

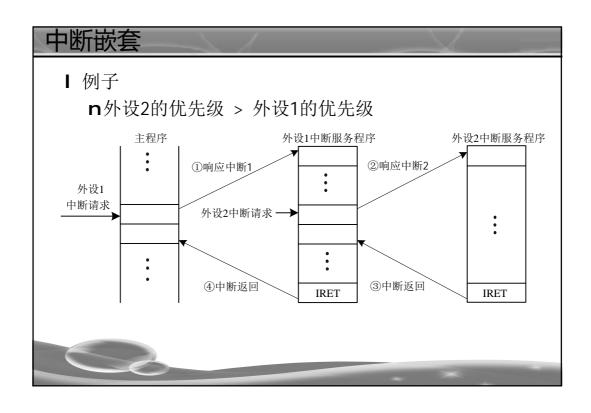


0 0 0 0 0	
00000	
0 0 0 0 0	
$ \bigcirc \bigcirc \bigcirc \bigcirc \bigcirc \bigcirc \bigcirc \bigcirc $	
0 0 0 0 0	
-0-0-0-0-	<i>Κ</i> / <i>α</i> ++ + - - - - - - - - - -
0 0 0 0 0	第1节 中断的概念
0 0 0 0 0	
00000	
0 0 0 0 0	
00000	
0 0 0 0 0	
00000	
0 0 0 0 0	

- Ⅰ 中断的概念
 - nCPU的机制或过程
 - n CPU在正常运行程序时,由于内部/外部事件事件引起CPU中断正在运行的程序,转去执行为该事件预先安排的服务程序,服务完毕后,再返回原来的程序继续执行。
- Ⅰ 中断的作用和应用
 - n①同步操作/并行操作
 - n②实时处理
 - n③故障或异常处理

- Ⅰ 中断源和其分类
 - n中断源:引起中断的事件或发出中断的外设。
 - n分类
 - u外部中断
 - p可屏蔽中断 (INTR引脚)
 - ▲外设
 - p不可屏蔽中断(NMI引脚)
 - ▲电源,内存错误等
 - u内部中断
 - pCPU内部
 - **p**软件中断(指令中断)
 - **p**异常(指令执行时发生错误)

- Ⅰ 中断系统的功能
 - n 能发现和识别中断源,实现中断服务。
 - n①实现中断和从中断返回
 - u外设能发出中断请求;
 - uCPU能决定是否响应中断
 - u中断响应过程
 - **p**保护现场-清除中断请求-中断服务-恢复现场-返回主程序
 - n②实现优先权排队
 - u当有多个中断申请时,先响应优先级高的中断
 - n ③能实现中断嵌套。
 - u即高优先级中断能中断低优先级的中断。



CPU响应中断的条件

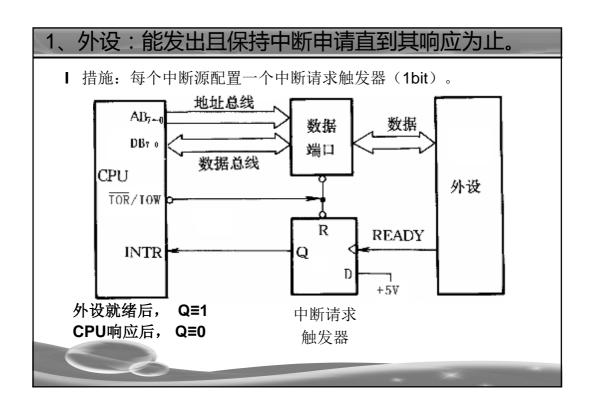
- Ⅰ ①必须执行完现行指令
 - n运行到最后一个总线周期的最后一个时钟状态(T4),CPU 检测INTR引脚是否有中断请求,有则响应中断,进入中断总 线周期。
- Ⅰ ②开中断状态
 - nSTI指令
- I ③无DMA操作

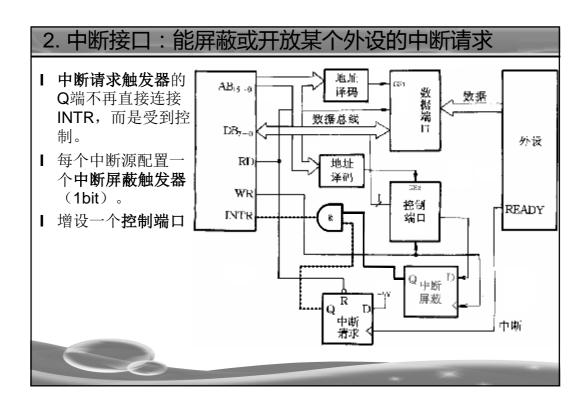
中断响应过程

- Ⅰ ①关中断——发出响应信号INTA,并自动关中断。
- Ⅰ ②保留断点——主程序下一条指令地址(CS:IP)入栈。
- Ⅰ ③保护现场——断点处各寄存器的内容入栈。
- Ⅰ ④根据中断入口地址,转入相应中断服务程序。
- Ⅰ ⑤恢复现场——把现场和断点的内容出栈。
- Ⅰ ⑥中断返回——在中断服务程序的最后执行IRET,并开中断。
- Ⅰ 注: ①②④由CPU自动完成;
 - ③⑤⑥由程序完成。

