

学号： WA2224013 专业： 机器人工程 姓名： 郭义月

实验日期： 2024. 12. 20 指导教师： 鲍华 实验成绩：

课程目标 1 (权重_____)	课程目标 2 (权重_____)	课程目标 3 (权重_____)	课程目标 4 (权重_____)	课程目标 5 (权重_____)	课程目标 6 (权重_____)	综合成绩 (目标数可增删)

安徽大学人工智能学院本科实验报告

【课程名称】 微型计算机原理及接口技术

【课程目标】

提升对内存操作、条件判断、循环控制与子程序的理解与应用能力。

【实验名称】 实验四：统计偶数个数与输出斐波那契数列

【实验目的】

掌握在汇编语言中如何有效地使用循环来处理数组数据，并学习字符与数值之间的转换技巧，以便于数据的正确输出和显示。

【实验原理及方法】

在计算机中，所有的数据都是以二进制形式存储的。对于8位无符号数来说，它们占用一个字节（8位）。偶数和奇数的区别在于其二进制表示的最低位（LSB）是0还是1。在二进制中，偶数的最低位是0，而奇数的最低位是1。通过判断最低位可以判断该数是奇数还是偶数。斐波那契数列的生成可以通过迭代的方式实现。在汇编语言中，我们需要维护两个寄存器或内存位置来存储前两个斐波那契数，并在每次迭代中更新这两个值以生成下一个斐波那契数

【实验内容及过程】

1. 从DAT开始的数据区存放10个8位无符号数：12, 34, 63, 25, 76, 88, 93, 46, 74, 27，编程统计其中偶数的个数，并将结果显示输出

实验代码:

```
DATAS SEGMENT
;此处输入数据段代码
DAT DB 12,34,63,25,76,88,93,46,74,27
DATAS ENDS

STACKS SEGMENT
;此处输入堆栈段代码
STACKS ENDS

CODES SEGMENT
ASSUME CS:CODES,DS:DATAS,SS:STACKS
START:
MOV AX,DATAS
MOV DS,AX
;此处输入代码段代码
MOV SI,OFFSET DAT
MOV CL,10
MOV BL,0
AGAIN:
MOV AX,[SI]
SHR AX,1
JC NEXT
INC BL
NEXT:
INC SI
DEC CL
JNZ AGAIN
MOV DL,BL
ADD DL,'0'
MOV AH,6
INT 21H
MOV AH,4CH
INT 21H
CODES ENDS
END START
```

2. 按照下面公式, 编程显示其前 10 个值。 $F(0)=1$, $F(1)=1$, $F(X)=F(X-1)+F(X-2)$ $X>2$

实验代码:

```
DATAS SEGMENT
;此处输入数据段代码
DAT DB 10 DUP(0)
DATAS ENDS

STACKS SEGMENT
;此处输入堆栈段代码
```

```

STACKS ENDS

CODES SEGMENT
    ASSUME CS:CODES,DS:DATAS,SS:STACKS
START:
    MOV AX,DATAS
    MOV DS,AX
    ;此处输入代码段代码
    MOV SI,OFFSET DAT
    MOV CL,8
    MOV DL,1
    AND DX,00FFH
    MOV [SI],DX
    MOV DL,1
    AND DX,00FFH
    INC SI
    MOV [SI],DX
    MOV BH,1
    MOV CH,1
AGAIN:
    MOV BL,BH
    ADD BH,CH
    MOV CH,BL
    MOV DL,BH
    AND DX,00FFH
    INC SI
    MOV [SI],DX
    DEC CL
    JNZ AGAIN
    MOV BH,0AH
    MOV SI,OFFSET DAT
L1:
    MOV DL,[SI]
    MOV CX,1;计数 初始为 1
    MOV BL,10;每次除 10
ONE:
    MOV AH,0;清零（存余数）
    MOV AL,DL
    DIV BL;除 10
    PUSH AX;进栈
    CMP AL,0;是否已除净
    JLE ENDL
    MOV DL,AL
    INC CX
    JMP ONE
ENDL:
    POP DX
    XCHG DH,DL;余数转入 DL
    ADD DL,30H
    MOV AH,2

