



重庆大学
CHONGQING UNIVERSITY

计算机学院

COLLEGE OF COMPUTER SCIENCE

输入输出(I/O)控制

第六章 输入输出(I/O)控制

■ 何谓I/O控制

■ 对I/O操作的软、硬件管理

♣ 本章主要从硬件角度讨论I/O控制

♣ 从软件角度讨论I/O控制是操作系统的主要内容之一

- 作业管理
- 进程管理
- 存储管理
- 设备管理
- 文件管理

第六章 输入输出(I/O)控制

□面临的挑战之一——I/O墙

- CPU性能：每年增长 > 60%
- I/O系统的性能受到机械延迟的限制(磁盘I/O)，
每年的性能增长 < 10%
- I/O瓶颈
 - 削减了高速CPU的性能，
 - 缩短了CPU执行程序的有效时间



重庆大学
CHONGQING UNIVERSITY

计算机学院

COLLEGE OF COMPUTER SCIENCE

第 6.0 节

概 述

6.0 概述

- 输入输出子系统是连接主处理机和I/O设备的子系统
 - 是计算机系统中最具多样性和复杂性的组成部分
 - 最典型地反映了软件与硬件的相互结合
 - 其复杂性都隐藏在操作系统中

直接内存访问 (DMA) 是一种完全由硬件执行 I/O 交换的工作方式

6.0 概述

输入输出系统的发展概况

以CPU为中心

1. 早期阶段

分散连接



CPU 和 I/O 串行工作

程序查询方式

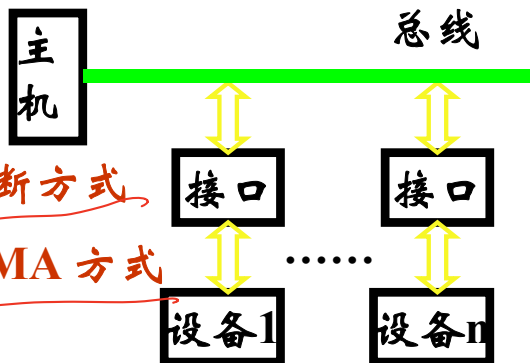
2. 接口模块和 DMA 阶段

总线连接

CPU 和 I/O 并行工作

中断方式

DMA 方式



3. 通道结构阶段

4. I/O 处理机阶段

通道

以主存为中心



6.0 概述

■ 输入输出系统的组成

1. I/O 软件

■ I/O 指令 CPU 指令的一部分

操作码	命令码	设备码
-----	-----	-----

■ 通道指令 通道自身的指令

指出数组的首地址、传送字数、操作命令、设备码
如 IBM/370 通道指令为 64 位

2. I/O 硬件

I/O 设备 I/O 接口

一个通道 → 多个设备控制器 → 多个设备

如：IBM/360 的一个通道连接 8 个设备控制器，一个设备控制器连接 8 台 I/O 设备

6.0 概述

■ I/O 与主机的联系方式

1. I/O 编址方式

■ 统一编址：存储器地址与I/O地址统一考虑，地址空间的一部分是存储器，另一部分是I/O，支持存储器操作的指令都可用于I/O操作

用取数、存数指令

■ 独立编址：存储器地址与I/O地址分开，CPU具有专用的I/O指令，系统总线中具有区别存储器读写和I/O操作的控制信号，以此区别地址总线的地址是存储器地址还是I/O地址

有专门的 I/O 指令

6.0 概述

■ I/O 与主机的联系方式

2. 设备选址

用设备选择电路识别是否被选中

3. 传送方式

■ 串行

■ 并行

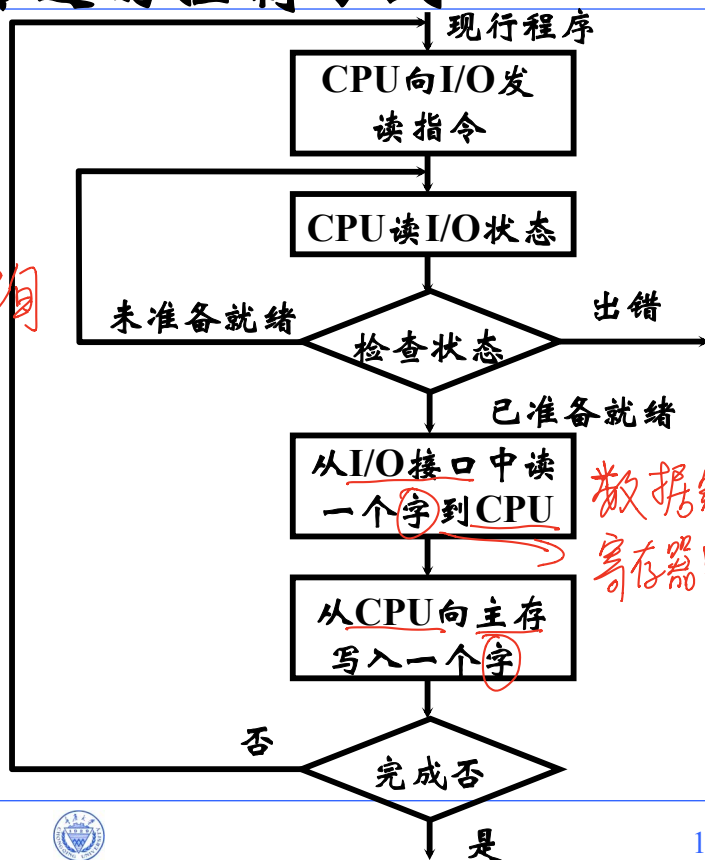
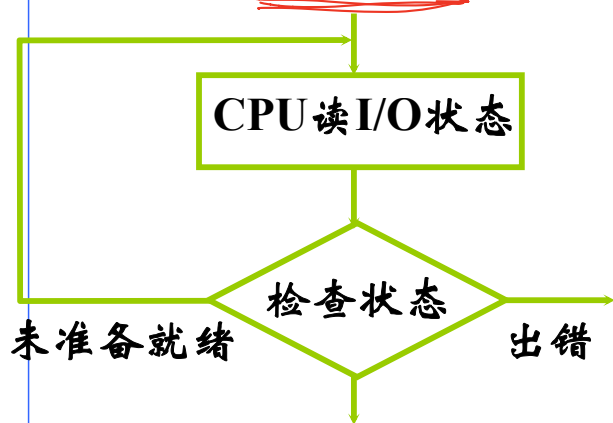
6.0 概述

I/O 与主机信息传送的控制方式

1. 程序查询方式

CPU 和 I/O 串行工作

踏步等待 重复查询



数据先放到寄存器中

6.0 概述

1. 程序查询方式 (Polling)

■ 特点

- 何时对何设备进行输入或输出操作完全受CPU控制
- CPU通过指令对设备进行测试才能知道设备的工作状态。设备空闲、准备就绪、正在忙等
- 数据的输入和输出都要经过CPU
- 用于连接低速外围设备，如终端、打印机等

■ 优点

- 灵活性好。可以很容易地改变各台外围设备的优先级

■ 缺点

- 实现处理机与外围设备并行工作困难

CPU 需要花费大量时间用于查询和等待，一段时间内只能和一台外设交换信息，效率较低

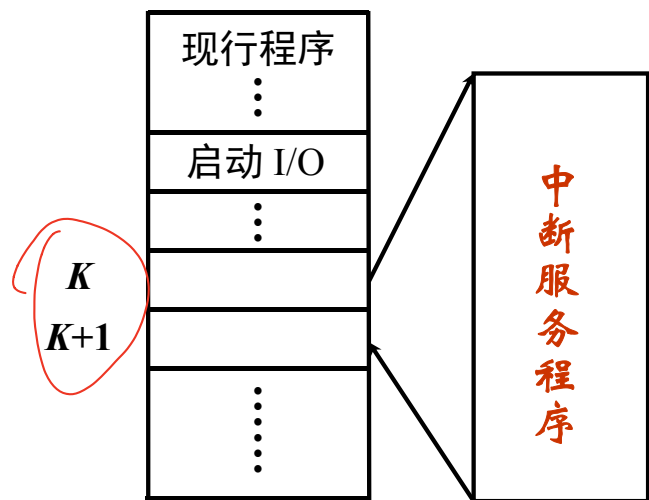
6.0 概述

2. 程序中断方式

I/O 工作 { 自身准备
与主机交换信息

CPU 不查询

CPU 暂停现行程序



CPU 和 I/O 并行工作

没有踏步等待现象

中断现行程序

6.0 概述

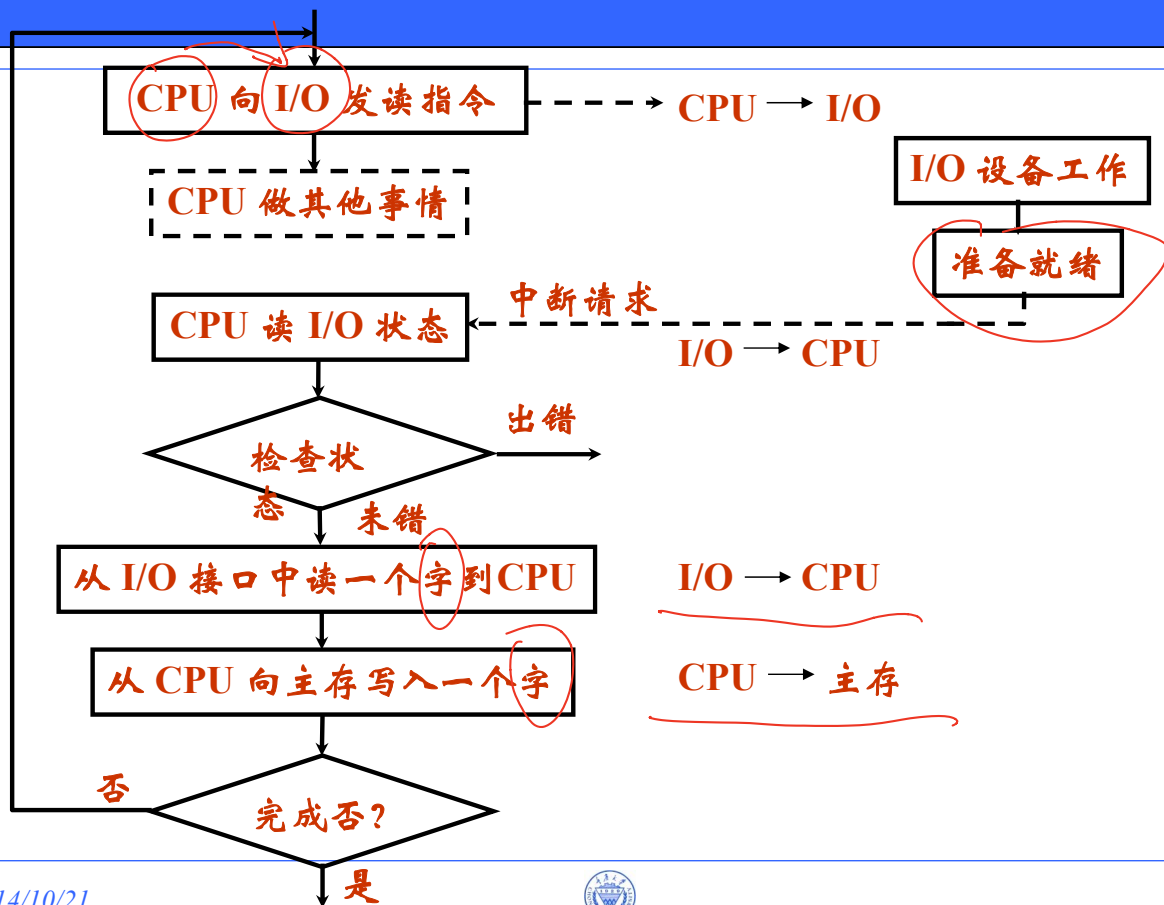
2. 程序中断方式

■ 特点

- ♣ CPU与外围设备能够并行工作
- ♣ 能够处理例外事件。例如，电源掉电、非法指令、地址越界、数据溢出、数据校验错、页面失效等
- ♣ 数据的输入和输出都要经过CPU
- ♣ 灵活性好
- ♣ 用于连接低速外围设备

