# Comp 3350: Computer Organization & Assembly Language

# HW # 4: Theme: Debugging, Flags, Data Declarations

*All main questions carry equal weight.*

*(Credit awarded to only those answers for which work has been shown. All questions must be worked by hand)*

1. A. Write a program fragment that will set the sign flag

.code

Mov eax, 3

Sub eax, 4 ; EAX = -1, SF = 1

B. Write a program fragment that will set the overflow flag.

.code

Mov al, 3

Add al, 125 ; OF = 1

C. Write a program fragment that sets the zero flag.

.code

Mov al, 1

Dec al ; AL = 0, ZF = 1

D. What will be the value of the parity flag after the following lines execute?

Mov al, 3 ; al = 0011b

Add al, 5 ; al = 1000b, PF = 0

1. Given the following data declarations:

.data

Alpha BYTE 1Ah, 2Bh, 3Ch, 4CH

Beta DWORD 11223344h

Delta DWORD 44332211h

Iota DWORD 434h

Zeta WORD 124h

1. Write instructions that move *Beta* into *EBX* and then adds *Alpha* to the same register

Mov ebx, Beta

Mov eax, DWORD PTR Alpha

Add ebx, eax

1. Write a set of instructions that adds the first and third elements of the array *Alpha* into AL

Mov al, Alpha

Add al, [Alpha + 2]

1. Write a set of instructions that moves *Iota* into *EAX,* adds the value stored in *Zeta* to the same register

Mov eax, Iota

Mov bx, Zeta

Movzx edx, bx

Add eax, edx

1. Write an instruction that moves the last two bytes in *ALPHA* into *AX*.

Mov ax, WORD PTR [Alpha + 2]

1. What are the contents of AX subsequent to part D of this question?

4C3Ch

1. Fill in the requested register values after executions of the instructions:

Show the memory map.

.data

myBytes BYTE 22h, 11h, 44h, 33h

myWords WORD 4321h,7856h, ABDFh, EFC0h, 3425h

myDoubles DWORD AB23h, B4C4h, CD54h, 8679h, 6F6Ah

myPointer DWORD myDoubles

.code

mov esi, OFFSET myBytes

mov ax, WORD PTR [esi+2] ; A. AX = 3344h

mov eax, DWORD PTR myWords ; B. EAX = 78564321h

mov esi, myPointer

mov ax, WORD PTR [esi+8] ; C. AX = CD54h

mov ax, WORD PTR [esi+1] ; D. AX = 23ABh

mov ax, WORD PTR [esi-6] ; E. AX = ABDFh

Memory Map:

Step 1: EAX = 00003344h

Step 2: EAX = 78564321h

Step 3: EAX = 7856CD54h

Step 4: EAX = 785623ABh

Step 5: EAX = 7856ABDFh

1. What is the value of ax after the follow:

.data

myArray DWORD 3 DUP (2), 21, 4, 65, 0CDE

.code

mov ax, TYPE myarray ; ax = 4

mov ax, lengthof myarray ; ax = 3 + 1 + 1 + 1 + 1 = 7

mov ax, sizeof my array ; ax = 4 \* 7 = 28

1. Fill in the requested register values after executions of the instructions (Do not let your eyes deceive you. There are some mov**S**x instructions and some mov**Z**x instructions.):

