

第11章 中断技术



数据传送方式

- 计算机主机有多种与外设传送数据的方式
 - 通过处理器执行I/O指令完成
 - ◆ 无条件传送
 - ◆ 查询传送
 - ◆ 中断传送
 - 以硬件为主，加快传输速度
 - ◆ 直接存储器存取 (DMA)
 - ◆ 使用专门的I/O处理机

1 无条件传送

- 外设总是处于“就绪”状态，随时可以进行数据传送
- 无条件传送的接口电路只考虑数据缓冲
- 无条件传送的软件编程十分简单
- 处理器与慢速设备交换数据时会发生数据丢失

输入时执行输入IN指令

```
mov dx, 8000h  
in al, dx  
mov bufin, al
```

输出时执行输出OUT指令

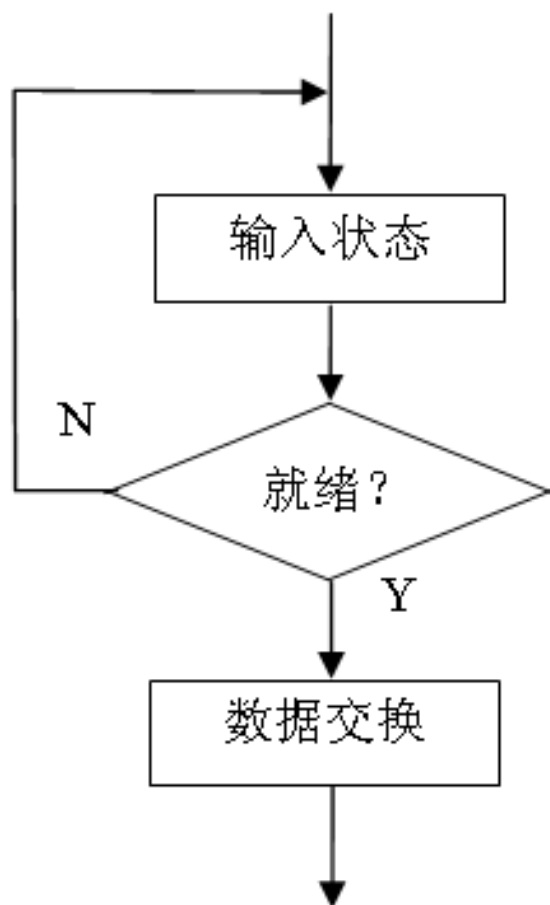
```
mov al,bufout  
mov dx,8000h  
out dx,al
```

例：无条件传送接口电路

- 输入接口电路连接**开关**：**读取开关状态**
- 输出接口电路连接**发光二极管LED**
 - **功能要求：开关闭合时，将相应LED点亮**
- 调用延时子程序DELAY保持一定时间

mov dx, 8000h	;DX指向输入端口
in al, dx	;从输入端口读开关状态
not al	;求反
mov dx, 8001h	;DX指向输出端口
out dx, al	;送输出端口显示
call delay	;调子程序DELAY进行延时

2 程序查询传送



- **查询传送有查询和传送两个环节**
 - **首先查询外设工作状态**
 - **检测、等待外设准备就绪**
 - **进行数据传输**

1. 查询过程

- 设计实现查询功能的电路
 - 连接外设的状态输入信号
 - 保存在状态寄存器中
 - 通过状态端口读取
- 外设的工作状态在状态寄存器中使用一位或若干位表达，查询通过输入指令来实现
- 有多个状态，按照一定原则轮流查询，先检测到就绪的外设先开始数据传送
- 实际中常引入超时判断

- 读取**状态端口**查询外设状态，若已就绪，读取**数据端口**得到外设提供的数据

mov dx,5001h ;DX指向状态端口

status: in al,dx ;读状态端口

test al,01h ;测试状态位D0

jz status ;D0 = 0, 未就绪, 继续查询

dec dx ;D0 = 1, 就绪, DX改指数据端口

in al,dx ;从数据端口输入数据

本章教学内容

1

中断技术的基本概念

2

中断技术的工作原理

3

系统配置的中断资源的应用

4

中断向量修改

5

中断服务程序的编写

-
- 11.1 中断的概念**
 - 11.2 中断类型**
 - 11.3 中断号**
 - 11.4 中断触发方式与排队方式**
 - 11.5 中断向量与中断向量表**
 - 11.6 中断处理过程**
 - 11.7 中断控制器**
 - 11.8 可屏蔽中断系统**
 - 11.9 用户对系统中断资源的使用**
 - 11.10 中断服务程序设计**

11.1 中 断

基本概念:

中断是指CPU在正常运行程序时，由于内部/外部事件或由程序预先安排的事件，而转到为中断事件服务的程序中去，服务完毕再返回执行原程序，这一过程称为中断。

硬件中断： 由外部硬件产生的中断。

软件中断： 由程序预先安排的中断。

中断的特征： 具有随机性。



一个完整的中断过程

(1) **中断请求** ——是指中断源（引起中断的事件或设备）向CPU发出的**请求中断的要求**。

(2) **中断判优** ——当有多个中断源发出中断请求时，需要通过适当的办法（软件的；硬件的；软、硬件结合的）决定**究竟先处理哪个中断请求**。

(3) **中断响应** ——是指CPU中止现执行程序**转至中断服务程序的过程**。

(4) **中断处理** ——是指CPU**执行中断服务程序**，完成中断请求所要求的操作。

(5) **中断返回** ——是指执行完中断服务程序后，**返回**到原先被中断的程序。

中断系统（包括82C59a中断控制器芯片和OS以及BIOS的中断服务）应具有的功能：

- (1) 中断接收功能**
- (2) 中断响应功能**
- (3) 中断屏蔽功能**
- (4) 中断排队功能**
- (5) 中断嵌套功能**
- (6) 中断返回功能**
- (7) 中断共享功能**

11.2 中断类型

分为两大类：**硬中断（外部中断）**和**软中断（内部中断）**

11.2.1 硬中断

- **可屏蔽中断INTR**

通过中断控制器用中断请求线INTR向微处理器申请而产生的中断，它可以用指令屏蔽与开放，CLI。

- **不可屏蔽中断NMI**

通过另一根中断请求线NMI向微处理器申请而产生的中断，它不可以用指令屏蔽与开放，且**不受中断允许标志位(IF)的影响**，所以是不可屏蔽中断。

11.2.2 软中断

- 发出中断指令**INT nH**产生

由软件中断指令引起的中断，例INT n 等。无中断响应周期，中断号由指令中给出；该中断无法屏蔽。

- ✓ DOS功能调用
- ✓ BIOS功能调用

- 内部突发事件引起

指令执行发生错误而引起的中断，这种情况下必须启动一个处理程序以解决问题，并恢复这条指令或下条指令的执行，如除以0的问题，32位CPU的取指令异常问题等。

11.3 中断号

11.3.1 中断号与中断号的获取

1、中断号

系统分配给每个中断源的代号，以便识别与处理。16位微机支持256个中断号，统一编号为0-255号。向量中断中，CPU通过中断号 $n \times 4$ 找到中断向量表中存放中断服务程序入口地址的位置，并转移到中断服务程序转移。

2、中断号的获取

- **可屏蔽中断**：在中断响应周期从中断控制器获取
- **软中断**：由中断指令直接给出
- **不可屏蔽中断**：预先设置

11.3.2 中断号的分配

系统对外部中断和内部中断一律统一编号，共有256个号，微机系统的中断号分配如后表所示。表中**两块灰色区域**的中断号分别是系统**分配给**PC微机系统中的**中断控制器主片与从片的中断号**，**用户可以采用修改其中断向量的方法进行应用。**

中断号	名称	中断号	名称
0	除零数	25H	磁盘扇区读
1	单步	26H	磁盘扇区写
2	NMI	27H	程序终止驻留
3	断点	28H	等待状态处理
4	溢出	29H	字符输出处理
5	屏幕打印	2AH	保留
6	保留	2BH	保留
7	保留	2CH	保留
8	日时钟中断	2DH	
9	键盘中断	2EH	命令执行处理
0AH	从片中断	2FH	多路复用处理
0BH	串行口 2 中断	30H	内部使用
0CH	串行口 1 中断	31H	内部使用
0DH	并行口 2 中断	32H	保留
0EH	软盘中断		
0FH	打印机/并行口 1 中断		

中断号	名称	中断号	名称
10H	视频显示 I/O	67H	用户保留
11H	设备配置检测	68H	保留
12H	内存容量检测		
13H	磁盘 I/O	6FH	保留
14H	串行通信 I/O	70H	时钟中断
15H	盒带/多功能实用	71H	改向 INTOAH
16H	键盘 I/O	72H	保留
17H	打印机 I/O	73H	保留
18H	ROM BASIC	74H	保留
19H	磁盘自举	75H	协处理器中断
1AH	日时钟/实时钟 I/O	76H	硬盘中断
1BH	Ctrl-Break 中断	77H	保留
1CH	定时器报时	78H	未使用区
1DH	视频显示方式参数		
1EH	软盘基数表	7FH	
1FH	图形显示扩展字符	80H	BASIC 使用区
20H	程序终止退出		
21H	系统功能调用	EFH	
22H	程序结束地址	FOH	内部使用区
23H	Ctrl-C 出口地址		
24H	严重错误出口地址	FFH	

11.4 中断触发方式与中断排队方式

1、中断触发方式

中断触发方式是指外部设备以什么逻辑信号向中断控制器申请中断，中断控制器**允许用边沿或电平信号申请中断**，即**边沿触发**和**电平触发**两种方式。**触发方式在中断控制器初始化时设定。**

2、中断的排队方式

- PC微机中断优先级的顺序：

软件中断>不可屏蔽中断> 可屏蔽中断

- 中断排队的方式：

- ① 按优先级排队 **(使用最多)**

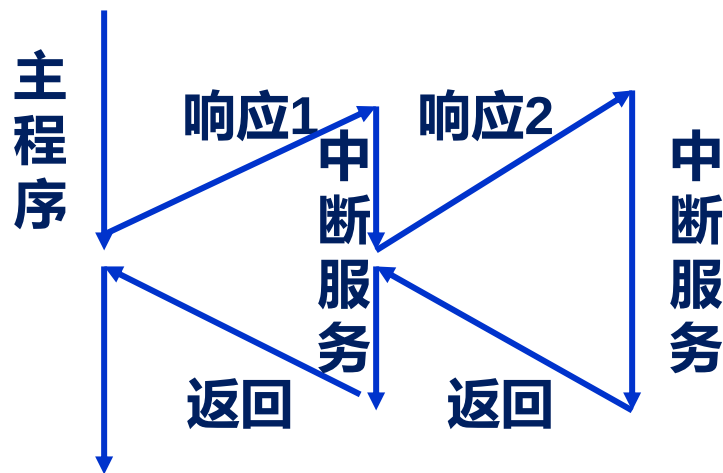
根据任务的轻重缓急，给每个中断源指定CPU响应的优先级，任务紧急的先响应，可以暂缓的后响应。

- ② 循环轮流排队

不分级别高低，CPU**轮流响应**各个中断源的中断请求，并为其服务。

3、中断嵌套

当CPU**正处理某个中断**时，出现**更高级**中断源的申请，CPU必须**暂停**该中断服务，转而去处理更高级中断的服务，待处理完后，再返回到被中断的服务程序继续执行。



11.5 中断向量与中断向量表

当发生中断，就意味着要发生程序的转移，即由主程序转移到服务程序去。那么，如何才能进入中断服务程序，即如何找到中断服务程序的入口地址是解决问题的关键。

为此，采用向量中断方式（不是查询中断方式），设置中断向量及中断向量表，通过中断向量表中的中断向量查找中断服务程序的入口地址。

1、中断向量

- CPU响应中断后，中断源提供中断号，再由中断号生成地址信息，由此地址信息对程序的执行进行导向，引导到中断服务程序中去，故把这个地址信息称为中断向量（存储单元）。
- 中断服务程序是预先设计好并存放在程序存储区的，因此，中断服务程序的入口地址由服务程序的段基址CS和偏移地址IP两部分共4个字节组成。中断向量包括中断服务程序的段基址CS，偏移地址IP共4个字节。
 - * 中断发生时，从中断向量表中取出相应的值赋值给CS和IP的过程是由系统硬件自动完成的，程序员不可见！程序员能够做的就是修改中断向量表中存储的值。

中断向量指针与中断类型号

什么叫“中断向量指针”？

指出中断向量存放在中断向量表的位置(或地址)。

中断类型号 $\times 4$ =中断向量最低字节的指针

中断号 $\times 4$ =偏移地址IP； 中断号 $\times 4+2$ =段基址CS

例如，8号中断中断源的中断向量在存储器中的地址：

IP8=8 $\times 4$ =20H~21H；

CS8=8 $\times 4+2$ =22H~23H.

在PC系列中由中断类型号提供的，即
向量地址=0000： 类型号×4

如：硬盘 “1NT13H”

它的向量地址=0000： 13H×4

=0000： 004CH

004CH开始连续4个单元中用来
存放 “INT 13H”的中断向量

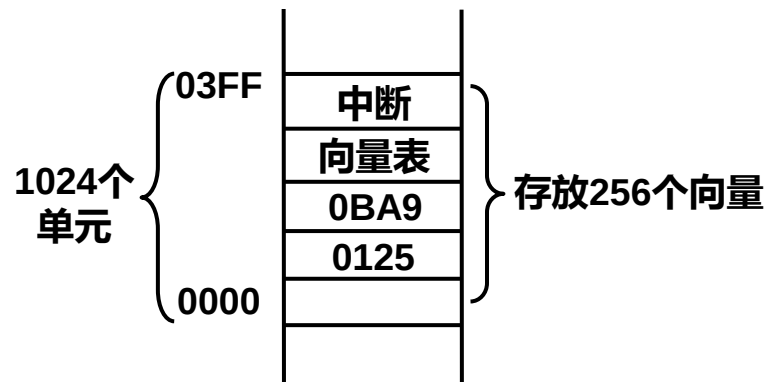
中断向量0070(CS):0FC9(IP)



2、中断向量表

- 所有的中断向量集中存放到存储器的某一区域，这一区域称之为中断向量表。
- 在实模式下，规定存储器的000H-03FFH作为中断向量存储区（共1024个字节），这表明中断向量表的起始地址是固定的，并且从存储器的物理地址0开始。
- 每个中断向量包含4个字节，这4个字节在中断向量表中的存放规律是向量的偏移量（IP）存放在两个低字节单元中，向量的基址（CS）存放在两个高字节单元中。可容纳256个中断向量，即可处理256个中断服务程序。

中断向量号 $n \times 4$ = 中断向量的地址指针



存储地址	中断向量	中断向量号
00	IP值-向量0 (IP ₀)	0号 (除法错误)
02	CS值-向量0 (CS ₀)	
04	IP ₁	1号 (调试)
06	CS ₁	
08	IP ₂	2号 (NMI)
0A	CS ₂	
0C	IP ₃	3号 (断点)
0E	CS ₃	
10	IP ₄	4号 (溢出错误)
12	CS ₄	
14	IP ₅	5号 (屏幕打印)
16	CS ₅	
18	IP ₆	6号 (保留)
1A	CS ₆	
1C	IP ₇	7号 (保留)
1E	CS ₇	
20	IP ₈	8号 (定时器中断)
22	CS ₈	
24	IP ₉	9号 (键盘中断)
26	CS ₉	
28	IP ₁₀	10号 (8259A从片中断)
2A	CS ₁₀	
2C	IP ₁₁	11号 (串口2中断)
2E	CS ₁₁	
30	IP ₁₂	12号 (串口1中断)
32	CS ₁₂	

34	IP ₁₃	13号 (并口2中断)
36	CS ₁₃	
38	IP ₁₄	14号 (软盘中断)
3A	CS ₁₄	
3C	IP ₁₅	15号 (打印机中断)
3E	CS ₁₅	
		⋮
3FC	IP ₂₅₅	255号
3FE	CS ₂₅₅	

中断向量表

3、中断向量表的填写

- 系统设置的中断服务程序，由**系统负责填写中断向量**。
系统加电后BIOS填写其提供的服务程序中中断向量；
启动DOS时填写其提供的服务程序中中断向量；
- 用户开发的中断服务系统，**用户负责填写中断向量**

例：用MOV指令填写中断类型为60H的中断向量

CLI	； 关中断
CLD	； 内存地址加 1， 设定让DI递增的方向
MOV AX, 0	； 中断服务程序的入口地址0000:DI
MOV ES, AX	； ES:DI配对使用 配合STOSW指令 可尽可能 缩短程序尺寸， 减少关中断时间
MOV DI, 4*60H	； 中断向量指针或偏移量 DI
MOV AX, OFFSET INTR60	； 中断服务程序的偏移地址 AX
STOSW	； AX → [DI][DI+1]中， 然后DI+2
MOV AX, SEG INTR60	； 中断服务程序的段基址 AX
STOSW	； AX → [DI][DI+1]
STI	； 开中断

