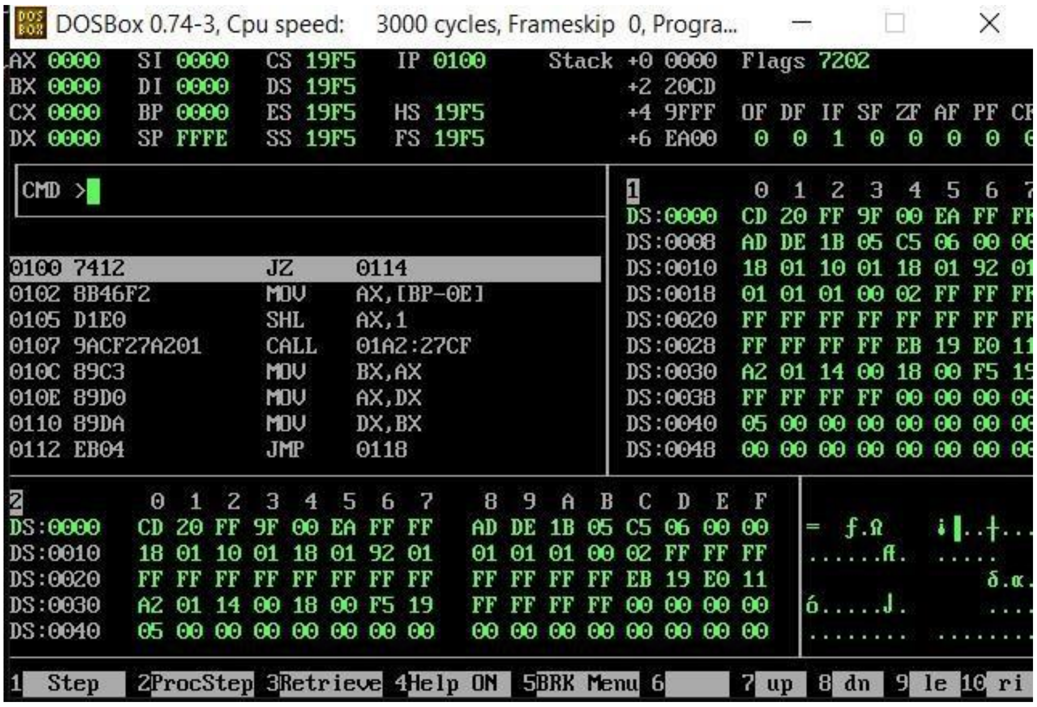
Lab 2

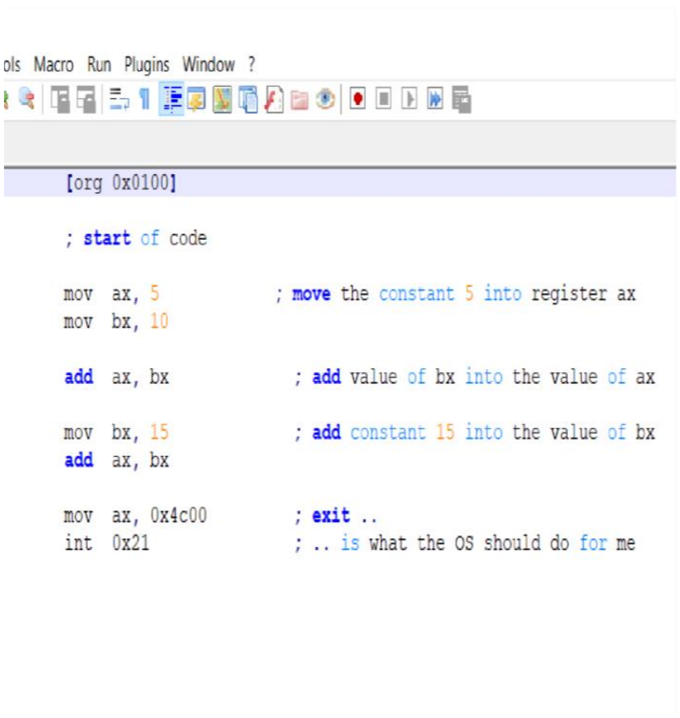
Title: Getting Familiar with our Virtual Machine.



