

# 计算机组成原理

翁睿

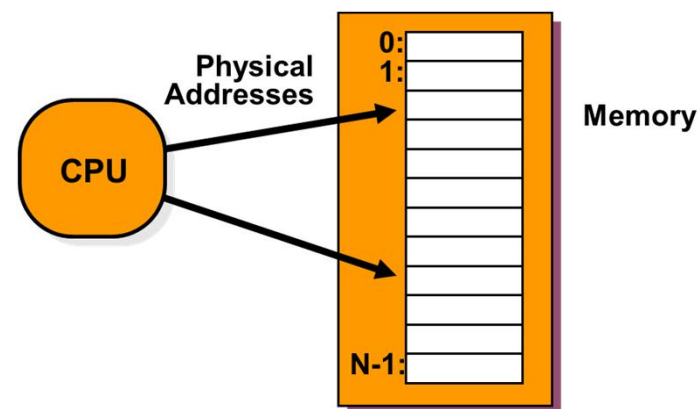
哈尔滨工业大学

## 4.4 虚拟存储器

- 虚拟存储器概述
- 页式虚拟存储器及页表管理
- 虚拟存储器地址映射与变换

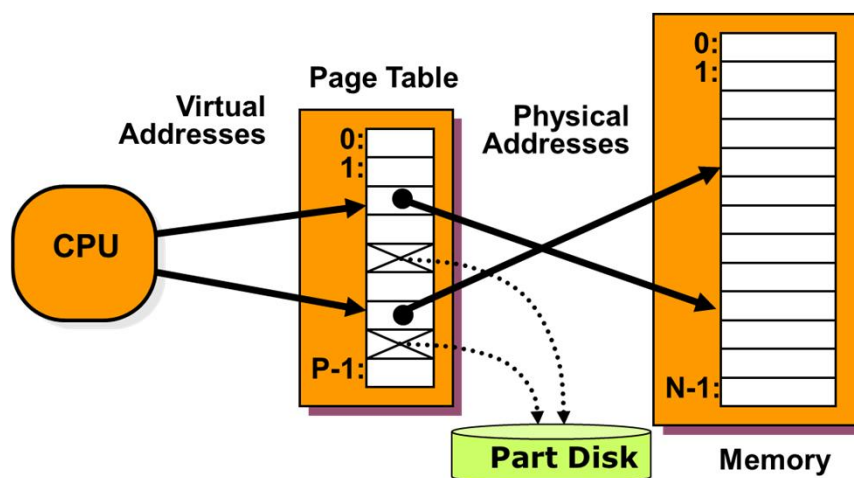
# 实地址计算机系统

- CPU地址：物理内存地址

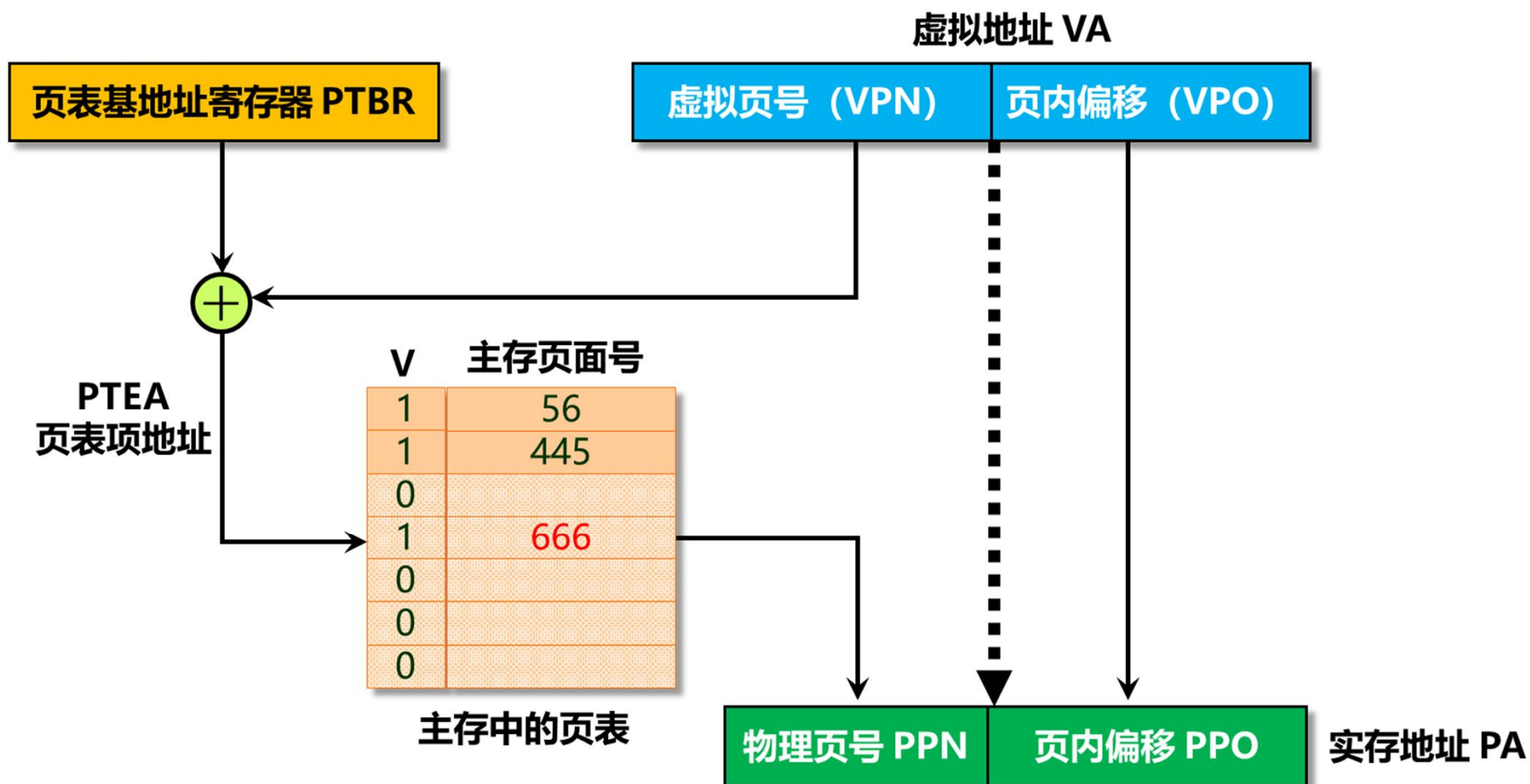


# 虚地址计算机系统

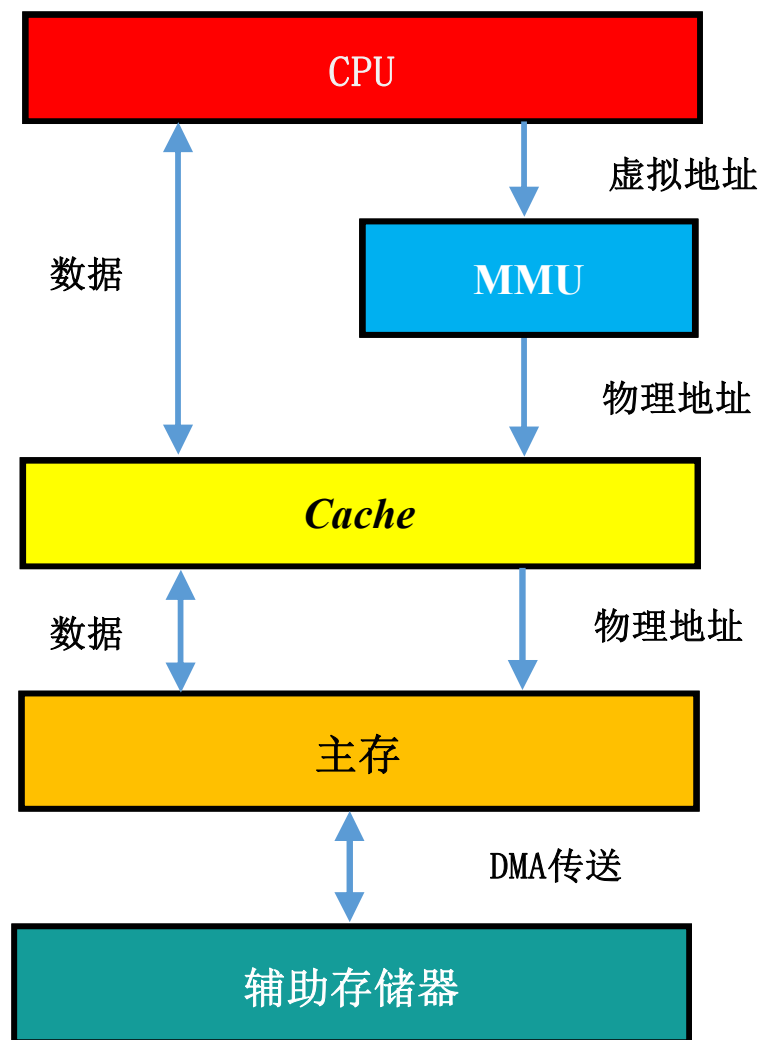
- CPU地址：虚拟地址
  - 需要虚实变换，硬件通过OS维护的页表将虚拟地址转换为物理地址



# 页式虚拟存储器结构



# 层次性结构



为什么Cache用物理地址访问？

# 快表提高地址转换速度

- 地址转换速度慢

- 访问页表，访问数据，需2次访存，速度慢
- 为缩小页表大小，OS普遍采用多级页表结构，速度更慢

- 加速方法：引入一个体积小的快表TLB

- 缓存页表中经常被访问的表项
- (Valid, VPN, PPN)

**Translation Look-aside Buffers**  
( 转译后备缓冲器 / 页表缓存 )

- 快表引入相联存储器机制，提高查找速度

- 采用随机替换算法

## 举例：三种不同缺失的组合

假设：Cache不缓存页表项

TLB缺失：页表项不在TLB中

页表缺失：目标页面不在主存中

缓存缺失：目标页面不在缓存中

TLB	Page table	Cache	Possible? If so, under what circumstance?
hit	hit	miss	可能，TLB命中则页表一定命中，但实际上不会查页表
miss	hit	hit	可能，TLB缺失但页表命中，信息在主存，就可能在Cache
miss	hit	miss	可能，TLB缺失但页表命中，信息在主存，但可能不在Cache
miss	miss	miss	可能，TLB缺失页表缺失，信息不在主存，一定也不在Cache
hit	miss	miss	不可能，页表缺失，信息不在主存，TLB中一定没有该页表项
hit	miss	hit	同上
miss	miss	hit	不可能，页表缺失，信息不在主存，Cache中一定也无该信息

