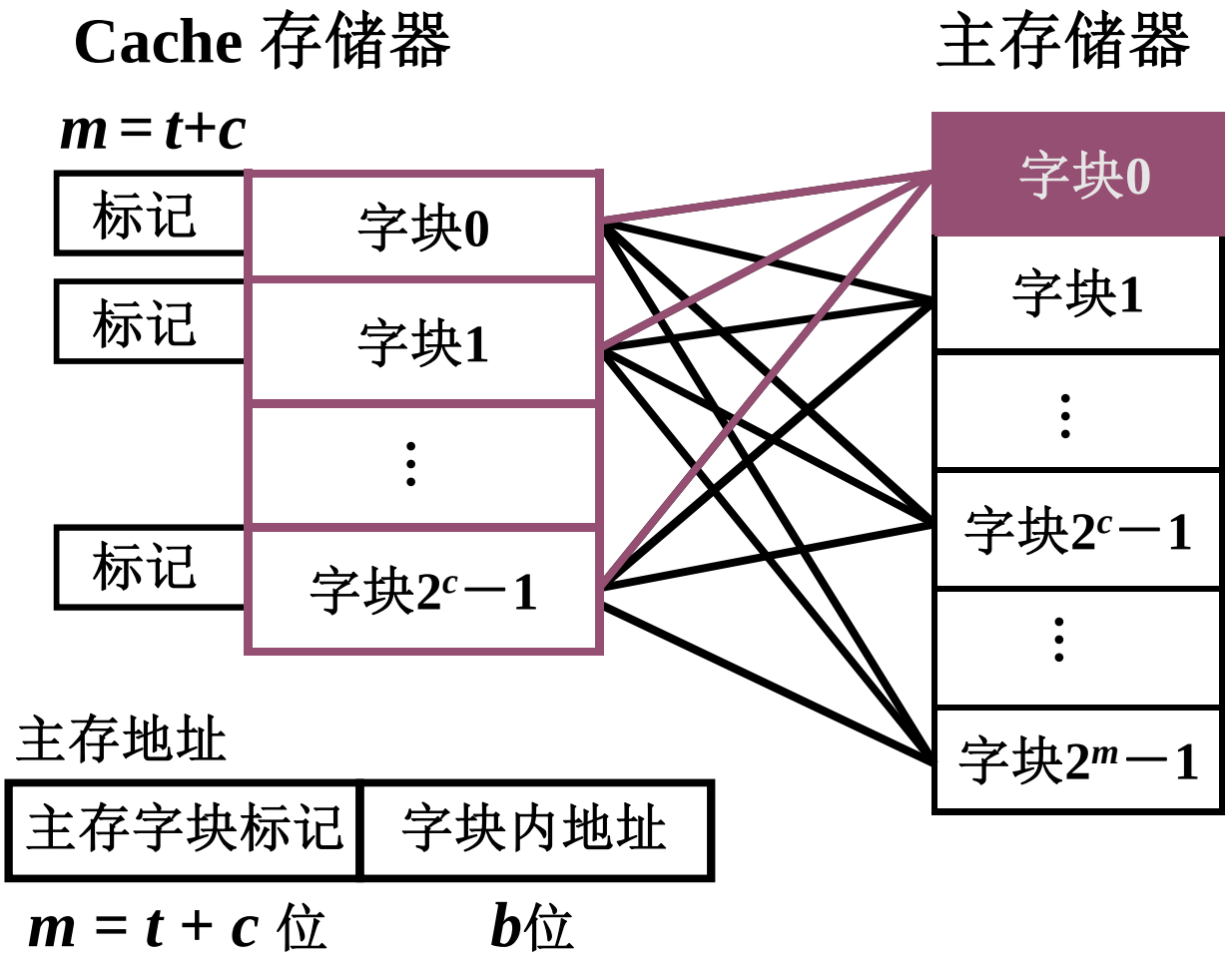
对比：　阅览室位置--随便坐

特点：　空间利用率最高，冲突概率最低，实现最复杂。



# 2. 全相联映射





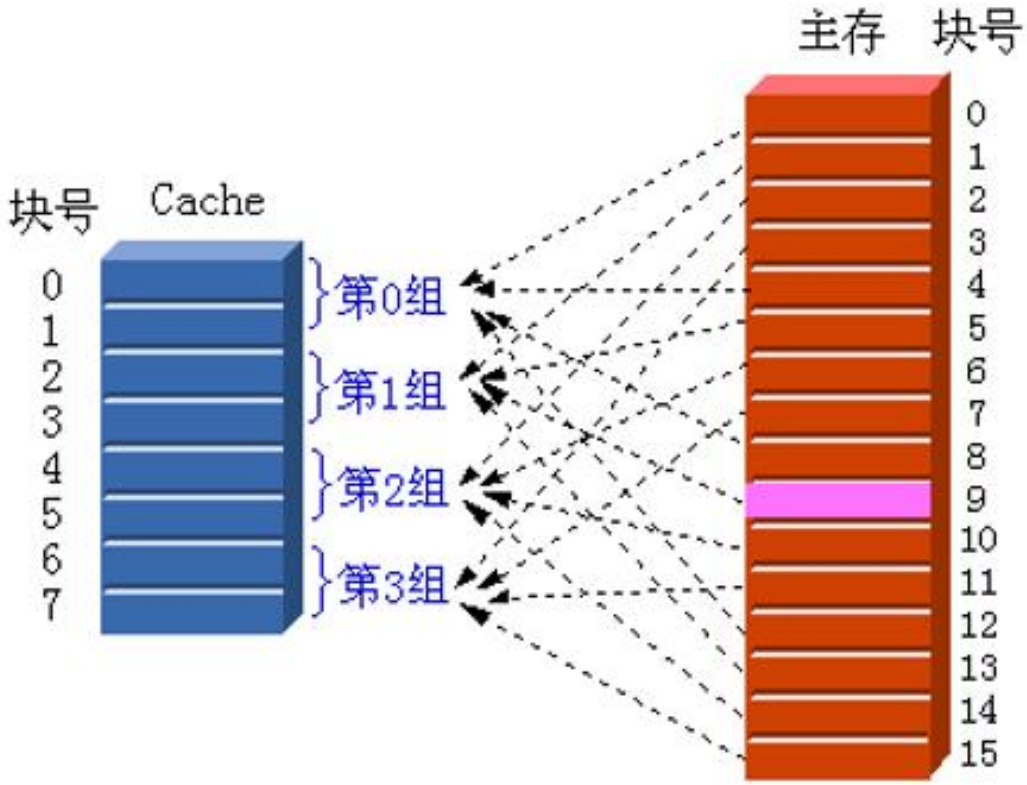

主存 中的 任一块 可以映射到 缓存 中的 任一块

# 3. 组相联映射



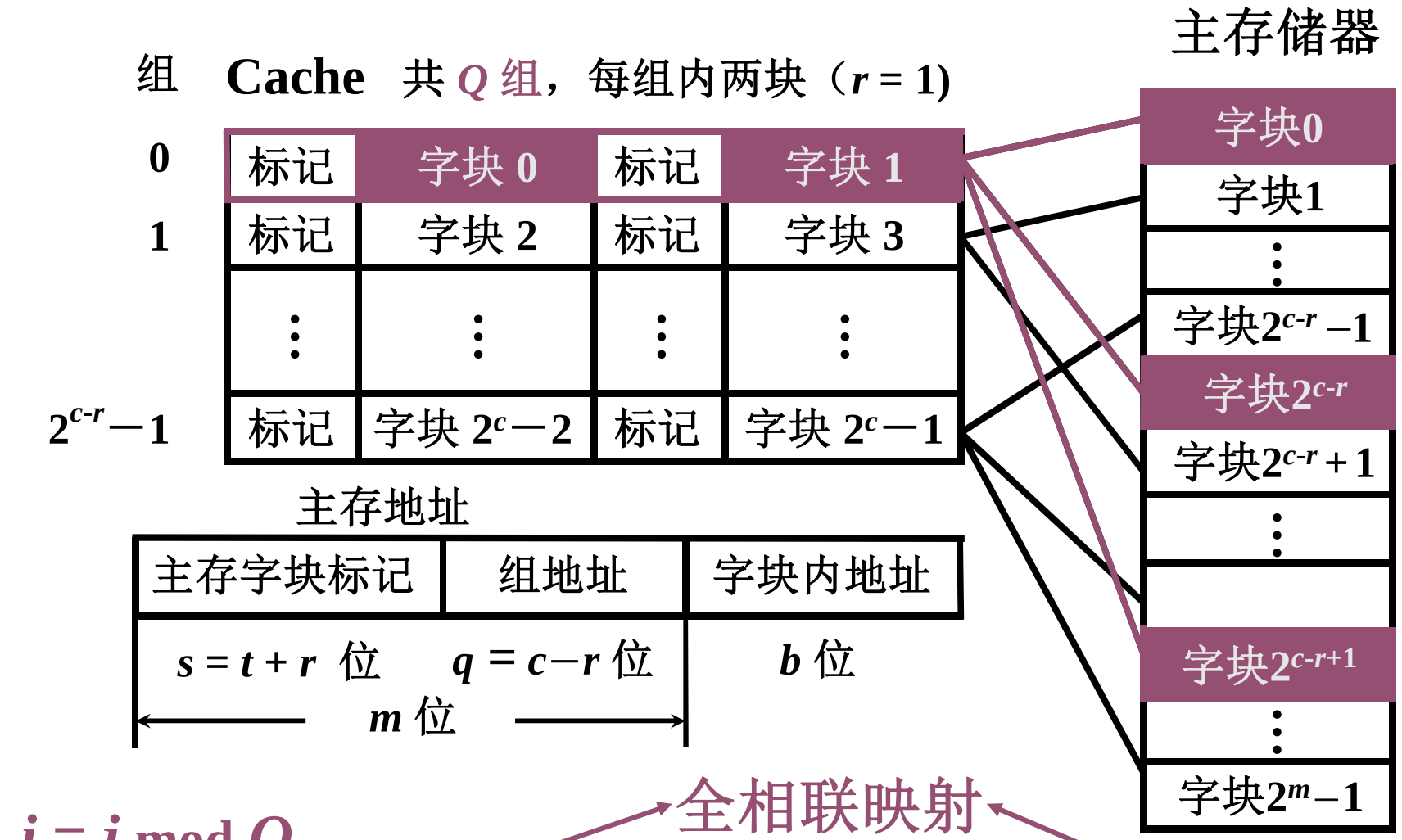
组相联：主存中的每一块可以被放置到Cache中唯一的一个组 中的 任何一个位置。

组相联是直接映射和全相联的一种折中



# 4. 组相联映射






$i = j \bmod Q$

全相联映射

直接映射

某一主存块 $j$ 按模 $Q$ 映射到 缓存 的第 $i$ 组中的 任一块

- ***n*** 路组相联：每组中有***n*** 个块(***n*** = ***M/G*** )，***n*** 称为相联度

  相联度越高，Cache空间的利用率就越高，块冲突概率就越低，失效率也就越低。

| | ***n*** (路数) | ***G*** (组数) |
|---|---|---|
| 全相联 | ***M*** | 1 |
| 直接映射 | 1 | ***M*** |
| 组相联 | $1<n<M$ | $1<G<M$ |

