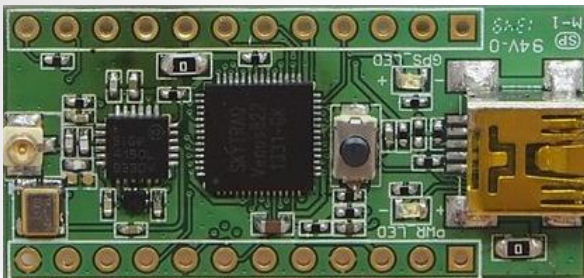


计算机组成原理

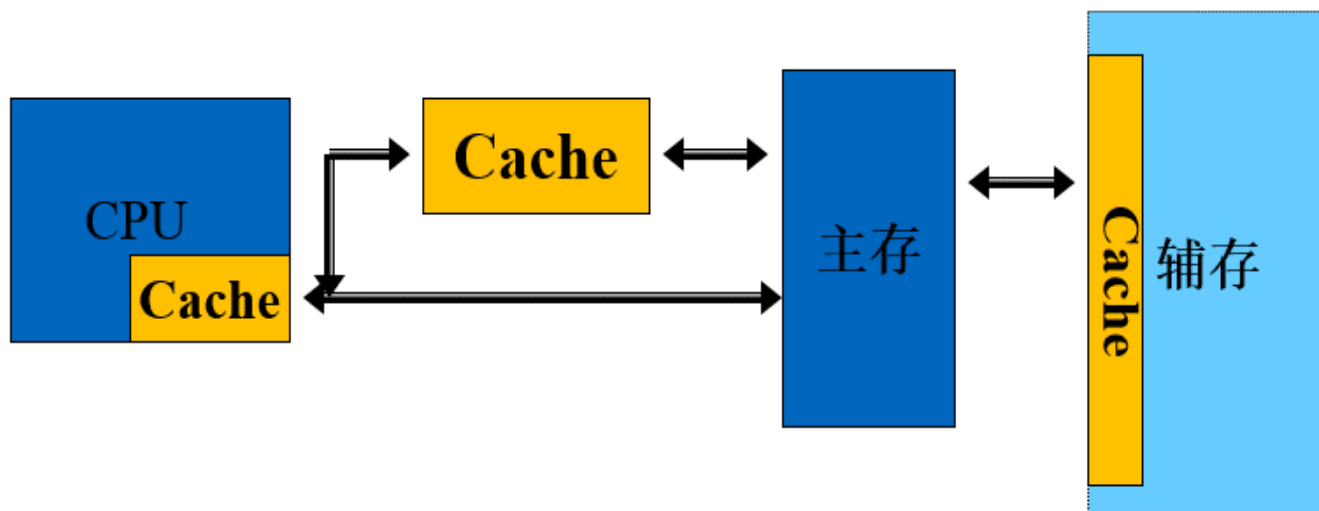
第四章 存储系统

4.9 Cache地址映射与变换方法



1

主存与cache地址映射概述



- 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

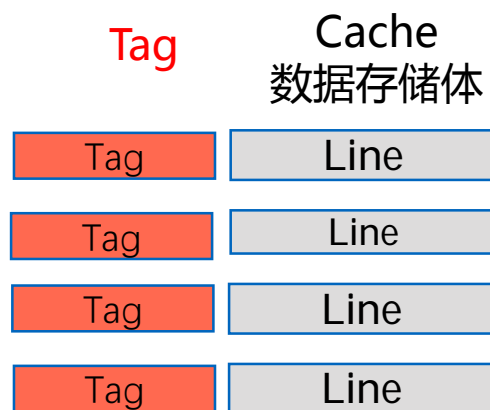
全相联 (fully-associated)

直接相联 (direct mapped)

组相联 (set-associated)

