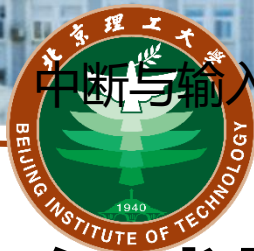




第六部分总线、中断与输入输出系统.ppt



计算机组成与结构 外部设备与输入输出系统

人工智能
主讲教师：王 娟

中央处理器（CPU）和主存储器（MM）构成计算机的主机。除主机以外，而又围绕着主机而设置的各种硬件装置叫做外部设备或外围设备，它们主要用来完成数据的输入、输出、成批存储以及对信息加工处理的任务。

本章将介绍磁介质存储器的存储原理，常用磁介质存储设备和其他辅助存储设备以及常见的输入输出设备的工作原理。

1.磁记录介质和磁头

(1)磁记录介质

在磁介质存储器中，信息是记录在一薄层磁性材料的表面上，这个薄层称为**磁层**。磁层与所附着的载体被称为记录介质或记录媒体。

(2)磁头

磁头是磁记录设备的关键部件之一，它是一种**电磁转换**元件。写磁头实现把电脉冲表示的二进制代码转换成磁记录介质上的磁化状态，即电 - 磁转换；读磁头实现把磁记录介质上的磁化状态转换成电脉冲，即磁 - 电转换。

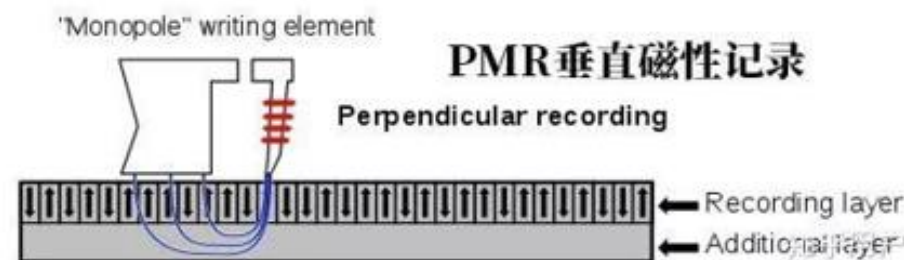
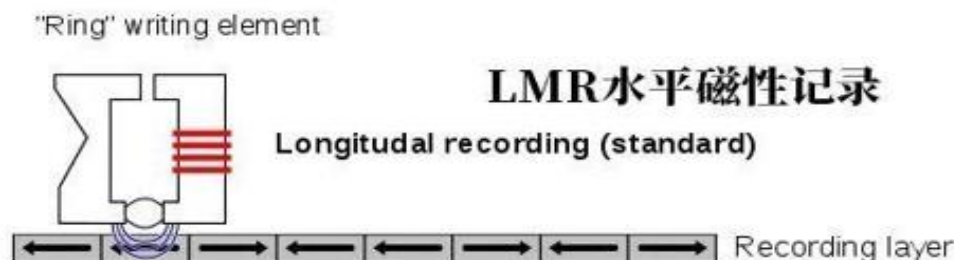
在读写过程中，记录介质与磁头之间相对运动，一般是记录介质运动而磁头不动。

✓ 读出数据

线圈中不加电流，磁层移动。当位单元的转变区经过磁头下方时，线圈两端会产生感应电势 e ，形成电流：**磁 - 电转换**。

✓ 数据写入

磁头线圈中加磁化电流（写入电流），磁层移动，形成连续的小段磁化区（位单元区）：**电 - 磁转换**。



磁记录方式是一种编码方式，即按照某种规律将一连串的二进制数字信息变换成记录介质上相应磁化翻转形式。

1.直接记录方式

当记录密度较低时，可以不编码，直接按记录信息的“0”、“1”排序记录。这类方式有：

(1)归零制（RZ）

记录“1”时，磁头线圈中通以正向脉冲电流；记录“0”时，通以反向脉冲电流。由于脉冲电流均要回到零，故称为归零制。归零制的两个脉冲之间有一段间隔没有电流，相应的这段磁层未被磁化。



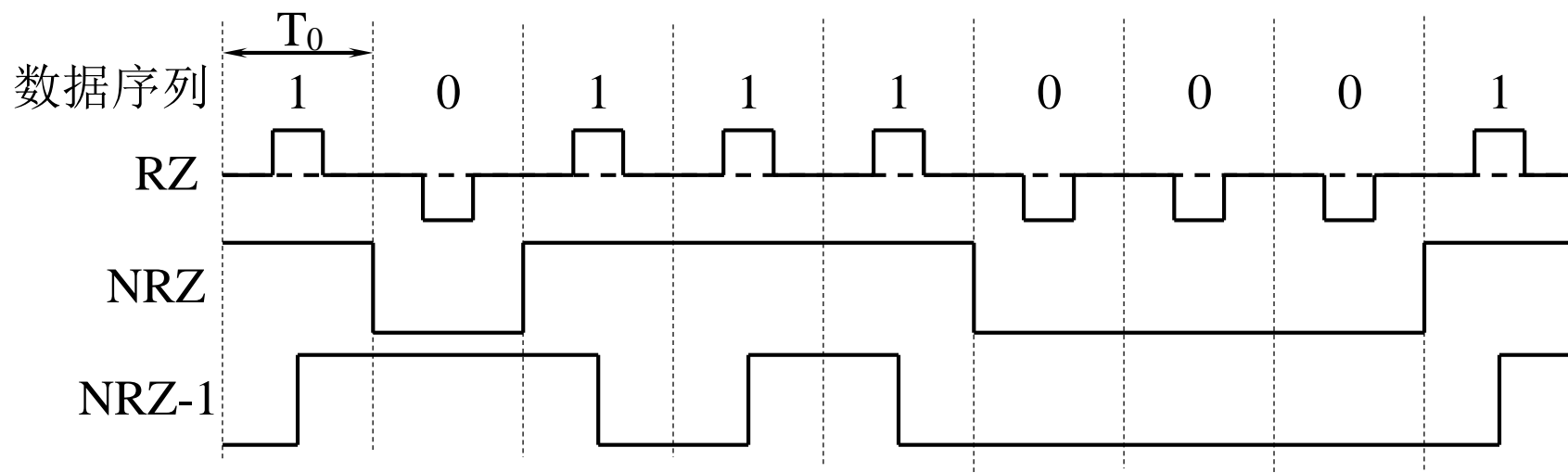
(2)不归零制 (NRZ)

记录“1”时，磁头线圈中通正向电流；记录“0”时，通以反向电流。磁头中电流不回到零。如果记录的相邻两位信息相同（即连续记录“1”或“0”）时，写电流方向不变；只有当记录的相邻两位信息不相同（即“0”和“1”交替）时，写电流才改变方向，所以又称为异码变化或“见变就翻”的不归零制。

(3)不归零 - 1制 (NRZ-1)

这是一种改进的不归零制，记录“1”时，磁头线圈中写电流改变方向，使磁层磁化翻转；而记录“0”时，写电流方向维持不变，保持原来的磁化状态，所以称之为见“1”就翻的不归零制。

8.2 磁介质存储器的性能和原理



以上各种记录方式，目前已很少应用，但不归零制是编码方式的基础，无论哪一种编码方式，只要数据序列变换成记录序列之后，均按照NRZ-1制规则记录到磁层上。

2.按位编码记录方式

(1)调相制 (PE)

调相制又称相位编码方式，它采用 0° 和 180° 相位的不同分别表示“1”或“0”。它的编码规则是：记录“1”时，写电流在位周期中间由负变正；记录“0”时，写电流在位周期中间由正变负。当连续出现两个或两个以上“1”或“0”时，为了维持上述原则，在位周期的边界上也要翻转一次。这种记录方式常用于磁带机中。

(2)调频制 (FM)

调频制是根据写电流的频率来区分记录“1”或“0”的。记录“1”时，写电流在位周期中间和边界各改变一次方向；记录“0”时，写电流仅在位周期边界改变一次方向。因此，记录“1”的磁化翻转频率为记录“0”时的两倍，故又称倍频制。若以 T_0 表示位周期，则调频制的磁化翻转间距为 $0.5T_0$ 和 T_0 。

(3)改进的调频制 (MFM)

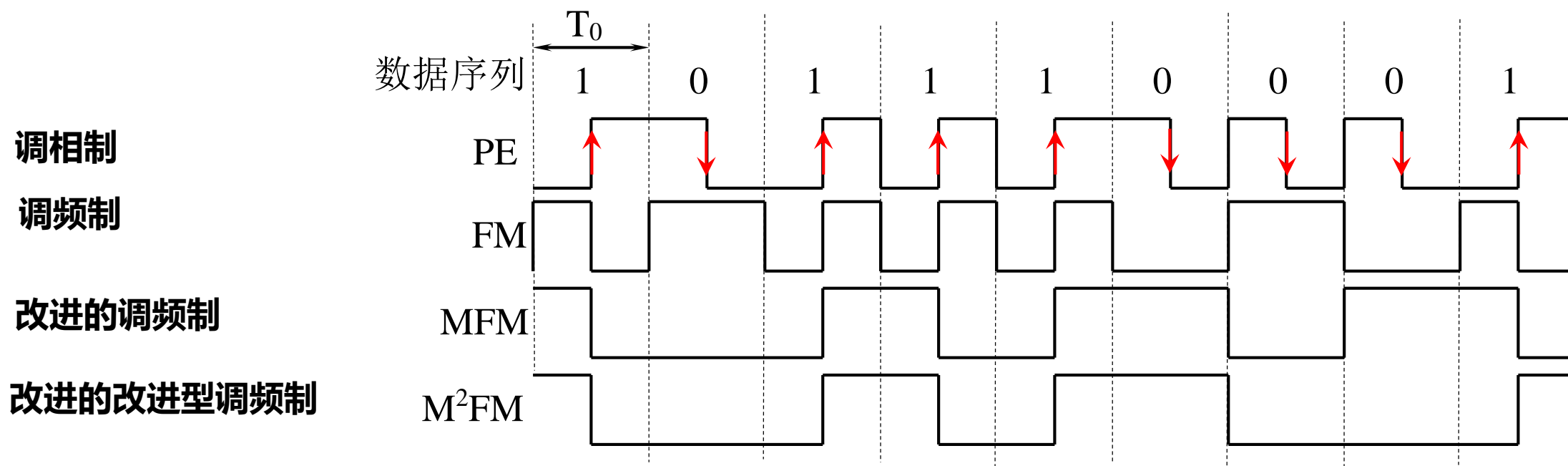
MFM制是在FM制基础上改进的一种记录方式，又称为延迟调制码或密勒码。其编码规则为：记录“1”时，写电流在位周期中间改变方向；记录独立的一个“0”，写电流不改变方向；记录连续的两个“0”，写电流在位周期边界改变方向。

