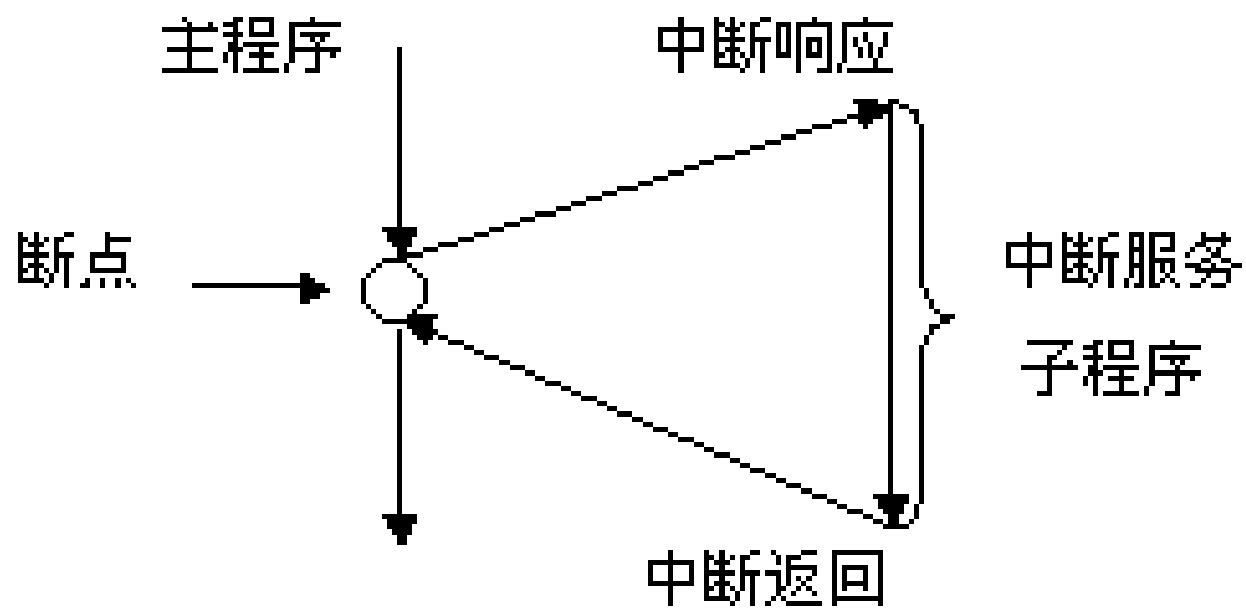


中断技术

第九章

中断基本原理



- Intel系列微处理器的对外的中断引脚包括两个申请中断的硬件引脚（INTR和NMI），一个响应INTR中断的硬件引脚（INTA）
- Intel系列CPU最多包含256个中断向量，其中前5个在8086~Pentium的所有Intel系列的微处理器中是相同的，一般保留前32个为Intel各种微处理器系列成员专用，用户可以使用后面224个向量。

- 保护模式下，把外部中断称为“中断”（Interrupt），把内部中断称为“异常”（Exception）。

中断

- 分为可屏蔽中断和不可屏蔽中断。
- CPU只有一个INTR引脚，外部中断源有很多，因此一般需要中断控制器对外部中断源进行管理，选择优先级最高的中断请求发送到CPU的INTR引脚，常见的中断控制器有8259芯片

异常

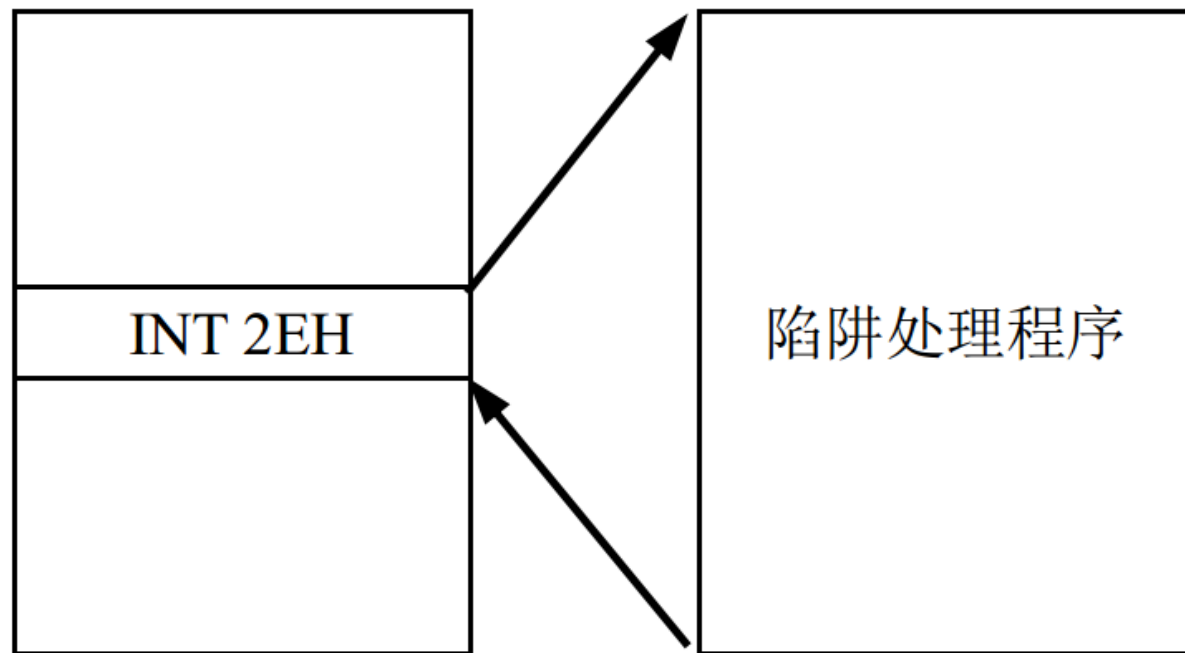
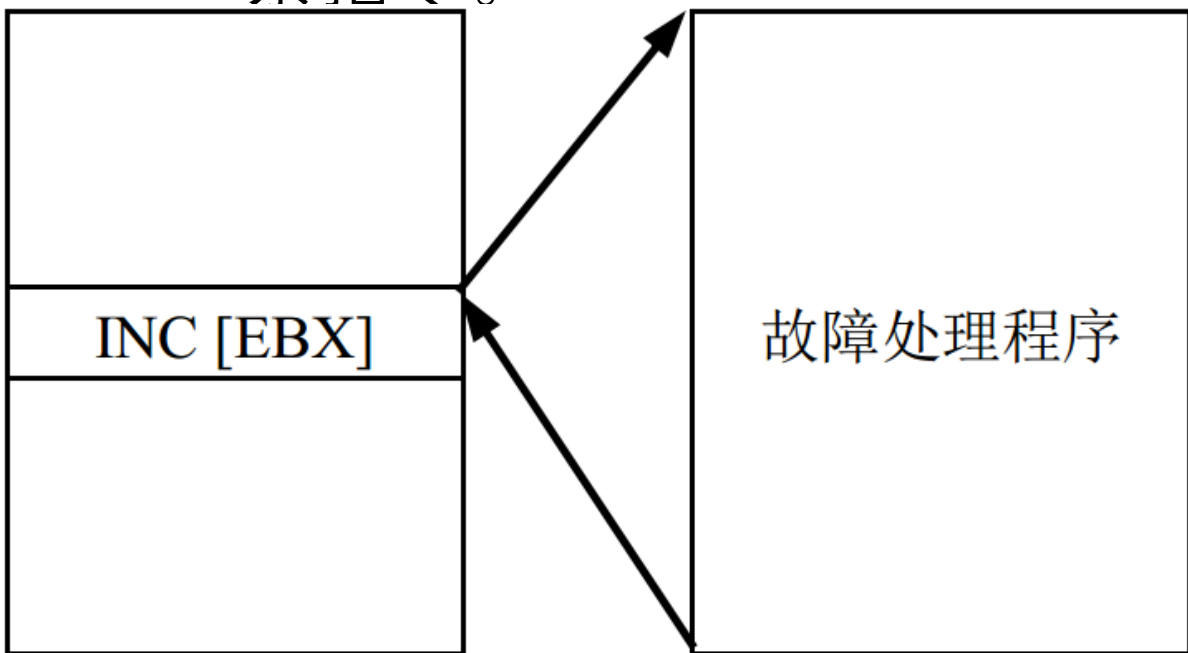
- 异常是不可屏蔽的，每一种异常类别具有不同的异常号码。
- 异常分为故障（Fault）、陷阱（Trap）和中止（Abort）3种

