第三章

指令和伪指令

汇编指令是机器指令的助记符，可以被汇编成机器指令，由CPU执行

伪指令是在汇编过程中实现数据定义、分配存储区、指示程序结束等功能，由汇编器来执行，汇编完成后伪指令就消失了

程序结构

段定义：
段定义伪指令：SEGMENT、ENDS

ASSUME语句：
该伪指令用于设定段寄存器与某个段的关联。

装填DS：
指令语句：　MOV AX,数据段名　　MOV DS,AX;使得DS获得数据段的段地址

返回DOS：
这是返回DOS的方法之一，通过执行4CH号DOS功能调用返回DOS。

结束语句：
伪指令语句：　END　标号　　本例中是：　END　START
表示程序到此为止，并从标号处开始的。即：汇编器在汇编过程中遇到END就结束汇编，同时指明程序从标号所在的一行开始。

语句

名字项　操作项　操作数项　　;注释项

名字项：符合命名规则的标识符,表示符号地址。
操作项：指令或伪指令的助记符.
操作数项：一个或多个表达式，是操作的对象。
注释项：以半角“;”开始,说明语句的功能,可选。

伪指令语句:
符号定义符　参数.....　　;注释

```
A       DB    12H,78H       ; 数据定义
```

段定义伪指令
段名 SEGMENT [定位类型] [组合类型] [使用类型] ['类别']

定位类型：PAGE、PARA、WORD、BYTE
组合类型：PRIVATE、PUBLIC、STACK、COMMON、MEMORY、AT 表达式
使用类型：用来说明使用的寻址方式
'类别'： 连接时组成段组的名字

数据的定义
变量：有自己的名字，在程序运行中可能发生变化，可以在定义时置一个"初值"。
常量：可以直接写在指令内，也可以事先定义在数据段内。
缓冲区：从键盘、外存储器输入时，需要在数据段里事先留出必要的存储单元。向输出设备输出数据时需要定义输出缓冲区。

DB（Define Byte）：每个数据占用一个字节（8b）

DW（Define Word）：每个数据占用一个字（16b）

DD（Define Double Word）：每个数据占用一个双字(32b)

DQ（Define QuartByte）：每个数据占用8个字节(64b)

DT（Define Ten Byte）：每个数据占用10个字节（80b）

格式:变量名 DB 操作数1，操作数2，..... ;注释
操作数可以是常数或表达式
操作数可以是字符串
操作数为"？"，保留空单元
操作数字段使用重复操作符DUP
也可以定义浮点数
操作数也可以为地址

变量的属性 地址属性和类型属性

数据定义语句前面的变量的值是该语句中第一个数据项在当前段内的第一个字节的偏移地址。每个存储单元与一种类型联系。

名字	大小
BYTE	1
WORD	2

名字	大小
DWORD	4
FWORD	6
QWORD	8
TBYTE	10

PTR:用来建立一个符号地址，但本身并不分配存储器，只是用来给已分配的存储器地址**赋于另一种属性。**

格式：type ptr expression

标号

标号是指令的地址
标号可以定义在目的指令的同一行的最前面，也可以在目的指令前一行单独一行定义。
标号通常作为**转移指令的目标地址**

同变量一样具有地址属性和类型属性 段,偏移量 near 还是far

符号定义

- 符号名 EQU 表达式

CNT EQU 15
MOVE EQU MOV
汇编时，对EQU定义的符号名用对应的表达式进行“替换”:
MOV CX, CNT+1 ;等价于MOV CX, 15+1
MOVE AX,[SI] ;等价于MOV AX,[SI]

- 符号名 = 表达式
  “=”定义符号名时，只能使用常数表达式，而且对一个符号名可以多次定义。一个新的定义出现后，原来的定义自动终止。

注意(1)用EQU定义的符号名**不允许重复定义**
(2)在同一源文件中，同一符号名不能同时使用EQU和“=”语句来分别定义。

- LABEL和THIS
  标识符 LABEL 类型
  标识符 EQU THIS 类型
  定义一个指定类型的变量名和标号

表达式

表达式是将常量、符号地址(变量和标号)、符号常量，用运算符合括号连接起来的句子。
指令中的**操作数，包括立即数和存储器操作数**都可以用一个表达式来代替。
这个表达式在汇编成目标的时候进行计算，它的结果用来产生目标代码。

常数

数值常数，如12,34H,101010B
字符串常数，如‘ASDF’
符号常数，如CNT EQU 100

数值表达式

算术运算符 + 、 - 、 * 、 / 、 MOD
逻辑运算符AND 、OR 、NOT 、XOR
关系运算符要求的两个操作数必须都为数字或是同一段内的两个存储器地址。结果为真表示为全1，结果为假表示为0。
关系运算符EQ、NE、LT（小于）、GT（大于）、LE（小于等于）、GE（大于等于）
专用操作符OFFSET、SEG、TYPE、LENGTH、SIZE
功能：分别回送**偏移地址、段地址、以字节数表示的类型值、使用DUP时的单元数、字节数。**
属性操作符（合成操作符） PTR、段操作符、SHORT、THIS