改进的调频制的磁化翻转间距有三种： T_0 、 $1.5T_0$ 、 $2T_0$ ，对应于三种不同的频率，所以又称为三频制。

(4)改进的改进型调频制 (M^2FM)

M^2FM 制的编码规则为：记录“1”时，写电流在位周期中间改变方向；记录独立的一个“0”，写电流不改变方向；记录连续的两个“0”，写电流在位周期边界处改变方向，产生磁化翻转；记录连续两个以上的“0”，写电流在前两个“0”的位周期边界处改变方向，以后每隔两个“0”的位周期边界处，写电流再改变一次方向，产生翻转翻转。

改进的改进型调频制的磁化翻转间距有四种： T_0 、 $1.5T_0$ 、 $2T_0$ 、 $2.5T_0$ ，对应于四种不同的频率，所以又称为四频制。



硬盘存储器

磁盘控制器 + 接口 → 磁盘适配器

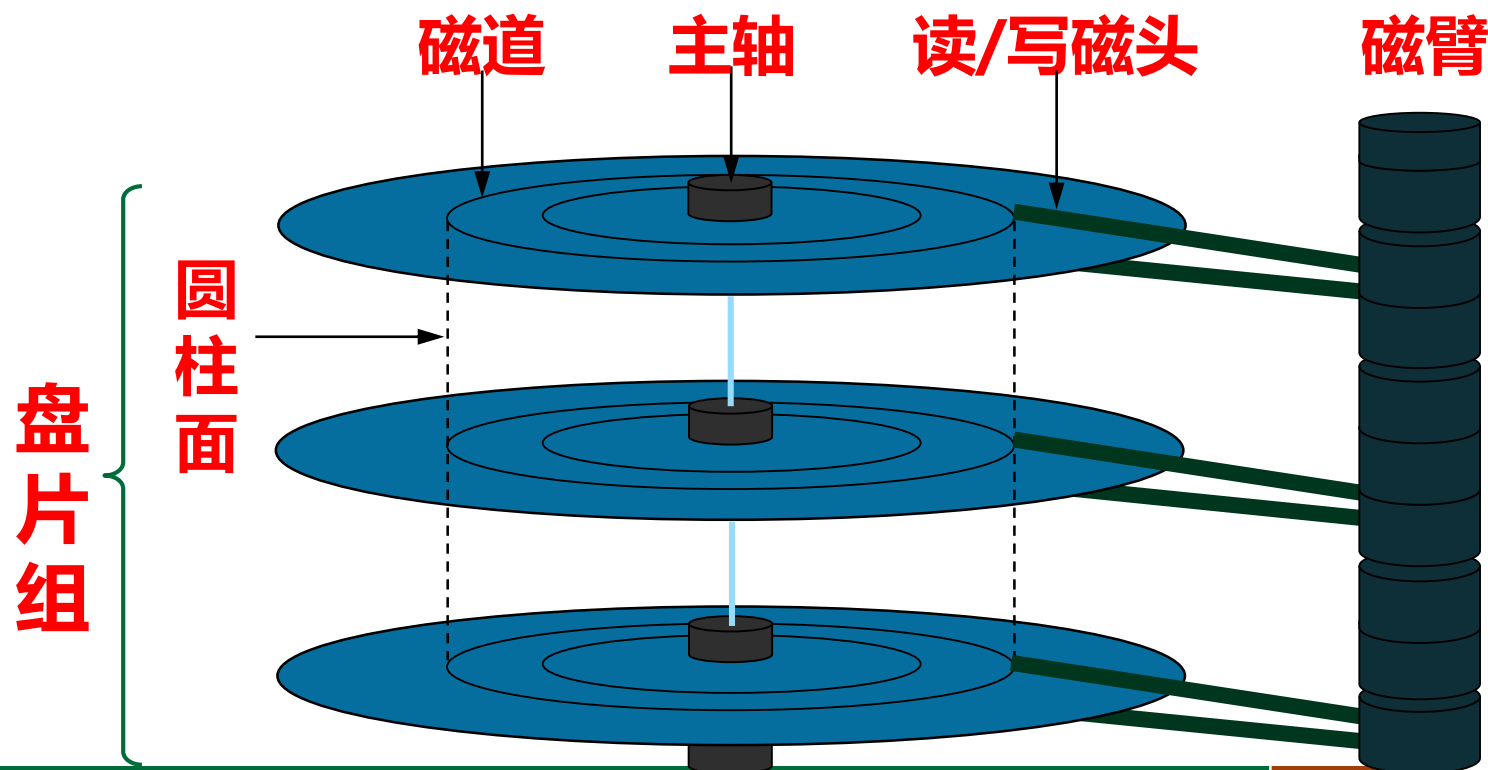
硬盘

磁盘驱动器：定位系统、传动系统、磁头

盘片：存储体

硬盘分类：

- 温彻斯特盘
- 非温彻斯特盘
- 固定头硬盘
- 活动头硬盘



在硬盘中信息分布呈如下层次：记录面、圆柱面、磁道、扇区。

(1)记录面

硬盘驱动器中可有多片盘片（数量为1~20片），每个盘片有两个记录面，每个记录面对应一个磁头。

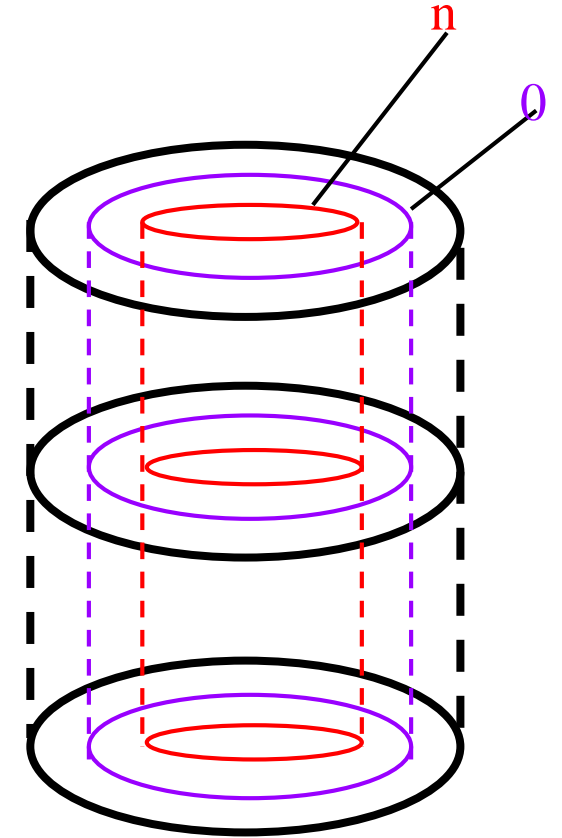
(2)磁道

在读/写时，磁头固定不动，盘片高速旋转，磁化区构成一个闭合圆环，称为磁道。在盘面上，一条条磁道形成一组同心圆，最外圈的磁道为0号，往内则磁道号逐步增加。（每个盘片可有500~2500条磁道）。

(3)圆柱面

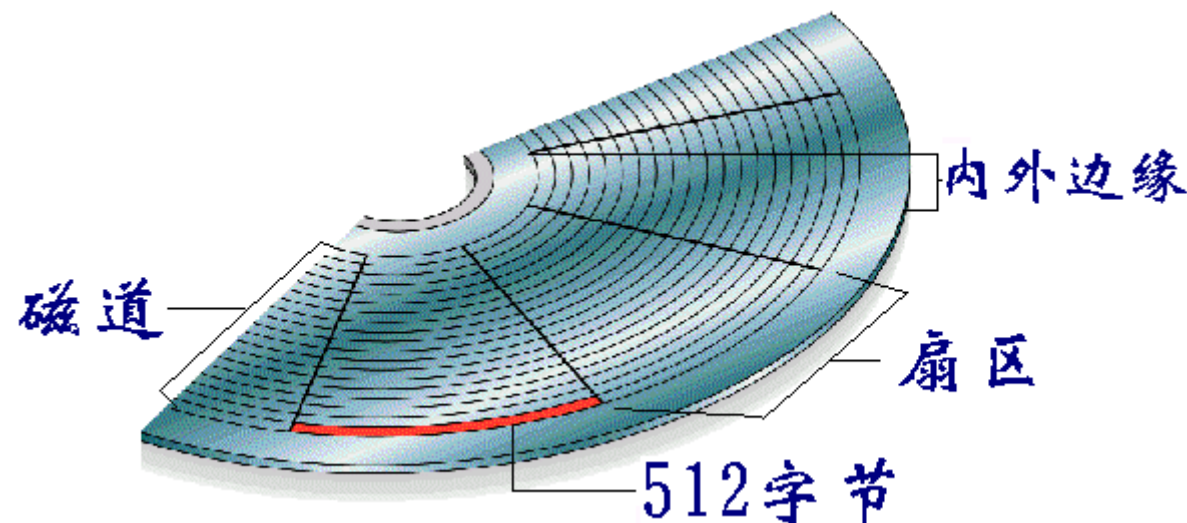
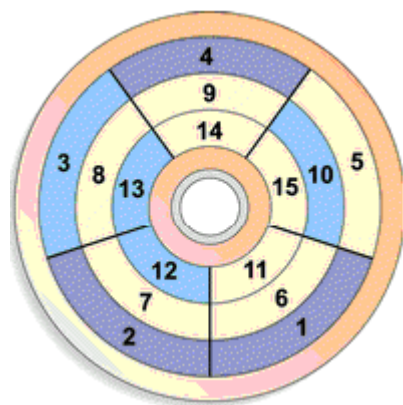
在一个盘组中，各记录面上相同编号（位置）的诸磁道构成一个圆柱面。例如，某驱动器有4片8面，则8个0号磁道构成0号圆柱面，8个1号磁道构成1号圆柱面，……。硬盘的圆柱面数等于一个记录面上的磁道数，圆柱面号即对应的磁道号。

思考：存入一个较长的文件时，若一条磁道存不完，就需要存放在几条磁道上。这时主机应选择位于同一记录面上的几条磁道？还是选择同一圆柱面上的几条磁道呢？



(4) 扇区

通常将一条磁道划分为若干个段，每个段称为一个扇区或扇段，每个扇区存放一个定长信息块（如512个字节）。一条磁道划分多少扇区，每个扇区可存放多少字节，一般由操作系统决定。磁道上的扇区编号从1号开始。



