**模拟计算器设计与实现**

**一、实验要求**

1.基本要求：实现多项式运算，如：Z=(X+Y)×8÷2,其中X,Y可自由设定值，验证结果，给出程序和调试结果图（截屏）。

要求：用x86软件平台实现Z=(X+Y)×8÷2多项式功能。

2.提高要求：在“基本要求”中内容的基础上实现计算器中任意两数+、-、×、÷功能及其扩展，可采用子程序编制加减乘除，也可采用宏命令。

3.人机交互实现：针对“提高要求”中内容，根据所选择的开发平台，完成下面的屏幕显示功能。

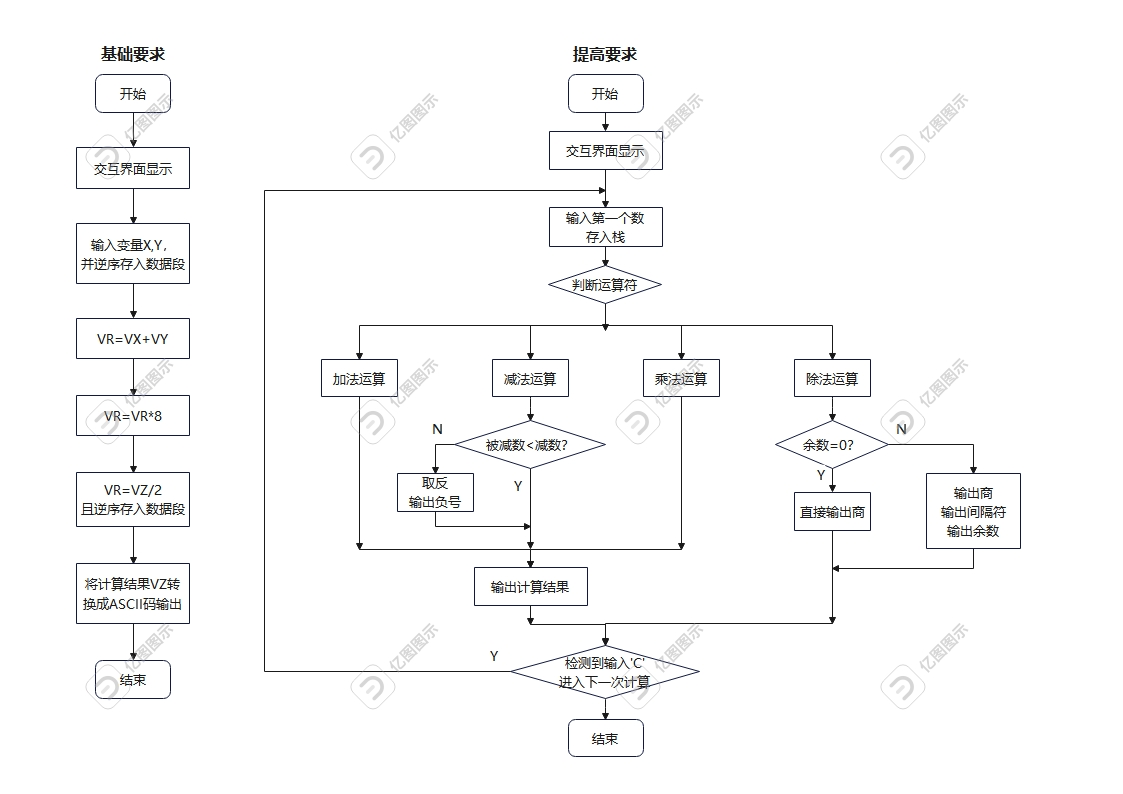
使用DOS功能调用增加屏幕的输入显示，输出显示功能。

**二、设计思路**

1. 基本要求：
2. 提高要求：
3. **程序框架**

**四、实验结果**

【程序流程图】



1.基本要求：

【源代码】

;--------------------------------------------------------------------------------

;可实现多项式Z=(X+Y)\*8/2在10位数以内的运算，不考虑除法溢出问题

;--------------------------------------------------------------------------------

CRLF MACRO ;显示回车和换行

MOV AH,02H

MOV DL,0DH

INT 21H

MOV AH,02H

MOV DL,0AH

INT 21H

ENDM

STACK1 SEGMENT

DB 256 DUP(?)

