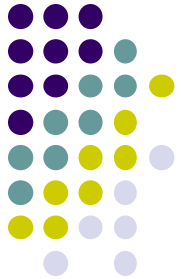


计算机组成与系统结构

第三章 多层次的存储器

吕昕晨

lvxinchen@bupt.edu.cn

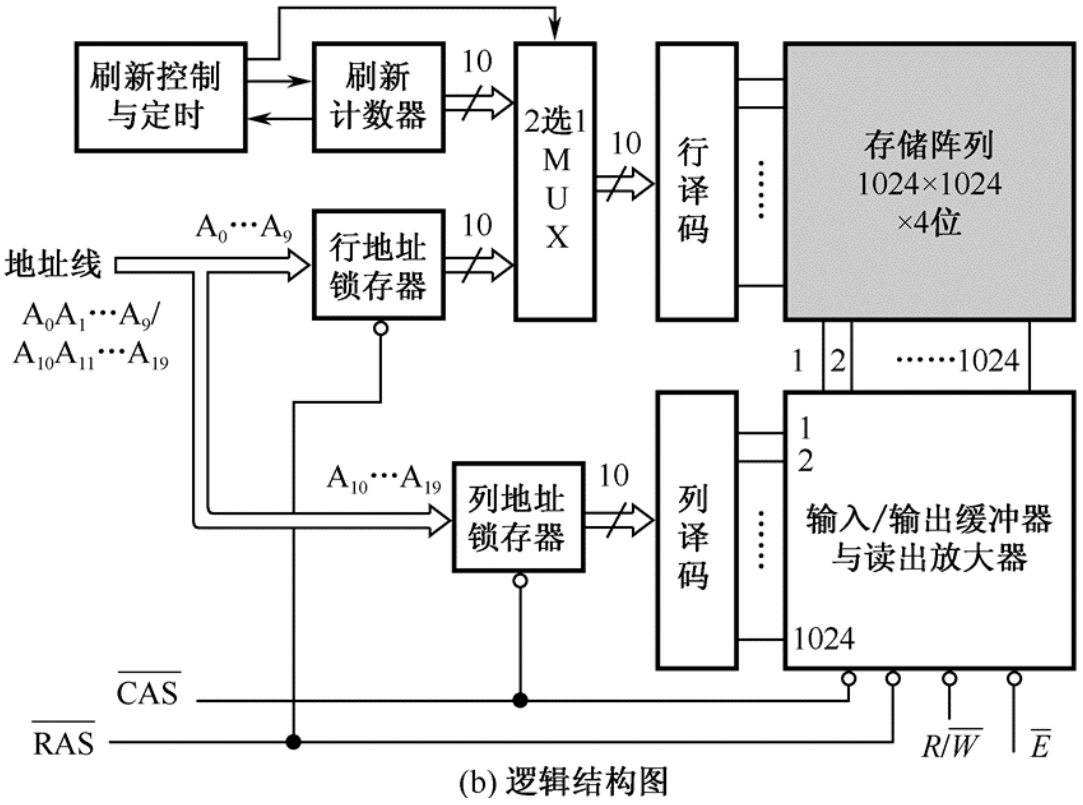
网络空间安全学院

复习

- DRAM芯片结构与刷新方式
- 字长、容量扩展
- 混合扩展习题

DRAM芯片结构

- DRAM芯片结构特点
  - 分时地址输入
  - 行地址　列地址
  - 缓冲器扩展
    - FPM/Cache
    - 局部性原理
- 控制信号
  - E端口
  - RAS信号
  - CAS信号
  - R/W信号
- 刷新方式
  - 按行刷新
  - 定时刷新各行



(b) 逻辑结构图

DRAM刷新周期与方式

- 刷新操作有两种刷新方式：
  - 集中式刷新
    - DRAM的所有行在每一个刷新周期中都被刷新
    - 例如刷新周期为8ms的内存来说，所有行的集中式刷新必须每隔8ms进行一次。为此将8ms时间分为两部分：前一段时间进行正常的读/写操作，后一段时间做为集中刷新操作时间
    - 读写周期0.5us，256*128*8bit芯片，刷新锁死128us
  - 分散式刷新
    - 每一行的刷新插入到正常的读/写周期之中
    - 异步刷新（以刷新周期计算刷新信号）
    - 如DRAM有1024行，如果刷新周期为8ms，则每一行必须每隔8ms÷1024=7.8us进行一次
    - 刷新信号设计7.5us

复习

- DRAM芯片结构与刷新方式
- 字长、容量扩展
- 混合扩展习题

字长位数扩展

- 字长扩展，即数据线扩展
  - 芯片8位，需求32位地址线
  - 多个芯片扩展字长位数
  - d=设计要求/芯片能力
- 设计方法
  - 共用地址线、控制线
  - 芯片数据线各位连接至数据线不同位

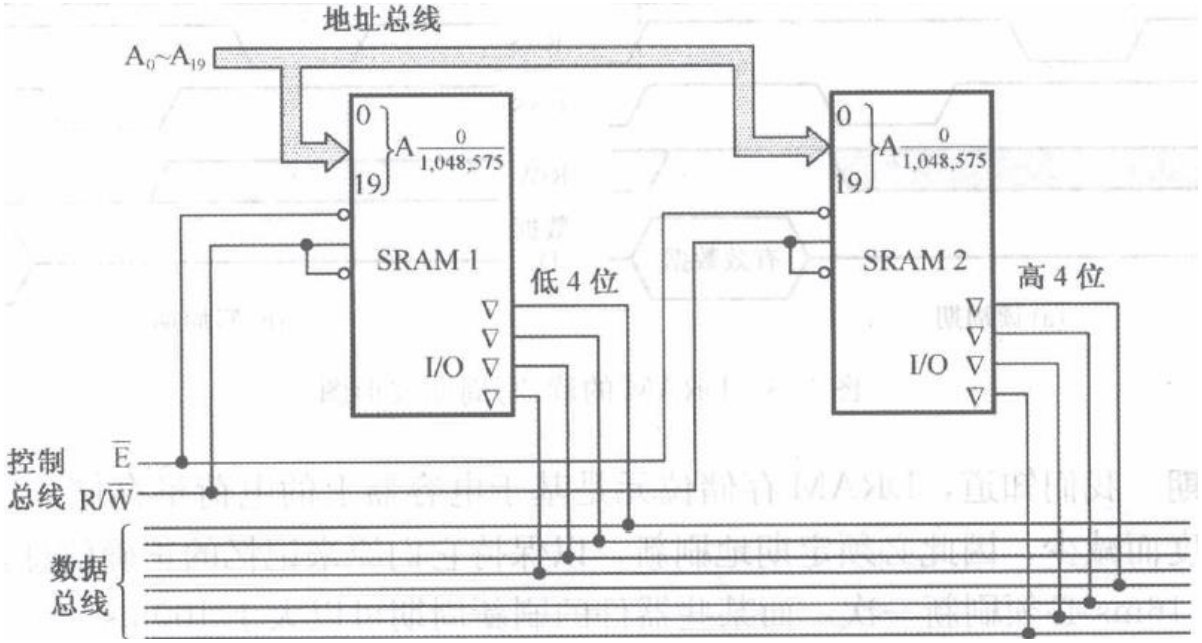
[例2] 利用1M×4位的SRAM芯片，设计一个存储容量为1M×8位的SRAM存储器。

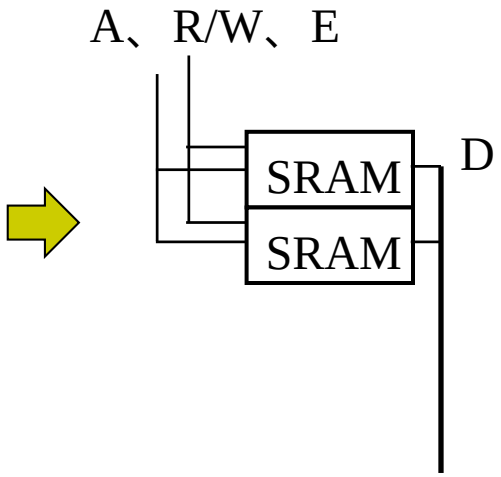
- 位数不足，8位需求 v.s. 4位芯片
- 所需芯片数量=(1M×8)/(1M×4)=2片

字长位数扩展

[例2] 利用1M×4位的SRAM芯片，设计一个存储容量为1M×8位的SRAM存储器。

- 设计方法
  - 共用地址线、控制线
  - 芯片数据线各位连接至数据线不同位（高低4位）





存储容量扩展

- 存储容量扩展，即地址线扩展
  - 芯片1M（20位），需求2M（21位）
  - 多个芯片扩展地址线
  - d=设计要求/芯片能力
- 设计方法
  - 共用部分地址线（低20位）、控制线、数据线
  - 高位地址线用于生成芯片片选信号
    - 1位扩展（非门），多位扩展（译码器）