微机中断的基本硬件机制

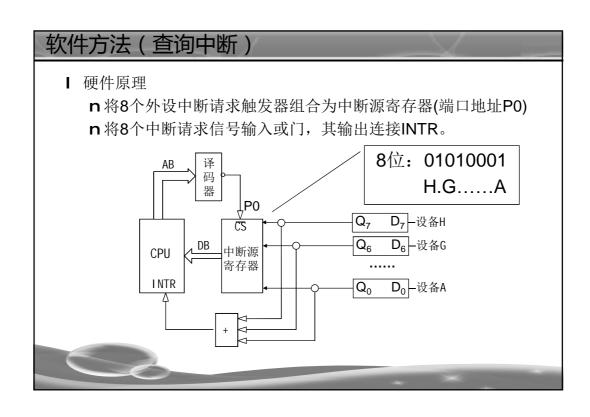
- Ⅰ 1、外设: 能发出且保持中断申请直到其响应为止。
- Ⅰ 2、中断接口: 能屏蔽或开放某个外设的中断请求。
- 1 3、CPU: 能开放(使能)或关闭(失能)系统的中断机制。
- Ⅰ 4、能识别中断源和管理中断优先级
- Ⅰ 5、确定发现中断的时机(T4态)

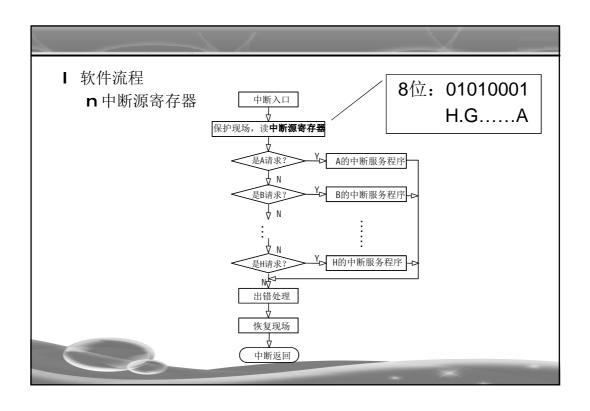


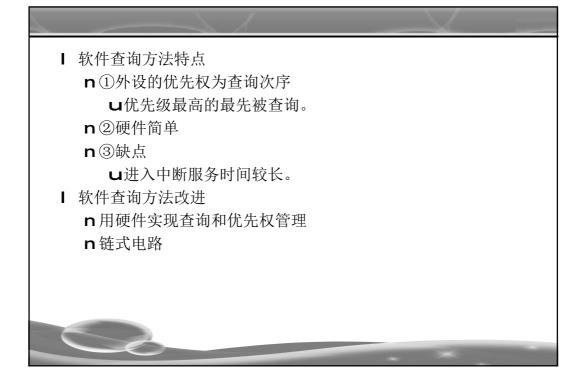


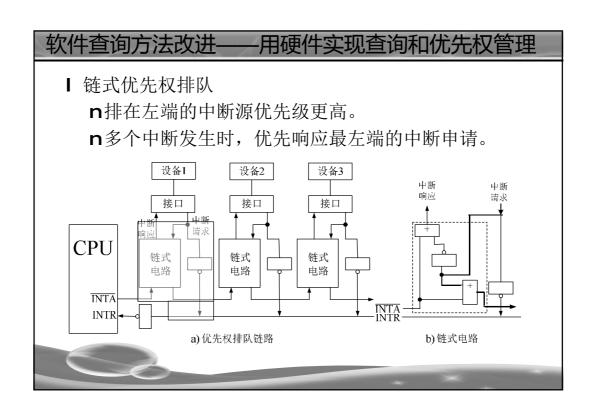
4、能识别中断源和管理中断优先级

- Ⅰ 识别中断源和中断优先级
 - n 所谓优先级,是指有多个中断源同时提出中断请求时,微处 理器响应中断的优先次序。
- Ⅰ 实现方法
 - n 软件方法(查询中断)
 - n 硬件方法(向量中断)





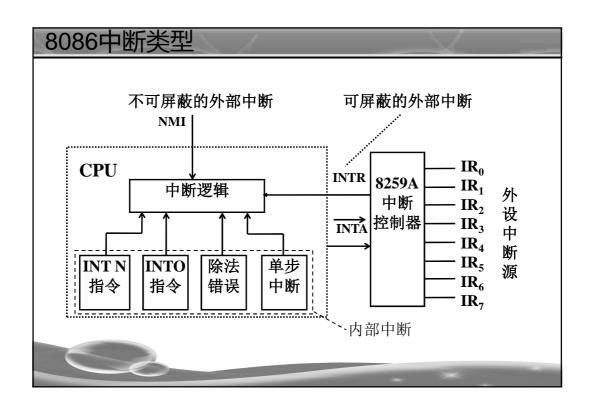




- Ⅰ 向量中断——用硬件方法确定中断源及优先权
 - n 给每一个外设都预先指定一个中断类型码(N)。当CPU识别出外设有请求中断并予以响应时,中断控制逻辑把N送入CPU,CPU据此计算中断服务程序的入口地址,并转入中断服务程序。
 - n特点:
 - u响应速度快
 - u硬件复杂
 - **u**典型芯片: 8259

- Ⅰ 中断向量
 - n中断服务程序的入口地址 CS:IP
 - u段基址CS: 偏移地址IP
 - u每个向量占4字节。
- Ⅰ 中断向量表
 - n中断向量表是按**序(中断源编号N)**存放中断向量的表。
 - n中断向量表存放在内存中(基地址)。
 - n查找N号中断源的中断向量的存放地址
 - u = 4 * N +中断向量表的基地址
 - n中断向量表是中断类型号(N)与中断向量之间的换算表。
 - n一般中断向量表从内存的0地址放起

00000	
0 0 0 0 0	
0 0 0 0	
\bigcirc	
0 0 0 0 0	
 	<u>₹</u> \$0++ 000/65+0₩€+0#U
00000	第2节 8086的中断机制
0 0 0 0 0	
00000	
0 0 0 0 0	
00000	
0 0 0 0 0	
00000	
0 0 0 0 0	



8086的中断类型

- 外部中断 (Interrupt)
 - n (1) 可屏蔽中断:来自外设,受IF控制, N=8~255
 - n (2) 非屏蔽中断:处理系统意外或故障,不受IF控制, N=2

