

第九章 中断技术



一、中断的基本概念

二、中断控制器 **Intel 8259A**

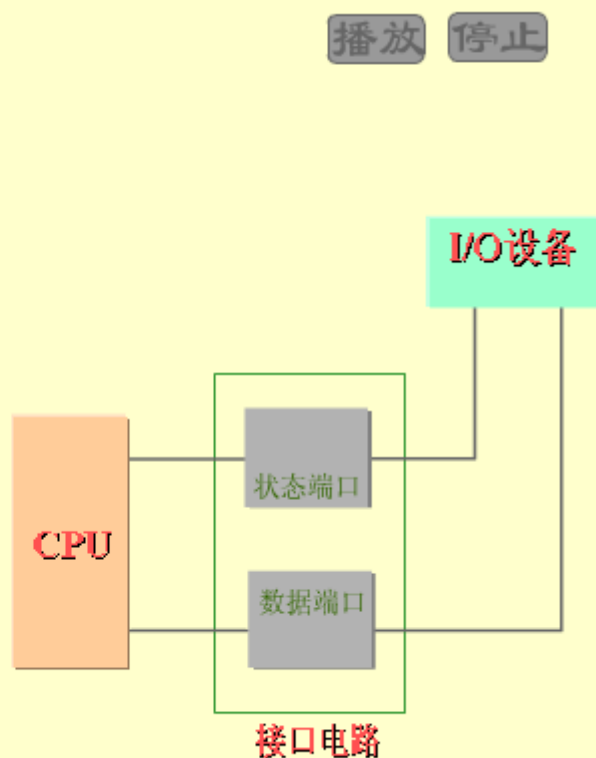
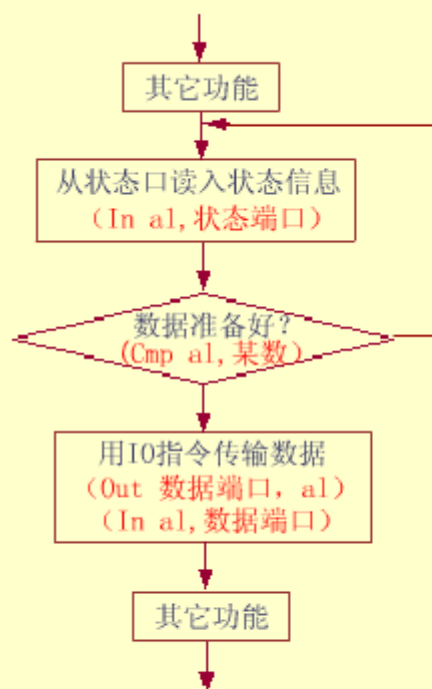
三、**IBM PC/XT** 的基本中断系统

四、应用举例

一、中断的基本概念

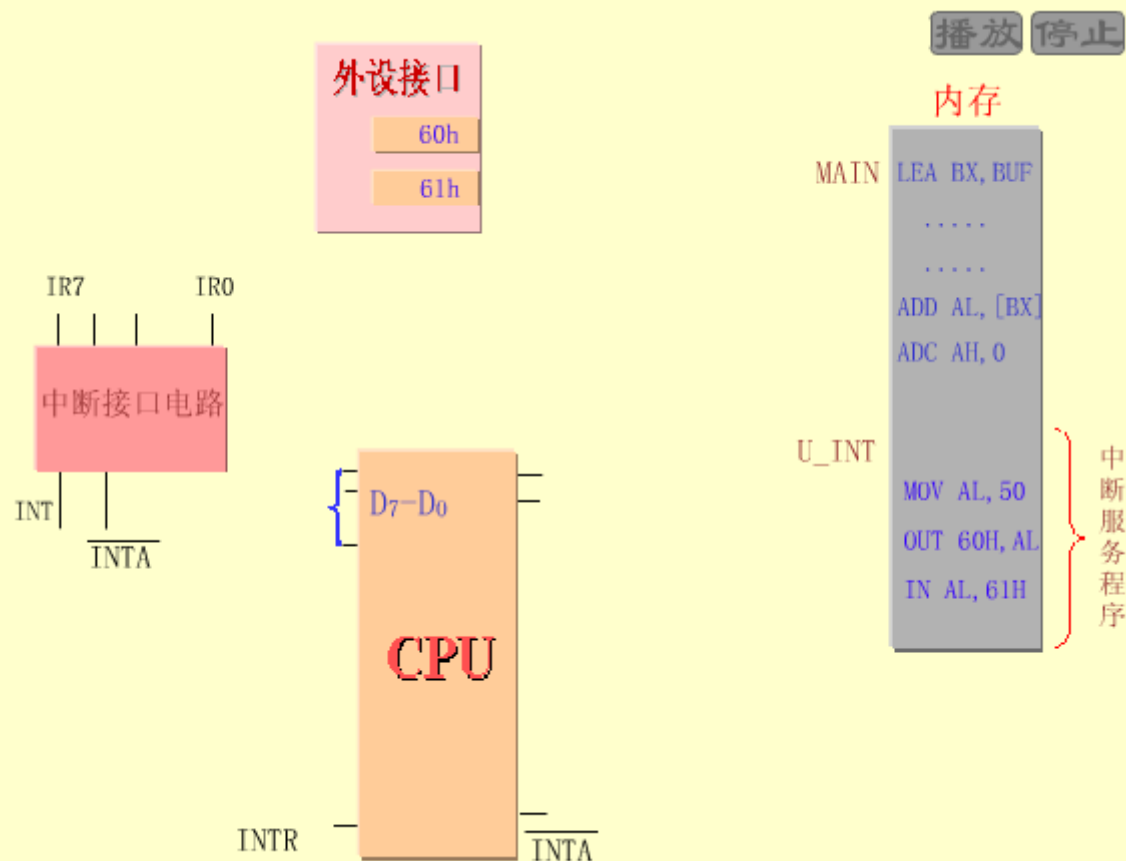
在计算机系统中，中断是提高CPU工作效率的重要手段之一。

当CPU与外设进行数据交换—— ① 查询方式



CPU在不断地查询外设，直到外设数据准备好。占用了CPU大量的时间。

② 中断方式



如果当外设将数据准备好后，**CPU**暂时停止当前的工作，进行与外设的数据交换，这显然要提高**CPU**的工作效率，也解决了慢速的外设与快速的**CPU**间在数据传输时的速度匹配的矛盾。

1. 中断

—— CPU在正常运行程序时，由于程序的预先安排或计算机内外部事件，引起CPU中断正在运行的程序，而转到为预先安排的事件或内外部事件服务的程序中去。

预先安排的事件：

```
mov ah, 1  
int 21h  
add al, 1  
mov ah, 2  
mov dl, al  
int 21h
```

计算机内外部事件：

例如：键盘输入；
打印机输出；
除法溢出；
等。



2. 中断的作用主要

(1) 实时处理

在计算机用于工业控制时，由于很多控制参数发生变化的频率很高，计算机要及时地获得它们的变化情况。当有关参数发生变化时，外部设备则向计算机发出中断请求信号，要求计算机进行处理，从而达到实时数据处理的目的。

(2) 故障处理

在计算机运行时，往往会发生一些无法事先预料到的故障，如电源、内存或运算溢出等。当这些故障出现时，故障源向CPU发出中断请求，CPU对故障进行自动处理。

(3) 同步操作

在很多计算机系统中，外设与CPU经常同时工作。由于外设的工作速度较慢，所以CPU启动外设后，继续运行其它程序，而这时外设也进行数据的准备工作。当外设将数据准备完毕后，向CPU发出中断请求，CPU暂停当前工作，进行相应的数据处理工作。



3. 中断源

在计算机系统中，发出中断请求的来源称为中断源，常见的中断源有：

- (1) 一般的输入、输出设备。如键盘、鼠标、磁盘等；
- (2) 实时时钟。如PC机的计时时钟；
- (3) 系统故障。如电源故障、计算溢出等；
- (4) 断点。通常为调试程序而人为设置的中断点。

4. 中断系统功能

一个完善的中断系统通常要具备以下几个功能：

- (1) 发出中断请求及在中断结束后返回到主程序。
- (2) 对多个中断源，中断系统要能够针对它们的优先级，对它们进行优先权排队
- (3) 高级中断源能中断低级的中断处理。



5. CPU响应中断的工作过程

当一个中断请求到达CPU后，如果CPU的中断允许是开放的，则CPU就对中断的请求进行响应，进入中断周期。其工作过程如下：

- (1) 关闭中断：多数CPU在发出响应中断信号时，自动关闭中断。
- (2) 保留断点：CPU响应中断后，为了在处理完中断请求后正确返回到主程序，CPU要将返回地址压入栈内保存起来。
- (3) 保护现场：由于中断具有很强的随机性，所以，为了保证中断返回后主程序正确运行，要将有关寄存器和运行状态压入栈内保护起来。
- (4) 进行中断服务程序：要进行中断服务程序，就要取得入口地址，不同的CPU有不同的取得入口地址方式，8086/8088CPU是通过中断矢量形成入口地址的。当得到入口地址后，CPU就开始运行中断服务程序。
- (5) 恢复现场：当中断服务程序结束时，将已保护的有关寄存器和状态从栈中弹出，为主程序的正确运行作好准备。
- (6) 开放中断并返回：在中断服务程序的最后，要开放中断为以后的中断请求作好准备，然后使用中断返回指令（如8086/8088中的IRET）返回到主程序。



6. 中断优先权

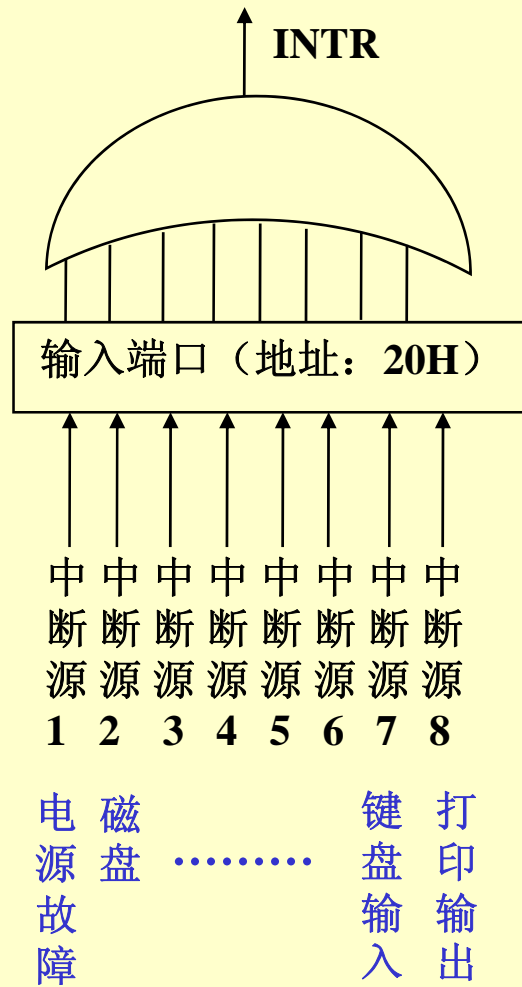
优先权—— 由于一个计算机系统中通常有多个中断源存在，为了保证它们有序工作，要对它们赋予相应的优先级，即优先权。

作用 —— ① 当多个中断请求到达时，**CPU**要响应优先权相对高的请求；
② 当**CPU**正在进行中断服务时，优先权相对高的中断请求能中断优先权相对低的中断服务。

