**电工电子实验中心**

**实验报告**

课程名称： 微机原理与接口技术实验

实验名称： 四则运算

姓名： 陈力以 学号: 161610108

评定成绩： 审阅教师：

实验时间： 2018.10.22

南京航空航天大学

# 实验目的要求

1．熟悉汇编语言程序的框架结构，掌握顺序结构的编程方法。

2．熟悉 Tddebug 调试环境和 Turbo Debugger 的使用。

3．理解 X86 内存数据的组织方式。

4．理解基本的内存寻址方式。

# 实验任务

完成 32 位无符号数的加法、减法，16 位乘以 16 位，32 位除以 16 位除法的四则运算练习。

# 实验代码（写出自己补全的代码，包含适当注释）

DATA SEGMENT

A DW 1234H, 5678H

B DW 0FEDCH, 123AH

C DW 2 DUP (0)

AD DW 0FEDCH, 0BA98H

BD DW 1234H, 789AH

CD DW 2 DUP (0)

A1 DW 0D678H

B1 DW 0012H

C1 DW 2 DUP (0)

A2 DW 5678H, 0234H

B2 DW 0F234H

C2 DW 2 DUP (0)

DATA ENDS

；用0初始化C,CD,C1,C2

STACK1 SEGMENT STACK

DB 100 DUP(0)

STACK1 ENDS

CODE SEGMENT

ASSUME CS:CODE,DS:DATA,SS:STACK1

START PROC FAR

PUSH DS

MOV AX,0

PUSH AX

MOV AX,DATA

MOV DS,AX

MOV AX,A ;add

ADD AX,B

MOV C,AX

MOV AX,A+2

ADC AX,B+2

MOV C+2,AX

MOV AX,AD ;sub

SUB AX,BD

MOV CD,AX

MOV AX,AD+2

SUB AX,BD+2

MOV CD+2,AX

；减法同加法

MOV AX,A1 ;mul

MUL B1

MOV C1,AX

MOV C1+2,DX

；乘法高位存在DX，低位存在AX

MOV AX,A2 ;div

MOV DX,A2+2

DIV B2

MOV C2,AX

MOV C2+2,DX

；被除数高位在DX，低位在AX，商在AX中，余数在DX中

POP AX ;balance

POP DS

RET

START ENDP

CODE ENDS

END START

# 探究内容（选做）

1.理解 X86 内存数据的组织方式:小端模式。分别写出加、减、乘、除 4 个运算的运算数， 及其在内存中的存放方式;

加法：1234H+FEDCH 5678H+123AH



减法：FEDCH-1234H BA98H-789AH

乘法：D678H\*0012H



除法：02345678H/F234H



2. 若需进行有符号数的运算，需要注意什么问题?如何实现?

在乘除法时指令要改为idiv和imul，有无符号的乘除法运算方法不同，需要区分。

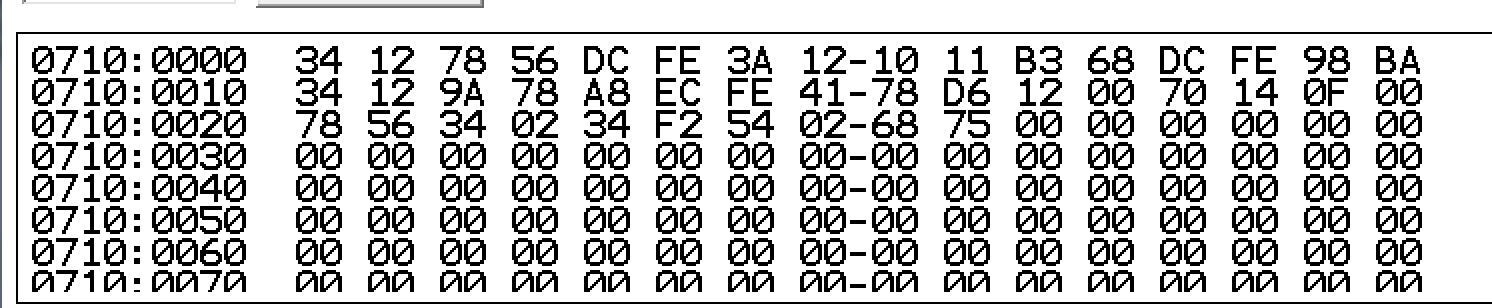
3.上述实验中，我们在 80X86 的实模式中实现了 32 位的四则运算。例如，乘法运算为 16位乘以 16 位，运算结果为 32 位。请思考如何利用 32 位指令，实现 64 位的四则运算?

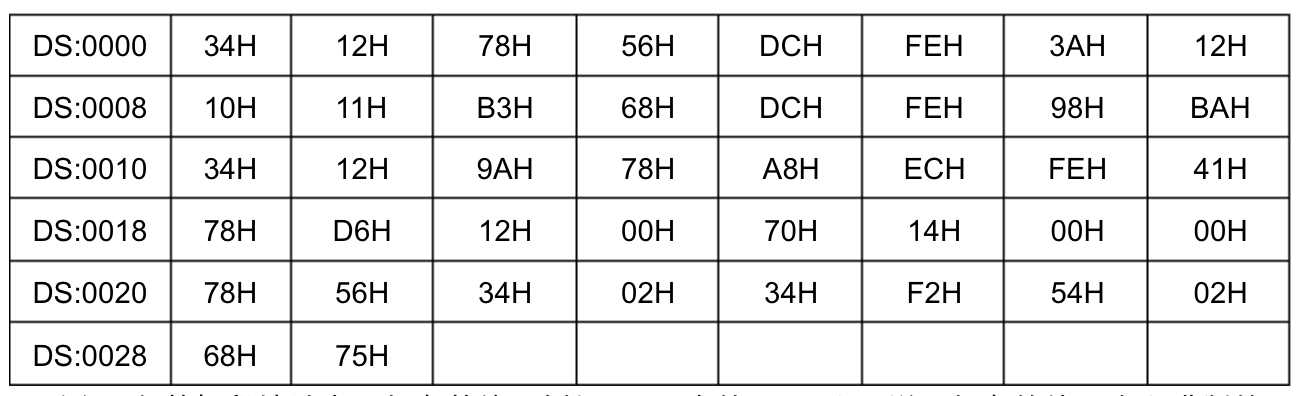
加减法可用adc和sbb做带进借位的加减法，先用add和sub做低位加减法，再用adc和sbb做高位加减法，低32位和高32位分开存储。

