

# 第八章 输入/输出系统

- 一、外围设备的速度分级与信息交换方式
- 二、程序查询方式
- 三、程序中断方式
- 四、**DMA**方式
- 五、通道方式
- 六、通用**I/O**标准接口

# 8.1 外围设备的速度分级 与信息交换方式

- 接口的定义：为CPU和主存、I/O设备之间传送信息而设的转换逻辑部件。
- 输入输出设备同CPU交换数据的过程：
  - ◆ 输入过程：
    - CPU送地址选择某一输入设备；
    - CPU等候数据成为有效；
    - CPU读入数据，存入相应的寄存器中。
  - ◆ 输出过程：
    - CPU送地址选择某一输出设备；
    - CPU把数据放在数据总线上；
    - 输出设备取数据。

- **CPU与外设之间定时方式：与速度相关。**
  - ◆ **速度极慢或简单的外设：直接交换。**
  - ◆ **中速的外设：采用应答式交换，即异步定时方式。**
  - ◆ **高速外设：同步定时方式**

## ➤ 主机与外设信息交换方式

◆ 直接程序控制方式：**CPU**通过**I/O 指令**对**I/O**设备进行访问，主机与外设交换信息的每一过程均在程序中表示出来。

- **立即程序传送方式**：不询问外设状态，根据程序情况随时向外设传送数据。
- **程序查询方式**：根据外设的工作状态，在相应外设准备好时再向外设传送数据。
  - ✓ 优点：操作简单。
  - ✓ 缺点：**CPU**效率低。

## ➤ 主机与外设信息交换方式（续）

◆ 程序中中断方式：当有某些随机事件发生时**CPU**暂停执行当前的程序，转去执行引起中断的程序，处理完后再返回继续执行原程序。

- 优点：**CPU**效率高；

- 缺点：大批量传送速度慢。

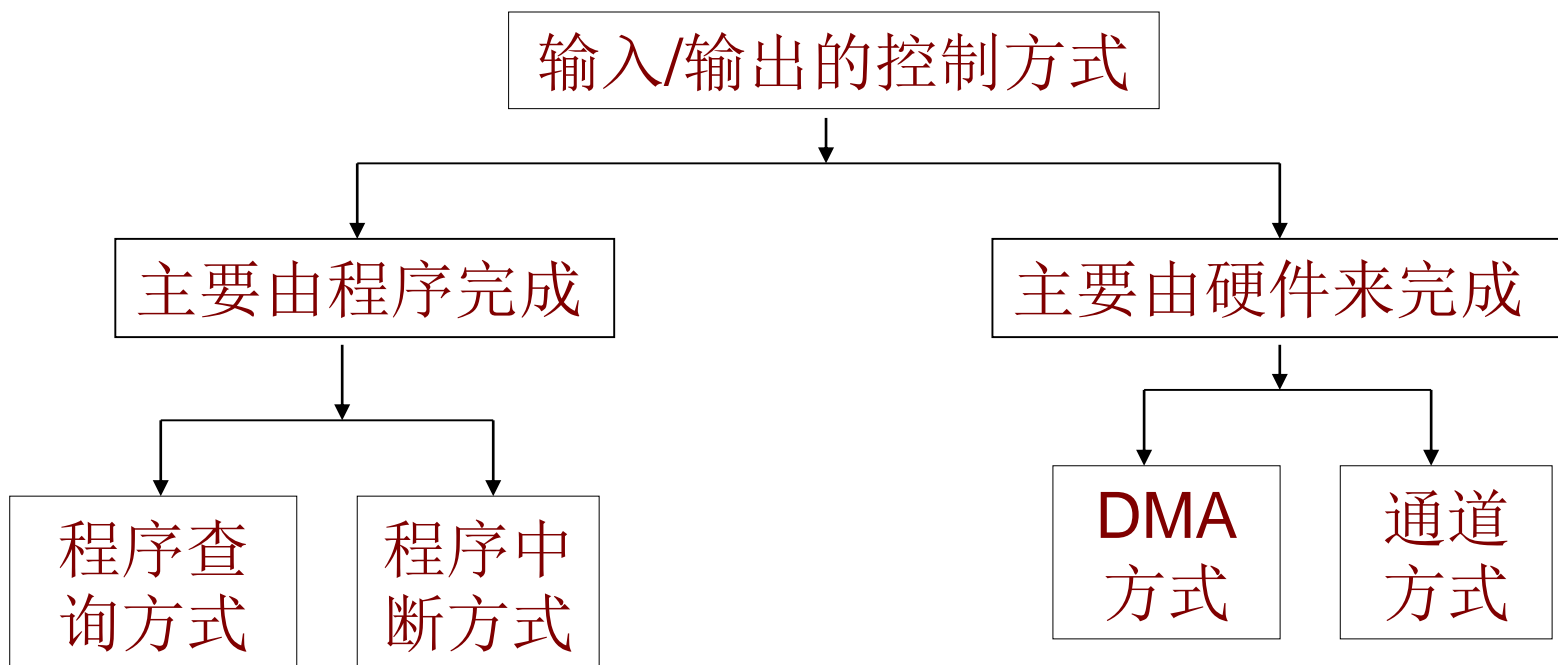
◆ 直接内存访问（**DMA**）方式：通过硬件控制总线，实现主存与**I/O**设备间的直接数据传送，在传送过程中无需**CPU**程序干预。

- 优点：**CPU**效率高，速度快，适合大批量数据传送；

- 缺点：增加硬件，成本较高。

## ➤ 主机与外设信息交换方式（续）

◆ 通道方式：通过执行通道（一种专用控制器）程序进行I/O操作的管理。



## 8.2 程序查询方式

- 特点：在**CPU**的主动控制下，通过执行程序完成**CPU**与外设之间的信息传送。

### 1、设备的编址方式

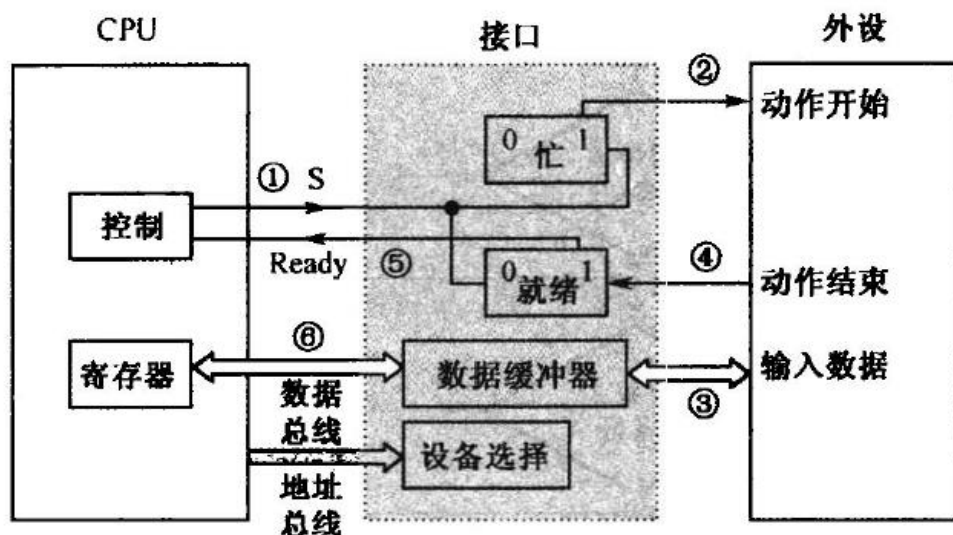
- 与存储器**统一**编址：在存储器总的地址空间中分出一个区域，作为**I/O**系统中的设备地址。
  - ◆ 优点：访问简单，操作灵活，不需专门的指令；
  - ◆ 缺点：占用部分存储器空间。
- 独立编址：内存与外围设备地址各自独立。
  - ◆ 优点：不占用存储器空间；
  - ◆ 缺点：需专门的指令。

