**北京邮电大学软件学院**

**2013-2014学年第一学期实验报告**

课程名称: 汇编语言设计实践

实验名称： 汇编程序中的子程序设计

姓 名：

学 号：

2014年10月20日

一 . 实验源代码 ：

data segment

var1 db 82h, 54h, 9eh, 66h, 0c7h, 02h, 0b2h, 14h, 54h, 0c2h ;变量var1保存长度为 10 个字节的有符号数 1

var2 db 2bh, 8ch, 4ch, 0c2h, 12h,0abh,7ah, 35h, 0a5h, 41h;变量var2保存长度为10个字节的有符号数 2

len equ $-var2 ;len计算每个加数所占字节数，len 不占用内存

sum db len dup(0) ;用于保存运算结果

ovr db '?' ;用于保存溢出标志

data ends

stack1 segment stack ;子程序设计必须定义堆栈段

dw 40 dup(0)

stack1 ends

code segment

assume cs:code,ds:data,ss:stack1

start:

mov ax,data

mov ds,ax

lea si,var1 ;通过堆栈传递入口参数

push si

lea si,var2

push si

mov si,len

push si

lea si,sum

push si

call longadd ;调用子程序

pop dx ;出口参数出栈

mov ovr,dl ;保存出口参数

mov ah,4ch

int 21h

longadd proc

push bp ;BP作为现场加以保护

mov bp,sp ;获取当前栈顶位置

pushf ;保护CPU现场

push ax

push bx

push cx

push si

push di

mov bx,[bp+4] ;取得保存运算结果的起始偏移量

mov cx,[bp+6] ;取得被加数、加数的长度

mov di,[bp+8] ;取得加数的起始偏移量

mov si,[bp+10] ;取得被加数的起始偏移量

clc ;清除cf标志，保证第一次执行ADC时等同于ADD

s:

mov ax,[si]

adc ax,[di] ;完成当前字的加法，引入了上次加法的进位

pushf ;保护OF标志

mov [bx],ax ;当前字的计算结果保存的指定位置

add si,2 ;修改SI,DI,BX，使他们指向下一个字

add di,2

add bx,2

popf ;恢复OF标志

loop s ;计数循环，完成所有字单元的加法后退出循环

jo overflow ;判断最后一次运算是否溢出

mov [bp+10],0 ;无溢出则将0送至DL保存，传递出口参数

jmp restore

overflow:

mov [bp+10],0ffh ;有溢出则将0FFH送至DL保存，传递出口参数

restore:

pop di ;恢复CPU现场

pop si

pop cx

pop bx

pop ax

popf

pop bp

ret 6 ;返回主程序，仅清除3个入口参数，因最先入栈的入口参数已被出口参数覆盖

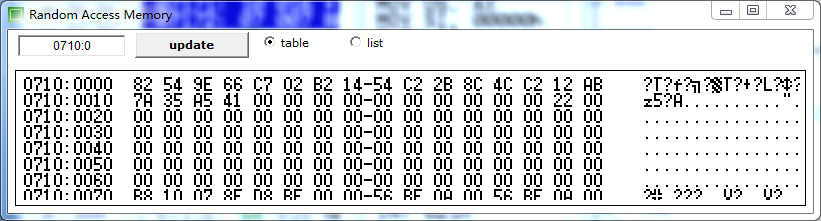
longadd endp

code ends

end start

二 .结果截图 ：

1、程序运行前DATA所在内存的值：



2、程序运行后DATA所在内存的值：

