



计算机组成原理

第五章 输入/输出系统

2025-8-22



信息与软件工程学院
School of Information and Software Engineering



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



- 01. 中断基本概念
- 02. 中断请求过程（外部可屏蔽中断）
- 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

一、中断基本概念

1、定义

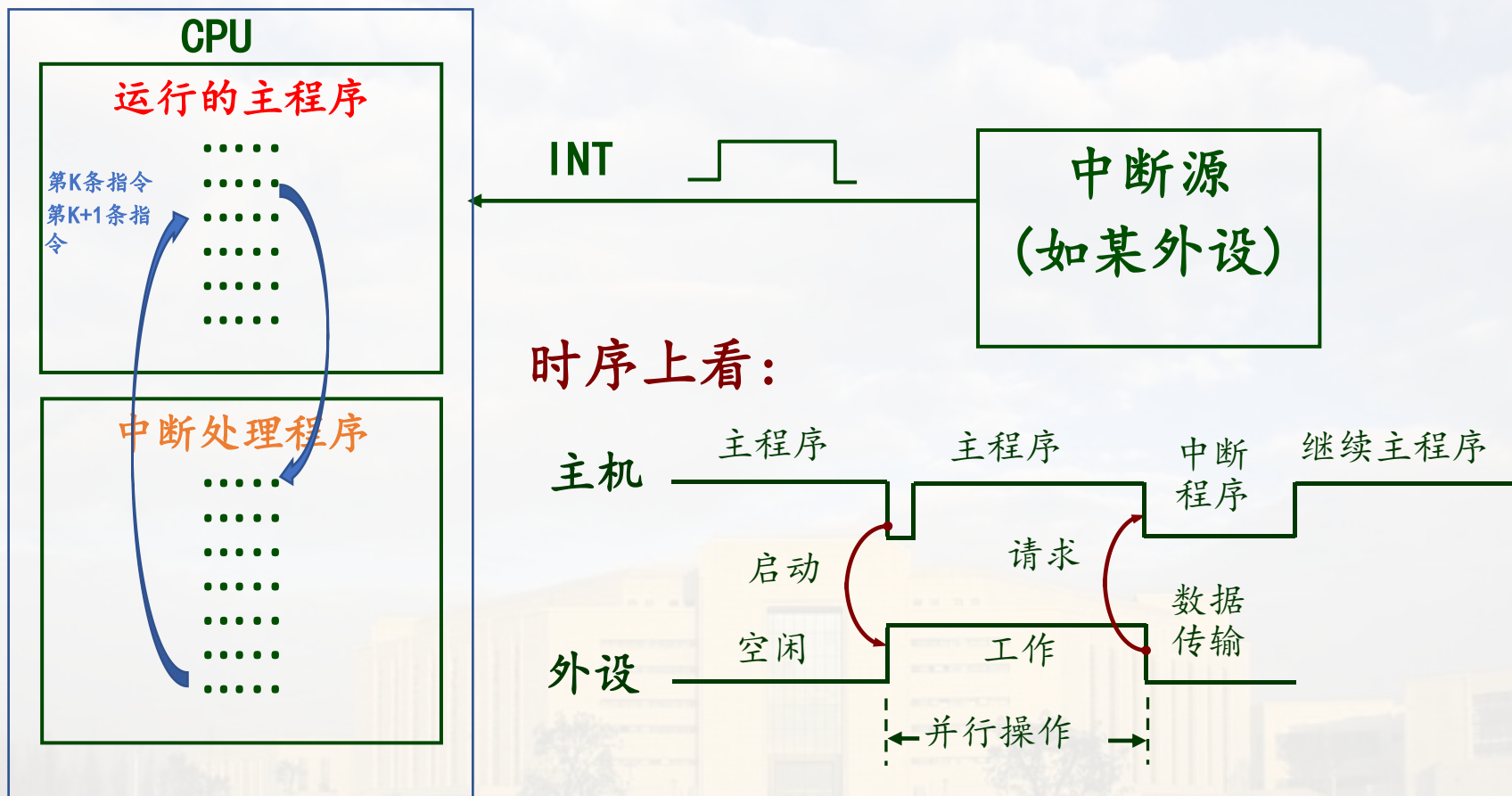
CPU暂时停止现行政程序的执行，转去执行为某个随机事态服务的中断处理程序。处理完毕后自动恢复原程序的执行。

2、实质与特点

1) 实质

程序切换 { 方法：保存断点，获取服务程序入口地址；(中断处理前)
返回断点。(中断处理后)
时间：一条指令完整结束时切换，保证程序的完整性。

中断示意图：



一、中断基本概念

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。

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

一、中断基本概念

3) 可屏蔽中断与非屏蔽中断

可通过屏蔽字屏蔽该类请求；关中断时不响应该类请求。

该类请求与屏蔽字无关；请求的响应与开/关中断无关。

4) 向量中断与非向量中断

由硬件提供服务程序入口地址

由软件提供服务程序入口地址

4、中断典型应用

1) 管理中、低速I/O操作

2) 处理故障

一、中断基本概念

3) 实时处理

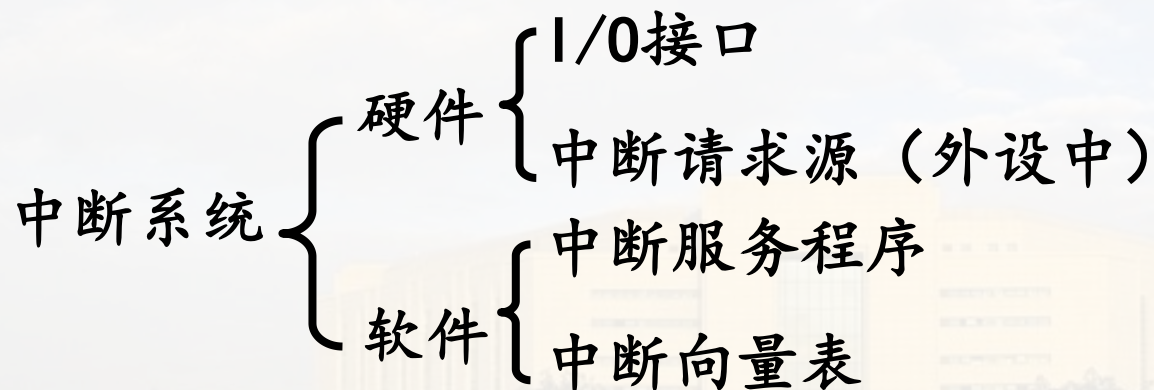
某事件出现的实际时间内及时处理，不是批量处理。

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



1) 中断请求源

- 外部硬件中断源：8种， $IREQ_0 \sim IREQ_7$
- 内部中断源：掉电、溢出、校验错中断等
- 软中断：模型机软中断为 $INT11 \sim INTn$

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、中断请求过程（外部可屏蔽中断）



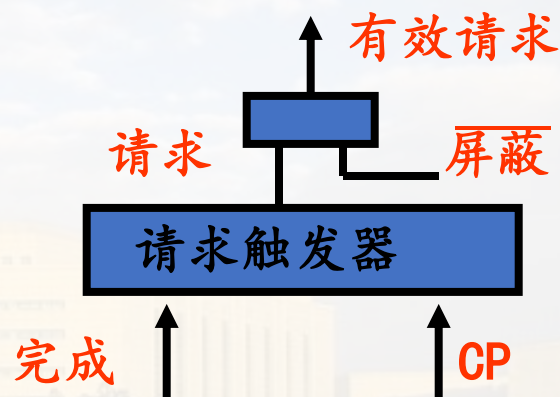
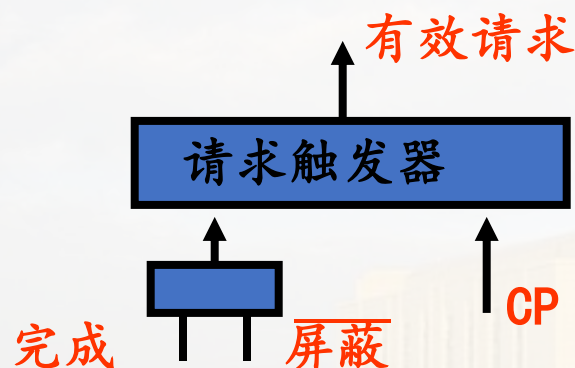
1、如何产生中断请求？

{ 外设工作完成：“完成”标志为1
CPU允许请求：“屏蔽”标志为0

先“屏蔽”，后请求

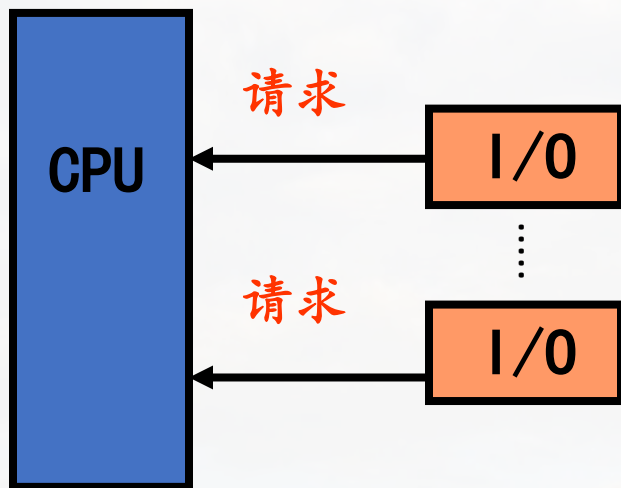
分散“屏蔽”

先请求，后“屏蔽”

