

# 操作系统 Operating System

必修课程，3.5学分，64学时：

● 1~12周上课48学时；实验16学时；

《操作系统精髓与设计原理》第9版

William Stallings, 陈向群等译, 电子工业出版社

先修课程：数据结构，C语言，计算机原理

孙微微 13711225896, svvsun@163.com

本级课程Q群：712355491



群名称:21计算机1234-OS-SVV

群 号:712355491



# 课程内容与目的



讲：第1～12章

不是如何使用操作系统

而是理解操作系统如何工作

- OS是所有软件、硬件的管理者
- OS功能、内部算法和数据结构、实现技术
- 设计OS过程中遇到的问题或矛盾、解决方案和折中权衡
- 有代表性的操作系统实例、发展方向

# 成绩评定



- **总评成绩 = 平时成绩 × 50% + 期末卷面成绩 × 50%**
- **期末考试：卷面满分100分。不足60时总评等于卷面。**
- **平时成绩组成：**
  - **作业 × 20% + 随堂测验 × 30% + 实验 × 50%**
  - **作业：**各次作业以A<sup>+</sup>=100, A=95, A<sup>-</sup>=90, B<sup>+</sup>=85, B=80, B<sup>-</sup>=75, C=50标准计分，不交0分。
  - **随堂测验：**随堂测验4次，每次100分，取4次测验的平均分。不考0分。（在线教育平台，课程伴侣app）
    - ✿ 第1~4章、5~6章、7~8章、9~12章各测验1次。
  - **实验：**待定。