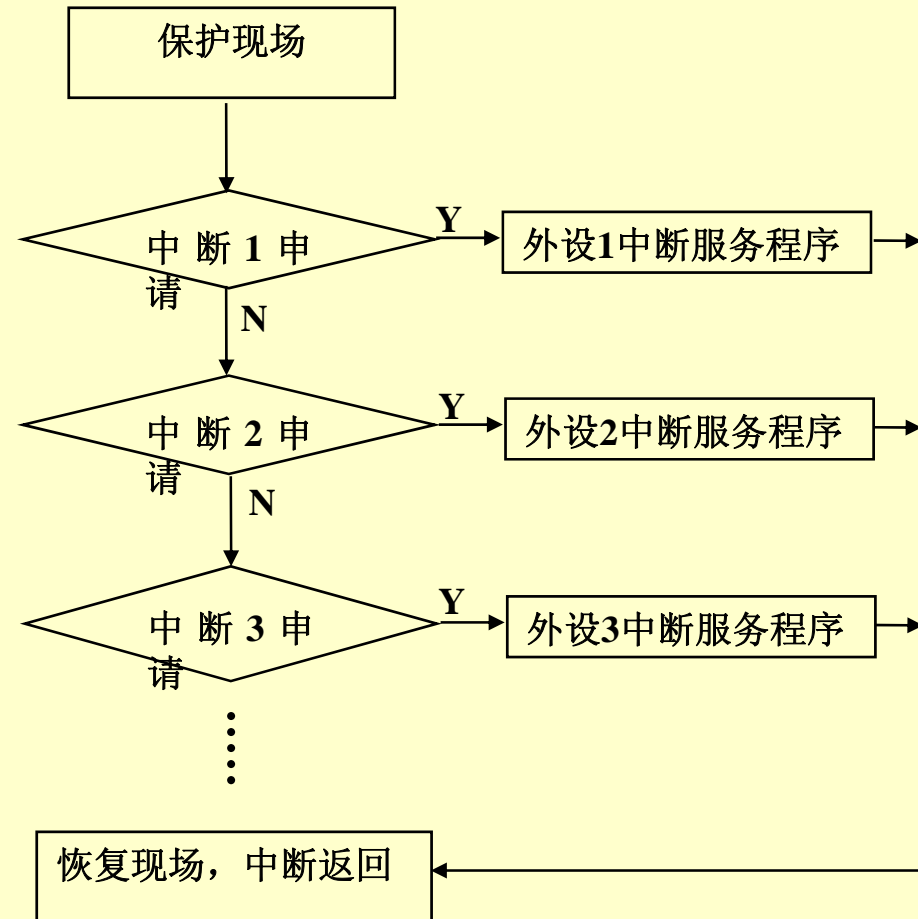
中断优先判断可分为软件处理和硬件处理两种方式。



- 软件方式



软件查询接口



软件查询流程

软件方式通常用于多个中断只有一个服务程序入口地址的情况，多个中断都从一个中断服务程序入口地址进行，在中断服务程序中进行中断优先权的处理。

利用软件进行中断优先权查询的程序有以下两种方式。

屏蔽法：

```
IN  AL, 20H
TEST AL, 80H
JNE  INT7S
IN  AL, 20H
TEST AL, 40H
JNE  INT6S
IN  AL, 20H
TEST AL, 20H
JNE  INT5S
.....
```

位移法：

```
IN  AL, 20H
SHL AL, 1
JC  INT7S
SHL AL, 1
JC  INT6S
SHL AL, 1
JC  INT5S
.....
```

软件查询具有节省硬件，不需要专门的优先权排队电路，程序中查询的次序就是优先权的次序。但由于每次进行中断后都要进行逐位查询，所以服务程序的进行时间相对较长。

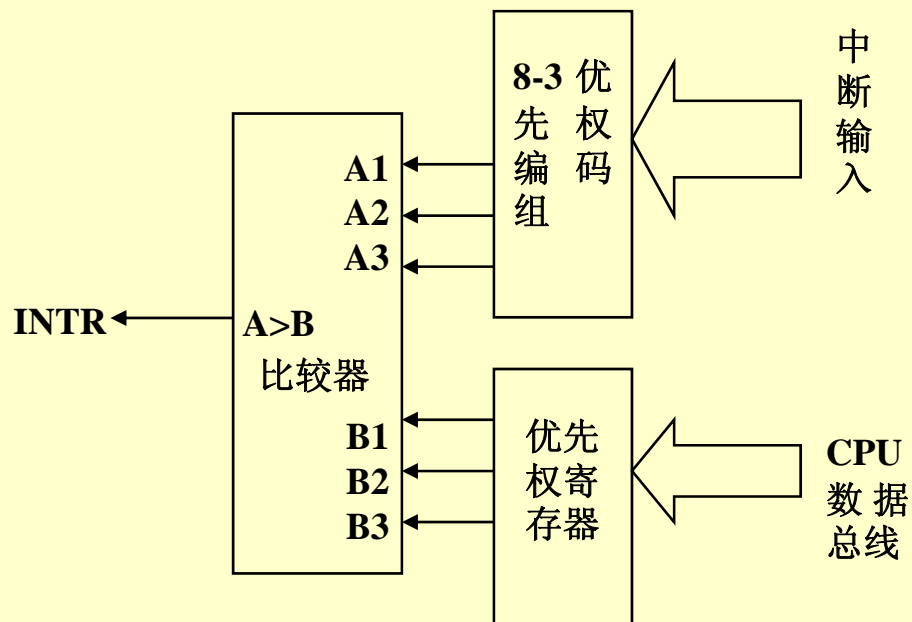


- 硬件优先权排队电路

使用硬件进行中断优先权排队主要有以下两种方式：

比较器方式

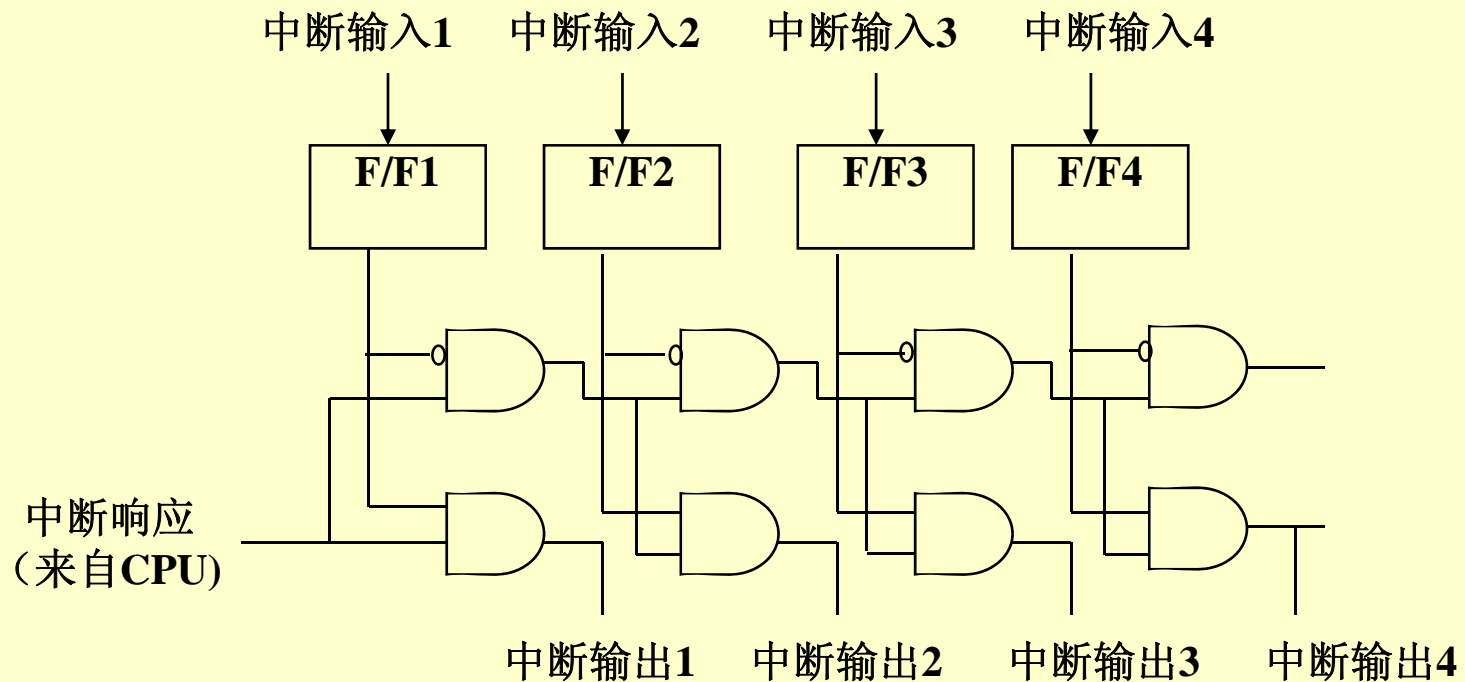
在图中优先权寄存器中存放的是当前在服务的中断码，如没有中断服务则为0，当有中断请求到来时，从中断输入A端输入的中断信号被编码，并与B端的在服务中断号进行比较，如A小于或等于B则从INTR端输出一低电平。否则为高电平，向CPU发出中断申请信号。当优先权较高的中断优先权较低的中断服务时，进入优先权较高的中断服务前，要先保存上次在服务的中断号，当本次中断结束时，要将上次在服务的中断号送回到优先权寄存



比较器方式中断优先排队电路

维菊花环电路方式

当中断输入 n 发中断信号时，如没有在服务中断，同时，中断是开放的，则CPU响应中断请求 n 。每个中断输入可屏蔽其下一级的中断请求信号的输出，并可将屏蔽信号传递到下面的各级。当有某级中断在服务时，比它级别高的中断输入请求中断时，高级别中断将被CPU响应，同时屏蔽其以下的各级中断。从而达到了中断优先排队目的。

