

# 计算机组成原理

PRINCIPLES OF COMPUTER ORGANIZATION

第15次课：7.1~7.2 高速缓冲存储器

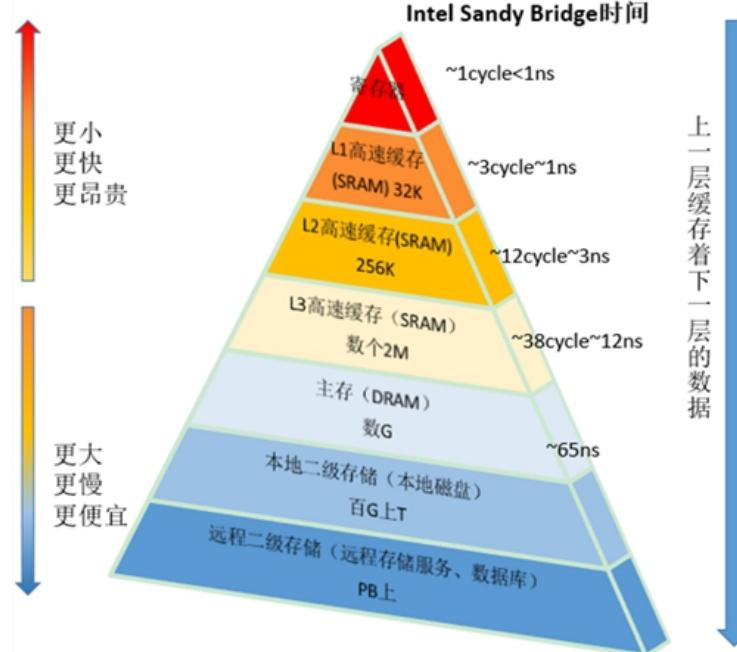
杜国栋

信息科学与工程学院计算机科学与工程系

gddu@ysu.edu.cn



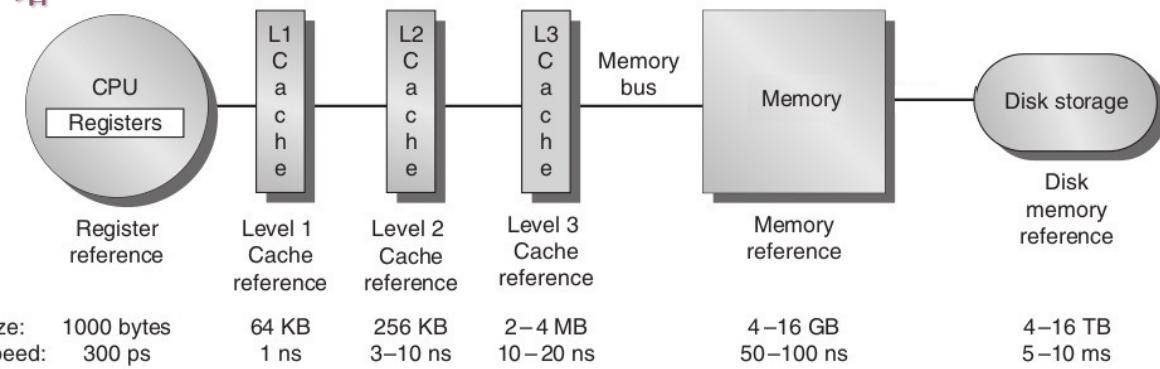
燕山大学  
YANSHAN UNIVERSITY



性能指标（速度、容量、价格）金字塔

起源：早期冯诺依曼式单存储→现代存储系统，即采用单一的存储模式很难满足速度快、容量大、价格低的要求。

存储系统：计算机中由存放程序和数据的各种存储设备（Cache、内存、硬盘等）、控制部件（SATA控制器、内存控制器）以及管理信息调度的硬件设备（MMU+OS、“增删改查”）和算法（块或段页替换算法）所组成的系统。