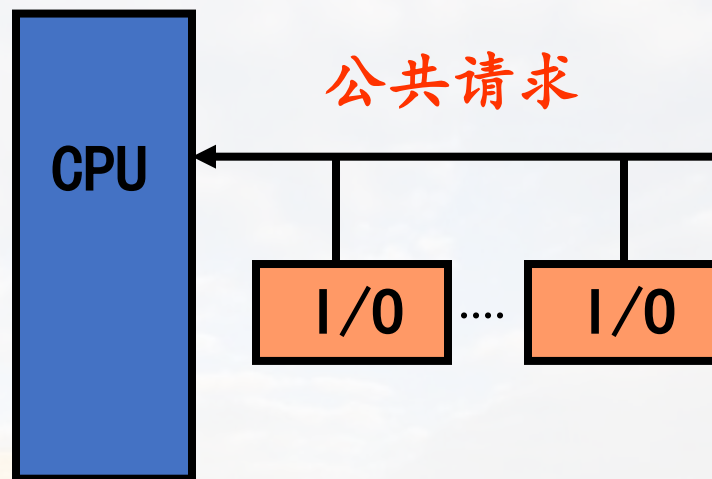


2、如何传送中断请求？

a. 使用单独请求线

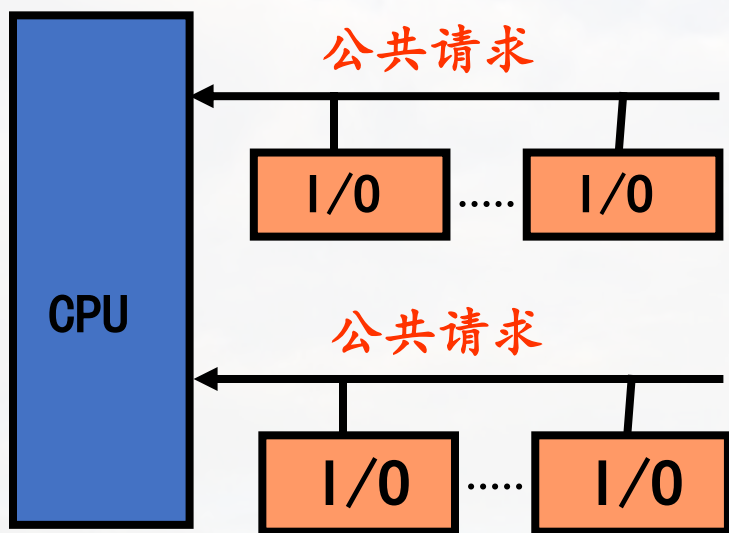


b. 使用公共请求线

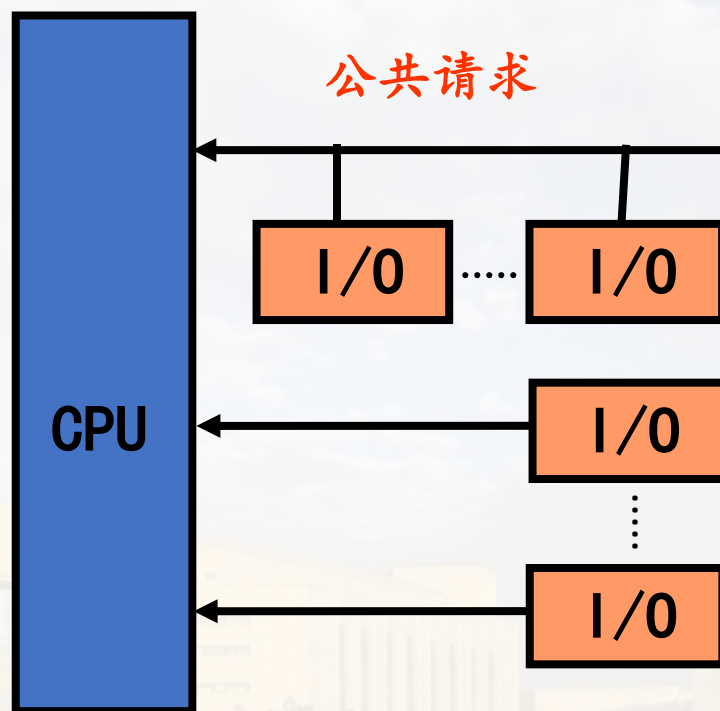


二、中断请求过程（外部可屏蔽中断）

c. 使用二维结构



d. 使用独立与公共请求线



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。

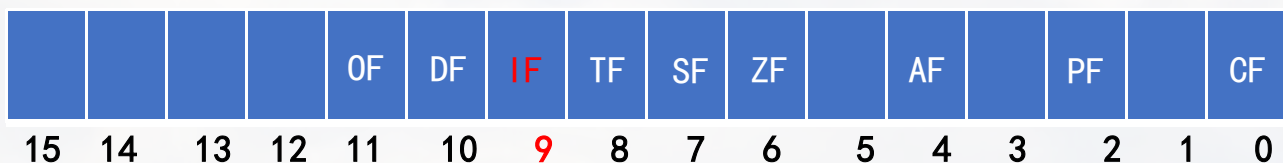
1. CPU（现执行程序）与中断请求间的判优

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。

开关中断 { $IF=1$ ，开中断
 $IF=0$ ，关中断

IF (Interrupt Enable Flag)

中断允许标志位。IF=1使CPU可以响应**可屏蔽**中断请求。

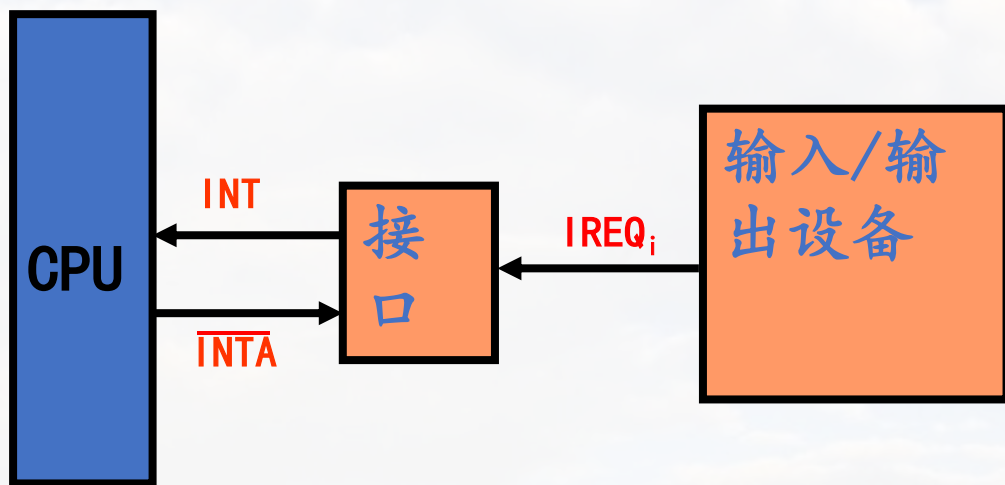


8088 /8086的标志寄存器FLAGS的组成

6个状态位：CF(进位标志位)、PF（奇偶标志位）、AF（辅助进位标志位）、ZF（零标志位）、SF（符号标志位）、OF（溢出标志位）

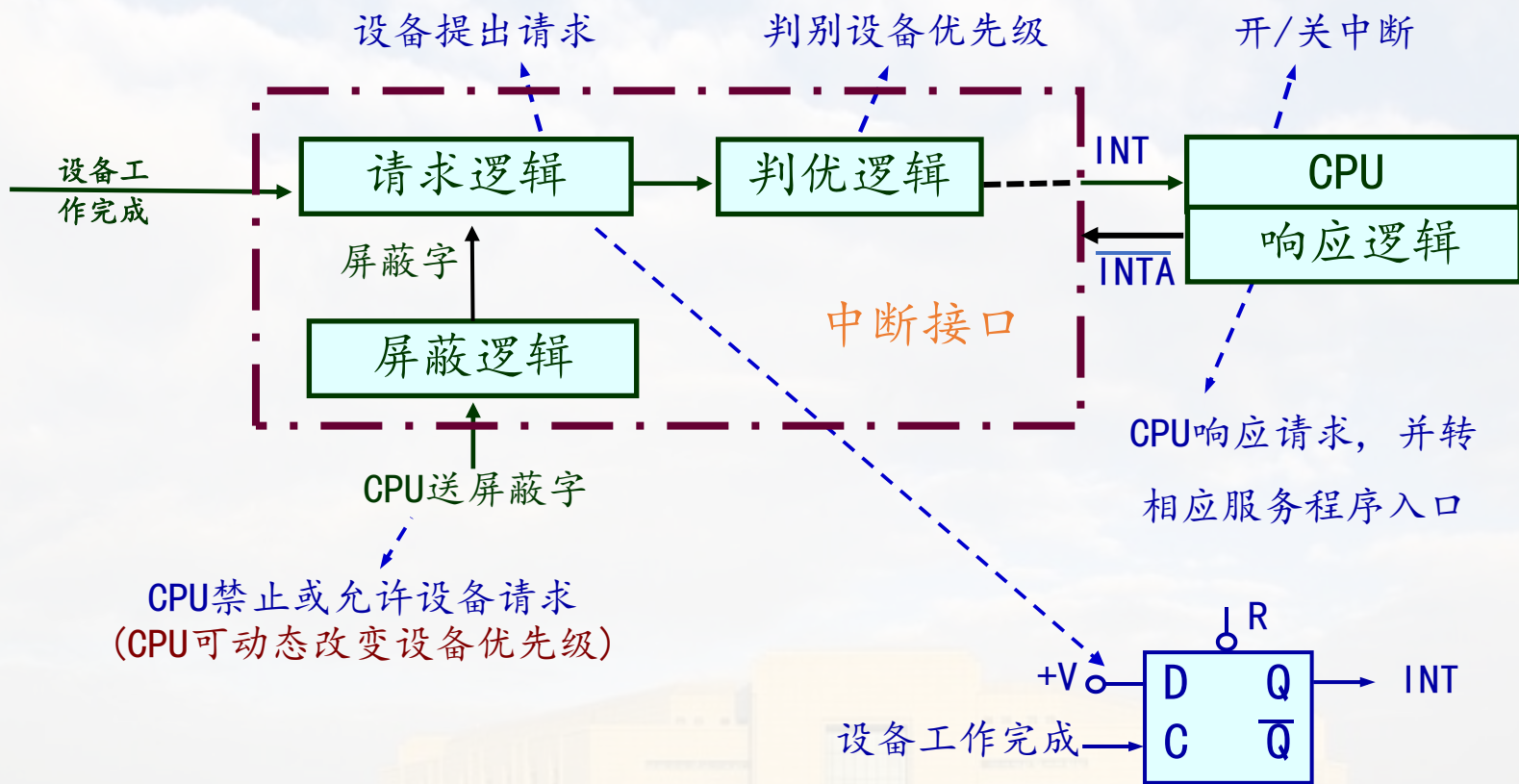
3个控制标志位：IF（中断允许标志位）、TF（陷阱标志位）、DF（方向标志位）

INT、IREQ_i 区别:



三、优先权逻辑与屏蔽技术

屏蔽、开/关中断区别：



区分：CPU对请求的屏蔽和对请求的响应

送屏蔽字

开/关中断

2. 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。

3. 各外设请求的判优方式

a. 软件判优

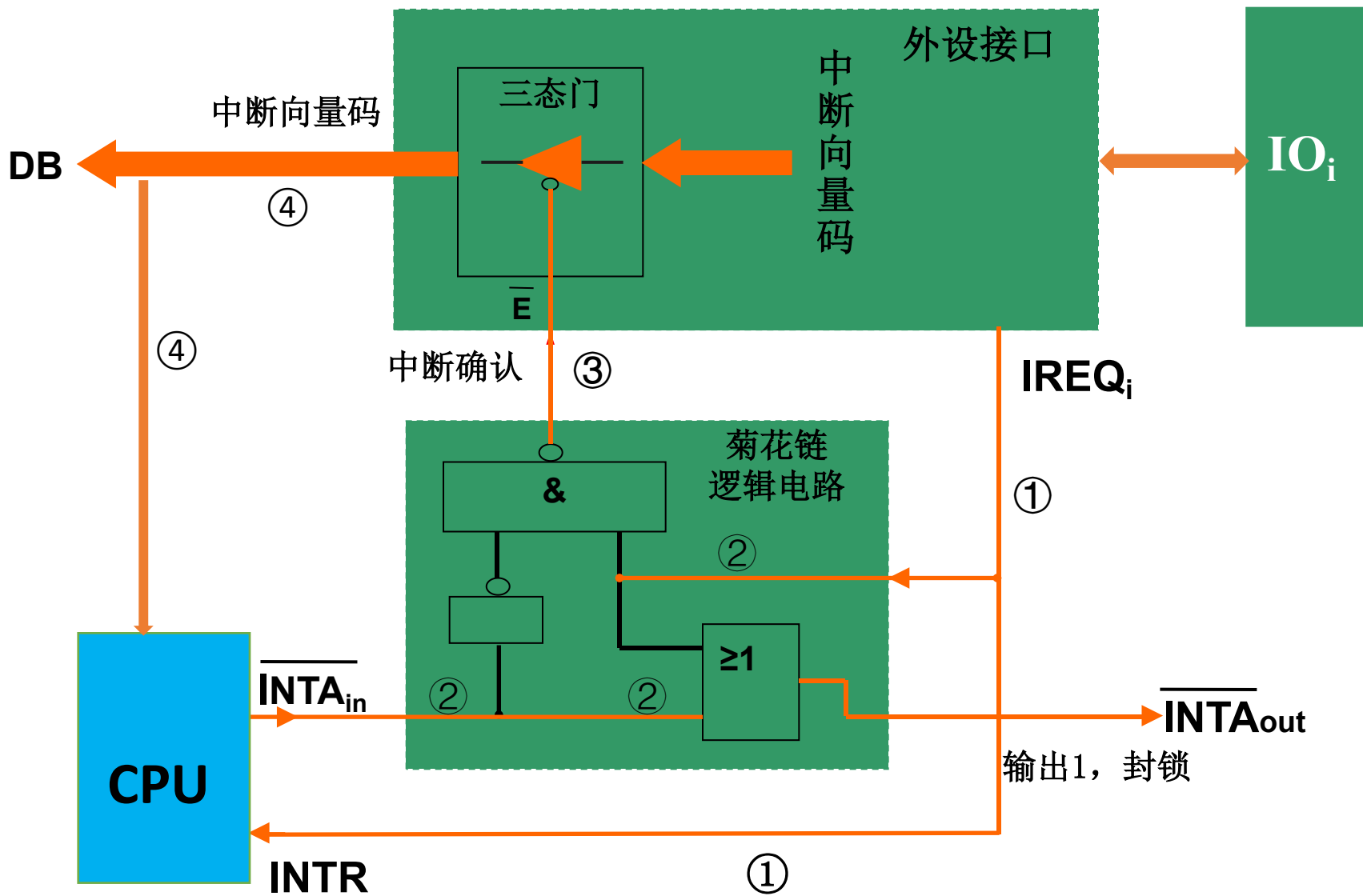
由程序查询顺序确定优先级。可灵活修改优先级。

b. 硬件判优

并行优先排队逻辑、链式优先排队逻辑、二维结构的优先排队、中断控制器判优等。

三、优先权逻辑与屏蔽技术

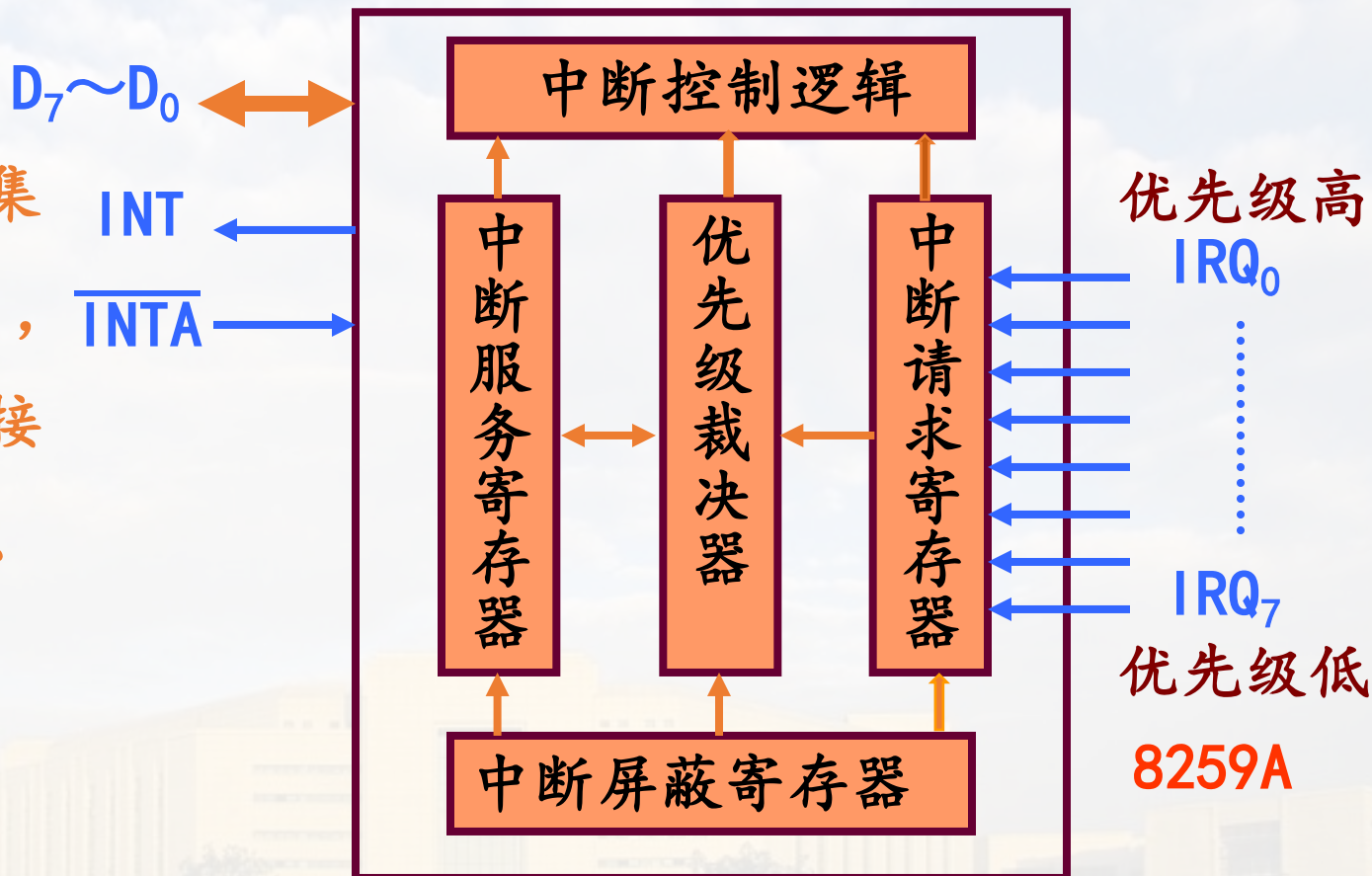
典型：1)链式优先排队逻辑电路



三、优先权逻辑与屏蔽技术

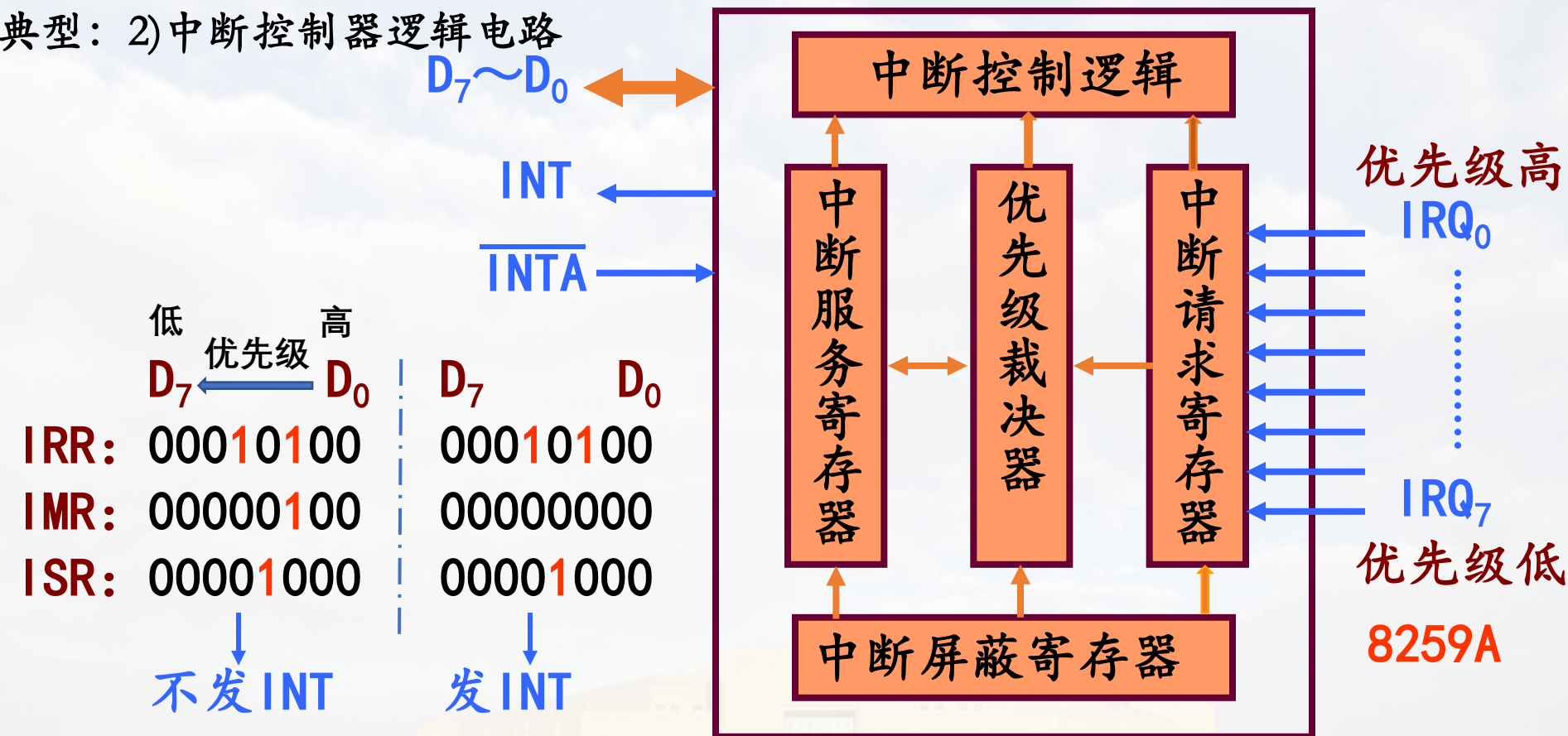
典型：2) 中断控制器逻辑电路

例：中断控制器集中判优(如8259A)，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

典型：2)中断控制器逻辑电路



中断请求 \rightarrow 8259A (未屏蔽的请求判优, 生成相应请求信号)