在现代计算机系统中，中断输入输出方式的作用已经远远超出了为外围设备服务的范畴，成为现代计算机系统中非常重要的一个组成部分。

■ 程序中断方式流程



6.0 概述

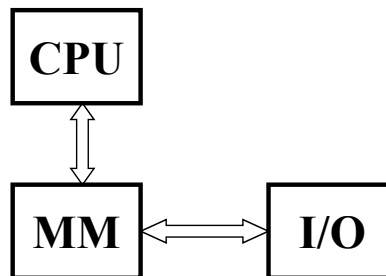
3. DMA 方式

主存和 I/O 之间有一条直接数据通道

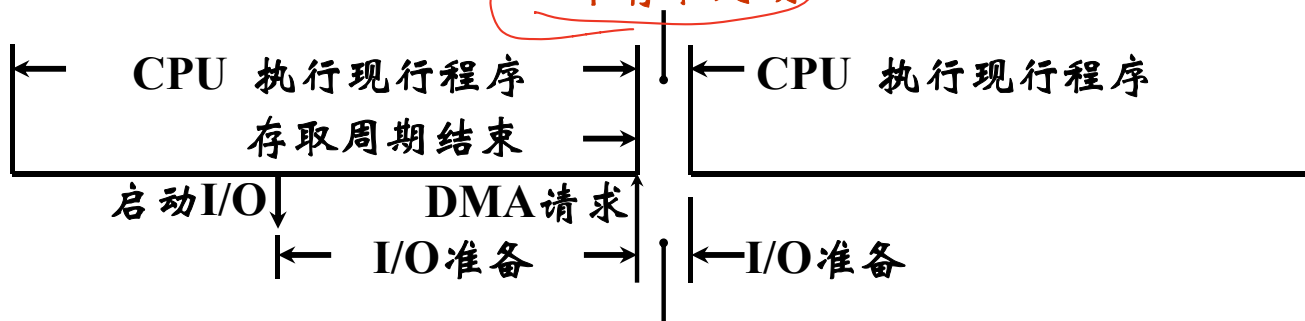
不中断现执行程序

周期挪用 (周期窃取)

CPU 和 I/O 并行工作



一个存取周期



实现 I/O 与主存之间的传送

6.0 概述

■ 特点

- 外围设备访问请求直接发往主存储器
- 不需要CPU做保存现场和恢复现场等工作
- 在DMA控制器中，需要设置数据寄存器、设备状态或控制寄存器、主存地址寄存器、设备地址寄存器和数据交换个数计数器
- 在DMA方式开始和结束时，需要处理机进行管理
 - 在DMA方式开始之前对DMA控制器进行初始化。传送主存缓冲区首地址、设备地址、数据块的长度等，并启动设备开始工作
 - 在DMA方式结束之后，向CPU申请中断，对数据缓冲区进行后处理
- 数据的传送过程不需要CPU的干预

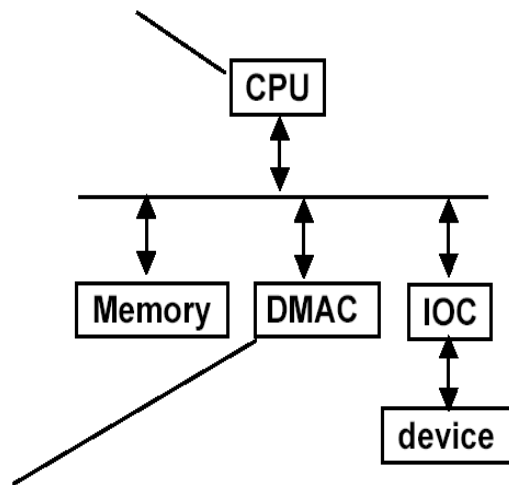
6.0 概述

■ DMA方式工作流程

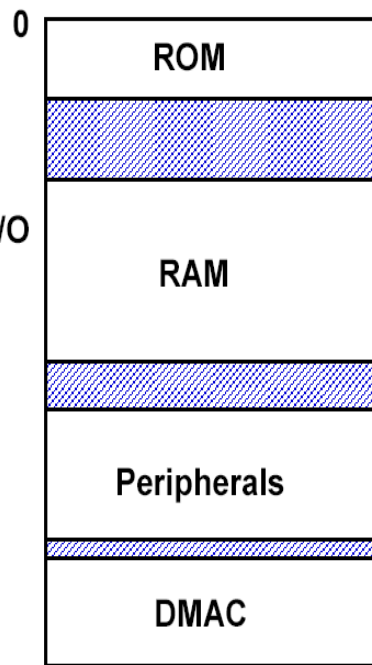
CPU向DMAC发送开始地址、方向；
然后，发射“开始”命令。

1 次中断服务 48 μsec

0.0001 秒的CPU时间

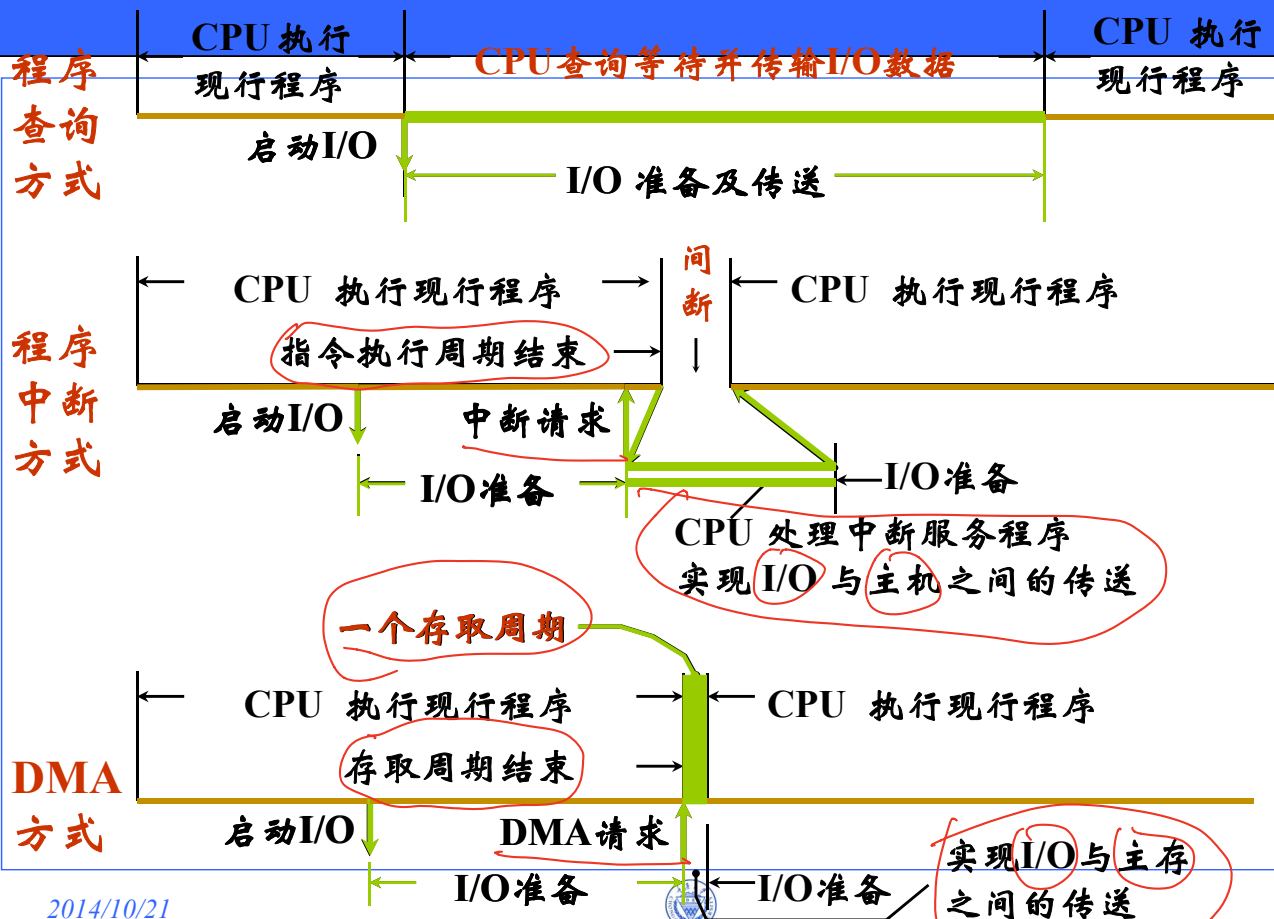


存储器映射 I/O



DMAC 向外设控制器提供握手信号，
向存储器提供存储地址和握手信号

■ 三种方式的 CPU 工作效率比较





第 6.4 节

中断及程序中中断控制传送

6.4 中断及程序中中断控制传送

□ 知识点

- 理解中断的有关问题
- 了解中断系统的结构组成
- 理解程序中中断控制传送及其接口

□ 重点

- 掌握中断的基本概念、原理和中断技术

6.4 中断及程序中中断控制传送

- ❑ 中断系统是现代计算机系统中最重要的一部分之一。只有采用中断系统，才能使处理机的各个部件之间、处理机与外部世界之间真正并行工作
- ❑ 中断系统需要硬件和软件共同来实现

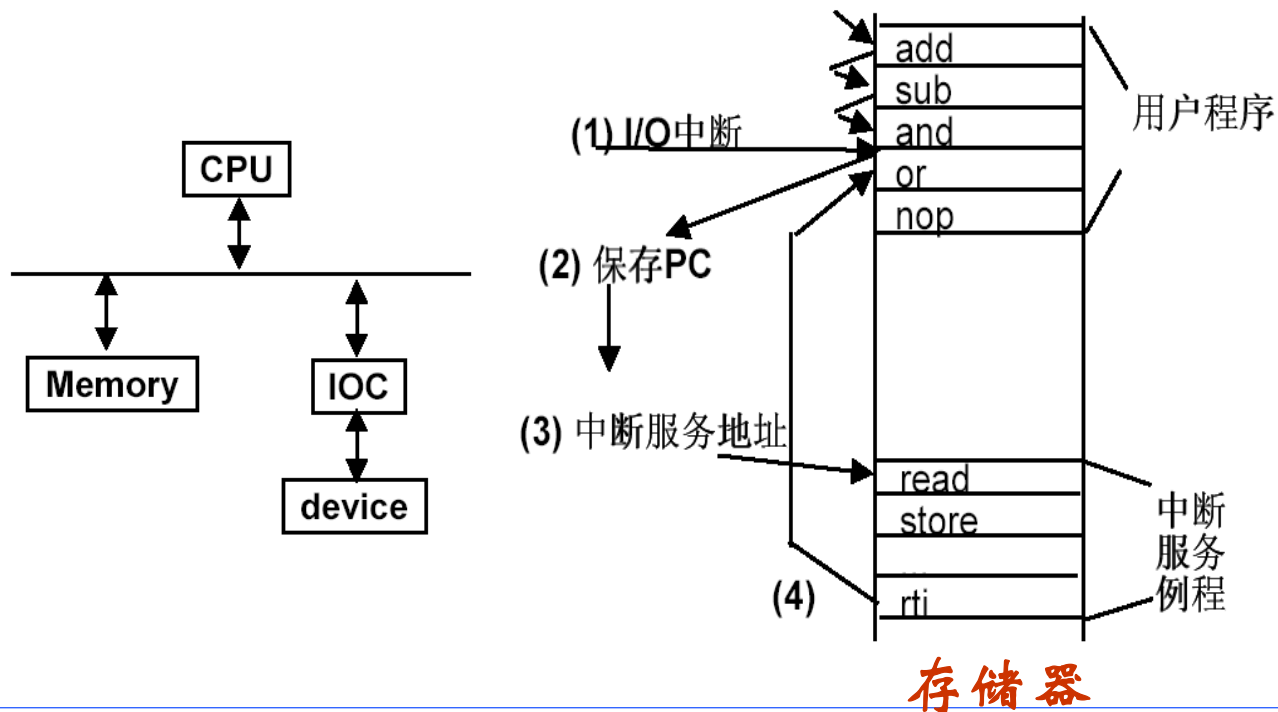
6.4.1 中断的有关问题

■ 3个名词术语

1. 何谓**中断**：CPU暂时中止现行程序，转去执行“处理随机发生的紧急事件或特殊请求”的程序，处理完后自动返回被中止的程序继续运行的功能
2. 何谓**中断系统**(Interrupt System)：计算机实现中断功能的软硬件总称
3. **中断源**(Interrupt Source)：引起处理机中断的事件，称之为中断源

