



廣西大學  
Guangxi University

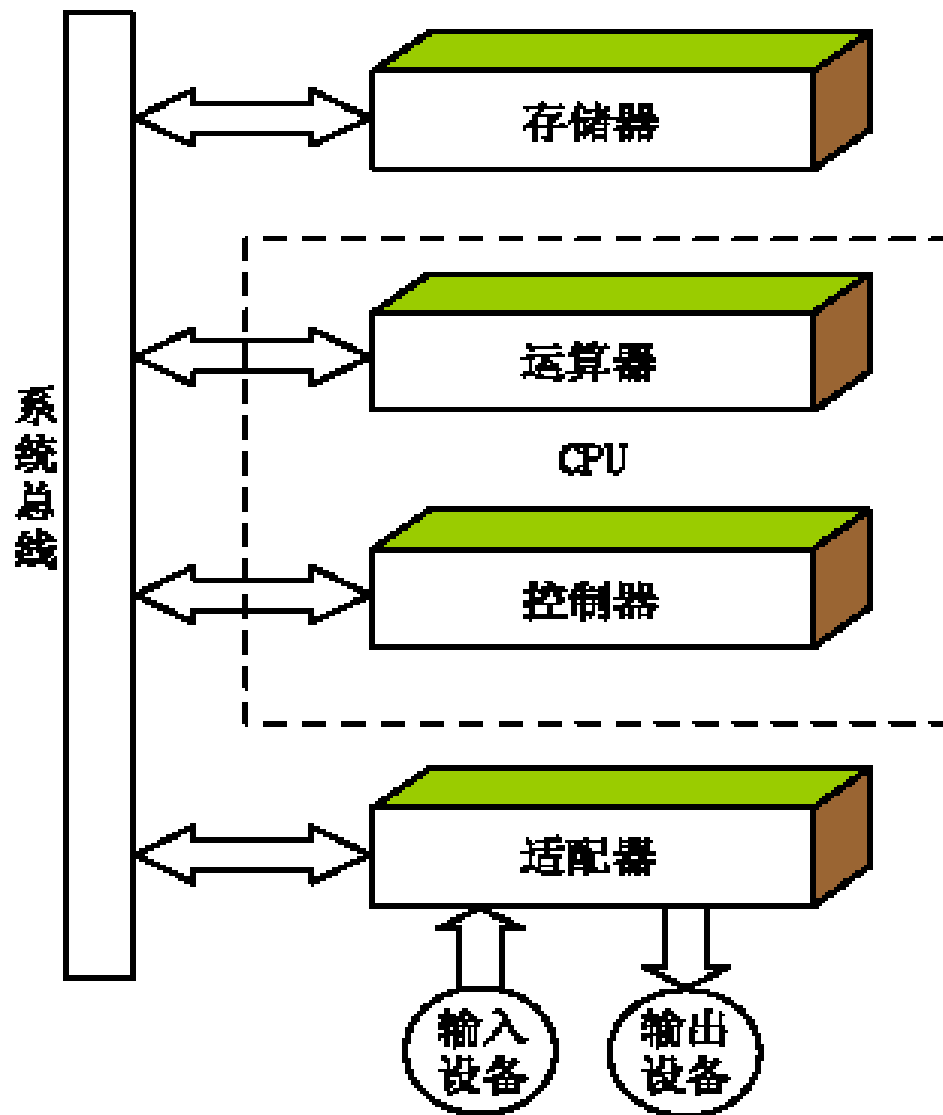
# 第8章 输入输出系统

广西大学计算机与电子信息学院

2022.5



## 回顾：第一章：计算机的主要组成结构





# 内容简介

CPU与外设间的信息交换方式

程序**查询**方式

程序**中断**方式

**DMA**方式、通道方式

I/O标准接口



## 8.1 CPU与外围设备之间的信息交换方式

## 8.2 程序查询方式

### 8.3 程序中中断方式

## 8.4 DMA方式

## 8.5 通道方式

## 8.6 通用I/O标准接口



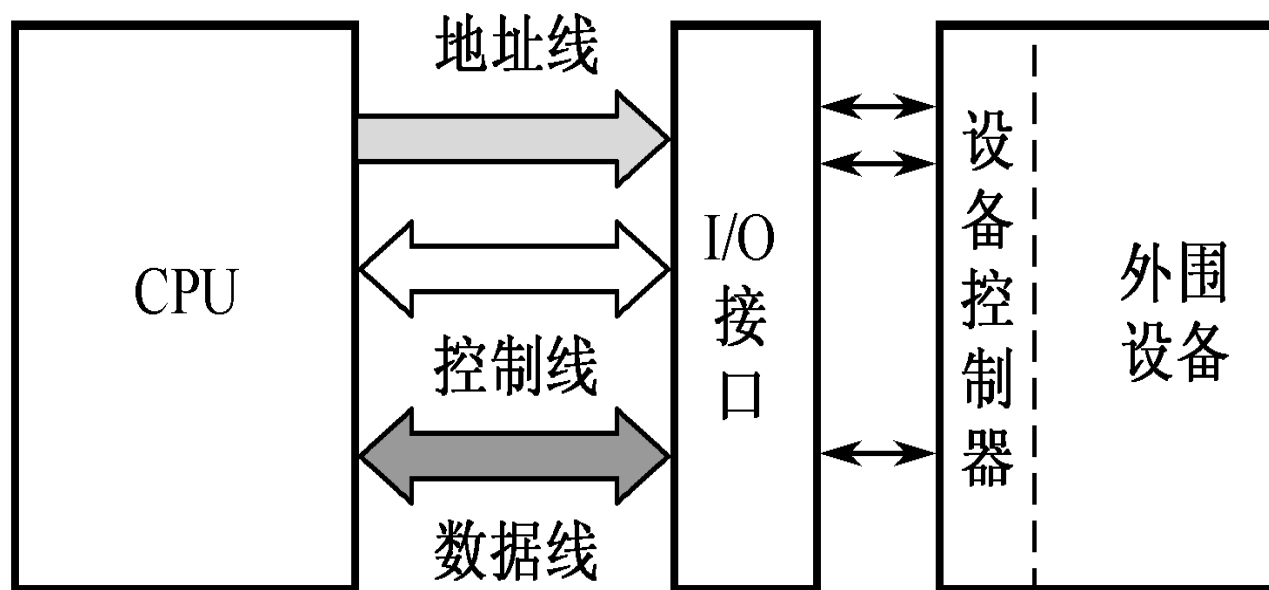
# 8.1 CPU与外围设备的信息交换方式

- 一. 输入输出接口与端口
- 二. 输入输出操作的一般过程
- 三. I/O接口与外设间的数据传送方式
- 四. CPU与I/O接口的数据传送方式



# 一、输入输出接口与端口

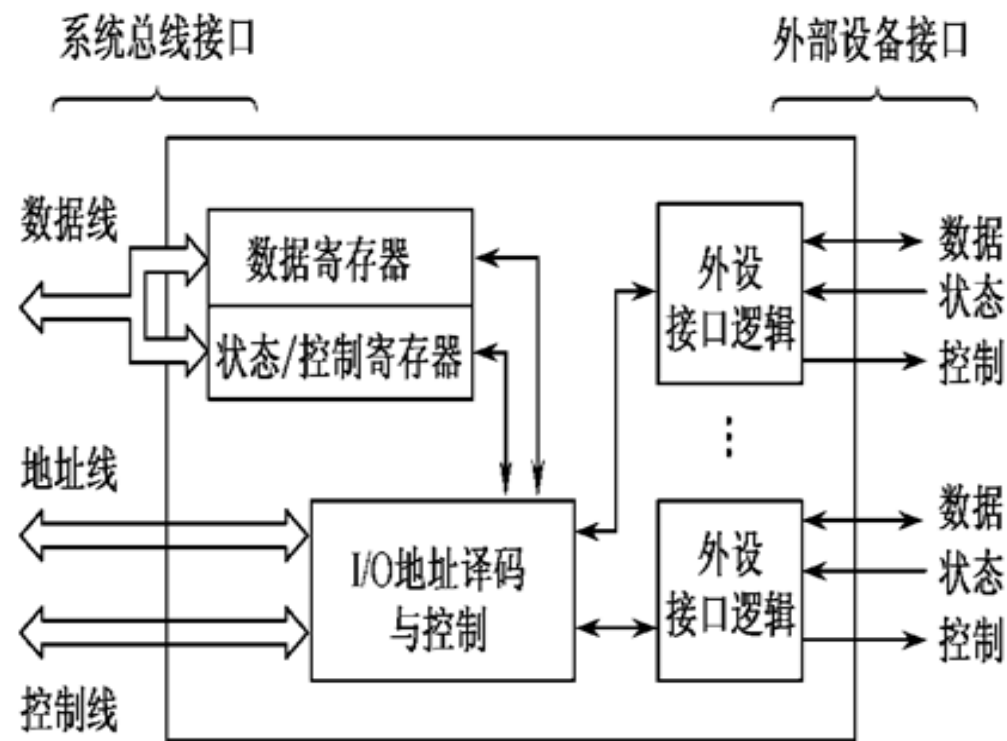
- 接口也叫适配器，是CPU和外设之间通过总线进行连接的标准化逻辑部件。
- 起转换器的作用，以便实现彼此之间的信息传送





# 一、输入输出接口与端口

- 为了与CPU交互信息的方便，接口内部一般要设置一些可以直接被CPU访问的寄存器（端口）。
- 命令口
- 状态口
- 数据口





# 一、输入输出接口与端口

- 1、设备编址

- **统一**编址：I/O设备接口中的控制寄存器、数据寄存器、状态寄存器等和内存单元一起编排地址
- **独立**编址：内存和I/O设备地址分开。要使用专门的I/O指令访问I/O设备。





## 二、输入输出操作的一般过程

- 外设同CPU交换数据的过程（输入）：
  - CPU把地址值放在地址总线上
  - CPU等待输入设备的数据有效
  - CPU从数据总线上读入数据，放到寄存器上。
- 外设同CPU交换数据的过程（输出）：
  - CPU把地址值放在地址总线上
  - CPU把数据放在数据总线上
  - 输出设备认为数据有效时把数据取走



## 二、输入输出操作的一般过程

- 什么时候数据才成为有效？
- 外设种类繁多，存在以下几种情况：
  - 不同类型的外设数据传输速率差别很大
  - 同一种设备在不同时刻传输速率也可能不同
- 高速的CPU与速度参差不齐的外设怎样联络？
- 实现在时间上同步？



## 二、输入输出操作的一般过程

- I/O对系统性能的影响
- 输入输出系统（I / O系统）作为计算机系统中的一个重要组成部分，其性能的好坏对CPU的性能有很大的影响。
- 【例8.1】



## 三、I/O接口与外设间的数据传送方式

- 高速的CPU与速度参差不齐的外设怎样在时间上同步呢（定时问题）？
- （1）速度极慢或简单的外设：**无条件**传送方式
  - 如机械开关、显示二极管
  - CPU只需要接受或者发送数据即可
  - 无需联络信号
  - 接口只需要实现数据缓冲和寻址功能



## 三、I/O接口与外设间的数据传送方式

- 高速的CPU与速度参差不齐的外设怎样在时间上同步呢（定时问题）？
- （2）慢速或者中速的设备：应答方式（异步传送方式）
  - 外设与主机速度不在一个数量级，或者设备本身的操作时间不规则
  - 如键盘、打印机等
  - 需要联络信号线
  - 异步定时的方式，应答式数据交换



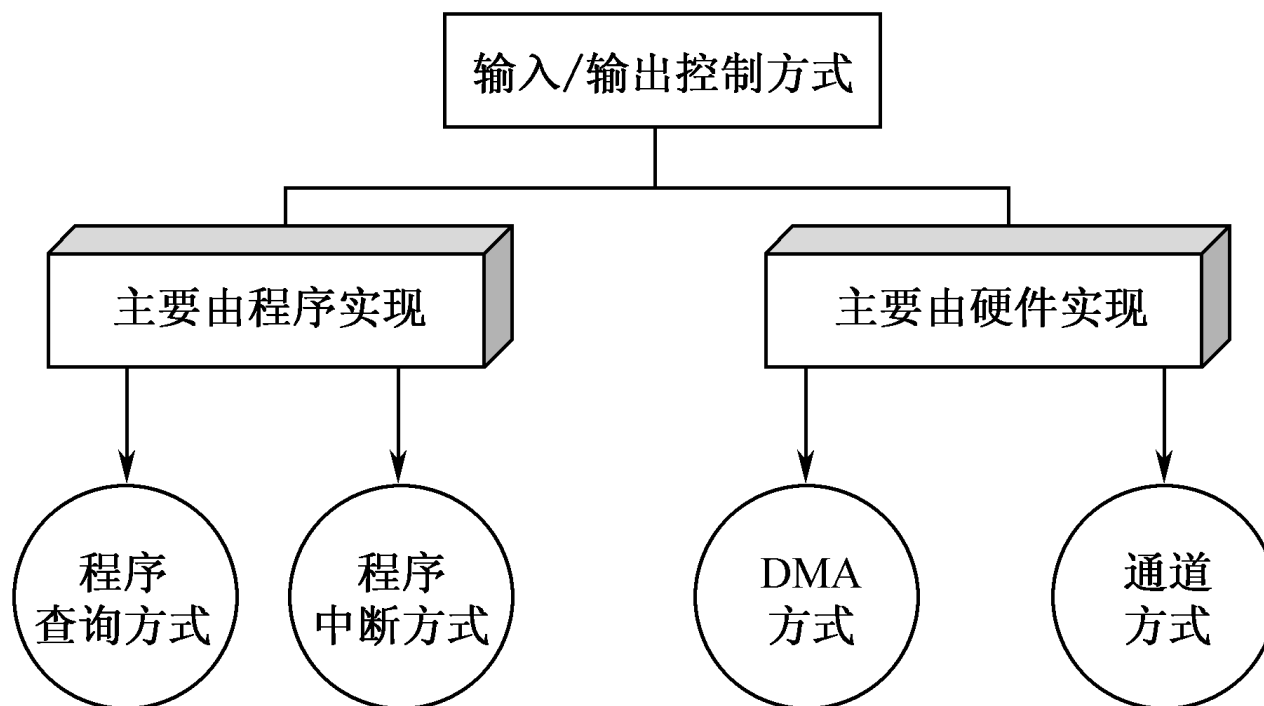
## 三、I/O接口与外设间的数据传送方式

- 高速的CPU与速度参差不齐的外设怎样在时间上同步呢（定时问题）？
  - （3）高速外设：**同步**传送方式
    - 按**规则间隔**工作的外部设备，接口可以某一确定的时钟速率和外设交换信息
    - 同步定时方式，时钟脉冲控制



## 四、CPU与I/O接口的数据传送方式

- 无条件传送方式
- 程序查询方式
- 程序中断方式
- DMA方式
- 通道方式





## 四、CPU与I/O接口的数据传送方式

- (1) 无条件传送方式
  - 外设始终处于准备就绪状态，CPU直接执行I/O指令进行数据传输
  - 接口与外设之间用无条件传送方式时，CPU与接口之间才能用





