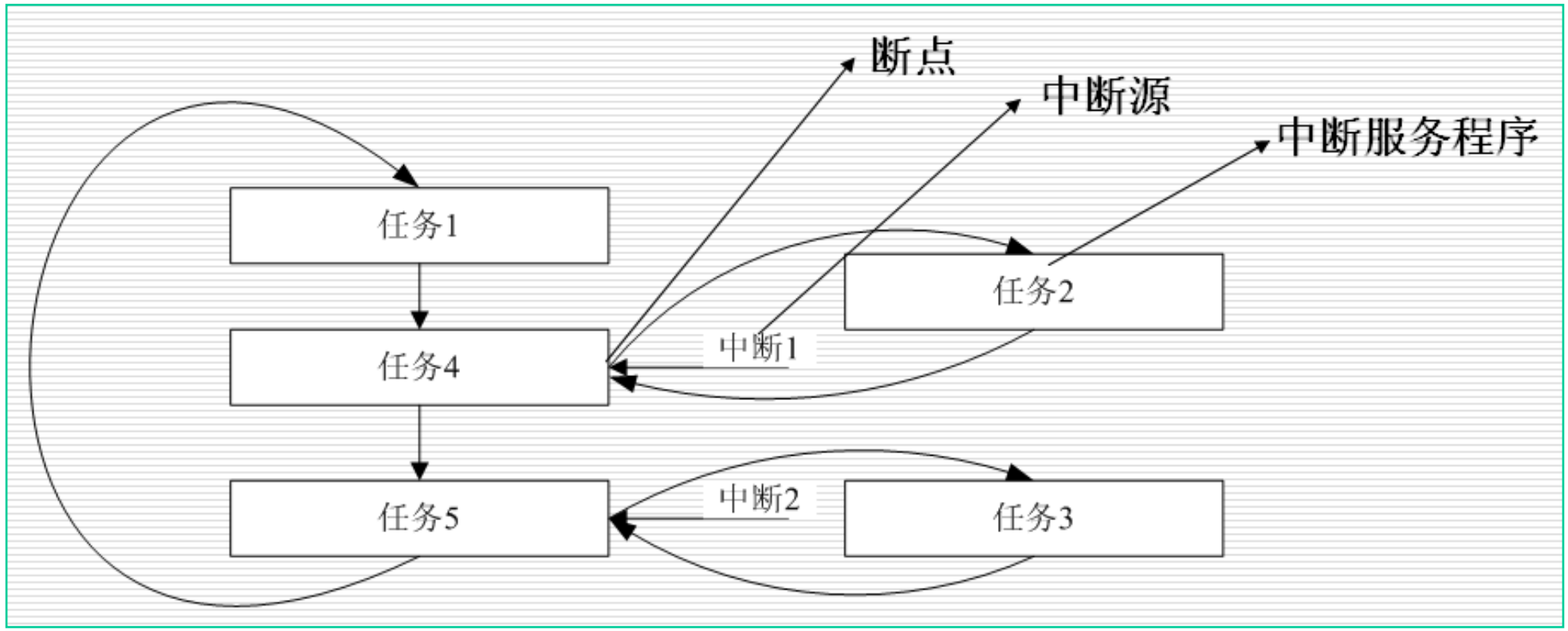
第8章 中断系统和中断控制器8259A

8.1 80X86中断系统

- 中断操作和中断系统

计算机的 CPU 在正常运行程序时，由于内部或外部某个紧急事件的发生，使CPU暂停正在运行的程序，在执行完当前指令后，转去执行请求中断的那个外设或事件的中断服务(处理)程序；待处理完后再返回被中断的程序，继续执行正常的程序。



中断请求

- **中断源**（发出请求的外部设备）需要CPU为其服务时，中断源将发出的“中断请求”信号，CPU被动响应外设要求。

基本中断源有：数据输入输出外设请求中断、定时时间到中断、故障报警中断、程序调试中断

- **中断屏蔽**

中断请求是否能够提出，通常在中断接口部件中有屏蔽电路，如果没有屏蔽，则传出中断请求信号；如屏蔽，则无法传出。

- **中断判优**

如果有多个中断源提出中断请求，则由中断接口部件的判优电路进行优先级判优，以决定高优先级的中断请求信号提出。

■ 中断响应

CPU根据中断申请的性质和执行程序情况，如果响应中断，则给设备发出一个中断应答信号，进入中断响应。

中断响应的主要任务：

- 关中断
- 保存断点和关键现场
- 形成入口地址（将指令指针指向中断服务程序的入口），程序转向中断服务程序

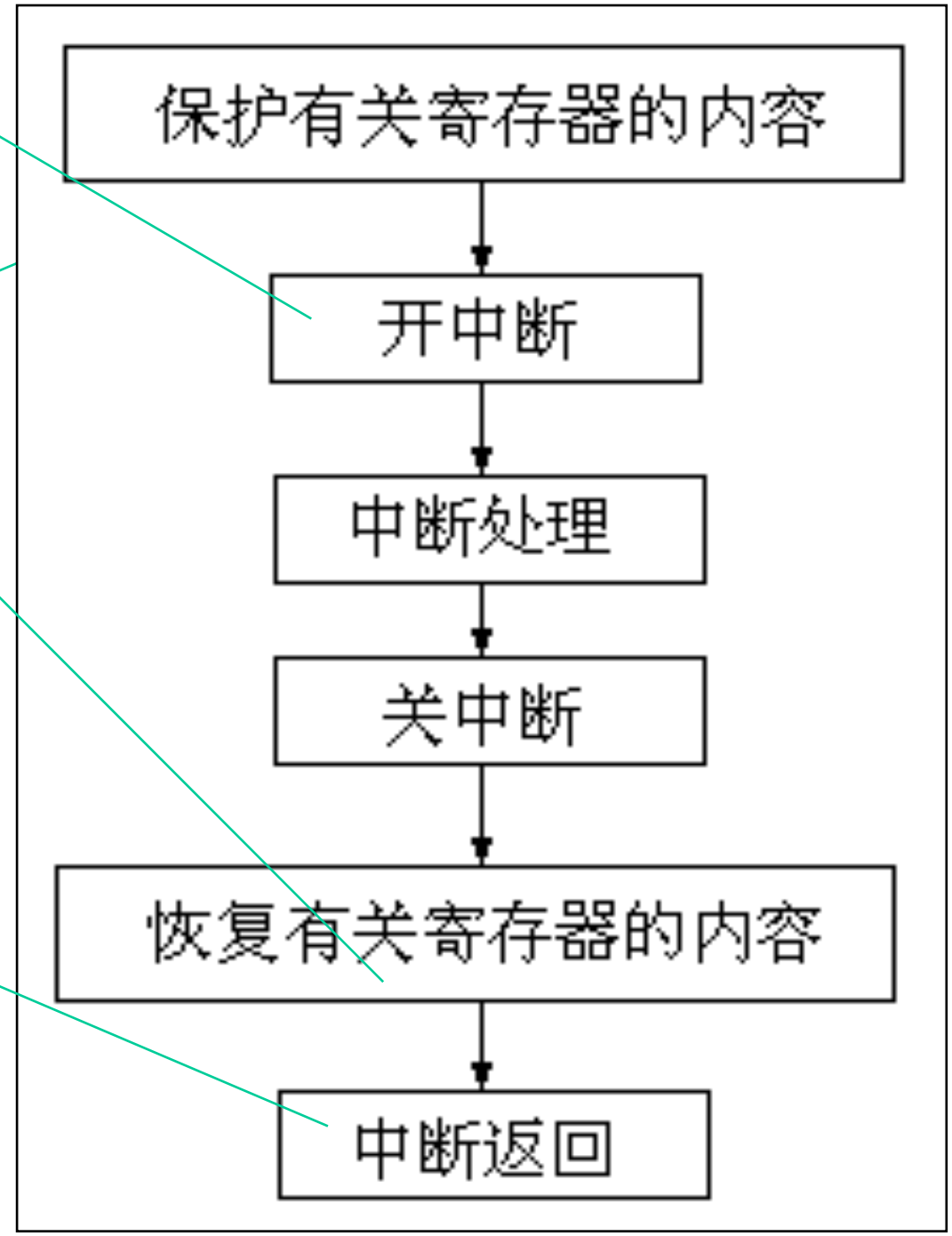
中断处理

CPU响应中断，转入响应的中断服务程序，中断处理完成的工作：

IF=1
表示在该服务程序运行过程中，该可以被中断而运行更高级别的其他中断服务程序

IF=0
恢复现场时，不允许再响应中断，以保证恢复的有序和正确

使用中断返回指令，结束中断服务，返回中断前的原程序



更详细的程序框架

```
入口地址：PUSH  AX          ；保护现场
          PUSH  BX
          PUSH  CX
          PUSH  DX
          PUSH  SI
          PUSH  DI
          PUSH  SP
          PUSH  BP
          STI               ；开中断
ZDFWCX:   …                 ；中断处理
          …
          CLI               ；关中断
          POP  BP           ；恢复现场
          POP  SP
          POP  DI
          POP  SI
          POP  DX
          POP  CX
          POP  BX
          POP  AX
          STI               ；开中断
          IRET              ；中断返回
```

