

[计算机组成原理]

第五章 输入输出系统



大纲

(一) I/O系统基本概念

(二) 外部设备

1. 输入设备：键盘、鼠标
2. 输出设备：显示器、打印机
3. 外存储器：硬盘存储器、磁盘阵列、光盘存储器

(三) I/O接口（I/O控制器）

1. I/O接口的功能和基本结构
2. I/O端口及其编址

(四) I/O方式

1. 程序查询方式
2. 程序中断方式

中断的基本概念；中断响应过程；中断处理过程；多重中断和中断屏蔽的概念。

3. DMA方式

DMA控制器的组成；DMA传送过程。

4. 通道方式

第五章 输入输出系统

5.1 节	概 述
5.2 节	外 部 设 备
5.3 节	I/O 接 口
5.4 节	程序查询方式
5.5 节	程序中中断方式
5.6 节	DMA方式

5.1 概述

一、输入输出系统的发展概况

1. 早期 分散连接 （程序查询方式）

CPU 和 I/O 串行 工作，I/O通过CPU与主机信息交换。

2. 接口模块和 DMA 阶段 总线连接 { 中断方式 DMA 方式

CPU 和 I/O 并行 工作，I/O通过接口与主机信息交换。

5.1 概 述

一、输入输出系统的发展概况

3. 具有通道结构的阶段

通道：一种专门用于输入输出操作的部件，是一种具有特殊功能的处理器。CPU启动通道后，可以继续运行自己的程序，而通道则同时执行通道程序，控制I/O设备进行直接数据传送。

4. 具有 I/O 处理机的阶段

I/O处理机 (IOP) 方式比通道方式具有更强的独立性和专用性。IOP有自己的指令系统，可以通过对IOP的编程，实现独立于CPU的I/O操作。



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
-----	-----	-----

操作码：I/O指令与其它指令的判别码

命令码：I/O设备具体操作

设备码：用于从多台I/O设备中选择某台设备

(2) 通道指令 通道自身的指令

指出数据组的首地址、传送字数或末地址、操作命令

如 IBM/370 通道指令为 64 位

二、输入输出系统的组成

5.1

1. I/O 软件

2. I/O 硬件

设备 I/O 接口 (总线型)

设备 设备控制器 通道 (通道型)

三、I/O 与主机的联系方式

5.1

1. I/O 编址方式

(1) 统一编址:指把I/O地址 (I/O端口地址) 当作存储器的单元进行分配。

在这种方式下CPU不需设置专门的I/O指令, 用统一的访问存储器的取数、存数指令就可访问I/O端口。

优点: 使CPU访问I/O的操作更灵活、更方便, 此外还可使端口有较大的编址空间。

缺点: 1) 使端口占用了存储器地址, 使内存容量变小。

2) 利用存储器编址的I/O设备进行数据输入输出操作执行速度变慢。

(2) 不统一编址

三、I/O 与主机的联系方式

5.1

1. I/O 编址方式

- (1) 统一编址:指把I/O地址 (I/O端口地址) 当作存储器的单元进行分配。
- (2) 不统一编址 (单独编址方式) : 指I/O端口地址与存储器地址无关, 另行单独编址。

在这种方式下, CPU需要设置专门的I/O指令访问端口。

优点: 输入输出指令与存储器指令有明显区别, 程序编制清晰、便于理解。

缺点: 输入输出指令少, 一般只能对端口进行传送操作, 尤其需要CPU提供存储器的读/写、I/O设备的读/写两组控制信号, 增加控制的复杂性。

三、I/O 与主机的联系方式

5.1

2. 设备寻址

每台设备有一个设备号，I/O指令设备码字段指出设备号，用设备选择电路识别设备是否被选中

3. 传送方式

(1) 串行：在主机与接口之间完全按照并行方式实现数据传输，但在接口与I/O设备之间是按照串行方式实现数据传输。（设置具有移位功能的数据缓冲器，实现数据格式的串 - 并转换。）（适用于低速的串行I/O设备及信息远距离传送设备。）

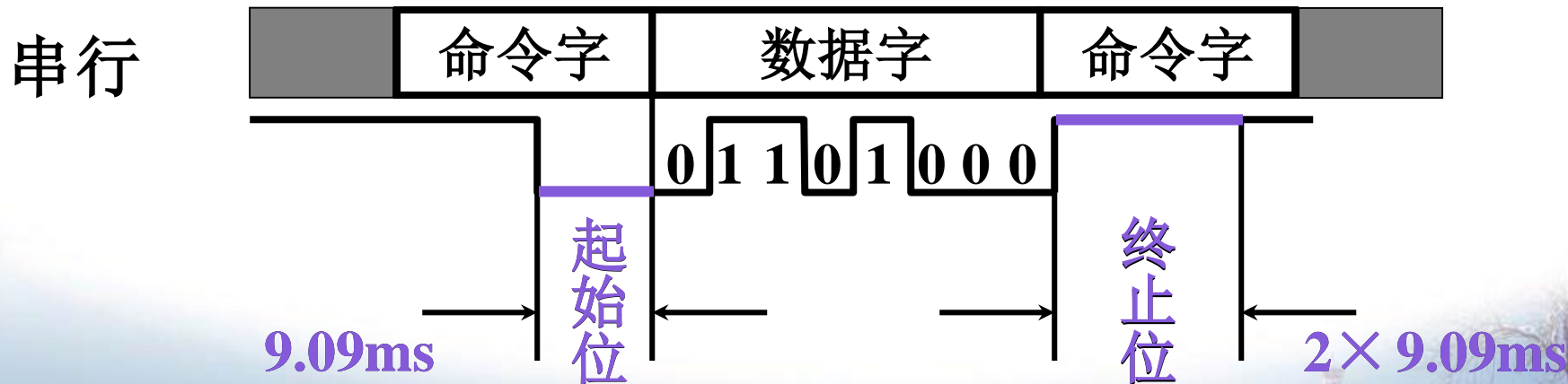
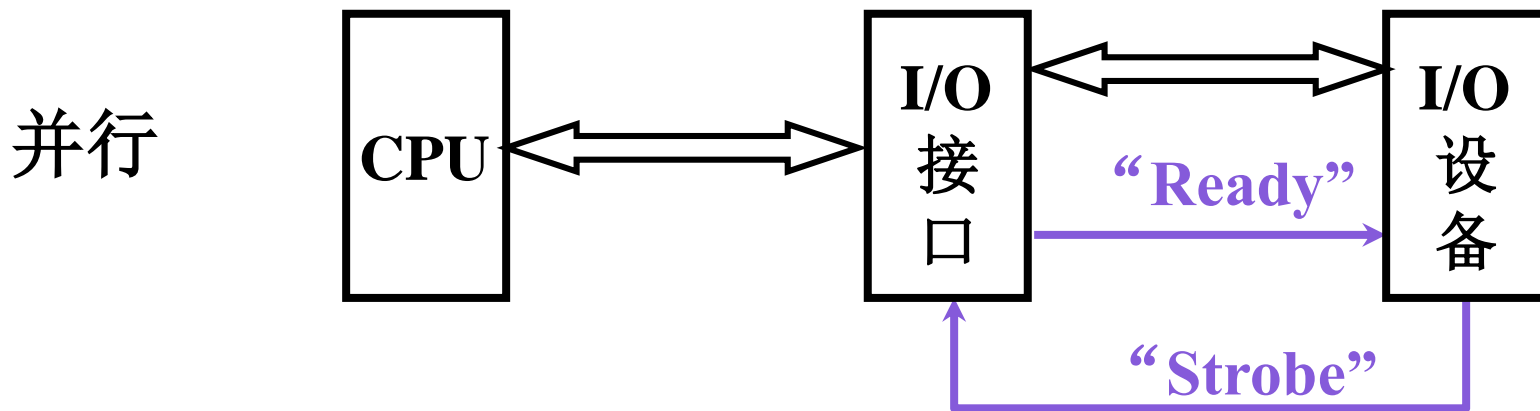
(2) 并行：在主机与接口、接口与I/O设备之间是按照并行方式完成信息传输。（当I/O设备本身按照并行方式工作，主机与I/O设备间距离较近，选用并口）

4. 联络方式

5.1

(1) 立即响应 (速度缓慢的I/O)

(2) 异步工作采用应答信号 (I/O设备与主机工作速度不匹配)

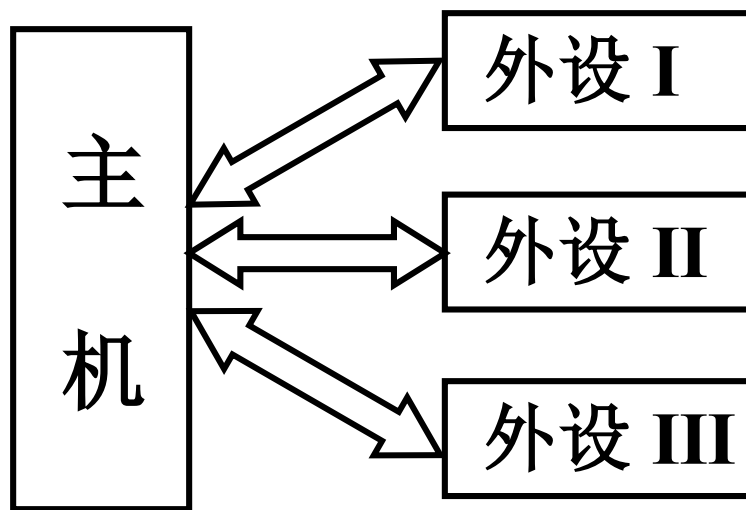


(3) 同步工作采用同步时标

5. I/O 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

5. I/O 与主机的连接方式

(1) 辐射式连接

(2) 总线连接

(3) 通道式连接

(4) 输入输出处理机方式连接

四、I/O 与主机信息传送的控制方式

5.1



1. 程序查询方式

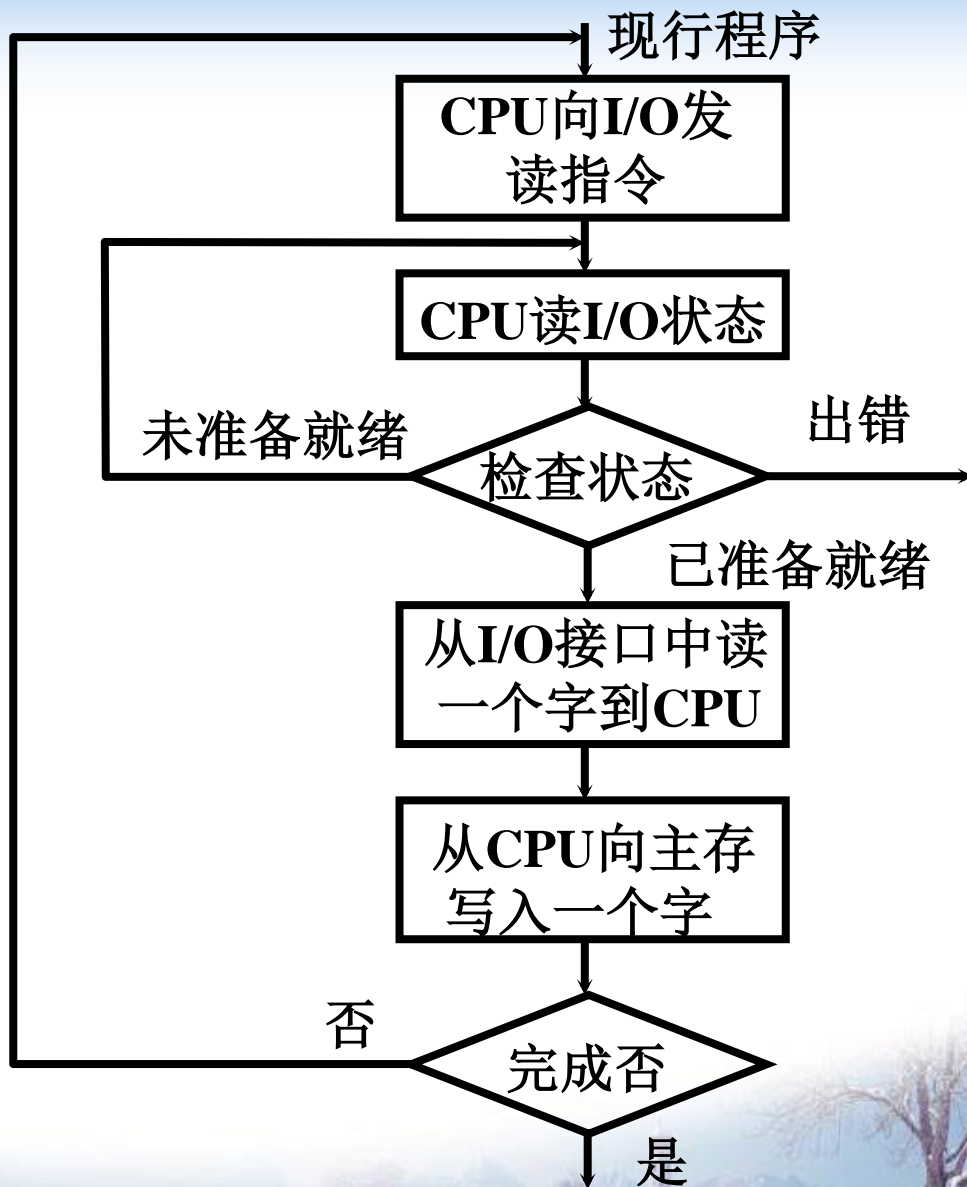
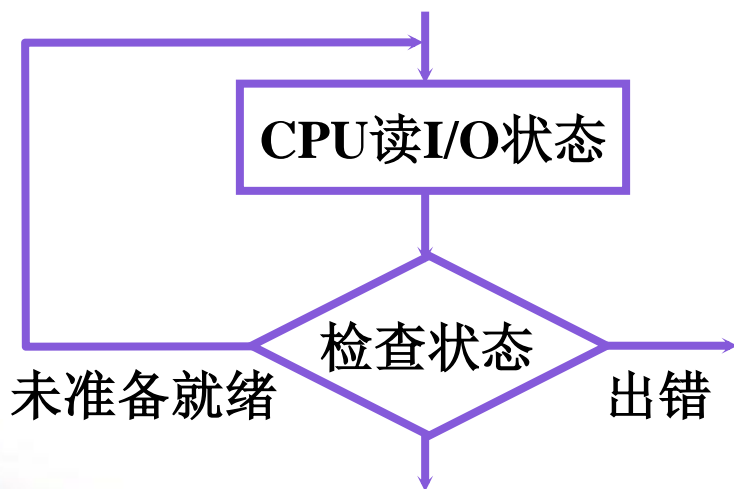
5.1

- 基本原理：用程序实现主机与外设之间的信息交换，即CPU直接利用I/O指令编程实现输入输出。
 - 通常方法是在用户程序中安排一段输入输出服务程序，当需要进行I/O操作时，直接控制I/O设备进行工作。
 - 开始时需要对设备所处的工作状态进行查询，所以称程序查询方式。
 - 程序每执行一条I/O指令（或传送指令），只能传送一个数据。每次都要查询设备所处的状态，只有当设备准备好后才能传送，否则主程序作循环等待。
 - CPU与I/O设备处于串行工作方式，CPU必须等待I/O设备完成操作后才能继续执行其它程序。因此，此方式数据的传送速度很慢，CPU的利用率较低。
 - Note：计算机带有多台I/O设备时，应根据各设备的重要程度对它们进行优先级排队。

1. 程序查询方式

CPU 和 I/O 串行工作

踏步等待



1. 程序查询方式

5.1

特点:

1. CPU能控制何时对何设备进行输入输出操作
2. 外设和CPU处于异步工作状态
3. 数据输入输出要经过CPU
4. 用于连接低速外设，如终端和打印机
5. 灵活性好
6. 不能实现CPU与外设的并行工作

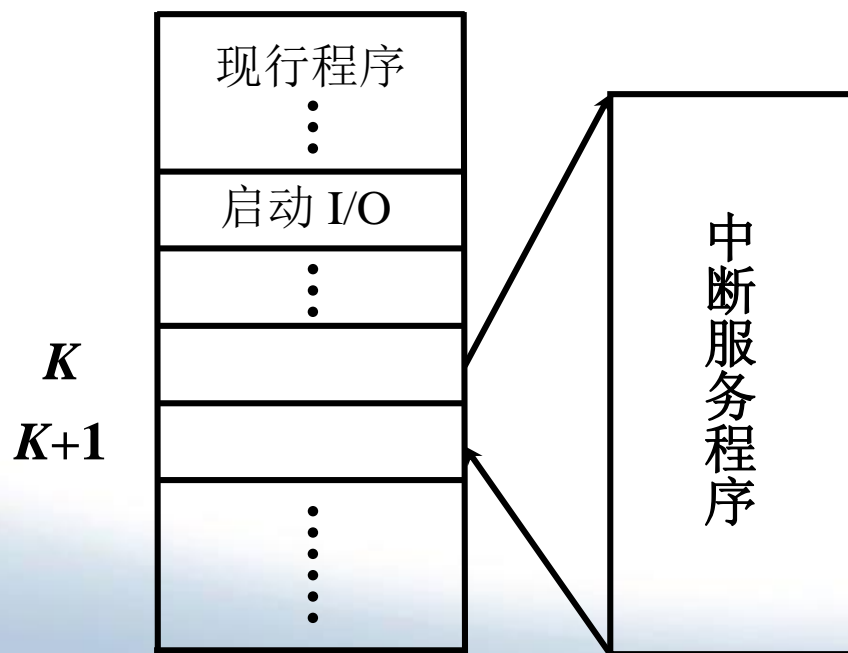
2. 程序中断方式

- 中断：CPU在执行程序的过程中，当出现异常情况或特殊请求处理时，CPU暂停当前正在执行的程序，转而去执行更为紧急的中断服务程序，并能在中断服务程序处理结束后，自动返回被暂停的程序继续执行。
- 中断系统：用于实现中断的软件和硬件组成的系统
- 一般适用于随机出现的服务，且一旦提出要求应立即实现。节省了CPU的时间，但硬件结构相对复杂。

2. 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 暂停现程序

CPU 和 I/O 并行工作

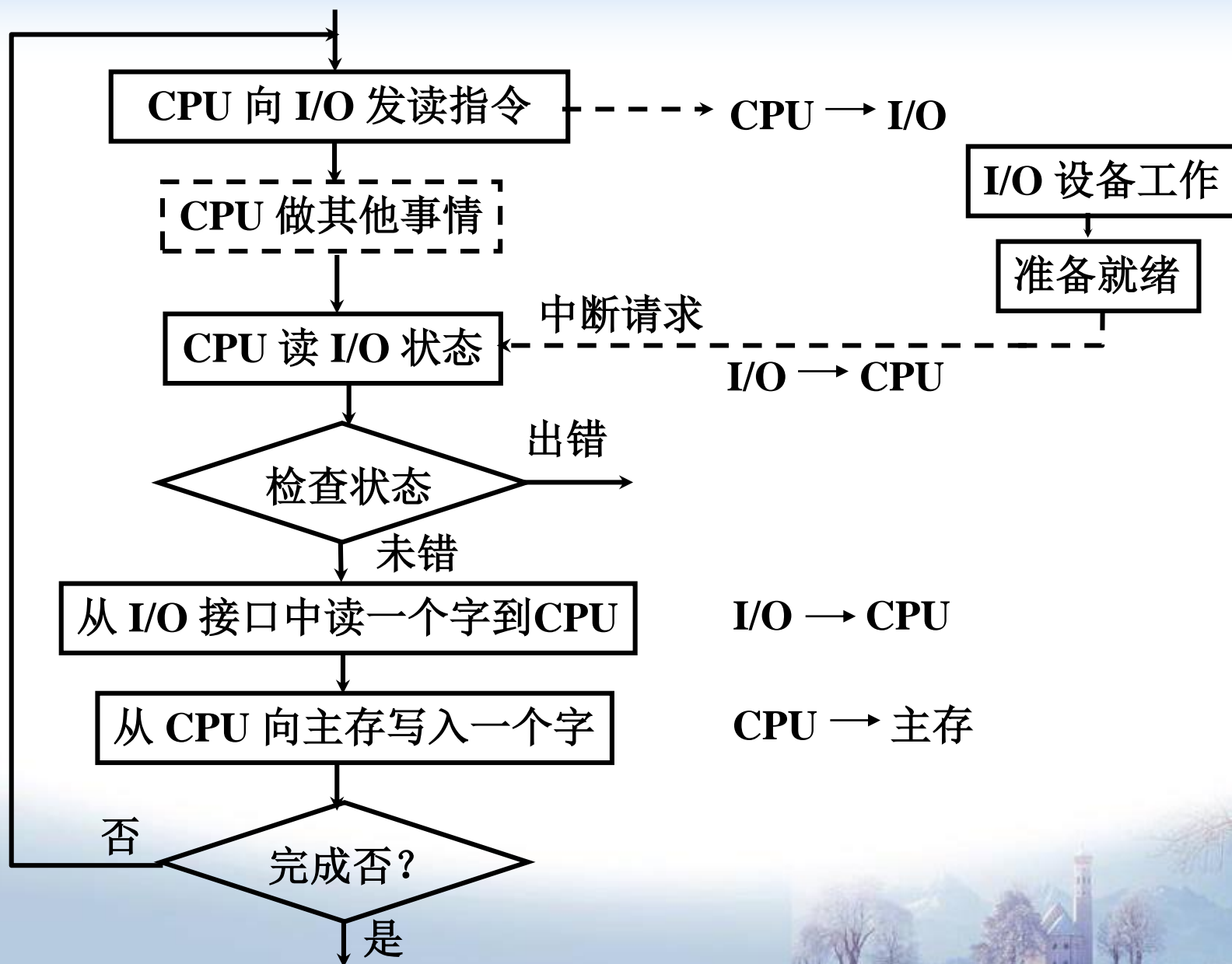


没有踏步等待现象

中断现程序

程序中中断方式流程

5.1



2. 程序中断方式

- 特点:

