**第1部分 指令**

**1.1数据传送**

1、通用数据传送

MOV dest，src

PUSH OPRD ;16位寄存器或者存储器字单元

POP OPRD ;不能从栈顶弹出一个字给CS

XCHG REG/MEM，MEM/REG;两操作数必须有一个是寄存器操作数,不允许使用段寄存器.

XLAT ;输入：用BX的内容代表表格首地址，AL内容为表内位移量， BX+AL得到要查找元素的偏移地址；输出：将BX+AL所指单元的内容送AL。

2、输入输出

IN acc，PORT ;PORT为端口地址，acc为累加器AL或AX

OUT PORT，acc

直接寻址：8位端口地址，地址范围：0-0FFH。 如 IN AL，30H

间接寻址：16位端口地址，由DX指定端口地址。如 IN AL，DX

3、取偏移地址

LEA REG，MEM

**1.2算术运算**

**影响标志位CF、AF 、SF、OF、PF、ZF**

1、加法运算

ADD OPRD1，OPRD2 ;**段寄存器不能作为操作数**

ADC OPRD1，OPRD2 ;常用于多字节数相加

INC OPRD ;不影响CF标志

2、减法运算

SUB OPRD1，OPRD2 ;OPRD1-OPRD2-->OPRD1

SBB OPRD1，OPRD2

DEC OPRD ;不影响CF标志

NEG OPRD

CMP OPRD1，OPRD2 ; OPRD1-OPRD2

对于无符号数：

（1）ZF=1，OPRD1=OPRD2

（2）CF=0，OPRD1>OPRD2

（3）CF=1，OPRD1<OPRD2

对于有符号数：

（1）ZF=1，OPRD1=OPRD2

（2）OF=SF，OPRD1>OPRD2

（3）OF≠SF，OPRD1<OPRD2

3、逻辑运算

；AND、OR、XOR、TEST使标志位OF=CF=0；

AND OPRD1，OPRD2 ;OPRD1,OPRD2按位与-->OPRD1

OR OPRD1，OPRD2

NOT OPRD ；对标志位无影响

XOR OPRD1，OPRD2

TEST OPRD1，OPRD2 ;OPRD1,OPRD2按位与

4、移位

移动次数一般用CL来指定，若只移位1次，也可以在指令中直接写常数“1”。

SAL OPRD，CL ;有符号，D7/D15🡪CF,D0=0

SHL OPRD，CL ;无符号，D7/D15🡪CF,D0=0

SAR OPRD，CL ;有符号，D7/D15🡪D7/D15,D0🡪CF

SHR OPRD，CL ;无符号, D7/D15=0,D0🡪CF

ROL OPRD，CL ;

ROR OPRD，CL ;

RCL OPRD，CL

RCR OPRD，CL

多字节单元数据联合移位：

SAL M，1

RCL M+2，1

RCL M+4，1

**1.3 串操作**

源串一般存放在数据段，偏移地址由SI指定。允许段重设；目标串必须在附加段，偏移地址由DI指定。

指令自动修改地址指针，修改方向由DF决定。DF=0，增地址方向；DF=1，减地址方向；用指令CLD或者STD，设置DF值。

数据块长度值由CX指定。

REP MOVSB ;初始化SI、DI、CX、DF

REP MOVSW ;初始化SI、DI、CX、DF

REPE CMPSB ;初始化SI、DI、CX、DF

REPNE CMPSW ;初始化SI、DI、CX、DF

REPNZ SCASB ;初始化AL、DI、CX、DF

REPNZ SCASW ;初始化AX、DI、CX、DF

LODSB ; 初始化SI，输出AL; [**DS:SI]🡪AL**

LODSW **;** 初始化SI，输出AX

REP STOSB ;初始化AL、DI、CX、DF

REP STOSW ;初始化AX、DI、CX、DF

**1.4程序控制**

JMP Label/JMP FAR PTR Label

*CODE1 SEGMENT*

*//代码段CODE1中定义SUB1\_FAR段间调用*

*SUB1\_FAR LABEL FAR*

*SUB1:MOVE AX,0FFH*

*…*

*JMP SUB1*

*CODE2 SEGMENT*

*//代码段CODE2中段间调用*

*JMP FAR SUB1\_FAR*

JMP WORD PTR[BX]/ JMP DWORD PTR[BX]