## 四、CPU与I/O接口的数据传送方式

- (2) 程序查询方式
  - 数据传送前，CPU需要通过接口对目标设备的状态进行查询
  - 若外设已准备好,则可以进行数据传送
  - 若外设没有准备好数据，则CPU不断地查询并等待，直到外设准备好
  - 查询循环会消耗大量的CPU时间



## 四、CPU与I/O接口的数据传送方式

- (3) 程序中断方式
  - 外设准备就绪后主动通知CPU，准备传送数据
  - 中断发生时，CPU暂停现行程序，转向中断处理程序，进行数据传送
  - 中断处理完毕，CPU再返回原来的地方执行程序
  - 节省CPU时间



## 四、CPU与I/O接口的数据传送方式

- (4) 直接内存访问 (DMA) 方式
  - 由**DMA控制器**接管对总线的控制，数据交换不经过CPU，直接在内存和外设之间进行
  - 高速传送数据，传送速率仅受内存访问时间的限制
  - 高速、**成批**传送数据
  - 适用于内存和高速外设之间大批量数据交换



## 四、CPU与I/O接口的数据传送方式

- (5) 通道/IOP方式
  - 对外设进行统一管理
  - 对外设与内存之间的数据 传送控制
  - 进一步提高CPU的效率
  - 耗费更多的硬件为个代价



## 程序查询、程序中断和DMA三种方式的性能比较

- 【1】数据传送方式角度：程序查询方式和程序中断方式主要依靠**软件**来实现，而**DMA**方式依靠**硬件**来实现。
- 【2】传送数据的基本单位角度：程序查询方式和中断方式是以**字**为单位进行传输的，而**DMA**是以块进行传输的。
- 【3】并行性角度：程序**查询**方式是CPU和I/O**串行**，而程序中断和DMA是CPU和I/O在**数据准备**阶段是**并行**的，传输的阶段和主程序是串行的。



# 程序查询、程序中断和DMA三种方式的性能比较

- 【4】 主动性角度：程序查询方式是CPU主动发出，而程序中断和DMA是设备主动发出。
- 【5】 传输速度角度：程序查询方式和程序中断方式都有CPU的参与，所以传输速度慢，而DMA方式没有CPU进行参与，是主存直接和DMA设备进行数据交换，传输速度快。
- 【6】 应用对象角度：程序查询方式应用于低速设备，程序中断方式应用于中速设备，DMA方式应用于高速设备或者批量传输。



## 8.2 程序查询方式

- 数据在CPU和外设之间的传送由程序控制
- 当需要进行输入/输出时，CPU暂停主程序，转去执行设备输入/输出的服务程序，根据服务程序中的I/O指令进行数据传送
- 只需要很少的硬件

## 8.2 程序查询方式

- 1、输入输出指令

- I/O指令的功能

- (1) 通过触发器控制设备动作

- (2) 测试设备的状态，以便决定下一步操作(如SKP)

- (3) 传送数据 (如:DOAS 2,13 ) (如:DICC 3,12)

- 如:

0 1	R0~R7	OP	控制	DMs
-----	-------	----	----	-----

0      1      2                  4   5                  7   8                  9   10                  15

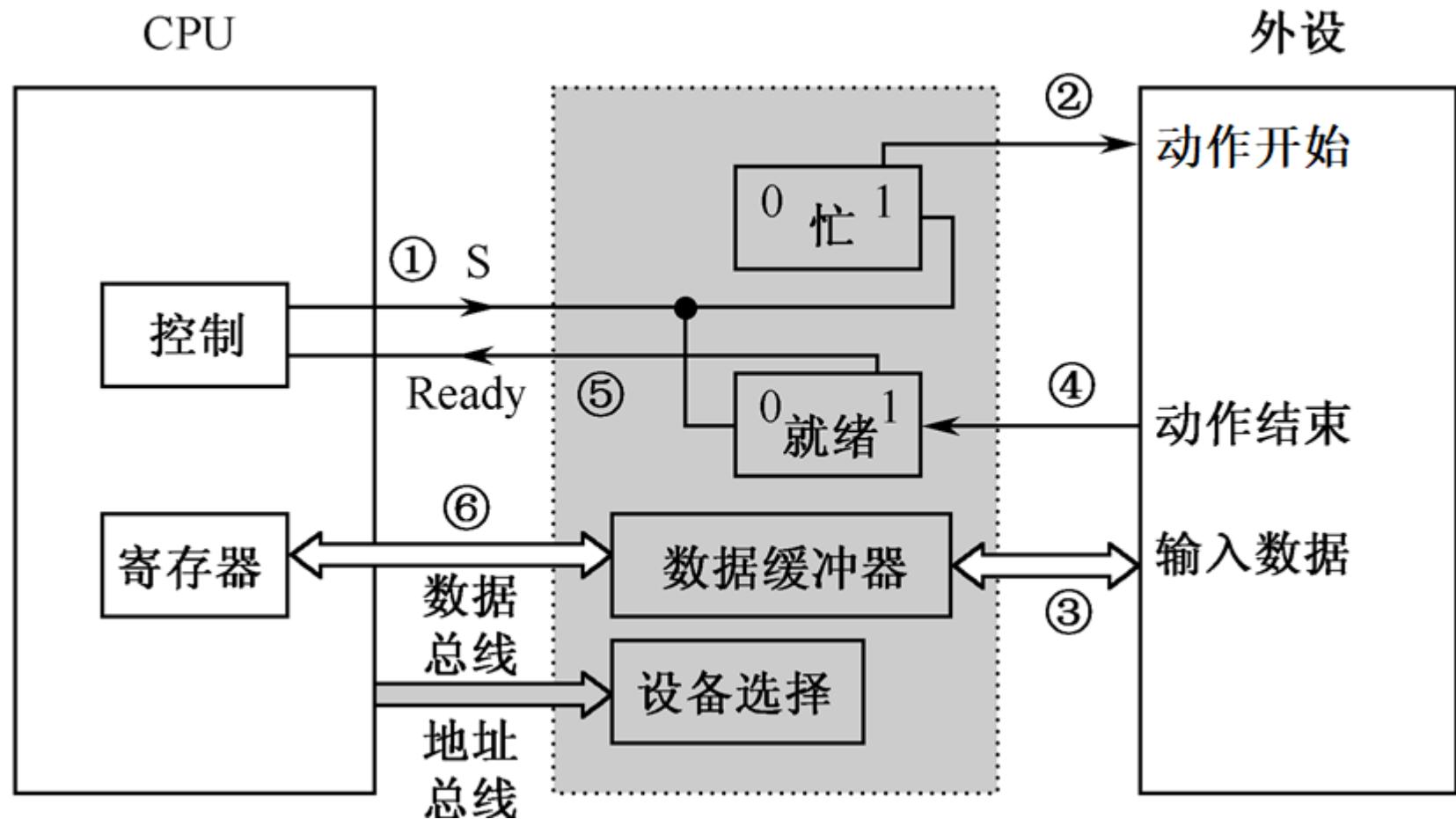


## 8.2 程序查询方式

- 2、程序查询接口：总线和外设之间的逻辑部件，作为转换器，保证外部设备用系统所要求的形式改善或接收。
- 主要组成
  - 设备选择电路
  - 数据缓冲寄存器
  - 设备状态寄存器



## 8.2 程序查询方式



- (1) 外设以I/O设备内部数据缓冲器数据缓冲器接收数据；
  - (2) CPU向I/O设备内部数据缓冲器数据缓冲器发送数据；
  - (3) CPU向I/O设备内部数据缓冲器数据缓冲器发送数据；
  - (4) CPU向I/O设备内部数据缓冲器数据缓冲器发送数据；
  - (5) CPU向I/O设备内部数据缓冲器数据缓冲器发送数据；
  - (6) CPU向I/O设备内部数据缓冲器数据缓冲器发送数据；
- 或者将数据从CPU输出至接口的数据缓冲器，发出准备就绪信号“Ready”为止；

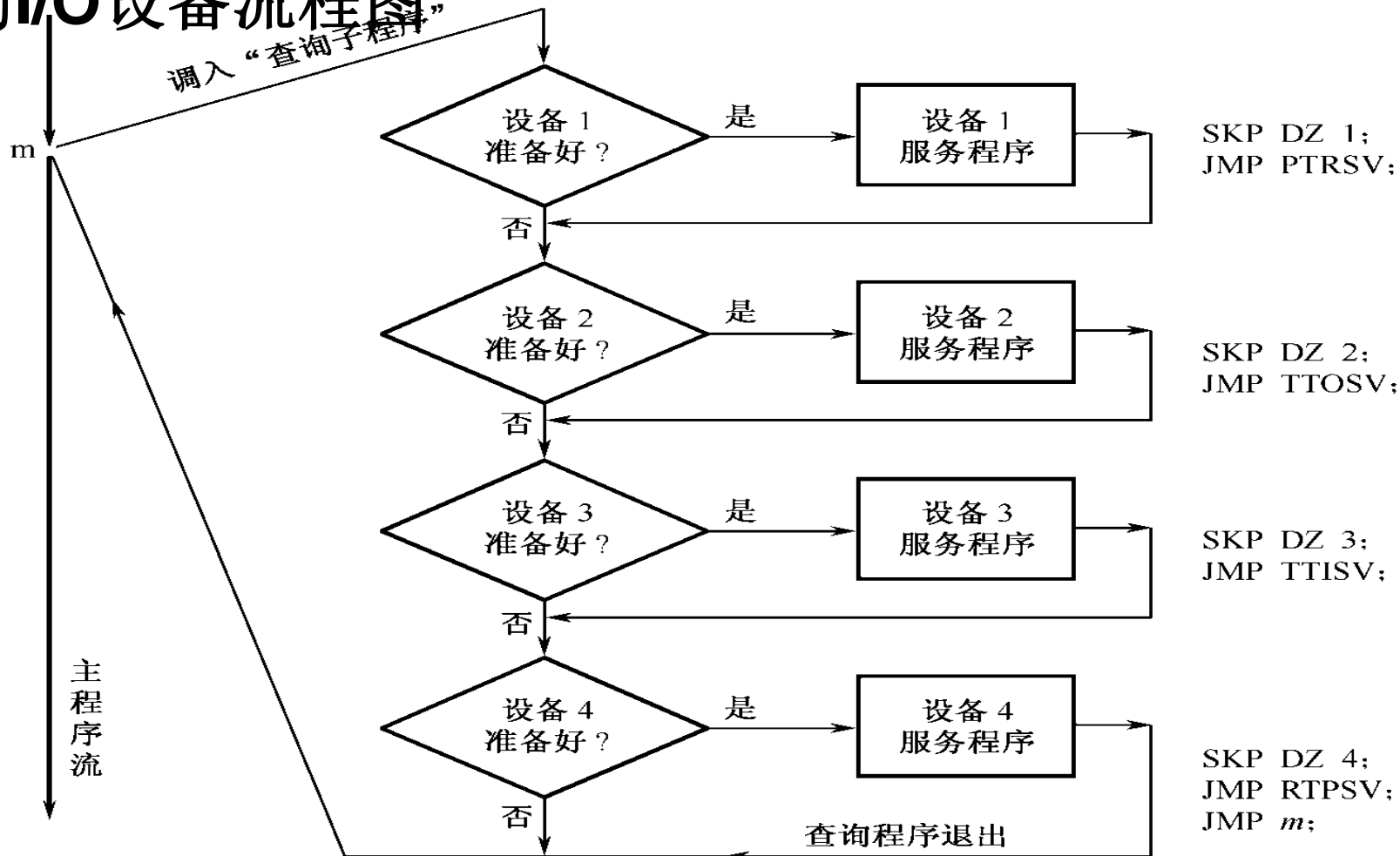
## 8.2 程序查询方式

- (1) 先向I/O设备发出命令字，请求进行数据传送
- (2) 从I/O接口读入状态字；
- (3) 检查状态字中的标志，看看数据交换是否可以进行；
- (4) 假如这个设备没有准备就绪，则第(2)、第(3)步重复进行，一直到这个设备准备好交换数据，发出准备就绪信号“Ready”；
- (5) CPU从I/O接口的数据缓冲寄存器输入数据，或者将数据从CPU输出至接口的数据缓冲寄存器。与此同时，CPU将接口中的状态标志复位。
- (6) 数据传送

## 8.2 程序查询方式

改进：周期性调用各外设询问子程序，依次测试各外设的状态是否“就绪”

程序查询I/O设备流程图





## 8.3 程序中斷方式

- 一. 中斷的基本概念
- 二. 程序中斷方式的基本I/O接口
- 三. 單級中斷
- 四. 多級中斷
- 五. 中斷控制器
- 六. Pentium中斷機制



# 一、中断的基本概念

- 中断是一种程序**随机**切换的方式，也称为异常。
- 典型应用
  - CPU与外界进行信息交换
  - 故障处理
  - 实时处理
  - 程序调度
  - 软中断，进行软件的升级维护



# 一、中断的基本概念

- 程序中断是外设准备好数据后，由设备主动发出中断信号，请求CPU暂时中断目前正在执行的程序而进行数据交换。
- CPU响应中断时，暂停正在执行的主程序，自动转移到设备的中断服务程序。
- 中断服务程序结束后，CPU又回到原来的主程序。



# 一、中断的基本概念

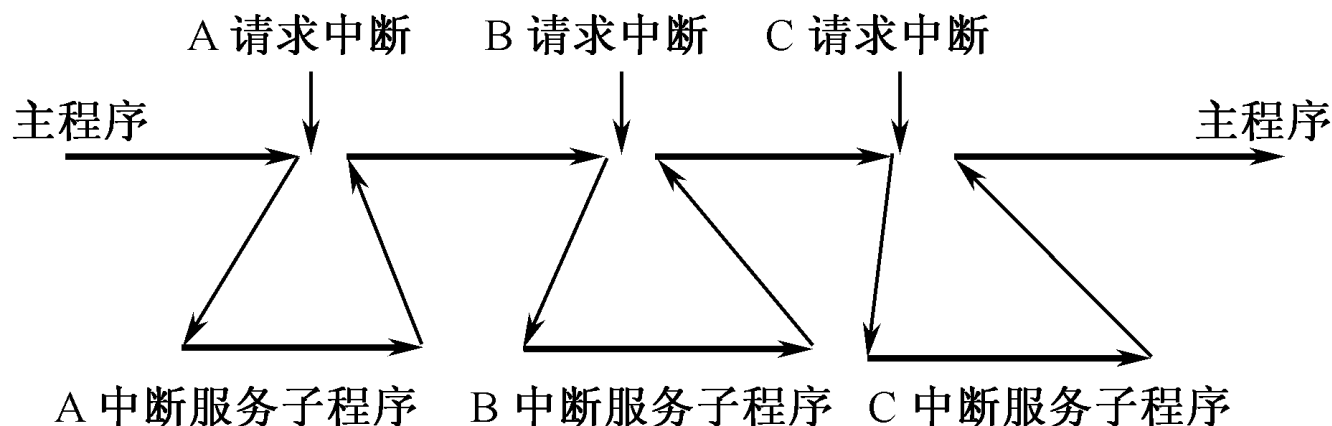
- 外设在准备数据时，CPU照常执行自己的主程序
- 程序中断方式CPU和I/O在数据准备阶段是并行的，传输阶段和主程序是串行的。
- 效果比程序查询方式提高了