最好的情况是hit、hit、hit，此时，访问主存几次？ 不需要访问主存！

以上组合中，最好的情况是？ hit、hit、miss和miss、hit、hit 访存1次

以上组合中，最坏的情况是？ miss、miss、miss 需访问磁盘、并访存至少2次

介于最坏和最好之间的是？ miss、hit、miss 不需访问磁盘、但访存至少2次

# 第4章 存储系统

## 4.1 概述

## 4.2 主存储器

## 4.3 高速缓冲存储器

## 4.4 虚拟存储器

## 4.5 辅助存储器

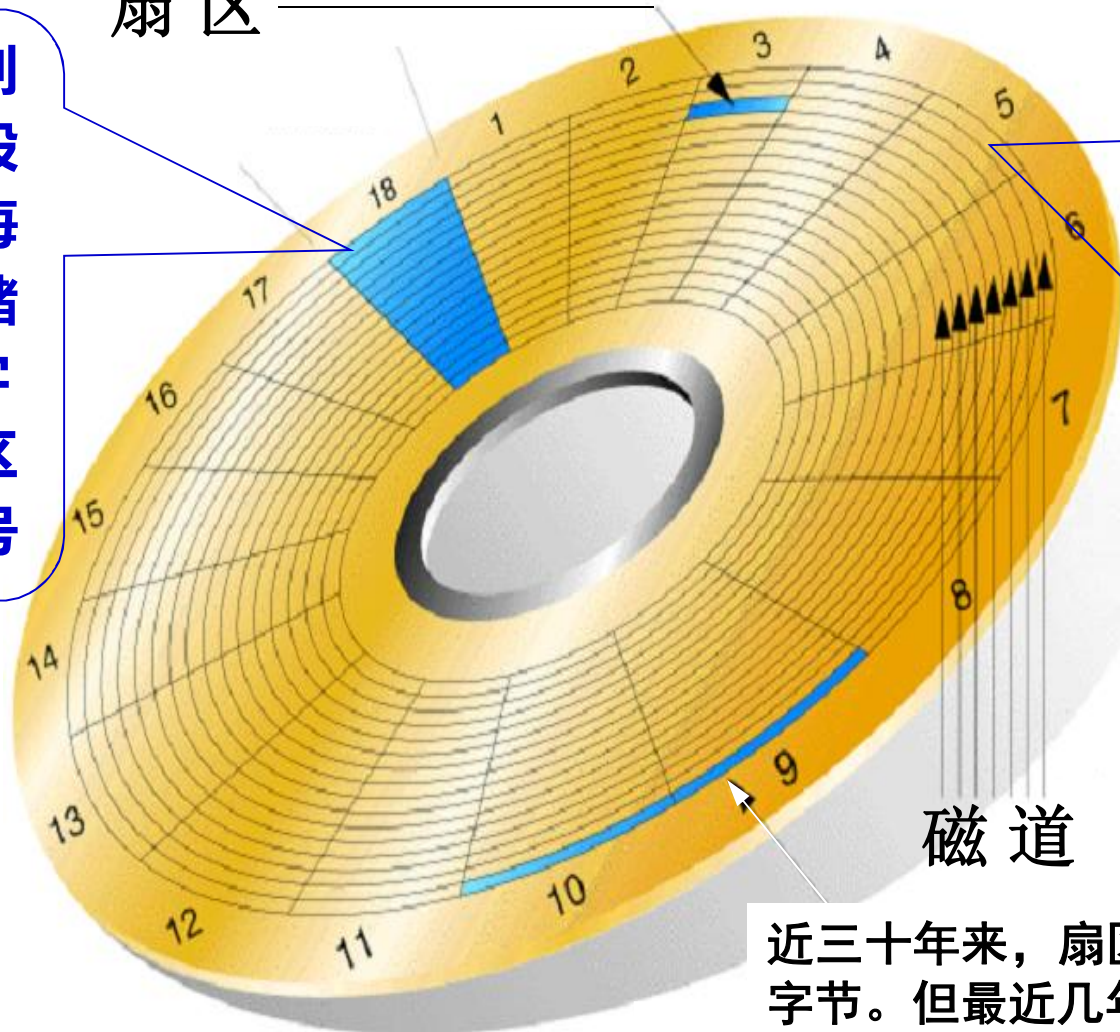


# 磁盘的磁道和扇区

## 4.5

每个磁道被划分为若干区段（扇区），每个扇区的存储容量为512字节。每个扇区都有一个编号

扇区



磁盘表面被分为许多同心圆，每个同心圆称为一个磁道。每个磁道都有一个编号，最外面的是0磁道。

磁道

近三十年来，扇区大小一直是512字节。但最近几年正迁移到更大、更高效的4096字节扇区，通常称为4K扇区。国际硬盘设备与材料协会（IDEMA）将之称为高级格式化。



## 4.5 辅助存储器

### 2. 磁表面存储器的技术指标

(1) 记录密度          道密度  $D_t$     位密度  $D_b$

(2) 存储容量           $C = n \times k \times s = \text{面数} \times \text{磁道数} \times \text{道容量}$

(3) 平均寻址时间    寻道时间 + 等待时间

辅存的速度  $\left\{ \begin{array}{l} \text{寻址时间} \\ \text{磁头读写时间} \end{array} \right.$

(4) 数据传输率       $D_r = D_b \times V = \text{记录密度} \times \text{线速度}$

(5) 误码率            出错信息位数与读出信息的总位数之比

## 磁盘响应时间计算举例

- 假定每个扇区512字节，磁盘转速为5400 RPM，声称寻道时间（最大寻道时间的一半）为12 ms, 数据传输率为4 MB/s, 磁盘控制器开销为1 ms, 不考虑排队时间，则磁盘响应时间为多少？

$$\begin{aligned}
 \text{Disk Response Time} &= \text{Seek time} + \text{Rotational Latency} + \text{Transfer time} \\
 &\quad + \text{Controller Time} + \text{Queuing Delay} \\
 &= 12 \text{ ms} + 0.5 / 5400 \text{ RPM} + 0.5 \text{ KB} / 4 \text{ MB/s} + 1 \text{ ms} + 0 \\
 &= 12 \text{ ms} + 0.5 / 90 \text{ RPS} + 0.125 / 1024 \text{ s} + 1 \text{ ms} + 0 \\
 &= 12 \text{ ms} + 5.5 \text{ ms} + 0.1 \text{ ms} + 1 \text{ ms} + 0 \text{ ms} \\
 &= 18.6 \text{ ms}
 \end{aligned}$$

如果实际的寻道时间只有1/3的话，则总时间变为10.6ms，这样旋转等待时间就占了约50%！

$$12/3 + 5.5 + 0.1 + 1 = 10.6 \text{ ms} \quad \text{所以，磁盘转速非常重要！}$$

为什么实际的寻道时间可能只有1/3？

访问局部性使得每次磁盘访问大多在局部几个磁道，实际寻道时间变少！

# 第 5 章 输入输出系统

## 5.1 概述

## 5.2 外部设备

## 5.3 I/O接口

## 5.4 程序查询方式

## 5.5 程序中断方式

## 5.6 DMA方式

## 5.1 概述

## 5.1

- 输入输出系统的功能
    - 解决各种形式信息的输入和输出
      - 即：用户如何将所需的信息（文字、图表、声音、视频等）通过不同的外设输入到计算机中，以及计算机内部处理的结果信息如何通过相应的外设输出给用户
  - 要实现上述功能，需解决以下一系列的问题
    - 怎样在CPU、主存和外设间建立一个高效信息传输 “通路”；
    - 怎样将用户的I/O请求转换成设备的命令；
    - 如何对外设进行编址；
    - I/O硬件和操作系统如何协调完成主机和外设之间的数据传送
- 等等

# 一、输入输出系统的发展概况

## 1. 早期

分散连接

CPU 和 I/O设备 串行 工作 程序查询方式

## 2. 接口模块和 DMA 阶段

总线连接

CPU 和 I/O设备 并行 工作 { 中断方式  
DMA 方式

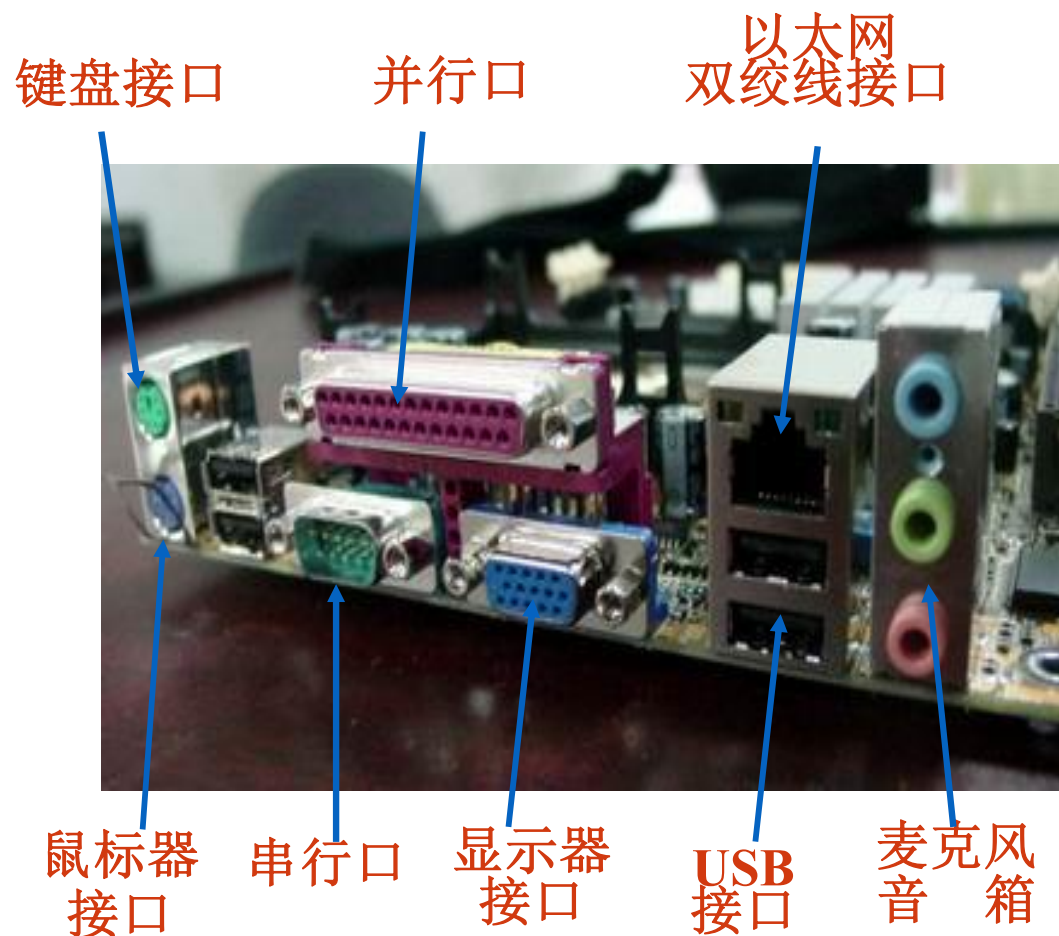
## 3. 具有通道结构的阶段

## 4. 具有 I/O 处理机的阶段

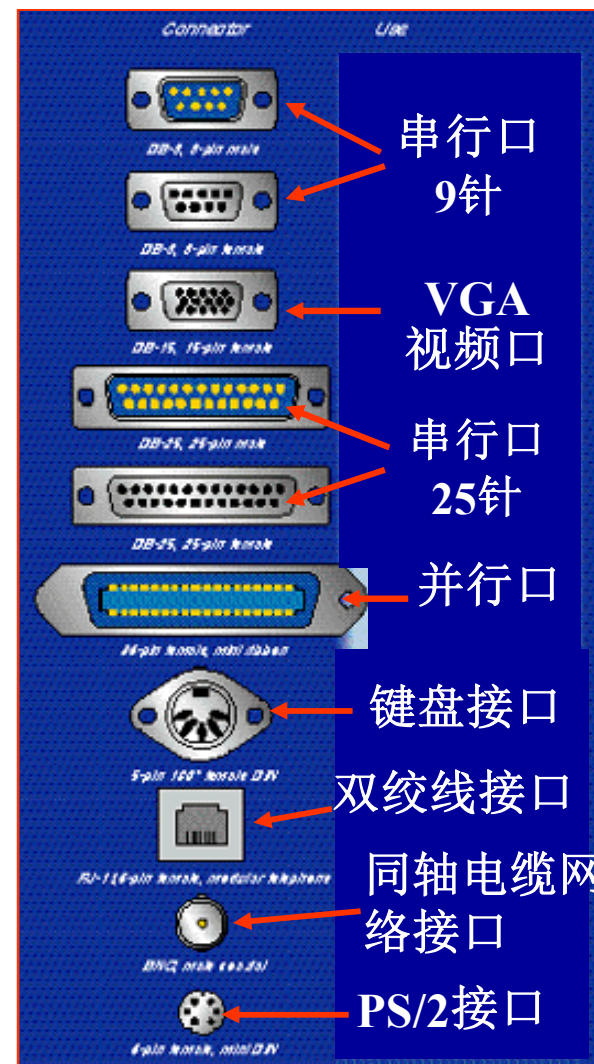


# I/O设备接口插座（连接器）

## 5.1



(安装在主板上的I/O设备接口插座)



