

第九章 中断

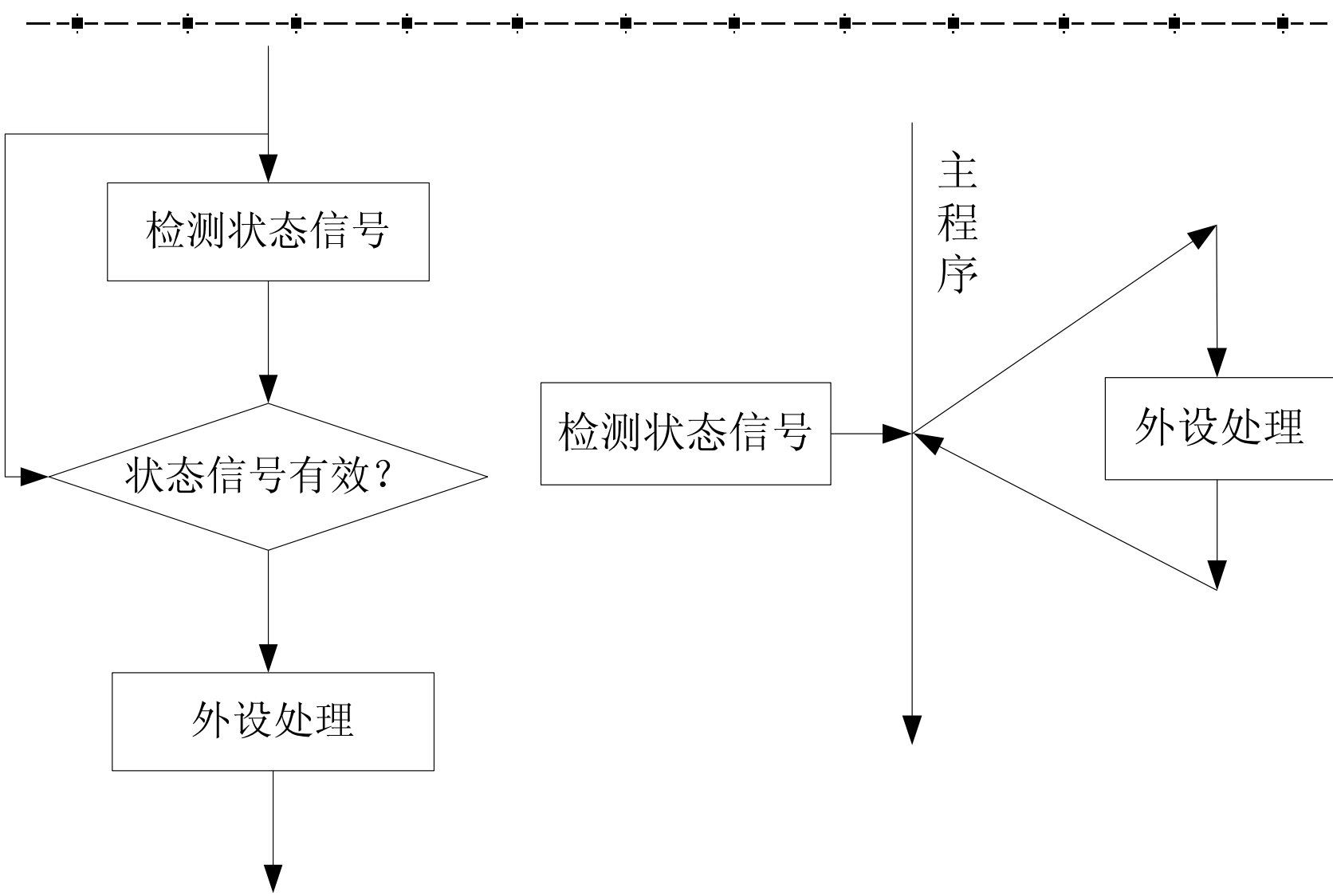
当CPU用查询的方式与外设交换信息时，CPU就要浪费很多时间去等待外设。这样就引出一个快速的CPU与慢速的外设之间数据传送的矛盾，这也是计算机在发展过程中遇到的严重问题之一。为了解决这个问题，一方面要提高外设的工作速度，另一方面发展了中断概念。中断系统是计算机的重要指标之一。

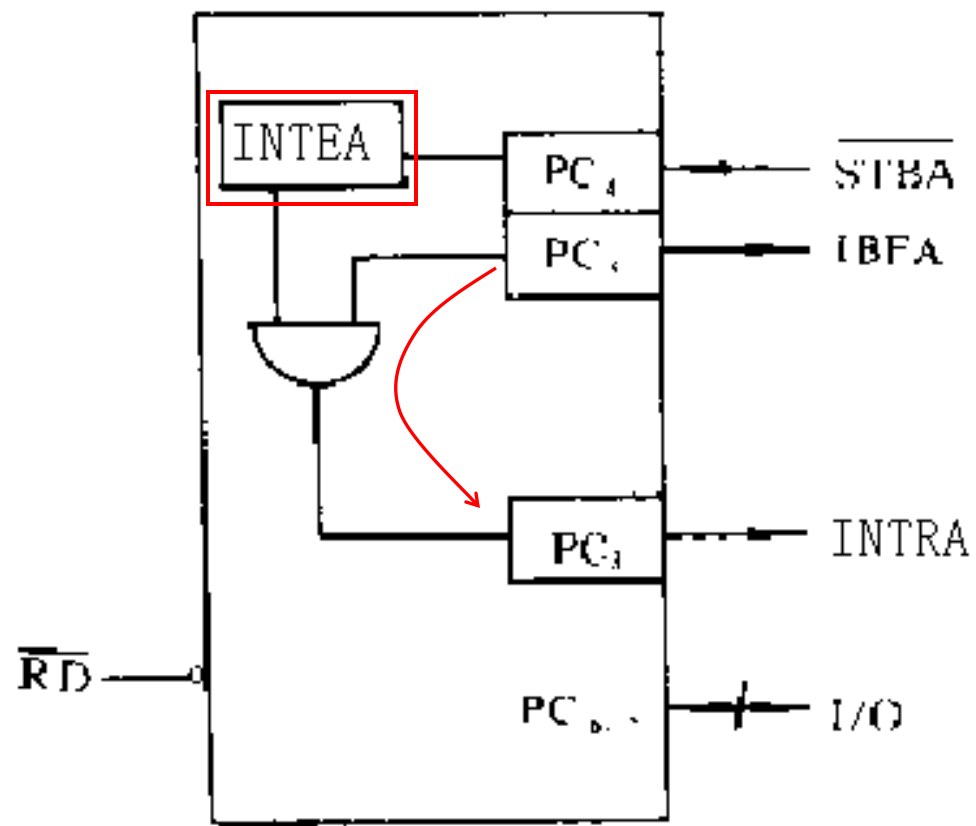
9. 1 中断原理

✦ 9. 1. 1 从无条件传送、条件传送到中断传送

条件传送最大的缺点就是为了CPU和外设在时间上配合正确，CPU花大量的时间用无条件方式对状态线进行查询，从而降低了整个系统的工作效率。

具有中断功能的CPU中，有一个硬件部件专门用于检测外设的状态线。





A 端口

9. 1 中断原理

✦ 9. 1. 2 中断概念

处理文档	→	执行主程序(日常事务程序)
电话铃响	→	中断申请信号有效(中断请求)
暂停文档	→	暂停执行主程序响应中断
文档中作暂停记号	→	当前PC及寄存器入栈(保护现场)
电话交谈	→	处理外设要求(中断服务)
找出暂停记号位置	→	寄存器及PC出栈(恢复现场及中断返回)
继续处理文档	→	继续执行主程序

9. 1 中断原理

✦ 9. 1. 2 中断概念

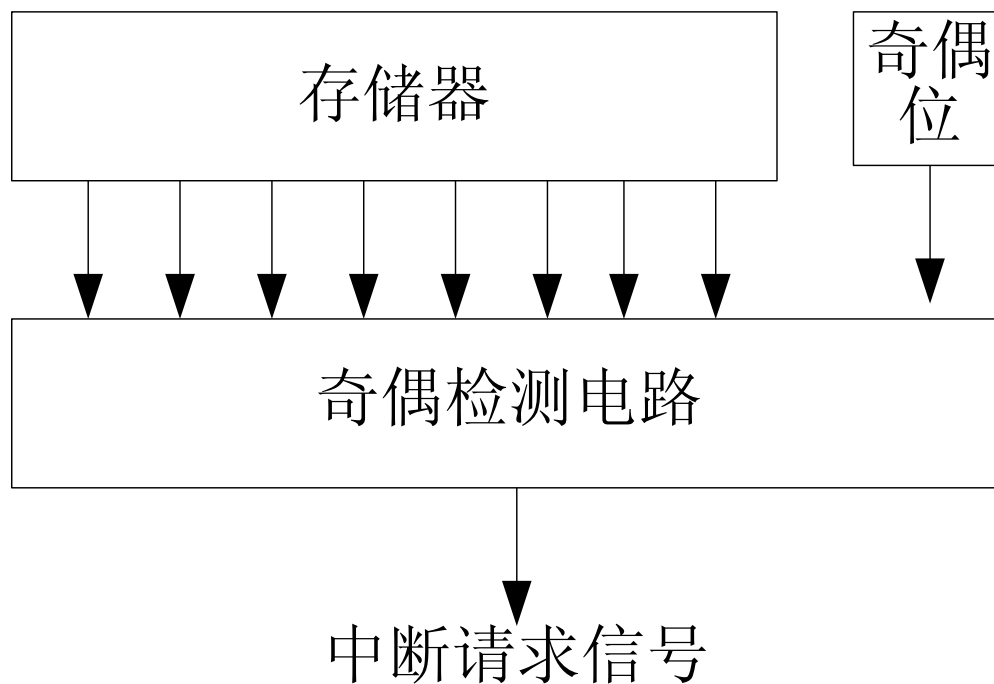
- ✦ 可见在中断传送方式下，外设应有请求CPU服务的权利，当外部设备准备好向CPU传送数据，或者外设已准备就绪接收CPU的数据，或者有某些紧急情况要求处理，或者是定时时间到等等。这时，**外设向CPU发出中断请求，CPU接收到请求并在一定条件下，暂时停止执行原来的程序而转去中断处理，处理好中断服务再返回来执行原来程序，这就是中断概念。**

9. 1 中断原理

✦ 9. 1. 3 中断应用

◆ 一、实时故障处理

存储器出错检测电路

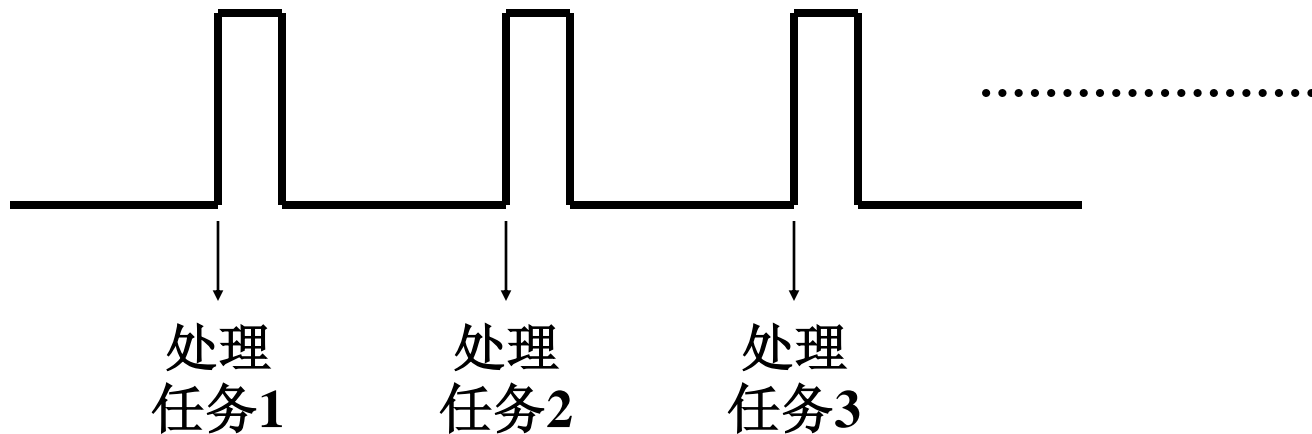


9. 1 中断原理

✦ 9. 1. 3 中断应用

◆ 二、分时操作，同时处理

有了中断功能，CPU可命令多个外设同时工作。虽然CPU在不同的时间点上为不同的任务工作，但宏观上看CPU几乎同时为不同的任务工作，极大地发挥了CPU高速性的特点。



9. 2 中断系统组成及其功能

✦ 9. 2. 1 三个与中断有关的触发器

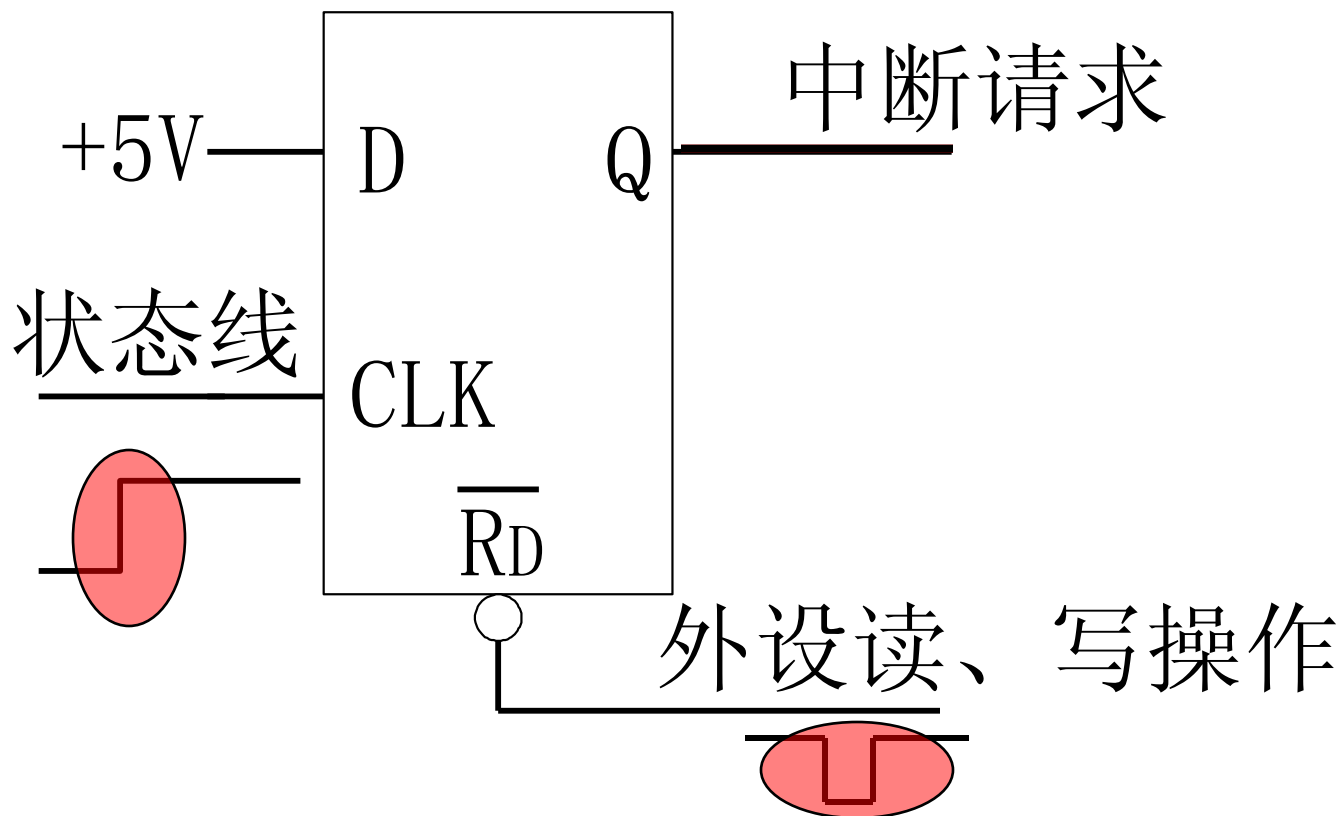
◆ 一、中断请求触发器

- 应该有两个特点：

- ◆ 1、它的输出可以作为中断请求信号，在满足一定条件的情况下把信号发送给CPU，并在CPU未响应时一直保存下去；
- ◆ 2、当CPU满足一定条件下响应了该中断请求信号，执行了相关的操作后，该中断请求信号可以被撤除。

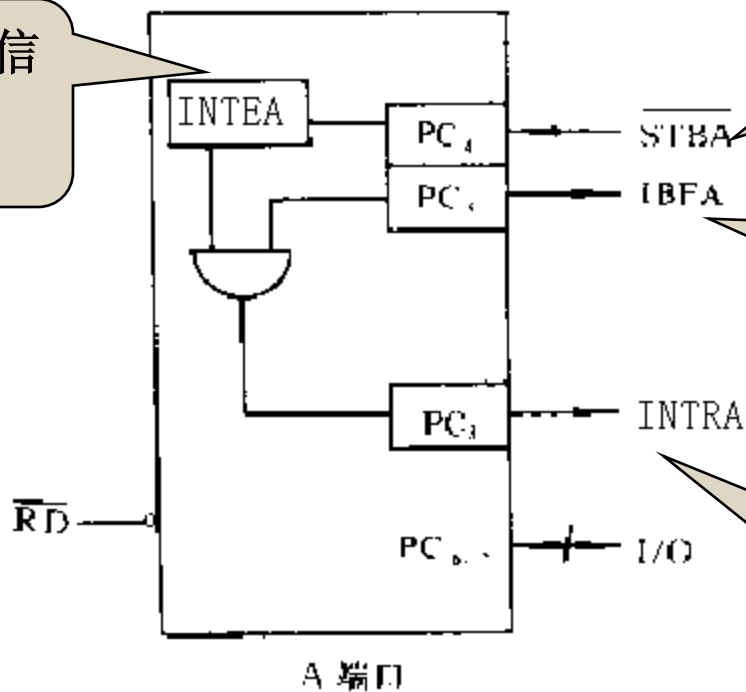
9. 2 中断系统组成及其功能

✦ 9. 2. 1 三个与中断有关的触发器



★ 8255A工作在方式1——输入

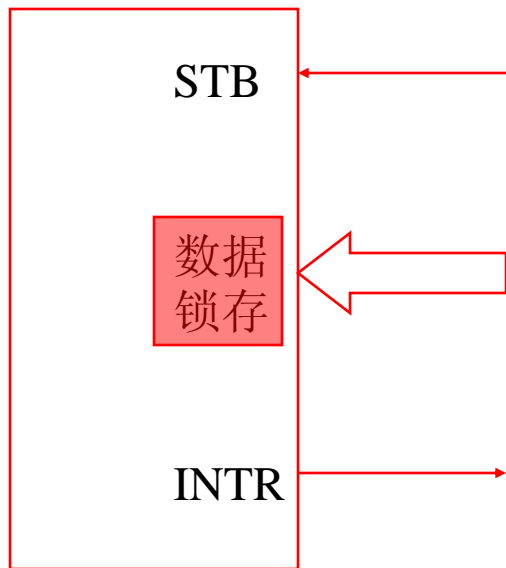
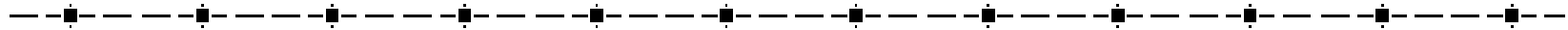
中断允许信号。