STACK1 ENDS

DATA SEGMENT

VX DB 10 DUP(?) ;预置变量X,Y,Z的存储空间

VY DB 10 DUP(?)

VR DB 12 DUP(?) ;中间变量

VZ DB 13 DUP(?)

SSE DB 50 DUP('+'),0DH,0AH,'$';交互界面分隔

SPO DB 'Polynomial:Z=(X+Y)\*8/2',0DH,0AH,'$';提示用户默认的多项式运算

SVX DB 'X=',0DH,0AH,'$' ;提示用户输入X

SVY DB 'Y=',0DH,0AH,'$' ;提示用户输入Y

SVZ DB 'Z=',0DH,0AH,'$' ;输入结果Z

DATA ENDS

CODE SEGMENT

ASSUME CS:CODE,DS:DATA

START: MOV AX,DATA ;数据段

MOV DS,AX

MOV AX,STACK1 ;栈段

MOV SS,AX

MOV AH,09H ;显示分割线

LEA DX,SSE

INT 21H

MOV AH,09H ;显示运算式

LEA DX,SPO

INT 21H

;输入变量X,并逆序存入数据段

MOV AH,09H ;提示用户输入变量X

LEA DX,SVX

INT 21H

MOV BX,0 ;统计字节长

MOV CX,10 ;设置循环次数上限

LEA SI,VX ;指针指向变量存储首地址

LIX1: MOV AH,08H ;逐字输入

INT 21H

CMP AL,0DH ;检测，输入回车符表示输入完成

JZ JIENDX ;完成，跳转

MOV DL,AL ;显示

MOV AH,02H

INT 21H

SUB AL,30H ;将ASCII码转换为十进制数，方便运算

PUSH AX ;入栈

INC BX ;计数

LOOP LIX1

JIENDX: CRLF

MOV CX,BX ;设置循环次数为字节长

LIX2: POP AX ;出栈

MOV [SI],AL ;未完成，则将字符存入

INC SI ;指针步进

LOOP LIX2

;输入变量Y,并逆序存入数据段

MOV AH,09H ;提示用户输入变量Y

LEA DX,SVY

INT 21H

MOV BX,0 ;统计字节长

MOV CX,10 ;设置循环次数上限

LEA SI,VY ;指针指向变量存储首地址

LIY1: MOV AH,08H ;逐字输入

INT 21H

CMP AL,0DH ;检测，输入回车符表示输入完成

JZ JIENDY ;完成，跳转

MOV DL,AL ;显示

MOV AH,02H

INT 21H

SUB AL,30H ;将ASCII码转换为十进制数，方便运算

PUSH AX ;入栈

INC BX ;计数

LOOP LIY1

JIENDY: CRLF

MOV CX,BX ;设置循环次数为字节长

LIY2: POP AX ;出栈

MOV [SI],AL ;未完成，则将字符存入

INC SI ;指针步进

LOOP LIY2

;计算及输出

CALL CADD ;加法计算

CALL CMUL ;乘法计算

CALL CDIV ;除法计算

CALL TRANS ;结果转换为ASCII码

CALL ORESULT ;输出计算结果

MOV AH,4CH

INT 21H

CADD PROC NEAR ;加法计算子程序

PUSH SI ;保护现场

PUSH DI

PUSH BX

PUSH CX

PUSH AX

LEA SI,VX ;指针SI指向变量X首地址

LEA DI,VY ;指针DI指向变量Y首地址

LEA BX,VR ;指针BX指向变量R首地址

MOV CX,10 ;设置循环次数

XOR AX,AX ;AX清零

CLC

LADD: MOV AL,[SI] ;逐位相加

ADC AL,[DI]

