

# 第 10 章 设备管理



**10.1 概述**

**10.2 数据传送控制方式**

**10.3 缓冲技术**

**10.4 设备分配**

**10.5 磁盘的调度**

# 10.1 引言

## 10.1.1 设备的类别

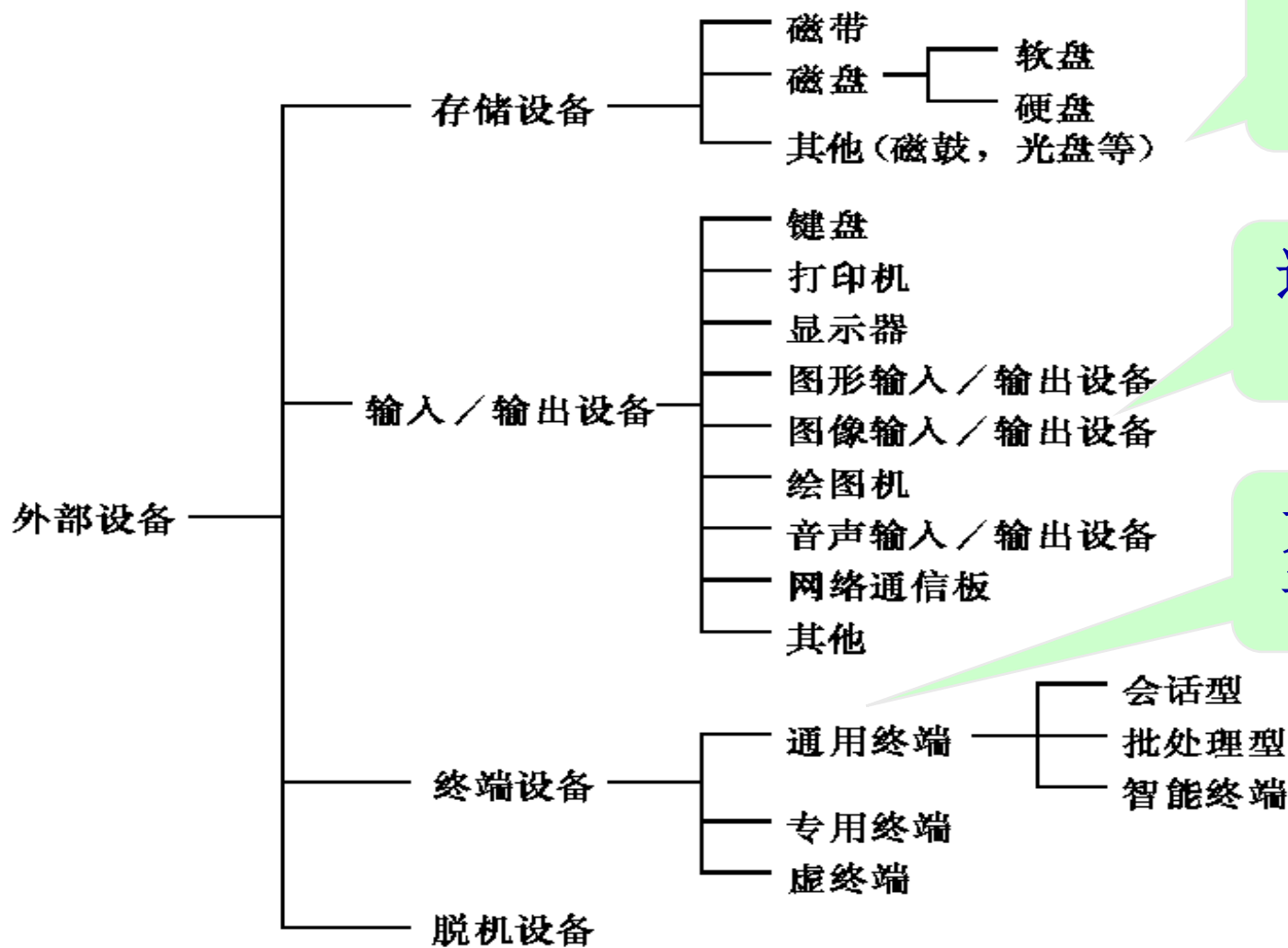
- 设备管理是OS的重要组成部分之一。
- 外部设备,即普通的硬设备。



除了CPU和内存之外



# (1) 按实用特性



还有：U盘  
mp3、mp4

还有：图像  
扫描仪

如：光笔  
语音录入



## (2) 按设备的从属关系

**系统设备：**在操作系统生成时就已配置好的各种标准设备

**用户设备：**由用户自己安装配置后由操作系统统一管理的设备。例如，网络系统中的各种网板、实时系统中的A/D

## (3) 按信息组织方式

**块：**通常是存储设备；以数据块为单位输入输出，有结构设备；传输速率高；可寻址；输入输出时采用DMA方式

**字符：**传输的单位是字符；无结构；种类多；不可寻址；输入输出时采用中断方式。如打印机、串口

## (4) 按速度

低：键盘

中：打印机

高：磁盘



## (5) 按资源分配角度

**独占设备：**在一段时间内只能有一个进程使用的设备，一般为低速I/O设备。且这类设备若分配不当，可能会造成死锁。如打印机、磁带等。

**共享设备：**在一段时间内可有多个进程共同使用的设备，多个进程以交叉的方式来使用设备，其资源利用率高。如硬盘。

**虚拟设备：**在一类设备上模拟另一类设备，常用共享设备模拟独占设备，用高速设备模拟低速设备。如SPOOLing技术。



## 10.1.2 设备管理的任务

- **选择和分配**输入输出设备
- **控制**输入输出设备和CPU及内存之间交换数据
- 为用户提供一个友好**接口**
- 提高设备和设备之间、CPU和设备之间，以及进程和进程之间的**并行操作**程度，以使操作系统获得最佳效率



