



# 第七章 存储系统(二)

---

秦磊华 计算机学院

---

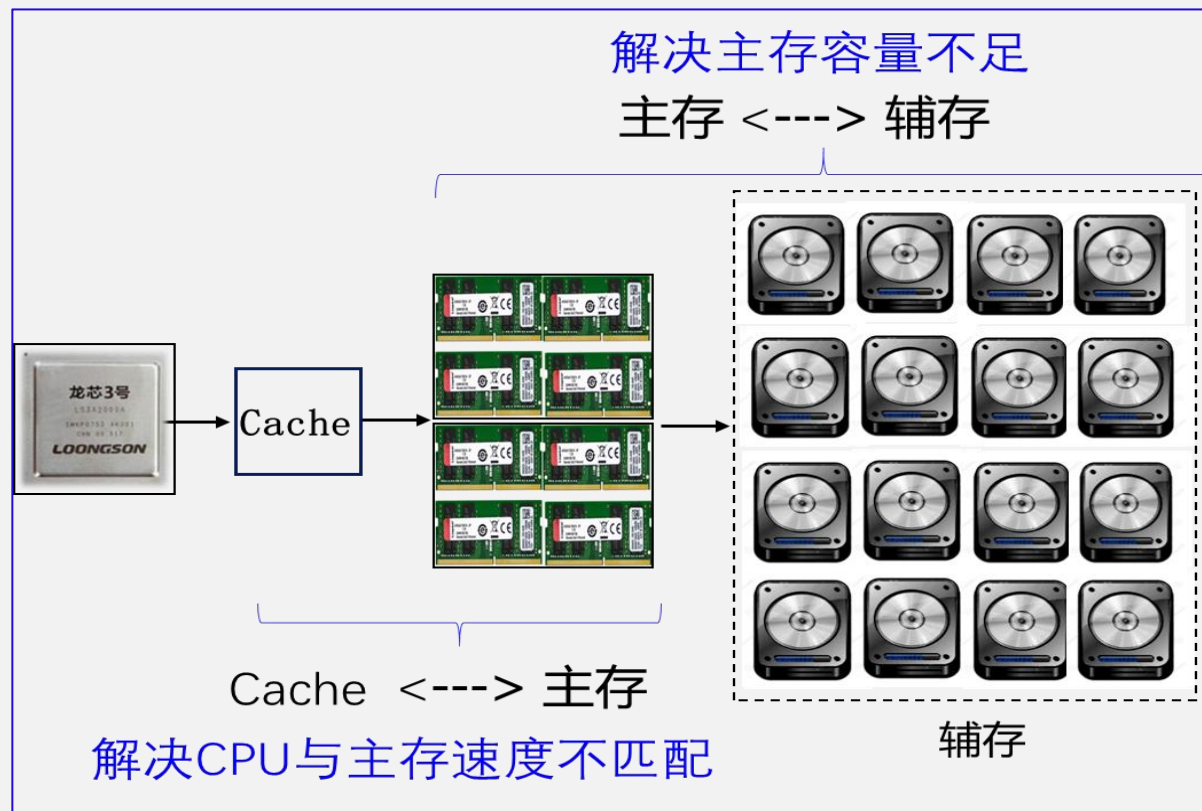
## 7.6 高速缓存存储器

CONTENT



## 7.6 高速缓冲存储器

### 1. 存储系统中的Cache视图



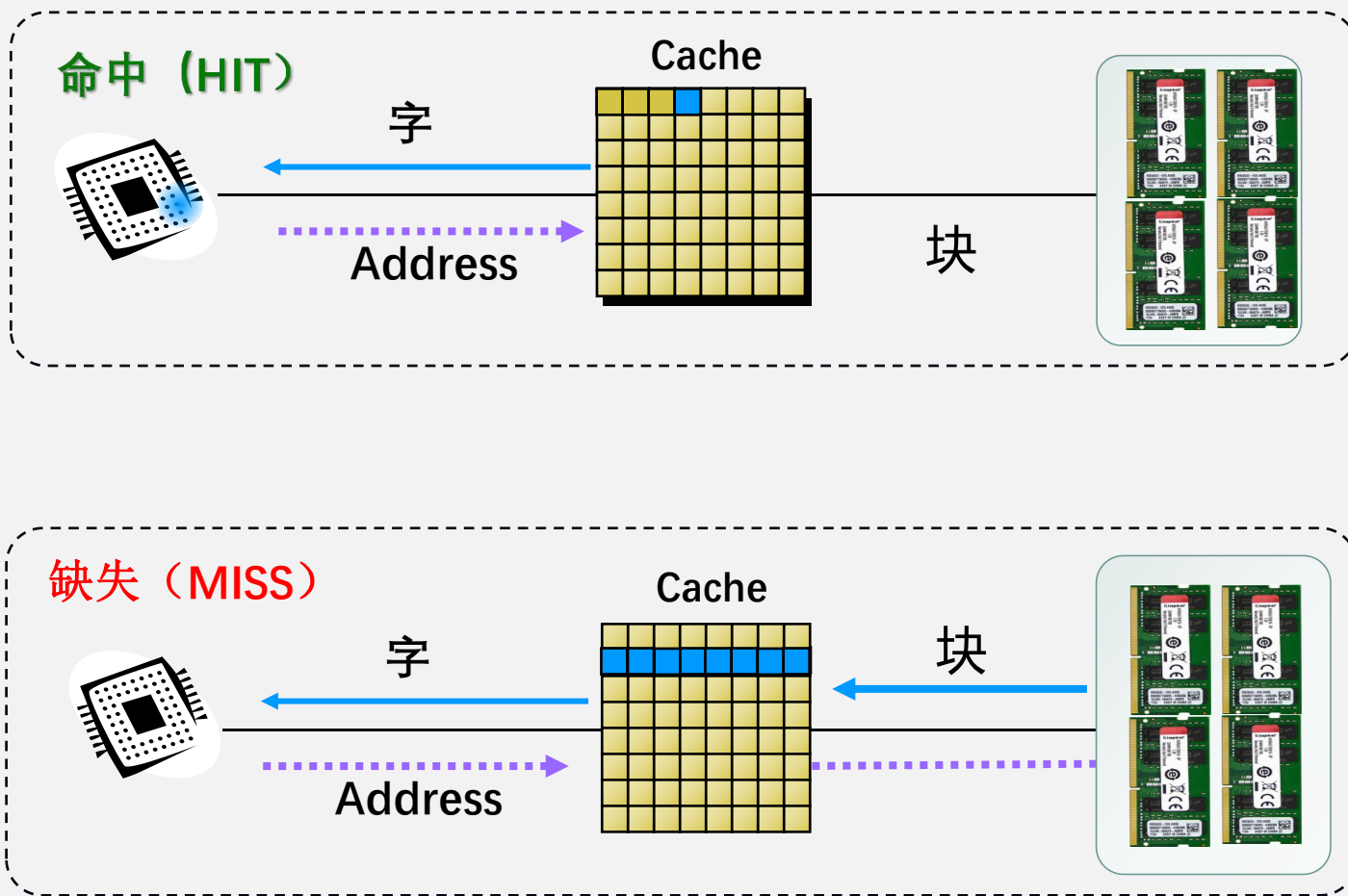
Cache的理论基础



时间局部性和空间  
局部性

## 7.6 高速缓冲存储器

### 2. Cache 的工作过程(读)



如何? 根据什么?  
判断数据在  
Cache中

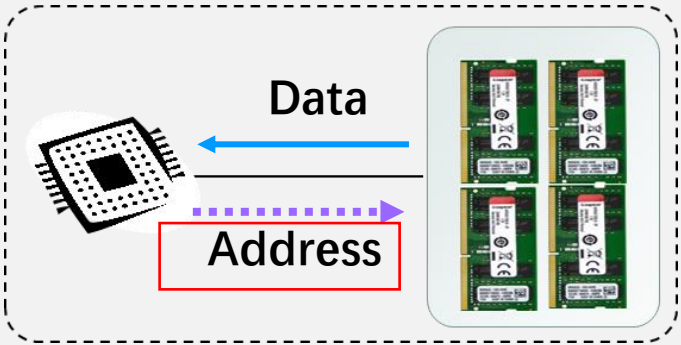
数据从主存调入  
Cache中该如何  
存放, 便于下次  
命中时好查找?

