


第 6 章 中断技术



第 6 章 中断技术



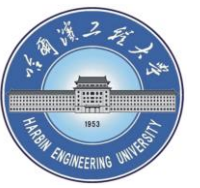
6.1 中断的基本概念

6.2 微机中断系统

6.3 可编程中断控制器8259A

6.4 8259A的应用举例

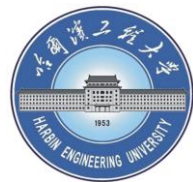
学习目的



通过对本章的学习，应该能够达到下列要求：

- 中断的概念及中断的用途
- 中断类型及中断向量
- 中断处理过程
- 8259A控制器编程结构
- 8259A控制器的应用

学习目的



重点

- 中断的概念
- 中断处理过程
- 硬件中断与软件中断的比较
- CPU响应可屏蔽中断的条件
- 8259A芯片的结构和命令
- 8259A芯片应用编程

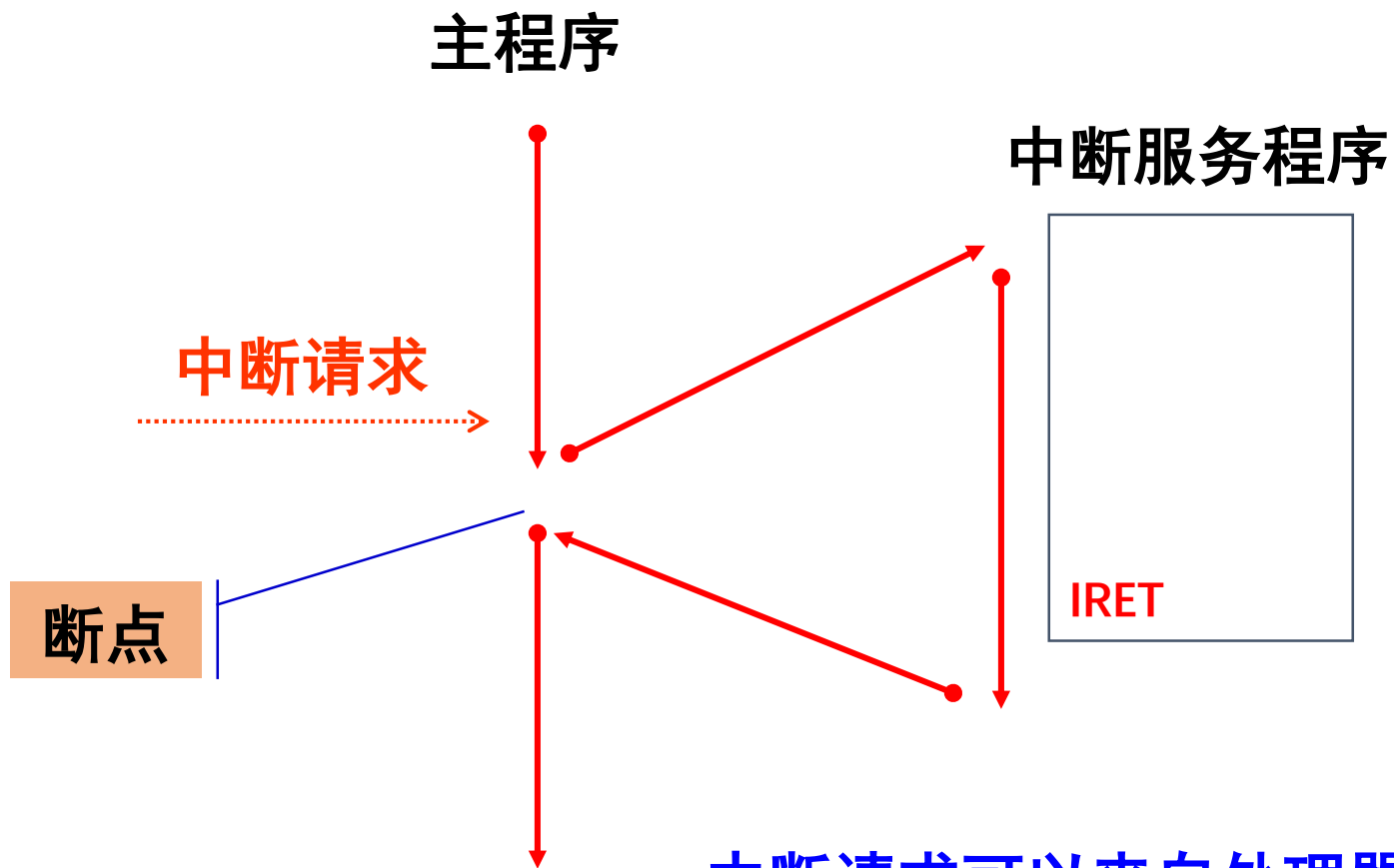


6.1 中断的概念

6.1.1 为什么要用中断？

- **中断**：CPU在正常运行程序时，由**内部/外部随机事件**或由**程序预先安排的事件**，引起CPU暂时中断正在运行的程序，而转到为事件服务的程序中，服务完毕，在返回原程序继续执行，就称为一个**中断过程**。
- **中断的实质就是程序的转移。**
- 中断源的**中断申请**是触发或引起这种程序转移的原因。

6.1 中断的概念



中断请求可以来自处理器外部的中断源，也可以由处理器执行指令引起：例如，执行INT n指令。



6.1 中断的概念

6.1.2 中断源与中断识别

- **1.中断源：**
 - (1) 数据输入/输出设备请求中断
 - (2) 定时时间到申请中断
 - (3) 满足规定条件申请中断
 - (4) 电源掉电
 - (5) 故障报警申请中断
 - (6) 程序调试设置中断
- **2.中断识别：**
 - **目的：**形成该中断服务程序的入口地址。



6.1 中断的概念

6.1.3 中断向量与中断向量表

- **中断类型码**：8086中断源共分256级中断，每个中断源对应一个0~255编号，称为**中断类型码或中断向量号**。
- **中断向量**：中断服务程序的**入口地址(CS, IP)**
- **中断向量表**：将所有中断向量集中放在一起。
- **中断向量表**：放在内存地址**0单元**开始的单元：**0000H~ 03FFH**。
- **中断向量指针**：

中断类型号*4=中断向量最低字节指针



6.1 中断的概念

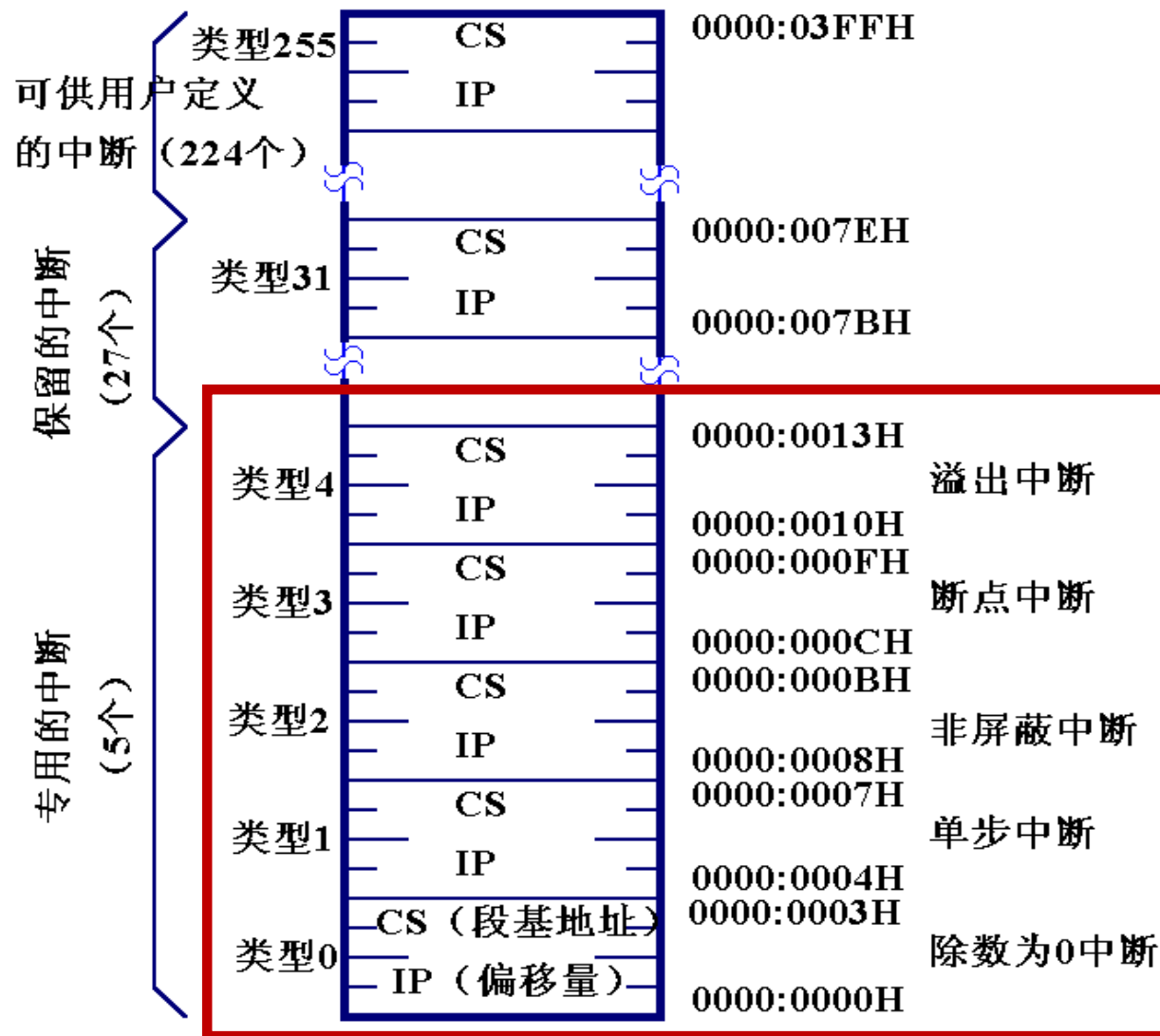
6.1.4 中断类型号与中断向量指针

- 共**256**个类型号
- 一个中断源对应一个类型号
- **8086**中断向量由四个字节组成：
 - 2个**高**地址字节：存放中断服务程序的代码段的**段基址**；
 - 2个**低**地址字节：存放中断服务程序的**偏移地址**。

6.1 中断的概念

■ 中断向量表

8086系统的**中断类型号**是**固定不变的**，而**中断类型号**对应的**中断向量**是可以改变的，即一个中断号所对应的中断服务不是唯一的。





6.1 中断的概念

例：中断类型为**20H**：

中断向量放在0000: 0080H开始的**4**个单元中。

中断向量**指针**= $20H * 4 = 80H$

20H号中断服务程序的**入口地址**:**4030H:2010H**

0000:	0080	10	CS=4030H IP=2010H
0000:	0081	20	
0000:	0082	30	
0000:	0083	40	



6.1 中断的概念

6.1.5 中断识别及优先级确定

■ 中断识别方法

- 软件查询识别
- 中断向量识别 (8086采用)

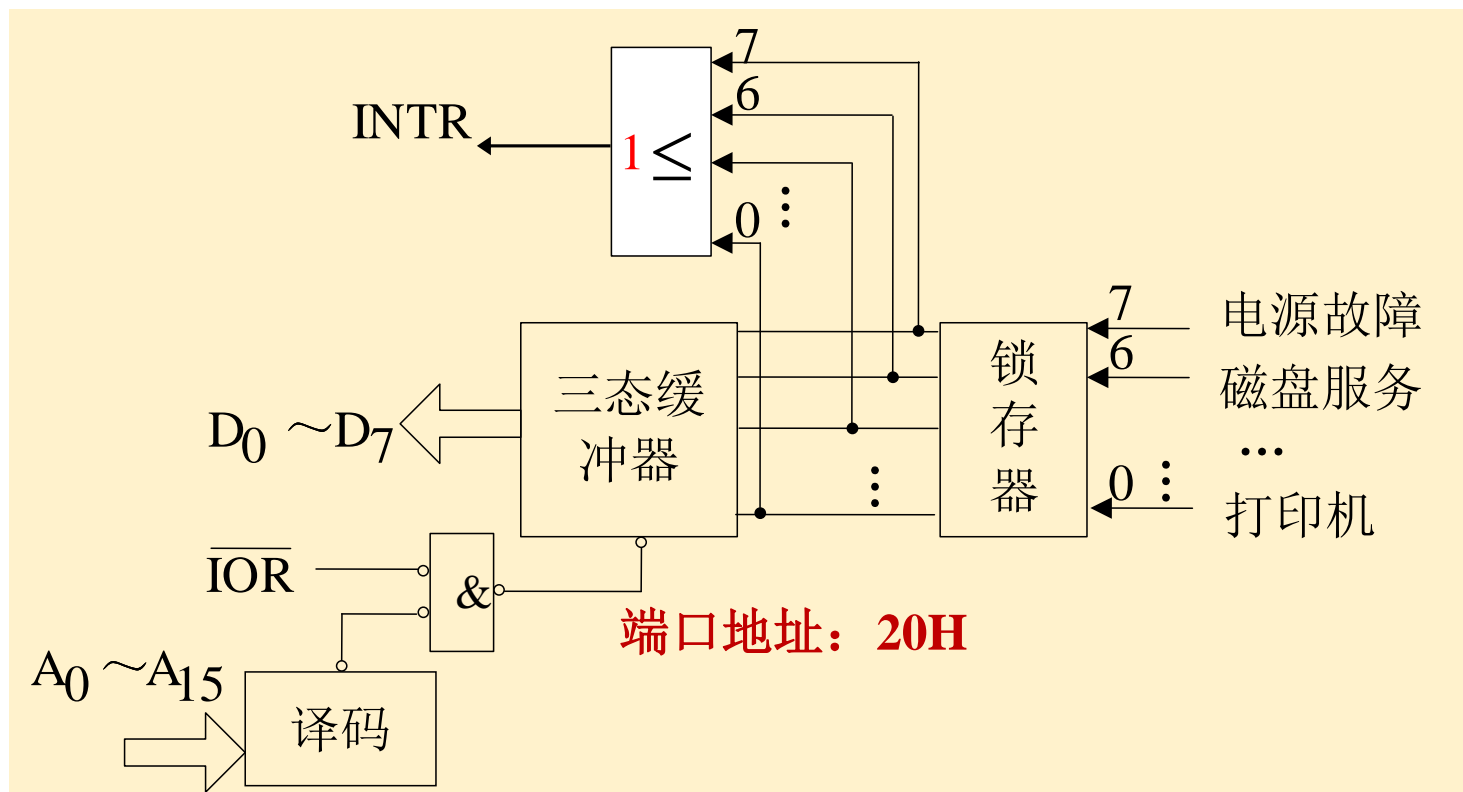
■ 确定中断优先级方法

- 软件查询
- 硬件排队电路

6.2 微机中断系统

软件查询中断方式与硬件优先权排队电路

1. 软件查询优先权





6.2 微机中断系统

例 软件查询中断程序

```
IN      AL, 20H      ; 读中断触发器状态
TEST   AL, 80H      ; 有电源故障?
JZ     B1           ; 没有, 继续查询
JMP    PWF         ; D7=1, 转电源故障处理
B1:  TEST AL, 40H  ; 有磁盘服务申请?
JZ     B2           ; 没有, 继续查询
JMP    DISS        ; D6=1, 转磁盘服务
B2:  TEST AL, 20H  ; 有磁盘服务申请?
JZ     B3           ; 没有, 继续查询
JMP    MT          ; D5=1, 转磁带服务
B3:  .....
```



