北京郵電大學
BEIJING UNIVERSITY OF POSTS AND TELECOMMUNICATIONS

3.6 Cache存储器

提纲

3.6.1 基本原理

3.6.2 主存与Cache的地址映射

3.6.3 替换策略

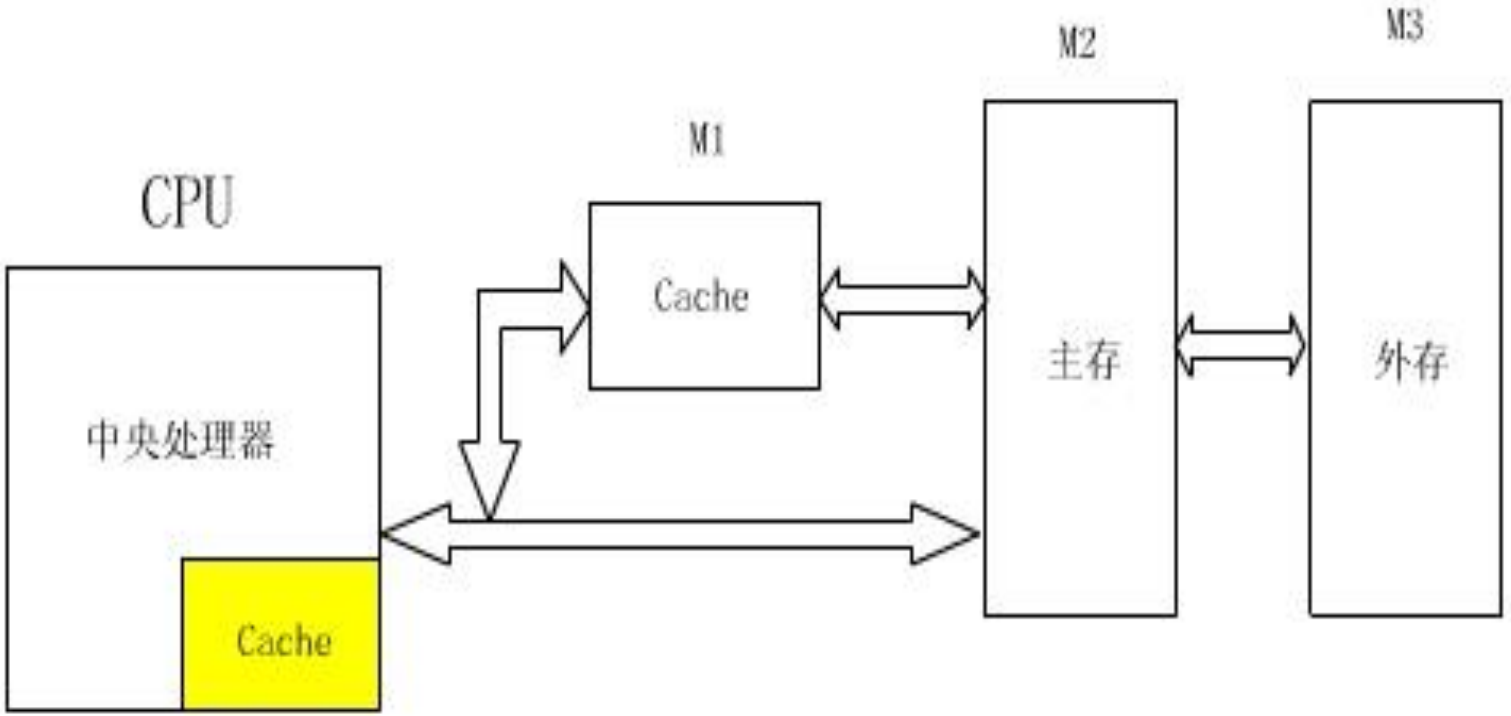
3.6.4 写操作策略

3.6.5 Pentium PC的Cache

3.6.1 基本原理

1、功能

- 解决CPU和主存之间的速度不匹配问题
- 一般采用高速的SRAM构成
- 原理基于程序运行中具有的空间局部性和时间局部性特征



3.6.1 基本原理

- CPU和主存之间的速度差别很大采用两级或多级Cache系统
- 早期的一级Cache在CPU内，二级在主板上
- 现在的CPU内带L1 Cache和L2 Cache

