



# 第8章 输入输出系统



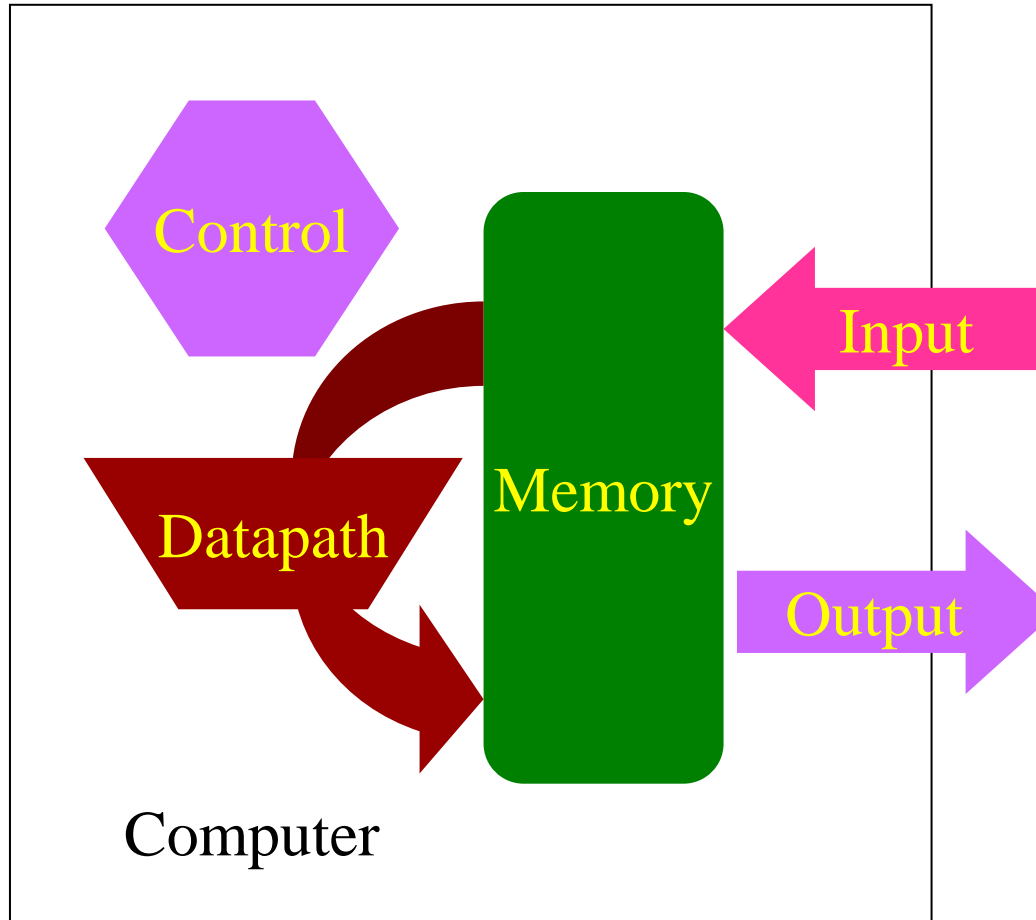


# 8.1 外围设备的定时方式和 信息交换方式





# 计算机组成



- **Datapath:** 完成算术和逻辑运算，通常包括其中的寄存器。
- **Control:** CPU的组成部分，它根据程序指令来指挥datapath, memory以及I/O运行，共同完成程序功能。
- **Memory:** 存放运行时程序及其所需要的数据的场所。
- **Input:** 信息进入计算机的设备，如键盘、鼠标等。
- **Output:** 将计算结果展示给用户的设备，如显示器、磁盘、打印机、喇叭等。





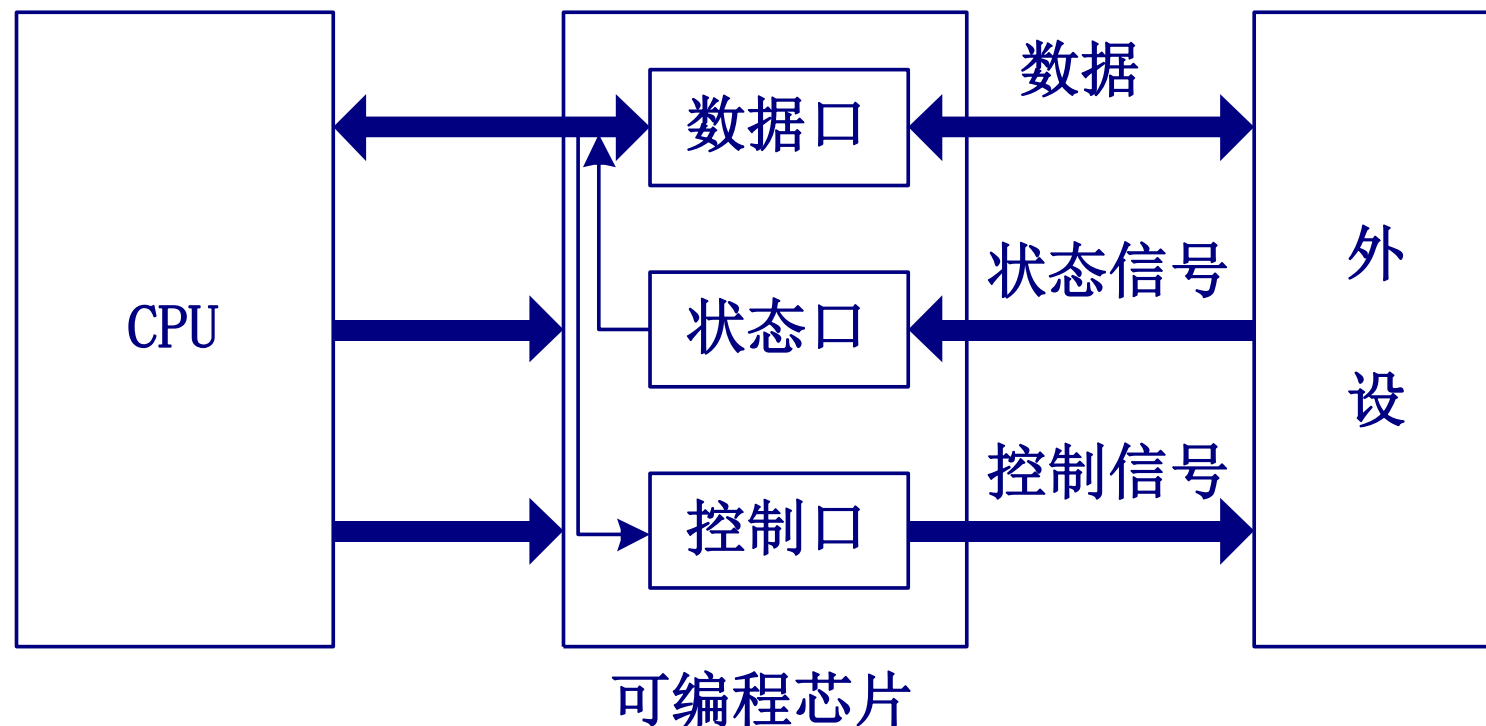
# 输入输出系统的组成

- I/O设备(外围设备)
- I/O接口
- I/O设备控制器
- I/O管理软件（驱动程序）





# 输入/输出操作的两个传输阶段



1. CPU与I/O接口之间的数据传送
2. I/O接口与外设间的数据传送





# CPU与外设之间的定时

- 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在时间上同步呢？
  - ◆ 解决办法时在CPU和外设之间数据传送时选择适当的定时方式





# CPU与外设之间的定时种类

- 速度极慢或简单的外设
  - ◆ CPU只需要接收或者发送数据即可，即无条件传送方式
- 慢速或者中速的设备
  - ◆ 可以采用异步传送（即：应答）方式
- 高速外设
  - ◆ 采用同步传送方式





# CPU与外设之间的数据传送式

- 程序查询（Polling）方式
- 程序中断方式
- 直接存储器存取（**DMA: Direct Memory Access**）方式
- 通道方式







# 1. 程序查询方式

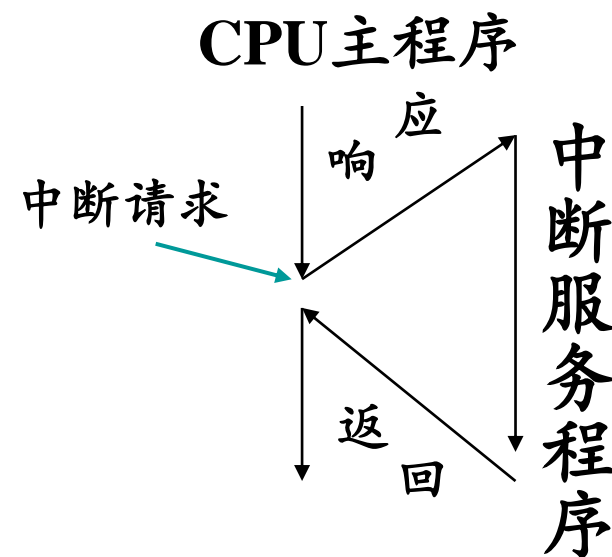
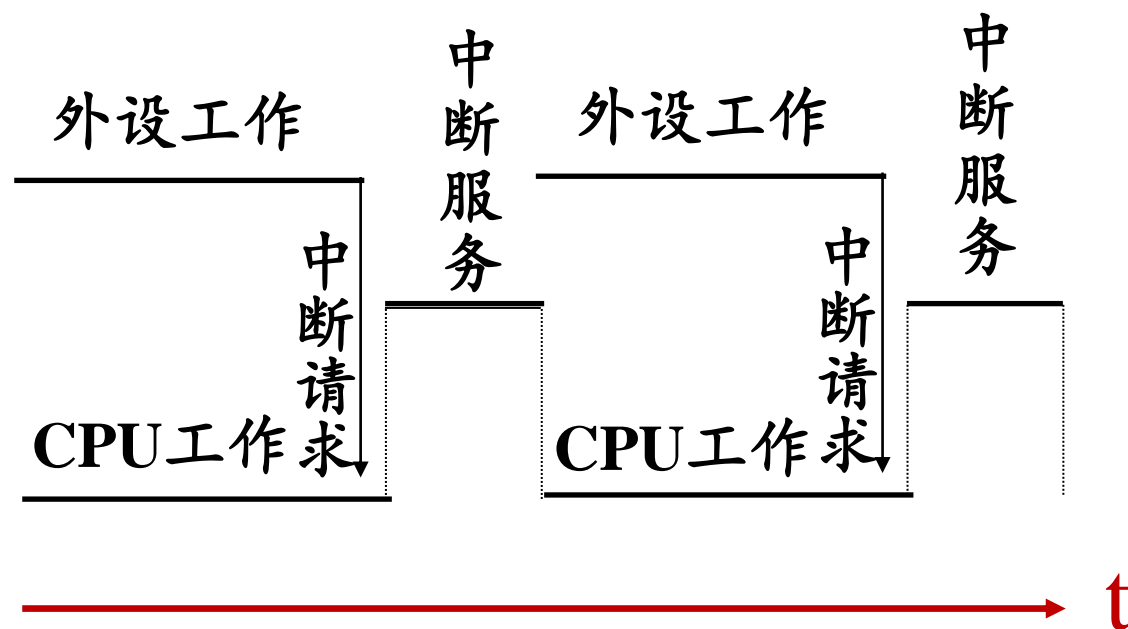
- 最简单的输入输出方式
- 程序根据查询到的外设状态，控制数据何时进行输入输出操作
- 特点是：
  - ◆ CPU需要不断地查询并等待相对慢速的外设
  - ◆ 效率低，CPU与外设不能并行工作
  - ◆ 接口电路简单，硬件开销小





## 2. 程序中中断方式

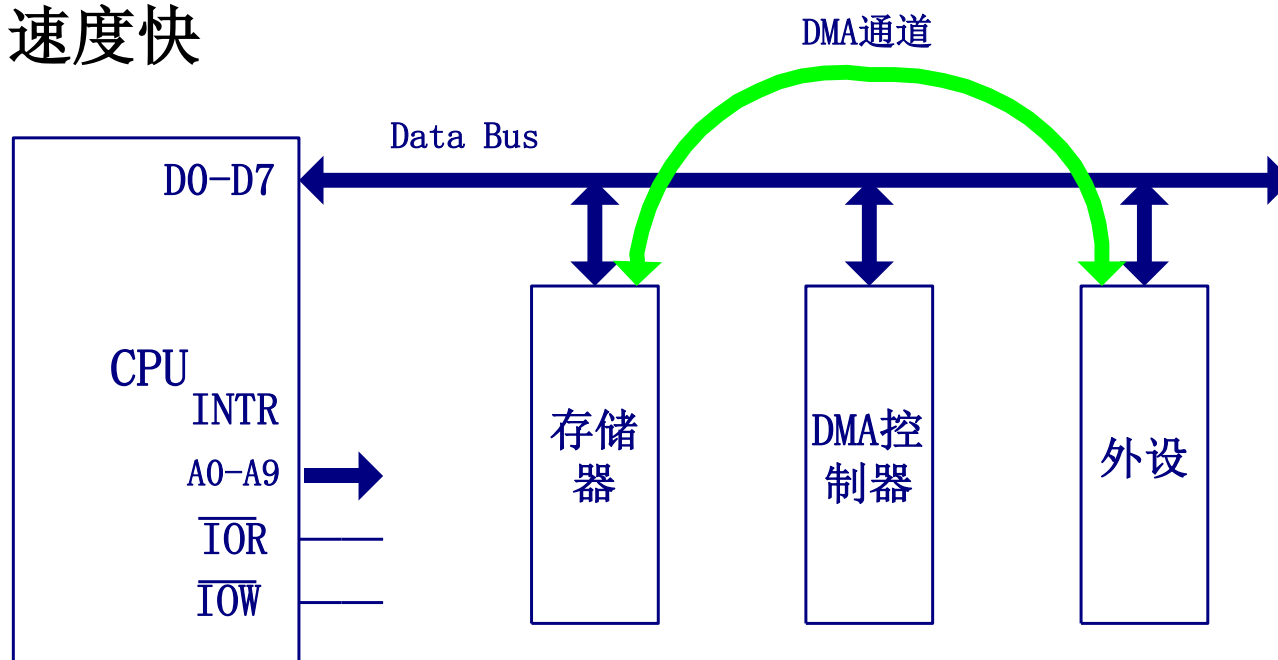
当外设需要CPU服务时，通过接口向CPU发出中断请求。CPU在当前机器指令执行完毕后暂停正在执行的主程序，转去执行中断服务程序，待中断服务程序执行完毕，再返回到原程序继续执行。



