

# 计算机组成原理

## 第十四讲

刘松波

哈工大计算学部

模式识别与智能系统研究中心

# 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

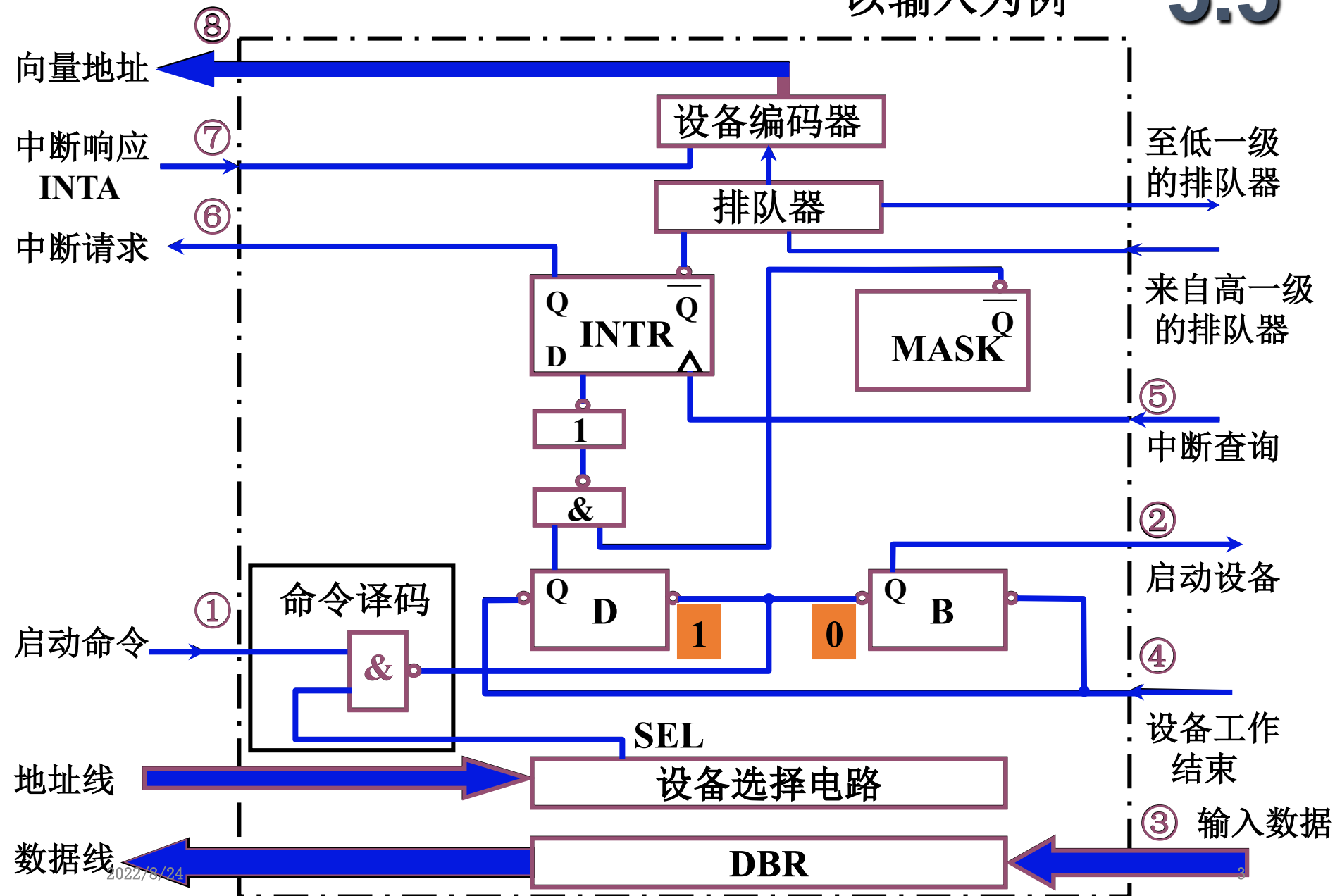
## 5.5 程序中断方式

## 5.6 DMA方式

## 2. I/O 中断处理过程

以输入为例

5.5



## 5.6 DMA方式

- 一、DMA方式的特点

- 1、DMA和程序中中断方式的数据通路
- 2、DMA与主存交换数据的三种方式
  - 停止CPU访问主存、周期挪用、DMA与CPU交替访问

- 二、DMA接口的功能和组成

- 1、DMA接口的功能
- 2、DMA接口的组成

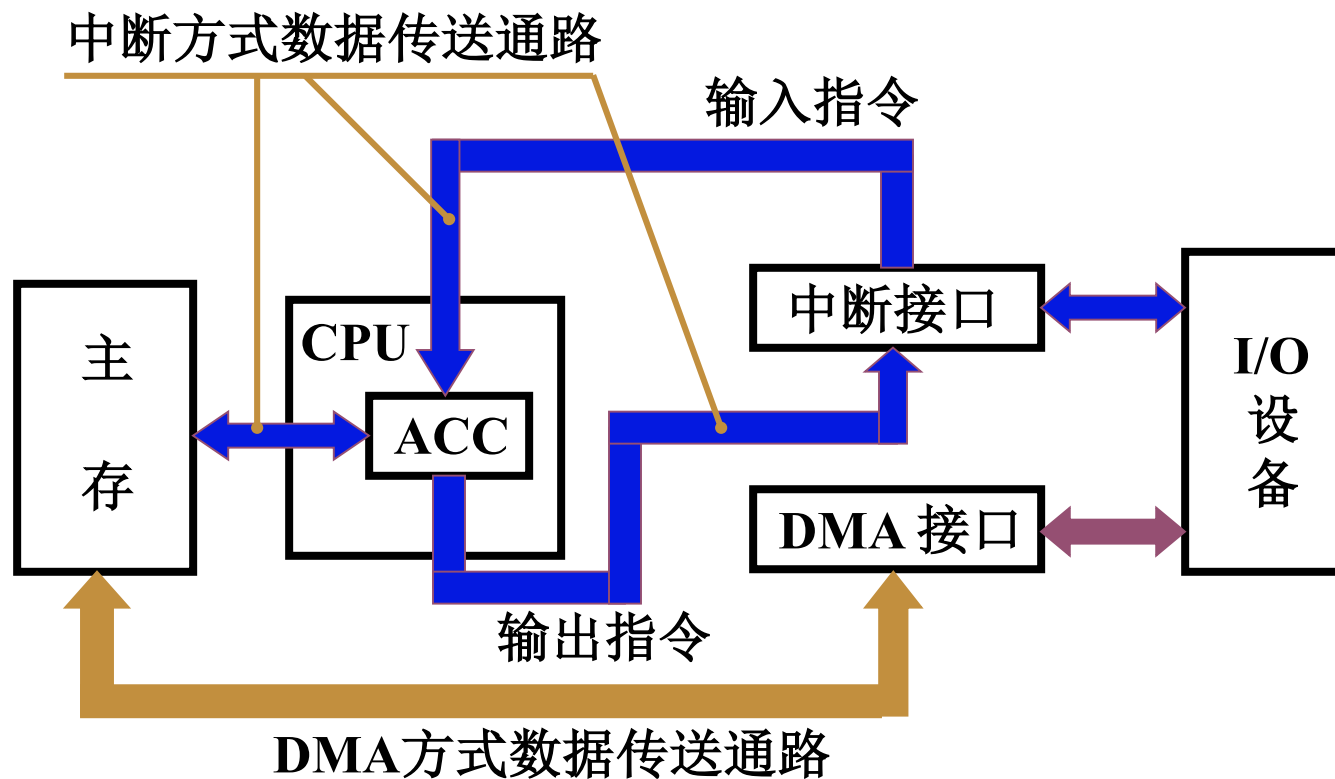
