



西安电子科技大学
XIDIAN UNIVERSITY

第四章 存储系统

主讲：张骏鹏（博士，副教授）

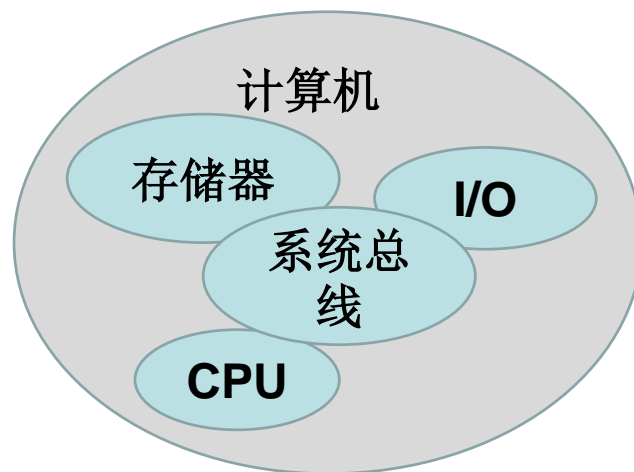
西安电子科技大学

人工智能学院



第四章 存储系统

- 存储系统概述
 - 存储器的分类
 - 存储器的层次结构
- 主存储器
- 高速缓冲存储器
- 辅助存储器





第四章 存储系统

- 存储系统概述
- 主存储器
 - 主存储器的逻辑结构
 - 主存储器的基本组成
 - 半导体存储芯片（基本结构和译码驱动方式）
 - **随机存取存储器与只读存储器**
 - 主存储器的设计方法
 - 提高访存速度的措施
- 高速缓冲存储器
- 辅助存储器



2. 随机存取存储器——SRAM与DRAM的对比

SRAM与DRAM的对比

	<div>主存</div> DRAM	SRAM <div>缓存</div>
存储原理	电容	触发器
集成度	高	低
芯片引脚	少	多
功耗	小	大
价格	低	高
速度	慢	快
刷新	有	无



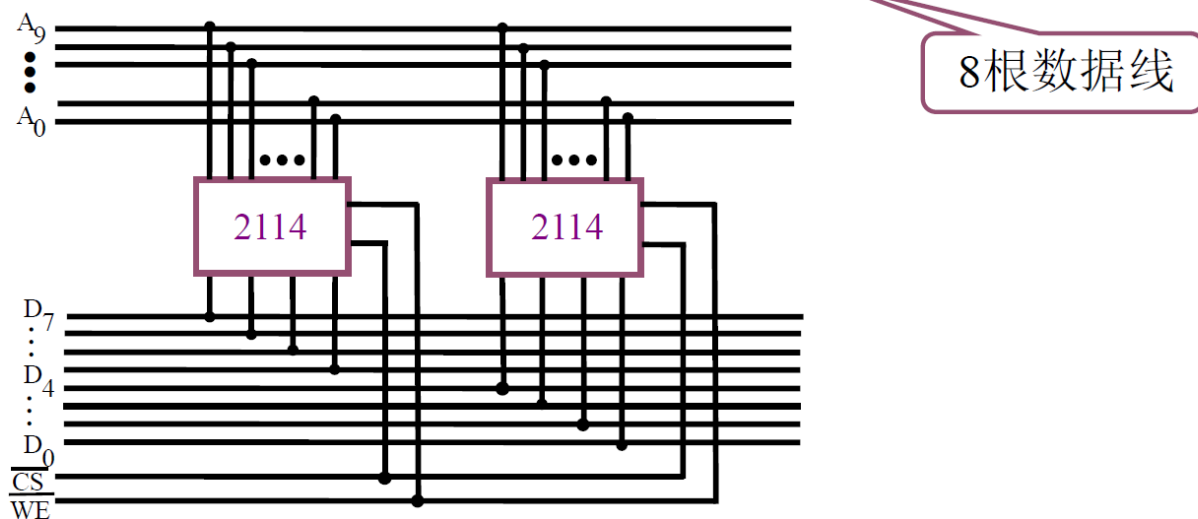
2. 随机存取存储器——主存储器设计方法

- 存储器容量的扩展;
- 存储器与CPU的连接;

