

计算机组成原理

顾崇林

计算机科学与技术学院

guchonglin@hit.edu.cn

第八章 输入输出系统

8.1 概述

8.2 外部设备

8.3 I/O接口

8.4 程序查询方式

8.5 程序中断方式

8.6 DMA方式

8.1概述

- 输入输出系统的发展概况
- 输入输出系统的组成
- I/O设备与主机的联系方式
- I/O设备与主机信息传送的控制方式

8.1概述

一、输入输出系统的发展概况

I. 早期

分散连接

CPU 和 I/O设备 串行 工作 程序查询方式

II. 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

III. 具有通道结构的阶段

IV. 具有 I/O 处理机的阶段

8.1概述

二、输入输出系统的组成

I. 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

II. I/O 硬件

设备

I/O 接口

设备

设备控制器

通道

8.1概述

三、I/O 设备与主机的联系方式

I. 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

II. 设备选址

用设备选择电路识别是否被选中

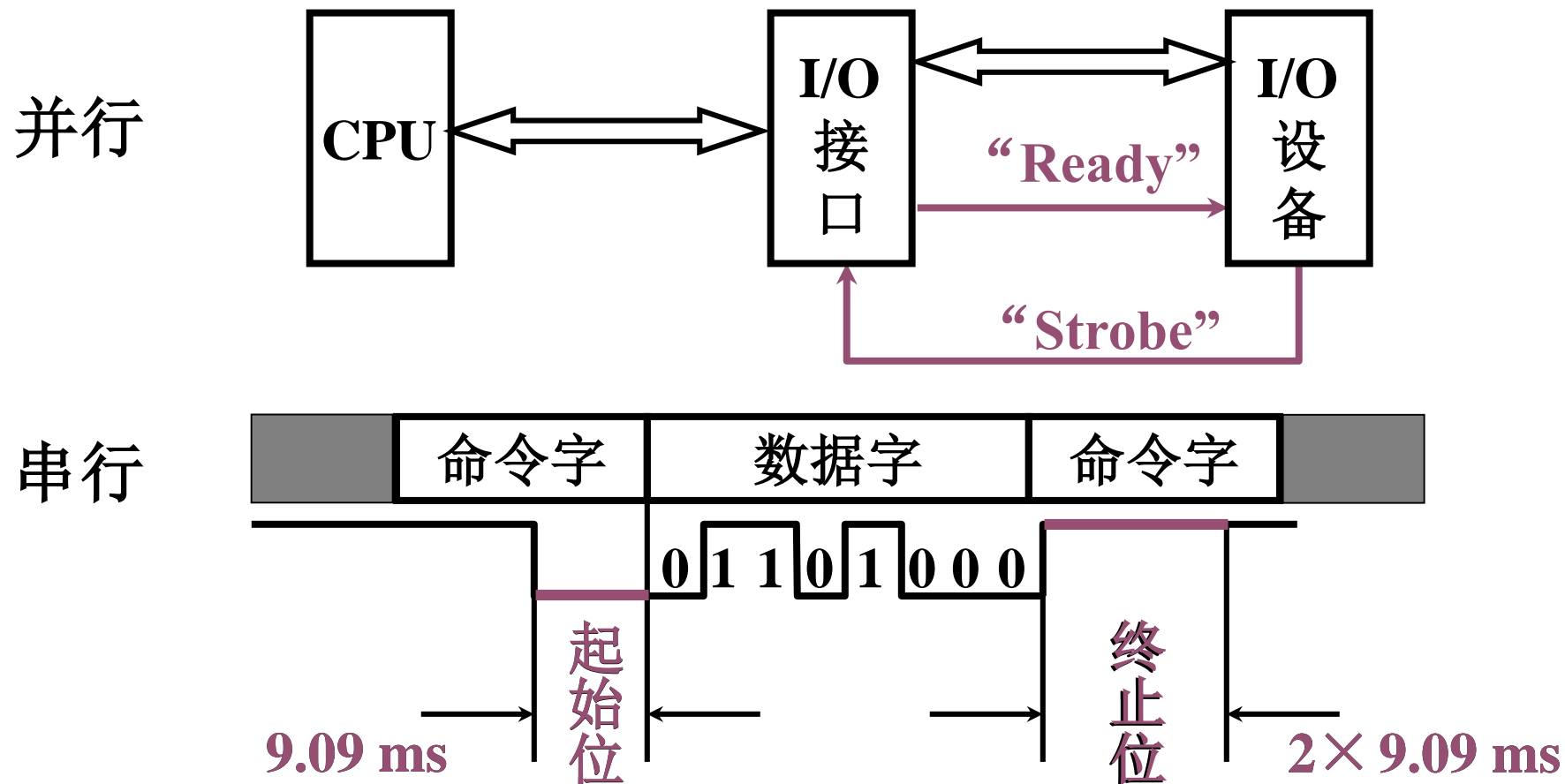
III. 传送方式

- (1) 串行
- (2) 并行

IV. 联络方式

(1) 立即响应

(2) 异步工作采用应答信号

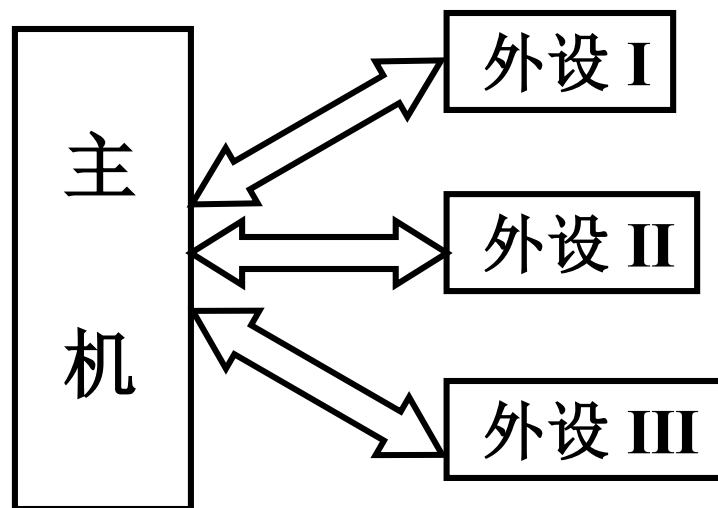


(3) 同步工作采用同步时标

8.1概述

V. I/O 设备与主机的连接方式

(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

不利于增删设备

(2) 总线连接

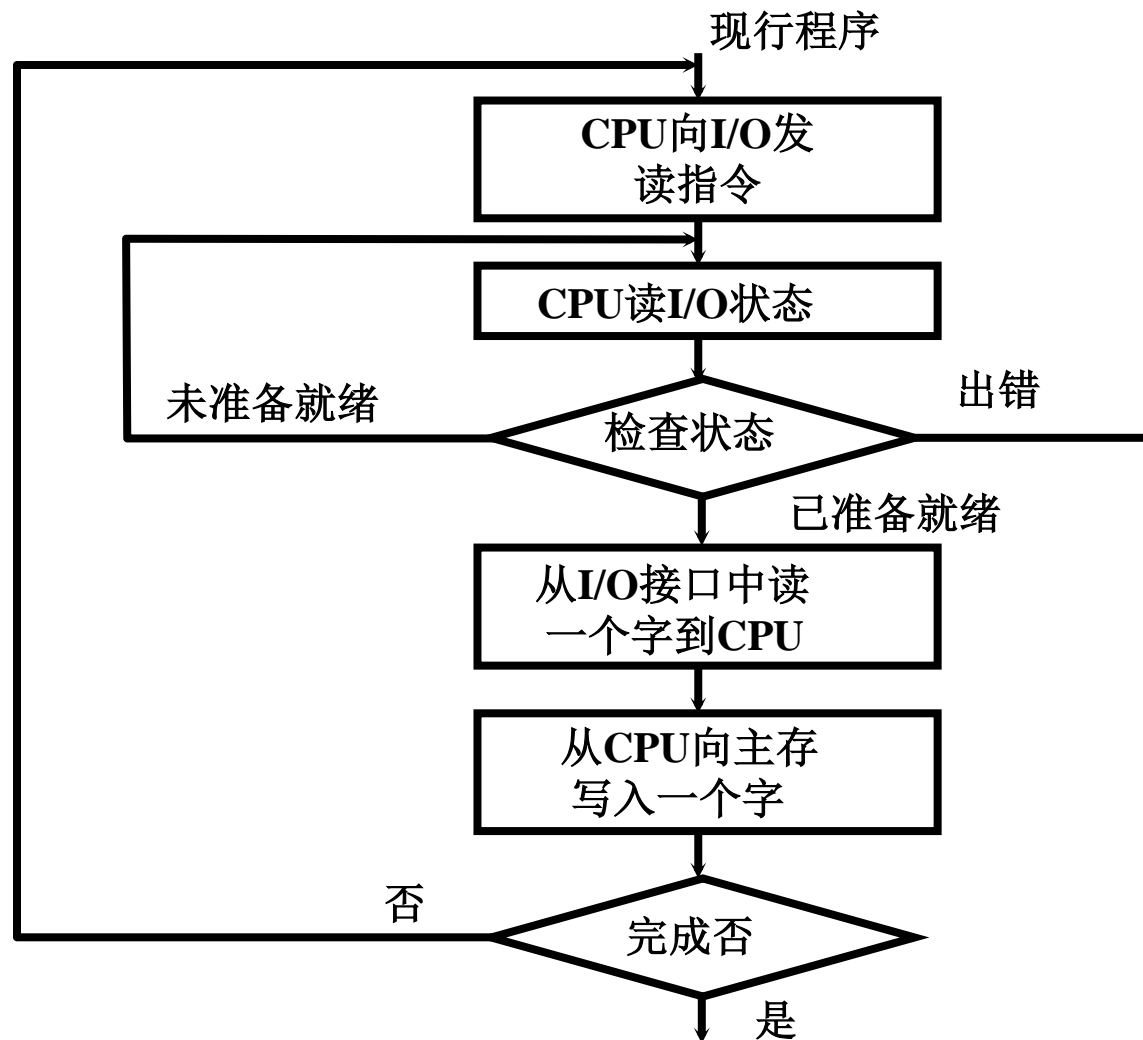
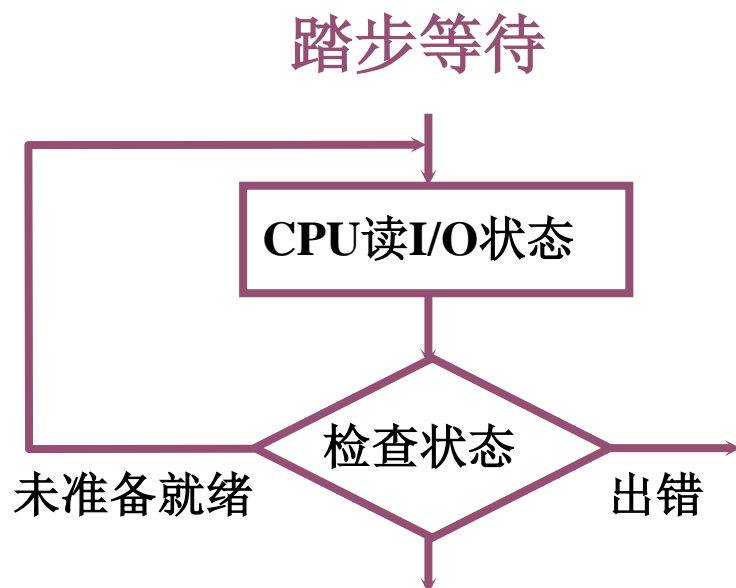
便于增删设备

