



计算机组成原理

第五章 输入/输出系统





主要内容

- 1 概述
- 2 模型机系统总线组成
- 3 直接程序传送方式与接口
- 4 中断方式及接口
- 5 DMA方式及接口

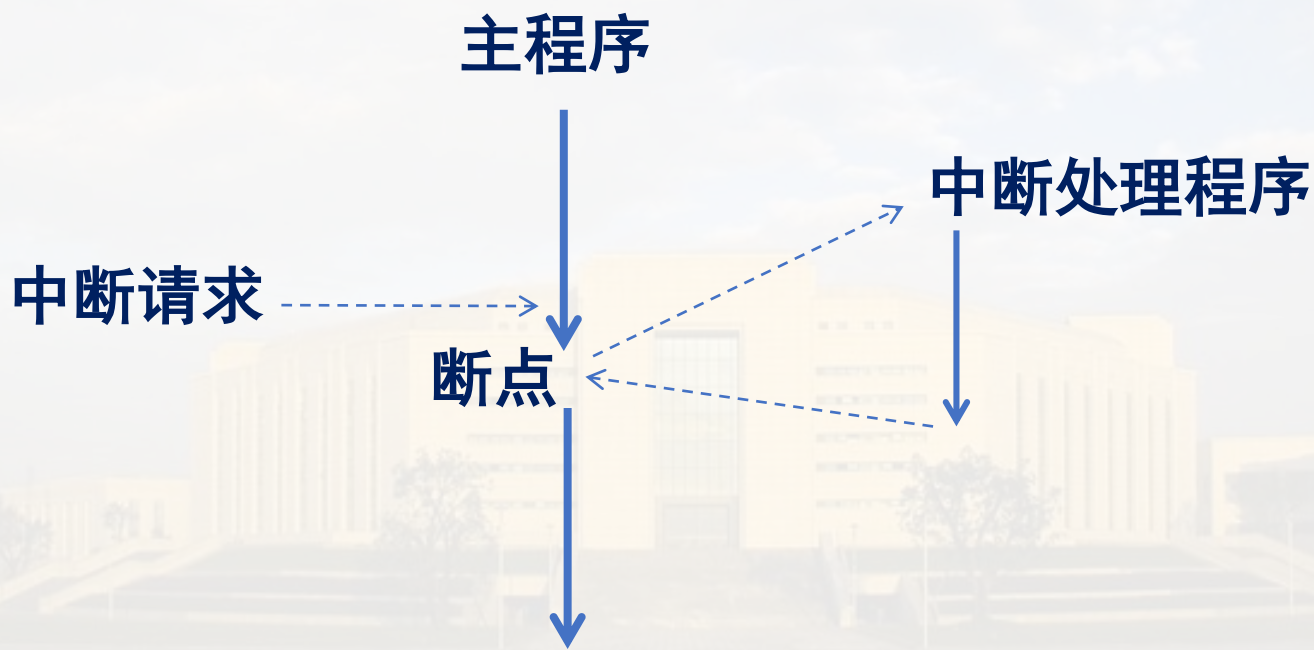
5.4 中断方式及接口

- 01. 中断基本概念
- 02. 中断全过程（外中断）
- 03. 优先权逻辑与屏蔽技术
- 04. 服务程序入口地址的获取方式
- 05. 中断处理
- 06. 中断接口

1、定义

CPU暂时中止现行政程序的执行，转去执行为某个**随机**事态服务的中断处理程序。处理完毕后自动恢复原程序的执行。

中断的这一定义包含了**程序切换**和**随机性**两个重要**特征**。



2、实质与特点

1) 实质

程序切换 { 方法: 保存断点, 保护现场;(中断处理前)
恢复现场, 返回断点.(中断处理后)
时间: 一条指令结束时切换。
保证程序的完整性。

2) 特点

随机性 { 随机发生的事态 (按键、故障)
有意调用,随机请求与处理的事态 (调用打印机)
随机插入的事态 (软中断指令插入程序任何位置)

注意中断与转子的区别。

中断与转子程序的区别:

- 子程序的执行由程序员事先安排,而中断服务程序的执行则是由随机中断事件触发。
- 子程序的执行受主程序或上层程序控制,而中断服务程序可能与被中断的现行程序无关。
- 一般不存在同时调用多个子程序的情况,但可能发生多个外设同时向CPU发出中断服务请求的情况。

一、中断基本概念

3、中断分类

1) 硬件中断与软中断 — 由软中断指令引发中断

由硬件请求信号引发中断

2) 内中断与外中断 — 中断源来自主机外部

中断源来自主机内部

3) 可屏蔽中断与非屏蔽中断 — 该类请求与屏蔽字无关；
请求的响应与开/关中断无关。

可通过屏蔽字屏蔽该类请求；
关中断时不响应该类请求。

4) 向量中断与非向量中断 — 由软件查询确定服务程序入口地址

由硬件方式确定服务程序入口地址

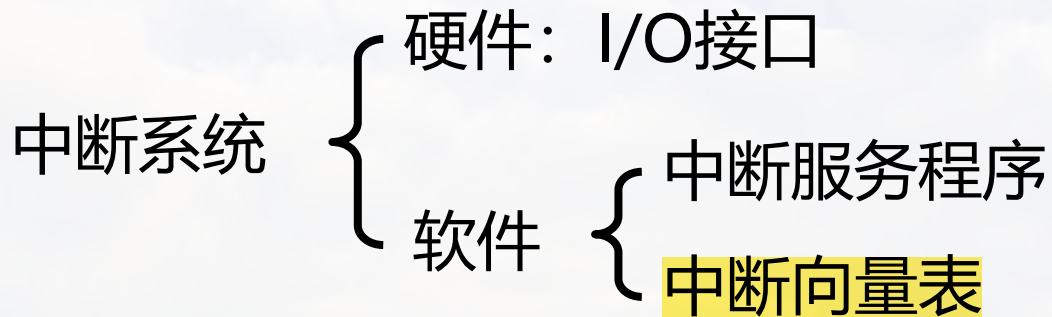
4、中断典型应用

- 1) 管理中、低速I/O操作
- 2) 处理故障
- 3) 实时处理

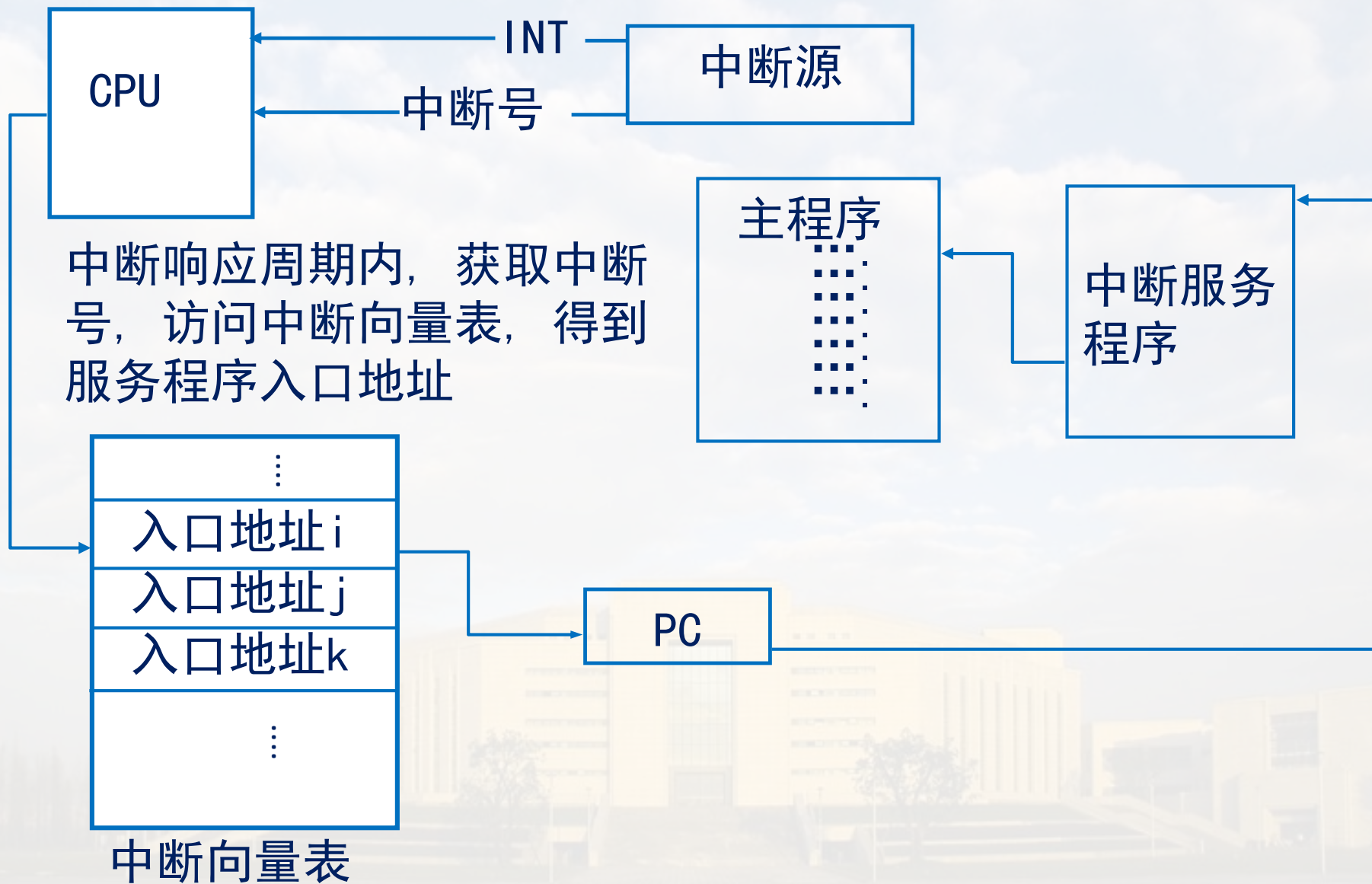
某事件出现的实际时间内及时处理，不是批量处理。
利用时钟中断定时采集参数，检测，调节。

- 4) 人机对话
- 5) 多机通信

5、中断系统的硬、软件组织



一、中断基本概念



【例】一种典型的中断组织方法（以模型机为例）

1) 首先列出系统中的各种中断请求源

外部硬件中断源：8种，IREQ0—IREQ7

IRQ0—系统时钟，如日历钟；IRQ1—实时时钟，供实时处理用；
IRQ2—通信中断，组成多机系统或联网时用；IRQ3—键盘；
IRQ4—CRT显示器；IRQ5—硬盘；IRQ6—软盘；IRQ7—打印机。

内部硬件中断源：掉电、溢出、校验错中断等

软中断：模型机软中断为INT11—INTn

2) 中断服务程序

在主存中的空间不必连续。

3) 中断向量表

存放在模型机中主存的2号单元开始；向量地址=中断号+2

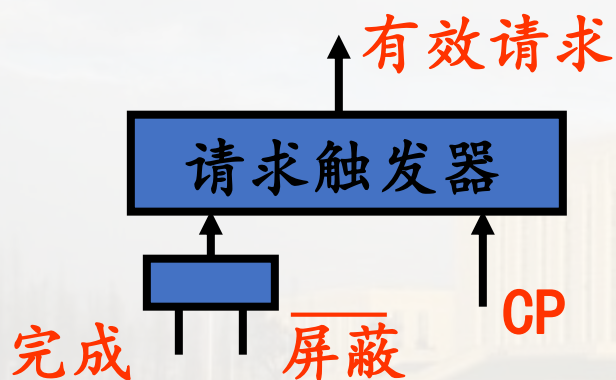
二、中断全过程（外中断）

1、如何产生中断请求？

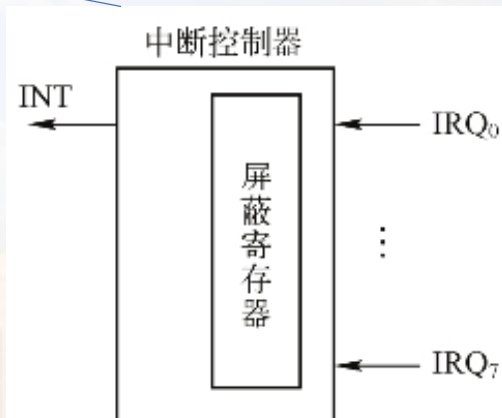
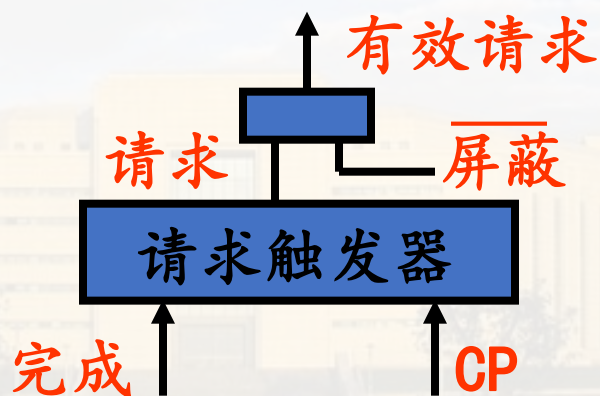
- 外设有产生中断的需要：“就绪或完成” 可用触发器状态 $T_D=1$ 表示
- 该中断源允许请求：“屏蔽” 屏蔽触发器 (T_M) 为 0，表示允许中断