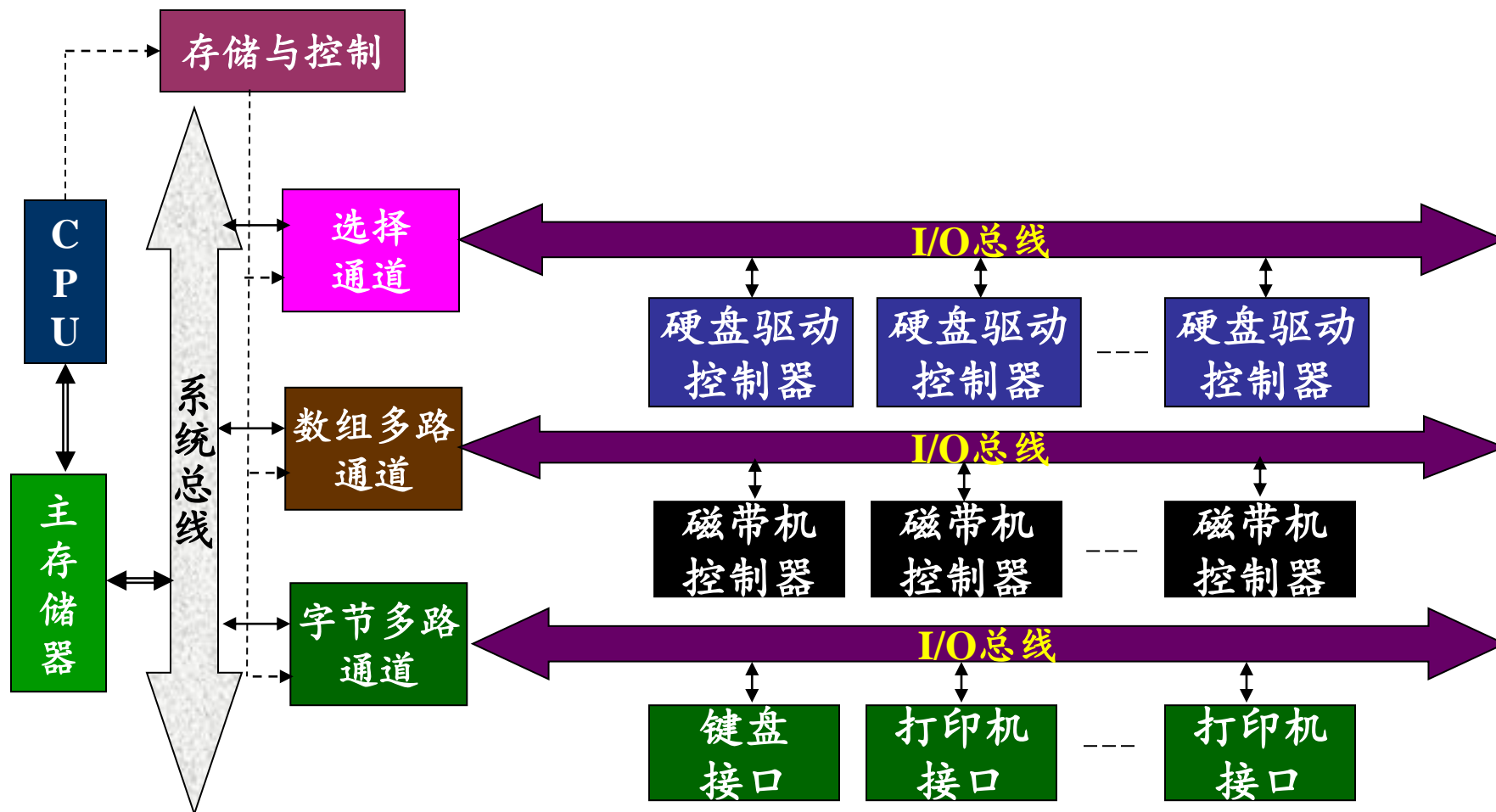


# 3. DMA方式

- 直接存储器访问DMA（Direct Memory Access）方式是为在主存储器与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



# 4. 通道方式





# 通道方式

- 计算机系统规模最大后，系统中I/O设备数量多、I/O操作频繁，仅采用中断处理方式和DMA方式不能满足系统的要求
- 通道是一种具有特定功能的处理器，它具有自己的指令和程序，专门负责数据输入输出的控制和管理
- 通道与CPU分时使用主存，实现CPU内部运算与I/O设备并行工作



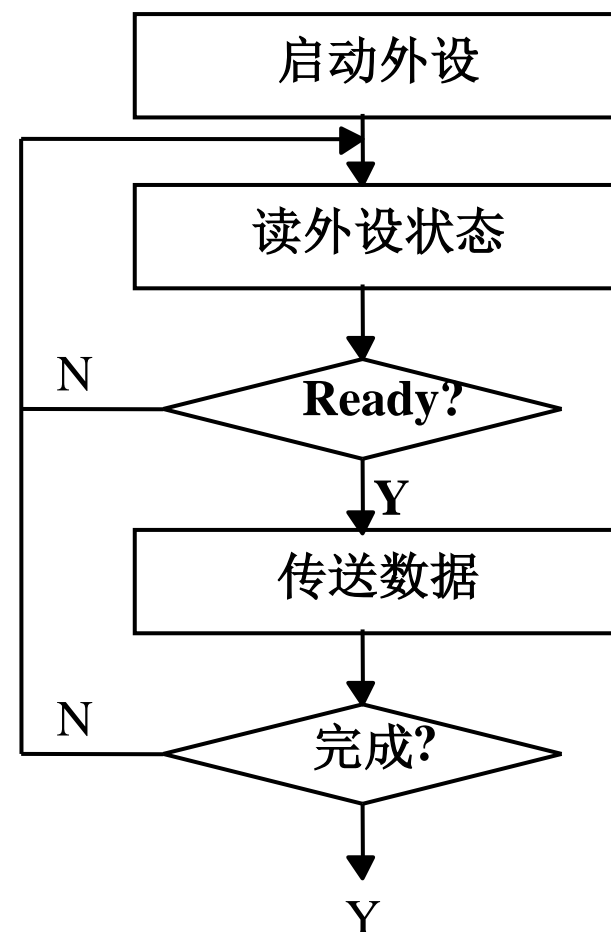
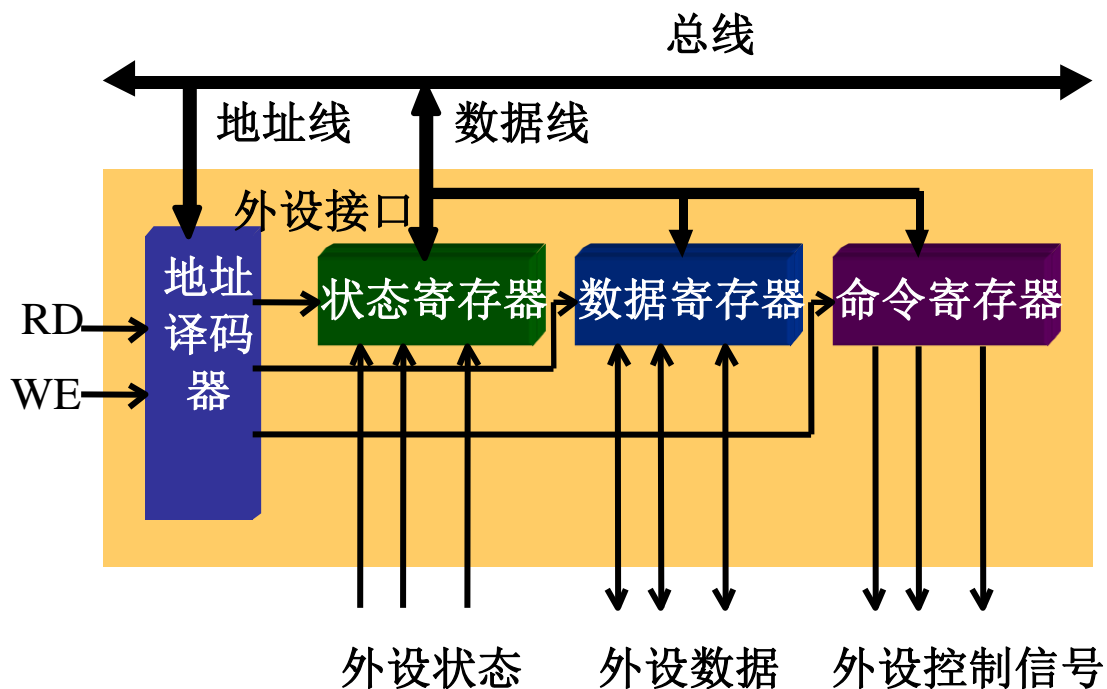


## 8.2 程序查询方式



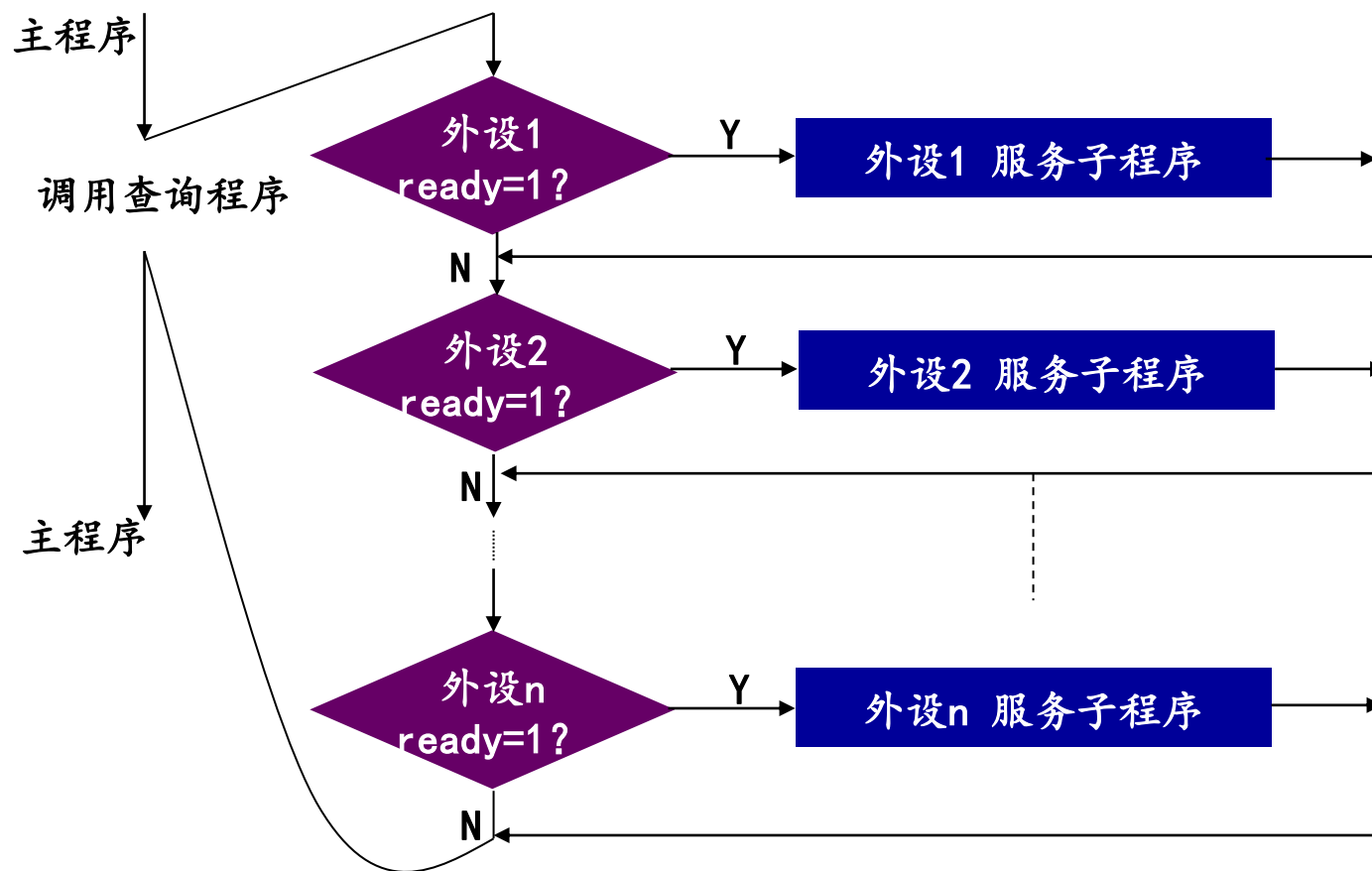


# 1. 查询传送方式





# 多台外设的程序查询过程







## 8.3 程序中断方式





# 中断的概念

- 中断（Interrupt）是指CPU暂时中止现行程序，转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，一般包括：
  - ◆ 在CPU中设置的中断机构
  - ◆ 在系统中设置的中断控制器
  - ◆ 在软件上设置相应的中断服务程序





# 中断方式的典型应用

- 以中断方式实现CPU与外界进行信息交换的握手联络，实现CPU与外设的并行工作
- 故障处理
- 实时处理
- 程序调度
- 软中断（程序自愿中断）





# 中断源

- 中断源：能够向CPU发出中断请求的事件（部件）
- 常见中断源有：
  - ◆ 输入、输出设备中断。如键盘、打印机等工作过程中已做好接收或发送准备
  - ◆ 数据通道中断。如磁盘、磁带等要同主机进行数据交换
  - ◆ 实时时钟中断
  - ◆ 故障中断。例如电源掉电、设备故障等要求CPU进行紧急处理等
  - ◆ 系统中断。如运算过程出现溢出、数据格式非法，数据传送过程出现校验错，控制器遇到非法指令等等
  - ◆ 为了调试程序而设置的中断





