Assembly Language Report (Week 4)

Group 7

104502517戴逸任

104502518劉冠聲

Program CODE:

.data

myID BYTE 1,0,4,5,0,2,5,1,8 ;Student ID

size\_ID = LENGTHOF myID ;size\_ID is length of myID

myID\_odd BYTE ?

myID\_even BYTE ?

myID\_result BYTE ?

.code

start@0 PROC

mov ecx,size\_ID ;count=size\_ID=9

xor esi,esi ;turn esi,bl,dl to 0

xor bl,bl

xor dl,dl

L1:

mov al,myID[esi] ;store the value in myID to al

test al,1 ;test al is a odd or even

jnz L2 ;if it is odd, jump to L2

add bl,al ;if it is even, add the value into bl

jmp L3 ;after adding, jump to L3

L2:

add dl,1 ;add one odd count

L3:

inc esi ;point to next myID address

loop L1

mov myID\_odd,dl ;store odd count into myID\_odd

mov myID\_even,bl ;store even sum into myID\_even

mov cl,myID\_odd ;use odd count to be loop count

xor al,al ;turn al to 0

L: ;this loop will add even sum for odd count times

add al,myID\_even

loop L

mov myID\_result,al

exit ;end

start@0 ENDP

END start@0

Program Step & Register state:

mov ecx,size\_ID ;ecx=0x0009

xor esi,esi ;esi=0x0000

xor bl,bl ;bl=0x00

xor dl,dl ;dl=0x00

L1:

mov al,myID[esi] ;al=0x01🡪0x00🡪0x04🡪…🡪0x08

test al,1

jnz L2

add bl,al ;bl=0x00🡪0x04🡪…🡪0x0e

jmp L3

L2:

add dl,1 ;dl=0x01🡪0x02🡪…🡪0x04

L3:

inc esi

loop L1

mov myID\_odd,dl ;myID\_odd=0x04

mov myID\_even,bl ;myID\_even=0x0e

mov cl,myID\_odd ;ecx=0x0004

xor al,al ;al=0x00

L:

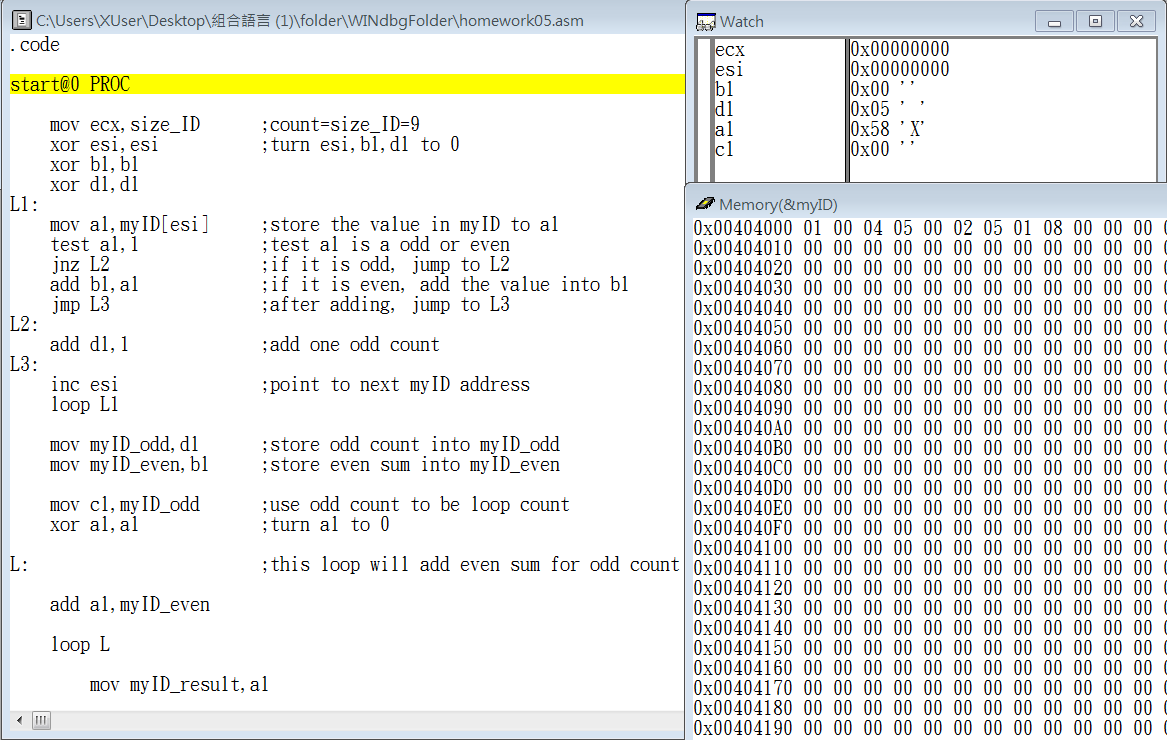
add al,myID\_even ;al=0x0e🡪…🡪0x38

loop L

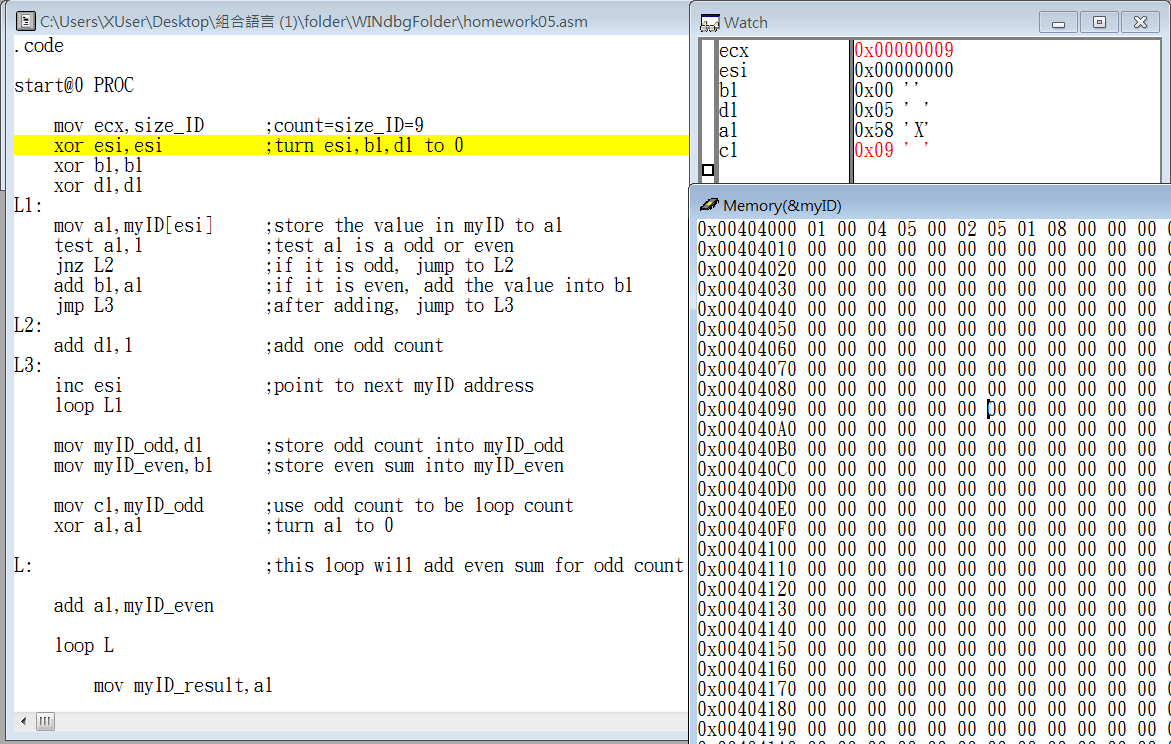
mov myID\_result,al ;myID\_result=0x38

exit

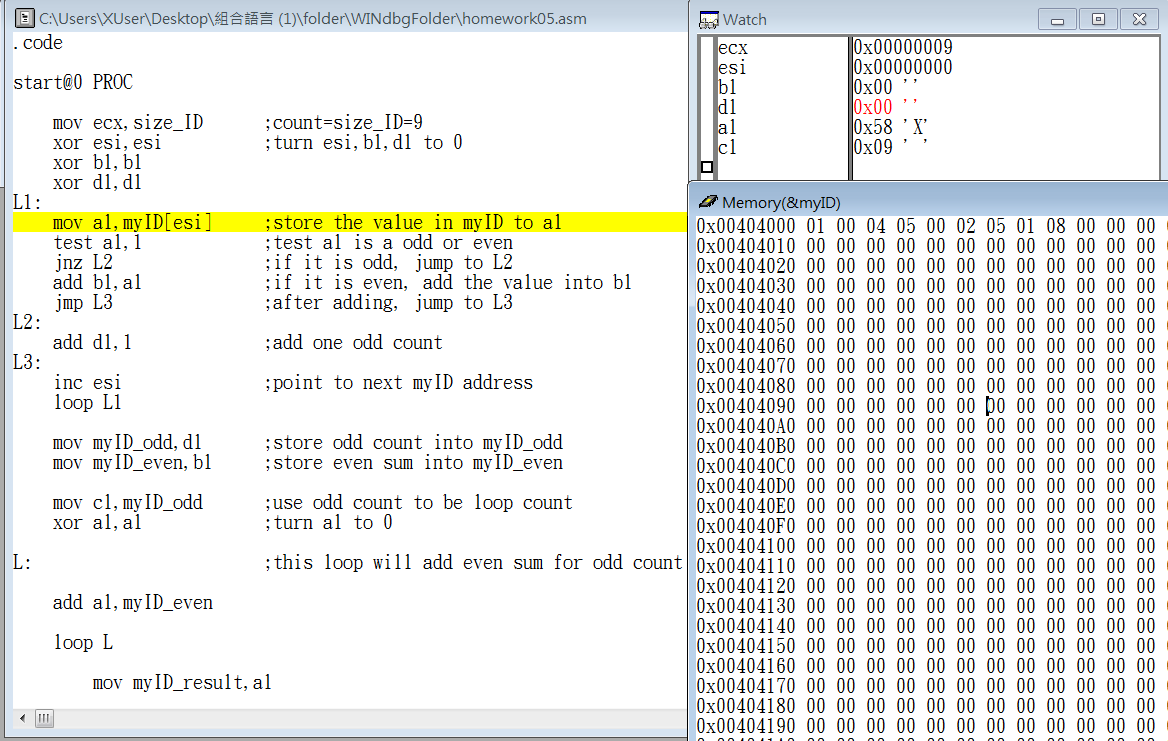
Picture & Discription:



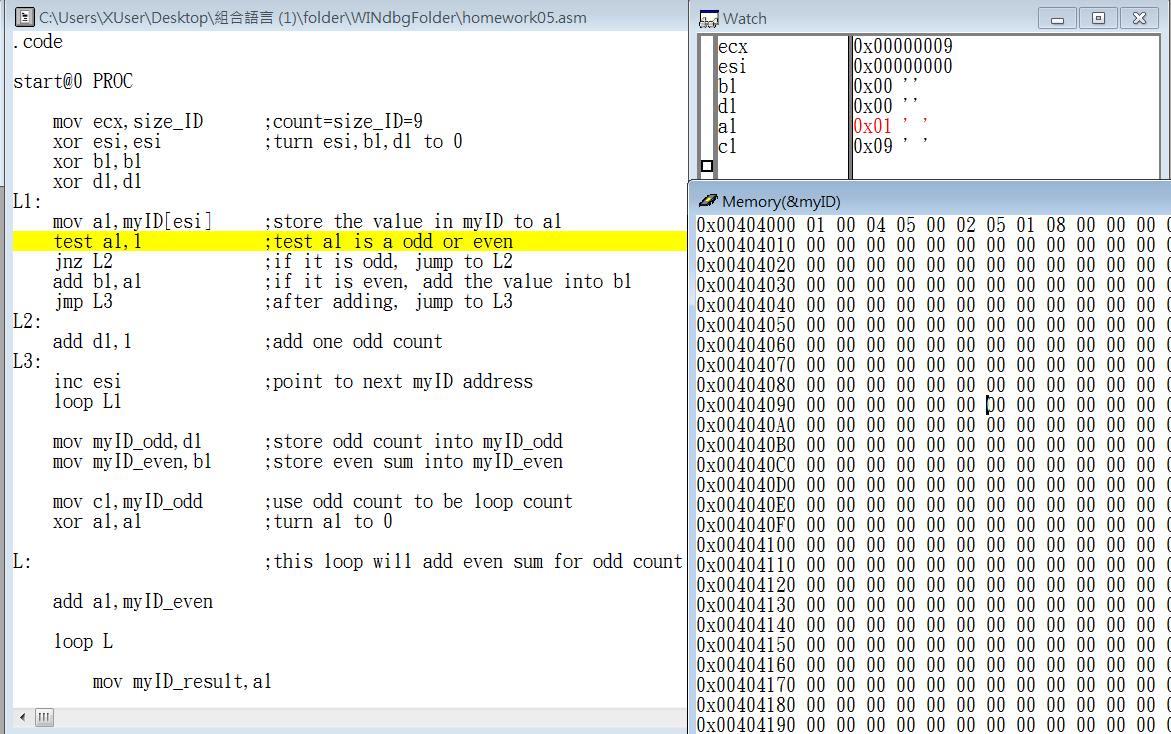
step1 start the program



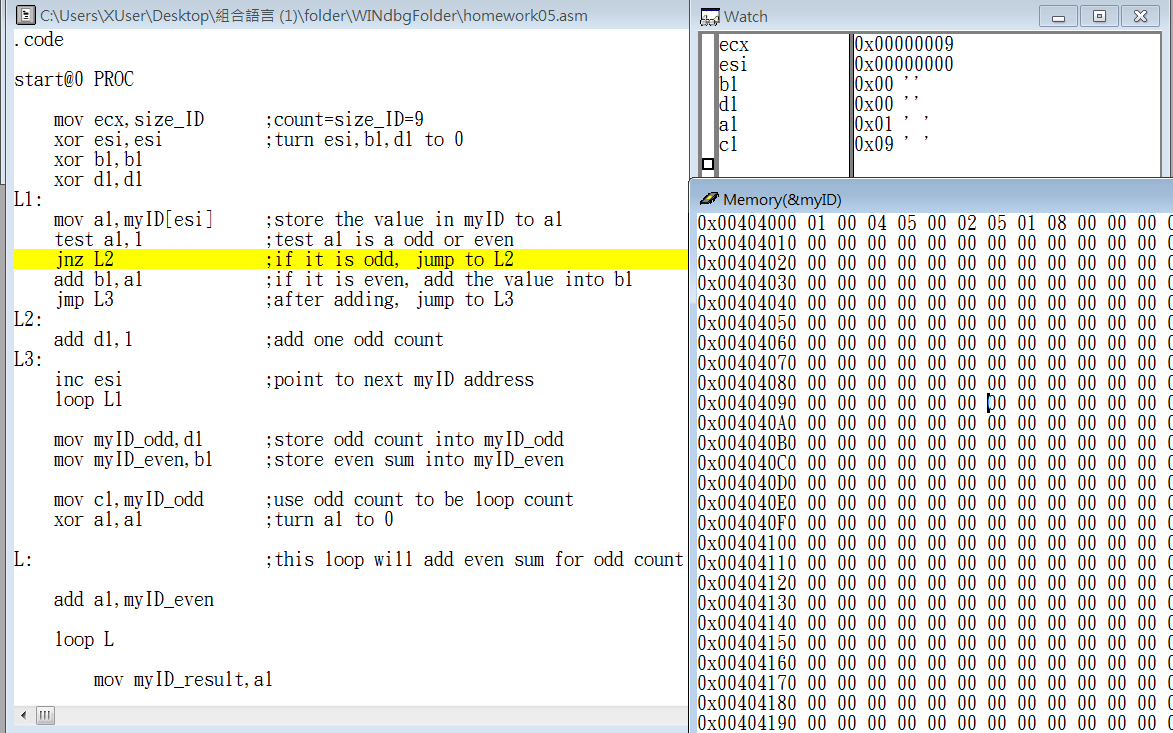
step2 move 9 into ecx



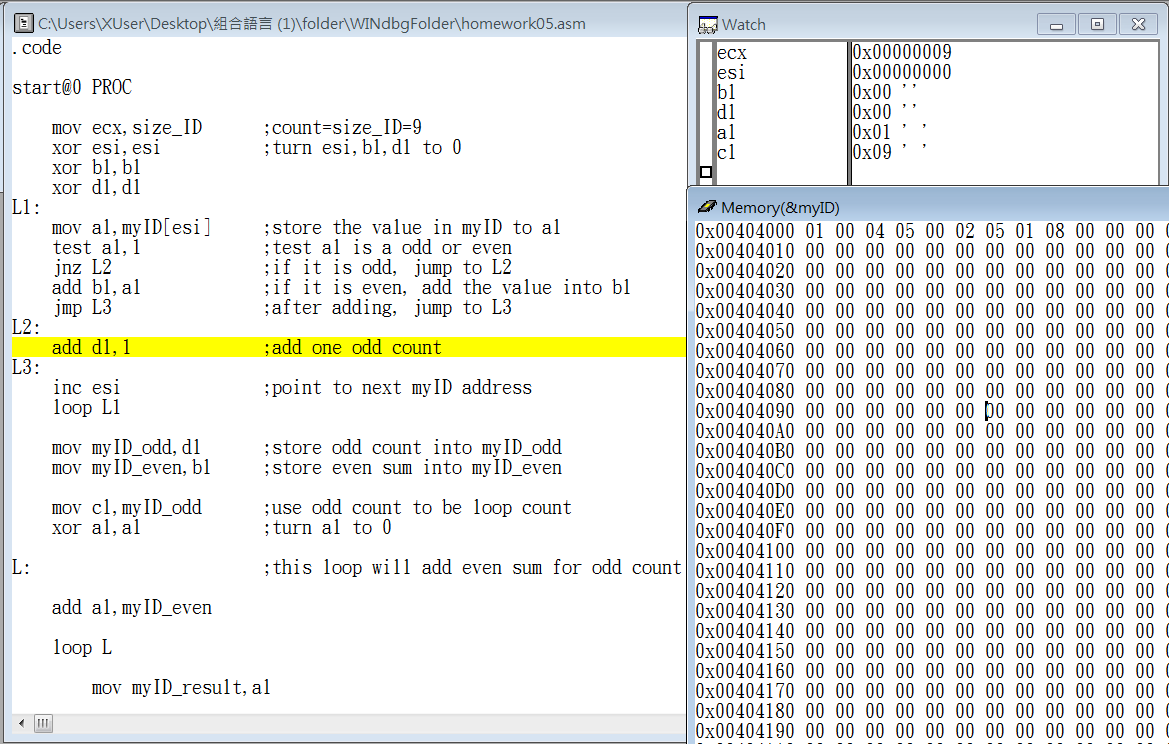
step3 reset the register esi,bl,dl into 0