## 5. 外设发展与分类

### 5.1

- 从交互方式上来分，外设分为：
  - 人-机交互设备
    - 输入/输出的信息是人可读的
    - 如：键盘、鼠标、扫描仪、打印机、显示器等
  - 机器可读设备
    - 输入/输出的信息是机器可读的，而人无法读取
    - 如：网络、Modem、D/A、A/D、磁盘、声音输入设备等



## 5. 外设发展与分类

### 5.1

- 从功能行为来分，外设分为：
  - 输入/输出设备（大部分为字符型设备）
    - 用于信息的输入/输出
    - 输入设备：键盘、鼠标、扫描仪等
    - 输出设备：打印机、显示器等
  - 外部存储设备（大部分为成块传送设备）
    - 用于信息的存储  
（其输入/出的信息是机器可读的）
    - 如：磁盘、磁带、光盘等

## 6. 常用外部设备

### 5.1

- 输入设备：
  - 键盘、触摸屏
  - 图形输入设备(鼠标、图形板、跟踪球、操纵杆、光笔)
  - 图像输入设备(摄像机、扫描仪、传真机)
  - 条形码阅读机、光学字符识别设备(OCR)
  - 音、视频输入设备
- 输出设备：
  - 显示器(字符、汉字、图形、图像)
  - 打印设备(点阵、激光、喷墨)
  - 绘图仪(平板式、滚筒式)
  - 声音输出设备
- 其它：
  - 终端设备(键盘+显示器)
  - 外存储器(磁盘、磁带、光盘)

## 二、输入输出系统的组成

## 5.1

### 1. I/O 软件

(1) I/O 指令      CPU 指令的一部分

操作码	命令码	设备码
-----	-----	-----

(2) 通道指令      通道自身的指令

指出数组的首地址、传送字数、操作命令

如 IBM/370 通道指令为 64 位

### 2. I/O 硬件

设备      I/O 接口

设备      设备控制器      通道

## 三、I/O 设备与主机的联系方式

## 5.1

### 1. I/O 设备编址方式

- (1) 统一编址      用取数、存数指令
- (2) 不统一编址    有专门的 I/O 指令

### 2. 设备选址

用设备选择电路识别是否被选中

### 3. 传送方式

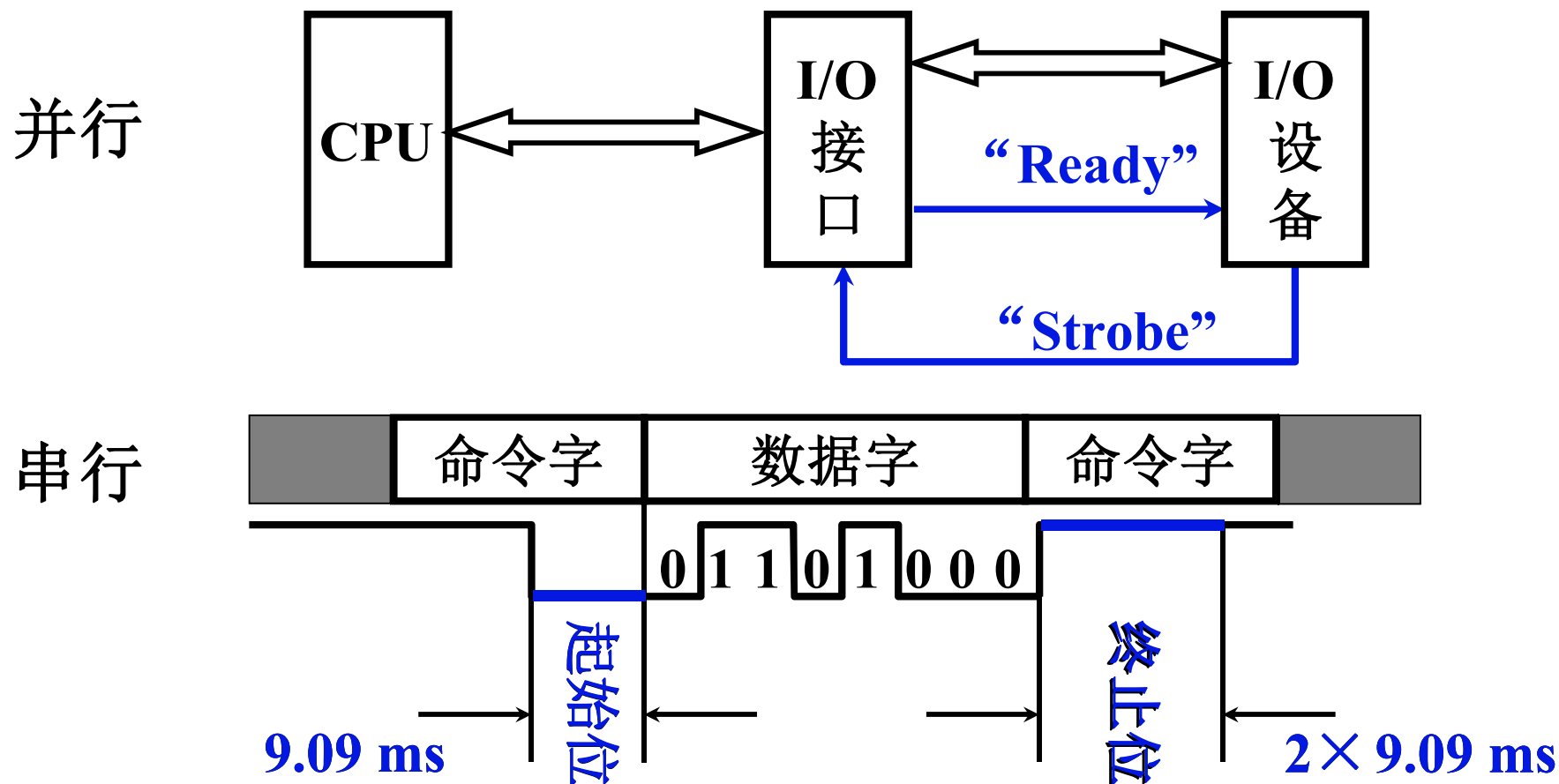
- (1) 串行
- (2) 并行

## 4. 联络方式

5.1

(1) 立即响应

(2) 异步工作采用应答信号

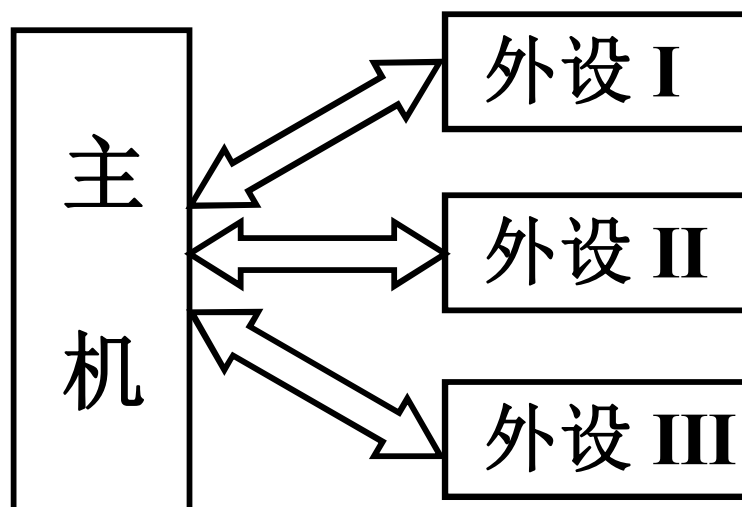


(3) 同步工作采用同步时标

## 5. I/O 设备与主机的连接方式

### 5.1

#### (1) 辐射式连接



每台设备都配有一套  
控制线路和一组信号线

不便于增删设备

#### (2) 总线连接

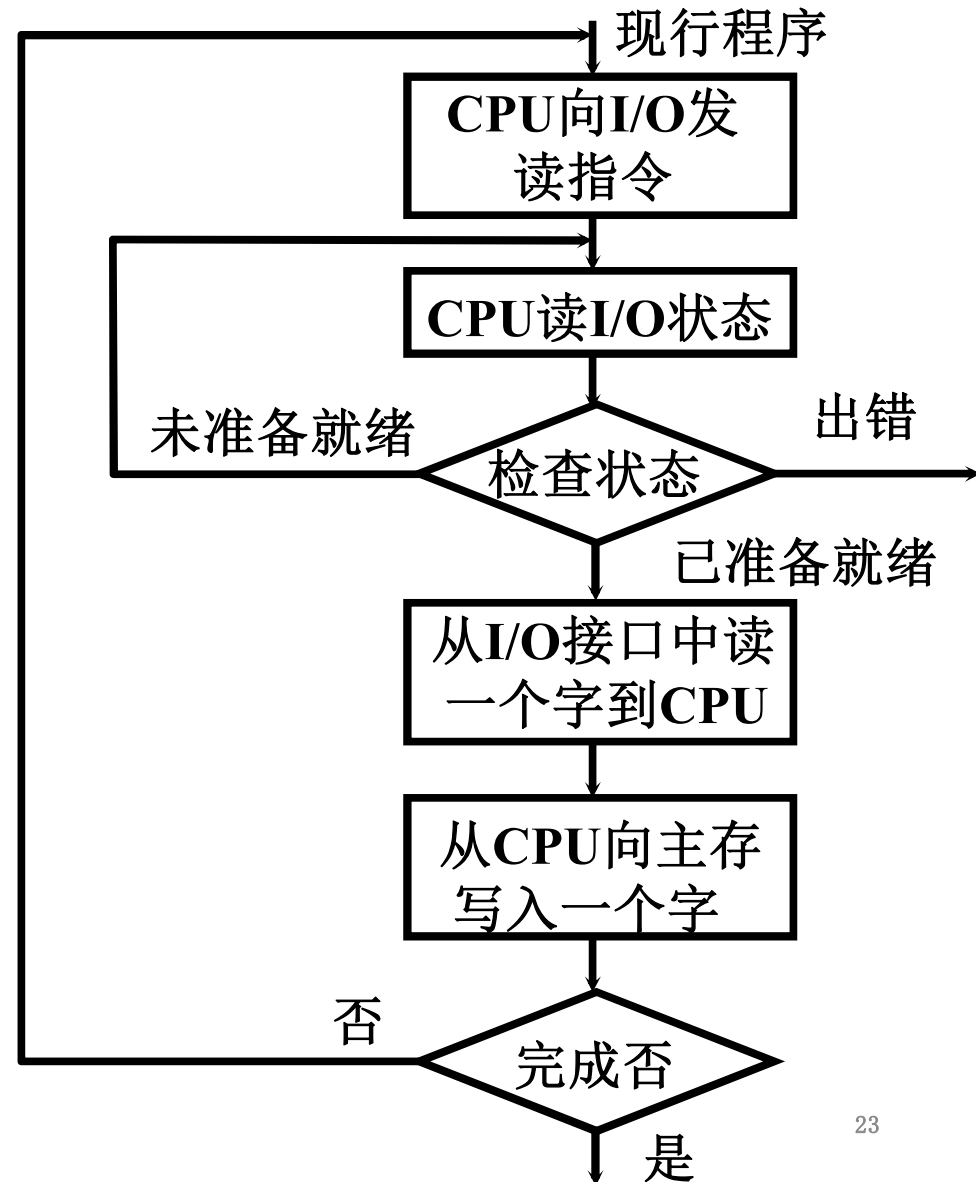
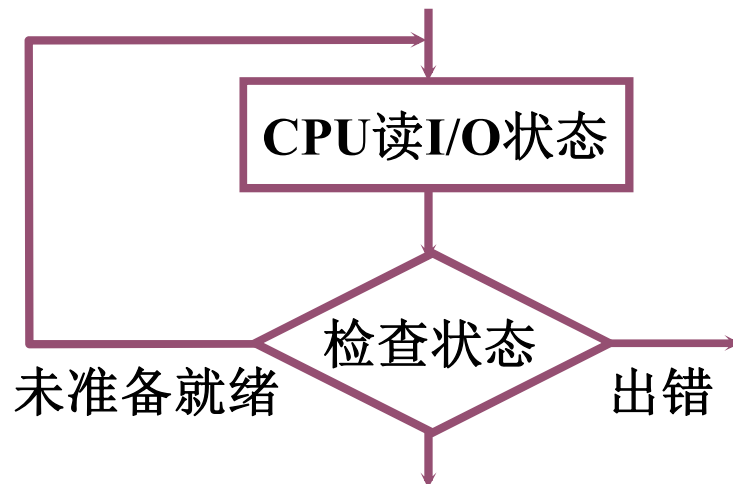
便于增删设备