# 课程目标

- 掌握Cache存储器的地址映像方式；
- 熟悉存储系统的层次结构；
- 了解多层次Cache存储器的工作原理。





# 存储系统的层次结构

➤ 对存储器性能的基本要求：

- 大容量：放得下所有系统软件、多个用户软件。
- 高速度：尽量和CPU匹配。
- 低价格：只占整个计算机系统硬件价格中一个较小而合理的比例。





# 存储系统的层次结构

➤ 存储体系的定义：

- 两个或两个以上速度、容量和价格各不相同的存储器用硬件、软件、或软件与硬件相结合的方法连接起来成为一个存储系统。这个存储系统对程序员是透明的，并且，从程序员看，它是一个存储器，这个存储器的速度接近速度最快的那个存储器，存储容量与容量最大的那个存储器相等，单位容量的价格接近最便宜的那个存储器。





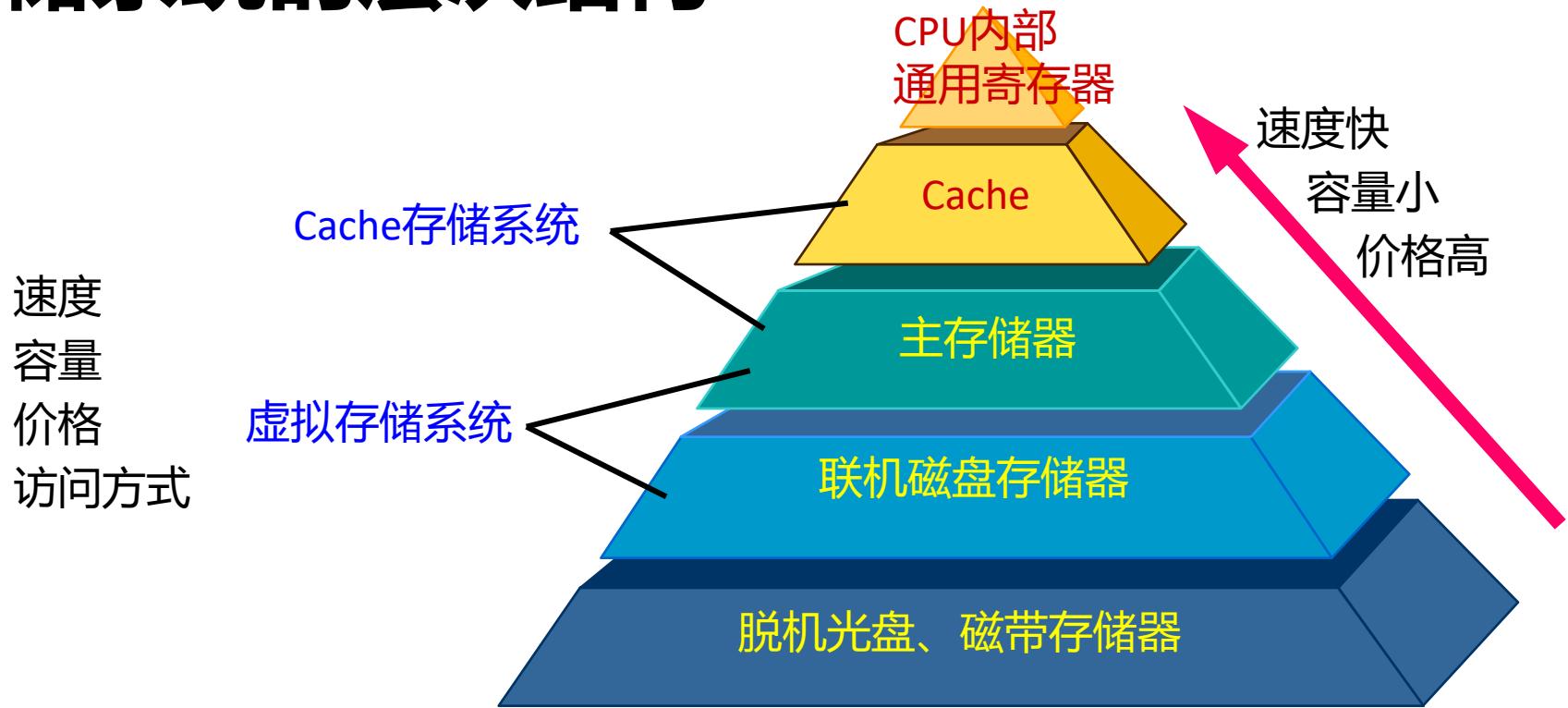
# 存储系统的层次结构

- 一般计算机系统中主要有两种存储体系：
- Cache存储体系：由Cache和主存储器构成；主要目的是提高存储器速度。
- 虚拟存储体系：由主存储器和磁盘存储器构成；主要目的是扩大存储器容量。
  - Cache存储系统：对系统程序员以上均透明
  - 虚拟存储器系统：对应用程序员透明





