

计算机组织与体系结构

第二十四讲

计算机科学与技术学院

张展

第九章 输入输出系统

第9章 输入输出系统

9.1 概述

9.2 外部设备

9.3 I/O接口

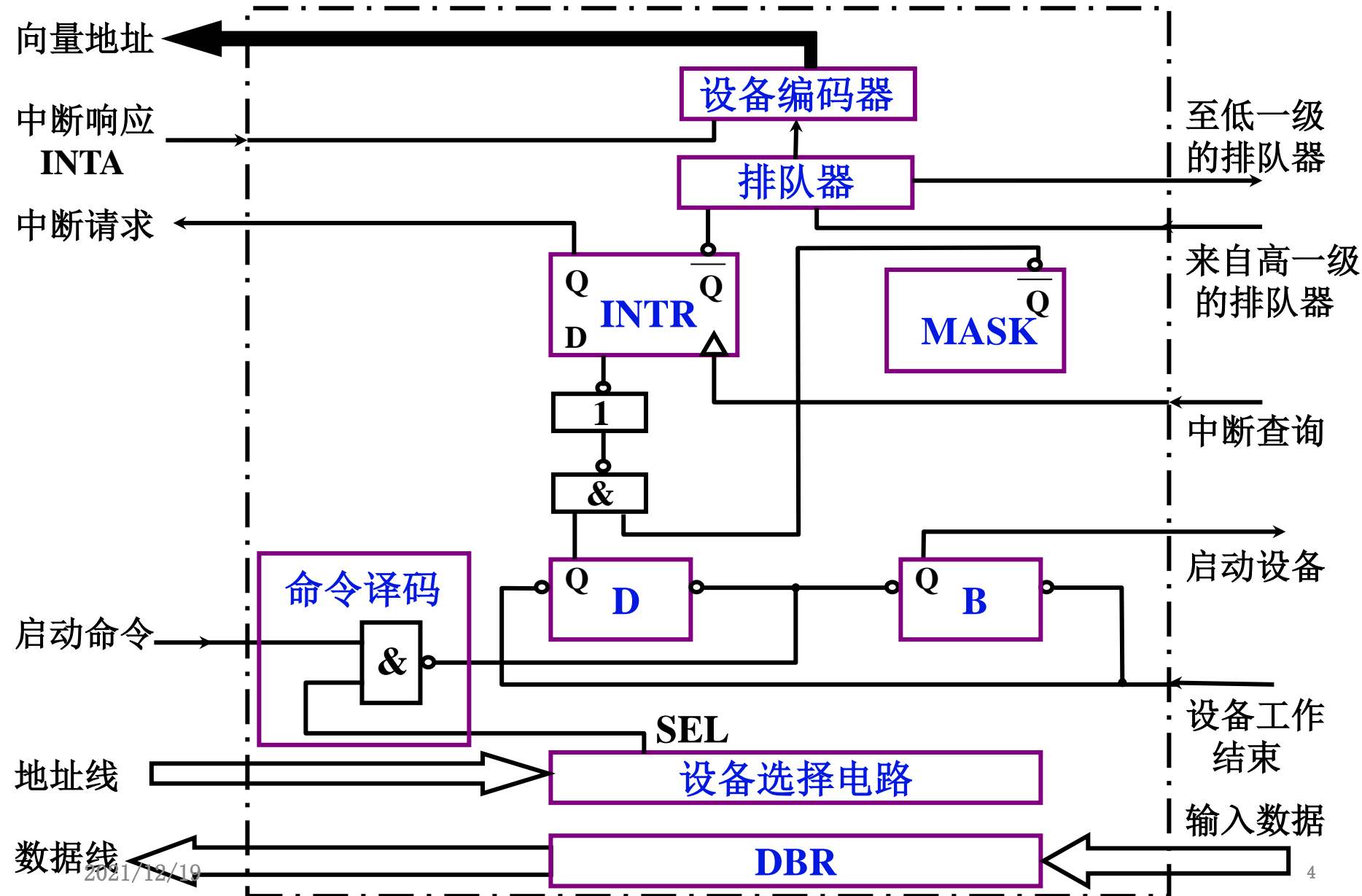
9.4 程序查询方式

9.5 程序中断方式

9.6 DMA方式

三、程序中中断方式的接口电路

9.5

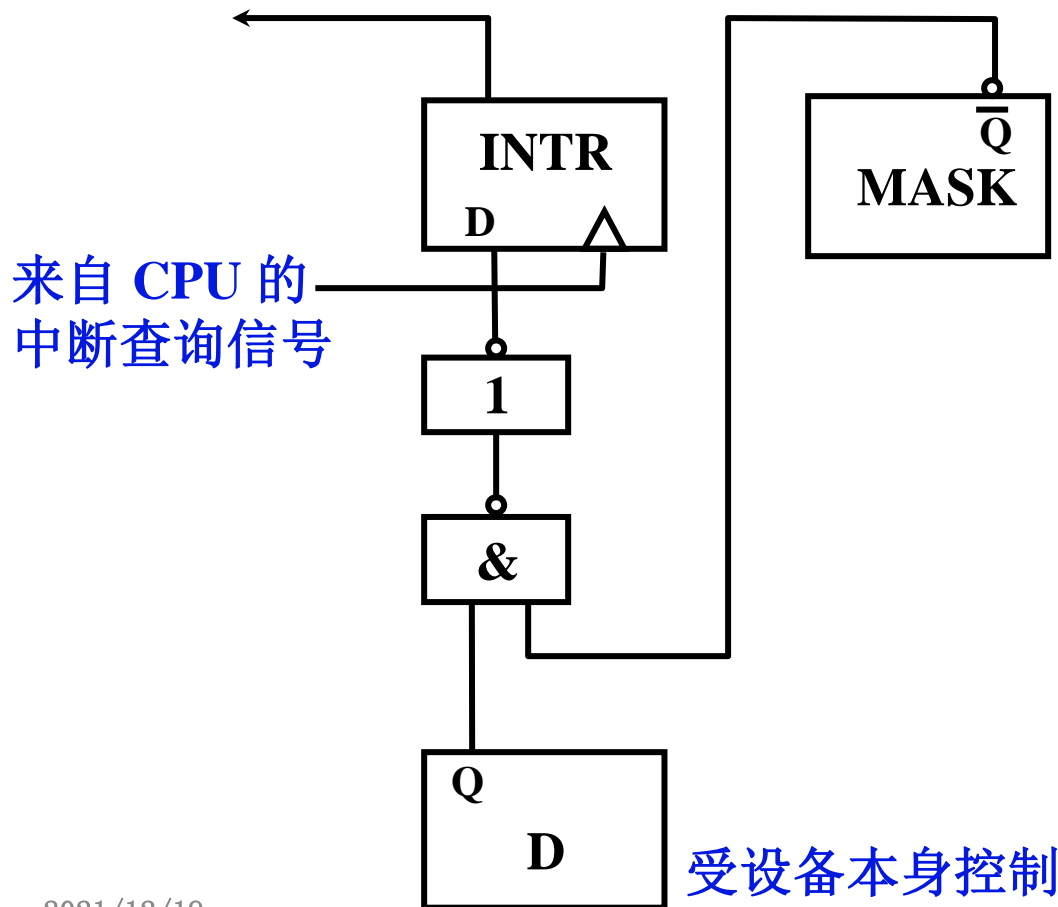


三、程序中中断方式的接口电路

9.5

1. 配置中断请求触发器和中断屏蔽触发器

中断请求



INTR

中断请求触发器

INTR = 1 有请求

MASK

中断屏蔽触发器

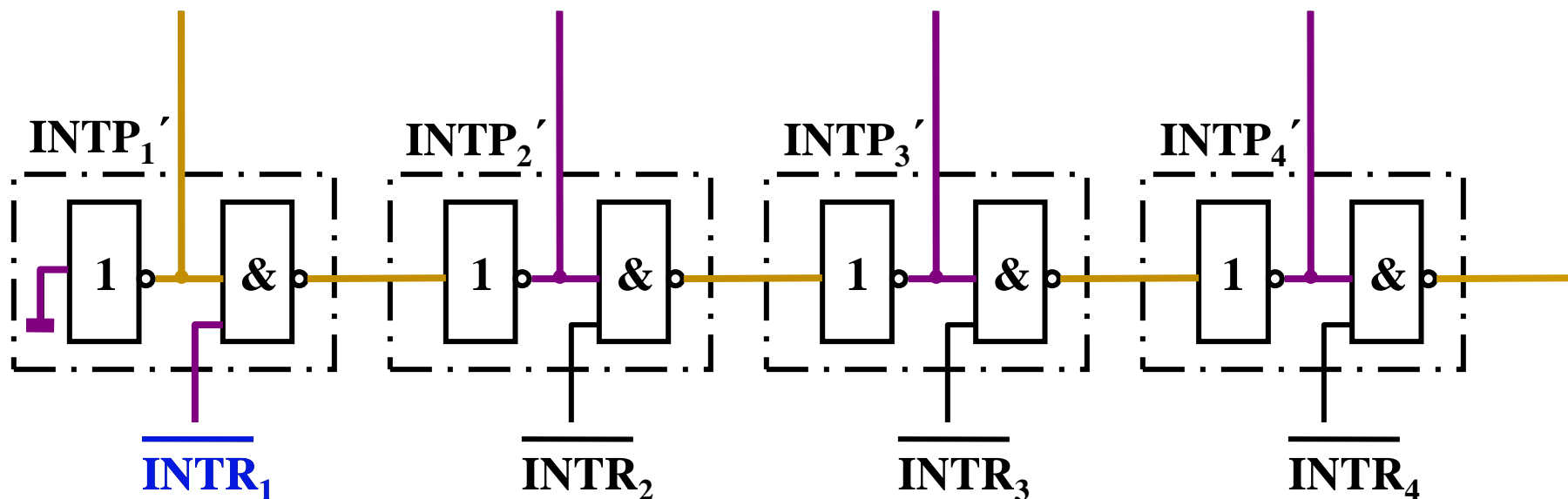
MASK = 1 被屏蔽

D 完成触发器

2. 排队器

9.5

排队 { 硬件 在 CPU 内或在接口电路中（链式排队器）
软件 程序查询方式（详见组成原理第8章）



设备 1[#]、2[#]、3[#]、4[#] 优先级按 降序排列

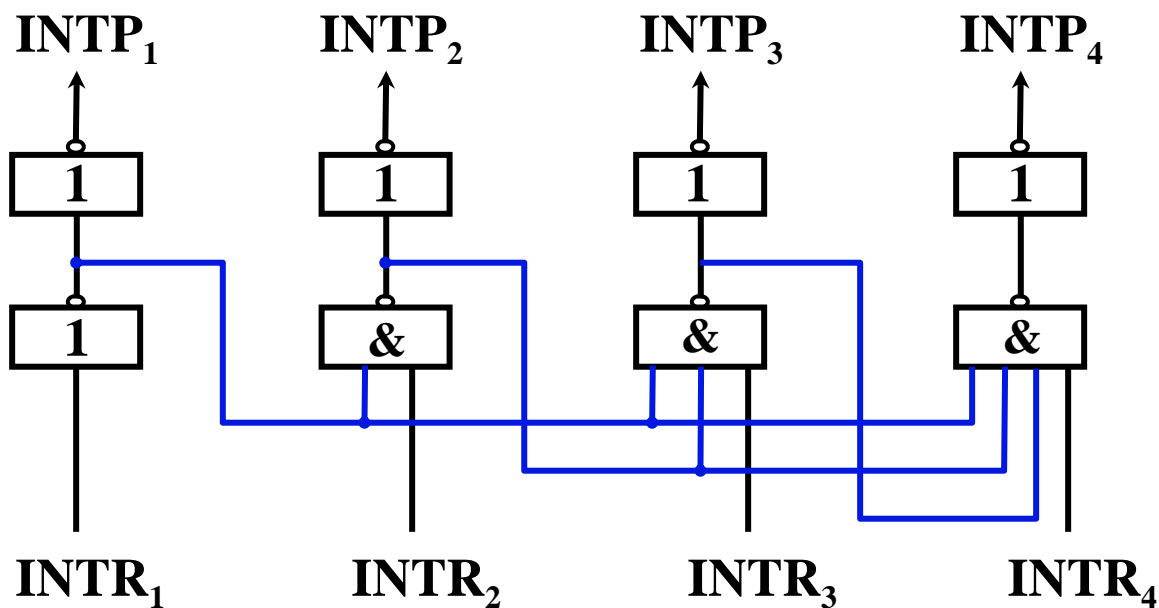
$\text{INTR}_i = 1$ 有请求 即 $\overline{\text{INTR}}_i = 0$