8.1概述

四、I/O设备与主机信息传送的控制方式

I. 程序查询方式

CPU 和 I/O 串行工作

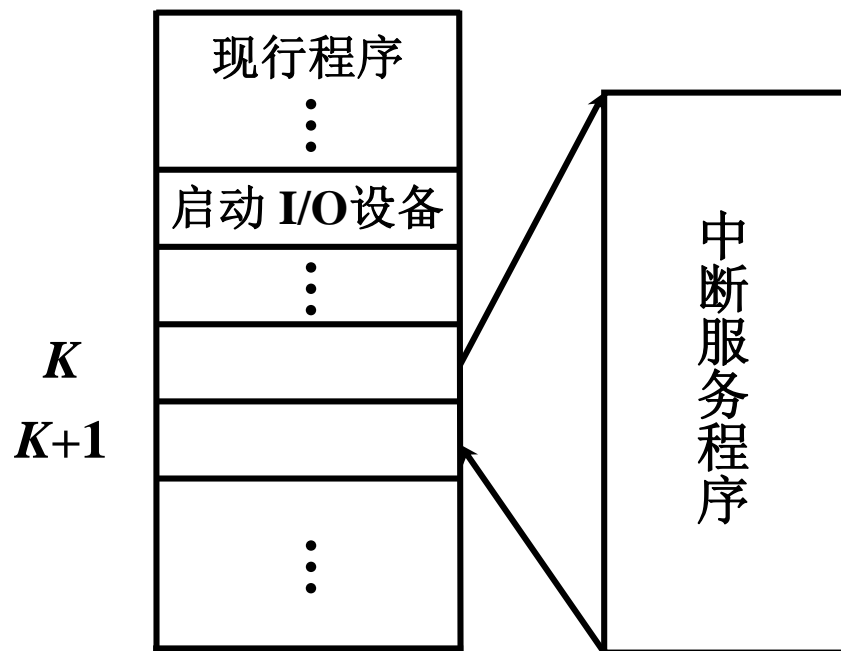


8.1 概述

II. 程序中断方式

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

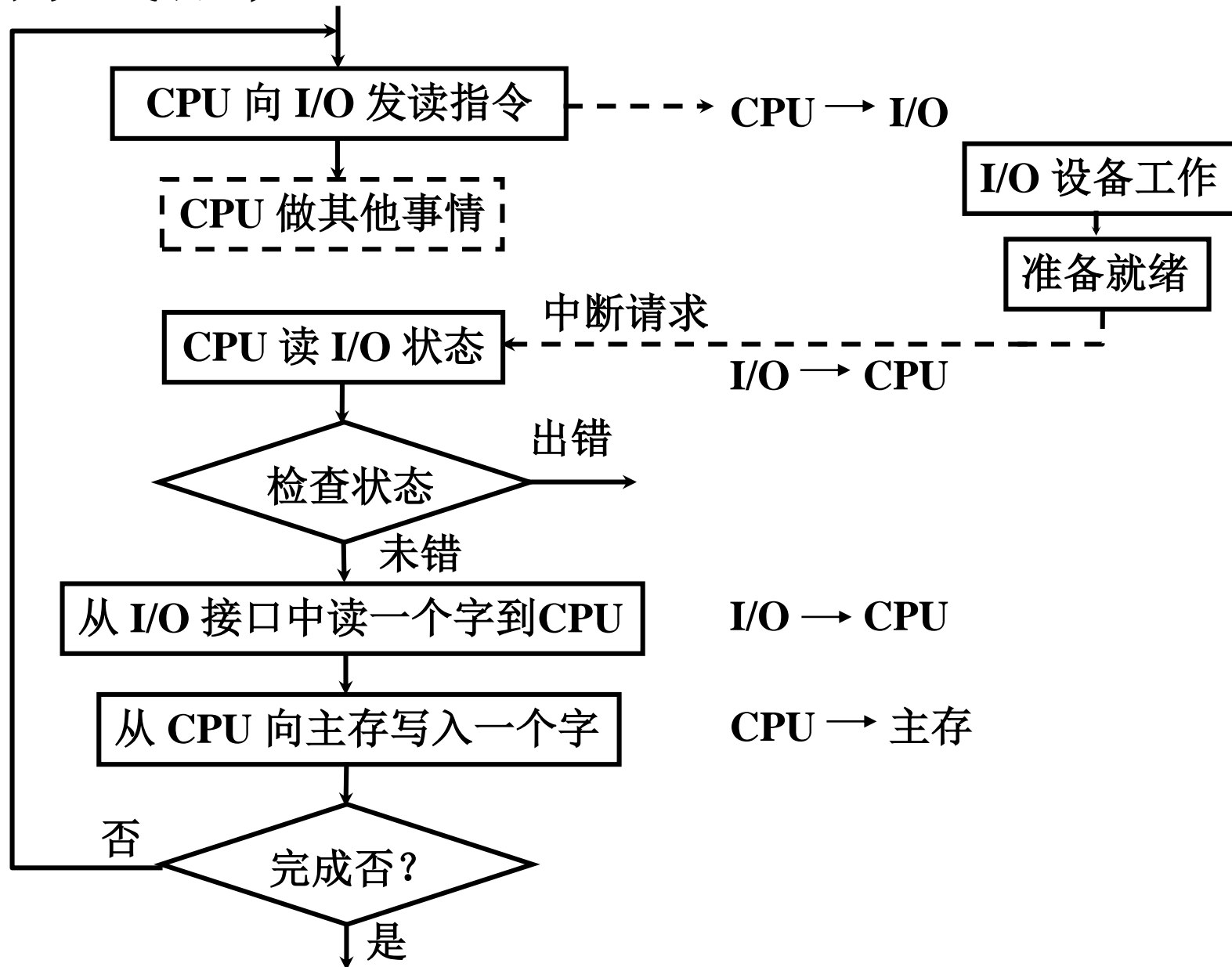
CPU 和 I/O 并行工作



没有踏步等待现象

中断现行政序

程序中断方式流程



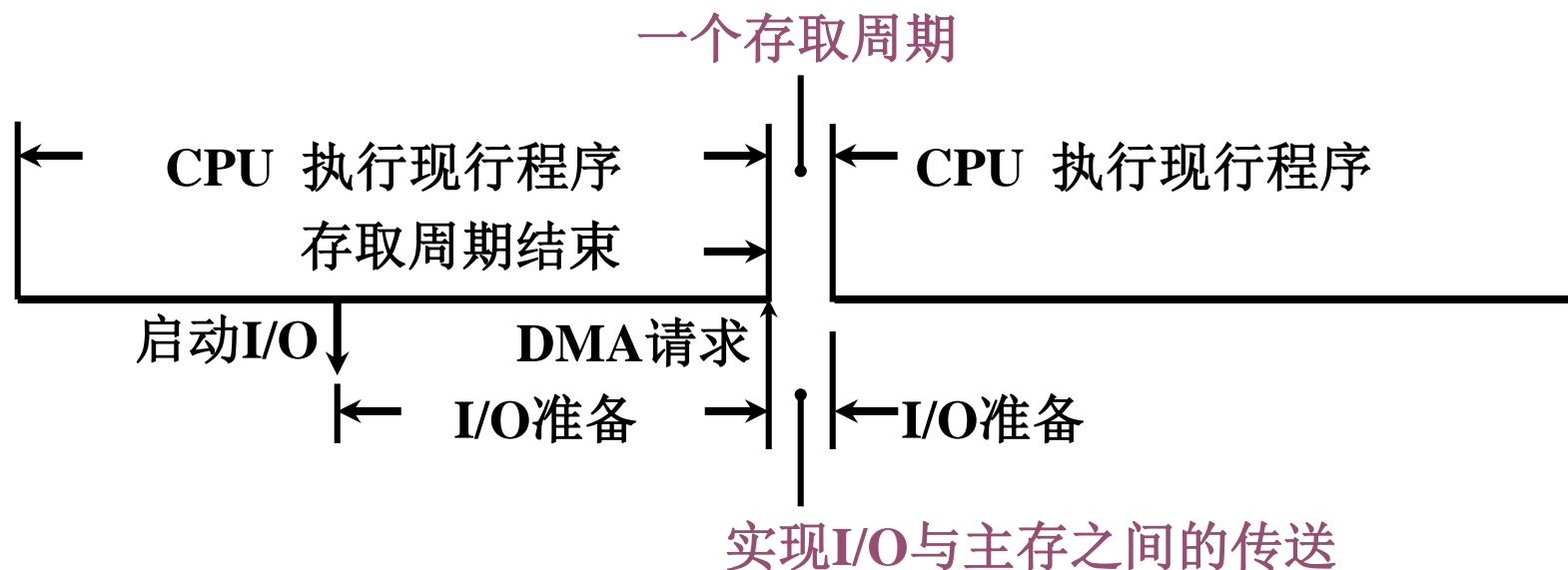
III. DMA 方式

主存和 I/O 之间有一条直接数据通道

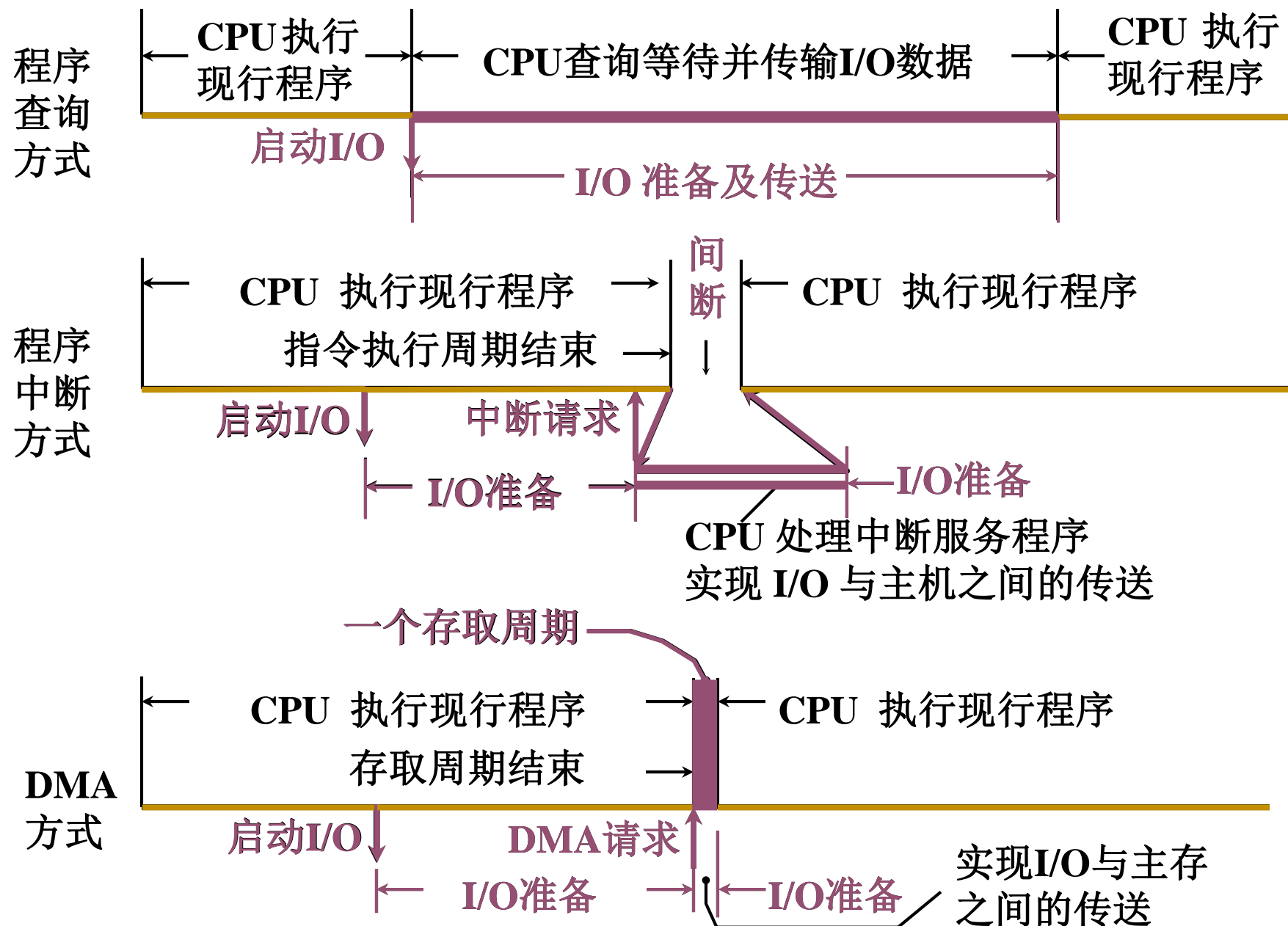
不中断现行程序

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



第八章 输入输出系统

8.1 概述

8.2 外部设备

8.3 I/O接口

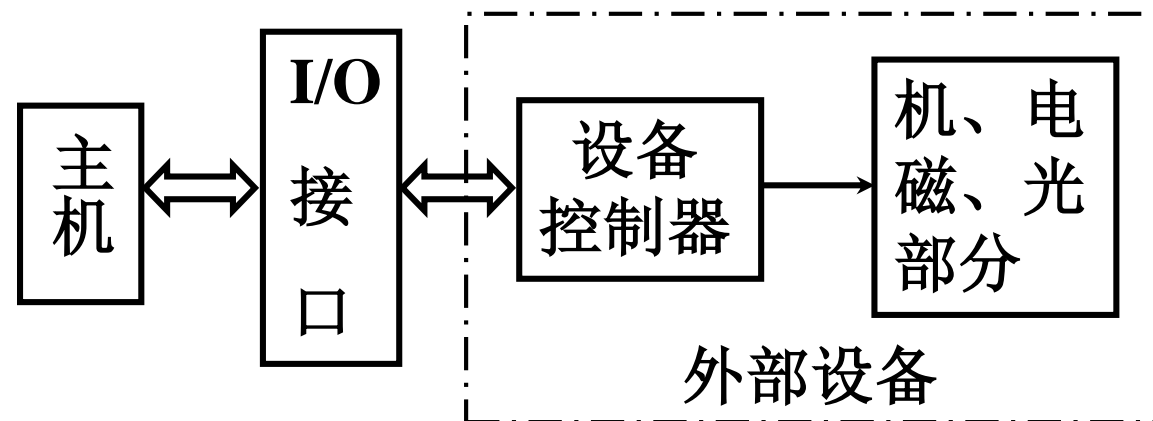
8.4 程序查询方式

8.5 程序中断方式

8.6 DMA方式

8.2 外部设备

一、概述



外部设备大致分三类

- | | |
|--------------|---------------|
| 1. 人机交互设备 | 键盘、鼠标、打印机、显示器 |
| 2. 计算机信息存储设备 | 磁盘、光盘、磁带 |
| 3. 机—机通信设备 | 调制解调器等 |

8.2外部设备

二、输入设备

1. 键盘

按键

判断哪个键按下

将此键翻译成 ASCII 码（编码键盘法）

2. 鼠标

机械式

金属球 电位器

光电式

光电转换器

3. 触摸屏

8.2外部设备

三、输出设备

1. 显示器

(1) 字符显示

字符发生器

(2) 图形显示

主观图像

(3) 图像显示

客观图像

2. 打印机

(1) 击打式

点阵式（逐字、逐行）

(2) 非击打式

激光（逐页） 喷墨（逐字）

8.2外部设备

四、其他

- 1. A/D、D/A 模拟/数字（数字/模拟）转换器
- 2. 终端 由键盘和显示器组成
 完成显示控制与存储、键盘管理及通信控制
- 3. 汉字处理 汉字输入、汉字存储、汉字输出

五、多媒体技术

- 1. 什么是多媒体
- 2. 多媒体计算机的关键技术

第八章 输入输出系统

8.1 概述

8.2 外部设备

8.3 I/O接口

8.4 程序查询方式

8.5 程序中断方式

8.6 DMA方式

8.3 I/O接口

一、概述

为什么要设置接口？

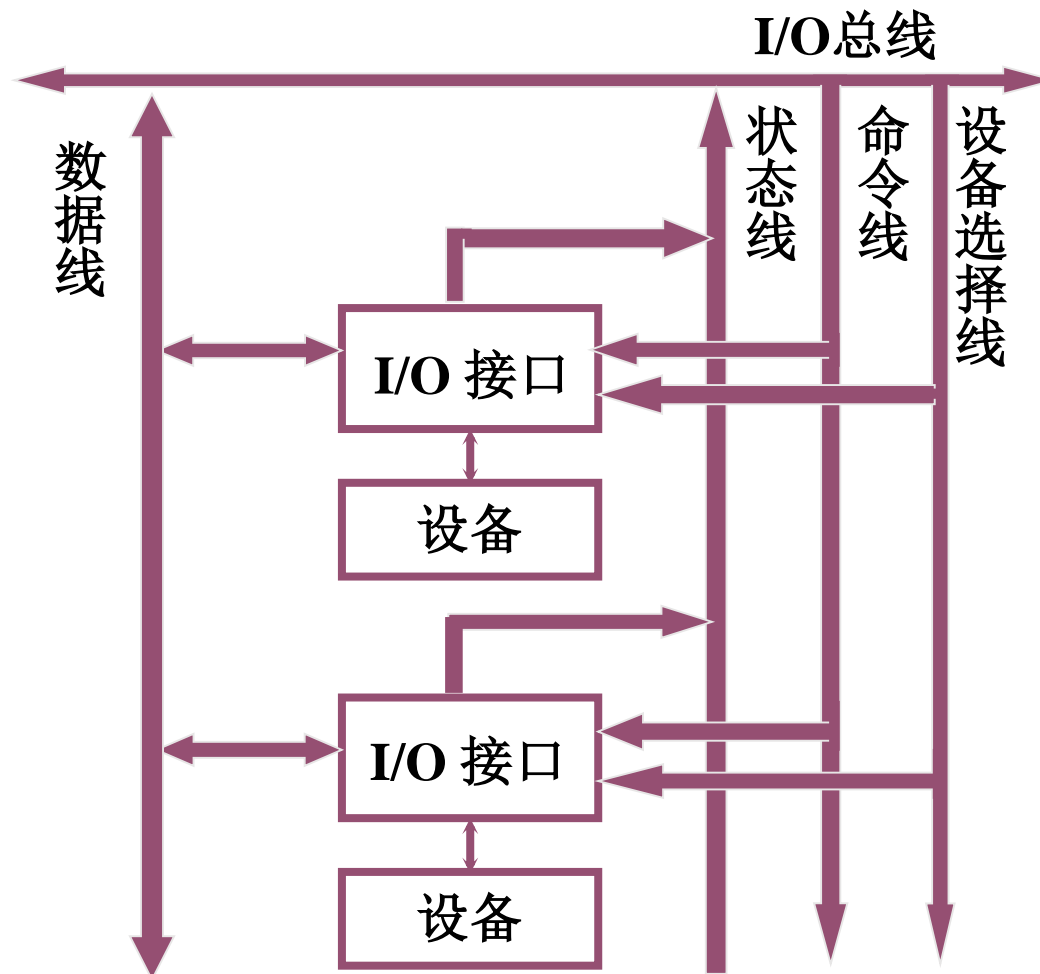
1. 实现设备的选择
2. 实现数据缓冲达到速度匹配
3. 实现数据串一并格式转换
4. 实现电平转换
5. 传送控制命令
6. 反映设备的状态（“忙”、“就绪”、“中断请求”）