可采取分散屏蔽和集中屏蔽两种方式

先“屏蔽”，后请求



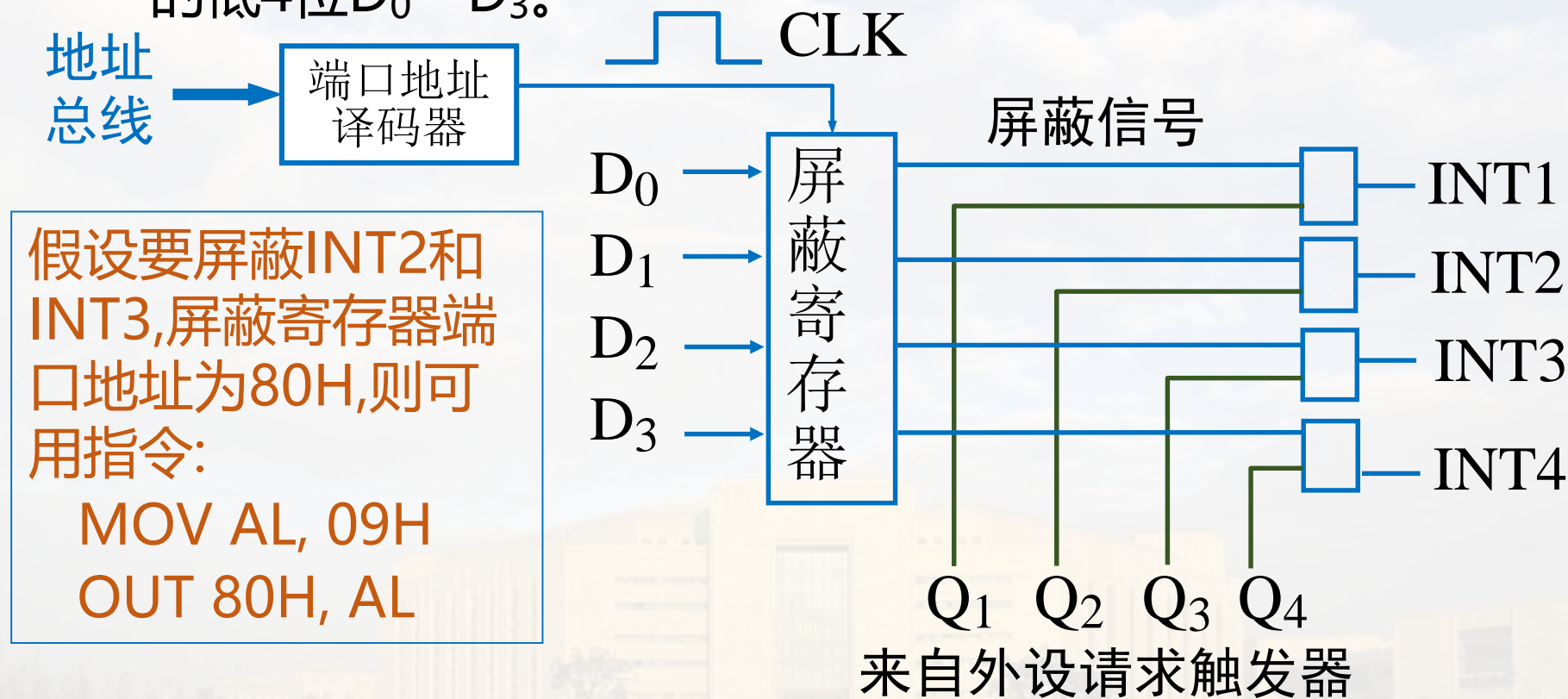
先请求，后“屏蔽”



二、中断全过程（外中断）

例：有多个中断源时设置屏蔽寄存器

假设有4个中断源，设置一个8位屏蔽寄存器，仅使用寄存器的低4位 $D_0 \sim D_3$ 。

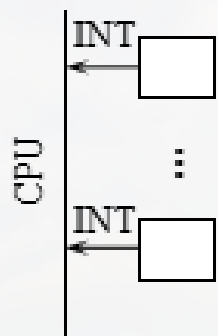


设置屏蔽位的优点:

- CPU可根据程序执行情况屏蔽或不屏蔽某些中断源
- 可以很容易改变中断源的优先级

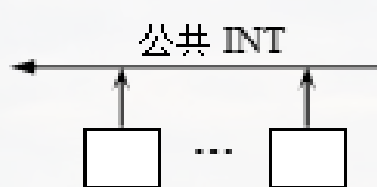
二、中断全过程（外中断）

2、如何传送中断请求？



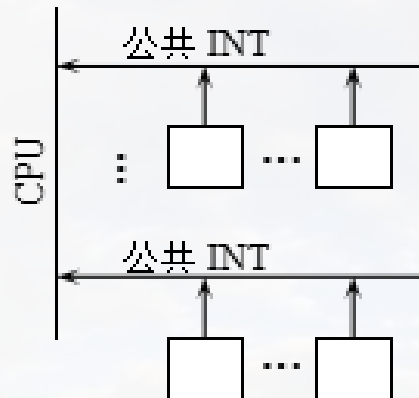
(a)

独立请求



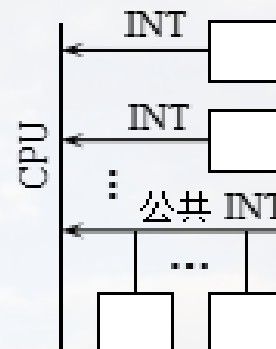
(b)

公共请求



(c)

二维结构



(d)

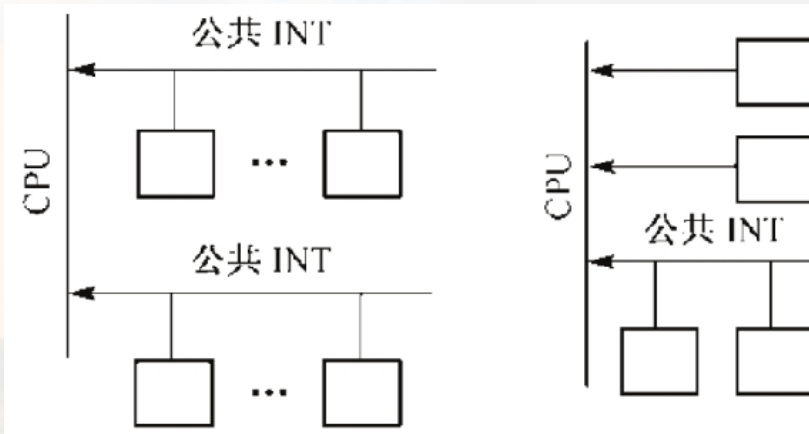
兼独立与
公共请求

c.采用二维结构

这是前面两种方式的折衷方案，CPU芯片具有多根中断请求输入线，它们能体现不同的优先级别。

每根中断请求线可以将具有相同优先级的中断请求源汇集在一起。这就综合了前两种模式的优点，既容易识别不同级别的优先级中断源，又可扩充中断源数目。

d.兼有公共请求与独立请求线结构



当这些中断源同时提出申请时，CPU该响应哪个请求？该中断请求是否能够中断CPU现执行程序执行？所有这些都由**中断判优处理逻辑电路**来处理。

1、CPU（现执行程序）与中断请求间的判优

在一般计算机中，CPU进行简单的判优处理，根据CPU标志寄存器中的“允许中断”控制位（IF）状态，确定是否响应该中断请求。

CPU设置允许中断标志 $\left\{ \begin{array}{l} \text{IF}=1, \text{ 开中断} \\ \text{IF}=0, \text{ 关中断} \end{array} \right.$ (模型机采用)

在性能更强的计算机中，除了设置IF中断控制位外，还在程序状态字PSW中设定现行程序优先级，以便进一步细分现行程序任务的重要程度。

b.CPU设置程序状态字的优先级字段

为现行程序赋予优先级	$\left\{ \begin{array}{l} < \text{外设请求优先级,} \\ \geq \text{外设请求优先级,} \end{array} \right.$	响应 不响应
------------	--	-----------

