

# 仲恺农业工程学院实验报告纸

\_\_\_\_自动化\_\_\_\_（院、系）\_\_\_\_自动化\_\_\_\_专业\_\_\_\_214\_\_\_\_班\_\_\_\_组\_\_\_\_微机原理与接口技术\_\_\_\_课

学号：202121724408 姓名：吕凯锋 实验日期:2023.12.8 教师评定

## 实验三 两种基本结构程序设计

### 一、实验目的

1. 掌握循环结构的程序设计
2. 掌握分支结构的程序设计

### 二、实验原理

通过使用 8086 汇编指令设计算法完成简单的运算,同时思考题中利用了-1 的补码存放特点使得 cx 中存放的数据为 FFFFH。实验步骤一中是通过设计一个循环,在这个循环中每执行一次 cx 就会减一, ax 则会加一,跳出循环的条件是当 cx 等于 0 时,因为 cx 初始化为 5,因此最后 ax 的数值为 0005H。

实验步骤二中通过循环来每次为 bx 加一,并且将所有 bx 相加的值放在 ax 中,最后放在偏移地址为 si 内数值的内存单元中,因为是 1 至 100 的数字相加,因此 cx 为 100。

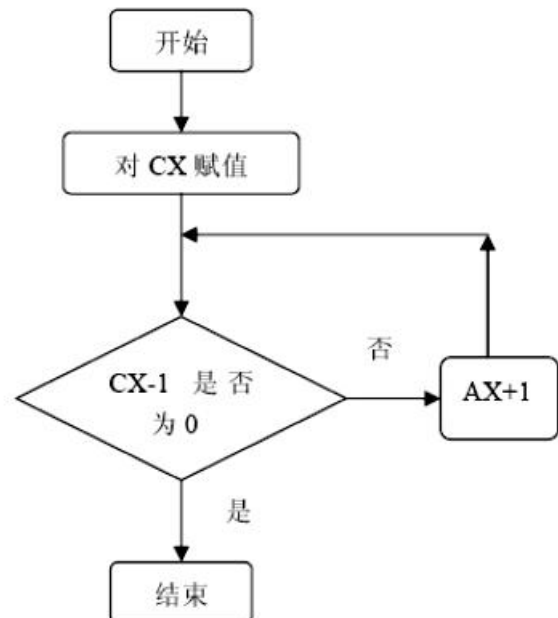
实验步骤三中通过 CMP 指令和 jg, jl 指令来判断是正数还是负数,并且通过 je 指令停止判断。

思考题中因为要让 50000H 到 5FFFFH 的内存单元数值为 55H,因此需要先将段地址 ds 的值初始化为 5000H,然后通过循环在每次写入一个 55H 后,就把数拿出来同 55H 比较检查写入的是否正确。

### 三、实验内容和步骤

1. 录入、调试并运行程序：通过给 CX 一个数值,再通过 LOOP 做一个判断 CX-1 是否为 0 的转移,实现程序的循环,循环的内容是执行 AX+1,程序流程如右图所示,记录程序运行完毕时 AX 寄存器中的内容并简要解释原因。

```
CODE SEGMENT
ASSUME CS: CODE
CON_A EQU 25
CON_B EQU 12
START:
MOV AX, 0
MOV CX, 5
INC_AX: NOP
INC AX
LOOP INC_AX
JMP $
```