# 第1章 计算机系统概述

## 1.1 基本构成

## 1.2 微处理器的发展演化

## 1.3 指令的执行

## 1.4\* 中断

## 1.5\* 存储器的层次结构

## 1.6 高速缓存

## 1.7\* 直接内存存取

## 1.8 多处理器和多核计算机组织结构

注：本课件中各知识点用下列符号表示其重要程度或难度：



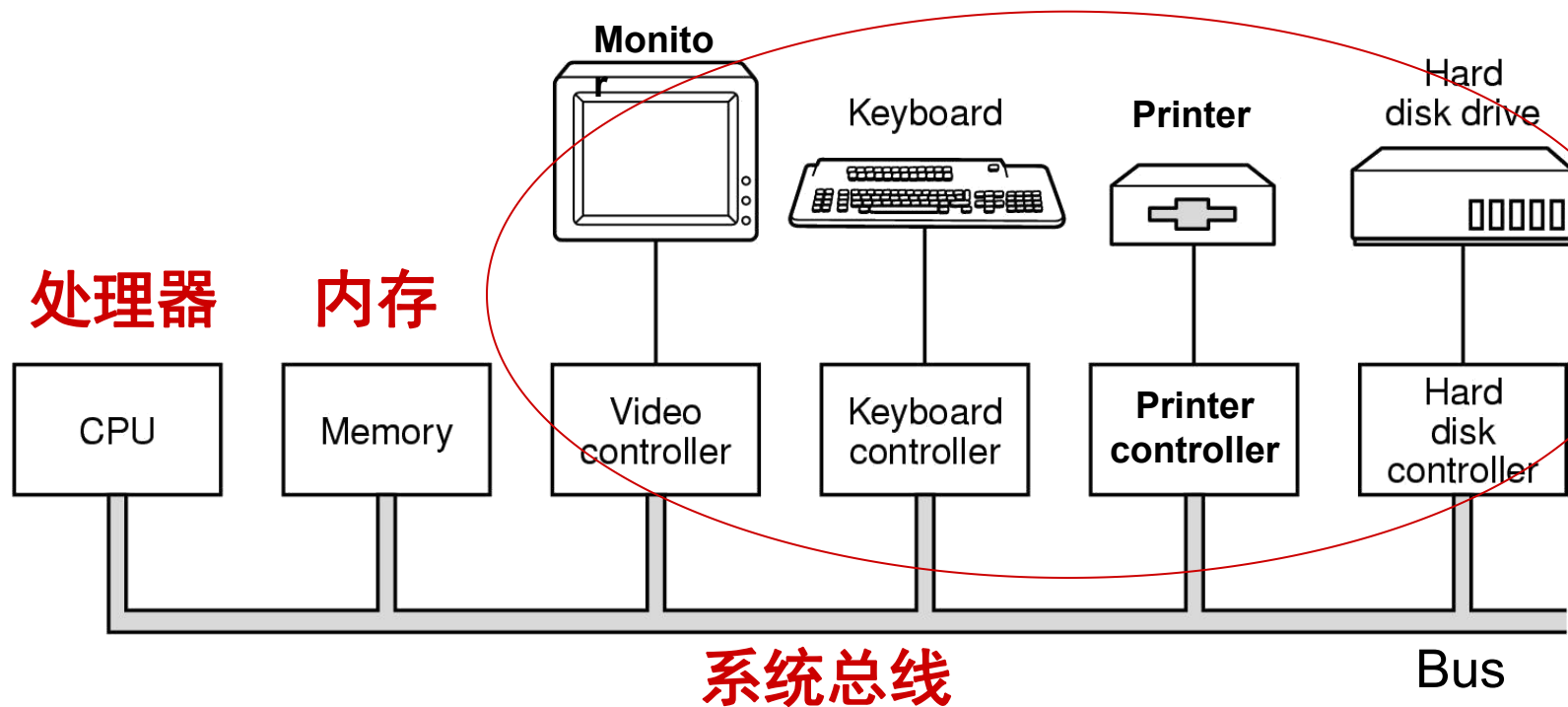
考试内容为三星及三星以上。

# 1.1 基本构成



## 计算机硬件的四个主要部分：

输入输出模块：即外部设备



# 1.1 基本构成



## 计算机部件内部结构：

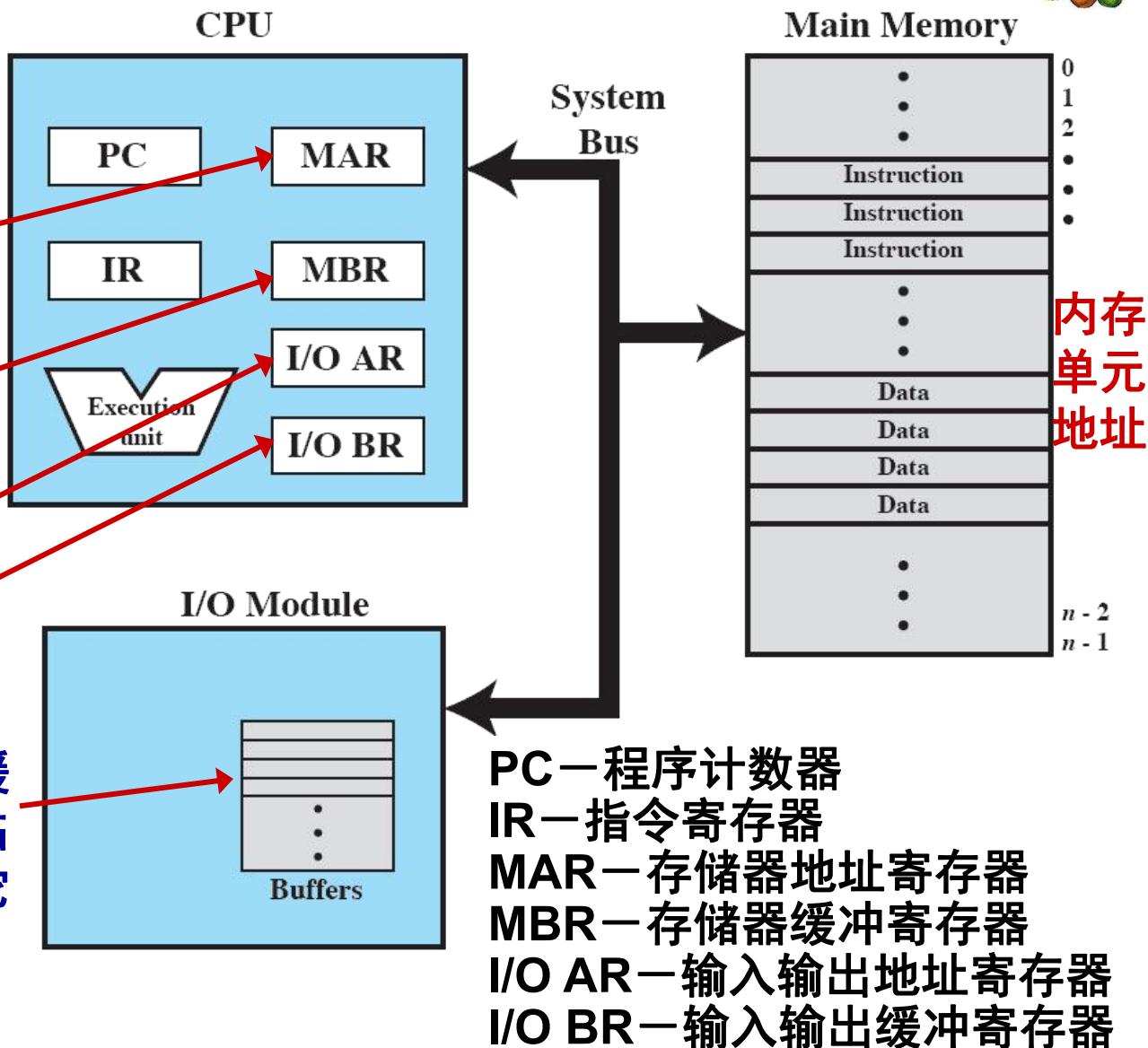
确定下次读写的内存单元地址

存放要写入内存或从内存读取的数据

确定某个I/O设备

CPU与I/O设备间输入输出的数据

输入输出设备需要缓冲区Buffer，用于临时保存数据，直到它们被输入输出完毕。



# 1.2 微处理器的发展演化



❑ 微处理器CPU：单核 ➡ 多核。一个芯片底座上集成了多个运算核心Core，甚至每个Core上再实现硬件的超线程（HyperThreading）。

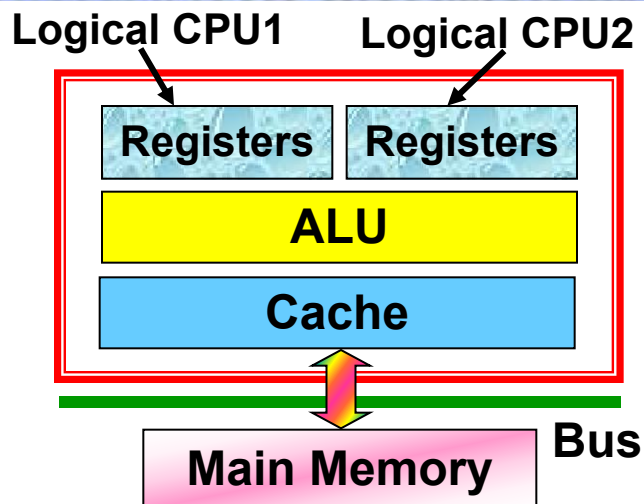
❑ Intel Pentium 4的超线程：

- 一颗实体CPU中有两个逻辑处理单元，每个逻辑处理单元有独立的寄存器组，能同时执行两个独立的线程代码流。（OS以为有两个CPU）