- 三、DMA的工作过程

- 1、DMA的传送过程
  - 预处理、DMA传送过程示意、数据的输入和输出过程
- 2、DMA接口与系统的连接方式
- 3、DMA方式与程序中中断方式的比较

## 5.6 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



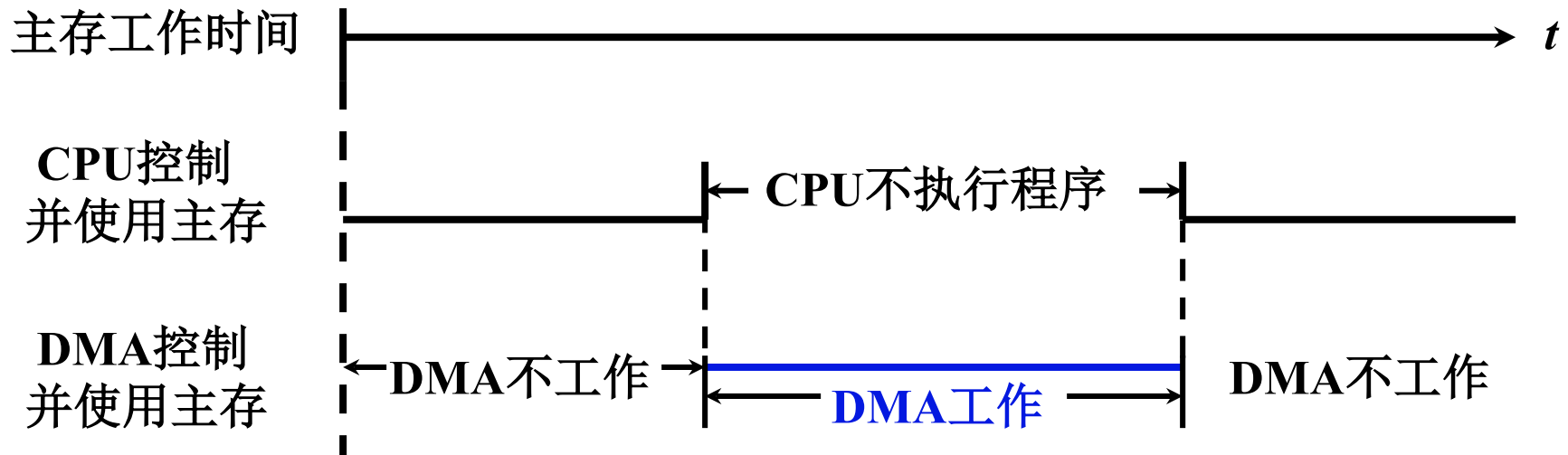
## 2. DMA 与主存交换数据的三种方式 5.6

### (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



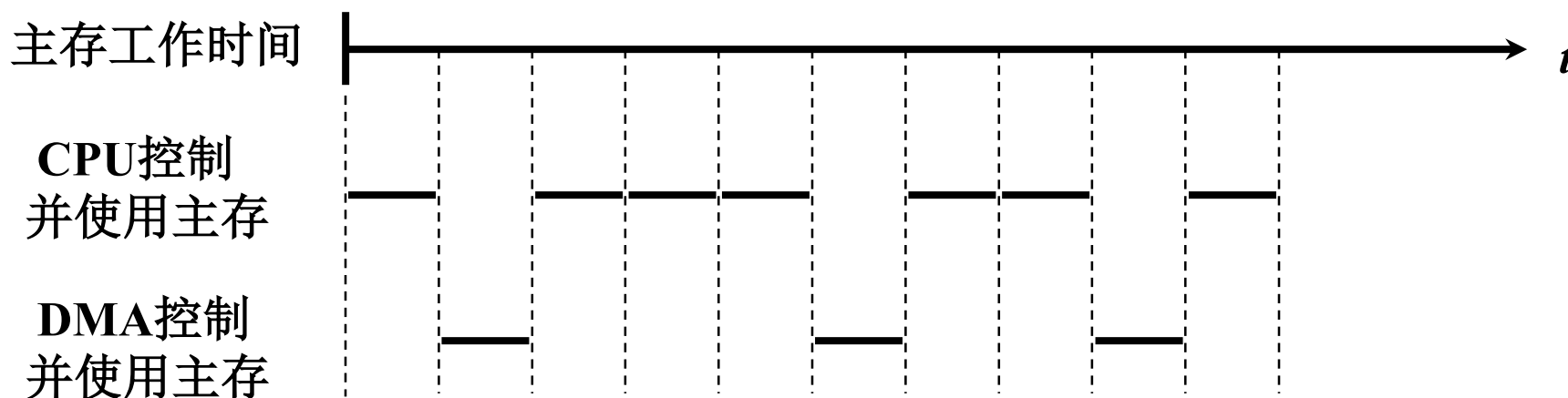
## (2) 周期挪用（或周期窃取）

5.6

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

