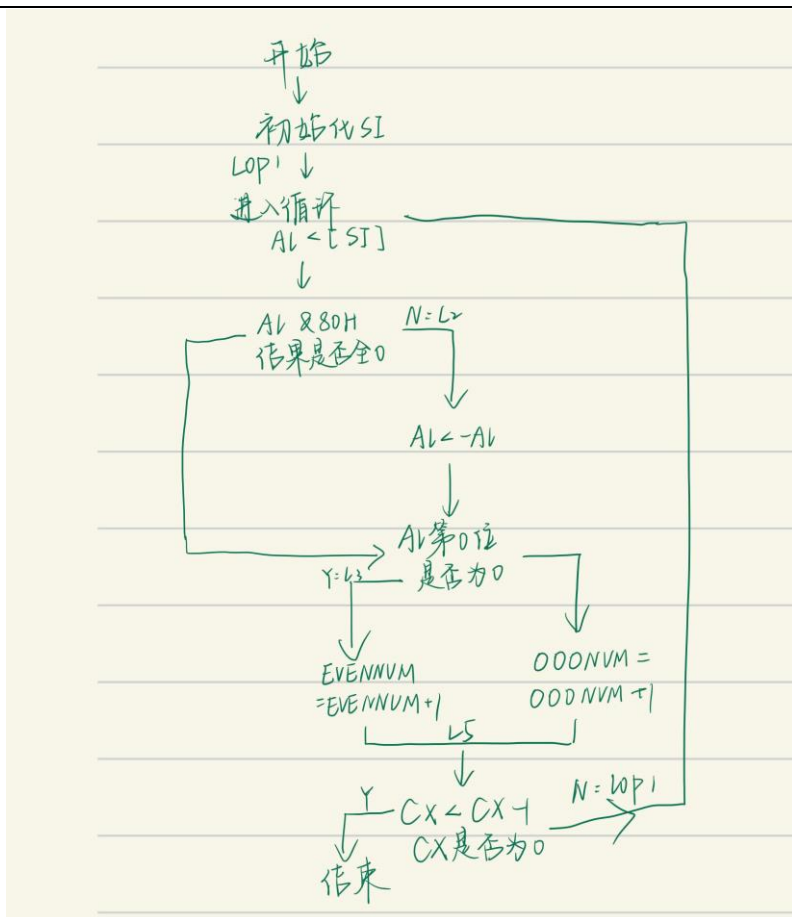


四川 大学 计算机 学院、软件 学院  
实 验 报 告

学号：\_\_\_\_2021141460301\_\_\_\_ 姓名：\_\_\_\_杨怡\_\_\_\_ 专业：\_\_计算机科学与技术\_\_\_\_  
\_\_\_\_ 班级：\_\_\_\_行政六班\_\_\_\_ 第 14 周

课程名称	汇编语言程序设计课程设计	实验课时	4
实验项目	分支与循环结构程序设计	实验时间	2022. 5. 25
实验目的	1. 掌握分支、循环结构程序设计方法，重点掌握结合程序流程图设计分支、循环结构入、出口的方法。 2. 掌握使用 DEBUG 调试分支、循环结构的方法。		
实验环境	文本编辑器，dosbox		
实验内容 (算法、程序、步骤和方法)	<p>本次实验指定的程序设计功能：</p> <p>给定一个字节型数组，长度为 10 字节：</p> <pre>VARB    DB    14, -25, 66, -17, 78, -33, 46</pre> <pre>          DB    -96, 71, 80</pre> <p>编写一个完整的源程序，统计数组中奇数和偶数的个数，统计结果分别保存在 ODDNUM 和 EVENNUM 单元中。</p> <p>一、设计程序流程图</p>		



- 1、将数据段段基值赋给 ds, mov 指令不影响标志位
- 2、xor bx, bx; BX 指向 VARB 第 0 个字节

mov cx, len; 初始化循环计数器

- 3、jcxz 15; 若计数值为 0, 则退出循环
- 4、11: test varb[bx], 00000001

jz 12

jnz 13;

- (1) 标号 11 执行测试指令. 其中 CF、OF 被强行置为 0, AF 不确定, SF、ZF、PF 解释与算术运算指令 AND 指令完全一致。
- (2) 若最低位为 0, 则测试结束 (DEST) 为 0, ZF=1, 跳转至标号 12, 否则顺序执行;
- (3) 最低位为 1, 则测试结束 (DEST) 为 1, ZF=0, 跳转至标号 13。