2.排队器

9.5

- ## ① 分散在各个中断源的接口电路中 链式排队器

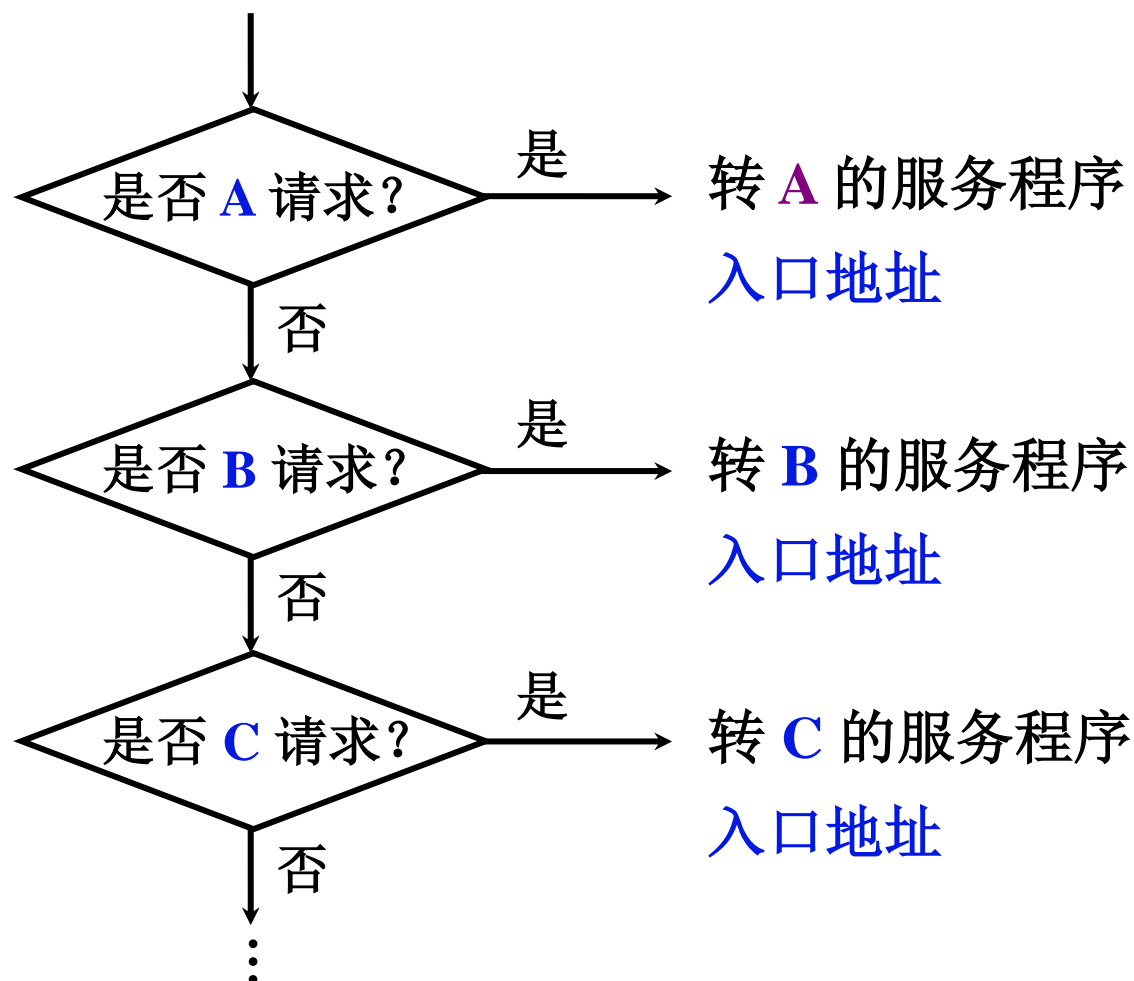
- ## ② 集中在 CPU 内



INTR₁、INTR₂、INTR₃、INTR₄ 优先级按降序排列

(2) 软件实现（程序查询）

A、B、C 优先级按 降序 排列



3. 中断向量地址形成部件

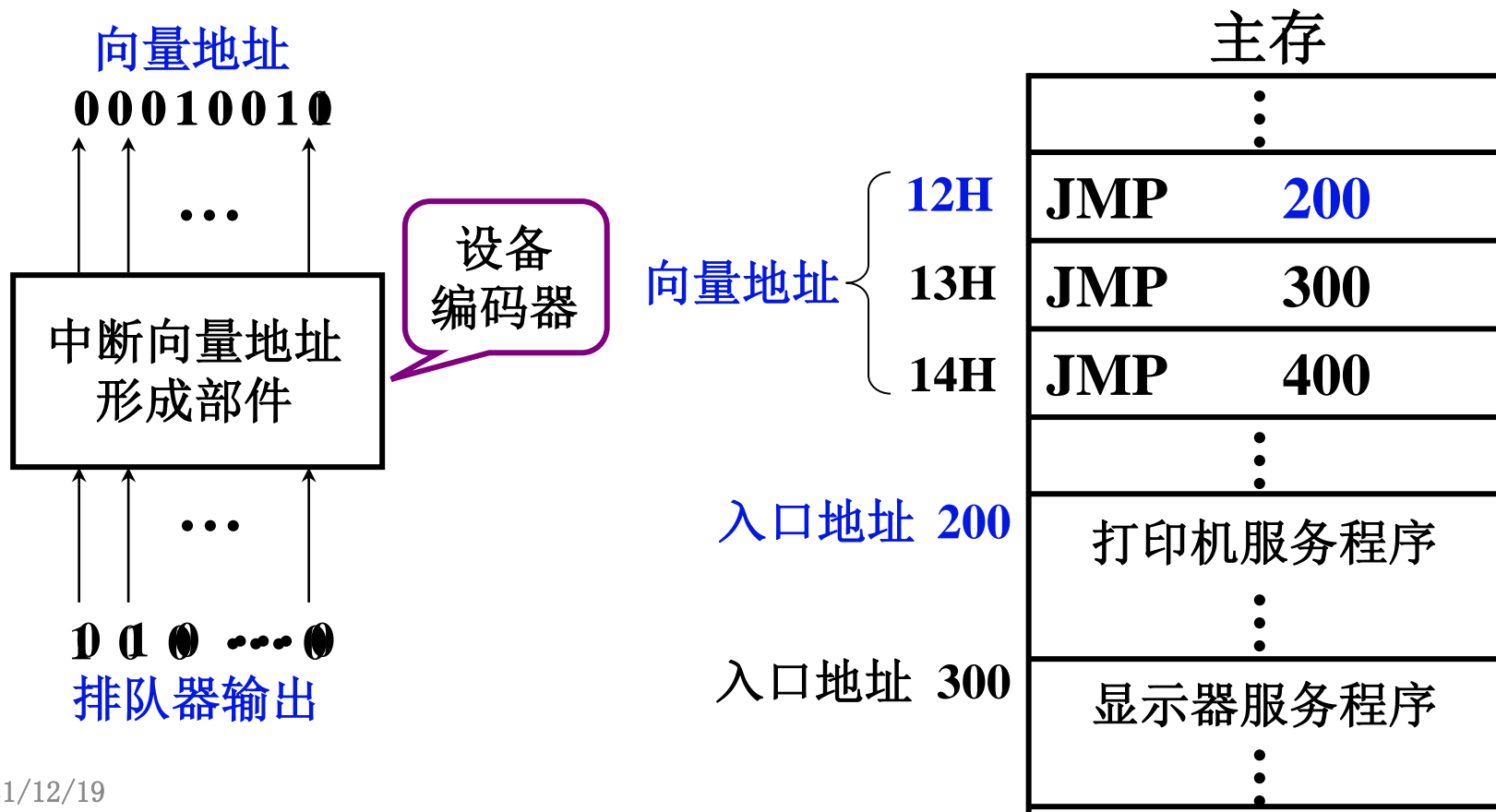
9.5

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**



四、I/O 中断处理过程

9.5

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

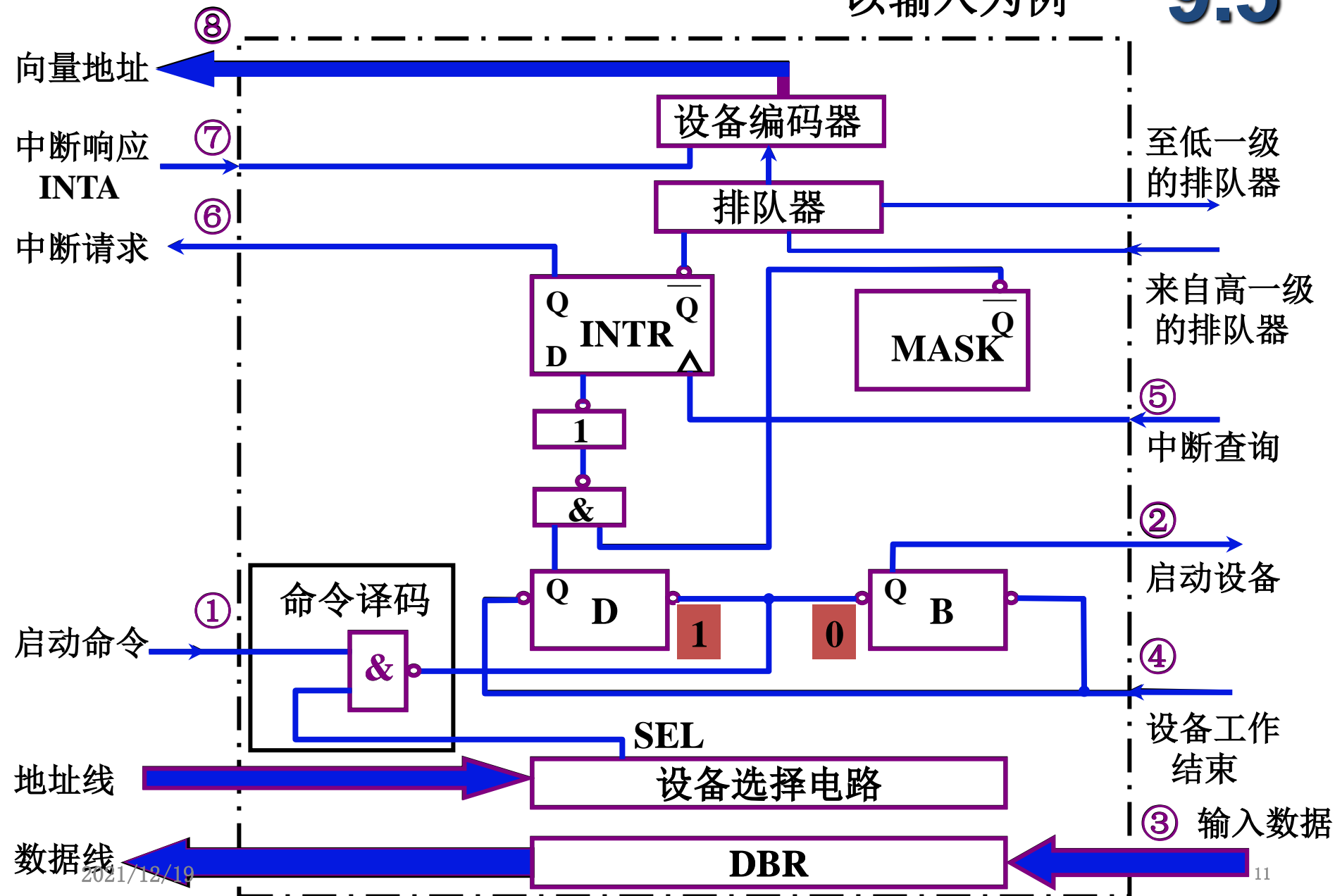
在每条指令执行阶段的结束前

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

9.5



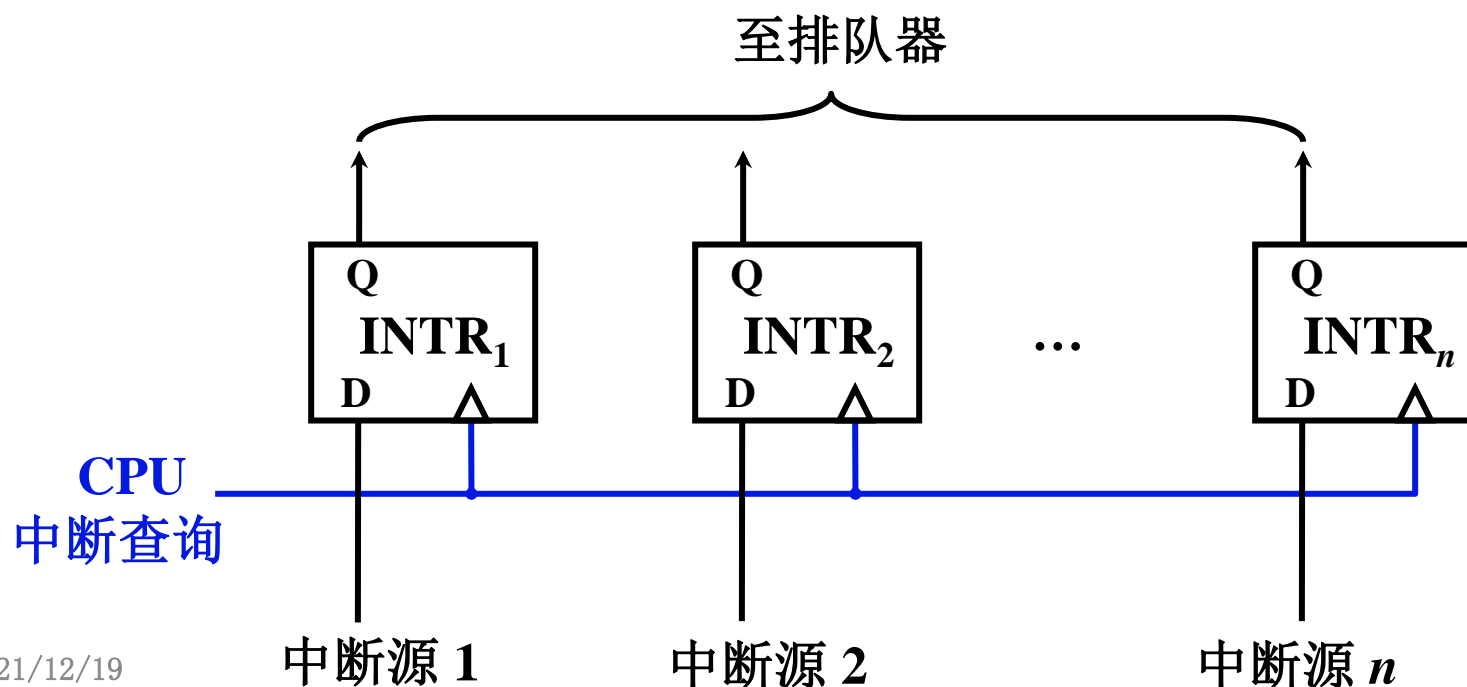
