**微机原理实验一 算术逻辑运算及求和运算**

班级：信息005

姓名：王靳朝

学号：2206113602

1. **实验目的**

1.熟悉8086汇编语言程序的基本格式，以及汇编、链接的基本过程。

2.熟悉和了解HQFC-A集成调试软件的使用操作方法，熟悉在PC机上建立、汇编、链接、调试和运行8086汇编语言程序的全过程。

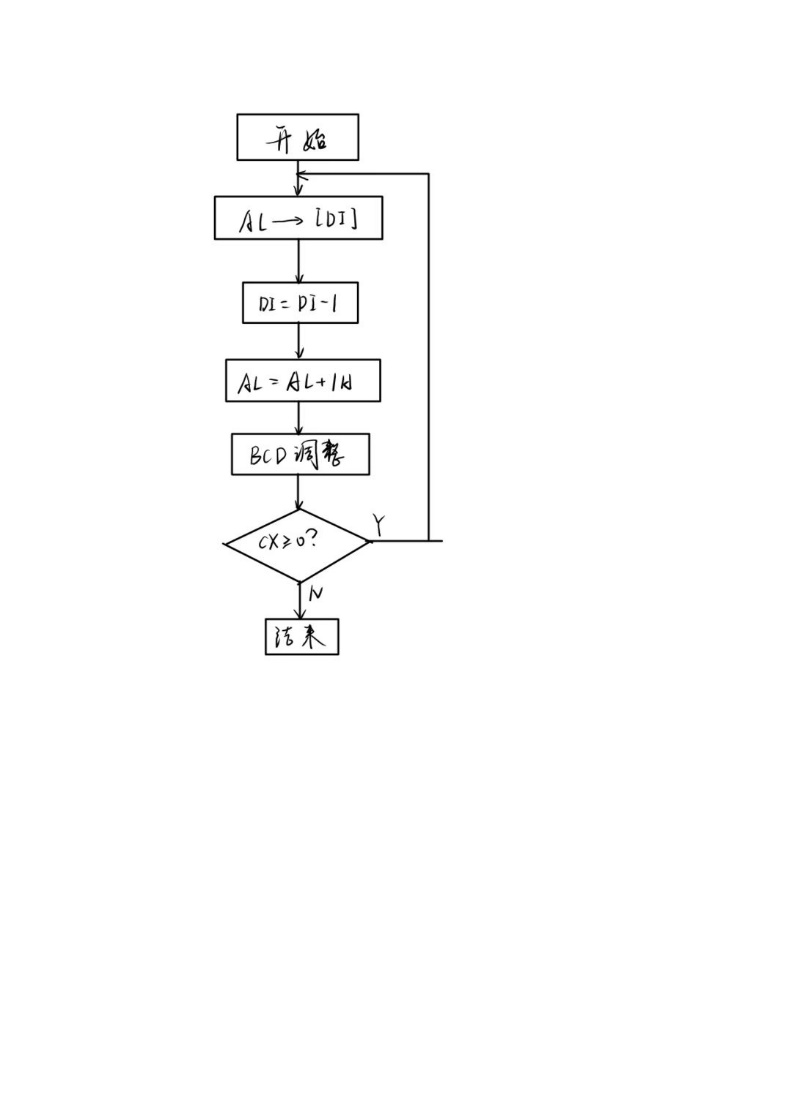
3.掌握使用各种寻址指令，和运算类指令编程及调试方法。

4.编程实现十进制数按照顺序在内存中排列。

1. **实验预习内容**
2. 复习8086指令系统中的算术逻辑类指令和移位指令
3. 了解HQFC-A集成调试软件的安装和使用方法。
4. 了解标志寄存器个标志位的含义和指令执行对他们的影响。
5. **实验内容及步骤**
6. 以下汇编语言程序实现在数据存储器中指定单元3500H～350FH中建立0～15共16个十进制数，程序代码如下：

|  |
| --- |
| CODE SEGMENT ;在内存指定单元建立十进制数  ASSUME CS:CODE  ORG 2000H ;程序起始地址  START: MOV DI,3500H ;设数据区首址  MOV CX,0010H ;字节数→ CX  MOV AX,0000 ;0000 → AX  CNT: MOV [DI],AL ;（AL）→ [DI]  INC DI ;修改地址指针  ADD AL,01 ;修改数据  DAA ;十进制调整  LOOP CNT ;未填完，继续  JMP $ ;填完，结束  CODE ENDS  END START |

