4.4　汇编语言及其程序设计

用指令的助记符、符号地址、标号、伪指令等符号写程序的语言称为汇编语言。
用这种汇编语言书写的程序叫做汇编语言源程序或称源程序。
把汇编语言源程序翻译成在机器上能执行的机器语言程序（目的代码程序）的过程叫做汇编。
完成汇编过程的系统程序为汇编程序。

4.4.1 汇编语言的语句格式

［标号］ 指令助记符 ［操作数］ ［;注解］

例如：

```
NEXT:  MOV  AL，[SI]     ；读（DS:SI）内存数据
```

4.4.2 常数

汇编语言语句中常用的常数可以如下表示：

```
MOV   AL，01000001B    ；二进制数

MOV   AL，65           ；十进制数
MOV   AL，65D          ；十进制数

MOV   AL，41H          ；十六进制数
MOV   AL，0F8H         ；十六进制数（当数字的第一个字符是A～F 时，在字符前添加
                       ;一个数字0，以示和变量的区别）

MOV   AL，'B'          ；字符和字符串
```

4.4.3 伪指令

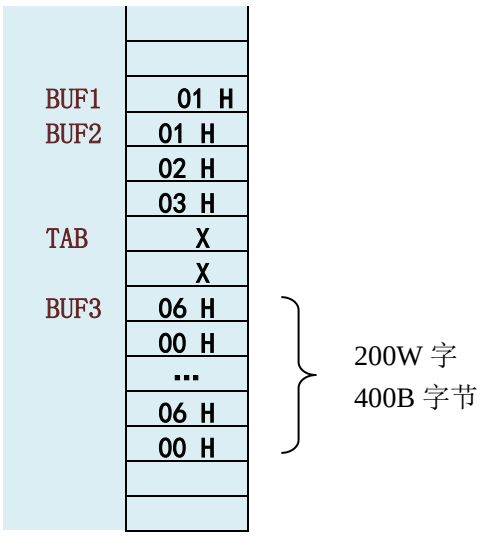
伪指令是用来对汇编程序进行控制，以实现对程序中的数据实现条件转移、列表、存储空间分配等处理。其格式和汇编指令一样，但是一般不产生目的代码，即不直接命令 CPU 去执行什么操作，这就是“伪”的含义。

1、 定义数据伪指令

DB ---- 定义字节
DW ---- 定义字

例如：

```
BUF1   DB   1
BUF2   DB   1，2，3
TAB    DW   ?
BUF3   DW   200 DUP(6)
```



2、符号定义伪指令 EQU

例如：

```
NUM    EQU 200
BUF3   DW  NUM DUP(6)
```

3、定义过程的伪指令 PROC 和 ENDP

在程序设计中，可将具有一定功能的程序段定义为一个过程。
过程由伪指令 PROC 和 ENDP 来定义，其格式为：

```
过程名  PROC  [ NEAR/FAR 类型]  ；近过程/远过程
          ┆    过程体
        RET
过程名 ENDP
```

过程可以用 CALL 指令调用或由 JMP 指令转移到过程。
过程可以嵌套也可以递归使用。

例如：一个过程可定义如下：

```
SOFTDLY  PROC
         PUSH  BX
         PUSH  CX
         MOV   BL，10
DELAY:   MOV   CX，2801
 WAIT:   LOOP  WAIT
         DEC   BL
         JNZ   DELAY
         POP   CX
         POP   BX
         RET
SOFTDLY  ENDP
```

4、段定义伪指令 SEGMENT 和 ENDS

段定义伪指令可将源程序划分成若干段。通常，一个完整的汇编源程序由 3 个段组成，即堆栈段、数据段和代码段。
段定义伪指令一般格式为：

```
段名  SEGMENT
        ┆    段体
段名  ENDS
```

例如：

```
STACK  SEGMENT
   DW  200 DUP(?)
STACK  ENDS
```

```
DATA    SEGMENT
    BUF  DB  1, 2, 3
    TAB  DW  ?
DATA    ENDS
```

5. 汇编结束伪指令 END

END [表达式]