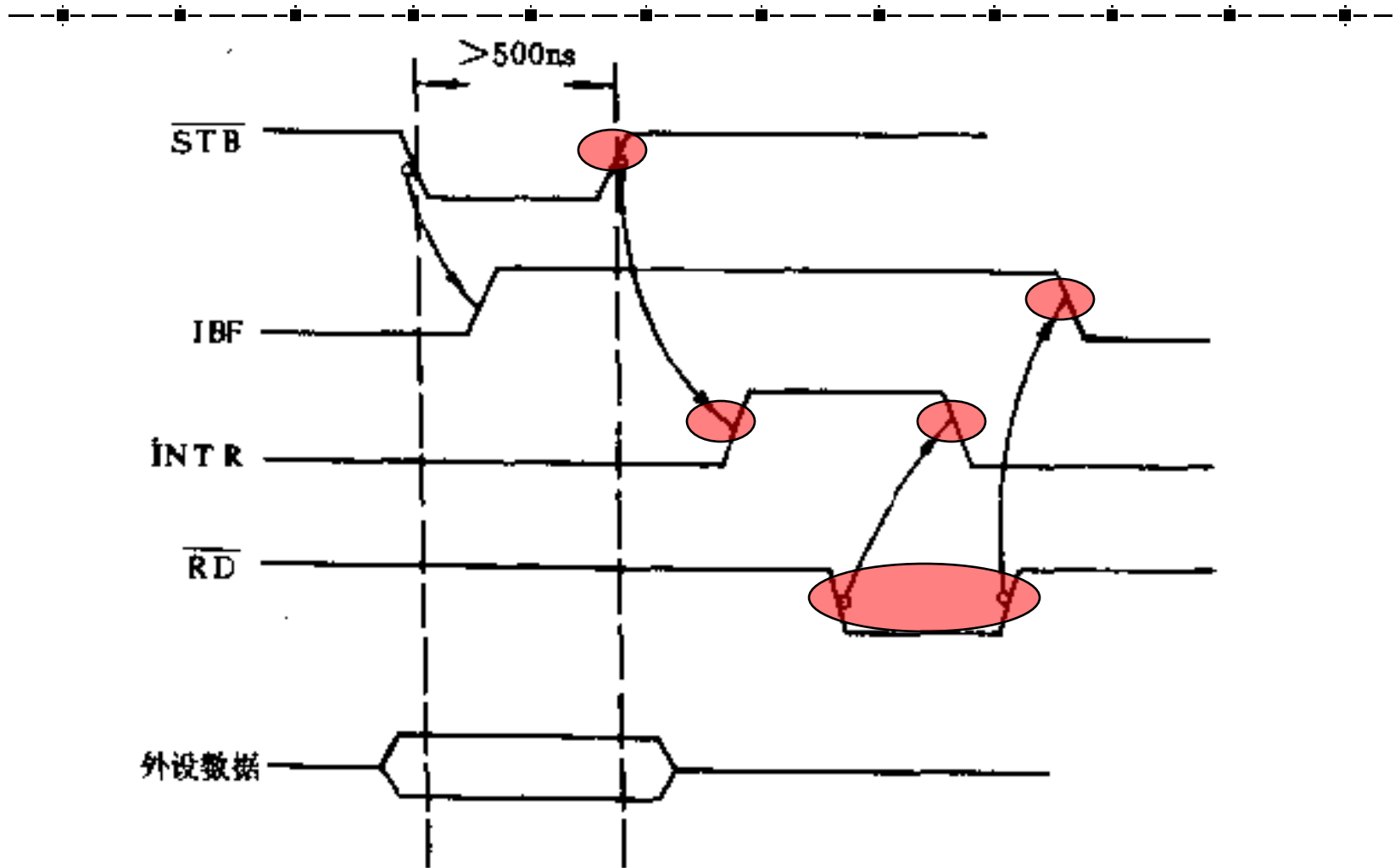


A口的选通信号，当其有效时，外设把数据打入A口的输入缓冲器

A口的输入缓冲器“满”信号，当其有效时表示A口的输入缓冲器已暂存一个有效数据。

A口的中断请求信号。当其有效时，8255A的A口向CPU申请中断，要求CPU从A口取数



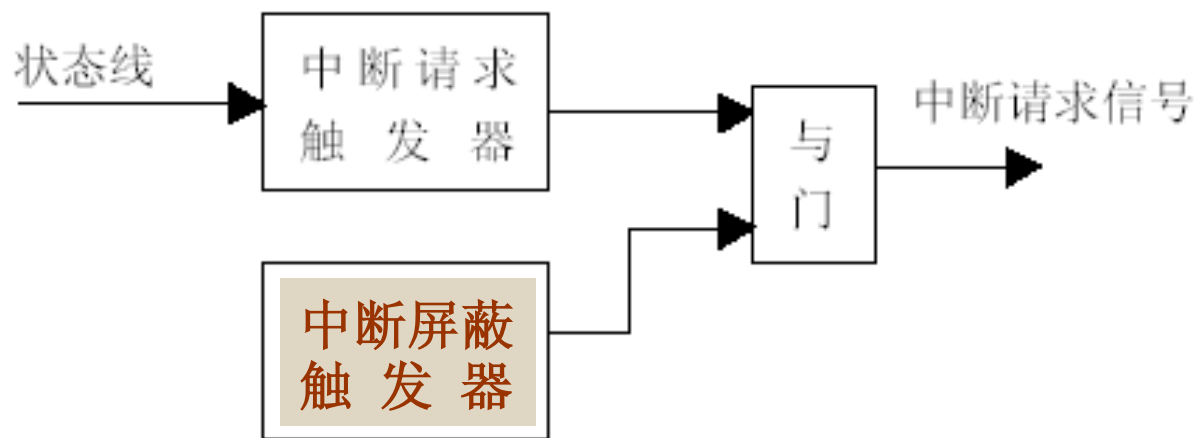


9. 2 中断系统组成及其功能

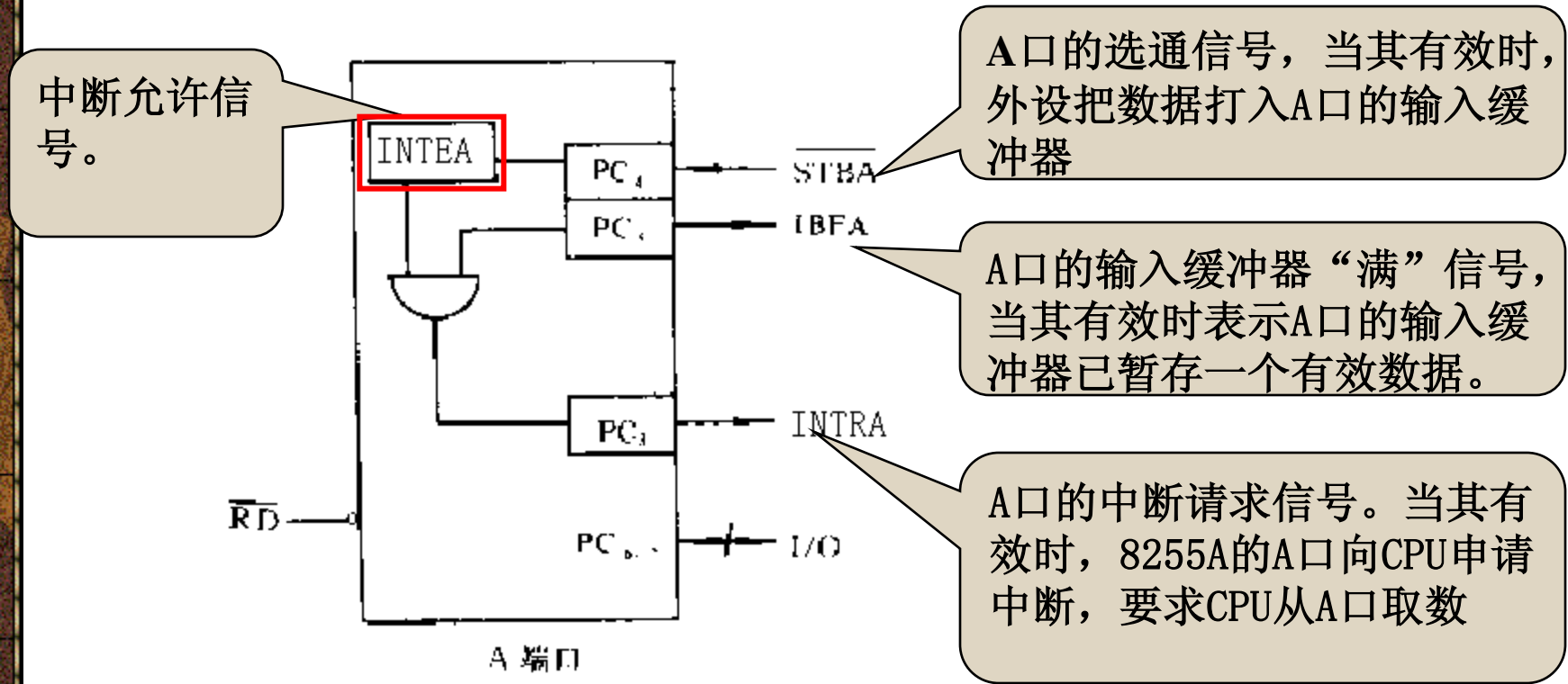
★ 9. 2. 1 三个与中断有关的触发器

◆ 二、中断屏蔽触发器

- 中断屏蔽触发器的功能就是决定中断请求触发器的输出信号是否可以作为中断请求信号发送给CPU，这样CPU通过对中断屏蔽触发器的设置就可以达到对中断源的控制。



★ 8255A工作在方式1——输入



MOV AL,00001001 ;允许发中断

MOV 控制口, AL

MOV AL,00001000 ;屏蔽中断

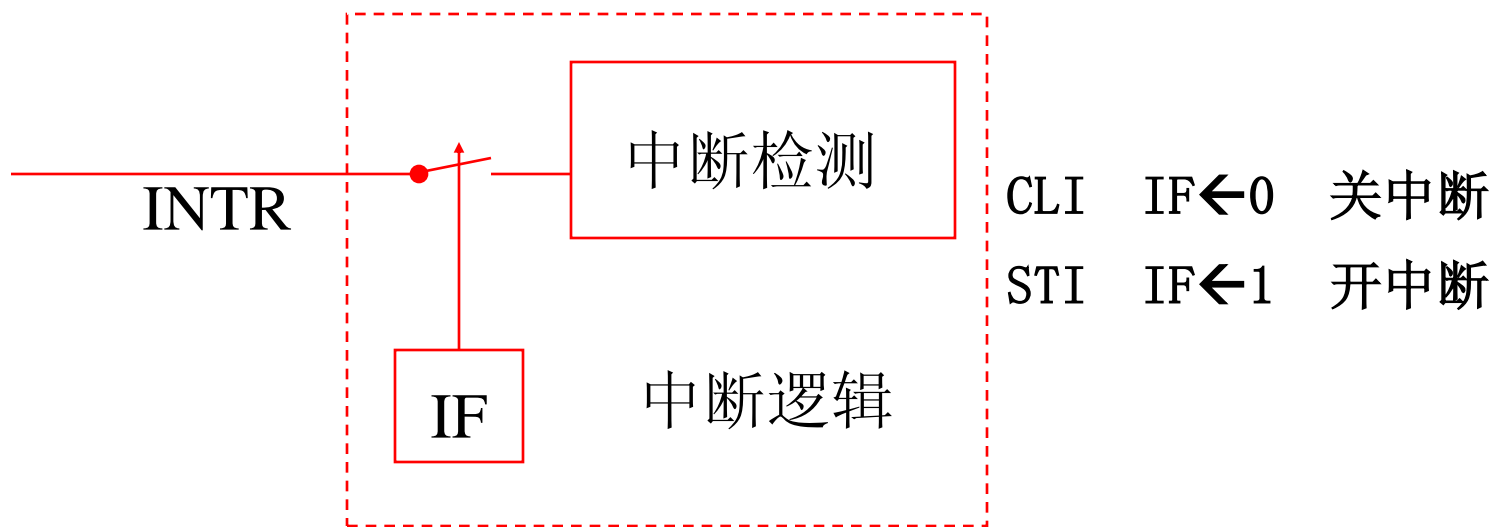
MOV 控制口, AL

9. 2 中断系统组成及其功能

✦ 9. 2. 1 三个与中断有关的触发器

◆ CPU内部的中断允许触发器IF

- CPU通过对它进行设置来决定是否对发给它的中断请求信号进行响应。

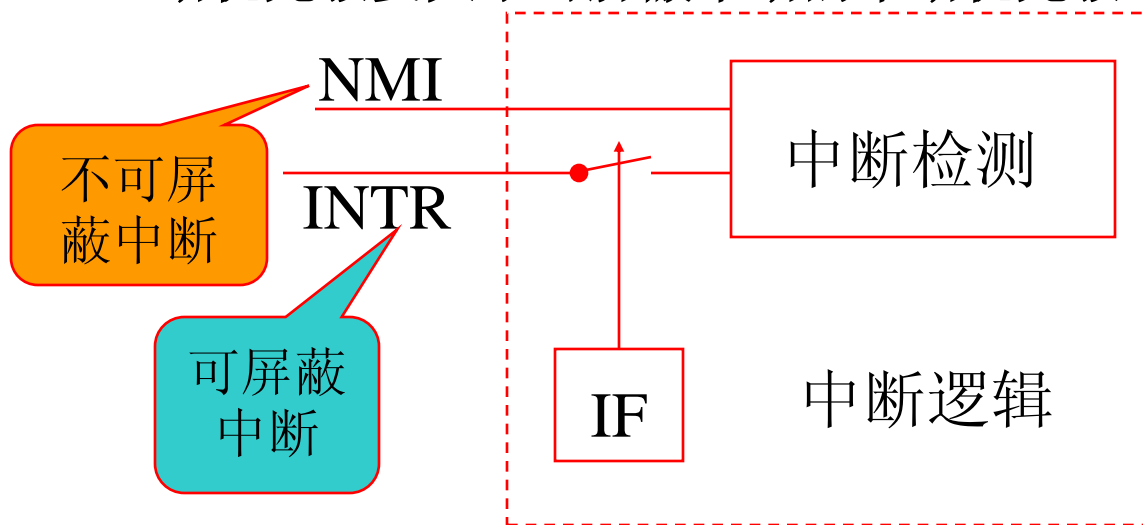


9. 2 中断系统组成及其功能

★ 9. 2. 1 三个与中断有关的触发器

◆ CPU内部的中断允许触发器IF

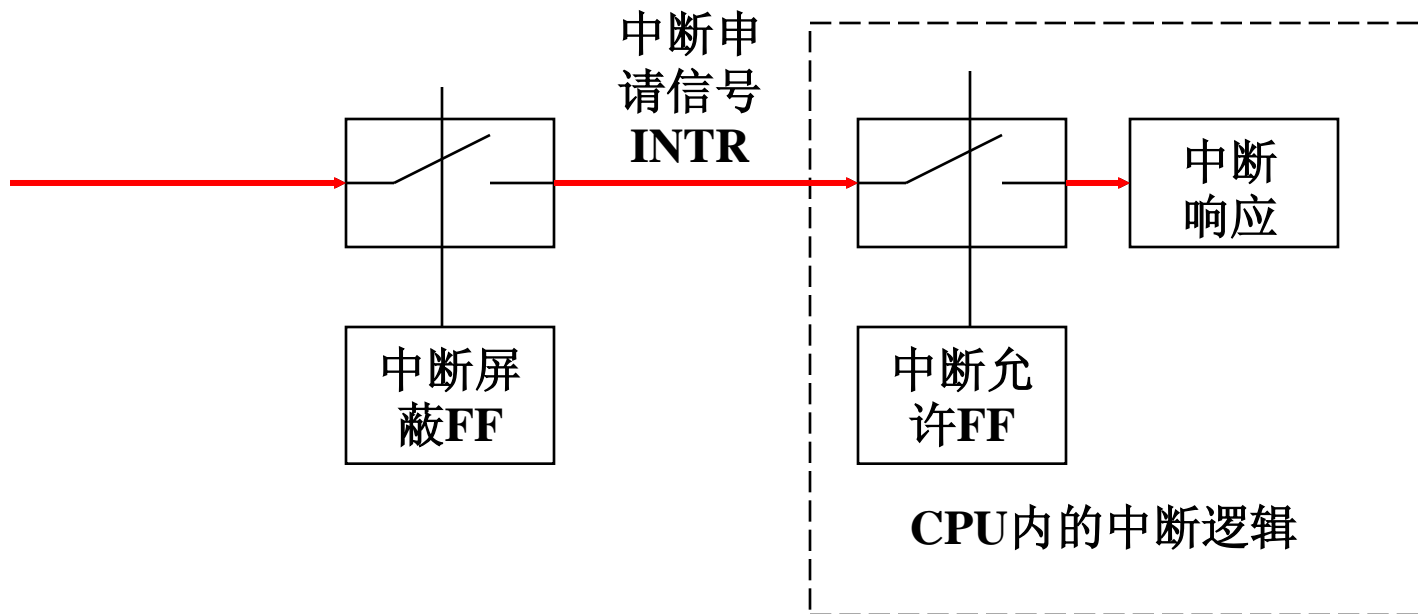
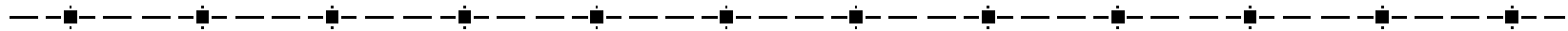
- 有些CPU，比如8086CPU，设置了两种中断类型：**可屏蔽中断**和**不可屏蔽中断**。可屏蔽中断受中断允许触发器控制，只有当IF为1时，CPU才能响应中断请求信号。而不可屏蔽中断不受中断允许触发器的控制，只要中断请求信号有效，不管IF是否为1，CPU就必须响应。因此不可屏蔽中断的中断优先级要大于可屏蔽中断的中断优先级。



9. 2 中断系统组成及其功能

✦ 9. 2. 2 再谈中断条件

- ✦ 外设的中断请求信号要想发给CPU并能最终得到CPU的响应，必须要满足如下两个条件：一个是中断屏蔽触发器处于非屏蔽状态。在这种情况下，中断请求信号才能发给CPU。但CPU是否相应这个中断，还要看中断允许触发器是否处于开中断状态。只有CPU是开中断的条件下，CPU才能进入中断响应过程，处理中断事务。这就是第二个条件。



9. 2 中断系统组成及其功能

✦ 9. 2. 3 中断响应过程

◆ 中断过程主要包括三个方面

- 外设发中断请求信号给CPU即中断请求
- CPU对中断请求信号所作出的反应即中断响应
- CPU执行对外设操作的子程序即中断处理。

9. 2 中断系统组成及其功能

✦ 9. 2. 3 中断响应过程

◆ 一、中断申请

- 当中断屏蔽触发器状态为1，则中断请求触发器输出的中断请求信号发给CPU。

◆ 二、中断响应

- CPU响应可屏蔽中断申请必须满足的3个条件：
 - ① 无总线请求；
 - ② CPU被允许中断；
 - ③ CPU执行完现行指令。

9. 2 中断系统组成及其功能

✦ 9. 2. 3 中断响应过程

✦ 三、中断处理

- CPU响应中断后要自动完成三项任务
 - 1、关闭中断；
 - 2、CS、IP以及FR的内容推入堆栈；
 - 3、中断服务程序段地址送CS中, 偏移地址送IP中。
- 一旦CPU响应中断, 就可转入中断服务程序之中。中断服务程序的结构如下:

push ax ; 1、保护现场

.....

push bx

sti ; 2、开中断

.....

; 3、中断处理

cli ; 4、关中断

pop bx ; 5、恢复现场

.....