1. 存储器容量的扩展

(1) 位扩展（增加存储字长）

用 片 $1K \times 4$ 位 存储芯片组成 $1K \times 8$ 位的存储器

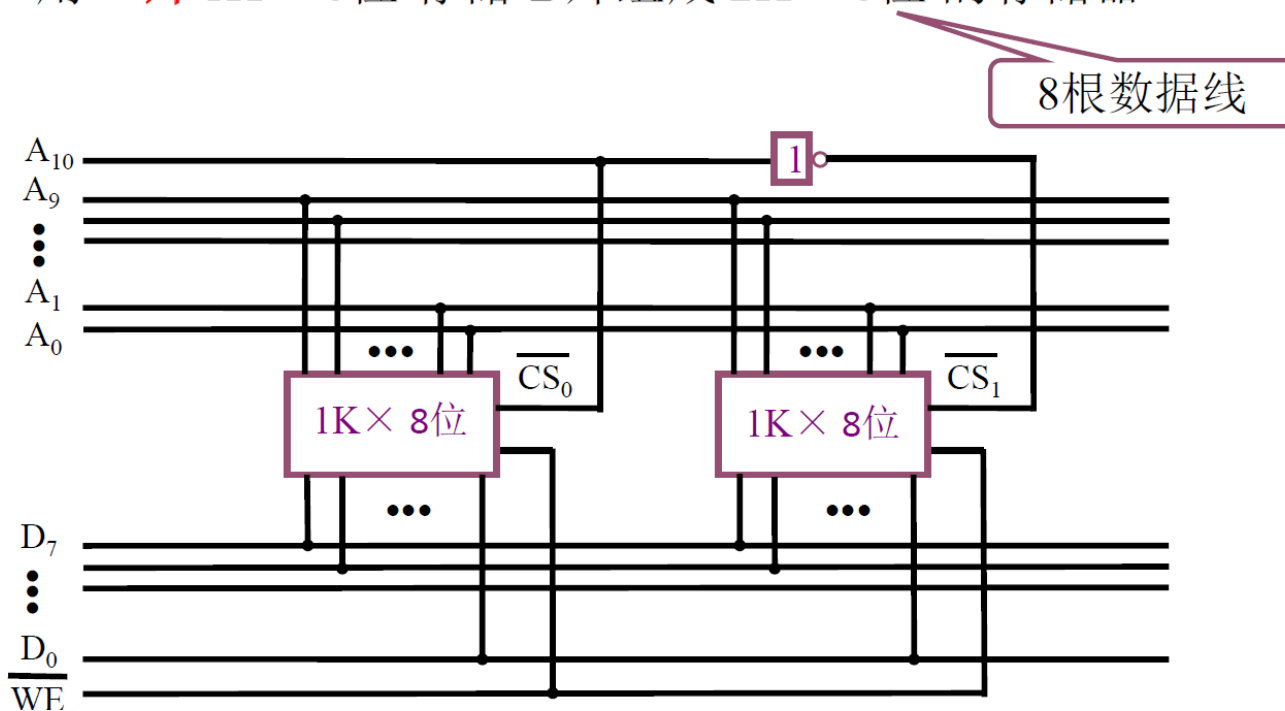




2. 随机存取存储器——主存储器设计方法

(2) 字扩展（增加存储字的数量）

用 片 $1\text{K} \times 8$ 位 存储芯片组成 $2\text{K} \times 8$ 位的存储器





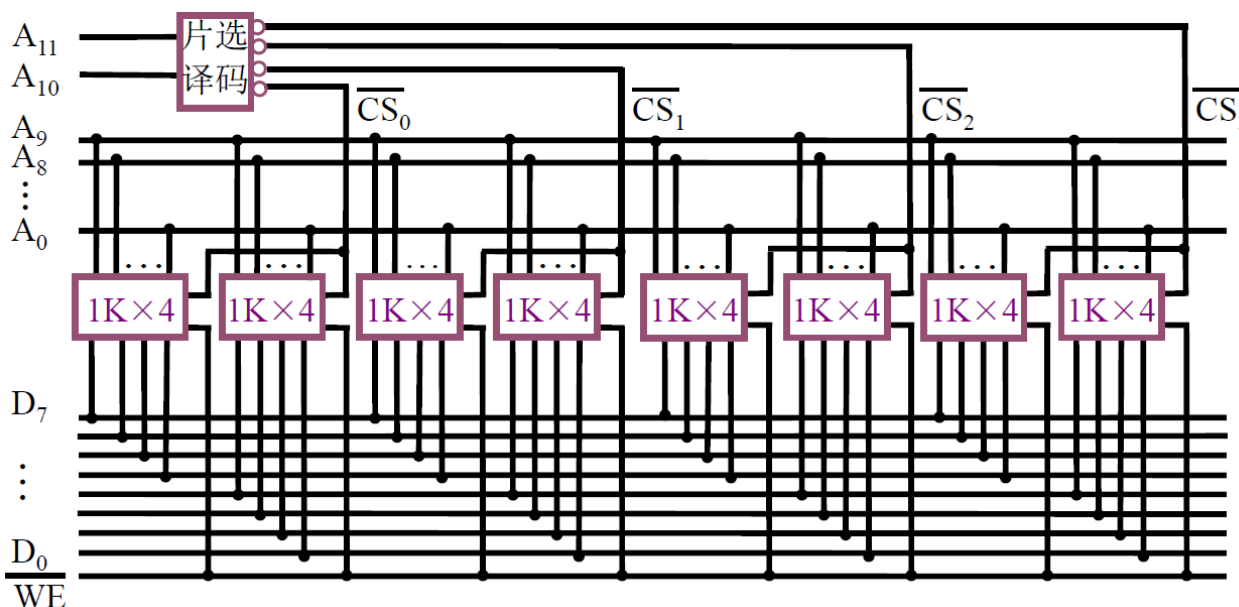
2. 随机存取存储器——主存储器设计方法

(3) 字、位扩展

用 8 片 $1\text{K} \times 4$ 位 存储芯片组成 $4\text{K} \times 8$ 位的存储器

12根地址线

8根数据线





2. 随机存取存储器——主存储器设计方法

2. 存储器与 CPU 的连接

- (1) 地址线的连接
- (2) 数据线的连接
- (3) 读/写命令线的连接
- (4) 片选线的连接
- (5) 合理选择存储芯片
- (6) 其他 时序、负载



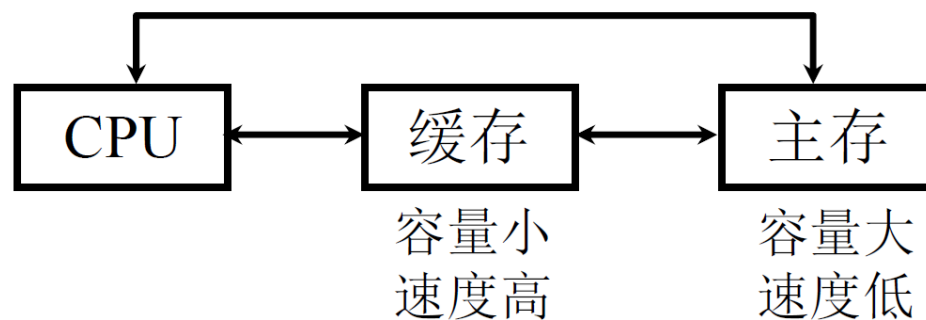
第四章 存储系统

- 存储系统概述
- 主存储器
- 高速缓冲存储器
 - 程序访问的局限性原理
 - Cache的工作原理与基本结构
 - Cache-主存地址映射
 - 替换算法
- 辅助存储器



3. 高速缓冲存储器

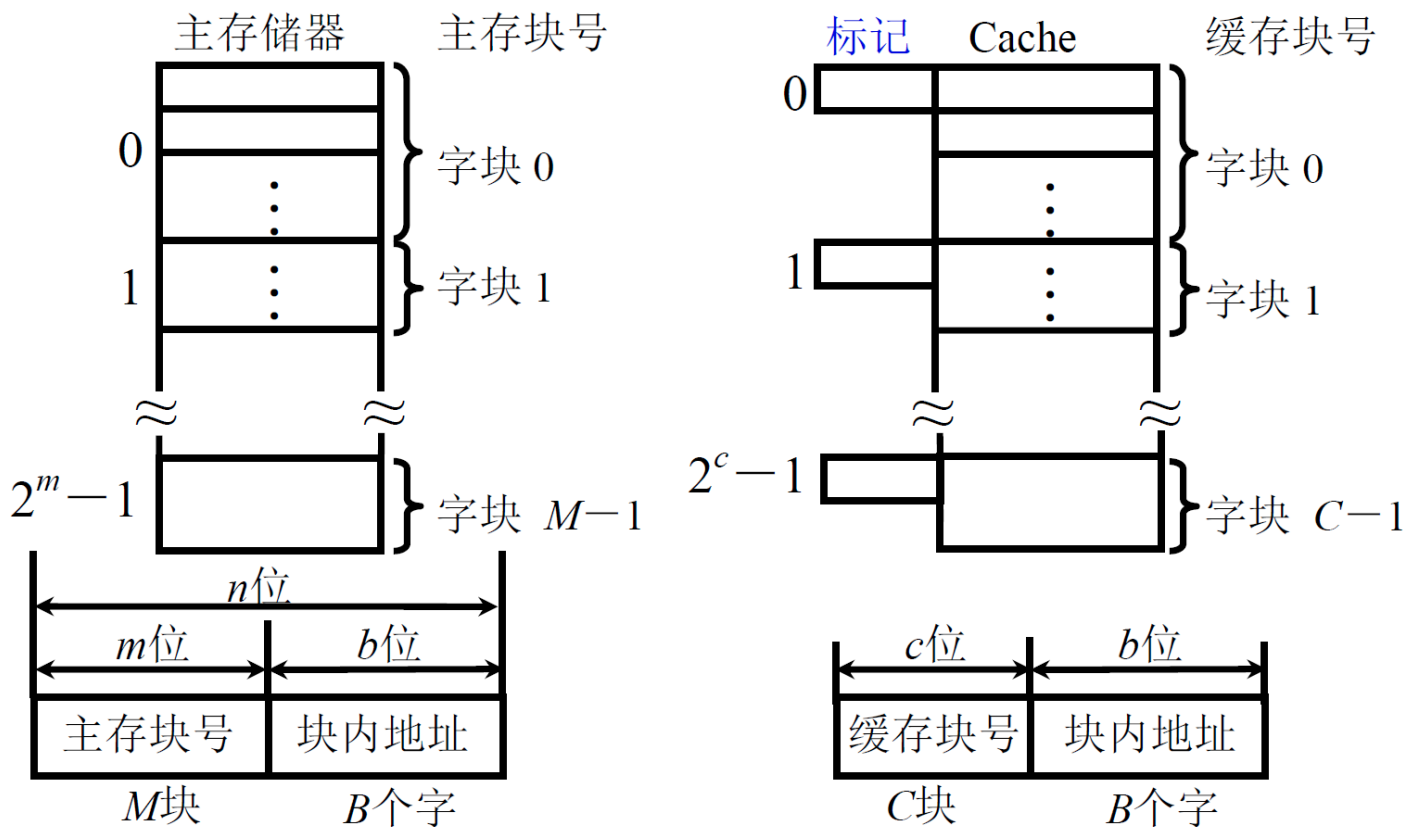
对大量程序执行情况的分析表明:在一段较短的时间间隔内, 程序集中在某一较小的主存地址空间上执行, 这就是**程序执行的局部性原理**。同样, 对数据的访问也存在局部性现象。



