



河海大学

第八章 输入输出系统

黄倩

huangqian@hhu.edu.cn

勤学楼4203



内容提要

- I/O系统概述
 - I/O系统的性能与功能
- I/O设备概述
 - I/O设备的通用模型
 - I/O接口
 - I/O设备的寻址方式
- CPU与I/O设备间的信息交换方式



1.1 I/O系统的性能

两个常用的性能指标：

吞吐率，即I/O带宽：

单位时间内从系统输入/输出多少数据？

单位时间内实现了多少次输入/输出操作？

(服务器更关注)

响应时间，即等待延迟：

在多长时间时间内完成请求的任务？

(台式机和嵌入式更关注)

不同的任务对性能的要求不同：

要求吞吐量高的场合：

如：多媒体应用（音/视频的播放要流畅！）

要求响应时间短的场合：

如：事务处理系统（存/取款的速度要快！）

要求吞吐率高且响应时间短的场合：

文件服务器、Web服务器等



1.2 I/O系统的功能

输入/出系统的功能：

解决各种形式信息的输入和输出

即：用户如何将所需的信息（文字、图表、声音、视频等）通过不同的外设输入到计算机中，以及计算机内部处理的结果信息如何通过相应的外设输出给用户

要实现上述功能需解决一系列的问题：

怎样在CPU、主存和外设间建立一个高效信息传输“通路”；

怎样将用户的I/O请求转换成设备的命令；

如何对外设进行编址；

怎样使CPU方便地寻找到要访问的外设；

I/O硬件和操作系统如何协调完成主机和外设之间的数据传送

.....

以上是本章的主要内容



内容提要

- I/O系统概述
 - I/O系统的性能与功能
- I/O设备概述
 - I/O设备的通用模型
 - I/O接口
 - I/O设备的寻址方式
- CPU与I/O设备间的信息交换方式



2.1 I/O设备的类型

从交互方式上，分为：

人-机交互设备

输入/输出的信息是人可读的

如：键盘、鼠标、扫描仪、打印机、显示器等

机器可读设备

输入/输出的信息是机器可读的，人无法读取

如：网卡、Modem、D/A、A/D、磁盘、声音输入设备等

从交互方向上，分为：

输入设备：键盘、鼠标、扫描仪等

输出设备：打印机、显示器等

双向交互设备：磁盘、网卡等

从数据的组织方式上，分为：

块设备（以数据块作为信息的存储和传输单位）：磁盘等

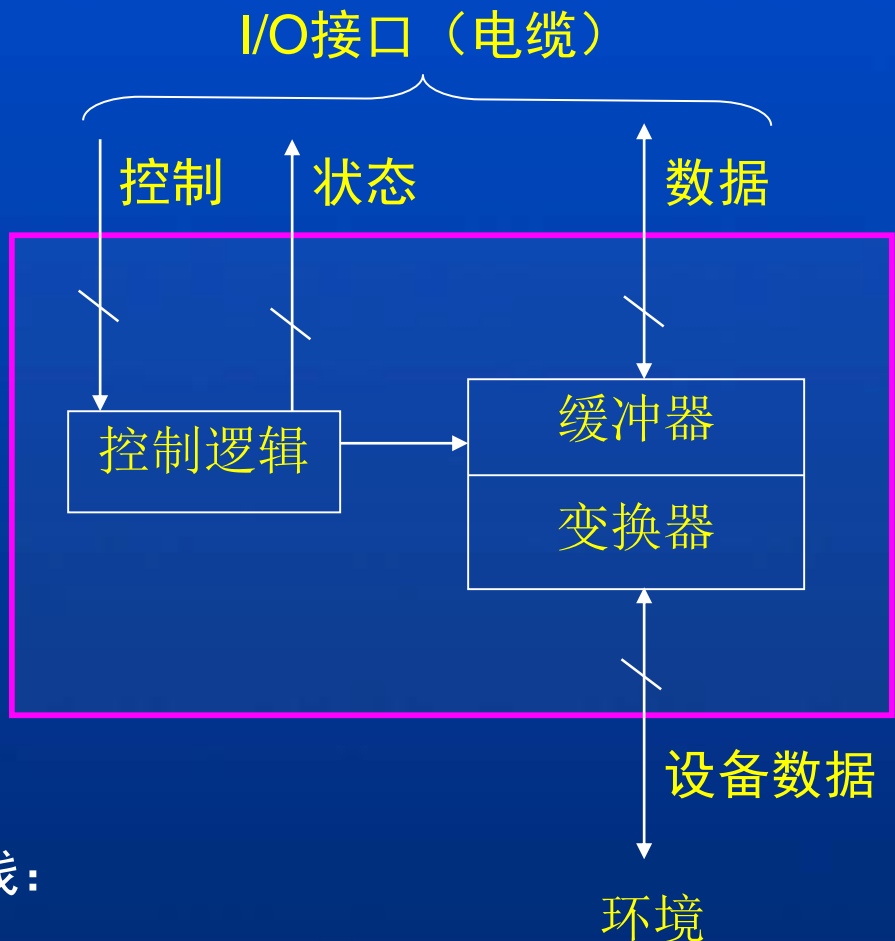
字符设备（以字符作为信息的存储和传输单位）：鼠标、键盘等

从速度上分.....



河海大学 2.2 I/O设备的通用模型

- 通过**电缆**与计算机内部I/O接口进行数据、状态和控制信息的传送
- 控制逻辑**根据控制信息控制设备的操作，并检测设备状态
- 缓冲器**用于保存交换的数据信息
- 变换器**用于在电信号形式（内部数据）和其他形式的设备数据之间进行转换



所有设备都可以抽象成这个通用模型！

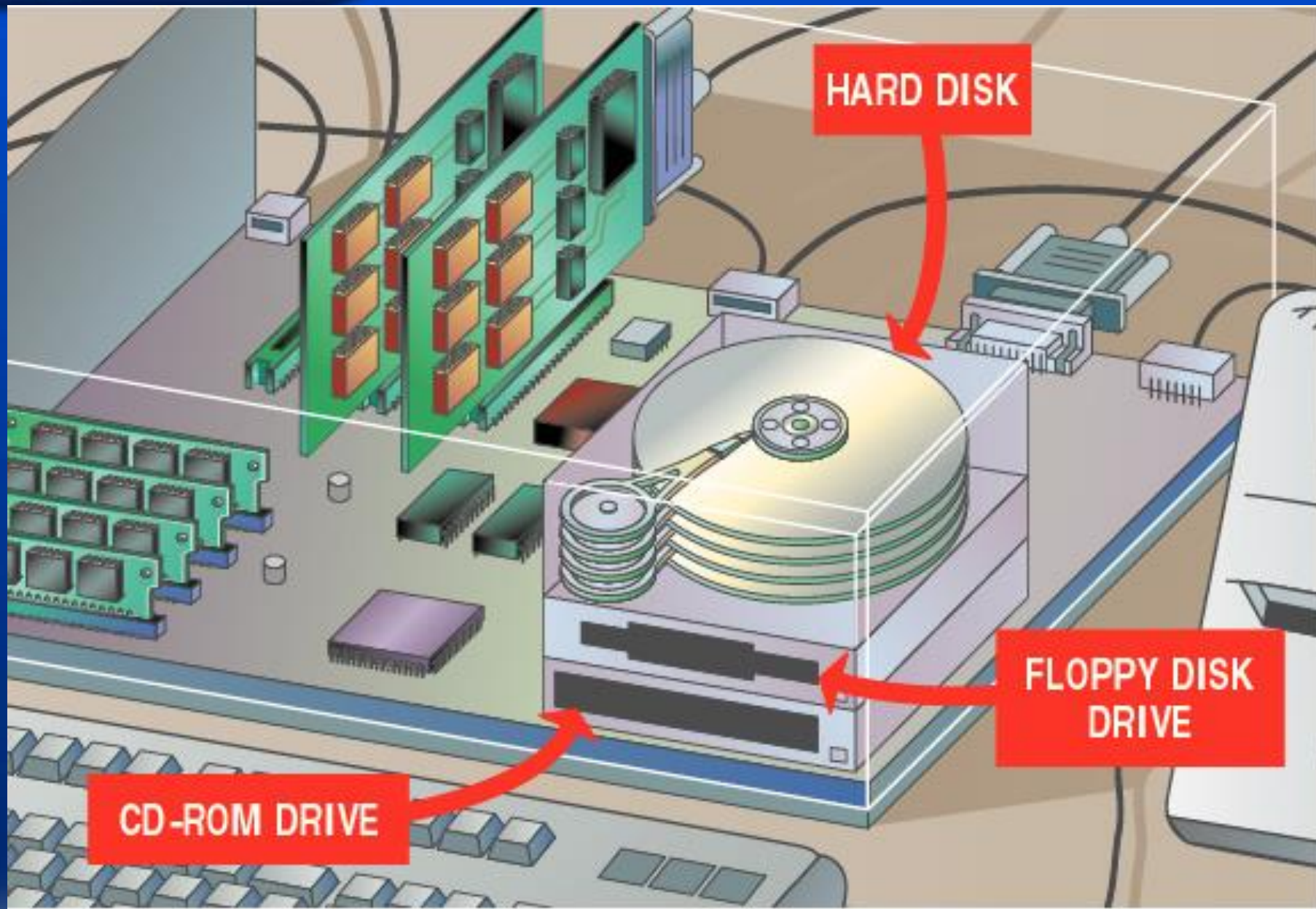
设备所用的电缆线中有以下三种信号线：

控制信号、状态信号、数据信号

下面以磁盘为例，说明I/O设备的工作原理

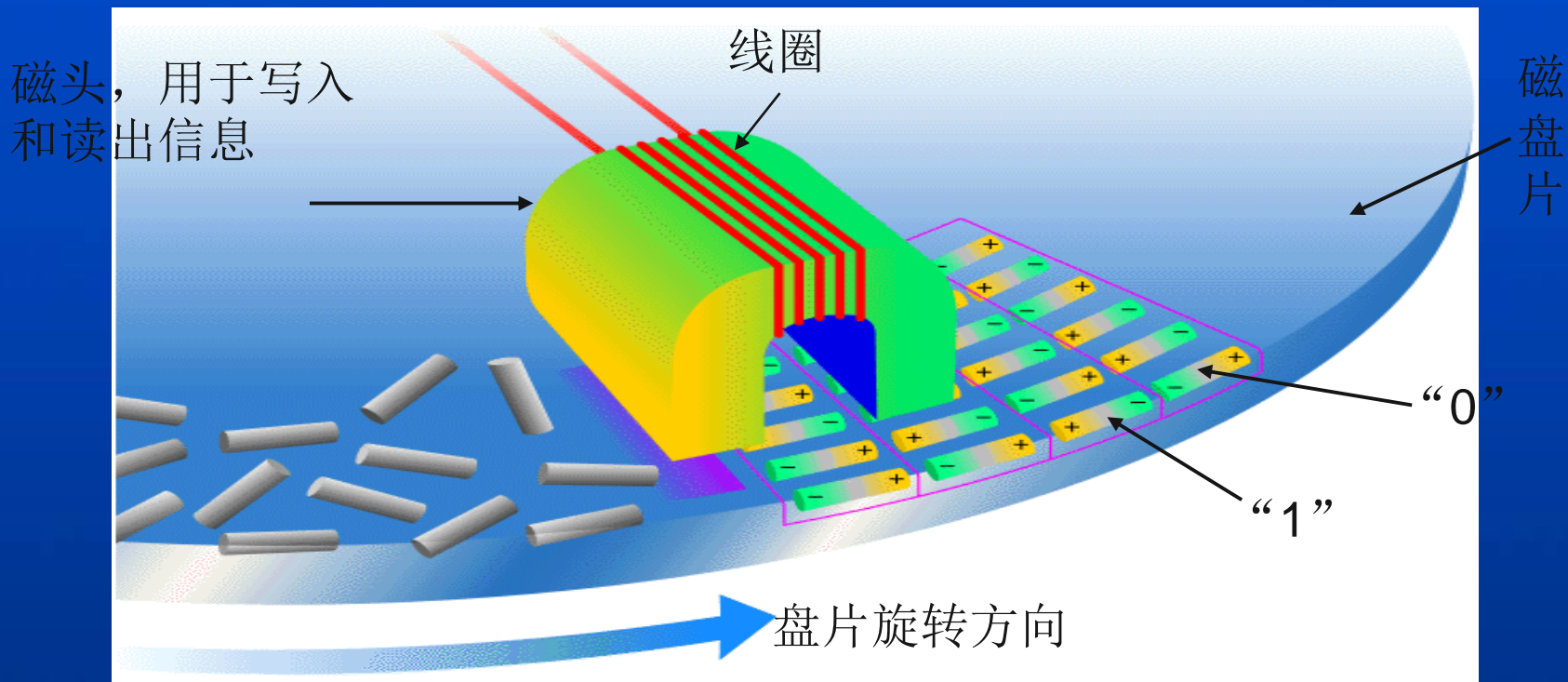


河海大学 2.2 I/O设备的通用模型





磁盘信息存储原理



写1：线圈通以正向电流，使呈N-S状态

写0：线圈通以反向电流，使呈S-N状态

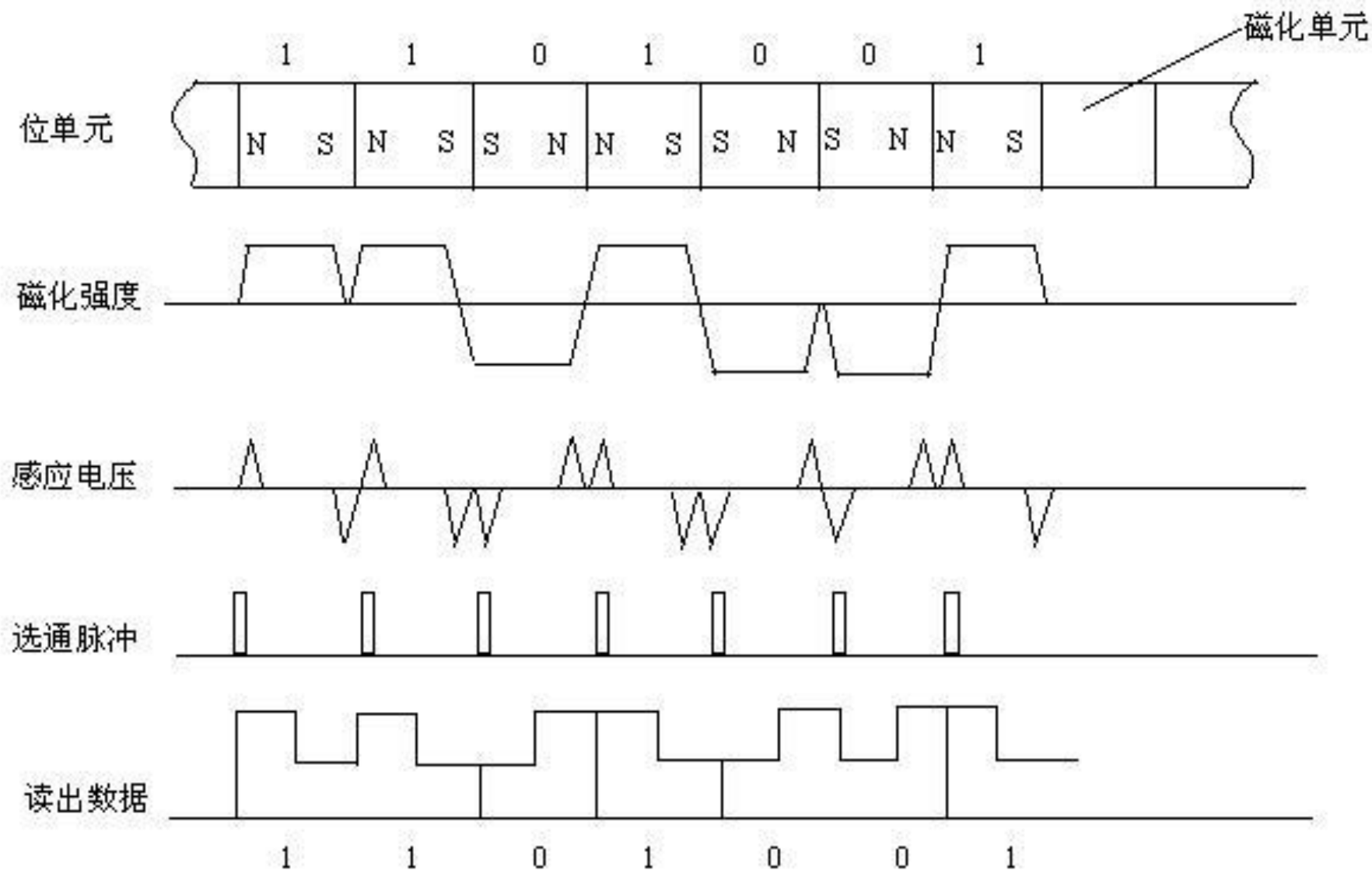
不同的磁化状态被记录在磁盘表面

读时：磁头固定不动，载体运动。因为载体上小的磁化单元外部的磁力线通过磁头铁芯形成闭合回路，在铁芯线圈两端得到感应电压。根据感应电压的不同的极性，可确定读出为0或1。



河海大学

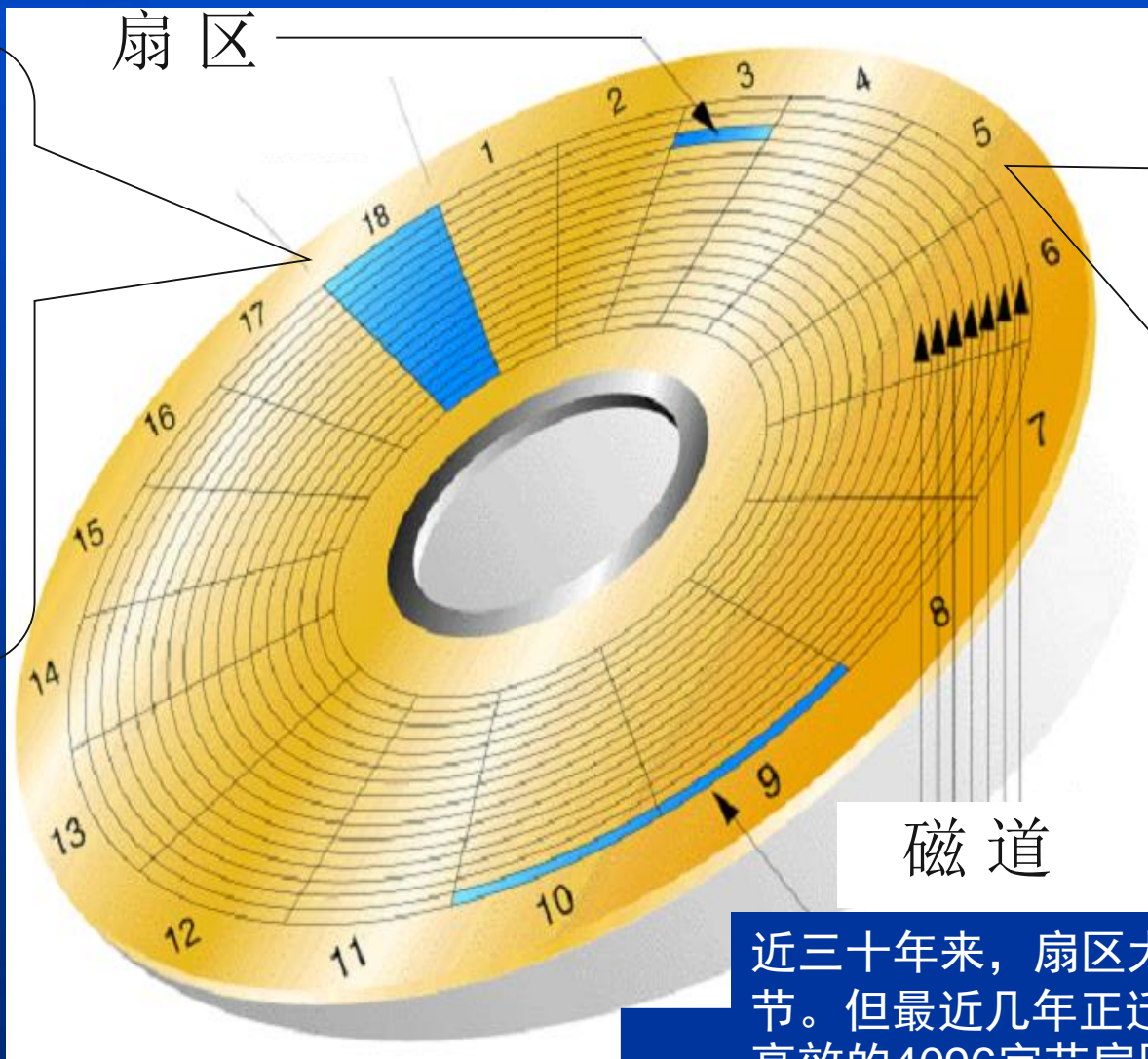
磁表面信息读出过程





磁道和扇区

每个磁道被划分为若干段（段又叫扇区），每个扇区的存储容量为512字节。每个扇区都有一个编号



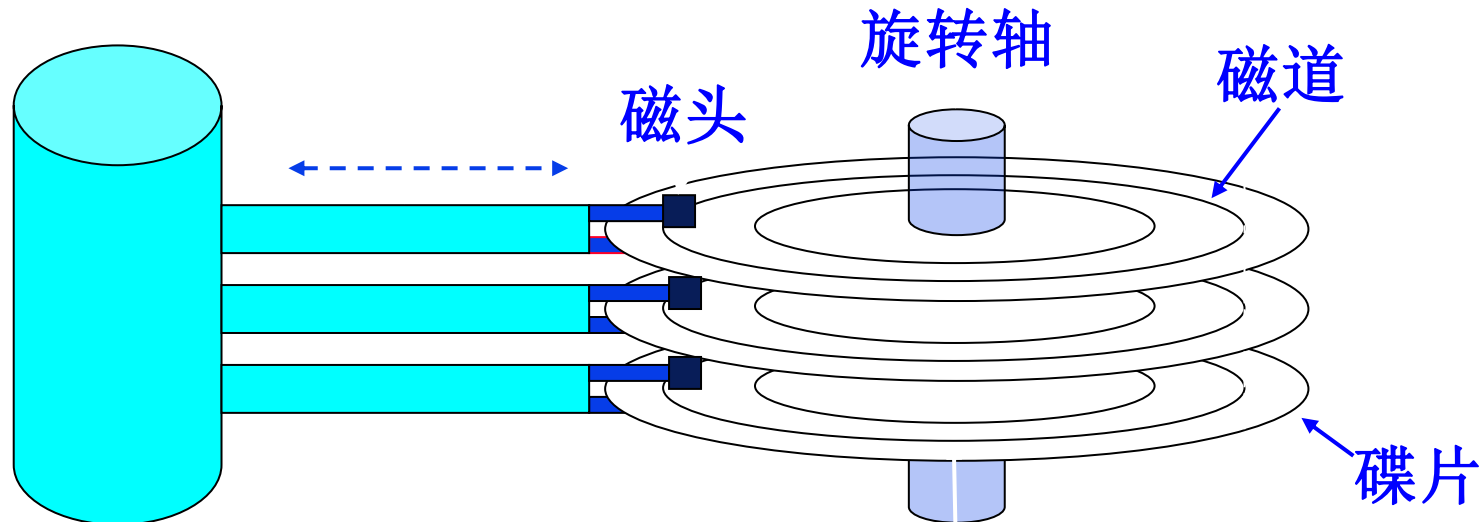
磁盘表面被分为许多同心圆，每个同心圆称为一个磁道。每个磁道都有一个编号，最外面的是0磁道