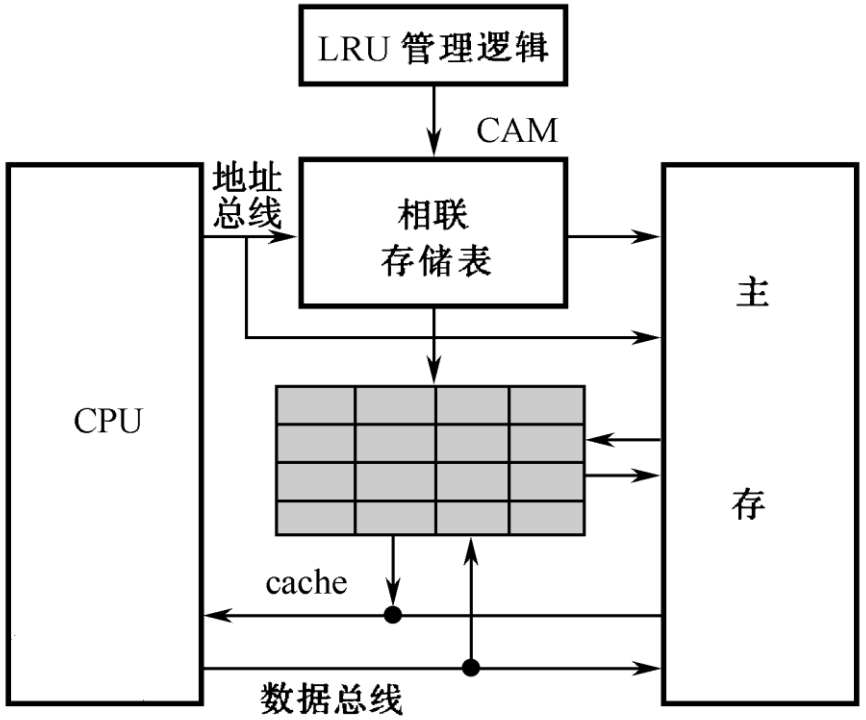
- 全由硬件调度，对用户透明

3.6.1 基本原理

2、Cache基本原理

- 若Cache在CPU外，其控制逻辑与主存控制逻辑合在一起，若在CPU内，由CPU提供控制逻辑

- 地址映射；
- 替换策略；
- 写一致性；
- 性能评价。



3.6.1 基本原理

- Cache基本原理小结：
  - cache是介于CPU和主存间的小容量存储器，但存取速度比主存快。主存容量配置几百MB的情况下，cache的典型值是几百KB。cache能高速地向CPU提供指令和数据，从而加快了程序的执行速度。
  - 从功能上看，它是主存的缓冲存储器，由高速的SRAM组成。为追求高速，包括管理在内的全部功能由硬件实现，因而对程序员是透明的。
  - Cache的设计依据：CPU这次访问过的数据，下次有很大的可能也是访问附近的数据。
  - CPU与Cache之间的数据传送是以字为单位
  - 主存与Cache之间的数据传送是以块（若干字）为单位
  - CPU读主存时，便把地址同时送给Cache和主存，Cache控制逻辑依据地址判断此字是否在Cache中，若在此字立即传送给CPU，否则，则用主存读周期把此字从主存读出送到CPU，与此同时，把含有这个字的整个数据块从主存读出送到cache中。

3.6.1 基本原理

2、Cache的命中率

- 从CPU来看，增加一个cache的目的，就是在性能上使主存的平均读出时间尽可能接近cache的读出时间
- 为了达到这个目的，在所有的存储器访问中由cache满足CPU需要的部分应占很高的比例，即cache的命中率应接近于1
- 由于程序访问的局部性，实现这个目标是可能的。

3.6.1 基本原理

- 在一个程序执行期间，设N_c表示cache完成存取的总次数，Nm表示主存完成存取的总次数，h定义为命中率，

$$h=N_c/（N_c+N_m）$$

- 若t_c表示命中时的cache访问时间，t_m表示未命中时的主存访问时间，1-h表示未命中率，则cache/主存系统的平均访问时间t_a为：

$$t_a=h*t_c+（1-h)t_m$$

- 我们追求的目标是，以较小的硬件代价使cache/主存系统的平均访问时间t_a越接近t_c越好。
- 设$r=t_m/t_c$表示主存慢于cache的倍率，e表示访问效率，则有

$$e=t_c/t_a=t_c/（h*t_c+（1-h）*t_m）$$

$$=1/（h+（1-h）*r）=1/（r+（1-r)*h）$$

- 由表达式看出，为提高访问效率，命中率h越接近1越好，r值以5—10为宜，不宜太大
- 命中率h与程序的行为、cache的容量、组织方式、块的大小有关。

3.6.1 基本原理

- 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

命中率 $$h = \frac{N_c}{N_c + N_m}$$

Cache/主存系统的平均访问时间 $$t_a = ht_c + (1-h)t_m$$

访问效率 $$e = \frac{t_c}{t_a} = \frac{1}{r + (1-r)h}$$

Cache与内存的速度比 $$r = t_m / t_c$$

3.6.1 基本原理

- 例解:

h=Nc/（Nc+Nm）=1900/(1900+100)=0.95

r=tm/tc=250ns/50ns=5

e=1/(r+(1-r)h)=1/(5+(1-5)×0.95)=83.3%

ta=tc/e=50ns/0.833=60ns

3.6.2 主存与Cache的地址映射

- Cache的内容是主存内容的一个子集，需要把主存地址定位到Cache中，称为地址映射
- 无论选择那种映射方式，都要把主存和cache划分为同样大小的“块”（连续的字），也称为Cache行

- 选择哪种映射方式，要考虑：
  - 硬件是否容易实现
  - 地址变换的速度是否快
  - 主存空间的利用率是否高
  - 主存装入一块时，发生冲突的概率
- 以下我们介绍三种映射方法