例如：

```
END  START
```

4.4.4 汇编语言的运算符

算术运算符：＋、－、×、 /

逻辑运算符：AND、 OR、 XOR、NOT

关系运算符：EQ、 NE、 LT、 GT、 LE、 GE

取值运算符

属性运算符

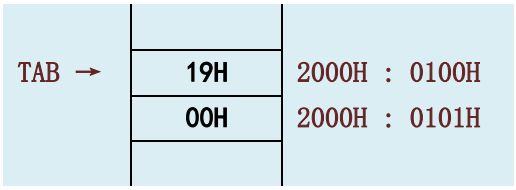
1. 取值运算符 SEG 和 OFFSET

例如：

```
TAB  DW  25

MOV  AX, TAB
  ; AX = 0019H
MOV  AX,    SEG TAB
  ; AX = 2000H
MOV  AX, OFFSET TAB
  ; AX = 0100H
```

区别！



2. 属性运算符 PTR

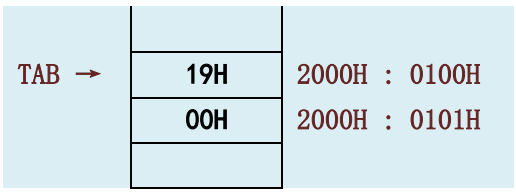
例如：

```
TAB  DW  25

MOV  AX, TAB          ; AX = 0019H
MOV  AL, BYTE PTR TAB ; AL = 19H

MOV  [BX], BYTE PTR TAB ;字节传送, 19H
MOV  [BX], WORD PTR TAB ;字传送, 0019H

MOV  [BX], 10           ; 问题 ?
```



4.4.5　汇编语言源程序的结构

源程序一般都有相同的结构，由代码段、数据段和堆栈段构成。
一个标准的程序结构如下：

```
STACK   SEGMENT
          DB        500 DUP(0)
STACK   ENDS

DATA    SEGMENT
    BUF1    DB  3，12H，
    BUF2    DW  3，1234H
    SUM     DB  100 DUP(0)
    CONT    DB  ?
    TIMES   EQU 100
          ¦
DATA    ENDS

CODE    SEGMENT
MAIN    PROC  FAR
        ASSUME  CS：  CODE，  DS：  DATA，  ES：  DATA，  SS：  STACK
START:  PUSH DS
        MOV  AX,0
        PUSH AX
        MOV  AX，DATA
        MOV  DS，  AX
        MOV  ES，  AX
          ¦
          ¦
        RET
CODE    ENDS
        END    START
```

例：　二进制（无符号字）加法程序。

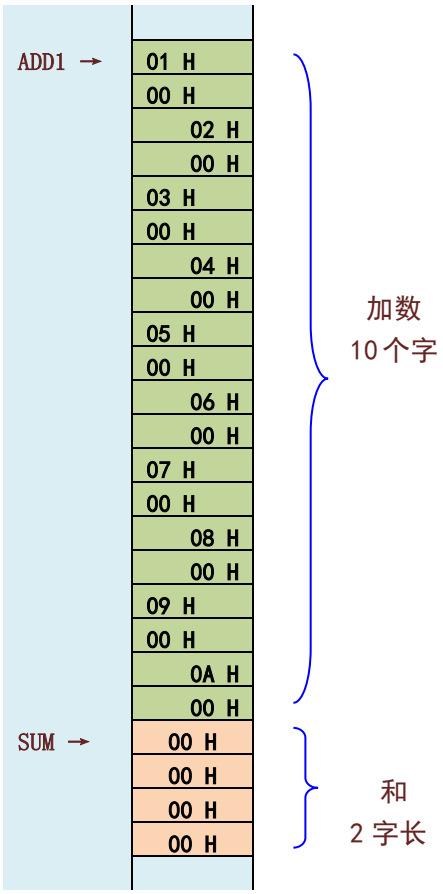
```
ADD1    DW  01H, 02H, 03H, 04H, 05H, 06H, 07H, 08H, 09H, 0AH
SUM     DW  2 DUP(0)

START: MOV   AX,  DATA
       MOV   DS,  AX
       MOV   SI,  OFFSET  ADD1     ; LEA  SI,ADD1
       XOR   AX, AX                ; AX = 00H
       XOR   DX, DX                ; DX = 00H
       MOV   CX, 10
 NEXT: MOV   BX, [SI]

       ADD   AX, BX
       ADC   DX, 0

       INC   SI
       INC   SI
       LOOP  NEXT
       MOV   SUM,   AX
       MOV   SUM+2, DX
       HLT
```

```
      DX , AX
        0 , BX
+     CF
      DX , AX
```

