St. Francis Institute of Technology

Class: SE-ITA/ITB Semester: IV; A.Y. 2023-2024
Subject: Microprocessor Lab

Experiment – 5: Count the number of 1's and 0's for a 16-bit number

1. **Aim**:
   Write an ALP to count the number of 1's and 0's for a 16-bit number.

2. **Requirements**
   DOSBox (an x86 emulator with DOS), Turbo Assembler, Turbo Debugger

3. **Pre-Experiment Exercise**
   Algorithm:
   a. Initialize the data segment. Load the 16 bit number in the AX register.
   b. Initialize BX register as a counter for zeros and DX register as a counter for ones.
   c. Initialize CX as a counter register.
   d. Keep rotating the AX register by 1 till the counter becomes zero.
   e. While rotating if carry is set, increment DX register, if carry is clear, increment BX.
   f. Store the result from DX and BX register in two separate memory locations.
   g. Stop

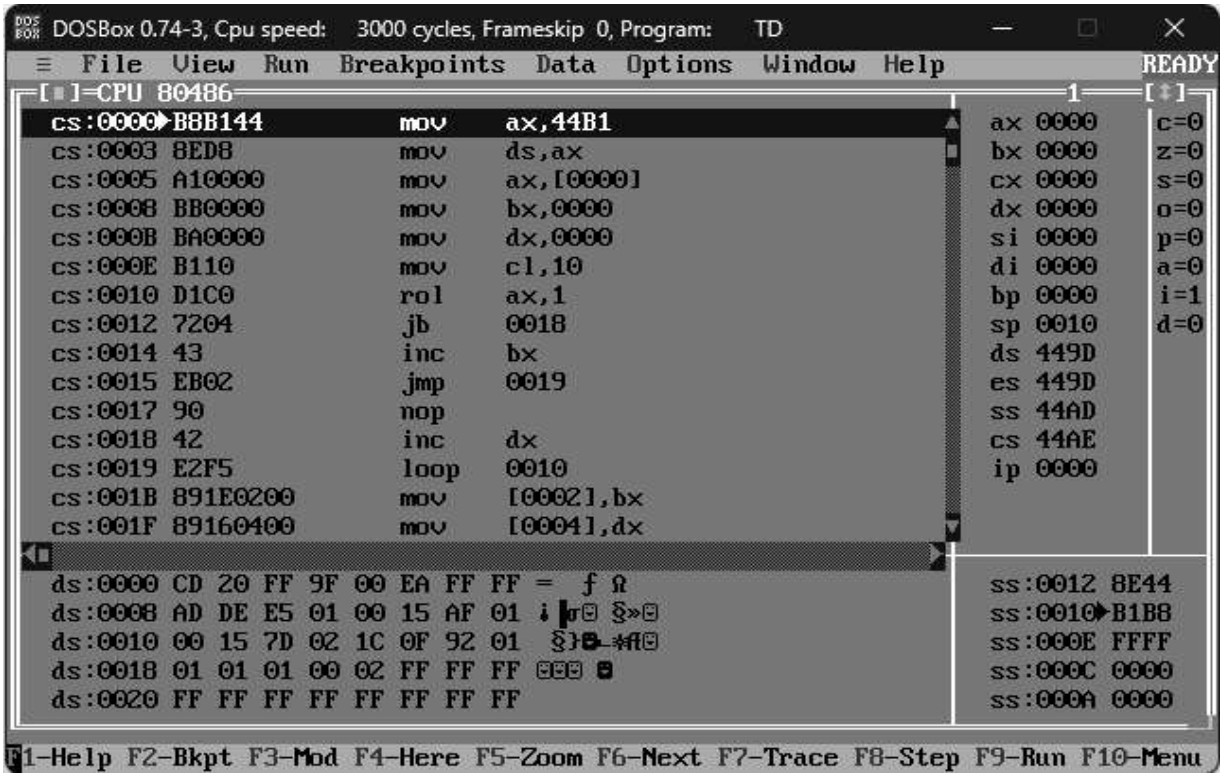
4. **Laboratory Exercise:**
   Procedure:
   a. Open DOSbox and go to TASM.
   b. Open a new document using the command - edit <filename>.asm
   c. Write the Program and save the changes to the same file.
   d. Assemble the program using the command - tasm <filename.asm>
   e. If any errors are displayed, then change the code in <filename>
   f. If no errors are displayed, execute command - tlink <filename>.obj to create the executable file.
   g. Next execute the command - td <filename>
   h. Try to RUN the program step by step and view the changes in the registers, flags, memory, etc.

