Microprocessors and Assembly Language - Coding Questions

١. Write an assembly language program that adds two 8-bit numbers stored in registers and stores the result in a third register. Ensure proper usage of flags and comments. (3 marks)

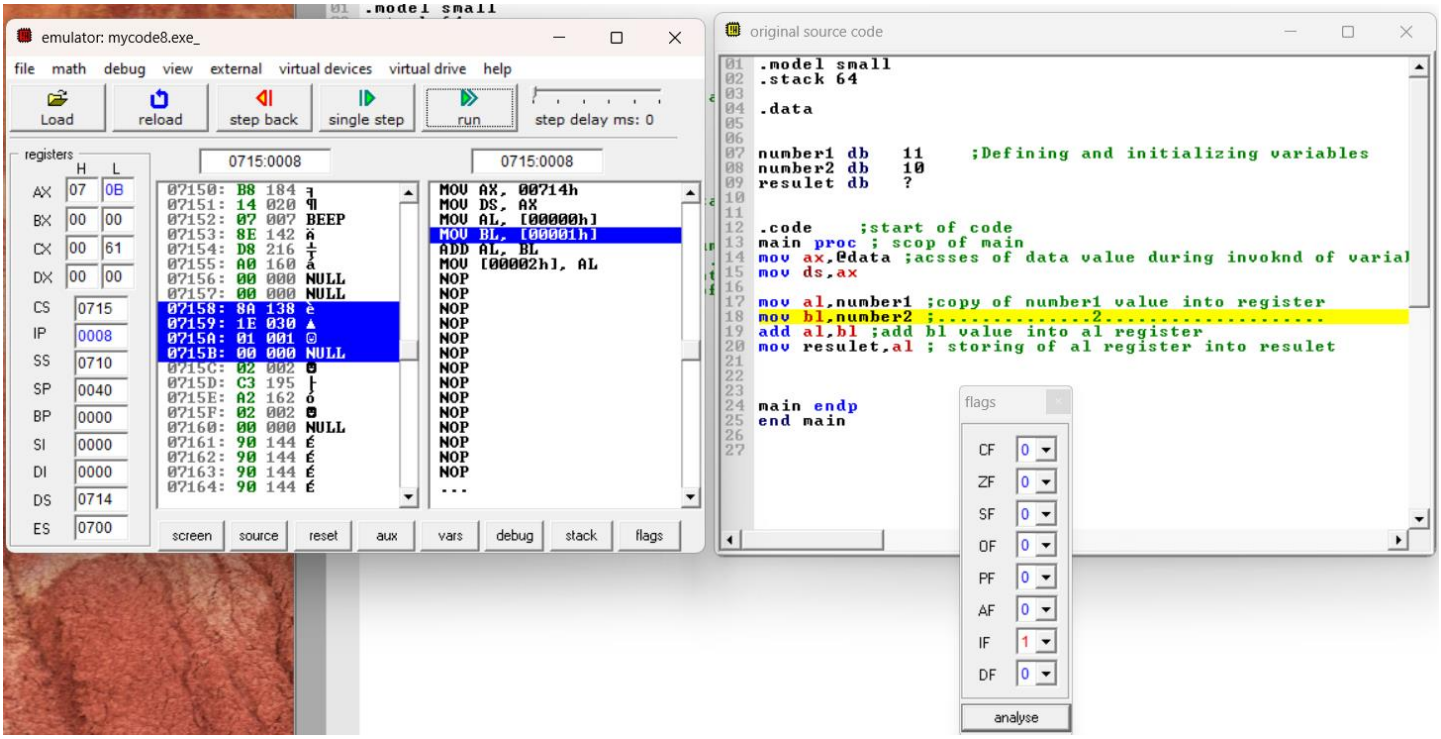
٢. Write a program in assembly language that moves data from memory to registers and then performs a bitwise AND operation. Explain how the operation affects the processor flags. (3 marks)