```

```

INT 21H
LOOP ENDL
INC SI
DEC BH
MOV DL, ' '
MOV AH,6
INT 21H
JNZ L1
MOV AH,4CH
INT 21H
CODES ENDS
END START

```

【实验结果】

实验1:

当数据段如图中所示时，12，34，76，88，46，74，有6个偶数，可以正确输出



此时，可以调试查看每个数据SHR之后对应的CF的值

进位标志 CF(Carry flag) CY(1) NC(0)

当AL=0CH，即12时，SHR之后，NC代表CF=0

```

DOSBox 0.74, Cpu speed: 3000 cycles, Frameskip 0, Program:...
AX=0770 BX=0000 CX=0036 DX=0000 SP=0000 BP=0000 SI=0000 DI=0000
DS=0770 ES=0760 SS=076F CS=0771 IP=0008  NV UP EI PL NZ NA PO NC
0771:0008 B10A      MOV     CL,0A
-p
AX=0770 BX=0000 CX=000A DX=0000 SP=0000 BP=0000 SI=0000 DI=0000
DS=0770 ES=0760 SS=076F CS=0771 IP=000A  NV UP EI PL NZ NA PO NC
0771:000A B300      MOV     BL,00
-p
AX=0770 BX=0000 CX=000A DX=0000 SP=0000 BP=0000 SI=0000 DI=0000
DS=0770 ES=0760 SS=076F CS=0771 IP=000C  NV UP EI PL NZ NA PO NC
0771:000C 8B04      MOV     AX,[SI]
DS:0000=220C
-p
AX=220C BX=0000 CX=000A DX=0000 SP=0000 BP=0000 SI=0000 DI=0000
DS=0770 ES=0760 SS=076F CS=0771 IP=000E  NV UP EI PL NZ NA PO NC
0771:000E D1E8      SHR     AX,1
-p
AX=1106 BX=0000 CX=000A DX=0000 SP=0000 BP=0000 SI=0000 DI=0000
DS=0770 ES=0760 SS=076F CS=0771 IP=0010  NV UP EI PL NZ AC PF NC
0771:0010 7202      JB      0014