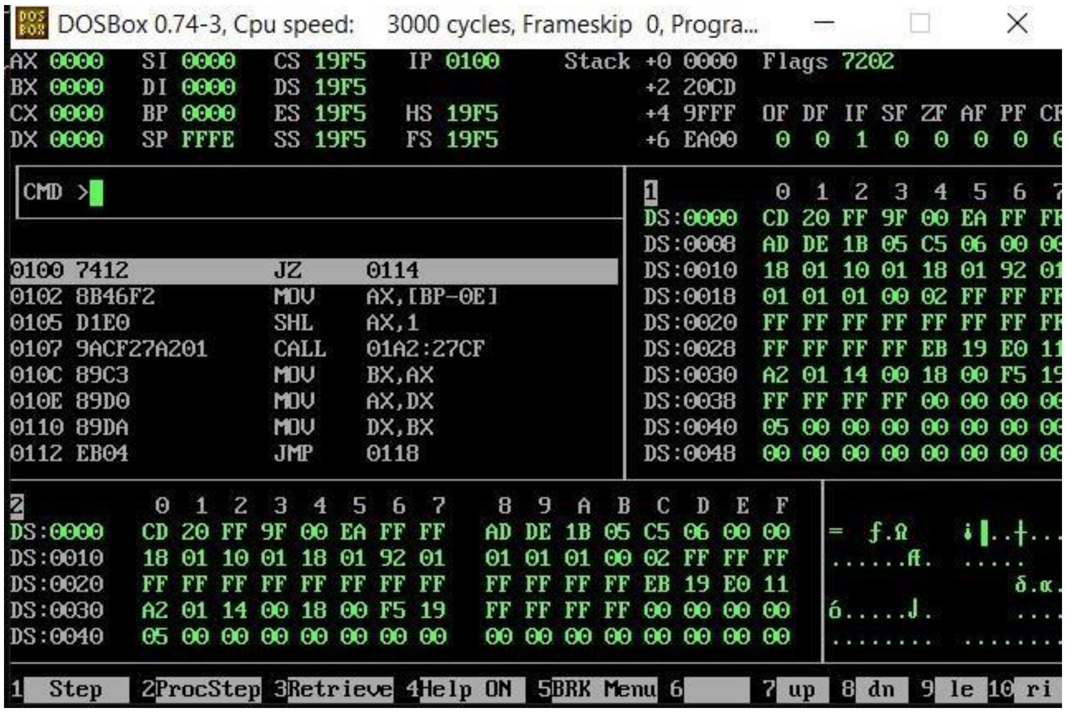
```
DOSBox 0.74-3, Cpu speed:   3000 cycles, Frameskip  0, Progra...
AX 0000   SI 0000   CS 19F5   IP 0100     Stack +0 0000  Flags 7202
BX 0000   DI 0000   DS 19F5                     +2 20CD
CX 0000   BP 0000   ES 19F5   HS 19F5           +4 9FFF  OF DF IF SF ZF AF PF CF
DX 0000   SP FFFE   SS 19F5   FS 19F5           +6 EA00   0  0  1  0  0  0  0  0

 CMD >                                          1         0  1  2  3  4  5  6  7
                                                DS:0000  CD 20 FF 9F 00 EA FF FF
                                                DS:0008  AD DE 1B 05 C5 06 00 00
0100 7412          JZ    0114                   DS:0010  18 01 10 01 18 01 92 01
0102 8B46F2        MOV   AX,[BP-0E]             DS:0018  01 01 01 00 02 FF FF FF
0105 D1E0          SHL   AX,1                   DS:0020  FF FF FF FF FF FF FF FF
0107 9ACF27A201    CALL  01A2:27CF              DS:0028  FF FF FF FF EB 19 E0 11
010C 89C3          MOV   BX,AX                  DS:0030  A2 01 14 00 18 00 F5 19
010E 89D0          MOV   AX,DX                  DS:0038  FF FF FF FF 00 00 00 00
0110 89DA          MOV   DX,BX                  DS:0040  05 00 00 00 00 00 00 00
0112 EB04          JMP   0118                   DS:0048  00 00 00 00 00 00 00 00

2         0  1  2  3  4  5  6  7    8  9  A  B  C  D  E  F
DS:0000  CD 20 FF 9F 00 EA FF FF   AD DE 1B 05 C5 06 00 00   = ƒ.Ω    ¡▌..┼...
DS:0010  18 01 10 01 18 01 92 01   01 01 01 00 02 FF FF FF   ......Æ.  .....
DS:0020  FF FF FF FF FF FF FF FF   FF FF FF FF EB 19 E0 11             δ.α.
DS:0030  A2 01 14 00 18 00 F5 19   FF FF FF FF 00 00 00 00   ó.....⌡.      ....
DS:0040  05 00 00 00 00 00 00 00   00 00 00 00 00 00 00 00   ........  ........

1 Step  2ProcStep 3Retrieve 4Help ON  5BRK Menu 6      7 up  8 dn  9 le 10 ri
```

This is how we visualize the working of our hardware components. On the top corner are ax,bx,cx registers followed by the value they hold in hexadecimal. Below this is our command line input, where we can enter different commands such as m1 0100 to move to memory location 0100 and so on. Below this we see in first column our memory location which can followed by opcodes and command or the values we have coded while using notepad++.

Title: Viewing Execution of Code.



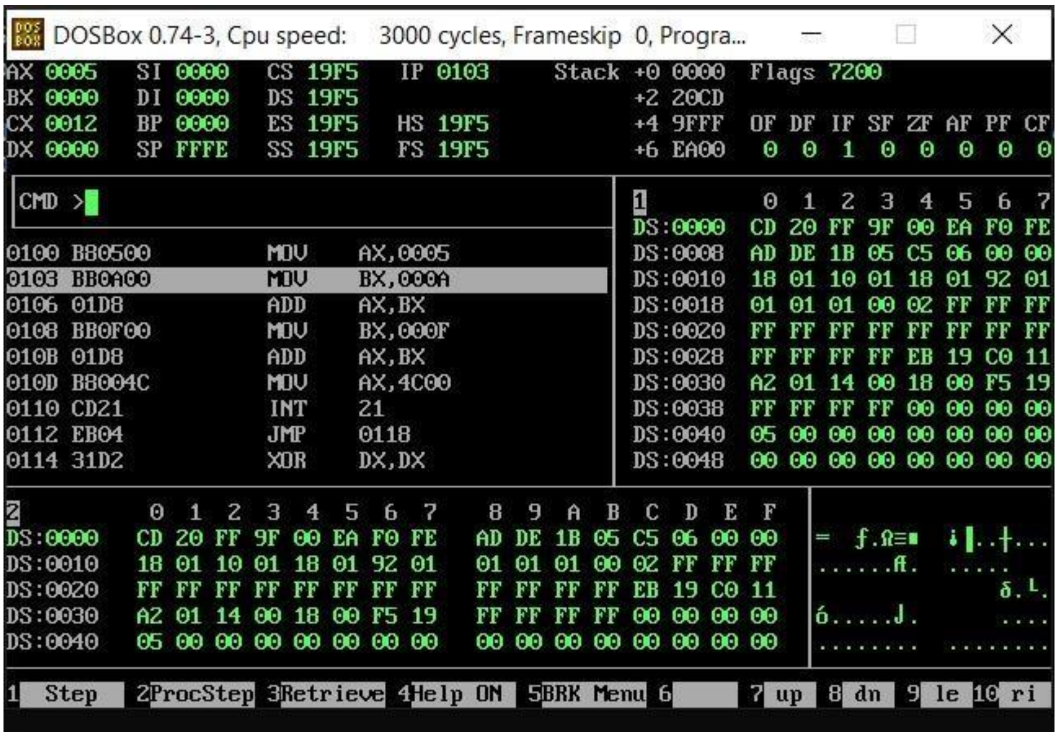
```
[org 0x0100]

; start of code

mov  ax, 5          ; move the constant 5 into register ax
mov  bx, 10

add  ax, bx           ; add value of bx into the value of ax

mov  bx, 15           ; add constant 15 into the value of bx
add  ax, bx

mov  ax, 0x4c00       ; exit ..
int  0x21             ; .. is what the OS should do for me
```