## 3. 相联存储器

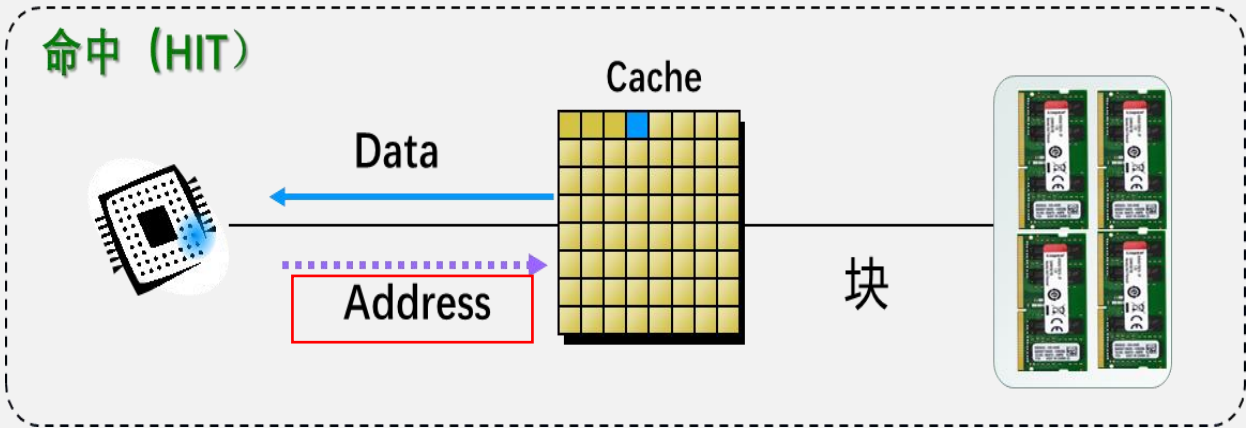
主存基于地址访问



速度慢



物理地址	工号	姓名	出生年月
N	001	张帅	1976/7
N+1	021	李猛	1978/9
N+2	023	郝一梅	1977/6

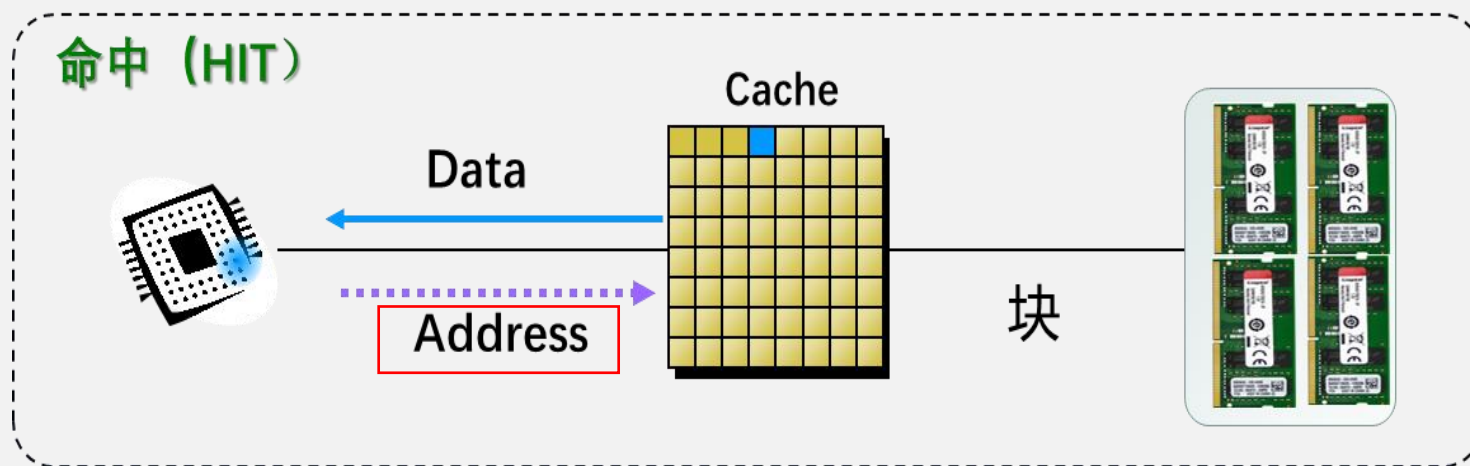


访问Cache和主存的地址相同？

地址与要访问的对象之间有必然联系吗？

## 7.6 高速缓冲存储器

### 3. 相联存储器



数据结构学过多种算法:顺序、二分查找、哈希查找等

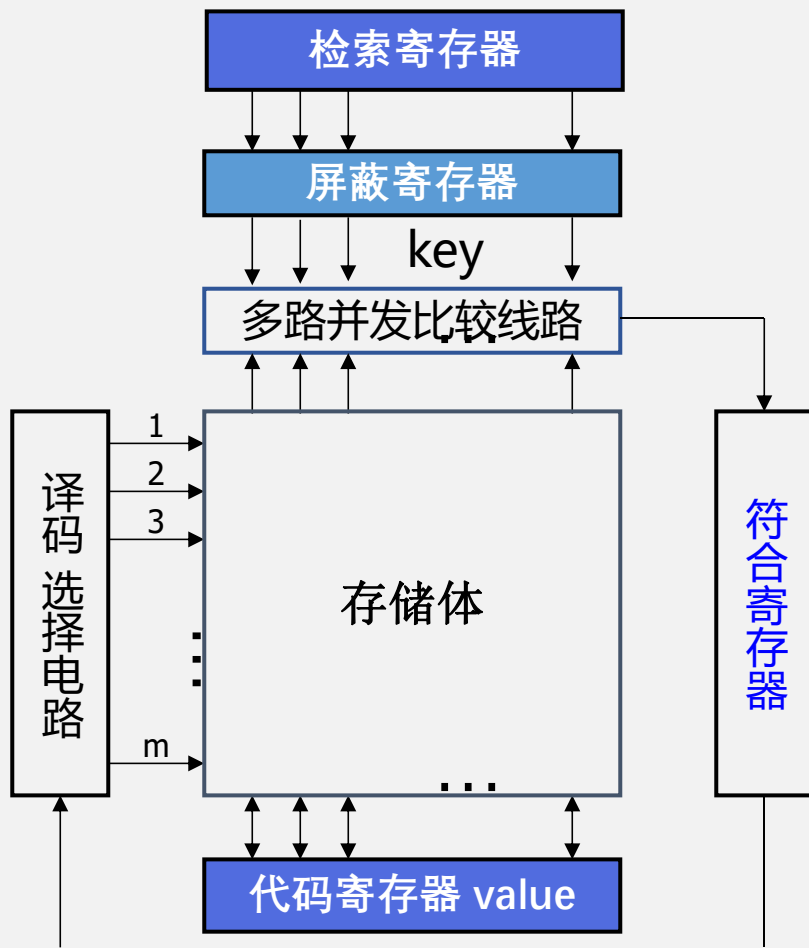
如何快速查找?

Cache

从地址中剥离特征值, 用**相联存储器**, 通过硬件并发查找

## 7.6 高速缓冲存储器

### 3. 相联存储器



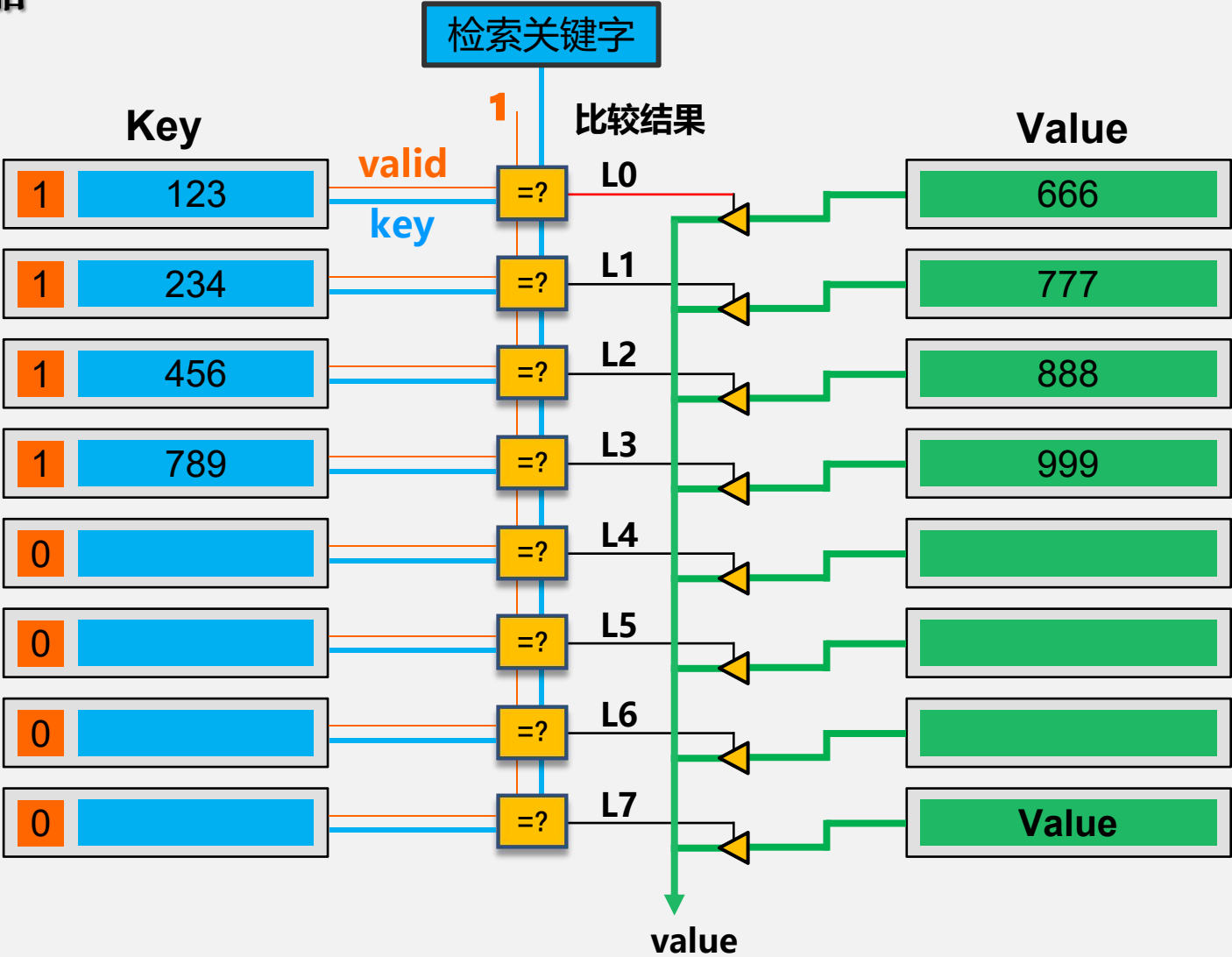
- ◆从地址中剥离出Key
- ◆以Key 作比较对应送硬件并发比较
- ◆存储体存放数据及Key等信息
- ◆符合寄存器存放Cache 命中行信息
- ◆根据符合寄存器的信息取出命中行的数据

