EECE.3170: Microprocessor Systems Design I

Spring 2016

Homework 3 Solution

Assume the initial state of an x86 processor's registers, memory, and carry flag are:

EAX: 0x00003170
EBX: 0x9876DCBA
ECX: 0x00001995
EDX: 0xAC921E14
ESI: 0x00008440
CF: 0

Address	***Lo***			***Hi***
0x8440	*FF*	*03*	*99*	*87*
0x8444	*08*	*09*	*F6*	*BB*
0x8448	*78*	*15*	*00*	*00*

*What is the result of each of the instructions listed below? Assume that the instructions execute in sequence—in other words, the result of each instruction may depend on the results of earlier instructions. Correctly evaluating each instruction will earn you **10 points**.*

Note that you may assume any constant values shown using less than 32 bits are zero-extended to 32 bits if necessary (for example, 0x000F = 0x0000000F).

ADD AX, BX

Solution: AX = AX + BX = 0x3170 + 0xDCBA = **0x0E2Ah**, **CF = 1**

ADC EAX, ECX

Solution: EAX = EAX + ECX + CF = 0x00000E2A + 0x00001995 + 1 = **0x000027C0**, **CF = 0**

INC WORD PTR [ESI]

Solution: Add 1 to word at address ESI = 0x00008440

→ Word @ 0x8440 = 0x03FF + 1 = **0x0400** (byte @ 0x8440 = 0x00,
byte @ 0x8441 = 0x04)

MUL BYTE PTR [ESI+4]

Solution: AX = AL * unsigned byte @ (ESI+4)

→ Address = ESI + 4 = 0x8440 + 4 = 0x8444; byte @ 0x8444 = 0x08

→ AX = 0xC0 * 0x08 = 192 * 8 = 1536 = **0x0600**

SUB AX, [ESI+8]

Solution: AX = AX – word @ ESI+8

→ Address = ESI + 8 = 0x8440 + 8 = 0x8448; word @ 0x8448 = 0x1578

→ AX = 0x0600 – 0x1578 = **0xF088**, **CF = 1** (since borrow out of MSB required)

DEC AH

Solution: AH = AH – 1 = 0xF0 – 1 = **0xEFh**

IMUL AH

Solution: AX = AL * AH (signed multiplication) = 0x88 * 0xEF = -120 * -17 = 2040 = **0x07F8**

IDIV DL

Solution: AL = AX / DL (signed division) = 0x07F8 / 0x14 = 2040 / 20 = 102 = **0x66**
AH = AX % DL (remainder) = 2040 % 20 = **0x00**

DIV DH

Solution: AL = AX / DH (unsigned division) = 0x0066 / 0x1E = 102 / 30 = **0x03**
AH = AX % DH (remainder) = 102 % 30 = 12 = **0x0C**

NEG AH

Solution: AH = -AH = -0x0C = -(0000 1100_2) = (1111 0011_2 + 1 = 1111 0100_2 = **0xF4h**