



東北大學
Northeastern University

汇编语言程序设计

主讲：东北大学计算机学院 刘松冉

第十二章 输入输出与中断

一. 输入输出指令

二. 中断指令

三. 外同步指令和空操作指令

四. CPU与外设间的数据传送

五. 中断系统概述

六. 8086/8088的中断系统

七. 8256A中断控制器及其程序设计

八. 中断程序设计方法

九. 软中断实例：PC机BIOS调用及DOS功能调用

在使用计算机时，需要把原始数据或程序通过输入装置送入计算机中去，运算结果需要通过输出装置在屏幕上显示出来，这就需要编写相应的输入输出程序。



一. 输入输出指令

▶ CPU与外设间的信息交换是通过输入输出指令实现的，8086提供以下两条指令：

1. IN (input byte or input word) 输入指令

指令汇编格式：IN acc, port

操作： $\text{acc} \leftarrow (\text{port})$

将指定端口的内容（字或字节）传送到累加器AX或AL中

受影响的标志位：无

2. OUT (output byte or output word) 输出指令

指令汇编格式：OUT port, acc

操作： $\text{port} \leftarrow \text{acc}$

将累加器AX或AL的内容传送到指定端口

受影响的标志位：无

二. 中断指令

1. INT 中断指令

指令汇编格式：INT n

操作：

$SP \leftarrow SP - 2, (SP, SP+1) \leftarrow F$	；标志寄存器F入栈
$IF \leftarrow 0, TF \leftarrow 0$	；中断标志、陷阱标志清零
$SP \leftarrow SP - 2, (SP, SP+1) \leftarrow CS$	；当前程序的CS入栈
$SP \leftarrow SP - 2, (SP, SP+1) \leftarrow IP$	；当前程序的IP入栈
$CS \leftarrow (n*4+2, n*4+3)$	；中断服务函数CS更新
$IP \leftarrow (n*4, n*4+1)$	；中断服务函数IP更新

受影响的标志位：IF，TF

说明：n称为中断类型号，必须是0~255之间的立即数

二. 中断指令

1. INT 中断指令 — 8086中断向量表

8086/8088 在其内存的前 1K 字节（地址为 00000H~003FFH）建立了一个中断向量表。表中内容分为 256 项，对应于类型号 0~255，每一项占用 4 个字节单元，用来存放相应类型的中断服务程序的入口地址，高两个字节存放服务程序的入口地址的段地址部分。低两个字节存放偏移量部分。其结构如下：

00000H	—	—	—	—	类型0的中断服务	—	—
					程序入口地址		
00003H 00004H	—	—	—	—	类型1的中断服务	—	—
					程序入口地址		
00007H 00008H	—	—	—	—	类型2的中断服务	—	—
					程序入口地址		
0000BH 0000CH					:		
					:		
003FBH 003FCH	—	—	—	—	类型255的中断服务	—	—
					程序入口地址		
003FFH							

二. 中断指令

2. INTO (Interrupt if overflow) 溢出中断指令

指令汇编格式：INTO

操作：如果OF=1，则产生一个中断类型为4的软件中断；否则，不产生任何操作。
即：如果OF=1，则

$SP \leftarrow SP - 2, (SP, SP+1) \leftarrow F$	；标志寄存器F入栈
$IF \leftarrow 0, TF \leftarrow 0$	；中断标志、陷阱标志清零
$SP \leftarrow SP - 2, (SP, SP+1) \leftarrow CS$	；当前程序的CS入栈
$SP \leftarrow SP - 2, (SP, SP+1) \leftarrow IP$	；当前程序的IP入栈
$CS \leftarrow (4*4+2, 4*4+3)$	；中断服务函数CS更新
$IP \leftarrow (4*4, 4*4+1)$	；中断服务函数IP更新

受影响的标志位：IF，TF

说明：INTO指令一般用在算术运算指令后面，在有溢出的情况下，启动一个溢出中断服务程序

二. 中断指令

▶ 3. IRET (Interrupt return) 从中断返回

指令汇编格式：IRET

操作：从堆栈中取出被中断了的程序段的段基址、偏移量和标志寄存器分别送入CS、IP、F。

$$IP \leftarrow (SP, SP+1), SP \leftarrow SP+2$$
$$CS \leftarrow (SP, SP+1), SP \leftarrow SP+2$$
$$F \leftarrow (SP, SP+1), SP \leftarrow SP+2$$

受影响的标志位：所有状态标志位

说明：IRET指令是任何中断服务程序的最后一条要执行的指令，它使CPU从中断服务程序返回被中断程序的断点处继续执行

三. 外同步指令和空操作指令

▶ 8086/8088CPU提供了一种与外部设备同步和使用协处理器（8087或8089）为其完成其本身不能完成的工作的指令。

1. HLT 停机指令

指令汇编格式：HLT

操作： 使CPU处于暂停状态（动态停机）

受影响的标志位：无

说明： CPU执行HLT指令后，就停止执行后续指令，只有当产生中断请求信号（发生中断）或有复位信号（RESET线变成低电平）时，才能退出暂停状态。中断信号产生后，CPU停止执行HLT指令，而去执行中断服务程序，执行IRET后，CPU返回被中断程序的断点处，执行HLT的后续指令