相联存储器本质上是一种按内容访问的存储器，  
简称为CAM (Content Addressable Memory)



# 7.6 高速缓冲存储器

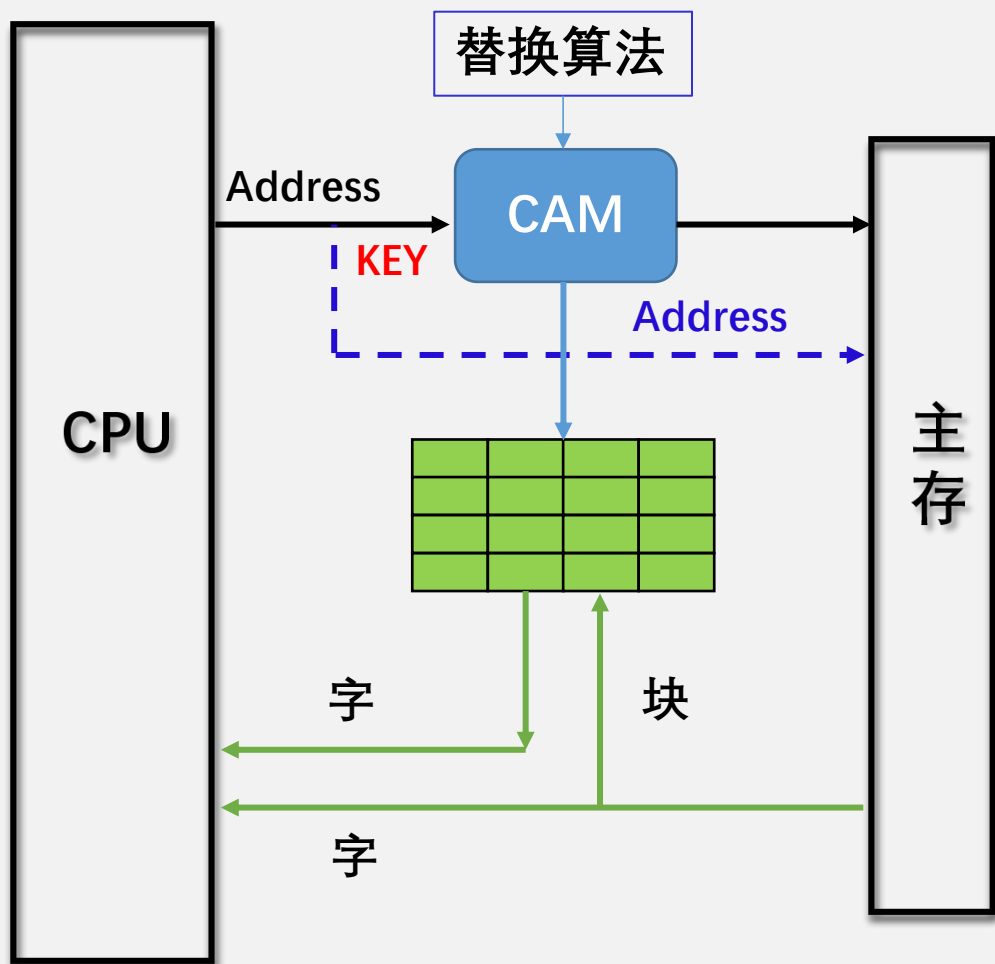
## 3. 相联存储器





## 7.6 高速缓冲存储器

### 4. Cache的基本结构与工作原理

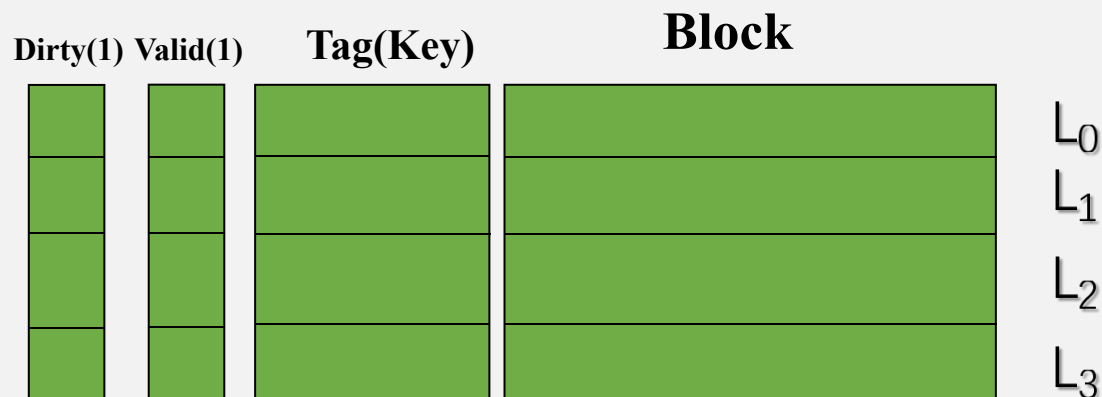
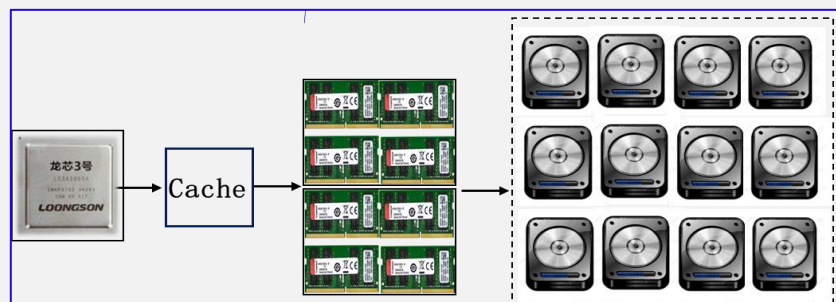


- ◆ 从地址中剥离出Key的方法
- ◆ 是否还需要剥离Cache中的定位信息
- ◆ 是否能减少非数据对Cache容量的消耗
- ◆ 是否有利于Cache中实现快速查找

## 7.6 高速缓冲存储器

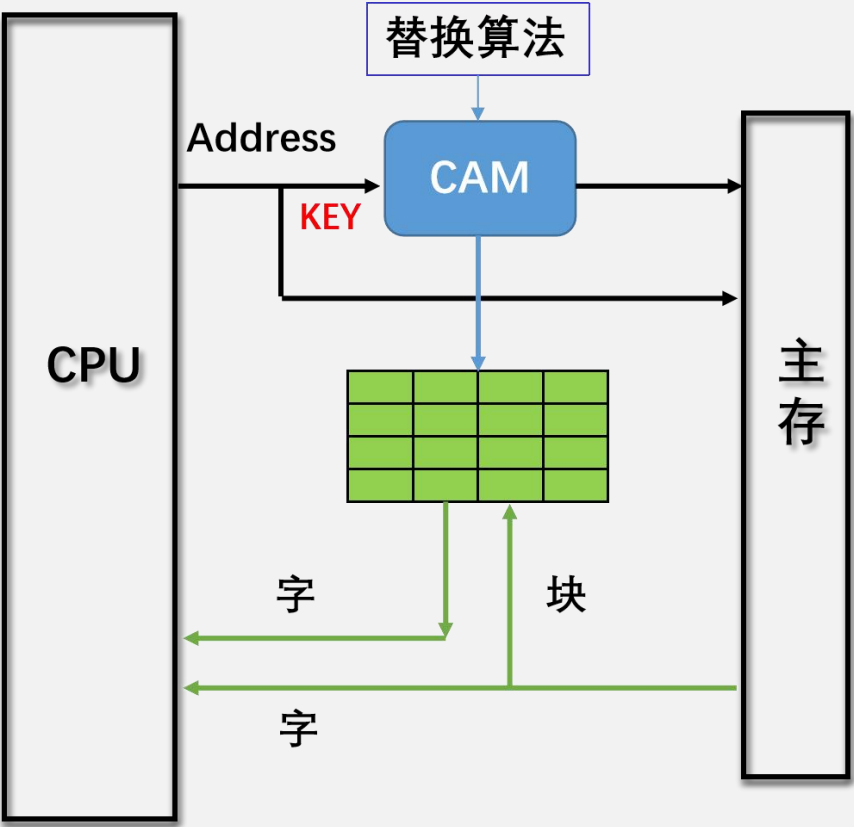
### 4. Cache的基本结构与工作原理

#### (1) Cache的存储体结构



- ◆ Cache被分成若干行，每行的大小与主存块相同
- ◆ Cache存储体存储的信息：数据块、Tag、Valid(1)、Dirty(1) （Cache状态信息）
- ◆ 脏位(修改位):表示Cache的内容是否被修改，被修改时设为1,用于写操作的一致性检查
- ◆ 有效位:Cache中的数据是否有效，有效时为1，用于读操作时数有效性判断。