### 2、输入/输出指令

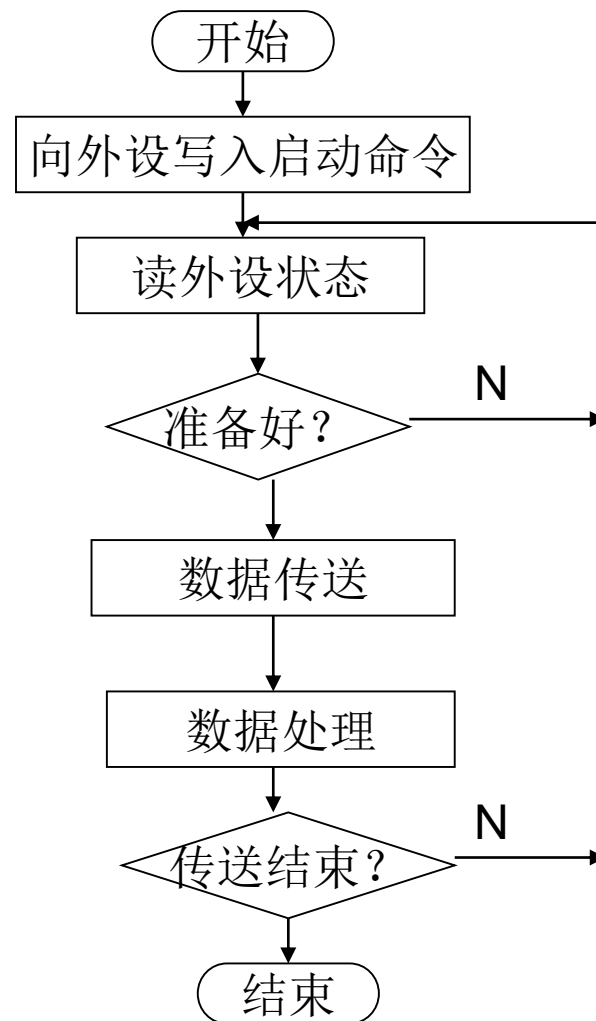
### 3、程序查询方式接口设计：

- 设备选择电路
- 数据缓冲寄存器
- 设备状态标志电路

### 4、程序查询方式流程



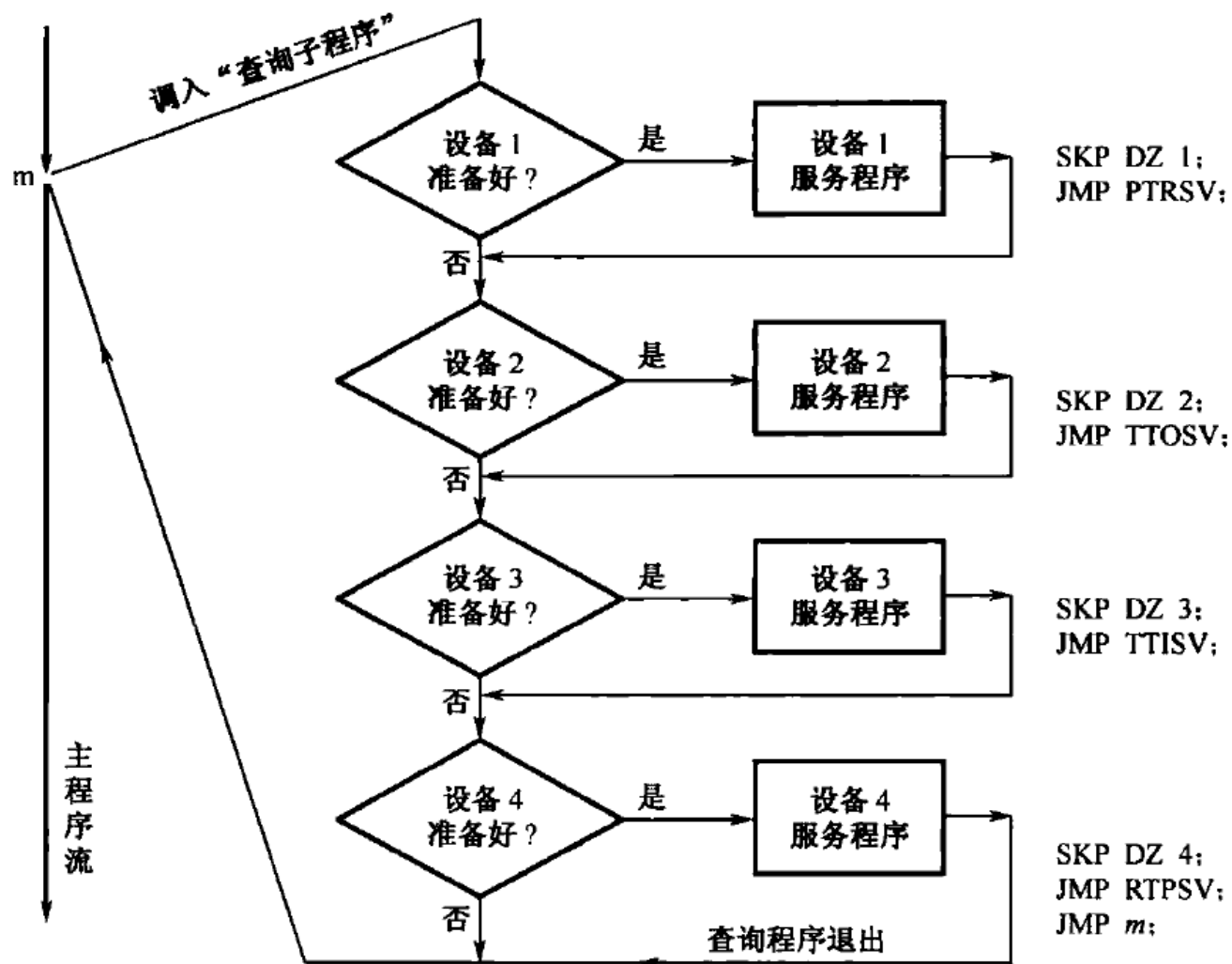
程序查询方式接口示意图



程序查询方式基本流程图



轮询时，一般  
先询问数据传  
输速率高的设  
备

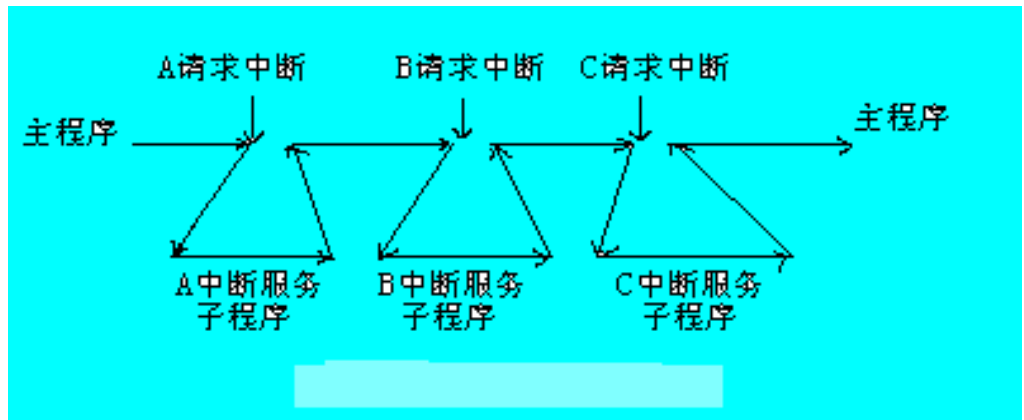


多个外围设备的程序查询方式流程图

# 8.3 程序的中断方式

## 8.3.1 中断概念

- 中断（程序中中断）：当有某些随机事件发生时CPU暂停执行当前的程序，转去执行引起中断的程序，处理完后再返回继续执行原程序。
- 中断特征：程序转换和随机性、排优性。
- 中断作用：
  - ◆ 实现CPU与外设并行工作；
  - ◆ 提高系统处理故障的能力，增强系统的可靠性；
  - ◆ 实现实时处理；
  - ◆ 系统调度；
  - ◆ 实现人机交换；
  - ◆ 实现多机通信。



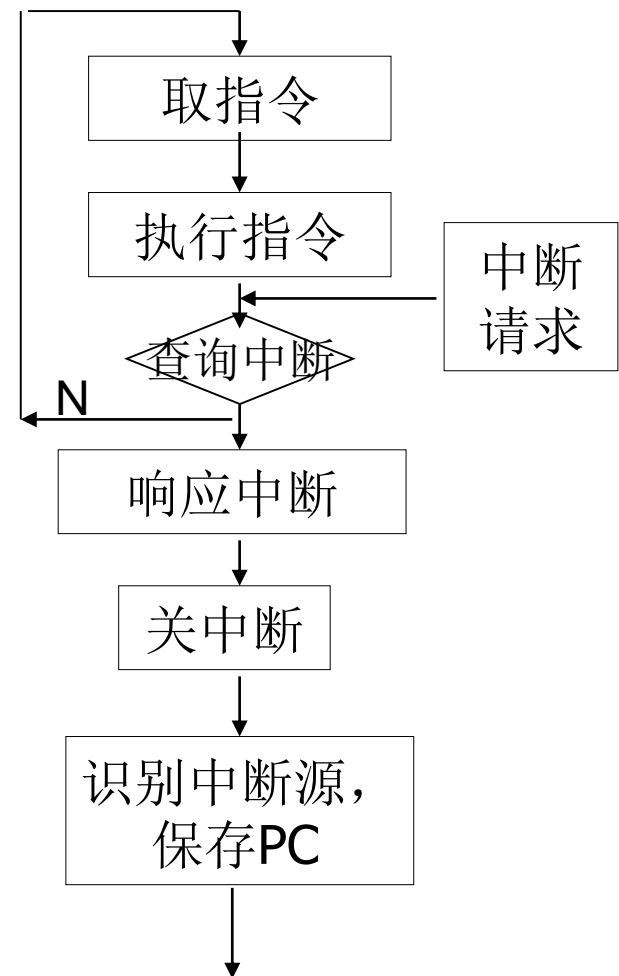
## ➤ 中断处理过程的流程

**1) 中断请求：** 中断源（分成可屏蔽中断和非屏蔽中断）请求CPU为自己服务的过程；

◆ **排优：** 当有几个中断同时请求时，按照优先排队顺序响应；

**2) 中断响应：（硬件完成）**

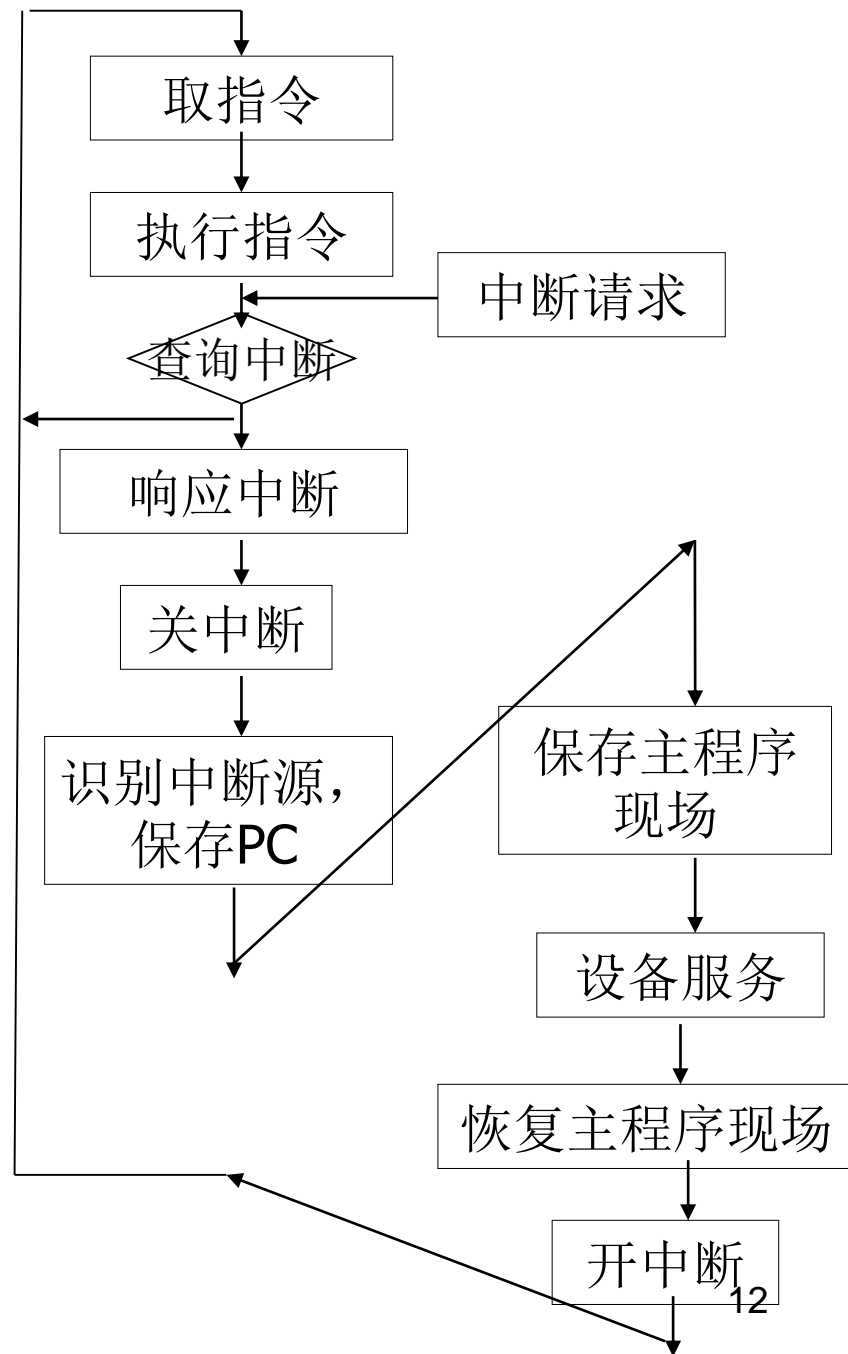
- CPU进入中断响应周期（INTA），如是非屏蔽中断马上响应；如是可屏蔽中断则在中断允许有效（IM=0）下响应中断；
- 保存程序计数器值和状态寄存器值，关中断允许标志（IM=1），读中断服务程序入口地址。