四、中断响应

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



五、中断服务程序流程

9.5

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

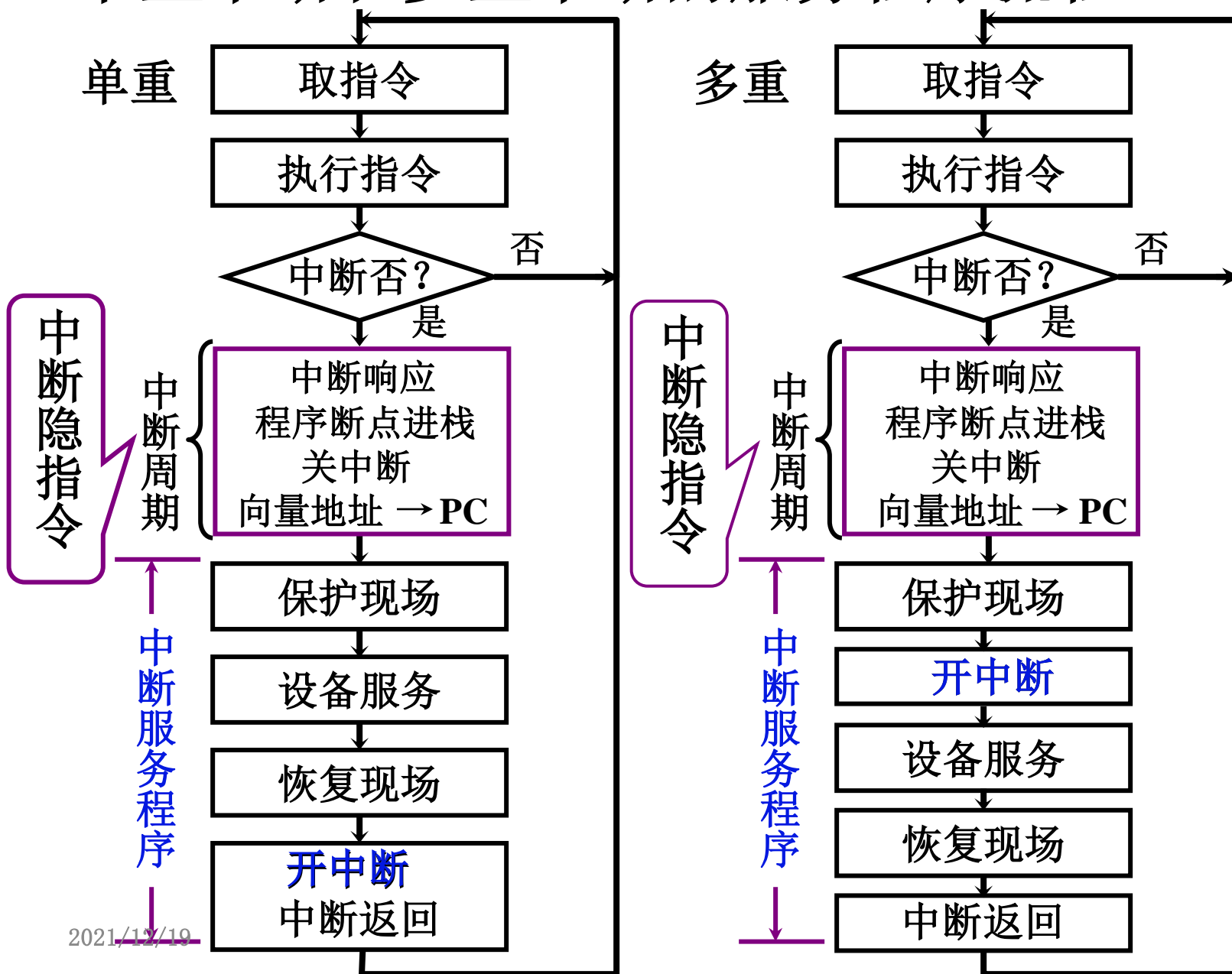
单重 中断 不允许中断 现行的 中断服务程序

多重 中断 允许级别更高 的中断源

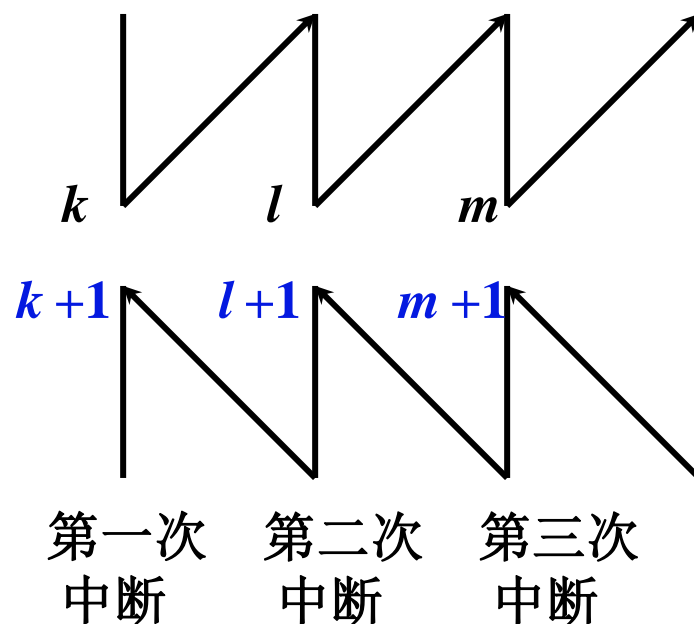
中断 现行的 中断服务程序

3. 单重中断和多重中断的服务程序流程

9.5



多重中断的概念



程序断点 $k+1$, $l+1$, $m+1$

多重中断的断点保护

9.5

(1) 断点进栈 中断隐指令 完成

(2) 断点存入 “0” 地址 中断隐指令 完成

中断周期 **0 → MAR**

命令存储器写

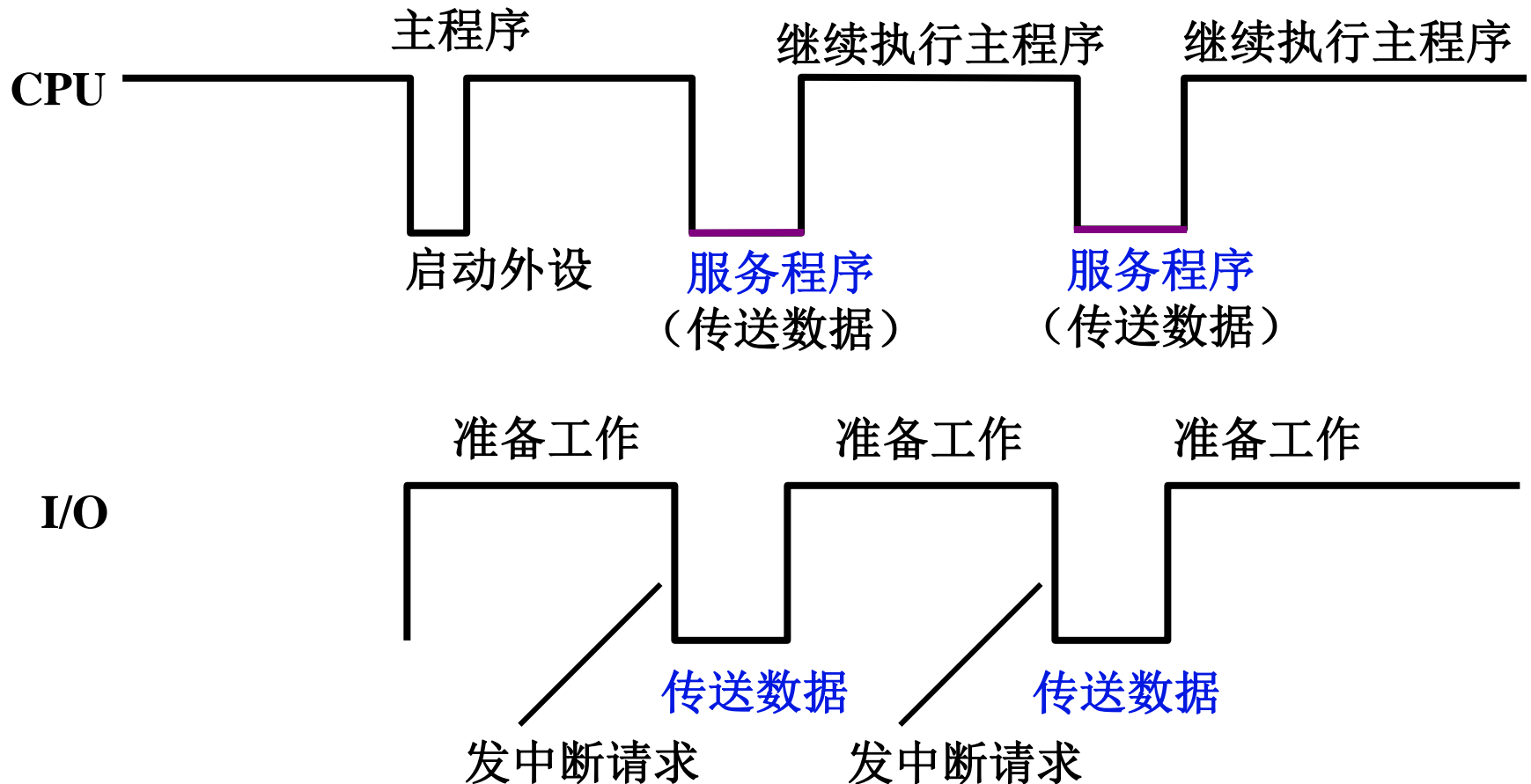
PC → MDR 断点 → MDR