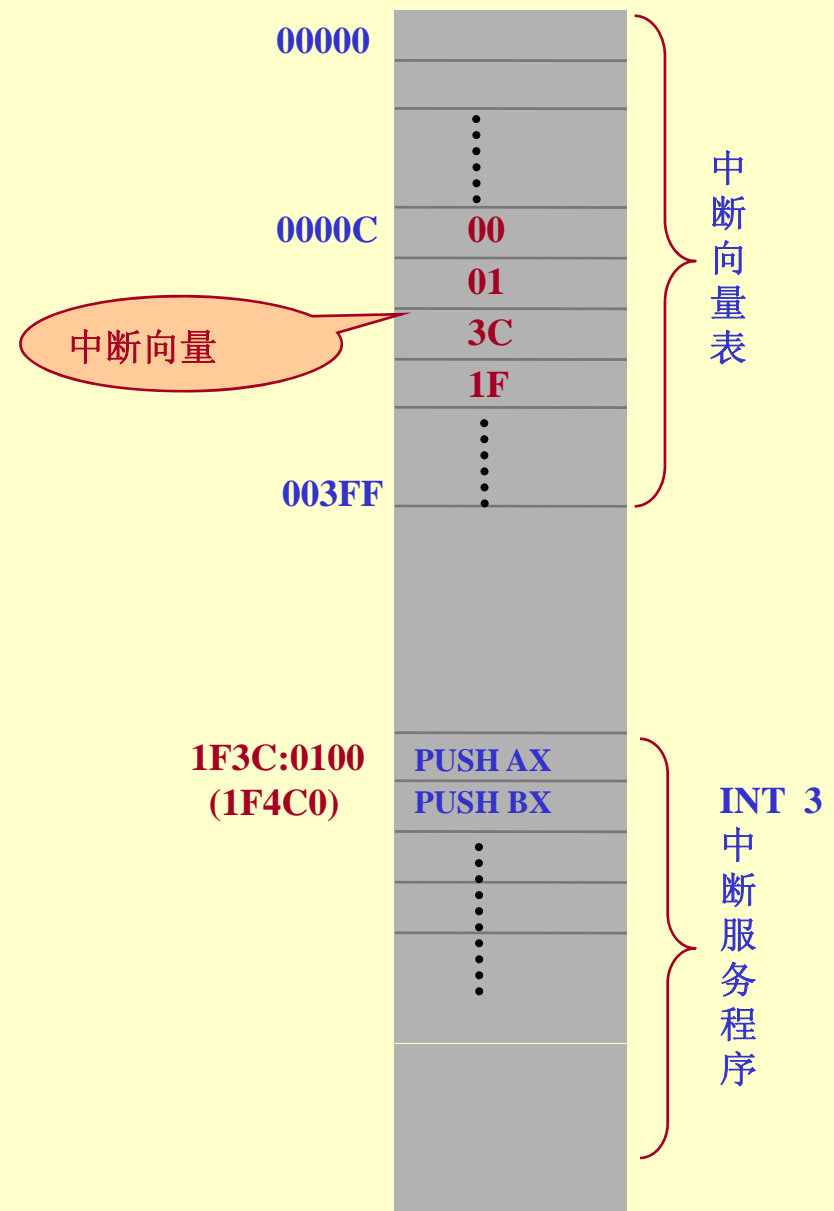


维菊花环电路优先权排队电路

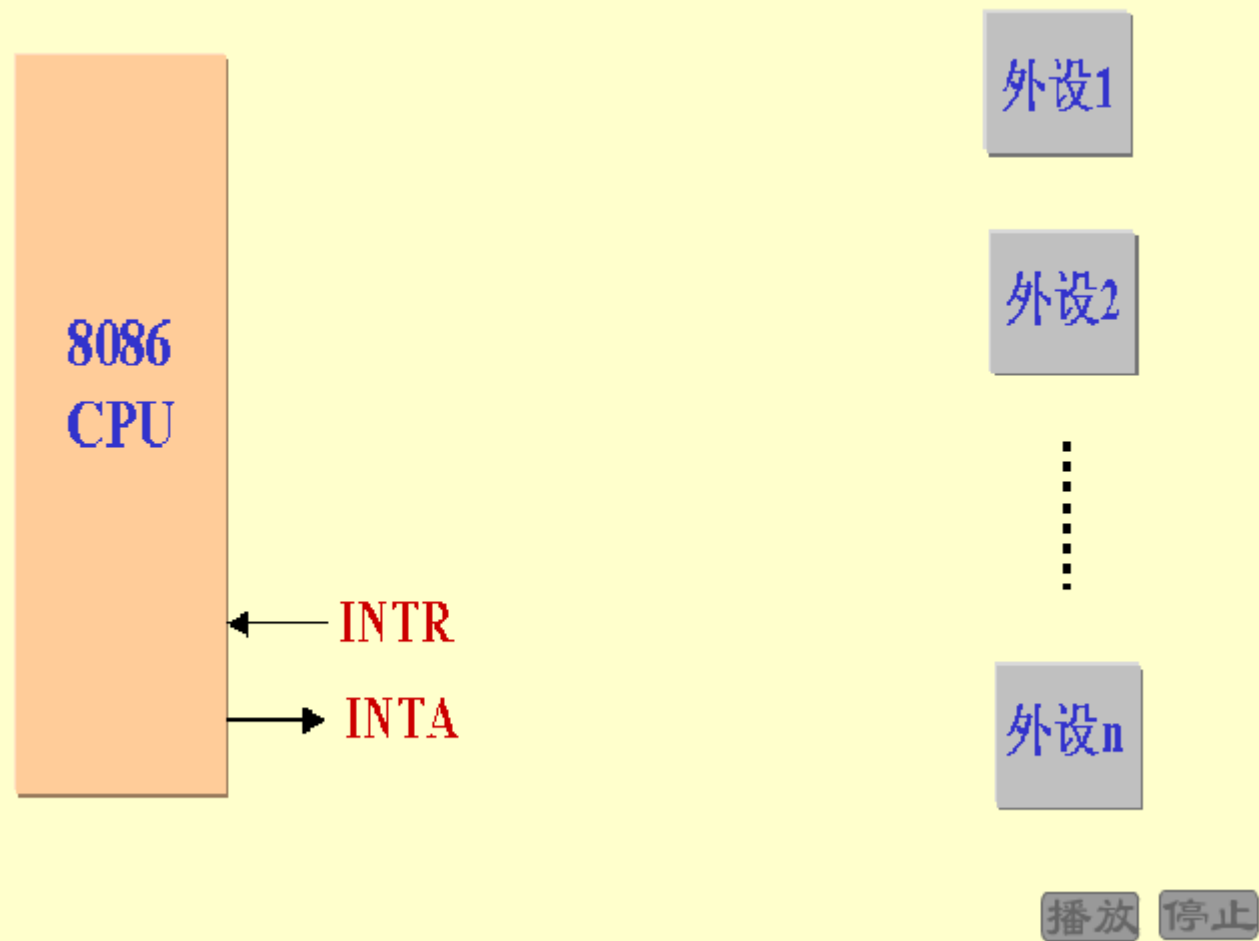
7. 中断向量

当中断请求被响应时，CPU将停止当前运行的程序，转入中断服务程序，而每个中断服务程序都有一个程序的入口地址，这个入口地址就是中断向量。内存中存放一组中断向量的区域被称为中断向量表。当中断被响应后，CPU根据中断的型号，计算出中断向量在中断向量表中的位置，然后从表中取出中断服务程序的入口地址（中断向量），并进入中断服务程序。

```
mov ah, 1  
int 21h  
add al, 1  
int 3
```



二、中断控制器 Intel 8259A



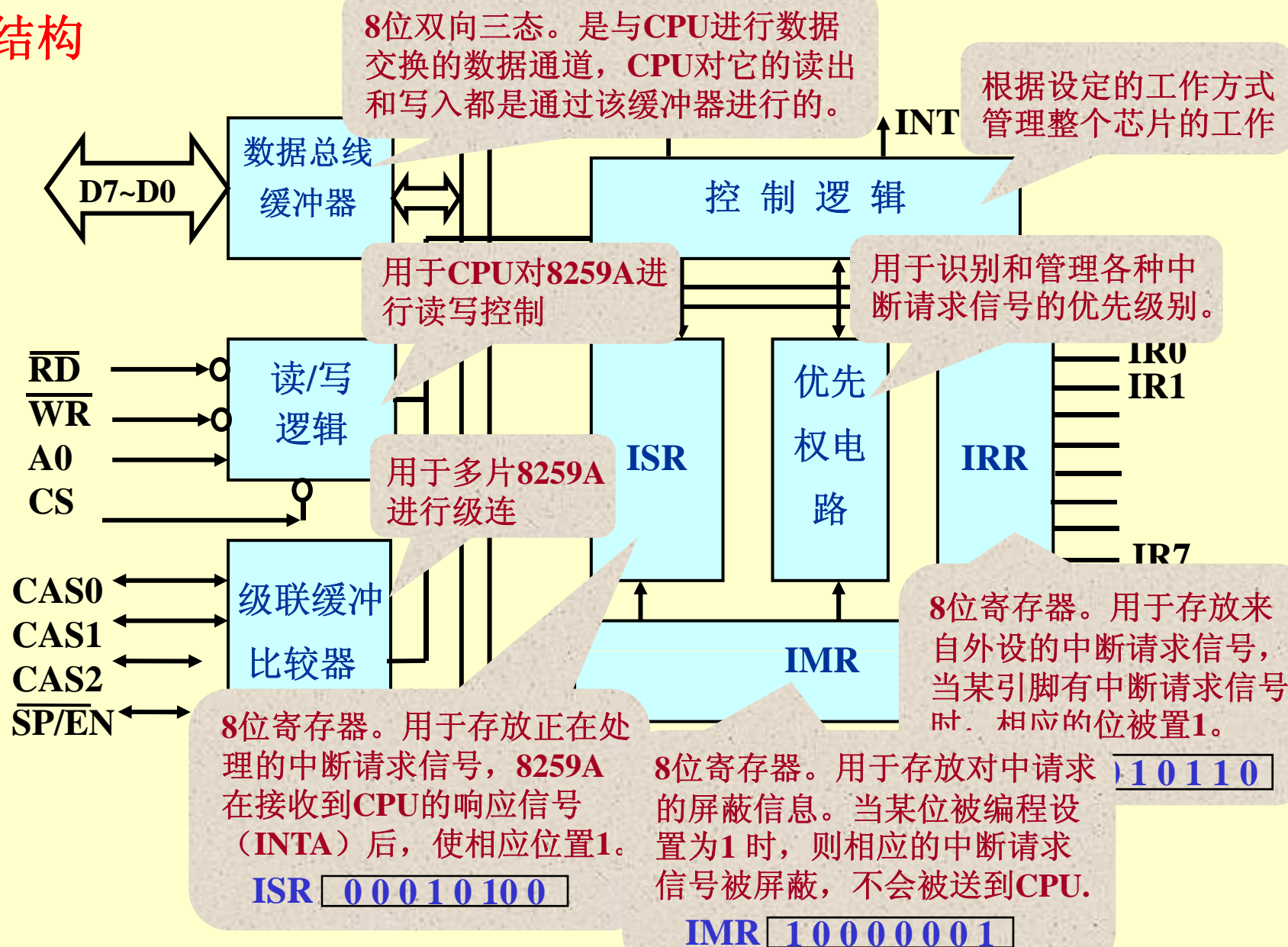
1.功能

Intel 8259A是兼容于8080/8085和8086/8088系列的可编程中断控制器，其功能主要有：

- 1) 单片8259A可管理8级中断；
- 2) 通过级连，可管理多于8级的中断，在不增加其它电路的前提下，可使用8片8259A构成64级主从式中断系统；
- 3) 每一级中断都可通过编程使之被屏蔽或被允许；
- 4) 8259A可通过编程被设定为几种不同的工作方式；
- 5) 8259采用NMOS工艺，只需要一组5V电源。



2. 结构



管脚信号

$\overline{\text{CS}}$	1	28	Vcc
$\overline{\text{WR}}$	2	27	A0
$\overline{\text{RD}}$	3	26	$\overline{\text{INTA}}$
D7	4	25	IR7
D6	5	24	IR6
D5	6	23	IR5
D4	7	22	IR4
D3	8	21	IR3
D2	9	20	IR2
D1	10	19	IR1
D0	11	18	IR0
CAS0	12	17	$\overline{\text{INT}}$
CAS1	13	16	$\overline{\text{SP/EN}}$
GND	14	15	CAS2

8259管脚对照表

D7~D0	数据总线（双向）
RD	读输入
WR	写输入
A0	选择内部寄存器地址
$\overline{\text{CS}}$	片选
CAS2~CAS2	级连线
SP/EN	从程序/允许缓冲
INT	中断输出
$\overline{\text{INTA}}$	中断响应输入
IR0~IR7	中断请求输入