磁盘地址一般表示为：

驱动器号、圆柱面（磁道）号、记录面（磁头）号、扇区号

1.记录密度

记录密度是指磁介质存储器单位长度或单位面积磁层表面所能存储的**二进制信息量**。通常以道密度和位密度表示，也可用两者的乘积面密度来表示。

(1)道密度

道密度又叫横向密度，是指**垂直于磁道方向上单位长度中的磁道数目**，道密度的单位是道/英寸（TPI）或道/毫米（TPM）。

磁道指的是磁头写入磁场在记录介质表面上形成的磁化轨迹。

(2)位密度

位密度又叫纵向密度，是指**沿磁道方向上单位长度中所能记录的二进制信息的位数**，位密度的单位为位/英寸（bpi）或位/毫米（bpm）。

2. 存储容量

存储容量是指整个磁介质存储器所能存储的二进制信息的总量，一般用位或字节为单位表示，它与存储介质尺寸和记录密度直接相关。

磁介质存储器的存储容量有非格式化容量和格式化容量两种指标。非格式化容量是指磁记录表面上可全部利用的磁化单元数；**格式化容量是指用户实际可以使用的存储容量**。格式化容量一般约为非格式化容量的60~80%左右。

以机械硬盘为例：

非格式化容量 = 最大位密度 × 最内圈磁道周长 × 总磁道数

格式化容量 = 每道扇区数 × 扇区容量 × 总磁道数

3.平均存取时间

当磁头接到读/写命令，从原来的位置移动到指定位置，并完成读/写操作的时间叫存取时间。对于磁盘存储器来说，存取时间主要包括4部分：第一部分是指磁头从原先位置移动到目的磁道所需要的时间，称为**定位时间或寻道时间**；第二部分是指在到达目的磁道以后，等待被访问的记录块旋转到磁头下方的等待时间，称为**旋转时间或等待时间**；第三部分是信息的读写操作时间，也称为传输时间；最后是磁盘控制器的开销。

$$T_a = T_s + T_w = \frac{T_{s\max} + T_{s\min}}{2} + \frac{T_{w\max} + T_{w\min}}{2}$$



4.数据传送率

磁介质存储器在单位时间内向主机传送数据的位数或字节数，称为数据传送率 D_r ，单位为位/秒或字节/秒。

5.误码率

误码率是衡量磁介质存储器出错概率的参数，它等于读出的出错信息位数和读出总的信息位数之比。



第八章8.4~8.8节作为自主学习任务。

同时：请根据自己学号最后一位做模5运算，结果对应的专题如下

0对应：磁盘阵列

1对应：光盘存储器

2对应：键盘输入设备

3对应：打印输出设备

4对应：显示设备

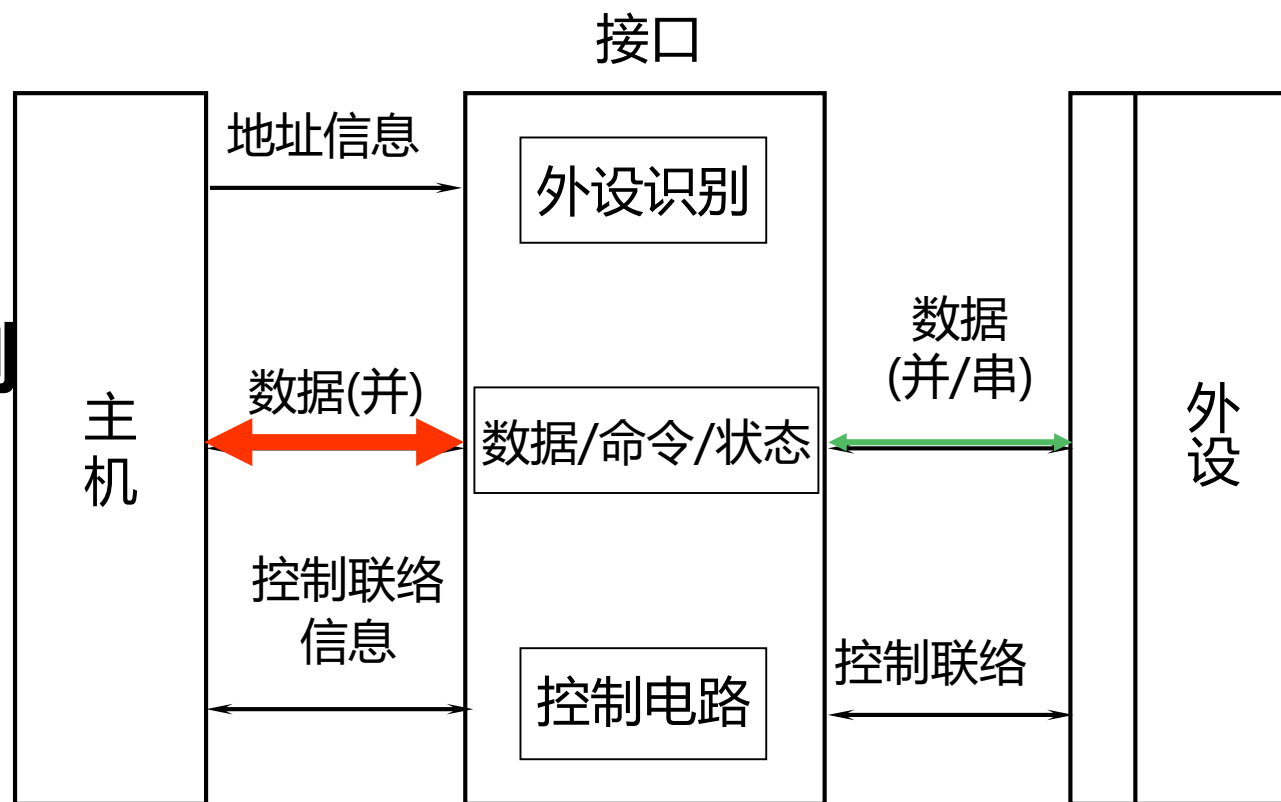
请各位依据结果准备专题汇报文档，调研专题最新发展情况（至少1页A纸，不超过6页），作为一次作业提交。

输入输出接口

主机和外设的连接方式有辐射型连接、总线型连接等。输入/输出接口（I/O接口）是主机和外设之间的交接界面，通过**接口**可以实现主机和外设之间的信息交换。

接口主要作用：

- (1) 实现主机和外设的通信联络控制
- (2) 进行地址译码和设备选择
- (3) 实现数据缓冲
- (4) 数据格式的变换
- (5) 传递控制命令和状态信息





端口是指接口电路中可以进行读/写的寄存器，若干个端口加上相应的控制逻辑电路才组成接口。

一个接口中包含有数据端口、控制端口和状态端口。有的接口电路中状态信息和控制信息共用一个寄存器，称之为设备的控制状态寄存器。

主机和外设之间需要交换的信息

- 1. 数据信息:** 这类信息可以通过输入设备送到计算机的输入数据，也可以是经过计算机运算处理和加工后，送到输出设备的结果数据。传送可以是并行的，也可以是串行的。
- 2. 控制信息:** CPU对外设的控制信息或管理命令。
- 3. 状态信息:** 标志外设的工作状态。
- 4. 联络信息:** 主机和外设间工作的时间配合信息。



接口的类型

- (1) 按数据传送方式分类 串行接口和并行接口。
- (2) 按主机访问I/O设备的控制方式分类
 程序查询式接口(PIO)、中断接口、DMA接口等。
- (3) 按功能选择的灵活性分类 可编程接口和不可编程接口。
- (4) 按通用性分类 通用接口和专用接口。
- (5) 按输入/输出的信号分类 有数字接口和模拟接口。
- (6) 按应用来分类
 - ① 运行辅助接口。
 - ② 用户交互接口。
 - ③ 传感接口。
 - ④ 控制接口。

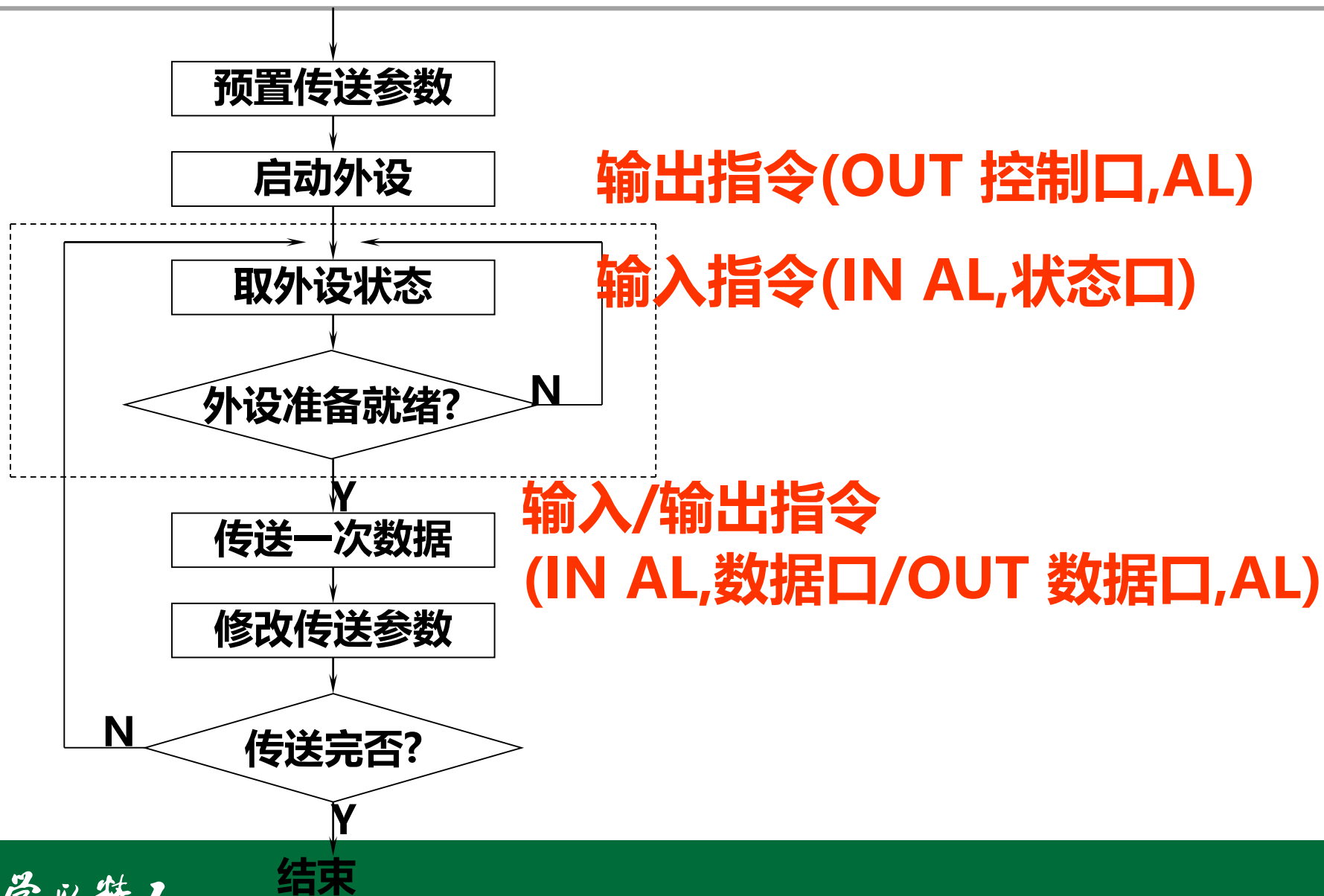
1. 程序查询方式

程序查询方式是最简单、经济的I/O方式，只需很少的硬件，输入和输出完全是通过CPU执行程序来完成的。

通常接口中**至少有两个寄存器**，一个是数据缓冲寄存器，即数据端口，用来存放与CPU进行传送的数据信息，另一个是供CPU查询的设备状态寄存器，即状态端口，这个寄存器由多个标志位组成，其中最重要的是设备准备就绪标志。当CPU得到这位信息后就进行判断，以决定下一步是继续循环等待还是进行I/O传送，也有些计算机仅设置状态标志触发器，其作用与设备状态寄存器相同。