8.1.2 80x86中断指令

- 开中断STI与关中断CLI

CLI：关闭CPU中断(将标志寄存器FLAGS的IF位复位为0)，

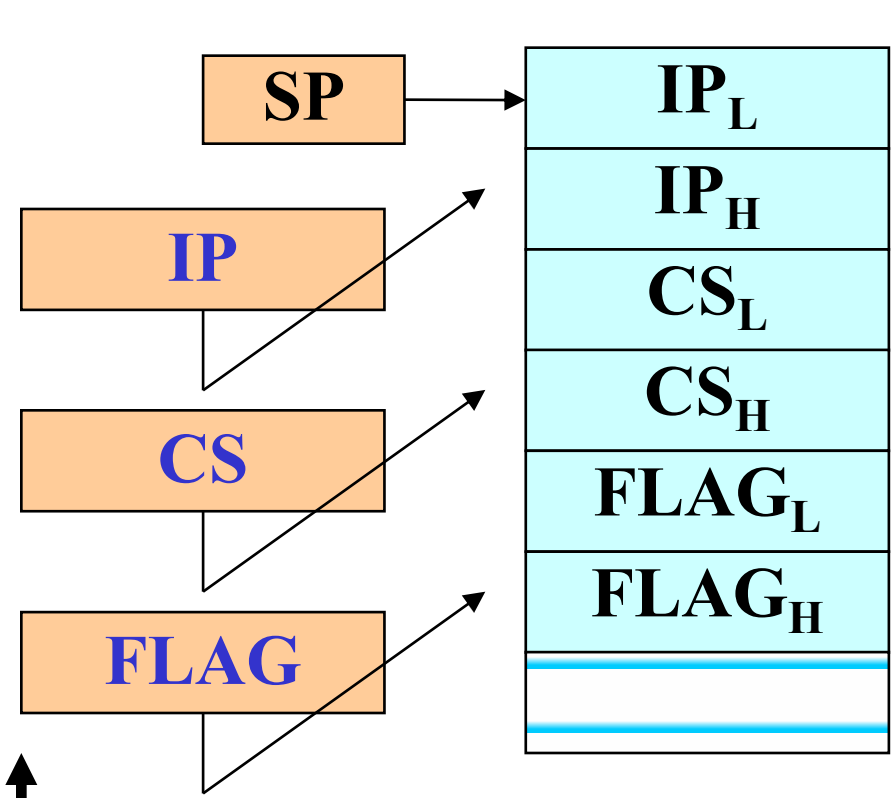
将屏蔽可屏蔽中断

STI：开启CPU中断(将标志寄存器FLAGS的IF位置位为1)，

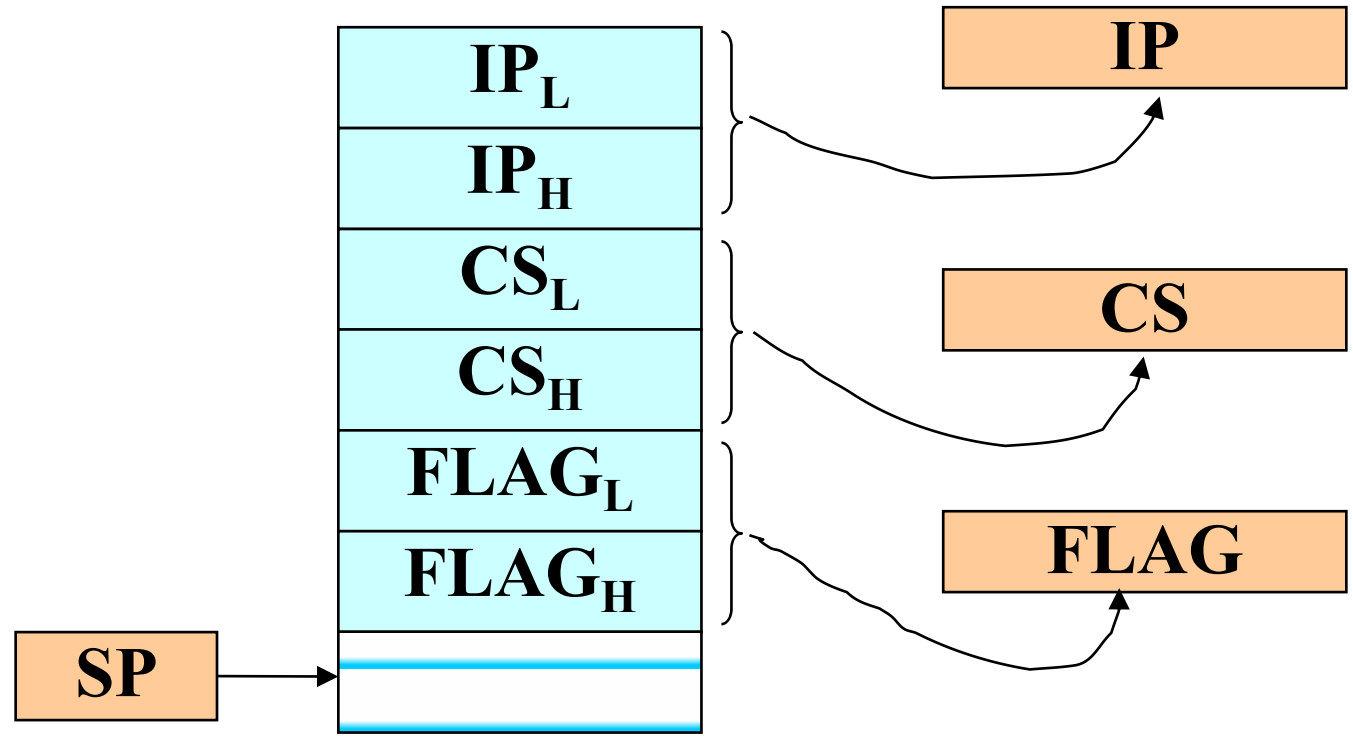
将响应可屏蔽中断

● INT和IRET中断指令

中断指令的执行过程中涉及的堆栈操作



中断响应 INT



中断返回 IRET

中断服务入口地址送入CS和IP，转去执行中断服务程序

所有的中断服务程序都必须通过IRET返回到断点

● 溢出中断指令INTO

– INTO指令首先检查溢出标志OF位
  - 如果OF=1，则启动一个中断号为4的中断过程
  - 如果OF=0，不做任何操作

– INTO指令一般安排在有符号数算术运算指令后面，用于进行运算结果的溢出检查

```
IMUL    BX              ;乘法指令
 INTO                   ;若溢出，则启动中断服务处理
MOV      RESULT, AX
MOV      RESULT+2, DX
```

■ 8086中断系统

8086中断类型

$16^2=4^8=2^8$

$$\begin{array}{r|l} 16 & 256 \\ \hline & 16 \\ \hline & 96 \end{array}$$

- ◆ 8086/8088的中断系统最多可以处理256种不同类型的中断
- ◆ 对于每个中断，都分配一个中断类型的编码，中断类型编码（中断号）为 0--255

专用中断： 类型号0～4 是专用中断，不允许做任何修改

保留中断： 类型号为5～31(05H～1FH)是系统保留中断；一般不允许用户改做它用(如打印屏幕(5)、定时器(8)、键盘输入(9)等)

用户中断： 类型号为32～255(20H～0FFH)为用户自定义中断

● 8086的中断类型

由CPU对状态标志寄存器中的单步标志TF的测试(TF＝1),而自动产生

此时，每执行一条指令，就会产生一个中断；

一般用在调试程序中

由指令INTO产生

当OF＝1时，执行 INTO 就会进入类型码为4的中断；其中断服务程序一定是处理运算过程中的溢出错误

CPU在执行除法指令时，若发现除数为0或商超…表达的范围，就立即产生了一…断;CPU不需要执行任何指令，…断，转去执行“除法错误中断服…

INT 3 指令

非屏蔽中断源

● 软件中断和硬件中断

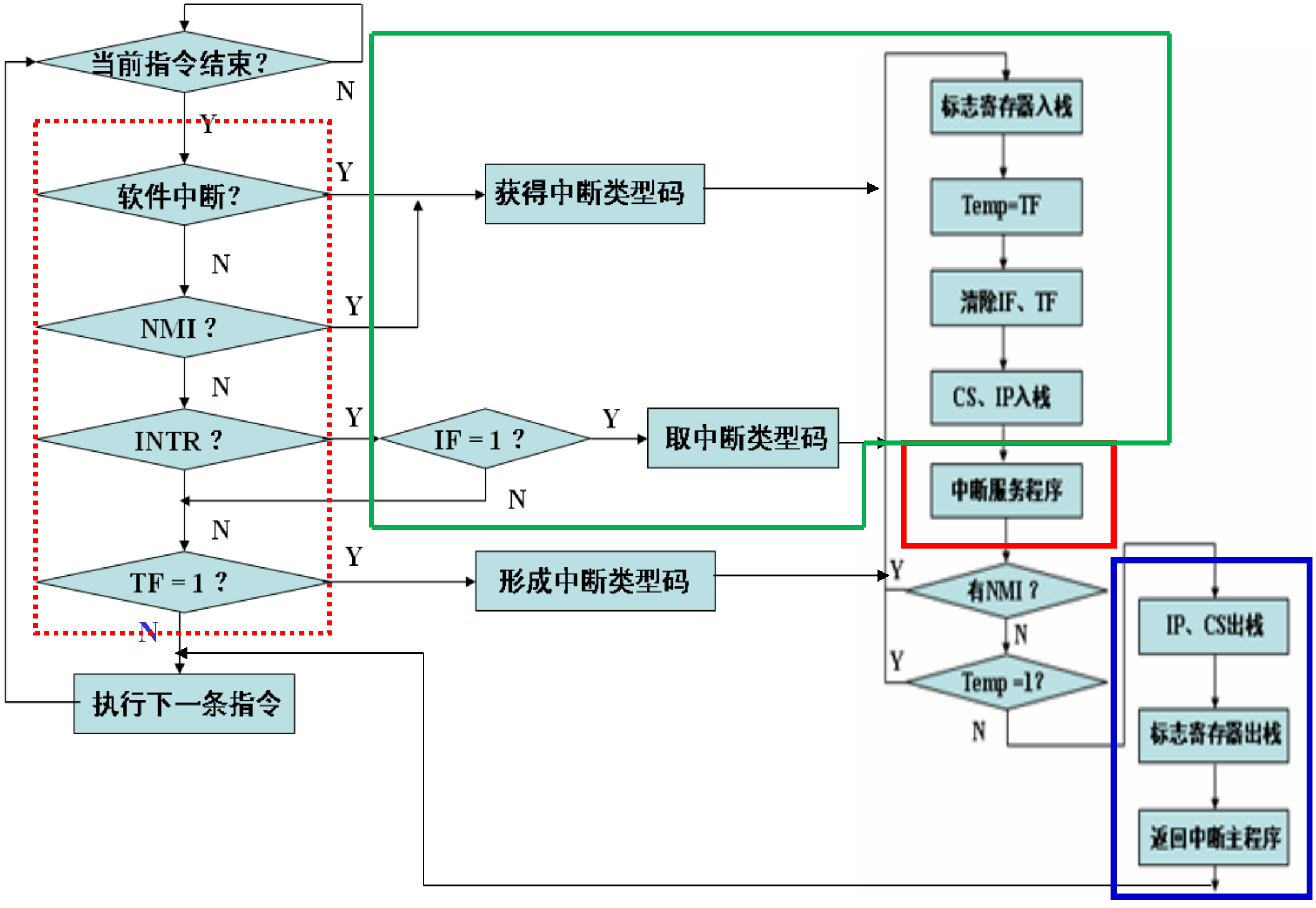
软件中断特点：

- 用一条指令进入中断处理子程序，类型号由指令给出
- 进入中断时，不需要执行中断响应总线周期
- 不受IF位的影响
- 正在执行软件中断时，如有外部硬件中断请求，则执行完当前指令后，根据条件允许否给以响应
- 软件中断没有随机性

硬件中断特点：

- CPU引脚信号NMI和INTR，随机传入
- 进入中断时，需要执行中断响应总线周期
- 可屏蔽中断的响应受IF位的影响

8086中断响应过程


当前指令结束?
N
Y
软件中断?
Y
N
NMI ?
Y
N
INTR ?
Y
N
TF = 1 ?
Y
N
执行下一条指令
获得中断类型码
IF = 1 ?
Y
N
取中断类型码
形成中断类型码
标志寄存器入栈
Temp=TF
清除IF、TF
CS、IP入栈
中断服务程序
有NMI ?
Y
N
Temp =1?
Y
N
IP、CS出栈
标志寄存器出栈
返回中断主程序

8086的中断向量

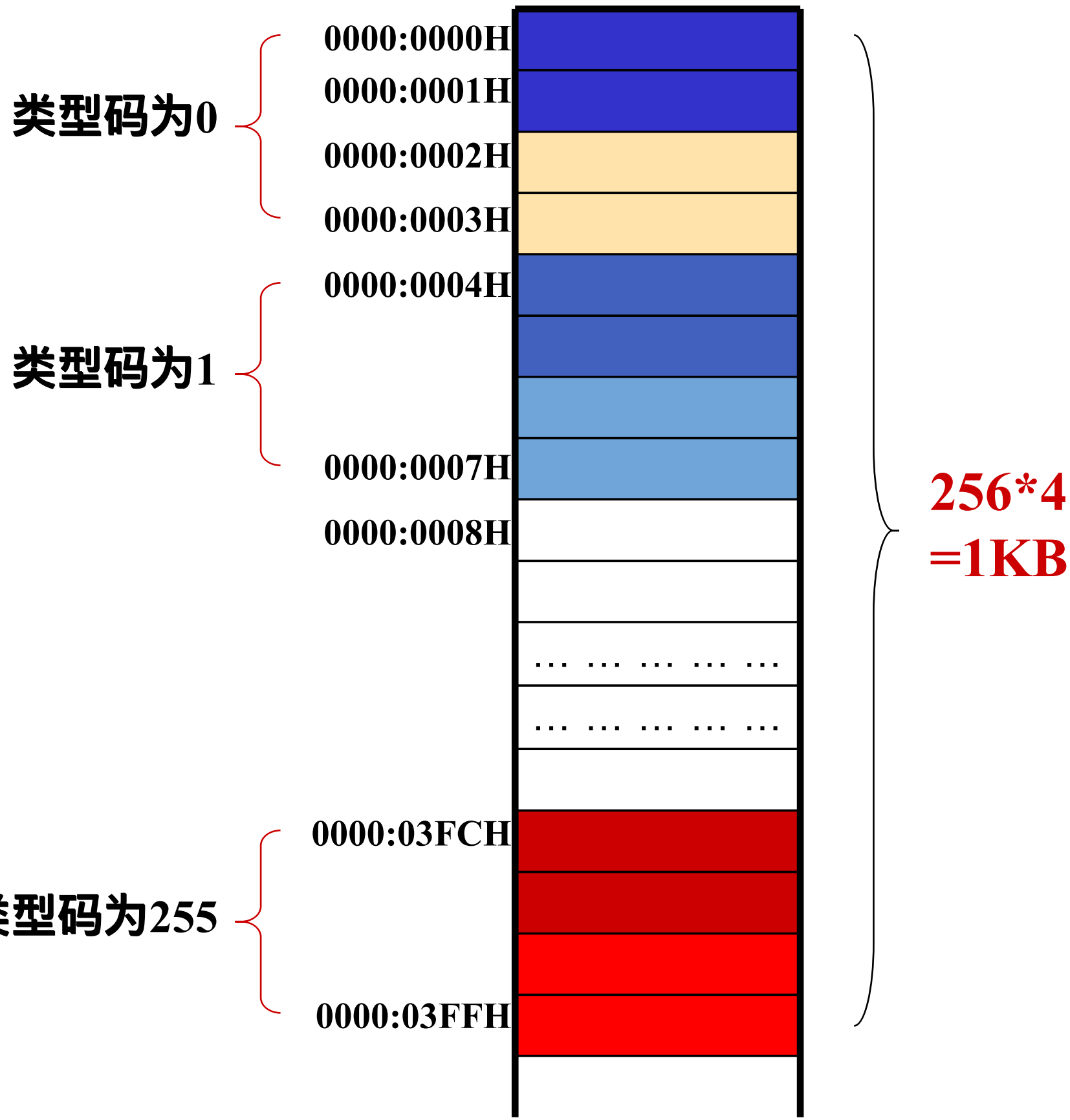
- 中断向量是中断服务程序的入口地址（中断服务程序在存储器中，存放单元的首地址）
  - 入口地址应该包含段基地址和偏移地址两个部分
  - 将占用4个字节
    - 低地址，存放中断服务程序的入口偏移地址
    - 高地址，存放其段基地址