# 中断处理过程

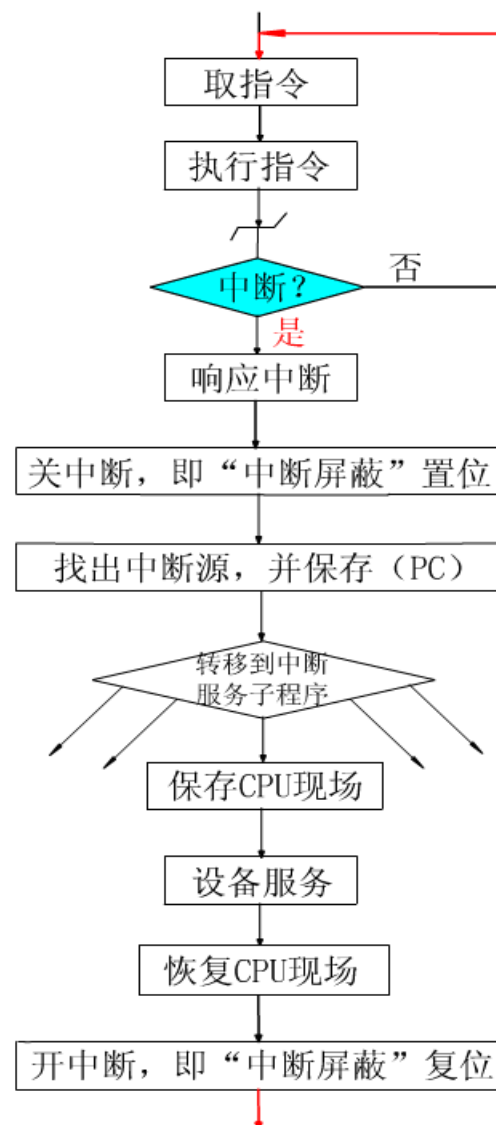
- 中断请求
- 中断排队与判优
- 中断响应
  - ◆ 暂时中断现程序的执行，发出中断响应信号
  - ◆ 保存断点（返回地址）
  - ◆ 关中断
  - ◆ 查找中断源，获取中断服务程序入口地址
- 中断处理
  - ◆ 保护现场
  - ◆ 中断服务
- 恢复现场与中断返回

硬件

中断周期

软件

中断服务子程序



返回主程序

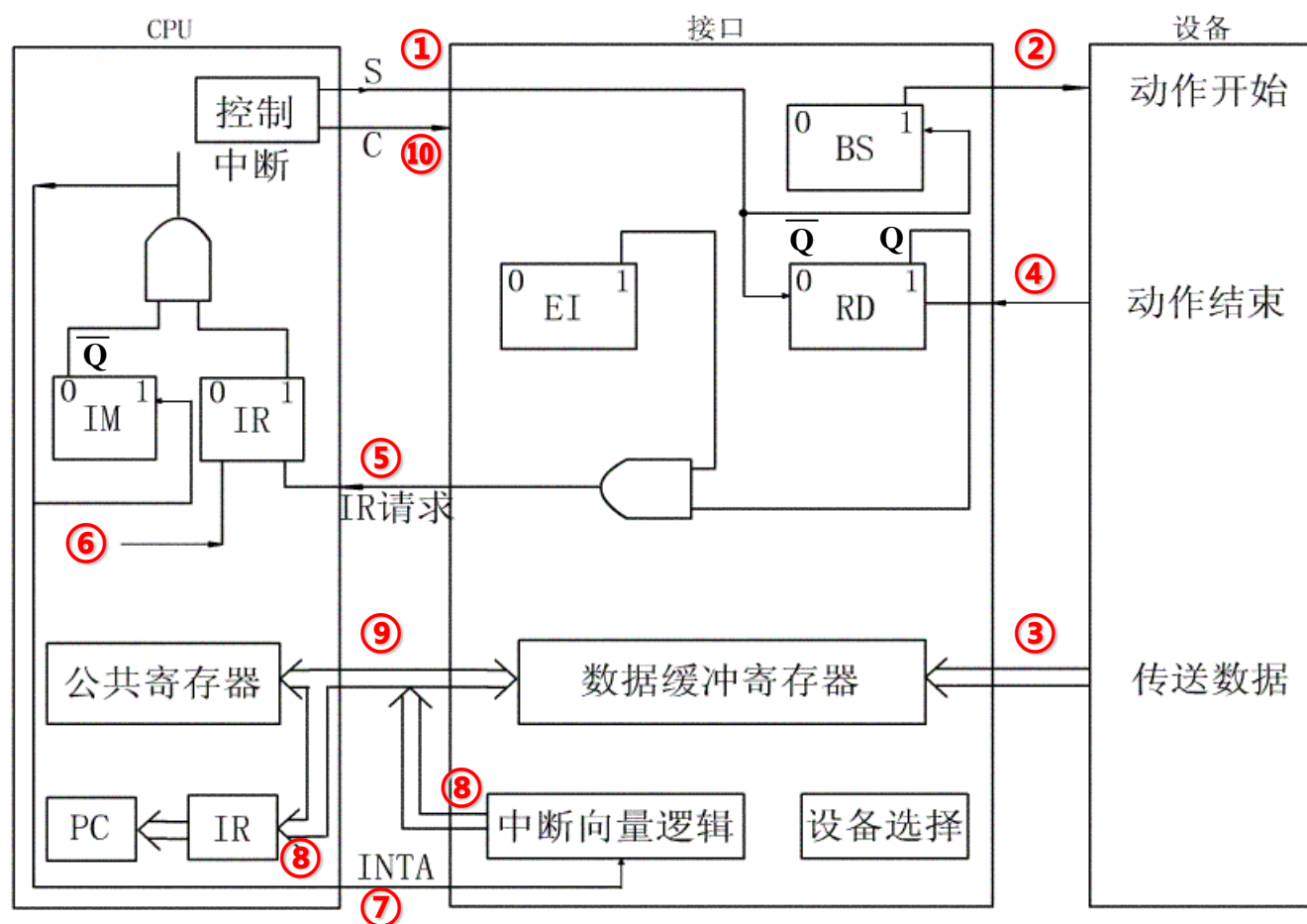




# 程序中中断方式基本接口

- RD: 准备就绪
- EI: 允许中断
- BS: 忙
- IM: 中断屏蔽
- IR: 中断请求

当EI=1, IM=0时  
响应中断请求。  
EI和IM可以通过软件进行设置





# 中断源的识别

- 如何确定哪一个中断源发出的中断请求，并转入被响应的中断服务程序入口地址，是中断处理必须要解决的问题

识别中断源：

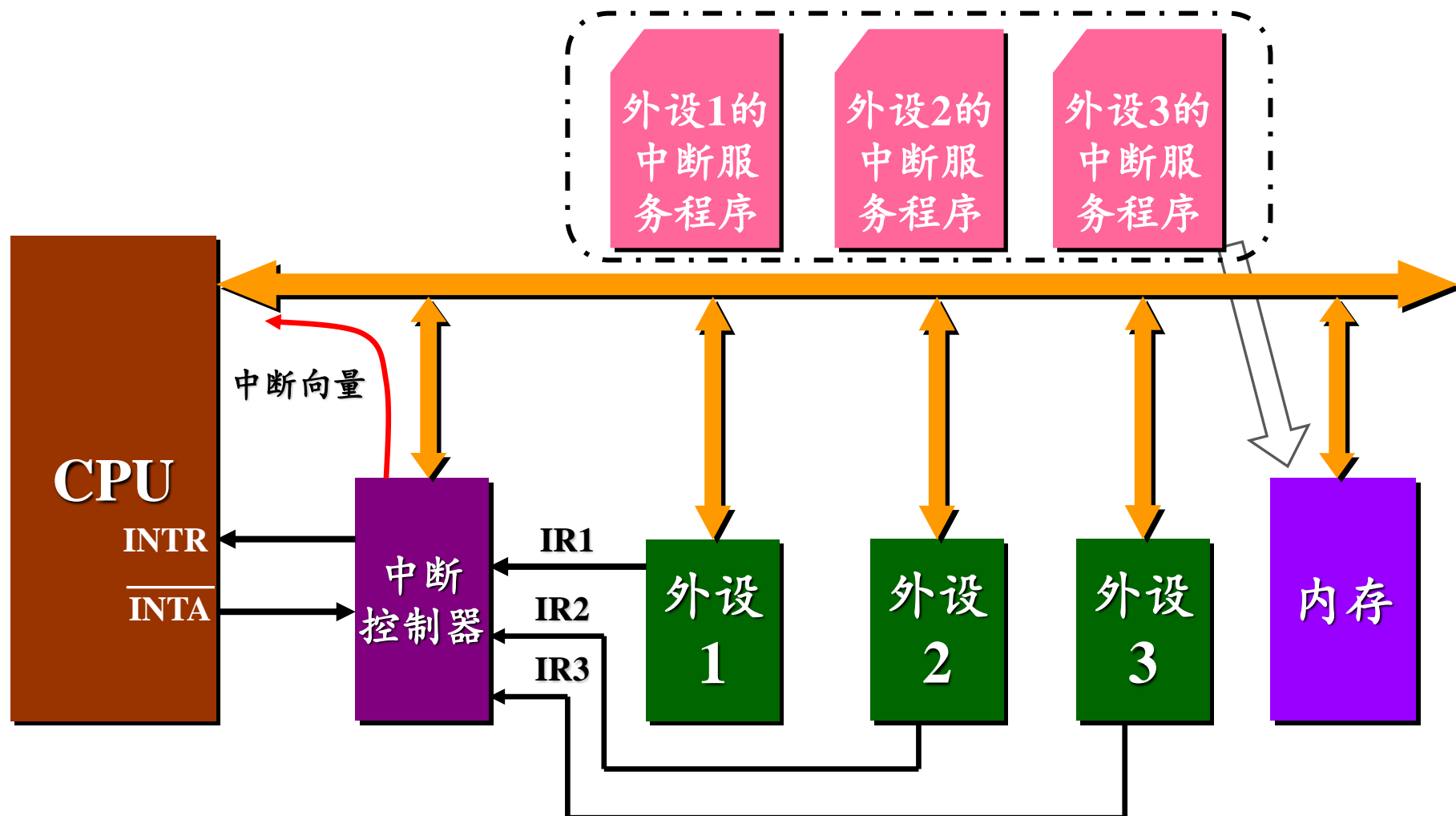
1. 在单级中断中，采用**串行排队链**法来实现具有公共请求线的中断源判优识别
2. 采用**中断控制器**，识别哪个设备发出的中断