- 但对于CPU中的共享资源，如运算器ALU、Cache、TLB、总线接口等的争用会影响性能。
- 适合支持多处理器的OS，如Win XP及以上。

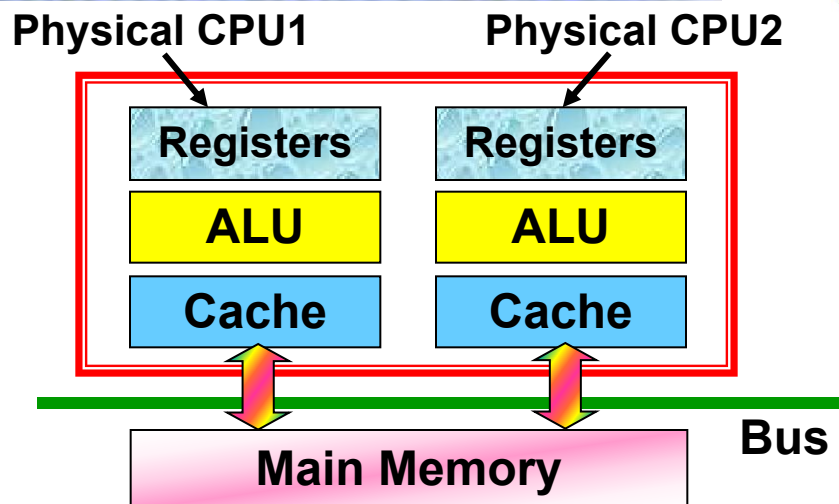
❑ 改进后的超线程用于Core i7处理器上，称为同步多线程技术（Simultaneous Multi-Threading, SMT）。



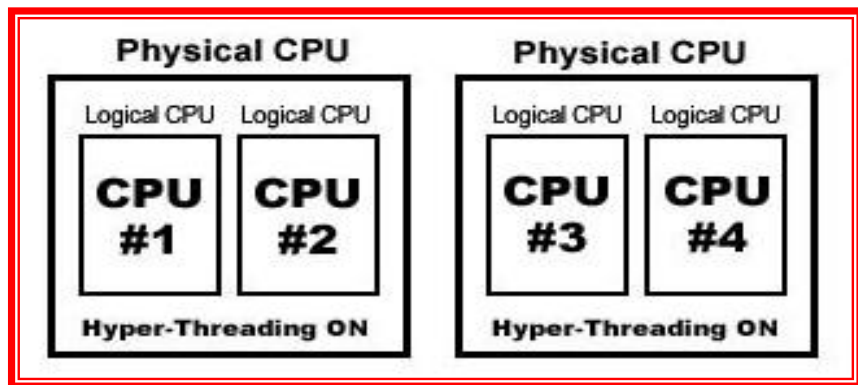
# 1.2 微处理器的发展演化



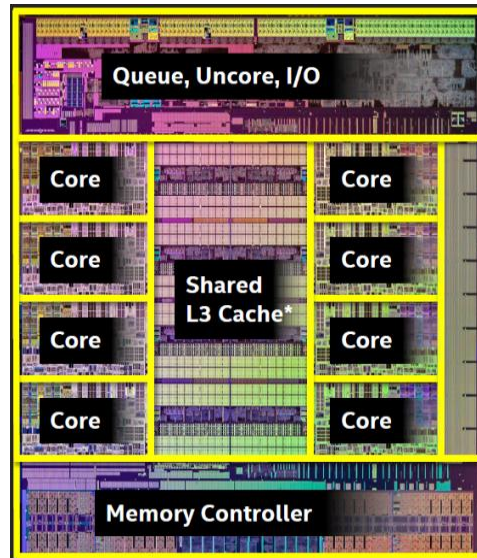
具有超线程技术的CPU



双核CPU



双核超线程CPU(4个逻辑CPU)  
P4 Extreme Edition



采用SMT的  
Core i7  
8核16线程



# 1.2 微处理器的发展演化



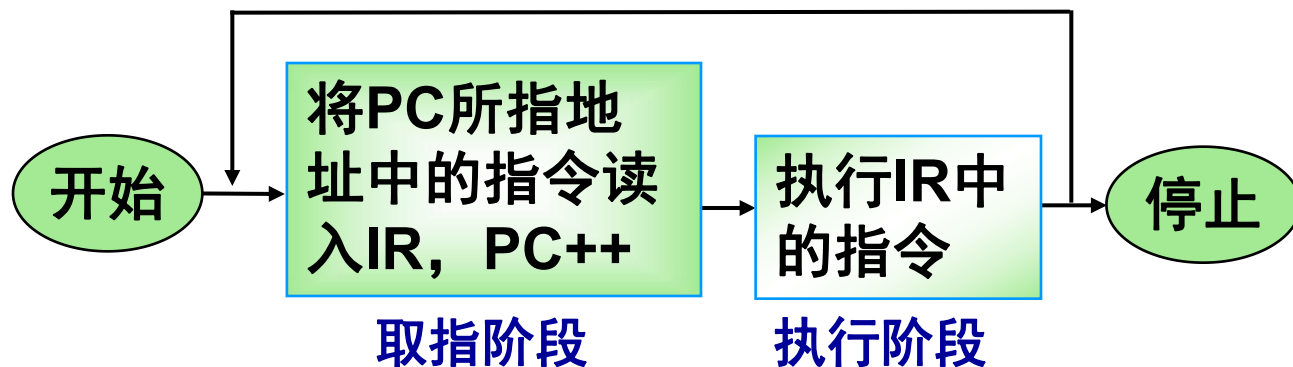
## 其它日趋重要的处理器：

- 图形处理单元(**G**raphic **P**rocessing **U**nit)：用于图形图像处理和并行运算。其单指令流多数据流的硬件架构，使之并行运算和浮点运算性能强大。
- 数字信号处理器(**D**igital **S**ignal **P**rocessor)：实时转换音频视频等模拟信号流和数字信号。数据总线与指令总线分离，可同时访问指令和数据。
- 片上系统(**S**ystem on a **C**hip)：在单个芯片上集成CPU、Cache、内存、I/O部件及其它硬件，常用于嵌入式系统和便携式设备。