8.3 I/O接口

二、接口的功能和组成

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

{ 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**

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

3. I/O 接口的基本组成



三、接口类型

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 8237

第八章 输入输出系统

8.1 概述

8.2 外部设备

8.3 I/O接口

8.4 程序查询方式

8.5 程序中断方式

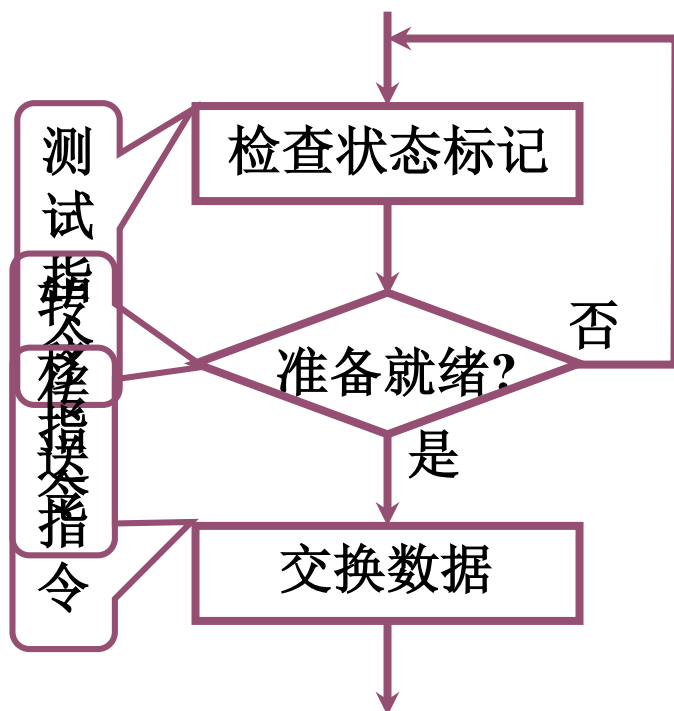
8.6 DMA方式

8.4 程序查询方式

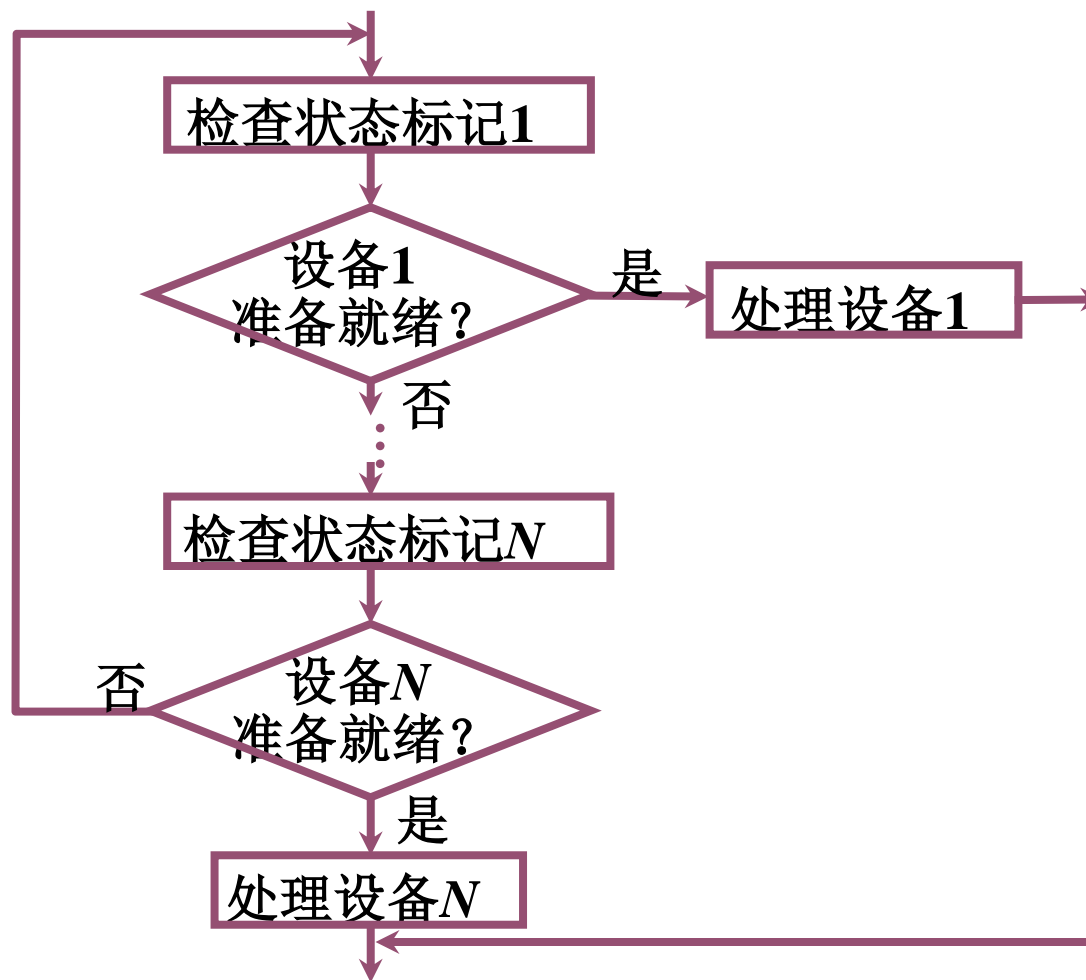
一、程序查询流程

1. 查询流程

单个设备

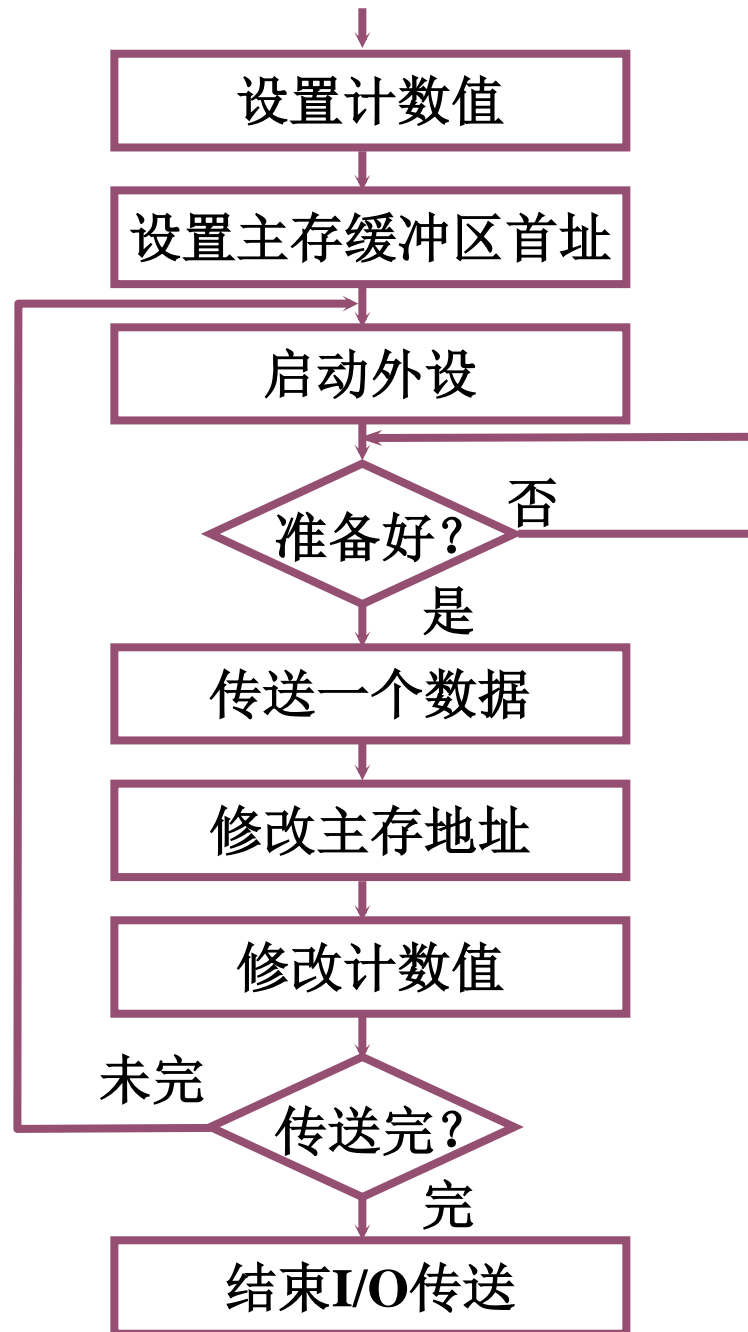


多个设备



2. 程序流程

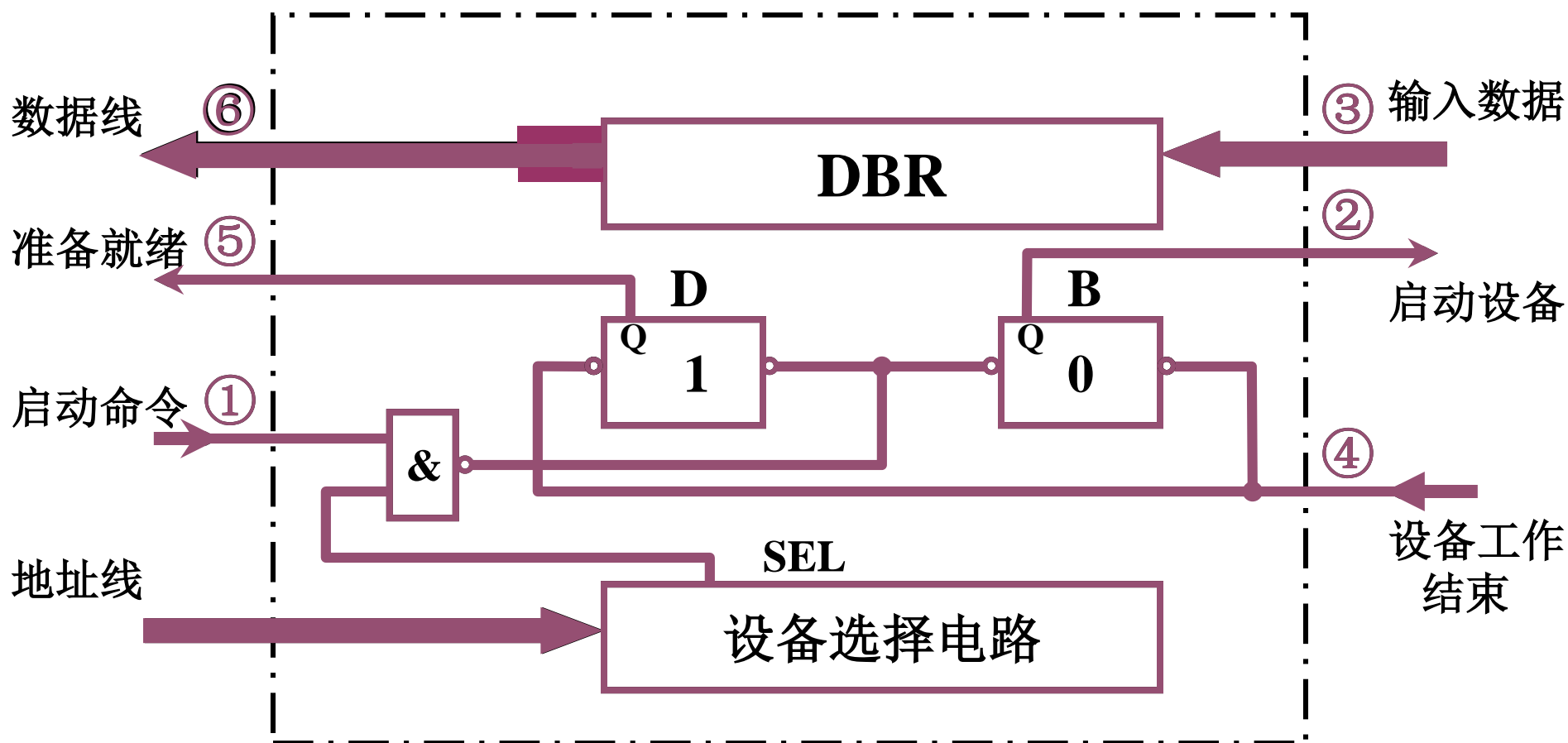
保存
寄存器内容



8.4 程序查询方式

二、程序查询方式的接口电路

以输入为例



例8.1 CPU 的时钟频率为 50 MHz

每个查询操作需要 100 个时钟周期

鼠标 30 次/秒查询

硬盘 以 32 位字长为单位传输数据 传输率为 2 MBps

求 CPU 对这两个设备查询所花费的时间比率

由此得出什么结论？

解：50 MHz 相当于 50×10^6 个时钟周期/秒

比率 $[3000 / (50 \times 10^6)] \times 100\% = 0.006\%$

(2) 硬盘 每秒查询 $2\text{MB}/4\text{B} = 512\text{ K}$ 次

占 $100 \times 512 \times 1024 = 52.4 \times 10^6$ 时钟周期/秒

比率 $[(52.4 \times 10^6) / (50 \times 10^6)] \times 100\% = 105\%$

结论：CPU 一般不采用 程序查询方式 与磁盘交换信息

第八章 输入输出系统

8.1 概述

8.2 外部设备

8.3 I/O接口

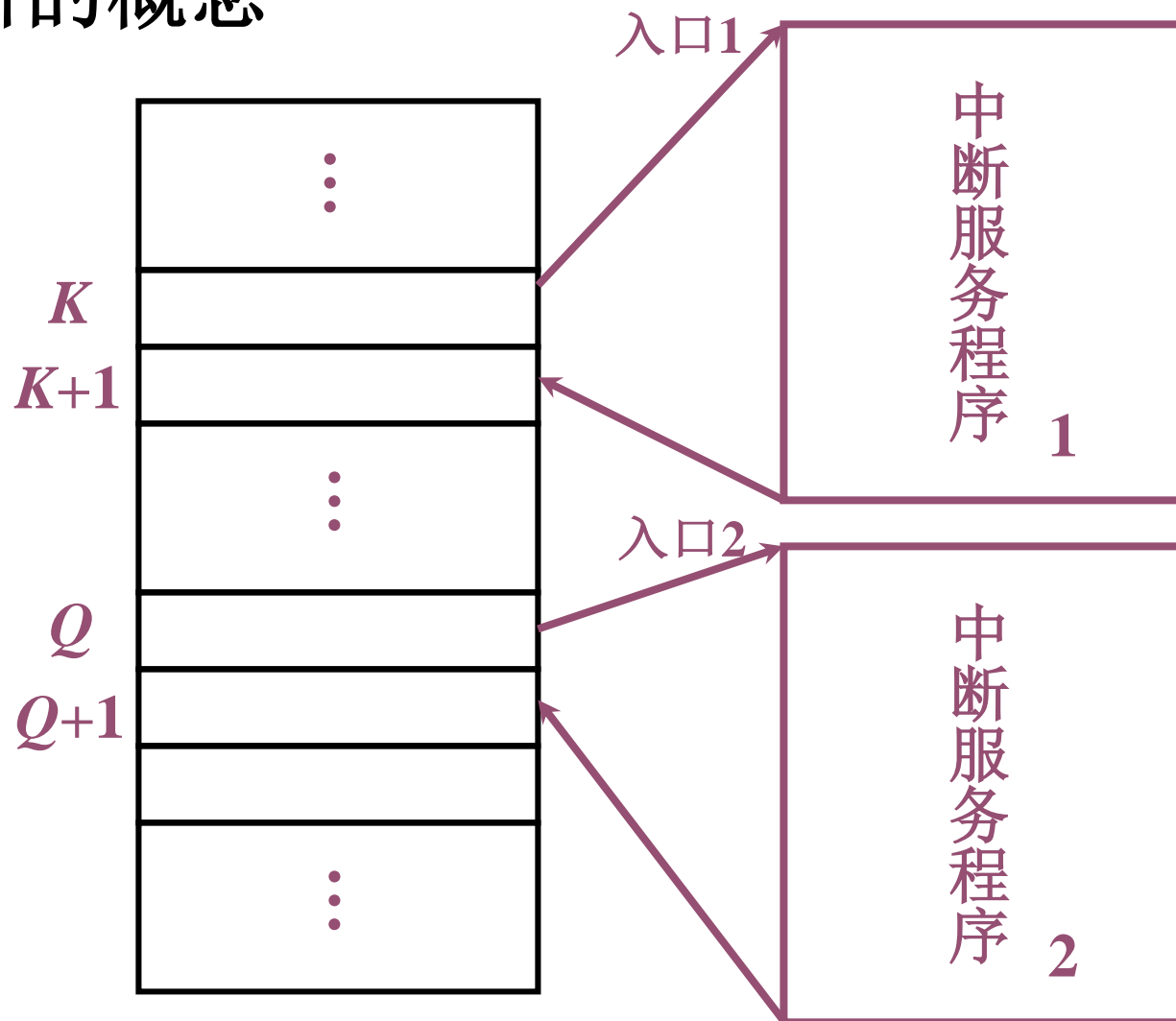
8.4 程序查询方式

8.5 程序中断方式

8.6 DMA方式