JC/JNC Label ; 条件转移指令均为段内（标号）短转移，即转移范围为：-128-+127

JZ/JNZ

JO/JNO

JP/JPE

**JA/JAE/JB/JBE 无符号数比较大小**

**JG/JGE/JL/JLE 符号数比较大小**

LOOP Label ;初始化CX，**-128～+127范围内循环**

CALL PROC/CALL WORD PTR[SI];段内直接调用和间接调用

CALL FAR PROC/CALL DWORD PTR[SI];段间直接调用和间接调用

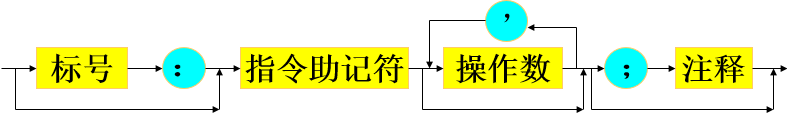
RET ;出现在子程序的最后

INT n/IRET;硬件完成：（1）压栈FLAGS、CS、IP；（2）n x 4🡪向量地址；（3）中断程序入口地址🡪IP、CS。

**CLI、STI、CLC、STC、CMC、CLD、STD**

**第二部分 伪指令**

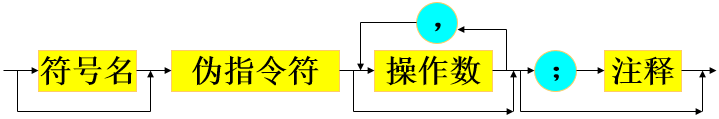
**指令性语句：**



例如： MOV CX,4167 ；CX和4167为操作数

W10MS: LOOP W10MS ；W10MS为标号

**指示性语句：**



伪指令：符号名、伪指令符和操作数

例如： DATA1 DB 12H，13H ；DATA1为符号名，DB为伪指令符，12H，13H操作数

**2.1标识符命名规则**

指令语句中的**标号**和伪指令语句中**符号名**统称为**标识符**。

**命名规则**：

* 1.字符的个数为1~31个；
* 2.第一个字符必须是字母、问号、@或下划线“\_”这4种字符之一；
* 3.从第二个字符开始，可以是字母、数字、@ 、 “\_”或问号“？”；
* 4.不能使用系统专用的保留字。

**保留字：**

* CPU中各寄存器名（如AX、CS等）
* 指令助记符（如MOV、ADD）
* 伪指令符（如SEGMENT、DB）
* 表达式中的运算符（如GE、EQ）以及属性操作符（如PTR、OFFSET等）

**2.2汇编语言数据**

**2.2.1常数**

1、表示形式：

01001001B、254O、0FEH、2.134 E +10。如果常数的第一个数符为字母，为了与标识符区别，必须在其前面冠以数字“0”。

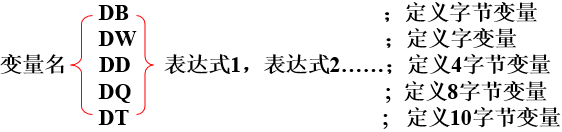
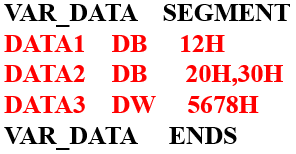
2、字符串常数：

用引号（单引号或双引号）括起来的一个或多个字符，其值为字符的ASCII码值。

**2.2.2变量**

给变量在内存中分配一定的存储单元。

1、一般格式

2、变量赋初值

（1）字符串表达式：

对于DB伪指令：字符串按从左到右，一个字符分配一个字节单元。将字符的ASCII码以地址递增的排列顺序依次存放。

对于DW伪指令：两个字符的存放顺序是前一个字符放在高地址，后一字符放低地址单元。

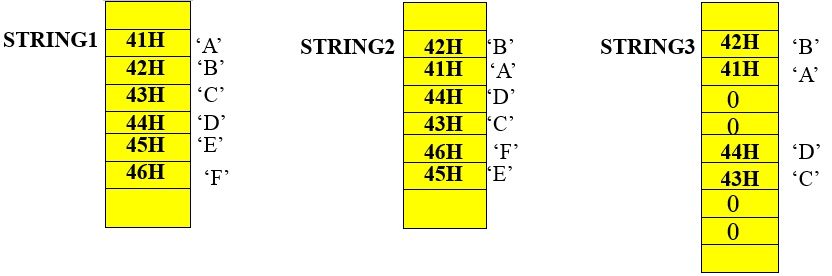
对于DD伪指令，只能给两个字符组成的字符串分配4个字节单元。两个字符存放在较低地址的两个字节单元中。

注意：DW和DD伪指令不能用两个以上字符构成的字符串赋初值，否则将出错。

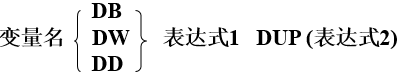
如 STRING1 DB ‘ABCDEF’

STRING2 DW ‘AB’, ‘CD’, ’EF’

STRING3 DD ‘AB’, ‘CD’



（2）DUP表达式



其中：表达式1是重复的次数，表达式2是重复的内容。

如DATA\_B DB 20H DUP(‘AB’)

3、变量的使用

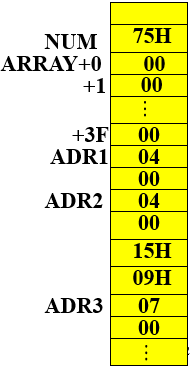
（1）在指令语句中引用

* 在指令语句中直接引用变量名就是对其存储单元的**内容**进行存取。

如 DA1 DB 0FEH

MOV AL,DA1 ;将0FEH传送到AL中

* 当变量出现在变址（基址）寻址或基址变址寻址的操作数中时表示取用该变量的**偏移量**。

如 DA3 DB 10H DUP(?)