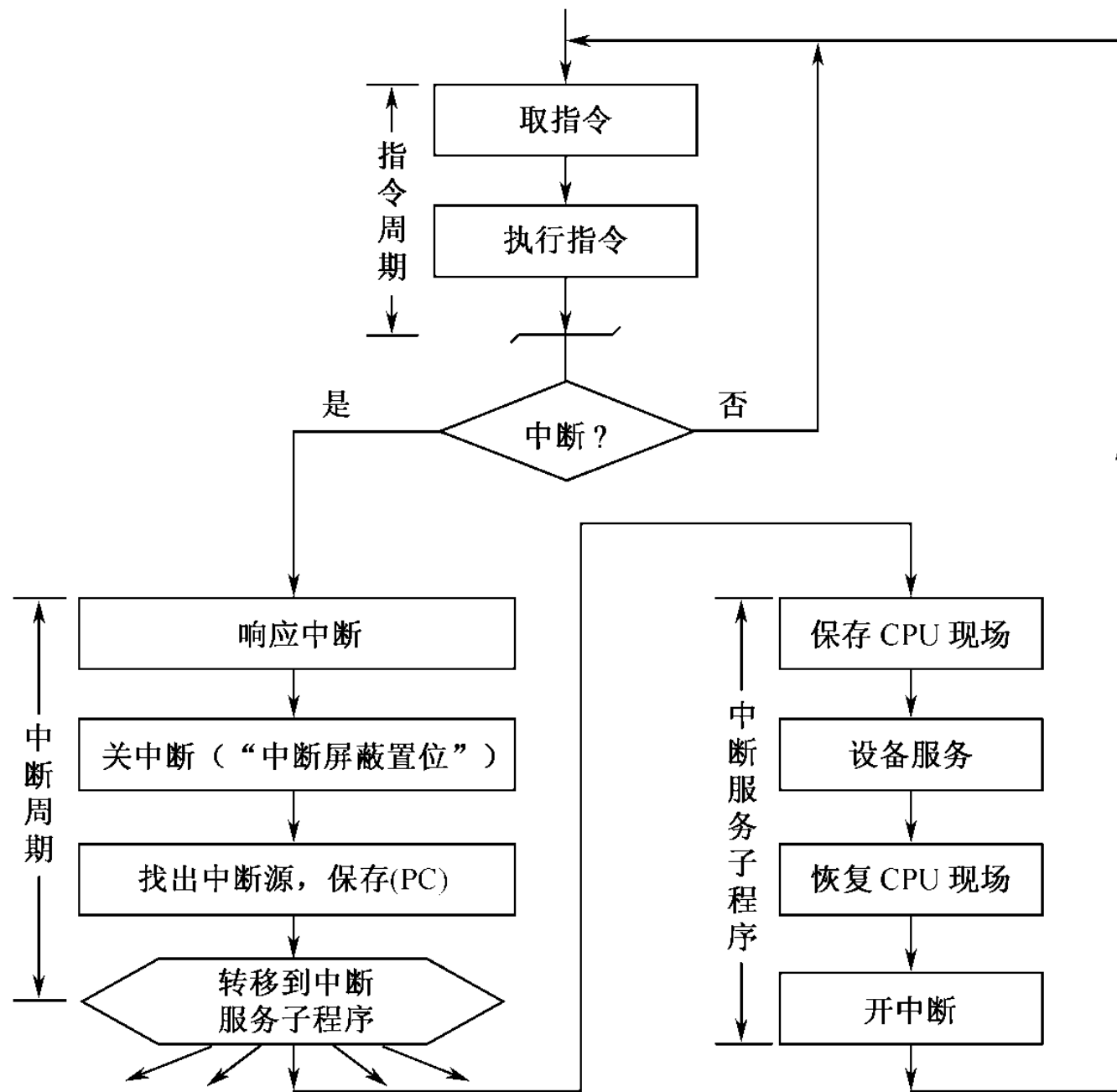
# 一、中断的基本概念

- 中断（Interrupt）是指CPU**暂停**现行程序，去处理随机发生的**紧急事件**，处理完后自动返回原程序的功能和技术。中断系统是计算机实现中断功能的软硬件总称。一般在CPU中设置**中断机构**，在外设接口中设置**中断控制器**，在软件上设置相应的**中断服务程序**。





## 中断处理过程





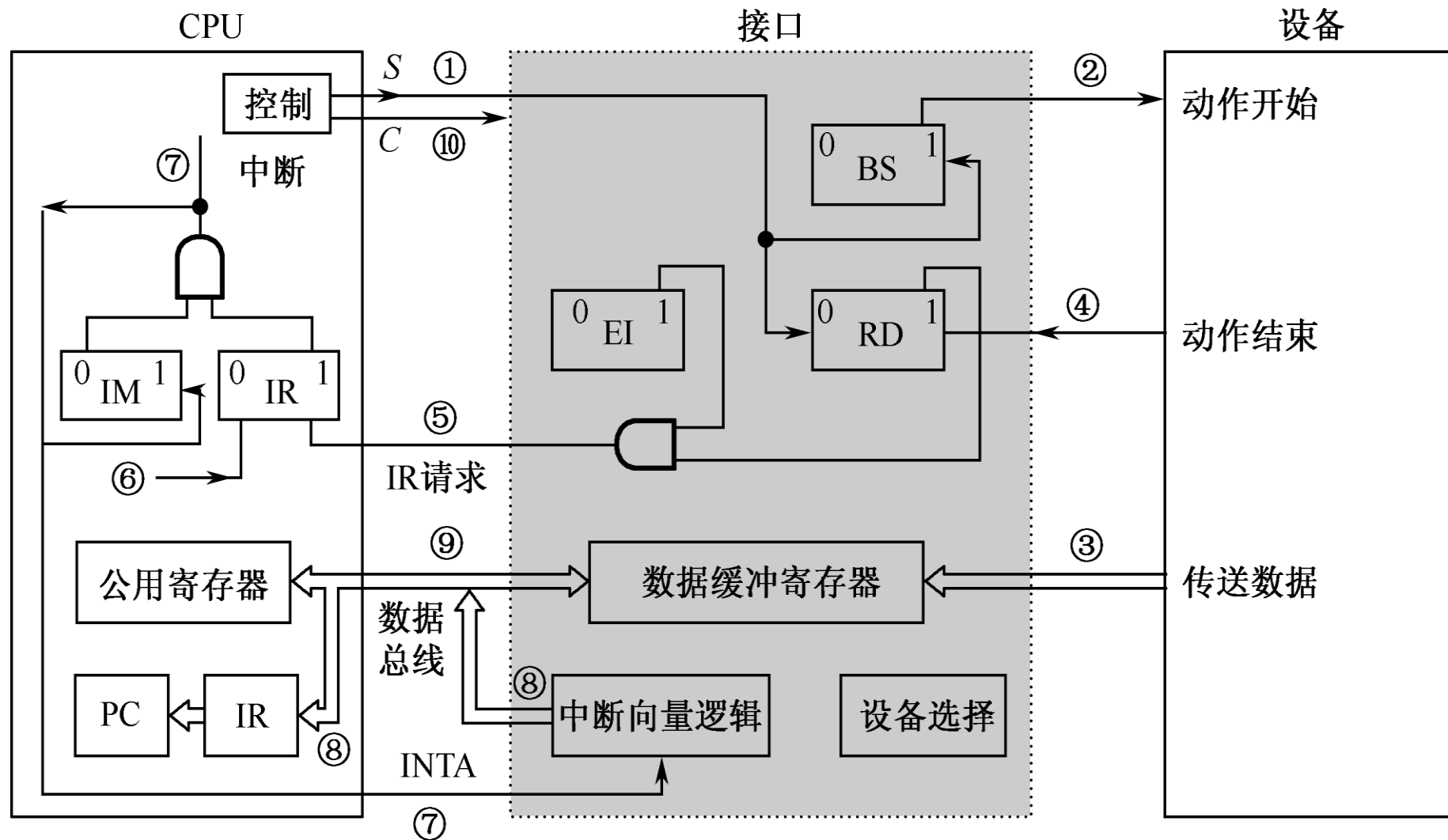
# 一、中断的基本概念

- 中断处理过程注意几个问题：
  - 响应中断时机：外界中断请求时随机的，但CPU只有在**当前指令执行完**，才转至公操作
  - **断点**保护问题（PC，寄存器内容和状态的保存）
  - 原子操作：开中断和关中断问题。
  - 中断是由软硬件结合来实现的
  - 中断分为内中断（异常）和外中断



## 二、程序中断方式的基本I/O接口

- 设备选择器。判别总线上送出的地址（或称呼叫设备）是否为本设备，它实际上是设备地址的译码比较电路。
  - BS外设接口忙（BuSy）标志
  - RD外设准备就绪（ReaDy）标志，数据准备好时，RD置1
  - EI（Enable Interrupt中断允许触发器），EI为“1”时，接口可以向CPU发出中断请求信号；
  - IR（Interrupt Request）中断请求触发器，CPU接收到“中断请求”时，标志IR=1；
  - IM（Interrupt Mask）中断屏蔽触发器，IM为“0”时，CPU受理外设的中断请求



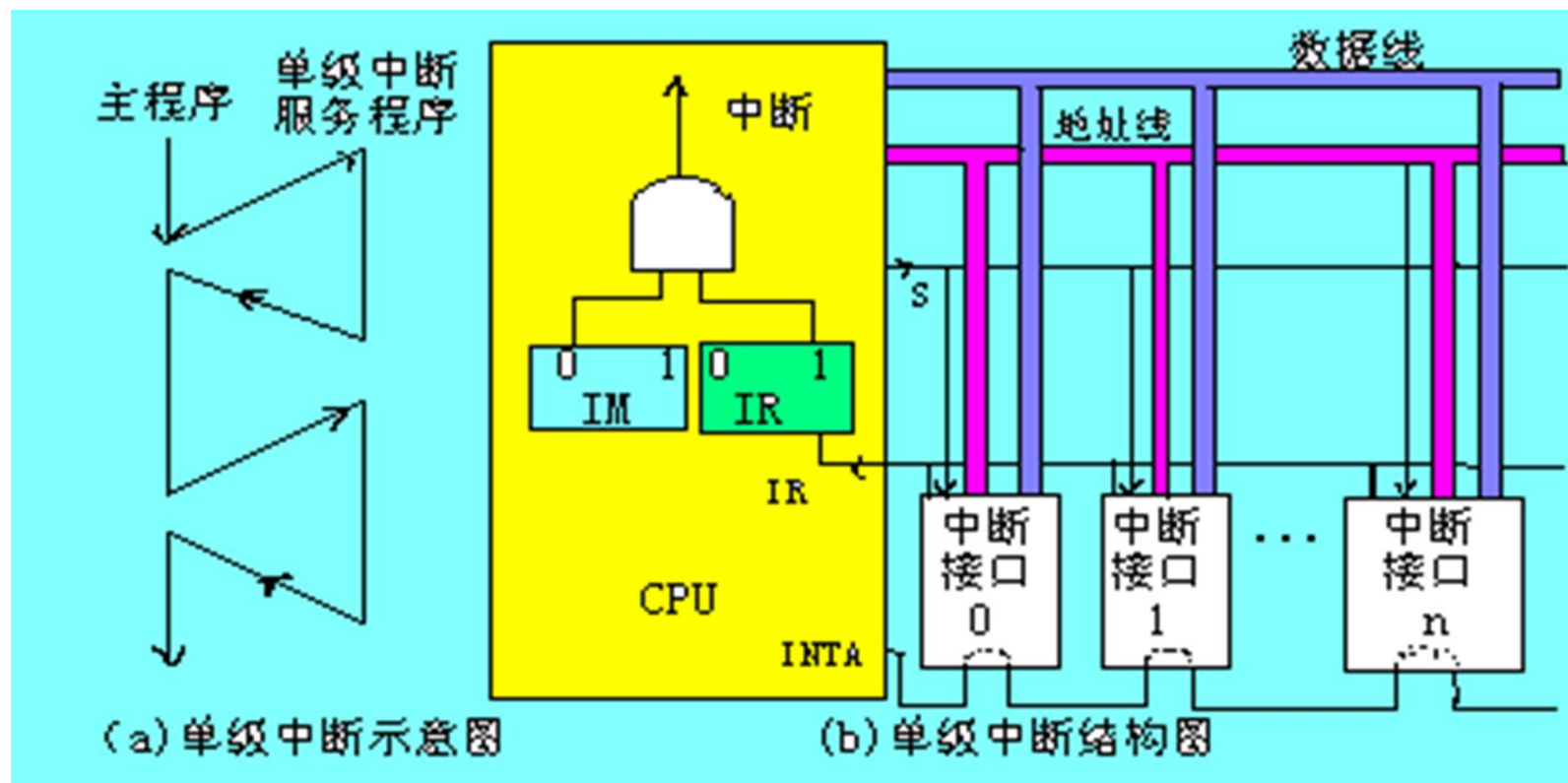
①表示CPU向接口发出控制信号，使接口开始接收数据并清除接收数据缓冲寄存器中的标志位。

②表示接口向设备发出动作开始信号，同时标志位BS置“1”，并向CPU发出中断请求，向外设发出响应中断信号并关闭中断；



## 三、单级中断

- 单级中断的概念：所有中断源属于同一级，离CPU越近，优先级越高。
- 中断源的识别：串行排队链法





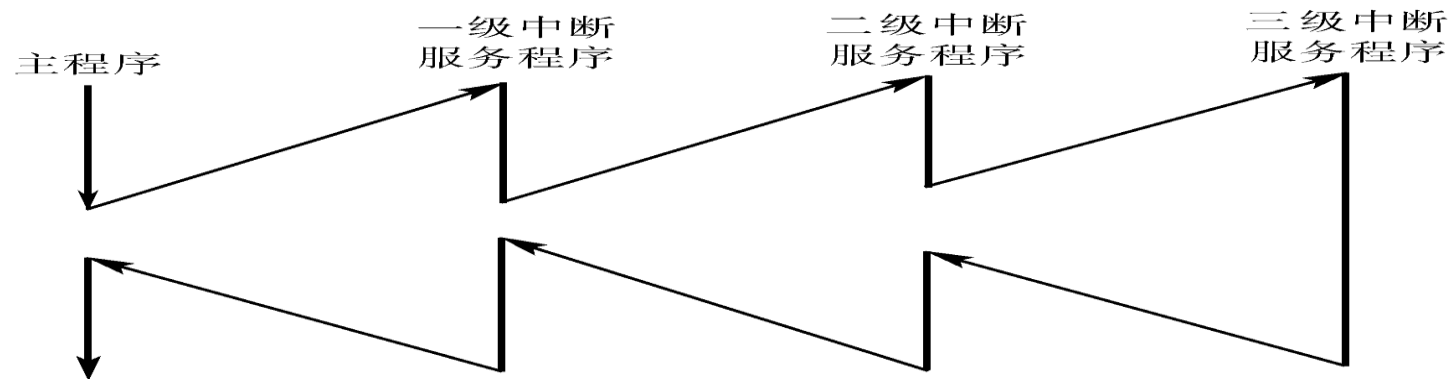
## 三、单级中断

- 中断向量：
  - 当CPU响应中断时，由硬件直接产生一个固定的地址(即向量地址)
  - 由向量地址指出每个中断源设备的**中断服务程序入口**，这种方法通常称为向量中断。

