

第五章 存储系统

第四讲 映象规则及其变换

谢长生

武汉光电国家研究中心

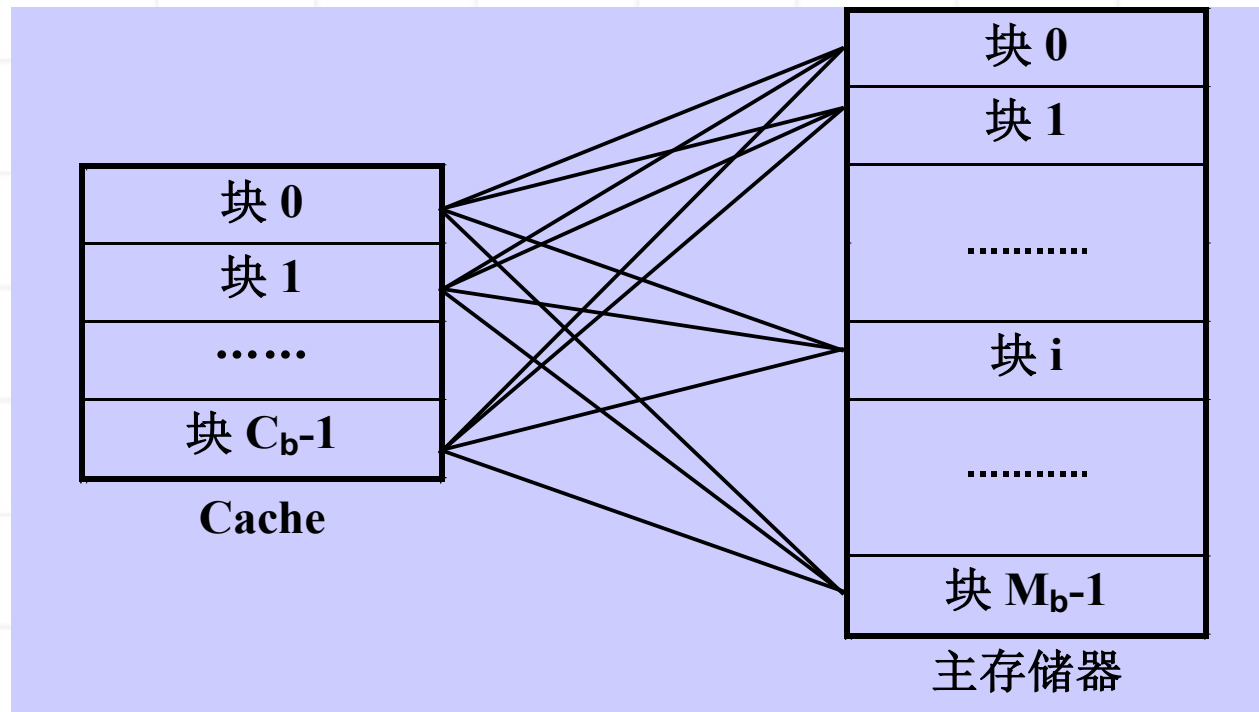
华中科技大学计算机科学与技术学院



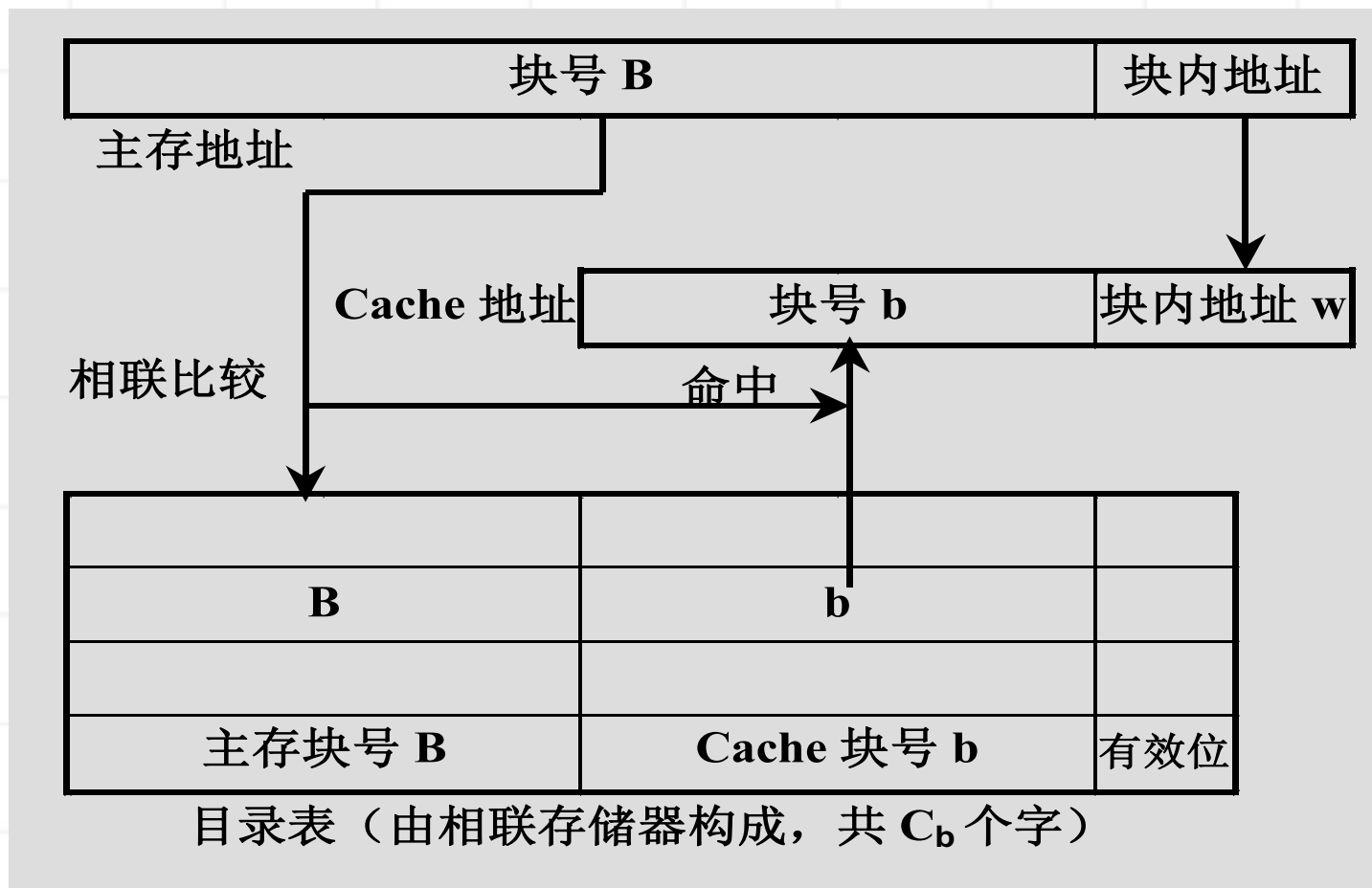
计算机系统结构

1. 全相联映象及其变换

