实验 4　顺序结构程序设计

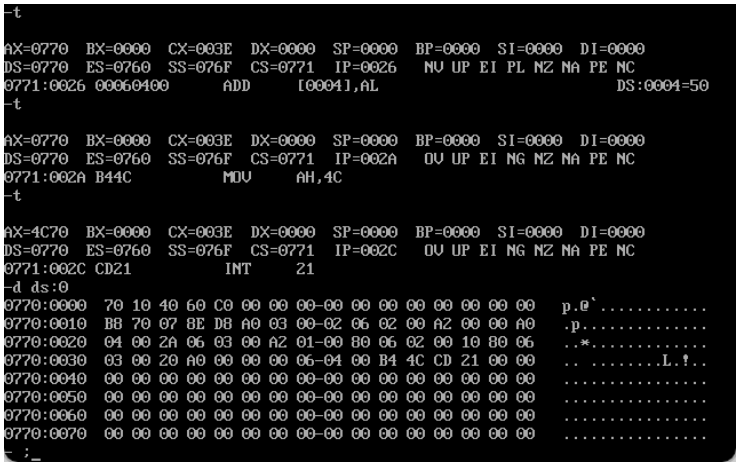
一、实验目的

熟悉汇编语言的上机环境，学习 EDIT,MASM,LINK 程序的使用，掌握建立、汇编、链接、运行汇编语言程序的过程；**掌握常用指令的用法；掌握完整的汇编程序编写过程；掌握汇编程序调试过程。**

二、程序阅读题

试问如下程序执行到 exit 时，字节单元 a、b、cc、cc+1、cc+2 中的内容各是什么？源程序如下：

```
DATA SEGMENT
    A   DB   0
    B   DB   0
    CC   DB   30H,40H,50H
DATA   ENDS
CODE   SEGMENT
       ASSUME   CS:CODE,DS:DATA
START:
          MOV AX, DATA
          MOV DS, AX
          MOV AL,CC+1
          ADD AL,CC
          MOV A,AL
          MOV AL,CC+2
          SUB AL,CC+1
          MOV B,AL
          ADD CC,10H
          ADD CC+1,20H
          MOV AL, A
          ADD CC+2,AL
EXIT:      MOV AH,4CH
          INT 21H
CODE ENDS
          END START
```



当程序执行到 exit 时，字节单元

- A = 70H
- B = 10H
- CC = 40H
- CC+1 = 60H
- CC+2 = C0H

阅读以上程序，**给程序添加注释，分析出结果，并上机验证。**

三、实验题

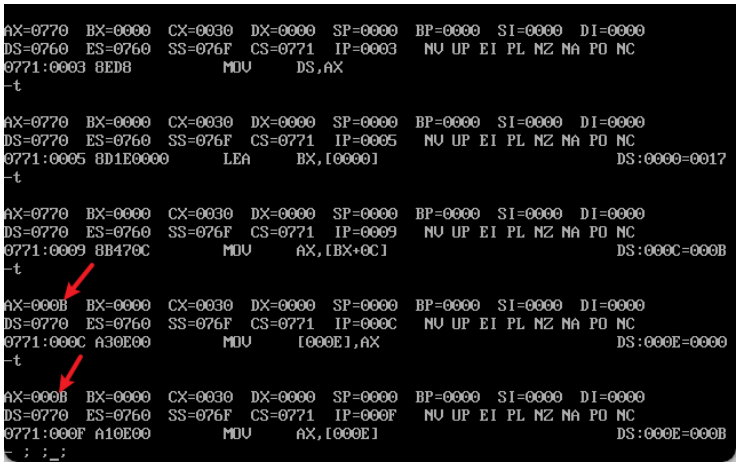
1. 在 ARRAY 数组中依次存储了七个字数据，紧接着是名为 ZERO 的字单元，表示如下：

```
ARRAY   DW    23, 36, 2, 100, 32000, 54, 11
ZERO    DW    ?
```

```
; 定义数据段
DATA SEGMENT
    ARRAY DW 23, 36, 2, 100, 32000, 54, 11  ; 七个字数据的数组
    ZERO  DW ?                              ; 未初始化的字单元
DATA ENDS

; 定义代码段
CODE SEGMENT
    ASSUME CS:CODE, DS:DATA

START:
    ; 初始化数据段寄存器
    MOV AX, DATA
    MOV DS, AX

    ; (1) 如果BX 包含数组ARRAY 的初始地址
    LEA BX, ARRAY           ; 将ARRAY 的地址加载到BX
    MOV AX, [BX+12]         ; 获取ARRAY 中偏移量为12 的数据(即11)
    MOV ZERO, AX            ; 将11 存储到ZERO 单元

    ; 可以添加一些指令来查看结果
    MOV AX, ZERO            ; 将ZERO 的值加载到AX

    ; (2) 如果BX 包含数据11 在数组中的偏移量
    MOV BX, 12              ; 11 在数组中的偏移量是12(6*2=12，因为每个DW 占2
字节)
    MOV AX, ARRAY[BX]       ; 获取ARRAY 中偏移量为BX 的数据(即11)
    MOV ZERO, AX            ; 将11 存储到ZERO 单元

    ; 程序退出
    MOV AH, 4CH
    INT 21H
CODE ENDS
    END START
```