6.4.1 中断的有关问题

■ 中断的概念

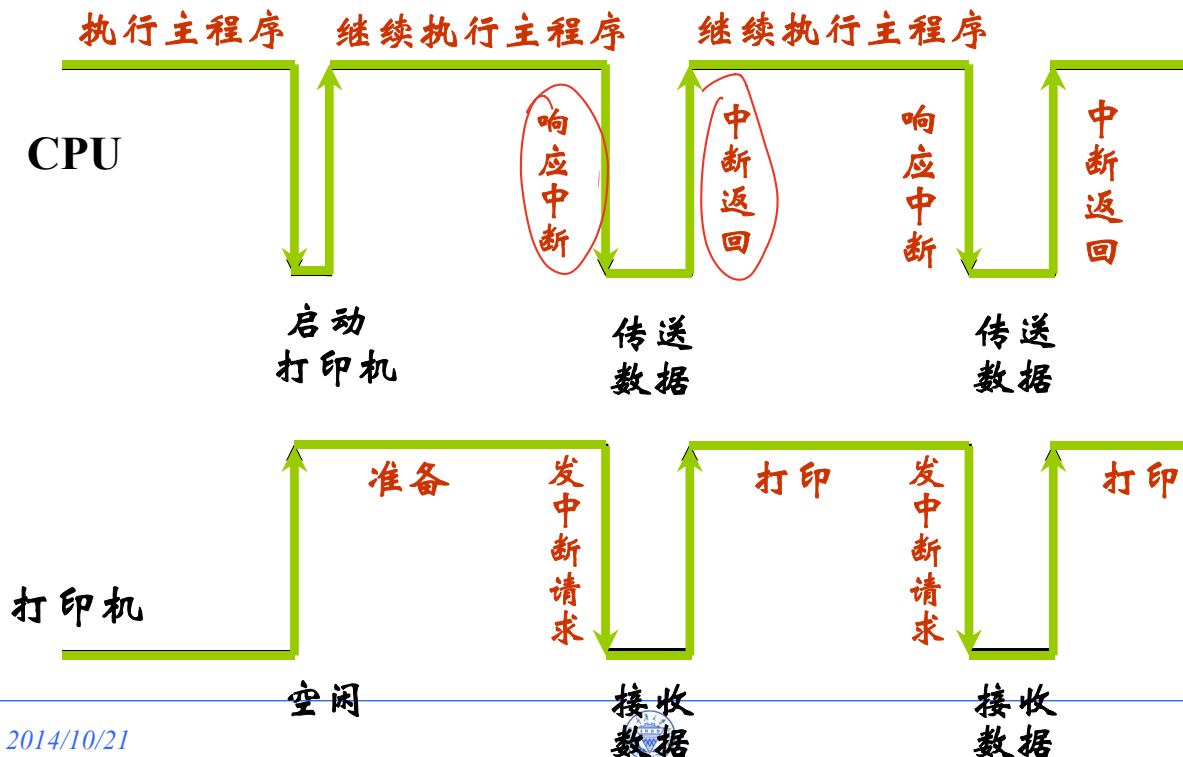


6.4.1 中断的有关问题

I/O 中断的产生

■ 以打印机为例

CPU 与打印机并行工作



6.4.1 中断的有关问题

■ 中断的作用

1. 实现主机与外设，外设与外设的并行工作
2. 实现故障的处理：如掉电、除数为“0”，非法操作码、打印机无纸等
3. 实现多道程序和分时操作：可由中断系统进行程序切换
4. 实现实时控制：受控对象靠中断把有关参数(炉温、湿度、速度等)和反馈信号送主机
5. 实现人机联系与通信：如了解工作状态等

6.4.1 中断的有关问题

■ 中断的类型

1. 自愿中断和强迫中断

■ 自愿中断：亦称程序自中断，是在程序中预先安排的由广义指令引起的中断。这种中断是预知的、可重现的

♣ 何谓广义指令：由一条特殊的转移指令和若干参数组成

如：自愿进管指令，它为操作系统编写通道程序提供必要信息

■ 强迫中断：不是在程序中预先安排的，而是随机产生的中断。这种中断是不可预知的

6.4.1 中断的有关问题

■ 中断的类型

2. 内中断和外中断

■ **内中断**：由主机内部硬件和软件原因引起的中断

♣ **硬件故障中断**：电源掉电、各种校验错等

♣ **陷阱(Trap)**：又称程序性中断。由于程序本身运行的原因引起的中断。如：非法操作码、阶码溢出、除数为“0”、栈溢出、缺页、地址越界等

注意：操作系统中的异常，包括自愿中断和陷阱

■ **外中断**：主机以外的部件引起的中断

如：I/O传输中断、I/O故障中断、定时中断等

6.4.1 中断的有关问题

■ 中断的类型

3. 单重中断系统和多重中断系统

■ 单重中断系统：执行中断服务程序时，不能再响应其它中断的系统称单重中断系统

■ 多重中断系统：执行中断服务程序时，还可响应更高优先级中断的系统称多重中断系统——中断嵌套

6.4.1 中断的有关问题

■ 中断的类型

4. 可屏蔽中断和不可屏蔽中断

■ 可屏蔽中断：可不响应或暂不响应，或有条件的响应的中断

■ 不可屏蔽中断：必须立即处理的、不能回避和禁止的中断。电源中断是具有最高优先级的不可屏蔽中断

程序方法有选择性地封锁部分中断，使之不发出中断请求，而允许其余中断发出中断请求并得到响应

6.4.2 中断系统的结构组成

■ 中断系统的任务（中断应解决的问题）

1. 接收并保存中断请求（硬件完成）
2. 进行中断判优（软、硬件均可）
3. 实施中断响应（硬件完成）
4. 实现中断处理（软件完成）
5. 返回被暂停(中止)的程序（软件完成IRET）

6.4.2 中断系统的结构组成

1. 接收并保存中断请求的寄存器(3个)

■ 中断寄存器

寄存随机产生的中断，每个中断源需设1位中断触发器，所有中断源的中断触发器组成中断寄存器。触发器为“1”表示有中断，“0”表示无中断。称中断寄存器的内容为中断编码或中断字。不可屏蔽中断由中断触发器发出中断请求

■ 中断请求寄存器

每个可屏蔽中断源还设1个中断请求触发器，所有可屏蔽中断源的中断请求触发器组成中断请求寄存器。触发器为“1”表示有中断请求，“0”表示无中断请求。可屏蔽中断由中断请求寄存器发出中断请求

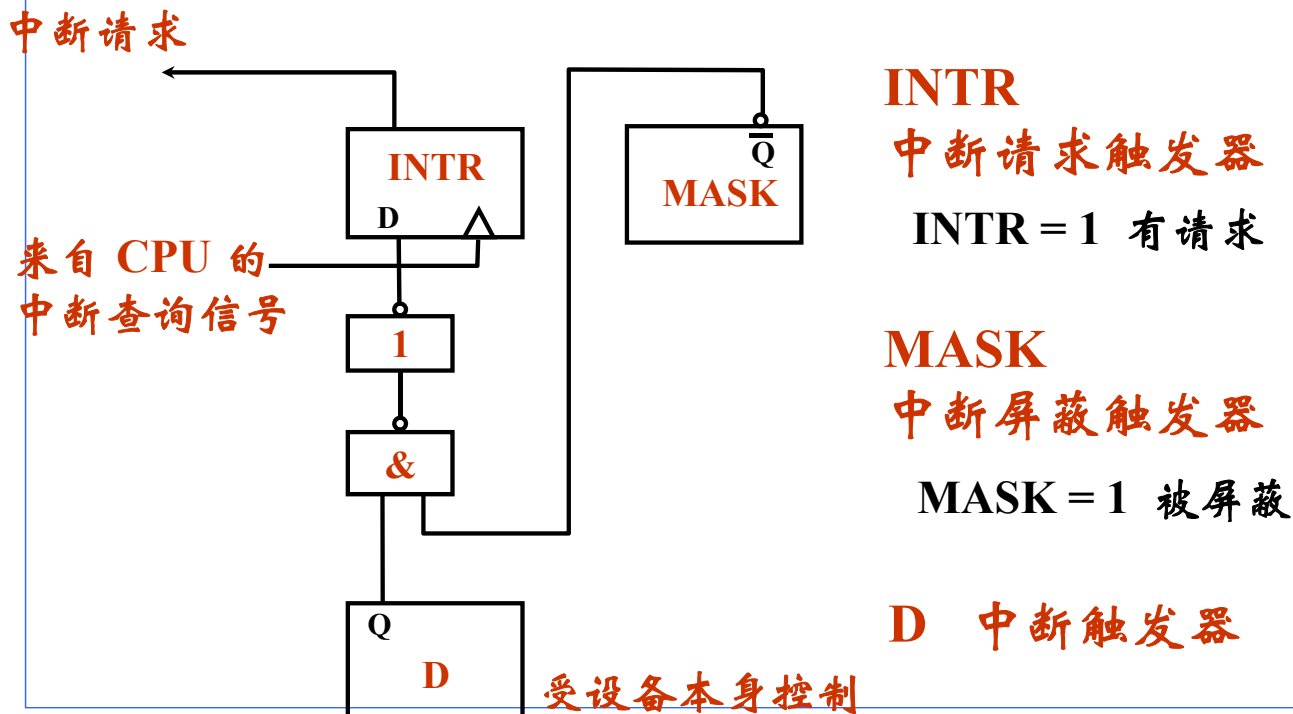
6.4.2 中断系统的结构组成

1. 接收并保存中断请求的寄存器

■ 中断屏蔽寄存器

每个可屏蔽中断源又设1个中断屏蔽触发器，所有可屏蔽中断源的中断屏蔽触发器组成中断屏蔽寄存器。功能是封锁相应中断触发器发出中断请求。称其内容为中断屏蔽码。“1”表示封锁相应位的中断请求，“0”表示不封锁即允许相应位发出中断请求

