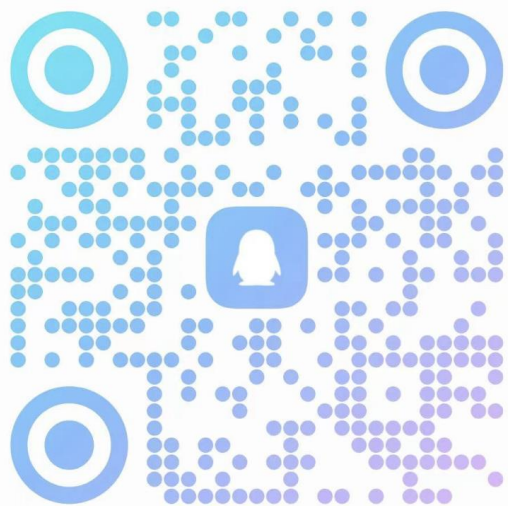


计算机组成原理

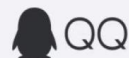


2025春计算机...

群号: 892932264



扫一扫二维码，入群聊



花忠云

<https://huazhongyun.github.io/>

<http://faculty.hitsz.edu.cn/huazhongyun>

计算机科学与技术学院

7.2 输入输出系统

7.2.1 概述

7.2.2 外部设备

7.2.3 I/O接口

7.2.4 程序查询方式

7.2.5 程序中断方式

7.2.6 DMA方式

7.2.1 概述

- 输入输出设备与特性
- 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

7.2.1 概述

一、输入输出设备与特性

- **输入输出设备**是计算机与人或者机器系统进行数据交互的装置，用于实现计算机内部二进制信息与外部不同形式信息的转换，简称**外部设备或外设**
 - **输入设备**：负责将数据、文字、图像、声音、电信号等转换成计算机可以识别的二进制信息，如键盘、鼠标、扫描仪、摄像头等
 - **输出设备**：负责将计算机处理结果转换成数字、文字、图形、图像、声音或电信号，如显示器、打印机等
 - **输入输出设备**：既能输入也能输出，如磁盘、网卡等
- **输入输出设备特性**
 - 异步性、实时性、独立性

7.2.1 概述

输入/输出系统的组成与功能

输入输出系统： 外部设备、接口部件、总线以及相应的管理软件的系统统称，简称I/O系统

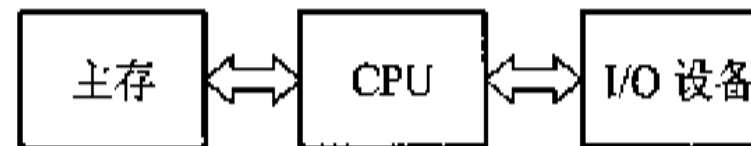
- 完成计算机内部二进制信息与外部多种信息形式间的交流
- 保证CPU能够正确选择I/O设备并实现对其控制，与数据传输
- 利用数据缓冲、合适的数据传送方式，实现主机外设间速度匹配



7.2.1 概述

二、输入输出系统的发展概况

I. 早期



- 分散连接

每个I/O设备都配有一套独立的逻辑电路与CPU相连，以完成I/O设备与主存的信息交换。

- CPU 和 I/O设备 串行 工作

输入输出过程穿插在程序执行过程之中进行，当I/O设备与主机交换信息时，CPU必须等待。

- 程序查询方式与主存交换信息

7.2.1 概述

二、输入输出系统的发展概况

II. 接口模块和 DMA 阶段

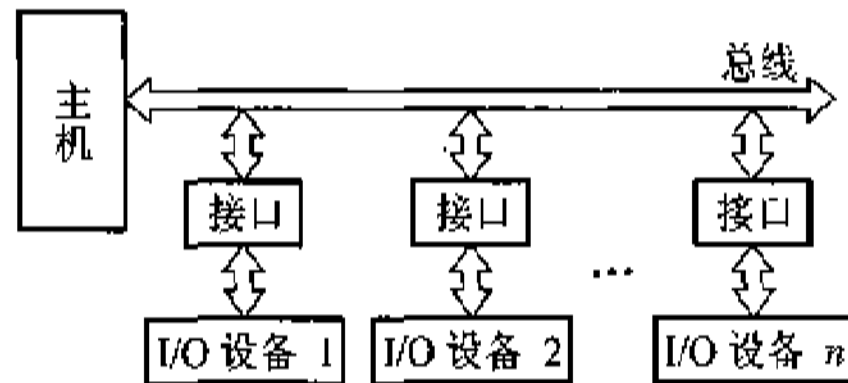
- 总线连接

计算机系统采用总线结构，I/O设备通过接口模块与主机连接。

- I/O设备和CPU按照 并行 方式工作

接口模块中包含有数据通路和控制通路，通过接口可以使得I/O设备和CPU按照并行方式工作，还可以使得多台I/O设备分时占用总线，使得I/O设备之间也可实现并行工作。

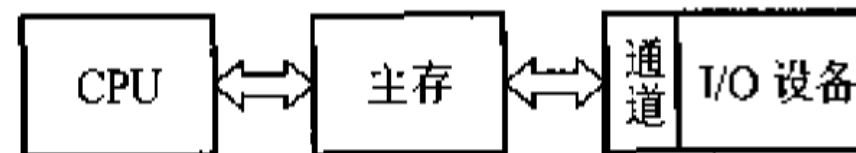
- 主要采用程序中断方式和DMA方式完成I/O设备与主机间的信息交换



7.2.1 概述

二、输入输出系统的发展概况

III. 具有通道结构的阶段



对于大型计算机，I/O设备数量非常多，数据传输频繁，采用DMA方式进行信息交换会出现成本高、控制复杂的问题。

- 大型计算机往往采用I/O通道的方式进行数据交换。
- **I/O通道**可以视为一个具有特殊功能的**处理器**，主要负责**管理**I/O设备以及实现主存与I/O设备之间的信息交换。
- 依赖I/O通道管理设备，不占用CPU的资源，**提高CPU的效率**。
- 从属于CPU的专用处理器：有专用通道指令，并不是完全独立的处理器，依据CPU的I/O指令进行启动、停止或改变工作状态。

7.2.1 概述

二、输入输出系统的发展概况

IV. 具有 I/O 处理机的阶段

- I/O处理机又称为外围处理机，独立于主机（CPU）工作。
- I/O处理机除了完成I/O通道需要完成的输入输出设备的I/O控制以外，还可以独立完成数据格式转换、数据检错纠错等任务。
- 具有I/O处理机的输入输出系统对主机来说具有更大的独立性，与CPU工作的并行性更高。

7.2.1 概述

三、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

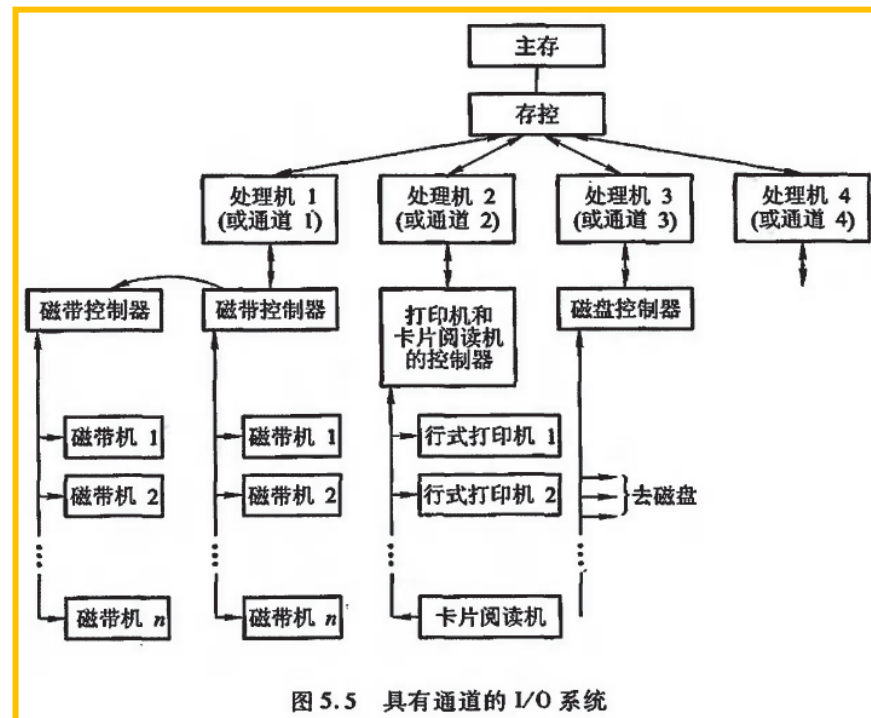
I/O 接口

设备

设备控制器

通道

设备码：给出的是I/O设备的编码/地址，或者是I/O设备当中某一个寄存器的地址。这些寄存器称之为I/O端口

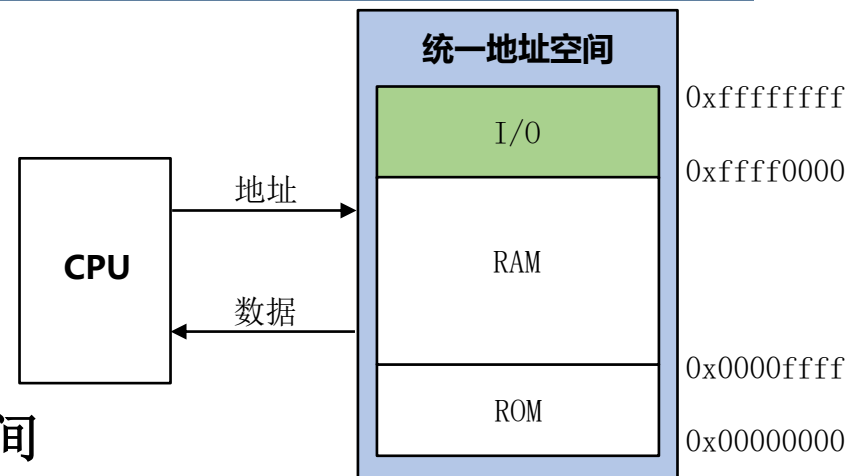


7.2.1 概述

四、I/O 设备与主机的联系方式

I. I/O 设备编址方式

- 统一编址
 - 内存映射编址（Memory-mapped），**占用主存**
 - 外设地址与内存地址统一编址，同一个地址空间
 - 不需要设置专用的I/O指令
 - 采用**访存指令**访问外设，具体访问什么设备取决于地址
- 独立编址（不统一编址）
 - 端口映射编址（Port-mapped），**不占用主存**
 - I/O地址空间与主存地址空间相互独立
 - I/O地址又称为I/O端口
 - 不同设备中的不同寄存器和存储器都有唯一的端口地址
 - **使用（专用）I/O指令访问外设**



7.2.1 概述

四、I/O 设备与主机的联系方式

II. 设备选址

每台I/O设备都有唯一设备号

用设备选择电路识别“I/O指令的设备码字段”是否被选中

III. 传送方式

(1) 串行

(2) 并行

7.2.1 概述

四、I/O 设备与主机的联系方式

IV. 联络方式

(1) 立即响应方式

- I/O设备与CPU发生联系时，通常已经处于某种等待状态。只要CPU的I/O指令一到，这些设备就可立即响应，不需要特殊的联络信号。
- 适用于工作速度十分缓慢的I/O设备，例如：指示灯的亮与灭。

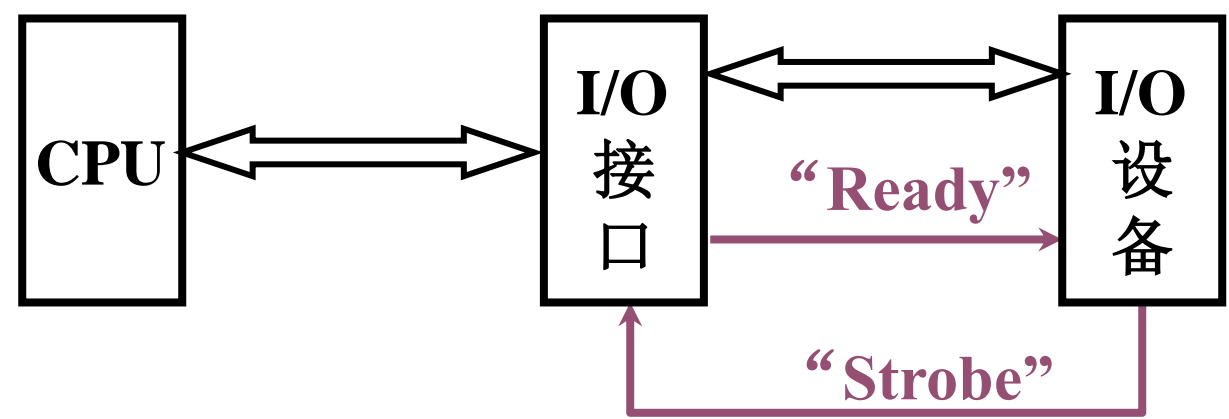
(2) 同步工作采用同步时标联络

- 需要在I/O设备和CPU之间配备**专有电路**，用以产生同步时标来控制同步工作。
- 例如：以I/O设备输入数据为例。假设I/O设备以2400bps的速率将数据传输到I/O接口，则CPU也必须以 $(1/2400)\text{s}$ 的速率接收每一位数据。
- 适用于I/O设备与CPU的工作速度完全同步的情况。

IV. 联络方式

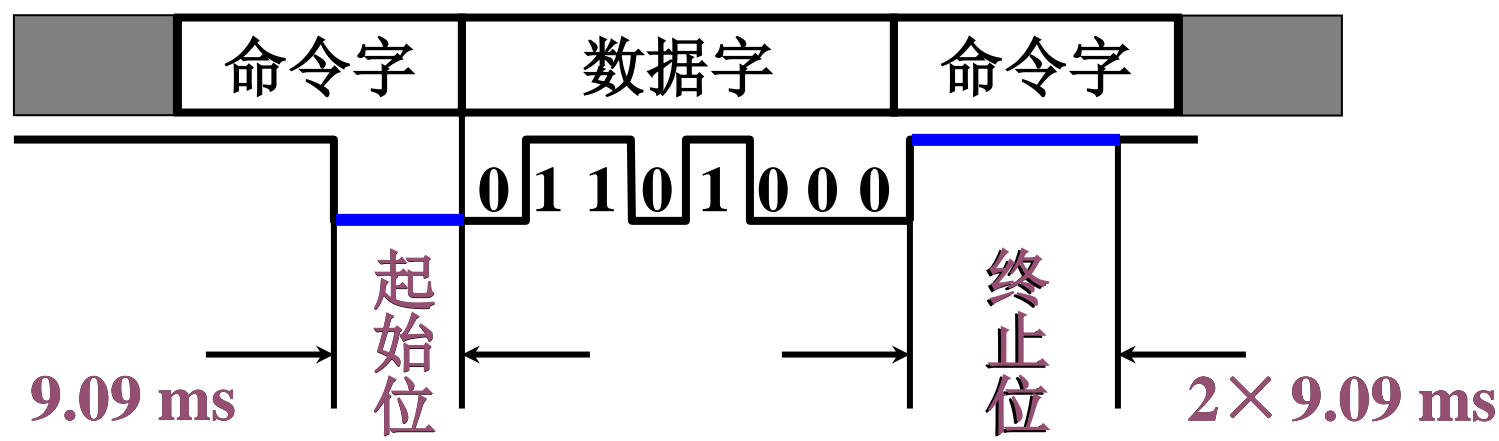
(3) 异步工作采用应答信号

并行



- 适用于I/O设备与主机工作速度不匹配情形。
- 在交换信息前，I/O设备和CPU各自执行自己的任务，一旦出现联络信号，彼此才交换信息。

串行 I/O设备与CPU双方设定一组特殊标记，用“起始”和“终止”来建立联系

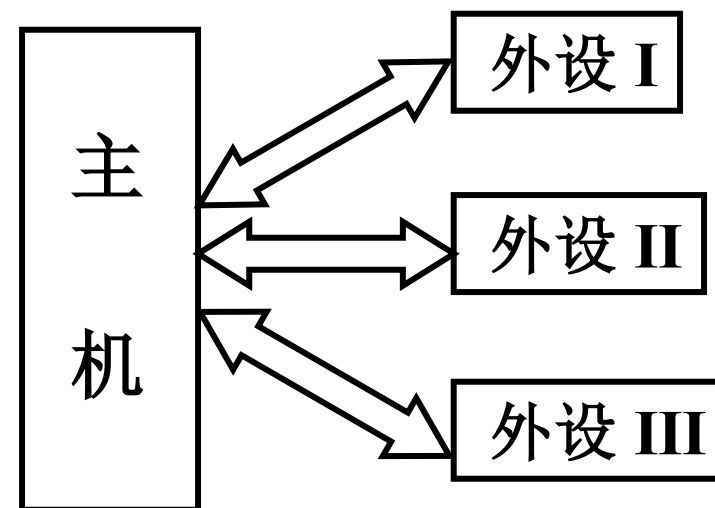


7.2.1 概述

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
- 不便于增删设备



(2) 总线连接

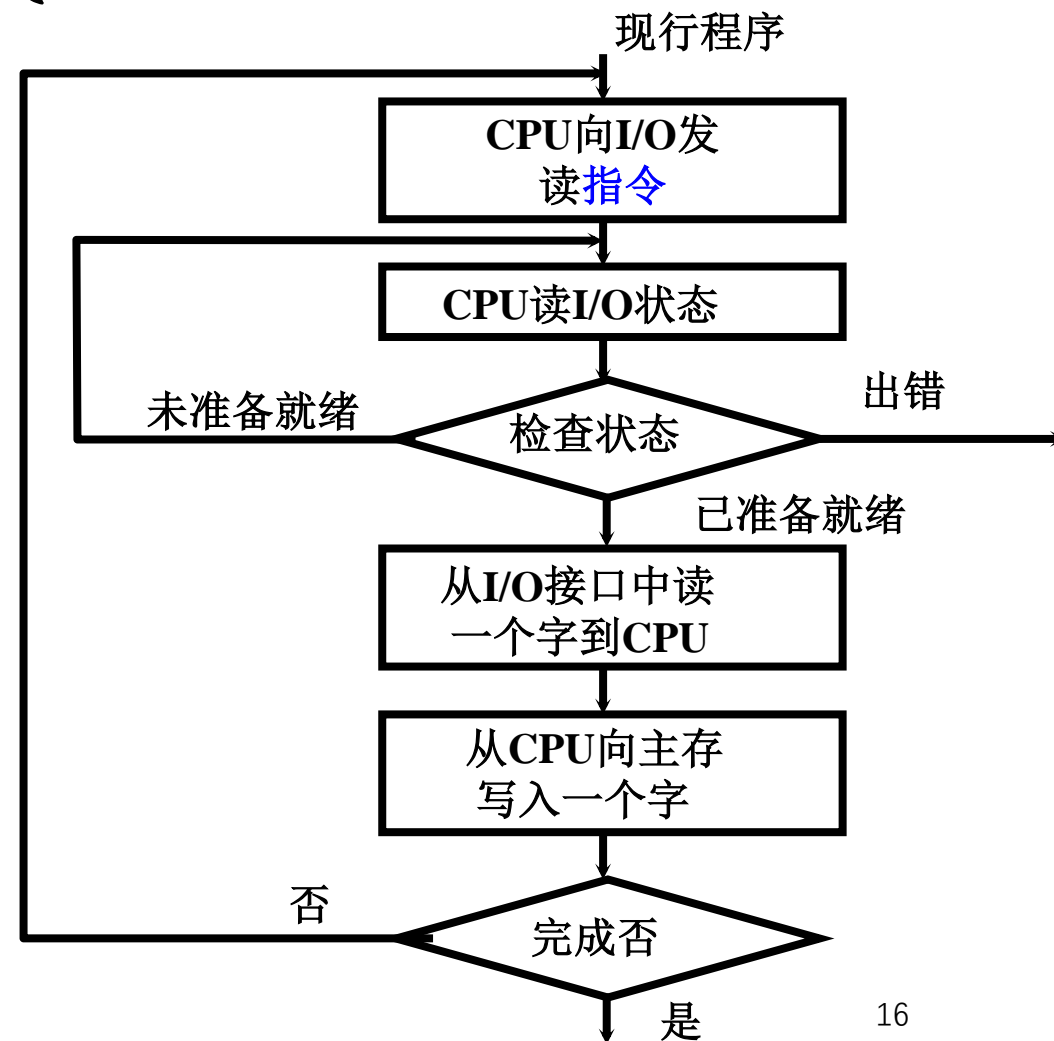
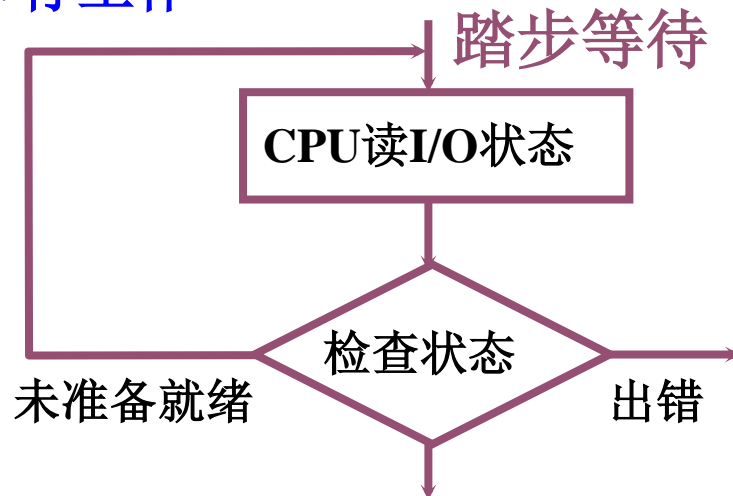
- 通过一组总线（包括地址线、数据线和控制线）将所有的I/O设备与主机相连。
- 便于增删设备

7.2.1 概述

五、I/O设备与主机信息传送的控制方式

I. 程序查询方式

- CPU不断查询I/O设备是否准备就绪，根据查询结果来决定I/O设备是否与主机交换信息的工作方式。
- 需要I/O接口设置一个能反映I/O设备是否准备就绪的状态标记，供CPU查询。
- CPU 和 I/O 串行工作

