Name: Khizar Ali

Roll No: 22P-9269

Subject: COAL_Lab

Submitted to: Usman Abbasi

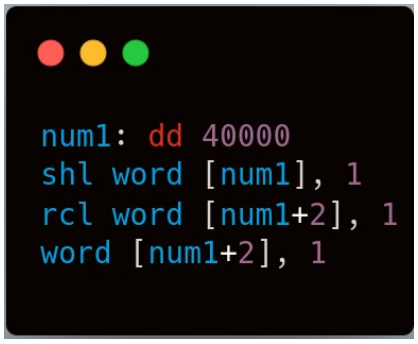
Problem Statement:

When multiplying numbers, especially larger ones that exceed the register size (typically 16 or 32 bits), conventional shifting operations like SHL and SHR may not be sufficient. These operations can only manipulate bits within the register size, leading to loss of significant bits when dealing with larger numbers. For instance, when multiplying two 16-bit numbers, the result can be up to 32 bits long, but if we're limited to 16-bit registers, we face challenges storing and processing the partial products without encountering overflow issues.

Solution:

To address the memory limitation, we employ extended shifting. This technique hinges on two primary instructions: SHL (Shift Left) and RCL (Rotate Carry Left). By utilizing this method, we can shift a 32-bit number left by 16 bits, thereby safeguarding against the loss of significant bits.

For Example:



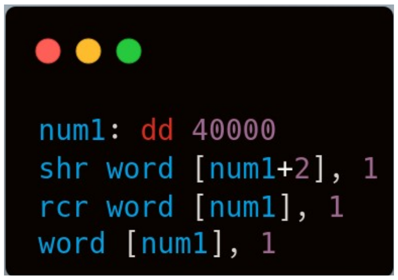
```
num1: dd 40000
shl word [num1], 1
rcl word [num1+2], 1
word [num1+2], 1
```

Explanation:

In this example, **num1** represents a **32-bit** number stored in memory. The **SHL** instruction performs a left shift operation on the lower 16 bits of **num1**, causing the most significant bit to be shifted out and stored in the carry flag. Subsequently, the **RCL** instruction rotates the carry flag into the least significant bit of the next word, thereby effectively combining the two 16-bit words.

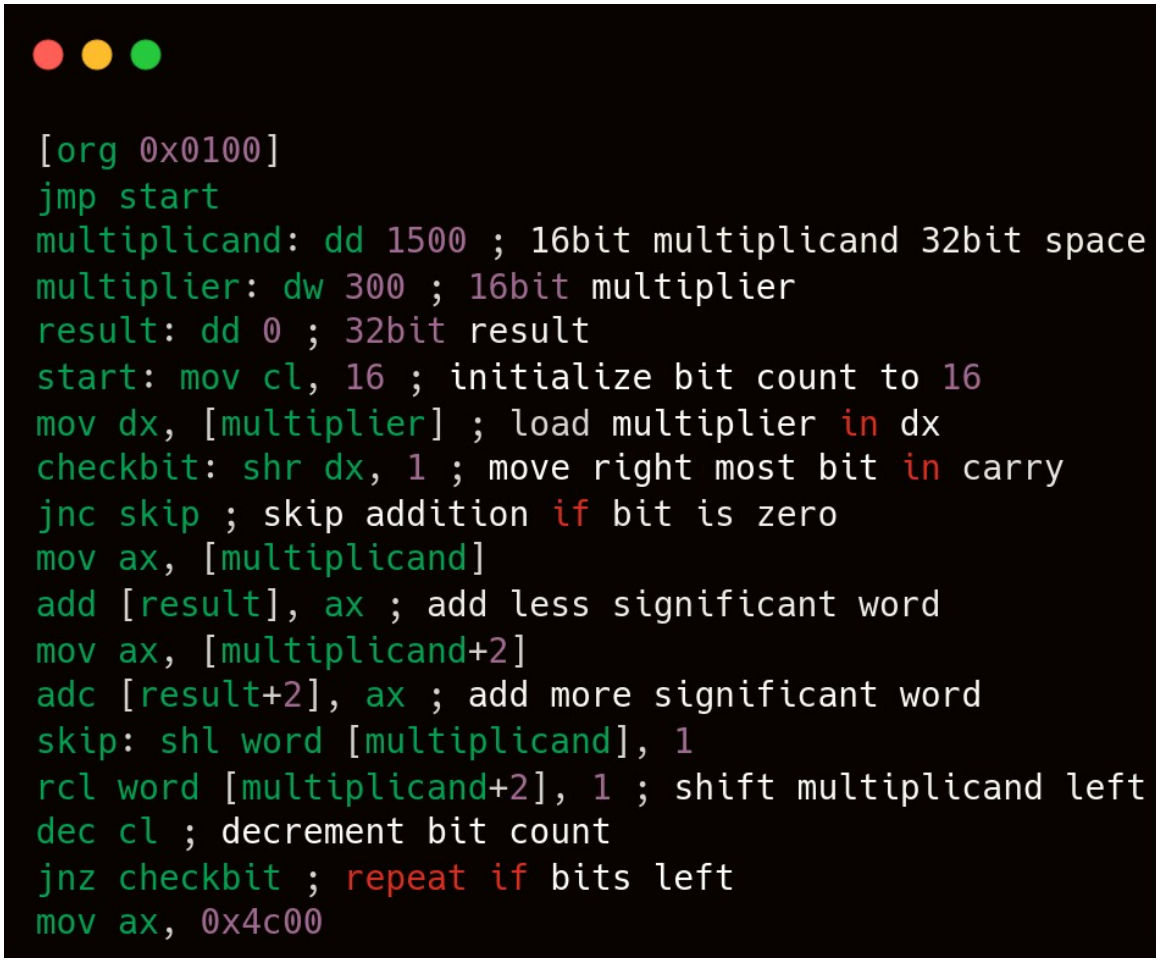
For shifting right, the process is reversed. The SHR (Shift Right) and RCR instructions are used

to ensure that no valuable bit is lost.



```
num1: dd 40000
shr word [num1+2], 1
rcr word [num1], 1
word [num1], 1
```

Code:



```
[org 0x0100]
jmp start
multiplicand: dd 1500 ; 16bit multiplicand 32bit space
multiplier: dw 300 ; 16bit multiplier
result: dd 0 ; 32bit result
start: mov cl, 16 ; initialize bit count to 16
mov dx, [multiplier] ; load multiplier in dx
checkbit: shr dx, 1 ; move right most bit in carry
jnc skip ; skip addition if bit is zero
mov ax, [multiplicand]
add [result], ax ; add less significant word
mov ax, [multiplicand+2]
adc [result+2], ax ; add more significant word
skip: shl word [multiplicand], 1
rcl word [multiplicand+2], 1 ; shift multiplicand left
dec cl ; decrement bit count
jnz checkbit ; repeat if bits left
mov ax, 0x4c00
```

Explanation of the code:

1. **[org 0x0100]**: This directive sets the origin of the program to memory address 0x0100.

2. **jmp start**: This instruction jumps to the **start** **label** to begin execution of the program.

3. **multiplicand: dd 1500**: This line declares a double word (**32 bits**) variable named **multiplicand** and initializes it with the value **1500** (in decimal). It allocates **32 bits** of memory for the multiplicand.

4. **multiplier: dw 300:** This line declares a word (**16 bits**) variable named multiplier and initializes it with the value **300 (in decimal)**. It allocates **16 bits** of memory for the multiplier.

5. **result: dd 0:** This line declares a double word (**32 bits**) variable named **result** and initializes it with the **value 0**. It allocates **32 bits** of memory for the result.

6. **start: mov cl, 16:** This instruction moves the value **16** into the **CL** register, initializing the bit count to **16**.

7. **mov dx, [multiplier]:** This instruction loads the value stored in the **multiplier** variable into the **DX** register.

8. **checkbit: shr dx, 1**: This instruction performs a **bitwise** shift right operation on the DX register, effectively shifting all bits to the right by one position. **The rightmost bit is moved into the carry flag.**

9. **jnc skip:** This instruction **jumps** to the **skip** label if the carry flag is not **set** (jumps if no carry).

10. **mov ax, [multiplicand]**: This instruction loads the **lower** 16 bits of the **multiplicand** variable into the **AX** register.

11. **add [result], ax:** This instruction adds the value in the **AX** register to the **memory location** pointed to by the result variable, effectively adding the less **significant word** of the **multiplicand** to the result.

12. **mov ax, [multiplicand+2]**: This instruction loads the **upper** **16 bits** of the **multiplicand** variable into the **AX** register.

13. **adc [result+2], ax**: This instruction adds the value in the **AX** register to the memory location **two bytes after result**, taking into **account** the carry flag from the **previous addition.**

14. **skip: shl word [multiplicand], 1**: This instruction performs a **bitwise** shift **left operation** on the lower **16 bits of the multiplicand variable**, effectively shifting it left by one position.

15. **rcl word [multiplicand+2], 1:** This instruction **rotates** the **carry flag** into the least significant bit of the **upper 16 bits** of the **multiplicand** variable, effectively shifting it left by **one position** and **preserving** the **carry** from the previous operation.

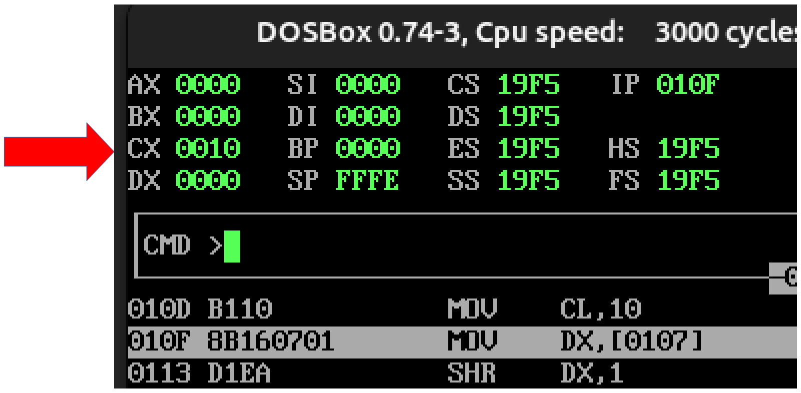
16. **dec cl**: This instruction **decrements** the value in the **CL** register.

17. **jnz checkbit:** This instruction jumps back to the **checkbit** label if the zero flag is not set (jumps if the bit count is not zero).

18. **<u>mov ax, 0x4c00:</u>** This instruction moves the value **<u>0x4c00</u>** into the AX register, which is typically used to signal the end of a program in DOS.

<u>Step by step execution of the code on DOS:</u>

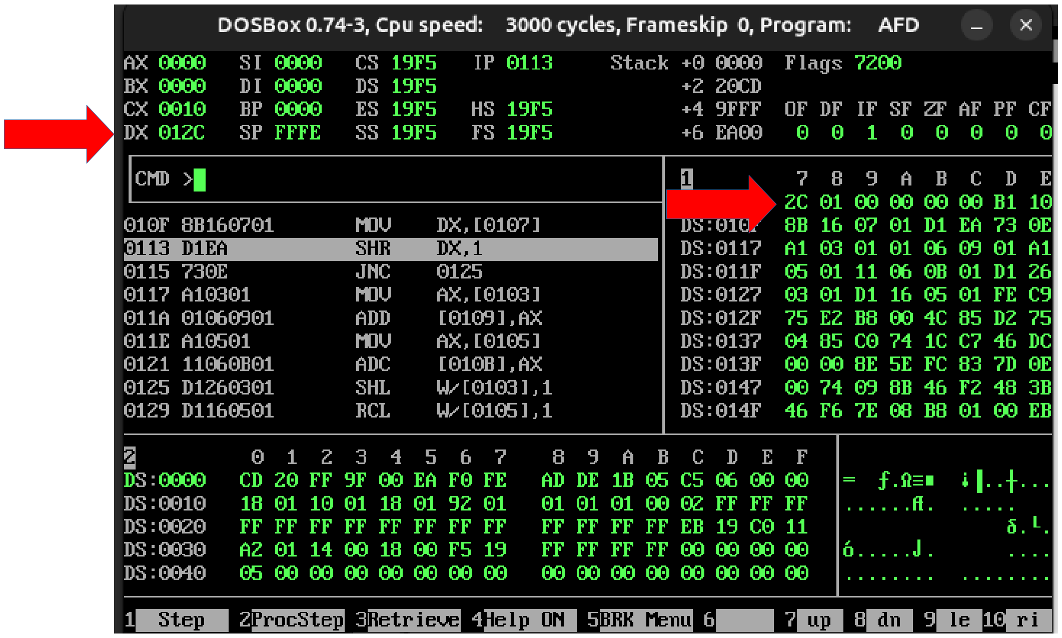
<u>Step 1:</u>



```
DOSBox 0.74-3, Cpu speed:   3000 cycle
AX 0000   SI 0000   CS 19F5   IP 010F
BX 0000   DI 0000   DS 19F5
CX 0010   BP 0000   ES 19F5   HS 19F5
DX 0000   SP FFFE   SS 19F5   FS 19F5

 CMD >

010D B110              MOV    CL,10
010F 8B160701          MOV    DX,[0107]
0113 D1EA              SHR    DX,1
```

Move the value **16** in **<u>CX</u>** register (10) in HEX

<u>Step 2:</u>

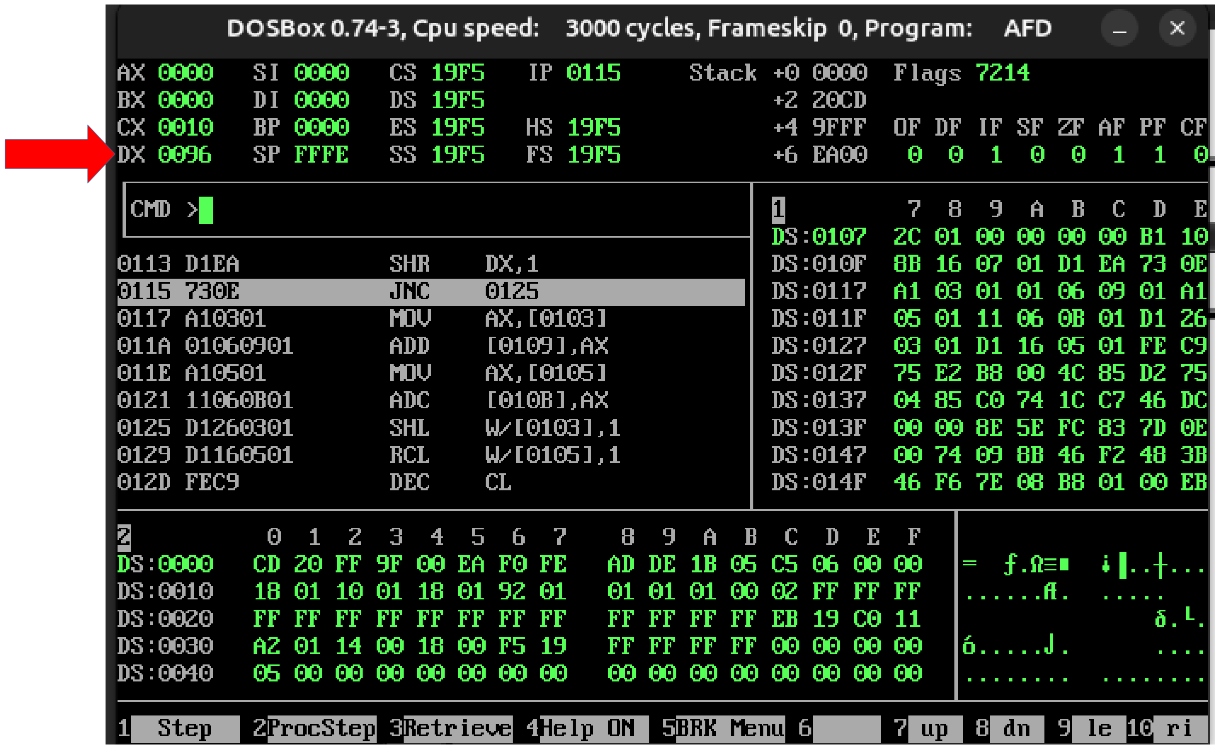


```
DOSBox 0.74-3, Cpu speed:   3000 cycles, Frameskip 0, Program:   AFD
AX 0000   SI 0000   CS 19F5   IP 0113     Stack +0 0000  Flags 7200
BX 0000   DI 0000   DS 19F5                     +2 20CD
CX 0010   BP 0000   ES 19F5   HS 19F5           +4 9FFF  OF DF IF SF ZF AF PF CF
DX 012C   SP FFFE   SS 19F5   FS 19F5           +6 EA00   0  0  1  0  0  0  0  0

 CMD >                                          1       7  8  9  A  B  C  D  E
                                                        2C 01 00 00 00 00 B1 10
010F 8B160701        MOV    DX,[0107]           DS:010   8B 16 07 01 D1 EA 73 0E
0113 D1EA            SHR    DX,1                DS:0117 A1 03 01 01 06 09 01 A1
0115 730E            JNC    0125                DS:011F 05 01 11 06 0B 01 D1 26
0117 A10301          MOV    AX,[0103]           DS:0127 03 01 D1 16 05 01 FE C9
011A 01060901        ADD    [0109],AX           DS:012F 75 E2 B8 00 4C 85 D2 75
011E A10501          MOV    AX,[0105]           DS:0137 04 85 C0 74 1C C7 46 DC
0121 11060B01        ADC    [010B],AX           DS:013F 00 00 8E 5E FC 83 7D 0E
0125 D1260301        SHL    W/[0103],1          DS:0147 00 74 09 8B 46 F2 48 3B
0129 D1160501        RCL    W/[0105],1          DS:014F 46 F6 7E 08 B8 01 00 EB

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
```