故障

- 故障是在引起异常的指令之前，把异常情况通知给系统的一种情况。故障的特点是可排除的。
- 当控制转移到故障处理程序时，在堆栈中保存的断点CS及EIP的值指向引起故障的指令。这样在故障处理程序将故障排除后，执行IRET返回到引起故障的程序，刚才引起故障的指令可重新得到执行。

陷阱

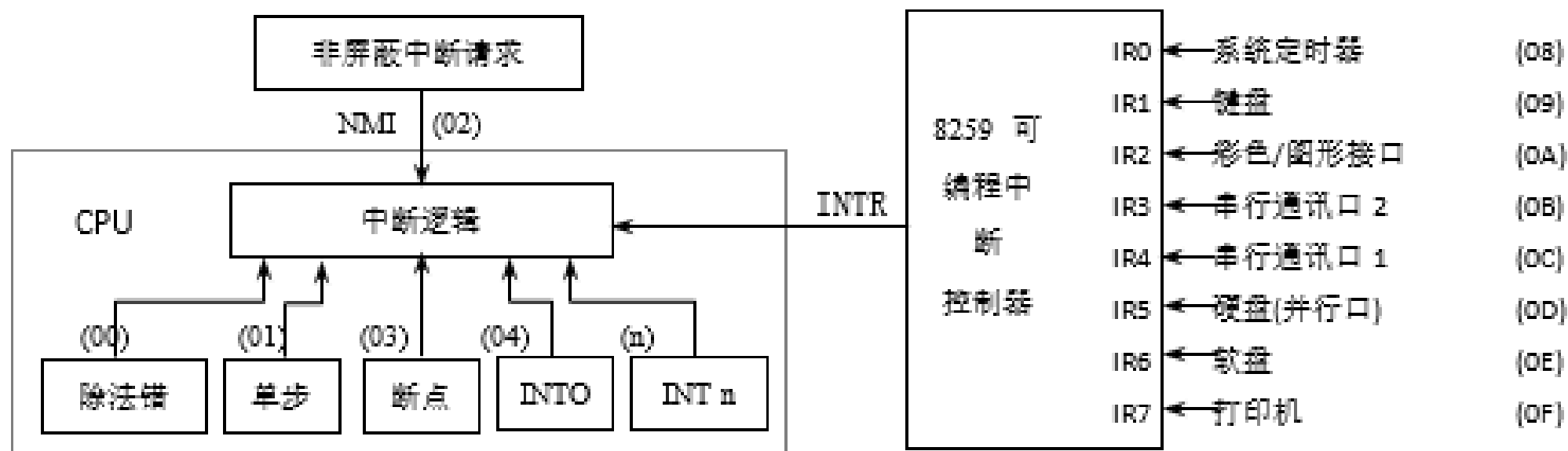
- 陷阱是在引起异常的指令执行之后触发的一种情况。
- 软中断指令“INT n”、单步异常等是陷阱的例子。
- 故障处理完毕后继续执行这一条指令，陷阱处理完毕后执行下一条指令。



中止

- 中止是在系统出现严重的不可恢复的事件时触发的一种异常，产生中止后，正执行的程序不能被恢复执行，系统要重新启动才能恢复正常运行状态。

- NMI和INTR引脚接收外部中断源产生的中断请求，由CPU外部中断源发出的中断请求叫做外部中断，实线框里面列出的是在CPU内部中断源产生的中断，叫做内部中断。



中断服务程序

- 响应中断时，CPU暂停当前正在执行的程序转而执行中断服务程序。中断服务程序包括保护现场、处理中断、发送中断结束命令、恢复现场、中断返回几个部分。

- (1) 中断服务程序入口设置
- 中断服务程序的入口必须设置到中断向量表或者中断描述符表中, 才会被CPU在中断响应时调用。因此, 在初始化时主程序需要设置中断向量, 在设置新的中断向量之前要保存旧的中断向量, 以便恢复。

- (2) DS、ES的赋值
- 要用DS、ES访问程序自己的数据段，那么必须先给它们赋值。可以使用被中断的程序的堆栈。

- (3) 软件中断的返回结果
- 对于某些软件中断（例如INT 21H服务程序），其返回参数保存在寄存器中，这时中断服务程序就不需要在堆栈中保存这些寄存器。

实模式的中断处理

- 当中断发生以后，CPU都能得到一个中断类型号。CPU以中断类型号做为索引，通过查找内存中的中断向量表来寻找中断服务程序的入口。
- 中断向量表（Interrupt Vector Table, IVT）就是各种中断类型的处理程序的地址表。中断向量表实质上就是由程序预先设置好的一块内存区域。256个中断服务程序的入口地址（段地址和偏移量，即中断向量）按中断类型码从小到大顺序存放。