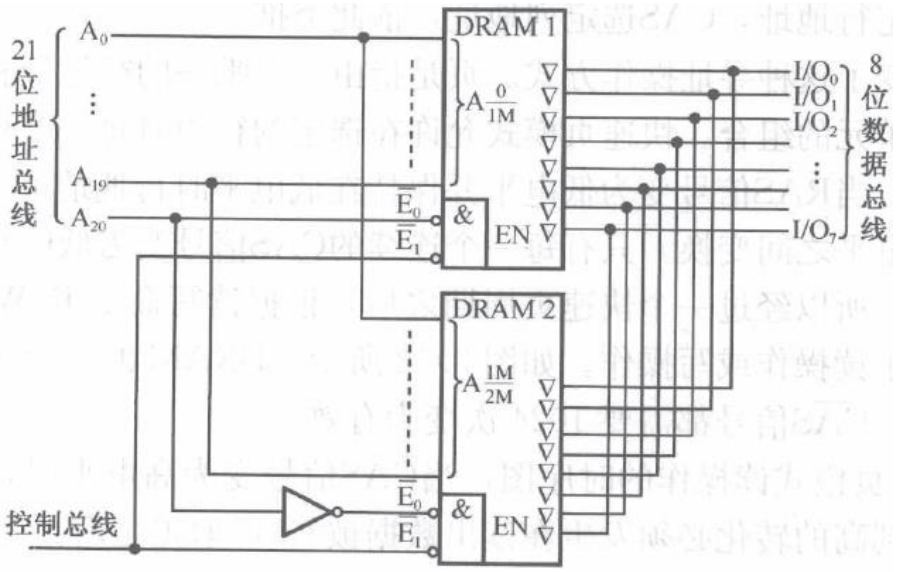
[例3]利用1M×8位的DRAM芯片设计2M×8位的DRAM存储器

- 容量不足：2M需求 v.s. 1M芯片
- 所需芯片数d=（2M×8）/（1M×8）=2(片)

存储容量扩展

[例3]利用1M×8位的DRAM芯片设计2M×8位的DRAM存储器

- 设计方法
  - 共用控制线、数据线
  - 共用低20位地址线；A0~A1同时连接到2片DRAM
  - 高位地址线（A20）用于生成芯片片选信号
    - 1位扩展（非门）



复习

- DRAM芯片结构与刷新方式
- 字长、容量扩展
- 混合扩展习题

习题3-1

- 设有一个具有20位地址和32位字长的存储器

1）该存储器能存储多少个字节的信息

2^{20}（地址线）*32（字长）/8（Byte）

=4 MB

2）如果存储器由512K*8位SRAM芯片组成，需要多少片

字长扩展 32/8=4

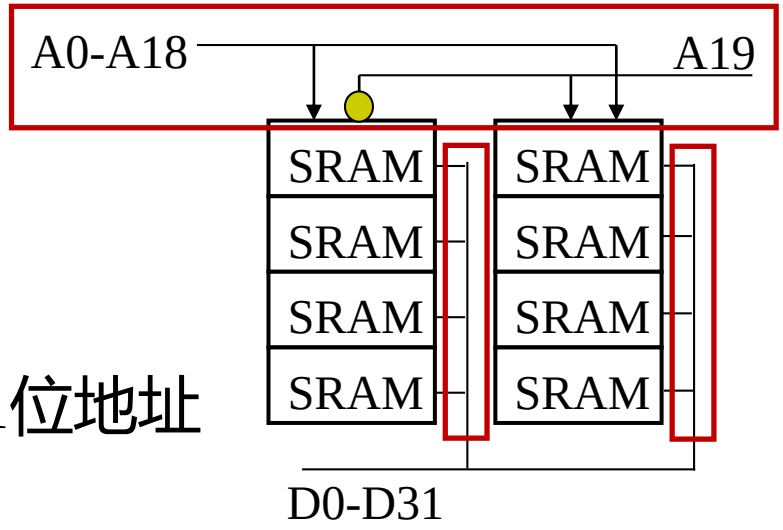
容量扩展 1M/512K=2

需要4*2=8片

3）需要多少位地址作芯片选择

地址选择（容量扩展）2倍　1位地址

4）画出逻辑框图



习题3-4

- 由128K*8位的DRAM芯片构成1024K*32位存储器

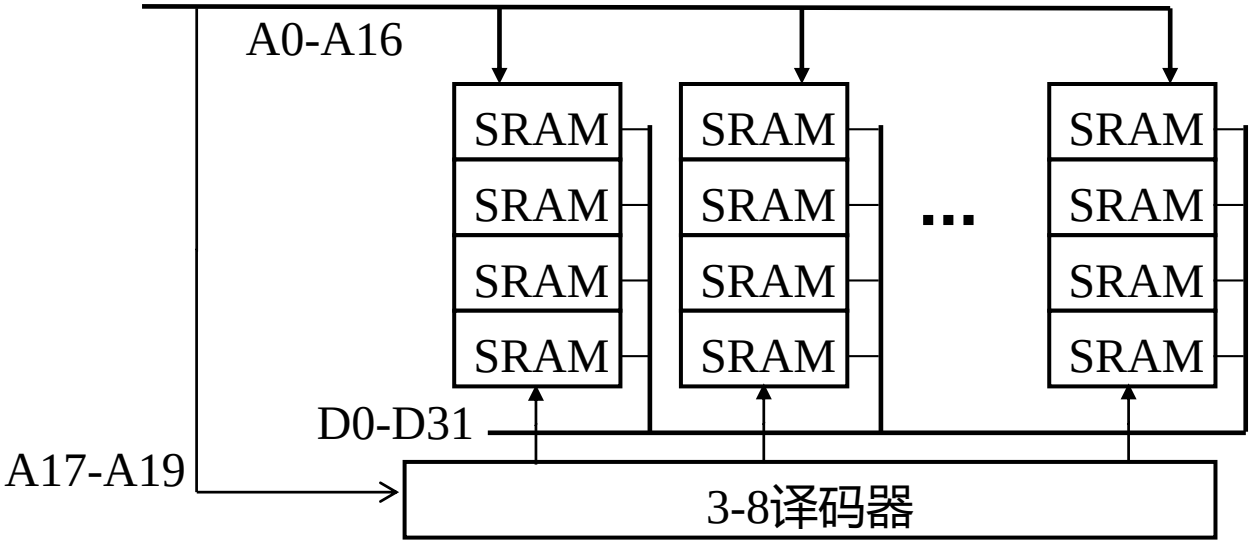
1）总共需要多少DRAM芯片

字长扩展 32/8=4 （4个DRAM组成32位）

容量扩展 1M/128K=8 （17位 20位，3-8译码器）

需要4*8=32片

2）画出存储器逻辑框图



习题3-4

- 由128K*8位的DRAM芯片构成1024K*32位存储器

3）存储器为读写周期0.5us，CPU在1us内至少访存一次，采用何种刷新方式?

假设存储器芯片为512*256*8bit（17位=9位+8位）

集中式刷新：0.5us*512=256us >> 1us，不可行

采用分散式刷新

4）刷新周期为8ms，刷新信号周期为?

刷新信号周期=刷新周期/行数

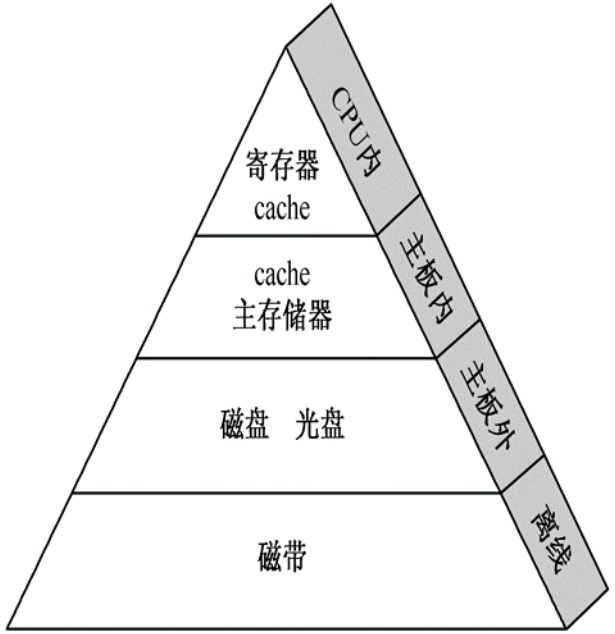
=8ms/512

=15.625us

=15.5 us （以读写周期向下取整）

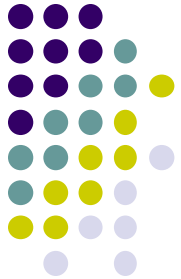
本周教学安排

- 直播内容
  - Cache存储器
    - 基本原理
    - 地址映射
    - 替换策略
  - 虚拟存储器
    - 概念（与Cache对比）
- 录播内容
  - 只读存储器（ROM）
  - 并行存储器
  - 虚拟存储器