## 10.2 数据传输控制方式

■ 程序直接控制方式

■ 中断控制方式

■ DMA (Direct Memory Access) 方式 } 微机

■ 通道方式 } 专用计算机、中、大型机

并行度依次提高



## 10.2.1 程序直接控制方式 (忙—等待方式)

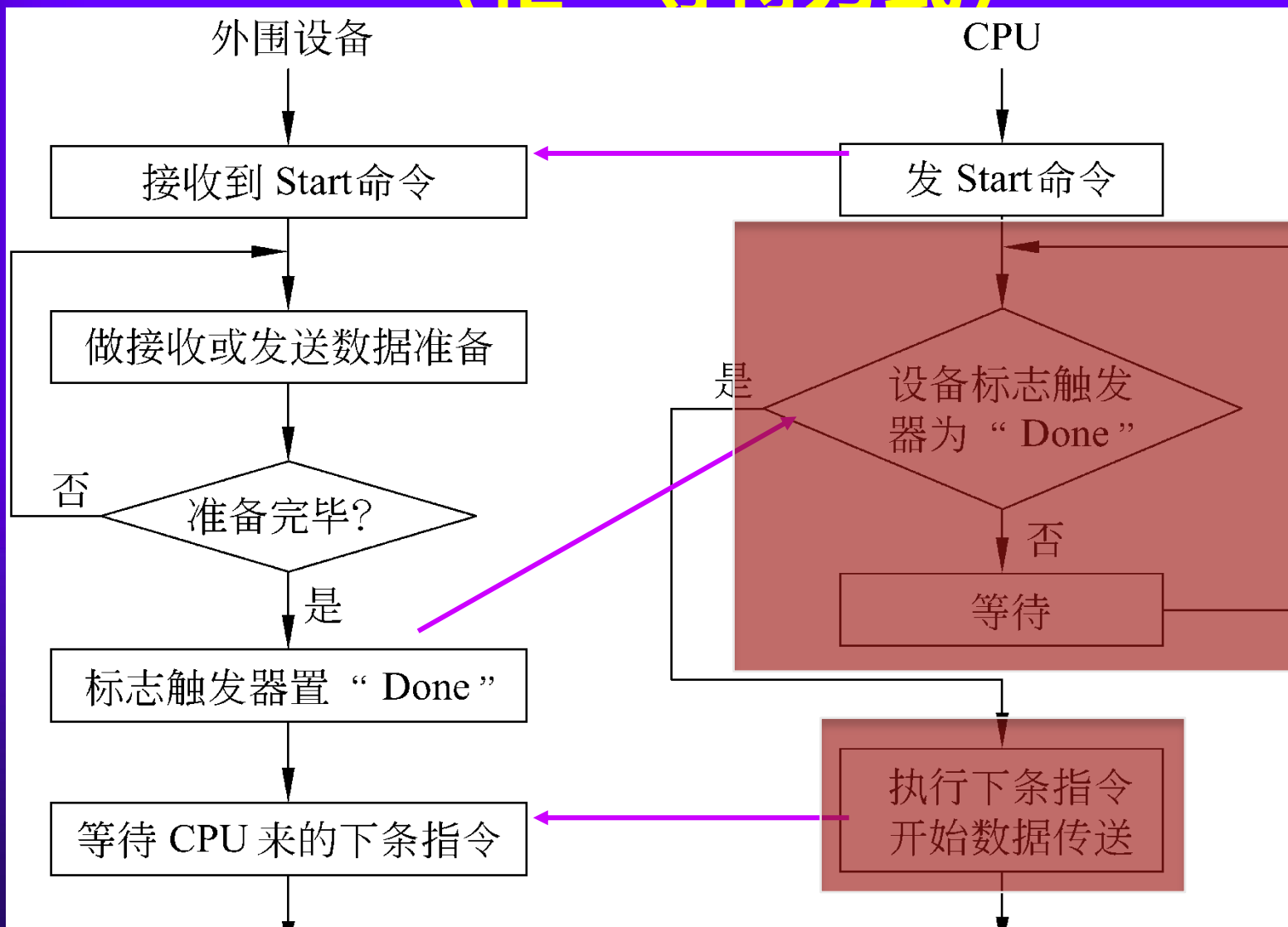


图 程序直接控制方式



- 一旦CPU启动I/O设备，便不断查询I/O设备的准备情况，终止原程序的执行，浪费CPU时间。
- I/O准备就绪后，CPU参与数据传送工作，而不能执行原程序。