转中断服务程序：

1. 预先规定中断服务程序的入口地址
2. 采用中断向量，通过查表方式找到入口地址



# 基于中断向量的中断源识别







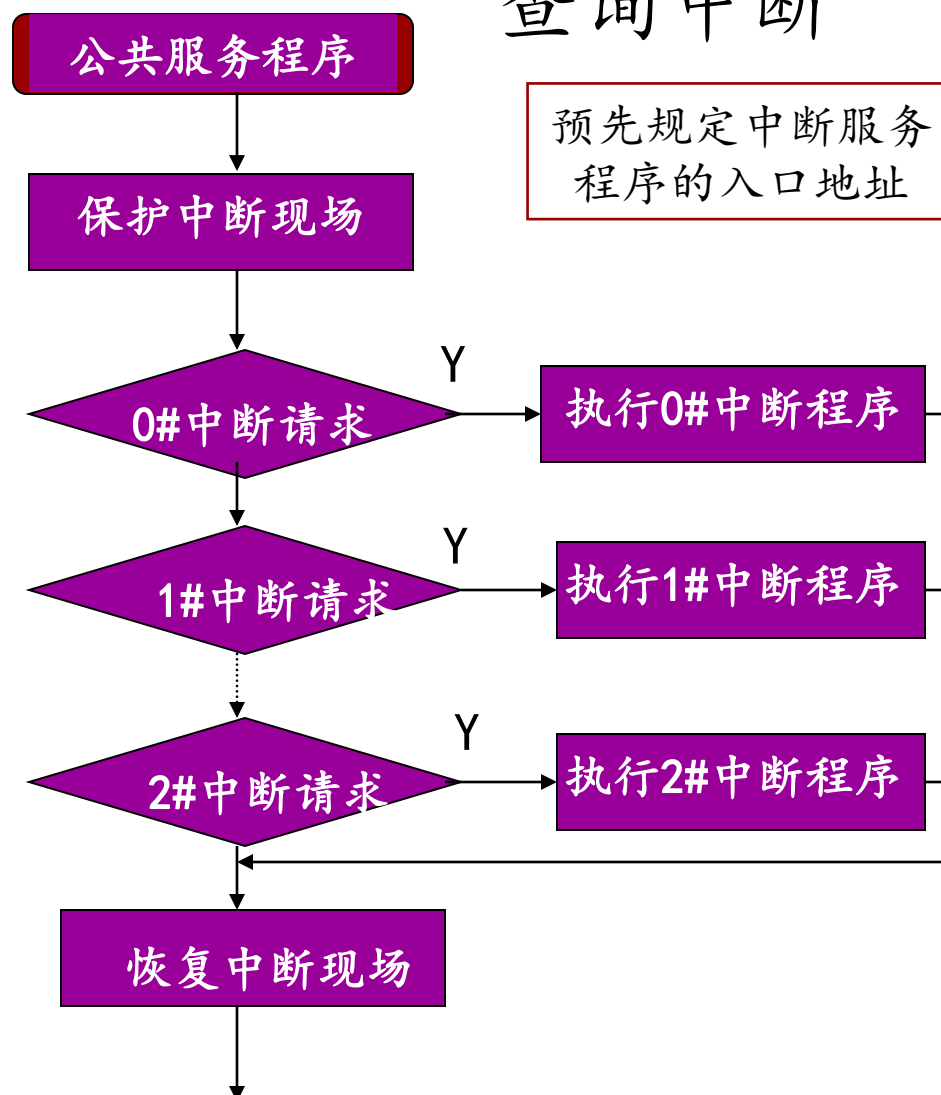
# 中断服务程序入口的获取

## 向量中断

类型255
类型33
类型32
类型31
类型5
类型4
类型3
类型2
类型1
类型0

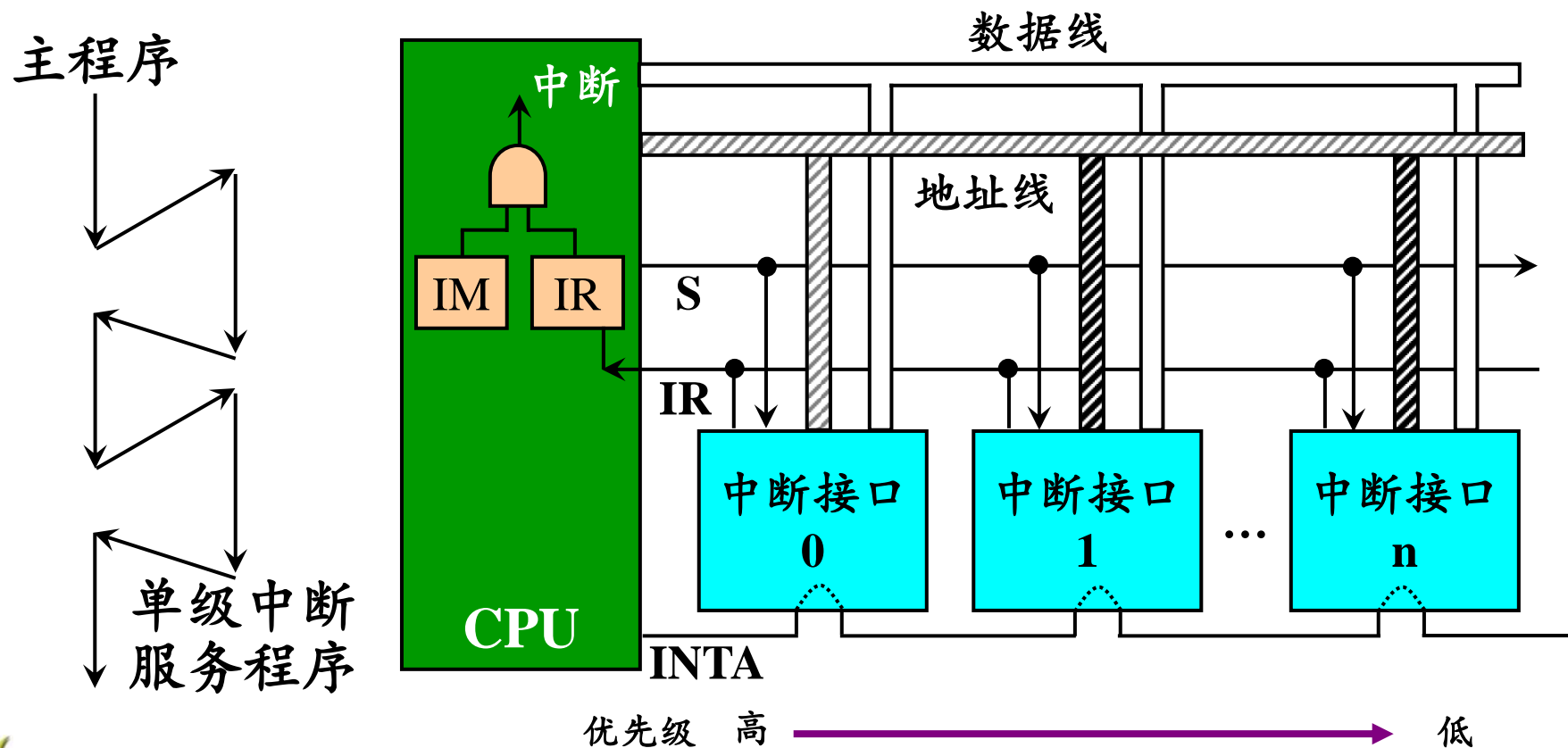
## 中断向量表

## 查询中断

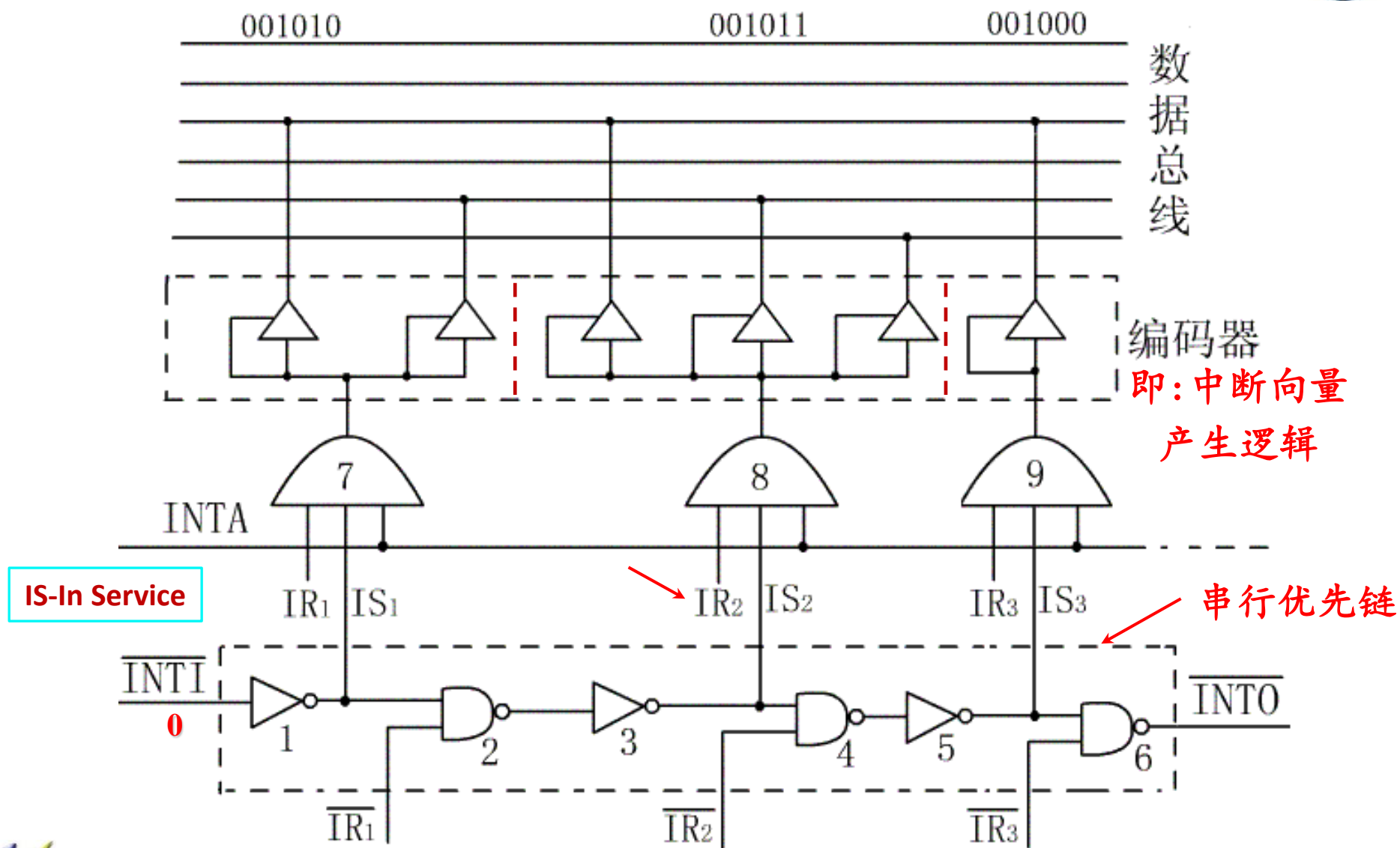


# 单级中断

- 当响应某一中断请求时，在执行中断服务程序的过程中，不允许其他中断源再次打断中断服务程序。即中断响应不允许嵌套。



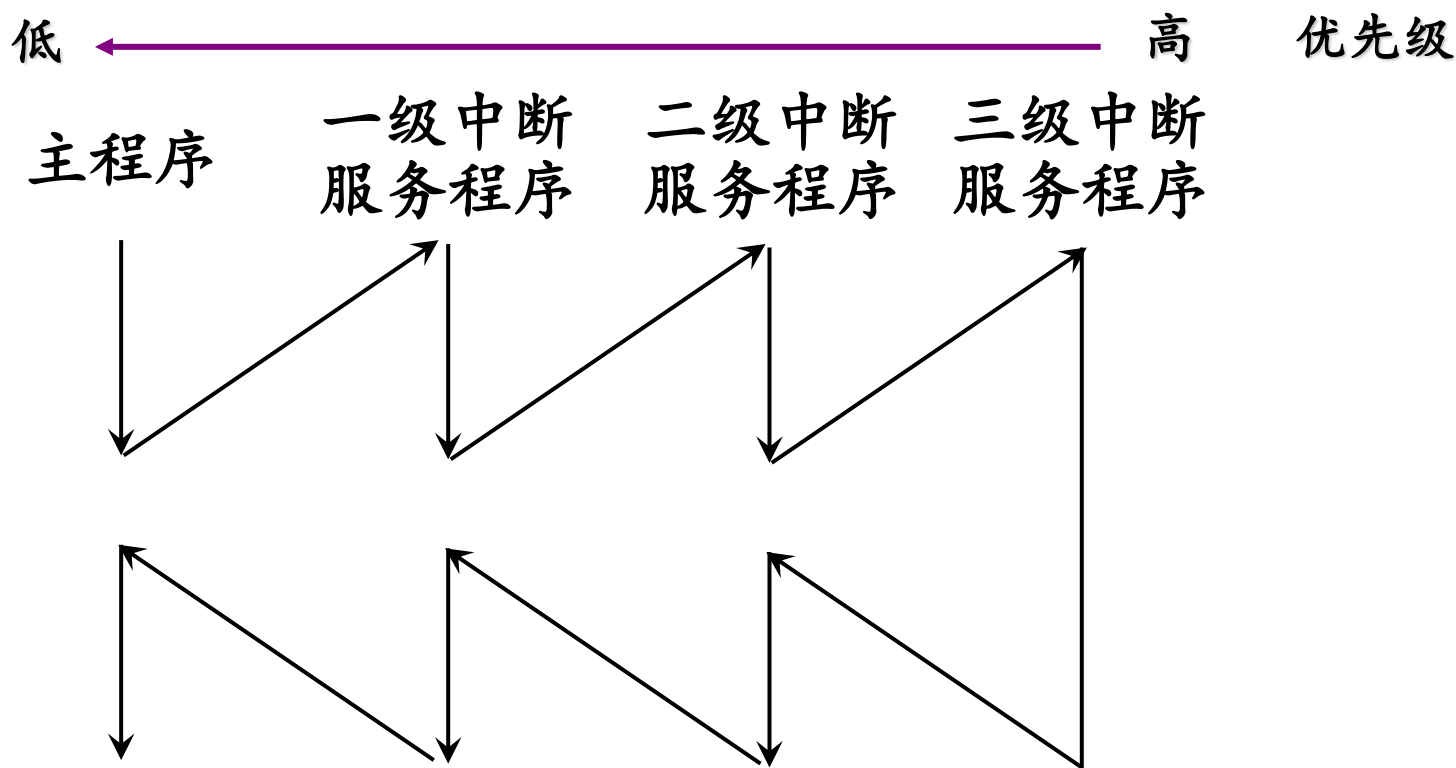
# 串行链式排队电路及中断向量的产生





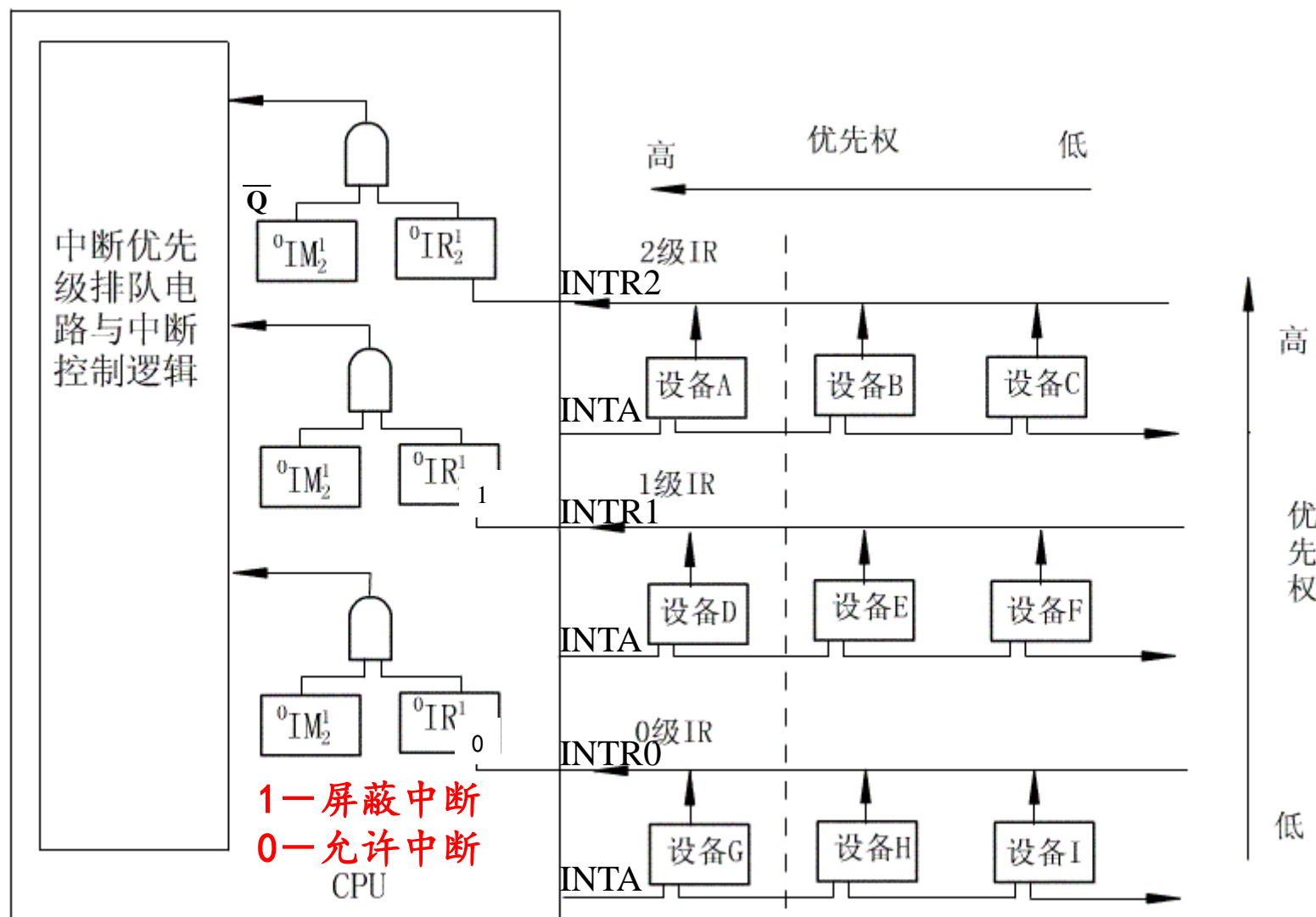
# 多级中断

- 多级中断通常称为**中断嵌套方式**
- 优先级高的中断级可以打断优先级低的中断服务程序，中断服务程序以嵌套方式进行工作





# 一、二维多级中断结构



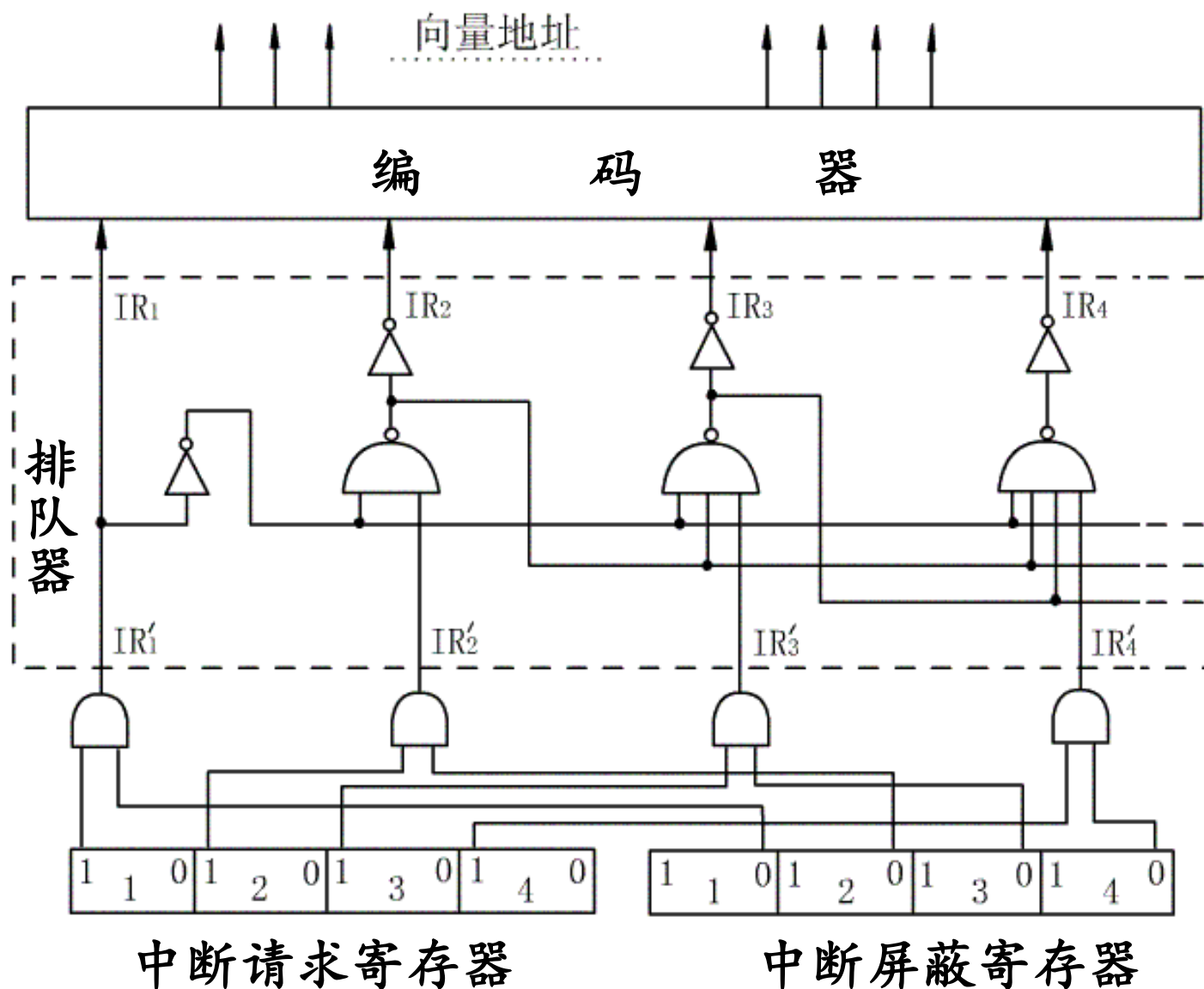


# 说明

- 一个系统有 $n$ 级中断，则CPU中有 $n$ 个IR， $n$ 个IM，某级中断被响应后，则关闭本级和低于本级的IM（即：对应的IM置1），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
- 中断响应时，确定哪一级中断和中断源采用硬件实现。这里采用独立请求方式+链式查询方式相结合的方式
- 使用堆栈保存现场（包括IM）



# 独立请求方式的优先级排队逻辑





# 例1

如图所示的二维中断系统。请问：

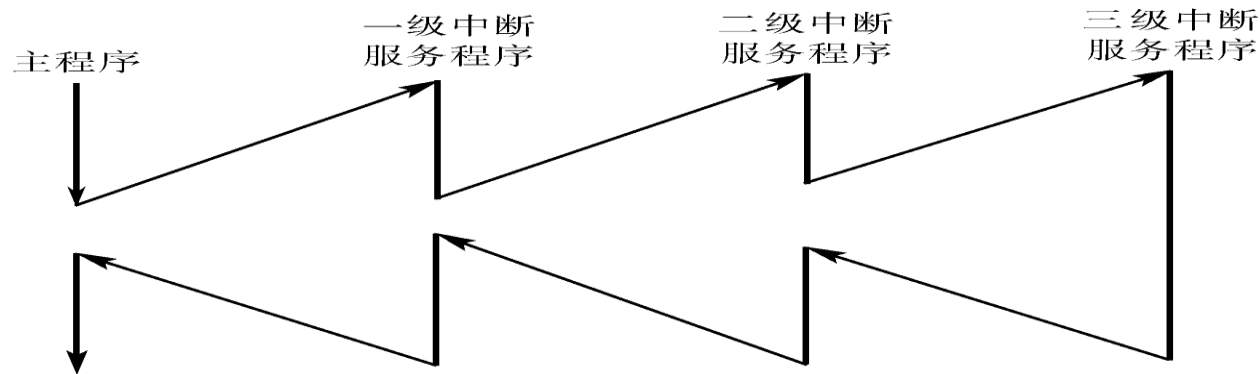
- (1)在中断情况下，**CPU**和设备的优先级如何考虑？请按降序排列各设备的中断优先级。
- (2)若**CPU**现执行设备**B**的中断服务程序，**IM2**，**IM1**，**IM0**的状态是什么？如果**CPU**执行设备**D**的中断服务程序，**IM2**，**IM1**，**IM0**的状态又是什么？
- (3)每一级的**IM**能否对某个优先级的个别设备单独进行屏蔽？如果不能，采取什么办法可达到目的？
- (4)假如设备**C**一提出中断请求，**CPU**立即进行响应，如何调整才能满足此要求？



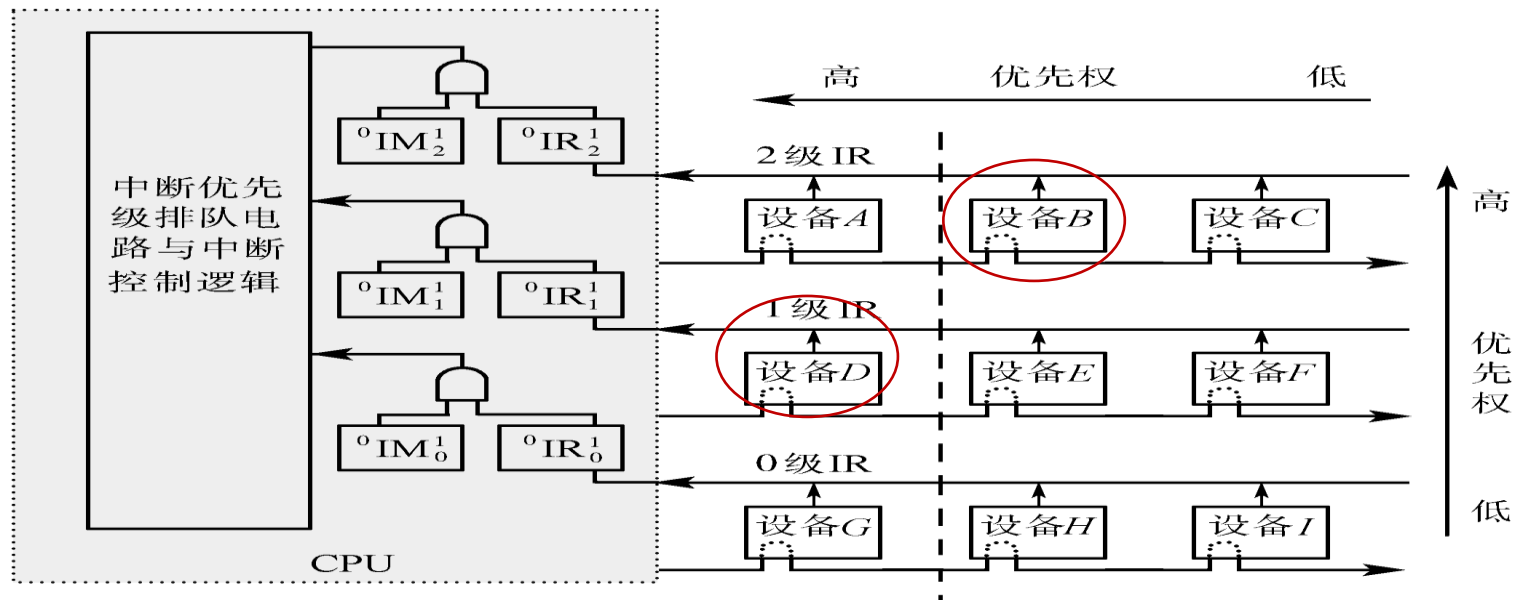