# 1.3 指令的执行



## 基本指令周期：取指，执行



执行指令时的PC值变化

199: .....  
200: ADD AX, BX  
201: JMP 300  
202: .....  
.....  
300: HALT

200
201
202
300

## 四类指令：

- **处理器—存储器**：在寄存器和内存单元间传送数据
- **处理器—I/O**：处理器和I/O设备间传送数据
- **数据处理**：算术/逻辑运算
- **控制转移**：设置PC值，改变执行顺序

# 1.4 中断



发生中断时 *Flash*



## 中断Interrupt:

- 由于发生某些事件，暂停当前程序在CPU上的运行，转而执行相应事件的中断处理程序。待处理完成后，再返回断点继续执行或调度其它程序执行。

## 中断分类:

- **程序中断**: 由程序指令的执行结果产生，如算术溢出、除数为0、非法指令、地址越界、缺页、断点调试等。
- **时钟中断**: CPU内部的定时器产生。
- **I/O中断**: I/O操作正常或异常结束时，由I/O控制器产生。
- **硬件失效中断**: 掉电、内存奇偶校验错等硬件故障。

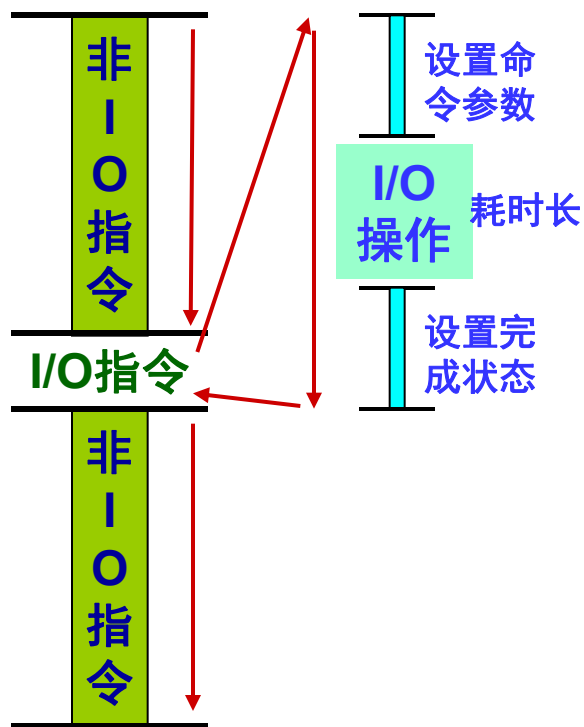
# 1.4 中断—无中断、有中断时的执行过程



## 无中断时的控制流程：

用户程序

I/O程序



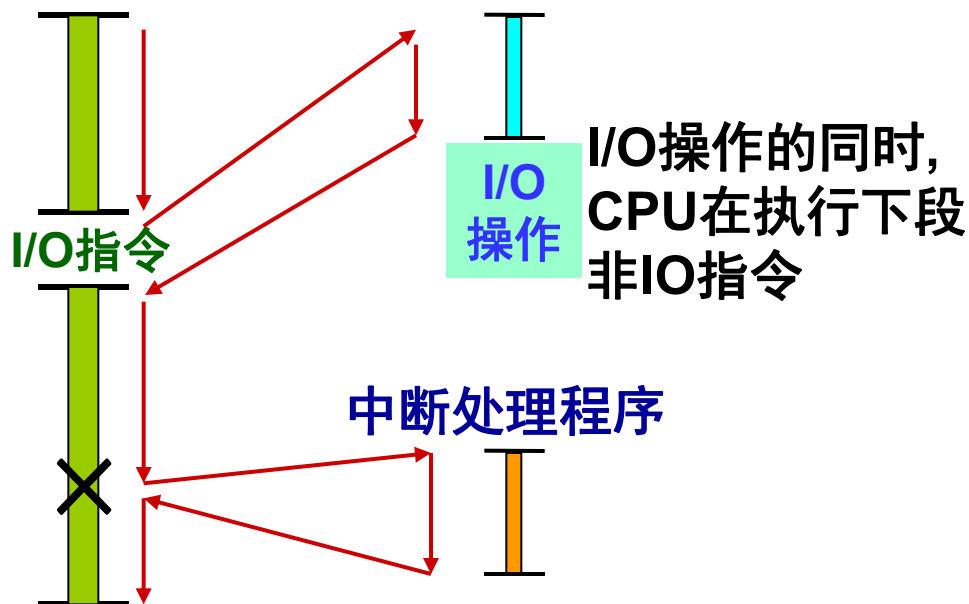
CPU等待I/O操作结束

图中红线为CPU动作轨迹

## 有中断时：以短I/O为例

用户程序

I/O程序



×：I/O操作结束时，发来中断信号。

**短I/O**：下段非IO指令尚未执行完，本次IO操作已结束，并发出中断。

**长I/O**：本次IO操作时间长，尚未结束时已发生下次I/O请求，则顺序处理各I/O请求。

# 1.4 中断—中断提高CPU利用率



❑ 无中断时，CPU和I/O设备串行工作：


CPU运行： \_\_\_\_\_ CPU等待I/O操作结束 \_\_\_\_\_  
I/O设备运行： \_\_\_\_\_

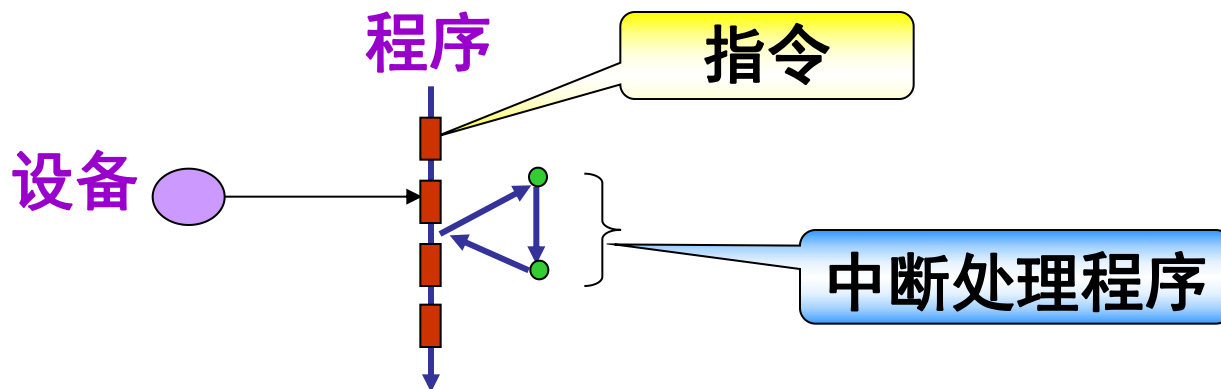
❑ 有中断时，CPU和I/O设备并行工作：  
CPU完成更多任务，提高了CPU利用率。