陷阱标志位TF=1时,每条

指令执行完引起中断。N=1

UNMI = Non Maskable Interrupt

- Ⅰ 内部中断(异常, Exception): 8088内部执行程序错误
 - n (1) DIV或IDIV指令

 - n (2) INT指令
 - n (3) INTO指令
 - n (4) 单步中断
 - n为用户提供发现、调试并解决程序执行异常的途径
 - **山**例如BIOS和DOS功能调用,DEBUG

部分中断的详细说明

I DIV或IDIV (无/有符号数除法)指令

n除数为零或商溢出

nN = 0

MOV BL, 0

IDIV BL ;除数BL=0,产生除法错中断

MOV AX, 200H

MOV BL, 1

DIV BL ; 商 200H, 溢出(不能用AL表达),产生中断

▮ 溢出中断

n 若溢出标志位OF=1,则INTO指令引起中断

nN = 4

MOV AX, 2000H ADD AX, 7000H

; 2000H + 7000H=9000H, 溢出: OF = 1

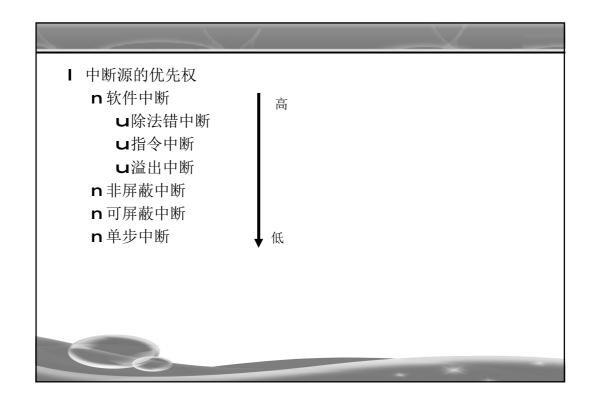
INTO ; 因为OF = 1,产生溢出中断

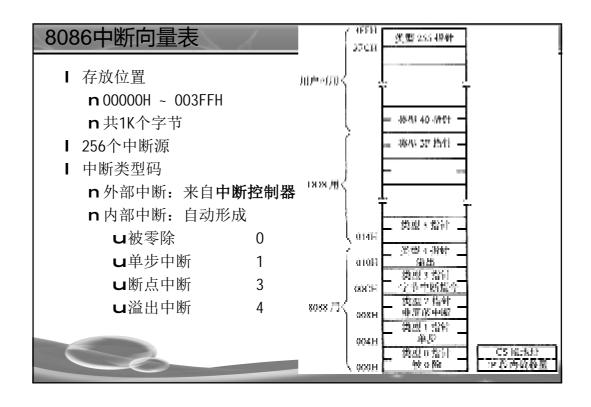
Ⅰ 软中断

n INT n指令产生一个中断

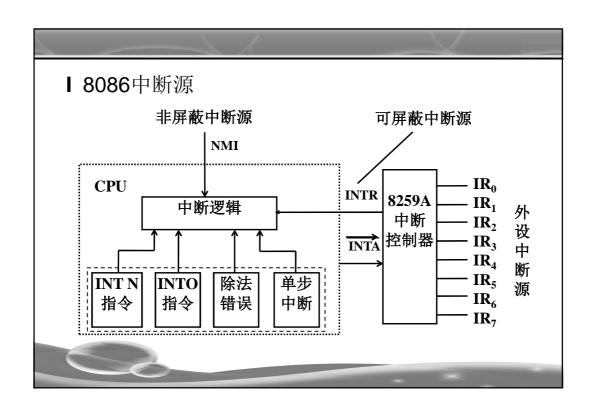
nN = n

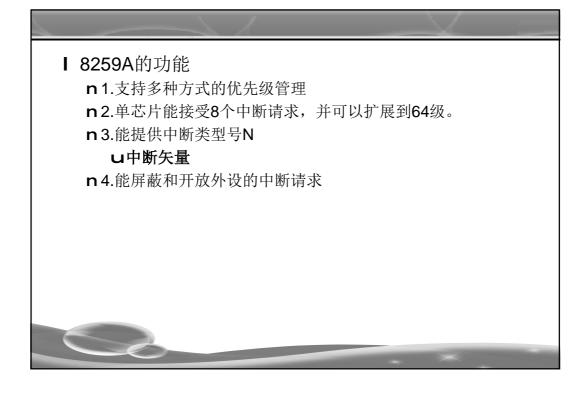
n例子: n = 21H, DOS系统功能调用

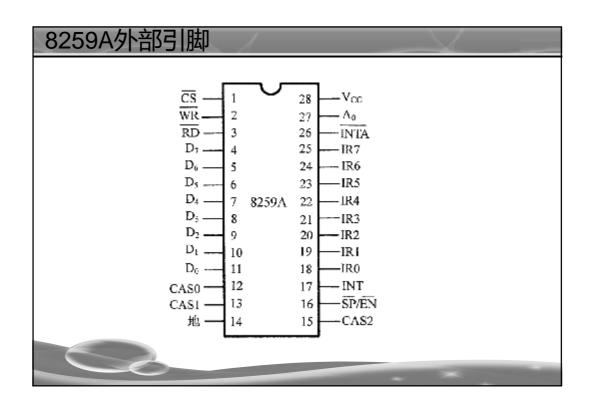




