PRACTICAL: 3(C11)

AIM: Multiply two 8-bit numbers stored in memory locations 4001H and 4006H by repetitive addition and store the result at 400AH location.(Use Data Set -3) (Note: Student need to implement FOR loop in this program: initialization, Compare, Decrement/Increment; also need to use JMP, JMx instructions.)
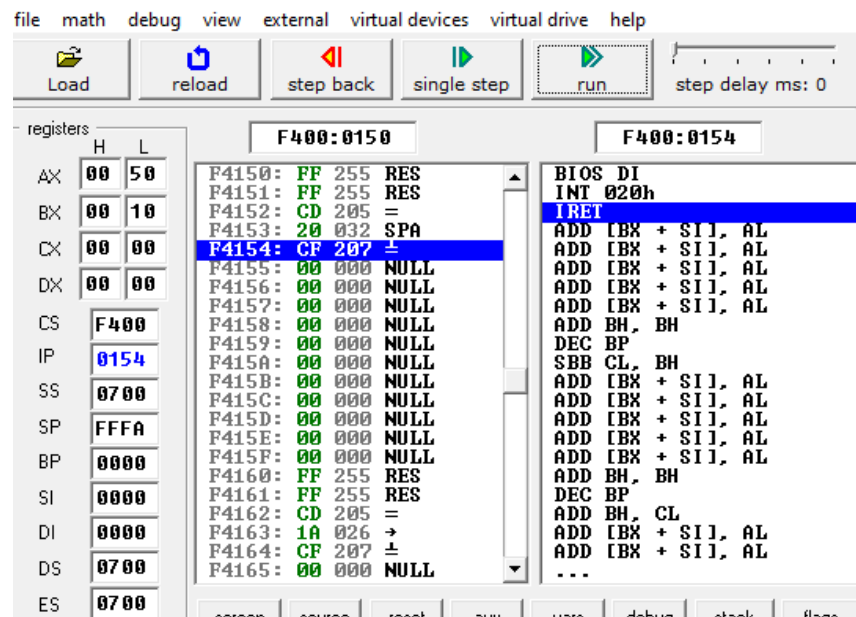
CODE:
```
org 100h
MOV [4001h],10h
mov bl,[4001h]
mov [4000h],0h

MOV cl,0005h
MOV al,[4000h]
l1:
add al,bl
loop l1

ret
```

OUTPUT:

CONCLUSION:
I learned how to use loop concept.

PRACTICAL: 3(C12)

AIM: Program to find average of n numbersorg 100h

CODE:

```
mov [4001h],1h
mov [4002h],2h
mov [4003h],3h
mov [4004h],4h
mov [4005h],5h

mov si,4001h
mov al,0h
mov cl,5h
mov dl,5h

l1:
mov bl,[si]
add al,bl
inc si
loop l1

div dl

mov [6000h],al

ret
```