注：所谓磁盘的格式化操作，指在盘面上划分磁道和扇区，并在扇区中填写扇区号等信息的过程

近三十年来，扇区大小一直是512字节。但最近几年正迁移到更大、更高效的4096字节扇区，通常称为4K扇区。国际硬盘设备与材料协会（IDEMA）将之称为高级格式化。



平均存取时间



硬盘的操作流程如下：

所有磁头同步寻道（由柱面号控制）→ 选择磁头（由磁头号控制）→
被选中磁头等待扇区到达磁头下方（由扇区号控制）→ 读写该扇区中数据

◦ 磁盘上的信息以扇区为单位进行读写，平均存取时间为：

$T = \text{寻道时间} + \text{旋转等待时间} + \text{数据传输时间（忽略不计）}$

- **寻道时间**——磁头寻找到指定磁道所需时间 (大约5ms)
- **旋转等待时间**——指定扇区旋转到磁头下方所需要的时间 (大约4~6ms) (转速：4200 / 5400 / 7200 / 10000rpm)
- **数据传输时间**——(大约0.01ms / 扇区)



河海大学磁盘响应时间计算举例

假定每个扇区512字节，磁盘转速为5400 RPM，广告声称寻道时间为12 ms，数据传输率为4 MB/s，磁盘控制器开销为1 ms，不考虑排队时间，则磁盘响应时间为多少？

$$\begin{aligned}\text{Disk Response Time} &= \text{Seek time} + \text{Rotational Latency} + \text{Transfer time} \\ &\quad + \text{Controller Time} + \text{Queuing Delay} \\ &= 12 \text{ ms} + 0.5 / 5400 \text{ RPM} + 0.5 \text{ KB} / 4 \text{ MB/s} + 1 \text{ ms} + 0 \\ &= 12 \text{ ms} + 0.5 / 90 \text{ RPS} + 0.125 / 1024 \text{ s} + 1 \text{ ms} + 0 \\ &= 12 \text{ ms} + 5.5 \text{ ms} + 0.1 \text{ ms} + 1 \text{ ms} + 0 \text{ ms} \\ &= 18.6 \text{ ms}\end{aligned}$$

如果实际的寻道时间只有1/3的话，则为10.6ms，这样旋转等待时间就占了近50%！

$$12/3 + 5.5 + 0.1 + 1 = 10.6 \text{ ms} \quad \text{所以，磁盘转速非常重要！}$$

为什么实际的寻道时间可能只有1/3？

访问局部性使得每次磁盘访问大多在局部磁道！

能否算出每道有多少扇区？

$$4 \text{ MB} \times 60 / 512 \text{ B} \times 5400 \approx 87 \text{ 个扇区}$$



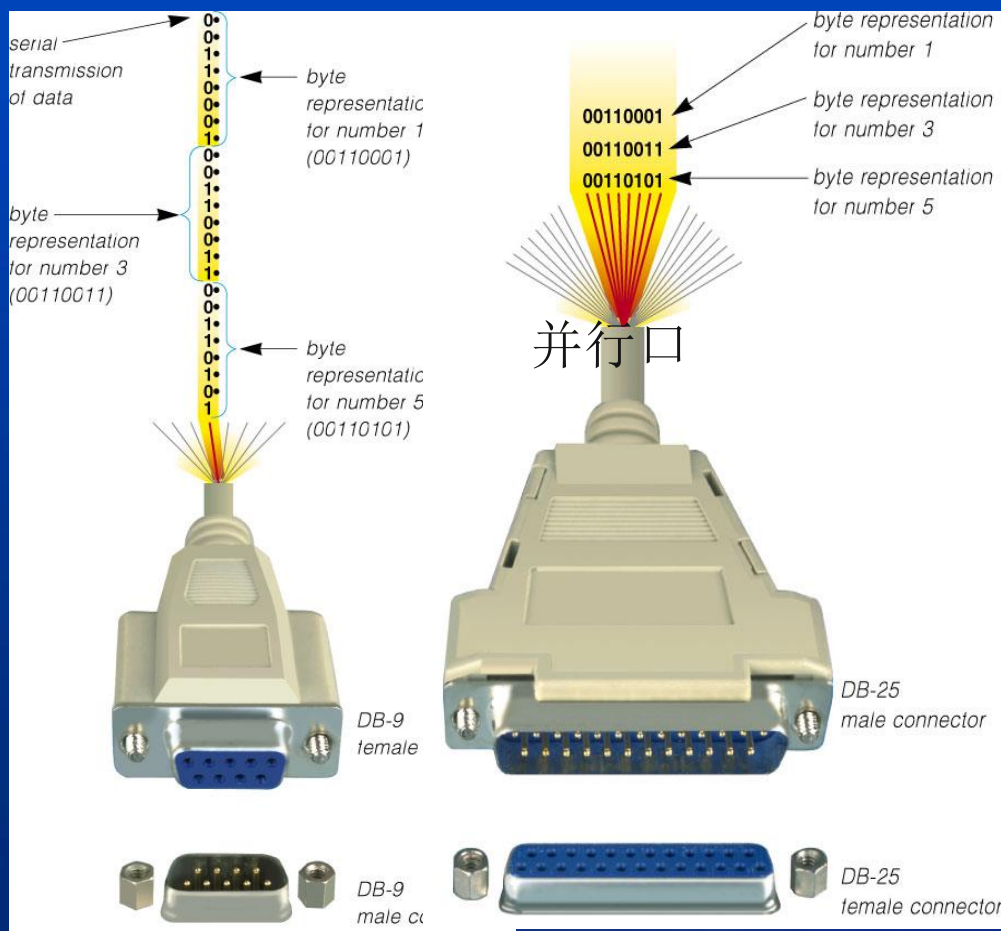
内容提要

- I/O系统概述
 - I/O系统的性能与功能
- I/O设备概述
 - I/O设备的通用模型
 - I/O接口
 - I/O设备的寻址方式
- CPU与I/O设备间的信息交换方式



2.3 I/O接口

- I/O接口：I/O设备与I/O控制器之间的连接器
包括：插头 / 插座的形式、通讯规程和电器特性等
- 分类：
 - 从数据传输方式来分：
 - 串行（一次只传输1位）
 - 并行（多位一起进行传输）
 - 从是否能连接多个设备来分：
 - 总线式（可连接多个设备）
 - 独占式（只能连接1个设备）
 - 从是否符合标准来分：
 - 标准接口（通用接口）
 - 专用接口（专用接口）
 - 按功能选择的灵活性来分：
 - 可编程接口
 - 不可编程接口

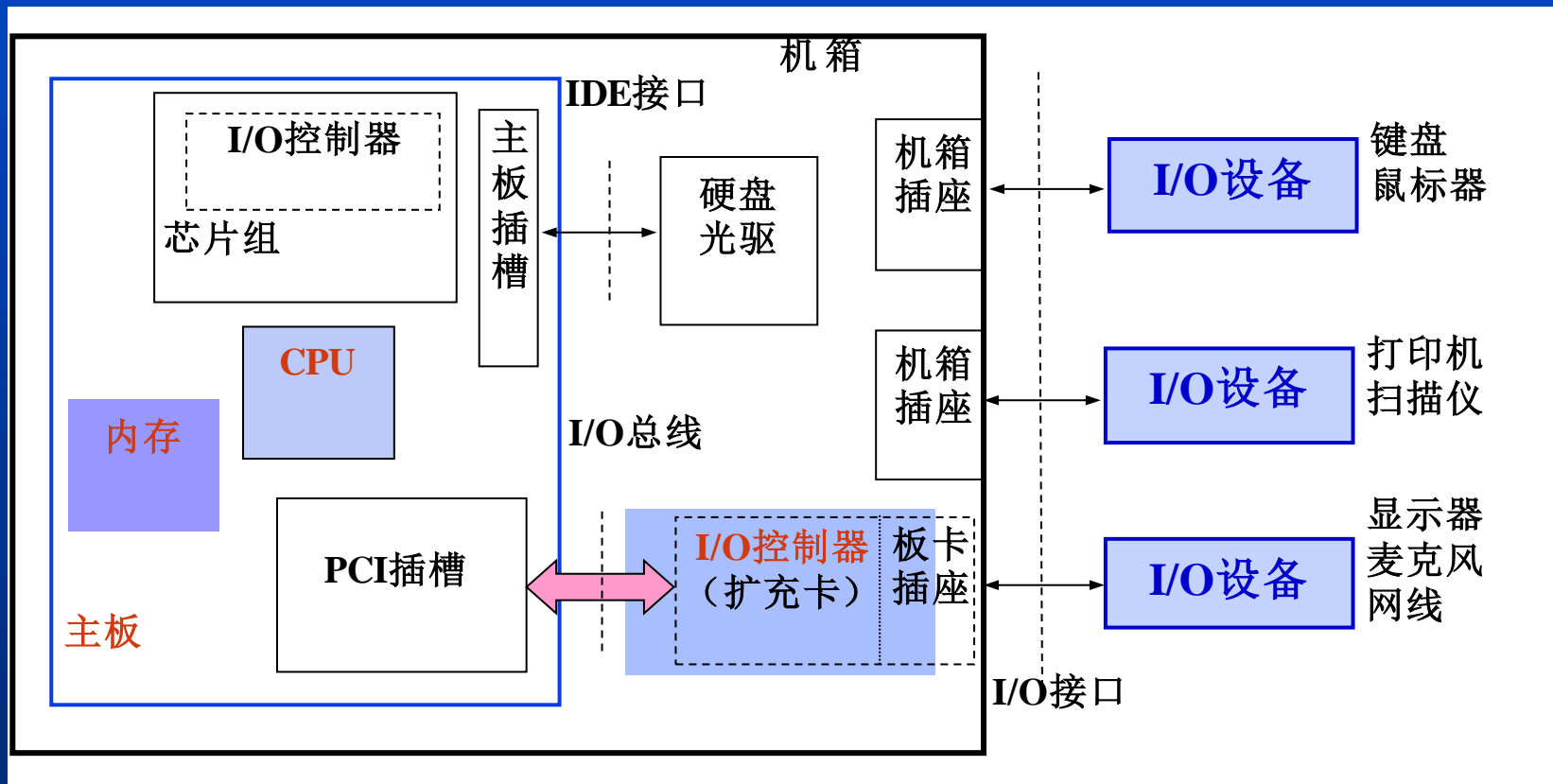


串行口



2.3 I/O接口

- I/O总线,I/O控制器,I/O接口与I/O设备的关系



本课程把I/O控制器和插座合起来称为I/O接口。



数据缓冲

提供数据缓冲寄存器，以达到主机和外设工作速度的匹配。

错误或状态检测

提供状态寄存器，以保存各种错误或状态信息供CPU查用。

控制和定时

提供控制和定时逻辑，以接受从系统总线来的控制定时信号。

数据格式转换

提供数据格式转换部件使通过外部接口得到的数据转换为内部接口需要的格式，或在相反的方向进行数据格式转换。

与主机和设备通信

上述功能通过I/O接口与主机之间、I/O接口与设备之间的通信来完成。



内容提要

- I/O系统概述
 - I/O系统的性能与功能
- I/O设备概述
 - I/O设备的通用模型
 - I/O接口
 - I/O设备的寻址方式
- CPU与I/O设备间的信息交换方式



河海大学2.4 I/O设备的寻址方式

- 对I/O端口读写，就是向I/O设备送出命令或从设备取得状态或读/写设备数据
- 一个I/O控制器可能会占有多个端口地址
- I/O端口必须编号后，CPU才能访问它
- I/O设备的寻址方式就是I/O端口的编号方式

(1) 统一编址方式（内存映射方式）

与主存空间统一编址，将主存空间分出一部分地址给I/O端口进行编号。

（因该方法是将I/O端口映射到某主存区域，故也称为“存储器映射方式”）

例如，RISC机器、Motorola公司的处理器等采用该方案

(2) 独立编址方式（特殊I/O指令方式）

不和主存单元一起编号，而是单独编号，使成为一个独立的I/O地址空间

（因需专门I/O指令，故也称为“特殊I/O指令方式”）

例如，Intel公司和Zilog公司的处理器就是独立编址方式



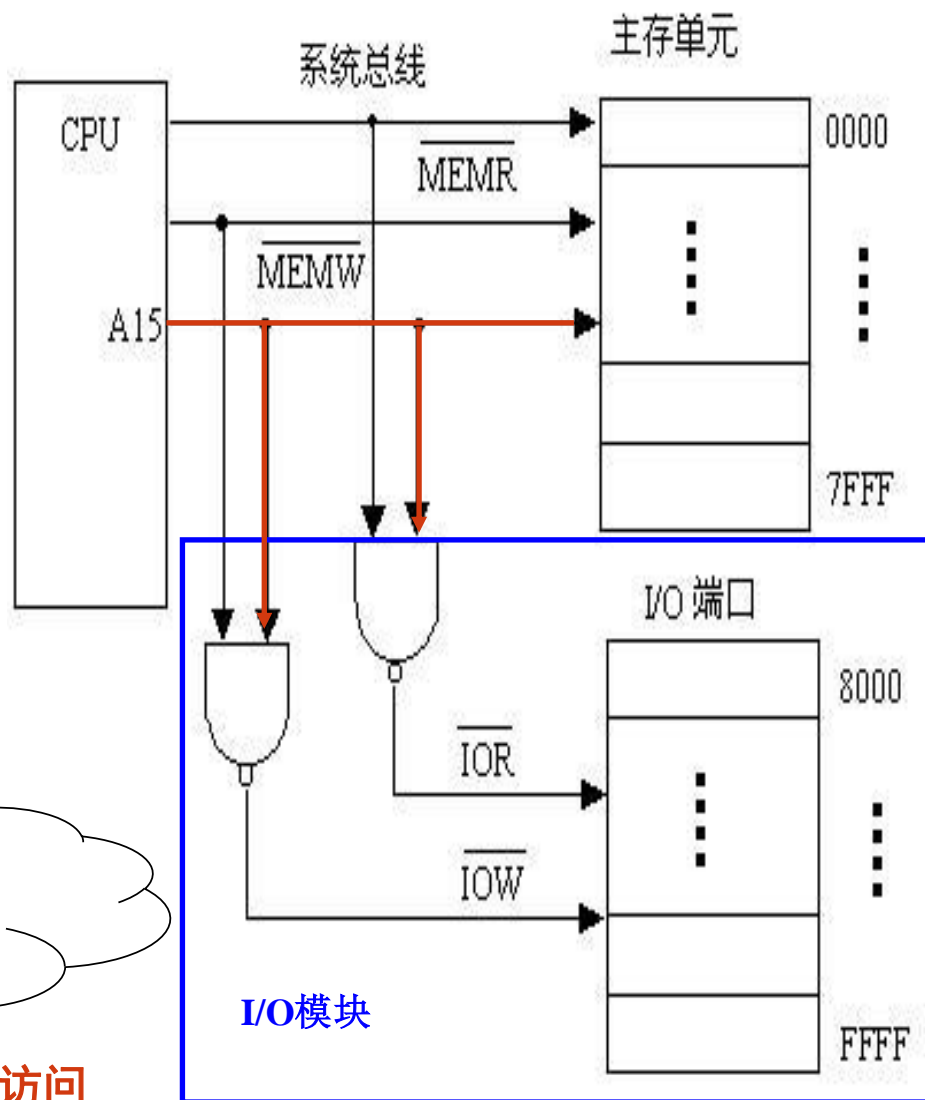
河海大学2.4 I/O设备的寻址方式

统一编址方式

- ° CPU不直接通过读写控制信号 \overline{IOR} 、 \overline{IOW} 对I/O端口读写，而是根据I/O端口在地址空间的位置，通过地址译码来实现。
- ° 地址线的高位参与片选控制逻辑。
- ° 无需设置专门I/O指令，只要用一般访存指令就可存取I/O端口。

MEMR或MEMW命令由访存指令发出， \overline{IOR} 和 \overline{IOW} 命令怎样呢？

也由访存指令发出，只是访问的地址范围不同！





统一编址方式的优缺点

◦ 主要优点：

- 与访存指令一致的I/O存取指令
 - 减少对专门I/O指令的设计
 - 利用丰富的访存指令进行I/O操作（不仅可对端口进行数据传送，而且还可直接对端口进行移位、测试等）。
- 便于扩大系统吞吐率
 - 外设或I/O寄存器数目几乎不受限制，而只受总存储容量的限制。这在大型控制或数据通信系统等特殊场合很有用。
- 读写控制逻辑简单

◦ 主要缺点：

- 主存空间减少。因为被I/O占用了存储空间。
- 外设寻址时间长。为了识别I/O端口,全部地址线都需参与地址译码,使译码电路复杂并需花很长时间。



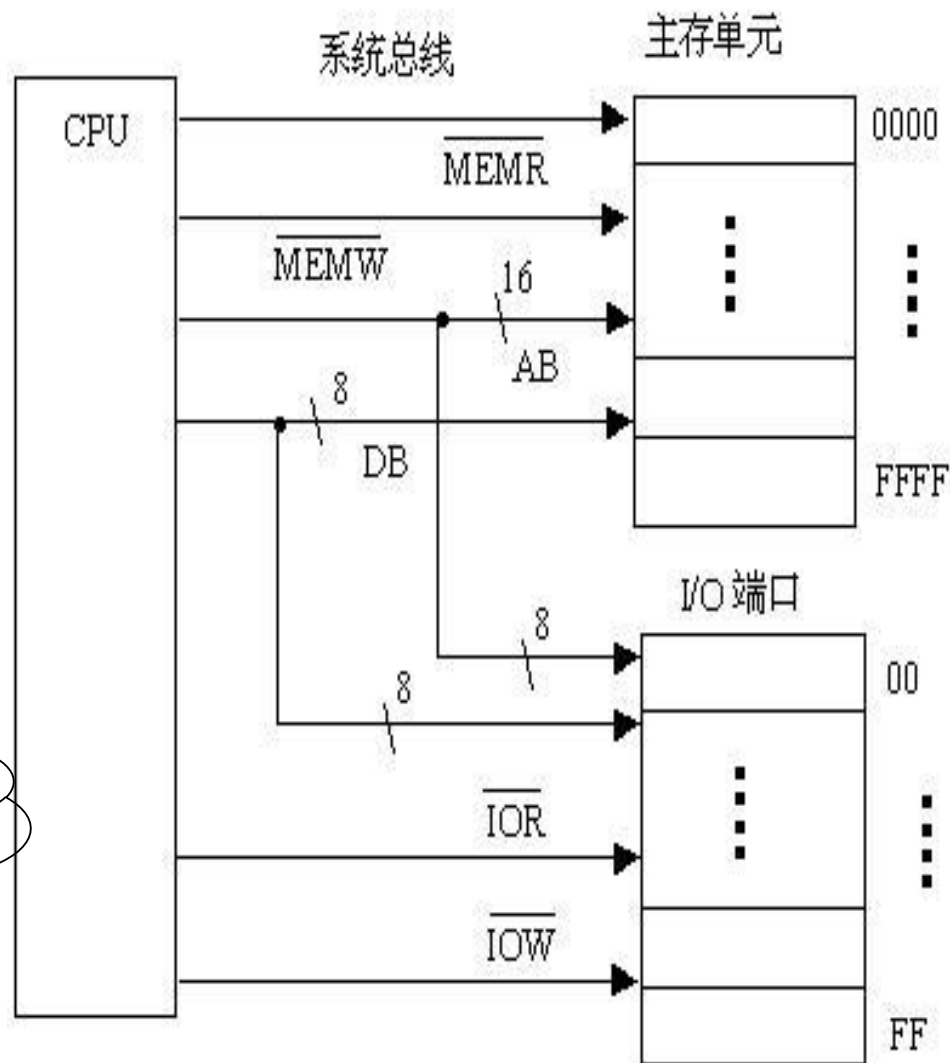
河海大学2.4 I/O设备的寻址方式

独立编址方式

- 通过不同的读写控制信号 $\overline{\text{IOR}}$ 、 $\overline{\text{IOW}}$ 和 $\overline{\text{MEMR}}$ 、 $\overline{\text{MEMW}}$ 来控制对 I/O 端口和存储器的读写。
- 一般 I/O 端口比存储器单元少，所以选择 I/O 端口时，只需少量地址线。
- 指令系统必须设计专门的 I/O 指令。

MEMR或MEMW命令
由访存指令发出，IOR
和IOW命令怎样呢？

由专门的 I/O 指令确定，指令中给的地址可能相同，但操作命令不同！





独立编址方式的优缺点

◦ 主要优点：

- I/O端口地址不占用存储器地址空间，故主存空间不受I/O地址的影响。
- I/O地址线较少，所以I/O端口译码简单，寻址速度快。
- 使用专用I/O指令，程序清晰，便于理解和检查。

◦ 主要缺点：

- 专用 I/O指令类型少，只提供简单的传输操作，故程序设计灵活性差。
- 要求处理器提供两组读写命令（MEMR / MEMW、IOR / IOW），增加了控制逻辑的复杂性和处理器引脚数。



内容提要

- I/O系统概述
 - I/O系统的性能与功能
- I/O设备概述
 - I/O设备的通用模型
 - I/O接口
 - I/O设备的寻址方式
- CPU与I/O设备间的信息交换方式



CPU与I/O设备间的信息交换方式

- CPU对外围设备的管理方式
- 程序查询方式
- 程序中断方式
- DMA方式
- 通道方式



外围设备的速度分级 信息交换方式



1、外围设备的多样性

外围设备种类繁多，有机械式和电动式，也有电子式和其他形式。

其输入输出信号，可以是数字式的电压，也可以是模拟式的电压和电流（A/D+D/A）。

各种外围设备的数据传输速率也相差很大。



2、外围设备输入输出过程

输入过程:

- (1) CPU把一个I/O地址放到地址总线上，选择指定的输入设备；
- (2) CPU等候输入设备的数据成为有效，放到数据总线上；
- (3) CPU从数据总线上读入数据，并放在相应的寄存器中。



2、外围设备输入输出过程

输出过程:

- (1) CPU把一个I/O地址放到地址总线上，选择指定的输出设备；
- (2) CPU把数据放在数据总线上；
- (3) 输出设备准备就绪，确认数据有效，从数据总线上把数据取走。



3、CPU与外围设备之间的定时

从外围设备输入输出过程分析中可以看出，“输入设备什么时候把数据准备好，放到数据总线上”；“输出设备什么时候准备就绪，确认数据总线上的数据有效”，有一个定时问题。

而且定时问题，与输入输出设备本身的速度关系很大，应按速度分级考虑。



3、CPU与外围设备之间的定时

(1) 速度极慢或简单的外围设备

对这类速度极慢或简单的外围设备，如机械开关、显示二极管，CPU只要根据需要进行发送或接收数据、无需任何前提条件，即**想发送就发送、想接收就接收**。

对机械开关来说，CPU可以认为输入的数据一定有效，因为其维持的时间足够CPU识别；

对显示二极管来说，CPU可以认为输出一定准备就绪，只要CPU给出数据，二极管就能显示。



3、CPU与外围设备之间的定时

(2) 慢速或中速的外围设备

由于这类设备的速度和CPU的速度不在一个数量级，或者由于设备（如键盘）本身是在不规则时间间隔下操作的，因此，CPU与这类设备之间的数据交换通常采用**异步定时方式**，即应答式数据交换。