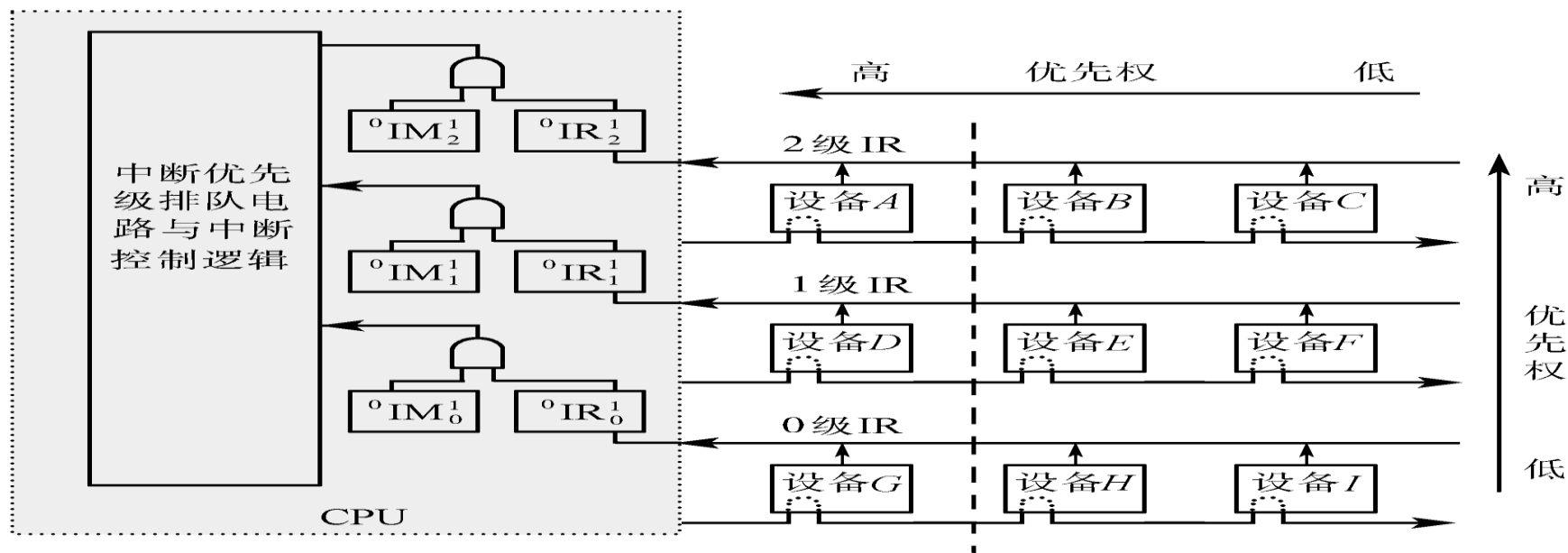
## 四、多级中断

### ● 概念

- 每级有一个中断优先权
- 一个系统有  $n$  级中断，则CPU中有  $n$  个IR，  $n$  个IM



(a) 多级中断示意图



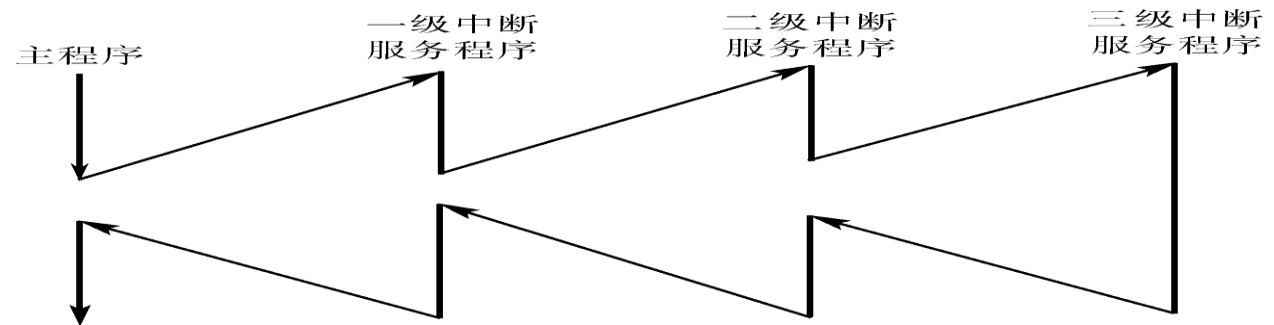
(b) 一维、二维多级中断结构





## 四、多级中断

- 某级中断被响应后，则关闭本级和低于本级的IM，开放更高级的IM。
- 多级中断可以嵌套，但同一级的中断不允许嵌套

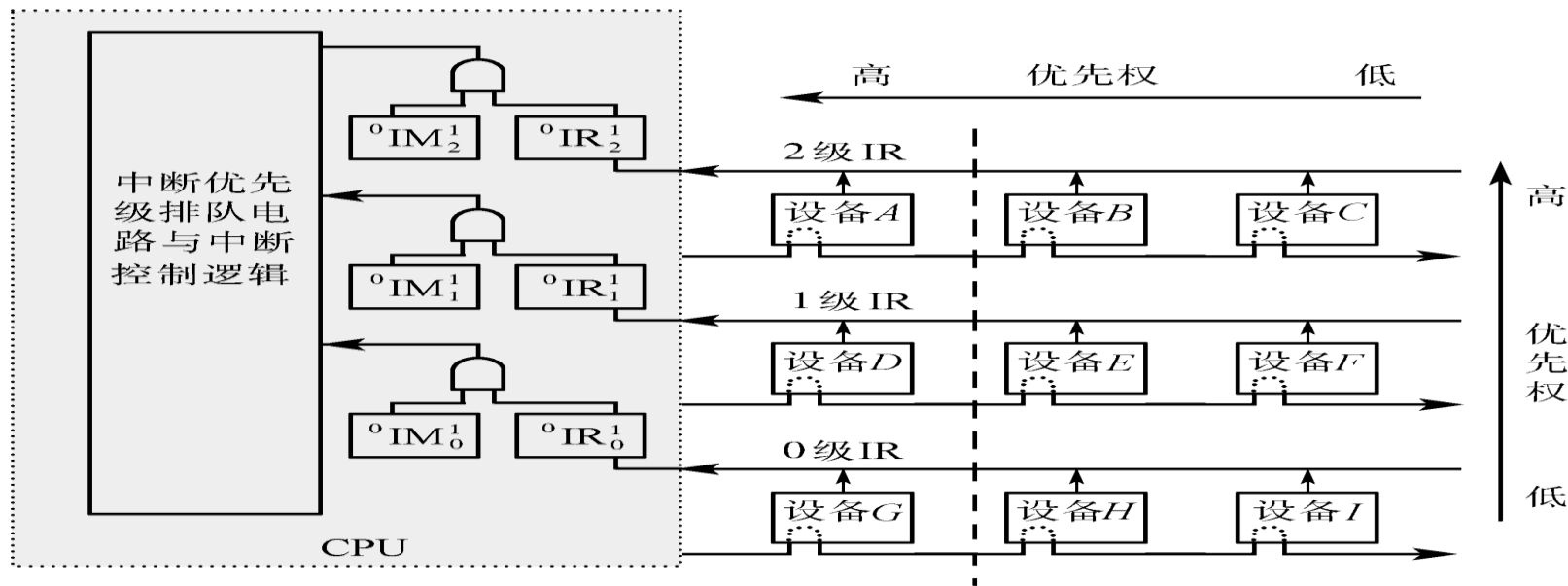


(a) 多级中断示意图



## 四、多级中断

- 中断响应时，确定哪一级中断和中断源采用硬件实现。
- 采用了独立请求方式和链式查询方式相结合的方式。
- 使用多级堆栈保存现场

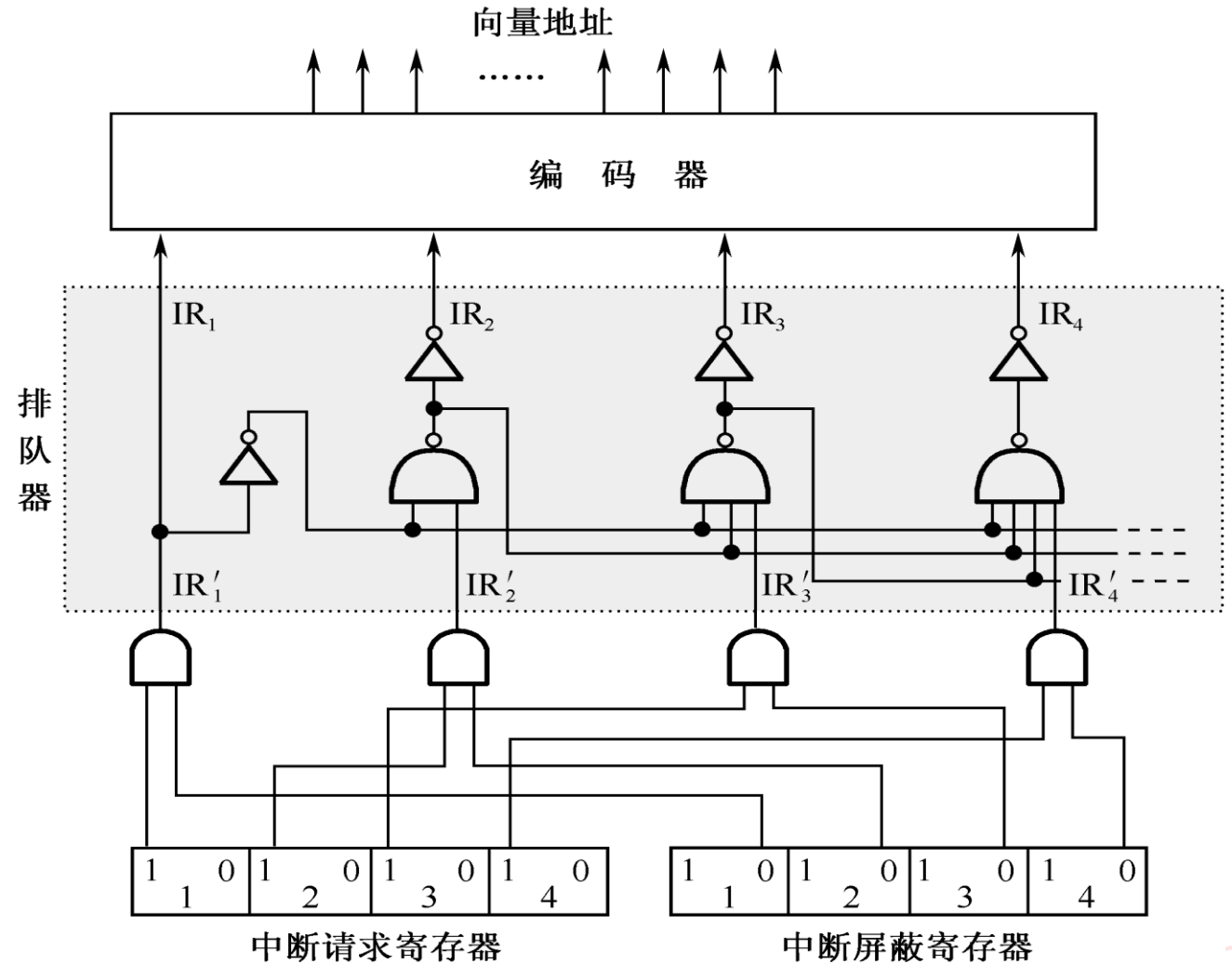


(b) 一维、二维多级中断结构



## 四、多级中断

- 多级中断源的识别
  - 中断优先排队电路
  - 中断向量产生电路

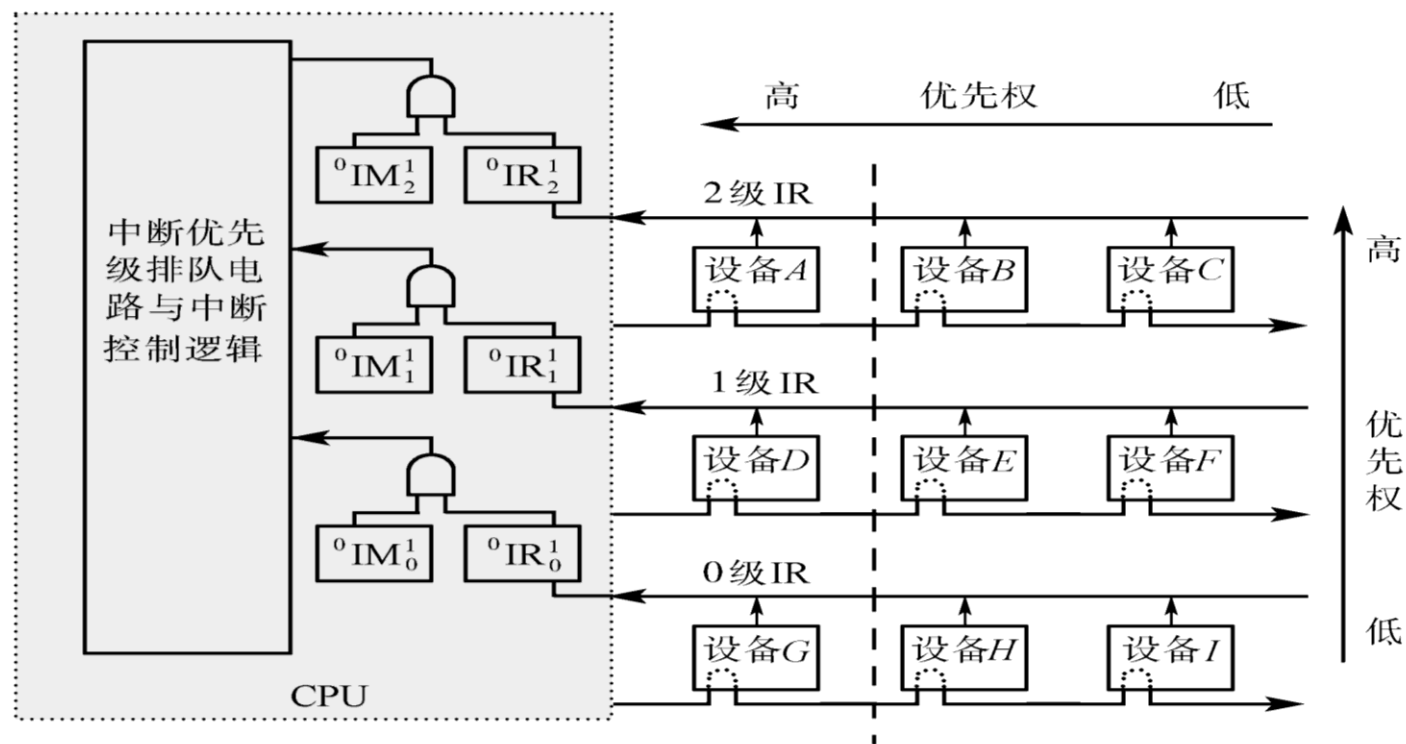


## 四、多级中断

- **【例8.2】**、参见图8.12所示的二维中断系统。请问：
- (1)在中断情况下，CPU和设备的优先级如何考虑?请按降序排列各设备的中断优先级。
- (2)若CPU现执行设备B的中断服务程序，IM2，IM1，IM0的状态是什么?如果CPU执行设备D的中断服务程序，IM2，IM1，IM0的状态又是什么?
- (3)每一级的IM能否对某个优先级的个别设备单独进行屏蔽?如果不能，采取什么办法可达到目的?
- (4)假如设备C一提出中断请求，CPU立即进行响应，如何调整才能满足此要求?