1. CPU与外围设备可以并行工作
2. 能够处理异常事件
3. 数据的输入输出要经过CPU
4. 一般用来连接低速设备

3. DMA 方式

- 直接存储器存取(Direct Memory Access, DMA)方式：一种直接依靠硬件在主存与设备之间进行数据传送，在数据传送期间不需要CPU程序干预的与I/O设备交换数据的工作方式。
- 在这种方式中，DMA控制器从CPU手中暂时接管对总线的控制，数据交换不必经过CPU而直接通过数据总线在内存和I/O设备之间直接进行。
- DMA方式几乎是目前所有计算机系统都要具备的一种重要工作方式。

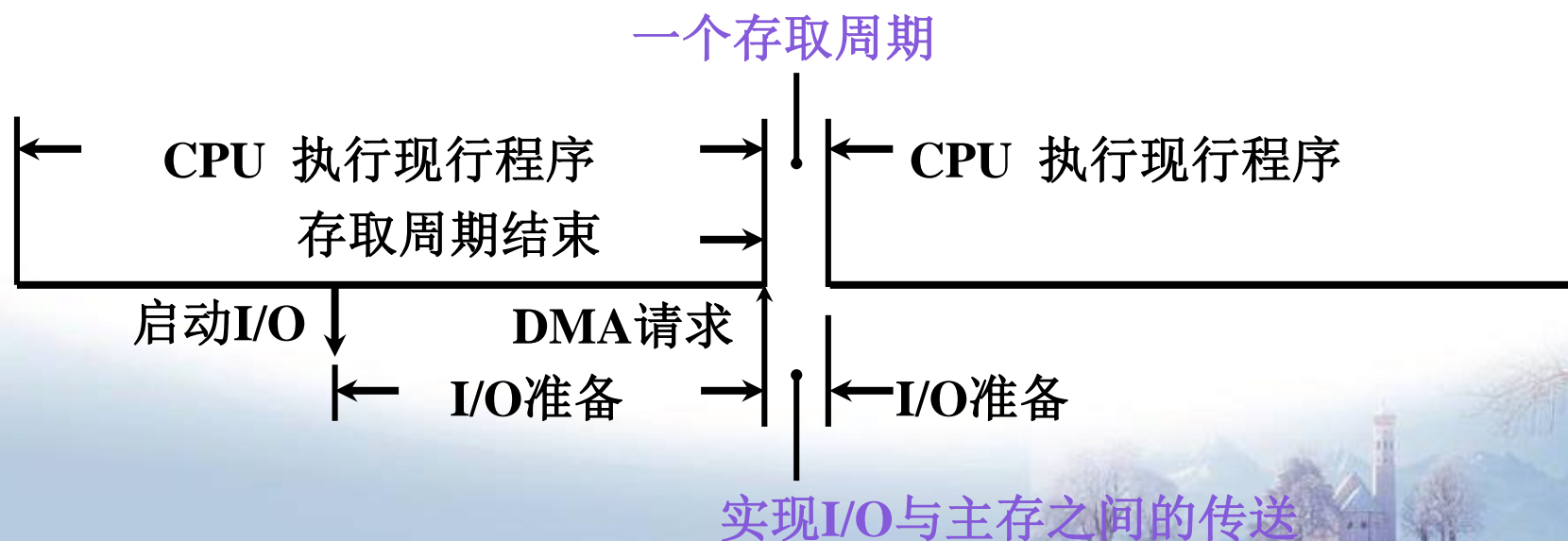
3. DMA 方式

主存和 I/O 之间有一条直接数据通道

不中断现行程序

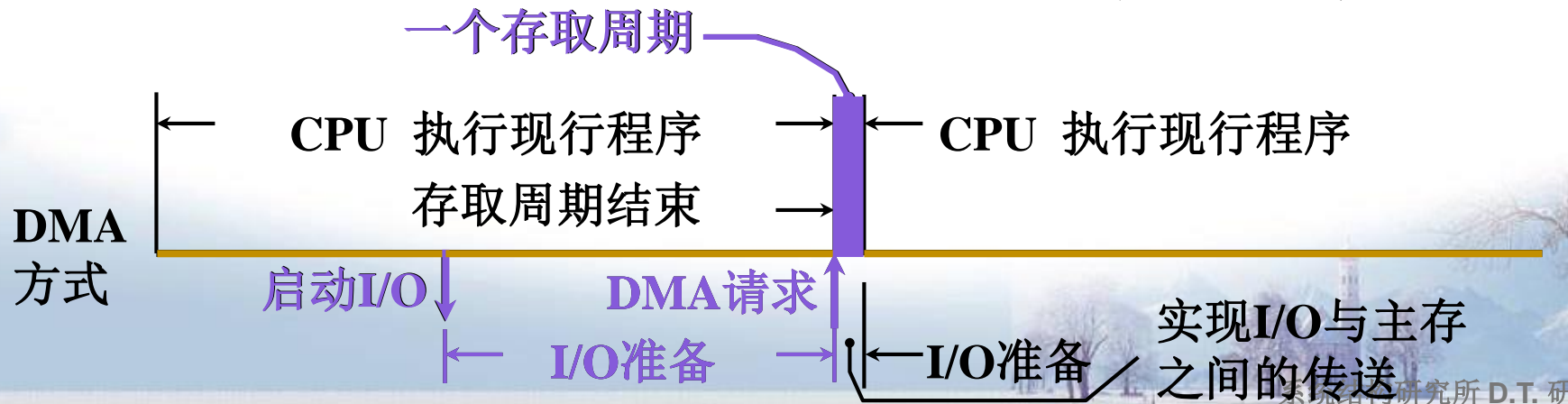
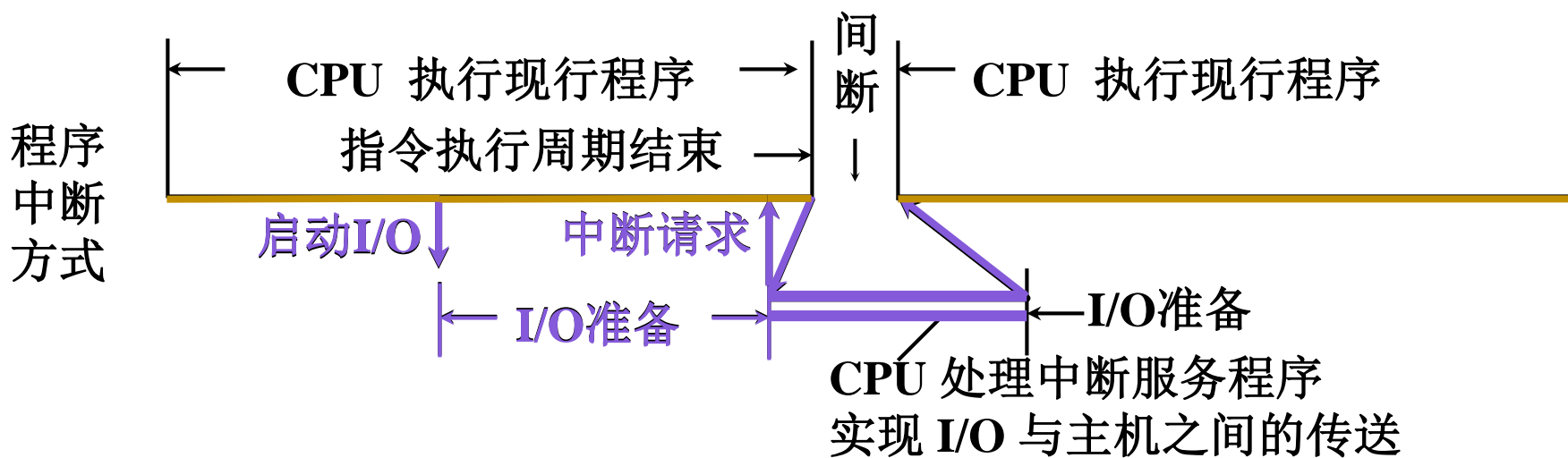
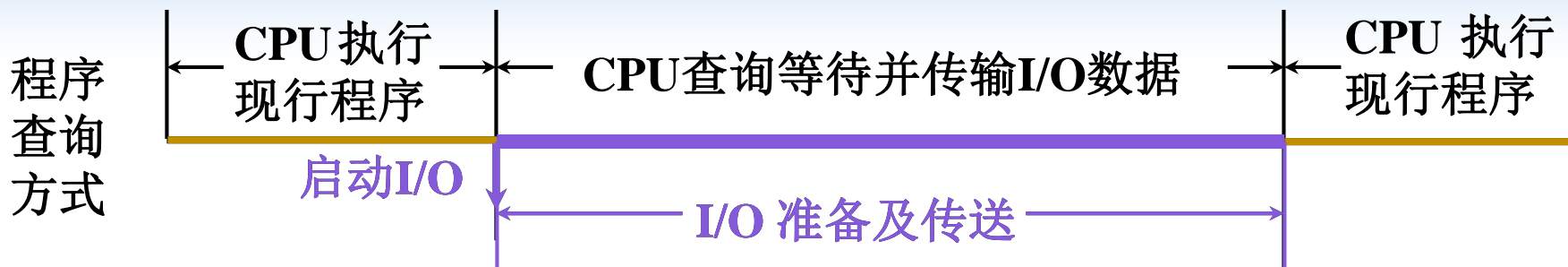
周期挪用（周期窃取）

CPU 和 I/O 并行工作



三种方式的 CPU 工作效率比较

5.1



5.2 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. 计算机信息驻留设备 | 磁盘 光盘 磁带 |
| 3. 机—机通信设备 | 调制解调器等 |

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、光电式
3. 触摸屏

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 | 喷墨（逐字） 激光（逐页） |

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

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”）

- I/O接口：I/O设备与I/O控制器之间的连接器

包括：插头 / 插座的形式、通讯规程和电器特性等

- 分类：

- 从数据传输方式来分：

- 串行（一次只传输1位）
- 并行（多位一起进行传输）

- 从是否能连接多个设备来分：

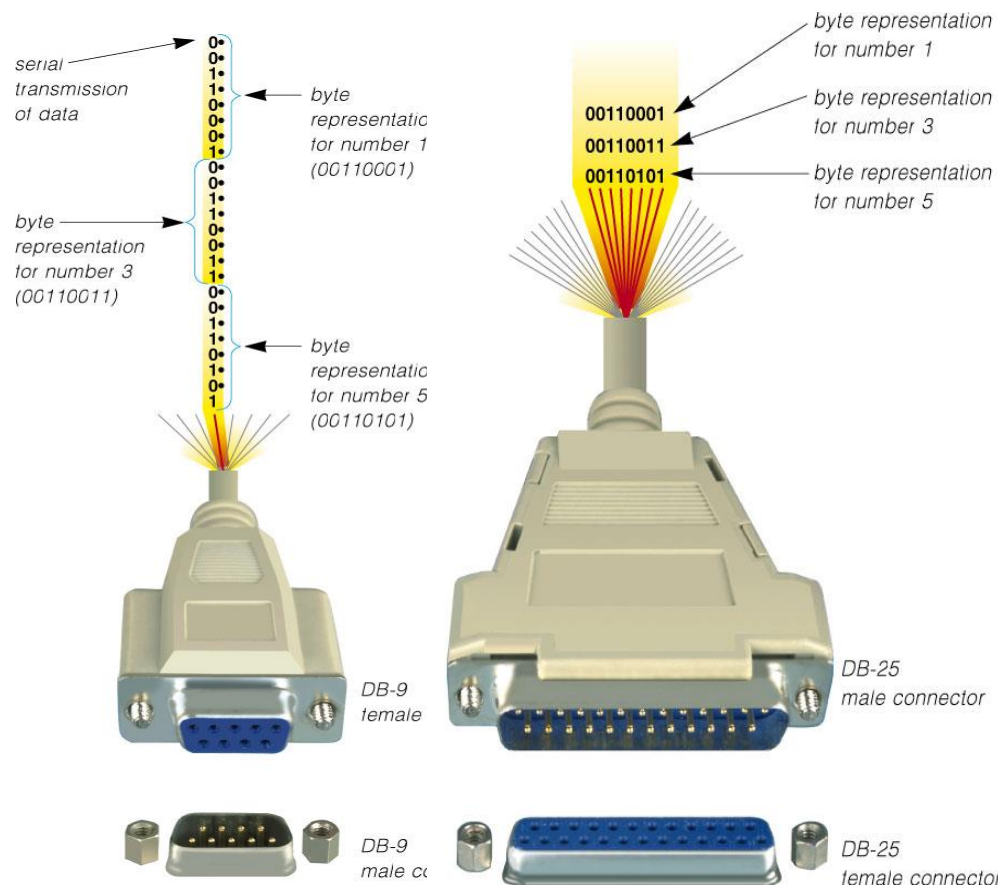
- 总线式（可连接多个设备）
- 独占式（只能连接1个设备）

- 从是否符合标准来分：

- 标准接口（通用接口）
- 专用接口（专用接口）

- 按功能选择的灵活性来分：

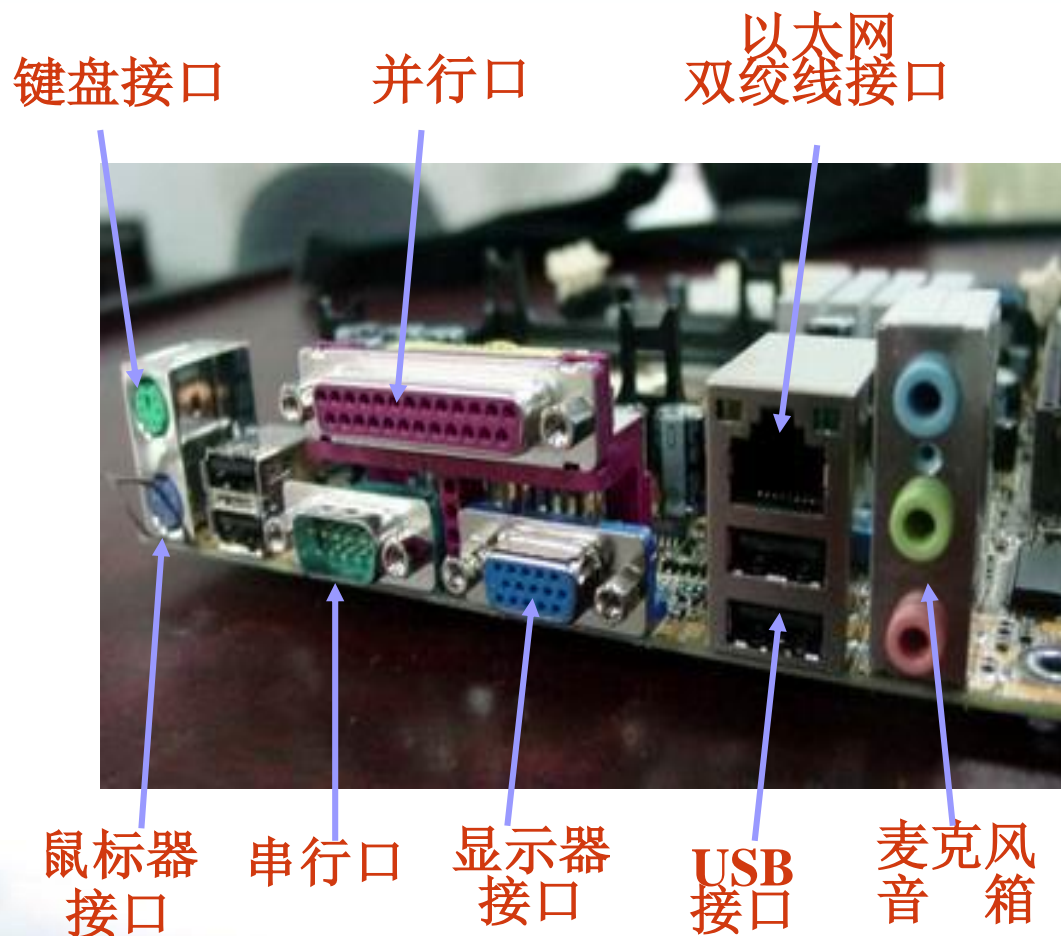
- 可编程接口
- 不可编程接口



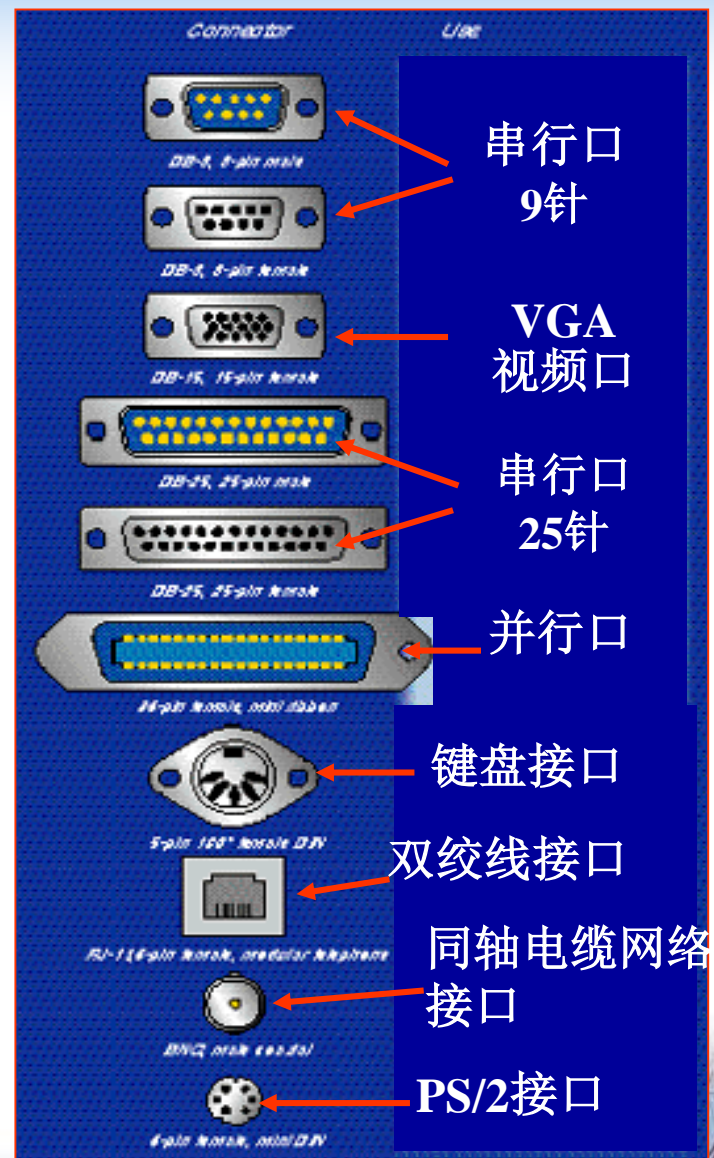
串行口

并行口

I/O设备接口



(安装在主板上的I/O设备接口)



接口 (Interface) 与端口(Port)