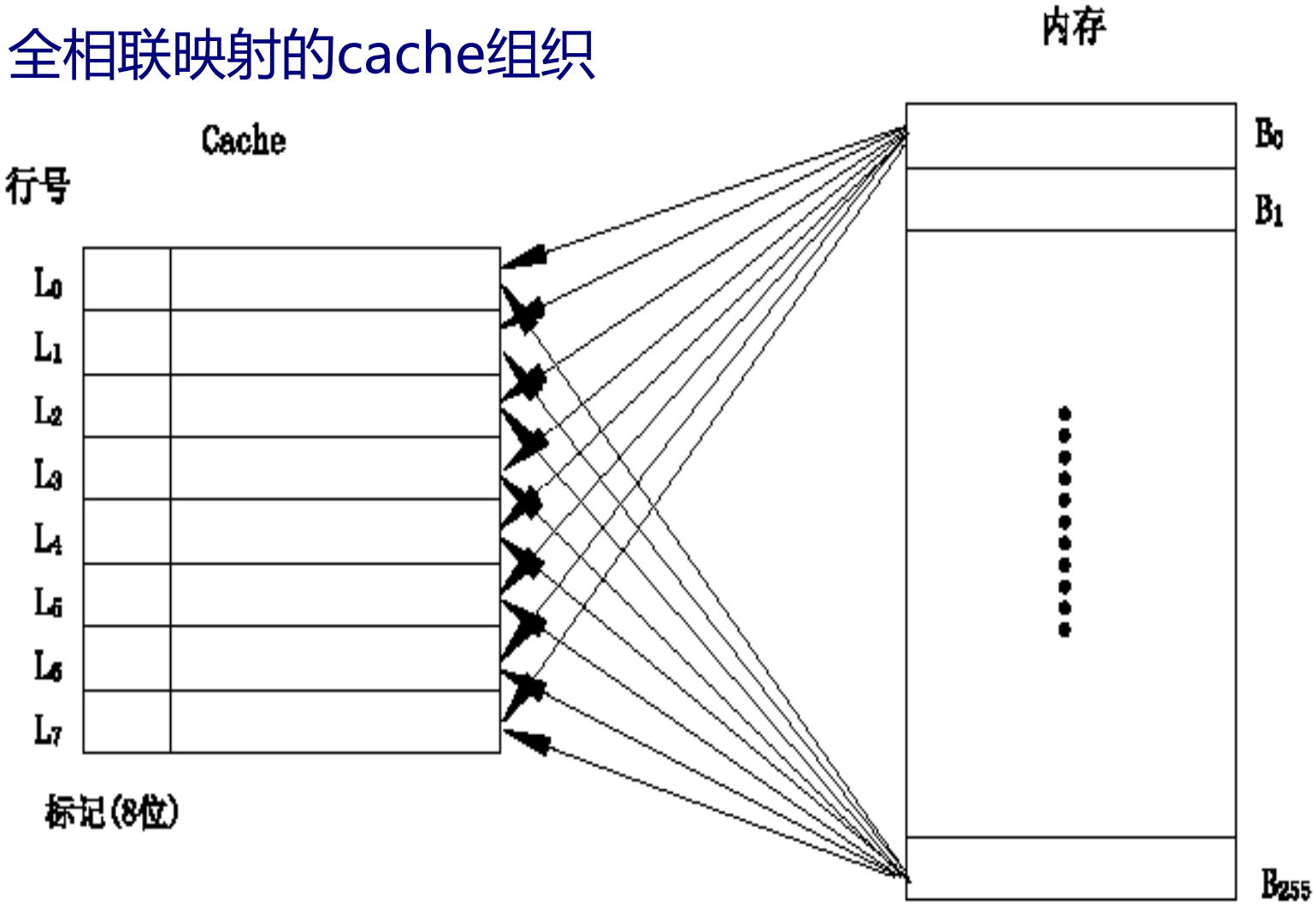
3.6.2 主存与Cache的地址映射

1、全相联的映射方式

- 映射方法（多对多）
  - 将主存中的一个块的地址（块号）与块的内容一起存于Cache的行中，其中块地址存于Cache行的标记部分中
  - 主存内容可以拷贝到任意行
- 地址变换
  - 标记实际上构成了一个目录表

3.6.2 主存与Cache的地址映射

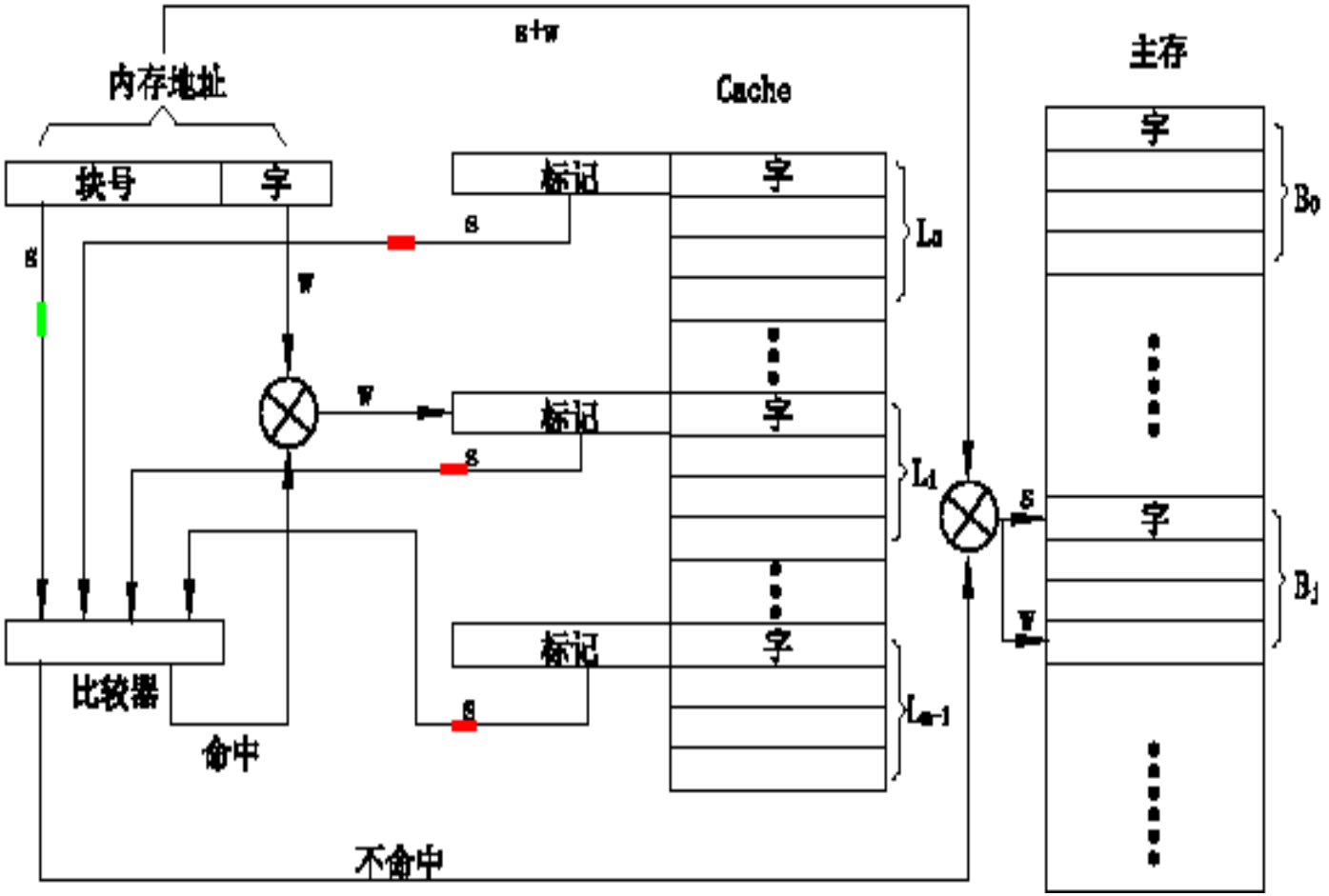
- 全相联映射的cache组织



示意图中Cache位8行，主存为256块.