MOV DA3[SI]，AL

（2）在伪指令语句中引用

如 NUM DB 75H

ARRAY DW 20H DUP(0)

ADR1 DW NUM ;取变量地址的**偏移量**

ADR2 DD NUM ;取变量段基值和偏移量。

设段基值为0915H，NUM的偏移量为0004H，则存储单元的分配情况如图所示。

（3）LABEL使用：**把字变量**DATA\_WORD**的高低字节作为字节变量**DATA\_BYTE

DATA\_BYTE LABEL BYTE

DATA\_WORD DW 20H DUP（？）

**2.2.3标号**

标号加在一条指令的前面，它就是该指令在内存的存放地址的符号表示，也就是指令地址的别名。

**距离属性（也叫类型属性）指定**

（1）隐含方式

当标号加在指令语句前面时，它隐含为NEAR属性。如SUB1：MOV AX，30H

（2）用LABEL伪指令给标号指定距离属性

SUB1\_FAR LABEL FAR

SUB1: MOV AX,30H

**2.3 符号定义语句**

1、等值语句

（1）常数或数值表达式，如 COUNT EQU 5；NUM EQU COUNT+5

（2）地址表达式，如 ADR1 EQU DS：[BP+14]；ADR1为在DS中以BP作基址寻址的一个存储单元。

（3）变量、寄存器名或指令助记符，如 CREG EQU CX

注意：在同一源程序中，同一符号不能用EQU定义多次。

2、等号语句

（1）等号语句可以对一个符号进行多次定义。

如 NUM=14H

NUM=NUM+10H

（2）等号语句不能为助记符定义别名

如 CBD=DAA；错误语句

注意：等值语句与等号语句都不会为符号分配存储单元。所定义的符号没有段、偏离量和类型等属性。

**2.4 表达式与运算符**

1、算术运算符

+、—、＊、 / 、MOD、SHL、SHR、[ ]

（1）运算符“+”和“-”也可作单目运算符，表示数的正负

（2）使用“+”、“-”、“＊”、和“/”运算符时，参加运算的数和运算结果都是整数。

（3）“/”运算为取商的整数部分，而“MOD”运算取除法运算的余数。

（4）“SHR ”和“SHL ”为逻辑移位运算符。空位补0，如：MOV AX , NUM SHL 1

注意：移位运算符的操作对象是某一具体的数（常数），在汇编时完成移位操作。而移位指令是对一个寄存器或存储单元内容在程序运行时执行移位操作。

（5）下标运算符“[ ]”具有相加的作用。

如 MOV AX，DA\_WORD[20H]

MOV AX，DA\_WORD+20H

下列语句错误：

MOV AX，ARRAY+DA\_WORD

MOV AX，ARRAY+BX+SI

2、逻辑运算符；按位运算

NOT、AND、OR、XOR，

MOV AX，NOT 0F0H；AX=0FF0FH

3、关系运算符

EQ(等于)、NE(不等于)、LT(小于)、 LE(小于等于)、GT(大于)、 GE(大于等于)

（1）若是常量，则按无符号数进行比较；若是变量，则比较它们的偏移量的大小。

（2）关系运算的结果只能是“真”（全1）或“假”（全0）

MOV AX，0FH EQ 1111B；MOV AX ,0FFFFH

VAR DW NUM LT 0ABH；根据符号常量NUM的大小来决定VAR存储单元的值，当NUM<0ABH时，则变量VAR的内容为0FFFFH，否则VAR的内容为0。

4、取值运算符和属性运算符

SEG、OFFSET、PTR

MOV AX , WORD PTR DA\_BYTE[10]

DATA SEGMENT  
VAR1 DB 20H DUP(0)  
VAR2 DW 5A49H  
ADDR DW VAR2 ;将VAR2的偏移量20H存入ADDR中  
 …….  
 MOV BX，VAR2；(BX)=5A49H  
 MOV SI , OFFSET VAR2 ;(SI)=20H

**TYPE：**取变量或标号的类型属性，并用数字形式表示

BYTE 1  
WORD 2  
DWORD 4  
QWORD 8  
TWORD 10

NEAR -1  
FAR -2

**LENGTH：**变量的长度

* 如果变量是用重复数据操作符DUP说明的,则LENGTH运算取外层DUP给定的值。
* 如果没有用DUP说明，则LENGTH运算返回值总是1。

K3 DW 20H DUP（0，1，2 DUP（0）  
K4 DB ‘ABCDEFGH’

**MOV CX, LENGTH K3 ; (CX)=20H  
MOV DX, LENGTH K4 ; (DX)=1**

**SIZE：**取值等于LENGTH和TYPE两个运算符返回值的乘积。

**$:**类似变量, 表示位置计数器的当前值

**ORG：**用来改变位置计数器的值$

DATA1 SEGMENT

ORG 30H

DB1 DB 12H,34H ;DB1在DATA1段内的偏移量为30H

ORG $+20H;保留20H个字节单元，其后再存放'ABCD....

STRING DB ‘ABCDEFGHI’

COUNT EQU $-STRING;计算STRING的长度

DB2 DW $; 取$的偏移量,类似变量的用法

DB3 DB $ ,$+20H ;此语句错误!