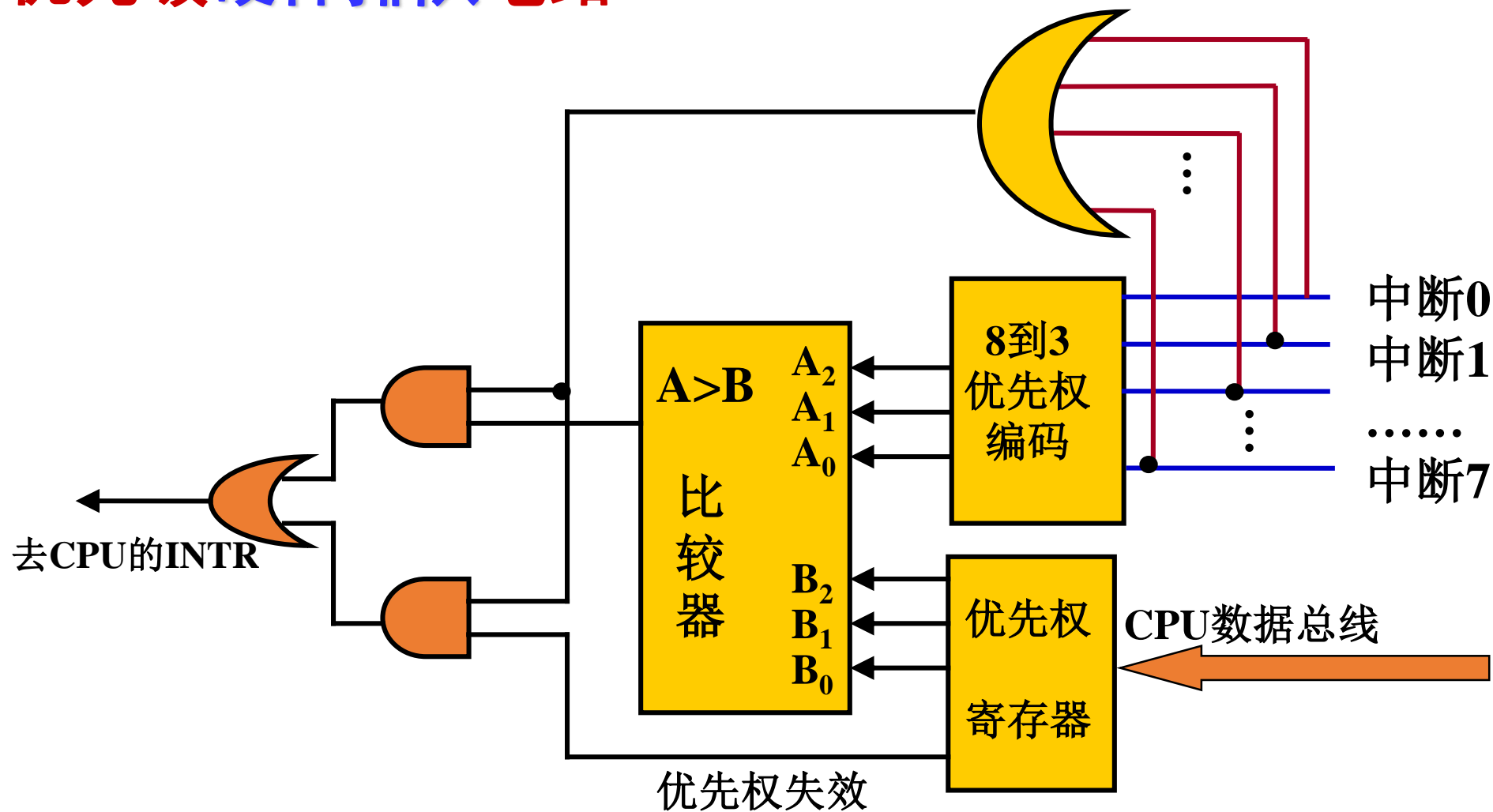
6.2 微机中断系统

软件查询优先权优点：

- (1) 询问次序就是优先权次序；
- (2) 完成中断源识别；
- (3) 省硬件，不需要优先权排队电路。

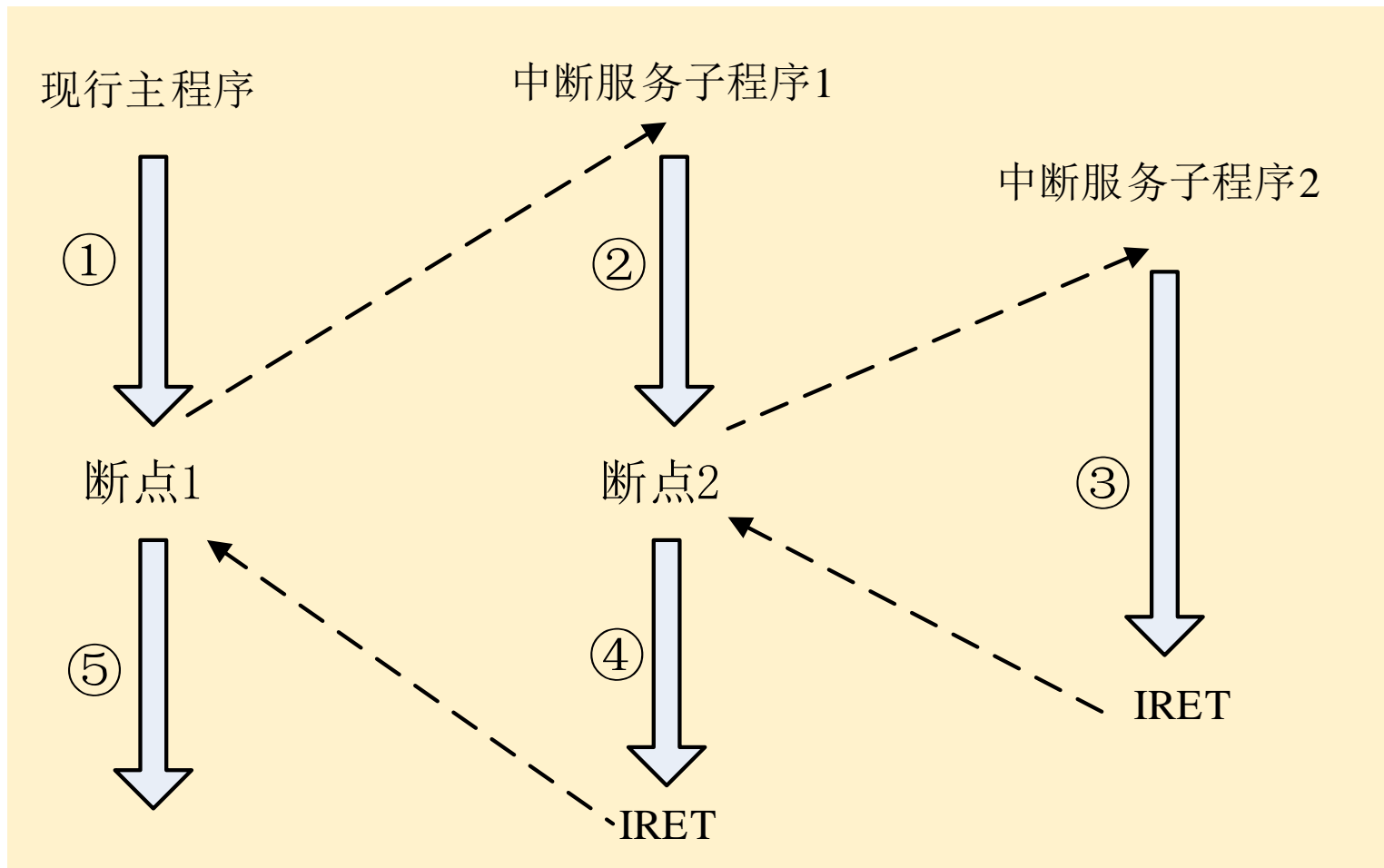
6.2 微机中断系统

2、优先级硬件排队电路



6.2 微机中断系统

6.1.6 中断嵌套

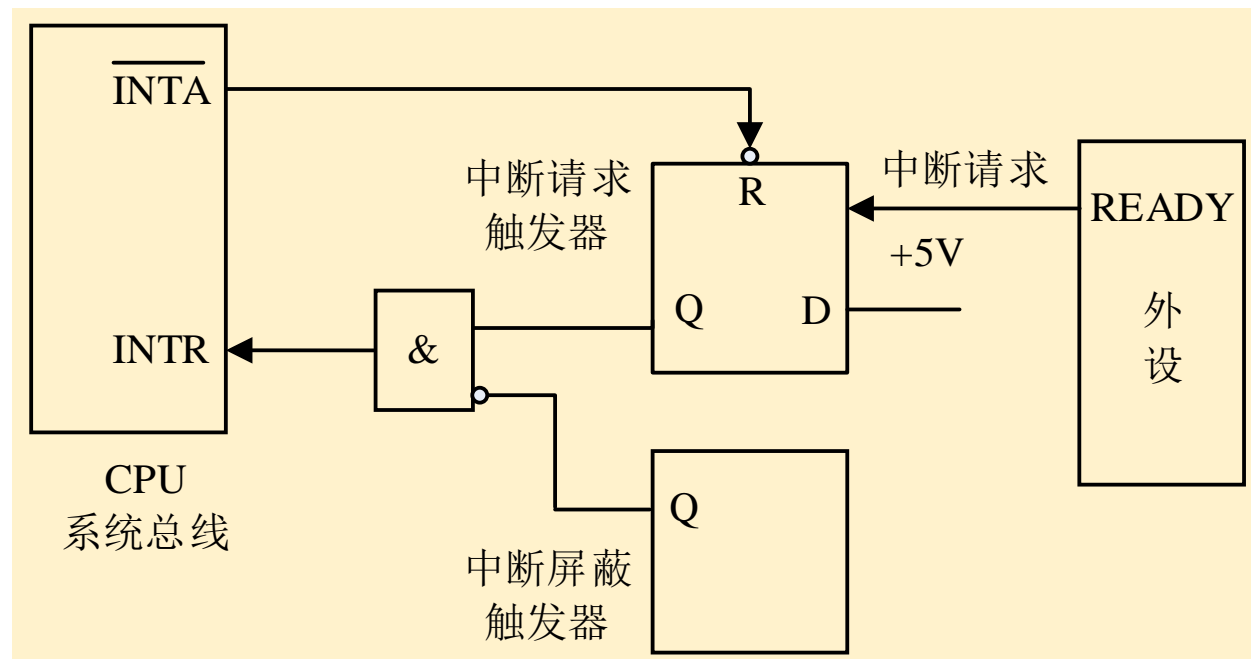


6.2 微机中断系统

6.2.1 中断系统组成

1. 与中断有关的触发器:

中断请求触发器， 中断屏蔽触发器， CPU内部的中断允许触发器。





6.2 微机中断系统

2. 微机中断系统的功能

- (1) 中断响应
- (2) 断点保护和中断处理
- (3) 中断优先权排队
- (4) 中断嵌套



6.2 微机中断系统

3. CPU响应外部可屏蔽中断的条件

- (1) 有中断源发出的中断请求信号
- (2) 开放总中断： $IF=1$
- (3) 在现行指令结束后响应中断



6.2 微机中断系统

6.2.2 8086的中断系统

- 8086的中断系统采用中断向量机制
- 能够处理256个中断
- 用中断向量号0 ~ 255区别
- 可屏蔽中断还需要借助专用中断控制器Intel 8259A实现
优先权管理



6.2 微机中断系统

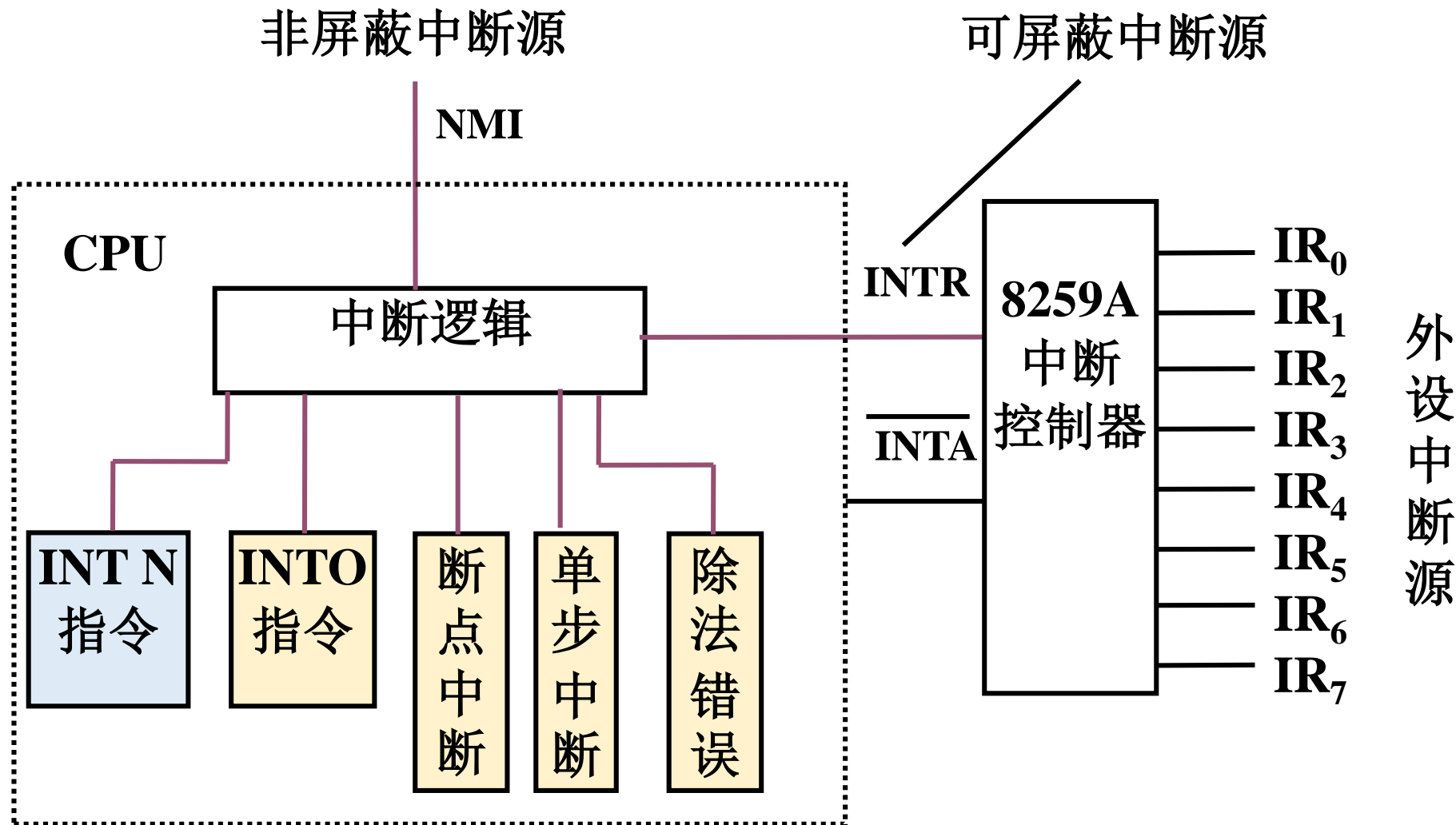
6.2.3 8086的中断方式

中断的分类:

- **软件中断**:与硬件无关, 不执行中断响应周期, 内部中断不可屏蔽, 优先级高于外部中断。根据情况又分为被动(溢出中断、除零中断等)和主动(INT指令)。
- **硬件中断**:外部中断
 - **非屏蔽中断 (NMI)**。由NMI引脚产生的中断。它不受IF的控制。
 - **可屏蔽中断 (INTR)**。由INTR引脚产生的中断。它受IF的控制。

6.2 微机中断系统

中断的分类





6.2 微机中断系统

■ 硬件中断—外部中断

- **硬件中断**是外设提出中断请求引起的程序中断;
- 利用外部中断, 8086系统可以实时响应外部设备的数据传送请求, 能够及时处理外部紧急事件;
- 外部中断的产生是**随机**的;
- 8086/8088CPU有两条引脚
 - **非屏蔽中断**: 由正跳变NMI引入, **不受中断标志位IF的控制**。
 - **可屏蔽中断**: 由高电平有效INTR引入, 可屏蔽中断是由用户定义的外部硬件中断, **受中断标志位IF的控制**。



6.2 微机中断系统

■ NMI非屏蔽中断

- 发出的中断由**NMI**引入，CPU立即响应
- 优先级在**硬件中断**中是最高的
- 处理重大事故
- 非屏蔽中断类型为：**02H**

■ 可屏蔽中断

- 由外设发出的中断由**INTR**引入
- 当**IF=1**时，CPU将响应这一中断请求
- **IF**的清除与设置，可由指令完成



6.2 微机中断系统

■ 软件中断—内部中断

- 内部中断又称软件中断。是由CPU内部事件引起的中断。包括除法出错中断0、单步中断1、断点中断3、溢出中断4这四个由内部硬件设置或自动引发的中断和指令设置的中断。
- 利用内部中断，微处理器为用户提供了发现、调试并解决程序执行时异常情况的有效途径。



6.2 微机中断系统

硬件中断与软件中断的比较：

- 软件中断是**确定或者随机的**，硬件中断是**随机的**
- 硬件中断需要执行总线周期，中断类型码由**中断控制器**提供；软件中断类型码**自动形成**，不需要执行总线周期
- 软件中断的**一号中断**受标志寄存器中另外一个标志既**TF**的影响，只有**TF为1**时，才能执行单步中断。



6.2 微机中断系统

各种中断源的优先权：

- 软件中断
 - 除法错中断
 - 指令中断
 - 溢出中断
- 非屏蔽中断
- 可屏蔽中断
- 单步中断

高

低





6.2 微机中断系统

6.2.4 中断处理过程

1. 中断申请
2. 中断响应
3. 中断服务程序
4. 中断返回



6.2 微机中断系统

1. 中断申请

当外部设备要求CPU为它服务时，都要发送一个“**中断请求**”信号给CPU进行中断申请，CPU在**执行完每条指令**后去检查“中断请求”输入线，看是否有外部发来的“中断请求”。CPU对外部的中断申请有权决定是否予以相应。



6.2 微机中断系统

2. 中断响应

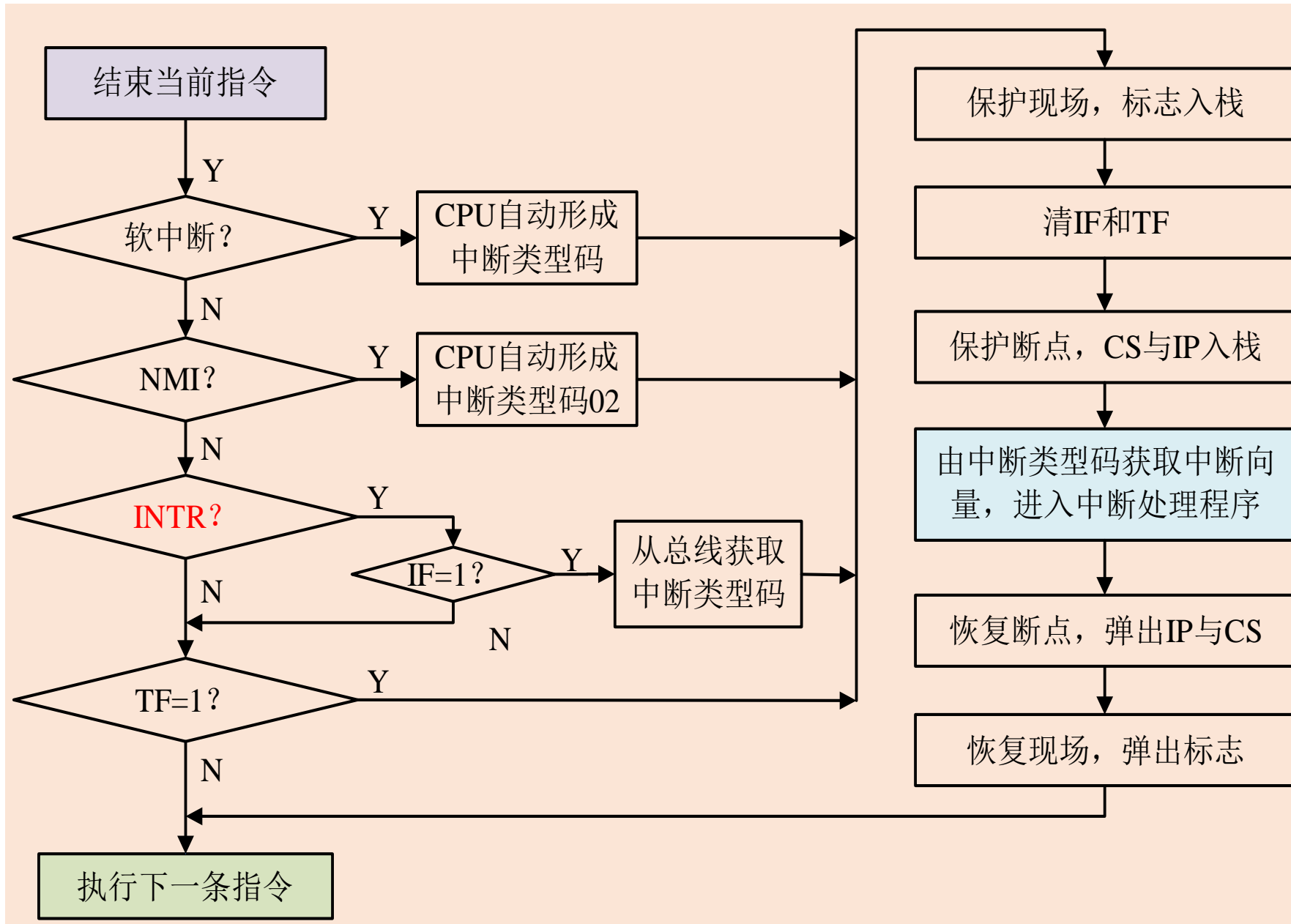
CPU收到中断请求后，如果中断开放，且没有其他设备申请DMA传送，则CPU在当前指令结束时响应中断。CPU响应中断前，进行断点及标志保存，即将当前正在执行的程序的段地址（CS）、偏移地址（IP）和标志（PSW）压入堆栈。然后通过响应周期中读取的中断类型号，找到中断源的中断服务程序入口地址，分别将它们装入CS和IP寄存器中，然后进入中断服务程序并开始执行。



6.2 微机中断系统

2. 中断响应

- 软中断：类型码为**0、1、3、4**的专用软中断已经指定了类型码，而通过执行中断指令INT n实现的软中断，类型码有指令给出的立即数n，CPU在取指时就获得了中断类型码。





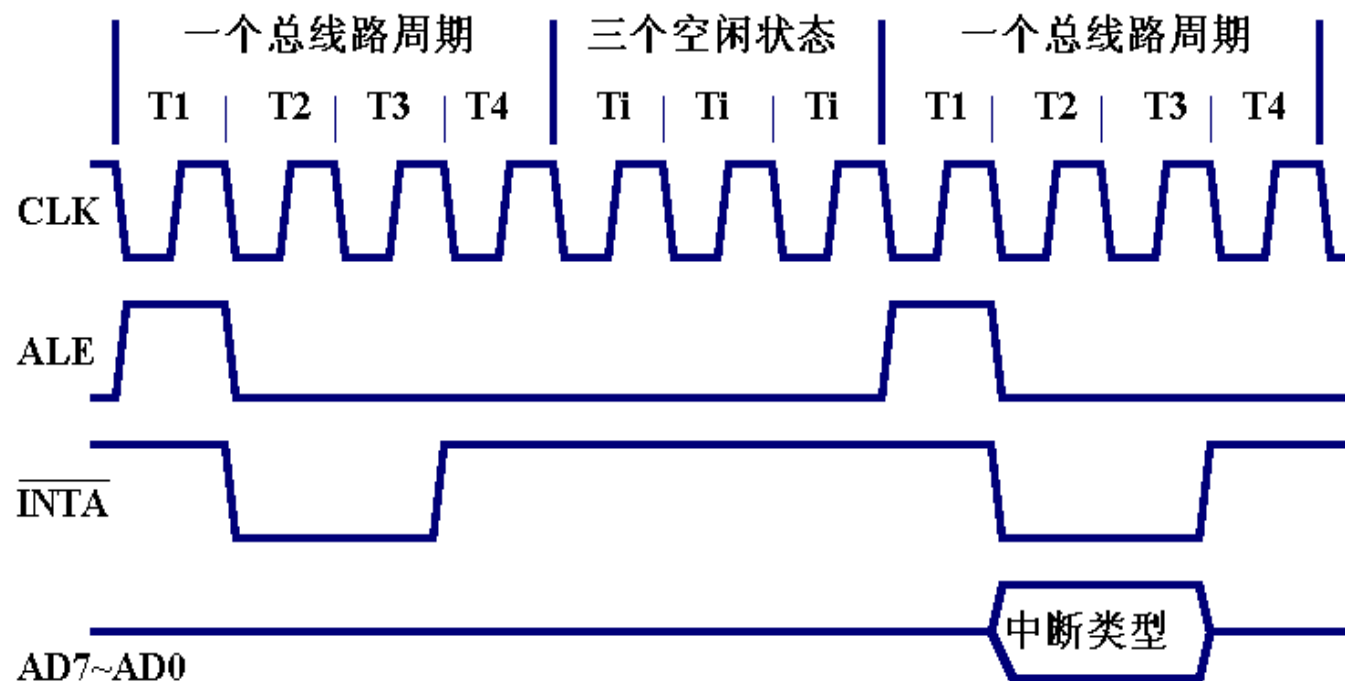
6.2 微机中断系统

- **NMI中断**：类型码为2，作为8086保留中断，有固定的中断类型码，当中断发生时，CPU自动形成中断类型码。
- **INTR中断**：可屏蔽中断INTR是专门提供给外设使用的中断源，由用户开发，对应的中断类型码也由用户确定。该类中断类型码CPU必须通过总线从外设获取，因此CPU必须执行总线周期。



6.2 微机中断系统

- 中断响应周期
- 在**第一个中断响应周期**，CPU送出中断响应信号，数据线浮空。
- 在**第二个中断响应周期**；被响应的外设数据线送一个字节的**中断类型码**，CPU采样数据线得到中断类型码。





6.2 微机中断系统

3. 中断服务程序

中断服务程序的功能与**中断源**的期望一致。中断服务程序在开头会将可能使用的寄存器中的内容一一进栈，这叫**保护现场**。中断处理完毕后，要把已入栈的寄存器内容弹出，还给各相应寄存器，这叫**恢复现场**。

如果要在中断服务处理程序执行当中响应更高优先级的中断请求，就要在保护现场后或恢复现场后写一条**开中断指令**。



6.2 微机中断系统

4. 中断返回

中断服务程序结束，执行“**中断返回**”，就会自动将保存在堆栈中的标志及被中断的程序断点弹出（**依次弹出6个字节为IP、CS和PSW**），并装入程序段地址（CS）寄存器和偏移地址（IP）寄存器，这叫恢复程序断点，使程序又回到中断前的地址继续执行。



6.2 微机中断系统

总结：CPU中断处理步骤

- 关中断；（提前保存PSW）
- 断点保护；
- 形成中断入口地址；
- CPU开放中断
- 中断服务程序
- CPU关中断
- 恢复现场
- 中断返回



6.2 微机中断系统

6.2.6 中断向量表的装入与修改

例 假设中断服务程序入口符号地址为**INT-SEV**，**N**为中断类型码，用MOV指令，中断向量的装入参考程序。

中断向量指针： $20\text{H} \times 4 = 0010\ 000000 = 80\text{H}$

$20\text{H} \times 4 + 2 = 82\text{H}$

中断服务程序框架：

```
INT-SEV PROC NEAR
```

```
.....
```

```
IRET
```

```
INT-SEV ENDP
```



6.2 微机中断系统

■ 中断向量的装入

CLI

PUSH DS

XOR AX, AX

MOV DS, AX

MOV BX, N*4

MOV AX, **OFFSET INT-SEV** ; 取中断服务程序的偏移地址

MOV WORD PTR **[BX]**, AX

MOV AX, **SEG INT-SEV** ; 取中断服务程序的段地址

MOV WORD PTR **[BX+2]**, AX

POP DS

STI



6.2 微机中断系统

■ 中断向量的修改

- 1)用直接装入的方法：适用于单板机
- 2)用DOS系统功能调用：用于PC机

- 取原中断向量：**INT 21H/35H**

中断向量=ES: BX

类型号=AL

- 装入中断向量：**INT 21H/25H**

中断向量=DS: DX

类型号=AL



6.2 微机中断系统

AH	功能	调用参数	返回参数
00	程序终止(同INT 20H)	CS=程序段前缀	
01	键盘输入并回显		AL=输入字符
02	显示输出	DL=输出字符	
03	异步通讯输入		AL=输入数据
04	异步通讯输出	DL=输出数据	
05	打印机输出	DL=输出字符	
06	直接控制台I/O	DL=FF(输入) DL=字符(输出)	AL=输入字符
07	键盘输入(无回显)		AL=输入字符
08	键盘输入(无回显) 检测Ctrl-Break		AL=输入字符