## 4. Cache的基本结构与工作原理

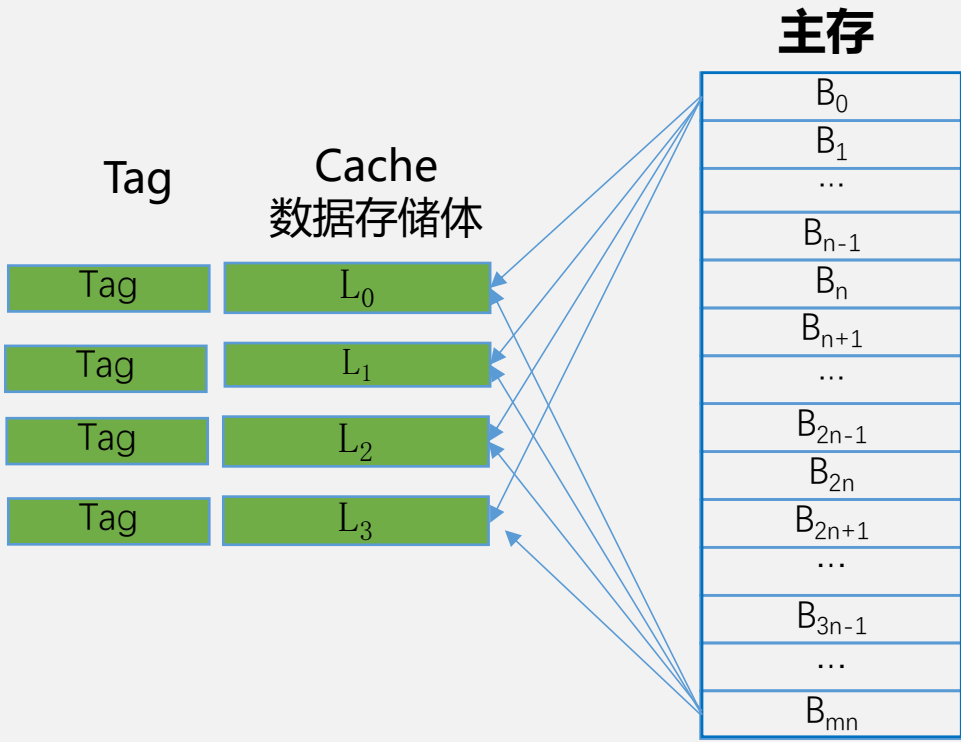


- ◆主存数据如何迁至Cache才能实现快速查找?
- ◆常见的三种映射方式
  - 全相联 (fully-associated)
  - 直接相联 (direct mapped)
  - 组相联 (set-associated)

# 7.6 高速缓冲存储器

## 4. Cache的基本结构与工作原理

### (2)全相联 (fully-associated)



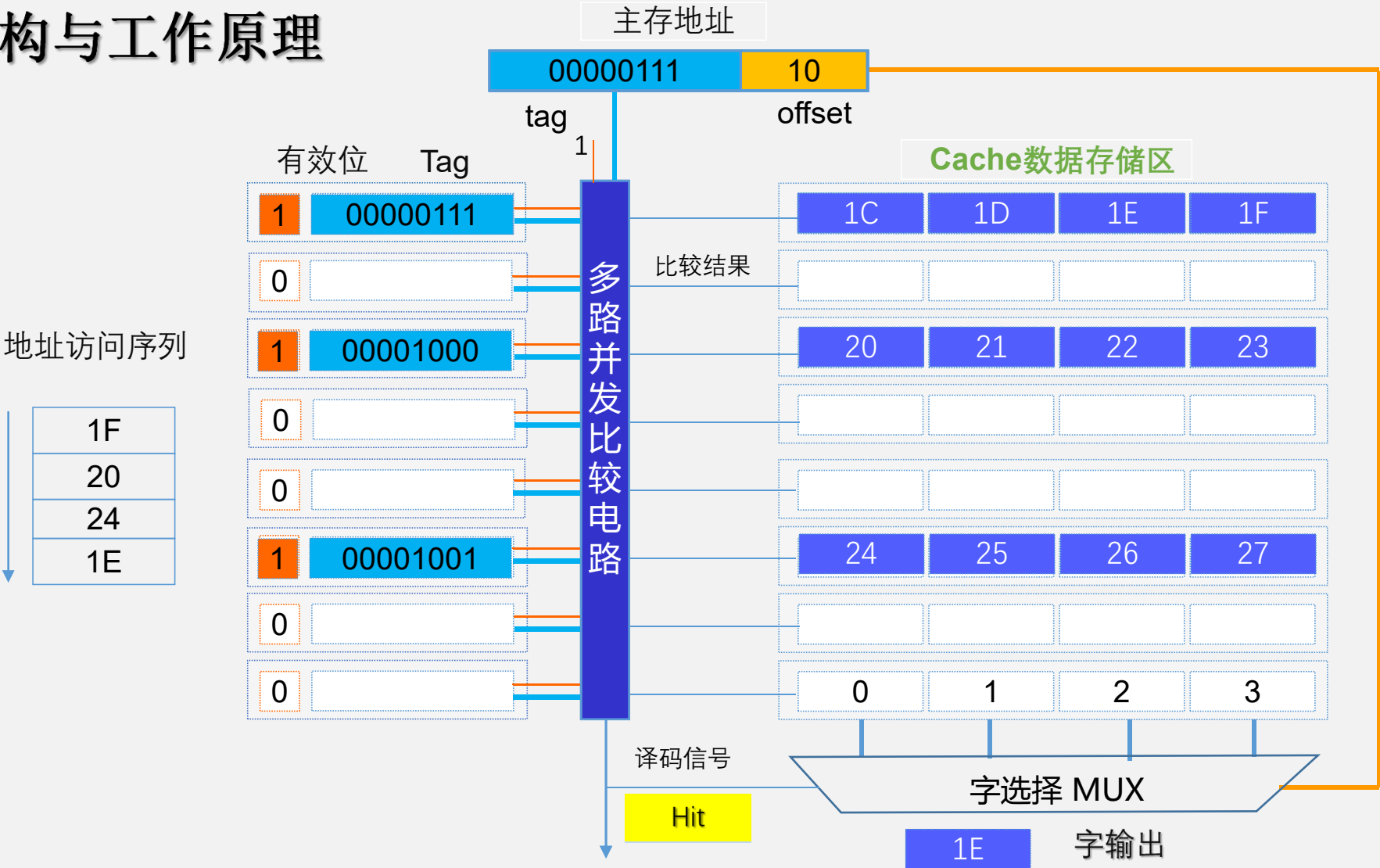
- ◆主存分块，Cache行（Line），两者大小相同；
- ◆设每块4个字，主存大小为1024字，则第61个字的主存地址为：  
 00001111 01 （块号 块内地址）
- ◆主存分块后地址从一维变成二维；
- ◆映射方法：主存数据块可映射到Cache全部行，数据从主存传输到Cache数据存储体的同时，将该数据块公共块号存入对应行的标记存储体中。