2

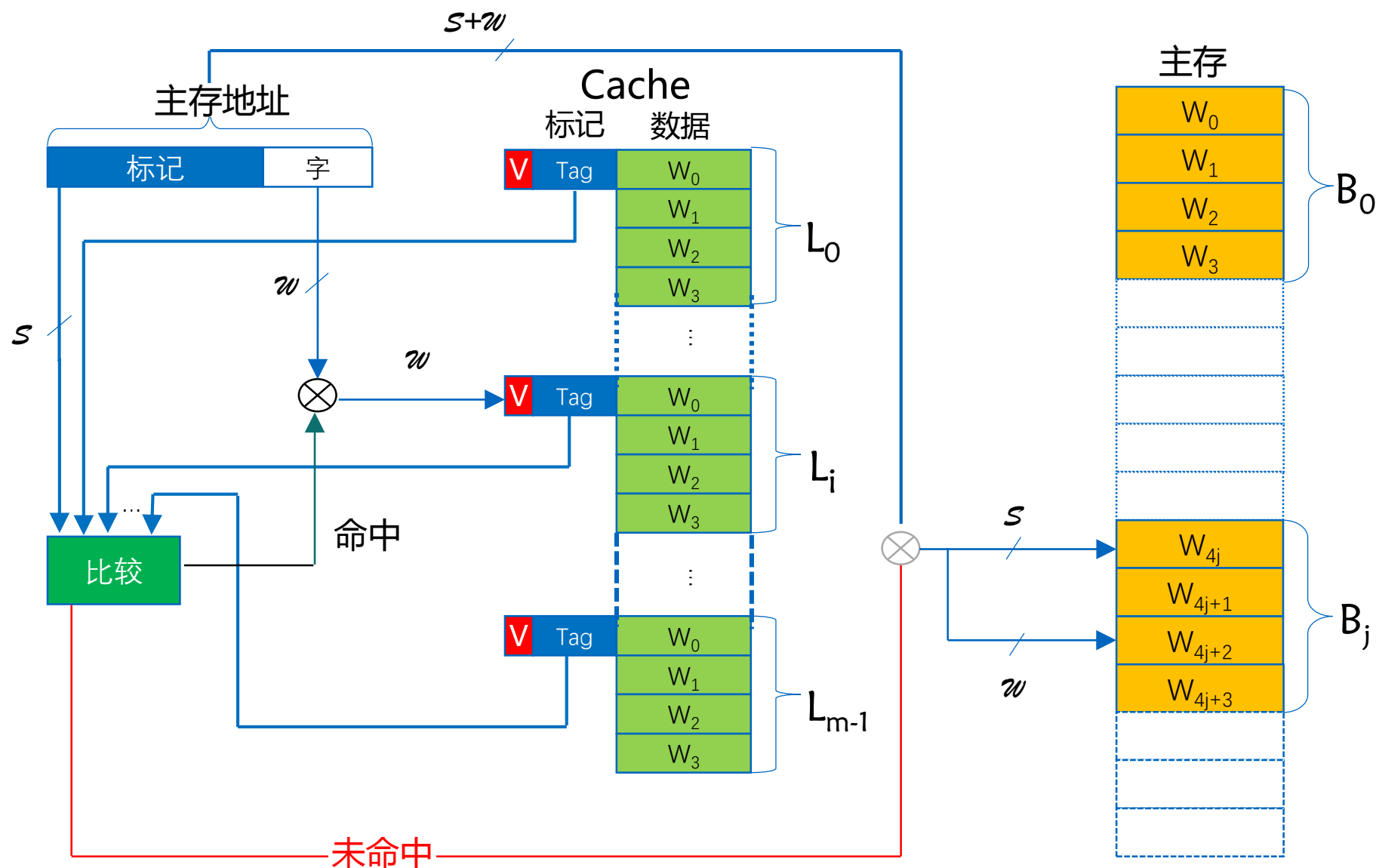
全相联映射的工作原理

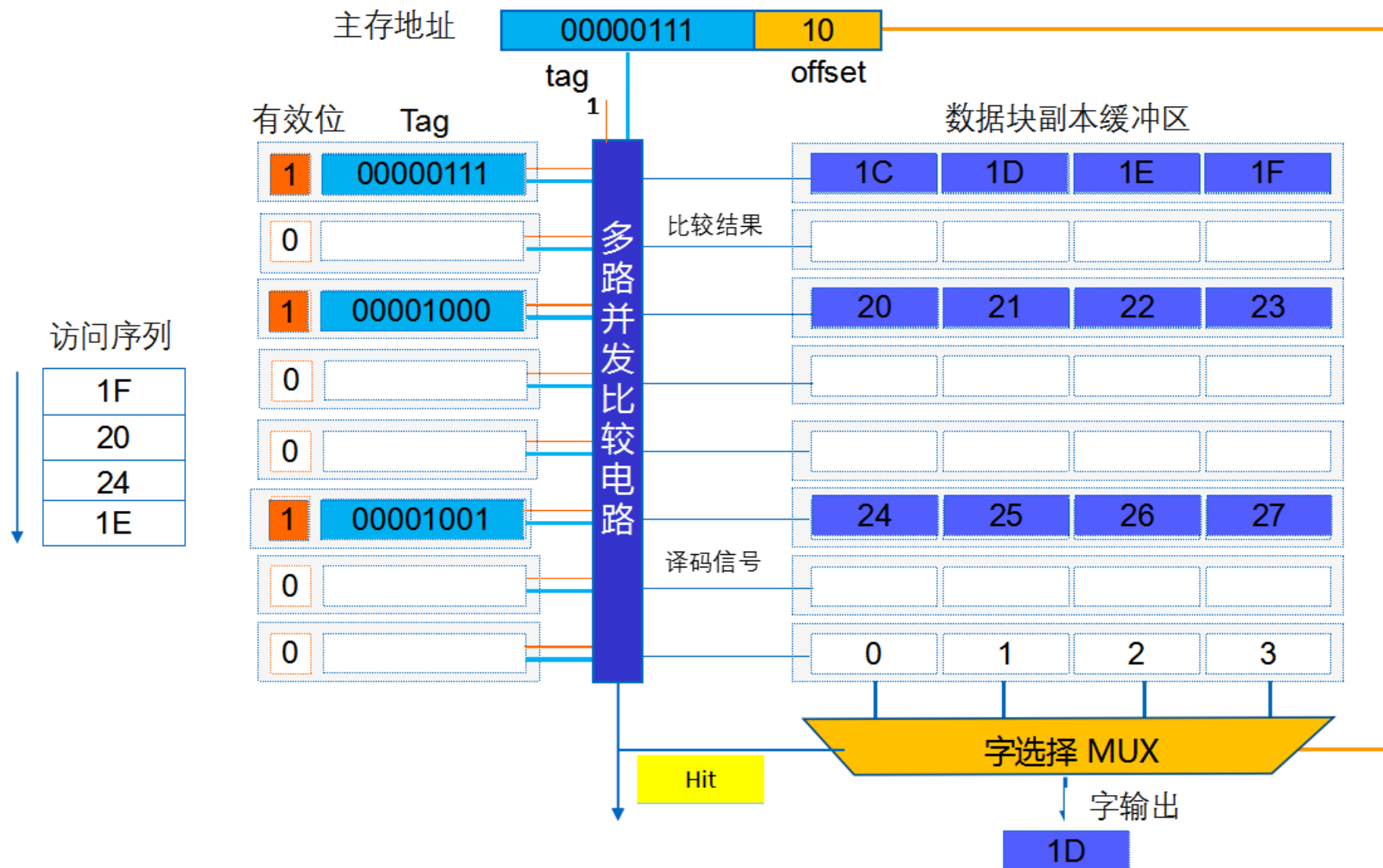


主存

B_0
B_1
...
B_{n-1}
B_n
B_{n+1}
...
B_{2n-1}
B_{2n}
B_{2n+1}
...
B_{3n-1}
...
B_{mn}

- 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个字，则第61个字的主存地址为：
 $00001111 \quad 01$ （块号 块内地址）
- 主存分块后地址就从一维变成二维；
- 映射算法：主存的数据块可映射到Cache任意行，同时将该数据块地址对应行的标记存储体中保存。





