Name: Amber Khurshid

Class: BAI-4A

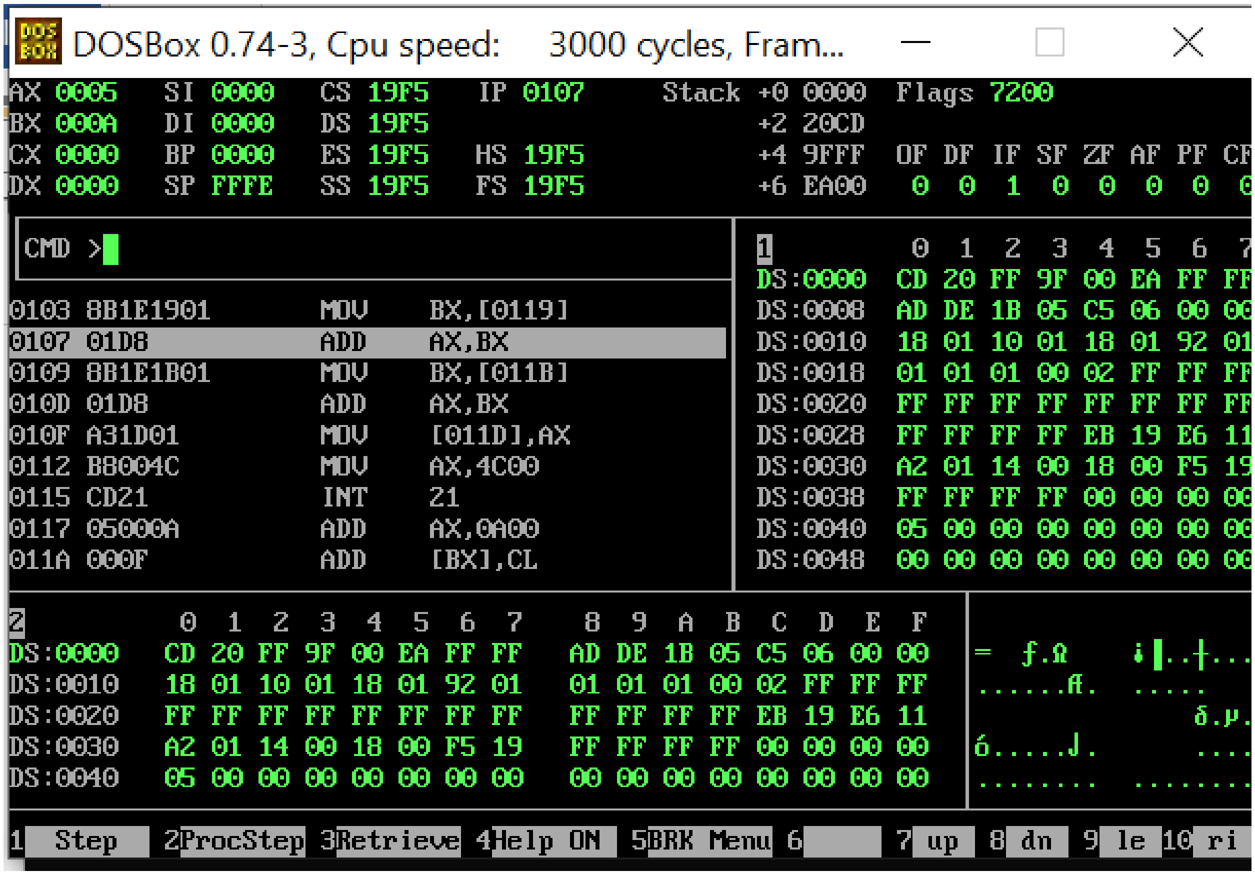
Roll No: 22P-9295

Lab # 05

1: Parity Flag

```
; a program to add three numbers accessed using a single label
[org 0x0100]


   mov  ax, [num1]


   mov  bx, [num1 + 2]     ; notice how we can do arithmetic here
   sub  ax, bx
   add  ax, bx            ; also, why +2 and not +1?
   mov  bx, [num1 + 4]
   add  ax, bx
   mov  [num1 + 6], ax     ; store sum at num1+6
   mov  ax, 0x4c00
   int  0x21


num1:   dw  5
      dw  10
      dw  15
      dw  0
```

Before:

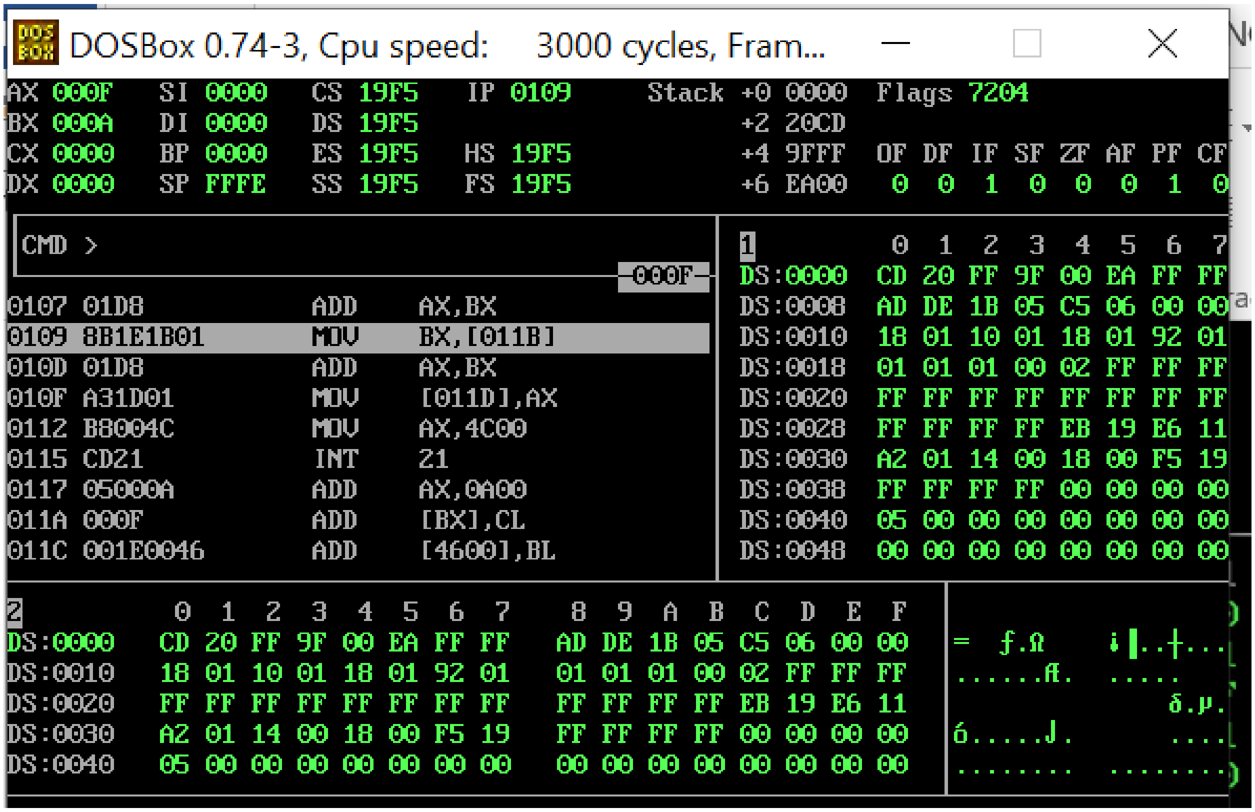


```
AX 0005    SI 0000    CS 19F5    IP 0107       Stack +0 0000  Flags 7200
BX 000A    DI 0000    DS 19F5                        +2 20CD
CX 0000    BP 0000    ES 19F5    HS 19F5             +4 9FFF  OF DF IF SF ZF AF PF CF
DX 0000    SP FFFE    SS 19F5    FS 19F5             +6 EA00   0  0  1  0  0  0  0  0

 CMD >                                                1        0  1  2  3  4  5  6  7
                                                      DS:0000  CD 20 FF 9F 00 EA FF FF
0103 8B1E1901          MOV     BX,[0119]              DS:0008  AD DE 1B 05 C5 06 00 00
0107 01D8              ADD     AX,BX                  DS:0010  18 01 10 01 18 01 92 01
0109 8B1E1B01          MOV     BX,[011B]              DS:0018  01 01 01 00 02 FF FF FF
010D 01D8              ADD     AX,BX                  DS:0020  FF FF FF FF FF FF FF FF
010F A31D01            MOV     [011D],AX              DS:0028  FF FF FF FF EB 19 E6 11
0112 B8004C            MOV     AX,4C00                DS:0030  A2 01 14 00 18 00 F5 19
0115 CD21              INT     21                     DS:0038  FF FF FF FF 00 00 00 00
0117 05000A            ADD     AX,0A00                DS:0040  05 00 00 00 00 00 00 00
011A 000F              ADD     [BX],CL                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 E6 11                δ.µ.
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
```

After:

The parity flag is set when addition of two numbers produces and even number of 1 bit.

For this code addition of 5(0101) and 10(1010) produces 1111 in binary so the parity flag is set to 1 .