Cache-主存-辅存
多级存储结构

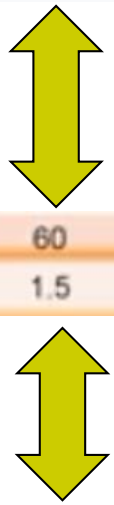
第三章 多层次的存储器



- Cache基本原理
- Cache读写操作
- Cache地址映射
- Cache替换策略与其它
- 虚拟存储器概念

CPU、SRAM与DRAM速率

		1980	1990	2000	2010	2010:1980
CPU	Name	8080	386	Pentium II	Core i7	/
	Clock rate(MHz)	1	20	600	2,500	2,500
	Cycle time(ns)	1,000	50	1.6	0.4	2,500
	Cores	1	1	1	4	4
	Effective Cycle time(ns)	1,000	50	1.6	0.1	10,000
SRAM	$/MB	19,200	320	100	60	320
	access time(ns)	300	35	3	1.5	200
DRAM	$/MB	8,000	100	1	0.06	130,000
	access time(ns)	375	100	60	40	9
	typical size(MB)	0.064	4	64	8,000	125,000

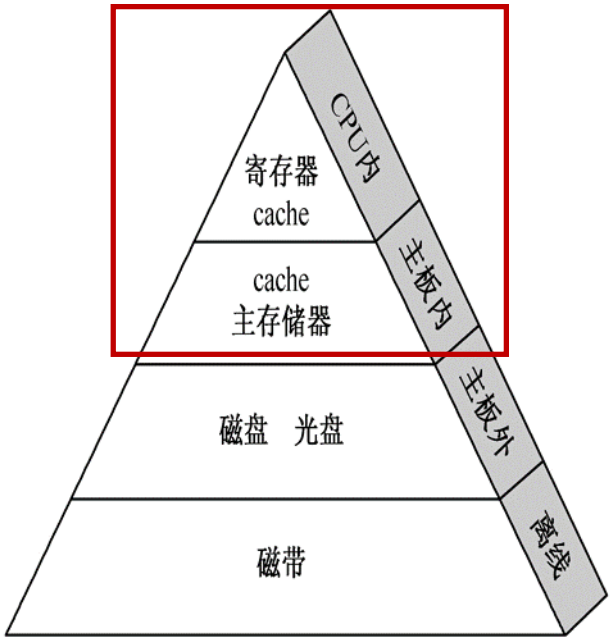


3~4倍

Cache整体目标

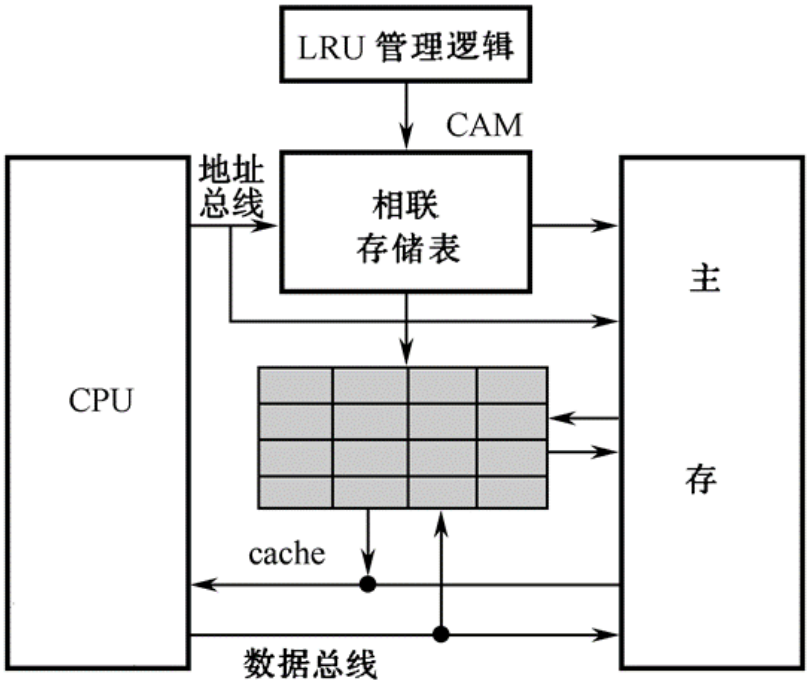
解决CPU和主存速度不匹配问题

- 一般采用高速的SRAM构成。
- 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



Cache逻辑结构与功能

- 组成结构
  - Cache存储表
  - 相联存储表
  - 控制器
- 功能
  - 地址映射
  - 替换策略
  - 写一致性
  - 性能评价



Cache基本原理（1）

- Cache基本原理——程序局部性原理
  - 经验性结论
  - 程序在时间与空间上都表现局部性
- 时间局部性
  - 最近被访问存储单元很快还会被访问
- 空间局部性
  - 正在被访问存储单元附近单元很快会被访问

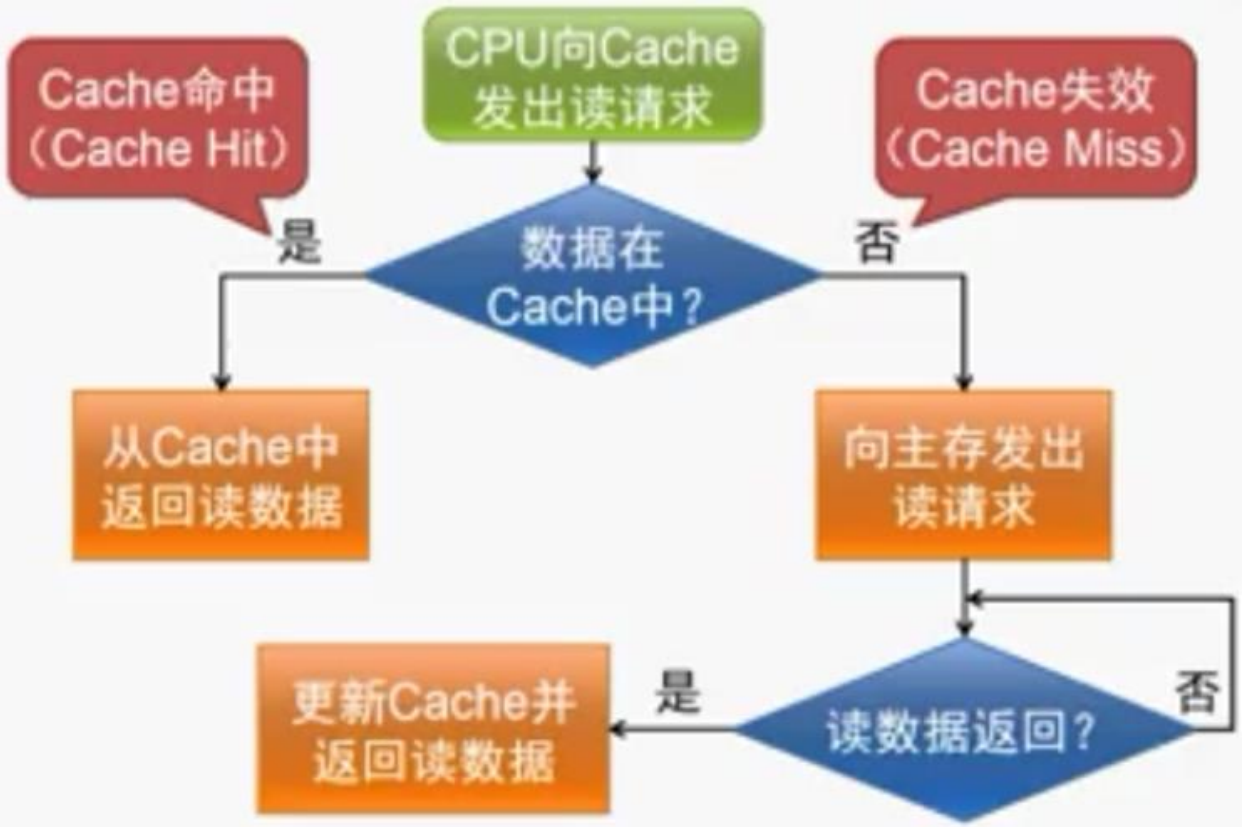
```
for (i=0; i<1000; i++)
   for (j=0; j<200; j++)
      sum += a[i][j];
```