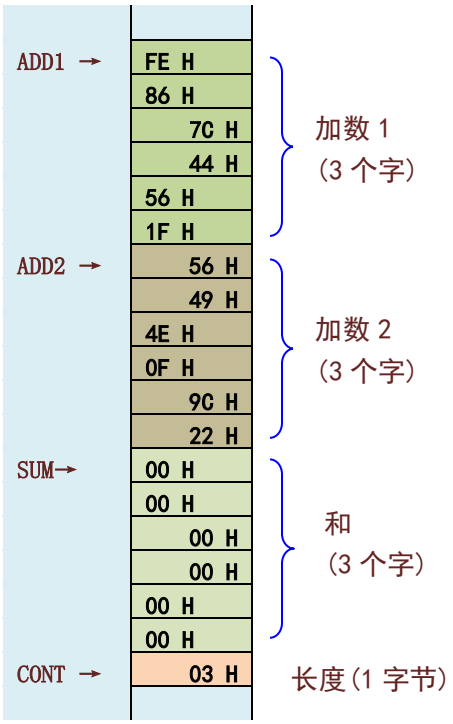


问题：　二进制（有符号字）加法程序。

例：　二进制加法程序（多字长加法）。

```
ADD1    DB   FEH, 86H, 7CH, 44H, 56H, 1FH
ADD2    DB   56H, 49H, 4EH, 0FH, 9CH, 22H
SUM     DW   3 DUP(0)
CONT    DB   3

START: MOV   AX,  DATA
       MOV   DS,  AX              ; 初始化数据段寄存器
       MOV   ES,  AX              ; 初始化辅助段寄存器
       MOV   SI,  OFFSET  ADD1    ; 被加数地址—SI
       MOV   DI,  OFFSET  ADD2    ; 加数地址—DI
       MOV   BX,  OFFSET  SUM     ; 和地址—BX
       MOV   CL,  BYTE PTR CONT
       MOV   CH, 0                ; 初始化相加字长度
       CLC
MADDB1: MOV   AX, [SI]
       ADC   AX, [DI]             ; 16 位相加
       INC   SI
       INC   SI
       INC   DI
       INC   DI
       MOV   [BX],  AX            ; 相加结果送结果单元
       INC   BX
       INC   BX
       LOOP  MADDB1               ; 执行循环
       HLT
```



例：

```
        BUF   DB   100 DUP (?)
        RES   DB   0

    START：MOV    AX，  DATA
           MOV    DS，  AX
           MOV    SI，  OFFSET  BUF
           MOV    RES,[SI]
           INC    SI
           MOV    CX，99
    AGAIN：MOV    AL，[SI]
           CMP    RES，AL
           JBE    NEXT
           MOV    RES，AL
     NEXT：INC    SI
           LOOP   AGAIN
           HLT
```

例：读下面的程序，说明程序完成的功能

```
        START：MOV   DX，DATA
               MOV   DS，DX
               LEA    SI，TAB
               LEA    DI，BUFFER
               MOV   DX，0000H
               MOV   BX，0000H
               MOV   CX，180
        GOON：MOV   AL，[SI]
               CMP    AL，90
               JC     NEXT3
               INC    DH
               JMP    STOR
        NEXT3：CMP    AL，75
               JC     NEXT5
               INC    DL
               JMP    STOR
        NEXT5：CMP    AL，60
               JC    NEXT7
               INC    BH
               JMP    STOR
        NEXT7：INC    BL
        STOR：  INC    SI
               LOOP   GOON
               MOV   [DI]，DH
               MOV   [DI+1]，DL
               MOV   [DI+2]，BH
               MOV   [DI+3]，BL
               HLT
```

例：

```
SENDAT  PROC  FAR
        PUSH  AX
        PUSH  CX
        PUSH  DS
        PUSH  ES
        PUSH  SI
        PUSH  DI
        MOV   AX, SEG BUF1
        MOV   DS, AX
        LEA   SI，BUF1
        MOV   AX, SEG BUF2
        MOV   ES, AX
        LEA   DI，BUF2
        MOV   CX，1024
        CMP   DI, SI
        JBW   LOWER
        STD
        ADD   SI, CX
        DEC   SI
        ADD   DI, CX
        DEC   DI
        JMP   MOVEM
LOWER:  CLD
MOVEM:  REP   MOVSB
        POP  DI
        POP  SI
        POP  ES
        POP  DS
        POP  CX
        POP  AX
        RET
SENDAT  ENDP
```