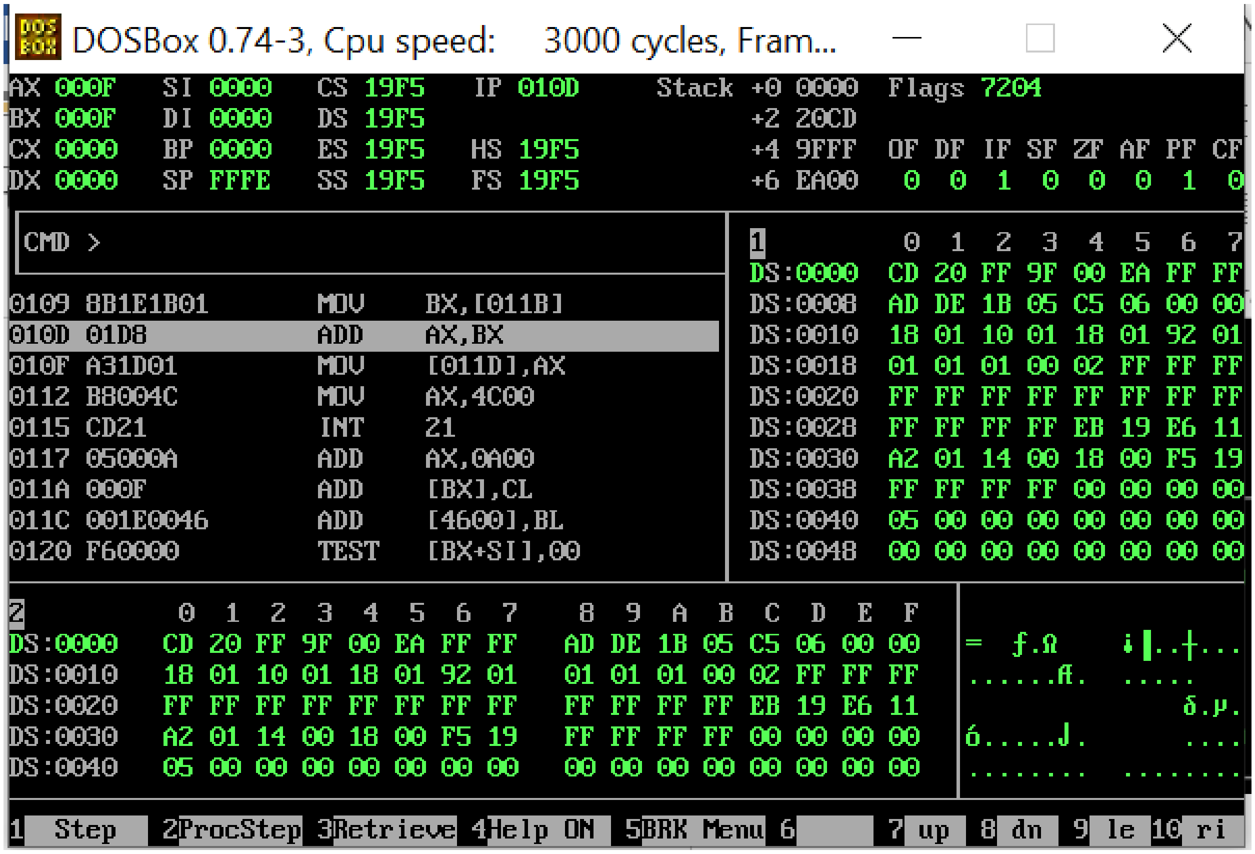
```
AX 000F   SI 0000   CS 19F5   IP 0109      Stack +0 0000  Flags 7204
BX 000A   DI 0000   DS 19F5                      +2 20CD
CX 0000   BP 0000   ES 19F5   HS 19F5            +4 9FFF  OF DF IF SF ZF AF PF CF
DX 0000   SP FFFE   SS 19F5   FS 19F5            +6 EA00   0  0  1  0  0  0  1  0

 CMD >                                          1         0  1  2  3  4  5  6  7
                                        -000F-  DS:0000   CD 20 FF 9F 00 EA FF FF
0107 01D8              ADD    AX,BX             DS:0008   AD DE 1B 05 C5 06 00 00
0109 8B1E1B01          MOV    BX,[011B]         DS:0010   18 01 10 01 18 01 92 01
010D 01D8              ADD    AX,BX             DS:0018   01 01 01 00 02 FF FF FF
010F A31D01            MOV    [011D],AX         DS:0020   FF FF FF FF FF FF FF FF
0112 B8004C            MOV    AX,4C00           DS:0028   FF FF FF FF EB 19 E6 11
0115 CD21              INT    21                DS:0030   A2 01 14 00 18 00 F5 19
0117 05000A            ADD    AX,0A00           DS:0038   FF FF FF FF 00 00 00 00
011A 000F              ADD    [BX],CL           DS:0040   05 00 00 00 00 00 00 00
011C 001E0046          ADD    [4600],BL         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 E6 11                δ.µ.
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   ........   ........
```

2: Auxiliary Carry

```
; a program to add three numbers accessed using a single label

[org 0x0100]


    mov  ax, [num1]


    mov  bx, [num1 + 2]     ; notice how we can do arithmetic here

    sub  ax, bx

    add  ax, bx             ; also, why +2 and not +1?

    mov  bx, [num1 + 4]

    add  ax, bx

    mov  [num1 + 6], ax     ; store sum at num1+6

    mov  ax, 0x4c00
```

```
   int  0x21


num1:   dw  5
        dw  10
        dw  15
        dw  0
```

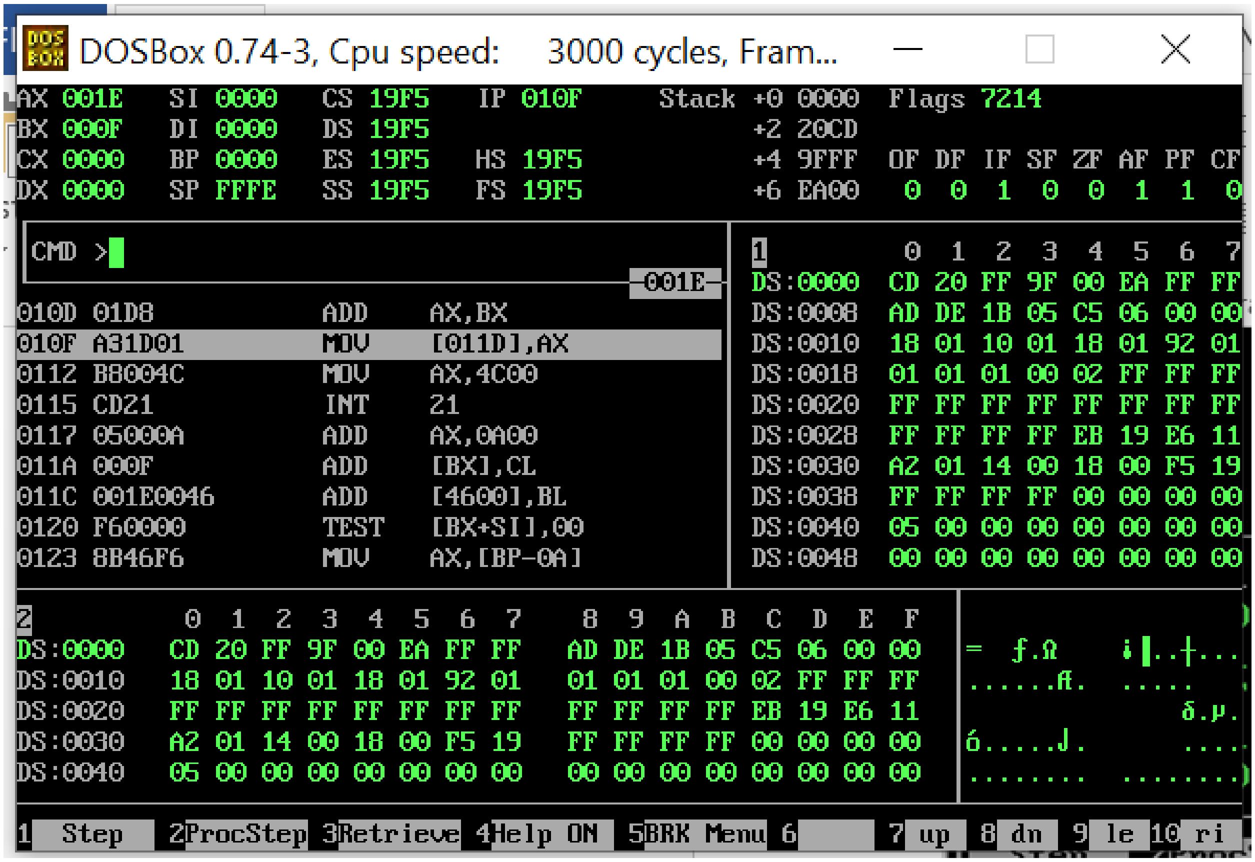
Before:



```
AX 000F    SI 0000    CS 19F5    IP 010D       Stack +0 0000   Flags 7204
BX 000F    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  1  0

 CMD >                                              1           0  1  2  3  4  5  6  7
                                                    DS:0000  CD 20 FF 9F 00 EA FF FF
0109 8B1E1B01         MOV     BX,[011B]             DS:0008  AD DE 1B 05 C5 06 00 00
010D 01D8             ADD     AX,BX                 DS:0010  18 01 10 01 18 01 92 01
010F A31D01           MOV     [011D],AX             DS:0018  01 01 01 00 02 FF FF FF
0112 B8004C           MOV     AX,4C00               DS:0020  FF FF FF FF FF FF FF FF
0115 CD21             INT     21                    DS:0028  FF FF FF FF EB 19 E6 11
0117 05000A           ADD     AX,0A00               DS:0030  A2 01 14 00 18 00 F5 19
011A 000F             ADD     [BX],CL               DS:0038  FF FF FF FF 00 00 00 00
011C 001E0046         ADD     [4600],BL             DS:0040  05 00 00 00 00 00 00 00
0120 F60000           TEST    [BX+SI],00            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 E6 11             δ.µ.
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
```

After:

Auxiliary carry is set when there is a carry from low nibble to high nibble. When we add 15(1111) + 15(1111) ,the result is 1 1110. We can see there is a carry from low nibble to high nibble so the auxiliary flag is raised.



```
AX 001E   SI 0000   CS 19F5   IP 010F     Stack +0 0000  Flags 7214
BX 000F   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  1  1  0

 CMD >                                           1        0  1  2  3  4  5  6  7
                                         -001E-  DS:0000  CD 20 FF 9F 00 EA FF FF
010D 01D8          ADD     AX,BX                 DS:0008  AD DE 1B 05 C5 06 00 00
010F A31D01        MOV     [011D],AX             DS:0010  18 01 10 01 18 01 92 01
0112 B8004C        MOV     AX,4C00               DS:0018  01 01 01 00 02 FF FF FF
0115 CD21          INT     21                    DS:0020  FF FF FF FF FF FF FF FF
0117 05000A        ADD     AX,0A00               DS:0028  FF FF FF FF EB 19 E6 11
011A 000F          ADD     [BX],CL               DS:0030  A2 01 14 00 18 00 F5 19
011C 001E0046      ADD     [4600],BL             DS:0038  FF FF FF FF 00 00 00 00
0120 F60000        TEST    [BX+SI],00            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 FF FF  AD DE 1B 05 C5 06 00 00   =  ƒ.Ω    i▌..┼...
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 E6 11             δ.µ.
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
```