- 段操作符:用来表示一个标号、变量或地址表达式的段属性。如：MOV AX, ES:[BX+SI]
- SHORT:用来表明JMP指令中转移地址的属性，指出转移地址是下一条指令地址的-128~127字节范围内。
- PTR:用来建立一个符号地址，但本身并不分配存储器，只是用来给已分配的存储器地址赋于另一种属性。
- THIS:同PTR一样建立一个指定类型的地址操作数。该操作数的段地址和偏移地址与下一个存储单元地址相同。

地址表达式

“+、-”运算符对构成有效地址的各个分量进行“加”、“减”操作
设变量X的偏移地址为1020H
MOV BL, X-10H ;产生EA=1010H

```
MOV  AL, X+6  ;产生EA=1026H
MOV  AX, 2[BX][DI]  ; [BX+DI+2]
```

运算符优先级

运算符与操作符的优先级:

1. 圆括号、方括号、结构变量、LENGTH、SIZE
2. 名：（段取代）
3. PTR、OFFSET、SEG、TYPE、THIS、段操作符
4. HIGH、LOW
5. *、/、MOD、SHL、SHR
6. +、-
7. EQ、NE、LT、LE、GT、GE
8. NOT
9. AND
10. OR、XOR
11. SHORT

其他伪指令

正常情况下，段内偏移地址从“段名 SEGMENT”以下,以0000H开始,每分配一个单元,偏移地址加1。

org

ORG语句
格式：ORG　　表达式
用来设置当前地址计数器的值。其中，表达式必须是一个可以计算得到正整数，数值范围是0～65535。

```
DATA  SEGMENT
ORG  100H
X  DB  12H
Y  DW  34H
Z  DD  56H
DATA  ENDS
```

这样，由于用ORG指令重新设置DATA段的起始偏移地址，所以，X单元的偏移地址为0100H、Y单元的偏移地址为0101H、Z单元的偏移地址为0103H。

地址计数器$

汇编器使用地址计数器来保存当前正在汇编的指令或数据的偏移地址 。
每个段开始时，地址计数器清0，然后根据指令或数据的具体汇编过程，地址计数器逐步增加。用“$”直接引用当前地址计数器的值

数据的传送

•指令语句
[标号:] 操作码 [操作数][,操作数] [;注释]
•伪指令语句
[符号] 定义符 参数1,…，参数n [;注释]
•宏指令语句
[标号:] 宏指名 参数1,…，参数n [;注释]
名字项 操作项 操作数项 注释项
指令语句的格式：
[标号:]程序员给这一行起的名字，后面跟上冒号，代表这一行的地址。
操作码是这条指令需要完成的操作，用指令助记符表示。
[操作数]是指令的操作对象，指令的操作数可以0～3个。
两个操作数时，右面的操作数称为“源操作数”，左面的操作数称为“目的操作数”。
“源操作数”参与指令操作，不保存结果，内容不会改变。“目的操作数”参与指令操作，还保存指令的操作结果。
[;注释]用来添加一些说明，例如说明本行指令的功能。

- 操作数的种类
  (1) 立即数——操作数本身，存放在指令代码中。
  (2) 寄存器操作数——操作数存放在CPU内部寄存器中。
  (3) 内存操作数——操作数存放在内存中，一般在数据段、附加段、甚至堆栈段。给出的是操作数的地址。
  (4) 端口操作数——在I/O指令中，给出的端口地址。如 IN AL，34H

寻址方式

1）立即寻址
2）寄存器寻址
（3）直接寻址
（4）间接寻址
（5）相对寻址
（6）基址变址
（7）相对基址变址
（8）比例因子

立即寻址

MOV AL，5
MOV AX，3064H
MOV EAX，12345678H

寄存器寻址

MOV AX，BX
MOV ECX，EDX
MOV DL，AL

直接寻址

DATA SEGMENT
A DB 12, 34, 56
ARRAY DW 55, 66, 77, 88, 99
DATA ENDS
设：已把DATA代表的段基址装入DS
MOV BL, A ; 也可以写作 MOV BL, [A]
MOV BH, A+1 ; 或MOV BH, [A+1]
或MOV BH, A [1]