```

当AL=3FH，即63时，SHR之后，CY代表CF=1

```

DOSBox 0.74, Cpu speed: 3000 cycles, Frameskip 0, Program:...
AX=1F91 BX=0002 CX=0009 DX=0000 SP=0000 BP=0000 SI=0002 DI=0000
DS=0770 ES=0760 SS=076F CS=0771 IP=0015  NV UP EI PL NZ NA PO NC
0771:0015 FEC9      DEC     CL
-p
AX=1F91 BX=0002 CX=0008 DX=0000 SP=0000 BP=0000 SI=0002 DI=0000
DS=0770 ES=0760 SS=076F CS=0771 IP=0017  NV UP EI PL NZ NA PO NC
0771:0017 75F3      JNZ     000C
-p
AX=1F91 BX=0002 CX=0008 DX=0000 SP=0000 BP=0000 SI=0002 DI=0000
DS=0770 ES=0760 SS=076F CS=0771 IP=000C  NV UP EI PL NZ NA PO NC
0771:000C 8B04      MOV     AX,[SI]
DS:0002=193F
-p
AX=193F BX=0002 CX=0008 DX=0000 SP=0000 BP=0000 SI=0002 DI=0000
DS=0770 ES=0760 SS=076F CS=0771 IP=000E  NV UP EI PL NZ NA PO NC
0771:000E D1E8      SHR     AX,1
-p
AX=0C9F BX=0002 CX=0008 DX=0000 SP=0000 BP=0000 SI=0002 DI=0000
DS=0770 ES=0760 SS=076F CS=0771 IP=0010  NV UP EI PL NZ AC PF CY
0771:0010 7202      JB      0014

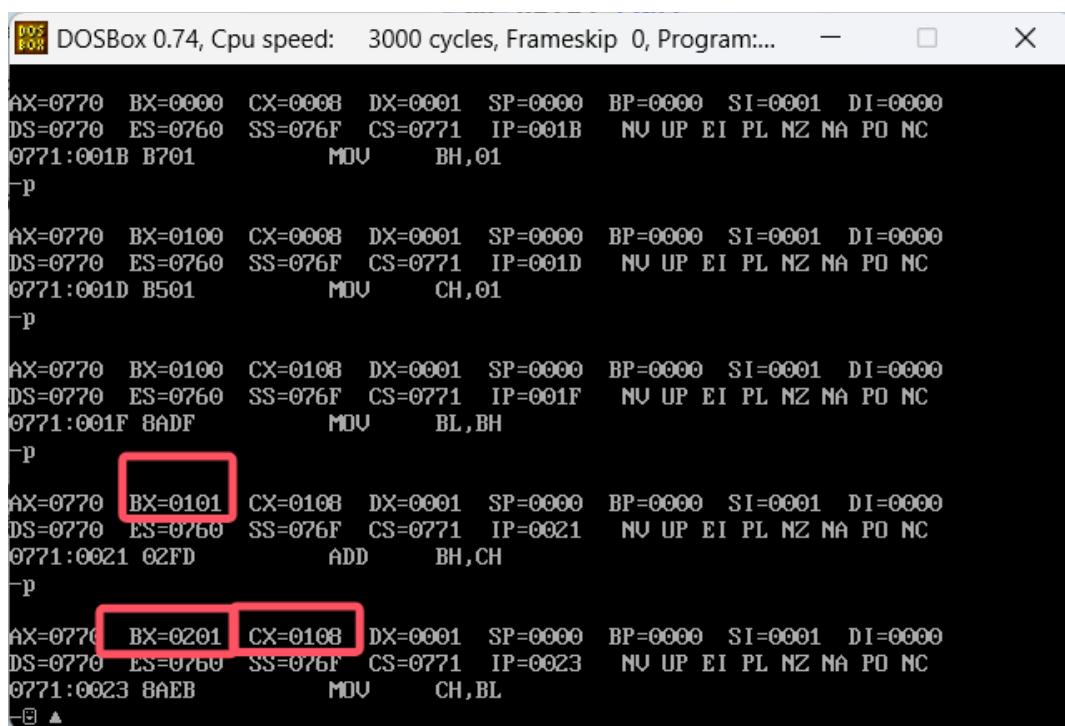
```

CF设置正确时候，只要按照对应的跳转函数对偶数实现加一操作就可以实现统计偶数个数的功能

实验2:

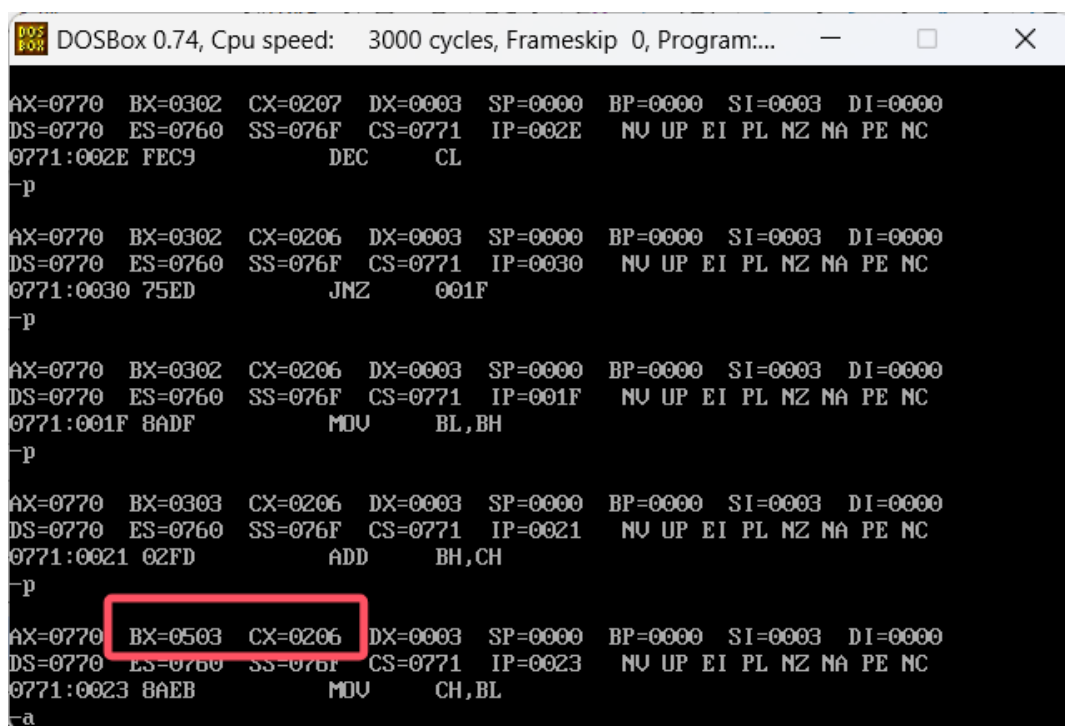
在编写计算前十个值时，以下代码的逻辑为：BL，CH中存放代求的数之前的两个值，在每次循环中，先把BL的值保留在BH，之后BH = BH + CH，再更新CH的值为BH之前的值，即BL，之