- 大多数计算机的Cache: $n \leq 4$

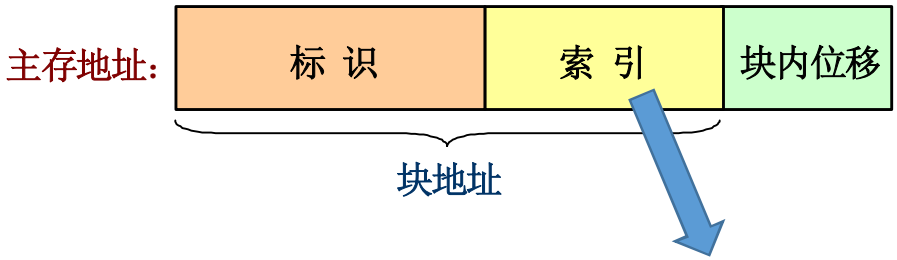
想一想：相联度是否越大越好？

# 四、查找方法

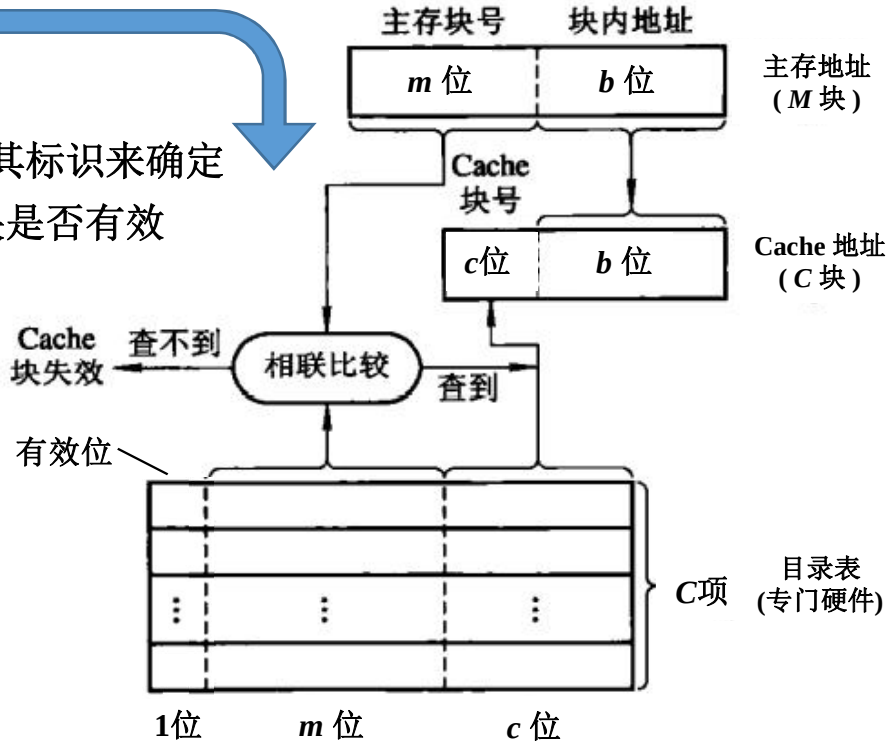
4.3

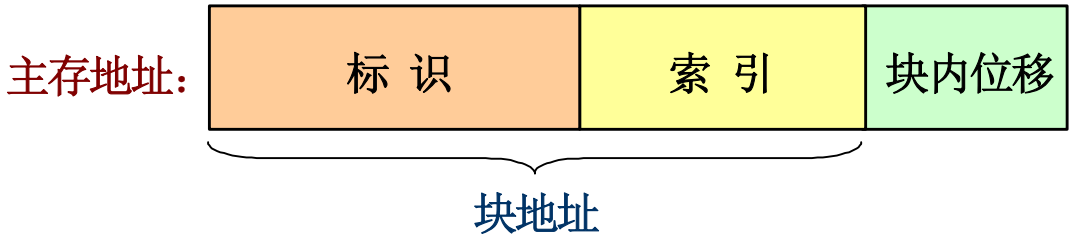
- 当CPU访问Cache时，如何确定Cache中是否有所要访问的块？
- 若有的话，如何确定其位置？

- 通过查找目录表来实现
  - 目录表的结构
    - 主存块的块地址的高位部分，称为标识
    - 在候选位置内，每个主存块能唯一地由其标识来确定
    - 每一项有一个有效位，指出Cache中的块是否有效