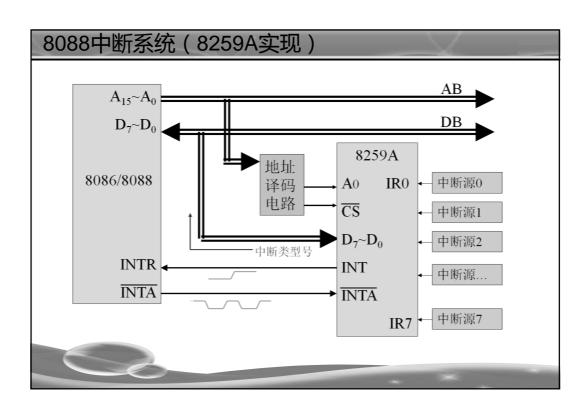
0000		
0 0 0 0 0		
00000		
0 0 0 0 0		
0000		
0 0 0 0 0		
0-0-0-0-	<u>たたっ</u> 井 0050 M M A A A A A A A A A A A A A A A A A	
0 0 0 0 0	第3节 8259A性能和结构	
00000		
00000		
0 0 0 0 0		
00000		
00000		
00000		
0 0 0 0 0		
$\boxed{\bigcirc \bigcirc \bigcirc \bigcirc \bigcirc \bigcirc \bigcirc \bigcirc}$		

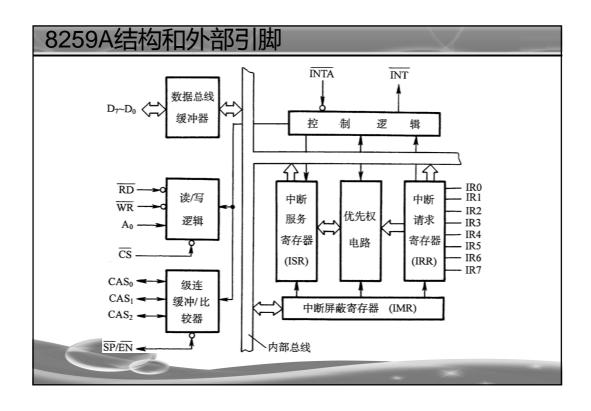






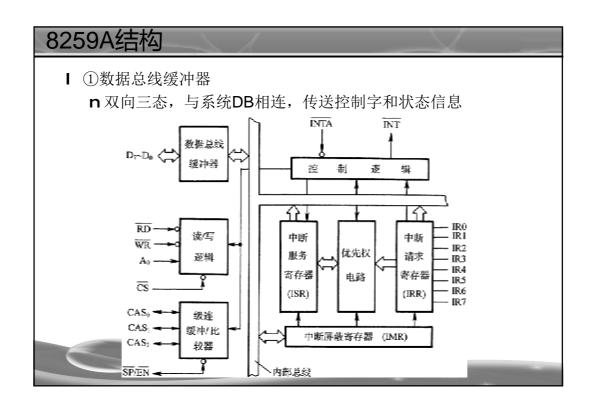


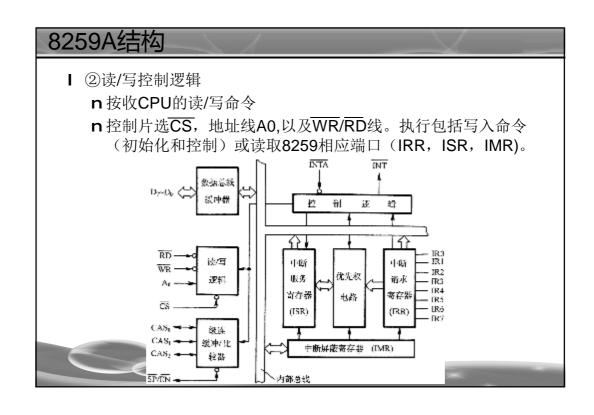


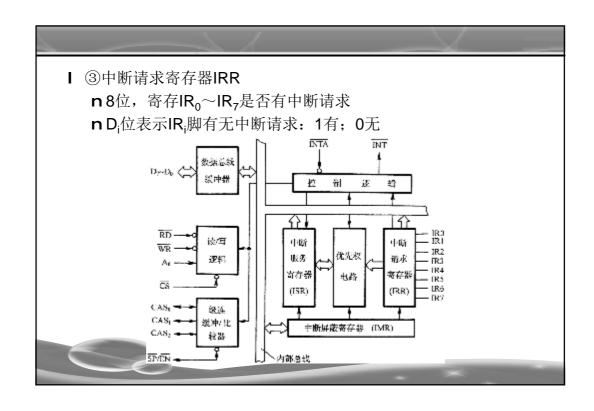


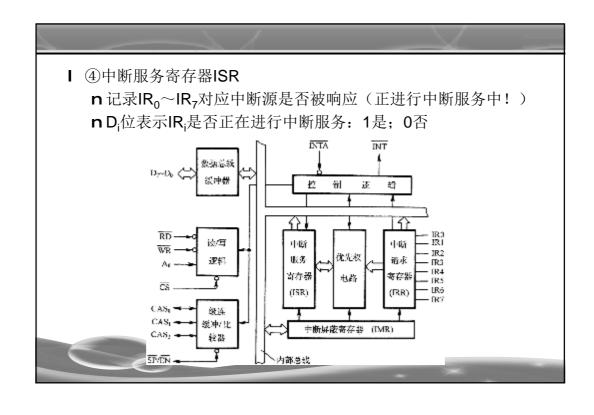
8259A结构

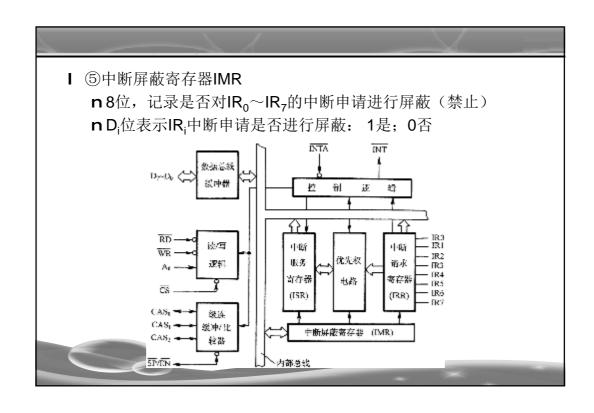
- Ⅰ 1.数据总线缓冲器
- I 2.读/写逻辑
- Ⅰ 3.中断请求寄存器IRR(Interrupt Request Register)
- Ⅰ 4.中断服务寄存器ISR (Interrupt Service Register)
- Ⅰ 5.中断屏蔽寄存器IMR (Interrupt Mask Register)
- Ⅰ 6.控制逻辑
- Ⅰ 7.级联缓冲/比较器
- Ⅰ 8.优先权电路