这种方式控制简单，但外设和主机不能同时工作，各外设之间也不能同时工作，系统效率很低，因此，仅适用于外设的数目不多，对I/O处理的实时要求不那么高，CPU的操作任务比较单一，并不很忙的情况。

程序查询方式流程图



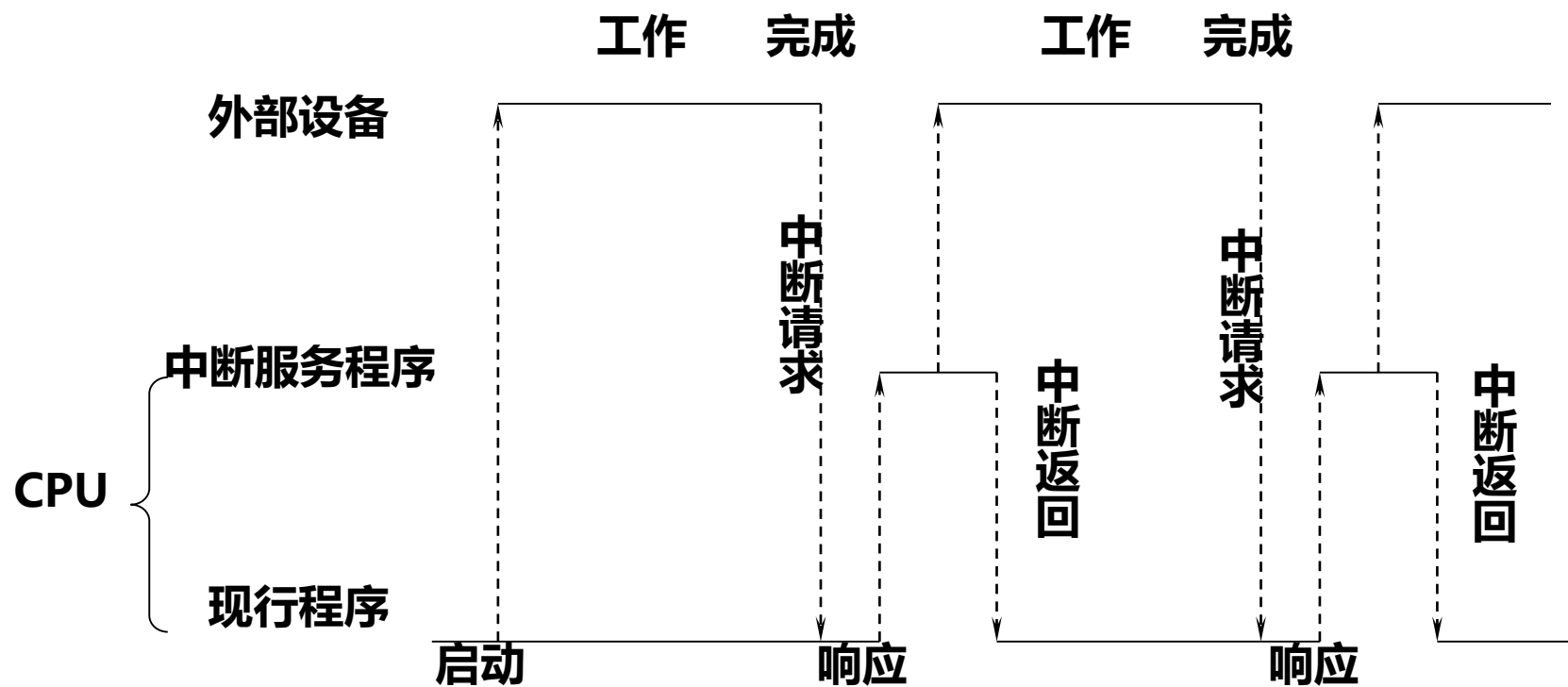


2、中断方式

程序查询方式存在着下列明显的缺点。

- ① 在查询过程中，CPU长期处于踏步等待状态，使系统效率大大降低。
- ② CPU在一段时间内只能和一台外设交换信息，其它设备不能同时工作。
- ③ 不能发现和处理预先无法估计的错误和异常情况。

现代计算机均具有中断能力。中断系统是计算机实现中断功能的软、硬件总称。一般在CPU中配置中断机构，在外设接口中配置中断控制器，在软件上设计相应的中断服务程序。



程序中中断与调用子程序的区别



表面上看起来，计算机的中断处理过程有点类似于调用子程序的过程，这里现行程序相当于主程序，中断服务程序相当于子程序。但是，它们之间却是有着本质上的区别的。

- (1)子程序的执行是由程序员事先安排好的（由一条调用子程序指令转入），而中断服务程序的执行则是由随机的中断事件引起的；**
- (2)子程序的执行受到主程序或上层子程序的控制，而中断服务程序一般与被中断的现行程序毫无关系；**
- (3)不存在同时调用多个子程序的情况，而有可能发生多个外设同时请求CPU为自己服务的情况。**



中断基本类型

(1) 自愿中断和强迫中断

自愿中断又称程序自中断，它不是随机产生的中断，而是在程序中安排的有关指令，这些指令可以使机器进入中断处理的过程，如：指令系统中的软件中断指令等。

强迫中断是随机产生的中断，不是程序中事先安排好的。当这种中断产生后，由中断系统强迫计算机中止现行程序并转入中断服务程序。

(2) 程序中断和简单中断

程序中断：主机在响应中断请求后，通过执行一段中断服务程序来处理更紧迫的任务。

简单中断就是DMA方式，不去执行中断服务程序，不破坏现行程序的状态。

(3) 内中断和外中断

内中断是指由于CPU内部硬件或软件原因引起的中断。

外中断是指CPU以外的部件引起的中断。

(4) 向量中断和非向量中断

向量中断是指那些中断服务程序的入口地址，即向量地址，是由中断事件自己提供的中断。**非向量中断**的中断事件不能直接提供中断服务程序的入口地址，而由CPU 查询之后得到。

(5) 单重中断和多重中断

单重中断在CPU执行中断服务程序的过程中不能被再打断。

多重中断在执行某个中断服务程序的过程中，CPU 可去响应级别更高的中断请求，又称为中断嵌套。

1. 中断源和中断请求信号

中断源是指中断的来源，即任何引起计算机中断的事件，一般计算机都有多个中断源。由于每个中断源向CPU发出中断请求的时间是随机的，为了记录中断事件并区分不同的中断源，可采用具有存储功能的触发器来记录中断源，称为中断请求触发器。当某一个中断源有中断请求时，其相应的中断请求触发器置成“1”状态，此时，该中断源向CPU发出中断请求信号。多个中断请求触发器构成一个中断请求寄存器，其中每一位对应一个中断源，中断请求寄存器的内容称为**中断字或中断码**，中断字中为“1”的位就表示对应的中断源有中断请求。

(1) 独立请求线

每个中断源单独设置中断请求线，将中断请求信号直接送往CPU，这种方式的特点是CPU在接到中断请求的同时也就知道了中断源是谁，其中断服务程序的入口地址在哪里。

(2) 公共请求线

多个中断源共有一根公共请求线，这种方式的特点是在负载允许的情况下，中断源的数目可随意扩充，但CPU在接到中断请求后，必须通过软件或硬件的方法来识别中断源，然后再找出中断服务程序的入口地址。

(3) 二维结构

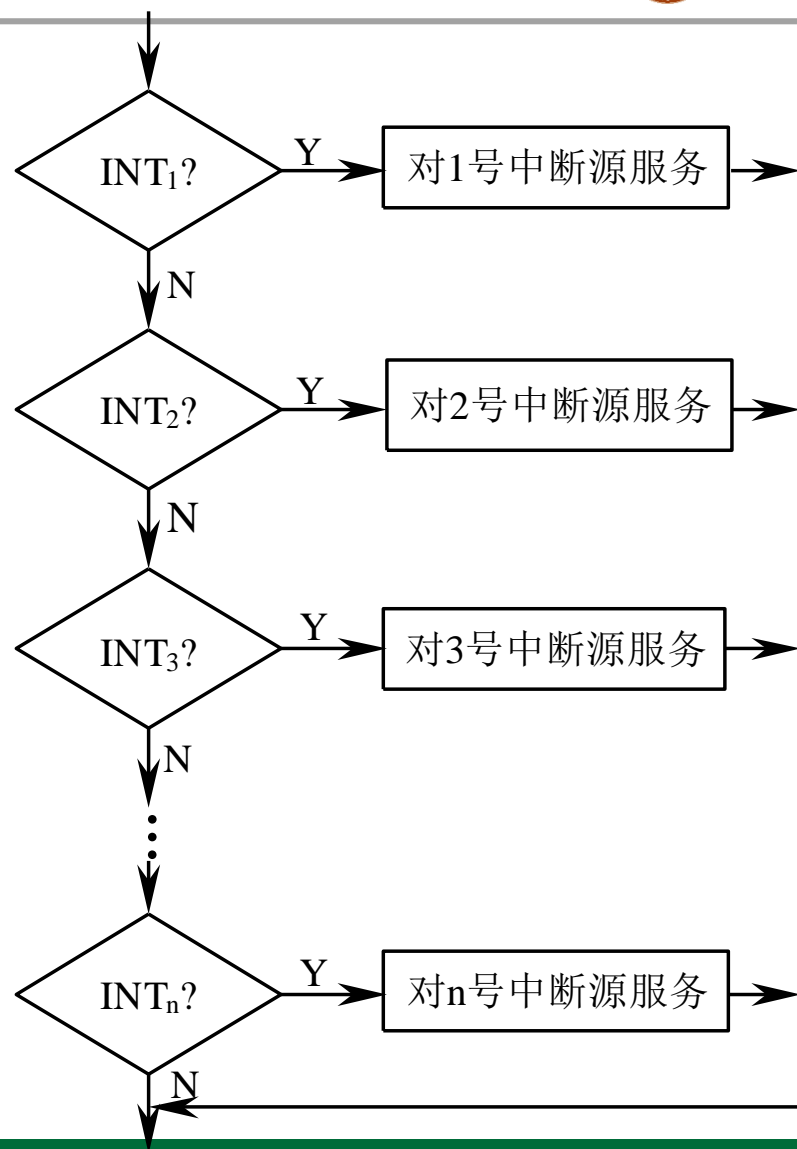
将中断请求线连成二维结构，同一优先级别的中断源，采用一根公共的请求线，不同请求线上的中断源优先级别不同，这种方式综合了前两种方式的优点，在中断源较多的系统中常采用这种方式。