## ■ 优点

- 控制简单
- 不需要硬件支持

控制者：用户进程

参与工作：

控制状态寄存器  
数据缓冲寄存器

## ■ 缺点

- CPU和I/O设备串行工作
- 适用于CPU执行速度较慢、且外围设备较少的系统



## 10.2.2 中断方式

- 向I/O发命令—执行其它任务
- I/O中断产生—CPU转相应中断处理程序
- 如：读数据，读完后以中断方式通知CPU，CPU完成数据从I/O—内存

要求：CPU与设备之间有相应的中断线



I/O -> CPU -> 内存

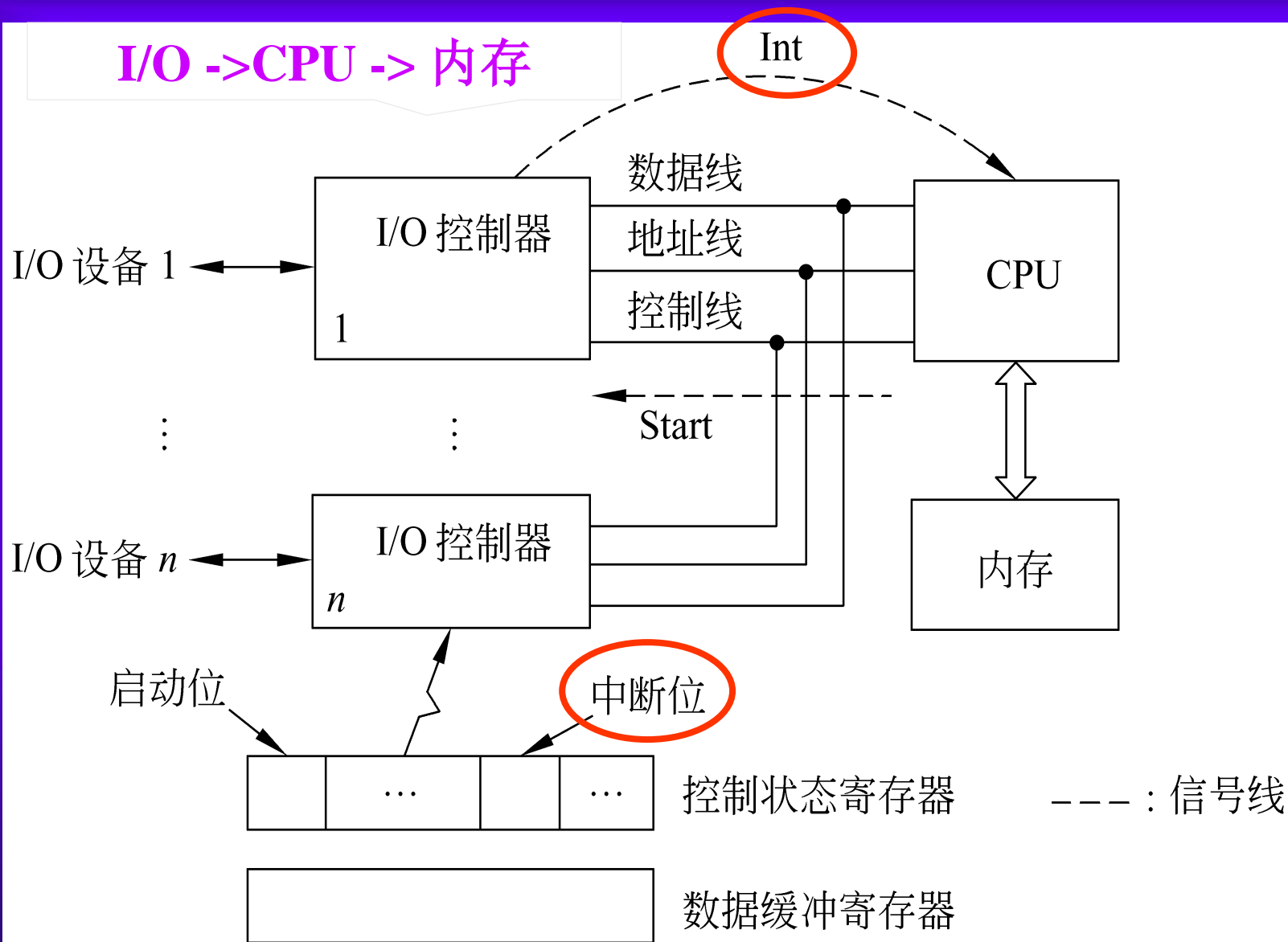


图 中断控制方式的传送结构

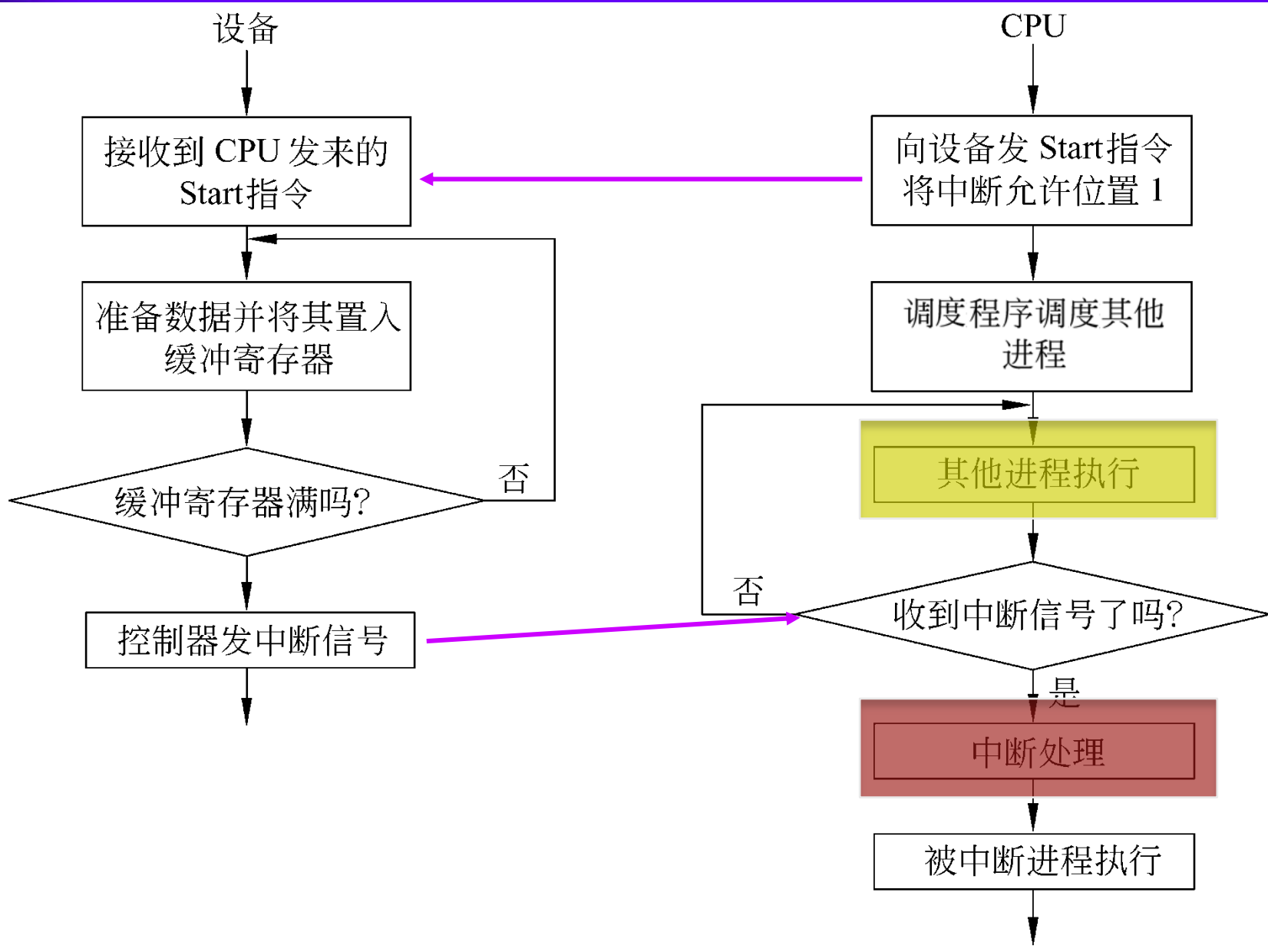


图 中断控制方式的处理过程

# 中断方式存在的问题

- 在一次数据传送过程中，发生中断次数较多。  
耗去大量的CPU处理时间。
  - 数据缓冲寄存通常较小，装满数据发生中断。
- 由于中断次数的急剧增加而造成CPU无法响应中断和出现数据丢失现象。