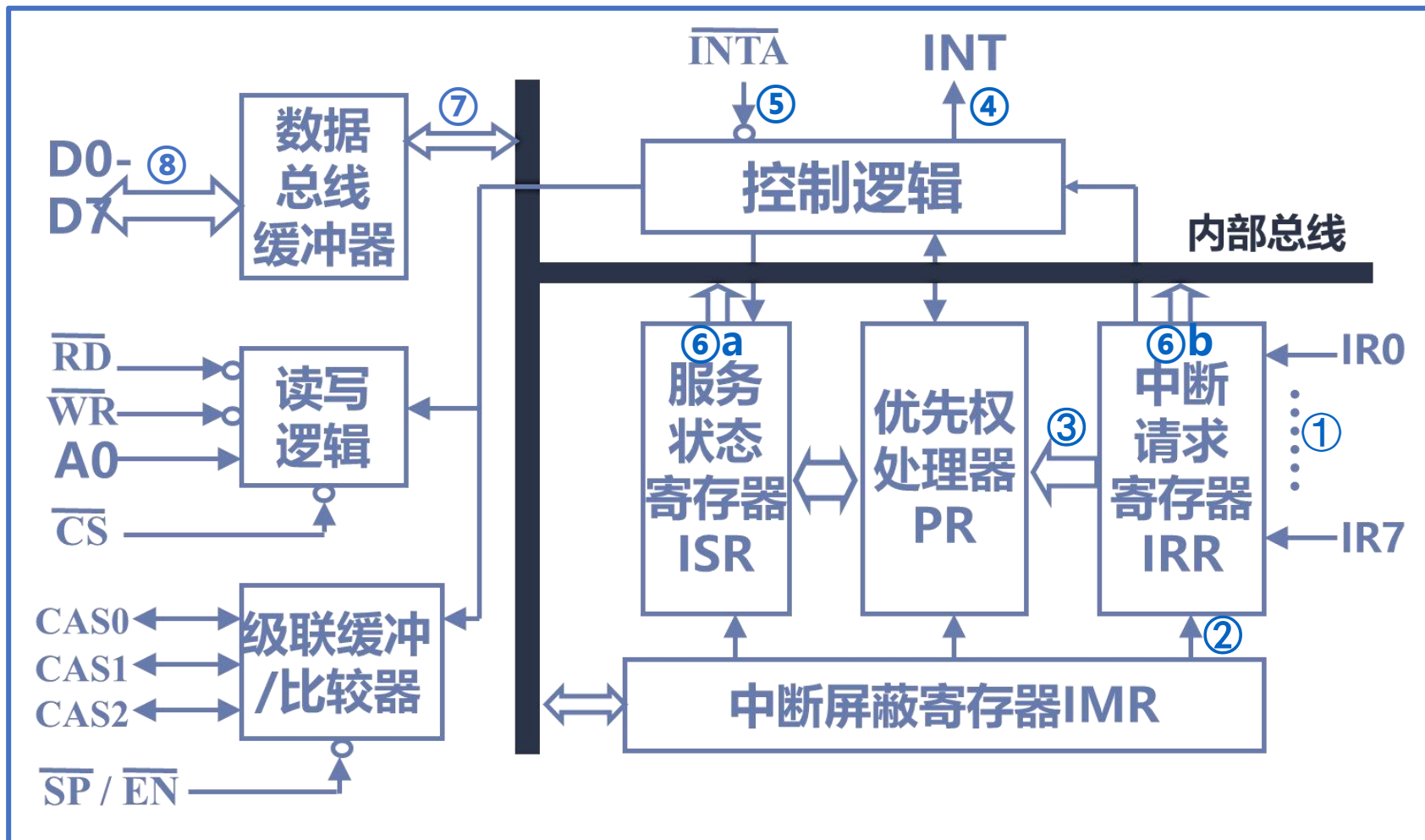
\rightarrow 公共请求 $INT \rightarrow$ CPU \rightarrow 响应信号 $\overline{INTA} \rightarrow$ 8259A

(CPU响应后, 取回中断类型码, 转入相应服务程序。)

三、优先权逻辑与屏蔽技术

8259A的结构(详细)

8259A内部结构-内部逻辑框图



4. 中断判优顺序

1) 优先顺序:故障中断、DMA、外中断(输入>输出)

2) CPU现执行程序与外设请求的判优

a. CPU设置允许中断标志 $\begin{cases} =1, & \text{开中断} \\ =0, & \text{关中断} \end{cases}$

b. CPU设置程序状态字的优先级字段

为现执行程序赋予优先级 $\begin{cases} < \text{外设请求优先级, 响应} \\ \geq \text{外设请求优先级, 不响应} \end{cases}$

四、服务程序入口地址的获取方式

获取方式 { 向量中断
非向量中断

1、向量中断

1) 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。

四、服务程序入口地址的获取方式

b. 中断向量表：即用来存放中断向量的一张表。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）组织成一张一维表格，并存放于主存的一段连续的存储区，此表就是**中断向量表**。

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。这些工作在中断周期IT中由**硬件直接实现（不需编写程序实现）**。

四、服务程序入口地址的获取方式

3) 中断向量表的组成

在模型机中，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。

中断向量 表的组成	专用区	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
专用区	0型	0000---0003 (H)	入口地址0
	1型	0004---0007 (H)	入口地址1
	。 。 。 。 。 。 。 。 。 。 。 。 。 。 。 。 。 。 。 。		
	4型	0010---0013 (H)	入口地址4
系统保留区	5型	0014---0017 (H)	入口地址5
	。 。 。 。 。 。 。 。 。 。 。 。 。 。 。 。 。 。 。 。		
	31型	007C---007F (H)	入口地址31
用户扩展区	32型	0080---0083 (H)	入口地址32
	。 。 。 。 。 。 。 。 。 。 。 。 。 。 。 。 。 。 。 。		
	255型	03FC---03FF (H)	入口地址255

四、服务程序入口地址的获取方式

4) 如何从中断向量表中获取中断服务程序入口地址

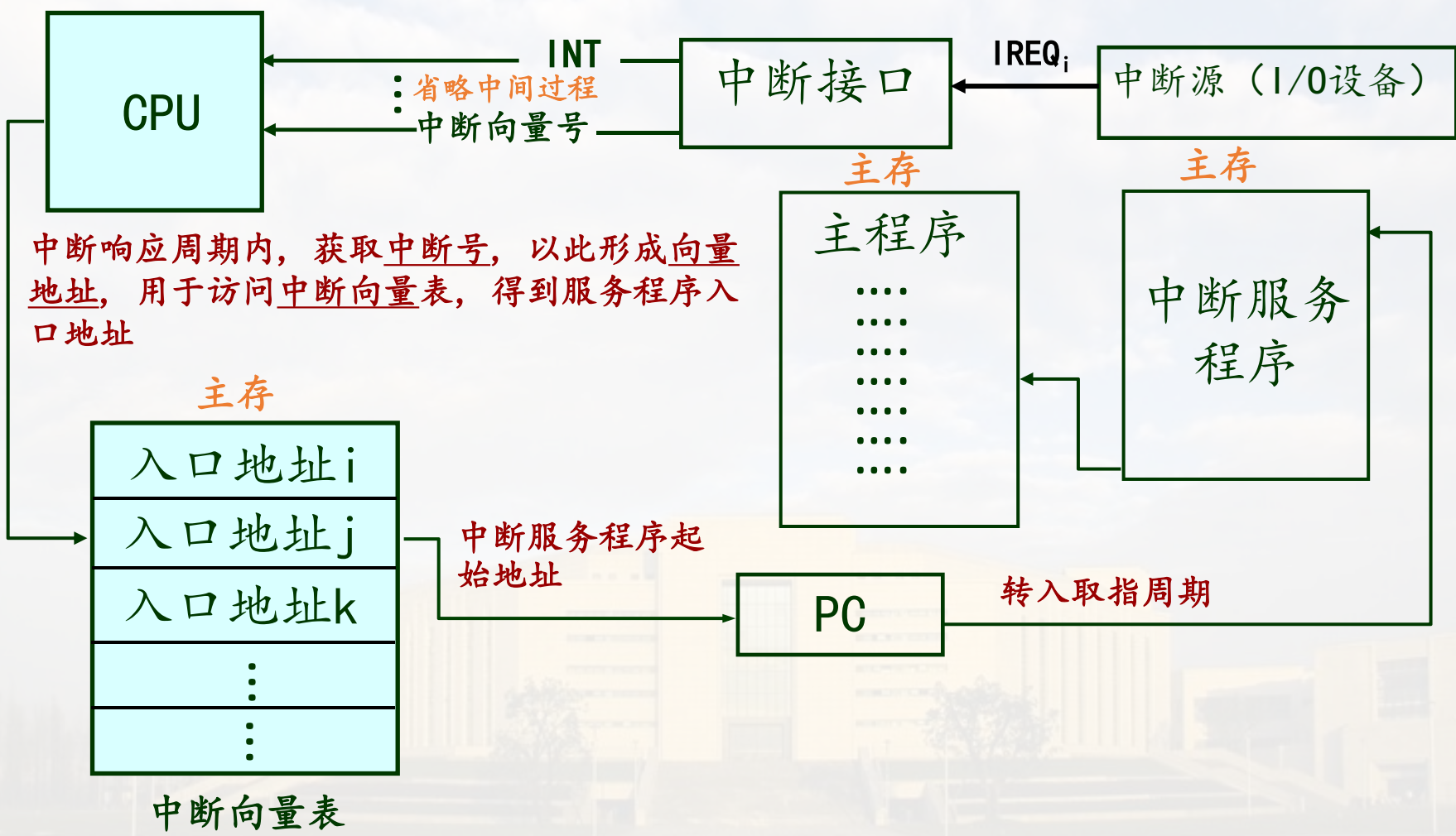
关键是如何形成向量地址 $\xrightarrow{\text{访问中断向量表}}$ 中断服务程序入口地址

向量地址
的形成

- 硬中断：向量地址 = 中断类型码 \times 4（模型机中每个中断源所占字节数）
- 软中断：向量地址 = 中断号 \times 4（模型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

中断服务程序、中断向量表之间的关系：



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)