当多个中断源同时发出中断请求时，CPU在任何瞬间只能接受一个中断源的全部请求。通常，把全部中断源按中断的性质和处理的轻重缓急安排优先级，并进行排队。

(1) 软件判优法

软件判优法，就是用程序来判别优先级，这是最简单的中断判优方法。

最先检测的中断源具有最高的优先级。





(2) 硬件判优电路

采用硬件实现中断优先级判定可节省CPU时间，而且速度快，但是成本较高。

根据中断请求信号的传送方式不同，有不同的优先排队电路，常见的有以下几种方案。

独立请求线的优先排队电路

公共请求线的优先排队电路

(1) CPU接收到中断请求信号

首先中断源要发出中断请求，同时CPU还要接收到这个中断请求信号。

(2) CPU允许中断

CPU允许中断即开中断。CPU内部有一个中断允许触发器，只有当其被置位时，CPU才可能响应中断源的中断请求（中断开放）。如其被复位，CPU处于不可中断状态，即使中断源有中断请求，CPU也不响应（中断关闭）。

(3) 一条指令执行完毕

一般情况下，CPU在一条指令执行完毕，且没有更紧迫的任务时才能响应中断请求。

CPU响应中断之后，经过某些操作，转去执行中断服务程序。这些操作是由硬件直接实现的，我们把它称为中断隐指令。**中断隐指令并不是指令系统中的一条真正的指令，它没有操作码，所以中断隐指令是一种不允许、也不可能为用户使用的特殊指令。**其所完成的操作主要有：

- (1) **保存断点** 将原来程序的断点（即程序计数器PC的内容）保存起来。
- (2) **暂不允许中断**

为了在用软件保护中断现场（即CPU的主要寄存器状态）时，不被新的中断所打断，从而保证被中断的程序在中断服务程序执行完毕之后能接着正确地执行下去。

- (3) **引出中断服务程序**

实质就是取出中断服务程序的入口地址送程序计数器。



假定断点保存在0号单元，中断周期需完成如下操作：

- (1)将特定地址 “0”送至存储器地址寄存器，记作 $0 \rightarrow \text{MAR}$ ；**
- (2)将PC的内容（断点）送至MDR，记作 $(\text{PC}) \rightarrow \text{MDR}$ ；**
- (3)向主存发写命令，启动存储器做写操作，记作Write；**
- (4)将MDR的内容通过数据总线写入到MAR所指示的主存单元（0号）中，记作 $\text{MDR} \rightarrow \text{M}(\text{MAR})$ ；**
- (5)向量地址形成部件的输出送至PC，为进入中断服务程序作准备，记作向量地址 $\rightarrow \text{PC}$ ；**
- (6)关中断，将中断允许触发器清0，记作 $0 \rightarrow \text{EINT}$ 。**

如果断点存入堆栈，只需将上述(1)改为堆栈指针 $\text{SP} \rightarrow \text{MAR}$ 。

(1) 向量地址是中断服务程序的入口地址

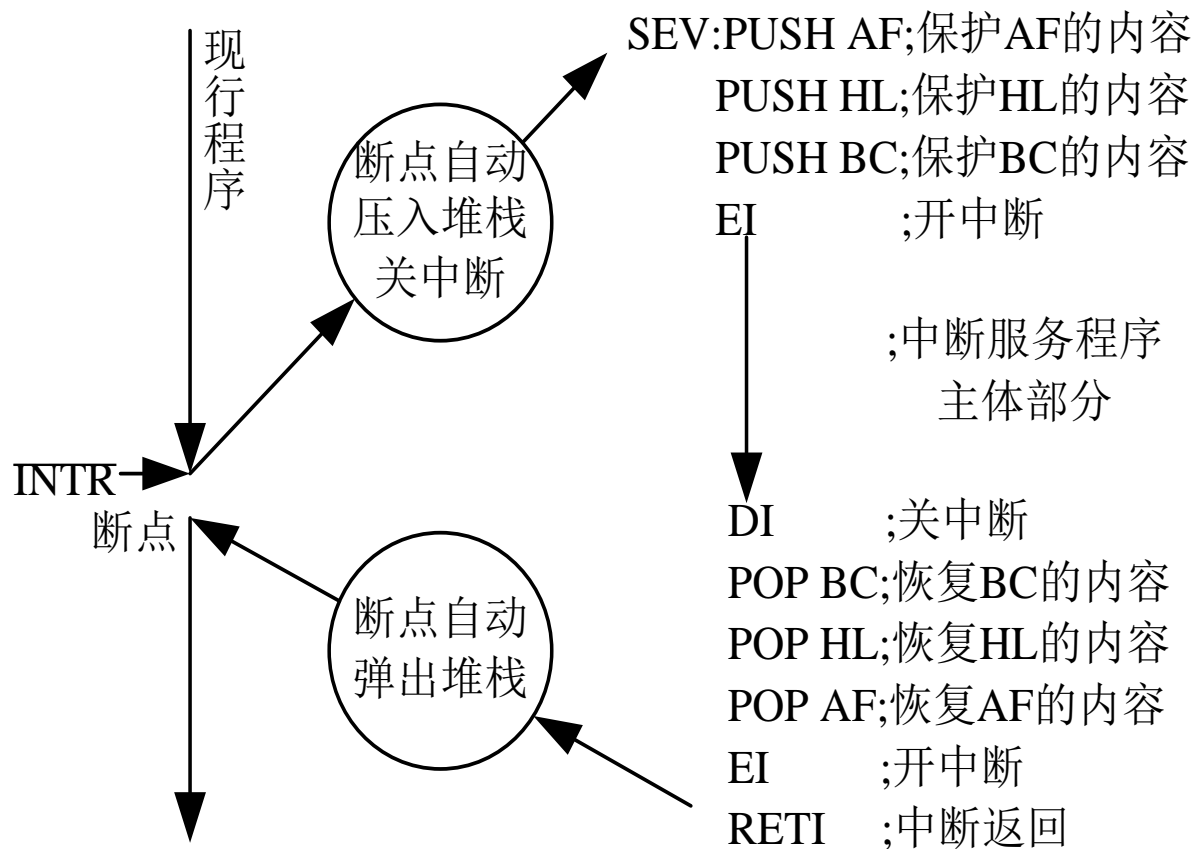
如果向量地址就是中断服务程序的入口地址，则CPU 不需要再经过处理就可以进入相应的中断服务程序。

$PC \leftarrow 8 \times NNN$ 转中断服务程序入口地址

由此可见，中断服务程序的入口地址依次是00H、08H、10H、.....、38H。

(2) 向量地址是中断向量表的指针

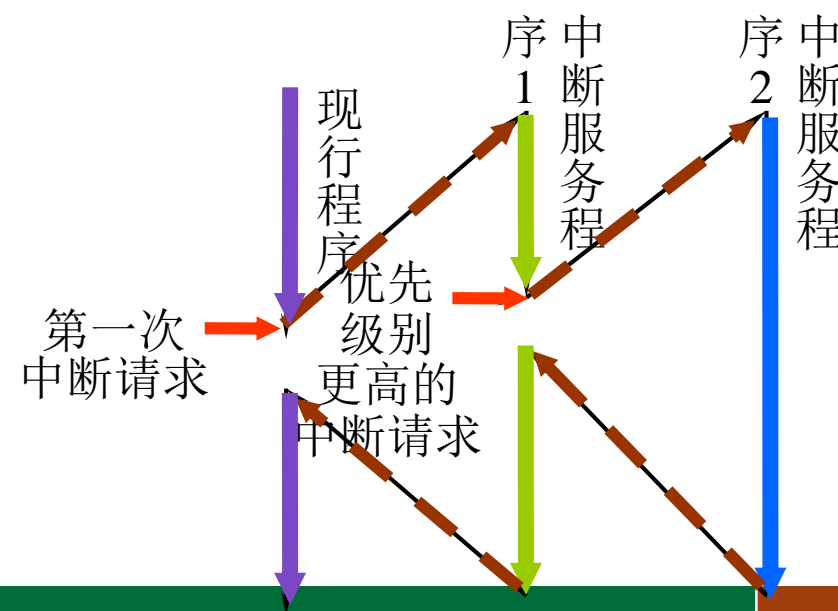
如果向量地址是中断向量表的指针，则向量地址指向一个中断向量表，从中断向量表的相应单元中再取出中断服务程序的入口地址，此时中断源给出的向量地址是中断服务程序入口地址的地址。