Moved the value of **<u>multiplier (300)</u>** 012C in HEX in the register DX

<u>Step 3:</u>

<u>loop starts itertation :1</u>

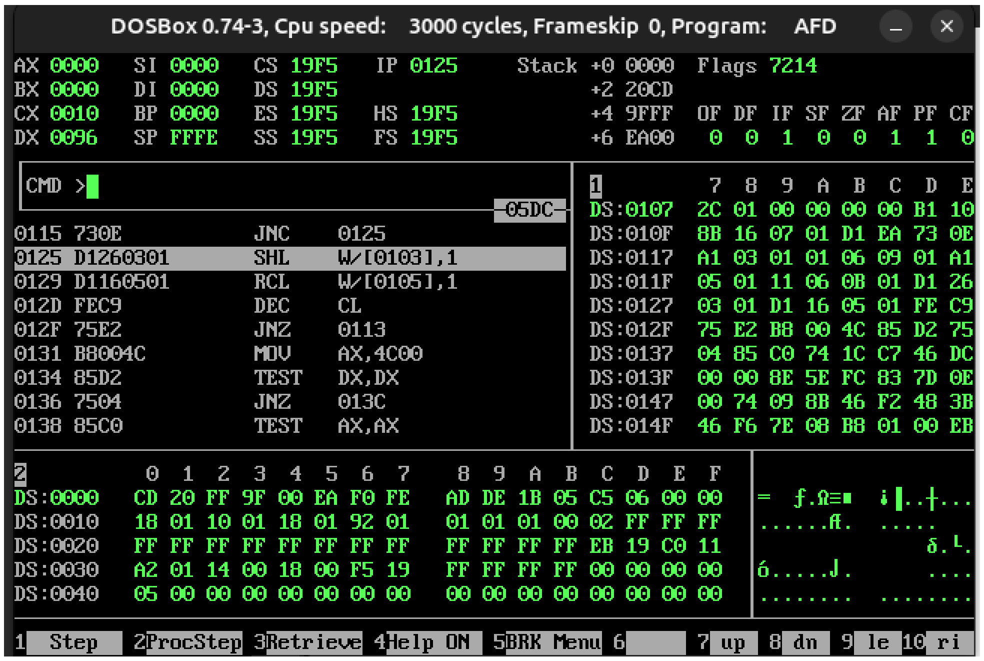


```
DOSBox 0.74-3, Cpu speed:   3000 cycles, Frameskip 0, Program:   AFD
AX 0000   SI 0000   CS 19F5   IP 0115     Stack +0 0000  Flags 7214
BX 0000   DI 0000   DS 19F5               +2 20CD
CX 0010   BP 0000   ES 19F5   HS 19F5     +4 9FFF  OF DF IF SF ZF AF PF CF
DX 0096   SP FFFE   SS 19F5   FS 19F5     +6 EA00   0  0  1  0  0  1  1  0

 CMD >                                   1        7  8  9  A  B  C  D  E
                                         DS:0107  2C 01 00 00 00 00 B1 10
0113 D1EA        SHR    DX,1             DS:010F  8B 16 07 01 D1 EA 73 0E
0115 730E        JNC    0125             DS:0117  A1 03 01 01 06 09 01 A1
0117 A10301      MOV    AX,[0103]        DS:011F  05 01 11 06 0B 01 D1 26
011A 01060901    ADD    [0109],AX        DS:0127  03 01 D1 16 05 01 FE C9
011E A10501      MOV    AX,[0105]        DS:012F  75 E2 B8 00 4C 85 D2 75
0121 11060B01    ADC    [010B],AX        DS:0137  04 85 C0 74 1C C7 46 DC
0125 D1260301    SHL    W/[0103],1       DS:013F  00 00 8E 5E FC 83 7D 0E
0129 D1160501    RCL    W/[0105],1       DS:0147  00 74 09 8B 46 F2 48 3B
012D FEC9        DEC    CL               DS:014F  46 F6 7E 08 B8 01 00 EB

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
```

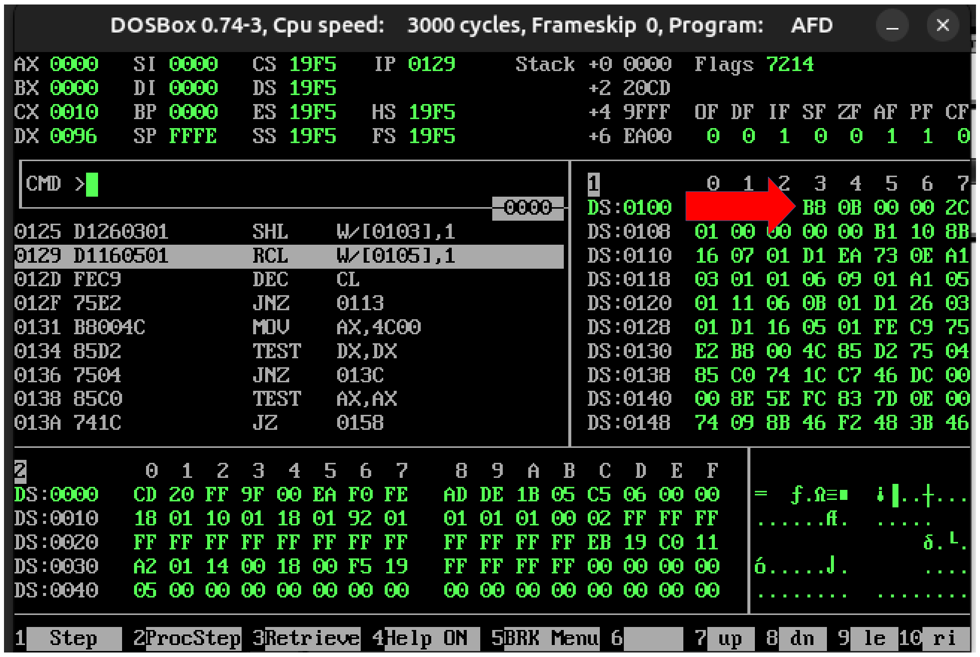
Shifted the multiplier right by one bit and moved the rightmost bit to the carry as the Value of **<u>DX</u>** changed to **<u>0096</u>**

<u>jnc skip ; skip the addition</u>

<u>As carry flag is not set we are skipping the addition.</u>



```
DOSBox 0.74-3, Cpu speed:   3000 cycles, Frameskip 0, Program:   AFD
AX 0000   SI 0000   CS 19F5   IP 0125     Stack +0 0000  Flags 7214
BX 0000   DI 0000   DS 19F5               +2 20CD
CX 0010   BP 0000   ES 19F5   HS 19F5     +4 9FFF  OF DF IF SF ZF AF PF CF
DX 0096   SP FFFE   SS 19F5   FS 19F5     +6 EA00   0  0  1  0  0  1  1  0

 CMD >                                   1        7  8  9  A  B  C  D  E
                                  05DC   DS:0107  2C 01 00 00 00 00 B1 10
0115 730E        JNC    0125             DS:010F  8B 16 07 01 D1 EA 73 0E
0125 D1260301    SHL    W/[0103],1       DS:0117  A1 03 01 01 06 09 01 A1
0129 D1160501    RCL    W/[0105],1       DS:011F  05 01 11 06 0B 01 D1 26
012D FEC9        DEC    CL               DS:0127  03 01 D1 16 05 01 FE C9
012F 75E2        JNZ    0113             DS:012F  75 E2 B8 00 4C 85 D2 75
0131 B8004C      MOV    AX,4C00          DS:0137  04 85 C0 74 1C C7 46 DC
0134 85D2        TEST   DX,DX            DS:013F  00 00 8E 5E FC 83 7D 0E
0136 7504        JNZ    013C             DS:0147  00 74 09 8B 46 F2 48 3B
0138 85C0        TEST   AX,AX            DS:014F  46 F6 7E 08 B8 01 00 EB

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
```

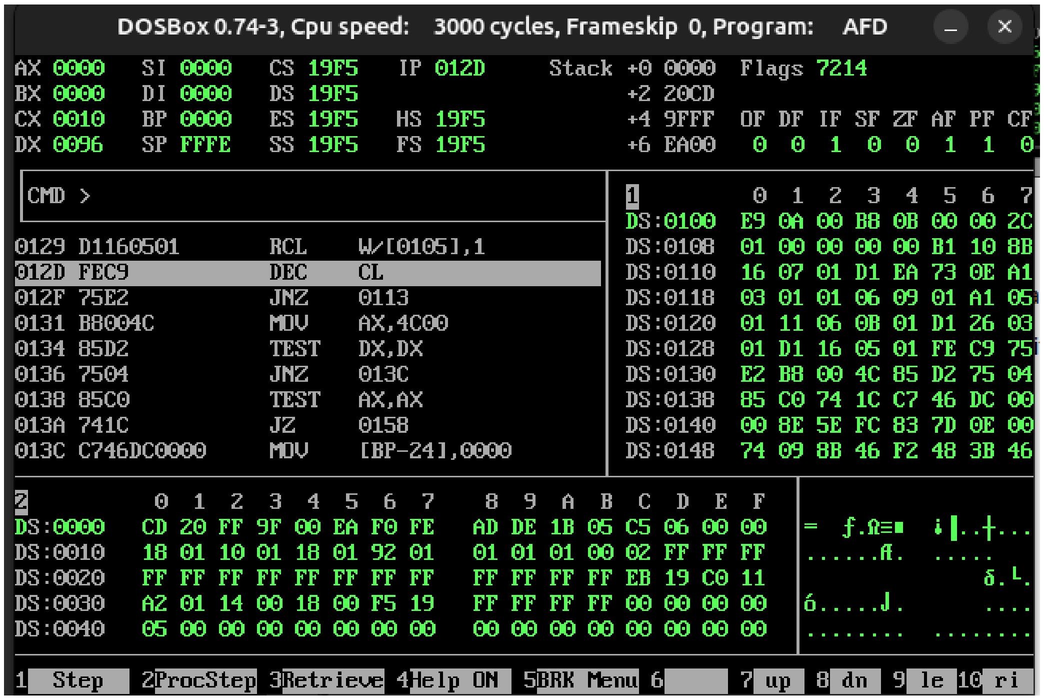


```
DOSBox 0.74-3, Cpu speed:   3000 cycles, Frameskip  0, Program:   AFD
AX 0000   SI 0000   CS 19F5   IP 0129      Stack +0 0000  Flags 7214
BX 0000   DI 0000   DS 19F5                      +2 20CD
CX 0010   BP 0000   ES 19F5   HS 19F5            +4 9FFF  OF DF IF SF ZF AF PF CF
DX 0096   SP FFFE   SS 19F5   FS 19F5            +6 EA00   0  0  1  0  0  1  1  0

CMD >                                  1        0  1  2  3  4  5  6  7
                              -0000-   DS:0100           B8 0B 00 00 2C
0125 D1260301     SHL   W/[0103],1     DS:0108 01 00 00 00 00 B1 10 8B
0129 D1160501     RCL   W/[0105],1     DS:0110 16 07 01 D1 EA 73 0E A1
012D FEC9         DEC   CL             DS:0118 03 01 01 06 09 01 A1 05
012F 75E2         JNZ   0113           DS:0120 01 11 06 0B 01 D1 26 03
0131 B8004C       MOV   AX,4C00        DS:0128 01 D1 16 05 01 FE C9 75
0134 85D2         TEST  DX,DX          DS:0130 E2 B8 00 4C 85 D2 75 04
0136 7504         JNZ   013C           DS:0138 85 C0 74 1C C7 46 DC 00
0138 85C0         TEST  AX,AX          DS:0140 00 8E 5E FC 83 7D 0E 00
013A 741C         JZ    0158           DS:0148 74 09 8B 46 F2 48 3B 46

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
```

skip: shl word [multiplicand], 1

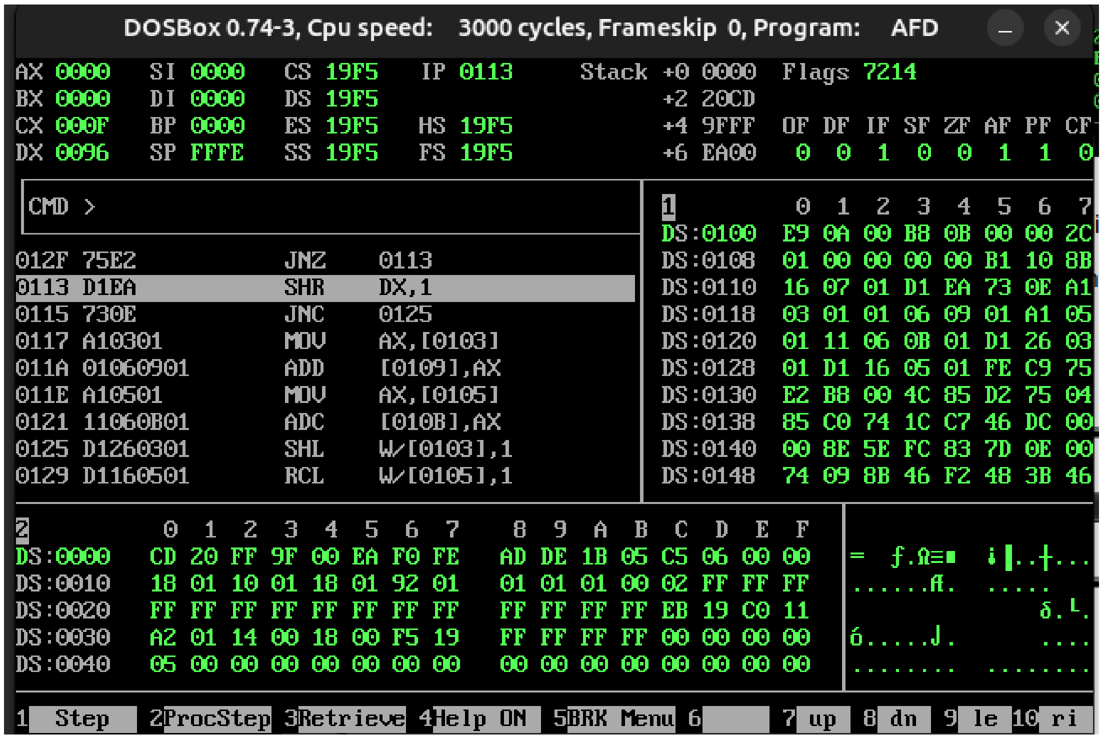
This instruction shifted the bits in the lower 16 bits of multiplicand to the left by one position.