4、中断向量的修改

DOS功能调用法【适合标准PC，不适合实验箱，原因是实验箱上DOS中断的25和35号功能未必是存/取中断向量的】，需分三步修改：

①用35H号功能，获取原中断向量,以便保存和恢复。

AL=待取中断号 ES:BX=中断向量

②用25H号功能，设置新中断向量，取代原中断向量。

DS:DX=中断向量 AL=待设中断号

③新服务程序执行完毕后，利用25H号功能恢复原中断向量。

例：修改中断号为60H的中断向量。

第①步：

```
MOV  AH, 35H          ; DOS中断的35号功能，用于取中断向量地址
MOV  AL, 60H          ; 60H为待取的中断号
INT  21H
MOV  OLD_SEG, ES      ; 保存原中断向量
MOV  OLD_OFF, BX
```

第②步:

```
MOV AH, 25H          ; 设置新中断向量
MOV AL, 60H          ; 中断号
MOV DX, SEG INTR
MOV DS, DX           ; DS指向新中断程序段地址
MOV DX, OFFSET INTR ; DX指向新中断程序偏移地址
INT 21H
```

第③步:

```
MOV AH, 25H          ; 恢复旧的中断向量
MOV AL, 60H          ; 中断号
MOV DX, OLD_SEG
MOV DS, DX           ; DS指向旧中断程序段址
MOV DX, OLD_OFF      ; DX指向旧中断程序偏移地址
INT 21H
```

11.6 中断处理过程

11.6.1 可屏蔽中断的处理过程

可屏蔽中断INTR包括4个阶段：

- (1) 中断申请与响应握手(硬件自动完成, 程序员不可见)
- (2) 标志位的处理与断点保存(硬件自动完成, 程序员不可见)
- (3) 向中断服务程序转移并执行中断服务程序
- (4) 返回断点

1. 中断申请与响应握手

当外部设备要求CPU服务时，需向CPU发出中断请求信号，**申请中断**。CPU若发现有外部中断请求，并且处在**开中断**条件（ $IF=1$ ），**又没有DMA申请**，则**CPU**在当前指令执行结束时，进入中断响应总线周期，**响应中断请求**，并且**通过中断回答信号**，**从中断控制器读取中断源的中断号**，完成中断申请与中断响应的握手过程。**这一阶段的主要目标是获取外部中断源的中断号。**

2. 标志位的处理与断点保存

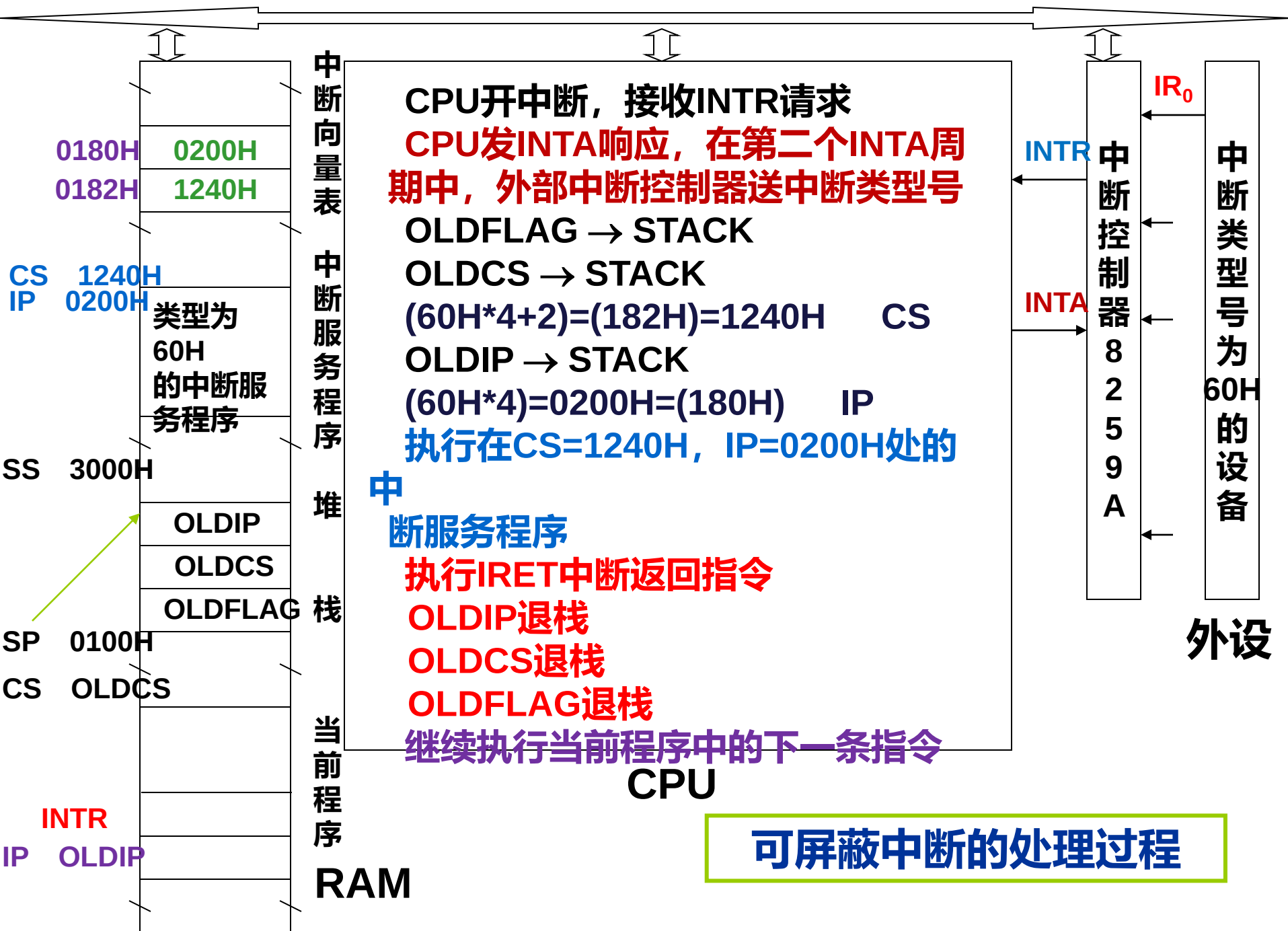
微处理器获得外部中断源的中断号后，**CPU把标志寄存器FLAGS压入堆栈**，并置 $IF=0$ ，**关闭中断**；置 $TF=0$ ，**防止单步执行**。然后将**当前程序的代码段寄存器CS和指令指针IP压入堆栈**，这样就把断点（返回地址）保存到了堆栈的栈顶。**这一阶段的主要目标是完成由主程序向服务程序转移前的准备工作。**

3. 向中断服务程序转移并执行中断服务程序

将已获得的**中断号乘以4得到地址指针**，在中断向量表中，**读取中断服务程序的入口地址CS:IP**，再把它写入**代码段和指令指示器**，实现程序控制的转移。**这一阶段的主要目标是完成主程序向中断服务程序的转移，或称为中断服务程序的加载。**

4. 返回断点

中断服务程序执行完毕后，要返回主程序，因此，一定要恢复断点和标志寄存器的内容，否则，主程序无法继续执行。为此，**在中断服务程序的末尾，执行中断返回指令IRET**，将栈顶的内容依次弹出到IP、CS和**FLAGS**，就恢复了主程序的执行。**这一阶段的主要目标是完成中断服务程序向主程序的转移。**

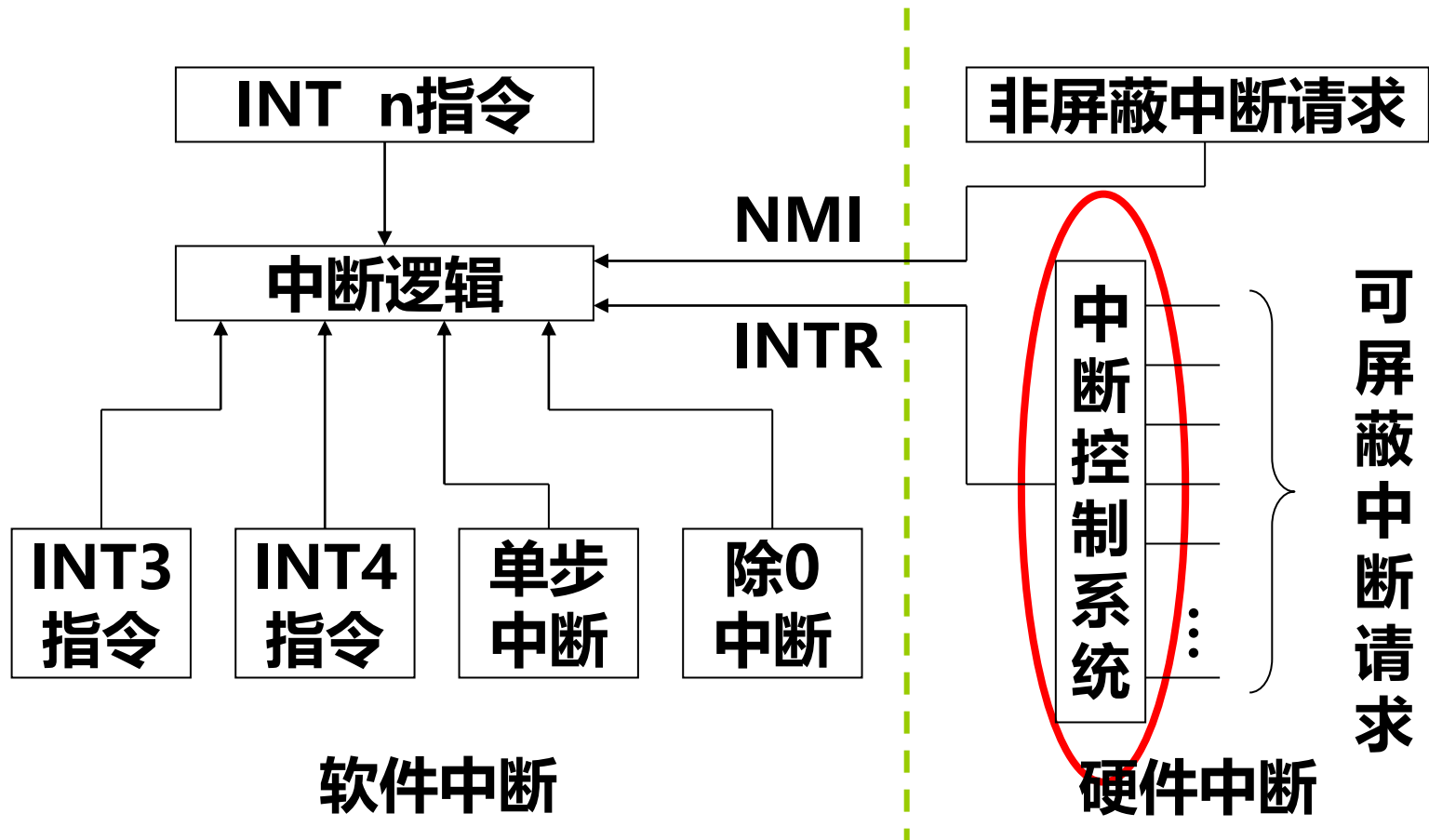


11.6.2 不可屏蔽中断和软件中断的处理过程

由于它们的不可屏蔽性，并且其中断号的获取方法与可屏蔽中断不一样，所以其中断处理过程也有所差别。其主要差别是：

- **不需通过中断响应周期获取中断号，是由系统分配的；**
- **中断服务程序结束，不需发中断结束命令EOI，是自动结束方式。**

11.7 中断控制器



16位机的中断系统

11.7 中断控制器82C59A

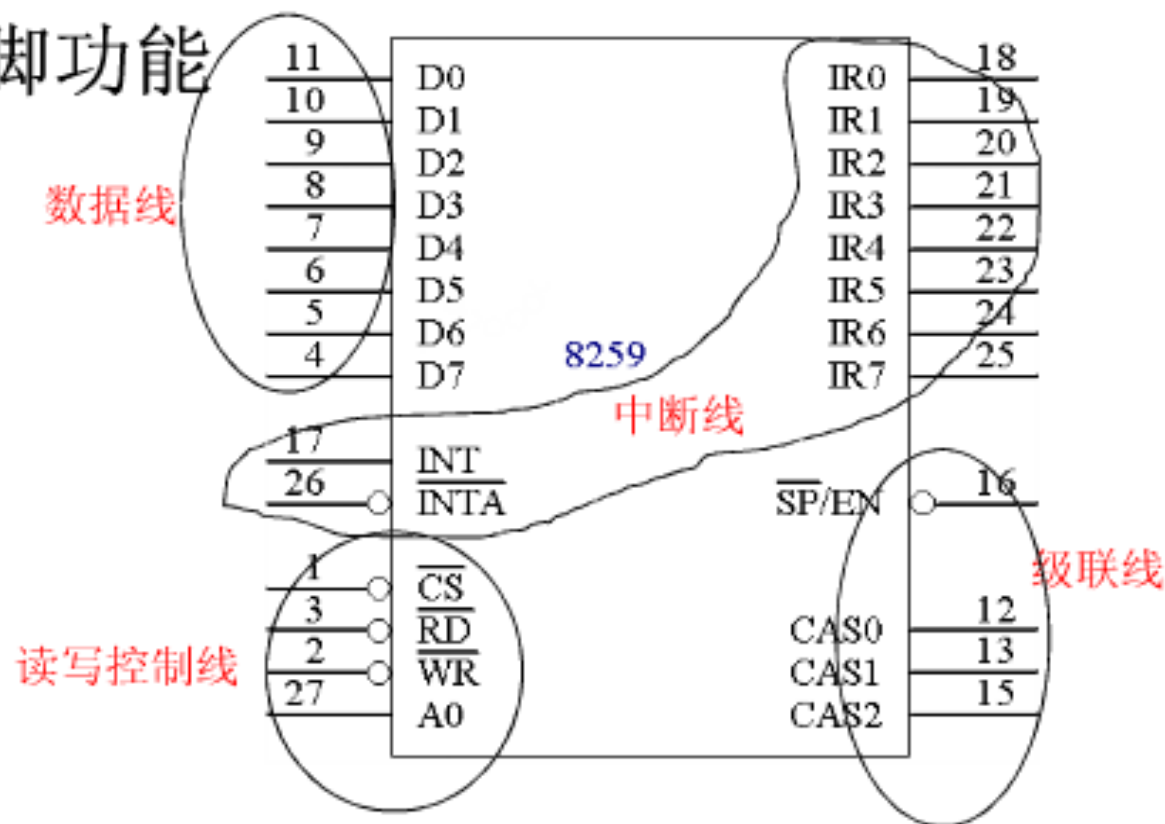
- Intel 82C59A是一种**可编程中断控制器**，可协助CPU对系统的中断进行**管理**。
- 82C59A的**功能**:
 - ▲ 接受和扩充外设的中断请求
 - ▲ 中断优先级排队
 - ▲ 提供中断类型号
 - ▲ 屏蔽或打开中断
 - ▲ 接受CPU命令或返回当前工作状态
- 一片82C59A可**直接管理8级中断**，通过级联可扩展至64级中断。
- 通过对82C59A进行编程，可设置多种工作方式，以满足不同中断系统的需要。