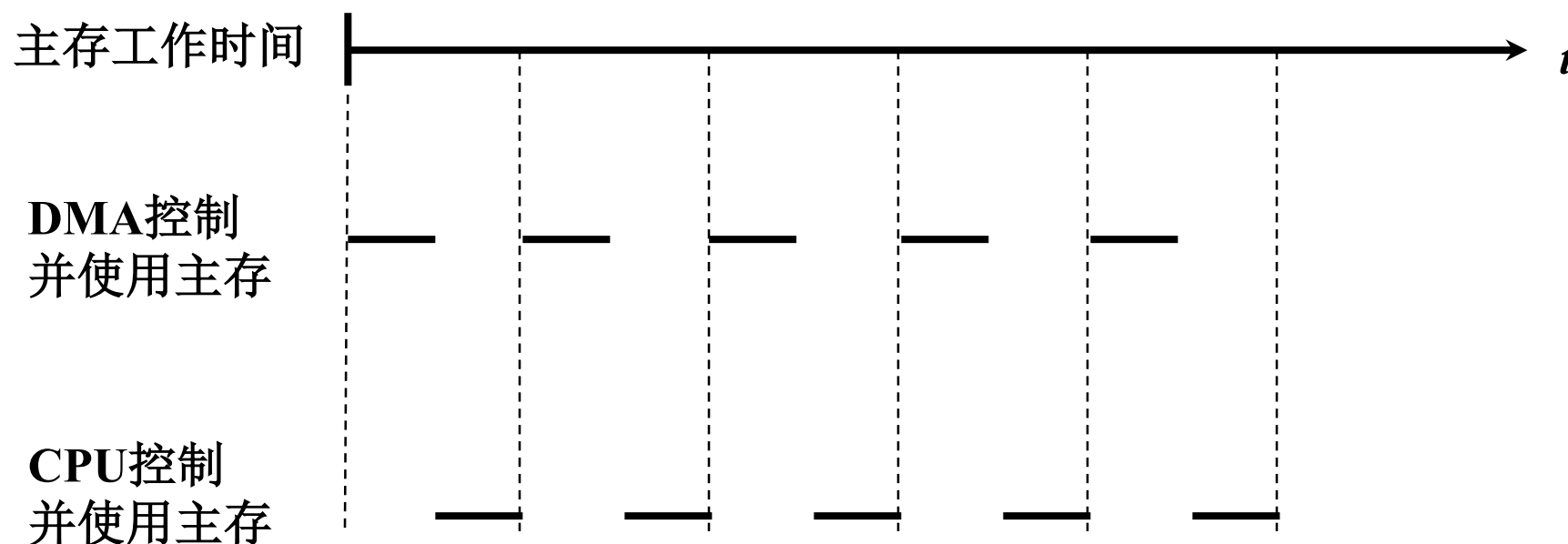
此时 CPU 将总线控制权让给 DMA



### (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

## 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 长度
- (5) DMA 传送结束时， 给出操作完成信号

# 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

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

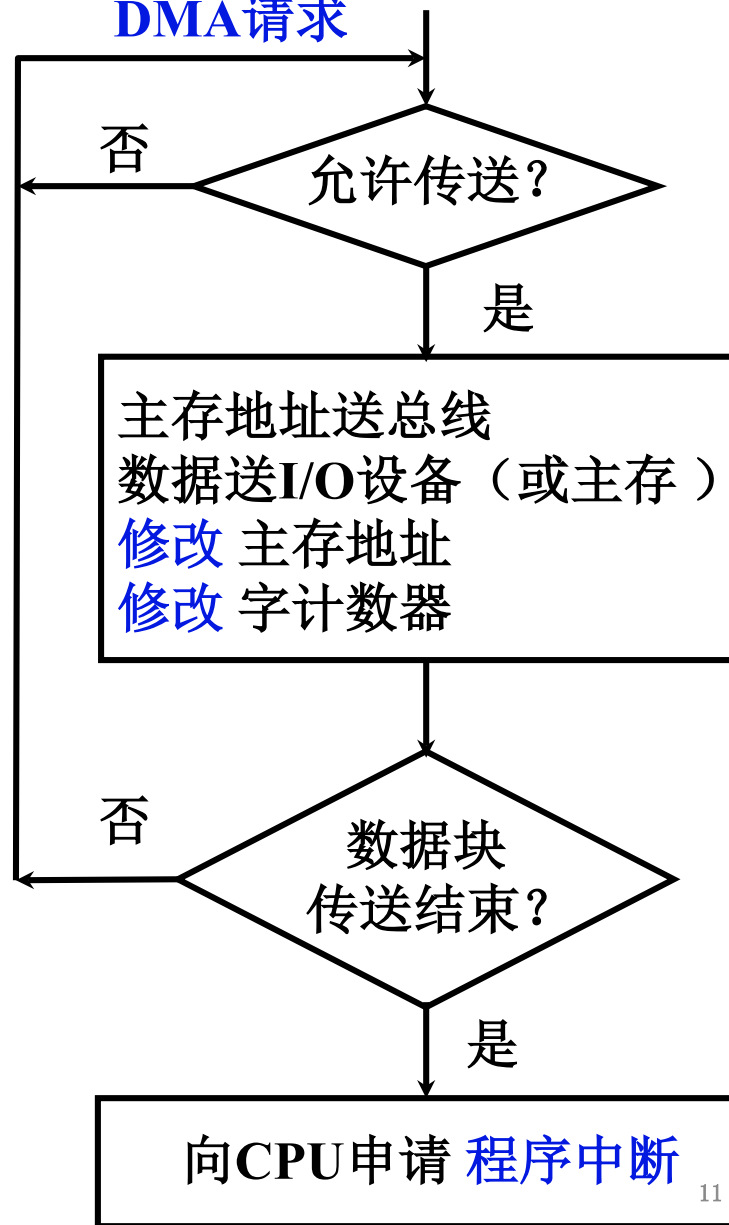
后处理:

中断服务程序  
做 DMA 结束处理

继续执行主程序

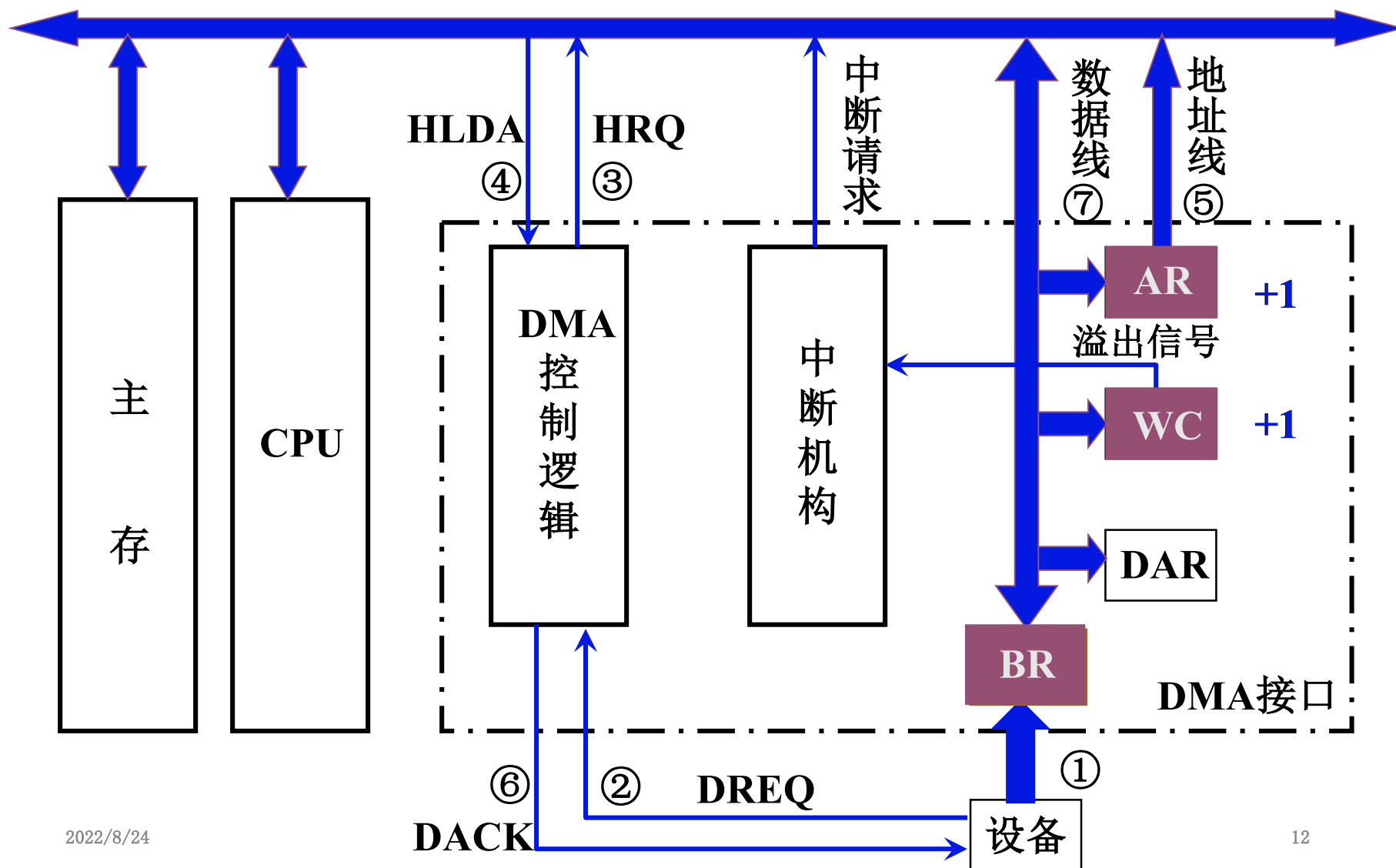
数据传送

DMA请求



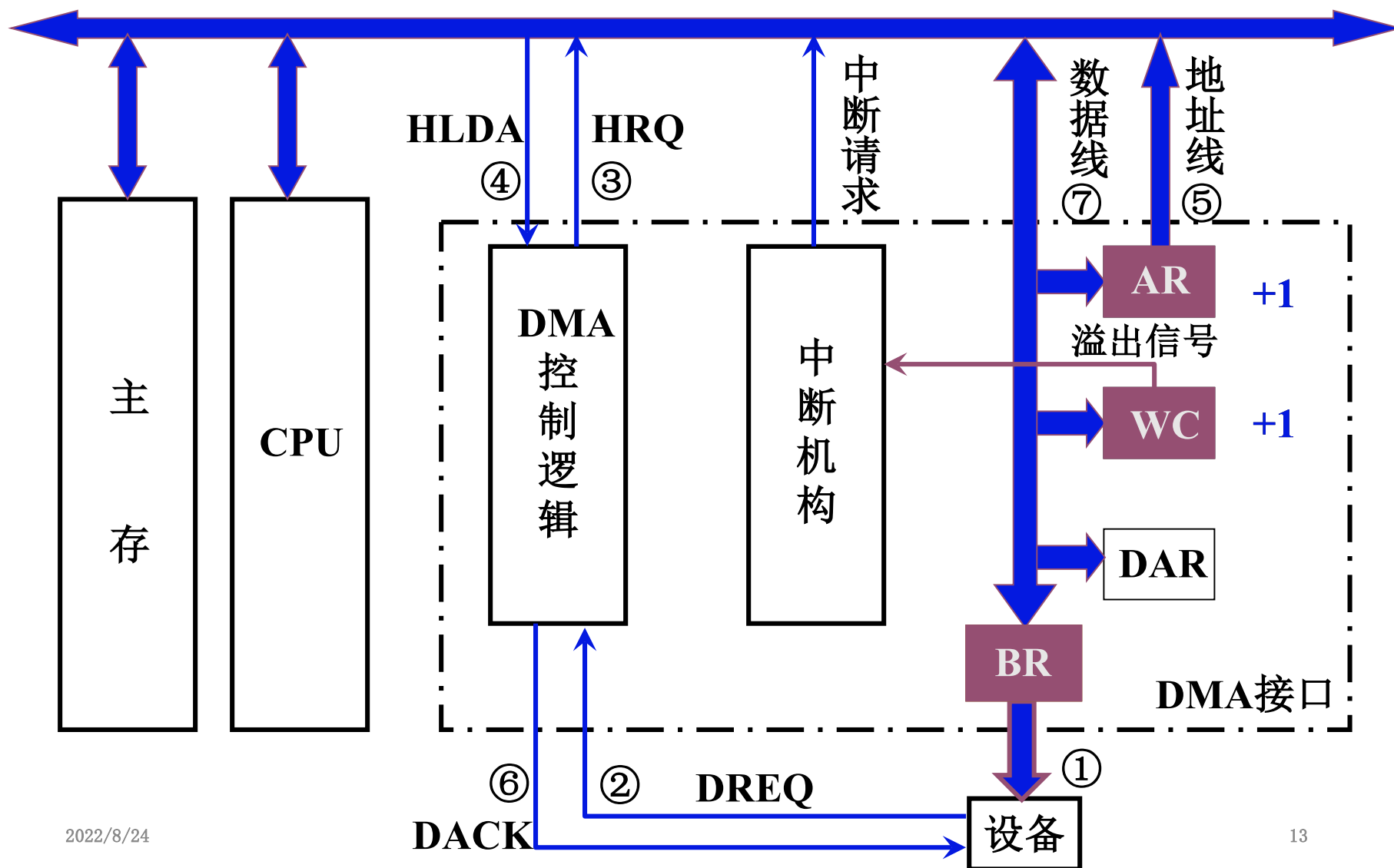
### (3) 数据传送过程（输入）

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