■ 8086中断向量表

8086微处理器规定：从物理地址00000H开始，依次存放中断向量，形成一个特殊的专门存储中断服务程序入口地址的存储区域，称为**中断向量表**

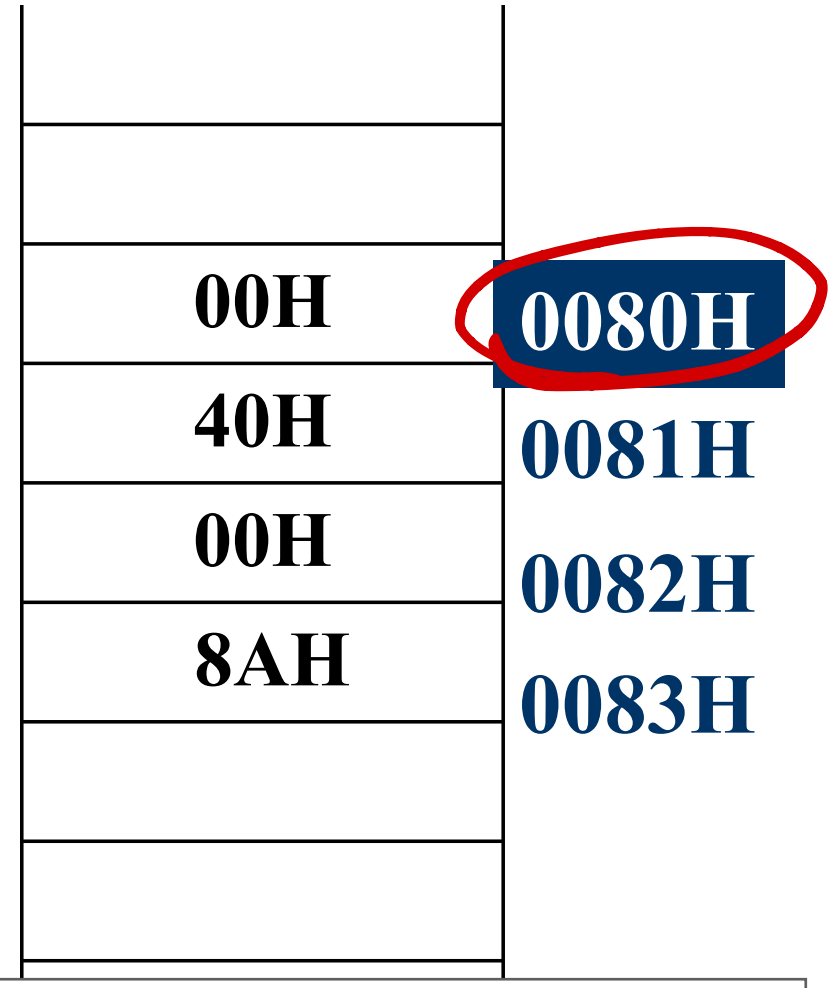
- 在该表中，各中断向量的存放顺序**按中断类型码排列**；并约定从 **0000H：0000H** 开始存放
- 256个不同类型的中断向量将占用1KB区域
- 故中断向量表建立在内存空间中最低1K地址，地址从00000—003FFH

中断向量表示意


0000:0000H
0000:0001H
0000:0002H
0000:0003H
0000:0004H
0000:0007H
0000:0008H
0000:03FCH
0000:03FFH
类型码为0
类型码为1
类型码为255
256*4
=1KB

中断向量地址指针

- 中断向量在中断向量表中的存放(首)地址称为中断向量地址指针
- 中断向量指针由CPU采用“中断类型码乘以4”的方式确定并管理
- 中断类型码联系着中断向量和中断向量地址指针



此例中的所存数据是类型码为32(20H)的中断服务程序的入口地址

中断向量相关概念例(1)

●**例如：** 对于中断类型码为24H的中断源

其中断向量地址指针为： 24H × 4＝0090H

说明

进一步分析!

中断向量相关概念例(2)

同样：

若中断类型码42H的中断向量为 5678H：1234H

说明

● 设置中断向量表

为中断服务程序安排了存储空间后，还须按照其中断类型码，确定中断向量指针，将它的入口地址置入中断向量表中；称为**中断向量表的设置**

设置中断向量表的作用和意义

中断向量表的设置方法

- **中断向量表的设置方法**
  - 用传送指令直接设置
  - 调用DOS的系统功能,并使用软中断指令INT 2lH 设置

- 传送指令直接设置

```
VECDATA   SEGMENT  AT  0000H        ；设置该段段基址的定位方式
          ORG  70H*4                ；对应类型码70H
          LOCA70    DW   2  DUP (?)
VECDATA   ENDS
CODE      SEGMENT
            ┆
          CLI                       ；关中断，设置过程中不再响应其他中断
          PUSH  DS                  ；保护现场
          MOV  AX，VECDATA
          MOV  DS，AX               ；指定段基址为 0000H
          MOV  LOCA70，OFFSET  INTSUB   ；存放中断向量
          MOV  LOCA70+2，SEG  INTSUB
          POP  DS                   ；恢复现场
          STI                       ；开中断
            ┆
CODE      ENDS
```

把本段装到指定的16位的段基址上

- **DOS调用设置中断向量表**

  - 调用DOS系统的25H号功能，设置中断向量表
  - 执行调用前需预置相关的入口参数

    1) AH中预置功能编号25H；

    2) AL中预置中断类型码；

    3) DS和DX中预置中断服务程序的入口地址(段地址和偏移地址分别置入DS和DX)

  - 预置完以上参数后，执行 INT 21H 指令就可把中断服务程序的入口地址置入中断向量表中适当的位置

DOS调用设置中断向量例

例：设中断服务程序的入口地址为INTSUB，中断类型码为70H，则用下列指令段可设置中断向量

```
PUSH   DS
MOV    AX，SEG  INTSUB
MOV    DS，AX
LEA     DX，INTSUB
MOV    AL，70H
MOV    AH，25H
INT    21H
POP    DS
```