3: Interrupt Flag

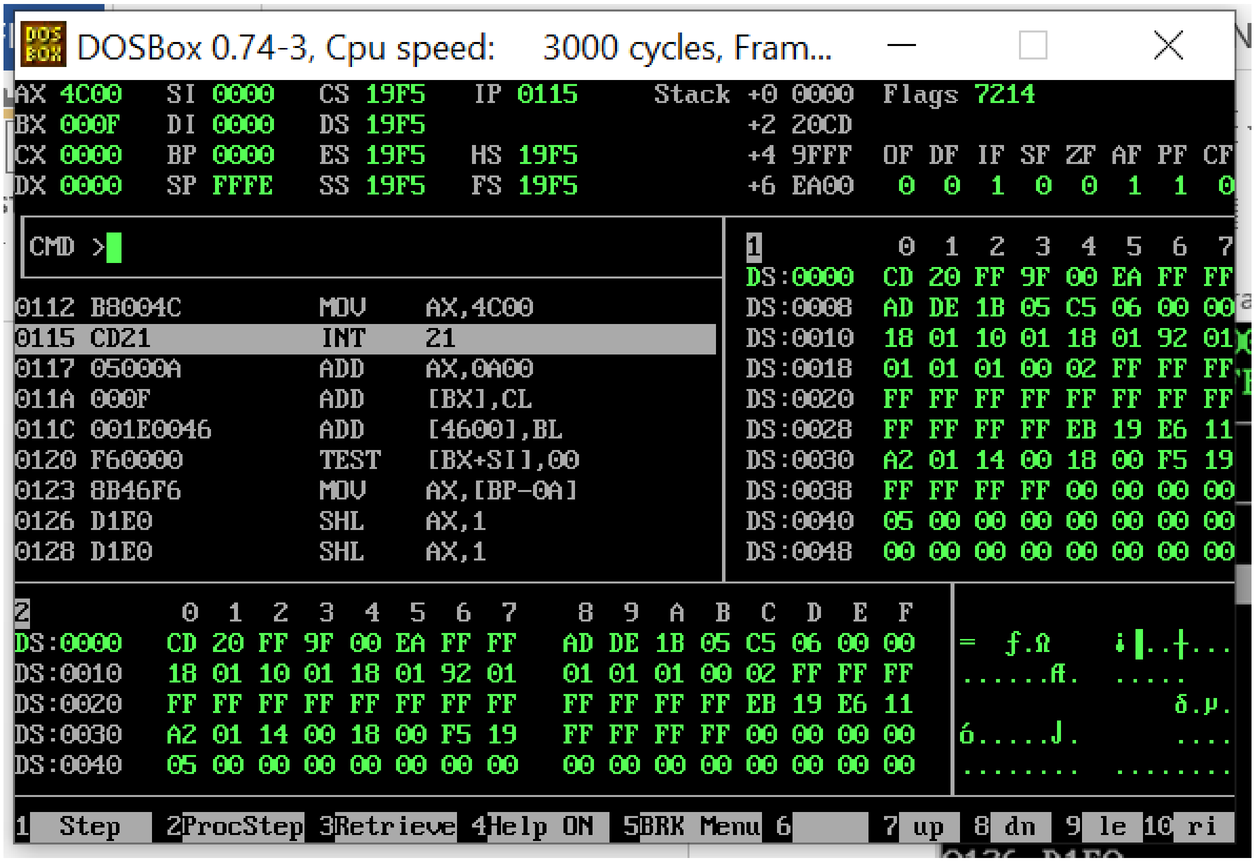
; a program to add three numbers accessed using a single label

[org 0x0100]

```
   mov  ax, [num1]

   mov  bx, [num1 + 2]     ; notice how we can do arithmetic here
   sub  ax, bx
   add  ax, bx             ; also, why +2 and not +1?
   mov  bx, [num1 + 4]
   add  ax, bx
   mov  [num1 + 6], ax     ; store sum at num1+6
   mov  ax, 0x4c00
   int  0x21


num1:   dw  5
        dw  10
        dw  15
        dw  0
```

Before:

Interrupt flag is initially set to 1.

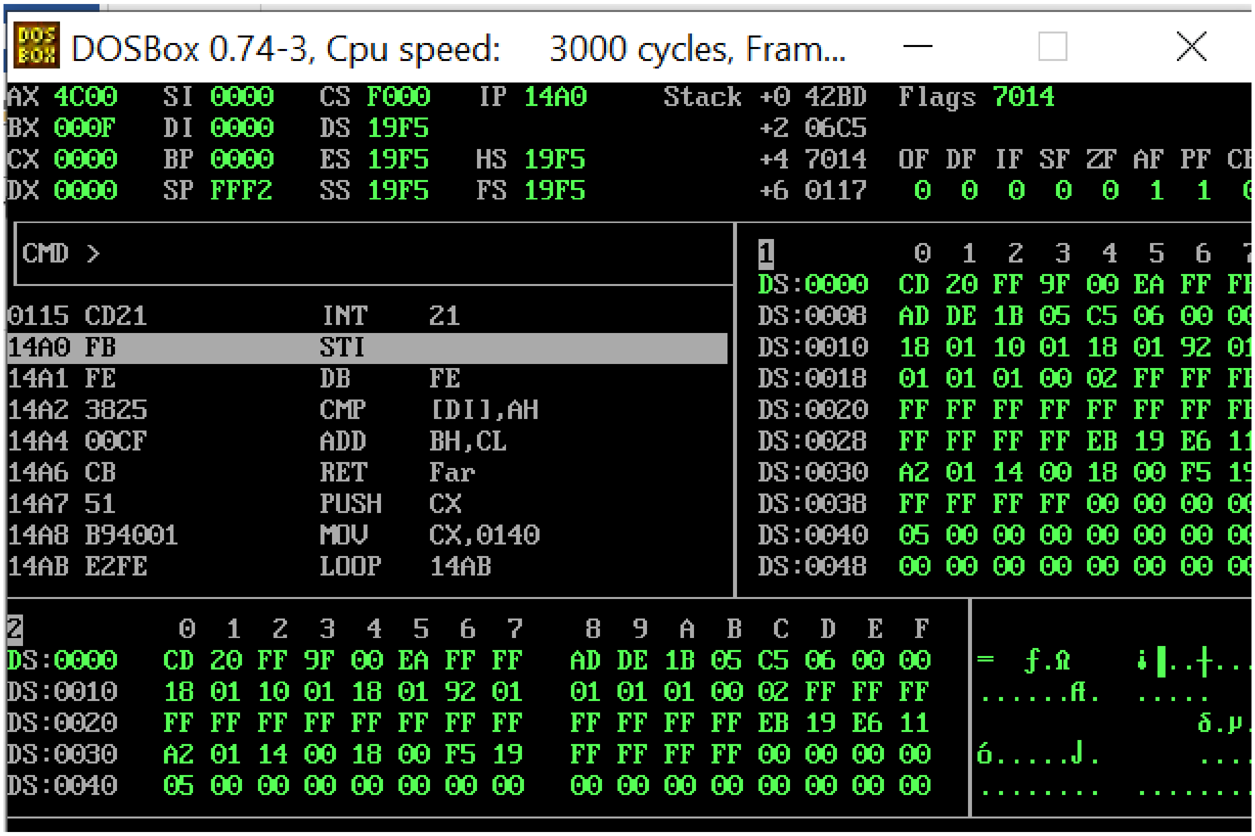


DOSBox 0.74-3, Cpu speed: 3000 cycles, Fram...

```
AX 4C00   SI 0000   CS 19F5   IP 0115       Stack +0 0000  Flags 7214
BX 000F   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  1  1  0

CMD >                                          1        0  1  2  3  4  5  6  7
                                               DS:0000  CD 20 FF 9F 00 EA FF FF
0112 B8004C          MOV    AX,4C00            DS:0008  AD DE 1B 05 C5 06 00 00
0115 CD21            INT    21                 DS:0010  18 01 10 01 18 01 92 01
0117 05000A          ADD    AX,0A00            DS:0018  01 01 01 00 02 FF FF FF
011A 000F            ADD    [BX],CL            DS:0020  FF FF FF FF FF FF FF FF
011C 001E0046        ADD    [4600],BL          DS:0028  FF FF FF FF EB 19 E6 11
0120 F60000          TEST   [BX+SI],00         DS:0030  A2 01 14 00 18 00 F5 19
0123 8B46F6          MOV    AX,[BP-0A]         DS:0038  FF FF FF FF 00 00 00 00
0126 D1E0            SHL    AX,1               DS:0040  05 00 00 00 00 00 00 00
0128 D1E0            SHL    AX,1               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 E6 11               δ.µ.
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
```