向量表

M按字节编址。一个入口地址32位，占4个编址单元。

向量地址 = 中断号 × 4

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

四、服务程序入口地址的获取方式

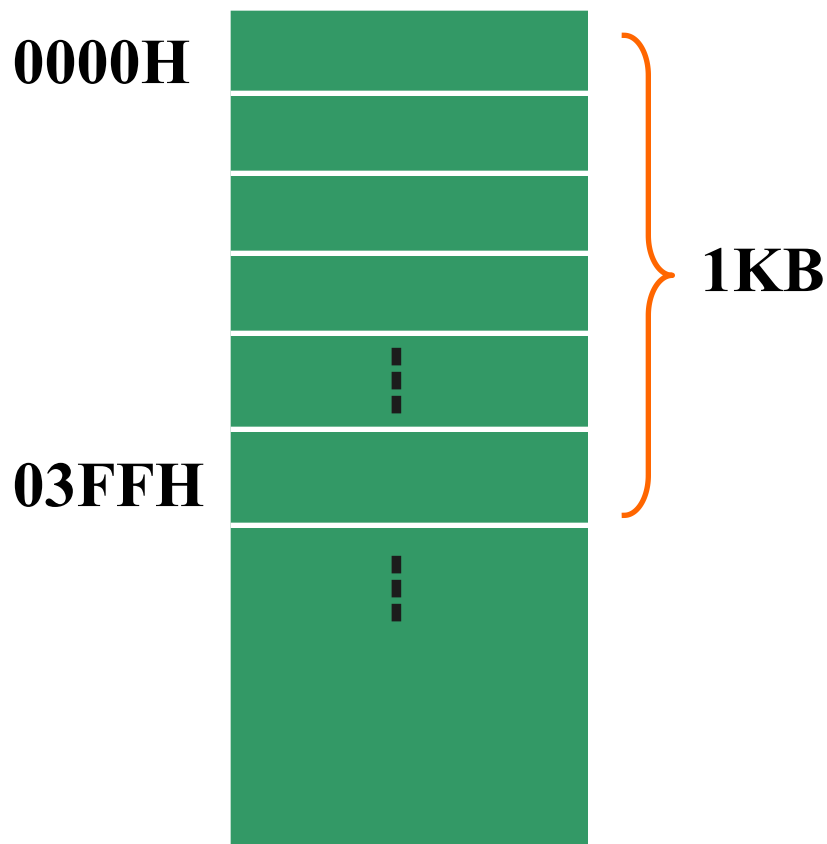
例3：中断类型码108型，求其中断服务程序入口地址。
(模型机)

向量地址 = $108 \times 4 = 432 \rightarrow 110110000B$ 即 $01B0H \rightarrow$ 得出中断服务程序入口的起始地址

四、服务程序入口地址的获取方式

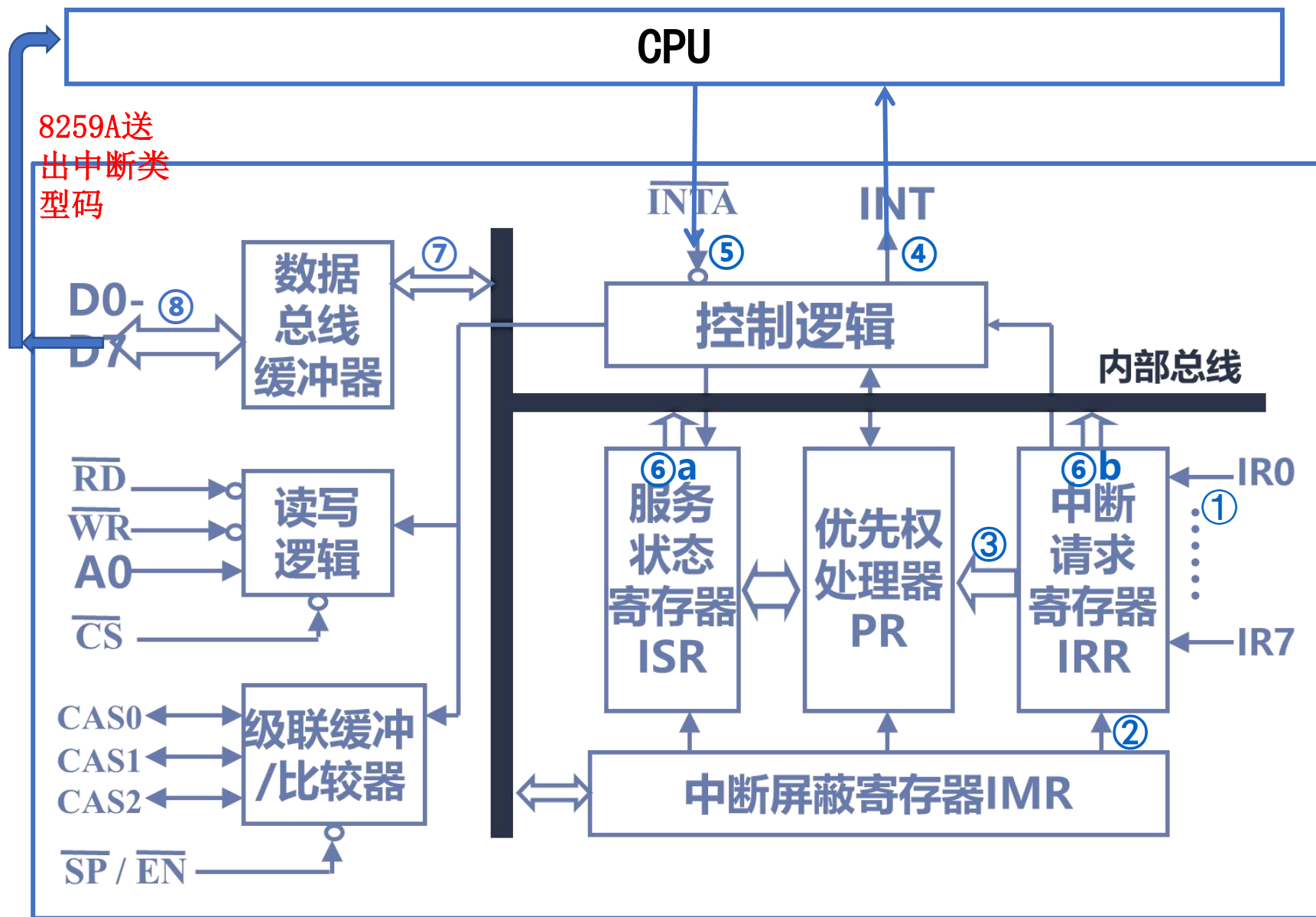
中断向量表 IVT

IBM PC机内中断向量表位于内存最低1KB



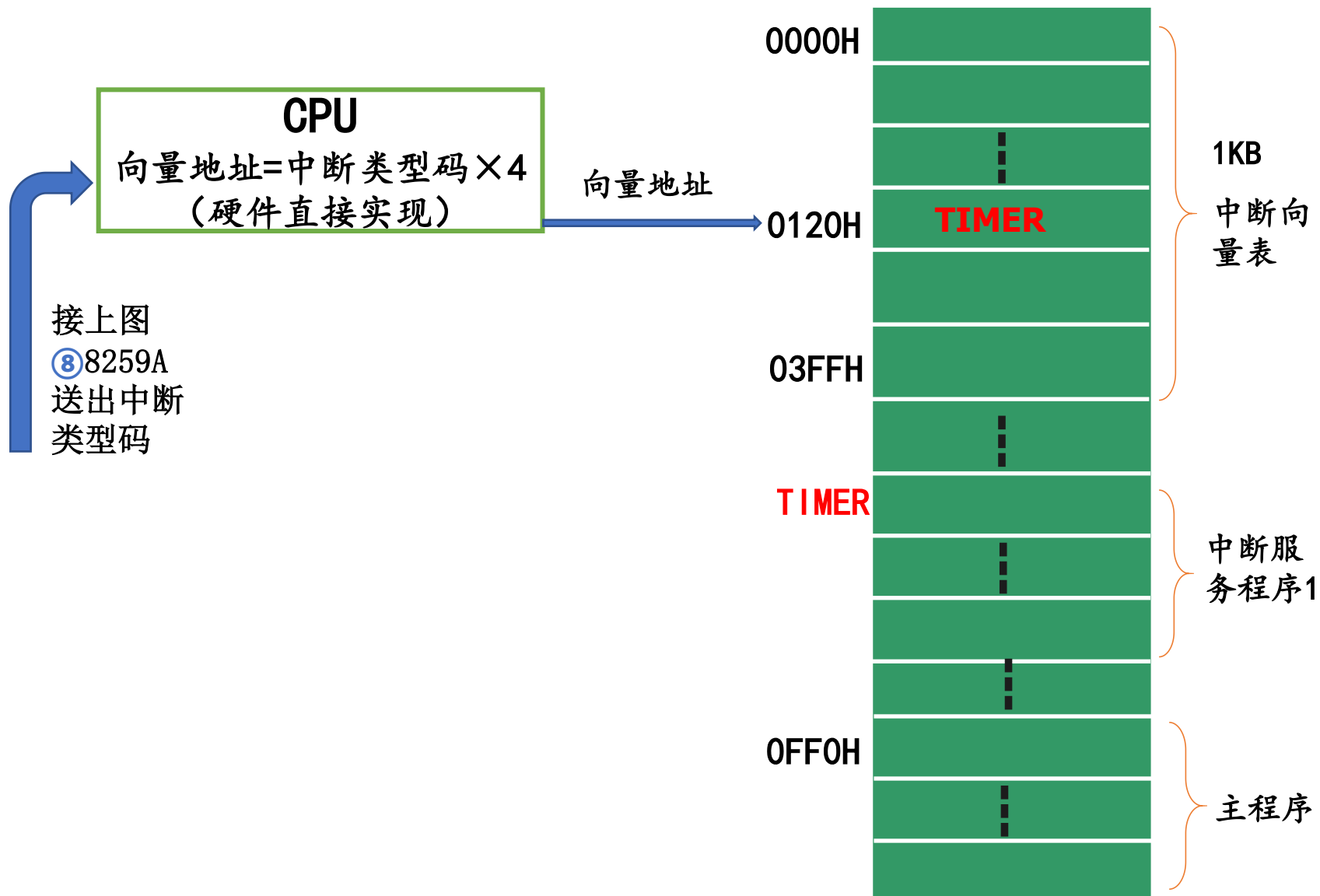
内存空间		
00000H	IP偏移地址 CS段基地址	0型中断向量
00004H	IP偏移地址 CS段基地址	1型中断向量
00008H	IP偏移地址 CS段基地址	2型中断向量
0000CH	IP偏移地址 CS段基地址	3型中断向量
00010H	IP偏移地址 CS段基地址	4型中断向量
00014H	⋮	⋮
00080H	IP偏移地址 CS段基地址	类型码为32 的中断向量
	⋮	⋮
003FFH	IP偏移地址 CS段基地址	类型码为255 的中断向量

三、优先权逻辑与屏蔽技术



四、服务程序入口地址的获取方式

中断向量表 IVT



» 中断向量表的初始化

将类型码为48H的中断服务子程序TIMER的中断向量用**MOV指令**放入向量表

- `MOV AX, 0000H`
- `MOV DS, AX` ; 初始化中断向量表
- `MOV SI, 0120H` ; $48H * 4$
- `MOV BX, OFFSET TIMER` ; 取段内偏移地址
- `MOV [SI], BX`
- `MOV BX, SEG TIMER` ; 取段首址
- `MOV [SI+2], BX`