# 存储系统的层次结构



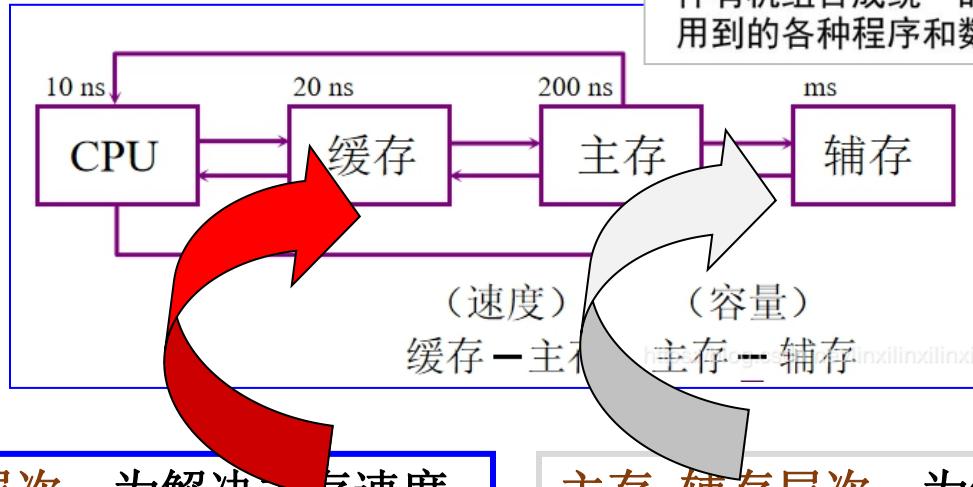
**存储系统**: 将两种或两种以上的**存储器**用**硬件**, **软件**, 或**硬件和软件相结合**的方式联接在一起，并对它们进行**管理**。





# 存储系统的层次结构

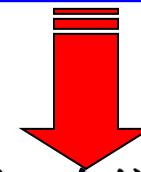
○ 层次存储系统是指把各种不同存储容量、存取速度、访问方式和单位存储价格的存储器，按照一定的层次结构组成多层存储器，并通过管理软件和辅助硬件有机组合成统一的存储体系，使计算机系统中使用到的各种程序和数据层次的分布到各个存储器中。



$1\text{ s (秒)} = 1000\text{ ms (毫秒)}$   
 $1\text{ ms (毫秒)} = 1000\text{ } \mu\text{ s (微秒)}$   
 $1\text{ } \mu\text{ s (微秒)} = 1000\text{ ns (纳秒)}$   
 $1\text{ ns (纳秒)} = 1000\text{ ps (皮秒)}$

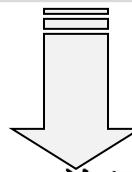
**Cache-主存层次：**为解决主存速度与CPU不匹配的问题，借助于硬件在二者之间增加了一—**CACHE**。

- (1) MM→Cache: 地址映像
- (2) 查询Cache: 地址变换
- (3) Cache→MM: 替换算法
- (4) CPU→Cache: 一致性



**主存-辅存层次：**为解决主存容量不足的问题，借助于软硬件在高速硬盘上开辟了一—**虚拟存储空间**。

- (1) 一次性 (内存容量不够、并行程序量减少)
- (2) 驻留性
- (3) 程序重定位 (虚地址→实地址)



构成了一个统一管理、统一调度，并且对用户来说透明的一体化的存储器系统，即**三级结构的存储器系统**！



- (1) 允许用户程序大于主存；
- (2) 每次访存都需要进行虚实地址变换；



# 存储系统的层次结构

➤ 存储体系的性能参数

➤ 存储体系的容量

➤ 要求:

- ◆ 提供尽可能大的地址空间
- ◆ 能够随机访问

➤ 方法:

- ◆ 只对系统中存储**容量最大的**那个存储器进行编址，其他存储器只在内部编址或不编址  
—— Cache存储系统
- ◆ 另外设计一个容量很大的**逻辑地址空间**，把相关存储器都映射这个地址空间中  
—— 虚拟存储系统