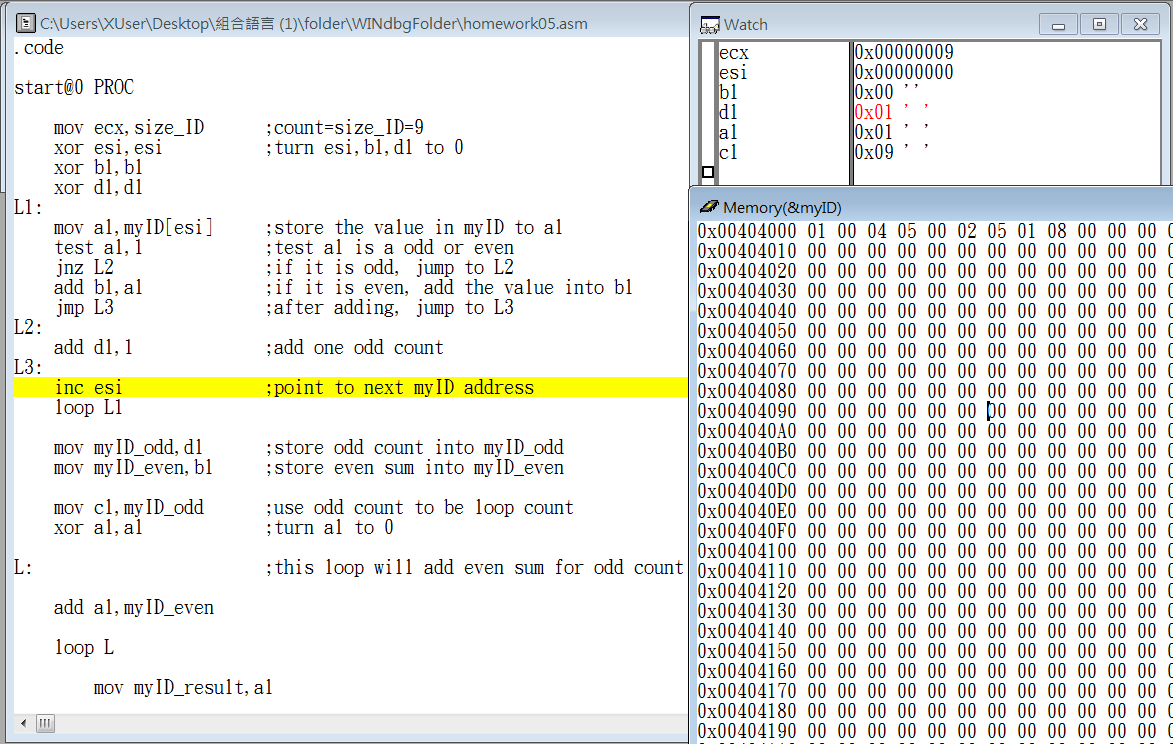
step4 move 1 into register al



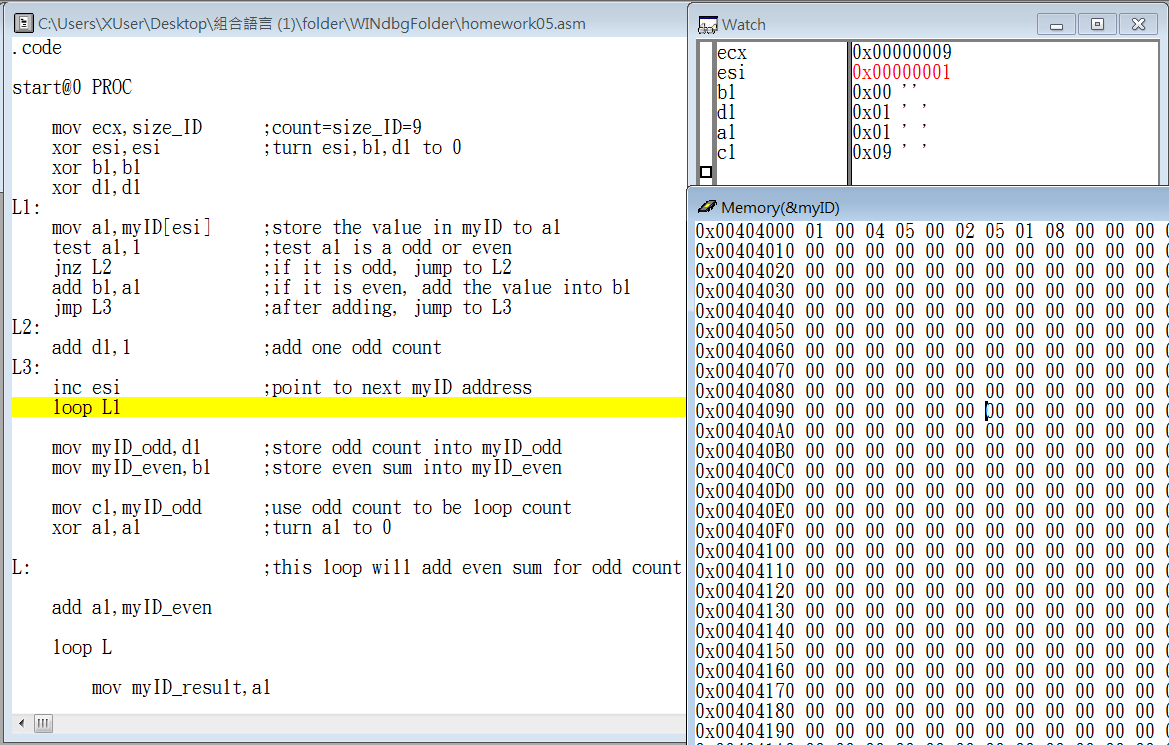
step5 if al is odd or even



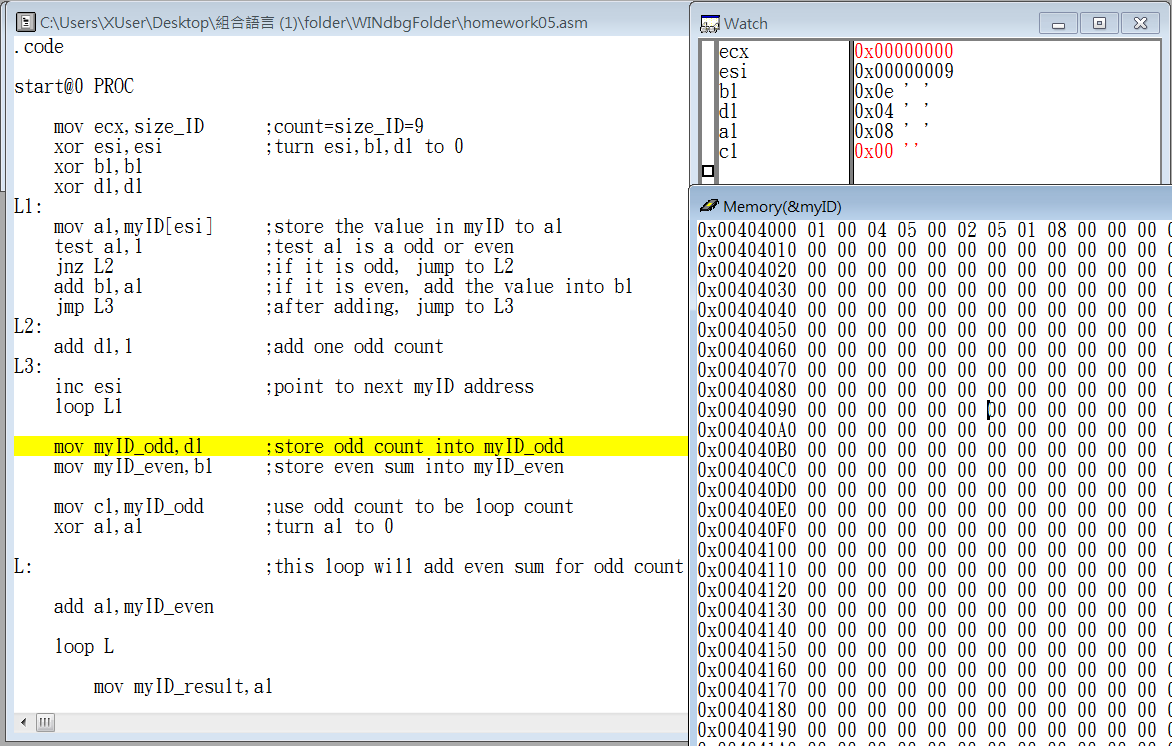
step6 if al is odd,jump to L2



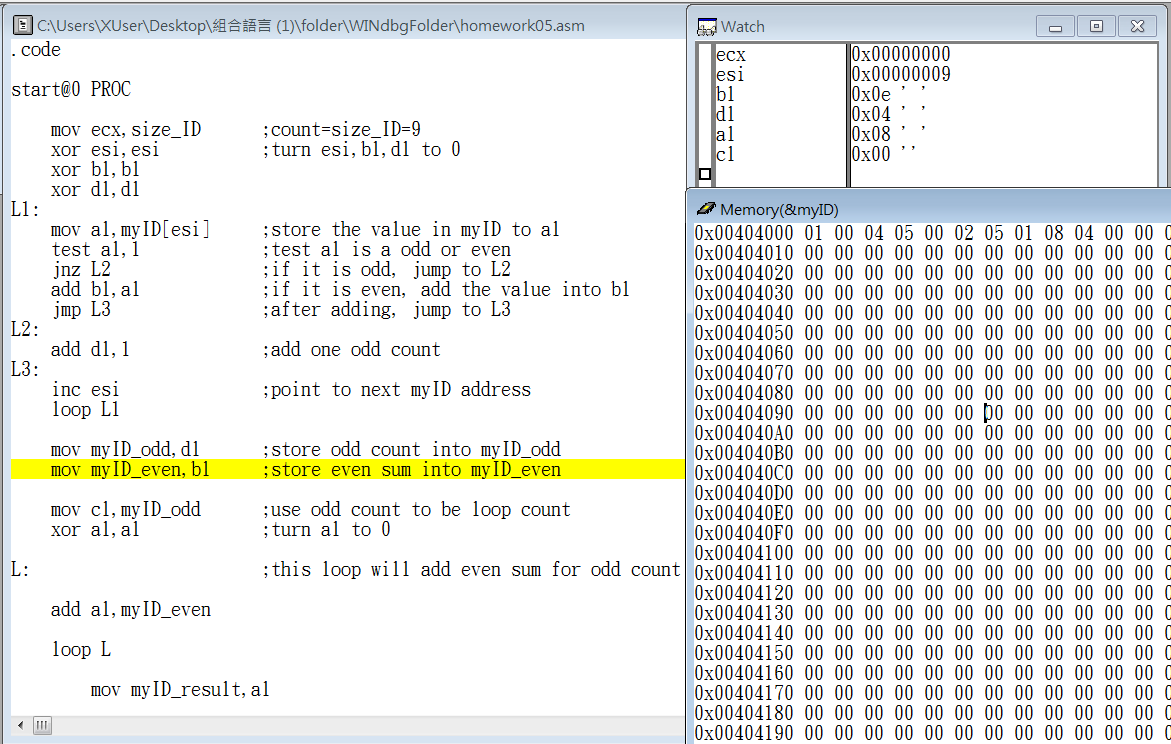