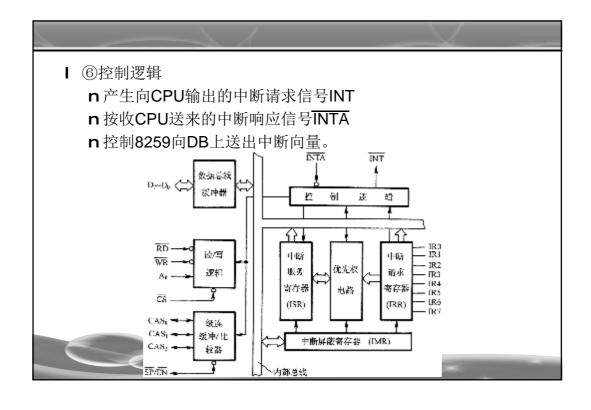


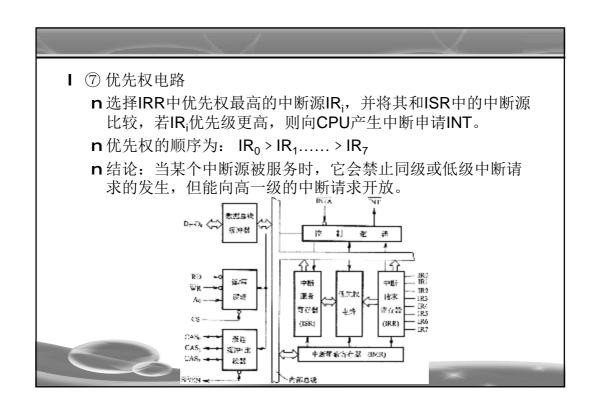


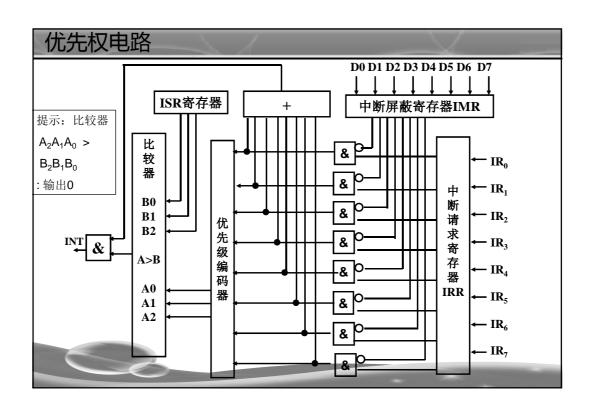






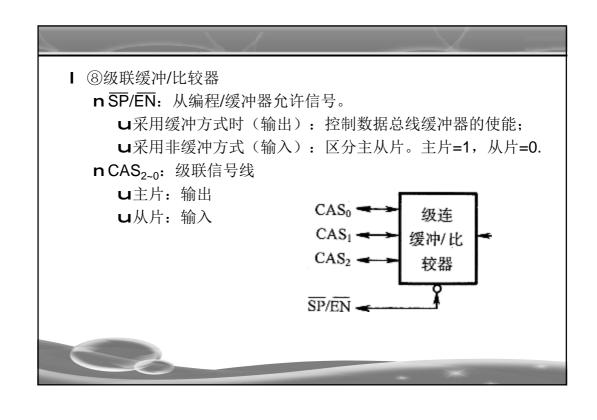


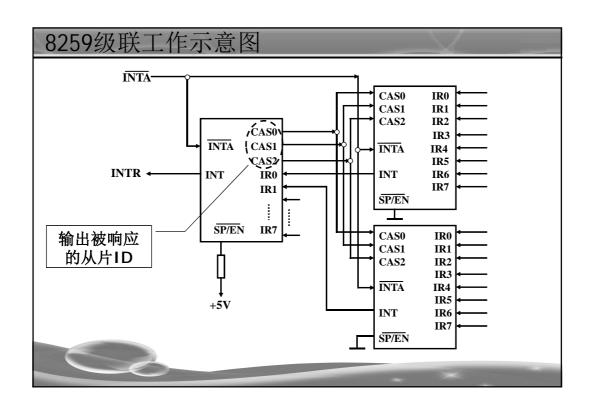




优先权电路

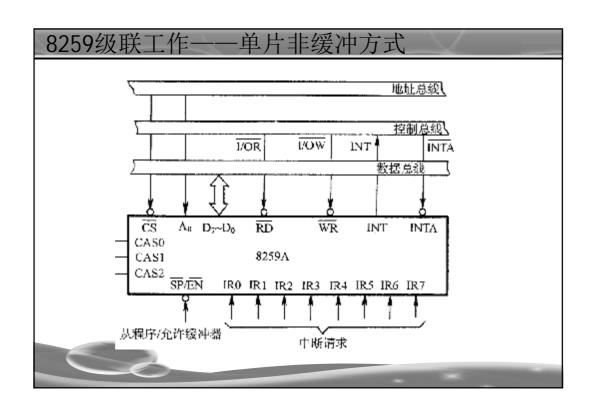
- Ⅰ 工作原理
 - n 首先,由8个"与"门逻辑选出参加中断优先级排队的中断请求级:由8位IRR与8位IMR分别送入"与"门输入端,只有当IRR位置"1"(有中断请求)且IMR位置"0"(开放中断请求)同时成立时,相应"与"门输出才为高电平,并送到优先级编码器的输入端参加编码。
 - **n** 其次,优先级编码器对参加排队的中断优先级进行编码,并 从中选出当前最高优先级的代码(A2,A1,A0)。
 - n最后,把ISR中当前正服务的优先级的代码(B2,B1,B0)与新来的中断请求的优先级代码(A2,A1,A0)一起送入比较器进行比较:当比较器输出有效且有中断请求时,与门将输出有效电平向CPU提出中断请求INT。
 - **n**结论: 当一个中断源被服务时,它会禁止同级或低级中断请求的发生,但能向高一级的中断请求开放。

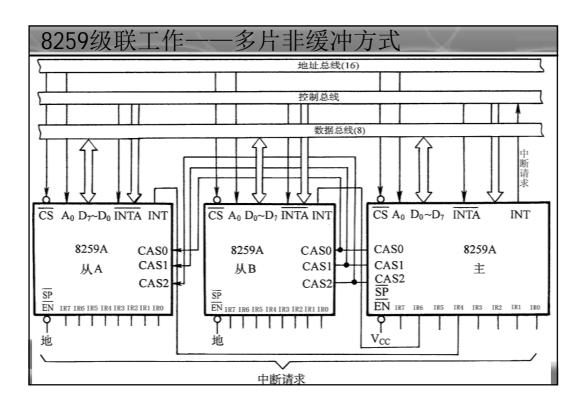


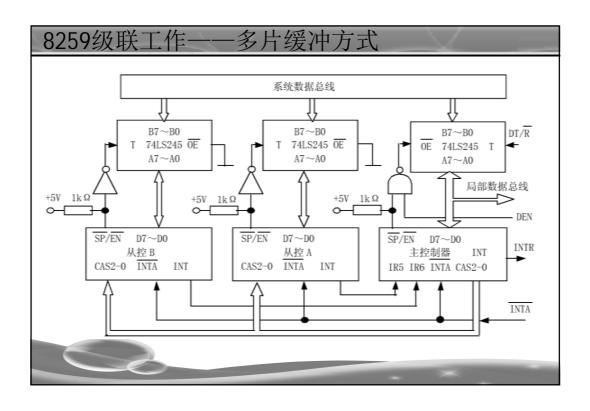


8259级联工作