The leftmost bit is moved to the carry flag, and the rightmost bit is filled with a 0.



```
DOSBox 0.74-3, Cpu speed:   3000 cycles, Frameskip  0, Program:   AFD
AX 0000   SI 0000   CS 19F5   IP 012D      Stack +0 0000  Flags 7214
BX 0000   DI 0000   DS 19F5                      +2 20CD
CX 0010   BP 0000   ES 19F5   HS 19F5            +4 9FFF  OF DF IF SF ZF AF PF CF
DX 0096   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:0100 E9 0A 00 B8 0B 00 00 2C
0129 D1160501     RCL   W/[0105],1     DS:0108 01 00 00 00 00 B1 10 8B
012D FEC9         DEC   CL             DS:0110 16 07 01 D1 EA 73 0E A1
012F 75E2         JNZ   0113           DS:0118 03 01 01 06 09 01 A1 05
0131 B8004C       MOV   AX,4C00        DS:0120 01 11 06 0B 01 D1 26 03
0134 85D2         TEST  DX,DX          DS:0128 01 D1 16 05 01 FE C9 75
0136 7504         JNZ   013C           DS:0130 E2 B8 00 4C 85 D2 75 04
0138 85C0         TEST  AX,AX          DS:0138 85 C0 74 1C C7 46 DC 00
013A 741C         JZ    0158           DS:0140 00 8E 5E FC 83 7D 0E 00
013C C746DC0000   MOV   [BP-24],0000   DS:0148 74 09 8B 46 F2 48 3B 46

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
```

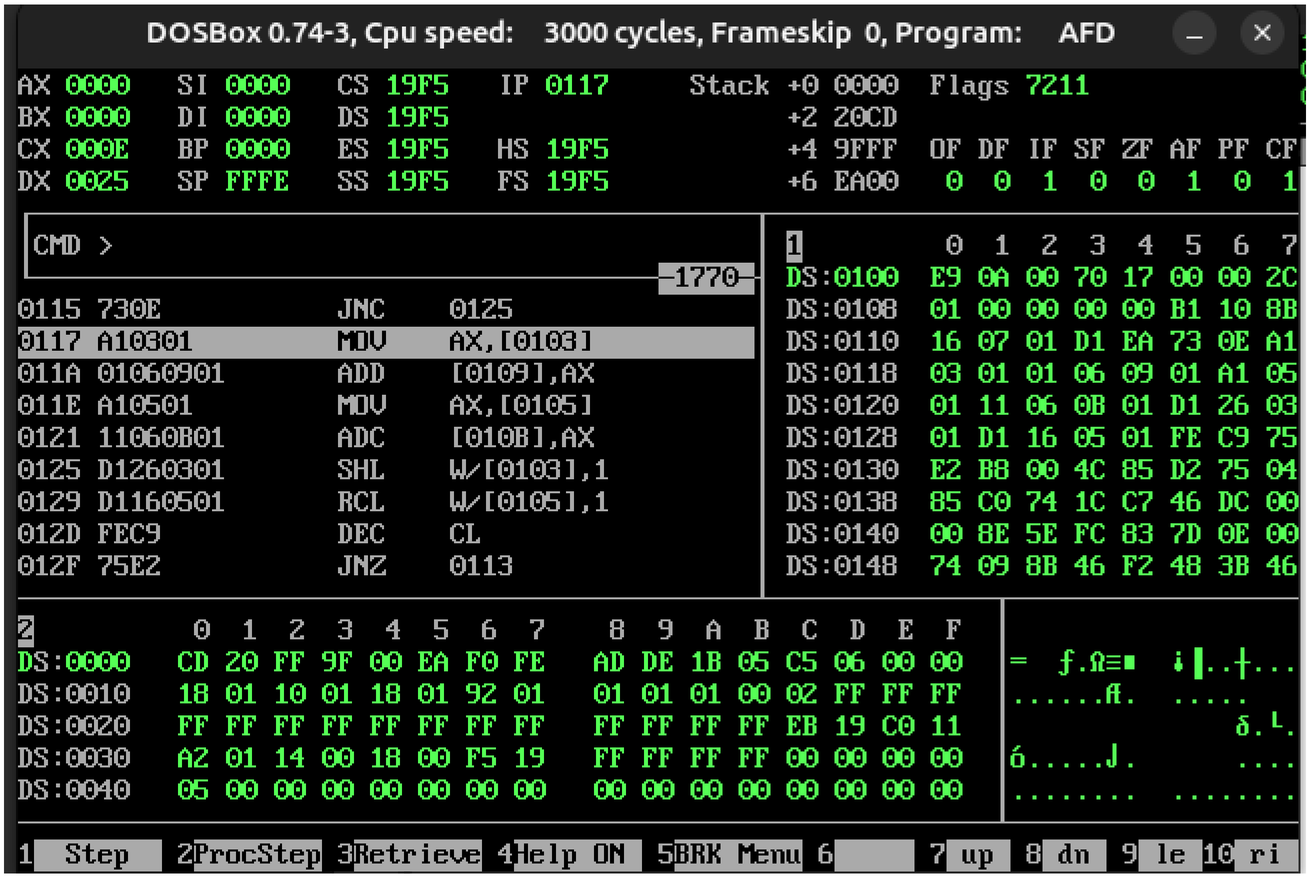
This instruction rotated the carry left into the least significant bit of the upper 16 bits of multiplicand. Since the carry flag was not set by the previous shift operation,it remain same



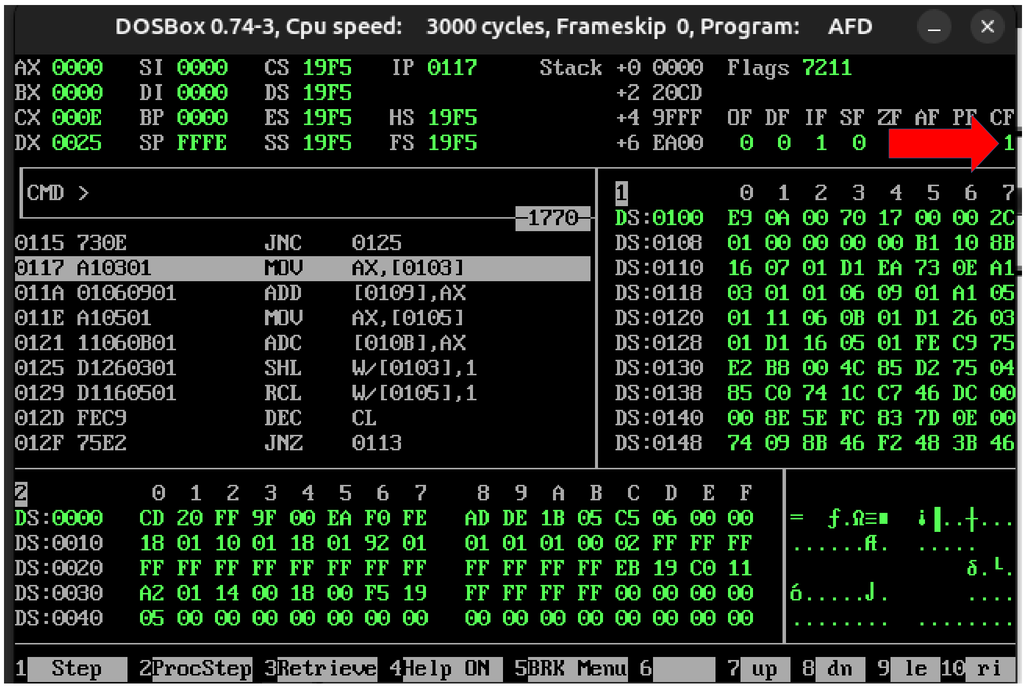
```
DOSBox 0.74-3, Cpu speed:   3000 cycles, Frameskip  0, Program:   AFD
AX 0000   SI 0000   CS 19F5   IP 0113     Stack +0 0000  Flags 7214
BX 0000   DI 0000   DS 19F5                     +2 20CD
CX 000F   BP 0000   ES 19F5   HS 19F5           +4 9FFF  OF DF IF SF ZF AF PF CF
DX 0096   SP FFFE   SS 19F5   FS 19F5           +6 EA00   0  0  1  0  0  1  1  0

 CMD >                                        1        0  1  2  3  4  5  6  7
                                              DS:0100  E9 0A 00 B8 0B 00 00 2C
012F 75E2          JNZ     0113               DS:0108  01 00 00 00 00 B1 10 8B
0113 D1EA          SHR     DX,1               DS:0110  16 07 01 D1 EA 73 0E A1
0115 730E          JNC     0125               DS:0118  03 01 01 06 09 01 A1 05
0117 A10301        MOV     AX,[0103]          DS:0120  01 11 06 0B 01 D1 26 03
011A 01060901      ADD     [0109],AX          DS:0128  01 D1 16 05 01 FE C9 75
011E A10501        MOV     AX,[0105]          DS:0130  E2 B8 00 4C 85 D2 75 04
0121 11060B01      ADC     [010B],AX          DS:0138  85 C0 74 1C C7 46 DC 00
0125 D1260301      SHL     W/[0103],1         DS:0140  00 8E 5E FC 83 7D 0E 00
0129 D1160501      RCL     W/[0105],1         DS:0148  74 09 8B 46 F2 48 3B 46

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
```

AS CX was not zero jumped back to the checkbit labels

Iteration 2:

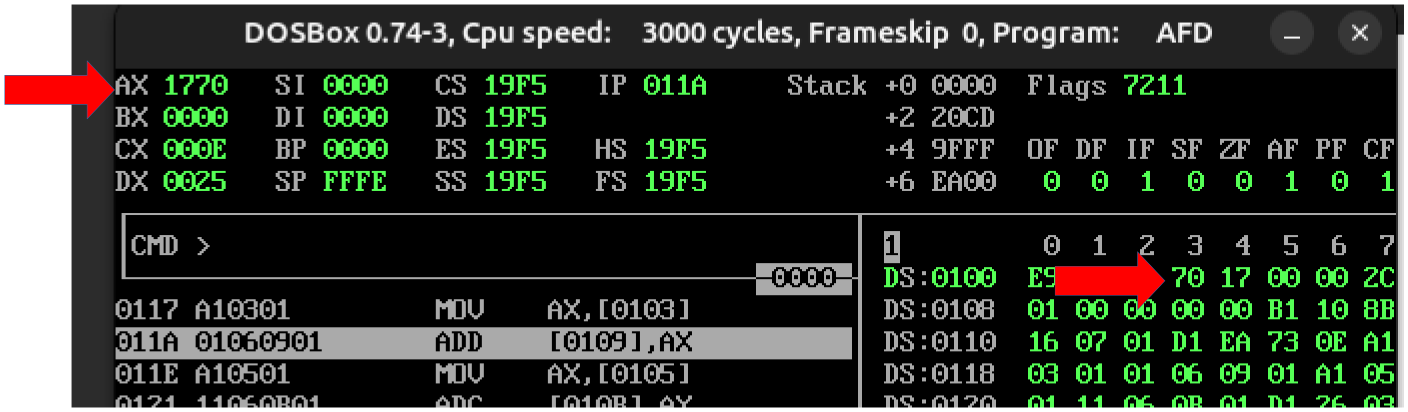


```
DOSBox 0.74-3, Cpu speed:   3000 cycles, Frameskip  0, Program:   AFD
AX 0000   SI 0000   CS 19F5   IP 0117     Stack +0 0000  Flags 7211
BX 0000   DI 0000   DS 19F5                     +2 20CD
CX 000E   BP 0000   ES 19F5   HS 19F5           +4 9FFF  OF DF IF SF ZF AF PF CF
DX 0025   SP FFFE   SS 19F5   FS 19F5           +6 EA00   0  0  1  0  0  1  0  1

 CMD >                                        1        0  1  2  3  4  5  6  7
                                   -1770-     DS:0100  E9 0A 00 70 17 00 00 2C
0115 730E          JNC     0125               DS:0108  01 00 00 00 00 B1 10 8B
0117 A10301        MOV     AX,[0103]          DS:0110  16 07 01 D1 EA 73 0E A1
011A 01060901      ADD     [0109],AX          DS:0118  03 01 01 06 09 01 A1 05
011E A10501        MOV     AX,[0105]          DS:0120  01 11 06 0B 01 D1 26 03
0121 11060B01      ADC     [010B],AX          DS:0128  01 D1 16 05 01 FE C9 75
0125 D1260301      SHL     W/[0103],1         DS:0130  E2 B8 00 4C 85 D2 75 04
0129 D1160501      RCL     W/[0105],1         DS:0138  85 C0 74 1C C7 46 DC 00
012D FEC9          DEC     CL                 DS:0140  00 8E 5E FC 83 7D 0E 00
012F 75E2          JNZ     0113               DS:0148  74 09 8B 46 F2 48 3B 46

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
```



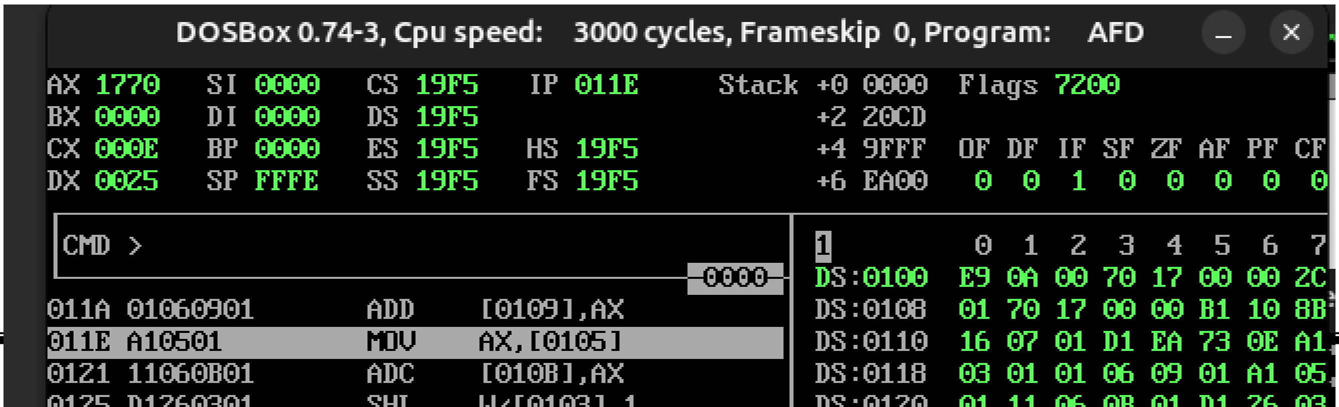
```
DOSBox 0.74-3, Cpu speed:   3000 cycles, Frameskip  0, Program:   AFD
AX 0000   SI 0000   CS 19F5   IP 0117     Stack +0 0000  Flags 7211
BX 0000   DI 0000   DS 19F5                     +2 20CD
CX 000E   BP 0000   ES 19F5   HS 19F5           +4 9FFF  OF DF IF SF ZF AF PF CF
DX 0025   SP FFFE   SS 19F5   FS 19F5           +6 EA00   0  0  1  0           1
 CMD >                                          1      0  1  2  3  4  5  6  7
                                          -1770-  DS:0100 E9 0A 00 70 17 00 00 2C
0115 730E             JNC     0125              DS:0108 01 00 00 00 00 B1 10 8B
0117 A10301           MOV     AX,[0103]         DS:0110 16 07 01 D1 EA 73 0E A1
011A 01060901         ADD     [0109],AX         DS:0118 03 01 01 06 09 01 A1 05
011E A10501           MOV     AX,[0105]         DS:0120 01 11 06 0B 01 D1 26 03
0121 11060B01         ADC     [010B],AX         DS:0128 01 D1 16 05 01 FE C9 75
0125 D1260301         SHL     W/[0103],1        DS:0130 E2 B8 00 4C 85 D2 75 04
0129 D1160501         RCL     W/[0105],1        DS:0138 85 C0 74 1C C7 46 DC 00
012D FEC9             DEC     CL                DS:0140 00 8E 5E FC 83 7D 0E 00
012F 75E2             JNZ     0113              DS:0148 74 09 8B 46 F2 48 3B 46
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
```

After shift right carry flag is set.