## 四、多级中断

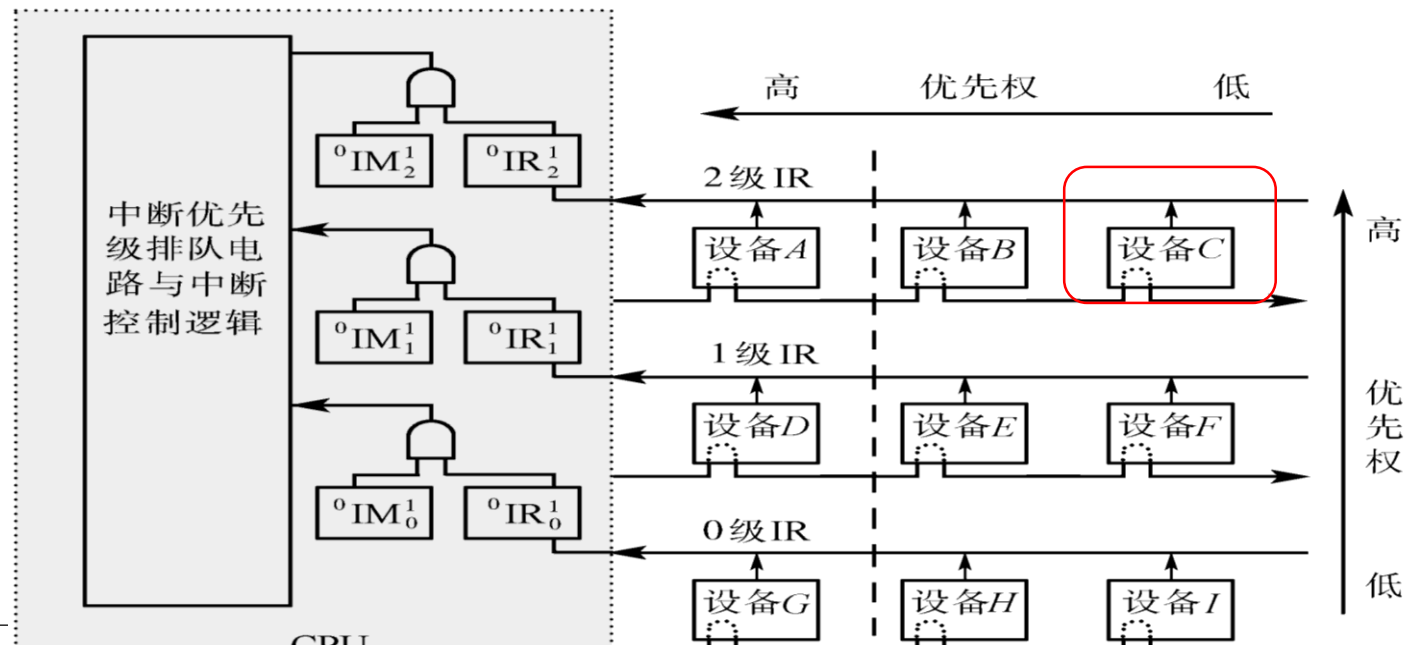
- (1) 在中断情况下，CPU 的优先级最低。各设备的优先次序是： $A \rightarrow B \rightarrow C \rightarrow D \rightarrow E \rightarrow F \rightarrow G \rightarrow H \rightarrow I \rightarrow \text{CPU}$ 。
- (2) 执行设备 B 的中断服务程序时  $IM_2IM_1IM_0 = 111$ ；执行设备 D 的中断服务程序时， $IM_2IM_1IM_0 = 011$ 。



(b) 一维、二维多级中断结构

## 四、多级中断

- (3)每一级的IM标志不能对某个优先级的个别设备单独屏蔽。可将接口中的EI(中断允许)标志清“0”，它禁止设备发出中断请求。
- (4)要使设备C的中断请求及时得到响应，可将设备C从第2级取出来，单独放在第3级上，使第3级的优先级最高，即令IM3=0即可。



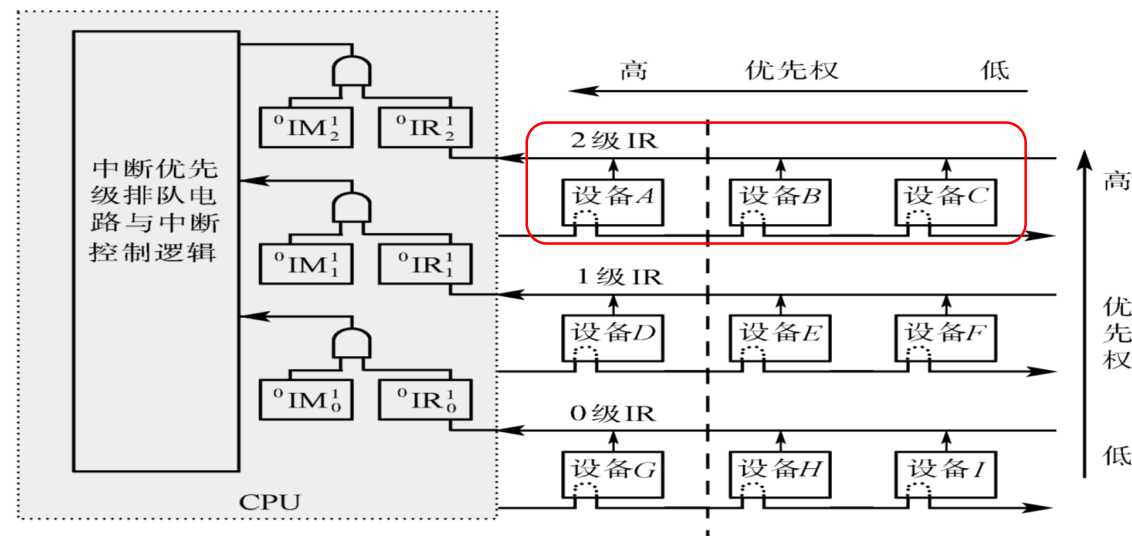


## 四、多级中断

- **【例8.3】** 参见图8.12所示的系统，只考虑A，B，C三个设备组成的单级中断结构，它要求CPU在执行完当前指令时对中断请求进行服务。假设：(1)CPU“中断批准”机构在响应一个新的中断之前，先要让被中断的程序的一条指令一定要执行完毕；(2) $T_{DC}$ 为查询链中每个设备的延迟时间；(3) $T_A$ ， $T_B$ ， $T_C$ 分别为设备A，B，C的服务程序所需的执行时间；(4) $T_S, T_R$ 为保存现场和恢复现场所需的时间；(5)主存工作周期为 $T_M$ 。试问：就这个中断请求环境来说，系统在什么情况下达到中断饱和？

## 四、多级中断

- [例8.3] 解：中断处理流程，并假设执行一条指令的时间也为 $T_M$ 。如果三个设备同时发出中断请求，那么依次分别处理设备A、设备B、设备C的时间如下：
- $t_A = 2T_M + T_{DC} + T_S + T_A + T_R$
- $t_B = 2T_M + 2T_{DC} + T_S + T_B + T_R$
- $t_C = 2T_M + 3T_{DC} + T_S + T_C + T_R$
- 处理三个设备所需的总时间为：
- $T = t_A + t_B + t_C$
- $T$ 是达到中断饱和的最小时间，即中断极限频率为： $f = 1/T$



(b) 一维、二维多级中断结构





## 8.4 DMA方式

- 一. DMA的基本概念
- 二. DMA传送方式
- 三. 基本的DMA原理
- 四. 选择型和多路型DMA控制器



# 一、DMA的基本概念

- **直接存储器访问**（Direct Memory Address）DMA方式是为了在**主存储器与I / O设备**间高速**交换批量**数据而设置的。
- 基本思想是：通过硬件控制实现主存与I / O设备间的**直接数据传送**，在传送过程中**无需CPU的干预**。数据传送是在DMA控制器控制下进行的，
- 优点：速度快。有利于发挥CPU的效率。



# 一、DMA的基本概念

- DMA的基本操作。
  - 从外设发出DMA 请求
  - CPU响应请求， DMA控制器获得总线控制权
  - DMA控制器对内存寻址， 确定内存单元地址和传送数据量
  - 数据传送
  - 向CPU报告DMA操作结束

## 二、DMA传送方式

- DMA与CPU分时使用内存的三种方法：
  - 1、停止CPU访问内存
  - 2、周期挪用方式
  - 3、DMA与CPU交替访内



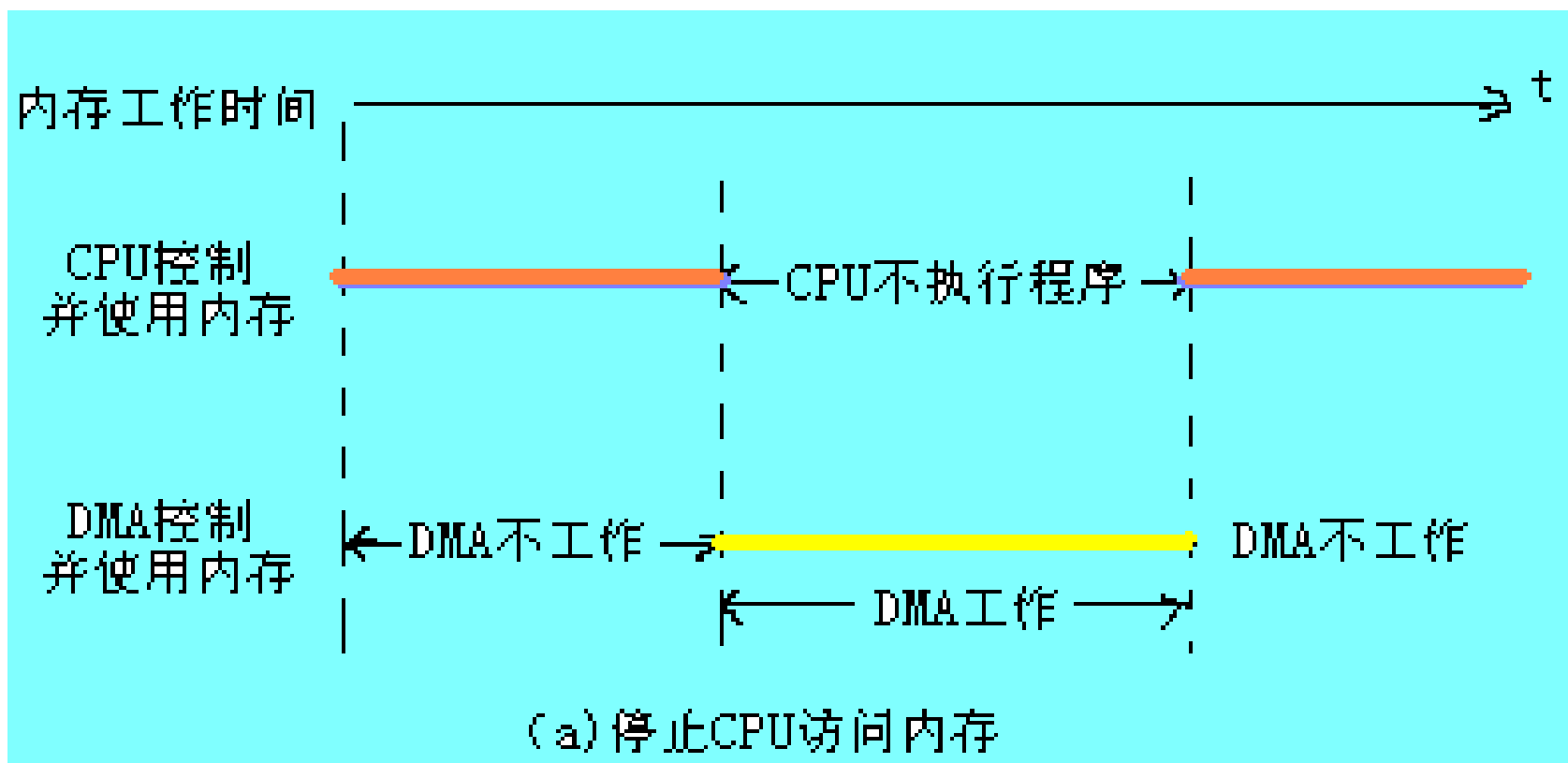
## 二、DMA传送方式

- 1、停止CPU访问内存
- 主机响应DMA请求后，**让出总线**。直到一组数据**传送完毕**，DMA控制器才把总线控制权**交还**给CPU。
- 采用DMA方式的I / O设备，在其接口中一般设置有小容量存储器，I / O设备先与小容量存储器交换数据，**小容量存储器**再与主机交换数据，**减少DMA传送占用总线的时间**，也即减少了CPU暂停工作的时间。