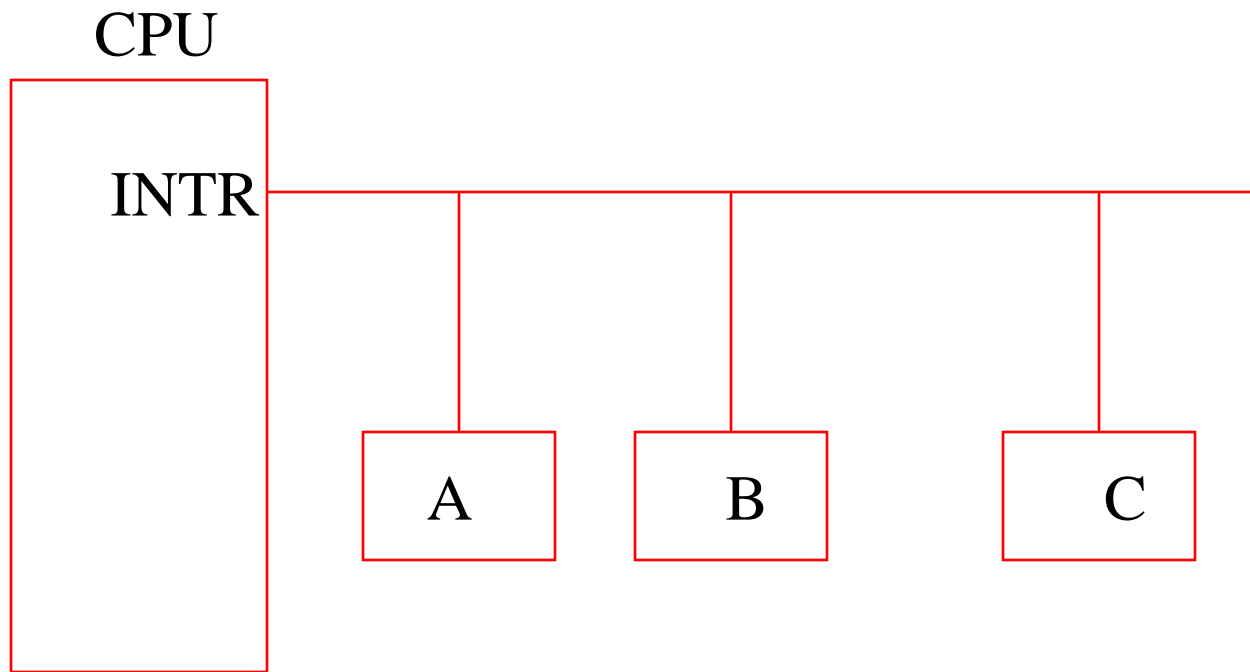
pop ax

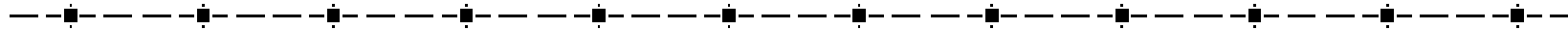
sti ; 6、开中断和中断返回

reti

9. 3 中断源识别及中断优先权

在中断系统中一个非常关键的问题是CPU如何知道是哪一个中断源发出的中断申请信号。只有正确地确定中断源，CPU才能转到相应的中断服务程序为之服务。这里，确定中断源的方法被称为中断源识别或中断方式。





?!!!

9. 3 中断源识别及中断优先权

✦ 9. 3. 1 中断源识别

包括两个方面：

- 1、确定中断源，

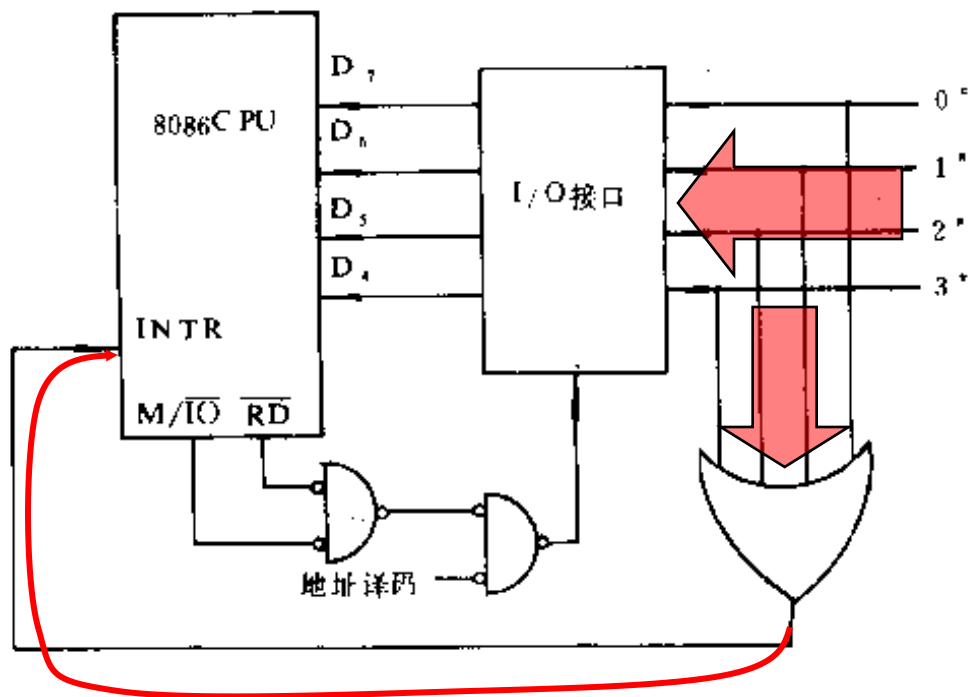
- 2、找到该中断服务程序的首地址。

下面我们给出解决问题的两种方案。

9. 3 中断源识别及中断优先权

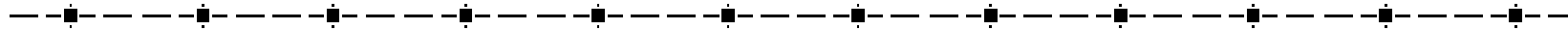
✦ 9. 3. 1 中断源识别

◆ 一、查询中断



IN AL, IPORT	;	从输入接口取中断信息
TEST AL, 80H	;	是0号设备请求吗?
JNZ SEVO	;	是, 转0号设备服务程序
TEST AL, 40H	;	否, 是1号设备请求吗?
JNZ SEV1	;	是, 转1号设备服务程序
TEST AL, 20H	;	否, 是2号设备请求吗?
JNZ SEV2	;	是, 转2号设备服务程序
TEST AL, 10H	;	否, 是3号设备请求吗?
JNZ SEV3	;	是, 转3号设备服务程序

条件传送=查询中断?



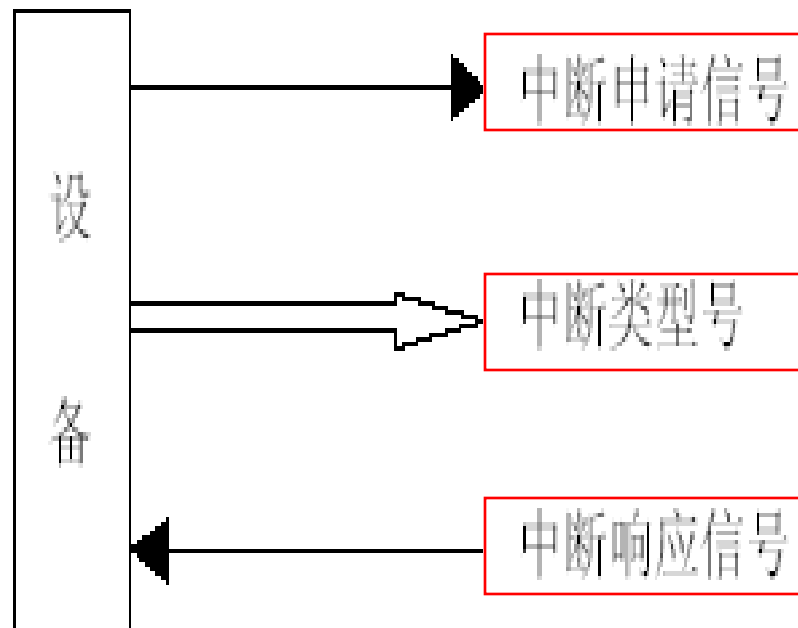
特点：简单
响应慢

9. 3 中断源识别及中断优先级

★ 9. 3. 1 中断源识别

◆ 二、矢量中断

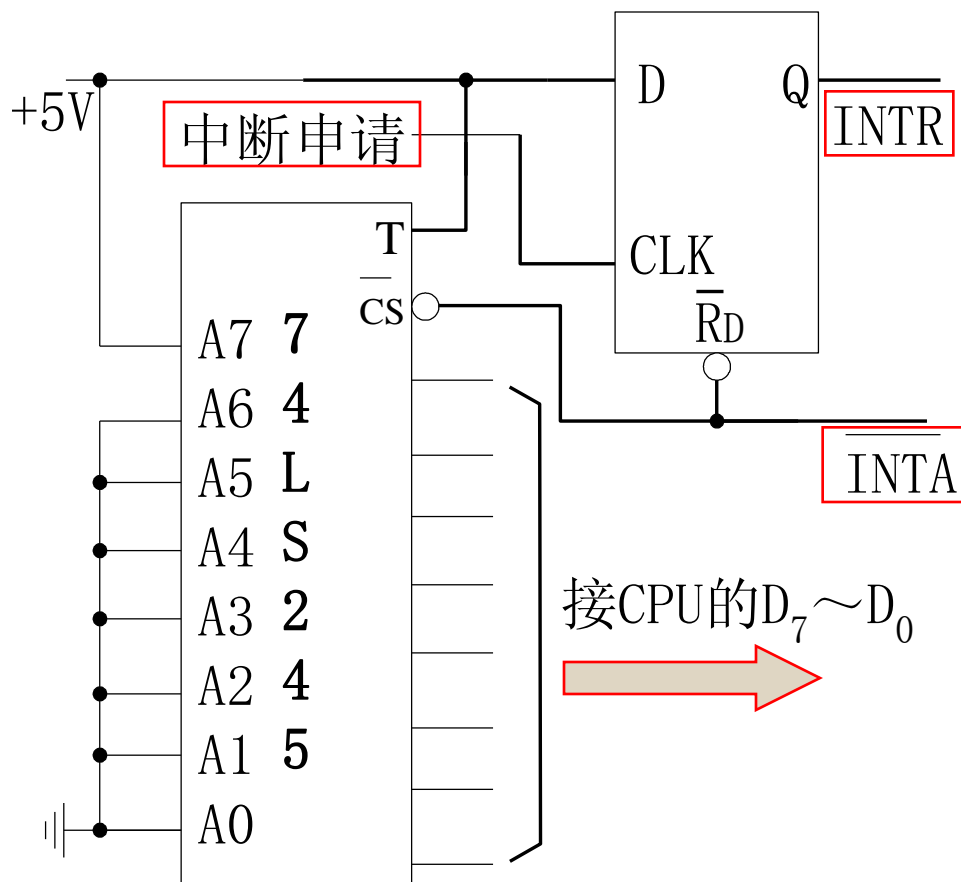
中断申请信号 **INTR** 和中断响应信号 **INTA** 是一对握手信号。在驱动一个中断事件过程中，中断请求信号是外设发给CPU的，当其有效时，表示外设请求CPU为之服务。而中断响应信号是CPU发给外设的，当其有效时，表明CPU可以为这个外设服务，同时要求外设提供**中断类型号**。



9. 3 中断源识别及中断优先级

✦ 9. 3. 1 中断源识别

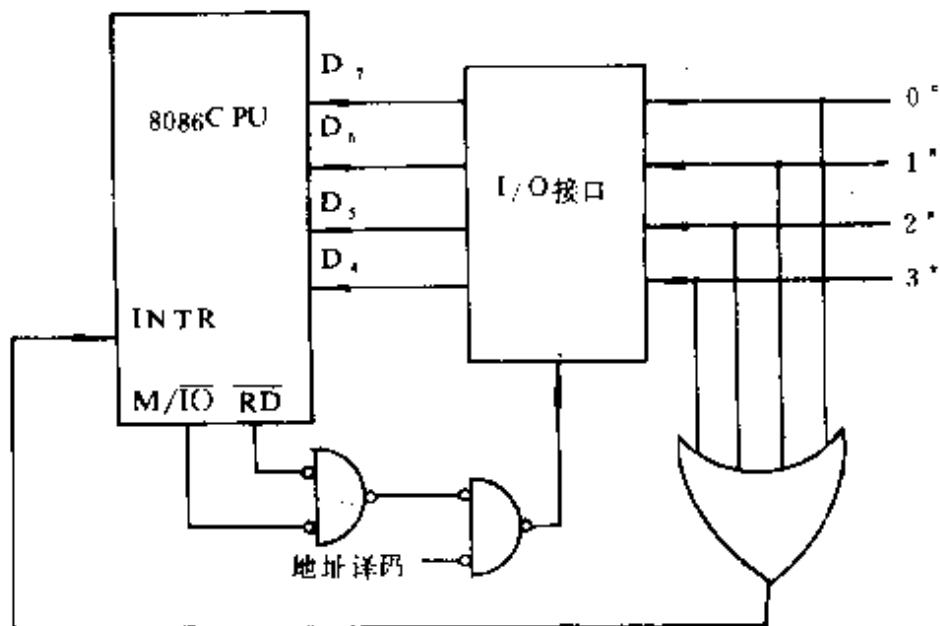
◆ 二、矢量中断



9. 3 中断源识别及中断优先权

★ 9. 3. 2 中断优先权

◆ 一、软件方案



软件查询确定优先权的缺点是，响应中断慢

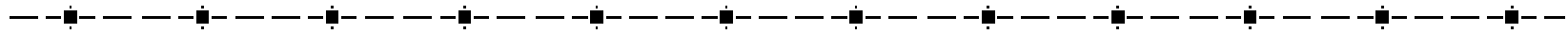
9. 3 中断源识别及中断优先权

✦ 9. 3. 2 中断优先权

◆ 二、硬件方案

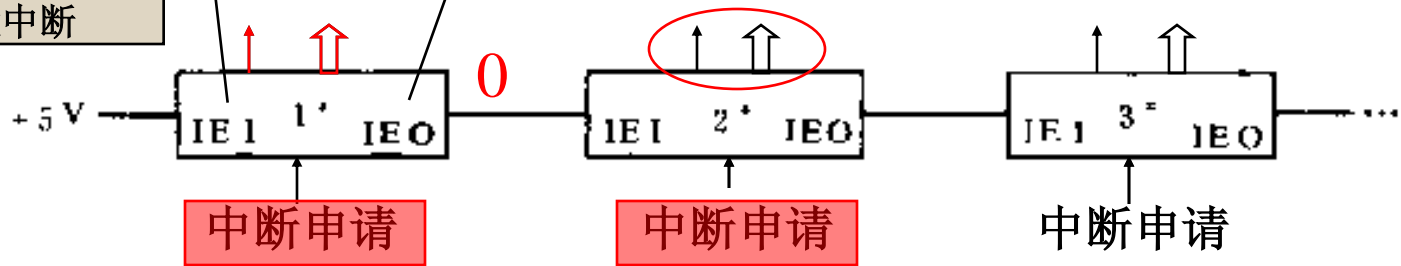
• 1、链形电路

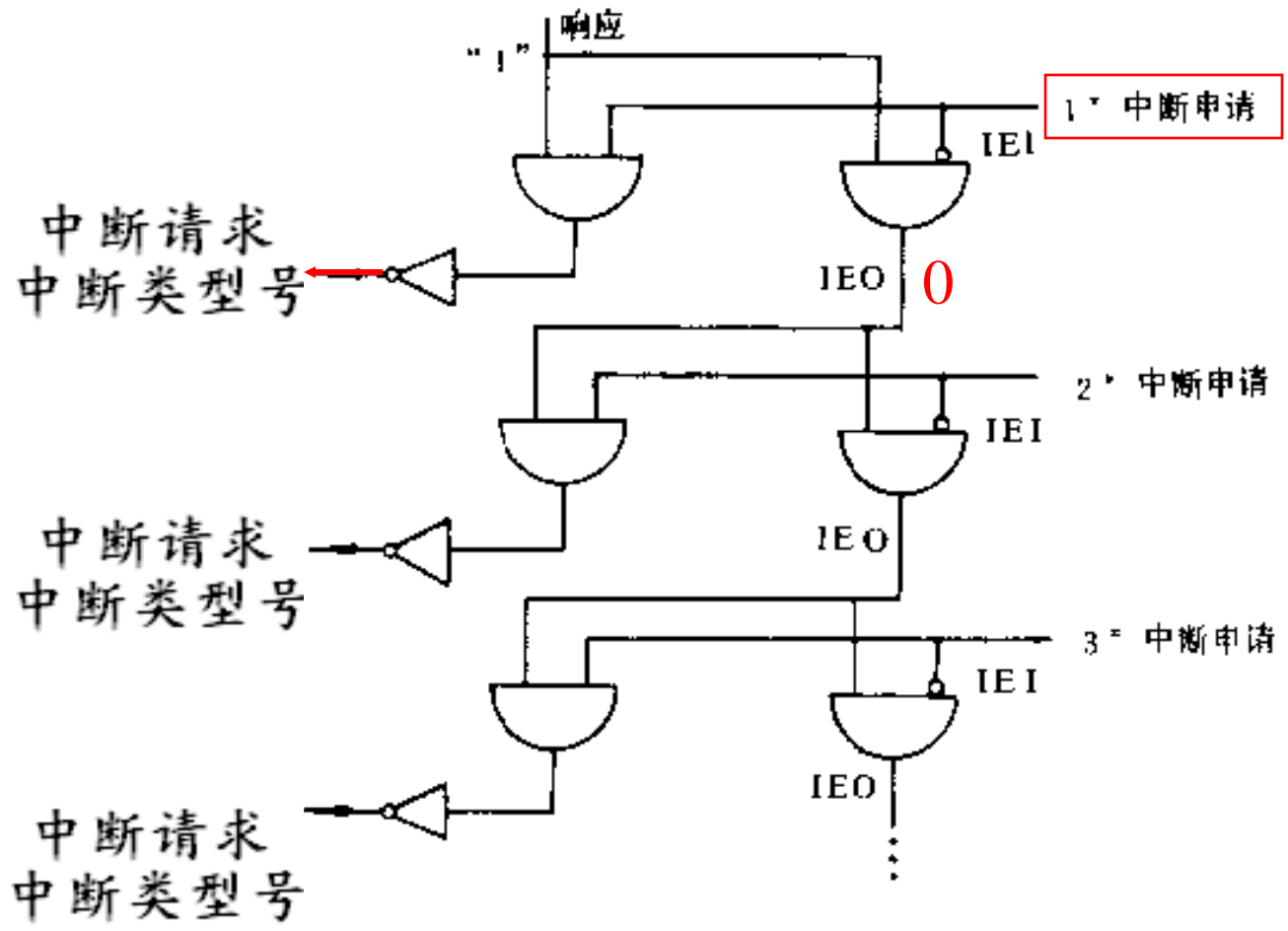
这种方法是利用外设系统中的物理位置来决定其中断优先权的

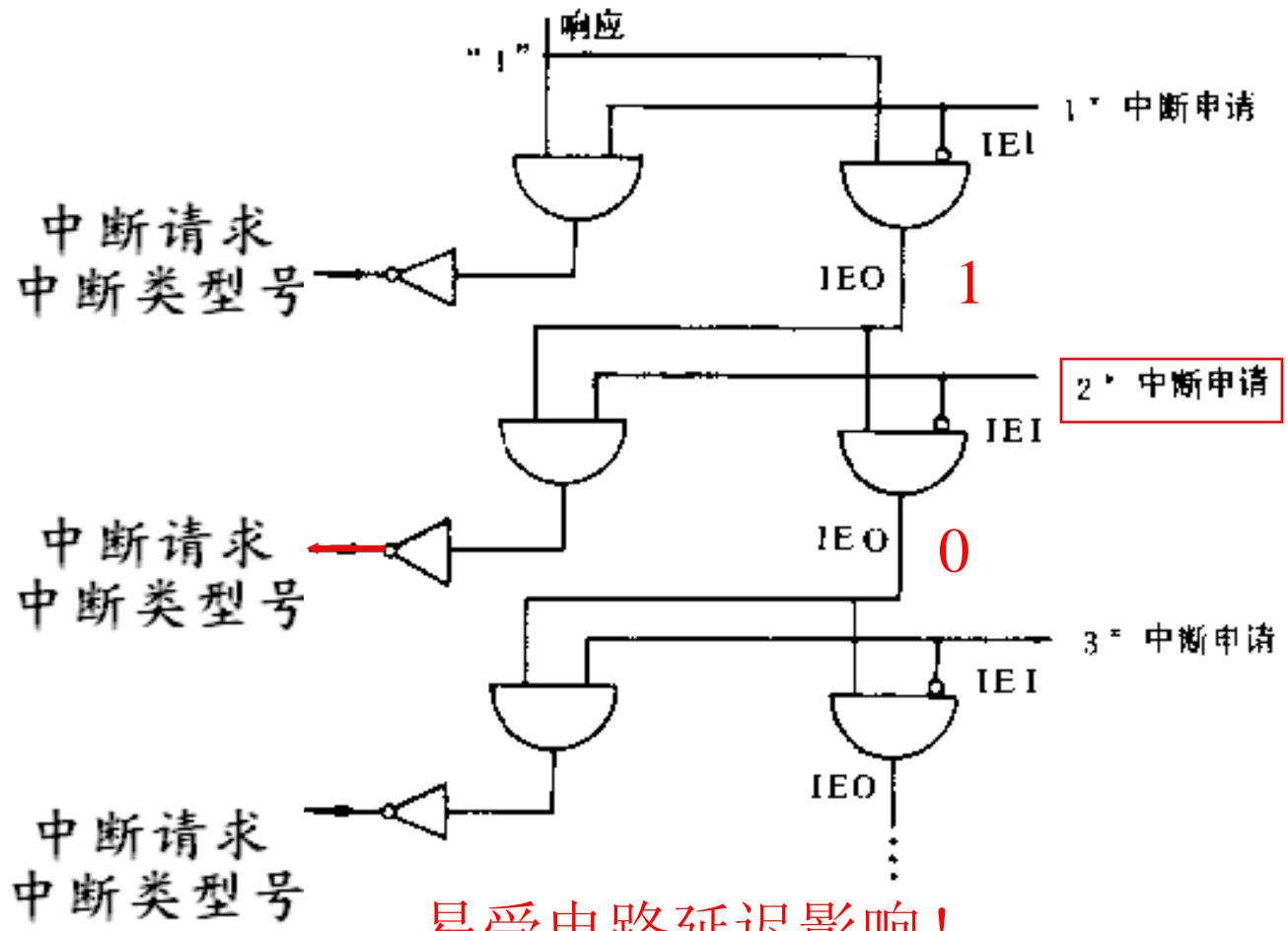


中断允许入
 $IEI=1$,
允许发中断

中断允许出 $IEO = IEI \overline{INTR}$







易受电路延迟影响!