DOS调用读取中断向量例

- DOS功能调用35H用来读取中断向量表
- 入口、出口参数如下：

AH：预置DOS功能编号35H

AL：中断类型号

ES：读出的中断服务程序入口的段地址

BX：读出的中断服务程序入口的偏移地址

```
MOV   AL，70H
MOV   AH，35H
INT   21H
```

80486的中断系统

- 中断类型

1) 外部中断

(1) NMI

(2) INTR

2) 异常中断

(1)故障中断

(2)陷阱中断—软件中断

(3)异常终止

- 中断服务程序地址的获取

1) 实地址模式

与8086方式相同

2) 保护模式

● 中断描述符

采用中断描述符表示中断服务程序的入口地址（类似段描述符）

31 ~ 16	15 ~ 0
偏移量31~16	参数
段选择子	偏移量15~0

● 中断描述符表IDT

每个中断描述符占8个字节，所有中断描述符集中存放在中断描述符表IDT。

可定义256个不同中断或异常，IDT大小为：256 × 8=2KB空间

IDT的基地址由CPU中的中断描述符表寄存器IDTR提供。

- **中断描述符类型**
  - 任务门在多任务系统中完成任务的切换
  - 陷阱门对应异常处理子程序
  - 中断门对应外部中断的处理子程序

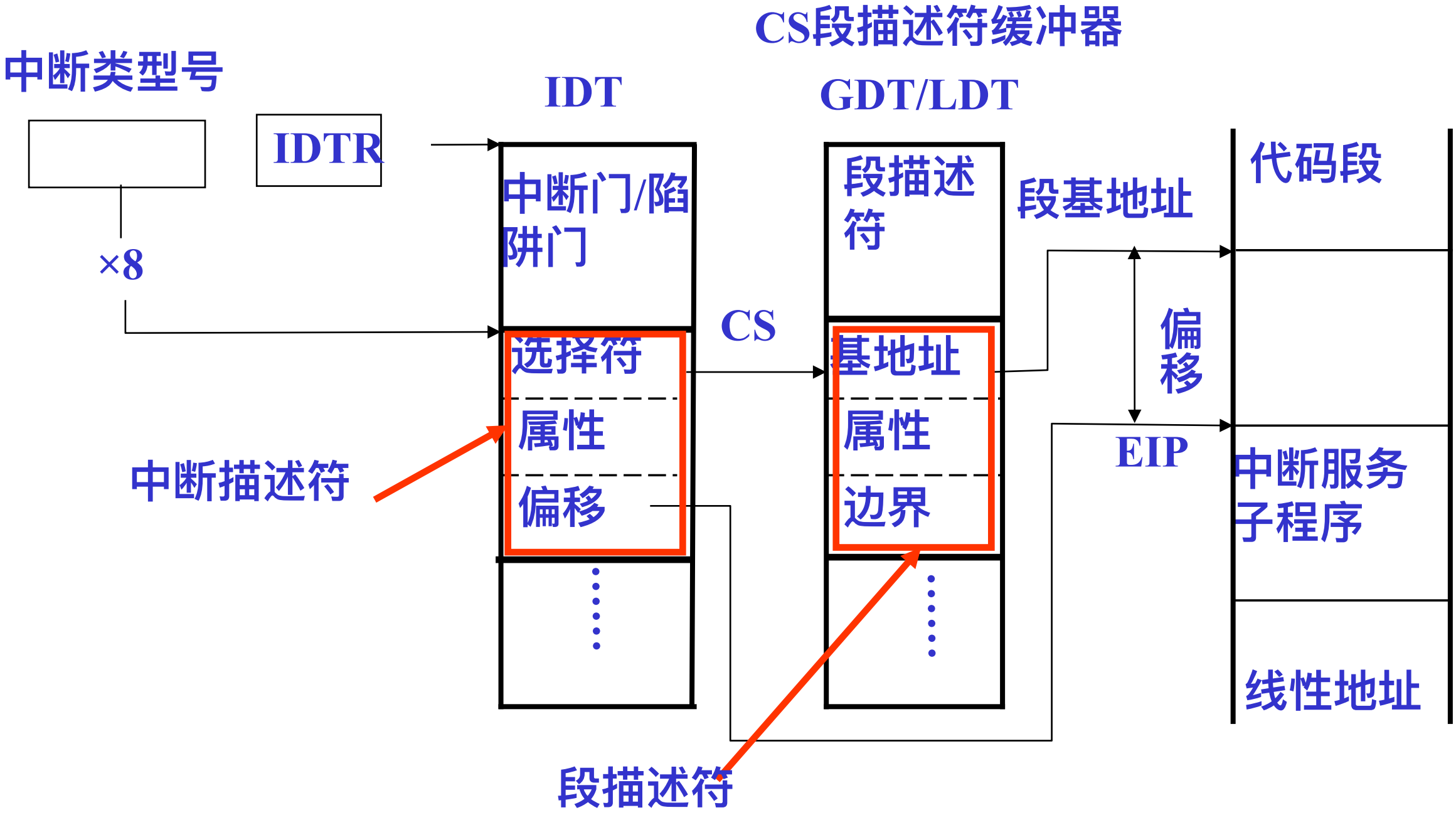
- **中断描述符指针**

中断描述符指针是中断描述符在IDT中的地址

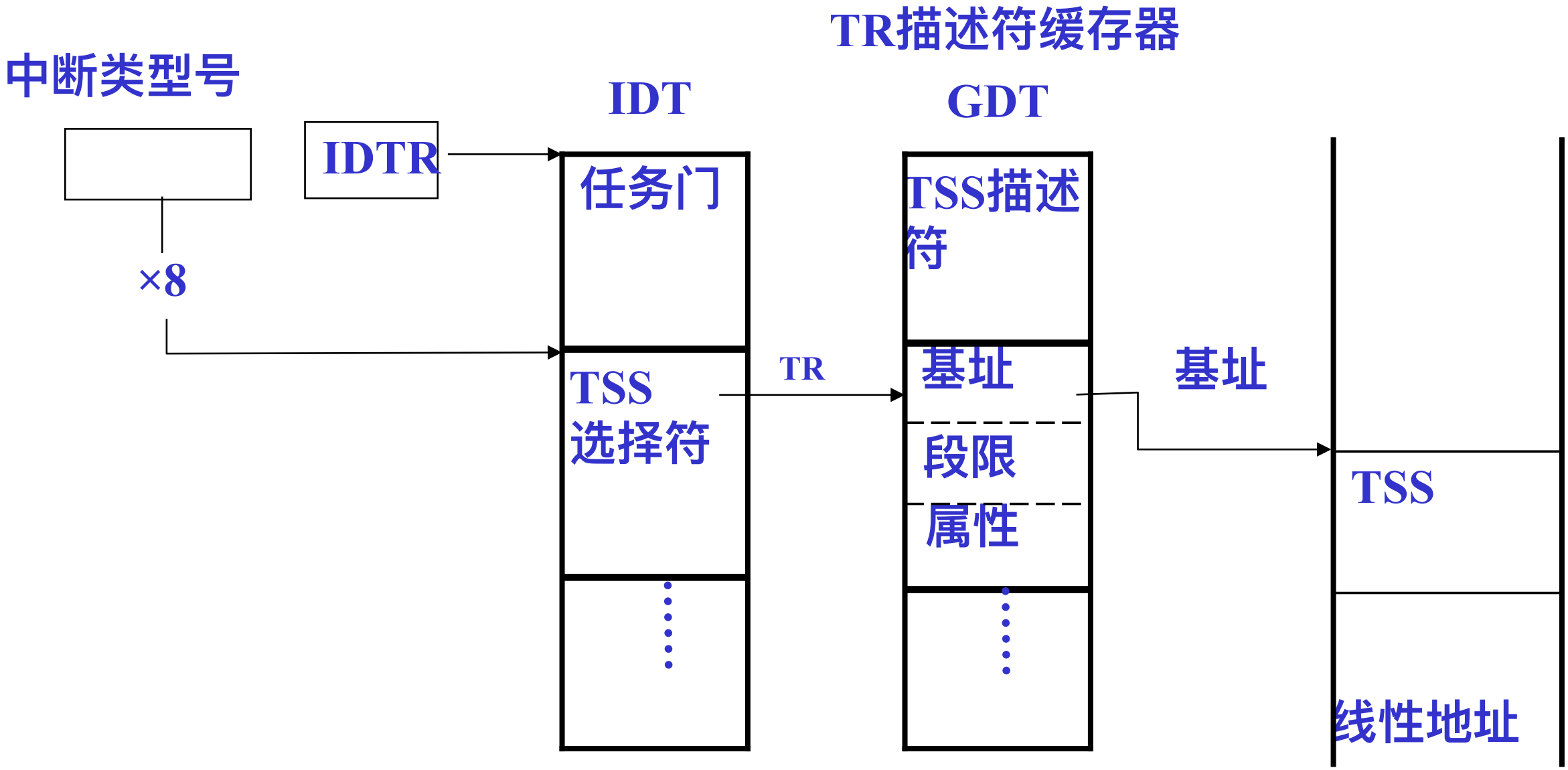
中断描述符指针=IDTR基地址+中断类型号× 8

- 中断服务程序地址的获取

1) 中断门/陷阱门的地址计算



2) 任务门的地址计算


中断类型号
×8
IDTR
IDT
任务门
TSS
选择符
TR
TR描述符缓存器
GDT
TSS描述符
基址
段限
属性
基址
TSS
线性地址

可屏蔽中断

键盘 | 磁盘 | 传感器 | 定时器 | 光电输入驱动 | 磁盘驱动 | 打印机 | 鼠标

(部分)外部硬件中断

IRiA

中断类型码

正常运行的程序

中断控制器
如8259A芯片

中断服务程序
入口地址

中断屏蔽

中断判优

IF=0/1

8086
cpu

CS

IP

中断请求应答

- 中断请求
- 中断屏蔽与优先级判别
- 中断响应与保存现场
- 中断服务
- 恢复现场与中断返回

不可屏蔽中断
软件中断、除法运算溢出
CPU内部硬件中断
(部分)外部硬件中断

现场进/出栈
保护现场

中断嵌套

中断返回

中断响应

中断处理过程

中断及相关概念示意

● INTR中断响应的总线时序

● **INTR中断请求**被响应时，CPU实际执行的总线时序全过程如下：

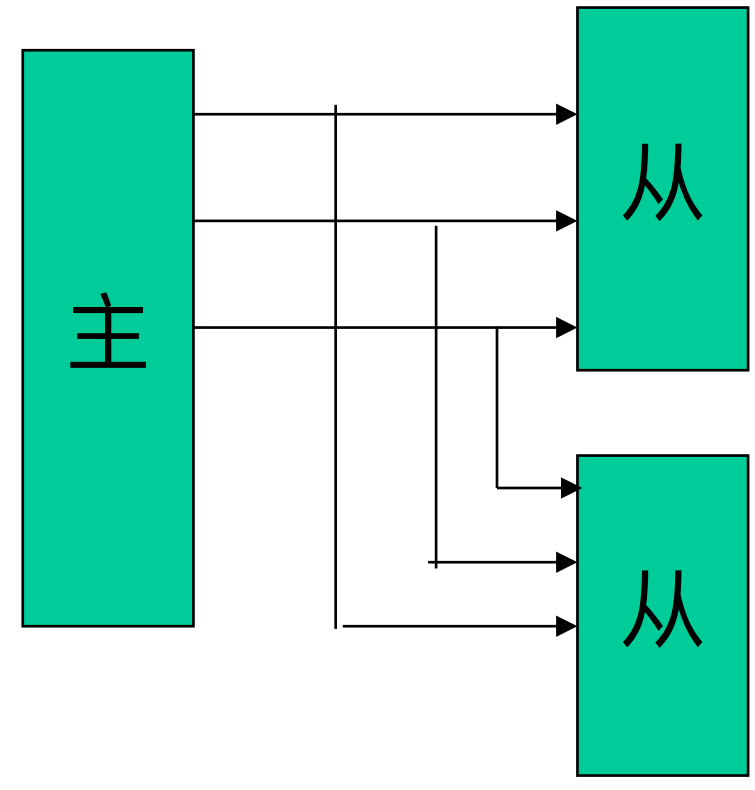
1) 执行两个中断响应总线周期；被响应的中断源在第二个中断响应总线周期，由低8位数据线送回一个单字节的中断类型码；CPU接收后，左移两位(× 4)，作为中断向量首字节地址，存入CPU内部暂存器。
2) 执行一个总线写周期，把状态标志寄存器FR推入堆栈
3) 把FR中的中断允许标志IF和单步标志TF置0，禁止中断响应过程中，其他可屏蔽中断的进入；同时禁止中断处理过程中单步中断
4) 执行一个总线写周期，把CS的内容推入堆栈
5) 执行一个总线写周期，把IP的内容推入堆栈
6) 执行一个总线读周期，把中断向量前两个字节读入，送到IP
7) 执行一个总线读周期，把中断向量后两个字节读入，送到CS

● **非屏蔽中断或软件中断**，则由第2步开始执行，因为此时中断类型码已确定无需从数据线上读取

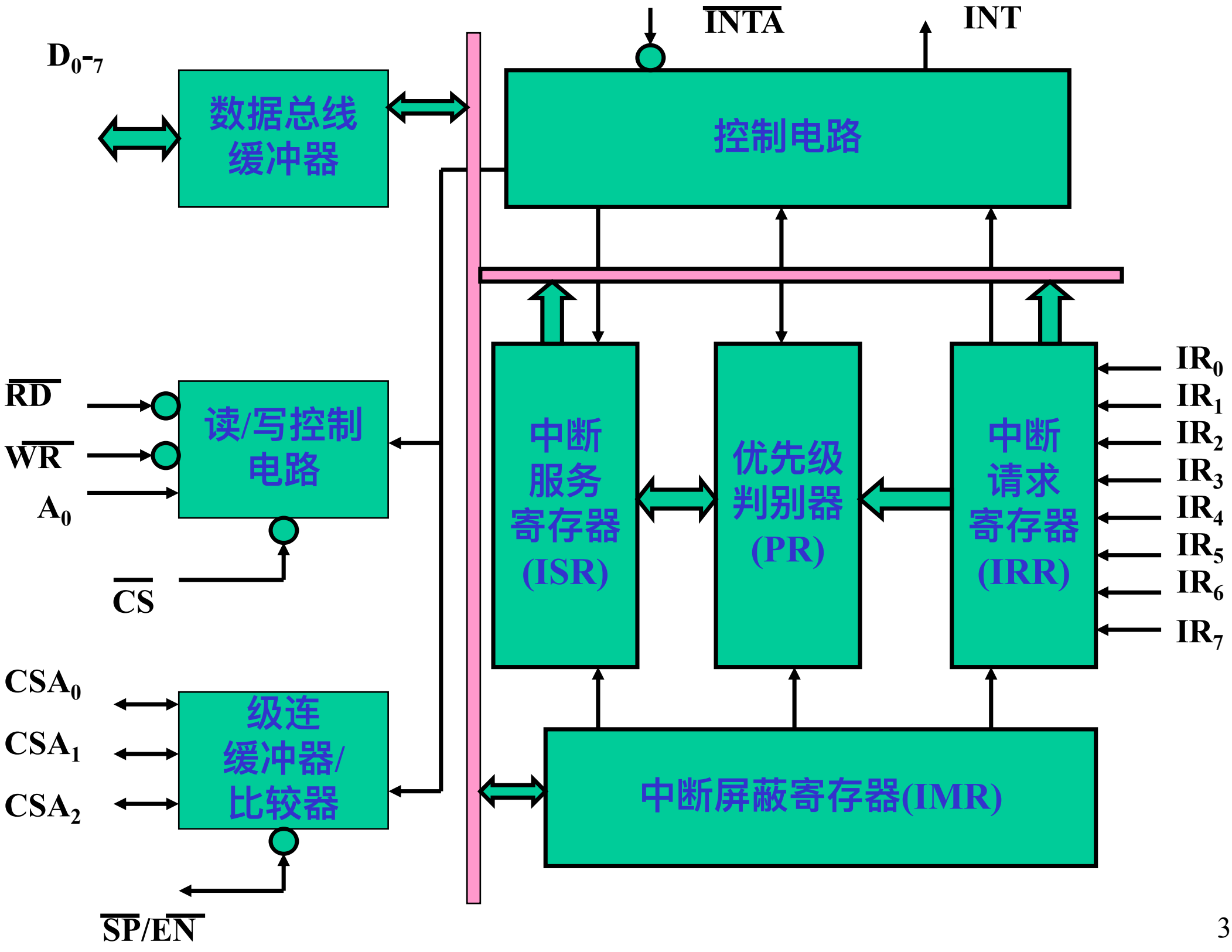
8.2 中断控制器8259A

■ 8259A芯片的功能

- 单片8259A芯片可管理8个中断源。
- 采用9片8259A芯片可管理8 × 8=64个中断源。
- 采用多片8259A芯片时，其中一片为主芯片，其余为从芯片。
- 最多可以有一个主芯片，8个从芯片。
- 可编程为高电平触发或脉冲边沿触发。
- 8个中断输入的任何一个可由软件屏蔽。
- 中断优先级可由软件设置。



8259A内部组成


D0-7
数据总线
缓冲器
INTA
INT
控制电路
RD
WR
A0
读/写控制
电路
CS
中断
服务
寄存器
(ISR)
优先级
判别器
(PR)
中断
请求
寄存器
(IRR)
IR0
IR1
IR2
IR3
IR4
IR5
IR6
IR7
CSA0
CSA1
CSA2
级连
缓冲器/
比较器
SP/EN
中断屏蔽寄存器(IMR)

1 IRR

用于存放外部的请求信号。当有请求时，使相应位置1。当收到/INTA信号时，使相 应位置0（复位）。

2 IMR

IMR_{0_7}与IR_{0_7}相对应，用于对中断请求信号进行控制。若IMR_i为0，则允许IR_i向外发中断请求。若为1，则不允许。

3 ISR

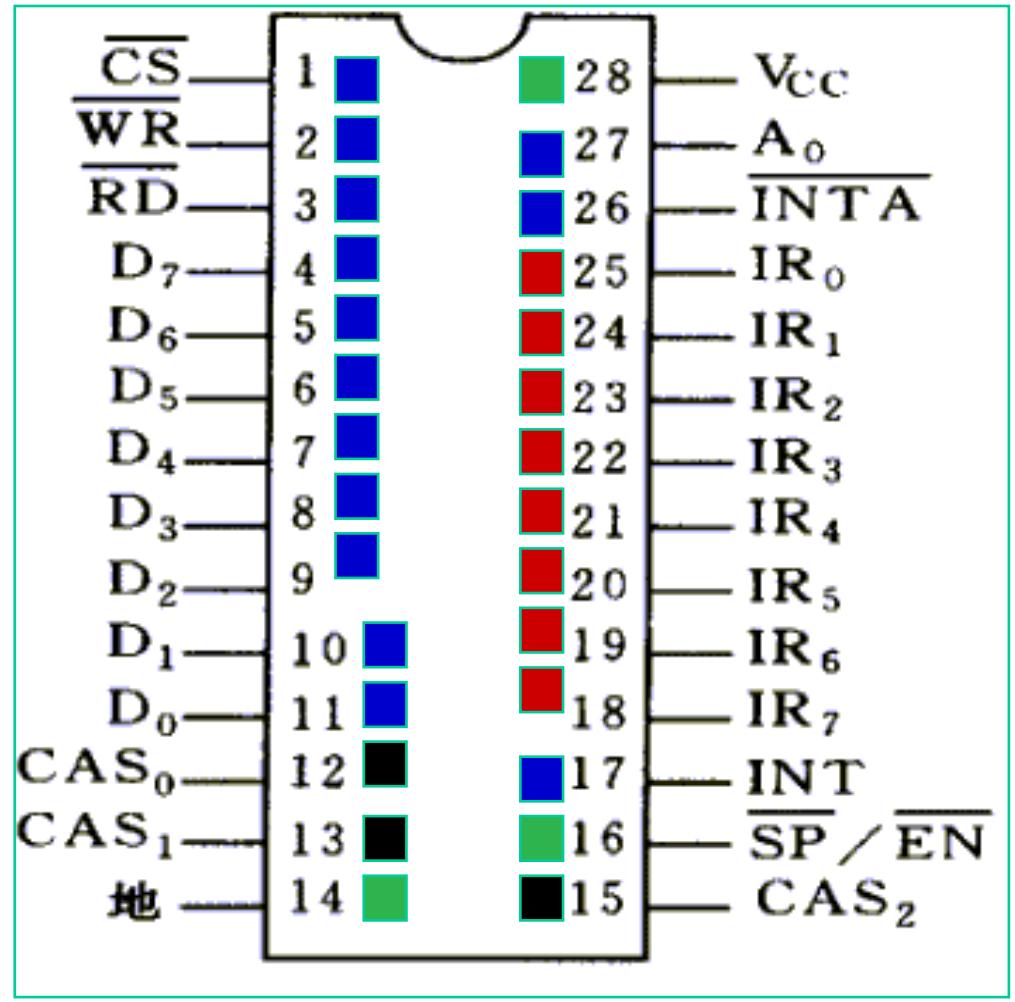
ISR_I = 1表示8259A正在请求服务中。服务结束后可自动复位（置0），或由软件命令复位，这要由8259A的初始化来决定。