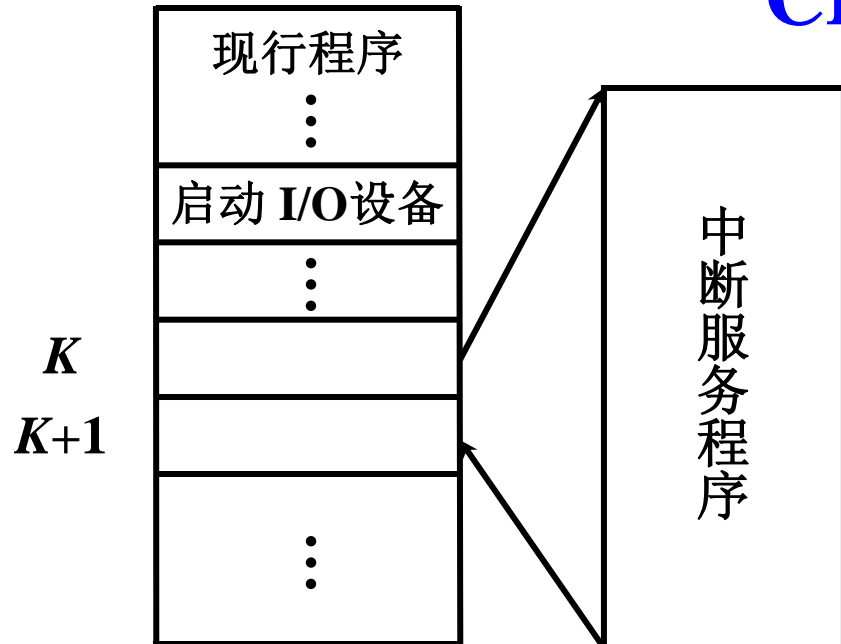


7.2.1 概述

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

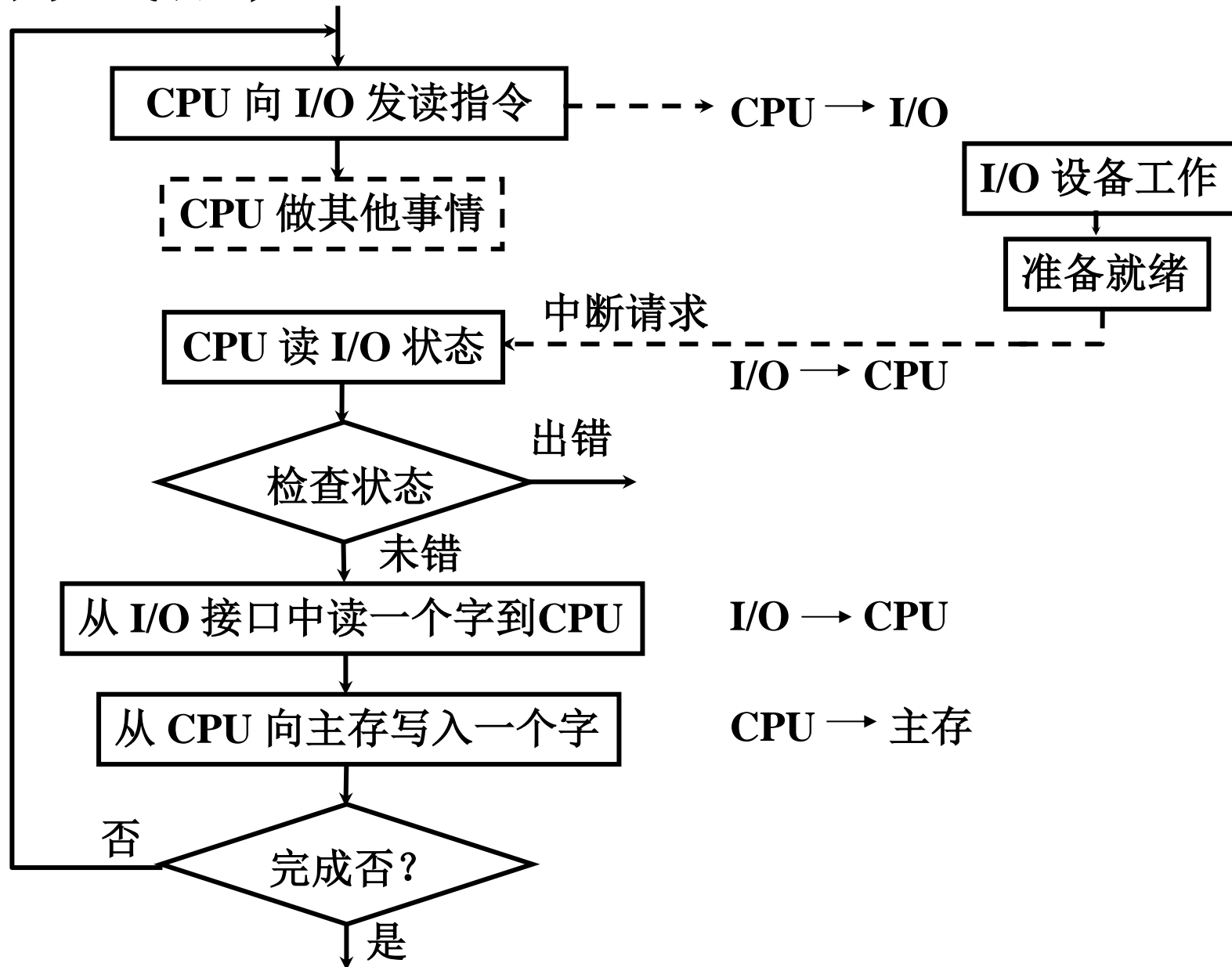
CPU 和 I/O 部分的并行工作



没有踏步等待现象

中断现行政程序

程序中中断方式流程



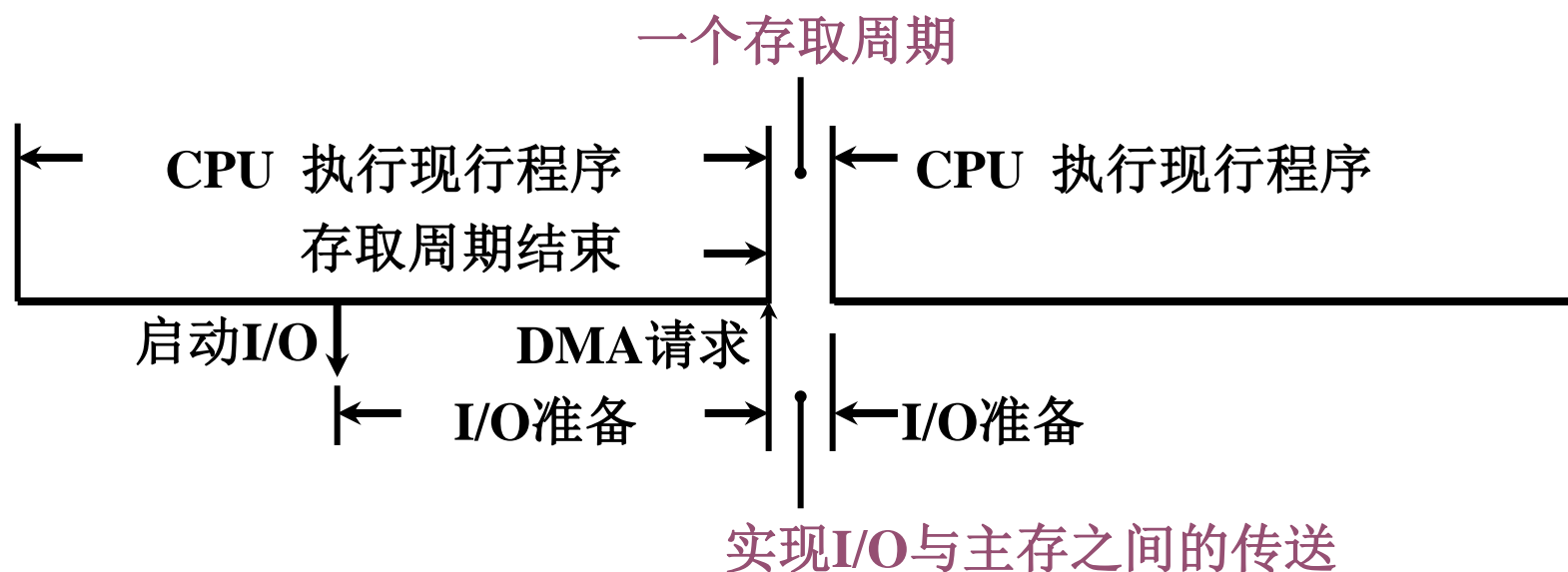
III. DMA 方式

主存和 I/O 之间有一条直接数据通道

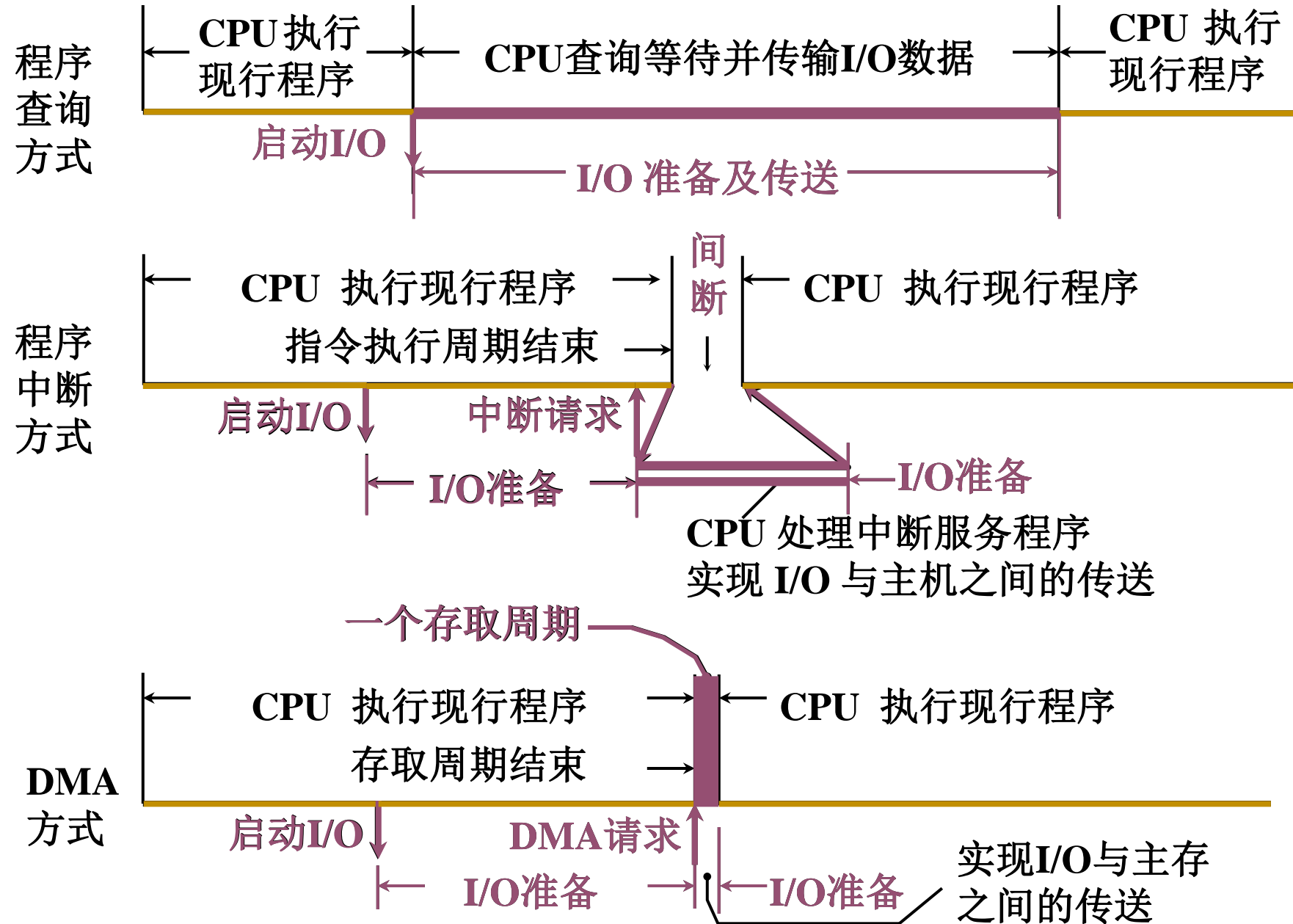
不中断现行程序

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



7.2 输入输出系统

7.2.1 概述

7.2.2 外部设备

7.2.3 I/O接口

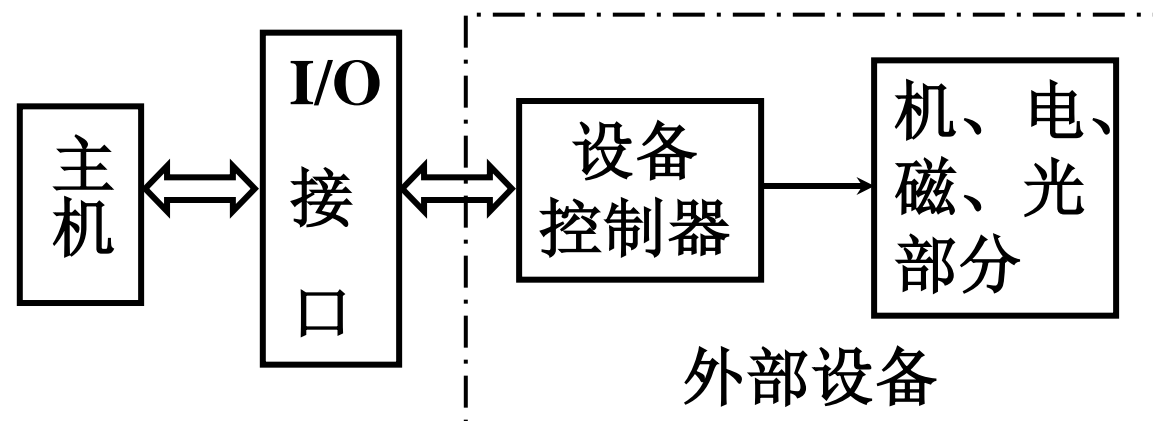
7.2.4 程序查询方式

7.2.5 程序中断方式

7.2.6 DMA方式

7.2.2 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 |



7.2.2 外部设备

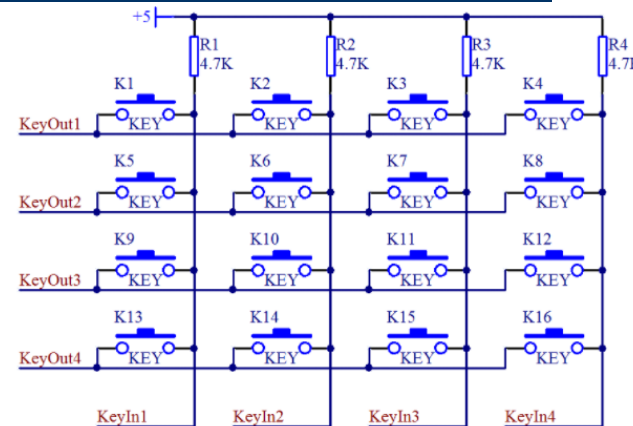
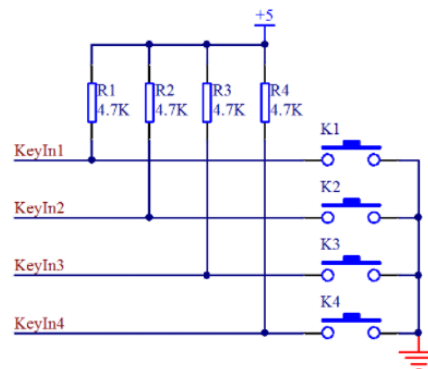
二、输入设备

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

7.2.2 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

由点、线、面组合成的平面或立体图形，
常用于计算机辅助设计（CAD）

(3) 图像显示

客观图像

自然景物、医学图像等



2. 打印机

(1) 击打式

点阵式（逐字、逐行）

通过机械动作使印字机构与色带和纸撞击，从而完成字符的打印。逐行打印比逐字打印更快

(2) 非击打式

激光（逐页）

喷墨（逐字）



7.2.2 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？ 媒体：音乐、语言、图片、文件、视频等。
多媒体：集成多种媒体功能的系统。
- 2. 多媒体计算机的关键技术
视频和音频压缩与解压缩、多媒体芯片、大容量存储器、
多媒体软件等各种核心计算机技术。

7.2 输入输出系统

7.2.1 概述

7.2.2 外部设备

7.2.3 I/O接口

7.2.4 程序查询方式

7.2.5 程序中断方式

7.2.6 DMA方式

7.2.3 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”）

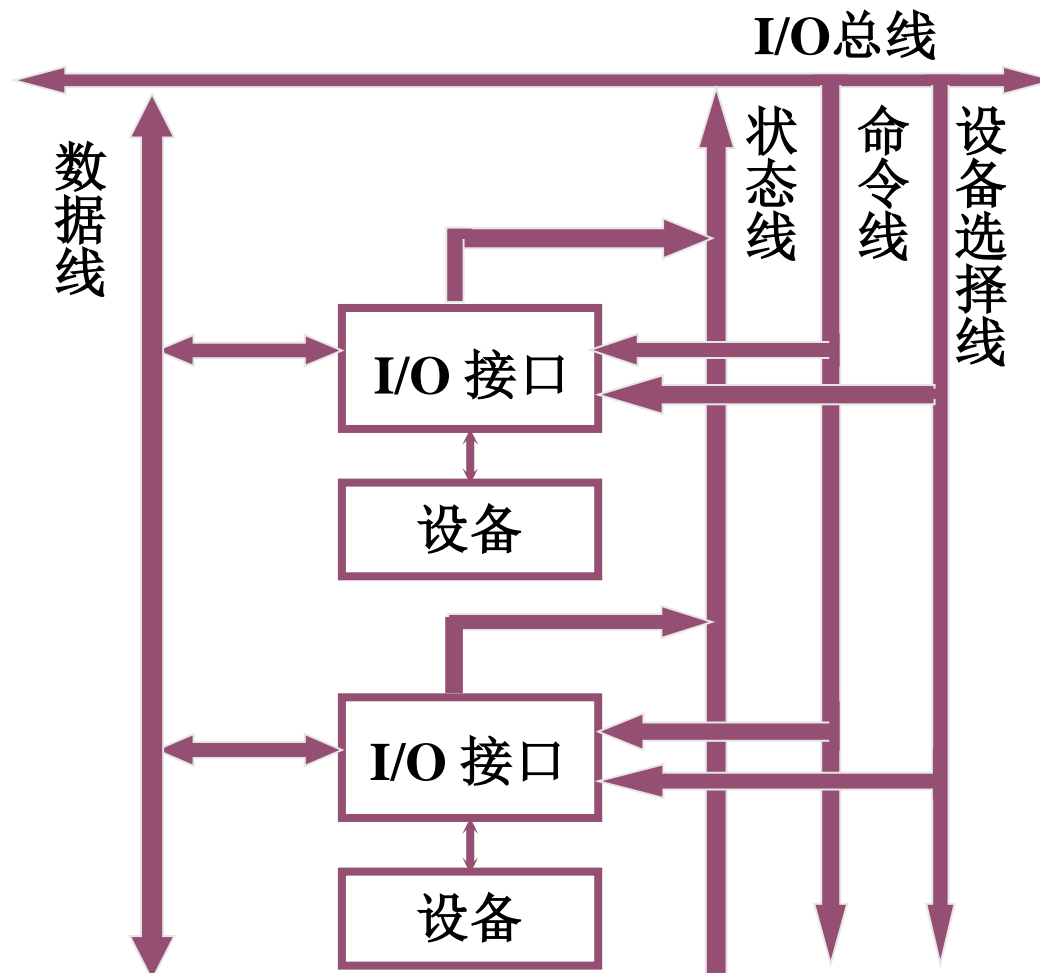


7.2.3 I/O接口

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**

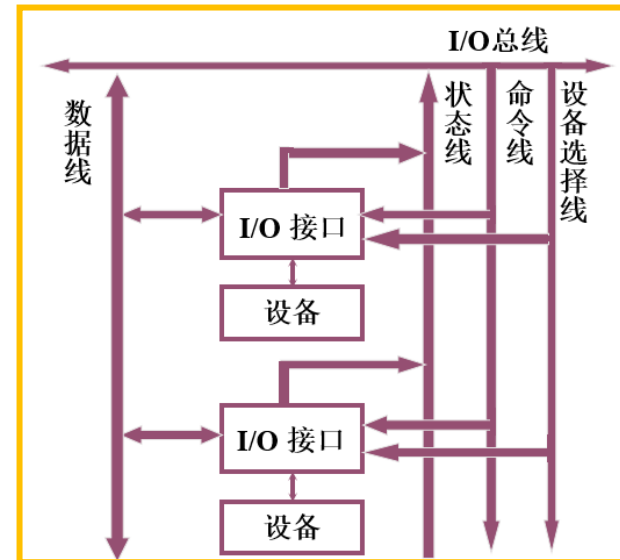
组成

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



例如：用完成触发器D和工作触发器B的值来标记设备所处的状态：

D=0, B=0时，表示设备处于暂停状态；

D=1, B=0时，表示设备处于就绪状态；

D=0, B=1时，表示设备处于准备中状态。

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

可编程并行I/O接口芯片，
有3个8位并行I/O口

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 （键盘、显示器）

4. 按数据传送的 控制方式 分类

中断接口

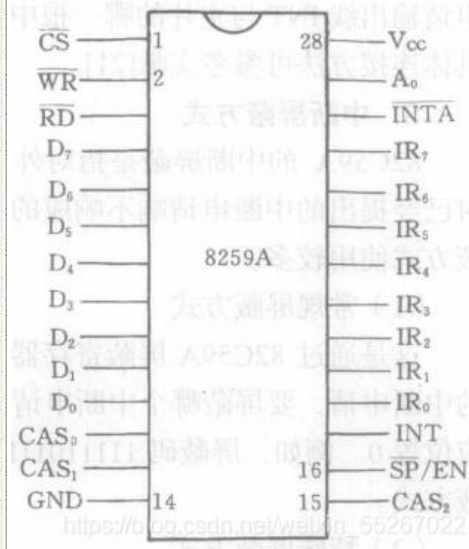
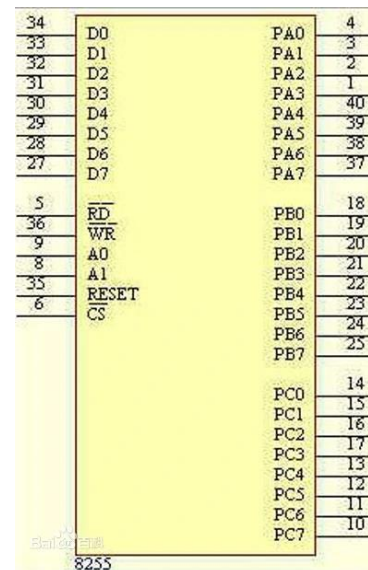
Intel 8259