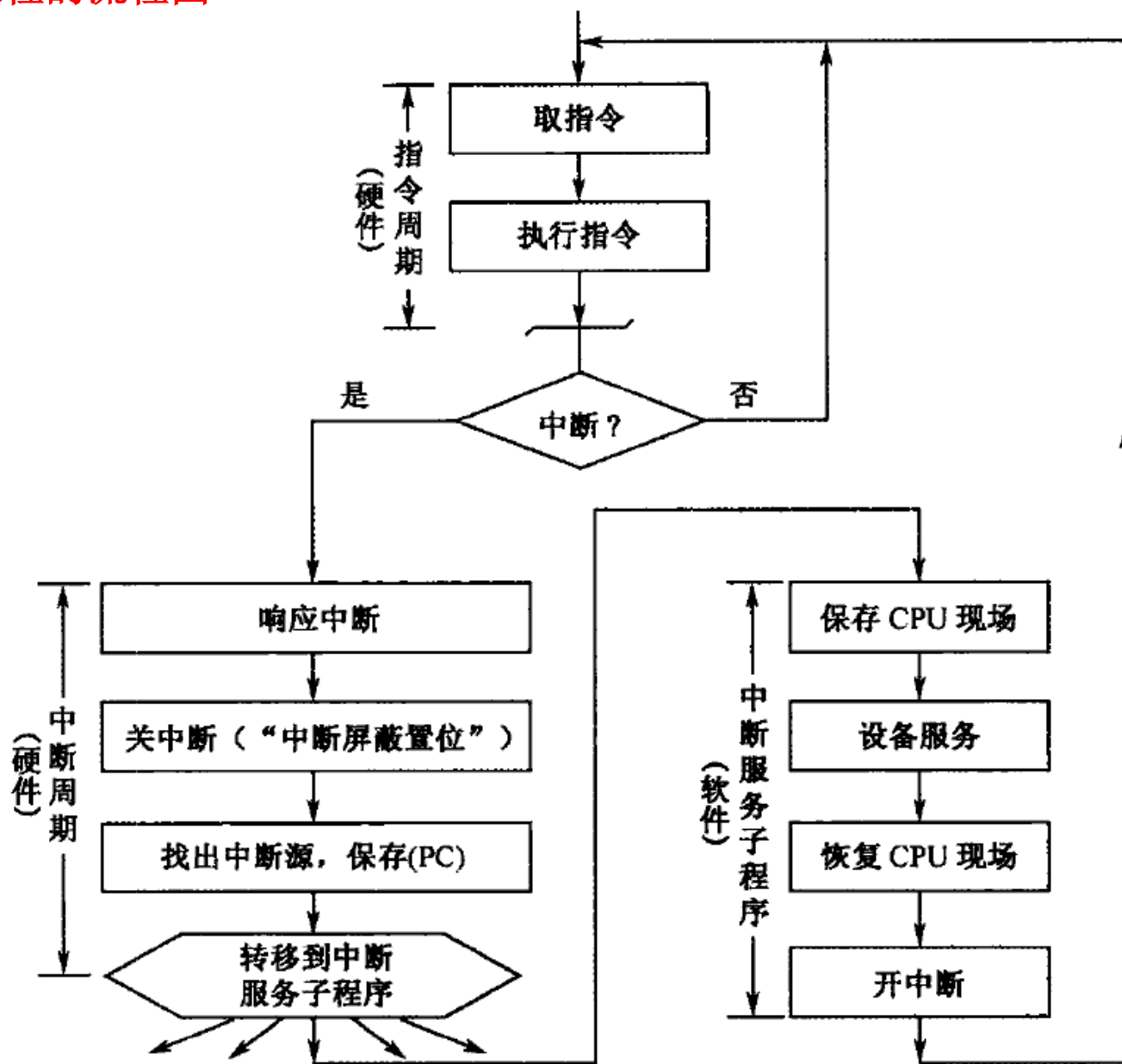
### 3) 中断服务（软件）

◆**处理中断**：（即运行中断服务子程序）保护现场，执行中断服务程序，恢复现场，开中断。

◆**中断返回**：恢复被中断程序处地址，继续执行原程序。



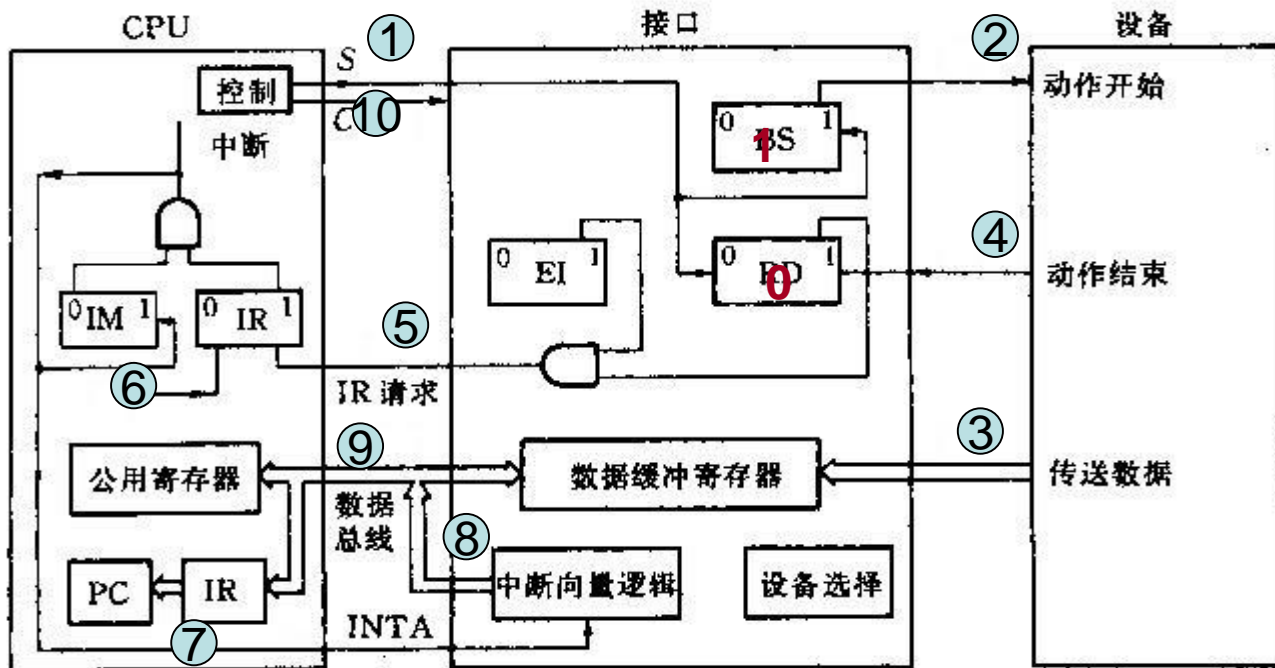
## 中断处理过程的流程图



## 8.3.2 程序中中断方式的基本接口电路

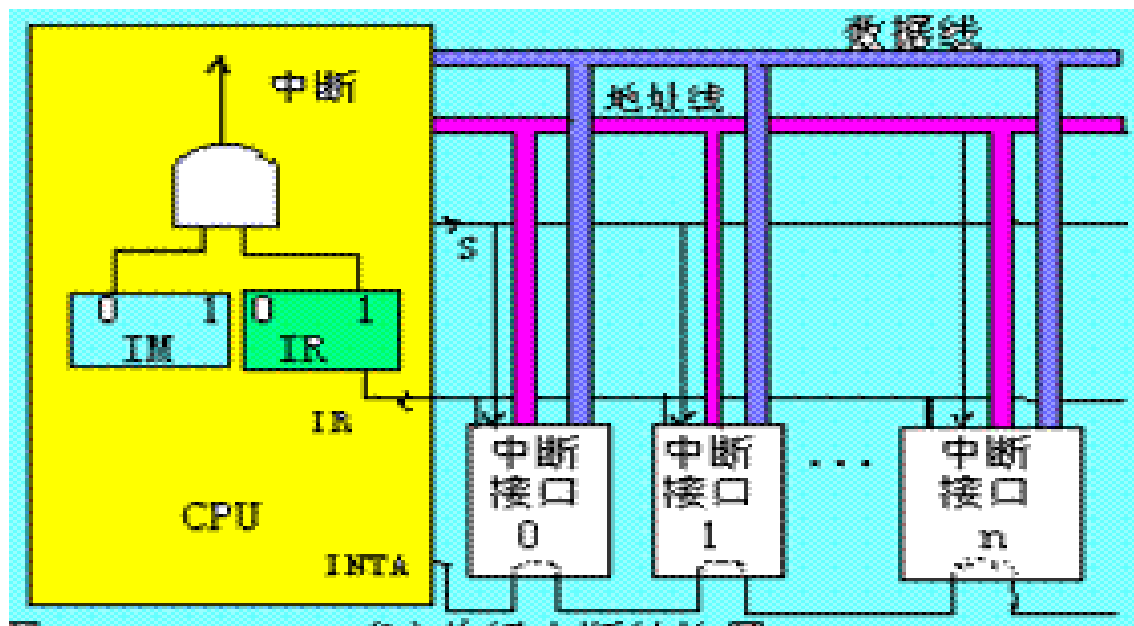
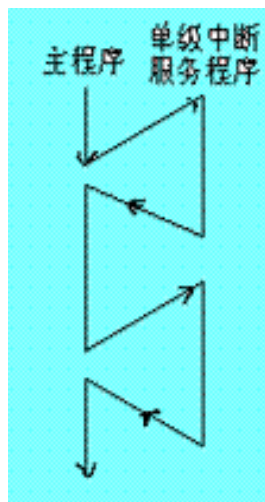
- **BS**: 启动接口工作**标志**触发器。
- **RD**: 外设准备就绪**标志**触发器。
- **EI**: 允许外设中断触发器。
- **中断向量逻辑**: 形成中断服务子程序的入口地址。
- **数据缓冲寄存器**: 暂存数据。
- **设备选择**: 地址译码逻辑。

设置EI标志的目的就是：通过软件来控制是否允许某设备发出中断请求



## 8.3.3 单级中断

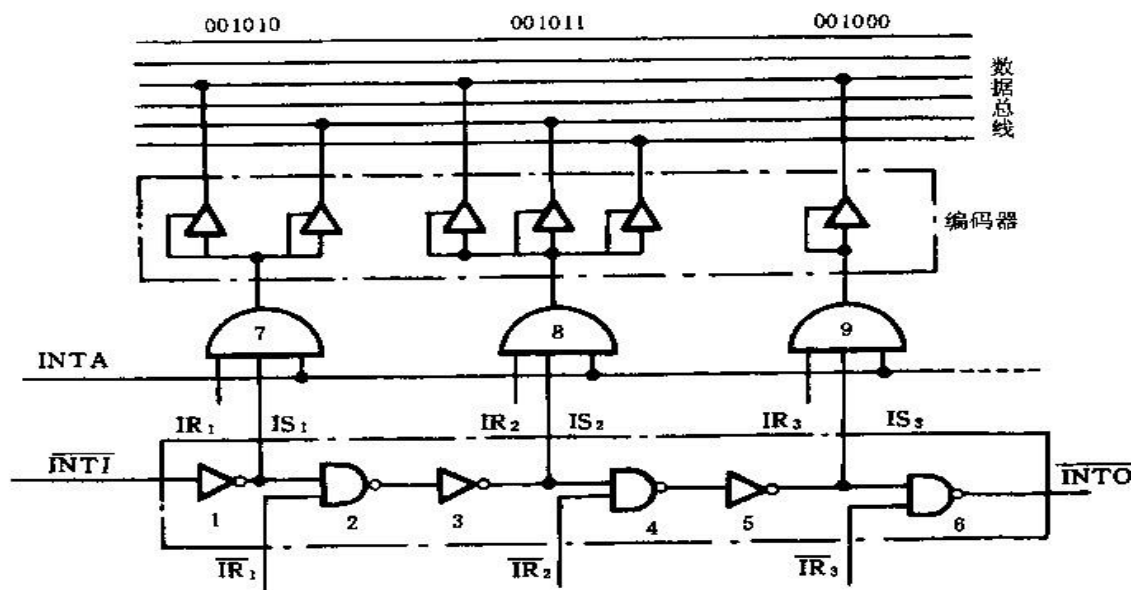
- 所有中断优先级相同，一旦响应一个中断后，只有服务完才可响应其他中断。
- 系统结构：公共请求线方式。
- 中断源识别方式：链式查询方式
  - ◆ 查询顺序决定优先级。



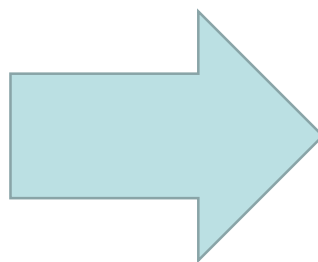
➤ 中断向量地址的产生及单级中断源识别电路：当CPU 响应中断时，由硬件直接产生一个固定的地址，即**向量地址**。由向量地址指出每个中断源设备的中断服务程序入口，此方法称为**向量中断**。