4 优先权判别器（PR）

当输入端IR7～IR0中有多个中断请求信号同时产生时，由PR判定哪个中断请求具有最高优先权，并在脉冲期间把它置入中断服务寄存器ISR的相应位。

● 8259A外部特征

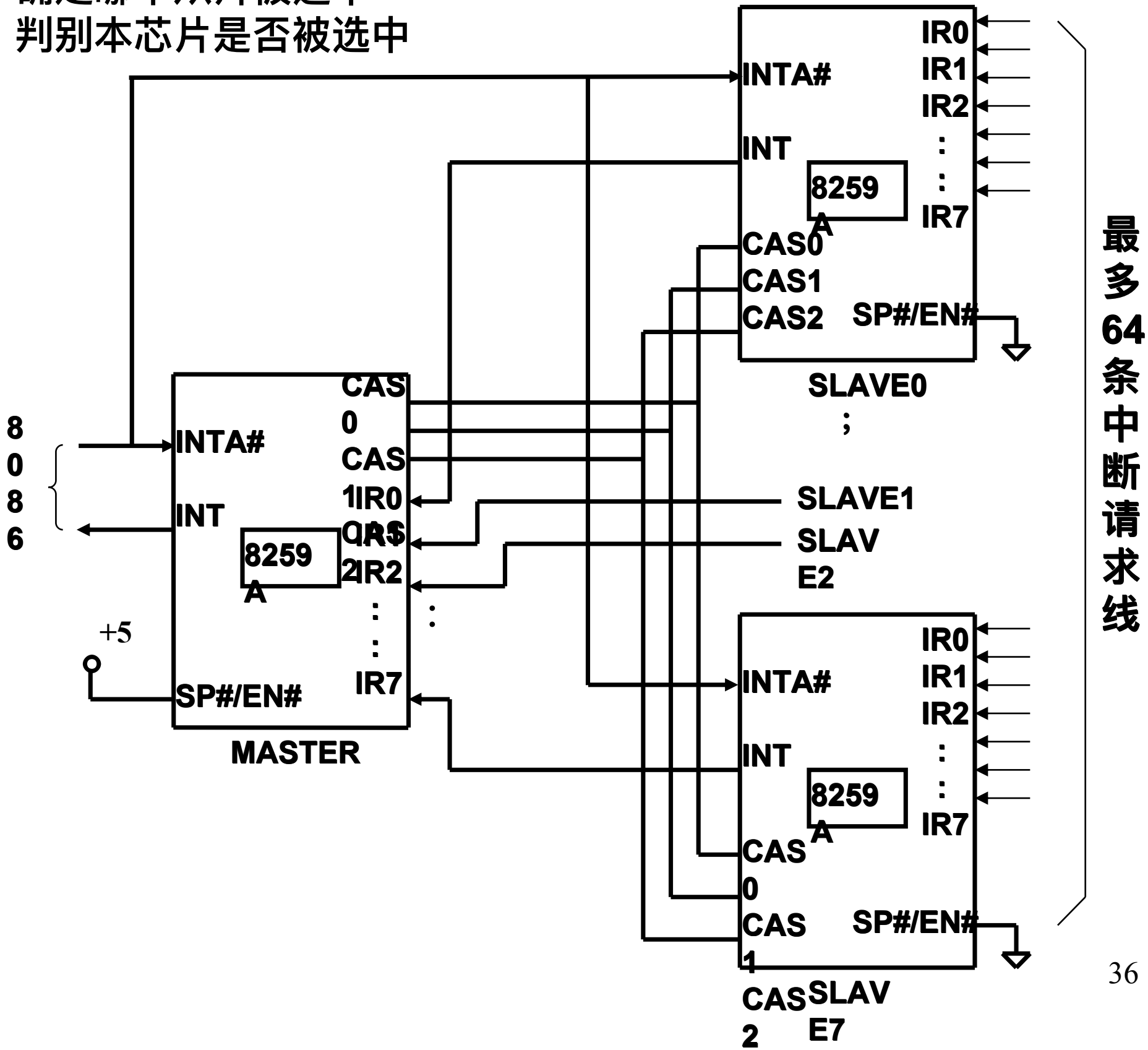
名称	输入 / 输出	功　能
CS	输入	片选信号
WR	输入	写命令信号
RD	输入	读命令信号
D_7-D_0	输入/输出	双向数据总线
CAS_0-CAS_3	输入/输出	级联总线
SP/EN	输入/输出	主从定义/缓冲器方向
INT	输出	8259A中断申请
IR_0-IR_7	输入	外设的中断请求
INTA	输入	中断响应信号
A_0	输入	A_0地址线



$CAS_{0—2}$：多片8259A的级连信号

主芯片：输出，确定哪个从片被选中

从芯片：输入，判别本芯片是否被选中



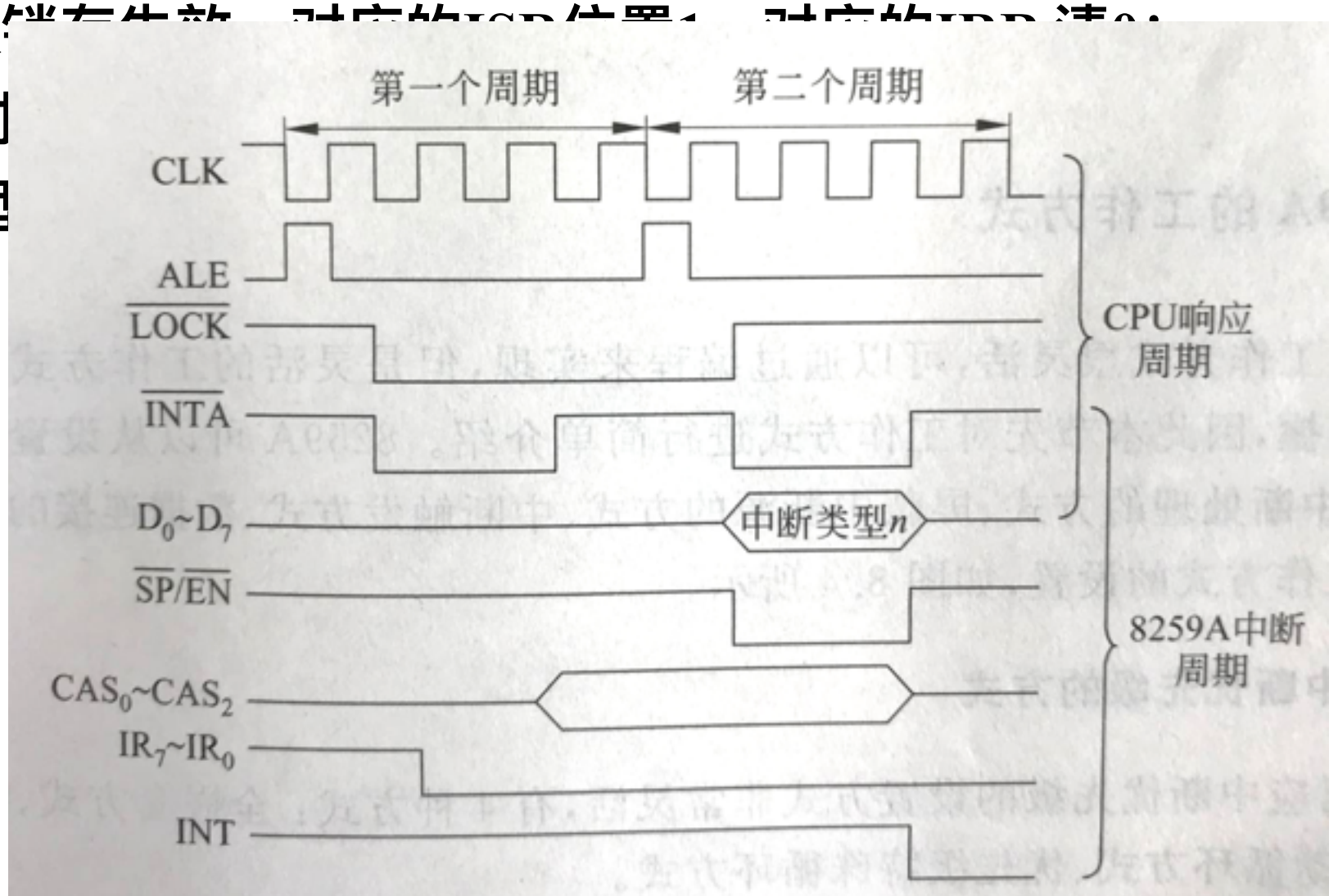
● 8259A芯片的中断响应过程

1) 当IR_i中有一条线变高时，将把IRR中的相应位置1；
2) 若IMR_I中的相应位为0，则表示允许该中断请求信号进入PR；
3) PR对新进入的中断请求信号和当前正在处理的中断进行判优， 如新进入的中断请求信号优先级高，则发出INT；
4) 若CPU允许中断IF=1，则发出/INTA。