主要用于速度较慢的I/O设备（打印机等）

DMA 接口

Intel 8237

主要用于高速的I/O设备（磁盘等）



7.2 输入输出系统

7.2.1 概述

7.2.2 外部设备

7.2.3 I/O接口

7.2.4 程序查询方式

7.2.5 程序中断方式

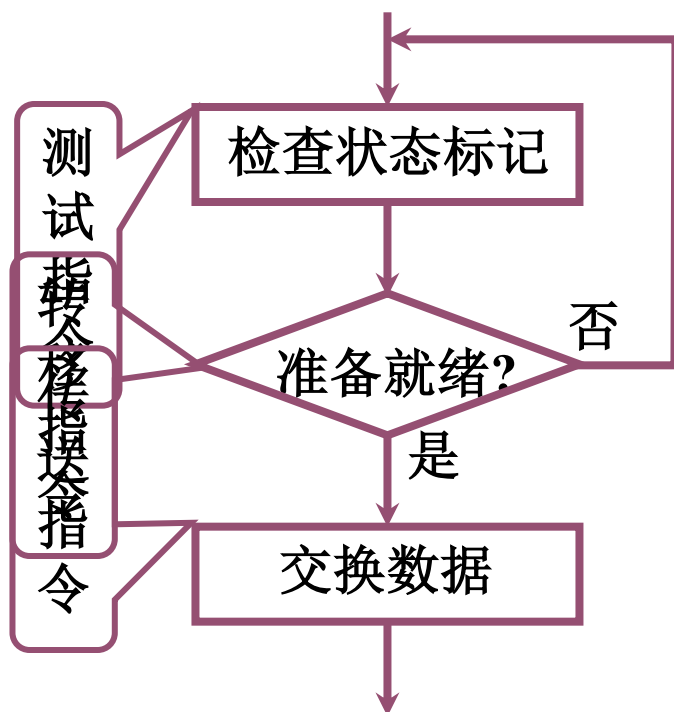
7.2.6 DMA方式

7.2.4 程序查询方式

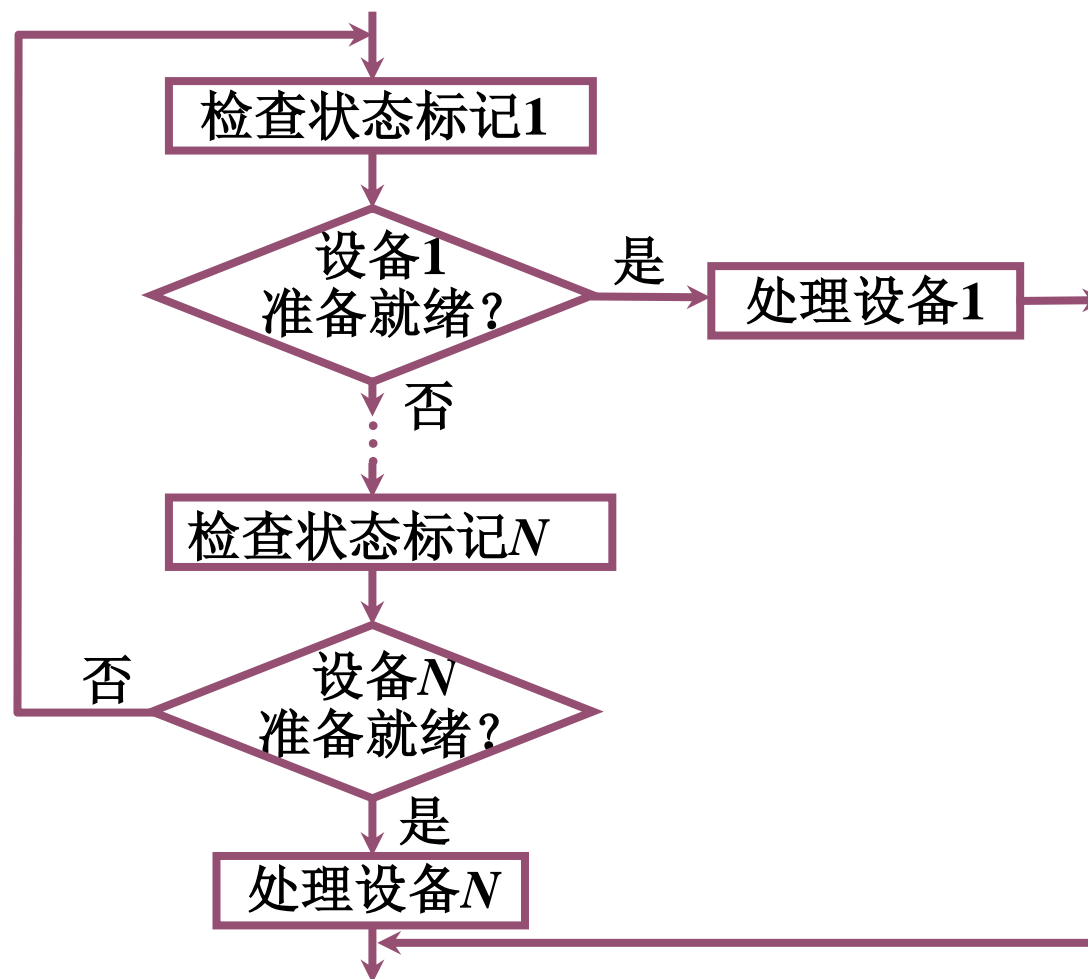
一、程序查询流程

1. 查询流程

单个设备



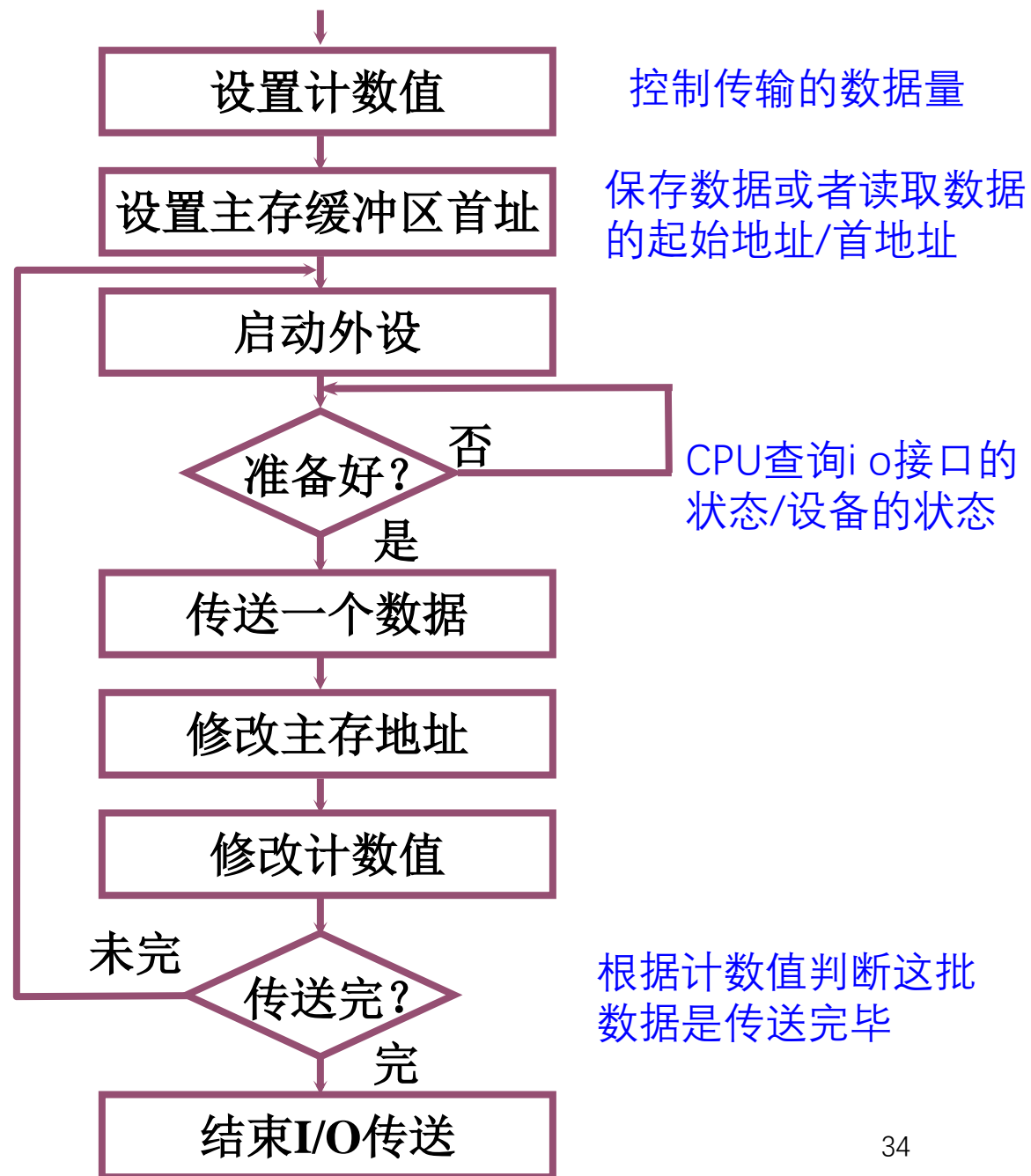
多个设备



2. 程序流程

保存寄存器内容

- 内存与外设之间的数据输入输出需要借助CPU当中的**某一个寄存器**对数据进行暂存
- 如果这个寄存器中的原数据是后面程序需要使用的，就需要对这个数据进行保存，把该寄存器中的值**写入到某一个内存单元中（压栈）**或**保存到其他闲置寄存器**



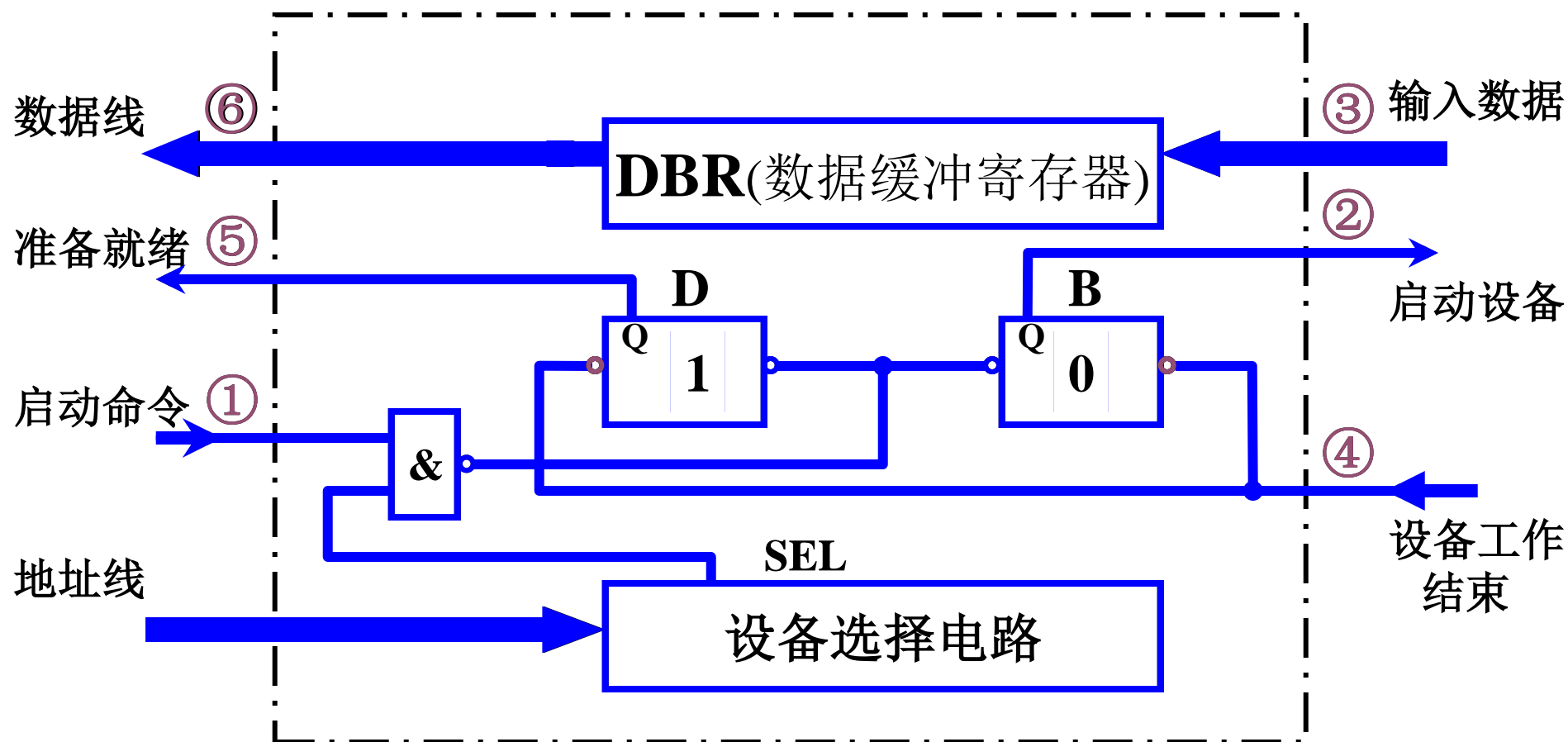
7.2.4 程序查询方式

二、程序查询方式的接口电路

以输入为例：外部数据输入到主机内存

完成触发器 D

工作触发器 B



I/O指令实现的数据传送通常发生在（）

- ☐ A I/O设备与I/O端口之间
- ☐ B 通用寄存器和I/O设备之间
- ☐ C I/O端口和I/O端口之间
- ☒ D 通用寄存器和I/O端口之间

提交

I/O端口：接口电路中可被CPU访问的寄存器；I/O接口中用于缓冲信息的寄存器

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  “操作”占CPU的时间片段比例
由此得出什么结论？

解: **50 MHz** 相当于 50×10^6 个时钟周期/秒

鼠标查询 比率 $[30 \times 100 / (50 \times 10^6)] \times 100\% = 0.006\%$

鼠标查询基本不
影响CPU性能

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

CPU将全部时间用于
硬盘查询都不够用~

结论: CPU 一般 不采用 程序查询方式 与磁盘交换信息

7.2 输入输出系统

7.2.1 概述

7.2.2 外部设备

7.2.3 I/O接口

7.2.4 程序查询方式

7.2.5 程序中断方式

7.2.6 DMA方式

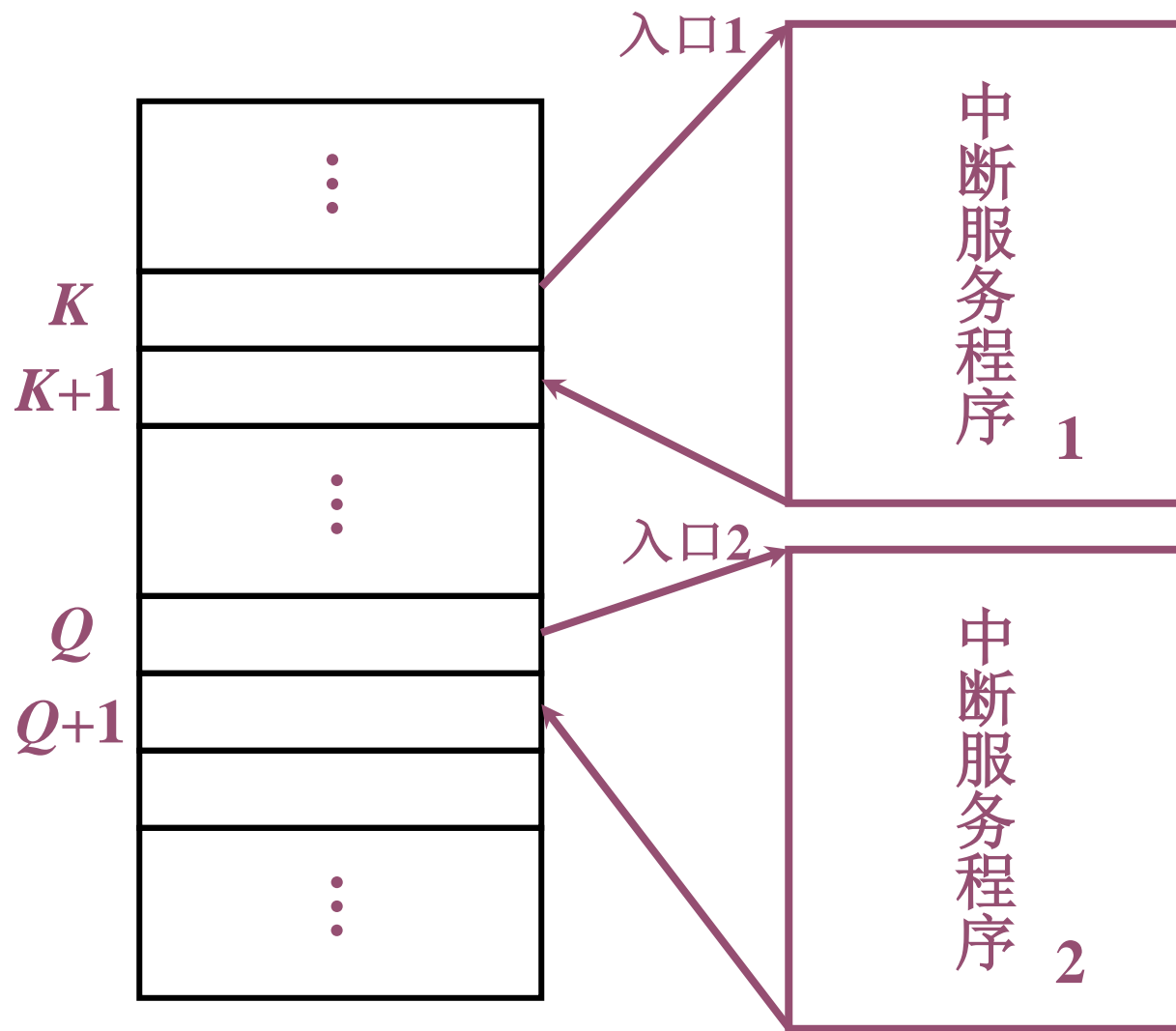
7.2.5 程序中中断方式

- 一. 中断的概念
- 二. 中断的产生
- 三. 程序中中断方式的接口电路
- 四. 中断处理过程(中断响应)
- 五. **单重/多重**中断服务程序流程（CPU）
- 六. 中断屏蔽技术（CPU）

一、中断的概念

计算机在执行程序的过程中，当出现异常情况或特殊请求时，

- 计算机停止现程序的运行
- 转向对这些异常情况或特殊请求的处理，
- 处理结束后再返回到现程序的间断处，继续执行原程序。



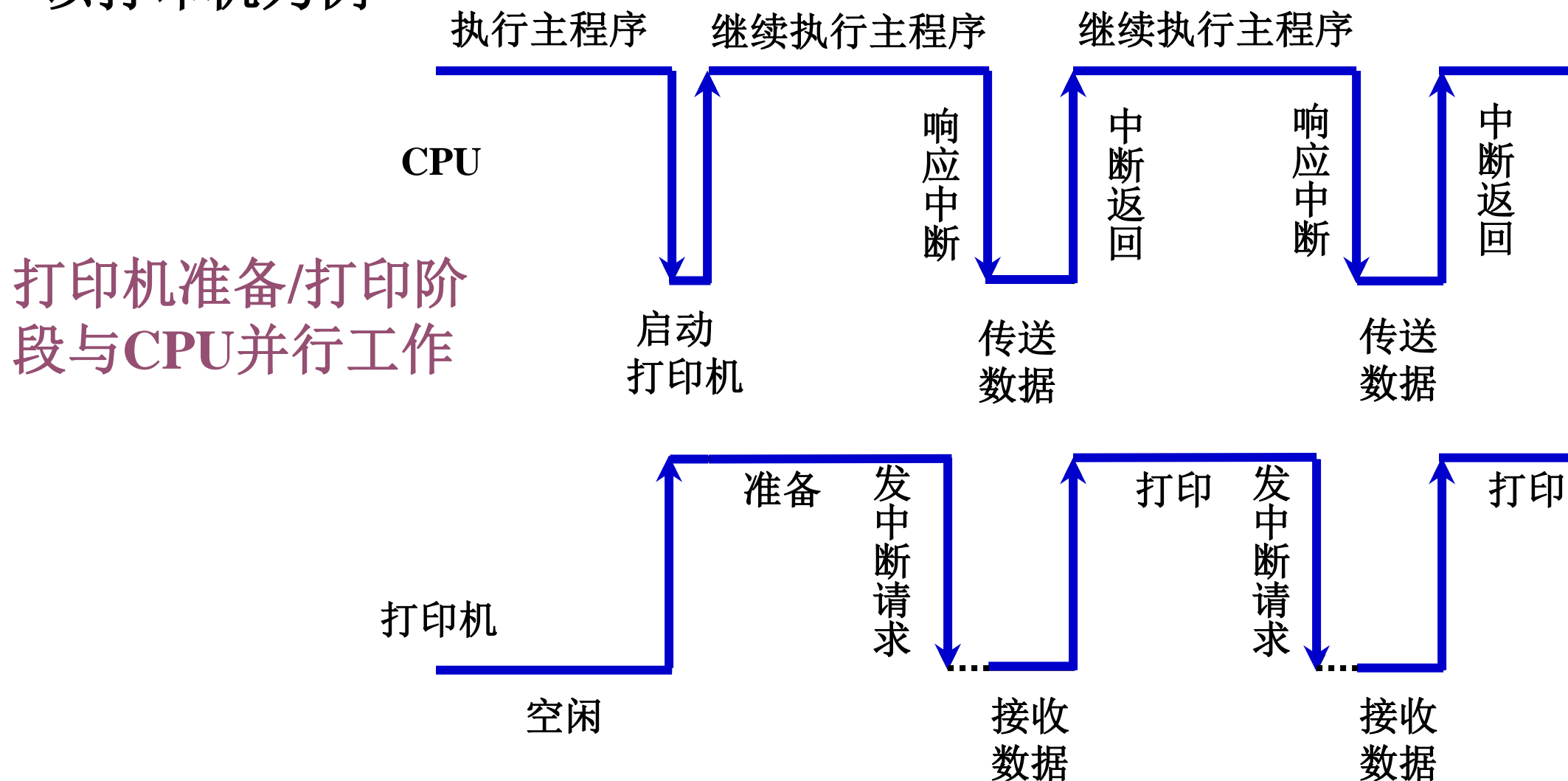
中断源：引起中断的各种因素

- 人为设置的中断 转管指令、IBM PC (Intel 8086) 的 INT TYPE系统调用指令
- 程序异常 溢出、操作码不能识别、除法非法
- 硬件故障 磁表面损坏、电源掉电、插件接触不良
- I/O 设备
- 外部事件 用 键盘中断 现行程序