# 例1图



(a) 多级中断示意图



(b) 一维、二维多级中断结构





# 例1解

- (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， $IM_2IM_1IM_0=011$ 。
- (3)每一级的IM标志不能对某个优先级的个别设备进行单独屏蔽。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 $IM_3=0$ 即可。





## 例2

- 参见例1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？





# 解

参阅中断处理流程，并**假设**执行一条指令的时间也为  $T_M$ 。**假定**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：

$$t_A = T_{DC} + T_S + T_A + T_R + 2T_M$$

$$t_B = 2T_{DC} + T_S + T_B + T_R + 2T_M$$

$$t_C = 3T_{DC} + T_S + T_C + T_R + 2T_M$$

$2T_M$ 表示取指令并执行指令的时间

处理三个设备所需的总时间为： $T = t_A + t_B + t_C$

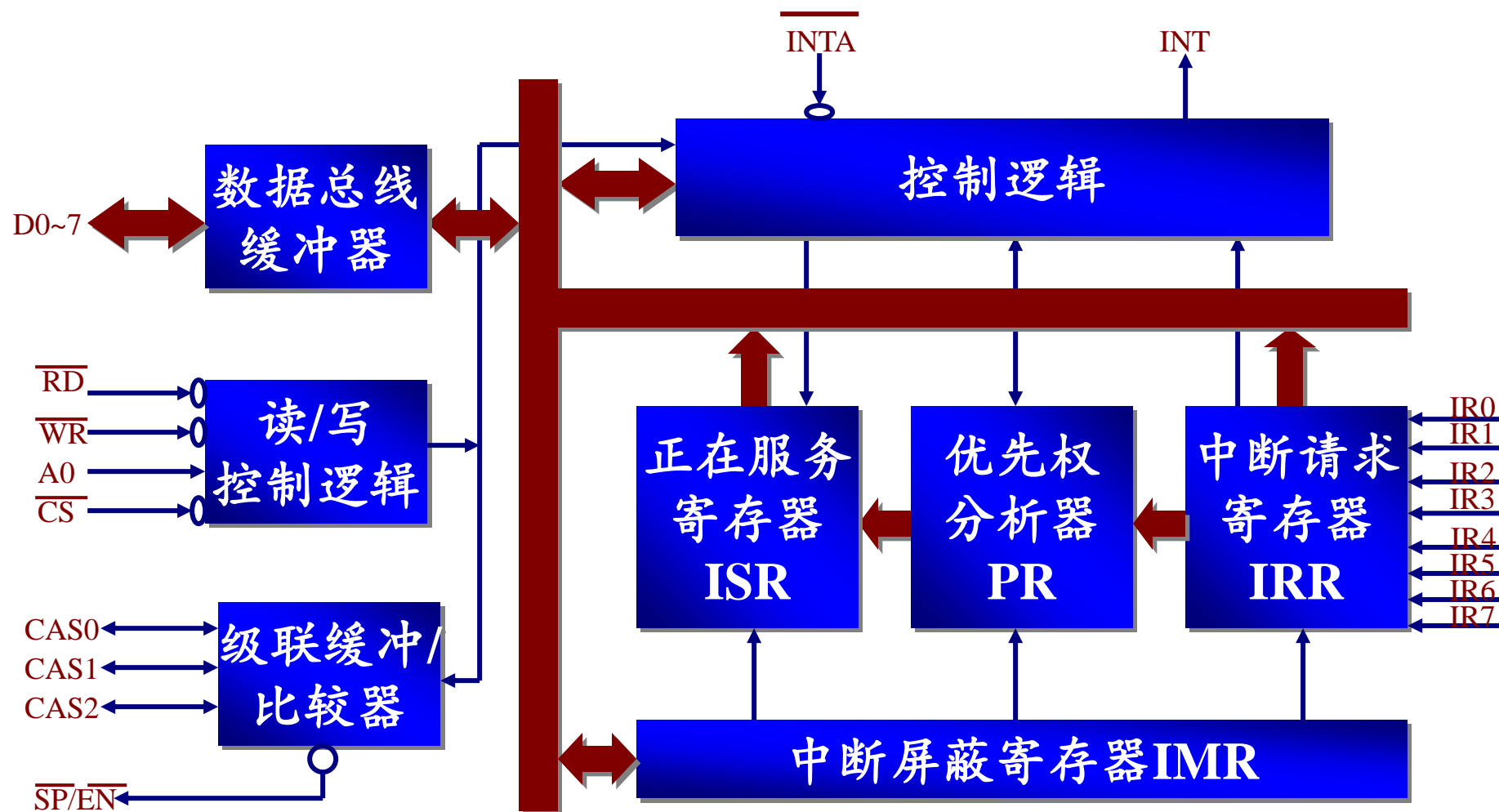
$T$ 是达到中断饱和的最小时间，即中断极限频率为：

$$f = 1/T$$





# 中断控制器





# 主要功能

- 中断请求寄存器（**IRR**）记录外部中断源发出的中断请求信号， **Di**位为1表示**IRi**引脚有中断请求。具有锁存功能
- 正在服务寄存器（**ISR**）存放正在被服务中的中断请求信号。
- 优先权分析器（**PR**）把**IRR**的内容与**ISR**的内容进行比较，向**CPU**提交优先级高的中断请求
- 中断屏蔽寄存器（**IMR**）“0”允许中断； “1” 屏蔽中断





# 10

单级中断系统中，中断服务程序执行顺序是（ ）

**I、保护现场 II、开中断 III、关中断 IV、保存断点  
V、中断事件处理 VI、恢复现场 VII、中断返回**

**A: I、V、VI、II、VII**

**B: III、I、V、VII**

**C: III、IV、V、VI、VII**

**D: IV、I、V、VI、VII**





# 11

某设备中断请求的响应和处理时间为100ns，每400ns发出一次中断请求，中断响应所容许的最长延迟时间为50ns，则在该设备持续工作过程中CPU用于该设备的I/O时间占整个CPU时间百分比至少是

