Jasmine Ishigami

ICS 312

ICS312 Homework #4

**Exercise 1:**

mov ax, 00035h  
mov bl, **0??h**  
movsx bx, bl   
add ax, bx   
movzx eax, ax   
call print\_int   
call print\_nl

Working backwards:

call print\_nl

-this will just print a new line so we can skip this, it doesn't alter anything.

call print\_int

-this just prints the result, which we know is “65462”

movzx eax, ax

-movzx zero extends the unsigned value in ax to fit in eax

- we need to first convert the integer “65462” into hex

65462 = 15 (16^3) + 4022

4022 = 15 (16^2) + 182

182 = 11 (16^1) + 6

6 = 6 (16^0)

65462d = FF B6h

- since eax is a 32-bit register (4-bytes) and was zero extended its actually

eax = 00 00 FF B6 therefore since ax is a 16-bit register (2-bytes) ax = FF B6

add ax, bx

- this means ax(new) = ax(old) + bx

- we know from above the new value of ax is FF B6 therefore:

FF B6 = ax + bx

- to find the old value of ax, we look at the first instruction:

mov ax, 00035h

ax = 00 35

-thus we now have:

FF B6 = 00 35 + bx

bx = FF 81

movsx bx, bl

-movsx sign extends the signed value in bl to fit in bx

-since bl is an 8-bit (1-byte) register, we look at the lower byte and look at its most significant bit (81h = 1000 0001) and since it is a 1 the sign extension would have extended all 1s, which it did as we can see from the FF byte. Thus we can drop the FF, therefore:

bl = 81

mov bl, **0??h**

-**So the missing value in this set of instructions is 081h.**

**Exercise #2:**

CF = carry flag, OF = overflow flag

1. **2-byte** quantities: 8FF0 + A026

ccc

8FF0 **CF = 1**

+ A026

13016

8FF0 => 2's complement => 700F => -28688

A026 => 2's complement => 5FDA => -24538

-53226 < -32768

=> **OF = 1**

movsx eax 3016

eax = 0000 3016

call print\_int

0000 3016 => **12310**

2. **2-byte** quantities: 6043 + 7ABC

6043 **CF = 0**

+ 7ABC

DAFF

6043 => 24643

7ABC => 31436

56079 > 32767 => **OF = 1**

movsx eax DAFF

eax = FFFF DAFF

call print\_int

FFFF DAFF => **-9473**

3. **1-byte** quantities: F3 + 0D

cc

F3 **CF = 1**

+ 0D

100

F3 => 2's complement => 0D => -13

0D => => 13

0 => **OF = 0**

movsx eax 00

eax = 0000 0000

call print\_int

0000 0000 => **0**

4. **1-byte** quantities: E5 + 03

E5 **CF = 0**

+ 03

E8

E5 => 2's complement => 1B => -27

03 => => 3

-24 > -128 => **OF = 0**

movsx eax E8

eax = FFFF FFE8

call print\_int

0000 0000 => **-24**