高速缓冲存储器的工作建立在程序及数据访问的局部性原理之上。



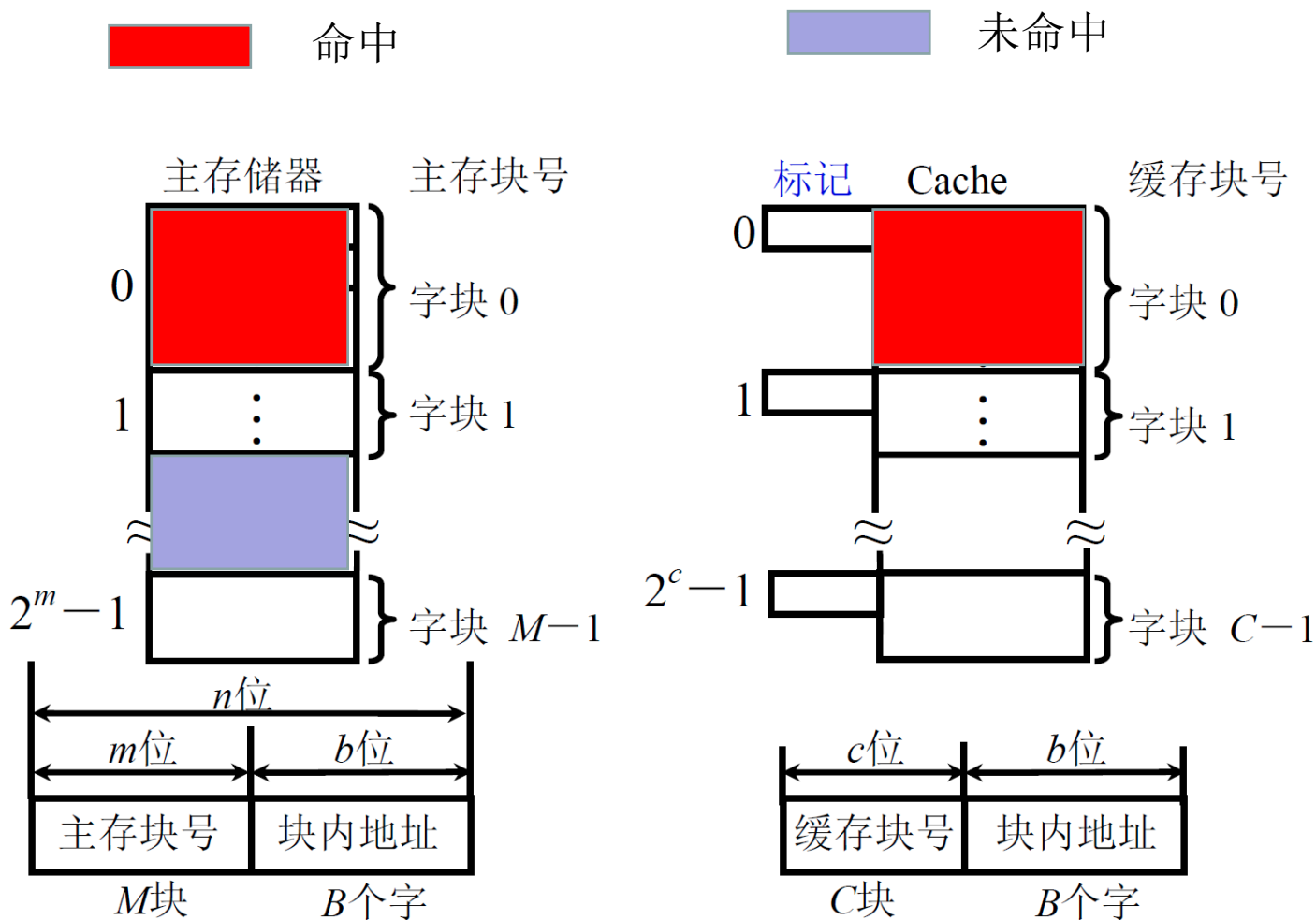
3. 高速缓冲存储器——Cache的工作原理与基本结构



高速缓冲存储器的工作建立在程序及数据访问的局部性原理之上。



3. 高速缓冲存储器——Cache的工作原理与基本结构

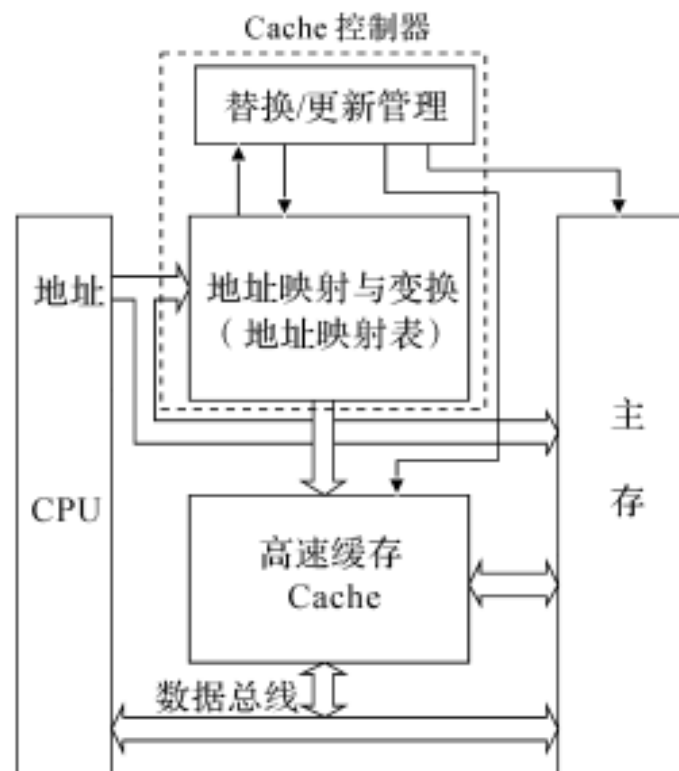




3. 高速缓冲存储器——Cache的工作原理与基本结构

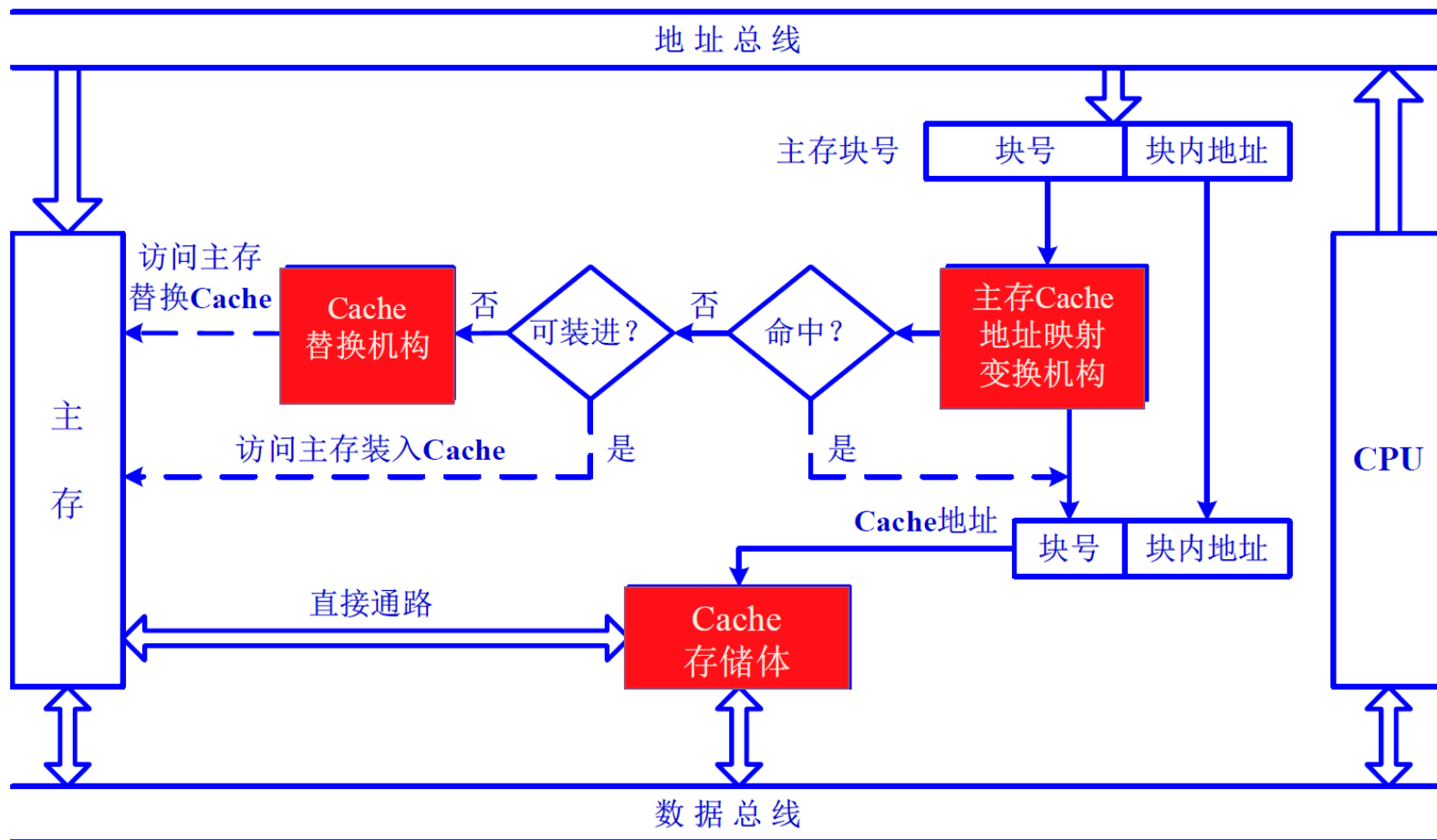
在Cache中面临三个关键问题：

- 如何判断是否命中；
- （若命中）Cache-主存地址映射与变换；
- Cache内容如何更新。





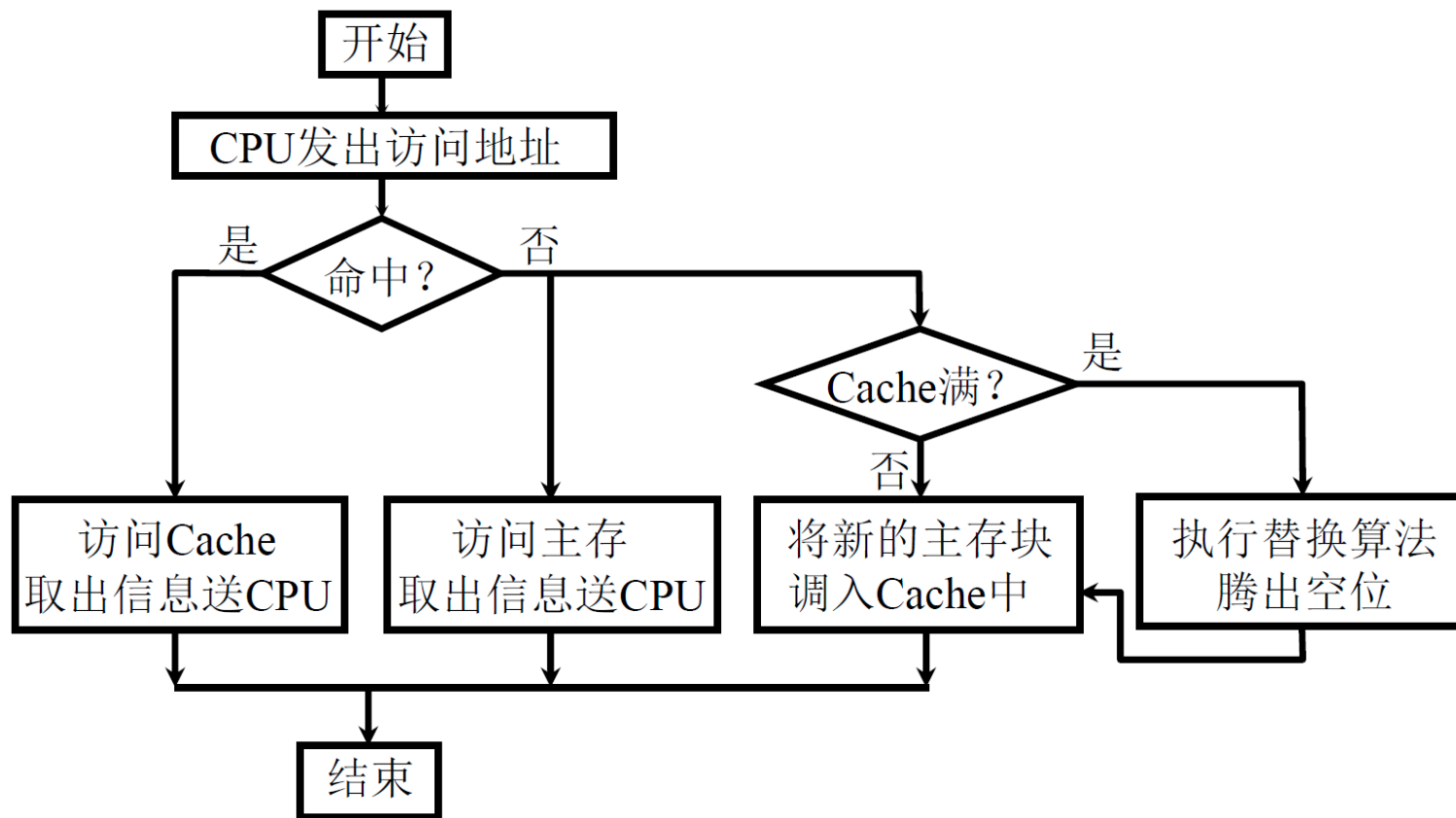
3. 高速缓冲存储器——Cache的工作原理与基本结构





3. 高速缓冲存储器——Cache的工作原理与基本结构

➤ Cache的读操作





3. 高速缓冲存储器——Cache的工作原理与基本结构

➤ Cache的写操作

Cache 和主存的一致性

- 写直达法 (Write – through)