基于堆栈的中断现场的保护和恢复，支持中断嵌套。

思考：最外层的中断服务程序优先级高还是最内层的优先级高？

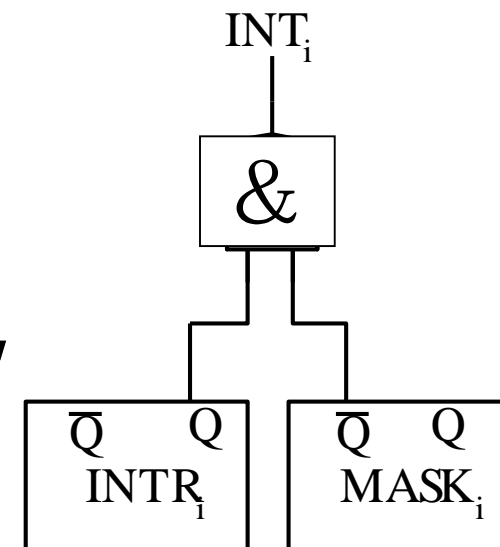
思考：允许与禁止中断的时机？

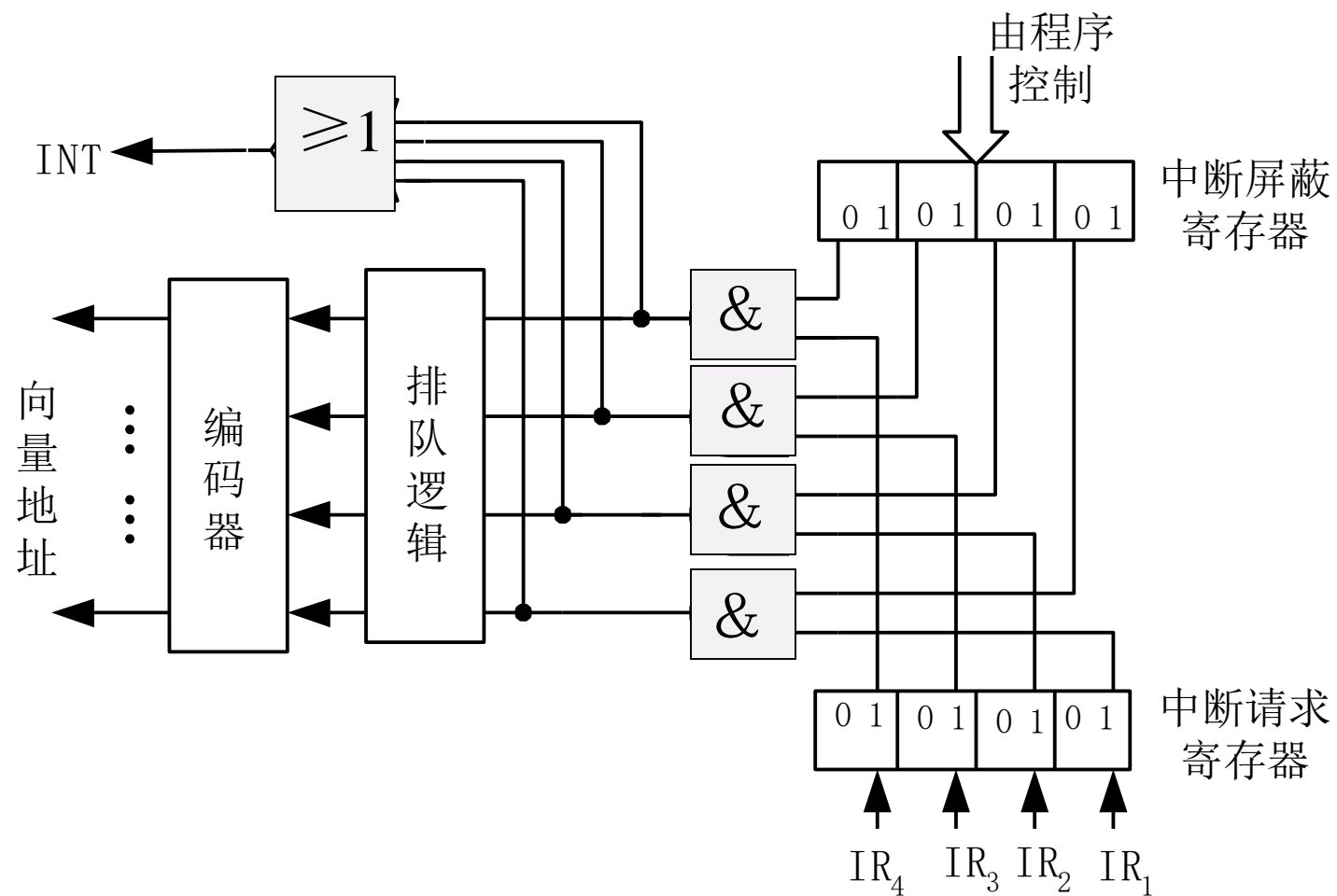


中断源发出中断请求之后，这个中断请求并不一定能真正送到CPU去，在有些情况下，可以用程序方式有选择地封锁部分中断，这就是中断屏蔽。

中断屏蔽寄存器的作用：用程序设置的方法将屏蔽寄存器中的某一位位置“1”，则对应的中断请求被封锁，无法去参加排队判优；若屏蔽寄存器中的某一位位置“0”，才允许对应的中断请求送往CPU。

中断屏蔽字可以改变中断的处理次序，实现**中断软升级**。中断处理次序和中断响应次序是两个不同的概念，**中断响应次序是由硬件排队电路决定的，无法改变。但是，中断处理次序是可以由屏蔽码来改变的，**故把屏蔽码看成软排队器。中断处理次序可以不同于中断响应次序。





某计算机的中断系统有4个中断源，每个中断源对应一个屏蔽码。中断响应的优先次序为1→2→3→4。中断的处理次序和中断的响应次序是一致的。

| 程序级别 | 屏蔽码 | | | |
|------|-----|----|----|----|
| | 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 |

中条写变，断为
变的改改序中改
改序过以次使
不次通可理要
在应，码处，次
响下蔽断如理
断件屏中例处
1→4→3→2。

| 程序级 别 | 屏 蔽 码 | | | |
|----------|-------|----|----|----|
| | 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 |

中断全过程



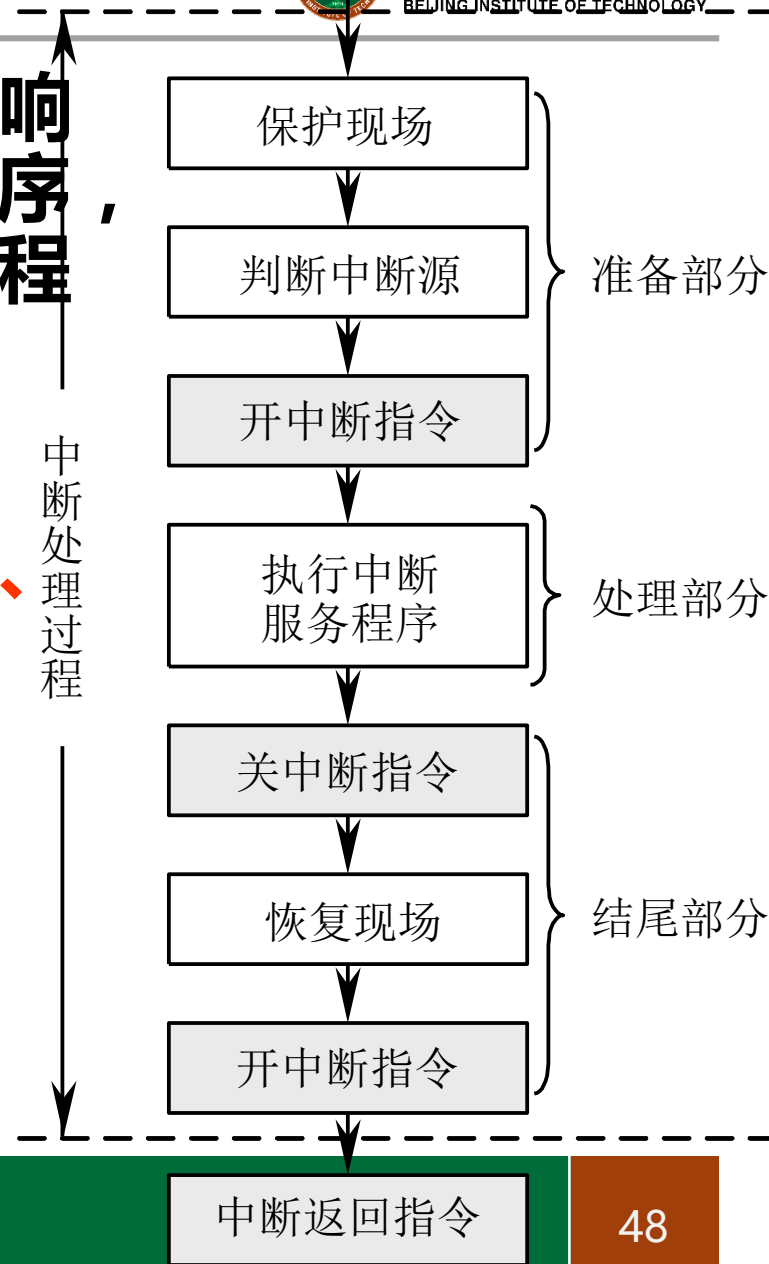
北京理工大学
BEIJING INSTITUTE OF TECHNOLOGY

中断全过程是指从中断源发出中断请求开始，CPU响应这个请求，现行程序被中断，转至中断服务程序，直至中断服务程序执行完毕，CPU再返回原来的程序继续执行的整个过程。

中断全过程分为五个阶段：

中断请求、中断判优、中断响应、中断处理、中断返回。

中断处理就是执行中断服务程序。



无论程序查询还是程序中断方式，主要的工作都是由CPU执行程序完成的，这需要花费时间，因此不能实现高速外设与主机的信息交换。

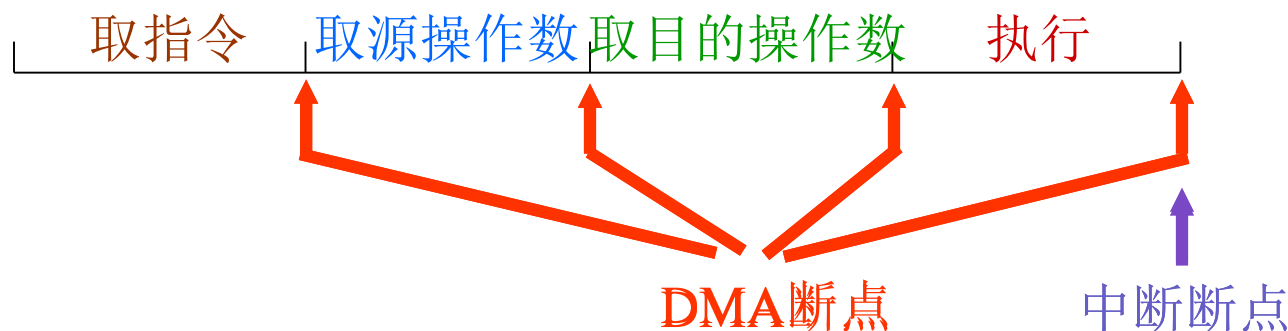
直接存储器访问DMA方式是在**外设和主存储器之间**开辟一条“**直接数据通道**”，在不需CPU 干预也不需要软件介入的情况下在两者之间进行的高速数据传送方式。

在DMA传送方式中，对数据传送过程进行控制的硬件称为DMA控制器。当外设需要进行数据传送时，通过DMA控制器向CPU提出DMA传送请求，CPU响应之后将让出系统总线，由DMA控制器接管总线进行数据传送。

