



3.16 宏指令

- 宏指令是用户自己定义的指令，是由指令和伪指令构成的一个程序段；
- 宏指令用一个标识符（称为宏指令名）来表示指定的一段程序，在调用该宏指令时可以在程序汇编时展开成相应的程序段（这一过程称为宏展开）。
- 可以将程序中多次重复的程序段经过整理后定义成宏指令，从而使汇编语言的程序设计得到简化。
- 采用宏指令时，必须先定义后调用；

3.16 宏指令

一、宏指令的定义

宏指令名 **MACRO** <形式参数> ;宏定义开始伪指令

宏体（由指令语句组成的能完成某一特定功能的一段程序）

ENDM ;宏定义结束伪指令



3.16 宏指令

- <宏指令名>为标识符，即定义的宏指令的名称；
- <形式参数>为宏体中使用的参数，它允许宏指令中具有待定的参数，在宏调用时必须用实际的参数代替它，因此每次宏指令调用时可以通过改变实际参数而改变宏指令的功能，使宏指令的设计更加灵活。
- 宏指令可有多个形式参数，它们之间以逗号间隔，但总长度不得超过132。调用时，实际参数的个数和类型必须与形式参数一致。
- 宏指令也可以不带任何形式参数。



3.16 宏指令

二、宏调用与宏展开

宏调用的格式为：

宏指令名 <实际参数>

注：实际参数应该与宏指令定义时的形式参数一一对应。

特点：

- 在汇编过程中，当汇编到宏调用语句时，将用宏体中的一段程序来代替这一条宏指令，并且用实际参数取代形式参数；
- 在汇编过程产生的LST（列表）文件中，可看到宏展开语句的前面有一个“+”号或其它符号，以表示与其它指令的区别。



3.16 宏指令

例：设计对通用寄存器左移4位的宏指令。

解：设形式参数REG表示任一个调用寄存器，则宏指令LSHIFT4定义为：

```
LSHIFT4    MACRO    REG
            MOV     CL, 4
            SHL     REG, CL
            ENDM
```

应用：在程序中，如果要对BX的内容左移4位，则

```
LSHIFT4    BX
```

如果要对DL的内容左移4位，则

```
LSHIFT4    DL
```



3.16 宏指令

例：设计对通用寄存器的左移、右移N位的宏指令。

解：宏指令SHIFTN定义为如下形式，其中，REG表示任一个调用寄存器，N表示移位次数，CC表示移位方式。

```
SHIFTN    MACRO    N,    REG,    CC
```

```
            MOV      CL, N
```

```
            S&CC     REG, CL
```

```
        ENDM
```

- CC取HR时，表示采用SHR移位；
- CC取HL时，表示采用SHL移位；
- CC取AR时，表示采用SAR移位；
- CC取AL时，表示采用SAL移位；

符号 “&”为连接符，表示字符串CC与S直接连接。



3.16 宏指令

三、宏指令中的标号与变量

- 当宏指令定义中出现标号和/或变量定义语句时，由于多次调用会出现相同的标号和/或变量的定义，从而导致汇编出错；
- 应该采用伪指令LOCAL将宏指令中的标号和/或变量指定为局部标号或变量，这样在宏展开时，系统会自动将其中标号和/或变量赋以新的标识符，约定依次采用“??0000”、“??0001”、“??0002”等等。
- 局部标号或变量的定义格式为：
LOCAL 标号或变量



3.16 宏指令

例：延时宏指令的宏定义：

```
DELAY MACRO VALUE1, VALUE2
    LOCAL AGAIN1, AGAIN2
    PUSH    AX
    PUSH    CX
    MOV     CX, VALUE1
AGAIN1: MOV  AX, VALUE2
AGAIN2: DEC  AX
        JNZ  AGAIN2
    LOOP    AGAIN1
    POP     CX
    POP     AX
ENDM
```




3.16 宏指令

宏体中的标号**AGAIN1**及**AGAIN2**被定义为局部标号，则如果多次调用**DELAY**宏指令，如：

...

DELAY 6789H, 0FFFFH

...

DELAY 0FFF0H, 80000H

...

宏展开后为如下程序片段：



3.16 宏指令

```
... ..  
    DELAY    6789H, 0FFFFH  
+    PUSH    AX  
+    PUSH    CX  
+    MOV     CX, 6789H  
+ ??0000:  
    MOV     AX, 0FFFFH  
+??0001:  
    DEC     AX  
+    JNZ     ??0001  
+    LOOP    ??0000  
+    POP     CX  
+    POP     AX
```

```
... ..  
    DELAY    0FFF0H, 8000H  
+    PUSH    AX  
+    PUSH    CX  
+    MOV     CX, 0FFF0H  
+ ??0002:  
    MOV     AX, 8000H  
+??0003:  
    DEC     AX  
+    JNZ     ??0003  
+    LOOP    ??0002  
+    POP     CX  
+    POP     AX  
... ..
```

3.16 宏指令



注：标号**AGAIN1**及**AGAIN2**第一次出现时编号分别为??0000和??0001，第二次出现时编号分别为??0002和??0003，避免多次重复出现**AGAIN1**和**AGAIN2**标号的错误。



3.16 宏指令

四、宏指令与子程序比较

- 宏指令是采用一个标识符来代替一段程序，它在汇编时要进行宏展开，对程序执行来说并没有增加额外的指令，因此执行速度快，但由于通过宏展开，使宏体在程序中多处出现，因此会占用较多的存储空间。
- 子程序也是一段功能独立的子程序，但它只出现一次，每次需要时，通过CALL指令进行调用，会额外引入子程序调用、子程序返回指令，因此会影响程序的执行速度。



3.16 宏指令

宏指令和子程序优缺点

	优点	缺点
宏指令	<ul style="list-style-type: none">1. 执行速度快2. 设计灵活（可借助参数改变功能）3. 修改方便	占用内存多
子程序	占用内存少	<ul style="list-style-type: none">1. 执行速度慢2. 设计不太灵活