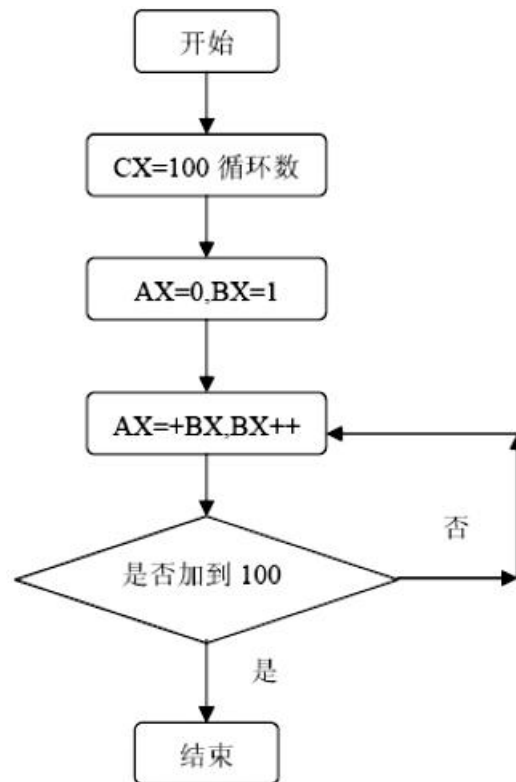


CODE ENDS

2、录入、调试并运行程序：实现 100 以内的自然数累加，即：1+2+3+...+100，程序流程如右图所示，记录运行结果。

参考程序：

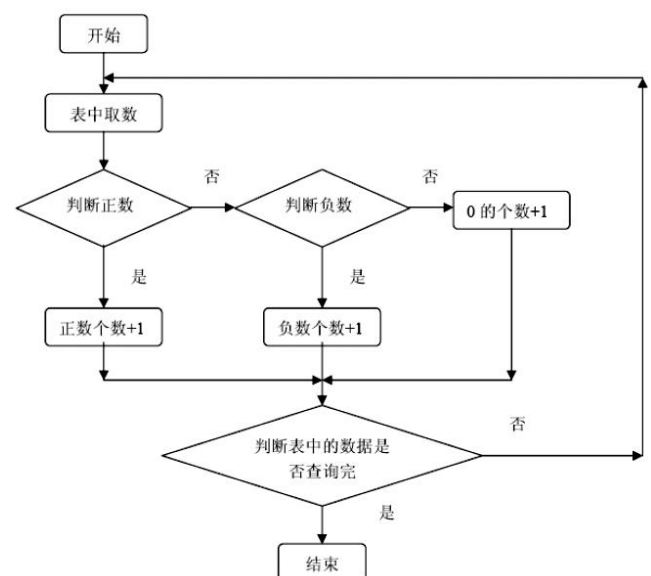
```
DATA SEGMENT
total DW 0000H
DATA ENDS
CODE SEGMENT
ASSUME CS: CODE, DS: DATA
BEG: MOV AX, DATA
MOV DS, AX
MOV SI, OFFSET total
MOV CX, 100
MOV AX, 0
MOV BX, 1
add_100: ADD AX, BX
INC BX
LOOP add_100
MOV [SI], AX
JMP $
CODE ENDS
END BEG
```



