微机原理上机实验（七）

实验目的和要求：

1、 掌握重复汇编技术的应用；

2、 掌握条件汇编的实现方法。

实验内容：

1、 请编写程序实现以下功能：若某名称为 X 的字符串长度大于 5 时，将下列指令汇编 10 次：ADD AX, AX。

代码：

```
1.  data      segment
2.  x         db 'hello world', '$'
3.  xl        equ $-x
4.  data      ends
5.
6.  code      segment
7.  assume    cs:code, ds:data
8.
9.  main      proc    far
10.           push    ds
11.           sub     ax, ax
12.           push    ax
13.           mov     ax, data
14.           mov     ds, ax
15.
16.           if      xl gt 5
17.           rept    10
18.           add     ax, ax
19.           endm
20.           endif
21.
22.           ret
23. main      endp
24. code      ends
25. end       main
```

说明：

因为在第一遍扫视的时候需要确定条件宏汇编需要的所有数值，xl 则保存字串 x 的长度，大于 5，则重复 10 遍，展开 add ax, ax。

反汇编结果：

```
C:\MASM>debug 0701.exe
-u
076B:0000 1E            PUSH    DS
076B:0001 2BC0          SUB     AX,AX
076B:0003 50            PUSH    AX
076B:0004 B86A07        MOV     AX,076A
076B:0007 8ED8          MOV     DS,AX
076B:0009 03C0          ADD     AX,AX
076B:000B 03C0          ADD     AX,AX
076B:000D 03C0          ADD     AX,AX
076B:000F 03C0          ADD     AX,AX
076B:0011 03C0          ADD     AX,AX
076B:0013 03C0          ADD     AX,AX
076B:0015 03C0          ADD     AX,AX
076B:0017 03C0          ADD     AX,AX
076B:0019 03C0          ADD     AX,AX
076B:001B 03C0          ADD     AX,AX
076B:001D CB            RETF
076B:001E 0000          ADD     [BX+SI],AL
-_
```

2、 请编写程序使汇编程序根据 SIGN 中的不同值分别产生不同的指令：如果(SIGN)=0，则用字节变量 DIVD 中的无符号数除以字节变量 SCALE；如果(SIGN)=1，则用字节变量 DIVD 中的带符号数除以字节变量 SCALE，结果都存放在字节变量 RESULT 中。

代码：

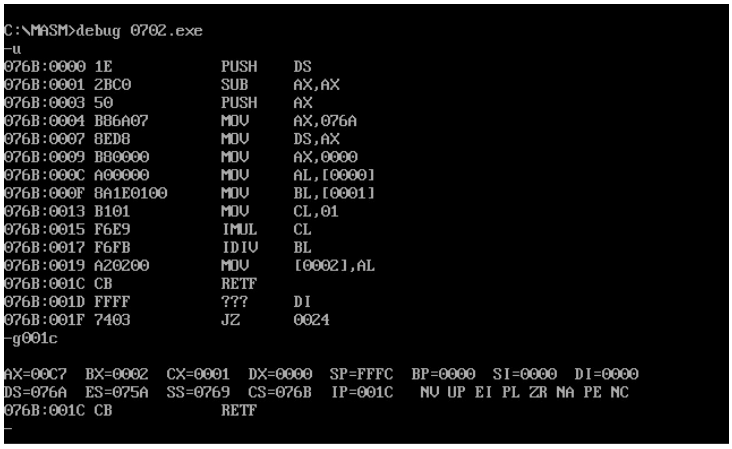
```
1.  sign       equ     1
2.
3.  data       segment
4.  divd       db 10001110b
5.  scale      db 2
6.  result     db ?
7.  data       ends
8.
9.  code       segment
10. assume     cs:code, ds:data
11.
12. main       proc    far
13.            push    ds
14.            sub     ax, ax
15.            push    ax
16.            mov     ax, data
17.            mov     ds, ax
18.
19.            mov     ax, 0
20.            mov     al, divd
21.            mov     bl, scale
22.            if  sign
23.                mov cl, 1
24.                imul cl
25.                idiv bl
26.            else
27.                div bl
28.            endif
29.            mov     result, al
30.
31.            ret
32. main       endp
33. code       ends
34. end        main
```