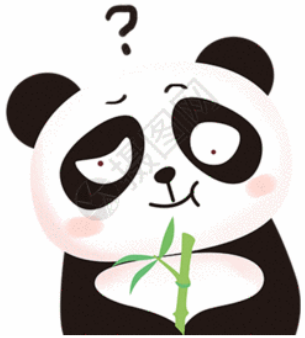
中断源 { 不可屏蔽中断：如电源掉电
 可屏蔽中断：CPU可根据该中断源是否被屏蔽来确定是否给与响应

二、 中断的产生——打印机为例

以打印机为例



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

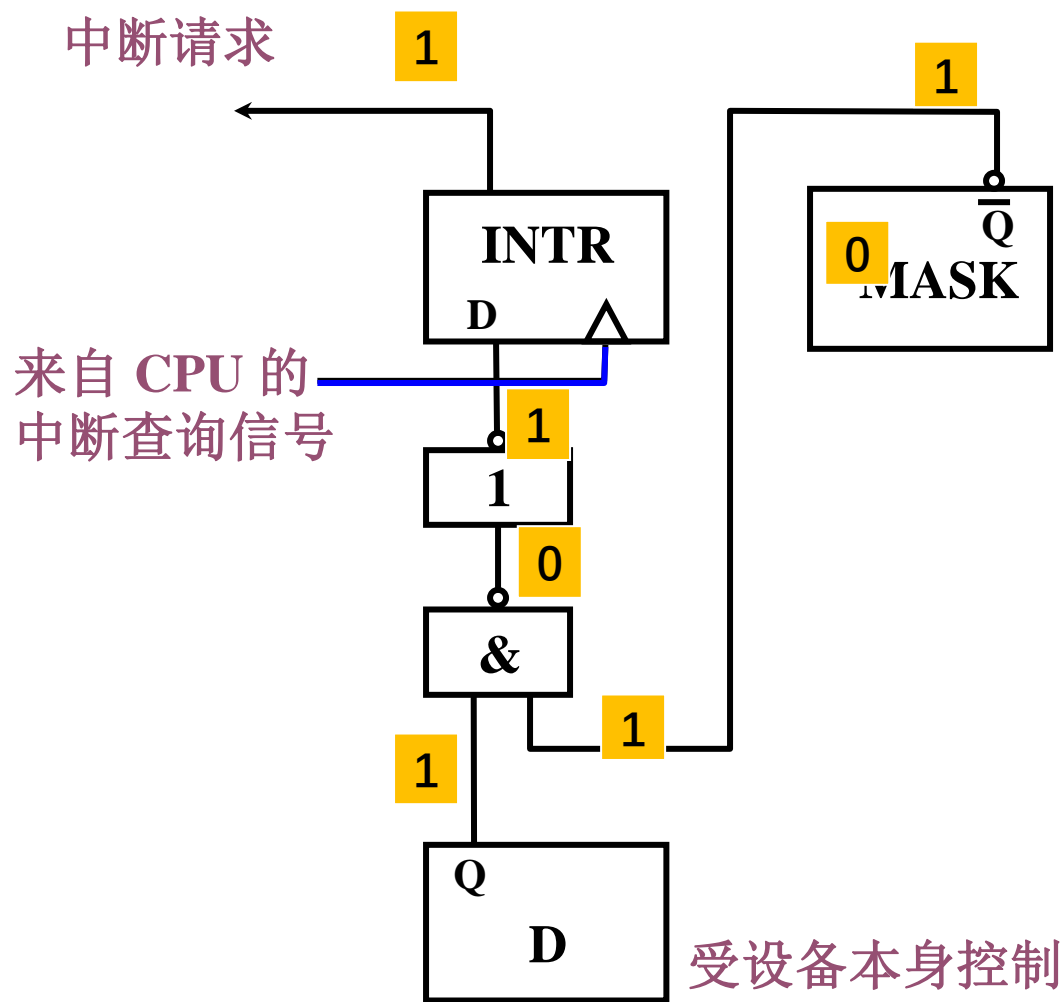
三、程序中断方式的接口电路

- 配置中断请求触发器和中断屏蔽触发器
- （中断）排队器——中断判优逻辑
- 中断向量地址形成部件
- 程序中断方式接口电路的基本组成

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

当且仅当设备准备就绪，且该设备未被屏蔽时，CPU的中断查询信号可将中断请求触发器置1



INTR

中断请求触发器

INTR = 1 有请求

MASK

中断屏蔽触发器

MASK = 1 被屏蔽

D 完成触发器

中断请求触发器INTR

一个中断请求源 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

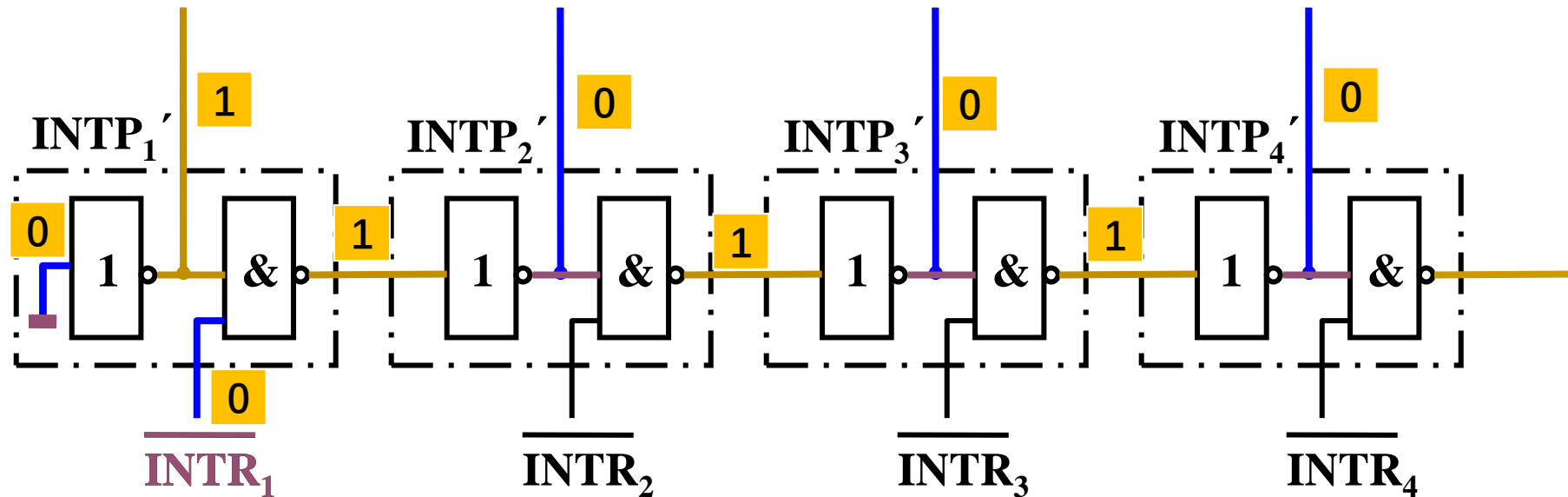
7.2.5 程序中中断方式

- 一. 中断的概念
- 二. 中断的产生
- 三. 程序中中断方式的接口电路
- 四. 中断处理过程(中断响应)
- 五. 单重/多重中断服务程序流程 (CPU)
- 六. 中断屏蔽技术 (CPU)

三、程序中中断方式的接口电路

2. (中断) 排队器

排队 { 硬件 在 CPU 内或在接口电路中 (链式排队器)
 软件

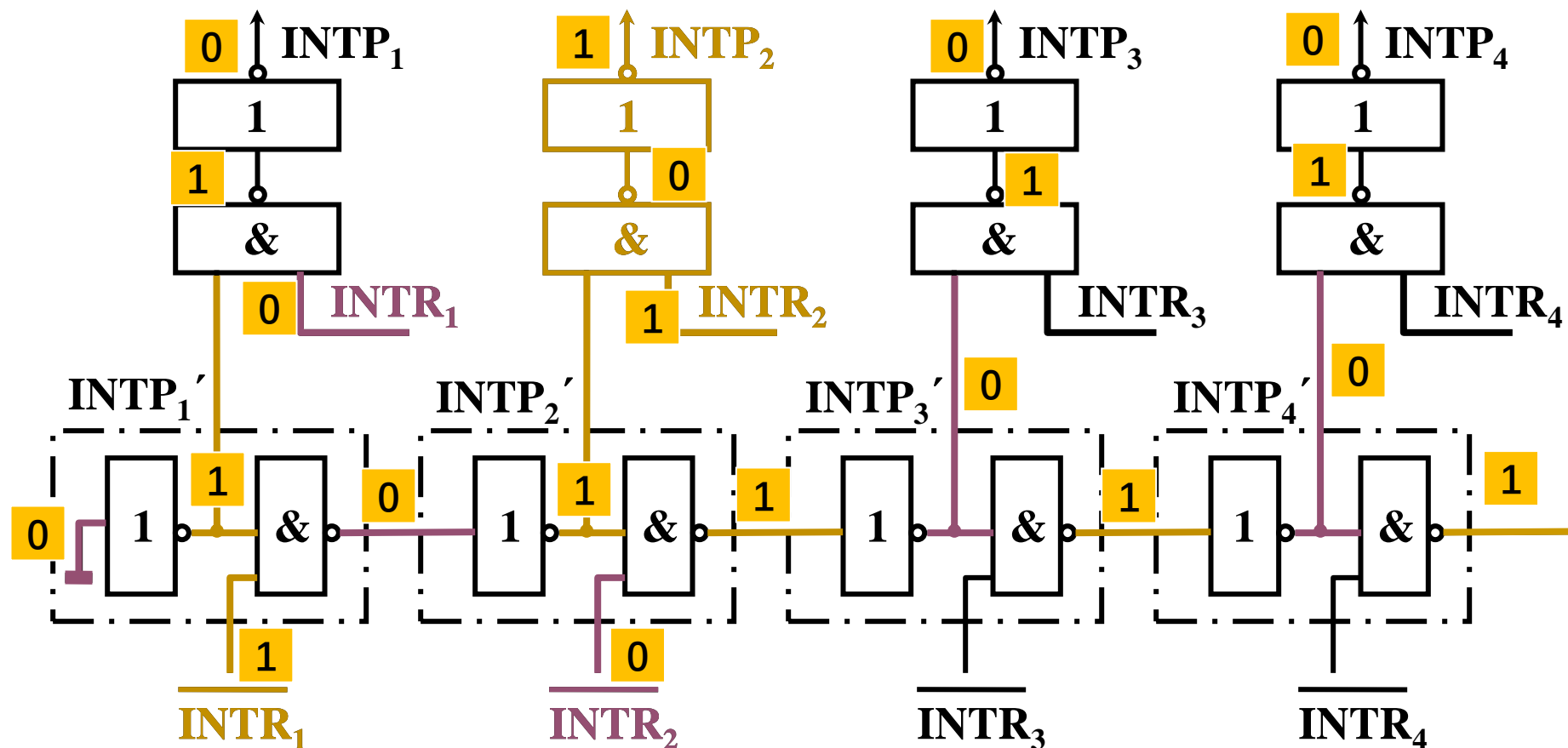


设备 1[#]、2[#]、3[#]、4[#] 优先级按 降序排列

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

三、程序中断方式的接口电路

2. (中断) 排队器

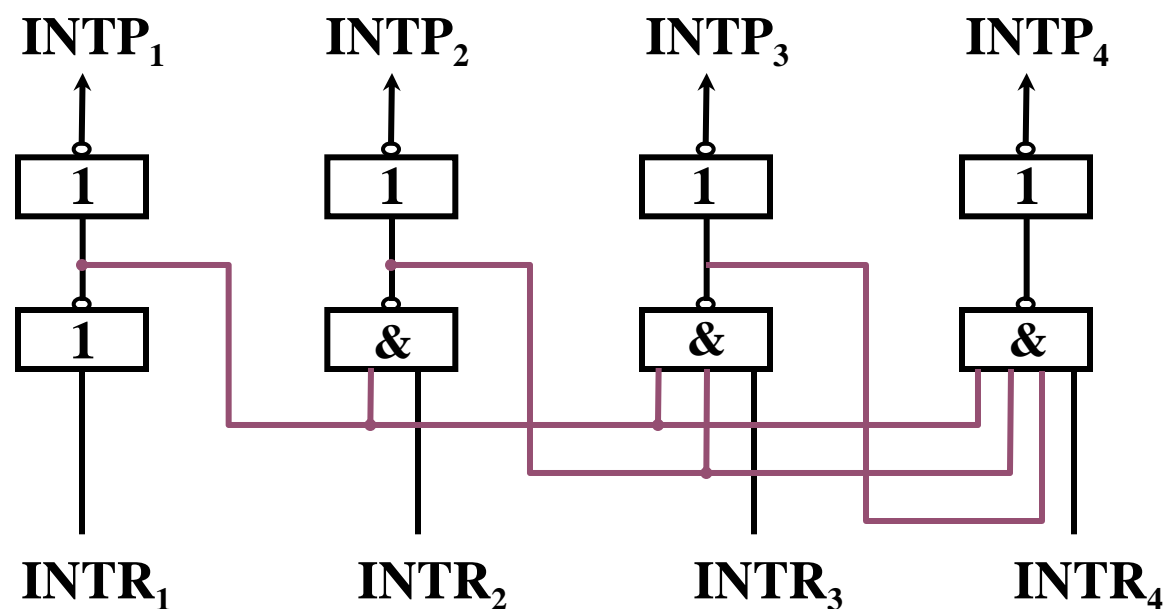


中断判优逻辑（CPU端）

(1) 硬件实现（排队器）

① 分散 在各个中断源的 接口电路中 链式排队器

② 集中 在 CPU 内

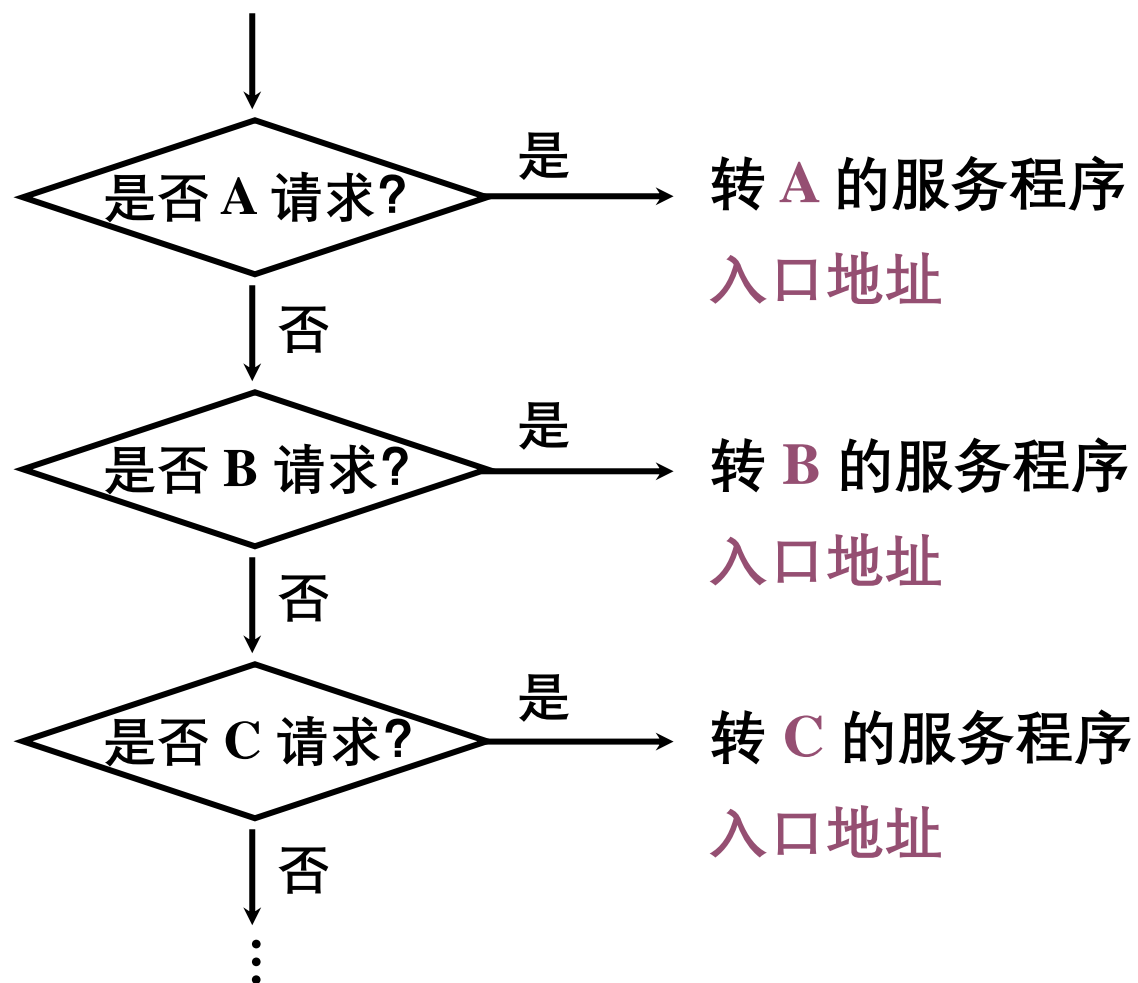


INTR₁、INTR₂、INTR₃、INTR₄ 优先级按降序排列

（中断）排队器——软件实现（程序查询）

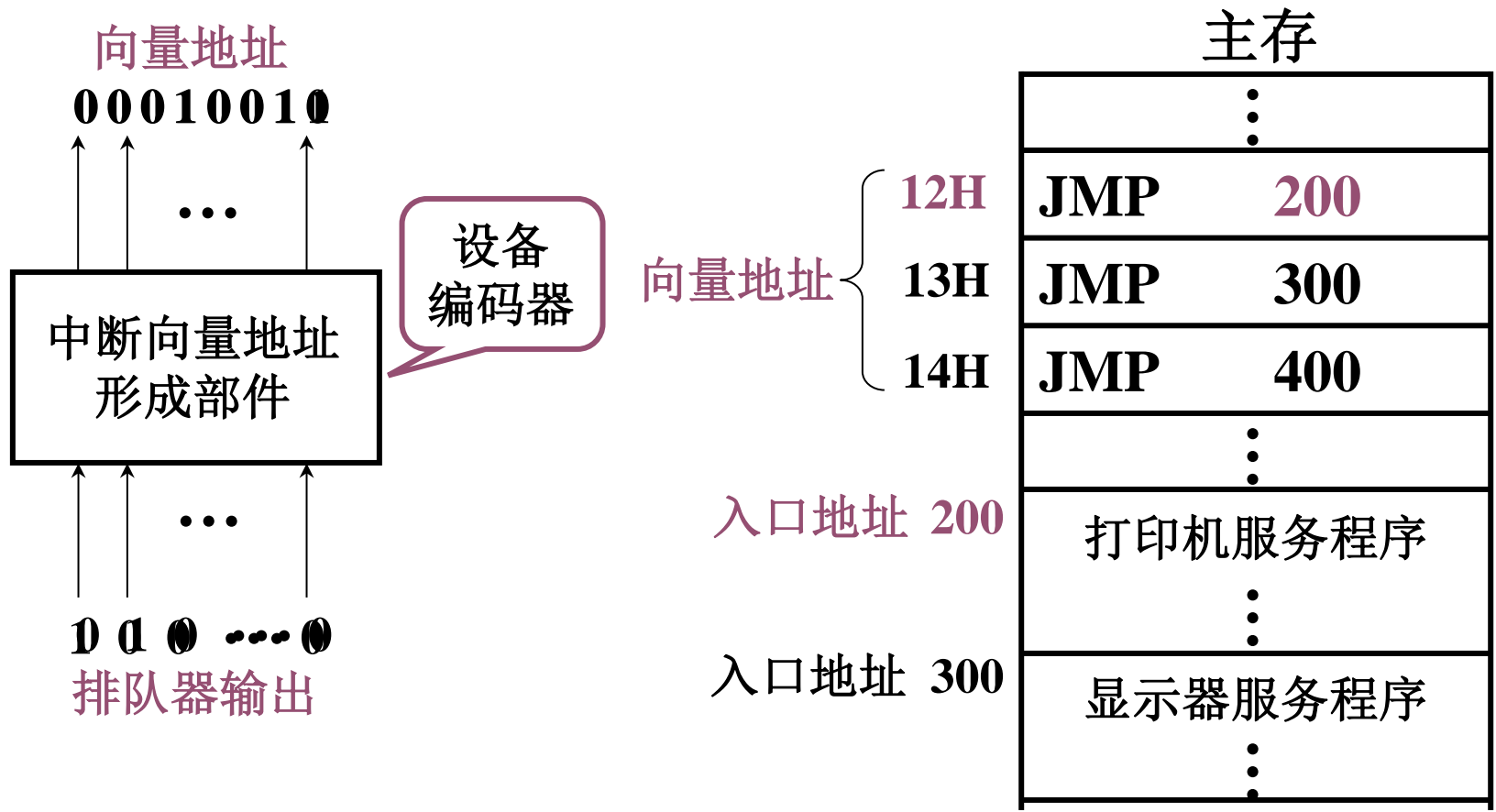
A、B、C 优先级按 降序 排列

优先级从高到低逐级查询



3. 中断向量地址形成部件——查找中断服务程序入口

中断服务程序入口地址 { 由软件产生
 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



中断服务程序入口地址——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**)