- 接口 (Interface) 和端口(Port)不同:
- 接口: 指一种为在主机和外设之间传送信息而设置的硬件线路, 也可以指两个软件之间的共同逻辑边界。
- 端口: 接口电路中的一些寄存器, 这些寄存器分别用来存放数据信息、控制信息和状态信息, 相应的就是数据端口、控制端口和状态端口。
- 若干个端口加上相应的控制逻辑才能组成接口。CPU通过输入指令, 从端口读入信息, 通过输出指令, 可将信息写入到端口中。

二、接口的功能和组成

5.3

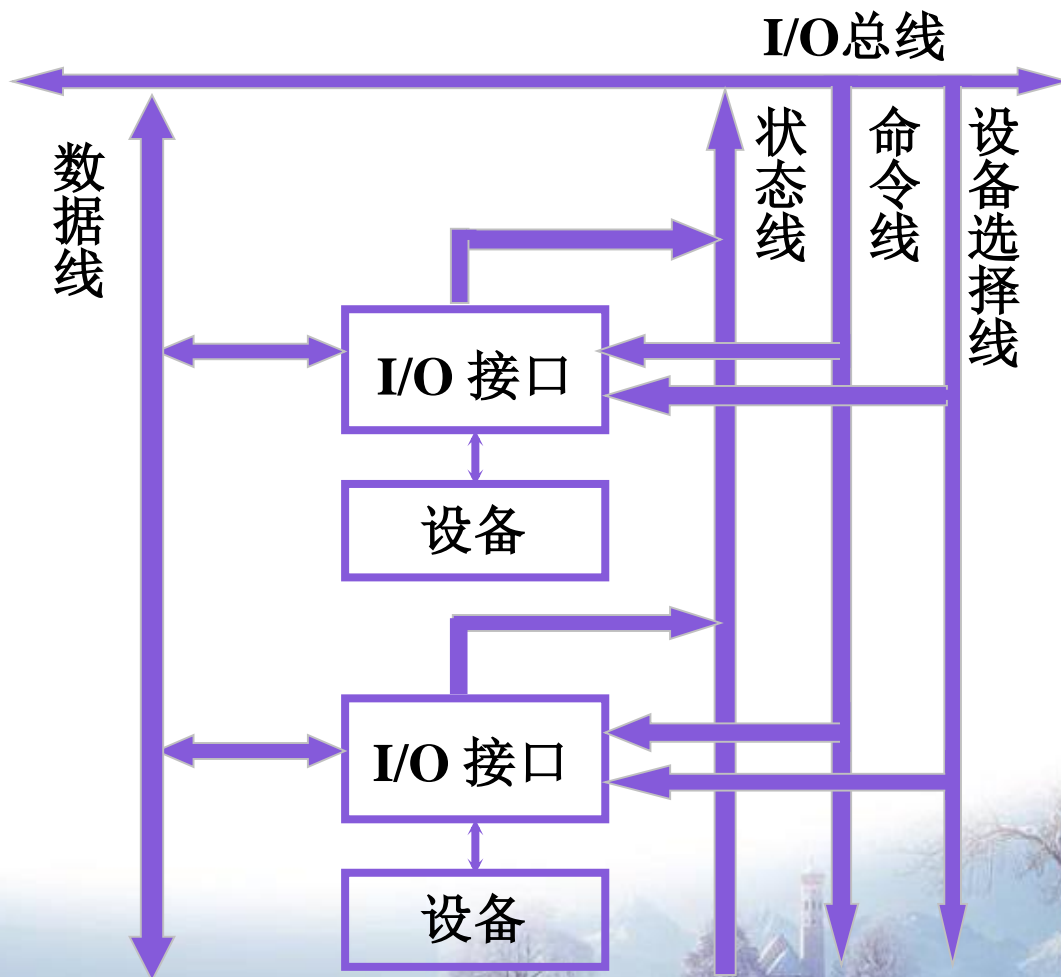
1. 总线连接方式的 I/O 接口电路

(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，即选择设备并传送地址。当设备选择线上的设备码与本设备码相符，发出设备选中信号SEL，通过接口内的设备选择电路实现。
- 传送主机命令，即CPU向I/O发出的命令。
- 数据的缓冲功能，CPU与外设的速度往往不匹配，为消除速度差异，接口必须提供数据缓冲功能。通过数据缓冲寄存器DBR（Data Buffer Register）实现

2. 接口的功能和组成

- 反映设备的工作状态, 内部设置有反映工作状态的触发器 - 设备状态标记。
- 输入输出功能, 接口能按照读写信号从总线上接收CPU送来的数据和控制信息或把数据和状态信息送到总线上。
- 数据转换功能, 不同外设信息格式不同, 与主机信息格式也不同, 接口提供主机与外设信息格式的转换, 如正负逻辑的转换、串 - 并转换、数/模或模/数转换。
- 除以上功能外, 接口还具有检错纠错功能, 中断功能、时序控制功能等。