DOSBox 0.74-3, Cpu speed: 3000 cycles, Frameskip 0, Progra...

```
AX 0000   SI 0000   CS 19F5   IP 0100     Stack +0 0000   Flags 7202
BX 0000   DI 0000   DS 19F5                     +2 20CD
CX 0000   BP 0000   ES 19F5   HS 19F5           +4 9FFF   OF DF IF SF ZF AF PF CF
DX 0000   SP FFFE   SS 19F5   FS 19F5           +6 EA00    0  0  1  0  0  0  0  0

 CMD >                                         1       0  1  2  3  4  5  6  7
                                               DS:0000 CD 20 FF 9F 00 EA FF FF
                                               DS:0008 AD DE 1B 05 C5 06 00 00
0100 7412           JZ     0114                DS:0010 18 01 10 01 18 01 92 01
0102 8B46F2         MOV    AX,[BP-0E]          DS:0018 01 01 01 00 02 FF FF FF
0105 D1E0           SHL    AX,1                DS:0020 FF FF FF FF FF FF FF FF
0107 9ACF27A201     CALL   01A2:27CF           DS:0028 FF FF FF FF EB 19 E0 11
010C 89C3           MOV    BX,AX               DS:0030 A2 01 14 00 18 00 F5 19
010E 89D0           MOV    AX,DX               DS:0038 FF FF FF FF 00 00 00 00
0110 89DA           MOV    DX,BX               DS:0040 05 00 00 00 00 00 00 00
0112 EB04           JMP    0118                DS:0048 00 00 00 00 00 00 00 00

2          0  1  2  3  4  5  6  7   8  9  A  B  C  D  E  F
DS:0000   CD 20 FF 9F 00 EA FF FF  AD DE 1B 05 C5 06 00 00   =  ƒ.Ω     ¡▌..┼...
DS:0010   18 01 10 01 18 01 92 01  01 01 01 00 02 FF FF FF   ......Æ.  .....
DS:0020   FF FF FF FF FF FF FF FF  FF FF FF FF EB 19 E0 11             δ.α.
DS:0030   A2 01 14 00 18 00 F5 19  FF FF FF FF 00 00 00 00   ó.....⌡.      ....
DS:0040   05 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ........  ........

1 Step  2ProcStep 3Retrieve 4Help ON 5BRK Menu 6      7 up  8 dn  9 le 10 ri
```

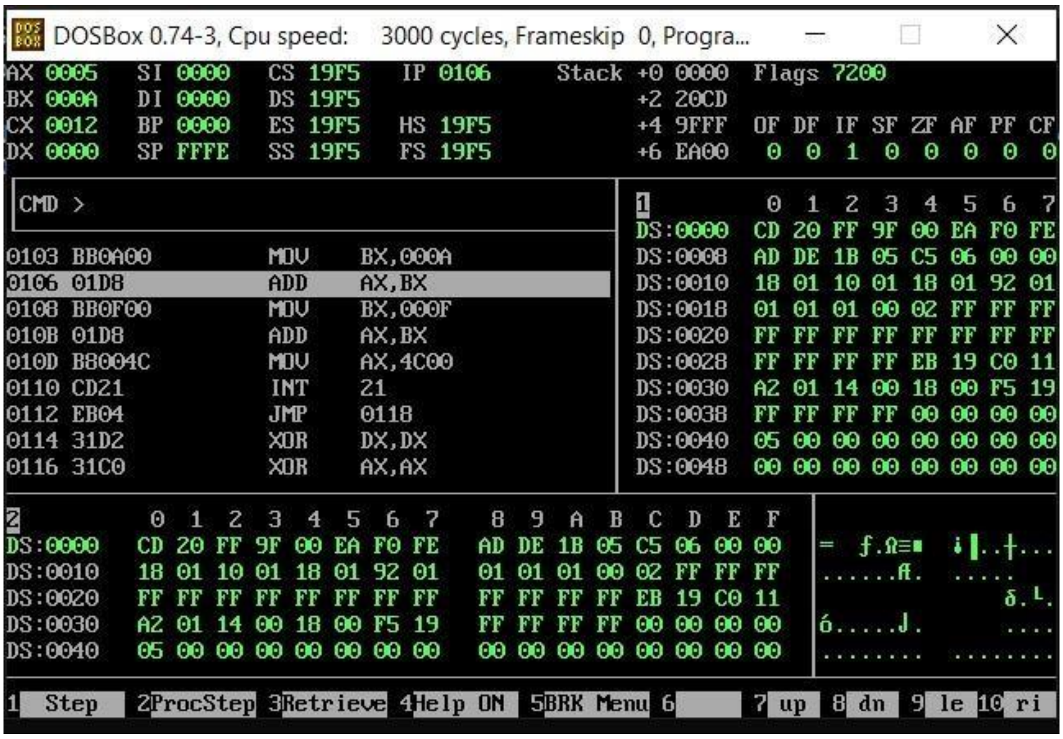
So, whenever we press f2 key, each line of our code is executed and we are able to view how well it performs. We are using afd for this execution and debugging.



DOSBox 0.74-3, Cpu speed: 3000 cycles, Frameskip 0, Progra...