2、中断请求之间的判优

1) 优先顺序 故障、DMA、外中断。输入优先级高于输出优先级

2) 各外设请求的判优方法

a.软件判优

由程序查询顺序确定优先级。

可灵活修改优先级。

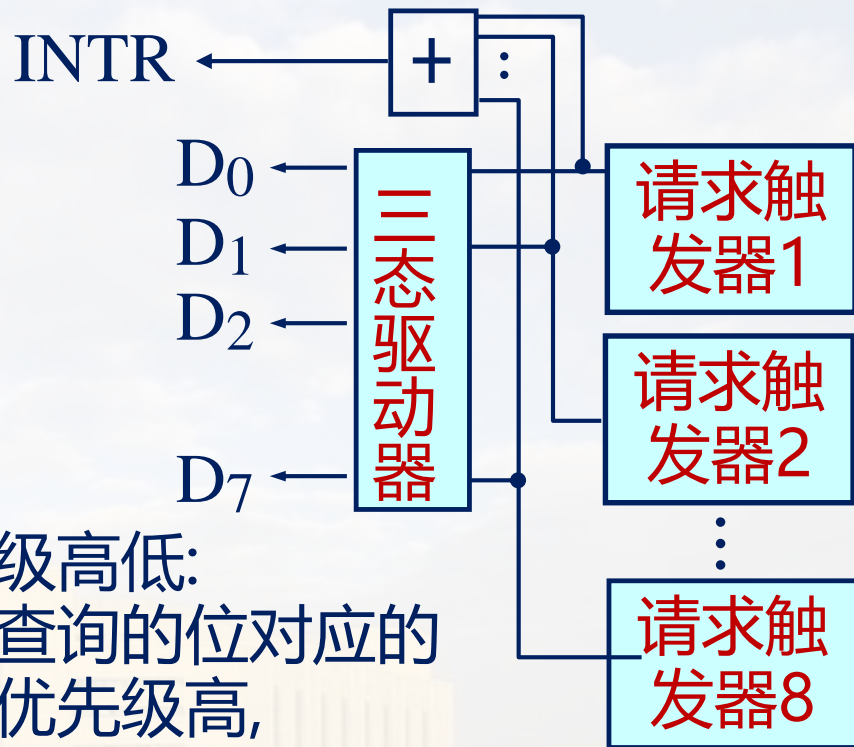
b.硬件判优

例.中断控制器判优

中断控制器(如8259)集中解决请求信号的接收、屏蔽、判优、编码等问题。

三、优先权逻辑与屏蔽技术

a) 软件判优 由程序查询顺序确定优先级。

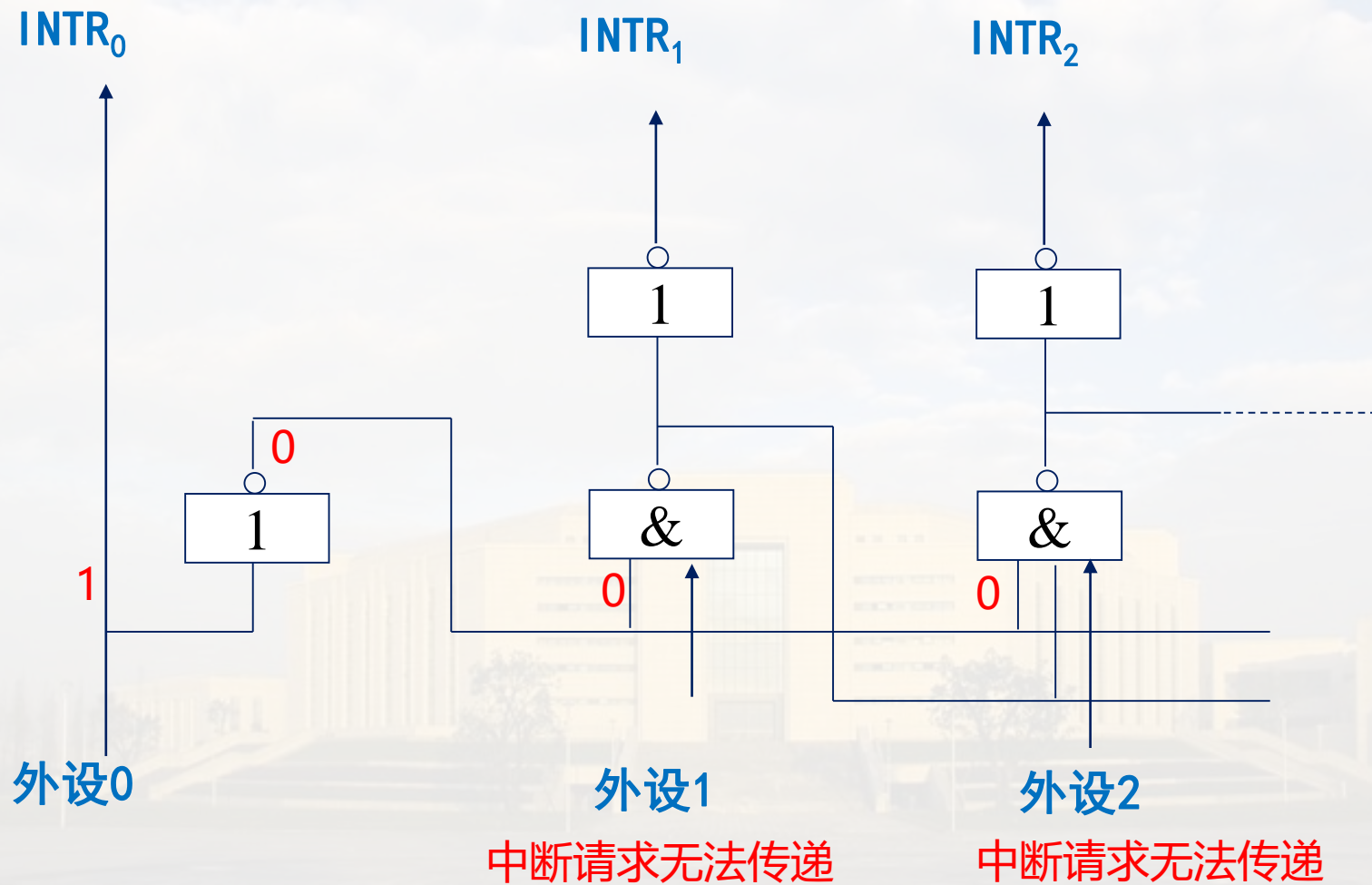


优先级高低:
先被查询的位对应的
请求优先级高,

通过修改查询软件易于改
变中断源的优先级

b) 硬件判优

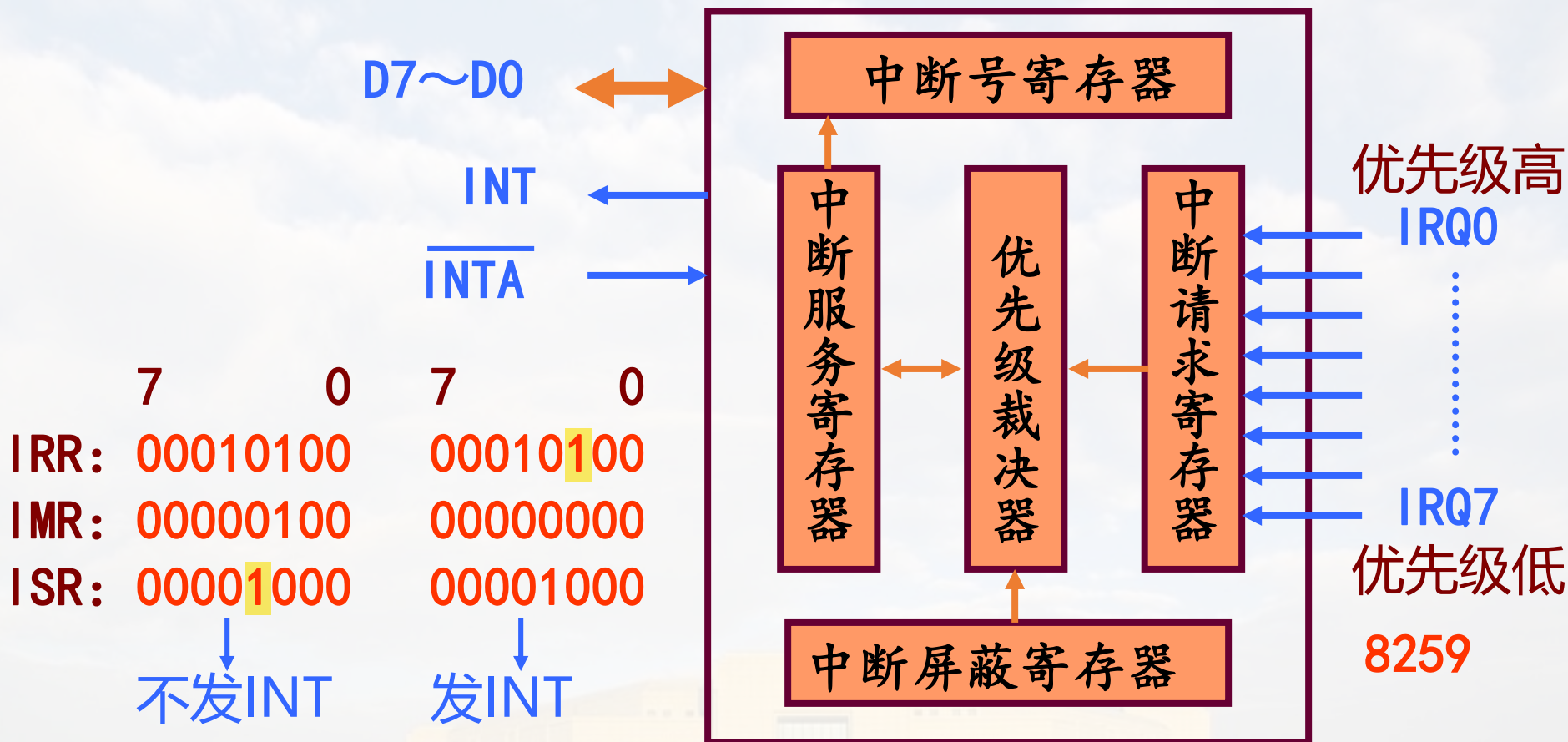
① 一种采用独立请求线的并行判优逻辑 (了解)



① 一种采用独立请求线的并行判优逻辑



三、优先权逻辑与屏蔽技术



中断请求 → 8259 (未屏蔽的请求判优,生成相应中断号)

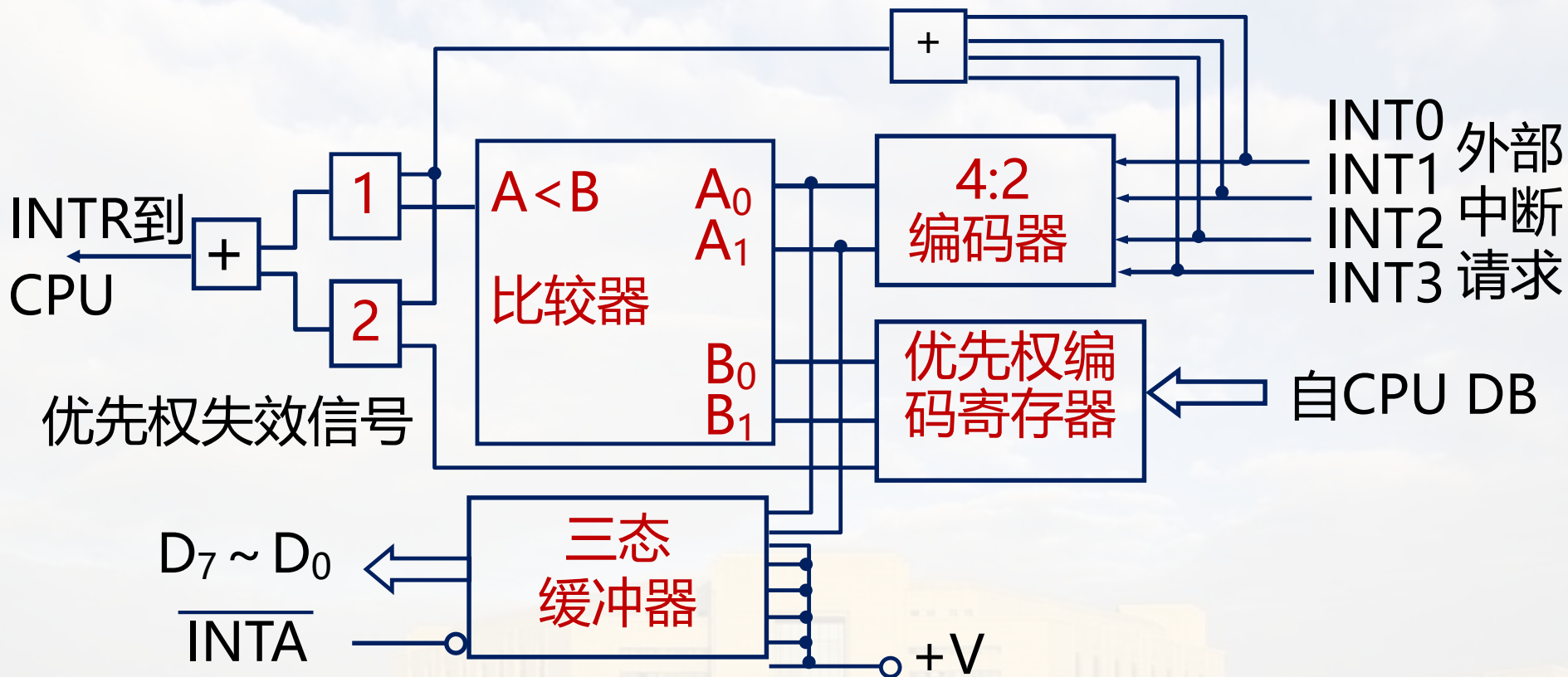
→ 公共请求INT → CPU

(CPU响应后, 取回中断号, 转入相应服务程序。)

中断源的序号

三、优先权逻辑与屏蔽技术

一种典型并行判优电路 (四个中断源): (了解)