CPU运行： \_\_\_\_\_ I/O操作期间，CPU继续执行其它任务 \_\_\_\_\_  
I/O设备运行： \_\_\_\_\_

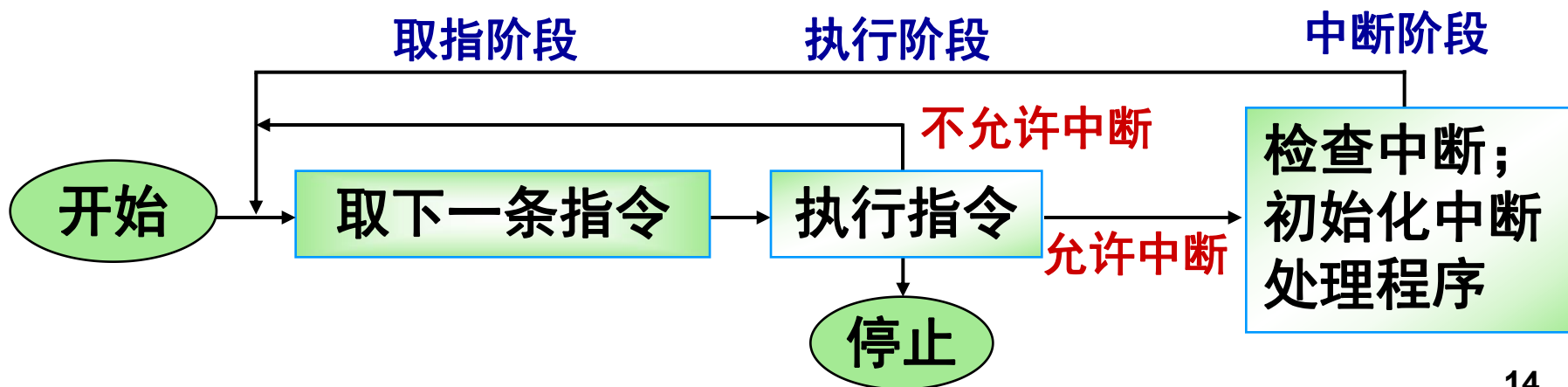
# 1.4.1 中断和指令周期



 CPU在执行完一条指令之后、执行下一条指令之前，检测并处理中断。



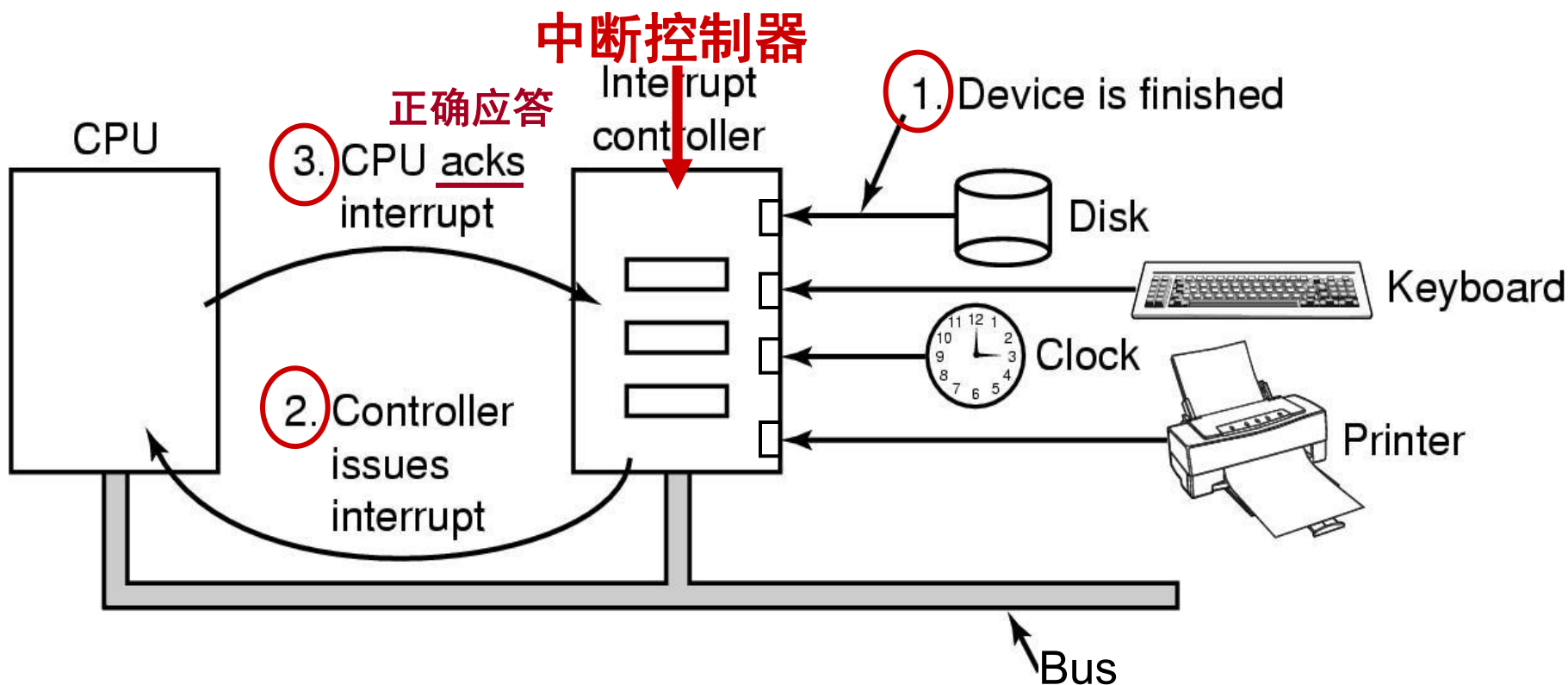
 有中断时的指令周期：



## 1.4.2 中断处理—硬件连接



 CPU和**中断控制器**之间用总线上的**中断请求线**连接。设备发出中断后，CPU响应中断。





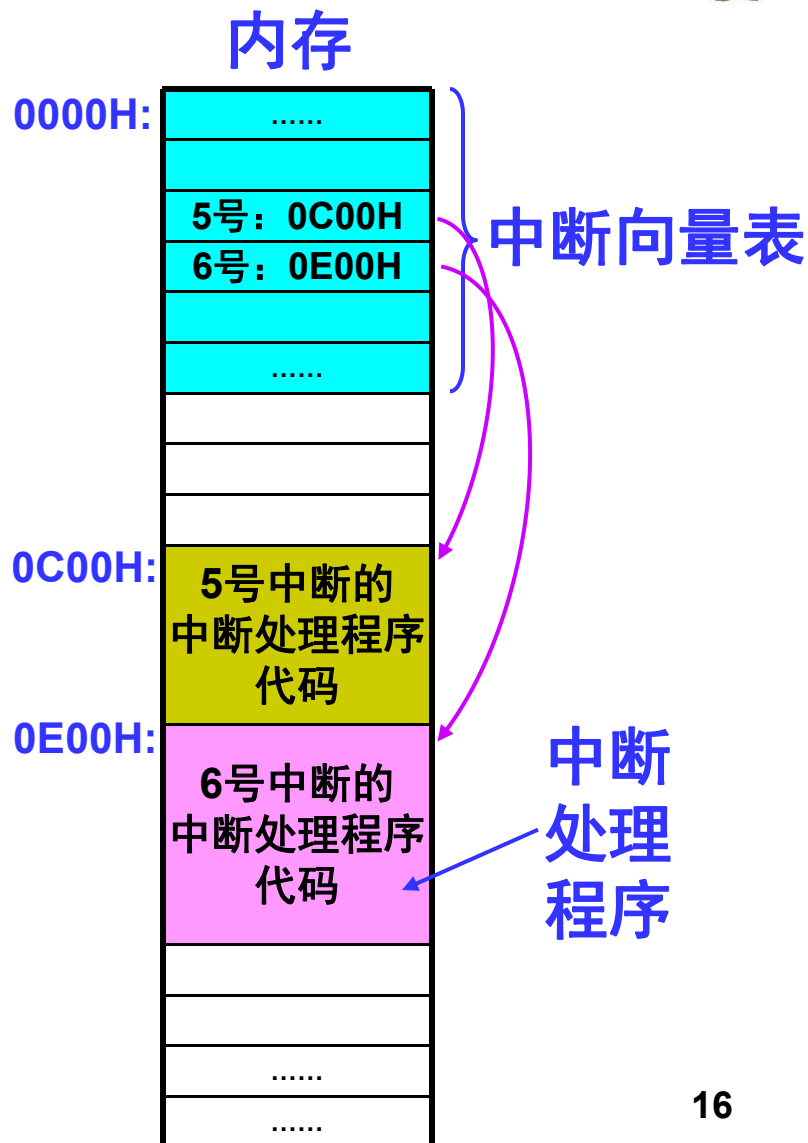
# 1.4.2 中断处理—软件实现



■每个中断有一个**中断向量号**（如：Intel x86 CPU允许0~255），是该中断在中断向量表中的编号。

■**中断向量**：中断处理程序的(内存)入口地址。4B/向量。

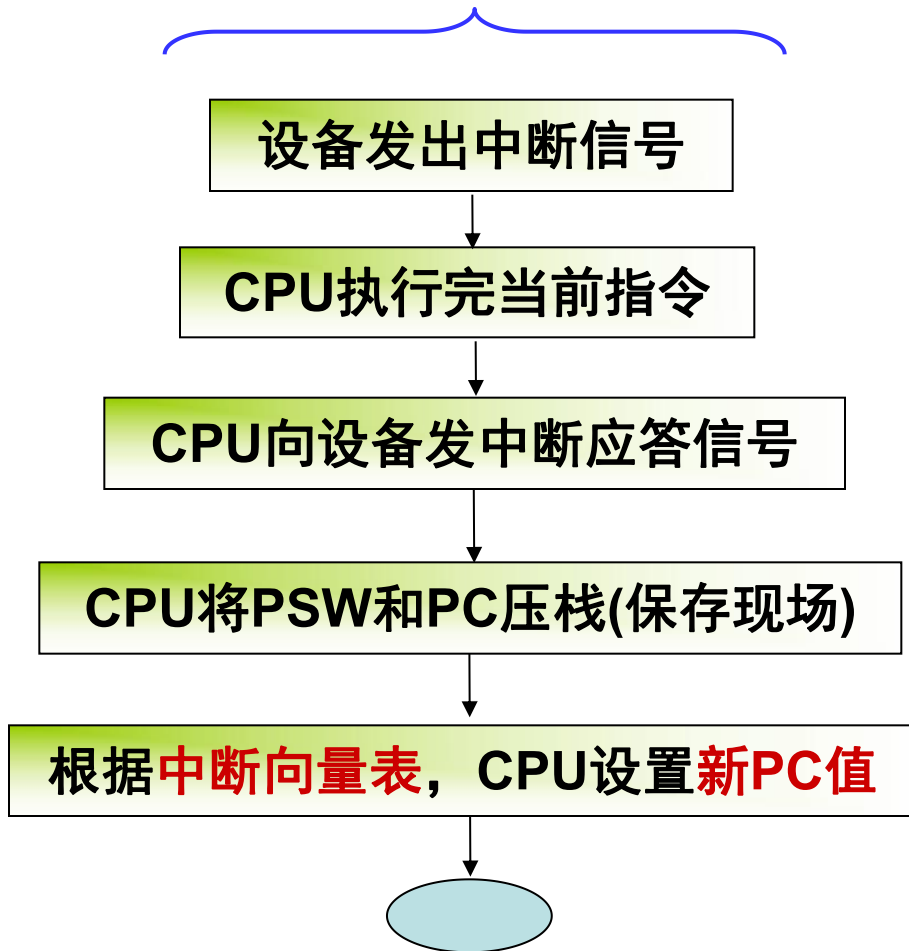
■**中断向量表**：所有中断向量的集合。一般位于内存的0地址起始处（长1KB），在开机启动时从磁盘装入内存。