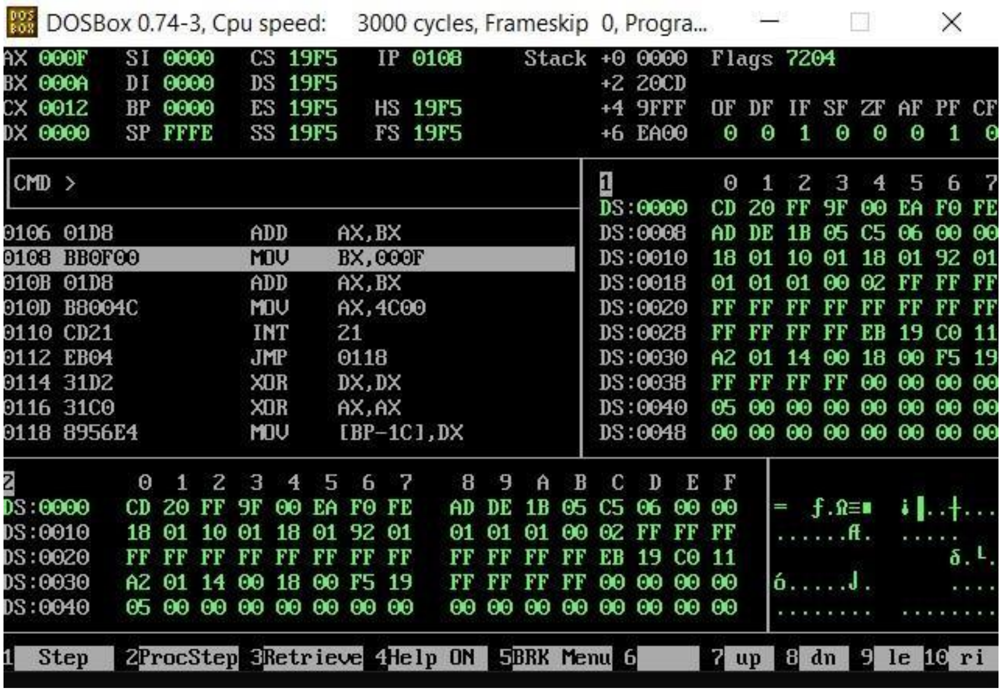
```
AX 0005   SI 0000   CS 19F5   IP 0103     Stack +0 0000   Flags 7200
BX 0000   DI 0000   DS 19F5                     +2 20CD
CX 0012   BP 0000   ES 19F5   HS 19F5           +4 9FFF   OF DF IF SF ZF AF PF CF
DX 0000   SP FFFE   SS 19F5   FS 19F5           +6 EA00    0  0  1  0  0  0  0  0

 CMD >                                         1       0  1  2  3  4  5  6  7
                                               DS:0000 CD 20 FF 9F 00 EA F0 FE
0100 BB0500         MOV    AX,0005             DS:0008 AD DE 1B 05 C5 06 00 00
0103 BB0A00         MOV    BX,000A             DS:0010 18 01 10 01 18 01 92 01
0106 01D8           ADD    AX,BX               DS:0018 01 01 01 00 02 FF FF FF
0108 BB0F00         MOV    BX,000F             DS:0020 FF FF FF FF FF FF FF FF
010B 01D8           ADD    AX,BX               DS:0028 FF FF FF FF EB 19 C0 11
010D BB004C         MOV    AX,4C00             DS:0030 A2 01 14 00 18 00 F5 19
0110 CD21           INT    21                  DS:0038 FF FF FF FF 00 00 00 00
0112 EB04           JMP    0118                DS:0040 05 00 00 00 00 00 00 00
0114 31D2           XOR    DX,DX               DS:0048 00 00 00 00 00 00 00 00

2          0  1  2  3  4  5  6  7   8  9  A  B  C  D  E  F
DS:0000   CD 20 FF 9F 00 EA F0 FE  AD DE 1B 05 C5 06 00 00   =  ƒ.Ω≡■   ¡▌..┼...
DS:0010   18 01 10 01 18 01 92 01  01 01 01 00 02 FF FF FF   ......Æ.  .....
DS:0020   FF FF FF FF FF FF FF FF  FF FF FF FF EB 19 C0 11             δ.└.
DS:0030   A2 01 14 00 18 00 F5 19  FF FF FF FF 00 00 00 00   ó.....⌡.      ....
DS:0040   05 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ........  ........

1 Step  2ProcStep 3Retrieve 4Help ON 5BRK Menu 6      7 up  8 dn  9 le 10 ri
```

The value is added to ax.



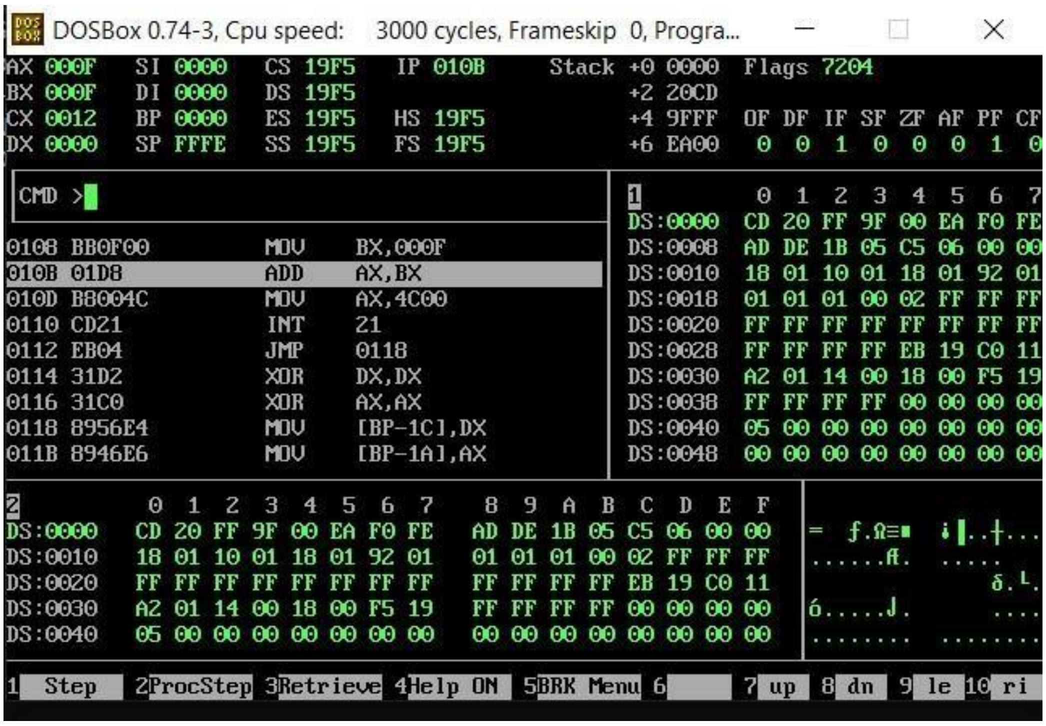
```
DOSBox 0.74-3, Cpu speed:    3000 cycles, Frameskip  0, Progra...
AX 0005   SI 0000   CS 19F5   IP 0106     Stack +0 0000  Flags 7200
BX 000A   DI 0000   DS 19F5                     +2 20CD
CX 0012   BP 0000   ES 19F5   HS 19F5           +4 9FFF   OF DF IF SF ZF AF PF CF
DX 0000   SP FFFE   SS 19F5   FS 19F5           +6 EA00    0  0  1  0  0  0  0  0

CMD >                                          1          0  1  2  3  4  5  6  7
                                               DS:0000  CD 20 FF 9F 00 EA F0 FE
0103 BB0A00         MOV     BX,000A            DS:0008  AD DE 1B 05 C5 06 00 00
0106 01D8           ADD     AX,BX              DS:0010  18 01 10 01 18 01 92 01
0108 BB0F00         MOV     BX,000F            DS:0018  01 01 01 00 02 FF FF FF
010B 01D8           ADD     AX,BX              DS:0020  FF FF FF FF FF FF FF FF
010D B8004C         MOV     AX,4C00            DS:0028  FF FF FF FF EB 19 C0 11
0110 CD21           INT     21                 DS:0030  A2 01 14 00 18 00 F5 19
0112 EB04           JMP     0118               DS:0038  FF FF FF FF 00 00 00 00
0114 31D2           XOR     DX,DX              DS:0040  05 00 00 00 00 00 00 00
0116 31C0           XOR     AX,AX              DS:0048  00 00 00 00 00 00 00 00

2          0  1  2  3  4  5  6  7   8  9  A  B  C  D  E  F
DS:0000   CD 20 FF 9F 00 EA F0 FE  AD DE 1B 05 C5 06 00 00   = ƒ.Ω≡■  ¡▌..┼...
DS:0010   18 01 10 01 18 01 92 01  01 01 01 00 02 FF FF FF   ......Æ.  .....
DS:0020   FF FF FF FF FF FF FF FF  FF FF FF FF EB 19 C0 11             δ.└.
DS:0030   A2 01 14 00 18 00 F5 19  FF FF FF FF 00 00 00 00   ó.....J.      ....
DS:0040   05 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ........  ........

1 Step  2ProcStep 3Retrieve 4Help ON  5BRK Menu 6      7 up  8 dn  9 le 10 ri
```