# 7.6 高速缓冲存储器

## 4. Cache的基本结构与工作原理

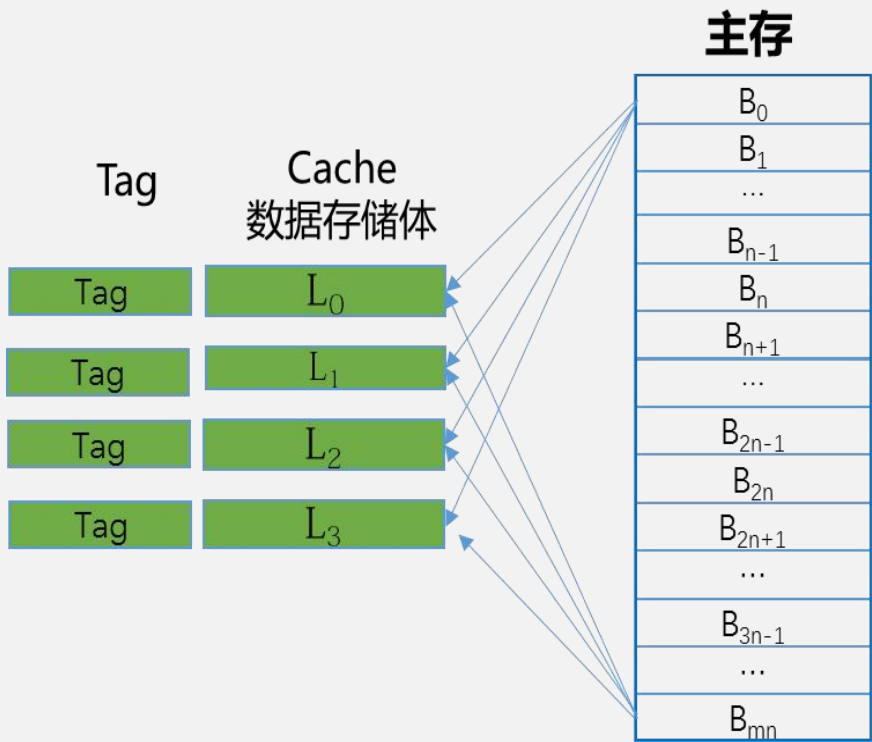
### (2)全相联



## 7.6 高速缓冲存储器

### 4. Cache的基本结构与工作原理

#### (2)全相联



#### 全相联映射的特点

- Cache利用率高
- 冲突率低
- 比较电路复杂

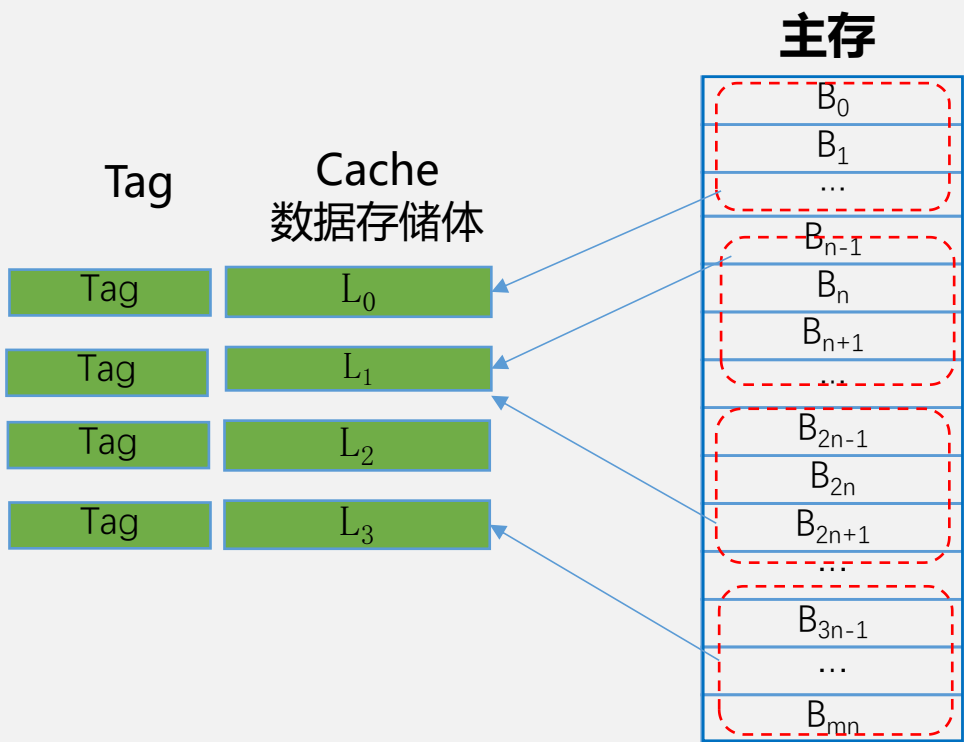
#### 应用场合

- 小容量Cache

## 7.6 高速缓冲存储器

### 4. Cache的基本结构与工作原理

#### (3)直接映射



◆主存分块，Cache行（Line），两者大小相同；

◆主存分块后还将以Cache行数为标准进行分区，

◆设每块4个字，主存大小为1024个字，

Cache分4行(每区4块)，第61个字的主存地址为：

000011 11 01 （区号，区内块号，块内地址）

主存地址从一维变成三维；

◆映射方法：

Cache共n行，主存第j块映射到Cache的行号为：

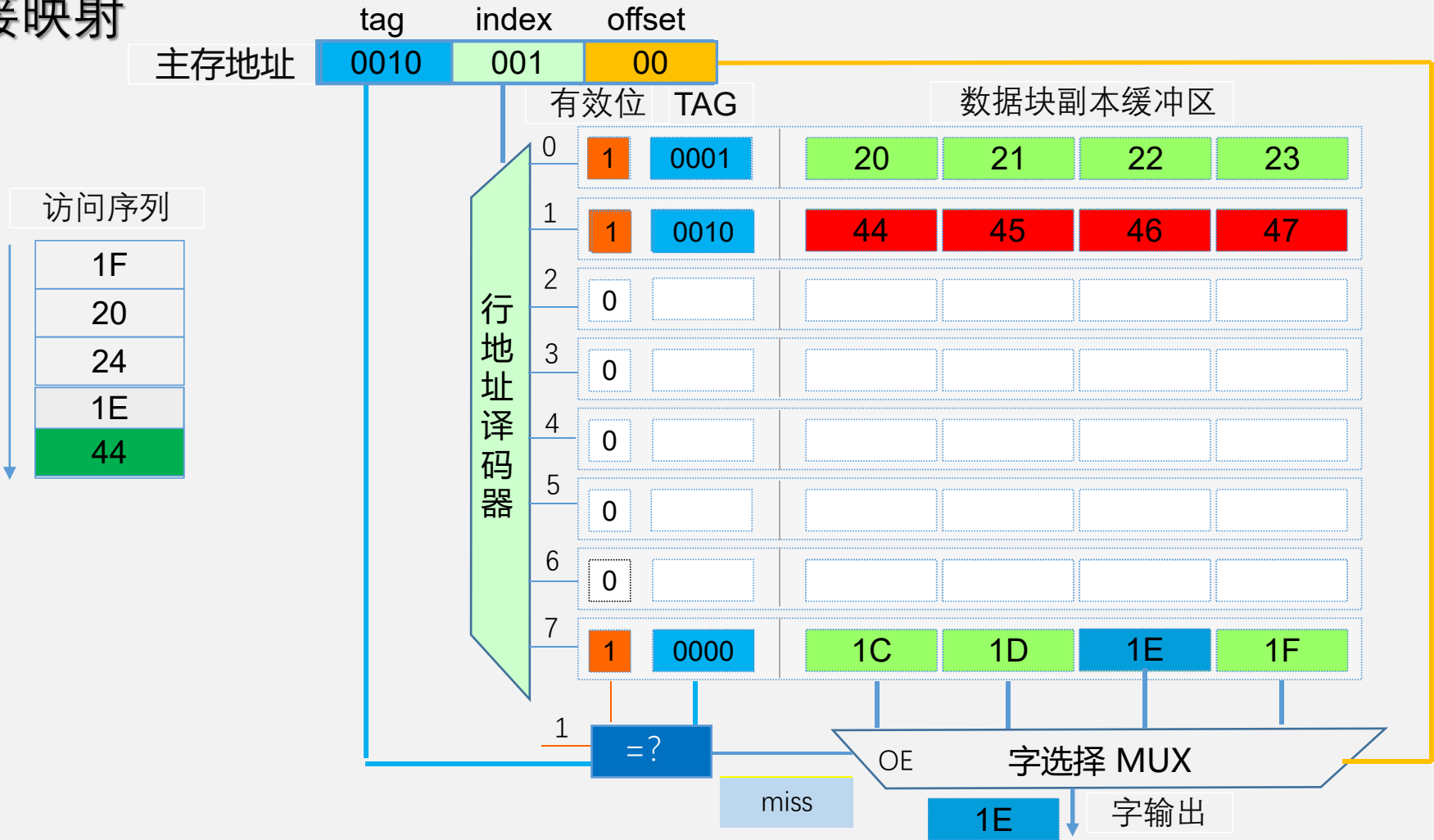
$$i = j \bmod n$$

即主存的数据块映射到Cache特定行