四、服务程序入口地址的获取方式

2、非向量中断

CPU响应中断时只产生一个**固定的地址**，由此读取中断查询程序的**入口地址**，从而转向查询程序，通过软件查询，确定被优先批准的中断源，然后分支进入相应的中断服务程序。

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 $TIEN=1$ （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，或是其优先权高于程序中中断的DMA请求等）。
- 5) CPU刚刚执行的指令**不是**停机指令（HLT指令）。
- 6) 在一条指令**完整**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，不
需编写程序实现)

发响应信号INTA，进入中断周期

关中断，保存断点

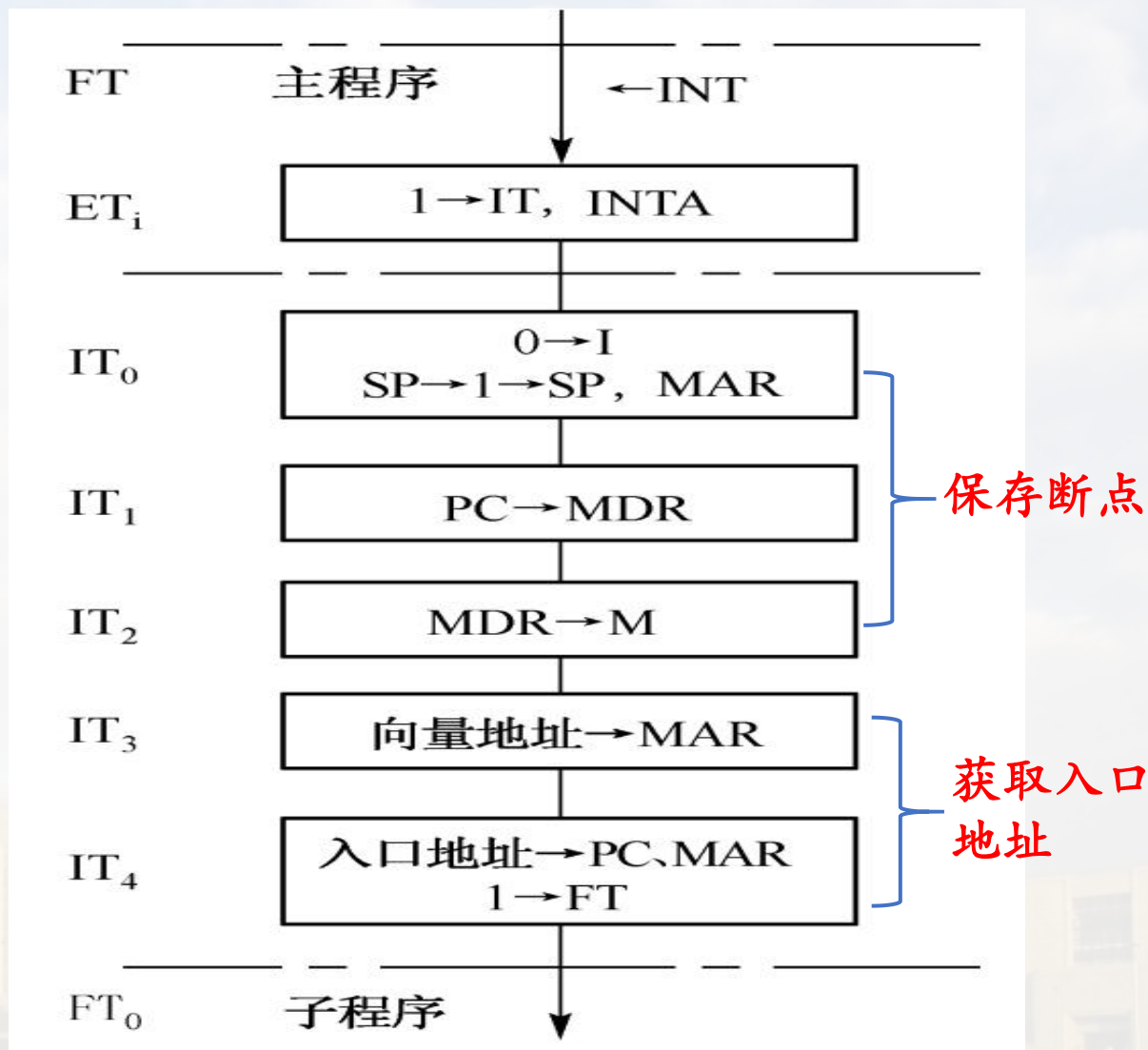
获得中断号，转换为向量地址，查向量表

取中断向量，转中断服务程序

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：
(指令流程表示)