说明：

将被除数 divd 移至 al，除数 scale 移至 bl，判断 sign 是否置为 0：为 0，则直接无符号除法 div bl；为 1，则通过符号乘法和+1 相乘，再有符号除法 idiv bl。商位于 al 中，移至 result。

运行结果：

sign = 1, divd = -114 (10001110B, 8EH), scale = 2 → result = -57 (C7H)



sign = 1, divd = 14 (00001110B, 0EH), scale = 2 → result = 7 (07H)

```
C:\MASM>debug 0702.exe
-u
076B:0000 1E            PUSH    DS
076B:0001 2BC0          SUB     AX,AX
076B:0003 50            PUSH    AX
076B:0004 B86A07        MOV     AX,076A
076B:0007 8ED8          MOV     DS,AX
076B:0009 B80000        MOV     AX,0000
076B:000C A00000        MOV     AL,[0000]
076B:000F 8A1E0100      MOV     BL,[0001]
076B:0013 B101          MOV     CL,01
076B:0015 F6E9          IMUL    CL
076B:0017 F6FB          IDIV    BL
076B:0019 A20200        MOV     [0002],AL
076B:001C CB            RETF
076B:001D FFFF          ???     DI
076B:001F 7403          JZ      0024
-g001c

AX=0007  BX=0002  CX=0001  DX=0000  SP=FFFC  BP=0000  SI=0000  DI=0000
DS=076A  ES=075A  SS=0769  CS=076B  IP=001C   NV UP EI PL ZR NA PE NC
076B:001C CB            RETF
-_
```

sign = 0, divd = 142 (10001110B, 8EH), scale = 2 → result = 71 (47H)

```
C:\MASM>debug 0702.exe
-u
076B:0000 1E            PUSH    DS
076B:0001 2BC0          SUB     AX,AX
076B:0003 50            PUSH    AX
076B:0004 B86A07        MOV     AX,076A
076B:0007 8ED8          MOV     DS,AX
076B:0009 B80000        MOV     AX,0000
076B:000C A00000        MOV     AL,[0000]
076B:000F 8A1E0100      MOV     BL,[0001]
076B:0013 F6F3          DIV     BL
076B:0015 A20200        MOV     [0002],AL
076B:0018 CB            RETF
076B:0019 83C404        ADD     SP,+04
076B:001C 3DFFFF        CMP     AX,FFFF
076B:001F 7403          JZ      0024
-g0018

AX=0047  BX=0002  CX=0029  DX=0000  SP=FFFC  BP=0000  SI=0000  DI=0000
DS=076A  ES=075A  SS=0769  CS=076B  IP=0018   NV UP EI PL ZR NA PE NC
076B:0018 CB            RETF
-_
```

实验思考：

1、 IRP DUMMY, <ARGUMENT LIST>和 IRPC DUMMY, STRING，将哑元按照参数列表顺序取值重复。区别是 IRPC 自变量只能是字符串。

2、 IF EXPRESSION　　汇编程序求出表达式的值,如此值不为 0 则满足条件；

IFE EXPRESSION　　如求出表达式的值为 0 则满足条件；

IFDEF SYMBOL　　如符号已在程序中定义，或者已用 EXTRN 伪操作说明该符号是在外部定义的，则满足条件；

IFNDEF SYMBOL　　如符号未定义或未通过 EXTRN 说明为外部符号则满足条件；

IFB <ARGUMENT>　如自变量为空则满足条件；

IFNB<ARGUMENT>　如自变量不为空则满足条件；

IFIDN <ARGU-1>,<ARGU-2>　　如果字符串<ARG-1>和字符串<ARG-2>相同，则满足条件；

IFDIF <ARGU-1>,<ARGU-2>　　如果字符串<ARG-1>和字符串<ARG-2>不相同，则满足条件。

思考心得：

类似于 C 语言，很方便。

FUCK YOU BITCH!