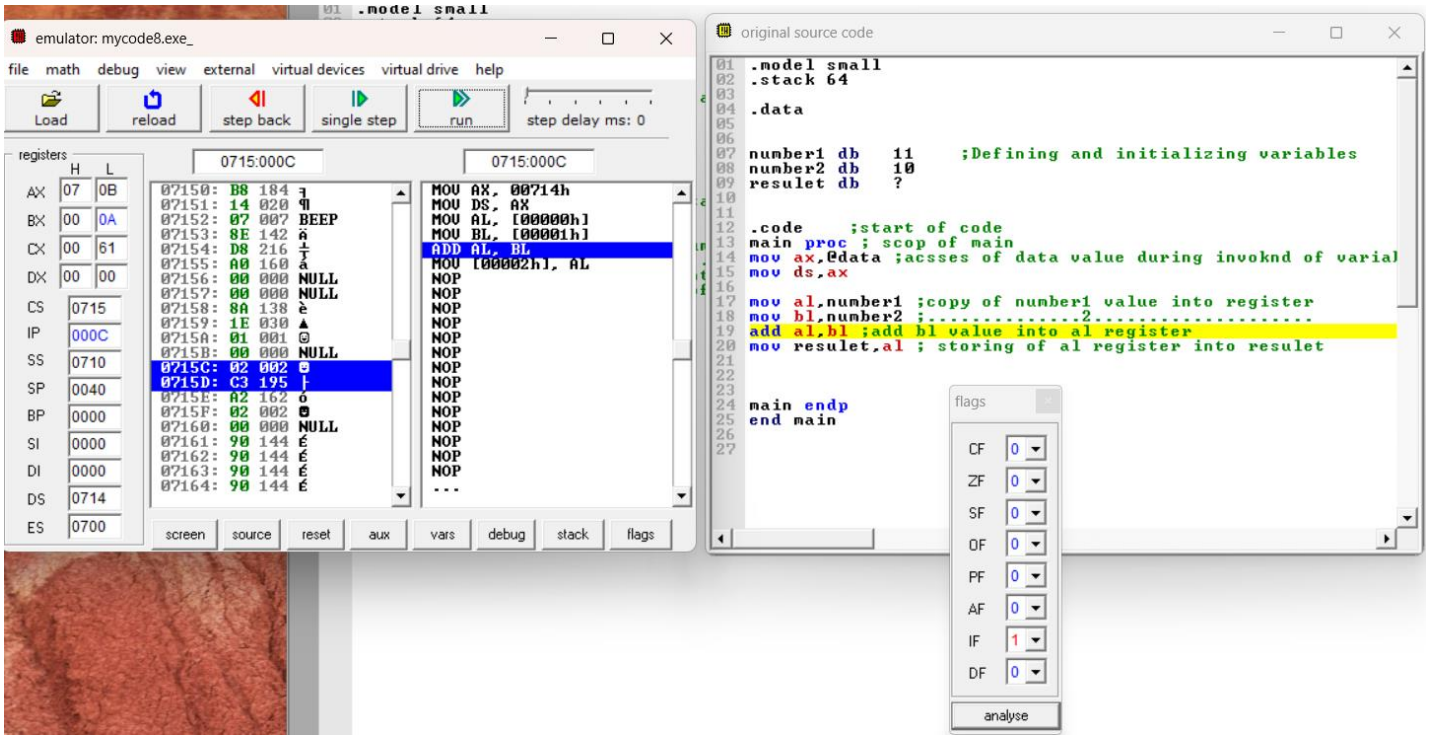
السؤال الأول 1

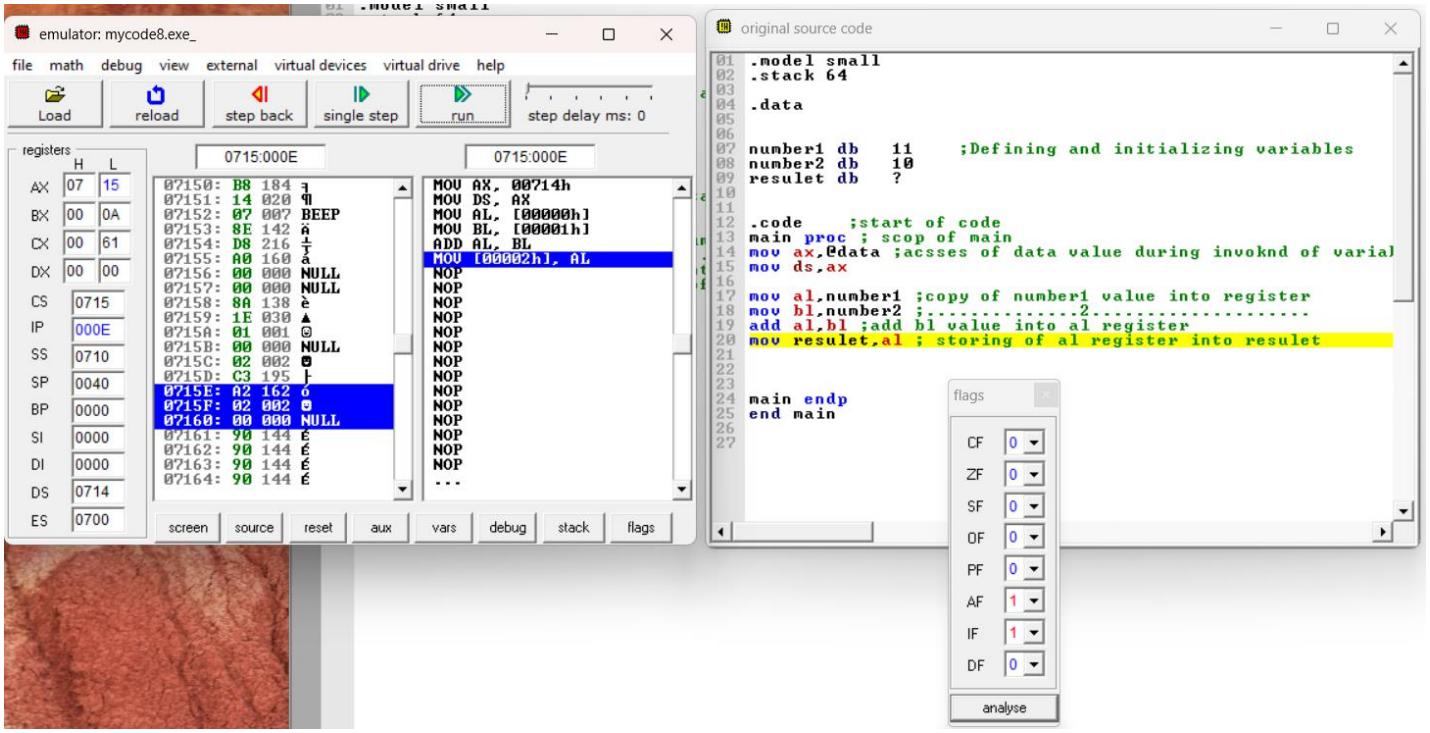
-استدعاء قيمة المتغير number1 من جزء data.وتخزينها في المسجل al