AAA

MOV [BX],AL ;存入结果

INC SI ;指针步进

INC DI

INC BX

LOOP LADD

JNC JEND

MOV [BX],01H ;进位计入

JEND: POP AX ;恢复现场

POP CX

POP BX

POP DI

POP SI

RET

CADD ENDP

CMUL PROC NEAR ;乘法计算子程序

PUSH DX ;保护现场

PUSH SI

PUSH BX

PUSH CX

PUSH AX

MOV DL,08H ;乘数

LEA SI,VR ;指针SI指向变量R首地址

MOV BL,00H ;存放进位

MOV CX,12 ;设置循环次数

CLC

LMUL: MOV AL,[SI]

MUL DL ;逐位相乘

AAM

ADD AL,BL ;加入进位

AAA

MOV [SI],AL ;结果低四位存入AL

MOV BL,AH ;进位存入BL

XOR AX,AX ;AX清零

INC SI ;指针步进

LOOP LMUL

MOV [SI],BL ;存入进位

POP AX ;恢复现场

POP CX

POP BX

POP SI

POP DX

RET

CMUL ENDP

CDIV PROC NEAR ;除法计算子程序

PUSH DX ;保护现场

PUSH DI

PUSH SI

PUSH AX

PUSH CX

MOV DL,02H ;存入除数

LEA DI,VZ ;指针DI指向变量Z首地址

LEA SI,VZ ;指针SI指向变量R末地址

DEC SI

MOV AH,[SI] ;AH装入

MOV AL,00H ;AL清零

DEC SI

INC DI

MOV CX,11 ;设置循环次数(将第0、1位装入AX，避免除法溢出）

CLC

LDIV: MOV AL,[SI] ;取出变量Z的字节

AAD

DIV DL ;逐位相除

MOV [DI],AL ;结果低四位存入AL

XOR AL,AL ;AL清零

DEC SI ;指针步退

INC DI ;指针步进

LOOP LDIV

POP CX ;恢复现场

POP AX

POP SI

POP DI

POP DX

RET

CDIV ENDP

TRANS PROC NEAR ;将加法计算结果转换为ASCII码

PUSH SI ;保护现场

PUSH AX

LEA SI,VZ ;指针指向存储首地址

MOV CX,12 ;设置循环次数

LTRANS: MOV AL,[SI]

ADD AL,30H

MOV [SI],AL

INC SI ;指针步进

LOOP LTRANS

MOV [SI],'$' ;字符串结束标识符存入

POP AX ;恢复现场

POP SI

RET

TRANS ENDP

ORESULT PROC NEAR ;将计算结果输出到屏幕

PUSH AX ;保护现场

PUSH DX

MOV AH,09H ;提示输出结果

LEA DX,SVZ

INT 21H

MOV CX,12 ;设置循环次数

LEA SI,VZ ;指针指向结果Z

LO: MOV DL,[SI]