全相联映射的特点

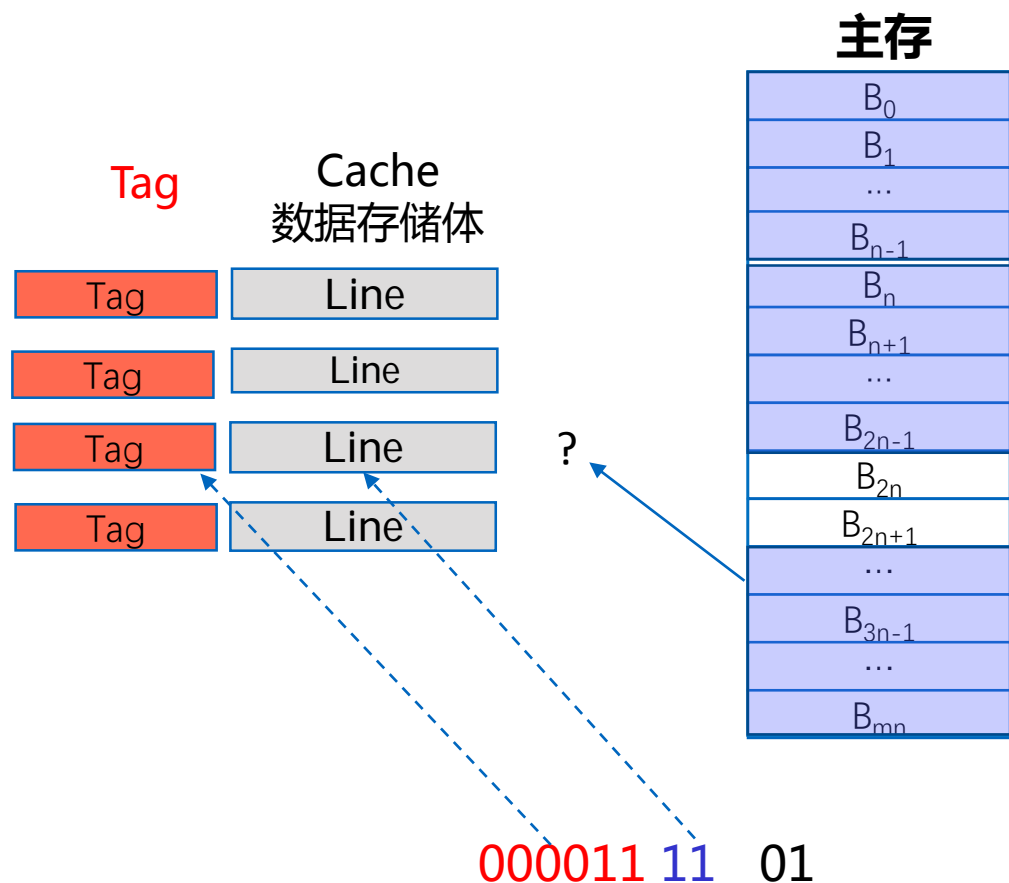
- Cache利用率高
- 块冲突率低
- 淘汰算法复杂

应用场合

- 小容量Cache

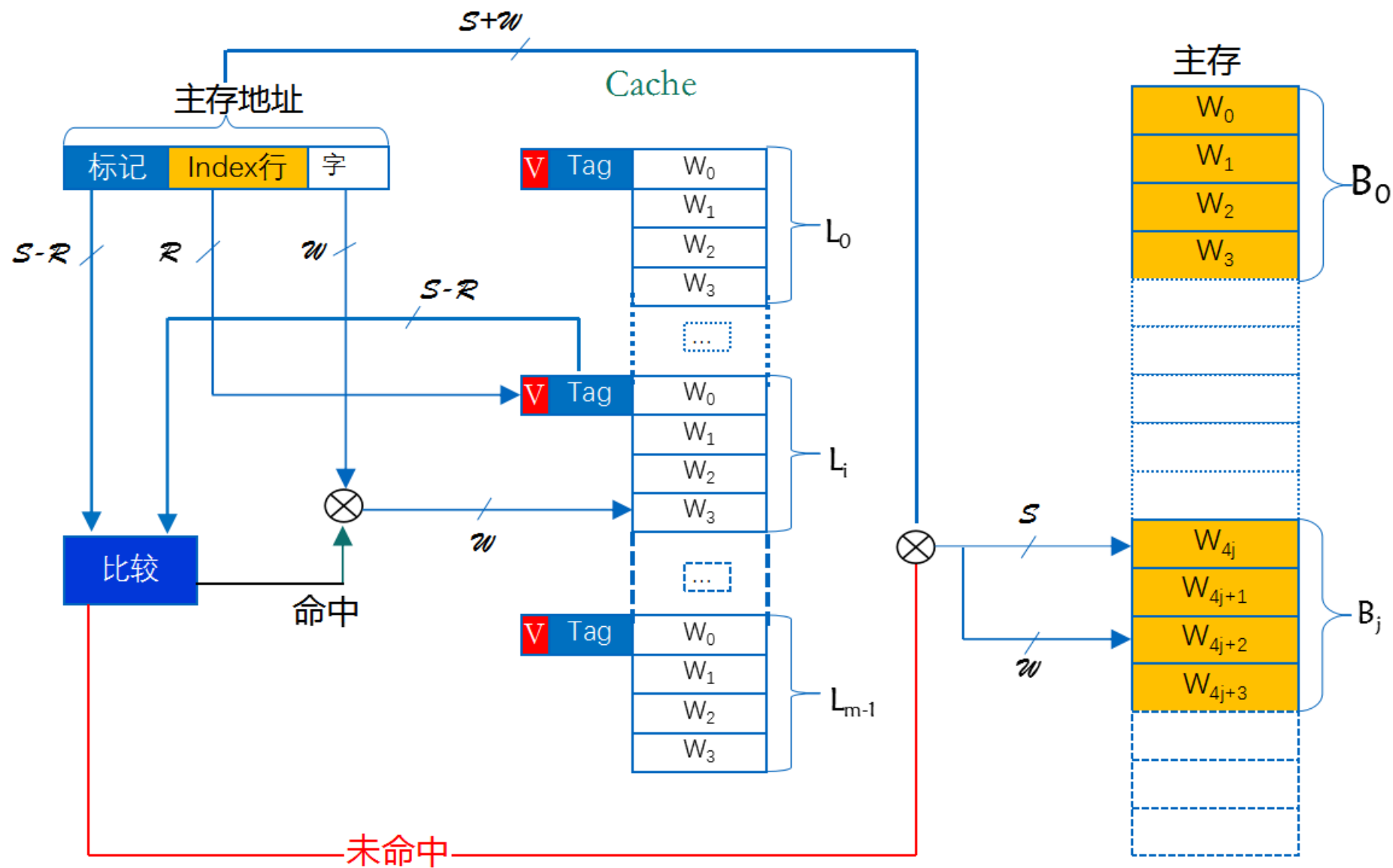
3

直接映射的工作原理



- 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分为4行，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；
- 映射算法：
Cache共n行，主存第j块号映射到Cache的行号为：
$$i = j \bmod n$$