11.7 中断控制器82C59A

11.7.1 82C59A外部特性和内部寄存器

- 外部特性：有三组信号线
 - ① 面向CPU的信号
 - ② 面向外设的8根中断请求线
 - ③ 面向同类芯片的中断请求级联线

8259引脚功能



INT: 0

INTA: I

8086 CPU的INTA (Interrupt Acknowledge) 中断响应信号输出引脚，低电平有效，该引脚是CPU响应中断请求后，向8259发出的认可信号，用以通知8259作出相关准备，比如提供中断号、对内部寄存器做相关操作等。

*8259无外部时钟

8259的总线连接方式和Sp/En

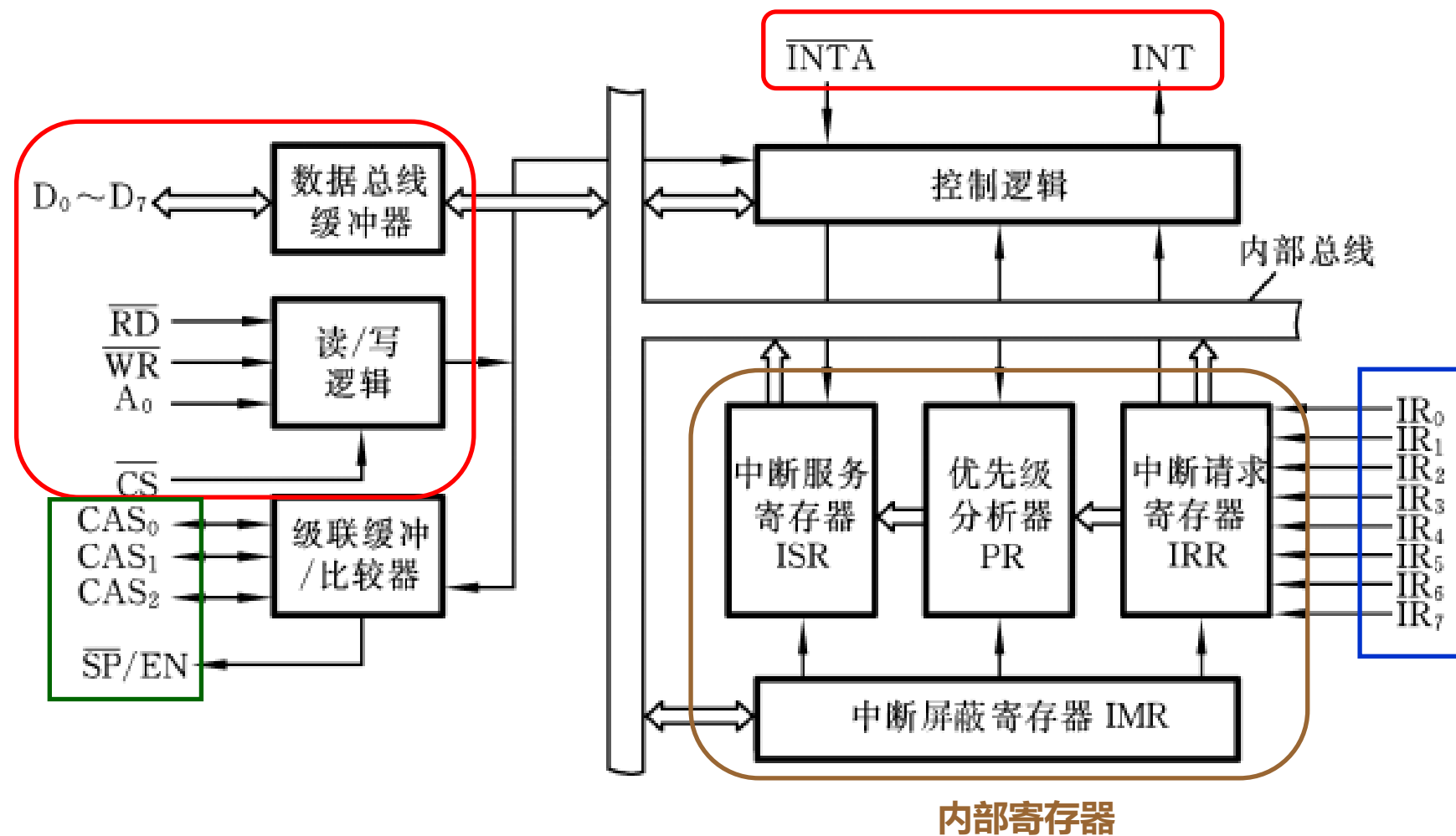
- **缓冲方式:**8259A的数据线引脚设计是可与系统数据总线直接相连，但有时有一片或多片8259A希望有自己的局部数据总线，这时需要在8259A的数据线与系统数据总线之间加一个缓冲器。
- **双向控制:** 因为数据传送是双向的，所以该缓冲器也应是双向的。在缓冲器方式下，SP/EN为输出信号，用作控制缓冲器是接收方向还是发送方向。
- **主从区分:** 当不处于缓冲方式时，SP/EN为输入信号，用来指明该8259A是主片 (SP=1) 还是从片 (SP=0)。

级联方式下，CAS0-CAS2：

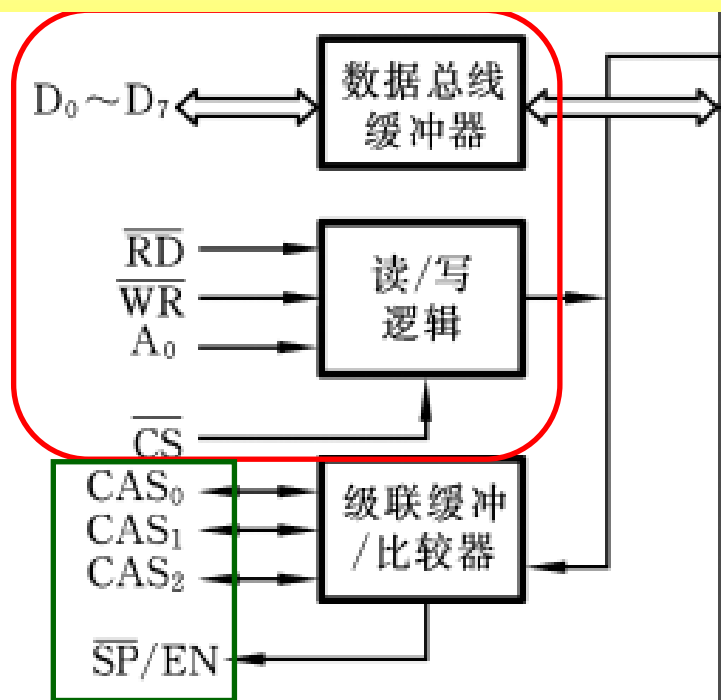
- 主、从8259的级联线，传送从片识别码，仅使用主片工作时不用这三根线。
- 在一主多从的级联方式下，主8259的CAS2-CAS0与所有从8259的CAS2 ~ CAS0联在一起，主8259的CAS2 ~ CAS0作为输出，从8259的CAS2 ~ CAS0作为输入，3-8译码选择。
- 当第一个INTA信号到来时，主8259根据自身优先级排列结果进行判断，若认为当前优先级最高的中断请求是来自从片x，那么主片的CAS2 ~ CAS0输出从片x的编码ID2 ~ ID0,用于从多个从片中选出从片x出来。每一个从8259收到该编码后，与其自身的ID2 ~ ID0（在ICW3中设置）比较，如果相等，则该从片x根据自身的中断源优先级排队情况选出一个优先级最高的中断源y，在第二个INTA到来时，该从片x在总线上发出y对应的中断类型号。

打个比方

- CPU是首长，主8259是大秘，从8259是二秘。大秘、二秘都有小本本(4页，即框图中的REG)，记录需要向首长汇报的事项。哪些事项马上需要让领导处理的、哪些暂缓的，都由秘书决定（事先得了首长的吩咐了ICW/OCW）。
- 首长只能决定是否要处理中断，但不能决定是哪个中断，得由大秘或二秘决定。
- 首长只是不断的处理大秘发过来的请求，但具体是什么请求得由大秘或二秘告知其中断号才行。各中断源都会收到来自CPU的INTA信号，但此时都不能发出自己的中断号。
 - 只有一个大秘时，大秘直接向首长发出中断号。
 - 当有多个二秘时，只有收到大秘CAS信号的那个二秘才能发出中断号，其它的二秘不能发出，否则就乱套了。大秘向哪个二秘发出CAS信号，从自己的小本本上选最高优先级那个二秘。



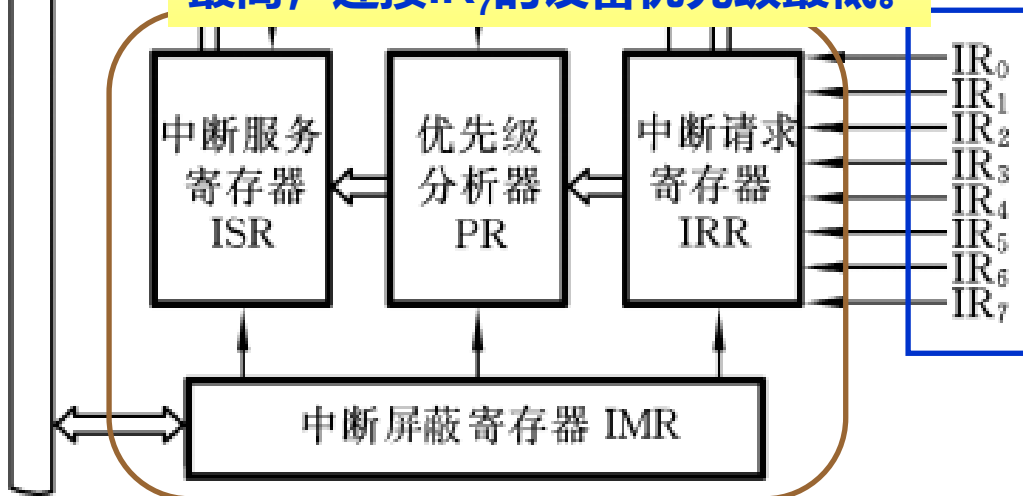
面向CPU的信号线。包括用于CPU发命令及读取中断号的8根数据线 $D_0 \sim D_7$ ，一对中断请求线INT和中断回答线 $\overline{INTA}$ ，以及控制线 $\overline{RD}$ 、 $\overline{WR}$ 与地址线 $\overline{CS}$ 、 A_0 。



面向同类芯片的中断级联信号线，用于扩展中断源，包括主/从芯片的设定线 $\overline{SP/EN}$ ，3根用以传送从片识别码的级联线 $CAS_0 \sim CAS_2$ 。



面向外设的8根中断申请线 $IR_0 \sim IR_7$ ，其作用有二：一是接收外设的中断申请；二是作外部中断优先级排队用。连接 IR_0 的设备优先级最高，连接 IR_7 的设备优先级最低。



内部寄存器

几个缩写字母的含义

- ▲ IRR: 中断请求寄存器 (Interrupt Request Register)
 - ▲ IMR: 中断屏蔽寄存器 (Interrupt Mask Register)
 - ▲ ISR: 中断服务寄存器 (Interrupt Service Register)
 - ▲ PR: 优先级分析器 (Priority Resolver)
-
- ▲ ICW: 初始化命令字 (Initialization Command Word)
 - ▲ OCW: 操作命令字 (Operation Command Word)

■ 内部寄存器

① 中断请求寄存器IRR

8位，以逻辑1记录已经提出中断请求的中断级，等待CPU响应。当提出中断请求的外设产生中断时，由82C59A置位，直到中断被响应才自动清零。IRR的内容可以由CPU通过OCW3命令读出。

② 中断屏蔽寄存器IMR

8位，对中断请求 IR_i 起屏蔽作用。哪一级中断被屏蔽，哪一位就置“1”，即禁止 IR_i 提出中断请求。该寄存器不可读。

$$D_i = \begin{cases} 0 & \text{开中断} \\ 1 & \text{关中断} \end{cases}$$

③ 中断服务寄存器ISR

8位，以逻辑1记录已经已被响应并正在服务的中断级，包括尚未服务完中途被挂起的中断级，以便与后面新来的中断请求的级别进行比较。

当中断请求 IR_i 获得中断响应时， $\overline{INTA_1}$ 将 ISR_i 置1，表明 IR_i 正处于中断服务，则该位对应的中断源之后发出的INT请求被挂起，其中断请求不会被响应。因此中断服务结束后必须对 IR_i 清零。

清零有两种方法：由 $\overline{INTA_2}$ 将该位自动清零；在中断服务程序中用中断结束命令EOI强制清零。

④ 优先权分析器PR

中断请求的判优电路。PR把新中断请求的优先级与ISR寄存器中记录在案优先级进行比较，谁的优先级高就让用户申请中断。其操作过程全部由硬件完成，因此**用户是不可访问的**，它不属于82C59A的编程模型之内。

11.7.2 82C59A端口地址

主片端口地址：20H和21H

从片端口地址：0A0H和0A1H

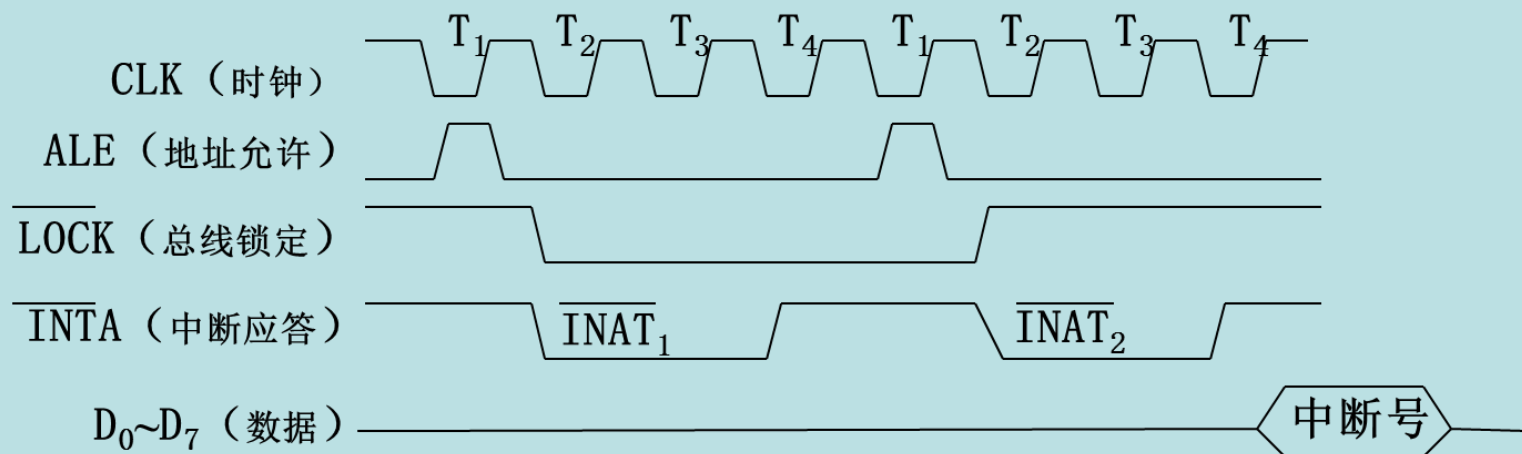
具体使用哪个端口地址由初始化命令ICW和操作命令OCW的标志位A0指示。

下面是中断申请与响应握手过程(硬件自动完成, 程序员不可见)

① INTA_1 是CPU输出的中断应答信号, 用于通知8259:

“你发出的中断请求已被响应”; 然后8259A将允许的中断级在ISR (中断服务寄存器) 中某位置进行置位, 清零相应IRR位 (中断请求寄存器), 然后准备好中断号。