组相联：只需查找候选位置所对应的目录表项



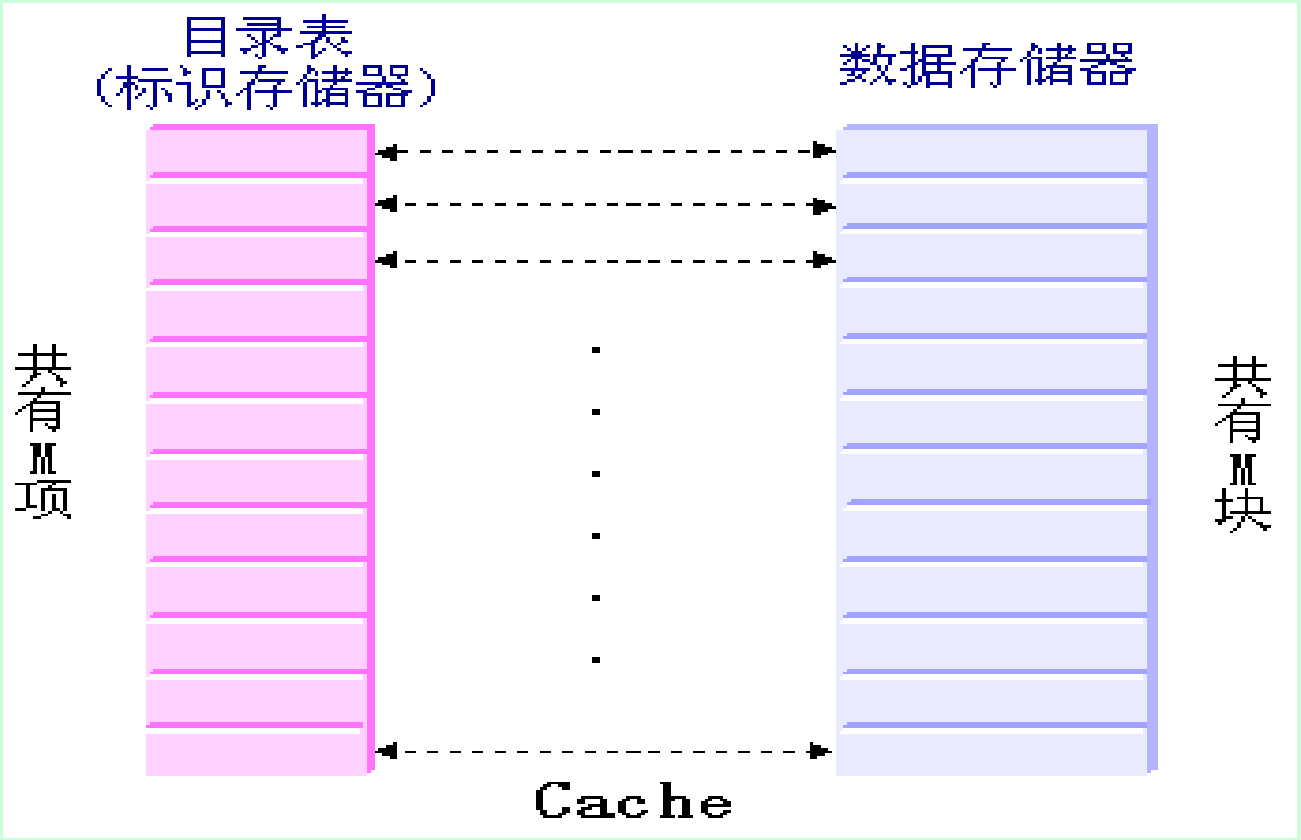



- 块内地址偏移用来从块中选出需要的数据
- 索引域用来选择组（组相联映射）
- 通过比较标识域和有效位来判断是否发生命中

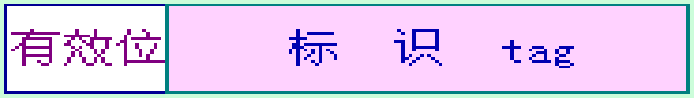
两种情况无需比较

- 块内偏移是没有必要比较的，这是因为Cache命中与否的单位是整个块
- 由于索引域是用来选择被检查的组的，所以对索引域的比较也是多余的