## 二、DMA传送方式

- 1、停止CPU访问内存





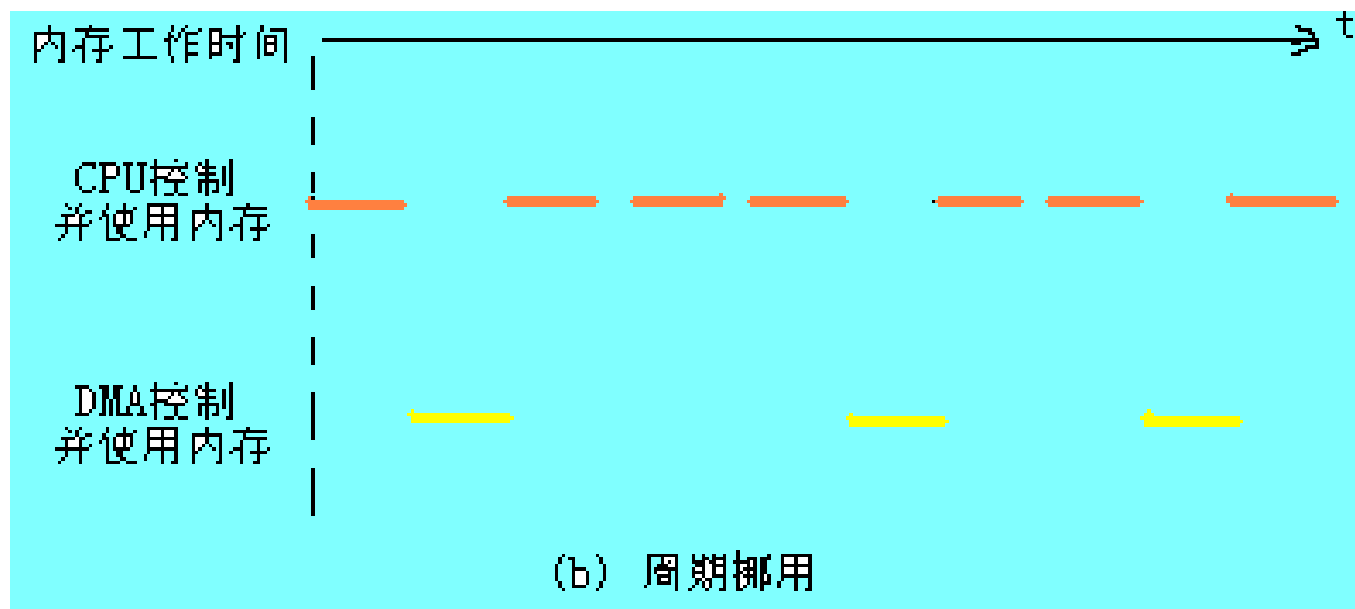
## 二、DMA传送方式

- 1、停止CPU访问内存
  - **优点：**控制简单，适用于数据传输率很高的设备进行成组传送。
  - **缺点：**在DMA控制器访存阶段，内存的效能没有充分发挥，相当一部分内存工作周期是空闲的。



## 二、DMA传送方式

- 2、周期挪用方式
  - DMA控制器与主存储器之间传送数据时，占用（窃取）一个或多内存周期。







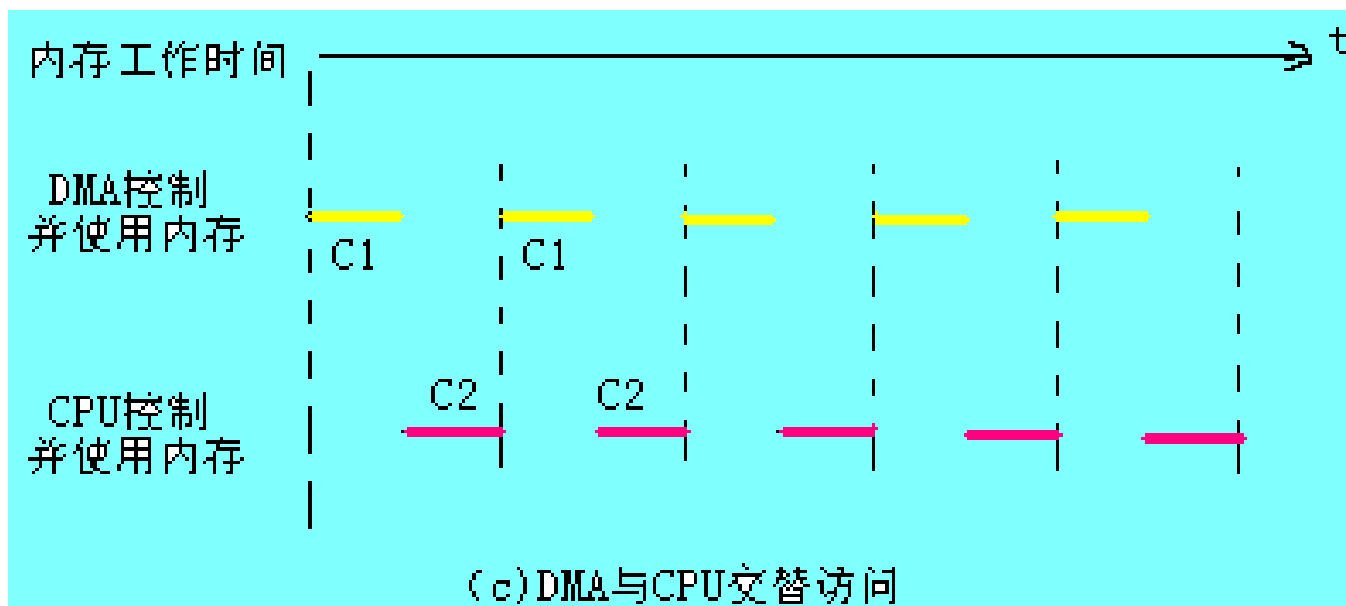
## 二、DMA传送方式

- 2、周期挪用方式
- 两种情况：
  - （1）CPU在DMA传送时不需要访问内存，I/O设备挪用一两个内存周期对CPU执行程序没有影响。
  - （2）I/O设备访存时，CPU也需要访存，会产生冲突。CPU延缓指令执行，让外设
- 适用于I/O设备读写周期大于内存存储周期的情况。



## 二、DMA传送方式

- 3、DMA与CPU交替访内
  - 如果CPU的工作周期比内存存取周期长很多，可以采用交替访内方法
  - 总线控制权的转移速度快，DMA效率高。



- 1、DMA基本构

- (1) 内存地址计数器：  
数据的起始地址
- (2) 字计数器： 传送  
数据块的长度。进  
位时触发中断机构  
发生中断请求



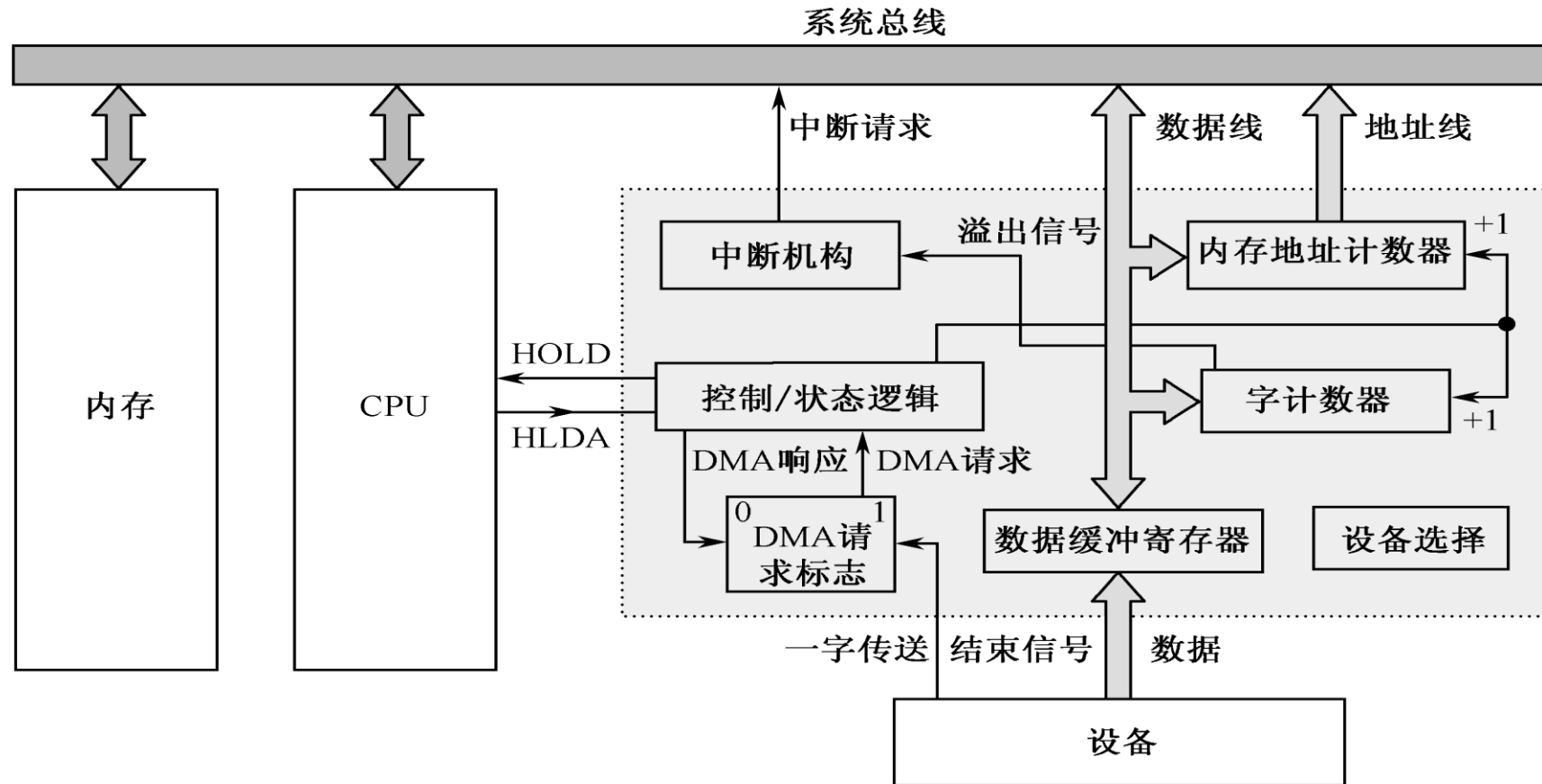


# 三、基本的DMA原理

## • 1、DMA基本构成

成

- (3)数据缓冲寄存器
- (4)“DMA请求”标志
- (5)“控制/状态”逻辑
- (6)中断机构





## 三、基本的DMA原理

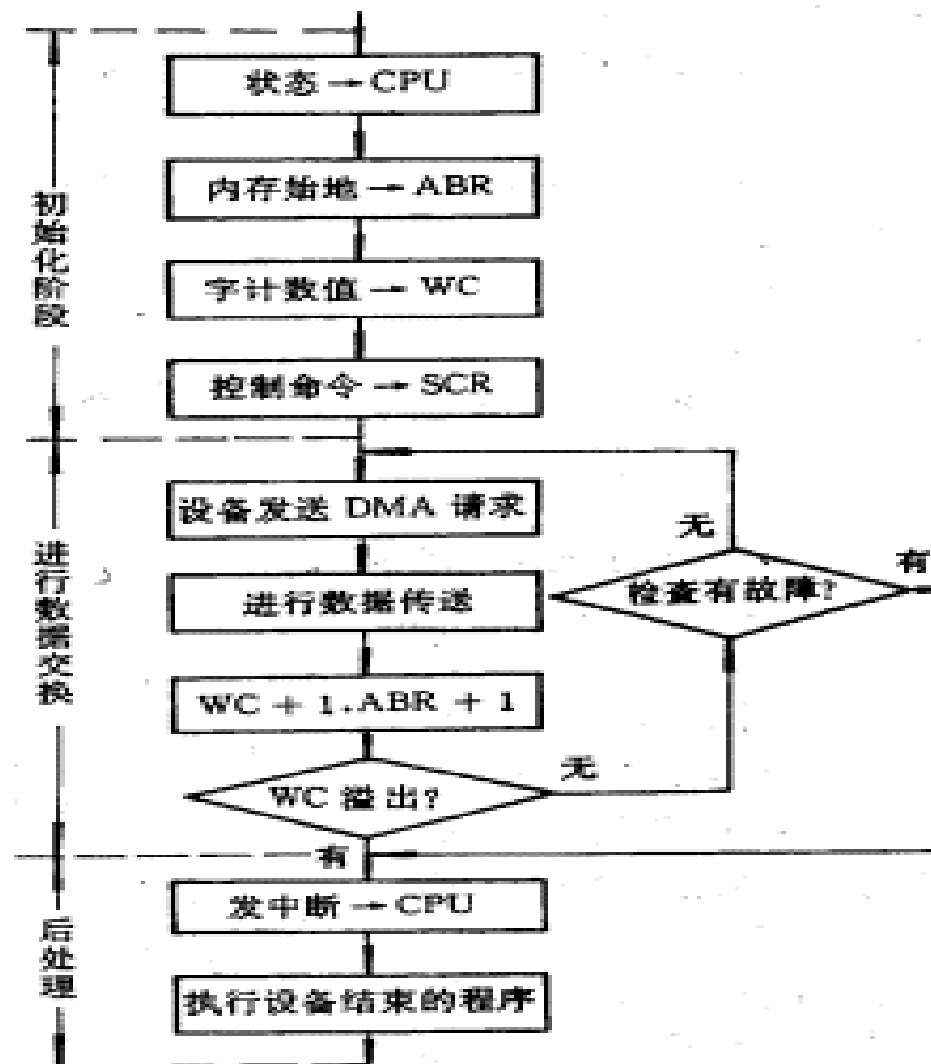
- 2、DMA数据传送过程
- 当外设有DMA请求时，通常CPU在本机器周期结束后，响应DMA请求。