# 存储系统的层次结构

- 存储体系的性能参数
- 存储体系的速度
  - 访问时间  $T_A$ 、存储周期  $T_M$ 、频宽（带宽）  $Bm$
  - 命中率  $H$ : CPU产生的逻辑地址能在  $M_1$  中访问到的概率。

$$H = \frac{N_1}{N_1 + N_2}$$

在  $M_1$  中访问到的次数

在  $M_2$  中（未调到  $M_1$  中）的次数

访问时间与命中率的关系:

$$T_A = HT_{A1} + (1 - H)T_{A2}$$

当命中率  $H \rightarrow 1$  时,  $T_A \rightarrow T_{A1}$



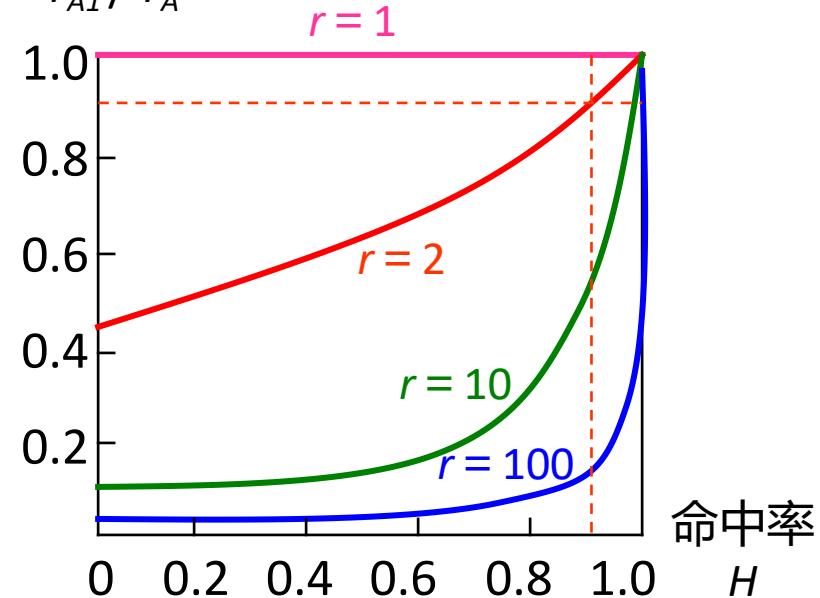


# 存储系统的层次结构

- 存储体系的性能参数
- 存储体系的速度
  - 提高存储体系速度的途径：
    - 提高命中率  $H$ ;
    - 两个存储器的速度不要相差太大，即减小  $r$ 。