后把BH赋值给DL，把DL存放到预留出来的DAT数组对应的位置即可。

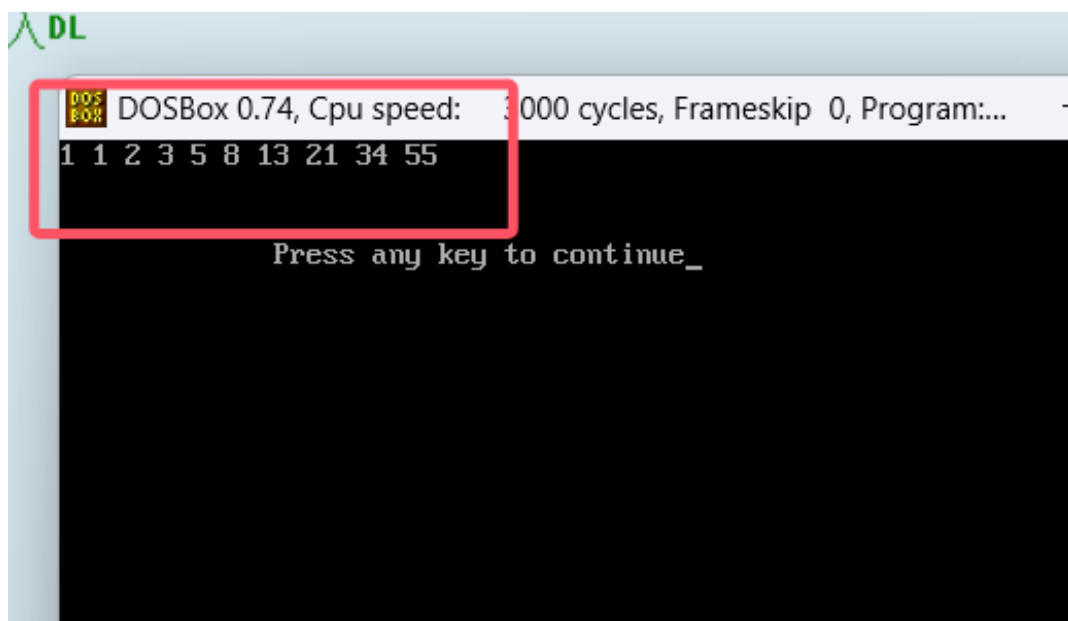


```
DOSBox 0.74, Cpu speed: 3000 cycles, Frameskip 0, Program:...
AX=0770 BX=0000 CX=0008 DX=0001 SP=0000 BP=0000 SI=0001 DI=0000
DS=0770 ES=0760 SS=076F CS=0771 IP=001B  NU UP EI PL NZ NA PO NC
0771:001B B701      MOV     BH,01
-p
AX=0770 BX=0100 CX=0008 DX=0001 SP=0000 BP=0000 SI=0001 DI=0000
DS=0770 ES=0760 SS=076F CS=0771 IP=001D  NU UP EI PL NZ NA PO NC
0771:001D B501      MOV     CH,01
-p
AX=0770 BX=0100 CX=0108 DX=0001 SP=0000 BP=0000 SI=0001 DI=0000
DS=0770 ES=0760 SS=076F CS=0771 IP=001F  NU UP EI PL NZ NA PO NC
0771:001F 8ADF      MOV     BL,BH
-p
AX=0770 BX=0101 CX=0108 DX=0001 SP=0000 BP=0000 SI=0001 DI=0000
DS=0770 ES=0760 SS=076F CS=0771 IP=0021  NU UP EI PL NZ NA PO NC
0771:0021 02FD      ADD     BH,CH
-p
AX=0770 BX=0201 CX=0108 DX=0001 SP=0000 BP=0000 SI=0001 DI=0000
DS=0770 ES=0760 SS=076F CS=0771 IP=0023  NU UP EI PL NZ NA PO NC
0771:0023 8AEB      MOV     CH,BL
```