2. 接口的功能和组成

5.3

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

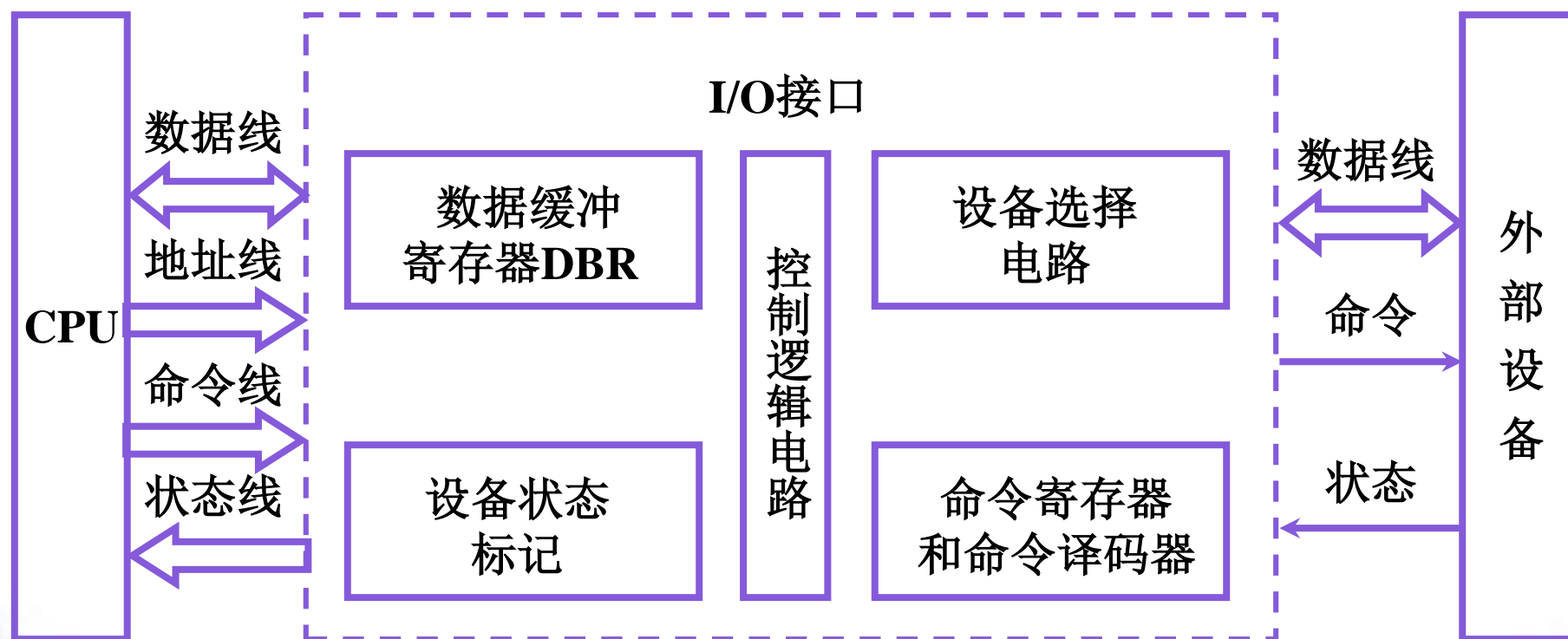
设备选择电路

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

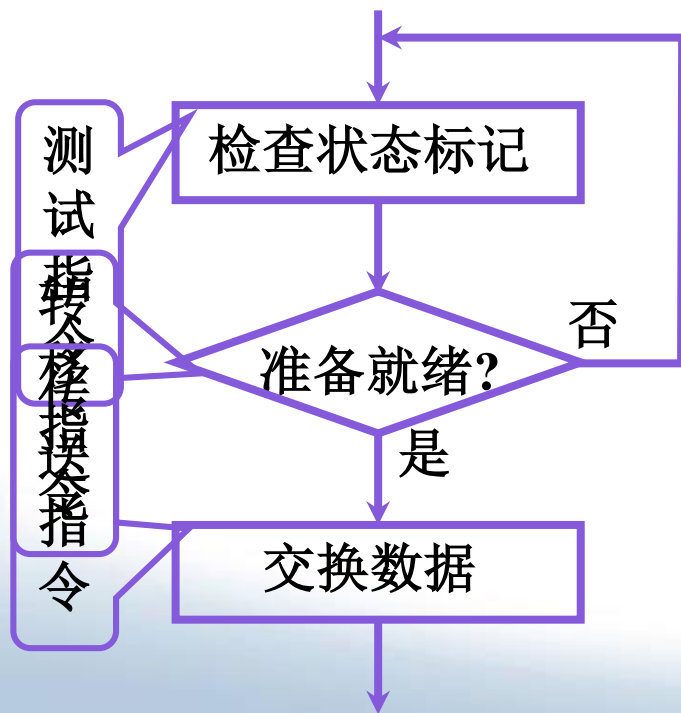
DMA 接口 Intel 8257

5.4 程序查询方式

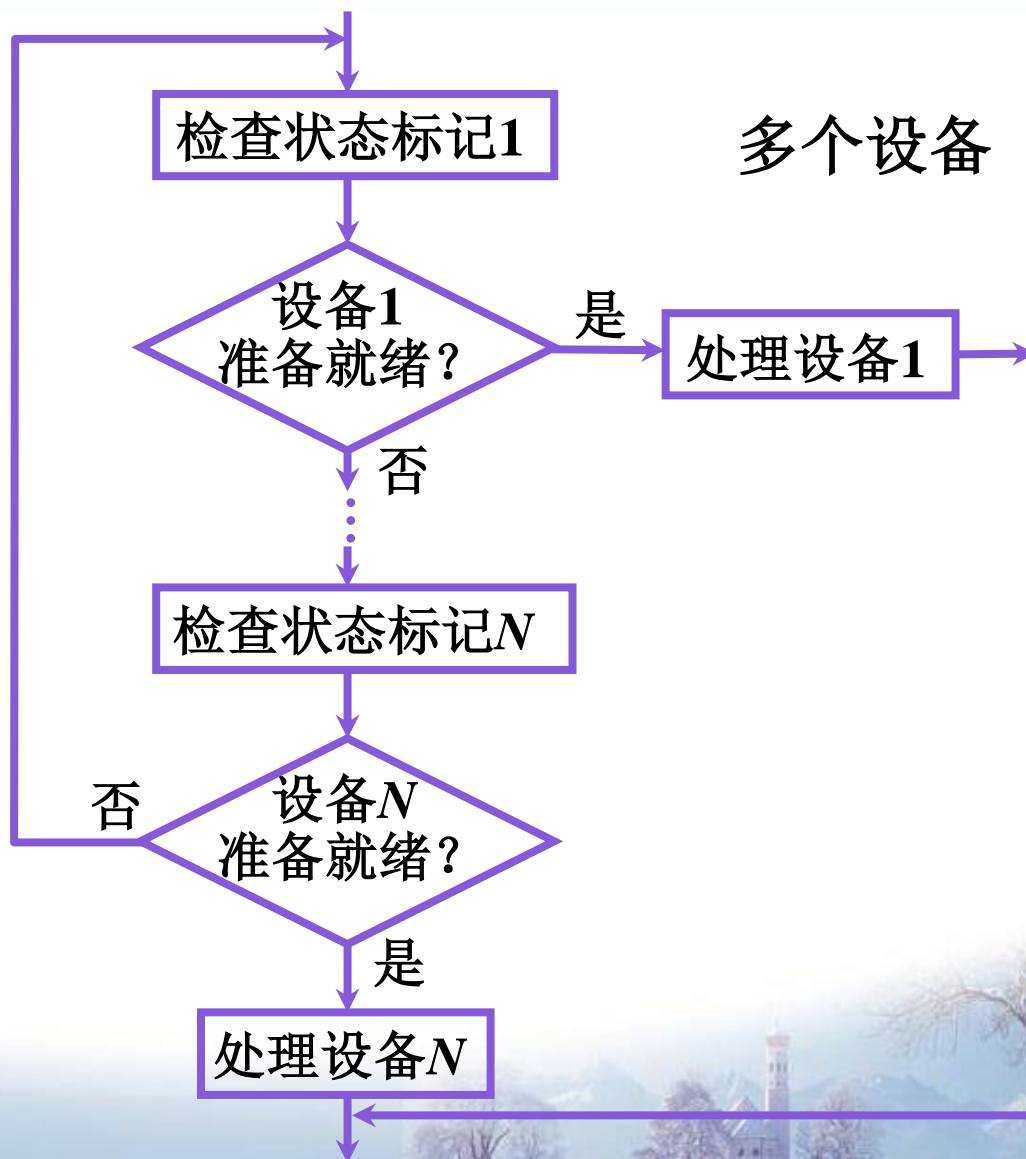
一、程序查询流程

1. 查询流程

单个设备

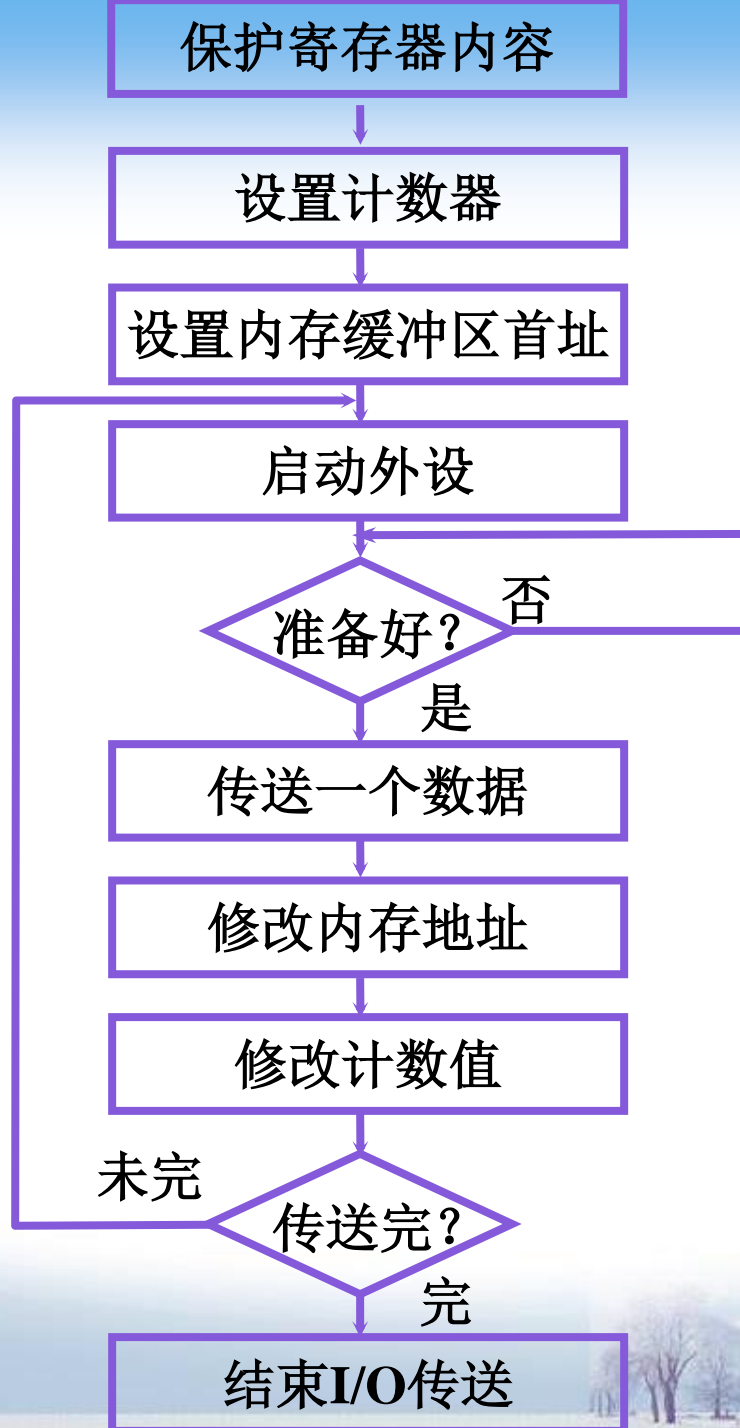


多个设备



2. 程序流程

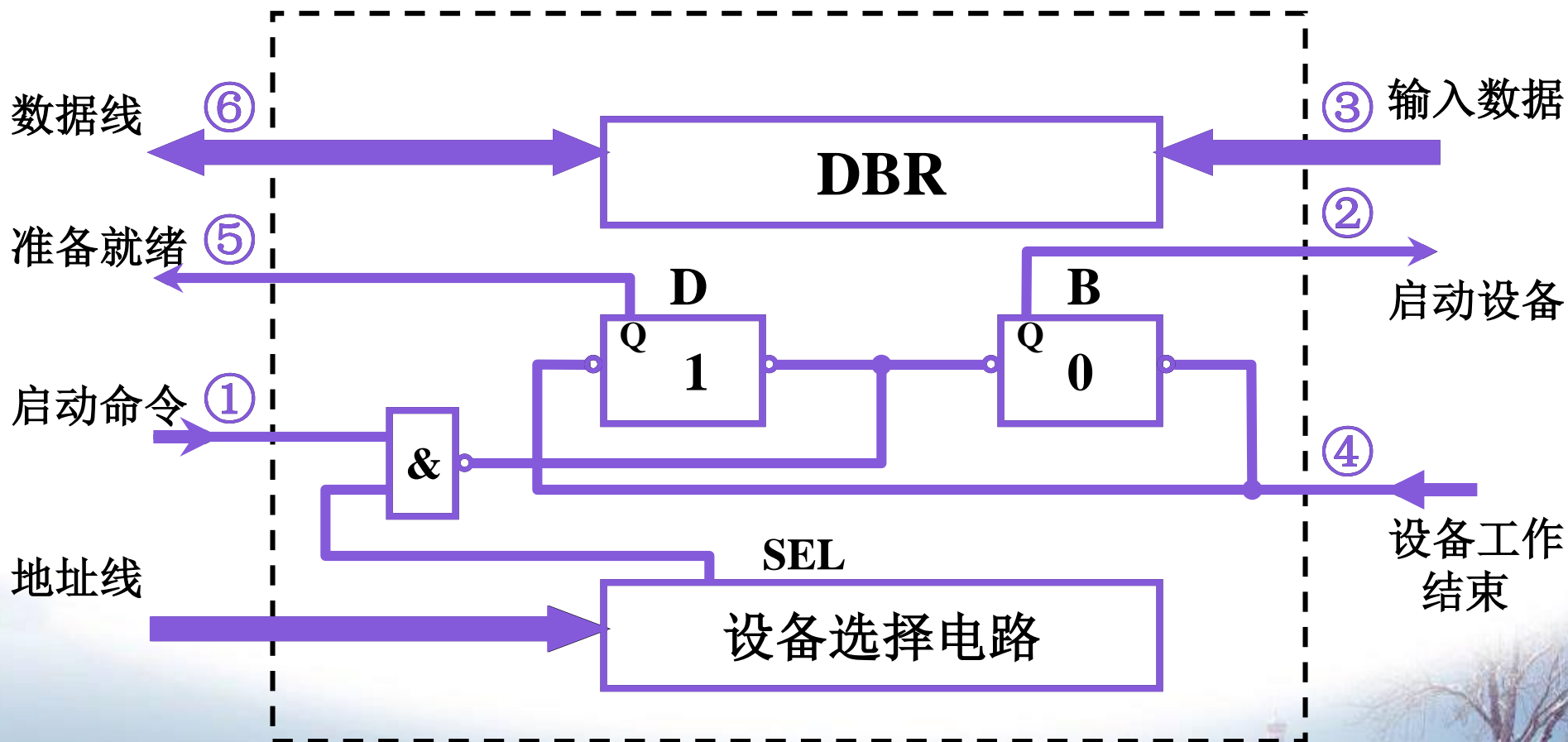
5.4



二、程序查询方式的接口电路

5.4

以输入为例

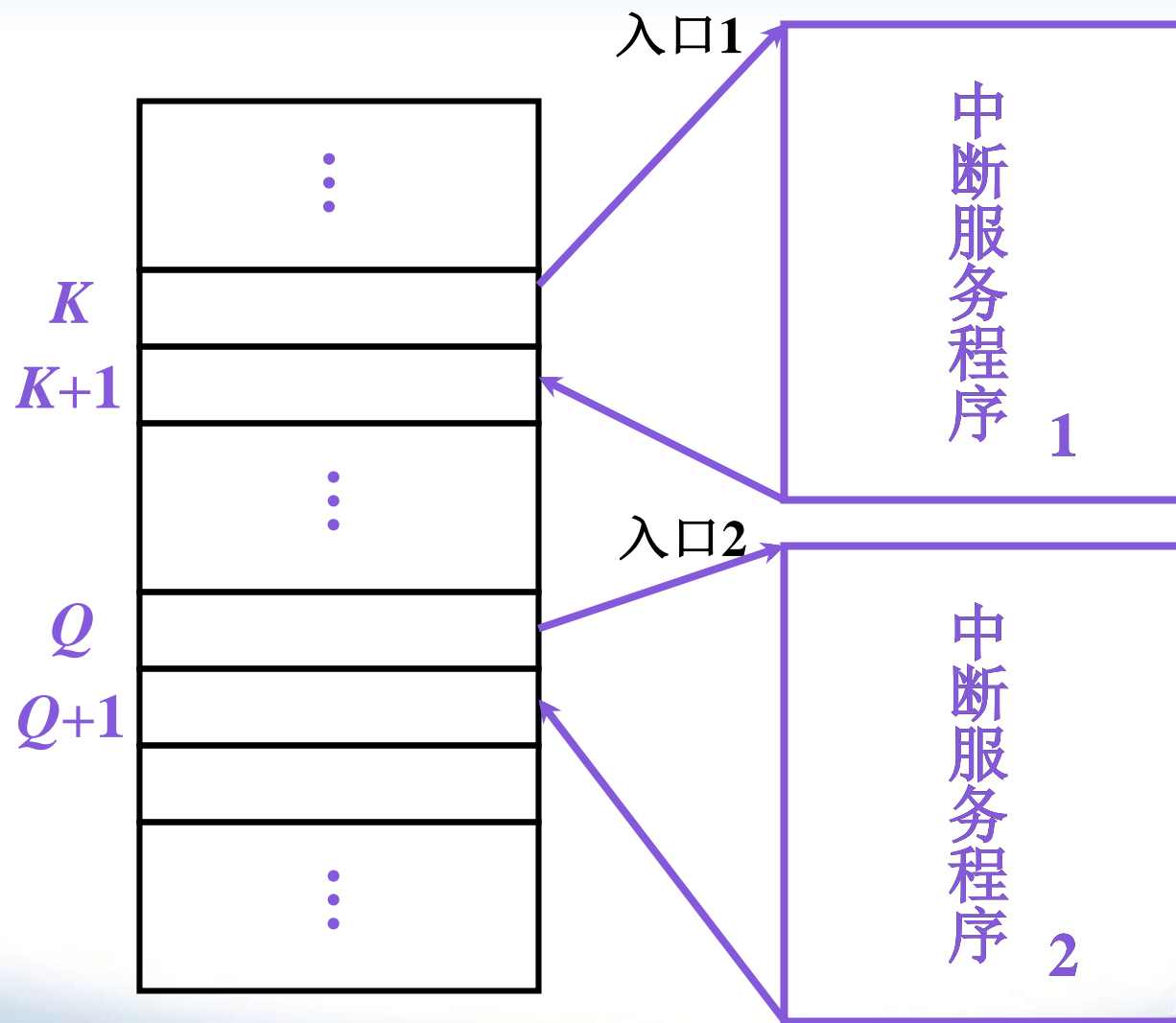


5.5 程序中中断方式

一、中断的概念

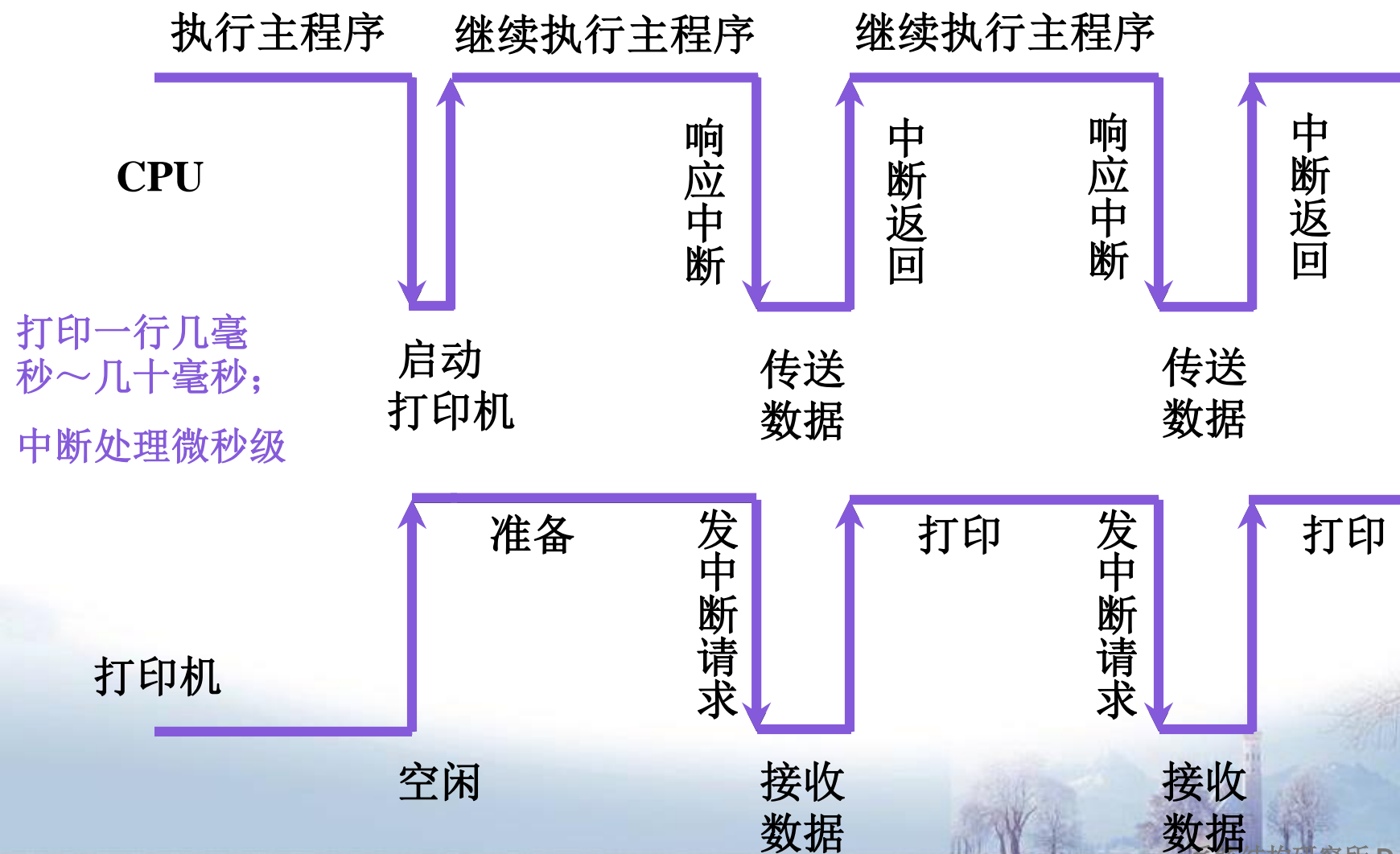
- 中断：CPU在执行程序的过程中，当出现异常情况或特殊请求处理时，CPU暂停当前正在执行的程序，转而去执行更为紧急的中断服务程序，并能在中断服务程序处理结束后，自动返回被暂停的程序继续执行。
- 中断系统：用于实现中断的软件和硬件组成的系统
- 一般适用于随机出现的服务，且一旦提出要求应立即实现。这种方式节省了CPU的时间，但硬件结构对复杂一些。

5.5



二、中断的作用

1. 实现CPU与I/O设备并行工作 （以打印机为例）



二、中断的作用

1. 实现CPU与I/O设备并行工作
2. 硬件故障：计算机运行时，如果硬件出现某些故障机器中断系统发出中断请求，CPU响应中断后自动进行处理。
3. 实现人机联系：在计算机工作过程中，如果用户要干预机器，如抽查计算中间结果、了解机器工作状态、给机器下达临时命令，一般通过中断系统完成。
4. 实现多道程序和分时操作：实现多道程序运行是提高计算机效率的有效手段。多道程序切换运行需要借助中断系统。在一道程序运行中，由I/O中断系统切换到另一道程序运行。也可以通过给每道程序分配一个固定时间片，利用时钟定时发中断请求进行程序切换

二、中断的作用

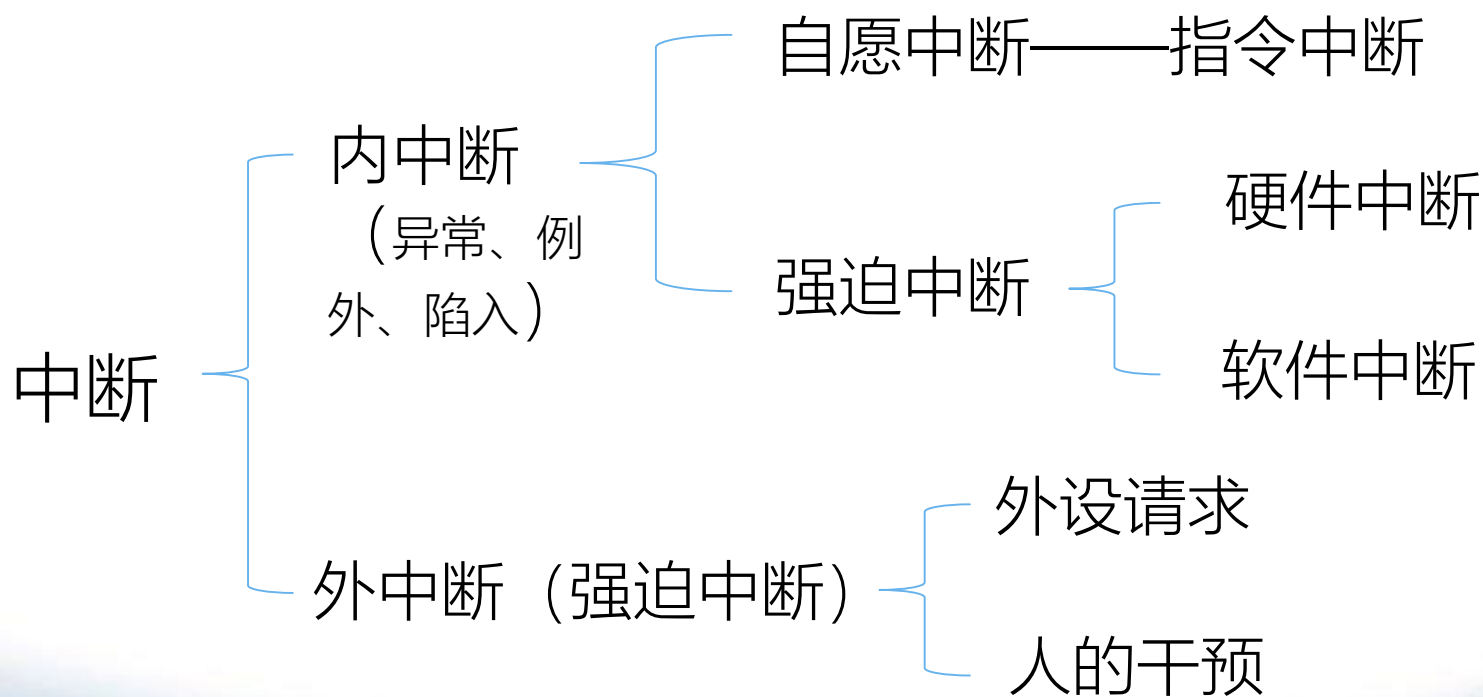
5. 实现实时处理：实时处理指在某个事件或现象出现时及时地进行处理，而不是集中起来再进行批处理。
6. 实现应用程序和操作系统（管态程序）的联系：可以在用户程序中安排一条“Trap”指令进入操作系统，称为“软中断”。其中断处理过程与其它中断类似。
7. 多处理机系统各处理机间的联系：在多处理机系统中，可以通过中断来实现处理机和处理机之间的信息交流和任务切换。

二、中断的分类

1. 按中断来源分：

(1) 内中断：发生在主机内部的中断称为内中断

(2) 外中断：由主机外部事件引起的中断称为外中断



二、中断的分类

1. 按中断来源分：

(1) **内中断**：发生在主机内部的中断称为内中断

① 强迫中断是在CPU没有事先预料的情况下发生的，此时CPU不得不停止现行的工作。产生的原因有硬件故障和软件出错等

- 硬件故障包括部件中集成电路芯片、元件、器件、印刷线路版、导线及焊点引起的故障，电源电压的下降也属于硬件故障。
- 软件出错包括指令出错、程序出错、地址出错、数据出错等。

② 自愿中断是出于计算机系统管理的需要而自愿地进入中断。

- 计算机系统为了方便用户调试软件、检查程序、调用外部设备，设置了自中断指令和进管指令。CPU执行程序时遇到这类指令就进入中断。在中断过程中调出相应的管理程序。自愿中断是可以预料的。即如果程序重复执行，断点的位置不改变。

(2) **外中断**：由主机外部事件引起的中断称为外中断，外中断是强迫中断

二、中断的分类

1. 按中断来源分

2. 按中断服务程序入口地址的获取方式分：

- (1) **向量中断**：外部设备在提出中断请求的同时，通过**硬件**自动形成**中断服务程序入口地址**。CPU响应中断后，将根据向量地址直接转入相应中断服务程序。这种具有产生向量地址的中断称为向量中断。
- (2) **非向量中断**：非向量中断在产生中断的同时不能直接提供中断服务程序入口地址，而只产生一个**中断查询程序的入口地址**。需要通过中断查询程序确定中断源和中断服务程序的入口地址。

二、中断的分类

3. 按中断源位置分：

- 硬件中断：由外设引发的，大部分可屏蔽
- 软件中断：由程序调用发生的，不可屏蔽

4. 按是否可屏蔽分：CPU根据需要可以禁止响应某些中断源的中断请求。

- 可屏蔽中断：CPU可以禁止响应的中断
- 不可屏蔽中断：CPU必须响应的中断。

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

(1) 中断请求触发器

- 当中断事件引发中断时，将中断请求触发器INTR 置“1”。
- 当中断请求触发器为“1”时，向CPU发出“中断请求”信号。
- 每个中断源有一个**中断请求触发器**。全机的多个中断请求触发器构成**中断请求寄存器**，其内容称为**中断字或中断码**。CPU进行中断处理时，根据中断字确定中断源，转入相应的中断服务程序。

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

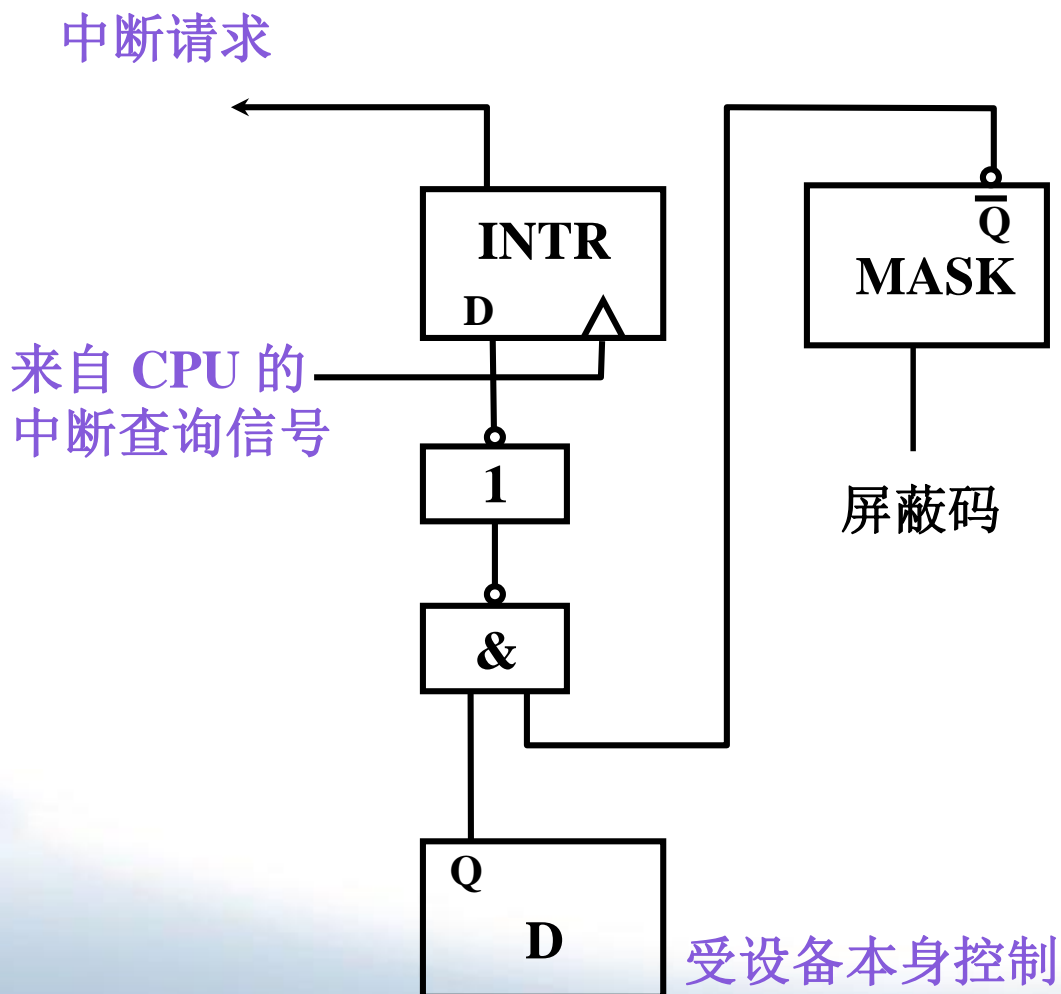
(2) 中断屏蔽：为了控制中断请求信号的产生，也为了利用屏蔽码改变中断处理的级别，当产生中断请求后，用程序方式有选择地封锁部分中断，允许其余部分中断仍得到响应，称为中断屏蔽。

- 实现方法是为每个中断源设置一个中断屏蔽触发器MASK来屏蔽该设备的中断请求。可以用程序方式将该触发器置“1”，则对应的设备中断被封锁，若将其置“0”，才允许该设备的中断请求得到响应。

(2) 中断屏蔽：

- 各设备的中断屏蔽触发器组成中断屏蔽寄存器。
- 中断屏蔽寄存器与中断请求寄存器位数相同。
- CPU可以根据需要给中断屏蔽寄存器写入屏蔽字（二进制代码），代码的每一位可屏蔽一种中断源。如当中断屏蔽寄存器的某位为“1”时，表示该位所对应的中断被屏蔽。
- 有些中断请求是不可屏蔽的。不管中断系统是否开中断，这些中断源的中断请求一旦提出，CPU必须立即响应。
 - 如：电源掉电就是不可屏蔽中断。
 - 不可屏蔽中断具有最高优先权。