21	随机读	DS:DX=FCB首地址	AL=00 读成功 =01 文件结束 =02 缓冲区溢出 =03 缓冲区不满
22	随机写	DS:DX=FCB首地址	AL=00 写成功 =01 盘满 =02 缓冲区溢出
23	测定文件大小	DS:DX=FCB首地址	AL=00 成功(文件长度填入FCB) AL=FF 未找到
24	设置随机记录号	DS:DX=FCB首地址	
25	设置中断向量	DS:DX=中断向量 AL=中断类型号	
26	建立程序段前缀	DX=新的程序段前缀	
27	随机分块读	DS:DX=FCB首地址 CX=记录数	AL=00 读成功 =01 文件结束 =02 缓冲区太小,传输结束 =03 缓冲区不满
28	随机分块写	DS:DX=FCB首地址 CX=记录数	AL=00 写成功 =01 盘满 =02 缓冲区溢出



6.2 微机中断系统

2E	置磁盘自动读写标志	AL=00 关闭标志 AL=01 打开标志	
2F	取磁盘缓冲区的地址		ES:BX=缓冲区首址
30	取DOS版本号		AH=发行号,AL=版本
31	结束并驻留	AL=返回码 DX=驻留区大小	
33	Ctrl-Break检测	AL=00 取状态 =01 置状态(DL) DL=00 关闭检测 =01 打开检测	DL=00 关闭Ctrl-Break检测 =01 打开Ctrl-Break检测
35	取中断向量	AL=中断类型	ES:BX=中断向量
36	取空闲磁盘空间	DL=驱动器号 0=缺省,1=A,2=B,...	成功:AX=每簇扇区数 BX=有效簇数 CX=每扇区字节数 DX=总簇数 失败:AX=FFFF
38	置/取国家信息	DS:DX=信息区首地址	BX=国家码(国际电话前缀码) AX=错误码



6.2 微机中断系统

例 用INT 21H/25H功能号设置新中断向量:

MOV AH, 35H ; 取原中断向量

MOV AL, nH

INT 21H

MOV OLD_SEG, ES ; 保存原中断向量

MOV OLD_OFF, BX ;

.

.

MOV AH, 25H ; 设置新中断向量

MOV AL, nH

MOV DX, SEG_INTR

MOV DS, DX ; DS指向新中断程序段址

MOV DX, OFFSET_INTR ; DX指向新中断程序偏移地址

INT 21H



6.2 微机中断系统

```
MOV AH, 25H ; 恢复原中断向量  
MOV AL, nH  
MOV DX, OLD_SEG  
MOV DS, DX  
MOV DX, OLD_OFF  
INT 21H
```



6.3 中断控制器8259A

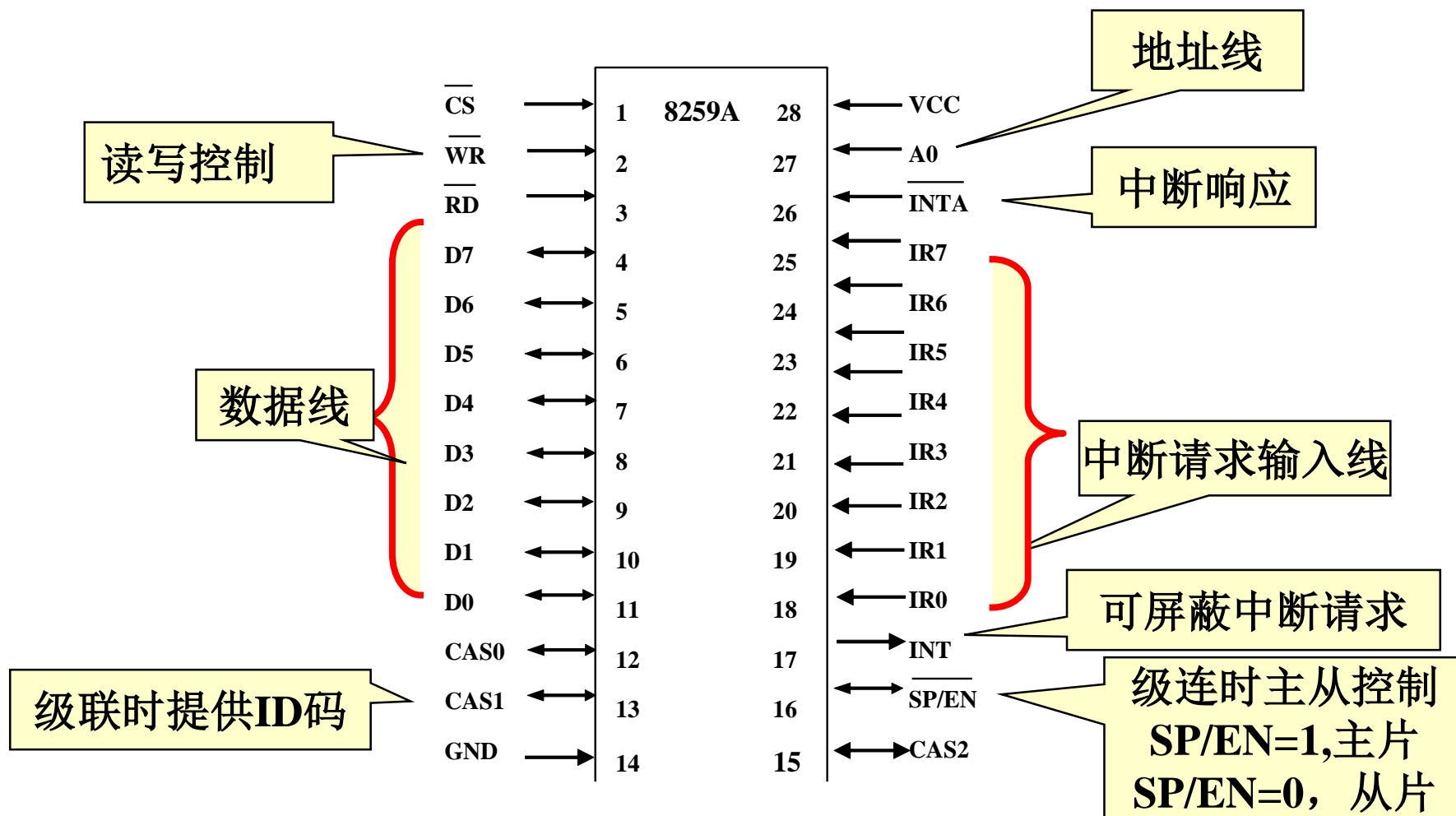
6.3.1 8259A概述

- 8259A的功能：
 - 中断优先权管理
 - 接受中断请求
 - 提供中断类型号
 - 具有屏蔽与开放中断的功能
- 8259A的特点：
 - 单片8259可处理**8级**中断
 - 级联时：最多可用**9片**处理**64级**中断
- 8259A的片内寻址
 - A0=1、0，**两个端口地址**



6.3 中断控制器8259A

6.3.2 8259A的引脚





6.3 中断控制器8259A

- ① $D_0 \sim D_7$: 双向数据总线。CPU用来传送命令，接收状态和读取中断向量。
- ② $CAS_0 \sim CAS_2$: 级联总线，双向。主控8259A与从控8259A的连接线。主控时该总线为输出，从控时则为输入。
- ③ $IR_0 \sim IR_7$: 外设中断请求线，输入。从外设送来的中断请求由这些引脚输入到8259A。在边沿触发方式中IR输入应有由低到高的上升沿，此后保持为高，直到被响应。在电平触发方式中，IR输入应保持高电平直到被响应为止。
- ④ INT : 中断申请线，输出。8259A中断申请。当8259A收到从外设经IR脚送来的中断请求时，由它输出高电平，对CPU提出中断申请。该引脚连接CPU的INTR引脚。



6.3 中断控制器8259A

- ⑤ **SP/EN**: 主从定义/缓冲器方向引脚，是一个双功能引脚。在非缓冲方式中用作输入线，指定8259A为主控制器 (=1) 或是从控制器 (=0)。在缓冲方式中用作输出线，控制缓冲器的接收/发送。
- ⑥ **INTA**: 中断响应引脚，输入。接收CPU发送来的2个中断响应脉冲，第一个用来通知8259A，其中断请求已被响应；第二个作为读操作信号，读取8259A所提供的中断类型号。
- ⑦ **CS**: 片选信号，输入，低电平有效。



6.3 中断控制器8259A

- ⑧ **WR**: 写信号, 输入, 低电平有效。
- ⑨ **RD**: 读信号, 输入, 低电平有效。
- ⑩ **A0**: 地址线, 输入。该引脚与片选、读/写信号联合使用, 作为对8259A内部端口寻址。
- ⑪ **Vcc和GND**: 8259A的+5V电源和地线。



6.3 中断控制器8259A

6.3.3 8259A的编程

■ 初始化编程

- 8259A开始工作前，必须进行初始化编程
- 给8259A写入初始化命令字ICW

■ 中断操作编程

- 在8259A工作期间
- 可以写入操作命令字OCW将选定的操作传送给8259A，使之按新的要求工作
- 还可以读取8259A的信息，以便了解他的工作状态

6.3 中断控制器8259A

■ 8259A的芯片内部结构

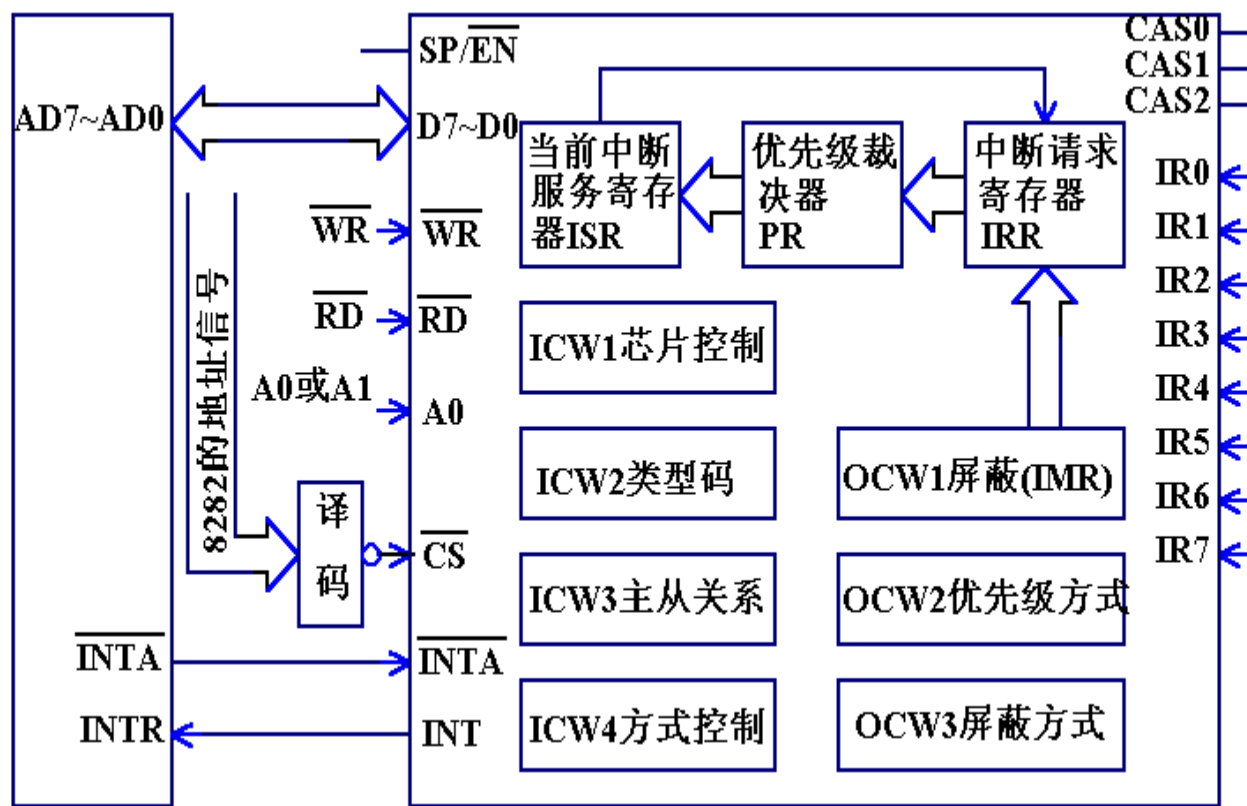


图 编程结构图



6.3 中断控制器8259A

■ 中断请求寄存器IRR

- 保留8条中断请求线 $IR_7 \sim IR_0$ 的中断请求状态。
- D_i 位为1表示 IR_i 引脚有中断请求；为0表示无请求

■ 中断服务寄存器ISR

- 保存正在被8259A服务着的中断状态
- D_i 位为1表示 IR_i 中断正在服务中；为0表示没有被服务



6.3 中断控制器8259A

■ 中断屏蔽寄存器IMR

- 对应位为**1**表示此中断已加屏蔽，为**0**表示此中断未加屏蔽，可通过进入中断优先级裁决器**PR**作裁决。

■ 优先权分析器PR

- 对新进入的中断请求和ISR中当前正在处理的中断比较，从而决定哪一个优先级更高，是否使8259A的输出端INT为1。



6.3 中断控制器8259A

■ 8259A的编程结构

- $ICW_1 \sim ICW_4$ 这4个寄存器为一组，用来放**初始化命令字**。
- $OCW_1 \sim OCW_3$ 这3个寄存器为一组，用来容纳**操作命令字**。
- 编程结构图的上半部分是8259A的处理部件。有**IRR**，**PR**和**ISR**组成。其功能是接受和处理从引脚 $IR_7 \sim IR_0$ 进入的中断。