8259A的读写操作

A ₀	D ₄	D ₃	RD	WR	CS	输入操作（读）
0			0	1	0	IRR、ISR 或中断类型码 → 数据总线
1			0	1	0	IMR → 数据总线
						输出操作（写）
0	0	0	1	0	0	数据总线 → OCW ₂
0	0	1	1	0	0	数据总线 → OCW ₃
0	1	X	1	0	0	数据总线 → ICW ₁
1	X	X	1	0	0	数据总线 → ICW ₂ 、ICW ₃ 、ICW ₄ 、OCW ₁
						断开功能
X	X	X	1	1	0	数据总线 —— 3态（无操作）
X	X	X	X	X	1	数据总线 —— 3态（无操作）



3. 工作方式

[1] 设置优先级的方式

(1) 全嵌套方式

全嵌套方式是**8259**最常用的工作方式。初始化后若没有再设置其它优先级方式（如优先权旋转），**8259**自动进入全嵌套方式。其特点是：

- 中断请求优先级顺序是**IRQ₀**——**IRQ₇**。
- 当一个中断被响应时，中断服务寄存器**ISR**得到相应的那一位被置为“1”，且保持到中断服务程序在返回前发中断结束命令为止。
- 在**ISR**位置位期间，禁止再发生同级或较低级的中断，较高级的中断请求仍能产生中断。
- 每个中断请求可分别由**IMR**相应位屏蔽。

(2) 特殊全嵌套

特殊全嵌套方式同全嵌套方式只有一点不同，即：在特殊全嵌套方式下，当正处理某一级中断时，又有同级的中断请求产生时，**8259A**也会响应，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于多片**8259A**级连的系统中。



(3) 优先级自动旋转方式

优先级自动旋转方式一般用于系统中有多多个优先级相等的中断源的情况。在这种方式下，优先级队列是变化的，一个设备受到服务后，其优先级自动降到最低。初始优先级队列是固定的，为：

IRQ_0 IRQ_1 IRQ_2 IRQ_3 IRQ_4 IRQ_5 IRQ_6 IRQ_7 （由高到低）。

如果这时 IRQ_1 有中断请求，则在完成其中断服务后，优先级队列变为（由高到低）： IRQ_0 IRQ_2 IRQ_3 IRQ_4 IRQ_5 IRQ_6 IRQ_7 IRQ_1

优先级自动旋转方式由操作命令字 OCW_2 设定。

(4) 优先级特殊旋转方式

优先级特殊旋转方式与优先级自动旋转方式只有一点不同，即：在优先级特殊旋转方式中，初始的最低优先级由编程来决定，例如，设定 IRQ_3 为最低优先级，那么， IRQ_4 即为最高优先级。

优先级自动旋转方式也是由操作命令字 OCW_2 设定。



[2] 屏蔽中断源的方式

(1) 普通屏蔽方式

在该方式中，8259A的每一条中断请求线都可根据需要设置屏蔽，从而使这个中断请求不能从8259A送到CPU。

可以通过操作命令字OCW₁来设置屏蔽。8259A内部有一个中断屏蔽寄存器IMR，它的每一位对应了一个中断请求输入，通过设置OCW₁，使IMR中任一位或几位为1，当某位为1时，对应的中断请求被屏蔽。

(2) 特殊屏蔽方式

在某些应用场合，希望能动态地改变系统的优先级结构，即在执行中断服务程序的某一部分时，希望禁止某些较低级的中断请求，而在执行中断服务程序的另一部分时，又能够开放比自己级别低的中断请求。要达到这种目的，可采用特殊屏蔽方式。

设置了特殊屏蔽方式后，在中断服务程序中用OCW₁将本级中断屏蔽的同时，使当前中断服务寄存器ISR的对应位自动清0，从而可以接受其它级别较低的中断。（注意：在正常屏蔽方式中，设置OCW₁屏蔽本级中断时，ISR的对应位仍为1，故不能接受较低级的中断请求。）

特殊屏蔽方式由操作命令字OCW₃来设置。



[3] 结束中断处理的方式

(1) 自动中断结束方式 (AEIO)

自动中断结束方式由初始化命令字ICW₄设置。

(2) 非自动中断结束方式

在这种方式下，当中断服务程序结束返回时，必须向8259A送中断结束命令（EOI），8259A在接到该命令后，将中断服务寄存器中相应位复位。它有两种情况：

- 非特殊EOI命令：在全嵌套方式下，EOI命令能自动地把当前ISR中优先级最高的一位清0。因为在全嵌套方式中，最高的非零ISR位对应了最后一次被响应和被处理的中断。也就是当前正在处理的中断，所以，它的复位相当于结束了当前正在处理的中断。
- 特殊EOI命令：在非全嵌套方式下，因无法确定最后响应的是哪一级中断，故应向8259A发特殊EOI命令。这个命令指出了要清除当前中断服务寄存器ISR中的哪一位。

两种EOI命令由操作命令字OCW₂设定。



[4] 中断查询方式

8259A不向CPU发INT请求信号，而由CPU发查询命令来获得对请求设备的中断优先级。其特点是：

- 外部设备仍然通过8259A发中断请求信号要求CPU进行服务，但8259A不是用INT信号向CPU发中断请求信号。
- CPU内部的中断触发器复位，禁止外部对CPU的中断请求。
- CPU使用软件查询确认中断源，进而对外部设备进行中断服务。

通过OCW₃发出查询命令。

[5] 多片级连方式

一个8259A主片至多带8个从片，因此可扩展控制64个中断源。所有的主片和从片均分别进行初始化，且可社定位不同的工作方式。

[6] 缓冲方式

- 缓冲方式：在多片8259A级连的大系统中，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8259A使，一般将它直接与数据总线相连；另外，在一些不太大的系统中，即使有几片8259A级连工作，也可以将8259A直接与数据总线相连。这种连接即位非缓冲方式。

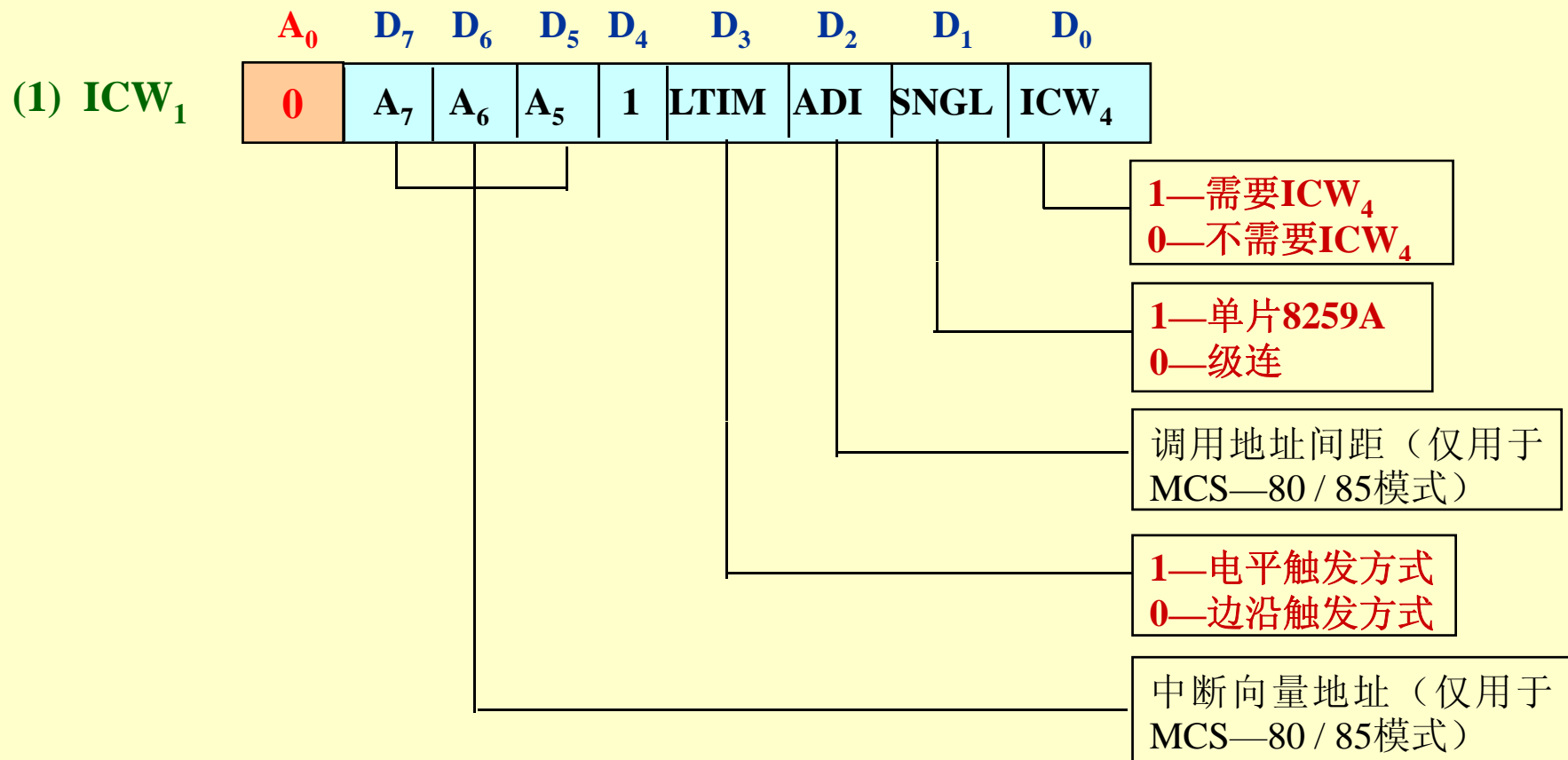
缓冲方式由ICW₄设置。



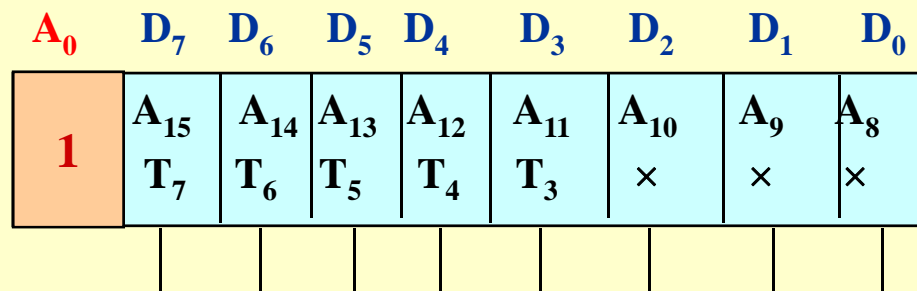
4. 8259A的编程

[1] 初始化命令字

初始化命令字ICW（Initialization Command Word）通常是系统开机时，由初始化程序填写，且在整个系统工作过程中保持不变。初始化命令字必须按顺序填写。
下面我们讨论来讨论它的格式及填写规则。



(2) ICW₂



中断向量地址的A₁₅~A₈位
(MCS—8080 / 8085 模式)

中断向量地址的T₇~T₃位
(8086 / 8088模式)

