# EECE.3170: Microprocessor Systems Design I

Summer 2017

Homework 2 Solution

*1. (70 points) Assume the state of an x86 processor's registers and memory are:*

*EAX: 0xEECE3170*
*EBX: 0x00000001*
*ECX: 0x00000002*
*EDX: 0x00000004*
*ESI: 0x00020100*
*EDI: 0x00020110*

| ***Address*** | ***Lo*** | | | ***Hi*** |
|---|---|---|---|---|
| *0x20100* | *10* | *00* | *08* | *00* |
| *0x20104* | *10* | *10* | *FF* | *FF* |
| *0x20108* | *08* | *00* | *19* | *91* |
| *0x2010C* | *20* | *40* | *60* | *80* |
| *0x20110* | *02* | *00* | *AB* | *0F* |
| *0x20114* | *30* | *99* | *11* | *55* |
| *0x20118* | *40* | *AA* | *7C* | *EE* |
| *0x2011C* | *FF* | *BB* | *42* | *D2* |
| *0x20120* | *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 7* ***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).*

*MOV DL, 0xFE*

**Solution:** DL = **0xFE**

*MOV DH, AL*

**Solution:** DH = AL = **0x70** (EDX now = 0x000070FE)

*MOVSX BX, BYTE PTR [ESI+0x000F]*

**Solution:** BX = sign-extended byte at address ESI+0x000F = 0x00020100 + 0x000F = 0x0002010F

→ BX = 0x80 sign-extended = **0xFF80**

*MOV [EDI+ECX], EBX*

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

EDI+ECX = 0x00020110 + 0x00000002 = 0x00020112

→ (0x20112) = EBX = **0x0000FF80** (bytes ordered as 0x80, 0xFF, 0x00, 0x00)

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

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

ESI + 4*ECX = 0x20100 + 4 * 2 = 0x20108

→ (0x20108) = **0x3170** (bytes ordered as 0x70, 0x31)

*XCHG CL, [ESI]*

**Solution:** Swap byte values in CL, address 0x20110 → CL = **0x10**, (0x20110) = **0x02**

*MOVZX EAX, WORD PTR [EDI+ECX]*

**Solution:** EAX = zero-extended word at address EDI+ECX = 0x20110 + 0x00000010 = 0x20120

→ EAX = **0x0000CC30** (original word underlined)

*MOV DX, [EDI+0xFFFFFFFA]*

**Solution:** DX = word at address EDI+0xFFFFFFFA = 0x20110 + (-6) = 0x2010A

→ DX = **0x9119**

*LEA ECX, [ESI+EBX+0x0017]*

**Solution:** ECX = ESI + EBX + 0x0017h = 0x20100 + 0x0000FF80 + 0x0017h = **0x30097**

*MOVSX EBX, BYTE PTR [ESI+4]*

**Solution:** EBX = sign-extended byte at address 0x20104h = **0x00000010** (original byte underlined)

*2. (80 points) 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 **8 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*

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

*DIV DH*

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

*NEG AH*

**<u>Solution:</u>** AH = -AH = -0x0C = -($0000\ 1100_2$) = ($1111\ 0011_2$ + 1 = $1111\ 0100_2$ = **0xF4h**