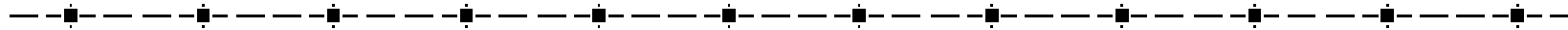
9. 3 中断源识别及中断优先权

✦ 9. 3. 2 中断优先权

◆ 二、硬件方案

• 2、编码电路

74LS148是一个8到3线的优先权编码器，它是一个16个管脚双列直插式TTL器件。



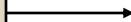
I0



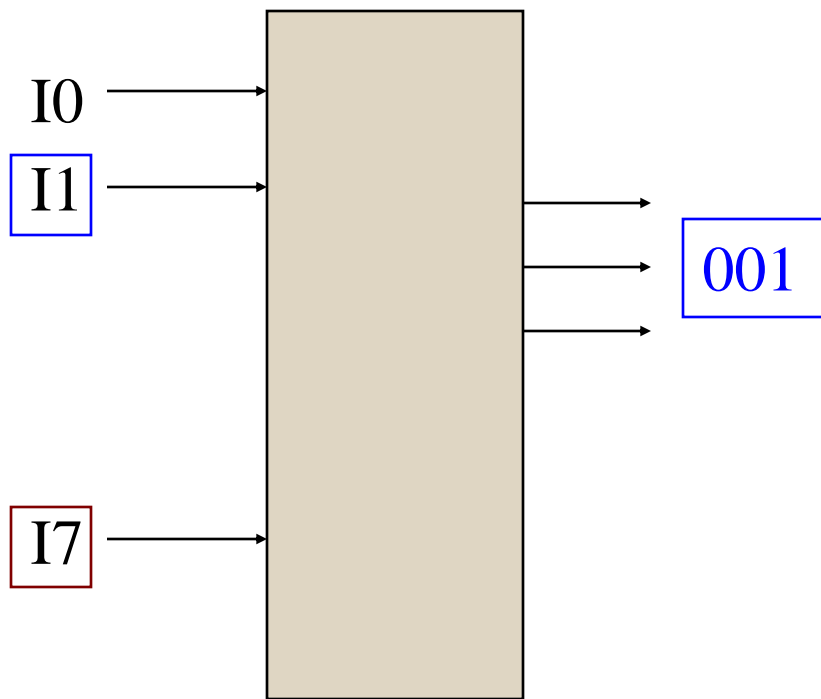
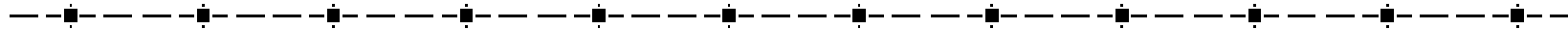
I1

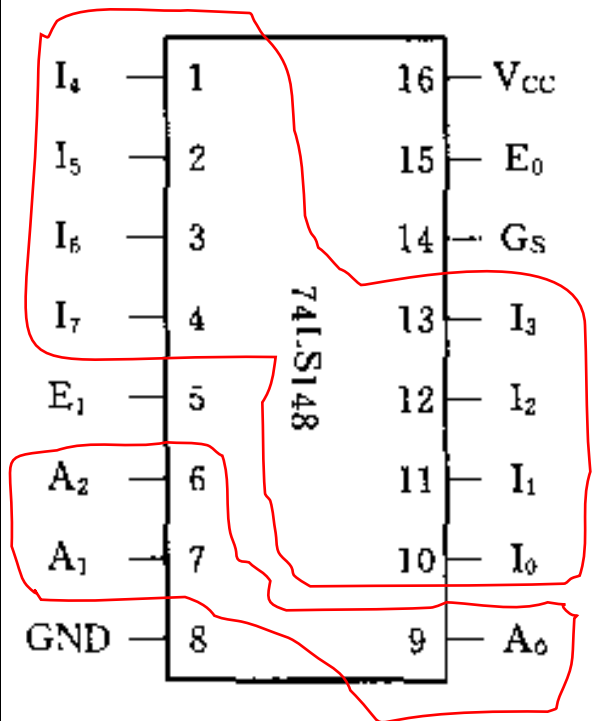


I7



000



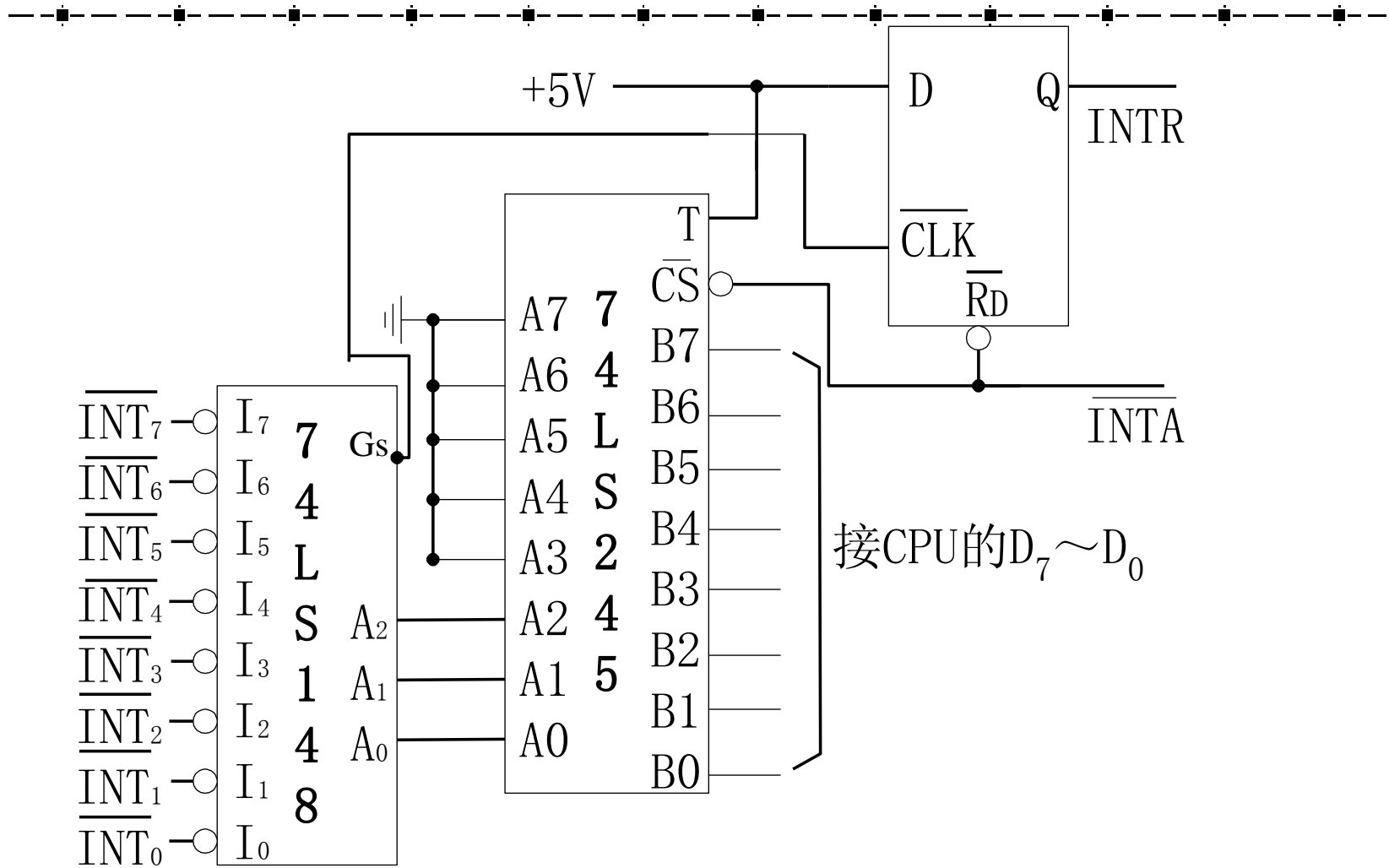


中断申请信号

输 入								输 出					
E_1	0	1	2	3	4	5	6	7	A_2	A_1	A_0	G_s	E_0
1	×	×	×	×	×	×	×	×	1	1	1	1	1
0	1	1	1	1	1	1	1	1	1	1	1	1	0
0	×	×	×	×	×	×	×	0	0	0	0	0	1
0	×	×	×	×	×	×	0	1	0	0	1	0	1
0	×	×	×	×	×	0	1	1	0	1	0	0	1
0	×	×	×	×	0	1	1	1	0	1	1	0	1
0	×	×	×	0	1	1	1	1	1	0	0	0	1
0	×	×	0	1	1	1	1	1	1	0	1	0	1
0	×	0	1	1	1	1	1	1	1	1	0	0	1
0	0	1	1	1	1	1	1	1	1	1	1	0	1

级联控制信号

图 7.4 74LS148 编码器管脚图及真值表



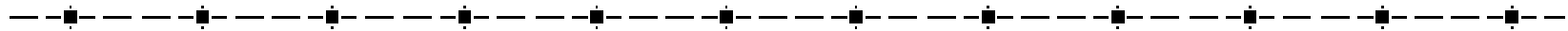
9. 4 8086中断系统

8086中断系统有两大类型的中断源：

一类是由外部设备产生的中断，我们称之为**硬件中断**，硬件中断有时又称**外中断**。对于硬件中断，又分为**不可屏蔽中断**和**可屏蔽中断**，硬件中断是通过CPU芯片的INTR管脚或NMI管脚从外部引入的。

另一类是由指令在某种运行结果时产生的中断，我们称之为**软件中断**。

中断类型号和中断服务子程序的入口地址之间的关系：



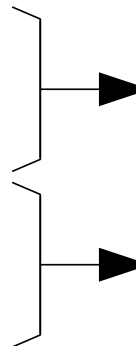
中断类
型号n

×4

0000:4n+0
0000:4n+1
0000:4n+2
0000:4n+3

内存

00
10
00
20



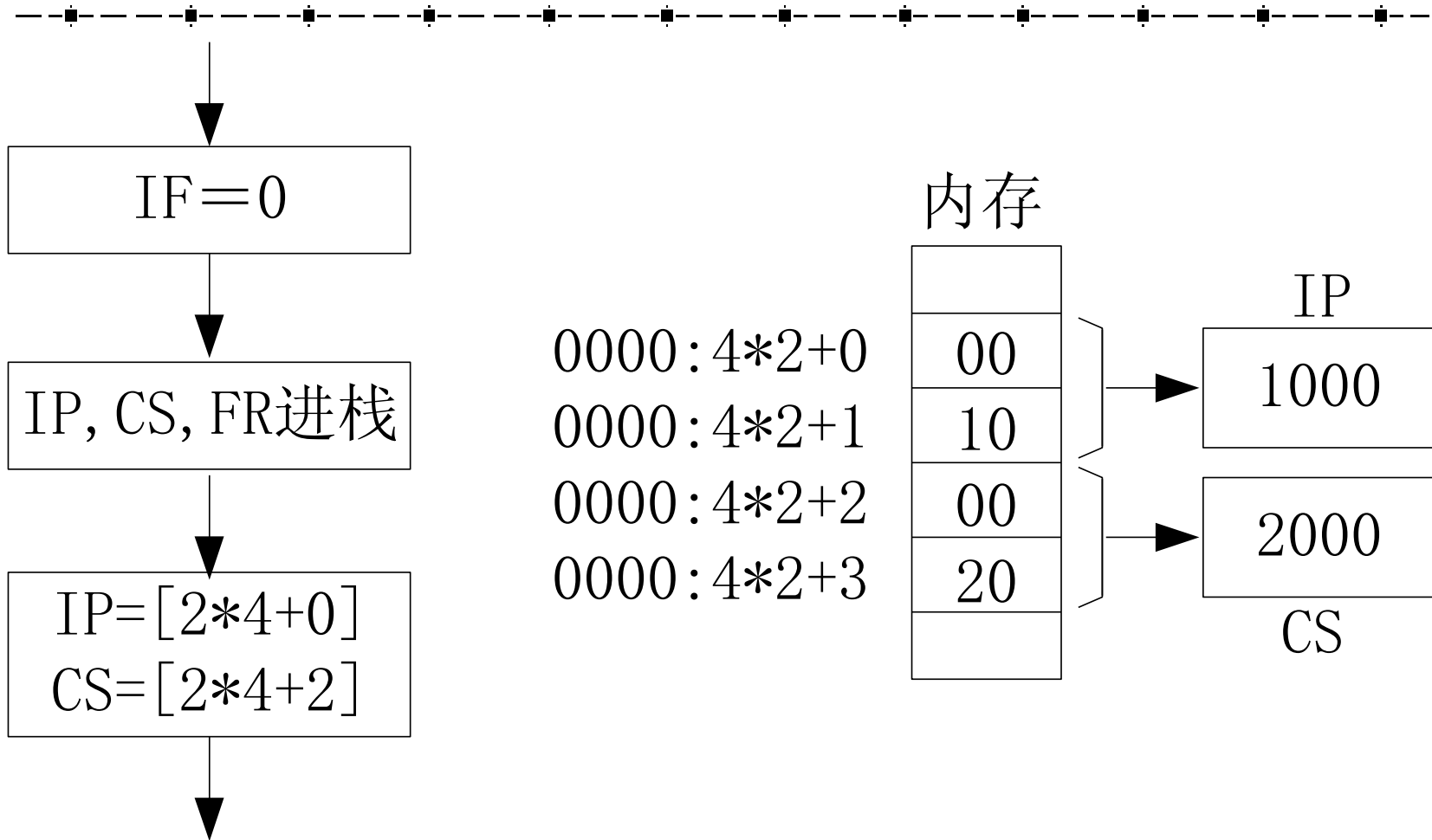
IP
1000

2000
CS

9. 4 8086中断系统

✦ 9. 4. 1 不可屏蔽中断

所谓不可屏蔽中断就是用户不能通过CPU内的中断允许触发器IF控制的中断，由8086CPU的NMI管脚引入。NMI中断请求采用上升沿触发方式，这种中断一旦产生，在CPU内部直接生成中断类型号02。



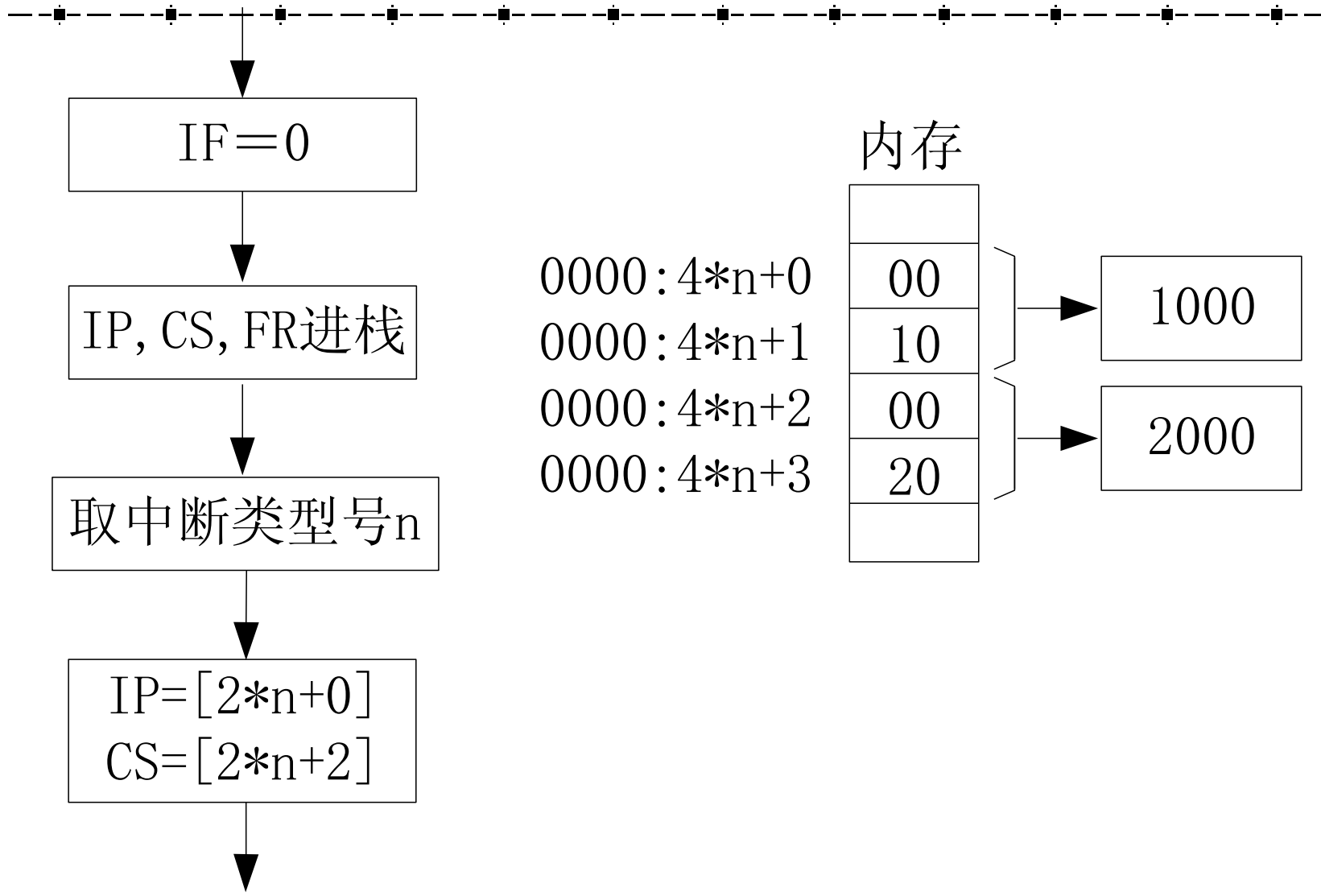


✦ 如何处理多个NMI中断源？

9. 4 8086中断系统

✦ 9. 4. 2 可屏蔽中断

可屏蔽中断就是用户可以控制的中断，其途径是通过对CPU内的中断允许触发器IF的设置来禁止和允许CPU响应中断。可屏蔽中断由8086CPU的INTR管脚引入。这种中断请求需要设备提供中断类型号，CPU响应中断后，取中断类型号的4倍作为中断服务入口地址表的地址，通过查表得到相应的中断服务程序首地址，转去执行相应的中断服务程序。