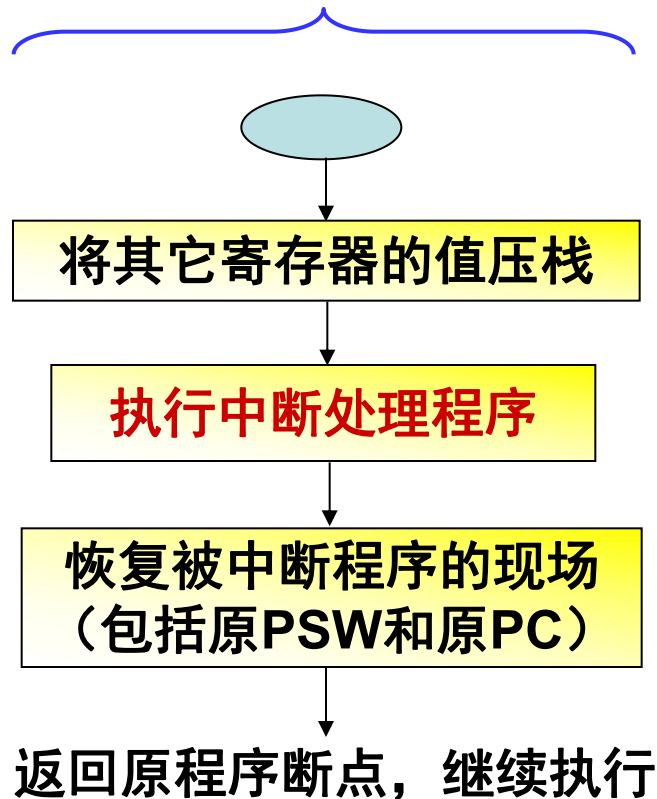
# 1.4.2 中断处理一过程



## 由硬件执行



## 由中断处理程序执行

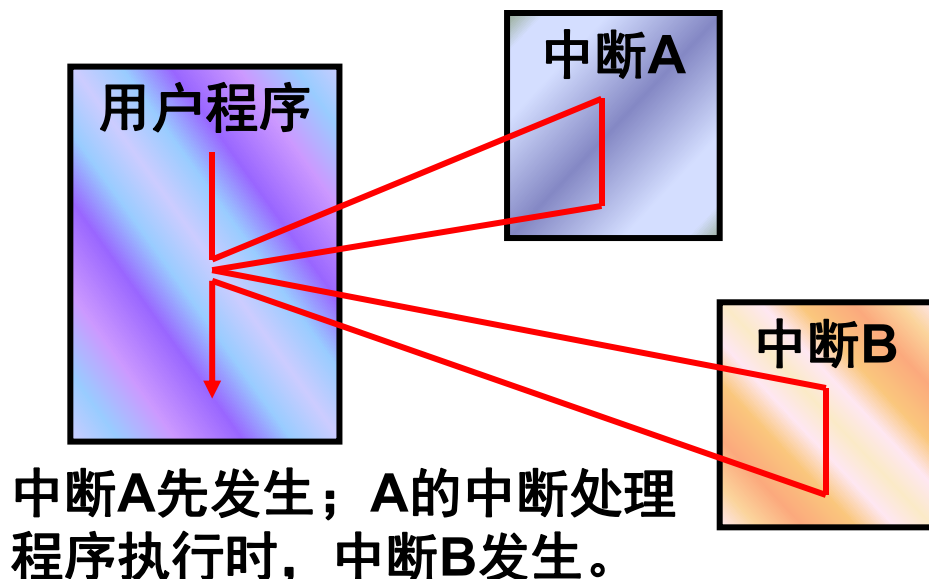


## 1.4.3 多个中断—两种处理方法



①、**顺序处理**：当处理一个中断时，禁止中断。  
多个中断被顺序处理。

- **禁止中断（关中断）**：CPU的PSW的中断禁止位=0，则不响应新发生的中断。
- **开中断**：中断禁止位=1，CPU可以接收中断。



**中断顺序处理** *Flash*

中断A和中断B几乎同时发生时，依次顺序处理。

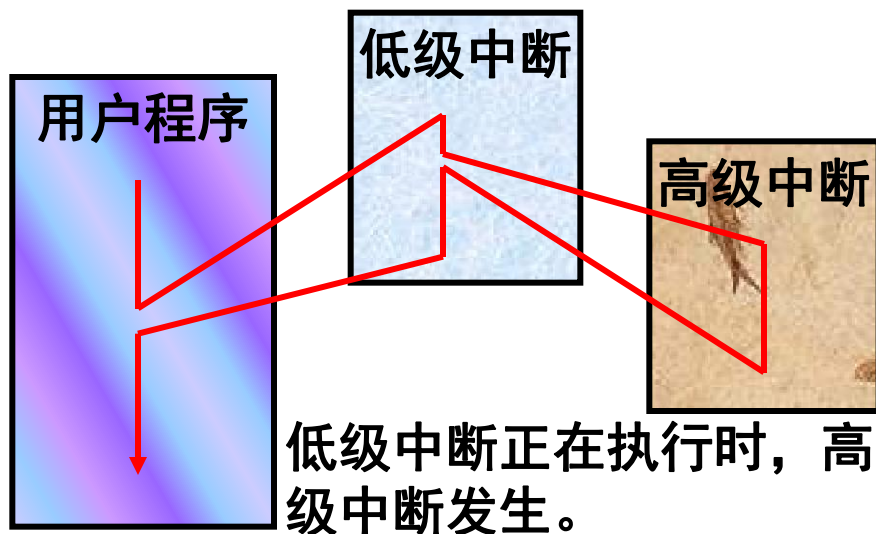
## 1.4.3 多个中断—两种处理方法



②、**中断嵌套**：允许高优先级的中断请求打断低级中断的处理。通常，中断源速度越快，其优先级越高。

- **中断屏蔽**：设置PSW的中断屏蔽码（当前程序的中断优先级），选择性地封锁部分中断，当发生更高优先级的中断时才响应。

- 有些中断(如掉电)是不能屏蔽甚至不能禁止的。



### 多个中断的嵌套 Flash

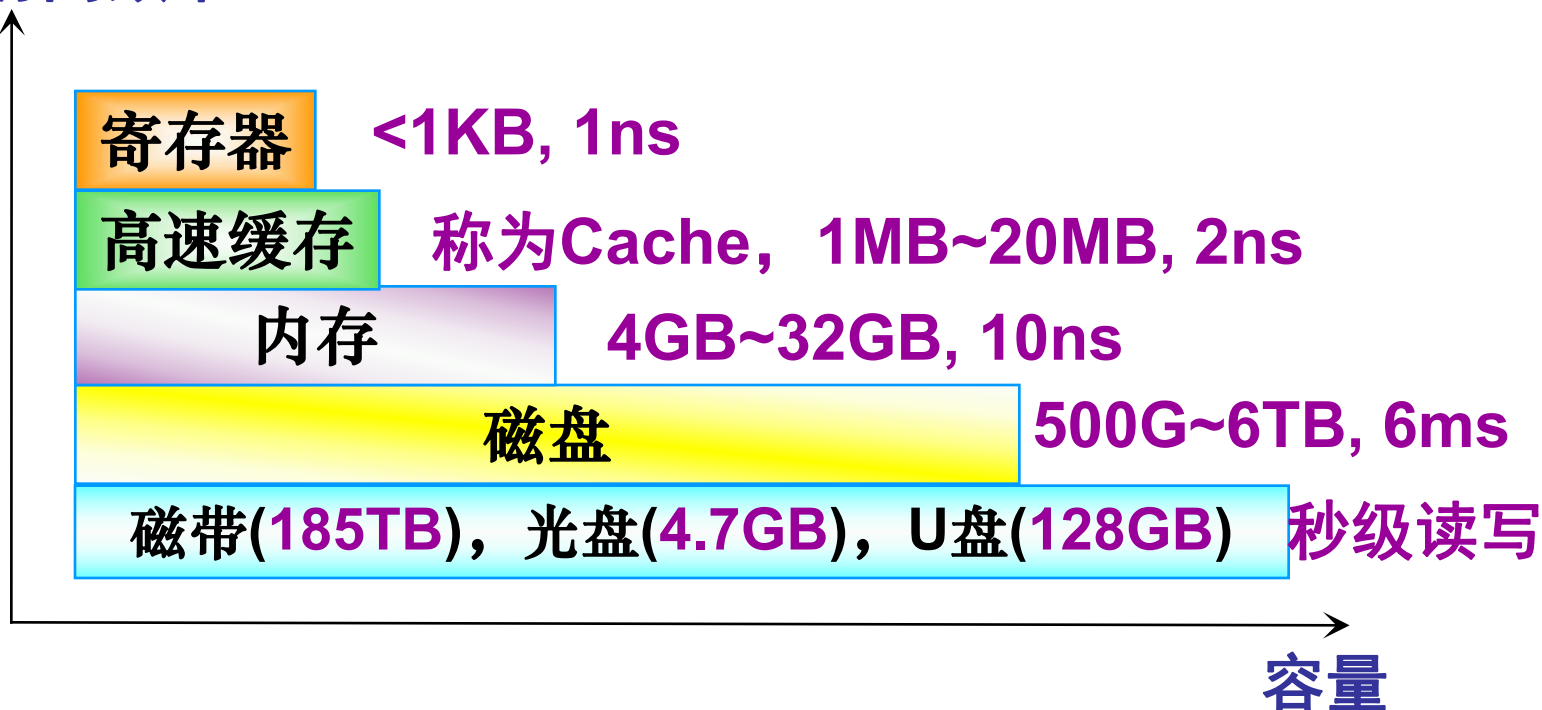
多个中断按优先级嵌套处理。  
例：图1.13中，优先级顺序为  
通信中断>磁盘中断>打印机中断

# 1.5 存储器的层次结构



把速度、容量、价格不同的多种物理存储器组成一个存储器层次系统。

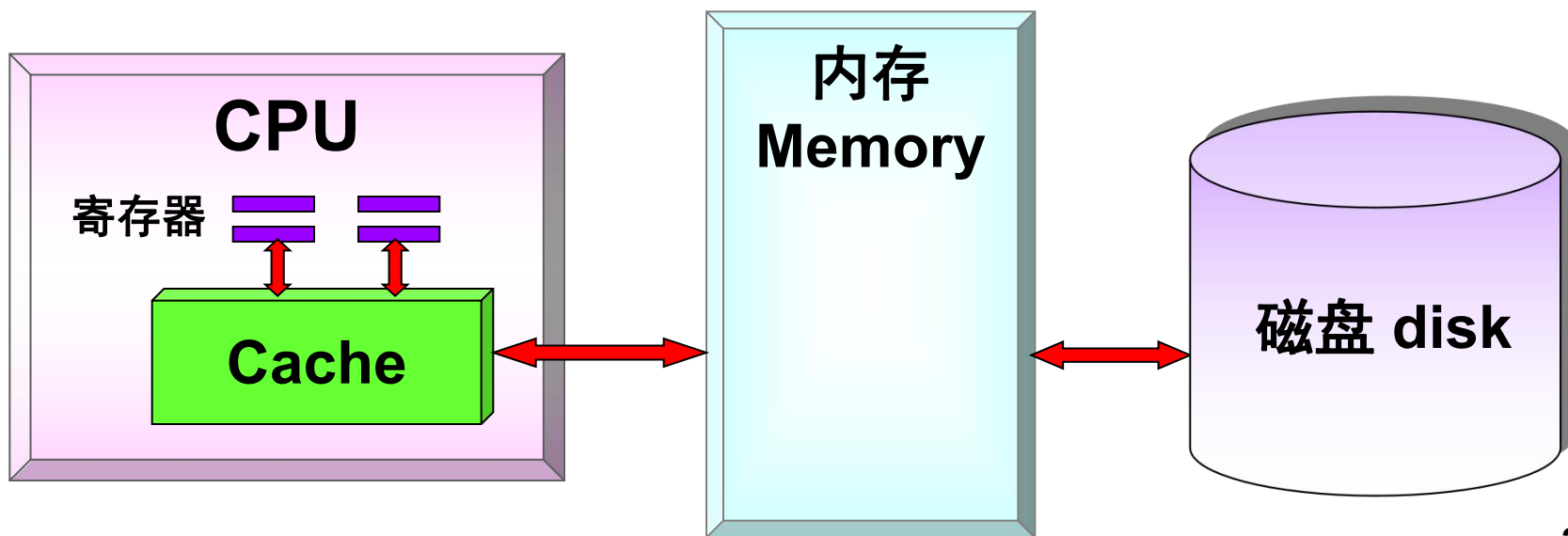
速度、单价和  
CPU访问频率



# 1.5 存储器的层次结构



- **内存**：CPU能直接访问的唯一大型存储介质。
- 为**扩充内存容量**，用**磁盘和内存**实现虚拟内存。
- 为**加快内存访问速度**，CPU首先访问**Cache**，不命中时再访问内存且复制进Cache。Cache与内存比为4:1000时(如256KCache: 64M内存)，命中率可达90%。

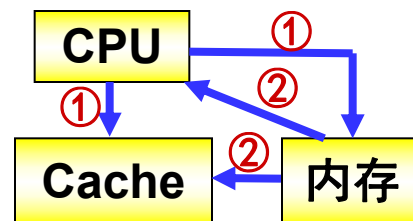
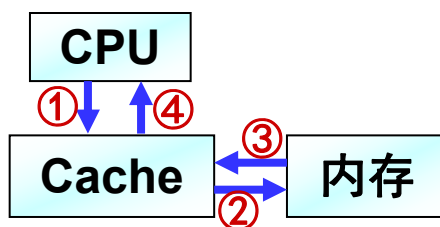