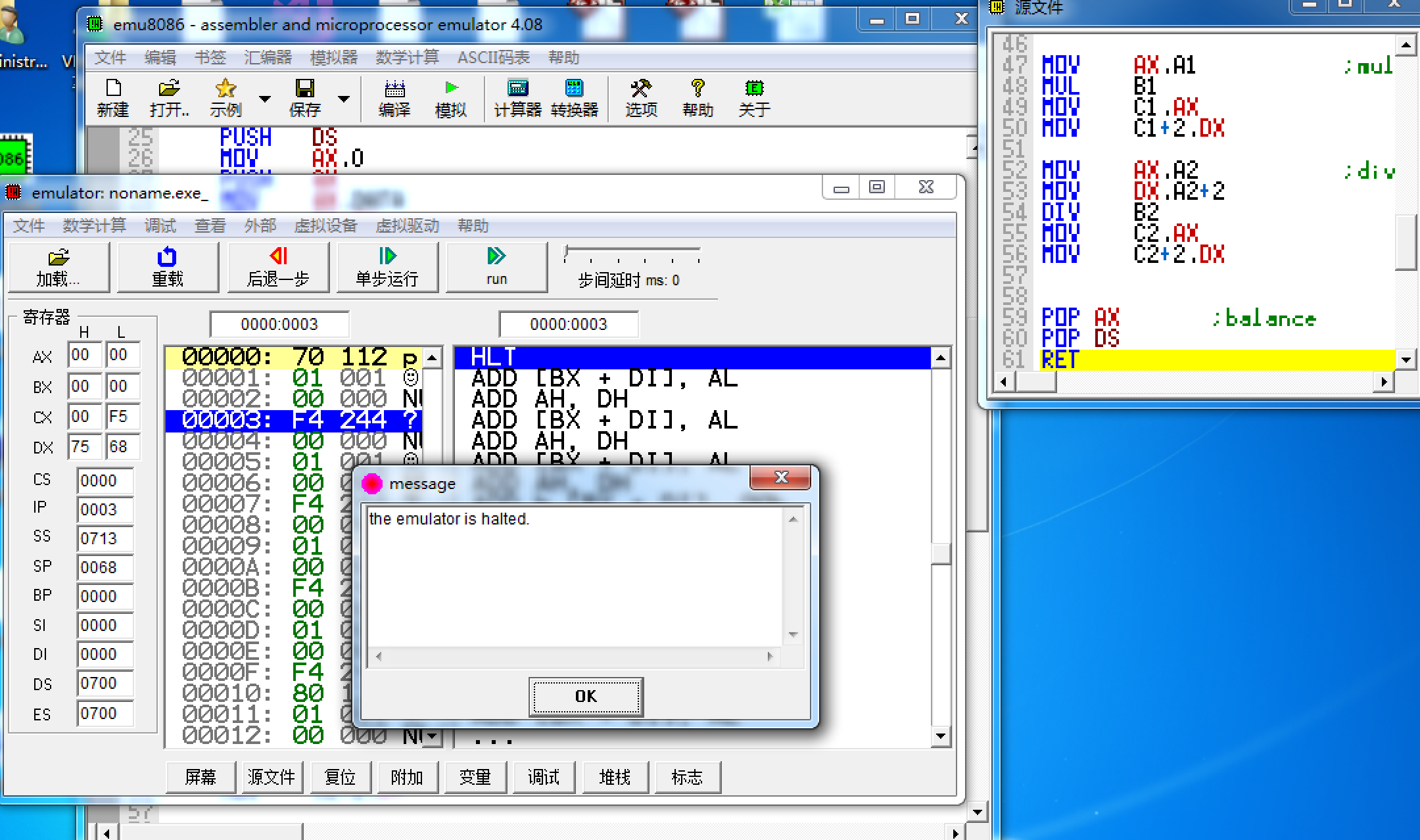
# 实验的运行数据及分析

1. 实验数据记录

(包括内存数据表及运行结果截图)







1. 数据分析：

（结果分析）

C中存了两个和，分别为1234H+FEDCH=1110H和5678H+123AH=68B3H

CD中存了两个差，分别为FEDCH-1234H=ECA8H和BA98H-789AH=41FEH

C1中存了乘积，为D678H\*0012H=000F1470H

C2中存了商和余数，02345678H/F234H=0254H 02345678H%F234H=7568H

# 实验讨论及心得体会

（对实验过程、方案选择、设计调整、实验结果等进行讨论分析）

加减的方法类似，用add和sub指令可仿照示例写出。乘法使用mul指令，结果高位存在DX中，低位存在AX中，将AX先存入结果，然后再存DX。除法将被除数低位放在AX中，高位放在DX中，结果商在AX中，余数在DX中。先存AX，再存DX。

实验前面PUSH DS和PUSH AX，故程序后部要加上POP AX和POP DS，以平衡堆栈。