## 10.2.3 DMA方式—用于块设备中

- **基本思想：** 在外围设备和内存之间开辟直接的数据交换通路。
- **增加硬件**
  - 传送字节计数器、内存地址寄存器等。



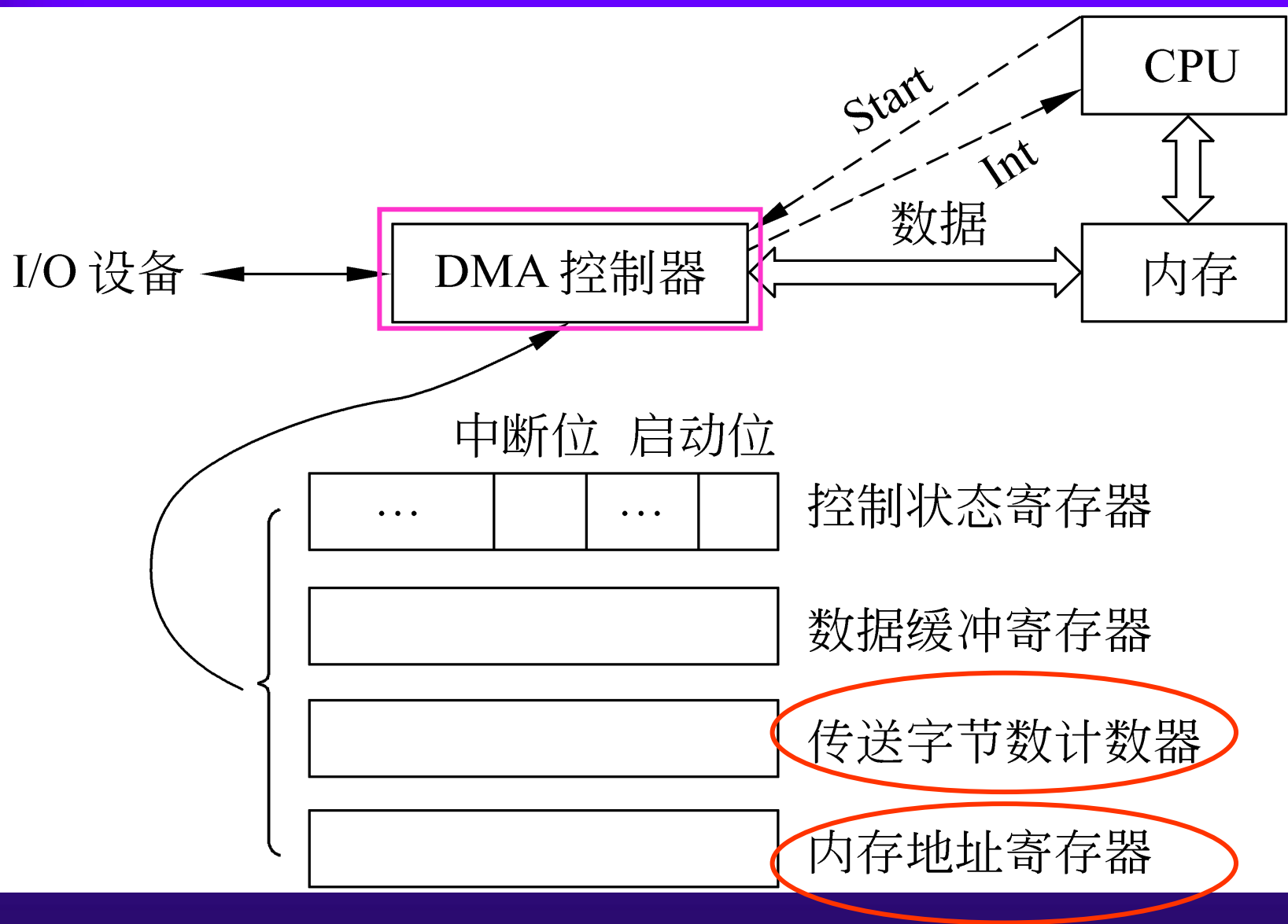


图 DMA方式的传送结构





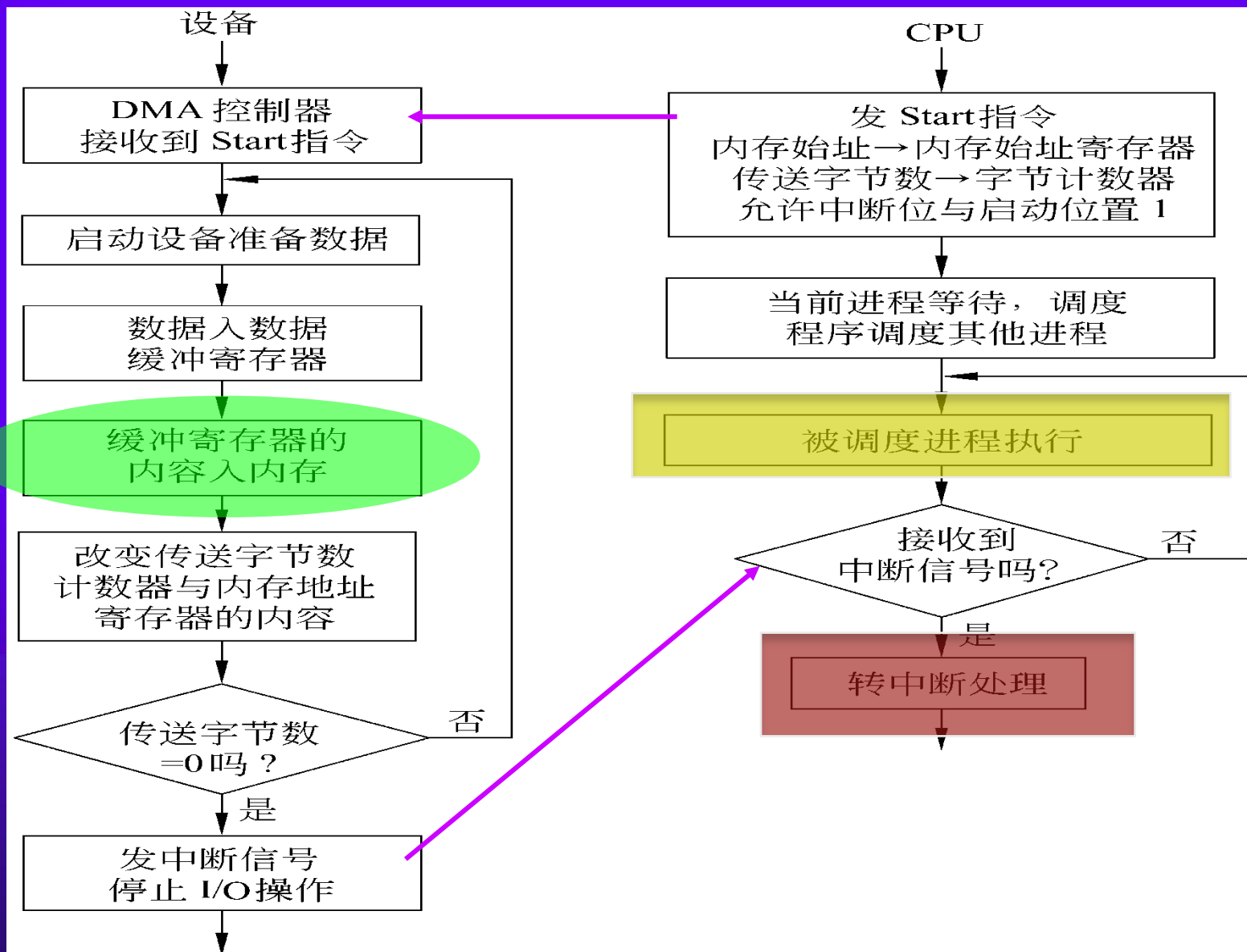


图 DMA方式的数据传送处理过程

# DMA方式

## ■ 比较

### ■ CPU进行中断处理的次数

- 中断：在数据缓冲寄存器满之后
- DMA：在所要传送的数据块全部传送结束时

### ■ 数据传送的控制

- 中断：CPU
- DMA：DMA控制器
  - 排除了因并行操作设备过多时CPU来不及处理或因速度不匹配而造成数据丢失等现象。

## ■ 局限性

- 当设备很多，DMA控制器很多引起内存地址冲突、控制过程复杂；
- 多个DMA控制器的使用不经济。



## 10.2.4 通道控制方式

### ■引入目的

- ◆DMA方式对需多个离散块的读取仍需要多次中断。解脱CPU对I/O的组织、管理。

### ■通道

- 一种特殊的执行I/O指令的处理机，控制设备与内存直接进行数据交换。
- CPU只需发送I/O命令给通道，通道调用内存中的通道程序完成任务。
- CPU只需给出通道程序首址、要访问I/O设备，就可完成一组块操作