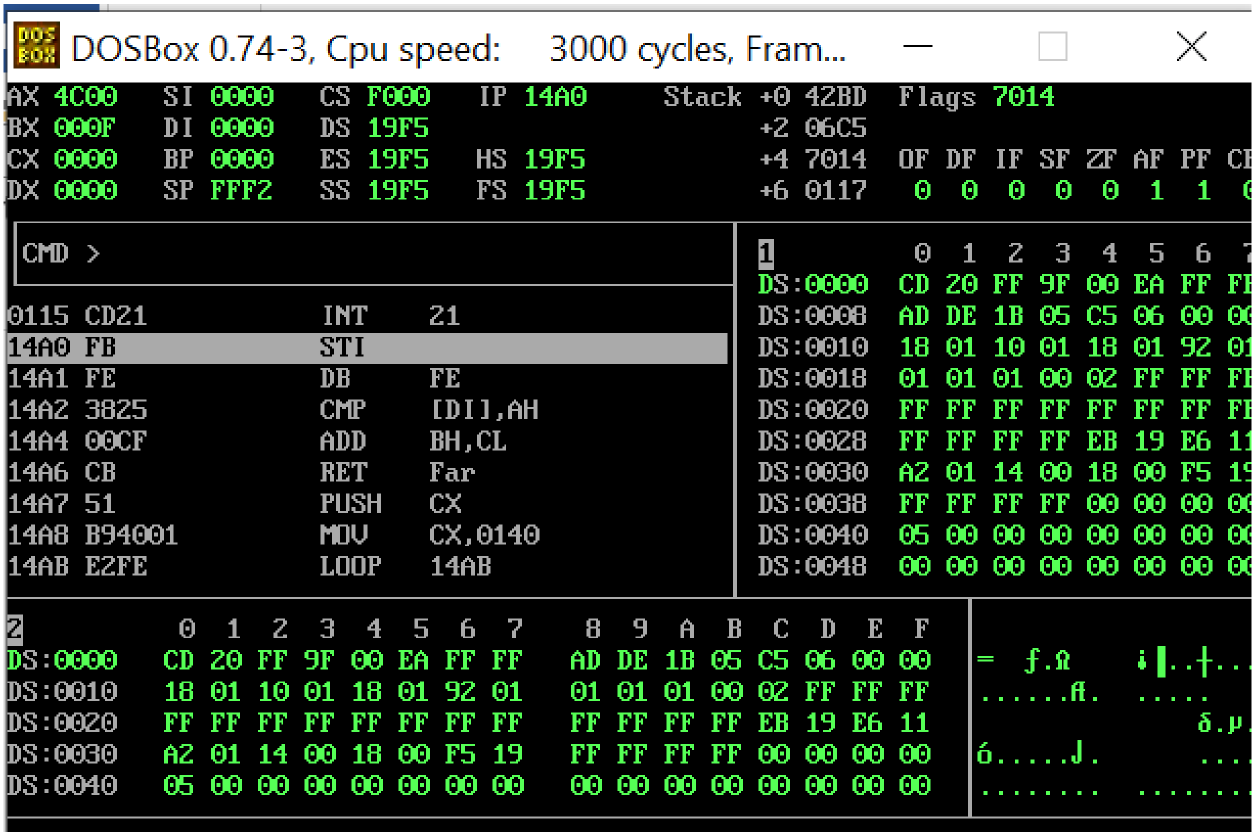
After:

When interrupt is about to be exected the interrupt flag is set to zero.



```
AX 4C00   SI 0000   CS F000   IP 14A0      Stack +0 42BD  Flags 7014
BX 000F   DI 0000   DS 19F5                      +2 06C5
CX 0000   BP 0000   ES 19F5   HS 19F5            +4 7014  OF DF IF SF ZF AF PF C
DX 0000   SP FFF2   SS 19F5   FS 19F5            +6 0117   0  0  0  0  0  1  1

 CMD >                                       1        0  1  2  3  4  5  6
                                             DS:0000  CD 20 FF 9F 00 EA FF F
0115 CD21           INT    21                DS:0008  AD DE 1B 05 C5 06 00 0
14A0 FB             STI                      DS:0010  18 01 10 01 18 01 92 0
14A1 FE             DB     FE                DS:0018  01 01 01 00 02 FF FF F
14A2 3825           CMP    [DI],AH           DS:0020  FF FF FF FF FF FF FF F
14A4 00CF           ADD    BH,CL             DS:0028  FF FF FF FF EB 19 E6 1
14A6 CB             RET    Far               DS:0030  A2 01 14 00 18 00 F5 1
14A7 51             PUSH   CX                DS:0038  FF FF FF FF 00 00 00 0
14A8 B94001         MOV    CX,0140           DS:0040  05 00 00 00 00 00 00 0
14AB E2FE           LOOP   14AB              DS:0048  00 00 00 00 00 00 00 0

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 E6 11             δ.µ.
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   ........  ........
```

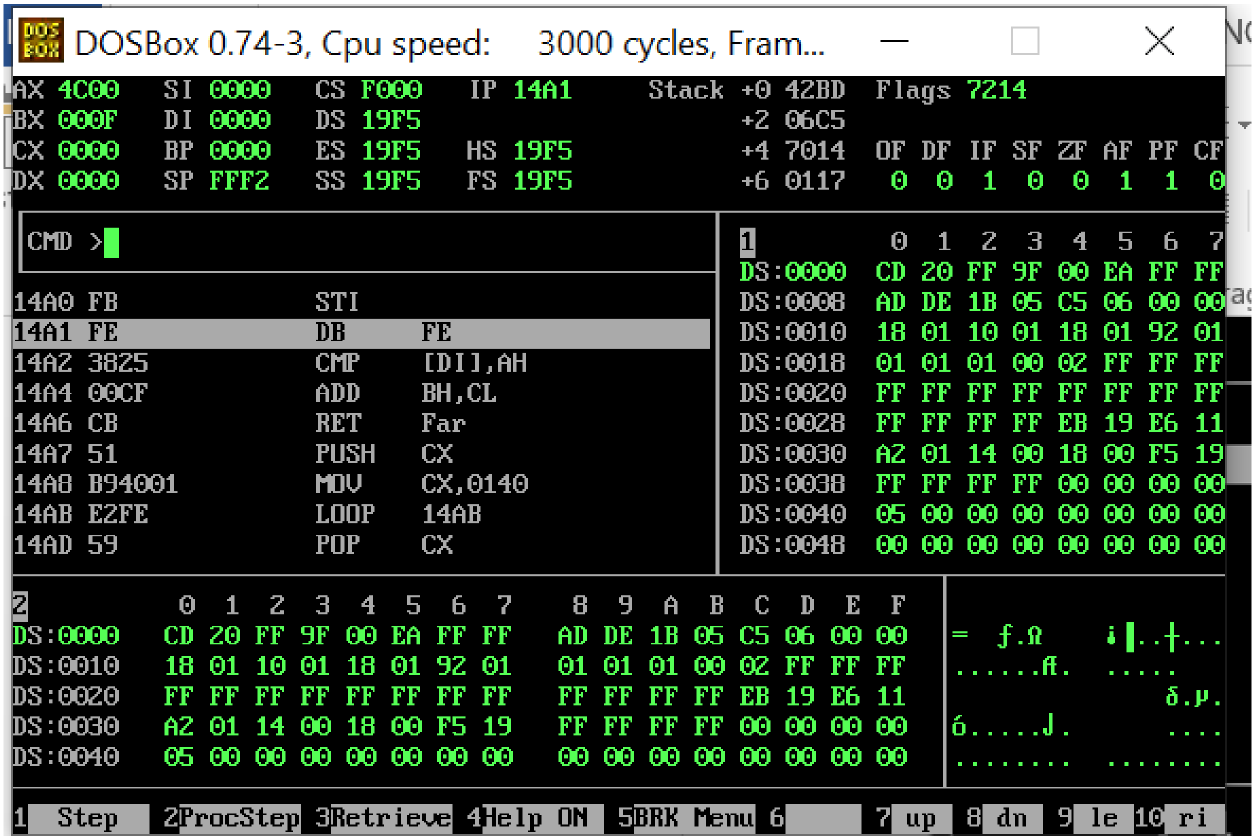
Before:



```
AX 4C00   SI 0000   CS F000   IP 14A0      Stack +0 42BD  Flags 7014
BX 000F   DI 0000   DS 19F5                      +2 06C5
CX 0000   BP 0000   ES 19F5   HS 19F5            +4 7014  OF DF IF SF ZF AF PF C
DX 0000   SP FFF2   SS 19F5   FS 19F5            +6 0117   0  0  0  0  0  1  1  0

 CMD >                                         1          0  1  2  3  4  5  6  
                                               DS:0000  CD 20 FF 9F 00 EA FF F
0115 CD21              INT    21               DS:0008  AD DE 1B 05 C5 06 00 0
14A0 FB                STI                     DS:0010  18 01 10 01 18 01 92 0
14A1 FE                DB     FE               DS:0018  01 01 01 00 02 FF FF F
14A2 3825              CMP    [DI],AH          DS:0020  FF FF FF FF FF FF FF F
14A4 00CF              ADD    BH,CL            DS:0028  FF FF FF FF EB 19 E6 1
14A6 CB                RET    Far              DS:0030  A2 01 14 00 18 00 F5 1
14A7 51                PUSH   CX               DS:0038  FF FF FF FF 00 00 00 0
14A8 B94001            MOV    CX,0140          DS:0040  05 00 00 00 00 00 00 0
14AB E2FE              LOOP   14AB             DS:0048  00 00 00 00 00 00 00 0

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 E6 11             δ.μ.
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   ........  ........
```

After:

When interrupt command is executed, the interrupt flag is again set.