3.6.2 主存与Cache的地址映射

- CPU给出访问地址后，将地址分为两部分（块号和字），比较电路将块号与Cache 表中的标记进行比较，相同表示命中，访问相应单元；如果没有命中，CPU 直接访问内存，并将被访问内存的相对应块写入Cache



3.6.2 主存与Cache的地址映射

- 全部标记用一个相联存储器实现，全部数据用一个普通RAM实现

- 特点
  - 优点：冲突概率小，Cache的利用高。
  - 缺点：比较器难实现，需要一个访问速度很快代价高的相联存储器

- 应用场合
  - 适用于小容量的Cache

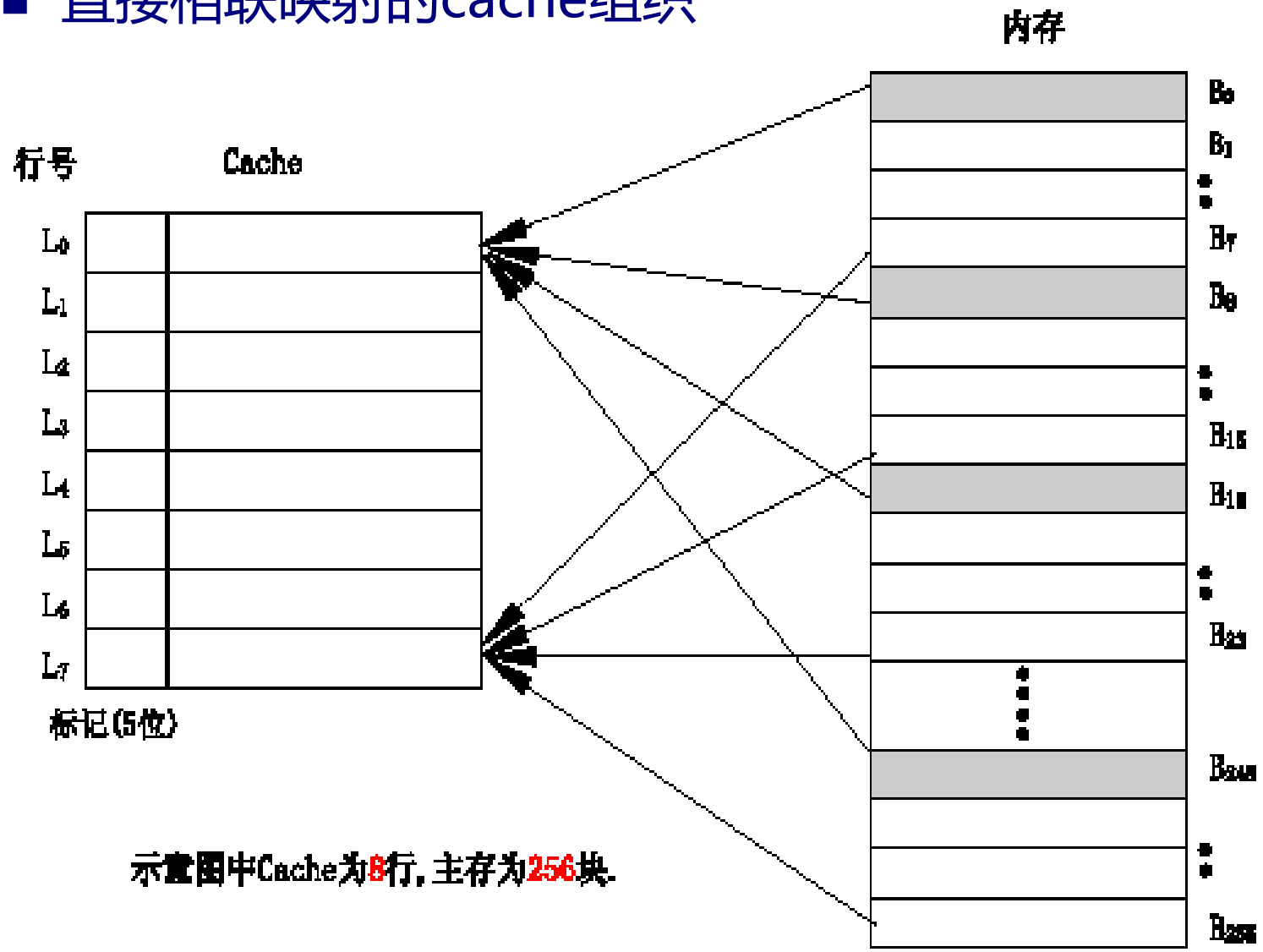
3.6.2 主存与Cache的地址映射

2、直接映射方式

- 映射方法（多对一）如:
  - i= j mod m
  - 主存第j块内容拷贝到Cache的i行
  - 一般i和m都是2N级
- [例] Cache容量16字，主存容量256字，则地址2，18，34.....242等都存放在cache的地址2内，如果第一次2在cache中，下次访问34内容，则不管cache其他位置的内容访问情况，都会引起2块内容的替换