The value is added to bx



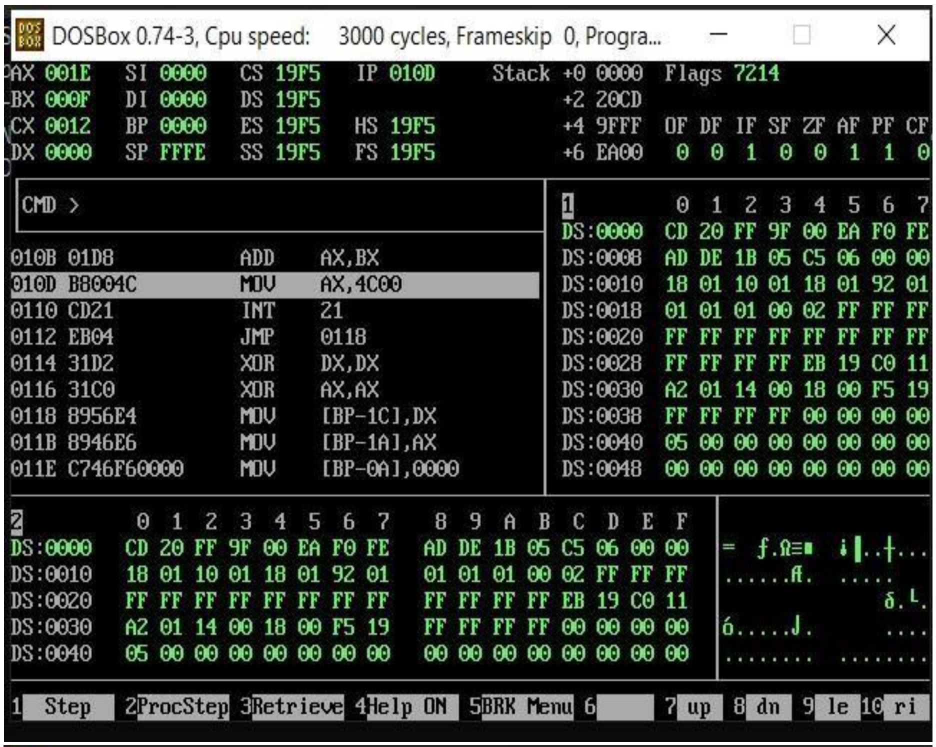
```
DOSBox 0.74-3, Cpu speed:    3000 cycles, Frameskip  0, Progra...
AX 000F   SI 0000   CS 19F5   IP 0108     Stack +0 0000  Flags 7204
BX 000A   DI 0000   DS 19F5                     +2 20CD
CX 0012   BP 0000   ES 19F5   HS 19F5           +4 9FFF   OF DF IF SF ZF AF PF CF
DX 0000   SP FFFE   SS 19F5   FS 19F5           +6 EA00    0  0  1  0  0  0  1  0

CMD >                                          1          0  1  2  3  4  5  6  7
                                               DS:0000  CD 20 FF 9F 00 EA F0 FE
0106 01D8           ADD     AX,BX              DS:0008  AD DE 1B 05 C5 06 00 00
0108 BB0F00         MOV     BX,000F            DS:0010  18 01 10 01 18 01 92 01
010B 01D8           ADD     AX,BX              DS:0018  01 01 01 00 02 FF FF FF
010D B8004C         MOV     AX,4C00            DS:0020  FF FF FF FF FF FF FF FF
0110 CD21           INT     21                 DS:0028  FF FF FF FF EB 19 C0 11
0112 EB04           JMP     0118               DS:0030  A2 01 14 00 18 00 F5 19
0114 31D2           XOR     DX,DX              DS:0038  FF FF FF FF 00 00 00 00
0116 31C0           XOR     AX,AX              DS:0040  05 00 00 00 00 00 00 00
0118 8956E4         MOV     [BP-1C],DX         DS:0048  00 00 00 00 00 00 00 00

2          0  1  2  3  4  5  6  7   8  9  A  B  C  D  E  F
DS:0000   CD 20 FF 9F 00 EA F0 FE  AD DE 1B 05 C5 06 00 00   = ƒ.Ω≡■  ¡▌..┼...
DS:0010   18 01 10 01 18 01 92 01  01 01 01 00 02 FF FF FF   ......Æ.  .....
DS:0020   FF FF FF FF FF FF FF FF  FF FF FF FF EB 19 C0 11             δ.└.
DS:0030   A2 01 14 00 18 00 F5 19  FF FF FF FF 00 00 00 00   ó.....J.      ....
DS:0040   05 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ........  ........

1 Step  2ProcStep 3Retrieve 4Help ON  5BRK Menu 6      7 up  8 dn  9 le 10 ri
```