典型程序段

Cache基本原理（2）

- Cache对空间局部性利用
  - 从主存中取回数据时，同时取回与对应位置相邻的主存单元数据
  - 以数据块为单位进行数据交换（行）
- Cache对时间局部性利用
  - 优先保存近期频繁被访问主存单元数据
  - 替换策略：LRU、LFU

Cache访问流程


CPU向Cache
发出读请求
Cache命中
（Cache Hit）
Cache失效
（Cache Miss）
是
数据在
Cache中？
否
从Cache中
返回读数据
向主存发出
读请求
更新Cache并
返回读数据
是
读数据返回？
否

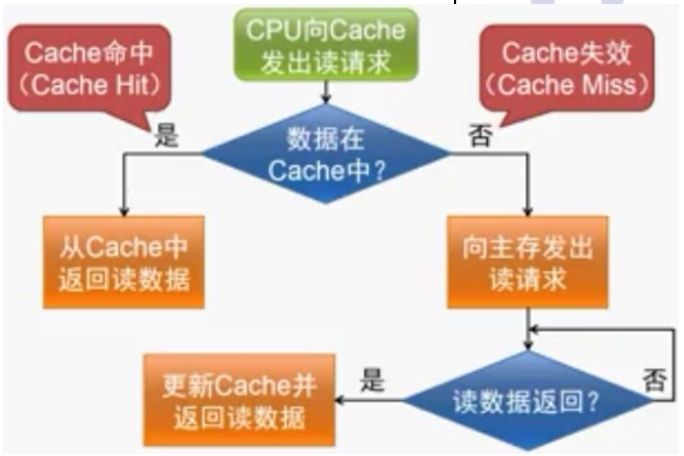
Cache命中率

命中率 $$h = \frac{N_e}{N_e + N_m}$$

平均访问时间 $$t_a = ht_c + (1-h)t_m$$

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

Cache与内存的速度比 $$r = t_m / t_c$$



Cache命中率与效率

- 访存时间对比
  - CPU v.s. Cache：3倍
  - CPU v.s. 内存：100倍
- 平均访存时间
  - 命中率 h=95%

    $3 * 0.95 + 0.05 * 100 = 7.85$
  - 命中率 h=99%

    $3 * 0.99 + 0.01 * 100 = 3.97$
- 命中率提高4%，性能提升1倍
- cache的命中率应尽可能接近于1

		1980	1990	2000	2010	2010:1980
CPU	Name	8080	386	Pentium II	Core i7	/
	Clock rate(MHz)	1	20	600	2,500	2,500
	Cycle time(ns)	1,000	50	1.6	0.4	2,500
	Cores	1	1	1	4	4
	Effective Cycle time(ns)	1,000	50	1.6	0.1	10,000
SRAM	$/MB	19,200	320	100	60	320
	access time(ns)	300	35	3	1.5	200
DRAM	$/MB	8,000	100	1	0.06	130,000
	access time(ns)	375	100	60	40	9
	typical size(MB)	0.064	4	64	8,000	125,000

单选题 1分

设置

此题未设置答案，请点击右侧设置按钮

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

1）Cache命中率?

A 0.8

B 0.85

C 0.9

D 0.95

$$h = \frac{N_e}{N_e + N_m}$$

h=1900/(1900+100)

=0.95

提交

单选题 1分

设置

此题未设置答案，请点击右侧设置按钮

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

2）平均访存时间?

A 55

B 60

C 65

D 70

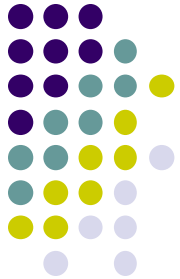
$$t_a = ht_c + (1-h)t_m$$

ta=0.95*50+0.05*250

=60

提交

第三章 多层次的存储器



- Cache基本原理
- Cache读写操作
- Cache地址映射
- Cache替换策略与其它
- 虚拟存储器概念

Cache主体——SRAM矩阵

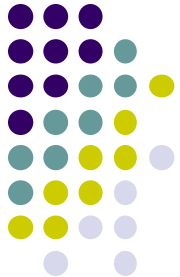
- 组织方式
  - 按行组织
  - 行数：16行
  - 列数：数据16字节+有效位+标签

Cache读操作

- 读操作程序示例
  - MOV AL, [2011H]
  - MOV BL, [4011H]
  - MOV DL, [401FH]

1. 未命中，读2010H，分配表项1
2. 未命中，读401H，替换表项1
3. 命中，读401H

Cache写操作策略



- 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。可选用如下三种写操作策略
  - 写回法（效率高、不一致隐患）
    - 命中时，只修改Cache内容
    - 替换时，写回至主存（判断是否修改）
  - 全写法（效率低、控制逻辑简单）
    - 无论是否命中，同时修改Cache与内存
    - 不命中时，取回/不取回至Cache
  - 写一次法（混合方法）
    - 结合写回与全写法，与写回法基本一致，第一次命中时采用全写法
    - 奔腾CPU策略，全部Cache的一致性

单选题 1分

设置

此题未设置答案，请点击右侧设置按钮

有如下程序段，Cache采用写回法，16*16字节

MOV [2011H], AL

MOV [2011H], BL （AL ≠ BL）

MOV DL, [4011H]

执行第3条语句，Cache的操作包括

A 写[2011H]单元，读[4011H]单元至表项1

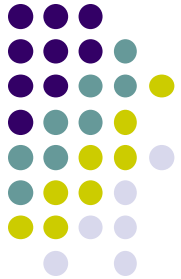
B 写[2011H]单元，替换[4010H]数据块至表项1

C 替换[4010H]数据块至表项1

D 写[2011H]单元，替换[4011H]单元至表项40H

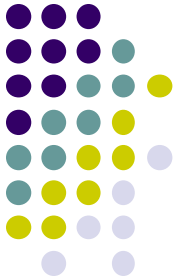
提交

第三章 多层次的存储器



- Cache基本原理
- Cache读写操作
- Cache地址映射
- Cache替换策略与其它
- 虚拟存储器概念

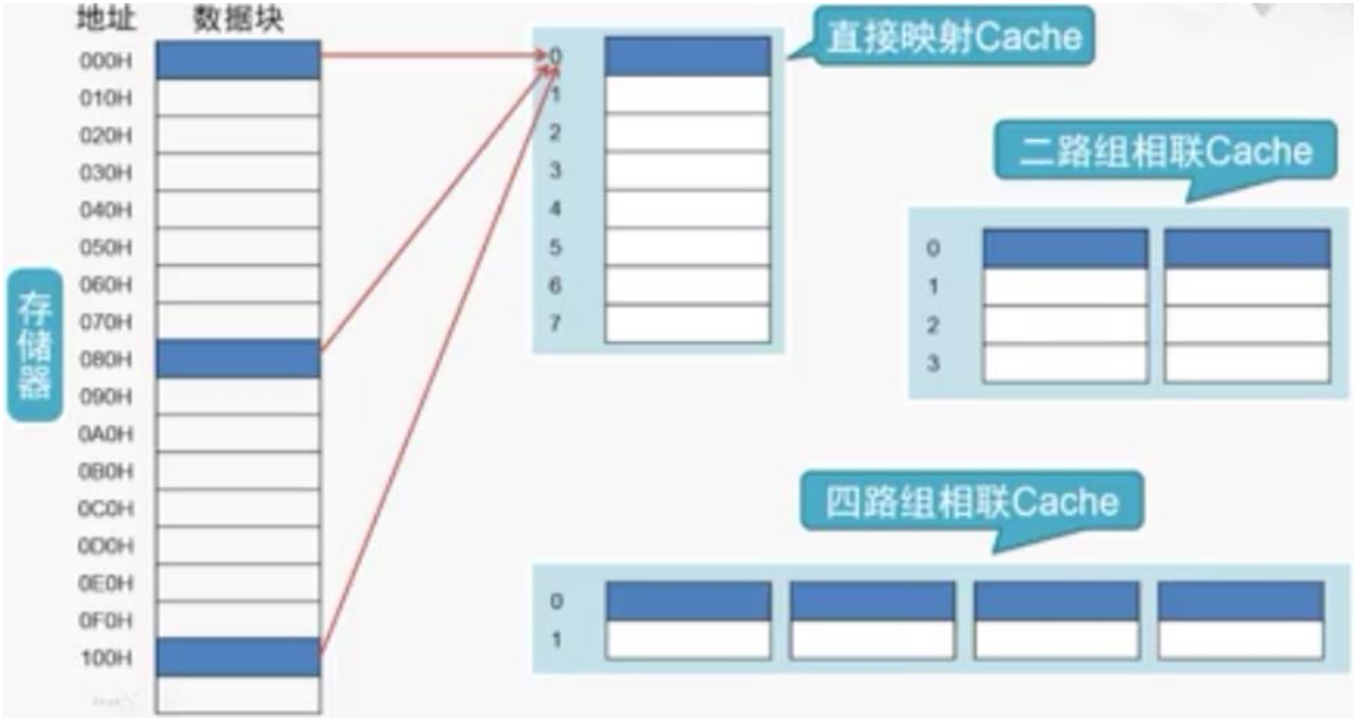
Cache地址映射设计目标



- Cache是否命中，依赖于Cache是否冲突
  - HASH方式设计
- 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
- 常用地址映射
  - 直接映射、组相联、全相联