做出程序框图如下：

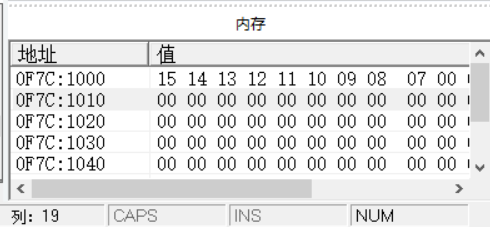


1. 按照要求对程序进行修改，在数据存储器中指定单元1000H～100FH中建立15～0共16个十进制数，且使其建立的数据为**降序排列**的十进制数。

由于数的总量不变，因此CX不需要改变。起始地址改变时，只需要改变DI的初始值位1000H即可。需要降序排列只需要将AL的初值设置为15，之后对AL的值在每次循环中进行减操作即可。源代码如下：

|  |
| --- |
| CODE SEGMENT ;在内存指定单元建立十进制数  ASSUME CS:CODE  ORG 2000H ;程序起始地址  START: MOV DI,1000H ;设数据区首址  MOV CX,0010H ;字节数→ CX  MOV AX,000FH ;000F → AX  DAA ;十进制调整，变为00 00 00 15  CNT: MOV [DI],AL ;（AL）→ [DI]  INC DI ;修改地址指针  SUB AL,01 ;修改数据，AL-1  DAS ;十进制调整，减法调整DAS  LOOP CNT ;未填完，继续  JMP $ ;填完，结束  CODE ENDS  END START |

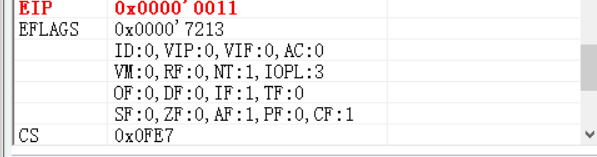
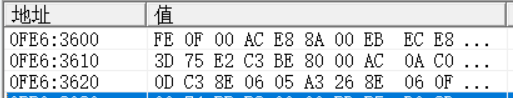
可以观察到，在DS：1000H开始的寄存器中写入了正确的降序排序数：



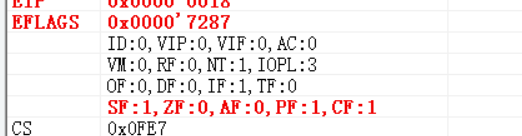
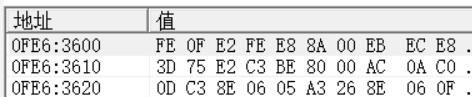
1. 完成二进制32位加法运算，计算Z=X+Y。并将结果存入DS:3600H。更改X，Y的值，测试标志寄存器各标志位的意义和指令执行对它的影响。

由于是32位加法，8086是16位，因此需要将低16位和高16位加法分离计算，需要CF进位位用来标识低位是否向高位进位，因此低位的加法可以使用ADD指令，而高位的加法需要使用ADC指令用来加上进位位的影响.源代码如下：

|  |
| --- |
| DATA SEGMENT  XL DW 0FFFFH ;X低16位  XH DW 0FFE0H ;X高16位  YL DW 00FFFH ;Y低16位  YH DW 0FF01H ;Y高16位  DATA ENDS  CODE SEGMENT  ASSUME CS:CODE,DS:DATA  START:MOV AX,DATA  MOV DS,AX  MOV AX,XL ;X低16位移到AX  ADD AX,YL ;低16位无符号数相加  MOV DI,3600H ;偏移地址送给DI  MOV [DI],AX ;已经计算的低16位结果送给3600H、3601H    MOV AX,XH ;X高16位移到AX  ADC AX,YH ;高16位相加，视作带进位的相加，可能有进位、溢出等  MOV DI,3602H ;低16位相加存入之后偏移地址已经+2  MOV [DI],AX ;高16位结果送给3602H、3603H  CODE ENDS  END START |

当我们把低16位设置为0FFFH和FFFFH之后，低十六位相加存在进位，因此CF寄存器数值为1，同时低16位保持的计算结果位0FFEH结果正确。

当计算高16位时，由于考虑了低位的进位，因此最后一位结果为2，同时高位存在加法之后的进位和符号位改变，因此SF=1，CF=1。

1. **实验思考和总结**
2. 这次实验虽然比较简单，但在观察寄存器状态窗口的内容时不很熟练，导致调试效率低下。不过通过本次实验，还是学会了HQFC-A集成调试软件的使用流程以及基本的debug技能，对以后的实验会有较大的帮助。
3. 本次实验能帮助我思考溢出位、进位位和符号位的关系。实验中我会思考计算机如何区分有符号数和无符号数的加减法运算。思考之后得出结论：计算机中并不区分二者，所有的数据全部视为有符号数，并且按照补码规则输入计算机。各符号位的作用如下：

ZF标志位------0标志位，记录相关指令执行之后，其结果是否为0。为0，则ZF=1；不为0，则ZF=0。

PF标志位-----奇偶标志位，如果有奇数个1，则PF=0;如果有偶数个1，则PF=1;

SF标志位----符号标志位。计算机在执行加减运算的时候，不区分有符号数和无符号数，但是会在标志位上有所体现。如果进行的是无符号运算，SF标志位无意义；有符号运算。如果SF=1,表示结果为负；如果SF=0,表示结果为正。

CF标志位-----进位，仅针对无符号运算，记录了运算结果的最高有效位向更高位的进位值，或从更高位的借位值。

OF标志位-------溢出，仅针对有符号数的运算。有符号运算。SF记录正负，OF关心溢出，CF无意义。无符号运算。SF无意义，OF无意义，关心CF的溢出。