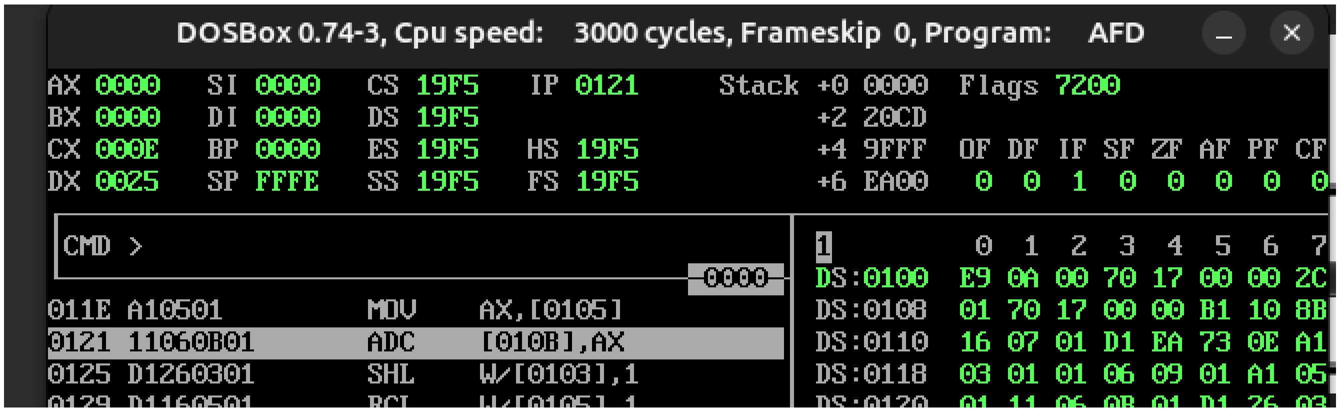
```
DOSBox 0.74-3, Cpu speed:   3000 cycles, Frameskip  0, Program:   AFD
AX 1770   SI 0000   CS 19F5   IP 011A     Stack +0 0000  Flags 7211
BX 0000   DI 0000   DS 19F5                     +2 20CD
CX 000E   BP 0000   ES 19F5   HS 19F5           +4 9FFF  OF DF IF SF ZF AF PF CF
DX 0025   SP FFFE   SS 19F5   FS 19F5           +6 EA00   0  0  1  0  0  1  0  1
 CMD >                                          1      0  1  2  3  4  5  6  7
                                          -0000-  DS:0100 E9       70 17 00 00 2C
0117 A10301           MOV     AX,[0103]         DS:0108 01 00 00 00 00 B1 10 8B
011A 01060901         ADD     [0109],AX         DS:0110 16 07 01 D1 EA 73 0E A1
011E A10501           MOV     AX,[0105]         DS:0118 03 01 01 06 09 01 A1 05
```

Moved the the value of multiplicand to AX

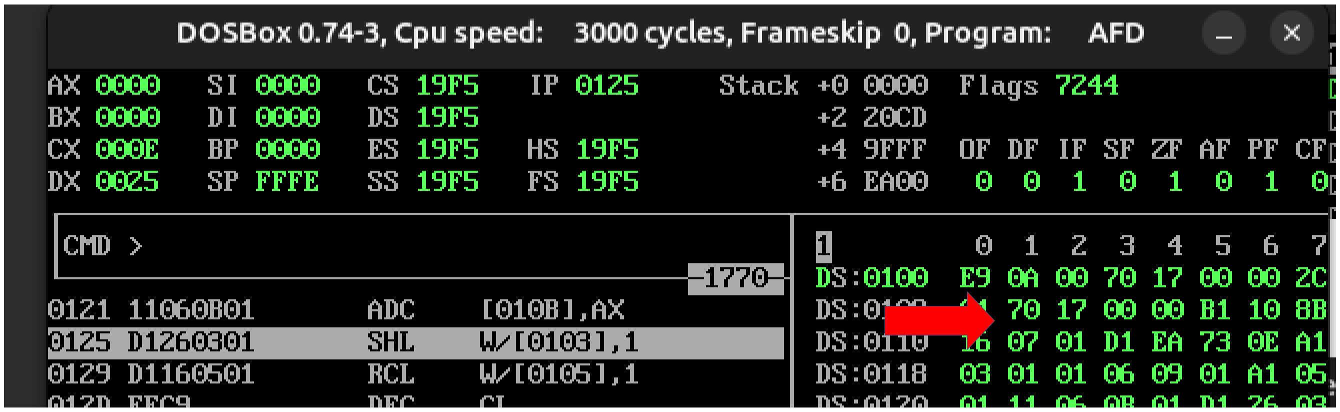


```
DOSBox 0.74-3, Cpu speed:   3000 cycles, Frameskip  0, Program:   AFD
AX 1770   SI 0000   CS 19F5   IP 011E     Stack +0 0000  Flags 7200
BX 0000   DI 0000   DS 19F5                     +2 20CD
CX 000E   BP 0000   ES 19F5   HS 19F5           +4 9FFF  OF DF IF SF ZF AF PF CF
DX 0025   SP FFFE   SS 19F5   FS 19F5           +6 EA00   0  0  1  0  0  0  0  0
 CMD >                                          1      0  1  2  3  4  5  6  7
                                          -0000-  DS:0100 E9 0A 00 70 17 00 00 2C
011A 01060901         ADD     [0109],AX         DS:0108 01 70 17 00 00 B1 10 8B
011E A10501           MOV     AX,[0105]         DS:0110 16 07 01 D1 EA 73 0E A1
0121 11060B01         ADC     [010B],AX         DS:0118 03 01 01 06 09 01 A1 05
```

Added the Value int the Ax register to the result

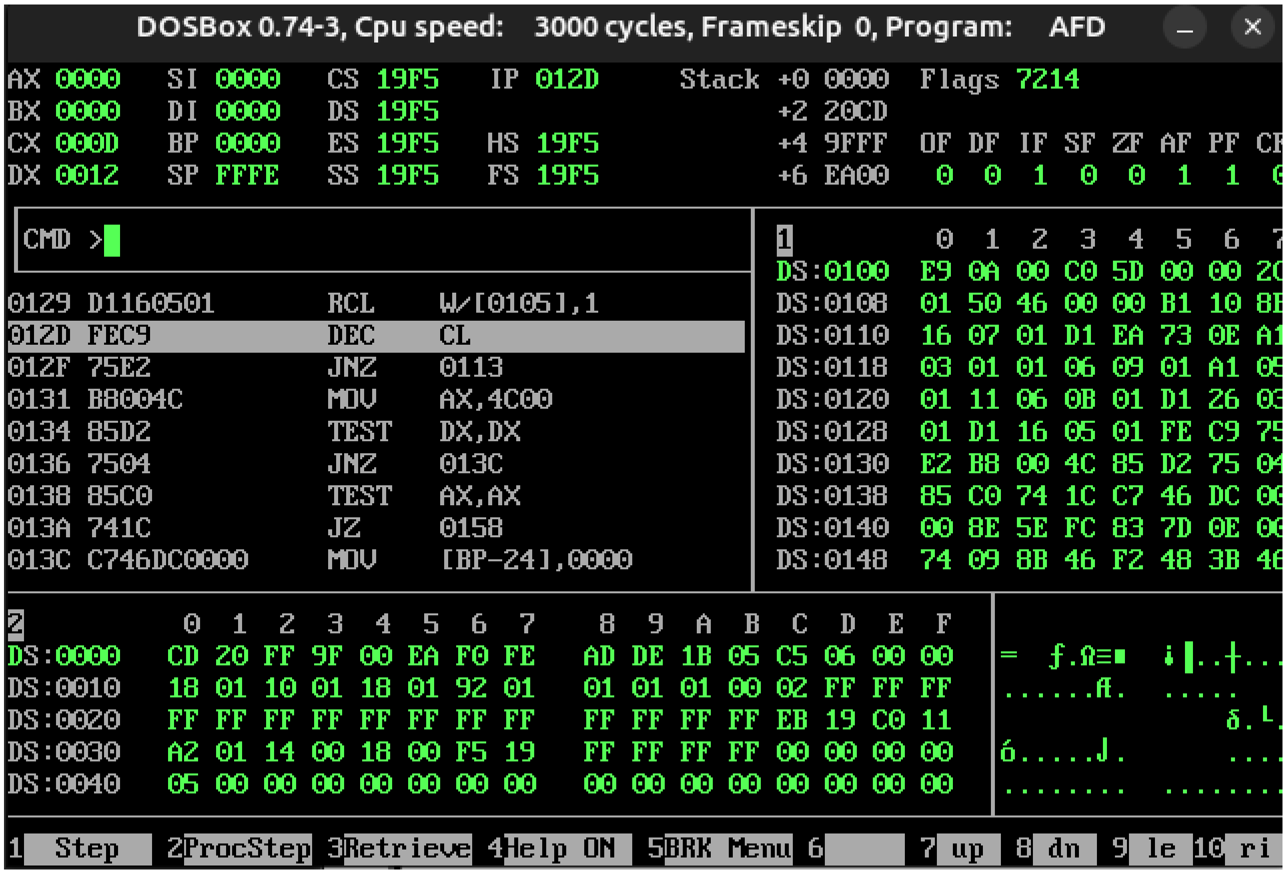

```
DOSBox 0.74-3, Cpu speed:   3000 cycles, Frameskip  0, Program:   AFD
AX 0000   SI 0000   CS 19F5   IP 0121      Stack +0 0000  Flags 7200
BX 0000   DI 0000   DS 19F5                      +2 20CD
CX 000E   BP 0000   ES 19F5   HS 19F5            +4 9FFF  OF DF IF SF ZF AF PF CF
DX 0025   SP FFFE   SS 19F5   FS 19F5            +6 EA00   0  0  1  0  0  0  0  0
 CMD >                                          1          0  1  2  3  4  5  6  7
                                        0000    DS:0100  E9 0A 00 70 17 00 00 2C
011E A10501          MOV    AX,[0105]           DS:0108  01 70 17 00 00 B1 10 8B
0121 11060B01        ADC    [010B],AX           DS:0110  16 07 01 D1 EA 73 0E A1
0125 D1260301        SHL    W/[0103],1          DS:0118  03 01 01 06 09 01 A1 05
```

moved **upper** **16 bits** of the **multiplicand** variable into the **AX** register


```
DOSBox 0.74-3, Cpu speed:   3000 cycles, Frameskip  0, Program:   AFD
AX 0000   SI 0000   CS 19F5   IP 0125      Stack +0 0000  Flags 7244
BX 0000   DI 0000   DS 19F5                      +2 20CD
CX 000E   BP 0000   ES 19F5   HS 19F5            +4 9FFF  OF DF IF SF ZF AF PF CF
DX 0025   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
                                        1770    DS:0100  E9 0A 00 70 17 00 00 2C
0121 11060B01        ADC    [010B],AX           DS:0108  01 70 17 00 00 B1 10 8B
0125 D1260301        SHL    W/[0103],1          DS:0110  16 07 01 D1 EA 73 0E A1
0129 D1160501        RCL    W/[0105],1          DS:0118  03 01 01 06 09 01 A1 05
```

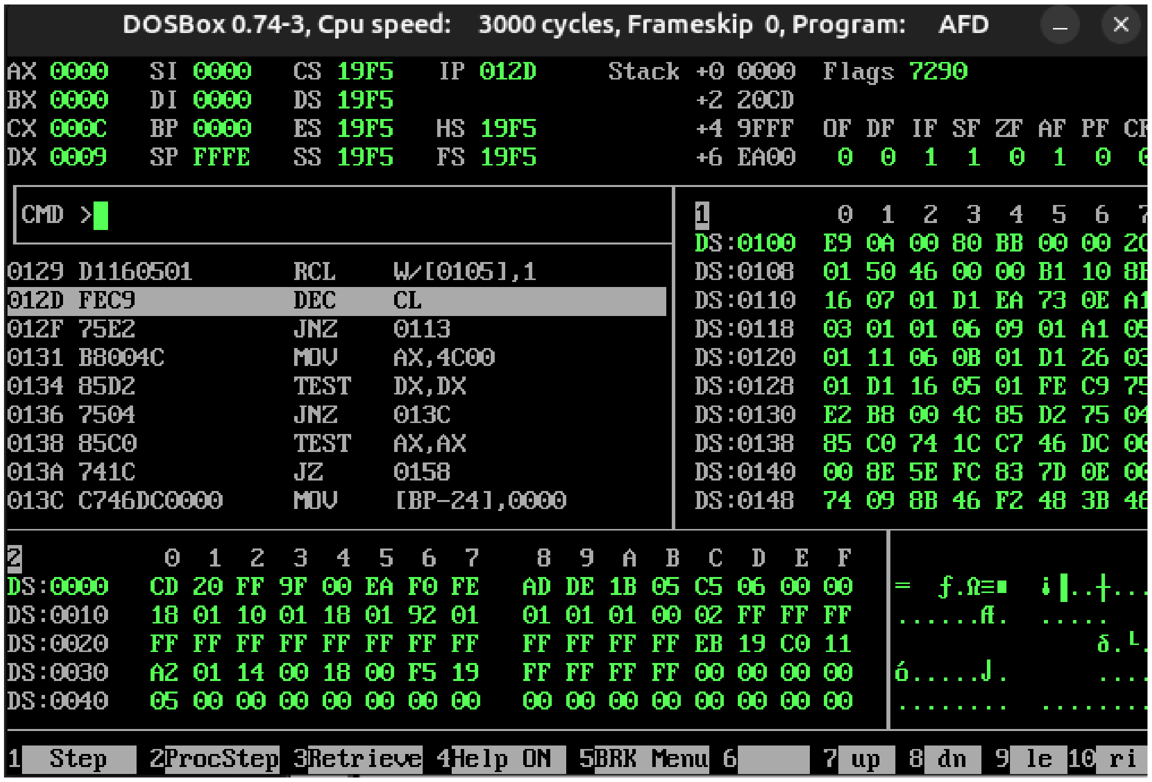
added more significant word.