# 1.5 存储器的层次结构



- ❑ **“Cache—内存” 两级存储器**：当Cache命中率很高时，平均存取时间接近于Cache访问速度。
- ❑ 设忽略用于确定Cache是否命中的时间。Cache不命中时，数据先由内存送到Cache，CPU再访问Cache。
- ❑ 若  $T_1$  为Cache存取时间， $T_2$  为内存存取时间， $H$  为Cache命中率，则访问数据的平均存取时间为：  
$$H \times T_1 + (1 - H) \times (T_2 + T_1)$$
- ❑ 例：设访问一次Cache需 $0.1\mu s$ ，访问一次内存需 $1\mu s$ ，Cache命中率为95%，则访问一个字节的平均存取时间：  
$$0.95 \times 0.1\mu s + 0.05 \times (1\mu s + 0.1\mu s) = 0.15\mu s$$

注：不同CPU访问Cache-内存时的方式不同。第一种在Cache不命中时数据从内存→Cache→寄存器；第二种则从内存直接送到寄存器。





# Question:



❏ 课后习题1.12: 一个存储器系统具有如下参数:

$$T_{\text{cache}} = 100\text{ns}, C_c = 0.01\text{分/位}$$

$$T_{\text{memory}} = 1200\text{ns}, C_m = 0.001\text{分/位}$$

❏ 问: 1MB的内存价格是多少?  $C_m \times 8 \times 10^6 = 80\text{元}$

❏ 若使用Cache技术, 1MB内存价格是多少?

$$C_c \times 8 \times 10^6 = 800\text{元}$$

❏ 若要求有效存取时间仅比Cache存取时间多10%, 则命中率H为多少?

$$1.1 \times T_c = H \times T_c + (1-H) \times (T_c + T_m) = T_c + (1-H) \times T_m$$

$$\Rightarrow 0.1 \times 100 = (1-H) \times 1200 \Rightarrow H = 1190/1200 = 99.2\%$$

# 1.5 存储器的层次结构



## ❑ 两级存储器有效的原因：程序的局部性原理

- 在一段小的时间间隔内，被访问过的某指令或数据，可能很快会被再次访问。——时间局部性
- 在一段小的时间间隔内，程序访问的地址空间往往集中在某个区域（簇）。——空间局部性
- 大多顺序执行；经常有循环；过程调用深度有限；数据常为数组、记录；不是所有代码都需执行。

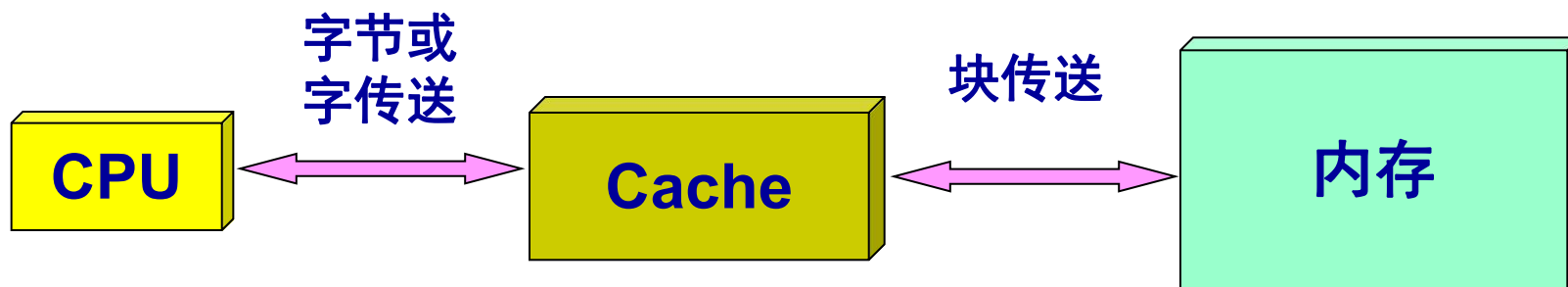
❑ 因此，让内存包含所有的（或大部分）指令和数据，而当前访问的“簇”包含在Cache中，可达到高命中率。

❑ 类似地，在虚拟内存中，由“Cache—内存—磁盘”构成三级存储器。

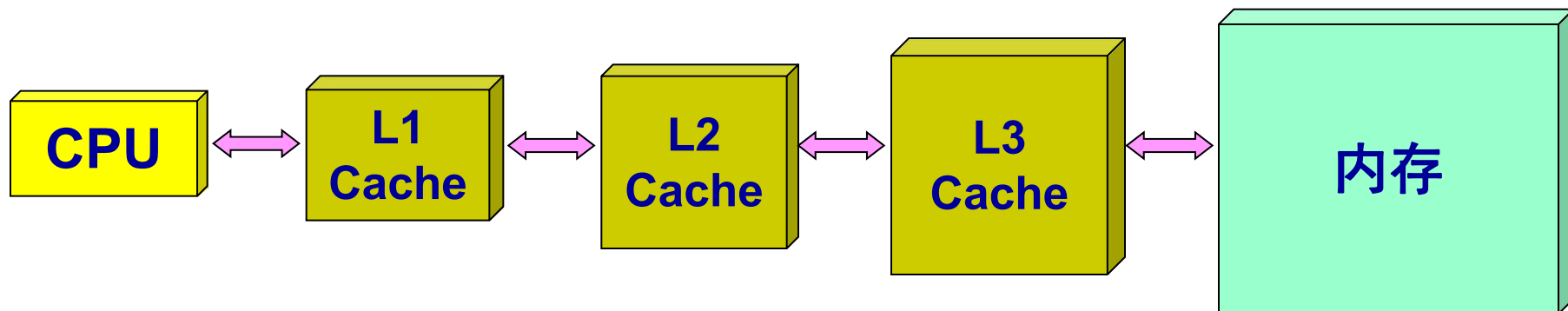
# 1.6 高速缓存



Cache位于CPU和内存之间，由硬件的Cache控制器管理，对操作系统和程序员不可见。



有些CPU支持多级Cache，如 L1，L2，L3 Cache。容量逐渐增大，速度逐渐减慢。



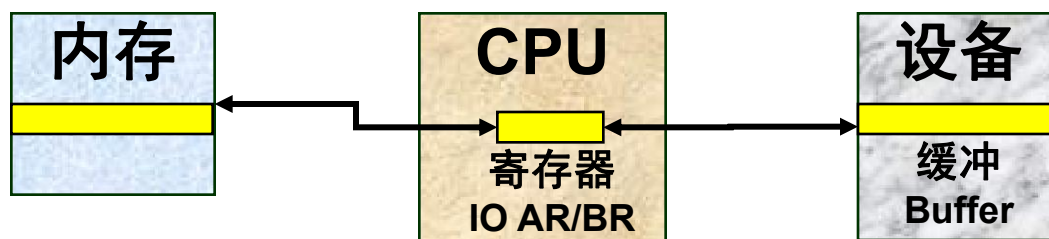
# 1.7 直接内存存取 — Direct Memory Access



较早的另外两种低效的I/O通信方式：

- **可编程I/O操作**：无中断。CPU发出I/O指令后，循环测试I/O设备的状态，并将数据送入内存。**CPU忙等I/O结束**，CPU和设备串行工作，资源利用率低。
- **中断驱动I/O**：CPU发出I/O指令后，可以去处理其它工作。设备每I/O完一批字(设备控制器的数据缓冲区长度)后，以**中断请求方式**通知CPU，CPU把这些字送入内存，然后设备再I/O下一批字。CPU和设备并行但仍低效。

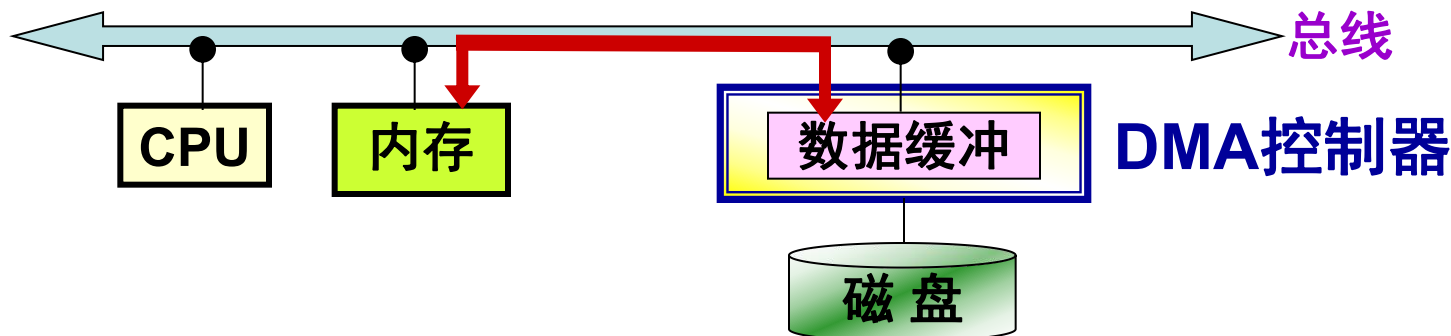
这两种方式都需要CPU主动干预内存和I/O设备间的数据传送，所传送的数据需通过CPU中转。