三. 外同步指令和空操作指令

▶ 8086/8088CPU提供了一种与外部设备同步和使用协处理器（8087或8089）为其完成其本身不能完成的工作的指令。

2. WAIT 等待指令

指令汇编格式：WAIT

操作： CPU进入等待状态，检测TEST信号线，当此信号线为低电平时，退出等待状态，执行WAIT的后续指令

受影响的标志位：无

说明： CPU执行WAIT后，只有当TEST信号变成低电平时，才能执行WAIT的后续指令。除此之外，任何情况也不能使CPU停止执行WAIT指令（退出等待状态）

三. 外同步指令和空操作指令

▶ 8086/8088CPU提供了一种与外部设备同步和使用协处理器（8087或8089）为其完成其本身不能完成的工作的指令。

3. ESC 交权指令

指令汇编格式：ESC ext_op, src

操作：如果src是寄存器操作数，则将其内容放置在数据总线上，如果src是存储器操作数，CPU则从存储器中取出操作数放置在数据总线上

受影响的标志位：无

说明：ESC指令称为换码指令，它使协处理器能够从8086/8088的指令流中获得一个操作码和一个操作数。

指令中exp_op为操作码，是一个6位二进制的数，src是3.4节介绍的寻址方式3~8中的任何一种方式。

三. 外同步指令和空操作指令

▶ 8086/8088CPU提供了一种与外部设备同步和使用协处理器（8087或8089）为其完成其本身不能完成的工作的指令。

4. LOCK 锁定指令

指令汇编格式： LOCK

操作：在执行某指令期间保持一个总线锁定信号（LOCK引线变为低电平）

受影响的标志位：无

说明： LOCK指令是一个封锁前缀，它可以加在任何指令的前面，使CPU在执行该指令期间保持一个总线锁定信号。在多处理机系统中用其实现对共享资源的存取控制。

例如，在重复数据串传送指令前面加上一个封锁前缀（LOCK REP MOVS），将阻止系统中其它处理器在重复数据串执行的整个过程中对存储器进行存取。

三. 外同步指令和空操作指令

▶ 8086/8088CPU提供了一种与外部设备同步和使用协处理器（8087或8089）为其完成其本身不能完成的工作的指令。

5. NOP 空指令

指令汇编格式：NOP

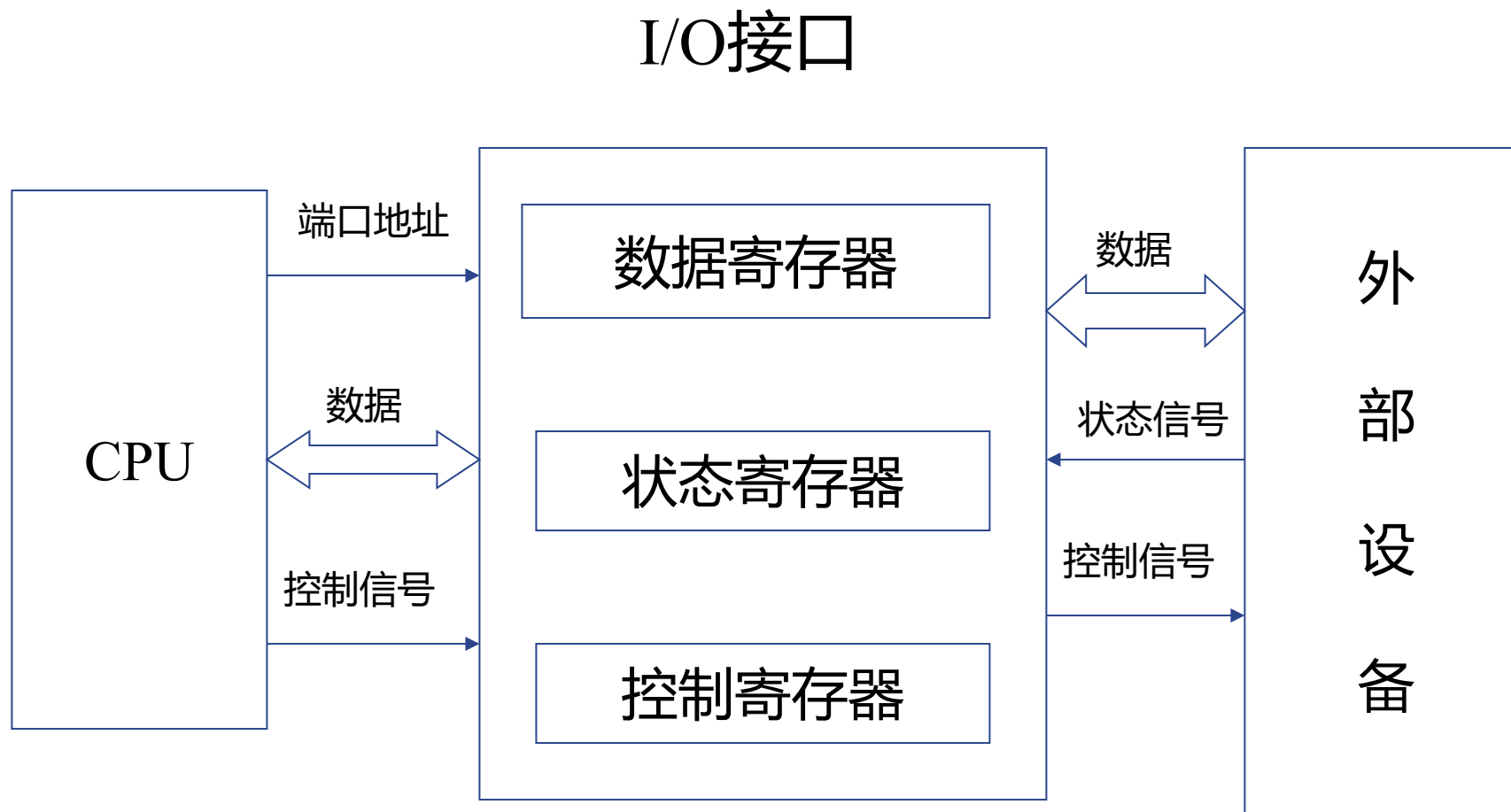
操作：无

受影响的标志位：无

说明：空操作指令NOP使CPU不进行任何操作，但要花费3个时钟周期，然后执行其后续指令，一般用于编制软件延时程序或用纠错程序调试程序时抹掉多余的指令。

四. CPU与外设间的数据传送

1. 输入输出接口



四. CPU与外设间的数据传送

▶ 2. CPU与外设之间的接口信号

- 数据信息 – 这是CPU和外设之间真正要交换的信息
- 状态信息 – 它用来反映外设接口电路或外设的状态，CPU可根据这些状态信息决定对外设的操作或控制
- 控制信息 – 用于控制输入输出设备装置的启动或停止，设备的工作方式等

以上三种不同性质的信息通过不同的端口传送，每个端口都有自己的地址，CPU寻址的是端口地址，而不是笼统的外设。

四. CPU与外设间的数据传送

▶ 3. CPU与外设间的数据传送方式

- **无条件传送** – 当外设的数据传送是定时的，且时间是已知的情况下，CPU定时取入数据，而当CPU去取数据时，数据肯定已准备好；或CPU定时发送数据，外设肯定已准备好接收数据，这时采用无条件传送。无条件传送是最简单的数据传送方式。
- **查询传送方式** – 在外设比CPU工作速度低得多的情况下，在输入时，CPU要查询外设是否就绪（准备好输入数据），若就绪，才能读入数据；在输出时，则不断查询外设是否空闲（准备好接收数据），若空闲，则CPU输出一个数据。

例：从外部设备端口输入并存入内存一组数据，设SR为状态寄存器，其第7位为就绪位，DR为数据寄存器，则程序段如下

```
      :  
      MOV      CX, 0050H  
      MOV      BX, 0200H  
WAIT : IN       AL, SR  
      TEST     AL, 80H  
      JZ       WAIT  
      IN       AL, DR  
      MOV      [BX], AL  
      INC      BX  
      LOOP     WAIT  
      :
```

四. CPU与外设间的数据传送

▶ 3. CPU与外设间的数据传送方式

- **中断传送** – 所谓中断传送方式就是，当CPU与外设交换数据时，无须连续不断地查询外设的状态，而是在需要时，由外设主动地向CPU提出请求，让CPU为其服务。在输入时，当外设数据就绪时，外设向CPU提出中断请求，CPU接到该请求后，暂停当前程序的执行，转去执行相应的中断服务程序，完成所需的数据输入，然后再返回去执行原来中断了的程序；在输出时，当外设空闲时，外设向CPU提出中断请求，CPU接到该请求后，暂停当前程序的执行，转去执行相应的中断服务程序，完成所需的数据输出，然后再返回去执行原来中断了的程序。采用中断传送方式，可以大大提高系统的输入输出效率，节省大量的CPU时间。
- **DMA（直接存储器存取）方式**

五. 中断系统概述

▶ **中断**：当CPU正在执行某程序时，由于外界事件的需要向CPU发出申请，CPU暂停现行程序的执行而转去处理临时发生的事件，处理完后再返回到被中断程序的断点处，继续向下执行，这个过程称为中断。