## 四、I/O设备与主机信息传送的控制方式 5.1

### 1. 程序查询方式

CPU 和 I/O 串行工作

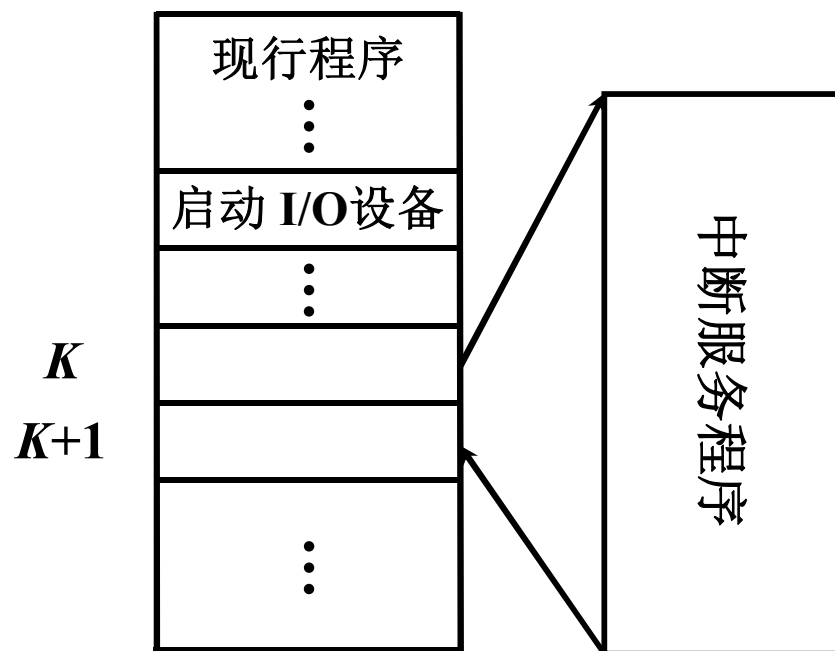
踏步等待



## 2. 程序中中断方式

I/O 工作 { 自身准备      CPU 不查询  
与主机交换信息    CPU 暂停现行程序

**CPU 和 I/O 并行工作**



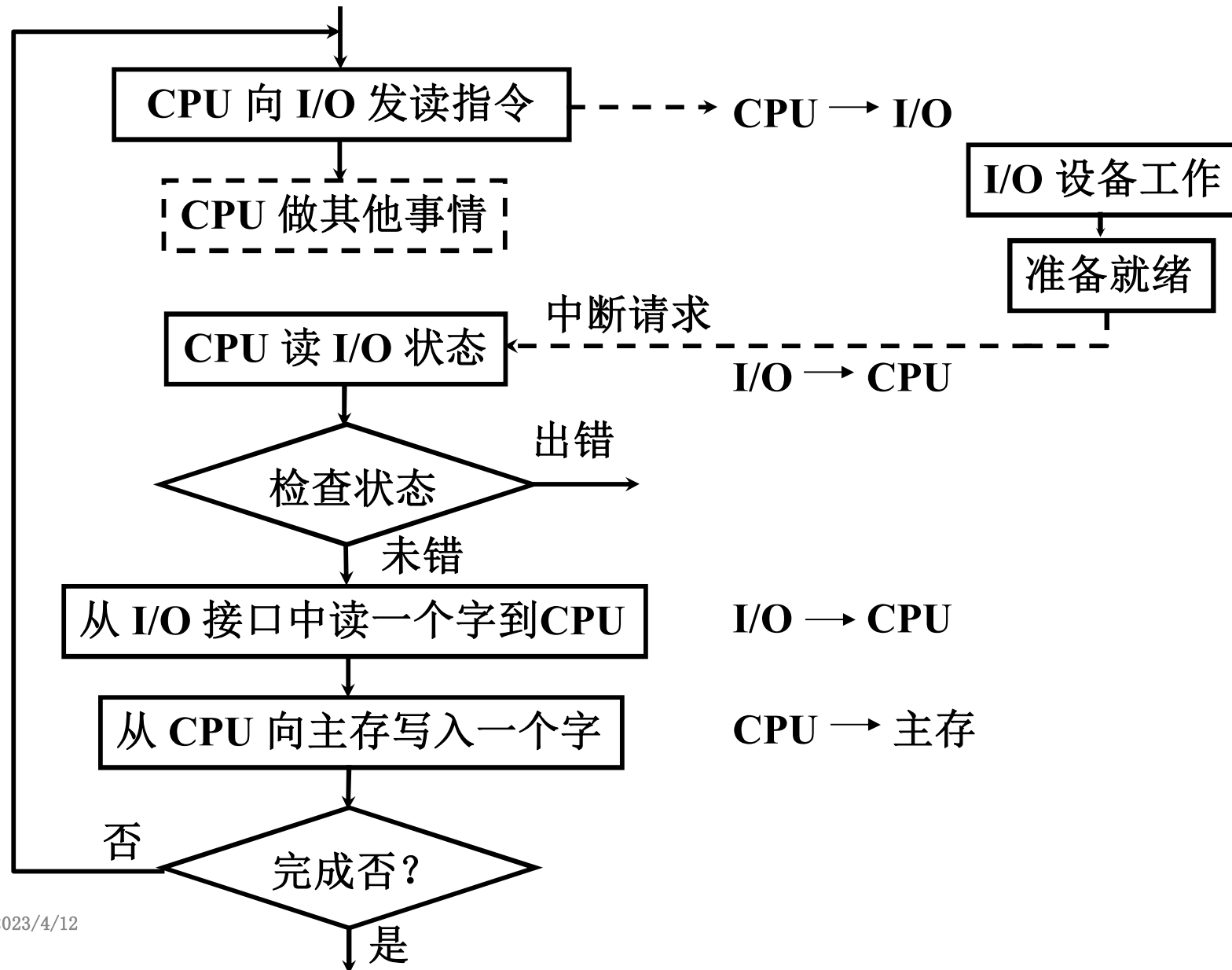
**没有踏步等待现象**

**中断现行程序**



# 程序中中断方式流程

## 5.1



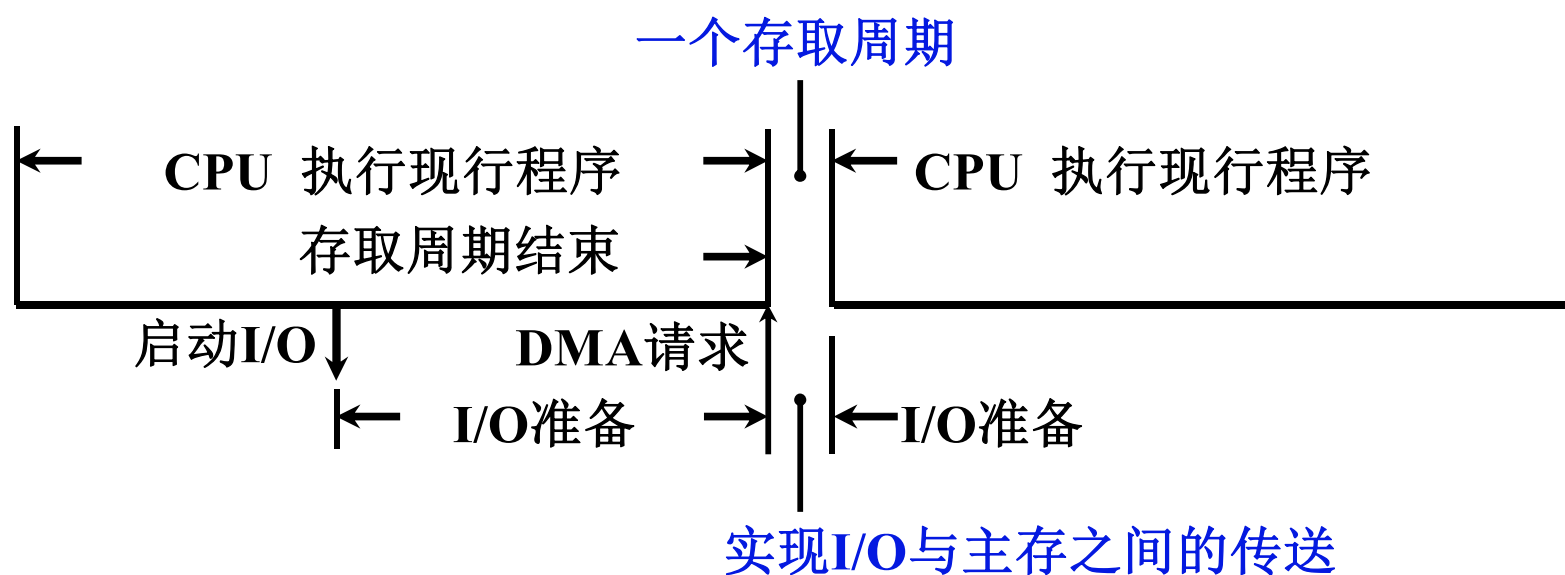
### 3. DMA 方式

主存和 I/O 之间有一条直接数据通道

不中断现行程序

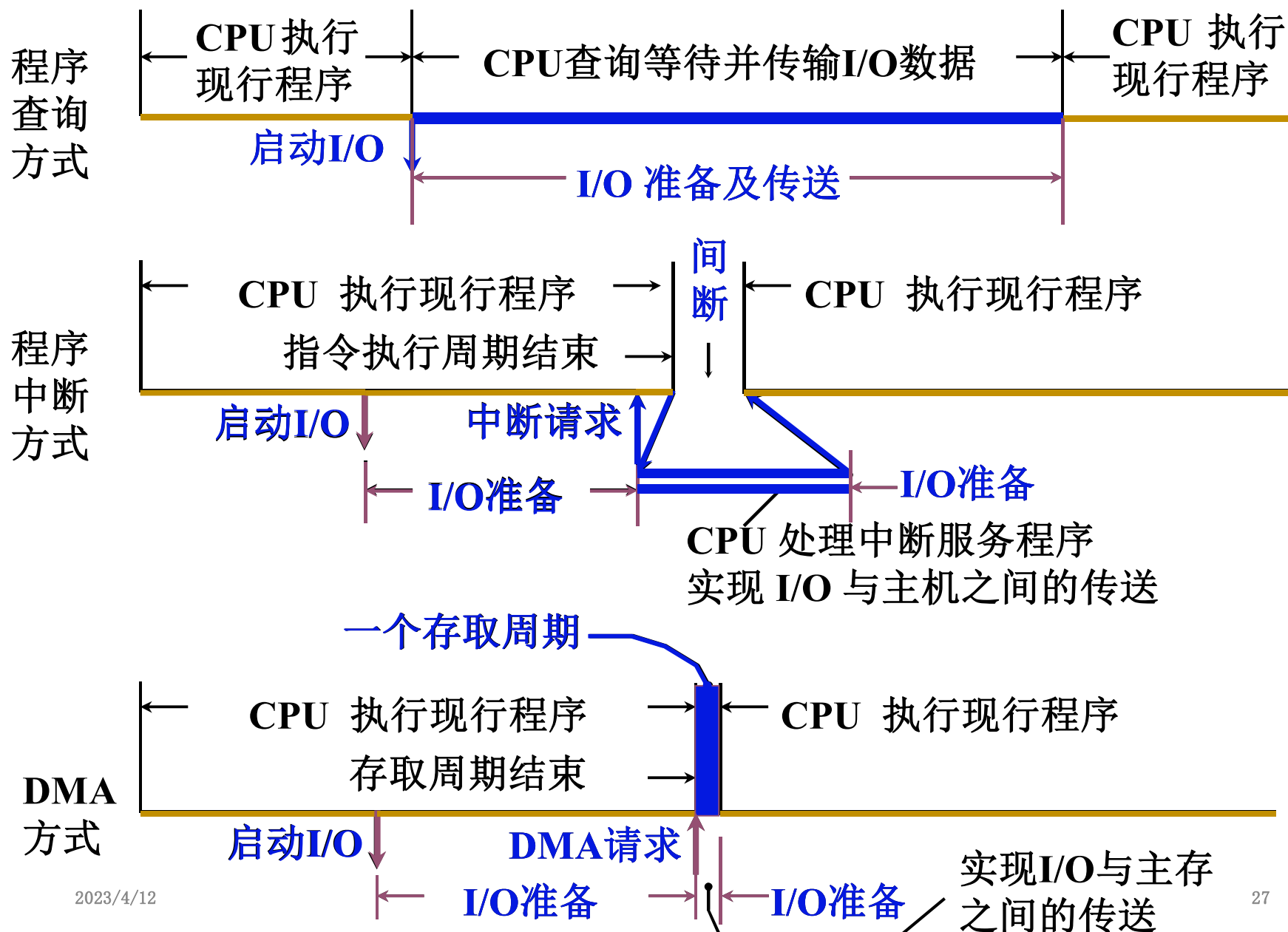
周期挪用（周期窃取）

**CPU 和 I/O 并行工作**



# 三种方式的 CPU 工作效率比较

## 5.1



## 5.3 I/O 接口

### 一、概述

为什么要设置接口？

1. 实现设备的选择
2. 实现数据缓冲达到速度匹配
3. 实现数据串一并格式转换
4. 实现电平转换
5. 传送控制命令
6. 反映设备的状态（“忙”、“就绪”、“中断请求”）