.code

mov bx, 0F42Ch

movzx eax, bx ; A. EAX = 0000F42Ch

movzx edx, bl ; B. EDX = 0000002Ch

movzx cx, bh ; C. CX = 00F4h

mov bx, 0EF32h

movsx eax, bx ; D. EAX = FFFFEF32h

movsx edx, bl ; E. EDX = 00000032h

movsx cx, bh ; F. CX = FFEFh

1. What will be the value of the destination operand after each of the following instructions execute?

.data

var1 SBYTE -1, -2, -3, -4

var2 WORD A000h, B000h, C000h, D000h

var3 SWORD -16, -42

var4 DWORD 11, 22, 33, 44

.code

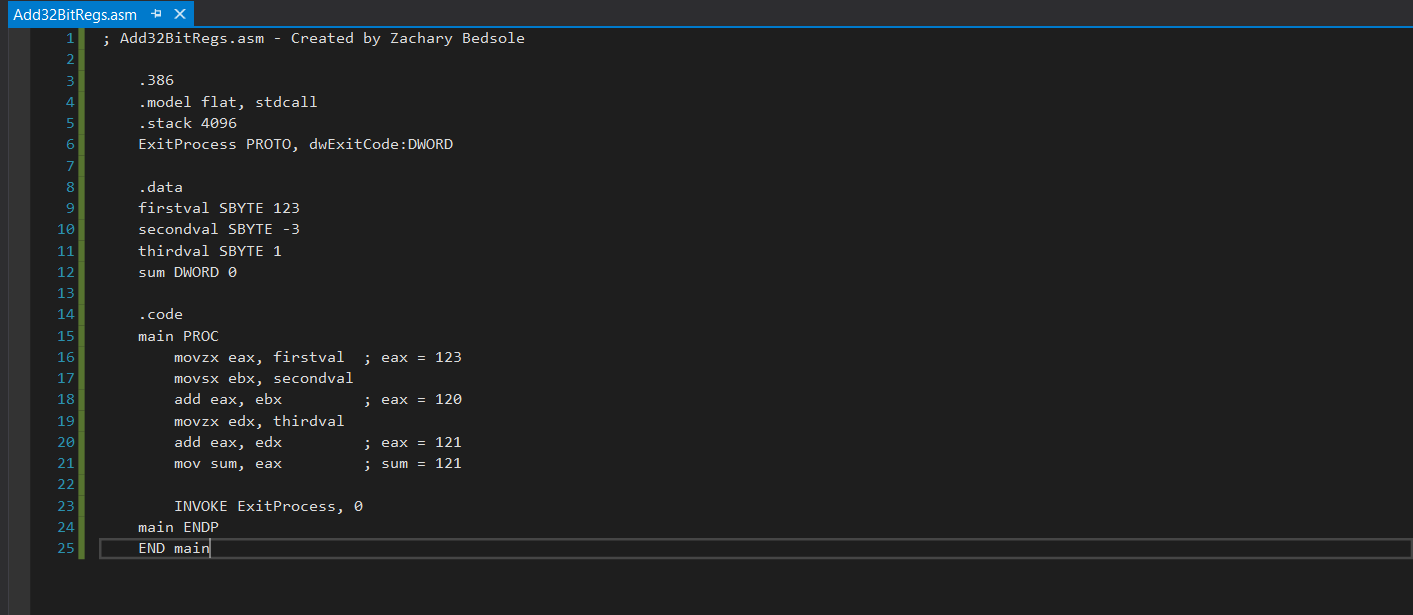
mov ax, var2 ; ax = A000h

mov ax, [var2+4] ; ax = C000h

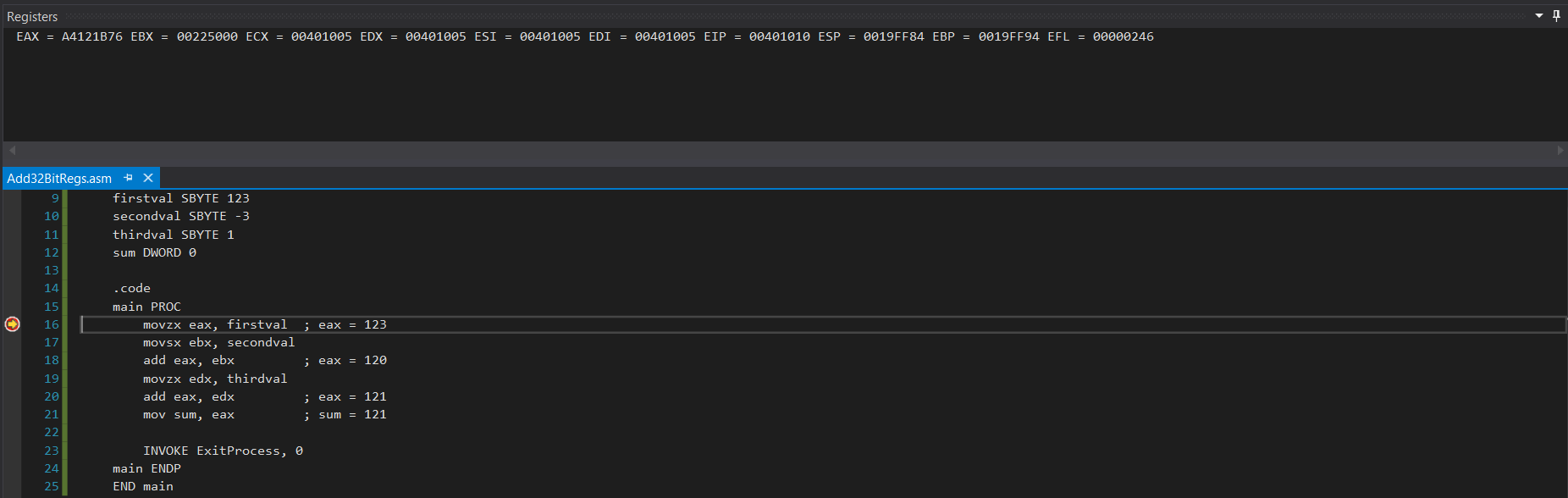
mov ax, var3 ; ax = -16

mov ax, [var3-2] ; ax = D000h

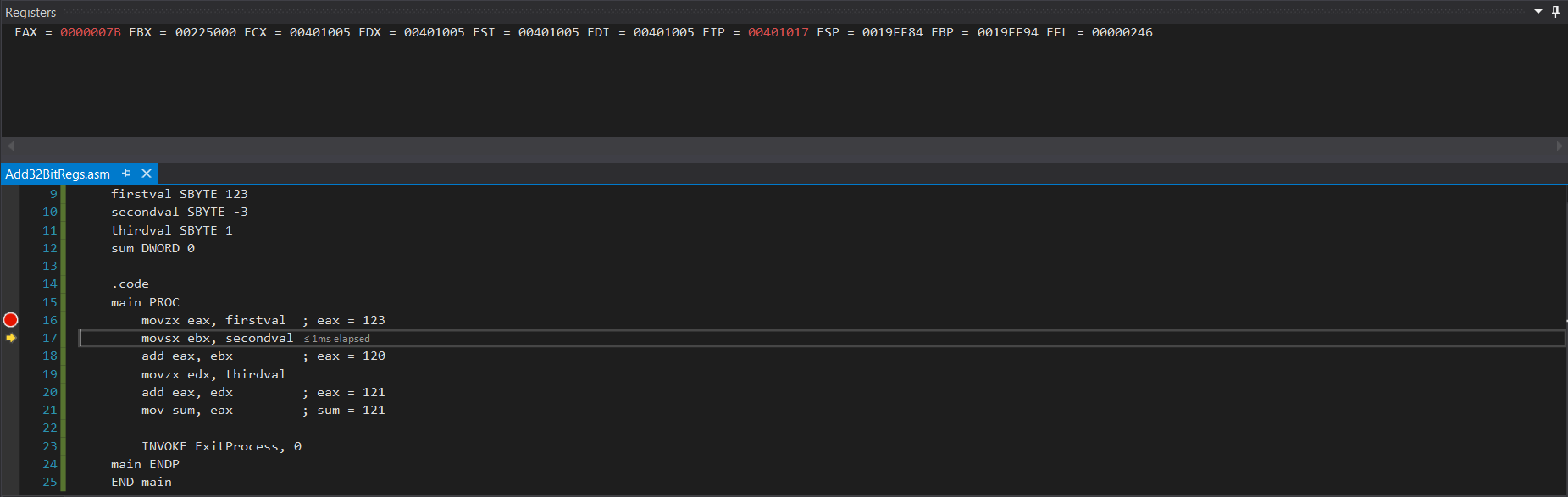
1. Using the *AddVariables* program from the textbook as a reference, write a program that adds three signed byte sized integers using only *32-bit registers*. Please embed program code into your homework submission.