在中断过程中执行的事件处理程序称为**中断服务程序**。

1. 中断请求与中断源

I/O设备或者事件需要CPU中断处理时，必须向CPU发出中断请求信号。该信号作为CPU的输入，当CPU收到该信号时，可引起中断。

引起中断的原因，或者说发出中断请求信号的源称为**中断源**。通常，中断源有以下几种：

- 一般的输入输出设备。如键盘、打印机、通信接口等。
- 数据通道中断源。如磁盘。
- 实时时钟。
- 故障源。
- 为调试程序而设置的中断源。

五. 中断系统概述

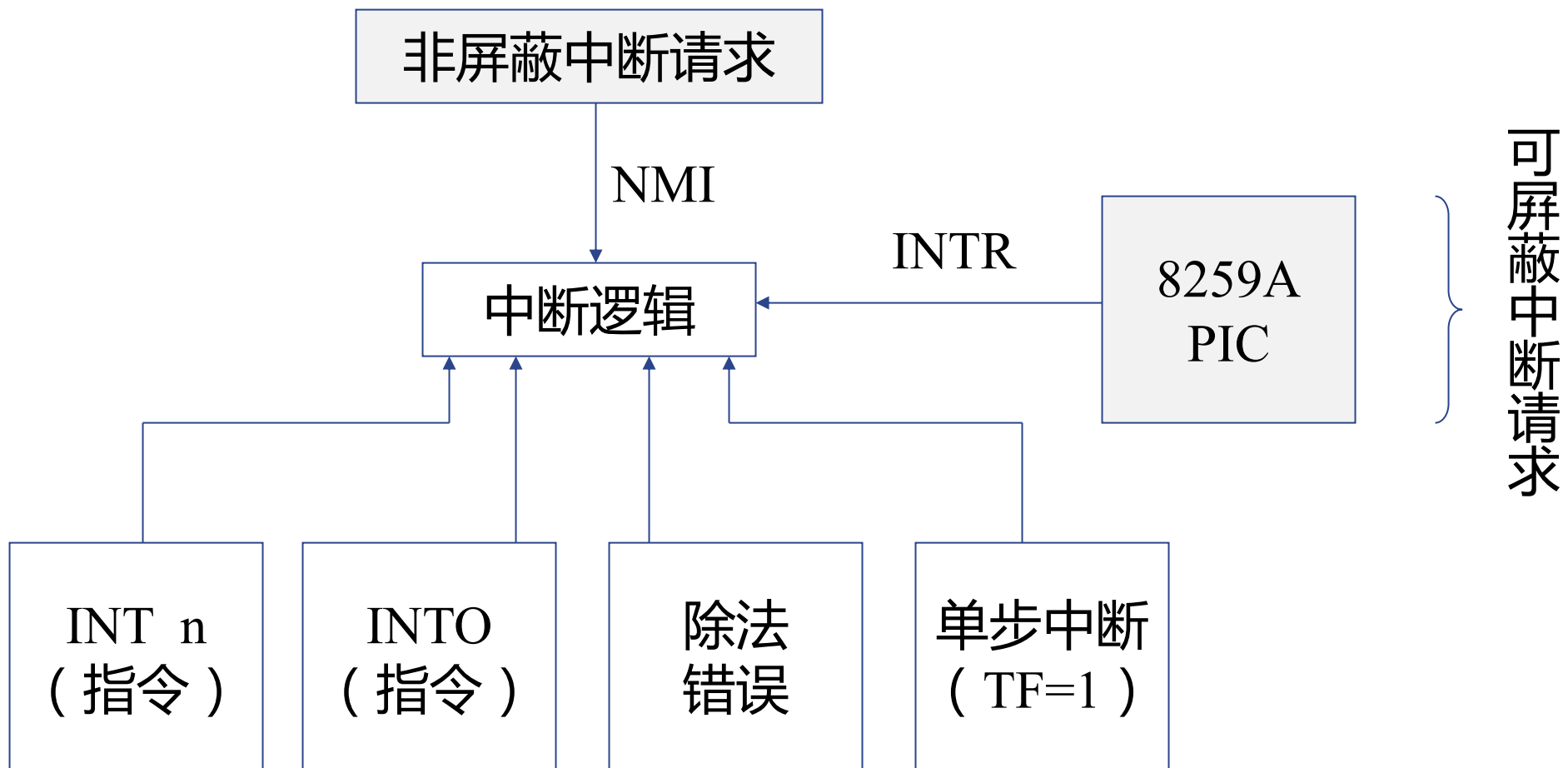
▶ 2. 中断系统的功能

- 实现中断与返回；
- 能实现优先级排队；
- 能实现中断嵌套

3. 中断响应

- CPU响应中断的条件
 - 1) 有中断请求信号；
 - 2) 中断请求没有被屏蔽；
 - 3) CPU的中断是开放的；
 - 4) CPU在现行指令执行完毕时，响应中断。
- 中断响应时及中断响应期间CPU应做的工作
 - 1) 关中断；
 - 2) 保存断点；
 - 3) 保护现场；
 - 4) 给出中断服务程序入口，并转入该服务程序；
 - 5) 恢复现场；
 - 6) 从中断返回。

六. 8086/8088的中断系统



六. 8086/8088的中断系统 — 外部中断

▶ 8086/8088CPU有两种中断请求输入，即非屏蔽中断请求输入和可屏蔽中断请求输入。

1. 可屏蔽中断 — 用户可以控制的中断称为可屏蔽中断

1) 为可屏蔽中断设置的指令

当用户允许CPU响应外部设备发出的中断请求时，IF置1；否则IF清0，使CPU不响应INTR线上的中断请求。可使用指令STI、CLI。

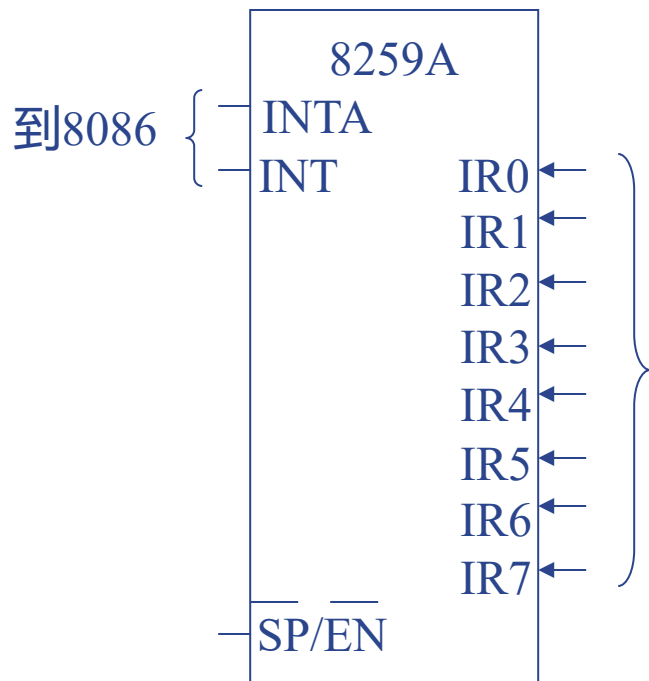
2) 8259A中断控制器

用来对可屏蔽中断源进行管理，其主要任务是接收外部设备的中断请求，然后根据优先级的高低和预先规定的排优规则决定哪个设备能够申请中断，由8259A向CPU发中断请求信号。当CPU响应此请求并给出中断应答之后，8259A还要把一个中断方式码通过数据总线交给CPU，供CPU识别是谁发出的中断请求。

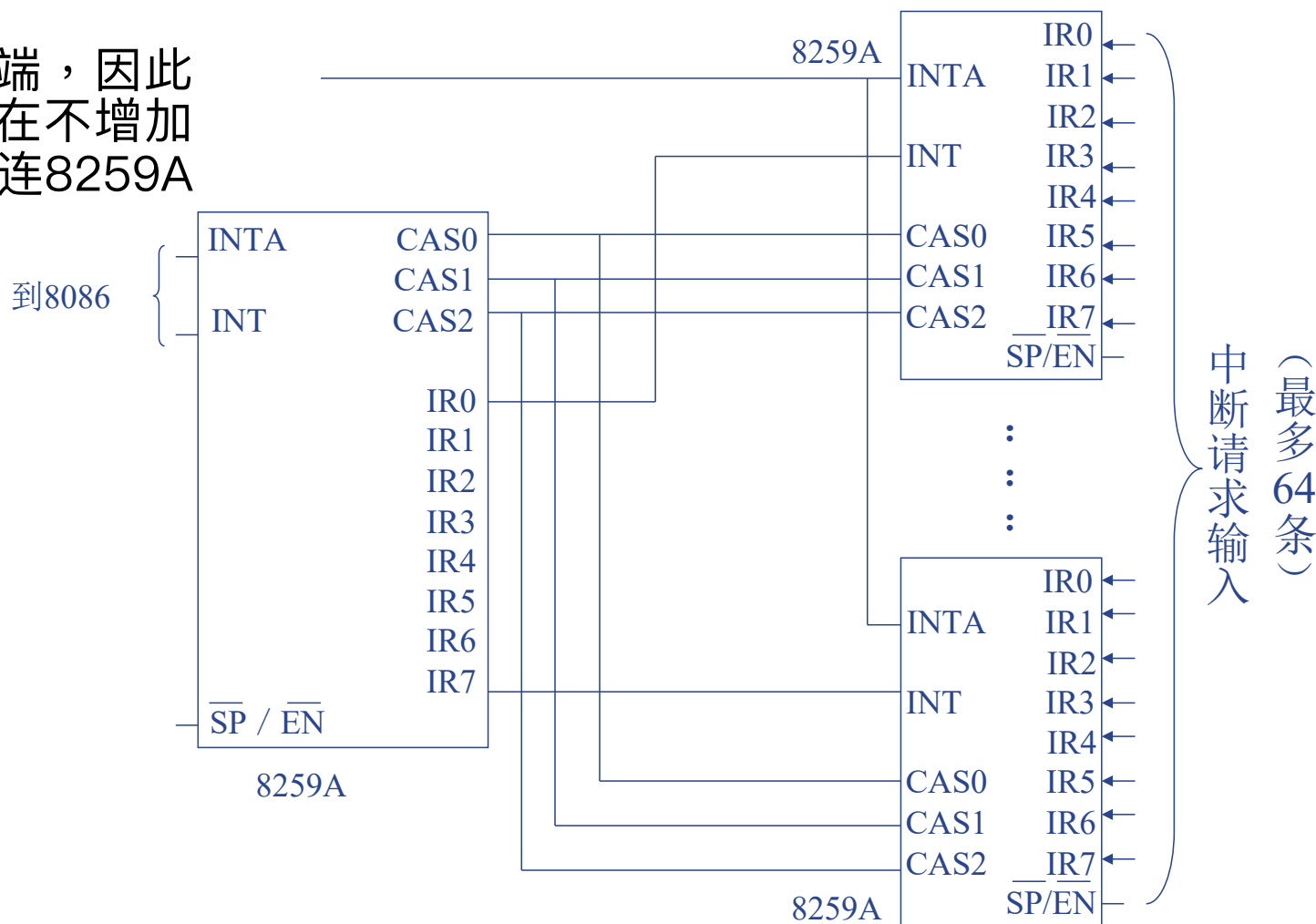
六. 8086/8088的中断系统 — 外部中断

2) 8259A中断控制器

每个8259A有8个中断请求输入端，因此单个8259A可以处理8级中断。在不增加任何硬件线路的情况下，通过级连8259A最多可以管理64级中断。



每个8259A有8个中断请求输入端，因此单个8259A可以处理8级中断。在不增加任何硬件线路的情况下，通过级连8259A最多可以管理64级中断。



六. 8086/8088的中断系统 — 外部中断



3) CPU对可屏蔽中断的响应

当外设通过8259A向CPU提出申请，且CPU的IF=1时，CPU就挂起正在处理的任务，进行中断响应和处理。整个过程如下：

- ①响应中断；
 - ②读取8259A送来的方式码；
 - ③将标志寄存器F的内容压栈；
 - ④清除中断允许标志位IF和陷阱标志位TF；
 - ⑤将代码段寄存器CS和指令指针IP的内容压栈；
 - ⑥根据方式码找到服务程序入口，且调用服务程序；
 - ⑦执行用户中断服务程序；
 - ⑧将保存在栈中的IP和CS的内容从栈中弹回到IP和CS；
 - ⑨将保存在栈中的标志寄存器F的内容从栈中弹回到标志寄存器F；
 - ⑩从中断返回。
- 以上过程的①~⑦由**硬件自动完成**，⑧~⑩执行IRET指令实现。