## 三、基本的DMA原理

### • 2、 DMA数据传送过程:

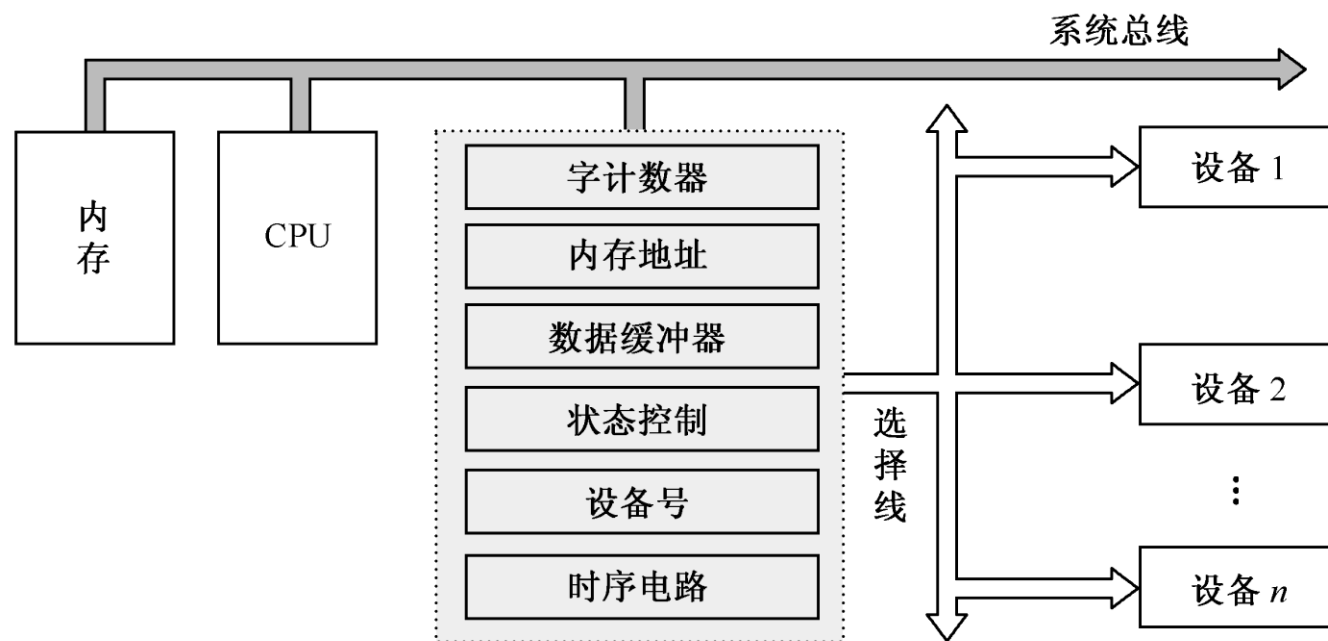
- 传送前预处理（初始化）。
- 正式传送
- 传送后处理





## 四、选择型和多路型DMA控制器

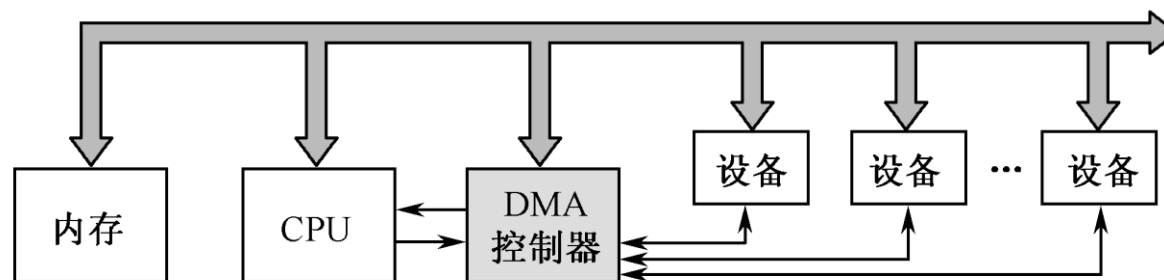
- 选择型
- 物理上可以连接多个设备，逻辑上只允许一个





## 四、选择型和多路型DMA控制器

- 多路型
- 物理上可以连接多个设备，逻辑上也允许连接多个设备
- 独立的信号和独立的寄存器

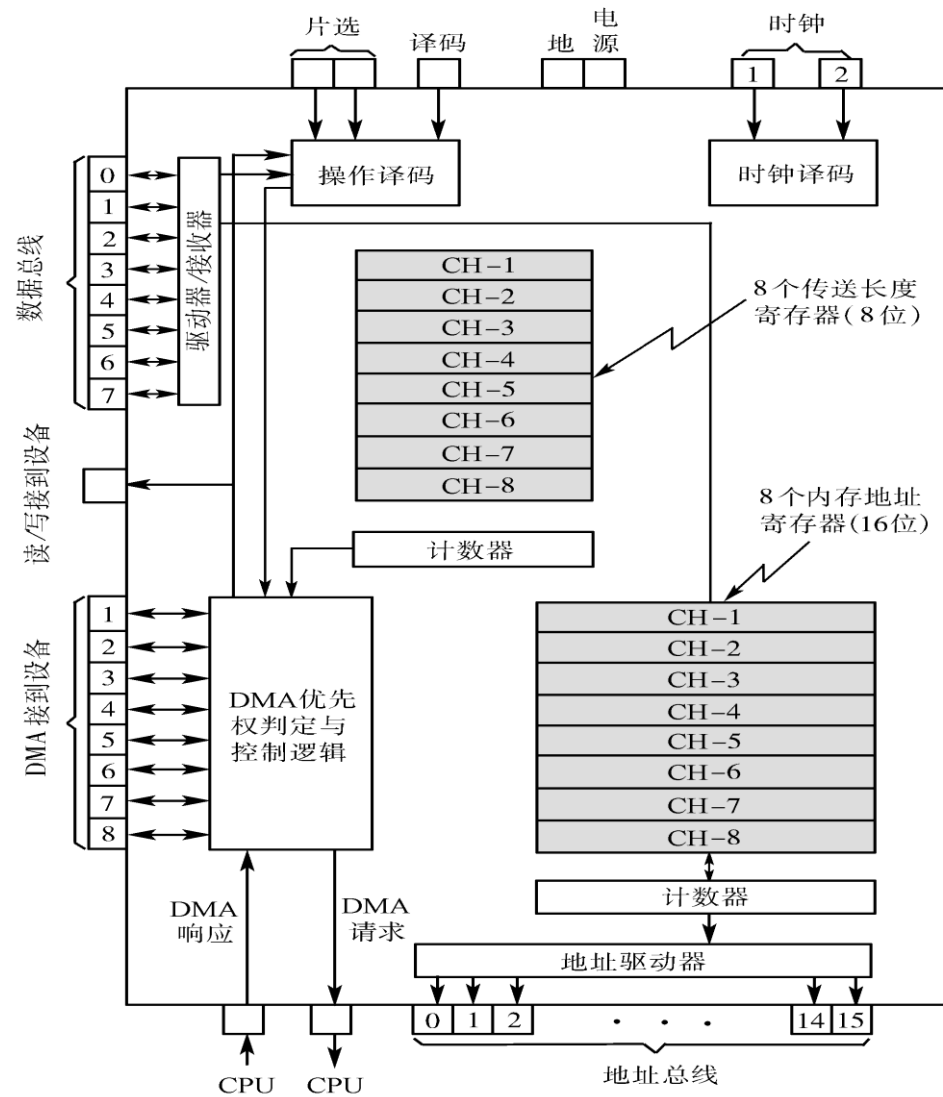






## 四、选择型和多路型DMA控制器

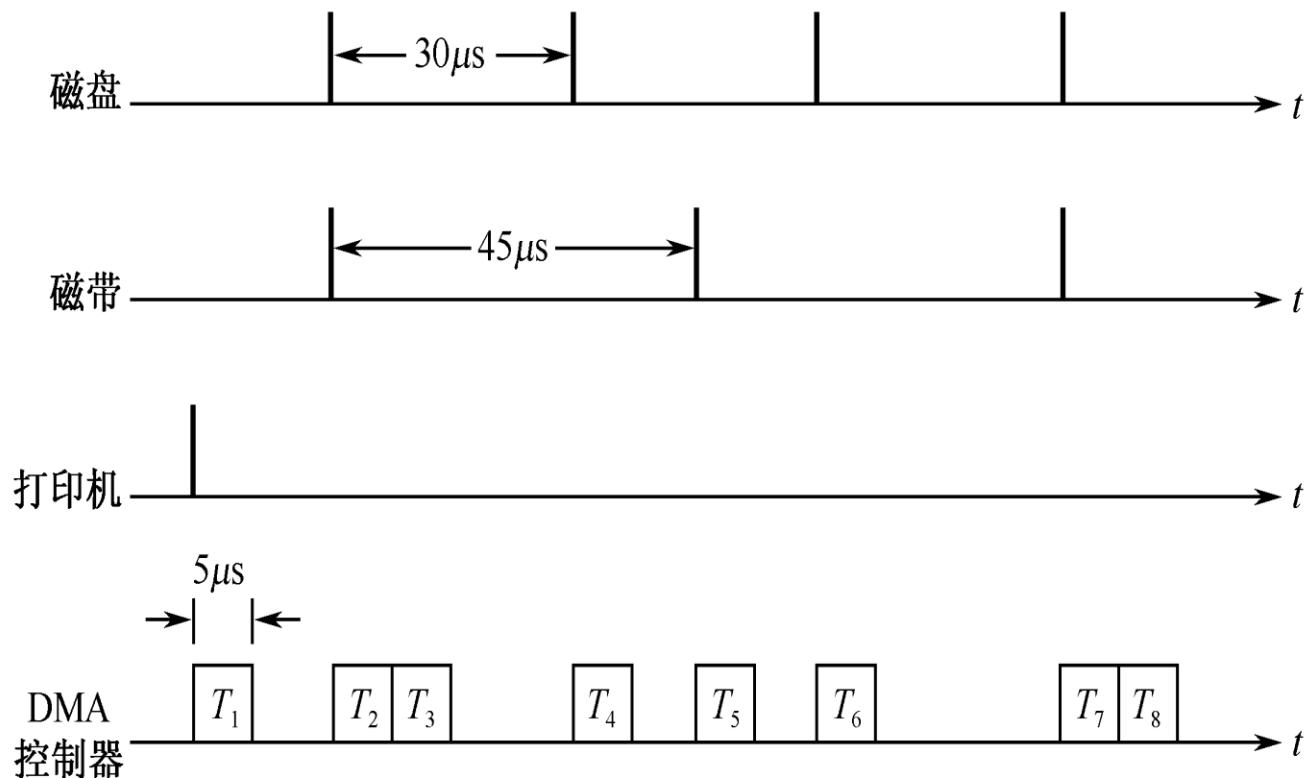
### ● 典型DMA芯片





## 四、选择型和多路型DMA控制器

- 【例8.4】：假设有磁盘、磁带、打印机三个设备同时工作。磁盘以 $30\mu\text{s}$ 的间隔向控制器发DMA请求，磁带以 $45\mu\text{s}$ 的间隔发DMA请求，打印机以 $150\mu\text{s}$ 间隔发DMA请求。
- 磁盘优先权最高，磁带次之，打印机最低。假设DMA控制器每完成一次DMA传送所需的时间是 $5\mu\text{s}$ 。
- 若采用多路型DMA控制器，请画出DMA控制器服务三个设备的工作时间图。





## 四、选择型和多路型DMA控制器

- 解：T1间隔中控制器首先为打印机服务，因为此时只有打印机有请求。
- T2间隔前沿磁盘、磁带同时有请求，首先为优先权高的磁盘服务，然后为磁带服务，每次服务传送一个字节。
- 在120 $\mu$ s时间阶段中，为打印机服务只有一次(T1)，为磁盘服务四次(T2, T4, T6, T7)，为磁带服务三次(T3, T5, T8)。
- DMA尚有空闲时间，说明控制器还可以容纳更多设备。



## 8.5 通道方式

- 一. 通道的功能
- 二. 通道的类型
- 三. 通道结构的发展



# 一、通道的功能

- 执行通道指令，组织外围设备和内存进行数据传输，按I/O指令要求启动外围设备，向CPU报告中断等，具体有以下五项任务：
  - (1) 接受CPU的I/O指令，按指令要求与指定的外围设备进行通信。



# 一、通道的功能

- 执行通道指令，组织外围设备和内存进行数据传输，按I/O指令要求启动外围设备，向CPU报告中断等，具体有以下五项任务：
  - (2) 从内存选取属于该通道程序的通道指令，经译码后向设备控制器和设备发送各种命令。
  - (3) 组织外围设备和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。



# 一、通道的功能

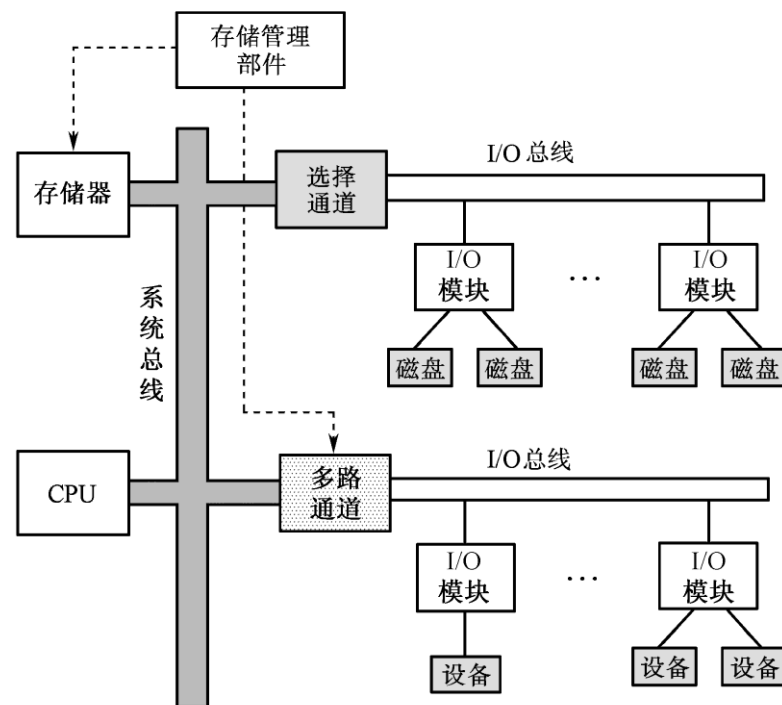
- 执行通道指令，组织外围设备和内存进行数据传输，按I/O指令要求启动外围设备，向CPU报告中断等，具体有以下五项任务：
  - (4) 从外围设备得到设备的状态信息，形成并保存通道本身的状态信息，根据要求将这些状态信息送到内存的指定单元，供CPU使用。
  - (5) 将外围设备的中断请求和通道本身的中断请求，按次序及时报告CPU。



# 一、通道的功能

- 通道结构

- 在一般用户程序中，通过调用通道来完成一次数据输入输出的过程如图1所示。
- CPU执行用户程序和管理程序，通道处理机执行通道程序的时间关系如图所示。







## 二、通道的类型

### ● 选择通道

- 选择通道每次只能从所连接的设备中**选择一台**I / O设备的通道程序，此刻该通道程序独占了**整个通道**。连接在选择通道上的若干设备，只能依次使用通道与主存传送数据。
- 数据传送以**成组**（数据块）方式进行，每次传送一个数据块，因此，传送速率很高。选择通道多适合于快速设备（磁盘），这些设备相邻字之间的传送空闲时间极短。



## 二、通道的类型

### ● 字节多路通道 (Byte Multiplexor Channel)

- 是一种简单的共享通道，在时间分割的基础上，服务于多台低速和中速面向字符的外围设备。
- 字节多路通道包括多个子通道，每个子通道服务于一个设备控制器，可以独立地执行通道指令。每个子通道都需要有字符缓冲寄存器、I / O请求标志 / 控制寄存器、主存地址寄存器和字节计数寄存器。而所有子通道的控制部分是公共的，由所有子通道所共享。



## 二、通道的类型

- **字节多路通道 (Byte Multiplexor Channel)**
  - 通常，每个通道的有关指令和参量存放在主存固定单元中。当通道在逻辑上与某一设备连通时，将这些指令和参量取出来，送入公共控制部分的寄存器中使用。
  - 字节多路通道要求每种**设备分时占用**一个很短的时间片，不同的设备在各自分得的时间片内与通道建立传输连接，实现数据的传送。

## 二、通道的类型

- 数组多路通道（Block Multiplexor Channel）
  - 数组多路通道把字节多路通道和选择通道的特点结合起来。它有多个子通道，既可以执行多路通道程序，象字节多路通道那样，所有子通道分时共享总通道；又可以用选择通道那样的方式传送数据。
  - 数组多路通道和字节多路通道的比较



## 三、通道结构的发展

- 输入输出处理机（IOP）
  - 输入输出处理机（IOP）不是一台独立的计算机，而是计算机系统中的一个部件。IOP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。此外，有些IOP还提供数据的变换、搜索和字装配 / 分拆能力。
  - 8位和16位微机中使用的Intel 8089 I / O处理器就是这种通道型I / O处理器