9. 4 8086中断系统

✦ 9. 4. 3 软件中断

软件中断是由8086指令系统中的某些指令产生，或由这些指令运行后某种特定的结果产生。

9. 4 8086中断系统

✦ 9. 4. 3 软件中断

◆ 一、除法中断

当进行除法运算时，若除数为0或除数太小，使得商数大于相应寄存器所能表示的最大值，被称作除法出错。这时除法指令就相当于一个中断源，它向CPU发出类型0中断。

9. 4 8086中断系统

✧ 9. 4. 3 软件中断

◆ 二、溢出中断

当算术运算产生溢出时，将在INTO指令控制下向CPU发出类型4的中断，即溢出中断。

比如

```
mov al,40h
```

```
add al,40h
```

```
into
```

9. 4 8086中断系统

★ 9. 4. 3 软件中断

◆ 三、单步中断和断点中断

- (1) 单步中断

当8086CPU的标志寄存器中的TF标志为1时，8086CPU处于单步工作方式，这时CPU在每条指令执行后自动产生类型1的中断。

TF在标志寄存器的第8位。

- (2) 断点中断

INT3指令是1字节指令(0CCH)，

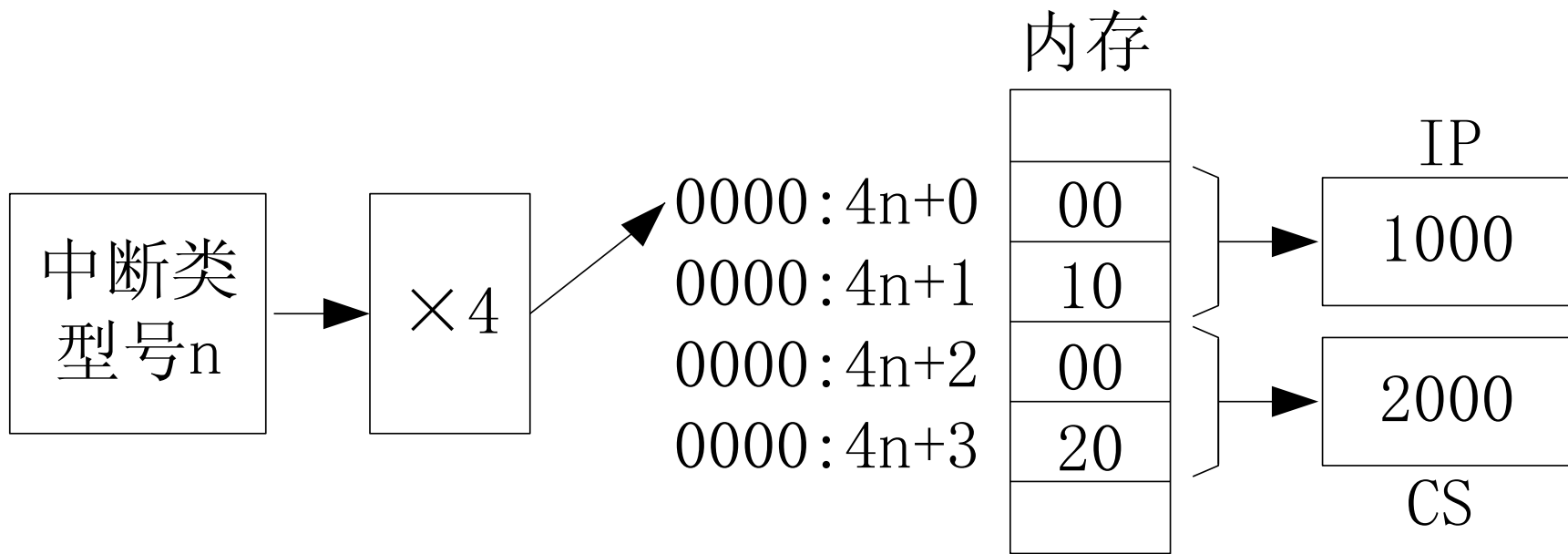
如何实现断点调试？

9. 4 8086中断系统

✦ 9. 4. 3 软件中断

◆ 四、软中断

软中断是由中断指令引起的。中断指令的指令格式为 **INT n** (CD 21)，操作数n就是中断类型号。当CPU执行完毕中断指令INT n后，就会立即产生一个中断类型号为n的中断。



int n → call far [4*n]

9.4 8086中断系统

✦ 9.4.4 中断概念的再讨论

◆ 一、中断与调用子程序之间的关系

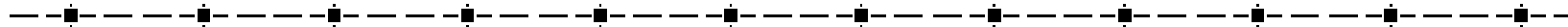
- 1、中断过程实际上是CPU从执行当前主程序转到执行为外设服务的子程序，因此从这个角度来看，是中断过程一个调用子程序的过程。
- 2、但中断过程与子程序调用还是有很大差别的，比如
 - ◆ 首先调用子程序的过程是一个无条件过程，但中断过程的中断服务程序的调用一般是有条件的。
 - ◆ 其次子程序调用在整个程序执行中的位置是固定的。但对于硬件中断过程，只要条件满足，在整个程序执行的任意一时间点都有可能发生从主程序向中断服务子程序的转移事件，也就是说硬件中断产生的调用过程是随机的，不可预测的。

9.4 8086中断系统

★ 9.4.4 中断概念的再讨论

◆ 二、外部中断和 INT n 之间的关系

结合中断指令INT n，我们可以这样理解外部中断：当外部中断源发中断给CPU时，如果CPU满足一定的条件，处于开中断状态，CPU就可以响应中断，这时外设正在CPU正在执行指令与其下一个指令之间，等效“插入”了一个INT n指令，这里的n就是外设提供的中断类型号。注意这里用“等效”两字表示实际过程中是不存在插入INT n指令的操作，但CPU确实完成了类似INT n指令的功能，实现了主程序向中断程序的转移。



.....

INT n



MOV AL,10



MOV BX,100

.....

INT n



MOV BL,100



ADD AL,BL

.....

9.5 8086CPU的中断管理

✦ 9.5.1 8086CPU的中断处理顺序

8086CPU的中断优先权排列从高到低为：

- ◆ 1、除法出错中断，溢出中断，INT n；
- ◆ 2、NM1；
- ◆ 3、INTR；
- ◆ 4、单步中断。

9.5 8086CPU的中断管理

★ 9.5.2 8086CPU的中断服务入口地址表

专用中断	000H	除法中断入口	IP
	004H	单步中断入口	CS
	008H	NMI 中断入口	
	00CH	断点中断入口	
	010H	溢出中断入口	
	014H	类型 5 中断入口	
		⋮	
	07FH	类型 31 中断入口	
	080H	类型 32 中断入口	
		⋮	
	3FCH	类型 255 中断入口	

9.5 8086CPU的中断管理

★ 9.5.3 中断入口地址设置

◆ 一、用串指令

CLI ; 关中断
MOV AX, 0
MOV ES, AX ; 置附件段基地址为0
MOV DI, n*4 ; 置附件段偏移地址到DI
MOV AX, OFFSET INT_VCE ; 置中断程序首地址的偏移量到AX
CLD
STOSW ; 填偏移量到中断地址表
MOV AX, SEG INT_VCE ; 置中断程序的段基地址到AX
STOSW ; 填段基地址到中断地址表
STI ; 开中断

.....

9.5 8086CPU的中断管理

★ 9.5.3 中断入口地址设置

◆ 二、用伪指令

```
INT—TBL SEGMENT AT 0 ; 定义INT—TBL段，段基地址为0
    ORG n*4           ; 指定偏移地址
    DD INT—VCE       ; 存中断程序入口地址
INT—TBL ENDS
    .....           ; 其他处理
MCODE SEGMENT        ; 主程序
    .....           ; 其他处理
INT—VCE PROC FAR    ; 中断类型4的服务程序
    .....
    IRET
INT—VCE ENDP
    .....
```