CPU启动外设以后，不断询问外设的状态，只有外设回答“数据准备好”、“接收工作准备就绪”后，才能进行数据交换。



3、CPU与外围设备之间的定时

(3) 高速的外围设备

由于这类设备是以相等的时间间隔操作的，而CPU也是以等间隔的速率执行输入输出操作，这种方式称为**同步定时方式**。一旦CPU和外设发生同步，它们之间的数据交换便依靠同步时钟信号（闹钟）来控制。

如外设是一条传送2400位/秒的传输线，那么CPU每隔 $1/2400$ 秒执行一次串行的输入操作。



假设一个幼儿园老师带10个孩子，要给每个孩子分二块水果糖，要求孩子们把二块水果糖吃完。她可以采用哪些办法呢？

方法1： 她先给孩子甲一块糖，盯着甲吃完，然后再给第二块；接着给孩子乙，其过程与孩子甲完全一样。依次类推，直至到第10个孩子吃完。

程序查询方式

方法2： 每人发一块糖各自去吃，并约定谁吃完就举手报告，再发第二块。

程序中断方式

方法3： 进行批处理：每人拿2块糖各自去吃，吃完2块糖后再向老师举手报告。

DMA方式

方法4： 权力下放：把发糖的事交给生活阿姨管，开始、结束时过问一下即可。

通道方式

外围处理机方式



信息交换方式

信息交换方式：无条件方式、程序查询方式、程序中断方式、DMA方式、通道方式、外围处理机方式。

适用范围：程序查询方式、程序中断方式适用于数据传输速率比较低的外围设备，而DMA、通道、PPU适用于数据传输速率比较高的设备。目前，程序中断方式和DMA方式多用在微机系统中，通道方式用在大型计算机系统中。

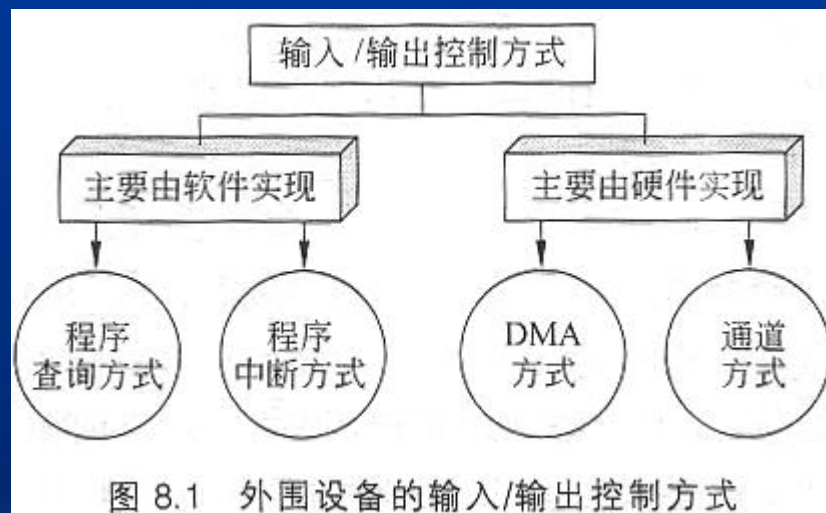


图 8.1 外围设备的输入/输出控制方式



CPU与I/O设备间的信息交换方式

- CPU对外围设备的管理方式
- 程序查询方式
- 程序中断方式
- DMA方式
- 通道方式



设备编址

输入/输出指令

程序查询方式的接口

程序查询输入/输出方式

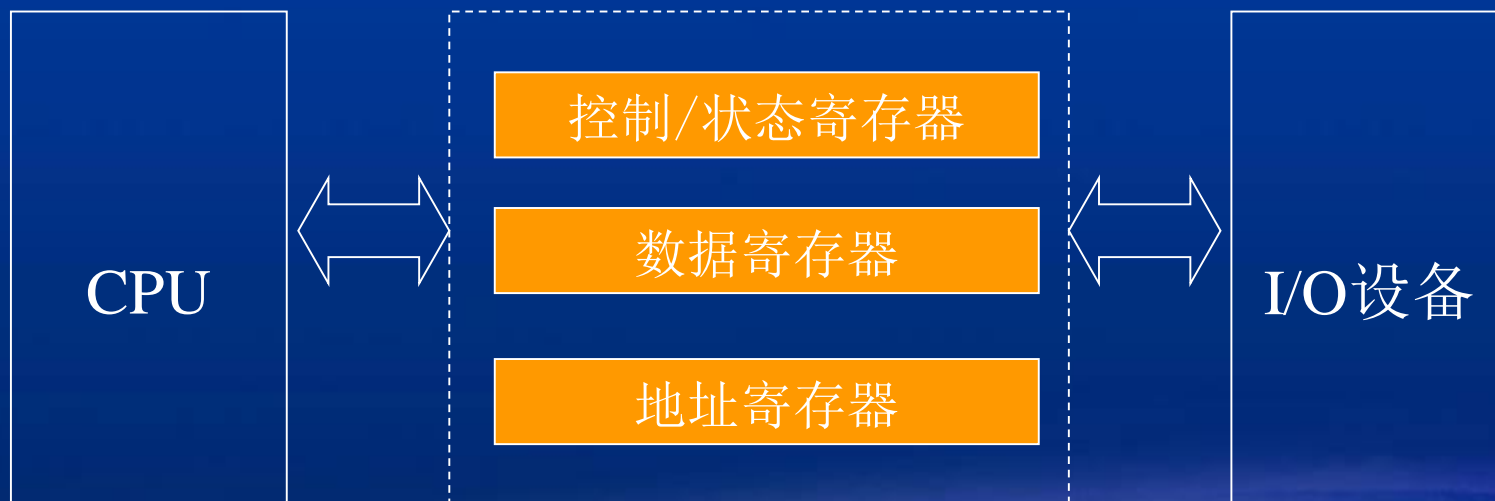


1、设备编址

I/O接口、I/O端口；

CPU与I/O设备之间交换的信息包括：地址信息、控制信息、状态信息、数据信息；

CPU访问I/O设备，其信息通过I/O接口中的I/O端口（实际上就是寄存器）传递。





1、设备编址

统一编址：I/O接口中的控制/状态寄存器、地址寄存器、数据寄存器等（也称为I/O端口）和内存单元同等看待，统一进行编址，这样可以使用访问内存的指令去访问I/O设备，但占用了部分内存地址空间。

单独编址：内存和I/O设备地址空间分开，访问内存和访问I/O设备使用不同的指令，或者说有专门的输入/输出指令。如8086内存地址空间00000H-FFFFFFH、输入输出地址空间0000H-FFFFH



2、输入输出指令

IN AL, 87H (0087H端口) → AL

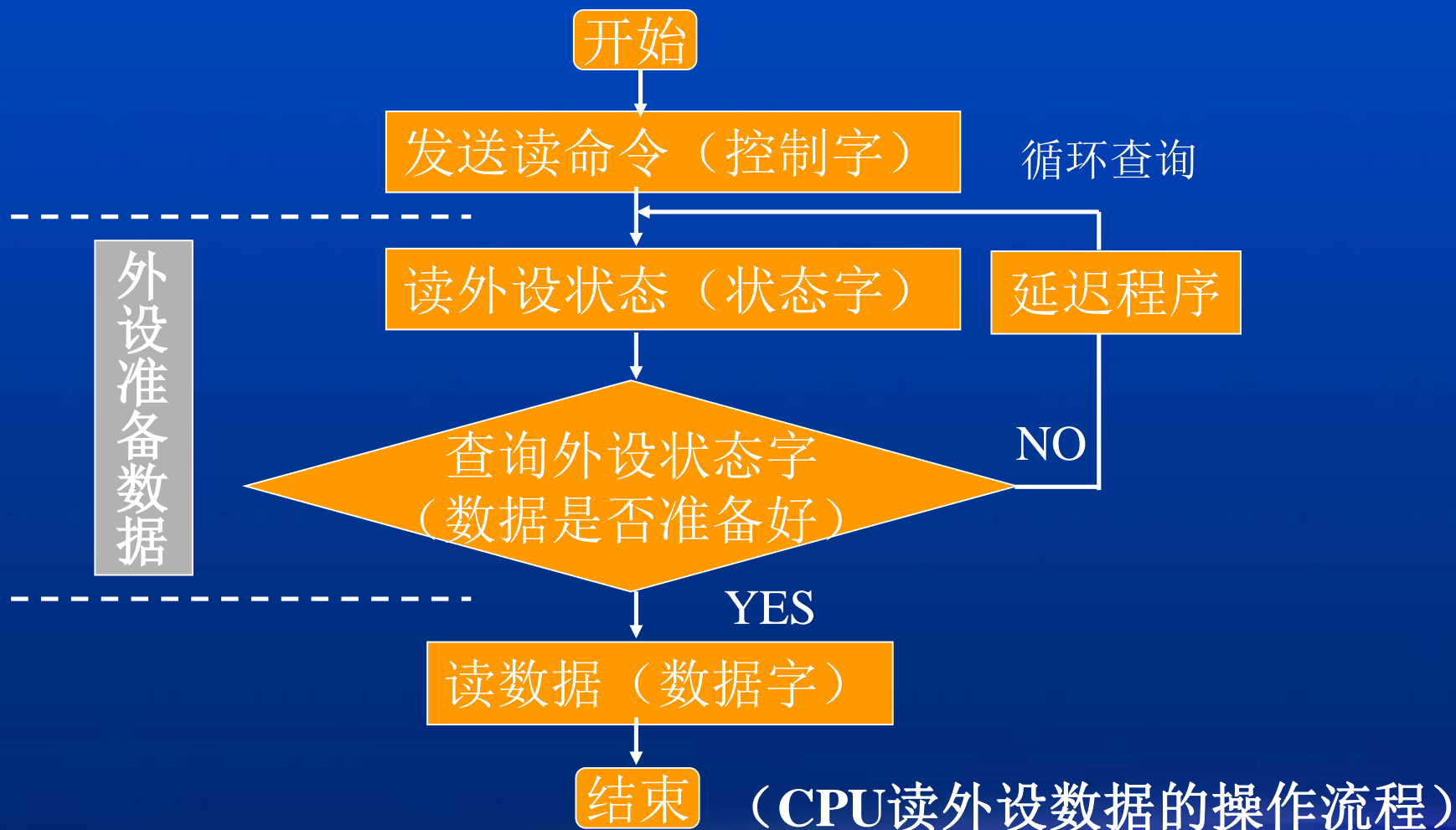
OUT 88H, AL (AL) → (0088H端口)

IN AL, DX (DX间接寻址端口) → AL

OUT DX, AL (AL) → (DX间接寻址端口)



3、程序查询方式的接口





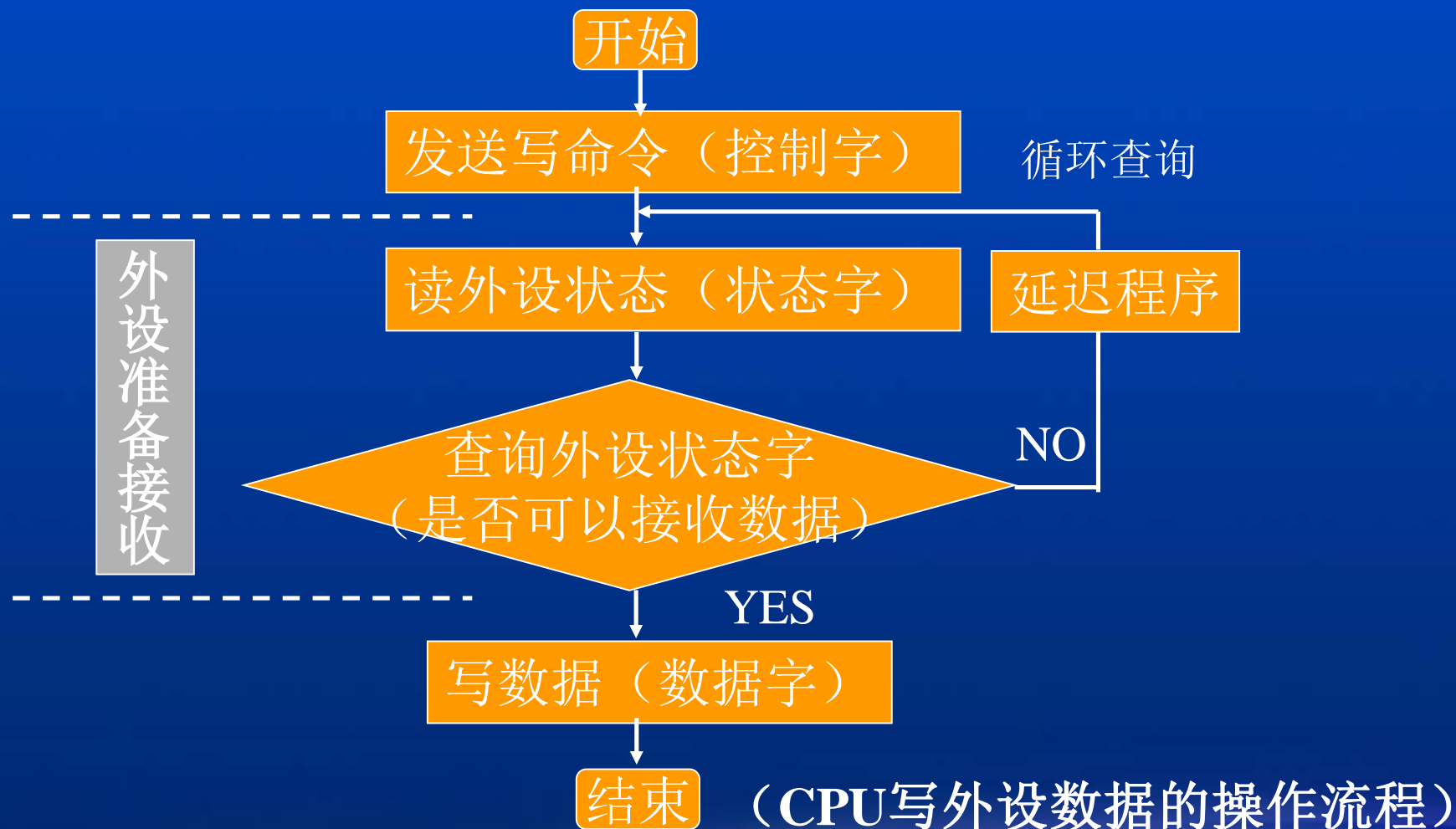
3、程序查询方式的接口

```
.....  
OUT 87H, 控制命令字; 控制端口0087H  
CALL delay1; 延迟一段时间, 外设准备数据  
start:: IN AL, 88H; 读状态字, 状态端口0088H  
IF AL.7=0 THEN  
    CALL delay2; 调用延迟子程序  
    GOTO start  
ELSE  
    IN AL, 89H; 读有效数据; 数据端口0089H  
    .....  
ENDIF
```

.....



3、程序查询方式的接口





3、程序查询方式的接口

- (1) **设备选择电路**：接到总线上的每个设备预先都给定了设备地址码（与地址总线的连接有关）。从地址总线上接收并暂存设备地址，并判断总线上呼叫的设备是否为本设备，若是则进入工作状态，否则不予理睬；
- (2) **数据缓冲寄存器**：暂存来自CPU/设备的数据；
- (3) **设备控制和状态标志**：来自CPU的启动信号、读写信号；来自设备的“数据准备好”、“准备就绪”、“错误”等状态信息。

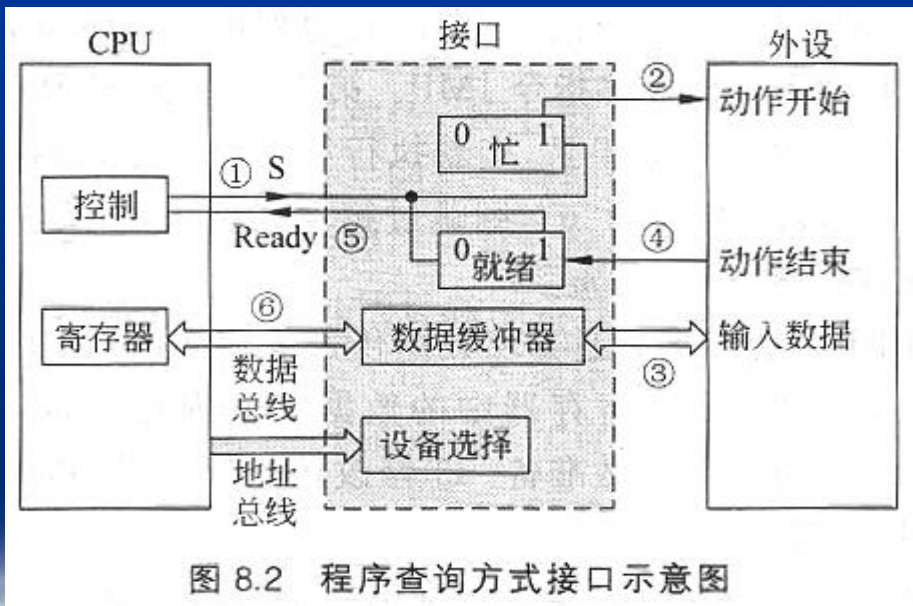


图 8.2 程序查询方式接口示意图



4、程序查询输入/输出方式

采用程序查询方式如果一个一个设备的处理，CPU资源浪费太严重。

通常做如下改进： CPU在执行主程序的过程中，可启动多个外部设备并行工作，然后周期性地调用各外部设备的查询子程序。若查询到某个设备准备好，则进入该设备的服务程序。

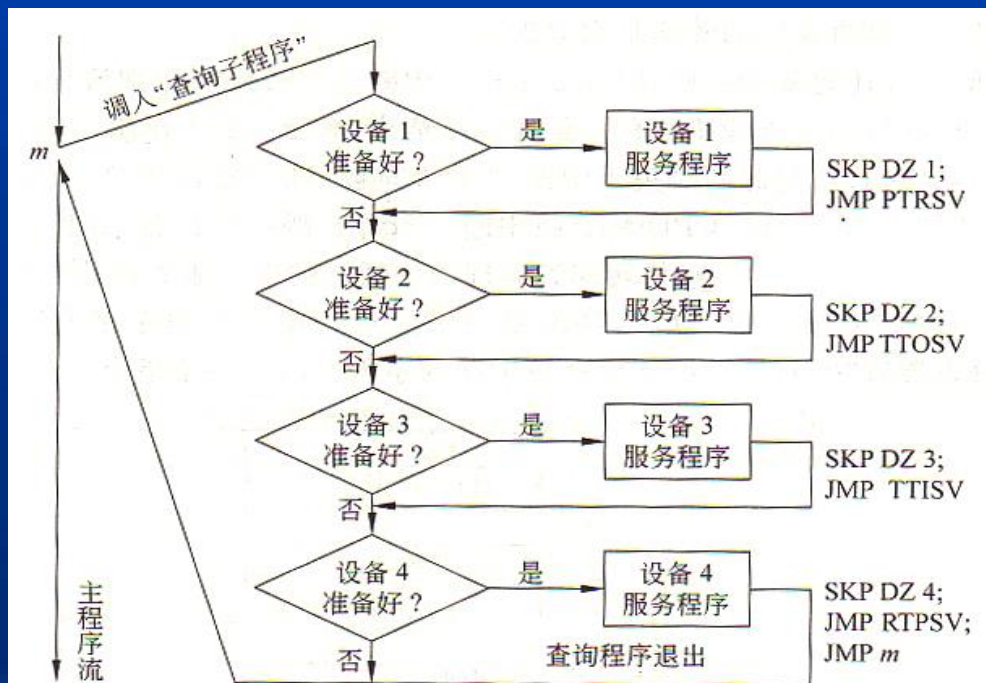


图 8.3 程序查询 I/O 设备流程图



CPU与I/O设备间的信息交换方式

- CPU对外围设备的管理方式
- 程序查询方式
- 程序中断方式
- DMA方式
- 通道方式



中断的基本概念

程序中中断方式的基本I/O接口

单级中断

多级中断



1、为什么要引入中断？

例：假设在程序查询方式下，打印机的打印速度是100字符/秒，那么当一个字符被写入到打印机的数据寄存器之后，CPU需要等待10ms才能把下一个字符写进去。而在这10ms中，CPU一直在执行循环等待，这样就造成了CPU时间的浪费。

解决方法：不让CPU在这10ms时间内干等着，而是让它去做一些其他事情，等打印机处理完下一个字符的时候再接着输出。

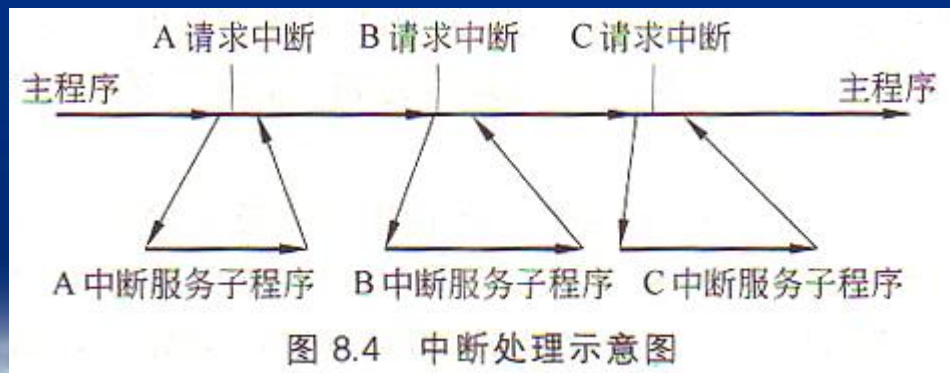


2、中断基本思想

CPU执行主程序，可同时启动多个外围设备并行工作，然后继续执行主程序。

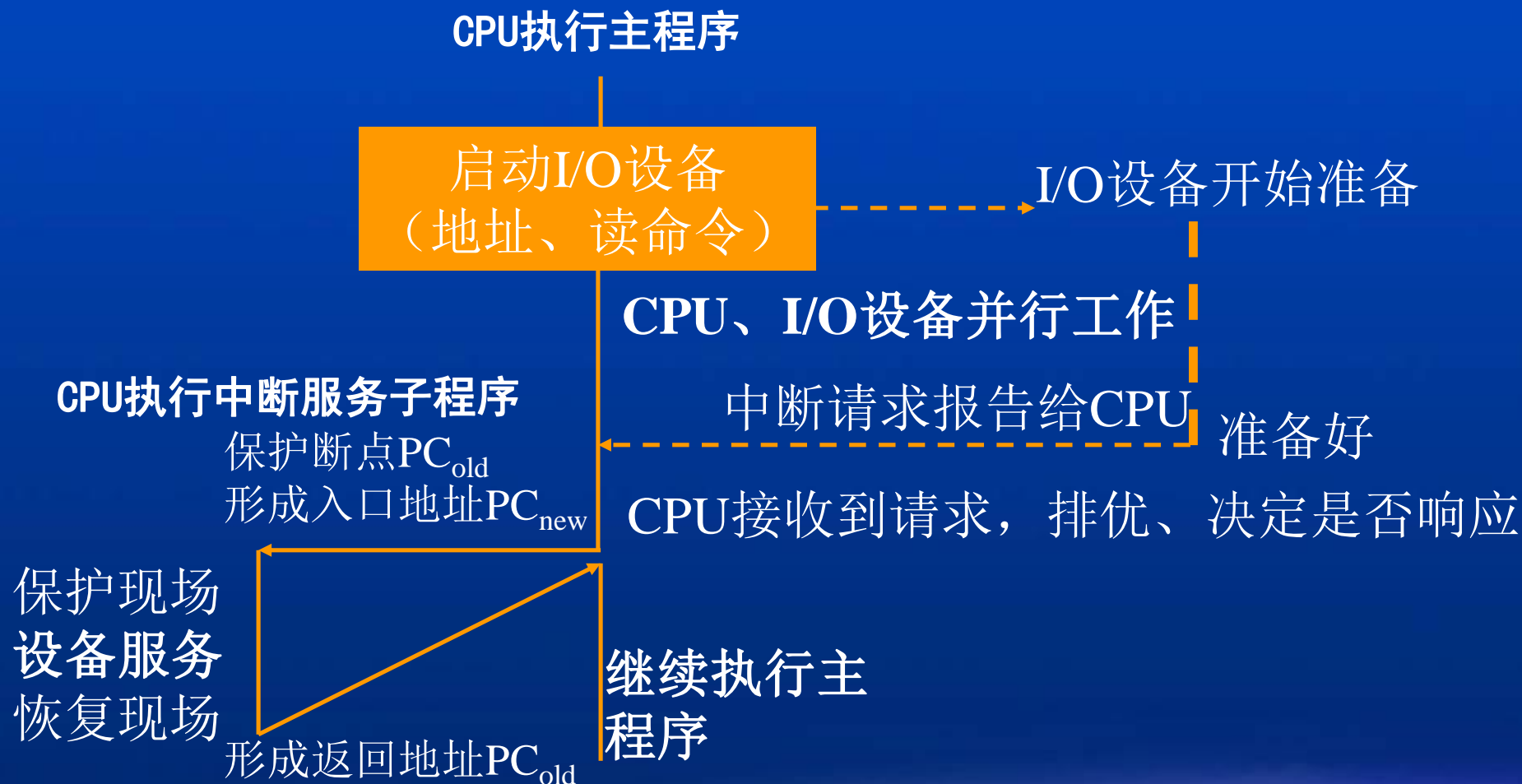
当某个外围设备准备好时，主动向CPU报告，然后CPU中断正在执行的主程序，执行中断服务子程序，进行信息交换和处理，结束后继续执行主程序。