此时Tag为多少位？

## 4. Cache的基本结构与工作原理

### (3)直接映射

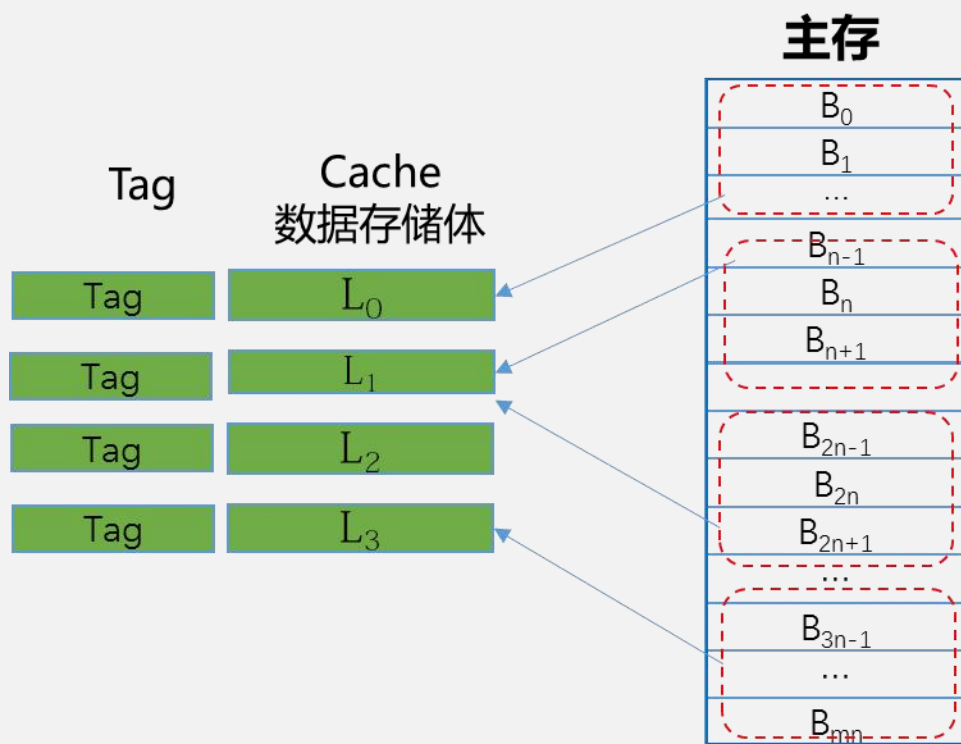




## 7.6 高速缓冲存储器

### 4. Cache的基本结构与工作原理

#### (3)直接映射



#### 直接映射的特点

- Cache利用率低
- 块冲突率高
- 比较电路简单

#### 应用场合

- 大容量Cache

#### 全相联映射的特点

- Cache利用率高
- 块冲突率低
- 比较电路复杂

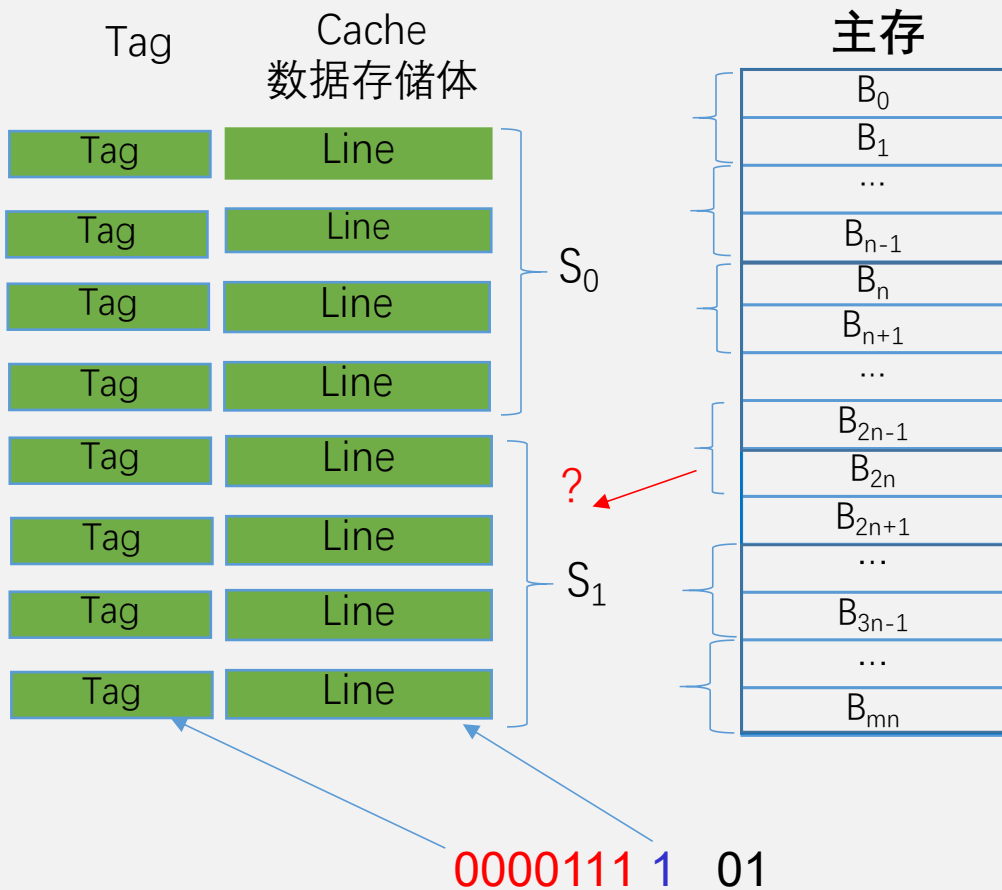
#### 应用场合

- 小容量Cache

## 7.6 高速缓冲存储器

### 4. Cache的基本结构与工作原理

#### (4)组相联映射



◆主存分块，Cache行（Line），两者大小相同；

◆ **Cache分组**（每组中包k行），本例假定K=4

◆ 主存分块后还将以**Cache组数**为标准进行分组；

◆ 设每块4个字，主存大小为1024个字，  
Cache分为8行，2组，第61个字的主存地址为：  
**0000111 1 01**（主存组号，组内块号，块内地址）