访问效率

$$e = T_{A1} / T_A$$



其中：第二条有时做不到（如虚拟存储器），这时，只能依靠提高命中率。





# 高速缓冲存储器

- 计算机中的存储系统，分为三个层次：



**外 存：**确保计算机具有足够大的存储容量；  
确保数据能脱机保存；

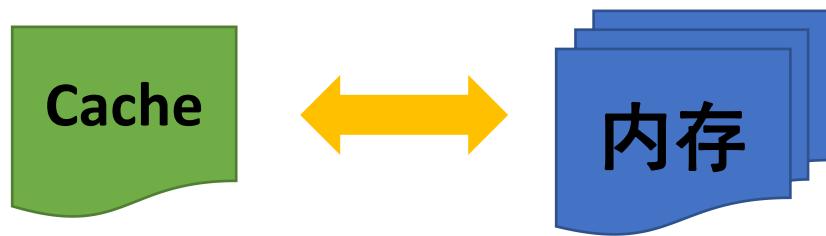
**内 存：**存储运行期指令/数据，确保CPU能快速读取；

**Cache：**强化CPU快速读取指令和数据的速度；



# 高速缓冲存储器

## ※ Cache和内存的关联



内存中的数据，调入Cache  
Cache中的数据，写回内存

以**数据块**为单位整体操作：

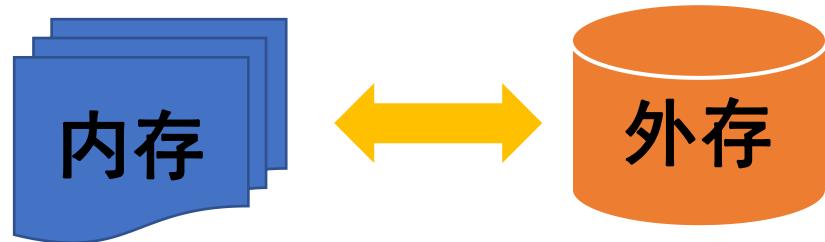
**直接映射、全相联映射、组相联映射**





# 高速缓冲存储器

## ※内存和外存的关联



外存中的数据，调入内存  
内存中的数据，写回外存

虚拟存储技术： 页式、段式、段页式





# 高速缓冲存储器

## ➤ 设置Cache的原因

※ 为解决**CPU和主存速度不匹配**而采用的一项技术，使访问主存的平均速度接近于访问Cache的速度。

※ 由**硬件系统**实现，对用户透明。

※ 已在CPU内集成，**两级以上的Cache系统**。



# 高速缓冲存储器

➤ 指令的执行具有局部性特征：

CPU从主存中取指令、数据，在一定时间内地址范围常局限于主存的某个小区域。

因此可以将正在使用的部分（热点区指令和数据），提前预取并存储到一个高速的、小容量的Cache中。

CPU访存



CPU访问Cache

[技术效果]

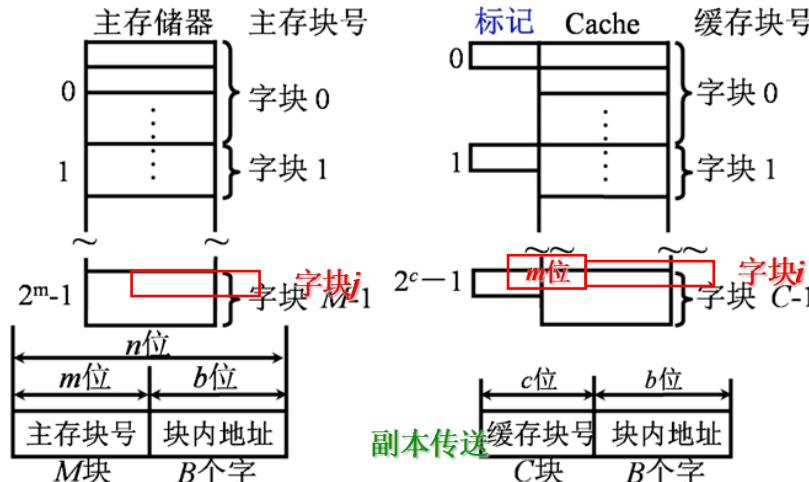
能使CPU读写指令、数据的速度大大提高。



# 高速缓冲存储器

## Cache 的工作原理

### (1) 主存和缓存的编址



主存和缓存按块存储传送(字节地址适用)

假设：内存的空间被分为  $M_m(0), M_m(1), \dots, M_m(j), \dots, M_m(2^m - 1)$  共  $2^m (M=2^m)$  个块；Cache 的空间被分为  $M_c(0), M_c(1), \dots, M_c(i), \dots, M_c(2^c - 1)$  共  $2^c (C=2^c)$  个块。在两个主存储器中，块的大小一致，其地址位数占  $b$  位，即有  $B = 2^b$  个字。

主存容量： $N=2^n$ 字  
一个块大小： $B=2^b$ 字

结合主存储器章节中介绍的相关概念，  
主存中块的数量为： $2^{n-b}$ ，即块号为  $n-b$  位  
令  $m=n-b$ ,  $M=B/N=2^{n-b}=2^m$

主存和Cache分成大小相等的块，  
主存有  $M$  块， Cache 有  $C$  块，  
 $C << M$ 。主存地址由主存块号和  
块内地址构成， Cache 地址作用不  
大。两者块大小完全相同，在内  
存和 Cache 间传送是整体进行传送，  
块内字节顺序不会发生变化。  
Cache 块的标记指主存块号，当  
CPU 给出一个内存地址，将该地  
址的主存块号和标记比较，如果  
相等且该 Cache 块有效，可直接从  
Cache 中获取该信息，速度得到很  
大提高。