◆ 一级向量方式：地址码直接对应中断程序入口地址；

◆ 二级向量方式：地址码经转换后得到中断程序入口地址。



如何实现中  
断的嵌套？



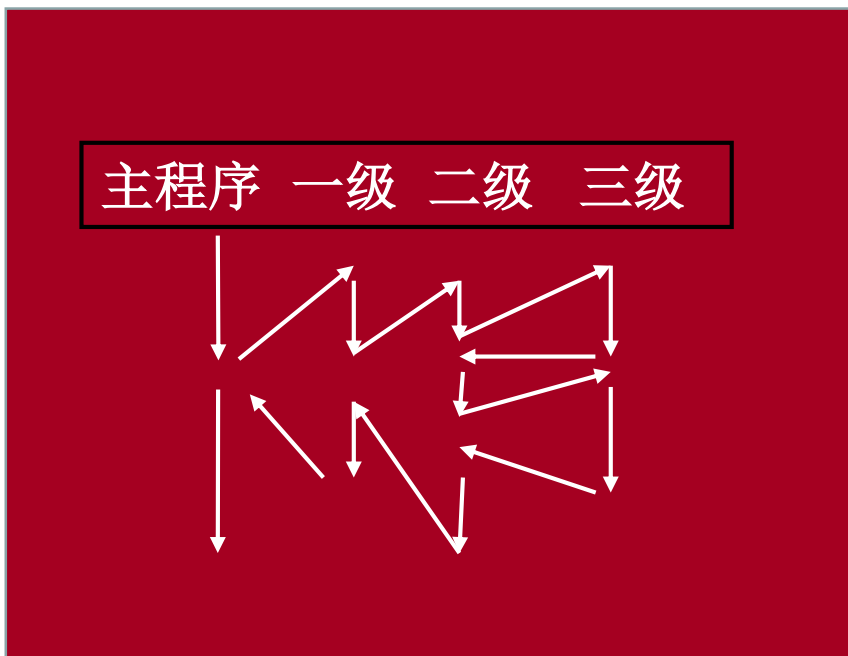
多级中断



## 8.3.4 多级中断

### 1、什么是多级中断

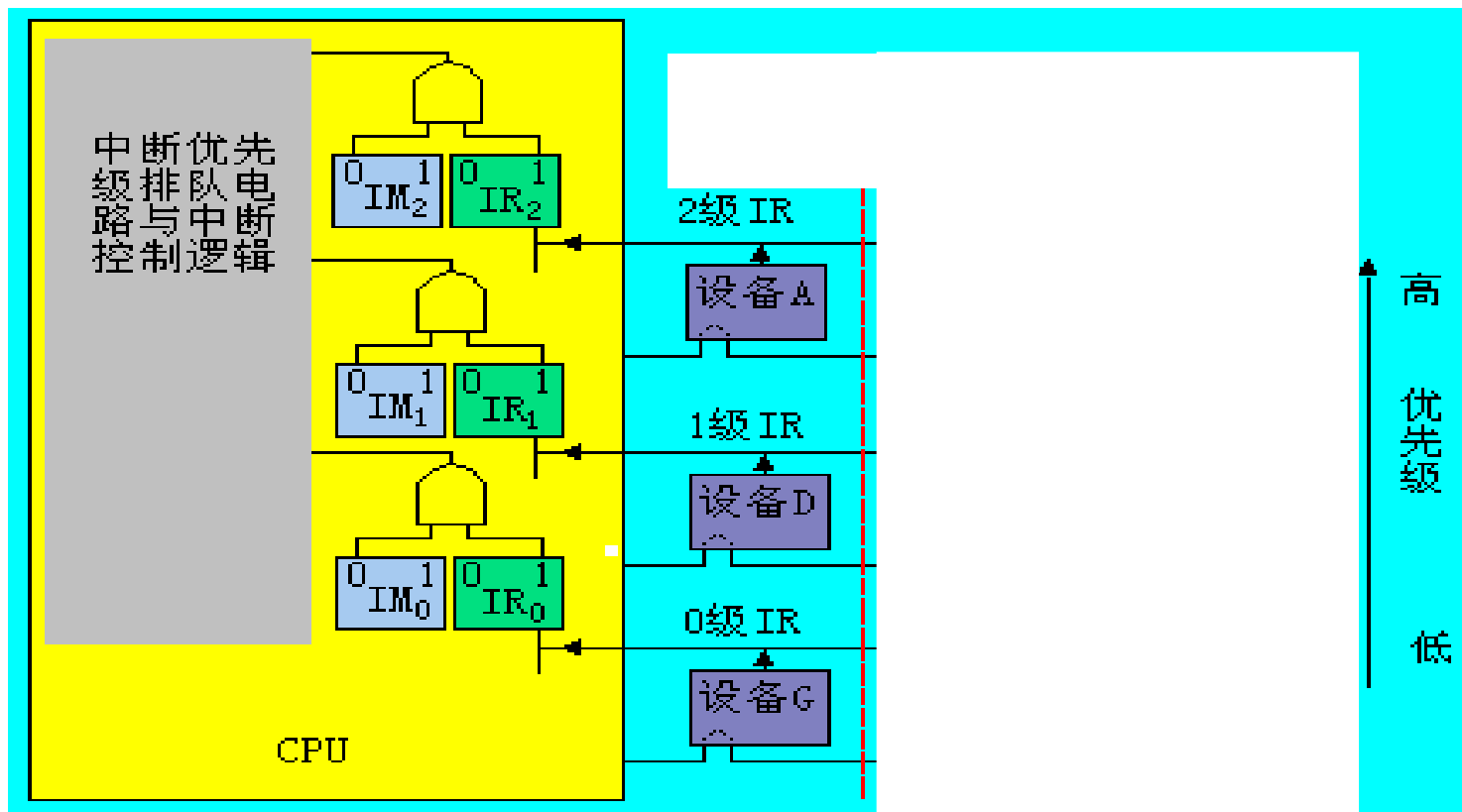
系统中的**多个中断源**具有不同的优先级。低优先级中断处理过程中可以被高优先级的中断源打断，实现中断嵌套。



**特点1:** 在多级中断之间可以实现中断嵌套，但在**同一级不可以嵌套**；

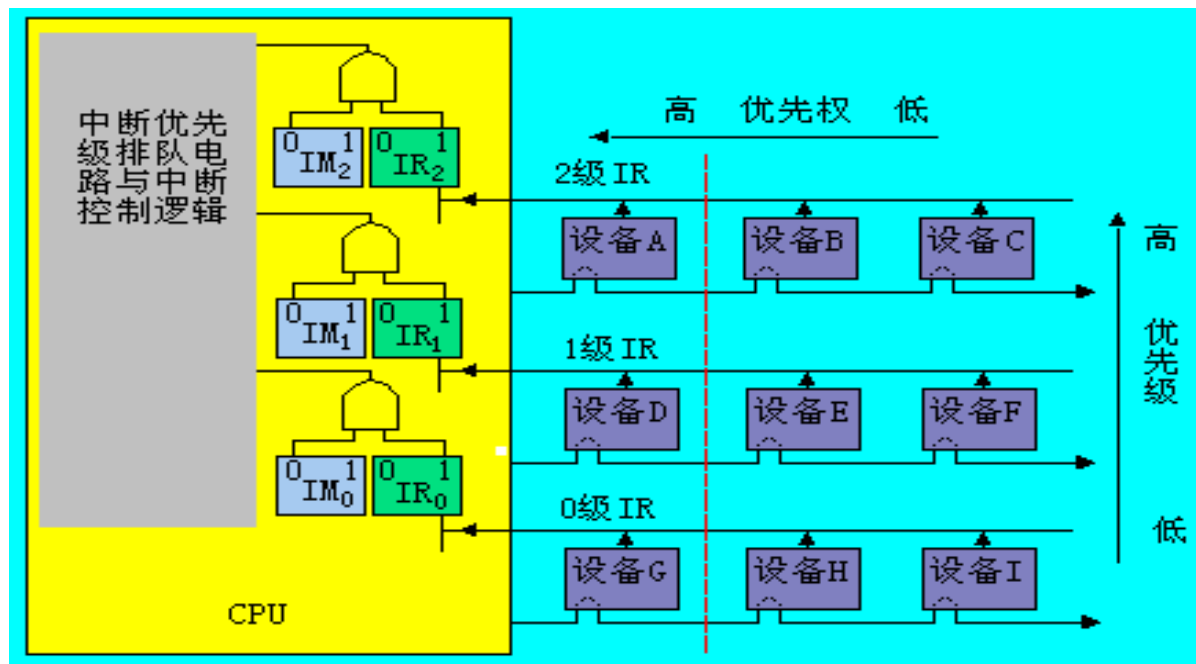
**特点2:** 由**中断堆栈**按顺序保护现场。

## 2、多级中断的类型



一维中断：同一个优先级里只有一个中断源。  
二维中断：同一个优先级里有多个中断源。

### 3、多级中断的请求与响应



**特点3:** 每级中断，CPU中都有请求触发器IR，及屏蔽触发器IM

**特点4:** 中断响应优先顺序由硬件设定。

- 当 $IR=1$ 且 $IM=0$ 时，该级中断请求进入排队电路等待响应
- 中断响应后，由硬件直接修改各级IM的值，保证相应次序将本级和优先级低于本级的IM设为1，将更高级的IM设为0