-استدعاء قيمة المتغير number2 من جزء data.وتخزينها في المسجل bl

- تفيذ عملية add ولايوجد تأثيرات على Carry Flag قيمته صفر





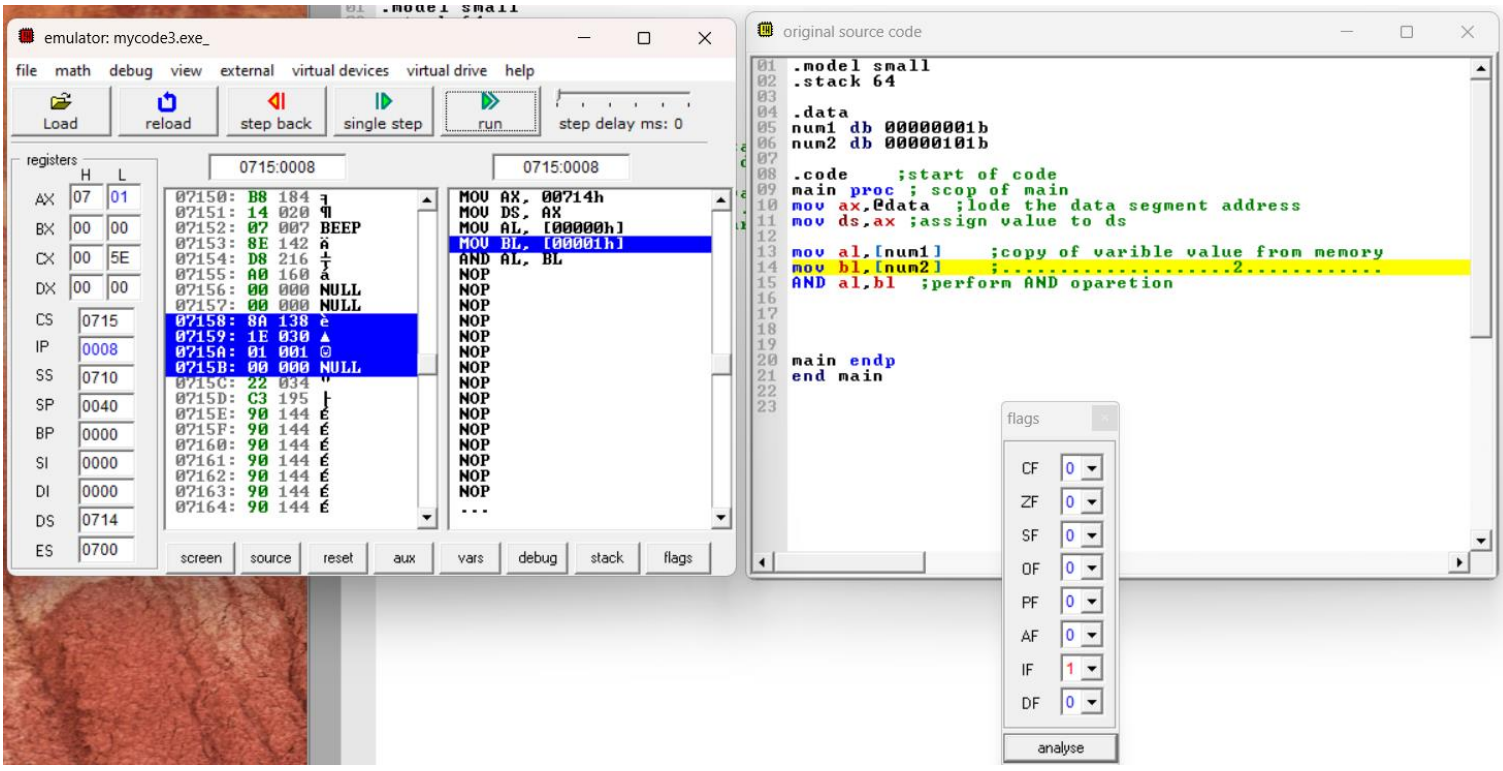


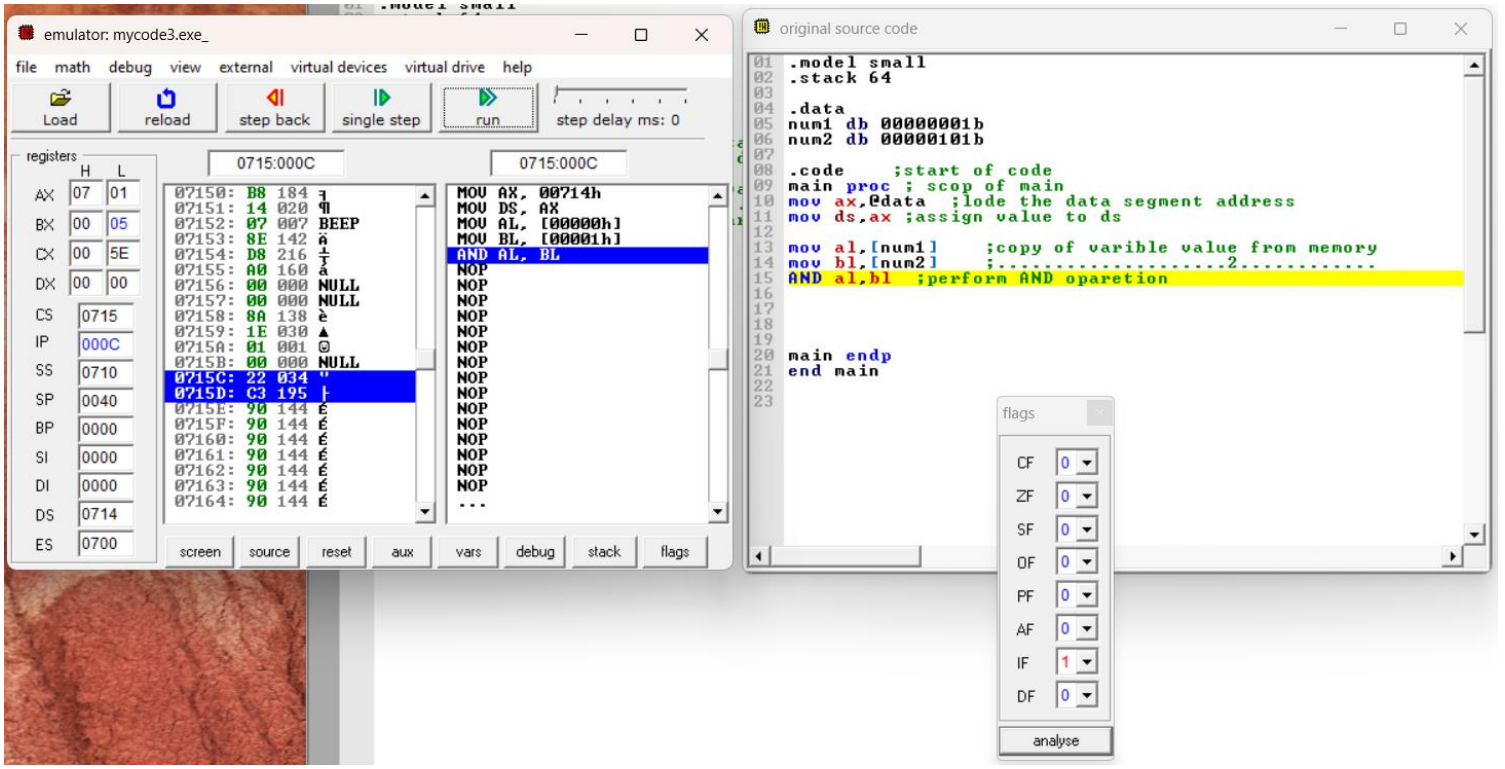
السؤال الثاني2