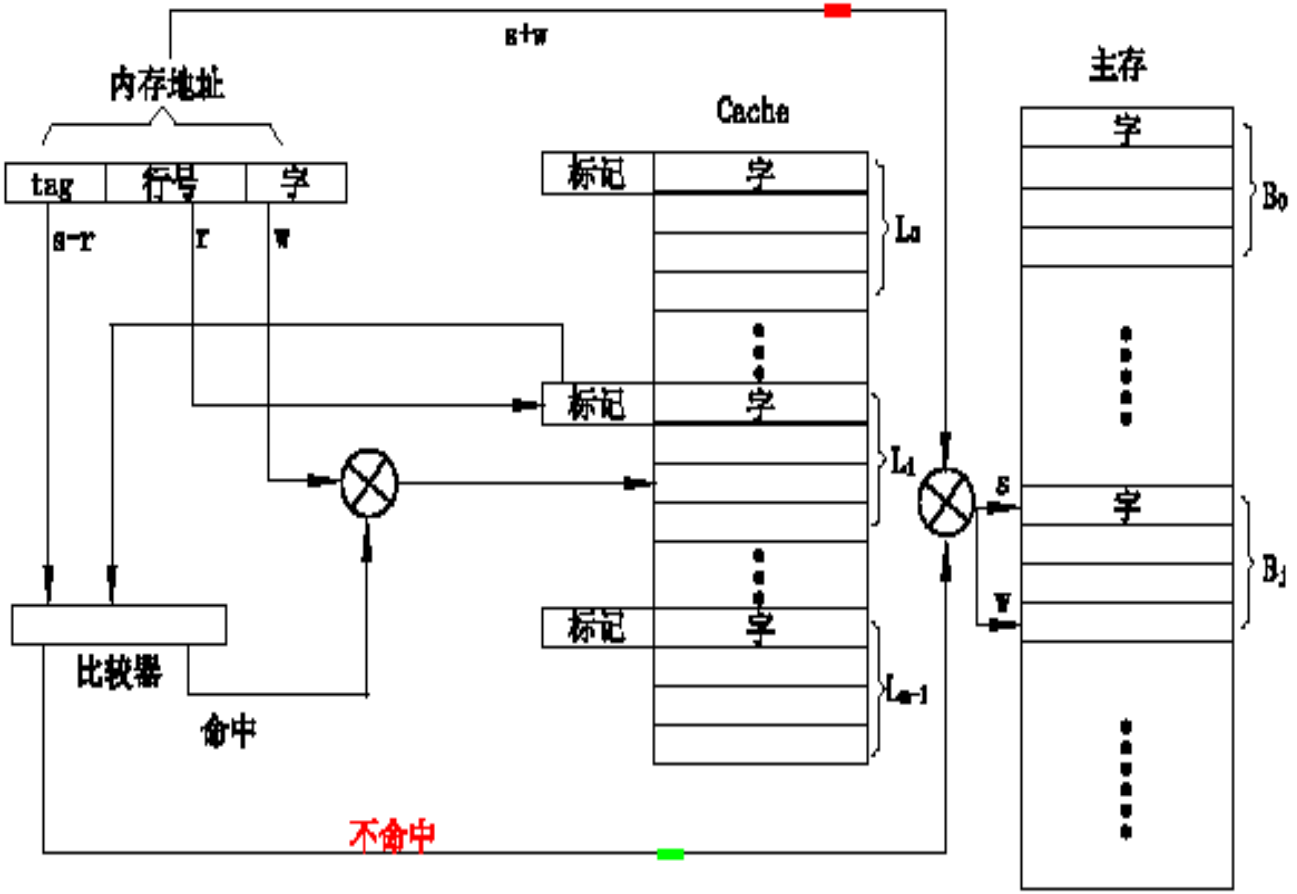
3.6.2 主存与Cache的地址映射

- 直接相联映射的cache组织



3.6.2 主存与Cache的地址映射

- 利用行号选择相应行；把行标记与CPU访问地址进行比较，相同表示命中，访问Cache；如果没有命中，访问内存，并将相应块写入Cache



3.6.2 主存与Cache的地址映射

- 特点
  - 优点：比较电路少m倍线路，所以硬件实现简单，Cache地址为主存地址的低几位，不需变换。
  - 缺点：冲突概率高（抖动）
- 应用场合
  - 适合大容量Cache

3.6.2 主存与Cache的地址映射

3、组相联映射方式

- 前两者的组合
- Cache分组，组间采用直接映射方式，组内采用全相联的映射方式
  - Cache分组 （多对多）
  - Cache 总行数 m=u X v
  - 组号 q= j mod u
  - 主存第j块内容拷贝到Cache的q组中的某行
  - 低序的位用于表示Cache的组号，高序的位作为标记与块数据一起存于此组的某一行中