MOV AH,02H ;输出结果

INT 21H

INC SI ;指针步进

LOOP LO

CRLF

POP DX ;恢复现场

POP AX

RET

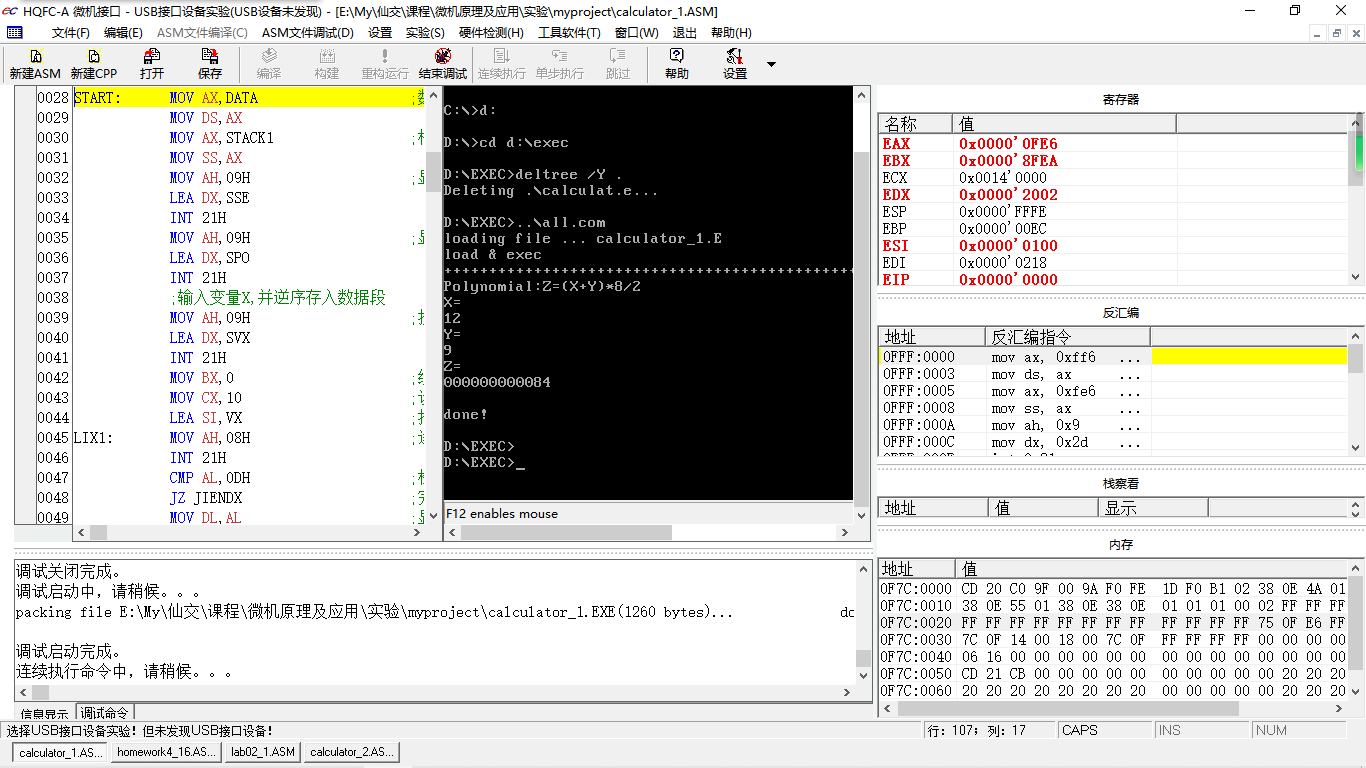
ORESULT ENDP

CODE ENDS

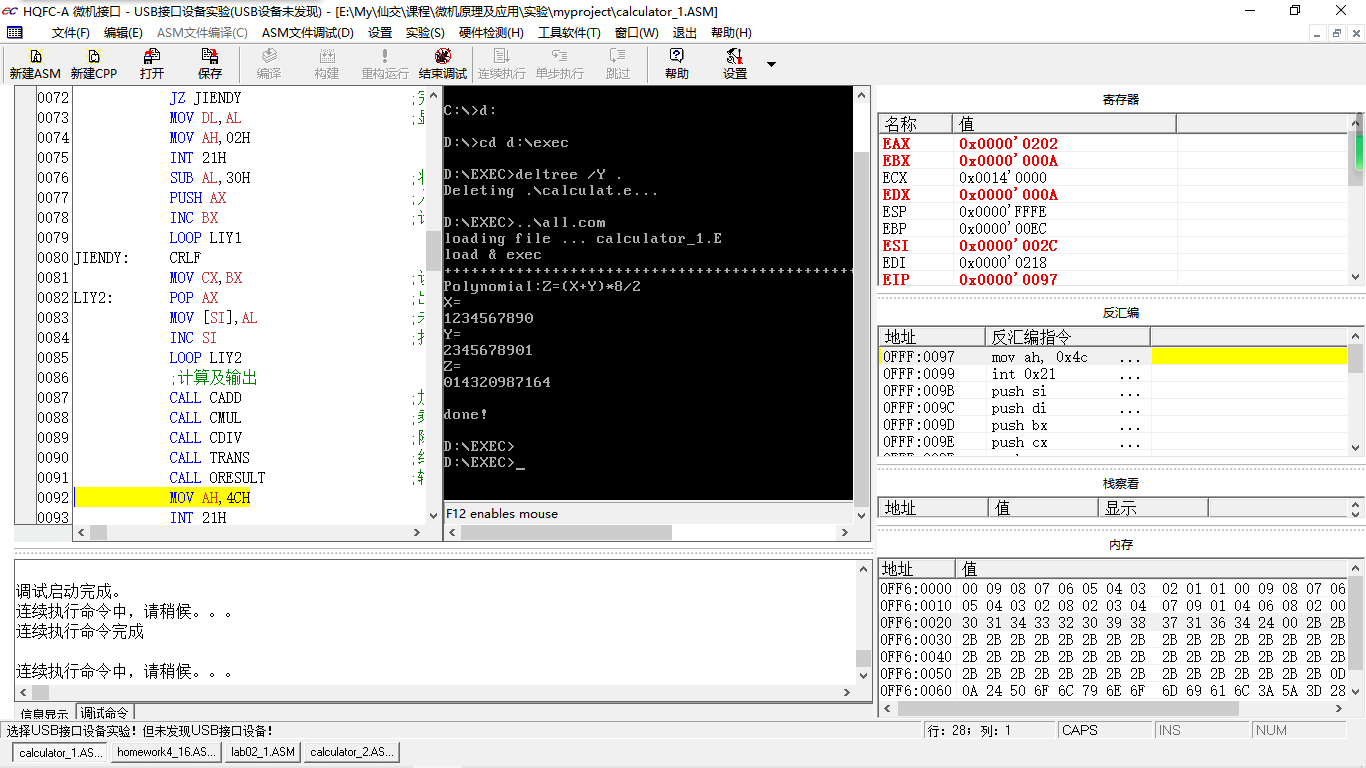
END START

【运行结果】

X=12,Y=9,得到Z=84



X=1234567890,Y=2345678901,得到Z=14320987164



2.提高要求：

【源代码】

;--------------------------------------------------------------------------------

;通过入栈/出栈操作记录输入数据，实现两个数之间的加、减、乘、除运算，其中

;加法：两位加数、和范围均为：0~65536

;减法：被减数、减数、差范围均为：-32767~32767

;乘法：两位乘数、积范围均为：0~65536

;除法：被除数、除数、商、余数范围均为：0~32767

;'='--输出结果，'C'--归零

;--------------------------------------------------------------------------------

CRLF MACRO ;显示回车和换行

MOV AH,02H

MOV DL,0DH

INT 21H

MOV AH,02H

MOV DL,0AH

INT 21H

ENDM

STACK1 SEGMENT

DB 256 DUP(?)

STACK1 ENDS

DATA SEGMENT

RE DB 0,0,0,0,0,'$' ;计算结果的ASCII码

DIVI DW 10000,1000,100,10,1 ;万、千、百、十、个

SE DB 6 DUP('.'),'$' ;除法间隔符

SSE DB 50 DUP('+'),0DH,0AH,'$';交互界面分隔

STIP DB 'My Calculator:',0DH,0AH,'$';提示用户输入计算式

DATA ENDS

CODE SEGMENT

ASSUME CS:CODE,DS:DATA

START: MOV AX,DATA ;数据段

MOV DS,AX

MOV AX,STACK1 ;栈段

MOV SS,AX

MOV AH,09H ;显示分割线

LEA DX,SSE

INT 21H

MOV AH,09H ;显示提示词

LEA DX,STIP

INT 21H

REFRESH: XOR AX,AX ;AX清零

CALL INPUT ;输入第一个数

PUSH BX ;存入栈

;判断运算符，跳转

CMP AL,'+'