六. 8086/8088的中断系统 — 外部中断

▶ 8086/8088CPU有两种中断请求输入，即非屏蔽中断请求输入和可屏蔽中断请求输入。

2. 非可屏蔽中断 — 这类中断不能被屏蔽，即使在关中断的情况下（IF=0），在执行完当前指令后，CPU也会立即响应。非屏蔽中断的方式码规定为2。响应非屏蔽中断的过程如：

- ①将标志寄存器F的内容压栈；
- ②清除中断允许标志位IF和陷阱标志位TF；
- ③将代码段寄存器CS和指令指针IP的内容压栈；
- ④根据方式码（=2）找到服务程序入口，且调用服务程序；
- ⑤执行非屏蔽中断服务程序；
- ⑥将保存在栈中的IP和CS的内容从栈中弹回到IP和CS；
- ⑦将保存在栈中的标志寄存器F的内容从栈中弹回到标志寄存器F；
- ⑧从中断返回。

以上过程的①~⑤由硬件自动完成，⑥~⑧ 执行IRET指令实现。

六. 8086/8088的中断系统 — 内部中断

► 为了解决程序运行过程中发生的一些意外情况和调试程序的需要，8086/8088设置了一些内部中断，共有以下五类：

- **方式0 – 除法错中断**

- > 在执行DIV/IDIV时，若商超过机器所能表示的最大值，CPU立即产生一个方式0中断，除法中断不能被屏蔽且是除法指令的一部分。

- **方式1 – 单步中断**

- > 8086/8088CPU中有一个陷阱标志TF。如果TF置位，在每执行完一条指令后，CPU就自动产生一个方式1中断。我们称CPU每执行完一条指令就进入中断的工作方式称为单步工作方式。

- **方式3 – 断点中断**

- > 8086/8088CPU中有一条用来设置程序断点的单字节指令。

- **方式4 – 溢出中断**

- > 在程序执行过程中如果溢出标志OF被置位或执行INTO指令以后就会立即产生一个方式4中断。

- **用户定义的软件中断：利用INT指令产生的**

- > 用户可以通过INT指令定义自己的软中断。

六. 8086/8088的中断系统 — 内部中断

▶ 所有的内部中断都具有下述特点：

- 1) 中断向量码或者包含在指令中，或者是预定的；
- 2) 除单步中断外，内部中断都无法禁止；
- 3) 除单步中断外，任何内部中断的优先级都比任何外部中断的高。

表12.1

中断优先级

中 断	优先级
除法错、INT nn、INTO	最高
NMI	↓
INTR	
单步中断	
	最低

六. 8086/8088的中断系统 — 中断向量表

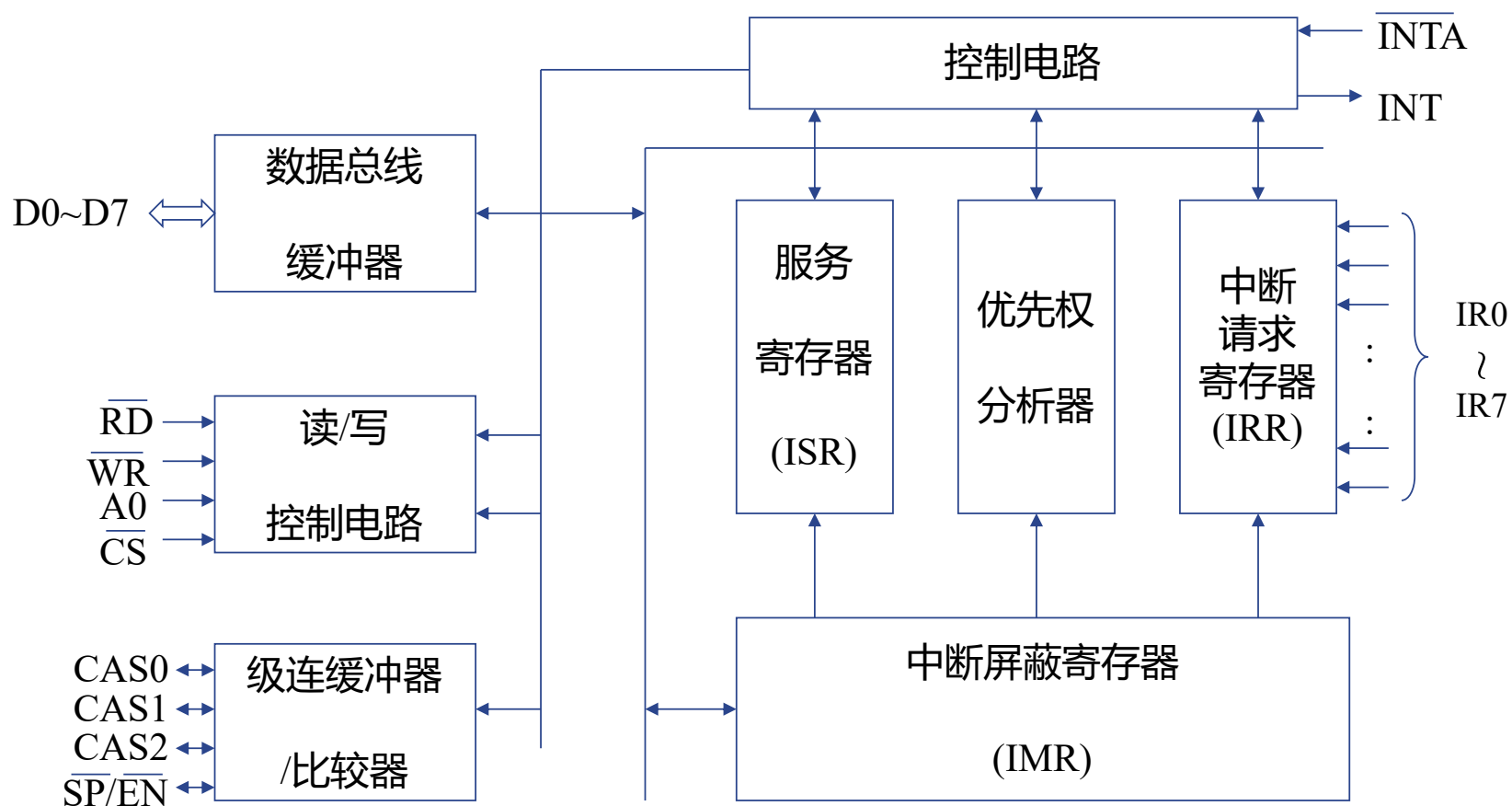
▶ 中断向量表的前5项已被系统占用，系统又保留了27项以备将来扩充用，余下的224项可为用户使用。

00000H	除法错中断服务
	程序入口地址
00003H 00004H	单步中断服务
	程序入口地址
00007H 00008H	非屏蔽中断服务
	程序入口地址
0000BH 0000CH	断点中断服务
	程序入口地址
0000FH 00010H	溢出中断服务
	程序入口地址
00013H	

七. 8259A中断控制器及其程序设计

1. 8259A的结构

- 中断请求寄存器 (IRR) 和服务寄存器 (ISR)
- 优先级分析器
- 中断屏蔽寄存器 (IMR)
- 中断输出信号 (INT)
- 中断响应信号 (INTA)
- 级连缓冲器/比较器



七. 8259A中断控制器及其程序设计

▶ 2. 8259A的工作过程

- 1) 中断源在中断请求输入端IR0~IR7上产生中断请求。
- 2) 中断请求被锁存在IRR中，并与IMR的内容相“与”，结果送给优先级分析器。
- 3) 优先级分析器检出优先级最高的中断请求位，设置ISR中的对应位。
- 4) 控制电路接受中断请求，向CPU输出INT信号。
- 5) CPU接受INT信号，进入连续两次的INTA周期。
- 6) 如果8259A是作为主控的中断控制器设置的，则在第一个INTA周期把级连地址从CAS0~CAS2送出。
- 7) 若8259A是单独使用或是由CAS0~CAS2选择的从属控制器，就在第二个INTA周期将一个中断向量码输出到数据总线。
- 8) 8086/8088读取该中断向量码，转移到相应的中断处理程序。
- 9) 中断结束是通过向8259A送一条EOI命令，使ISR复位来实现的。