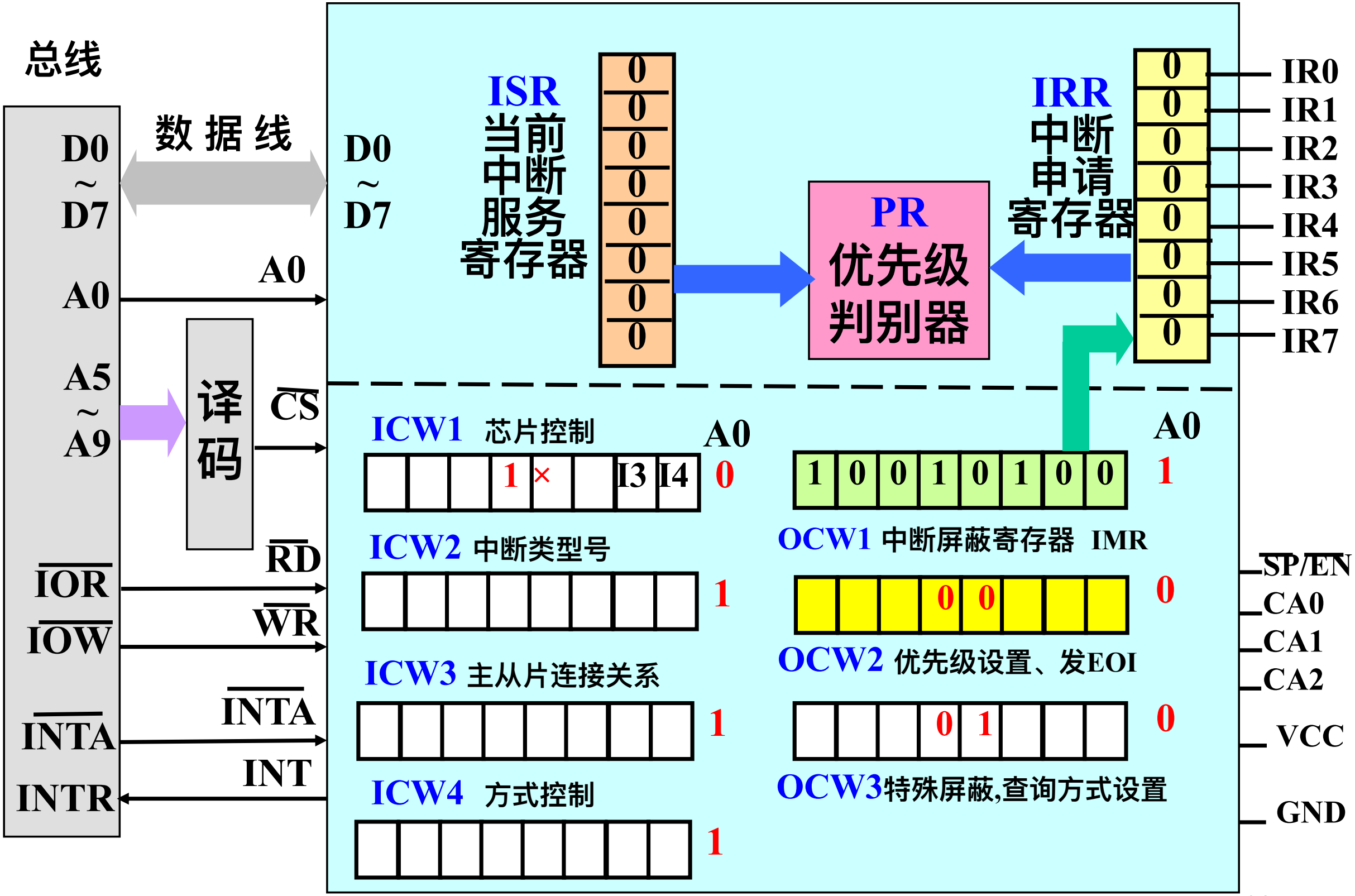
收到/INTA，使IRR

6) 收到第二个/INTA时

7) CPU收到此中断类型



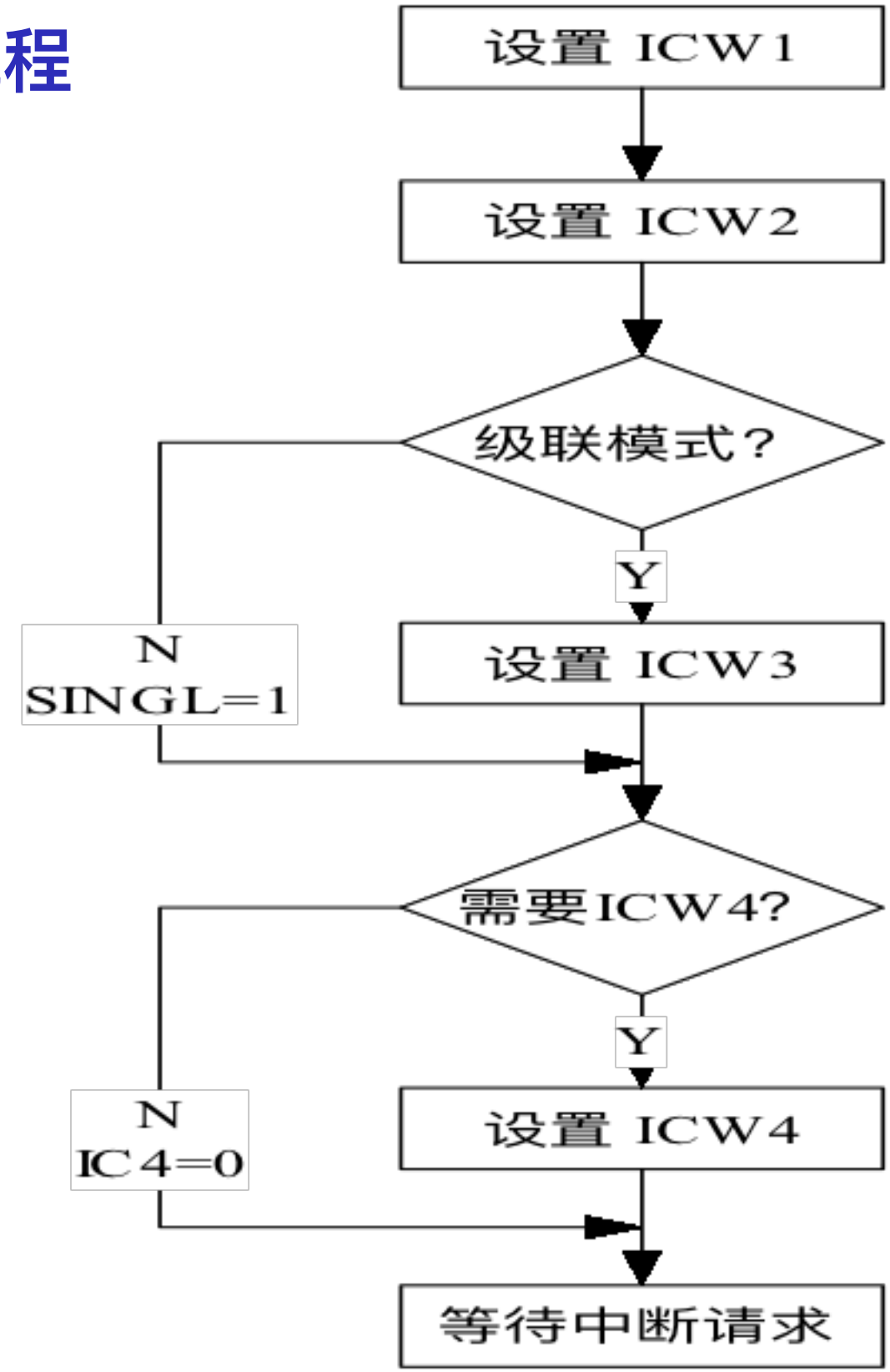
● 8259A芯片的编程结构



- 8259A内部寄存器的地址

A_0	D_4	D_3	/RD	/WR	/CS	输出（CPU读8259A）
0			0	1	0	IRR，ISR或中断号送数据总线
1			0	1	0	IMR送数据总线
						输入（CPU写8259A）
0	0	0	1	0	0	数据总线送OCW2
0	0	1	1	0	0	数据总线送OCW3
0	1	X	1	0	0	数据总线送ICW1
1	X	X	1	0	0	数据总线送OCW1，ICW2，ICW3，ICW4
						禁止
X	X	X	1	1	0	数据总线为高阻
X	X	X	X	X	1	数据总线为高阻

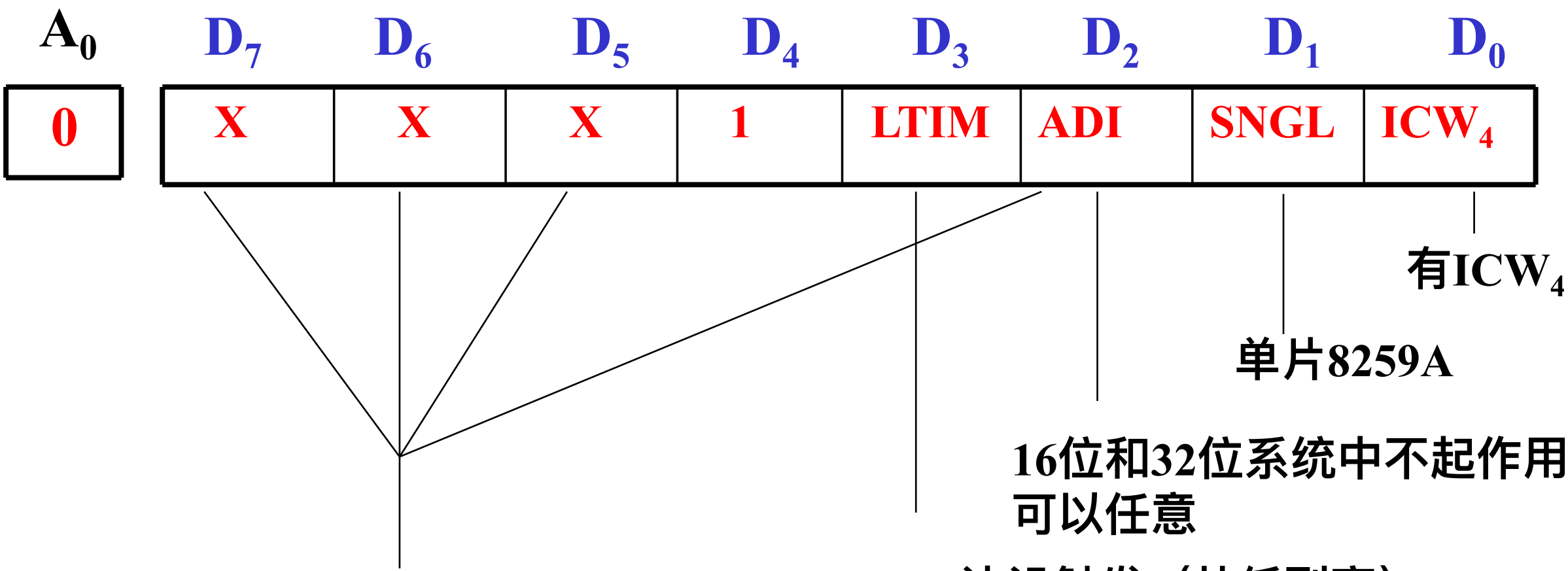
- **8259A初始化流程**


设置 ICW1
设置 ICW2
级联模式？
Y
N
SINGL=1
设置 ICW3
需要ICW4?
Y
N
IC4=0
设置 ICW4
等待中断请求

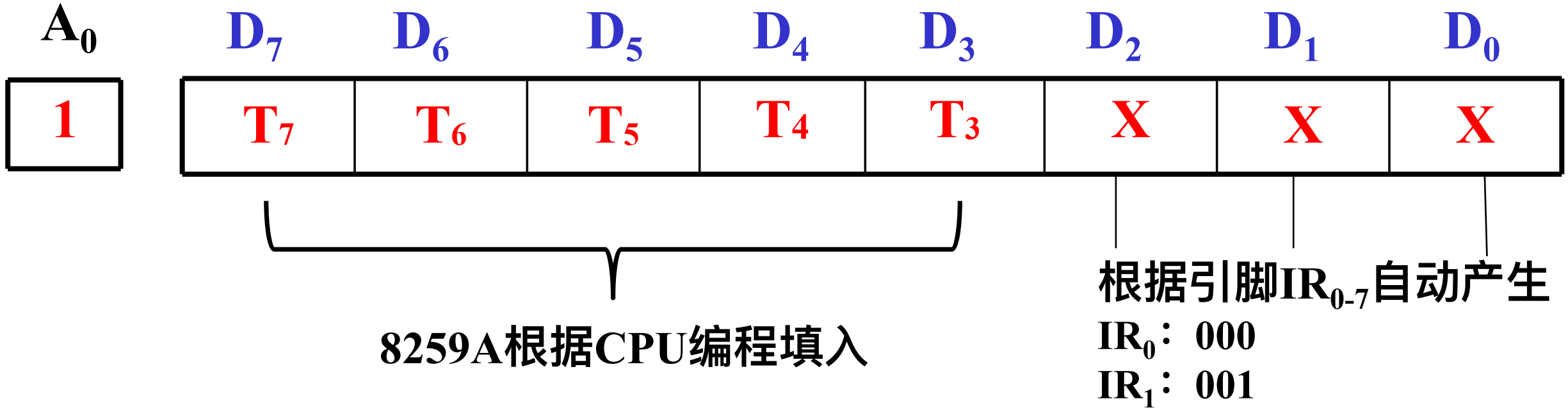
- **8259A的初始化命令字和操作命令字**

1 初始化命令字

1） ICW1的格式和含义



2） ICW2的格式和含义



- 定义8259A管理的中断类型号
- 当8259A和8086相连时

  D_{3_7}由8259A根据CPU编程填入， D_{0_2}由对应引脚自动产生，
- 第2个中断响应周期，8259A送出ICW2; CPU收到后乘以4得到向量地址。

3） ICW3的格式和含义

A_0	D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
1	IR_7	IR_6	IR_5	IR_4	IR_3	IR_2	IR_1	IR_0

- 当8259作为主芯片时

$$D_{0—7} = \begin{cases} 1 & IR_i\text{端的输入有8259A芯片} \\ 0 & IR_i\text{端的输入无8259A芯片} \end{cases}$$

- 当8259作为从芯片时

  $D_{0—2}$用于规定从芯片的地址，当主芯片给出的$CAS_{0—2}$状态编码与$D_{0—2}$相同时，则中断类型号就由此芯片给出。

4） ICW4的格式和含义

A_0	D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
1	0	0	0	SFNM	BUF	M/S	AEOI	uPM

0---同8080 连
1---同8086 连

0---非自动中断结束方式
1---自动中断结束方式

BUF=1时用
1---8259A作主芯片
0--- 8259A作从芯片

0---全嵌套方式
1---特殊全嵌套方式

1 = 缓冲方式

当BUF=0 时,8259A作主芯片还是从芯片要由SP/EN管脚的连接决定。
SP/EN=1 表示主芯片。

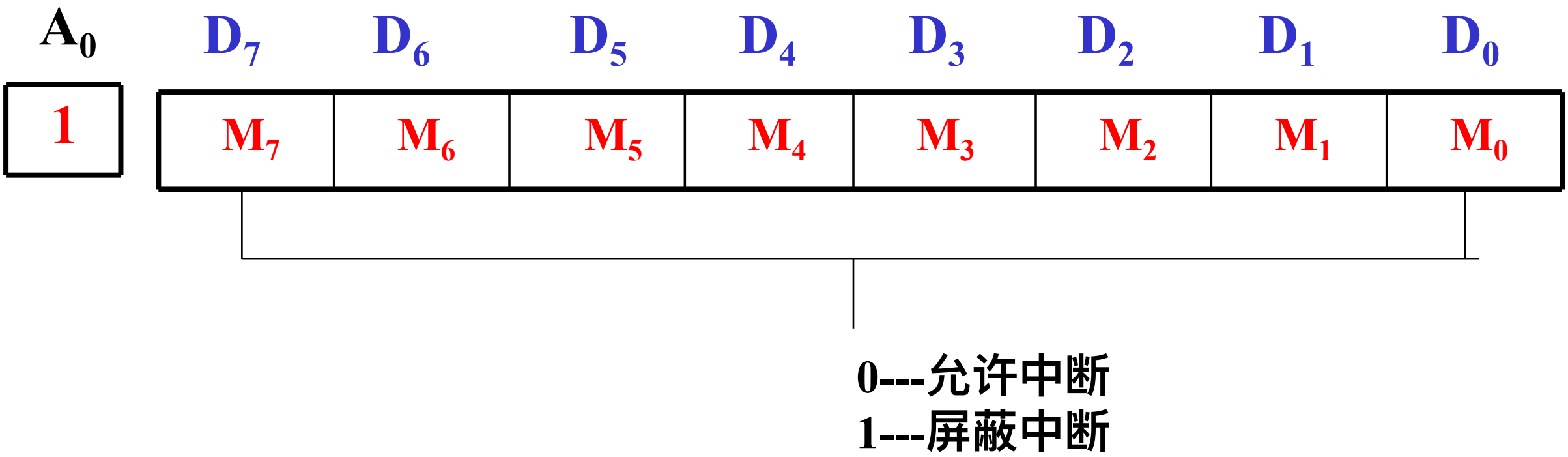
设8259A的端口地址为80H、81H。下面是该8259A的初始化程序段：

```
MOV   AL，13H
OUT   80H，AL        } ICW1

MOV   AL，18H
OUT   81H，AL        } ICW2

MOV   AL，0DH
OUT   81H，AL        } ICW4
```