# Cache目录表的结构

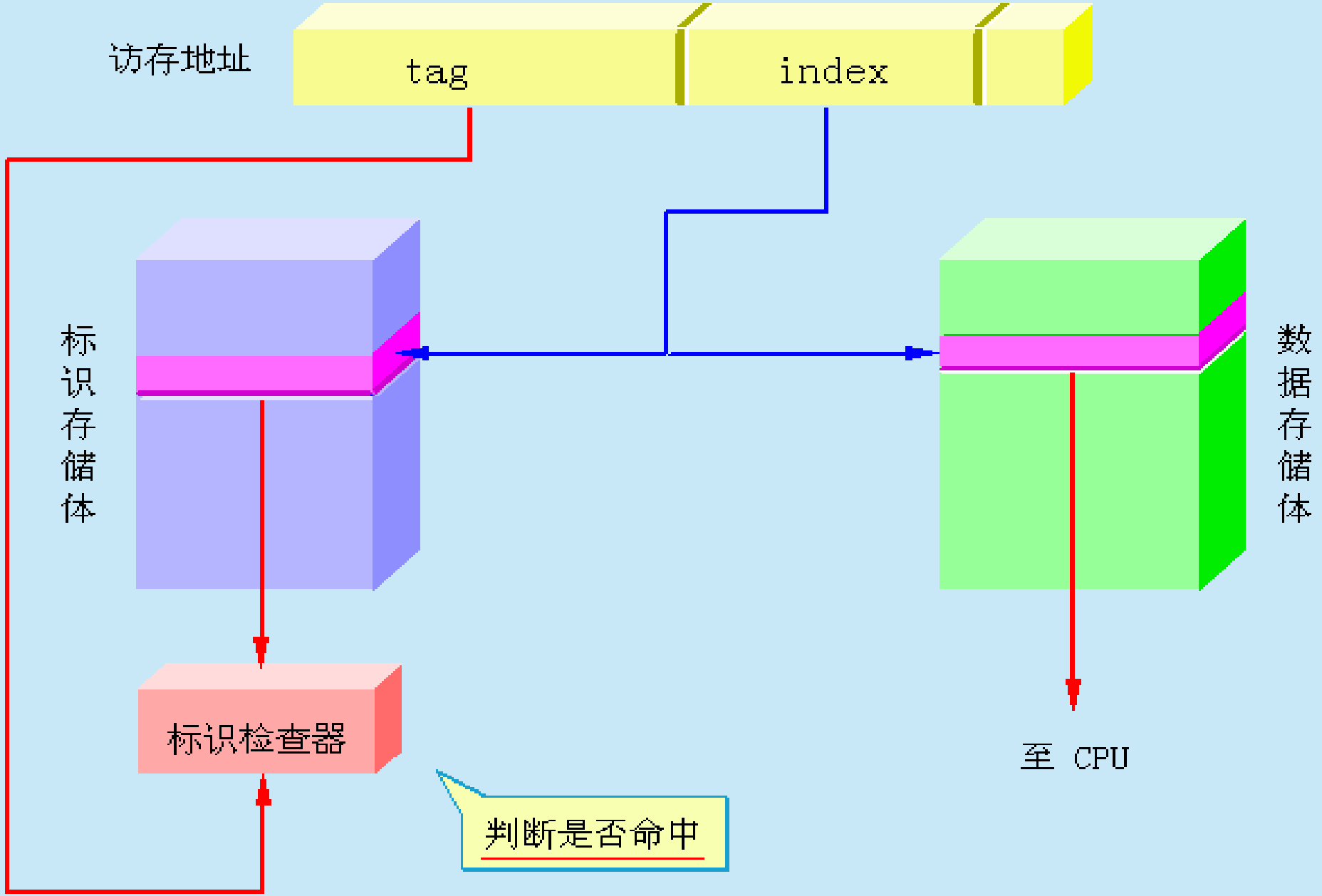



目录表项：

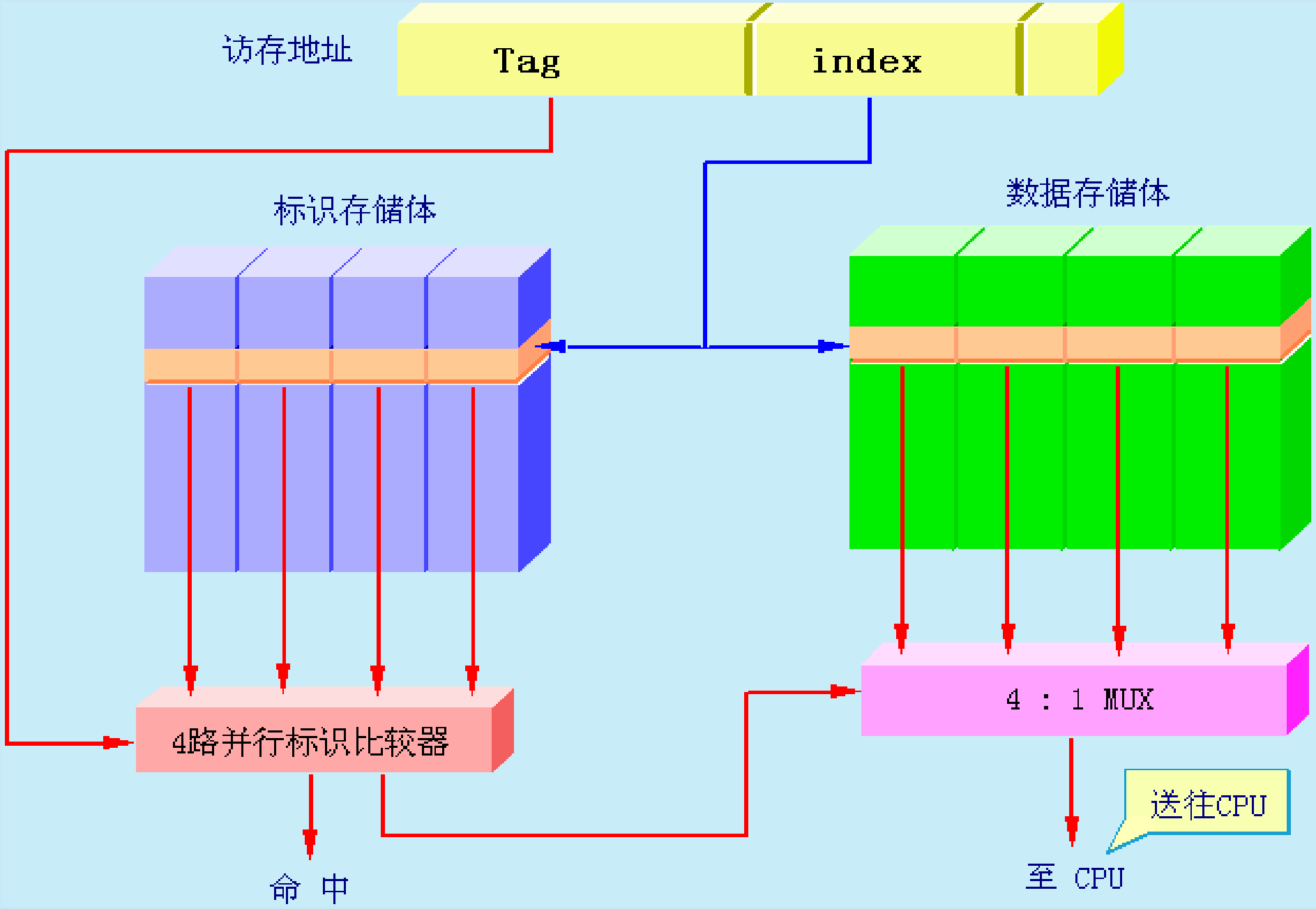



访存地址：

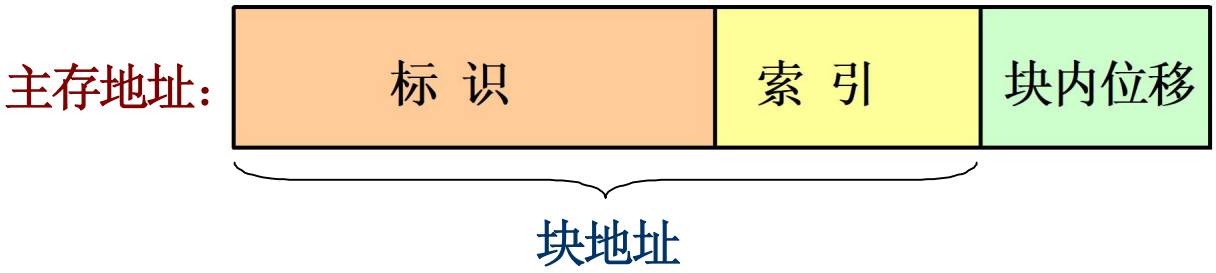
# ◆ 直接映象Cache的查找过程



# ◆ 4 路组相联Cache的查找过程



- 组相联Cache的优缺点
  - 优点：不必采用相联存储器，而是用按地址访问的存储器来实现。
  - 缺点：当相联度 ***n*** 增加时，不仅比较器的个数会增加，而且比较器的位数也会增加。