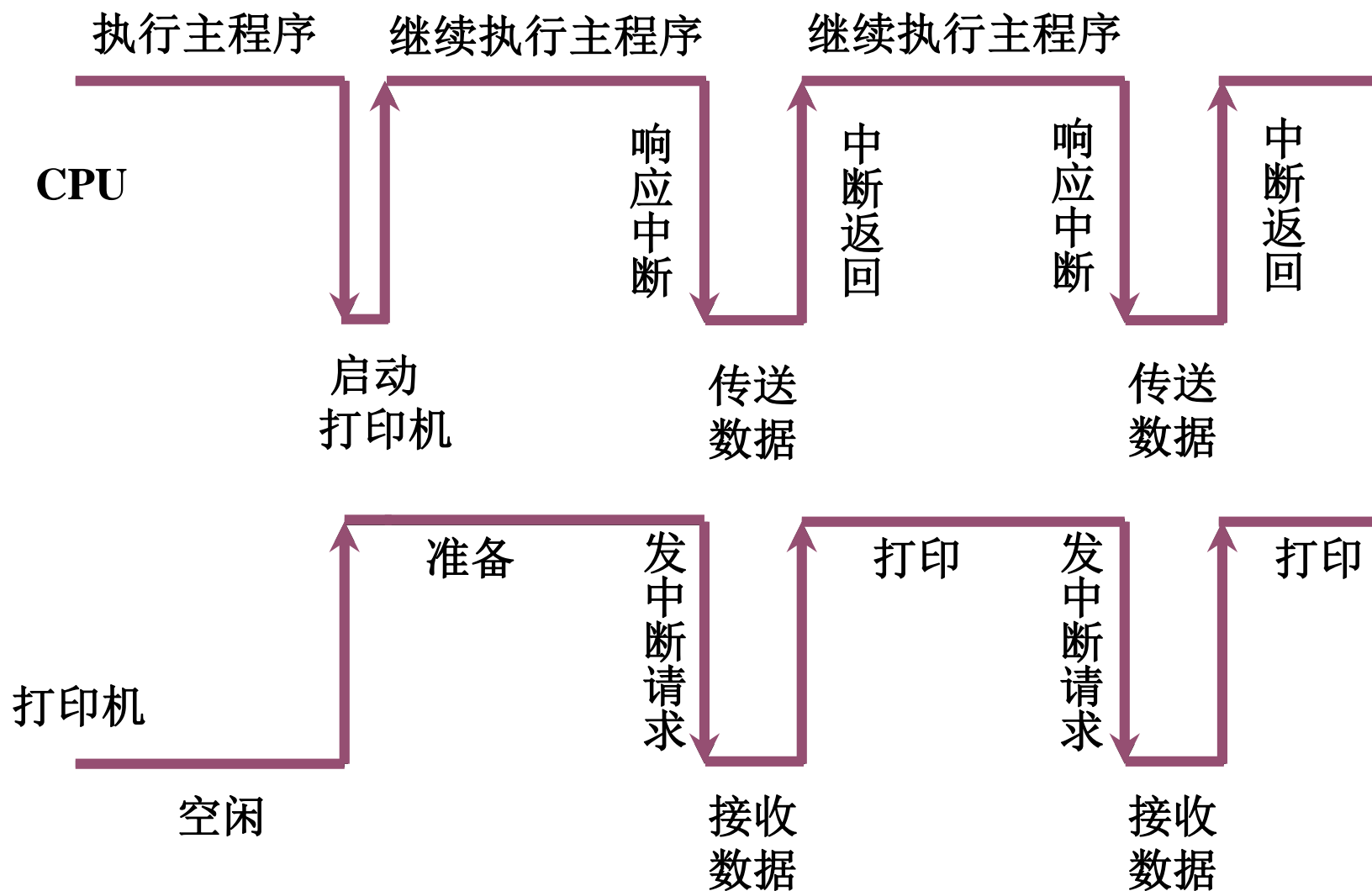
8.5 程序中中断方式

一、中断的概念



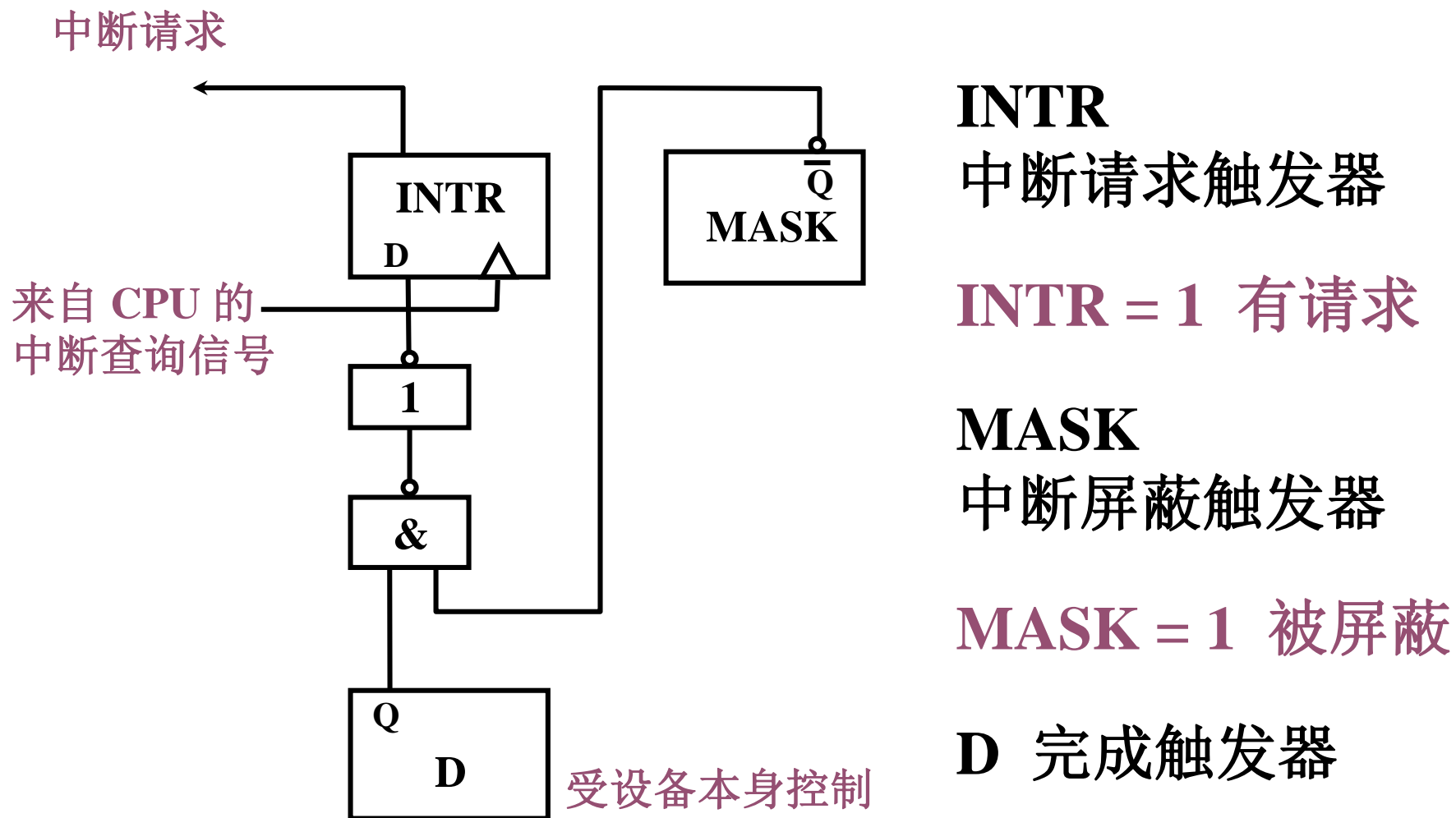
二、I/O 中断的产生

以打印机为例 CPU 与打印机并行工作



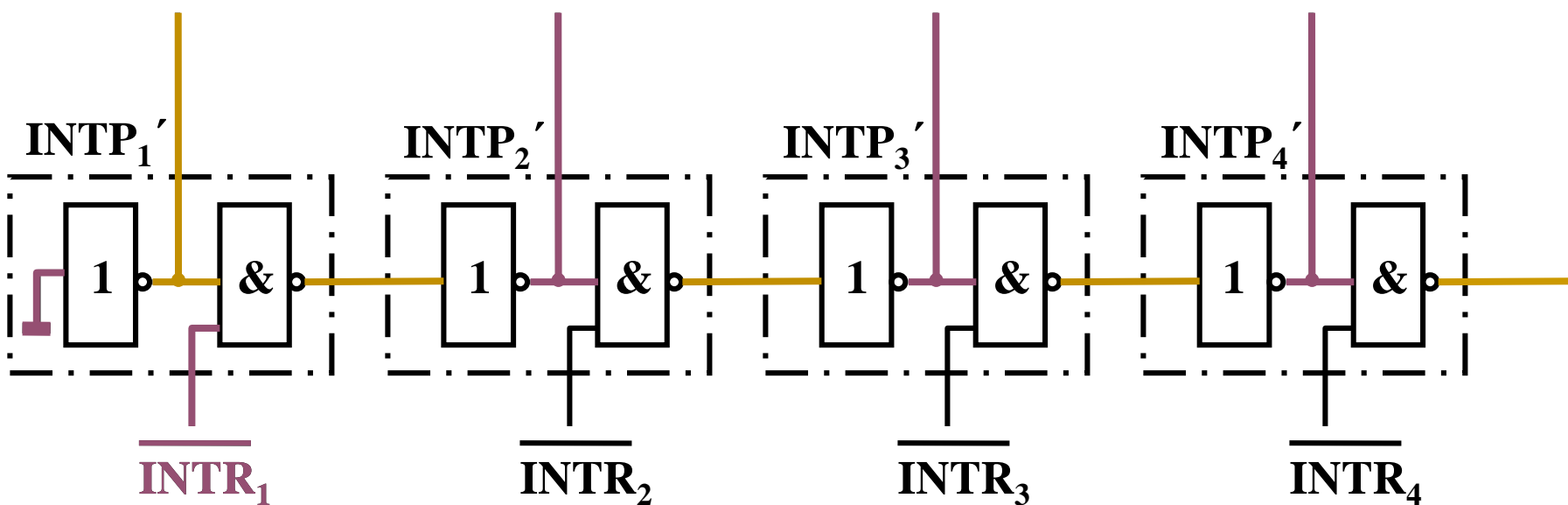
三、程序中断方式的接口电路

1. 配置中断请求触发器和中断屏蔽触发器



2. 排队器

排队 { 硬件 在 CPU 内或在接口电路中（链式排队器）
软件 详见第八章

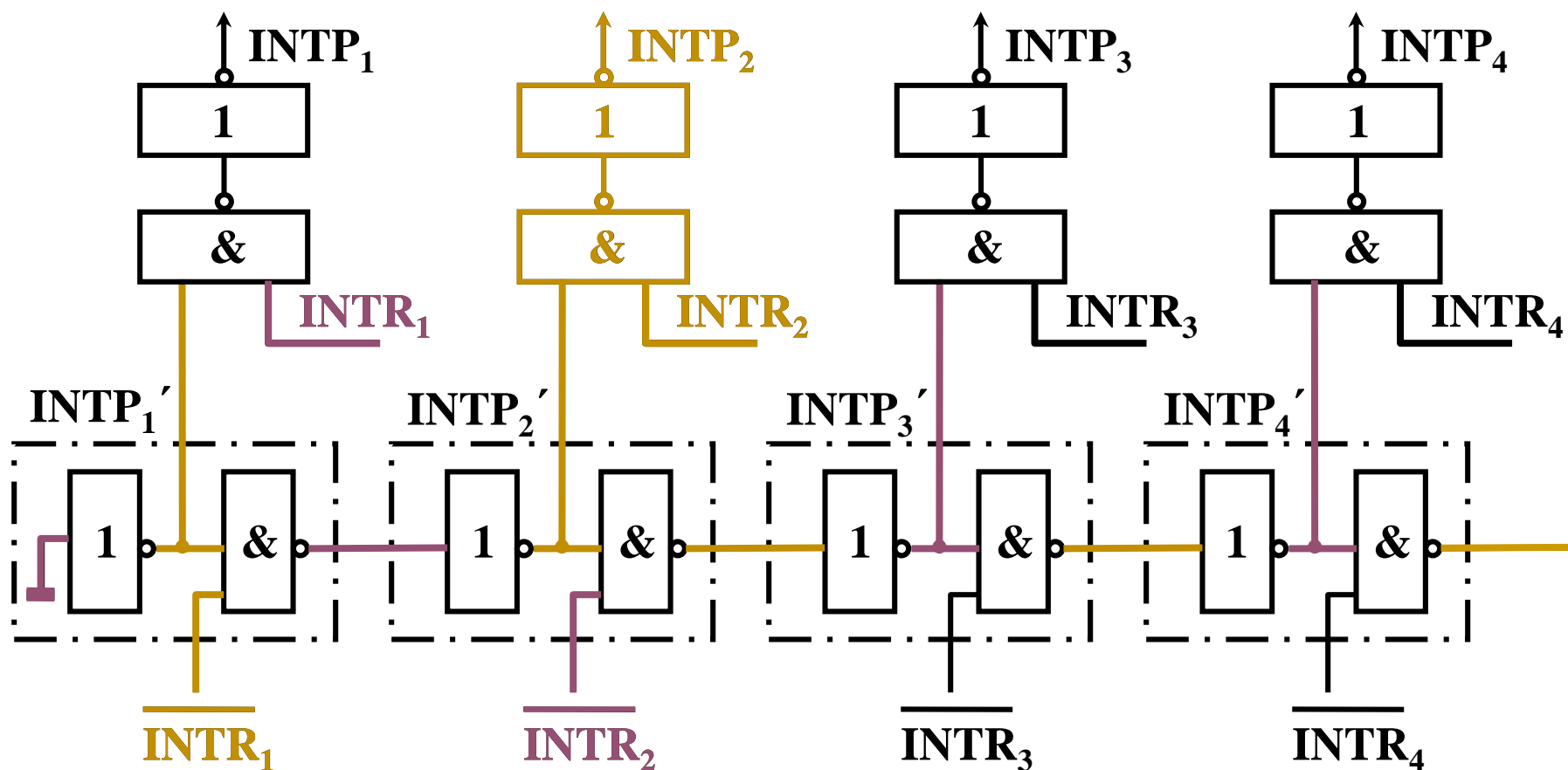


设备 1[#]、2[#]、3[#]、4[#] 优先级按 降序排列

$\text{INTR}_i = 1$ 有请求 即 $\overline{\text{INTR}}_i = 0$

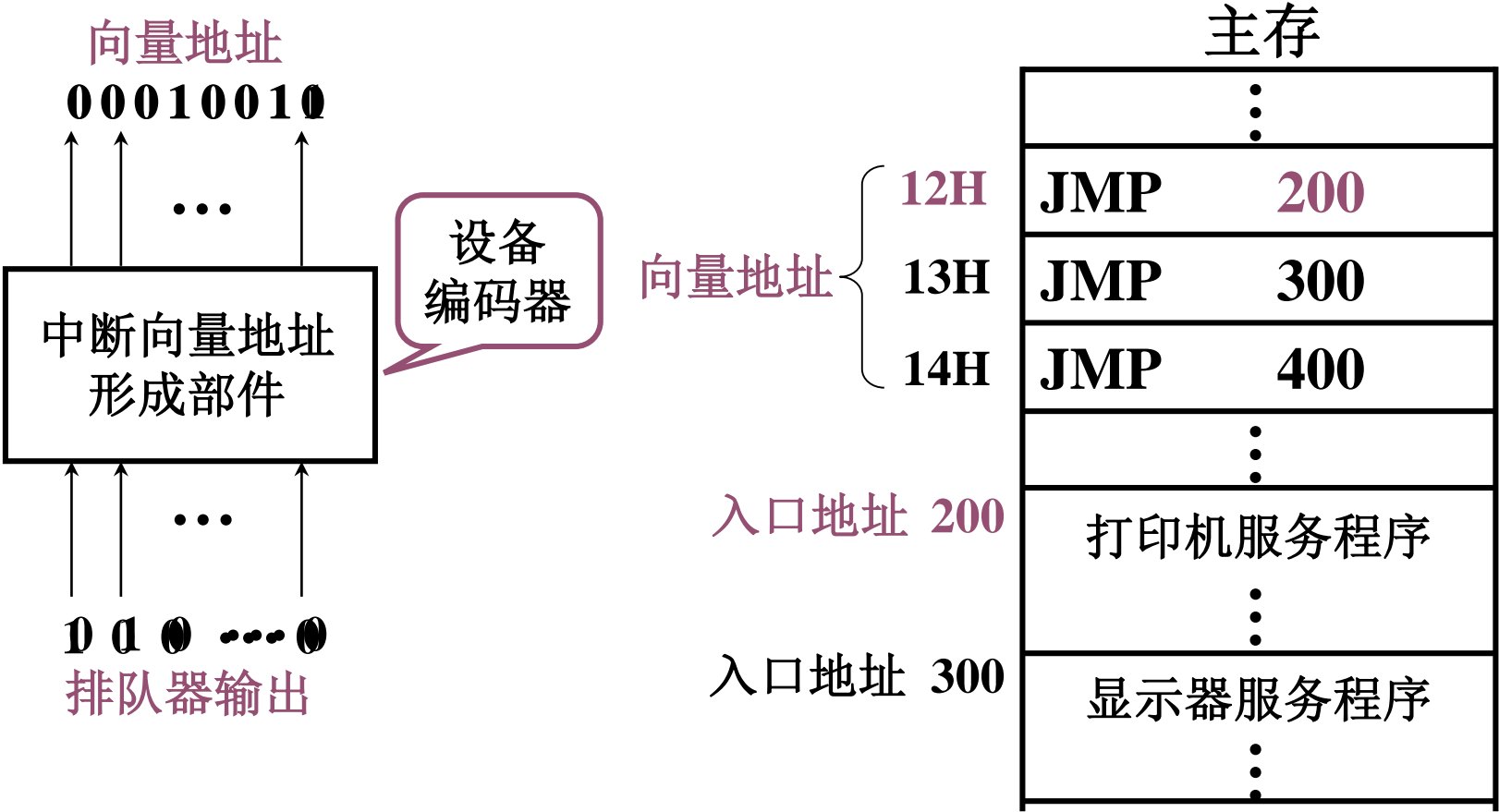
2. 排队器

排队 { 硬件 在 CPU 内或在接口电路中（链式排队器）
软件 详见第八章

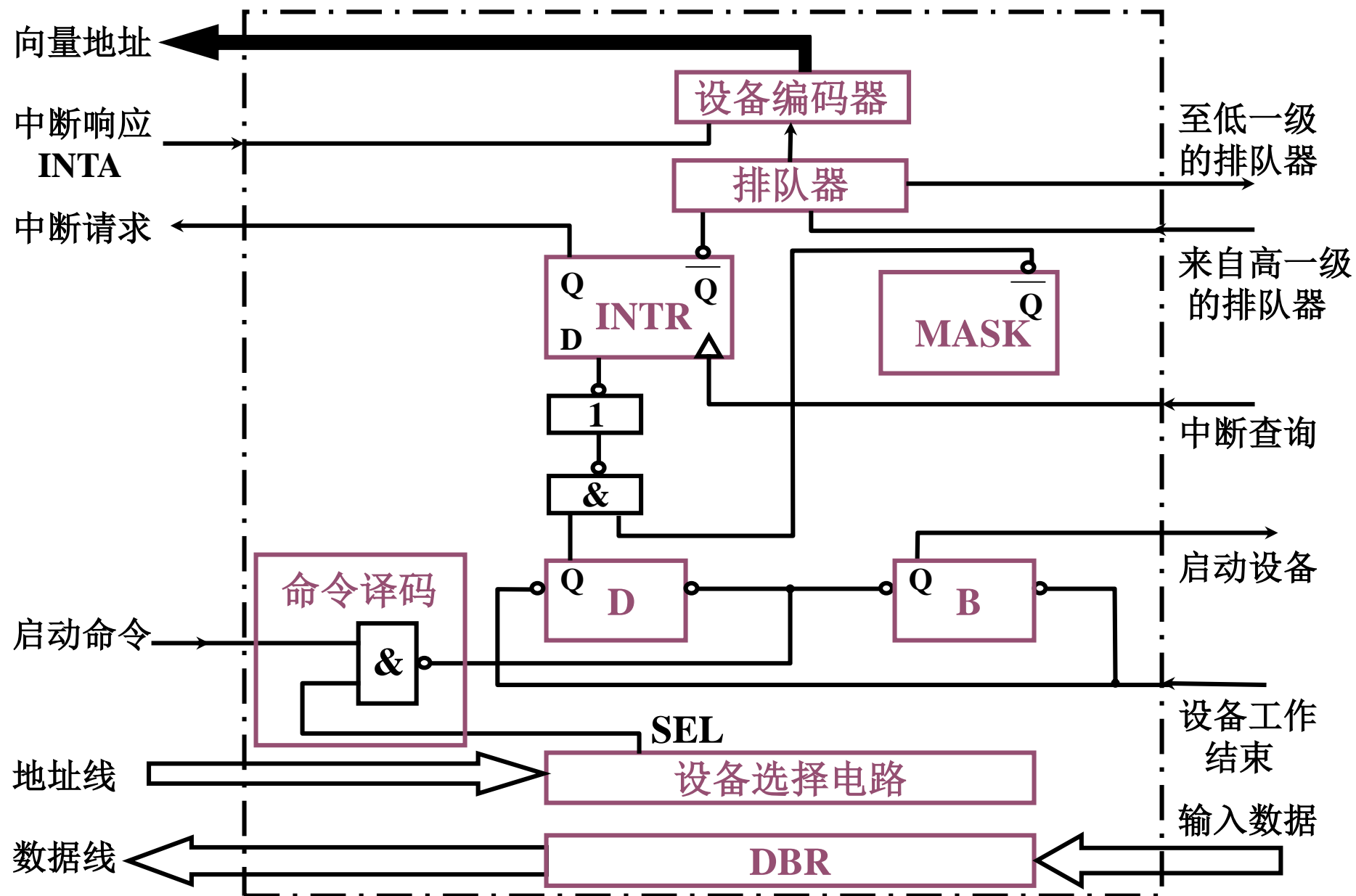


3. 中断向量地址形成部件

入口地址 { 由软件产生 详见第八章
 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



4. 程序中中断方式接口电路的基本组成



8.5 程序中中断方式

四、I/O 中断处理过程

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**” 或硬件 **自动复位**

(2) 时间

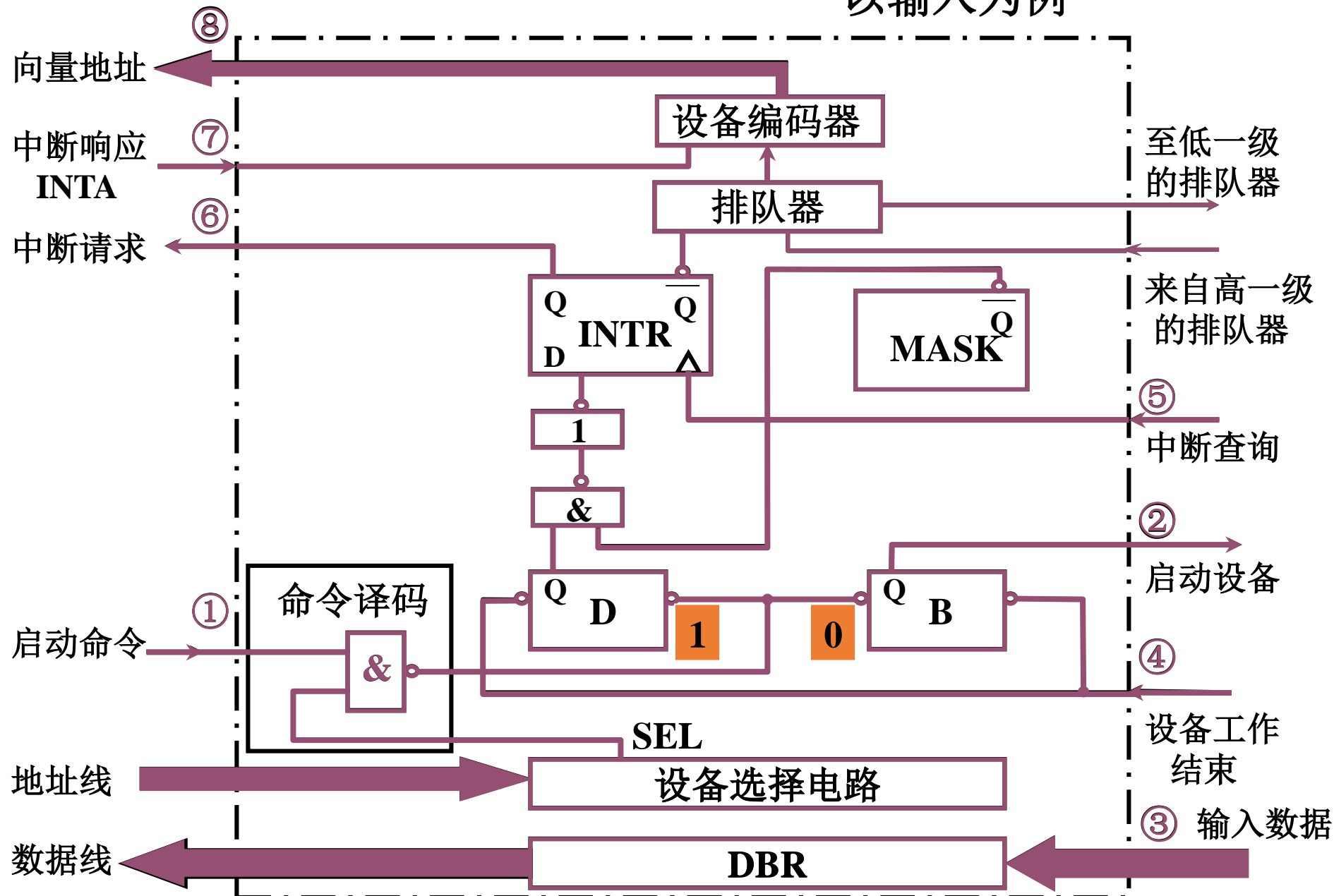
当 **D = 1**（随机）且 **MASK = 0** 时

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



五、中断服务程序流程

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

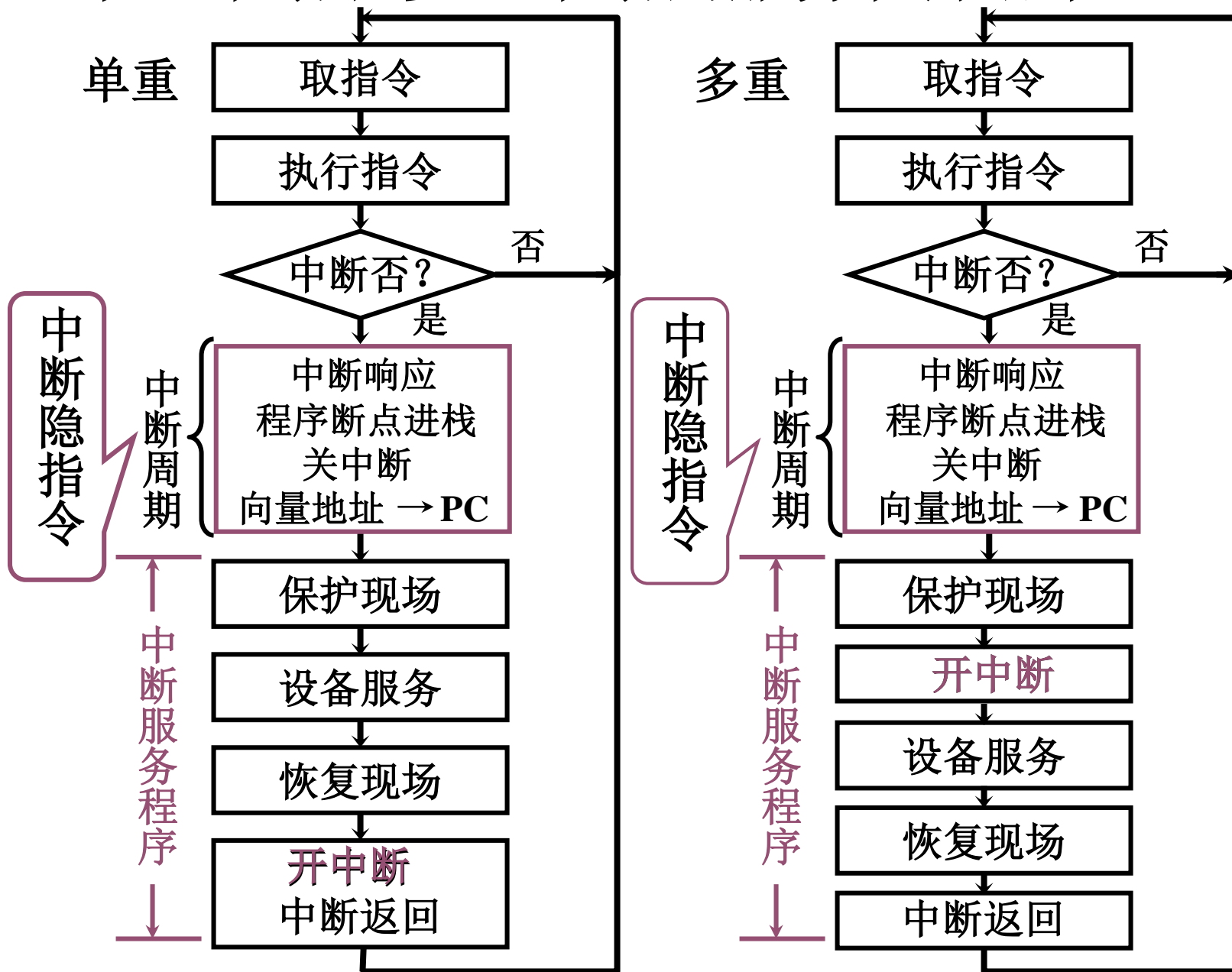
2. 单重中断和多重中断

单重 中断 不允许中断 现行的 中断服务程序

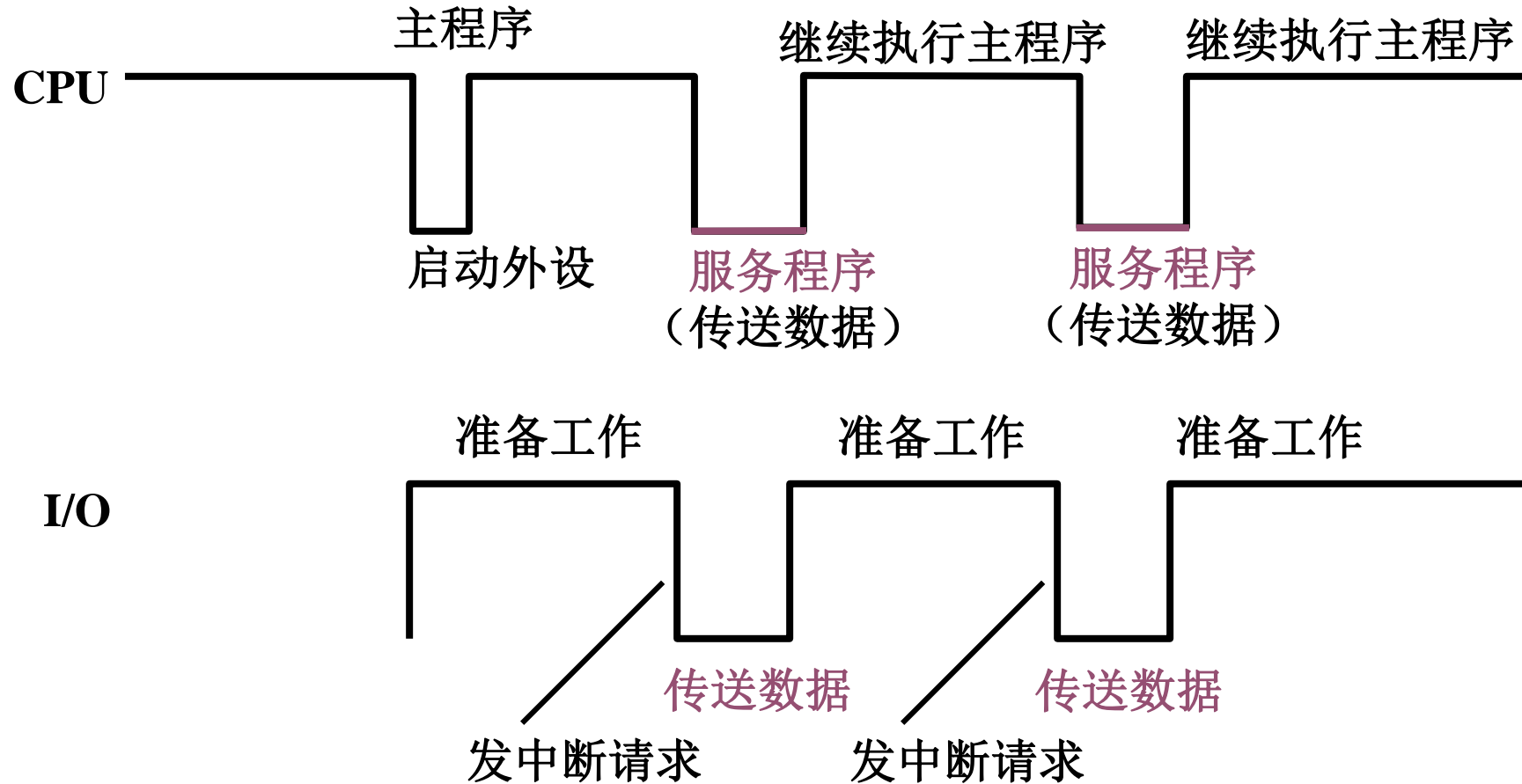
多重 中断 允许级别更高 的中断源

中断 现行的 中断服务程序

3. 单重中断和多重中断的服务程序流程



主程序和服务程序抢占 CPU 示意图



宏观上 CPU 和 I/O 并行工作