DMA和中断的区别



① 中断方式是程序切换，需要保护和恢复现场；而DMA方式除了开始和结尾时，不占用CPU的任何资源。



② 对中断请求的响应只能发生在每条指令执行完毕时；而对DMA请求的响应可以发生在每个机器周期结束时。

③ 中断传送过程需要CPU的干预；而DMA传送过程不需要CPU的干预，故数据传送速率非常高，适合于高速外设的成组数据传送。

④ DMA请求的优先级高于中断请求。

⑤ 中断方式具有对异常事件的处理能力；而DMA方式仅局限于完成传送信息块的I/O操作。

(1) 主存地址计数器

用来存放主存中要交换数据的地址，该计数器的初始值为主存缓冲区的首地址，当DMA传送时，每传送一个数据，将地址计数器加“1”，从而以增量方式给出主存中要交换的一批数据的地址，直至这批数据传送完毕为止。

(2) 传送长度计数器

用来记录传送数据块的长度，其初始值为传送数据的总字数或总字节数，每传送一个字或一个字节，计数器自动减“1”，当其内容为“0”时表示数据已全部传送完毕。

(3) 数据缓冲寄存器

用来暂存每次传送的数据。输入时，数据由外设（如磁盘）先送往数据缓冲寄存器，再通过数据总线送到主存。反之，输出时，数据由主存通过数据总线送到数据缓冲寄存器，然后再送到外设。

(4) DMA请求触发器

每当外设准备好一个数据后给出一个控制信号，使DMA请求触发器置位，控制/状态逻辑经系统总线向CPU发出总线请求（HOLD），如果CPU响应，发回批准信号（HLDA），DMA控制器接管总线控制权，向系统总线送出传送命令与总线地址。控制/状态逻辑接收此信号后使DMA请求触发器复位，为交换下一个数据做准备。

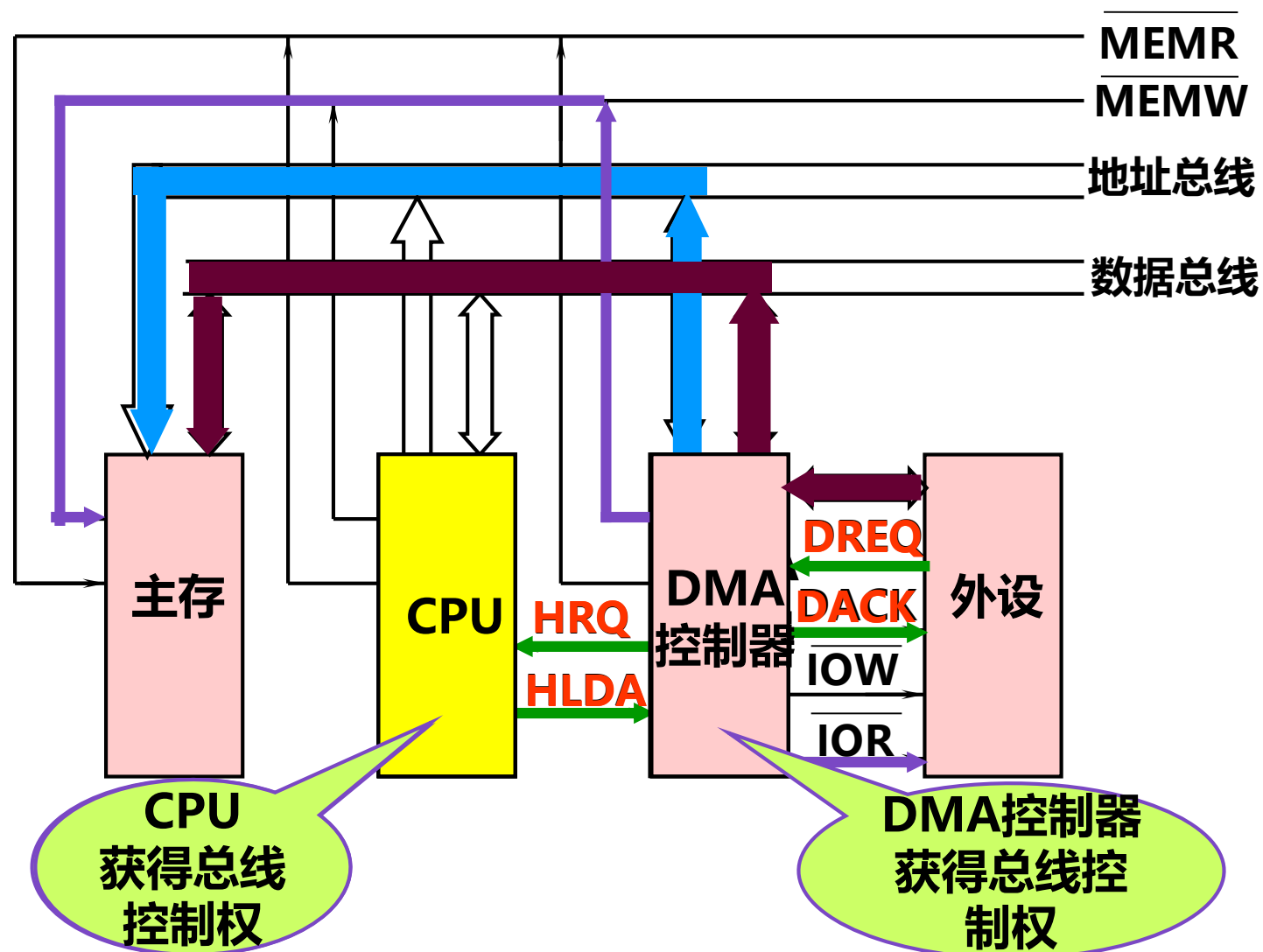
(5) 控制/状态逻辑

它由控制和时序电路以及状态标志等组成，用于指定传送方向，修改传送参数，并对DMA请求信号和CPU响应信号进行协调和同步。

(6) 中断机构

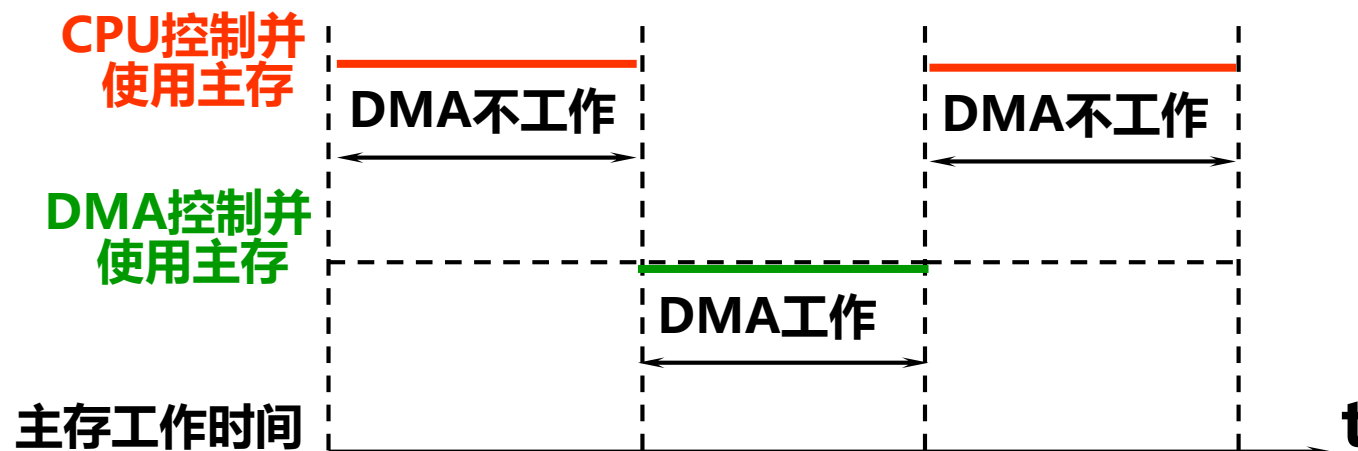
当一个数据块传送完毕，由溢出信号触发中断机构，向CPU提出中断请求，CPU将进行DMA传送的结尾处理。

DMA控制器的连接和传送示意



(1) CPU停止访问主存法

用DMA请求信号迫使CPU让出总线控制权，CPU在现行机器周期执行完成之后，使其数据、地址总线处于三态，并输出总线批准信号。每次DMA请求获得批准后，DMA控制器获得总线控制权以后，连续占用若干个存取周期（总线周期）进行成组连续的数据传送，直至批量传送结束，DMA控制器才把总线控制权交回CPU。在DMA操作期间，CPU处于保持状态，停止访问主存，仅能进行一些与总线无关的内部操作。这种方法只适用于高速外设的成组传送。

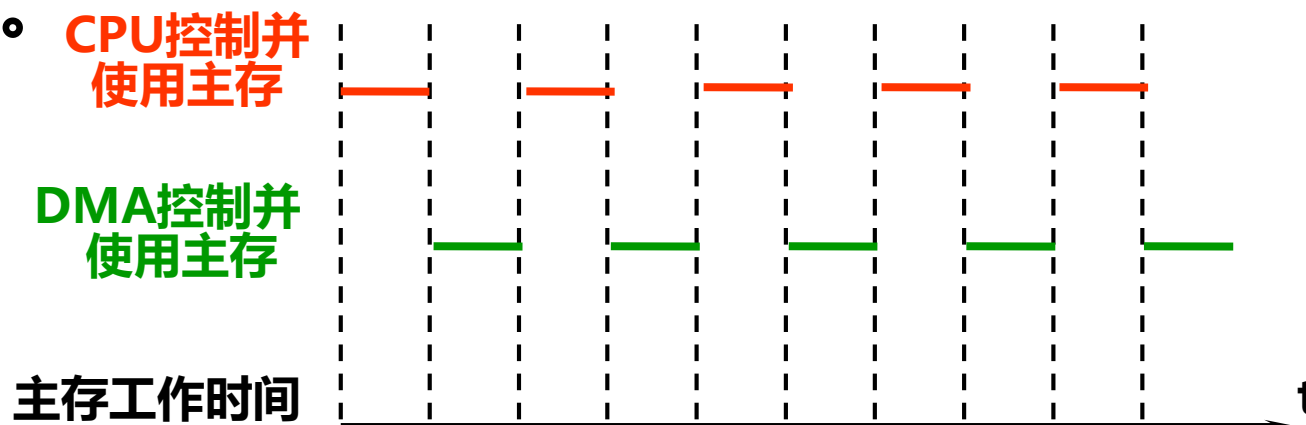




(2) 存储器分时法

把原来的一个存取周期分成两个时间片，一片给CPU，一片给DMA，使CPU和DMA交替地访问主存。这种方法不需要申请和归还总线，使总线控制权的转移几乎不需要什么时间，所以对DMA传送来讲效率是很高的，而且CPU既不停止现行程序的运行，也不进入保持状态，在CPU不知不觉中便进行了DMA传送，但这种方法需要主存在原来的存取周期内为两个部件服务，如果要维持CPU的访存速度不变，就要求主存的工作速度提高一倍。