The value is moved from bx to ax

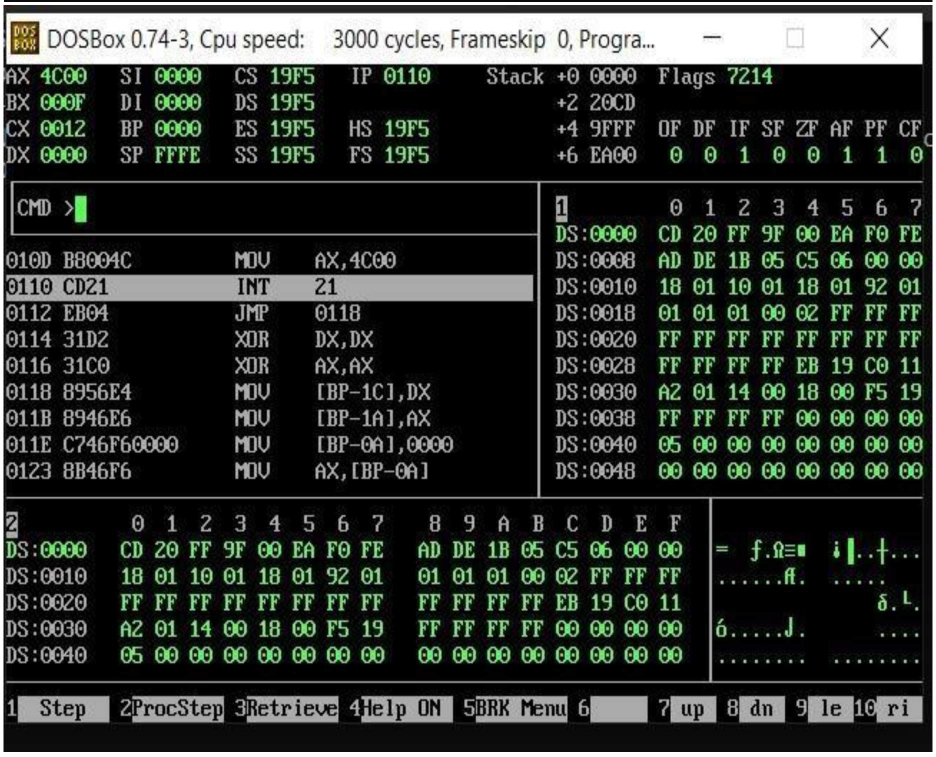

DOSBox 0.74-3, Cpu speed: 3000 cycles, Frameskip 0, Progra...

```
AX 000F   SI 0000   CS 19F5   IP 010B     Stack +0 0000  Flags 7204
BX 000F   DI 0000   DS 19F5                     +2 20CD
CX 0012   BP 0000   ES 19F5   HS 19F5           +4 9FFF  OF DF IF SF ZF AF PF CF
DX 0000   SP FFFE   SS 19F5   FS 19F5           +6 EA00   0  0  1  0  0  0  1  0

 CMD >                                        1          0  1  2  3  4  5  6  7
                                              DS:0000  CD 20 FF 9F 00 EA F0 FE
0108 BB0F00         MOV     BX,000F           DS:0008  AD DE 1B 05 C5 06 00 00
010B 01D8           ADD     AX,BX             DS:0010  18 01 10 01 18 01 92 01
010D B8004C         MOV     AX,4C00           DS:0018  01 01 01 00 02 FF FF FF
0110 CD21           INT     21                DS:0020  FF FF FF FF FF FF FF FF
0112 EB04           JMP     0118              DS:0028  FF FF FF FF EB 19 C0 11
0114 31D2           XOR     DX,DX             DS:0030  A2 01 14 00 18 00 F5 19
0116 31C0           XOR     AX,AX             DS:0038  FF FF FF FF 00 00 00 00
0118 8956E4         MOV     [BP-1C],DX        DS:0040  05 00 00 00 00 00 00 00
011B 8946E6         MOV     [BP-1A],AX        DS:0048  00 00 00 00 00 00 00 00

2        0  1  2  3  4  5  6  7    8  9  A  B  C  D  E  F
DS:0000  CD 20 FF 9F 00 EA F0 FE   AD DE 1B 05 C5 06 00 00   = ƒ.Ω≡■  ↓▌..┼...
DS:0010  18 01 10 01 18 01 92 01   01 01 01 00 02 FF FF FF   ......Æ.  .....
DS:0020  FF FF FF FF FF FF FF FF   FF FF FF FF EB 19 C0 11             δ.└.
DS:0030  A2 01 14 00 18 00 F5 19   FF FF FF FF 00 00 00 00   ó.....⌡.      ....
DS:0040  05 00 00 00 00 00 00 00   00 00 00 00 00 00 00 00   ........  ........

1 Step  2ProcStep 3Retrieve 4Help ON  5BRK Menu 6     7 up  8 dn  9 le 10 ri
```