8259A传送的中断向量

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

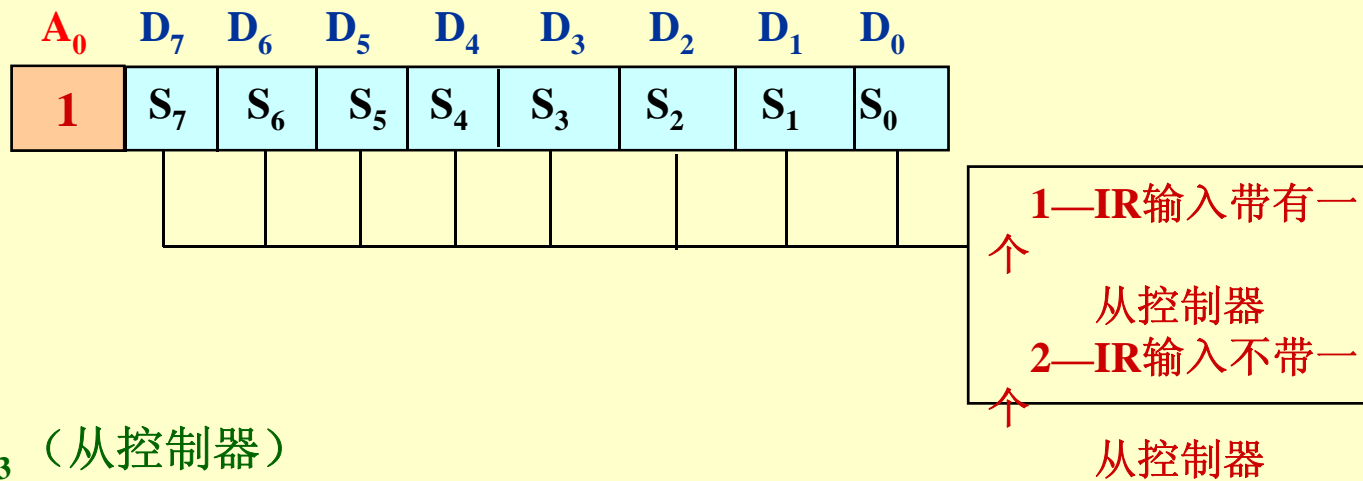
注:

T₇~T₃ — 中断向量类型码高位,
由编程决定, 用于
PX86—86系统。

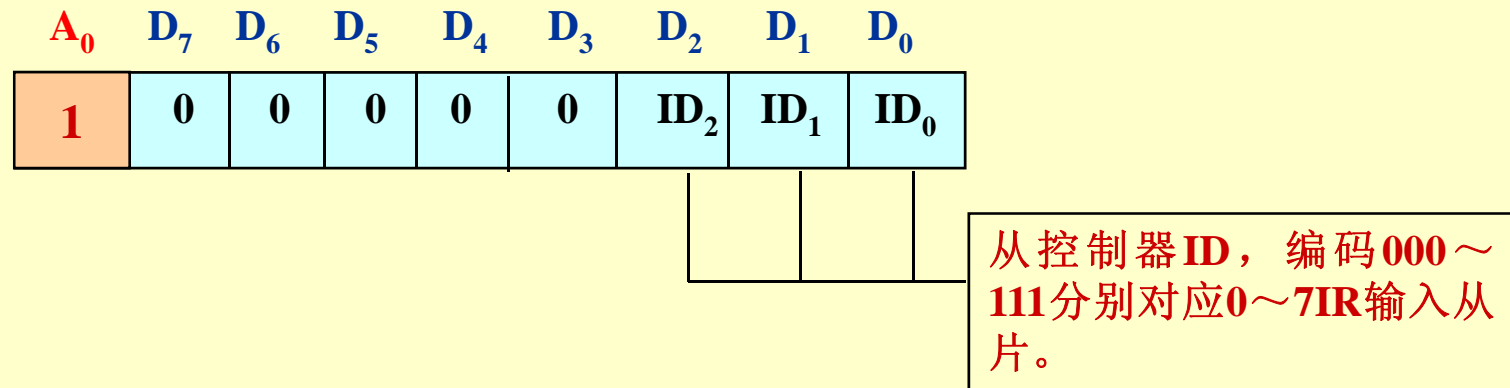
低三位—自动设置为 IR0 ~ IR7
的类型编码。



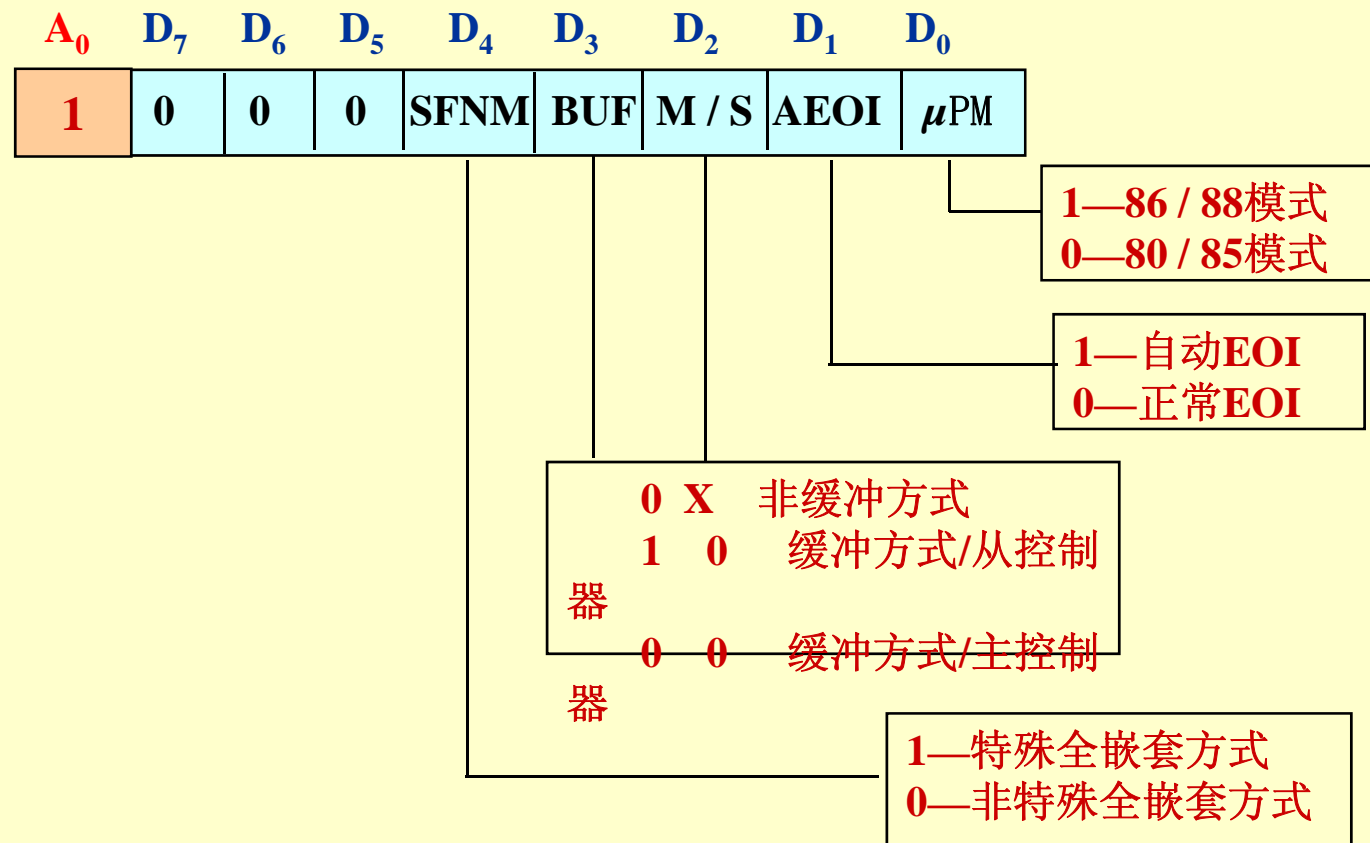
(3) ICW₃ (主控制器)



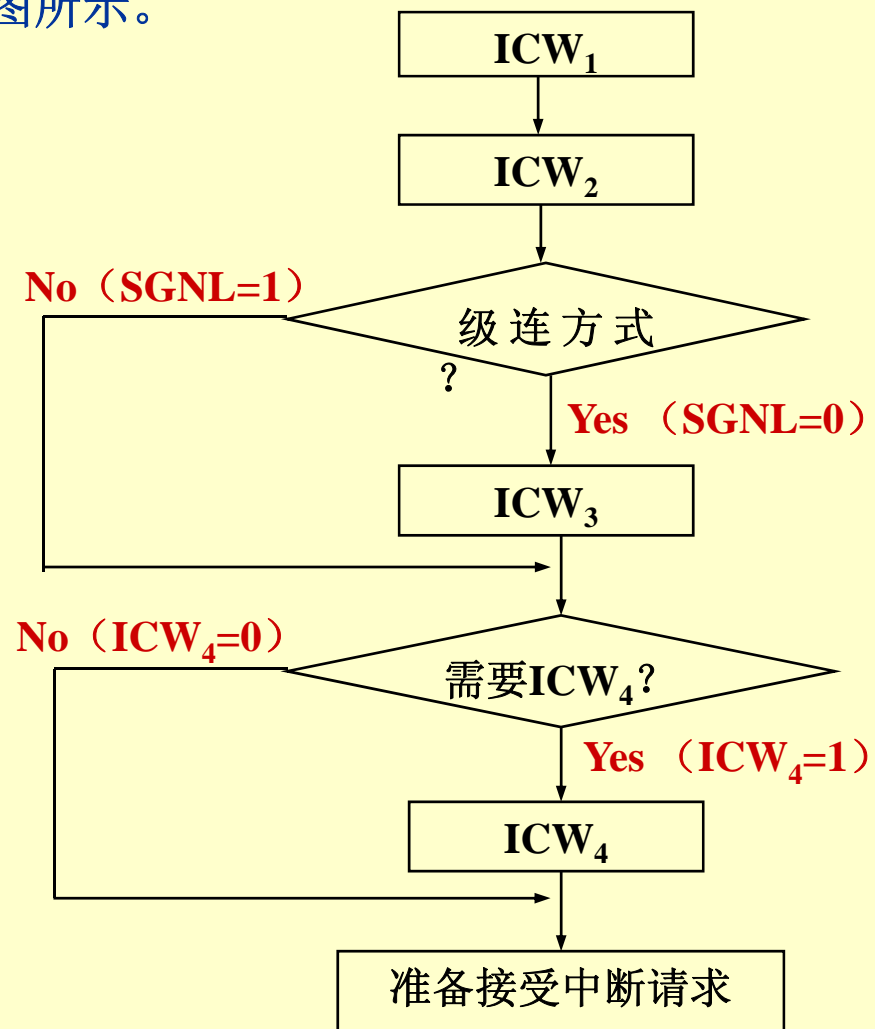
ICW₃ (从控制器)