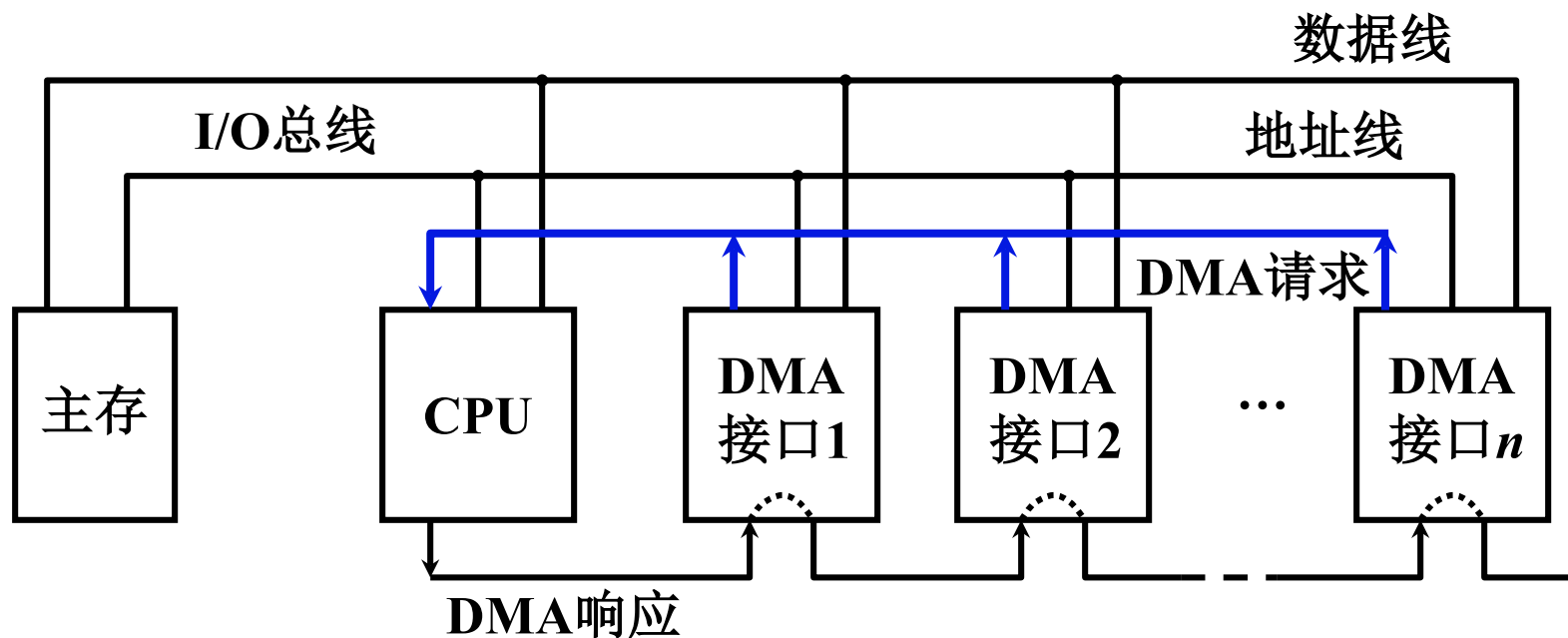
是否继续用 **DMA**

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

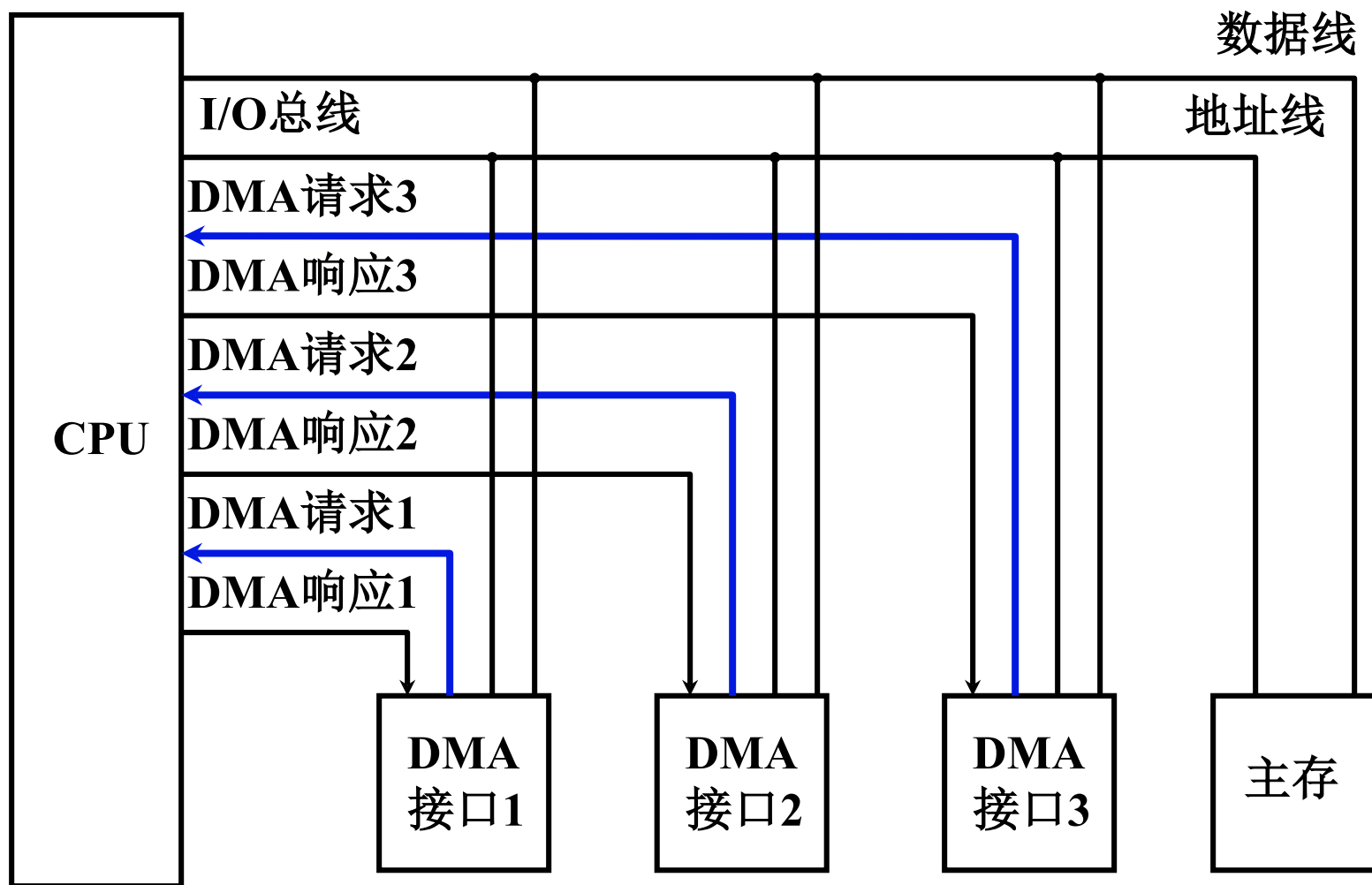
## 2. DMA 接口与系统的连接方式

### (1) 具有公共请求线的 DMA 请求



## (2) 独立的 DMA 请求

5.6