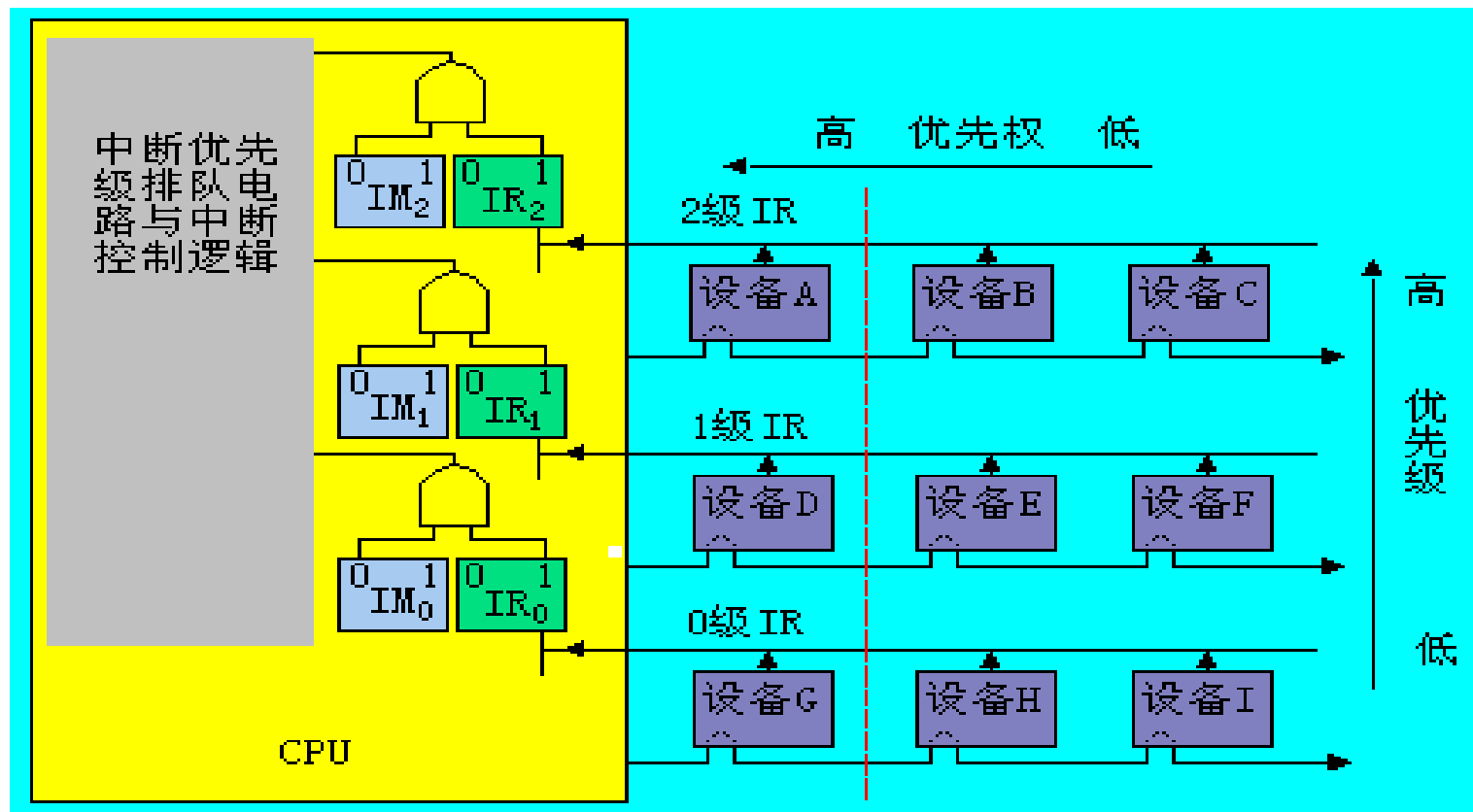


例1：在各设备同时提出中断请求的情况下，各个设备的优先级如何？

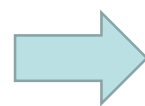
**A→B→C→D→E→F→G→H→I**

若CPU执行设备D的服务子程序，IM2，IM1，IM0的状态又是什么？

**IM2-0: 0 1 1**



**问题：中断响应顺序由硬件决定，能否根据需求调整中断处理的顺序？**



**中断服务子程序中对各级的IM进行修改**

**例2**：某计算机系统共有五级中断，其中断响应优先级从高到低为1→2→3→4→5。现按如下规定修改：各级中断处理时均屏蔽本级中断，且处理1级中断时屏蔽2、3、4和5级中断；处理2级中断时屏蔽3、4、5级中断；处理4级中断时不屏蔽其他级中断；处理3级中断时屏蔽4级和5级中断；处理5级中断时屏蔽4级中断。试问中断处理优先级(从高到低)顺序如何排列？并给出各级中断处理程序的中断屏蔽字？（设“0”为允许，“1”为禁止。）

**解：实际中断处理优先级(从高到低)顺序应为**

**1 → 2 → 3 → 5 → 4。**

**1级中断屏蔽字为 11111；**

**2级中断屏蔽字为 01111；**

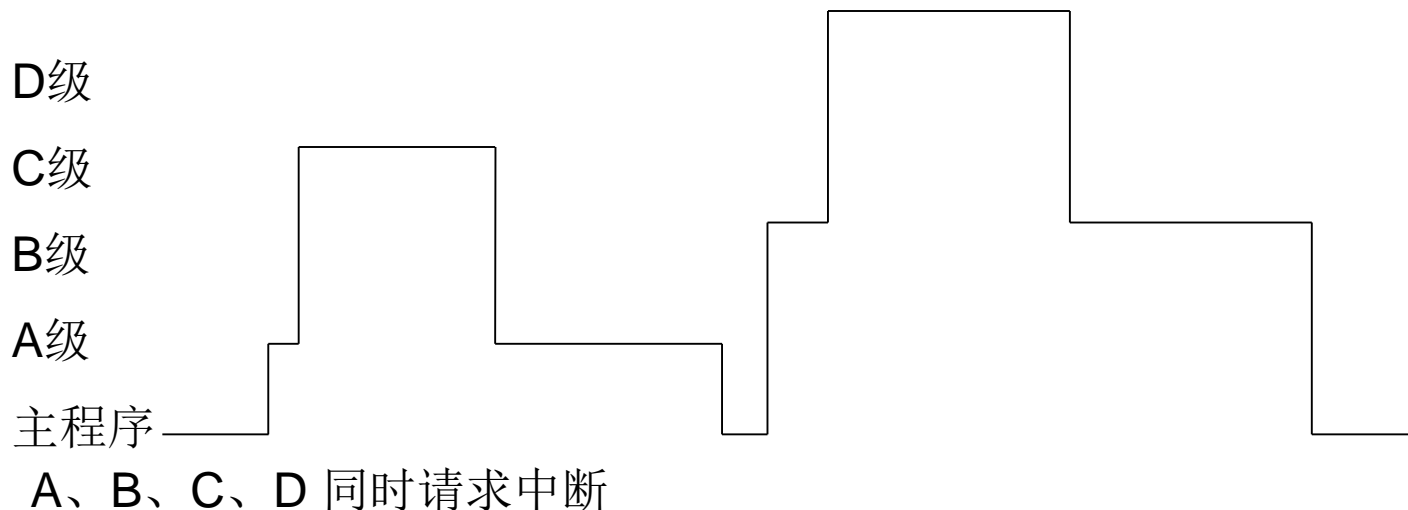
**3级中断屏蔽字为 00111；**

**4级中断屏蔽字为 00010；**

**5级中断屏蔽字为 00011。**

**例3** 某计算机有四级中断，优先级从高到低为A〉B〉C〉D。假定各级中断程序的屏蔽位设置为：A：1101；B：0100；C：1111；D：0101。请给出中断处理次序。设A、B、C、D同时请求中断，试画出CPU 执行程序的轨迹。

解：中断处理次序：C〉A〉D〉B。

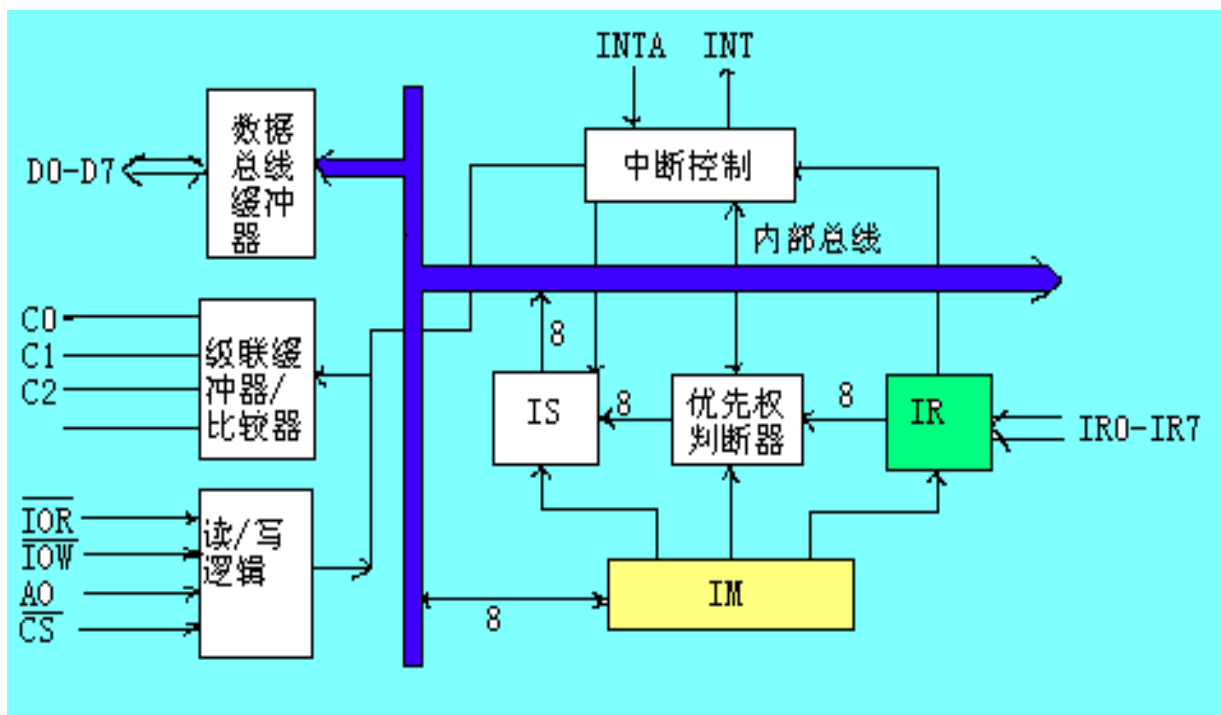


# 多级中断小结

- ❑ 每级中断，**CPU**中都有对应的**IR**和**IM**。
- ❑ 在多级中断之间可以实现中断嵌套，但在同一级不可以，由中断堆栈按顺序保护现场。
- ❑ 中断响应优先顺序由硬件设定，中断处理（或中断执行）的完成顺序可以由软件设定

## 8.3.5 中断控制器

- 微机系统中，常将其中可公用的中断控制逻辑从I/O接口中分离出来，集成芯片来实现向量产生、排优电路、中断屏蔽等中断控制逻辑。
- 例8259：可编程中断控制器
  - ◆ 8259可管理8路中断请求IR0~IR7，通过级联方式，最多可扩展为64级中断。具有多种工作模式。

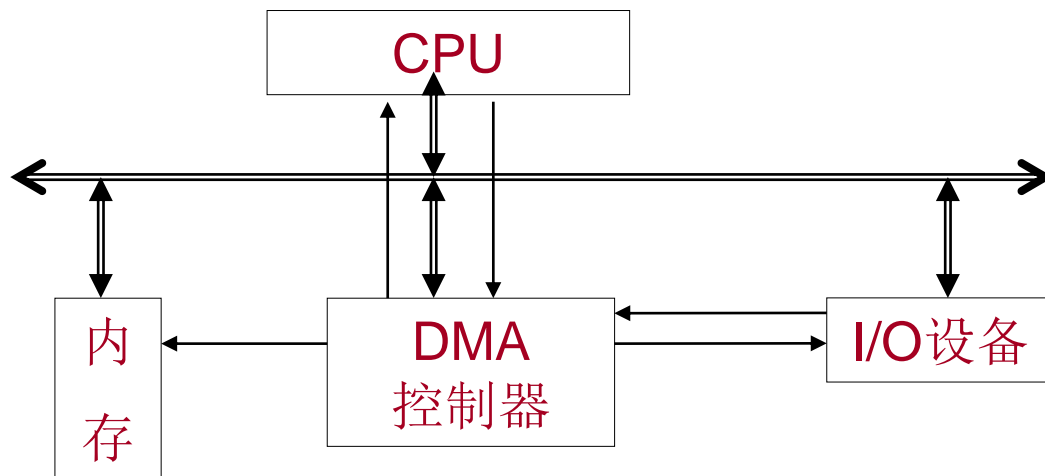




## 8.4 DMA方式

### 1 概念

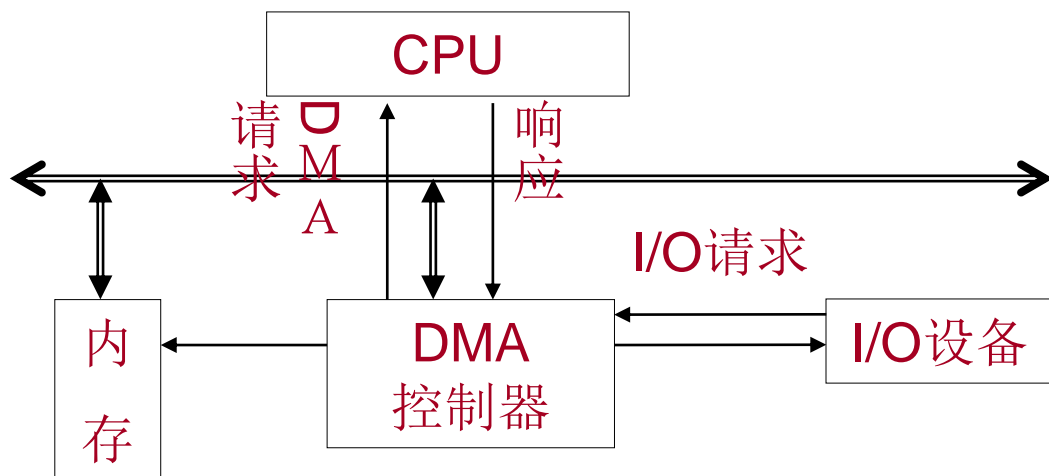
- 特点：通过硬件控制总线实现主存与I/O设备间的直接数据传送，在传送过程中无需CPU程序干预。



