ECE 3724/CS 3124 Test #1 SOLUTIONS - Fall 2001 – Reese

1. (4 pts) Convert the value -53 to its 8-bit 2's complement value.

Magnitude of 53 in hex is 15H. Since we are converting a negative number, take 2's complement to get answer = CBh

2. 2. (10 pts) For each HEX sum below, give the 8-bit SUM, and the values of the indicated flags

34H + 6AH = _____9E_____ C = _0__ V = __1__ S = _1___ Z = __0__
sign flag set because result MSB =1, V set because positive+positive resulted in negative value.

FEH + 02H = ____00_____ C = _1__ V = __0__ S = __0___ Z = __1__
Carry flag set because of carry out of MSB, zero flag set because result is zero.

3. (12 pts) Assume the following register and memory contents, all values in hex.

```
0BC5:0000  1F EB 02 0E 1F 8B F7 5D-5F 07 58 C3 56 53 33 F6   .......]_.X.VS3.
0BC5:0010  33 C9 80 FE FF 75 08 8B-BC 90 91 8B DF EB 25 F6   3....u........%.
0BC5:0020  C6 02 74 08 C4 BC 74 91-8C C3 EB 18 3D 13 00 72   ..t...t.....=..r
0BC5:0030  0D 3D 27 00 77 08 C4 BC-80 91 8C C3 EB 06 C4 BC   .='.w...........
0BC5:0040  64 91 8C C3 83 FB FF 75-16 80 FE FF 75 03 F9 EB   d......u....u...
0BC5:0050  0C A3 AC 91 B8 FF FF BD-01 00 33 F6 F8 EB 0C 83   ..........3.....
0BC5:0060  FB 00 74 03 E8 54 00 83-C6 04 F8 72 06 0B C9 75   ..t..T.....r...u
0BC5:0070  02 EB 9F 9C 80 FE 01 75-29 52 55 51 06 57 50 B8   .......u)RUQ.WP.
```

CS: 0BC5, DS: 0BC4 SS: 0BC6, AX = 8391 BX :0028 , CX = 9040 DX = 85A3 BP: 0051, SP= 005E

What is value of the affected register after each of the following memory read? Give all values in hex.

a. mov ax, [bx +4]
ax = M[DS:BX+4] = M[0BC4:0028+4] = M[0BC4:002C] = M[0BC5:001C] = EBDF

b. mov cx, [bp - 2]
cx = M[SS:BP-2] = M[0BC6:0051-2] = M[0BC6:004F] = M[0BC5:005F] = FB83

c. mov ax, bx
ax = bx = 0028 (this is not a memory read)

4. (5 pts) If AX = 02FE , BX = 0A05 what is the value of AX after the following operation?

imul bl

*signed multiply, AX = al * bl = FEh * 05 = -2 * 5 = -10 = FFF6 h*

5. (5 pts) If AX = FFF1 , BX = 00FD, what is the value of AX after the following operation (remember that quotient goes in AL, remainder in AH).

```
idiv    bl
```

Signed divide, AX/BL = FFF1/FD = -15/-3 = quotient = 5, rmdr = 0, so AX = 0005

6. (5 pts) For the following instruction:

```
div  cl
```

provide values for AX, CX that will cause divide overflow.

Overflow occurs when quotient cannot fit in 8 bits. Operation is AX/CL, so AX=FFFF CX = 0001

7. (10 pts) Mark each of the branches in the following code sequences as taken or not taken assuming the following register contents.

CS: 0BC5, DS: 0BC4 SS: 0BC6, AX = 8391 BX :0028 , CX = 9040 DX = 85A3 BP: 0051, SP= 005E

a.
```
cmp  ax, bx
ja   THERE
```
TAKEN NOT TAKEN

unsigned compared, ax is larger in magnitude than BX, branch taken

b.
```
cmp  cx,bx
jl   THERE
```
TAKEN NOT TAKEN

signed compared, cx is negative, bx is positive, so cx is less than bx, branch taken.

c.
```
test  ax, 1
jz    THERE
```
TAKEN **NOT TAKEN**

result is ax and'ed with 1, if LSB of AX is zero, then result is zero. LSB of AX = 1, so branch not taken.

d.
```
and  cl, 0Fh
jz   THERE
```
TAKEN NOT TAKEN

cl = cl OR 0fH = 40 OR 0Fh = 00h , result is ZERO, so branch taken.

e.
```
cmp  bl, al
jb   THERE
```
TAKEN NOT TAKEN

unsigned compare, BL is lower in magnitude than AL, so branch taken.