8.5 8086CPU的中断管理

✦ 8.5.3 中断入口地址设置

◆ 三、用系统调用

使用25H功能时要求：

AL = 中断类型号

DS: DX = 中断服务程序首地址的段、偏移地址

注意DS保护



PUSH DS

MOV DX, SEG INT60H ; 段基地址送DS

MOV DS, DX

MOV DX, OFFSET INT60H ; 偏移地址送DX

MOV AL, 60H ; 中断类型号送AL

MOV AH, 25H

INT 21H ; 25H功能调用

POP DS

8.5 8086CPU的中断管理

✦ 8.5.3 中断入口地址设置

◆ 四、直接装入法

若外设的中断型号为6BH，则此中断型号对应的中断矢量表地址为从001ACH开始的四个存储单元。设中断服务程序段地址是1000H，偏移地址为2000H。



MOV AX,, 0

MOV DS, AX ; 置数据段段基地址为0

MOV AX, 2000H

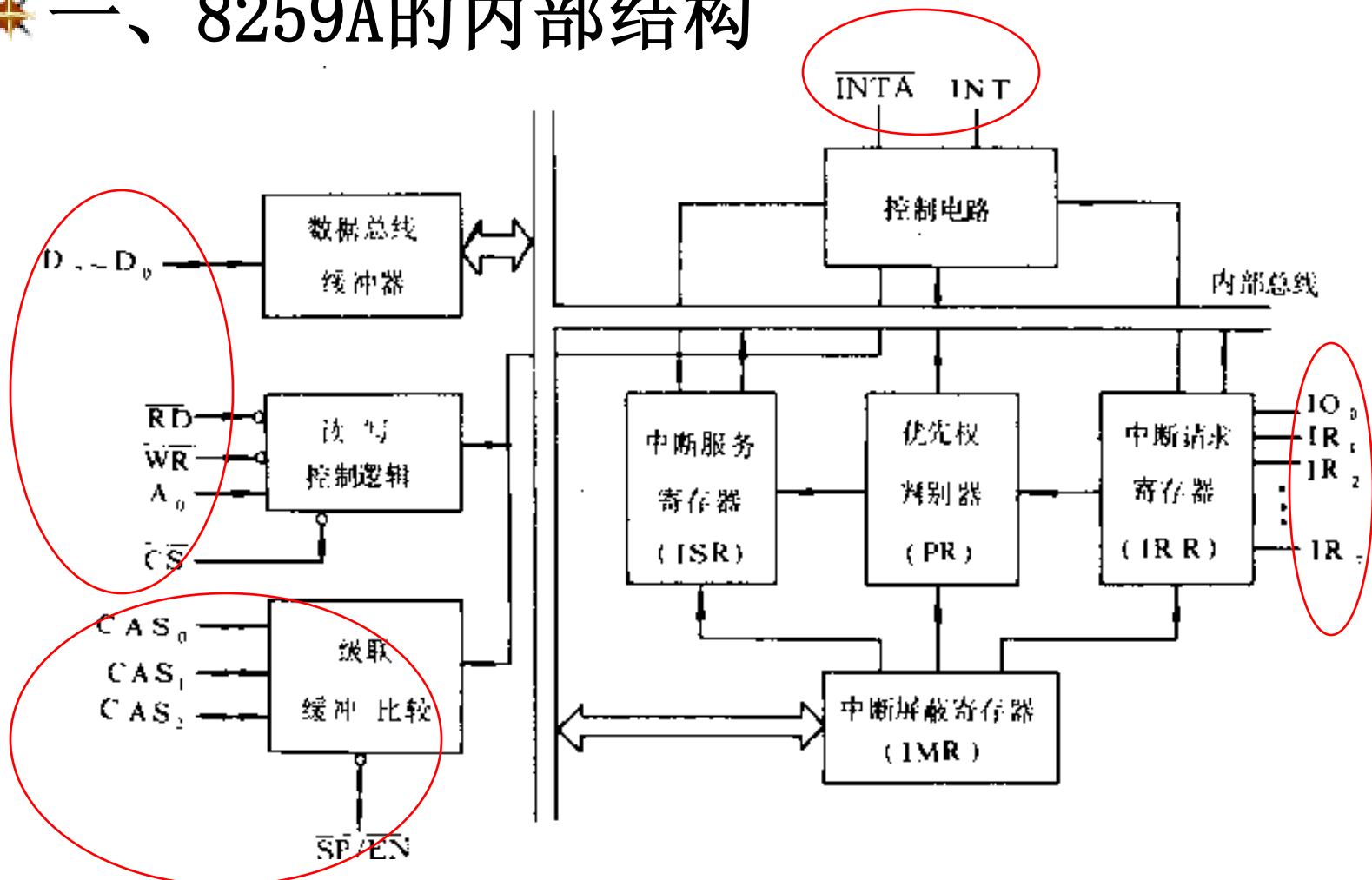
MOV [01ACH], AX ; 对偏移地址为01AC的单元

MOV AX,1000H ; 送双字

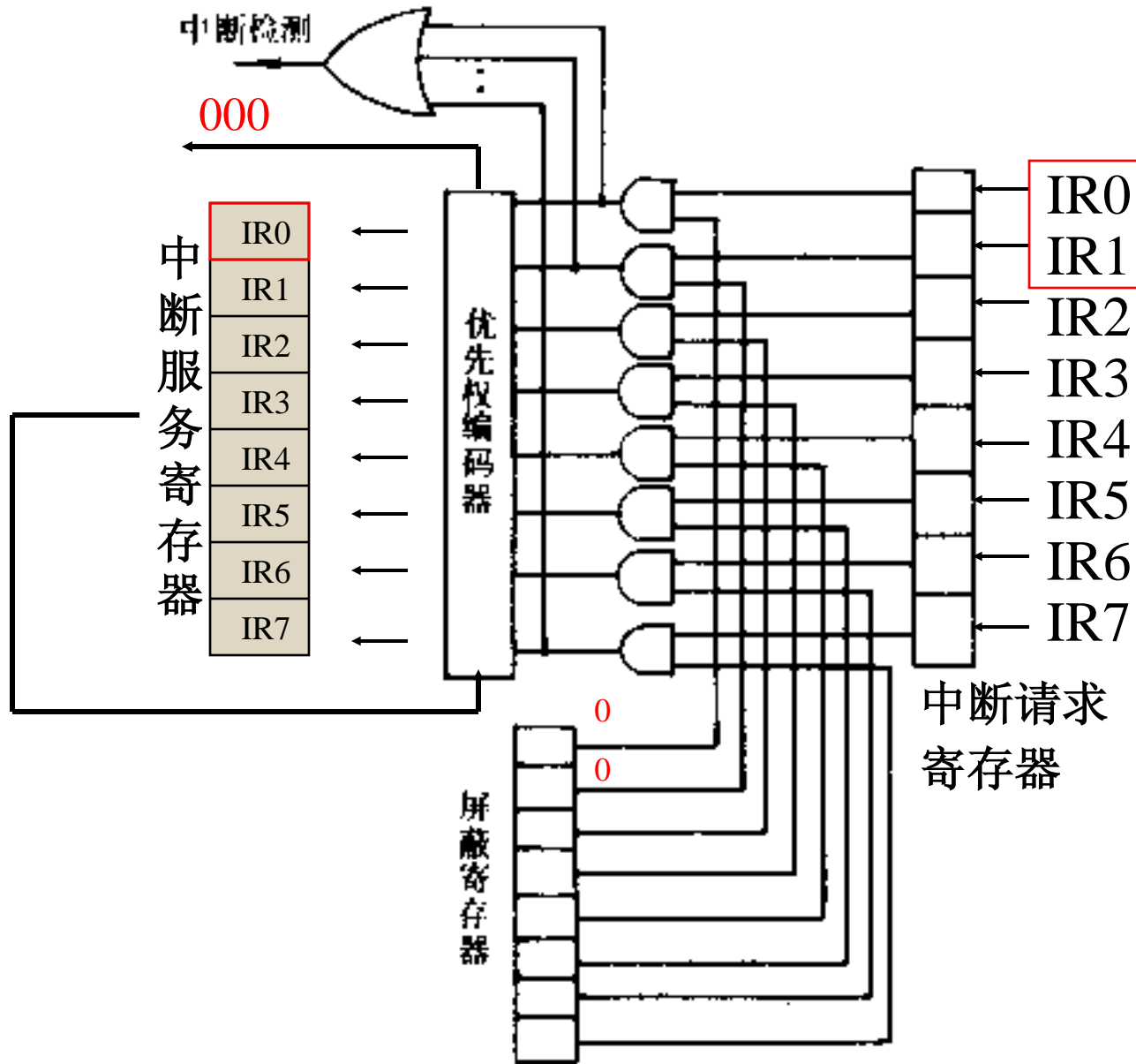
MOV [01ACH+2], AX

9.6 可编程中断控制器8259

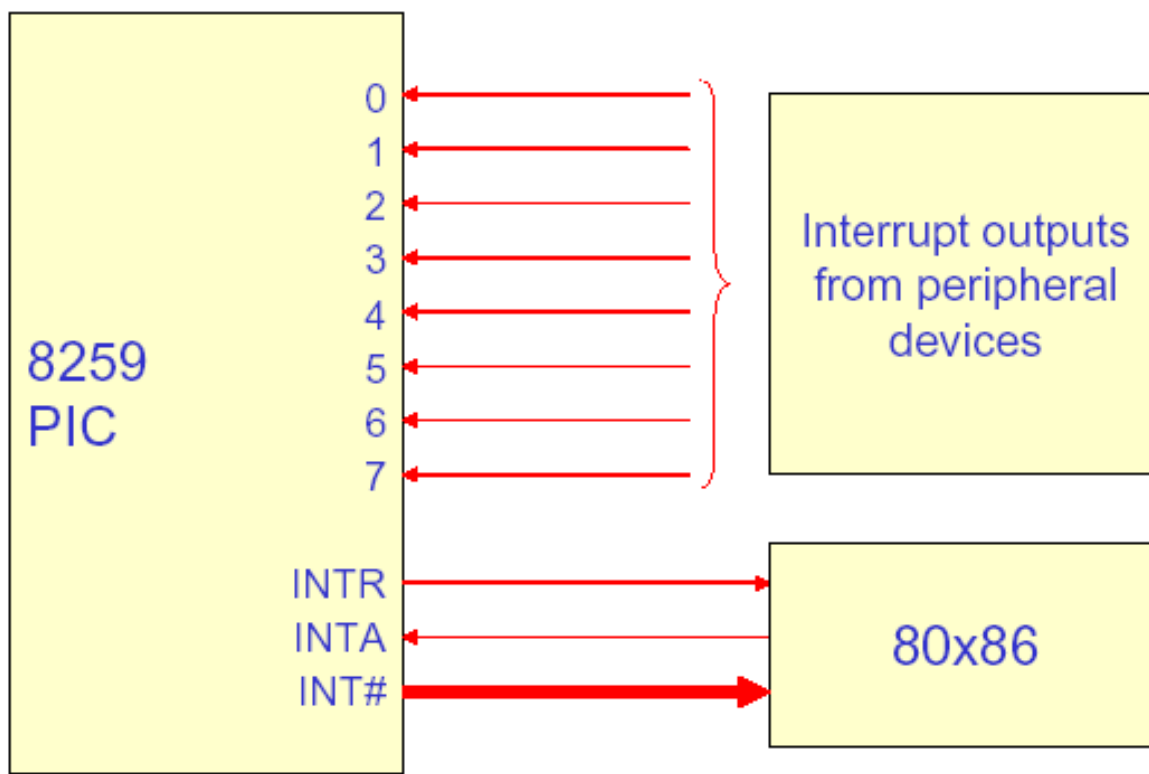
一、8259A的内部结构

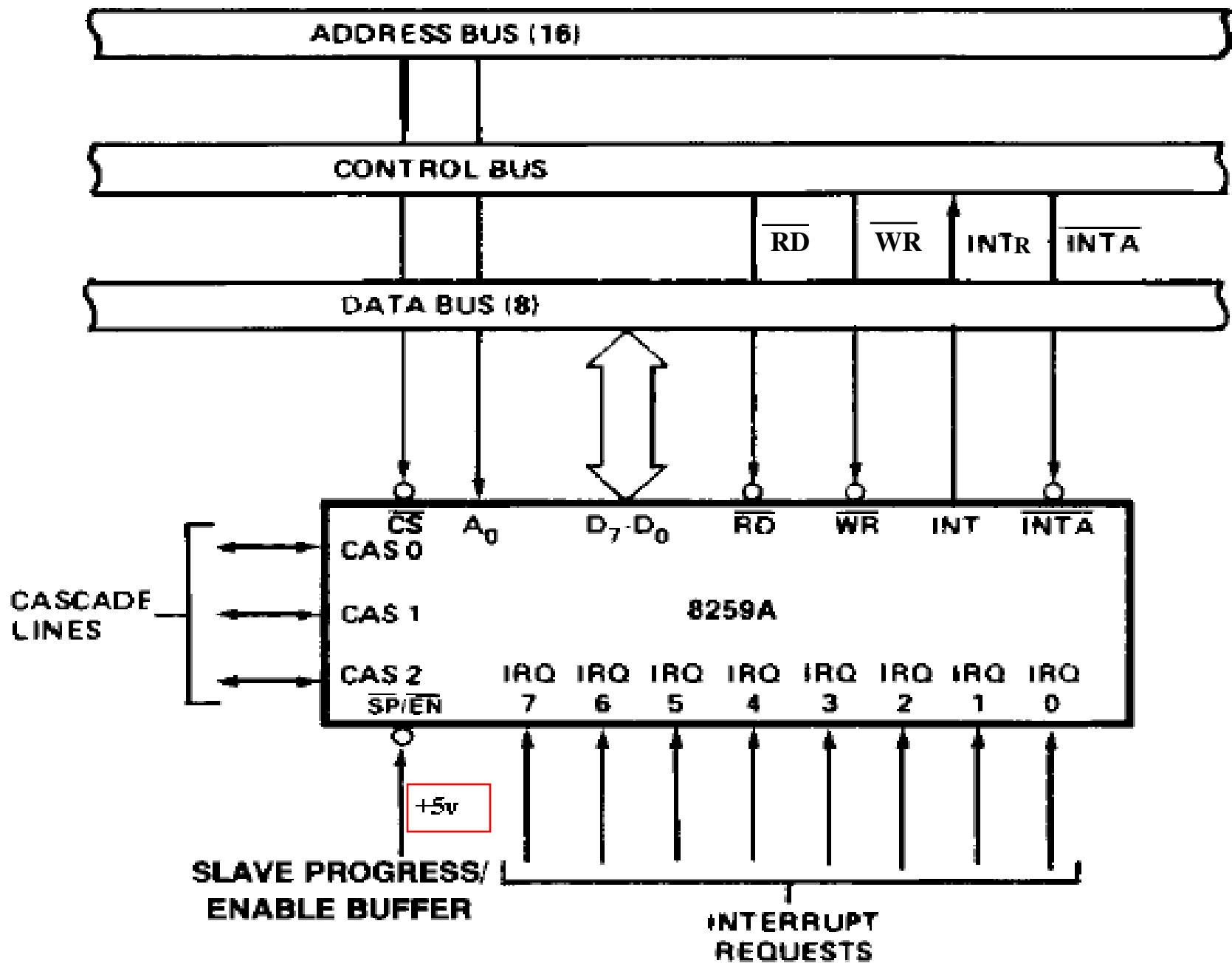


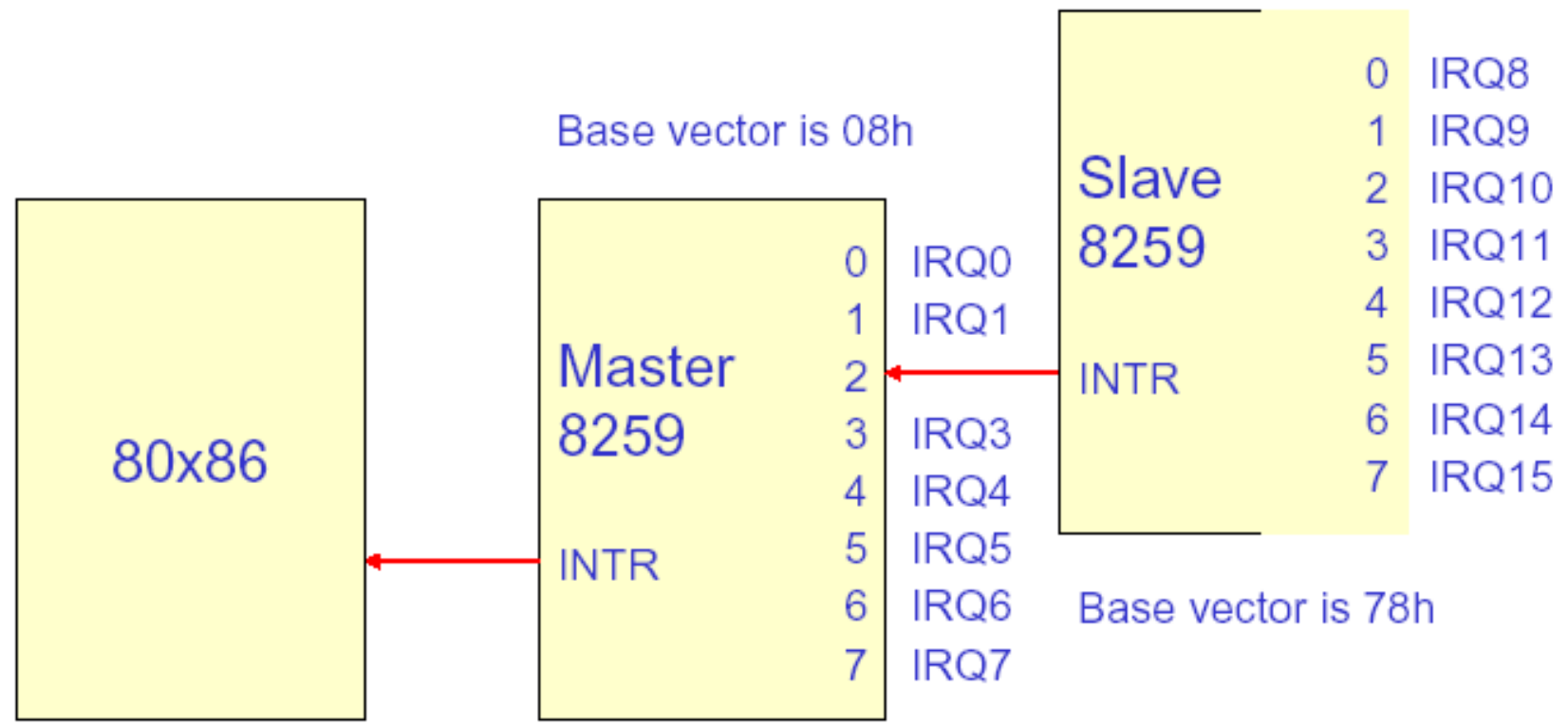
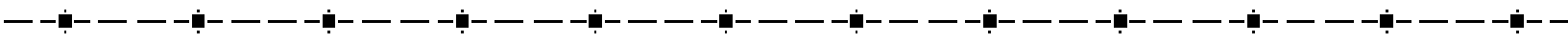
1、工作原理

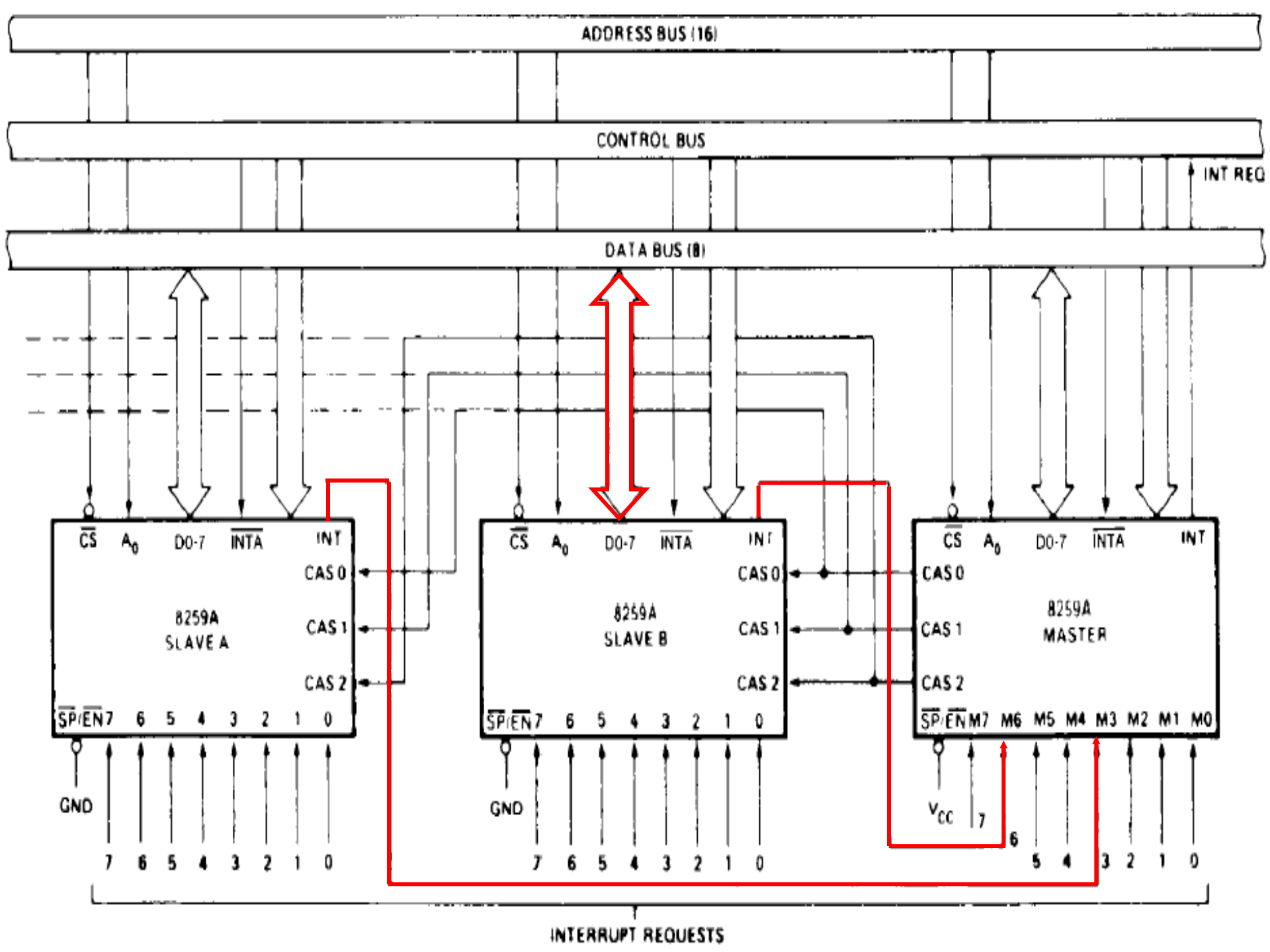


2、单级8259与8086CPU联接

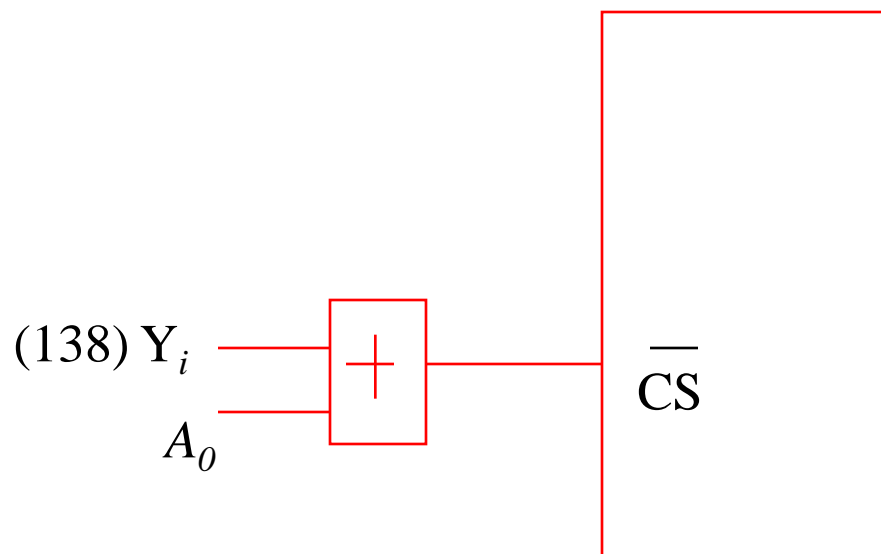








3、接口电路(设置偶地址)



9.6 可编程中断控制器8259

✧ 二、8259A的中断管理方式

◆ (一)、中断优先权管理

• 1. 完全嵌套方式

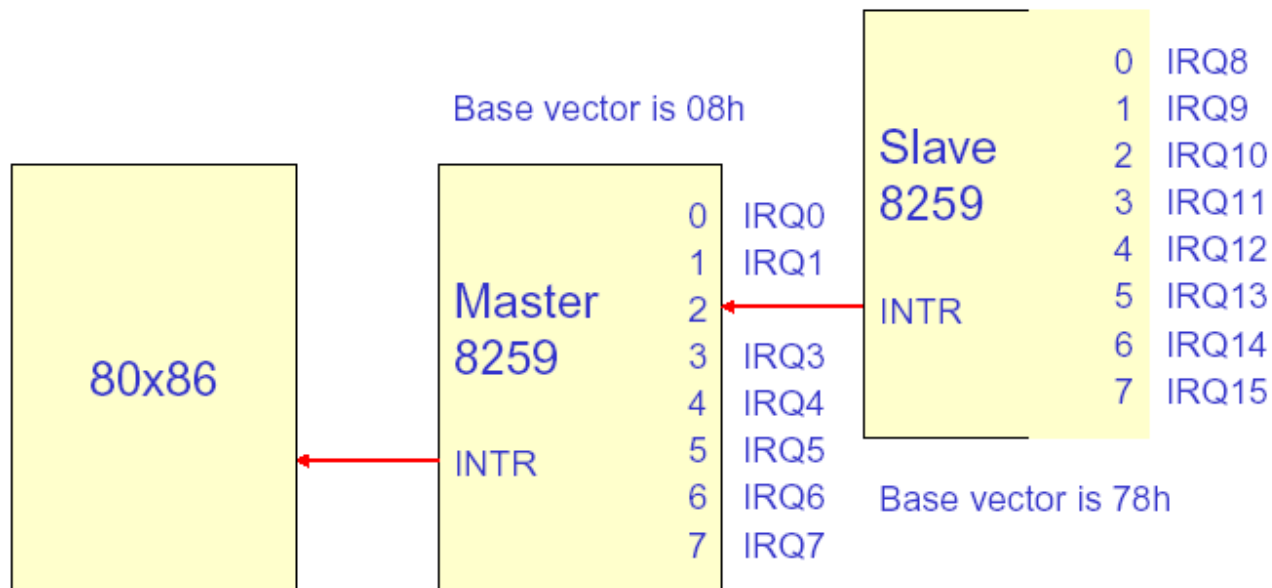
8259A的中断请求输入端引入的中断具有固定的优先权排队顺序， IR_0 为最高优先级， IR_1 为次高优先级，依次类推， IR_7 为最低优先级。

9.6 可编程中断控制器8259

- 2. 特殊全嵌套方式

特殊全嵌套方式和全嵌套方式基本相同，只有一点不同，就是在特殊全嵌套方式下，当处理某一级中断时，如果有同级的中断请求，那么，也会给予响应，从而实现一种对同级中断请求的特殊嵌套。

特殊全嵌套方式一般用在8259A级连的系统中。



9.6 可编程中断控制器8259

- 3. 自动循环方式

从 IR_0 — IR_7 引入的中断轮流具有最高优先权。

初始优先级队列规定为

$IR_0 \succ IR_1 \succ IR_2 \succ IR_3 \succ IR_4 \succ IR_5 \succ IR_6 \succ IR_7$

如果 IR_0 端正好有中断请求，处理完后

$IR_1 \succ IR_2 \succ IR_3 \succ IR_4 \succ IR_5 \succ IR_6 \succ IR_7 \succ IR_0$

如果 IR_4 端正好有中断请求，处理完后

$IR_5 \succ IR_6 \succ IR_7 \succ IR_0 \succ IR_1 \succ IR_2 \succ IR_3 \succ IR_4$