(MDR) → 存入存储器

三次中断，三个断点都存入 “0” 地址

? 如何保证断点不丢失？

主程序和服务程序抢占 CPU 示意图 9.5



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

第9章 输入输出系统

9.1 概述

9.2 外部设备

9.3 I/O接口

9.4 程序查询方式

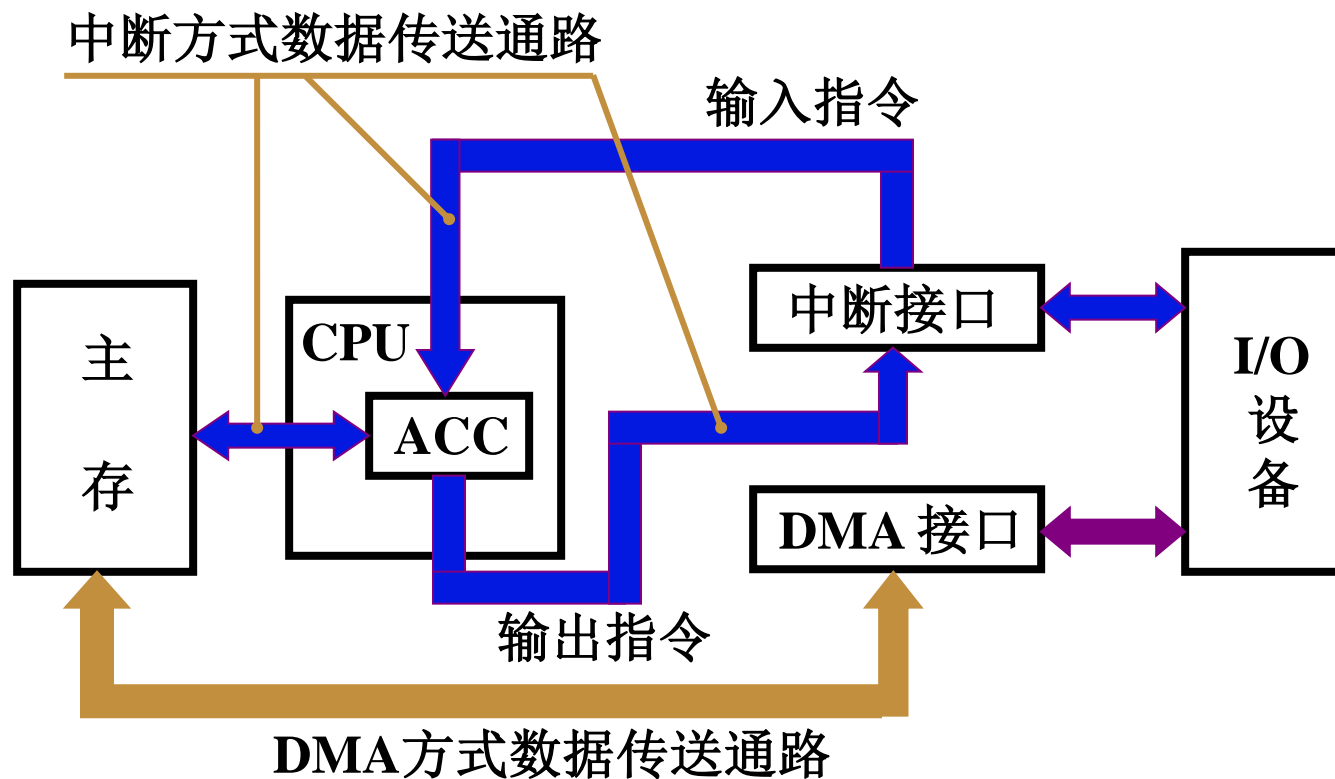
9.5 程序中断方式

9.6 DMA方式

9.6 DMA 方式

一、DMA 方式的特点

1. DMA 和程序中中断两种方式的数据通路



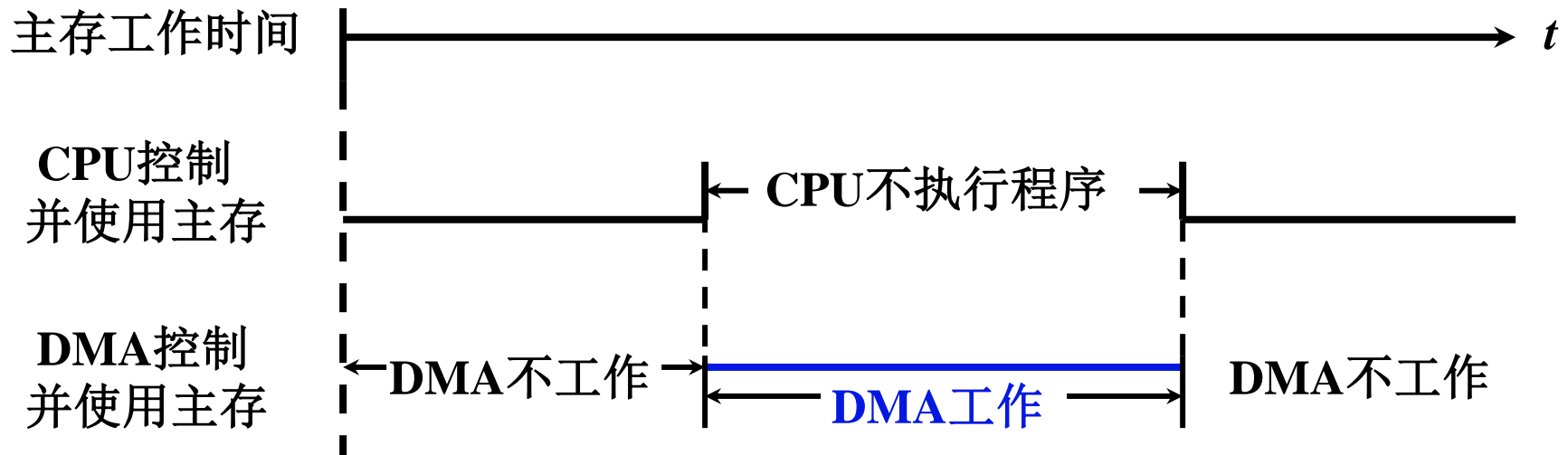
2. DMA 与主存交换数据的三种方式 9.6

(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



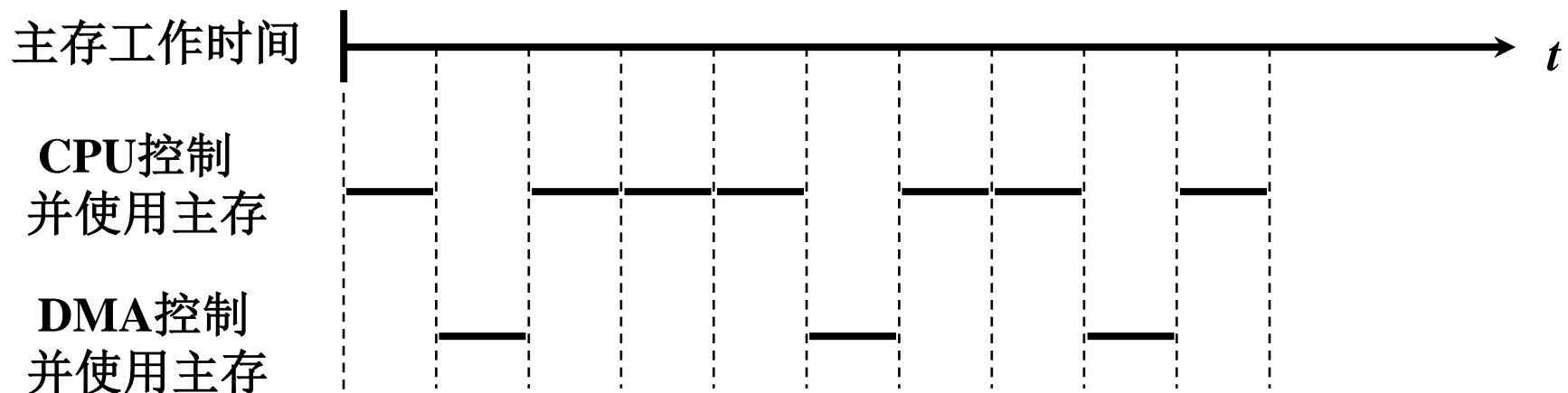
(2) 周期挪用（或周期窃取）

9.6