## 二、接口的功能和组成

## 5.3

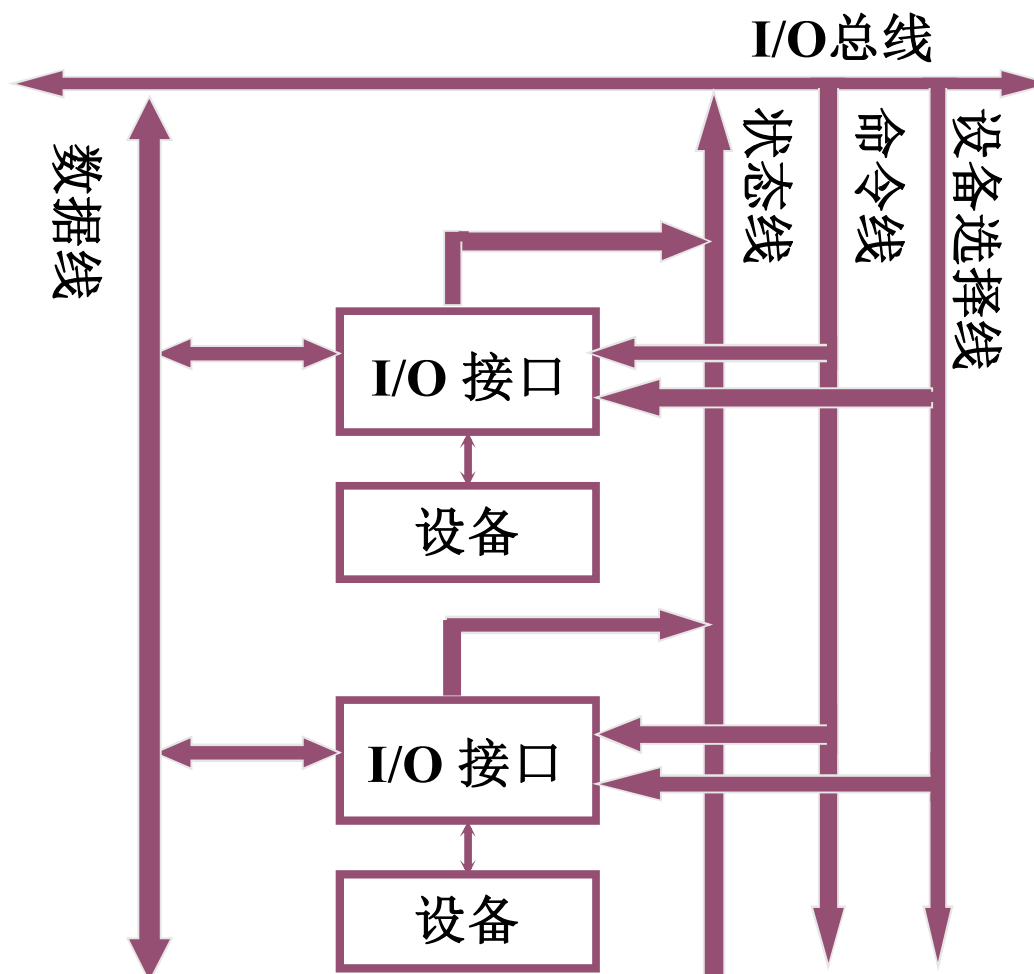
### 1. 总线连接方式的 I/O 接口电路

(1) 设备选择线

(2) 数据线

(3) 命令线

(4) 状态线



## 2. 接口的功能和组成

### 功能

选址功能

传送命令的功能

传送数据的功能

反映设备状态的功能

### 组成

设备选择电路

命令寄存器、命令译码器

数据缓冲寄存器

设备状态标记

完成触发器 **D**

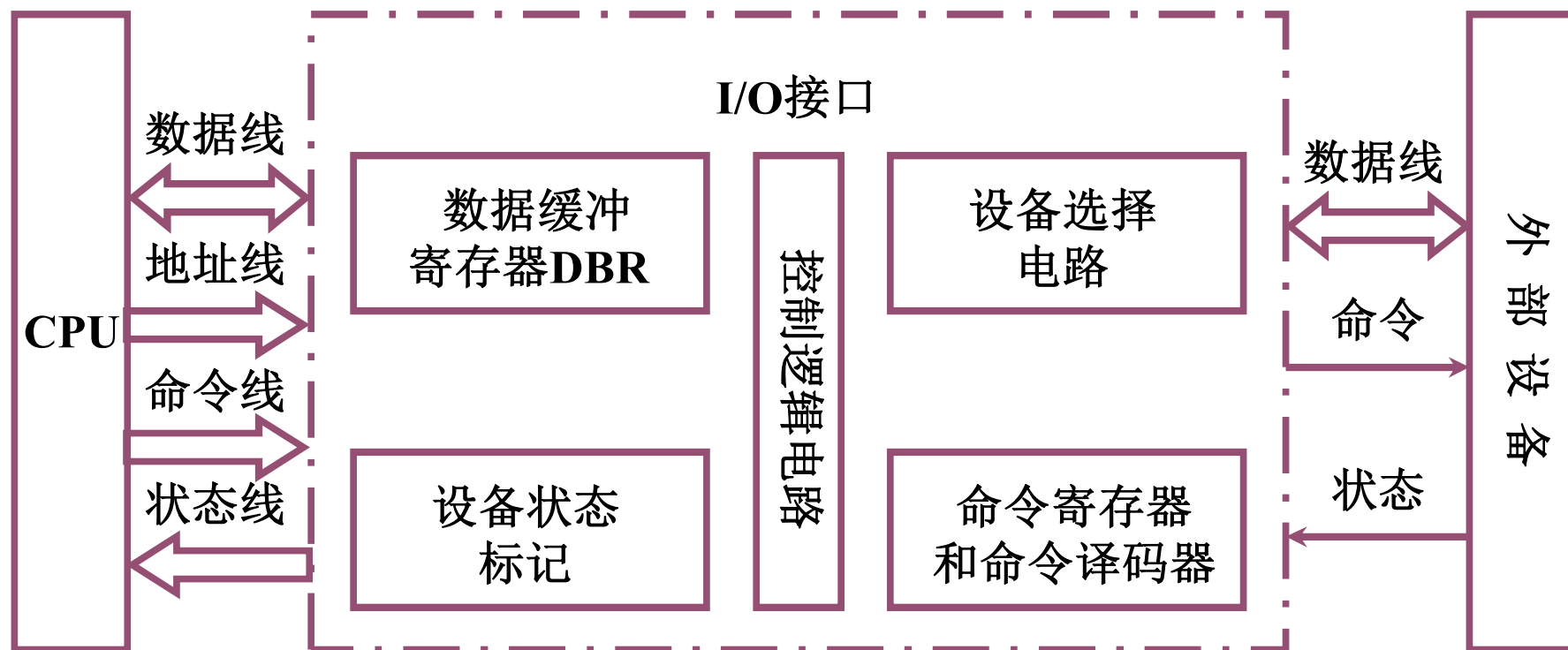
工作触发器 **B**

中断请求触发器 **INTR**

屏蔽触发器 **MASK**

### 3. I/O 接口的基本组成

## 5.3



## 三、接口类型

## 5.3

### 1. 按数据 传送方式 分类

并行接口          Intel 8255

串行接口          Intel 8251

### 2. 按功能 选择的灵活性 分类

可编程接口      Intel 8255、 Intel 8251

不可编程接口   Intel 8212

### 3. 按 通用性 分类

通用接口          Intel 8255、 Intel 8251

专用接口          Intel 8279、 Intel 8275

### 4. 按数据传送的 控制方式 分类

中断接口          Intel 8259

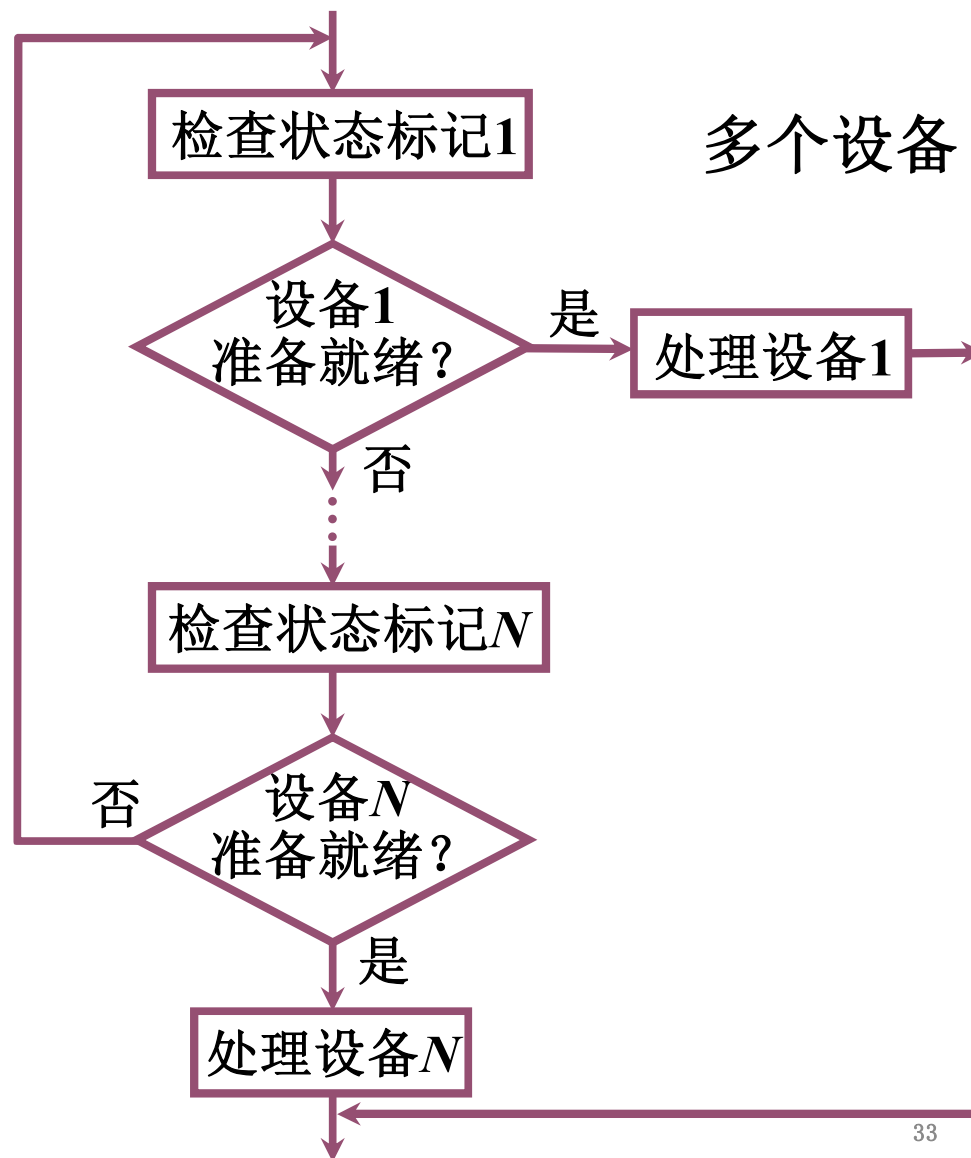
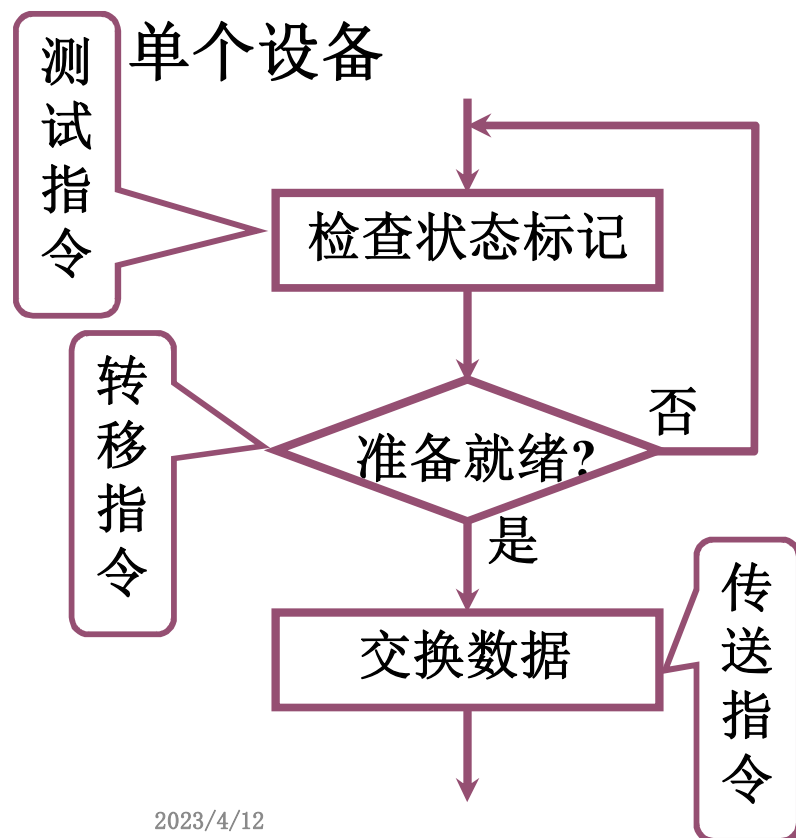
DMA 接口          Intel 8257