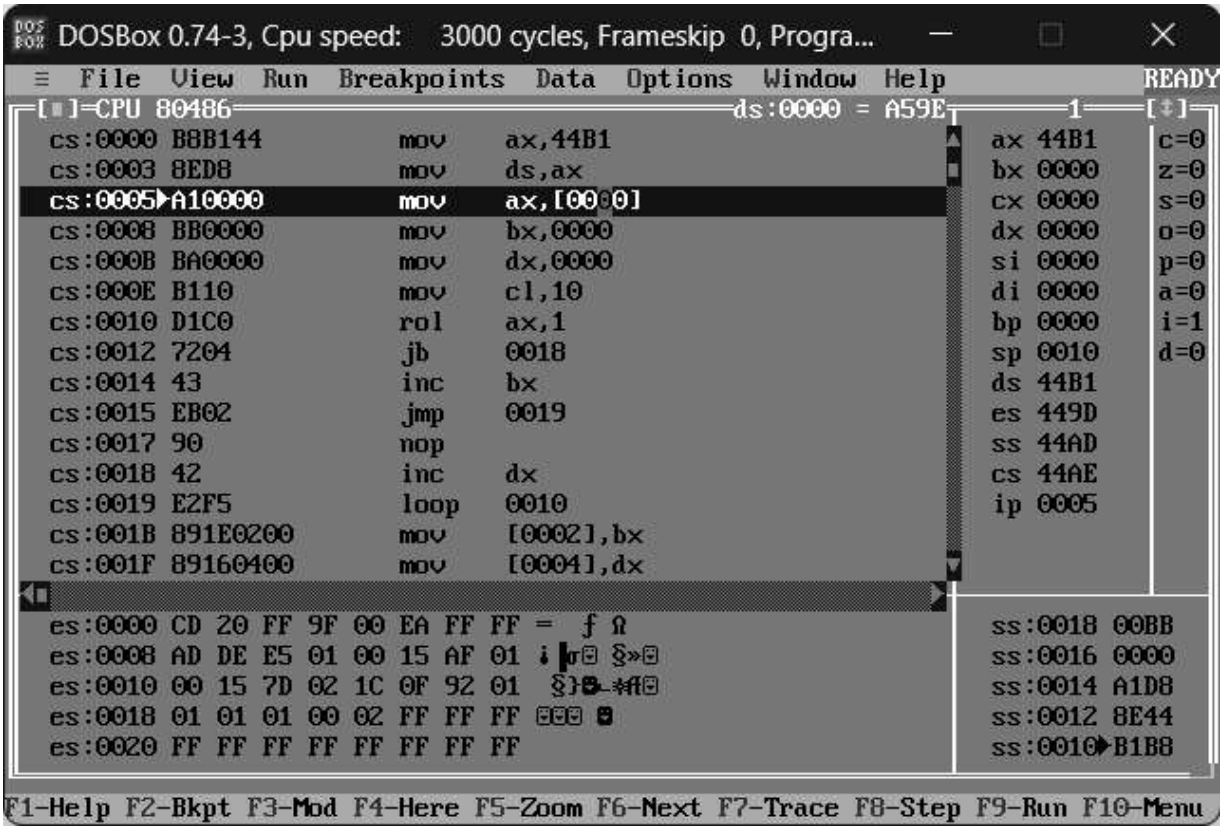
5. **Post Experiment Exercise:**
   a. Results/Calculations/Observations:
      i. Attach appropriate screenshots of internal registers, flag register and memory location along with the ALP.
      ii. Draw the flowchart for the above code.
   b. Questions:
      i. Write an ALP in TASM to find whether the number entered is odd or even. Attach appropriate screenshots.
   c. Conclusion:
      Write the conclusion/comments based on the experiment performed and the output obtained.
   d. References:
      Mention two book references and two web references.