1、中断处理：CPU执行中断服务程序。

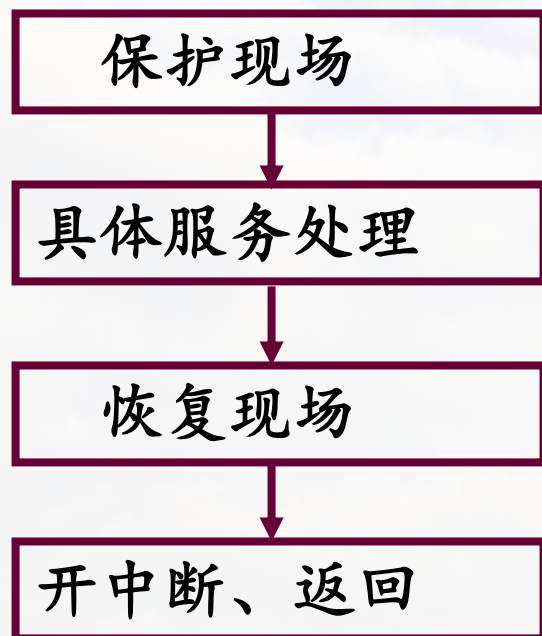
1) 单级中断

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。

2) 多重中断：

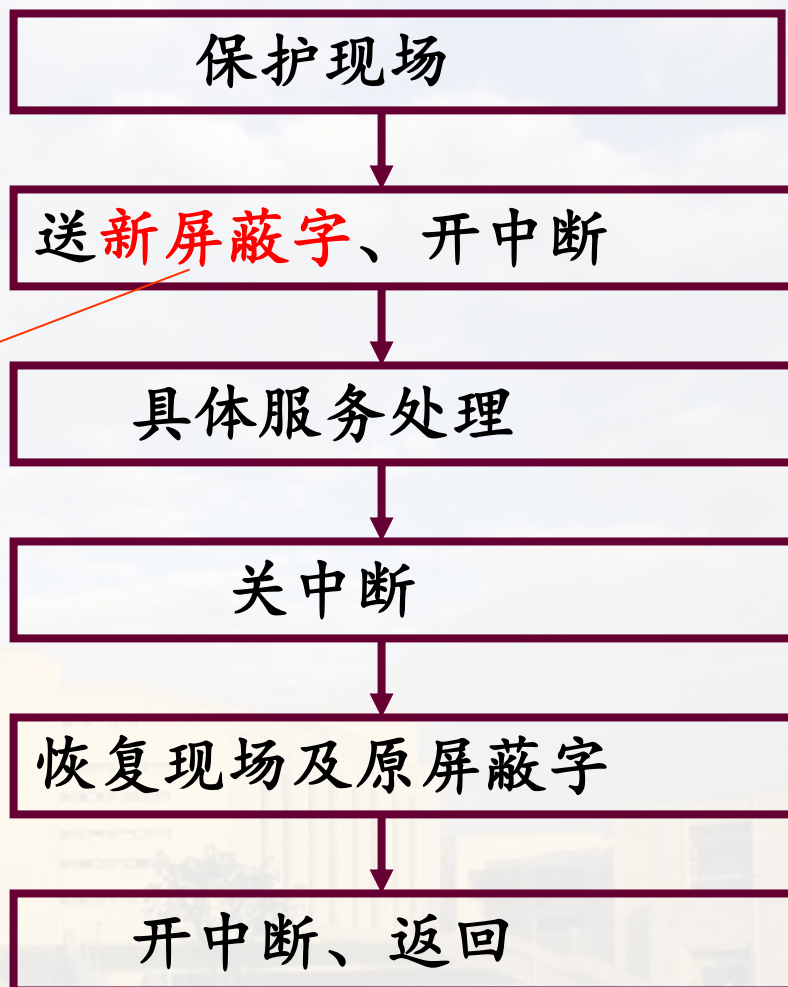
在某次中断服务过程中，允许响应处理更高级别的中断请求。

单级中断流程：



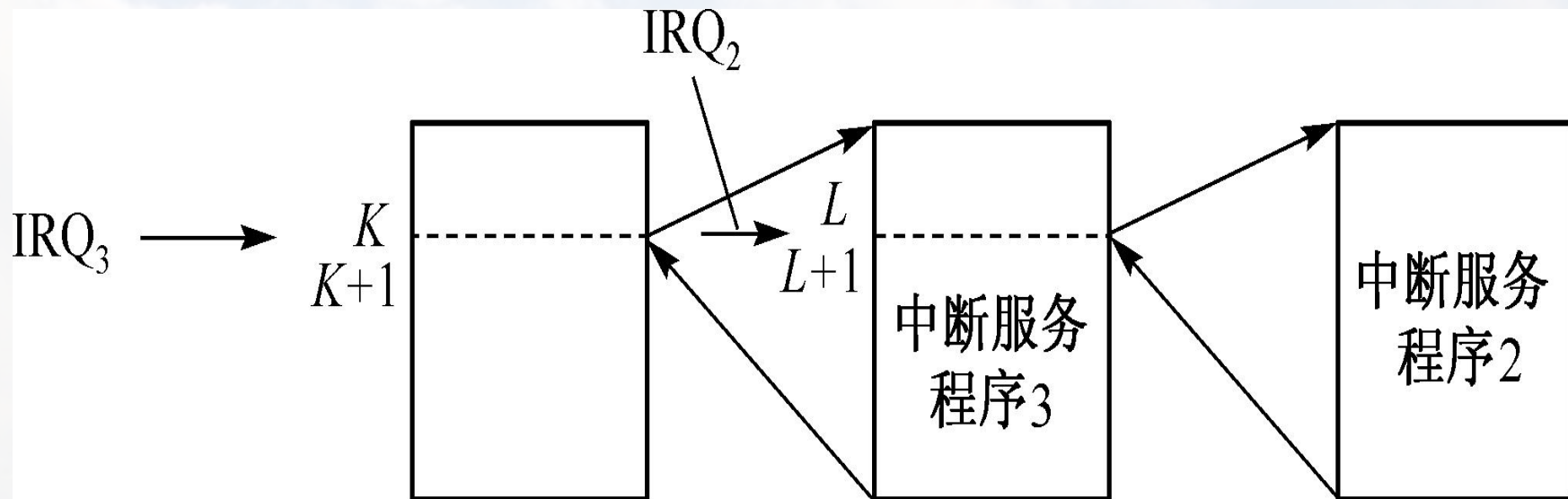
外中断：
数据传送

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

2、允许多重中断的处理方式

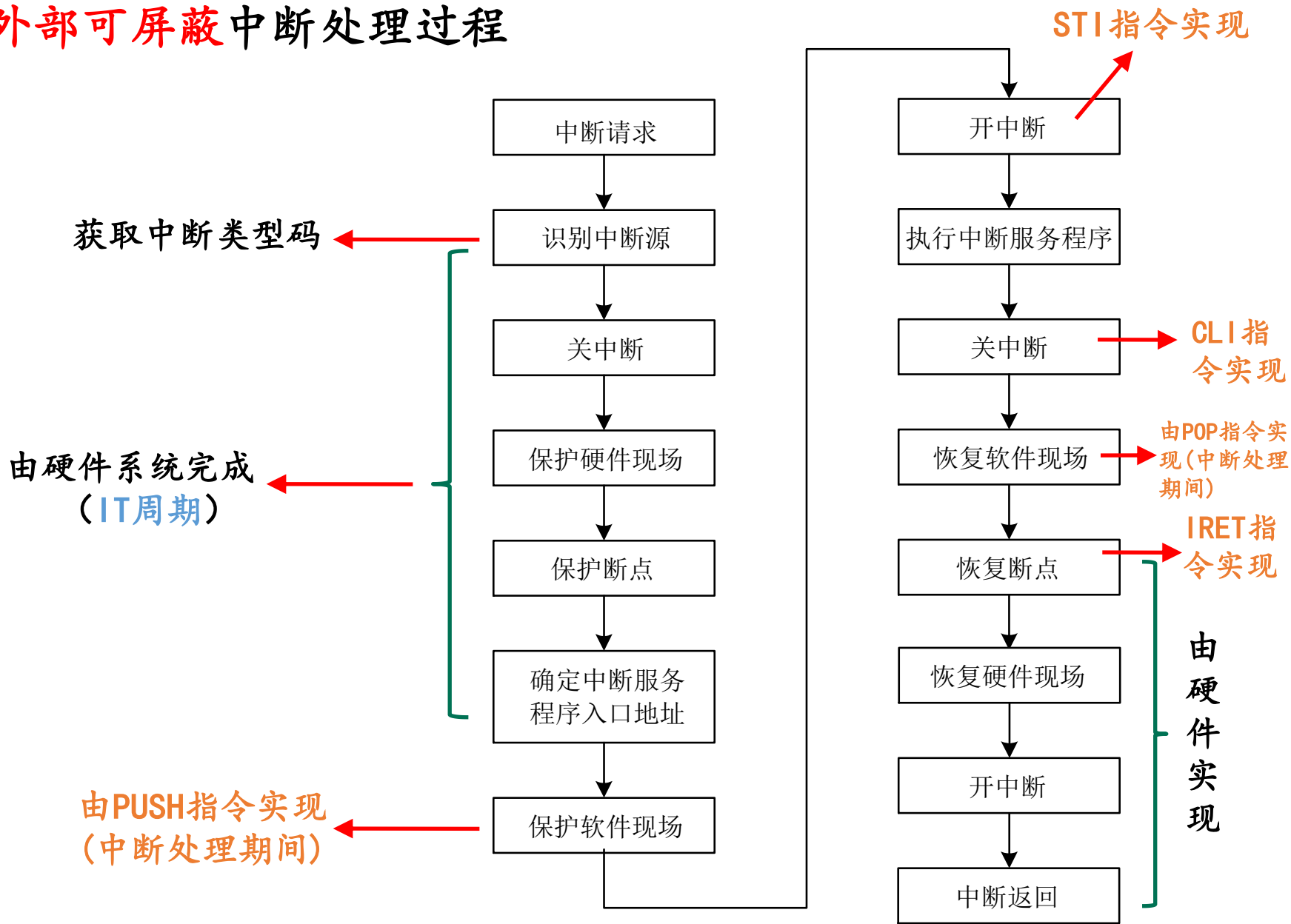


3、恢复现场与返回原程序

在编制中断服务程序时应遵循一个原则：在响应过程、保护现场、恢复现场等过渡状态中，应当关中断，使之不受干扰。

六、中断处理过程总结

外部可屏蔽中断处理过程



七、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。

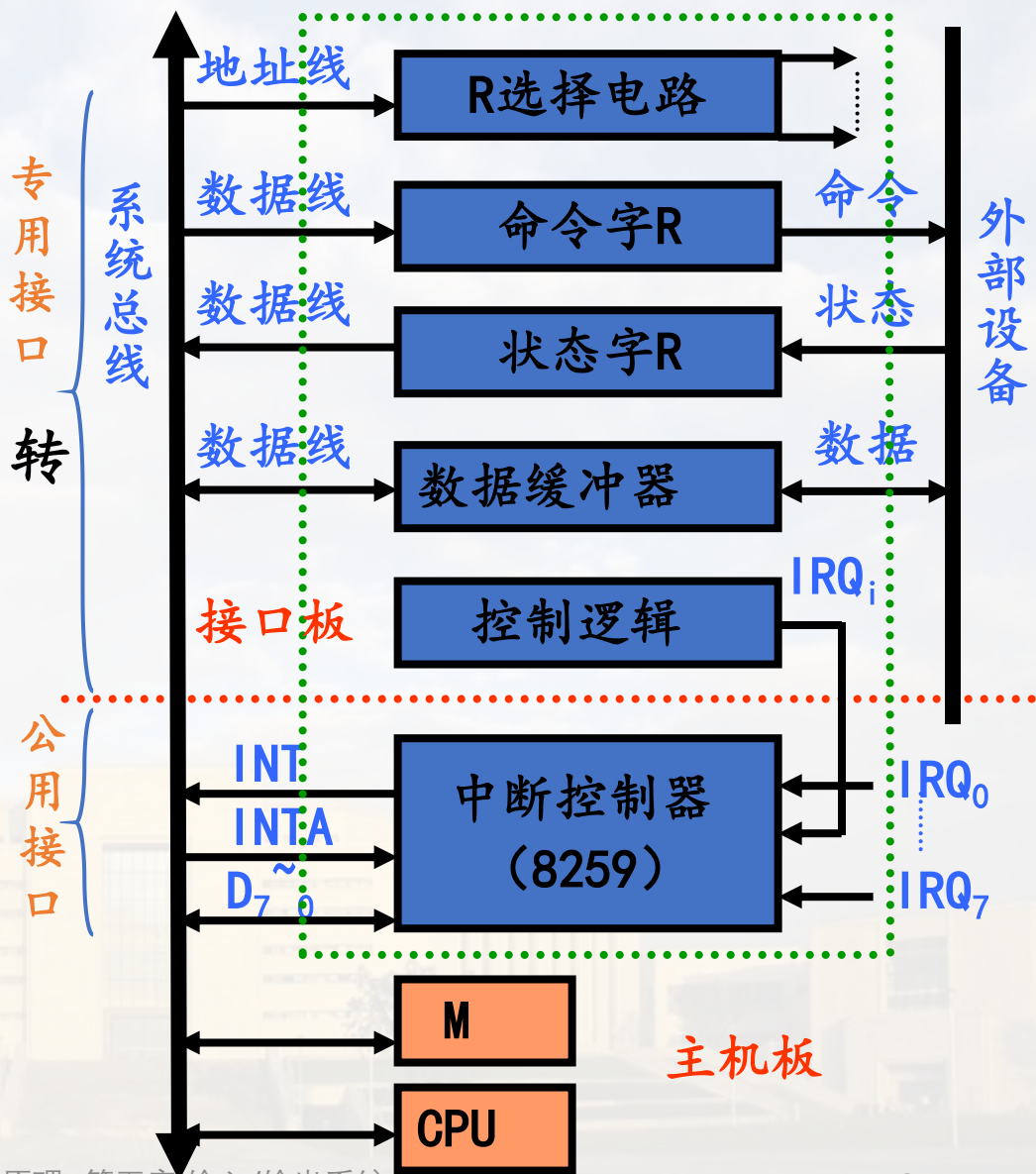
命令字格式的拟定：

用代码表示各种命令

代码位数
代码含义

3) 状态字寄存器

反映设备和接口的运行状态。



主机板

七、中断接口

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。

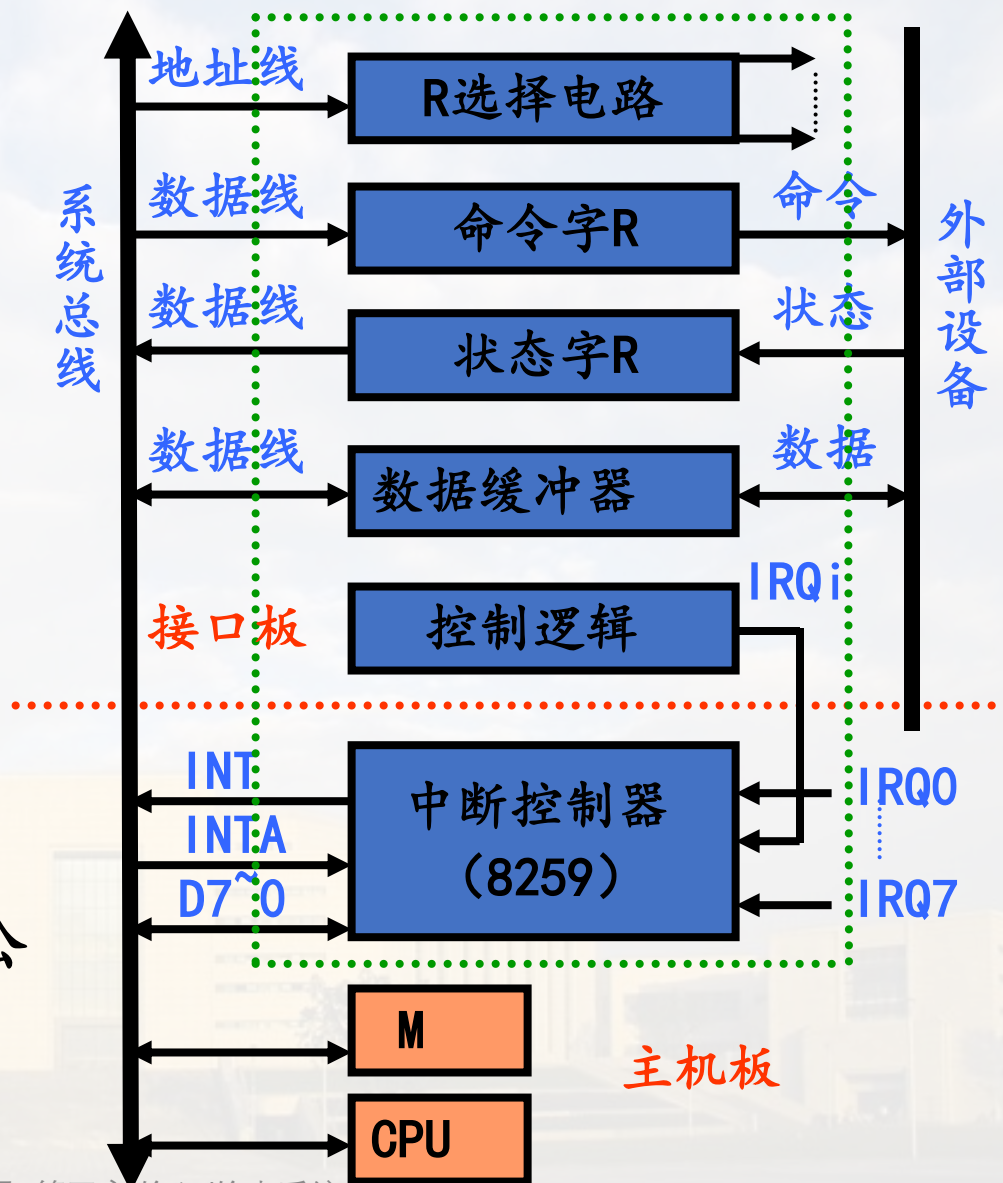
5) 控制逻辑

- 请求信号产生逻辑
- 电平转换逻辑
- 串-并转换逻辑(串口)
- 针对设备特性的逻辑