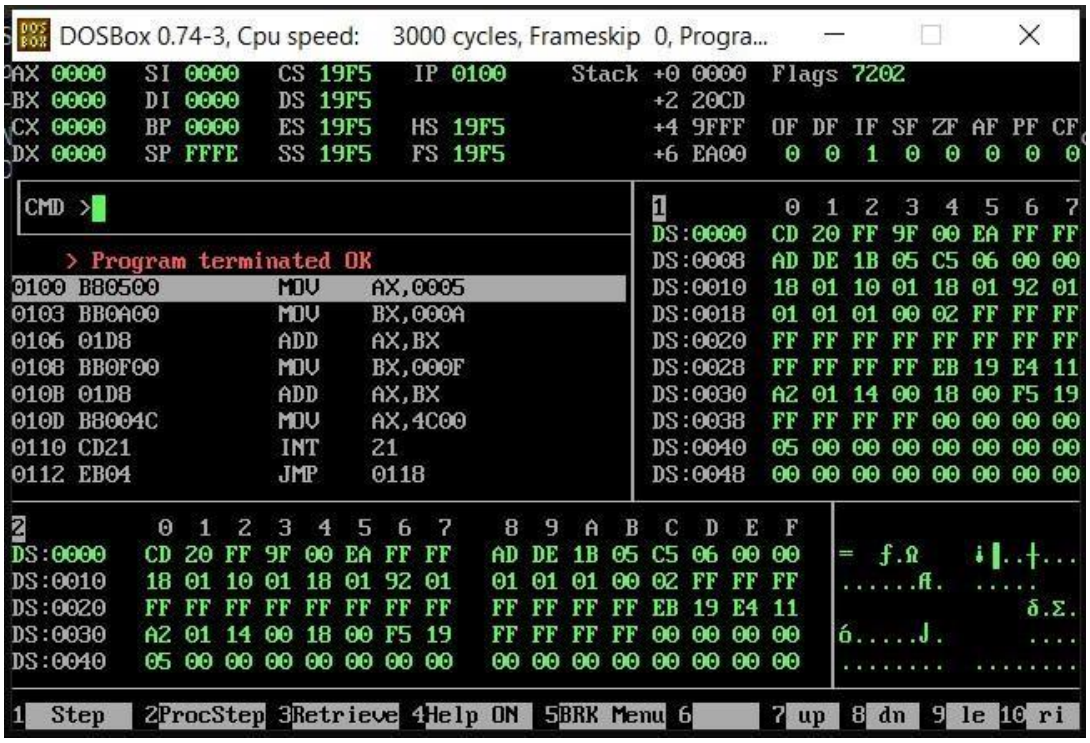
Both ax and bx holds value 15…


DOSBox 0.74-3, Cpu speed: 3000 cycles, Frameskip 0, Progra...

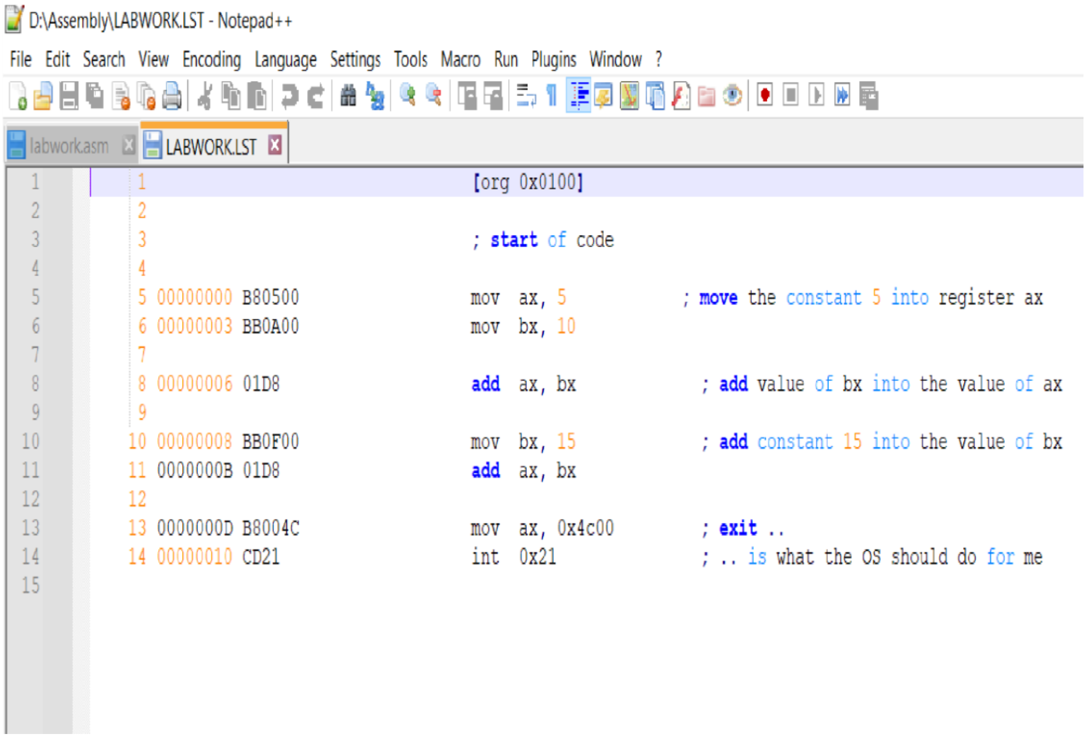
```
AX 001E   SI 0000   CS 19F5   IP 010D      Stack +0 0000   Flags 7214
BX 000F   DI 0000   DS 19F5                      +2 20CD
CX 0012   BP 0000   ES 19F5   HS 19F5            +4 9FFF   OF DF IF SF ZF AF PF CF
DX 0000   SP FFFE   SS 19F5   FS 19F5            +6 EA00    0  0  1  0  0  1  1  0

CMD >                                    1          0  1  2  3  4  5  6  7
                                         DS:0000   CD 20 FF 9F 00 EA F0 FE
010B 01D8         ADD    AX,BX           DS:0008   AD DE 1B 05 C5 06 00 00
010D B8004C       MOV    AX,4C00         DS:0010   18 01 10 01 18 01 92 01
0110 CD21         INT    21              DS:0018   01 01 01 00 02 FF FF FF
0112 EB04         JMP    0118            DS:0020   FF FF FF FF FF FF FF FF
0114 31D2         XOR    DX,DX           DS:0028   FF FF FF FF EB 19 C0 11
0116 31C0         XOR    AX,AX           DS:0030   A2 01 14 00 18 00 F5 19
0118 8956E4       MOV    [BP-1C],DX      DS:0038   FF FF FF FF 00 00 00 00
011B 8946E6       MOV    [BP-1A],AX      DS:0040   05 00 00 00 00 00 00 00
011E C746F60000   MOV    [BP-0A],0000    DS:0048   00 00 00 00 00 00 00 00

2          0  1  2  3  4  5  6  7    8  9  A  B  C  D  E  F
DS:0000   CD 20 FF 9F 00 EA F0 FE   AD DE 1B 05 C5 06 00 00   = ƒ.Ω≡■  ¡▌..┼...
DS:0010   18 01 10 01 18 01 92 01   01 01 01 00 02 FF FF FF   ......Æ.  .....
DS:0020   FF FF FF FF FF FF FF FF   FF FF FF FF EB 19 C0 11             δ.└.
DS:0030   A2 01 14 00 18 00 F5 19   FF FF FF FF 00 00 00 00   ó.....J.      ....
DS:0040   05 00 00 00 00 00 00 00   00 00 00 00 00 00 00 00   ........  ........

1 Step  2ProcStep 3Retrieve 4Help ON  5BRK Menu 6      7 up  8 dn  9 le 10 ri
```