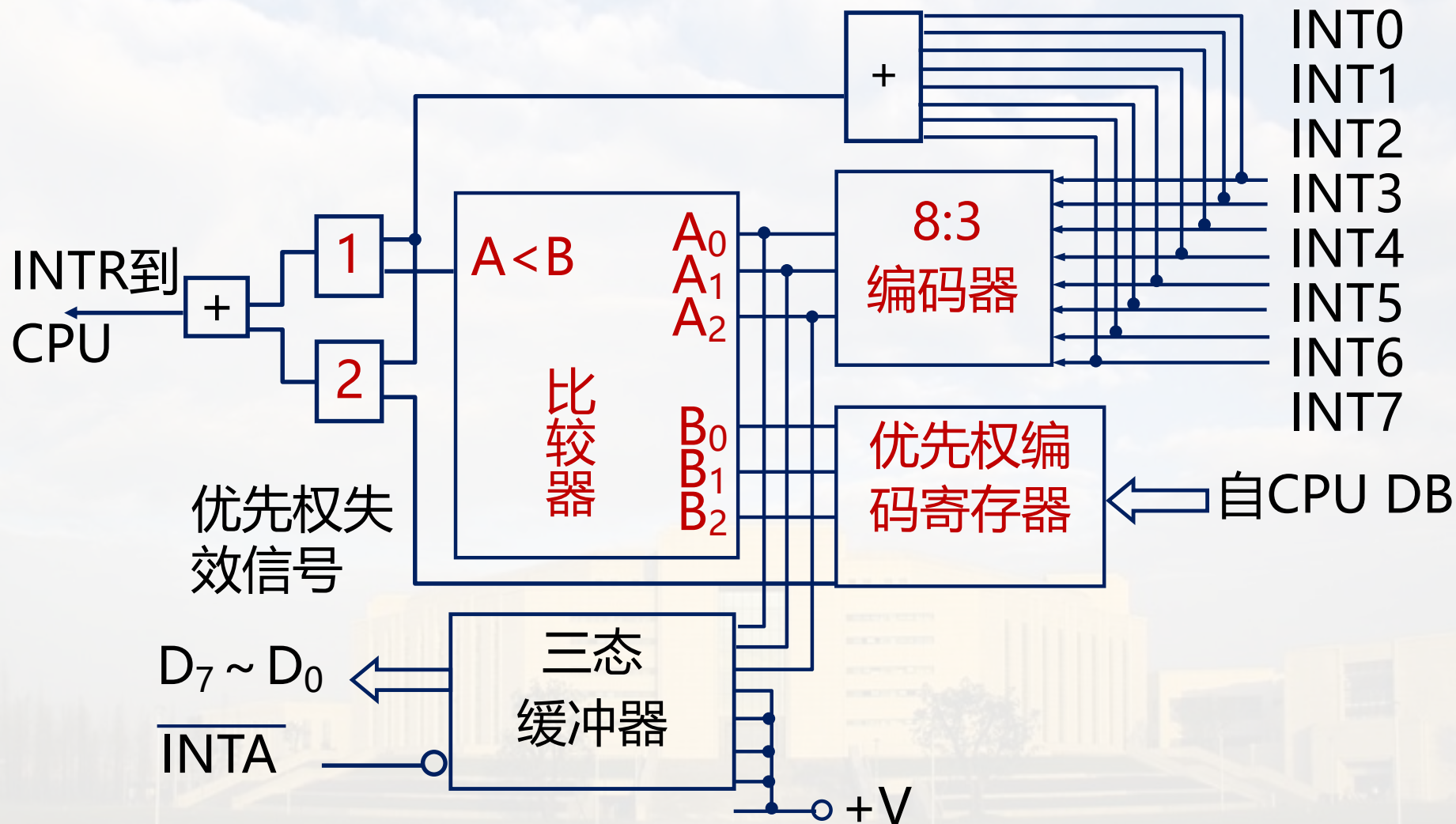


如果 $A_1A_0 < B_1B_0$, 则比较器的引脚 "A < B" 输出1, INT_i 可通过 "与门1" 到达CPU; 如果 $A \geq B$, 则引脚 "A < B" 输出0

- (1) 当编码器有多个输入(INT_i)同时为1时, 编码器只输出优先级最高的编码;
- (2) 正在处理的中断优先权编码通过CPU执行软件送往优先权寄存器;
- (3) CPU响应中断并进入中断响应周期送出INTA信号, 打开三态缓冲器, 新产生的中断对应的中断源的优先权编码送往CPU; CPU识别后产生新的中断向量, 转去执行优先级更高的中断服务程序;
- (4) 如果当前CPU没有执行任何中断服务程序, 则设置优先权失效信号为“1”, 使任何请求信号都 INT_i 能通过“与门2”到达CPU。

三、优先权逻辑与屏蔽技术

如何设计一个支持八个中断源的并行判优器？（了解）



INTR到CPU

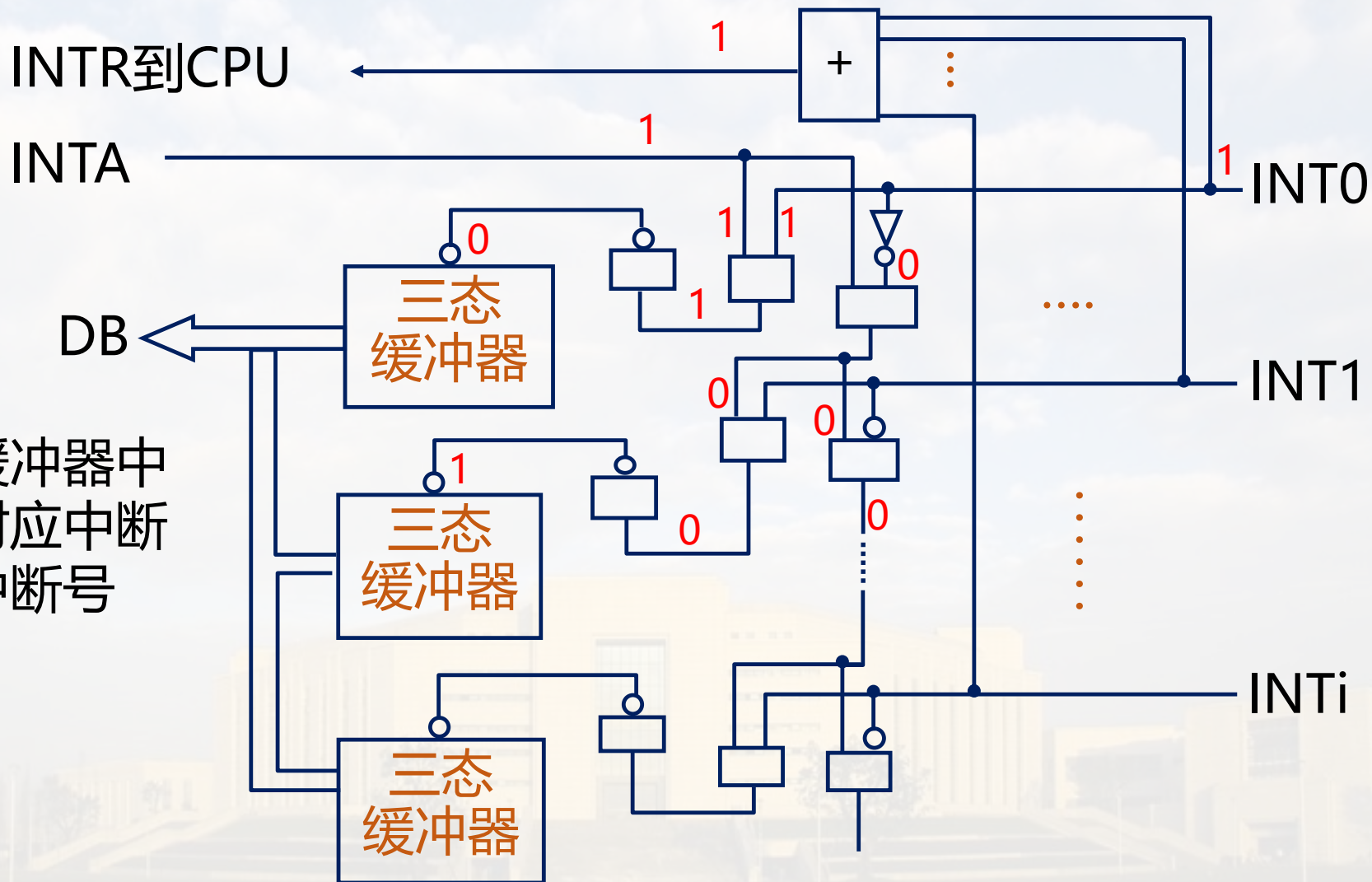
INTA

DB

三态缓冲器

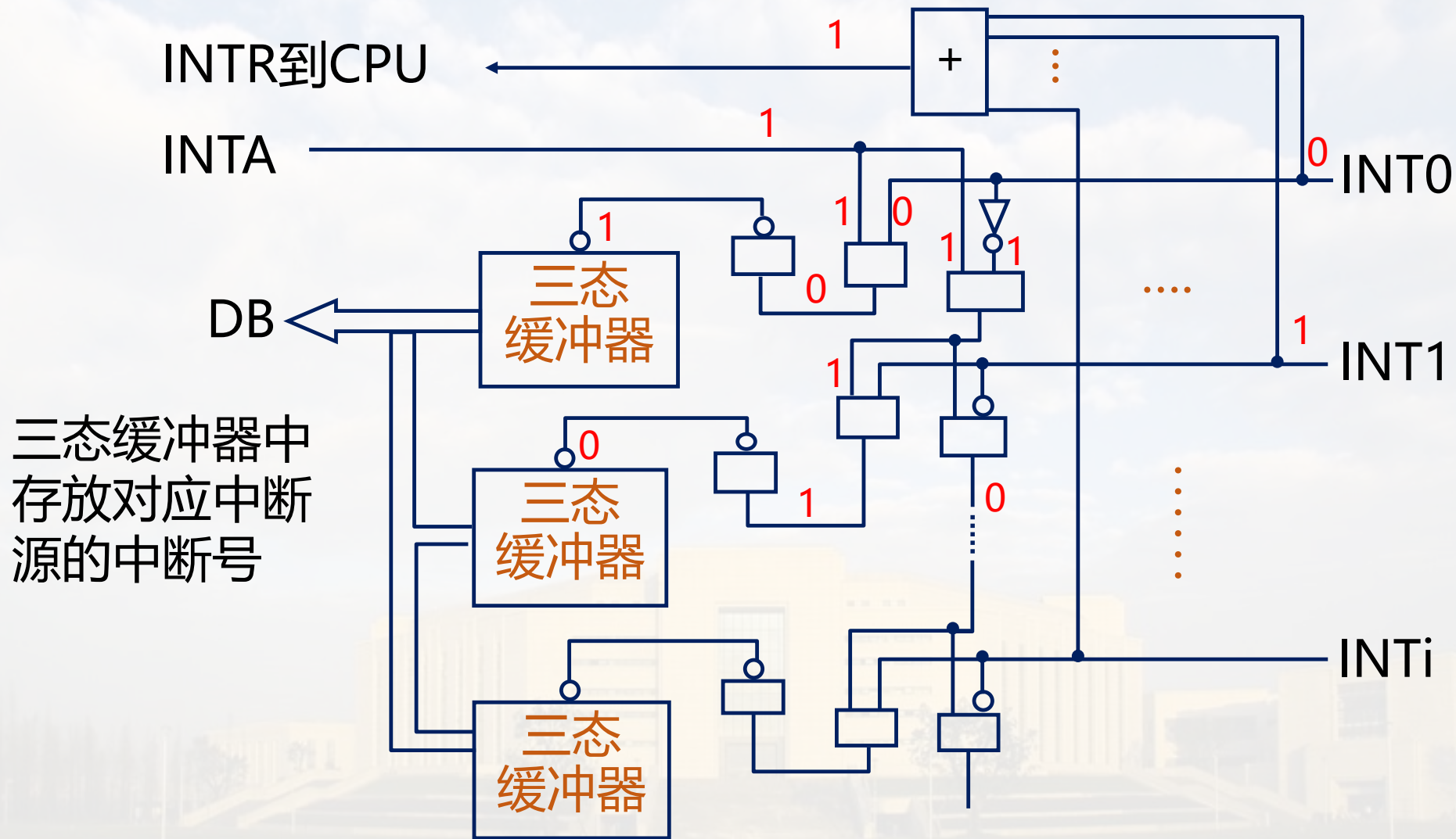
三态缓冲器

三态缓冲器



三、优先权逻辑与屏蔽技术

③ 链式优先权判优逻辑 (了解)



四、服务程序入口地址的获取方式

如何获取中断程序的入口地址？ **获取方式** { 向量中断
非向量中断

1、非向量中断

将服务程序入口组织在查询程序中，CPU响应时，执行查询程序，查询中断源，转入相应的服务程序。



如何转入查询程序？

CPU会提供相应支持, 在中断响应周期提供并转向一个固定内存地址, 将查询程序从该地址存放

四、服务程序入口地址的获取方式

2、向量中断

1) 向量中断相关的几个概念

a. 中断向量:

采用向量化的中断响应方式，将中断服务程序的入口地址及其程序状态字存放在特定的存储区中，所有的中断服务程序入口地址和状态字一起，称为中断向量。

b. 中断向量表:

即用来存放中断向量的一种表。在实际的系统中，常将所有中断服务程序的入口地址（或包括服务程序状态字）组织成一个一维表格，并存放于一段连续的存储区，此表就是中断向量表。



四、服务程序入口地址的获取方式

c. 向量地址:

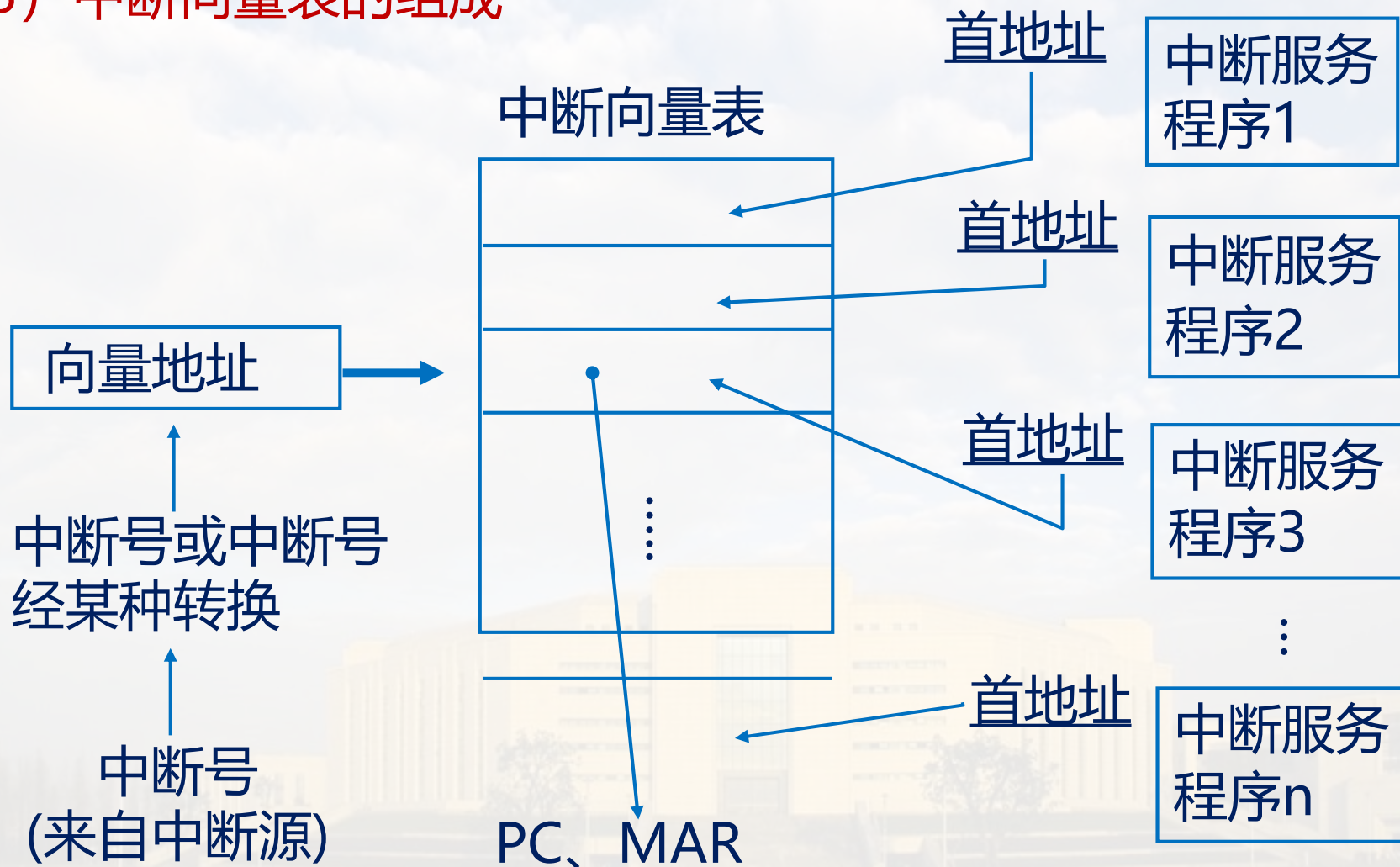
访问中断向量表的地址码，即读取中断向量所需的地址（也可称为中断指针）。

2) 向量中断

将各个中断服务程序的入口地址（或包括状态字）组织成中断向量表；响应中断时，由硬件直接产生对应于中断源的向量地址；据此访问中断向量表，从中读取服务程序入口地址，由此转向服务程序的执行。这些工作中断周期IT中由硬件直接实现。

四、服务程序入口地址的获取方式

3) 中断向量表的组成



比如：在IBM PC机中，中断向量表在主存中占用0—1023号地址单元（即1K），每个中断源占用4个单元，因此，该表中可存放256个中断源。

组成	{	专业区	中断类型码：0—4型
		系统保留区	中断类型码：5—31型
		用户扩展区	中断类型码：32—255型

四、服务程序入口地址的获取方式

中断类型码 向量地址 中断向量表

0型	0000—0003	(H)	
1型	0004—0007	(H)	专用区
• • • • •			
4型	0011—0014	(H)	
5型	0015—0018	(H)	
• • • • •			系统保留区
31型	007C—007F	(H)	
32型	0080—0083	(H)	
• • • • •			用户扩展区
255型	03FC—03FF	(H)	

4) 如何从中断向量表中获取中断服务程序入口地址

关键是如何形成向量地址 $\xrightarrow{\text{中断向量表}}$ 中断服务程序入口地址

向量地址的
形成

硬中断：向量地址 = 中断类型码 \times 4 (IBM PC
机中每个中断源所占字节数)

软中断：向量地址 = 中断号 \times 4 (IBM PC机中
每个中断源所占字节数)

四、服务程序入口地址的获取方式

例1.模型机向量表

M按**字**编址。一个入口地址16位，占一个编址单元。

向量表

向量地址 = 中断号 + 2
(单元地址)

2#
3#

入口地址0
入口地址1
⋮

0号中断源
1号中断源

例2.IBM PC向量表

(从主存0#单元开始安排)

M按字节编址。一个入口地址32位，占4个编址单元。

向量表

0#
4#

入口偏移0
入口基址0
入口偏移1
入口基址1
⋮

0号中断源
1号中断源

向量地址 = 中断号 × 4

四、服务程序入口地址的获取方式

例3：中断类型码108型，求其中断服务程序入口地址。

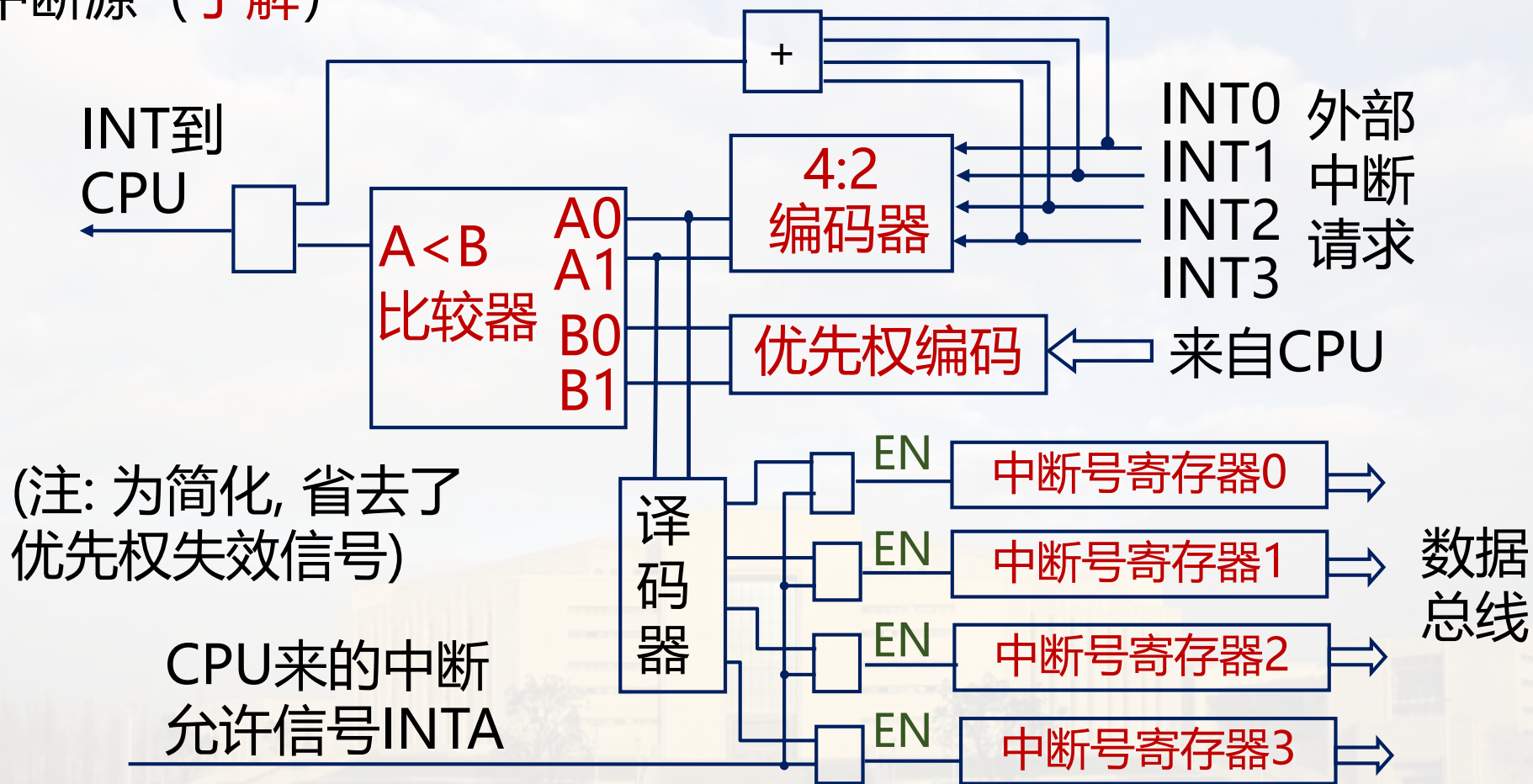
(IBM PC机)

向量地址 = $108 \times 4 = 432 \rightarrow 110110000B$ 即 $01B0H \rightarrow$ 得

出中断服务程序入口地址

四、服务程序入口地址的获取方式

中断判优及响应例(设置存放中断号的可编程寄存器): 假设有四个中断源 (了解)



四、服务程序入口地址的获取方式

3、响应中断的条件

- ① 有中断请求信号发生，如 $IREQ_i$ 或 $INT\ n$ 。
- ② 该中断请求未被屏蔽。
- ③ CPU处于开中断状态，即中断允许触发器 $TIE_n=1$ （或中断允许标志位 $IF=1$ ）。
- ④ 没有更重要的事件要处理（如因故障引起的内部中断，或是其优先级高于程序中中断的DMA请求等）。
- ⑤ CPU刚刚执行的指令不是停机指令。
- ⑥ 在一条指令结束时响应（因为程序中断的过程是程序切换过程，不能在一条指令执行的中间就切换）。

四、服务程序入口地址的获取方式

4、响应过程

向量中断方式：

CPU执行中断
隐指令
(硬件完成)