6) 公用中断控制器

接收外设请求，判优，送出公共请求；

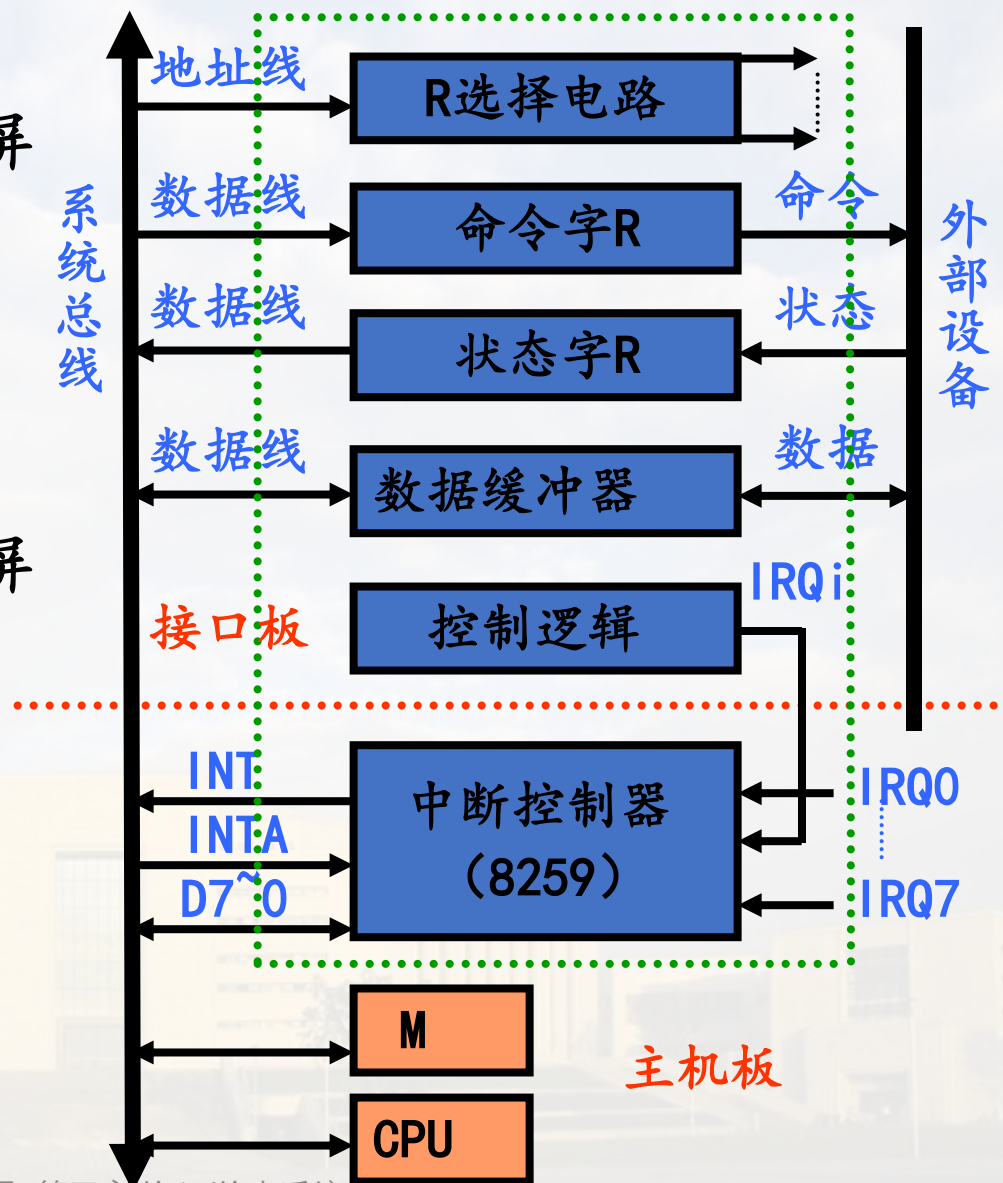
接收中断批准，送出中断号
(中断类型码)。



七、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。
- 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。



七、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。

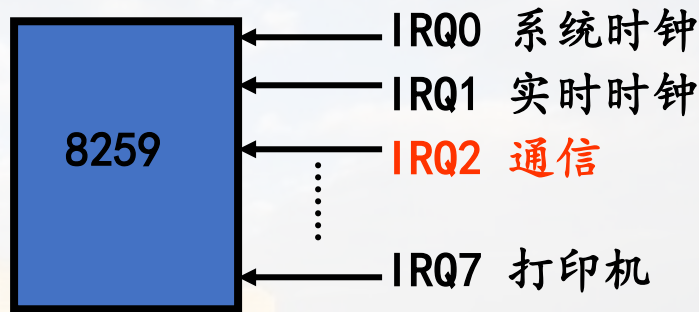
主机发向外设的命令包括：启动、停止、数据选通；

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：

5	4	3	2	1	0
启动1	停止1	选通1	启动2	停止2	选通2

状态字格式：

忙1	完成1	出错1	忙2	完成2	出错2
----	-----	-----	----	-----	-----

七、中断接口

2) 判断中断源

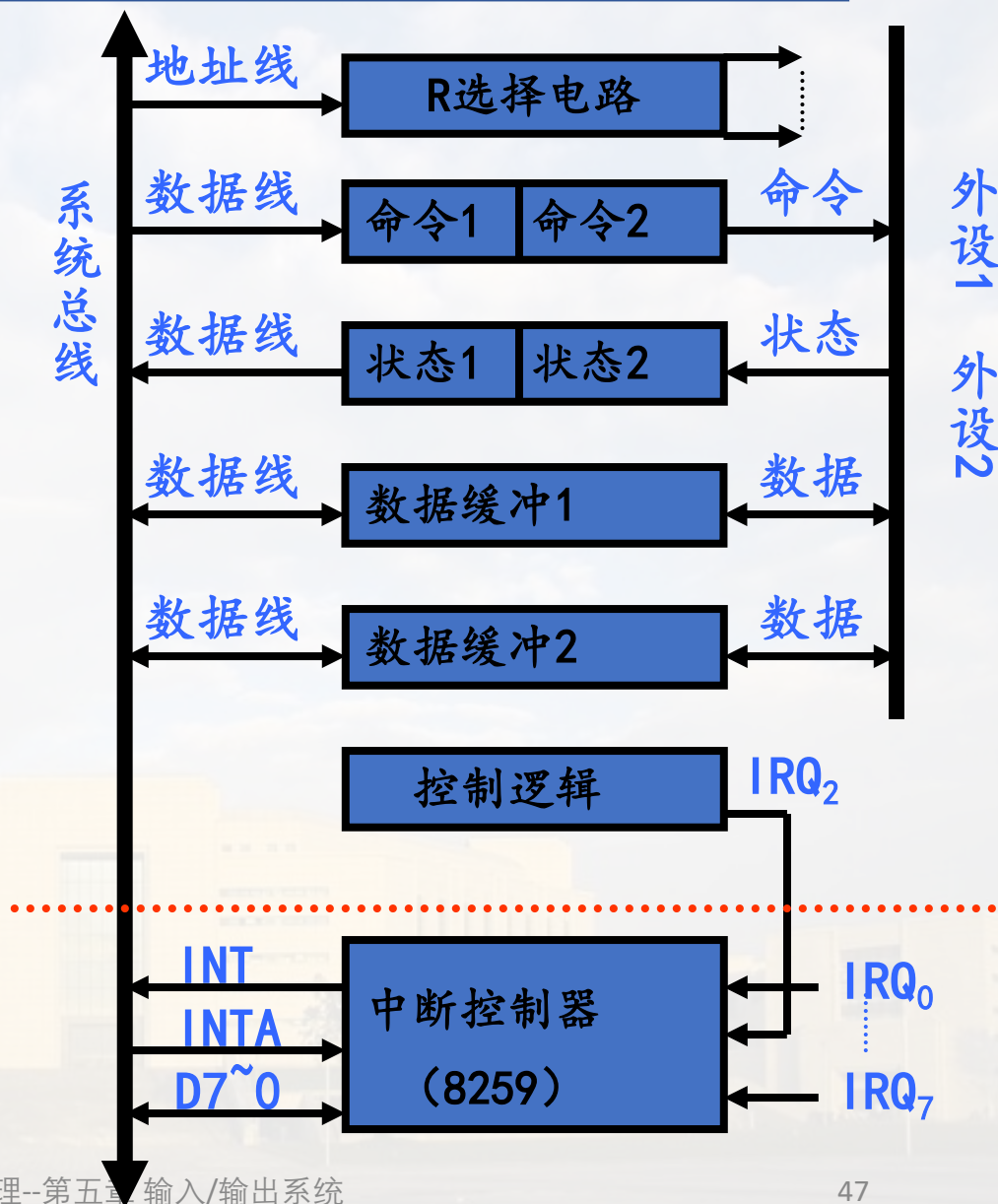
向量中断与非向量中断相结合 (软件扩展)

请求1与请求2在控制逻辑中形成公共请求 IRQ_2 ，送入8259参加判优；

CPU响应后执行 IRQ_2 服务程序 (向量中断过程)；

CPU在 IRQ_2 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)。





谢谢观看

计算机组成原理

2025-8-22



信息与软件工程学院
School of Information and Software Engineering