中断方式特别适合外围设备准备时间不确定、随机事件的服务。



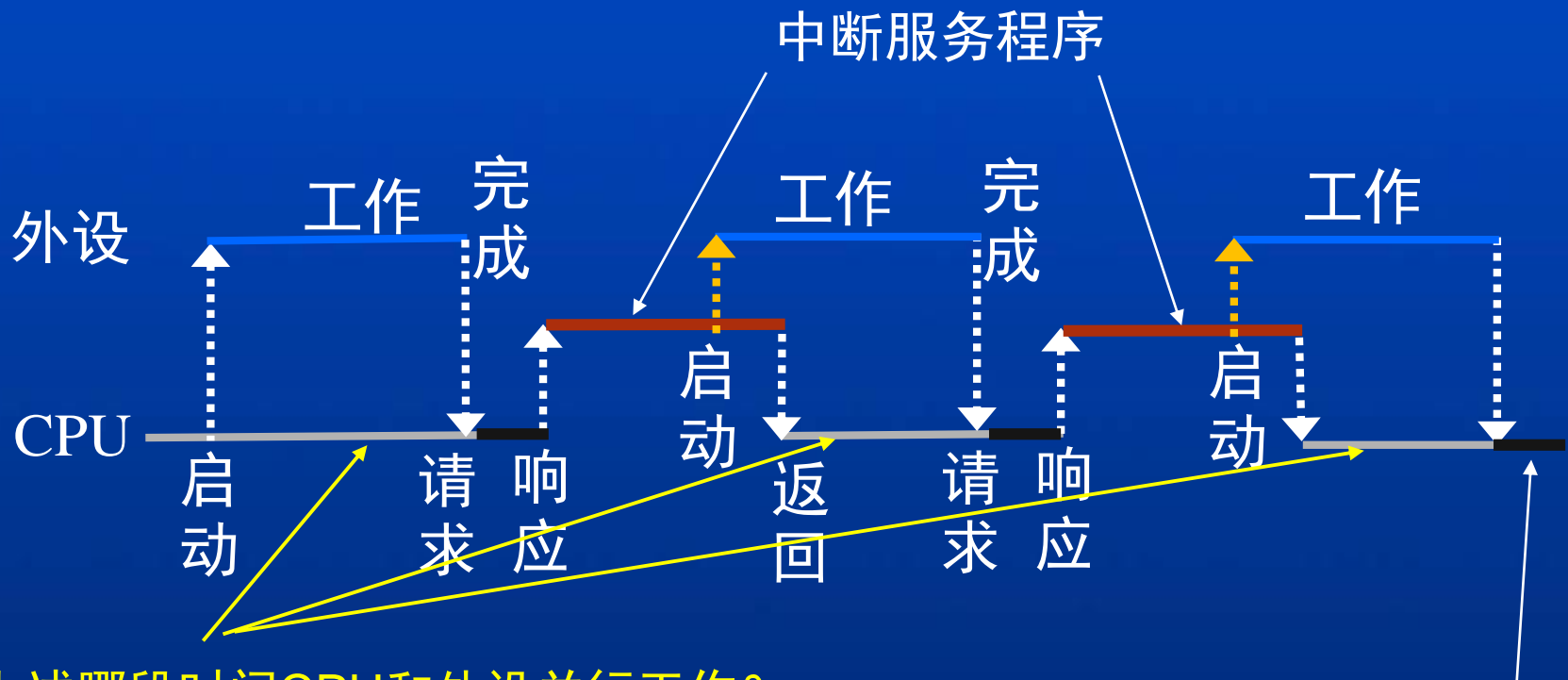


2、中断基本思想





2、中断基本思想



上述哪段时间CPU和外设并行工作？

程序切换（响应中断过程）由硬件完成，即执行“中断隐指令”，时间为



3、中断处理过程

(1) 一条指令完整执行完以后，查询是否有中断请求；

(2) 如果有中断请求，则判断是否响应该中断（优先级等）；

(3) 关中断（中断屏蔽置位）：单级中断系统中，一个中断被响应，必须该中断处理结束以后才能响应其它中断请求；多级中断系统中，允许中断嵌套，即允许高优先级中断低优先级的服务子程序；

(4) 找出中断源：确定是哪个设备的中断请求；

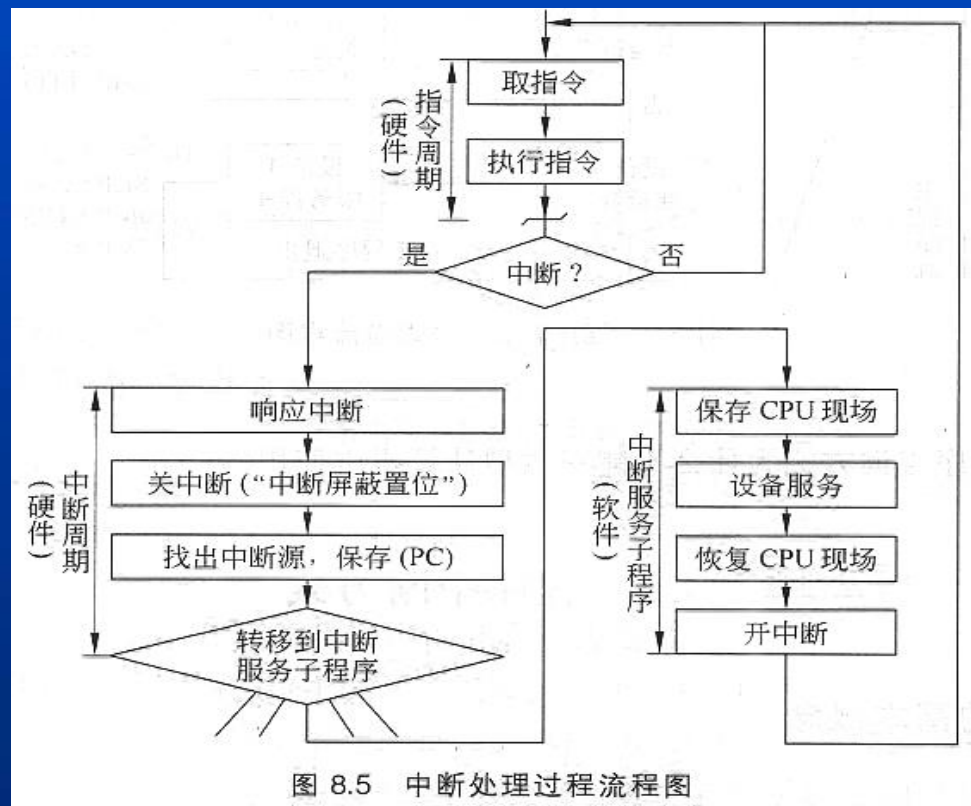


图 8.5 中断处理过程流程图



3、中断处理过程

- (5) 保存PC：即保存返回地址，硬件实现自动推入堆栈保存；
- (6) 形成中断服务子程序的入口地址：常采用向量中断法；
- (7) 保存CPU现场：标志寄存器、通用寄存器内容送入堆栈保存；
- (8) 设备服务：CPU与I/O设备之间交换一个机器字的数据；
- (9) 恢复CPU现场：从堆栈中把原来的状态寄存器、通用寄存器内容拿出来；
- (10) 开中断：本次中断请求处理完成，允许响应其它中断请求；
- (11) 中断返回：从堆栈中把返回地址拿出来，送PC。

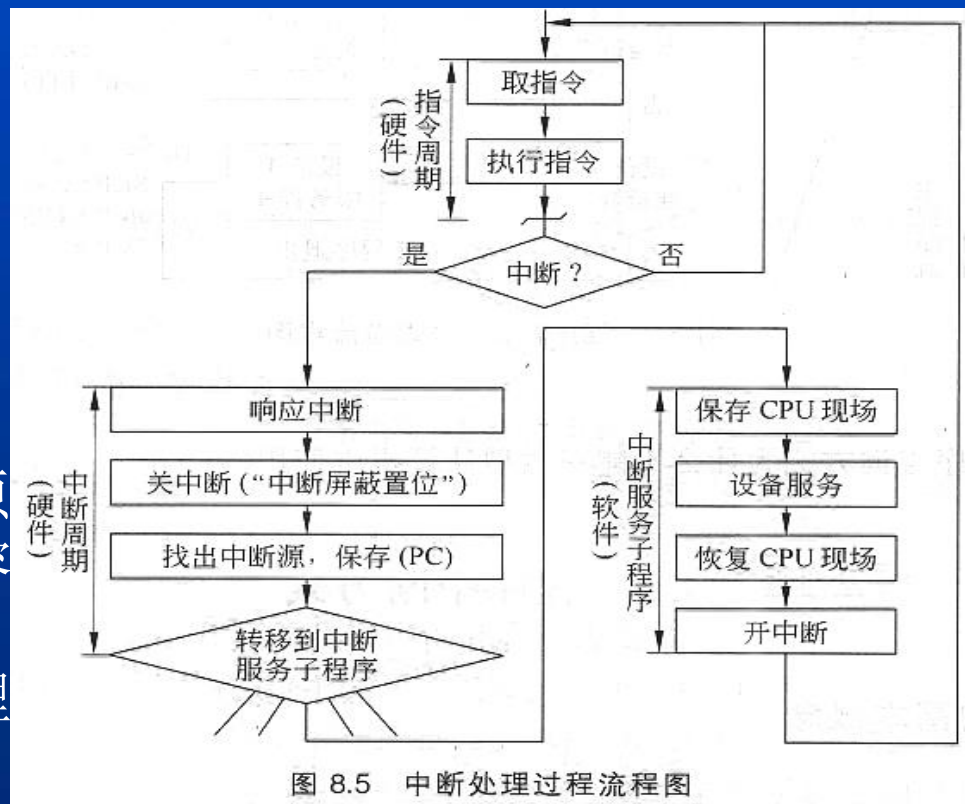


图 8.5 中断处理过程流程图



3、中断处理过程

主程序结构：

.....

MOV IE, XX; 中断允许设置

MOV IP, XX; 中断优先级设置

EI; 开中断

.....

; 随机中断

中断服务子程序结构：

DI; 关中断

PUSH AX

.....

PUSH DX; 保护现场

; 设备服务

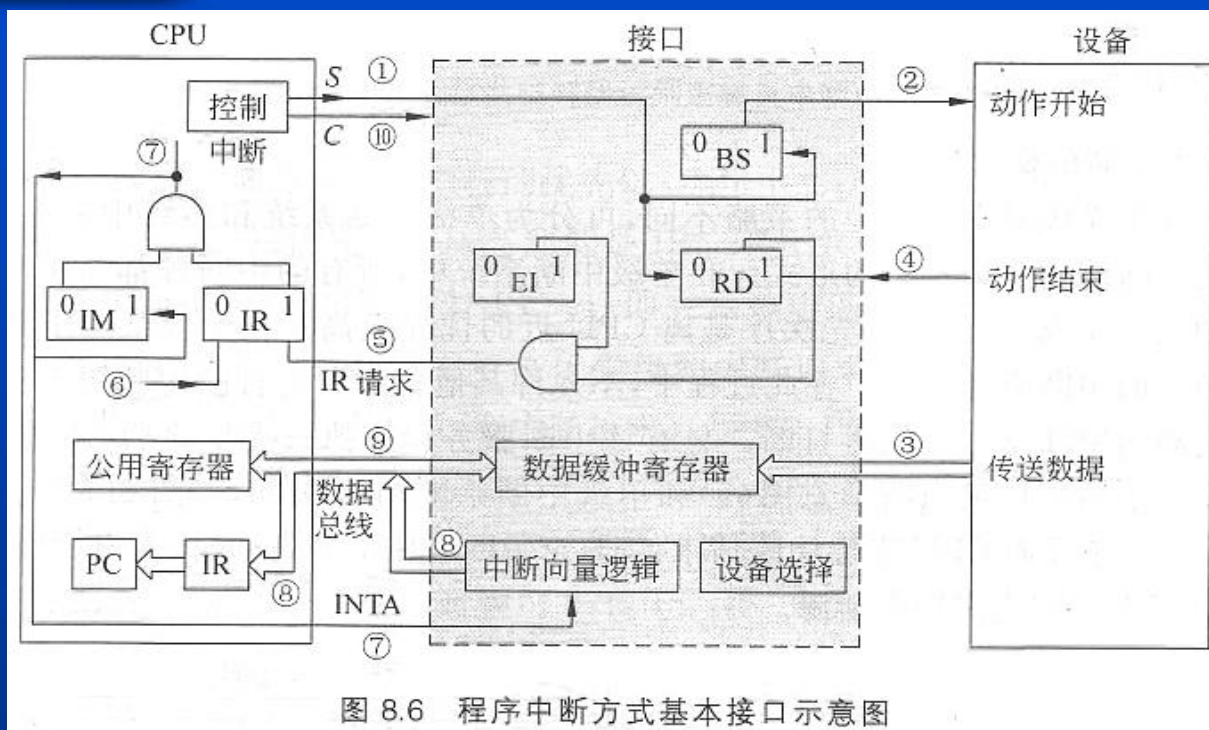
POP DX

.....

POP AX; 恢复现场

EI; 开中断

RETI; 中断返回



- (1) 启动外设S、置忙标志BS=1、准备就绪标志RD=0;
- (2) 通过BS标志启动外设，外设开始准备数据;
- (3) 数据准备好后，从外设送到数据缓冲寄存器，同时置准备就绪标志RD=1。
- (4) RD=1且EI=1，允许中断，形成中断请求IR=1，向CPU申请。
- (5) CPU公操作期间检测到中断请求，进入中断周期。