- 8086/8088的中断向量表位于存储器的00000H~003FFH单元，占据1024个字节，这1024个字节被分为256个中断向量，每个中断向量占4个字节。这个4字节的中断向量包含了中断服务程序的段地址和偏移量。

中断处理过程

- 标志寄存器的内容压入堆栈，保护各个标志位。
- 清除中断标志（IF）和陷阱标志（TF），禁止可屏蔽中断INTR引脚和陷阱或单步功能。
- 保存断点；将断点逻辑地址（返回地址）压入堆栈，先将代码段（CS）内容压入堆栈，然后指令指针（IP）内容压入堆栈。
- 从中断类型号乘4的主存地址中取出中断向量内容，送CS和IP。
- 开始执行中断服务程序。
- 当执行到中断服务程序的指令IRET时，IRET指令从堆栈中移出6个字节：IP两个字节，CS两个字节以及FLAGS两个字节。弹出的断点值（返回地址）送CS和IP寄存器，同时弹出标志寄存器，使IF和TF回到中断前的状态。

写中断向量表

- old40 DD ? ;保存旧的中断向量
- New40 PROCFAR ;中断必须被定义为far程序
- ;新的40号中断服务程序的主体
- IRET ;中断服务程序必须有IRET
- New40 ENDP
- ;初始化

DOS系统功能调用

- 除了直接修改中断向量表主存内容之外，还可以调用DOS功能调用INT 21H的25H、35H功能来设置和读取中断向量。
- (1) 中断服务程序入口地址获取
- 功能号：AH=35H
- 入口参数：AL=中断向量号
- 出口参数：ES:BX为中断服务程序入口地址（段基址：偏移地址）

- 利用INT 21H获取40h号中断服务程序入口地址。
- MOV AX, 3540H ;利用DOS功能35H号
- INT 21H ;获取原40H中断服务程序入口地址
- MOV old40, BX ;保存偏移地址
- MOV old40+2, ES ;保存段基地址

中断服务程序入口地址设置

- 功能号：AH=25H
- 入口参数：AL=中断向量号，DS:DX为中断服务程序入口地址（段基址：偏移地址）
- 设置新中断服务程序入口地址。
- PUSH DS
- MOV DX, OFFSET new40 ;取中断程序偏移地址
- MOV AX, SEG new40 ;取中断程序段地址
- MOV DS, AX
- MOV AX, 2540H
- INT 21H
- POP DS

保护模式的中断处理

- 保护模式下使用中断描述符表（Interrupt Descriptor Table, IDT）来存储256个中断向量。
- 中断类型号作为中断描述符表IDT中描述符的索引，取得一个描述符，从中得到中断/异常处理程序的入口地址。
- 单核系统中只有一个中断描述符表IDT，多核系统中每个CPU核都有自己的IDT
- 中断描述符表寄存器IDTR指示IDT在内存中的位置。通过SIDT指令可以获得IDT的地址
- 中断描述符表IDT所包含的描述符只能是中断门、陷阱门和任务门

门描述符格式

15								8	7								0
+0	偏移（位 15~0）																
+2	段选择符（位 15~0）																
+4	P	DPL	S=0	D	1	1	T	00000000									
+6	偏移（位 31~16）																

中断和异常的处理过程

- 1、中断和异常响应
- 如果是异常处理，首先根据异常类型确定返回地址（CS:EIP），对于故障，CS:EIP指向引起故障的指令；对于陷阱，CS:EIP指向引起陷阱的指令的下一条指令。如果有出错代码，就把出错码压入堆栈。为了保证堆栈指针按双字边界对齐，16位的出错码以32位的形式压入，其中高16位的值为0。
- 判断中断类型号要索引的门描述符是否超出IDT的界限。若超出界限，就触发通用保护故障。

- 从IDT中取得对应的门描述符，分解出选择符、偏移量和属性字节，并进行权限相关的检查。
- 根据门描述符类型，分别转入中断或异常处理程序。如果中断类型号所指示的门描述符是中断门或陷阱门，那么控制转移到当前任务的一个处理程序过程，并且可以变换特权级。