5、 inc oddnum

jmp 14;

- (1) 数组元素为偶数时，跳转至 12 执行，内存单元 oddnum 自加 1，影响 OF, SF, ZF, AF, PF 五个状态标志位
- (2) 自加后无条件跳转至标号 14

6、 inc evennum;

- (1) 若数组该元素为奇数，跳转至 13 执行，内存单元 evennum 自加 1，影响 OF, SF, ZF, AF, PF 五个状态标志位
- (2) 自加后顺序执行标号 14

7、 inc bx

loop 11;

- (1) 数组下标加 1
- (2) 使用 loop 指令，首先 (CX) -1=>CX，该减法不会影响标志位，然后进行判断。若 (CX) 不等于 0，则转移至标号 11，继续循环；否则，循环完成，退出循环，顺序执行

8、 MOV AH, 4CH

INT 21H;

二、 DATAS **SEGMENT** 段定义 **SEGMENT ENDS** 关键字

VARB **DB** 14, -25, 66, -17, 78, -33, 46

**DB** -96, 71, 80 定义数组

ODDNUM **DB** 0

EVENNUM **DB** 0

DATAS **ENDS**

STACKS **SEGMENT** STACK 堆栈段定义 **SEGMENT ENDS** 关键字

**DW** 20H **DUP** (0) 把0重复分配20H次 16位单元即循 STACKS **ENDS** 环  
分配内存空间

CODES **SEGMENT**

ASSUME CS:CODES, DS:DATAS, SS:STACKS

伪指令区分段CODES与CS对应, DATAS与DS对应, STACKS与SS:对应, 保证访问时使用段前缀正确

START:

MOV AX, DATAS                    初始化DS、ES

MOV DS, AX

MOV SI, OFFSET BYTE PTR VARB        将数组引入si

MOV CX, 10                    为CX赋值10, 循环10次

lop1: MOV AL, [SI]            将当前si指向的值赋给al

TEST AL, 80H                判断最高位是1还是0以此判断是正数还是负数

JNZ L2                    如果结果不是全0, 即AL中是负数, 则跳转到L2

JMP L1                    否则无条件跳转到L1, 即当AL中为正数时

L2: NEG AL                    取得负数的绝对值

L1: TEST AL, 01H            判断最低位是什么, 是0则是偶数, 是1则为奇数

JZ L3                    结果全0, 即是偶数, 则跳转到L3

JMP L4                    否则无条件跳转到L4, 即当AL中为奇数时

L3: INC EVENNUM            偶数计数器加一

JMP L5                    无条件跳转到L5

L4: INC ODDNUM            奇数计数器加一

L5: DEC CX                    CX减一, 表面循环进行了一次

INC SI                    令SI指向下一个数值

JNZ lop1    若结果为非0, 即还未执行到十次, 则跳转回lop1, 以完成循环

MOV AH, 4CH                返回DOS

INT 21H

CODES ENDS

END START                结束源程序

三、程序设计原理:

(1) 补码具有保奇偶性, 可以直接判断最低位。

(2) 用测试指令直接判断所有数字的最低位, 为0则为偶数, 内存单元 oddnum 自加1, 为1则为奇数, 内存单元 evennum 自加1

(3) 判断是否对数组中的10个数都进行奇偶判断, 决定继续循环或退出循环。

(4) 在文本编辑器中输入代码生成test.asm文件后, 在dosbox中进行汇编; 查看目录, 生成了test.obj文件; 使用link程序连接, 生成test.exe文件; 使用debug程序进行调试。

```
-d0770:0000
0770:0000 0E E7 42 EF 4E DF 2E A0-47 50 00 00 00 00 00 00 ..B.N...GP.....
0770:0010 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 .....
0770:0020 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 .....
0770:0030 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 .....
0770:0040 00 00 00 00 00 00 00 00-00 00 00 00 00 FF FF .....
0770:0050 B8 70 07 8E D8 BE 00 00-B9 09 00 F6 04 80 75 02 .p.....u.
0770:0060 EB 02 F6 D8 A8 01 74 02-EB 06 FE 06 0B 00 EB 04 .....t.....
0770:0070 FE 06 0A 00 46 49 75 E3-B4 4C CD 21 00 00 00 00 ....Flu..L.!....
```

```
AX=0770 BX=0000 CX=007D DX=0000 SP=0040 BP=0000 SI=0000 DI=0000
DS=0760 ES=0760 SS=0771 CS=0775 IP=0003 NU UP EI PL NZ NA PO NC
0775:0003 8ED8 MOV DS,AX
-t
AX=0770 BX=0000 CX=007D DX=0000 SP=0040 BP=0000 SI=0000 DI=0000
DS=0770 ES=0760 SS=0771 CS=0775 IP=0005 NU UP EI PL NZ NA PO NC
0775:0005 BE0000 MOV SI,0000
-t
AX=0770 BX=0000 CX=007D DX=0000 SP=0040 BP=0000 SI=0000 DI=0000
DS=0770 ES=0760 SS=0771 CS=0775 IP=0008 NU UP EI PL NZ NA PO NC
0775:0008 B90A00 MOV CX,000A
-t
AX=0770 BX=0000 CX=000A DX=0000 SP=0040 BP=0000 SI=0000 DI=0000
DS=0770 ES=0760 SS=0771 CS=0775 IP=000B NU UP EI PL NZ NA PO NC
0775:000B 8A04 MOV AL,[SI] DS:0000=0E
-t
AX=070E BX=0000 CX=000A DX=0000 SP=0040 BP=0000 SI=0000 DI=0000
DS=0770 ES=0760 SS=0771 CS=0775 IP=000D NU UP EI PL NZ NA PO NC
0775:000D A880 TEST AL,80
```

```
AX=070E BX=0000 CX=000A DX=0000 SP=0040 BP=0000 SI=0000 DI=0000
DS=0770 ES=0760 SS=0771 CS=0775 IP=001F NU UP EI PL NZ NA PO NC
0775:001F EB04 JMP 0025
-t
AX=070E BX=0000 CX=000A DX=0000 SP=0040 BP=0000 SI=0000 DI=0000
DS=0770 ES=0760 SS=0771 CS=0775 IP=0025 NU UP EI PL NZ NA PO NC
0775:0025 49 DEC CX
-t
AX=070E BX=0000 CX=0009 DX=0000 SP=0040 BP=0000 SI=0000 DI=0000
DS=0770 ES=0760 SS=0771 CS=0775 IP=0026 NU UP EI PL NZ NA PE NC
0775:0026 46 INC SI
-t
AX=070E BX=0000 CX=0009 DX=0000 SP=0040 BP=0000 SI=0001 DI=0000
DS=0770 ES=0760 SS=0771 CS=0775 IP=0027 NU UP EI PL NZ NA PO NC
0775:0027 75E2 JNZ 000B
-t
AX=070E BX=0000 CX=0009 DX=0000 SP=0040 BP=0000 SI=0001 DI=0000
DS=0770 ES=0760 SS=0771 CS=0775 IP=000B NU UP EI PL NZ NA PO NC
0775:000B 8A04 MOV AL,[SI] DS:0001=E7
```

```

AX=070E BX=0000 CX=000A DX=0000 SP=0040 BP=0000 SI=0000 DI=0000
DS=0770 ES=0760 SS=0771 CS=0775 IP=000F  NU UP EI PL ZR NA PE NC
0775:000F 7502          JNZ     0013
-t

AX=070E BX=0000 CX=000A DX=0000 SP=0040 BP=0000 SI=0000 DI=0000
DS=0770 ES=0760 SS=0771 CS=0775 IP=0011  NU UP EI PL ZR NA PE NC
0775:0011 EB02          JMP     0015
-t

AX=070E BX=0000 CX=000A DX=0000 SP=0040 BP=0000 SI=0000 DI=0000
DS=0770 ES=0760 SS=0771 CS=0775 IP=0015  NU UP EI PL ZR NA PE NC
0775:0015 A801          TEST    AL,01
-t

AX=070E BX=0000 CX=000A DX=0000 SP=0040 BP=0000 SI=0000 DI=0000
DS=0770 ES=0760 SS=0771 CS=0775 IP=0017  NU UP EI PL ZR NA PE NC
0775:0017 7402          JZ      001B
-t

AX=070E BX=0000 CX=000A DX=0000 SP=0040 BP=0000 SI=0000 DI=0000
DS=0770 ES=0760 SS=0771 CS=0775 IP=001B  NU UP EI PL ZR NA PE NC
0775:001B FE060B00      INC     BYTE PTR [000B]          DS:000B=00