step7 add 1 into register dl



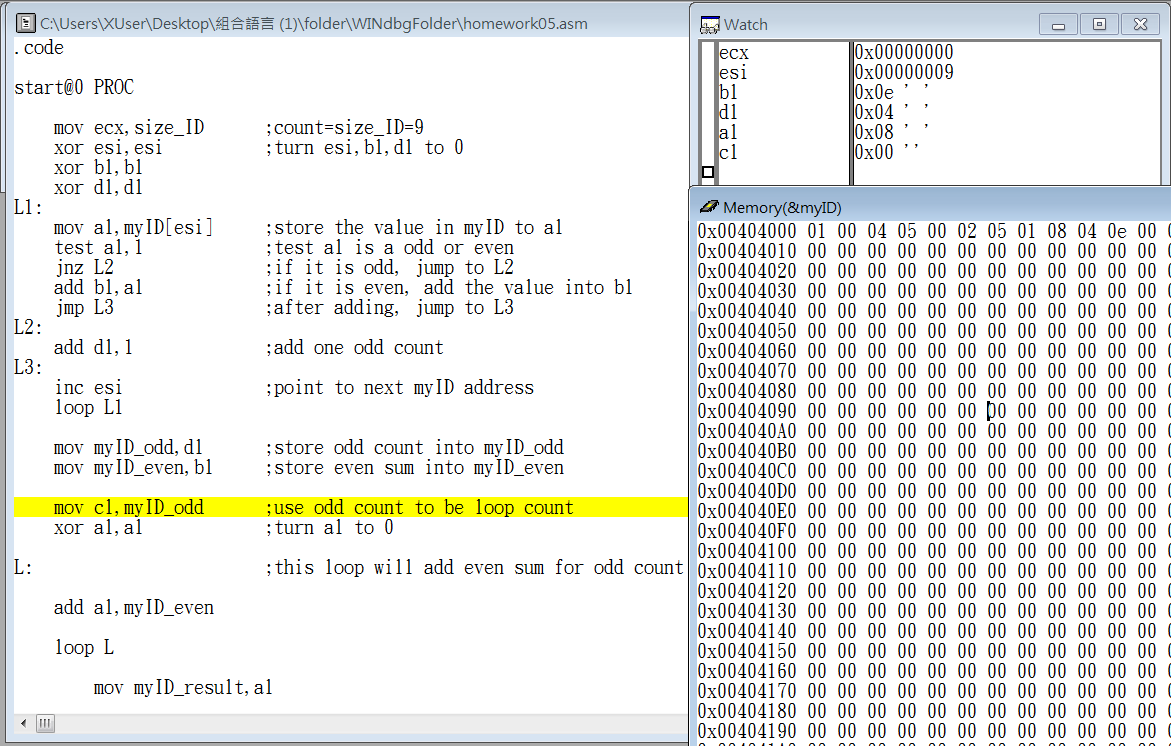
step8 esi=esi+1



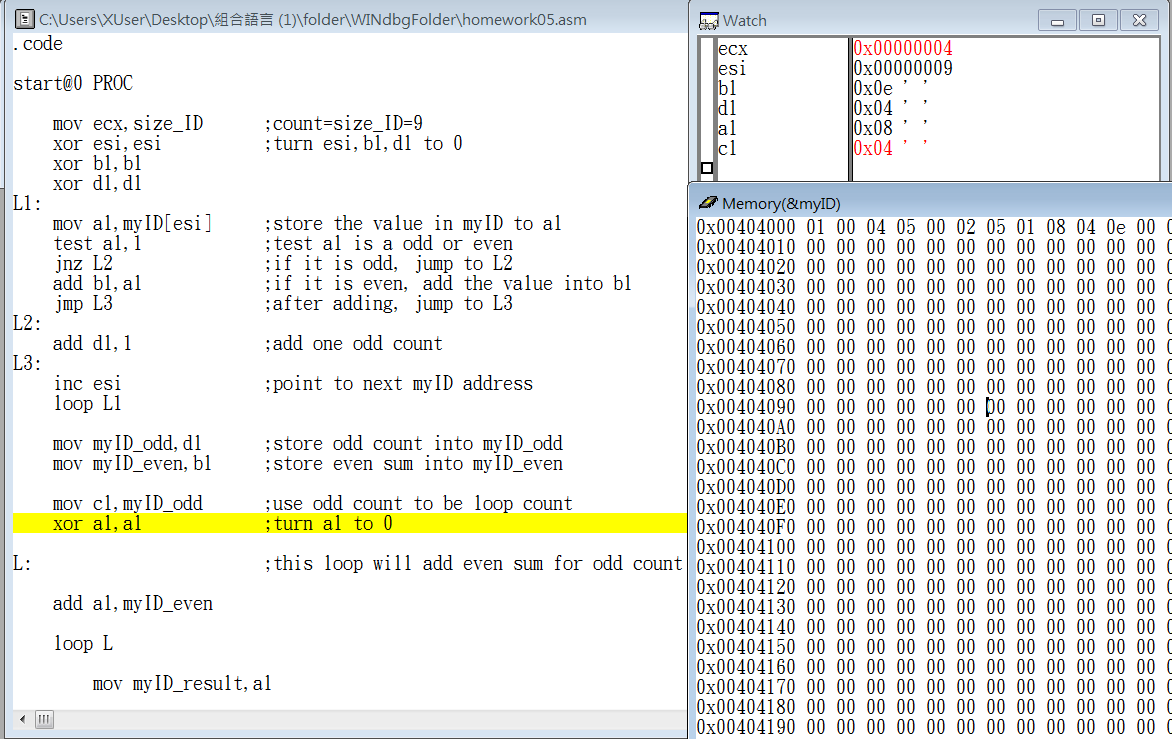
step9 end loop L1



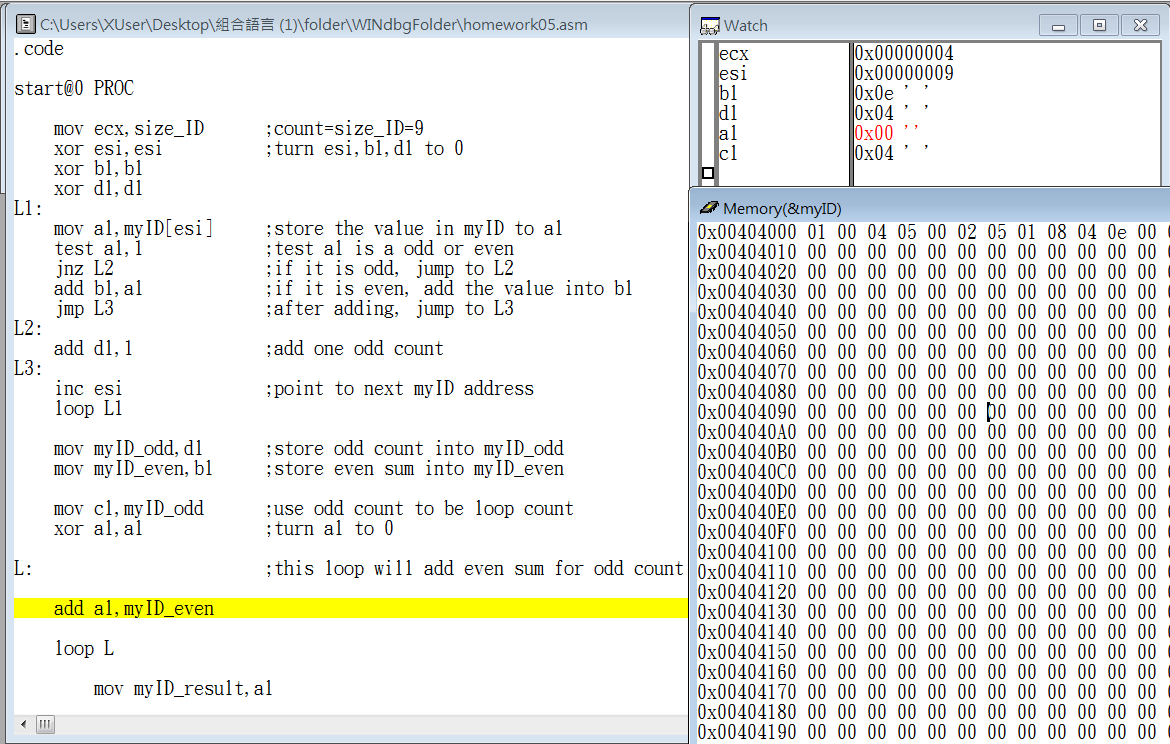
step10 move 4 inte register myID\_odd



