EECE.3170: Microprocessor Systems Design I

Summer 2016

Homework 2 Solution

1. *Assume the state of an x86 processor's registers and memory are:*

EAX: EECE3170h
EBX: 00000001h
ECX: 00000002h
EDX: 00000004h
ESI: 00020100h
EDI: 00020110h

Address	Lo			Hi
20100h	10	00	08	00
20104h	10	10	FF	FF
20108h	08	00	19	91
2010Ch	20	40	60	80
20110h	02	00	AB	0F
20114h	30	99	11	55
20118h	40	AA	7C	EE
2011Ch	FF	BB	42	D2
20120h	30	CC	30	90

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 ***5 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, 000Fh = 0000000Fh).

MOV DL, FEh

Solution: DL = **FEh**

MOV DH, AL

Solution: DH = AL = **70h** (EDX now = 000070FEh)

MOVSX BX, BYTE PTR [ESI+000Fh]

Solution: BX = sign-extended byte at address ESI+000Fh = 00020100h + 000Fh = 0002010Fh

→ BX = 80h sign-extended = **FF80h**

MOV [EDI+ECX], EBX

Solution: Double-word at address EDI+ECX = EBX

EDI+ECX = 00020110h + 00000002h = 00020112h

→ (20112h) = EBX = **0000FF80h** (bytes ordered as 80h, FFh, 00h, 00h)

*MOV [ESI+4*ECX], AX*

Solution: Word at address ESI+4*ECX = AX

ESI + 4*ECX = 20100h + 4 * 2 = 20108h

→ (20108h) = **3170h** (bytes ordered as 70h, 31h)

XCHG CL, [ESI]

Solution: Swap byte values in CL, address 20110h → CL = **10h**, (20110h) = **02h**

MOVZX EAX, WORD PTR [EDI+ECX]

Solution: EAX = zero-extended word at address EDI+ECX = 20110h + 00000010h = 20120h

→ EAX = **0000CC30h** (original word underlined)

MOV DX, [EDI+FFFFFFFAh]

Solution: DX = word at address EDI+FFFFFFFAh = 20110h + (-6) = 2010Ah

→ DX = **9119h**

LEA ECX, [ESI+EBX+0017h]

Solution: ECX = ESI + EBX + 0017h = 20100h + 0000FF80h + 0017h = **30097h**

MOVSX EBX, BYTE PTR [ESI+4]

Solution: EBX = sign-extended byte at address 20104h = **00000010h** (original byte underlined)

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

EAX: 00003170h
EBX: 9876DCBAh
ECX: 00001995h
EDX: AC921E14h
ESI: 00008440h
CF: 0

Address	Lo			Hi
8440h	FF	03	99	87
8444h	08	09	F6	BB
8448h	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 **5 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, 000Fh = 0000000Fh).

ADD AX, BX

Solution: AX = AX + BX = 3170h + DCBAh = **0E2Ah**, **CF = 1**

ADC EAX, ECX

Solution: EAX = EAX + ECX + CF = 00000E2Ah + 00001995h + 1 = **000027C0h**, **CF = 0**

INC WORD PTR [ESI]

Solution: Add 1 to word at address ESI = 00008440h
→ Word @ 8440h = 03FFh + 1 = **0400h** (byte @ 8440h = 00h, byte @ 8441h = 04h)

MUL BYTE PTR [ESI+4]

Solution: AX = AL * unsigned byte @ (ESI+4)
→ Address = ESI + 4 = 8440h + 4 = 8444h; byte @ 8444h = 08h
→ AX = C0h * 08h = 192 * 8 = 1536 = **0600h**

SUB AX, [ESI+8]

Solution: AX = AX – word @ ESI+8
→ Address = ESI + 8 = 8440h + 8 = 8448h; word @ 8448h = 1578h
→ AX = 0600 – 1578h = **F088h**, **CF = 1** (since borrow out of MSB required)

DEC AH

Solution: AH = AH – 1 = F0 – 1 = **EFh**

IMUL AH

Solution: AX = AL * AH (signed multiplication) = 88h * EFh = -120 * -17 = 2040 = **07F8h**

IDIV DL

Solution: AL = AX / DL (signed division) = 07F8h / 14h = 2040 / 20 = 102 = **66h**
AH = AX % DL (remainder) = 2040 % 20 = **00h**

DIV DH

Solution: AL = AX / DH (unsigned division) = 0066h / 1Eh = 102 / 30 = **03h**
AH = AX % DH (remainder) = 102 % 30 = 12 = **0Ch**

NEG AH

Solution: AH = -AH = -0Ch = -$(0000\ 1100_2)$ = $(1111\ 0011_2 + 1 = 1111\ 0100_2$ = **F4h**