

实验五

子程序设计实验

1.实验目的:

掌握子程序设计的基本方法。

2.实验要求:

求一个多字节数据的绝对值及求两个多字节数据的和的功能分别用子程序来实现；分别用三组数据调试程序，检验程序的正确性：

- ① 两个多字节数据都是正数；
- ② 两个多字节数据都是负数；
- ③ 两个多字节数据一正一负。

3.实验内容:

已知两个长度相等的带符号的多字节数据分别存放在内存 DATA1 和 DATA2 开始的连续单元中，数据长度存放在 LEN 单元。编制程序，计算两个数据的绝对值之和，将结果存放在 SUM 开始的连续单元中。

4.源程序:

; 绝对值，多字节求和

assume ss:stacksg, cs:codesg, ds:datasg

datasg segment

DATA1 db 0ffH, 0ffH, 0ffH, 0ffH, 7fH

DATA2 db 0ffH, 0ffH, 0ffH, 0ffH, 0ffH

SUM db 0, 0, 0, 0, 0, 0

LEN db 5

datasg ends

stacksg segment

db 16 dup(0)

stacksg ends

codesg segment

; data1 的开始数据位 si,长度在 cx 中，数据段在 ds

; data2 的开始数据位 di,进入前需要清除 CF

; sum 的开始数据位 bx

add_2 PROC

push si

push di

push bx; 用于指向 sum

push cx

r:

mov al, [si]

adc al, [di]

```

    mov [bx], al

    inc bx
    inc si
    inc di
    loop r
    adc byte ptr [bx], 0 ; 进位
return:
    pop cx
    pop bx
    pop di
    pop si
    ret
add_2 ENDP
; 处理一个数据
; 长度存在 bx, 起始地址在 si 中, 数据段地址在 ds
; 程序执行后, 源地址为其绝对值, 进入前要求 CF 为 1
abs PROC
    push ax
    push cx
    push si

    mov al, [si + bx-1]
    and al, al
    mov cx, bx
    jns retu
    ; body
    STC
s:
    not byte ptr [si]
    adc byte ptr [si], 0
    inc si
    loop s
retu:
    pop si
    pop cx
    pop ax
    ret
abs ENDP

start:
    mov ax, stacksg
    mov ss, ax
    mov sp, 16

```

```

mov ax, datasg
mov ds, ax

mov bx, 0
mov bl, LEN
LEA si, data1
call abs

mov bx, 0
mov bl, LEN
LEA si, data2
call abs

LEA si, data1
LEA di, data2
LEA bx, SUM
mov cx, 0
mov cl, [LEN]
CLC
call add_2

mov ax, 4c00h
int 21h
codesg ends
end start

```

5.运行结果:

程序成功运行，第一组数据，原始数据与结果如下：

```

-d ds:0 32
076C:0000 FF 11 22 35 06 FF 11 77-88 02 00 00 00 00 00 00
076C:0010 05 00 00 00 00 00 00 00-00 00 00 00 00 00 00
076C:0020 00 00 00 00 00 00 6C 07-00 00 44 00 6F 07 A4 01
076C:0030 56 57 53

```

```

-d ds:0 32
076C:0000 FF 11 22 35 06 FF 11 77-88 02 FE 23 99 BD 08 00
076C:0010 05 00 00 00 00 00 00 00-00 00 00 00 00 00 00
076C:0020 0A 00 00 00 05 05 EB 0C-6C 16 6F 07 D8 C1 FF FF
076C:0030 56 57 53

```

第二组数据：使用了 -1 和 -1 进行验证，最后程序中取绝对值和最后的加和都正确

```

-d ds:0
076C:0000 FF FF FF FF FF FF FF-FF FF 00 00 00 00 00 00
076C:0010 05 00 00 00 00 00 00 00-00 00 00 00 00 00 00
-d ds:0
076C:0000 01 00 00 00 00 01 00 00-00 00 02 00 00 00 00
076C:0010 05 00 00 00 00 00 00 00-00 00 00 00 00 00 00

```

第三组数据：-1 和 5 进行验证

```
-d ds:0
976C:0000  05 00 00 00 00 01 00 00-00 00 06 00 00 00 00 00
976C:0010  05 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00
```

6.心得体会:

实验五使我明白：清晰的接口约定至关重要，要按照要求在调用前满足需满足的状态。以及必须保护调用者环境，子程序内部使用的寄存器在开始时应 PUSH 保存，在结束 RET 前 POP 恢复，避免干扰主程序逻辑。并且还需要细致管理和利用标志位。