发响应信号INTA，进入中断周期

关中断，保存断点

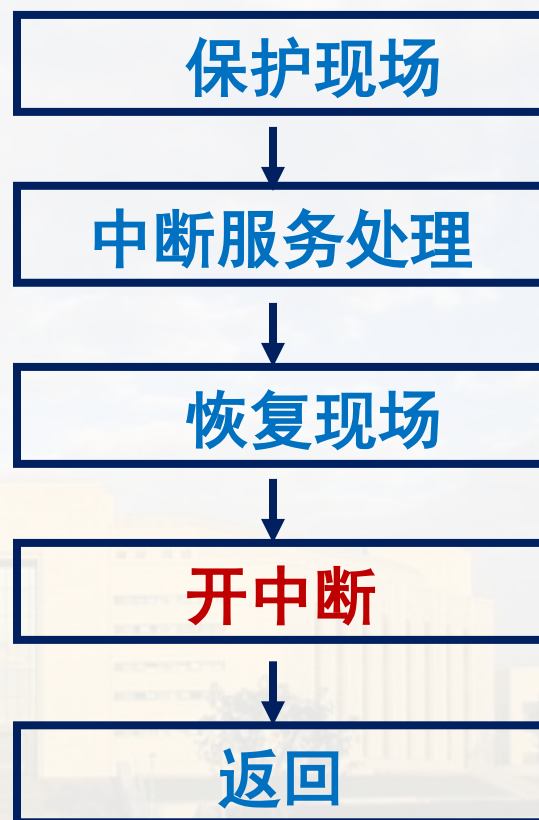
获得中断号，转换为向量地址，查向量表

取中断向量，转中断服务程序

1、中断处理：CPU执行中断服务程序。

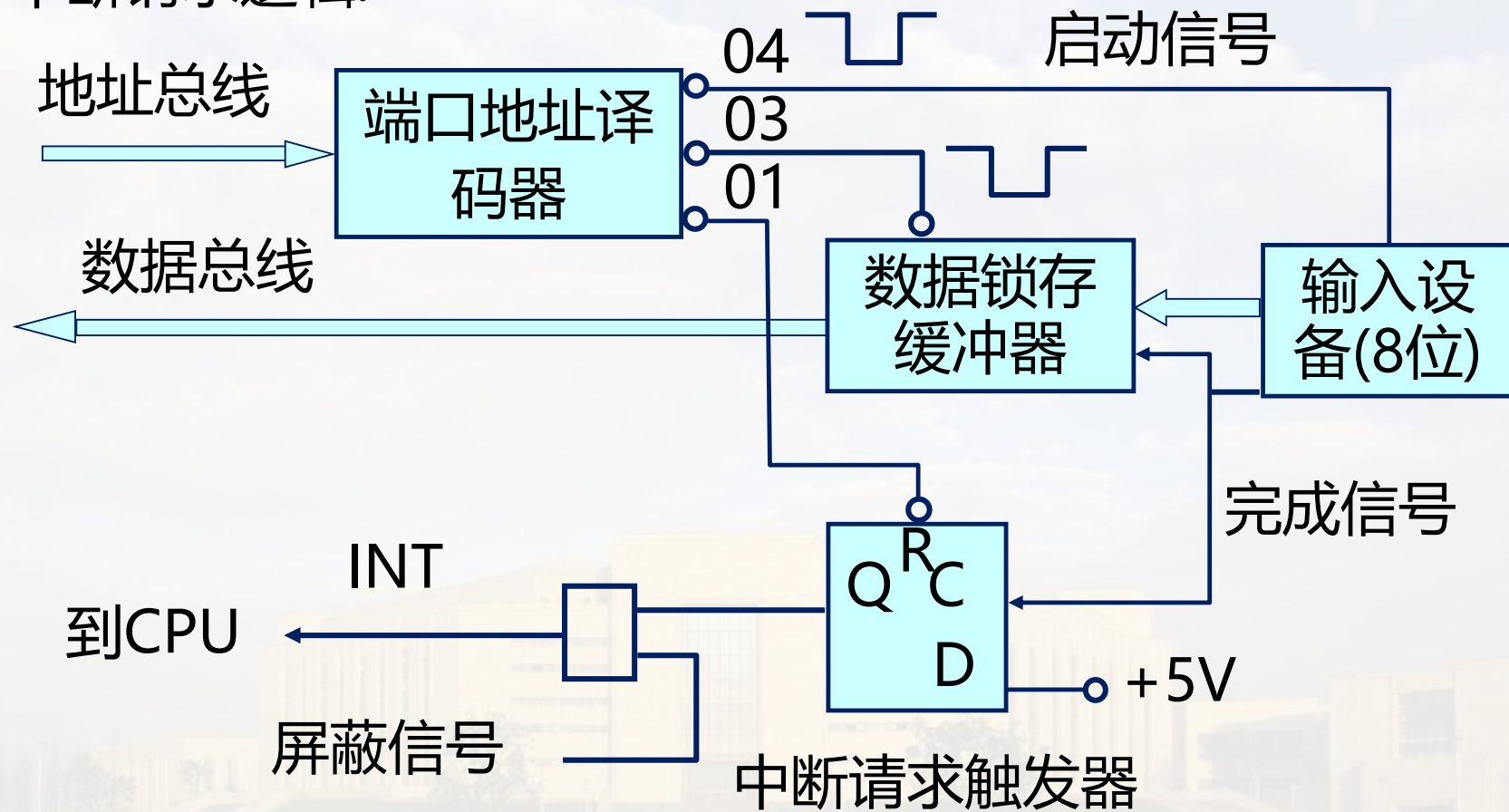
1) 单级中断

CPU响应后只处理一个中断源的请求，处理完毕后才能响应新的请求。



例：用中断方式实现I/O (用前述查询的例子, 了解)

中断请求逻辑:



按中断方式的软件设计(假设输入5个数据):

启动端口地址04H, 数据端口地址03H, 复位端口地址01H

主程序:

MOV (2000H), DSTOR ; 将存放数据的内存首地址存入
2000H单元

MOV (2001), 05 ; 将计数初值存入2001H单元

OUT 01, AL ; 请求触发器复位

STI ; 开中断

OUT 04, AL ; 启动输入设备

.....
.....
.....
.....
.....

} CPU继续执行主程序

中断服务程序(保存断点即返回地址通常由硬件自动完成):

PUSH AX	; AX压堆栈保存	} 保护现场
PUSH BX	; BX压堆栈保存	
PUSH CX	; CX压堆栈保存	
PUSH DI	; DI压堆栈保存	
IN AL, 03	; 读数据	
MOV DI, (2000H)	; 取出内存地址	
MOV [DI], AL	; 数据送内存 (寄存器间址寻址)	
INC DI	; 地址加1	
MOV (2000H), DI	; 存回内存地址	

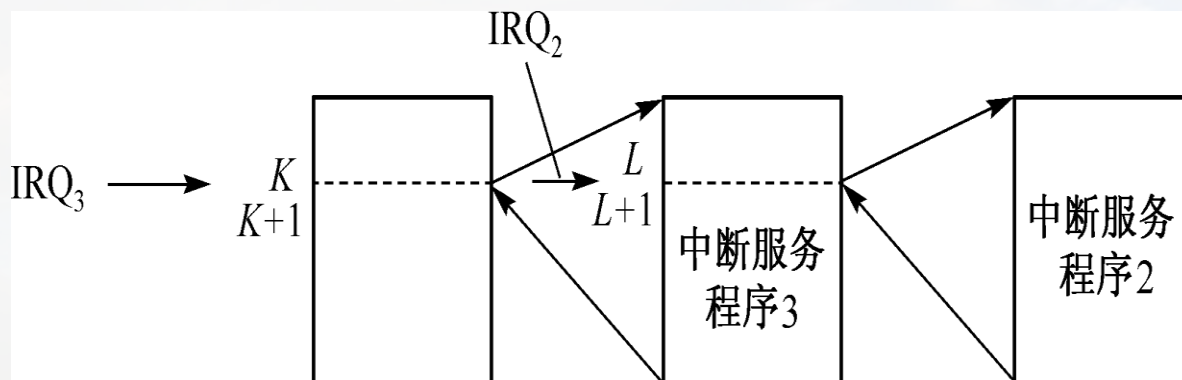
POP AX	; 恢复AX	} 恢复现场
POP BX	; 恢复BX	
POP CX	; 恢复CX	
POP DI	; 恢复DX	
DEC (2001H)	; 计数值减1	
JZ ROUT	; 计数值为0转移	
OUT 01, AL	; 请求触发器复位	
OUT 04, AL	; 启动输入设备	
ROUT: STI	; 开中断	
IRET	; 中断返回	