1. 接收并保存中断请求的寄存器



6.4.2 中断系统的结构组成

2. 中断判优

■ 几个名词和术语

♣ 中断优先权INTP(INTerrupt Priority): 中断响应的先后次序

♣ 中断排队: 按中断的性质和请求处理的轻重缓急, 给中断源排出中断响应的先后次序

♣ 中断判优: 通过中断排队, 从中选出需优先处理的中断源

注意:中断优先权在中断系统设计时, 就已考虑好。一般在程序运行中不变, 但也有个别中断的优先权可在程序运行中由程序员通过改变屏蔽码确定或变更。

6.4.2 中断系统的结构组成

2. 中断判优

■ 中断源分级

- ♣ 处于同一级的中断，其优先权为左高右低
- ♣ 不同级的中断源由系统决定级别的高低

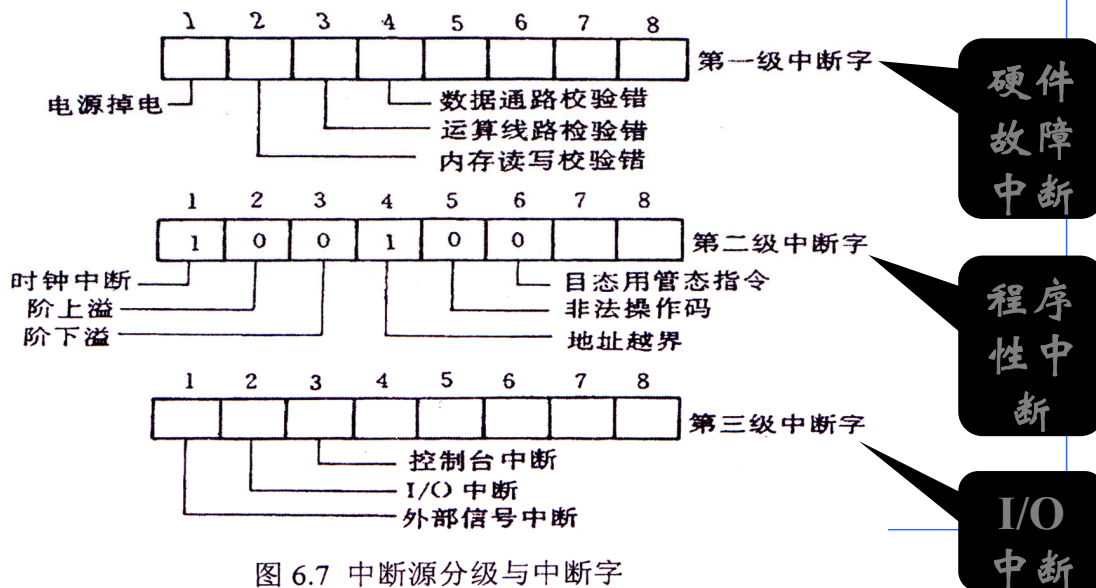


图 6.7 中断源分级与中断字

6.4.2 中断系统的结构组成

2. 中断判优

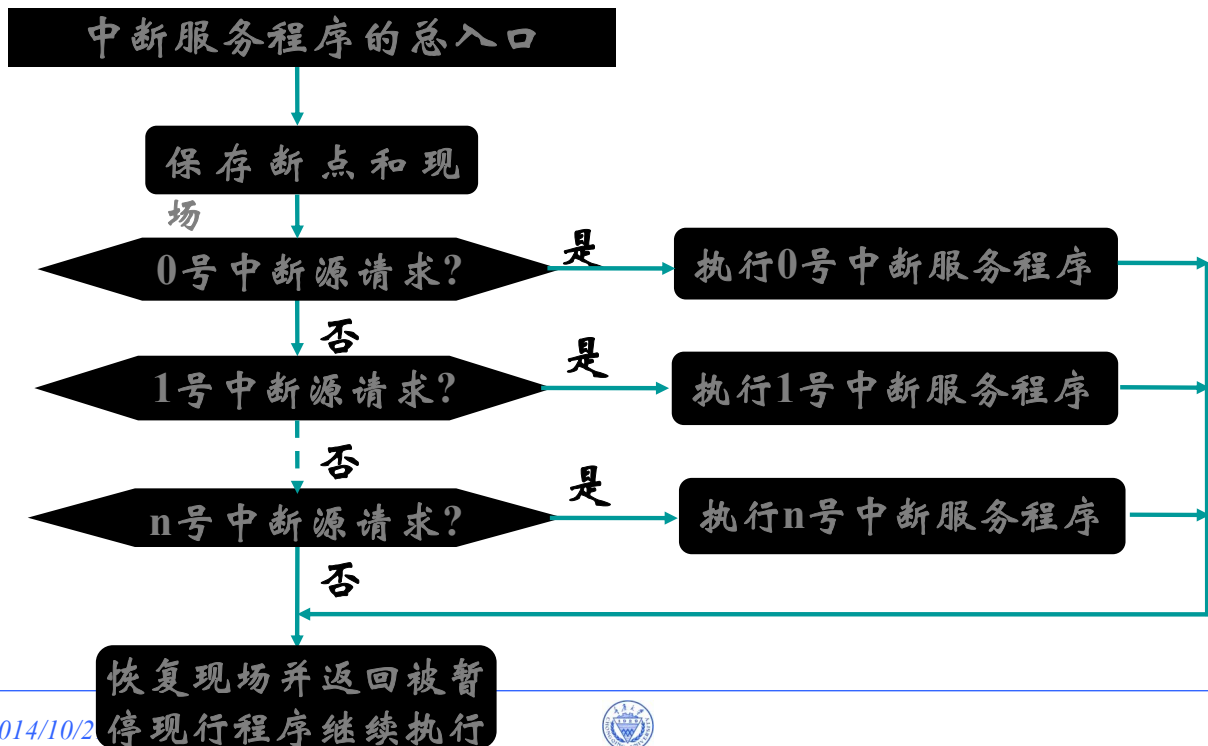
1) 硬件判优线路

- ♣ 最左判优电路

- ♣ 链型排队电路

6.4.2 中断系统的结构组成

2) 软件判优：软件判优是**通过执行中断查询程序寻找优先中断源**，而后**转入相应的中断服务程序**



6.4.2 中断系统的结构组成

3. 中断响应

1) 中断响应的条件

♣ CPU允许中断，通常设一个中断允许触发器INT实现

- 中断允许触发器INT为1，允许响应中断
- INT为0，不允许响应中断
- 专设两条指令：开中断指令，置“1”INT；关中断指令，清“0”INT

♣ CPU的一条指令执行完毕，且无更紧迫的任务或事件

如无DMA传送、掉电中断

♣ 中断源有中断请求

6.4.2 中断系统的结构组成

3. 中断响应

2) 中断响应完成的任务

- ♣ 暂停现行程序
- ♣ 进入中断服务(处理)程序

注意：中断响应的任务，通常由**中断隐指令**实现。

♣ **中断隐指令：**响应中断时才由硬件产生以便完成中断响应的各项工作，因它象一条指令，但CPU指令系统中又无此指令，故称中断隐指令

♣ 为实现中断隐指令的功能，有的CPU控制器特设一个**中断(PI)周期**。只要CPU响应中断，执行周期结束时，不是进入取指周期，而是进入中断周期，中断周期执行完后，才进入取指周期

6.4.2 中断系统的结构组成

3. 中断响应

3) 暂停现行程序时需要完成的工作

- ♣ **关中断**：进入不可再次响应中断的状态，以便保证保存中断现场的正确性
 - ♣ **保存断（返回）点**：保存程序计数器的内容，即保存被暂停程序要执行的第一条指令的地址
 - ♣ **保存硬件的其它现场**：保存程序状态字中的Flag、中断码、条件码等
 - ♣ **撤消当前正被响应的中断请求**，防止一次中断被多次响应
- 上述四项工作必须是由中断隐指令完成

硬件完成

6.4.2 中断系统的结构组成

3. 中断响应

4) 进入中断服务程序的方法

寻找中断服务程序的入口地址

a) **硬件法**：亦称**向量中断法**。由中断隐指令完成

♣ **中断向量表**：所有中断服务程序入口地址组成的一维表格，存放在一段连续的存储区域中

♣ **中断向量**：所有中断服务程序入口地址的有序集合

♣ **向量地址**：访问中断向量表所需地址，中断指针

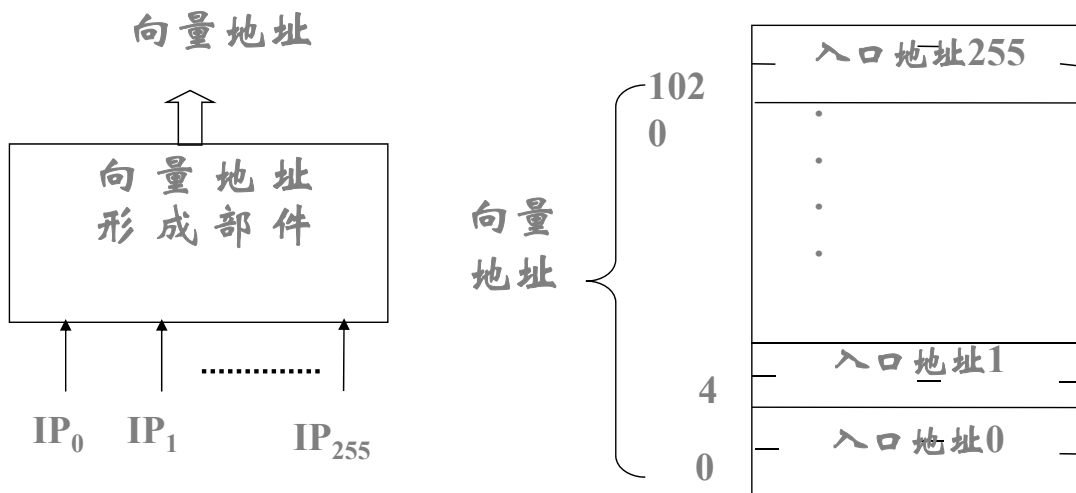
经典的向量中断法：响应中断时，由硬件自动产生相应中断源的向量地址。据此访问中断向量表，从中取出中断服务程序的入口地址，转入中断服务程序

6.4.2 中断系统的结构组成

4) 进入中断服务程序的方法

✿ 向量地址产生方法

- ① 中断响应时,由硬件直接产生与中断源对应的向量地址



6.4.2 中断系统的结构组成

3. 中断响应

4) 进入中断服务程序的方法

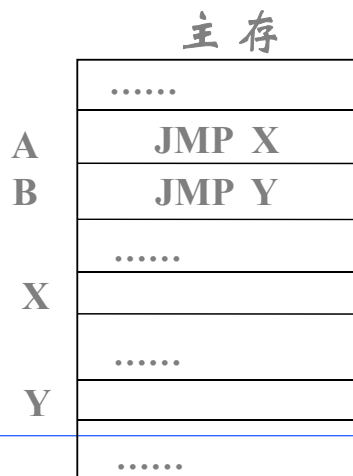
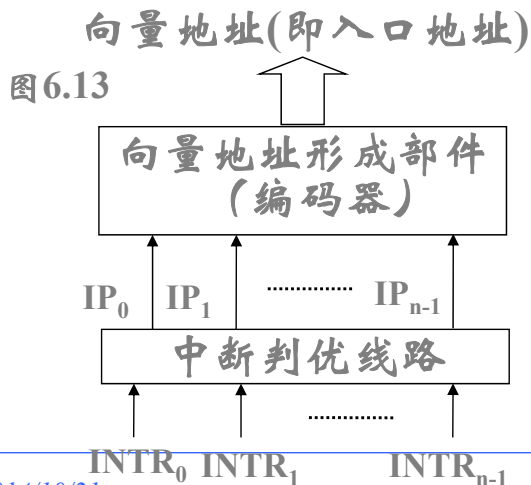
♣ 向量地址产生方法

- ② 中断响应时，硬件产生的只是相对于中断向量表始地址的偏移量。而向量表始地址已存入向量表基址寄存器，向量地址为该寄存器内容与偏移量之和，中断向量表可存放在主存的任何位置
- ③ 中断响应时，硬件产生与中断源对应的向量地址的低位，而向量地址的高位存放在CPU中断向量寄存器中，二者拼接形成向量地址，中断向量表可存放在主存的任何位置

6.4.2 中断系统的结构组成

❖ 向量地址产生方法

- ④ 在具有多根中断请求线的计算机系统中，可通过对请求线编码产生各中断源的向量地址（见图6.13），该向量地址就是相应中断服务程序的入口地址
- ⑤ 中断响应时，硬件直接产生与中断源对应的向量地址，但该地址中存储的不是中断服务程序的入口地址，而是一条无条件转移到中断服务程序的指令



6.4.2 中断系统的结构组成

3. 中断响应

4) 进入中断服务程序的方法

b) 软件法：中断响应后，转到一个处理中断的总入口，通过执行判优先中断源的程序，找到优先中断源。然后转入相应中断服务程序。**即判优和中断响应合并为一步（见图6.10）**

6.4.2 中断系统的结构组成

4. 实现中断处理

- 执行中断服务(处理)程序
- 对每一个中断源都有一个中断服务程序, 执行中断服务程序, 就是对相应紧急事件的处理

5. 返回被暂停的程序(即中断返回)

- 中断服务程序最后执行的一条指令就是中断返回指令(**IRET**)
- 功能: 返回被暂停的程序

6.4.2 中断系统的结构组成

■ 中断服务程序的组成

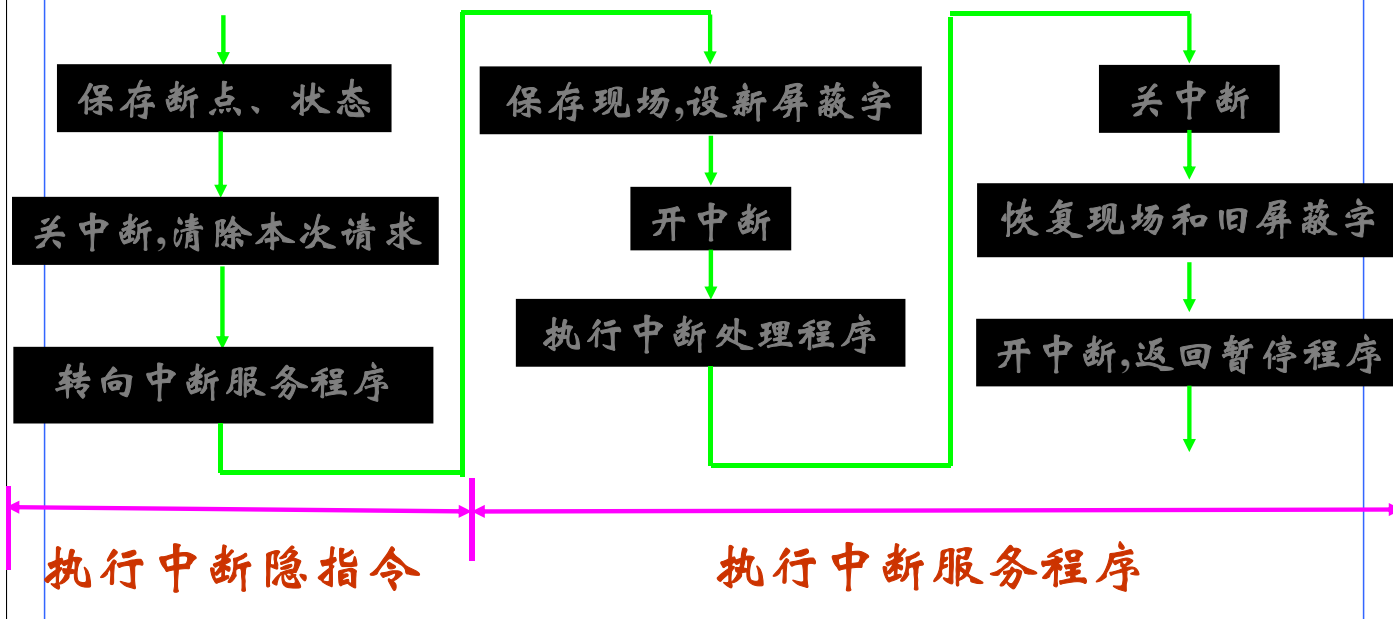
■ 中断隐指令：完成保存断点、关中断、保存Flag、撤消本次中断请求，进入中断服务程序