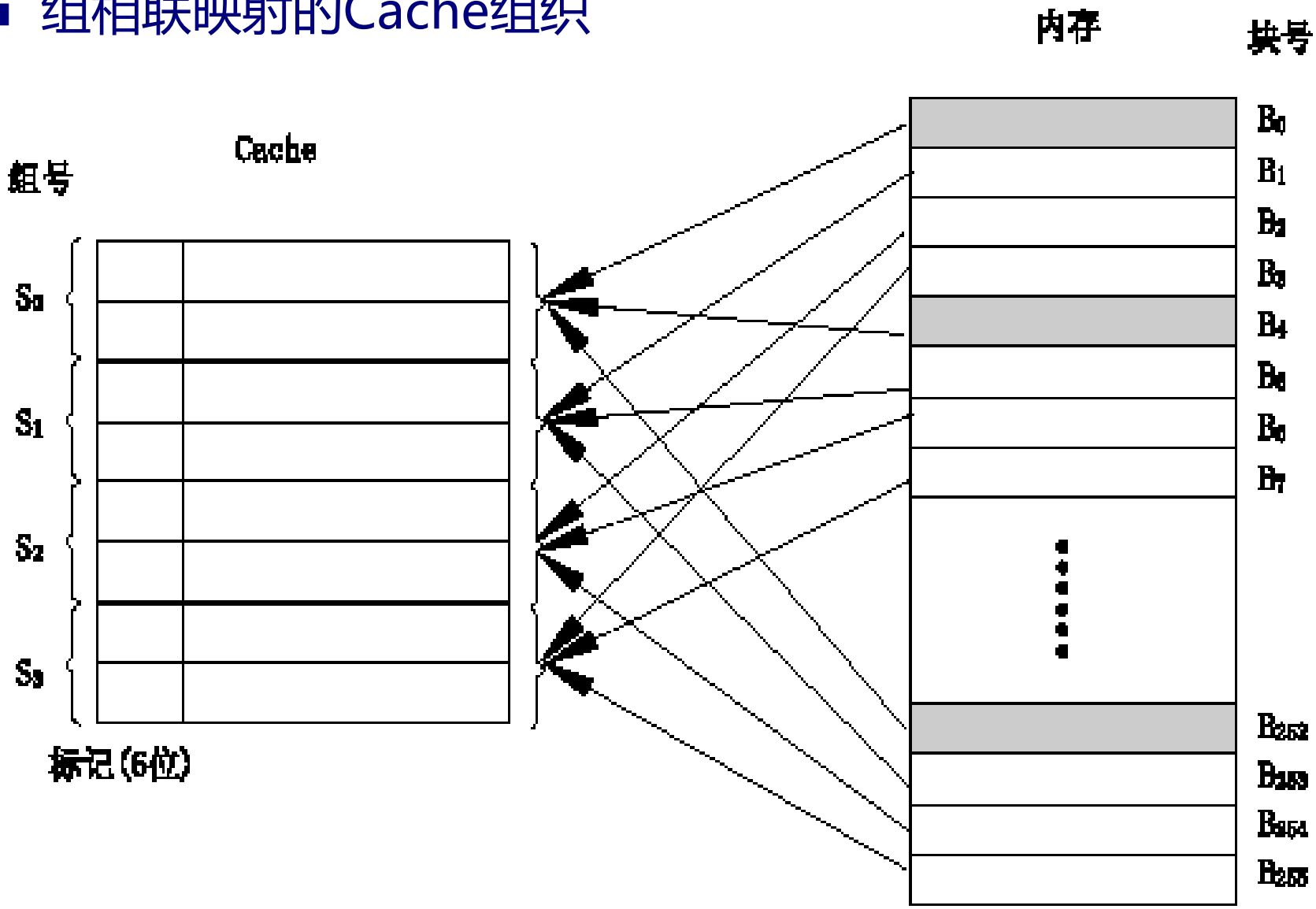
3.6.2 主存与Cache的地址映射

- 地址变换
  - 设主存地址x，看是不是在cache中，先y= x mod u，则在y组中一次查找

- 分析：比全相联容易实现，冲突低
  - v=1，则为直接相联映射方式
  - u=1，则为全相联映射方式
  - v的取值一般比较小， 一般是2的幂，称之为v路组相联Cache

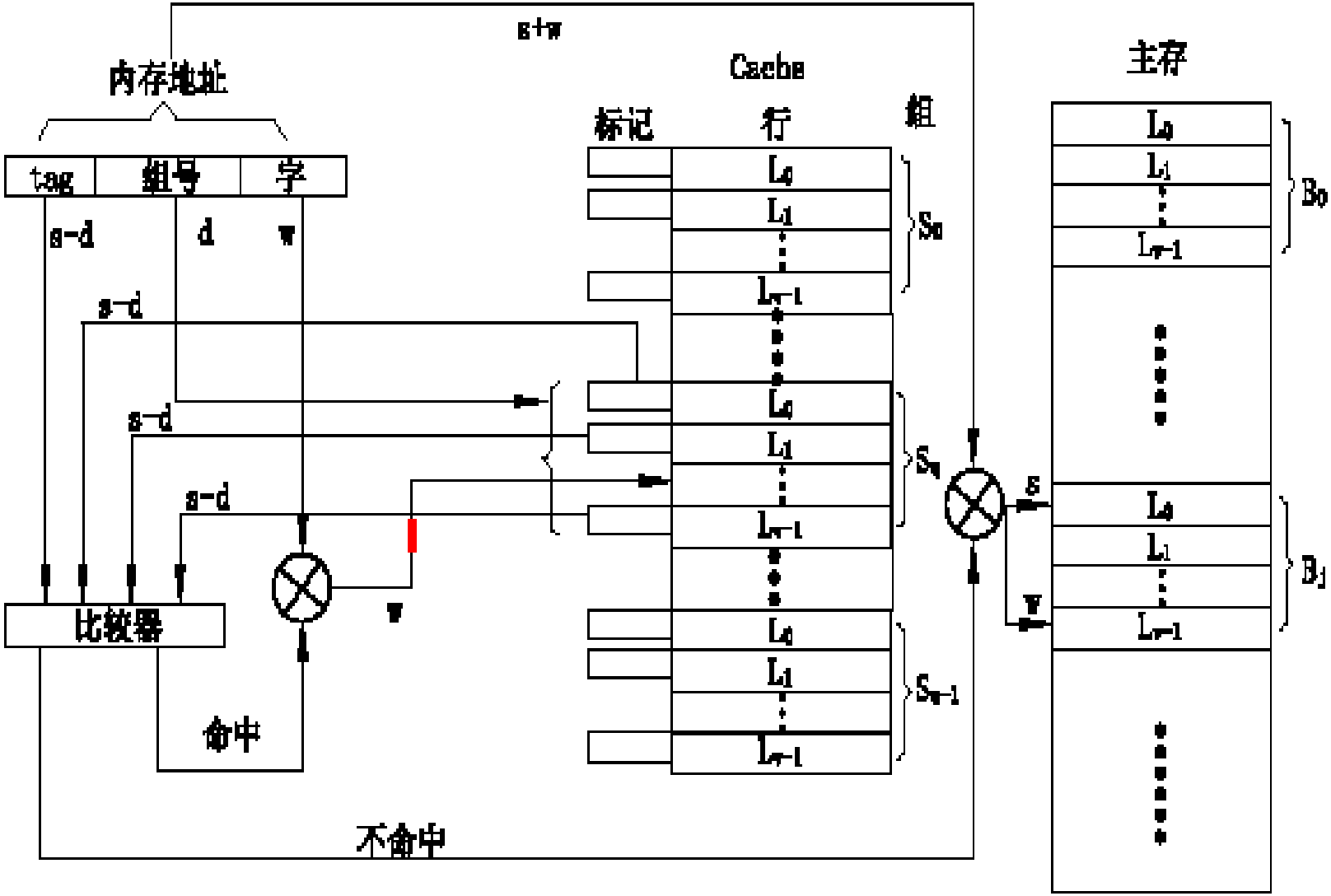
3.6.2 主存与Cache的地址映射

- 组相联映射的Cache组织



3.6.2 主存与Cache的地址映射

- 组相联映射的Cache的检索过程



3.6.2 主存与Cache的地址映射

- 有一个处理器，主存容量是1MB，字长1B，块大小16B，cache容量64KB。若采用全相联映射，对内存地址$(B0010)_{16}$给出相应的标记和字地址。
- 解：块大小=行大小=2^4字节=2^w字节，所以w=4位