OUTPUT:
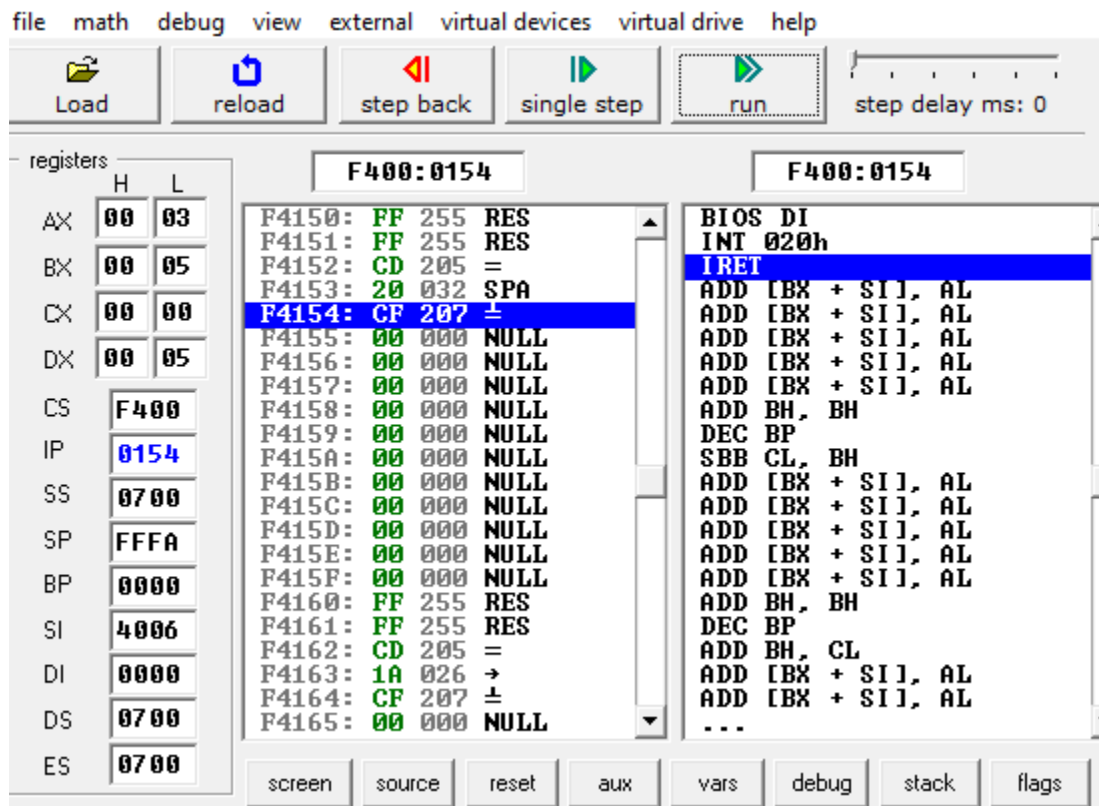
CONCLUSION:

I learned how to use loop concept using Source index.

PRACTICAL: 3(C13)

AIM: Write an assembly language program to find the no. of odd numbers and even numbers, given an array of n numbers.

CODE:

```
org 100h

mov [6000h], 01h
mov [6001h], 02h
mov [6002h], 03h
mov [6003h], 04h
mov [6004h], 05h
mov SI, 6000h
mov cl, 05h
mov dl, 00h
l1:
mov al, [SI]
INC SI
shr al, 1
JNC even
INC dl
even:
loop l1
mov [400Ah], dl
```

OUTPUT:

```
file   math   debug   view   external   virtual devices   virtual drive   help

   Load      reload    step back    single step      run       step delay ms: 0

registers                F400:0154                    F400:0154
        H    L
AX   00  02     F4150: FF  255  RES      BIOS DI
                F4151: FF  255  RES      INT 020h
BX   00  00     F4152: CD  205  =        IRET
                F4153: 20  032  SPA      ADD [BX + SI], AL
CX   00  00     F4154: CF  207  ±        ADD [BX + SI], AL
                F4155: 00  000  NULL     ADD [BX + SI], AL
DX   00  03     F4156: 00  000  NULL     ADD [BX + SI], AL
                F4157: 00  000  NULL     ADD [BX + SI], AL
CS   F400       F4158: 00  000  NULL     ADD BH, BH
                F4159: 00  000  NULL     DEC BP
IP   0154       F415A: 00  000  NULL     SBB CL, BH
                F415B: 00  000  NULL     ADD [BX + SI], AL
SS   0700       F415C: 00  000  NULL     ADD [BX + SI], AL
                F415D: 00  000  NULL     ADD [BX + SI], AL
SP   FFFA       F415E: 00  000  NULL     ADD [BX + SI], AL
                F415F: 00  000  NULL     ADD [BX + SI], AL
BP   0000       F4160: FF  255  RES      ADD BH, BH
                F4161: FF  255  RES      DEC BP
SI   6005       F4162: CD  205  =        ADD BH, CL
                F4163: 1A  026  →        ADD [BX + SI], AL
DI   0000       F4164: CF  207  ±        ADD [BX + SI], AL
                F4165: 00  000  NULL     ...
DS   0700
ES   0700
          screen  source  reset  aux   vars  debug  stack  flags
```

```
Random Access Memory                                    —    □    ×

   700:400a        update       ● table    ○ list

0700:400A   03 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00   ♥...............
0700:401A   00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00   ................
0700:402A   00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00   ................
0700:403A   00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00   ................
0700:404A   00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00   ................
0700:405A   00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00   ................
0700:406A   00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00   ................
0700:407A   00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00
```

CONCLUSION:

I learned how to check whether given number is odd or even.