写操作时数据既写入Cache又写入主存

写操作时间就是访问主存的时间，Cache块退出时，不需要对主存执行写操作，更新策略比较容易实现

- 写回法 (Write – back)

写操作时只把数据写入 Cache 而不写入主存

当 Cache 数据被替换出去时才写回主存

写操作时间就是访问 Cache 的时间，

Cache块退出时，被替换的块需写回主存，增加了 Cache的复杂性



3. 高速缓冲存储器——地址映射与变换

在 Cache 工作过程中，需要将主存的信息拷贝到 Cache 中，这就需建立 主存地址与 Cache 地址之间的映射关系，并将该关系存于 地址映射表 中，这就是主存地址到 Cache 地址的地址映射。

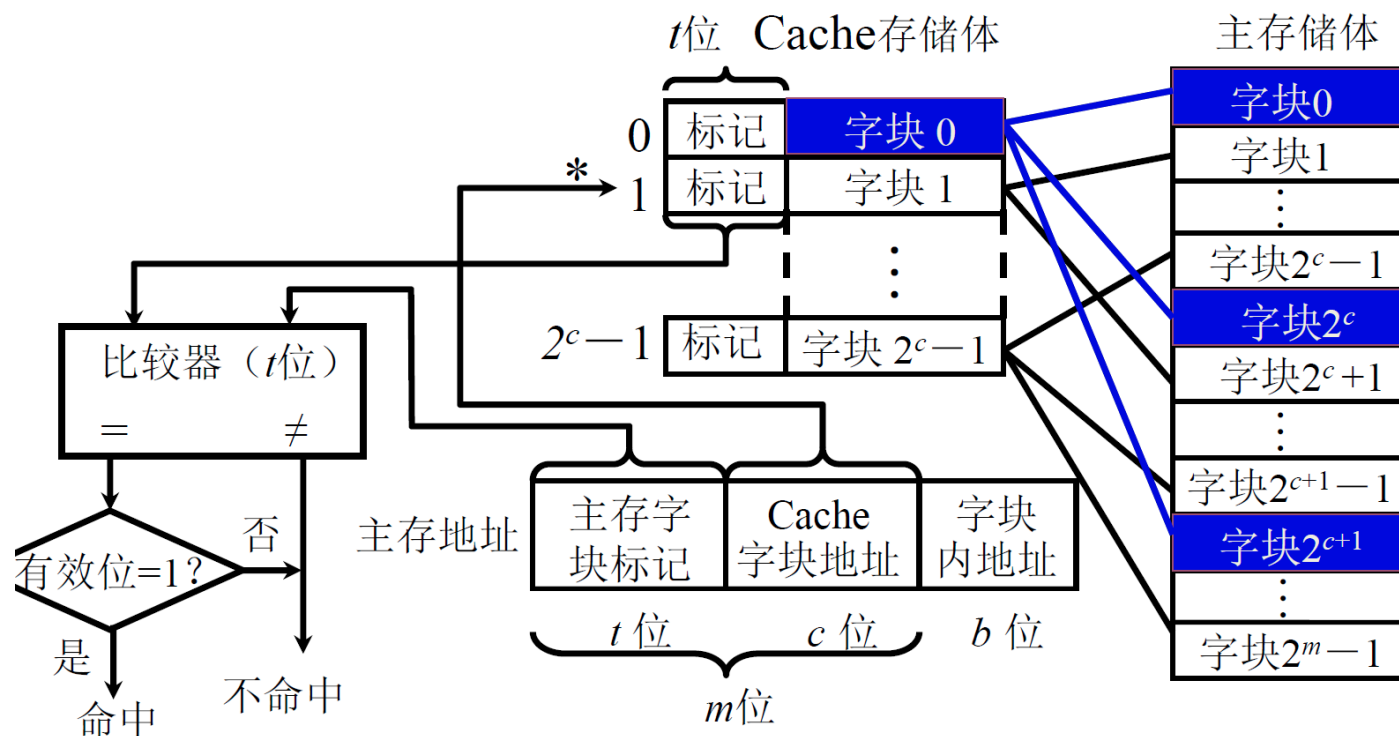
地址映射与变换有如下三种基本方式：

- 全相联地址映射方式
- 直接地址映射方式
- 组相联地址映射方式



3. 高速缓冲存储器——Cache的工作原理与基本结构

➤ 直接地址映射方式



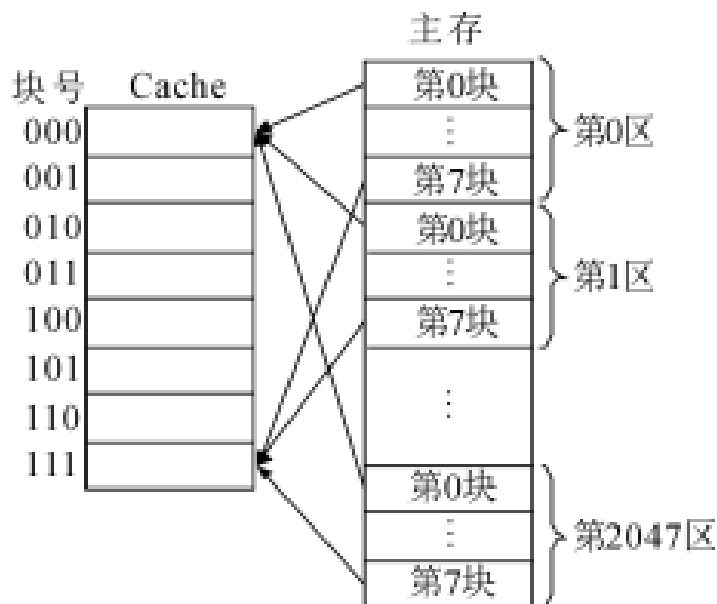
每个缓存块 i 可以和 若干个 主存块 对应
每个主存块 j 只能和 一个 缓存块 对应