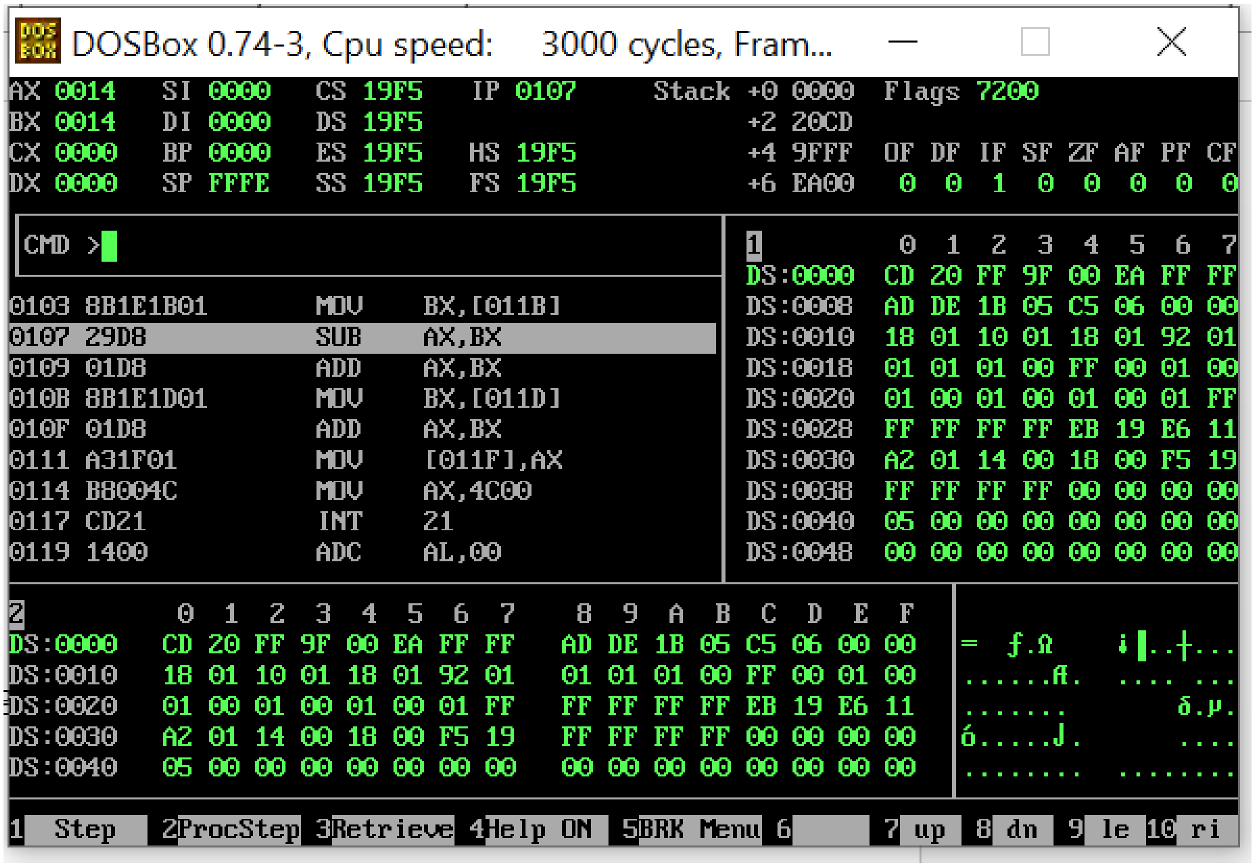
```
AX 4C00   SI 0000   CS F000   IP 14A1      Stack +0 42BD  Flags 7214
BX 000F   DI 0000   DS 19F5                      +2 06C5
CX 0000   BP 0000   ES 19F5   HS 19F5            +4 7014  OF DF IF SF ZF AF PF CF
DX 0000   SP FFF2   SS 19F5   FS 19F5            +6 0117   0  0  1  0  0  1  1  0

 CMD >                                           1          0  1  2  3  4  5  6  7
                                                 DS:0000  CD 20 FF 9F 00 EA FF FF
14A0 FB                 STI                      DS:0008  AD DE 1B 05 C5 06 00 00
14A1 FE                 DB      FE               DS:0010  18 01 10 01 18 01 92 01
14A2 3825               CMP     [DI],AH          DS:0018  01 01 01 00 02 FF FF FF
14A4 00CF               ADD     BH,CL            DS:0020  FF FF FF FF FF FF FF FF
14A6 CB                 RET     Far              DS:0028  FF FF FF FF EB 19 E6 11
14A7 51                 PUSH    CX               DS:0030  A2 01 14 00 18 00 F5 19
14A8 B94001             MOV     CX,0140          DS:0038  FF FF FF FF 00 00 00 00
14AB E2FE               LOOP    14AB             DS:0040  05 00 00 00 00 00 00 00
14AD 59                 POP     CX               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 E6 11             δ.µ.
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
```

4: Zero Flag:

```
; a program to add three numbers accessed using a single label
[org 0x0100]

   mov  ax, [num1]

   mov  bx, [num1 + 2]     ; notice how we can do arithmetic here
   sub  ax, bx
   add  ax, bx            ; also, why +2 and not +1?
   mov  bx, [num1 + 4]

   add  ax, bx

   mov  [num1 + 6], ax     ; store sum at num1+6

   mov  ax, 0x4c00

   int  0x21


num1:   dw  20

        dw  20

        dw  15

        dw  0
```

Zero flag is set when the addition or subtraction of two numbers gives us a zero.

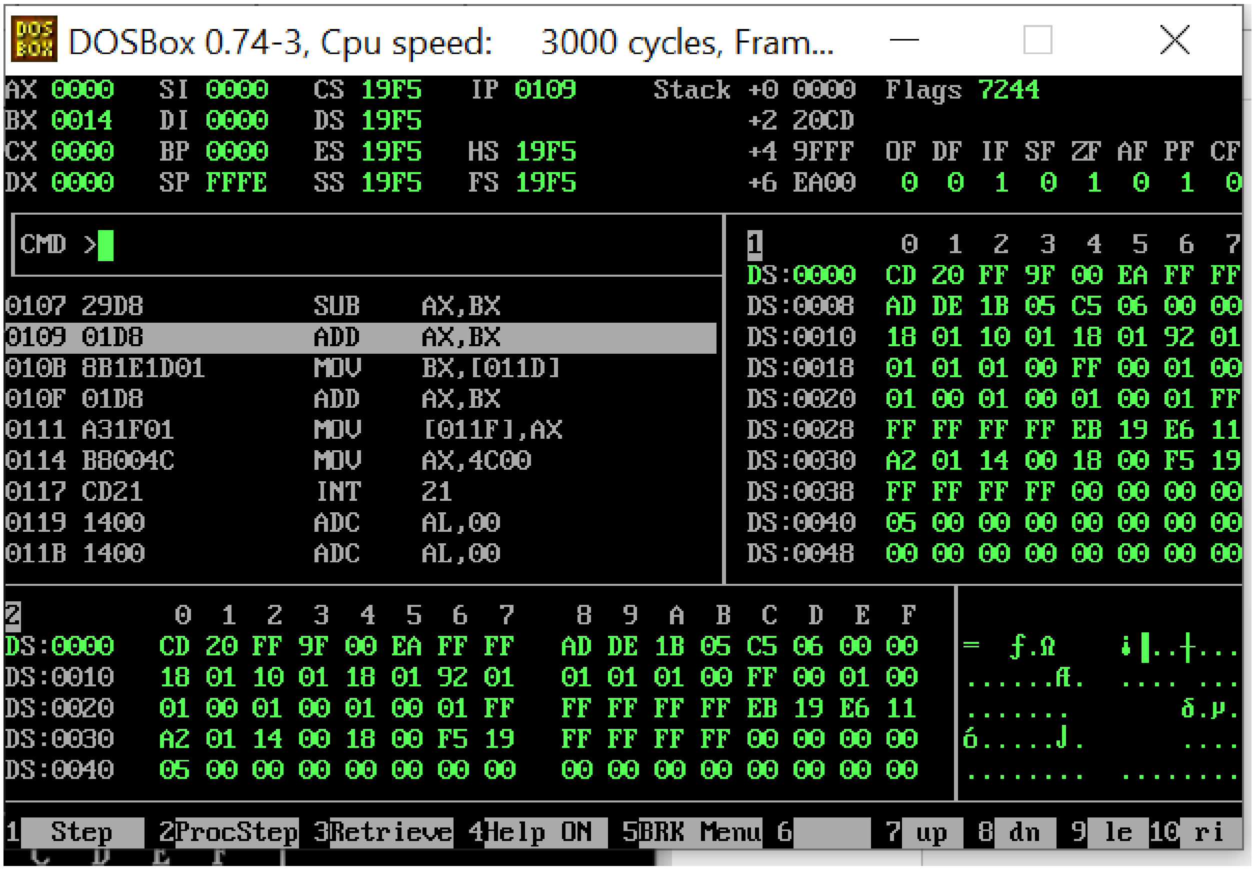
Before:



```
AX 0014    SI 0000    CS 19F5    IP 0107      Stack +0 0000  Flags 7200
BX 0014    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
0103 8B1E1B01        MOV     BX,[011B]             DS:0008   AD DE 1B 05 C5 06 00 00
0107 29D8            SUB     AX,BX                 DS:0010   18 01 10 01 18 01 92 01
0109 01D8            ADD     AX,BX                 DS:0018   01 01 01 00 FF 00 01 00
010B 8B1E1D01        MOV     BX,[011D]             DS:0020   01 00 01 00 01 00 01 FF
010F 01D8            ADD     AX,BX                 DS:0028   FF FF FF FF EB 19 E6 11
0111 A31F01          MOV     [011F],AX             DS:0030   A2 01 14 00 18 00 F5 19
0114 B8004C          MOV     AX,4C00               DS:0038   FF FF FF FF 00 00 00 00
0117 CD21            INT     21                    DS:0040   05 00 00 00 00 00 00 00
0119 1400            ADC     AL,00                 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 FF 00 01 00   ......Æ.  .... ...
DS:0020      01 00 01 00 01 00 01 FF   FF FF FF FF EB 19 E6 11   .......       δ.µ.
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
```

After:

Here when we subtracted 20 from 20 the zero flag was set as the result is 0.



DOSBox 0.74-3, Cpu speed: 3000 cycles, Fram...

```
AX 0000   SI 0000   CS 19F5   IP 0109      Stack +0 0000  Flags 7244
BX 0014   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  1  0  1  0

 CMD >                                          1        0  1  2  3  4  5  6  7
                                                DS:0000  CD 20 FF 9F 00 EA FF FF
0107 29D8           SUB    AX,BX                DS:0008  AD DE 1B 05 C5 06 00 00
0109 01D8           ADD    AX,BX                DS:0010  18 01 10 01 18 01 92 01
010B 8B1E1D01       MOV    BX,[011D]            DS:0018  01 01 01 00 FF 00 01 00
010F 01D8           ADD    AX,BX                DS:0020  01 00 01 00 01 00 01 FF
0111 A31F01         MOV    [011F],AX            DS:0028  FF FF FF FF EB 19 E6 11
0114 B8004C         MOV    AX,4C00              DS:0030  A2 01 14 00 18 00 F5 19
0117 CD21           INT    21                   DS:0038  FF FF FF FF 00 00 00 00
0119 1400           ADC    AL,00                DS:0040  05 00 00 00 00 00 00 00
011B 1400           ADC    AL,00                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 FF 00 01 00   ......Æ.  .... ...
DS:0020   01 00 01 00 01 00 01 FF  FF FF FF FF EB 19 E6 11   .......       δ.µ.
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
```