6.3 中断控制器8259A

■ 8259A寄存器及I/O端口的识别

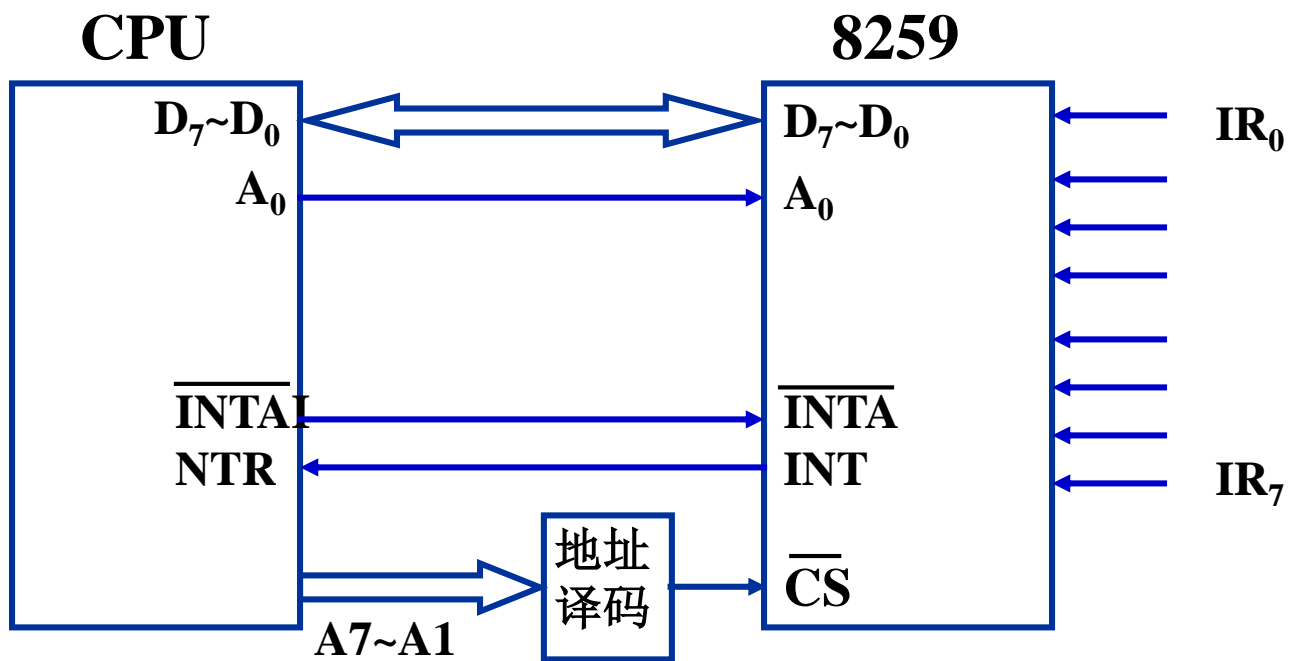
表 8259A读写操作及地址

CS	RD	WR	A0	功能	8259A端口	PC/XT机端口
0	0	1	0	读IRR, ISR	偶地址	20H
0	0	1	1	读IMR	奇地址	21H
0	1	0	0	写ICW1, OCW2, OCW3	偶地址	20H
0	1	0	1	写ICW2, ICW3, ICW4, OCW1	奇地址	21H
0	1	1	×	无操作		
1	×	×	×	无操作		

6.3 中断控制器8259A

■ 端口地址的确定

假设： 8259A的端口地址为: **20H, 21H**, 某8位CPU和8259A的地址线如何连接?



A7~A0:

0010 000**0** 20H

0010 000**1** 21H



6.3 中断控制器8259A

- 8259A应用于8086系统，在中断响应周期向CPU输出的内容
 - 第一个中断响应周期：AD15-AD0浮空；8259不向CPU输出。
 - 第二个中断响应周期：8259A向CPU送一个字节的**中断向量类型**。

D7	D6	D5	D4	D3	D2	D1	D0
T7	T6	T5	T4	T3			

- T7~T3由8259初始化编程 (**ICW2**) 确定
- **低三位**由8259根据中断请求输入IR0~IR7自动插入
- 低三位**000~111**分别代表：**IR0~IR7**



6.3 中断控制器8259A

■ 问题：

- 一片8259A可以管理多少级中断？
- 8259A片内端口地址有几个？
- 中断屏蔽寄存器IMR的作用是什么？
- 中断请求寄存器IRR的作用是什么？
- 中断服务寄存器ISR的作用是什么？

■ 进一步：

- 8259A的工作方式是怎样的？
- 如何进行8259A芯片的初始化编程？



6.3 中断控制器8259A

6.3.4 8259A的工作方式

- **8259A的工作方式有如下几类：**
 - 引入中断请求的方式
 - 中断优先级设置方式
 - 屏蔽中断源的方式
 - 中断结束处理方式
 - 连接系统总线的方式
 - 级联工作方式



6.3 中断控制器8259A

1. 中断触发方式

- **边沿触发方式**：此方式由ICW1来设置。
- **电平触发方式**：当中断输入端出现1个中断请求并得到响应后，输入端必须及时撤除高电平。
- **中断查询方式**：CPU用软件查询是否有中断请求？

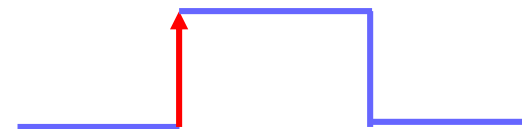
2. 连接总线方式

- **缓冲方式**
- **非缓冲方式**

6.3 中断控制器8259A

■ 边沿触发方式

- 8259A将中断请求输入端出现的上升沿作为中断请求信号。IR线以正跳沿向请求中断，在上升沿后可一直维持高电平，不会再产生中断。



■ 电平触发方式

- 中断请求端出现的高电平是有效的中断请求信号。IR线以高电平向8259A请求中断，但在响应中断后必须及时清除该高电平，以免引起第二次误中断。





6.3 中断控制器8259A

3. 中断屏蔽方式

- 普通屏蔽方式(重点掌握)

用OCW1命令将中断屏蔽寄存器IMR的相应位置“1”。

- 特殊屏蔽方式

用于中断处理程序中。当要允许优先级低的中断进入，可采用特殊屏蔽方式。可用OCW3的 $D_6D_5=11$ 设置，并将IMR对应于本级中断的位置1来实现。此时除IMR中置1位对应的中断被屏蔽外，其它置0位对应的中断无论优先级高低都可得到响应。



6.3 中断控制器8259A

4. 优先级的管理方式

- 一般完全嵌套方式(重点掌握)
- 特殊完全嵌套方式(重点掌握)
- 优先级自动循环方式
- 优先级特殊循环方式



6.3 中断控制器8259A

■ (1) 一般完全嵌套方式：

- 在对8259进行初始化后，没有设置其它优先级方式，则自动按此方式工作。
- **特点：**中断请求按优先级IR0~IR7级进行处理，IR0级中断的优先级最高。当一个中断被响应时，ISR中的对应位IS_n被置1，中断类型码被放到数据总线上，然后进入中断服务程序。

一般情况下（除了中断自动结束方式外），在CPU发出中断结束命令（EOI）前，此对应位一直保持“1”。

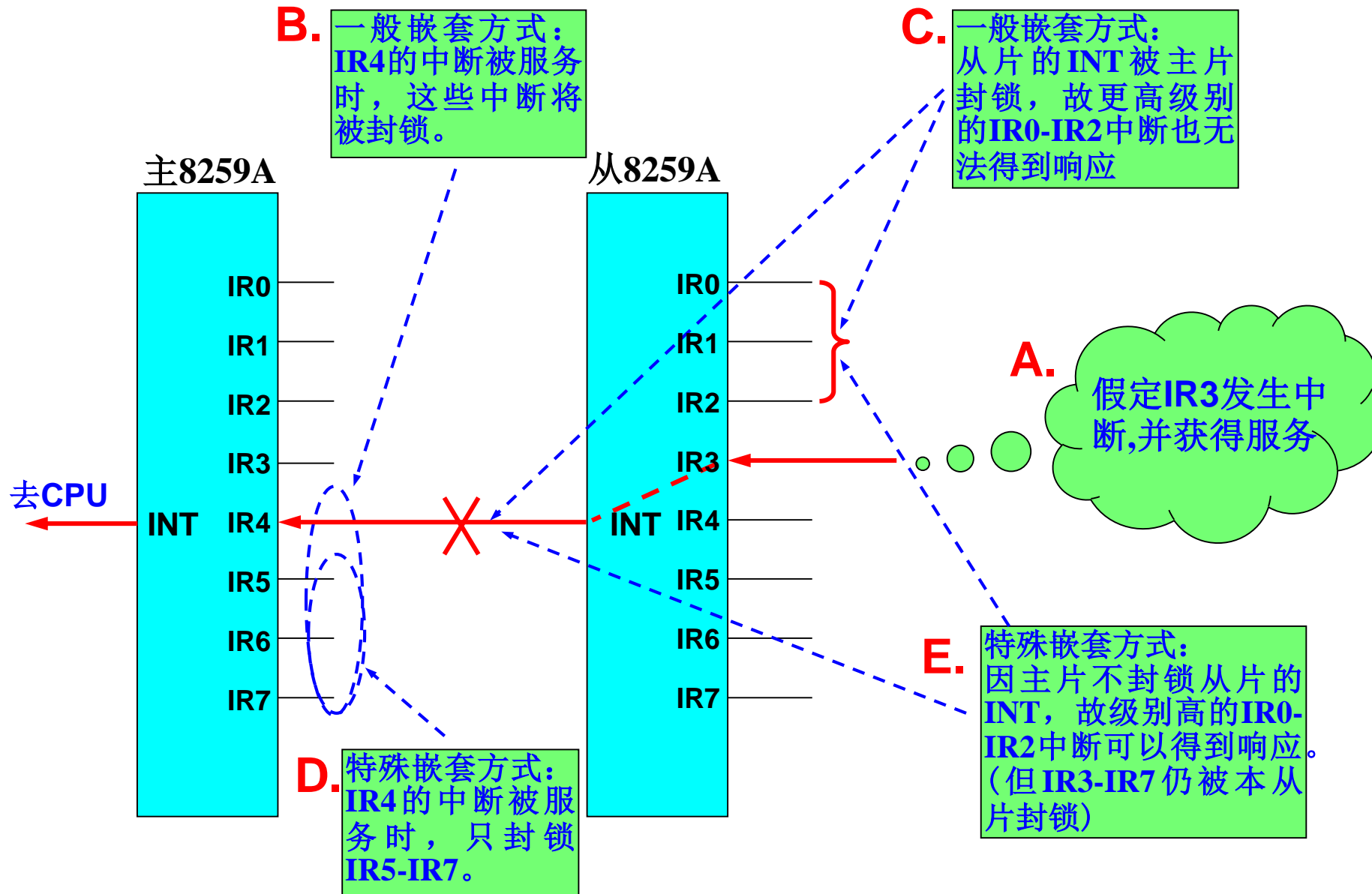


6.3 中断控制器8259A

■ (2) 特殊全嵌套方式

- **用途：**多片级联方式，特殊全嵌套方式仅设置在主片中。
- **特点：**和全嵌套方式基本相同，不同的是：在处理以及中断时，可响应同级中断，从而实现同级中断请求的特殊嵌套。

■ 一般全嵌套方式与特殊全嵌套方式的区别





6.3 中断控制器8259A

■ (3) 优先级自动循环方式:

- **用途:** 多个中断源优先级相等的场合。
- **特点:** 优先级是在变化的, 一个设备中断处理完后, 它的优先级自动降为最低。

由8259A的操作命令字OCW₂决定。



6.3 中断控制器8259A

■ (4) 优先级指定方式

用户根据要求用置优先级命令指定最低优先级，例如设置IR_i为最低优先级，则最高优先级为IR_{i+1}，其它各级按循环方式类推。

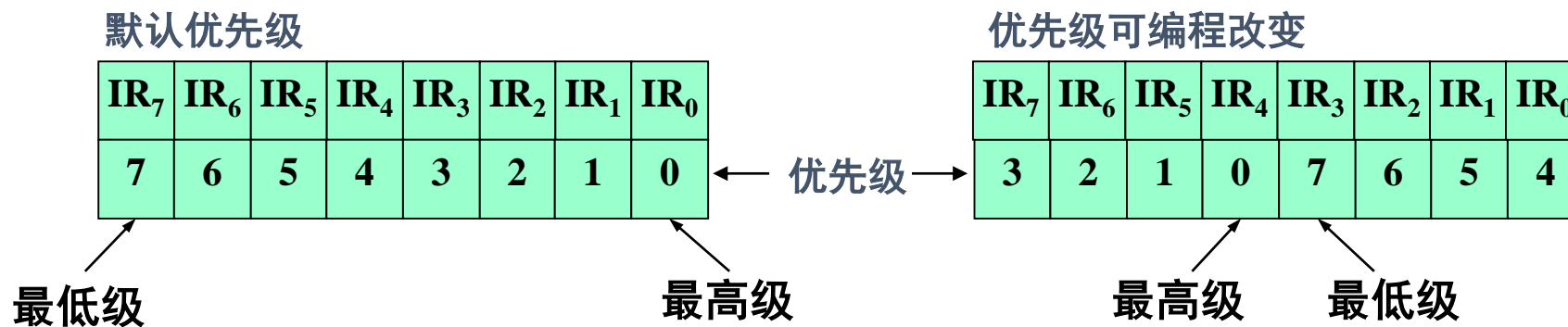


6.3 中断控制器8259A

■ 中断优先方式与中断嵌套

■ 固定优先级方式

- 所有中断请求 IR_i 的中断优先级固定不变
- 优先级排列顺序可编程改变
- 加电后8259A的默认方式，默认优先级顺序从高到低为 $IR_0 \sim IR_7$

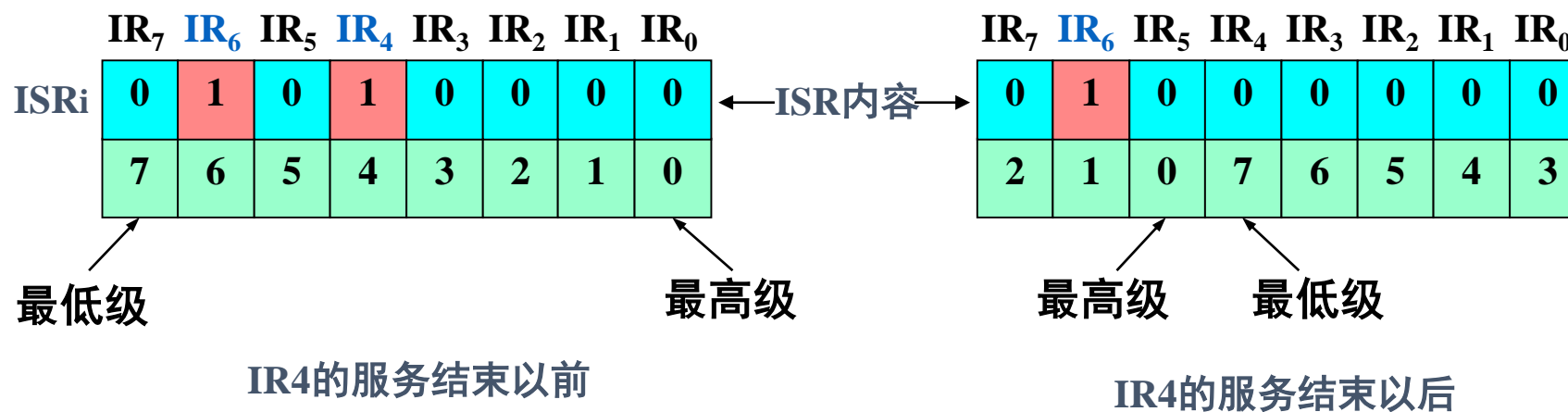


6.3 中断控制器8259A

■ 中断优先方式与中断嵌套（续）

■ 循环优先级方式

- 中断源轮流处于最高优先级，即自动中断优先级循环
- 初始优先级顺序可用编程改变
- 某中断请求 IR_i 被处理后，其优先级别自动降为最低，原来比它低一级的中断上升为最高级。