- A. 12.5%
- B. 25%
- C. 37.5%
- D. 50%







# 5

21. 下列选项中，在 I/O 总线的数据线上传输的信息包括

I. I/O 接口中的命令字

II. I/O 接口中的状态字

III. 中断类型号

A. 仅 I、II

B. 仅 I、III

C. 仅 II、III

D. I、II、III

22. 响应外部中断的过程中，中断隐指令完成的操作，除保护断点外，还包括

I. 关中断

II. 保存通用寄存器的内容

III. 形成中断服务程序入口地址并送 PC

A. 仅 I、II

B. 仅 I、III

C. 仅 II、III

D. I、II、III





## 8.4 DMA方式





# 基本概念

- 直接存储器访问(DMA)是一种完全由硬件实现I/O交换的工作方式
- 在这种方式中，DMA控制器从CPU接管对总线的控制，**数据交换不经过CPU**，而直接在内存和I/O设备之间进行
- DMA控制器向内存发出地址和控制信号，并修改地址，对所传送的字的个数计数。用中断方式或状态位方式向CPU报告传送操作结束
- DMA方式的主要优点是速度快，能满足高速I/O设备传送的要求





# DMA传送过程

## ■ 申请阶段

- ◆ DMA控制器收到DMA传送请求后，向CPU发出总线请求信号，申请占用总线

## ■ 响应阶段

- ◆ CPU响应总线请求，释放总线（即将三总线变为高阻），并回送总线应答信号，表示已让出总线

## ■ 数据传送阶段

- ◆ DMA控制器收到总线应答信号后，占用总线，分别向存储器和外设发出读/写控制信号，完成数据传送操作，并修改地址记录传送字的个数

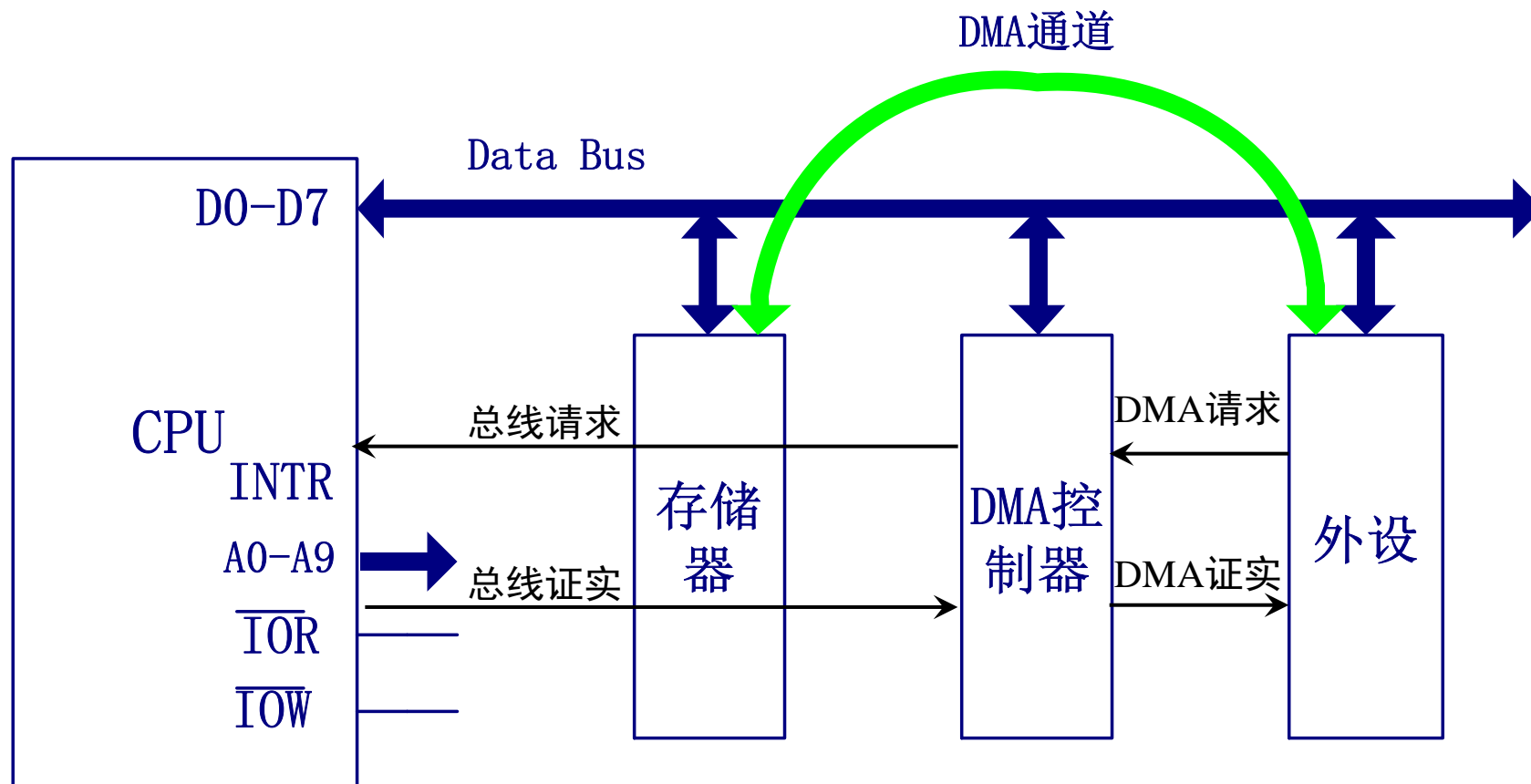
## ■ 传送结束阶段

- ◆ 在规定字数传送完后，DMA控制器以中断方式或状态位的方式向CPU报告传送操作结束，释放总线，CPU重新占用总线。





# DMA方式的数据传送过程





# DMA传送方式

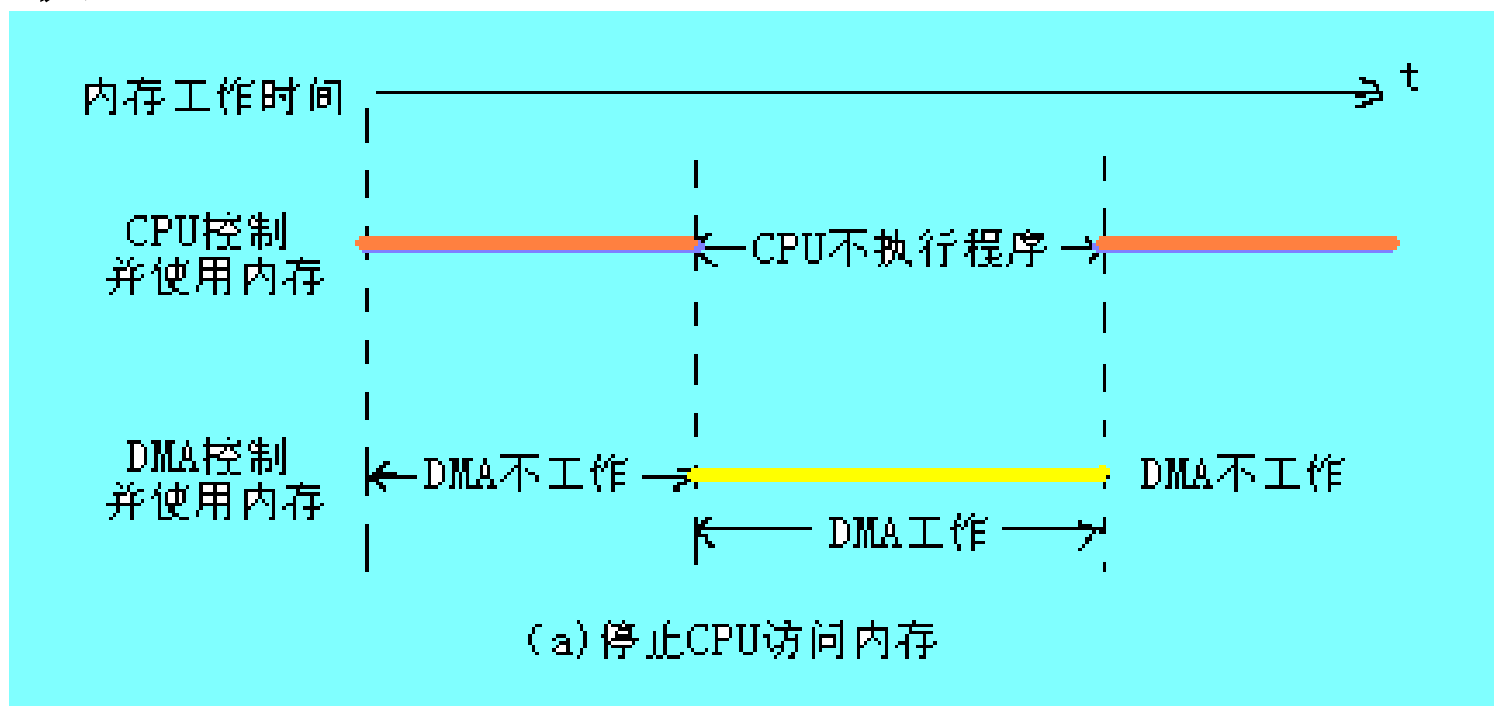
- 成组传送方式（停止CPU访问内存）
- 周期窃取方式
- DMA与CPU交替访内存方式





# 成组传送方式

- CPU响应总线请求后，让出总线。DMA控制器把一组数据传送完毕后，才把总线控制权交还给CPU。
- DMA传送过程中，CPU基本处于不工作状态或者说保持状态





# 特点

- 优点：控制简单，它适用于数据传输率很高的设备进行成组传送。
- 缺点：在DMA传送过程中，CPU没有总线控制权，无法处理紧急事件。
- 成组传送的数据块不能太大

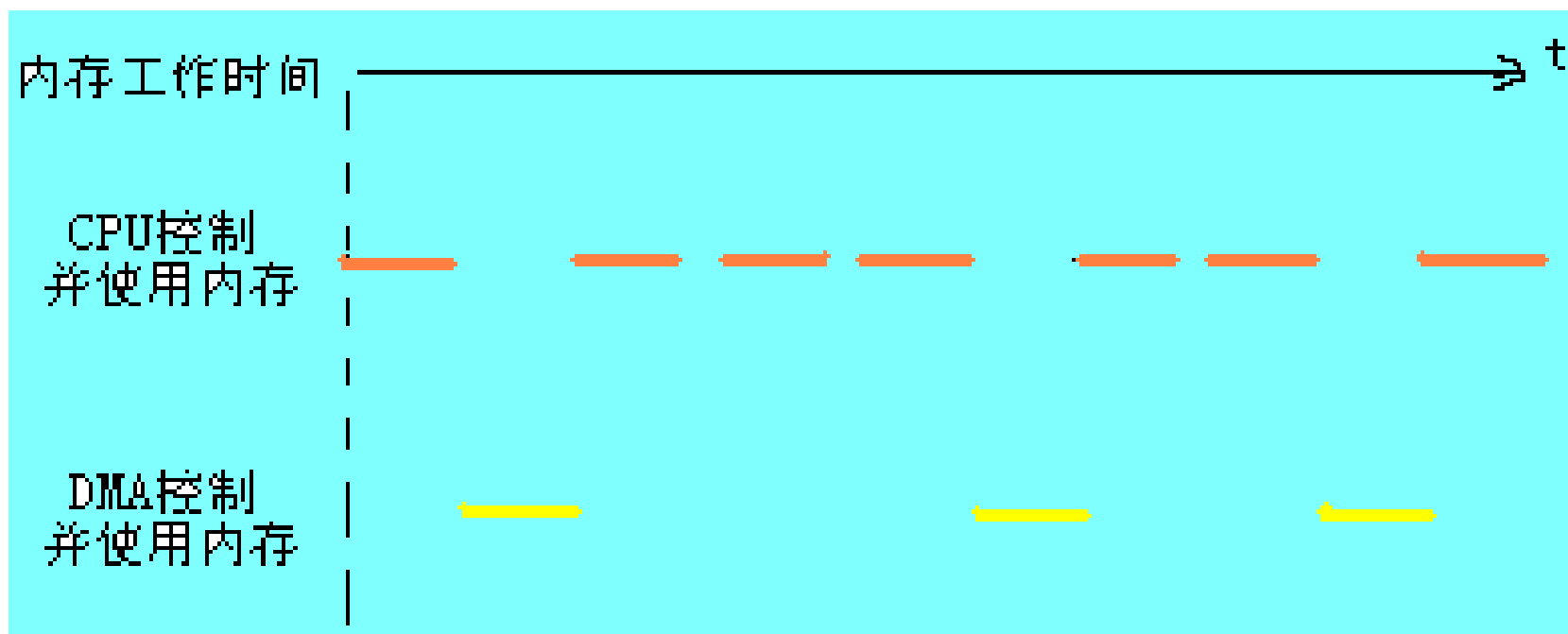






# 周期窃取方式

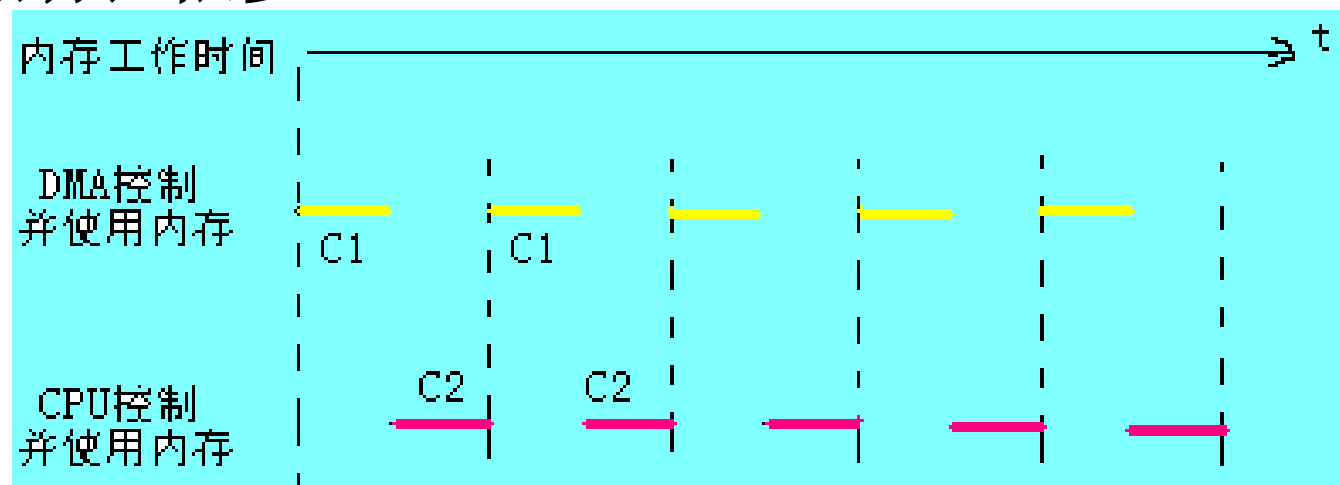
- **DMA控制器与主存储器之间传送一个数据，占用（窃取）一个CPU周期，即CPU暂停工作一个周期，然后继续执行程序**
- **特点：总线申请和归还存在过多的时间开销**