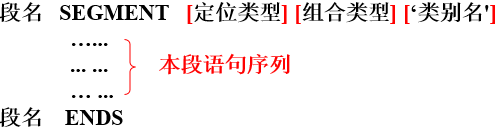
DATA1 ENDS

5、运算符的优先级



**2.5程序的段结构**

1、一般格式



（1）定位类型

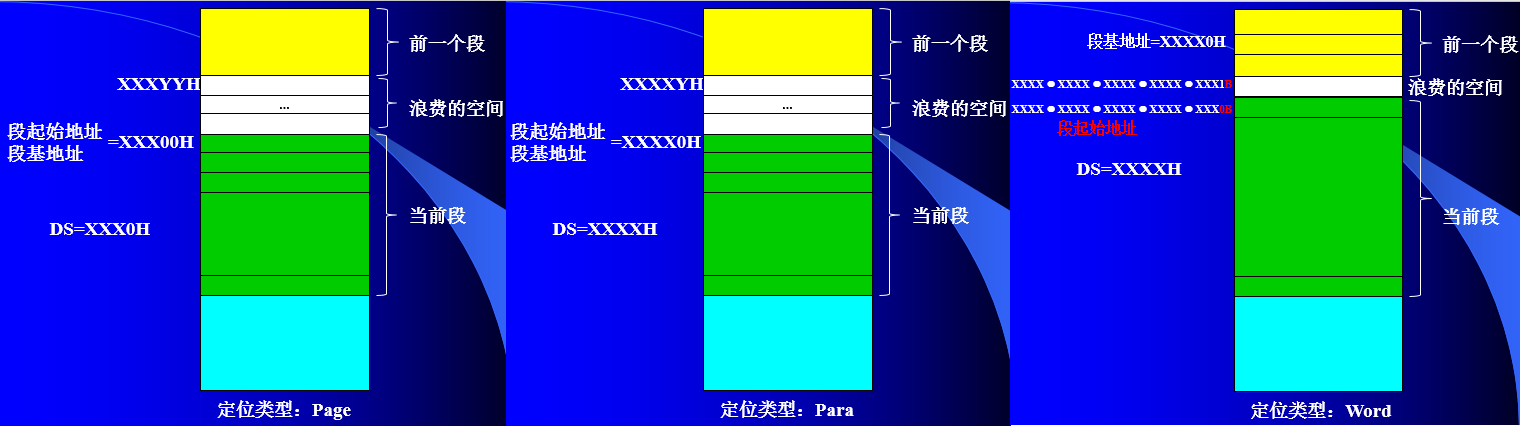
PAGE: 表示该段从一个页面的边界开始存放数据

PARA:表示该段从一个小节的边界开始存放数据，缺省为PARA。

WORD:表示该段从一个偶数字节地址开始存放数据。

BYTE:表示该段起始数据单元地址可以是任一地址值。

注意：定位类型为PAGE和PARA时，段基址直接选用段起始地址，即它们是重合的。定位类型为WORD和BYTE时，段基址与段起始地址可能不同。



（2）组合类型

用来指定段与段之间的连接关系和定位。

NONE：若未指定组合类型，表示本段与其它段无连接关系。

PUBLIC:在满足定位类型的前提下，将与该段**同名的段**邻接在一起，形成一个新的逻辑段，共用一个**段基址**。

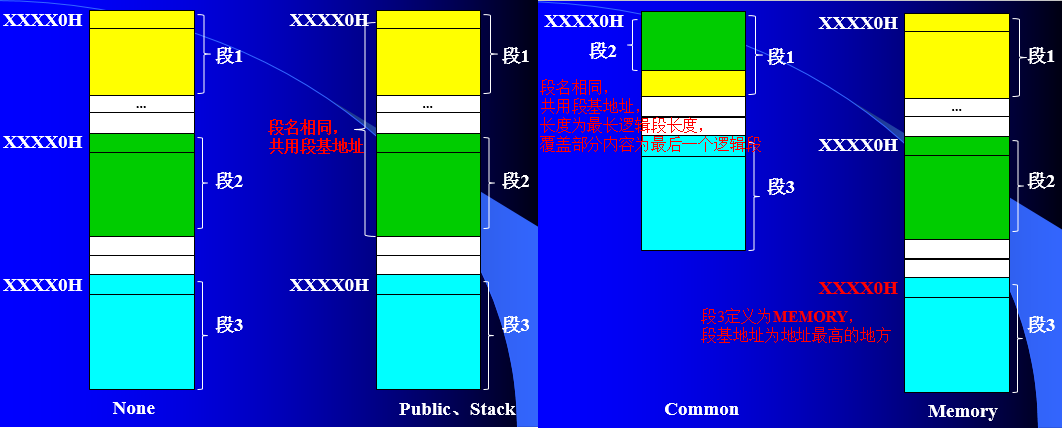
COMMON: 产生一个覆盖段。在多个模块连接时，把该段与其它也用COMMON说明的**同名段**置成相同的段基址，这样可达到共享同一存储区。

STACK:把所有**同名段**连接成一个连续段，且系统自动对SS段寄存器初始化为该连续段的段基址。并初始化堆栈指针SP。

用户程序中应至少有一个段用STACK说明，否则需要用户程序自己初始化SS和SP。

AT表达式：表示本段可定位在表达式所指示的小节边界上。表达式的值也就是段基值。

MEMORY:表示本段在存储器中应定位在所有其它段之后的最高地址上。如果有多个用MEMORY说明的段，则只处理第一个用MEMORY说明的段，其余的被视为COMMON。