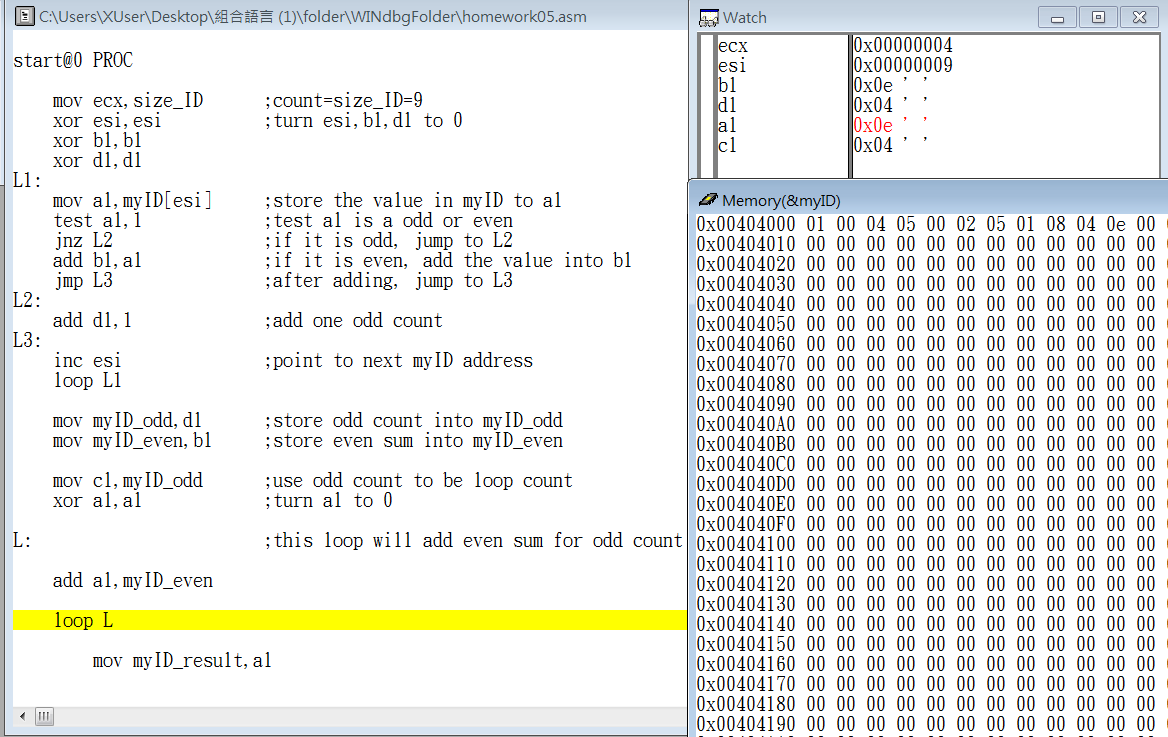
step11 move 14 into register myID\_even



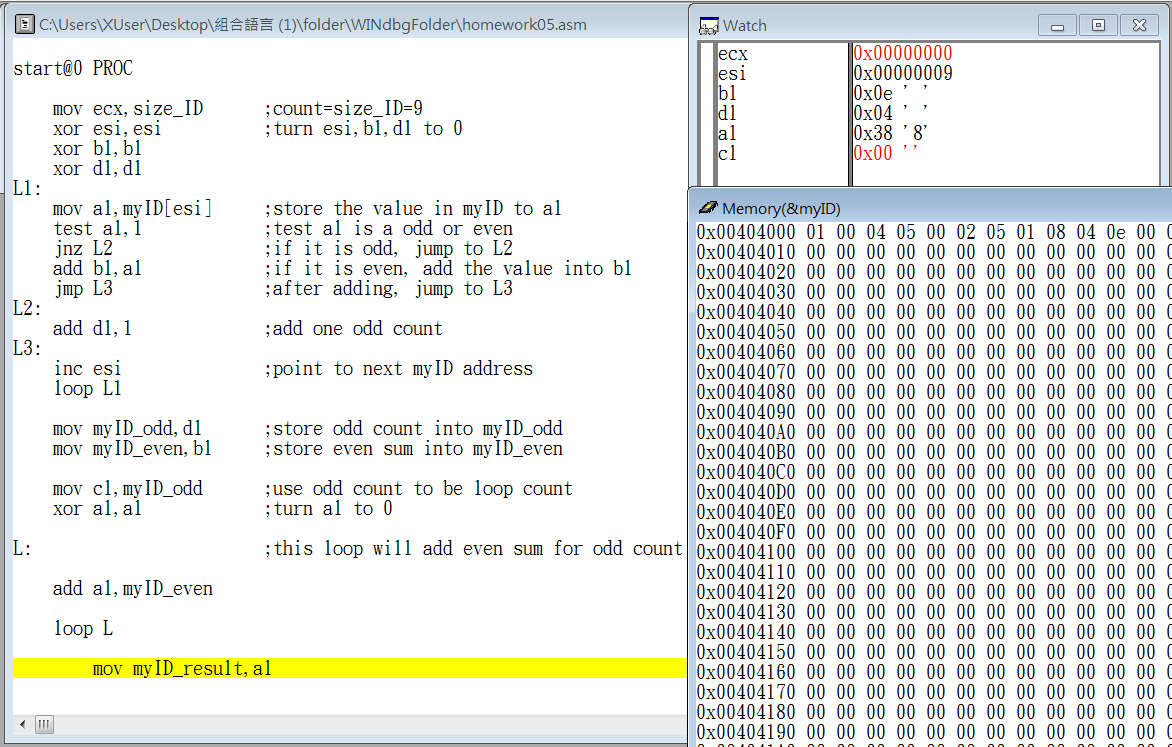
step12 move 5 into register cl



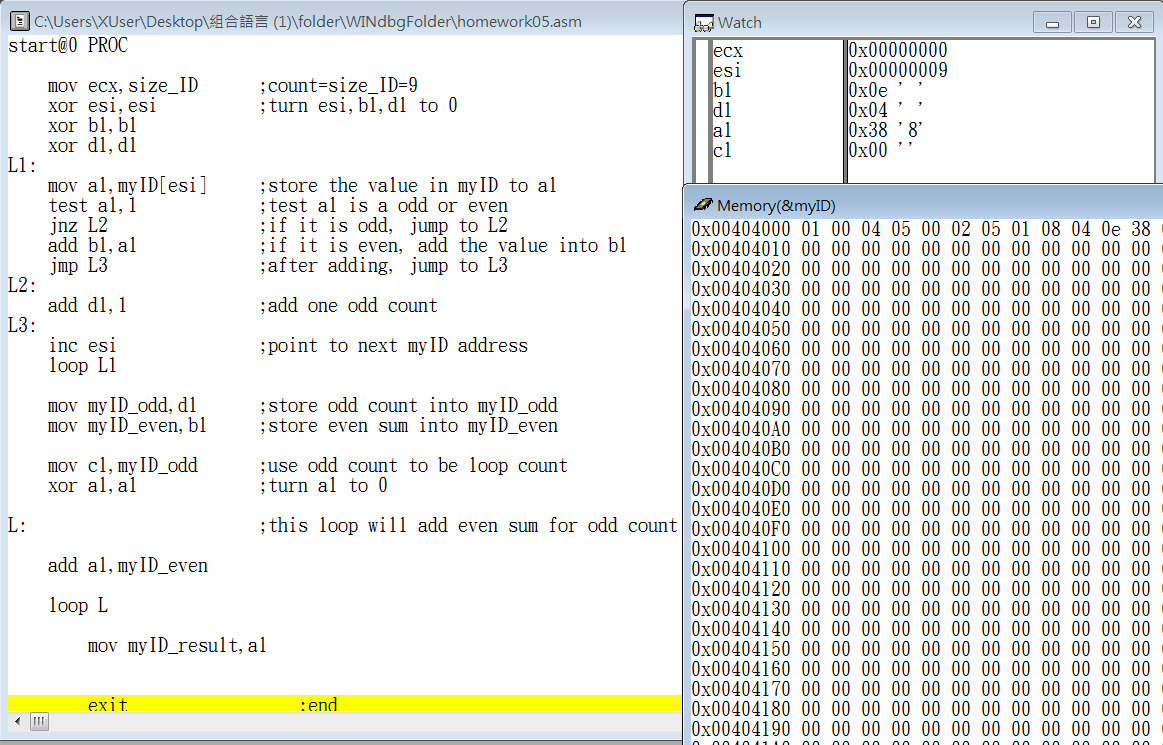
step13 reset the register al into 0