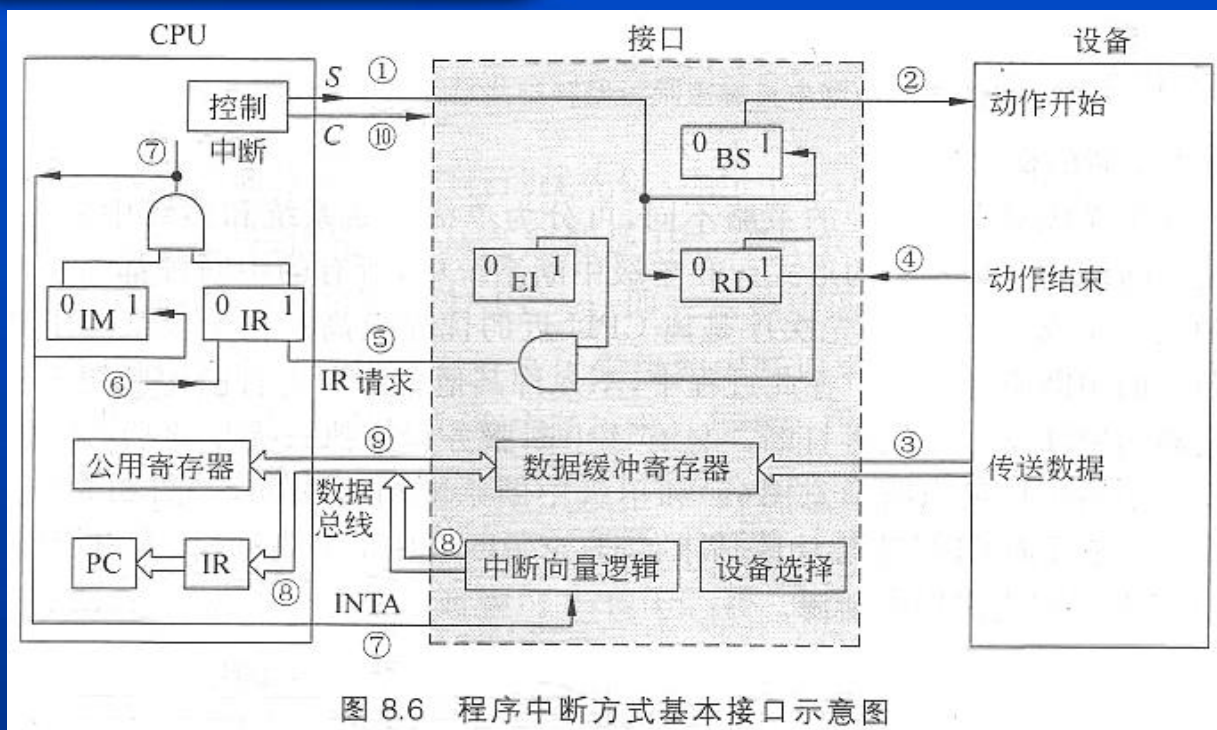


图 8.6 程序中断方式基本接口示意图

(8) CPU接收到中断向量，按一定的规则形成中断服务子程序的入口地址，改变PC、从而转到中断服务子程序执行。

(9) 中断处理结束，CPU输出C复位信号使BS=0、RD=0，一次中断操作过程结束。

(6) 中断周期：IR=1且IM=0，则响应中断，中断响应信号INTA输出到接口的中断向量逻辑。

(7) 中断向量逻辑根据设备号，形成该设备唯一的 interrupt 向量，经过DBUS到CPU内部寄存器。

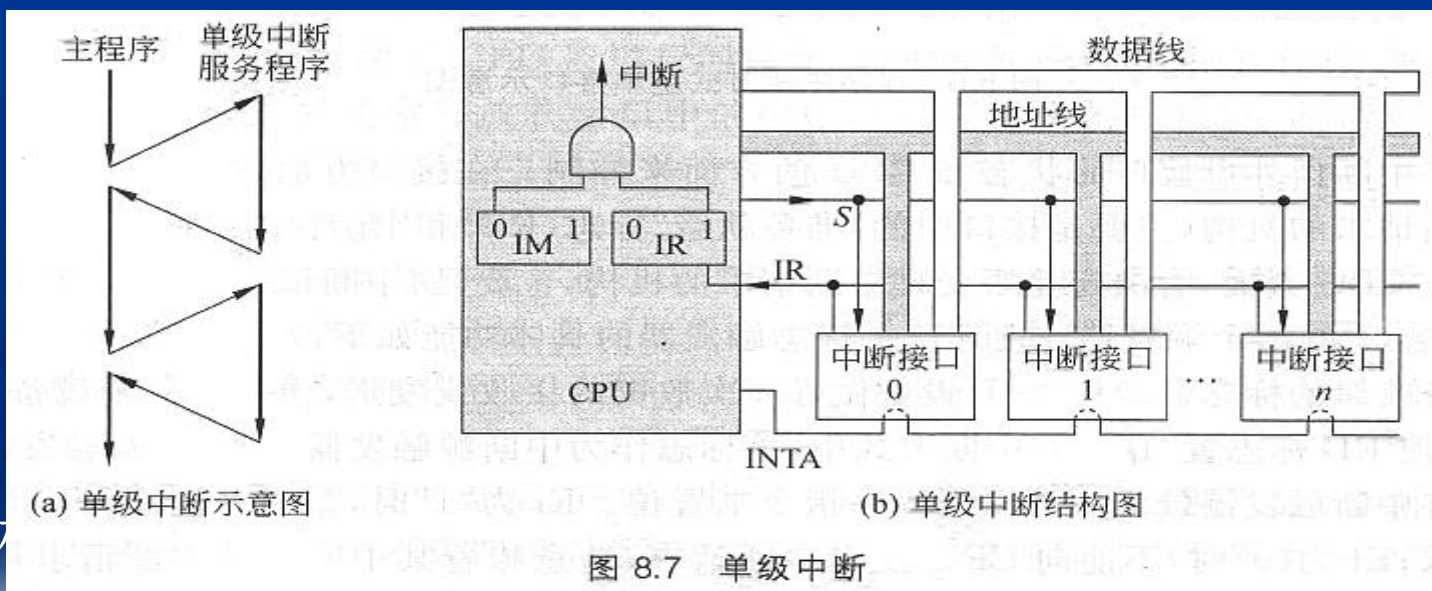


1、单级中断的概念

单级中断系统中，所有中断源都属于同一级。

当响应某个中断请求、执行该中断源的服务子程序时，不允许其它中断源再打断该中断服务子程序。

多个中断源发出中断请求IR，CPU发出中断响应信号INTA，采用链式查询方式沿“菊花链”寻找中断源。

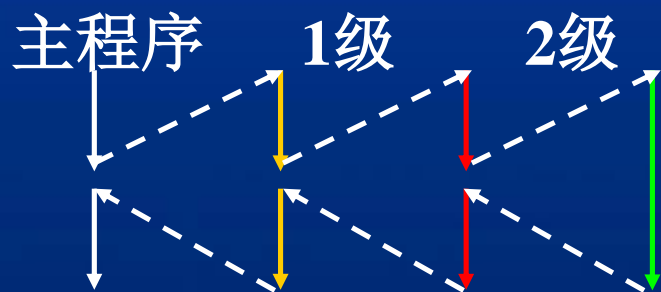




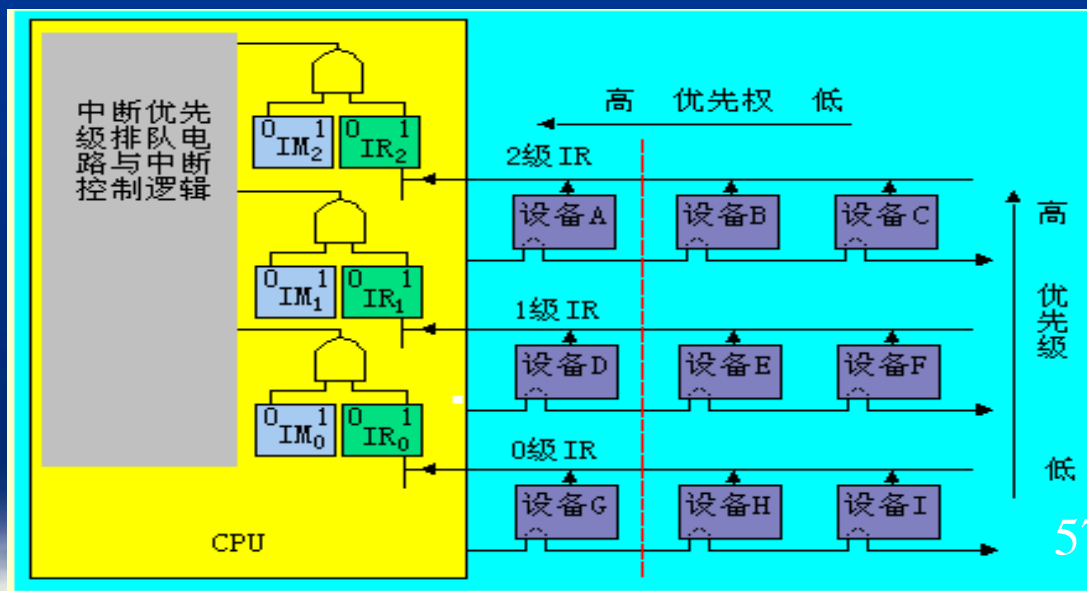
2、多级中断的概念

计算机系统的中断源，根据各中断源事件的轻重缓急程度不同而分成若干级别，每一中断级别分配一个优先权。

优先权高的中断级可以打断优先权低的中断服务子程序，以程序嵌套方式进行工作。



计算机组成与体系结构

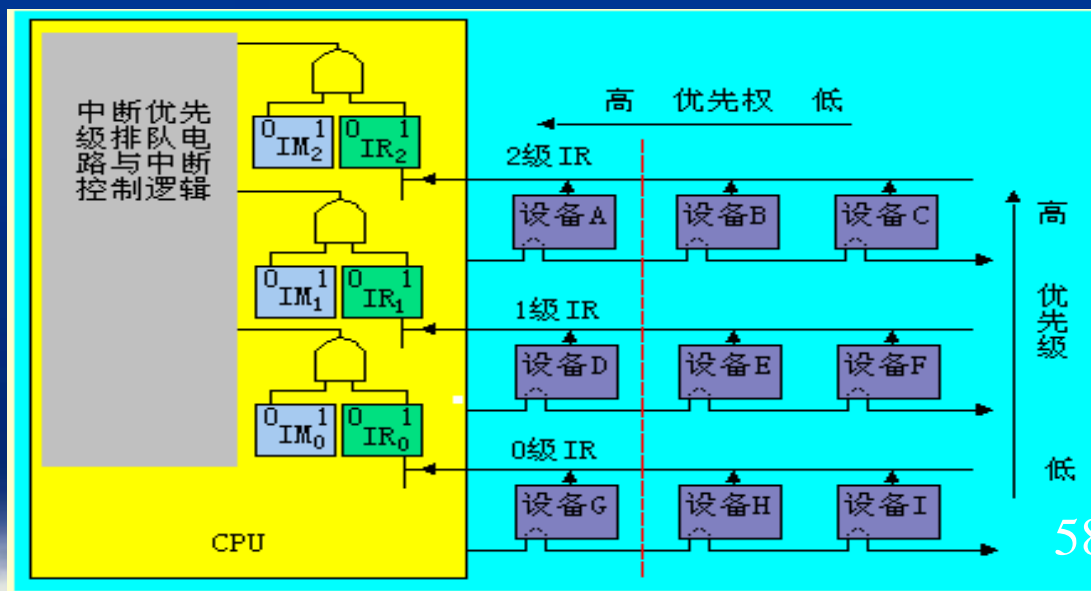




2、多级中断的概念

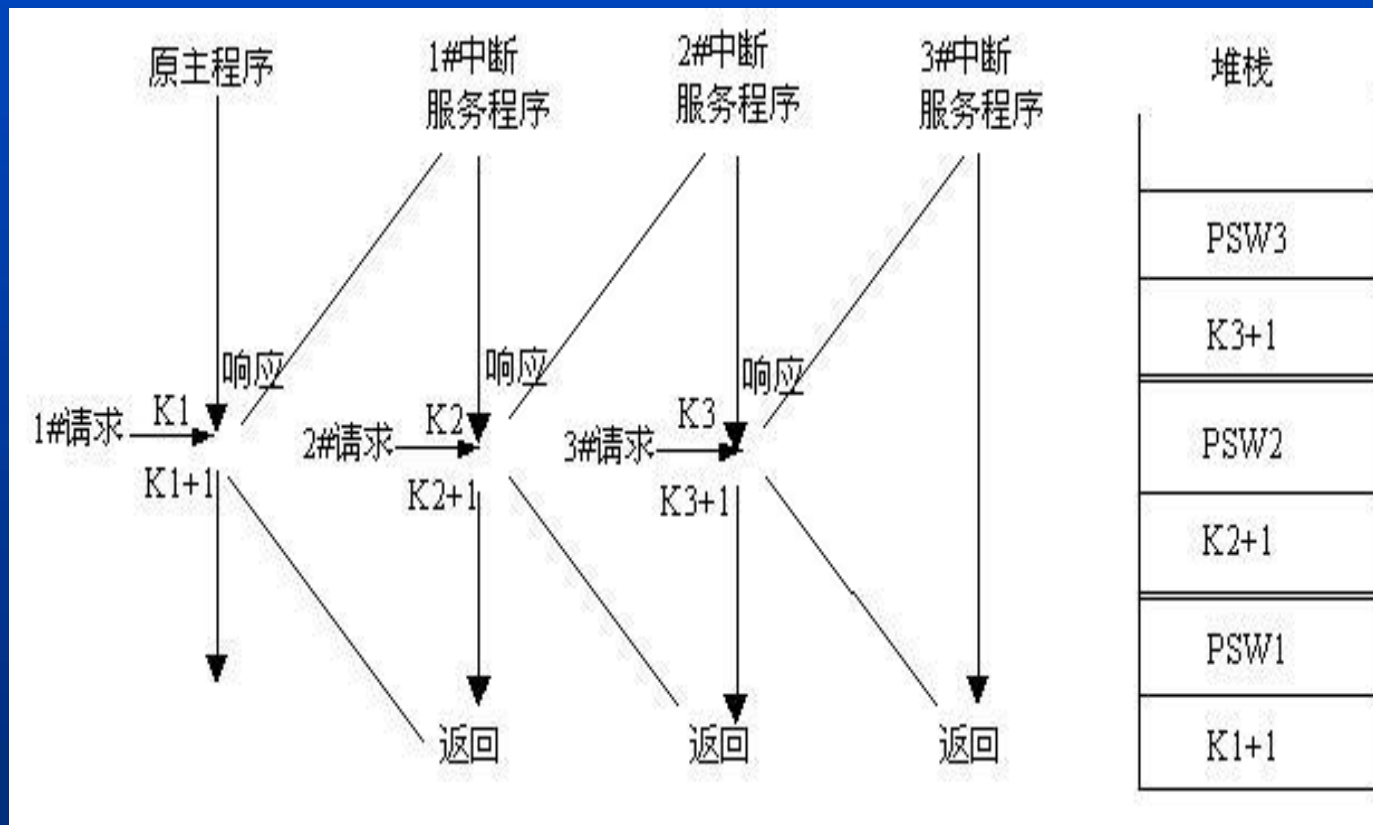
寻找中断源：首先，优先级排队，决定响应哪个优先级的中断请求；其次，链式查询，识别出究竟是选中优先级中的哪个设备提出的中断请求。

中断屏蔽：如果某级中断被响应，则需要修改中断屏蔽寄存器的内容，即将高优先级的对应标志位置“0”，而将本级、低优先级的标志位置“1”。





2、多级中断的概念



中断优先级的顺序是：

$3\# > 2\# > 1\#$

1#对2#开放（不屏蔽）

2#对3#开放（不屏蔽）



1、两类中断源

中断：常称为外部中断，由来自于CPU的中断请求线上的信号引发的。

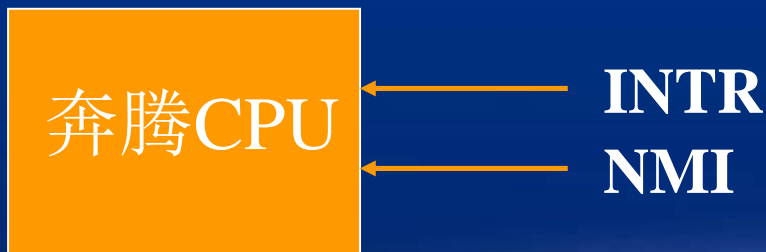
异常：常称为内部中断，由指令执行而引发的（如中断指令INT、除数为零）。



2、外部中断

可屏蔽中断：来自CPU的INTR引脚的中断请求信号，如果CPU中PSW. IF=1时，允许中断；PSW. IF=0时，中断请求信号在CPU内部被禁止。

非屏蔽中断：来自CPU的NMI引脚收到的中断请求信号而引发的中断，这类中断不能被禁止。一般用于计算机本身的故障处理系统，如掉电监测系统。





3、内部中断（异常）

执行异常： CPU执行一条指令过程中出现错误、故障等不正常条件引发的中断。如除数为零、溢出等。

执行软件中断指令： 执行INT n指令。执行过程中，只要是软件中断指令，肯定会中断，类似于子程序调用指令。



3、内部中断（异常）

[例]键盘输入一个字符。

```
MOV AH, 1
```

```
INT 21H
```

； 执行时肯定中断，去执行一个OS功能程序，从键盘接收一个字符，字符的ASCII码在AL寄存器中。

[例]显示一个字符。

```
MOV AH, 2
```

```
MOV DL, 显示字符的ASCII码
```

```
INT 21H
```

； 执行时肯定中断，去执行一个OS功能程序，在显示器上显示指定字符。



4、中断源

pentium共支持256个中断源。每个中断源给予一个唯一的编号，称为中断向量号或中断类型号（0-255、00H-FFH）。

对于一个确定的计算机系统来说，一部分中断向量号被硬件、指令系统、操作系统固定占用了，只有一部分中断向量号可以用户自定义使用。



5、中断服务子程序入口地址的组成

Pentium CPU 实模式（即访问1M内存，8086的内存地址管理模式）下，一个中断服务子程序入口地址由16位段地址、16位偏移地址构成，占用4B。

偏移地址 L

偏移地址 H

段地址 L

段地址 H



6、中断向量表IVT

将256个中断源的中断服务子程序入口地址集中保存在00000H-003FFH的1K区域中，称为中断向量表IVT。

00000H	偏移地址0# L	0#中断源
00001H	偏移地址0# H	
00002H	段地址0# L	
00003H	段地址0# H	
00004H	偏移地址1# L	1#中断源
00005H	偏移地址1# H	
00006H	段地址1# L	
00007H	段地址1# H	
00008H	-----	
-----	-----	
003FCH	偏移地址255#L	
003FDH	偏移地址255#H	
003FEH	段地址255# L	255#中断源
003FFH	段地址255# H	

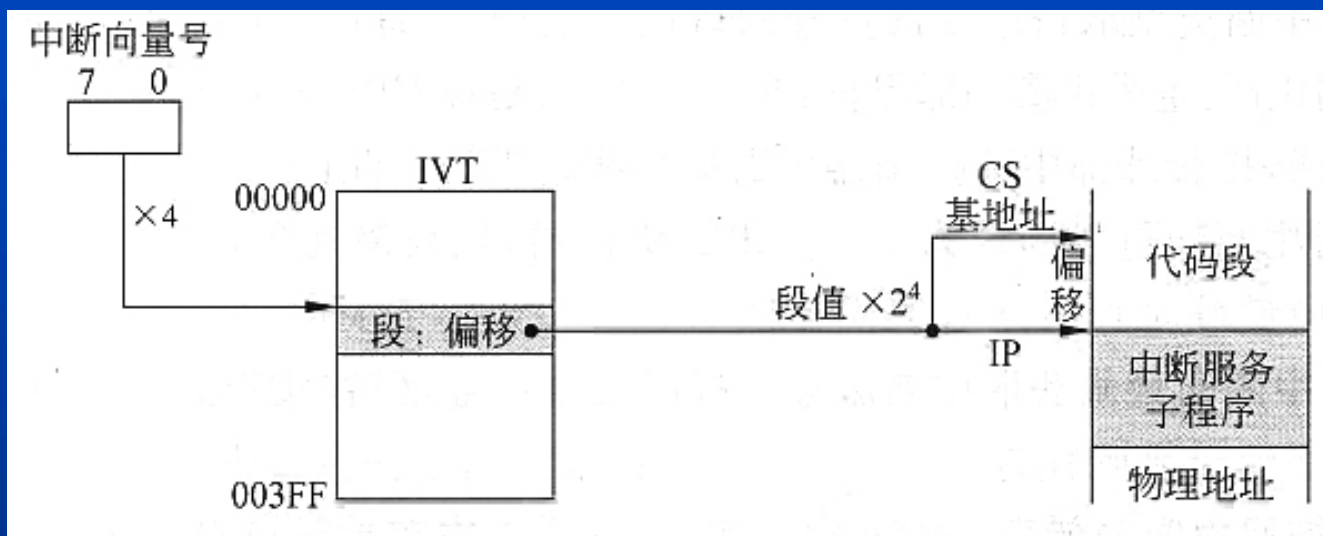


6、中断向量表IVT

中断 向量 号	入口地址		00000H	偏移地址0# L
			00001H	偏移地址0# H
	偏移地址	段地址	00002H	段地址0# L
			00003H	段地址0# H
0	$(4*0+1, 4*0)$	$(4*0+3, 4*0+2)$	00004H	偏移地址1# L
			00005H	偏移地址1# H
1	$(4*1+1, 4*1)$	$(4*1+3, 4*1+2)$	00006H	段地址1# L
			00007H	段地址1# H
...	00008H	-----
N	$(4*n+1, 4*n)$	$(4*n+3, 4*n+2)$	-----	-----
			003FCH	偏移地址255#L
...	003FDH	偏移地址255#H
			003FEH	段地址255# L
255	$(4*255+1, 4*255)$	$(4*255+3, 4*255+2)$	003FFH	段地址255# H



7、中断服务子程序入口地址形成



- (1) 中断周期获取中断向量号N;
- (2) 根据N, 查中断向量表IVT;
- (3) $(4N+1, 4N) \rightarrow (IP)$ 、 $(4N+3, 4N+2) \rightarrow (CS)$;
- (4) 中断服务子程序入口地址为: $16d * (CS) + (IP)$ 。



举例：假定某中断系统有四个中断源，其响应优先级为 $1>2>3>4$ 。假定在用户程序时同时发生1、3、和4级中断请求，执行3级中断服务程序时发生2级中断请求。分别写出处理优先级为 $1>2>3>4$ 和 $1>4>3>2$ 时各中断的屏蔽字及CPU完成中断处理的过程。

(1) 中断处理优先级为 $1>2>3>4$ 时：

中断程序级别	屏蔽字			
	1 级	2 级	3 级	4 级
第 1 级	1	1	1	1
第 2 级	0	1	1	1
第 3 级	0	0	1	1
第 4 级	0	0	0	1

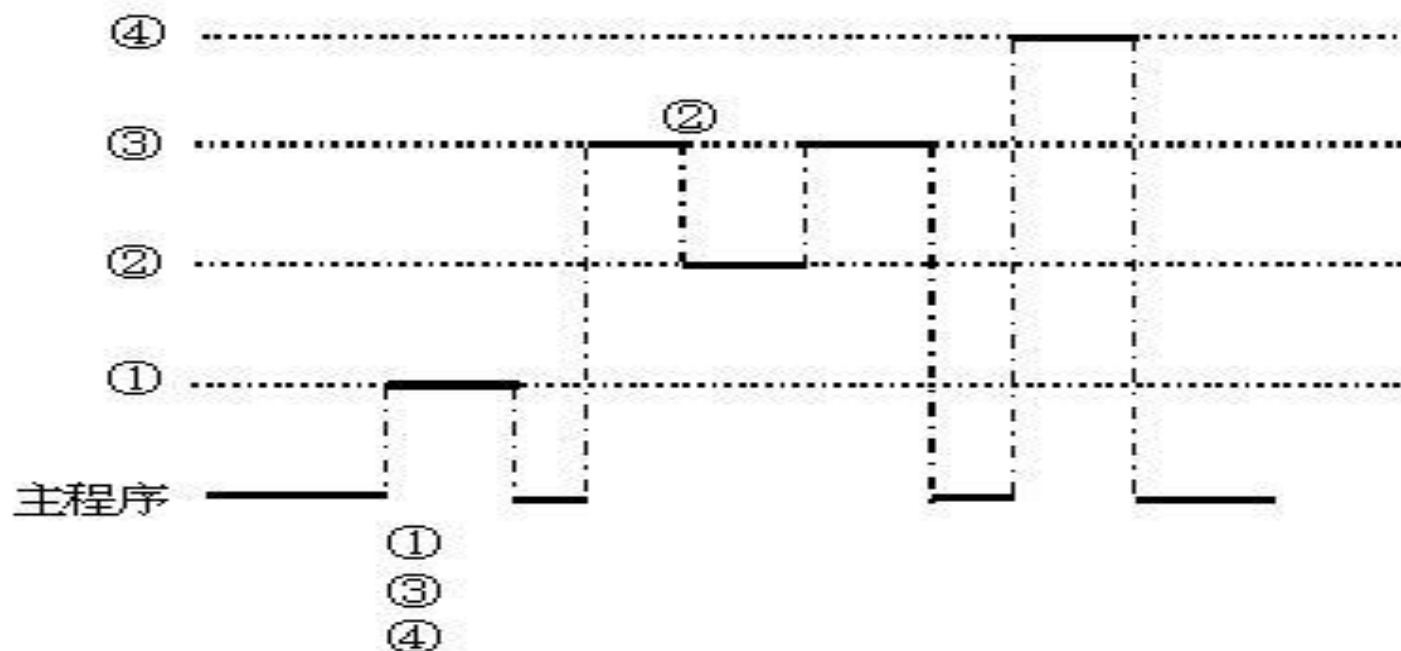
(假定 1 是屏蔽，0 是开放)



举例：假定某中断系统有四个中断源，其响应优先级为 $1 > 2 > 3 > 4$ 。假定在用户程序时同时发生1、3、和4级中断请求，执行3级中断服务程序时发生2级中断请求。分别写出处理优先级为 $1 > 2 > 3 > 4$ 和 $1 > 4 > 3 > 2$ 时各中断的屏蔽字及CPU完成中断处理的过程。

(1) 中断处理优先级为 $1 > 2 > 3 > 4$ 时：

中断服务程序





举例：假定某中断系统有四个中断源，其响应优先级为 $1>2>3>4$ 。假定在用户程序时同时发生1、3、和4级中断请求，执行3级中断服务程序时发生2级中断请求。分别写出处理优先级为 $1>2>3>4$ 和 $1>4>3>2$ 时各中断的屏蔽字及CPU完成中断处理的过程。

(2) 中断处理优先级为 $1>4>3>2$ 时：

中断程序级别	屏蔽字			
	1 级	2 级	3 级	4 级
第 1 级	1	1	1	1
第 2 级	0	1	0	0
第 3 级	0	1	1	0
第 4 级	0	1	1	1

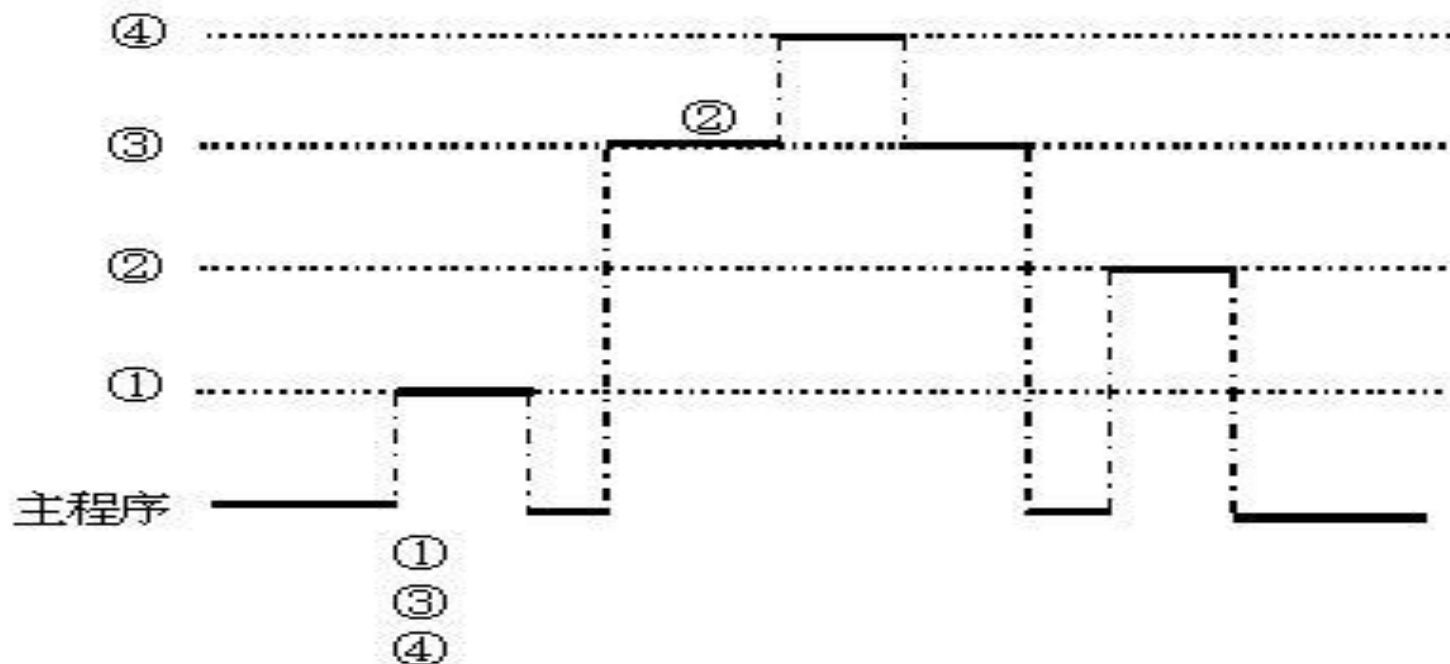
(假定 1 是屏蔽，0 是开放)



举例：假定某中断系统有四个中断源，其响应优先级为 $1>2>3>4$ 。假定在用户程序时同时发生1、3、和4级中断请求，执行3级中断服务程序时发生2级中断请求。分别写出处理优先级为 $1>2>3>4$ 和 $1>4>3>2$ 时各中断的屏蔽字及CPU完成中断处理的过程。