## 10.3 缓冲技术

### ■ 缓冲的引入

- 缓和I/O设备与CPU间速度不匹配的矛盾。  
如：计算—打印buffer—打印
- 减少对CPU的中断频率，放宽CPU对中断的响应时间。如：buffer越大，“buffer满”信号发生频率越低。
- 提高CPU和I/O并行性，即解决DMA或通道方式的瓶颈问题



# 缓冲的实现方法

## ■ 根据I/O控制方式

### ■ 专用硬件缓冲器

如I/O控制器中的数据缓冲寄存器。

### ■ 软件缓冲

在**内存**划出一个具有n个单元的**专用缓冲区**，存放输入输出的数据。



## 10.4 设备分配

### 10.4.1 设备分配用的数据结构

- (1) 设备控制表DCT;
- (2) 控制器控制表 (COCT) ;
- (3) 通道表 (CHCT) ;
- (4) 系统设备表 (SDT) :

记录了系统中全部设备及其驱动程序地址。

# 设备分配和管理的数据结构

<b>设备控制表 (DCT)</b>	<b>每个设备一个</b>
<b>系统设备表 (SDT)</b>	<b>整个系统一个</b>
<b>控制器表 (COCT)</b>	<b>每个控制器一个</b>
<b>通道控制表(CHCT)</b>	<b>每个通道一个</b>



