EECE.3170: Microprocessor Systems Design I

Summer 2017

Homework 2

Due **1:00 PM, Monday, 5/22/17**

Notes:

- While typed solutions are preferred, handwritten solutions are acceptable.
- Any electronic submission must be in a single file. Archive files will not be accepted.
- Electronic submissions should be e-mailed to Dr. Geiger at Michael_Geiger@uml.edu. Please include your name as part of your filename (for example, mgeiger_hw2.pdf).
- This assignment is worth 150 points.

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
MOV     DH, AL
MOVSX   BX, BYTE PTR [ESI+0x000F]
MOV     [EDI+ECX], EBX
MOV     [ESI+4*ECX], AX
XCHG    CL, [ESI]
MOVZX   EAX, WORD PTR [EDI+ECX]
MOV     DX, [EDI+0xFFFFFFFA]
LEA     ECX, [ESI+EBX+0x0017]
MOVSX   EBX, BYTE PTR [ESI+4]
```

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
ADC     EAX, ECX
INC     WORD PTR [ESI]
MUL     BYTE PTR [ESI+4]
SUB     AX, [ESI+8]
DEC     AH
IMUL    AH
IDIV    DL
DIV     DH
NEG     AH
```