(1) 如果 BX 包含数组 ARRAY 的初始地址，请编写指令将数据 11 传送给 ZERO 单元。使用 debug 命令查看运行结果，截图放在下面。

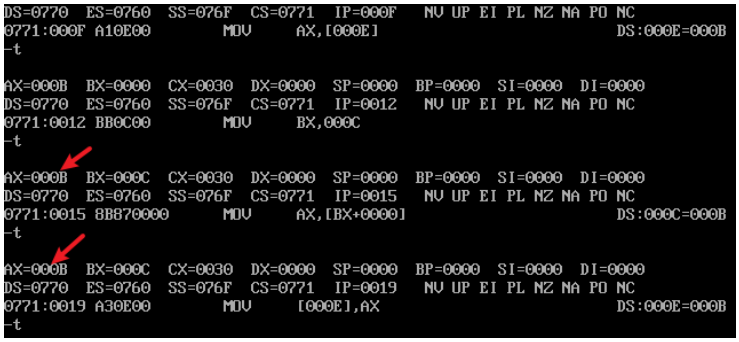
Mov AX [BX+12]



(2) 如果 BX 包含数据 11 在数组中的位移量,请编写指令将数据 11 传送给 ZERO 单元。使用 debug 命令查看运行结果，截图放在下面。Zero 设置的是[000E]

MOV BX, 12

MOV AX, ARRAY[BX]



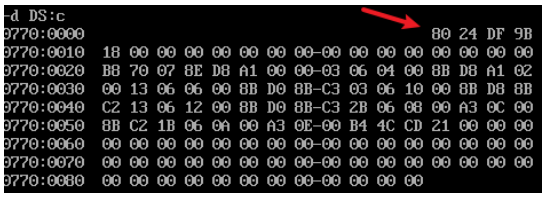
2. 设 x, y, z 均为双精度数，它们分别存放在地址为 X，X+2；Y，Y+2；Z，Z+2 的存储单元中，存放时高位字在高地址中，低位字在低地址中。用指令序列实现 X+Y+24-Z，并把结果存入 W 单元。把源程序附在下面，并把 debug 单步调试的截图附在下面。(备注：x,y,z 的值自行给)

```
DATA SEGMENT
    X DW 1234H, 5678H    ; 双精度数 x (低位在前，高位在后)
    Y DW 2345H, 6789H    ; 双精度数 y (低位在前，高位在后)
    Z DW 1111H, 2222H    ; 双精度数 z (低位在前，高位在后)
    W DW 0, 0            ; 结果存放位置 (低位在前，高位在后)
    CONST_24 DW 24, 0    ; 常数 24 (双精度)
DATA ENDS

CODE SEGMENT
    ASSUME CS:CODE, DS:DATA
START:
    MOV AX, DATA
    MOV DS, AX
```

```
    ; 计算 X + Y
    MOV AX, X           ; 加载 X 的低位
    ADD AX, Y           ; 加上 Y 的低位
    MOV BX, AX          ; 保存低位结果

    MOV AX, X+2         ; 加载 X 的高位
    ADC AX, Y+2         ; 带进位加上 Y 的高位
    MOV DX, AX          ; 保存高位结果

    ; 计算 (X + Y) + 24
    MOV AX, BX          ; 恢复低位结果
    ADD AX, CONST_24    ; 加上 24 的低位
    MOV BX, AX          ; 保存低位结果

    MOV AX, DX          ; 恢复高位结果
    ADC AX, CONST_24+2  ; 带进位加上 24 的高位
    MOV DX, AX          ; 保存高位结果

    ; 计算 (X + Y + 24) - Z
    MOV AX, BX          ; 恢复低位结果
    SUB AX, Z           ; 减去 Z 的低位
    MOV W, AX           ; 保存结果的低位

    MOV AX, DX          ; 恢复高位结果
    SBB AX, Z+2         ; 带借位减去 Z 的高位
    MOV W+2, AX         ; 保存结果的高位

    MOV AH, 4CH
    INT 21H
CODE ENDS
END START
```

```
0772:0003 8ED8          MOV     DS,AX
0772:0005 A10000        MOV     AX,[0000]
0772:0008 03060400      ADD     AX,[0004]
0772:000C 8BD8          MOV     BX,AX
0772:000E A10200        MOV     AX,[0002]
0772:0011 13060600      ADC     AX,[0006]
0772:0015 8BD0          MOV     DX,AX
0772:0017 8BC3          MOV     AX,BX
0772:0019 03061000      ADD     AX,[0010]
0772:001D 8BD8          MOV     BX,AX
0772:001F 8BC2          MOV     AX,DX
-      u
0772:0021 13061200      ADC     AX,[0012]
0772:0025 8BD0          MOV     DX,AX
0772:0027 8BC3          MOV     AX,BX
0772:0029 2B060800      SUB     AX,[0008]
0772:002D A30C00        MOV     [000C],AX
0772:0030 8BC2          MOV     AX,DX
0772:0032 1B060A00      SBB     AX,[000A]
0772:0036 A30E00        MOV     [000E],AX
0772:0039 B44C          MOV     AH,4C
0772:003B CD21          INT     21
0772:003D 0000          ADD     [BX+SI],AL
0772:003F 0000          ADD     [BX+SI],AL
- ;
```