微观上 CPU 中断现行程序为 I/O 服务

第八章 输入输出系统

8.1 概述

8.2 外部设备

8.3 I/O接口

8.4 程序查询方式

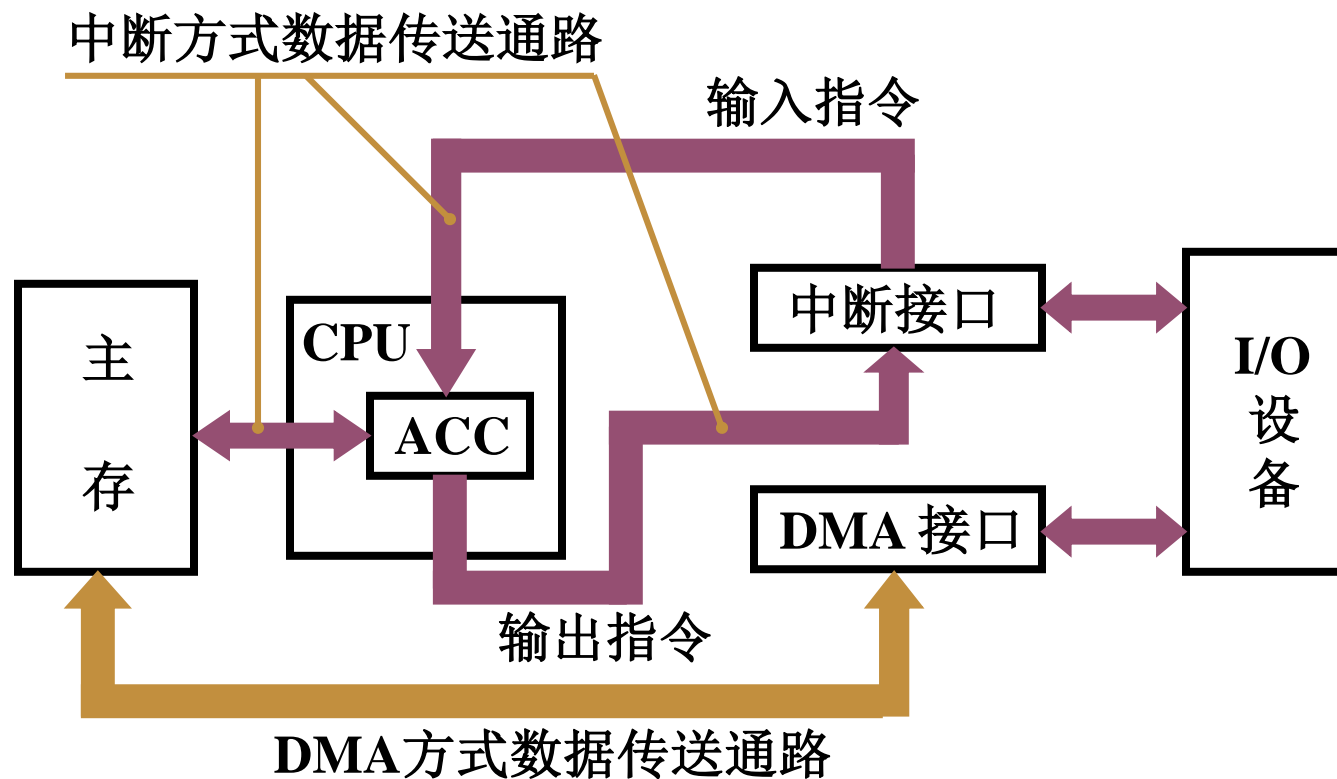
8.5 程序中断方式

8.6 DMA方式

8.6 DMA方式

一、DMA 方式的特点

1. DMA 和程序中断两种方式的数据通路



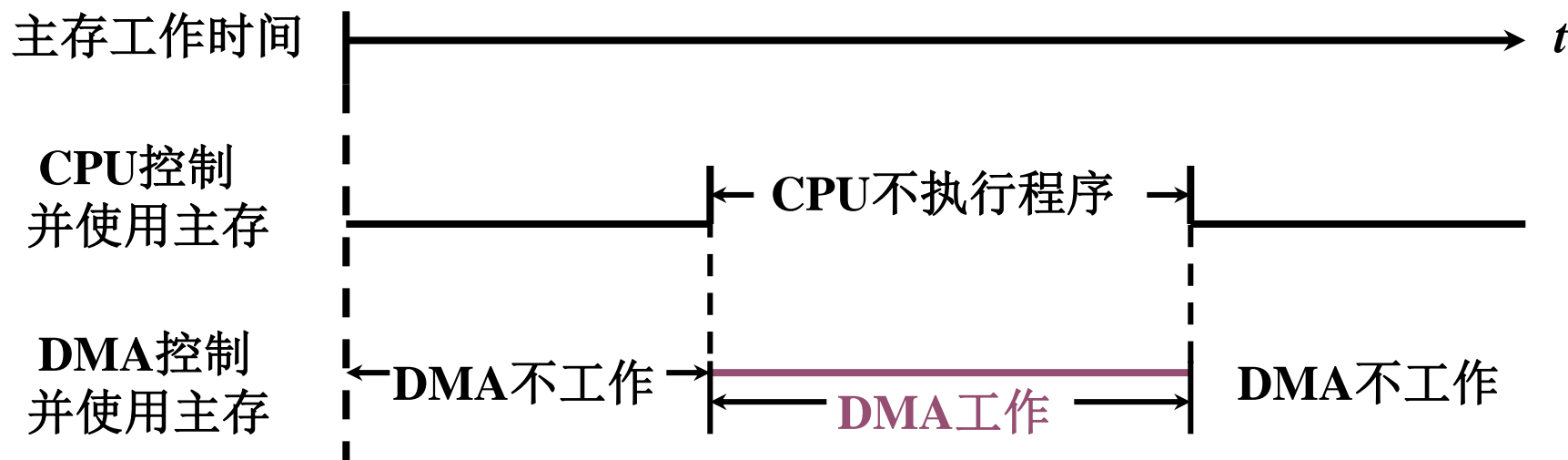
2. DMA 与主存交换数据的三种方式

(1) 停止 CPU 访问主存

控制简单

CPU 处于不工作状态或保持状态

未充分发挥 CPU 对主存的利用率

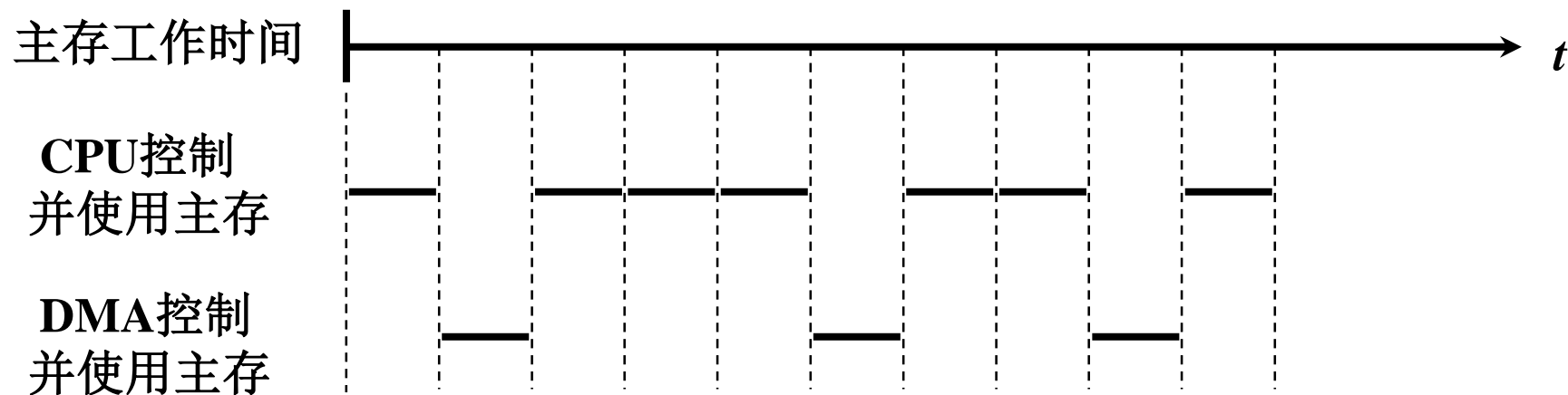


(2) 周期挪用 (或周期窃取)

DMA 访问主存有三种可能

- CPU 此时不访存
- CPU 正在访存
- CPU 与 DMA 同时请求访存

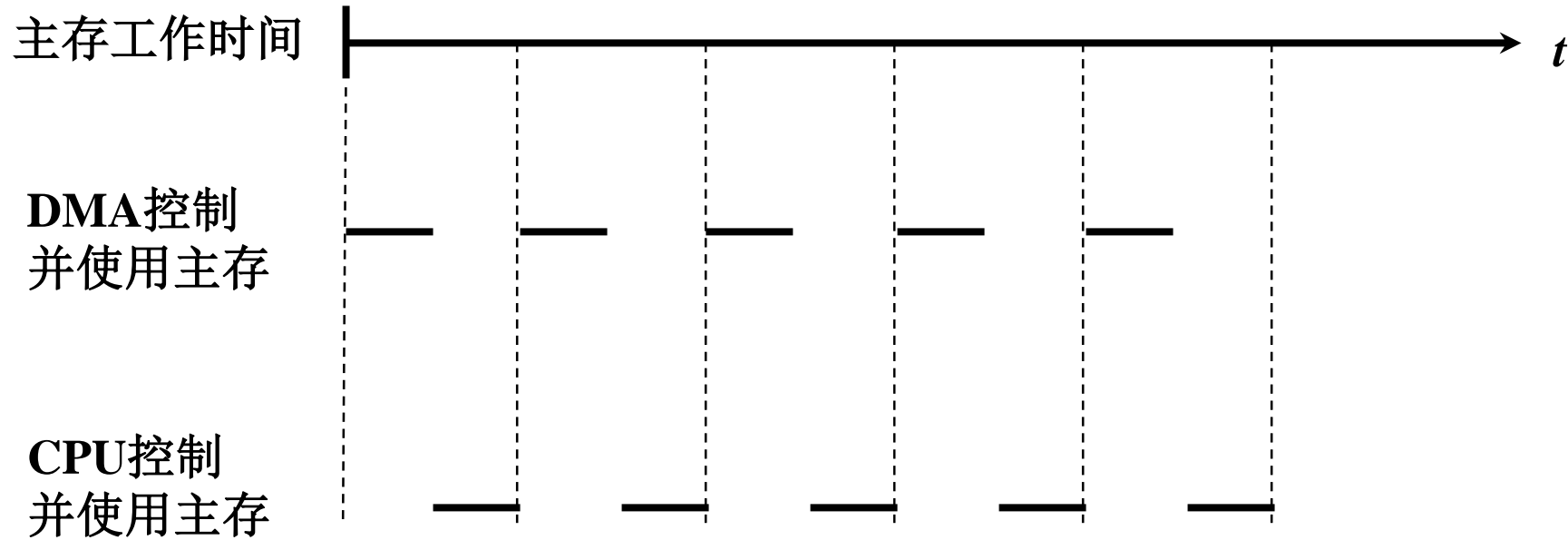
此时 CPU 将总线控制权让给 DMA



(3) DMA 与 CPU 交替访问

CPU 工作周期 $\begin{cases} C_1 \text{ 专供 DMA 访存} \\ C_2 \text{ 专供 CPU 访存} \end{cases}$

所有指令执行过程中的一个基准时间



不需要 申请建立和归还 总线的使用权

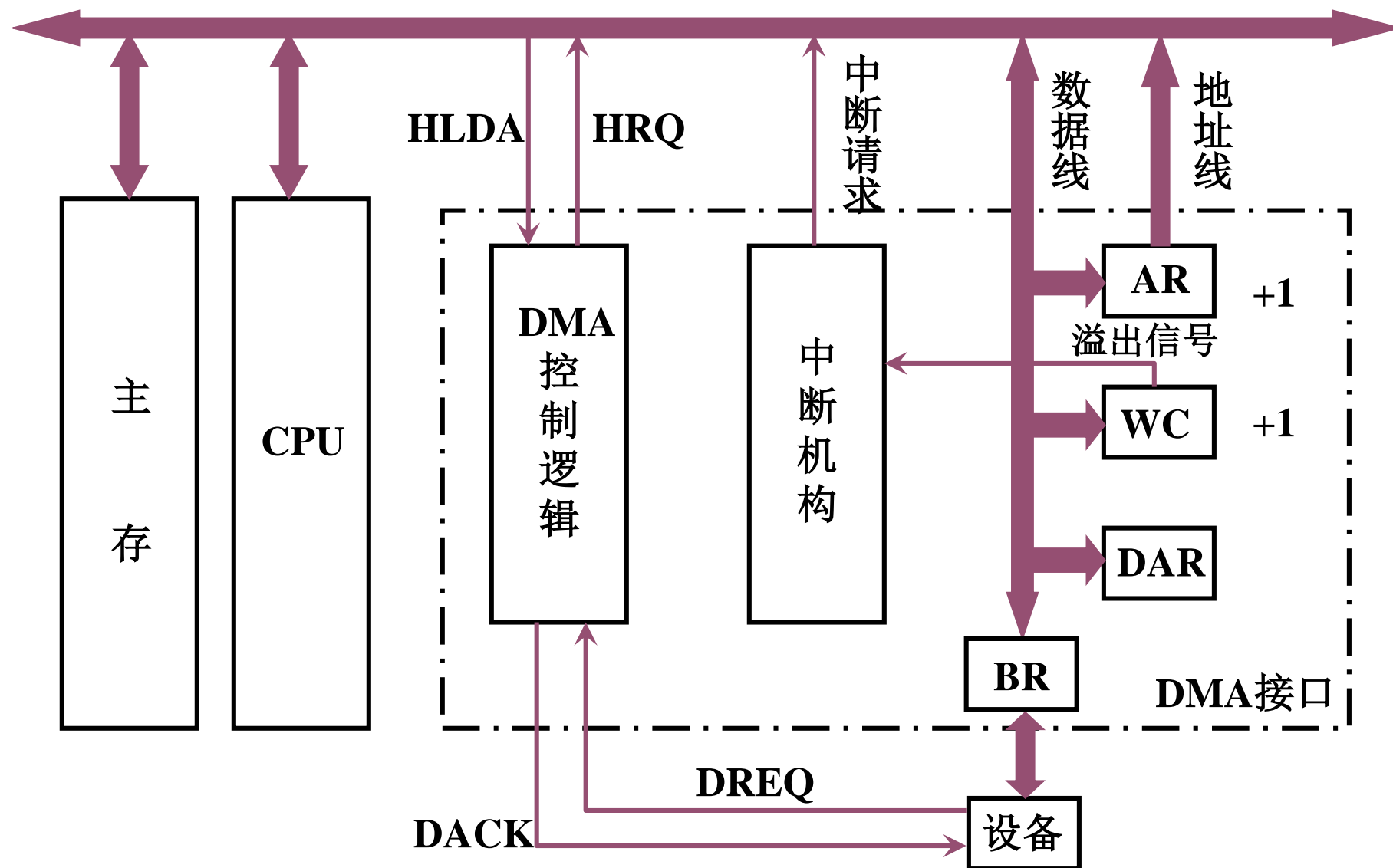
8.6 DMA方式

二、DMA 接口的功能和组成

1. DMA 接口功能

- (1) 向 CPU 申请 DMA 传送
- (2) 处理总线 控制权的转交
- (3) 管理 系统总线、控制 数据传送
- (4) 确定 数据传送的 首地址和长度
修正 传送过程中的数据 地址 和 长度
- (5) DMA 传送结束时， 给出操作完成信号

2. DMA 接口组成



8.6 DMA方式

三、DMA 的工作过程

1. DMA 传送过程

预处理、数据传送、后处理

(1) 预处理

通过几条输入输出指令预置如下信息

- 通知 DMA 控制逻辑传送方向（入/出）
- 设备地址 → DMA 的 DAR
- 主存地址 → DMA 的 AR
- 传送字数 → DMA 的 WC

(2) DMA 传送过程示意

CPU

预处理:

主存起始地址 → DMA
设备地址 → DMA
传送数据个数 → DMA
启动设备

数据传送:

继续执行主程序
同时完成一批数据传送

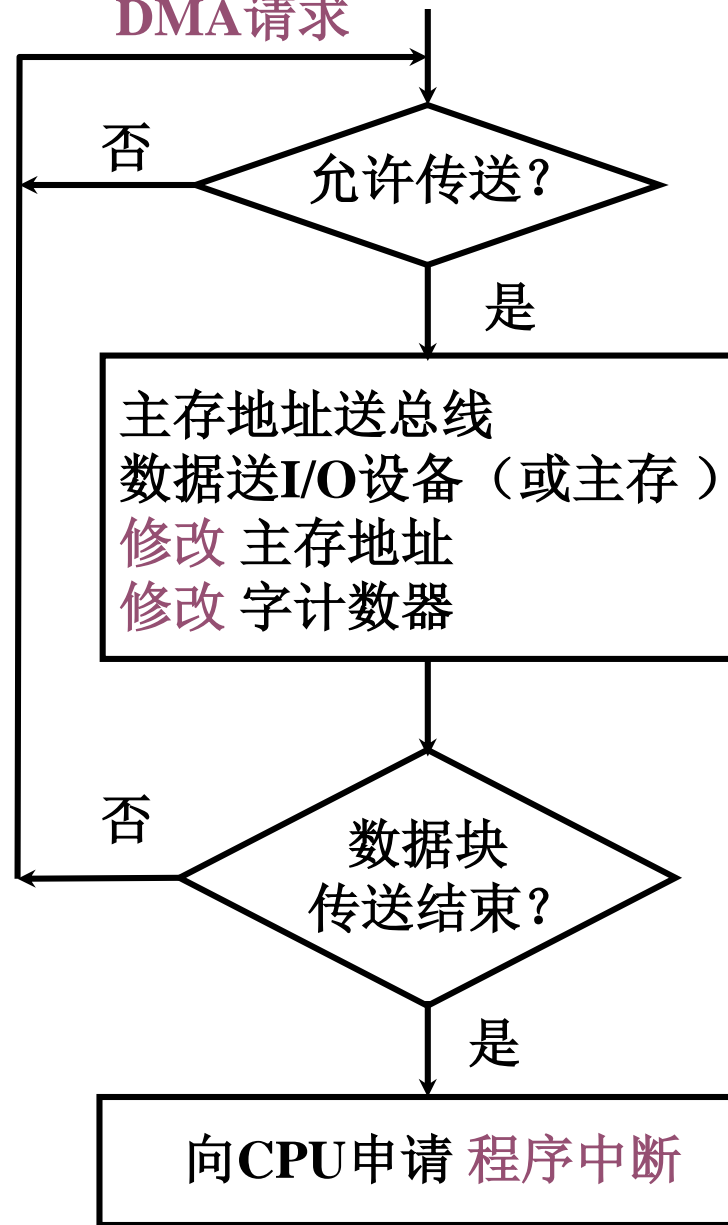
后处理:

中断服务程序
做 DMA 结束处理

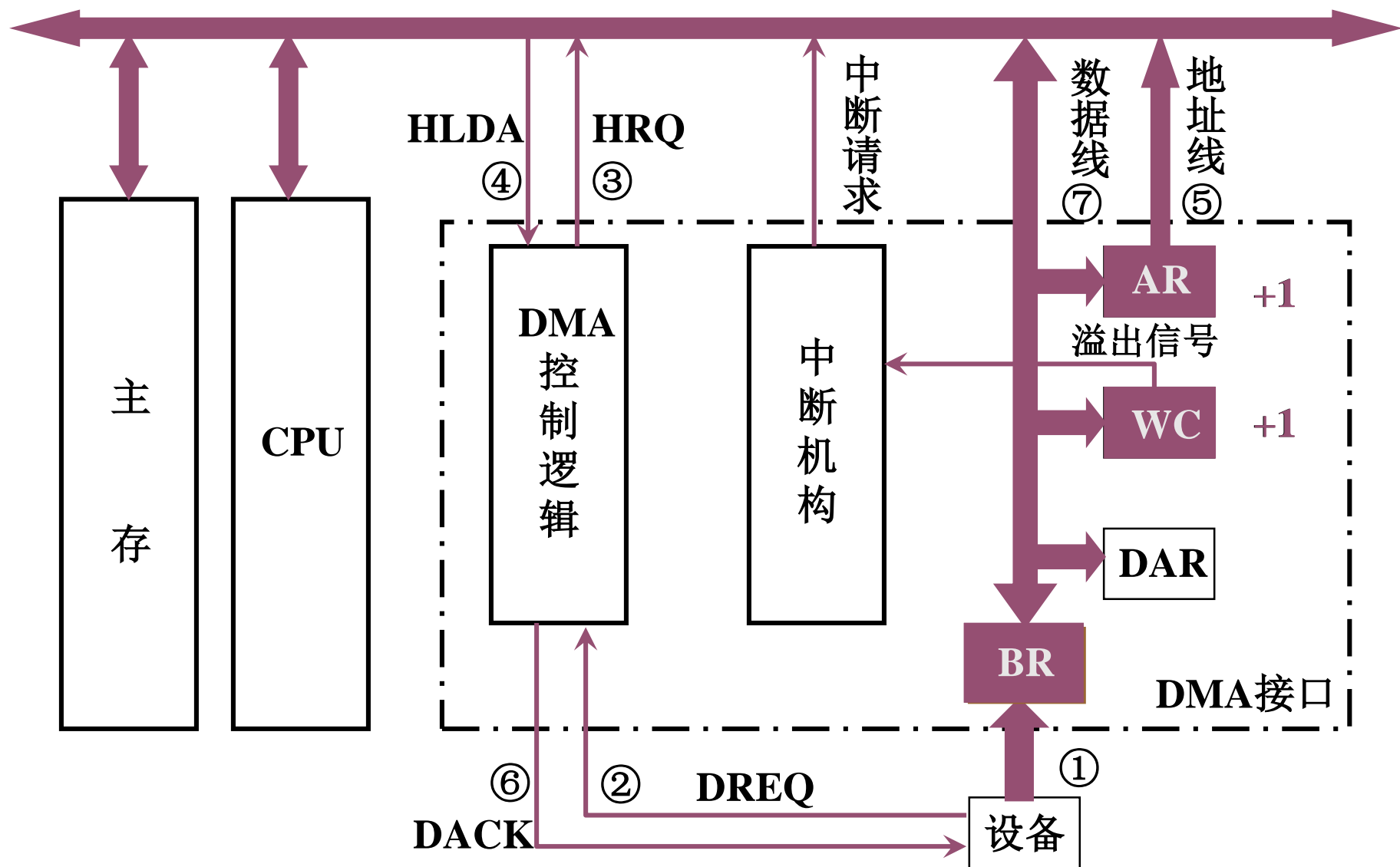
继续执行主程序

数据传送

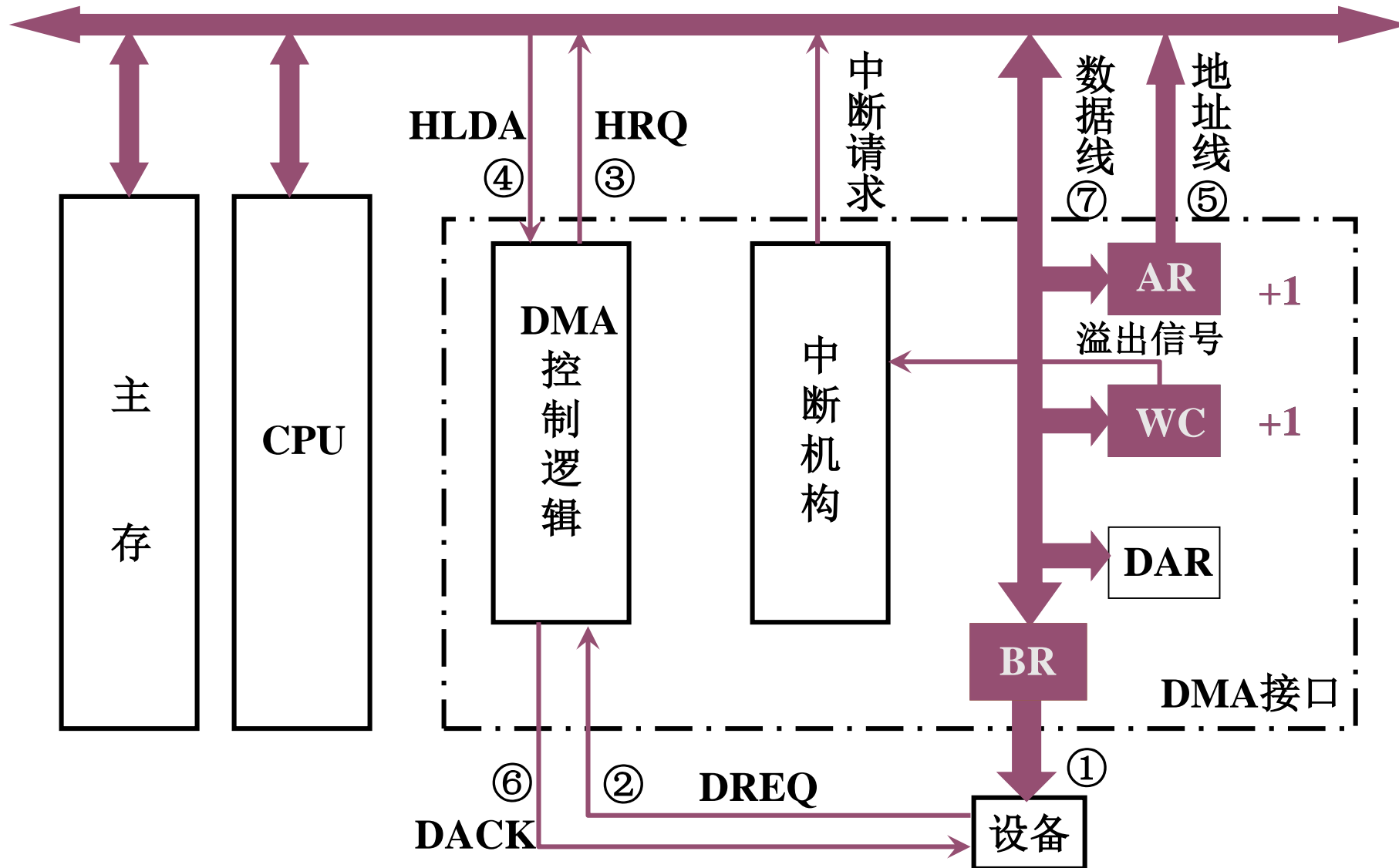
DMA 请求



(3) 数据传送过程（输入）



(4) 数据传送过程（输出）



(5) 后处理

校验送入主存的数是否正确

是否继续用 **DMA**

测试传送过程是否正确，错则转诊断程序

由中断服务程序完成

例8.2

设字符设备的传输率为 **9600 bps**

采用周期窃取方式把字符传送到存储器

其最大批量为 **400** 字节（忽略预处理时间）

若存取周期为 **100 ns**，处理一次中断需 **5 μs**

试问(1) 用**DMA** 方式，每秒因数据传输需多少时间？

(2) 用中断方式，每秒因数据传输需多少时间？

解：根据字符设备的传输率 **9600 bps**，得

每秒能传输 $9600/8 = \mathbf{1200\ B}$ ，即 **1200** 个字符

DMA $0.1\ \mu\text{s} \times 1200 + 5\ \mu\text{s} \times (1200 / 400) = \mathbf{135\ \mu\text{s}}$

中断 $5\ \mu\text{s} \times 1200 = \mathbf{6000\ \mu\text{s}}$

例8.3

采用 DMA 方式实现磁盘与主机交换信息

设磁盘传输速率为 2 MBps，传输的数据长度为 4 KB

DMA的预处理需 1000 个时钟周期

DMA 的后处理需 500 个时钟周期

试问 50 MHz 的处理器需用多少时间比率进行DMA辅助操作

解：DMA 传送 4 KB 的数据长度需

$$(4\text{KB}) / (2\text{MBps}) \approx 0.002 \text{ s}$$

若磁盘不断进行传输，每秒 所需 DMA 辅助操作的时钟周期数

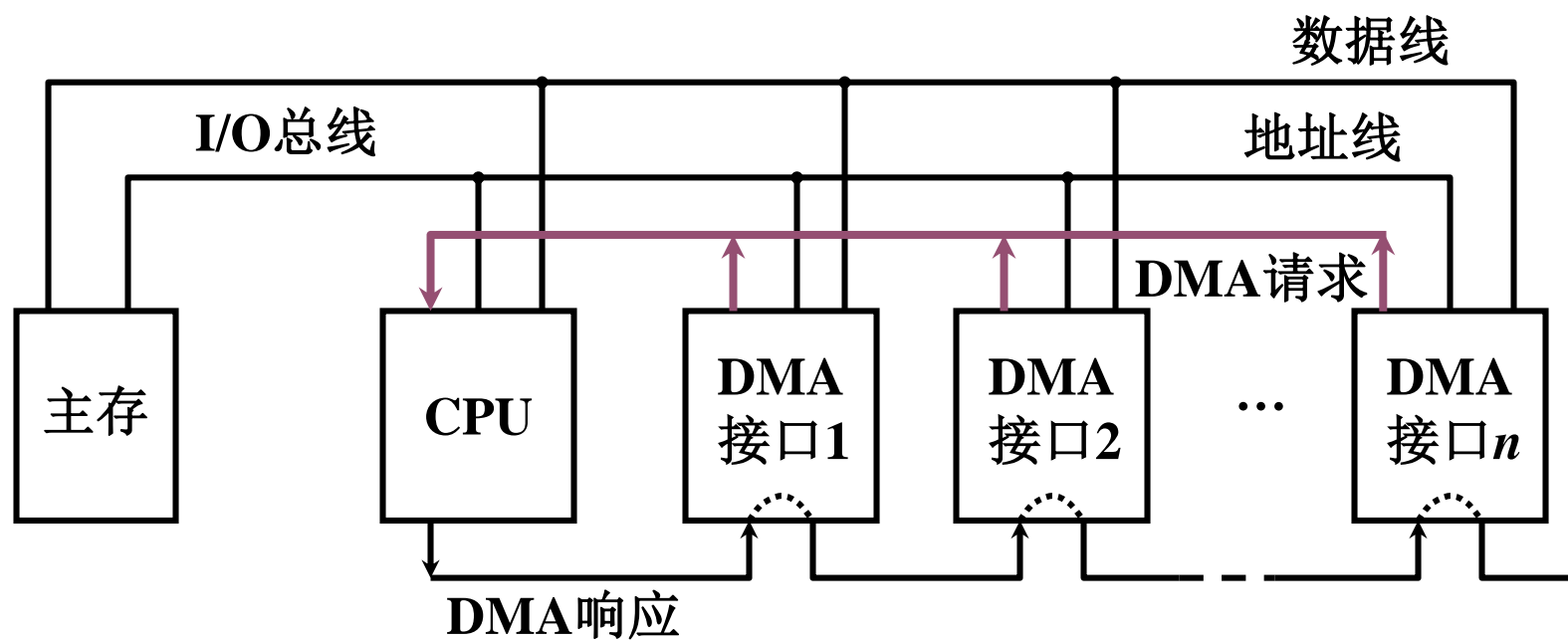
$$(1000 + 500) / 0.002 = 750000$$

故 DMA 辅助操作占用 CPU 的时间比率为

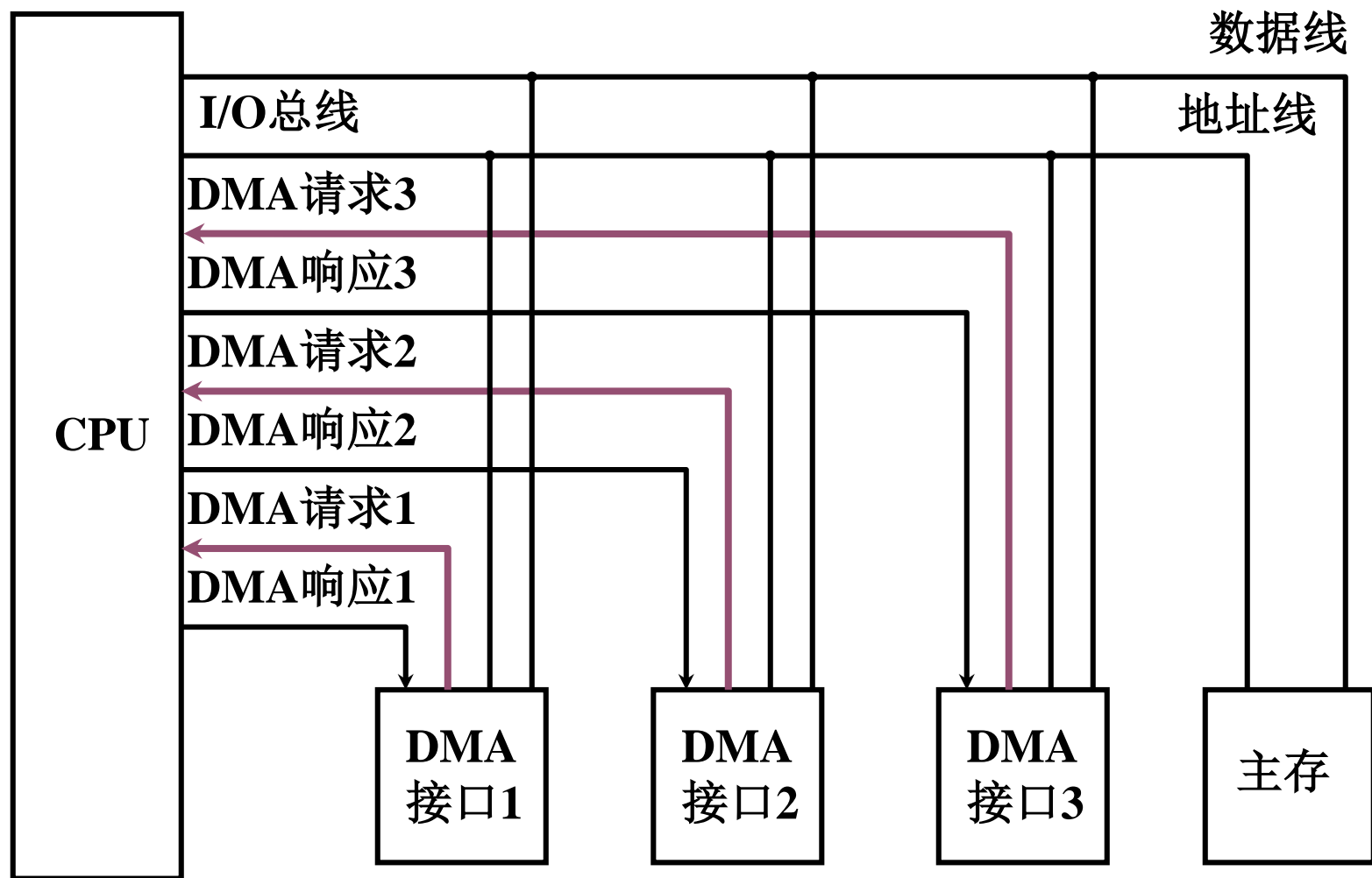
$$[750000 / (50 \times 10^6)] \times 100\% = 1.5 \%$$

2. DMA 接口与系统的连接方式

(1) 具有公共请求线的 DMA 请求



(2) 独立的 DMA 请求



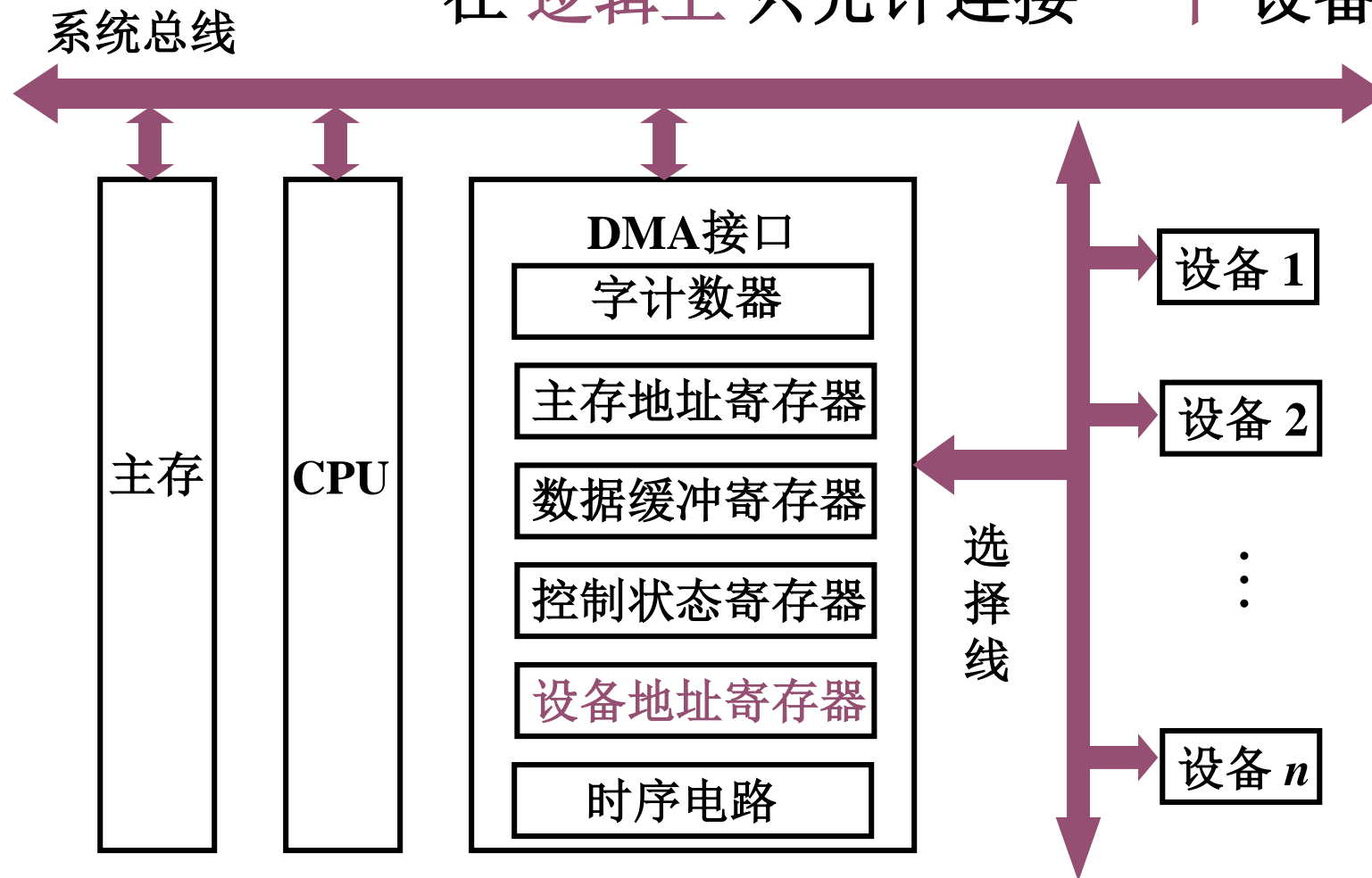
3. DMA 方式与程序中断方式的比较

	中断方式	DMA 方式
(1) 数据传送	程序	硬件
(2) 响应时间	指令执行结束	存取周期结束
(3) 处理异常情况	能	不能
(4) 中断请求	传送数据	后处理
(5) 优先级	低	高