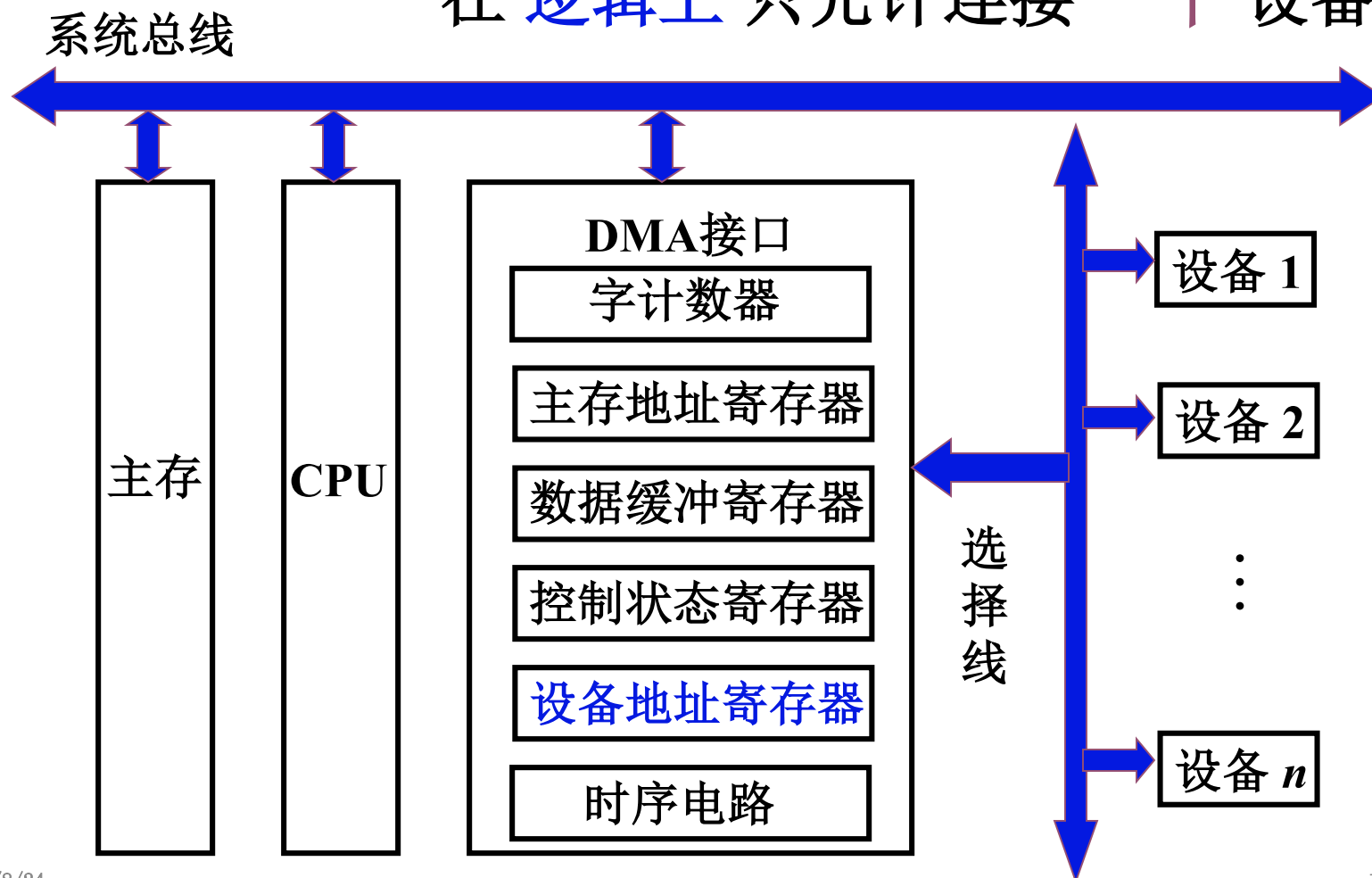
### 3. DMA 方式与程序中断方式的比较 5.6

	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

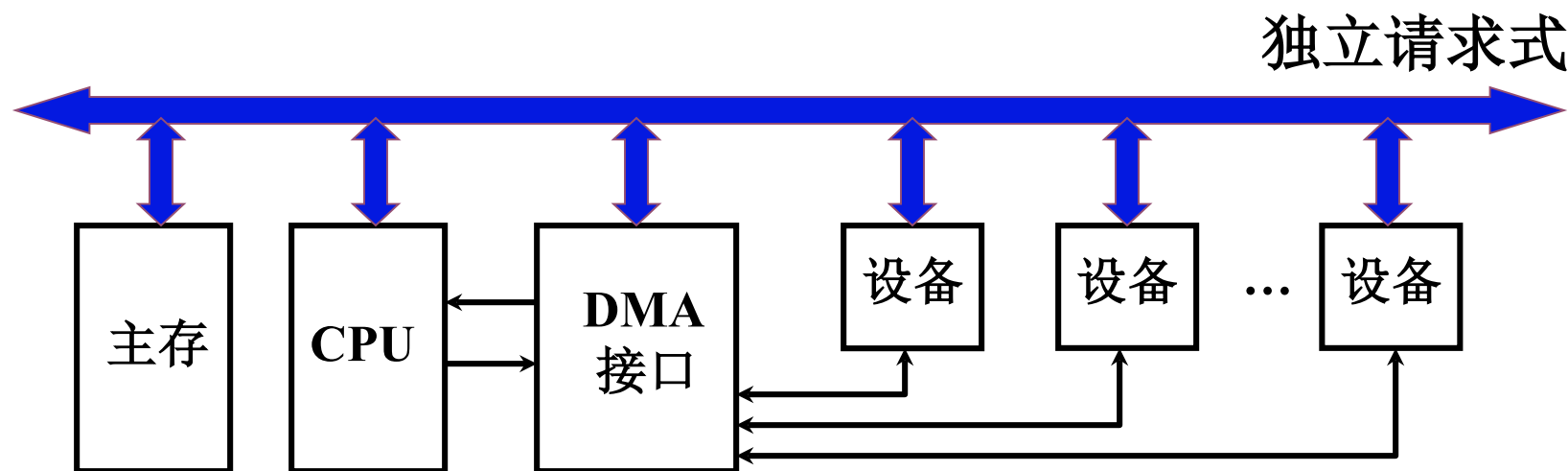
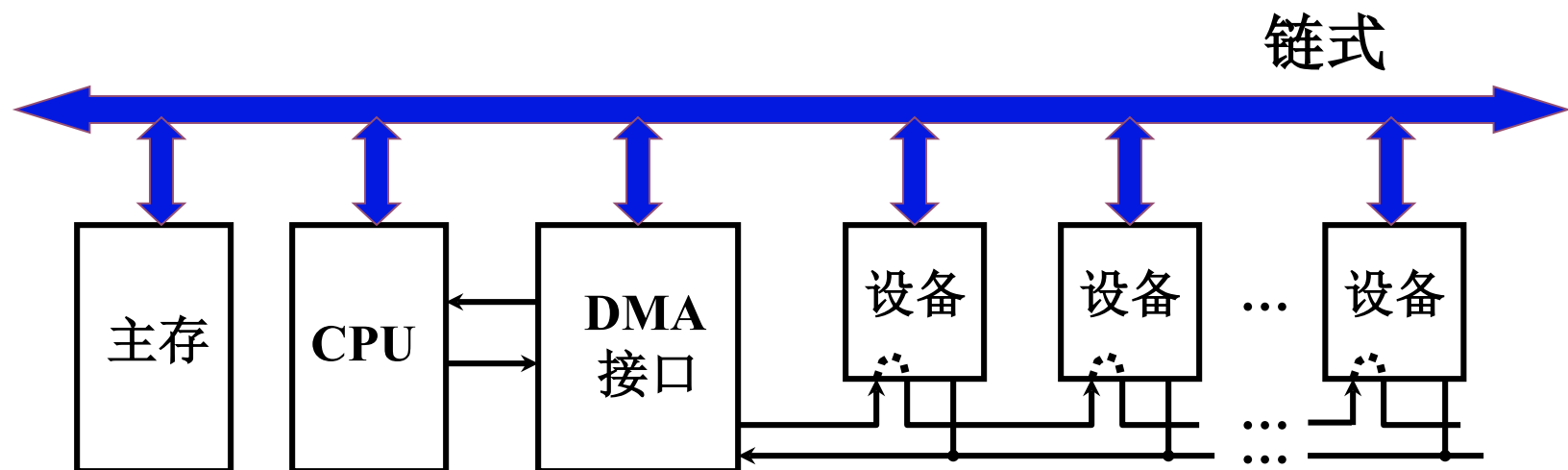
### 1. 选择型