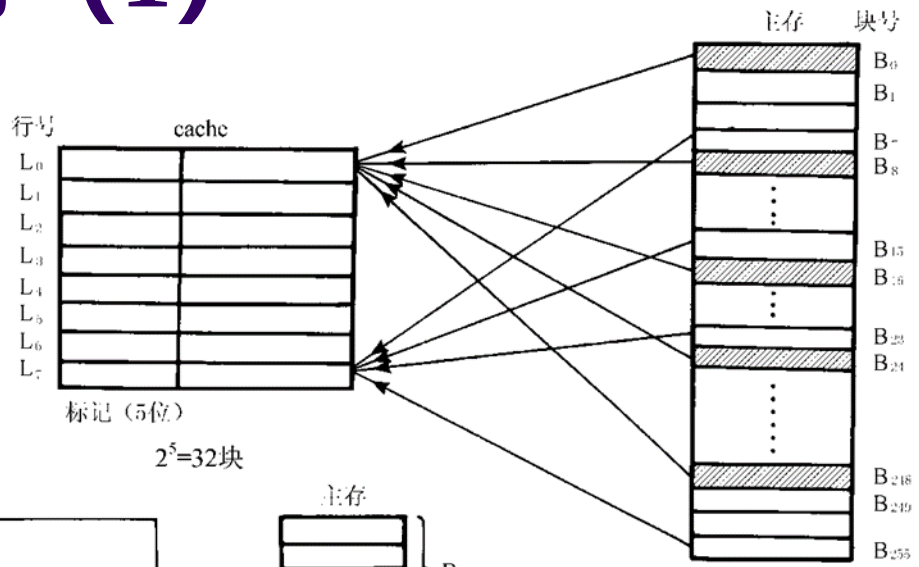
Cache地址映射概述

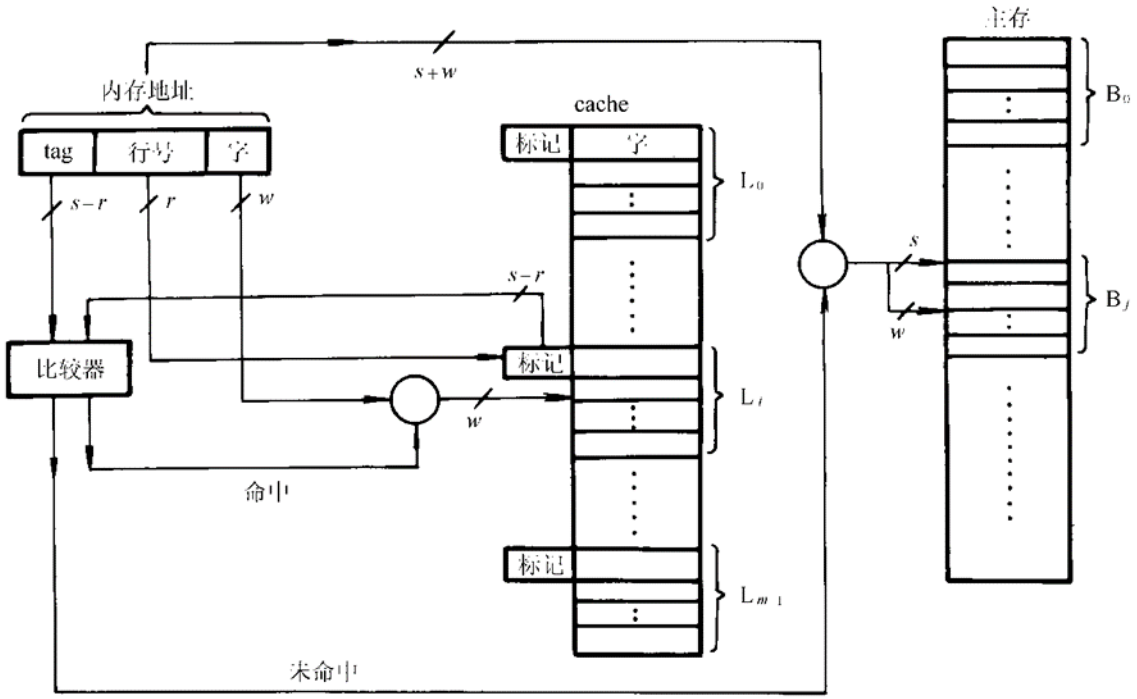
- 直接映射：末位HASH（8*1数组）
- 二路组相联：4*2数组
- 全相联：1*8数组
- 组数越多，命中时间越高