3. 录入、调试并运行程序：实现统计数组中正数、负数与 0 的个数，程序流程如下图所示，记录运行的结果。

参考程序：

```
DATA SEGMENT
DBUF DB 1, 2, -3, -4, -5, 1, 2, 3, 4, 5, ' #'
PLUS DB 1
MINS DB 1
DATA ENDS
STACK SEGMENT
STACK DB 200 DUP(0)
STACK ENDS
CODE SEGMENT
ASSUME DS: DATA, SS: STACK, CS: CODE
```



```

START: MOV  AX,  DATA
MOV  DS,  AX
MOV  PLUS,  010
MOV  MINS,  0
MOV  DI,  0
BG:  CMP  DBUF[DI],  '#'
JE  EN
INC  DI
CMP  DBUF[DI],  0
JG  L1
JL  L2
L1:  INC  PLUS
JMP  BG
L2:  INC  MINS
JMP  BG
EN:  MOV  AH,  4CH
INT  21H

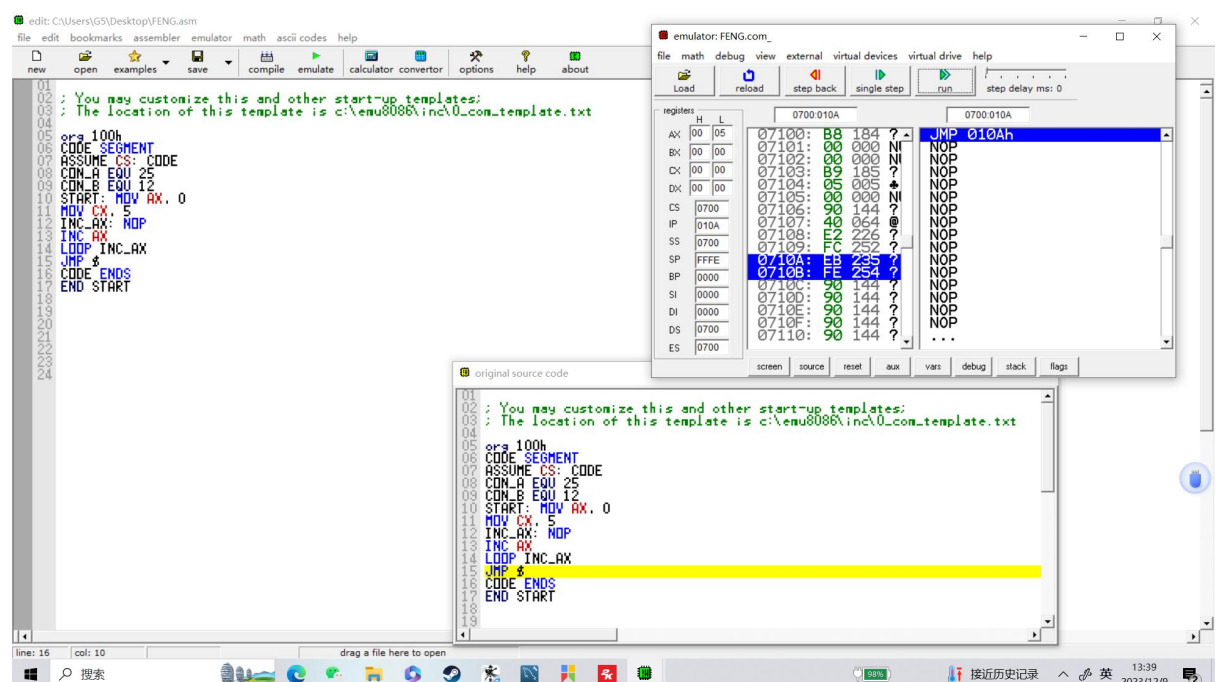
```

#### 4. 思考题

编程将内存从 50000H 到 5FFFFH 的每个单元均写入数 55H，并再逐个单元读出比较，看写入的与读出的是否一致。若全对，则将 AL 置 0；否则将 AL 置 0FFH。请画出程序流程图，写出程序代码，录入、调试并运行。

## 四、实验数据和处理

步骤一运行结果如下：



The screenshot shows a Windows 10 desktop with a debugger (emu8086) running a program. The main window displays assembly code with comments. A 'Random Access Memory' window is open, showing memory addresses and values. A 'Registers' window is also open, showing the state of various registers. The taskbar at the bottom shows the Start button, search bar, and several application icons.

**Assembly Code (Main Window):**

```

01 ; You may customize this and other start-up templates;
02 ; The location of this template is c:\emu8086\inc\0_com.template.txt
03
04
05 org 100h
06 DATA SEGMENT
07 total dw 0000h
08 DATA ENDS
09 CODE SEGMENT
10 ASSUME CS: CODE, DS: DATA
11 BEG: MOV AX, DATA
12 MOV DS, AX
13 MOV SI, OFFSET total
14 MOV CX, 100
15 MOV AX, 0
16 MOV BX, 1
17 add 100: ADD AX, BX
18 INC BX
19 LOOP add 100
20 MOV [SI], AX
21 JMP $
22 CODE ENDS
23 END BEG