6.3 中断控制器8259A

5. 结束中断处理的方式 (EOI)

- 中断自动结束方式
- 一般的中断结束方式(重点掌握)
- 特殊的中断结束方式



6.3 中断控制器8259A

(1) 中断自动结束方式:

用于系统中**只有一片8259A**，**多个中断不会嵌套的情形**。系统一进入中断处理，就将当前**中断服务寄存器ISR**的对应位清除。对8259A来说，好像已经结束了当前中断。

在命令字ICW4中将AEOI (D1) 位置“1”。



6.3 中断控制器8259A

(2) 普通的中断结束方式(非自动EOI):

用在全嵌套的情形。CPU用OUT指令往8259A **偶地址**发一个EOI命令，8259A将使ISR最高非零IS位清0。结束当前正在处理的中断。

用操作命令字OCW2中的EOI=1，SL=0，R=0的设置来实现。

在命令字OCW2=**0010000B=20H**



6.3 中断控制器8259A

(3) 特殊的中断结束方式(非自动EOI)

- 在非全嵌套方式下，通过向8259A发一条特殊中断结束命令来判断要清除ISR中置1位中的哪一位。
- 用操作命令字OCW2中的**EOI=1, SL=1, R=0**的设置来实现，此时OCW2中的**L2、L1、L0**这3位指出了到底要清除ISR中的哪一位。

$$\text{OCW2} = 01100L_2L_1L_0$$

往偶地址端口输出OCW2命令。

OCW2中：**EOI=1、SL=1、R=0**是特殊中断结束命令。



6.3 中断控制器8259A

6.中断级联方式

- **缓冲方式：**

8259A通过总线驱动器和数据总线相连，这就是缓冲方式。

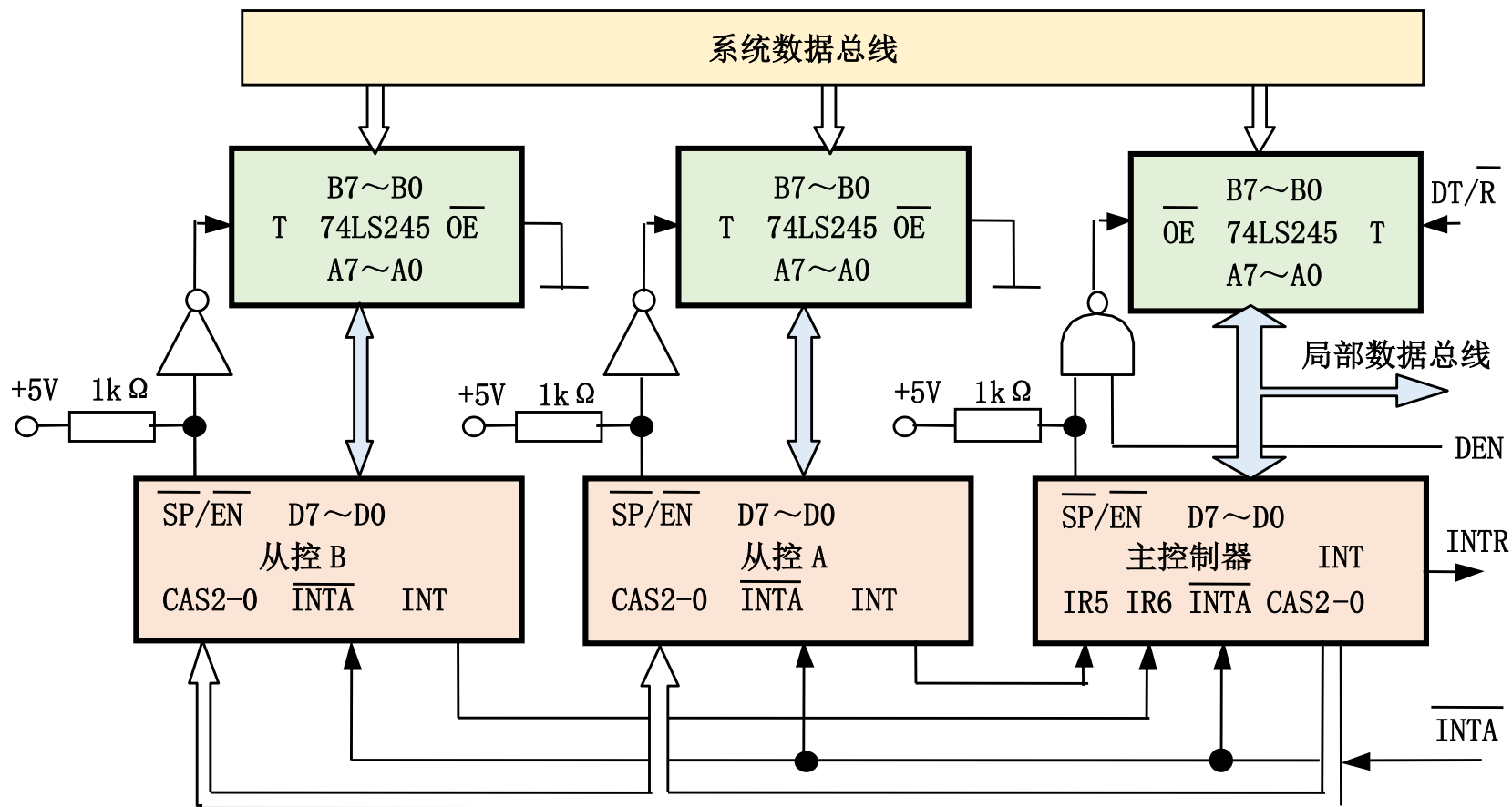
- **非缓冲方式：**

当系统中只有单片或少量8259A时，一般将它直接与数据总线相连。

6.3 中断控制器8259A

■ 中断级联方式

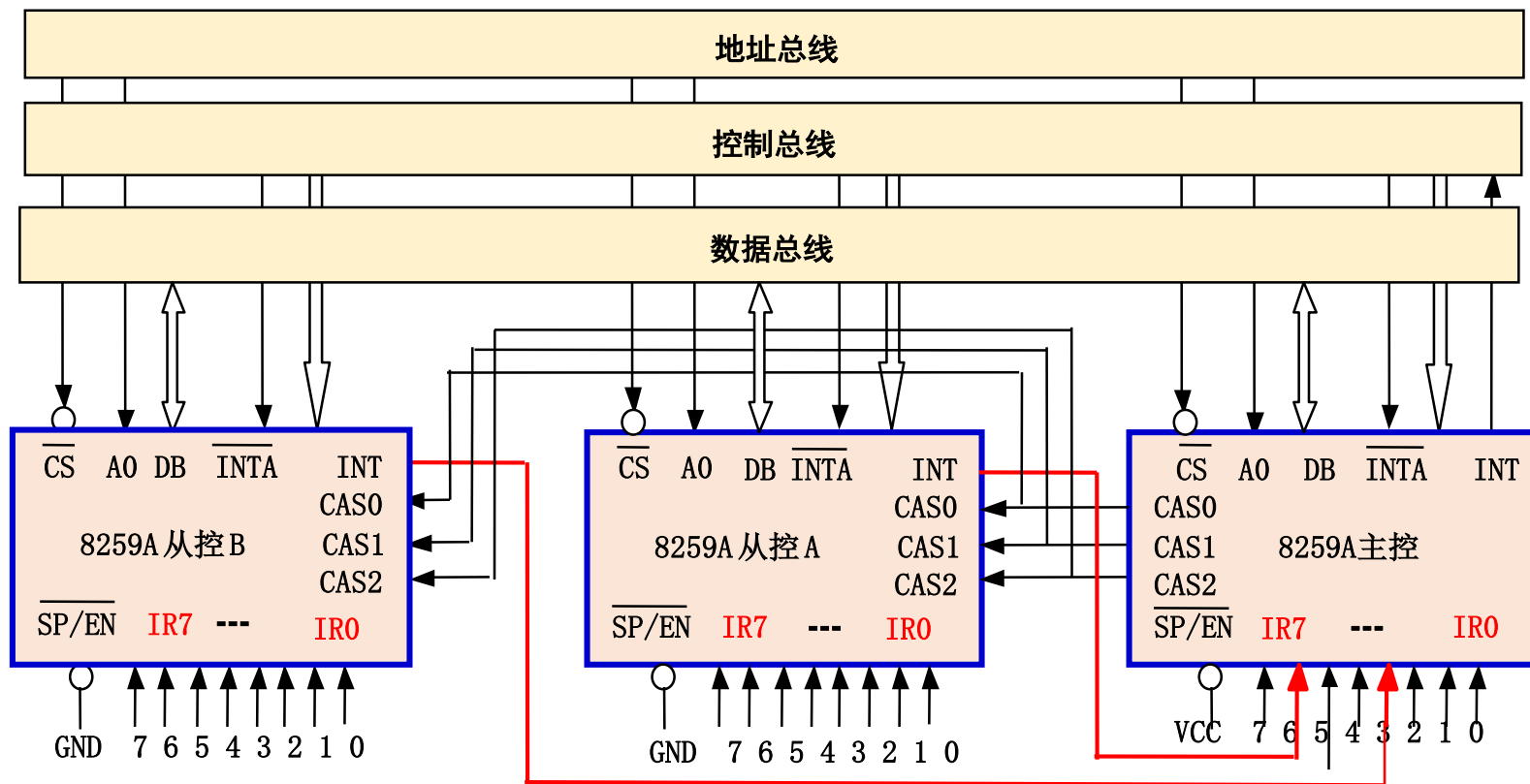
图 8259A缓冲方式下级连结构



6.3 中断控制器8259A

■ 中断级联方式

图 8259A非缓冲方式下级连结构





6.3 中断控制器8259A

6.3.4 8259A 的控制字

- **8259A的初始化命令字: ICW_i ($i=1\sim4$)**
 - **ICW1: 芯片控制初始化命令字**
 - **ICW2: 设置中断类型码**
 - **ICW3: 主片/从片的初始化命令字**
 - **ICW4: 方式控制初始化命令字**



6.3 中断控制器8259A

(1) ICW1 (A0=0) __芯片初始化命令

ICW1必须写入偶地址端口中。

D7	D6	D5	D4	D3	D2	D1	D0
0	0	0	1	LTIM	ADI	SNGL	IC4

$D_7 \sim D_5$: 在系统中不用，为1为0都可以。

D_4 : 设置为1，指示ICW₁的标志。

D_3 (LTIM) :=1，表示中断请求为电平触发方式，
=0，表示中断请求为边沿触发方式。

D_2 (ADI) : 在8086/8088中不起作用。

D_1 (SNGL) : 指出8259A有否级联。

=1，表示系统中有一片8259A，

=0，表示系统中有多片8259A。

D_0 (IC₄) : 指出后面是否设置ICW₄。使用ICW₄时IC₄必须为1。



6.3 中断控制器8259A

(2) ICW2(A0=1)__中断类型码的设置

ICW2写入8295A的奇地址端口。

D7	D6	D5	D4	D3	D2	D1	D0
T7	T6	T5	T4	T3			

例1: 8259A采用**前沿触发**，**单片**使用，需要**ICW4**，假设IR0的类型码为**08H**，试确定ICW1，ICW2。8259A端口地址：**20H**，**21H**