该8259A芯片的工作状态为：单片8259A，中断请求上升沿触发，中断类型码为18H，19H，1AH，1BH，1CH，1DH，1EH，1FH分别对应IR_0～IR_7，普通全嵌套方式，缓冲方式，用于8086／8088系统中，普通中断结束方式

2 操作命令字

1） OCW1 的格式和含义



8259A工作时，任意时间，可从奇地址口写入、读出IMR

2） OCW2 的格式和含义

A_0	D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
0	R	SL	EOI	0	0	L_2	L_1	L_0

EOI= 1 表示把$L_{0—2}$指定的ISR_i位清0

SL = 1 表示$L_{0—2}$有效

R = 1 表示循环方式

R	SL	EOI	
0	0	0	结束优先级循环方式
0	0	1	把当前处理的中断对应的ISR_i清0
0	1	0	无意义
0	1	1	把$L_{0—2}$指定的ISR_i位清0
1	0	0	使工作在优先级循环方式
1	0	1	把当前处理的中断对应的ISR_i清0,置优先级循环方式
1	1	0	把$L_{0—2}$指定的IR_i置为最低优先级
1	1	1	把$L_{0—2}$指定的ISR_i位清0并置$L_{0—2}$指定的IR_i为最低优先级

000---相应于$ISR_0(IR_0)$

001---相应于$ISR_1(IR_1)$

111---相应于$ISR_7(IR_7)$

3） OCW3 的格式和含义

A_0	D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
0	0	ESMM	SMM	0	1	P	ERIS	RIS

00---不起作用
01---不起作用
10---设置成普通屏蔽方式
11---设置成特殊屏蔽方式

00---不起作用
01---不起作用
10---用下一个/RD信号读IRR
11---用下一个/RD信号读ISR

P = 1 查询方式

用下一个/RD信号 读中断申请中最高优先级号。

- 查询字格式

P = 1　中断查询方式，读出当前中断申请优先级最高的IRi

A_0	D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
0	1	X	X	X	X	W2	W2	W0

000---IR_0
001---IR_1
010---IR_2
…
111---IR_7

8259A工作方式

■ 中断嵌套方式

1 全嵌套方式

1) 中断请求优先级按IR_0到IR_7顺序排列， IR_0优先级最高。

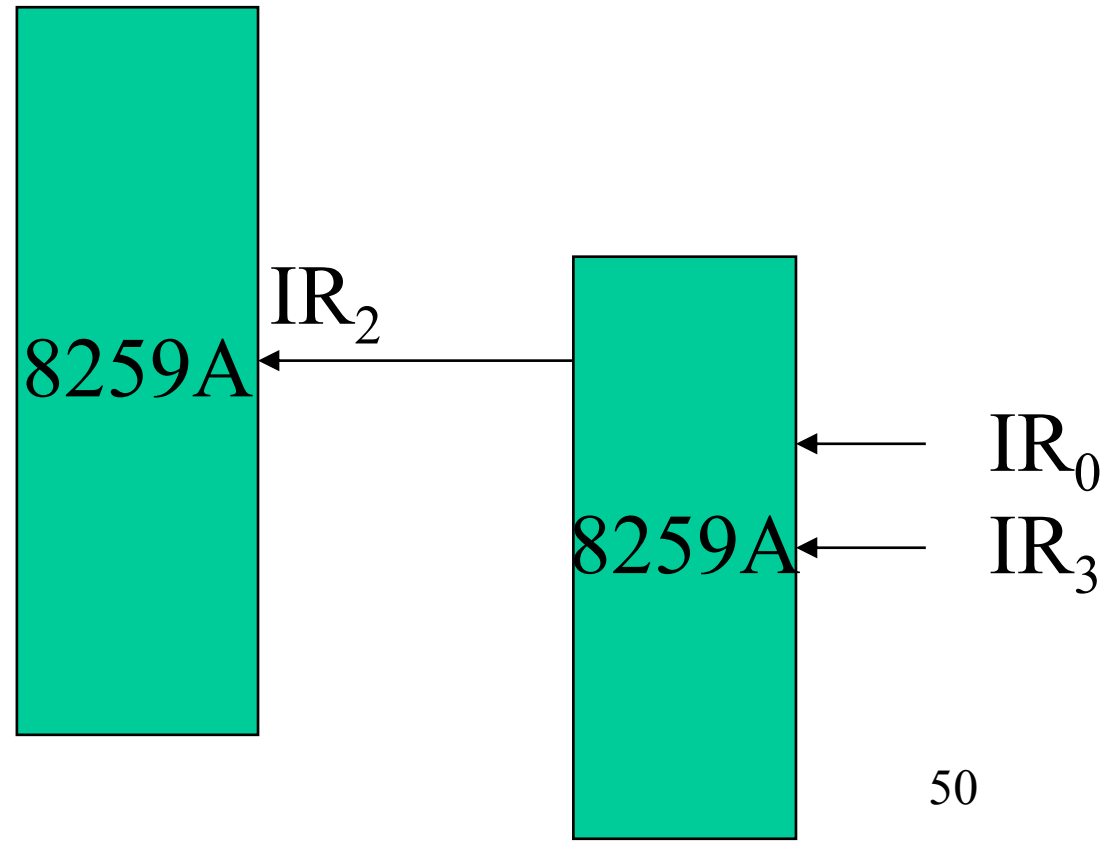
2) 如果8259A初始化后没有设置其它优先级方式，则按全嵌套方式。

3) 同级或低级的中断申请被屏蔽。（响应过程中ISRi一直保持1）

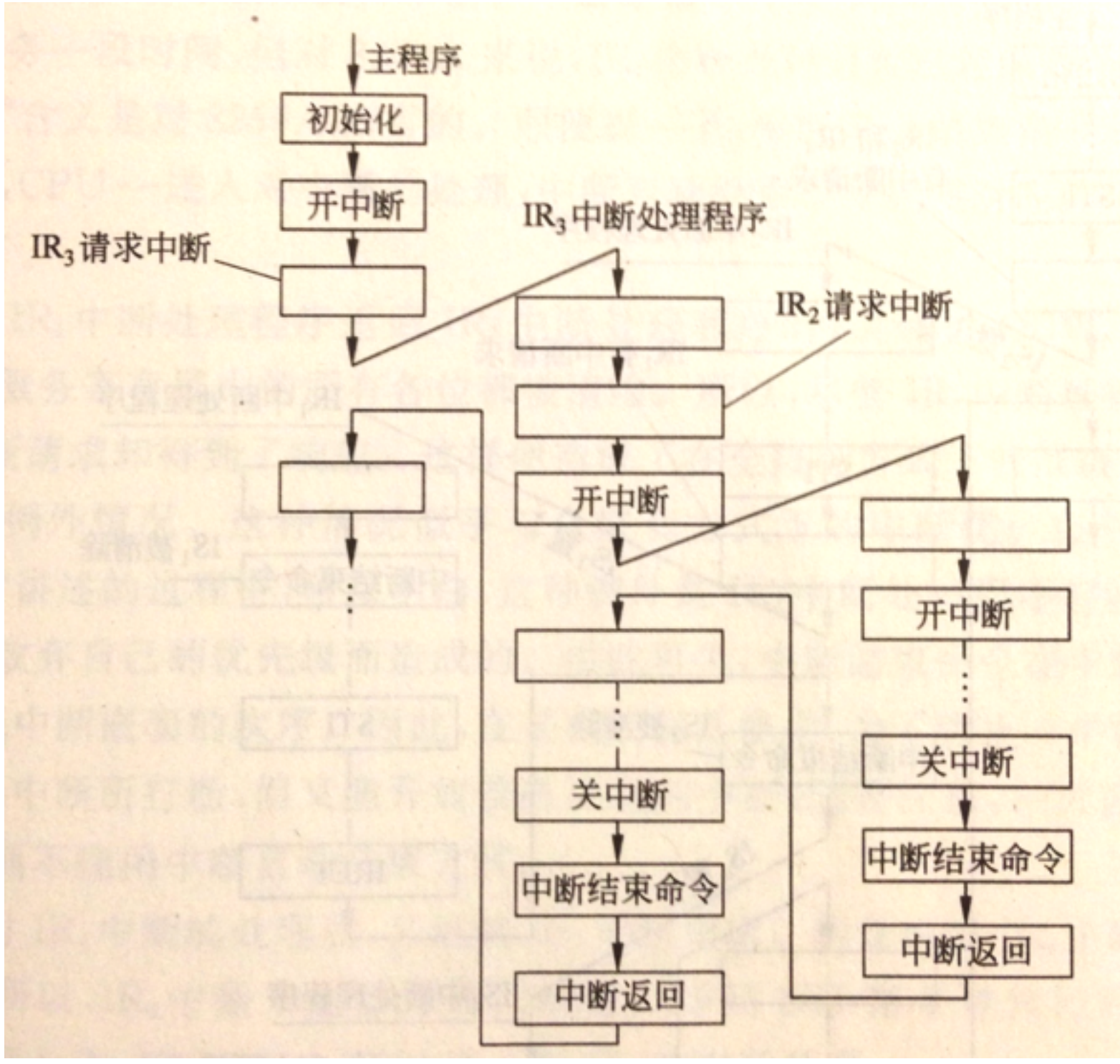
2 特殊全嵌套方式

1）一般用于多片8259A的级连中。

2）能响应同级的中断请求。



全嵌套方式例：



条件：

1.主程序必须执行开中断指令，IF=1，才可能响应中断；

2.进入中断处理程序，系统会自动关中断；中断处理程序中再次开中断，才能形成中断嵌套；

3.每个中断处理程序结束后，必须执行中断结束命令，清除对应的ISi，才能返回断点。

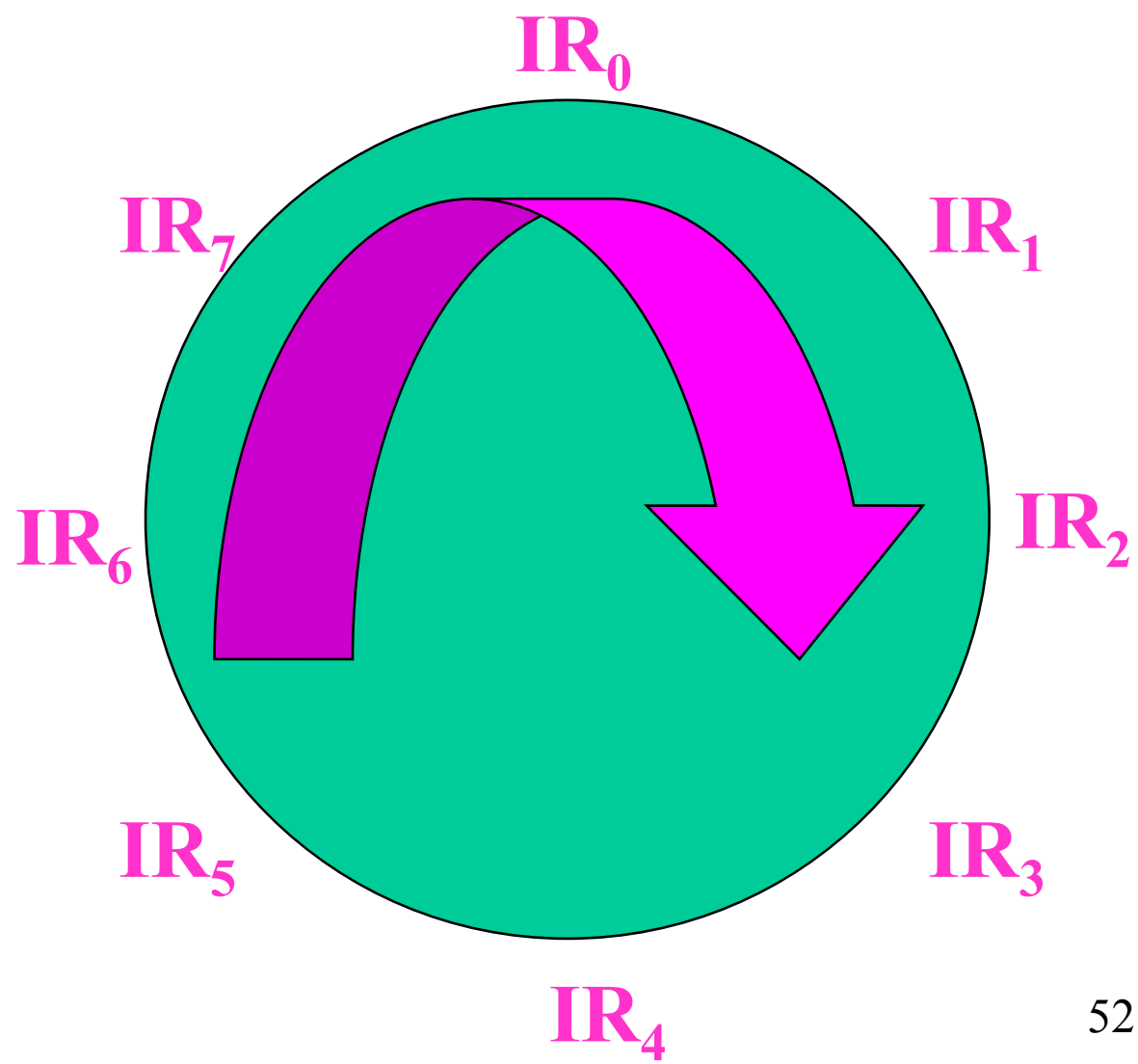
■ 中断优先级

1) 自动循环方式

自动按IR_0到IR_7的顺序旋转。

2) 特殊循环方式

用OCW2中的L_{0---2}来确定级别为最低的源。如：若设定IR_3为最低，则IR_4就被选为最高。



优先级例：

8259A的IR3和IR5分别连接有从片A和B，从片都采用全嵌套方式，主片中当前优先级从高到低位：IR2, IR3, IR4, IR5, IR6, IR7, IR0。分析该中断系统的优先级顺序。