```

**Random Access Memory Window:**

Address	Value
0010:0000	BA 13 00 F4 00 01 00 F4-00 01 00 F4 00 01 00 F4
0010:0010	00 01 00 00 F4 00 01 00 F4-00 01 00 F4 00 01 00 F4
0010:0020	00 01 00 00 F4 00 01 00 F4-00 01 00 F4 00 01 00 F4
0010:0030	00 01 00 00 F4 00 01 00 F4-00 01 00 F4 00 01 00 F4
0010:0040	00 01 00 00 F4 00 01 00 F4-00 01 00 F4 00 01 00 F4
0010:0050	00 01 00 00 F4 00 01 00 F4-00 01 00 F4 00 01 00 F4
0010:0060	00 01 00 00 F4 00 01 00 F4-00 01 00 F4 00 01 00 F4
0010:0070	00 01 00 00 F4 00 01 00 F4-00 01 00 F4 00 01 00 F4

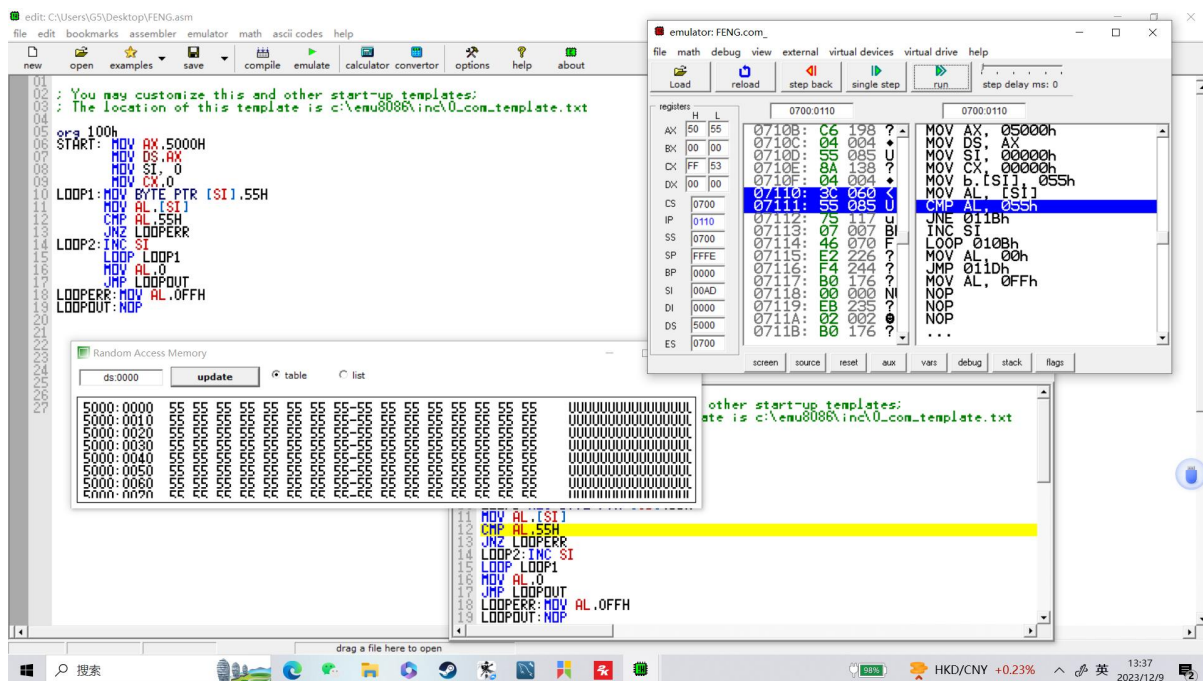
**Registers Window:**

Register	Value
AX	13 BA
BX	00 65
CX	00 00
DX	00 00
SI	0712: 00 00
DI	0000
ES	0700

**Taskbar:** The taskbar shows the Start button, a search bar, and several application icons including File Explorer, Edge, and the debugger.

[illegible]

思考题运行结果如下：



## 五、实验器材

IBM-PC 机

## 六、实验总结

本次实验收获良多，让我了解到负数在计算机中的存储方式，一开始我还在为 `cx` 的数值发愁，直到发现了 -1 对应的补码即为 `FFFFH` 后茅塞顿开。同时对于条件转移指令的了解和使用更加的熟练，也了解了更多的编程思想。