DOSBox 0.74-3, Cpu speed: 3000 cycles, Frameskip 0, Progra...

```
AX 4C00   SI 0000   CS 19F5   IP 0110      Stack +0 0000   Flags 7214
BX 000F   DI 0000   DS 19F5                      +2 20CD
CX 0012   BP 0000   ES 19F5   HS 19F5            +4 9FFF   OF DF IF SF ZF AF PF CF
DX 0000   SP FFFE   SS 19F5   FS 19F5            +6 EA00    0  0  1  0  0  1  1  0

CMD >                                    1          0  1  2  3  4  5  6  7
                                         DS:0000   CD 20 FF 9F 00 EA F0 FE
010D B8004C       MOV    AX,4C00         DS:0008   AD DE 1B 05 C5 06 00 00
0110 CD21         INT    21              DS:0010   18 01 10 01 18 01 92 01
0112 EB04         JMP    0118            DS:0018   01 01 01 00 02 FF FF FF
0114 31D2         XOR    DX,DX           DS:0020   FF FF FF FF FF FF FF FF
0116 31C0         XOR    AX,AX           DS:0028   FF FF FF FF EB 19 C0 11
0118 8956E4       MOV    [BP-1C],DX      DS:0030   A2 01 14 00 18 00 F5 19
011B 8946E6       MOV    [BP-1A],AX      DS:0038   FF FF FF FF 00 00 00 00
011E C746F60000   MOV    [BP-0A],0000    DS:0040   05 00 00 00 00 00 00 00
0123 8B46F6       MOV    AX,[BP-0A]      DS:0048   00 00 00 00 00 00 00 00

2          0  1  2  3  4  5  6  7    8  9  A  B  C  D  E  F
DS:0000   CD 20 FF 9F 00 EA F0 FE   AD DE 1B 05 C5 06 00 00   = ƒ.Ω≡■  ¡▌..┼...
DS:0010   18 01 10 01 18 01 92 01   01 01 01 00 02 FF FF FF   ......Æ.  .....
DS:0020   FF FF FF FF FF FF FF FF   FF FF FF FF EB 19 C0 11             δ.└.
DS:0030   A2 01 14 00 18 00 F5 19   FF FF FF FF 00 00 00 00   ó.....J.      ....
DS:0040   05 00 00 00 00 00 00 00   00 00 00 00 00 00 00 00   ........  ........

1 Step  2ProcStep 3Retrieve 4Help ON  5BRK Menu 6      7 up  8 dn  9 le 10 ri
```



```
DOSBox 0.74-3, Cpu speed:   3000 cycles, Frameskip  0, Progra...
AX 0000   SI 0000   CS 19F5   IP 0100     Stack +0 0000  Flags 7202
BX 0000   DI 0000   DS 19F5                     +2 20CD
CX 0000   BP 0000   ES 19F5   HS 19F5           +4 9FFF  OF DF IF SF ZF AF PF CF
DX 0000   SP FFFE   SS 19F5   FS 19F5           +6 EA00   0  0  1  0  0  0  0  0

 CMD >                                    1         0  1  2  3  4  5  6  7
                                          DS:0000  CD 20 FF 9F 00 EA FF FF
    > Program terminated OK               DS:0008  AD DE 1B 05 C5 06 00 00
0100 B80500         MOV    AX,0005        DS:0010  18 01 10 01 18 01 92 01
0103 BB0A00         MOV    BX,000A        DS:0018  01 01 01 00 02 FF FF FF
0106 01D8           ADD    AX,BX          DS:0020  FF FF FF FF FF FF FF FF
0108 BB0F00         MOV    BX,000F        DS:0028  FF FF FF FF EB 19 E4 11
010B 01D8           ADD    AX,BX          DS:0030  A2 01 14 00 18 00 F5 19
010D B8004C         MOV    AX,4C00        DS:0038  FF FF FF FF 00 00 00 00
0110 CD21           INT    21             DS:0040  05 00 00 00 00 00 00 00
0112 EB04           JMP    0118           DS:0048  00 00 00 00 00 00 00 00

2          0  1  2  3  4  5  6  7    8  9  A  B  C  D  E  F
DS:0000   CD 20 FF 9F 00 EA FF FF   AD DE 1B 05 C5 06 00 00   =  ƒ.Ω     ¡▌..┼...
DS:0010   18 01 10 01 18 01 92 01   01 01 01 00 02 FF FF FF   ......Æ.   .....
DS:0020   FF FF FF FF FF FF FF FF   FF FF FF FF EB 19 E4 11                δ.Σ.
DS:0030   A2 01 14 00 18 00 F5 19   FF FF FF FF 00 00 00 00   ó.....J.       ....
DS:0040   05 00 00 00 00 00 00 00   00 00 00 00 00 00 00 00   ........   ........

1 Step  2ProcStep  3Retrieve  4Help ON  5BRK Menu  6      7 up  8 dn  9 le  10 ri
```



D:\Assembly\LABWORK.LST - Notepad++

File Edit Search View Encoding Language Settings Tools Macro Run Plugins Window ?
labwork.asm
LABWORK.LST

```
     1                                  [org 0x0100]
     2
     3                                  ; start of code
     4
     5 00000000 B80500                  mov  ax, 5           ; move the constant 5 into register ax
     6 00000003 BB0A00                  mov  bx, 10
     7
     8 00000006 01D8                    add  ax, bx            ; add value of bx into the value of ax
     9
    10 00000008 BB0F00                  mov  bx, 15            ; add constant 15 into the value of bx
    11 0000000B 01D8                    add  ax, bx
    12
    13 0000000D B8004C                  mov  ax, 0x4c00        ; exit ..
    14 00000010 CD21                    int  0x21              ; .. is what the OS should do for me
```

.Lst file gives us the detailed information of our code and its memory

Title: Analysing Behaviour of our hardware.

Using this we can analyse how well our code performs and how well it is optimised for our hardware, how we are able to make it more efficient and faster, less time consuming as we know one of the benefits of assembly language is, its time efficiency