8. (6 pts) Assuming the LSB (rightmost) is numbered as b0, and the MSB (leftmost) is numbered as b7, let AL = $b_7b_6b_5b_4b_3b_2b_1b_0$. Write an instruction sequence of no more than 4 instructions that will leave the new value of AL as : $100000b_5b_4$.

one possible solution:

```
shr     al,4    move bit5,b4 to lower 2 bits (0000 b7b6b5b4 )
and     al,3    mask upper 6 bits to zero (000000 b5b4 )
or      al,80h  set MSB = 1. (100000 b5b4 )
```

9. (6 pts) Assuming the register contents in Problem #3, what are the new values of AX, SP if the following instruction is executed:

```
Pop    AX
```

Ax = M[SS:SP], M[0BC6:005E] = M[0BC5:006E] = 75C9
SP = SP +2, SP = 005E +2 = 0060

10. (5 pts) The LOOP instruction is a combination of two operations. What two instruction sequence could be use to replace the instruction:

```
LOOP    THERE
```

LOOP is equivalent to:

```
Dec     cx
Jnz     THERE
```

11. (6 pts) Write an instruction sequence that will add the perform the 32-bit addition:

AX:BX = AX:BX + CX:DX

where AX:BX is one 32-bit number and CX:DX is another 32 bit number. You can only use 16-bit registers and operations.

```
Add     bx,dx
Adc     ax,cx
```

12. (5 pts) The following subroutine adds ax to bx and returns the answer in ax. However, there is a functional problem with this subroutine, what is it?

```
Mysub   Proc    near
        push    cx
        Push    dx
        Add     ax,bx
        Pop     dx
        Ret
Mysub   endp
```

CX, DX is pushed on stack in procedure, but CX is NOT popped off stack before return. This means that when the RET instruction is executed, the return address that get popped of the stack is actually the CX value, which will cause the procedure to return to the wrong location.

13. (5pts). When the loop is exited (reach HERE), what is the value of AL?

```
                xor     al, al
THERE           inc     al
                Jmp     THERE
HERE            …..
```

The JMP instruction is an unconditional jump. This means that the loop is NEVER exited!
Yes, this was a trick question.

14. (5 pts) When the loop is exited (reach HERE), , what is the value of AL?

```
                xor     al, al
THERE           inc     al
                js      THERE
HERE            ….
```

The XOR AL,AL instruction sets AL = 0. When AL is incremented the first time, its value goes from 0 to 1. This is a positive result, which means the JS (jump on signed) is NOT TAKEN. The value of AL when the loop is exited is 01.

15. (5 pts) When the loop is exited (reach HERE),, what is the value of AL?

```
                xor     al, al
THERE           inc     al
                jnz     THERE
HERE            …..
```

The XOR AL,AL instruction sets AL = 0. The first time through the loop AL = 01. This is a non-zero result, so the branch is taken. AL keeps incrementing, when it reaches FFH the next time it is incremented it rolls over to 00H (because of 8 bit increment). This result is zero, so branch is not taken, value of AL when loop is exited is 00H.

16. (3 pts) Give a value for AL that will give a different final result if "shr al, 1" is executed versus "sar al, 1".

Any value where the MSB of AL =1 (8xH thru Fx H) will cause a different result because the sign bit will be shifted by SAR (arithmetic shift right) and a '0' bit will be shifted in by SHR (logical shift right).

16. (3 pts). How many WORDS (2 bytes = 1 word) are pushed on the stack for a FAR procedure call?

Both CS, IP are pushed on stack for FAR call, so 2 words are pushed on stack.