### (2) Cache 的命中

**命中：** 主存块调入缓存，主存块与缓存块建立了对应关系。用标记记录与某缓存块建立了对应关系的主存块号。

**命中率：** CPU 欲访问的信息在 Cache 中的比率。与 Cache 的容量与块长有关。

一般每块可取 4~8 个字，块长取一个存取周期内从主存调出的信息长度。

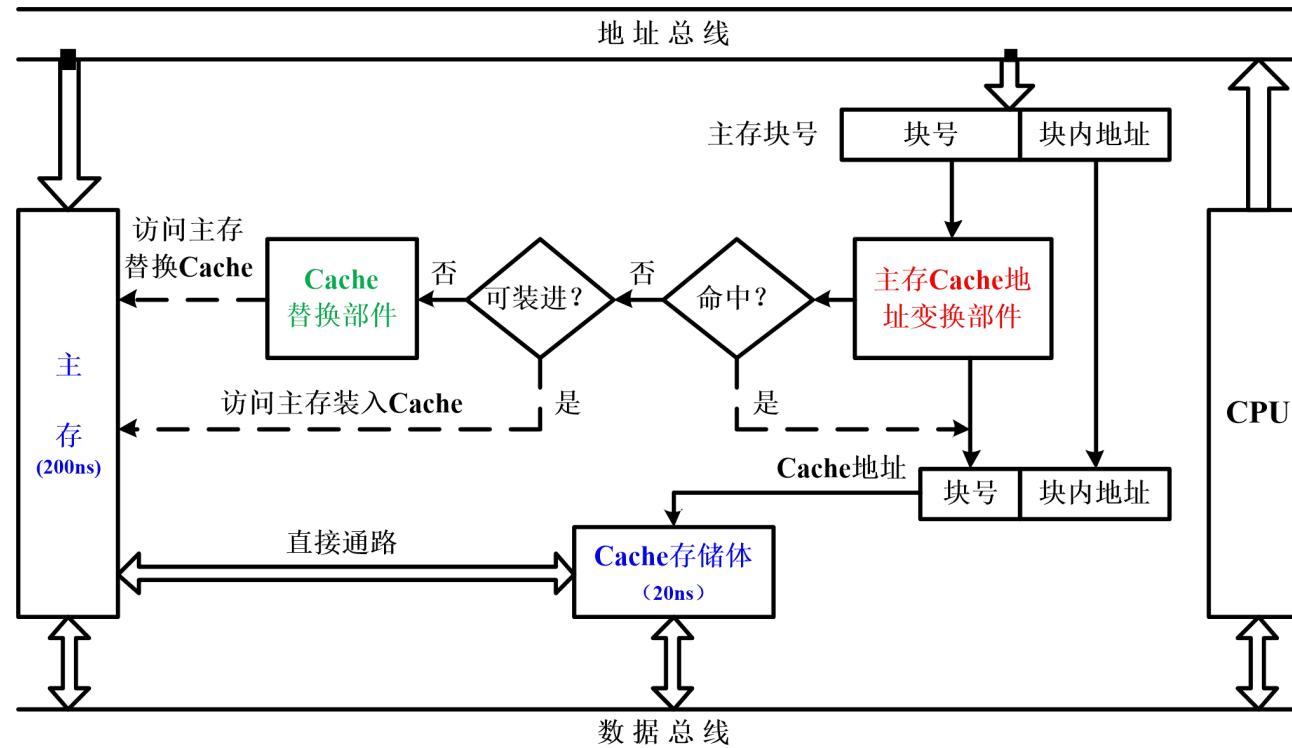


# 高速缓冲存储器

## ➤ Cache的基本结构：

Cache存储器主要由三个部分组成：

- (1) Cache存储器，用于存放由主存储器调入的指令与数据块；
- (2) 地址转换部件，用于实现主存储器地址到Cache存储器地址的转换；
- (3) 替换部件，当缓存满时根据指定策略进行数据块替换，并对地址转换部件做对应修改。