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

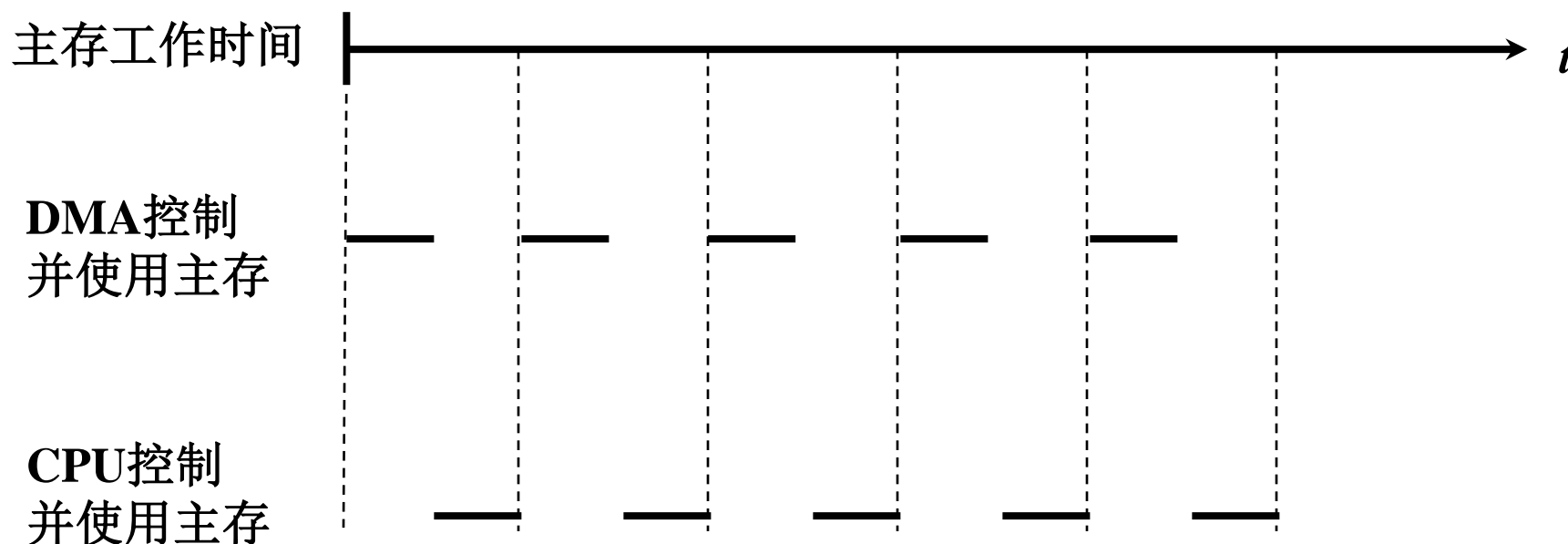
此时 CPU 将总线控制权让给 DMA



(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

二、DMA 接口的功能和组成

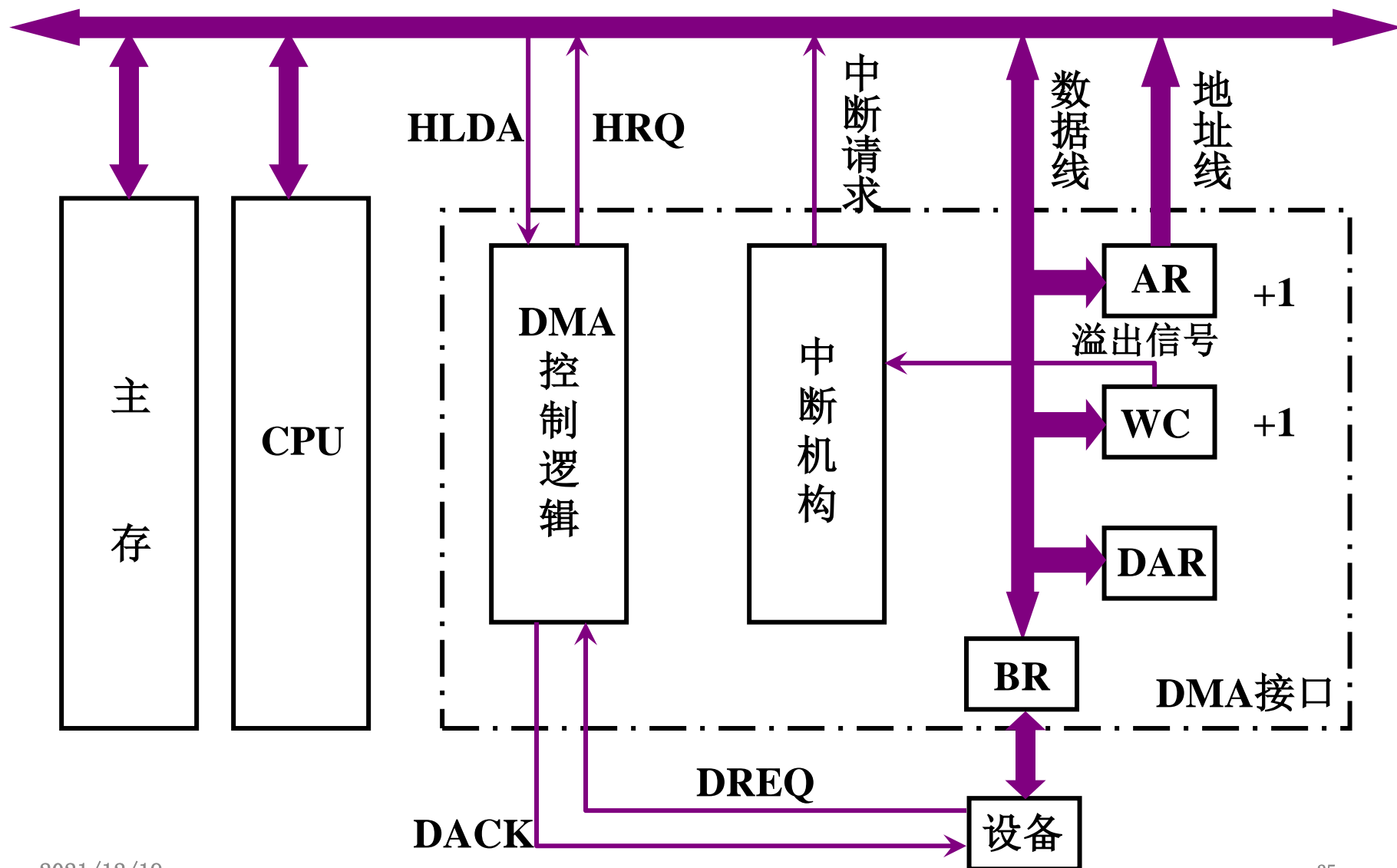
9.6

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 长度
- (5) DMA 传送结束时， 给出操作完成信号

2. DMA 接口组成

9.6



三、DMA 的工作过程

9.6

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

9.6

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

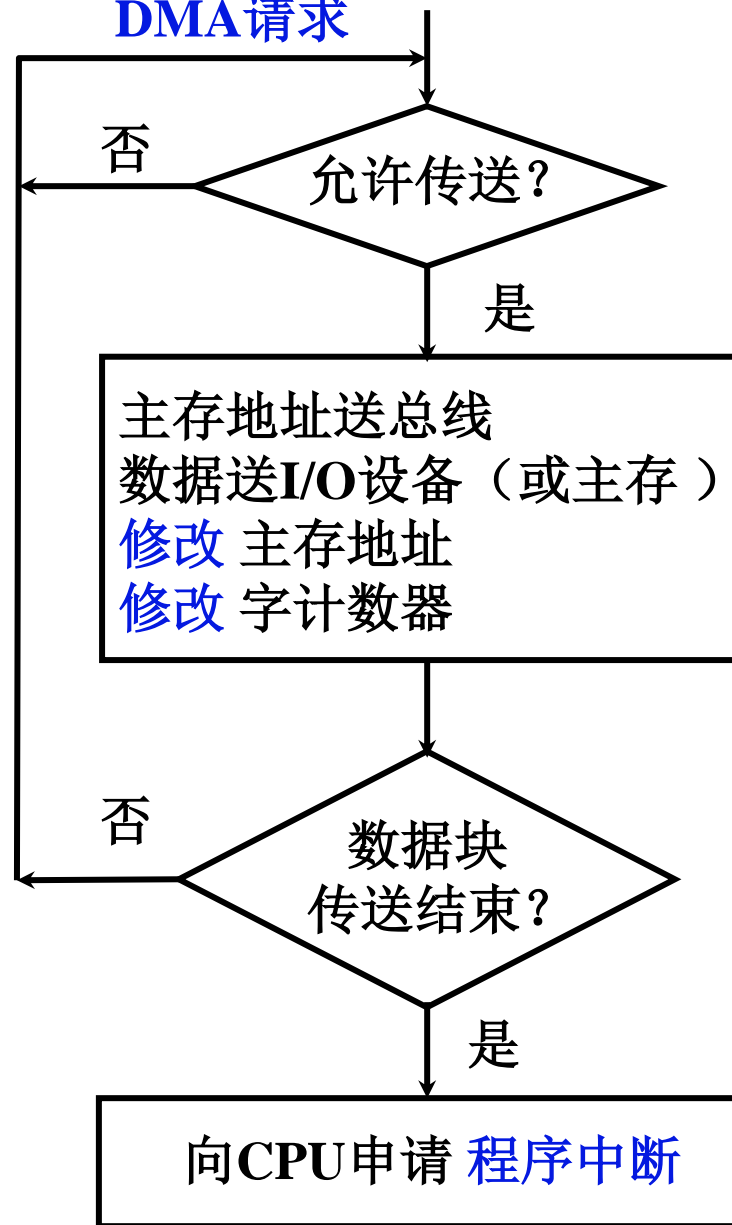
后处理:

中断服务程序
做 DMA 结束处理

继续执行主程序

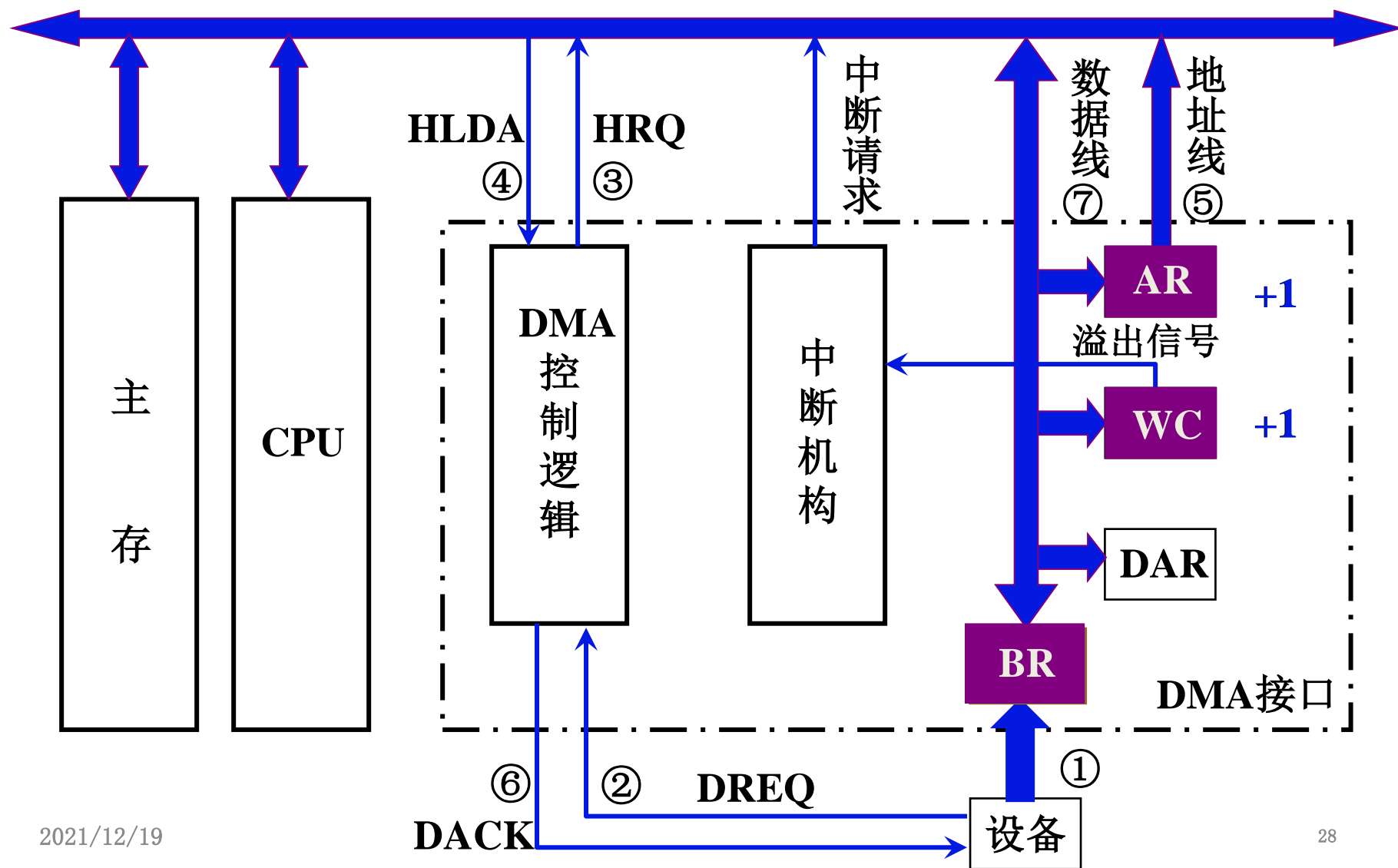
数据传送

DMA 请求



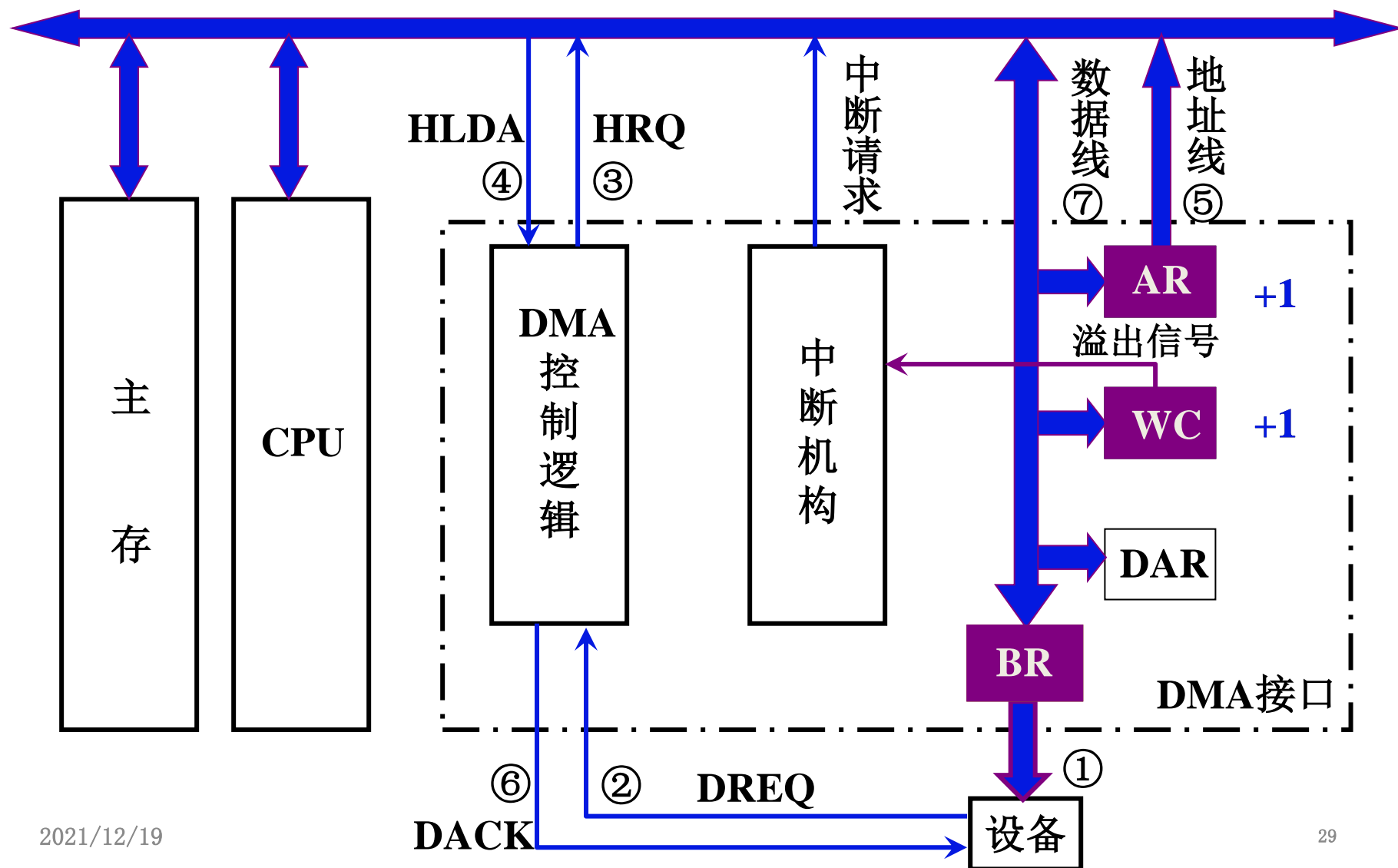
(3) 数据传送过程（输入）

9.6



(4) 数据传送过程（输出）

9.6



(5) 后处理

校验送入主存的数是否正确

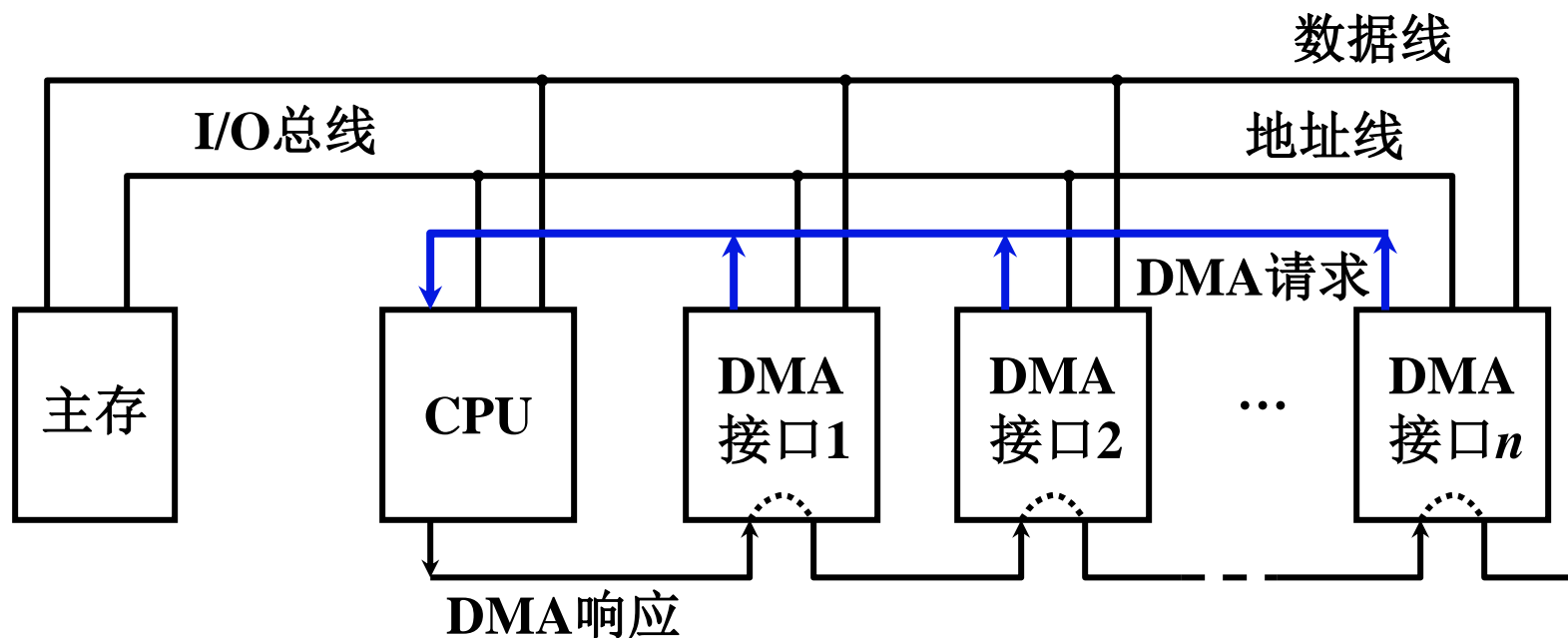
是否继续用 **DMA**

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