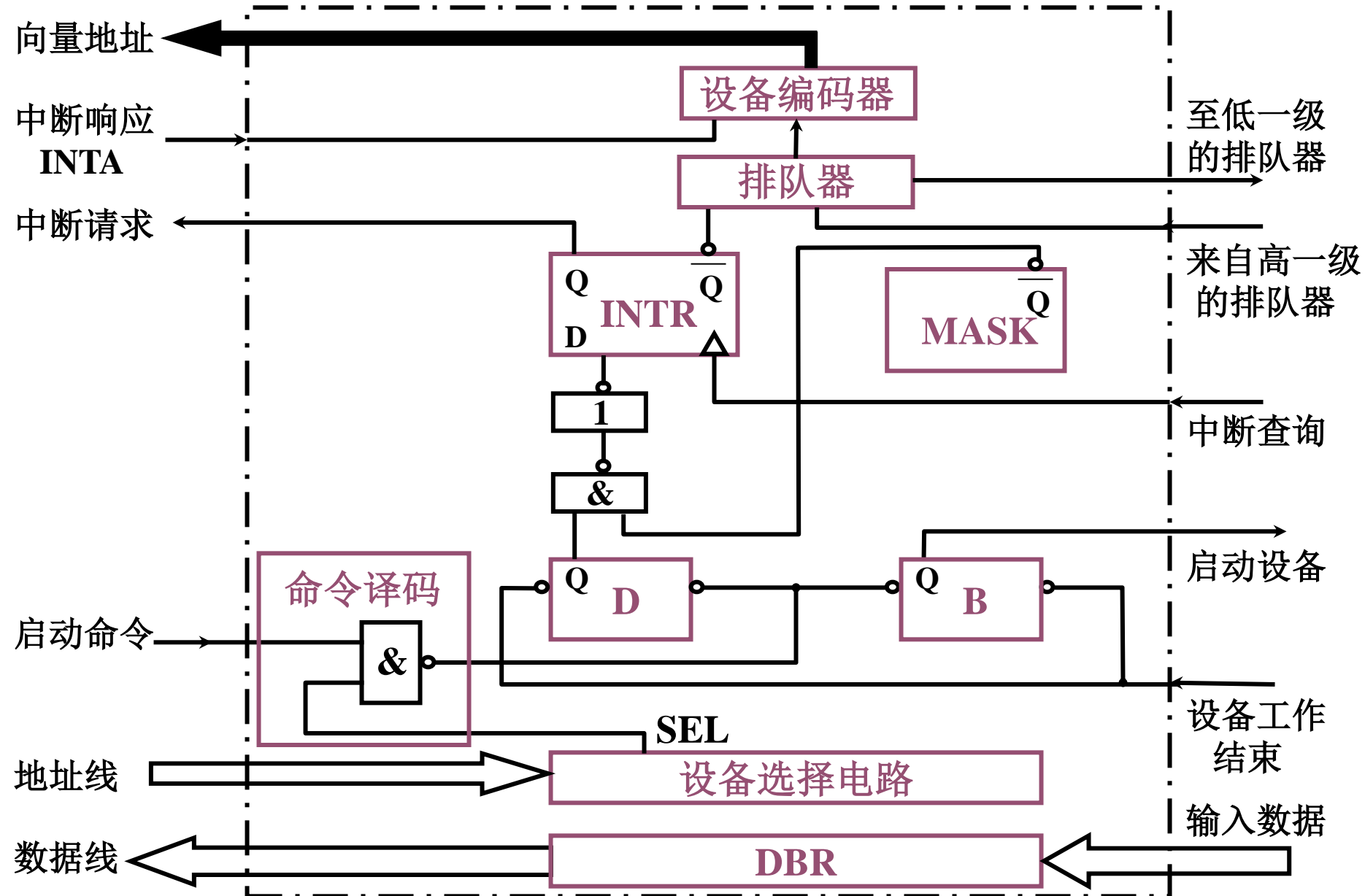
用软件寻找中断服务程序入口地址的方法称为软件查询法

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

当查到某一中断源有中断请求时，接着执行一条转移指令，直接指向此中断源的中断服务程序入口地址

- 各中断源对应的入口地址，由程序员（系统）事先确定。
- 不涉及硬件设备，查询时间较长。

4. 程序中断方式接口电路的基本组成



7.2.5 程序中中断方式

- 一. 中断的概念
- 二. 中断的产生
- 三. 程序中中断方式的接口电路
- 四. 中断处理过程(中断响应)
- 五. 单重/多重中断服务程序流程 (CPU)
- 六. 中断屏蔽技术 (CPU)

四、中断响应

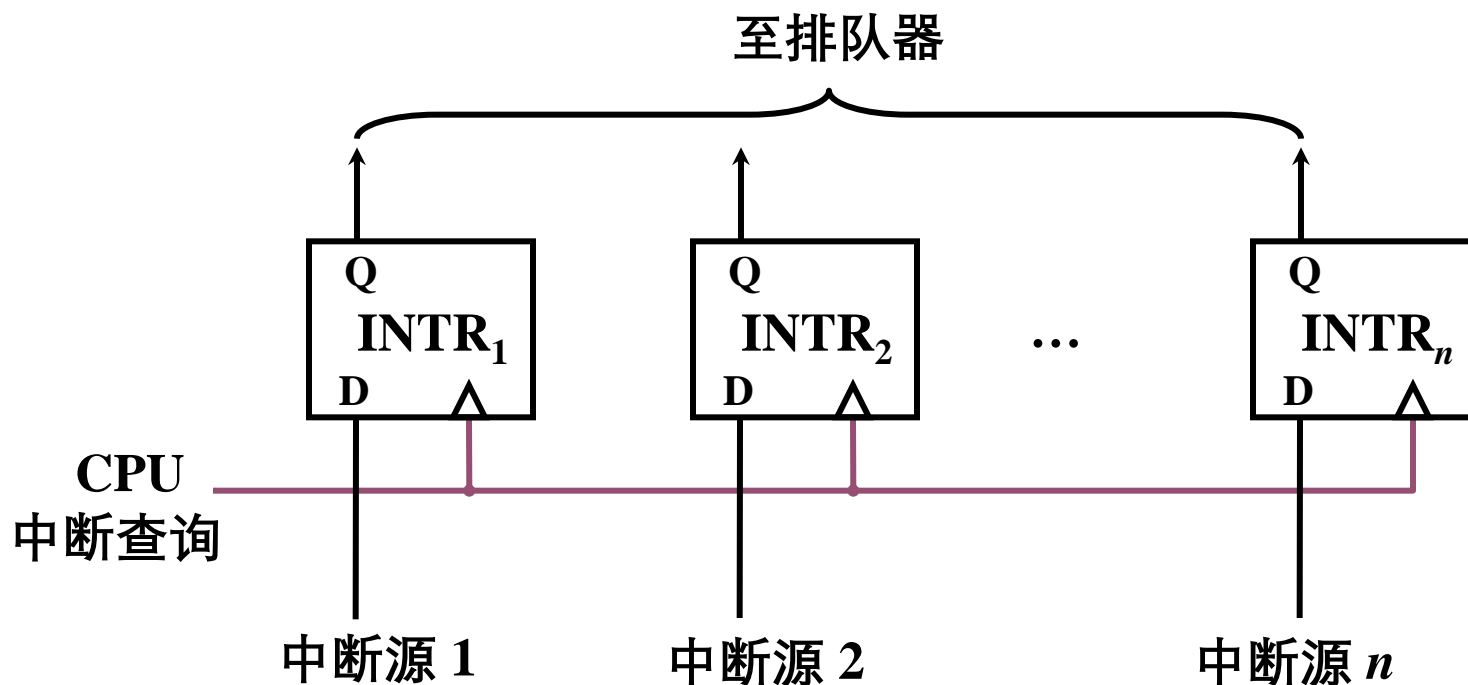
在某些计算机中，由于指令执行时间很长，需要在指令执行过程中设置若干个查询断点，CPU在每个“查询断点”时刻均发出中断查询信号，以便及时响应中断请求。

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

隐指令：机器指令系统没有的指令，它是CPU在中断周期内由硬件自动完成的一条指令

(1) 保护程序断点

- 保护程序断点、寻找入口地址、关中断在中断周期内由一条隐指令完成。

断点（程序计数器PC的内容）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 中断标记触发器

EINT 允许中断触发器

R-S 触发器

INT=1, EINT=0

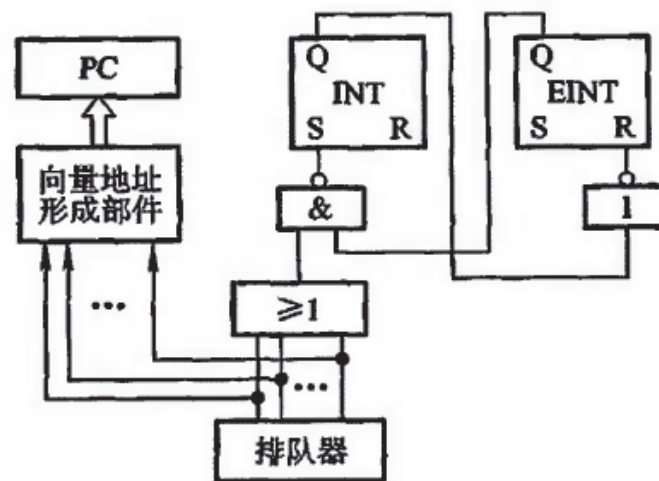


图 8.30 硬件关中断示意图

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

在每条指令执行周期结束时刻

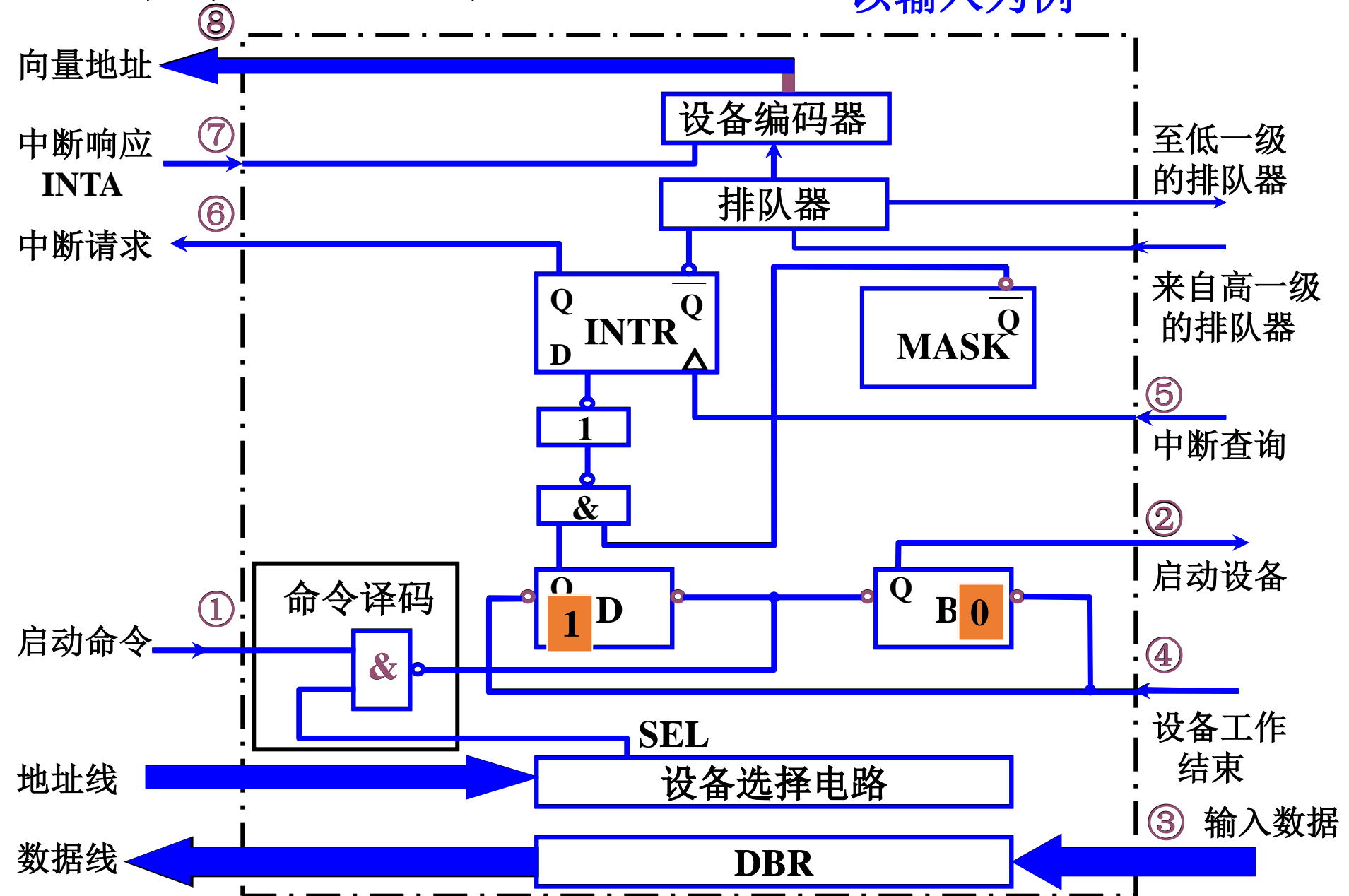
CPU 发 中断查询信号（将 **INTR** 置 “**1**”）

问题： 考虑到有些指令执行时间较长，若CPU的查询信号一律安排在执行周期结束时刻，有可能因CPU发现中断请求过迟而出差错。

解决方案： 可在指令执行过程中，设置若干查询断点

2. I/O 中断处理过程

以输入为例



7.2.5 程序中中断方式

- 一. 中断的概念
- 二. 中断的产生
- 三. 程序中中断方式的接口电路
- 四. 中断处理过程(中断响应)
- 五. 单重/多重中断服务程序流程 (CPU)
- 六. 中断屏蔽技术 (CPU)