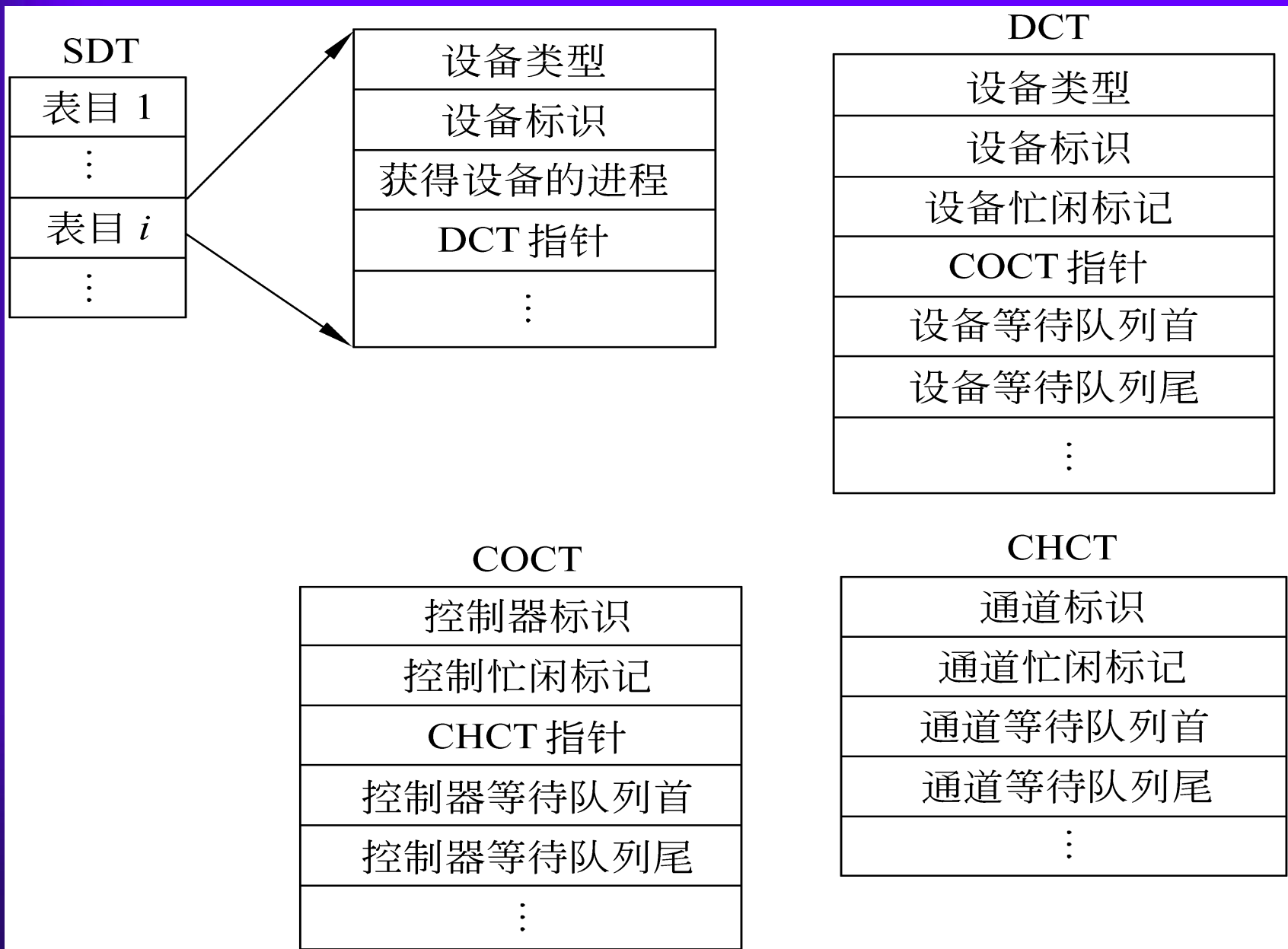


图 数据结构表



## 10.4.2 设备分配的原则

### ■ 设备分配的总原则

- 充分发挥设备的使用效率;
- 尽可能的让设备忙;
- 尽量避免因不合理的分配方法造成的进程死锁;
- 使用户程序面对逻辑设备。

### ■ 设备分配策略

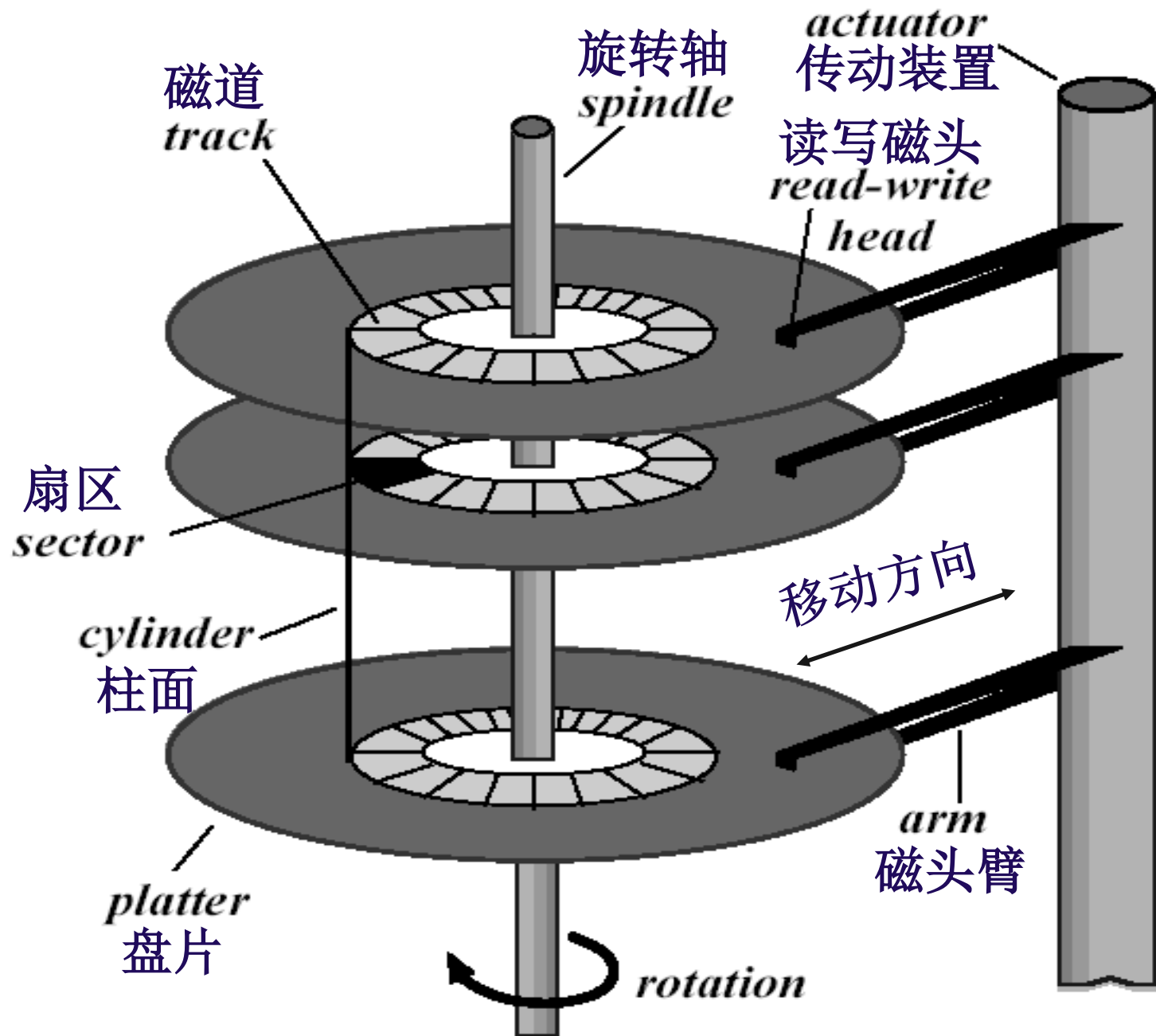
- (1) FIFO
- (2) 优先权



## 10.5 磁盘的调度

- **磁道**：当传动装置固定在某个位置时，若盘面旋转一圈，磁头所能访问的圆环区域；
- **柱面**：在所有盘面上，半径相同的所有磁道即组成一个柱面；
- **扇区**：每一个磁道被划分为若干个扇区；
- **磁盘的访问过程**：以扇区为最小的寻址和存取单位。
  - 移动磁头，定位正确的柱面。
  - 选中相应的**磁头**，想要的扇区正好路过这个磁头正下方的时候，对它进行**访问**。





# 磁盘输入/输出花费的时间

占整个时间的  
70%

## (1) 查找时间/寻道时间

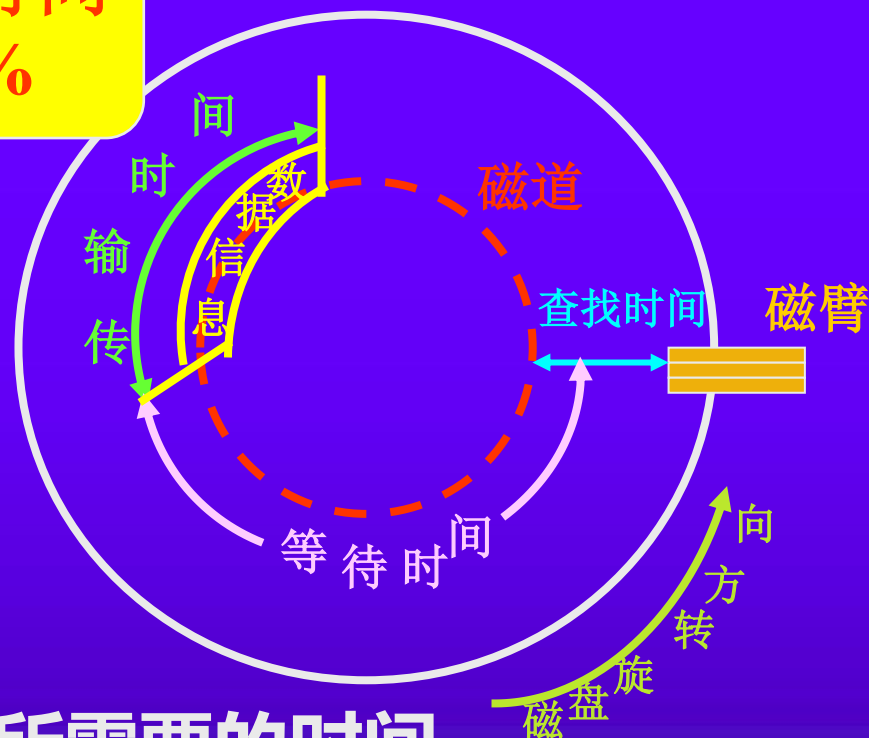
在磁臂的带动下，把磁头移动到指定柱面所需要的时间。

## (2) 等待时间/延迟时间

将指定的扇区旋转到磁头下所需要的时间。  
与磁盘转速有关。

## (3) 传输时间

由磁头进行读/写，完成信息传送所需要的时间。



# 移臂调度与旋转调度

## (1) 移臂调度

减少查找时间着手

## (2) 旋转调度

减少等待时间着手。



**[例] 假定读写磁头位于53号柱面。开始调度时，有若干个进程顺序提出了对如下柱面的I/O请求：**

**98、183、37、122、14、124、65、67。**

**求出采用以下磁盘调度算法移动臂将移动多少个柱面。**

- 1.先来先服务调度算法 (FCFS)**
- 2.最短查找时间优先调度算法 (SSTF)**
- 3.电梯调度算法 (基本扫描) (SCAN)**
- 4.单向扫描调度算法 (循环扫描) (CSCAN)**