1. 类别名

类别名为某一个段或几个相同类型段设定类型名称。系统在进行连接处理时，把**类别名相同的段**存放在相邻的存储区，但段的划分与使用仍按原来的设定。

2、段寻址伪指令

ASSUME的作用是告诉汇编程序,在处理源程序时，定义的段与哪个寄存器关联。

如 ASSUME CS:CODE,DS:DATA1,ES:DATA2

3、段寄存器的装入

（1）DS和ES的装入，用传送数据指令。

如 MOV AX,DATA1

MOV DS,AX

（2）SS装入

* 在段定义伪指令的组合类型项中，使用STACK参数，并在段寻址伪指令ASSUME语句中把该段与SS段寄存器关联。
* 如果在段定义伪指令的组合类型中，未使用STACK参数，或者是在程序中要调换到另一个堆栈，这时，可以使用传送数据指令装入方法。

如 DATA\_STACK SEGMENT

DB 40H DUP(?)

TOP LABEL WORD  
DATA\_STACK ENDS

CODE SEGMENT

......

MOV AX,DATA\_STACK

MOV SS,AX

MOV SP,OFFSET TOP

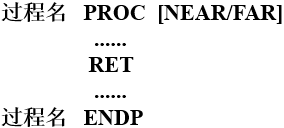
（3）CS的装入

* 由系统软件按照结束伪指令指定的地址装入初始的CS和IP。根据END伪指令的起始地址。

**END 起始地址**

* 段间调用CALL和RET、JMP、中断服务程序。

**4.6子程序**



4.7 编写程序

1、框架(记住)

DSEG SEGMENT

……

DSEG ENDS ;定义数据段

ESEG SEGMENT

……

ESEG ENDS ;定义附加段

SSEG SEGMENT PARA STACK

DW 20H DUP (0)

SSEG ENDS ;定义堆栈段

CSEG SEGMENT

ASSUME CS:CSEG,DS:DSEG, ES:ESEG,SS:SSEG

START: ;指令开始地址

MOV AX,DSEG ;

MOV DS,AX ;初始化DS

MOV AX,ESEG;

MOV ES,AX ;初始化ES

……

MOV AH, 4CH ;

INT 21H ;返回DOS操作系统

CSEG ENDS

END START ;汇编结束标志

2、PPT“3.4 汇编语言程序设计-3” P3 “例1 利用学号查学生的数学成绩表。”

3、PPT“3.4 汇编语言程序设计-3” P9 “例2 10个无符号数找最大值/最小值；有符号数找最大值/最小值。”

（1）CMP

（2）JAE/JBE；无符号；JGE/JLE：有符号

4、PPT“3.4 汇编语言程序设计-3” P12 “例3 实现将存储器中的源数据块传送到目的数据块。”

5、PPT“3.4 汇编语言程序设计-3” P34 “例6 将字单元VARW 中含1的个数(含1的个数是指用二进制表示时,有多少个1)统计出来，存入CONT单元中”

6、字符串中查找某个字符 AL= ‘A’ ，XLAT

7、10个有符号/无符号数进行排序

**第3部分 输入输出和中断系统**

**3.1输入输出系统**

1、8088/8086的I/O端口编址

* 采用I/O独立编址方式(但地址线与存储器共用)
* 地址线上的地址信号用IO/#M来区分
* I/O操作只使用20根地址线中的16根：A15～A0
* 可寻址的I/O端口数为64K(65536)个
* I/O地址范围为0～FFFFH
* IBM PC只使用了1024个I/O地址(0～3FFH)

2、I/O地址译码

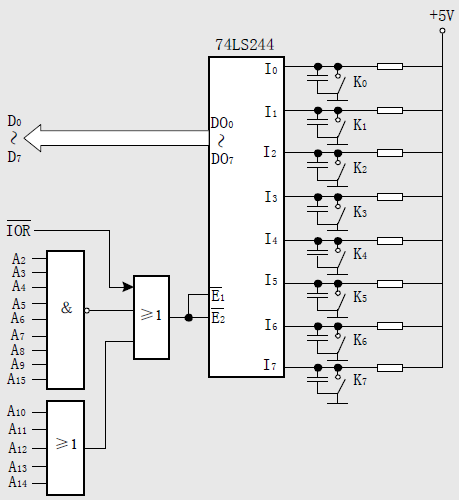
* 一个接口电路中可以有多个端口，也可以只有一个端口。
* 对全地址译码：
  + 当接口只有一个端口时， 16位地址线应全部参与译码，译码输出直接选择该端口；
  + 当接口具有多个端口时，则16位地址线的高位参与译码（决定接口的基地址），低位用于寻址接口中要访问的端口。
* 对部分地址译码：
  + 仅用部分地址信号参与译码
  + 对含多个端口的接口，则为部分高位地址。