五、单重/多重中断服务程序流程

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

{ 通用寄存器和状态寄存器内容的保护

进栈指令（中断服务程序完成）

(2) 中断服务

对不同的 I/O 设备服务（不同中断请求源，中断服务操作内容不同）

(3) 恢复现场

取数/出栈指令，将保存在存储器/堆栈中的信息送回原寄存器

(4) 中断返回

中断返回指令

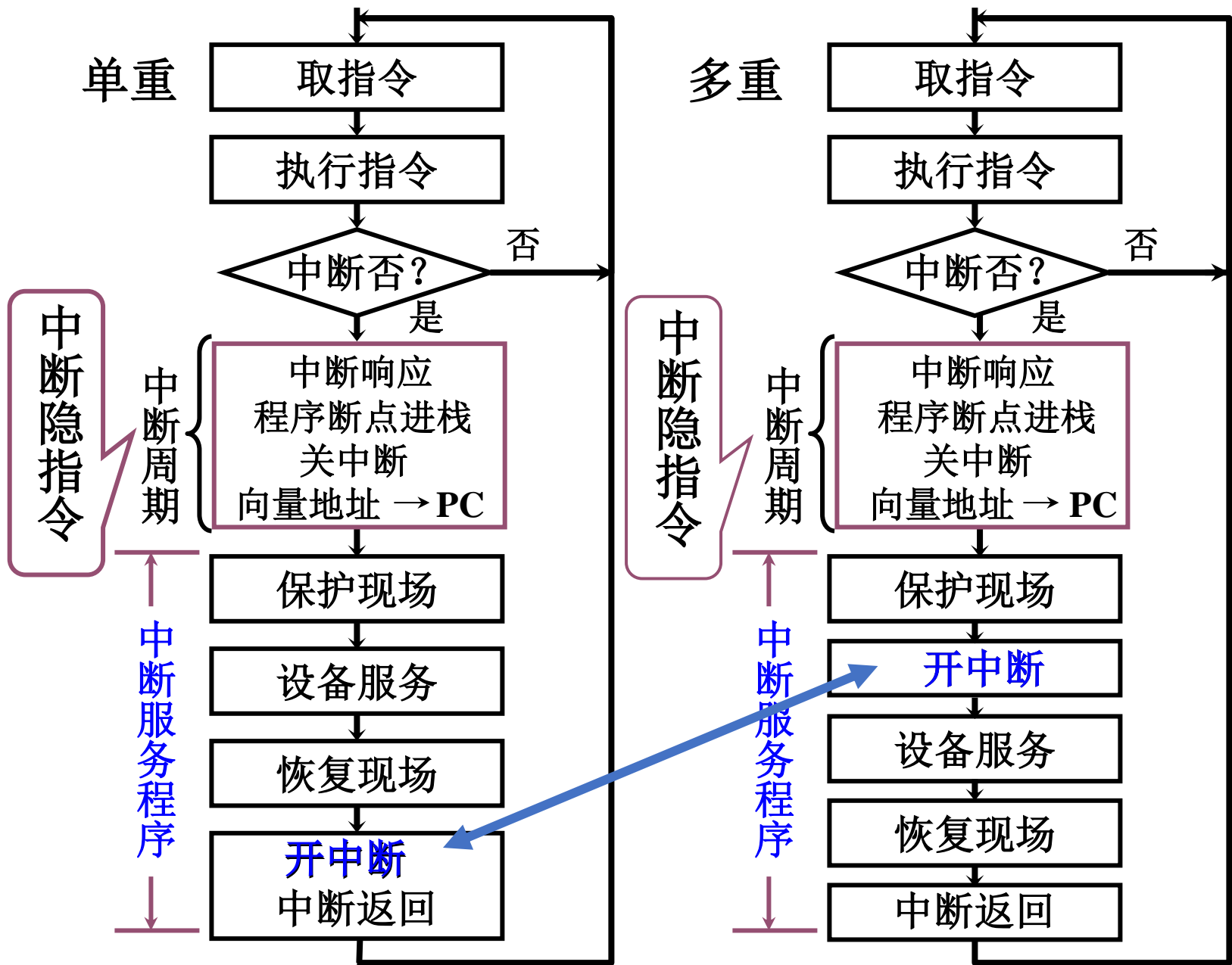
中断服务程序的最后一条指令，返回原程序的断点处，继续执行原程序

2. 单重中断和多重中断

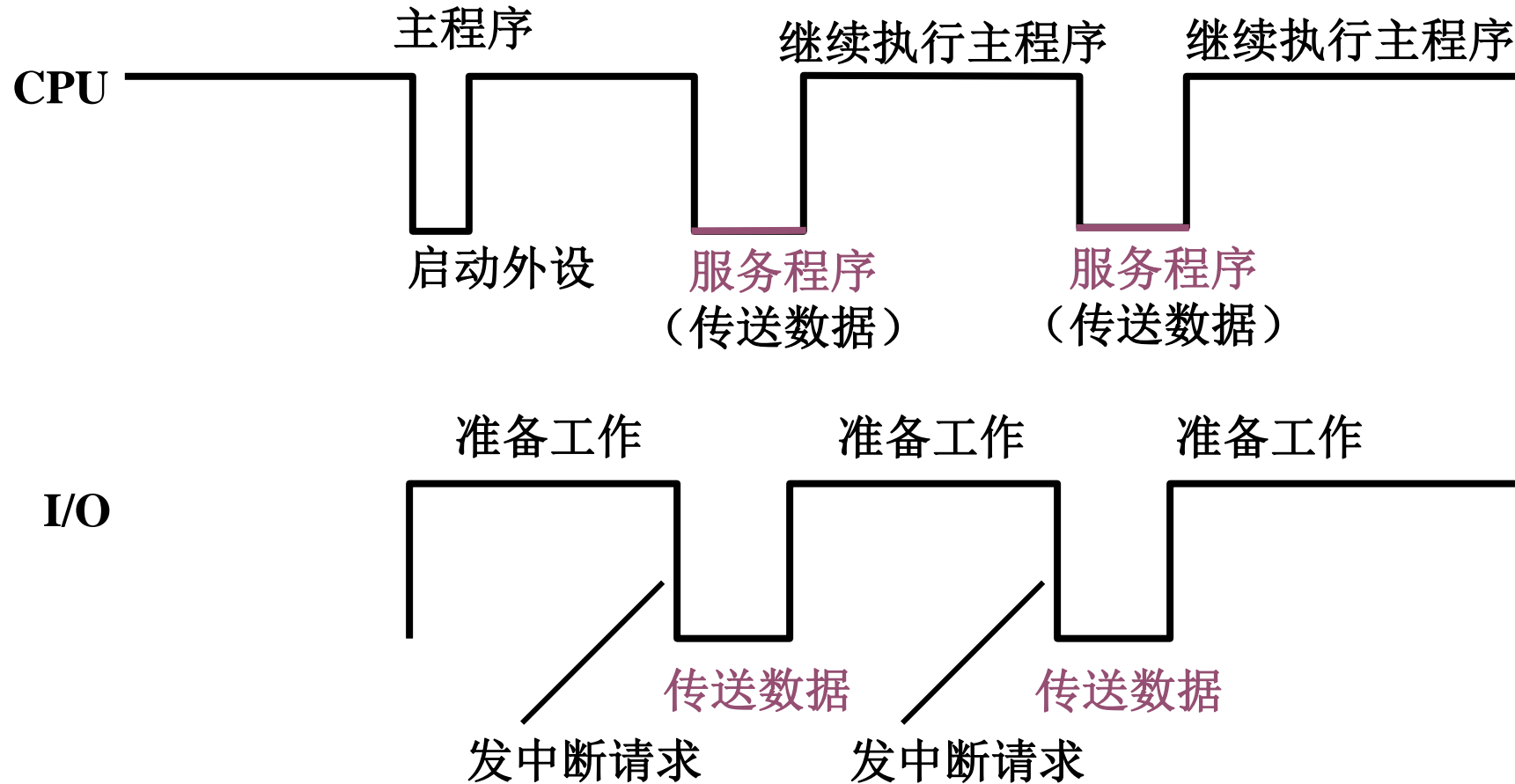
单重 中断 不允许中断 现行的 中断服务程序

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



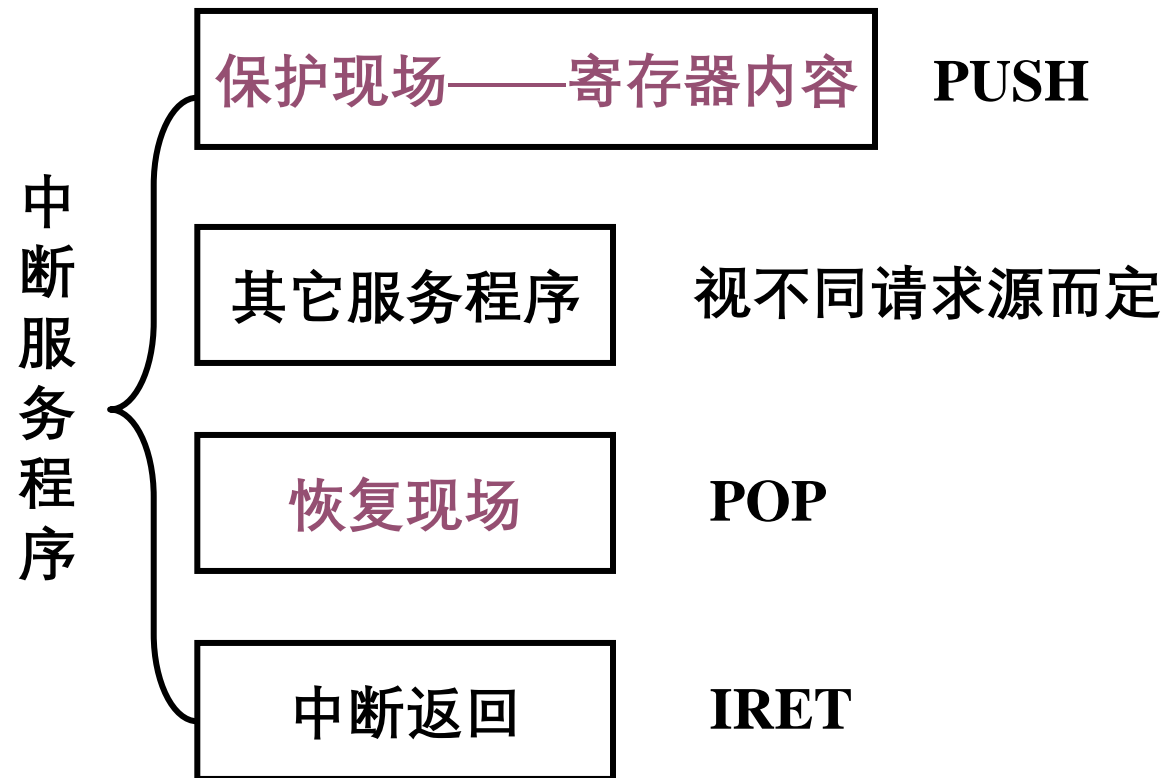
主程序和服务程序抢占 CPU 示意图



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

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



7.2.5 程序中中断方式

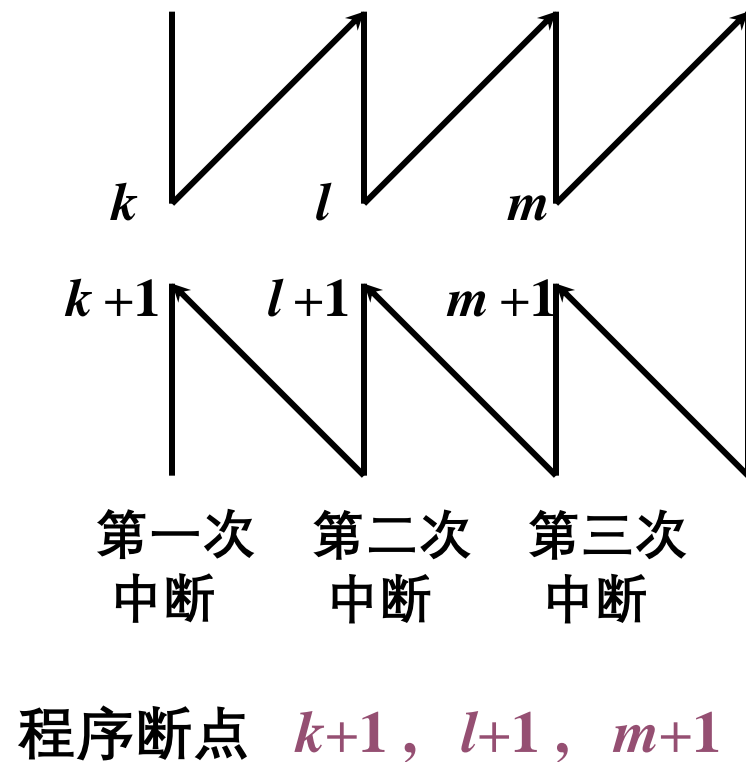
- 一. 中断的概念
- 二. 中断的产生
- 三. 程序中中断方式的接口电路
- 四. 中断处理过程(中断响应)
- 五. 中断服务程序流程 (CPU)
- 六. 中断屏蔽技术 (CPU)

六、中断屏蔽技术

1. 多重中断的概念

多重中断又称“中断嵌套”：

- CPU正在执行某个中断服务程序时，另一个中断源又提出了新的中断请求，而CPU又响应了这个新的请求，暂时停止正在运行的服务程序，转去执行新的中断服务程序。

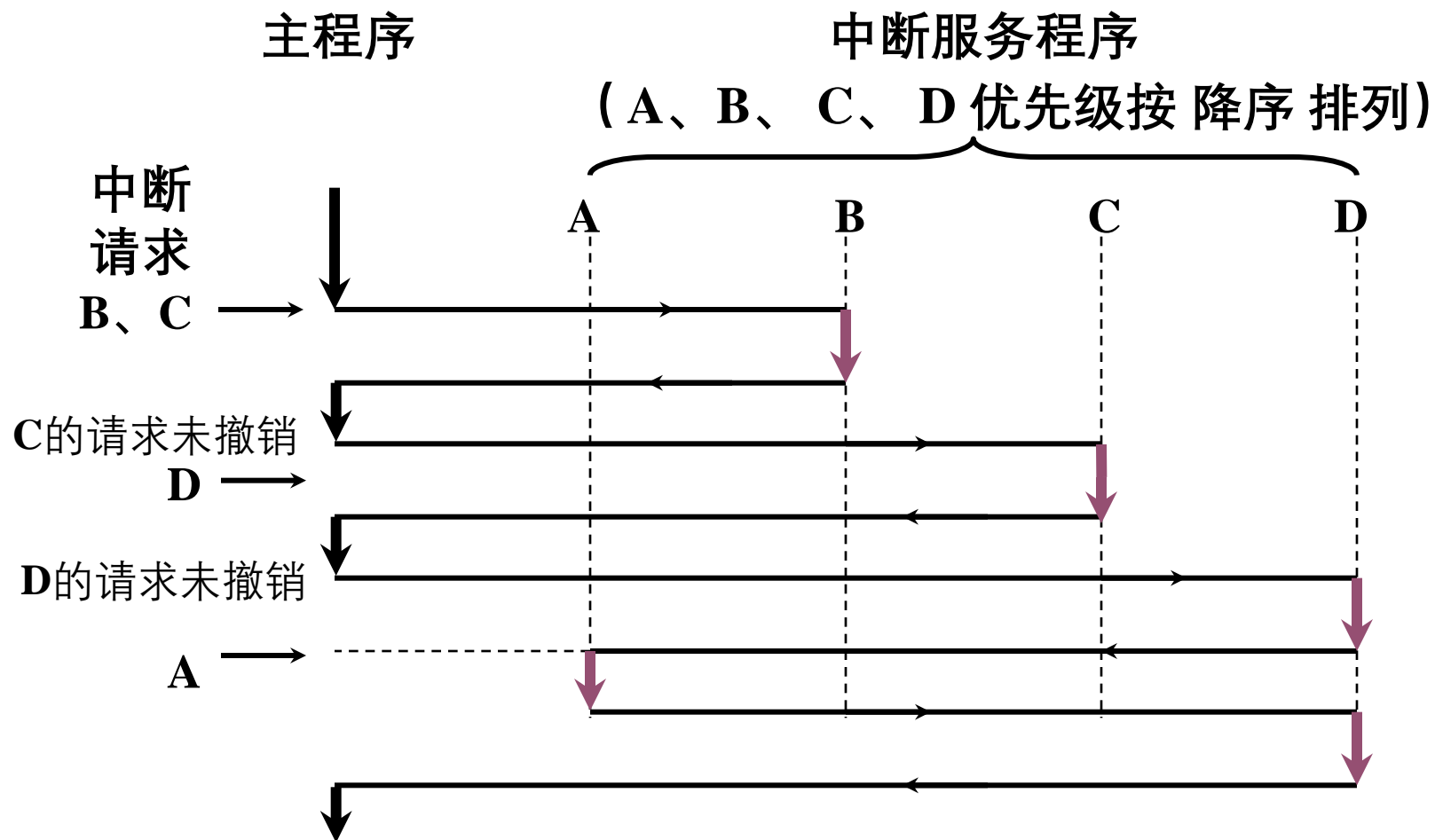


2. 实现多重中断的条件

要允许CPU在执行某个中断服务程序时，
响应新的中断请求

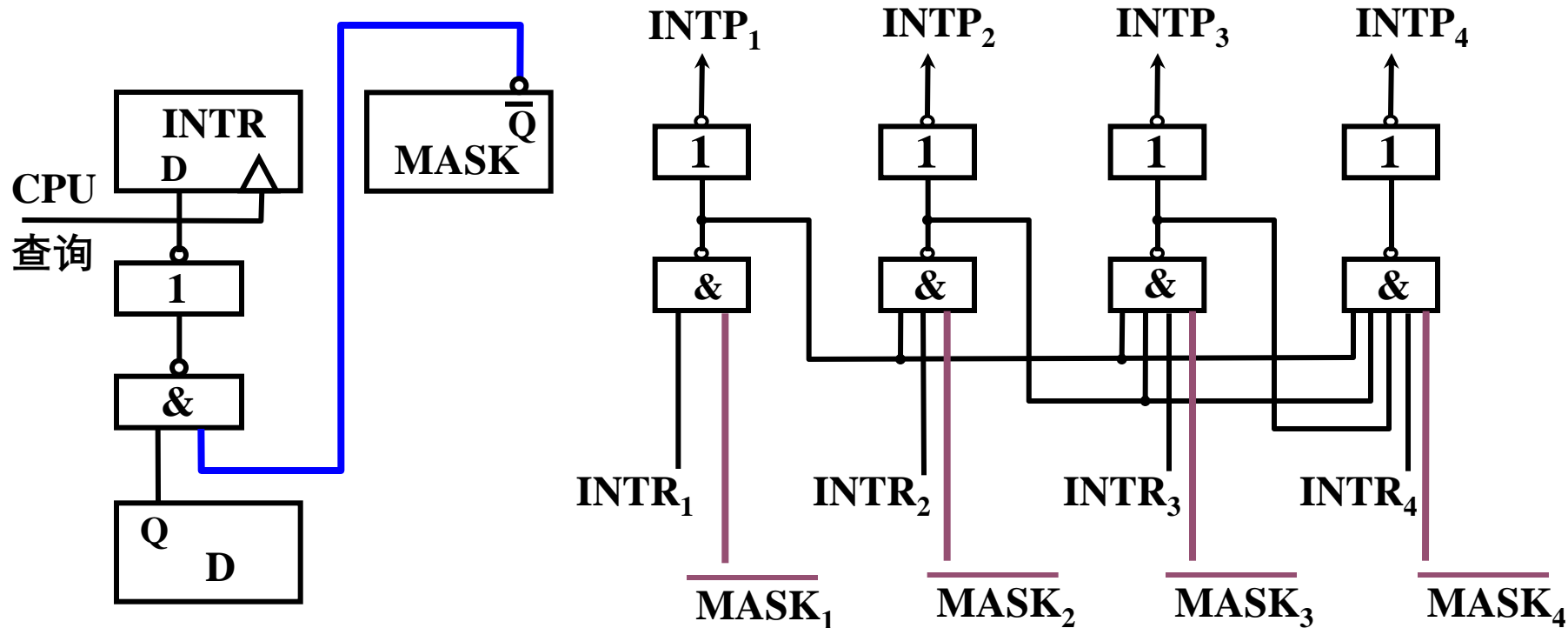
(1) 提前 设置 开中断 指令

(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)

$INTP_i = 0$ (不能被排队选中)

(2) 屏蔽字

每个中断请求触发器都有一个中断屏蔽触发器，所有屏蔽触发器组合在一起便构成了一个屏蔽寄存器，屏蔽寄存器的内容称为屏蔽字。

16个中断源 1, 2, 3, ..., 16 按降序排列

屏蔽字对应中断源的优先级别

优先级	屏蔽字
1	1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
2	0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
3	0 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1
4	0 0 0 1 1 1 1 1 1 1 1 1 1 1 1 1
5	0 0 0 0 1 1 1 1 1 1 1 1 1 1 1 1
6	0 0 0 0 0 1 1 1 1 1 1 1 1 1 1 1
⋮	⋮
15	0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1
16	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1

(3) 屏蔽技术可改变**处理**优先等级

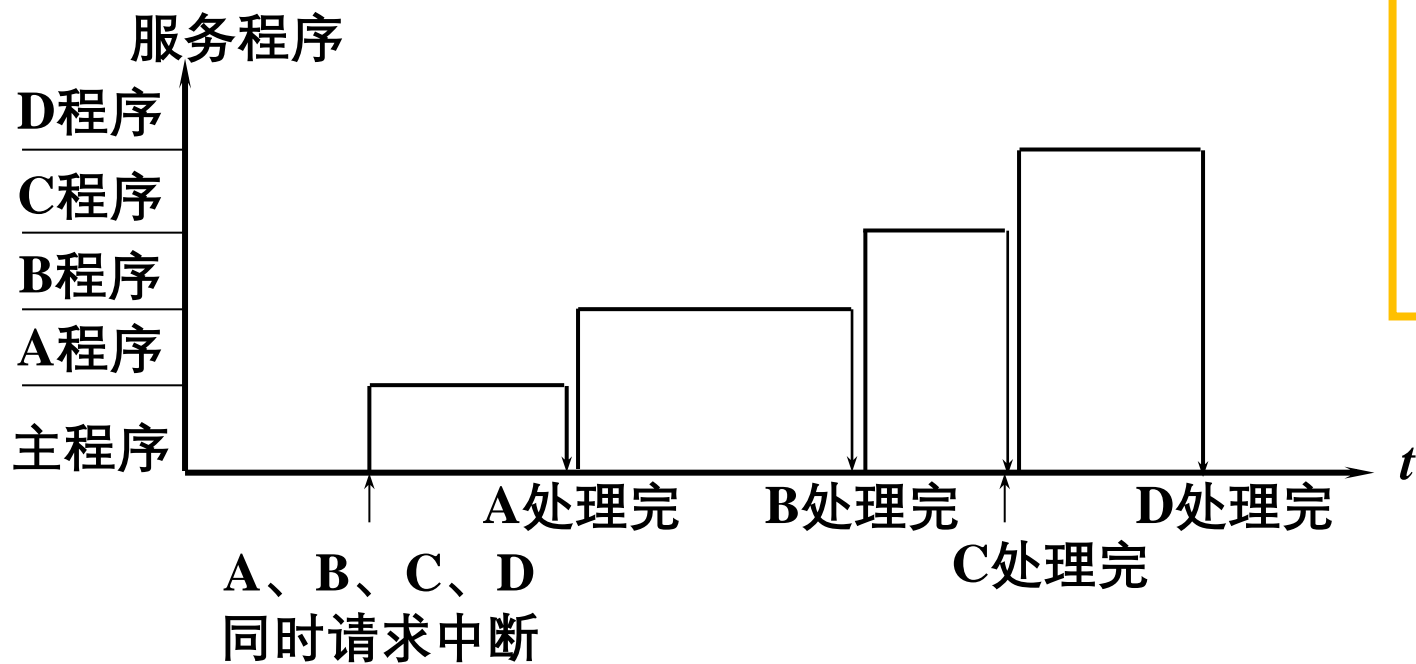
{	响应优先级	CPU响应各中断源请求的优先次序，一般是硬件电路已经设置好，不可改变
	处理优先级	CPU实际对各中断源请求的处理优先次序。可改变（通过重新设置屏蔽字）

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



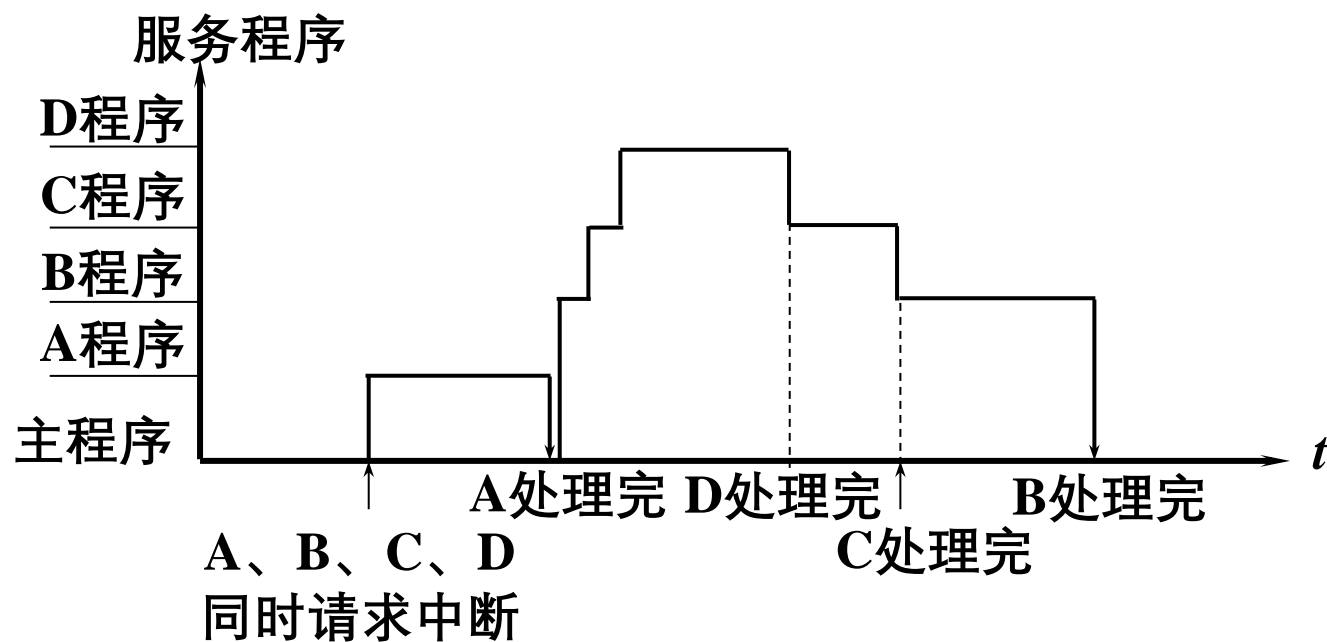
CPU 执行程序轨迹 (原屏蔽字)

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 (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

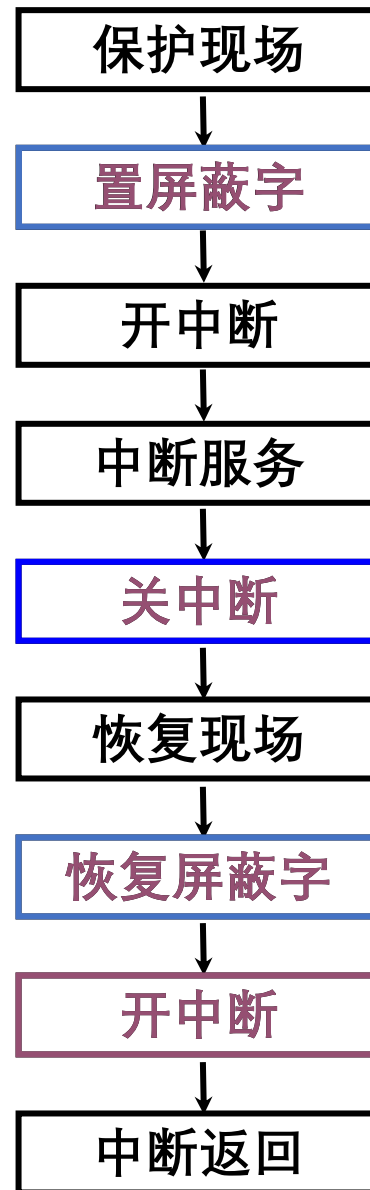
中断源	原屏蔽字	新屏蔽字
A	1 1 1 1	1 1 1 1
B	0 1 1 1	0 1 0 0
C	0 0 1 1	0 1 1 0
D	0 0 0 1	0 1 1 1