在物理上连接多个设备  
在逻辑上只允许连接一个设备



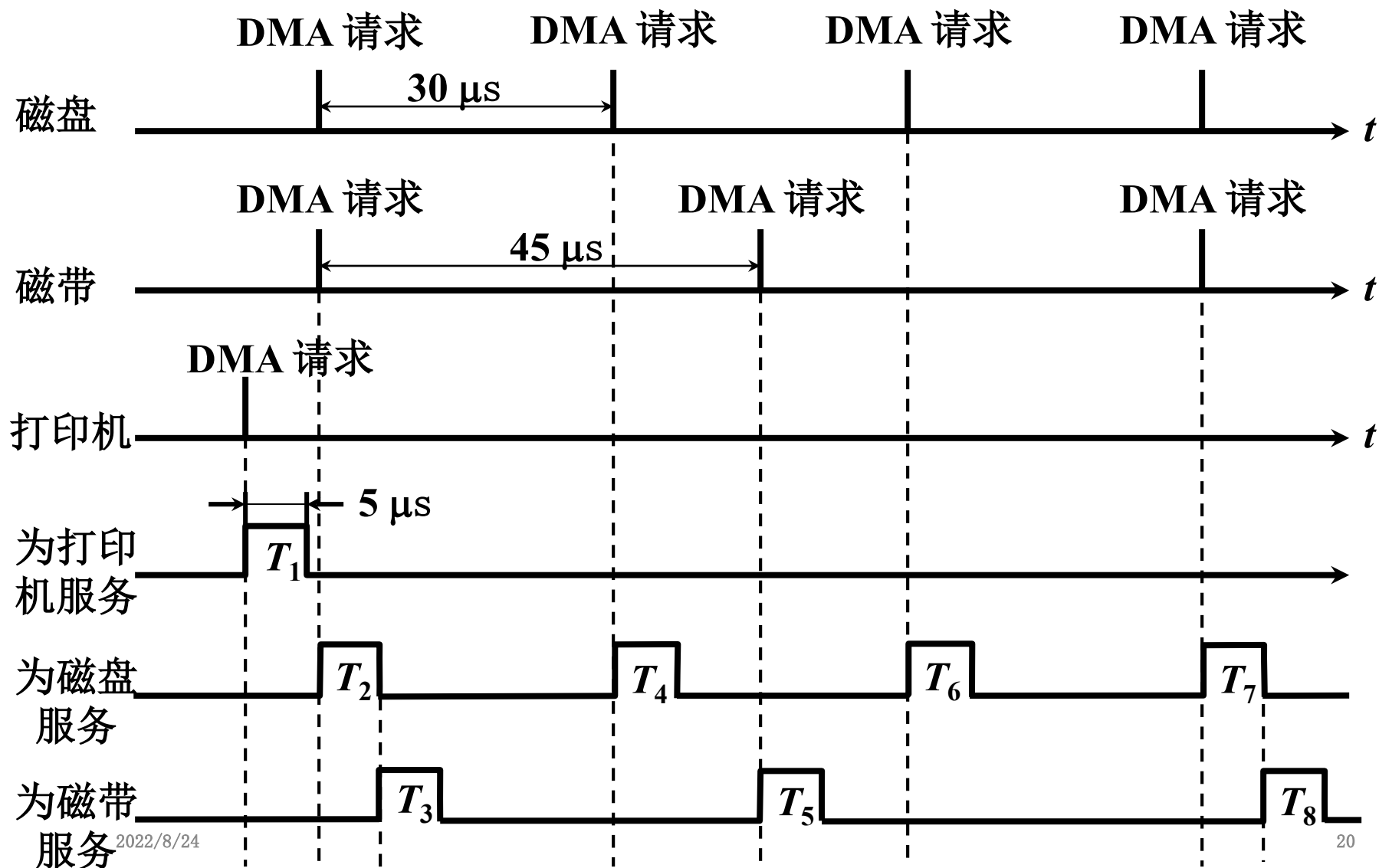
## 2. 多路型

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



# 第 6 章 计算机的运算方法

## 6.1 无符号数和有符号数

## 6.2 数的定点表示和浮点表示

## 6.3 定点运算

## 6.4 浮点四则运算

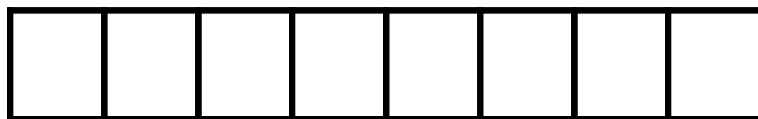
## 6.5 算术逻辑单元

## 6.1 无符号数和有符号数

### 一、无符号数

寄存器的位数

反映无符号数的表示范围



**8 位**

**0 ~ 255**



**16 位**

**0 ~ 65535**

## 二、有符号数

## 6.1

### 1. 机器数与真值

真值

带符号的数

**+ 0.1011**

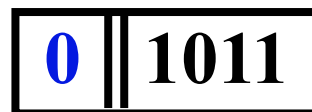
**- 0.1011**

**+ 1100**

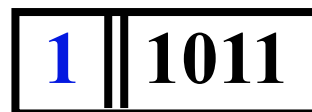
**- 1100**

机器数

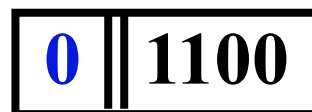
符号数字化的数



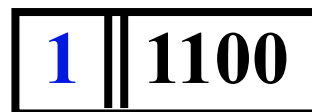
↑ 小数点的位置



↑ 小数点的位置



↑ 小数点的位置



↑ 小数点的位置

## 2. 原码表示法

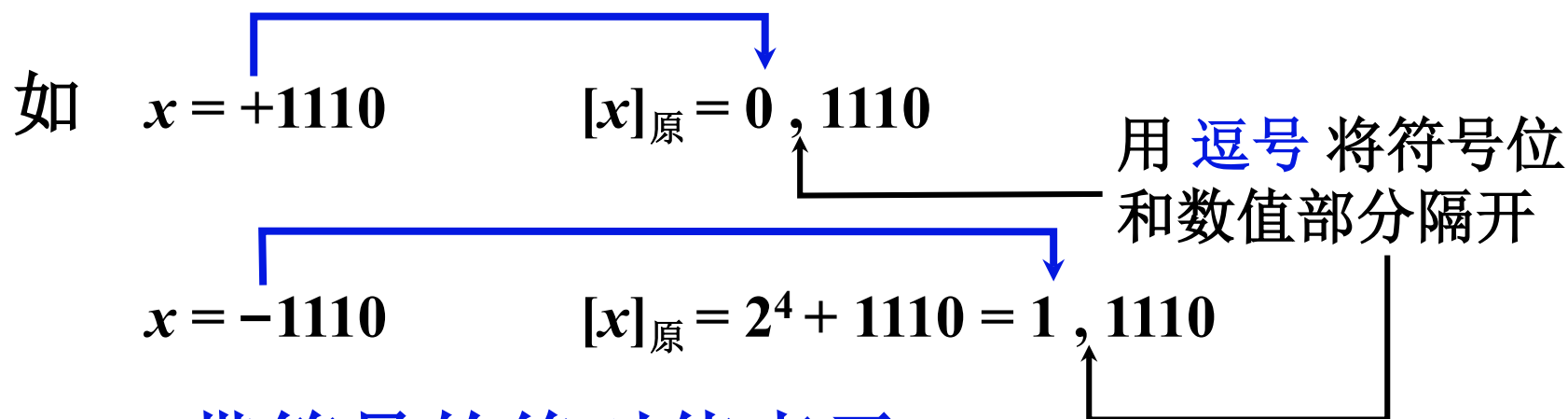
# 6.1

### (1) 定义

整数

$$[x]_{\text{原}} = \begin{cases} 0, & x & 2^n > x \geq 0 \\ 2^n - x & 0 \geq x > -2^n \end{cases}$$

$x$  为真值       $n$  为整数的位数



带符号的绝对值表示



# 小数

## 6.1

$$[x]_{\text{原}} = \begin{cases} x & 1 > x \geq 0 \\ 1 - x & 0 \geq x > -1 \end{cases}$$

$x$  为真值

如  $x = +0.1101$        $[x]_{\text{原}} = 0.1101$       用 **小数点** 将符号位和数值部分隔开