**I/O系统中，因地址资源丰富，多采用部分地址译码。**

3、接口特点

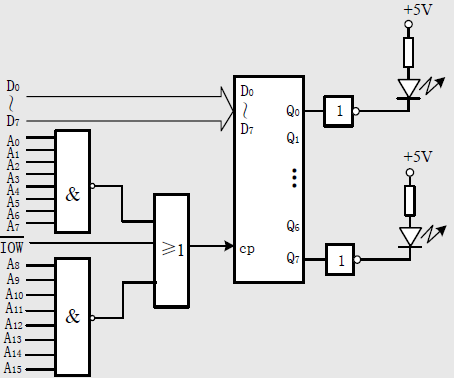
* 输入接口：
  + 要求对数据具有控制能力
  + 常用三态门实现
* 输出接口：
  + 要求对数据具有锁存能力
  + 常用锁存器实现

4、三态门74LS244

****

接口芯片地址：83FCH～83FFH，当地址总线上A15～A0出现此4个地址之一，#E1#E2端有效，三态门导通。

5、锁存器74LS273

****

锁存器74LS273的端口地址：FFFFH

6、PPT“3.5 输入输出和中断技术-1”P32 编程实现“当开关闭合时，使输出编码使发光二极管亮。”

**7、**PPT“3.5 输入输出和中断技术-1”P36 “例1”

8、PPT“3.5 输入输出和中断技术-1”P41 “例2”

**3.2 8086/8088中断类型**

1、内部中断

0=除法出错中断；1=单步中断；3=断点中断；4=溢出中断；n型中断（DOS软中断【20H-3FH】和BIOS【11H-1AH】软中断）：INT n。

中断响应过程：**无#INTA，中断类型码固定或者指令直接给出。**

（1）将类型码乘4,得到中断向量地址

（2）硬件保护现场，将FLAGS压栈

（3）清除TF和IF标志

（4）保存断点，CS和IP压栈

（5）中断服务程序入口地址送CS和IP

（6）执行中断服务程序。

2、外部中断

（1）非屏蔽中断

2=NMI

（2）可屏蔽中断

08H~0FH：主8259管理，包括定时器、键盘、串口、并口、软盘驱动器。

70H~77H：从8259管理，包括实时时钟、协处理器、硬盘控制器。

（3）可屏蔽中断响应

两个中断响应总线周期。

第一个总线周期：地址/数据总线置高阻（ALE=0），送出#INTA响应信号，禁止其他总线请求。最大模式下，送#LOCK信号，禁止其他处理器访问总线。

第二个总线周期：送第2个#INTA，通知8259送出中断类型码n。

3、中断向量表

向量地址=n×4

用户中断服务程序，需要用户主程序初始化中断向量表中，该中断号对应的服务程序地址。如将类型码为48H的中断服务子程序TIMER的中断向量放入向量表。

* + MOV AX，0000H
  + MOV DS，AX
  + MOV SI，**0120**H
  + MOV BX，OFFSET TIMER
  + MOV [SI]，BX
  + MOV BX，SEG TIMER
  + MOV [SI+2]，BX

**3.3 中断控制器8259**

3.3.1 8259A中断处理过程

1、中断请求输入线（IR0-IR7）上有中断请求，则IRR相应位=1

2、若IMR该位未屏蔽，则8259由INT引脚向CPU发出中断请求信号INTR

3、CPU处于开中断，CPU执行完指令后，CPU用#INTA信号作为对INTR的响应。

4、8259收到CPU的第1个#INTA脉冲后，将最高优先权的ISR位置1，并使IRR相应位复位。

5、8259在第2个总线周期，将中断类型号送到数据总线上。

6、若8259工作在AEOI，8259在第2个#INTA结束时，复位ISR相应位。非AEOI，则需要CPU在中断服务程序结束时，写入操作命令字OCW2中置位EOI位，8259接收到名字字后将ISR的该位复位。

3.3.2 8259工作方式

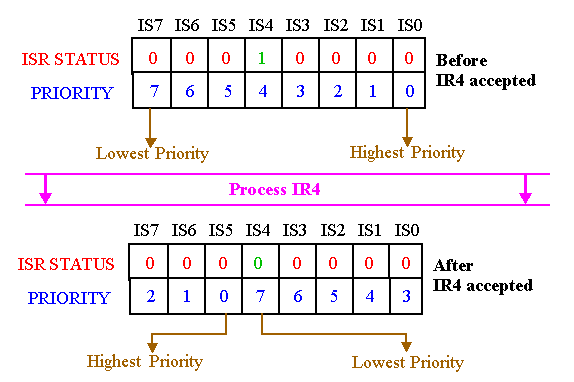
1、中断嵌套

* 不允许嵌套
* 全嵌套（普通嵌套，允许优先级高的中断请求）
* **特殊全嵌套**（允许响应同级中断请求）P270 图6-28

在多片8259A级联的情况下，主片通常设置为特殊完全嵌套方式，从片设置为普通嵌套方式。当主片响应某一个从片的中断请求时，从片中的 IR7～IR0 的请求都是通过主片中的某个IRi请求引入的。因此从片的 IR7～IR0 对于主片IRi来说，它们属于同级，只有主片工作于特殊完全嵌套方式时，从片才能实现完全嵌套。

2、中断优先方式

* 固定优先级（从高到低顺序为IR0-IR7）
* 优先级自动循环方式（多个中断源优先级相等的场合）
* 普通自动循环：初始优先级为默认，从高到低顺序为IR0-IR7。
* 特殊自动循环：初始的最低优先级是由OCW2确定



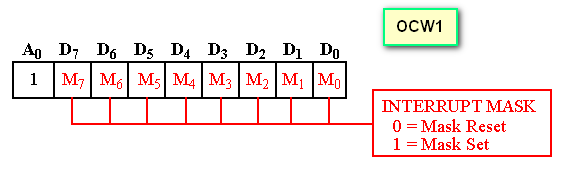
开始的时候，优先级从高到低是IR0, IR1, IR2, …, IR7。某时刻，IR4正在被CPU处理。处理完成后，IR4的优先级变为最低，IR5的优先级变为最高，也就是说，新的优先级序列从高到低是IR5,IR6,IR7,IR0,IR1,IR2,IR3,IR4.