单步调试可以看到，BH中存放计算出的值，BL存放前两个数中较小的值，BH存放前两个数中较大的值，当BH需要改变时，就先把BH存放到CH中，之后再吧BL的值更新为CH（BH原来的值）即可



```
DOSBox 0.74, Cpu speed: 3000 cycles, Frameskip 0, Program:...
AX=0770 BX=0302 CX=0207 DX=0003 SP=0000 BP=0000 SI=0003 DI=0000
DS=0770 ES=0760 SS=076F CS=0771 IP=002E  NU UP EI PL NZ NA PE NC
0771:002E FEC9      DEC     CL
-p
AX=0770 BX=0302 CX=0206 DX=0003 SP=0000 BP=0000 SI=0003 DI=0000
DS=0770 ES=0760 SS=076F CS=0771 IP=0030  NU UP EI PL NZ NA PE NC
0771:0030 75ED      JNZ     001F
-p
AX=0770 BX=0302 CX=0206 DX=0003 SP=0000 BP=0000 SI=0003 DI=0000
DS=0770 ES=0760 SS=076F CS=0771 IP=001F  NU UP EI PL NZ NA PE NC
0771:001F 8ADF      MOV     BL,BH
-p
AX=0770 BX=0303 CX=0206 DX=0003 SP=0000 BP=0000 SI=0003 DI=0000
DS=0770 ES=0760 SS=076F CS=0771 IP=0021  NU UP EI PL NZ NA PE NC
0771:0021 02FD      ADD     BH,CH
-p
AX=0770 BX=0503 CX=0206 DX=0003 SP=0000 BP=0000 SI=0003 DI=0000
DS=0770 ES=0760 SS=076F CS=0771 IP=0023  NU UP EI PL NZ NA PE NC
0771:0023 8AEB      MOV     CH,BL
```



【数据分析及处理】

在实验1中，编写代码的逻辑为，判断奇数偶数可以通过二进制的最后一位判断，如果二进制的最后一位为1，说明该数是奇数，如果二进制的最后一位是零，说明该数是偶数，可以通过SHR移位一位，此时最后一位就会被存放在CF中，通过判断CF是否为1，用JC来实现跳转，如果CF为0，代表是偶数，计数器+1。

注意，由于需要输出的是10进制数，需要将ascii转化为十进制输出，可以加'0'来实现。

```
ADD DL, '0'  
MOV AH, 6  
INT 21H
```