## 5.4 程序查询方式

### 一、程序查询流程

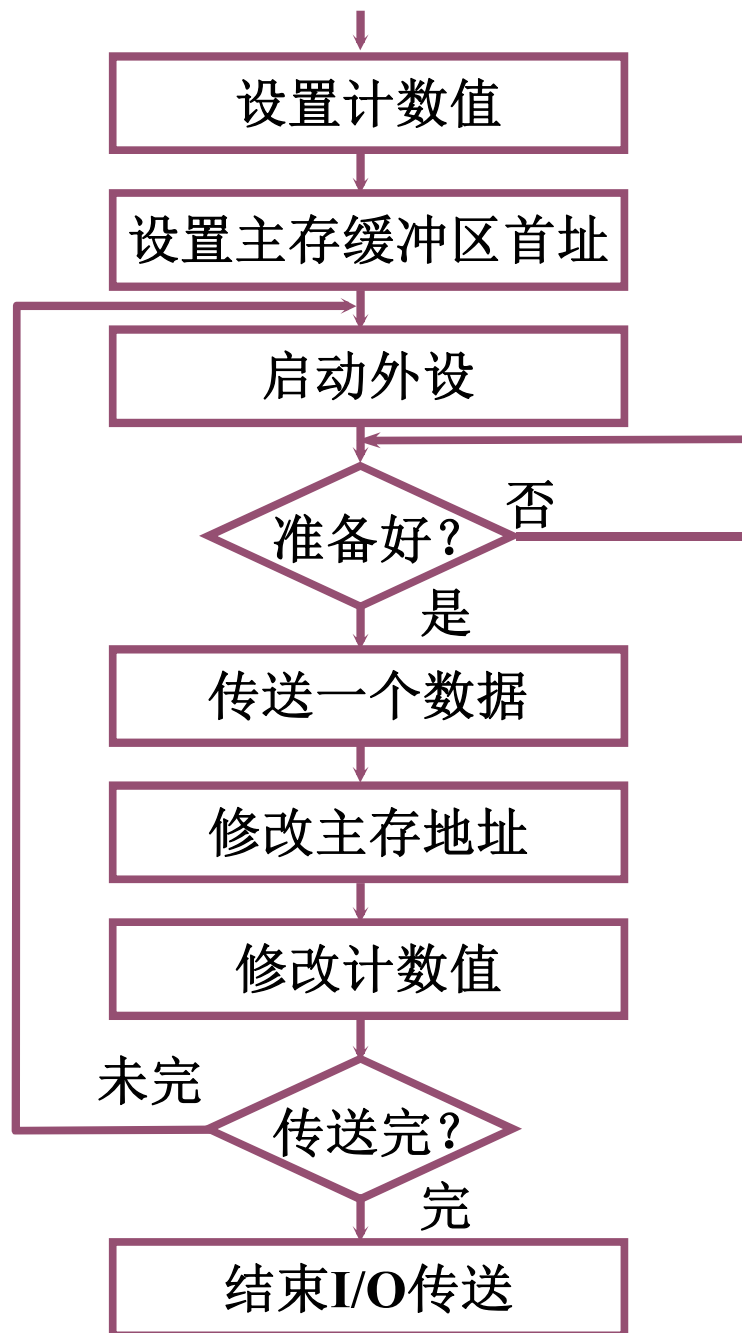
#### 1. 查询流程



## 2. 程序流程

5.4

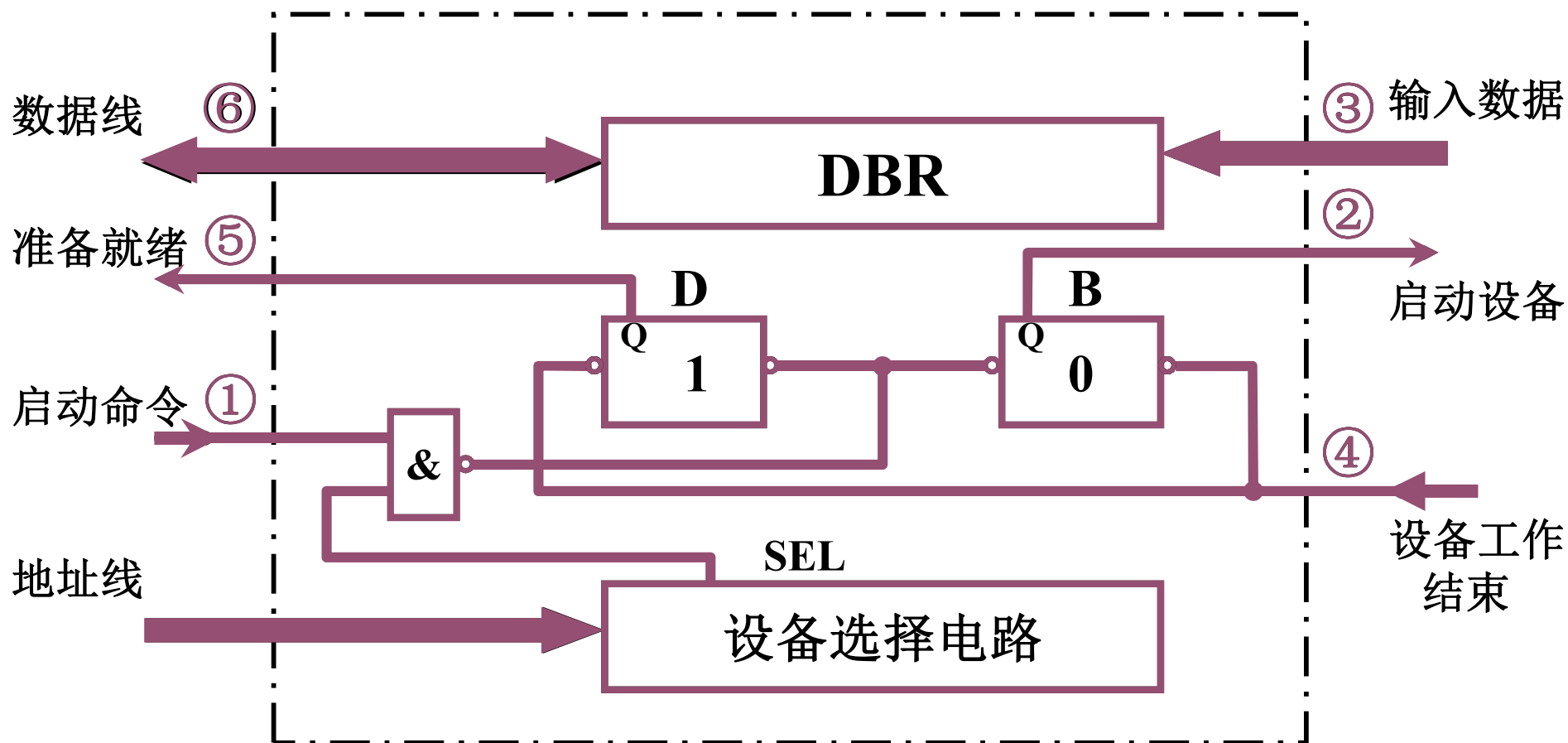
保存  
寄存器内容



## 二、程序查询方式的接口电路

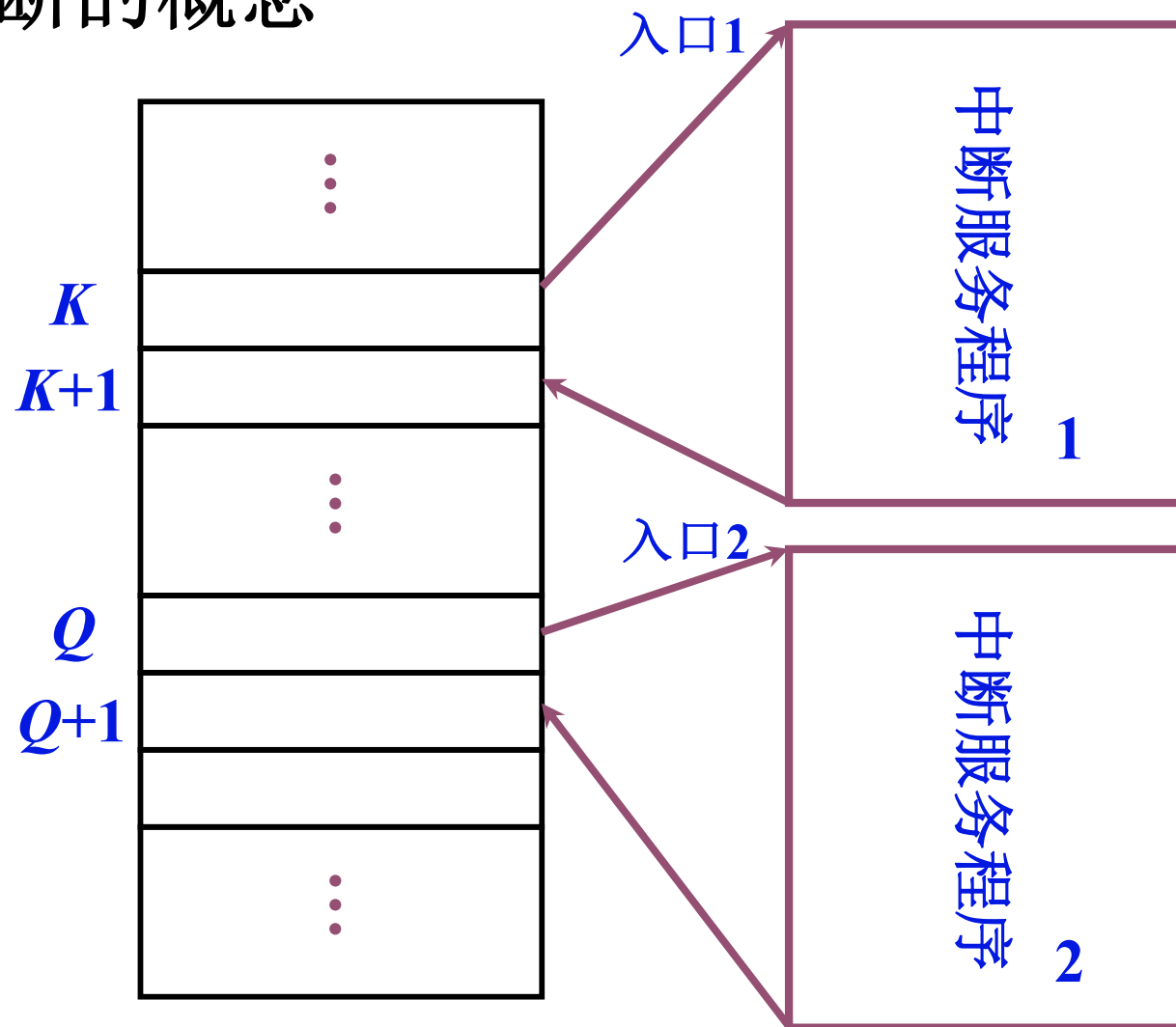
## 5.4

以输入为例



## 5.5 程序中断方式

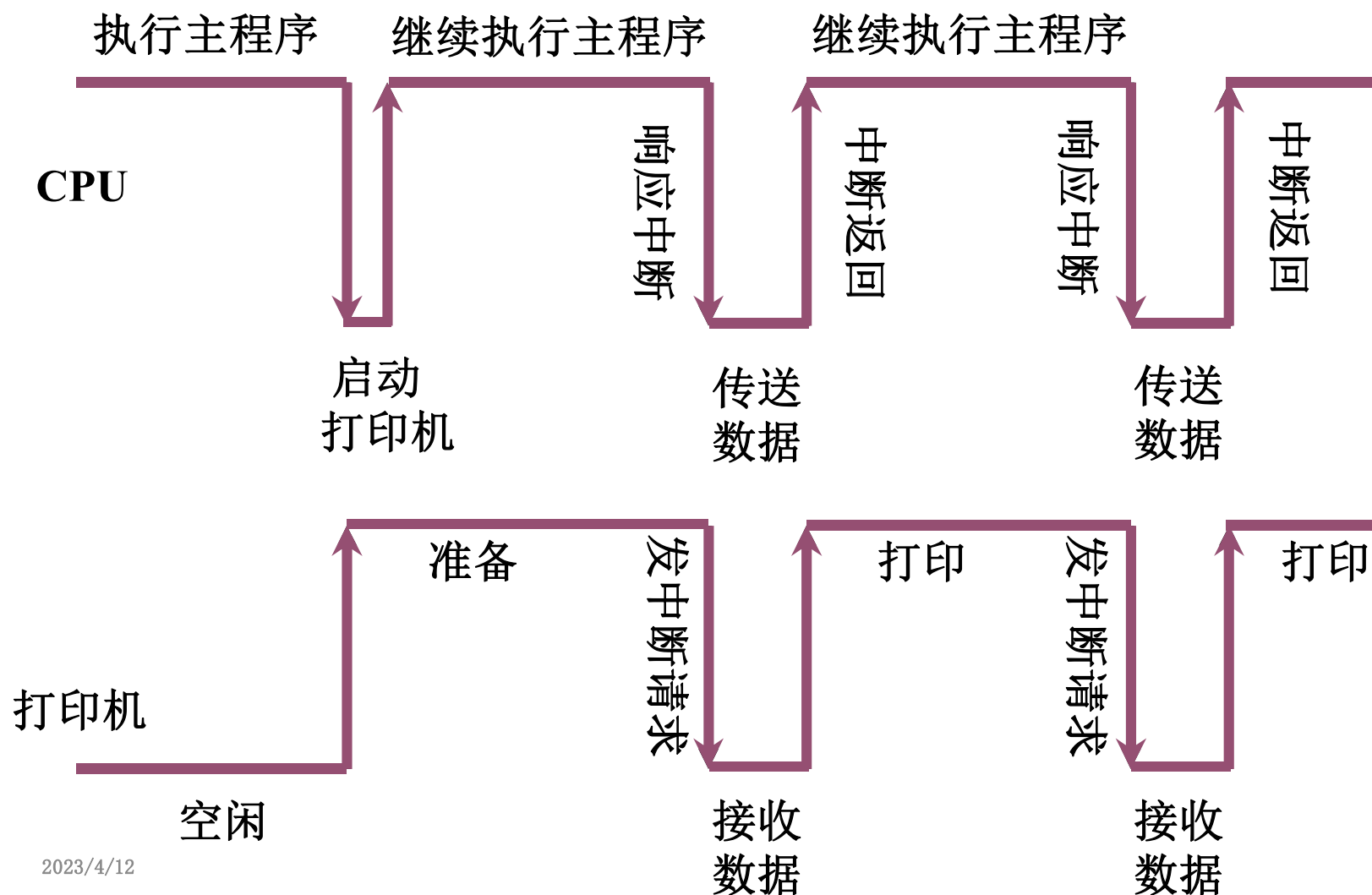
### 一、中断的概念



## 二、I/O 中断的产生

5.5

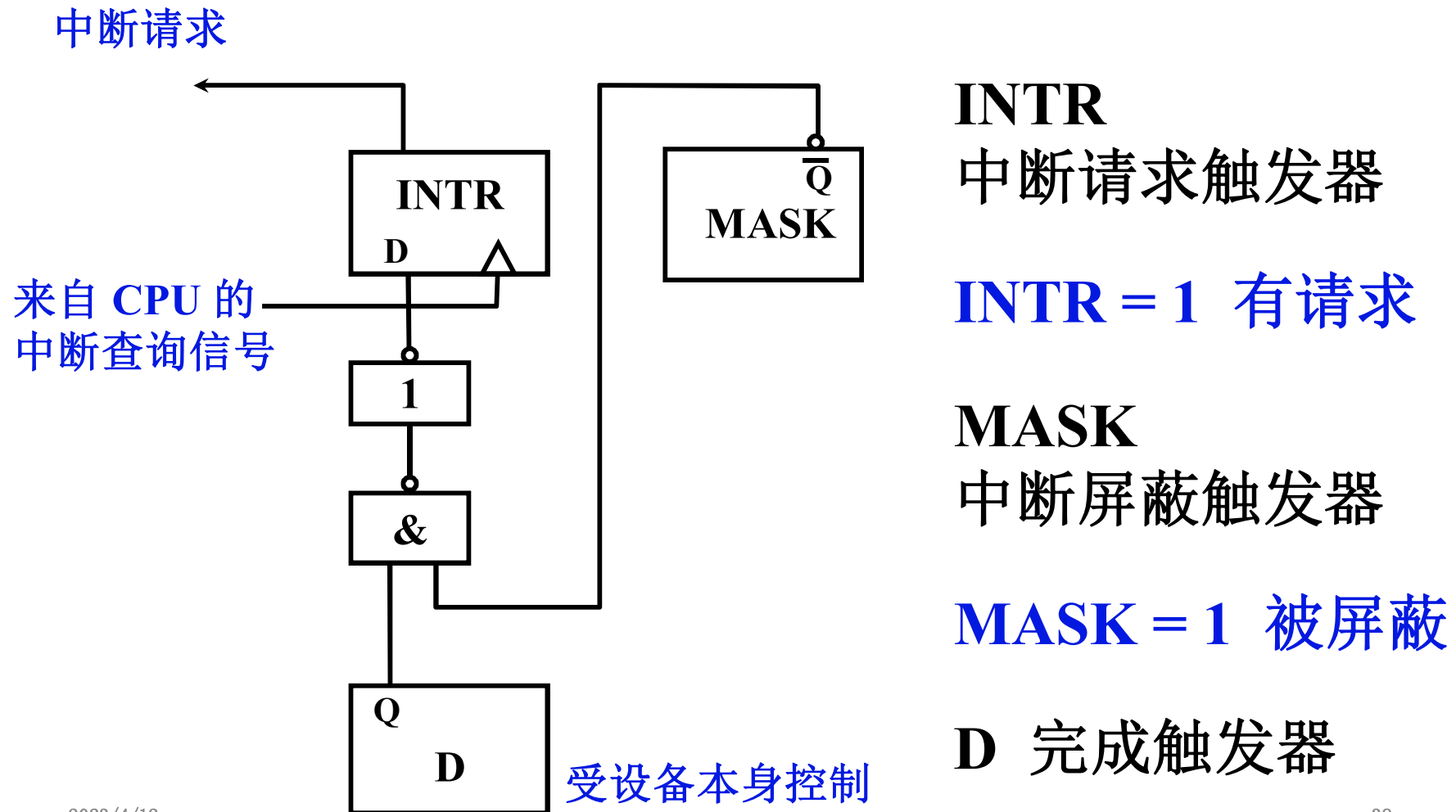
以打印机为例 CPU 与打印机并行工作



### 三、程序中断方式的接口电路

## 5.5

#### 1. 配置中断请求触发器和中断屏蔽触发器



## 2. 排队器

5.5