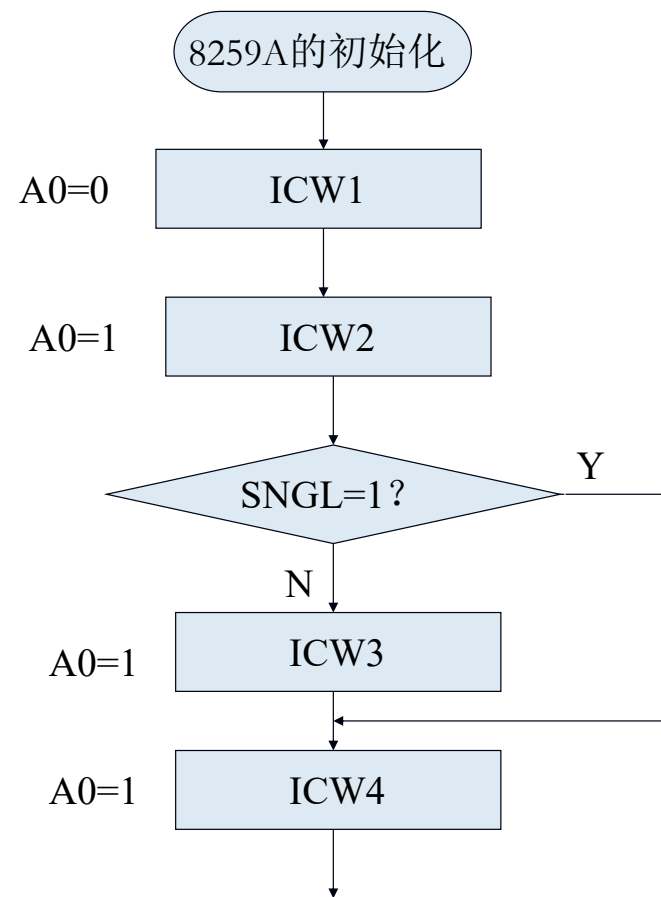
七. 8259A中断控制器及其程序设计

3. 8259A的程序设计

8259A是可编程的中断控制器，可根据不同的需要通过软件编程配置其工作方式。8259A的程序设计是通过使用两种命令字来完成的。这两种命令字由4个初始化命令字和3个操作命令字组成。8259A只有两个地址（奇地址A0=1和偶地址A0=0），操作是要严格按顺序执行。

1) 初始化命令字ICW

ICW是初始设置时使用的命令。它包括ICW1~ICW4四种命令，用于设置8259A的工作方式、中断向量的高5位等。初始化命令字的设置过程必须按一定的顺序完成。



七. 8259A中断控制器及其程序设计

3. 8259A的程序设计

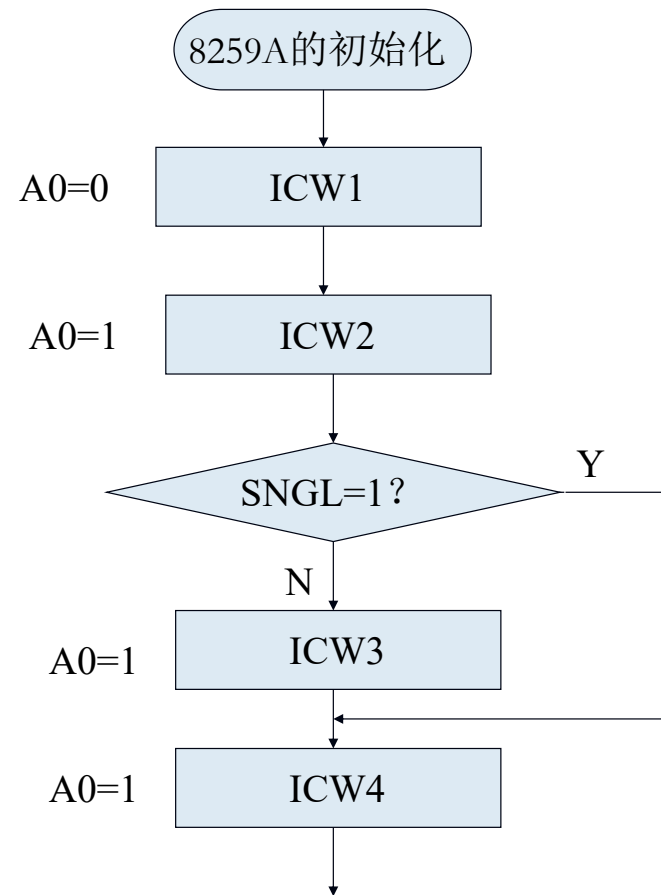
1) 初始化命令字ICW

	D7	D6	D5	D4	D3	D2	D1	D0
ICW1	×	×	×	1	LTIM	×	SNGL	1

D4位为ICW1的特征位，必须为1。无论8259A处于什么状态，只要向8259A的偶地址写入一个第四位为1的命令字，8259A就认为是ICW1，从而进入到初始化操作，并将其后写入到奇地址的1~3字节的命令字作为ICW2~ICW4。

LTIM：中断检测位，1=电平检测，0=边缘检测

SNGL：1=单独使用，0=级连使用



七. 8259A中断控制器及其程序设计

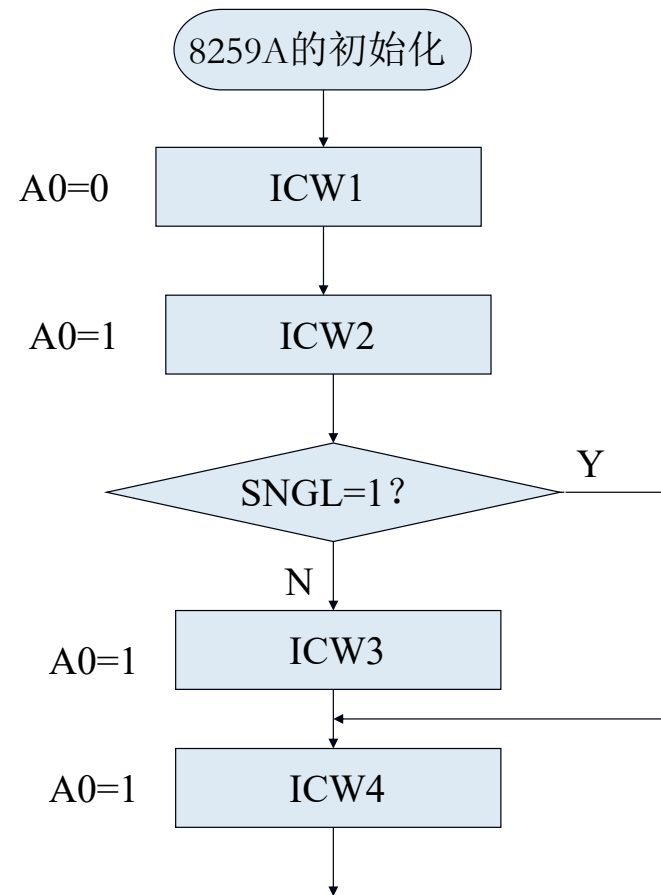
3. 8259A的程序设计

1) 初始化命令字ICW



其高五位状态T7~T3即是中断向量码的高五位，低三位状态可为任意值，不使用。

一旦ICW2设定后，8259A的8个输入端所管理的中断源的中断方式码即可确定下来。某一IR端的中断方式码的确定方法如下：中断方式码的高5位即是ICW2的高5位，而低3位则是该IR端的二进制编码值。
例：初始化8259A时，设定的ICW2=15H，则IR3端的中断方式码则为13H。



