16.317: Microprocessor Systems Design I

Fall 2015

Homework 2 Solution

1. ~~(60 points)~~ ***(75 points)*** Assume the state of the x86 registers and memory are:

EAX: 00000010H
EBX: 00000020H
ECX: 00000030H
EDX: 00000040H
CF: 1
ESI: 00020100H
EDI: 00020100H

Address				
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	00	11	55
20118H	40	00	7C	EE
2011CH	FF	00	42	D2
20120H	30	00	30	90

What is the result produced in the destination operand by each of the instructions listed below? Assume that the instructions execute in sequence.

ADD AX, 00FFH

Solution: AX = AX + 00FFH = 0010H + 00FFH = **010FH, CF = 0**

ADC CX, AX

Solution: CX = CX + AX + CF = 0030H + 010FH + 0 = **013FH, CF = 0**

INC BYTE PTR [20100H]

Solution: Increment byte at address 20100

→ (byte at 20100h) = 10 + 1 = **11H**

SUB DL, BL

Solution: DL = DL – BL = 40H – 20H = **20H, CF = 0**

SBB DL, [20114H]

Solution: DL = DL – (byte at 20114h) – CF = 20H – 30H – 0 = **F0H**

1 (cont.)

DEC BYTE PTR [EDI+EBX]

Solution: Decrement byte at address EDI+EBX = 00020100h+00000020h = 00020120h

→ (byte at 20120h) = 30H – 1 = **2FH**

NEG BYTE PTR [EDI+0018H]

Solution: Negate byte at address 00020100h + 0018h = 00020118h

→ (byte at 20118h) = -40H = -0100 0000_2 = 1011 1111_2 + 1 = 1100 0000_2 = **C0H**

MUL DX

Solution: (DX,AX) = DX * AX (unsigned multiplication of 16-bit values)

→ (DX,AX) = 00F0 * 010F = 240 * 271 = 65040 = 0000FE10H

→ DX = upper 16 bits of result = 0000H; AX = lower 16 bits of result = FE10H

IMUL BYTE PTR [ESI+0006h]

Solution: AX = AL * byte at address ESI+0006h (signed multiplication of 8-bit values)

→ Address = 00020100h + 0006h = 20106H; byte = FF

→AX = 10H * FFH = 16 * -1 (must account for signs!) = -16

→-16 = -(10H) = -(0000 0000 0001 0000_2) = 1111 1111 1110 1111_2 + 1

= 1111 1111 1111 0000_2 = **FFF0H**

DIV BYTE PTR [ESI+0008h]

Solution: Divide AX by byte at address ESI+0008h; AL = quotient, AH = remainder (unsigned integer division of 8-bit values)

→ Address = 00020100h + 0008h = 20108h; byte = 08H

→ Dividing FFF0 / 08H = 65520 / 8 = 8190 R0 = 1FFEH R0 (keep only lowest byte)

→ AL = FEH, AH = 00H

IDIV BYTE PTR[ESI+0010H]

Solution: Divide AX by byte at address ESI+0010h; AL = quotient, AH = remainder (signed integer division of 8-bit values)

→ Address = 00020100 + 0010h = 20110h; byte = 02

→ Dividing 00FEH / 02H = 254 / 2 = 127 R0

→ AL = 7FH, AH = 00H

2. ~~(40 points)~~ ***(25 points, +1 extra credit point per correct instruction)*** Assume the state of the 80386DX's registers and memory are:

EAX: 00005555H
EBX: 00045010H
ECX: 00000010H
EDX: 0000AAAAH
ESI: 000000F2H
EDI: 00000200H

Address				
45100H	0F	F0	00	FF
	…			
45200H	30	00	19	91
	…			
45210H	AA	AA	AB	0F
	…			
45220H	55	55	7C	EE
	…			
45300H	AA	55	30	90

What is the result produced in the destination operand by each of the instructions listed below? Assume that the instructions execute in sequence.

AND BYTE PTR [45300H], 0FH

Solution: Byte at address 45300h = Byte at 45300H AND 0FH

→ (45300h) = AAH AND 0FH = **0AH**

SAR DX, 8

Solution: DX = DX >> 8 (arithmetic shift)

→ DX = AAAAH >> 8 = **FFAAH**; CF = last bit shifted out = **1**

OR [EBX+EDI], AX

Solution: Word at address EBX+EDI = Word at EBX+EDI OR AX

→ EBX+EDI = 00045010h+00000200h = 45210H

→ (45210H) = AAAAH OR 5555H = **FFFFH**

SHL AX, 2

Solution: AX = AX << 2

→ AX = 5555H << 2 bits = 0101 0101 0101 0101 << 2

→ AX = 0101 0101 0101 0100 = **5554H**, CF = last bit rotated out = **1**

2 (cont.)
XOR AX, [ESI+EBX]

<u>Solution:</u> AX = AX XOR (Word at address ESI+EBX)

→ ESI+EBX = 000000F2h + 00045010h = 45102H

→ AX = 5554H XOR FF00H = **AA54H**

NOT BYTE PTR [45300H]

<u>Solution:</u> Flip all bits of byte at 45300h

→ (45300H) = NOT 0AH = **F5H**

SHR AX, 4

<u>Solution:</u> AX = AX >> 4 (logical shift)

→ AX = AA54H >> 4 = 1010 1010 0101 <u>0100</u> >> 4

→ AX = 0000 1010 1010 0101 = **0AA5H**, CF = last bit rotated out = **0**