排队 { 硬件 在 CPU 内或在接口电路中（链式排队器）  
软件 通过中断识别程序判断优先级

## 3. 中断向量地址形成部件

入口地址 { 由软件产生 按请求来源跳转到对应入口点  
硬件向量法 由 硬件 产生 向量地址  
再由 向量地址 找到 入口地址

## 四、I/O 中断处理过程

## 5.5

### 1. CPU 响应中断的条件和时间

#### (1) 条件

允许中断触发器 **EINT = 1**

用 **开中断** 指令将 **EINT** 置 “**1**”

用 **关中断** 指令将 **EINT** 置 “**0**”  
(或由硬件 **自动复位**)

#### (2) 时间

当 **D = 1** (随机发生) 且 **MASK = 0** 时

在每条指令执行阶段的结束前

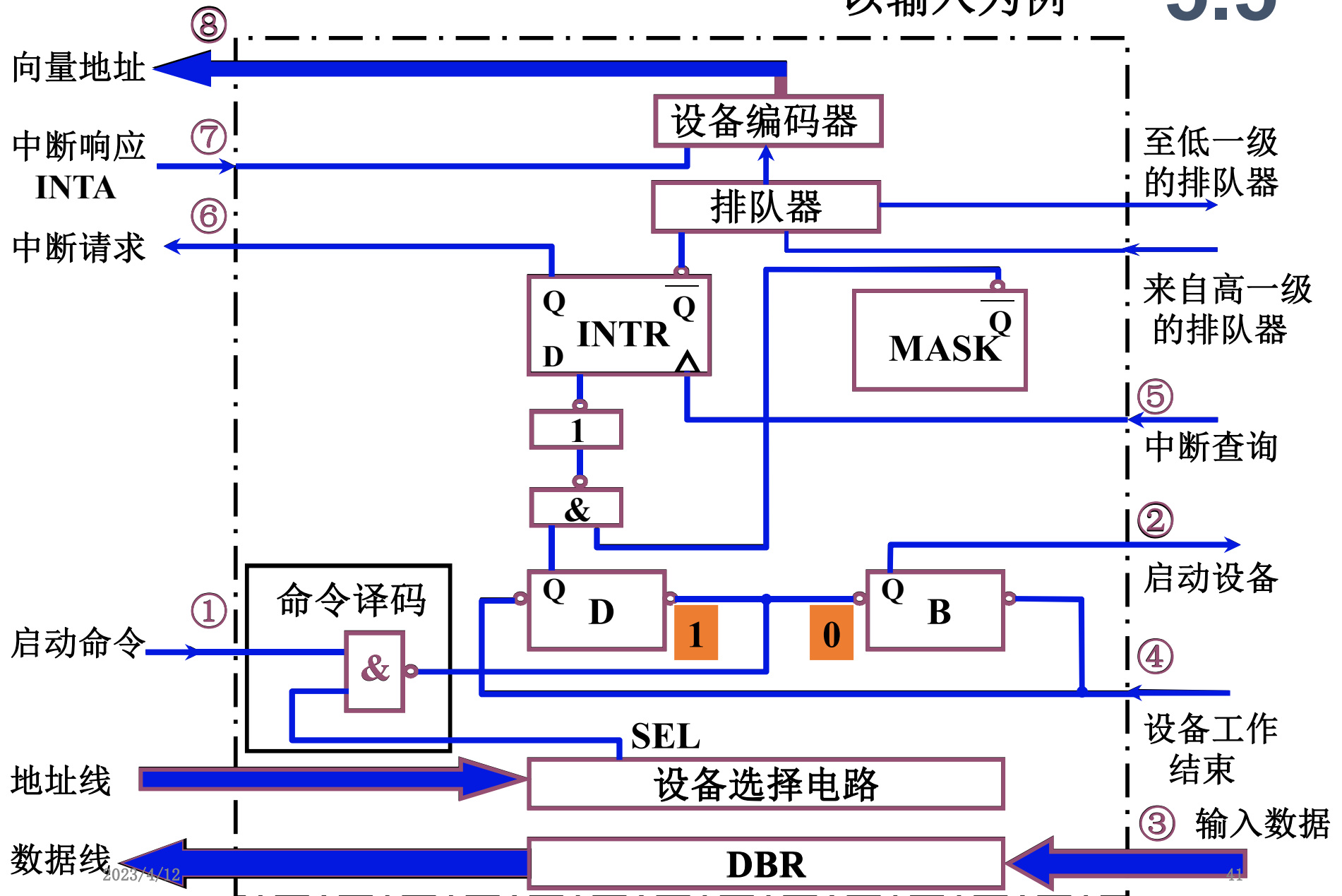