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

(5) 新屏蔽字的设置

中断服务程序中设置新的屏蔽字



防止恢复现场过程中出现新的中断

7.2 输入输出系统

7.2.1 概述

7.2.2 外部设备

7.2.3 I/O接口

7.2.4 程序查询方式

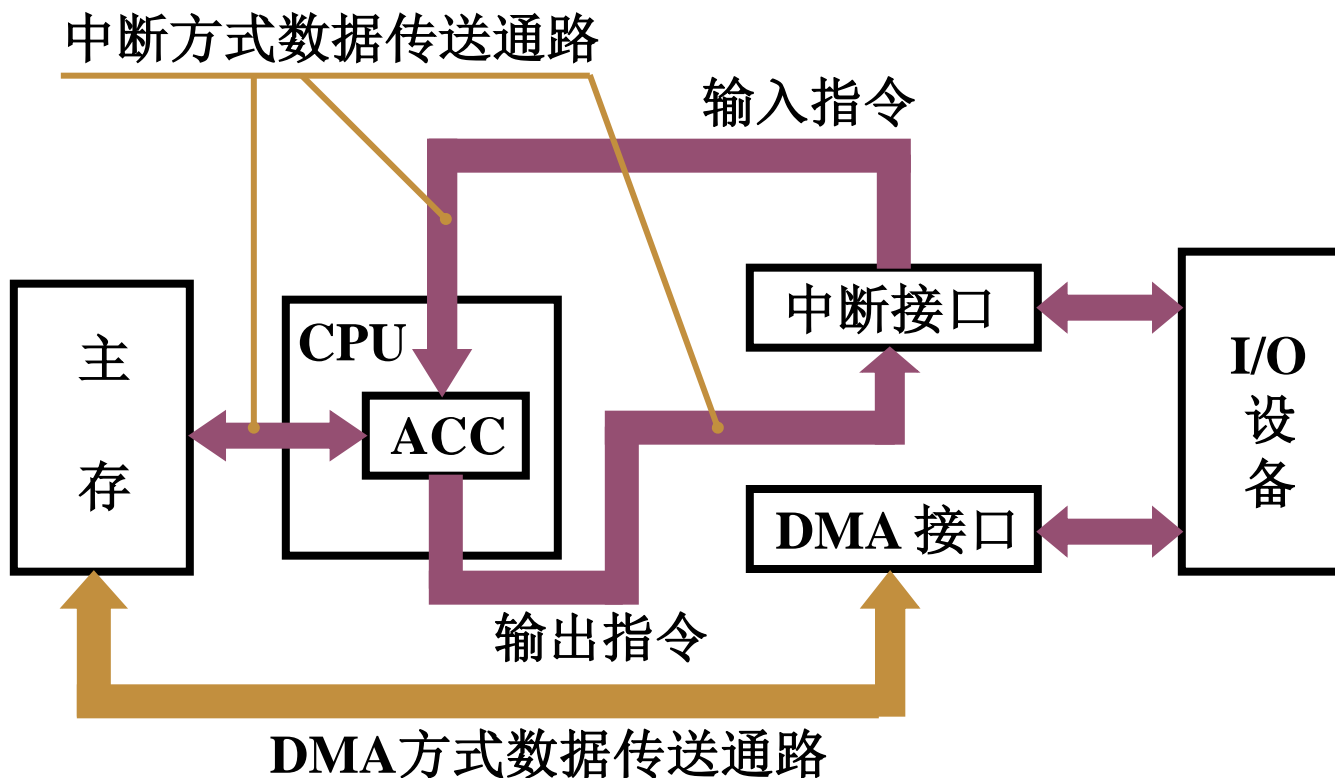
7.2.5 程序中断方式

7.2.6 DMA方式

7.2.6 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



• 中断方式

- 传送一个数据执行一次中断服务子程序（几十条指令）
- 频繁占用CPU，效率低下
- 不适合于高速传输的系统

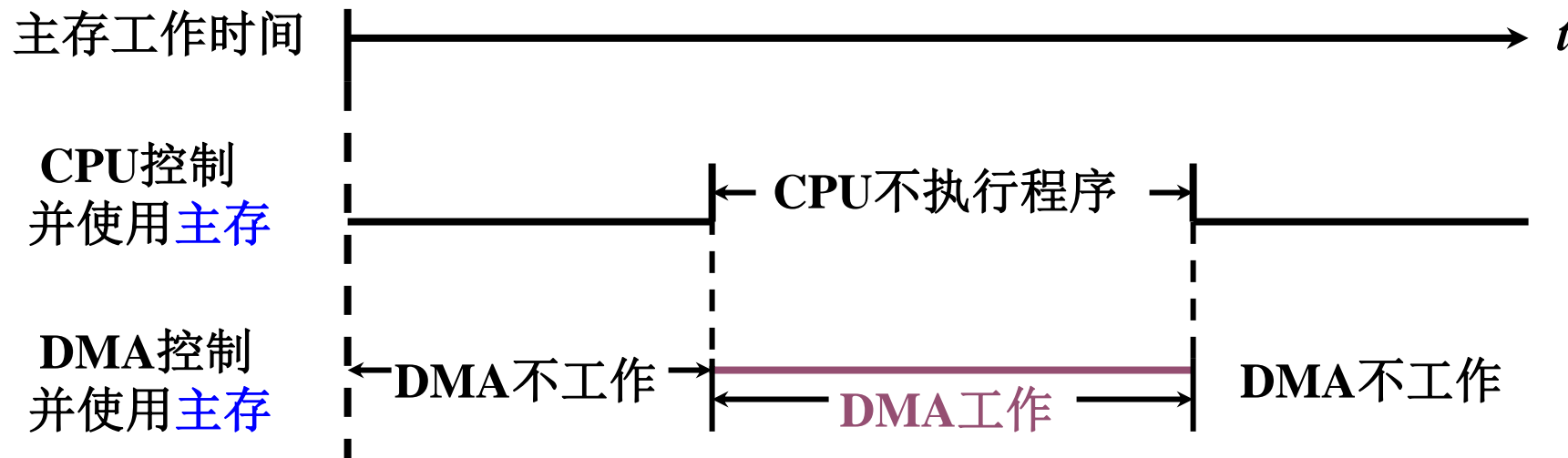
• DMA方式

- 外设与主存间建立一个由硬件管理的数据通路
- CPU不介入外设与主存的数据传送操作
- 减少CPU开销，提升效率
- 适合成组交换数据的场合

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

- 当外设要求传送一批数据时，DMA接口向CPU发送一个**停止信号**，要求CPU放弃地址、数据和相关控制线的使用权；数据传送结束后，DMA接口**通知**CPU可以使用主存
- 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

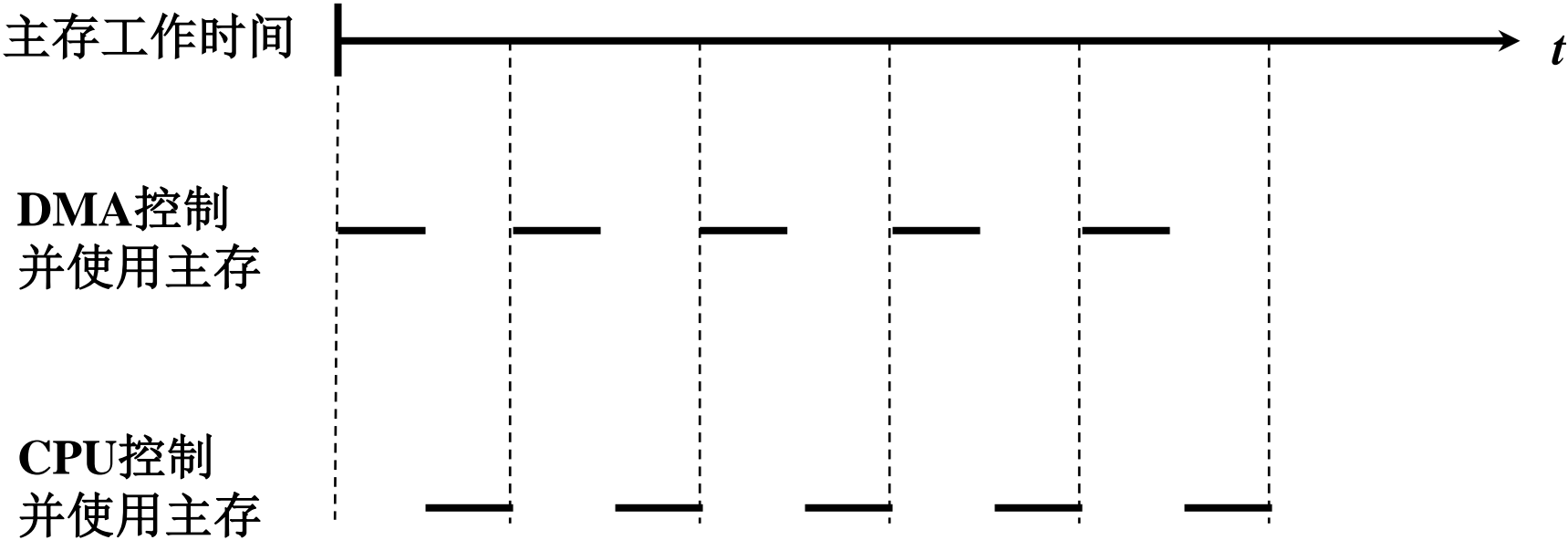


(2) DMA 与 CPU 交替访问

适合于CPU的工作周期比主存取取周期长的情况

CPU 工作周期 { C₁ 专供 DMA 访存
C₂ 专供 CPU 访存

所有指令执行过程中的一个基准时间



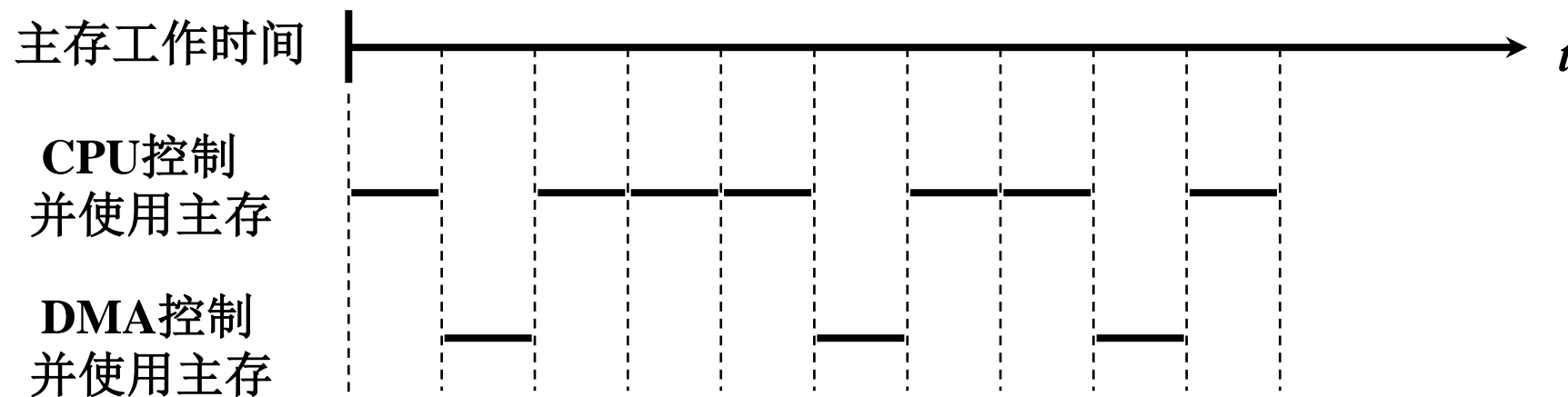
- CPU和DMA接口各自有独立的访存地址寄存器、数据寄存器、读/写信号。
- 总线控制权的转移几乎不需要时间，DMA传送速率高
- 硬件逻辑比较复杂

不需要 申请建立和归还 总线的使用权。（总线使用权由C₁和C₂控制）

(3) 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 等待CPU存取周期结束，让出总线
- CPU 与 DMA 同时请求访存
此时 CPU 将总线控制权让给 DMA



7.2.6 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

避免因进入DMA工作而影响CPU正常活动
或者引起总线竞争

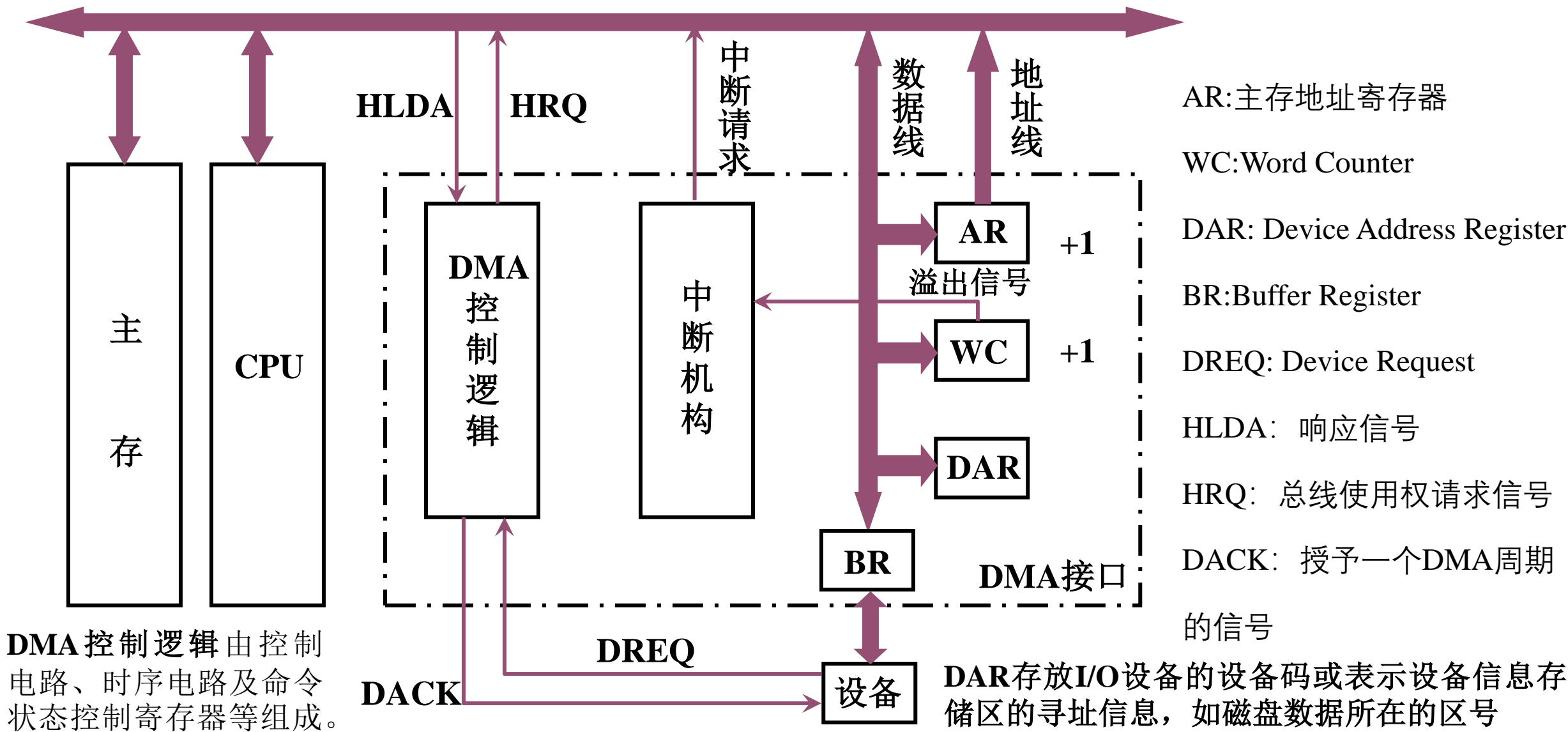
(3) 管理 系统总线、控制 数据传送

(4) 确定 数据传送的 首地址和长度

更新 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



7.2.6 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(主机发送给DMA接口)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

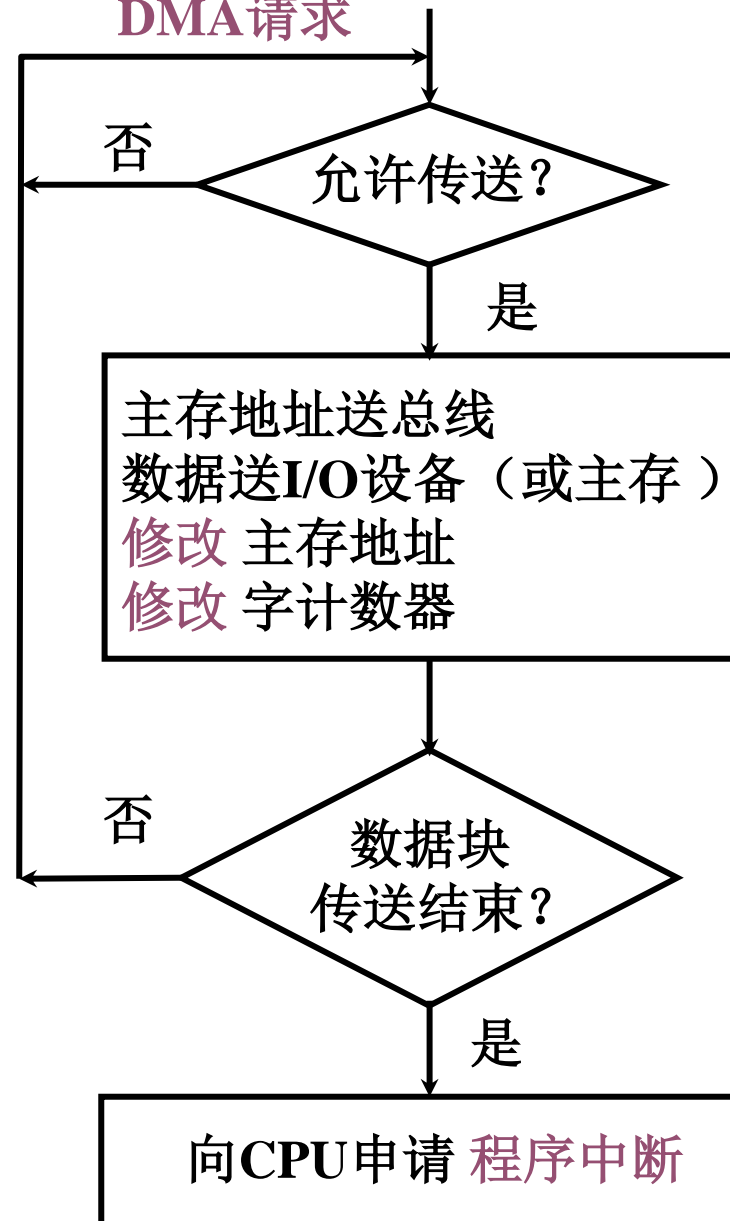
后处理:

中断服务程序
做 DMA 结束处理

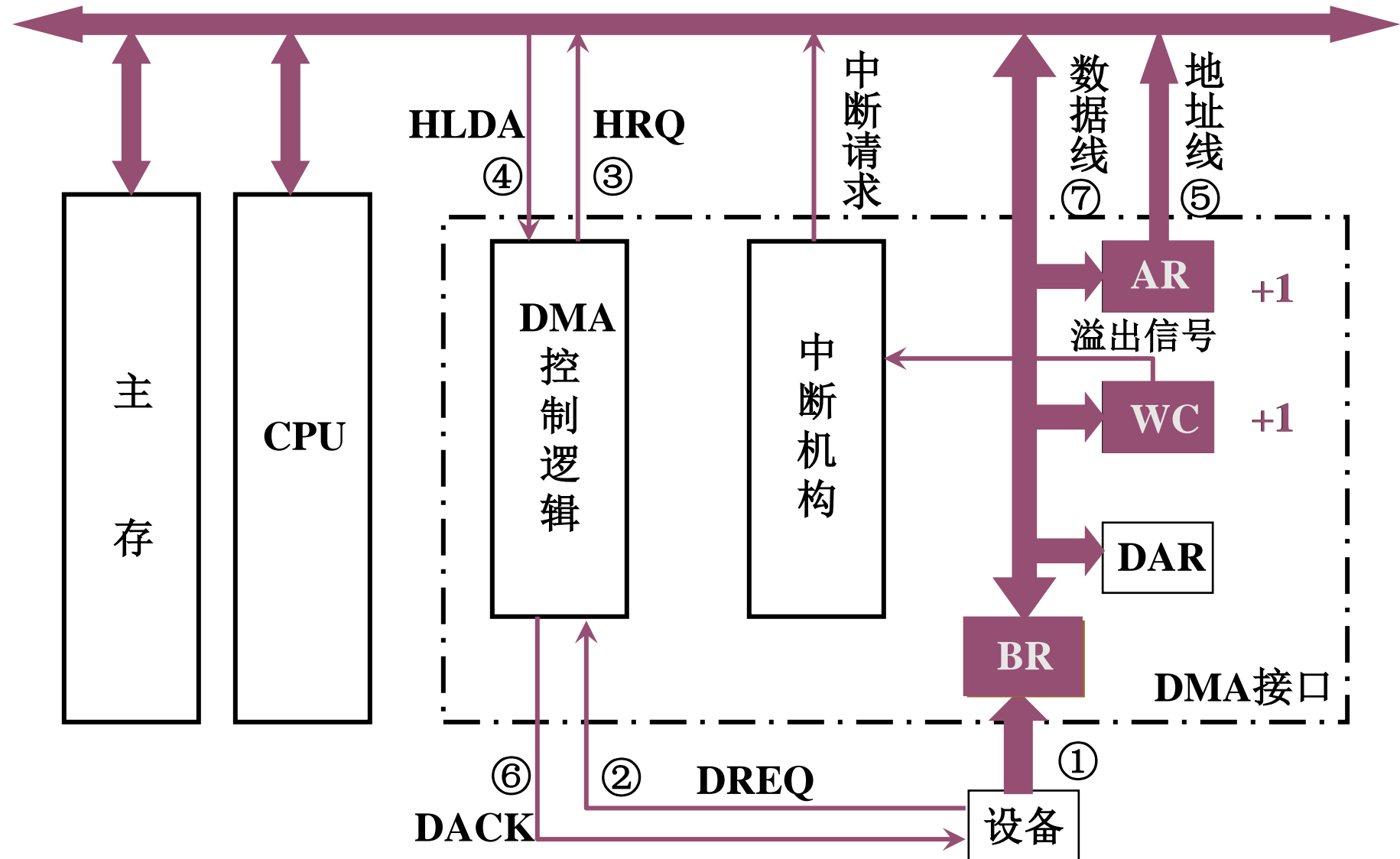
继续执行主程序

数据传送

DMA 请求

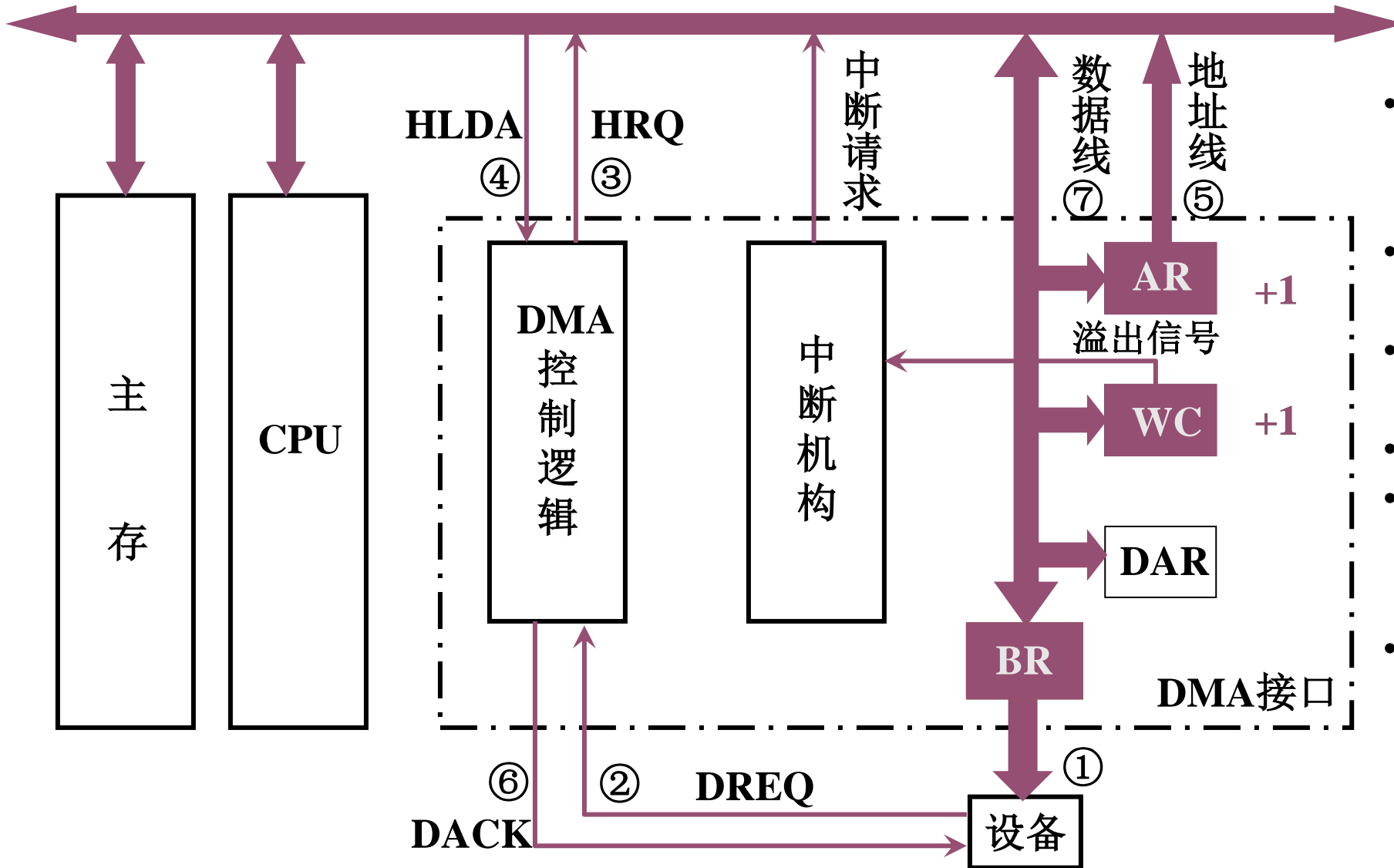


(3) 数据传送过程（输入）



- 设备准备好一个数据字（一个字传送结束）；
- 向DMA接口提出申请（DREQ）；
- DMA控制逻辑向CPU请求DMA服务，发出HRQ；
- 待收到CPU发出的HLDA；
- DMA控制逻辑开始管理DMA传送全过程，包括对AR和WC的修改、指定传送类型（入/出）、通知设备DACK信号。