本例为单级中断模式！

1、中断处理：CPU执行中断服务程序。

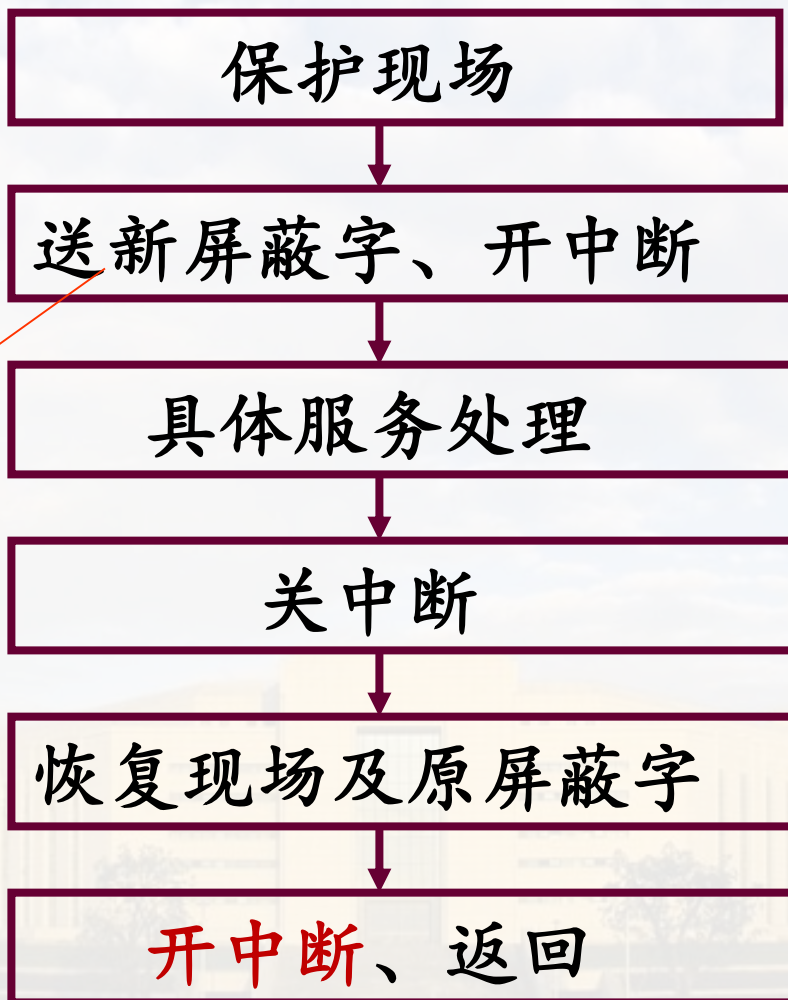
2) 多重中断：

在某次中断服务过程中，允许响应处理更高级别的中断请求。



多重中断流程：

禁止同级或更低级别的请求，
开放更高级别的请求。



六、中断接口

1、组成（寄存器级）

1) 寄存器选择

对接口寄存器寻址。

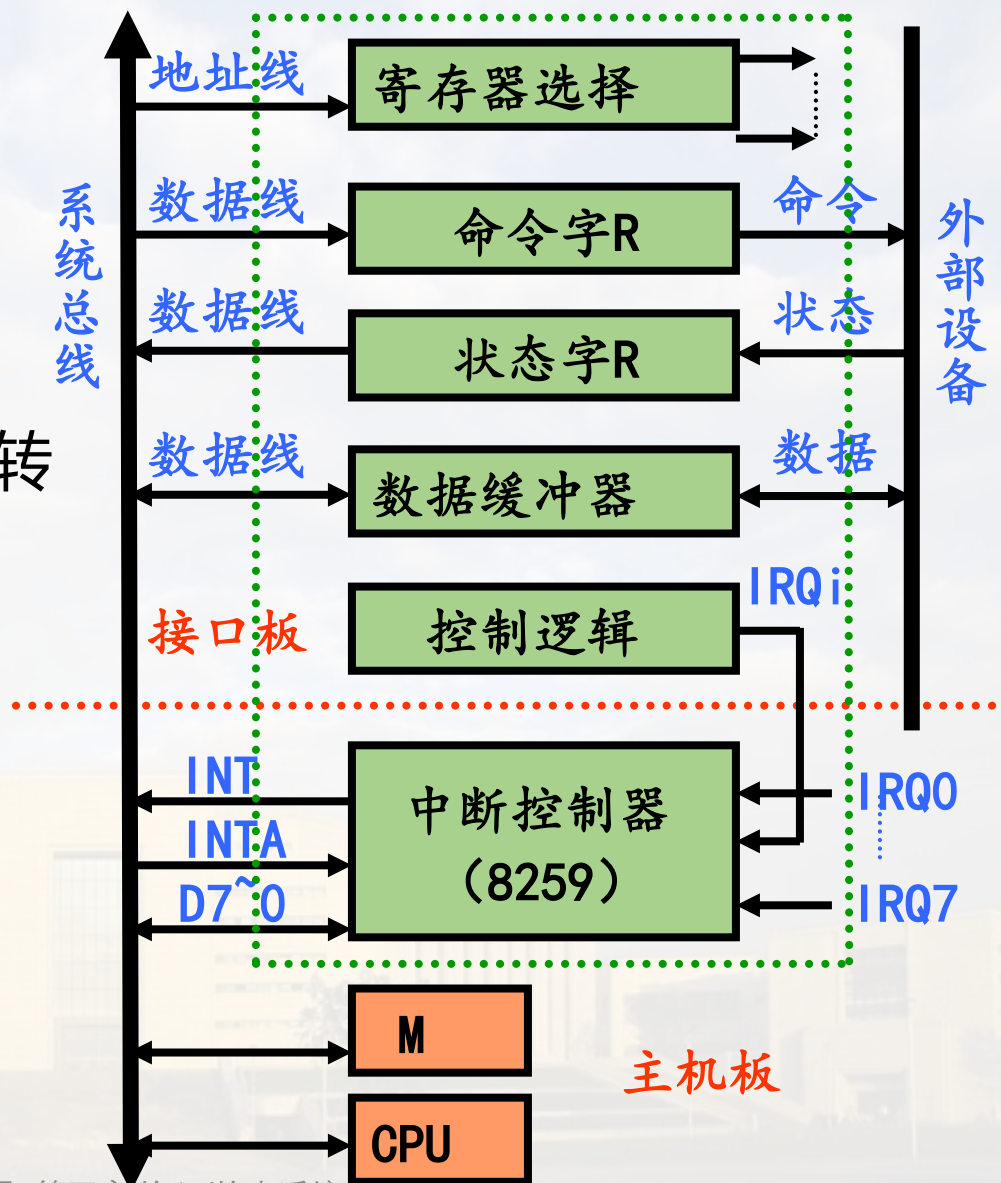
2) 命令字寄存器

接收CPU发向外设的命令字，转换为相应操作命令送外设。

命令字格式的拟定：

用代码表示各种命令

代码位数
代码含义



六、中断接口

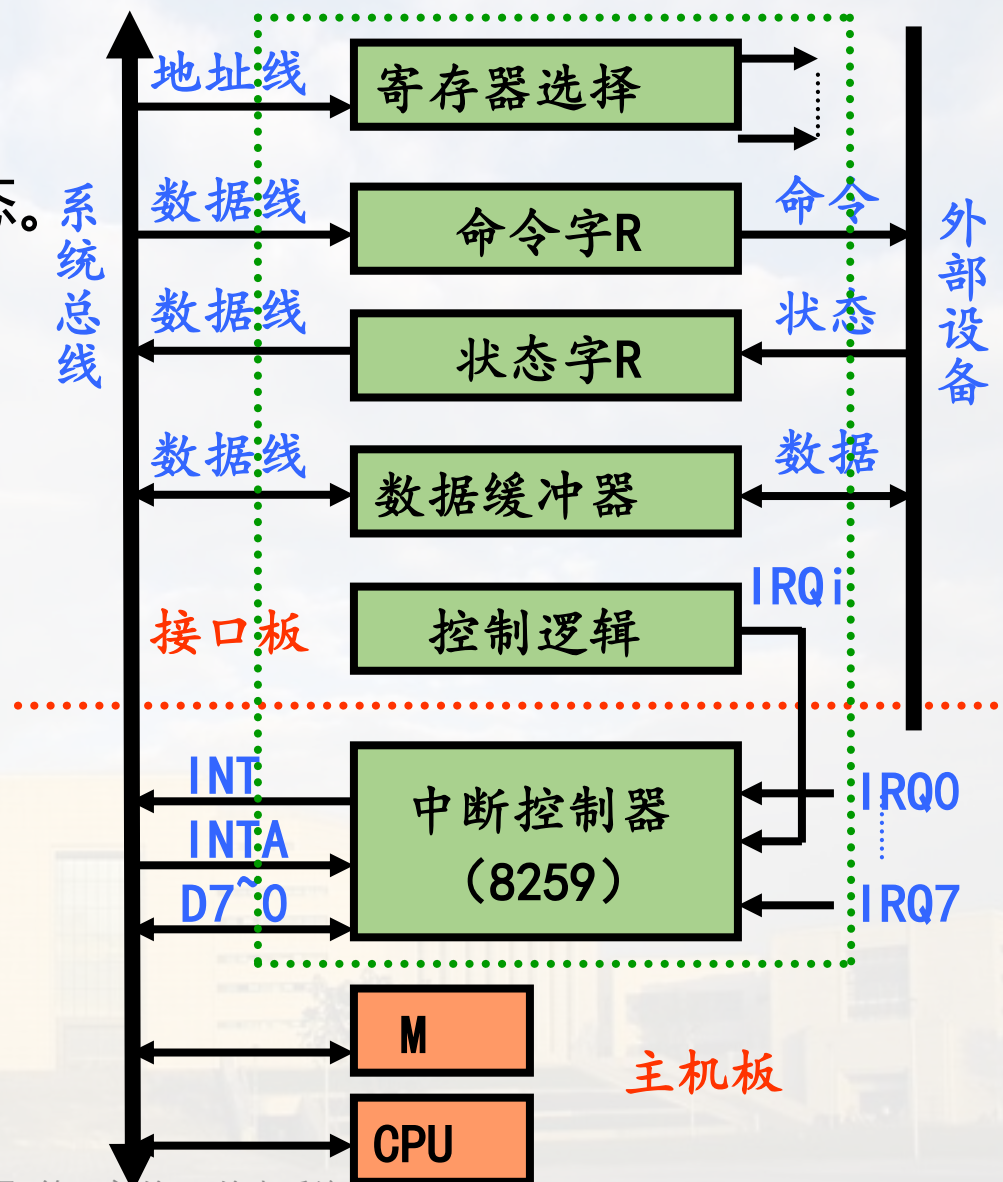
3) 状态字寄存器

反映设备和接口的运行状态。

状态字格式的拟定：
用代码表示各种状态。

4) 数据缓冲器

传送数据，实现缓冲。



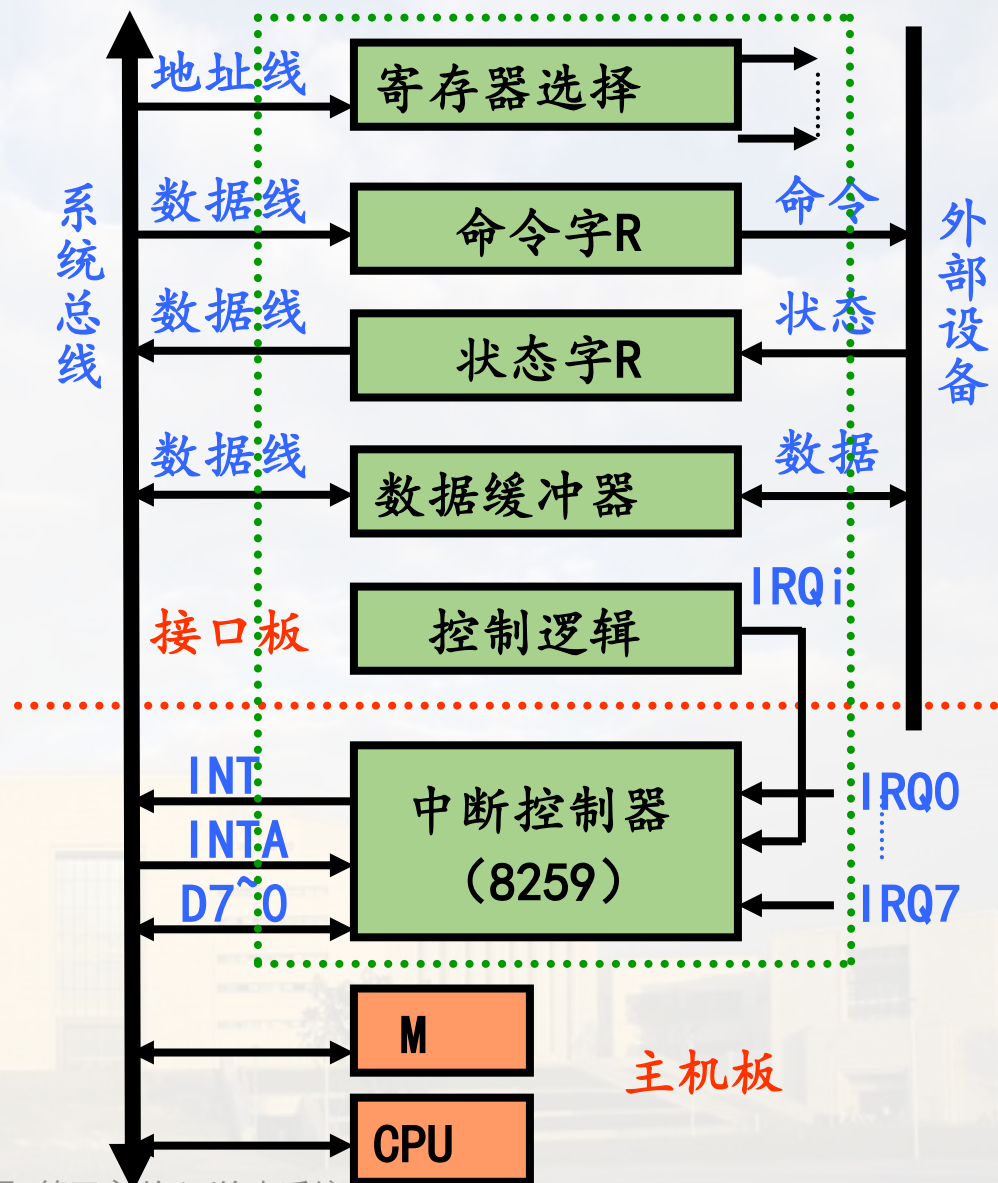
六、中断接口

5) 控制逻辑

中断请求信号产生逻辑
与主机间的应答逻辑
控制时序（比如串并转换）
针对设备特性的逻辑
智能控制器

6) 公用中断控制器

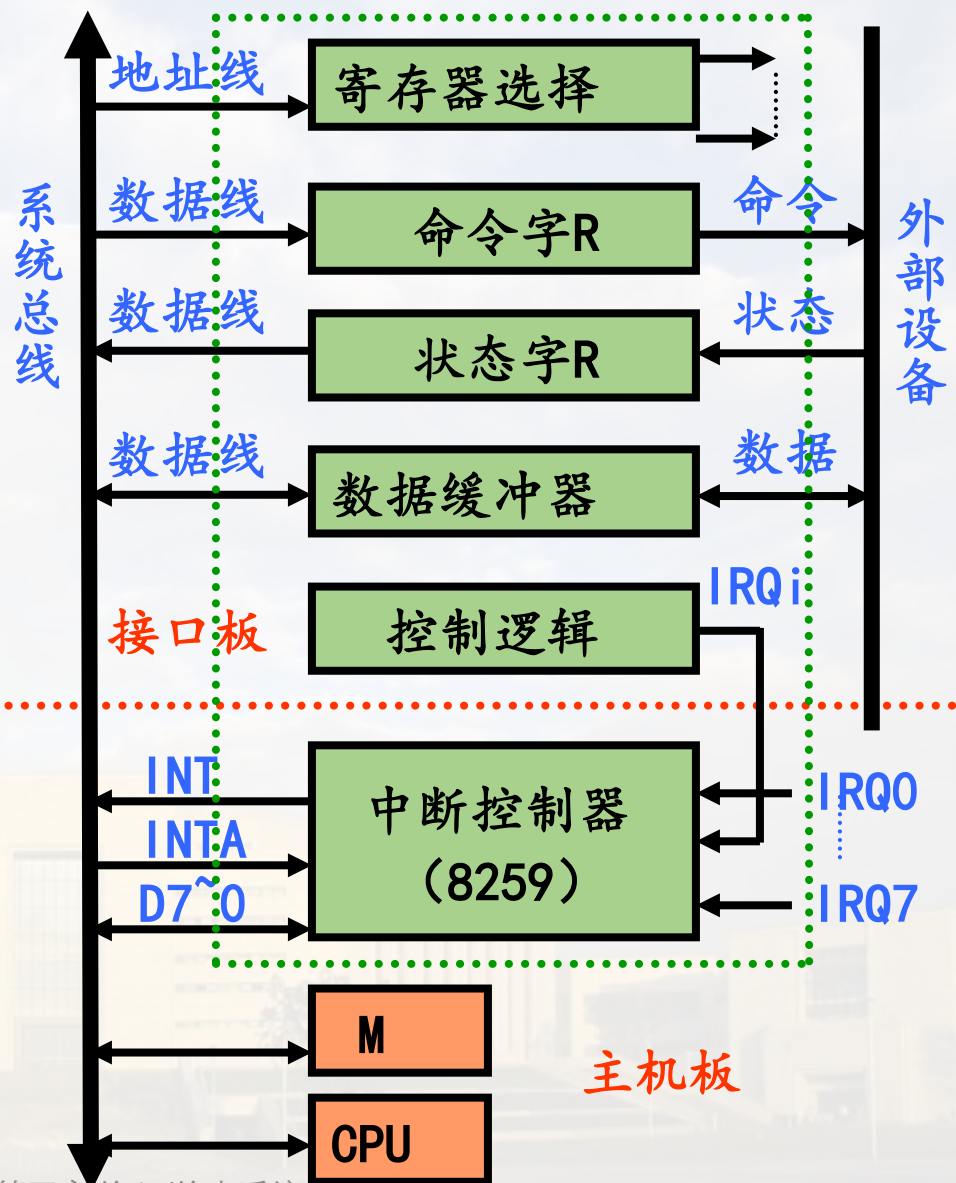
接收外设请求，判优，
送出公共请求；
接收中断批准，送出中断
号（中断类型码）。



六、中断接口

2、工作过程（外中断）

- 1) **初始化**：设置工作方式，送屏蔽字，送中断号(确定高位)。
- 2) **发启动命令**(送命令字)，启动设备。
- 3) **设备完成工作**，申请中断。
- 4) **中断控制器**汇集各请求，经屏蔽、判优，形成中断号，并向CPU送INT。