3、中断结束方式

* 自动中断结束AEOI：适用于没有中断嵌套的情况，初始化命令字ICW4的AEOI位置1实现。
* 常规中断结束EOI：全嵌套，操作命令字OCW2中的EOI=1、SL=0、R=0。输出到8259A的偶地址端口（A0=0）
* 特殊中断结束SEOI：非全嵌套OCW2的EOI=1、SL=1、R=0，L2、L1、L0这三位指出了对ISR中的哪一位进行清除。

4、屏蔽中断源方式

* 常规屏蔽：IMR某位为“1”表示屏蔽对应的中断请求。



* 特殊屏蔽SMM：8259A工作在特殊屏蔽方式时，所有未被屏蔽的优先级中断请求（较高的和较低的）均可在某个中断过程中被响应，即低优先级别的中断可以打断正在服务的高优先级中断。

在特殊屏蔽方式中，可在**中断服务子程序**中用中断屏蔽命令屏蔽当前正在处理的中断级，同时可使其在ISR中的对应位清零，这样一来不仅屏蔽了当前正在处理的中断级，而且也真正开放了较低级别的中断请求。在这种情况下，虽然CPU仍然继续执行较高级别的中断服务子程序，但由于ISR中对应位已经清零，就如同没有响应该中断一样。所以，此时对于较低级别的中断请求，CPU可以响应。

特殊屏蔽方式由于打乱了正常的全嵌套结构，低优先级中断可以打断正在服务的高优先级中断，高优先级中断也可以打断正在服务的低优先级中断，此时，根据ISR的内容无法确定出刚刚所处理的中断。**只能用SEOI命令**（命令中指定出要复位的中断级）来清除ISR中的指定位。

5、中断触发方式

8259A中断请求输入端 IR7～IR0 的触发方式有电平触发和边沿触发两种，由初始化命令字ICW1中的LTIM位来设定。

（1）电平触发

当 LTIM=1 时，为电平触发方式。当8259A检测到IRi（i＝0～7）端有高电平时产生中断。在这种触发方式中，要求触发电平必须保持到中断响应信号#INTA有效为止。在CPU响应中断后，应及时撤销该请求信号，以防止CPU再次响应，出现重复中断现象。

（2）边沿触发

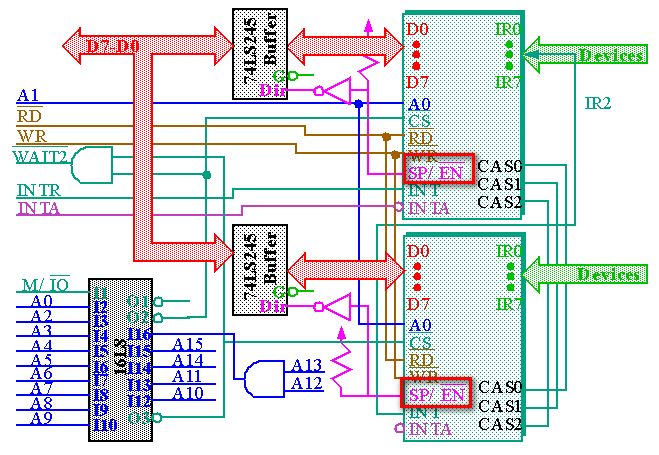
当 LTIM=0 时，为边沿触发方式。当8259A检测到IRi端有由低到高的跳变（上升沿）信号时产生中断。

5、数据总线连接方式

8259A的数据线与系统数据总线的连接有缓冲和非缓冲两种方式。

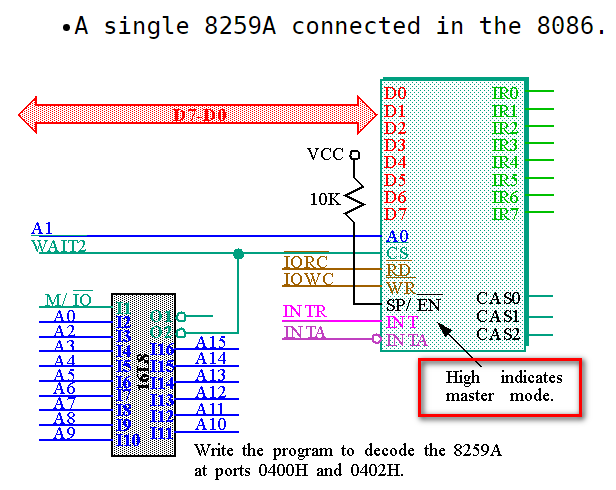
（1）缓冲方式

如果8259A通过总线驱动器和系统数据总线连接，则应选择缓冲方式。此时EN为输出引脚。在8259A输出中断类型号的时候，EN输出一个低电平，用此信号作为总线驱动器的启动信号。在缓冲方式下，由ICW4的M/S位来标识本8259A是主片还是从片。