主存地址从一维变成三维；

◆映射方法法：

Cache共n组，主存第j块映射到Cache 的组号为：

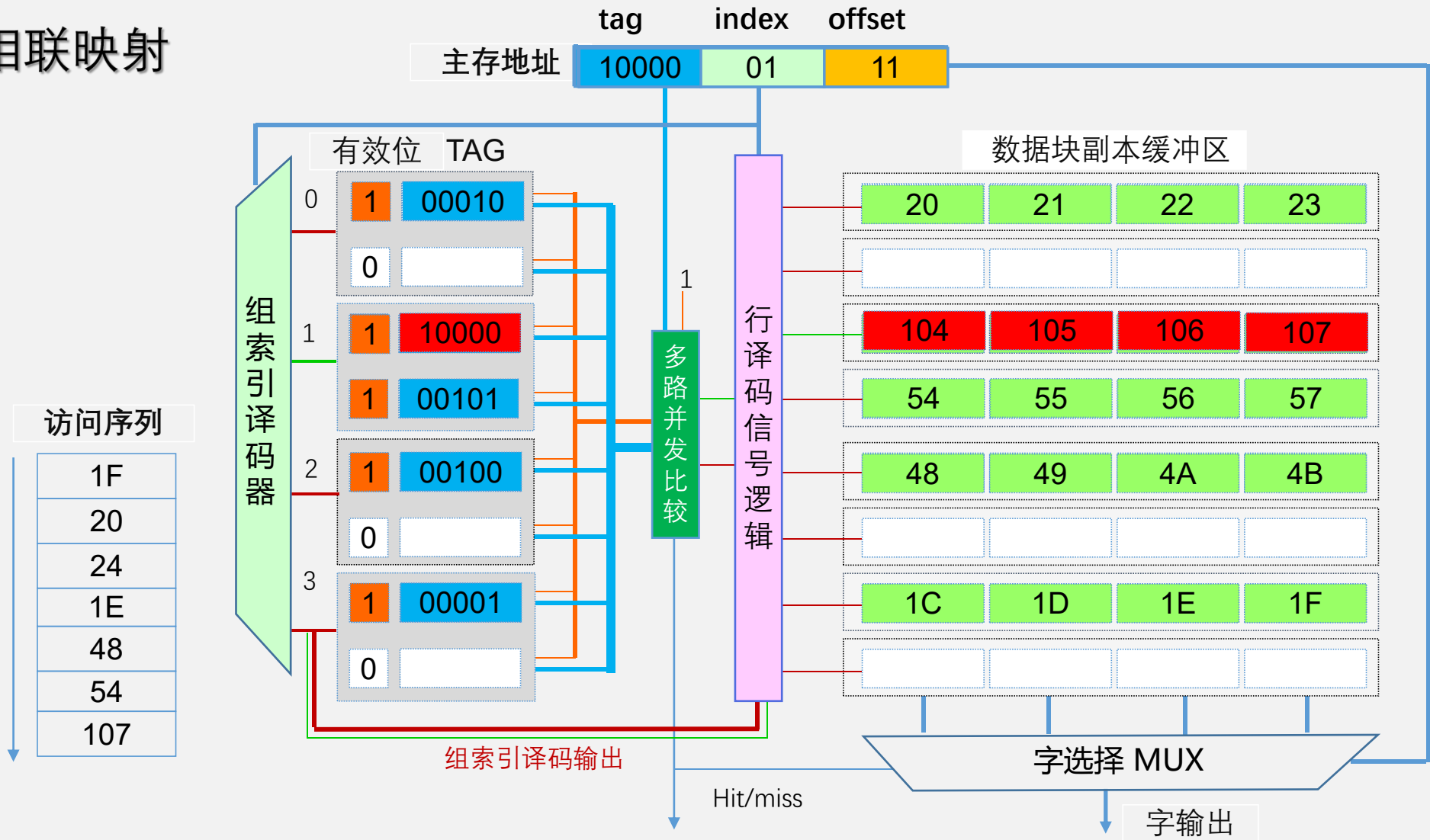
$$i = j \bmod n$$

即主存的数据块映射到Cache**特定组**的**任意行**

## 7.6 高速缓冲存储器

### 4. Cache的基本结构与工作原理

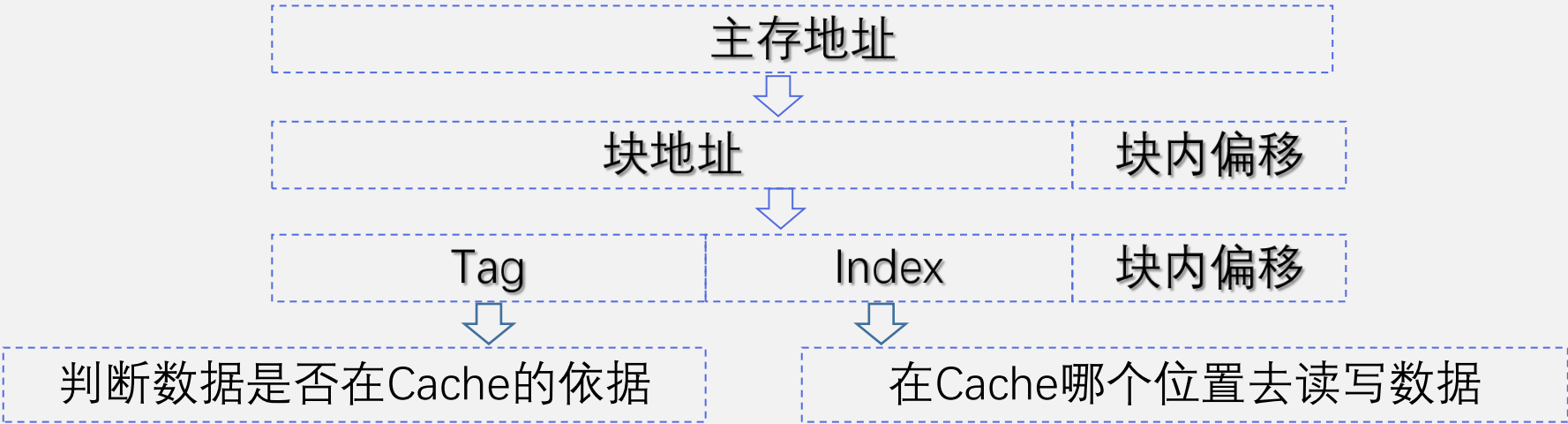
#### (4)组相联映射



# 7.6 高速缓冲存储器

## 4. Cache的基本结构与工作原理

### (5)三种映射机制中的主存地址变化特征

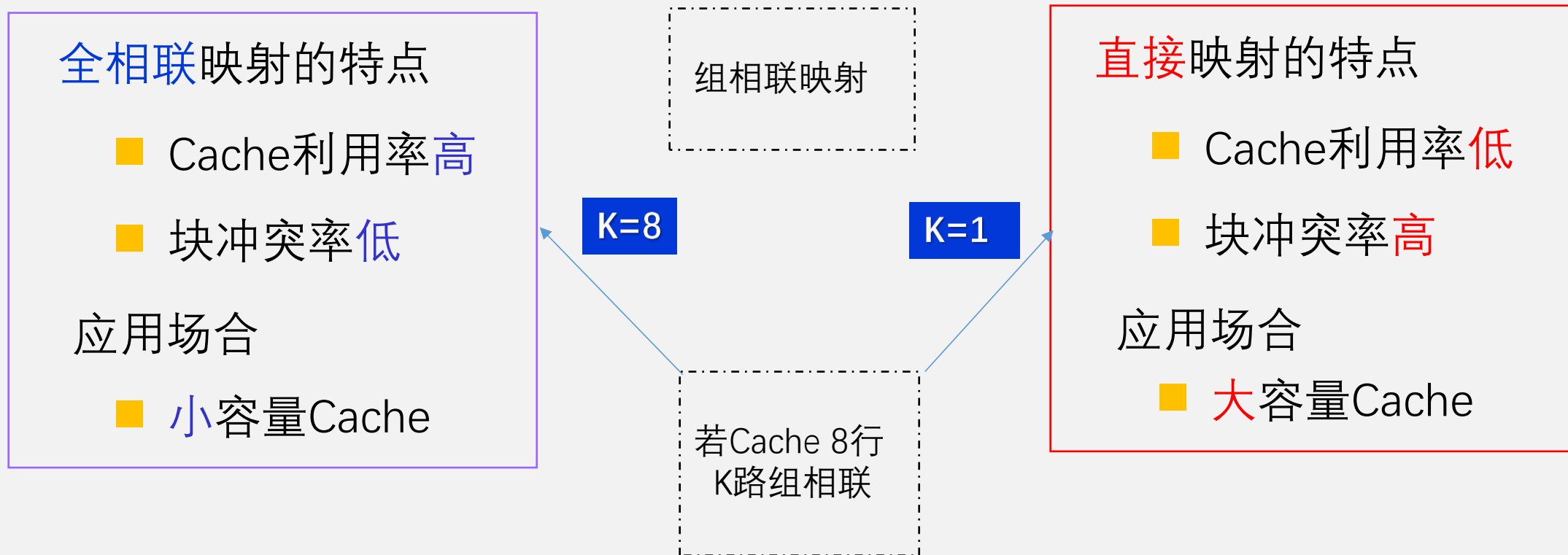


全相联	主存块地址 (标记)		块内偏移
直接相联	区号标记	Cache行索引	块内偏移
组相联	组号标记	Cache组索引	块内偏移

## 7.6 高速缓冲存储器

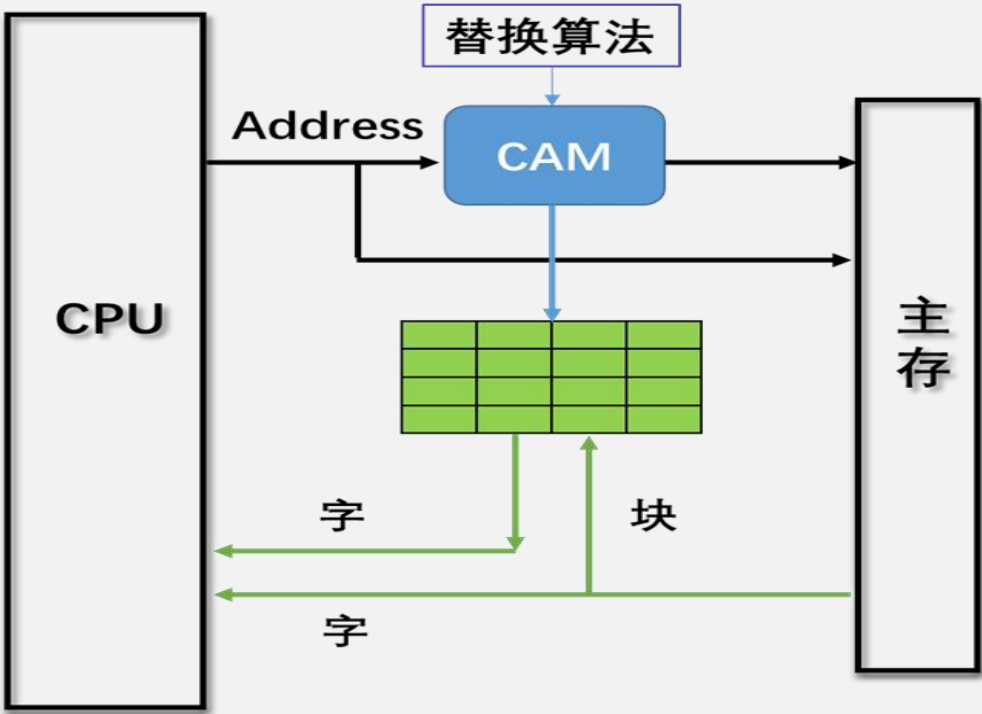
### 4. Cache的基本结构与工作原理

#### (5)三种映射机制中的主存地址变化特征



## 7.6 高速缓冲存储器



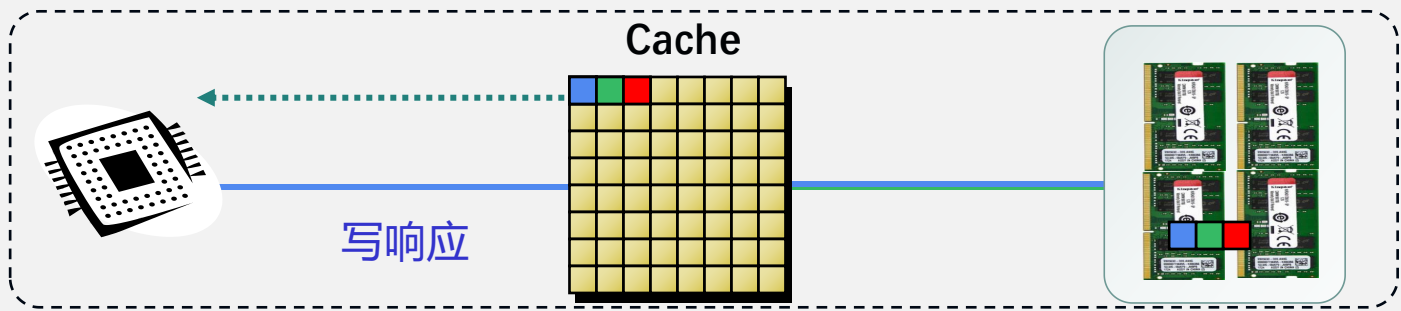


CPU执行写数据时，涉及到数据修改，导致Cache与主存数据不一致，因此，写策略要考虑的问题较为复杂！

## 5. Cache的写操作

### (1) 写回法(Write Back)

写Cache命中时：将结果写入Cache，当Cache数据被替换出时才写回主存



- ◆ 写操作时Cache的作用发挥了吗？
- ◆ 写操作未命中时要将地址对应的主存块迁移到Cache中？
  - WBWA (Write--Back--with--Write--Allocate)
  - WBWNA (Write Back-Write-NO-Allocate)