(2) 中断处理优先级为 $1>4>3>2$ 时：

中断服务程序

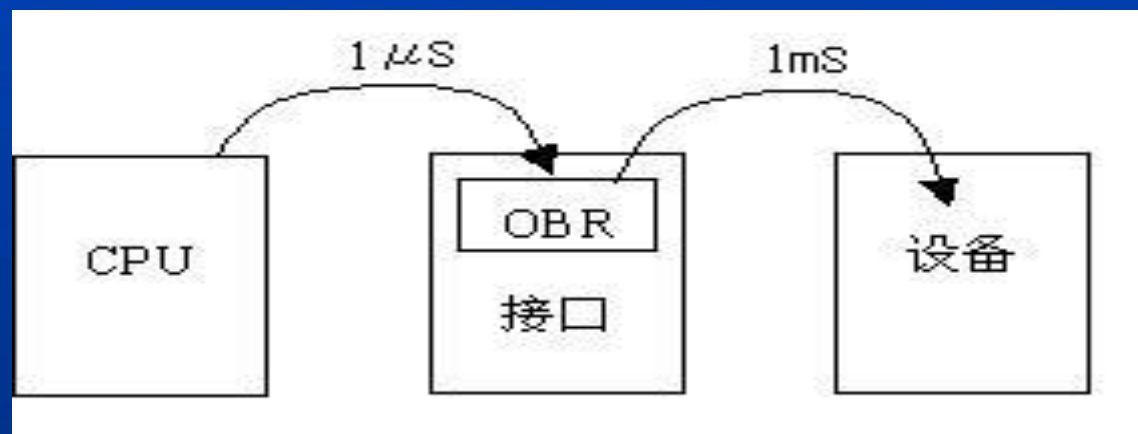




- 举例：假定某机控制一台设备输出一批数据。数据由主机输出到接口的数据缓冲器OBR，需要 $1\mu\text{s}$ 。再由OBR输出到设备，需要 1ms 。设一条指令的执行时间为 $1\mu\text{s}$ (包括隐指令)。试计算采用程序查询方式和中断方式的数据传输速度和对主机的占用率。

问题：CPU如何把数据送到OBR，I/O接口如何把OBR中的数据送到设备？

CPU执行I/O指令来将数据送OBR；而I/O接口则是自动把数据送到设备。



对主机占用率：

在进行I/O操作过程中，处理器有多少时间花费在输入/出操作上。

数据传送速度（吞吐量、I/O带宽）：

单位时间内传送的数据量。

假定每个数据的传送都要重新启动！即是字符型设备



(1) 程序直接控制传送方式

若查询程序有10条，第5条为启动设备的指令，则：

数据传输率为： $1/(1000+5)\mu s$ ，约为每秒995个数据。

主机占用率=100%

(2) 中断传送方式

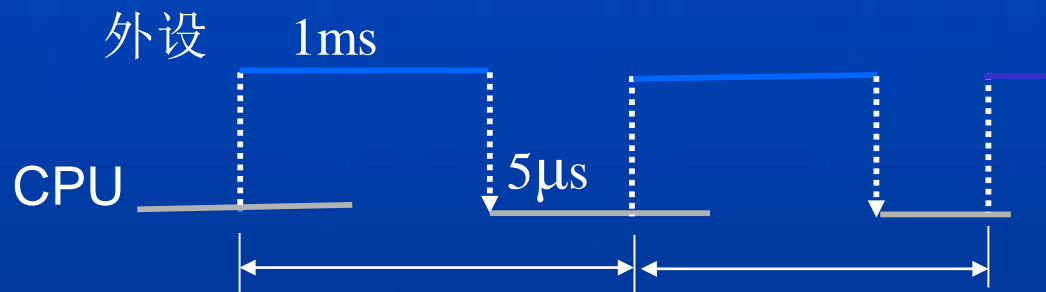
若中断服务程序有30条，
在第20条启动设备，则：

数据传输率为：

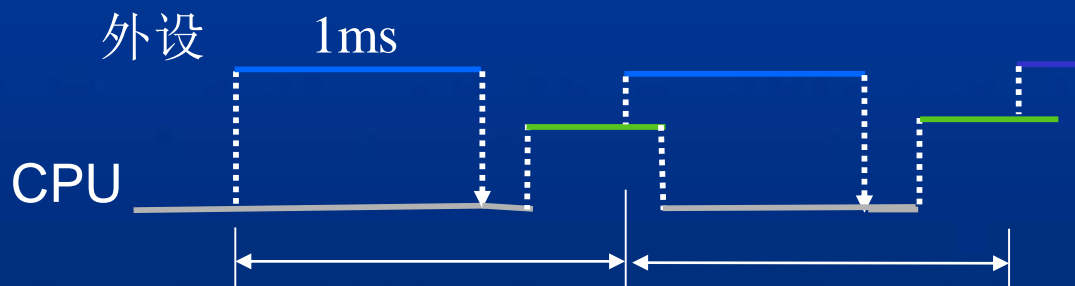
$1/(1000+1+20)\mu s$ ，约为
每秒979个数据。

主机占用率为：

$(1+30)/(1000+1+20)=3\%$



程序传送方式



中断传送方式

为什么中断服务程序比查询程序长？

因为中断服务程序有额外开销，如：保存现场、保存旧屏蔽字、开中断、（查询中断源）等



CPU与I/O设备间的信息交换方式

- CPU对外围设备的管理方式
- 程序查询方式
- 程序中断方式
- **DMA方式**
- 通道方式



1. DMA的基本概念
2. DMA传送方式
3. 基本的DMA控制器
4. 选择型和多路型DMA控制器



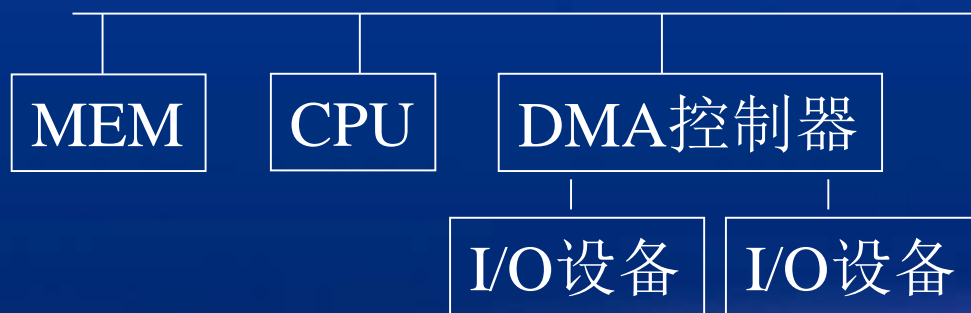
1.1 什么是DMA方式

- DMA的全称
 - Direct Memory Access（直接存储器存取/直接内存访问）
- 为什么要引入DMA方式？
 - 程序查询方式受“踏步”现象的限制，效率低下，不适合高速设备和主机间的数据传送。
 - 中断控制方式虽比程序查询方式有效，CPU和外设有一定的并行度，但由于下列原因也不适合高速设备和主机间的数据传送：
 - **对I/O请求响应慢**。每传送一个数据都要等待外设的中断请求，并增加许多中断响应和中断处理前、后的附加开销（保护断点、现场等），不能及时响应I/O请求。
 - **数据传送速度慢**。数据传送由软件完成（由CPU执行相应的中断服务程序来完成），速度慢。



1.1 什么是DMA方式

直接内存访问(DMA)方式是一种完全由**硬件**（称为DMA控制器）执行、在**I/O设备**和**内存**之间直接交换**批量信息**的工作方式，包括DMA请求、DMA响应、DMA传输、DMA结束处理等步骤。在DMA传输过程中，DMA控制器从CPU接管总线控制权，向内存发出地址和控制信号、修改地址、计数、以中断方式向CPU报告DMA传输结束，数据交换不经过CPU，而直接在内存和I/O设备之间进行。





1.1 什么是DMA方式

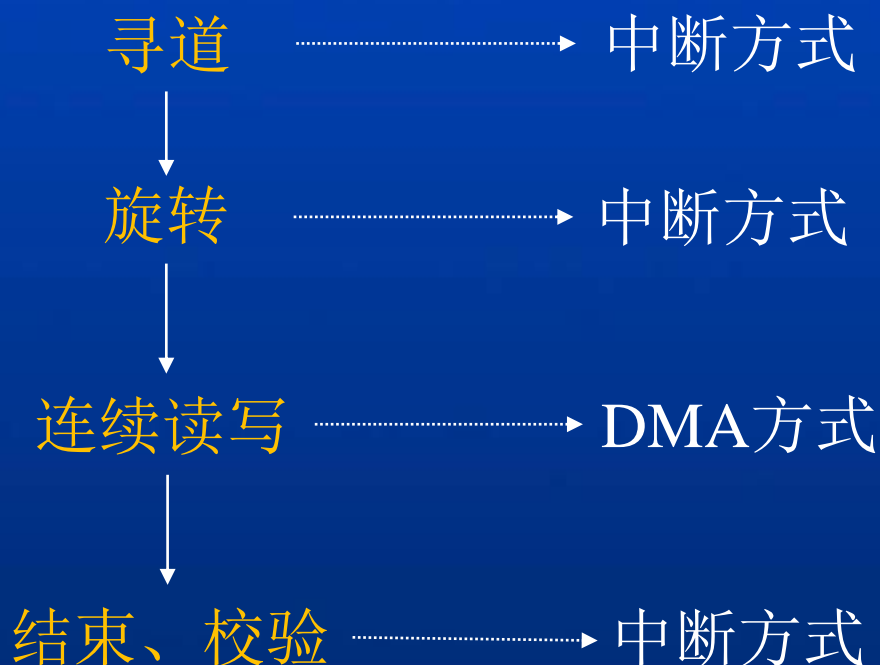
DMA方式的主要优点是**速度快**。由于CPU不参加传送操作，因此就省去了CPU取指令、取数、送数等操作；在数据传送过程中，没有保护现场、恢复现场之类的工作；内存地址修改、计数等工作由DMA控制器（硬件）直接实现。

DMA方式在计算机中广泛采用，适用于高速设备之间的批量数据交换，例如硬盘、光盘和内存之间的**数据块**交换。



1.1 什么是DMA方式

应用举例：磁盘和主存间数据交换





1.2 DMA的基本操作过程

(1) 外围设备发出DMA请求;

(2) **CPU**在指令执行公操作期间, 查询是否有DMA请求, 决定是否响应设备的DMA请求; 若响应请求, 把CPU工作改成DMA操作方式, CPU初始化DMA控制器(内存起始地址、字个数), DMA控制器从CPU接管总线控制权;

(3) **DMA控制器**负责执行一个个数据传送操作; 修改内存地址指针、计数; 数据块传送结束时以中断方式向CPU报告;

(4) **CPU**响应DMA传送结束的中断请求, 善后处理, 收回总线控制权, 一次DMA传送操作结束。



2.1 问题的提出

如果像DMA基本操作过程中所描述的那样，在DMA控制器负责的数据块传送过程中，由于CPU不能使用总线，当然就无法取指令，实际上什么事情都做不了，显然这段时间CPU的效率就没有发挥。

不管怎样，I/O设备每准备一个数据是需要时间的，相对于CPU速度来说时间较长。所以我们要研究：**在DMA数据块传送过程中，是否可以实现CPU和DMA控制器的并行操作、分时使用总线，从而提高CPU的利用率。**



2.2 三种数据传送方式

(1) CPU停止方式(成组传送)

DMA传输时，CPU脱离总线，停止访问主存，直到DMA传送一块数据结束。

(2) 周期挪用(窃取)方式(单字传送)

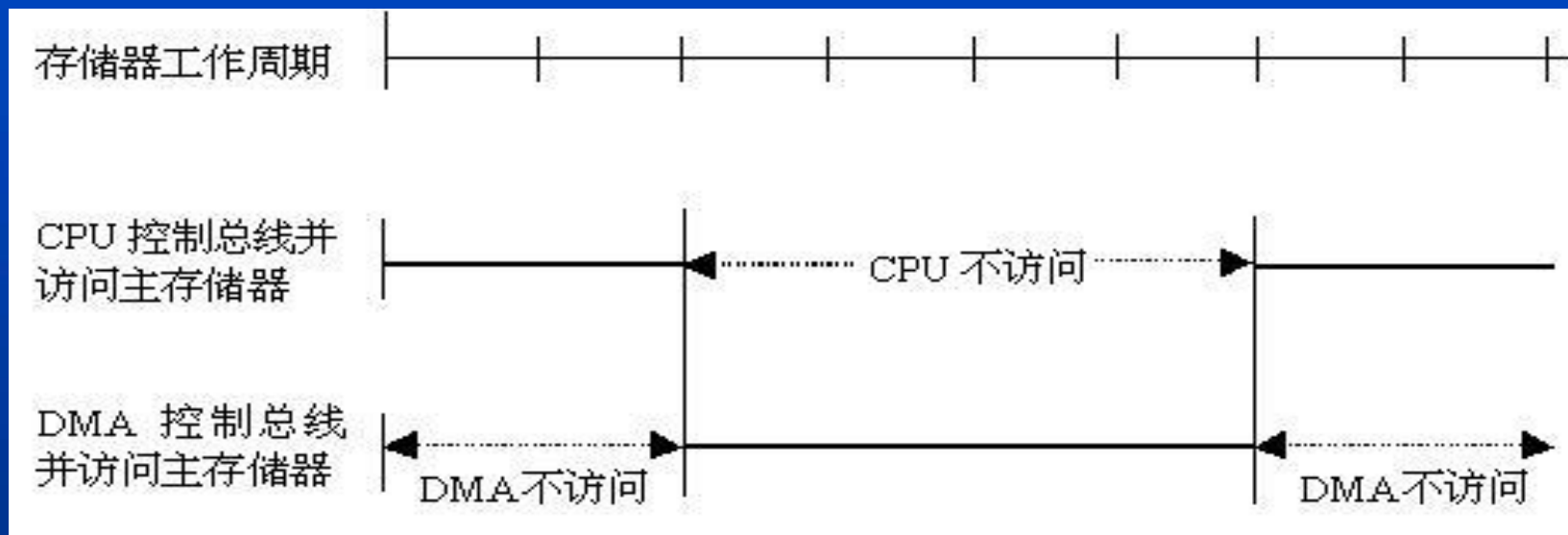
DMA传输时，CPU让出一个总线事务周期，由DMA控制总线来访问主存，传送完一个数据后立即释放总线。

(3) 交替分时访存方式

每个存储周期分成两个时间片，一个给CPU，一个给DMA，这样在每个存储周期内，CPU和DMA都可访问存储器。



2.2.1 CPU停止方式



优点：控制简单、适用于传输率很高的外设实现成组数据传送。

缺点：CPU工作受影响。DMA访存时CPU基本上处于停止状态。主存周期没有被充分利用。即使I/O设备高速运行，但两个数据之间的准备间隔时间也总大于一个存储周期，所以主存周期没有被充分利用。



2.2.1 CPU停止方式

- 弥补CPU停止方式缺点的做法：

- 在DMA接口中引入缓冲器

在DMA接口中采用一个小容量的半导体存储器，使I/O设备先和这个小容量存储器交换数据，然后再使用总线由小容量存储器与主存进行数据交换。这样可减少DMA传送数据时占用总线的时间，也就减少了CPU的等待时间。

- 采用周期挪用（窃取）方式

挪用一个小容量的存储周期进行外设和主存的一个数据交换。

每次DMA传送完一个数据就释放总线，使在外设准备下一数据时，CPU能抽空访问主存。



2.2.2 周期挪用方式

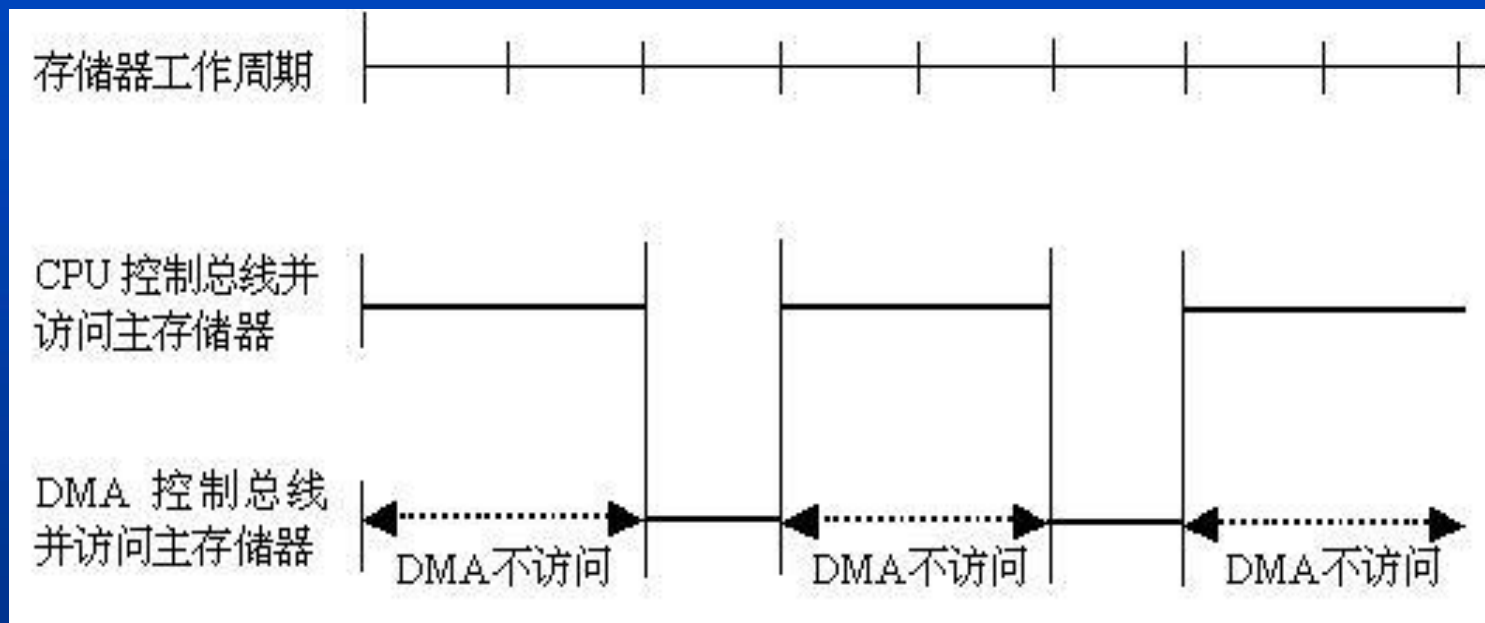
当CPU响应DMA请求、初始化DMA控制器之后，I/O设备去做准备，DMA控制器并不立即获得总线控制权，CPU继续获得总线控制权。

I/O设备每准备好发送/接收一个数据后，由DMA控制器向CPU申请获得一个总线周期的控制权，传输一个字数据，然后释放总线控制权交给CPU，I/O设备继续准备。

在整个DMA数据传送过程中，CPU和DMA控制器交替控制总线，可以充分发挥CPU和内存的效率，是DMA广泛采用的方式。



2.2.2 周期挪用方式



优点：既能及时响应I/O请求，又能较好地发挥CPU和主存的效率。这种方式下，在下一数据的准备阶段，主存周期被CPU充分利用。因此适合于I/O设备的读写周期大于主存周期的情况。

缺点：每次DMA访存都要申请总线控制权、占用总线进行传送、释放总线，因此，会增加传输开销。



2.2.2 周期挪用方式

- I/O设备要求DMA传送时可能会遇到以下三种情况之一：

- CPU不需访问主存

此时，不会发生冲突，两者并行。

如: CPU正在执行乘法指令，要花很长时间而不需马上访存。

- CPU正在访问主存

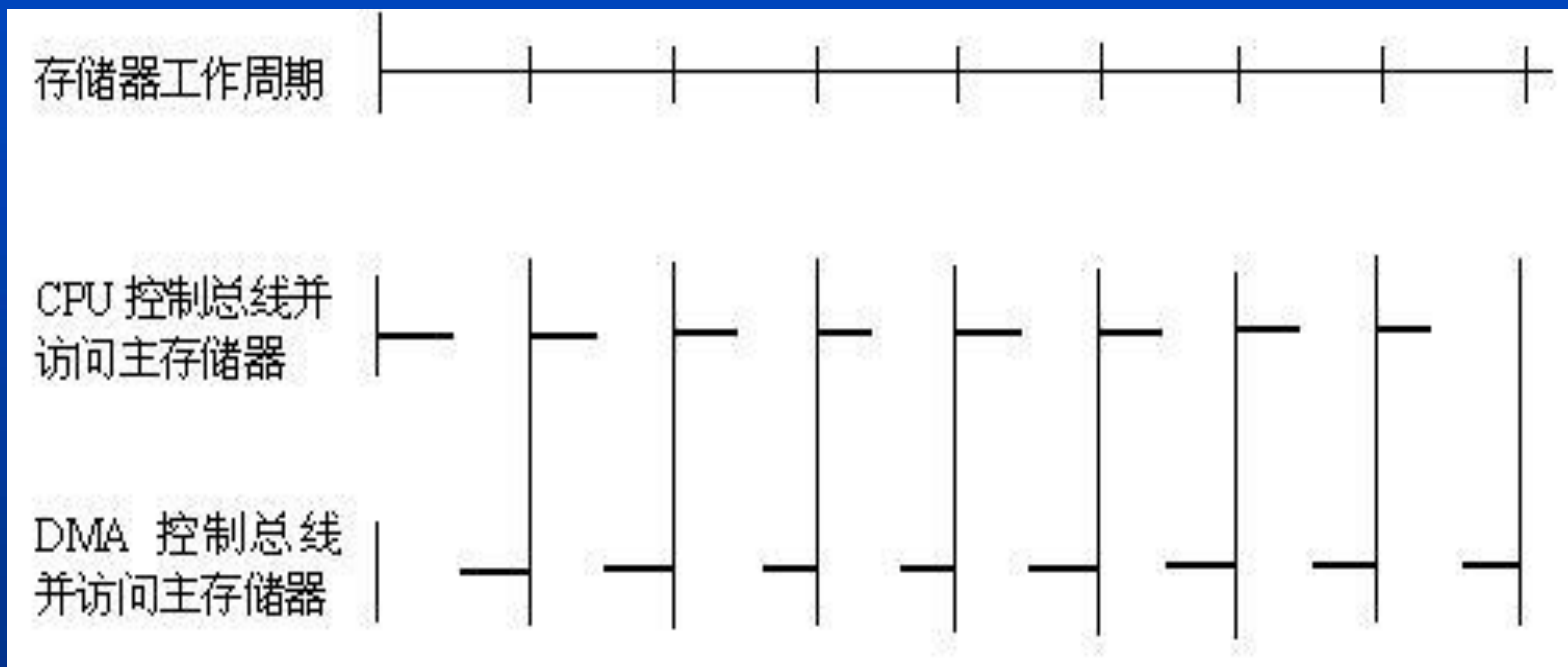
此时须等到存储周期结束，CPU让出总线，DMA才能访存。

- CPU也同时要访问主存

此时出现访存冲突。因为不马上响应DMA请求的话，高速设备可能会发生数据丢失，所以，DMA的总线优先权比CPU高。这时，先让DMA占用总线，窃取一个主存周期，完成一个数据的交换。这样，CPU便要延迟一段时间才能访存。