间接寻址

若已经定义：A DB 12, 34
MOV SI, OFFSET A ; 把变量A的偏移地址装入SI
; OFFSET是保留字，表示取出后面变量的偏移地址
MOV BL, [SI] ; 变量A的第一个值送BL
MOV BH, [SI+1] ; 第二个值送BH, MOV BH, 1[SI]
16位80X86微处理器只有BX，BP，SI，DI这4个寄存器可以用来“**间接寻址**”。不另加说明的话，使用BP时自动用SS的值作为段基址，使用BX，SI，DI时自动用DS的值作为段基址。

相对寻址

MOV AX,COUNT[BX]

基址变址

MOV AX,[SI][BX]

相对基址变址

MOV AX,COUNT[BX][DI]

32位存储器寻址

[基址+比例因子×变址+位移量]
MOV AX, ARRAY[4] ;直接寻址，有效地址=ARRAY+4
MOV AX, [ECX] ;可以用任何一个通用寄存器间接寻址
MOV AX, [EAX+4] ;寄存器相对寻址
MOV AX, [EBX+ECX] ;基址（EBX）变址（ECX）寻址
MOV AX, [EBP+EDX+4]
;相对基址（EBP）变址（EDX）寻址，使用SS
MOV AX, [EBX+4*ESI]
;变址寄存器可以乘上比例因子1，2，4，8
MOV AX, [8*EBP+ECX+6]
;相对基址（ECX）变址（EBP）寻址，使用DS

传送指令

MOV指令

格式：MOV DST，SRC ；DW/W/B
功能：（DST）←（SRC）
说明：
DST：目的操作数，可以是M（存储器）、R（寄存器）、A（累加器）；
SRC：源操作数，可以是M、R、A、立即数。
不影响标志位
注意:
（1） 源操作数与目的操作数可以是字节、字或双字，但必须有相同的类型；
（2）源操作数与目的操作数不能同时为存储器操作数；
（3）目的操作数不能是立即数；
（4）FLAGS、EFLAGS、IP、EIP不能用作操作数。
(5）段寄存器作为操作数时：
•源操作数与目的操作数不能同时为段寄存器；
•目的操作数是段寄存器时，源操作数只能是寄存器或存储器，不能是立即数；
• CS不能用作目的操作数。

LEA(Load Effective Address, 装载有效地址)指令

格式：LEA REG16, MEM
功能：把源操作数的偏移地址装入目的操作数。
REG16表示一个16b通用寄存器；
MEM是一个存储器操作数。

地址传送指令LDS,LES,LFS,LGS

格式：LDS REG16, MEM32
功能：从存储器取出4B，送入REG16和DS 。

扩展传送指令MOVSX, MOVZX

•MOVSX 带符号扩展传送指令
•MOVZX 带零扩展传送指令

扩展指令CBW,CWD,CWDE,CDQ

CBW ；将AL寄存器内容符号扩展成16b，送入AX
CWD ；将AX寄存器内容符号扩展成32b, 送入DX和AX
CWDE ；将AX寄存器内容符号扩展成32b，送入EAX
CDQ ；将EAX寄存器内容符号扩展成64b,送入EDX和EAX

XCHG,XLAT

XCHG REG/MEM, REG/MEM ;交换指令
交换源、目的操作数的内容，两个操作数有相同的类型，**不能同时为存储器操作数。**
XLAT ； AL←DS: [BX+AL] ;换码指令
用AL寄存器的内容查表，结果存回AL寄存器。表格的首地址事先存放在**DS: BX**中。

堆栈操作指令

PUSH 进栈指令

格式：PUSH SRC ； W/DW
功能：将寄存器或存储单元中的内容送入堆栈。
操作：SP←SP-1 (SP) ←SRCH SP←SP-1 (SP) ←SRCL

或 SP←SP-2 (SP) ← (SRC)

说明：SRC为16/32位操作数，常为R或M
不允许字节操作，8086不允许立即数。

POP 出栈操作

格式：POP DST ；W/DW

功能：将SP所指的栈顶字单元的内容弹出，送入DST指定的寄存器或字存储单元中。

操作：DSTH←（SP），SP←SP+1　DSTL←（SP），SP←SP+1

或　（DST）←（SP）　SP←SP+2

说明：DST为16/32位操作数，可为R或M。
不允许POP CS，也不可能是立即数

•PUSHA/PUSHAD 所有寄存器进栈指令
•POPA/POPAD　所有寄存器出栈指令

顺序：AX，CX，DX，BX，SP，BP，SI，DI。
或者： EAX，ECX，EDX，EBX，ESP，EBP，ESI，EDI。
•PUSHF/PUSHFD（PUSH the Flags or eflags）
标志进栈
•POPF/POPFD（POP the Flags or eflags）
标志出栈
PUSHF和POPF指令一般用于在子程序和中断处理程序的首尾，起到保护和恢复主程序中运行结果的标志的作用。