跳转到中断服务程序

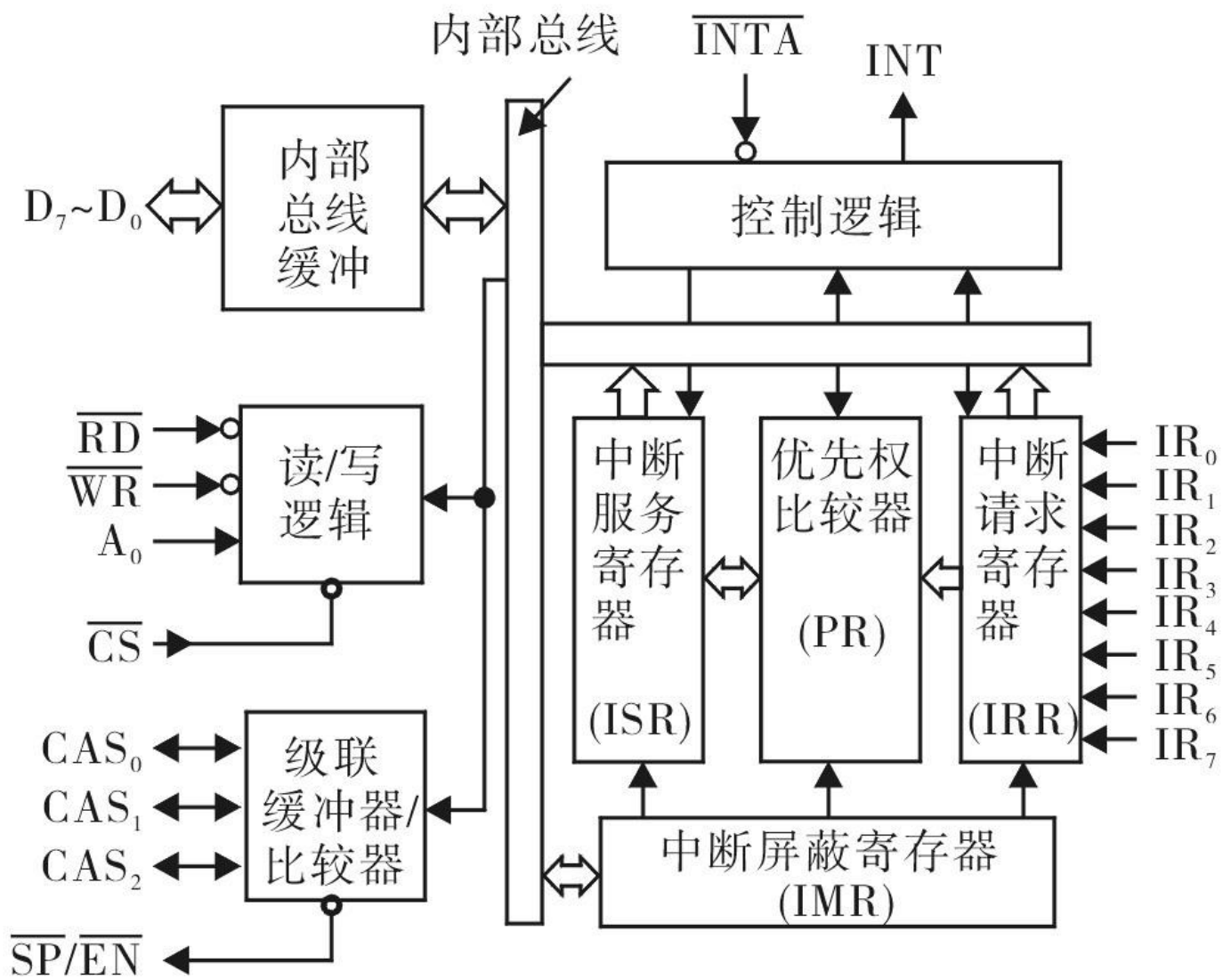
- (1) 通过中断门或者陷阱门的跳转
- 使用中断门或陷阱门实现由中断/异常处理程序进行处理，不进行任务切换，运行环境在当前的任务中
- (2) 通过任务门的跳转
- 如果以中断类型号为索引，在中断描述符中取出的是一个任务门描述符，那么控制将转移到新的任务。将任务门放在IDT表中，在响应对应的中断或异常时，可根据该任务门实现任务的自动调度。

中断或异常处理后的返回

- 中断返回指令IRET用于从中断或异常处理程序中返回。该指令的执行根据任务嵌套标志NT位是否为1分为两种情形。由任务门转入中断或异常处理程序时，NT位被置1；由中断门或陷阱门转入中断或异常处理程序时，NT位被清0。
- NT位为1时，IRET执行的是嵌套任务的返回。当前TSS中的链接字段保存前一任务的TSS的选择符，取出该选择符进行任务切换就完成了返回。这种情形在通过任务门转入的处理程序返回时出现。
- NT位为0时，IRET执行的是当前任务内的返回。这种情形在通过中断门或陷阱门转入的处理程序返回时出现。○

可编程中断控制器8259

- 一片8259可以管理8个硬件中断源，最多支持64个中断源，包括1个主片和8个从片。每一个中断源的请求都可以通过程序进行屏蔽或者许可。在中断响应周期，8259提供中断源的中断类型号。8259有多种工作方式，可以通过编程来选择设置。



外部引脚及功能

- (1) 面向CPU的信号14个
- D7 ~ D0: 三态8位双向数据总线，与CPU数据总线连接，完成命令、状态信息的传送，中断类型号也是由数据缓冲器送到CPU。
- CS#: 片选使能输入信号，低电平有效。
- WR#: 接CPU的写选通信号 (IOW#)，写控制输入信号。
- RD#: 接CPU的读选通信号 (IOR#)，读控制输入信号。
- A0: 地址信号，接CPU的地址总线，用来对8259内部不同命令字进行选择。
- INT: 中断输出引脚，高电平有效。主8259的INT与微处理器的INTR相连，从8259的INT接主片8259的IR引脚。
- INTA#: 中断响应，与微处理器的INTA#引脚相连。

- (2) 面向外设的信号8个
- $IR_7 \sim IR_0$: 8个中断请求输入信号, 高电平或上升沿有效(可编程决定), 用于接收外部设备(中断源)的中断请求。在电平触发方式下, IR引脚变为高电平, 表示向CPU申请中断。CPU响应中断后, 必须将IR恢复为低电平, 否则会引起第2次重复中断。在边沿触发方式下, IR引脚从低电平变为高电平时, 表示向CPU申请中断。对于外设来说, 实现这种方式比较简单, 没有重复中断的问题。

- (3) 面向同类芯片的信号4个
- CAS2 ~ CAS0: 级联信号。由多片8259构成的主从结构中, 只有一个主片, 一个或多个从片, 从片最多有8个。主片和从片的CAS2 ~ CAS0全部对应相连, 在中断响应时主片发送从片的标识码(0~7)。在第2个INTA#脉冲期间, 只有标识码匹配的从片才把中断类型码送至数据总线。
- SP#/EN#: 主从/使能信号。当8259工作在缓冲方式时, SP#/EN#是输出信号, 用作数据总线缓冲器的使能信号(EN#), 即用它来控制数据收发器的工作; 当8259工作在非缓冲方式时, SP#/EN#是输入信号, 用来指明该8259是主片还是从片。SP#/EN#=0时, 8259为从片; =1时, 为主片。8259是否工作在缓冲方式也是由CPU编程来决定的。

内部模块及功能

- (1) 数据总线缓冲器
- 8位的双向三态缓冲器与CPU数据总线D7 ~ D0连接，完成命令、状态信息的传送，中断类型号也是由数据缓冲器送到CPU。
- (2) 读写控制逻辑
- 接收来自CPU的读写命令，完成规定的操作。操作过程由CS#、A0、RD#、WR#等输入信号共同控制。在CPU写8259时，把数据送至相应的命令寄存器中。在CPU读8259时，将相应寄存器的内容输出到数据总线上。

- (3) 级联缓冲/比较器
- 级联缓冲/比较器用于实现多个8259之间的级联，使得中断源由8个扩展至最多64个。
- (4) 控制逻辑
- 控制逻辑按初始化设置的工作方式控制8259的全部工作。该电路可根据中断请求寄存器的内容和优先权判断结果向CPU发中断请求信号INT，并接受CPU发回的中断响应信号INTA#，使8259进入中断服务状态。