(4) ICW₄



8259A初始化过程的流程图如图所示。



[2] 操作命令字

由CPU向8259A送三个字节的操作命令字OCW（Operation Command Word），用以规定8259A的工作方式，如：

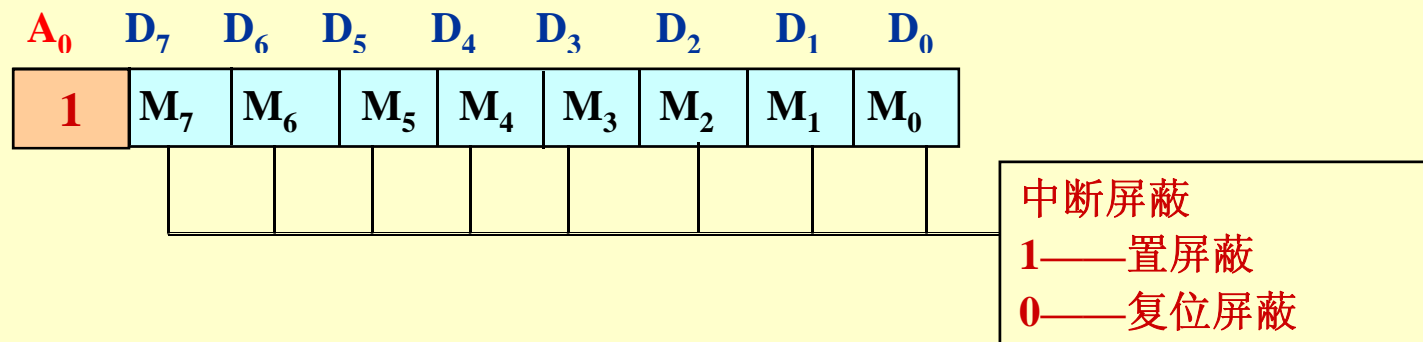
- 中断屏蔽
- 结束中断
- 优先级旋转
- 中断状态 等。

操作命令字是在应用程序中设置的，设置时，顺序上没有严格的要求。

(1) OCW₁

OCW₁是中断屏蔽命令字，命令格式如图。

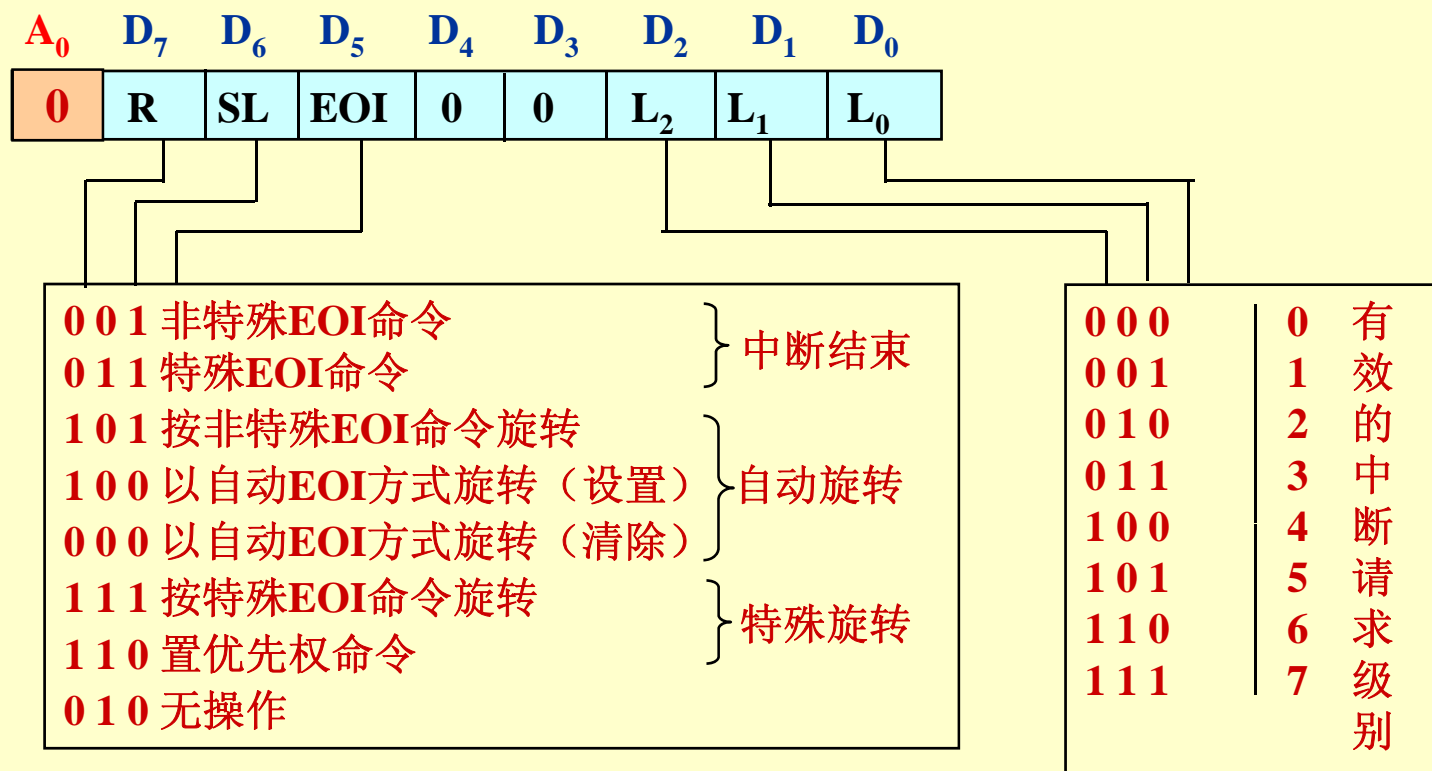
命令字中的M₇~M₀对应IMR的各位。M_i=1表示该中断被屏蔽，M_i=0表示该中断开放。



(2) OCW₂

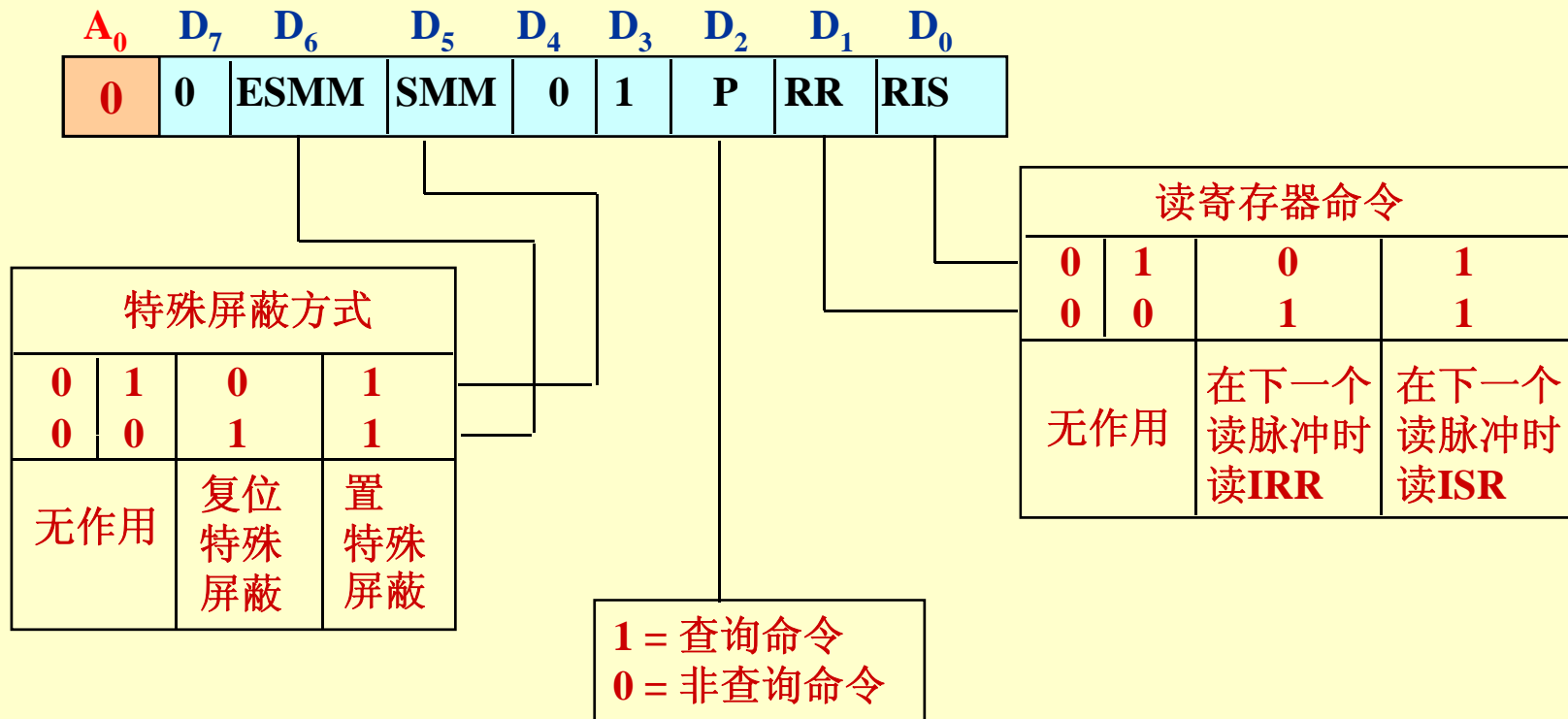
命令格式如图。

- R、SL、EOI的不同组合决定了8259A的几种工作方式。
- L₂~L₀ 这几位确定当SL=1时指定的特殊结束或特殊循环时的中断优先级。



(3) OCW₃

用于控制8259A的运行方式。命令格式如图。



[3] 初始化编程

PC机中使用单片8259A，定义地址为：20H、21H，8259A的工作方式设置如下：

- (1) 固定优先权级别：IR₀最高，IR₇最低；
- (2) 8个中断类型码分别是 08H~ 0FH；
- (3) 中断请求信号采用边沿触发方式；
- (4) 中断服务结束时，采用一般中断结束命令（EOI）；
- (5) 采用一般屏蔽方式。