② INTA_2 是CPU第二次输出的中断应答信号, 用于读取8259提供的中断类型号, 8259A将准备好的中断号送到数据总线上。



书上有误

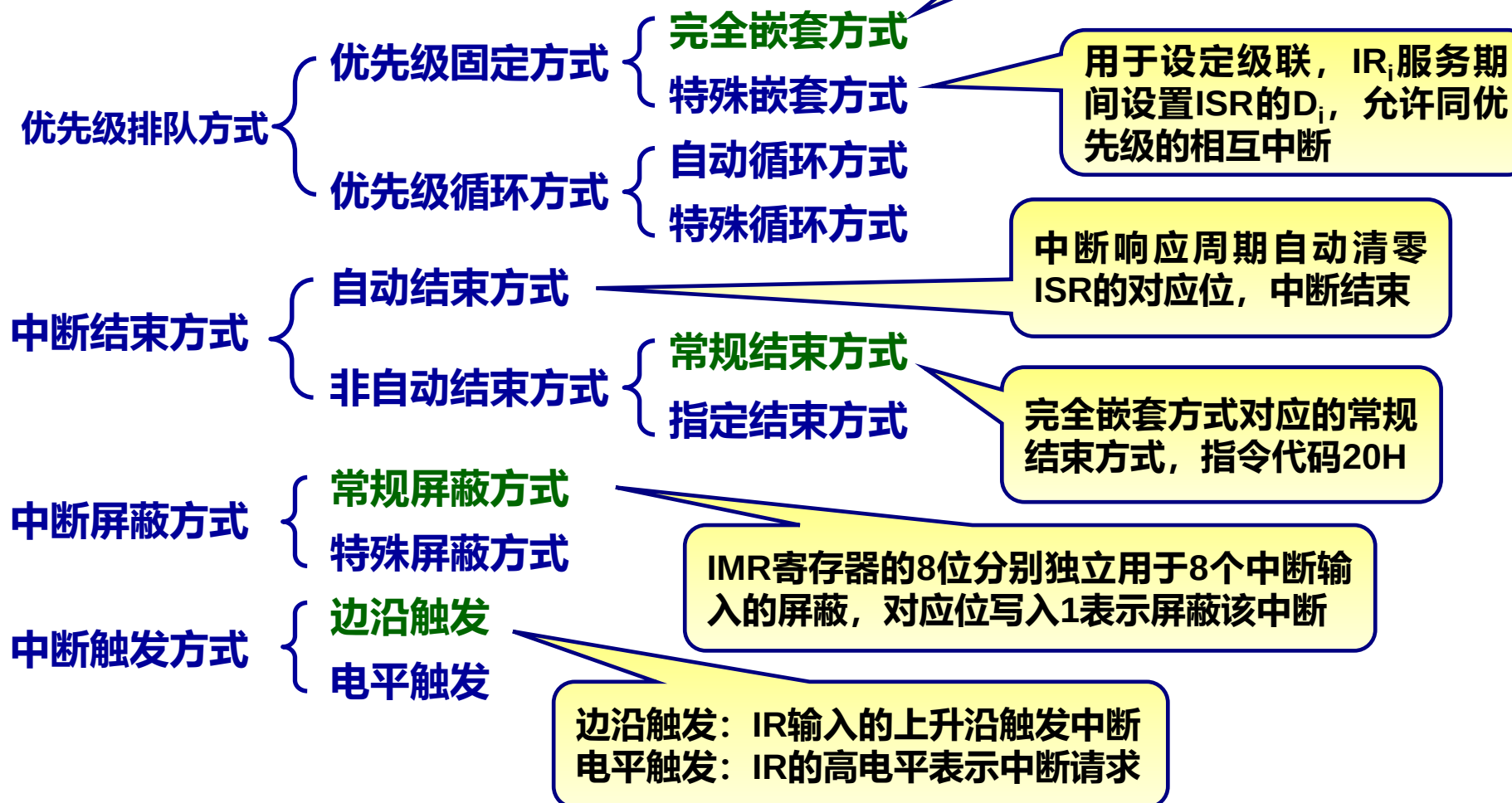
1. 置位中断服务寄存器ISR

当总线控制器发出**第一个脉冲 $\overline{INTA}$** 时，CPU输出有效的**总线锁定信号 $\overline{LOCK}$** （低电平），使总线在此期间处于**封锁状态**，防止其他处理器或DMA控制器占用总线。与此同时，**中断控制器将判优后允许的中断级在ISR中的相应位置1**，以**登记正在服务的中断级别**，在中断服务程序执行完毕之后，**该寄存器自身不能清零**，需要向中断控制器发中断结束命令EOI才能清零。

2. 读取中断号

当总线控制器发出**第二个2脉冲 $\overline{INTA}$** 时，**总线锁定信号撤除**，总线被解封，地址允许信号ALE也变为低电平（无效），即允许数据线工作。正好此时**中断控制器将当前中断服务程序的中断号送到数据线上**，由**CPU读入**。

11.7.3 82C59A的工作方式



1、中断触发方式

确定中断请求的启动方式

- ① **边沿触发方式**。IR0-IR7线上出现**由低到高的跳变**，表示有中断请求。
- ② **电平触发方式**。IR0-IR7线上出现**高电平时**，表示有中断请求。

2、中断级联方式

82C59A可以单片使用，也可以多片使用，级联方式应用时有**主片和从片**的问题。最大级联方式为1主片8从片，64个外部中断。

3、中断屏蔽方式

允许还是不允许外设提出中断请求，而不是对已经提出的中断申请响不响应的问题

① 常规屏蔽方式

通过屏蔽寄存器写入8位屏蔽码来屏蔽或开放8个中断申请线 IR_0 - IR_7 上的中断请求。屏蔽哪个中断申请则将屏蔽码的相应位置为1，反之则置为0。

② 特殊屏蔽方式

用于开放低级别的中断申请，允许比正在服务的中断级别低的中断申请。该方式很少使用。

4、中断优先级排队方式

- ① **完全嵌套方式**：只能高优先级中断低优先级的中断
- ② **特殊完全嵌套方式**：同级优先级的中断可以相互中断
- ③ **优先级轮换方式**：被服务完的中断源优先级将自动变为最低，而其后的中断源的优先级将变为最高。优先级轮换方式有两种：

优先级**自动轮换**方式：当一个中断被服务后，自动降为最低，初始顺序是： $IR_0, IR_1, \dots$

优先级**指定轮换**方式：与 的差别是，初始轮换的优先级可以由程序指定

5、中断结束方式

① 自动结束方式（少用）

在中断响应周期，就自动清零了该中断源的ISR寄存器中被置1的位。这常用于没有中断嵌套的系统。

② 非自动结束方式（常用）

在中断服务程序后，必须使用**中断结束命令EOI**才能清除ISR中的相应位。

指定中断结束命令：即要指定结束中断请求 IR_i 的编号。**代码 6XH(X:0-7)**

常规中断结束命令：不指定 IR_i 的编号。

用于支持中断嵌套功能的系统。**代码 20H**

11.7.4 82C59A的编程命令

中断操作功能：

①请求 ②屏蔽 ③排队 ④结束

⑤级联 ⑥中断类型号 ⑦查询

下面讨论为实现这些功能的各个命令字的含义，为编程使用82C59A提供一些思路。

1、初始化命令字（ICW1-ICW4）

- 4个初始化命令的作用：
 - ICW1：进行**中断触发和单片/多片设置**；
 - ICW2：进行**中断号设置**；
 - ICW3：进行**级联设置**；
 - ICW4：进行**优先级和结束方式设置**；
- 设置中断屏蔽方式是**默认常规屏蔽方式**，如要改动，可以执行操作命令OCW3

**注意：必须在对82C59A初始化时完成；
需要按顺序发送ICW1 – ICW4；**

特征位F

中断触发和单片/多片设置 ICW1

D ₇	D ₆	D ₅	D ₄	D ₃	D ₂	D ₁	D ₀
0	0	0	1	LTIM	0	SNGL	IC4

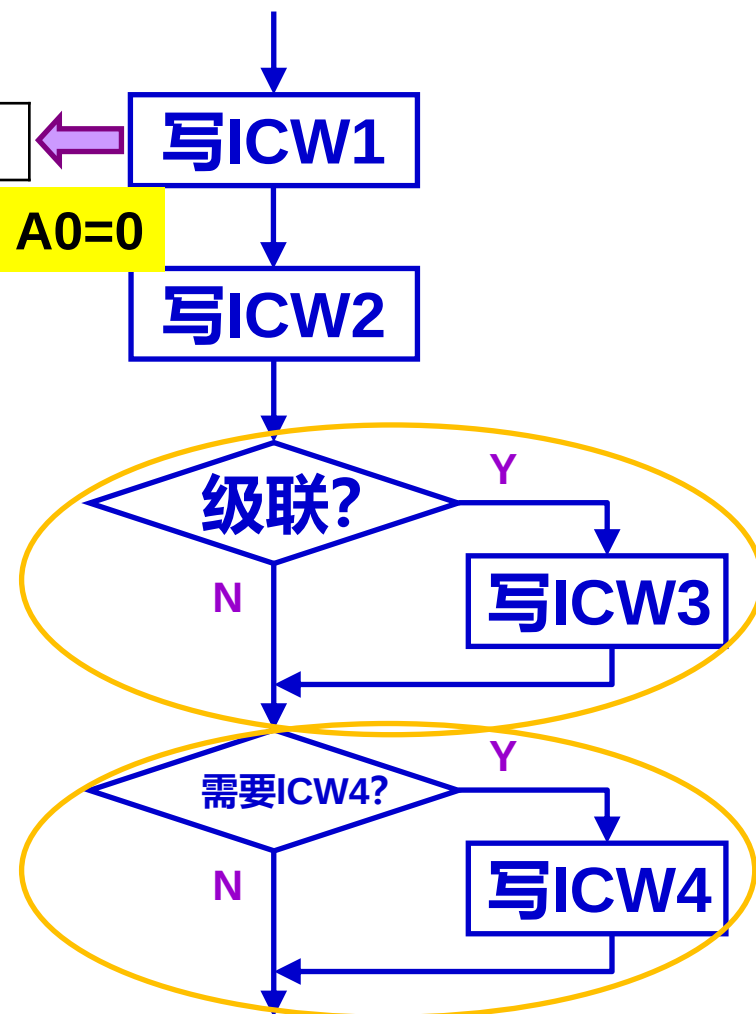
触发电平:
0=边沿触发
1=电平触发

级联模式:
0=级联方式
1=单片方式

ICW4选项:
0=不写ICW4
1=写ICW4

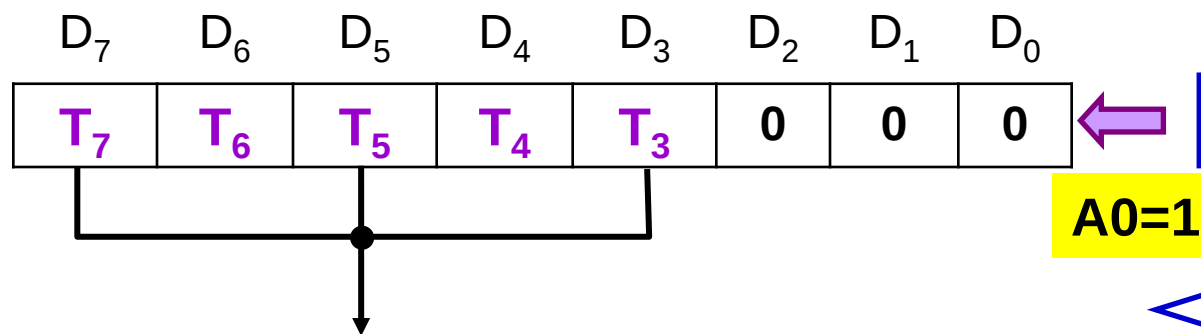
例.若8259A采用电平触发, 单片使用, 需要ICW₄ (对16位机以上, D₀=1), 则:

MOV AL, 00011011B ; ICW₁的内容
OUT 20H, AL ; 写入ICW₁端口(A₀=0)

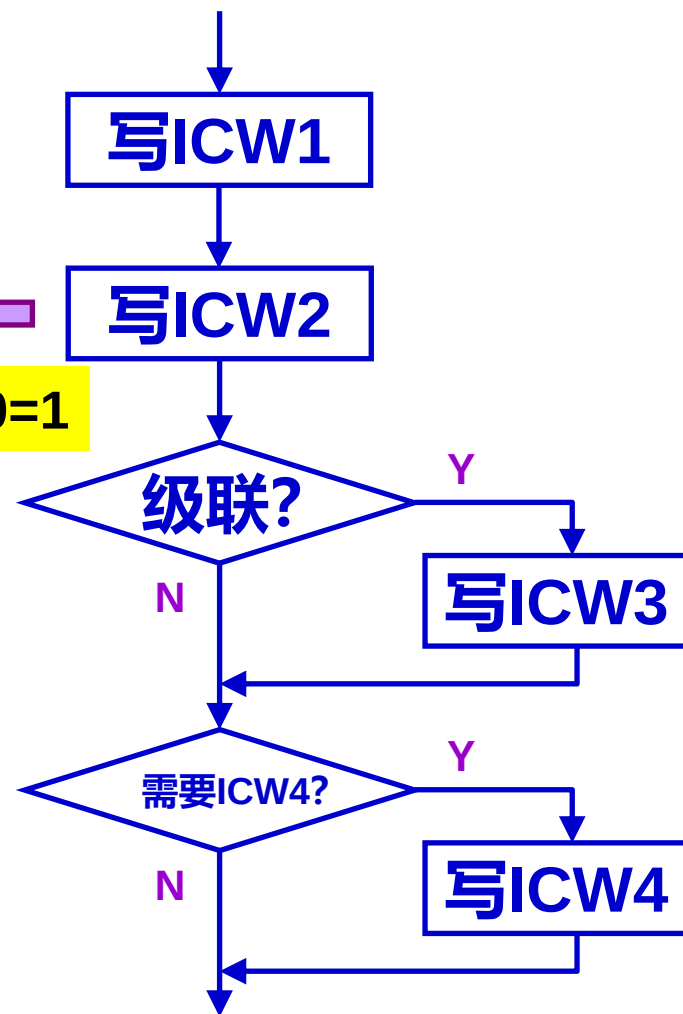


初始化的命令顺序

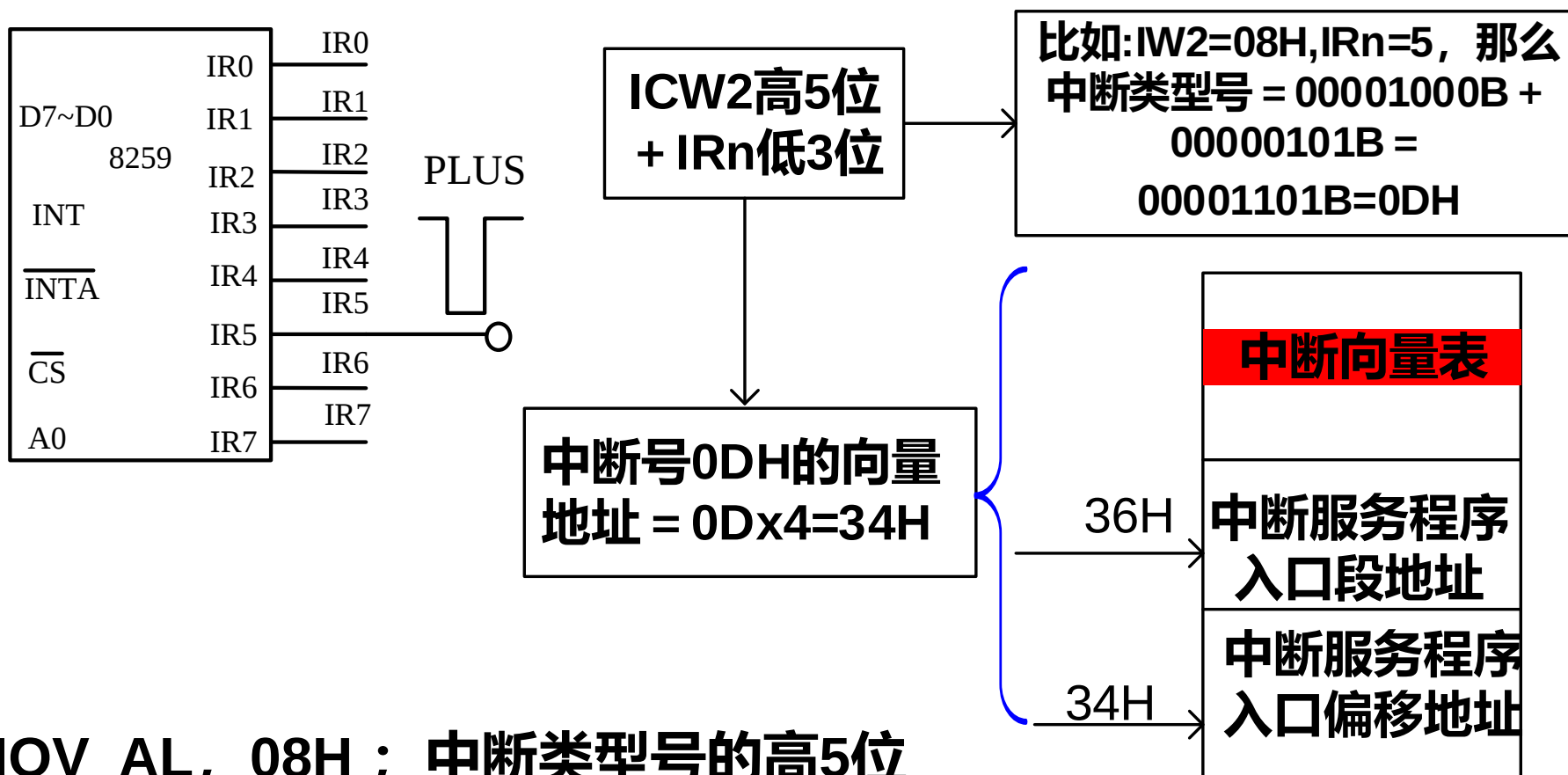
ICW2 中断号设置



中断号高5位：用于在中断响应周期送出中断号。低3位根据中断申请IR_i的引脚编号i决定。由82C59A自动填写，只写高5位



例：硬盘中断号0DH，即00001**101**B



MOV AL, 08H ; 中断类型号的高5位
OUT 21H, AL ; 写入ICW₂的端口 (A₀=1)

主片ICW3设置:

1:主片的某个IR上连有从片

0:主片的某个IR上未连从片

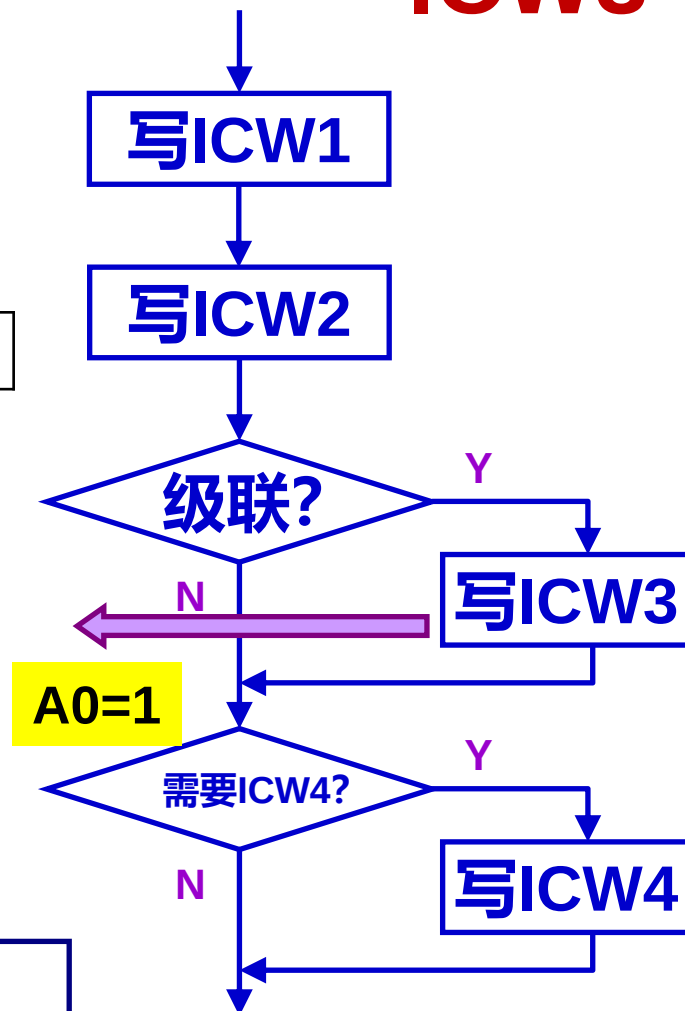
D ₇	D ₆	D ₅	D ₄	D ₃	D ₂	D ₁	D ₀
S7	S6	S5	S4	S3	S2	S1	S0

D ₇	D ₆	D ₅	D ₄	D ₃	D ₂	D ₁	D ₀
X	X	X	X	X	ID2	ID1	ID0

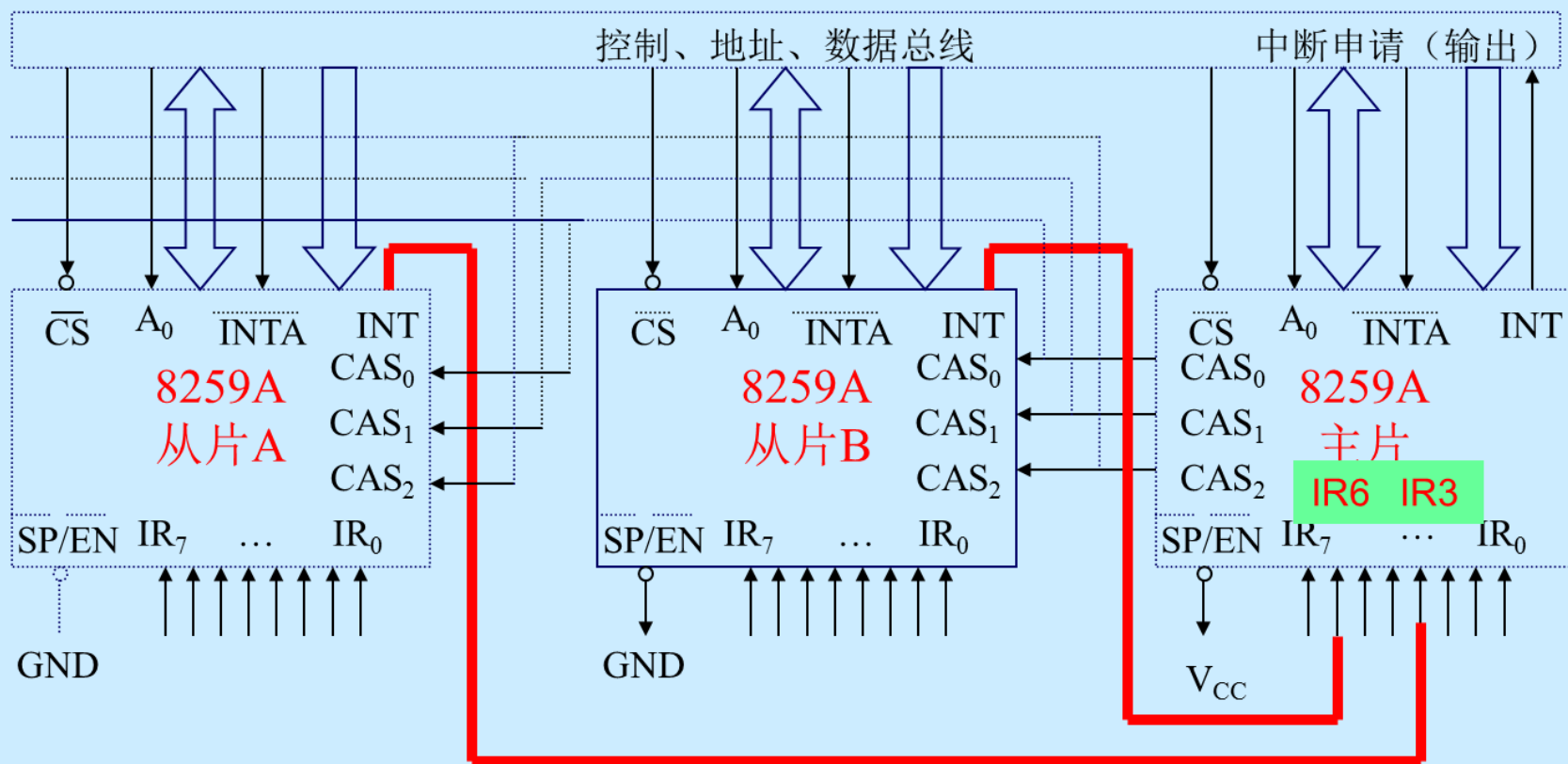
从片ICW3:

从片的INT引脚连到主片的IR引脚编号: 000-111

级联设置 ICW3



例如：从片A和从片B的请求线INT连到主片的IR₃和IR₆



则初始化程序:

主片的 $ICW_3=01001000B=48H$

```
MOV AL, 48H      ; 主片的 $ICW_3$   
OUT 21H, AL
```

从片A的 $ICW_3=00000011B=03H$

```
MOV AL, 03H      ; 从片A的 $ICW_3$   
OUT 从1, AL
```

从片B的 $ICW_3=00000110B=06H$

```
MOV AL, 06H      ; 从片B的 $ICW_3$   
OUT 从2, AL
```

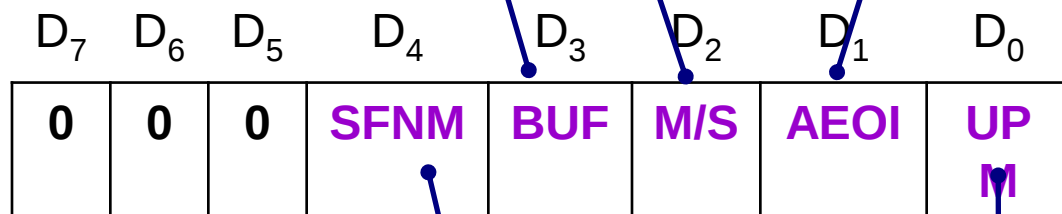
优先级和结束方式设置

当 BUF=0时
M/S位不起作用

主/从片设定:
0=从片
1=主片

数据线缓冲方式:
0=非缓冲
1=缓冲

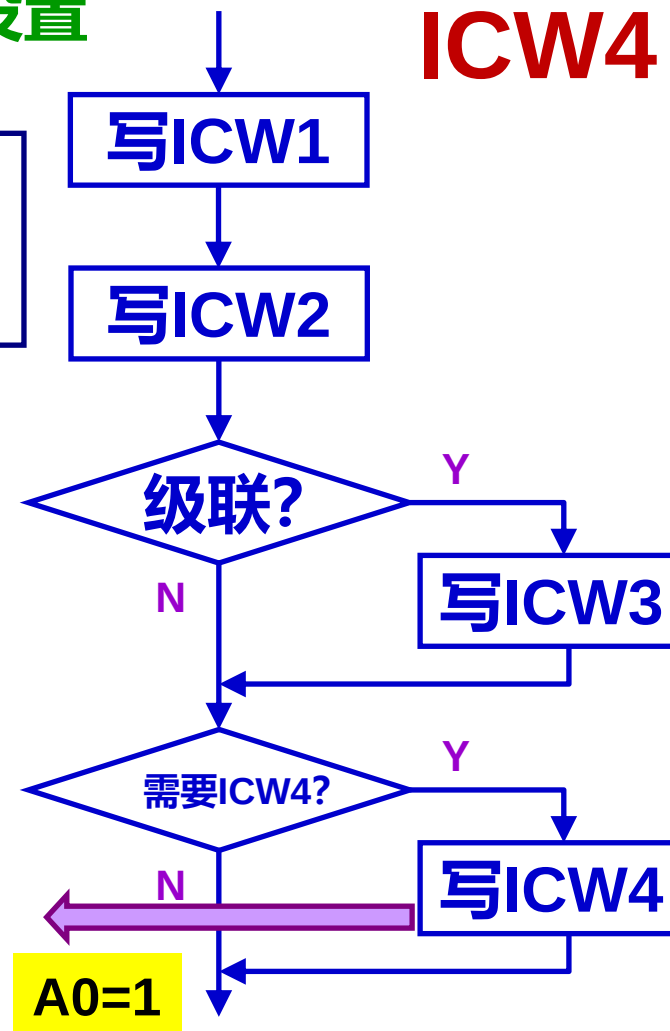
中断结束方式:
0=非自动结束方式
1=自动结束方式



优先权方式:
0=完全嵌套(高抢低)
1=特殊嵌套(同级可抢)

CPU选择:
0=8位系统
1=16位系统

ICW4



例 某计算机的CPU为8086，8259A与系统总线之间采用缓冲器连接，非自动结束方式，只用1片8259A，正常完全嵌套。

则：ICW₄=

00001101B=0DH

相应的初始化程序段：

```
MOV AL, 0DH
```

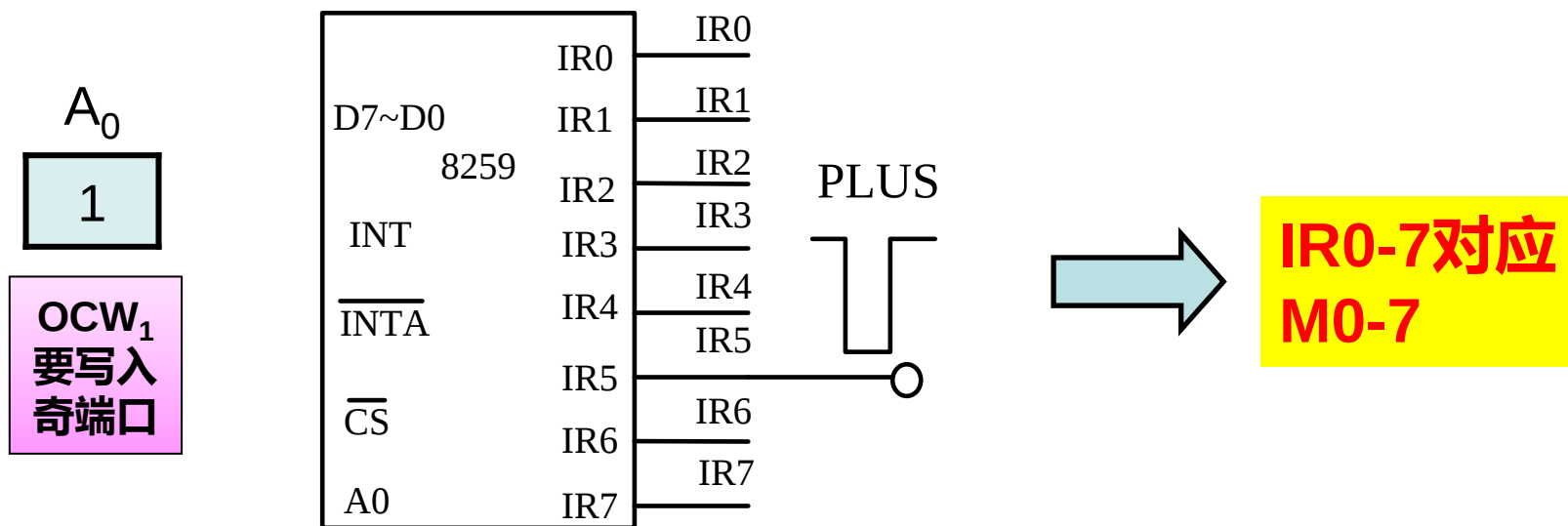
```
OUT 21H, AL
```

2、操作命令字（OCW₁-OCW₃）

- 初始化后，使用OCW命令对中断控制器进行**动态控制**
- **不需要按顺序发送**，在程序中任意安排
- OCW1-OCW3的功能：
 - ✓ OCW1：进行常规的**屏蔽开放操作**
 - ✓ OCW2：进行**中断结束和优先级轮换排队操作**
 - ✓ OCW3：**设定屏蔽方式和读取状态**

- OCW1: 用于设定中断屏蔽, 每一位对应一个中断输入。
0=开放, 1=屏蔽

D ₇	D ₆	D ₅	D ₄	D ₃	D ₂	D ₁	D ₀
M ₇	M ₆	M ₅	M ₄	M ₃	M ₂	M ₁	M ₀



例 要使中断源IR₃开放，其余被屏蔽

则：OCW₁=

11110111B=F7H

在主程序中，在开中断之前，写相应的程序段：

MOV AL, 0F7H

OUT 21H, AL ; OCW₁内容送端口

此程序的最大问题是影响了其它中断屏蔽位

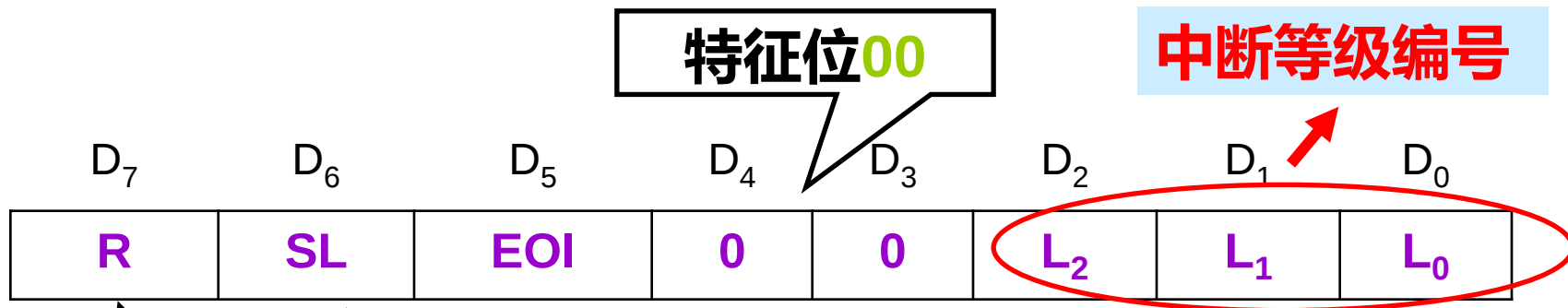
如果不影响其它中断，应该写成下列程序段：

IN AL, 21H ; 回读21H端口的内容

AND AL, 11110111B

OUT 21H, AL

● OCW2: 用于中断结束操作和优先级循环排队操作



R: 优先级轮换
0: 完全嵌套方式
1: 轮换优先级

EOI: 中断结束
0: 自动中断结束 (清最高级)
1: 非自动结束

SL: 指定中断结束
 当 EOI=0: 最高级中断结束;
 当 EOI=1:
 SL=0 : 结束当前中断, L₂L₁L₀ 任意值;
 SL=1 : 结束指定中断, 由 L₂L₁L₀ 指定

A₀

0

OCW₂
要写入
偶端口

中断结束控制 (OCW₂格式中的D₆)

当初始化命令ICW₄选用非自动结束方式时

(AEOI=0) , 就利用OCW₂来控制中断结束:

SL (D₆) =0 , 结束当前中断

OCW₂=00100XXXXB(L₂~L₀=任意值, 常用000)

SL (D₆) =1, 结束指定中断

OCW₂=01100L₂L₁L₀

例: 指定IR₅的中断结束, 则:

OCW₂=01100101B=65H

作中断优先级排队控制 (OCW₂格式中的D₇)

D₇ (R) { 1: 需要轮换优先级, 分二种 SL= { =1: 优先权指定轮换
=0: 自动轮换
0: 优先级固定方式, 则为完全嵌套方式

注意: 中断结束方式和中断排队方式都用OCW₂命令来实现, 因此它的某些位功能相互交叉显然比较复杂, 但其作用只有两个:

- 控制中断结束方式
- 控制中断优先权轮换方式

OCW2小结

R	SL	EOI	0 0	L2 L1 L0	功能
0	0	0	0 0	0 0 0	自动EOI、优先级固定命令(完全嵌套)
0	0	1	0 0	0 0 0	非自动EOI[不指定]、优先级固定命令20H
0	1	1	0 0	L2 L1 L0	非自动EOI[指定]、优先级固定命令6XH
1	0	0	0 0	0 0 0	自动EOI、优先级自动轮转命令
1	0	1	0 0	0 0 0	非自动EOI[不指定]、优先级自动轮转命令
1	1	1	0 0	L2 L1 L0	非自动EOI[指定]、优先级指定轮转命令
1	1	0	0 0	L2 L1 L0	自动EOI、优先级指定轮转命令

重点

例1 要对 IR_3 中断指定中断结束方式,

$OCW_2 = 01100011B = 63H$ 。在中断服务程序中,
中断返回指令IRET前, 写相应的程序段:

```
MOV AL, 63H
```

```
OUT 20H, AL ; 写入 $OCW_2$ 端口( $A_0=0$ )
```

例2 对 IR_3 中断采用不指定中断结束方式,

$OCW_2 = 00100000B = 20H$ 。在中断服务程序中,
中断返回指令IRET前, 写相应的程序段:

```
MOV AL, 20H
```

```
OUT 20H, AL ; 写入 $OCW_2$ 端口( $A_0=0$ )
```

- OCW3的功能：用于设定屏蔽方式和读取状态

读取8259A的IRR、ISR寄存器的内容

设置特定屏蔽方式

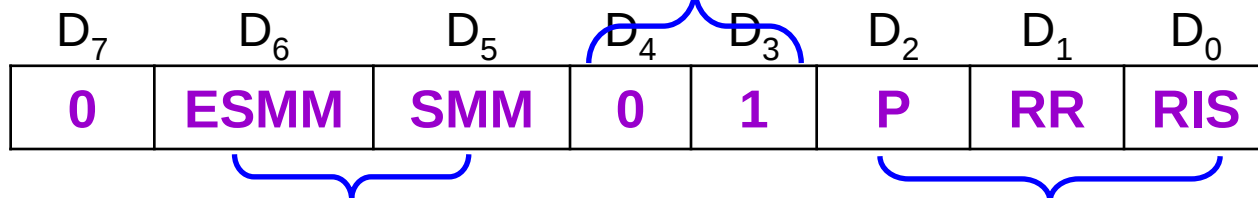
查询中断方式

特征位01

A₀

0

OCW₃
要写入
偶端口



屏蔽方式:

0X=无操作

10=常规屏蔽方式

11=特殊屏蔽方式

读取选择命令：写完OCW3后，接下读取

00X=无操作

010=IRR

011=ISR

1XX=查询字，可查询当前中断状态

读状态操作 (OCW₃格式中的D₁D₀)

8259A 内部有3个寄存器IRR、ISR、IMR可供CPU读出当前的内容。

(1) 读IRR和ISR的方法

CPU先指定读哪个寄存器，然后再发IN指令，才能读取ISR和IRR中的内容。当8259A初始化后，自动指向读IRR寄存器。

(2) 读IMR的方法

不需要先发指定命令，只要读奇地址端口。

例1：读ISR寄存器的内容

分两步：

通过OCW₃指定被读的寄存器，

OCW₃为0001011B，表示下个RD要读ISR

用IN指令读出ISR寄存器的内容

程序段如下：

MOV AL, 0BH ; OCW₃表示读ISR

OUT 20H, AL ; 20H为OCW₃的端口地址

IN AL, 20H ; 读ISR寄存器

例2：读IRR寄存器的内容

分两步：

通过OCW3指定被读的寄存器

OCW3为0001010B，表示下个RD要读IRR

用IN指令读出IRR寄存器的内容

程序段如下：

```
MOV AL, 0AH ; OCW3表示读IRR
```

```
OUT 20H, AL ; 20H为OCW3的口地址
```

```
IN AL, 20H ; 读IRR寄存器
```

```
MOV AH, AL ; 保存IRR->AH
```

例3：读**查询字**(读回中断状态)

分两步：

通过OCW3发送出查询命令

OCW3为0**00**01**100**B，表示下个RD要读查询字

用IN指令读出查询字的内容

程序段如下：

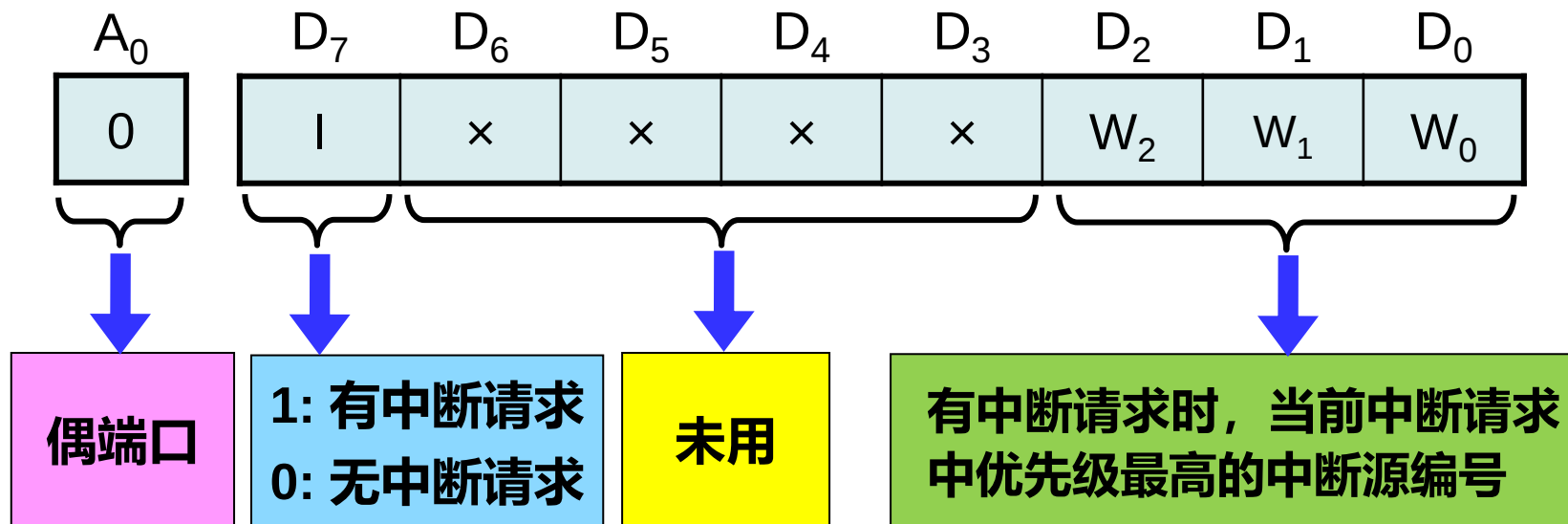
MOV AL, 0CH ; 用OCW3发送查询命令

OUT 20H, AL ; 20H为OCW3的口地址

IN AL, 20H ; 读查询字

MOV AH, AL ; 保存查询字->AH

查询字的格式



当有中断请求时，8259A将IRR中的相应位置1，同时将查询字送到数据总线上供CPU读取。

查询中断前CPU先关闭中断申请，再发查询命令

8259A的编程命令的使用总结

8259A有两类编程命令，初始化命令字ICW和操作命令字OCW，都是通过D0-D7传输的。

- ICW:** 在初始化里完成，必须按一定的顺序
在PC机中已有系统软件完成，不需要也不允许由用户自己去设置
一般只对于没有配置完善操作系统的单板微机上进行
- OCW:** 没有一定的顺序，一般也不安排在程序的开头
在初始化后的任何时候进行动态控制

8259A的同一端口如何区分ICW和OCW?

A0 = 0 (偶地址端口) 写入:

D4=1----ICW1

D4=D3=0----OCW2

D4=0,D3=1----OCW3

A0=1 (奇地址) 写入:

按顺序写。

ICW2->(ICW3->ICW4)-

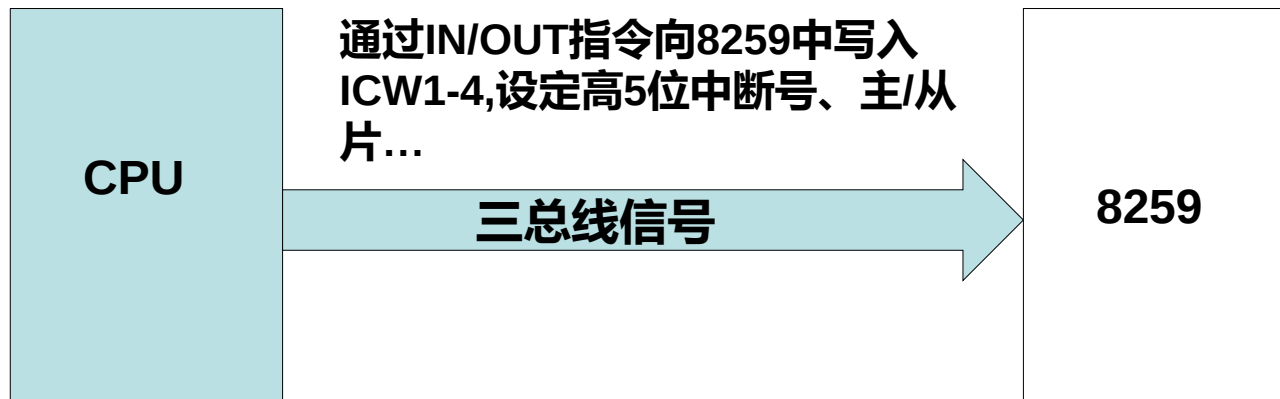
>OCW1



8259工作原理小结（1）

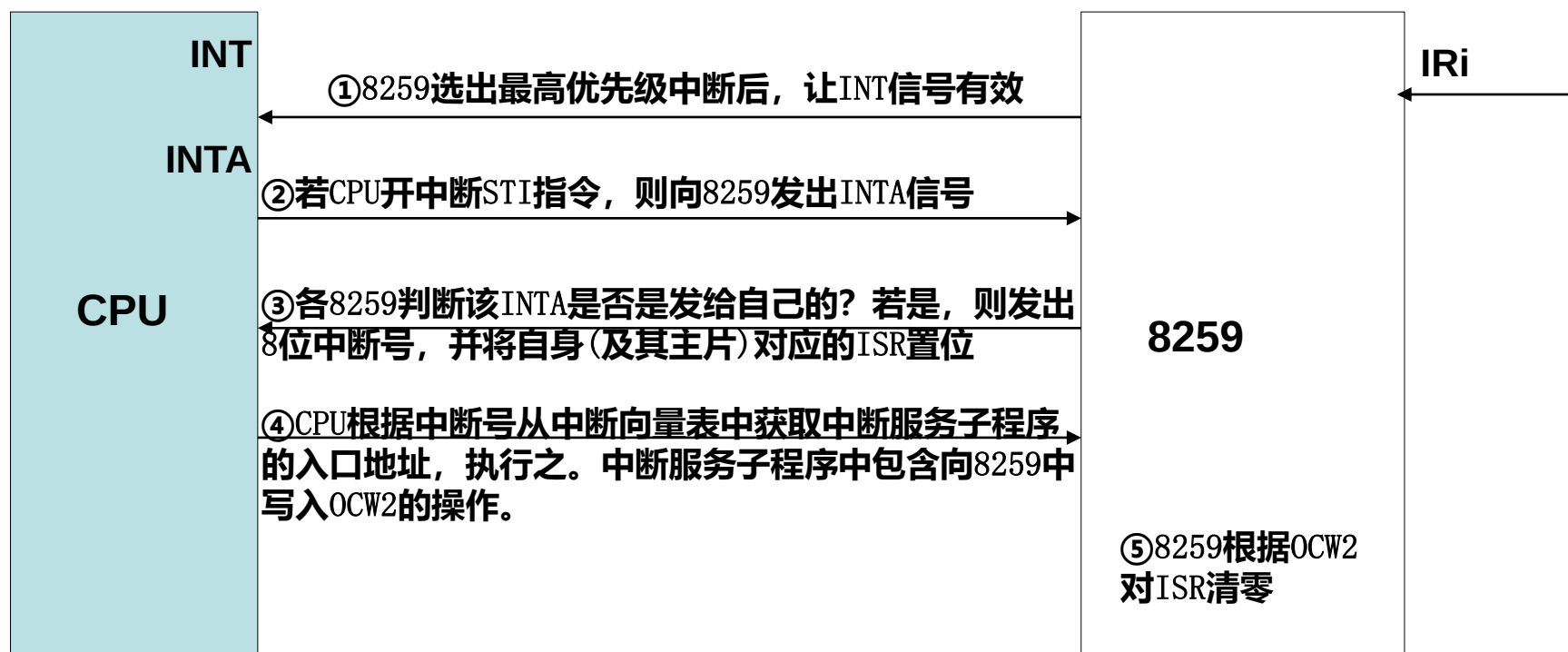
两个阶段：初始化阶段和工作阶段

初始化阶段：



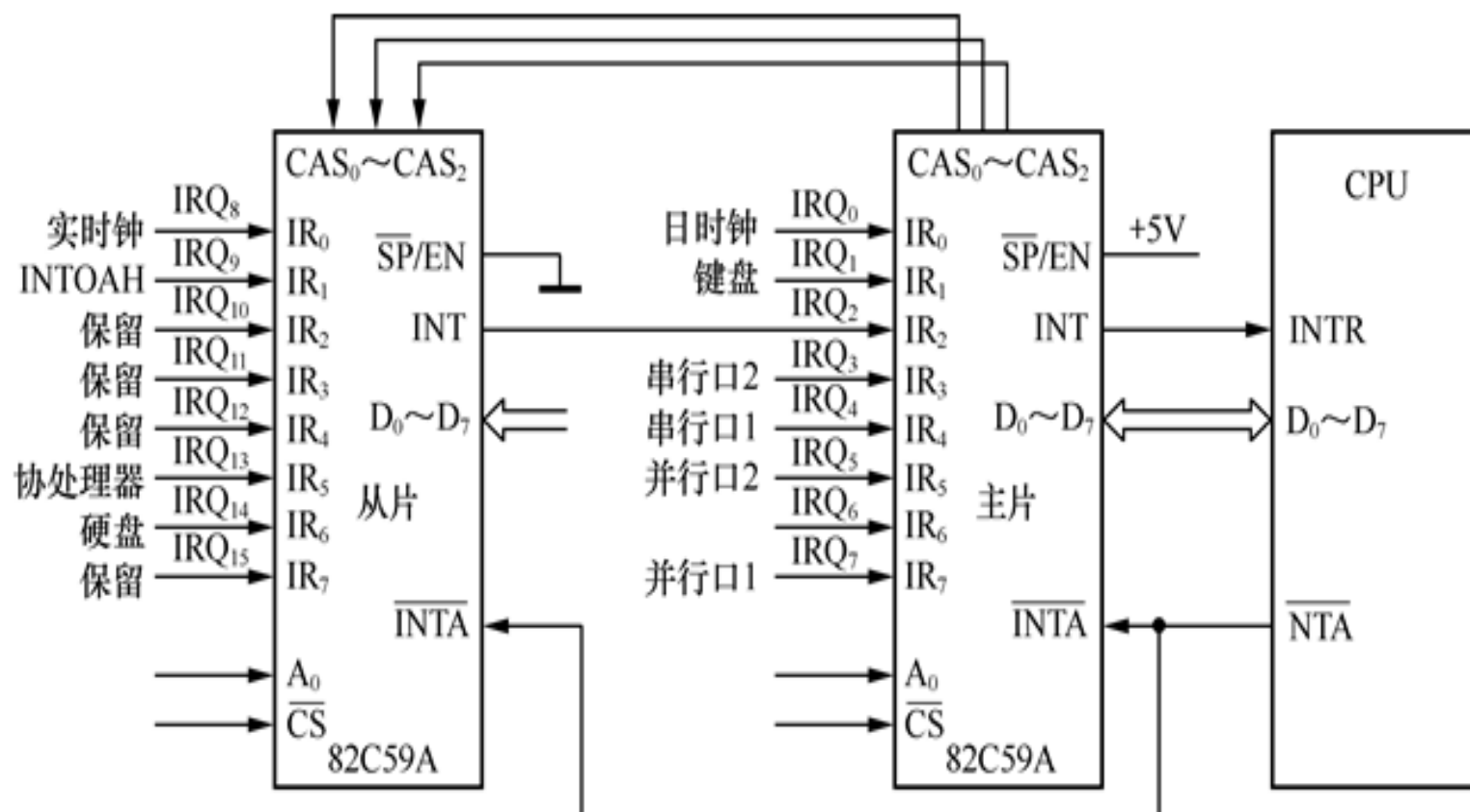
8259工作原理小结 (2)

