Lecture 21

Macros

Text: Chapter 22

Macros

- allow one assembly language statement to expand to many statements
- may be used for code or data generation
- allow conditional assembly
- may be put in a library file for repeated/shared use

Example:

Suppose you are often adding three numbers and storing the answer in a fourth, such as

```
MOV    AX,A
ADD    AX,B
ADD    AX,C
MOV    D,AX
```

To avoid typing this, you would prefer to type

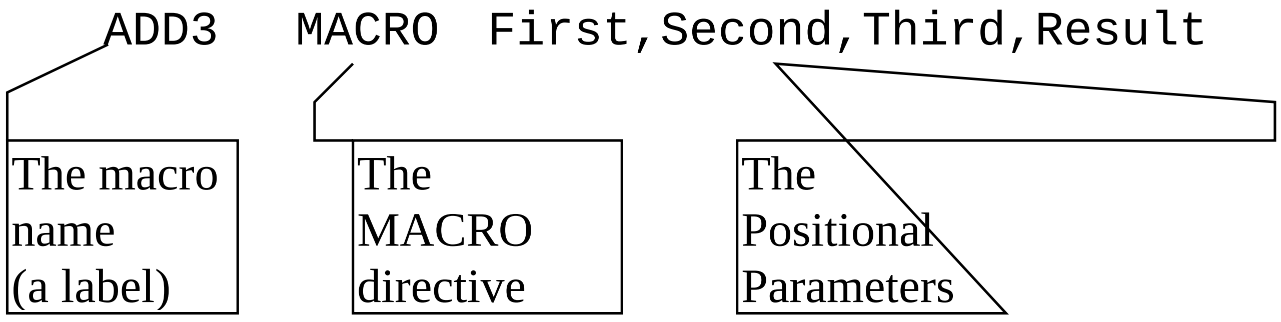
```
ADD3   A,B,C,D  ; D:=A+B+C
```

or

```
ADD3   X,Y,Z,Q  ; Q:=X+Y+Z
```

This macro has a NAME (“ADD3”) and parameters (in the operands field).

To define a macro, you need to give the name and the parameters. The directive “MACRO” is used to do this:



```
    ADD3   MACRO  First,Second,Third,Result
```

Following the macro definition is the body of the macro (what it is to do), and it is concluded with the ENDM (“end Macro”) directive.

```
ADD3   MACRO  First,Second,Third,Result
       MOV    AX,First
       ADD    AX,Second
       ADD    AX,Third
       MOV    Result,AX
       ENDM
```

Note that “First”, Second” and “Third” are NOT variables in your program. They are just place holders for the macro.

The macro must be placed in your program before any defined segment.

```
Turbo Assembler  Version 3.2   Page 1
cs201\macro.ASM
P22MACR1 (EXE) Simple macro to initialize

  1; ----------------------------------------
  2  ADD3   MACRO First,Second,Third,Result
  3          MOV  AX,First
  4          ADD  AX,Second
  5          ADD  AX,Third
  6          MOV  Result,AX
  7  ADD3    ENDM
  8 ;End macro
  9; ----------------------------------------
  10 0000           .MODEL  SMALL
  11 0000           .STACK  64
  12; ---------------------------------------
  13 0000                  .DATA
  14 0000  0003     A      DW  3
  15 0002  0004     B      DW  4
  16 0004  0005     C      DW  5
  17 0006  0006     D      DW  6
  18; ---------------------------------------
  19 0008 .CODE
  20 0000  BEGIN           PROC FAR
  21 0000  B8 0000s        MOV AX,@DATA
  22 0003  8E D8           MOV DS,AX
  23 0005  8E C0           MOV ES,AX
  24;
  25                       ADD3 A,B,C,D
1 26 0007  A1 0000r        MOV  AX,A
1 27 000A  03 06 0006r     ADD  AX,B
1 28 000E  03 06 0004r     ADD  AX,C
1 29 0012  A3 0006r        MOV  D,AX
  30;
  31 0015  CD 21           INT 21H
  32 0017  B8 4C00         MOV AX,4C00H;Exit
```

Repetition Directives

REPT ***expression***

Repeat the statements until the closing ENDM *expression* number of times.

Define all the lower case letters:

```
ASCII=61h       ; 61h is ’a’
REPT   26
DB     ASCII
ASCII=ASCII+1
ENDM
```

IRP ***variable,<arguments>***

Repeat the statements until the closing ENDM as the *variable* takes on each value in the list of *arguments.*

```
IRP  D,<1,5,8,11,12>
DB   D
ENDM
```

IRPC ***variable,string***

Repeat the statements until the closing ENDM as the *variable* takes on the value of each individual character in the *string*.

```
IRPC   Vowels,AEIOU
DB     Vowels
ENDM
```

CONDITIONAL ASSEMBLY

IFxx *condition*

Statements here may be executed depending on the type of IF statement used.

ELSE

Optional; If present, statements here are executed if the above statements are not executed.

ENDIF

Example:

Generate a table of 256 characters containing zeros except for the lower case letters (61h-7Ah):

```
listlow  macro
         n=0
         rept 256
         if (n ge 61h) and (n le 7Ah)
             db  n
         else
             db  0
         endif
         n=n+1
         endm
         endm
```

File: GENDATA.LIB

```
GENDATA  MACRO  STARTER,ENDER,TOTAL
         IF (ENDER) LE (STARTER)
           EXITM
         ENDIF
         IFB  <TOTAL>
           M=256
         ELSE
           M=  TOTAL
         ENDIF
         N=0
         REPT M
         IF (N GE STARTER) AND (N LE ENDER)
            DB  N
         ELSE
            DB  0
         ENDIF
         N=N+1
         ENDM    ; REPT
         ENDM    ; GENDATA
```

```
TITLE  Example in INCLUDE with macro
; -----------------------------------------
     include   c:\bp\bin\cs201\initz.lib
     include   c:\bp\bin\cs201\cond.lib
; -----------------------------------------
              .MODEL  SMALL
              .STACK  64
; -----------------------------------------
              .DATA
    GENDATA  1,10,10
    GENDATA  60h,30h
```

Exercises - Lecture 21

Write macros to do the following:

1. Declare an array of words containing the numbers 1 through *n*, where *n* is passed as a parameter.

2. Suppose a "secret code" is devised where each letter in a message has a number added to its ASCII value. For example, if the number is 2, the message "the cat is black" would appear as "vjg ecv ku dncem" (notice the spaces are unchanged).

   Write a macro that will define an appropriate translation table. The "offset number" (above, 2) should be passed as a parameter.