  主存寻址单元数=2^{s+w}=1M=2^{20}，所以s+w=20，s=16位

  主存的块书=2^s=2^{16}

  标记大小=s=16位

  内存地址如下所示：

标记s	字地址w
16位	4位

故内存地址$(B0010)_{16}$=$(1011\ 0000\ 0000\ 0001\ 0000)_2$

故对应的标记s=$(1011\ 0000\ 0000\ 0001)_2$，字地址w=$(0000)_2$

3.6.2 主存与Cache的地址映射

- 直接映射方式的内存地址格式如下所示：

标记s-r	行r	字w
8位	14位	2位

- 若主存地址用十六进制表示为BBBBBB，请用十六进制格式表示直接映射方法Cache的标记、行、字地址的值。
- 解：$(BBBBBB)_{16}=(1011\ 1011\ 1011\ 1011\ 1011\ 1011)_2$

  标记s-r=$(1011\ 1011)_2=(BB)_{16}$

  行r=$(1011\ 1011\ 1011)_2=(2EEE)_{16}$

  字地址w=$(11)_2=(3)_{16}$

3.6.2 主存与Cache的地址映射

- 一个组相联cache由64个行组成，每组4行。主存储器包含4K个块，每个块128字。请表示内存地址的格式。
- 解：块大小=行大小=2^w个字，$2^w=128=2^7$，所以w=7

  每组的行数k=4

  cache的行数 = kv = $2^d \times k = 2^d \times 4 = 64$，所以d=4

  组数v=$2^d=2^4=16$

  主存的块数=$2^s=4K=2^2\times2^{10}=2^{12}$，所以s=12

  标记大小=s-d=12-4=8（位）

  主存地址长度=s+w=12+7=19（位）

  主存寻址单元数=$2^{s+w}=2^{19}$

  ∴v=4路组相联的内存地址格式如下所示

8位	4位	7位
标记s-d	cache组号d	块内地址w

3.6.3 替换策略

- 当一个新的主存块需要拷贝到Cache，而允许存放此块的行位置都被其它主存块占满时，就要产生替换

- 对直接Cache，由于映射位置是固定的，直接替换即可
- 对于组相联和全相联Cache，则需要替换策略

3.6.3 替换策略

- LFU（最不经常使用，频率）
  - 被访问的行计数器增加1，换值小的行，不能反映近期cache的访问情况，
- LRU（近期最少使用，时间）
  - 被访问的行计数器置0，其他的计数器增加1，换值大的行，符合cache的工作原理(更符合局部性原理)
- 随机替换
  - 随机替换策略实际上是不要什么算法，从特定的行位置中随机地选取一行换出即可。这种策略在硬件上容易实现，且速度也比前两种策略快。缺点是随意换出的数据很可能马上又要使用，从而降低命中率和cache工作效率。但这个不足随着cache容量增大而减小。随机替换策略的功效只是稍逊于前两种策略。

3.6.3 替换策略

- 例：设cache有1、2、3、4共4个块，a、b、c、d等为主存中的块,访问顺序一次如下：a、b、c、d、b、b、c、c、d、d、a ,下次若要再访问e块。
  问，采用LFU和LRU算法替换结果是不是相同?