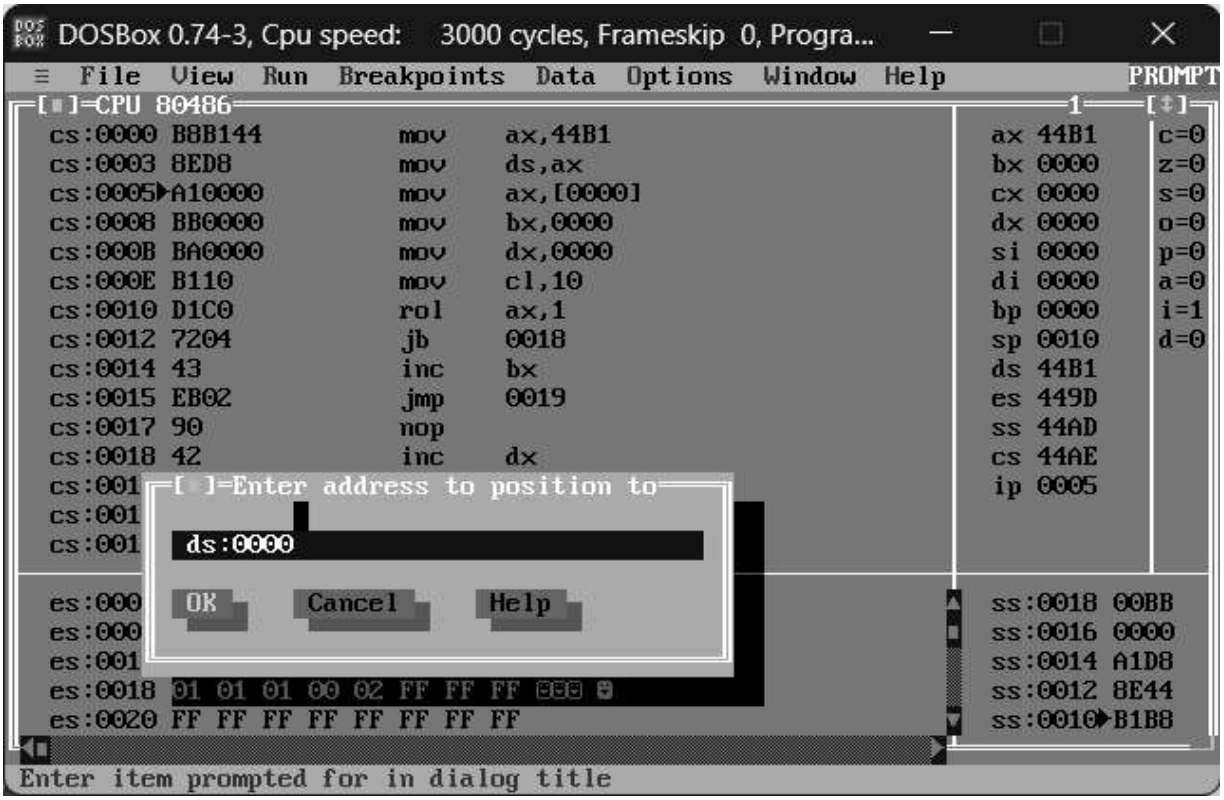
EXPERIMENT 5 MPL

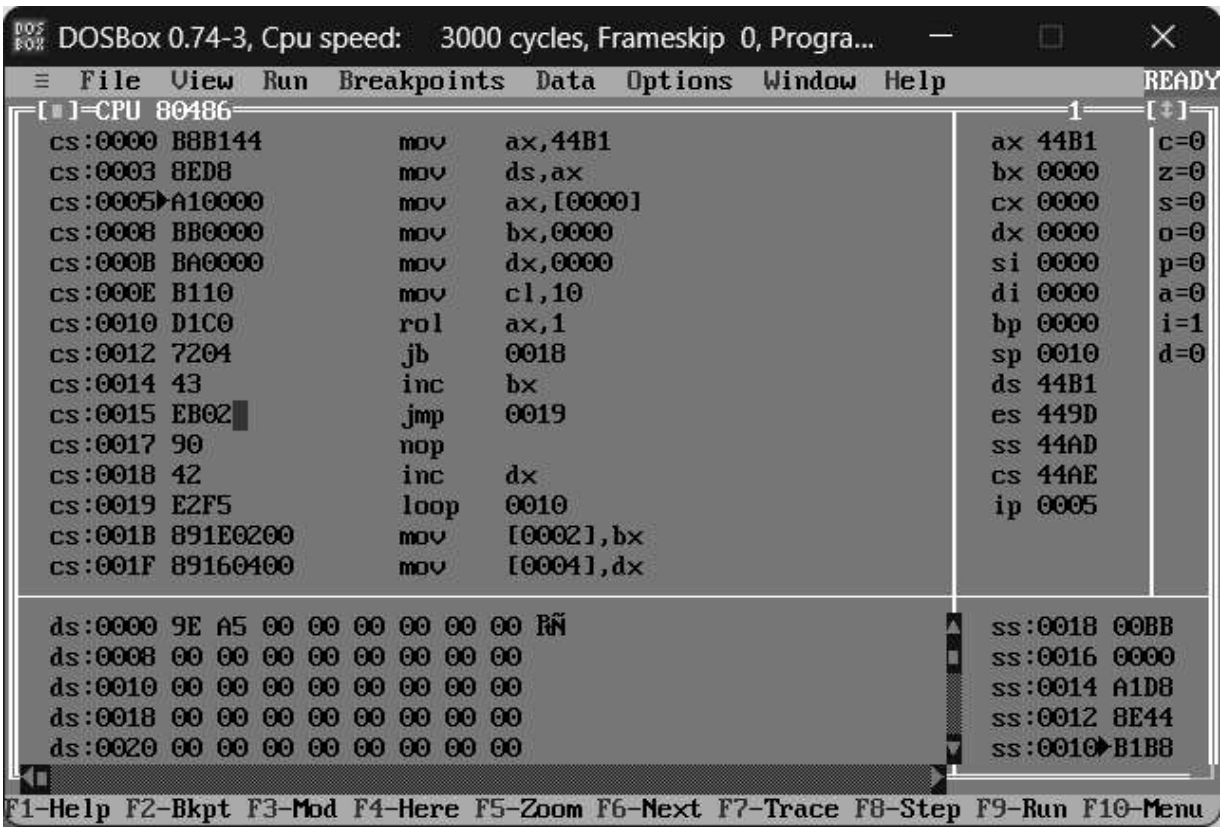
Write an ALP to count the number of 1's and 0's for a 16-bit number