# 二、地址映射



小结

成本高　不灵活

直接　某一 主存块 只能固定 映射到 某一 缓存块

全相联　某一 主存块 能 映射到 任一 缓存块

组相联　某一 主存块 能 映射到 某一 缓存 组 中的 任一块

# 三、替换算法 4.3

**所要解决的问题：当新调入一块，而该块能够占用的Cache位置已被占满时，替换哪一块?**

- **直接映象Cache中的替换很简单**

  **因为只有一个块，别无选择。**

- **在组相联和全相联Cache中，则有多个块供选择。**
  - **主要的替换算法有三种**
    - **随机法**

      **优点：实现简单**
    - **先进先出法FIFO**
    - **最近最少使用法LRU**

# 三、替换算法



- **最近最少使用法LRU**

**• 选择近期最少被访问的块作为被替换的块。**

**（实际实现起来比较困难）**

**• 实际上：选择最久没有被访问过的块作为被替换的块。**

**• 优点：命中率较高**

- **LRU和随机法分别因其不命中率低和实现简单而被广泛采用。**

**• 模拟数据表明，对于容量足够大的Cache，LRU和随机法的命中率差别不大。**

# 四、写策略 4.3

1. “写”操作所占的比例

Load指令：26%　　Store指令：9%

“写”在所有访存操作中所占的比例：

9%/(100%＋26%＋9%)≈7%

“写”在访问数据Cache操作中所占的比例：

9%/(26%＋9%)≈25%

2. “写”操作必须在确认是否命中后才可进行

3. “写”访问有可能导致Cache和主存内容的不一致

4．两种写策略

- 写直达法：执行“写”操作时，不仅写入Cache，而且也写入下一级存储器。
- 写回法：执行“写”操作时，只写入Cache。仅当Cache中相应的块被替换时，才写回主存。（设置“脏位”）