(3) 接口电路中几个信号间的关系



INTR
中断请求触发器

INTR = 1 有请求

MASK
中断屏蔽触发器

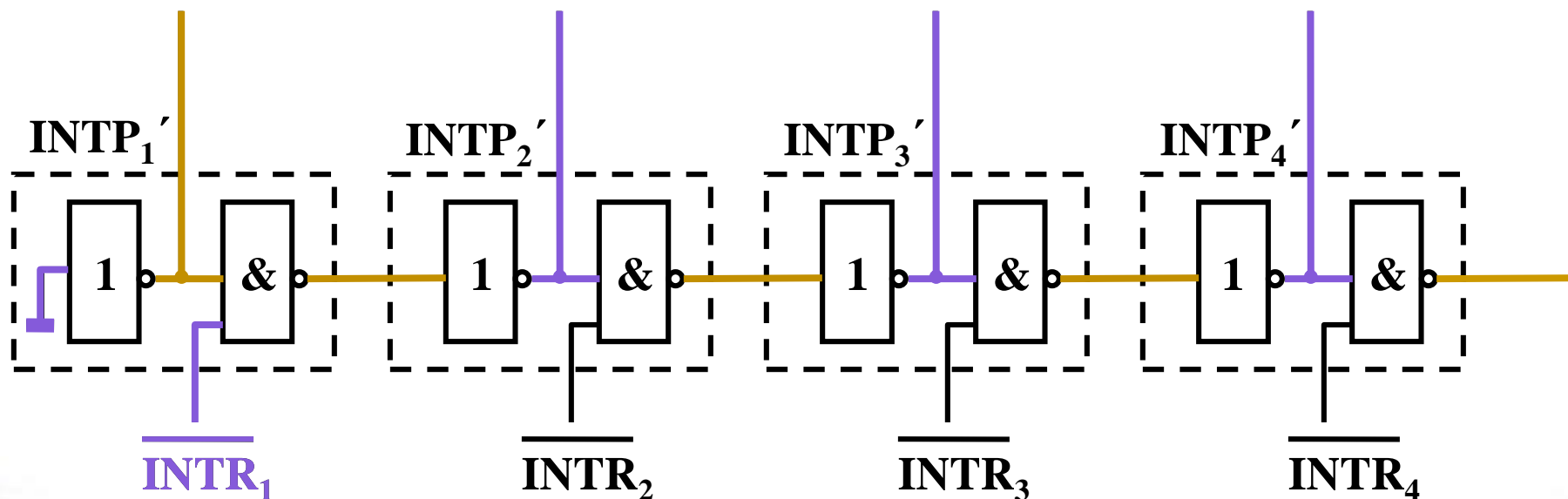
MASK = 1 被屏蔽

D 完成触发器

2. 排队器

5.5

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



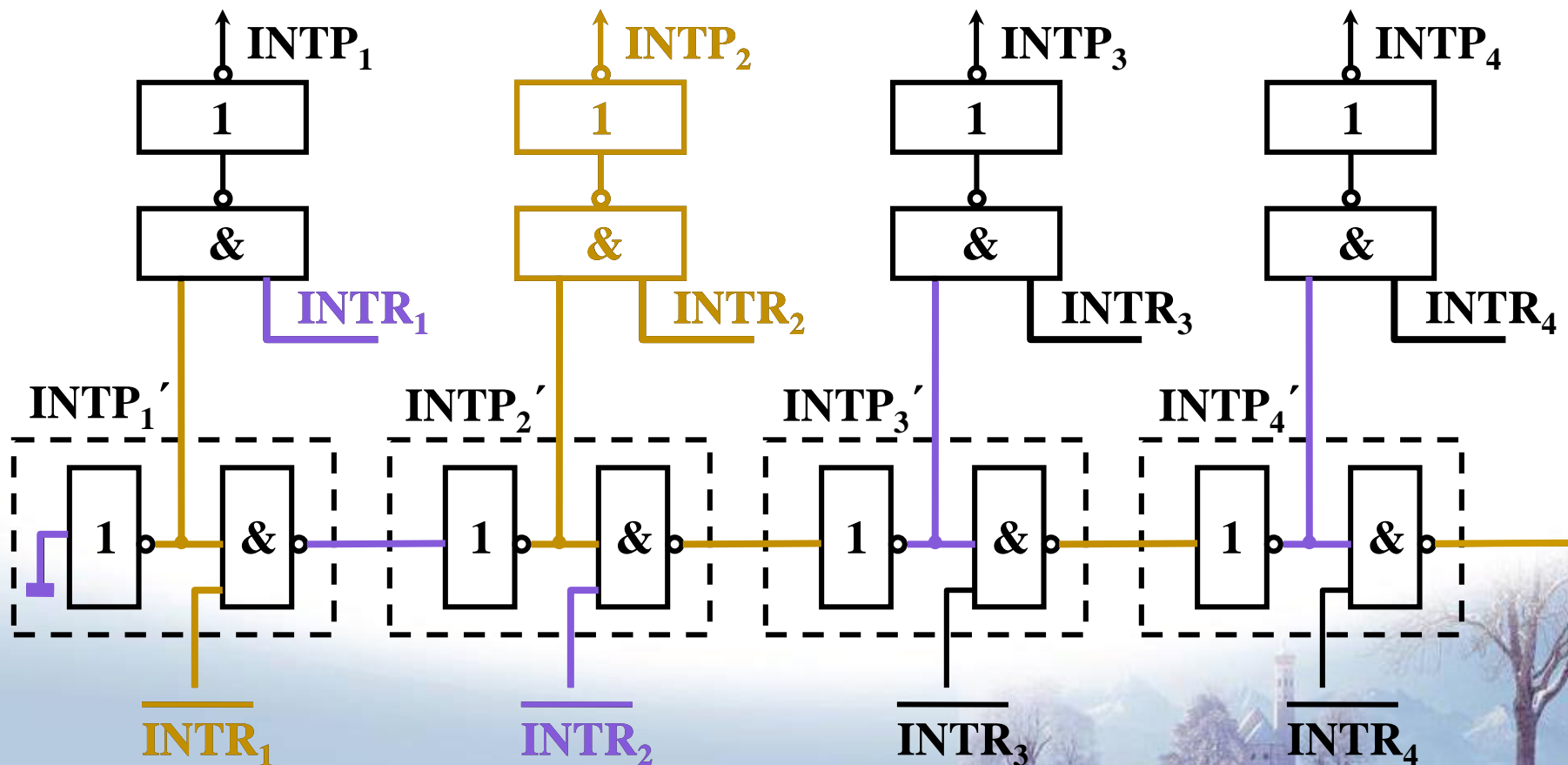
设备 1[#]、2[#]、3[#]、4[#] 优先级按 降序排列

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

2. 排队器

5.5

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



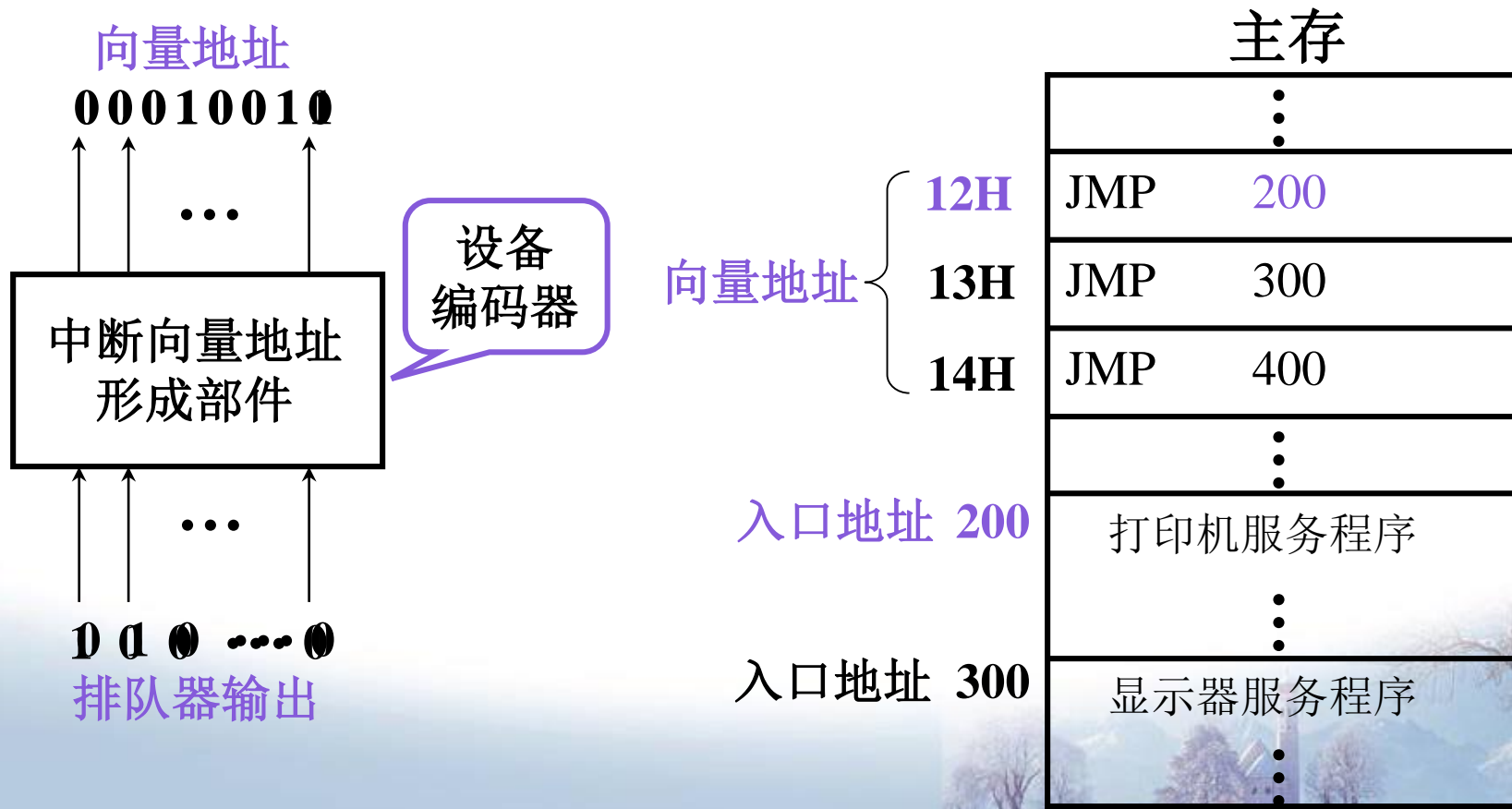
3. 中断向量地址形成部件

入口地址 { 由软件产生
硬件向量法

详见第八章

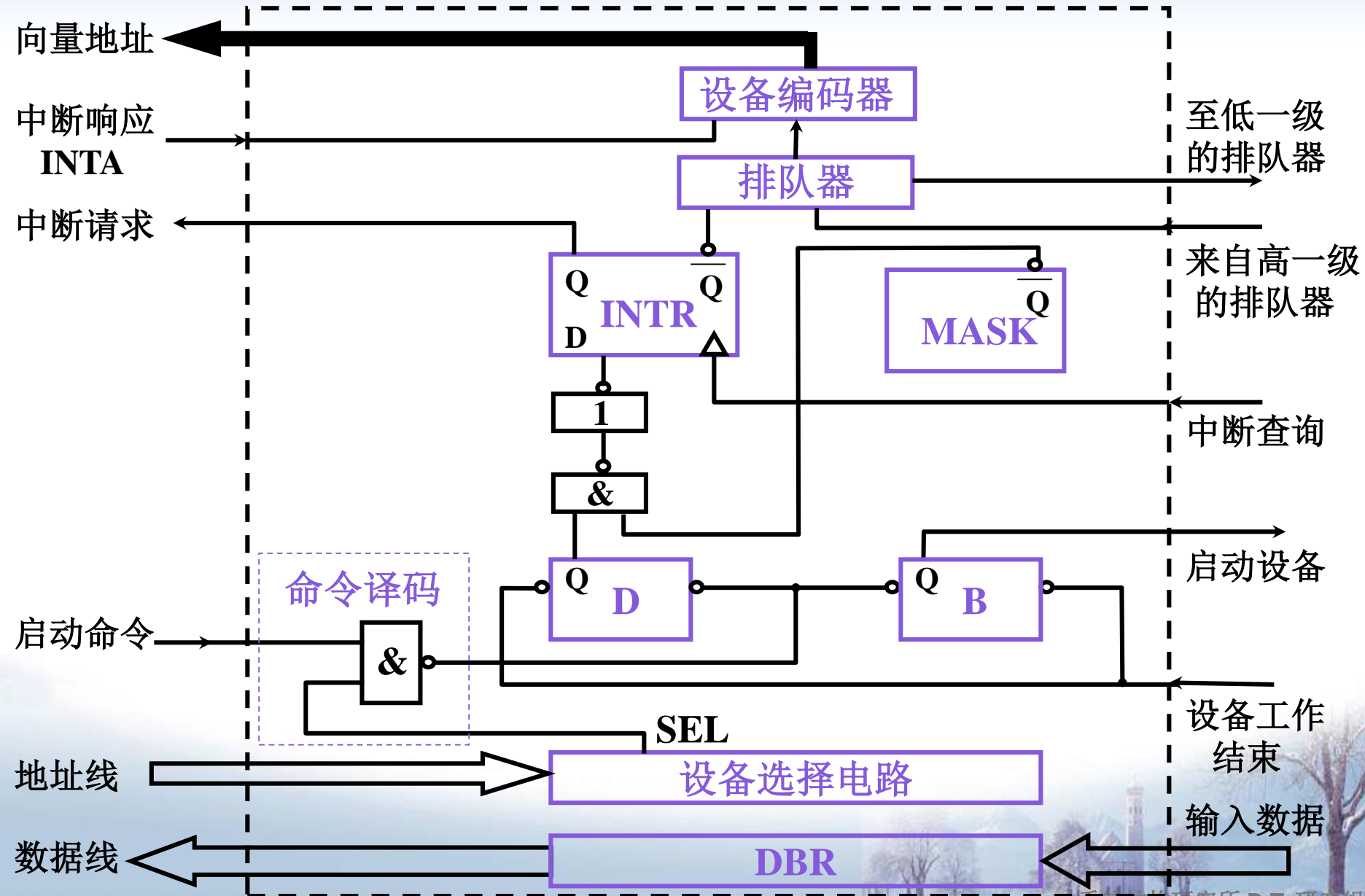
由 硬件 产生 向量地址

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

5.5



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

中断允许触发器 **EINT = 1**

用 **开中断** 指令置 “**1**” **EINT**

用 **关中断** 指令置 “**0**” **EINT** 或硬件 **自动复位**

(2) 时间

当 **D = 1**（随机）且 **MASK = 0** 时

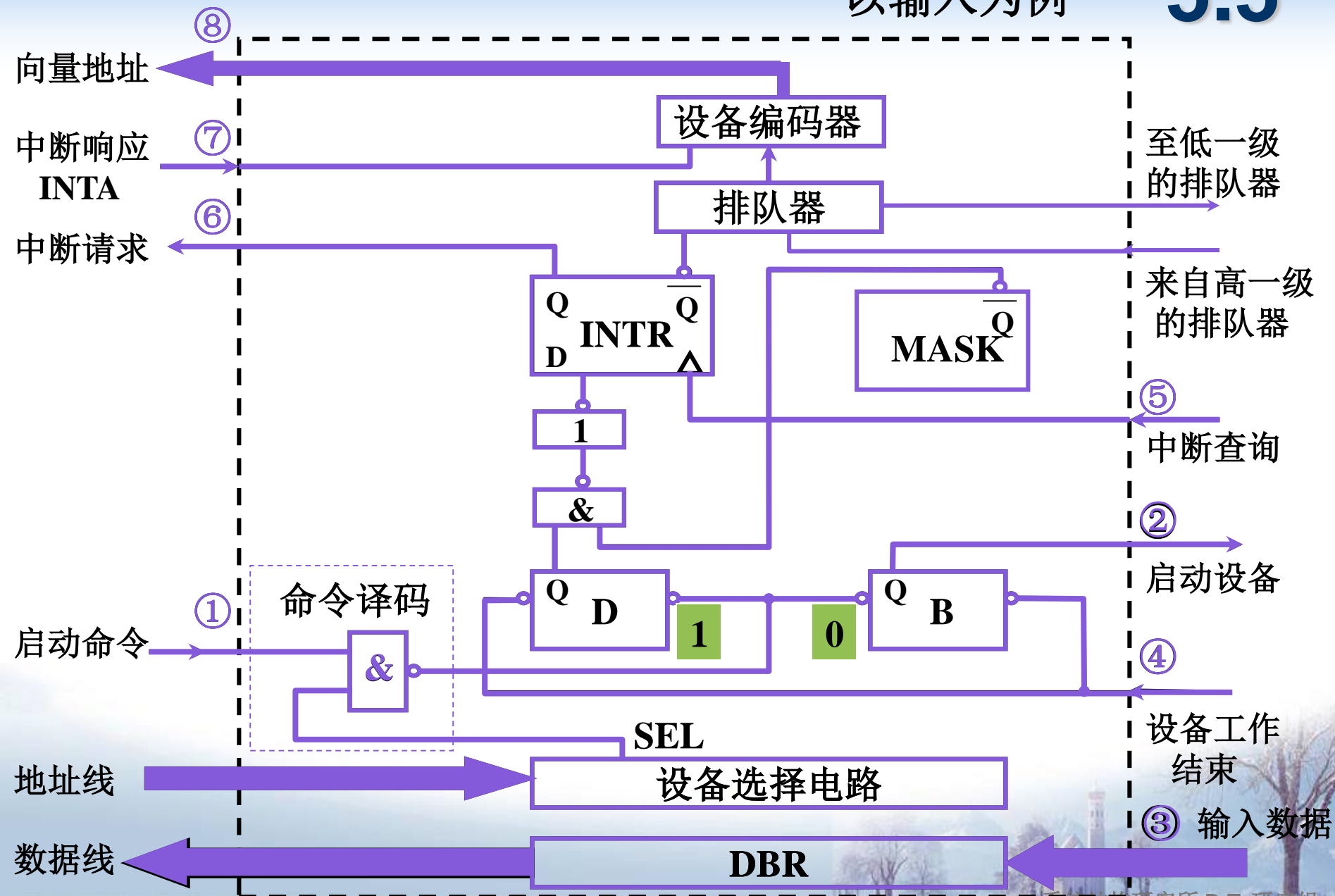
在每条指令执行阶段的结束前

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

5.5



五、中断服务程序流程

在机器指令系统中没有的指令，它是**CPU**在中断周期内由硬件自动完成的一条指令。

1. 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

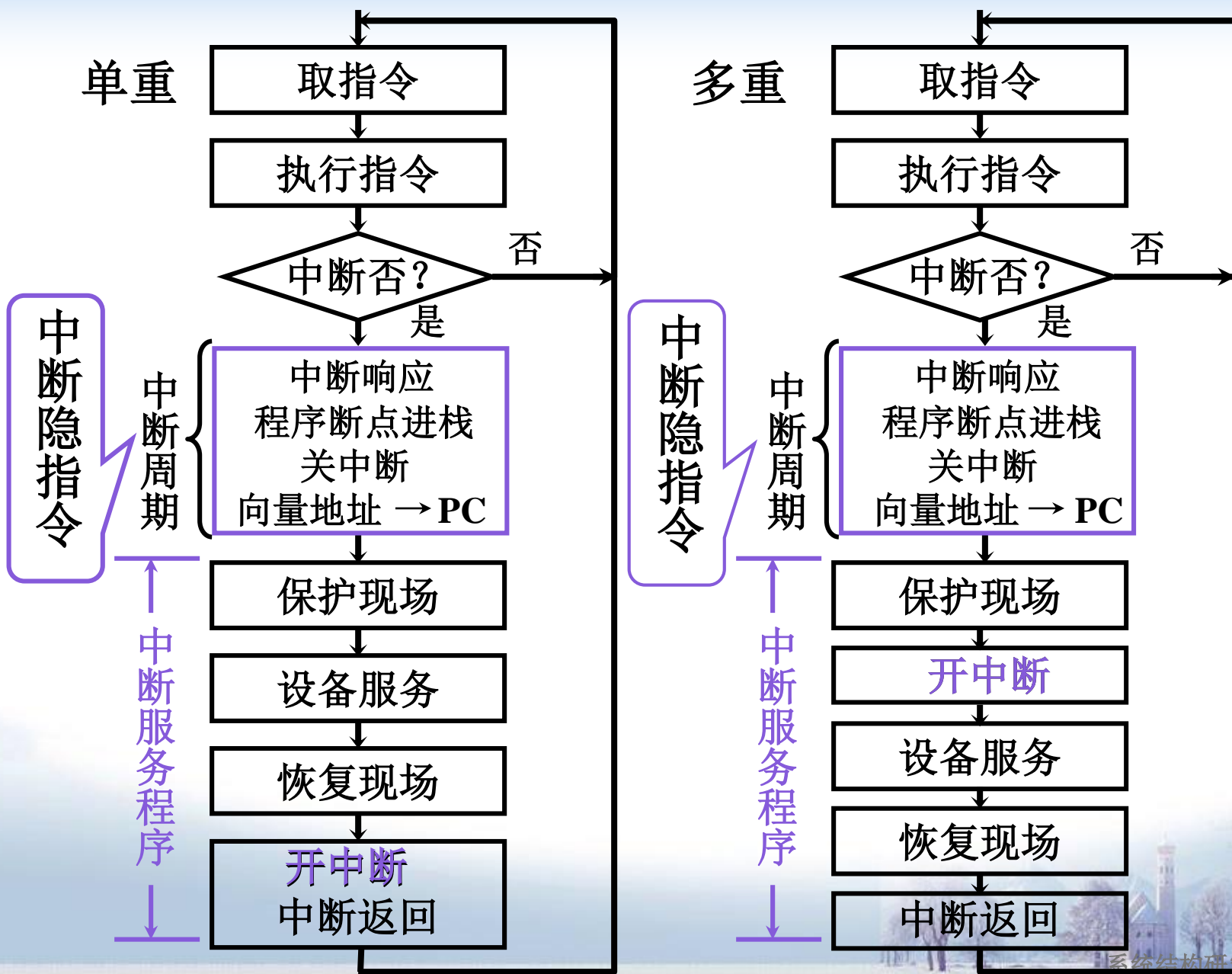
(4) 中断返回

中断返回指令

2. 单重中断和多重中断

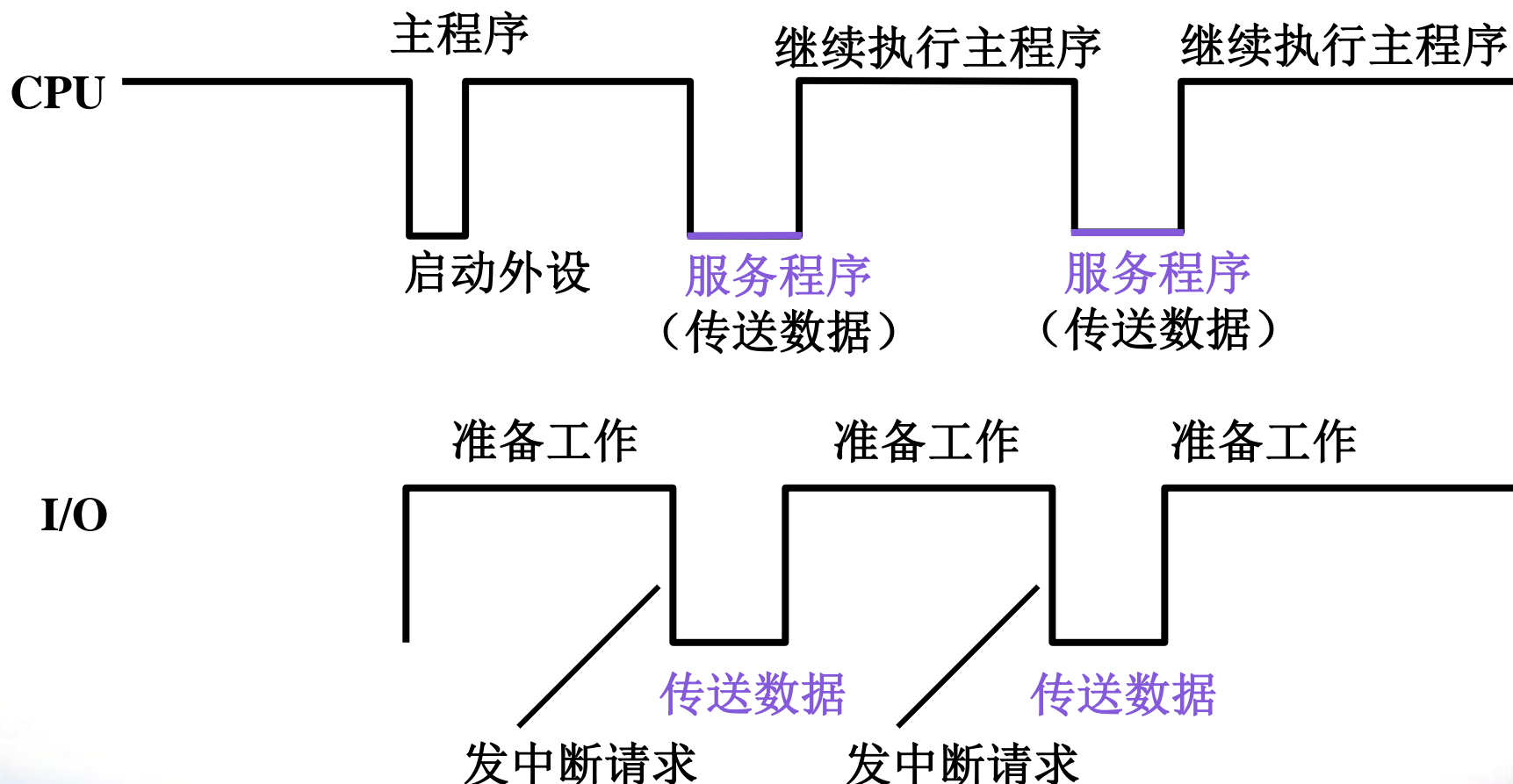
单重 中断	不允许中断 现行的 中断服务程序
多重 中断	允许级别更高 的中断源
(中断嵌套)	中断 现行的 中断服务程序