即主存的数据块映射到Cache特定行

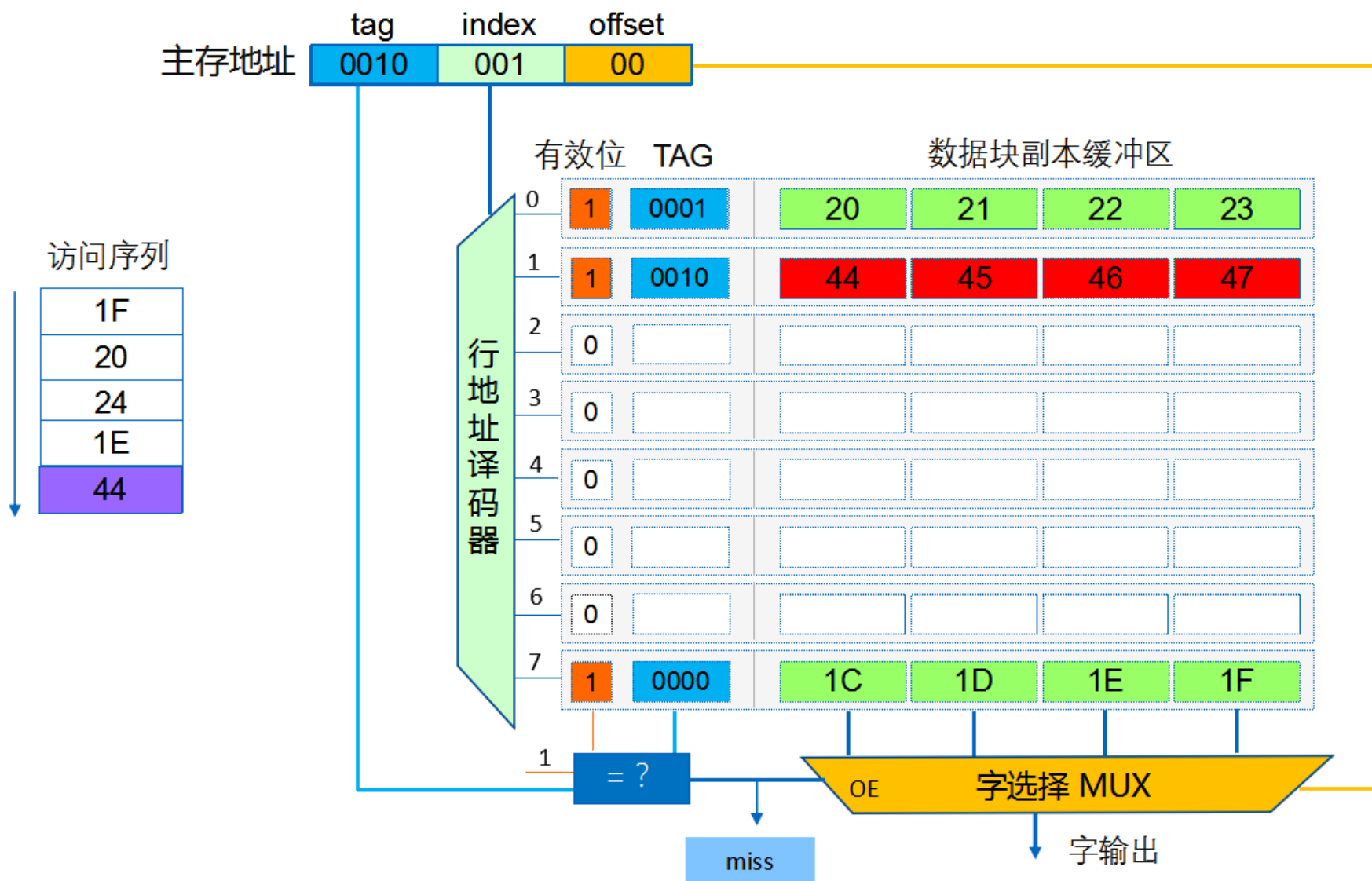


4.9 Cache地址映射与变换方法

3

直接映射的工作原理

工程观视图



全相联映射的特点

- Cache利用率**高**
- 块冲突率**低**
- 淘汰算法**复杂**

应用场合

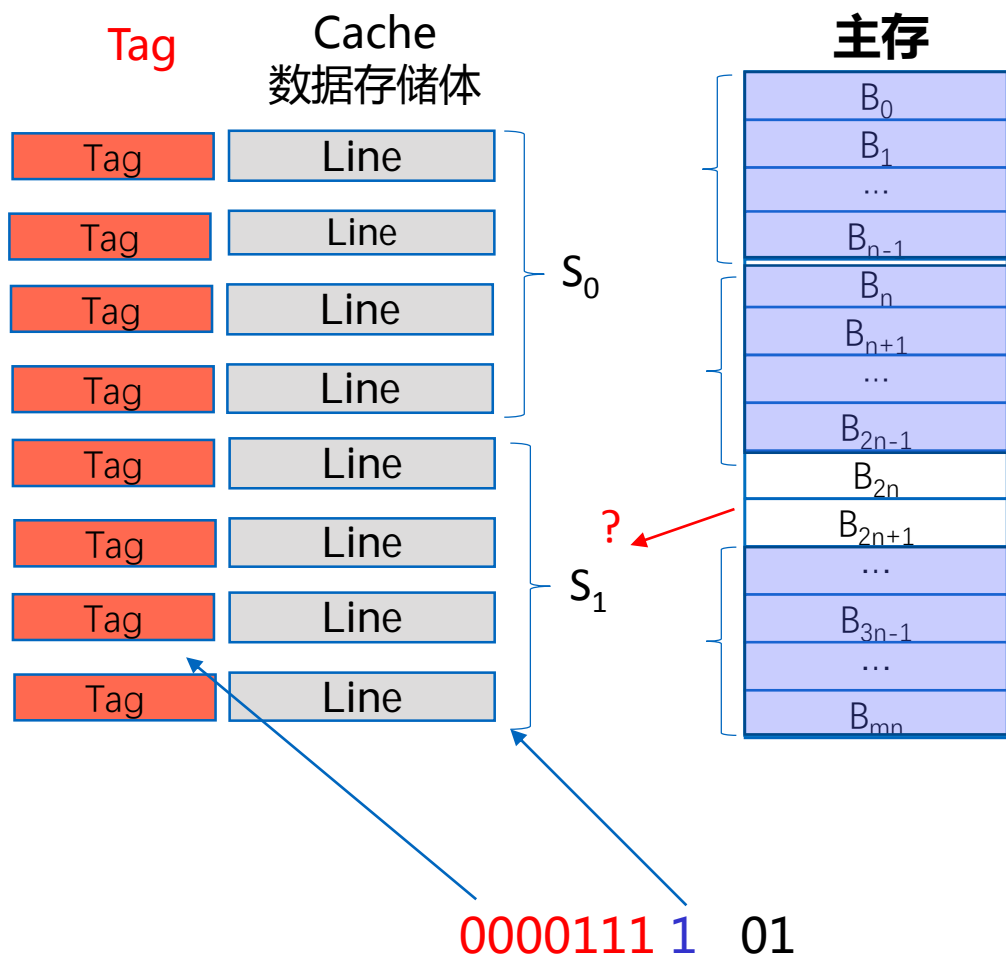
- **小**容量Cache

直接映射的特点

- Cache利用率**低**
- 块冲突率**高**
- 淘汰算法**简单**

应用场合

- **大**容量Cache



- 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4行，第61个字的主存地址为：
0000111 1 01（组号，组内块号，块内地址）
主存地址从一维变成三维；
- 映射算法：
Cache共n组，主存第j块号映射到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