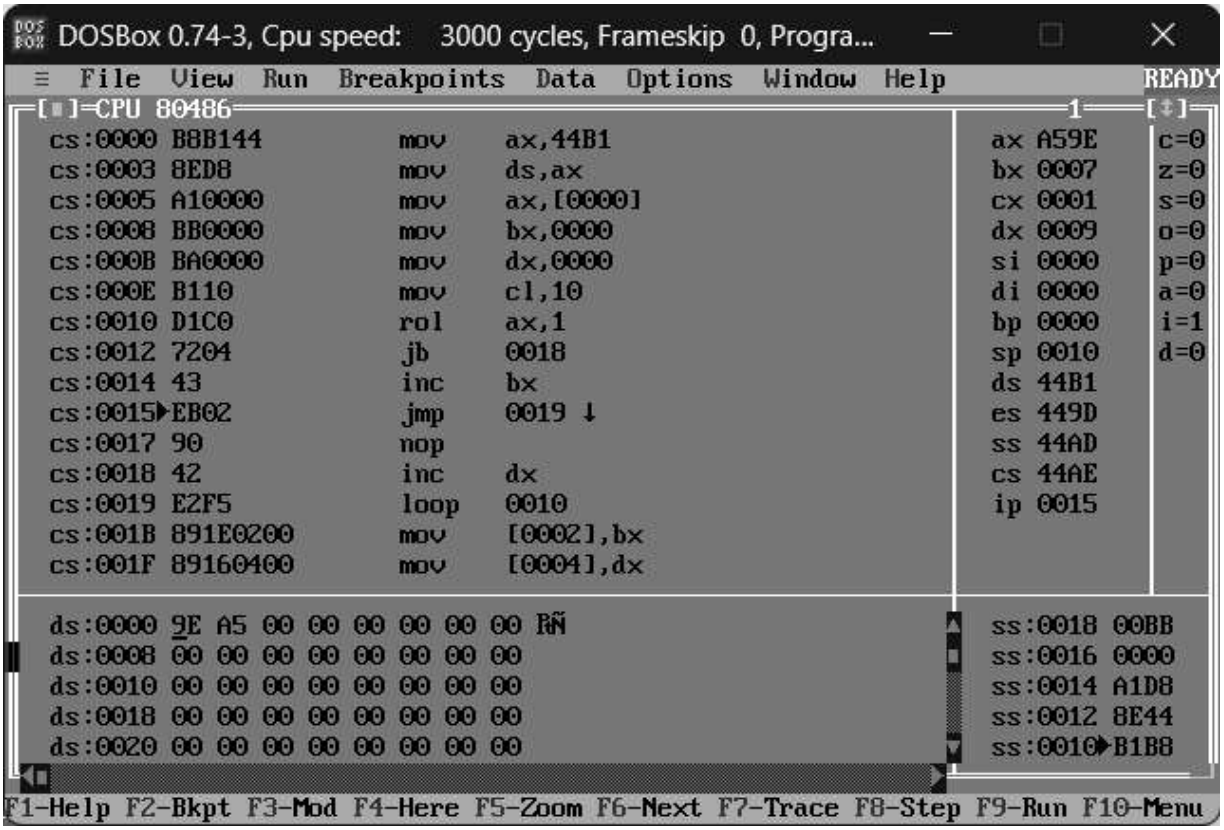
```
model small
stack 10h
data segment
        num dw 0A59Eh
        zeros dw 00h
        ones dw 00h
data ends
code segment
        assume cs:code,ds:data
start:
        mov ax,data
        mov ds,ax
        mov ax,num
        mov bx,00h
        mov dx,00h
        mov cl,10h
     up:rol ax,01h
        jc one
        inc bx
        jmp down
     one:inc dx
     down:loop up
          mov zeros,bx
          mov ones,dx
        mov ah,4ch
        int 21h
code ends
end start
```