ICW1= 000**1** **0011**B ;**前沿触发**，**单片**，**需ICW4**

ICW2= 0000 1000B

MOV AL, 00010011B

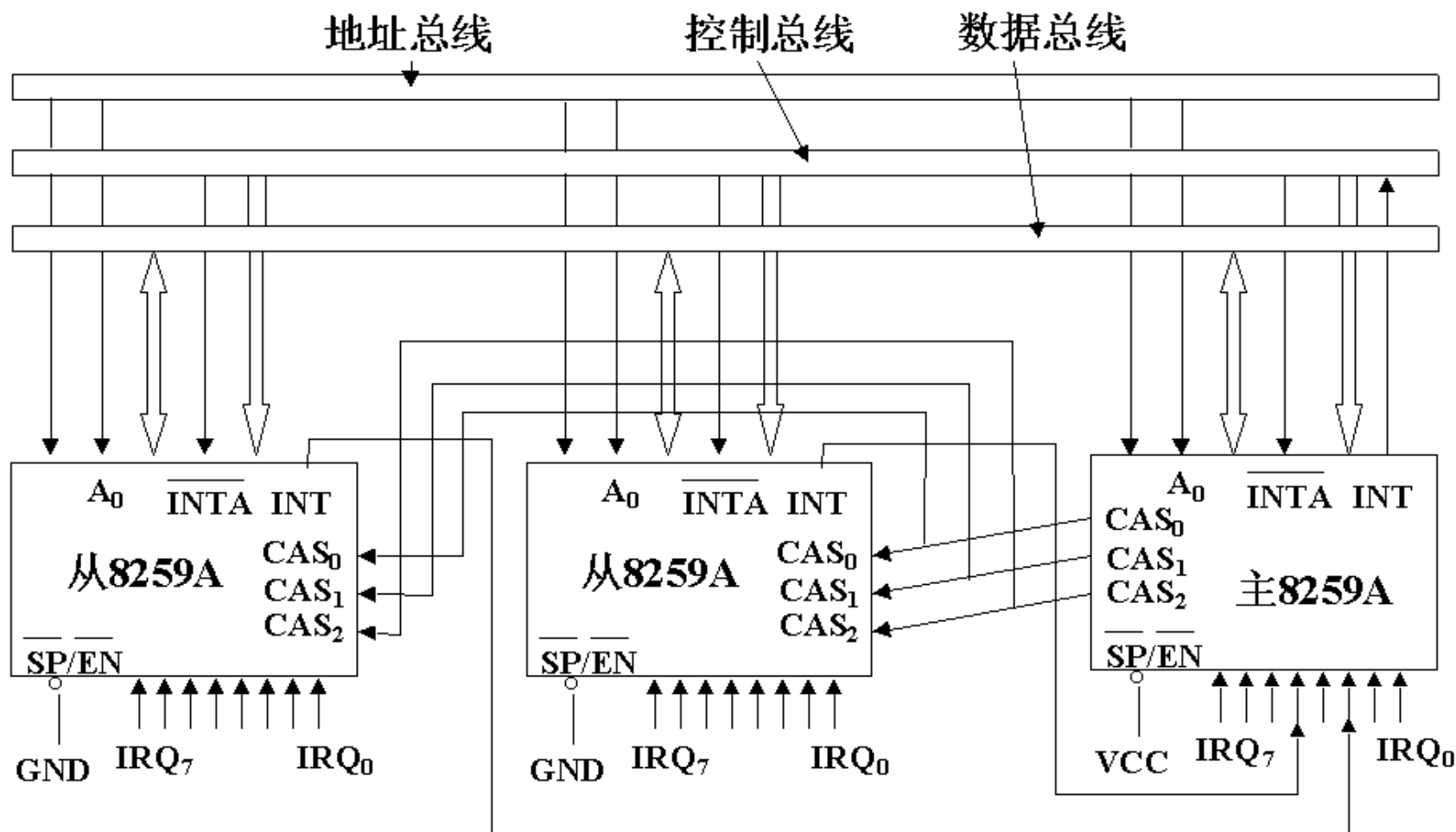
OUT 20H, AL

MOV AL, **00001000**B; 设置ICW2的**高5位**

OUT 21H, AL

6.3 中断控制器8259A

(3) 初始化命令ICW3: 级联方式结构



需要分别对主片和从片进行设置。



6.3 中断控制器8259A

(4) ICW3(A0=1)___级连方式的设置

ICW₃写入8259A的奇地址端口。

主片

D7	D6	D5	D4	D3	D2	D1	D0
IR7	IR6	IR5	IR4	IR3	IR2	IR1	IR0

从片

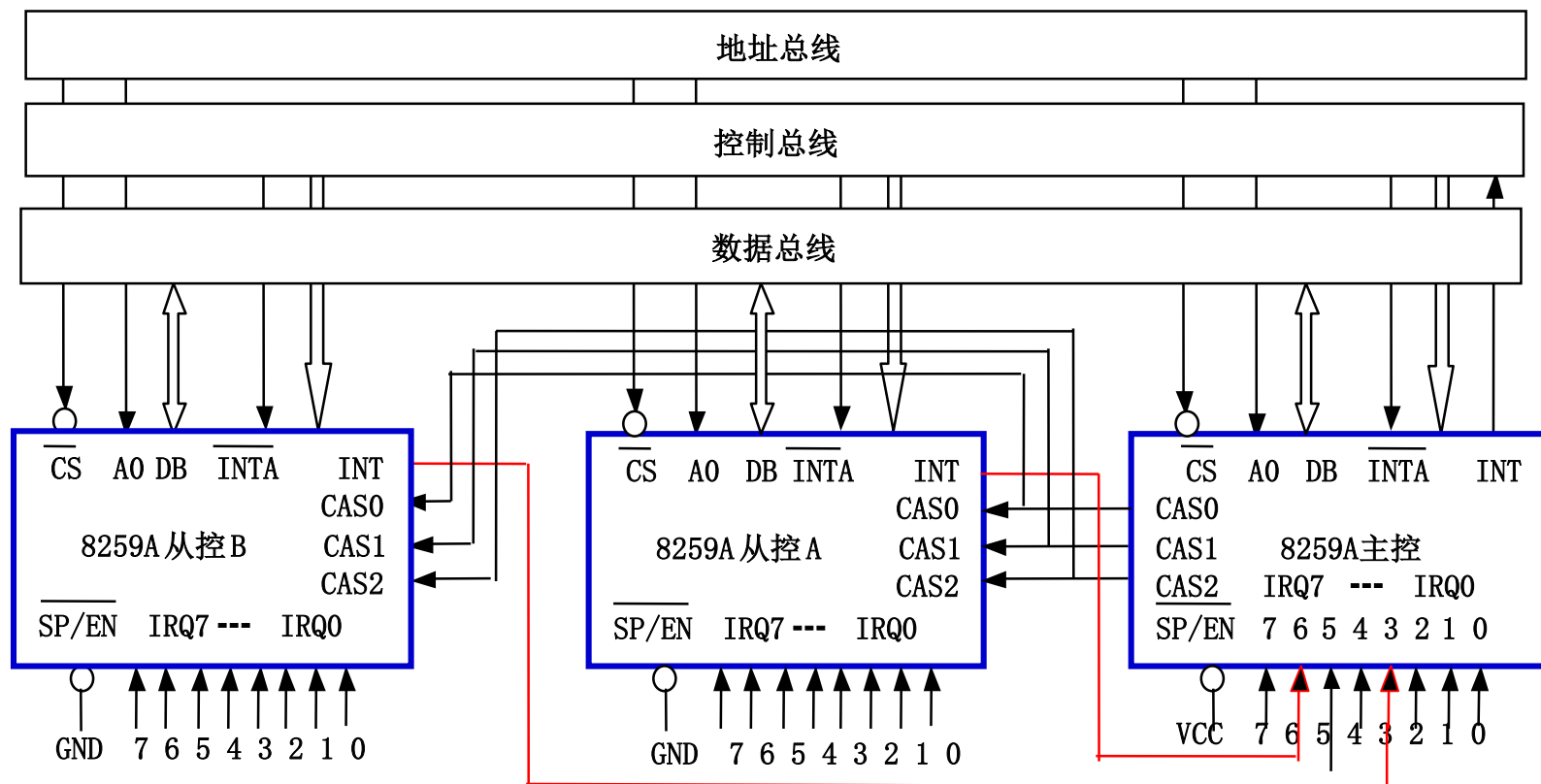
D7	D6	D5	D4	D3	D2	D1	D0
0	0	0	0	0	ID2	ID1	ID0

- **主片ICW3的格式**：指出主8259那个引脚连有从8259，对应位置一。
- **从片ICW3的格式**：指出从片8259 连在主8259那个引脚，由D2D1D0指出。

6.3 中断控制器8259A

例 8259A的级联使用

主片：SP/EN接+5V;从片：SP/EN接地。





6.3 中断控制器8259A

例: 图中主片8259A的**IR3**和**IR6**接了2个从片。主片端口地址: 20H, 21H;
1#从片端口地:0A0H, 0A1H。试设置ICW3。

主片: ICW3= 0**100** **1000**B =48H

从片1#: ICW3= 0000 **0011** =03H

从片2#: ICW3= 0000 **0110** =06H

主片ICW3的初始化程序段:

```
MOV    AL, 48H
```

```
OUT    21H, AL
```

1#从片ICW3的初始化:

```
MOV    AL, 03H
```

```
OUT    0A1H,AL
```



6.3 中断控制器8259A

1#从片挂在主片的IR6

2#从片挂在主片的IR3

- 2#从片
- IR5有申请
IRR: 000**1** 0000
ISR: 000**1** 0000
- 向主片发INT信号
1
- 2#从片与识别码一致2#从片送出中断类型码

- 主片: IR3有申请
IRR: 0000 **1**000
ISR: 0000 **1**000
- 主片向CPU发INTR信号
- CPU发回答信号INTA
- 2个从片都收到INTA
- 主片向从片送识别码:
 $CAS_0 \sim CAS_2 = 011$



6.3 中断控制器8259A

(4) ICW4的格式 (A0=1)

ICW₄为方式控制初始化命令字，**写入奇地址端口。**

- **SFNM**: =0,一般全嵌套方式;
=1,特殊全嵌套方式(Specific Full Nested Mode)
- **BUF**: =0,非缓冲方式;
=1,缓冲方式(buffered)
- **D2(M/S)**:主片(Master)/从片(Slave)
M/S=1,作主片; M/S=0,作从片
- **AEOI**: =0,中断非自动结束;
=1,中断自动结束(Automatic End Of Interrupt)
- **uPM**: =0,用于8位机; =1,用于16位机

D7	D6	D5	D4	D3	D2	D1	D0
0	0	0	SFNM	BUF	M/S	AEOI	uPM



6.3 中断控制器8259A

8259A初始化在微机系统中的使用

(1) 初始化编程流程 ICW1~ICW4

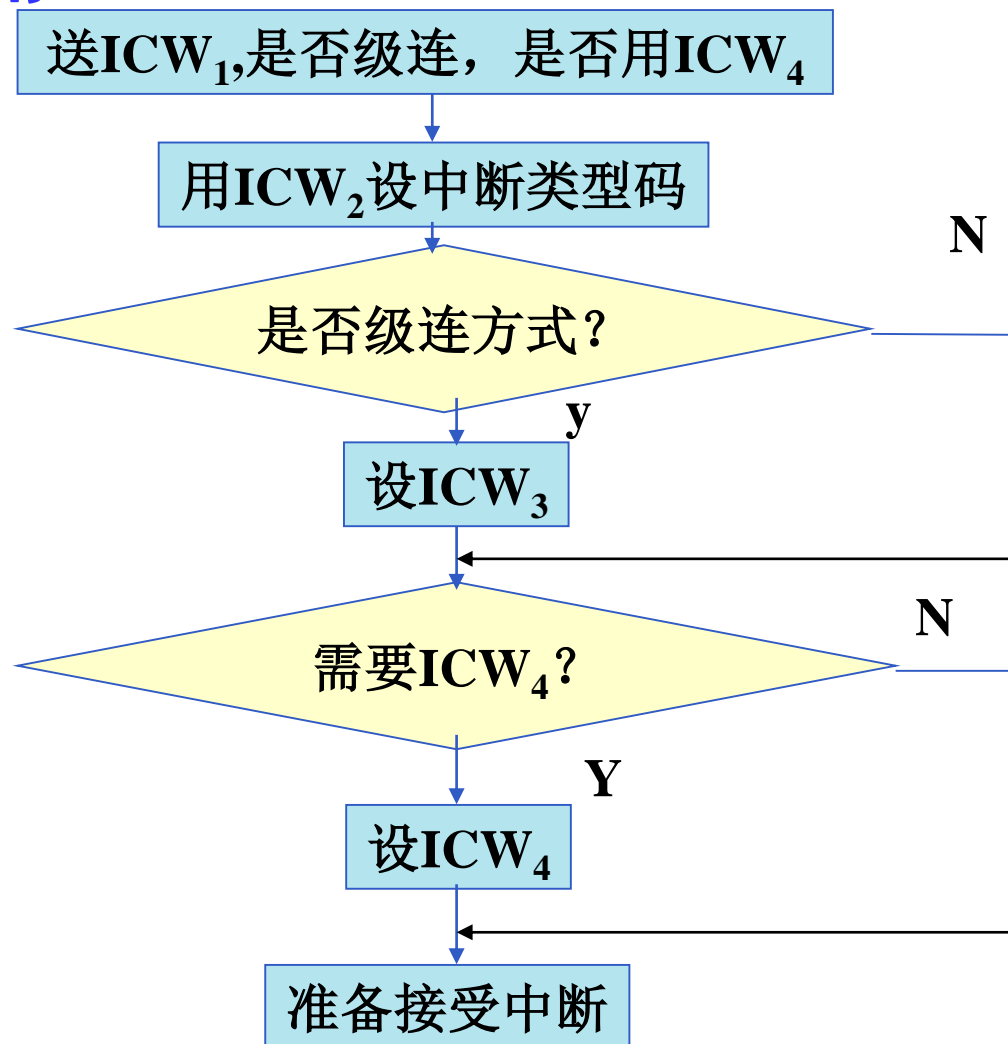


图 8259A的初始化流程



8259A 的初始化命令字

(2) 8259A的级连设置

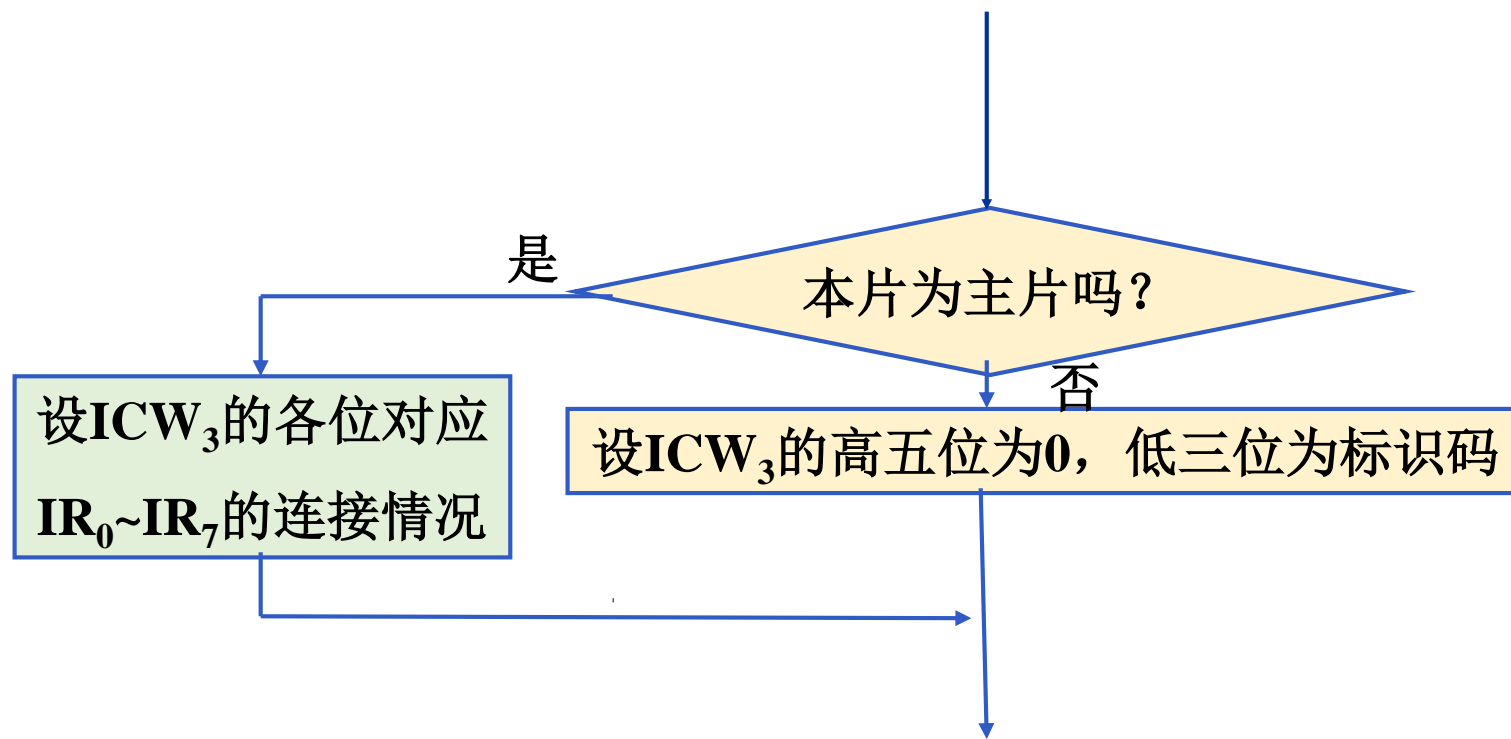


图 8259A的初始化流程



6.3 中断控制器8259A

(3) 8259A的使用

例4:单片使用8259A的初始化

(1)要求: IBM PC/XT 8259A, 单片, 边缘触发, 全嵌套方式, 非自动EOI, **设定0级中断类型码为08H。**

端口地址: 20H, 21H。

(2)硬件连接: CAS2~CAS0不用, SP/EN接+5V。

(3)初始化命令字:

ICW1=000**1** 00**1**1B ; **边缘触发,单片,需ICW4**

ICW2=00001 000B ;设置类型码的高5位

ICW4=0000 **000**1B ;**全嵌套,非缓冲,非自动EOI,16位机**



6.3 中断控制器8259A

8259A初始化编程:

....

```
MOV    AL, 13H           ;ICW1:边沿触发, 单片, 要ICW4
OUT    20H, AL
MOV    AL, 08H           ;ICW2:IRQ0中断类型为08H
OUT    21H, AL
MOV    AL, 01H           ;ICW4全嵌套, 8086系统, 普通EOI方式
OUT    21H, AL
```



6.3 中断控制器8259A

问题：

8259A仅占用两个I/O地址，它是如何区别4条ICW命令？

1) 4条ICW命令依次写入

ICW1必须写入偶地址端口 (A0=0)

ICW2必须写入奇地址端口 (A0=1)

ICW3只有在ICW1中的SNGL=0即级联时写入

ICW4只有在ICW1中IC4=1时才写入

2) 三条OCW命令次序上没有要求

但OCW1写入偶地址端口，2、3写入奇地址端口

D4、D3位为00时为OCW2，为01时为OCW3



6.3 中断控制器8259A

问题:

- 8259A的ICW2设置了中断类型码的哪几位？说明对8259A分别设置ICW2为30H、38H、36H有什么差别？

30H=00110000B

38H=00111000B

36H=00110110B

- 如何设置8259屏蔽字？
- 可以读出8259中的哪些寄存器内容？



6.3 中断控制器8259A

6.3.5 8259A的操作命令字

OCWi(i=1~3)

- 设屏蔽字
- 设优先级循环方式与中断结束 (EOI) 方式
- 读IRR、ISR
- 查询是否有：中断请求？

8259A的地址：偶地址端口 (A0=0)

奇地址端口 (A0=1)



6.3 中断控制器8259A

(1) OCW1的格式 (A0=1)

- OCW1称为**屏蔽命令字**，写入8259A的**奇地址**端口。
- 当OCW1某位=1，表示对应的中断请求被屏蔽。

例如：屏蔽IR3与IR4，则OCW1=000**11**000

```
MOV    AL, 00011000B
```

```
OUT    21H,AL ;OCW1送8259奇地址端口
```



6.3 中断控制器8259A

(2) OCW2的格式 (A0=0)

- 用于设置**优先级循环方式**和**中断结束方式**，写入**偶地址**单元。

D7	D6	D5	D4	D3	D2	D1	D0
R	SL	EOI	0	0	L2	L1	L0

D₄D₃: =00, **特征位**

D7(R): =1, 优先级循环方式; R=0, 非循环方式。

D5(EOI): =1, 在中断服务完毕, 发中断结束命令, 使**当前ISR的对应位IS_n复位**。

D5(EOI): =0, 不发EOI, 自动清除对应IS_n。

D6(SL): =1, 用L2、L1、L0指定中断等级。

在ICW4中, 若AEOI=1, 自动清除当前IS_n。

AEOI=0, IS_n要用EOI清除。



6.3 中断控制器8259A

■ 优先级排队方式

- | | |
|---------------------|--|
| $D_7 D_6 D_5 = 000$ | 自动EOI方式，第二个INTA后沿自动将ISR中相应位清0 |
| $D_7 D_6 D_5 = 001$ | 正常EOI方式，将ISR中优先级最高的置1位清0 |
| $D_7 D_6 D_5 = 010$ | 无效 |
| $D_7 D_6 D_5 = 011$ | 特殊EOI方式，按 $D_2 \sim D_0$ 编码级别将ISR中相应位清0 |
| $D_7 D_6 D_5 = 100$ | 自动EOI循环方式，第二个INTA清ISR相应位，并赋予最低优先级，循环优先级 |
| $D_7 D_6 D_5 = 101$ | 正常EOI循环方式，将ISR中优先级最高的置1位清0，并赋予最低优先级，循环优先级 |
| $D_7 D_6 D_5 = 110$ | 优先级置位，将最低优先级赋予 $D_2 \sim D_0$ 编码的中断源 |
| $D_7 D_6 D_5 = 111$ | 特定EOI循环方式，按 $D_2 \sim D_0$ 编码级别清ISR，并赋予最低优先级，循环优先级 |



6.3 中断控制器8259A

(3) OCW3的格式 (A0=0)

- 用于提供查询方式，设置与撤销特殊屏蔽方式，读8259寄存器等操作。

D7	D6	D5	D4	D3	D2	D1	D0
×	ESMM	SMM	0	1	P	RR	RIS

$D_4D_3=01$ ，特征位

设置P=1时，查询是否有中断请求正在处理。

RR=1，RIS=0，读IRR

RR=1，RIS=1，读ISR

ESMM=1，SMM=1，置特殊屏蔽方式

ESMM=1，SMM=0，撤销特殊屏蔽方式



6.3 中断控制器8259A

怎样用OCW3设置中断查询命令？

查询步骤：

(1)用OCW3构成查询命令：OCW3=000**01100**

(2)从偶地址端口读查询字，得到**最高优先级**的中断

例5：假设IR4，IR6引脚上有中断请求，查询程序片断：

CLI

MOV AL, 00001100B ;设置OCW3

OUT 20H, AL ;输出到8259偶地址端口

IN AL, 20H ;读查询字

D7	D6	D5	D4	D3	D2	D1	D0
I	---	---	---	---	W2	W1	W0
1	---	---	---	---	1	0	0



6.3 中断控制器8259A

怎样读寄存器IRR,ISR?

- 使OCW3的 $D_4D_3=01$ ，特征位， $D_2(P)=0$ ，
- OCW3=000**010**10B，RR=1，RIS=0，读IRR；
- OCW3=000**010**11B，RR=1，RIS=1，读ISR。

程序片断：

```
MOV  AL, 00001010B    ;设置OCW3
```

```
OUT  20H,AL
```

```
IN   20H, AL           ;从偶地址读IRR
```

```
MOV  AL, 00001011B    ;设置OCW3
```

```
OUT  20H,AL
```

```
IN   20H, AL           ;偶地址读ISR
```



6.3 中断控制器8259A

怎样读寄存器IMR?

- 若要读IMR，不须设OCW3的查询命令，用奇地址端口读IMR。
- `IN AL, 21H` ;从奇地址端口读IMR



6.3 中断控制器8259A

特殊屏蔽怎样操作？

- **ESMM=1,SMM=1**,置特殊屏蔽方式，8259A脱离当前优先级方式，如**IF=1**，系统可以响应任何未被屏蔽的中断申请，好像优先级规则不起作用。
- **ESMM=1, SMM=0**，撤销特殊屏蔽方式

程序片断：

```
MOV AL, 01101000B    ;设置OCW3
OUT 20H, AL          ;送特殊屏蔽字到8259
```



6.3 中断控制器8259A

小结： 8259A那些寄存器内容可读出？

设8259A地址20H, 21H。

1) 读中断屏蔽寄存器IMR内容(任何时候可以读)

IN AL, 21H ; 读奇地址端口, $A_0=1$

2) 读IRR: 设置OCW3: $RR=1, RIS=0$

MOV AL, 00001010B

OUT 20H, AL

IN AL, 20H ; 读偶地址端口

3) 读ISR: 设置OCW3: $RR=1, RIS=1$

MOV AL, 00001011B

OUT 20H, AL

IN AL, 20H ; 读偶地址端口



6.4 8259A应用举例

例：假设1片8259A工作于全嵌套方式。若要求在IR3的中断服务中，允许IR5进入，该如何作？

1) 在IR3中断服务程序开始

- 关中断(CLI),屏蔽IR3
- 用OCW3设置特殊屏蔽方式
- 取原屏蔽字保存,
- 将IMR4,IMR6,IMR7置1,只允许IR5进入
- 开中断(STI)

2) 在IR3中断服务程序

- 用OCW3撤销特殊屏蔽方式
- 适当处恢复原屏蔽字



6.4 8259A应用举例

例 某系统中用两片8259A级联组成中断系统。

- 1) 8259A主片的IR1、IR5上引入两个中断源，其中断类型码为31H、35H，中断服务程序的入口地址为1000：2000H和1000：3000H。
- 2) 从片接在主片的IR3上，从片的IR4、IR5上引入两个中断源，其中断类型码为44H、45H，中断服务程序的入口地址为2000：3600H和2000：4500H。
- 3) 8259A主片口地址为C8H/C9H，从片口地址为CAH/ CBH。
- 4) 边沿触发、非缓冲

要求画出硬件连线图，并编写 8259A的主片和从片的初始化程序。

6.4 8259A应用举例

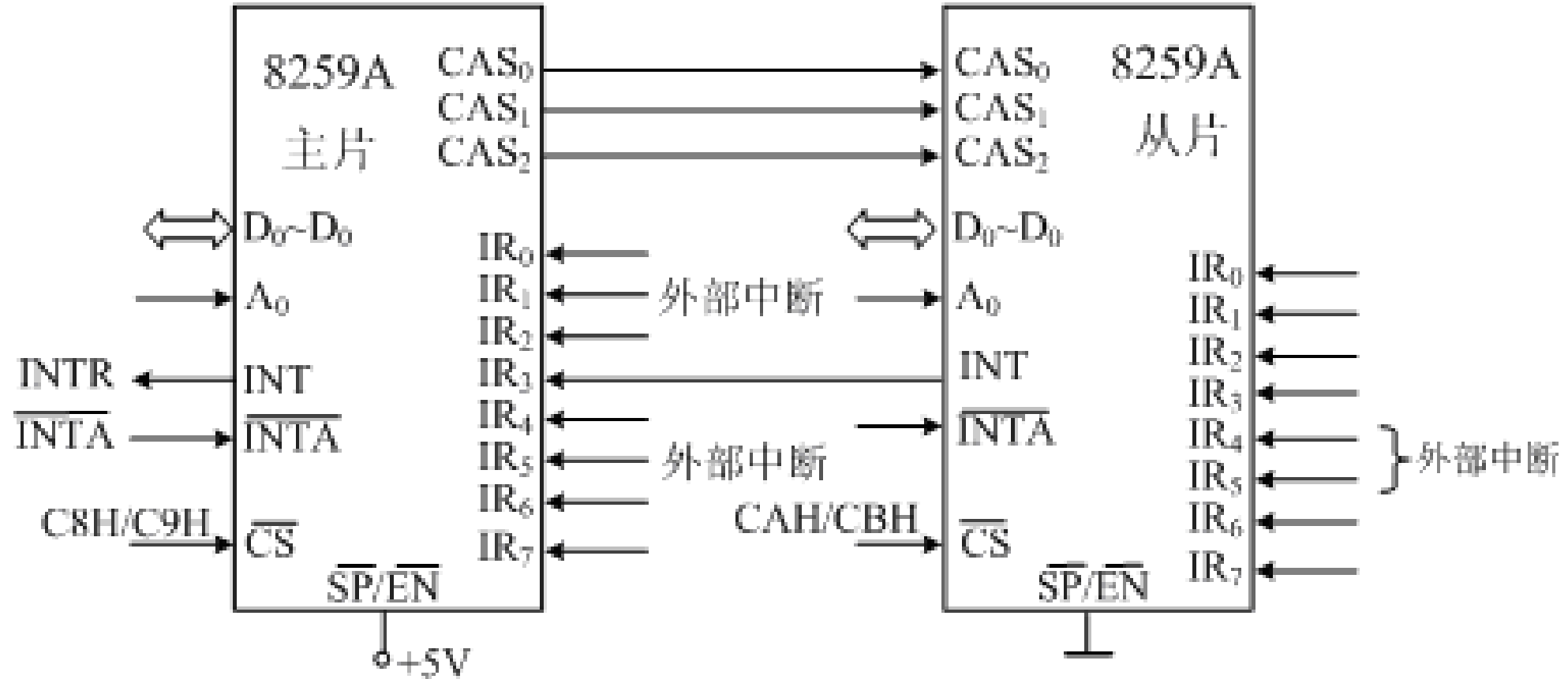


图 级联电路硬件连线图

4个中断服务程序的入口地址表如图所示。

地址	中断向量			
000C4	00	IP	31H号中断 入口地址	8259A主片引入 的中断请求
	20			
000C6	00	CS		
	10			
000D4	00	IP	35H号中断 入口地址	
	30			
000D6	00	CS		
	10			
	⋮			
00110	00	IP	44H号中断 入口地址	8259A从片引入 的中断请求
	36			
00112	00	CS		
	20			
00114	00	IP	45H号中断 入口地址	
	45			
00116	00	CS		
	20			

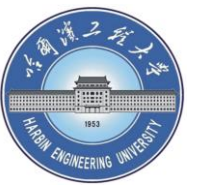
图 8.22 中断服务程序入口地址表

31H号中断的中断服务程序的入口地址存放在 $31H \times 4 = 0C4H$ 开始的4个连续单元中，它等于1000:2000H。

35H号中断向量位于0D4H开始的单元中，中断服务程序的入口地址为1000:3000H中。

其余中断也可用类似方法求得。

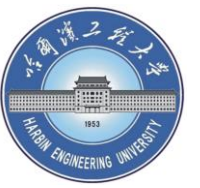
6.4 8259A应用举例



对8259A从片初始化程序:

MOV	AL, 11H	; ICW ₁ , 同主片
OUT	0CAH, AL	
MOV	AL, 40H	; ICW ₂ , 中断类型码n=40H~47H
OUT	0CBH, AL	
MOV	AL, 0000011B	; ICW ₃ , 从片接在主片的IR ₃ 上
OUT	0CBH, AL	
MOV	AL, 0000001B	; ICW ₄ , 全嵌套, 非AEOI
		; 方式结束中断
OUT	0CBH, AL	
MOV	AL, 1100111B	; OCW ₁ , 允许从IR ₅ 、IR ₄ 引
		; 入中断, 其余屏蔽
OUT	0CBH, AL	

第6章 小结



- 中断的概念
- 8086中断系统：NMI/INTR
- 中断向量，中断向量表，中断服务程序框架
- 8259A中断控制器特点：
 - 管理中断能力
 - 开放与屏蔽中断
 - ISR，IRR，IMR作用
 - 中断向量的设置



**本章结束，
谢 谢！**