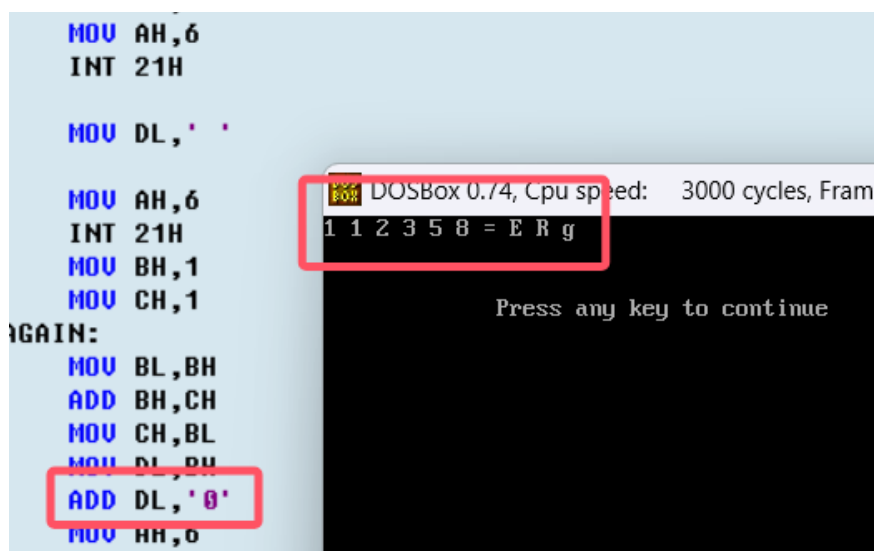
实验2：代码编写的逻辑为：先计算出来按照公式算出来的前10个值，存放在预先分配好的DAT中，再循环依次将ascii转十进制输出。

在编写计算前十个值时，以下代码的逻辑为：BL，CH中存放代求的数之前的两个值，在每次循环中，先把BL的值保留在BH，之后 $BH = BH + CH$ ，再更新CH的值为BH之前的值，即BL，之后把BH赋值给DL，把DL存放到预留出来的DAT数组对应的位置即可。

注意，因为要不停的调用ah中断，每次调用都会改变ax的值，所以不能给ax赋值下一个循环需要用的变量，这样可用的寄存器就很有有限。

注意：实验1中因为输出的数在0-9之间，所以可以通过加上字符零来实现ascii转十进制，但是在实验2中，计算出来的结果有大于9的数，此时如果还加字符零输出，输出就不是我们想要的结果，如下图，前六个数小于10，可以正常输出，第七个数算出来是13，就无法正常输

出十进制。



```
MOV AH,6
INT 21H

MOV DL,' '

MOV AH,6
INT 21H
MOV BH,1
MOV CH,1
AGAIN:
MOV BL,BH
ADD BH,CH
MOV CH,BL
MOV DL,BH
ADD DL,'0'
MOV AH,0
```

DOSBox 0.74, Cpu speed: 3000 cycles, Fram
1 1 2 3 5 8 = E R g
Press any key to continue

之后把计算出的值依次将ascii转十进制输出即可，ascii转10进制的代码为：

```
L1:
    MOV DL,[SI]
    MOV CX,1;计数 初始为1
    MOV BL,10;每次除10
ONE:
    MOV AH,0;清零（存余数）
    MOV AL,DL
    DIV BL;除10
    PUSH AX;进栈
    CMP AL,0;是否已除净
    JLE ENDL
    MOV DL,AL
    INC CX
    JMP ONE
ENDL:
    POP DX
    XCHG DH,DL;余数转入DL
    ADD DL,30H
    MOV AH,2
    INT 21H
    LOOP ENDL
    INC SI
    DEC BH
```

注意每次输出一个数之后，都要再输出一个空格，这样可以较为方便的检验输出是否正确

```
MOV DL,' '
MOV AH,6
INT 21H
```


除此之外还要注意，在本实验中，如果不使用DAT存储计算出的第三个值，而是计算出来一个输出一个时，理论上是可行的，但是由于将ascii转化为十进制输出时，需要借助循环，这样就会用到CX寄存器，在计算时，BX，CX中都存储有计算函数时要用的信息，此时寄存器就会不够用，所以需要先把计算出的10个数存放到数组内，然后用SI依次读取数组中的元素，依次转化，这样就不会有寄存器被其他函数占用其值不能改变导致的寄存器不够用的情况。

【总结或讨论】

加深了我对循环的理解以及输出时需要把ascii转化位十进制的应用