- Ⅰ 8259A可以级连,1个主片最多可以级连8个从片
 - **n**级连时,主片 $CAS_0 \sim CAS_2$ 连至每个从片的 $CAS_0 \sim CAS_2$,输出被选中的从片ID,
 - \mathbf{n} 每个从片的中断请求信号INT,连至主片8259A某个中断请求输入端IR_i;主片的INT线连至CPU的中断请求输入端INTR
 - n主片在第1个响应周期内通过CAS_{2~0}送出从片ID,相应的从片在第2个响应周期内则将中断类型码N发送到数据总线上。

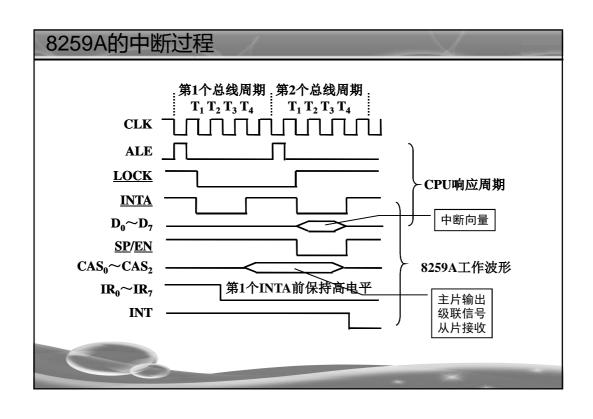






8259A的中断过程

- Ⅰ 1.当中断输入线(IR₇₋₀)有变高,中断请求寄存器IRR相应位置1
- I 2.若中断请求线中至少有一位是中断允许的,则8259A通过INT引脚向CPU送出中断请求
- Ⅰ 3.若CPU开中断,则用INTA响应。第一个响应周期输出从片ID。
- 4.8259A收到INTA响应后,使最高优先权的ISR位置位,相应IRR位复位。优先权的顺序为: IR0 > IR1...... > IR7
- Ⅰ 5.CPU启动第2个响应周期,8259A向DB送出中断类型号N。
- I 6.CPU计算中断向量,转向中断服务程序



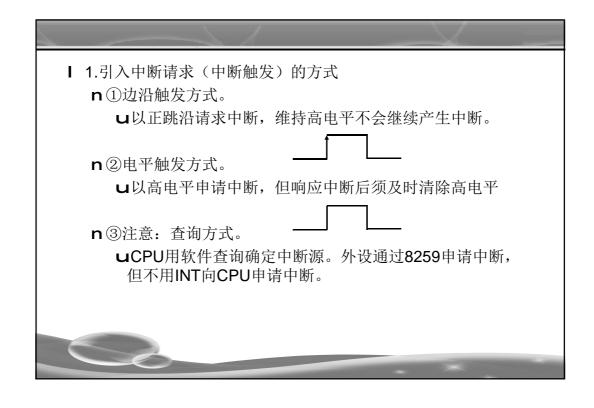
8259A的端口和操作

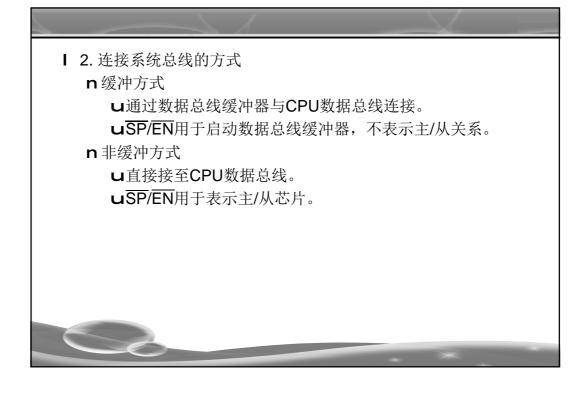
- I 2个端口
 - n 按端口地址区分命令(偶地址Ao=**0**和奇地址A0=**1**)
 - n按顺序或特征位区分命令(同一端口地址)