DMA工作方式原理图

## ➤ DMA的基本工作处理过程

- ◆**DMA请求**：外设通过接口向 **CPU** 发**DMA**请求信号。
- ◆**DMA响应**：**CPU**将工作改为**DMA**操作方式，将总线控制权交给**DMA**控制器。
- ◆**DMA数据传送**：**DMA**控制器发总线信号，在主存和**I/O** 寄存器 之间传送数据。
- ◆**结束处理**：数据传送完后，发中断请求，通知**CPU**进行后处理。



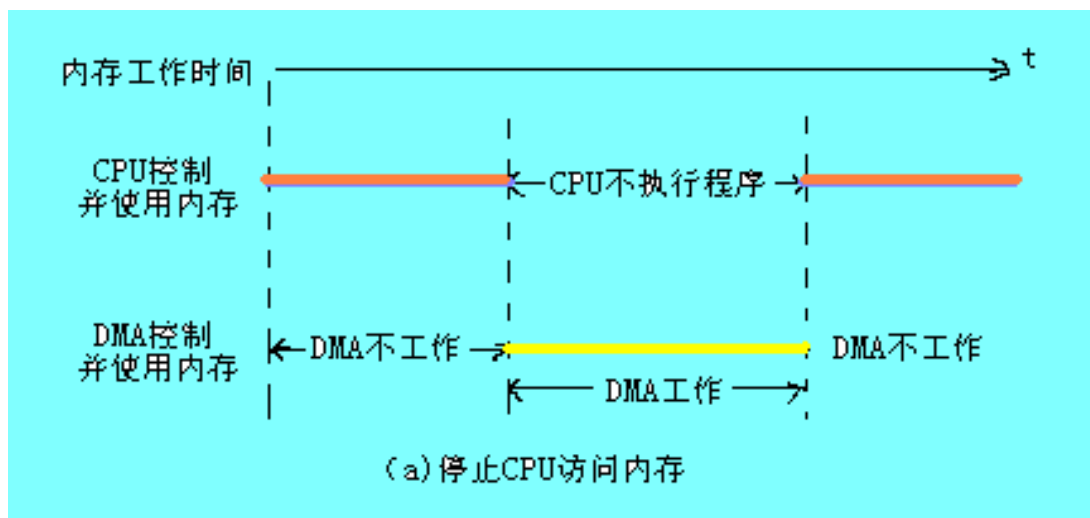
DMA工作方式原理图

## 2 DMA传送方式

### ➤ CPU与DMA控制器控制内存方式

#### ◆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: 内存的效能没有充分发挥。



## 2 DMA传送方式

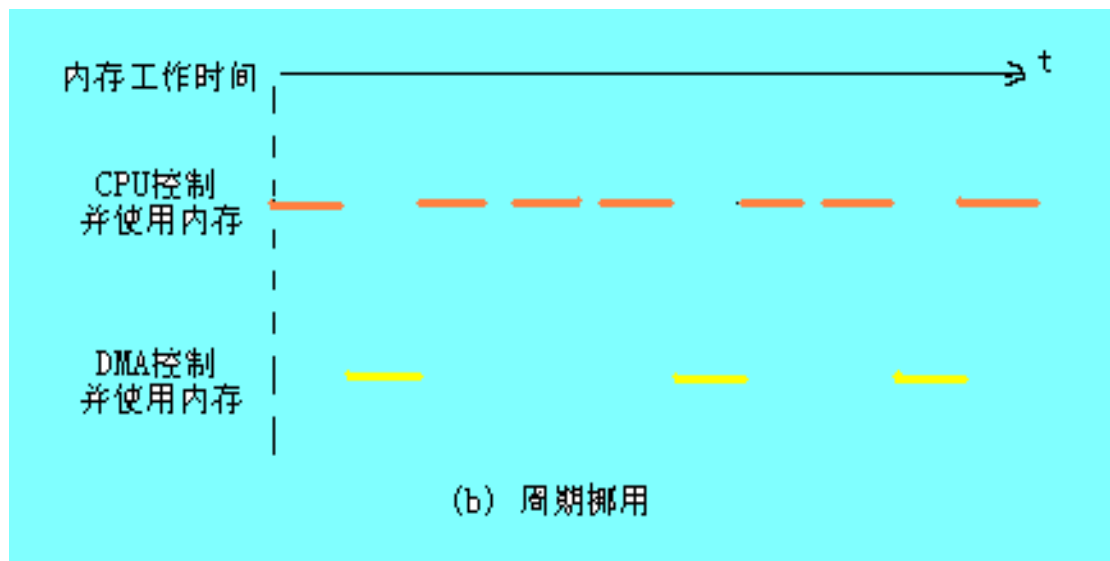
### ➤ CPU与DMA控制器控制内存方式

◆ **周期挪用**：指在**CPU**执行访内指令的过程中插入**DMA**请求，挪用了一二个内存周期。

● **优点**：可以提高**CPU**和内存的效率，适用于I/O设备读写周期大于内存存储周期的情况

● **缺点**：申请频繁，速度较慢。

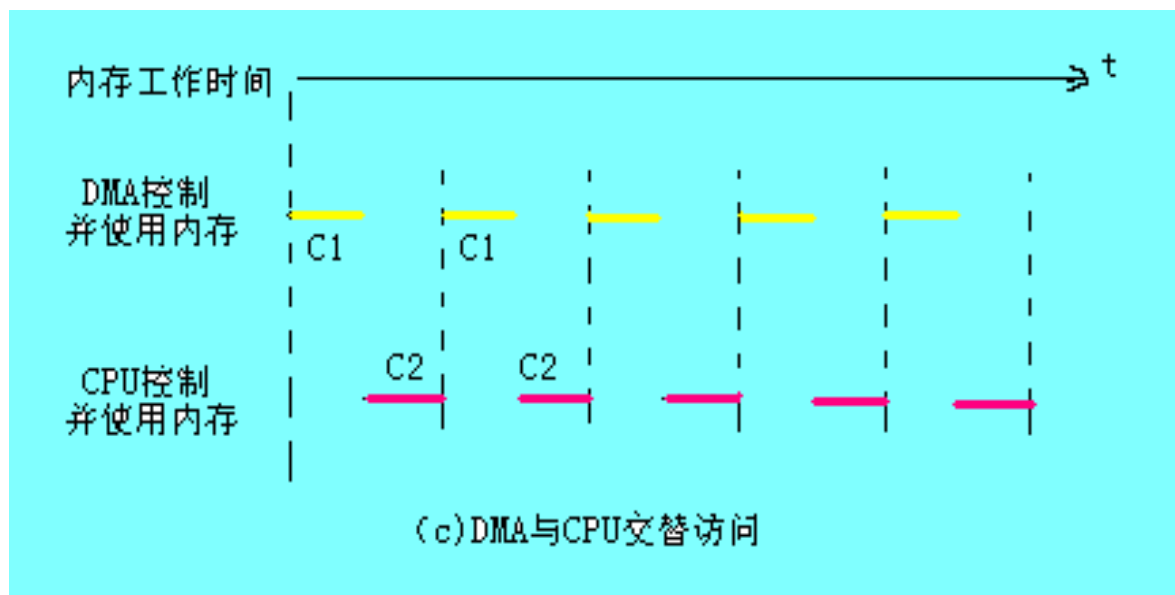
每次周期挪用都有申请总线控制权、建立总线控制权和归还总线控制权



## ➤ CPU与DMA控制器控制内存方式

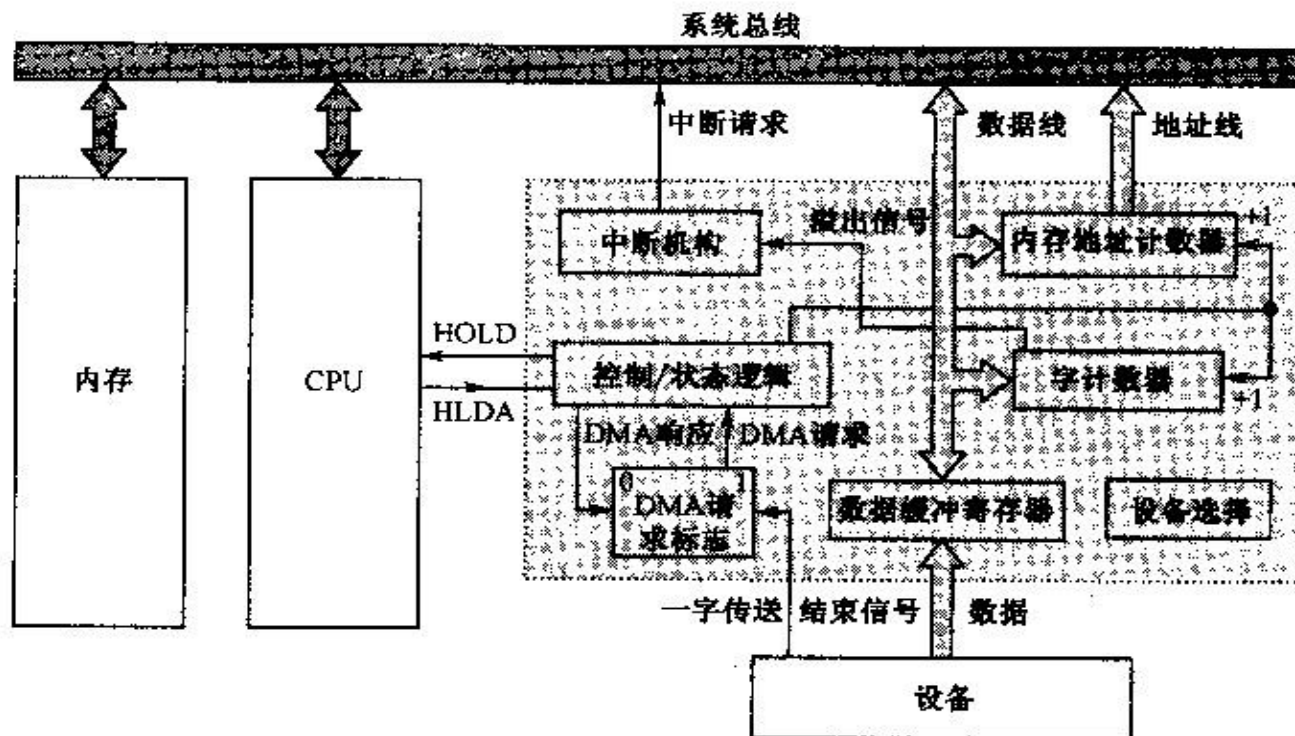
◆ **DMA与CPU交替访内**：DMA与CPU分时交替访问内存。

- **优点**：DMA不需申请，适用于CPU工作周期比内存周期长的系统。
- **缺点**：设计复杂。



### 3 基本的DMA控制器结构:

- ◆内存地址计数器: 用于存放内存中要交换的数据的地址。
- ◆字计数器: 用于记录传送数据块的长度(多少字数)。
- ◆DMA请求标志: 产生DMA请求信号控制逻辑;
- ◆控制状态逻辑: 由控制和时序电路、状态标志等组成;
- ◆中断机构: 数据传送完后产生中断请求信号。