七. 8259A中断控制器及其程序设计

3. 8259A的程序设计

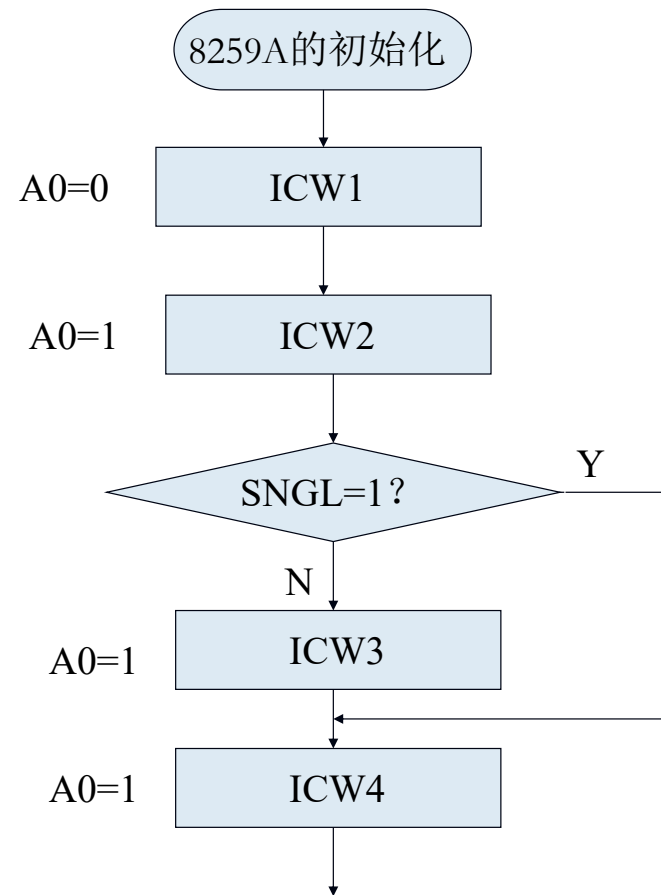
1) 初始化命令字ICW

	D7	D6	D5	D4	D3	D2	D1	D0
ICW3	S7	S6	S5	S4	S3	S2/ID2	S1/ID1	S0/ID0

若8259A单独使用，则不需设置该命令字，只有在8259A级连使用时，才需要该命令字。此时，若8259A作主控制器使用，则S7~S0的某一位为1，表示该8259A的相应的IR端连接有一个从属8259A，若S7~S0的某一位状态为0，则表示对应的IR端未连从属8259A；

若8259A是作从属控制器使用的，则ICW3的高5位为0，而低3位则是该从属8259A的识别地址，即它连接到主8259A的IR端的二进制编码值。

例：假设系统中使用了两片8259A管理15级中断，从属控制器连接在主控制器的IR4端，则主控制器的ICW3应设置为10H，而从属8259A的ICW3应设置为04H。



七. 8259A中断控制器及其程序设计

3. 8259A的程序设计

1) 初始化命令字ICW

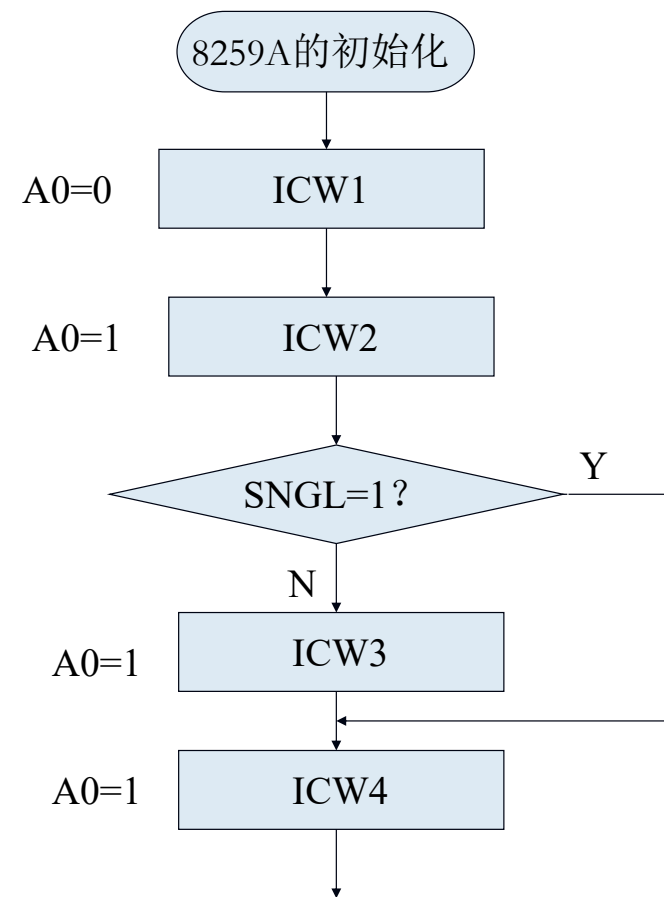
	D7	D6	D5	D4	D3	D2	D1	D0
ICW4	0	0	0	SFNM	BUF	M/S	AEOI	1

SFNM：1=特殊完全嵌套方式，0=完全嵌套方式。

BUF：1=缓冲方式，0=非缓冲方式。

M/S：若BUF=0，则该位可忽略。若系统中只有一个8259A，则该位为1；否则，该位为1表示主控制器，为0表示从属控制器。

AEOI：1=自动EOI方式，0=EOI方式。



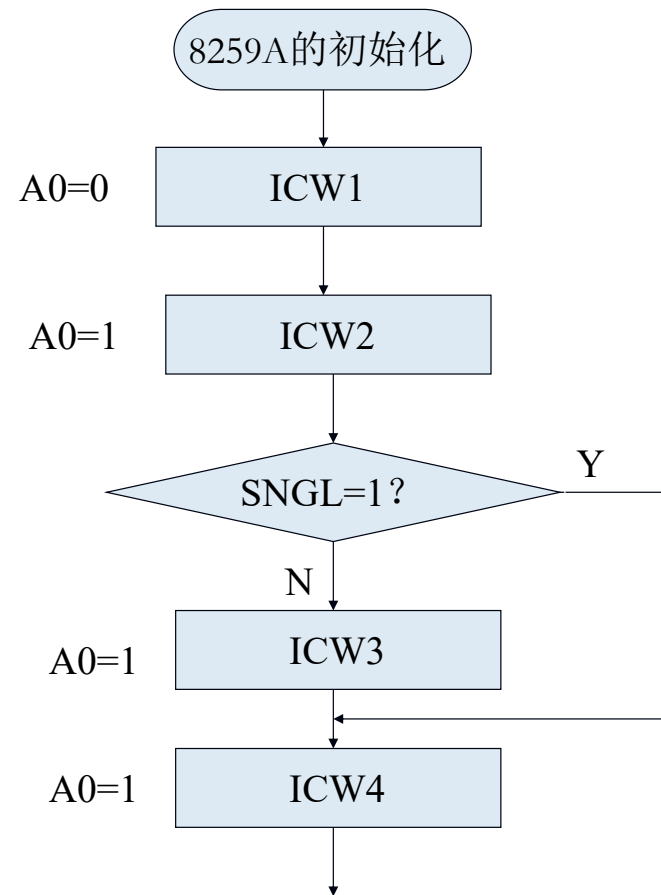
七. 8259A中断控制器及其程序设计

3. 8259A的程序设计

1) 初始化命令字ICW — 例1

在IBM PC、IBM PC/XT系统中，使用一片8259A管理中斷，系统分配给8259A的地址是20H和21H。在系统初启时，已经对8259A进行了初始化，程序段如下：

1	MOV	AL, 13H	; 8259A单独使用，边缘检测，不需要ICW3
2	OUT	20H, AL	
3	MOV	AL, 08H	; IR0~IR7八个输入端的中断方式码分别为08H~0FH
4	OUT	21H, AL	
5	MOV	AL, 09H	; 使用缓冲方式和EOI方式
6	OUT	21H, AL	



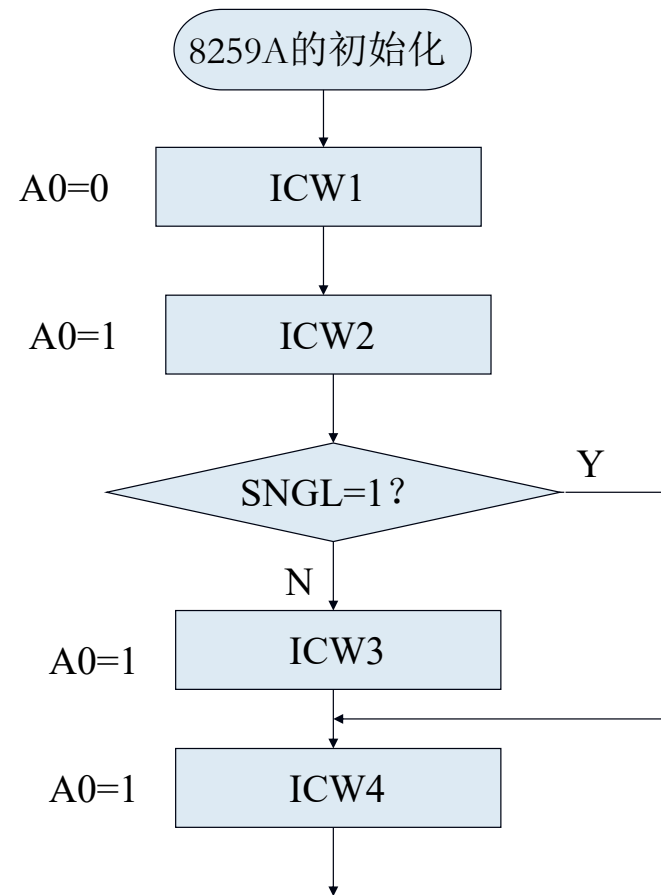
七. 8259A中断控制器及其程序设计

3. 8259A的程序设计

1) 初始化命令字ICW — 例2

在IBM PC/AT系统中，使用两片8259A管理中斷，从属8259A连接在主8259A的IR2端，系统分配给主8259A的地址是20H和21H，分配给从8259A的地址是0A0H和0A1H。在系统初启时，已经对两片8259A进行了初始化，程序段如下：

```
8      ;初始化主8259A
9      MOV     AL, 11H      ;8259A级连使用,边缘检测,需要ICW3
10     OUT     20H, AL
11     MOV     AL, 08H      ;IR0~IR7八个输入端的中断方式码分别为08H~0FH
12     OUT     21H, AL
13     MOV     AL, 04H      ;在IR2端接有一从属8259A
14     OUT     21H, AL
15     MOV     AL, 11H      ;使用特殊完全嵌套方式、非缓冲方式和EOI方式
16     OUT     21H, AL
```