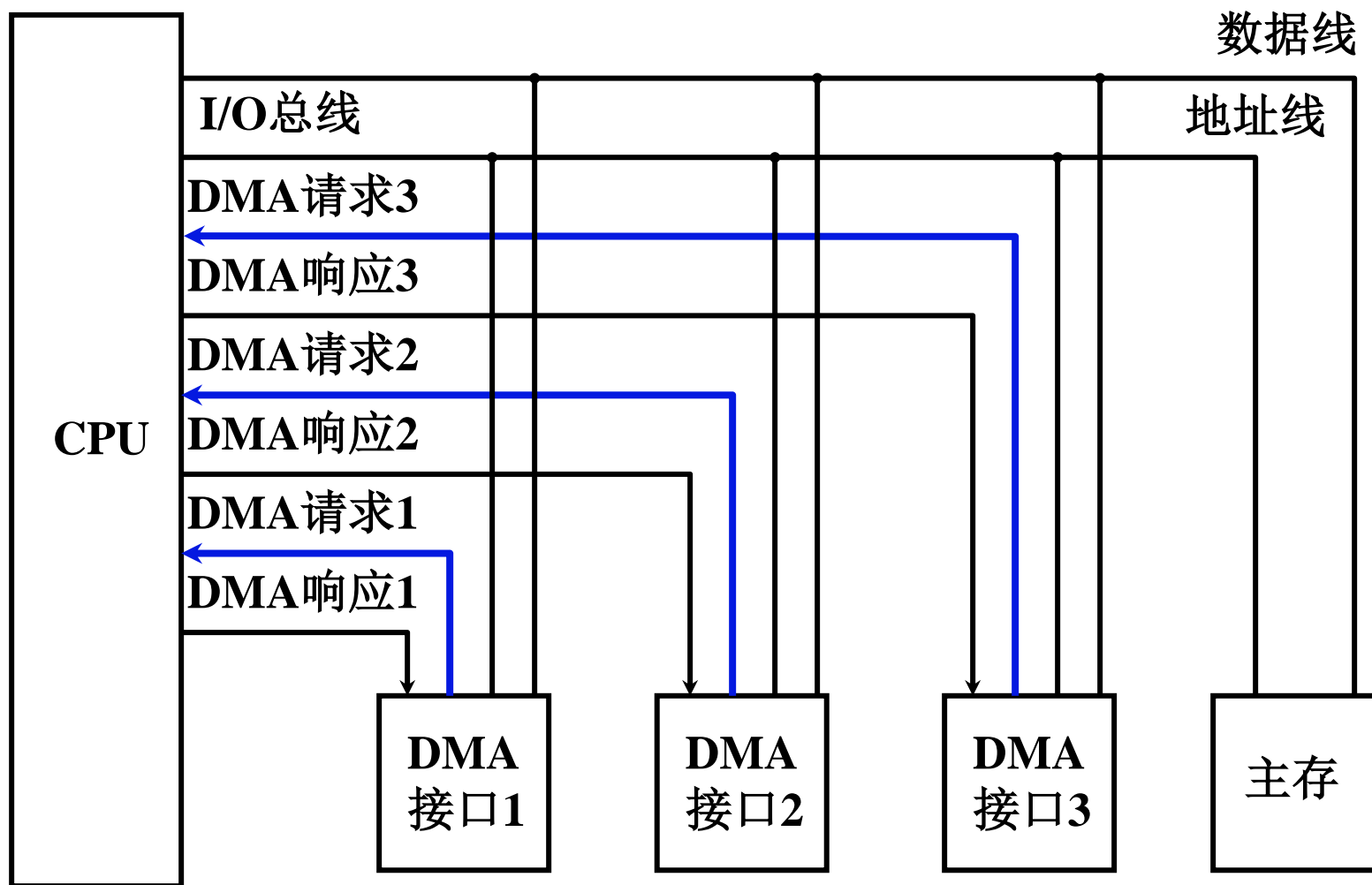
2. DMA 接口与系统的连接方式

(1) 具有公共请求线的 DMA 请求



(2) 独立的 DMA 请求

9.6



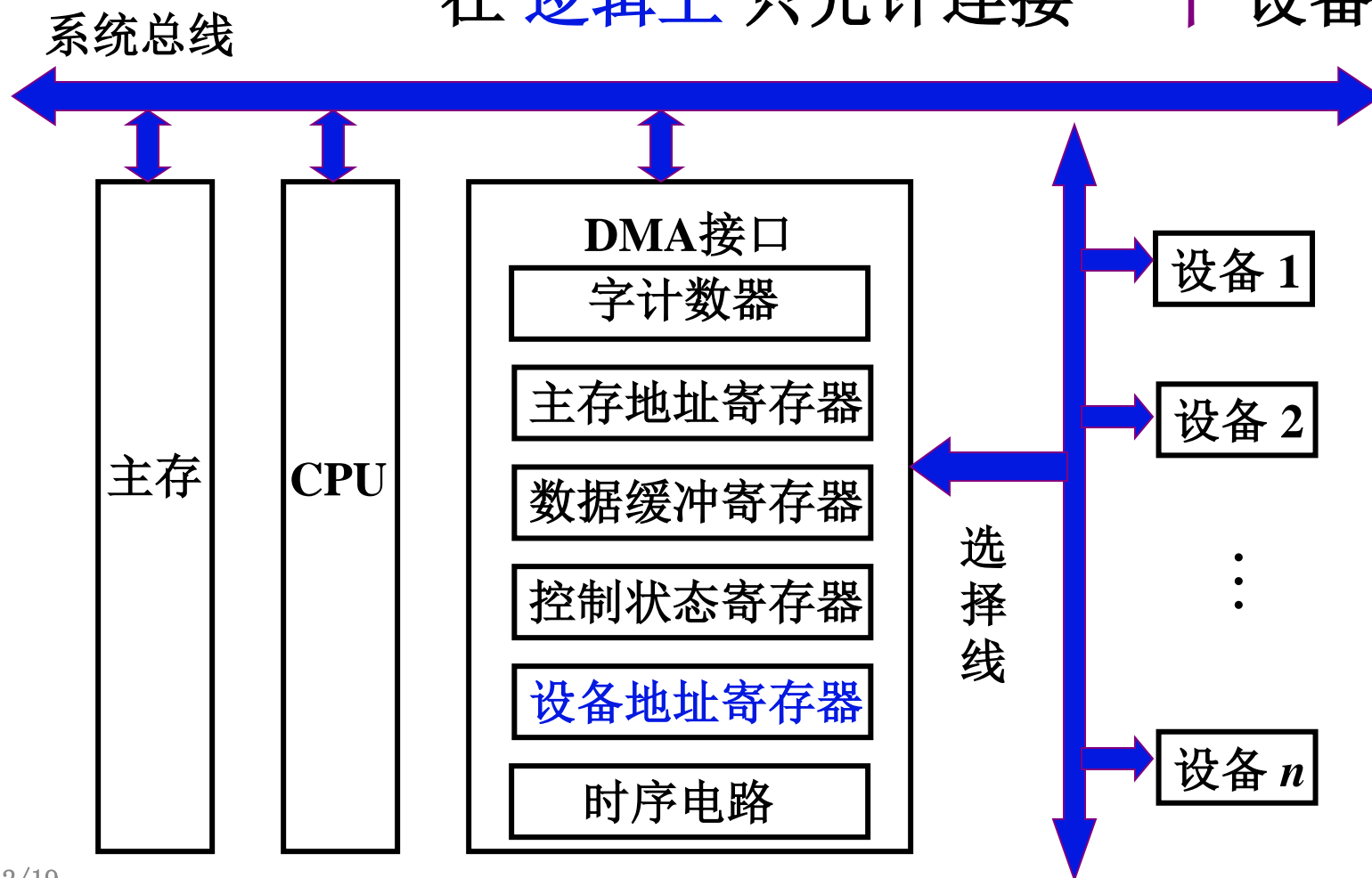
3. DMA 方式与程序中断方式的比较 9.6

	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

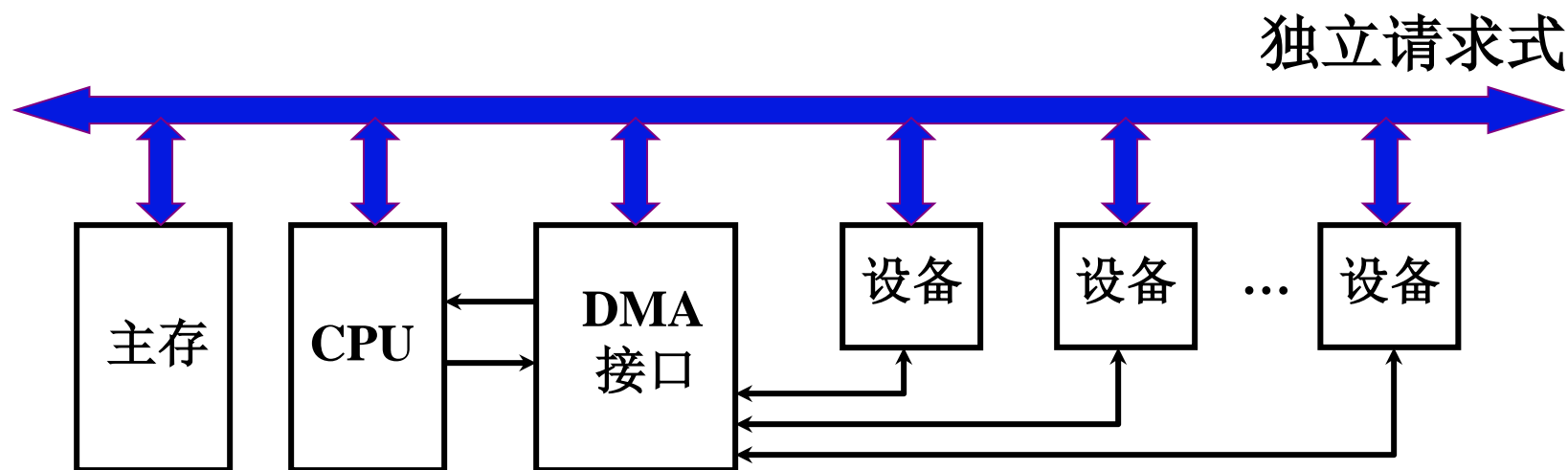
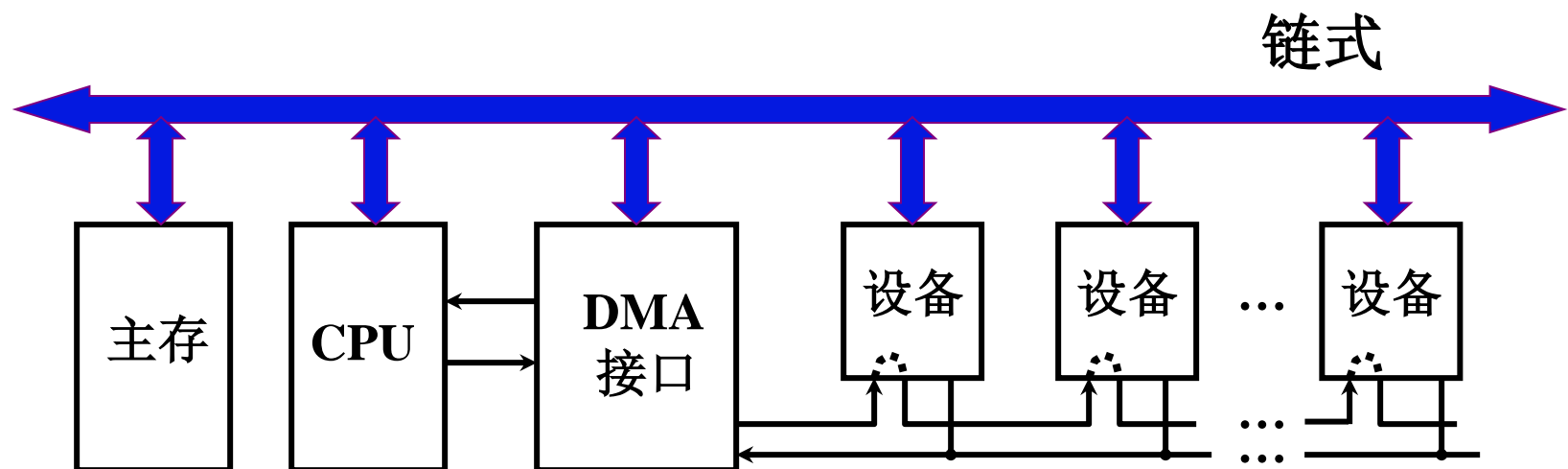
1. 选择型