3. 高速缓冲存储器——Cache的工作原理与基本结构

➤ 直接地址映射方式

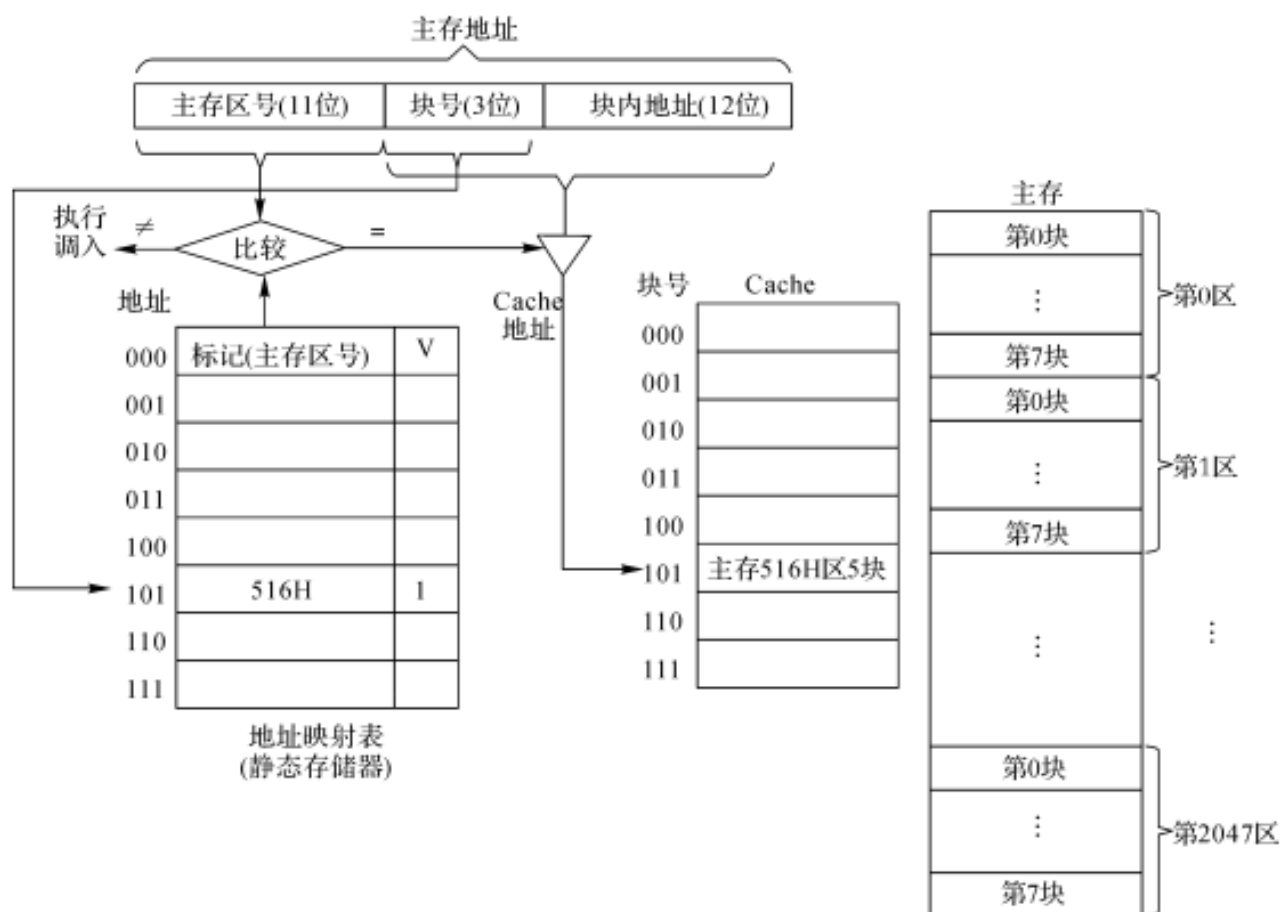
主存为64MB，Cache为32KB，要将主存先以Cache的容量(32KB)分区，则64MB的主存就被分成0~2047个区，即2K个区，而后Cache和主存的每个区再以4KB分块。





3. 高速缓冲存储器——Cache的工作原理与基本结构

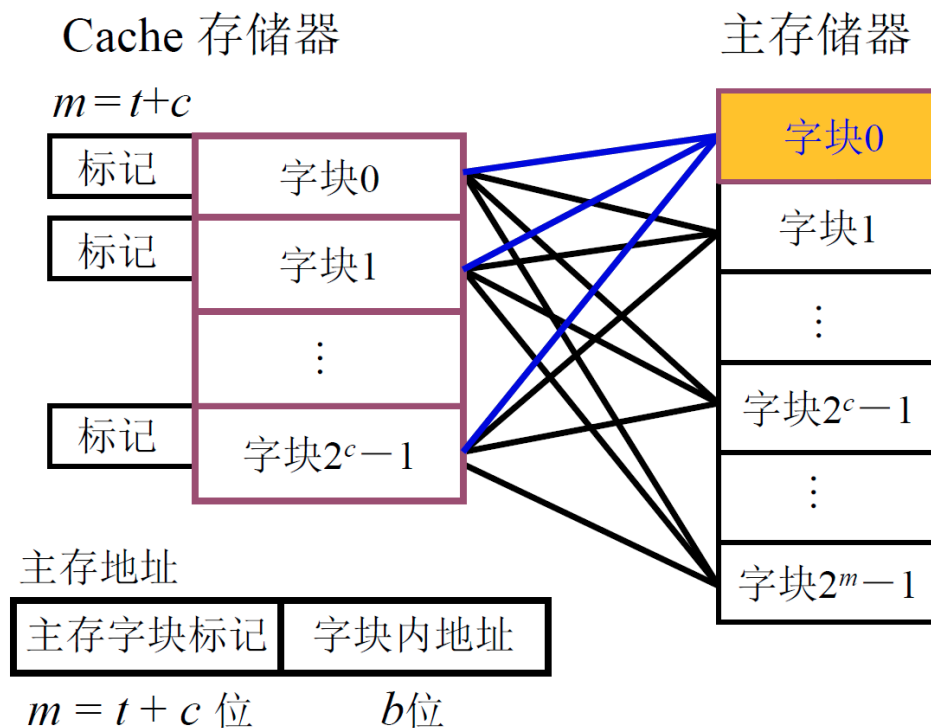
➤ 直接地址映射方式





3. 高速缓冲存储器——地址映射与变换

➤ 全相联地址映射方式



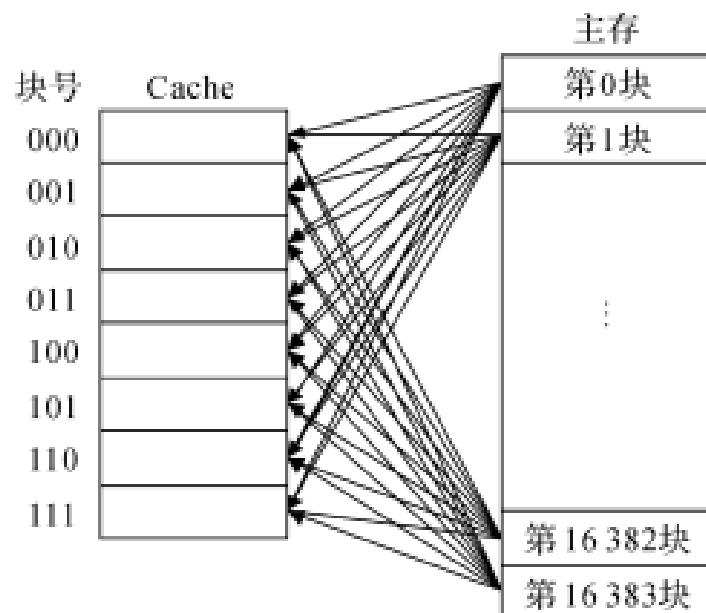
主存 中的 任一块 可以映射到 缓存 中的 任一块



3. 高速缓冲存储器——地址映射与变换

➤ 全相联地址映射方式

主存为64MB，Cache为32KB，若以4KB大小分块，Cache被分为块号为0~7的8个块，块号可用3位二进制编码表示；而主存分为16K 个块，块号为0~16383，块号需要用14位二进制编码表示。