四、DMA 接口的类型

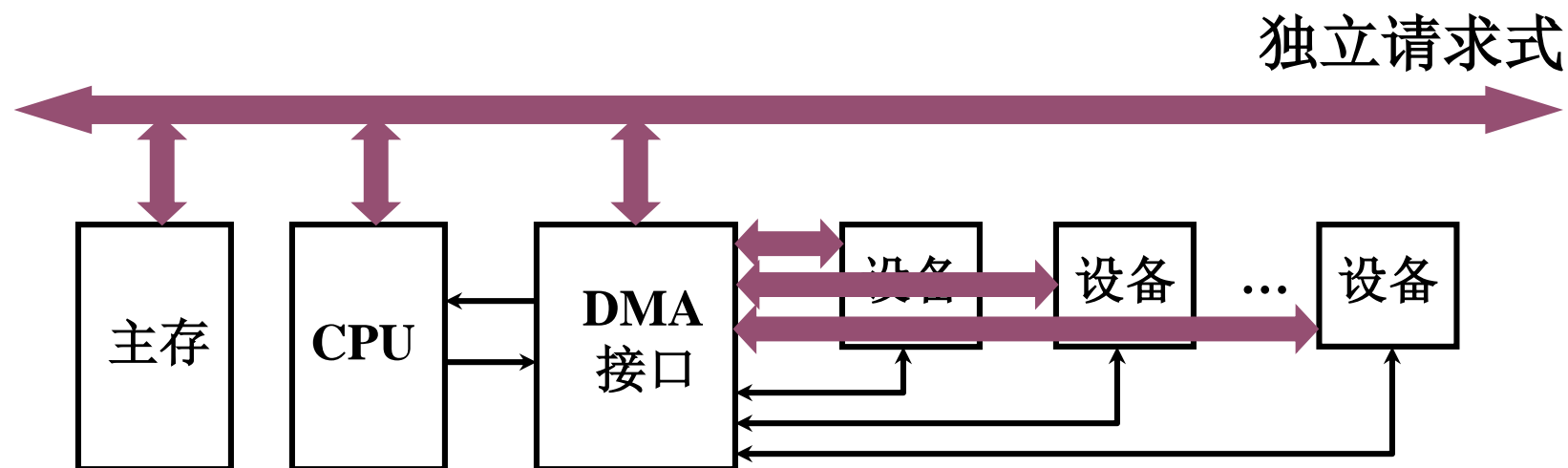
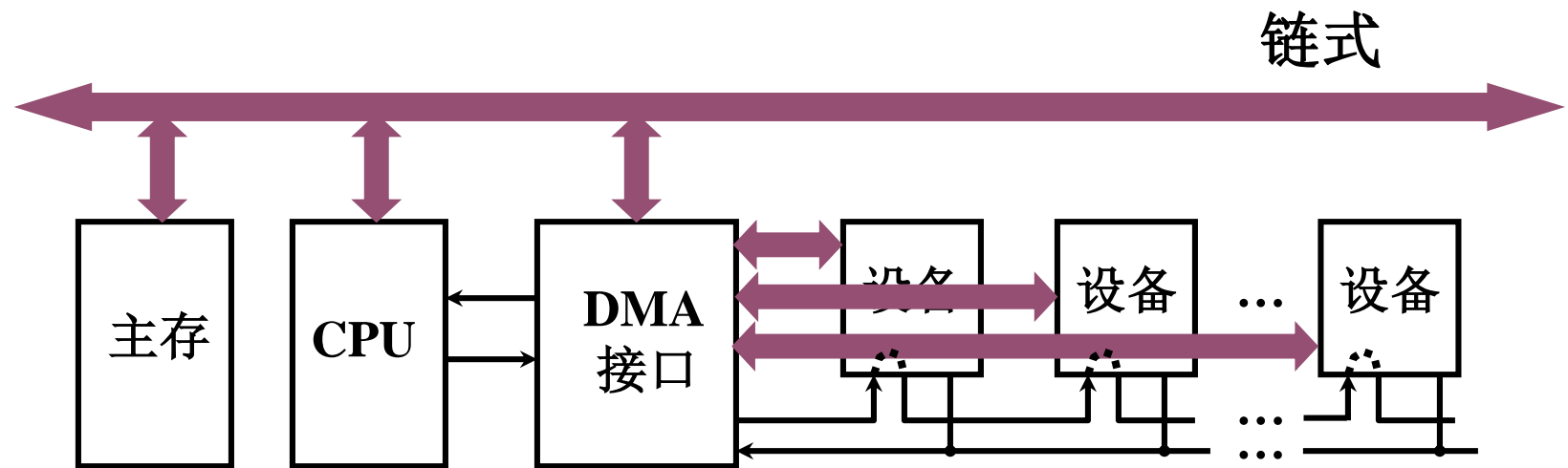
1. 选择型

在物理上连接多个设备
在逻辑上只允许连接一个设备

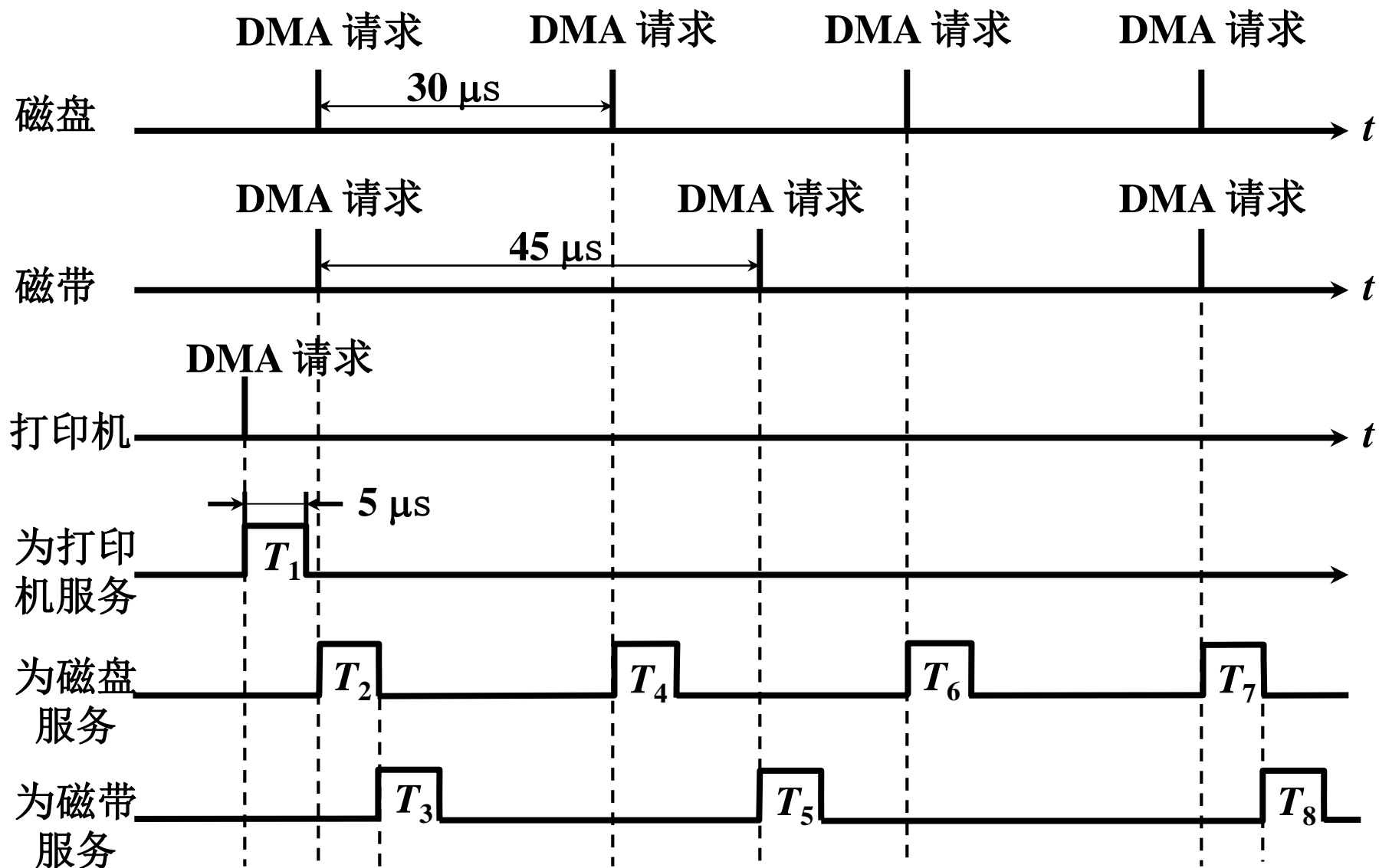


2. 多路型

在物理上连接多个设备
在逻辑上允许连接多个设备同时工作



3. 多路型 DMA 接口的工作原理



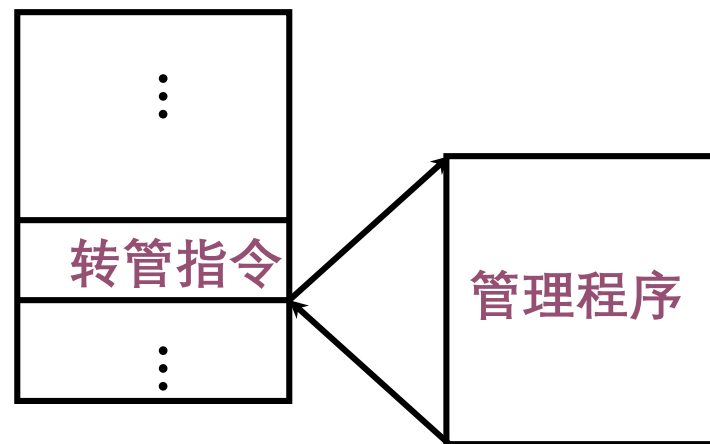
附加内容： 中断系统

一、概述

1. 引起中断的各种因素

(1) 人为设置的中断

如 转管指令



(2) 程序性事故 溢出、操作码不能识别、除法非法

(3) 硬件故障

(4) I/O 设备

(5) 外部事件 用 键盘中断 现行程序

2. 中断系统需解决的问题

- (1) 各中断源 如何 向 CPU 提出请求？
- (2) 各中断源 同时 提出 请求 怎么办？
- (3) CPU 什么 条件、什么 时间、以什么 方式响应中断？
- (4) 如何 保护现场？
- (5) 如何 寻找入口地址？
- (6) 如何 恢复现场，如何 返回？
- (7) 处理中断的过程中又 出现新的中断 怎么办？

硬件 + 软件

二、中断请求标记和中断判优逻辑

1. 中断请求标记 **INTR**

一个请求源 一个 **INTR** 中断请求标记触发器

多个**INTR** 组成 中断请求标记寄存器

1	2	3	4	5			<i>n</i>
掉电	过热	主存读写校验错	阶上溢	非法除法		键盘输入	打印机输出

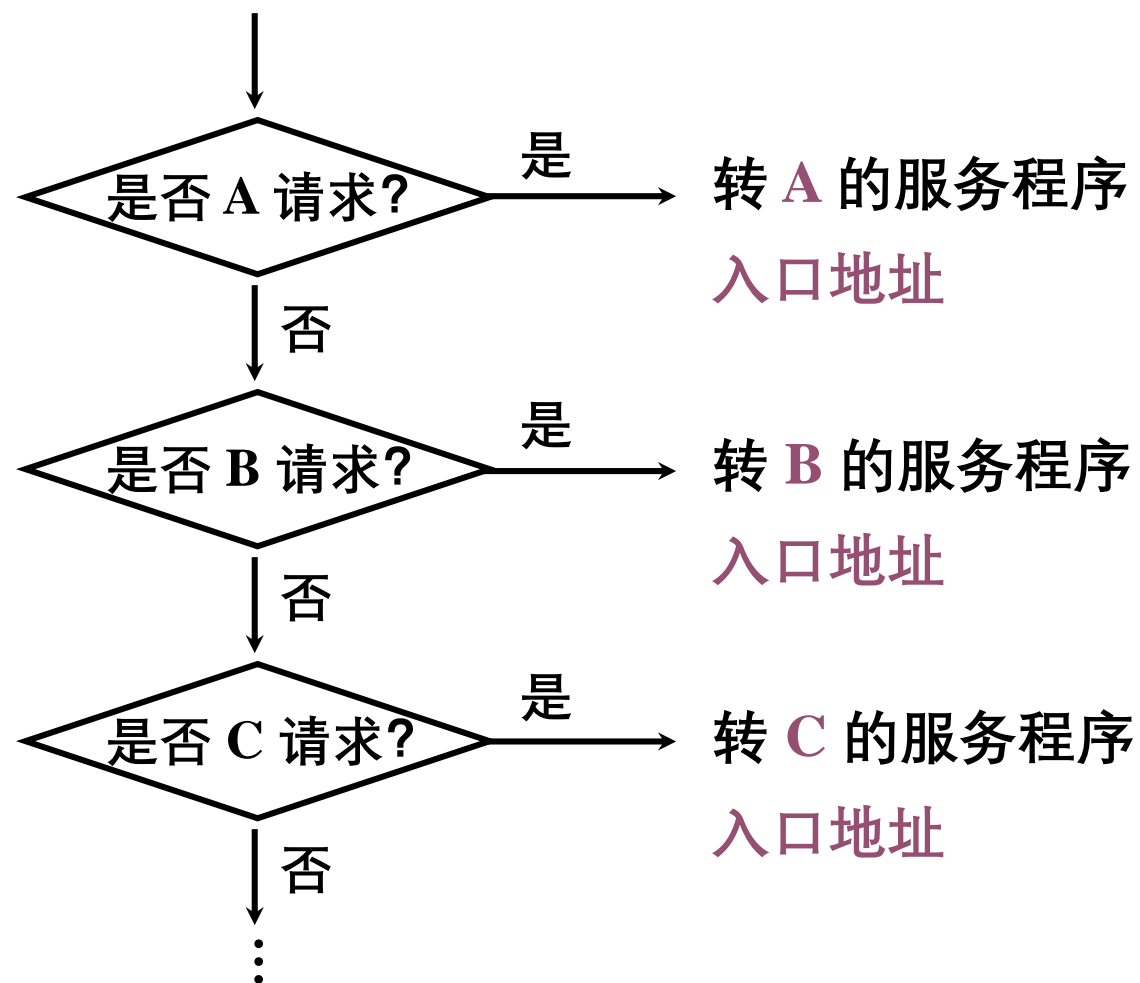
INTR 分散 在各个中断源的 接口电路中

INTR 集中 在 **CPU** 的中断系统 内

INTR₁、INTR₂、INTR₃、INTR₄ 优先级按降序排列

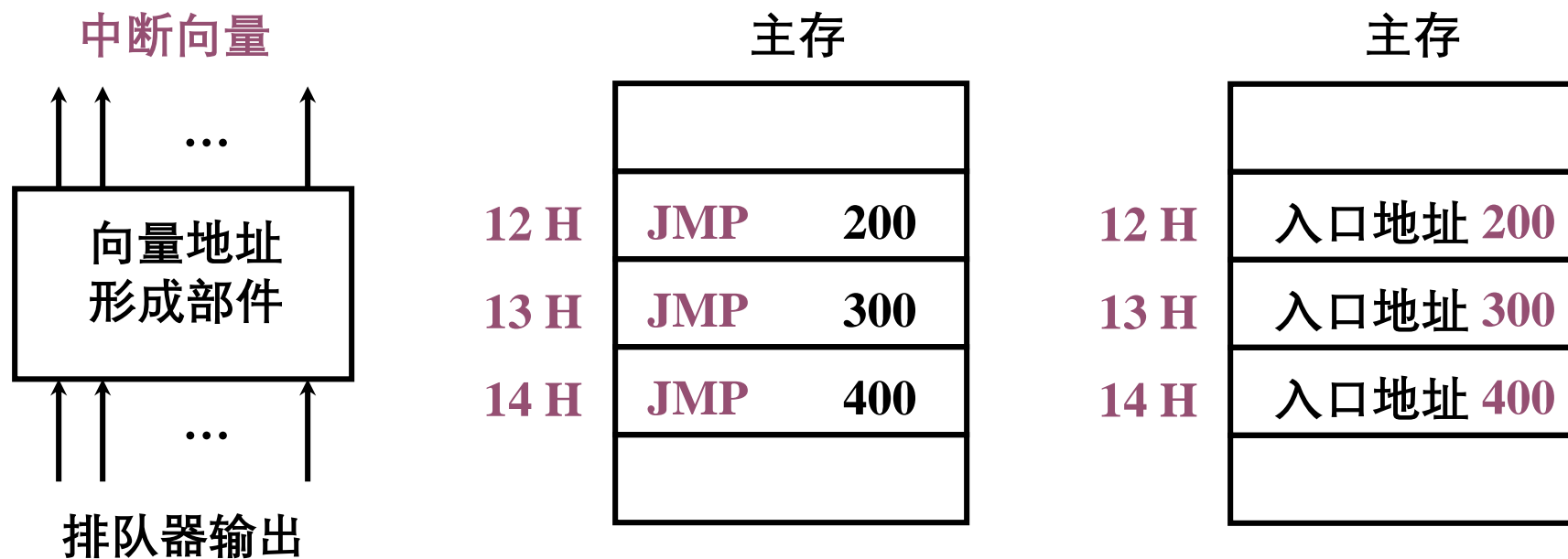
(2) 软件实现（程序查询）

A、B、C 优先级按 降序 排列



三、中断服务程序入口地址的寻找

1. 硬件向量法



向量地址 12H、13H、14H

入口地址 200、300、400

2. 软件查询法

八个中断源 1, 2, ... 8 按 **降序** 排列

中断识别程序 (入口地址 **M**)

地 址	指 令	说 明
M	SKP DZ 1 [#]	1 [#] D = 0 跳 (D为完成触发器)
	JMP 1 [#] SR1	1 [#] D = 1 转1 [#] 服务程序
	SKP DZ 2 [#]	2 [#] D = 0 跳
	JMP 2 [#] SR2	2 [#] D = 1 转2 [#] 服务程序
	⋮	
	SKP DZ 8 [#]	8 [#] D = 0 跳
	JMP 8 [#] SR	8 [#] D = 1 转8 [#] 服务程序

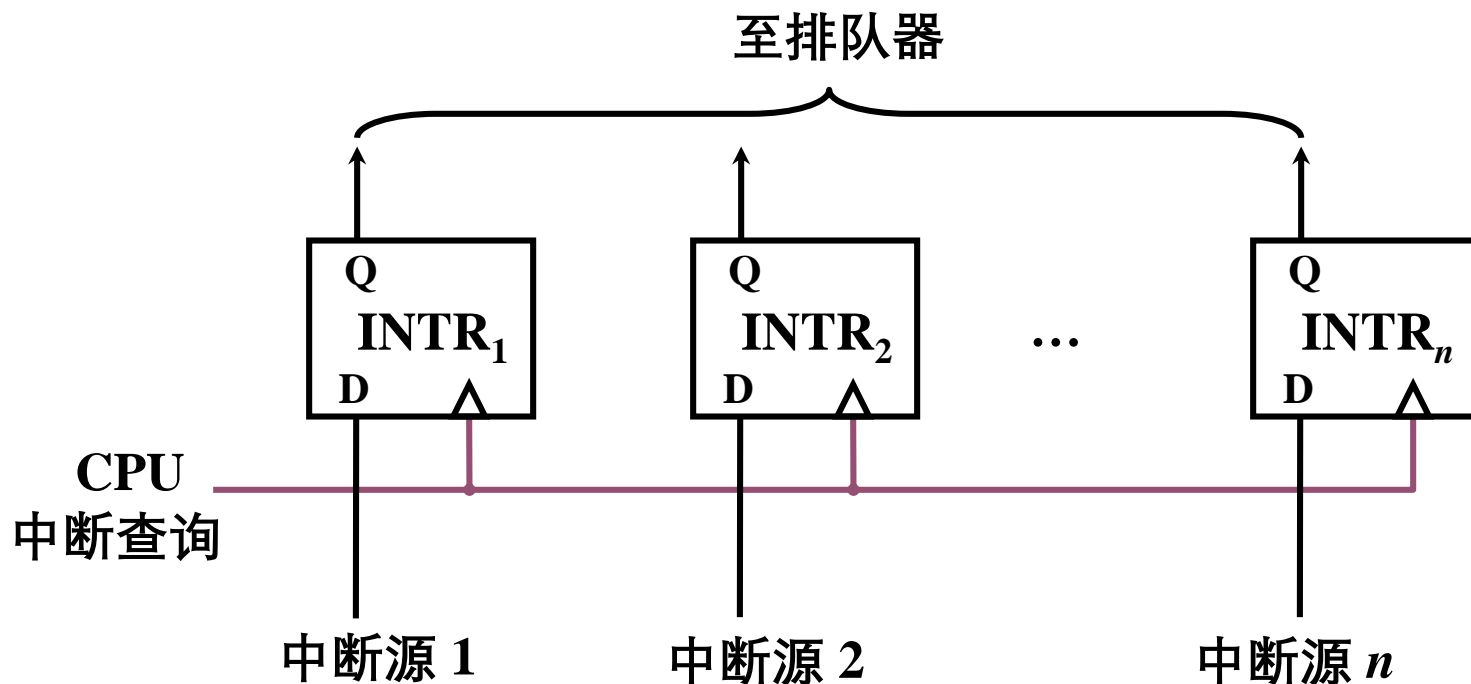
四、中断响应

1. 响应中断的条件

允许中断触发器 $EINT = 1$

2. 响应中断的时间

指令执行周期结束时刻由CPU发查询信号



3. 中断隐指令

(1) 保护程序断点

断点存于 特定地址（0 号地址）内 断点 进栈

(2) 寻找服务程序入口地址

向量地址 \longrightarrow PC (硬件向量法)