**CPU** 发 **中断查询信号** (将 **INTR** 置 “**1**” )



## 2. I/O 中断处理过程

以输入为例

5.5



# 五、中断服务程序流程

## 5.5

### 1. 中断服务程序的流程

#### (1) 保护现场

{	程序断点的保护	中断隐指令完成
	寄存器内容的保护	进栈指令

#### (2) 中断服务

对不同的 I/O 设备具有不同内容的设备服务

#### (3) 恢复现场

出栈指令

#### (4) 中断返回

中断返回指令

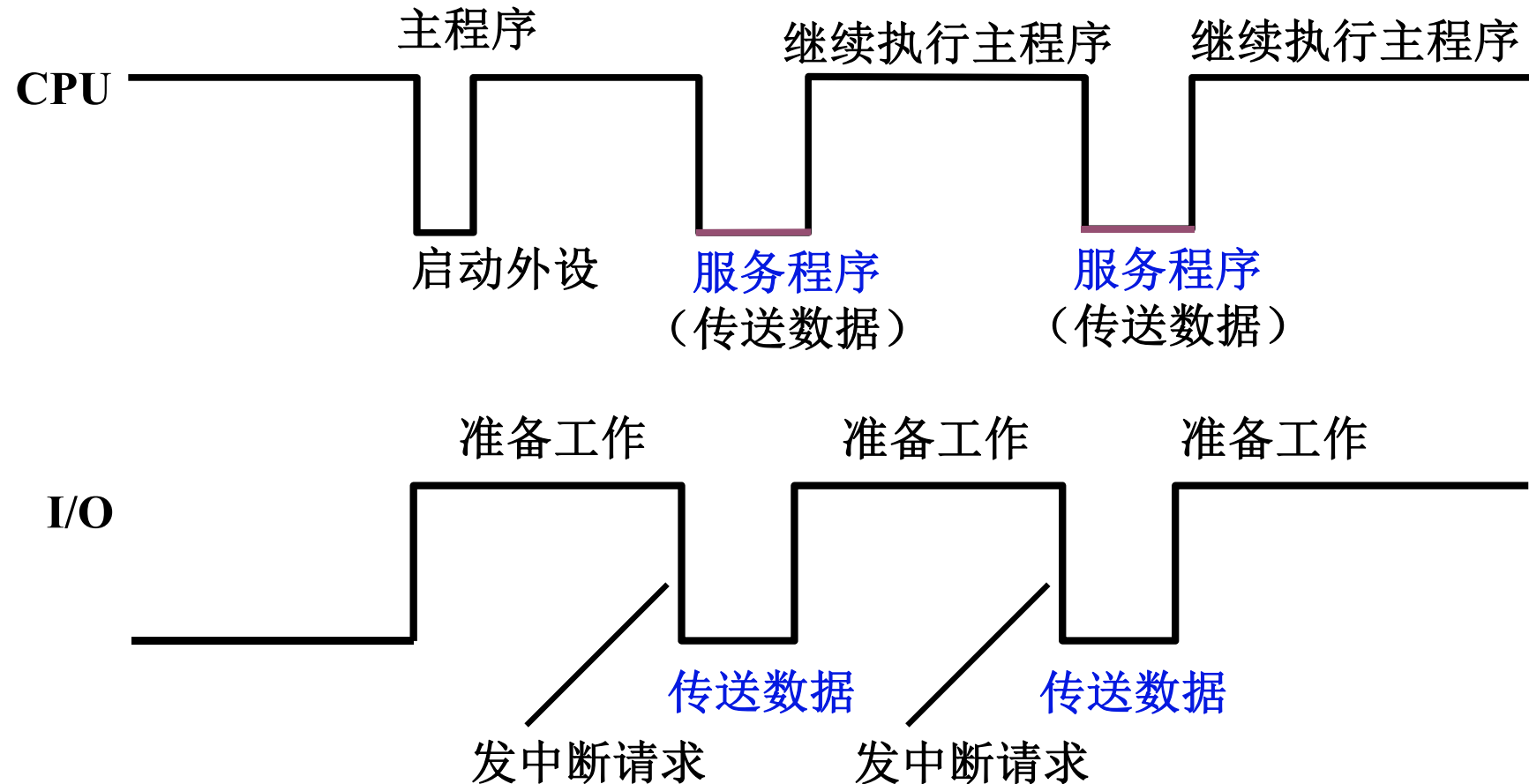
### 2. 单重中断和多重中断

单重 中断 不允许中断 现行的 中断服务程序

多重 中断 允许级别更高 的中断源

中断 现行的 中断服务程序

# 主程序和服务程序抢占 CPU 示意图 5.5



宏观上 CPU 和 I/O 并行工作

微观上 CPU 中断现行政程序为 I/O 服务