在物理上连接多个设备
在逻辑上只允许连接一个设备



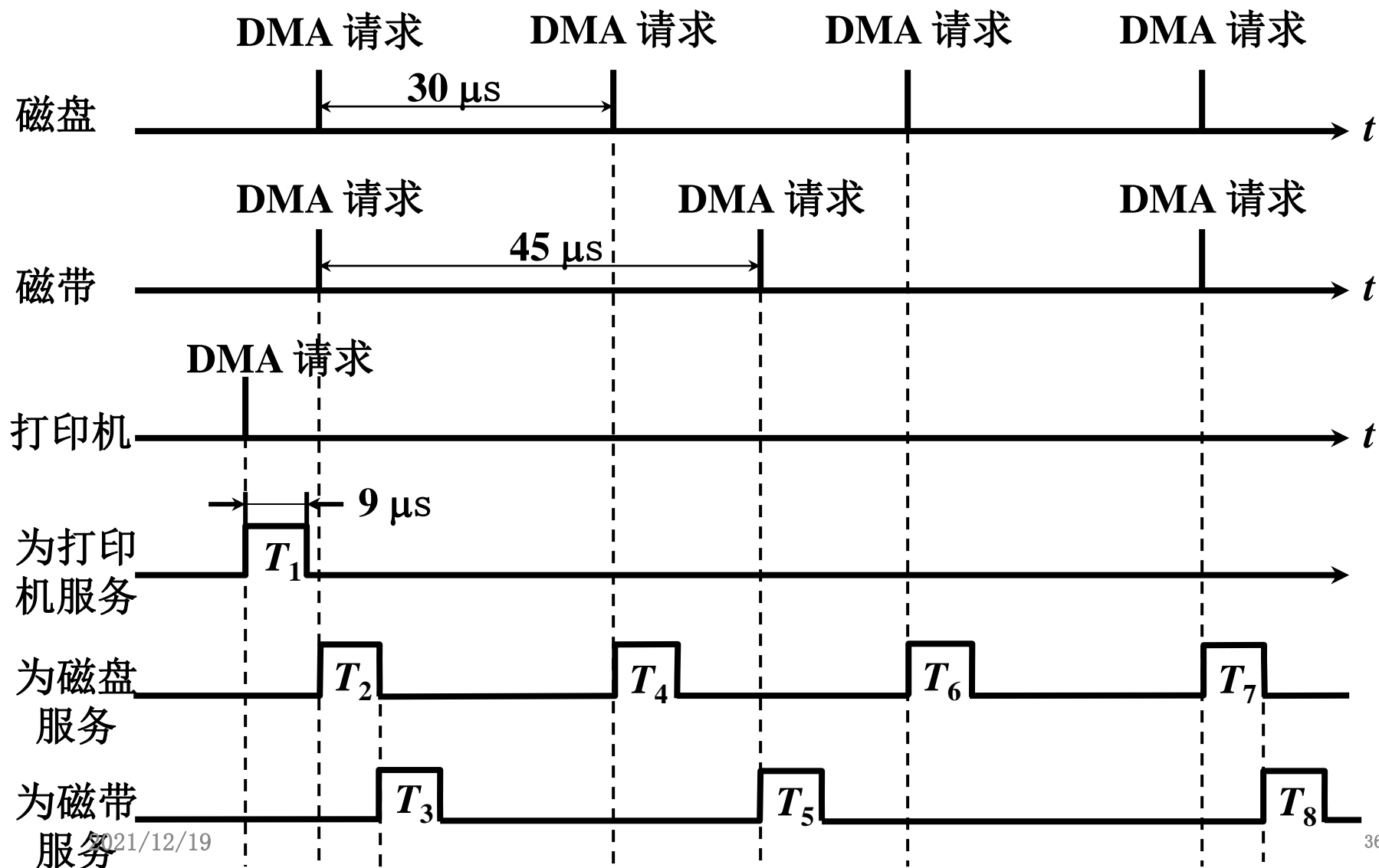
2. 多路型

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

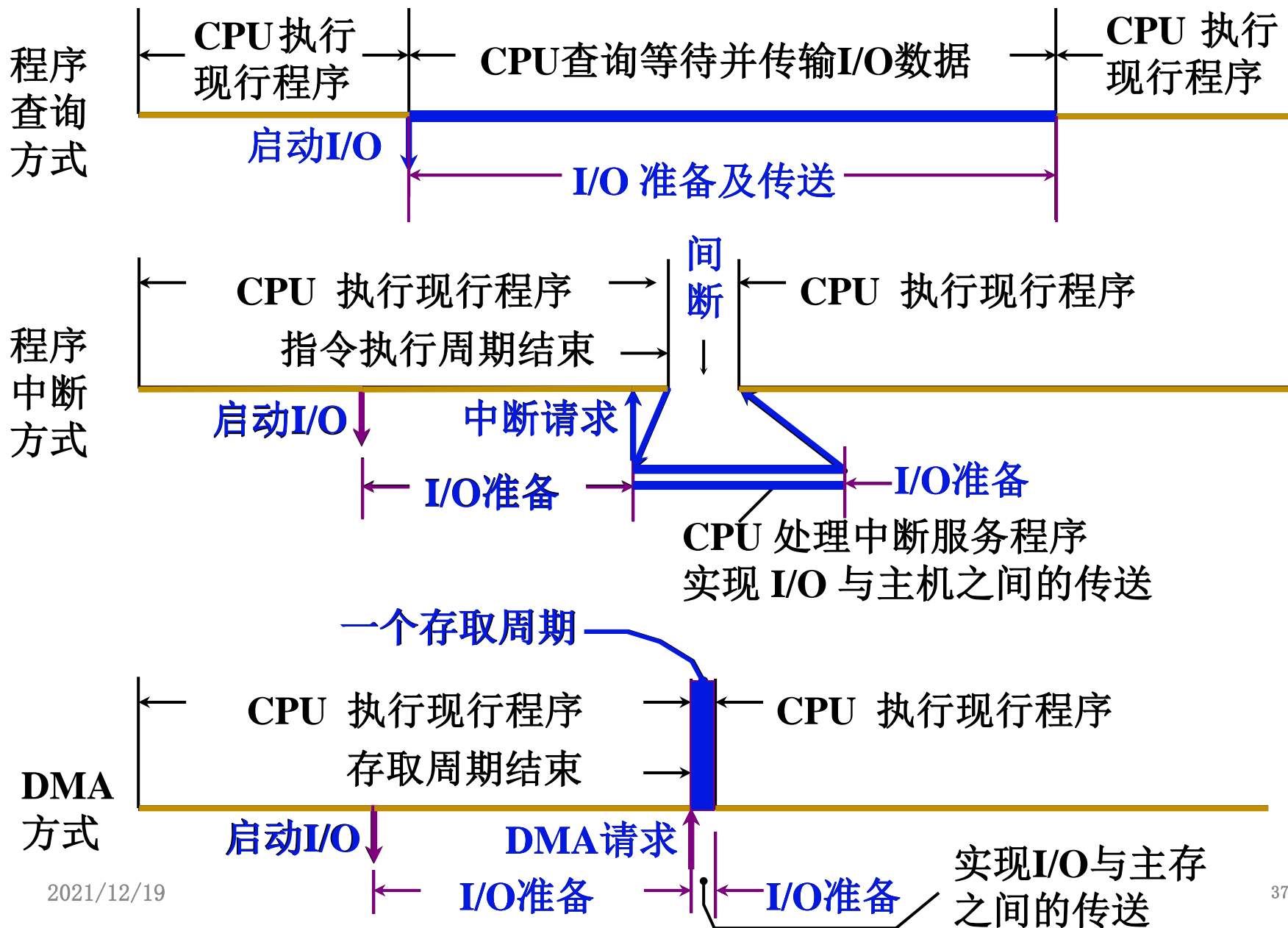


3. 多路型 DMA 接口的工作原理

9.6



三种方式的 CPU 工作效率比较



附： 通道

通道的功能

- 接受CPU发来的I/O指令，根据指令要求选择一台指定的外围设备与通道相连接。
- 执行CPU为通道组织的通道程序，从主存中取出通道指令，对通道指令进行译码，并根据需要向被选中的设备控制器发出各种操作命令。

通道的功能

- 给出外围设备的有关地址，即进行读 / 写操作的数据所在的位置。如，磁盘存储器的柱面号、磁头号、扇区号等。
- 给出主存缓冲区的首地址，这个缓冲区用来暂时存放从外围设备上输入的数据，或者暂时存放将要输出到外围设备中去的数据。

通道的功能

- 控制外围设备与主存缓冲区之间数据交换的个数，对交换的数据个数进行计数，并判断数据传送工作是否结束。
- 指定传送工作结束时要进行的操作。例如，将外围设备的中断请求及通道的中断请求送往CPU等。

通道的功能

- 检查外围设备的工作状态，是正常或故障。根据需要将设备的状态信息送往主存指定单元保存。
- 在数据传输过程中完成必要的格式变换，例如，把字拆卸为字节，或者把字节装配成字等。

通道的种类

- 通道分为三种类型：
 - **字节多路通道**：简单的共享通道，为多台低速或中速的外围设备服务。采用分时方式工作。
 - **选择通道**：为高速外围设备（如磁盘存储器等）服务。在传送数据期间，只能为一台高速外围设备服务，在不同的时间内可以选择不同的设备。
 - **数组多路通道**：为高速设备服务。各台高速设备重迭操作。

谢谢！