JZ CADD

CMP AL,'-'

JZ CSUB

CMP AL,'\*'

JZ CMUL

CMP AL,'/'

JZ CDIV

CADD: CALL INPUT ;输入第二个数

POP AX ;第一个数取出

ADD AX,BX ;相加

JMP NEXT ;跳出

CSUB: CALL INPUT ;输入第二个数

POP AX ;第一个数取出

CMP AX,BX ;判断被减数与减数的大小关系

JL LESS ;小于，跳转

SUB AX,BX ;大于，直接相减

JMP NEXT

LESS: SUB AX,BX ;相减为负则取反

NEG AX

PUSH AX ;保护AX

MOV DL,'-' ;输出负号

MOV AH,02H

INT 21H

POP AX ;送回AX

JMP NEXT ;跳出

CMUL: CALL INPUT ;输入第二个数

POP AX ;第一个数取出

MUL BX ;相乘

JMP NEXT ;跳出

CDIV: CALL INPUT ;输入第二个数

POP AX ;第一个数取出

DIV BX ;相除

CMP DX,0 ;检测余数

JZ NEXT ;直接跳出

PUSH DX ;保护DX

CALL OUTPUT ;输出商

MOV AH,09H ;显示间隔符

LEA DX,SE

INT 21H

POP DX ;送回DX

MOV AX,DX ;输出余数

CALL OUTPUT

JMP REFRESH ;跳出，刷新

NEXT: CALL OUTPUT ;结果输出

JMP REFRESH ;跳出，刷新

MOV AH,4CH

INT 21H

INPUT PROC NEAR ;输入子程序

MOV BX,0 ;记录已输入的数值，初始化BX清零

INNUM: MOV AH,01H ;字符输入

INT 21H

CMP AL,'C' ;归零跳转

JZ CLEAR

CMP AL,'+' ;运算符跳出

JZ EXIT

CMP AL,'-'

JZ EXIT

CMP AL,'\*'

JZ EXIT

CMP AL,'/'

JZ EXIT

SUB AL,30H ;ASCII码转换为十进制

JL EXIT ;排除0~9以外的字符

CMP AL,9

JG EXIT

CBW ;按数位依次将输入的运算数存入

XCHG AX,BX ;已输入数值送入

MOV CX,10 ;十进制

MUL CX ;十进制下左移一位

XCHG AX,BX ;送回

ADD BX,AX ;当前输入位加入

JMP INNUM ;跳转输入下一位

CLEAR: CRLF ;回车换行

XOR AX,AX ;寄存器AX,BX清零

XOR BX,BX

JMP INNUM ;跳转输入下一位

EXIT: RET

INPUT ENDP

OUTPUT PROC NEAR ;输出子程序

MOV SI,OFFSET RE ;指针

MOV DI,OFFSET DIVI

MOV CX,5 ;设置循环次数

LTRANS: MOV DX,0 ;DX清零,存储未转换数值

DIV WORD PTR [DI] ;位数

ADD AL,30H ;十进制转换为ASCII码

MOV BYTE PTR [SI],AL ;存储

INC SI ;指针步进

ADD DI,2

MOV AX,DX ;未转换值送回

LOOP LTRANS

MOV SI,OFFSET RE ;指针

MOV CX,4 ;设置循环次数

LPRINT: CMP BYTE PTR [SI],30H ;从第一个有效位开始输出

JNZ PRINT

INC SI ;指针步进

LOOP LPRINT

PRINT: MOV DX,SI ;计算结果输出

MOV AH,09H

INT 21H

RET

OUTPUT ENDP

CODE ENDS

END START

【运行结果】