- (5) 中断请求寄存器
- IRR是与外部接口的中断请求线相连的寄存器，请求中断处理的外设通过 $IR_0 \sim IR_7$ 向8259请求中断服务，并把中断请求信号锁存在中断请求寄存器中。有中断请求发生时，对应位置为1，同时有多个中断源发出中断请求时，IRR里面会有多个1，如当引脚 IR_0 、 IR_1 上有中断请求发生时，IRR的内容为03H。IRR的内容为0FH表示引脚 $IR_0 \sim IR_3$ 上有中断请求输入， $IR_4 \sim IR_7$ 引脚上没有中断请求输入等。
- (6) 中断屏蔽寄存器
- IMR用来设置中断请求的屏蔽信息。当IMR中某一位设为1时，8259就屏蔽对应IR引脚发出的中断请求信号，不向CPU发送该中断源的中断请求。如设置IMR中断屏蔽字为FF，表示屏蔽所有8259芯片8个IR引脚的中断请求输入。

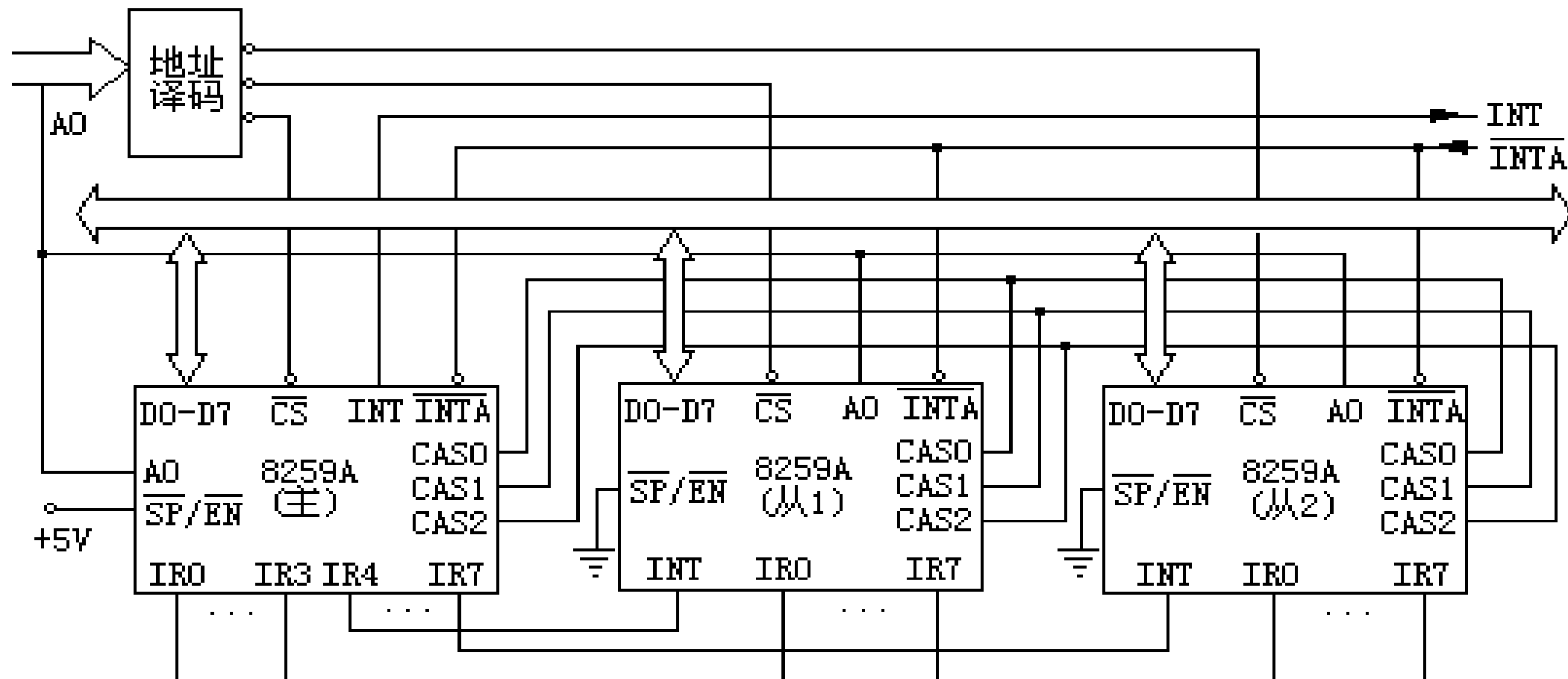
- (7) 中断服务寄存器
- ISR用于存放当前正在进行处理的中断源。ISR中 D_i 为1表示 IR_i 正在服务，为0表示没有被服务。ISR中置1的位在中断响应时由8259清除，也可以由CPU向8259发送中断结束命令来清除，具体的清除方式可以由程序控制。
- (8) 优先权电路
- 检测到来自IR引脚的中断请求后，优先权电路负责检查该中断源的优先级，并与ISR中记录的中断（正在服务的中断）进行比较，以确定是否将这个中断请求送给CPU。假定它比正在服务中的中断具有更高的优先级，则PR就使INT线变为高电平，送给CPU，提出中断申请，并在中断响应时将它记入ISR的对应位中。如果它等于或低于正在服务中的中断优先级，则PR不为其提出申请，直到ISR中比它优先级高的位被清除。

8259中断过程

- 单片8259的中断处理过程为：当一条或多条中断请求线IR0 ~ IR7变高时，设置相应的IRR位为1；然后PR对中断优先权和中断屏蔽寄存器的状态进行判断之后，如果某中断优先权最高且为允许中断状态，就向CPU发高电平中断请求信号INT，请求中断服务；要注意与NMI不同，CPU的可屏蔽中断INTR是电平触发的，因此INT请求时必须保持高电平直到中断申请被CPU识别为止。

- CPU响应中断时，送出中断响应信号INTA#。CPU响应中断时会发出两个INTA#。8259接到来自CPU的第一个INTA#信号时，将当前中断服务寄存器中相应位置位，并把IRR中相应位复位。同时，8259准备向数据总线发送中断类型码。在第二个INTA#负脉冲期间，中断类型码被读入CPU，如果是在AEIOI（自动结束中断）方式下，在第二个INTA#负脉冲结束时8259会复位ISR的相应位。在非自动中断结束方式下，ISR相应位要由中断服务程序结束时发出的EOI命令来复位。

8259的级联



- CPU的中断应答信号INTA#被连接到所有的8259上，主片收到INTA#后，确认是从它的IR₄产生的。由于IR₄连接在从片上，所以主片把100B（编号为4）送到CAS2~CAS0上。从片1和从片2都收到了INTA#，它们检查CAS2~CAS0上的信号与它自己的标示码是否一致。