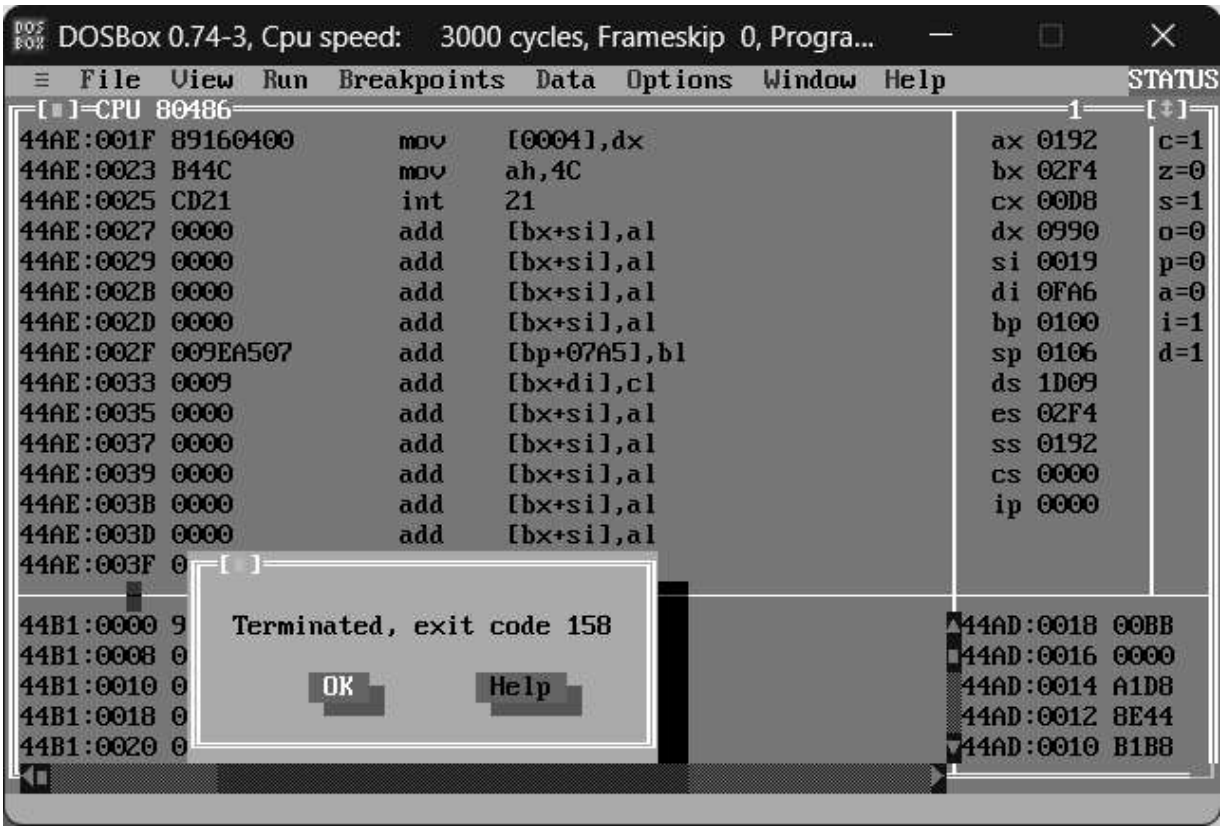

DOSBox 0.74-3, Cpu speed: 3000 cycles, Frameskip 0, Program: TD
File View Run Breakpoints Data Options Window Help
READY
CPU 80486
cs:0000 B8B144 mov ax,44B1
cs:0003 8ED8 mov ds,ax
cs:0005 A10000 mov ax,[0000]
cs:0008 BB0000 mov bx,0000
cs:000B BA0000 mov dx,0000
cs:000E B110 mov cl,10
cs:0010 D1C0 rol ax,1
cs:0012 7204 jb 0018
cs:0014 43 inc bx
cs:0015 EB02 jmp 0019
cs:0017 90 nop
cs:0018 42 inc dx
cs:0019 E2F5 loop 0010
cs:001B 891E0200 mov [0002],bx
cs:001F 89160400 mov [0004],dx
ds:0000 CD 20 FF 9F 00 EA FF FF
ds:0008 AD DE E5 01 00 15 AF 01
ds:0010 00 15 7D 02 1C 0F 92 01
ds:0018 01 01 01 00 02 FF FF FF
ds:0020 FF FF FF FF FF FF FF FF
ax 0000
bx 0000
cx 0000
dx 0000
si 0000
di 0000
bp 0000
sp 0010
ds 449D
es 449D
ss 44AD
cs 44AE
ip 0000
c=0
z=0
s=0
o=0
p=0
a=0
i=1
d=0
ss:0012 8E44
ss:0010 B1B8
ss:000E FFFF
ss:000C 0000
ss:000A 0000
F1-Help F2-Bkpt F3-Mod F4-Here F5-Zoom F6-Next F7-Trace F8-Step F9-Run F10-Menu