Iteration 3:



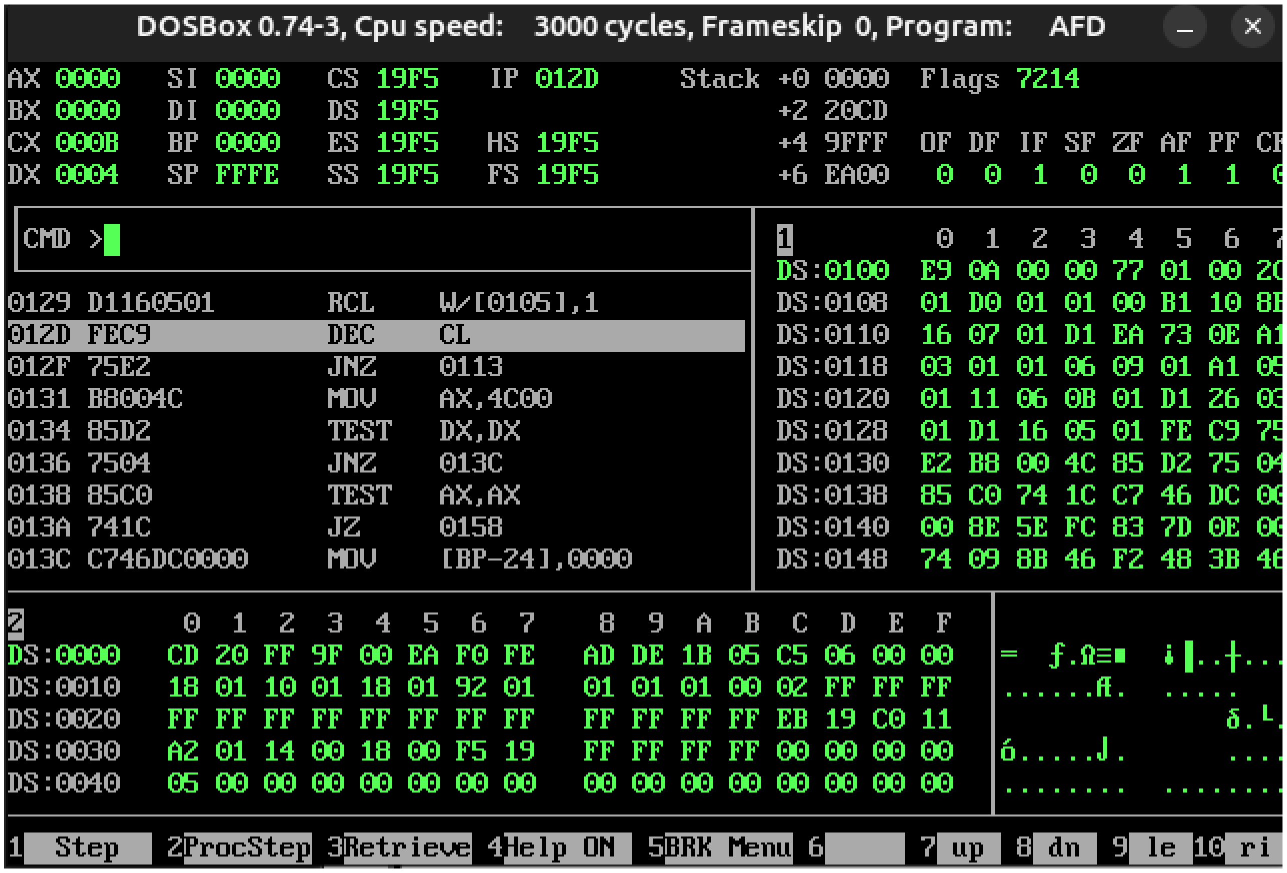
```
DOSBox 0.74-3, Cpu speed:   3000 cycles, Frameskip  0, Program:   AFD
AX 0000   SI 0000   CS 19F5   IP 012D      Stack +0 0000  Flags 7214
BX 0000   DI 0000   DS 19F5                      +2 20CD
CX 000D   BP 0000   ES 19F5   HS 19F5            +4 9FFF  OF DF IF SF ZF AF PF CF
DX 0012   SP FFFE   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:0100  E9 0A 00 C0 5D 00 00 20
0129 D1160501       RCL    W/[0105],1              DS:0108  01 50 46 00 00 B1 10 8B
012D FEC9           DEC    CL                      DS:0110  16 07 01 D1 EA 73 0E A1
012F 75E2           JNZ    0113                    DS:0118  03 01 01 06 09 01 A1 05
0131 B8004C         MOV    AX,4C00                 DS:0120  01 11 06 0B 01 D1 26 03
0134 85D2           TEST   DX,DX                   DS:0128  01 D1 16 05 01 FE C9 75
0136 7504           JNZ    013C                    DS:0130  E2 B8 00 4C 85 D2 75 04
0138 85C0           TEST   AX,AX                   DS:0138  85 C0 74 1C C7 46 DC 00
013A 741C           JZ     0158                    DS:0140  00 8E 5E FC 83 7D 0E 00
013C C746DC0000     MOV    [BP-24],0000            DS:0148  74 09 8B 46 F2 48 3B 46

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
```

Iteration 4:



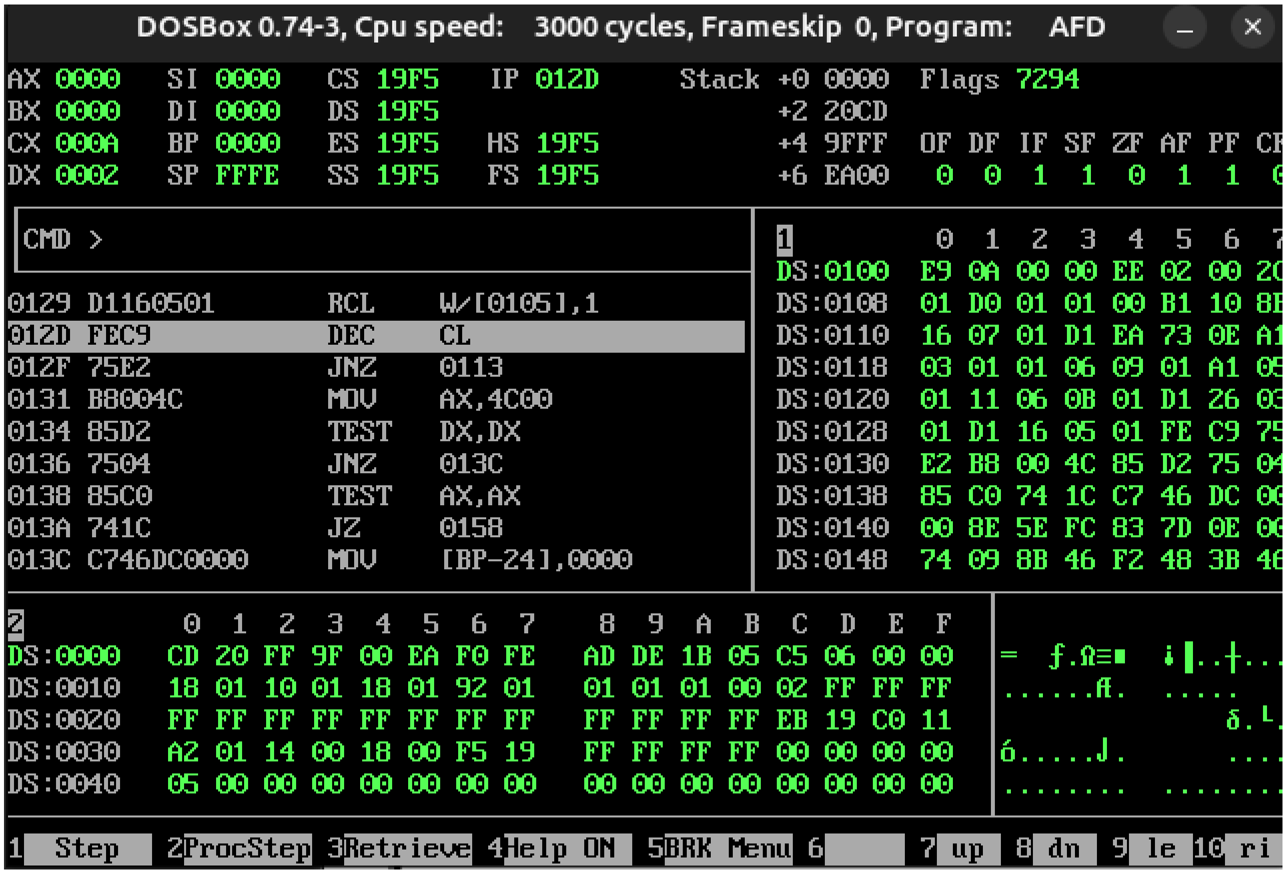
```
DOSBox 0.74-3, Cpu speed:   3000 cycles, Frameskip  0, Program:   AFD
AX 0000   SI 0000   CS 19F5   IP 012D      Stack +0 0000  Flags 7290
BX 0000   DI 0000   DS 19F5                      +2 20CD
CX 000C   BP 0000   ES 19F5   HS 19F5            +4 9FFF  OF DF IF SF ZF AF PF CF
DX 0009   SP FFFE   SS 19F5   FS 19F5            +6 EA00   0  0  1  1  0  1  0  0

 CMD >                                             1          0  1  2  3  4  5  6  7
                                                   DS:0100  E9 0A 00 80 BB 00 00 20
0129 D1160501       RCL    W/[0105],1              DS:0108  01 50 46 00 00 B1 10 8B
012D FEC9           DEC    CL                      DS:0110  16 07 01 D1 EA 73 0E A1
012F 75E2           JNZ    0113                    DS:0118  03 01 01 06 09 01 A1 05
0131 B8004C         MOV    AX,4C00                 DS:0120  01 11 06 0B 01 D1 26 03
0134 85D2           TEST   DX,DX                   DS:0128  01 D1 16 05 01 FE C9 75
0136 7504           JNZ    013C                    DS:0130  E2 B8 00 4C 85 D2 75 04
0138 85C0           TEST   AX,AX                   DS:0138  85 C0 74 1C C7 46 DC 00
013A 741C           JZ     0158                    DS:0140  00 8E 5E FC 83 7D 0E 00
013C C746DC0000     MOV    [BP-24],0000            DS:0148  74 09 8B 46 F2 48 3B 46

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
```

Iteration 5:



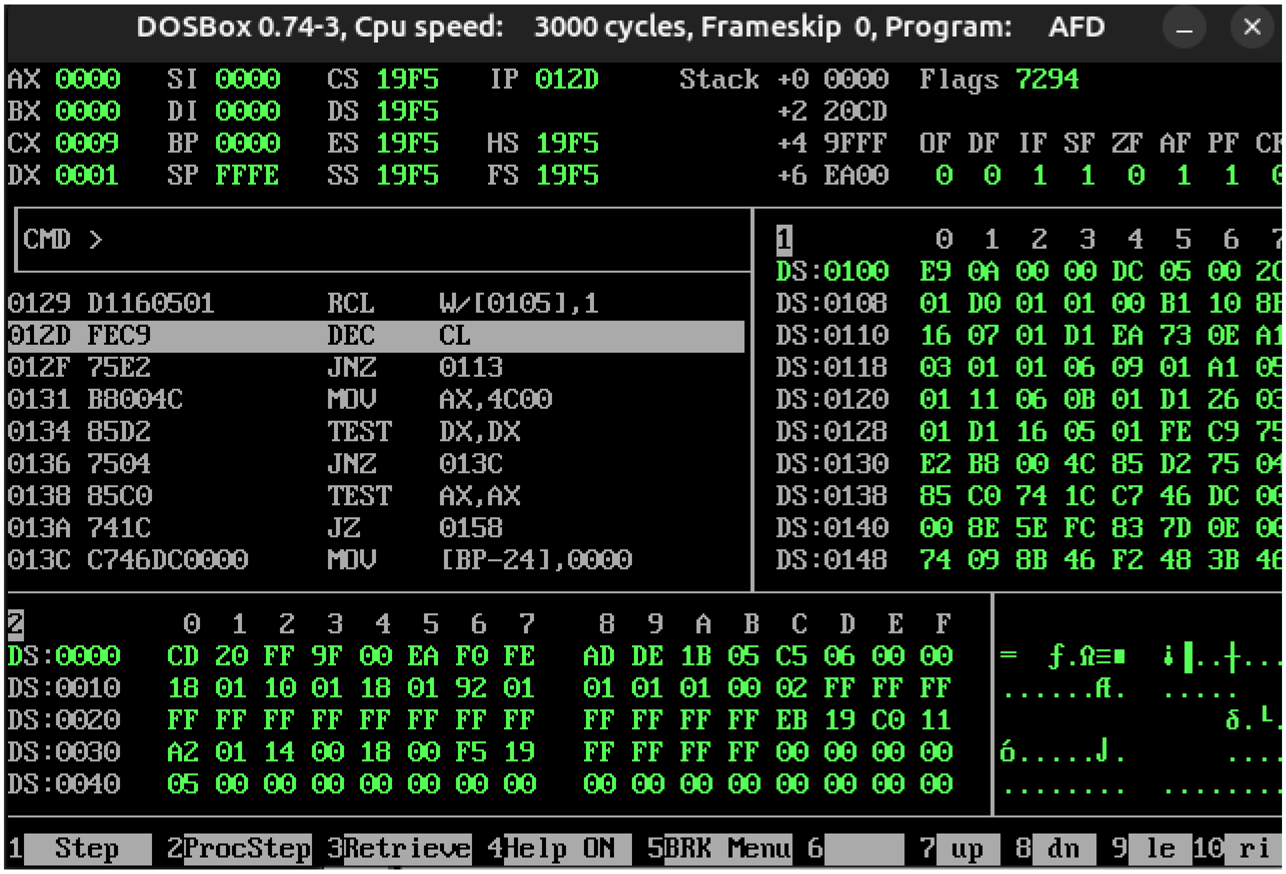
```
DOSBox 0.74-3, Cpu speed:   3000 cycles, Frameskip  0, Program:   AFD
AX 0000   SI 0000   CS 19F5   IP 012D     Stack +0 0000  Flags 7214
BX 0000   DI 0000   DS 19F5                     +2 20CD
CX 000B   BP 0000   ES 19F5   HS 19F5          +4 9FFF  OF DF IF SF ZF AF PF CF
DX 0004   SP FFFE   SS 19F5   FS 19F5          +6 EA00   0  0  1  0  0  1  1  0

 CMD >                                          1        0  1  2  3  4  5  6  7
                                                DS:0100  E9 0A 00 00 77 01 00 20
0129 D1160501        RCL     W/[0105],1         DS:0108  01 D0 01 01 00 B1 10 8B
012D FEC9            DEC     CL                 DS:0110  16 07 01 D1 EA 73 0E A1
012F 75E2            JNZ     0113               DS:0118  03 01 01 06 09 01 A1 05
0131 B8004C          MOV     AX,4C00            DS:0120  01 11 06 0B 01 D1 26 03
0134 85D2            TEST    DX,DX              DS:0128  01 D1 16 05 01 FE C9 75
0136 7504            JNZ     013C               DS:0130  E2 B8 00 4C 85 D2 75 04
0138 85C0            TEST    AX,AX              DS:0138  85 C0 74 1C C7 46 DC 00
013A 741C            JZ      0158               DS:0140  00 8E 5E FC 83 7D 0E 00
013C C746DC0000      MOV     [BP-24],0000       DS:0148  74 09 8B 46 F2 48 3B 46

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
```

Iteration 6:



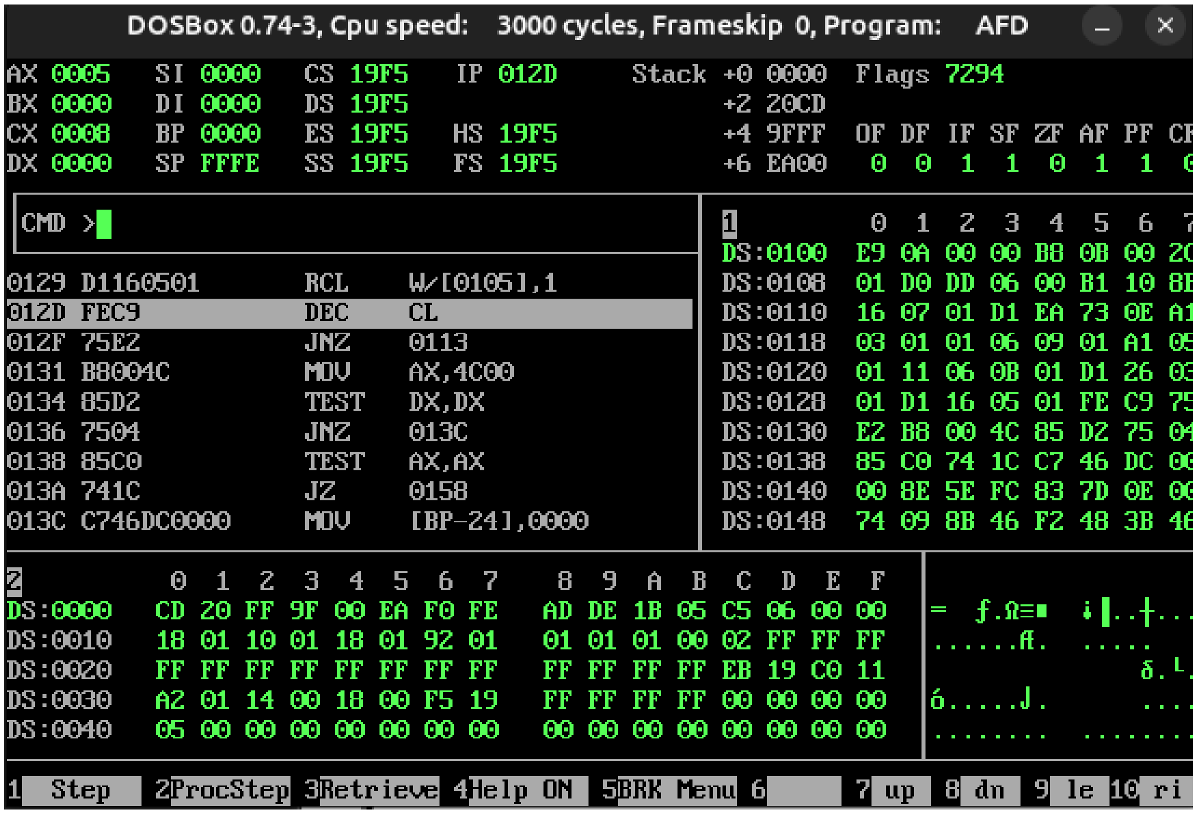
```
DOSBox 0.74-3, Cpu speed:   3000 cycles, Frameskip  0, Program:   AFD
AX 0000   SI 0000   CS 19F5   IP 012D     Stack +0 0000  Flags 7294
BX 0000   DI 0000   DS 19F5                     +2 20CD
CX 000A   BP 0000   ES 19F5   HS 19F5          +4 9FFF  OF DF IF SF ZF AF PF CF
DX 0002   SP FFFE   SS 19F5   FS 19F5          +6 EA00   0  0  1  1  0  1  1  0

 CMD >                                          1        0  1  2  3  4  5  6  7
                                                DS:0100  E9 0A 00 00 EE 02 00 20
0129 D1160501        RCL     W/[0105],1         DS:0108  01 D0 01 01 00 B1 10 8B
012D FEC9            DEC     CL                 DS:0110  16 07 01 D1 EA 73 0E A1
012F 75E2            JNZ     0113               DS:0118  03 01 01 06 09 01 A1 05
0131 B8004C          MOV     AX,4C00            DS:0120  01 11 06 0B 01 D1 26 03
0134 85D2            TEST    DX,DX              DS:0128  01 D1 16 05 01 FE C9 75
0136 7504            JNZ     013C               DS:0130  E2 B8 00 4C 85 D2 75 04
0138 85C0            TEST    AX,AX              DS:0138  85 C0 74 1C C7 46 DC 00
013A 741C            JZ      0158               DS:0140  00 8E 5E FC 83 7D 0E 00
013C C746DC0000      MOV     [BP-24],0000       DS:0148  74 09 8B 46 F2 48 3B 46

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
```

Iteration 7:



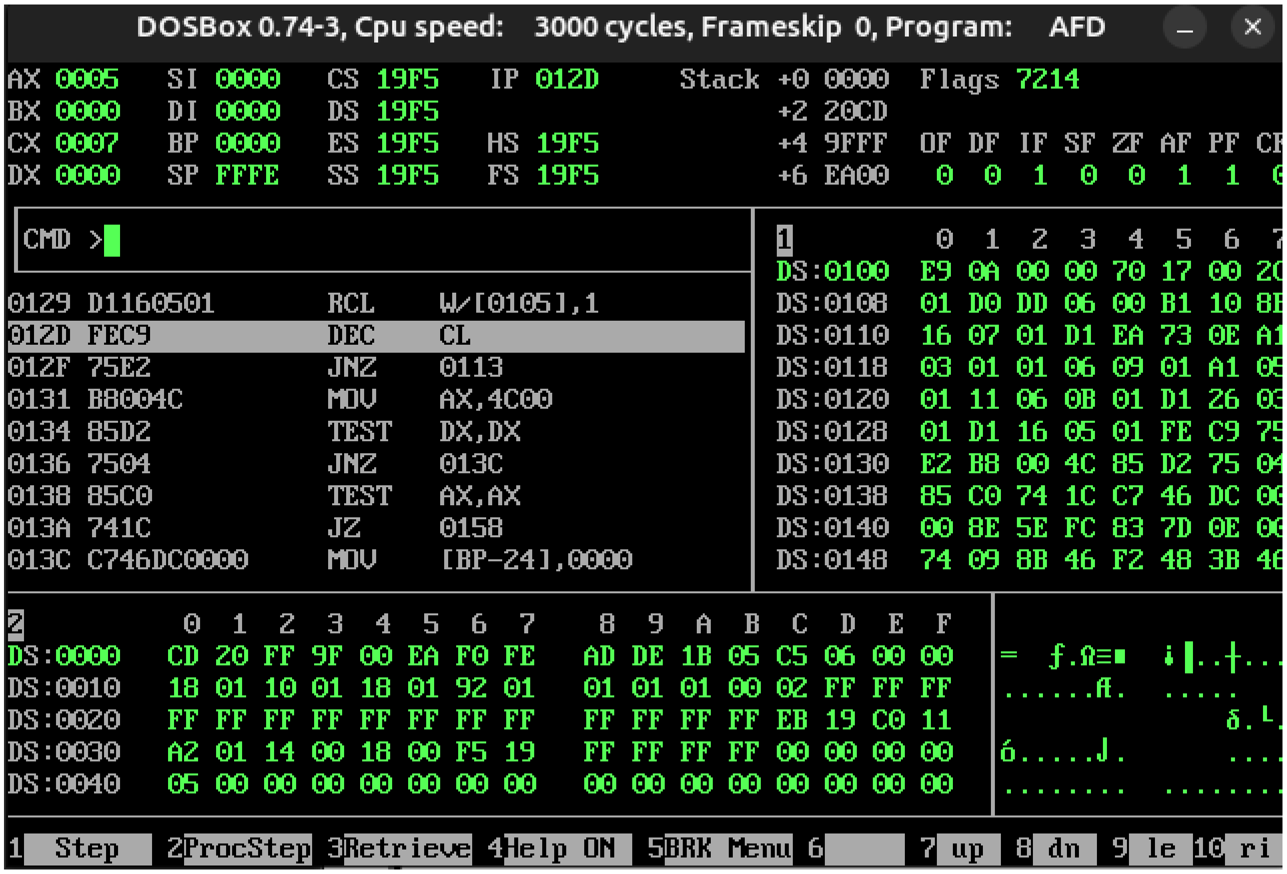
```
DOSBox 0.74-3, Cpu speed:   3000 cycles, Frameskip  0, Program:   AFD

AX 0000   SI 0000   CS 19F5   IP 012D      Stack +0 0000  Flags 7294
BX 0000   DI 0000   DS 19F5                      +2 20CD
CX 0009   BP 0000   ES 19F5   HS 19F5            +4 9FFF  OF DF IF SF ZF AF PF CF
DX 0001   SP FFFE   SS 19F5   FS 19F5            +6 EA00   0  0  1  1  0  1  1  0

 CMD >                                           1     0  1  2  3  4  5  6  7
                                                 DS:0100 E9 0A 00 00 DC 05 00 20
0129 D1160501          RCL     W/[0105],1        DS:0108 01 D0 01 01 00 B1 10 8B
012D FEC9              DEC     CL                DS:0110 16 07 01 D1 EA 73 0E A1
012F 75E2              JNZ     0113              DS:0118 03 01 01 06 09 01 A1 05
0131 B8004C            MOV     AX,4C00           DS:0120 01 11 06 0B 01 D1 26 03
0134 85D2              TEST    DX,DX             DS:0128 01 D1 16 05 01 FE C9 75
0136 7504              JNZ     013C              DS:0130 E2 B8 00 4C 85 D2 75 04
0138 85C0              TEST    AX,AX             DS:0138 85 C0 74 1C C7 46 DC 00
013A 741C              JZ      0158              DS:0140 00 8E 5E FC 83 7D 0E 00
013C C746DC0000        MOV     [BP-24],0000      DS:0148 74 09 8B 46 F2 48 3B 46

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
```

Iteration 8:



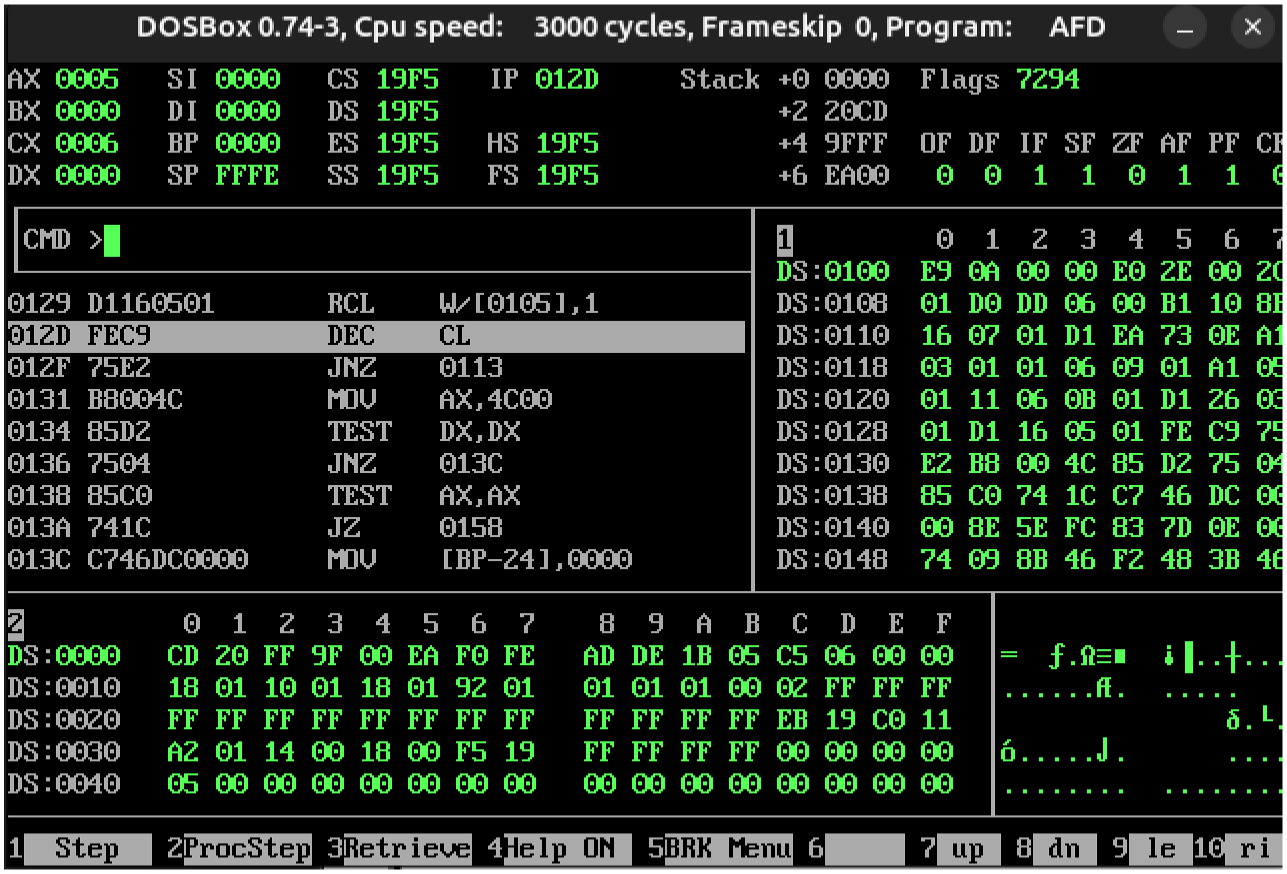
```
DOSBox 0.74-3, Cpu speed:   3000 cycles, Frameskip  0, Program:   AFD

AX 0005   SI 0000   CS 19F5   IP 012D      Stack +0 0000  Flags 7294
BX 0000   DI 0000   DS 19F5                      +2 20CD
CX 0008   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  1  0  1  1  0

 CMD >                                           1     0  1  2  3  4  5  6  7
                                                 DS:0100 E9 0A 00 00 B8 0B 00 20
0129 D1160501          RCL     W/[0105],1        DS:0108 01 D0 DD 06 00 B1 10 8B
012D FEC9              DEC     CL                DS:0110 16 07 01 D1 EA 73 0E A1
012F 75E2              JNZ     0113              DS:0118 03 01 01 06 09 01 A1 05
0131 B8004C            MOV     AX,4C00           DS:0120 01 11 06 0B 01 D1 26 03
0134 85D2              TEST    DX,DX             DS:0128 01 D1 16 05 01 FE C9 75
0136 7504              JNZ     013C              DS:0130 E2 B8 00 4C 85 D2 75 04
0138 85C0              TEST    AX,AX             DS:0138 85 C0 74 1C C7 46 DC 00
013A 741C              JZ      0158              DS:0140 00 8E 5E FC 83 7D 0E 00
013C C746DC0000        MOV     [BP-24],0000      DS:0148 74 09 8B 46 F2 48 3B 46

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
```

Iteration 9:



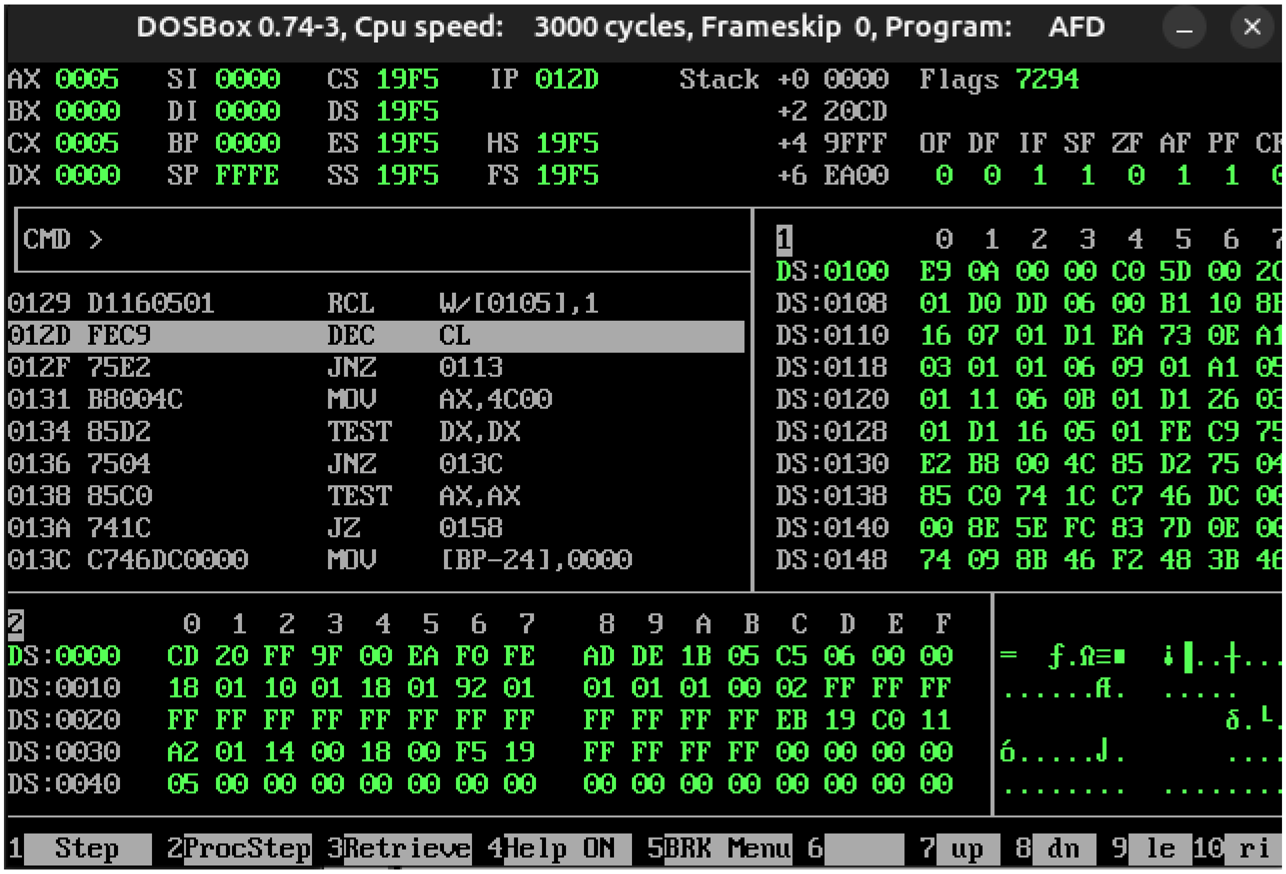
```
DOSBox 0.74-3, Cpu speed:   3000 cycles, Frameskip  0, Program:   AFD
AX 0005   SI 0000   CS 19F5   IP 012D      Stack +0 0000  Flags 7214
BX 0000   DI 0000   DS 19F5                      +2 20CD
CX 0007   BP 0000   ES 19F5   HS 19F5            +4 9FFF   OF DF IF SF ZF AF PF CF
DX 0000   SP FFFE   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:0100  E9 0A 00 00 70 17 00 20
0129 D1160501       RCL     W/[0105],1        DS:0108  01 D0 DD 06 00 B1 10 8B
012D FEC9           DEC     CL                DS:0110  16 07 01 D1 EA 73 0E A1
012F 75E2           JNZ     0113              DS:0118  03 01 01 06 09 01 A1 05
0131 B8004C         MOV     AX,4C00           DS:0120  01 11 06 0B 01 D1 26 03
0134 85D2           TEST    DX,DX             DS:0128  01 D1 16 05 01 FE C9 75
0136 7504           JNZ     013C              DS:0130  E2 B8 00 4C 85 D2 75 04
0138 85C0           TEST    AX,AX             DS:0138  85 C0 74 1C C7 46 DC 00
013A 741C           JZ      0158              DS:0140  00 8E 5E FC 83 7D 0E 00
013C C746DC0000     MOV     [BP-24],0000      DS:0148  74 09 8B 46 F2 48 3B 46

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
```

Iteration 10:



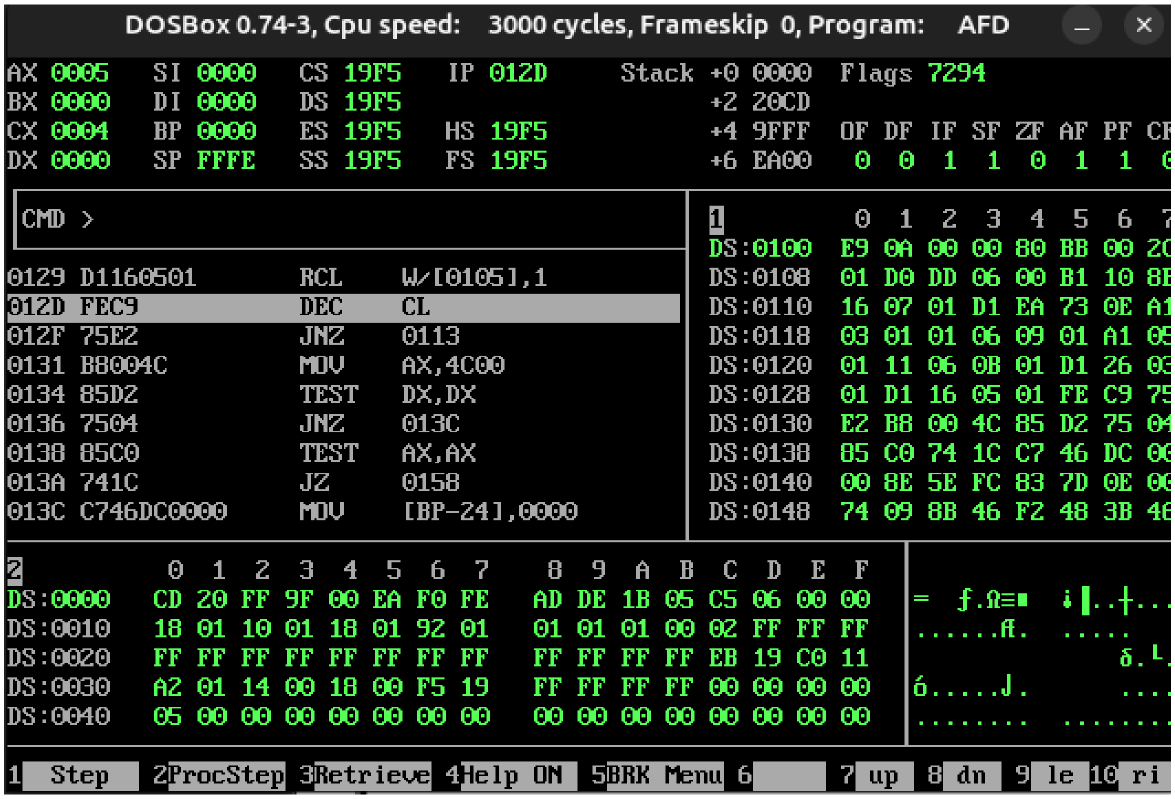
```
DOSBox 0.74-3, Cpu speed:   3000 cycles, Frameskip  0, Program:   AFD
AX 0005   SI 0000   CS 19F5   IP 012D      Stack +0 0000  Flags 7294
BX 0000   DI 0000   DS 19F5                      +2 20CD
CX 0006   BP 0000   ES 19F5   HS 19F5            +4 9FFF   OF DF IF SF ZF AF PF CF
DX 0000   SP FFFE   SS 19F5   FS 19F5            +6 EA00    0  0  1  1  0  1  1  0

 CMD >                                        1         0  1  2  3  4  5  6  7
                                              DS:0100  E9 0A 00 00 E0 2E 00 20
0129 D1160501       RCL     W/[0105],1        DS:0108  01 D0 DD 06 00 B1 10 8B
012D FEC9           DEC     CL                DS:0110  16 07 01 D1 EA 73 0E A1
012F 75E2           JNZ     0113              DS:0118  03 01 01 06 09 01 A1 05
0131 B8004C         MOV     AX,4C00           DS:0120  01 11 06 0B 01 D1 26 03
0134 85D2           TEST    DX,DX             DS:0128  01 D1 16 05 01 FE C9 75
0136 7504           JNZ     013C              DS:0130  E2 B8 00 4C 85 D2 75 04
0138 85C0           TEST    AX,AX             DS:0138  85 C0 74 1C C7 46 DC 00
013A 741C           JZ      0158              DS:0140  00 8E 5E FC 83 7D 0E 00
013C C746DC0000     MOV     [BP-24],0000      DS:0148  74 09 8B 46 F2 48 3B 46

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
```

Iteration 11:



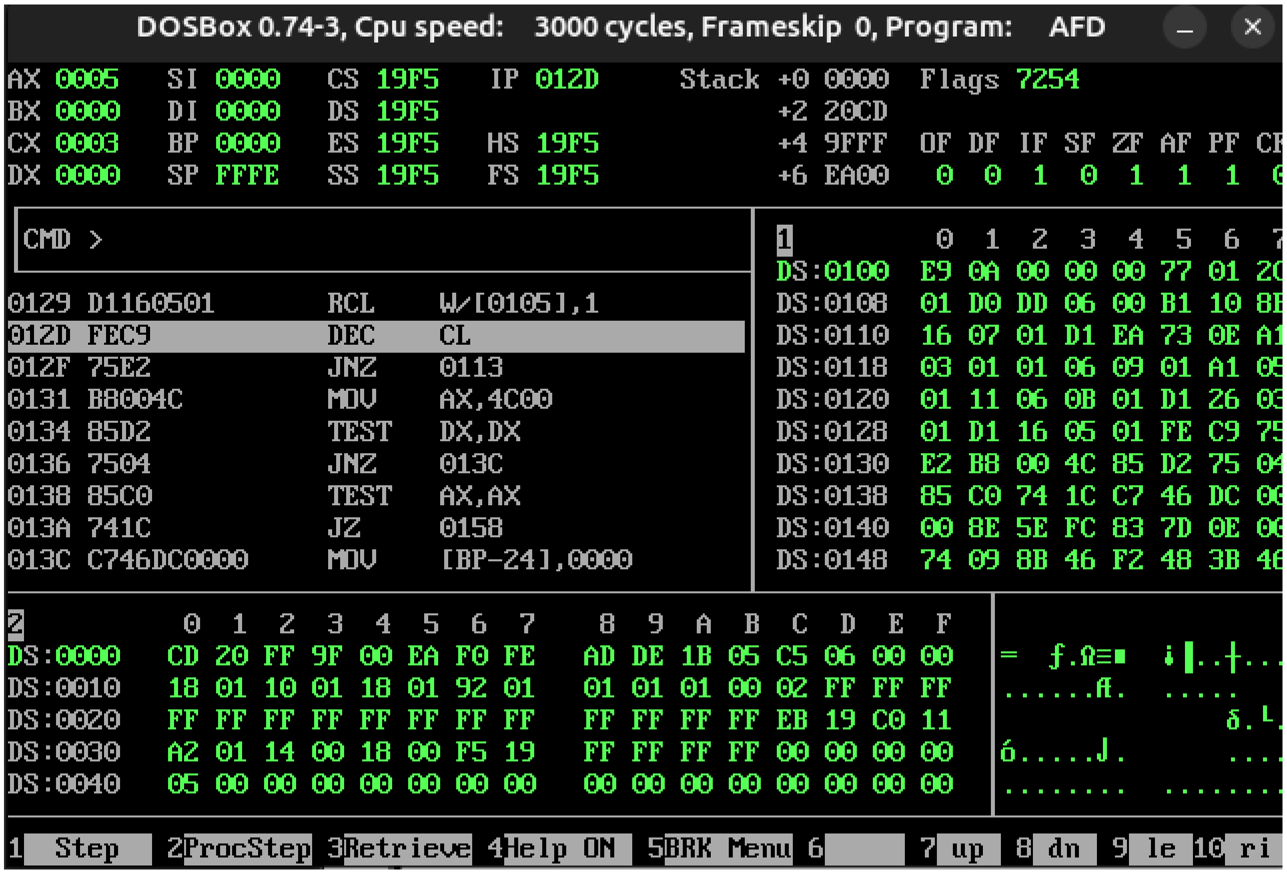
```
DOSBox 0.74-3, Cpu speed:   3000 cycles, Frameskip  0, Program:   AFD

AX 0005   SI 0000   CS 19F5   IP 012D     Stack +0 0000  Flags 7294
BX 0000   DI 0000   DS 19F5                     +2 20CD
CX 0005   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  1  0  1  1  0

 CMD >                                           1     0  1  2  3  4  5  6  7
                                                 DS:0100  E9 0A 00 00 C0 5D 00 20
0129 D1160501        RCL     W/[0105],1          DS:0108  01 D0 DD 06 00 B1 10 8B
012D FEC9            DEC     CL                  DS:0110  16 07 01 D1 EA 73 0E A1
012F 75E2            JNZ     0113                DS:0118  03 01 01 06 09 01 A1 05
0131 B8004C          MOV     AX,4C00             DS:0120  01 11 06 0B 01 D1 26 03
0134 85D2            TEST    DX,DX               DS:0128  01 D1 16 05 01 FE C9 75
0136 7504            JNZ     013C                DS:0130  E2 B8 00 4C 85 D2 75 04
0138 85C0            TEST    AX,AX               DS:0138  85 C0 74 1C C7 46 DC 00
013A 741C            JZ      0158                DS:0140  00 8E 5E FC 83 7D 0E 00
013C C746DC0000      MOV     [BP-24],0000        DS:0148  74 09 8B 46 F2 48 3B 46

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
```

Iteration 12:



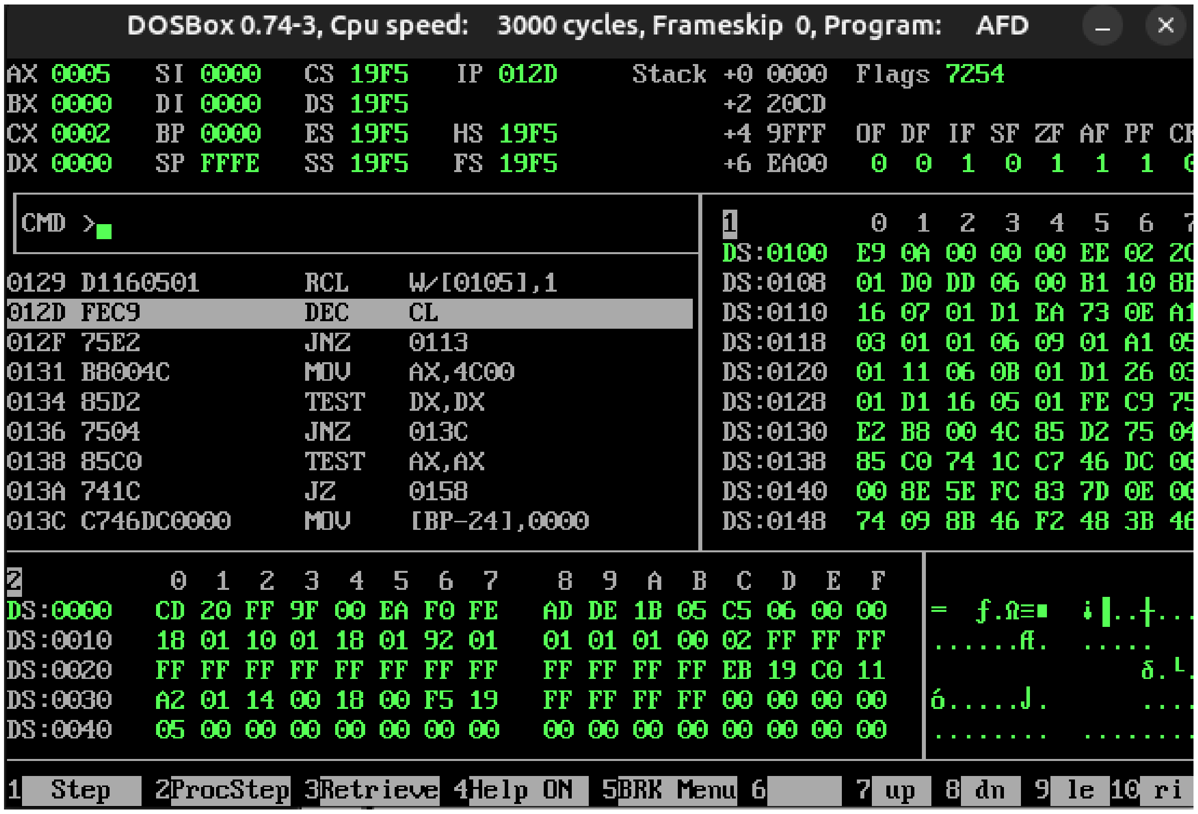
```
DOSBox 0.74-3, Cpu speed:   3000 cycles, Frameskip  0, Program:   AFD

AX 0005   SI 0000   CS 19F5   IP 012D     Stack +0 0000  Flags 7294
BX 0000   DI 0000   DS 19F5                     +2 20CD
CX 0004   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  1  0  1  1  0

 CMD >                                           1     0  1  2  3  4  5  6  7
                                                 DS:0100  E9 0A 00 00 80 BB 00 20
0129 D1160501        RCL     W/[0105],1          DS:0108  01 D0 DD 06 00 B1 10 8B
012D FEC9            DEC     CL                  DS:0110  16 07 01 D1 EA 73 0E A1
012F 75E2            JNZ     0113                DS:0118  03 01 01 06 09 01 A1 05
0131 B8004C          MOV     AX,4C00             DS:0120  01 11 06 0B 01 D1 26 03
0134 85D2            TEST    DX,DX               DS:0128  01 D1 16 05 01 FE C9 75
0136 7504            JNZ     013C                DS:0130  E2 B8 00 4C 85 D2 75 04
0138 85C0            TEST    AX,AX               DS:0138  85 C0 74 1C C7 46 DC 00
013A 741C            JZ      0158                DS:0140  00 8E 5E FC 83 7D 0E 00
013C C746DC0000      MOV     [BP-24],0000        DS:0148  74 09 8B 46 F2 48 3B 46

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
```