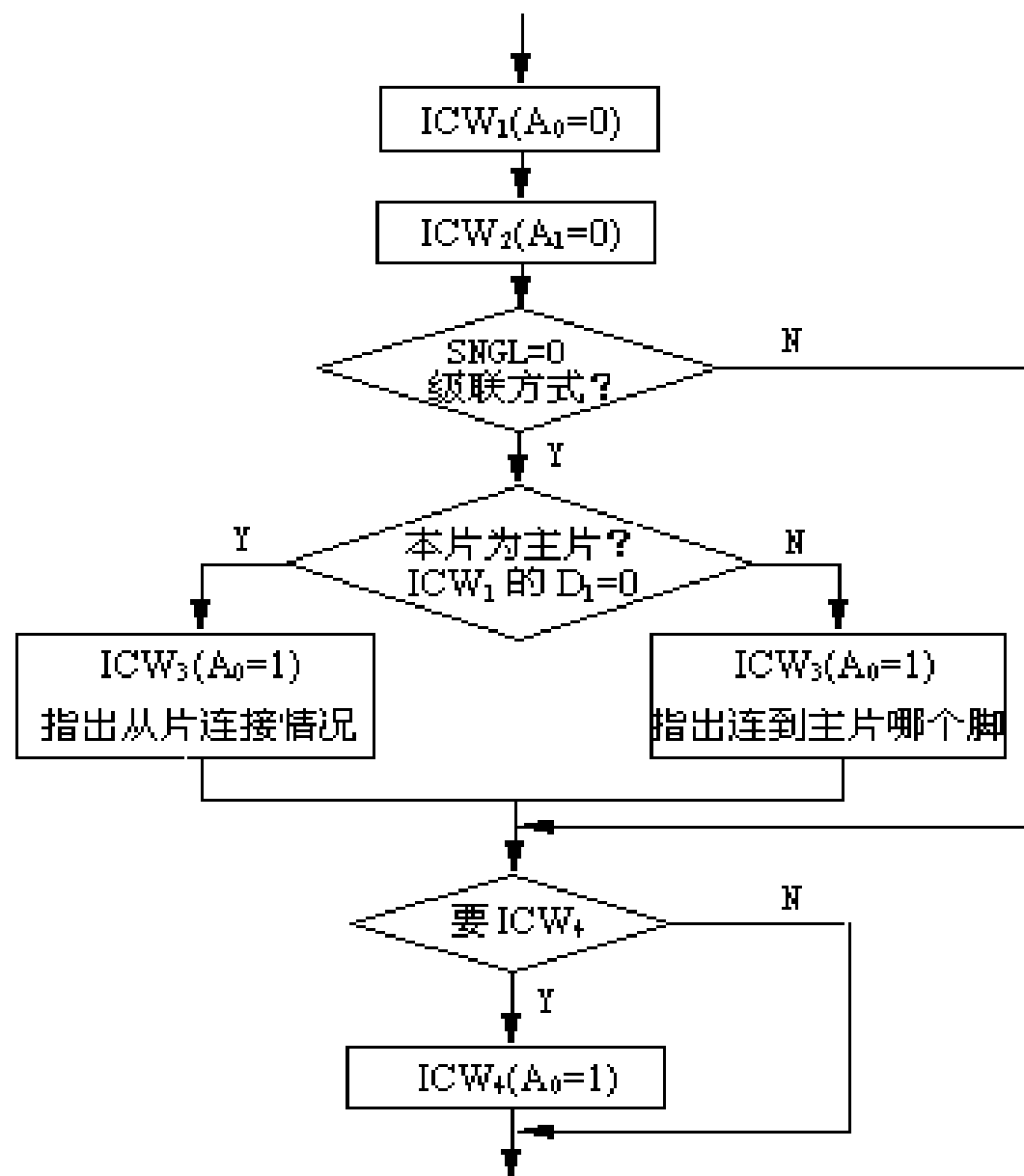
8259的编程

- CPU的命令字分两类：初始化命令字（ICW1 ~ ICW4）和操作命令字（OCW0 ~ OCW4）。
- 初始化命令字（Initialization Command Word, ICW）往往是在系统启动时，由初始化程序（BIOS或操作系统）来设置的。初始化命令字一旦设定，一般在系统工作过程中就不再改变。
- 操作命令字（Operation Command Word, OCW）则是在计算机系统运行过程中，由CPU利用这些控制字来控制8259执行不同的操作，如中断屏蔽、中断结束、优先权循环和中断状态的读出和查询等。OCW可在初始化之后的任何时刻写入8259，并可多次设置。

- 8259只有一个地址线A0，一般将A0与CPU的A0相连接，此时A0=0时就是偶地址，A0=1时就是奇地址。

CS#	WR#	RD#	A0	读写操作
0	0	1	0	写 ICW1、OCW2、OCW3
0	0	1	1	写 ICW2、ICW3、ICW4、OCW1
0	1	0	0	读 IRR、ISR、查询字
0	1	0	1	读 IMR

初始化命令字



7	6	5	4	3	2	1	0
D7	D6	D5	1	LTIM	D2	SNGL	IC4

D7~D5	当 8259 与 8086/8088/Pentium 连接时，此位无意义。
D4	必须为 1。表示这是一个 ICW1 命令字。
LTIM	=0，边沿触发；=1，电平触发。
D2	无意义。
SNGL	=1，系统中只有 1 片 8259，单片使用；=0，多片 8259 级联使用。
IC4	=0，不需要写入 ICW4；=1,需要写入 ICW4。

- 8088系统中，使用单片8259，中断请求信号为上升沿触发，需要设置ICW4，该片8259的端口地址为20H和21H，则ICW1应为多少，请写出初始化程序。
- ICW1为：00010011B = 13H
- 设置ICW1的指令为：
- MOV AL, 13H
- OUT 20H, AL

7	6	5	4	3	2	1	0
T7	T6	T5	T4	T3	0	0	0

T7~T3	中断响应码的高 5 位。
-------	--------------

- 假设系统中使用单片8259，该片8259的端口地址为20H和21H，8个中断源的中断类型码为08H~0FH(00001000B~00001111B)，则应如何初始化ICW2？
- 初始化命令字ICW2应为：00001000b = 08H
- 设置ICW2的指令为：
- MOV AL, 08H
- OUT 21H, AL