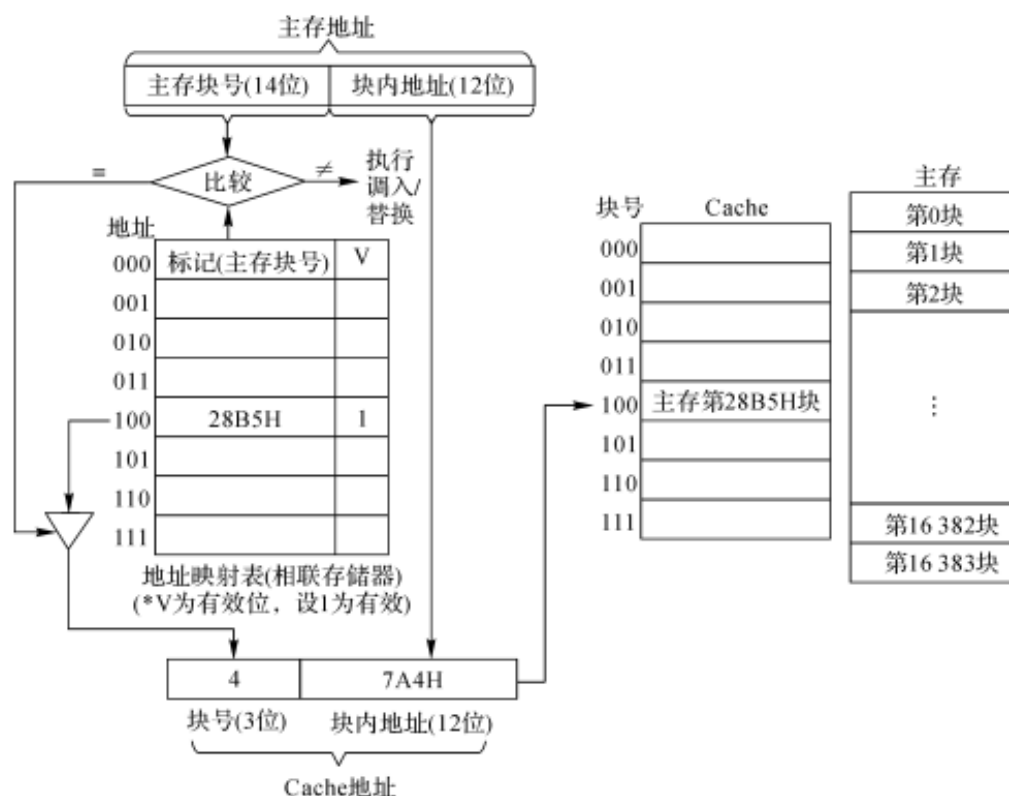




3. 高速缓冲存储器——地址映射与变换

➤ 全相联地址映射方式与实现

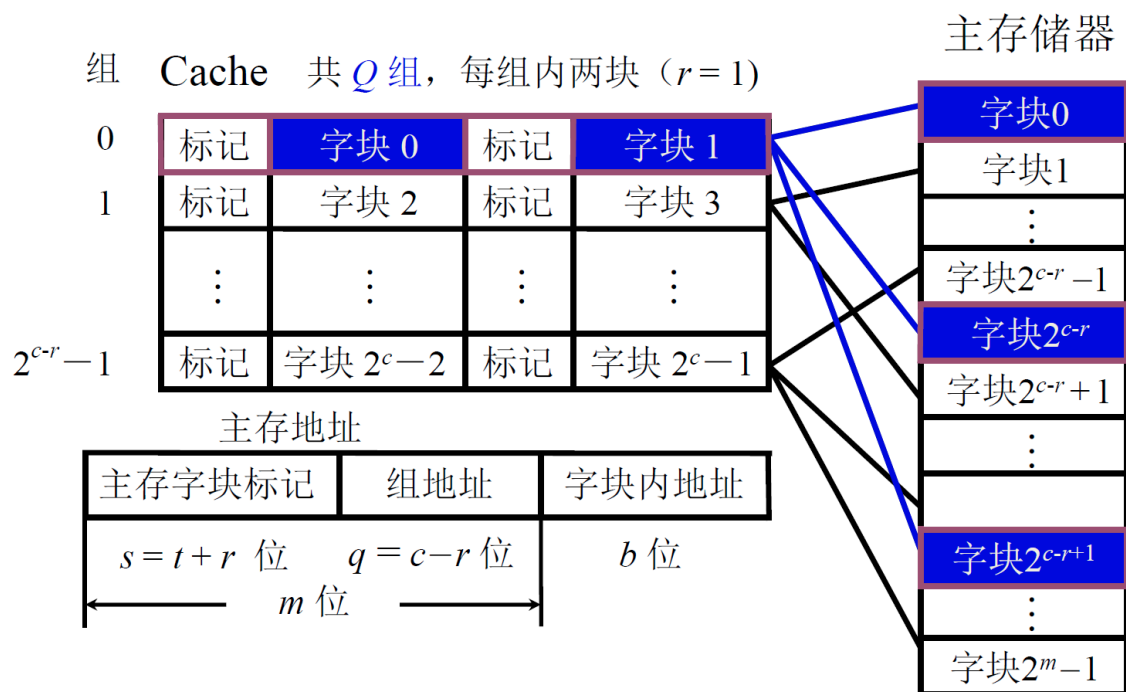
假定某一时刻 CPU 欲访问的主存地址为 28B57A4H





3. 高速缓冲存储器——地址映射与变换

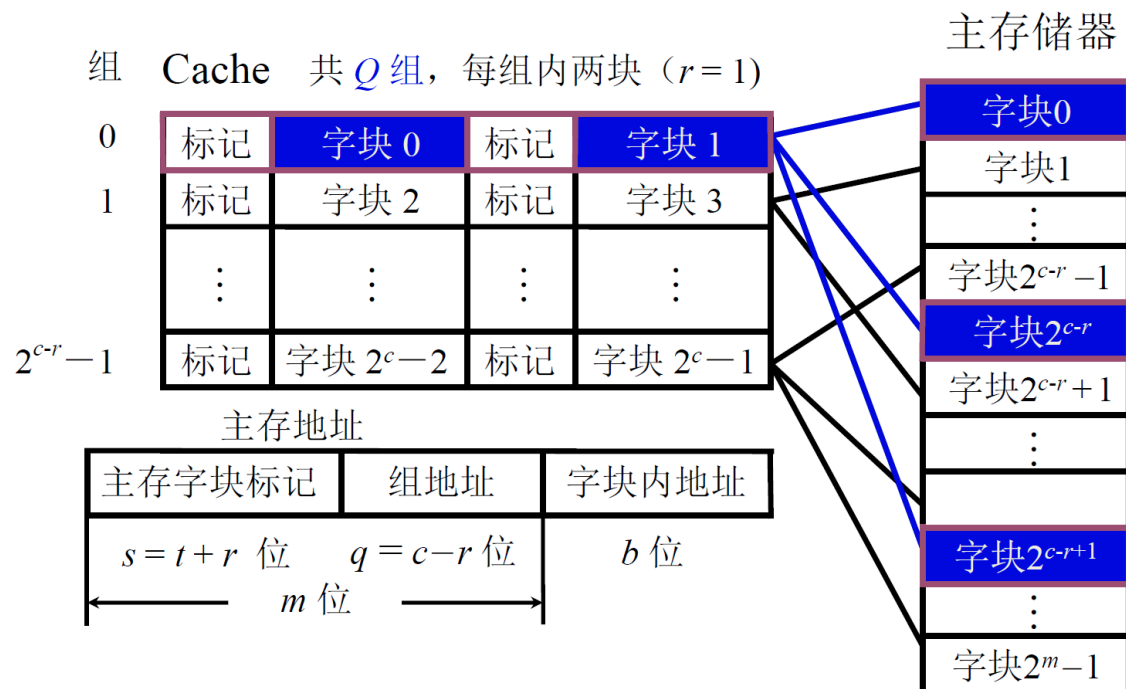
➤ 组相联地址映射方式



- 将 Cache 先分组，组内再分块。
- 主存结构是先分区，区内按 Cache 的方法分组，组内再分块。



3. 高速缓冲存储器——地址映射与变换



组号	组内块号	块内地址
----	------	------

(a) Cache地址结构

区号	区内组号	组内块号	块内地址
----	------	------	------