9.6 可编程中断控制器8259

- 4. 优先级特殊循环方式

最低优先级是由编程确定的，从而最高优先级也由此而定。

比如定IR5最低，则IR6最高

IR6>IR7>IR0>IR1>IR2>IR3>IR4>IR5

9.6 可编程中断控制器8259

- (三) . 中断屏蔽方式

有两种方法：

第一、普通屏蔽方式。

第二、采用特殊屏蔽方式。

系统正在处理高级中断时，可以响应低级别中断

9.6 可编程中断控制器8259

◆ (四)、8259A中断结束的管理方式

可采用三种中断结束方式：

自动EOI方式

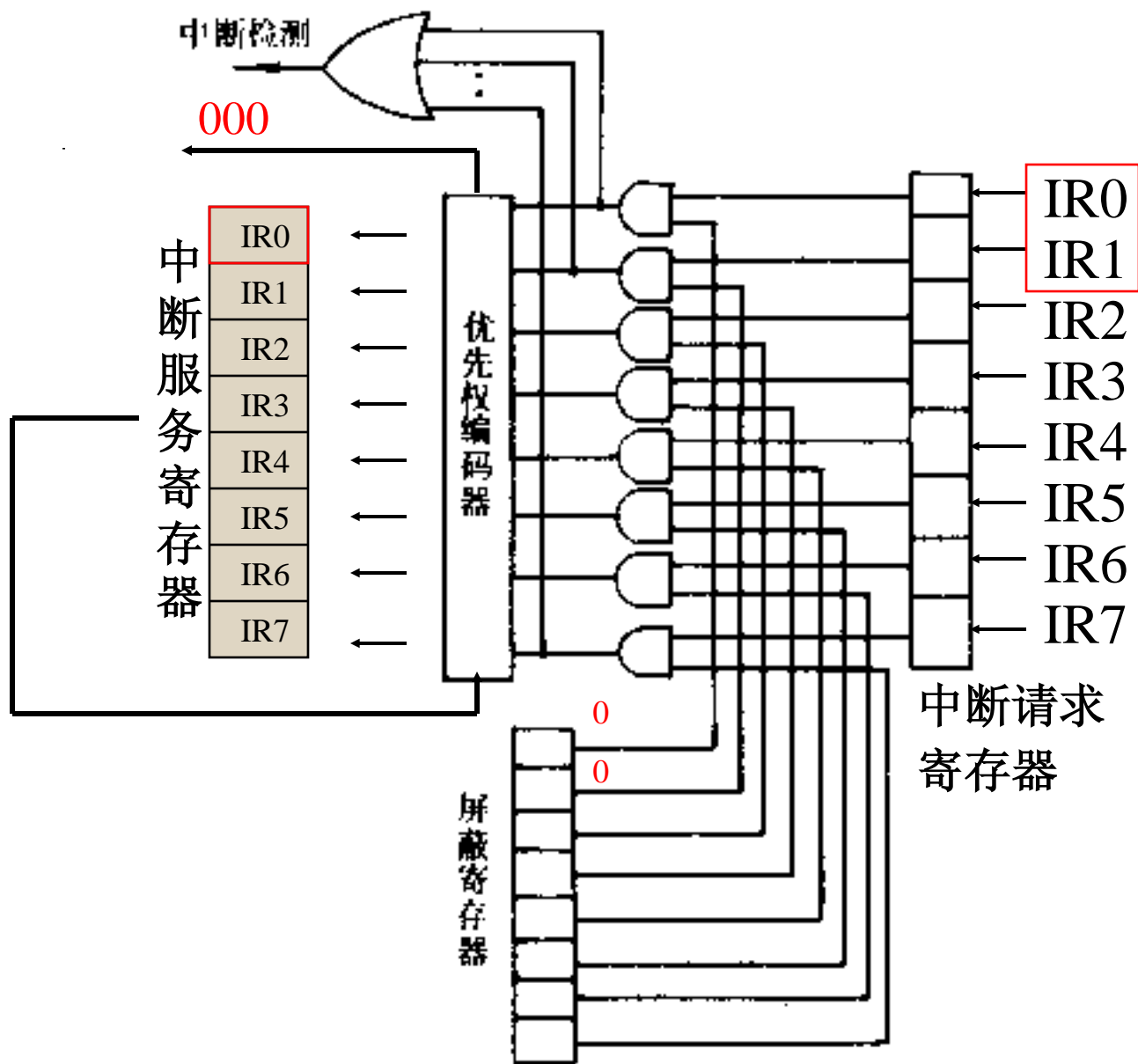
/INTA清除中断服务寄存器高级别位

一般EOI方式

**程序对8259发中断结束命令，清除
中断服务寄存器高级别位**

特殊EOI方式

**程序对8259发中断结束命令，清除
中断服务寄存器指定位**



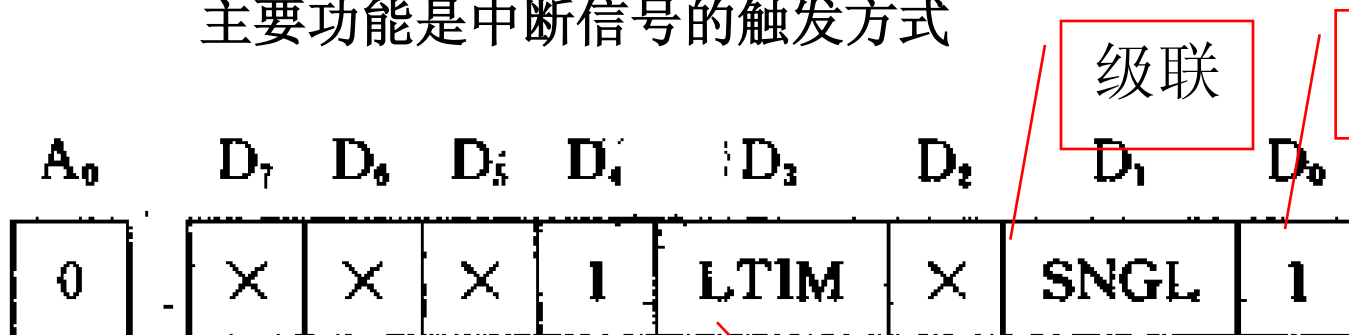
9.6 可编程中断控制器8259

◆ 三、 8259A编程

- (一)、8259A的初始化命令字

- ◆ ICW1的格式和含义

主要功能是中断信号的触发方式



例如：上升沿触发，级联

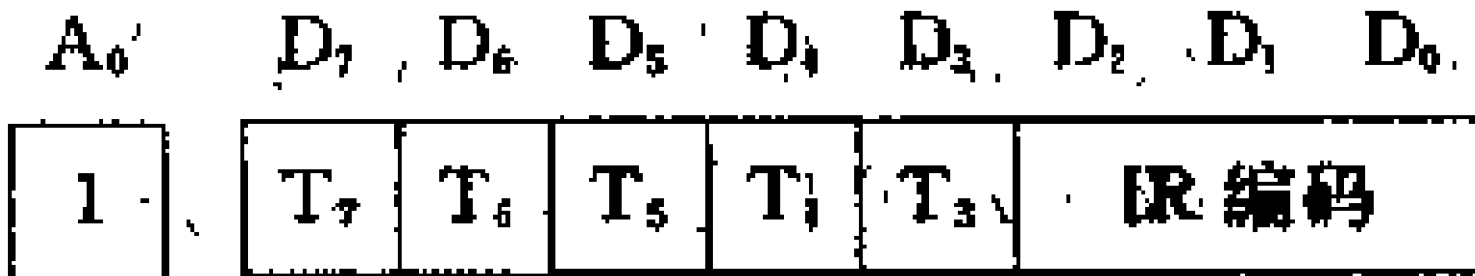
0 0 0 1 0 0 0 1

9.6 可编程中断控制器8259

◆ 三、 8259A编程

- (一)、8259A的初始化命令字

- ◆ ICW2的格式和含义
功能是中断类型号



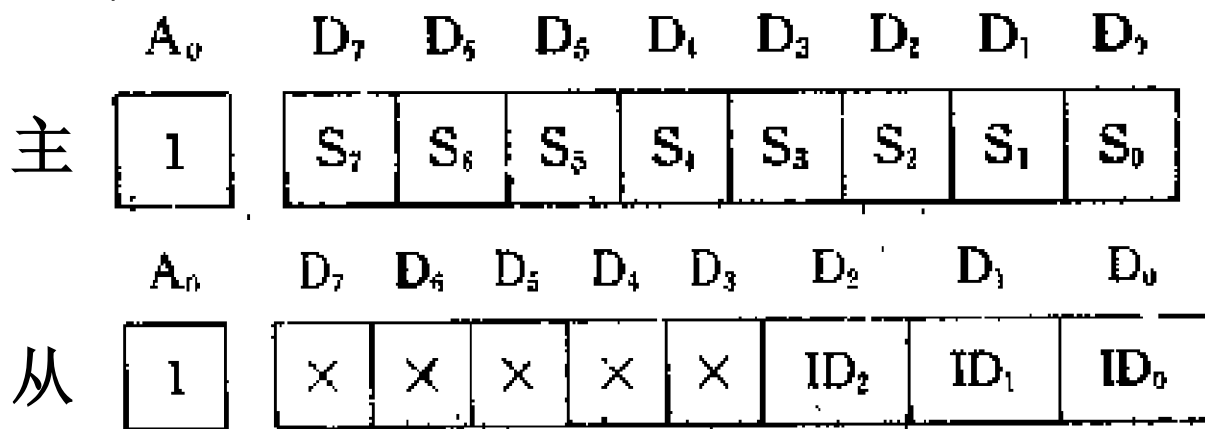
例如：中断类型号为08~0F

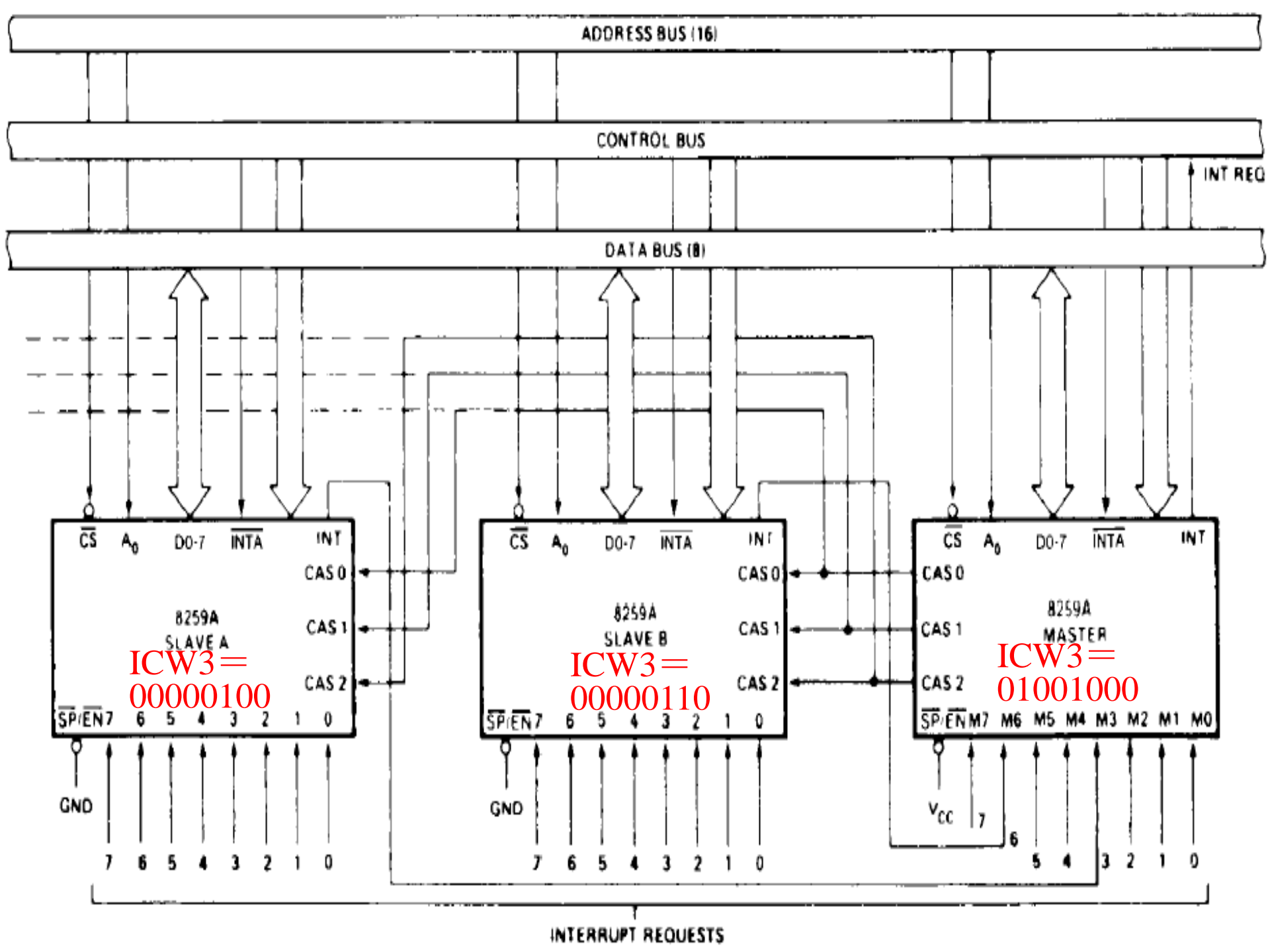
0 0 0 0 1 x x x

9.6 可编程中断控制器8259

◆三、 8259A编程

- (一)、8259A的初始化命令字
 - ◆ ICW3的格式和含义
在8259A有级联情况下使用





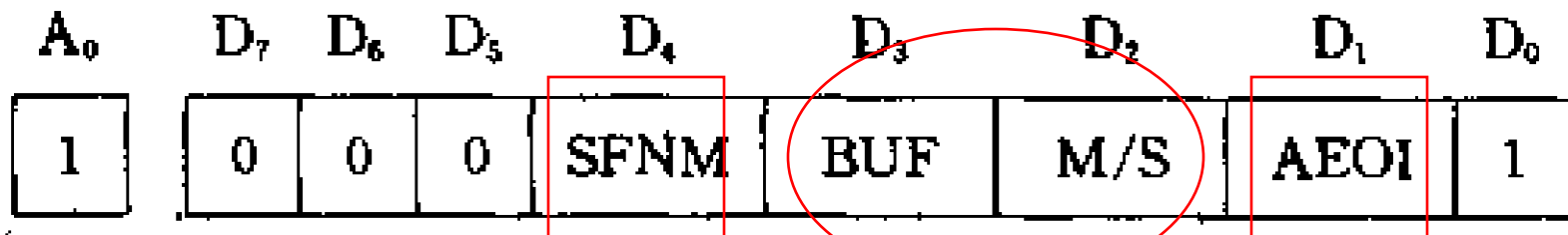
9.6 可编程中断控制器8259

◆ 三、 8259A编程

- (一)、8259A的初始化命令字

- ◆ ICW4的格式和含义

主要功能是中断嵌套方式选择, 使能中断结束方式



D1=1为自动EOI, D1=0为非自动EOI

D4位SFNM为有级联时定义的, D4=1为特殊完全嵌套方式, D4=0完全嵌套方式.

例如: 0 0 0 0 0 0 0 0 1

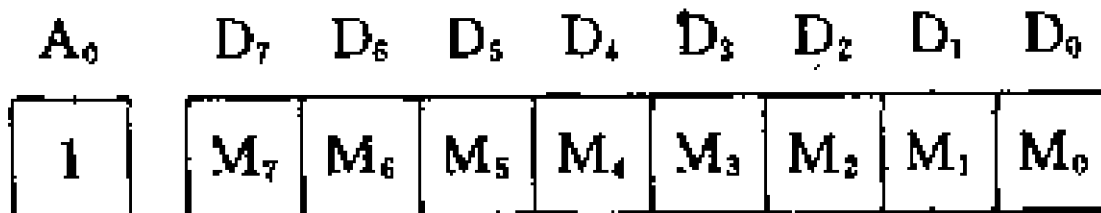
9.6 可编程中断控制器8259

◆ 三、 8259A编程

- (二)、8259A的操作命令字

 - ◆ OCW1的格式和含义

主要功能是确定屏蔽位



1表示禁止中断

0 0 0 1 0 0 0 1

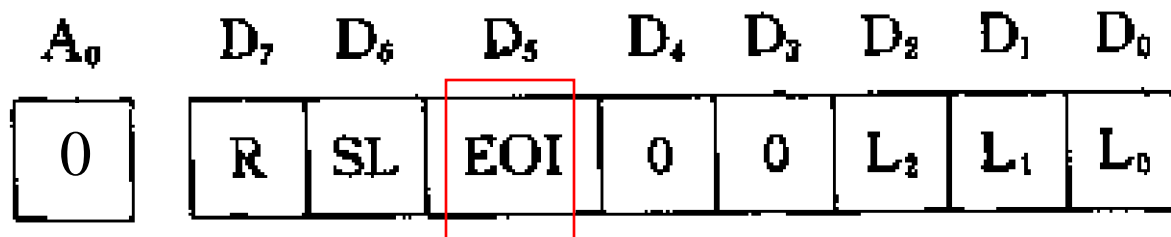
9.6 可编程中断控制器8259

◆ 三、 8259A编程

- (二)、8259A的操作命令字

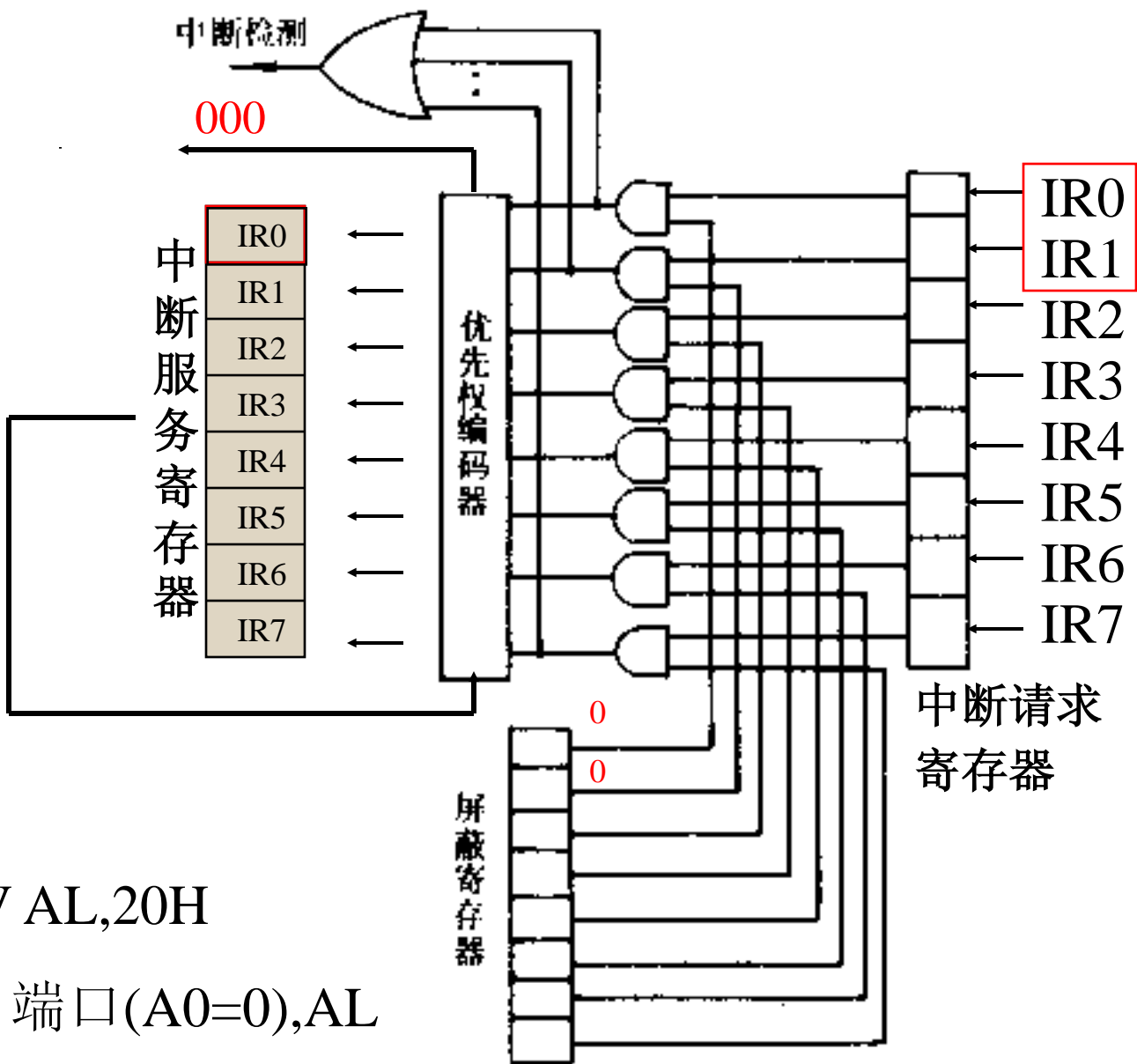
- ◆ OCW2的格式和含义

主要功能是**控制中断结束**，优先级循环等操作



如果采用一般EOI方式

0 0 1 0 0 0 0 0



MOV AL,20H

OUT 端口(A0=0),AL

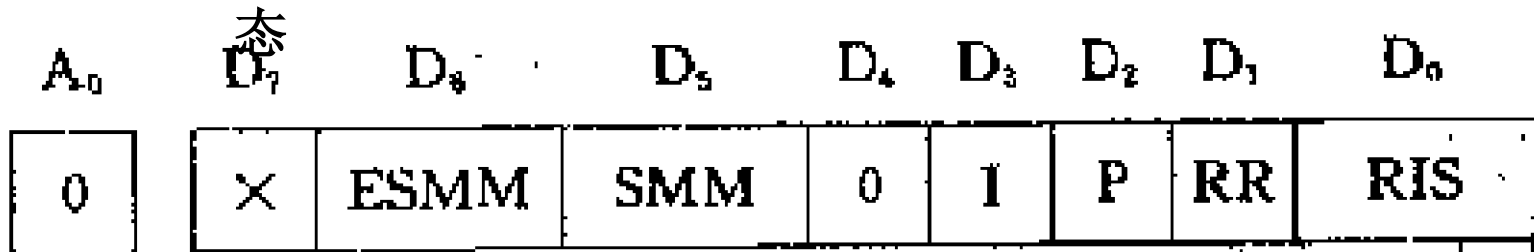
9.6 可编程中断控制器8259

◆ 三、 8259A编程

- (二)、8259A的操作命令字

- ◆ 0CW3的格式和含义

主要功能是控制8259A的中断屏蔽和读取寄存器的状



一般不用

9.7 中断举例

中断传送方式的设计过程是：

一、硬件方面：

 设定硬件的中断类型号。

二、软件方面：

 分主程序和中断子程序

 中断子程序：编写中断服务子程序。

 主程序：

- 1、关中断；
- 2、填写中断入口地址表；
- 3、设置8259A及其他编程芯片；
- 4、开中断；

举例一：软件中断

✦ INTO中断


- ✦ 执行结果由两个加数决定，

 - 有溢出：BL=0FFH；

 - 无溢出：BL=00。



```
INTO1    PROC           ; 中断溢出处理
          PUSH AX
          MOV BL,0FFH
          POP AX
          IRET          ; 中断返回
INTO1    ENDP
```



```
DATA    SEGMENT          ; 定义数据段
ADD1    DB ?
ADD2    DB ?
DATA    ENDS
CODE    SEGMENT          ; 定义代码段
        ASSUME CS: CODE,DS:DATA
START:  MOV AX,DATA
        MOV DS,AX
```