98、183、37、122、14、124、65、67。

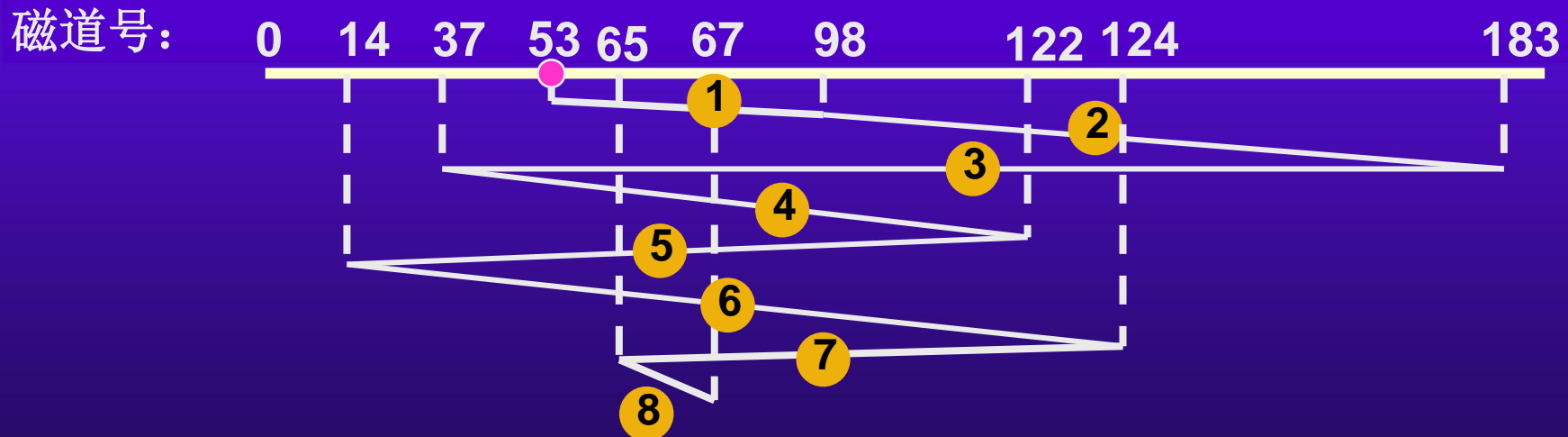
## 先来先服务调度算法 (FCFS)

以I/O请求到达的先后次序作为磁盘调度的顺序。

总共划过的磁道数

$$=(98-53)+(183-98)+(183-37)+(122-37)+(122-14)+(124-14)+(124-65)+(67-65)$$

$$=45+85+146+85+108+110+59+2= \mathbf{640}$$



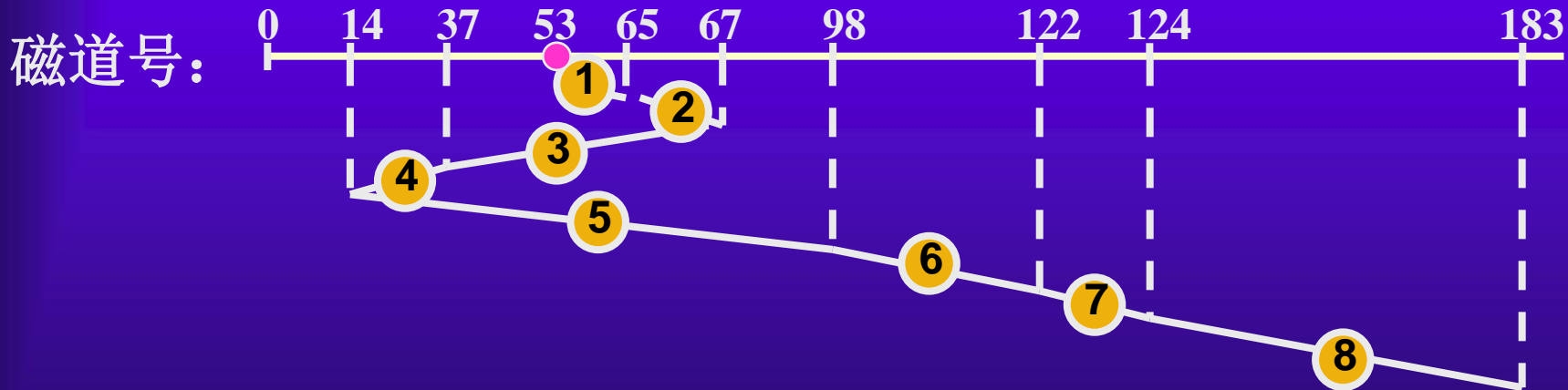
98、183、37、122、14、124、65、67。

## 最短查找时间优先调度算法(SSTF)

把距离磁头当前位置最近的请求作为下一次调度的对象。

总共划过的磁道数

$$\begin{aligned} &= (65-53) + (67-65) + (67-37) + (37-14) + (98-14) + (122-98) + (124-122) \\ &+ (183-124) = 12 + 2 + 30 + 23 + 84 + 24 + 2 + 59 = \mathbf{236} \end{aligned}$$





98、183、37、122、14、124、65、67。

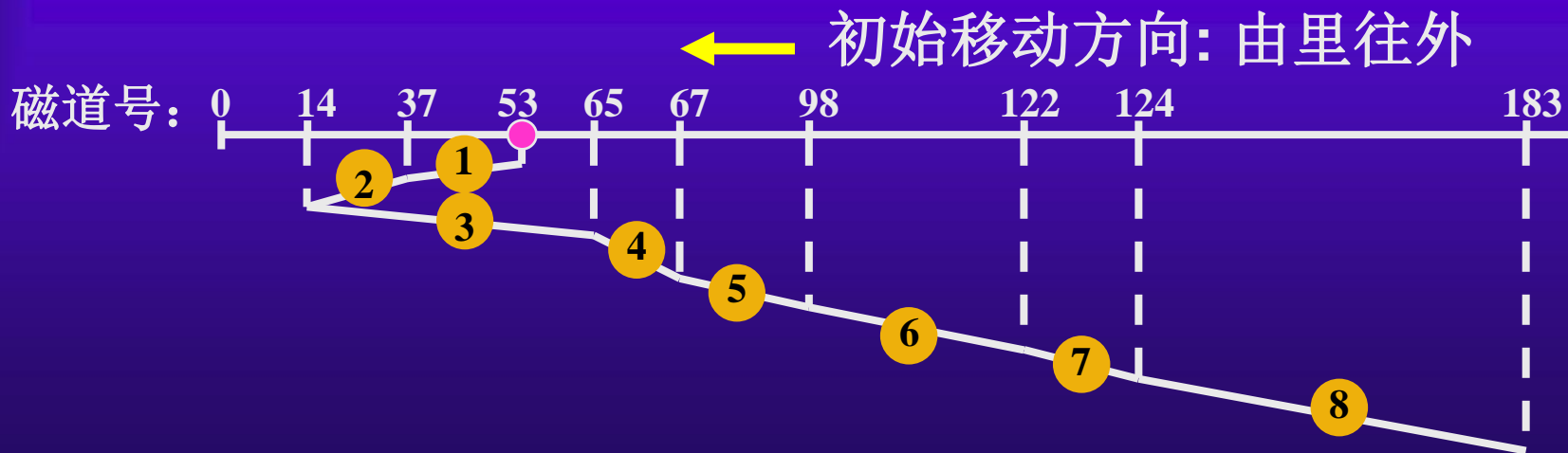
## 电梯调度算法(SCAN)