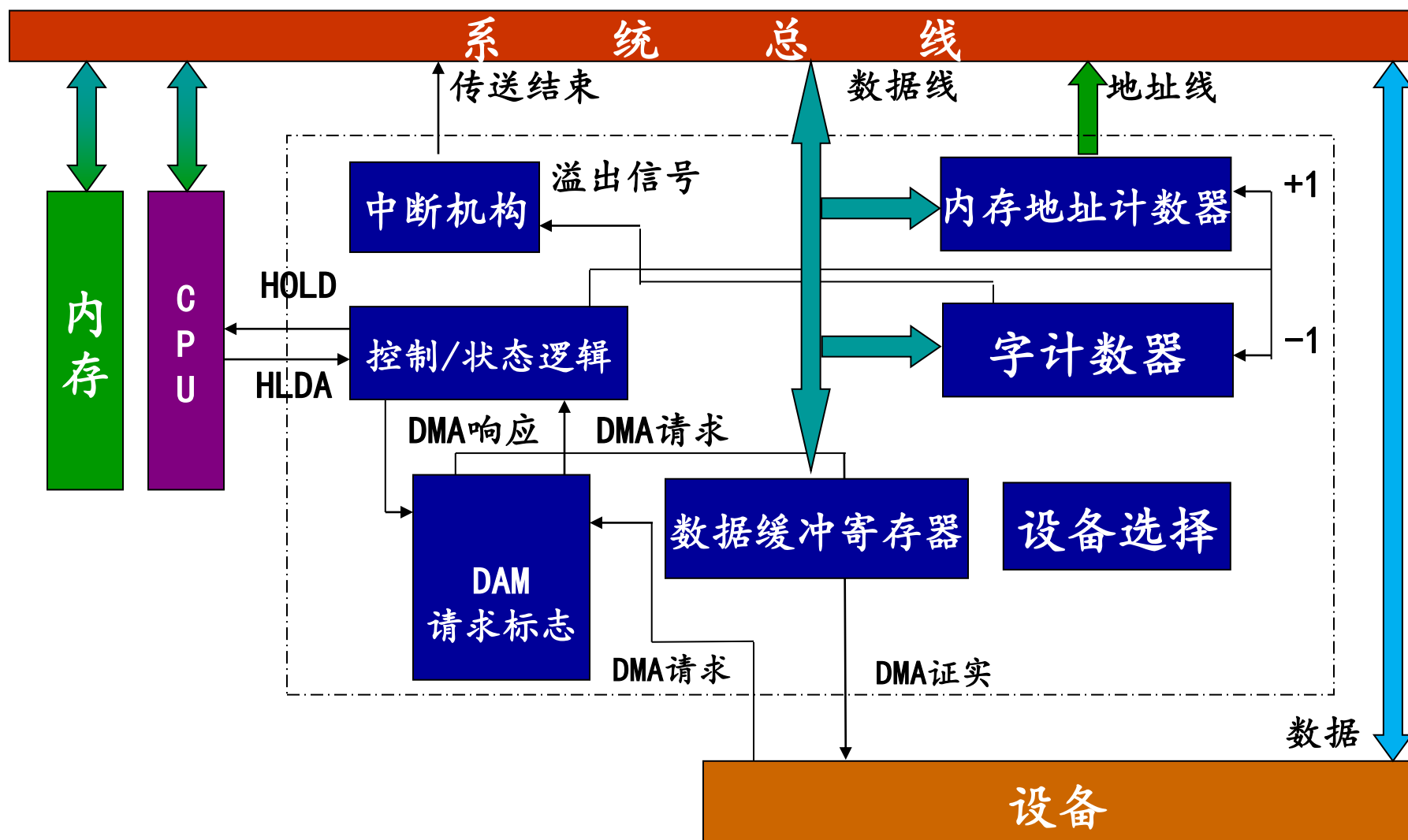
# DMA与CPU交替访内存方式

- 不需要总线使用权的申请、建立和归还过程，总线使用权是通过C1和C2分时进行的（ $C1+C2=\text{CPU周期}$ ）
- 这种传送方式也称为“透明的DMA”方式。CPU既不停止主程序的运行，也不进入等待状态
- 使用该方式的前提条件：**CPU**的工作周期比内存存取周期长很多





# DMA控制器的基本组成





# 说明

- 内存地址计数器：内存当前地址指针。传送前，初值为内存缓冲区首地址，每传送一次数据，自动加1
- 字计数器：传送数据块长度计数，每传送一次数据，自动减1
- 数据缓冲寄存器：采用从存储器到存储器传送方式时，暂存每次传送的数据
- 控制/状态逻辑：完成控制功能
- 中断机构：传送结束信号。表示DMA传送过程结束。可用作向CPU发出传送结束请求。





# DMA数据传送处理过程

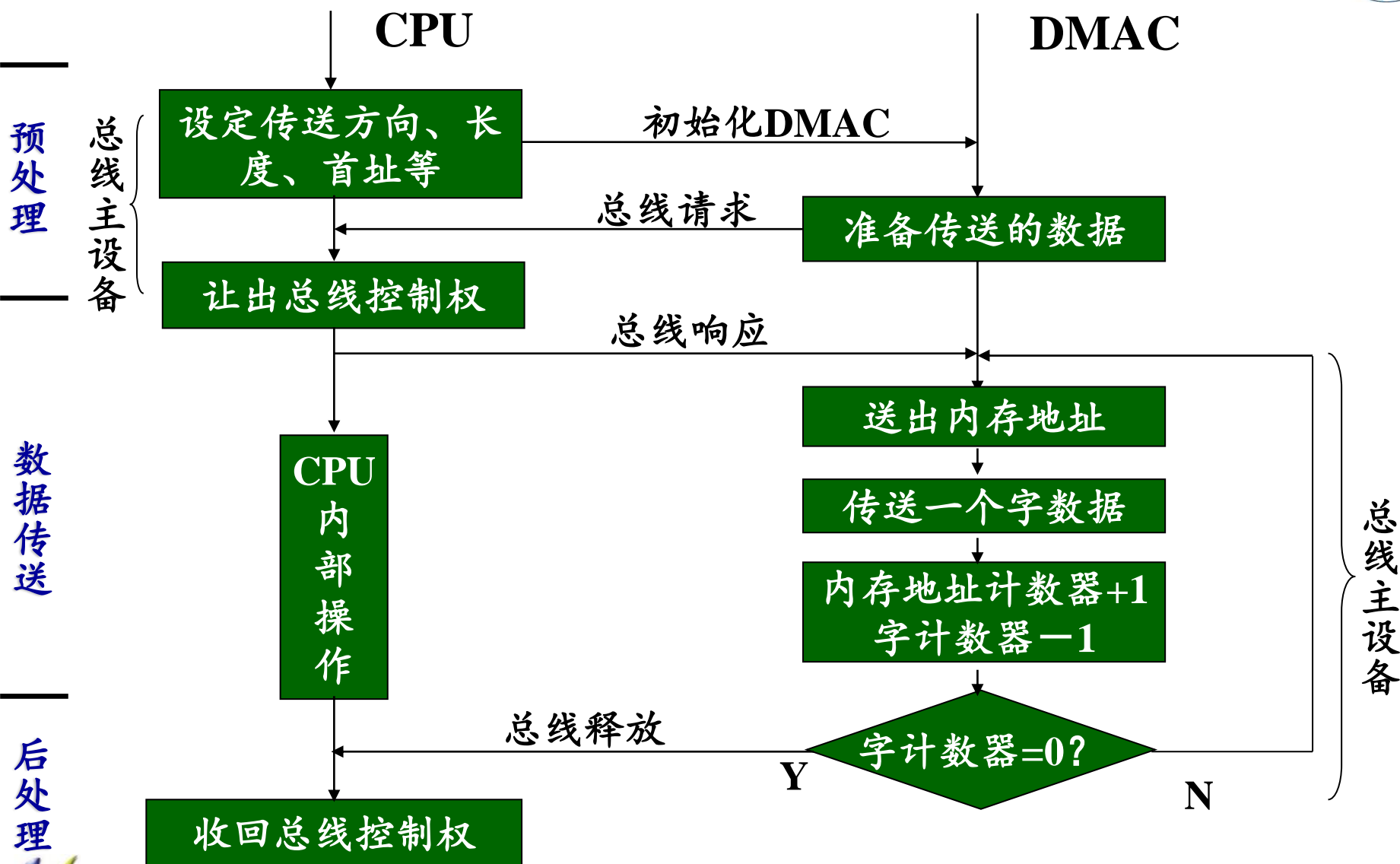
传送一个数据块可以分为三个阶段：

- 预处理：CPU对DMA控制器预置初值，读取状态和送传送需要的有关参数。
- 数据传送：DMA控制器控制外设与主存之间的数据交换。
- 后处理：
  - ◆ 若需要继续交换数据，则再次对DMA进行初始化
  - ◆ 若不需要交换数据，则停止外设
  - ◆ 若为出错，则转错误诊断及处理程序





# DMA数据传送流程图



# 选择型和多路型DMA控制器

## ■ 选择型

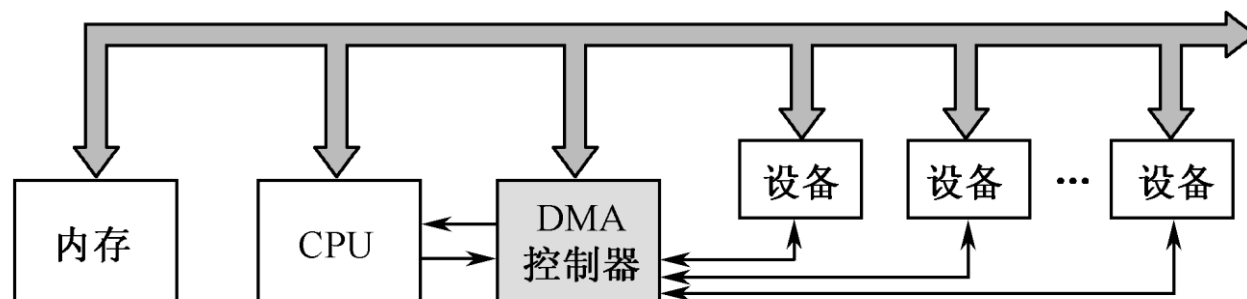
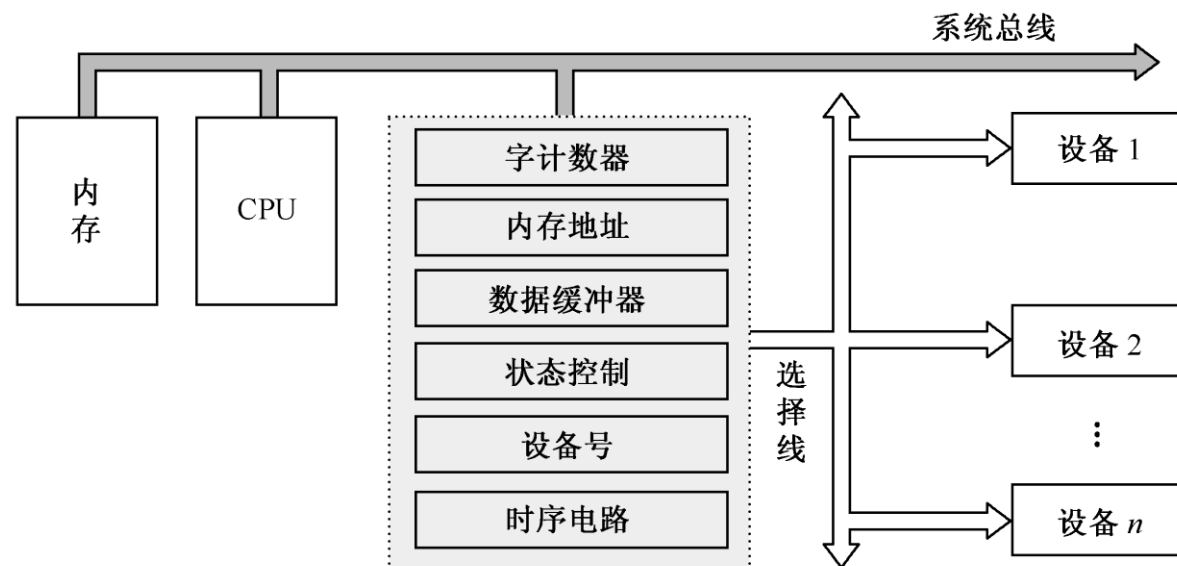
- ◆ 在某一段时间内只能为一个设备服务