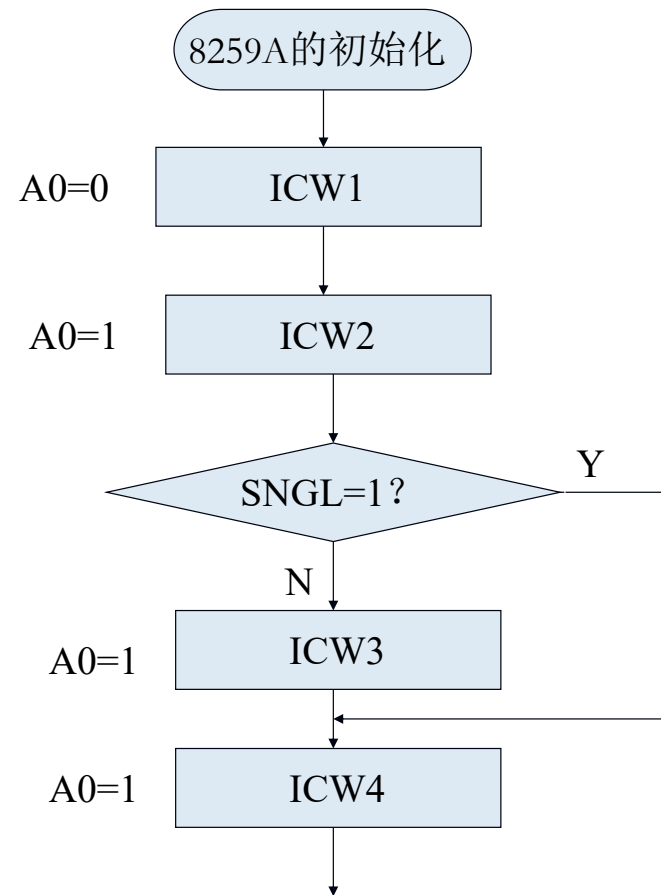
七. 8259A中断控制器及其程序设计

3. 8259A的程序设计

1) 初始化命令字ICW — 例2

在IBM PC/AT系统中，使用两片8259A管理中断，从属8259A连接在主8259A的IR2端，系统分配给主8259A的地址是20H和21H，分配给从8259A的地址是0A0H和0A1H。在系统初启时，已经对两片8259A进行了初始化，程序段如下：

```
18      ;初始化从8259A
19      MOV     AL, 11H      ;8259A级连使用,边缘检测,需要ICW3
20      OUT     0A0H, AL
21      MOV     AL, 70H      ;IR0~IR7八个输入端的中断方式码分别为70H~77H
22      OUT     0A1H, AL
23      MOV     AL, 02H      ;从属8259A连接在主8259A的IR2端
24      OUT     0A1H, AL
25      MOV     AL, 01H      ;使用非缓冲方式和EOI方式
26      OUT     0A1H, AL
```



七. 8259A中断控制器及其程序设计

3. 8259A的程序设计

2) 操作命令字OCW

OCW是在操作过程中给出的命令字，初始设定结束后的命令都可以看作是OCW命令。

	D7	D6	D5	D4	D3	D2	D1	D0
OCW1	M7	M6	M5	M4	M3	M2	M1	M0

OCW1：中断屏蔽命令字

中断屏蔽命令字是中断屏蔽寄存器置位、复位的命令。如果某条中断请求输入线对应的屏蔽位为1，该条线上的中断请求就被屏蔽了。通过8259A的奇地址输出给8259A的一个字节就是OCW1。例如：

```
MOV    AL, 13H
```

```
OUT    21H, AL
```

执行这两条指令就把IR0、IR1和IR4上的中断请求封锁了。

七. 8259A中断控制器及其程序设计

3. 8259A的程序设计

2) 操作命令字OCW

OCW是在操作过程中给出的命令字，初始设定结束后的命令都可以看作是OCW命令。

	D7	D6	D5	D4	D3	D2	D1	D0
OCW2	R	SL	EOI	0	0	L2	L1	L0

OCW2：中断结束命令（EOI命令）

通过8259A的偶地址输出给8259A的命令被解释为ICW1、OCW2或OCW3。这三个命令是靠命令字的第3、4位区分的。

D4、D3=00是OCW2，D4、D3=01是OCW3，D4=1是ICW1。

EOI命令用于指定复位的ISR位及改变中断优先级。

```
MOV AL,20H
```

```
OUT 20H,AL
```

八. 中断程序设计方法

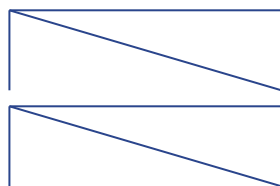
1. 主程序设计

- 1) CPU部分 – 包括设置堆栈指针（SS和SP），设置中断向量和开放中断。
- 2) 中断控制器8259A部分 – 选择8259A的工作方式。
- 3) 通用外设接口部分 – 选择接口的工作方式和设置接口的中断开放位等。



主程序

中断服务程序



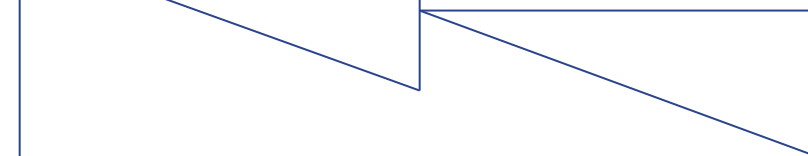
单级中断服务程序结构



主程序

中断服务程序1

中断服务程序2



多级中断服务程序结构

九. 软中断实例：PC机BIOS调用及DOS功能调用

1. BIOS调用

- 1) **概述** – 在内存的0FE000H开始的8KB ROM中存放着基本输入输出系统BIOS (Basic I/O System)。BIOS在汇编语言级上向用户程序和系统程序提供PC机的一些主要外设的设备控制功能，包括开机自检，显示器、通信接口、键盘、打印机的字符传送，图形发生，磁盘的读写等，这些操作使得用户无需考虑外设的I/O地址等细节。BIOS是用户与硬件之间的第一隔离层，它使外设的更改或增减均对用户透明。
- 2) **BIOS的用法** – BIOS功能调用是利用INT指令实现的。关于BIOS各子程序的功能、入口参数和出口参数详在教材的附录C中有详细说明。只要设置好功能调用的入口参数，即可通过一条INT指令调用所需的子程序从而完成相应的操作。

九. 软中断实例：PC机BIOS调用及DOS功能调用

1. BIOS调用

- 1) **例子** – 利用BIOS功能，实现从键盘上接收若干字符并将其ASCII码与键盘扫描码以十六进制形式显示出来，直到用户按下[Ctrl_C]为止。[Ctrl_C]的ASCII码为03H。

BIOS的键盘输入功能为INT 16H，显示功能为INT 10H。

- 从键盘读入按键的ASCII码和扫描码

```
28  MOV    AH, 0           ;功能号为0
29  INT     16H            ;执行后，AH是按键的扫描码，AL是按键的ASCII码
```

- 在屏幕上显示字符

```
31  MOV    AH, 0EH        ;功能号为14
32  MOV    AL, 'A'        ;欲显示的字符
33  INT     10H
```

九. 软中断实例：PC机BIOS调用及DOS功能调用

1. BIOS调用

1) 例子

```
1  SSEG      SEGMENT STACK
2  DB 80H DUP (?)
3  SSEG      ENDS
4  CSEG      SEGMENT
5  ASSUME CS:CSEG, SS:SSEG
6  ; 在当前光标处显示AL中的字符
7  DISASC    PROC
8  PUSH     AX
9  MOV      AH, 14
10 INT      10H
11 POP      AX
12 RET
13 DISASC    ENDP
```

```
27 BIOS      PROC      FAR
28           PUSH      DS
29           XOR        AX, AX
30           PUSH      AX
31 AGAIN:    MOV        AH, 0          ;INT 16H的0号功能
32           INT        16H
33           PUSH      AX
34           CALL       DISAL
35           MOV        AL, '-'
36           CALL       DISASC
37           MOV        AL, AH
38           CALL       DISAL
39           MOV        AL, ' '
40           CALL       DISASC
41           POP        AX
42           CMP        AL, 03H
43           JNZ        AGAIN
44           RET
45 BIOS      ENDP
```