沿着移动臂的移动方向选择距离磁头当前位置最近的请求作为下一次调度的对象。

如果该方向上已无I/O请求，则改变方向再做选择。

总共划过的磁道数

$$\begin{aligned} &= (53-37) + (37-14) + (65-14) + (67-65) + (98-67) + (122-98) + (124-122) \\ &+ (183-124) = 16 + 23 + 51 + 2 + 31 + 24 + 2 + 59 = \mathbf{208} \end{aligned}$$



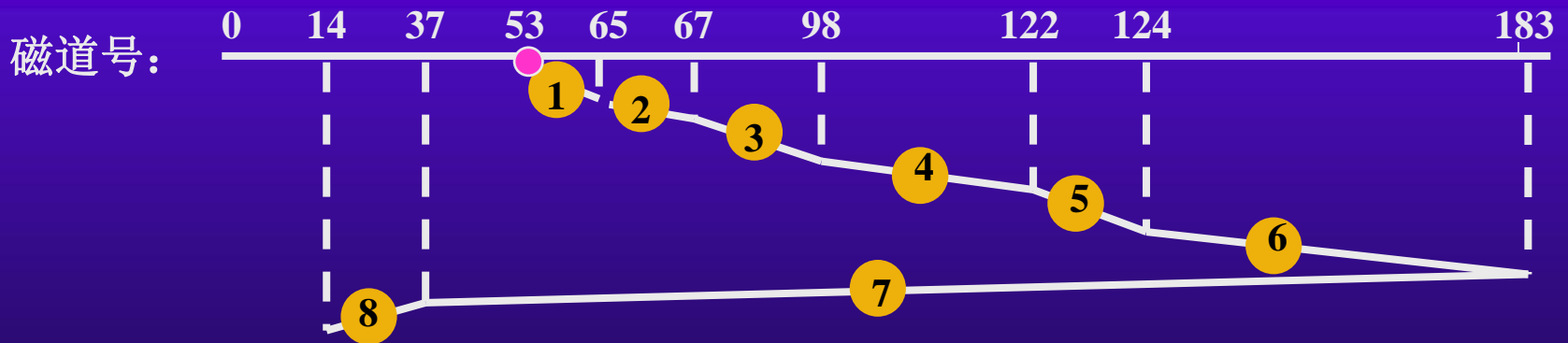
98、183、37、122、14、124、65、67。

## 电梯调度算法（序）

总共划过的磁道数

$$\begin{aligned} &= (65-53) + (67-65) + (98-67) + (122-98) + (124-122) + (183-124) \\ &+ (183-37) + (37-14) = 12 + 2 + 31 + 24 + 2 + 59 + 146 + 23 = \mathbf{299} \end{aligned}$$

初始移动方向: 由外往里 →

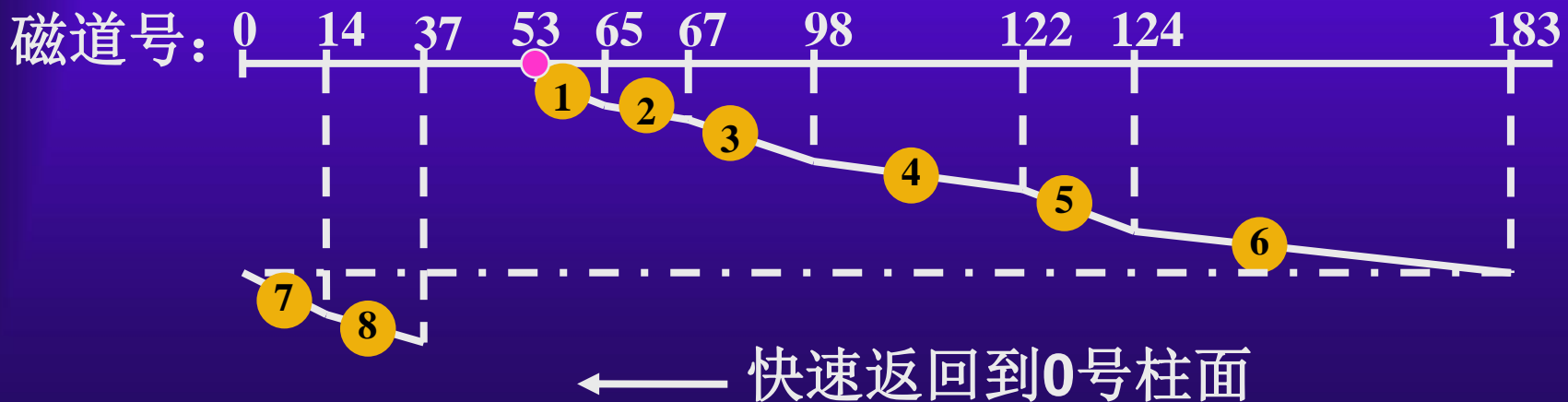


98、183、37、122、14、124、65、67。

## 循环（单向）扫描调度算法(CSCAN)

从0号柱面开始往里移动磁臂，遇到有请求就进行处理，直到到达最后一个请求柱面。移动臂立即带动磁头不做任何服务地快速返回到0号柱面，开始下一次扫描。

总共划过的磁道数=350



# FCFS调度算法

100道开始	
磁道	移动距离
55	45
58	3
39	19
18	21
90	72
160	70
150	10
38	112
184	146
平均寻道长度: 55.3	

# SSTF调度算法

100道开始	
磁道	移动距离
90	10
58	32
55	3
39	16
38	1
18	20
150	132
160	10
184	24
平均寻道长度: 27.5	

55,58,39,18,90,160,150,38,184

## SCAN调度算法

100道开始，增加方向

磁道	移动距离
150	50
160	10
184	24
90	94
58	32
55	3
39	16
38	1
18	20

平均寻道长度: 27.8

## CSCAN调度算法

100道开始，增加方向

磁道	移动距离
150	50
160	10
184	24
18	184+18
38	20
39	1
55	16
58	3
90	32

平均寻道长度: 35.7

55,58,39,18,90,160,150,38,184