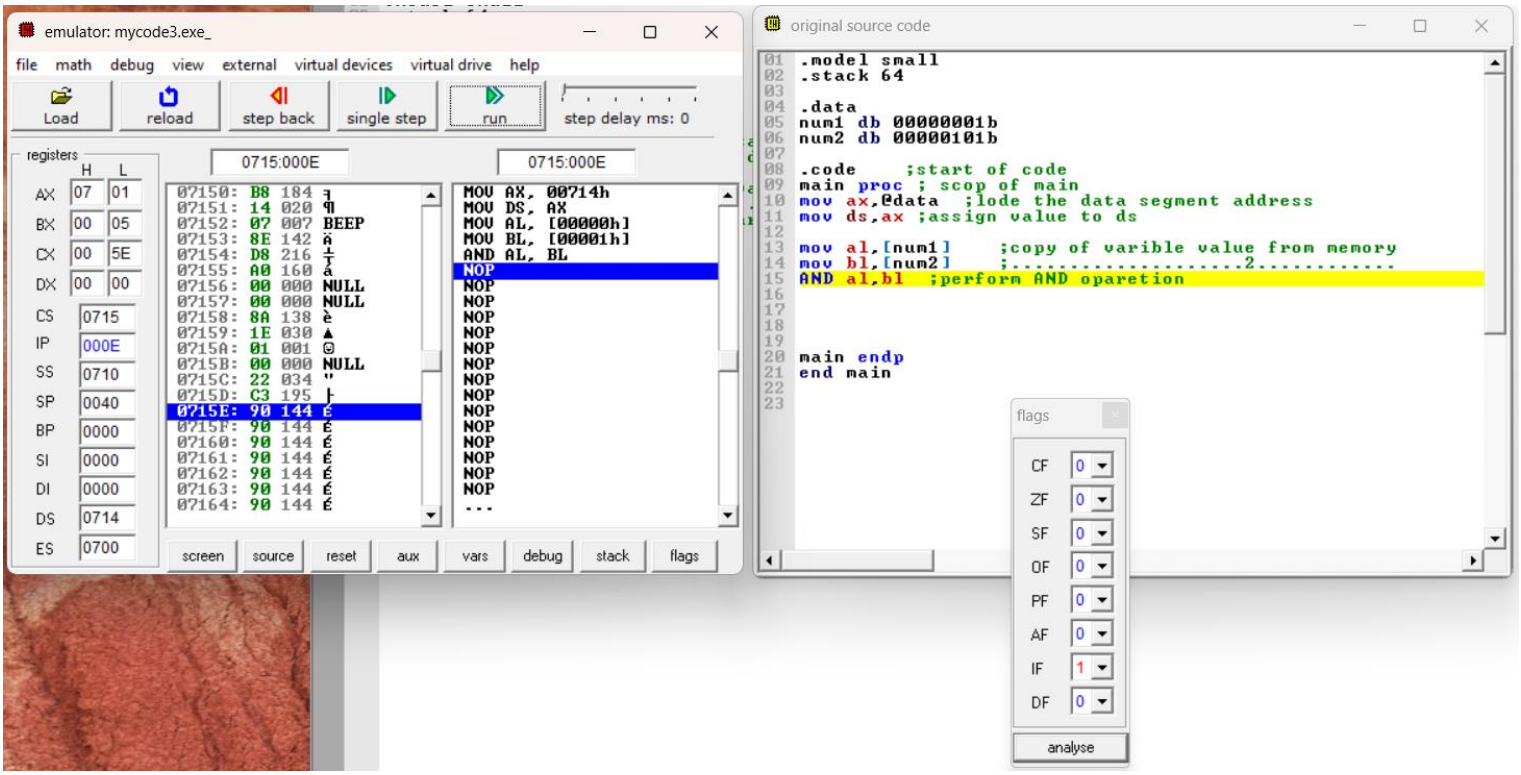
-استدعاء قيمة المتغير num1 من جزء .dataوتخزينها في المسجل al