2.2.3 交替分时访存方式



特点：适用于CPU访存周期比主存存取周期更长的情况。

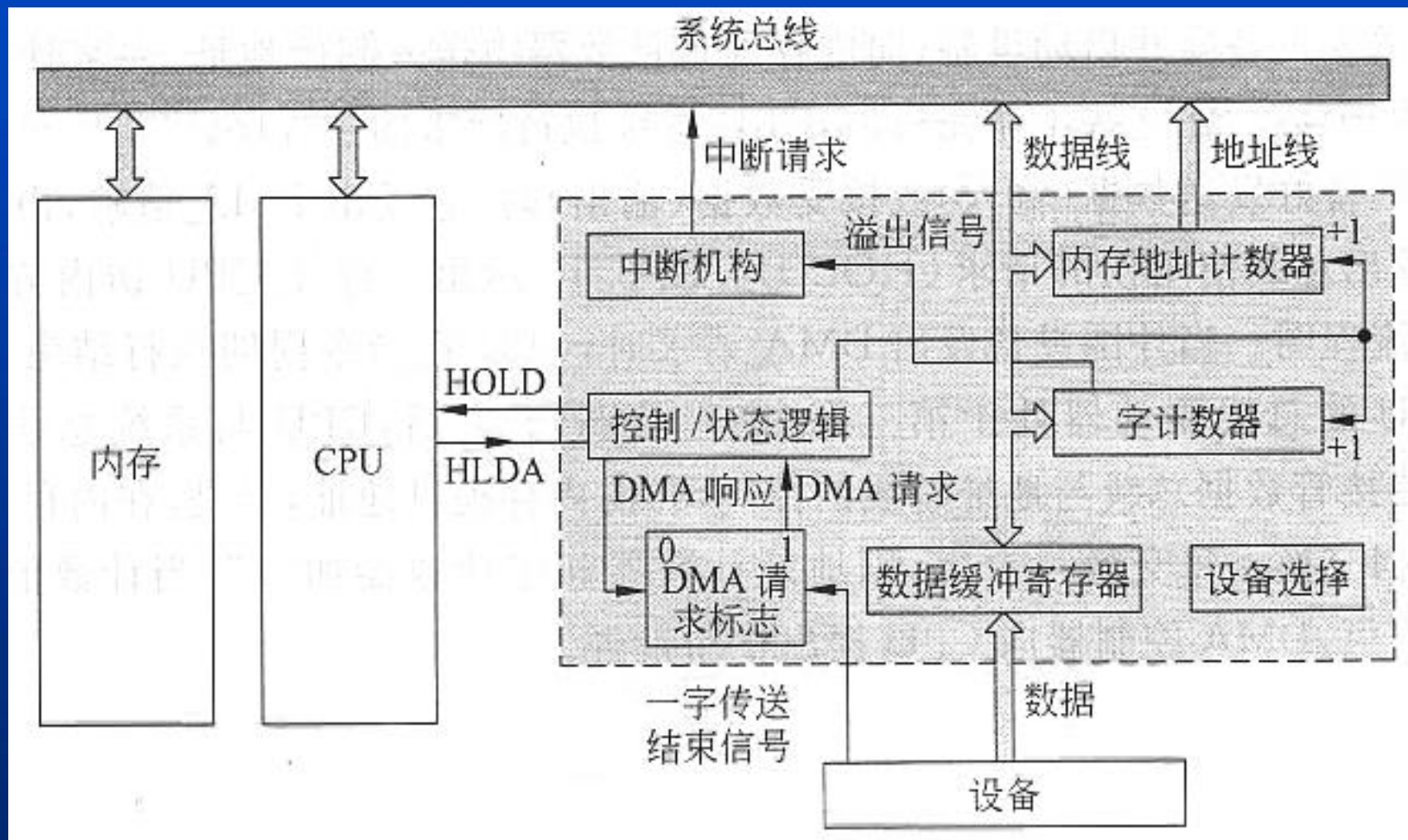
不需要总线使用权的申请和释放。

在这种方式下，CPU既不停止主程序运行也不进入等待状态，在CPU工作过程中，不知不觉完成DMA数据传送，故又被称为“透明DMA”方式。



3. 基本的DMA控制器

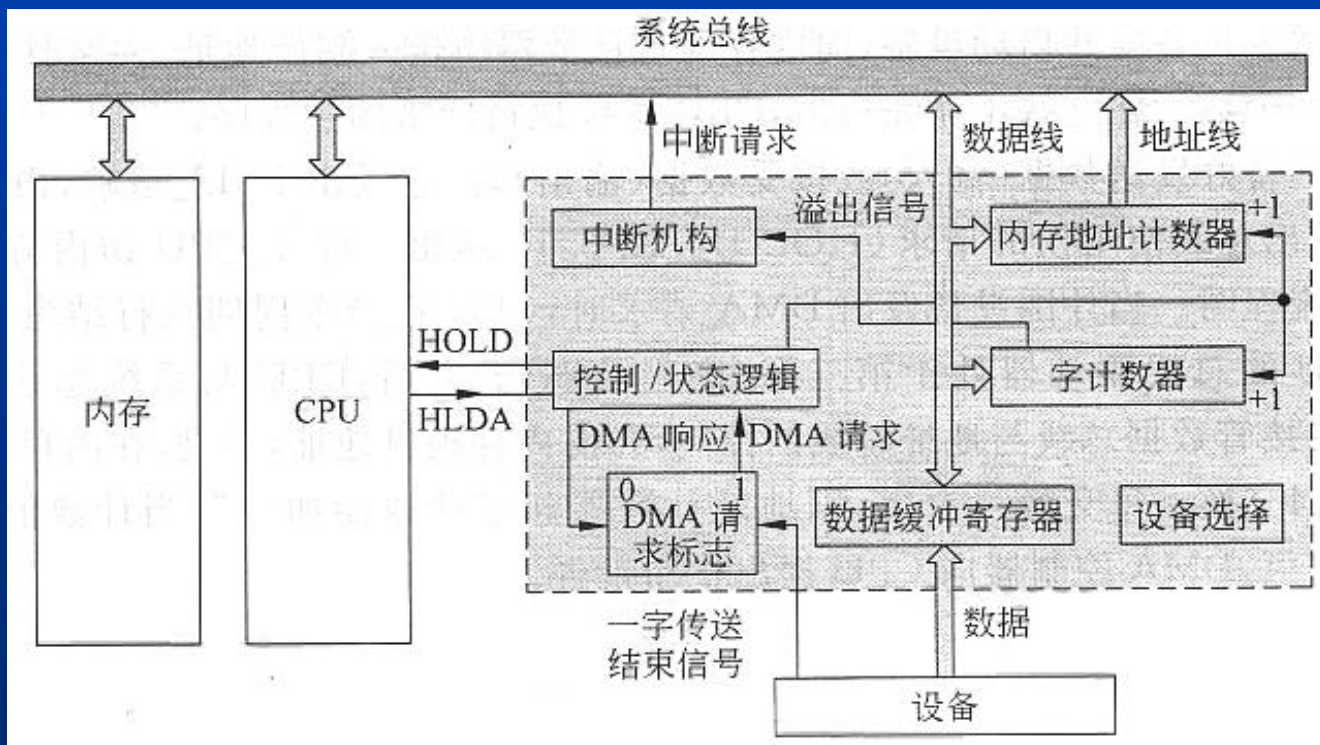
3.1 DMA控制器的基本组成





3.1.1 内存地址计数器

CPU初始化DMA控制器时，写入内存数据块起始地址。
每交换一个字数据后，地址计数器自动增量“加1”。





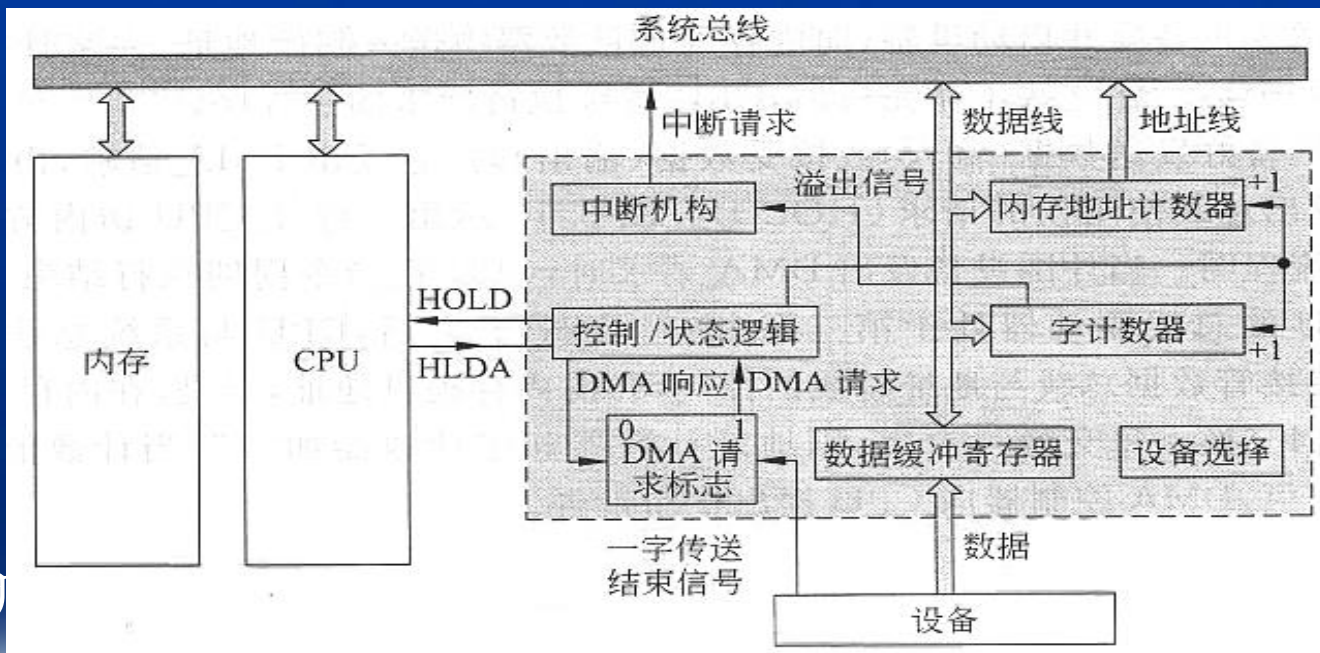
3. 基本的DMA控制器

3.1.2 字计数器

CPU初始化DMA控制器时，写入数据块长度初值。

初值通常以补码 ($2^n - x$) 形式表示，如字计数器8位，数据块长度125，则初值 $256 - 125 = 131$ 。

每传送一个字，字计数器加1。当计数器溢出时，表示数据块传送结束，溢出信号通过中断机构，向CPU申请中断。





3.2 DMA控制器的功能

DMA数据传送过程由DMA接口的控制逻辑完成，其功能为：

- (1) **请求**。能接收外设发来的“DMA请求”信号，并能向CPU发“总线请求”信号。
- (2) **响应**。当CPU发出“总线响应”信号响应请求后，能接管对总线的控制。
- (3) **修改主存地址**。能在地址线上给出主存地址，并自动修改主存地址。
- (4) **识别传送方向**。能识别传送方向以在控制线上给出正确的读写控制信息。
- (5) **确定传送数据个数**。
- (6) **发出DMA结束信号**。引起一次DMA中断，进行数据校验等一些后处理。



3.3 DMA数据传送过程（操作步骤）

(1) DMA控制器的预置(初始化)----软件实现

- 准备内存
- 设置参数
- 启动外设

(2) DMA数据传送----硬件实现

- DMA请求：选通-〉DMA请求-〉总线请求
- DMA响应：总线响应(CPU让出总线)-〉DMA响应
- DMA传送：DMA控制总线进行数据传送

(3) DMA结束处理----软件实现

根据计数值为“0”，发出DMA结束信号去接口控制产生DMA中断请求信号，转入中断服务程序，做一些数据校验等后处理工作。



3.3.1 DMA控制器的预置(初始化)

– 准备内存区

- * 输入：在内存设置好缓冲区

- * 输出：先在内存准备好数据

– 设置传送参数

执行I/O指令，测试外设状态，对DMA控制器设置各种参数：

- * 内存首址=〉地址寄存器

- * 字计数值=〉字计数器

- * 传送方向=〉控制寄存器

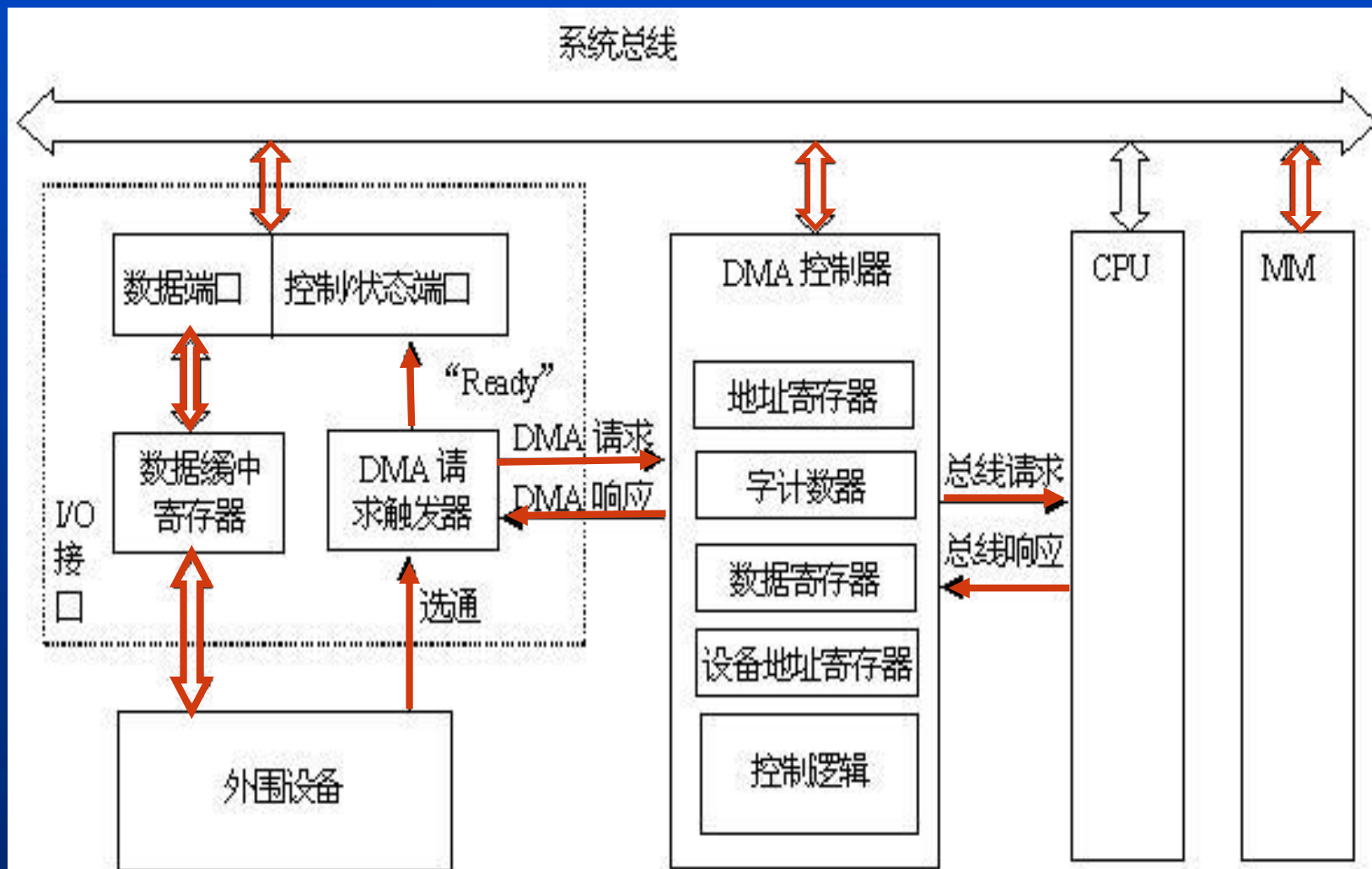
- * 设备地址=〉设备地址寄存器

– 启动外设，然后执行其他程序



3. 基本的DMA控制器

3.3.2 DMA数据的传送





3.3.2 DMA数据的传送

- (1) 当外设准备好数据，或准备好接收数据时，发“选通”信号，使数据送数据缓冲寄存器，同时DMA请求触发器置“1”。
- (2) DMA请求触发器向控制/状态端口发“Ready”信号，同时向DMA控制器发“DMA请求”信号。
- (3) DMA控制器向CPU发“总线请求”信号。
- (4) CPU完成现行机器周期后，响应DMA请求，发出“总线响应”信号。DMA控制器接受到该信号后，发出“DMA响应”信号，使DMA请求触发器复位。此时，CPU浮动它的总线，让出总线控制权，由DMA控制器控制总线。
- (5) DMA控制器给出内存地址，并在其读/写线上发出“读”或“写”命令，随后在数据总线上给出数据。
- (6) 根据读写命令，将数据总线上的数据写入存储器中，或写入数据端口，并进行主存地址增量，计数值减1，
若采用“CPU停止法”，则循环第6步，直到计数值为“0”。
若采用“周期挪用法”，则释放总线（下次数据传送时再按过程(1)到(6)进行）。

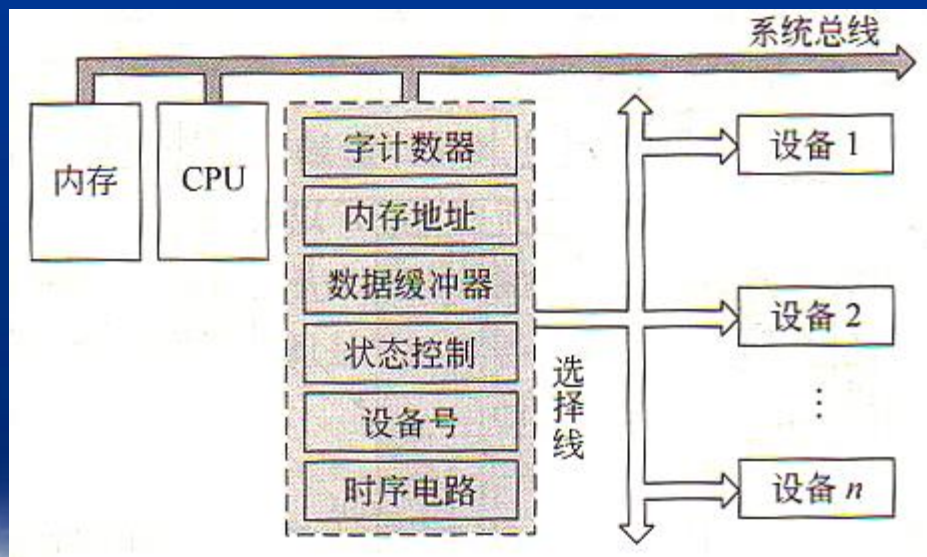


4.1 选择型DMA控制器

物理上可以连接多个设备，而逻辑上只允许连接一个设备，即在某段时间内只能为一个设备服务。

CPU响应DMA请求，初始化DMA控制器（设备号、操作命令、内存起始地址、字个数），然后专为此指定设备提供DMA数据传送服务。

适用于高速I/O设备。



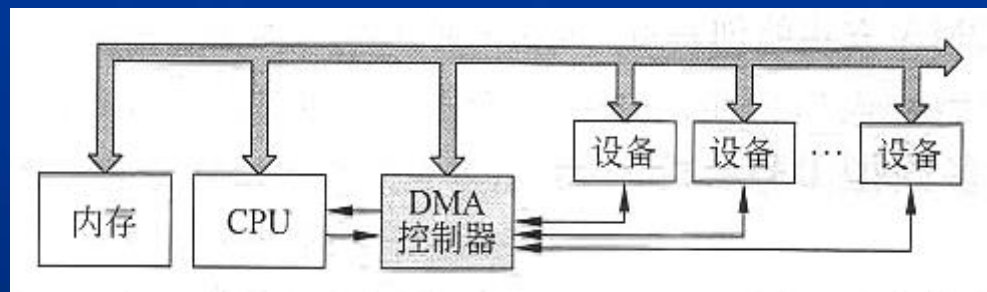


河海大学选择型和多路型DMA控制器

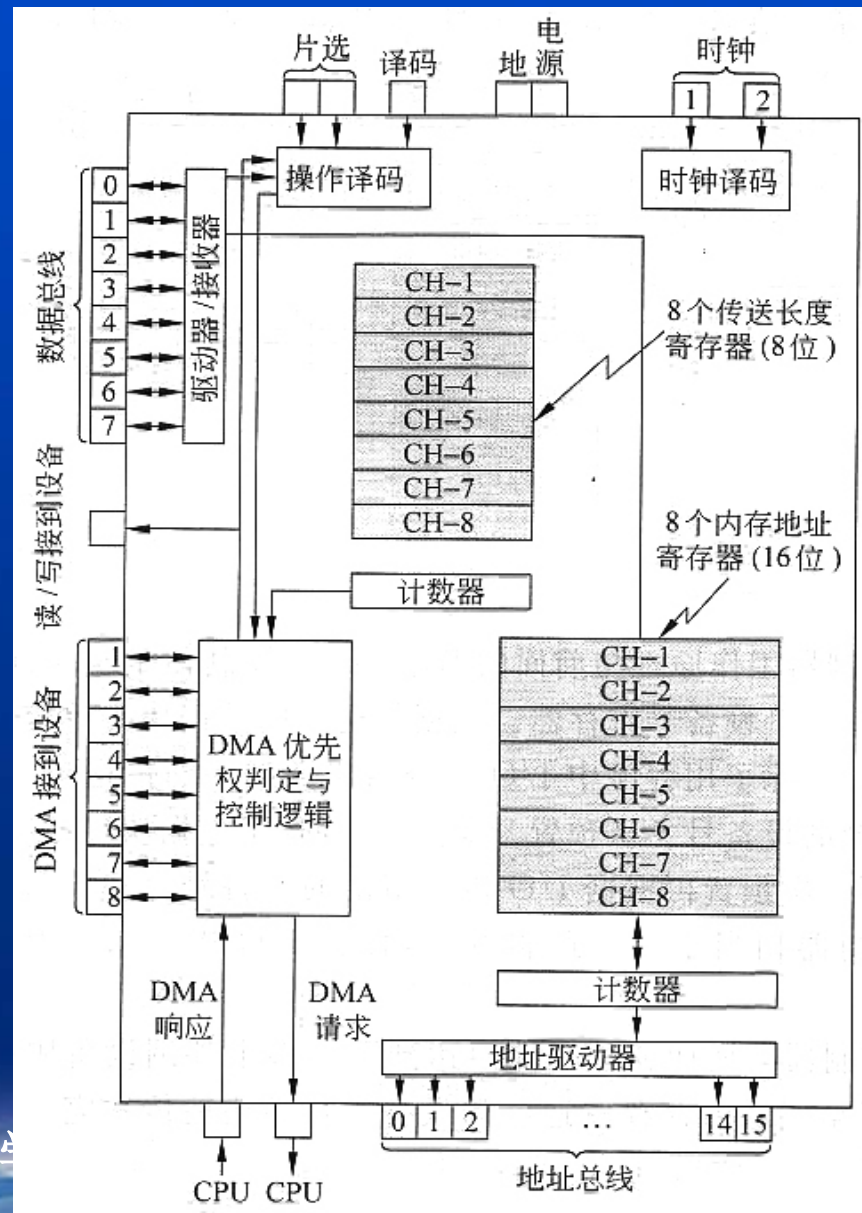
4.2 多路型DMA控制器

物理上可以连接多个设备，
逻辑上可同时为多个设备提供
DMA数据传送服务。

适用于慢速I/O设备。



(独立请求方式的多路型DMA控制器)





河海大学DMA和中断方式的区别

- (1) DMA方式下数据传送由硬件(DMA控制器)完成；中断方式下，数据传送由软件（CPU执行中断服务程序）完成。
- (2) DMA请求的是对存储器访问，也即对总线控制权的请求，没有中止现行程序的必要；而中断请求要处理器转去执行中断服务程序，因此要中止现行程序，保存断点、现场等。
- (3) 中断除了能完成外设和主机的数据交换，还能处理异常事件；而DMA方式下不能处理异常事件。
- (4) 中断响应在一个指令周期结束后；而DMA响应是在一个总线周期后。
- (5) DMA方式用于高速设备；而中断方式用于低、慢速设备。
- (6) DMA方式下，外设与CPU并行度高；而中断方式下，外设与CPU并行度低。（体现在数据传送时的并行性）



河海大学DMA vs. 中断——搬箱子

想象一下：假定车间门口有一个箱子，可放16个零件。现在车床技工想请搬运工将车间中的一批零件运到仓库，可以有几种方法？

中断方式：搬运工每装满一个箱子，通知车床技工准备下一批零件，然后搬运工把装满的箱子从车间运到仓库。

DMA方式：车床技工停下来告诉搬运工说，一次要把8000个零件放到仓库固定的地方，然后回到车床工作；搬运工开始分两组工作，一组从车间搬货到箱子中，另一组将箱子直接运到仓库指定地方，8000个零件运完后，技工停下来检查（e.g. 仓库保管员的签收单）；然后继续下一次8000个零件的搬运，

.....