3.6.3 替换策略

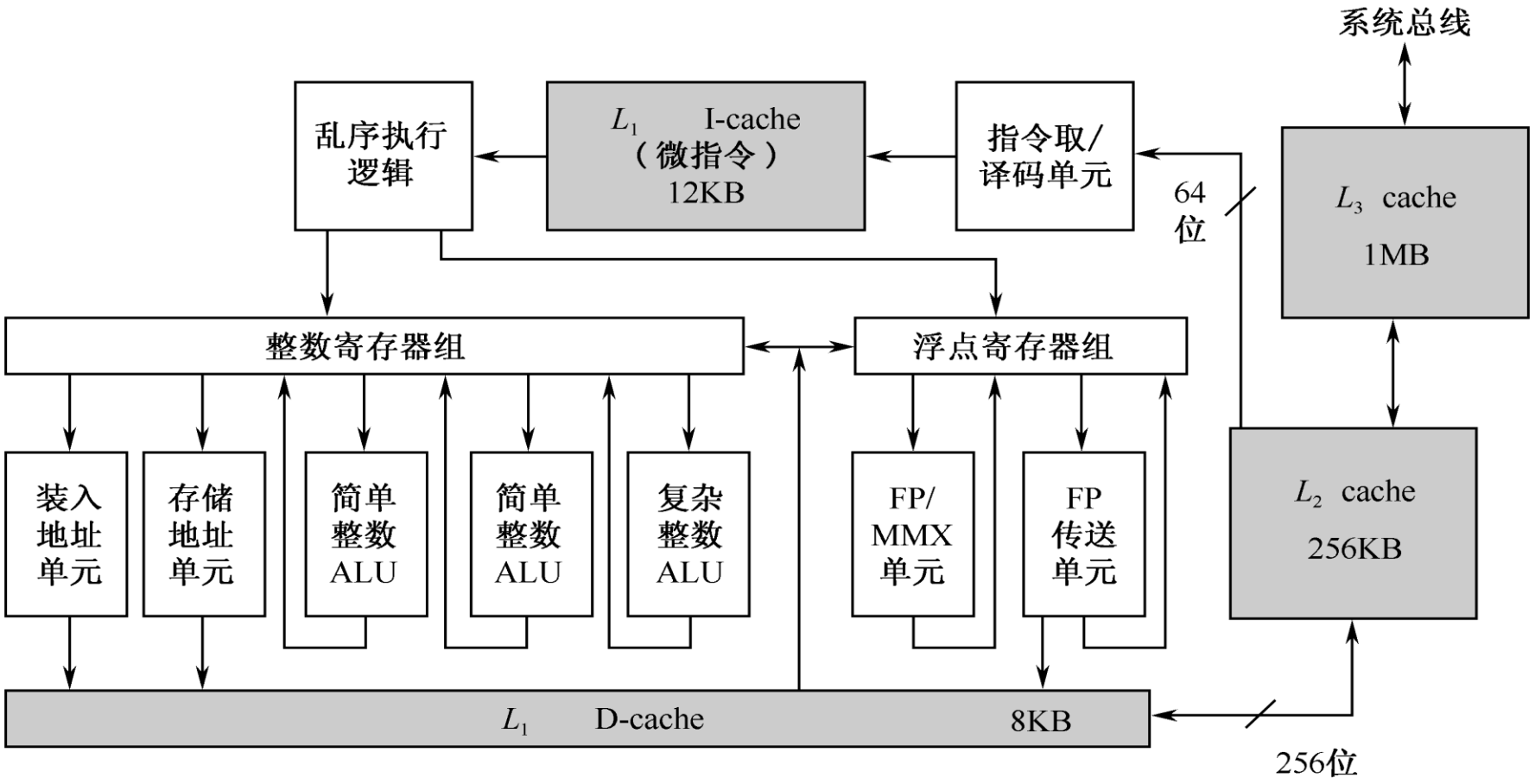
	LFU（最不经常使用）					LRU（近期最少使用）				
	说明	1块	2块	3块	4块	说明	1块	2块	3块	4块
a	a进入	1	0	0	0	a进入	0	1	1	1
b	b进入	1	1	0	0	b进入	1	0	2	2
c	c进入	1	1	1	0	c进入	2	1	0	3
d	d进入	1	1	1	1	d进入	3	2	1	0
b	命中	1	2	1	1	命中	4	0	2	1
b	命中	1	3	1	1	命中	5	0	3	2
c	命中	1	3	2	1	命中	6	1	0	3
c	命中	1	3	3	1	命中	7	2	0	4
d	命中	1	3	3	2	命中	8	3	1	0
d	命中	1	3	3	3	命中	9	4	2	0
a	命中	2	3	3	3	命中	0	5	3	1
e	替换a	1	0	0	0	替换b	1	0	4	2

3.6.4 写操作策略

- 由于cache的内容只是主存部分内容的拷贝，它应当与主存内容保持一致。而CPU对cache的写入更改了cache的内容。如何与主存内容保持一致，可选用如下三种写操作策略
  - 写回法：写命中时，只修改Cache的内容；需要增加修改位；换出时，对行的修改位进行判断，决定是写回还是舍掉
  - 全写法（写直达、write-through）：写命中时，Cache与内存一起写
  - 写一次法：与写回法一致，但是第一次Cache命中时采用全写法。

3.6.5 Pentium PC的Cache

- 基本原理



3.6.5 Pentium PC的Cache

- 3级cache结构
  - L3内容是主存的子集
  - L2内容是L3的子集
  - L1内容是L2的子集
- L1分成12K的指令cache和8K的数据cache
  - 指令cache是单端口256位，只读
  - 数据cache是双端口（每个32位），读写，采用2路组相联结构128组*2行/组*32字节/行=8KB字节

3.6.5 Pentium PC的Cache

- 存储器读写总线周期
  - 256位淬发式传送（只需给出块的起始地址，然后对固定块长度的数据一个接一个地读出或写入）
  - 64位传送
- 数据一致性的保持
  - L1采用写一次法
  - L2采用写回法

本章小结

- 对存储器的要求是容量大、速度快、成本低。为了解决了这三方面的矛盾，计算机采用多级存储体系结构，即cache、主存和外存。CPU能直接访问内存(cache、主存），但不能直接访问外存。存储器的技术指标有存储容量、存取时间、存储周期、存储器带宽。

- 广泛使用的SRAM和DRAM都是半导体随机读写存储器，前者速度比后者快，但集成度不如后者高。二者的优点是体积小，可靠性高，价格低廉，缺点是断电后不能保存信息。

本章小结

- 只读存储器和闪速存储器正好弥补了SRAM和DRAM的缺点，即使断电也仍然保存原先写入的数据。特别是闪速存储器能提供高性能、低功耗、高可靠性以及移动性，是一种全新的存储器体系结构

- 双端口存储器和多模块交叉存储器属于并行存储器结构。前者采用空间并行技术，后者采用时间并行技术。这两种类型的存储器在科研和工程中大量使用

本章小结

- cache是一种高速缓冲存储器，是为了解决CPU和主存之间速度不匹配而采用的一项重要的硬件技术，并且发展为多级cache体系，指令cache与数据cache分设体系。要求cache的命中率接近于1。主存与cache的地址映射有全相联、直接、组相联三种方式。其中组相联方式是前二者的折衷方案，适度地兼顾了二者的优点又尽量避免其缺点，从灵活性、命中率、硬件投资来说较为理想，因而得到了普遍采用。

作业

- 1; 2; 4; 6; 8; 9; 10

作业

- Cache命中率和哪些因素有关？如果CPU执行一段程序，访问 Cache 3800次(即N_c)，访问主存200次(即N_m)，Cache的存取周期T_c = 50ns，主存存取周期T_m = 250ns。求Cache命中率，平均访存时间及Cache—主存系统效率。
- 设主存容量1MB，有16KB直接相联映像的Cache，假定该Cache的块为8个32位的字。解答下列问题：（1）写出Cache的地址格式。（2）写出主存的地址格式。（3）块表的容量有多大？（4）画出直接方式地址映像及变换示意图；（5）主存地址为DE8F8H的单元在Cache中的什么位置？