工作阶段：



11.8 可屏蔽中断系统

11.8.1 可屏蔽中断系统的组成



15级可屏蔽中断的中断号分配情况

主片82C59A8级硬中断源的中断号

中断源	中断号高5位	低3位	中断号
日时钟	08H	IR ₀ (0)	08H
键盘	08H	IR ₁ (1)	09H
保留	08H	IR ₂ (2)	0AH
通信 (二)	08H	IR ₃ (3)	0BH
通信 (一)	08H	IR ₄ (4)	0CH
硬盘	08H	IR ₅ (5)	0DH
软盘	08H	IR ₆ (6)	0EH
打印机	08H	IR ₇ (7)	0FH

从片82C59A8级硬中断源的中断号

中断源	中断号高5位	低3位	中断号
实时钟	70H	IR ₀ (0)	70H
改向 INTOA	70H	IR ₁ (1)	71H
保留	70H	IR ₂ (2)	72H
保留	70H	IR ₃ (3)	73H
保留	70H	IR ₄ (4)	74H
协处理器	70H	IR ₅ (5)	75H
硬盘	70H	IR ₆ (6)	76H
保留	70H	IR ₇ (7)	77H

1. 初始化设置的内容

- (1) **中断触发方式**采用**边沿触发**，上跳变有效。
- (2) **中断屏蔽方式**采用**常规屏蔽方式**，即使用OCW1向IMR写入屏蔽码。
- (3) **中断优先级排队方式**采用**固定优先级的完全嵌套方式**。
- (4) **中断结束方式**采用**非自动结束方式**中的不指定结束和指定结束两种命令格式，即在中断服务程序服务完毕中断返回之前，发结束命令代码20H或6XH均可（X为0~7）。
- (5) **级联方式**采用两片主/从连接方式，并且规定**把从片的中断申请输出引脚INT**连到主片的**中断请求输入引脚IR2**上。两片级联处理15级中断。
- (6) **15级中断号的分配为**：中断号08H~0FH对应IRQ0~IRQ7，中断号70H~77H对应IRQ8~IRQ15。
- (7) **两片82C59A的端口地址分配为**：主片82C59A的两个端口是20H和21H；从片82C59A的两个端口是0A0H和0A1H

2. 初始化设置的程序

■ 初始化流程

系统上电期间，分别对82C59A的主片和从片进行初始化。

主片：

ICW1=11H；边沿触发，多片，要ICW4

ICW2=08H；只写中断号高5位

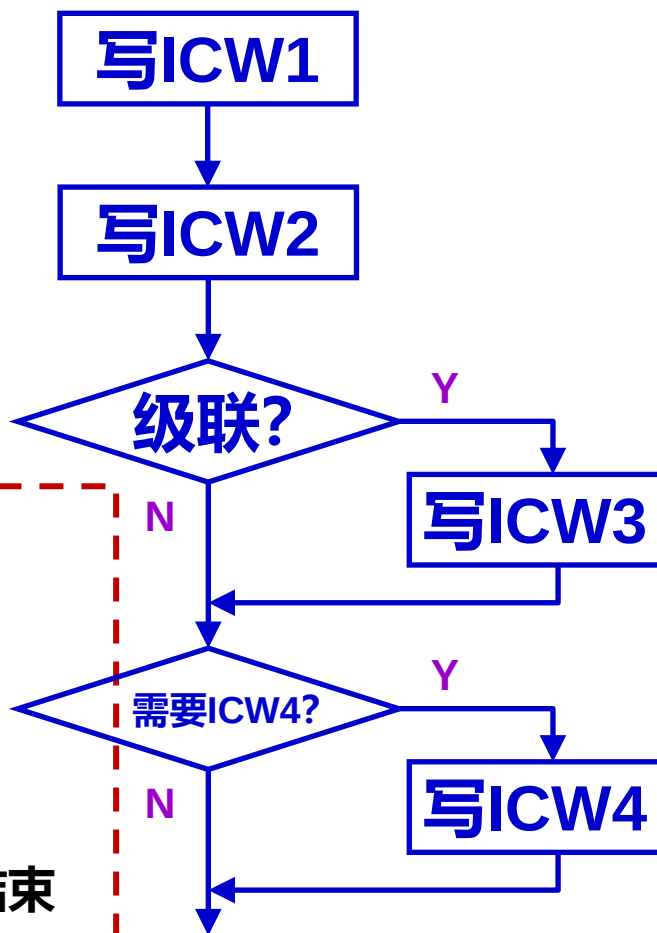
ICW3=04H；主片的IR2接到从片上，D2=1

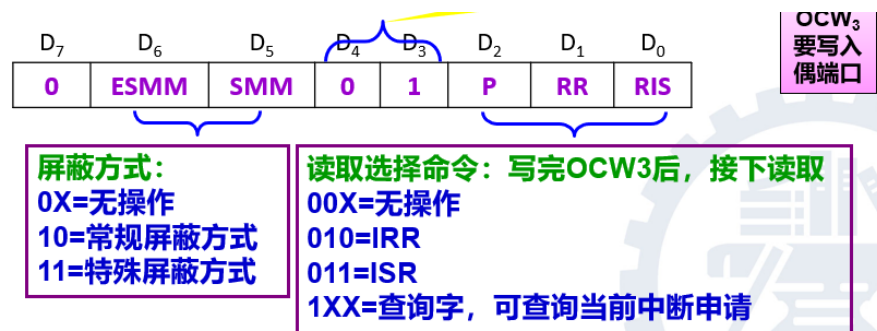
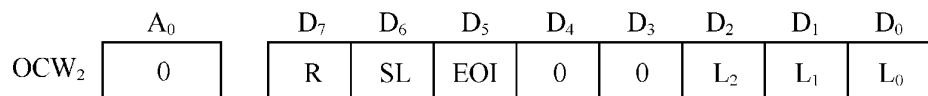
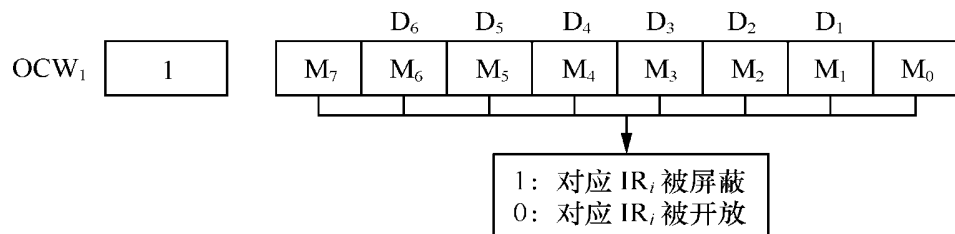
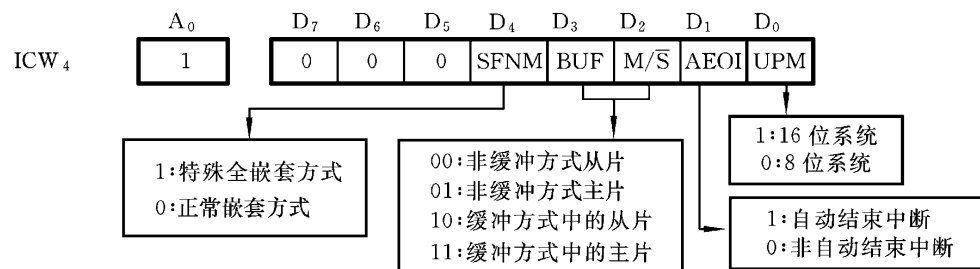
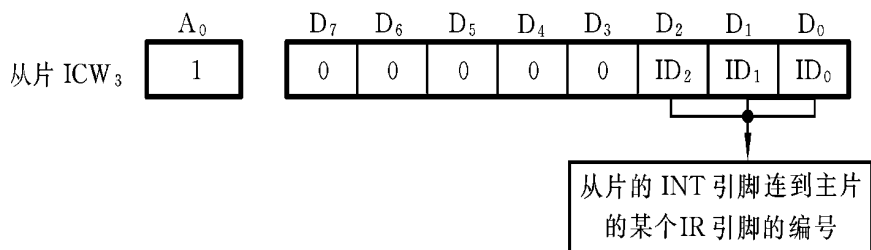
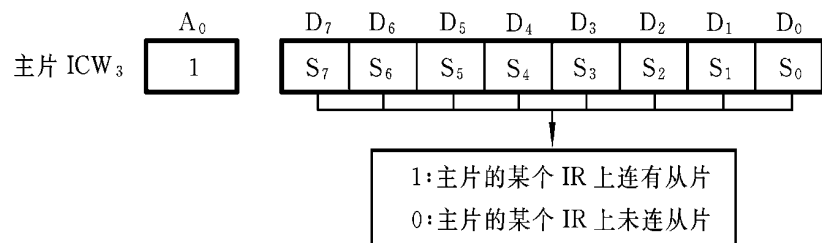
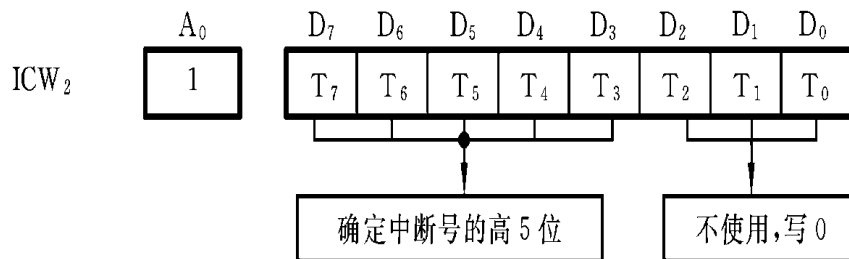
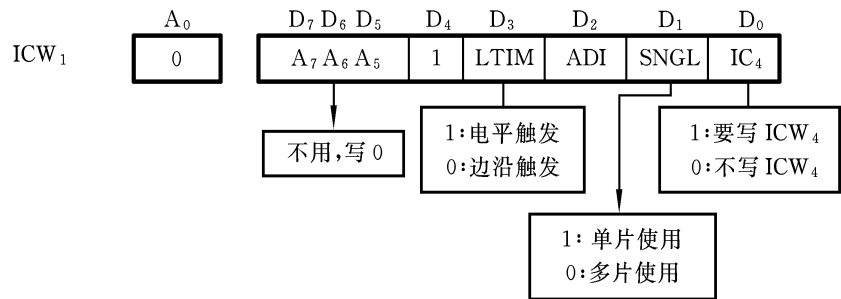
ICW4=01H；非缓冲，全嵌套，16位CPU，非自动结束

从片：

ICW1=11H, ICW2=70H, ICW3=02H, ICW4=01H

分❏析





■ 系统对中断控制器初始化程序段：

① 82C59A主片汇编语言初始化程序段如下

INTA00 EQU 020H	;82C59A主片端口(A0=0)
INTA01 EQU 021H	;82C59A主片端口(A0=1)
MOV AL,11H	;ICW1: 边沿触发, 多片, 要ICW4 (A0=0)
OUT INTA00,AL	
JMP SHORT\$+2	;I/O端口延时要求 (下同)
MOV AL,8H	;ICW2: 中断号的高5位 (A0=1)
OUT INTA01,AL	
JMP SHORT\$+2	
MOV AL,04H	;ICW3: 主片的IR2上接从片 (A0=1)
OUT INTA01,AL	
JMP SHORT\$+2	
MOV AL,01H	;ICW4: 非缓冲, 全嵌套, 16位的CPU, 非自动结束
OUT INTA01,AL	

② 82C59A从片汇编语言初始化程序段如下。

INTB00 EQU 0A0H	;82C59A从片端口(A0=0)
INTB01 EQU 0A1H	;82C59A从片端口(A0=1)
MOV AL,11H	;ICW1: 边沿触发, 多片, 要ICW4
OUT INTB00, AL	
JMP SHORT\$+2	
MOV AL,70H	;ICW2: 中断号的高5位
OUT INTB01, AL	
JMP SHORT\$+2	
MOV AL,02H	;ICW3: 从片接主片的IR2 (ID2ID1ID0=010)
OUT INTB01,AL	
JMP SHORT\$+2	
MOV AL,01H	;ICW4: 非缓冲, 全嵌套, 16位的CPU, 非自动结束
OUT INTB01,AL	

系统一旦完成了对82C59A的初始化, 所有外部硬件中断源和服务程序 (包括已开发和未开发的) 都必须按初始化的规定去做, 因此, 为慎重起见, 对系统的82C59A初始化编程不由用户去做, 而是在微机启动后由处理器按初始化设置要求自动完成。

11.9 用户对系统中断资源的使用

- 系统的82C59A初始化编程由BIOS自动完成
【不涉及ICWX的使用】
- 用户的工作是：
 - 修改中断向量
 - 发中断屏蔽/开放及中断结束命令
 - 编写中断服务程序

11.9.1 修改中断向量

1. 中断向量修改的方法

中断向量修改的方法是**利用DOS功能调用INT21H的35H号功能和25H号功能**。INT21H系统功能调用为用户程序修改中断向量提供了两个读/写中断向量的功能号。

获取中断向量表项

功能编号	AH = 35H
入口参数	AL = 向量号
出口参数	ES:BX = 中断向量
功能说明	从中断向量表中读取中断向量

设置中断向量表项

功能编号	AH = 25H
入口参数	AL = 向量号 DS:DX = 中断向量
功能说明	向中断向量表中写入中断向量

2. 中断向量修改与恢复的步骤

- ① 调用35H号功能，从向量表中读取某一中断号的原中断向量，并保存在双字节变量中。
- ② 调用25H号功能，将新中断向量写入中断向量表中原中断向量的位置，取代原中断向量。
- ③ 新中断服务程序完毕后，再用25H号功能将保存在双字节变量中的原中断向量写回去，恢复原中断向量。

1

```
CLI ; 修改中断向量时要关中断!  
MOV AH, 35H  
MOV AL, N  
INT 21H  
MOV OLD_SEG, ES ; 保存原中断向量  
MOV OLD_OFF, BX
```

2

```
MOV DX, SEG INTRnew ; 取新中断向量段基地址  
MOV DS, DX  
MOV DX, OFFSET INTnew ; 取新中断向量偏移地址  
MOV AL, N ; 中断类型为N  
MOV AH, 25H  
INT 21H
```

3

```
...  
MOV DS, OLD_SEG  
MOV DX, OLD_OFF  
MOV AL, N  
MOV AH, 25H  
INT 21H
```

11.9.2 发中断屏蔽/开放和中断结束指令

- 主程序中写 OCW_1 执行中断屏蔽与开放
- 中断服务程序中写 OCW_2 ，发中断结束信号EOI和使用IRET中断返回。

11.9.3 编写中断服务程序 ① 一般格式

```
NEW_INT PROC FAR
; 寄存器进栈      ; 保护现场
CLI                ; 关中断
; 服务程序主体
...
...
; 向主/从82C59A发中断结束命令
MOV AL, 20H
OUT 20H, AL
; 寄存器出栈      ; 恢复现场
...
STI                ; 开中断
; 中断返回
IRET
NEW_INT ENDP
```

② 编写中断服务程序需要注意的几个问题

- 注意现场的保护

在中断服务程序中**要使用的某些寄存器，要进栈保存**，以免破坏主程序所使用的这些寄存器内容。

- 注意堆栈操作的对称性

在堆栈中存放的中断返回地址及现场信息，要特别注意进栈与出栈的对称性，即**进栈与出栈的内容和顺序都要一一对应**，不要出现进栈与出栈不一致，以免发生中断结束后不能正确返回断点，造成严重后果。

- 中断服务程序要尽可能短

为避免对同级或低级的中断源造成阻塞和干扰，要求中断服务程序执行要快。因此，能在主程序中做的工作尽可能安排在主程序中进行，而不要放到中断服务程序中去。

另外要注意：

开放IRQ₂中断的方法：

IN AL, 21H ; 读IMR寄存器

AND AL, 0FBH ; 对应IRQ₂的位置0

OUT 21H, AL

关闭IRQ₂中断的方法：

IN AL, 21H ; 读IMR寄存器

OR AL, 04H ; 对应IRQ₂的位置1

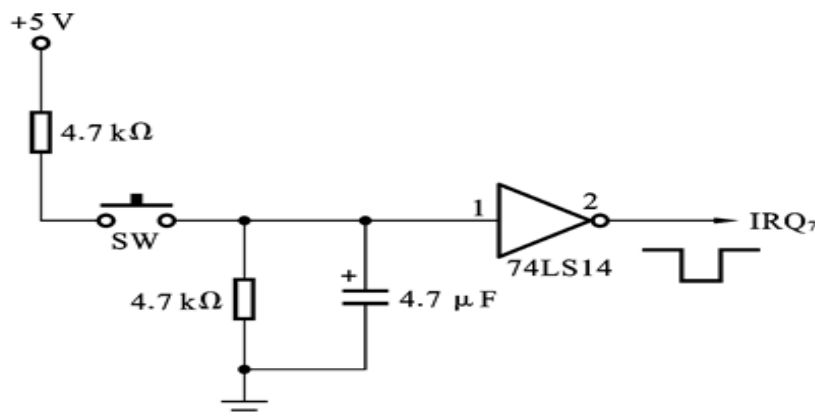
OUT 21H, AL

11.10 中断服务程序设计

例1 利用82C59A主片设计中断服务程序

要求：电路如下图所示。微动开关SW中断请求连接到IRQ₇。每按下一次申请1次中断。按8次后显示“OK!”，程序结束。

分析：IRQ₇的中断号为0FH，系统分配给打印机中断。当打印机空闲不用时，可以通过修改中断向量加以利用。



程序设计

; 堆栈段定义

STACK SEGMENT

DW 200 DUP(?)