# 高速缓冲存储器

## ➤ Cache技术指标：

Cache-主存系统的命中率：在一个程序执行期间，设  $N_c$  为访问 Cache 的总命中次数， $N_m$  为访问主存的总次数，则命中率  $h$  为：

$$\text{命中率: } h = \frac{N_c}{N_c + N_m}$$

Cache-主存系统的平均访问时间：设  $t_c$  为命中时的 Cache 访问时间， $t_m$  为未命中时访问内存的访问时间， $1-h$  为未命中率，则 Cache-主存系统的平均访问时间  $t_a$  为：

$$\text{平均访问时间1: } t_a = h \cdot t_c + (1 - h) \cdot t_m$$

上述公式只适用于 Cache 和内存同时被访问的情况。

下述公式适用于 Cache 先被访问，如果不命中，内存后被访问的情况。

如果不命中则访问内存的情况，按照上述理由，只是在不命中时多出了一个访问 Cache 的时间  $t_c$ ，即 Cache-主存系统的平均访问时间  $t_a$  为：

$$\text{平均访问时间2: } t_a = h \cdot t_c + (1 - h) \cdot (t_m + t_c) \quad \text{教材上所使用的公式}$$





# 高速缓冲存储器

## ➤ Cache技术指标：

Cache-主存系统的访问效率：以较小的硬件代价使 Cache-主存系统的平均访问时间  $t_a$  越接近于  $t_c$  越好。用  $e$  表示访问效率，则有

$$\text{访问效率: } e = \frac{t_c}{t_a} \times 100\% = \frac{t_c}{h \cdot t_c + (1-h) \cdot t} \times 100\%$$



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，Cache/主存系统的效率为( [填空1] )%(保留一位小数点),平均访问时间为 ( [填空2] ) ns(保留整数部分即可)。

作答



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

- Cache 命中次数  $N_c = 1900$  主存命中次数  $N_m = 100$
- Cache存取周期  $t_c = 50\text{ns}$  主存存取周期  $t_m = 250\text{ ns}$
- 命中率  $h = N_c / (N_c + N_m) = 1900 / (1900 + 100) = 0.95$
- 主存慢于cache的倍率  $r = t_m / t_c = 250\text{ns} / 50\text{ns} = 5$
- Cache/主存系统的效率：  $e = 1 / (r + (1 - r)h) = 1 / (5 + (1 - 5) \times 0.95) = 83.3\%$
- 平均访问时间：  $ta = t_c / e = 50\text{ns} / 0.833 = 60\text{ns}$



设某流水线计算机有一个指令和数据合一的Cache，已知Cache的读/写时间为10ns，主存的读/写时间为100ns，取指的命中率为98%，数据的命中率为95%，在执行程序时，约有1/5指令需要存/取一个操作数，为简化起见，假设指令流水线在任何时候都不阻塞。问设置Cache后，与无Cache比较，计算机的运算速度可提高（[填空1]）倍？

作答



已知cache的读/写时间为10ns，主存的读/写时间为100ns，取指的命中率为98%，数据的命中率为95%，在执行程序时，约有1/5指令需要存/取一个操作数，为简化起见，假设指令流水线任何时候都不阻塞。问设置cache后，与无cache比较，计算机的运算速度可提高（ ）倍

- Cache存取周期  $tc = 10\text{ns}$  主存存取周期  $tm = 100\text{ ns}$
- 有cache平均访存时间：  
$$(10\text{ns} \times 0.98 + (10+100) \times 0.02) + (10\text{ns} \times 0.95 + (10+100) \times 0.05) \times 1/5 = 15\text{ns}$$
- 无cache平均访存时间：  $100 \times 1 + 100 \times 1/5 = 120\text{ns}$
- 速度提高倍数：  $120\text{ns}/15\text{ns} = 8$