直接地址映射（1）

地址映射方式，
根据行号进行标记

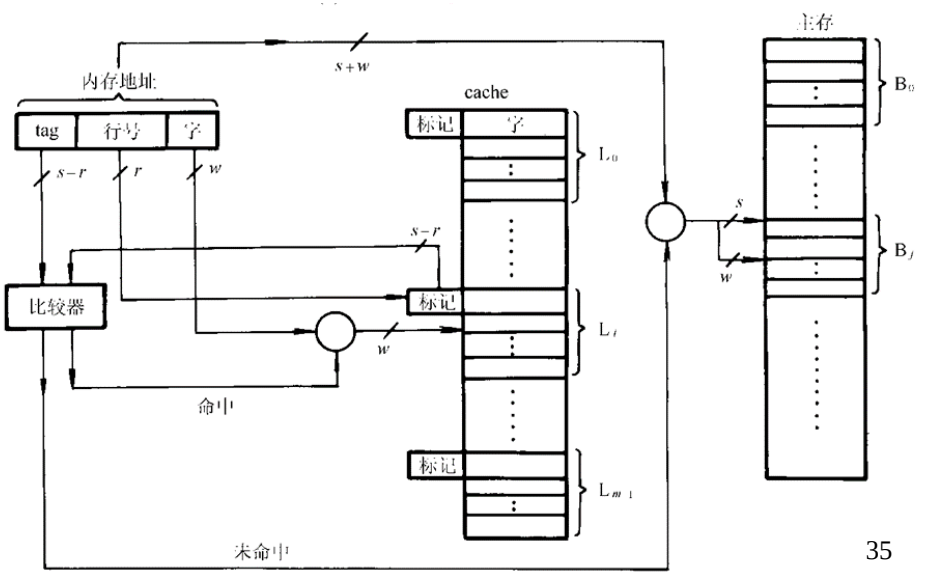




是否命中
[4011H]，16位字长
块号401H
寻找表项1
标记是否为40H

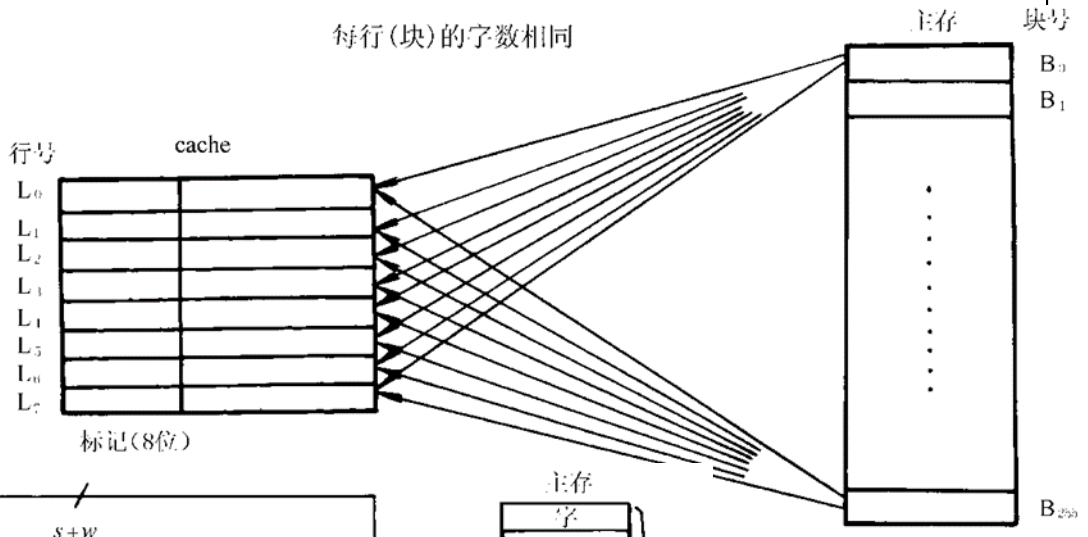
直接地址映射（2）

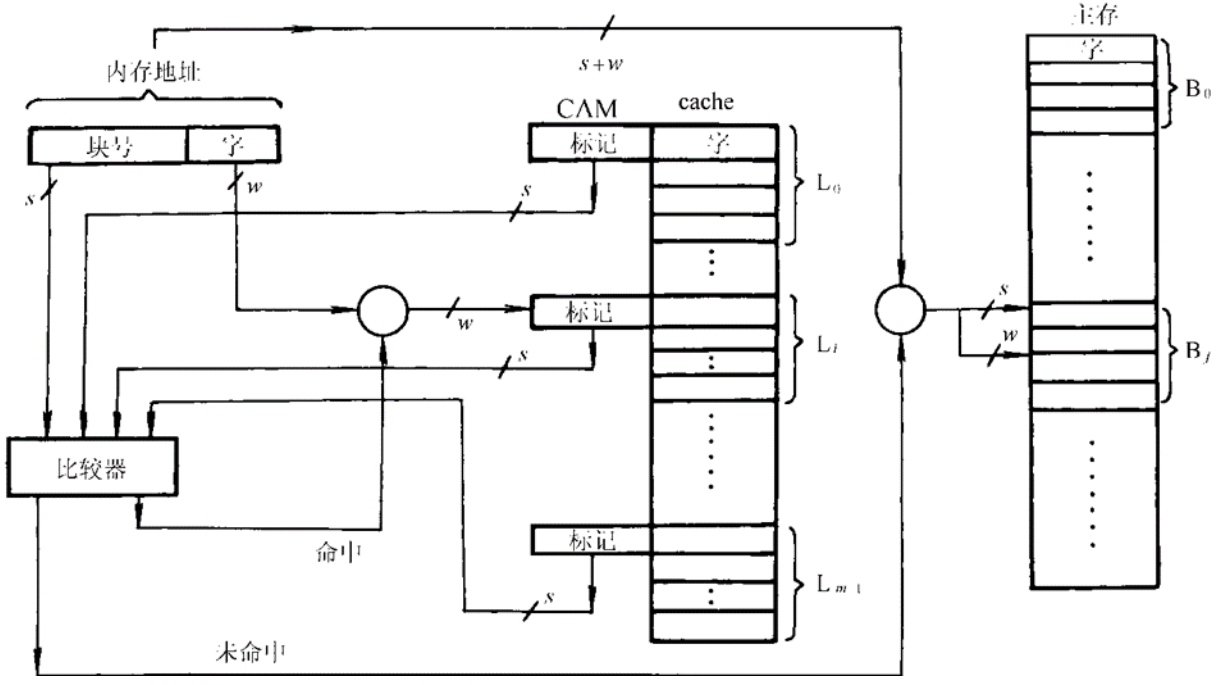
- 特点
  - 优点：比较电路少m倍线路，所以硬件实现简单，Cache地址为主存地址的低几位，不需变换。
  - 缺点：冲突概率高（抖动）
- 应用场合
  - 适合大容量Cache
  - 减少冲突概率



全相联地址映射（1）

地址映射方式，
与内存位置无关

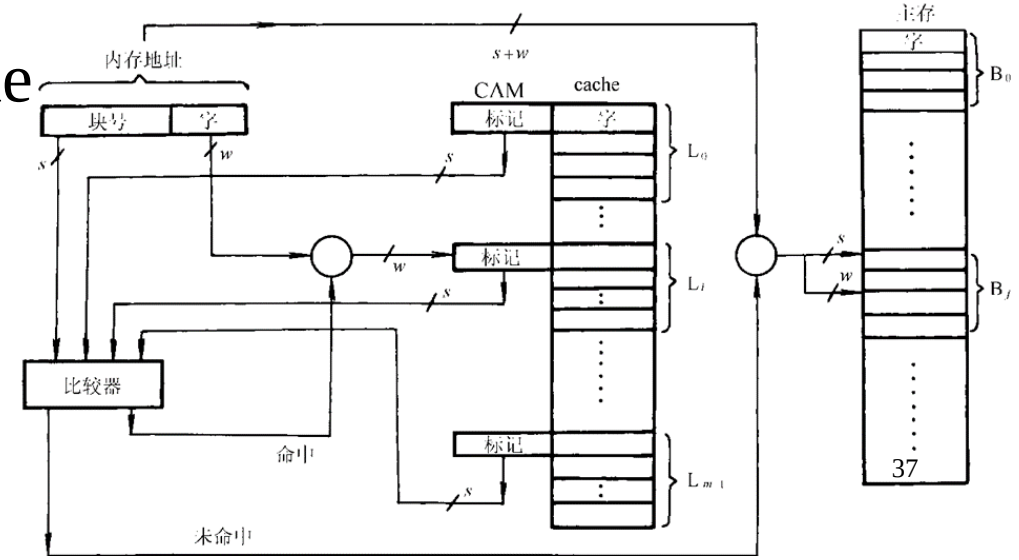




比较块号是否命中
[4011H]，16位字长
块号401H
字号01H

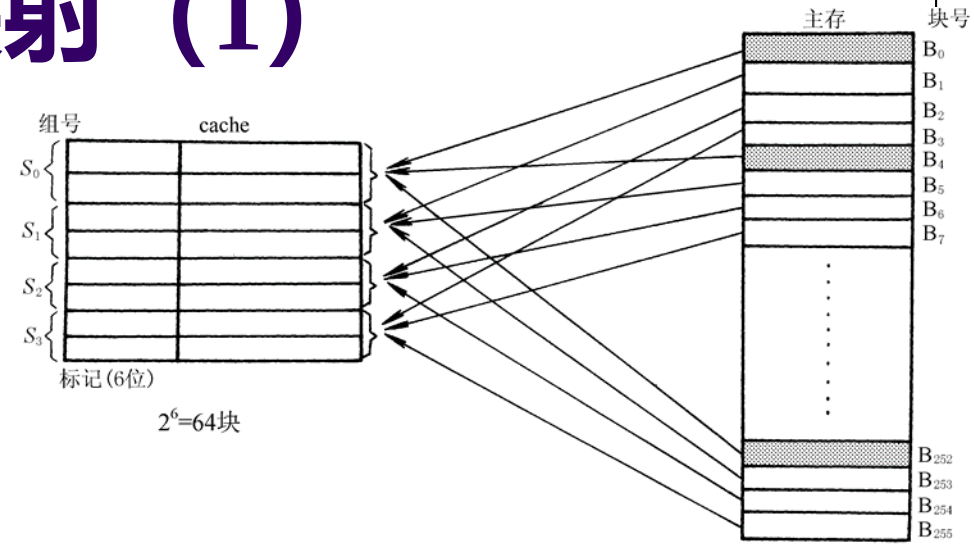
全相联地址映射（2）

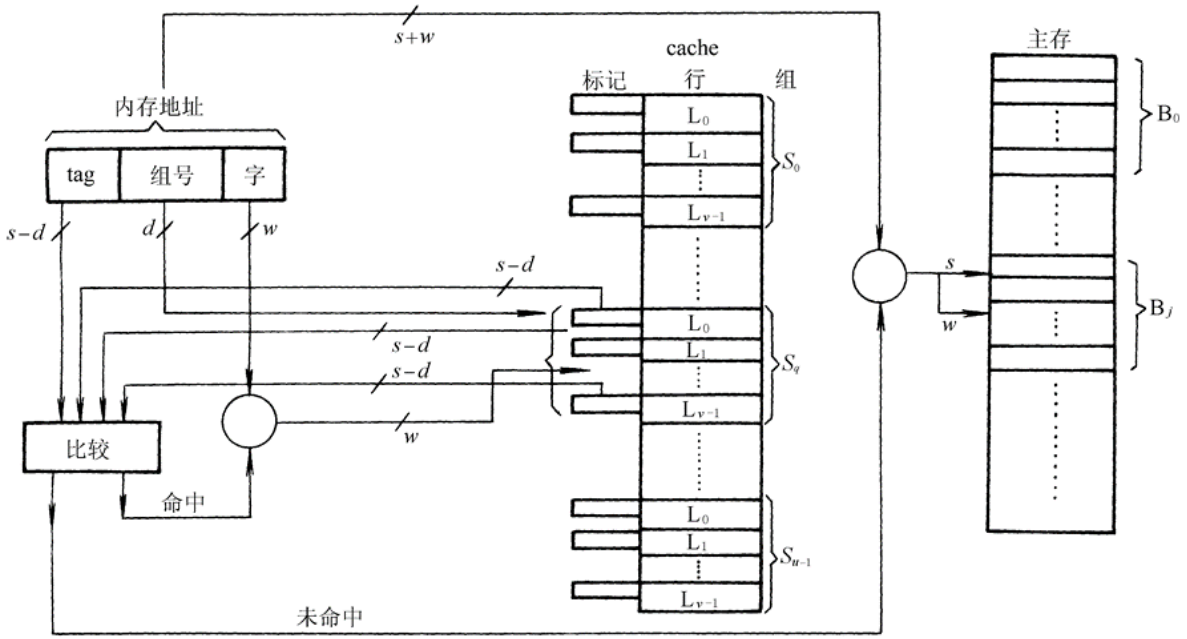
- 特点：
  - 优点：冲突概率小，Cache的利用高。
  - 缺点：比较器难实现，需要一个访问速度很快代价高的相联存储器
- 应用场合：
  - 适用于小容量的Cache