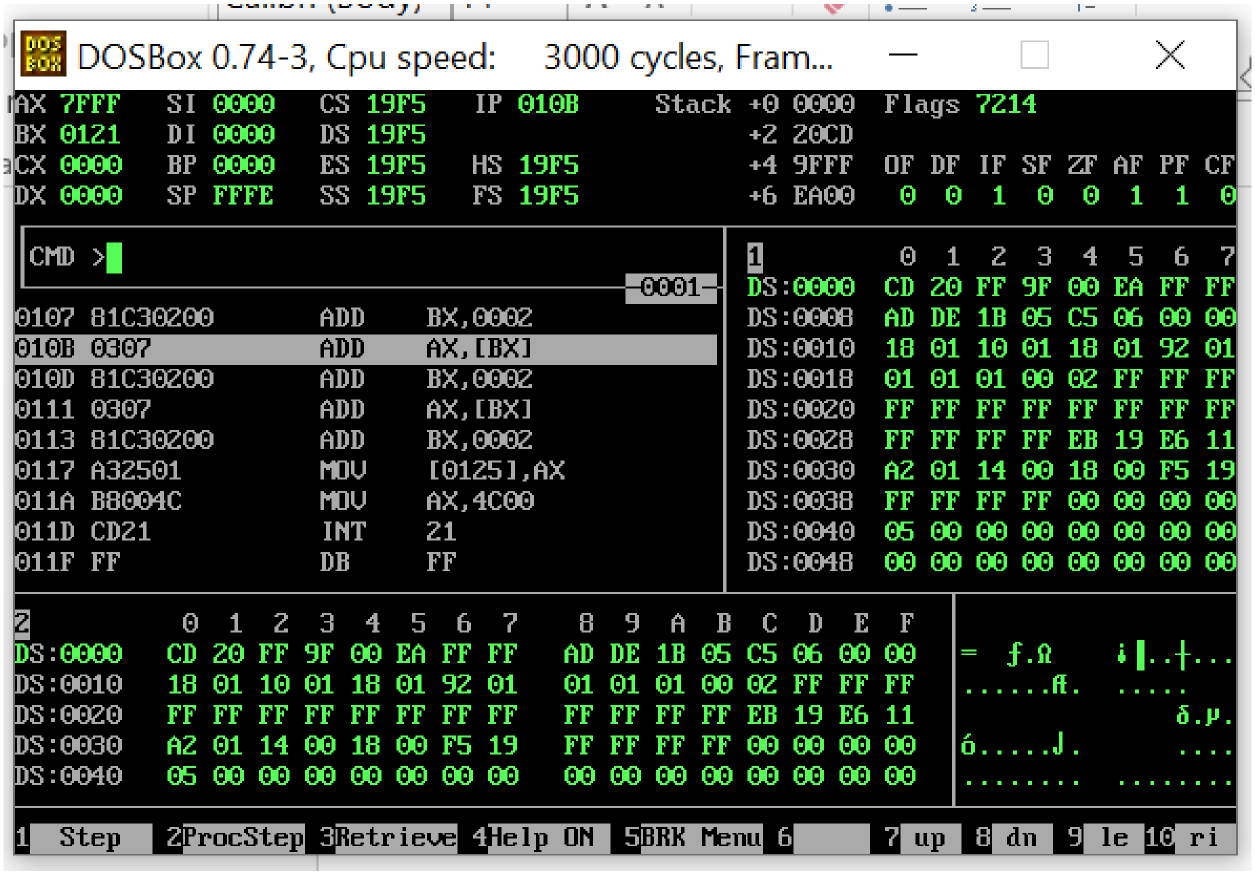
5: Overflow Flag:

An overflow flag is set when the result of addition or subtraction exceeds the maximum representable value for a signed 16-bit integer.

```
; a program to add three numbers using byte variables

[org 0x0100]

    xor  ax, ax    ; check effect on ZF


    mov  bx, num1
```

```
   add  ax, [bx]
   add  bx, 2

   add  ax, [bx]
   add  bx, 2

   add  ax, [bx]
   add  bx, 2


   mov  [result], ax

   mov  ax, 0x4c00
   int  0x21


   ; to turn this into an iteration, we need a couple of things:
   ; - branching instruction
   ; - checking constraints -- e.g. c > 0      ; Intel Sotware Developer Manual - Figure 3-8 (Page 80)


num1: dw  32767, 1, 15
result: dw 0
```

Before:

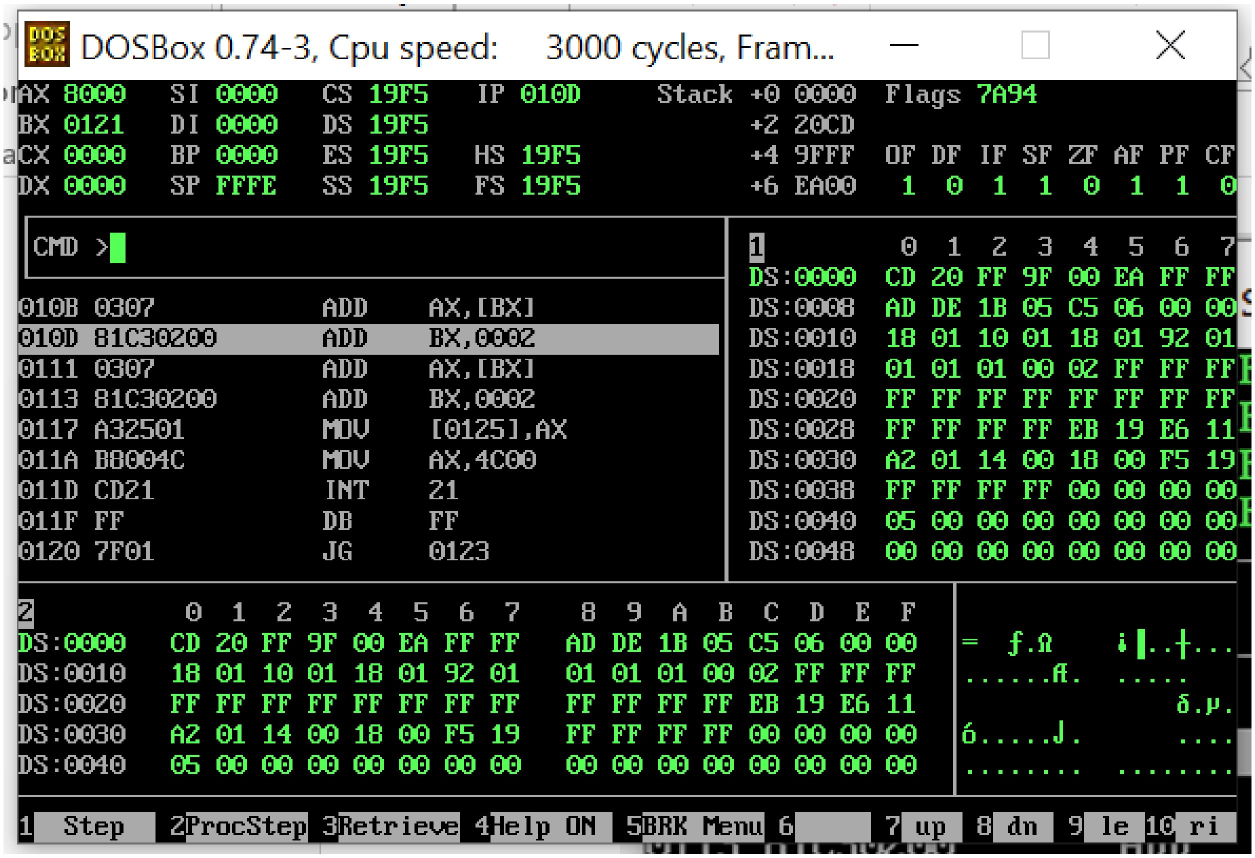


DOSBox 0.74-3, Cpu speed: 3000 cycles, Fram...

```
AX 7FFF    SI 0000    CS 19F5    IP 010B      Stack +0 0000   Flags 7214
BX 0121    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  1  1  0

CMD >                                          1          0  1  2  3  4  5  6  7
                                      -0001-   DS:0000   CD 20 FF 9F 00 EA FF FF
0107 81C30200      ADD     BX,0002             DS:0008   AD DE 1B 05 C5 06 00 00
010B 0307          ADD     AX,[BX]             DS:0010   18 01 10 01 18 01 92 01
010D 81C30200      ADD     BX,0002             DS:0018   01 01 01 00 02 FF FF FF
0111 0307          ADD     AX,[BX]             DS:0020   FF FF FF FF FF FF FF FF
0113 81C30200      ADD     BX,0002             DS:0028   FF FF FF FF EB 19 E6 11
0117 A32501        MOV     [0125],AX           DS:0030   A2 01 14 00 18 00 F5 19
011A B8004C        MOV     AX,4C00             DS:0038   FF FF FF FF 00 00 00 00
011D CD21          INT     21                  DS:0040   05 00 00 00 00 00 00 00
011F FF            DB      FF                  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 E6 11               δ.µ.
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
```

After:

The AX register can store a maximum positive value of 32767(0111111111111111). If 1 is added to this value the binary value becomes (1000000000000000) , it causes an overflow, leading the highest bit to change and the overflow flag to be set, resulting in a transition to a negative value.