-استدعاء قيمة المتغير num2 من جزء .dataوتخزينها في المسجل bl

- تفيذ عملية AND ولايوجد تأثيرات على Carry Flag قيمته صفر




emulator: mycode3.exe_
file math debug view external virtual devices virtual drive help
Load
reload
step back
single step
run
step delay ms: 0
registers
H L
AX 07 01
BX 00 05
CX 00 5E
DX 00 00
CS 0715
IP 000C
SS 0710
SP 0040
BP 0000
SI 0000
DI 0000
DS 0714
ES 0700
0715:000C
0715:000C
07150: B8 184
07151: 14 020
07152: 07 007 BEEP
07153: 8E 142
07154: D8 216
07155: A0 160
07156: 00 000 NULL
07157: 00 000 NULL
07158: 8A 138
07159: 1E 030
0715A: 01 001
0715B: 00 000 NULL
0715C: 22 034 "
0715D: C3 195
0715E: 90 144
0715F: 90 144
07160: 90 144
07161: 90 144
07162: 90 144
07163: 90 144
07164: 90 144
MOV AX, 00714h
MOV DS, AX
MOV AL, [00000h]
MOV BL, [00001h]
AND AL, BL
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
...
screen
source
reset
aux
vars
debug
stack
flags
original source code
01 .model small
02 .stack 64
03
04 .data
05 num1 db 00000001b
06 num2 db 00000101b
07
08 .code ;start of code
09 main proc ; scop of main
10 mov ax,@data ;lode the data segment address
11 mov ds,ax ;assign value to ds
12
13 mov al,[num1] ;copy of varible value from memory
14 mov bl,[num2] ;.....2.....
15 AND al,bl ;perform AND oparetion
16
17
18
19
20 main endp
21 end main
22
23
flags
CF 0
ZF 0
SF 0
OF 0
PF 0
AF 0
IF 1
DF 0
analyse


emulator: mycode3.exe_
file math debug view external virtual devices virtual drive help
Load
reload
step back
single step
run
step delay ms: 0
registers
H L
AX 07 01
BX 00 05
CX 00 5E
DX 00 00
CS 0715
IP 000E
SS 0710
SP 0040
BP 0000
SI 0000
DI 0000
DS 0714
ES 0700
0715:000E
0715:000E
07150: B8 184
07151: 14 020
07152: 07 007 BEEP
07153: 8E 142
07154: D8 216
07155: A0 160
07156: 00 000 NULL
07157: 00 000 NULL
07158: 8A 138
07159: 1E 030
0715A: 01 001
0715B: 00 000 NULL
0715C: 22 034 "
0715D: C3 195
0715E: 90 144
0715F: 90 144
07160: 90 144
07161: 90 144
07162: 90 144
07163: 90 144
07164: 90 144
MOV AX, 00714h
MOV DS, AX
MOV AL, [00000h]
MOV BL, [00001h]
AND AL, BL
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
...
screen
source
reset
aux
vars
debug
stack
flags
original source code
01 .model small
02 .stack 64
03
04 .data
05 num1 db 00000001b
06 num2 db 00000101b
07
08 .code ;start of code
09 main proc ; scop of main
10 mov ax,@data ;lode the data segment address
11 mov ds,ax ;assign value to ds
12
13 mov al,[num1] ;copy of varible value from memory
14 mov bl,[num2] ;.....2.....
15 AND al,bl ;perform AND oparetion
16
17
18
19
20 main endp
21 end main
22
23
flags
CF 0
ZF 0
SF 0
OF 0
PF 0
AF 0
IF 1
DF 0
analyse