■ 中断服务程序的组成

- ① 保存现场：保存凡中断服务程序用到的寄存器
- ② 屏蔽比本级低的中断：送新的屏蔽码
- ③ 开中断：允许响应比本级更高级的中断
- ④ 实现对突发事件的处理：执行相应中断源的处理程序
- ⑤ 关中断：不允许响应任何可屏蔽中断
- ⑥ 恢复现场：将保存的寄存器内容予以恢复
- ⑦ 开中断：允许响应中断请求
- ⑧ 返回被暂停的程序：中断服务程序最后一条是中断返回指令,系统恢复到被中断程序运行状态

6.4.2 中断系统的结构组成

□ 中断响应和中断处理过程



6.4.2 中断系统的结构组成

■ 中断屏蔽技术与多重中断的实现

1) 中断屏蔽技术

■ **硬件**：设置屏蔽触发器及其相关电路

■ **软件**：依中断源的**优先级**为每个中断源**预先**设1个中断屏蔽码。屏蔽码与中断源的**优先级别一一对应**

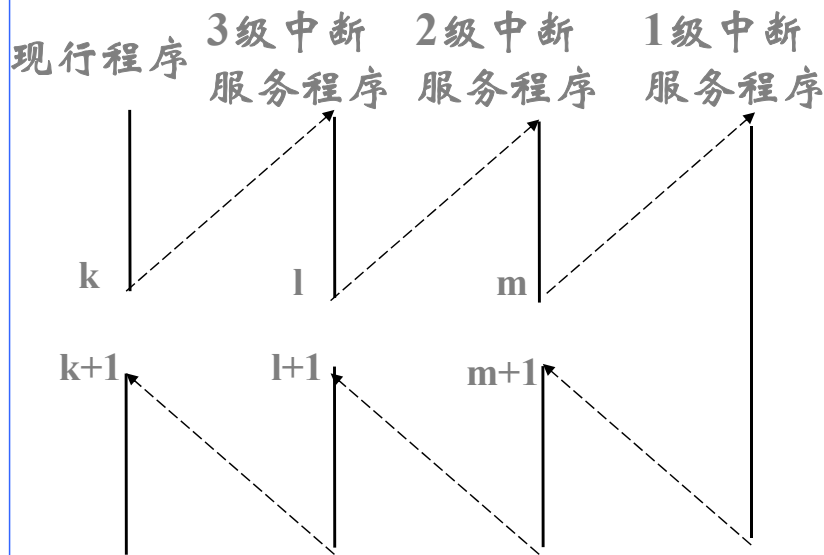
中断源序号	优先级	中断屏蔽码
1	1	11111111
2	2	01111111
3	3	00111111
4	4	00011111
5	5	00001111
6	6	00000111
7	7	00000011
8	8	00000001

进入中断服务程序保护现场之后，通过屏蔽指令送新的屏蔽码达到中断屏蔽目的，实现了多重中断。

6.4.2 中断系统的结构组成

■ 中断屏蔽与多重中断的实现

2) 多重中断的实现



■ 借助中断屏蔽寄存器和屏蔽码，通过屏蔽指令实现多重中断

1级中断的屏蔽码

11111111

2级中断的屏蔽码

01111111

3级中断的屏蔽码

00111111

6.4.2 中断系统的结构组成

■ 中断的全过程（小结）

■ 中断请求

■ 中断判优

■ 中断响应

■ 中断处理

■ 中断返回

} 软件判优时两步合为一步

} 软件实现

6.4.2 中断系统的结构组成

■ 中断服务程序流程

1. 中断服务程序的流程

(1) 保护现场

{	程序断点的保护	中断隐指令完成
	寄存器内容的保护	进栈指令

(2) 中断服务

对不同的 I/O 设备具有不同内容的设备服务

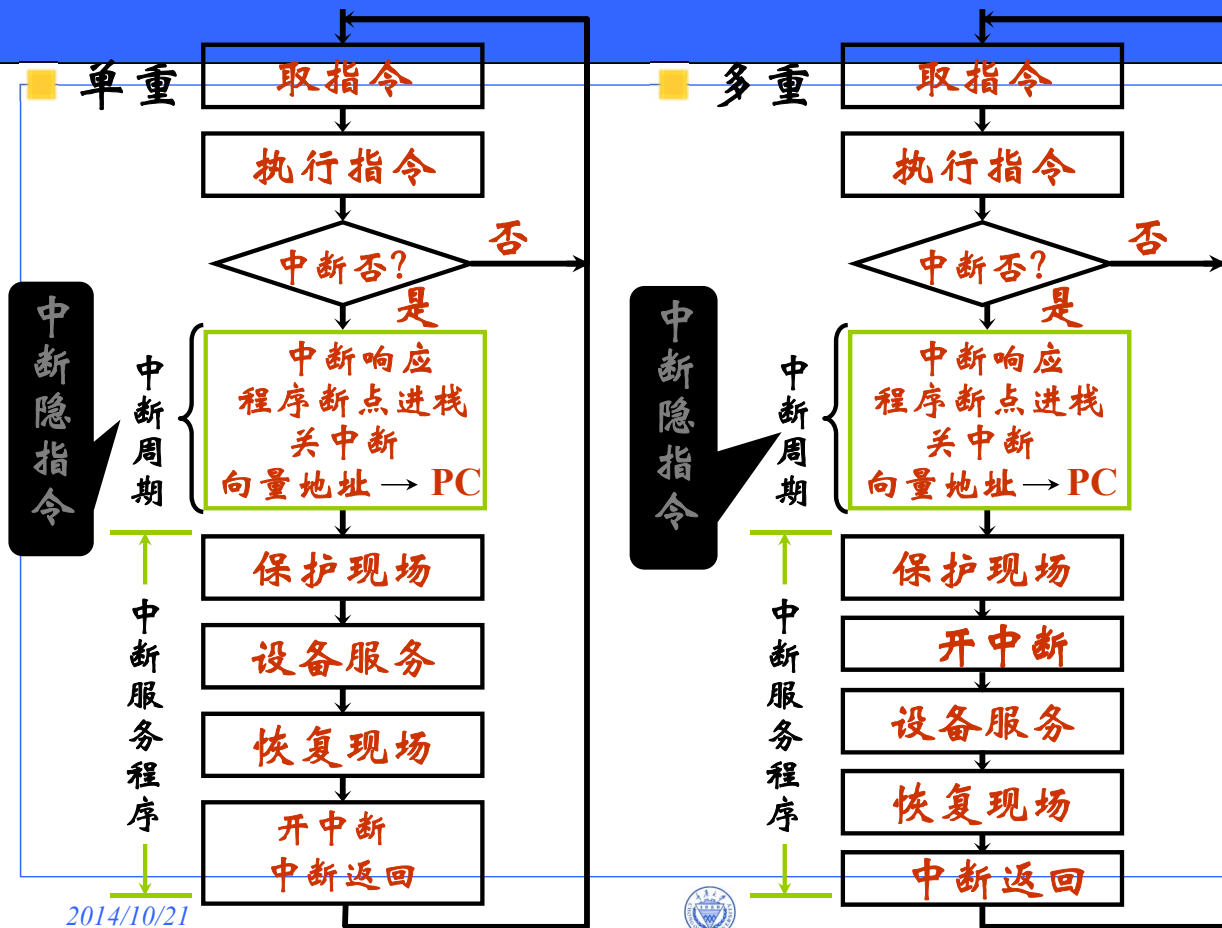
(3) 恢复现场

出栈指令

(4) 中断返回

中断返回指令

2. 单重中断和多重中断的服务程序流程



例：有四个中断源 D_1 、 D_2 、 D_3 和 D_4 ，它们的中断优先级从高到低分别是1级、2级、3级和4级。这些中断源的正常中断屏蔽码和改变后的中断屏蔽码见下表。每个中断源一位，共4位屏蔽码。

中断源名称	中断优先级	正常中断屏蔽码				改变后的中断屏蔽码			
		D_1	D_2	D_3	D_4	D_1	D_2	D_3	D_4
D1	1	1	1	1	1	1	0	0	0
D2	2	0	1	1	1	1	1	0	0
D3	3	0	0	1	1	1	1	1	0
D4	4	0	0	0	1	1	1	1	1

解：

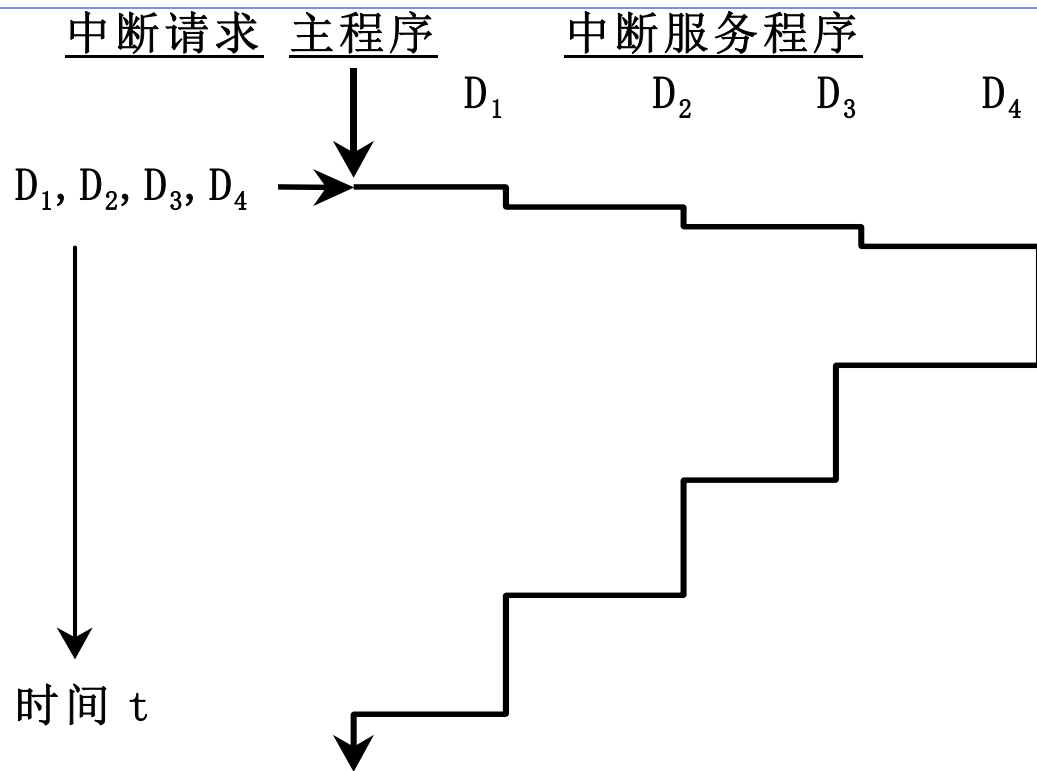
如果4个中断源都使用正常的中断屏蔽码，处理机的中断服务顺序将严格按照中断源的中断优先级进行。