STACK ENDS

; 数据段定义

DATA SEGMENT

OLD_IV DD ? ; 保存原中断向量的双字单元

MK_BUF DB ? ; 保存原屏蔽字的字节单元

BUF DB 'OK!',ODH,OAH,\$; 提示符

DATA ENDS

CODE SEGMENT

ASSUME CS:CODE, DS:DATA, SS:STACL

; 发中断开放指令

START: MOV AX,DATA

MOV DS,AX

IN AL,21H ; 保存原屏蔽字

MOV MK_BUF,AL

CLI ; 关中断

AND AL,01111111B(7FH) ; 开放0FH号中断(OCW1)

OUT 21H,AL

; 修改中断向量

CALL GET_IV ; 获取原中断向量, 并保存

CALL SET_IV ; 设置用户程序新中断向量

; 主程序主体

XOR DX,DX ; 清空计数器

L1: STI ; 开中断

CMP DX,8 ; 计数是否到8

JNZ L1 ; 未到, 继续等待微动开关中断

; 中断8次结束, 恢复原来的中断向量和屏蔽字

CLI ; 已到, 关中断

CALL RENEW_IV ; 恢复原中断向量

MOV AL,MK_BUF ; 恢复原屏蔽字(OCW1)

OUT 21H,AL

STI ; 开中断

; 中断结束, 显示 “OK!”

MOV AX, SEG BUF

MOV DS, AX

MOV DX, OFFSET BUF

MOV AH,09H

INT 21H

; 返回DOS

MOV AX,4C00H

INT 21H

; 中断服务程序SW_INT

SW_INT PROC FAR

PUSH AX

CLI ; 关中断

INC DX ; 计数加1

MOV AL,67H ; 发中断结束命令 (OCW2)

OUT 20H,AL

POP AX

STI ; 开中断

IRET ; 中断返回

SW_INT ENDP

; 获取原中断向量子程序

GET_IV PROC NEAR

MOV AX,350FH

INT 21H

MOV WORD PTR OLD_IV, BX

MOV BX, ES

MOV WORD PTR OLD_IV+2,BX

RET

GET_IV ENDP

; 设置新中断向量子程序

SET_IV PROC NEAR

PUSH DS

MOV AX,SEG SW_INT

MOV DS,AX

MOV DX,OFFSET SW_INT

MOV AX,250FH

INT 21H

POP DS

RET

SET_IV ENDP

; 恢复原来的中断向量子程序

RENEW_IV PROC NEAR

PUSH DS

MOV DX,WORD PTR OLD_IV

MOV AX,WORD PTR OLD_IV+2

MOV DS,AX

MOV AX,250FH

INT 21H

POP DS

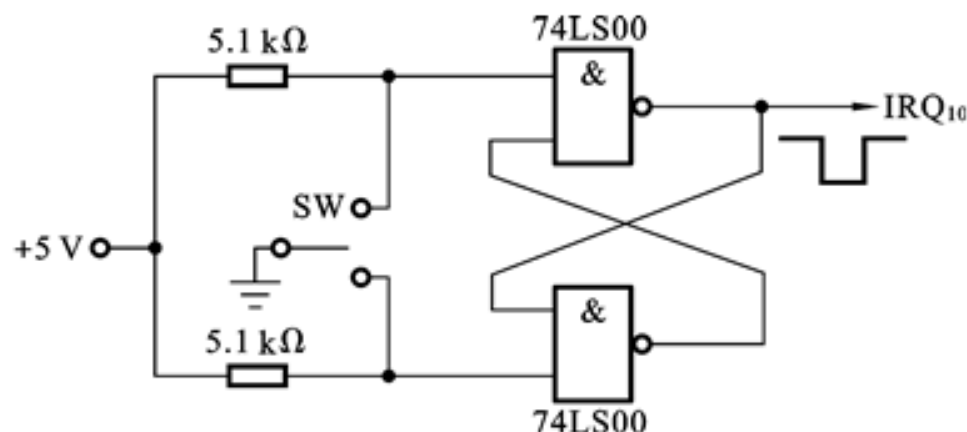
RET

RENEW_IV ENDP

例2 利用82C59A从片设计中断服务程序

要求：电路如图所示。微动开关SW中断请求连接到IRQ₁₀。每按下一次申请1次中断,显示 “OK!”,程序结束

分析：IRQ₁₀是从片的IR2引脚上的中断请求，中断号为72H，是系统保留的，用户可以使用。由于是从片，**在执行屏蔽开放和中断结束命令时，要考虑主片的相应操作。**



程序设计

； 数据段定义

DATA SEGMENT

INT_OFF DW ?

INT_SEG DW ?

BUF DB 'OK!',ODH,OAH,\$

DATA ENDS

； 段地址定义STACK

SEGMENT

DW 200 DUP(?)

STACK ENDS

START: MOV AX, DATA

MOV DS, AX

MOV ES, AX

MOV AX, STACK

MOV SS, AX

; 获取原中断向量

MOV AX,3572H

INT 21H

MOV INT_OFF,BX

MOV BX,ES

MOV INT_SEG,BX

; 装入新中断向量

CLI

MOV DX,SEG SW_INT

MOV DS,DX

MOV DX,OFFSET SW_INT

MOV AX,2572H

INT 21H

MOV AX,DATA

MOV DS,AX

; 开放主片IRQ₂的中断屏蔽

IN AL,21H

AND AL,11111011B(FBH)

OUT 21H,AL

; 开放从片IQR₂的中断屏蔽

IN AL,0A1H

AND AL,11111011B(FBH)

OUT 0A1H,AL

; 主程序

STI

HLT

; 中断结束恢复原中断向量

CLI

MOV DX,INT_SEG

MOV DS,DX

MOV DX,INT_OFF

MOV AX,2572H

INT 21H

MOV AX,DATA

MOV DS,AX

; 中断结束屏蔽主片IRQ₂

IN AL,21H

OR AL,00000100B(04H)

OUT 21H,AL

; 中断结束屏蔽从片IQR₂

IN AL,0A1H

OR AL,00000100B(04H)

OUT 0A1H,AL

STI

; 中断结束, 显示 “OK!”

MOV AX, SEG BUF

MOV DS, AX

MOV DX, OFFSET BUF

MOV AH, 09H

INT 21H

; 返回DOS

MOV AX, 4C00H

INT 21H

; 中断服务程序SW_INT

SW_INT PROC FAR

PUSH AX

CLI

MOV AL, 20H ; 主片发EOI命令

OUT 20H, AL

MOV AL, 62H ; 从片发EOI命令

OUT 0A0H, AL

STI

POP AX

IRET

SW_INT ENDP

9.2 8259应用举例

微机系统

例：在一个微机系统中，要求采用中断方式将一个长**100**字节的数据块从首址为BUF的存储区传送到端口地址为**37FH**的接口缓冲区中，每次中断只传送一个字节。设该中断的类型码为**0AH**，试写出其主程序和中断服务子程序。具体要求：主程序需要完成寄存器的初始化，中断向量修改和恢复，中断的屏蔽与开放；而中断服务子程序则需要完成数据的发送（中断采用非自动结束方式，8259A的片选地址为20H）。

- 利用微机已经定义的中断号完成新的中断服务，需要将原中断信息保留，设置新的中断向量，完成新的中断处理后再恢复既有的中断信息。
- DOS功能调用INT 21H的35号和25号功能存取中断向量分三步：
 - ✓ 用35H号功能，获取原中断向量。
 - ✓ 用25H号功能，设置新中断向量，取代原中断向量。
 - ✓ 新服务程序执行完毕后，利用25H号功能恢复原中断向量。

9.2 8259应用举例

微机系统

INT 21H的35号和25号功能

中断号	功 能	调 用 参 数
21H	从 中 断 向 量 表 读 取 中 断 向 量	入口参数: AH=35H, AL=中断号 出口参数: ES:BX=读取向量段 基址: 偏移量
21H	向 中 断 向 量 表 写 入 中 断 向 量	入口参数: AH=25H, AL=中断号 出口参数: DS:DX=写入向量段 基址: 偏移量

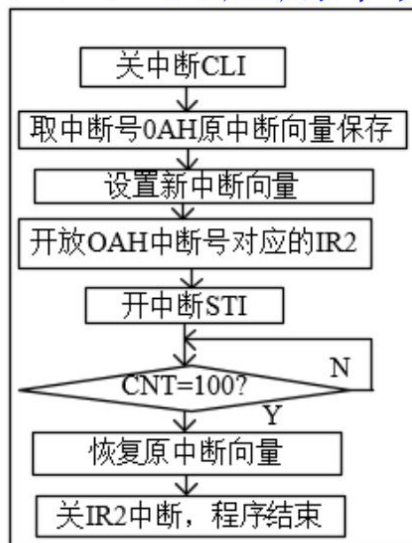
```
MOV AL,0AH           ;取0AH号原中断向量并保存
MOV AH,35H
INT 21H
MOV OLDSEG, ES
MOV OLDOFF, BX
```

```
PUSH DS               ;设置新中断向量
MOV AX,SEG SENDER     ;取当前中断程序入口段基址
MOV DS,AX
MOV DX,OFFSET SENDER  ;取当前中断程序入口偏移量
MOV AL,0AH            ;中断类型号为0AH
MOV AH,25H
INT 21H               ;将当前中断入口地址送向量表
POP DS
```

```
MOV DS,OLDSEG        ;发完数据后恢复原中断向量
MOV DX, OLDOFF
MOV AL,0AH
MOV AH,25H
INT 21H
```

9.2 8259应用举例

主程序



中断程序



主程序



1	BUF DB 100 DUP (?)	;数据块
2	CNT DB 0	;已传个数
3	OLDOFF DW ?	;旧中断向量 偏移地址
4	OLDSEG DW ?	;旧中断向量 的段基址
5	START PROC FAR	;主程序
6	MOV AX,DATA	
7	MOV DS,AX	
8	CLI	
9		;取0AH号原中断向量
10		;设置新中断向量
11	IN AL,21H	;8259开中断
12	AND AL, 0FBH	;1111 1011B 开放IR2
13	OUT 21H, AL	
14	STI	
15	WAT :	;等待
16	CMP CNT,100	
17	JNZ WAT	
18		;恢复原中断向量
19	IN AL,21H	;8259关IR2
20	OR AL, 04H	
21	OUT 21H, AL	
22	RET	
23	START ENDP	

中断程序



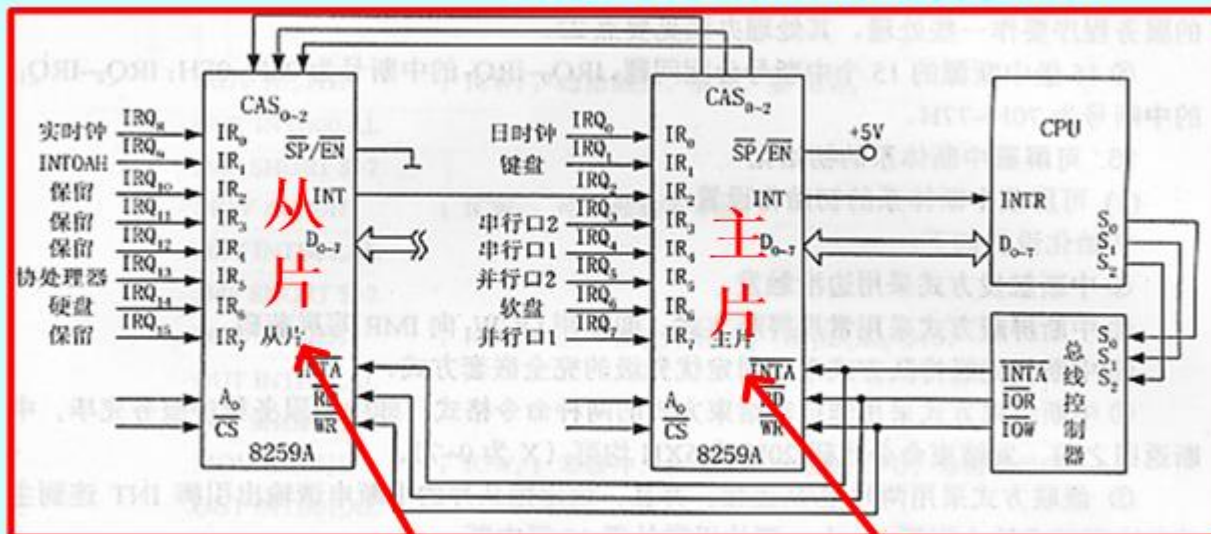
微机系统

24	SENDER: PROC FAR	;中断程序
25	PUSH AX	
26	PUSH DX	
27	PUSH SI	
28	MOV SI,OFFSET BUF	;SI指向首址
29	ADD SI,CNT	
30	MOV AL,[SI]	
31	MOV DX,37FH	
32	OUT DX,AL	
33	INC CNT	
34	MOV AL,20H	
35	OUT 20,AL	
36	POP SI	
37	POP DX	
38	POP AX	
39	IRET	

TPC实验平台-中断模块

- 386CPU内部有2个8259，级联采用与PC机相同的方式，其主从8259使用的命令字和端口地址与PC DOS兼容。
- 本实验箱允许使用MIRQ3(对应中断号0BH)上和SIRQ10（对应中断号72H）
- 注意，主从8259已经初始化，并且固化，用户无需对8259初始化，只能写入中断屏蔽字和中断向量。

386 CPU内部主从8259中断



SIRQ10

MIRQ3

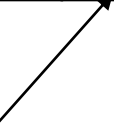
主从片8259A的端口地址：

用户中断输入引脚	对应8259引脚输入	中断屏蔽字	中断类型号	8259端口地址
MIRQx (MIRQ3)	主片IR3	11110111B	0BH	20H,21H
SIRQx (SIRQ10)	从片IR2	11111011B	72H	0A0H, 0A1H

实验平台主从8259A的中断类型号与中断源的对应关系

中断源	IR0	IR1	IR2	IR3	IR4	IR5	IR6	IR7
主片中断向量	08H	09H	0AH	0BH	0CH	0DH	0EH	0FH
从片中断向量	70H	71H	72H	73H	74H	75H	76H	77H

中断号基数



TPC实验平台 - 注意几点

- 1、不需要对8259A初始化（ICW1-ICW4），但要设置中断向量和中断屏蔽字；
- 2、TPC采用的是非自动结束，则需要在中断结束前（中断服务程序的最后）发中断结束命令：

MOV AL,20H

OUT 20H,AL

- 3、写入屏蔽信息需要采取读取 - 修改 - 写入顺序：

主8259: IN AL,21H

AND AL,0F7H(11110111B)

OUT 21H,AL

4、修改中断向量的方法（不支持DOS的35H和25H功能）

```
WRINTVER PROC NEAR
    PUSH DS
    MOV AX, 0
    MOV DS, AX
    MOV DI, 4*0BH      ; 中断向量地址(TPC=0BH)
    CLI
; 置新中断向量偏移地址
    MOV BX, OFFSET INTR
    MOV [DI], BX
    ADD DI, 2
; 置新中断向量段基地址
    MOV BX, SEG INTR
    MOV [DI], BX
    POP DS
    ret
WRINTVER ENDP
```