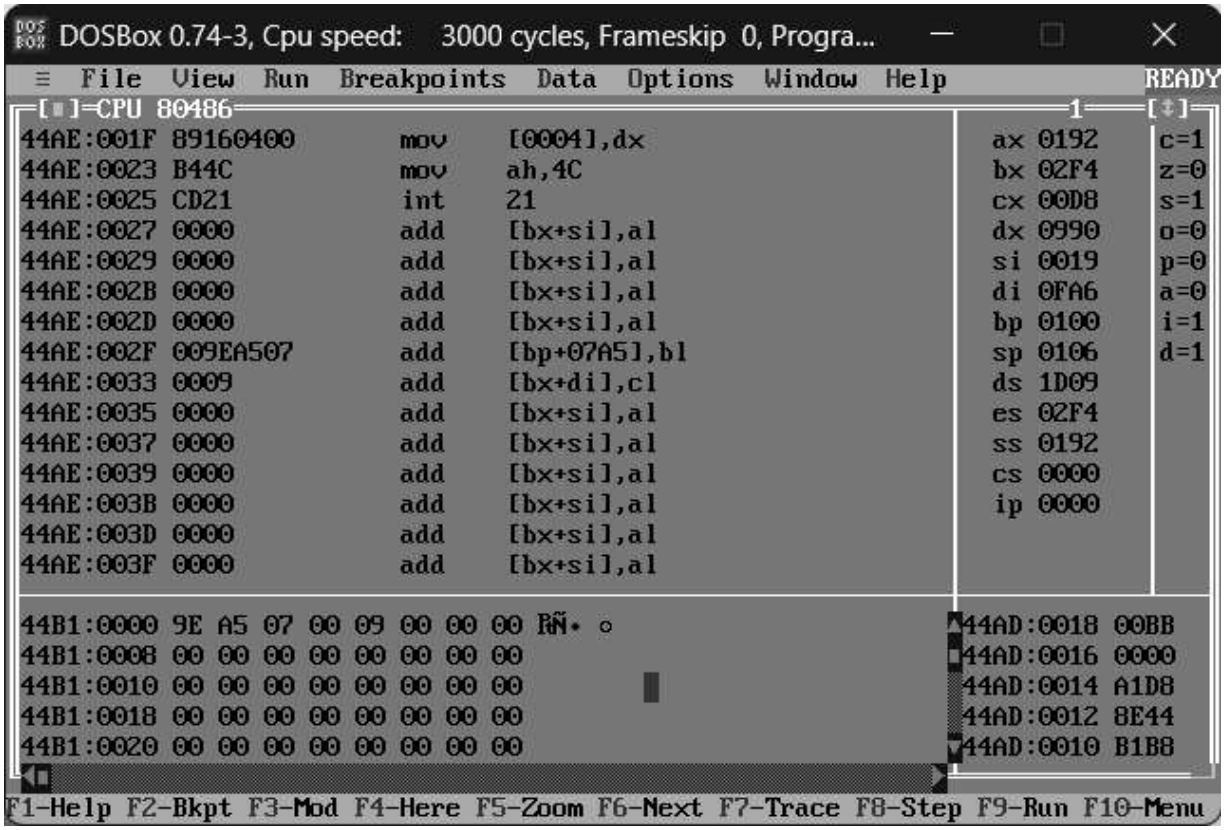

DOSBox 0.74-3, Cpu speed: 3000 cycles, Frameskip 0, Progra...
File View Run Breakpoints Data Options Window Help
READY
CPU 80486
ds:0000 = A59E
cs:0000 B8B144 mov ax,44B1
cs:0003 8ED8 mov ds,ax
cs:0005 A10000 mov ax,[0000]
cs:0008 BB0000 mov bx,0000
cs:000B BA0000 mov dx,0000
cs:000E B110 mov cl,10
cs:0010 D1C0 rol ax,1
cs:0012 7204 jb 0018
cs:0014 43 inc bx
cs:0015 EB02 jmp 0019
cs:0017 90 nop
cs:0018 42 inc dx
cs:0019 E2F5 loop 0010
cs:001B 891E0200 mov [0002],bx
cs:001F 89160400 mov [0004],dx
es:0000 CD 20 FF 9F 00 EA FF FF
es:0008 AD DE E5 01 00 15 AF 01
es:0010 00 15 7D 02 1C 0F 92 01
es:0018 01 01 01 00 02 FF FF FF
es:0020 FF FF FF FF FF FF FF FF
ax 44B1
bx 0000
cx 0000
dx 0000
si 0000
di 0000
bp 0000
sp 0010
ds 44B1
es 449D
ss 44AD
cs 44AE
ip 0005
c=0
z=0
s=0
o=0
p=0
a=0
i=1
d=0
ss:0018 00BB
ss:0016 0000
ss:0014 A1D8
ss:0012 8E44
ss:0010 B1B8
F1-Help F2-Bkpt F3-Mod F4-Here F5-Zoom F6-Next F7-Trace F8-Step F9-Run F10-Menu


DOSBox 0.74-3, Cpu speed: 3000 cycles, Frameskip 0, Progra...
File View Run Breakpoints Data Options Window Help
PROMPT
CPU 80486
cs:0000 B8B144 mov ax,44B1
cs:0003 8ED8 mov ds,ax
cs:0005 A10000 mov ax,[0000]
cs:0008 BB0000 mov bx,0000
cs:000B BA0000 mov dx,0000
cs:000E B110 mov cl,10
cs:0010 D1C0 rol ax,1
cs:0012 7204 jb 0018
cs:0014 43 inc bx
cs:0015 EB02 jmp 0019
cs:0017 90 nop
cs:0018 42 inc dx
ax 44B1
bx 0000
cx 0000
dx 0000
si 0000
di 0000
bp 0000
sp 0010
ds 44B1
es 449D
ss 44AD
cs 44AE
ip 0005
c=0
z=0
s=0
o=0
p=0
a=0
i=1
d=0
Enter address to position to
ds:0000
OK
Cancel
Help
es:0018 01 01 01 00 02 FF FF FF
es:0020 FF FF FF FF FF FF FF FF
ss:0018 00BB
ss:0016 0000
ss:0014 A1D8
ss:0012 8E44
ss:0010 B1B8
Enter item prompted for in dialog title


DOSBox 0.74-3, Cpu speed: 3000 cycles, Frameskip 0, Progra...
File View Run Breakpoints Data Options Window Help
READY
CPU 80486
cs:0000 B8B144 mov ax,44B1
cs:0003 8ED8 mov ds,ax
cs:0005 A10000 mov ax,[0000]
cs:0008 BB0000 mov bx,0000
cs:000B BA0000 mov dx,0000
cs:000E B110 mov cl,10
cs:0010 D1C0 rol ax,1
cs:0012 7204 jb 0018
cs:0014 43 inc bx
cs:0015 EB02 jmp 0019
cs:0017 90 nop
cs:0018 42 inc dx
cs:0019 E2F5 loop 0010
cs:001B 891E0200 mov [0002],bx
cs:001F 89160400 mov [0004],dx
ax 44B1
bx 0000
cx 0000
dx 0000
si 0000
di 0000
bp 0000
sp 0010
ds 44B1
es 449D
ss 44AD
cs 44AE
ip 0005
c=0
z=0
s=0
o=0
p=0
a=0
i=1
d=0
ds:0000 9E A5 00 00 00 00 00 00
ds:0008 00 00 00 00 00 00 00 00
ds:0010 00 00 00 00 00 00 00 00
ds:0018 00 00 00 00 00 00 00 00
ds:0020 00 00 00 00 00 00 00 00
ss:0018 00BB
ss:0016 0000
ss:0014 A1D8
ss:0012 8E44
ss:0010 B1B8
F1-Help F2-Bkpt F3-Mod F4-Here F5-Zoom F6-Next F7-Trace F8-Step F9-Run F10-Menu


