

Lab10

Hyunki Lee

Problem1

- a. AL = 0019FF2D ZF = 0 SF = 1
- b. AL = 0019FF48 ZF = 0 SF = 1
- c. AL = 0019FF6F ZF = 0 SF = 1
- d. AL = 0019FFA3 ZF = 0 SF = 1 PF = 1

Yes changed. After 'and, or, xor' are operated, the result will be stored in the destination operand.

Therefore, the value of al which is the destination operand is changed.

Problem2

- a. CF = 0 ZF = 0 SF = 0 al = 0019FF0F
- b. CF = 0 ZF = 0 SF = 0 bl = 00231006
- c. CF = 1 ZF = 0 SF = 1 cl = 00401005

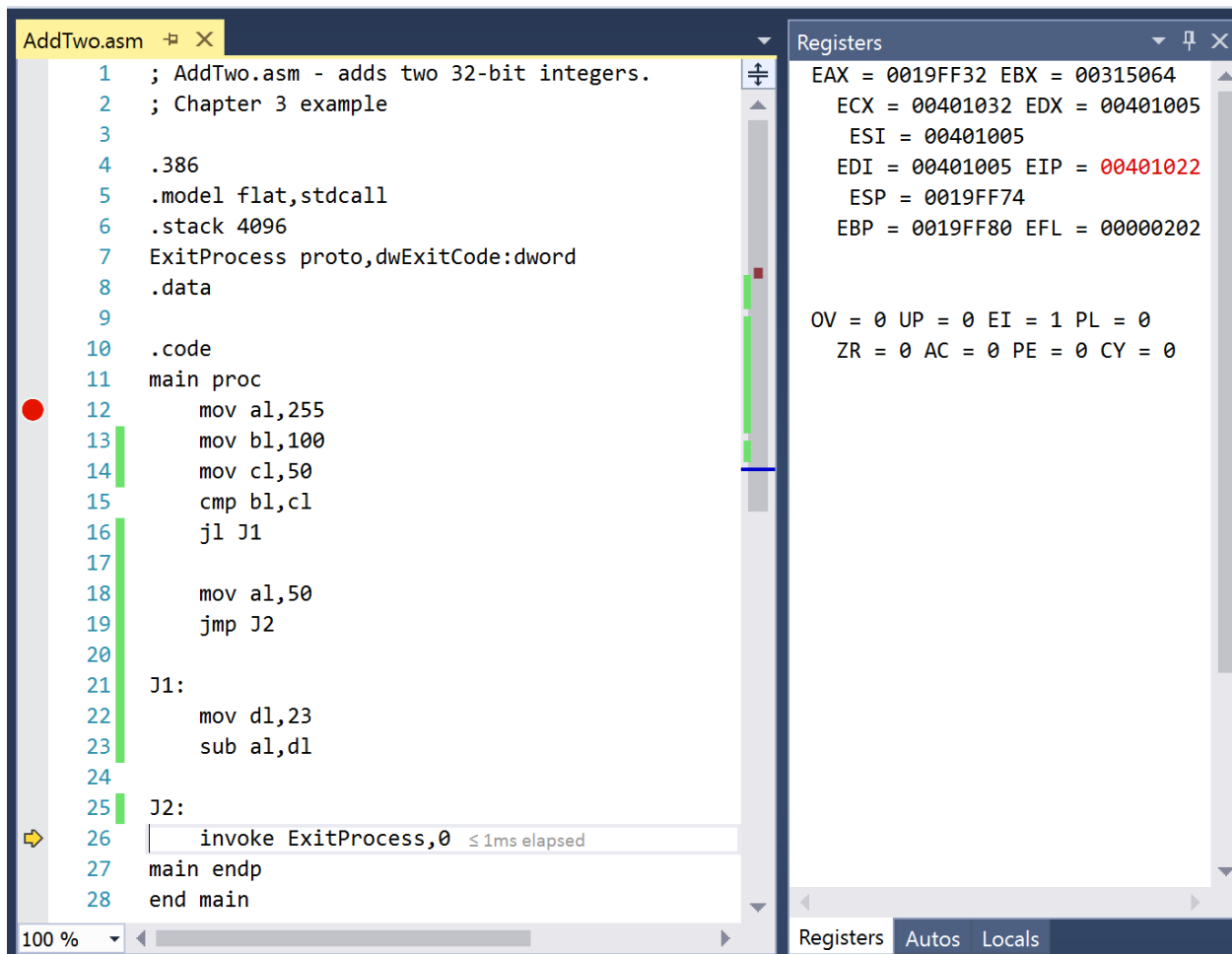
Test: no change

Cmp: no change

The two instructions – test and cmp – are not destructive subtraction of source from destination.

Therefore, the result value of destination operand is not changed.

Problem3



The screenshot shows a debugger window with two main panes. The left pane displays assembly code for a file named 'AddTwo.asm'. The code includes comments, directives for 386 architecture, flat model, stdcall convention, and a stack size of 4096. It defines a procedure 'main' that moves the value 255 into 'al', compares it with 100, and jumps if less (JL) to label 'J1'. Label 'J1' moves 23 into 'dl' and subtracts it from 'al'. Label 'J2' is also present. The right pane shows the state of CPU registers. EAX is 0019FF32, EBX is 00315064, ECX is 00401032, EDX is 00401005, ESI is 00401005, EDI is 00401005, EIP is 00401022 (highlighted in red), ESP is 0019FF74, and EBP is 0019FF80. EFL is 0000202. Below the registers, status flags are shown: OV=0, UP=0, EI=1, PL=0, ZR=0, AC=0, PE=0, CY=0. A status bar at the bottom indicates '100 %' zoom and tabs for 'Registers', 'Autos', and 'Locals'.

```
1 ; AddTwo.asm - adds two 32-bit integers.
2 ; Chapter 3 example
3
4 .386
5 .model flat,stdcall
6 .stack 4096
7 ExitProcess proto,dwExitCode:dword
8 .data
9
10 .code
11 main proc
12     mov al,255
13     mov bl,100
14     mov cl,50
15     cmp bl,cl
16     jl J1
17
18     mov al,50
19     jmp J2
20
21 J1:
22     mov dl,23
23     sub al,dl
24
25 J2:
26     invoke ExitProcess,0 ≤ 1ms elapsed
27 main endp
28 end main
```

Registers

EAX = 0019FF32 EBX = 00315064
ECX = 00401032 EDX = 00401005
ESI = 00401005
EDI = 00401005 EIP = 00401022
ESP = 0019FF74
EBP = 0019FF80 EFL = 0000202

OV = 0 UP = 0 EI = 1 PL = 0
ZR = 0 AC = 0 PE = 0 CY = 0


100 % Registers Autos Locals

Final value of al : 0019FF32

Problem4

Memory 1

Address: &sum



Columns: Auto

0x00404000	66	01	00	00	32	00	00	00	0a	00	00	00	3c	00	00	00	14	00	00	00	f...2.....<.....
0x00404014	21	00	00	00	48	00	00	00	59	00	00	00	2d	00	00	00	41	00	00	00	!...H...Y...-...A...
0x00404028	48	00	00	00	12	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	H.....
0x0040403C	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
0x00404050	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
0x00404064	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
0x00404078	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00

Sum: 00000166