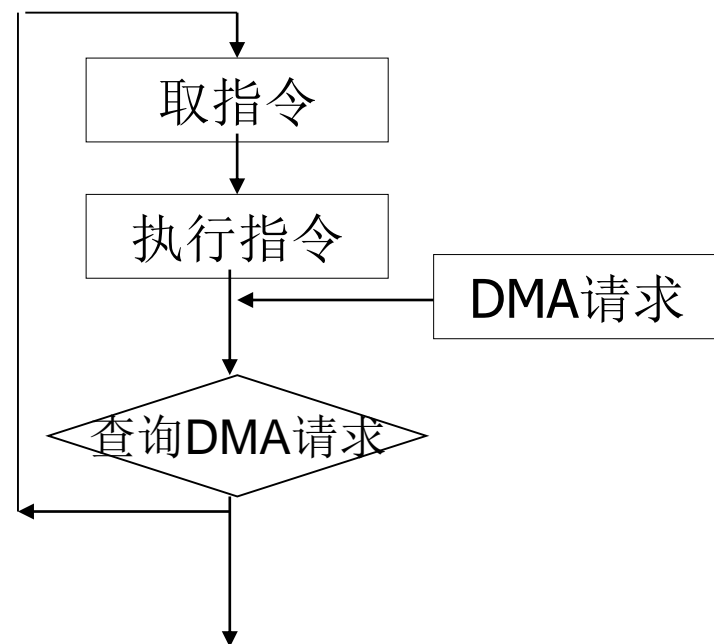
数据传送过程：传送前预处理、  
正式传送、传送后处理。

➤ 设：采用停止CPU访问方式

- ◆ CPU初始化DMA控制器，选择设备。

- ◆ 外设发DMA请求

- ◆ 若该外设未被屏蔽，且无更高优先级DMA通道正在工作，则CPU结束当前正在进行的基本操作后，释放总线的控制权，向DMA控制器发一个总线响应信号；



➤ **数据传送过程：** 传送前预处理、正式传送、传送后处理。

➤ **设：**

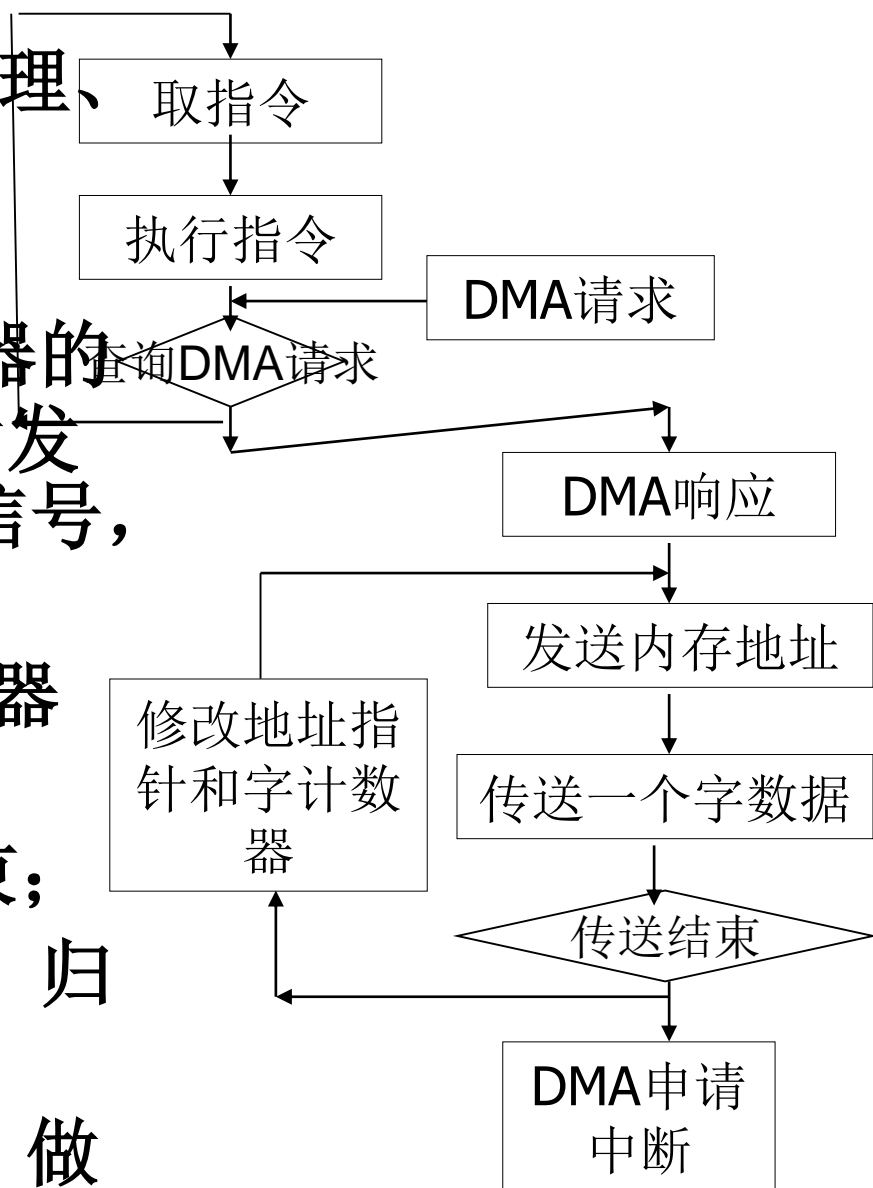
◆ **DMA控制器将地址寄存器的内容发往地址总线，同时发I/O读和存储器写等控制信号，以传送一个字数据。**

◆ **地址寄存器加1，字计数器加1；**

◆ **重复上述传送过程至结束；**

◆ **DMA结束，发中断请求，归还总线控制权。**

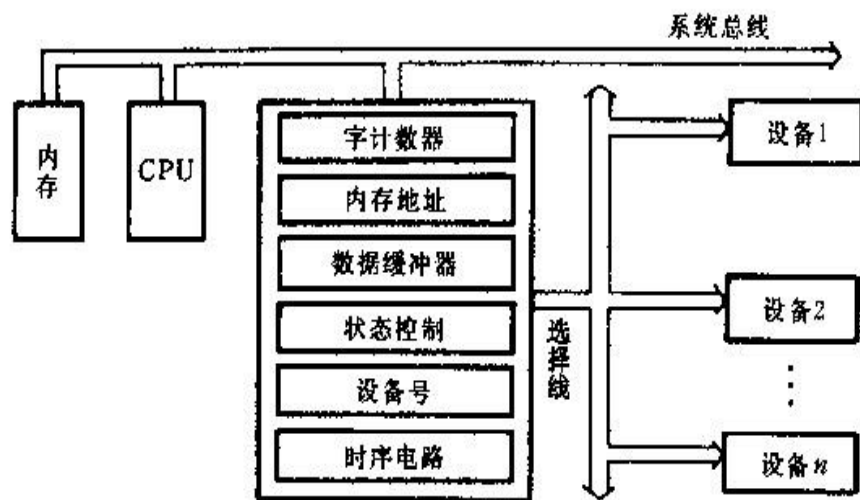
◆ **CPU执行中断服务程序，做后处理工作。**



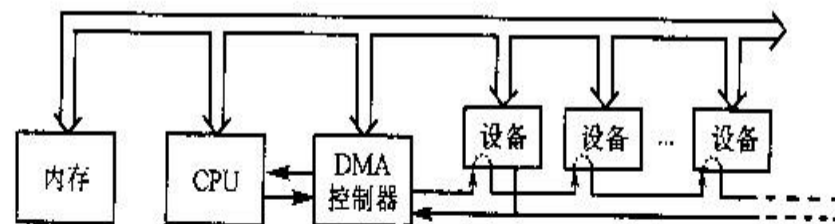


## 4 选择型和多路型DMA控制器

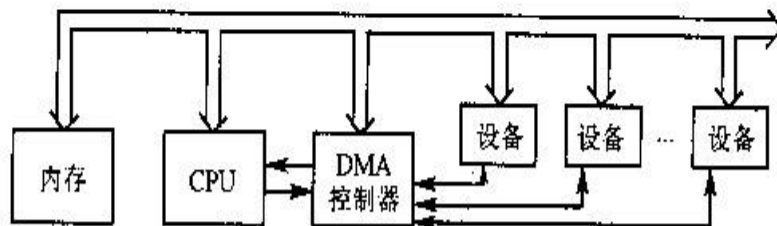
- ◆ **选择型DMA控制器**：DMA控制逻辑与I/O接口合并为一个整体，并可通过设备选择逻辑控制多台I/O设备，但逻辑上只允许连接一个设备；
- ◆ **多路型DMA控制器**：DMA控制逻辑公用部分与I/O接口分离，形成专用DMA控制器。多路型可同时控制多个设备工作。当多个外设同时发DMA请求时，根据判优逻辑响应优先级高的请求。