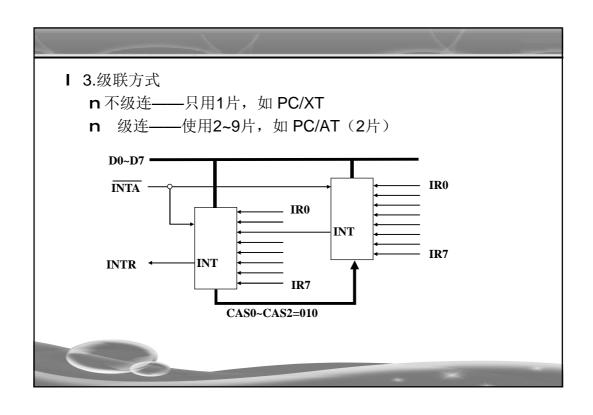
CS	WI	R RI) A0	读写操作
0	0	1	0	写ICW1,OCW2,OCW3
0	0	1	1	写ICW2~ICW4,OCW1
0	1	0	0	读IRR,ISR,查询字
0	1	0	1	读IMR

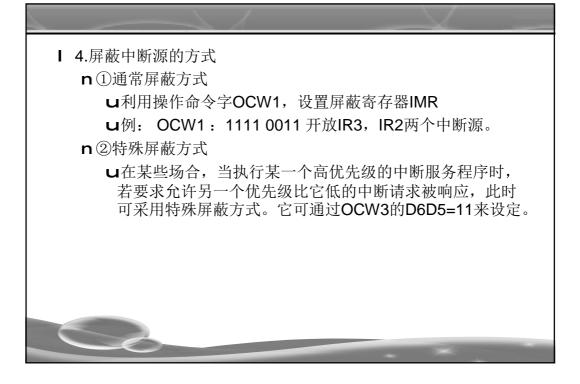
00000	
0 0 0 0 0	
00000	
0 0 0 0 0	
0 0 0 0 0	
	────────────────────────────────────
0 0 0 0	第4节 8259A的工作方式
0 0 0 0 0	
00000	
0 0 0 0 0	
00000	
0 0 0 0 0	
00000	







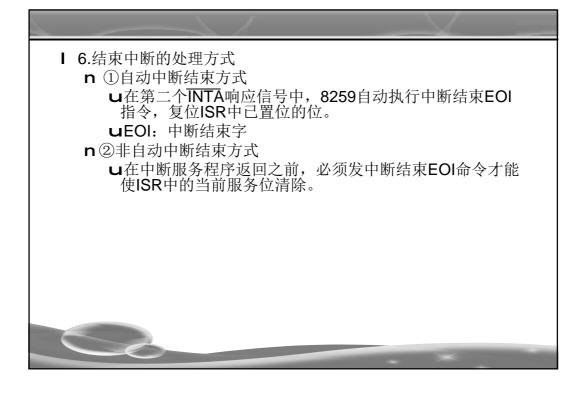


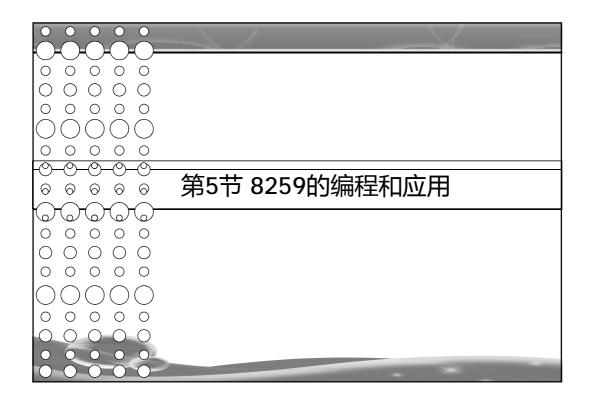


- Ⅰ 5.优先级排队的方式
 - n①全嵌套方式。
 - **山**优先级按0~7顺序排队,且只允许级别高的中断源去中断级别低的中断服务程序。常用缺省方式。
 - n②自动轮换方式
 - **山**中断服务结束后优先级降为最低(7),相邻的低优先级中断源自动升为最高,其余顺变。例: IR₂中断服务结束后:

IR0	IR1	IR2	IR3	IR4	IR5	IR6	IR7
5	6	7	0	1	2	3	4

- **山**每个中断源都有最高优先级的资格,故称"等优先级方式"
- n③优先级指定轮换方式
- n④特殊循环方式



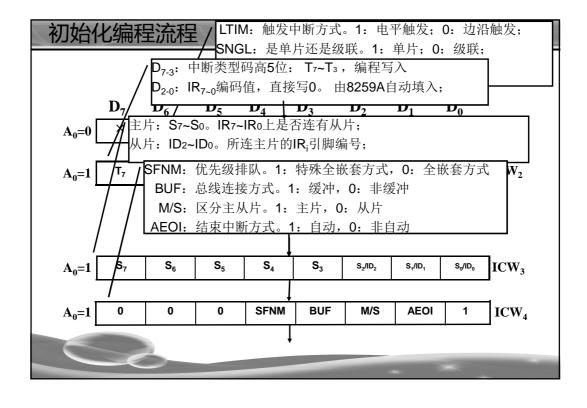


8259的编程

- Ⅰ 8259的编程分为两个阶段
 - n初始化阶段
 - u在系统加电或复位后由初始化程序完成。
 - **u**设定工作方式、工作条件、中断类型码等。
 - **山**初始化命令(Initialize Control Word)
 - \mathbf{p} ICW1 \sim ICW4
 - n操作阶段
 - u对8259的状态、中断方式和工作过程的控制。
 - **山**操作命令(Operation Control Word)
 - **p**0CW1~0CW3