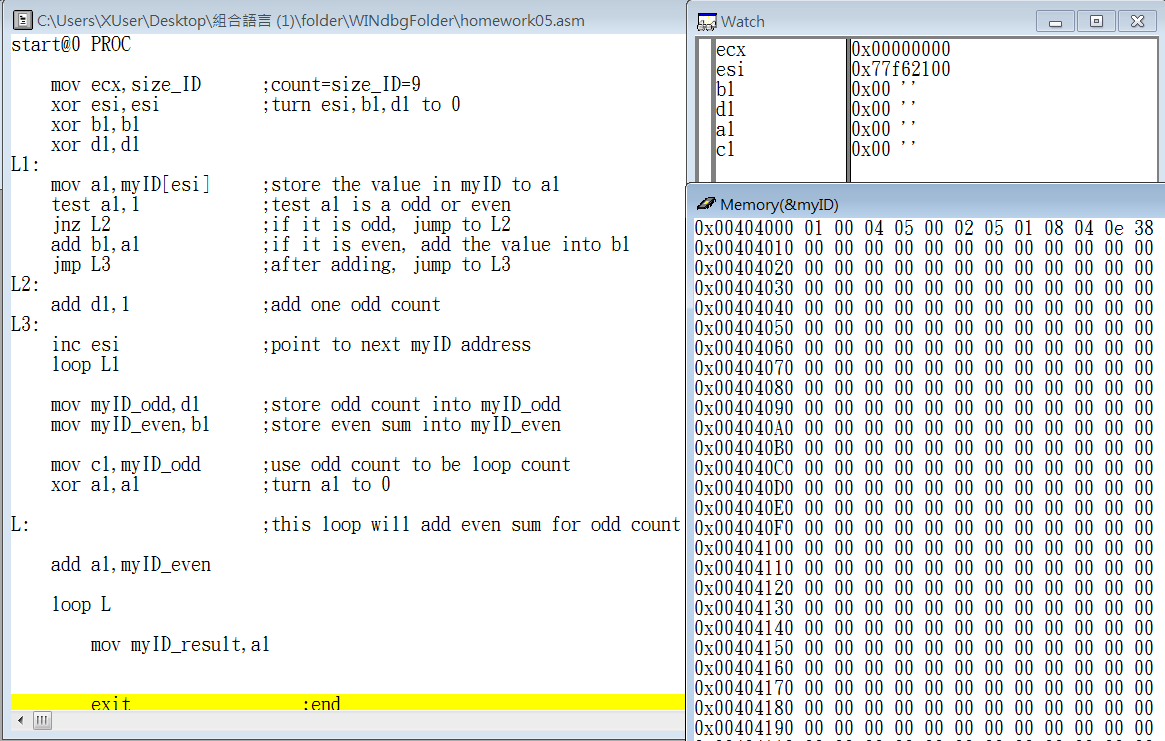
step14 move 14 into register al



step15 end loop L\



step16 move 56 into register myID\_result



step17 end the program

Review:

The class this work talk about and,xor,jump…, though these are basic concept when we study programing language, but in the assembly, it become a little difficult.

The good news is that the guiding note is clear, read the code step by step, it’s easy to understand how the program running.