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

5.5

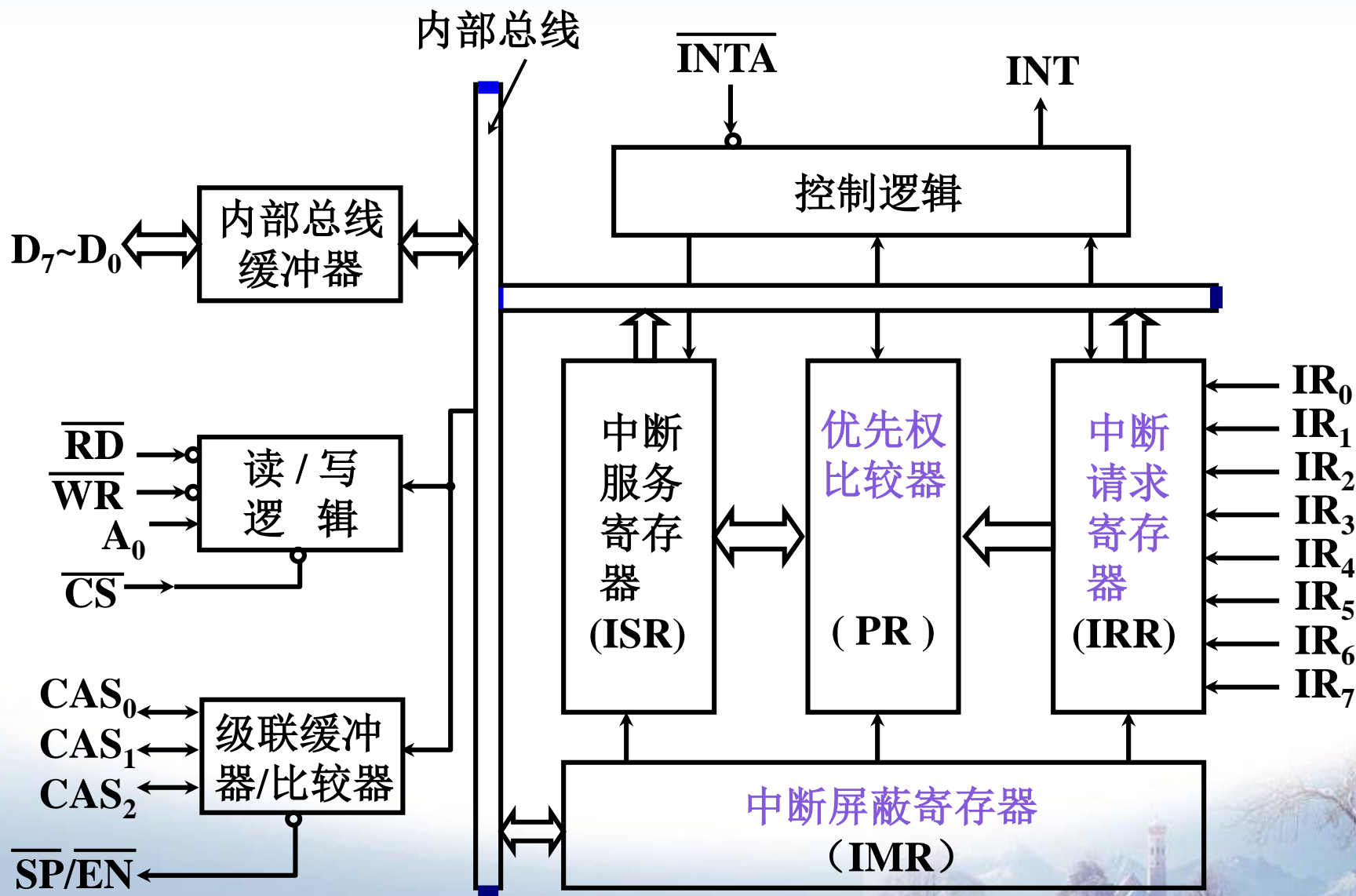


宏观上 CPU 和 I/O 并行工作

微观上 CPU 中断现行程序为 I/O 服务

程序中中断接口芯片 8259A 的内部结构

5.5



附加：多核CPU对中断的处理

多核CPU的中断处理和单核有很大不同。多核的各处理器核心之间需要通过中断方式进行通信，所以CPU芯片内部既有各处理器核心的**本地中断控制器**，又有负责仲裁各核之间中断分配的**全局中断控制器**。

现今的多核处理器在中断处理和中断控制方面主要使用的是**APIC**（Advanced Programmable Interrupt Controllers），即**高级编程中断控制器**。它是基于中断控制器两个基础功能单元——本地单元以及I/O单元的分布式体系结构。在多核系统中，多个本地和I/O APIC单元能够作为一个整体通过APIC总线互相操作。

APIC的功能有：

1. 接受来自处理器中断引脚的内部或外部I/O APIC的中断，然后将这些中断发送给处理器核心进行处理；
2. 在多核处理器系统中，接收和发送核内中断消息；

对于外部设备发出的中断请求，由全局中断控制器接收请求并决定交给CPU的哪一个核心处理。也可针对APIC编程，让所有的中断都被一个固定的CPU处理。

5.6 DMA 方式

5.5.1 DMA方式的特点

5.5.2 DMA接口的功能和组成

5.5.3 DMA的工作过程

5.5.4 DMA接口的类型



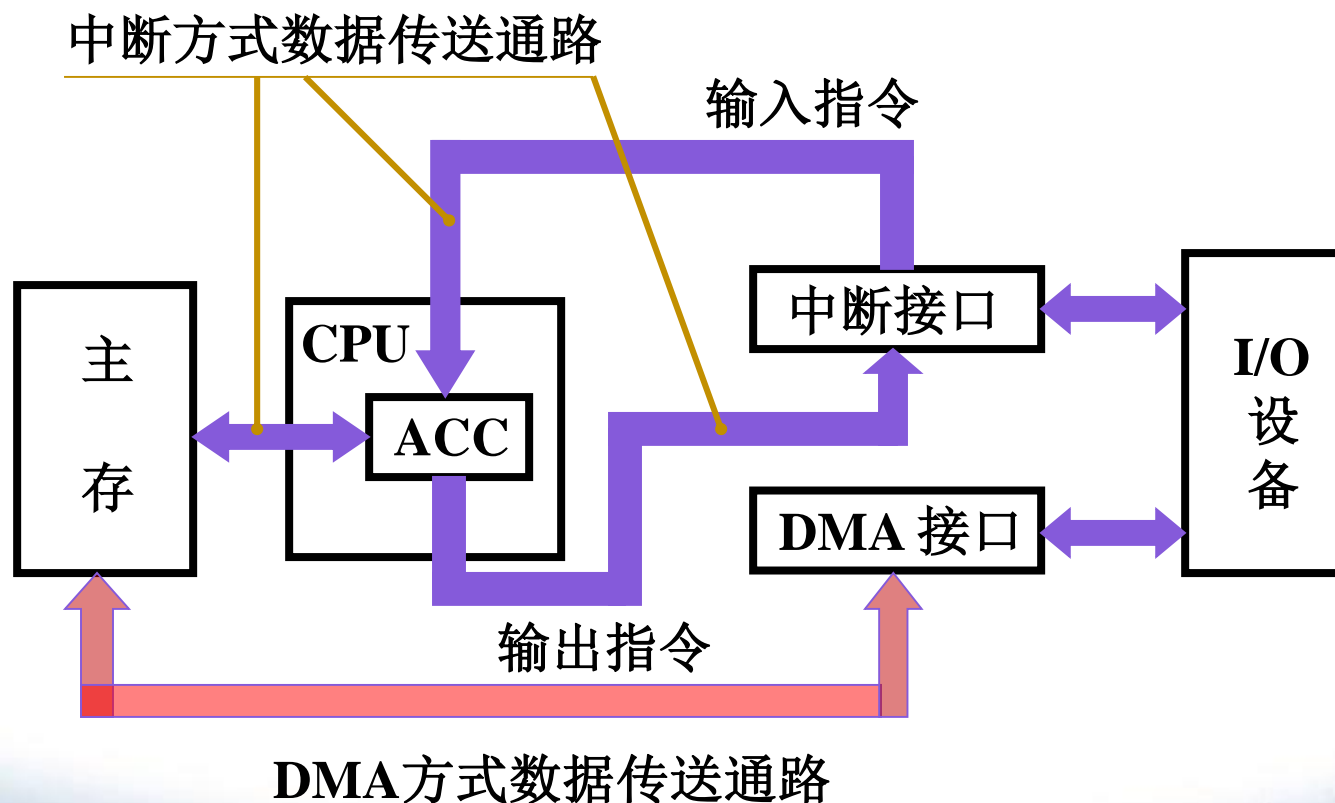
5.6 DMA 方式

一、DMA 方式的特点

- 特点: 主机与I/O并行工作, 主存和I/O之间有一条直接数据通路, CPU启动I/O后, 不必查询I/O是否准备好, 当I/O准备就绪后, 发出DMA请求, 此时CPU不直接参与I/O和主存间的信息交换, 只是把外部总线的使用权暂时交付DMA, 仍然可以完成自身内部的操作, 所以不必中断现行程序, 只需暂停一个或几个存取周期访存, CPU效率更高。

一、DMA 方式的特点

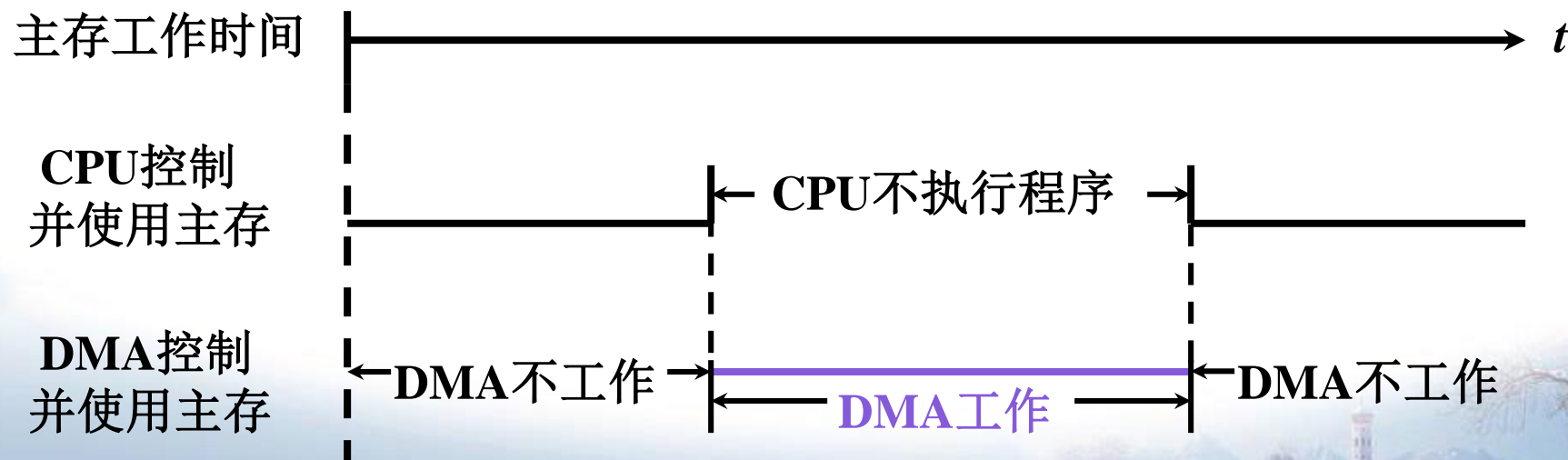
1. DMA 和程序中中断两种方式的数据通路



2. DMA 与主存交换数据的三种方式

(1) 停止 CPU 访问主存

外设要求传送一批数据，由DMA接口向CPU发一个停止信号，要求CPU放弃地址线、数据线及有关控制线的使用权。DMA接口获得总线控制权后，开始进行数据传送，在数据传送结束后，DMA接口通知CPU可以使用主存，把总线控制权交给CPU。