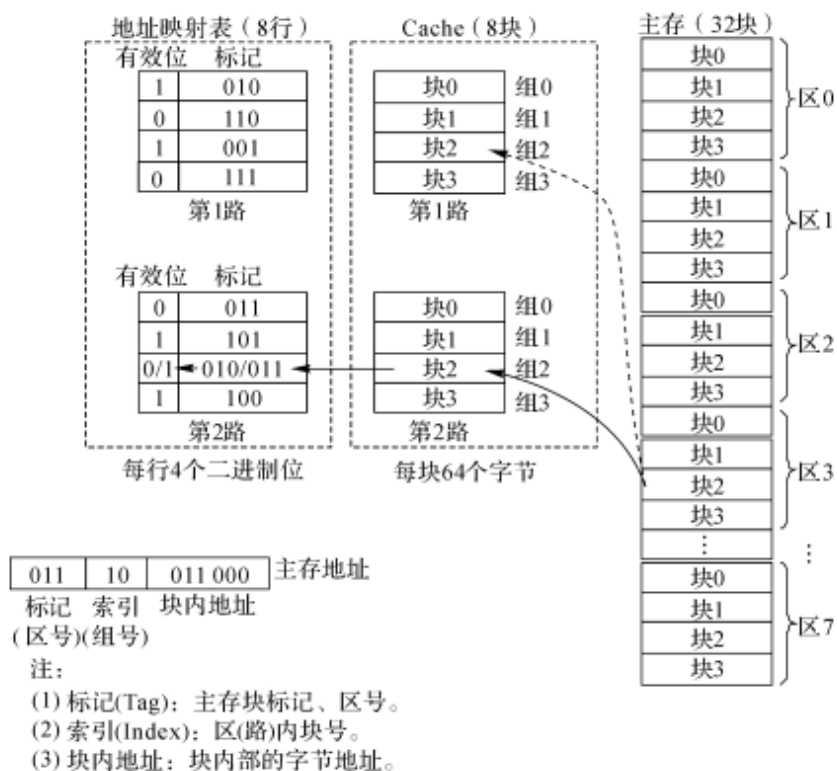
(b) 主存地址结构



3. 高速缓冲存储器——地址映射与变换

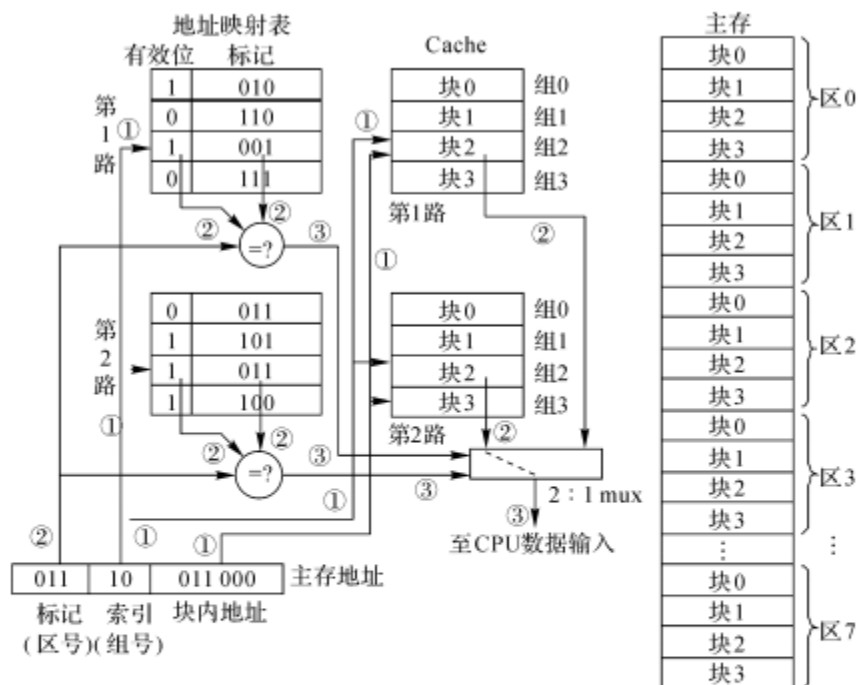
假设某计算机主存的最大寻址空间为2K，按字节编址，Cache共有512个字节，块大小为64个字节。因此，主存共32块(2KB/64B)，Cache共8块(512B/64B)。

若Cache采用2路组相联结构，则主存与Cache之间的地址映射方式如图





3. 高速缓冲存储器——地址映射与变换

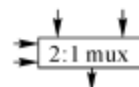


图例1:



数字比较器。
当G为“0”时，输出E为假。
当G为“1”时，若A≠B，
输出E为假；若A=B，输
出E为真

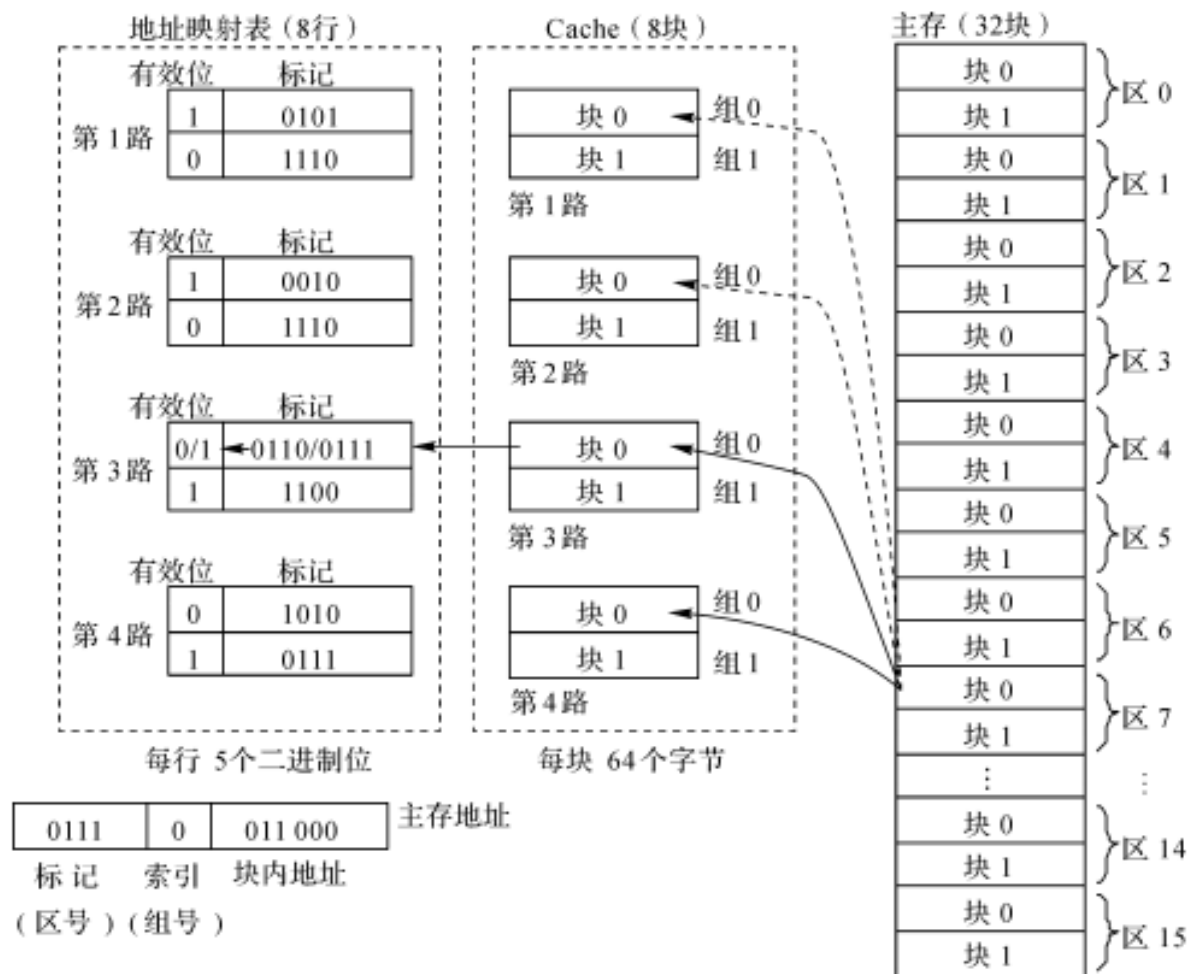
图例2:



二选一选择器。
上方两路数据输入，下方一
路数据输出，左侧是选择哪路
输入到达输出的控制信号

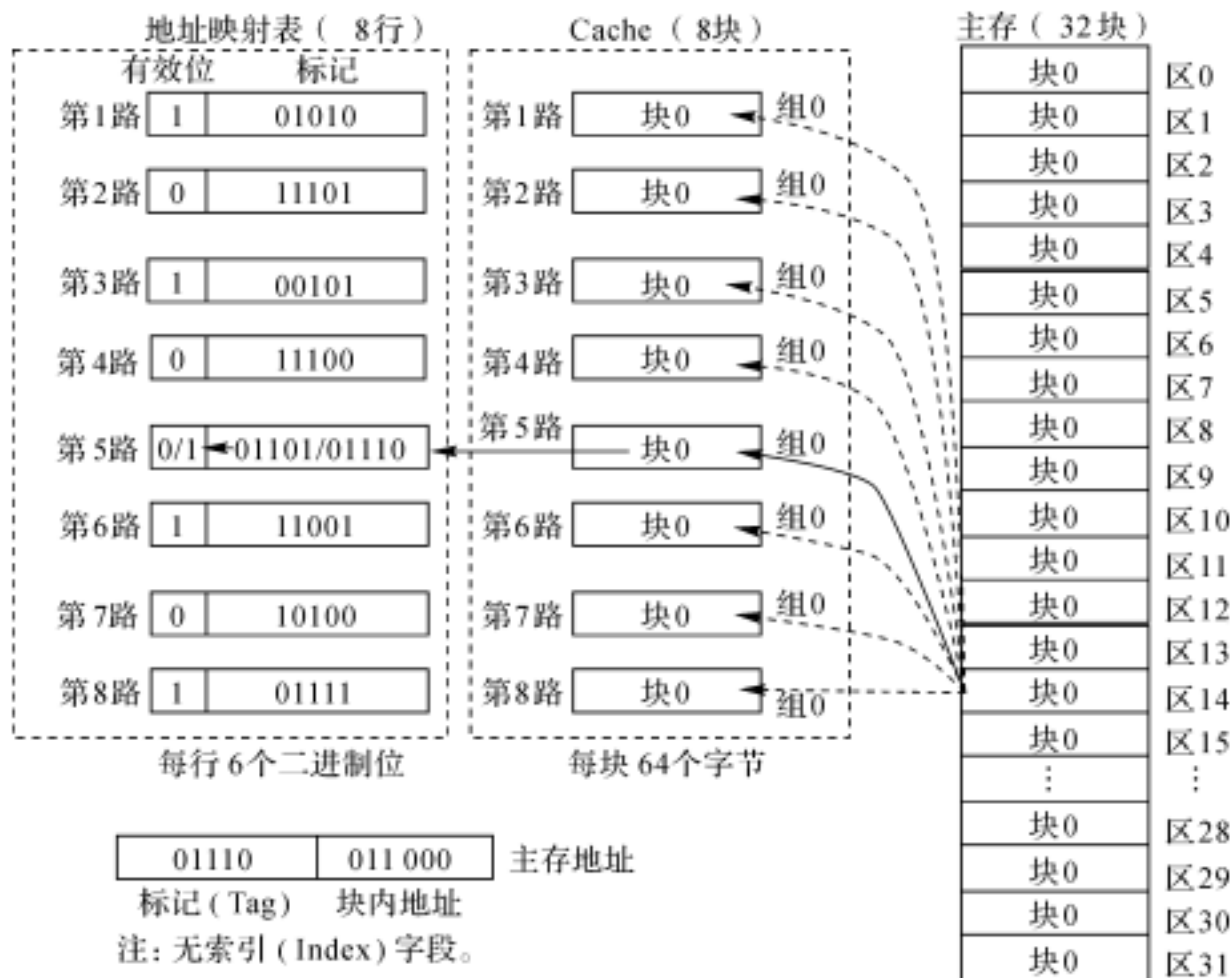


3. 高速缓冲存储器——地址映射与变换



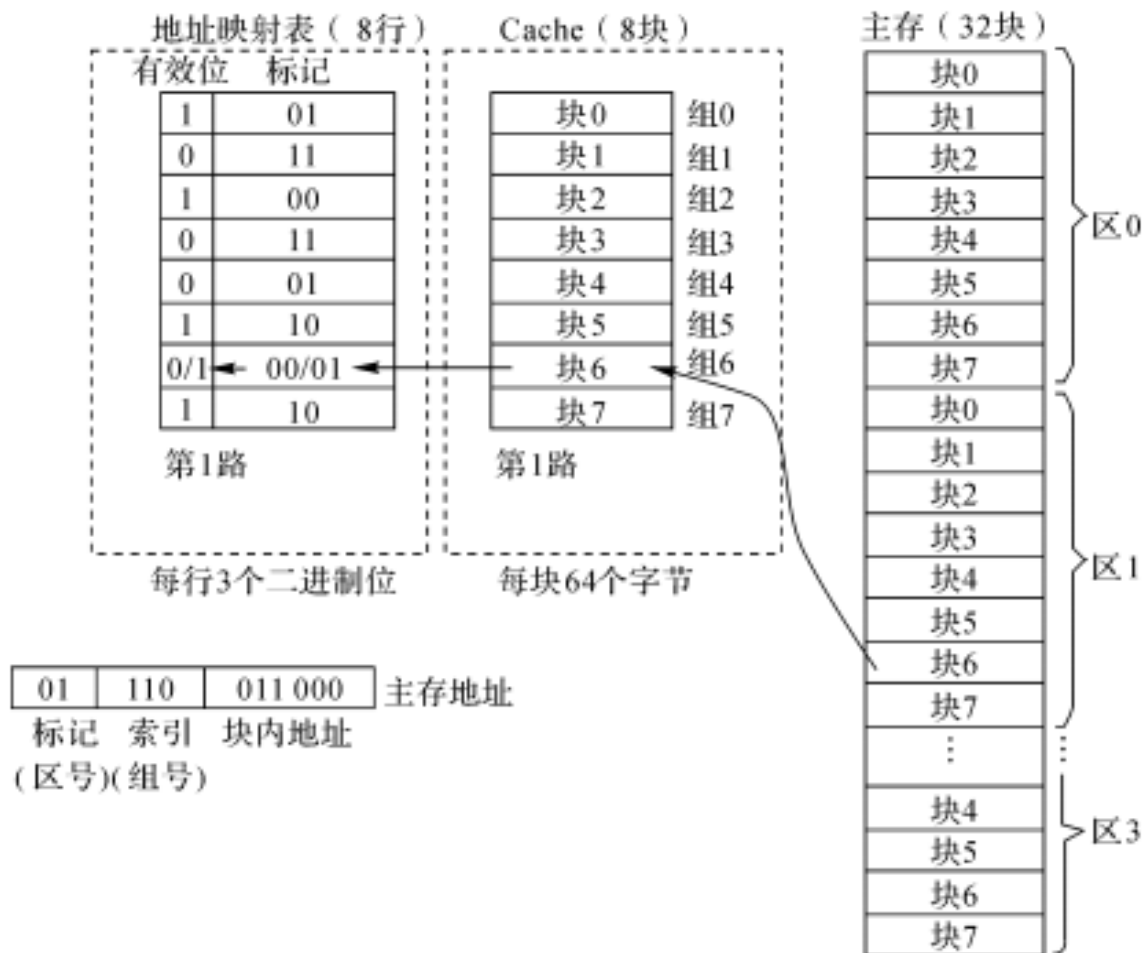


3. 高速缓冲存储器——地址映射与变换





3. 高速缓冲存储器——地址映射与变换





3. 高速缓冲存储器——替换算法

当要访问的主存块不在 Cache 中，即未命中时，就需要将主存块调入 Cache。在采用全相联地址映射和组相联地址映射时，如果当前 Cache 中没有空闲块，则需要在 Cache 中选择一个替换块，用于放置新调入的主存块。

主要的替换算法有如下几种：

- 随机替换(RAND)算法；
- 先进先出(First In First Out, FIFO)算法；
- 近期最少使用(Least Recently Used, LRU)算法；
- 最不经常使用(Least Frequently Used, LFU)算法；
- 最优替换(Optional Replacement, OPT)算法。



例4.4 假定程序在主存中占用了5个存储块,Cache容量为3个存储块,采用全相联映射。CPU 执行程序的顺序为P2、P3、 P2、 P1、 P5、 P2、 P4、 P5、 P3、 P2、 P5、 P2。试分析采用 FIFO 算法、LRU 算法和 OPT 三种算法的 Cache命中情况。

解 采用 FIFO 算法、LRU 算法和 OPT 三种算法的 Cache命中情况如表4.3所示。



表 4.3 例 4.4 三种替换算法的比较

时间序列		1	2	3	4	5	6	7	8	9	10	11	12
程序访问的块号		P2	P3	P2	P1	P5	P2	P4	P5	P3	P2	P5	P2
(a) FIFO 算法	缓存装入 的块号	P2	P2	P2	P2	P5	P5	P5	P5	P3	P3	P3	P3
			P3	P3	P3	P3	P2	P2	P2	P2	P2	P5	P5
					P1	P1	P1	P4	P4	P4	P4	P4	P2
	是否命中			★					★		★		
(b) LRU 算法	缓存装入 的块号	P2	P2	P2	P2	P2	P2	P2	P2	P3	P3	P3	P3
			P3	P3	P3	P5	P5	P5	P5	P5	P5	P5	P5
					P1	P1	P1	P4	P4	P4	P2	P2	P2
	是否命中			★			★		★			★	★
(c) OPT 算法	缓存装入 的块号	P2	P2	P2	P2	P2	P2	P4	P4	P4	P2	P2	P2
			P3	P3	P3	P3	P3	P3	P3	P3	P3	P3	P3
					P1	P5	P5	P5	P5	P5	P5	P5	P5
	是否命中			★			★		★	★		★	★

注：① 背景为灰色的块是下次将要被替换的块。

② ★标注 Cache 命中的时刻。



THE END !

THANKS