## 三、通道结构的发展

- 外围处理机
  - 外围处理机结构更接近于一般处理机，或者就是选用已有的通用机。外围机基本上是独立于主处理机工作的，应用于大型高效率的计算机系统中。

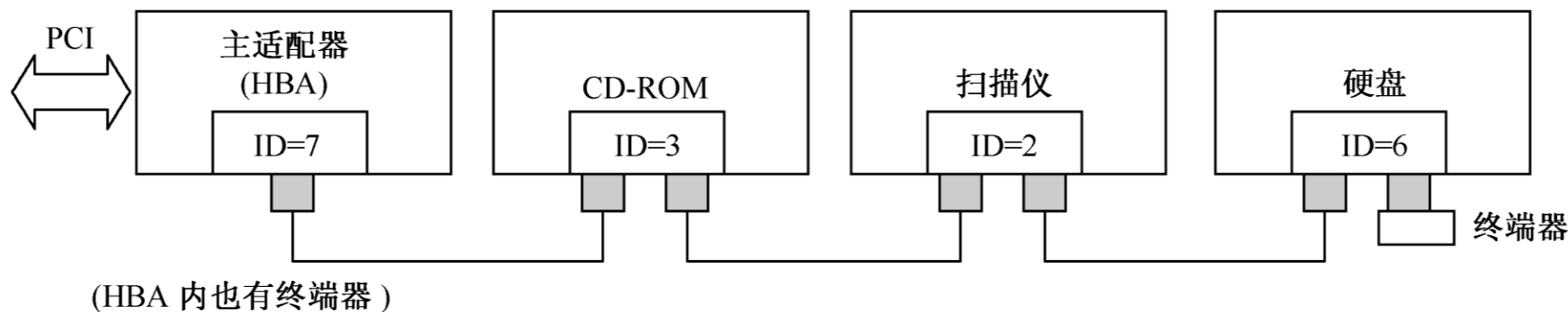


## 8.6 通用I/O接口标准

- 一. 并行I/O标准接口SCSI
- 二. 串行I/O标准接口IEEE1394

# 一、并行I/O标准接口SCSI

- 小型计算机系统接口的简称，它是一个高速智能接口，可以混接各种磁盘、光盘、磁带机、打印机、扫描仪、条码阅读器以及通信设备。

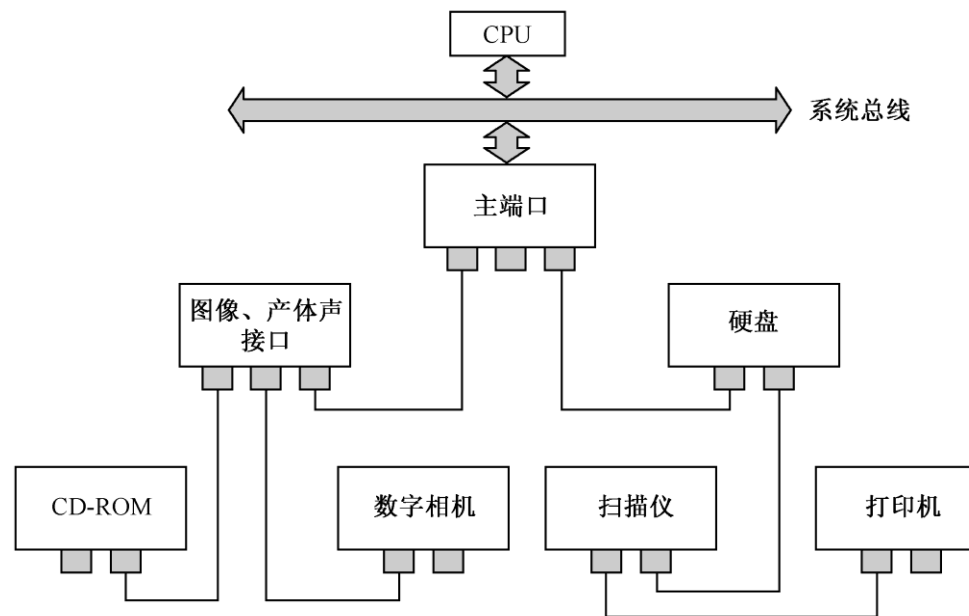






## 二、串行I/O标准接口IEEE1394

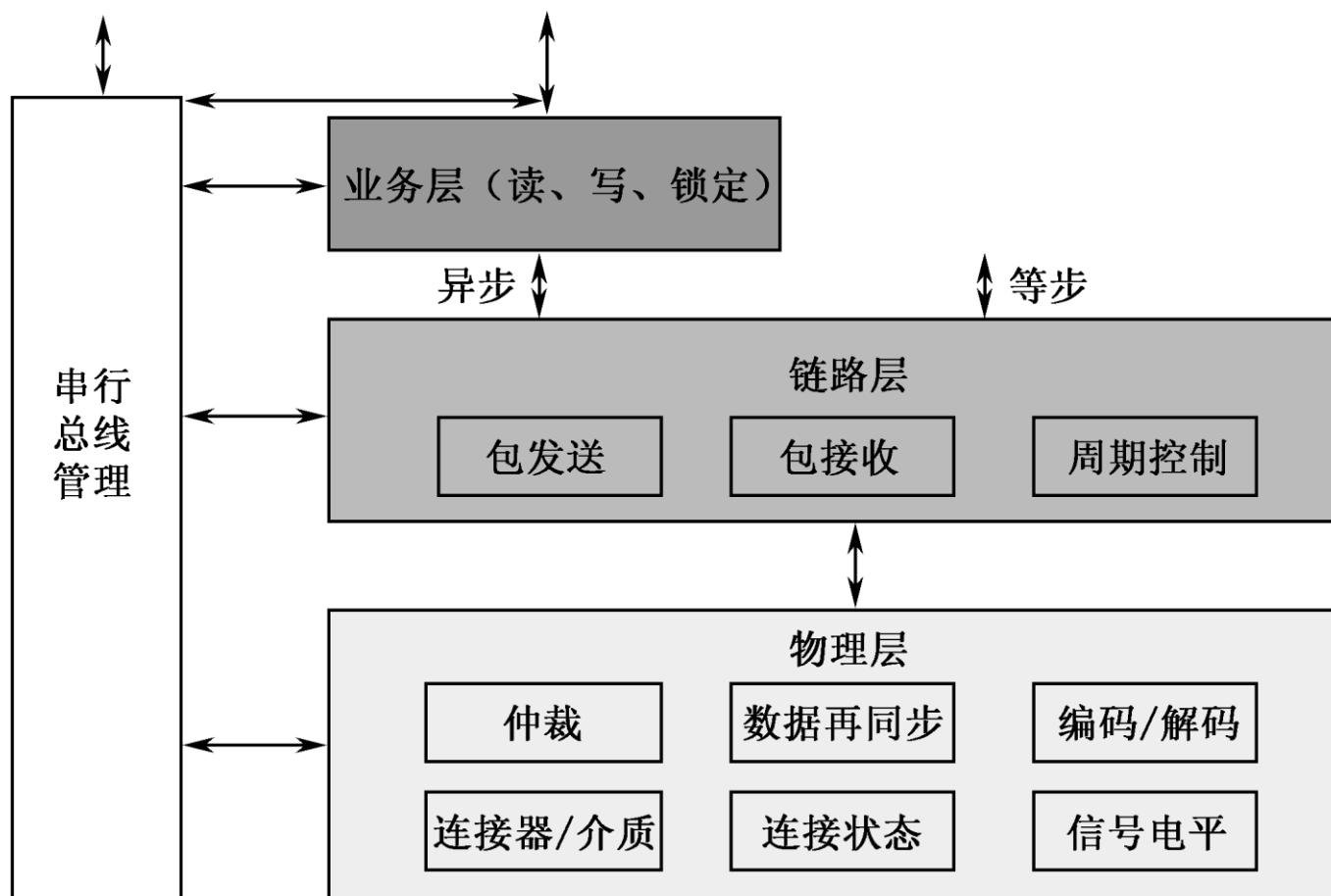
- IEEE 1394是一种高速串行I/O标准接口。各被连接装置的关系是平等的，不用PC介入也能自成系统。这意味着1394在家电等消费类设备的连接应用方面有很好的前景。
  - (1) 数据传送的高速性
  - (2) 数据传送的实时性
  - (3) 体积小易安装，连接方便





## 二、串行I/O标准接口IEEE1394

• 图8.23 IEEE 1394协议集



# 本章小结

- 各种外围设备的数据传输速率相差很大。如何保证主机与外围设备在时间上同步，则涉及外围设备的定时问题。在计算机系统中，CPU对外围设备的管理方式有：①程序查询方式；②程序中断方式；③DMA方式；④通道方式。每种方式都需要硬件和软件结合起来进行。
- **程序查询方式**是CPU管理I/O设备的最**简单**方式，CPU定期执行设备服务程序，主动来了解设备的工作状态。这种方式浪费CPU的宝贵资源。

# 本章小结

- **程序中斷方式**是各类计算机中**广泛**使用的一种数据交换方式。当某一**外设**的数据准备就绪后，它“**主动**”向CPU发出请求信号。CPU响应中断请求后，暂停运行主程序，自动转移到该设备的中断服务子程序，为该设备进行服务，结束时返回主程序。中断处理过程可以嵌套进行，优先级高的设备可以中断优先级低的中断服务程序。



# 本章小结

- DMA技术的出现，使得外围设备可以通过DMA控制器**直接访问内存**，与此同时，CPU可以继续程序。DMA方式采用以下三种方法：①停止CPU访内；②周期挪用；③DMA与CPU交替访内。DMA控制器按其组成结构，分为选择型和多路型两类。
- 通道是一个特殊功能的处理器。它有自己的指令和程序专门负责数据输入输出的传输控制，从而使CPU将“**传输控制**”的功能下放给**通道**，CPU只负责“数据处理”功能。这样，通道与CPU**分时使用内存**，实现了CPU内部的数据处理与I/O设备的平行工作。通道有两种类型：①选择通道；②多路通道。



# 本章小结

- 标准化是建立开放式系统的基础。CPU、系统总线、I/O总线及标准接口技术近年来取得了重大进步。其中并行I/O接口SCSI与串行I/O接口IEEE 1394是两个最具权威性和发展前景的标准接口技术。
- SCSI是系统级接口，是处于主适配器和智能设备控制器之间的并行I/O接口，改进的SCSI可允许连接1~15台不同类型的高速外围设备。SCSI的不足处在于硬件较昂贵，并需要通用设备驱动程序和各类设备的驱动程序模块的支持。

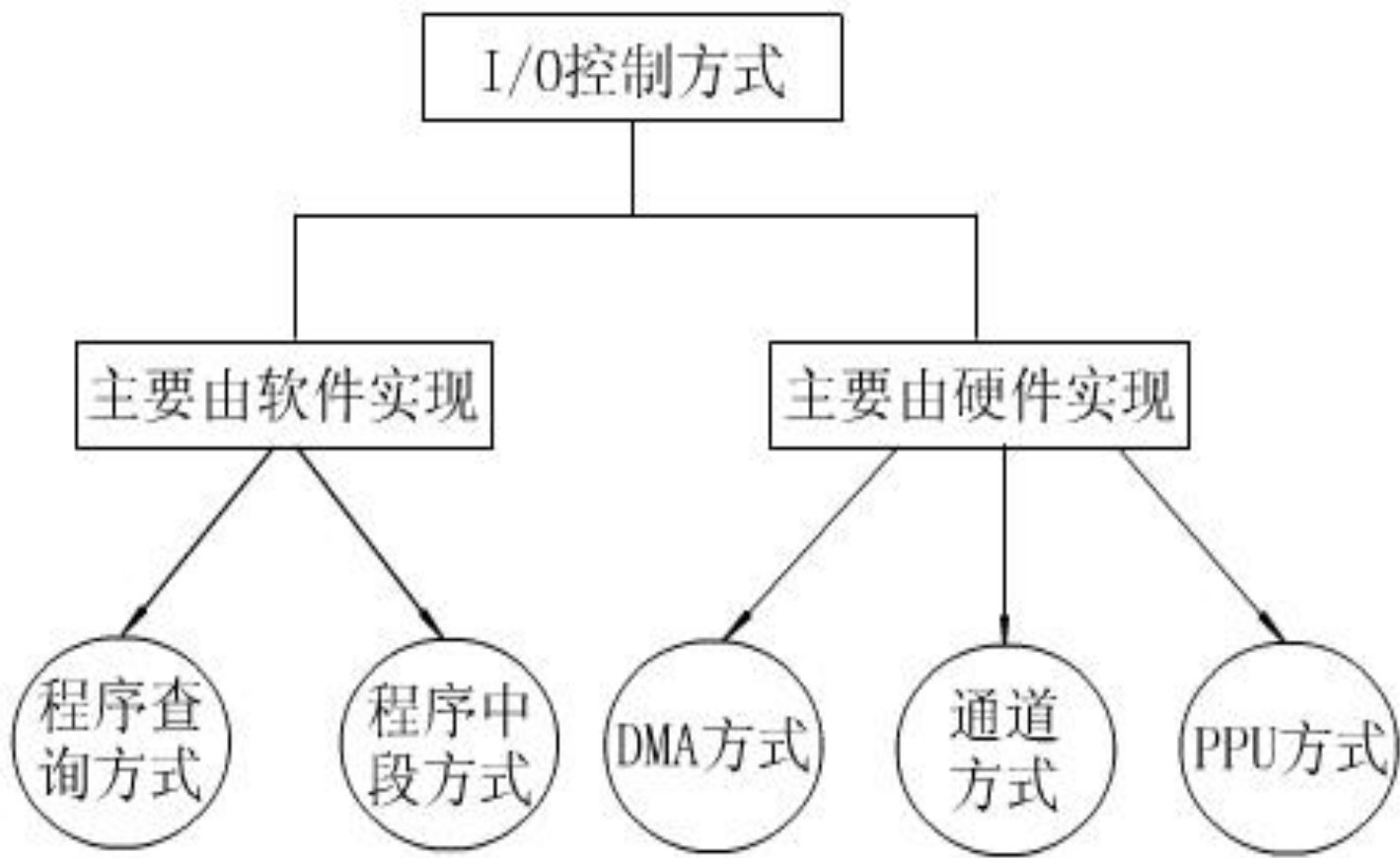
# 本章小结

- IEEE 1394是串行I/O标准接口。与SCSI并行I/O接口相比，它具有更高的数据传输速率和数据传送的实时性，具有更小的体积和连接的方便性。IEEE 1394的一个重大特点是，各被连接的设备的关系是平等的，不用PC介入也能自成系统。因此IEEE 1394已成为Intel、Microsoft等公司联手制定的新标准。



# 本章小结

- 图8.1 外围设备的输入/输出方式







# 作业

- 1. 在程序中断方式下，中断响应发生在\_\_\_\_
  - A. 一条指令执行结束
  - B. 一条指令执行开始
  - C. 一条指令执行中间
  - D. 一条指令执行的任何时刻



# 作业

- 2.DMA接口具有的功能有\_\_\_\_
  - A. 向CPU申请DMA传送
  - B. 在CPU允许DMA工作时，接管总线控制权
  - C. 在DMA期间管理系统总线，控制数据传送
  - D. 确定数据传送的起始地址和数据长度
  - E. 在数据块传送结束时给出DMA操作完成的信号



# 作业

- 3. DMA控制器和CPU在竞争内存时，可以使用以下方式\_\_\_\_
  - A. 停止CPU访问内存
  - B. 周期挪用
  - C. DMA与CPU交替访问
  - D. 停止DMA访问内存