如果改变中断屏蔽码，当 D_1 、 D_2 、 D_3 和 D_4 这4个中断源同时请求中断服务时，处理机实际为各个中断源服务的先后次序就会改变。

处理机响应的顺序是 D_1 、 D_2 、 D_3 、 D_4

实际服务的顺序是 D_4 、 D_3 、 D_2 、 D_1







6.5

内存直接存取(DMA)

6.5 内存直接存取(DMA)

□ 知识点

- DMA概述
- DMA接口的基本组成
- DMA的工作过程
- DMA传送接口类型

□ 重点

- 掌握DMA的基本概念及其工作过程

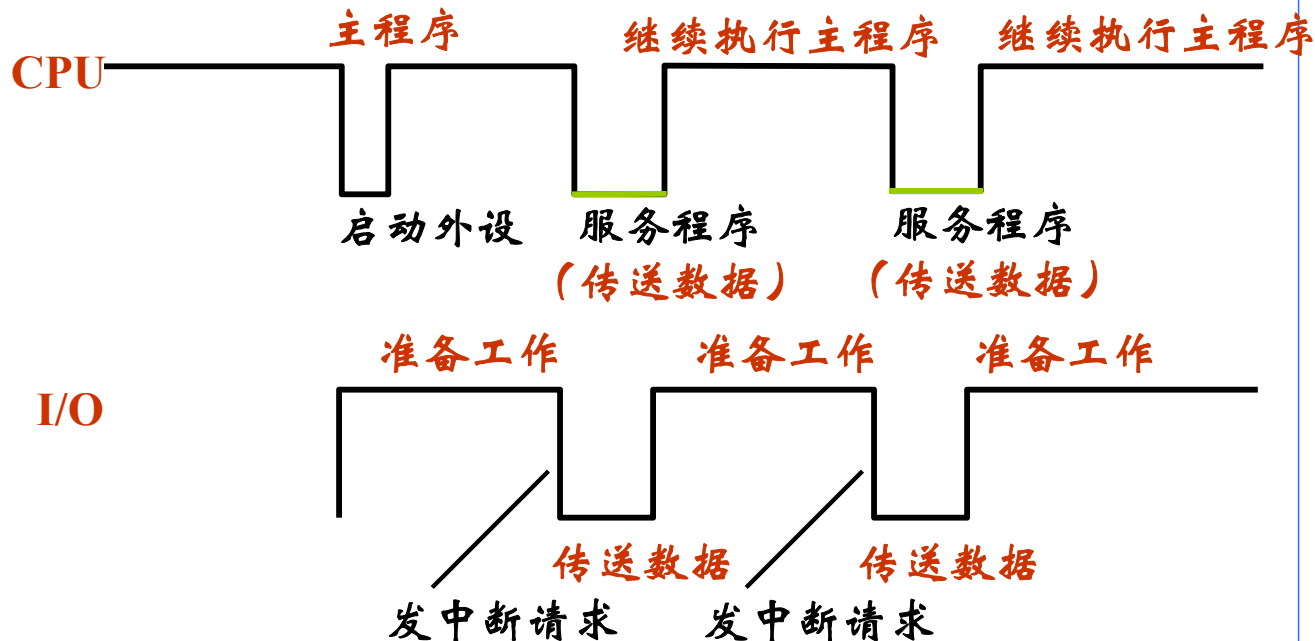
6.5.1 DMA概述

■ DMA提出

- 程序中断方式为主机和外设操作之间实现了一定程度的并行，提高了系统效率。但每交换一个单位数据就需要中断主机一次，保护现场、开关中断、恢复现场等要花费CPU时间
 - 程序中断方式适用于慢速外设
- 针对高速外设，单位数据之间的时间间隔是微秒级甚至更短，且数据的交换又是大量的、成批的。如果采用程序中断方式进行数据交换，可能出现：
 - 若主机响应请求进行数据传送，因单位数据之间的间隔短，甚至无法再利用，必须让一批数据交换完，这样交换数据便完全占用主机，又回到完全串行工作状态
 - 若主机一时不能马上响应请求，则由于单位数据不断完成交换的准备，便可能冲掉信息而造成数据丢失

6.5.1 DMA概述

主程序和服务程序抢占 CPU 示意



宏观上 CPU 和 I/O 并行工作

微观上 CPU 中断现行政程序为 I/O 服务

6.5.1 DMA概述

- 直接存储器服务方式又称为DMA（Direct Memory Access）方式
 - 以主存为中心，完全用硬件开辟外设和主存之间直接数据传输通路
 - 能使CPU工作效率更高的一种控制方式
 - DMA方式主要用来连接高速外围设备。例如，磁盘存储器，磁带存储器等

6.5.1 DMA概述

PI传送、DMA传送示意

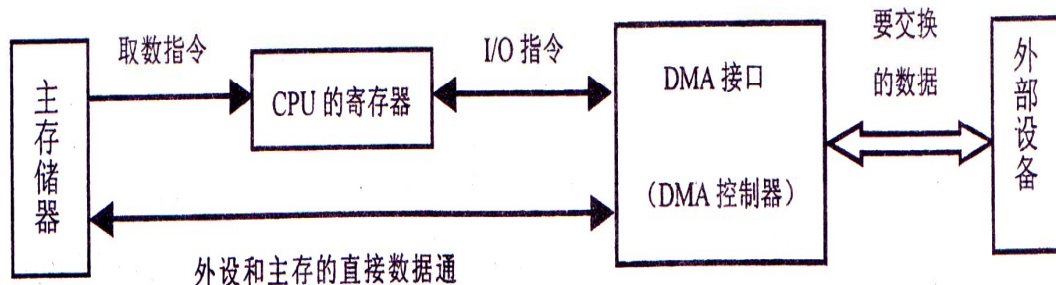
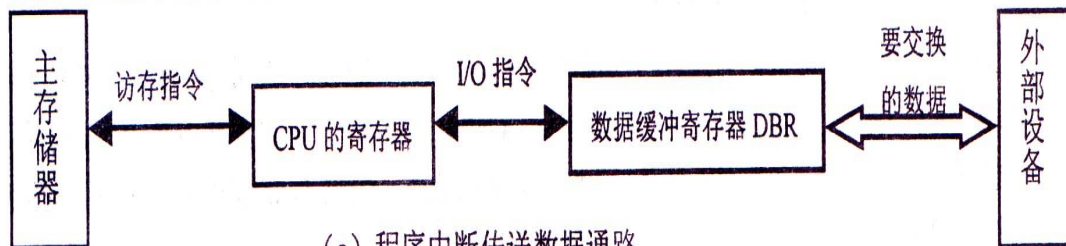


图 6.21 程序中断传送和 DMA 传送的数据通路

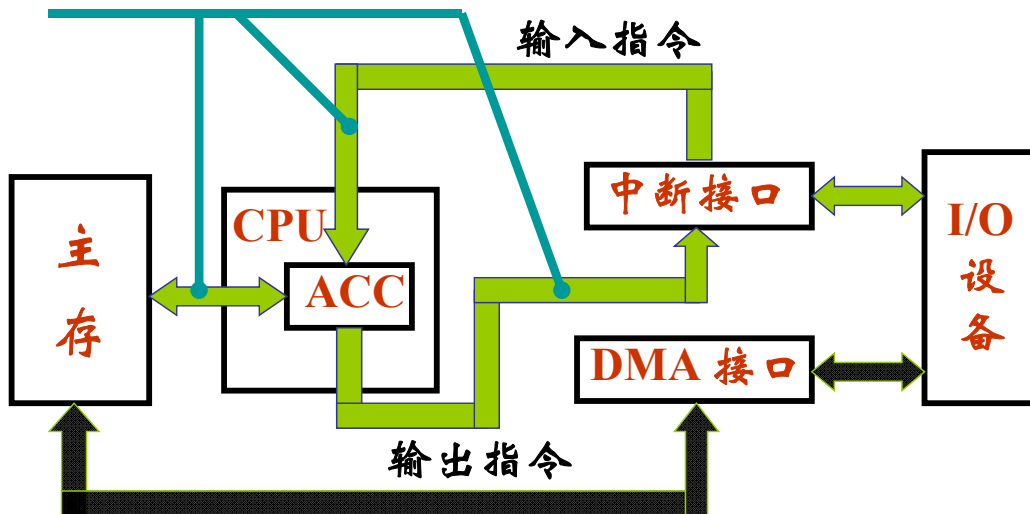
用于高速成组数据传送的外设，并按照连续地址直接访问主存

6.5.1 DMA概述

■ DMA 方式的特点

■ DMA和程序中断两种方式的数据通路

中断方式数据传送通路



DMA方式数据传送通路

6.5.1 DMA概述

■ DMA方式的特点

- 外围设备访问请求直接发往主存储器
- 不需要CPU做保存现场和恢复现场等工作
- DMA控制器中，需要设置数据寄存器、设备状态或控制寄存器、主存地址寄存器、设备地址寄存器和数据交换个数计数器
- DMA开始和结束时，需要处理机进行管理
 - 在DMA方式开始前，对DMA控制器进行初始化。传送主存缓冲区首地址、设备地址、数据块的长度等，并启动设备开始工作
 - 在DMA方式结束后，向CPU申请中断，对数据缓冲区进行后处理
- DMA执行中，数据的传送过程不需要CPU的干预

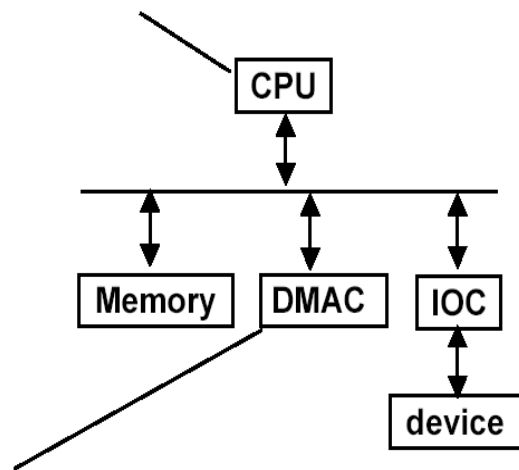
6.5.1 DMA概述

DMA方式的执行过程

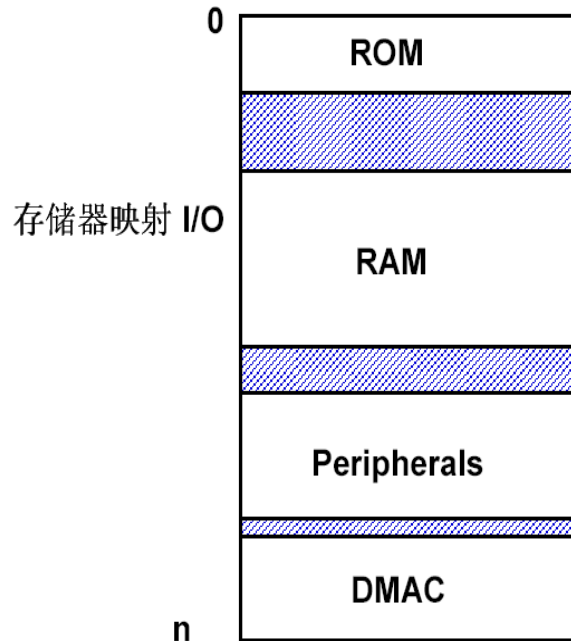
CPU向DMAC发送开始地址、方向；
然后，发射“开始”命令。

1 次中断服务 48 μsec

0.0001 秒的CPU时间



DMAC 向外设控制器提供握手信号，
向存储器提供存储地址和握手信号



6.5.1 DMA概述

DMA工作方式

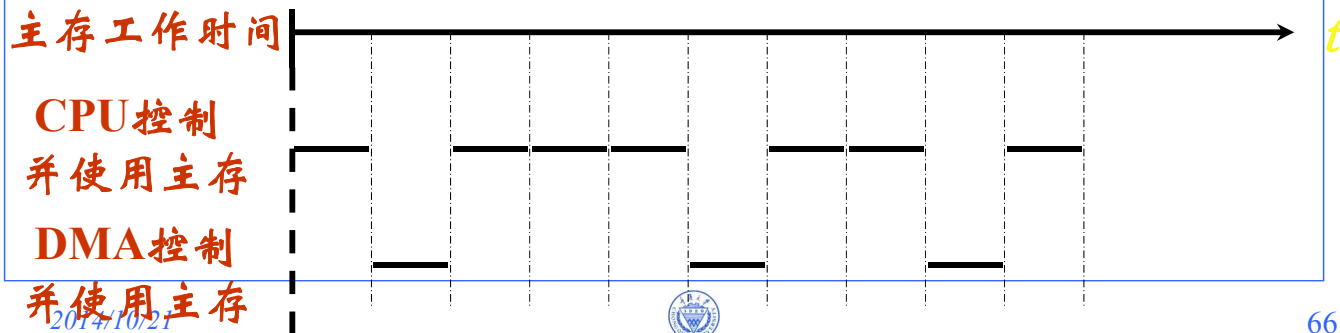
1. 周期挪用 (或称周期窃取)

DMA 要求访问主存有三种可能:

- CPU 此时不访存 (无冲突)
- CPU 正在访存 (需等待)
- CPU 与 DMA 同时请求访存 (冲突)

♣ DMA 必须优先于 CPU 访存?