映象规则：主存的任意一块可以映象到
Cache
中的任意一块。（映象关系有 $C_b \times M_b$ 种）



全相联地址变换规则 用硬件实现非常复杂





2. 直接映象及其变换

映象规则：

主存储器中一块只能映象到**Cache**的一个特定的块中。

Cache地址的计算公式：

$$b = B \bmod C_b$$

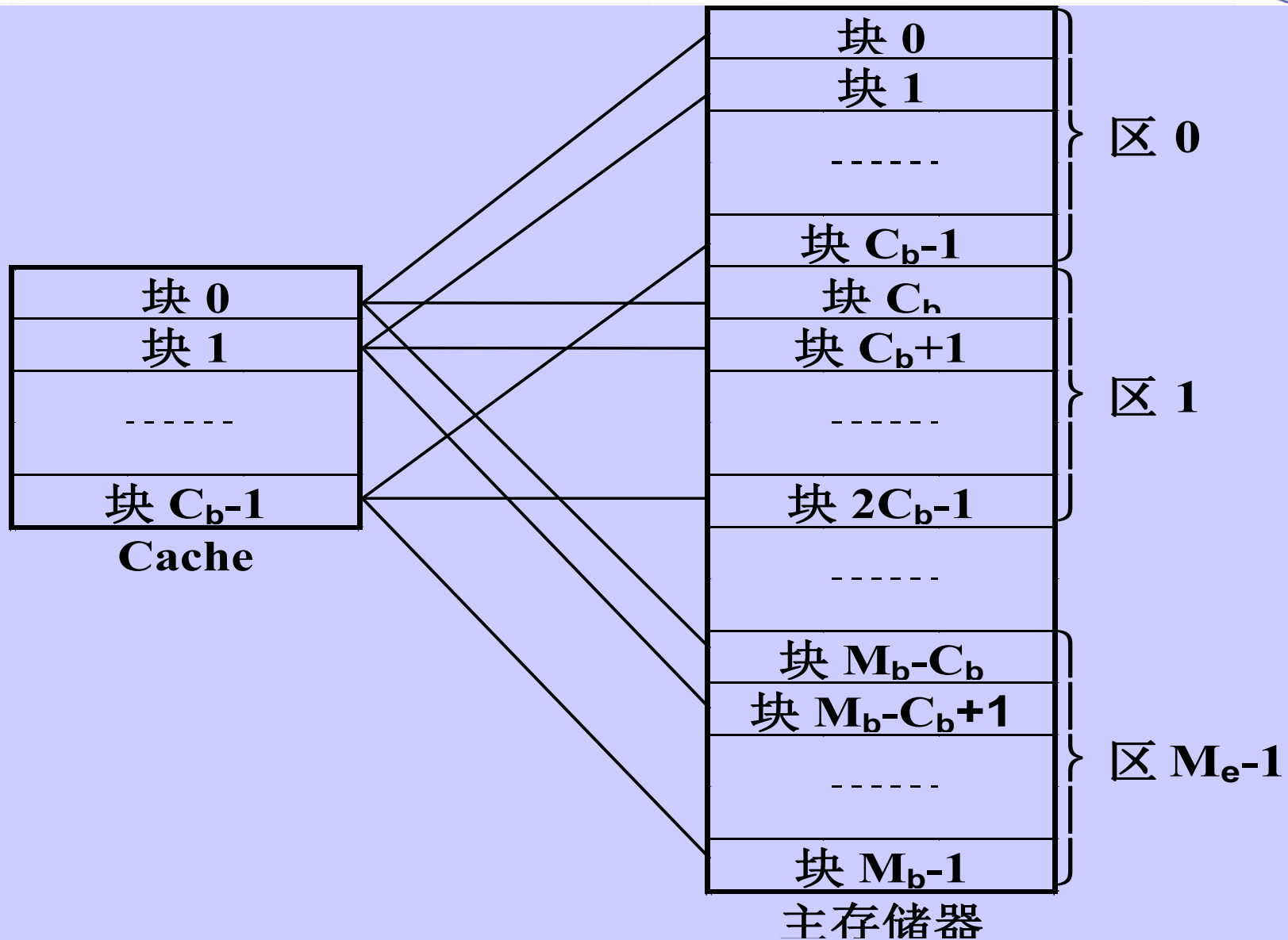
其中：b为Cache块号，

B是主存块号，

C_b 是Cache块数。

实际上，**Cache**地址与主存储器地址的低位部分完全相同。

直接映象方式的地址映象规则

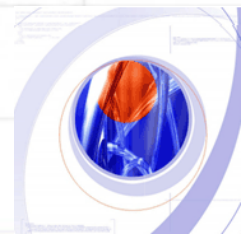


直接映象方式的地址变换过程:

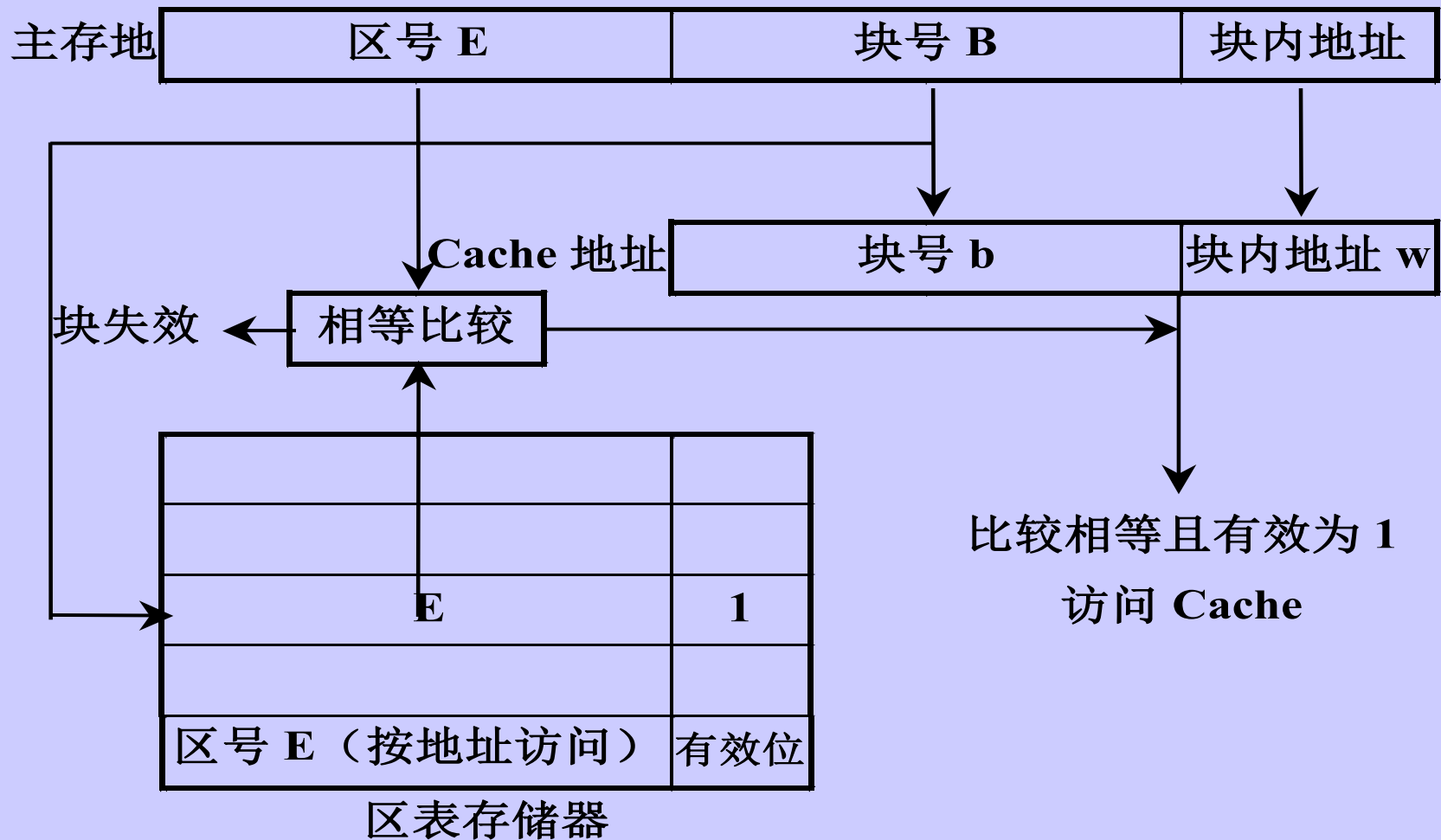
用主存地址中的块号**B**去访问区号存储器，把读出来的区号与主存地址中的区号**E**进行比较：

比较结果相等，有效位为1，则Cache命中，否则该块已经作废。

比较结果不相等，有效位为1，Cache中的该块是有用的，否则该块是空的。



直接映象方式的地址变换规则



2. 直接映象及其变换的优缺点

- 主要优点:

硬件实现很简单，不需要相联访问存储器

访问速度也比较快，实际上不需要进行地址变换

- 主要缺点:

块的冲突率比较高。

3. 组相联映象及其变换

映象规则:

主存和**Cache**按同样大小划分成块和组。

主存和**Cache**的组之间采用直接映象方式。

在两个对应的组内部采用全相联映象方式。

组相联映象方式的优点:

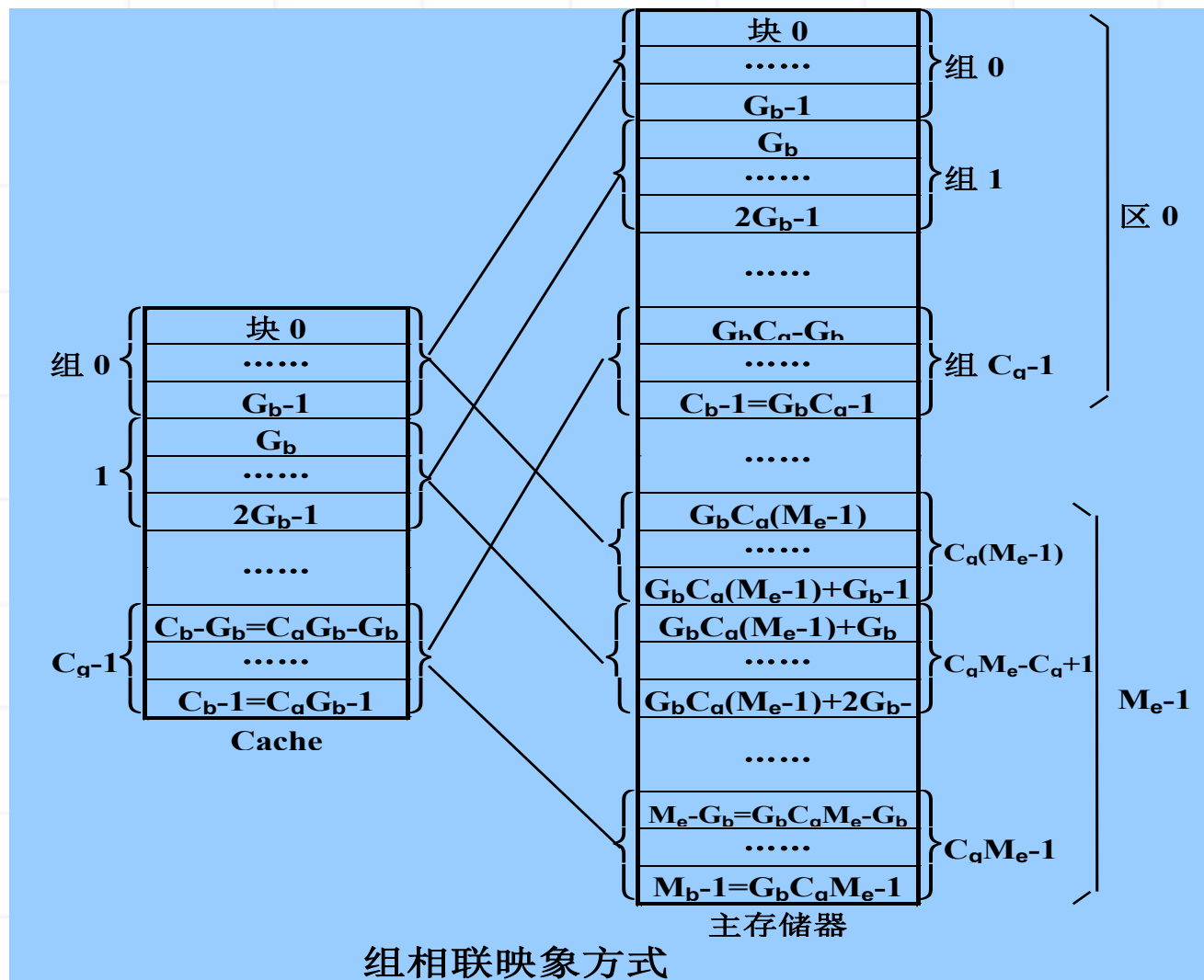
块的冲突概率比较低,

块的利用率大幅度提高,

块失效率明显降低。

组相联映象方式的缺点:

实现难度和造价要比直接映象方式高。



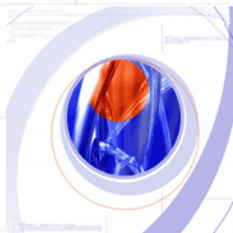
组相联映象的地址变换过程:

用主存地址中的组号**G**按地址访问块表存储器。

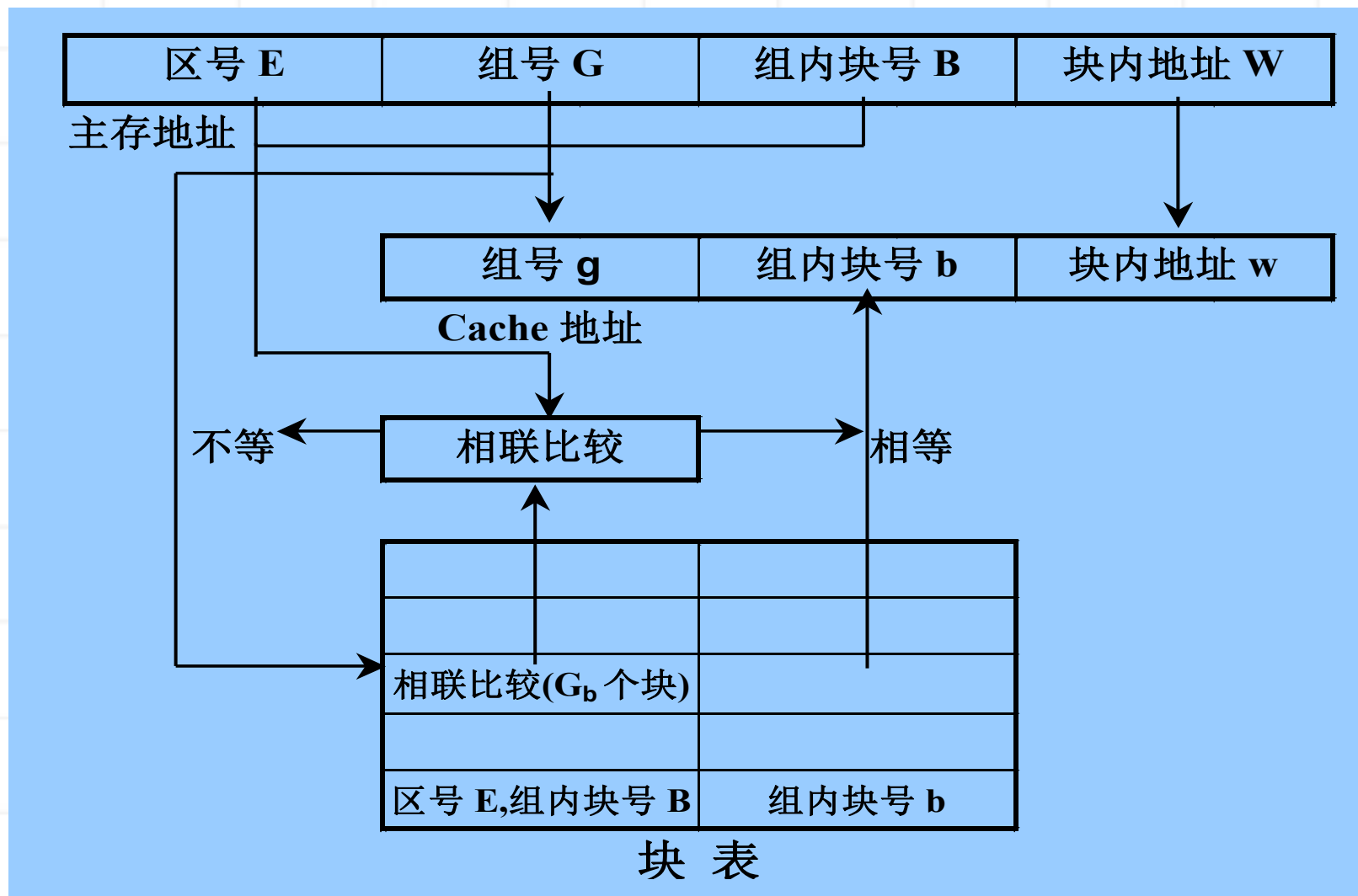
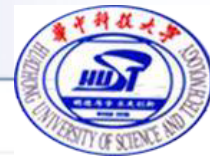
把读出来的一组区号和块号与主存地址中的区号和块号进行**相联比较**。