7	6	5	4	3	2	1	0
S7	S6	S5	S4	S3	S2	S1	S0
S7~S0	Si 位等于 0 时，IRi 不连接从片。等于 1 时，IRi 连接从片。i=0~7。						

图 9-9 主片 ICW3 的格式

7	6	5	4	3	2	1	0
0	0	0	0	0	ID2	ID1	ID0
ID2~ID0	从片的标识码(0~7)，即从片连接到主片的 IRi。i=0~7。						

图 9-10 从片 ICW3 的格式

- 在PC系统中，使用两片8259，主片8259的端口地址为20H和21H，从片8259的端口地址为0A0H和0A1H，从片8259的INT连接到主片的IR₂上。则应如何初始化ICW3？
- 主片的IR₂接从片，则主片ICW3为：00000100B=04H，从片ICW3为：00000010B=02H。
- MOV AL, 04H ;主片ICW3初始化
- OUT 21H, AL
- MOV AL, 02H ;从片ICW3初始化
- OUT 0A1H, AL

7	6	5	4	3	2	1	0
0	0	0	SFNM	BUF	M/S	AEOI	uPM

SFNM	=0，普通全嵌套方式。=1 时，特殊全嵌套方式。
BUF	=0，非缓冲模式。=1，缓冲模式。
M/S	=0，从片。=1，主片。BUF=0 时，此位无意义。
AEOI	=0，非自动结束方式；=1，中断自动结束方式。
uPM	=0，用于 8080/8085 等 8 位 CPU 系统；=1，用于 8088/8086/Pentium。

- D4: SFNM (Special Fully Nested Mode) 位等于0时, 工作在普通全嵌套方式; 等于1时, 8259工作于特殊全嵌套方式。普通全嵌套方式是8259初始化后自动进入的优先级管理方式。该方式下, IR引入的中断具有固定的优先级: 优先级从 $IR_0 \rightarrow IR_7$ 依次降低。若当前正在处理 IR_i 中断, PR允许比 IR_i 优先级高的中断进行中断嵌套。
- 在特殊全嵌套方式下, 当微处理器正在处理某一级中断请求时, 不但允许较高级的中断实现嵌套, 也允许平级中断实现嵌套。特殊全嵌套通常用在8259级联使用的场合, 主片工作在特殊全嵌套方式, 从片工作在普通全嵌套方式。

- D3: BUF位等于1时, 8259工作于缓冲模式, SP#/EN#管脚作为输出, 控制数据总线传输方向。
- D2: M/S只有在BUF位=1即缓冲模式才有意义, 这时主片、从片的选择要靠CPU来指定。

- D1: AEOI=1时, 为中断自动结束方式。在这种方式下, 当第2个中断响应负脉冲INTA#结束时, 将中断服务寄存器ISR的相应位清零, 不需要另外发送EOI (End of Interrupt)。这是最简单的中断结束方式。但是这种方式的使用是有前提的, 即在该中断结束之前不会产生较低级的中断。
- 在非自动结束方式下, 即普通中断结束方式下, 中断处理程序的IRET之前, 微处理器需向8259发送一个普通中断结束命令EOI (通过写OCW2实现), 使ISR中最高优先级的置1位清除, 以结束当前正在处理的中断。

- 在循环优先级方式下的中断结束处理时，8259无法用固定的优先级顺序判断该将哪一级中断结束，这时就不能用普通EOI结束中断。这时必须采用特殊中断结束方式。在特殊中断结束命令中，指出结束中断处理的ISR_i位的i值。通过程序写OCW₂完成。
- 在级联系统中，一般不使用中断自动结束方式，而用非自动结束方式。
- 不管是使用普通中断结束方式，还是使用特殊的中断处理方式，在中断服务程序结束时，都必发发出两次中断结束命令，一次是对主片，一次是对从片。

- D0: $\mu\text{PM}=0$ 时, 表示8259用于8080/8085等8位CPU机组成的系统。 $\mu\text{PM}=1$ 时, 表示8259用于8088/8086、Pentium等16、32位CPU组成的系统。一般应设置 $\mu\text{PM}=1$ 。
- 8086系统中, 假定包含两片8259。主片地址为20H和21H, 从片的地址为A0H和A1H; 两片都工作在特殊嵌套方式、非缓冲模式、采用非自动中断结束。试写出主片和从片的ICW4初始化程序。
- 主片和从片的ICW4均为00010001B=11H。
- MOV AL, 11H
- OUT 21H, AL
- OUT 0A1H, AL

操作命令字

7	6	5	4	3	2	1	0
M7	M6	M5	M4	M3	M2	M1	M0

M7~M0	M _i 位等于 0 时，该中断源 IR _i 不会被屏蔽。等于 1 时，IR _i 被屏蔽。
-------	---

- 8259的端口地址为20H和21H， 试编写程序屏蔽IR2、IR5两个中断源。
- OCW1=00100100B=24H
- MOV AL, 24H
- OUT 21H, AL

7	6	5	4	3	2	1	0
R	SL	EOI	0	0	L2	L1	L0

R	=1，优先级循环。
SL	=1，特定优先级，L2~L0 有效。=0 时，L2~L0 不起作用。
EOI	=1，中断结束命令，使中断服务寄存器 ISR 中的某一位清 0。
L2~L0	三位二进制编码，代表 0~7 一共 8 种中断源。

最高3位一共8种组合：

- 000B：在自动中断结束方式下，不使用循环优先级。优先级顺序始终为 IR_0 （最高）、 IR_1 、 IR_2 、 IR_3 、 IR_4 、 IR_5 、 IR_6 、 IR_7 （最低）。
- 001B：普通EOI命令（不使用L2~L0）。清除ISR中优先级最高的位。
- 010B：保留。
- 011B：特殊EOI命令（使用L2~L0）。清除ISR中的某一位（由L2~L0决定）。

- 100B: 在自动中断结束方式下, 使用循环优先级。某一个IR请求被响应后, 它自动变为最低优先级。例如, IR_2 请求被响应后, 优先级变为: IR_3 、 IR_4 、...、 IR_0 、 IR_1 、 IR_2 。
- 101B: 普通EOI命令 (不使用L2~L0), 使用循环优先级。和3位为000B相同, 但优先级自动循环。
- 110B: 指定L2~L0为最低优先级。
- 111B: 特殊EOI命令 (使用L2~L0), 使用循环优先级。和3位为001B相同, 但优先级自动循环。