♣ CPU 将总线控制权让给 DMA



6.5.1 DMA概述

1.周期挪用（简单中断）

■ 处理方法

- ♣ PD需要传送时，CPU暂停一个或若干个存储周期让给DMA
- ♣ PD交换一个单位数据后，CPU继续工作

■ 使用场合

- ♣ PD访存周期 \gg 主存存储周期，单位数据准备时间的间隔大，单位数据准备时间至少四倍于存储周期

■ 优缺点

- ♣ 存储器和CPU效率均较高，但在数据输入或输出过程中实际上占用了CPU的时间
- ♣ 控制较复杂

2014/10/21 ■ DMA大多采用这种方式

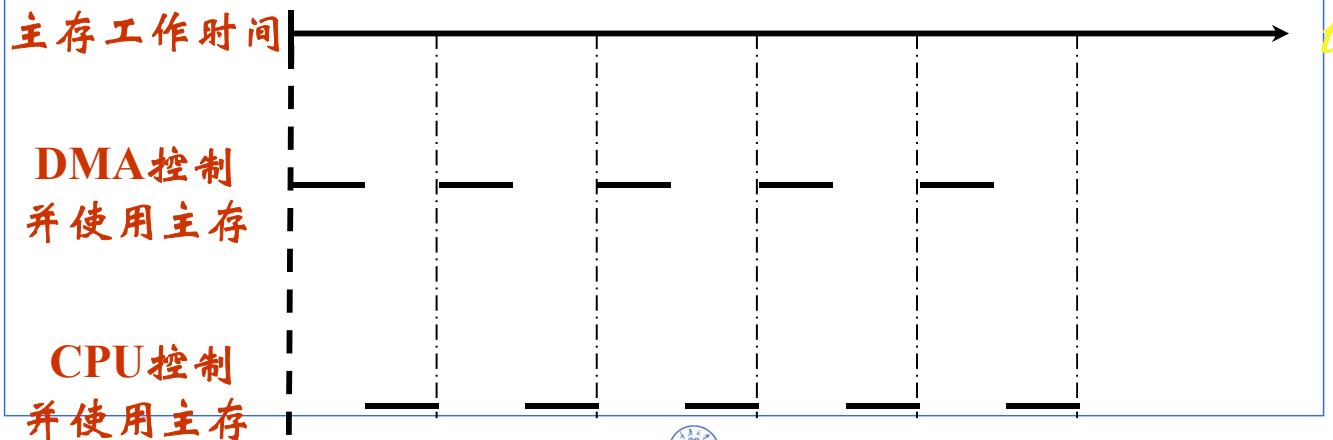
6.5.1 DMA概述

2. 交替访存方式

CPU 工作周期 $\left\{ \begin{array}{l} C_1 \text{ 专供 DMA 访存} \\ C_2 \text{ 专供 CPU 访存} \end{array} \right.$



所有指令执行过程中的一个基准时间



6.5.1 DMA概述

2. 交替访存方式（透明DMA）

■ 处理方法

- ♣ CPU、DMA各有自己的MAR、MBR，两者交替访存

■ 使用场合

- ♣ CPU工作周期（机器周期）至少是存储周期的两倍

■ 优缺点

- ♣ DMA对CPU无影响，CPU感觉不到DMA的存在，主机不停止程序运行，效率高
- ♣ 硬件控制逻辑复杂，且要求主存速度高

6.5.2 DMA接口的基本组成

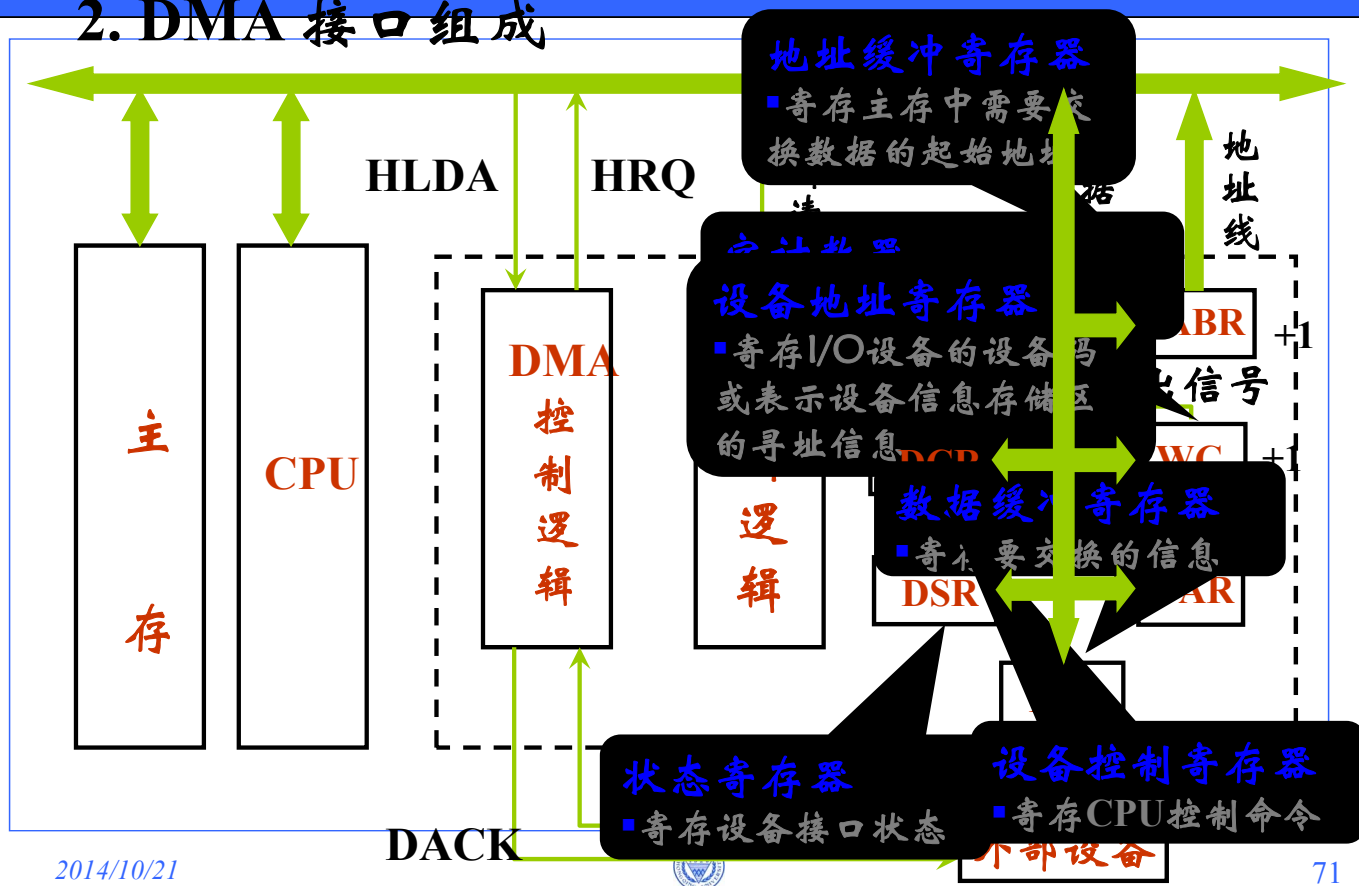
■ DMA 接口的功能和组成

1. DMA 接口功能

- (1) 向 CPU 申请 DMA 传送
- (2) 处理总线控制权的转交
- (3) 管理系统总线、控制数据传送
- (4) 确定数据传送的首地址和长度
修正 传送过程中的数据地址和长度
- (5) DMA 传送结束时，给出操作完成信号

6.5.2 DMA接口的基本组成

2. DMA 接口组成



6.5.3 DMA的工作过程

1. DMA 传送过程

■ 初始化处理、数据传送、结束处理

(1) 初始化处理 (CPU程序实现)

■ CPU通过一条I/O指令取状态

♣ 查询外设是否闲置良好

■ CPU通过几条I/O指令预置如下信息

♣ 通知 DMA 控制逻辑传送方向 (入/出)

♣ 设备地址 ——> DMA 的 DAR

♣ 主存地址 ——> DMA 的 ABR

♣ 传送字数 ——> DMA 的 WC

■ 用I/O指令启动外设, CPU继续执行现行程序

(2) DMA 传送过程示意

数据传送

CPU

初始化处理:

主存起始地址 → DMA
设备地址 → DMA
传送数据个数 → DMA
启动设备

数据传送:

继续执行主程序
同时完成一批数据传送

结束处理:

中断服务程序
做 DMA 结束处理

继续执行主程序

DMA 请求

否

允许传送?

是

主存地址送总线
数据送 I/O 设备 (或主存)
主存地址加 1
传送个数减 1

否

数据块
传送结束?

是

向 CPU 申请 程序中断

6.5.3 DMA的工作过程

(3) 结束处理

校验送入主存的数据是否正确

是否继续用 DMA

测试传送过程是否正确，错则转错误处理程序

由中断服务程序完成

6.5.3 DMA的工作过程

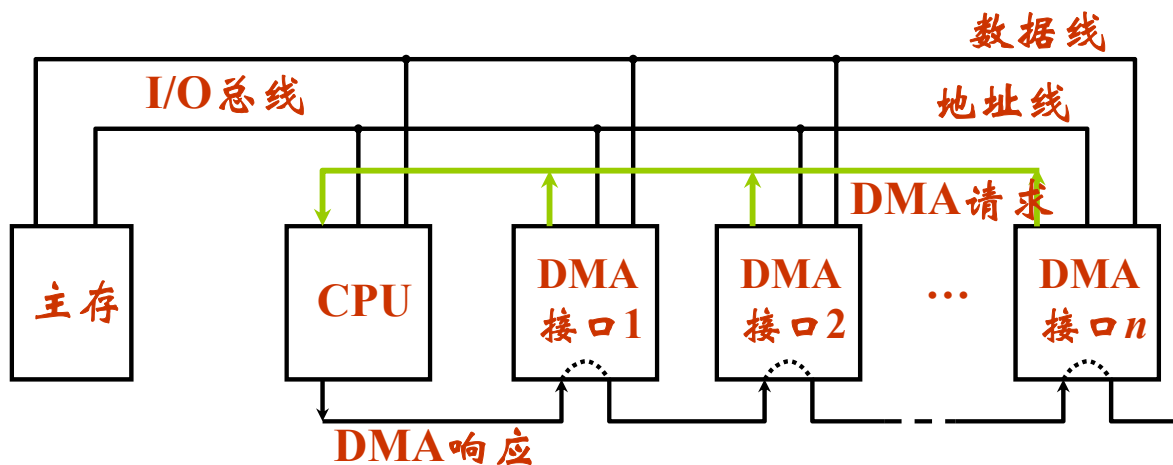
■ PIT与周期挪用DMA传送的区别

1. PIT主要靠软件实现数据传送，DMA主要靠硬件实现数据传送
2. DMA请求的响应，只要求CPU暂停1个访存周期，实现1个单位数据的传送。PIT则中止现行程序的运行，转去执行中断服务程序，实现1个单位数据的传送
3. PIT只适用于慢速外设，DMA适用于高速成组传送的外设
4. PIT的响应必须在一条指令执行之末，而DMA原则上可在CPU不访存的任何时刻
5. PI的功能强，可处理各种复杂突发事件，而DMA只限于数据传送