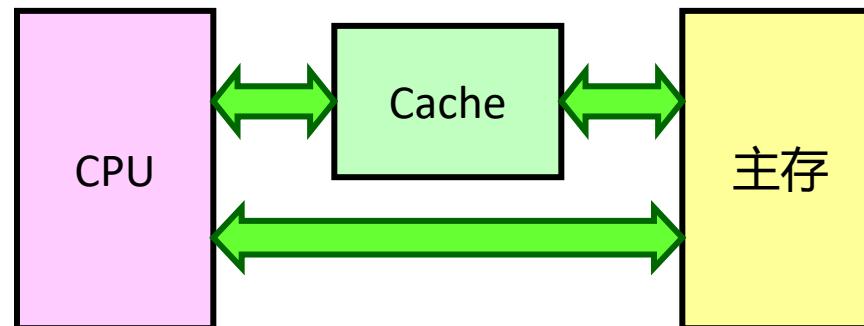


# 高速缓冲存储器

## ➤ 工作原理

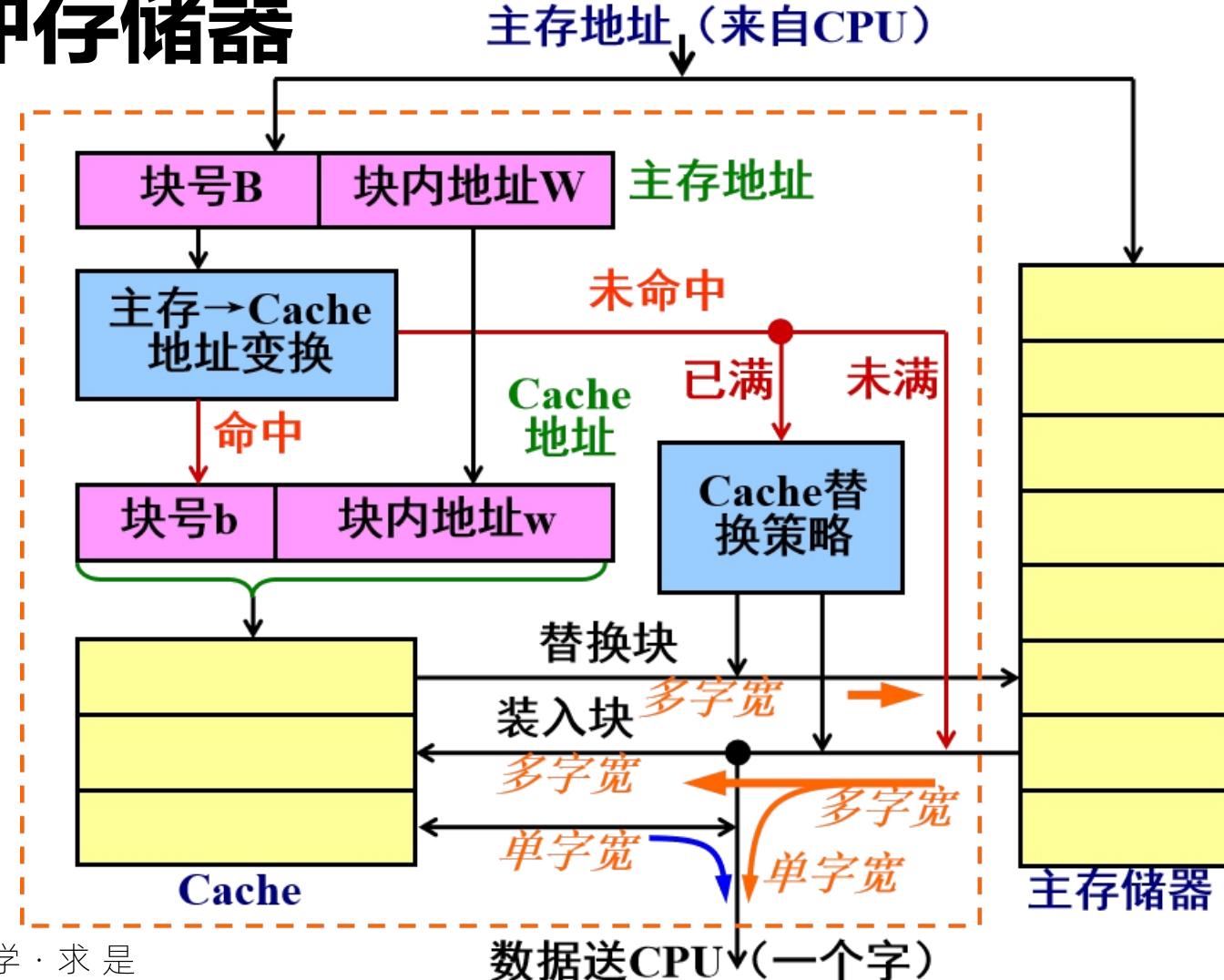
### ➤ Cache的工作过程

- 流水处理：地址变换、访问Cache
- Cache-处理机通路
- 主存-处理机直接通路
- 有Cache存储器的主存系统都采用多体交叉存储器。





# 高速缓冲存储器





# 有问题欢迎随时跟我讨论

办公地点：西校区信息馆423

邮 箱：[gddu@ysu.edu.cn](mailto:gddu@ysu.edu.cn)