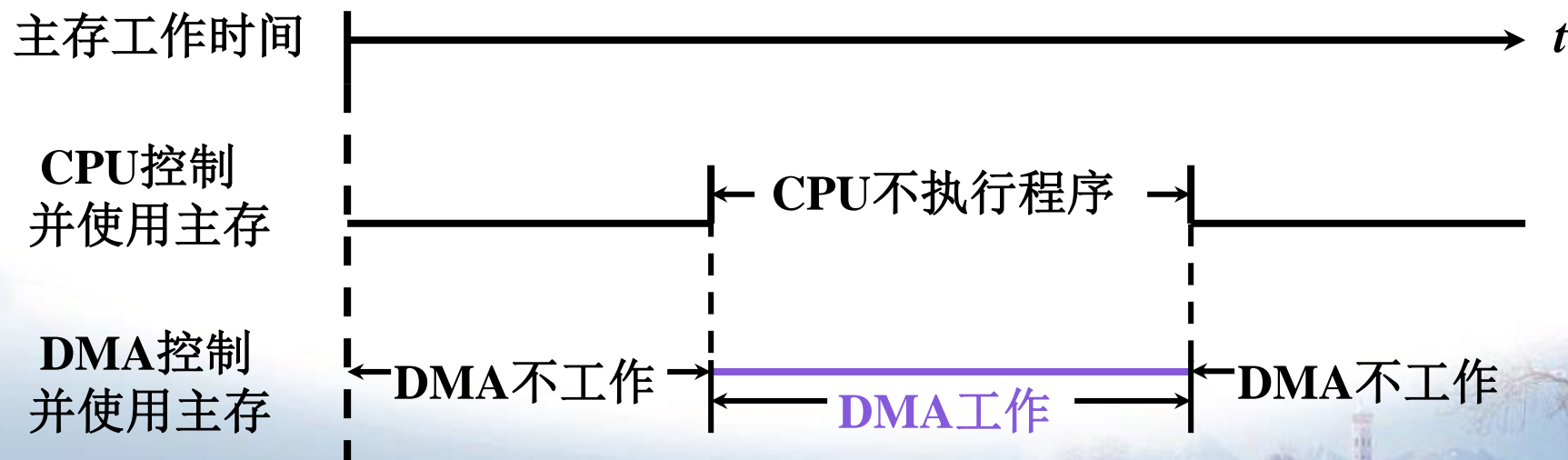


2. DMA 与主存交换数据的三种方式

(1) 停止 CPU 访问主存

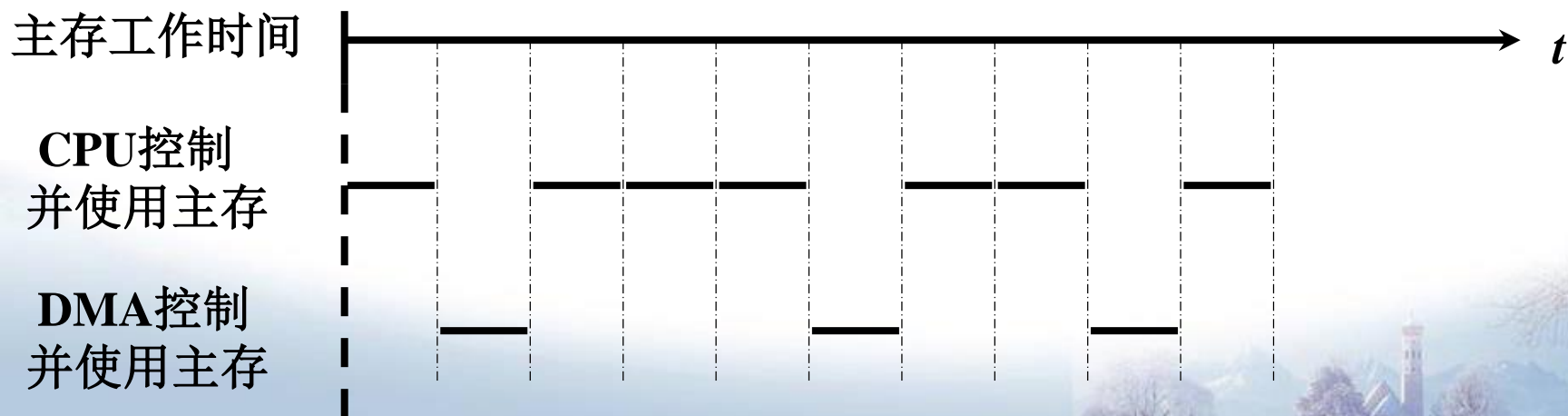
优点：控制简单

缺点：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）

每当I/O设备发出DMA请求时，I/O设备便挪用或窃取总线占用权一个或几个主存周期，DMA不请求时，CPU继续访问主存。



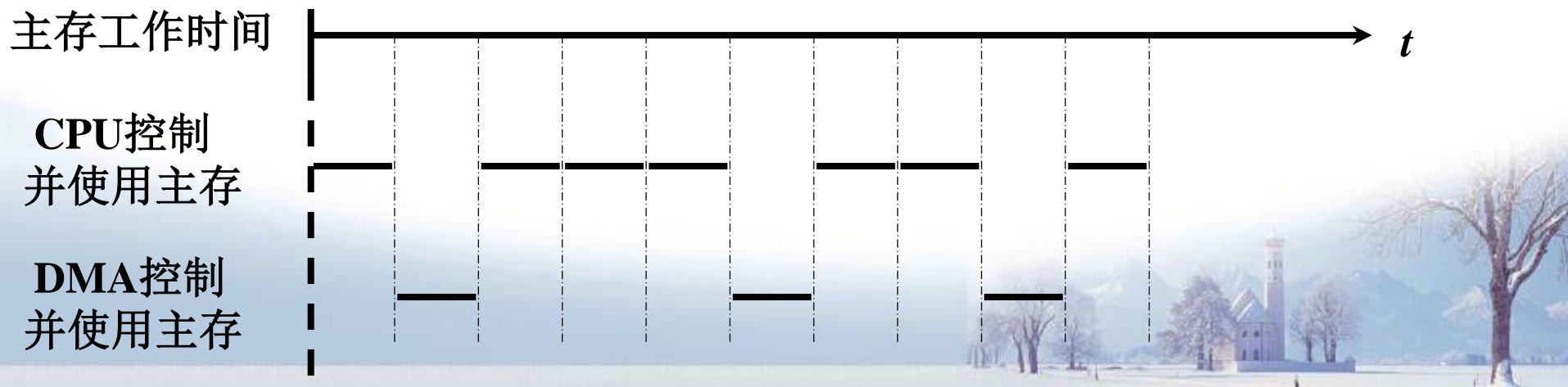
(2) 周期挪用 (或周期窃取)

DMA 访问主存有三种可能

- CPU 此时不访存
- CPU 正在访存，等存取周期结束CPU才能让出总线
- CPU 与 DMA 同时请求访存

此时 CPU 将总线控制权让给 DMA

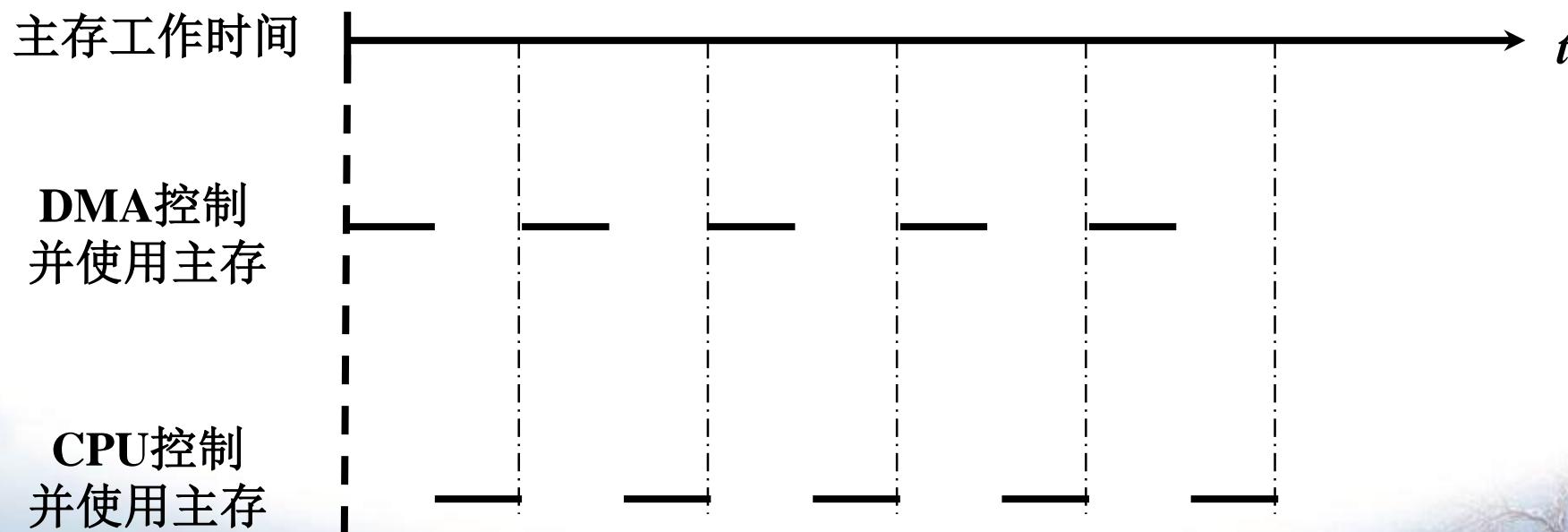
此方法适用于I/O设备读/写周期大于主存周期的情况。



(3) DMA 与 CPU 交替访问

适用于**CPU**工作周期大于主存存取周期的情况。

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



二、DMA 接口的功能和组成

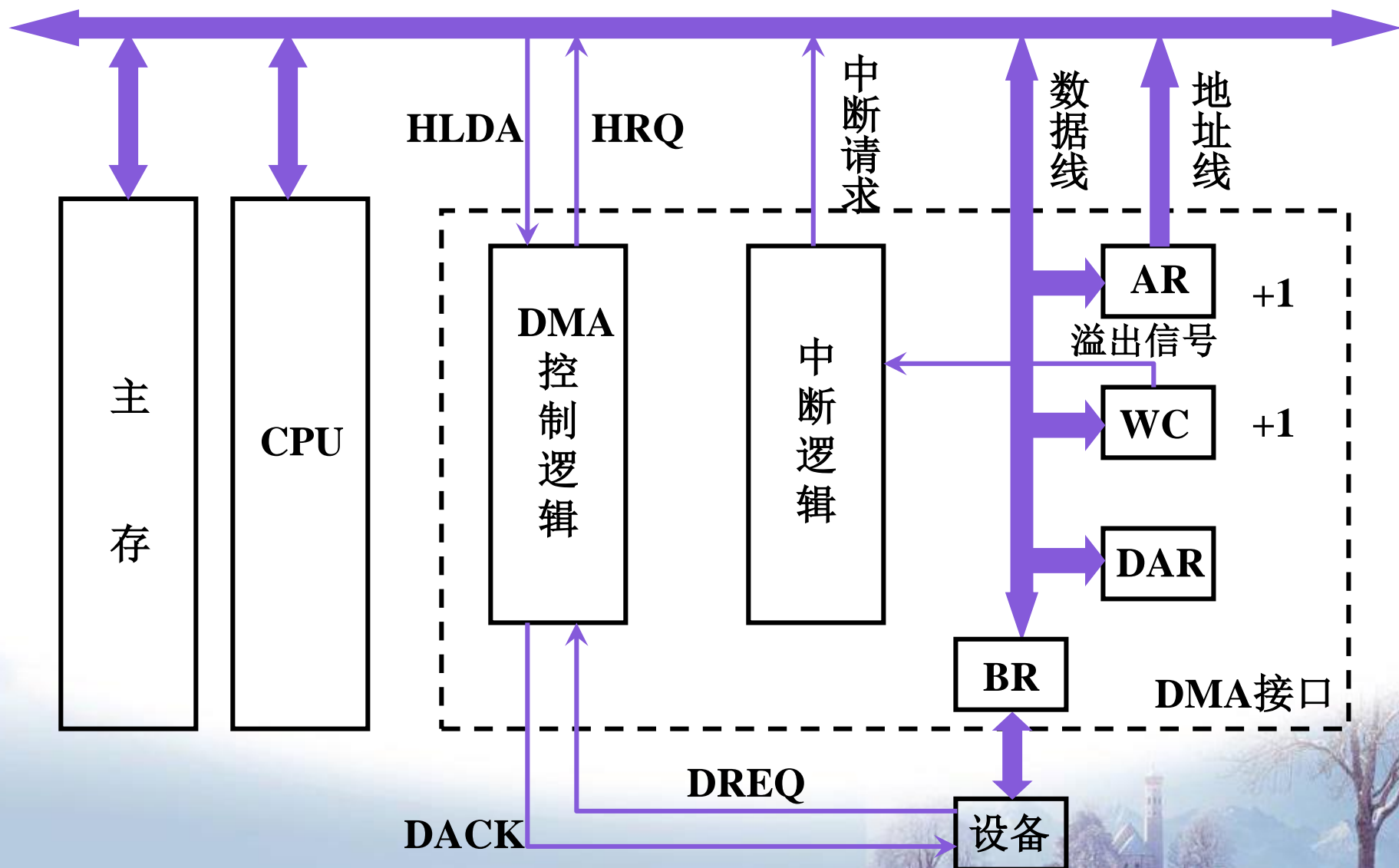
5.6

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数据长度
- (5) DMA 传送结束时， 给出操作完成信号

2. DMA 接口组成

存放 I/O 设备的设备码或表示设备信息存储区的寻址信息。在 I/O 设备转换的数据地址号，即盘面号和柱面号。存放每次传送的数据。有数据传送完毕。



2. DMA 接口组成

- DMAC (Direct Memory Access Controller) 是指直接内存访问控制器。
- 数据传送方式中，它是实现地址的修改与传送字节数技术的主要功能部件。

三、DMA 的工作过程

5.6

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 传送过程示意图

5.6

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

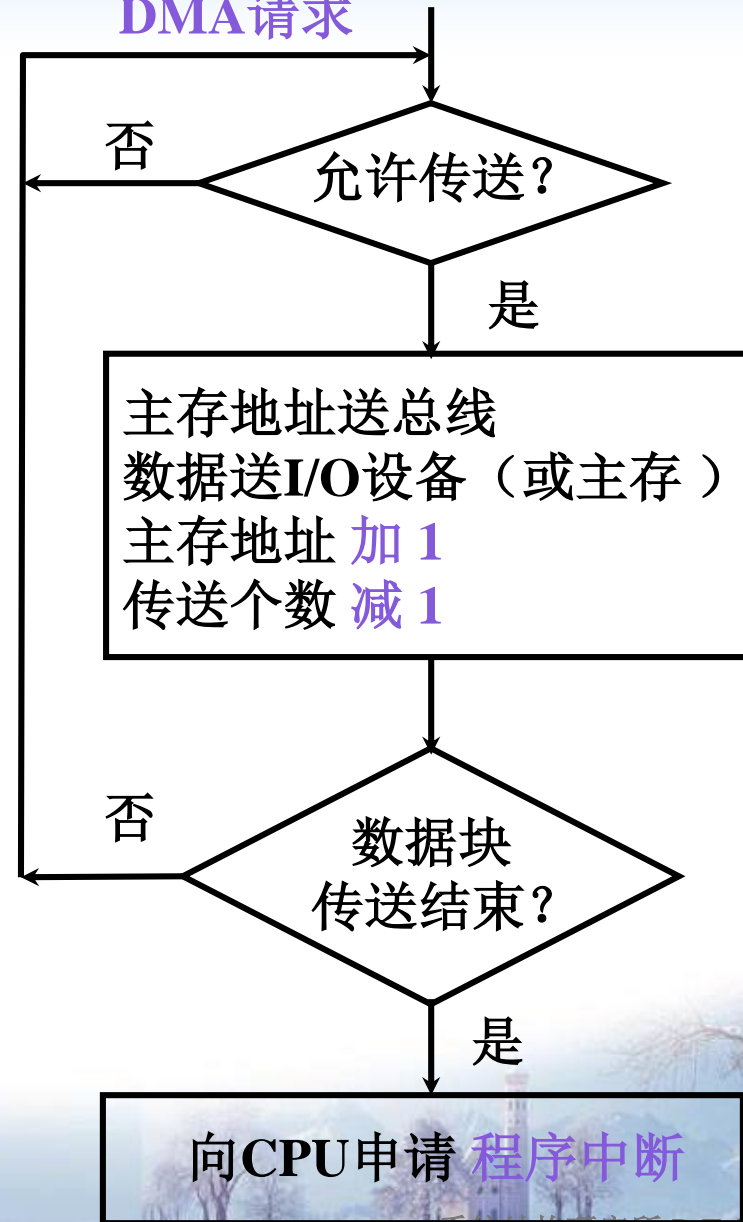
后处理:

中断服务程序
做 DMA 结束处理

继续执行主程序

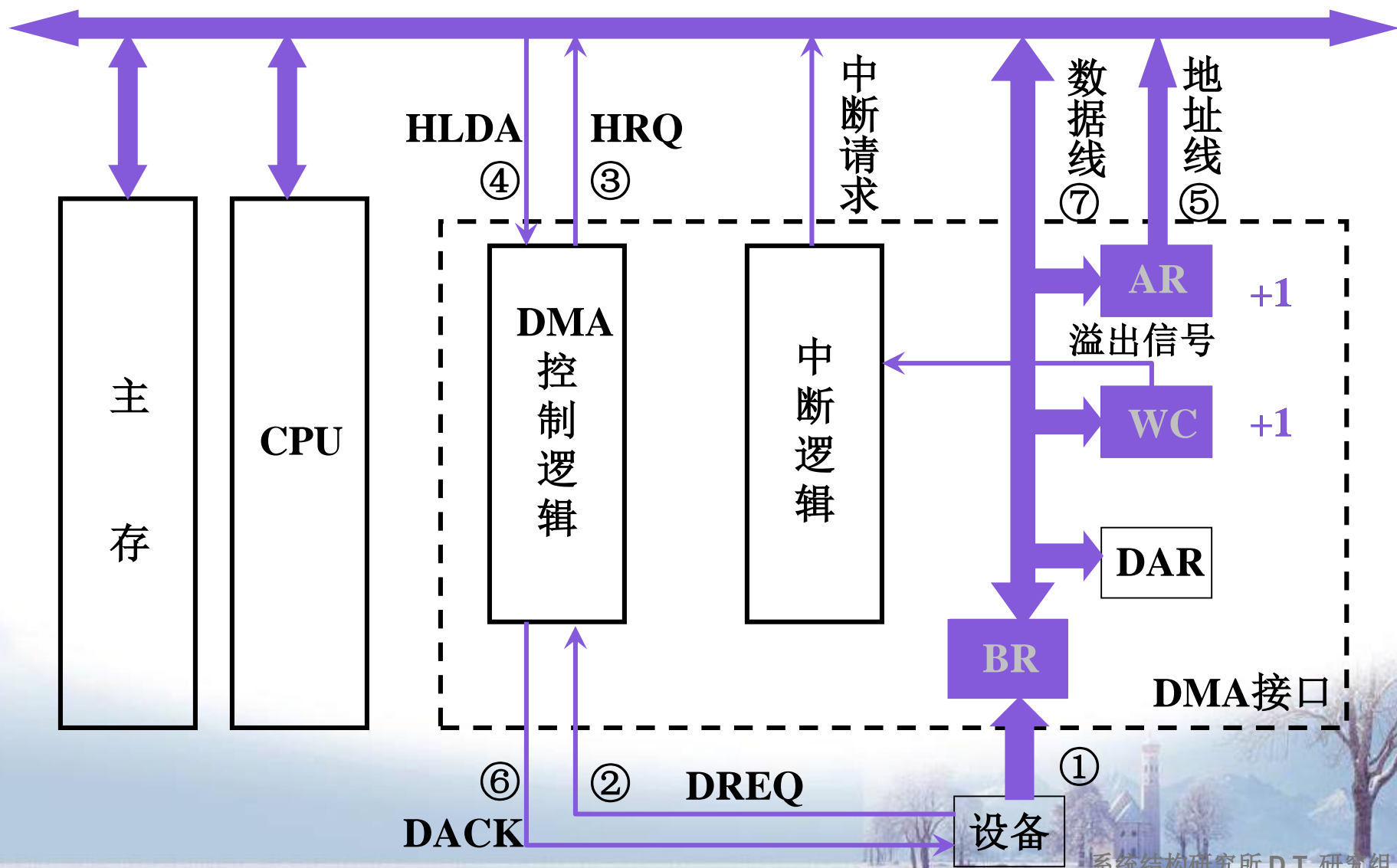
数据传送

DMA请求



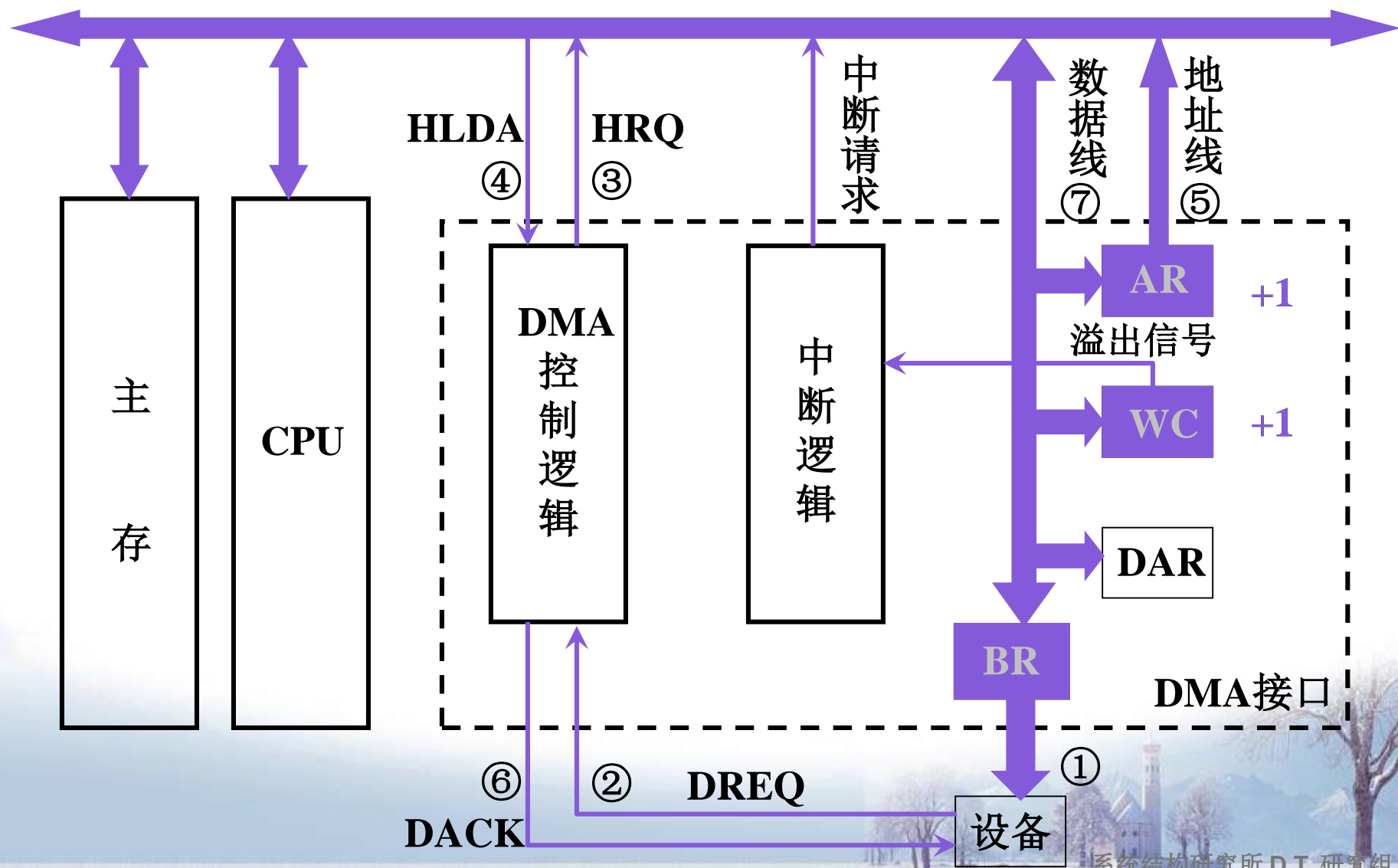
(3) 数据传送过程（输入）

5.6



(4) 数据传送过程（输出）

5.6



(5) 后处理

校验送入主存的数是否正确

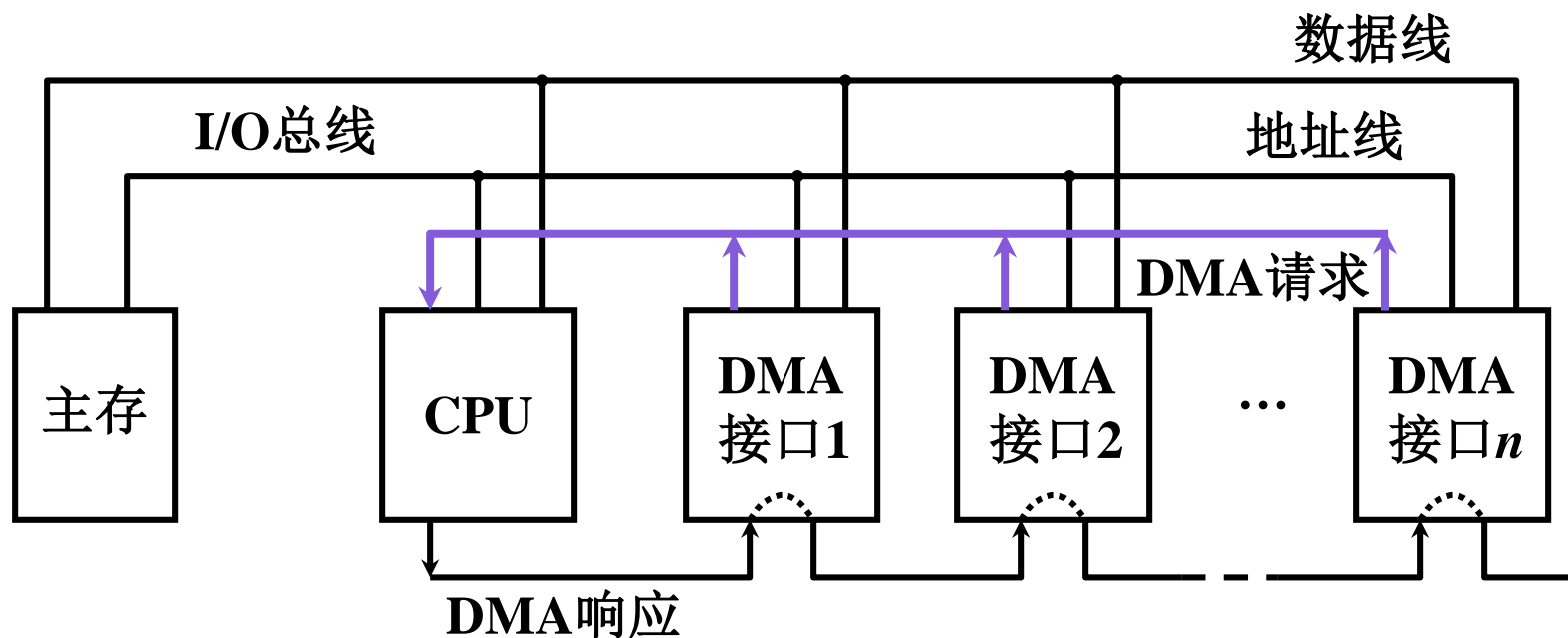
是否继续用 **DMA**

测试传送过程是否正确，错则转诊断程序

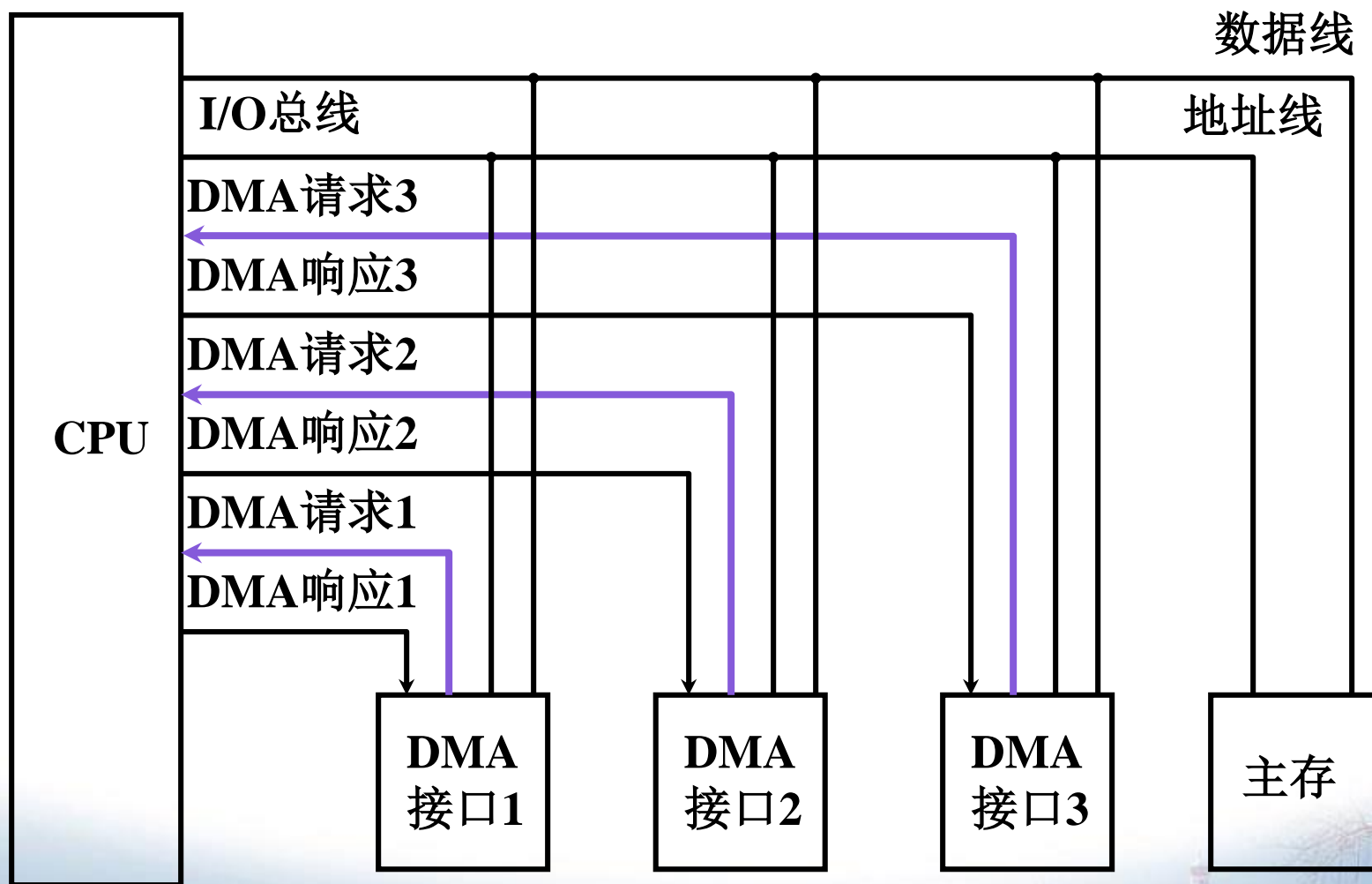
由中断服务程序完成

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



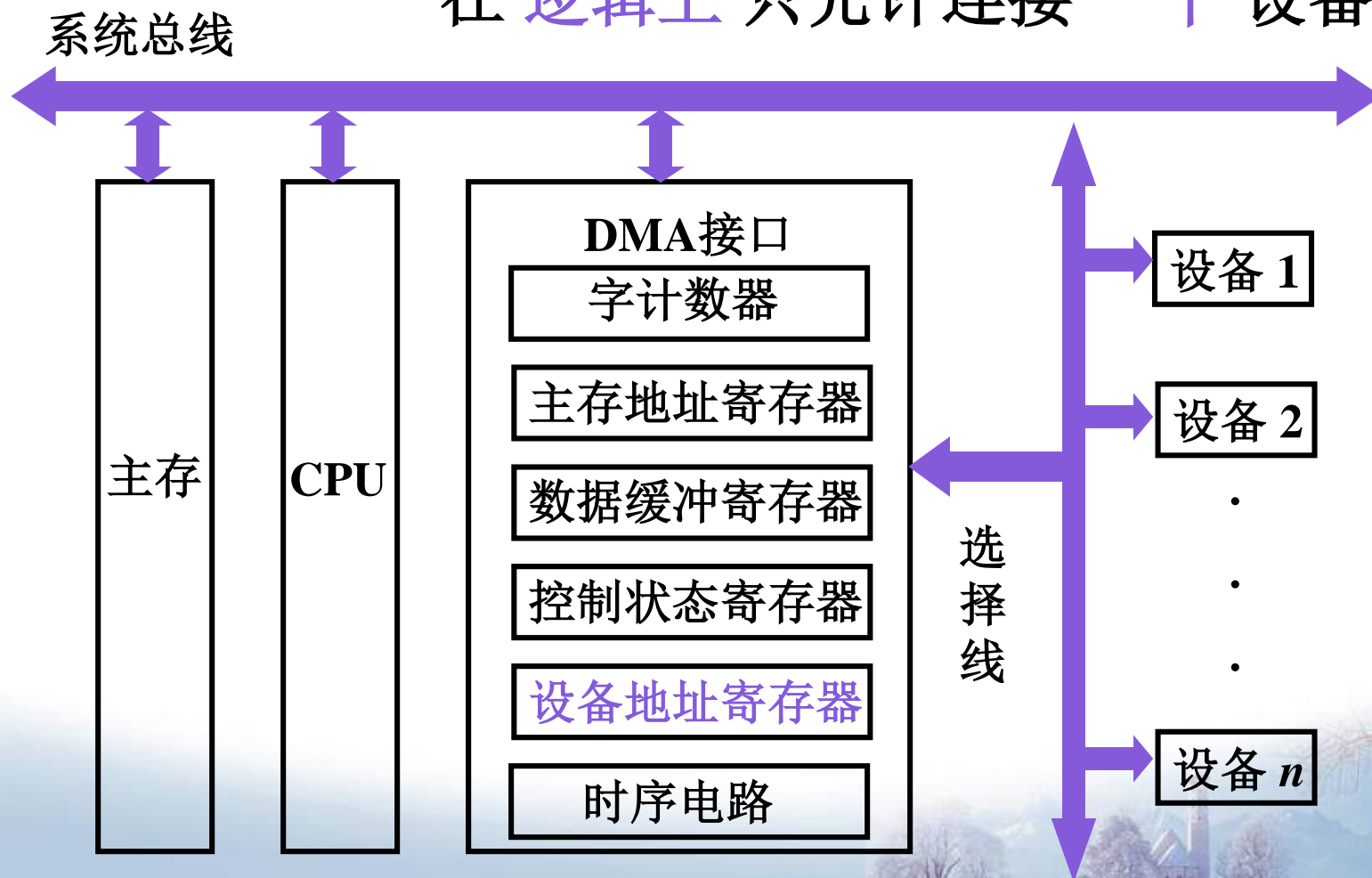
3. DMA 方式与程序中断方式的比较 5.6

	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

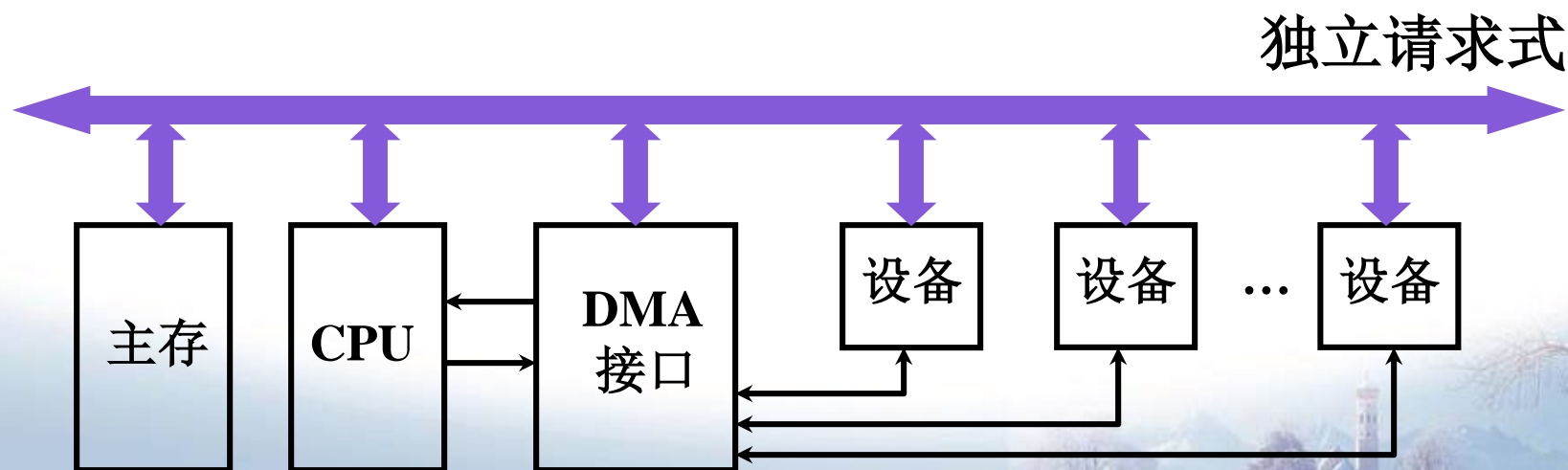
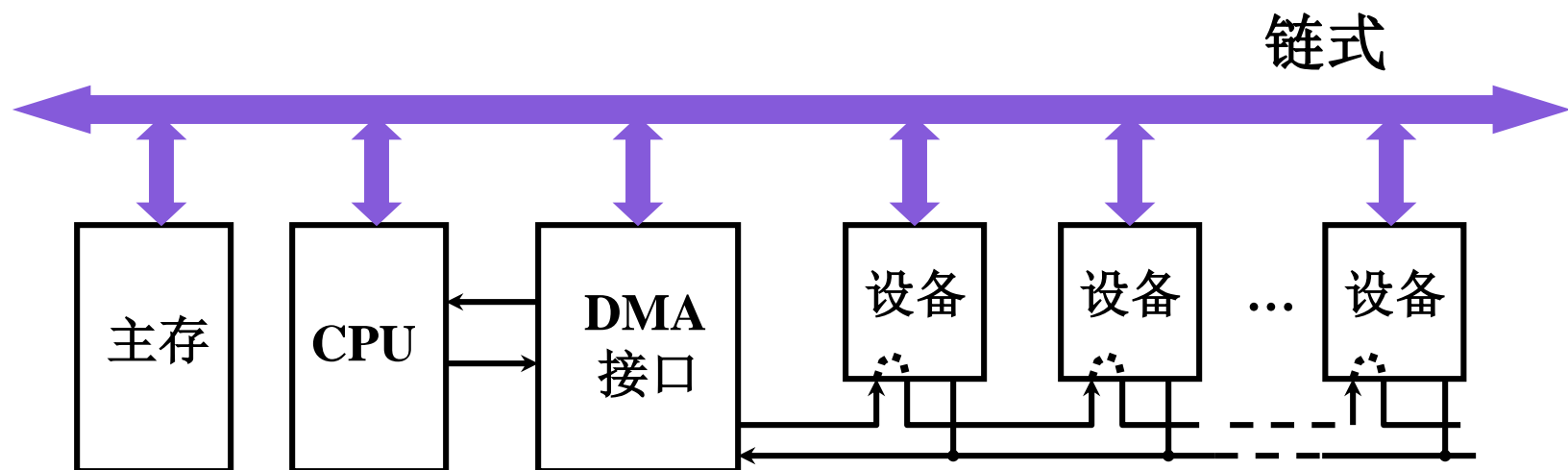
1. 选择型

在物理上连接多个设备
在逻辑上只允许连接一个设备



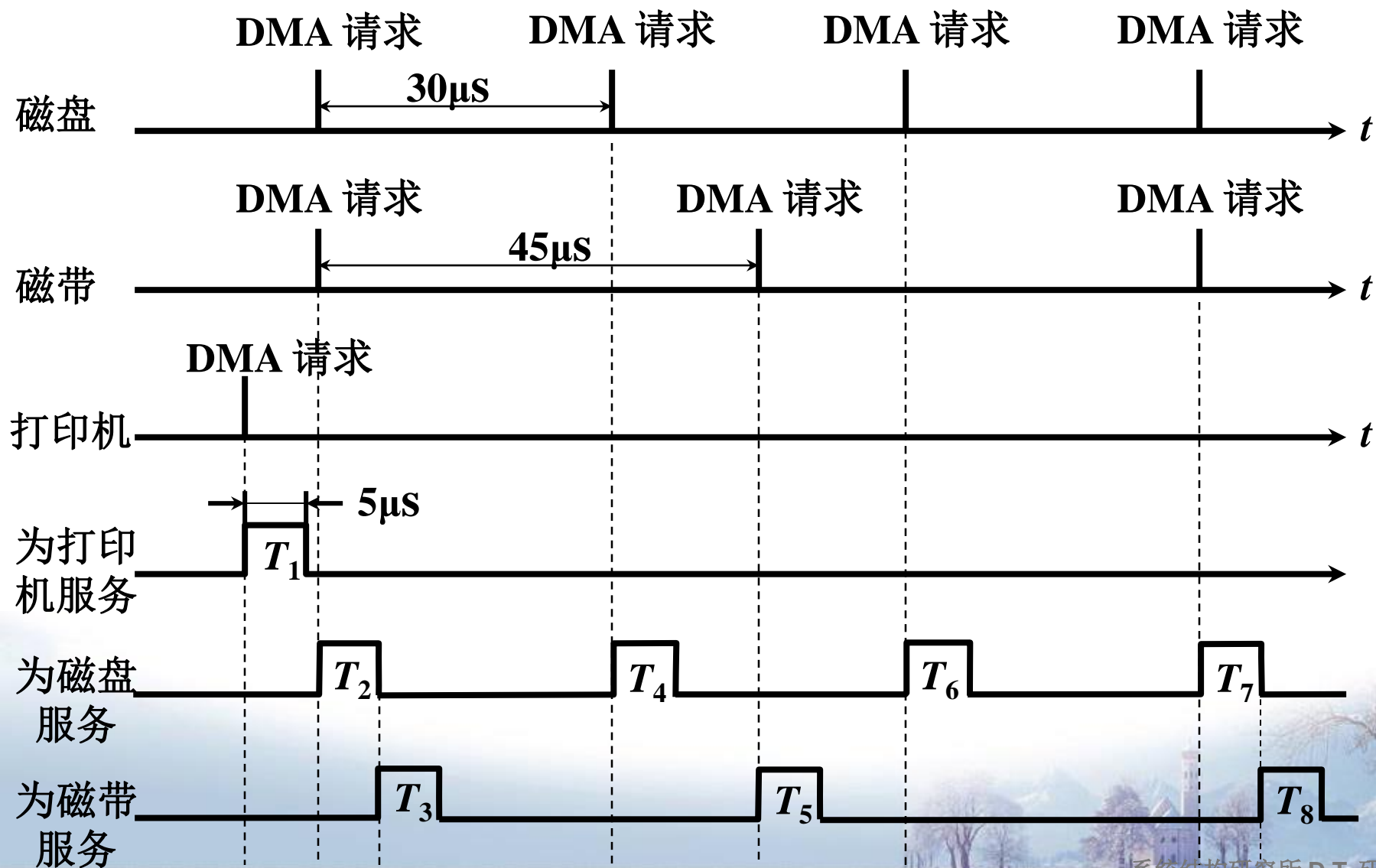
2. 多路型

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

5.6



Thank You !