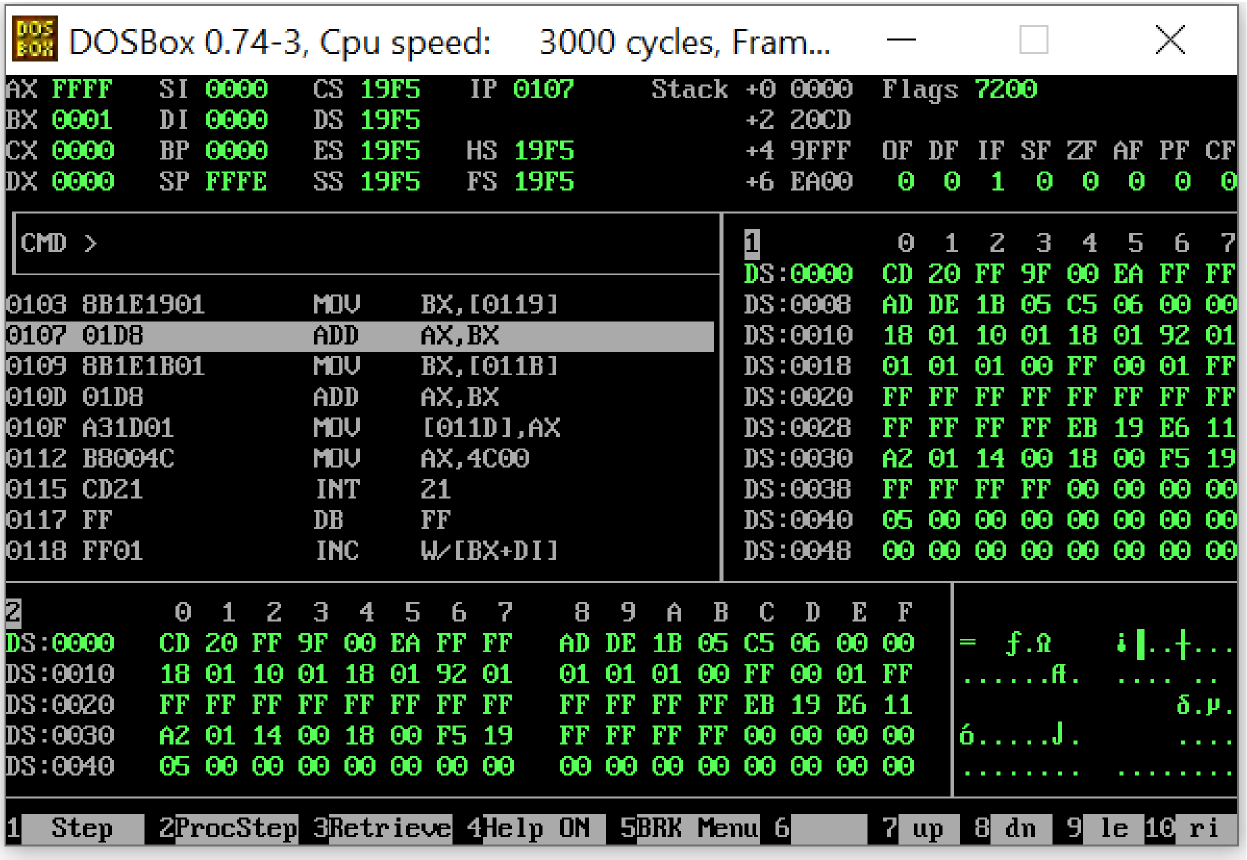
```
AX 8000    SI 0000    CS 19F5    IP 010D      Stack +0 0000  Flags 7A94
BX 0121    DI 0000    DS 19F5                       +2 20CD
CX 0000    BP 0000    ES 19F5    HS 19F5           +4 9FFF  OF DF IF SF ZF AF PF CF
DX 0000    SP FFFE    SS 19F5    FS 19F5           +6 EA00   1  0  1  1  0  1  1  0

 CMD >                                            1          0  1  2  3  4  5  6  7
                                                  DS:0000  CD 20 FF 9F 00 EA FF FF
010B 0307              ADD     AX,[BX]            DS:0008  AD DE 1B 05 C5 06 00 00
010D 81C30200          ADD     BX,0002            DS:0010  18 01 10 01 18 01 92 01
0111 0307              ADD     AX,[BX]            DS:0018  01 01 01 00 02 FF FF FF
0113 81C30200          ADD     BX,0002            DS:0020  FF FF FF FF FF FF FF FF
0117 A32501            MOV     [0125],AX          DS:0028  FF FF FF FF EB 19 E6 11
011A B8004C            MOV     AX,4C00            DS:0030  A2 01 14 00 18 00 F5 19
011D CD21              INT     21                 DS:0038  FF FF FF FF 00 00 00 00
011F FF                DB      FF                 DS:0040  05 00 00 00 00 00 00 00
0120 7F01              JG      0123               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 E6 11              δ.µ.
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

6: Carry Flag:

```
; a program to add three numbers using memory variables
[org 0x0100]

    mov  ax, [num1]        ; load first number in ax
    ; mov  [num1], [num2]     ; illegal
    mov  bx, [num2]
    add  ax, bx
    mov  bx, [num3]
    add  ax, bx

    mov  [num4], ax

    mov  ax, 0x4c00

    int  0x21



num1: dw   65535

num2: dw   1

num3: dw   15

num4: dw   0



; watch the listing carefully
```

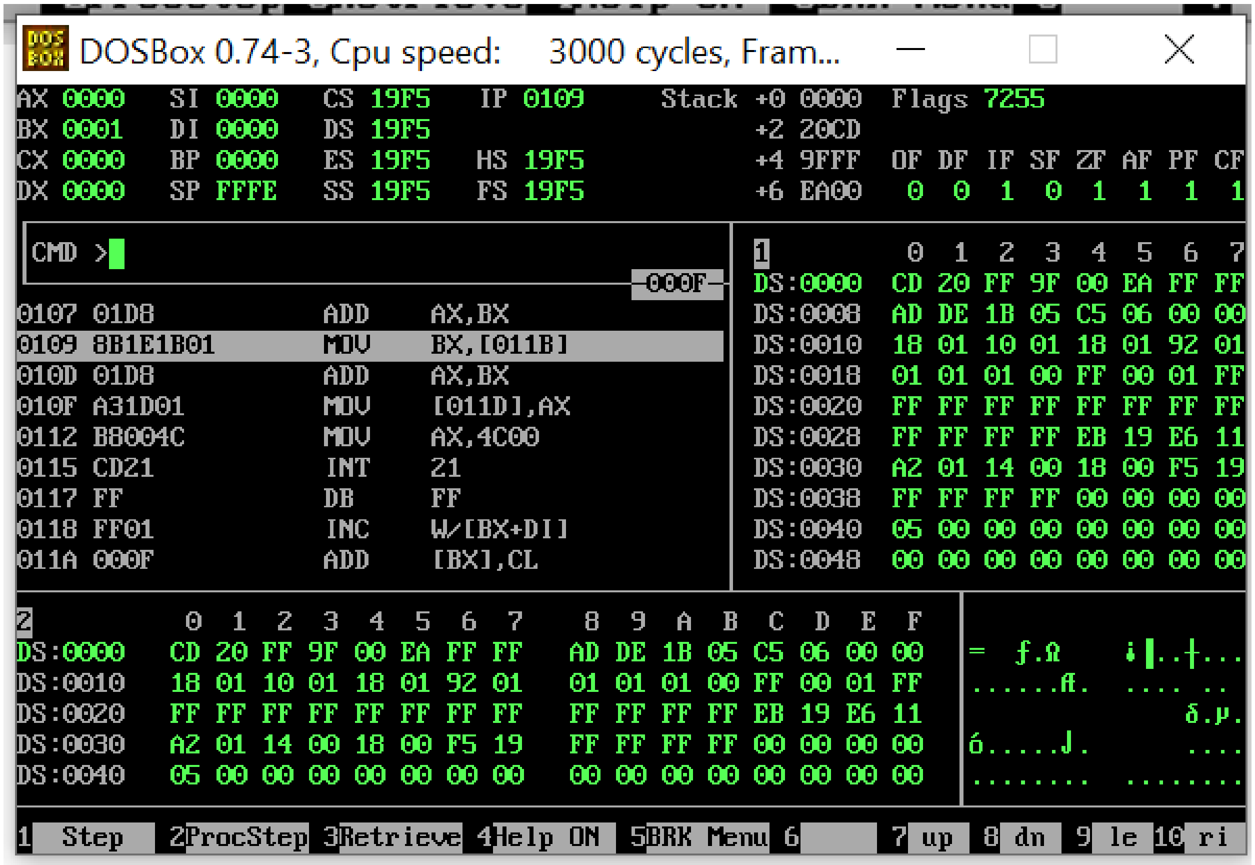
Before:



```
DOSBox 0.74-3, Cpu speed:     3000 cycles, Fram...
AX FFFF    SI 0000   CS 19F5    IP 0107      Stack +0 0000  Flags 7200
BX 0001    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
0103 8B1E1901        MOV     BX,[0119]            DS:0008  AD DE 1B 05 C5 06 00 00
0107 01D8            ADD     AX,BX                DS:0010  18 01 10 01 18 01 92 01
0109 8B1E1B01        MOV     BX,[011B]            DS:0018  01 01 01 00 FF 00 01 FF
010D 01D8            ADD     AX,BX                DS:0020  FF FF FF FF FF FF FF FF
010F A31D01          MOV     [011D],AX            DS:0028  FF FF FF FF EB 19 E6 11
0112 B8004C          MOV     AX,4C00              DS:0030  A2 01 14 00 18 00 F5 19
0115 CD21            INT     21                   DS:0038  FF FF FF FF 00 00 00 00
0117 FF              DB      FF                   DS:0040  05 00 00 00 00 00 00 00
0118 FF01            INC     W/[BX+DI]            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 FF 00 01 FF   ......Æ.  ....  ..
DS:0020       FF FF FF FF FF FF FF FF   FF FF FF FF EB 19 E6 11               δ.μ.
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
```

After:

When adding 65535 (which is the largest value a 16-bit register like AX can hold) to 1, the calculated result exceeds the register's capacity. This causes a carry, a signal that an overflow has occurred. The carry flag is specifically set to 1 to indicate this condition.



DOSBox 0.74-3, Cpu speed: 3000 cycles, Fram...

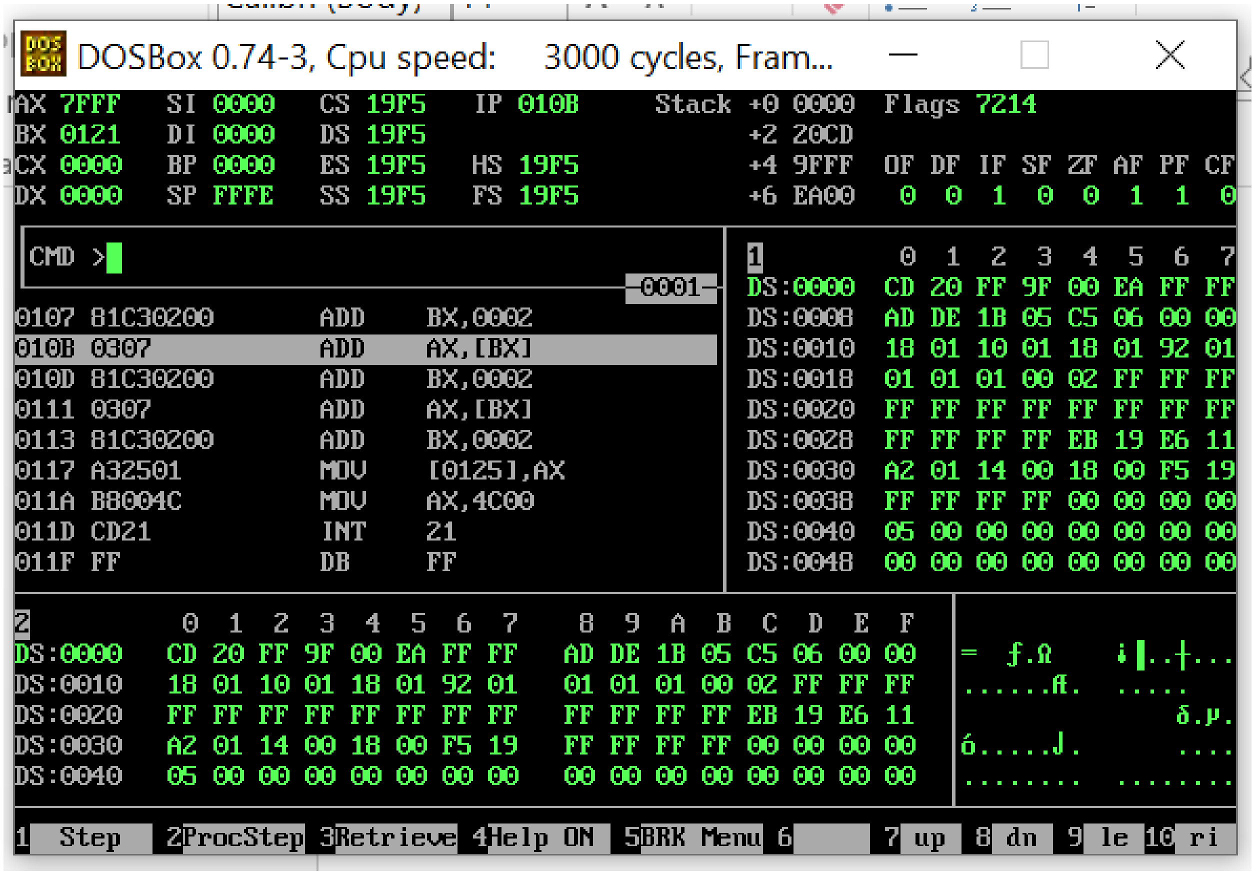
```
AX 0000    SI 0000    CS 19F5    IP 0109       Stack +0 0000  Flags 7255
BX 0001    DI 0000    DS 19F5                        +2 20CD
CX 0000    BP 0000    ES 19F5    HS 19F5             +4 9FFF  OF DF IF SF ZF AF PF CF
DX 0000    SP FFFE    SS 19F5    FS 19F5             +6 EA00   0  0  1  0  1  1  1  1