5．两种写策略的比较

- 写回法的优点：速度快，占用存储器频带低
- 写直达法的优点：易于实现，一致性好

## 6. 写缓冲器

## 7. “写”操作时的调块

- 按写分配(写时取)：写失效时，先把所写单元所在的块调入Cache，再行写入。
- 不按写分配(绕写法)：写失效时，直接写入下一级存储器而不调块。

## 8. 写策略与调块

写回法 —— 按写分配

写直达法 —— 不按写分配

# 五、Cache结构举例 4.3

例子：DEC的Alpha AXP21064中的内部数据Cache

1．简介

容量：8KB

块大小：32B

块数：256

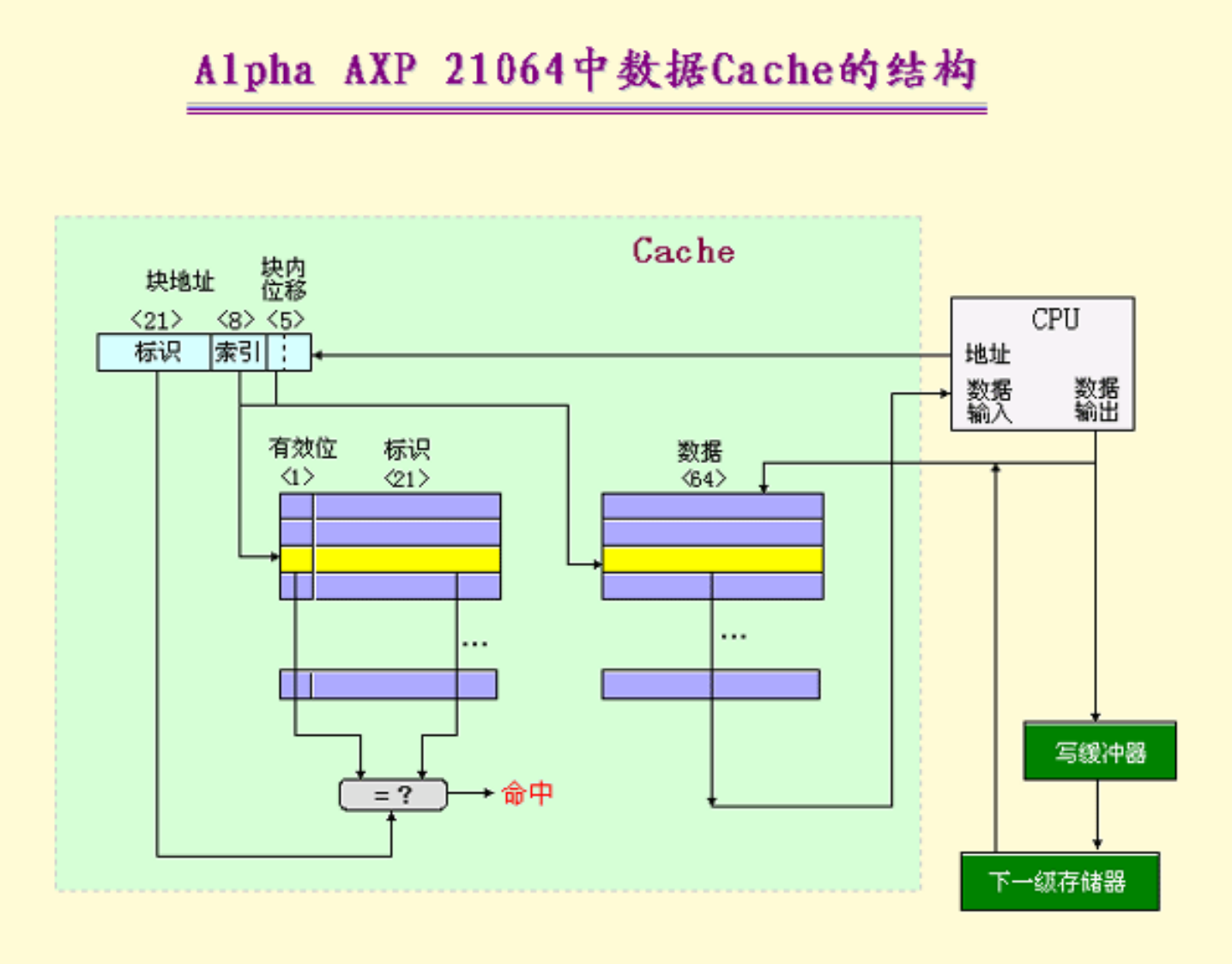
映射方法：直接映射

“写”策略：写直达—不按写分配

写缓冲器大小：4个块

内存地址：34位（块地址29位，块内地址5位）

## 结构图


Alpha AXP 21064中数据Cache的结构
Cache
块地址
块内位移
〈21〉
〈8〉
〈5〉
标识
索引
有效位
〈1〉
标识
〈21〉
数据
〈64〉
CPU
地址
数据输入
数据输出
写缓冲器
下一级存储器
= ?
命中