- 5) **CPU响应**，发INTA。
- 6) **中断控制器**送出中断号。
- 7) **CPU执行中断隐指令**操作，进入服务程序。



六、中断接口

3、接口设计

涉及命令字、状态字格式的拟定，中断源的扩展。

例.模型机需扩展两个外中断源，共用一个中断号。

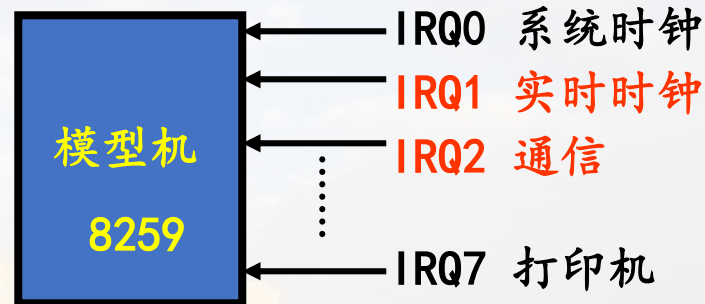
主机发向外设的命令包括：启动、停止、数据选通；

外设的状态包括：忙、完成、出错。

为两个扩展中断源设计中断接口。

模型机的外中断源安排：

通过IRQ2进行扩展。



1) 接口组成

两个扩展中断源共用一个接口。

命令字格式：

状态字格式：

5	4	3	2	1	0
启动1	停止1	选通1	启动2	停止2	选通2
忙1	完成1	出错1	忙2	完成2	出错2

六、中断接口

2) 判断中断源

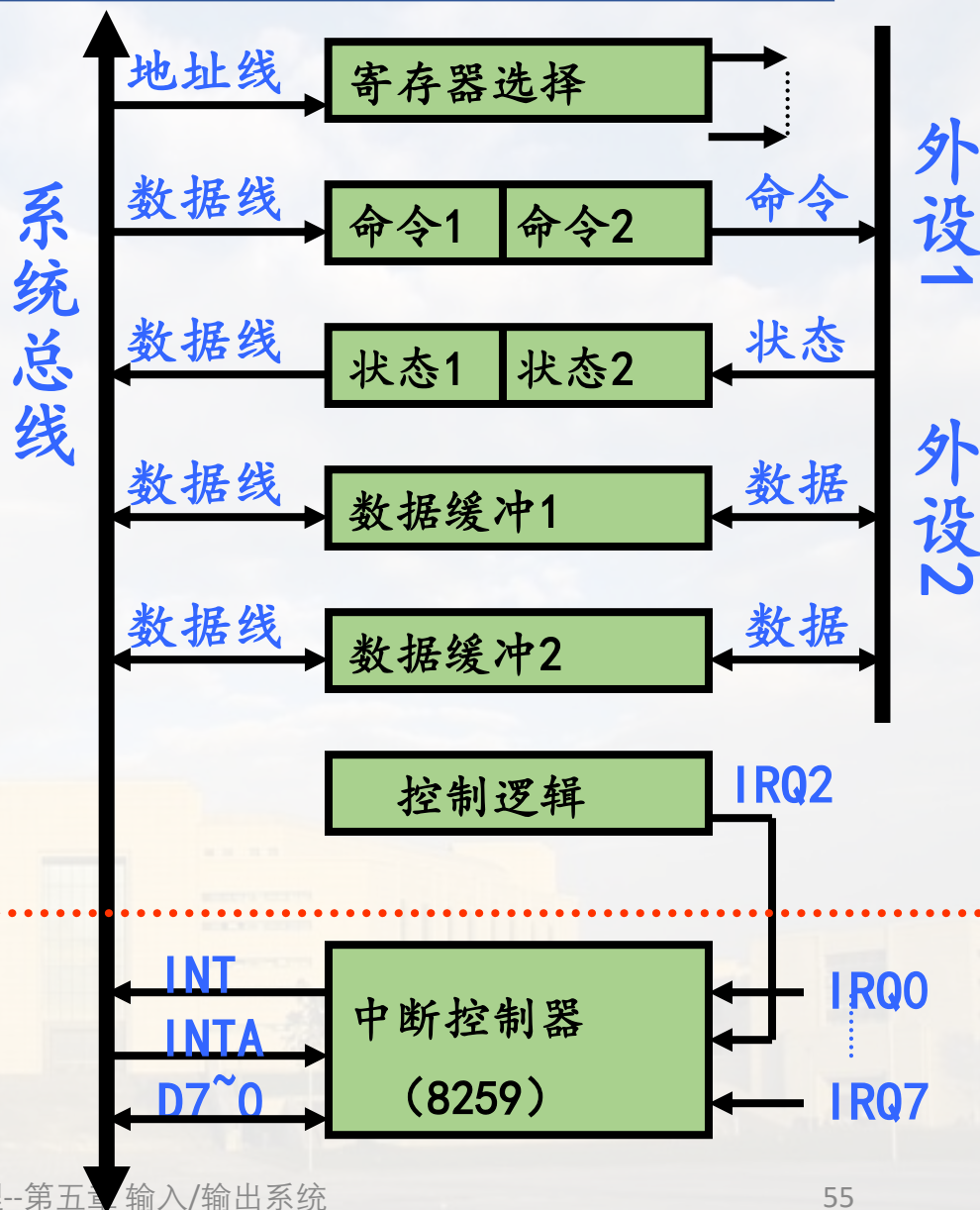
向量中断与非向量中断相结合
(软件扩展)

请求1与请求2在控制逻辑中形成公共请求IRQ2，送入8259参加判优；

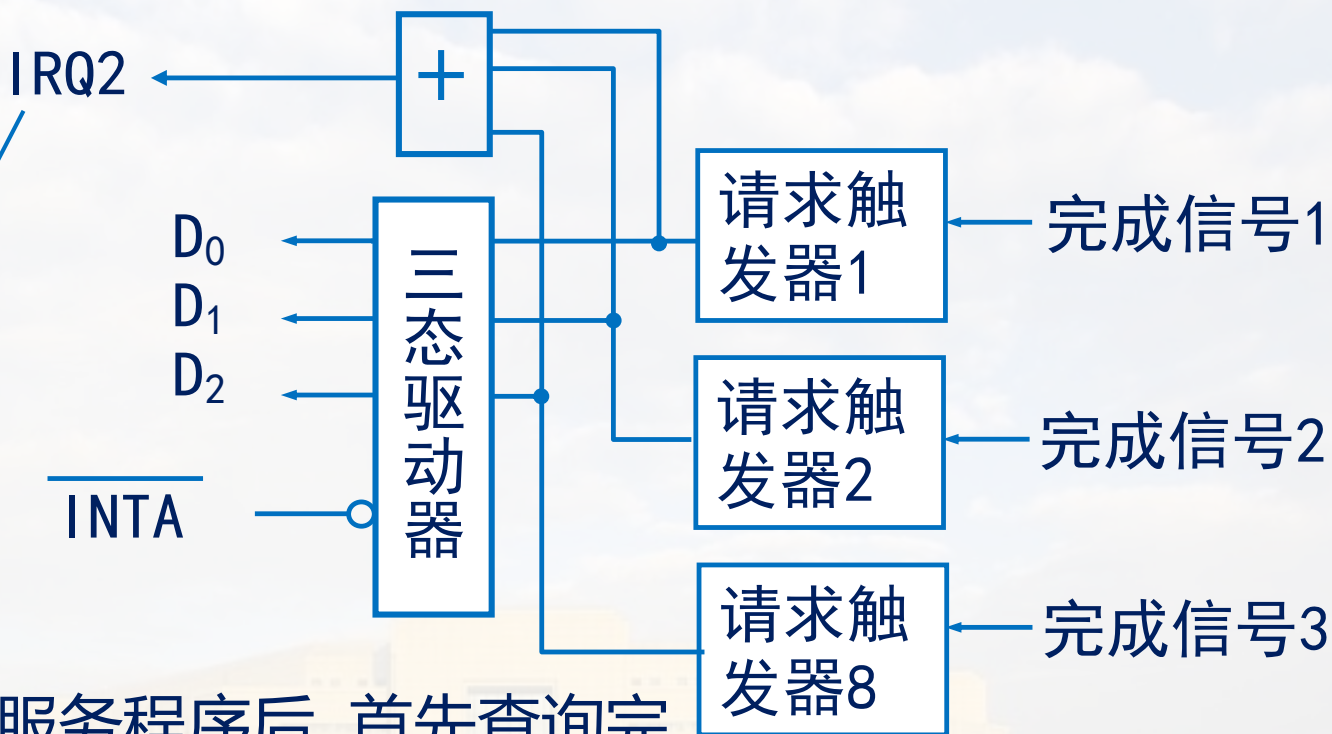
CPU响应后执行IRQ2服务程序
(向量中断过程)；

CPU在IRQ2服务程序中查询各设备状态，判中断源，转入相应设备服务程序

(非向量中断过程)。



— 多个设备共用一个中断源(比如三个设备)



转入IRQ2对应服务程序后, 首先查询完成信号, 再转入对应中断服务程序



谢谢观看

计算机组成原理

2022/11/7



信息与软件工程学院

School of Information and Software Engineering