从高到低顺序为：

主片IR2
从片A的IR0~IR7
主片IR4
从片B的IR0~IR7
主片IR6、IR7、IR0

■ 屏蔽中断源方式

1　普通屏蔽方式

通过对8259A的IMR的设置，屏蔽对应的IRi中断请求，屏蔽在8259A工作过程中可以随时通过OCW1设置和撤销

```
MOV   DX，201H
IN      AL，DX                    ; 读IMR

AND    AL，0F7H                   ; 开放IR3
OR     AL，22H                    ; 屏蔽IR1和IR5
OUT    DX，AL                     ; 写入IMR
```

2　特殊屏蔽方式

使8259A只对正在处理的中断申请进行屏蔽，比它优先权高或低的中断请求都可以响应，特殊屏蔽方式下，通过OCW3设置特殊屏蔽，再用OCW1设置屏蔽，则可以使IMRi为1的同时清除ISRi，达到 开放低优先级的中断的目的

特殊屏蔽例：

设8259A偶地址端口80H，奇地址端口81H，当前正服务IR4中断请求。

要求8259A可以开放低级中断中断高级中断处理程序，使用特殊屏蔽方式。

```
CLI                          ; 关中断
MOV   AL，68H                ; OCW3设置特殊屏蔽方式
OUT    80H，AL
IN       AL，81H             ; 读取系统原来的屏蔽字
OR      AL，10H              ; IR4对应的屏蔽位置1，新屏蔽字送8259A
OUT    81H，AL               ; 开中断
 STI
 …                           ; 继续执行IR4服务程序，已开放其它中断，如有IR7请求，可以响应

 CLI
 IN       AL，  81H          ; 读取系统屏蔽字
 AND    AL，0EFH             ; IR4对应的屏蔽位置0，新屏蔽字送8259A
OUT    81H，AL               ; 恢复系统原来的屏蔽字
MOV   AL，48H
OUT    80H，AL               ; OCW3撤销特殊屏蔽方式
STI                          ; 开中断
 …
```

■ 中断结束命令 ---------使ISR中的相应位复位

1 自动中断结束方式（AEOI）

8259A收到第二个/INTA的后沿，使ISR中的相应位复位，只能用于非中断嵌套方式。

2 非自动中断结束方式（EOI）

必须由软件用EOI命令复位，若为级连方式，则必须向主从芯片分别送EOI命令。在向从芯片送EOI命令后，必须检查从芯片中所有申请中断的源是否都已经服务了。

（1）一般的EOI命令

若8259A工作在全嵌套方式，则刚服务过的源就是中断优先级最高的源，可用一般的EOI命令复位。

（2）特殊的EOI命令

若8259A工作在特殊的全嵌套方式，8259A可能不能确定刚服务的源的等级，就要用特殊的EOI命令，此时OCW2中的$L_{0—2}$就是在ISR中要复位的编码。

级联方式下，一般不用自动中断结束方式，必须对主片和从片分别发送中断结束命令

主片中断结束：

```
MOV   AL，20H          ; OCW2设置中断结束
OUT   20H，AL
```

从片中断结束：

```
MOV   AL，20H          ; OCW2设置中断结束
OUT   0A0H, AL         ; 从片中断结束命令
OUT   20H，AL          ; 主片中断结束命令
```

中断结束命令例：

系统执行主程序时，IR2和IR4引脚同时出现中断请求，然后IR1有请求，最后IR3有请求

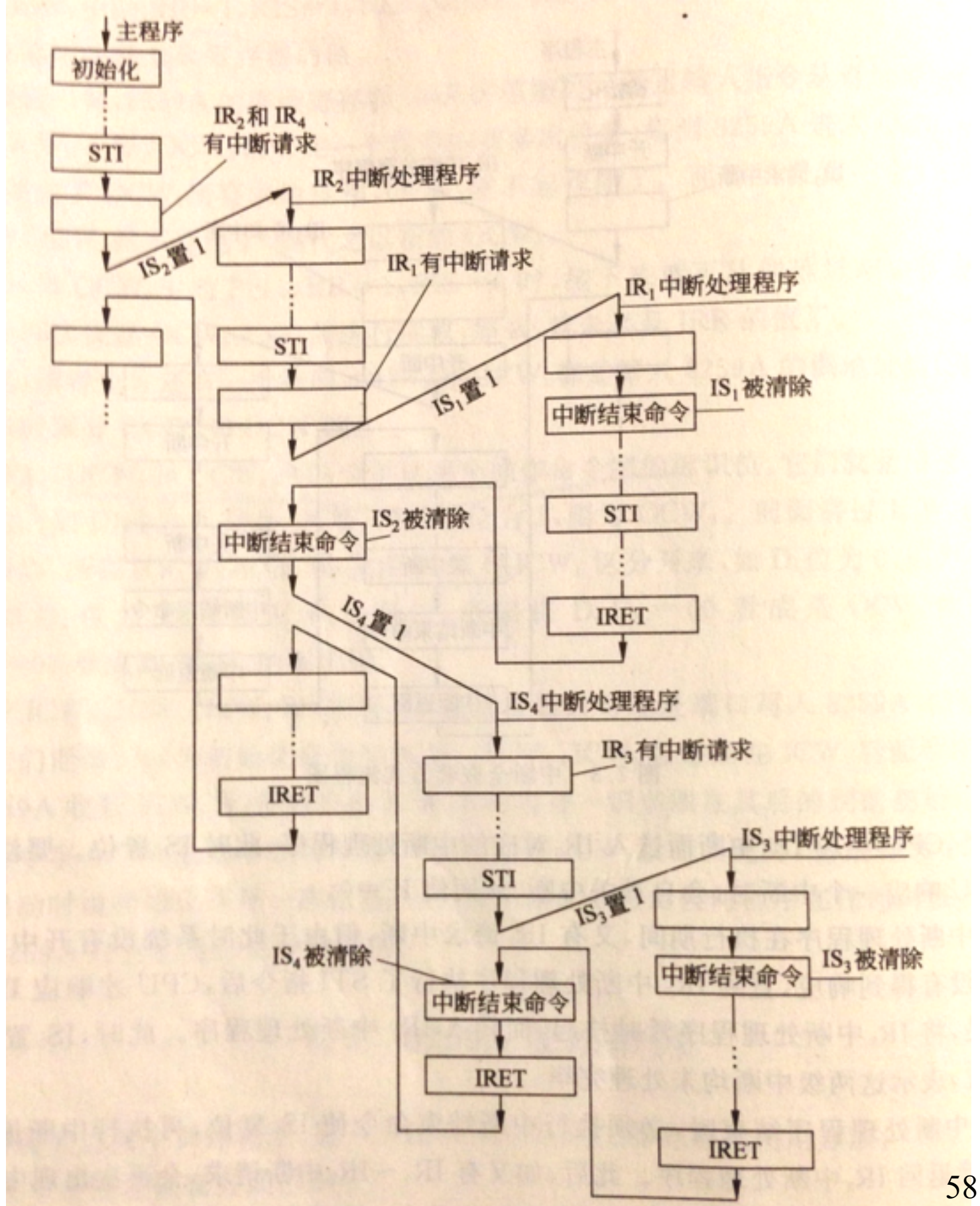


图 7.4　中断结束命令的使用

■ 连接总线方式

由ICW4设置，分为：

- ■ 缓冲方式

8259A通过总线驱动器接数据总线，$\overline{SP/EN}$输出，启动总线驱动器

- ■ 非缓冲方式

$\overline{SP/EN}$输入端，主片接高电平，从片接低电平

■ 引入中断请求方式

ICW1设置触发方式

- **■ 边沿触发**
- **■ 电平触发**

■ 查询方式

这种方式下，CPU必须把中断允许触发器置0，即：不允许外部中断，由CPU定时发出查询命令OCW3，然后从8259A中读出中断请求的优先级最高的序号。

8259A把查询信号后的/RD信号（/RD=0，/CS=0）作为CPU的响应信号来把ISR的相应位置1。

设置查询字例：

设8259A地址端口为80H，端口82H，设置中断查询方式并读入，设置读入ISR和IRR内容。

8 4 2 1
0 0 0 0 1 1 0 0

```
MOV   AL，0CH          ; OCW3设置查询方式
OUT   80H，AL
IN    AL，80H          ; 读取查询字

MOV   AL，0BH          ; OCW3设置读取ISR
OUT   80H，AL
IN    AL，80H          ; 读取ISR

MOV   AL，0AH          ; OCW3设置读取IRR
OUT   80H，AL
IN    AL，80H          ; 读取IRR
```

8259A小结：

- 8259A一般工作状态的设置——ICW1，ICW2设置
  - ◆触发方式
  - ◆单片/级联——如级联，ICW3设置
  - ◆中断类型码
- 优先级设置
  - ◆默认全嵌套方式——固定优先级，IR_0优先级最高
  - ◆特殊全嵌套方式——ICW4设置
  - ◆优先级自动和特殊循环——OCW2设置
- 中断结束方式设置
  - ◆自动结束——ICW4设置
  - ◆一般结束和特殊结束——OCW2设置，其中一般中断结束命令20H
- 优先级循环和中断结束一起设置
  - ◆OCW2设置
- 中断屏蔽——OCW1设置
- 读出IRR/ISR——先发OCW3

8259A编程：

□主程序工作

- ◆8259A初始化
- ◆中断向量置入中断向量表
- ◆根据需要送操作命令字

□ 中断服务程序工作

- ◆具体中断处理工作
- ◆如果中断嵌套，要开中断
- ◆通常需要保存寄存器信息到堆栈
- ◆IRET结束

8259A应用例

例1．2片8259级联，从片的INT接主片的IR2

- 假设端口地址：主片20H、21H，从片A0H、A1H
- 主片和从片均采用边沿触发；均为非自动结束方式
- 主片采用特殊全嵌套方式，从片是一般全嵌套方式
- 采用非缓冲方式，主片$\overline{SP/EN}$接 +5V,从片$\overline{SP/EN}$接地
- 主片的中断类型号为38H~3FH，从片的中断类型号为70H~77H
- 写出主8259A和从8259A的初始化程序段

初始化程序段如下：

```
I；---------- 主片8259A-----------
I M；---------- 从片8259A-----------
I O MOV  AL，11H        ；ICW1，边沿触发，多片，需ICW4
I M OUT  ICW1B，AL
I O MOV  AL，70H        ；ICW2，中断类型码
I M OUT  ICW2B，AL
I O MOV  AL，02H        ；ICW3，INT接主片的IR2
I M OUT  ICW3B，AL
    MOV  AL，01H        ；ICW4，非缓冲，普通全嵌套
  O                     ；非自动结束
    OUT  ICW4B，AL
```

例2：编程实现将8086系统中8259A的IRR，ISR，IMR三寄存器的内容读出，并送入从0080H开始的存储器保存。设8259A的端口地址为200H和201H。

解：

```
MOV   DI，0080H
MOV   DX，200H
MOV   AL，0AH
OUT   DX，AL          ; 00001010   偶地址口，写入OCW3，选择读IRR
IN    AL，DX
MOV   [DI]，AL
INC   DI              ; 读IRR，存入0080H
MOV   AL，0BH
OUT   DX，AL
IN    AL，DX          ; 00001011   偶地址口，写入OCW3，选择读ISR
MOV   [DI]，AL
INC   DX              ; 读ISR，存入0081H
IN    AL，DX
MOV   [DI]，AL
HLT                   ; 奇地址端口201H，读IMR
```

例3　试编程实现每按一次开关，使8086寄存器BL中的内容加一。设8259A的端口地址为20H和21H。

解：

```
MOV   AL，13H
OUT   20H，AL
MOV   AL，08H
OUT   21H，AL
MOV   AL，13H
OUT    21H，AL
MOV   AL，0F7H
OUT   21H，AL
CLI
XOR   AX,AX
 MOV  DS,AX
 MOV  AX,OFFSET  Intr
 MOV  [2CH],AX
 MOV  AX,SEG Intr
 MOV  [2EH],AX
 STI
 HLT

Intr：  INC    BL
        IRET
```