6.5.4 DMA传送接口类型

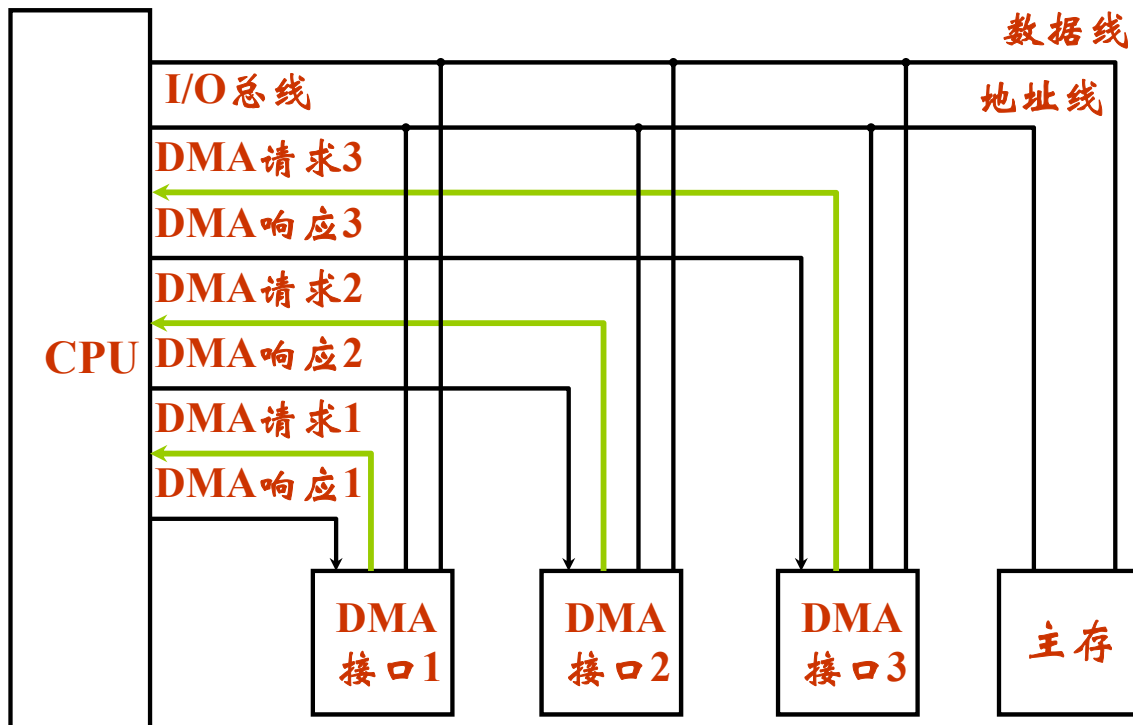
■ DMA接口与系统的连接方式

1. 具有公共请求线的DMA请求



6.5.4 DMA传送接口类型

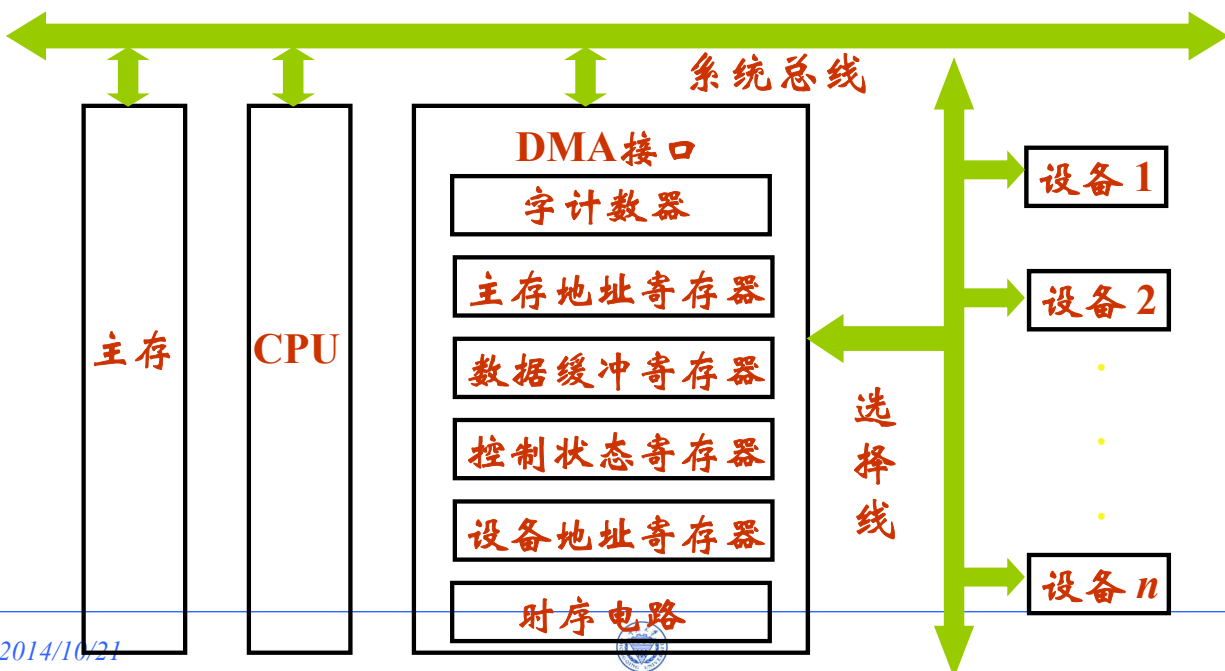
2. 独立的DMA请求



6.5.4 DMA传送接口类型

DMA接口的类型

1. 选择型DMA传送接口 { 在物理上连接多个设备
在逻辑上只允许连接一个设备

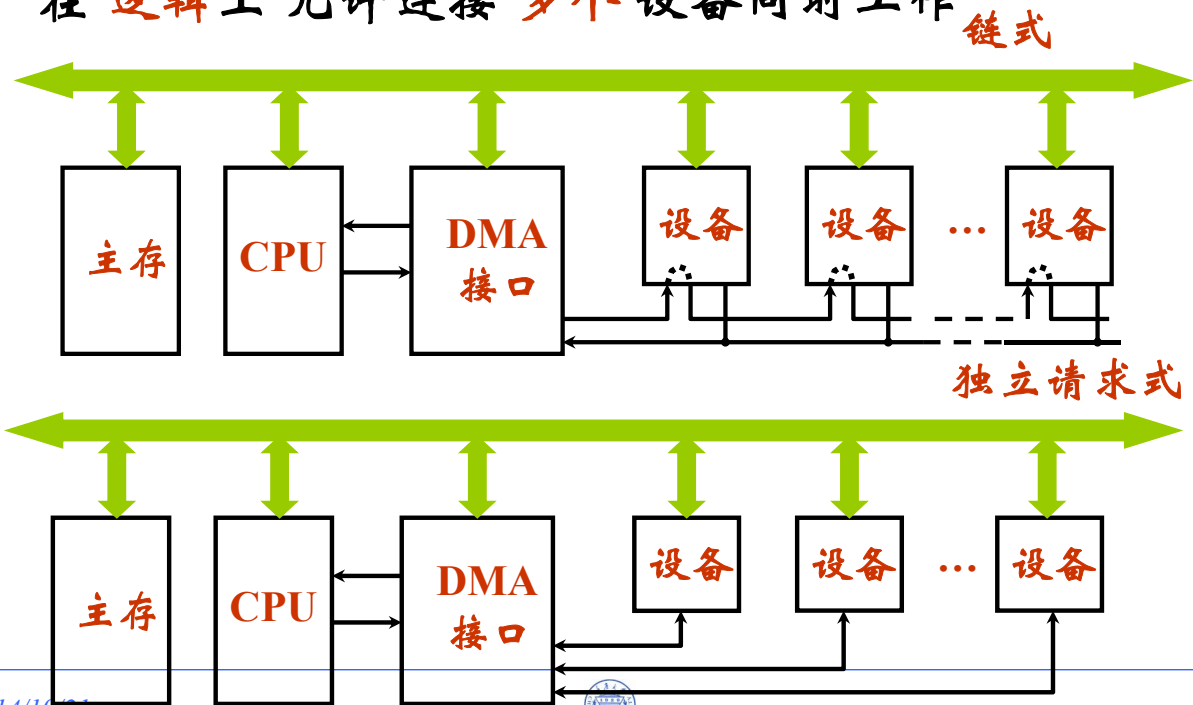


6.5.4 DMA传送接口类型

2. 多路型DMA传送接口

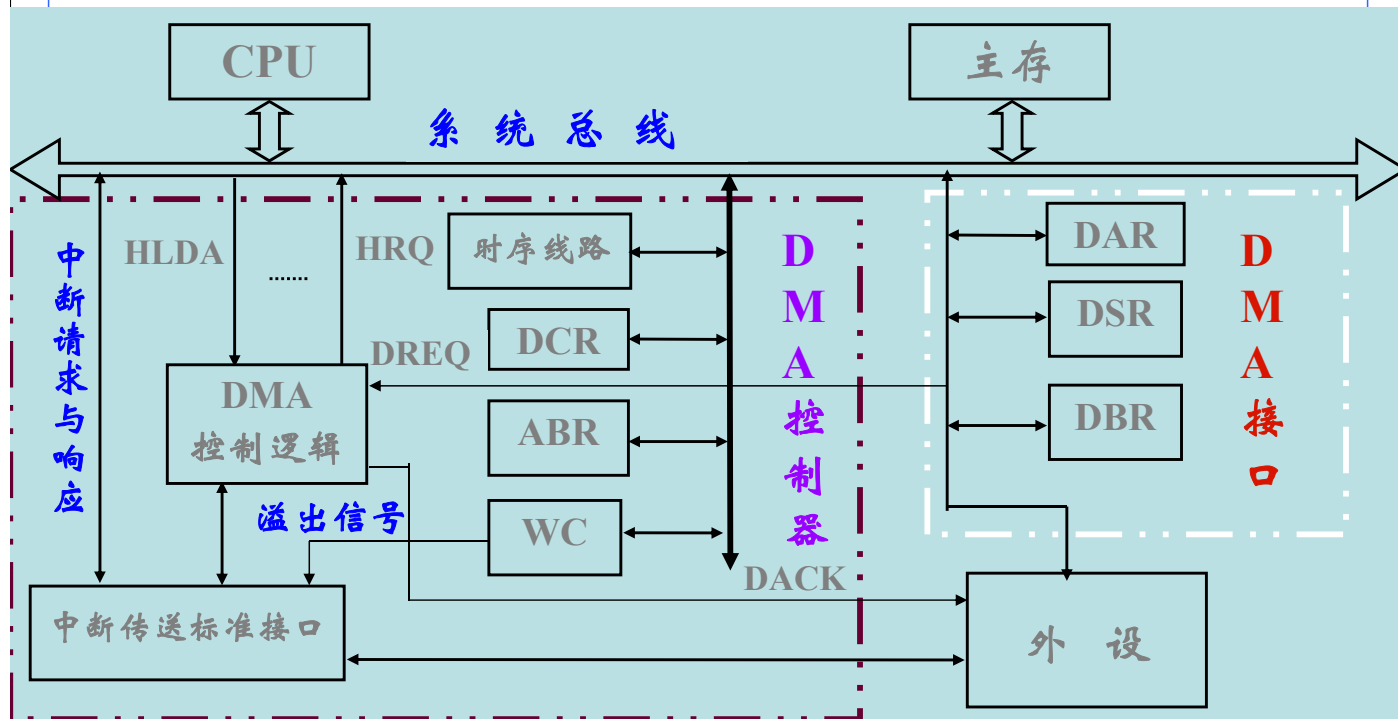
在物理上连接多个设备

在逻辑上允许连接多个设备同时工作



6.5.4 DMA传送接口类型

3. 广义型DMA传送接口



6.5.4 DMA传送接口类型

- DMA进行数据传送时，数据经过DMA接口与数据总线直接传送
- DMA只控制数据传输，即负责、接管总线，并将总线控制权转交给DMA接口
- Intel 公司生产的DMA控制器
 - 8237、8257和8258

小结

■ DMA 方式与程序中断方式的比较

中断方式

DMA 方式

(1) 数据传送	程序	硬件
(2) 响应时间	指令执行结束	存取周期结束
(3) 处理异常情况	能	不能
(4) 中断请求	传送数据	结束处理
(5) 优先级	低	高



重庆大学
CHONGQING UNIVERSITY

计算机学院

COLLEGE OF COMPUTER SCIENCE

敬请批评指正
谢 谢