$x = -0.1101$        $[x]_{\text{原}} = 1 - (-0.1101) = 1.1101$

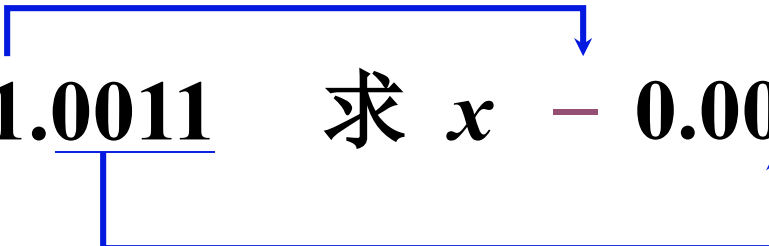
$x = +0.1000000$        $[x]_{\text{原}} = 0.1000000$       用 **小数点** 将符号位和数值部分隔开

$x = -0.1000000$        $[x]_{\text{原}} = 1 - (-0.1000000) = 1.1000000$

## (2) 举例

6.1

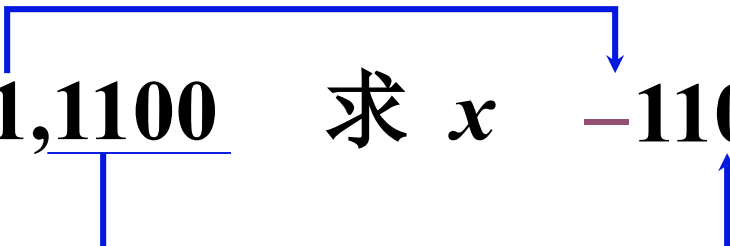
例 6.1 已知  $[x]_{\text{原}} = 1.0011$  求  $x - 0.0011$



解：由定义得

$$x = 1 - [x]_{\text{原}} = 1 - 1.0011 = -0.0011$$

例 6.2 已知  $[x]_{\text{原}} = 1,1100$  求  $x - 1100$



解：由定义得

$$x = 2^4 - [x]_{\text{原}} = 10000 - 1,1100 = -1100$$

例 6.3 已知  $[x]_{\text{原}} = 0.1101$  求  $x$

解：根据定义  $\because [x]_{\text{原}} = 0.1101$

$$\therefore x = +0.1101$$

例 6.4 求  $x = 0$  的原码

解：设  $x = +0.0000$   $[+0.0000]_{\text{原}} = 0.0000$

$x = -0.0000$   $[-0.0000]_{\text{原}} = 1.0000$

同理，对于整数  $[+0]_{\text{原}} = 0,0000$

$[-0]_{\text{原}} = 1,0000$

$$\therefore [+0]_{\text{原}} \neq [-0]_{\text{原}}$$

# 原码的特点：简单、直观

## 6.1

但是用原码作加法时，会出现如下问题：

要求	数1	数2	实际操作	结果符号
加法	正	正	加	正
加法	正	负	减	可正可负
加法	负	正	减	可正可负
加法	负	负	加	负

能否 只作加法？

找到一个与负数等价的正数 来代替这个负数

就可使 减  $\longrightarrow$  加