选择型

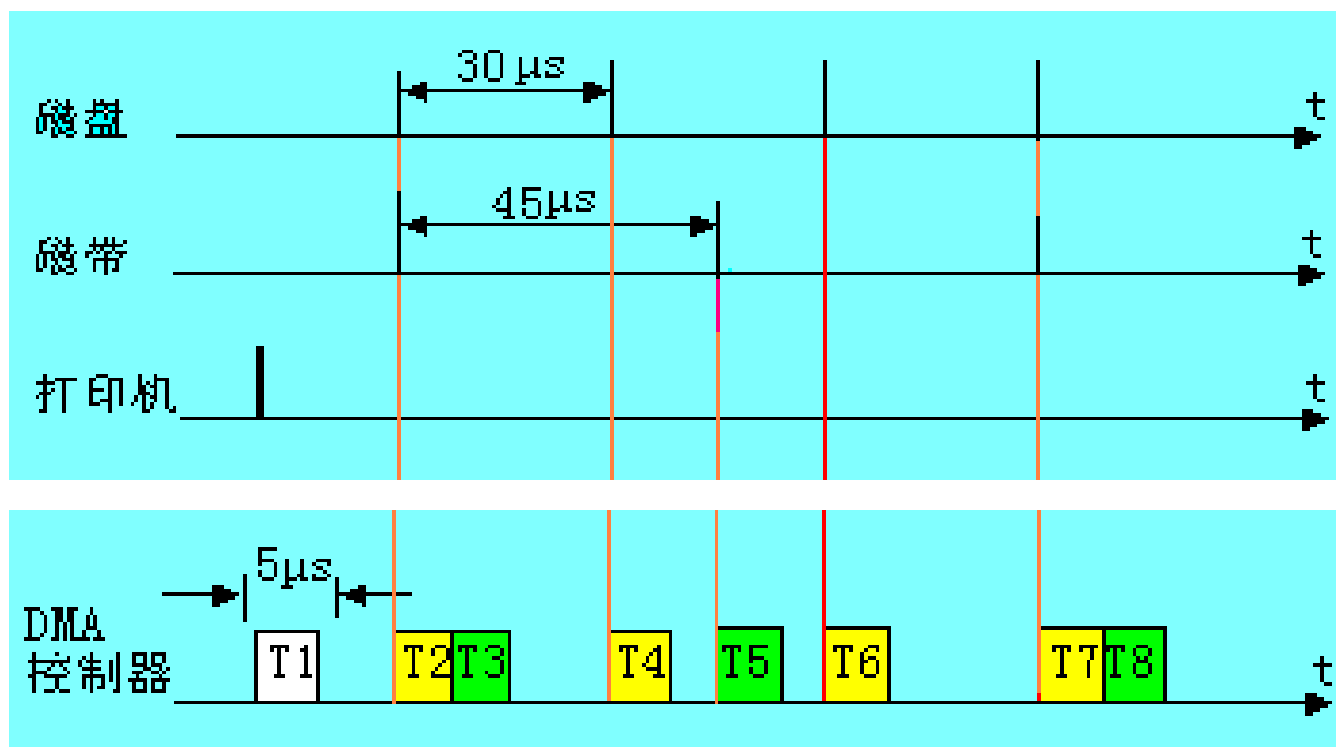


(a) 链式多路型DMA



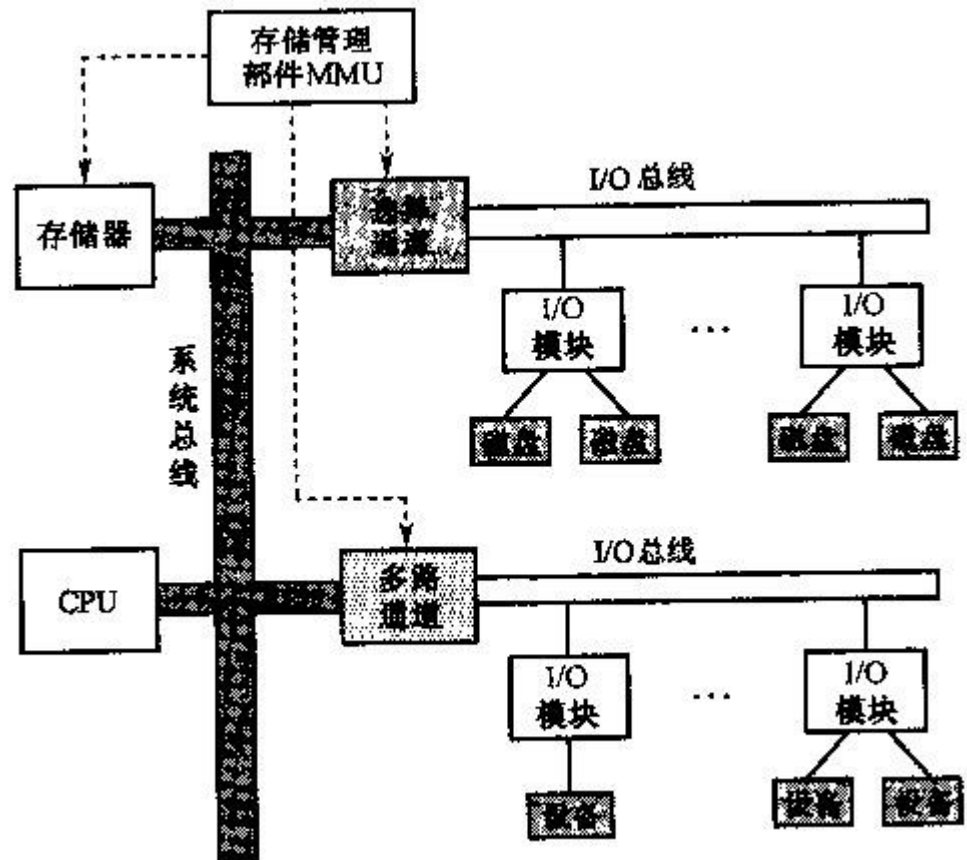
(b) 独立请求多路型DMA

- 【例5】图中假设有磁盘、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。根据传输速率，磁盘优先权最高，磁带次之，打印机最低，图中假设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。



## 8.5 通道方式

- 通道：一种专用**处理器**，通过执行通道程序进行**I/O**操作的管理，为主机与**I/O**设备提供数据传送通道。



## ➤ 通道的功能:

- ◆ 接收**CPU**的**I/O**指令，按指令要求与指定的外围设备进行通信；
- ◆ 从内存中或自己的局部存储器中取通道指令，经译码后向**I/O**发命令；
- ◆ 组织外设与内存之间进行数据传送，并根据需要提供数据传送的缓存空间，提供数据存入内存的地址和传送的数据量；
- ◆ 获取外设的状态信息，并将其存入内存，供**CPU**使用；
- ◆ 通过中断方式报告**I/O**设备及其本身的工作情况。

## ➤ **CPU**对通道的管理

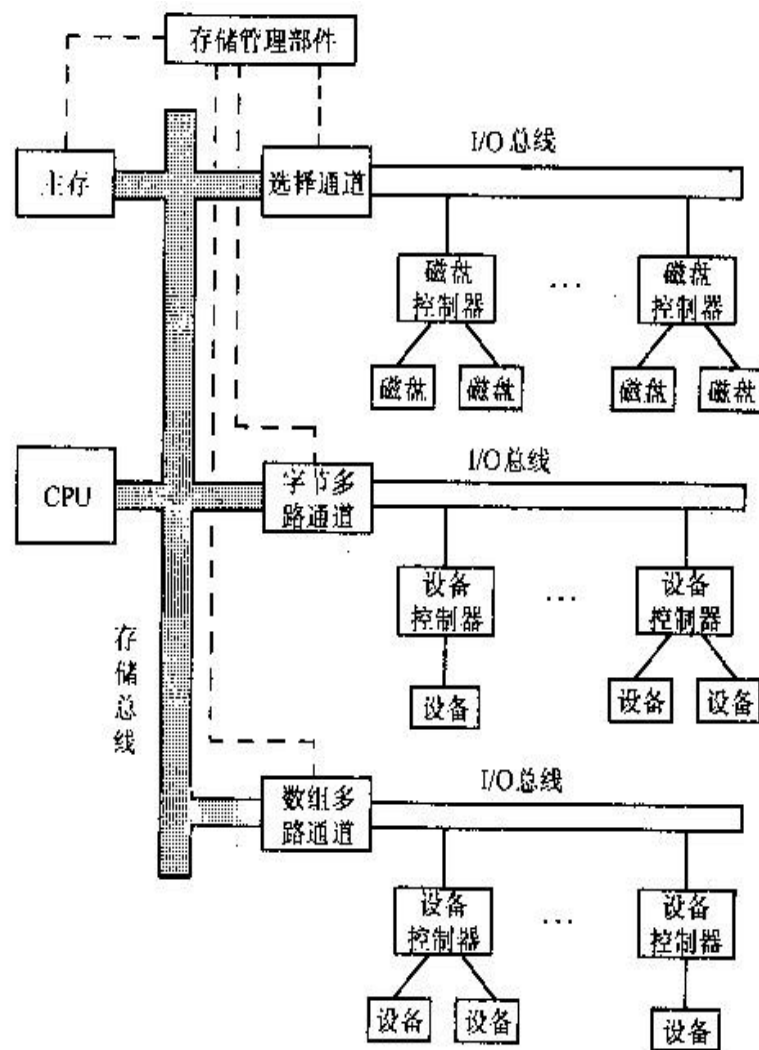
- ◆ 执行**I/O**指令
- ◆ 处理来自通道中断。

## ➤ 通道对设备的管理

- ◆ 通道指令

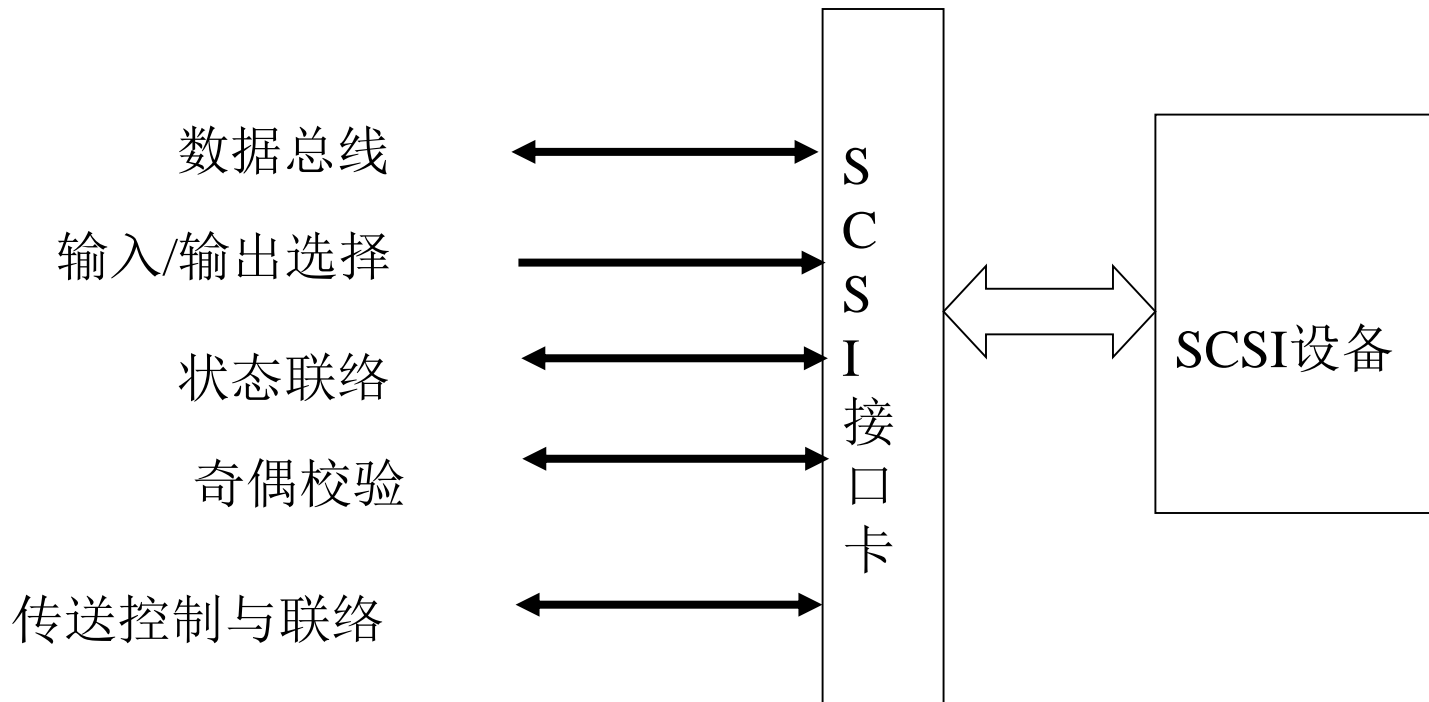
## ➤ 通道的种类:

- 选择通道: 当连接多台外设时, 一次只选择一个外设, 成组传送数据完成后才可选择其他外设, 用于连接多台高速外设;
- 数组多路通道: 当某设备进行数据传输时, 通道只为该设备服务; 当设备在进行寻址等控制性操作时, 通道暂时断开与该设备连接, 挂起该设备的通道程序, 去为其他设备服务;
- 字节多路通道: 以字节为单位采用分时交叉方式传送多台外设的数据; 用于连接多台低速外设;



## 8.6 通用IO标准接口

- **SCSI标准**：小型计算机系统接口，系统级接口，处于主适配器和智能设备控制器之间的并行接口，最高速率**40MB/s**，常用于服务器，一般**SCSI**接口与设备是分离的。分成**SCSI-1**，**SCSI-2**，**SCSI-3**，串行**SCSI**。



SCSI接口模型

- **IEEE1394**: 高速串行接口，最高速率**400Mb/s**，是一种外部串行总线标准，常用于新型高速硬盘和多媒体数据传送。
- **USB**: 通用串行总线接口，支持热插拔，即插即用的优点，分成**USB1.0**，**USB2.0**，最高速率**480Mb/s**。