初始化程序如下：

程序：

初始化命令字：

ICW₁: 0 0 0 1 0 0 1 1 B 13H

ICW₂: 0 0 0 0 1 0 0 0 B 08H

ICW₄: 0 0 0 0 0 0 0 1 B 01H

MOV AL,13H

OUT 20H,AL

MOV AL,08H

OUT 21H,AL

MOV AL,01H

OUT 21H,AL



三、IBM PC/XT 的基本中断系统

1. 中断类型

IBM PC / XT中有两种类型的中断：

(1) 内部中断：即软件中断。包括被零除、单步、溢出和中断指令（**INT n**）等。
这是由**8088**执行指令产生的中断。

(2) 外部中断：

8088A有两条外部中断请求线——非屏蔽中断请求线**NMI** 和
可屏蔽中断请求线**INTR**。

- 非屏蔽中断**NMI**。非屏蔽中断来源于三方：

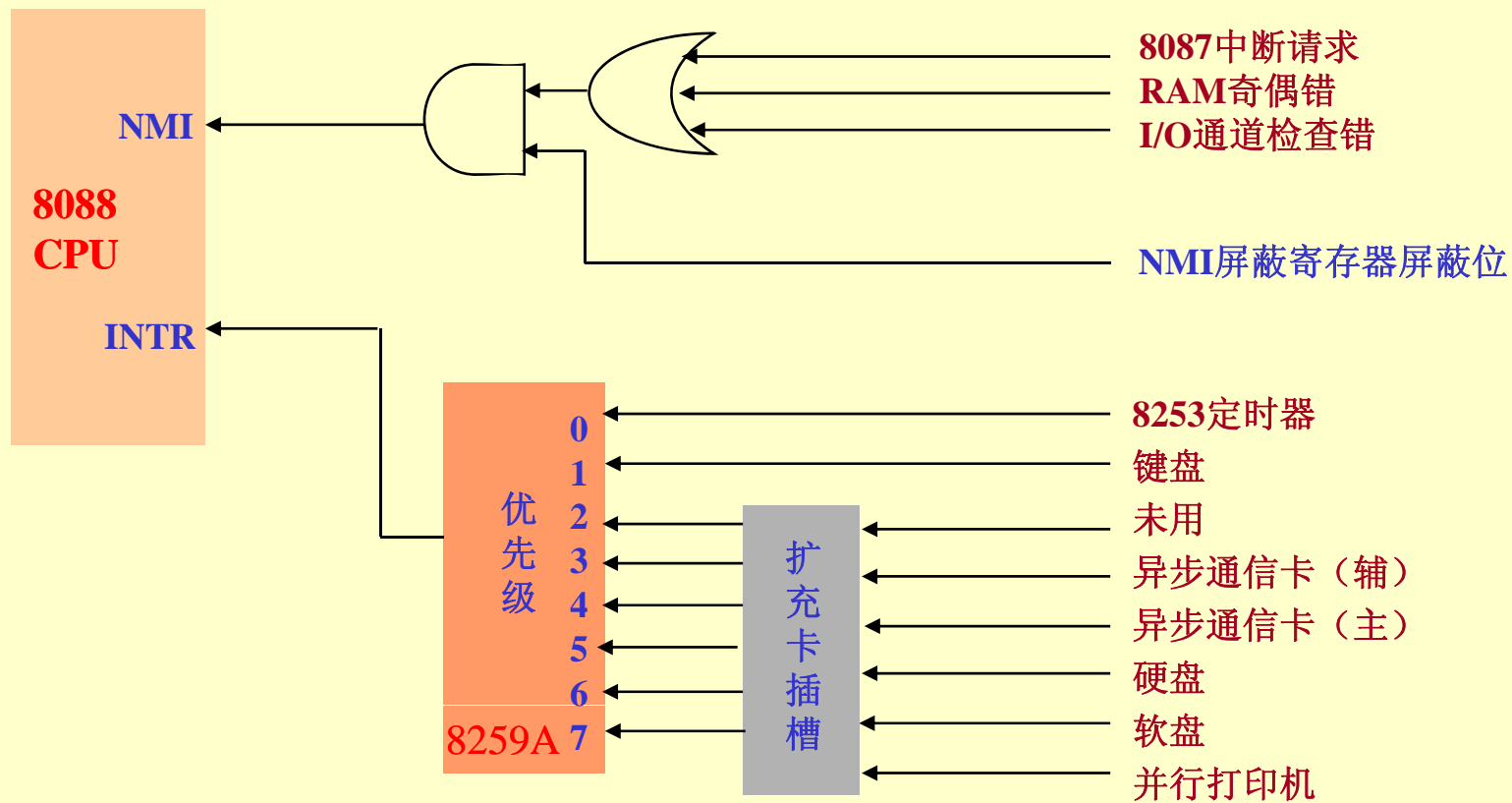
系统板的**RAM**在读写时产生奇偶校验错；

I/O通道中的扩展选件出现奇偶校验错；

协处理器**8087**的异常状态产生的中断。

- 可屏蔽中断**INTR**。由外部设备通过一片**8259A**产生的中断请求。





2. 中断向量表

8088在内存的前1k字节建立了一个中断向量表，可存储256个中断向量，每个中断向量占用4个字节，分别存放中断服务程序入口地址的段地址和偏移量。

每个中断向量以类型码加以区别，如：

Int 21h

Int 10h

键盘中断 类型码 9

当执行中断时，CPU将类型码乘以4后，得到中断向量地址，进而得到IP和CS的值，它就是中断服务程序的入口地址，程序由此转入中断服务程序执行。

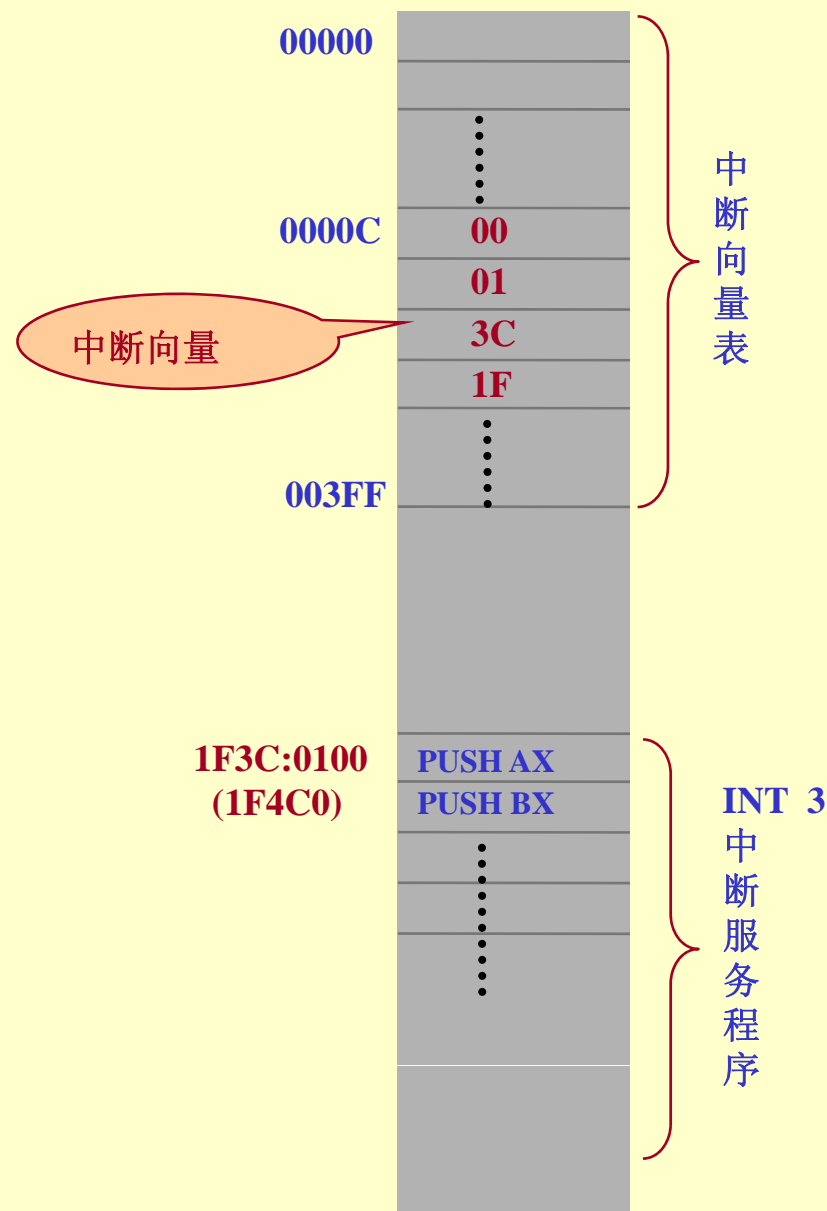


表6-3 中断向量表

地址 (Hex)	类型码	中断名称	地址 (Hex)	类型码	中断名称
0—3	0	除以0	40—43	10	视频显示I/O调用
4—7	1	单步	44—47	11	设备检测调用
8—B	2	非屏蔽	48—4B	12	存储器容量检测
C—F	3	断点	4C—4F	13	软/硬盘I/O调用
10—13	4	溢出	50—53	14	通信I/O调用
14—17	5	打印屏幕	54—57	15	盒式磁带I/O调用
18—1B	6	保留	58—5B	16	键盘I/O调用
1D—1F	7	保留	5C—5F	17	打印机I/O调用
20—23	8	定时器	60—63	18	ROM BASIC
24—27	9	键盘	64—67	19	引导程序入口
28—2B	A	保留的硬中断	68—6B	1A	时间调用
2C—2F	B	异步通信 (COM2)	6C—6F	1B	键盘CTRL-BREAK控制
30—33	C	异步通信 (COM1)	70—73	1C	定时器报时
34—37	D	硬盘	74—77	1D	显示器参数表
38—3B	E	软盘	78—7B	1E	软盘参数表
3C—3F	F	并行打印机			



地址 (Hex)	类型码	中断名称
7C—7F	1F	字符点阵结构参数表
80—83	20	程序结束, 返回DOS
84—87	21	DOS系统功能调用
88—8B	22	程序结束地址
8C—8F	23	CTRL_BREAK退出地址
90—93	24	标准错误出口地址
94—97	25	绝对磁盘读
98—9B	26	绝对磁盘写
9C—9F	27	程序结束, 驻留内存
A0—FF	28—3F	为DOS保留
100—17F	40—5F	保留
180—19F	60—67	为用户软中断保留
1A0—1FF	68—7F	不用
200—217	80—85	BASIC使用
218—3C3	86—F0	BASIC运行时, 用于解释
3C4—3FF	F1—FF	未用



3. 中断的执行

(1) 软中断

当PC机执行 INT n 等指令时，即产生了软中断，起执行过程如下：

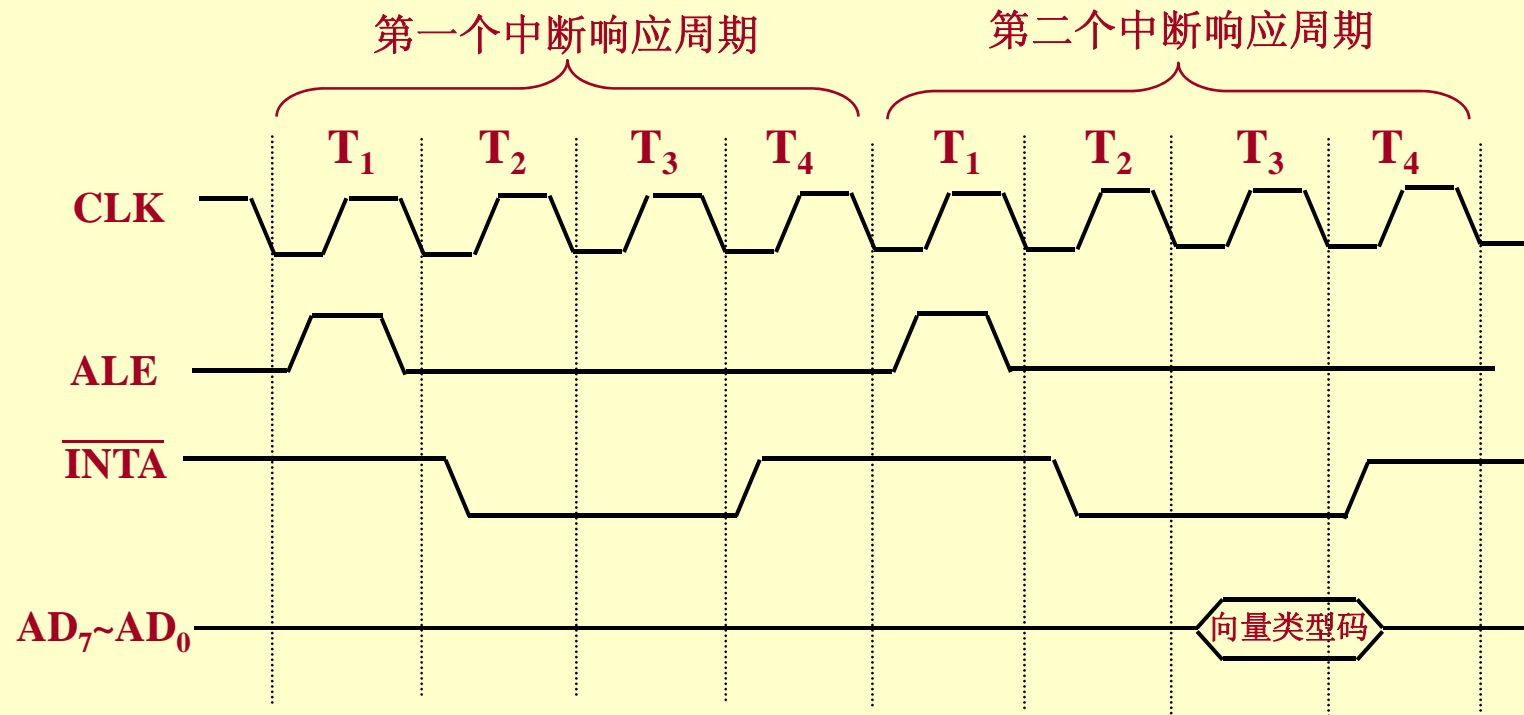
- 栈指针SP减2，标志寄存器内容入栈，然后清除T和I标志，以屏蔽中断。
- SP再次减2，将CS内容入栈。
- 用中断类型码 n乘 4，计算中断向量地址，将向量中的后两个字节送CS寄存器。
- SP再次减2，把IP内容入栈，把中断向量中的前两个字节送至IP。
- 由当前的CS和IP值作为段地址和偏移量，转入相应的中断服务程序中去执行。