 CMD >                                              1          0  1  2  3  4  5  6  7
                                         -000F-     DS:0000  CD 20 FF 9F 00 EA FF FF
0107 01D8            ADD     AX,BX                  DS:0008  AD DE 1B 05 C5 06 00 00
0109 8B1E1B01        MOV     BX,[011B]              DS:0010  18 01 10 01 18 01 92 01
010D 01D8            ADD     AX,BX                  DS:0018  01 01 01 00 FF 00 01 FF
010F A31D01          MOV     [011D],AX              DS:0020  FF FF FF FF FF FF FF FF
0112 B8004C          MOV     AX,4C00                DS:0028  FF FF FF FF EB 19 E6 11
0115 CD21            INT     21                     DS:0030  A2 01 14 00 18 00 F5 19
0117 FF              DB      FF                     DS:0038  FF FF FF FF 00 00 00 00
0118 FF01            INC     W/[BX+DI]              DS:0040  05 00 00 00 00 00 00 00
011A 000F            ADD     [BX],CL                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 FF 00 01 FF   ......Æ.  ....  ..
DS:0020       FF FF FF FF FF FF FF FF   FF FF FF FF EB 19 E6 11                 δ.µ.
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
```

7: Signed Flag

```
; a program to add three numbers using byte variables

[org 0x0100]

   xor  ax, ax    ; check effect on ZF


   mov  bx, num1


   add  ax, [bx]

   add  bx, 2


   add  ax, [bx]

   add  bx, 2
```

```
    add  ax, [bx]
    add  bx, 2


    mov  [result], ax

    mov  ax, 0x4c00
    int  0x21


    ; to turn this into an iteration, we need a couple of things:
    ; - branching instruction
    ; - checking constraints -- e.g. c > 0       ; Intel Sotware Developer Manual - Figure 3-8 (Page 80)


num1: dw  32767, 1, 15
result: dw 0
```

Before:

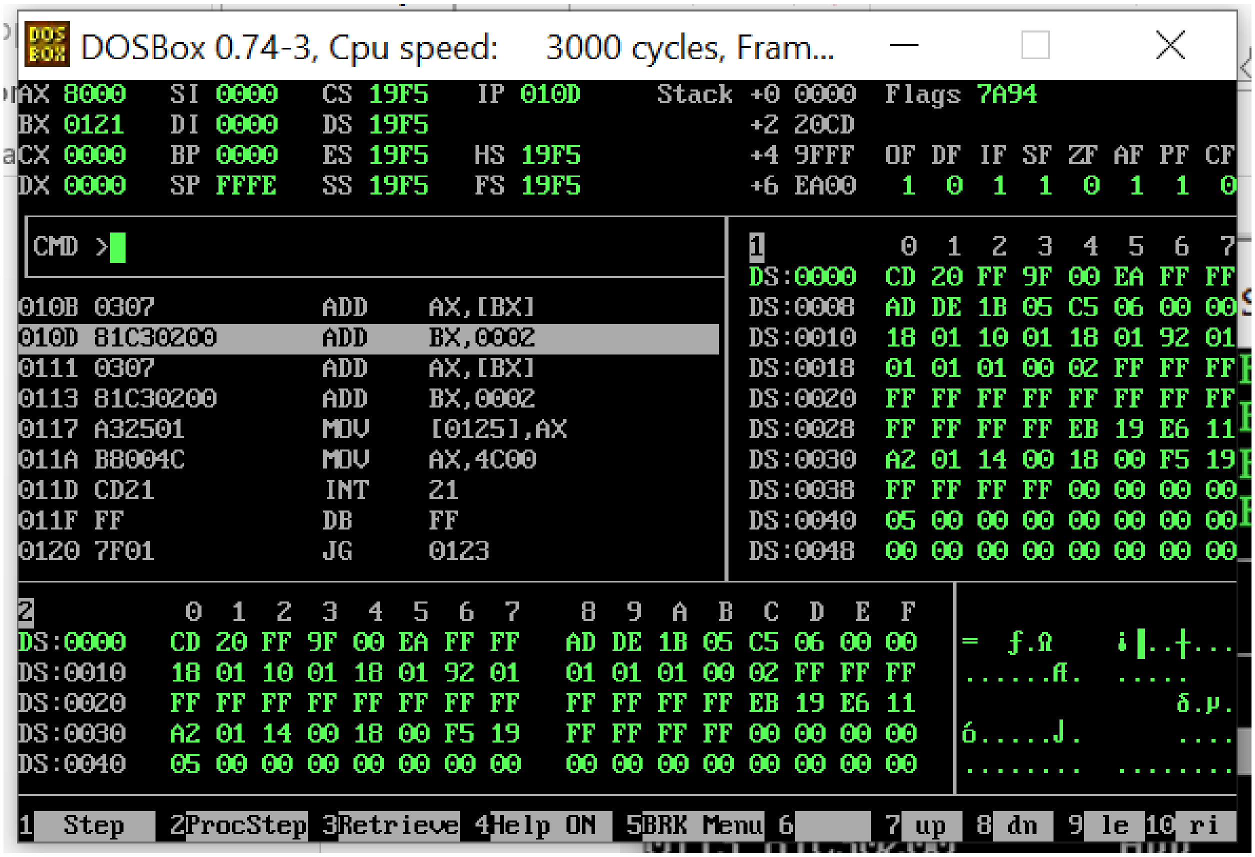


DOSBox 0.74-3, Cpu speed: 3000 cycles, Fram...

```
AX 7FFF   SI 0000   CS 19F5   IP 010B     Stack +0 0000  Flags 7214
BX 0121   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  1  1  0

 CMD >                                          1         0  1  2  3  4  5  6  7
                                       -0001-   DS:0000  CD 20 FF 9F 00 EA FF FF
0107 81C30200       ADD    BX,0002              DS:0008  AD DE 1B 05 C5 06 00 00
010B 0307           ADD    AX,[BX]              DS:0010  18 01 10 01 18 01 92 01
010D 81C30200       ADD    BX,0002              DS:0018  01 01 01 00 02 FF FF FF
0111 0307           ADD    AX,[BX]              DS:0020  FF FF FF FF FF FF FF FF
0113 81C30200       ADD    BX,0002              DS:0028  FF FF FF FF EB 19 E6 11
0117 A32501         MOV    [0125],AX            DS:0030  A2 01 14 00 18 00 F5 19
011A B8004C         MOV    AX,4C00              DS:0038  FF FF FF FF 00 00 00 00
011D CD21           INT    21                   DS:0040  05 00 00 00 00 00 00 00
011F FF             DB     FF                   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 E6 11               δ.µ.
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   ........ ........
```

After:

When we add 32767(0111111111111111) to 1 it becomes (1000000000000000) resulting in a transition to a negative value, thus setting the sign flag to 1.



```
AX 8000   SI 0000   CS 19F5   IP 010D      Stack +0 0000  Flags 7A94
BX 0121   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   1  0  1  1  0  1  1  0

 CMD >                                        1          0  1  2  3  4  5  6  7
                                              DS:0000   CD 20 FF 9F 00 EA FF FF
010B 0307           ADD    AX,[BX]            DS:0008   AD DE 1B 05 C5 06 00 00
010D 81C30200       ADD    BX,0002            DS:0010   18 01 10 01 18 01 92 01
0111 0307           ADD    AX,[BX]            DS:0018   01 01 01 00 02 FF FF FF
0113 81C30200       ADD    BX,0002            DS:0020   FF FF FF FF FF FF FF FF
0117 A32501         MOV    [0125],AX          DS:0028   FF FF FF FF EB 19 E6 11
011A B8004C         MOV    AX,4C00            DS:0030   A2 01 14 00 18 00 F5 19
011D CD21           INT    21                 DS:0038   FF FF FF FF 00 00 00 00
011F FF             DB     FF                 DS:0040   05 00 00 00 00 00 00 00
0120 7F01           JG     0123               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 E6 11                δ.µ.
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   ........  ........
```