上述两种方式中，哪种方式的生产效率更高？



例：计算开销

设处理器按500MHz的速度执行，硬盘控制器中有一个16B的数据缓存器，磁盘传输速率为4MB/Sec，在磁盘传输数据过程中，要求没有任何数据被错过，并假定CPU访存和DMA访存没有冲突。

(1) 若用中断驱动I/O，每次传送的开销（包括用于中断响应和处理的时间）是500个时钟周期。如果硬盘仅用5%的时间进行传送，那么处理器用在硬盘I/O操作上所花的时间百分比（主机占用率）为多少？

(2) 若用DMA方式，处理器花1000个时钟进行DMA传送的初始化设置，并且在DMA完成后的中断处理需要500个时钟。如果每次DMA传送8000B的数据块，那么当硬盘进行传送的时间占100%（即：硬盘一直进行读写，并传输数据）时，处理器用在硬盘I/O操作上的时间百分比（主机占用率）为多少？



例：计算开销

一旦磁盘被启动传送，就以4MB/s的速度进行，主机要保证没有数据丢失！

- 中断传送：

- 硬盘每次中断，可以以16字节为单位进行传送，为保证没有任何数据被错过，应达到每秒 $4\text{MB} / 16\text{B} = 250\text{k}$ 次中断的速度；
- 每秒钟用于中断的时钟周期数为 $250\text{k} \times 500 = 125 \times 10^6$ ；
- 在一次数据传输中，处理器花费在I/O上的时间的百分比为： $125 \times 10^6 / (500 \times 10^6) = 25\%$ ；
- 假定硬盘仅用其中5%的时间来传送数据，则处理器花费在I/O方面的百分比为 $25\% \times 5\% = 1.25\%$ 。

- DMA传送：

- 每次DMA传送将花费 $8000\text{B} / (4\text{MB}/\text{Sec}) \approx 2 \times 10^{-3}$ 秒；
- 一秒钟内有 $1 / (2 \times 10^{-3}) = 500$ 次DMA传送；
- 如果硬盘一直在传送数据的话，处理器必须每秒钟花 $(1000 + 500) \times 500 = 750 \times 10^3$ 个时钟周期来为硬盘I/O操作服务；
- 在硬盘I/O操作上处理器花费的时间占： $750 \times 10^3 / (500 \times 10^6) = 1.5 \times 10^{-3} = 0.15\%$ 。



CPU与I/O设备间的信息交换方式

- CPU对外围设备的管理方式
- 程序查询方式
- 程序中断方式
- DMA方式
- 通道方式



1. I/O方式的发展过程
2. 通道的功能
3. 通道的类型
4. 通道结构的发展



I/O方式的发展过程

第一阶段：CPU直接控制外设。

第二阶段：增加一个控制器或一个I/O模块，CPU使用编程I/O控制。

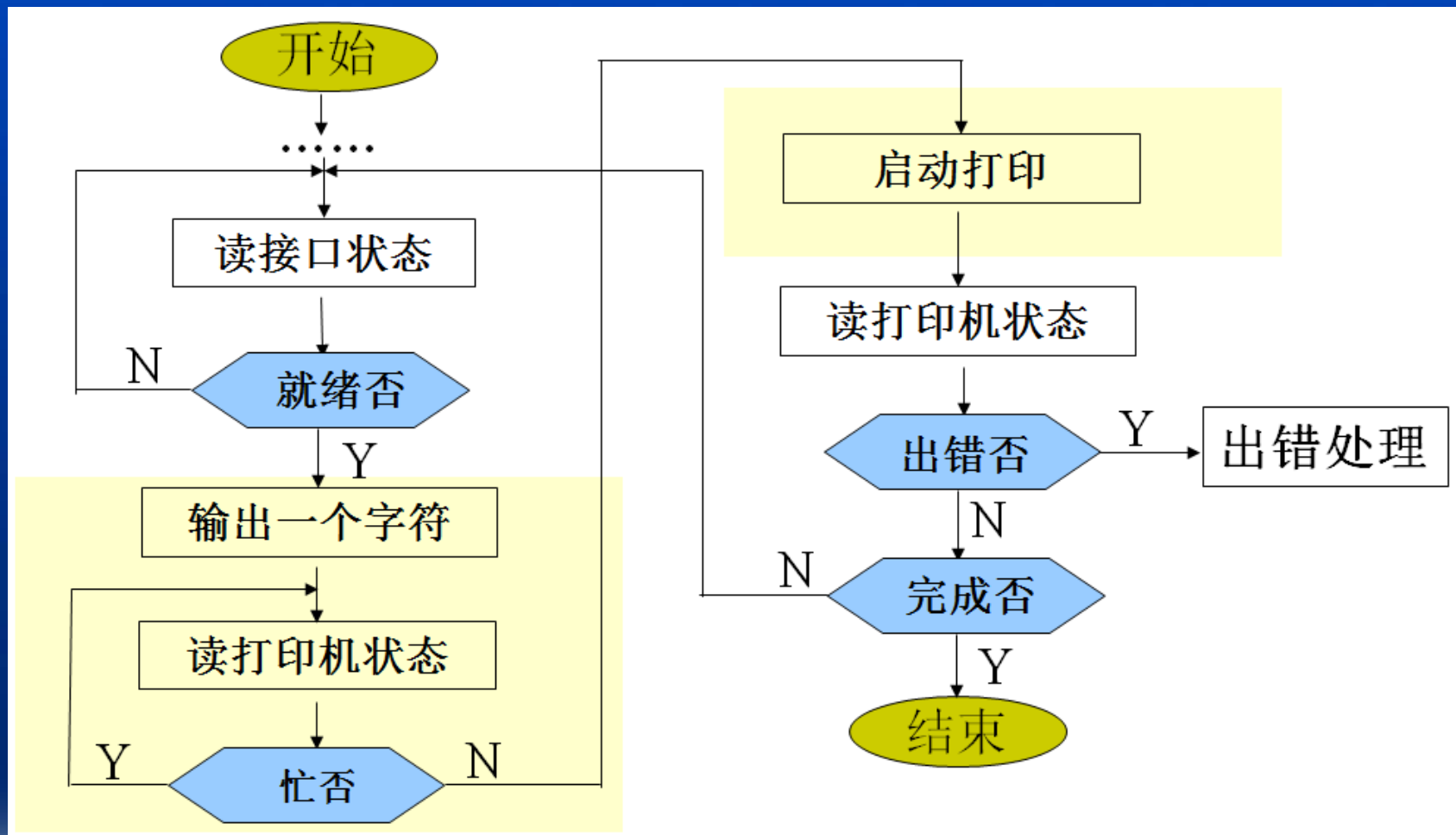
第三阶段：采用中断技术，CPU不需要花费时间等待外设执行I/O操作，实现了外设和CPU的并行。

第四阶段：I/O模块通过DMA直接传送一块数据到或从存储器传出，不需要CPU全部参与。



I/O方式的发展过程

第一阶段、第二阶段示例：控制打印输出





I/O方式的发展过程

第三阶段：中断

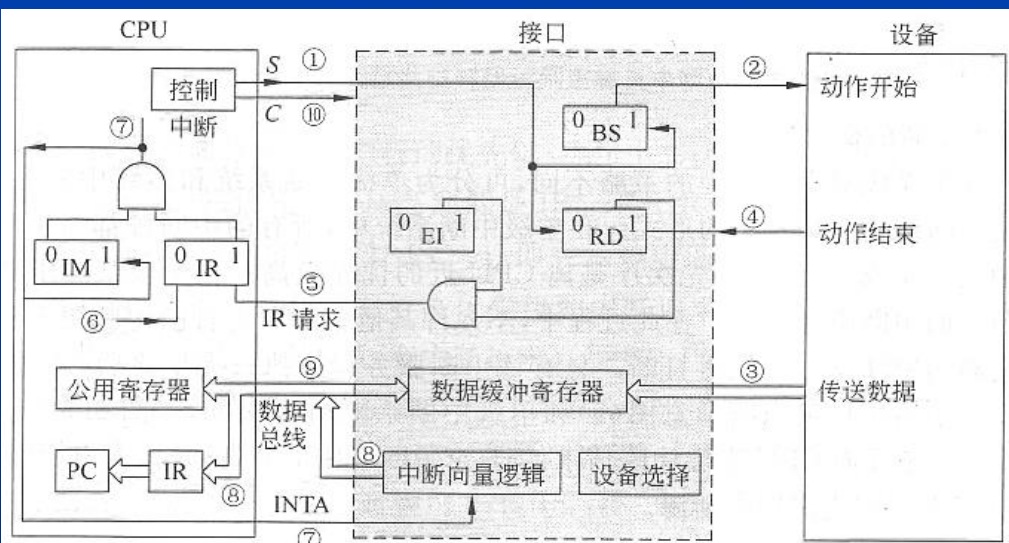


图 8.6 程序中断方式基本接口示意图

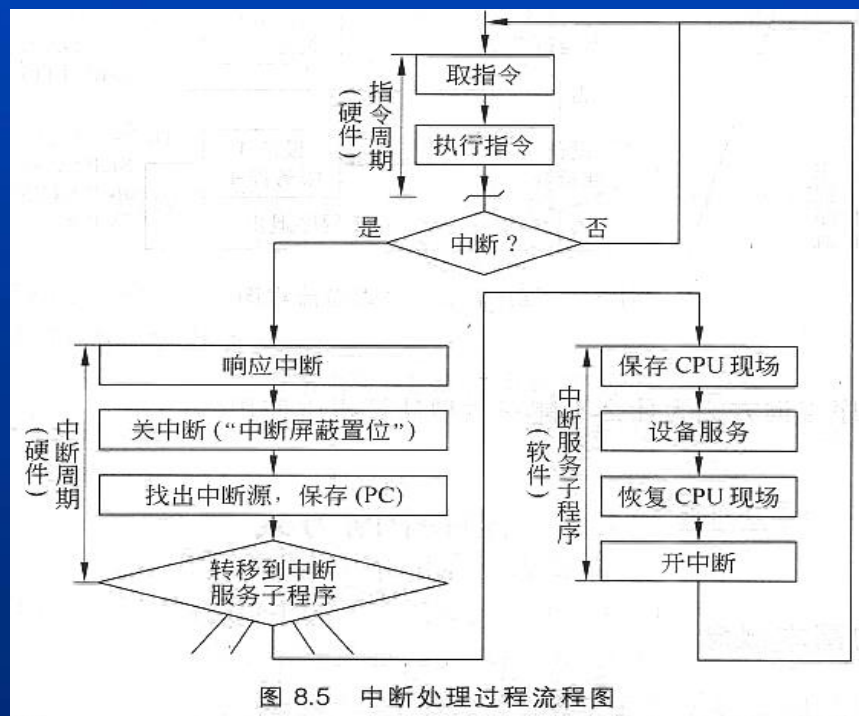
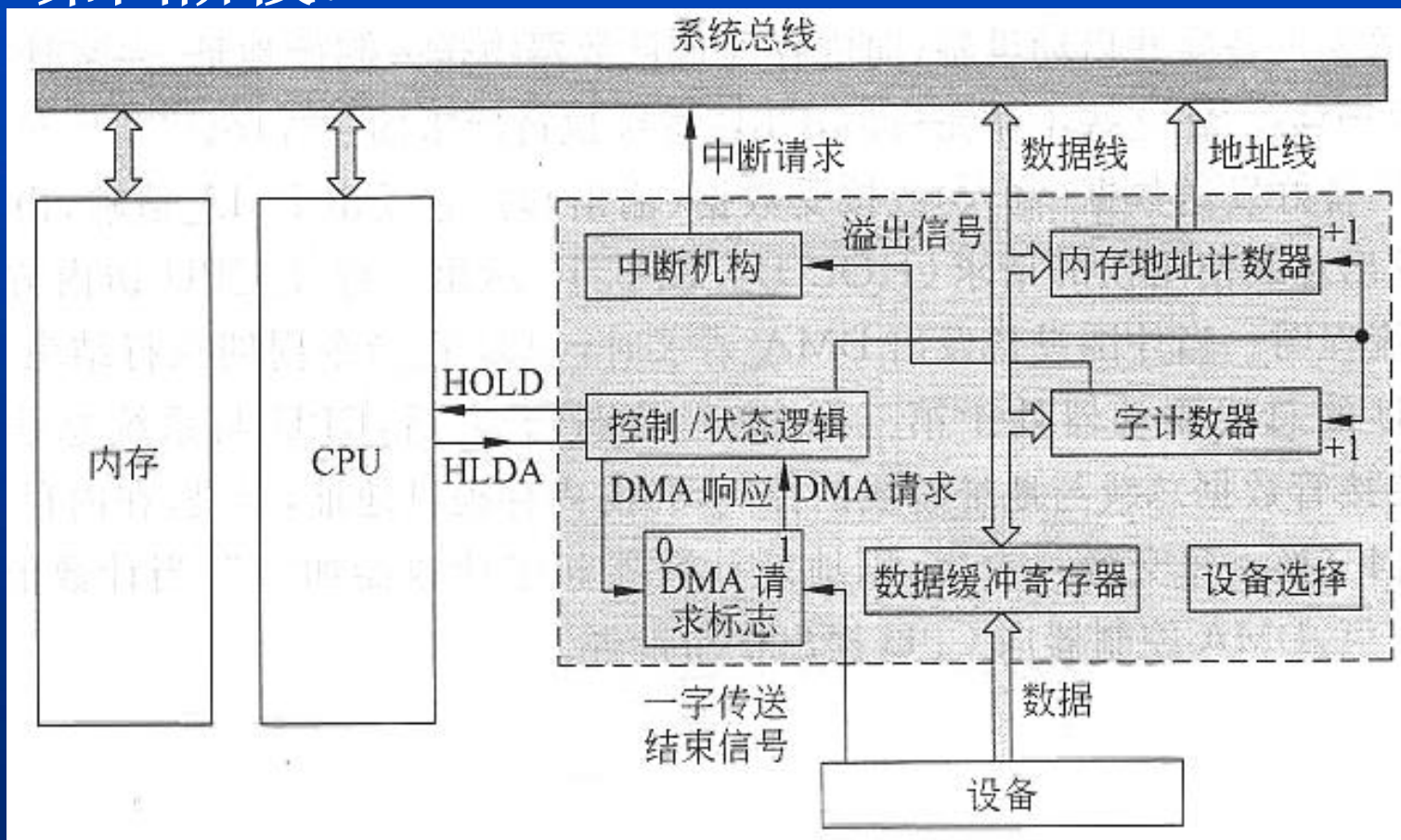


图 8.5 中断处理过程流程图



I/O方式的发展过程

第四阶段：DMA





2. 通道的功能

2.1 通道概述

DMA：减轻CPU对数据I/O的控制，提高CPU效率。

通道：一种特殊功能的处理器，使CPU的传输控制功能进一步下放。

- 有自己的指令和程序：通道指令、通道程序
- 与CPU分时使用总线和存储器

通道 vs. DMA：

- 通道在DMA控制器基础上发展而来
- 传输控制功能更强：DMA控制器只是一个接口部件，没有自己的指令和程序



2. 通道的功能

2.2 通道结构的计算机系统

系统总线：承担CPU、存储器、通道之间的数据传输交换。

I/O总线：也称通道总线，承担外围设备与通道之间的数据传输。

通道：两种总线采用不同的时序，通道需提供缓冲、转换等功能。

四级控制结构：

- (1) CPU与存储器
- (2) 通道
- (3) I/O模块（设备控制器）
- (4) I/O设备

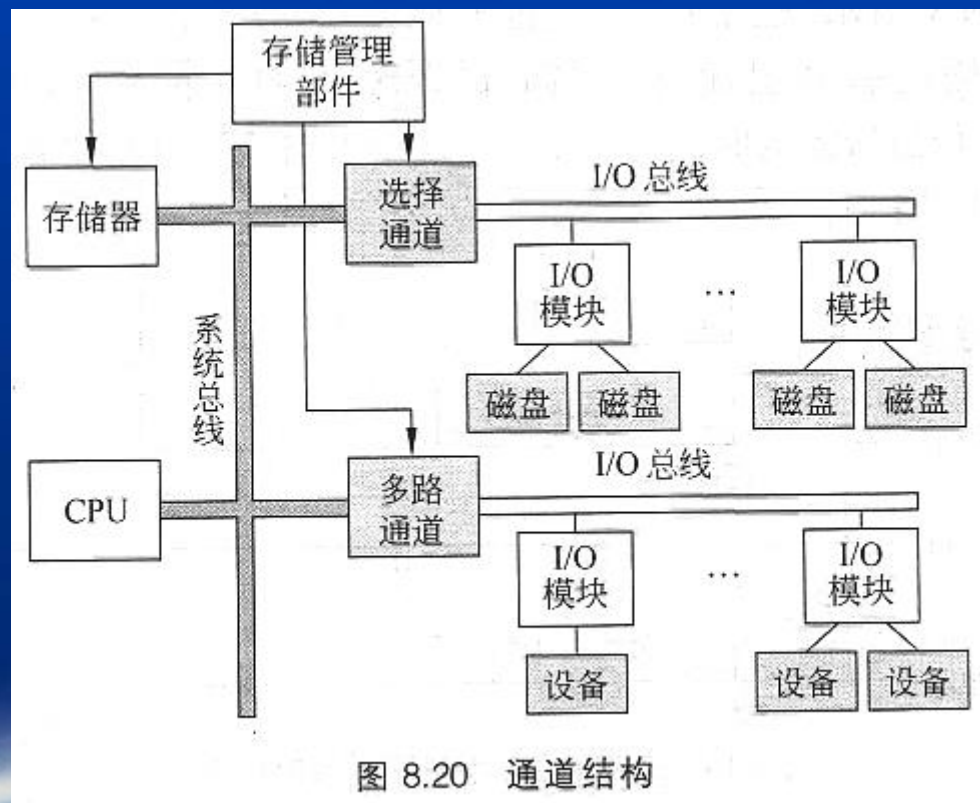


图 8.20 通道结构

2.3 通道的功能

- (1) CPU执行I/O指令，通道接收来自CPU输出的地址信息、控制信息，按指令要求与指定的外围设备进行通信。
- (2) 从内存选取属于指定设备的通道程序，逐条执行通道指令，向设备控制器发送各种命令。
- (3) 组织外围设备与CPU、内存之间进行数据传送，并根据需要提供数据缓存的空间，以及提供数据存入内存的地址和传送的数据量。
- (4) 从外围设备得到设备的状态信息，形成并保存通道本身的状态信息，根据要求将这些状态信息传送给CPU。
- (5) 将外围设备的中断请求和通道本身的中断请求，按次序及时报告CPU。

3. 通道的类型

3.1 选择通道

物理上可以连接多个设备，而逻辑上只允许连接一个设备，即在某段时间内只能为一个设备服务；

适用于高速I/O设备。

3.2 多路通道

物理上可以连接多个设备，逻辑上可同时为多个设备提供数据传送服务。

在同一时间段内能处理多个I/O设备的数据传输；
适用于低速I/O设备。



4.1 输入输出处理器(IOP)

IOP是通道结构的I/O处理器，可以和CPU并行工作，提供高速的DMA处理能力，实现数据的高速传送。但是它不是独立于CPU工作的，而是主机的一个部件，与CPU协同工作。

如INTEL 8086: CPU

INTEL 8087: 协处理器

INTEL 8089: IOP, I/O管理。



4.2 外围处理机(PPU)

外围处理机**PPU**基本上是独立于主机工作，它有自己的指令系统，只是侧重于I/O管理。结构上接近一台计算机、或者就是一台通用计算机，一般称为前置机。

在一些大型计算机系统中，可以设置多台PPU，分别承担I/O管理、通信、维护诊断等任务，实际上已演变为多处理机/多计算机系统。



1、基本概念

DMA方式、通道、外围处理机

2、程序查询方式处理过程

3、程序中断方式处理过程

4、基本DMA控制器的逻辑结构



第八章习题



设某机有5级中断： L_0 、 L_1 、 L_2 、 L_3 和 L_4 ，其中断响应优先次序为 $L_0 > L_1 > L_2 > L_3 > L_4$ ，现要求将中断处理次序改为 $L_1 > L_3 > L_0 > L_4 > L_2$ ，试问：

(1) 各级中断处理程序的中断屏蔽位如何设置？

(2) 若这五级中断同时都发出中断请求，按更改后的次序画出进入各级中断处理程序的过程示意图。

中断处理程序	中 断 处 理 级 屏 蔽 位				
	L0 级	L1 级	L2 级	L3 级	L4 级
L0 中断处理程序	1	0	1	0	1
L1 中断处理程序	1	1	1	1	1
L2 中断处理程序	0	0	1	0	0
L3 中断处理程序	1	0	1	1	1
L4 中断处理程序	0	0	1	0	1



P266第12题

下列陈述中正确的是（ ）

- A. 在DMA周期内，CPU不能执行程序
- B. 中断发生时，CPU首先执行入栈指令将程序计数器的内容保护起来
- C. DMA传送方式中，DMA控制器每传送一个数据就窃取一个指令周期
- D. 输入输出操作的最终目的是要实现CPU与外设之间的数据传输

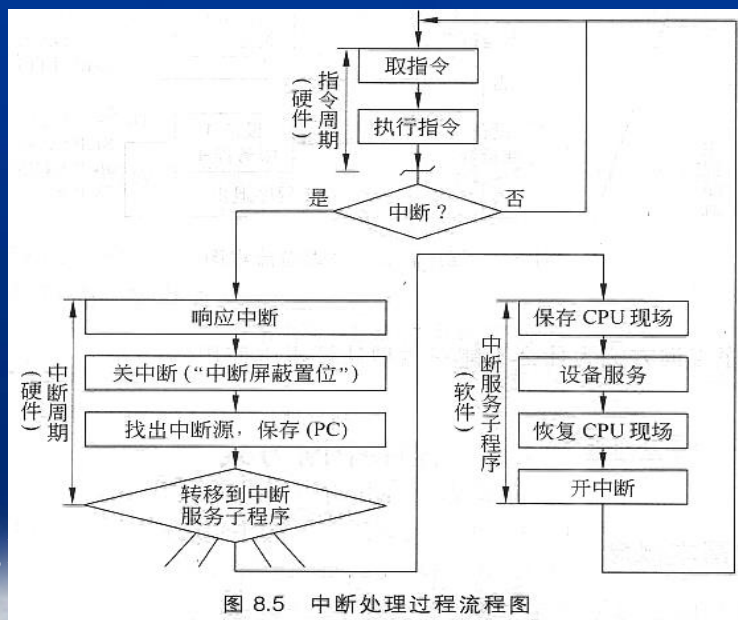


图 8.5 中断处理过程流程图



用多路DMA控制器控制光盘、软盘、打印机三个设备同时工作。光盘以 $20\mu\text{s}$ 的间隔向控制器发DMA请求，软盘以 $90\mu\text{s}$ 的间隔向控制器发DMA请求，打印机以 $180\mu\text{s}$ 的间隔向控制器发DMA请求。请画出多路DMA控制器的工作时空图。

假设DMA控制器完成一次DMA传送需要 $5\mu\text{s}$ ，可画图下图：