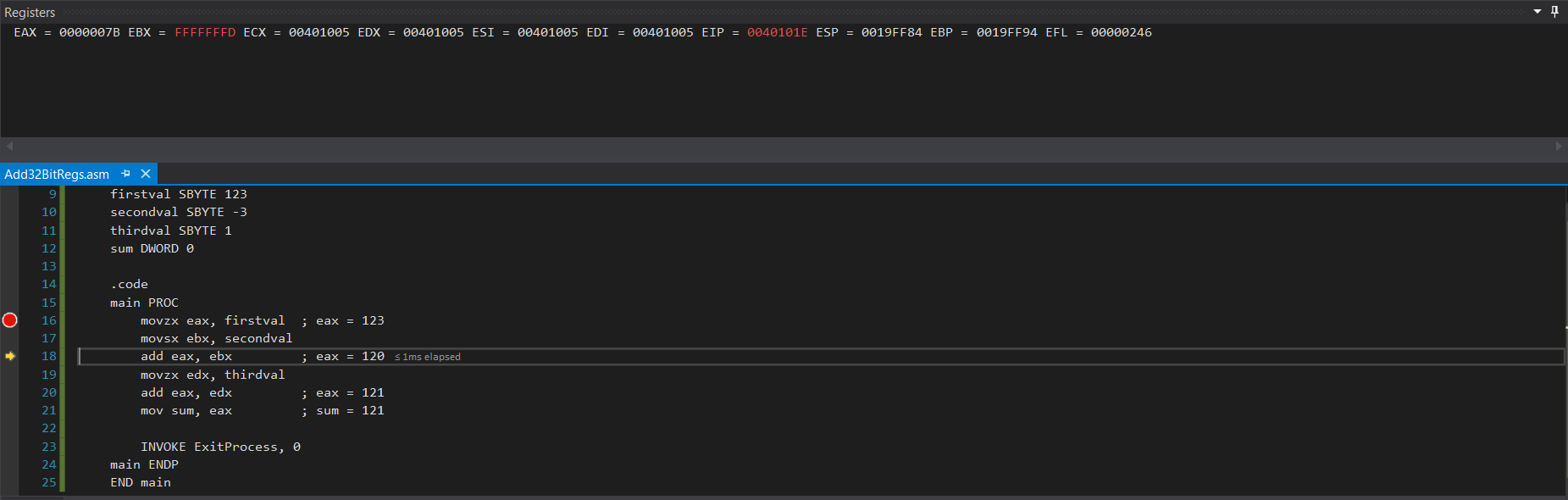
Step 1:



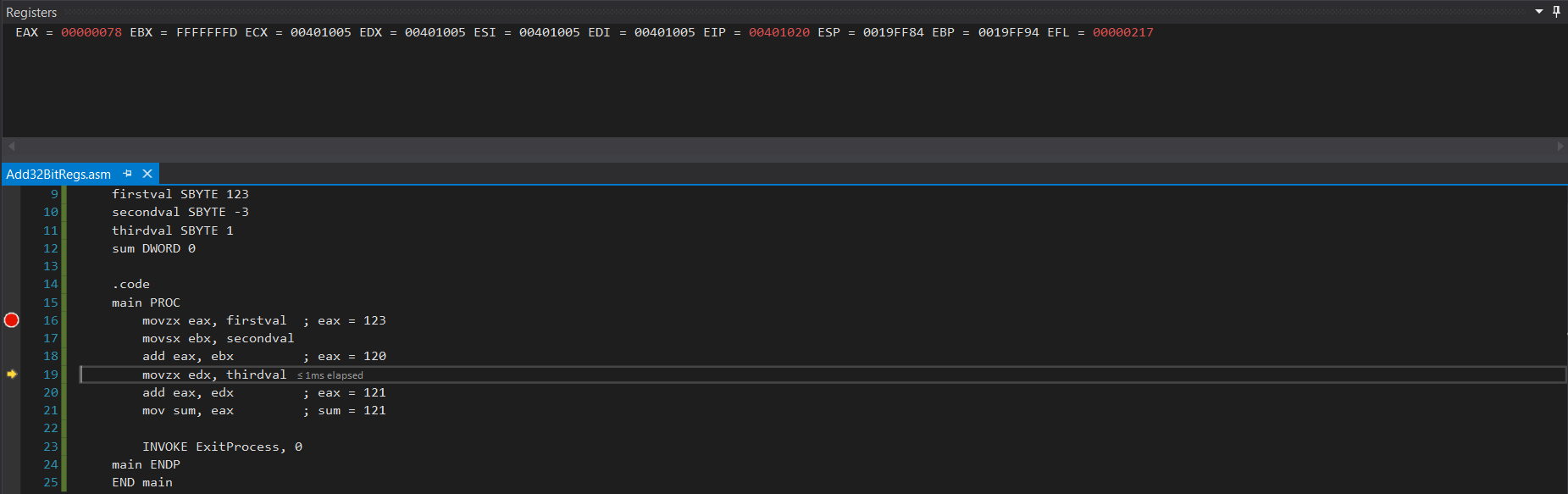
Step 2:



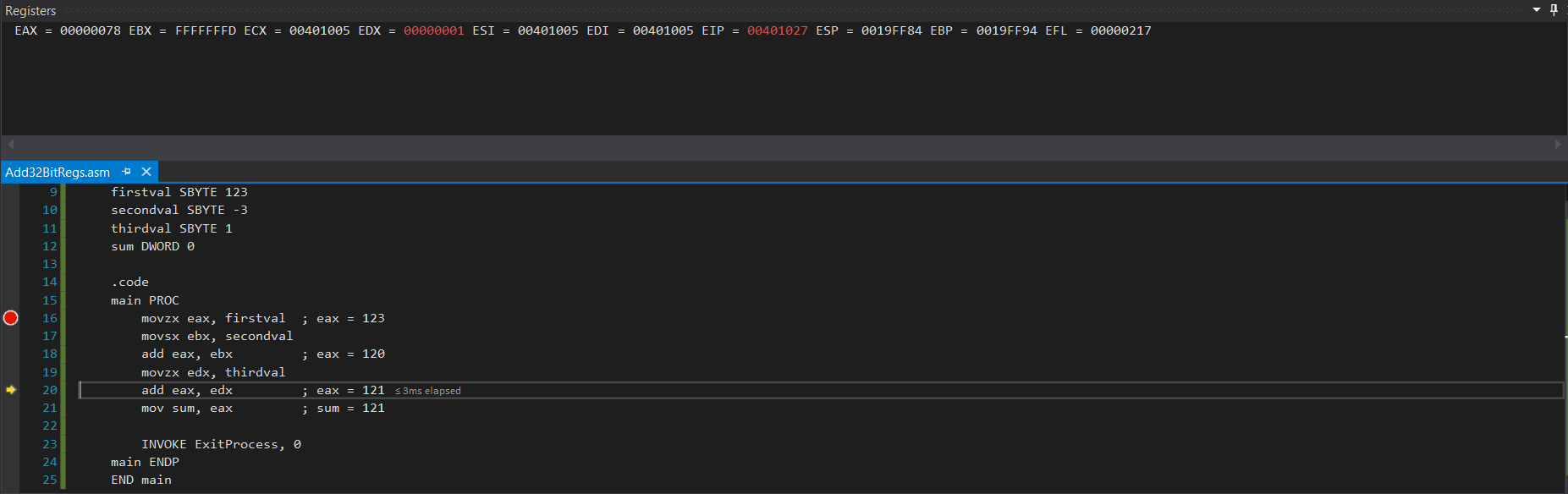
Step 3:



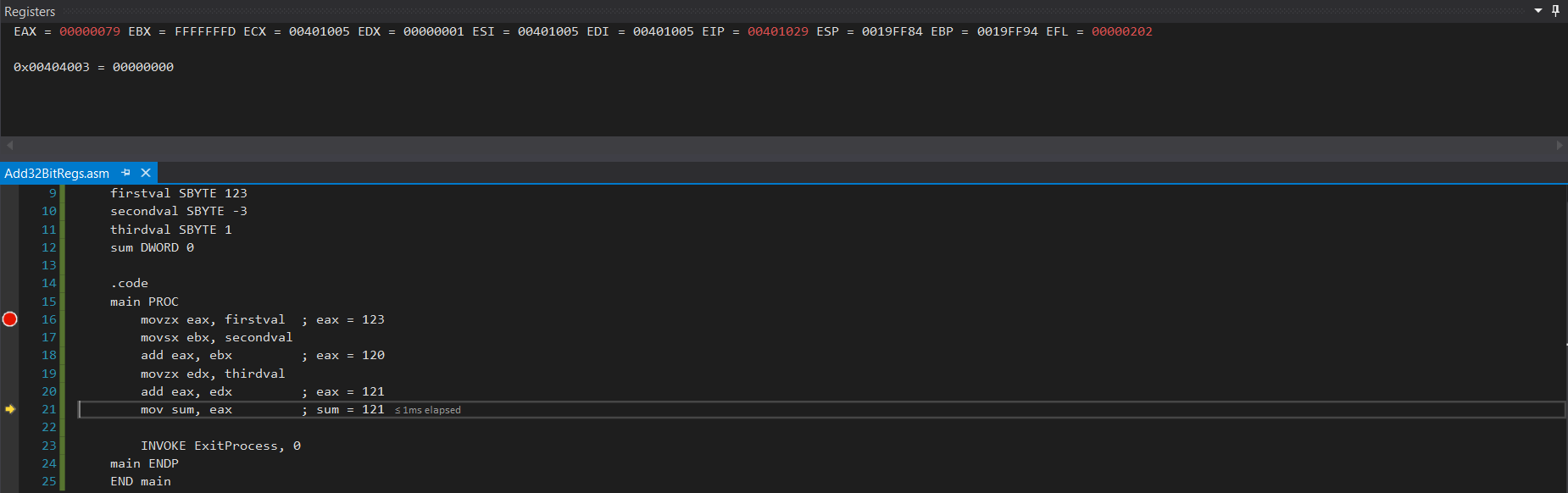
Step 4:



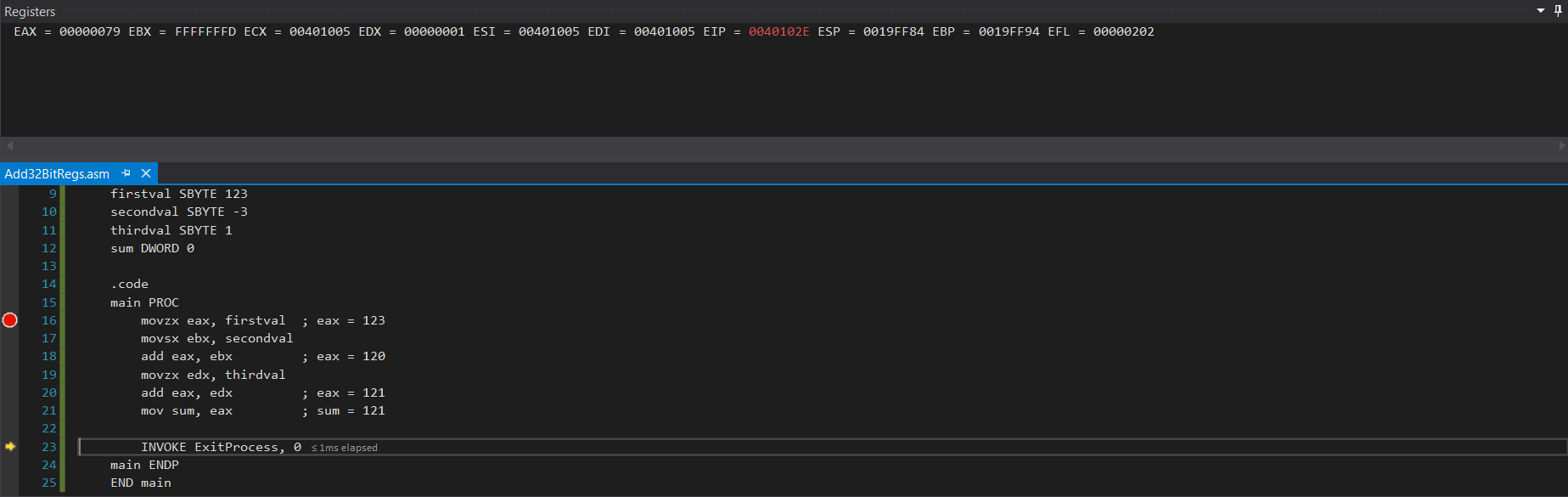
Step 5:



Step 6:



Step 7:



**Note:** Screenshots of each step were included to show modification of the registers in real time. This is an improvement of the answer in HW3.

1. Write a program that prints your *<FirstName Lastname>* on your screen. You can use the following. Assemble and generate the output using MASM and Visual Studio. Embed your output in your submission.

TITLE My first assembly program

INCLUDE Irvine32.inc

.DATA

Message BYTE “*FirstName Lastname*”,0

.CODE

main PROC

mov edx, offset message

Call WriteString

exit

main ENDP

END main