（2）非缓冲方式

如果8259A的数据线与系统数据总线直接相连，那么应选择非缓冲方式。此时SP为输入引脚，用其电平高低来标识本8259A是主片（SP=1）还是从片（SP=0）。在非缓冲方式下，ICW4的BUF＝0，M/S位无意义。



3.3.3 8259编程

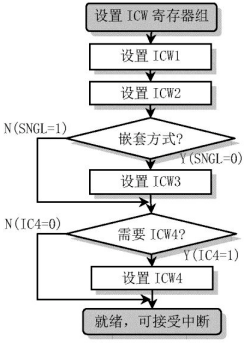
命令字端口地址:



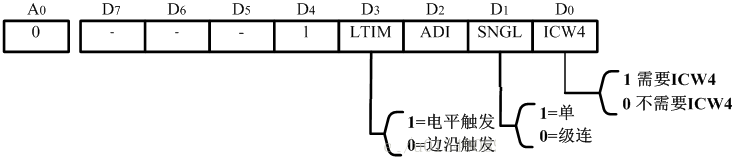
1、初始化命令字ICW

在8259A可以正常工作之前，必须首先设置初始化命令字 ICW (Initialization Command Words)寄存器组的内容。而在其工作过程中，则可以使用写入操作命令字 OCW (Operation Command Words)寄存器组来随时设置和管理8259A的工作方式。

A0线用于选择操作的寄存器。在PC/AT微机系统中，当A0=0时芯片的端口地址是0x20(主芯片)和0xA0(从芯片）；当 A0=1时端口就是0x21(主芯片)和0xA1(从芯片)。

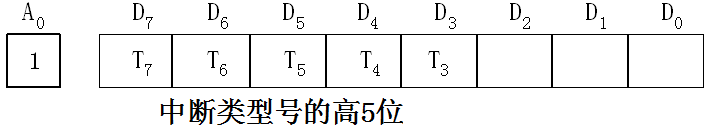


（1）初始化字ICW1

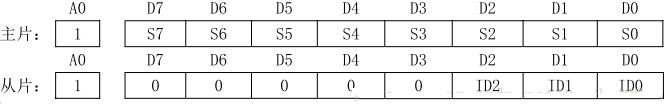


收到ICW1后，8259完成以下工作：清除ISR、IMR；初始化优先级IR0-IR7；普通屏蔽；非自动EOI；读出IRR。

（2）中断向量码ICW2



（3）级联控制字ICW3

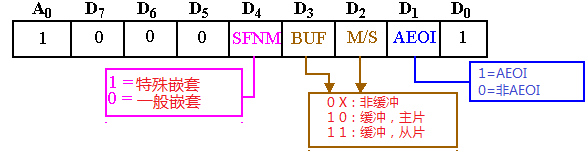


主芯片的端口地址是0x21，从芯片的端口地址是0xA1。

对于主片，Si=1,表示IRi接从片的INT引脚。主片 S7~S0 各比特位对应级联的从片。

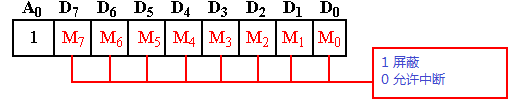
对于从片，ID2~ID0 对应各从片的标识号，即连接到主片的中断级。

（4）ICW4中断结束

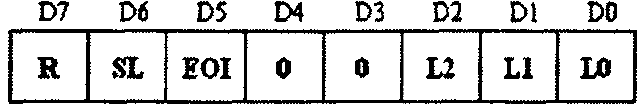


2、操作命令字OCW

（1）中断屏蔽字OCW1



（2）中断结束和优先级循环OCW2



D2DlDO(L2LlLO)：当D6=1时，这三位才有效。此时有两个用处：第一是在发特殊结束中断命令时，三位指出要清除ISR寄存器中的哪一位；第二是在特殊优先级循环方式中，三位指出循环开始时初始优先级队列的最低优先级。

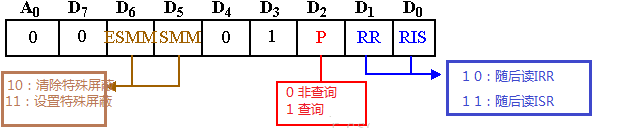
D4D3：OCw2的特征位，恒等于0；

D5(EOI)：中断结束命令位。1=在中断结束后需发送中断结束命令，使ISR相应位清0；0=不需要发送中断结束命令，即为中断自动结束方式。

D6(SL)：低三位有效位。l=低三位有效，0=低三位无效。

D7(R)：优先级循环方式位。1=设置优先级循环方式，O=非循环方式。

（3）屏蔽方式和状态读出控制字OCW3



* **查询方式：**对同一地址读，读出本片8259的中断请求。



* 读8259状态

IRR和ISR：由RR、RIS两位确定，且和OCW3同一地址读取。

IMR：A0=1（即主片21H或者从片A1H）时，读到的就是IMR的内容。