- 8259地址为20H和21H，编写程序完成如下操作：
- ①：清除IR₂对应的ISR
- ②：设置IR₄为最高优先级
- 清除IR2对应的ISR，应该向偶地址即20H，写入OCW2控制字。
- OCW2控制字为：01100010B=62H
- MOV AL, 62H
- OUT 20H, AL

- 设置某 IR_i 为最高优先级，只需要采用循环优先级，设置前一个 IR_{i-1} 为最低优先级即可。因此指定 IR_3 为最低优先级， $OCW2=11000011B=0C3H$ ，那么此时优先级由高到低的顺序为： IR_4 、 IR_5 、 IR_6 、 IR_7 、 IR_0 、 IR_1 、 IR_2 、 IR_3 。
- `MOV AL, C4H`
- `OUT 20H, AL`

7	6	5	4	3	2	1	0
0	ESMM	SMM	0	1	P	RR	RIS

ESMM SMM	ESMM=1 且 SMM=0 时，退出特殊屏蔽方式。 ESMM=1 且 SMM=1 时，进入特殊屏蔽方式。 ESMM=0 时，SMM 位无效
P	=1 时，执行中断查询命令。
RR	=0 时，RIS 位无效。=1 时，由 RIS 来确定读取 IRR 还是 ISR。
RIS	RR=1 且 RIS=0 时，下一次读取的是 IRR(中断请求寄存器)。 RR=1 且 RIS=1 时，下一次读取的是 ISR(中断服务寄存器)。

特殊屏蔽方式

- 8259可以通过OCW1命令将中断屏蔽寄存器IMR的相应位置“1”，实现普通屏蔽方式。此时当IR引脚上有中断请求产生时，只有IMR相对应的位不为1时，中断请求才被送入。对于固定优先级情况，只允许高级中断打断当前的服务。
- 如果要允许优先级低的中断进入，则需要采用特殊屏蔽方式。在特殊屏蔽方式下，8259用OCW1写入屏蔽寄存器时，只有屏蔽位等于1的中断源才会被屏蔽，而其它的屏蔽位等于0的中断源发出的请求都会被响应，即使这些中断源的优先级小于ISR中正在服务的优先级。

查询命令

7	6	5	4	3	2	1	0
I	0	0	0	0	W2	W1	W0

I	=0 时，没有中断请求。=1 时，有中断请求。
W2W1W0	有效中断请求(IR0~IR7)中优先级最高的中断源的编号。

读取IRR或ISR

- 8259内部有3个寄存器（IRR、ISR、IMR）可供CPU读出，CPU在发读命令之前，须先指定读取哪个寄存器，然后再发IN指令，才能读取IRR和ISR中的内容。
- 编写程序段读取8259中IRR和ISR的值。
- OCW3=00001010B=0AH或00001011B=0BH写入8259偶地址，然后对8259执行读操作（A0=0）可以获得IRR和ISR值。
- MOV AL, 0BH ;读取ISR寄存器
- OUT 20H, AL
- IN AL, 20H
- MOV AL, 0AH ;读取IRR寄存器
- OUT 20H, AL
- IN AL, 20H

8259初始化举例

- 要求：两片8259级联使用，从片连接在主片的IR₂引脚，主片端口地址为20H，21H，从片端口地址为A0H，A1H，要求主片中断向量号设置为20H~27H，从片中断向量号设置为28H~2FH。中断向量采用边沿触发的方式，主片采用特殊嵌套方式，从片采用普通嵌套方式，仅仅开启定时中断，屏蔽其他中断。

• MOV	AL, 11H	
• OUT	20H, AL	;主8259, ICW1
• OUT	0A0H, AL	;从8259, ICW1
• MOV	AL, 020H	;IRQ0 对应中断向量 0x20
• OUT	021H, AL	;主8259, ICW2
• MOV	AL, 028H	;IRQ8 对应中断向量 0x28
• OUT	0A1H, AL	;从8259, ICW2
• MOV	AL, 04H	;IR2 对应从8259
• OUT	021H, AL	;主8259, ICW3

- MOV AL, 02H ;对应主8259的 IR2
- OUT 0A1H, AL ;从8259, ICW3
- MOV AL, 11H
- OUT 021H, AL ;主8259, ICW4
- MOV AL, 01H
- OUT 0A1H, AL ;从8259, ICW4
- MOV AL, 11111110B ;仅仅开启定时器中断
- OUT 021H, AL ;主8259, OCW1
- MOV AL, 11111111B ;屏蔽从8259所有中断
- OUT 0A1H, AL ;从8259, OCW1

- 8259A的IRQ0（向量号为08H）中断请求来自定时器8253，每隔55ms产生一次
- 本程序的08H号中断服务程序，每次中断显示一串信息，显示10次
- 用内存单元（共享变量）在主程序与外部中断服务程序之间传递参数：中断次数
- 显示信息也安排在共同的数据段中。

数据段

- intmsg db 'A 8259A Interrupt !'
- db 0dh,0ah,0
- counter db 0

- ;保存原中断向量mov ax,3508h
- int 21h
- push bx ;保存偏移地址
- push es ;
- ;设置新中断向量
- cli ;关中断
- push ds
- mov dx,offset new08h
- mov ax,seg new08h
- mov ds,ax
- mov ax,2508h
- int 21h
- pop ds

- in al,21h
- push ax
- and al,0feh ;允许IRQ0
- out 21h,al
- mov counter,0 ;设置中断次数初值
- sti ;开中断

主程序中

- start1: cmp counter,10
- jb start1 ;中断10次退出
- cli ;主程序退出部分
- pop ax
- out 21h,al
- pop dx
- pop ds
- mov ax,2508h
- int 21h
- sti
- mov ax,4c00h
- int 21h

中断服务程序

- new08h proc
- sti ;开中断
- push ax ;保护寄存器
- push bx
- push ds
- mov ax,@data
- mov ds,ax ;设置数据段DS

- ;中断处理
- inc counter
- mov si,offset intmsg ;显示信息
- call dpstri
- ;退出部分
- mov al,20h
- out 20h,al
- pop ds ;恢复寄存器
- pop bx
- pop ax
- lret ;中断返回
- new08h endp

- ;显示字符串子程序
- dpstri proc
- push ax
- push bx
- dps1: lodsb
- cmp al,0
- jz dps2
- mov bx,0
- mov ah,0eh
- int 10h
- jmp dps1
- dps2: pop bx
- pop ax
- ret
- dpstri endp

- mov bx,0
- mov ah,0eh
- int 10h
- jmp dps1
- dps2: pop bx
- pop ax
- ret
- dpstri endp