- ◆ 共用硬件

## ■ 多路型

- ◆ DMA控制器**同时**为多个慢速设备服务

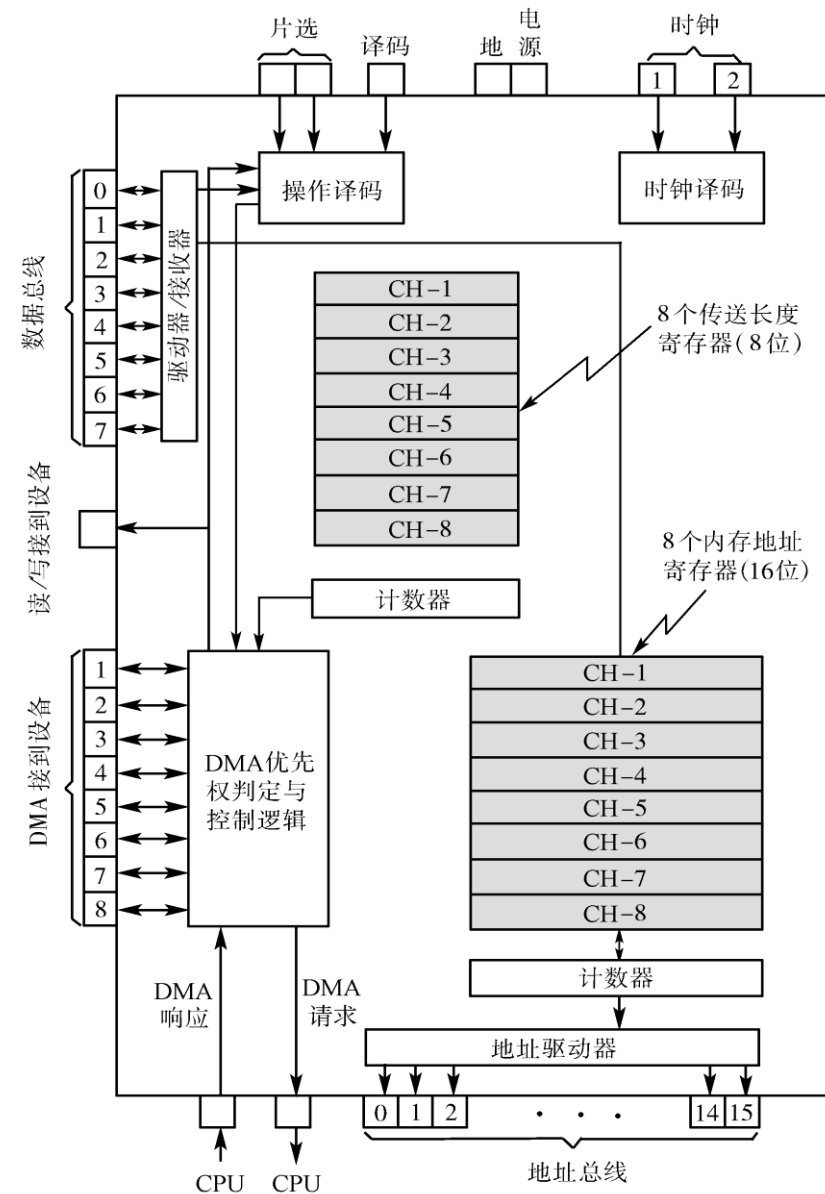
- ◆ 各设备能以字节交叉方式进行数据传送





# 典型多路DMA芯片

- 八个独立的DMA通道
- 优先级：通道0最高，通道7最低
- 8个16位地址寄存器
- 8个8位传送长度寄存器，即最大数据块长度为256字节







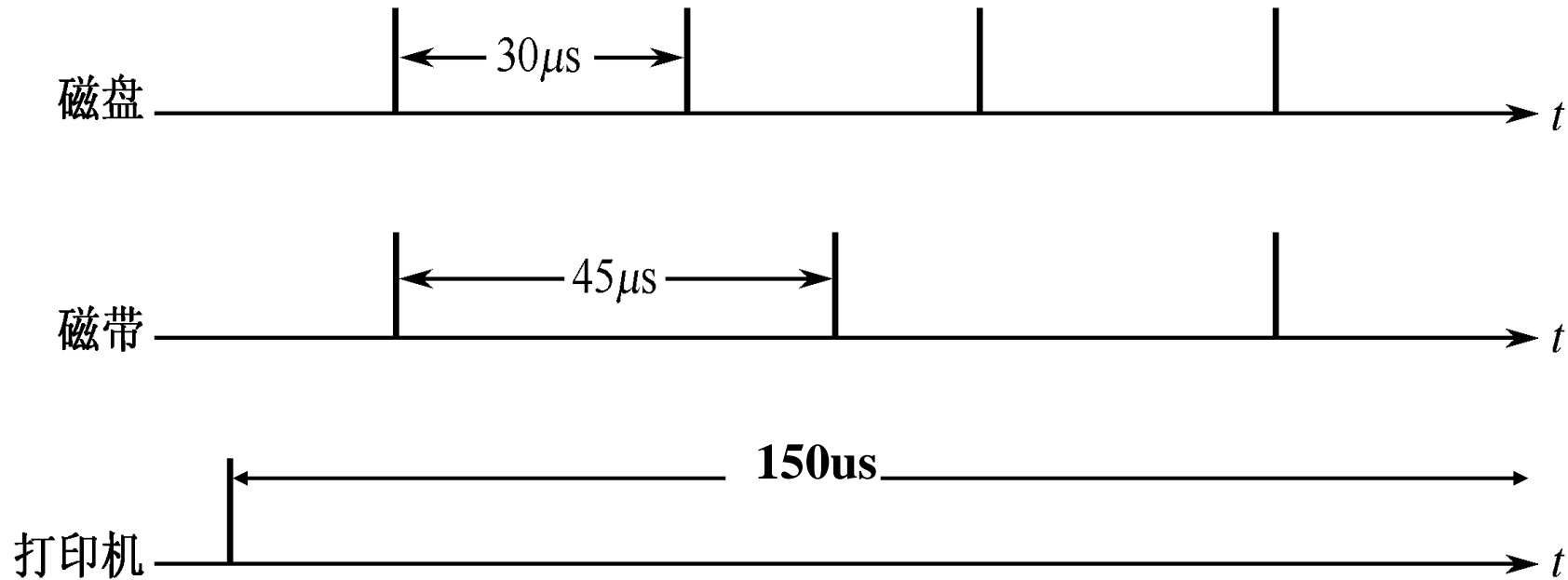
# 例3

下图中假设有磁盘、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。





# 工作时间图





# 分析

由图看出，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)。从图上看到，在这种情况下DMA尚有空闲时间，说明控制器还可以容纳更多设备。





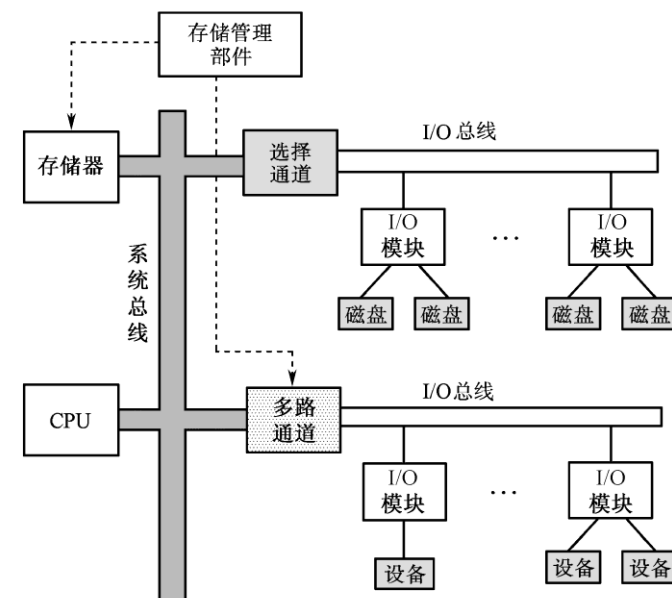
## 8.5 通道方式





# 基本概念

- 通道是计算机系统中代替CPU管理控制外设的独立部件，是一种能**执行有限I/O指令集**（通道命令）的特殊功能的处理机
- 在通道控制方式中，一个主机可以连接几个通道。每个通道又可连接多台I/O设备，这些设备可具有不同速度，可以是不同种类
- 采用通道方式组织输入输出系统，一般采用主机-通道-设备控制器-I/O设备四级连接方式。
- CPU发送一小段通道命令给通道控制器并启动通道，通道自动完成I/O数据操作而无需CPU介入，数据交换结束后向CPU发出中断请求，进行通道结束处理工作。
- 特点：增强了主机与通道操作的并行能力以及各通道之间、同一通道的各设备之间的并行操作能力。同时也为用户提供了增减外围设备的灵活性。





# 通道的功能

- 根据CPU要求选择某一指定外设与系统相连，向该外设发出操作命令，进行初始化
- 指定外设读/写信息的位置、与外设交换信息的主存缓冲区地址
- 控制外设与主存之间的数据交换
- 指定数据传送结束时的操作内容，检查外设的状态





# 通道的类型

## ■ 选择通道

- ◆ 选择一台外设独占整个通道，以成组传送方式传送数据块，效率高，适合快速设备

## ■ 字节多路通道

- ◆ 简单的共享通道，分时处理，面向低、中速字符设备

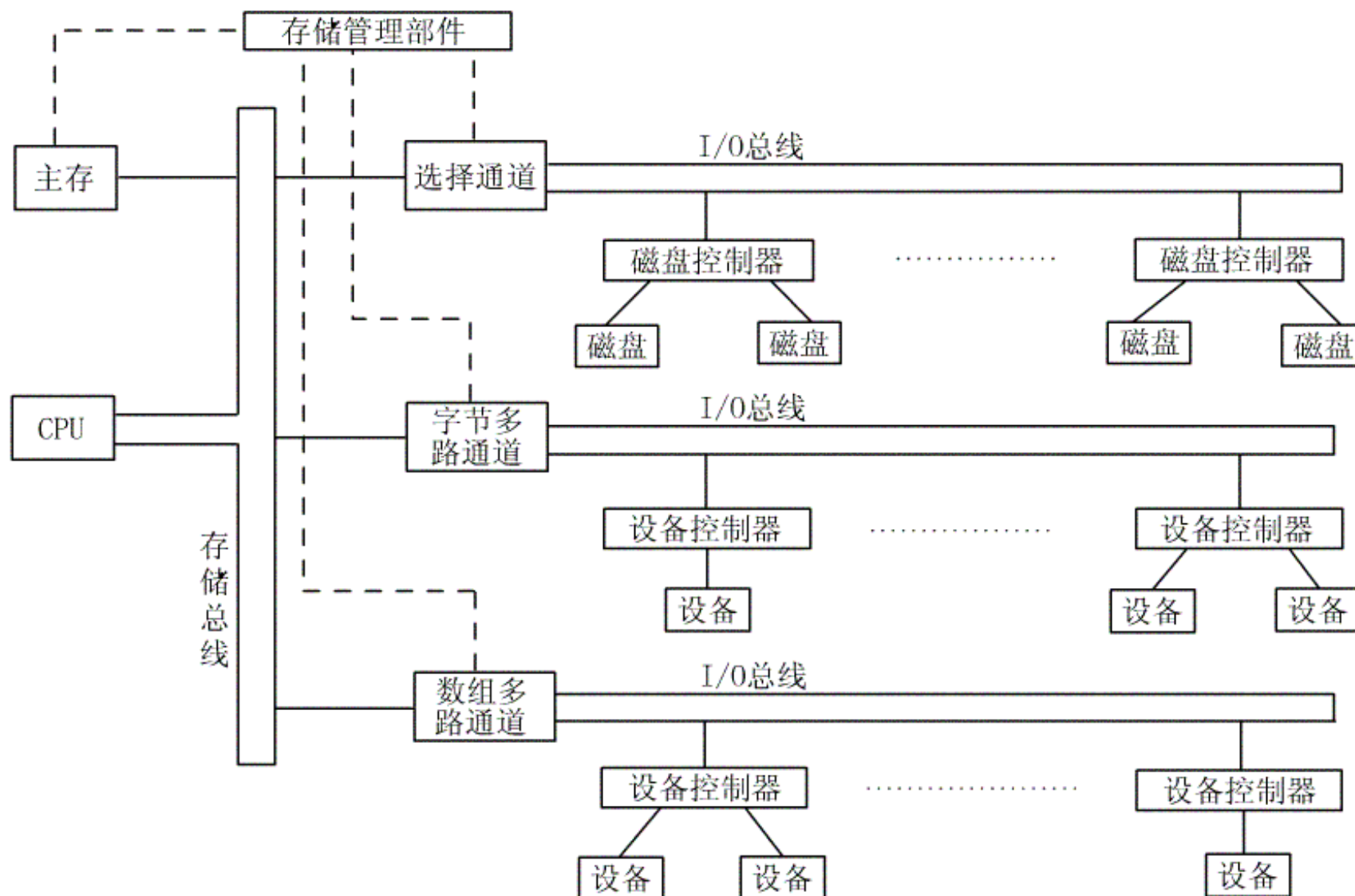
## ■ 数组多路通道

- ◆ 多个设备以数据块为单位交叉使用通道。是上两种方式的结合，效率高，控制较复杂





# IBM 4300系统I/O结构

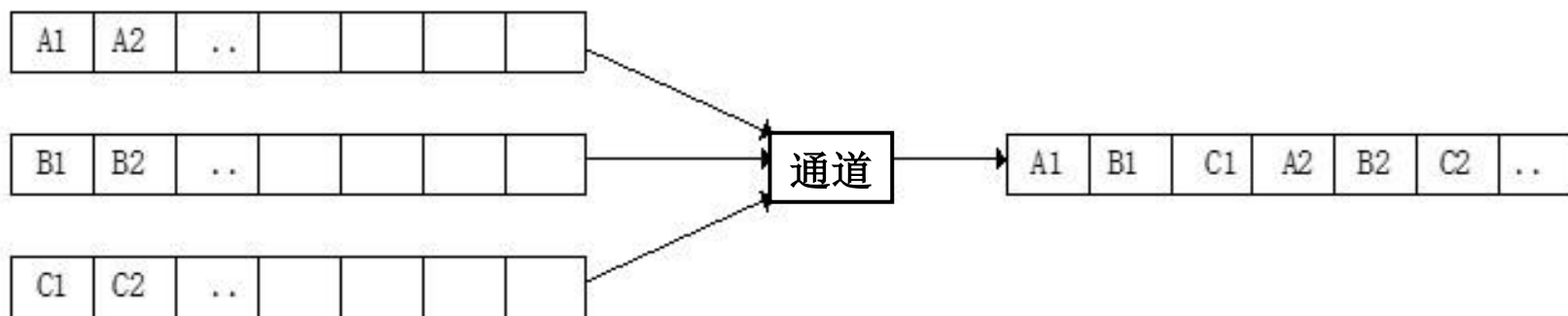






# 选择通道

- 选择通道（也称高速通道）每次只能从所连接的设备中选择一台I / O设备的通道程序执行，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。





# 字节多路通道

- 字节多路通道（Byte Multiplexor Channel）要求每种设备分时占用一个很短的时间片，不同的设备在指定的时间片内与通道建立传输连接，实现数据的传送。适用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。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，送入公共控制部件的寄存器中使用
- 在一段时间内能交替执行多个设备的通道程序





# 数组多路通道

- 数组多路通道（ **Block Multiplexer Channel** ）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
  - ◆ 在一段时间内能交替执行多个设备的通道程序





# 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—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

## ■ 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





# 12

下列选项中，能引起外部中断的事件是

- A. 键盘输入
- B. 除数为0
- C. 浮点运算下溢
- D. 访存缺页





# 本章小结

- 程序查询方式
- 程序中断方式
  - ◆ 中断处理过程、中断源的识别、单级中断、多级中断
- **DMA**方式
  - ◆ **DMA**传送过程
  - ◆ 三种**DMA**传送方式
- 通道方式
  - ◆ 选择通道
  - ◆ 字节多路通道
  - ◆ 数组多路通道