DOSBox 0.74-3, Cpu speed: 3000 cycles, Frameskip 0, Progra...
File View Run Breakpoints Data Options Window Help
READY
CPU 80486
cs:0000 B8B144 mov ax,44B1
cs:0003 8ED8 mov ds,ax
cs:0005 A10000 mov ax,[0000]
cs:0008 BB0000 mov bx,0000
cs:000B BA0000 mov dx,0000
cs:000E B110 mov cl,10
cs:0010 D1C0 rol ax,1
cs:0012 7204 jb 0018
cs:0014 43 inc bx
cs:0015 EB02 jmp 0019
cs:0017 90 nop
cs:0018 42 inc dx
cs:0019 E2F5 loop 0010
cs:001B 891E0200 mov [0002],bx
cs:001F 89160400 mov [0004],dx
ax A59E
bx 0007
cx 0001
dx 0009
si 0000
di 0000
bp 0000
sp 0010
ds 44B1
es 449D
ss 44AD
cs 44AE
ip 0015
c=0
z=0
s=0
o=0
p=0
a=0
i=1
d=0
ds:0000 9E A5 00 00 00 00 00 00
ds:0008 00 00 00 00 00 00 00 00
ds:0010 00 00 00 00 00 00 00 00
ds:0018 00 00 00 00 00 00 00 00
ds:0020 00 00 00 00 00 00 00 00
ss:0018 00BB
ss:0016 0000
ss:0014 A1D8
ss:0012 8E44
ss:0010 B1B8
F1-Help F2-Bkpt F3-Mod F4-Here F5-Zoom F6-Next F7-Trace F8-Step F9-Run F10-Menu


DOSBox 0.74-3, Cpu speed: 3000 cycles, Frameskip 0, Progra...
File View Run Breakpoints Data Options Window Help
STATUS
CPU 80486
44AE:001F 89160400 mov [0004],dx
44AE:0023 B44C mov ah,4C
44AE:0025 CD21 int 21
44AE:0027 0000 add [bx+si],al
44AE:0029 0000 add [bx+si],al
44AE:002B 0000 add [bx+si],al
44AE:002D 0000 add [bx+si],al
44AE:002F 009EA507 add [bp+07A5],bl
44AE:0033 0009 add [bx+di],cl
44AE:0035 0000 add [bx+si],al
44AE:0037 0000 add [bx+si],al
44AE:0039 0000 add [bx+si],al
44AE:003B 0000 add [bx+si],al
44AE:003D 0000 add [bx+si],al
44AE:003F 0
ax 0192
bx 02F4
cx 00D8
dx 0990
si 0019
di 0FA6
bp 0100
sp 0106
ds 1D09
es 02F4
ss 0192
cs 0000
ip 0000
c=1
z=0
s=1
o=0
p=0
a=0
i=1
d=1
Terminated, exit code 158
OK
Help
44B1:0000 9
44B1:0008 0
44B1:0010 0
44B1:0018 0
44B1:0020 0
44AD:0018 00BB
44AD:0016 0000
44AD:0014 A1D8
44AD:0012 8E44
44AD:0010 B1B8