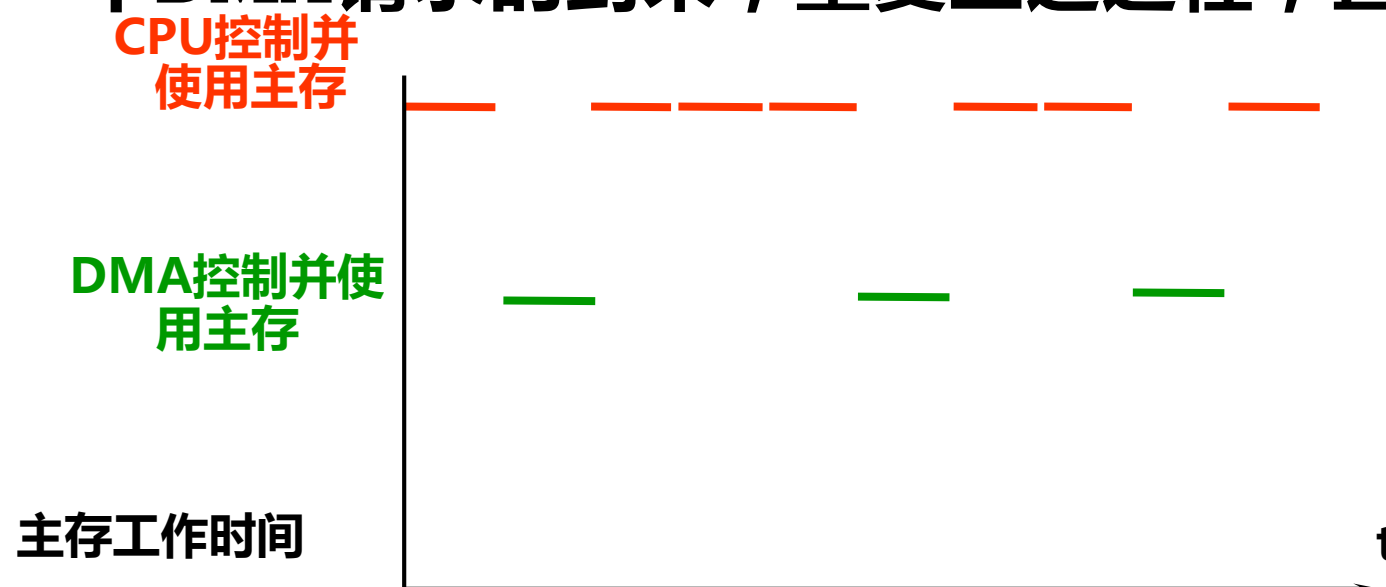
由于大多数外设的速度都不能与CPU相匹配，所以供DMA使用的时间片可能成为空操作，将会造成一些不必要的浪费。





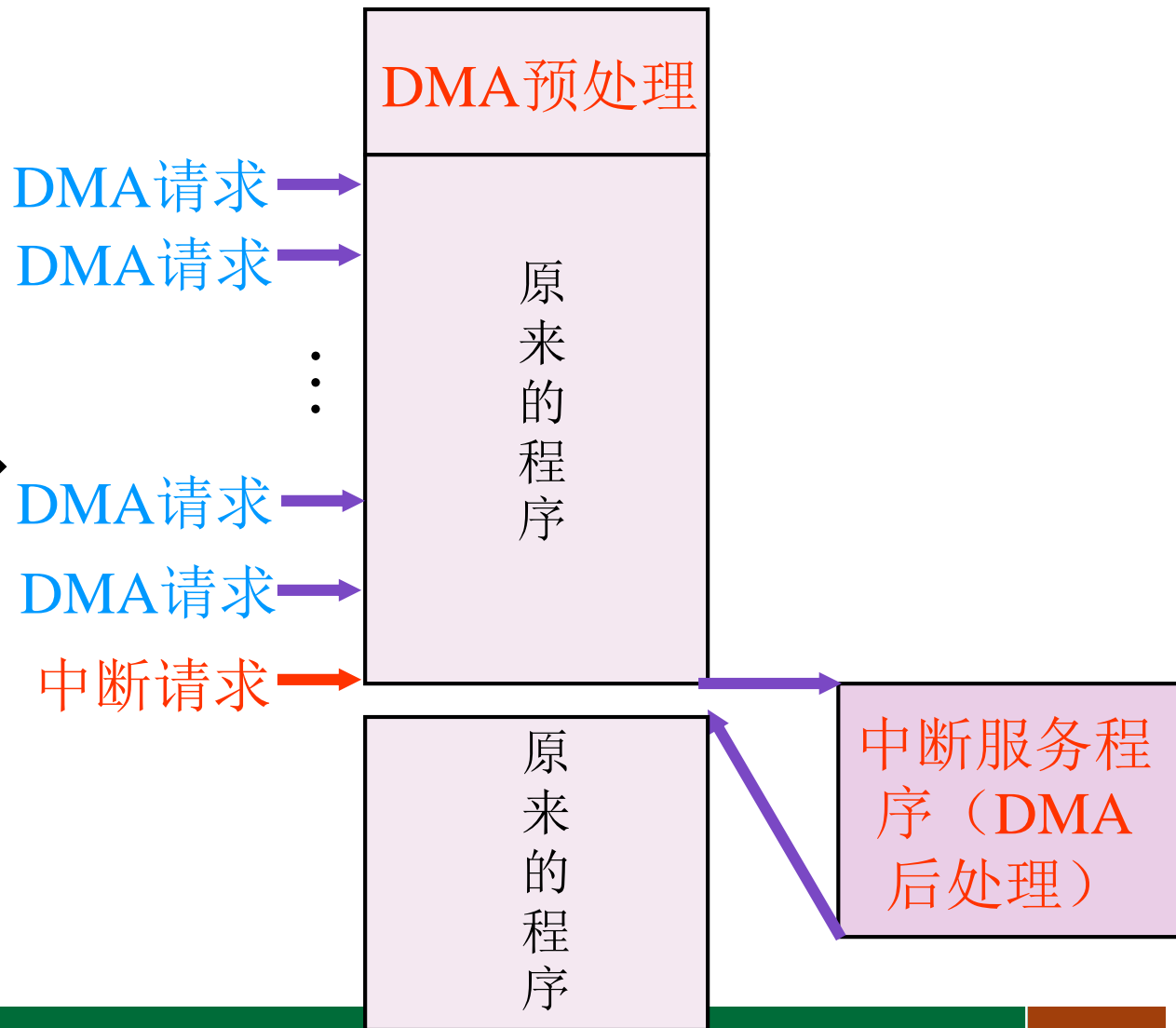
(3) 周期挪用法

周期挪用法是前两种方法的折中。一旦外设有DMA请求并获得CPU批准后，CPU让出一个周期的总线控制权，由DMA控制器控制系统总线，挪用一個存取周期进行一次数据传送，传送一个字节或一个字，然后，DMA控制器将总线控制权交回CPU，CPU继续进行自己的操作，等待下一个DMA请求的到来；重复上述过程，直至数据块传送完毕。



(1) DMA预处理

这是在DMA传送之前做的一些必要的准备工作，是由CPU来完成的。CPU首先执行几条I/O指令，用于测试外设的状态、向DMA控制器的有关寄存器置初值、设置传送方向、启动该外部设备等。





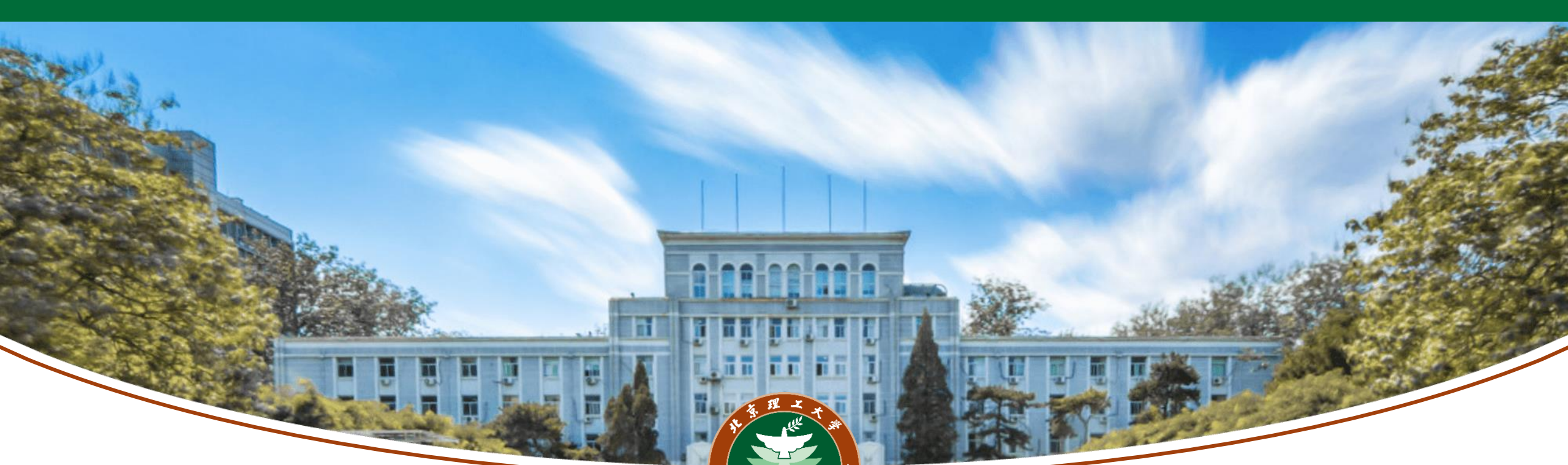
(2) 数据传送

DMA的数据传送可以是以单字节（或字）为基本单位，也可以以数据块为基本单位。对于以数据块为单位的传送，DMA占用总线后的数据输入和输出操作都是通过循环来实现的。

需要特别指出的是，这一循环不是由CPU执行程序实现的，而是由DMA控制器实现的。

(3) DMA后处理

当长度计数器计为0时，DMA操作结束，DMA控制器向CPU发中断请求，CPU 停止原来程序的执行，转去执行中断服务程序做DMA结束处理工作。



感谢聆听