九. 软中断实例：PC机BIOS调用及DOS功能调用

2. BIOS调用DOS中断和功能调用

1) DOS中断

编号从20H到3FH的软中断为DOS操作系统使用，称DOS中断。

中断类型	功能	入口参数	出口参数
INT 20H	程序正常退出	CS=程序段前缀段地址	
INT 21H	系统功能调用	AH=调用号 功能调用入口参数	功能调用出口参数
INT 22H	结束退出		
INT 23H	Ctrl-Break退出		
INT 24H	出错退出		
INT 25H	读盘	AL=盘号 CX=扇区数 DX=起始逻辑扇区号 DS:BX=缓冲区首地址	CF=1 表示出错 CF=0表示正常
INT 26H	写盘	同上	同上
INT 27H	驻留退出	DX=驻留长度	

九. 软中断实例：PC机BIOS调用及DOS功能调用

2. BIOS调用DOS中断和功能调用

2) DOS功能调用

DOS功能调用是指使用INT 21H软件中断时所完成的各种操作。调用时把功能号放在AH寄存器中，功能调用所需其它参数放在指定的寄存器中，然后发出INT 21H，即可完成预定的操作。

支持文件或设备的新增加的功能调用使用一个所谓的“文件号”作为文件的表示符（称为文件句柄）。当用这些功能调用建立或打开一个文件或设备时，在AX中返回一个16位的二进制数，这就是该文件的文件号，再引用该文件时，就利用此文件号。

下面一些设备文件是DOS预先定义的，且每一设备文件都有固定的文件号。这些设备文件用户程序也可以使用，且在使用之前不需要将它们打开，而直接引用它们的文件号即可。

0000 标准输出设备，通常为显示器，可重定向到其它设备或文件

0001 标准输入设备，通常为键盘，可重定向到其它设备或文件

0002 标准错误输出设备

0003 标准辅助输入输出设备，即异步通信接口

0004 标准打印机设备

九. 软中断实例：PC机BIOS调用及DOS功能调用

2. BIOS调用DOS中断和功能调用

3) DOS功能调用使用举例

利用DOS的功能调用，将一个文本文件的内容在屏幕上显示出来。

打开文件：

DS：DX存放文件名的首地址（文件名为ASCIIZ串）

AL中存放对文件的操作方式：0=读，1=写，2=读/写

AH中存放功能号3DH

执行INT 21H，如果打开文件成功，则在AX中返回一个文件号。

九. 软中断实例：PC机BIOS调用及DOS功能调用

2. BIOS调用DOS中断和功能调用

3) DOS功能调用使用举例

利用DOS的功能调用，将一个文本文件的内容在屏幕上显示出来。

读文件：

DS:DX存放保存存放读取的文件内容的缓冲区的首地址

BX中存放文件号

CX中存放一次读取的字节数

AH中存放功能号3FH

执行INT 21H，如果读文件成功，则在AX中返回实际读出的字节数。

九. 软中断实例：PC机BIOS调用及DOS功能调用

2. BIOS调用DOS中断和功能调用

3) DOS功能调用使用举例

利用DOS的功能调用，将一个文本文件的内容在屏幕上显示出来。

关闭文件：

BX中存放文件号

AH中存放功能号3EH

如果对文件的操作失败，功能调用则将CF置1，返回应用程序。

九. 软中断实例：PC机BIOS调用及DOS功能调用

2. BIOS调用DOS中断和功能调用

3) DOS功能调用使用举例

示例：

```
1  SSEG      SEGMENT STACK
2          DB  80H DUP (0)
3  SSEG      ENDS
4  DSEG      SEGMENT
5  ASKFN     DB  'File Name:$'
6  FNAME     DB  14,0,14 DUP (?)
7  ERRMSG    DB  'FILE ERROR!$'
8  BUFF      DB  512 DUP (?)
9  DSEG      ENDS
10 CSEG      SEGMENT
11          ASSUME CS:CSEG, DS:DSEG, SS:SSEG
12
13 FILE      PROC    FAR
14          PUSH    DS
15          XOR     AX,AX
16          PUSH    AX
17          MOV     AX,DSEG
18          MOV     DS,AX
19          LEA     DX,ASKFN
20          MOV     AH,9
21          INT     21H
```

九. 软中断实例：PC机BIOS调用及DOS功能调用

2. BIOS调用DOS中断和功能调用

3) DOS功能调用使用举例

示例：

```
1  SSEG  SEGMENT STACK
2      DB  80H  DUP (0)
3  SSEG  ENDS
4  DSEG  SEGMENT
5  ASKFN  DB  'File Name:$'
6  FNAME  DB  14,0,14  DUP (?)
7  ERRMSG DB  'FILE ERROR!$'
8  BUFF   DB  512  DUP (?)
9  DSEG  ENDS
10 CSEG  SEGMENT
11      ASSUME CS:CSEG, DS:DSEG, SS:SSEG
12
13 FILE  PROC  FAR
14      PUSH  DS
15      XOR   AX,AX
16      PUSH  AX
17      MOV   AX,DSEG
18      MOV   DS,AX
19      LEA   DX,ASKFN
20      MOV   AH,9
21      INT   21H
```

```
22      LEA   DX,FNAME
23      MOV   AH,0AH
24      INT   21H
25      MOV   DL,0AH
26      MOV   AH,2
27      INT   21H
28      MOV   BL,FNAME+1
29      MOV   BH,0
30      MOV   [BX+FNAME+2],BH
31      LEA   DX,FNAME+2
32      MOV   AX,3D00H
33      INT   21H
34      JC    EXIT
35      MOV   BX,AX
36 AGAIN: MOV   CX,512
37      LEA   DX,BUFF
38      MOV   AH,3FH
39      INT   21H
40      JC    EXIT
41      PUSH  AX
42      MOV   CX,AX
43      MOV   SI,OFFSET  BUFF
44 PUTC:  MOV   DL,[SI]
45      INC   SI
46      MOV   AH,2
47      INT   21H
48      LOOP PUTC
```

FNAME

0EH
00H

九. 软中断实例：PC机BIOS调用及DOS功能调用

2. BIOS调用DOS中断和功能调用

3) DOS功能调用使用举例

示例：

```
1  SSEG  SEGMENT STACK
2      DB  80H  DUP (0)
3  SSEG  ENDS
4  DSEG  SEGMENT
5  ASKFN  DB  'File Name:$'
6  FNAME  DB  14,0,14  DUP (?)
7  ERRMSG DB  'FILE ERROR!$'
8  BUFF   DB  512  DUP (?)
9  DSEG  ENDS
10 CSEG  SEGMENT
11      ASSUME CS:CSEG, DS:DSEG, SS:SSEG
12
13 FILE  PROC  FAR
14      PUSH  DS
15      XOR   AX,AX
16      PUSH  AX
17      MOV   AX,DSEG
18      MOV   DS,AX
19      LEA   DX,ASKFN
20      MOV   AH,9
21      INT   21H
```

```
22      LEA   DX,FNAME
23      MOV   AH,0AH
24      INT   21H
25      MOV   DL,0AH
26      MOV   AH,2
27      INT   21H
28      MOV   BL,FNAME+1
29      MOV   BH,0
30      MOV   [BX+FNAME+2],BH
31      LEA   DX,FNAME+2
32      MOV   AX,3D00H
33      INT   21H
34      JC    EXIT
35      MOV   BX,AX
36 AGAIN: MOV   CX,512
37      LEA   DX,BUFF
38      MOV   AH,3FH
39      INT   21H
40      JC    EXIT
41      PUSH  AX
42      MOV   CX,AX
43      MOV   SI,OFFSET  BUFF
44 PUTC:  MOV   DL,[SI]
45      INC   SI
46      MOV   AH,2
47      INT   21H
48      LOOP PUTC
```

FNAME

0EH

07H

'A'

'B'

'C'

','

'T'

'X'

'T'

0DH

九. 软中断实例：PC机BIOS调用及DOS功能调用

2. BIOS调用DOS中断和功能调用

3) DOS功能调用使用举例

示例：

49		POP	AX
50		CMP	AX, 512
51		JE	AGAIN
52		MOV	AH, 3EH
53		INT	21H
54		RET	
55	EXIT:	LEA	DX, ERRMSG
56		MOV	AH, 9
57		INT	21H
58		RET	
59	FILE	ENDP	
60	CSEG	ENDS	
61		END	FILE