(4) 数据传送过程（输出）



- 当BR已将输出数据送至I/O设备后，表示BR已“空”；
- 设备向DMA接口发请求（DREQ）；
- DMA接口向CPU申请总线控制权（HRQ）；
- CPU发回HLDA信号；
- 将DMA的AR中的主存地址送地址总线，并命令存储器读；
- 通知设备已被授予一个DMA周期（DACK），并为交换下一个字做准备

(5) 后处理

当DMA的中断请求得到响应后，CPU停止执行原程序，转去执行中断服务程序，做一些DMA的结束工作，包括：

- 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占用处理器多少时间？

(2) 用中断方式，需占处理器多少时间？

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

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

若磁盘不断进行传输，每秒 所需 DMA 辅助操作的时钟周期数

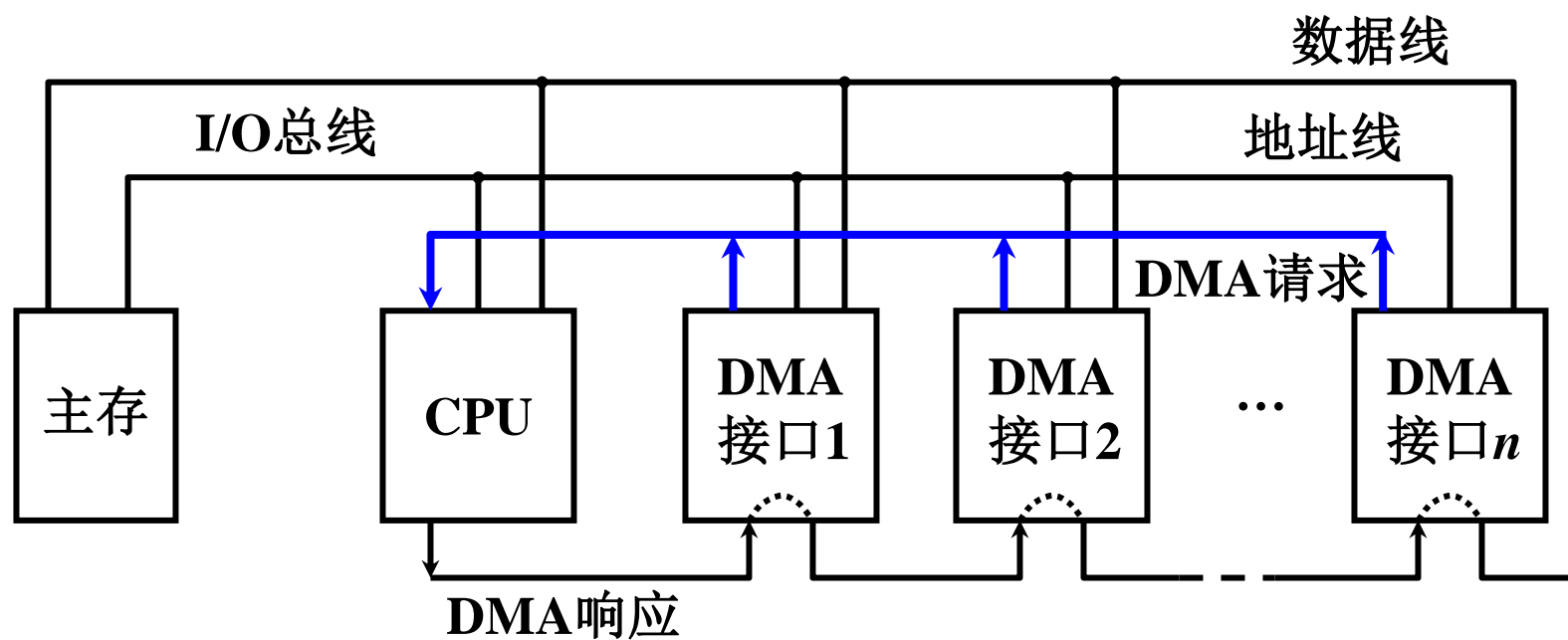
$$(1000 + 500) / 0.002 = 750000$$

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

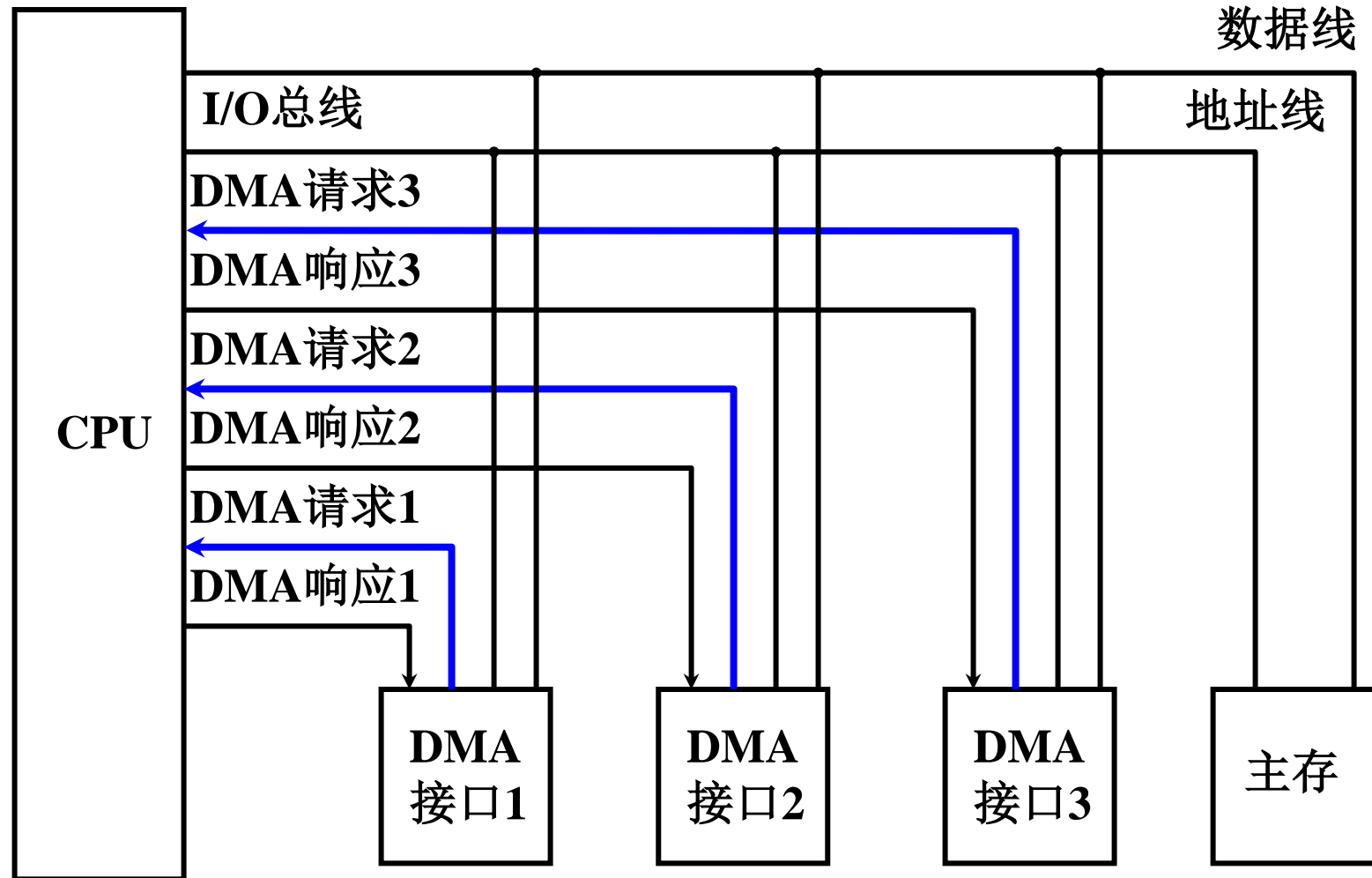
(1) 具有公共请求线的 DMA 请求



DMA 响应采用链式查询的方式

(2) 独立的 DMA 请求

由CPU的优先级判别机构裁决



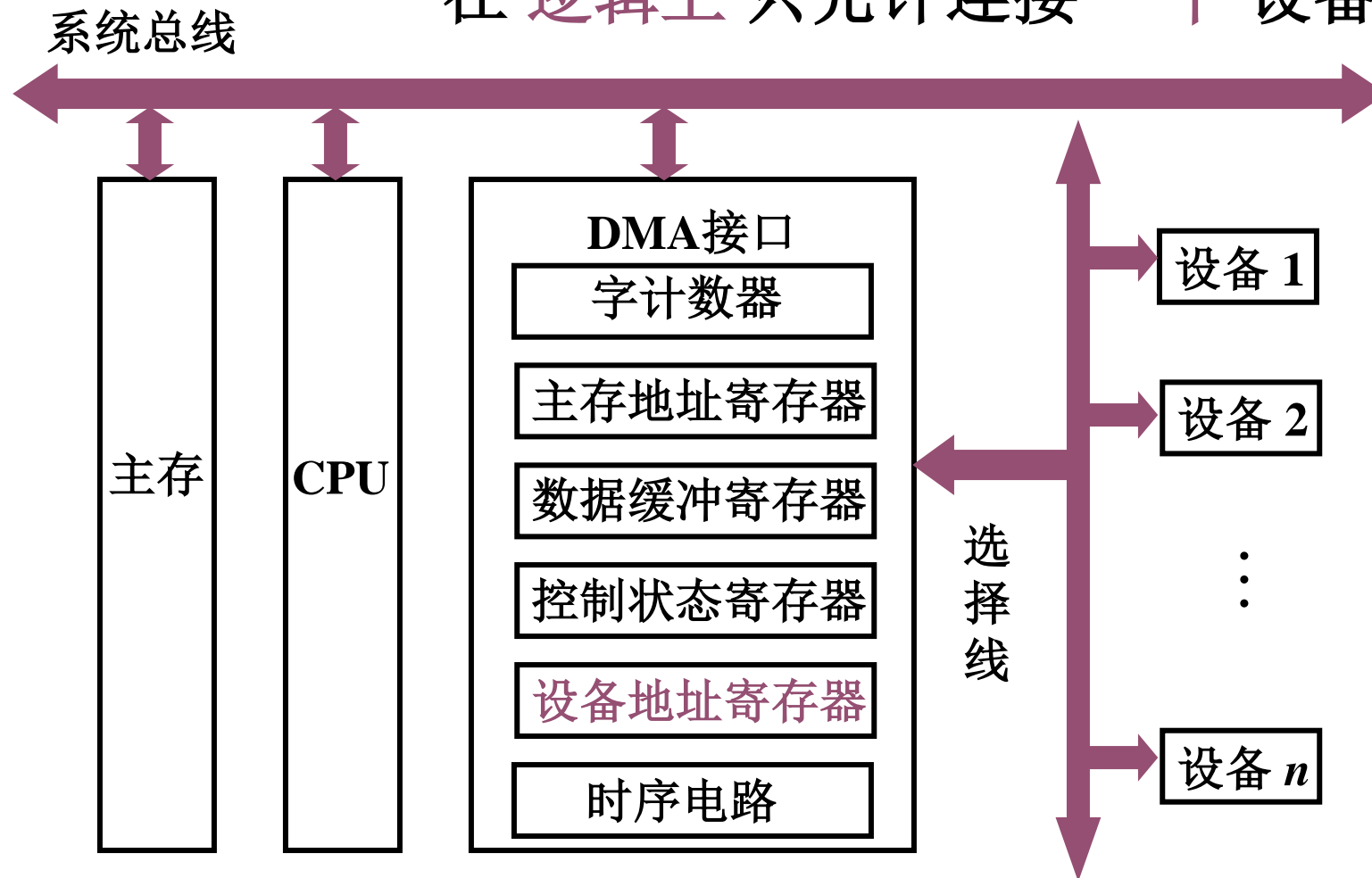
3. DMA 方式与程序中断方式的比较

	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

在物理上可连接多个设备
在逻辑上只允许连接一个设备

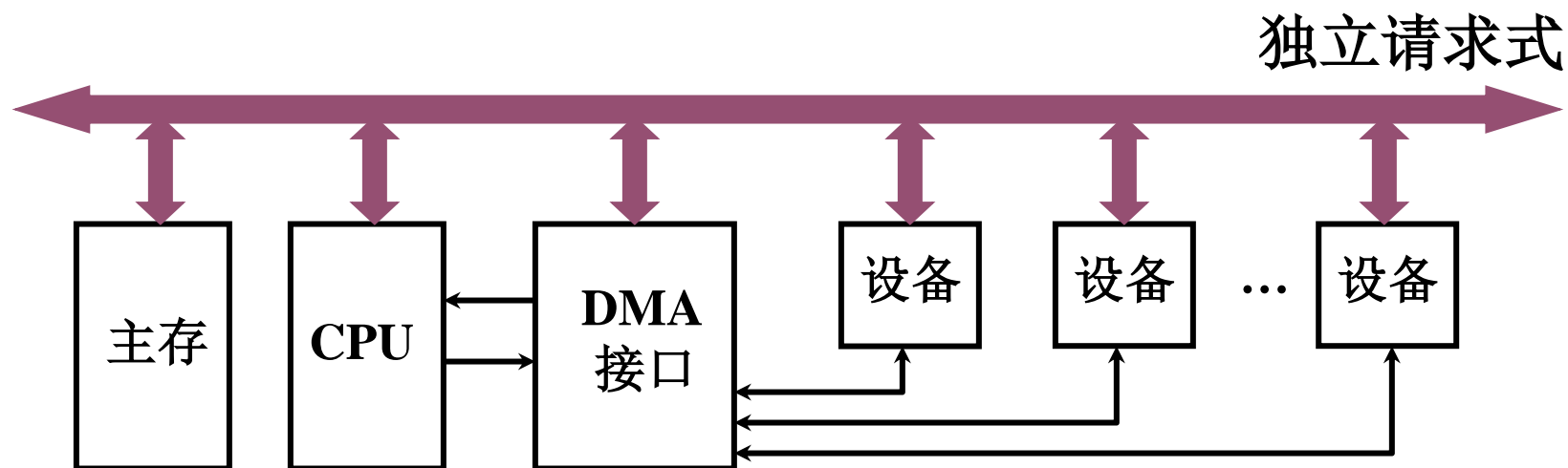
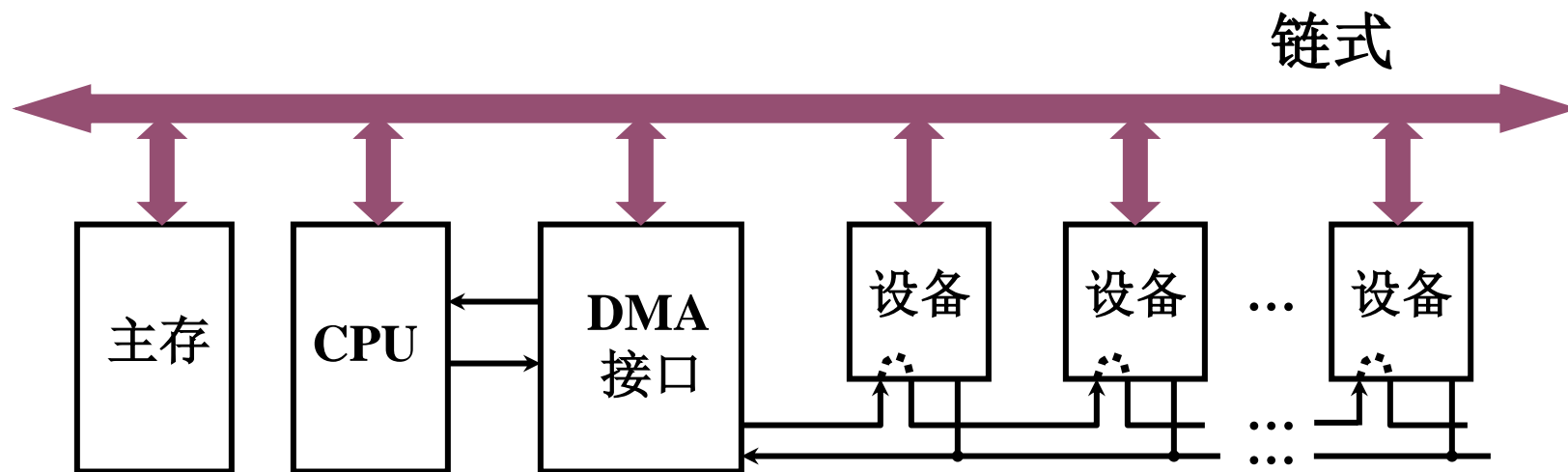


在预处理时，将所选设备的设备号送入设备地址寄存器

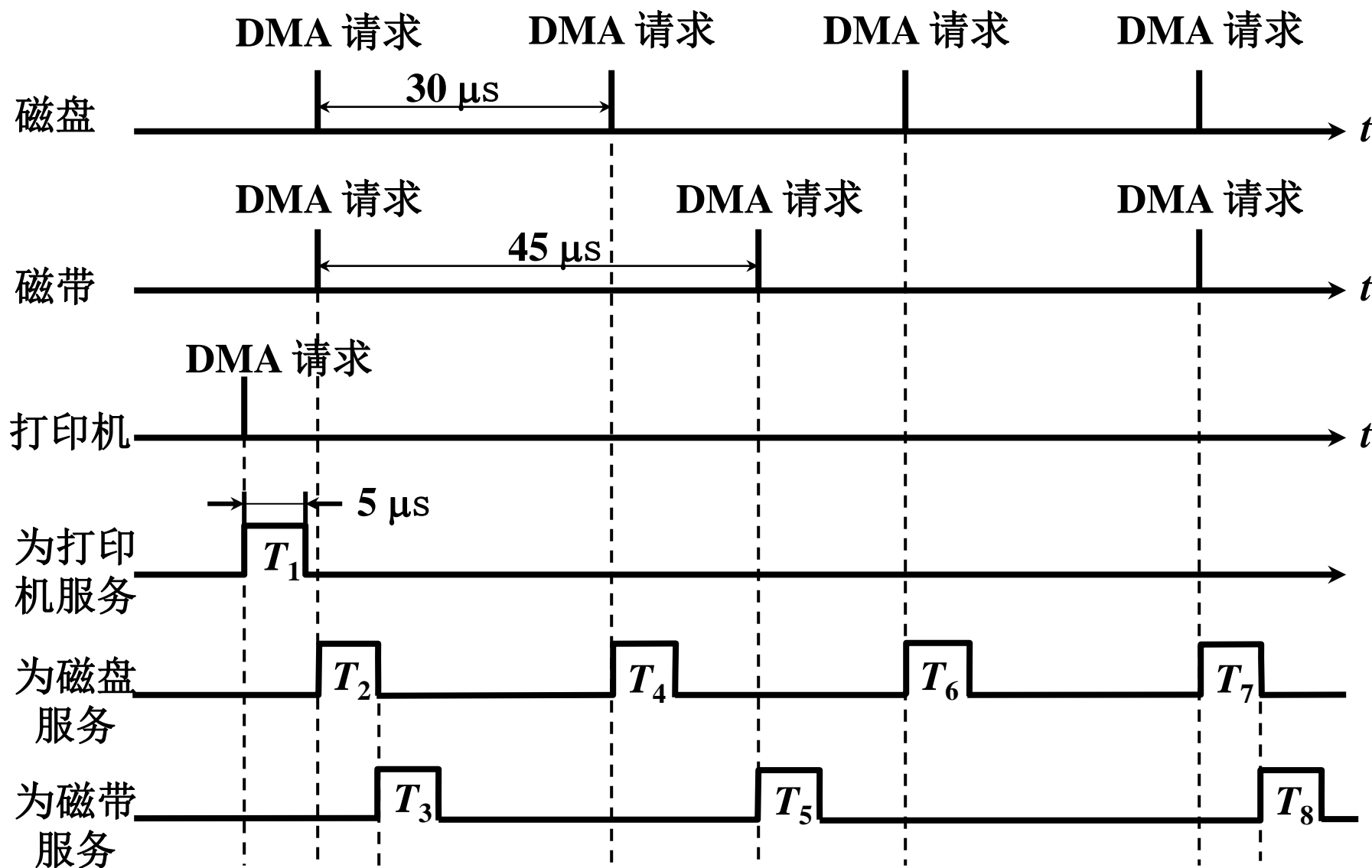
2. 多路型

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



- 磁盘、磁带、打印机同时工作
- 磁盘、磁带、打印机分别每隔 $30\ \mu\text{s}$ 、 $45\ \mu\text{s}$ 、 $150\ \mu\text{s}$ 向 DMA 接口发送 DMA 请求
- 优先级：磁盘 $>$ 磁带 $>$ 打印机
- 假设 DMA 接口完成一次 DMA 数据传送需要 $5\ \mu\text{s}$
- 打印机先发请求，故 DMA 接口首先为打印机服务 T_1 ；
- 接着磁盘、磁带同时发请求，根据优先级，先响应磁盘请求 T_2
- 再响应磁带请求 T_3

对本章哪部分内容有疑问？

A

IO接口

B

程序查询方式

C

程序中断方式

D

DMA方式

E

无

提交

下列事件中，属于外部中断事件的是（）

A

访存时缺页

B

定时器到时

C

网络数据包到达

提交