# 1.7 直接内存存取—DMA



- 在**DMA控制器**的控制下，**通过窃取总线周期**，在内存和设备间直接(不经过CPU)传送**整块数据**。
  - 在争用总线时，DMA控制器的优先级比CPU高；
  - DMA控制器不断地抢占总线(挪用CPU工作周期 *cycle stealing*)，在数据缓冲Buffer中和内存单元之间传送数据。结束时向CPU发出中断。
  - 仅在传送开始和结束时才需CPU干预。
- 当DMA传送时，CPU执行程序的速度稍微变慢。



# Question:



❏ 课后习题1.7：为什么DMA访问内存(或总线)的优先级要高于CPU访问内存(或总线)的优先级？

答：CPU欲访问内存时被挂起只是损耗一点时间。  
DMA在内存和设备间传输数据是以数据流方式，不能被打断，否则可能丢失数据，所以DMA的优先级更高。

# 1.8 多处理器和多核计算机组织结构



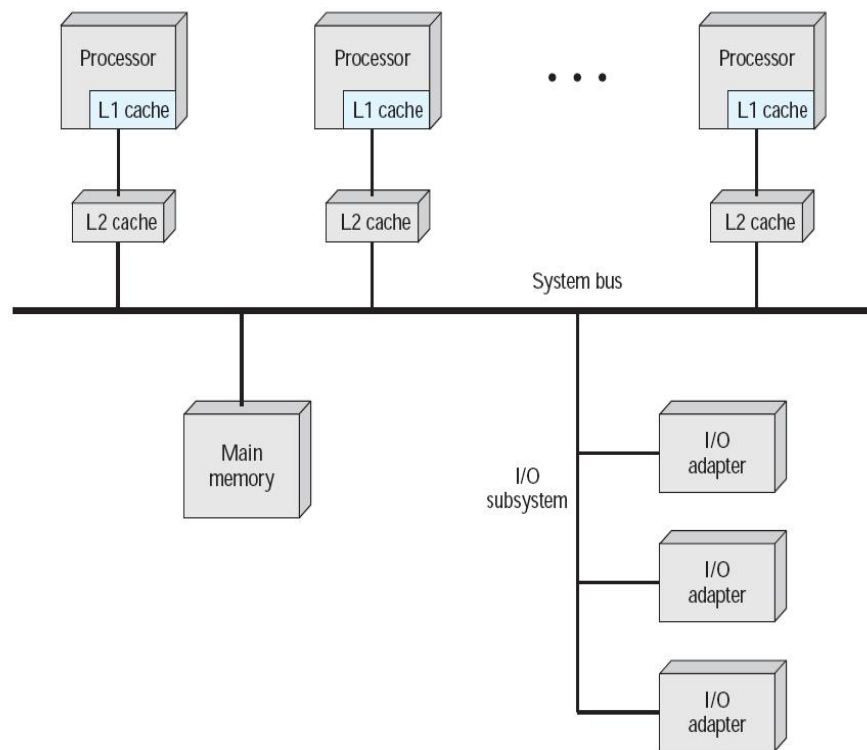
## 1.8.1 对称多处理器 Symmetric Multi-Processor

- 具有多个处理器，所有处理器地位和功能相同；
- 所有处理器通过总线相连，共享内存和I/O设备；
- 整个系统由一个操作系统管理和控制。

### SMP优点：

- 并行处理，性能好；
- 可用性好：单个CPU失效不会导致停机。
- 渐增性成长和可伸缩性：增加/减少CPU数量可提高/降低系统性能。

多核CPU就是一种SMP。



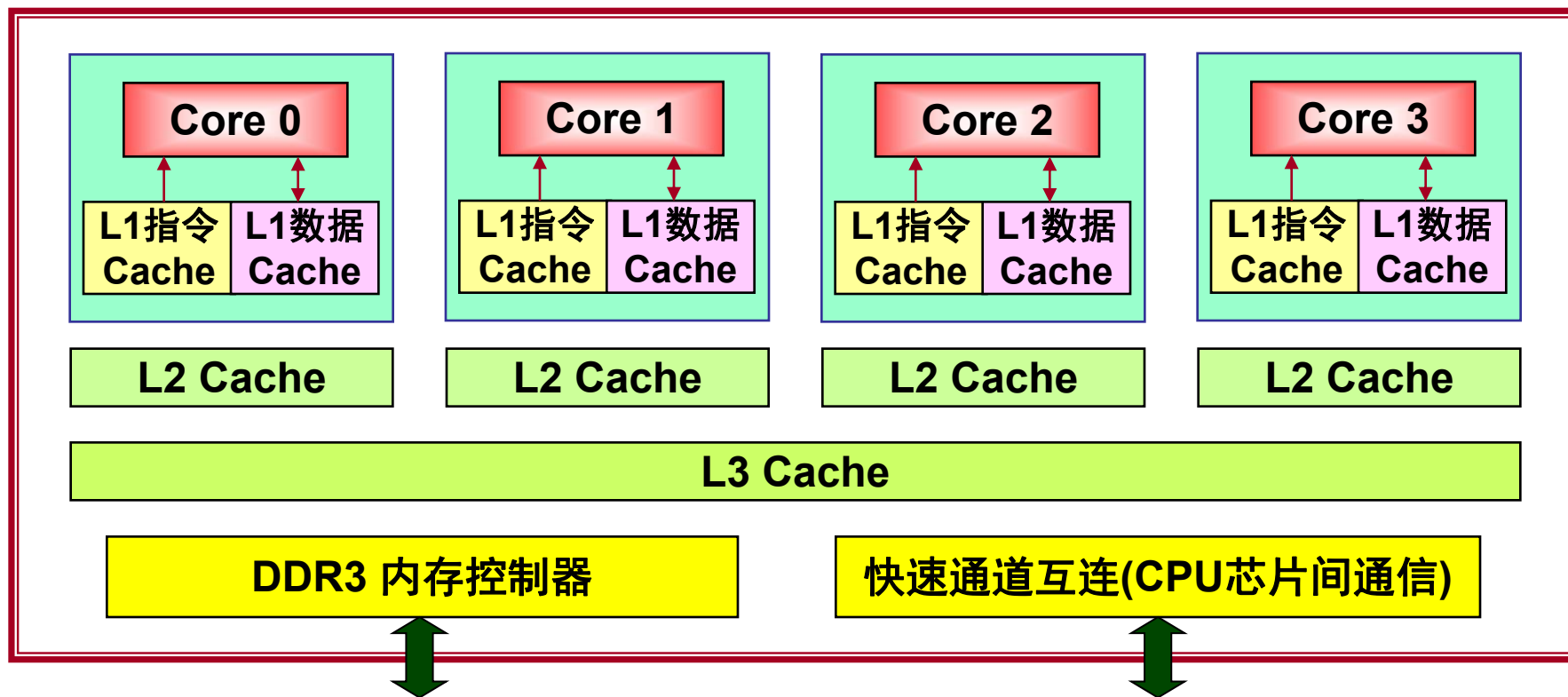


# 1.8.2 多核计算机—multicore



❏ 多核CPU：将多个CPU核心(Core)集成在一块硅芯片上。每个Core有独立的寄存器/ALU/控制器等。

❏ Intel Core i7 架构：64位四核CPU



# 作业：



■ 复习题：4, 5, 7

■ 习题：13

■ 作业不用抄题，标清题号即可。

■ 做在薄单行簿上。

■ 所有同学都要做，抽部分同学交。每次要看以前的作业是否已做。

■ 下次课时讲解作业，所有同学都带作业来即时更正。