MOV AX,0 ; 填写中断地址表

MOV ES,AX

MOV DI,04*4

; 存中断程序首地址的偏移量

MOV AX,OFFSET INTO1 ; 可用MOV AX,CS


CLD

STOSW


; 存中断程序首地址的段地址

MOV AX,SEG INTO1

STOSW



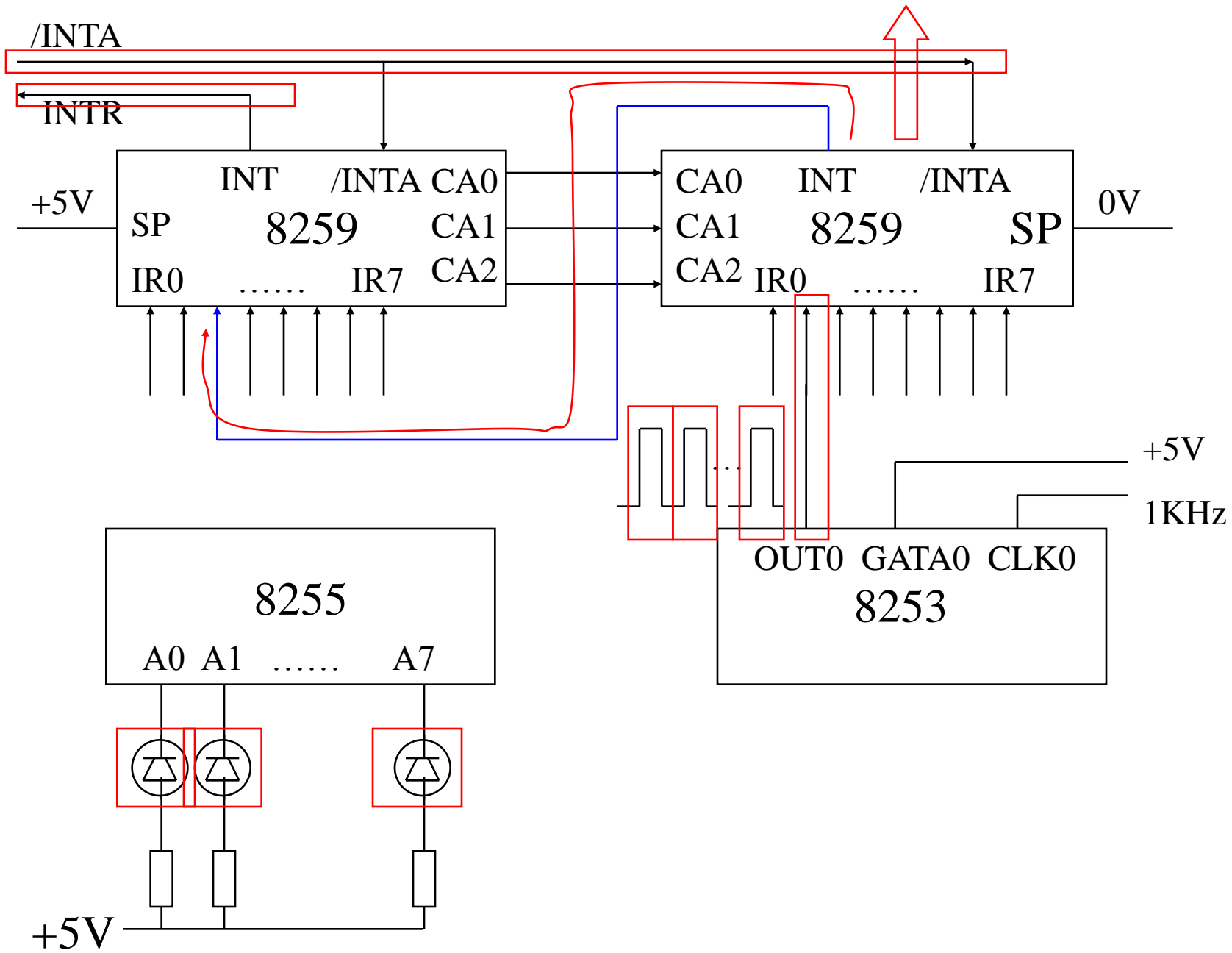
```
MOV BL,0           ; 主程序
MOV AL,ADD1        ; 计算ADD1+ADD2
ADD AL,ADDD2
INTO                ; 若有溢出, 转溢出处理
MOV AL,BL
MOV AX,4C00H
INT 21H
HLT
```



```
INTO1    PROC           ; 中断溢出处理
          PUSH AX
          MOV BL,0FFH
          POP AX
          IRET          ; 中断返回
INTO1    ENDP
          ;
CODE     ENDS
          END START
```


举例二：硬件中断

下图是一个接口电路，用中断传送方式使8255A的A口的发光二极管依次发光，要求每个二极管在一个循环中发光1S。设8255A端口地址为80H - 86H，8253端口地址为88H - 8EH，主8259端口地址为90H - 92H，从8259端口地址为98H - 9AH



-
- ✦ 8255A的A口工作方式可选为方式0的输出，而8253通道0的工作方式可定为方式0，即计数结束产生中断信号。
 - ✦ 8255A的方式控制字：10001011B。即A输出，B口输入，C口输入。
 - ✦ 8253通道0的方式控制字：00110001B。即工作方式为方式0，BCD码计数。



✦ 8259设置特点

- ◆ 1、初始化具有顺序性
- ◆ 2、顺序为 ICW1、ICW2、ICW3、ICW4

✦ 从8259设置

- ◆ ICW1: 上升沿触发, 级联
→ 端口98H: 00010001B
- ◆ ICW2: 中断类型码, 端口9AH: 30H
- ◆ ICW3: 主片从片标志, 端口9AH: 00000010B
- ◆ ICW4: 完全中断嵌套方式, 非自动中断结束方式
→ 端口9AH: 00000001B

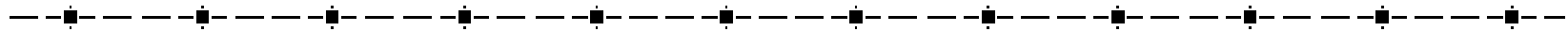




.....
SHIFT_DAT DB 0FEH

.....
CLI ; 关中断
CLD ; 填写中断入口地址表
MOV AX,0
MOV ES,AX
MOV DI,4*31H
MOV AX,OFFSET INT_31H
MOV ES:[DI],AX
INC DI
INC DI
MOV AX,SEG INT_31H
MOV ES:[DI],AX

MOV AL,11H ; 设置主8259
OUT 90H,AL ; 设置ICW1; 上升沿触发, 级联
MOV AL,08 ; 设置ICW2; 中断矢量
OUT 92H,AL
MOV AL,04 ; 设置ICW3; IR2级联
OUT 92H,AL
MOV AL,01 ;设置ICW4;非自动结束中断(清中断服务FF)
OUT 92H,AL
MOV AL,01101011B ; 设置IOW1,开放IR2, IR4, IR7中断
OUT 92H,AL




```
                                ; 设置从8259
MOV AL,11H                       ; 设置ICW1
OUT 98H,AL
MOV AL,30H                       ; 设置ICW2
OUT 9AH,AL
MOV AL,02                        ; 设置ICW3
OUT 9AH,AL
MOV AL,01                        ; 设置ICW4
OUT 9AH,AL
MOV AL,00000000B                 ; 设置IOW1,开放IR0...IR7中断
OUT 9AH,AL
```



```
;  
MOV AL,10001011B  
OUT 86H,AL  
MOV AL,0  
OUT 80H,AL  
MOV AL,00110001B  
OUT 8EH,A  
MOV AL,00  
OUT 88H,AL  
MOV AL,10H  
OUT 88H,AL  
STI  
.....
```

; 设置8255和8253
; 设置8255A
; 设置8253通道0
; 开中断
; 执行其他程序



中断服务子程序：

- ◆ 主要是对8255的A口进行操作，使变量SHIFT_DAT的内容左移，
- ◆ 重新设置8253通道0，为下一次中断信号产生作准备，
- ◆ 对8259A传送一个EOI命令，表示中断程序完成。

```
INT31H      PROC FAR           ; 中断服务子程序
            PUSH AX           ; 保护现场
            PUSHF
            MOV AL,SHIFT_DAT   ; 对8255的A口送数
            OUT 80H,AL
            ROL AL,1          ; 变量SHIFT_DAT左移
            MOV SHIFT_DAT, AL ; 准备下一次数据
```



MOV AL,00110001B ; 重新设置8253

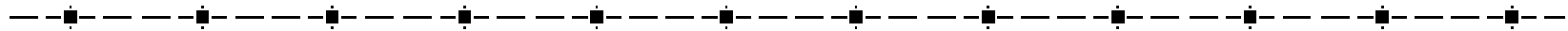
OUT 8EH,AL

MOV AL,00

OUT 88H,AL

MOV AL,10H

OUT 88H,AL

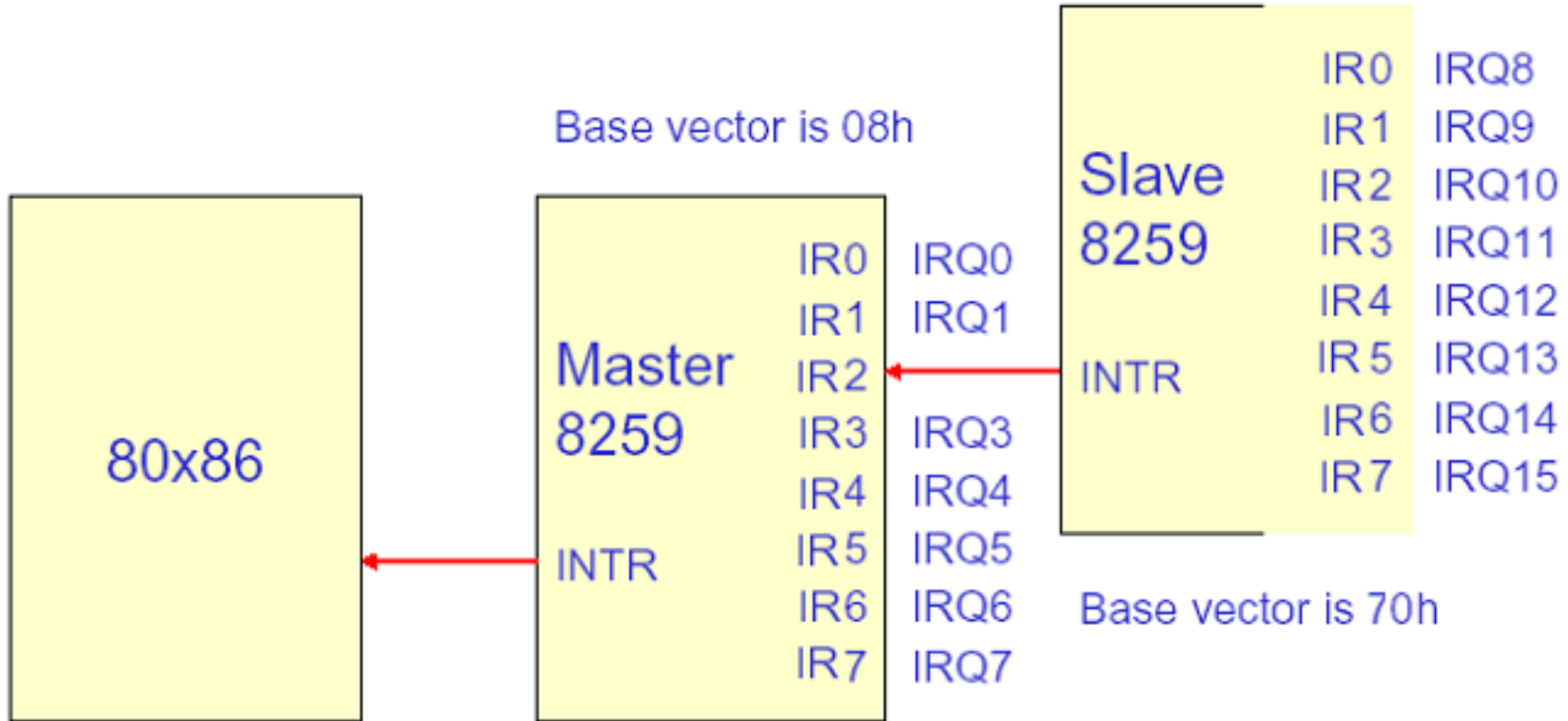
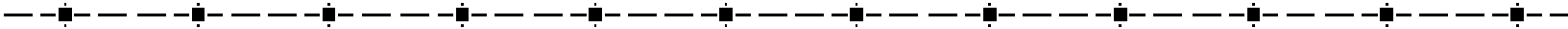


```
INT31H      ; 发EOI命令  
            ; 传送EOI命令给8259  
MOV AL,20H; ; 向从8259发EOI命令  
OUT 98H,AL ; 向主8259发EOI命令  
OUT 90H,AL ;  
            ;  
            ; 恢复现场  
POPF  
POP AX  
IRET      ; 中断返回  
ENDP
```

9.8 IBM-PC机硬件中断

✦ 9.8.1 中断设置

在PC/XT机中，有一片8259A中断控制芯片，可管理8个中断源。在PC/AT机中有两片8259A中断控制芯片，可管理15个中断源。



9.8 IBM-PC机硬件中断

✦ 9.8.1 中断设置

◆ 优先权

主片: IR0, IR1 →

从片: IR0, IR1, IR2, IR3, IR4, IR4, IR5, IR6, IR7 →

主片: IR3, IR4, IR5, IR5, IR7

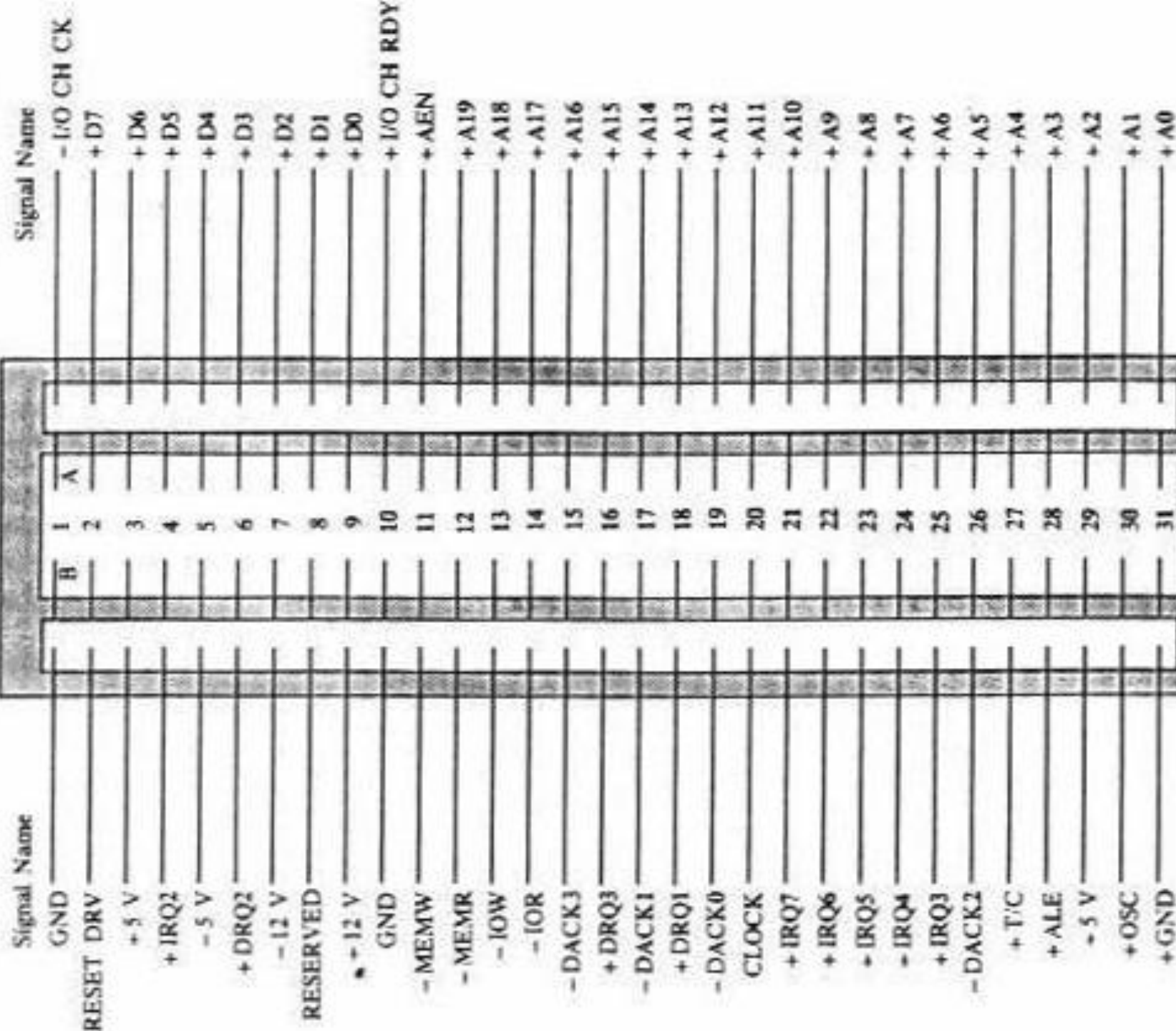
◆ 中断类型号

主片: 00H—07H

从片: 70H—77H

中断引入管脚		中 断 源
主 片	IR ₀	计数器 0 通道输出
	IR ₁	键盘中断
	IR ₂	从片 8259A
	IR ₃	网络通讯
	IR ₄	串行通讯
	IR ₅	硬盘控制
	IR ₆	软盘控制
	IR ₇	打印机控制
从 片	IR ₀	定时时钟
	IR ₁	IRQ ₉ 引入
	IR ₂	IRQ ₁₀ 引入
	IR ₃	IRQ ₁₁ 引入
	IR ₄	IRQ ₁₂ 引入
	IR ₅	IRQ ₁₃ 引入
	IR ₆	IRQ ₁₄ 引入
	IR ₇	IRQ ₁₅ 引入

COMPUTER REAR



COMPUTER FRONT


9.8 IBM-PC机硬件中断

✦ 9.8.2 PC机中断资源的使用

◆ 一、对中断屏蔽寄存器的操作

主片8259A中断屏蔽寄存器的I / O端口地址是21H, 它的8位对应控制8个外部设备, 通过设置这个寄存器的某位为0或为1来允许或禁止某外部设备的中断。某位为0表示允许某种外设中断请求, 某位为1表示某种外设的中断请求被屏蔽(禁止)。

从片8259A中断屏蔽寄存器的I / O端口地址是A1H



例如，只允许键盘中断，可设置如下中断屏蔽字：

```
MOV AL,1111101B
```

```
OUT 21H,AL
```

如果系统重新增设键盘中断，则可用下列指令实现：

```
IN AL,21H
```

```
AND AL,1111101B
```

```
OUT 21H,AL
```

9.8 IBM-PC机硬件中断

★ 9.8.2 PC机中断资源的使用

◆ 二、中断结束命令

在一次中断处理结束之前，还应给8259A可编程中断控制器的中断命令寄存器发出中断结束命令EOI，可采用一般的EOI方式，中断命令寄存器的I/O端口地址为20H，结束外中断用下面的指令：

```
MOV AL, 20H
```

```
OUT 20H, AL
```

同样，如果中断是从片8259A的中断输入线产生的，则要同时对主片和从片的中断命令寄存器发EOI命令。从片8259A中断命令寄存器的I/O端口地址是A0H