组相联地址映射（1）

地址映射方式，
根据行号进行标记
二路组相联
8*2数组

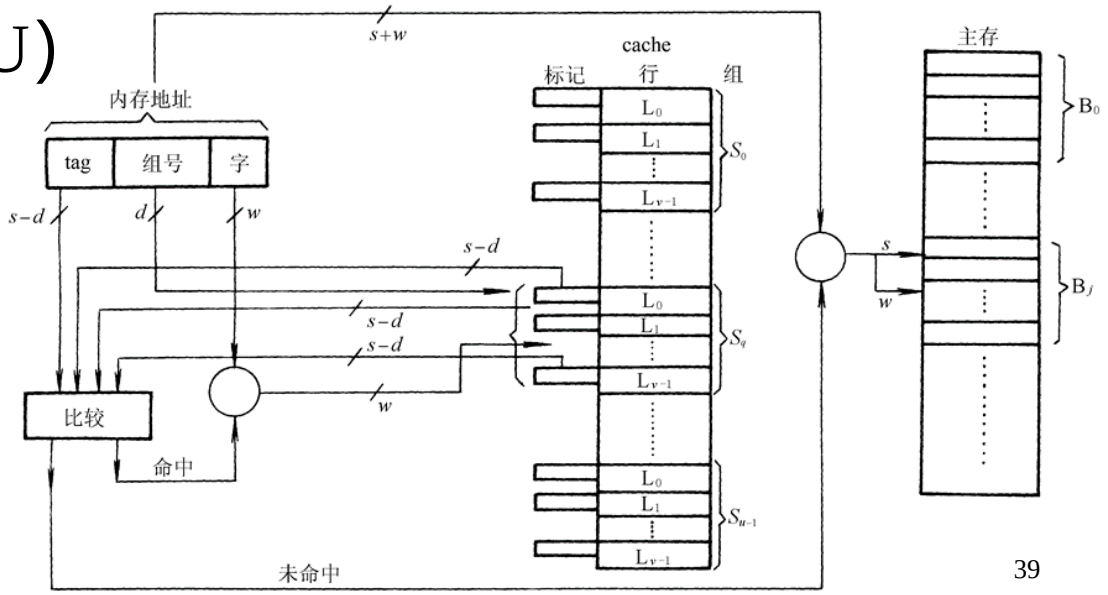




是否命中
比较两个表项内容
[4011H] mod 8
组1内容
表项3/4

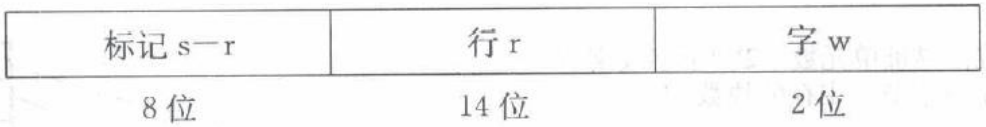
组相联地址映射（2）

- 特点
  - 优点：比全相联容易实现，冲突低
  - 缺点：随组数增加更为复杂
- 组成结构
  - U*V数组（mod U）
  - V一般为2的幂
  - V=1，直接相连
  - U=1，全相联



地址映射例题（1）

【例 7】　直接映射方式的内存地址格式如下所示：



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 $= (1011\ 1011\ 1011\ 10)_2 = (2EEE)_{16}$

字 w $= (11)_2 = (3)_{16}$

地址映射例题（2）

例8：一个组相联cache由64个行组成，每组4行。
主存包含4K个块，每块128字。请表示内存地址的格式

块大小 = 行大小 = $128 = 2^7$　　∴ w = 7

每组的行数k = 4

组号 = $64/4=16=2^d$　　∴ d = 4

主存总体块数：$4K= 2^{12}$　　∴ s = 12

标记大小(s-d)位 = 12-4 = 8位

8位	4位	7位
标记s-d	组号d	字号w

填空题 2分

设置

此题未设置答案，请点击右侧设置按钮

主存容量1MB，字长8位，块大小16B，Cache容量64KB，采用全相联映射，给出[F0010H]对应标记为 [填空1] 、字号为 [填空2] 。

主存容量1MB，块大小16B ∴ 字号w = 4

块数=1MB/16B= 2^{16} ∴ 标记s = 16

Cache容量：64KB/16B=2^{12}

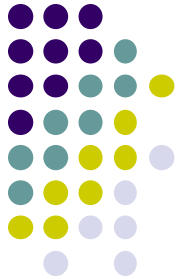
F0010H= 1111 0000 0000 0001 0000

标记 字号

正常使用填空题需3.0以上版本雨课堂

作答

第三章 多层次的存储器



- Cache基本原理
- Cache读写操作
- Cache地址映射
- Cache替换策略与其它
- 虚拟存储器概念

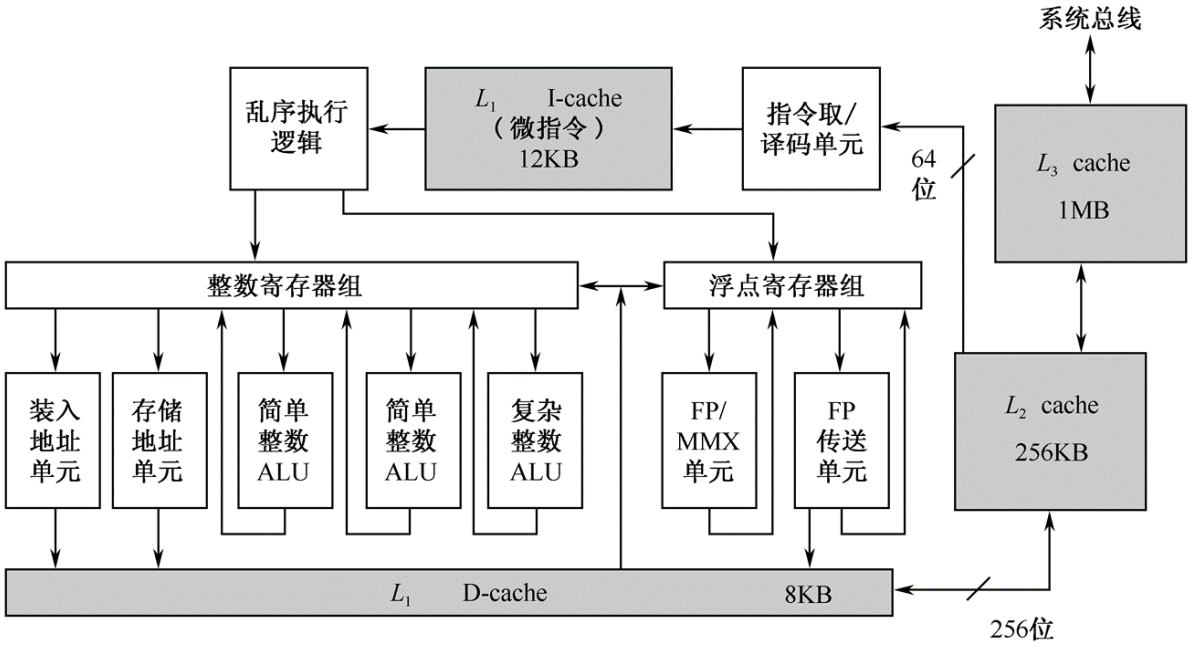
Cache替换策略

- LFU（最不经常使用，Least Frequently Used）
  - 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，Least Recently Used）
  - 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
- 随机替换
  - 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
  - 随机替换策略的功效只是稍逊于前两种策略