运行结果:



X DW 1234H, 5678H

Y DW 2345H, 6789H

Z DW 1111H, 2222H

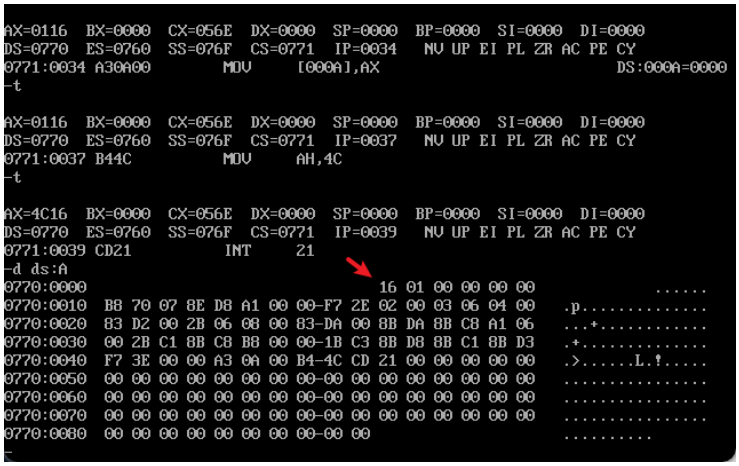
W->2480H,9BDFH,结果正确

3. 编写程序，x，y，z，v 均为 16 位带符号数，计算(v - (x*y + z - 540))/ x。把源程序附在下面，并把 debug 单步调试的截图附在下面。(备注：x,y,z，v 的值自行给)

```
; 定义数据段
DATA SEGMENT
    X DW 5          ; 16 位带符号数 x
    Y DW 10         ; 16 位带符号数 y
    Z DW 100        ; 16 位带符号数 z
    V DW 1000       ; 16 位带符号数 v
    CONST_540 DW 540  ; 常数 540
    RESULT DW ?   ; 存放计算结果
DATA ENDS

; 定义代码段
CODE SEGMENT
    ASSUME CS:CODE, DS:DATA

START:
    ; 初始化数据段寄存器
    MOV AX, DATA
    MOV DS, AX

    ; 计算 x*y
    MOV AX, X     ; 将x 加载到AX
    IMUL Y        ; 有符号乘法，结果在DX:AX

    ; 计算 x*y + z
    ADD AX, Z     ; 将z 加到结果的低16 位
    ADC DX, 0     ; 处理可能的进位
```

```
    ; 计算 x*y + z - 540
    SUB AX, CONST_540  ; 从结果中减去 540
    SBB DX, 0          ; 处理可能的借位

    ; 保存 x*y + z - 540 的结果到 BX:CX
    MOV BX, DX
    MOV CX, AX

    ; 计算 v - (x*y + z - 540)
    MOV AX, V          ; 将 v 加载到 AX
    SUB AX, CX         ; 减去之前计算的结果的低 16 位
    MOV CX, AX         ; 保存结果的低 16 位到 CX

    MOV AX, 0          ; 高 16 位设为 0（因为 v 是 16 位数）
    SBB AX, BX         ; 减去之前计算的结果的高 16 位，考虑借位
    MOV BX, AX         ; 保存结果的高 16 位到 BX

    ; 计算 (v - (x*y + z - 540)) / x
    ; 准备被除数 BX:CX
    MOV AX, CX         ; 将低 16 位移到 AX
    MOV DX, BX         ; 将高 16 位移到 DX

    ; 执行有符号除法
    IDIV X             ; 有符号除法，结果在 AX，余数在 DX

    ; 保存结果
    MOV RESULT, AX

    ; 程序退出
    MOV AH, 4CH
    INT 21H
CODE ENDS
    END START
```

RESULT 为 0770:000A



RESULT = 278 = 0116H

三 实验报告

要求： 1.每步操作过程及所用指令；2.用截图方式反应指令执行前后的相关寄存器或存储单元内容的变化，并对所得结果进行分析。