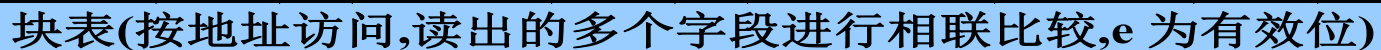
如果有相等的，表示**Cache**命中；

如果全部不相等，表示**Cache**没有命中。



• 组相联映象的地址变换





3、如何写？

写穿 (Write through) :

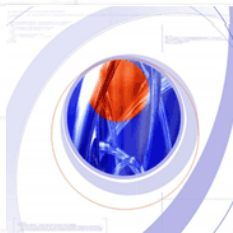
写cache, 同时写内存

优点: 一致性好, 缺点: 慢

写回 (write back):

只写cache, 只有被替换时才写回内存

优点: 快; 缺点: 出错率高一倍



4、如何换？ Cache替换算法及其实现使用的场合：

直接映象方式实际上不需要替换算法

全相联映象方式的替换算法最复杂

主要用于组相联、段相联等映象方式中

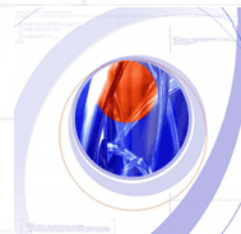
要解决的问题：

记录每次访问**Cache**的块号

在访问过程中，对记录的块号进行管理

根据记录和管理结果，找出替换的块号

主要特点：全部用硬件实现



1. 轮换法及其实现

用于组相联映象方式中，有两种实现方法。

方法一：每块一个计数器

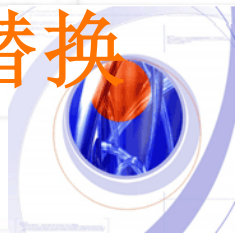
在块表内增加一个替换计数器字段，

计数器的长度与Cache地址中的组内块号字段的长度相同。

替换方法及计数器的管理规则：

新装入或替换的块，它的计数器清0，同组其它块的计数器都加“1”。

在同组中选择计数器的值最大的块作为被替换的块。



方法二：每组一个计数器

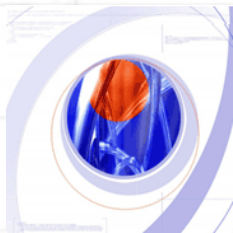
替换规则和计数器的管理：

本组有替换时，计数器加“1”，

计数器的值就是要被替换出去的块号。

轮换法的优点： 实现比较简单，能够利用历史上的块地址流情况

轮换法的缺点： 没有利用程序的局部性特点



2. LRU算法及其实现

为每一块设置一个计数器

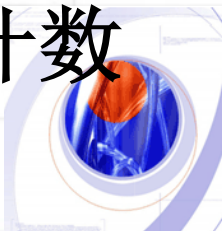
计数器的长度与块号字段的长度相同

计数器的使用及管理规则：

新装入或替换的块，计数器清**0**，同组中其它块的计数器加**1**。

命中块的计数器清**0**，同组的其它计数器中，凡计数器的值小于命中块计数器原来值的加**1**，其余计数器不变。

需要替换时，在同组的所有计数器中选择计数值最大的计数器，它所对应的块被替换。



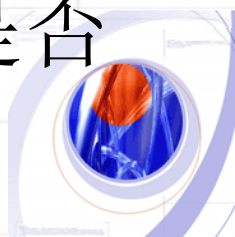
LRU算法的优缺点

主要优点：

- (1)命中率比较高，
- (2)能够比较正确地利用程序的局部性特点，
- (3)充分地利用历史上块地址流的分布情况，
- (4)是一种堆栈型算法，随着组内块数增加，命中率单调上升。

主要缺点：

控制逻辑复杂，因为增加了判断和处理是否命中的情况。





谢谢大家！