Pentium 4的Cache组织

主要包括四个部分：

- 取指/译码单元：顺序从L2cache中取程序指令，将它们译成一系列的微指令，并存入L1指令cache中。
- 乱序执行逻辑：依据数据相关性和资源可用性，调度微指令的执行，因而微指令可按不同于所取机器指令流的顺序被调度执行。

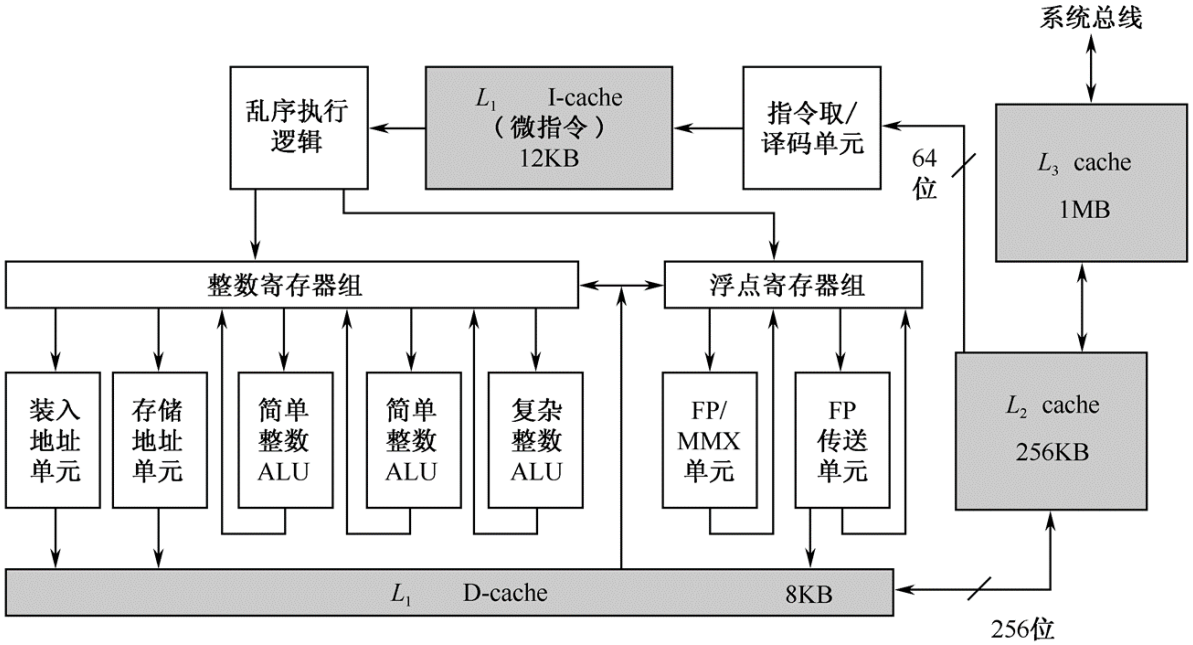


Pentium 4的Cache组织

主要包括四个部分：

- 执行单元：它执行微指令，从L1数据cache中取所需数据，并在寄存器组中暂存运算结果。
- 存储器子系统：这部分包括L2cache、L3cache和系统总线。当L1、L2cache未命中时，使用系统总线访问主存。系统总线还用于访问I/O资源。

Pentium 4主要特点：
将机器指令译成由微指令组成的简单RISC类指令，而使用简单定长的微指令可允许采用超标量流水线和调度技术，从而增强机器的性能



多级cache减少缺失损失

[例10] 现有一处理器，基本CPI为1.0，所有访问在第一级cache中命中，时钟频率5GHz。假定访问一次主存储器的时间为100ns，其中包括所有缺失处理。设平均每条指令在第一级cache中产生的缺失率为2%。若增加一个二级cache，命中或缺失的访问时间都为5ns，且容量大到可使必须访问主存的缺失率降为0.5%，问处理器速度提高多少。

CPI（每条指令周期数）：周期数/指令条数

主存周期数：100ns/0.2ns=500周期

总的CPI=基本CPI +存储器中停顿时钟周期

只有一级Cache：$1 + 500 * 0.02 = 11$

有两级Cache：$1 + 0.02 * 25 + 500 * 0.005 = 4$

后者是前者CPU性能的：$11.0 \div 4.0 = 2.8$倍

单选题 1分

设置

此题未设置答案，请点击右侧设置按钮

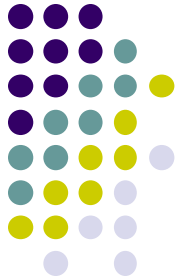
二路组相联Cache采用LRU替换算法
某次读Cache未命中，需进行替换操作
对应组内块计数器计数器值分别为：5、15
Cache此时选择替换掉块计数器值为？

A 5

B 15

提交

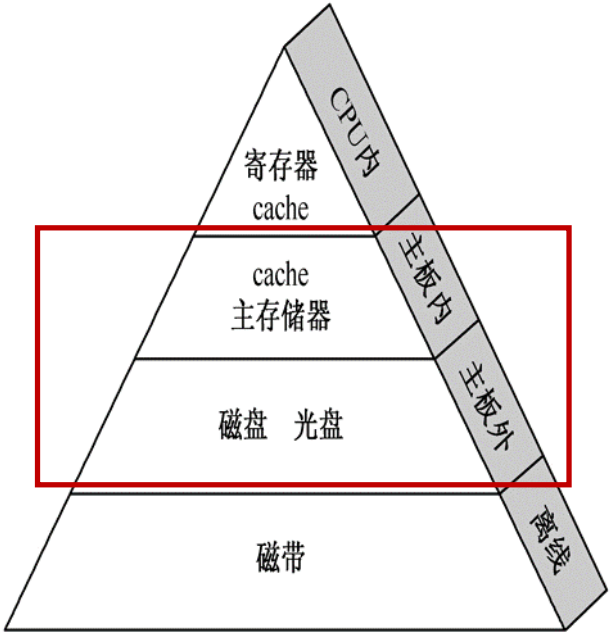
第三章 多层次的存储器



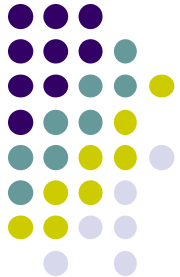
- Cache基本原理
- Cache读写操作
- Cache地址映射
- Cache替换策略与其它
- 虚拟存储器概念

虚拟存储整体目标

- 问题：主存与辅存速度/容量不匹配
- 实现方法：由硬件+操作系统调度
- 出发点相同
  - 为了提高存储系统的性能价格比而构造的分层存储体系
- 原理相同
  - 均利用了程序运行时的局部性原理
  - 把最近常用的信息块从相对慢速而大容量的存储器调入相对高速而小容量的存储器

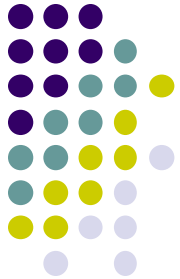


虚拟存储器与Cache异同（1）



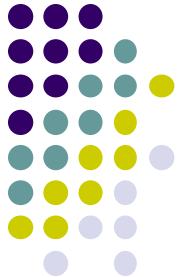
- 主存辅存的访问机制与Cache—主存的访问机制是类似的
- 这是由Cache存储器、主存和辅存构成的三级存储体系中的两个层次
- Cache和主存之间以及主存和辅存之间分别有辅助硬件和辅助软硬件负责地址变换与管理，以便各级存储器能够组成有机的三级存储体系
- Cache和主存构成了系统的内存，而主存和辅存依靠辅助软硬件的支持构成了虚拟存储器

虚拟存储器与Cache异同（2）



- 侧重点不同
  - cache主要解决主存与CPU的速度差异问题
  - 虚存主要是解决存储容量问题，另外还包括存储管理、主存分配和存储保护等方面
- 数据通路不同
  - CPU与cache和主存之间均有直接访问通路，cache不命中时可直接访问主存
  - 虚存所依赖的辅存与CPU之间不存在直接的数据通路，CPU最终还是要访问主存

虚拟存储器与Cache异同（3）



- 透明性不同
  - cache的管理完全由硬件完成，对系统程序员和应用程序员均透明
  - 虚存管理由软件（操作系统）和硬件共同完成，虚存对实现存储管理的系统程序员不透明，而只对应用程序员透明
- 未命中时的损失不同
  - 由于主存的存取时间是cache的存取时间的10倍左右，而主存的存取速度通常比辅存的存取速度快上千倍
  - 主存未命中时系统的性能损失要远大于cache未命中时的损失

总结

- Cache存储器
  - 基本原理
    - 局部性原理
  - 读写操作
    - 标记、行号、字号
    - 写策略
  - 地址映射
    - 直接、组相联、全相联
  - 替换策略
    - LFU、LRU、随机
- 虚拟存储器概念
  - 虚存与Cache对比