(2) 硬中断

中断响应周期

8088响应中断，到转入中断处理，要经过两个中断响应周期。



第一个 $\overline{\text{INTA}}$ 信号：将在服务寄存器ISR相应位置位，表示正在为响应的那一级服务；

同时将中断请求触发器复位。为本级再次中断请求做好准备。

第二个 $\overline{\text{INTA}}$ 信号：要求8259A输入中断向量类型码。

执行过程

- 接口设备通过中断请求线 $IR_0—IR_7$ 中的一条，将8259A内部中断触发器IRR相应位置1。
- 8259A收到 IR_n 信号后，将其与同时申请中断的信号及正挂起的中断信号比较分析优先级，若其级别最高，则由8259A的INT脚向CPU发出中断请求信号INTR。
- 8088响应中断，连续发出两个INTA信号，第一个INTA信号将在服务寄存器ISR相应位置位，表示正在为响应的那一级服务；同时将中断请求触发器复位。为本级再次中断请求做好准备。第二个INTA信号要求8259A输入中断向量类型码。
- 8088收到8位类型码后，将其乘4作为中断向量的地址。
- 8088屏蔽中断，保护现场，并由取得的中断向量得到IP和CS 的值。
- 以CS和IP值为段地址和偏移量，转入中断服务程序。



4. 中断服务程序的进入

当CPU响应中断，便从中断向量表中相继取出偏移地址和段地址作为中断服务程序的入口地址，而转移去执行中断服务程序。

因此，当用户要编写自己的中断服务程序时，必须将服务程序的入口地址填入系统的中断向量表中。

下面所示程序表示怎样填写中断向量表，以实现用户所定义的60H中断。

```
CLI
MOV  AX, 0
MOV  ES, AX
MOV  DI, 4*60H
MOV  AX, OFFSET INTR
STOSW
MOV  AX, SEG  INTR
STOSW
STI
⋮
```

当运行这个程序，中断向量地址
[DI] [DI+1] [DI+2] [DI+3]
(也即180—183H)中就装入了
中断服务程序INTR的入口地址。



也可以DOS功能调用实现这一功能。

⋮

CLI

PUSH DS

MOV AX, SEG INTR

MOV DS, AX

MOV DX, OFFSET INTR ; 中断服务程序入口地址放入DS:DX中

MOV AH, 25H ; 功能号25H送入AH中

MOV AL, 60H ; 类型码放入AL

INT 21H ; 调用21H 中断

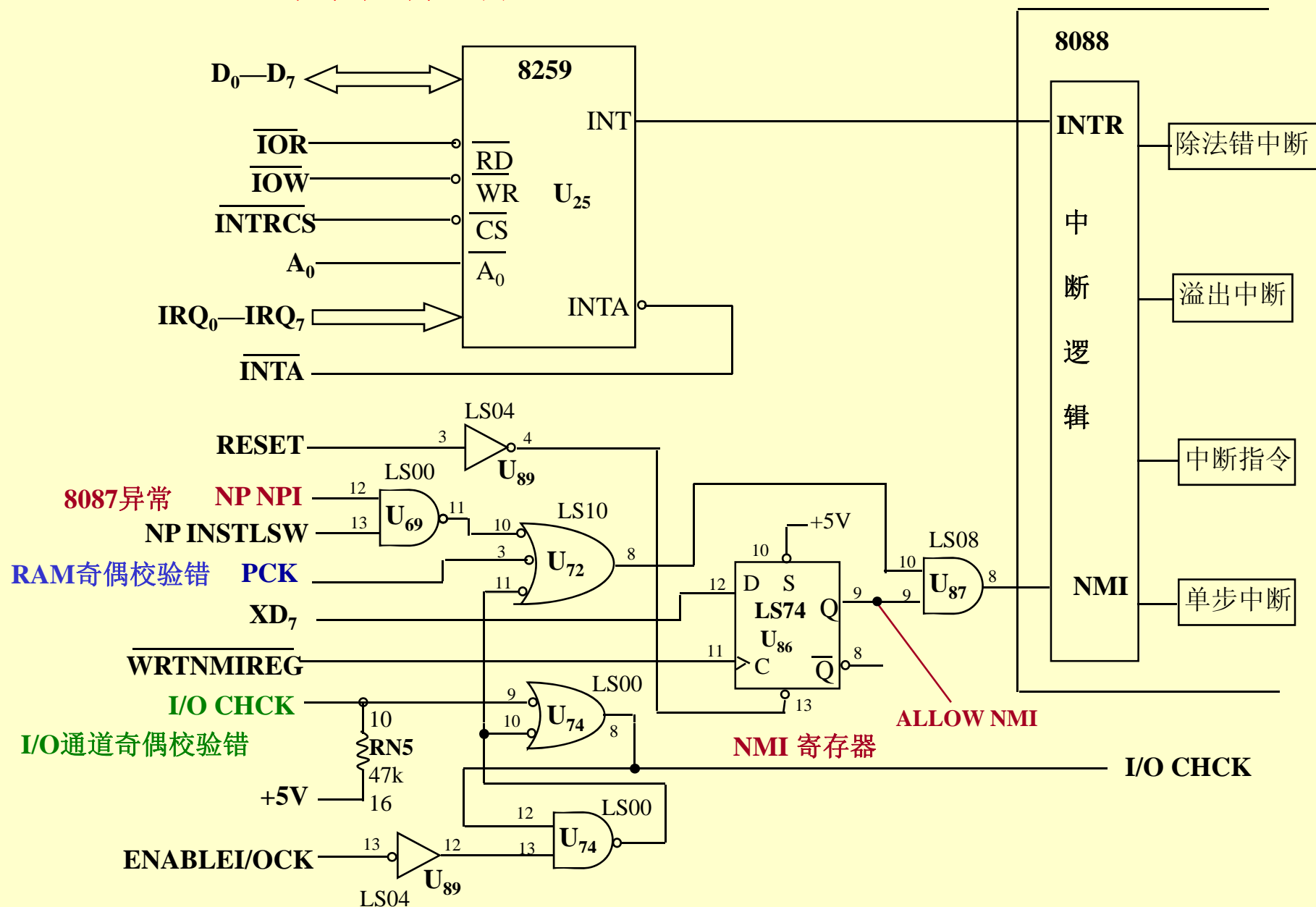
POP DS

STI

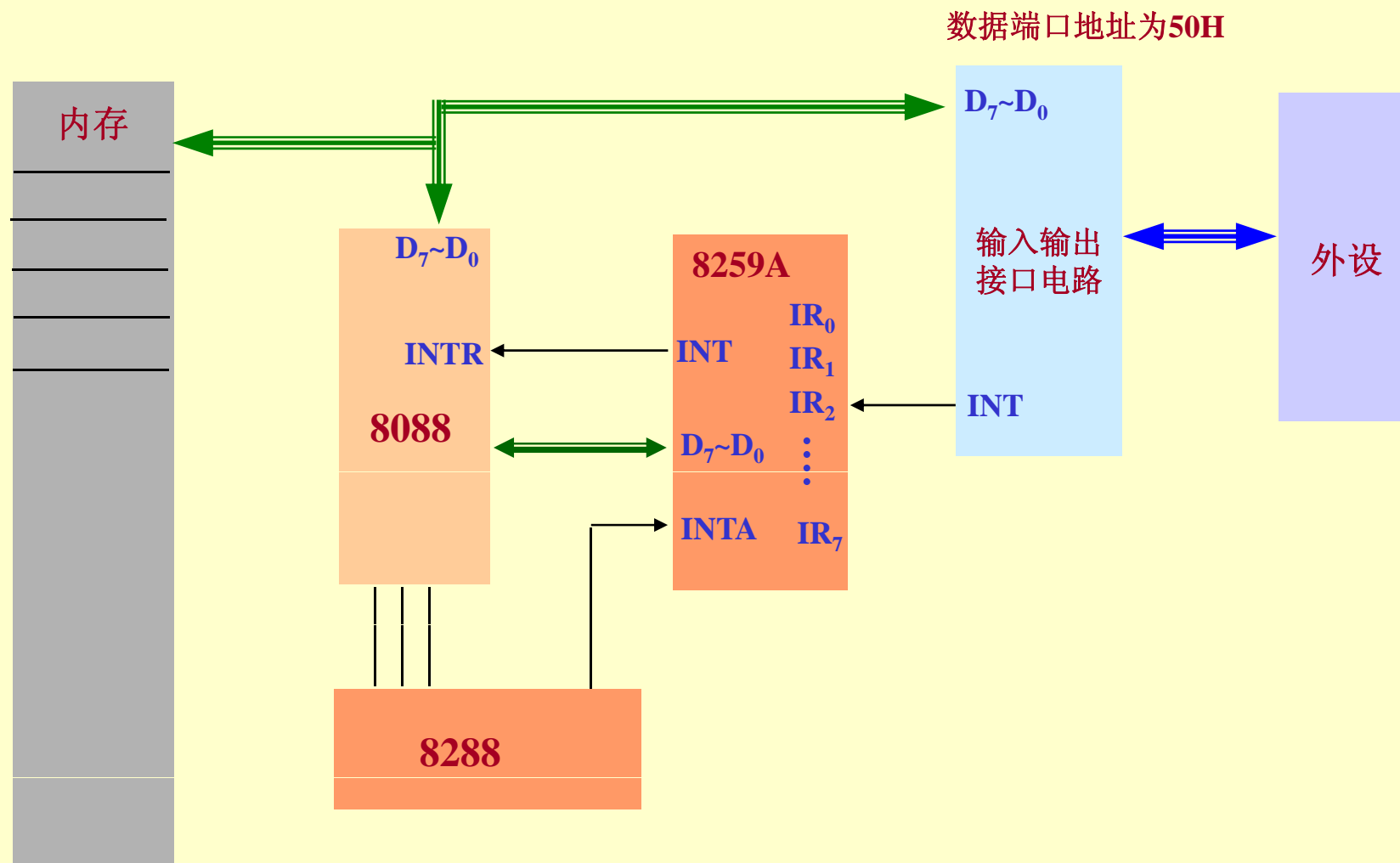
⋮



5. IBM PC/XT中断控制逻辑



四、应用举例 用中断方式将外设数据读入内存



```

Data segment
Buffer db 100 dup(?)
Data ends
Code segment
    assume cs:code,ds:data
Main proc far
Start :push ds
    xor ax, ax
    push ax
    mov ax, data
    mov ds, ax
    lea bx, buffer
    cli
    mov ax, 0
    mov es, ax
    mov di, 28h        ; 0AH*4
    mov ax, offset int_pr
    stosw
    mov ax, seg int_pr
    stosw

```

```

    sti
    :
    :
    :
    ret
main endp
Int_pr proc far
    sti
    push bx
    push ax
    in al, 50h
    mov [bx], al
    inc bx
    pop ax
    pop bx
    mov al, 20h
    out 20h, al
    iret
Int_pr endp
Code ends
    end start

```

若采用硬中断，则在中断服务程序结束时，要送一个EOI中断结束命令给8259A,对ISR复位。