初始化命令 (ICW: Initialize Control Word)

- I ICW1
 - n设置中断请求触发方式及芯片数目,使8259A复位
- I ICW2
 - n 设置中断类型号N: 8位
- I ICW3
 - n设置主从片的硬件连接方式
- I ICW4
 - n设置优先级嵌套方式,中断结束方式,缓冲方式,主从片



补充说明

- I ICW的识别
 - **n**只要命令字D4位为"1",地址位A0为"0",就是ICW1。
 - n接下来1~3字节就是ICW2~ICW4。
- I 在不同的初始化要求中,ICW2~ICW4并非都必须使用。
- I ICW1复位芯片
 - n ICW1清除IMR,缺省设置完全嵌套方式,IRQ_{0~7}优先级递降。
- I ICW2设置中断类型号N(8位)
 - n N高5位编程输入。同片8259A的8个中断源N高5位相同。
 - nN低3位根据IR_i编码
 - u如IR₀的编码为000
 - u如IR₄的编码为100
 - u如IR₇的编码为111

初始化的例子

- Ⅰ 例: 一片8259A, 边沿触发方式; 中断类型码为08H~0FH;
 - n用全嵌套、缓冲、非自动结束中断方式;
 - **n**8259A的端口地址为20H和21H。

MOV AL ,13H;ICW1: 边沿触发,单片,设置IC4 OUT 20H,AL

MOV AL ,8 ; ICW2: 中断类型码为08~0FH

OUT 21H, AL

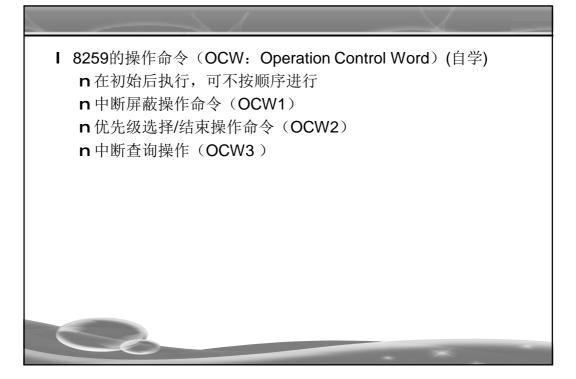
MOV AL, ODH; ICW4:全嵌套、缓冲、非自动结束中断方式

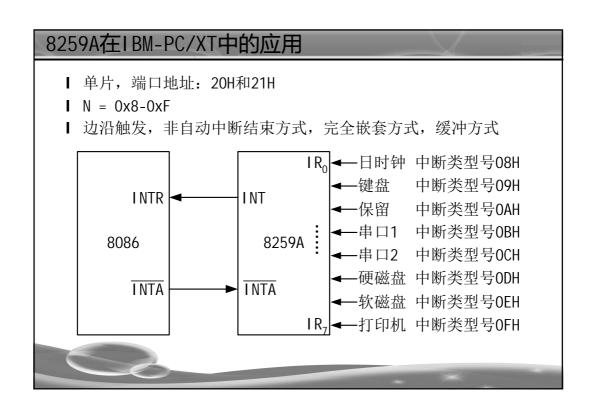
OUT 21H , AL

MOV AL ,13H ;ICW1:边沿触发,单片,设置IC4 OUT 20H, AL \mathbf{D}_6 \mathbf{D}_5 \mathbf{D}_4 \mathbf{D}_3 \mathbf{D}_2 \mathbf{D}_1 ICW_1 LTIM SNGL IC4 o 0 0 1 0 0 1 1

I 中断类型码: 08~0FH n N = 0000 1xxxn N'= 0000 1000 MOV AL ,8 ; ICW2:中断类型码为08~0FH OUT 21H ,AL T_6 **T**₅ T_3 $A_0=1$ ICW₂ 0 0 0 0 1 0 0 思考:如果中断类型码为 020~027H, 如何设置ICW2 ?

MOV AL,0DH ;ICW4:全嵌套、缓冲、非自动结束中断 OUT 21H,AL ICW_4 SFNM BUF M/S AEOI $A_0 = 1$ 0 1 0 0 0 1 1 0





	\mathbf{D}_{7}	\mathbf{D}_{6}	D ₅	D ₄	D ₃	D ₂	D ₁	D ₀
	×	×	×	1	LTIM	×	SNGL	IC4
Ⅰ 初始化编程	Т,	T ₆	T ₅	T ₄	T ₃	×	×	×
■ 79月9日7七9冊7宝	\mathbf{S}_{j}	S _E	S ₅	S⁴	S ₃	S ₂ ID ₂	S, (D,	S _p /D _p
	0	0	0	SFNM	BUF	M/S	AEOI	1
;ICW1,边沿触发,单户	十825	59A,	需IC	CW4				
MOV AL ,00010011	В							
OUT 20H ,AL								
;设置ICW2,中断类型 ^只	号高5	5位为	0000)1				
MOV AL ,00001000	В							
OUT 21H ,AL								
;设置ICW4,非自动中国	断结员	東方式	弋,	已全嵌	套方	式,	缓冲ス	方式
MOV AL ,00001101E	3							
- ,								
OUT 21H ,AL								