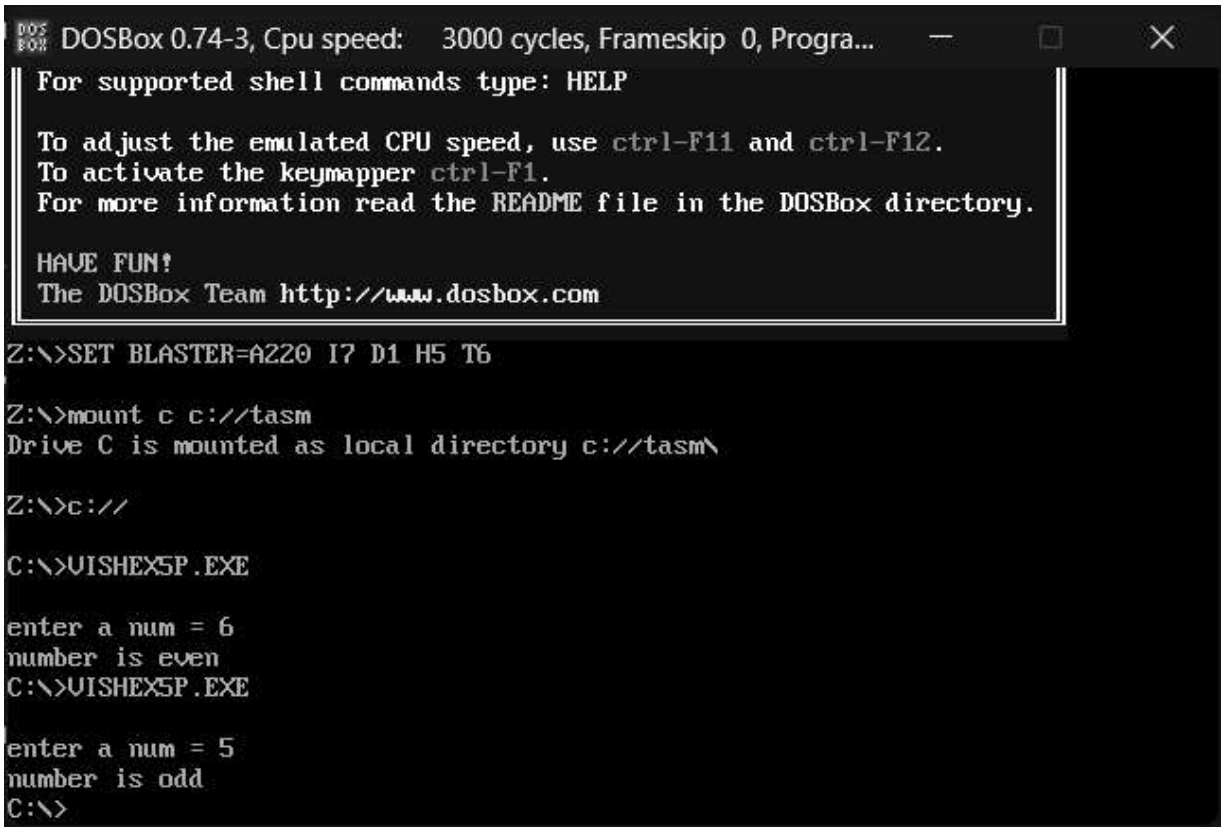

DOSBox 0.74-3, Cpu speed: 3000 cycles, Frameskip 0, Progra...
File View Run Breakpoints Data Options Window Help
READY
CPU 80486
44AE:001F 89160400 mov [0004],dx
44AE:0023 B44C mov ah,4C
44AE:0025 CD21 int 21
44AE:0027 0000 add [bx+si],al
44AE:0029 0000 add [bx+si],al
44AE:002B 0000 add [bx+si],al
44AE:002D 0000 add [bx+si],al
44AE:002F 009EA507 add [bp+07A5],bl
44AE:0033 0009 add [bx+di],cl
44AE:0035 0000 add [bx+si],al
44AE:0037 0000 add [bx+si],al
44AE:0039 0000 add [bx+si],al
44AE:003B 0000 add [bx+si],al
44AE:003D 0000 add [bx+si],al
44AE:003F 0000 add [bx+si],al
ax 0192
bx 02F4
cx 00D8
dx 0990
si 0019
di 0FA6
bp 0100
sp 0106
ds 1D09
es 02F4
ss 0192
cs 0000
ip 0000
c=1
z=0
s=1
o=0
p=0
a=0
i=1
d=1
44B1:0000 9E A5 07 00 09 00 00 00
44B1:0008 00 00 00 00 00 00 00 00
44B1:0010 00 00 00 00 00 00 00 00
44B1:0018 00 00 00 00 00 00 00 00
44B1:0020 00 00 00 00 00 00 00 00
44AD:0018 00BB
44AD:0016 0000
44AD:0014 A1D8
44AD:0012 8E44
44AD:0010 B1B8
F1-Help F2-Bkpt F3-Mod F4-Here F5-Zoom F6-Next F7-Trace F8-Step F9-Run F10-Menu

POST EXPERIMENT :

Q1.Write an ALP in TASM to find whether the number entered is odd or even.

```
assume cs:code, ds:data
data segment
        msg db 10,13, 'enter a num = $'
        msg1 db 10,13, 'number is even $'
        msg2 db 10,13, 'number is odd $'
data ends
code segment
Start:
        mov bx, data
        mov ds, bx
        print macro message
                lea dx, message
                mov ah, 09h
                int 21h
        endm
        print msg
        mov ah, 01h
        int 21h
        sar al, 01
        jc odd
        print msg1
        jmp terminate
 Odd:
         print msg2
 terminate:
        mov ah, 4ch
        int 21h
code ends
end start
```

Output:


DOSBox 0.74-3, Cpu speed: 3000 cycles, Frameskip 0, Progra...
For supported shell commands type: HELP
To adjust the emulated CPU speed, use ctrl-F11 and ctrl-F12.
To activate the keymapper ctrl-F1.
For more information read the README file in the DOSBox directory.
HAVE FUN!
The DOSBox Team http://www.dosbox.com
Z:\>SET BLASTER=A220 I7 D1 H5 T6
Z:\>mount c c://tasm
Drive C is mounted as local directory c://tasm\
Z:\>c://
C:\>VISHEX5P.EXE
enter a num = 6
number is even
C:\>VISHEX5P.EXE
enter a num = 5
number is odd
C:\>