中断识别程序 入口地址 M \longrightarrow PC (软件查询法)

(3) 硬件 关中断

INT 中断标记

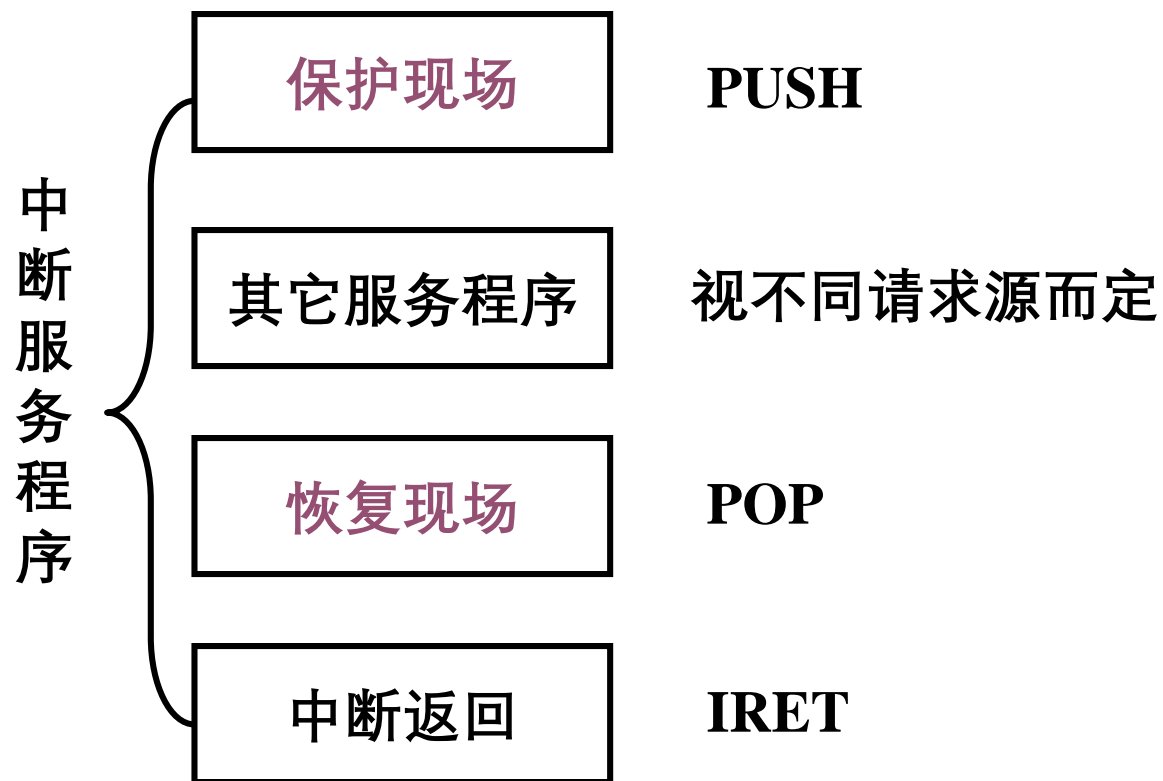
EINT 允许中断

R-S 触发器

五、保护现场和恢复现场

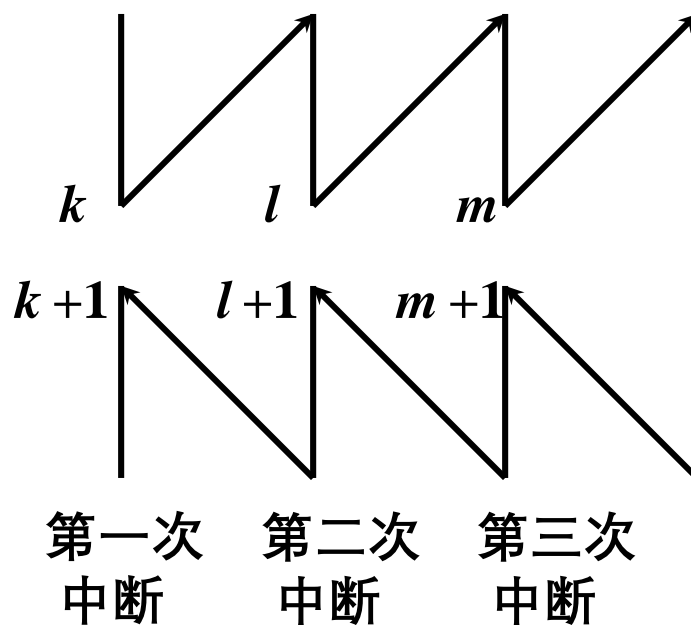
1. 保护现场 { 断点 中断隐指令 完成
 寄存器 内容 中断服务程序 完成

2. 恢复现场 中断服务程序 完成



六、中断屏蔽技术

1. 多重中断的概念

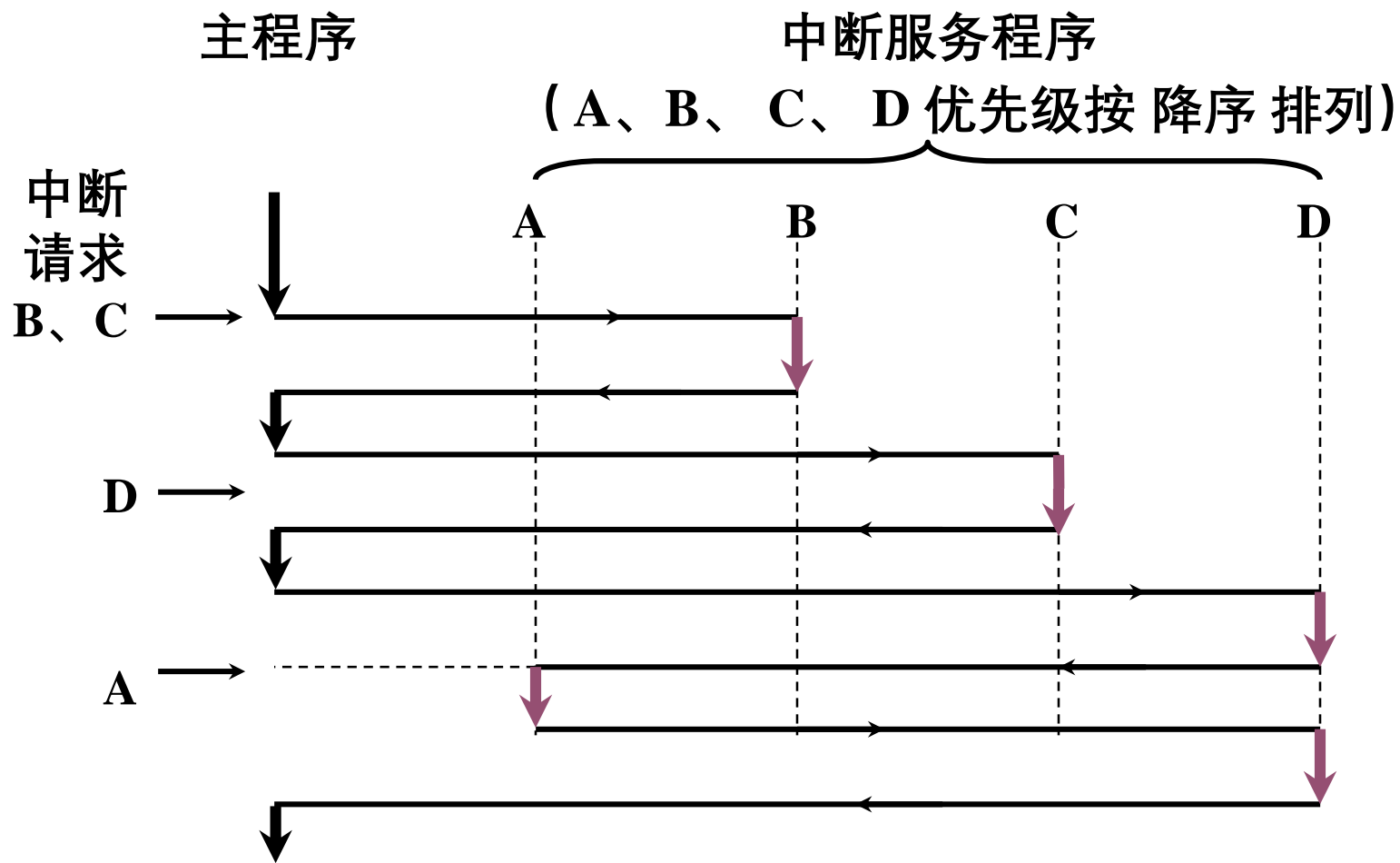


程序断点 $k+1, l+1, m+1$

2. 实现多重中断的条件

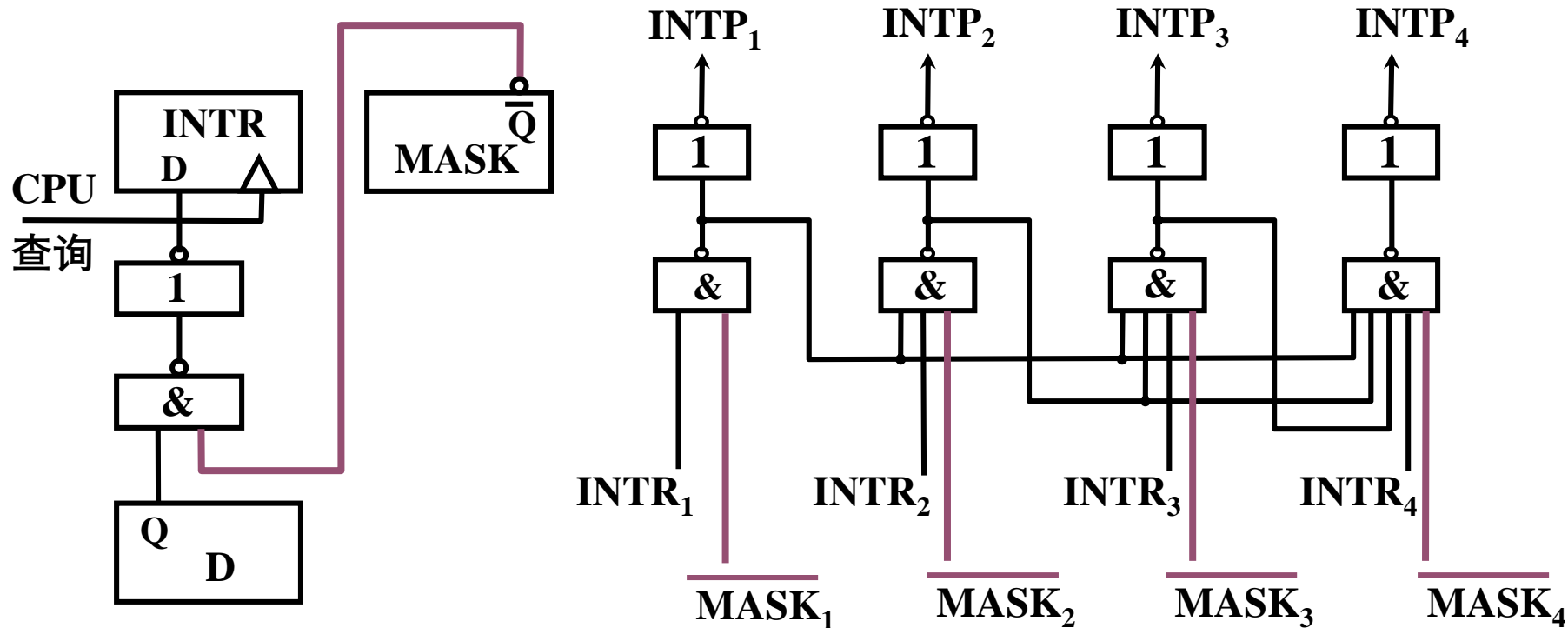
(1) 提前 设置 开中断 指令

(2) 优先级别高 的中断源 有权中断优先级别低 的中断源



3. 屏蔽技术

(1) 屏蔽触发器的作用



$MASK = 0$ (未屏蔽)

$INTR$ 能被置“1”

$MASK_i = 1$ (屏蔽)

$INTRP_i = 0$ (不能被排队选中)

(2) 屏蔽字

16个中断源 1, 2, 3, ... 16 按降序排列

优先级	屏	蔽	字
1	1	1	1
2	0	1	1
3	0	0	1
4	0	0	0
5	0	0	0
6	0	0	0
⋮			
15	0	0	0
16	0	0	0

(3) 屏蔽技术可改变处理优先等级

响应优先级 不可改变

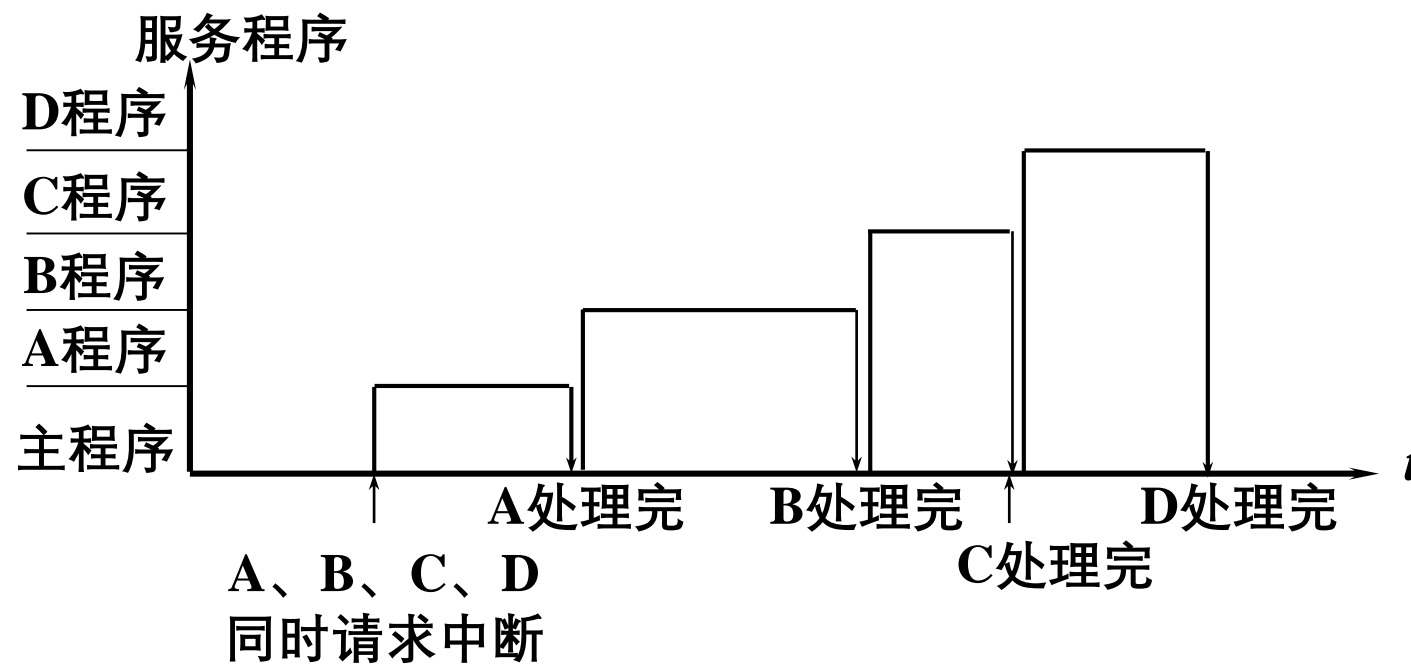
处理优先级 可改变（通过重新设置屏蔽字）

中断源	原屏蔽字	新屏蔽字
A	1 1 1 1	1 1 1 1
B	0 1 1 1	0 1 0 0
C	0 0 1 1	0 1 1 0
D	0 0 0 1	0 1 1 1

响应优先级 **A→B→C→D** 降序排列

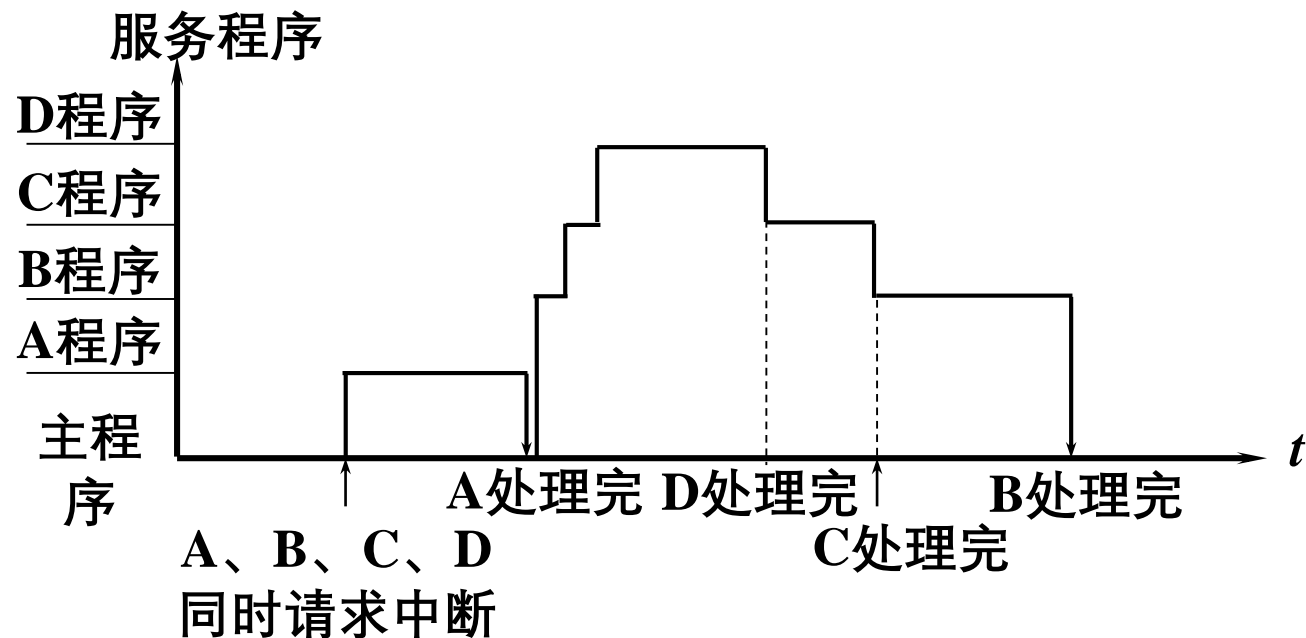
处理优先级 **A→D→C→B** 降序排列

(3) 屏蔽技术可改变处理优先等级



CPU 执行程序轨迹 (原屏蔽字)

(3) 屏蔽技术可改变处理优先等级



CPU 执行程序轨迹（新屏蔽字）

(4) 屏蔽技术的其他作用

可以 **人为地屏蔽** 某个中断源的请求

便于程序控制

(5) 新屏蔽字的设置