```

1. 执行 MOV TEST AL, 80 后

CF: 强制归 0    OF:强制归 0    SF; 符号不变    AF:不确定

PF:因为该数为正数，计算结果是 0 个 1，因此为 PE

ZF: 因为该数为正数，计算结果是 0，因此是 ZR

2. 执行 TEST AL, 01 后

CF: 强制归 0    OF:强制归 0    SF; 符号不变    AF:不确定

PF:因为该数为正数，计算结果是 0 个 1，因此为 PE

ZF: 因为该数为正数，计算结果是 0，因此是 ZR

3. 执行 INC ENENUM 后，因为原来为 0 加了 1 后

CF:无进位或借位，OF;无溢出，SF; 符号不变    AF:三到四位无进位

PF:因为该数由 0 变为 1，计算结果是 1 个 1，因此为 P0

ZF: 因为该数不全为 0，因此变为 NZ

4. 执行 DEC CX 后

CF:无进位或借位, 0F;无溢出, SF; 符号不变      AF:三到四位无借位

PF:计算结果是 9, 偶数个 1, 因此为 PE

ZF: 因为该数不全为 0, 因此变为 NZ

5. 执行 INC SI 后

CF:无进位或借位, 0F;无溢出, SF; 符号不变      AF:三到四位无借位

PF:计算结果是 1, 奇数个 1, 因此为 PO

ZF: 因为该数不全为 0, 因此变为 NZ

```
-d0770:0000
0770:0000  0E E7 42 EF 4E DF 2E A0-47 50 00 01 00 00 00 00  ..B.N...GP.....
0770:0010  00 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00  .....
0770:0020  00 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00  .....
0770:0030  00 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00  .....
0770:0040  00 00 00 00 00 00 0E 07-00 00 0B 00 75 07 A6 01  .....u...
0770:0050  B8 70 07 8E D8 BE 00 00-B9 0A 00 8A 04 A8 80 75  .p.....u
0770:0060  02 EB 02 F6 D8 A8 01 74-02 EB 06 FE 06 0B 00 EB  .....t.....
0770:0070  04 FE 06 0A 00 49 46 75-E2 B4 4C CD 21 00 00 00  ....IFu..L.!...
```

观察结果, 发现偶数计数器加了 1

最后经过判断, 还未完成循环后, 程序回到循环起始, 开始对第二个数据的判断

```
-g02a
AX=07B0 BX=0000 CX=0000 DX=0000 SP=0040 BP=0000 SI=000A DI=0000
DS=0770 ES=0760 SS=0771 CS=0775 IP=002A  NU UP EI PL ZR NA PE NC
0775:002A B4C          MOV     AH,4C
-d0770:0000
0770:0000  0E E7 42 EF 4E DF 2E A0-47 50 04 06 00 00 00 00  ..B.N...GP.....
0770:0010  00 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00  .....
0770:0020  00 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00  .....
0770:0030  00 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00  .....
0770:0040  00 00 00 00 00 00 00 00 00-00 00 2A 00 75 07 A6 01  .....*.u...
0770:0050  B8 70 07 8E D8 BE 00 00-B9 0A 00 F6 04 80 8A 04  .p.....
0770:0060  46 75 02 EB 02 F6 D8 A8-01 74 02 EB 06 FE 06 0B  Fu.....t.....
0770:0070  00 EB 04 FE 06 0A 00 49-75 E1 B4 4C CD 21 00 00  ....Iu..L.!...
```

运用 g 指令, 直接跳过中间 9 次循环, 观察结果, 发现奇数有 4 个, 偶数有 6 个, 与实际相符, 试验成功

结 论 (结 果)	
小 结	通过这次实验，我学会了画程序流程图，重点掌握了结合程序流程图设计分支、循环结构入、出口的方法，对汇编语言分支、循环结构程序设计方法的学习有了更深入的了解，对汇编源程序的编制有了更具象化的认识与体验。
指导 老师 评 议	<div>成绩评定：</div> <div>指导教师签名：</div>