# 3. 工作过程

① 处理器传送给Cache物理地址

- Cache的容量与索引index、相联度、块大小之间的关系

  Cache的容量 = $2^{index}$×相联度×块大小

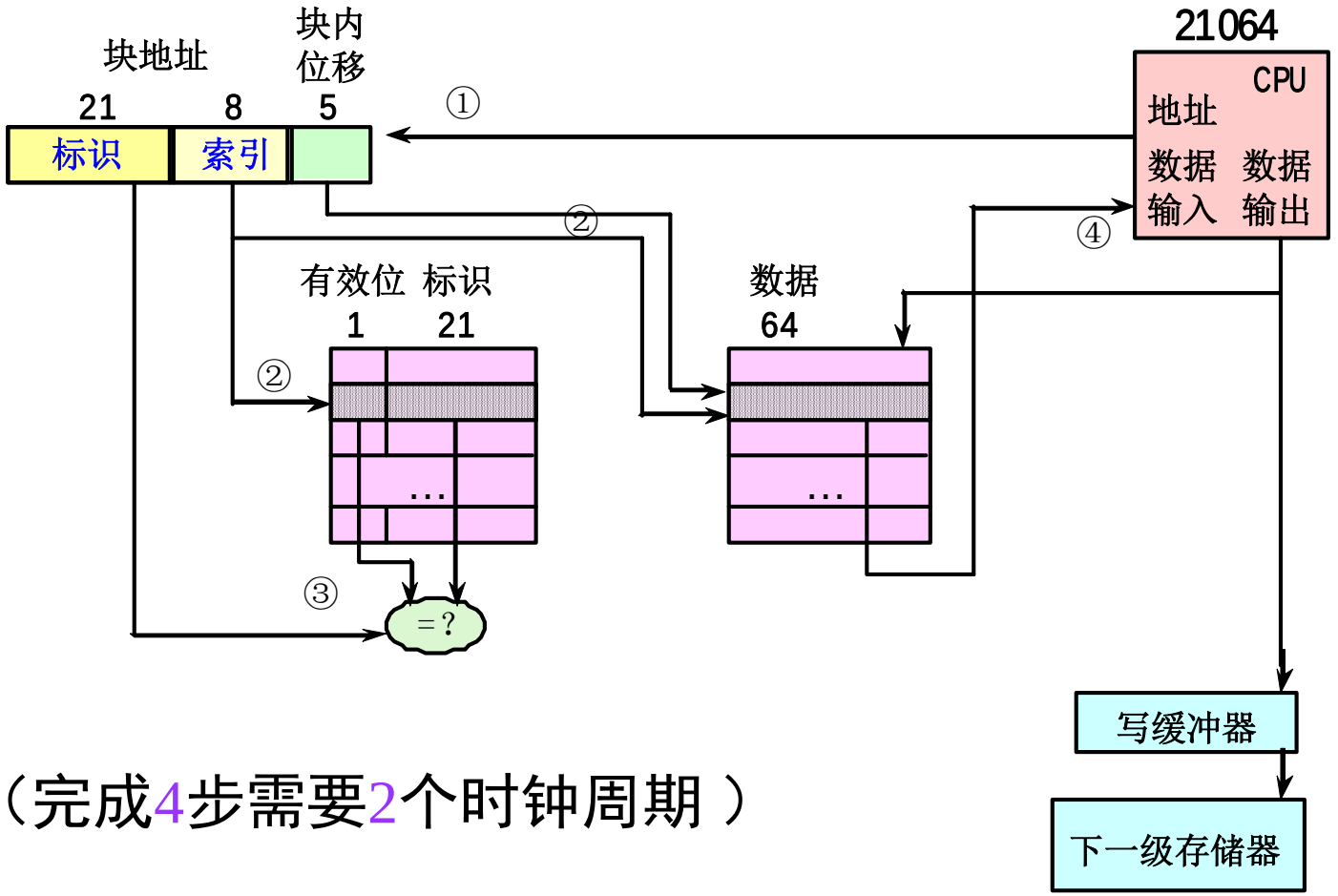
把容量为8192、相联度为1、块大小为32（字节）代入：

$$8192\ (Bytes) = 2^{index} \times 1 \times 32\ (Bytes)$$

➔ 索引 *index*：8位

标识：29－8＝21位

## 3. 工作过程




（完成4步需要2个时钟周期）

# 3. 工作过程

① 处理器传送给Cache物理地址

② 由索引选择标识的过程

→ 根据索引从目录项中读出相应的标识和有效位

③ 从Cache中读出标识之后，用来同从CPU发来的块地址中标志域部分进行比较

→ 为了保证包含有效的信息，必须要设置有效位

④ 如果有一个标识匹配，且标志位有效，则此次命中

→ 通知CPU取走数据

- “写”访问命中
  - 前三步一样，只有在确认标识匹配后才把数据写入

- 提高写入性能的方法：设置写缓冲器

  （提高“写”访问的速度）
  - 写缓冲器按字寻址，含有4个块，块大小为4个字。
  - 当要进行写入操作时，如果写缓冲器不满，那么就把数据和完整的地址写入缓冲器。对CPU而言，本次“写”访问已完成，CPU可以继续往下执行。由写缓冲器负责把该数据写入主存。
  - 在写入缓冲器时，要进行写合并检查。即检查本次写入数据的地址是否与缓冲器内某个有效块的地址匹配。如果匹配，就把新数据与该块合并 。

# 发生读不命中与写不命中时的操作 4.3

- **读不命中：** 向CPU发出一个暂停信号，通知它等待，并从下一级存储器中新调入一个数据块（32字节）
  - Cache与下一级存储的数据通路宽度为16B，传送一次需5个周期，因此，一次传送需要10个周期
- **写不命中：** 将使数据“绕过”Cache，直接写入主存。
  - 写直达 –不按写分配
- 因为是写直达，所以替换时不需要写回

# 六、改进Cache性能 4.3

平均访存时间＝命中时间＋失效率×失效开销

可以从三个方面改进Cache的性能：

(1) 降低失效率

例如：增加块大小、提高相联度

(2) 减少失效开销

例如：多级Cache、写缓存、请求字优先处理

(3) 减少Cache命中时间

例如：容量小且结构简单的Cache

# 例1



- **某计算机字长32位，采用直接映射cache,主存容量4MB，cache数据存储体容量为4KB，字块长度为8个字。**
  1. 画出直接映射方式下主存地址划分情况。
  2. 设cache初始状态为空，若CPU顺序访问0-99号单元，并从中读出100个字，假设访问主存一次读一个字，并重复此顺序10次，请计算cache命中率。
  3. 如果cache存取时间是2ns，主存访问时间是20ns，平均访问时间是多少。
  4. cache-主存系统访问效率。

# 例1



- **某计算机字长32位，采用直接映射cache,主存容量4MB，cache数据存储体容量为4KB，字块长度为8个字。**

1. 画出直接映射方式下主存地址划分情况。

解：主存字数：4MB/4B=1M=$2^{20}$（20位地址）

Cache字数：4KB/4B=1K=$2^{10}$（10位地址）

字块长度：8个字=$2^3$（3位块内地址）

| 字块标记 | 块号 | 块内地址 |
| --- | --- | --- |
| 10位 | 7位 | 3位 |

例1

| 字块标记 | 块号 | 块内地址 |
| --- | --- | --- |
| 10位 | 7位 | 3位 |

2. 设cache初始状态为空，若CPU顺序访问0-99号单元，并从中读出100个字，假设访问主存一次读一个字，并重复此顺序10次，请计算cache命中率。

解：　0地址 ➔ 0000 0000 00 00 0000 0 000

99地址 ➔ 0000 0000 00 00 0110 0 011

**地址范围覆盖块号0-12，共13个字块。**

**访问第一遍：**发生13次调块操作，填充缓存，其余100-13=87次均命中。

**访问剩余9遍：**全部命中，共100 × 9=900次。

例1

| 字块标记 | 块号 | 块内地址 |
| --- | --- | --- |
| 10位 | 7位 | 3位 |

2. 设cache初始状态为空，若CPU顺序访问0-99号单元，并从中读出100个字，假设访问主存一次读一个字，并重复此顺序10次，请计算cache命中率。

**解：**

**根据命中率的定义：**

$$H=(87+900)/(100\times 10)\times 100\%=98.7\%$$

例1

| 字块标记 | 块号 | 块内地址 |
| --- | --- | --- |
| 10位 | 7位 | 3位 |

3. 如果cache的存取时间是2ns，主存访问时间是20ns，平均访问时间是多少

**解：**

**根据平均访问时间的定义：**

$$T_a = H \times t_c + (1 - H) \times t_m$$

$$= 98.7\% \times 2 + (1 - 98.7\%) \times 20$$

$$= 2.234\text{ns}$$

例1


4. cache-主存系统的访问效率

解：

根据Cache-主存系统访问效率的定义：

$$e=t_c/T_a$$

$$=2ns/2.234ns$$

$$=0.895$$

【2016 统考真题】有如下 C 语言程序段：

```
for(k=0; k<1000; k++)
    a[k] = a[k]+ 32;
```

若数组 a 和变量 k 均为 int 型，int 型数据占 4B，数据 Cache 采用直接映射方式，数据区大小为 1KB、块大小为 16B，该程序段执行前 Cache 为空，则该程序段执行过程中访问数组 a 的 Cache 缺失率约为

A. 1.25%　　B. 2.5%　　C. 12.5%　　D. 25%

解：

| 字块标记 | 块号 | 块内地址 |
| --- | --- | --- |
| 若干位 | 6位 | 2位 |

1KB=64个块　1块=4字

**0-999：00 00 00 00 00 ~ 11 11 10 01 11**

**读缺失：每圈64次/256次访存≈0.25**

**写缺失：每圈0次/256次访存=0**

**总　共：每圈64次/512次访存≈0.125**