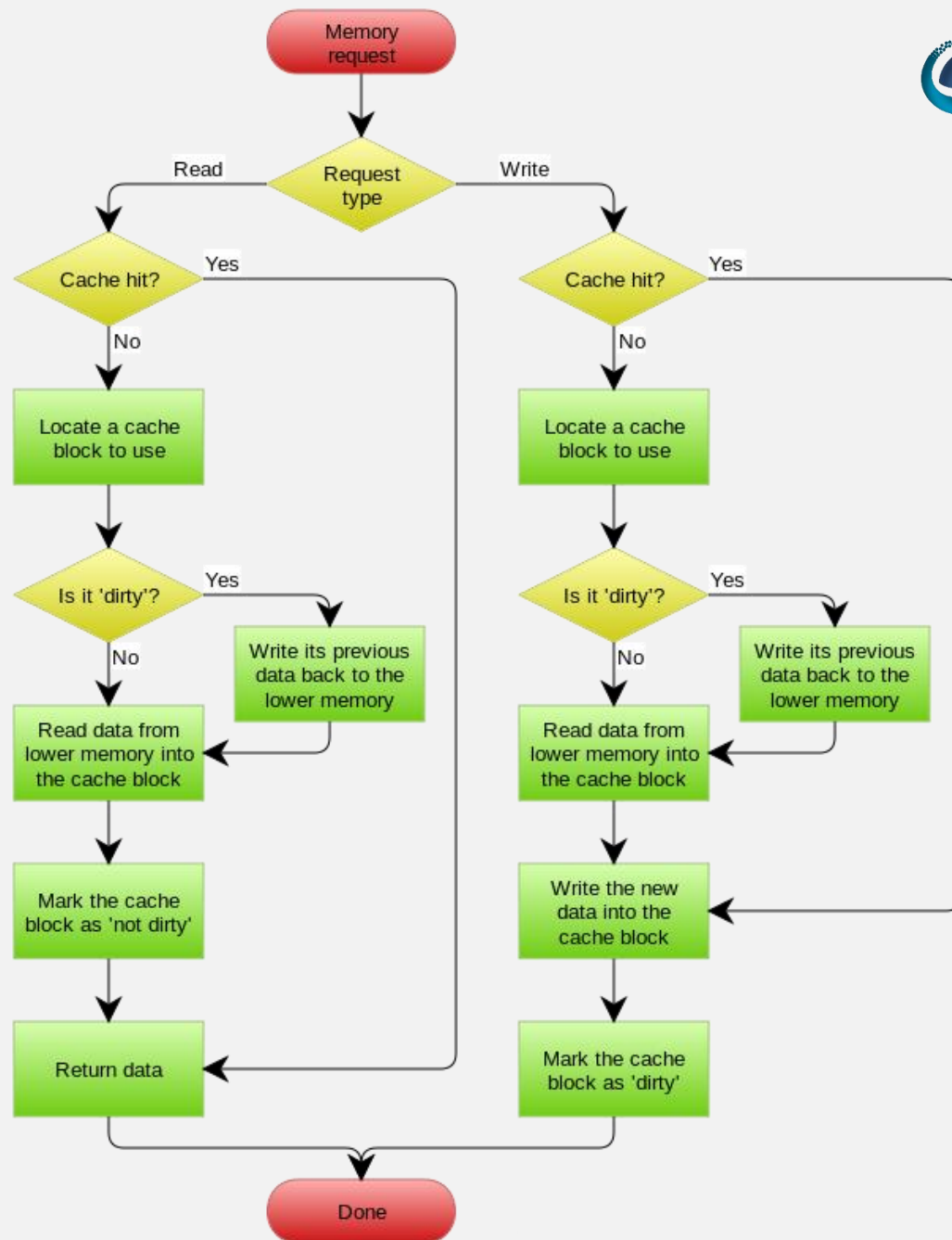


## 7.6 高速缓冲存储器

### 5. Cache的写操作

#### (1) 写回法(Write Back)

Write--Back--with--Write--Allocate

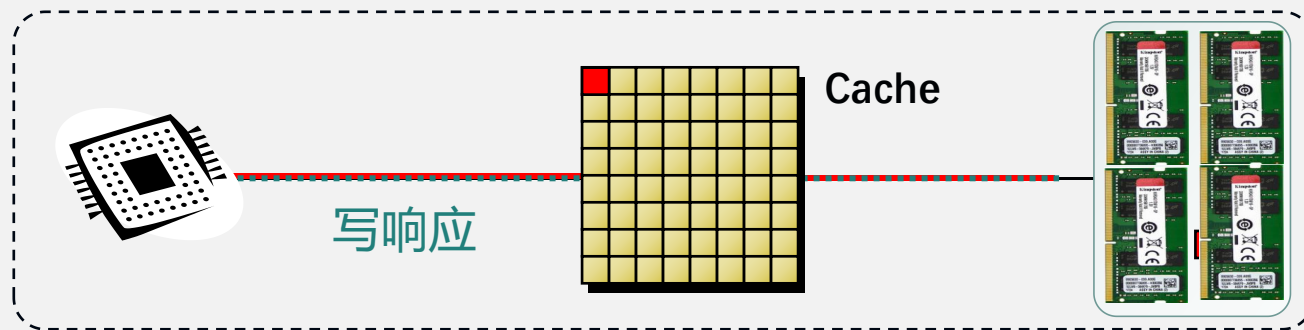


## 7.6 高速缓冲存储器

### 5. Cache的写操作

#### (2) 写直达法(Write Through)

写Cache命中时：既写入Cache时同时写入主存。



- ◆ 写操作时Cache的作用发挥了吗？
- ◆ 写操作未命中时要将地址对应的主存块迁移到Cache中吗？



WTNWA法 (Write Through--with NO-Write--Allocate)

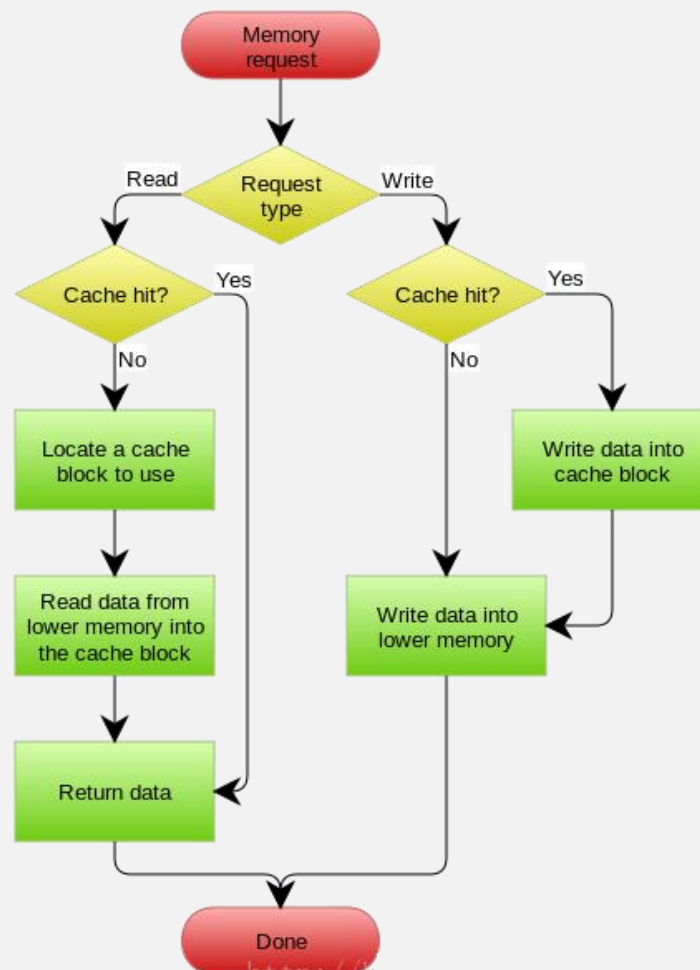
## 7.6 高速缓冲存储器

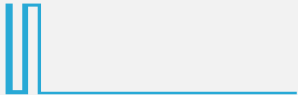
### 5. Cache的写操作

#### (2) 写直达法(Write Through)

WT (write through )+NWA(no write allocation)

此时不需要设置脏位!





# 第二部分完