Iteration 13:



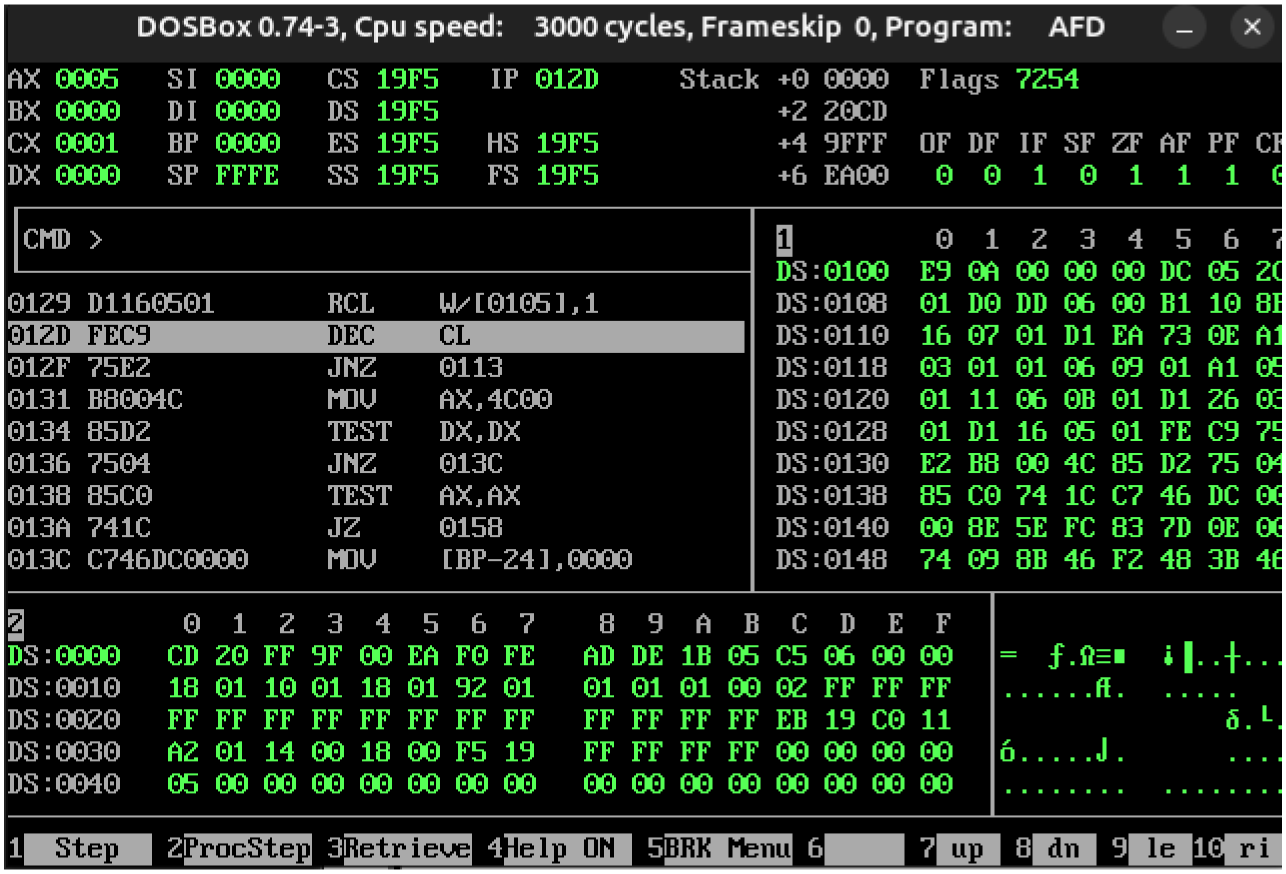
```
DOSBox 0.74-3, Cpu speed:   3000 cycles, Frameskip  0, Program:   AFD
AX 0005   SI 0000   CS 19F5   IP 012D     Stack +0 0000  Flags 7254
BX 0000   DI 0000   DS 19F5                     +2 20CD
CX 0003   BP 0000   ES 19F5   HS 19F5           +4 9FFF  OF DF IF SF ZF AF PF CF
DX 0000   SP FFFE   SS 19F5   FS 19F5           +6 EA00   0  0  1  0  1  1  1  0

 CMD >                                        1        0  1  2  3  4  5  6  7
                                              DS:0100  E9 0A 00 00 00 77 01 20
0129 D1160501         RCL     W/[0105],1      DS:0108  01 D0 DD 06 00 B1 10 8E
012D FEC9             DEC     CL              DS:0110  16 07 01 D1 EA 73 0E A1
012F 75E2             JNZ     0113            DS:0118  03 01 01 06 09 01 A1 05
0131 B8004C           MOV     AX,4C00         DS:0120  01 11 06 0B 01 D1 26 03
0134 85D2             TEST    DX,DX           DS:0128  01 D1 16 05 01 FE C9 75
0136 7504             JNZ     013C            DS:0130  E2 B8 00 4C 85 D2 75 04
0138 85C0             TEST    AX,AX           DS:0138  85 C0 74 1C C7 46 DC 00
013A 741C             JZ      0158            DS:0140  00 8E 5E FC 83 7D 0E 00
013C C746DC0000       MOV     [BP-24],0000    DS:0148  74 09 8B 46 F2 48 3B 46

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
```

Iteration 14



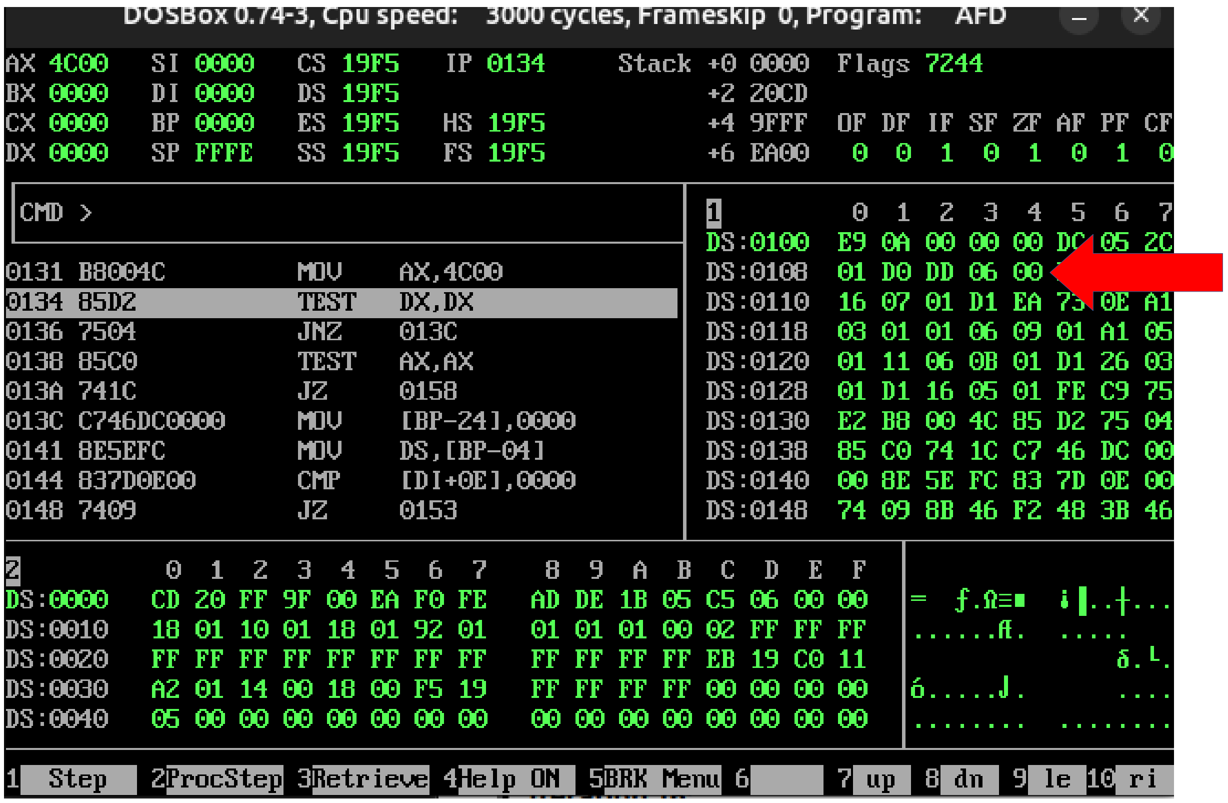
```
DOSBox 0.74-3, Cpu speed:   3000 cycles, Frameskip  0, Program:   AFD
AX 0005   SI 0000   CS 19F5   IP 012D     Stack +0 0000  Flags 7254
BX 0000   DI 0000   DS 19F5                     +2 20CD
CX 0002   BP 0000   ES 19F5   HS 19F5           +4 9FFF  OF DF IF SF ZF AF PF CF
DX 0000   SP FFFE   SS 19F5   FS 19F5           +6 EA00   0  0  1  0  1  1  1  0

 CMD >                                        1        0  1  2  3  4  5  6  7
                                              DS:0100  E9 0A 00 00 00 EE 02 20
0129 D1160501         RCL     W/[0105],1      DS:0108  01 D0 DD 06 00 B1 10 8E
012D FEC9             DEC     CL              DS:0110  16 07 01 D1 EA 73 0E A1
012F 75E2             JNZ     0113            DS:0118  03 01 01 06 09 01 A1 05
0131 B8004C           MOV     AX,4C00         DS:0120  01 11 06 0B 01 D1 26 03
0134 85D2             TEST    DX,DX           DS:0128  01 D1 16 05 01 FE C9 75
0136 7504             JNZ     013C            DS:0130  E2 B8 00 4C 85 D2 75 04
0138 85C0             TEST    AX,AX           DS:0138  85 C0 74 1C C7 46 DC 00
013A 741C             JZ      0158            DS:0140  00 8E 5E FC 83 7D 0E 00
013C C746DC0000       MOV     [BP-24],0000    DS:0148  74 09 8B 46 F2 48 3B 46

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
```

Iteration 15:

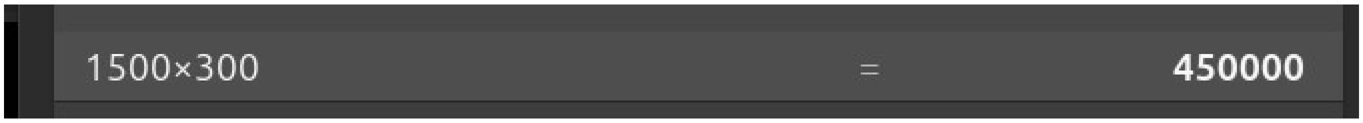


```
DOSBox 0.74-3, Cpu speed:   3000 cycles, Frameskip  0, Program:    AFD
AX 0005   SI 0000   CS 19F5   IP 012D     Stack +0 0000  Flags 7254
BX 0000   DI 0000   DS 19F5                     +2 20CD
CX 0001   BP 0000   ES 19F5   HS 19F5           +4 9FFF  OF DF IF SF ZF AF PF CF
DX 0000   SP FFFE   SS 19F5   FS 19F5           +6 EA00   0  0  1  0  1  1  1  0

 CMD >                                             1       0  1  2  3  4  5  6  7
                                                   DS:0100  E9 0A 00 00 00 DC 05 20
0129 D1160501       RCL    W/[0105],1              DS:0108  01 D0 DD 06 00 B1 10 8B
012D FEC9           DEC    CL                      DS:0110  16 07 01 D1 EA 73 0E A1
012F 75E2           JNZ    0113                    DS:0118  03 01 01 06 09 01 A1 05
0131 B8004C         MOV    AX,4C00                 DS:0120  01 11 06 0B 01 D1 26 03
0134 85D2           TEST   DX,DX                   DS:0128  01 D1 16 05 01 FE C9 75
0136 7504           JNZ    013C                    DS:0130  E2 B8 00 4C 85 D2 75 04
0138 85C0           TEST   AX,AX                   DS:0138  85 C0 74 1C C7 46 DC 00
013A 741C           JZ     0158                    DS:0140  00 8E 5E FC 83 7D 0E 00
013C C746DC0000     MOV    [BP-24],0000            DS:0148  74 09 8B 46 F2 48 3B 46

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
```

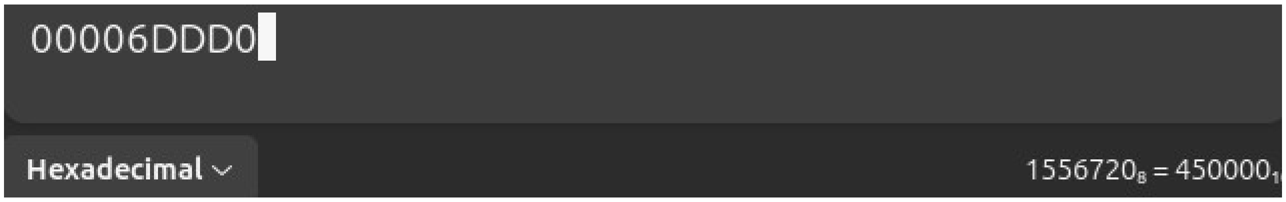
Iteration 16:



```
DOSBox 0.74-3, Cpu speed:   3000 cycles, Frameskip  0, Program:    AFD
AX 4C00   SI 0000   CS 19F5   IP 0134     Stack +0 0000  Flags 7244
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  1  0  1  0

 CMD >                                             1       0  1  2  3  4  5  6  7
                                                   DS:0100  E9 0A 00 00 00 DC 05 20
0131 B8004C         MOV    AX,4C00                 DS:0108  01 D0 DD 06 00
0134 85D2           TEST   DX,DX                   DS:0110  16 07 01 D1 EA 73 0E A1
0136 7504           JNZ    013C                    DS:0118  03 01 01 06 09 01 A1 05
0138 85C0           TEST   AX,AX                   DS:0120  01 11 06 0B 01 D1 26 03
013A 741C           JZ     0158                    DS:0128  01 D1 16 05 01 FE C9 75
013C C746DC0000     MOV    [BP-24],0000            DS:0130  E2 B8 00 4C 85 D2 75 04
0141 8E5EFC         MOV    DS,[BP-04]              DS:0138  85 C0 74 1C C7 46 DC 00
0144 837D0E00       CMP    [DI+0E],0000            DS:0140  00 8E 5E FC 83 7D 0E 00
0148 7409           JZ     0153                    DS:0148  74 09 8B 46 F2 48 3B 46

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
```



```
1500×300                                   =                                   450000
```



```
00006DDD0
```

$1556720_8 = 450000_{1}$