Homework #4
Due: 07/14/2003

Problem 1) Consider the following fragment of 8088 assembly language code:

```
mov cx, 10
xor ax, ax
mov si, 0
label:
    mov stuff[si], ax
    add si, 3
    shl ax
    inc ax
    loop label
```

1a) (20 Points) How many clock cycles does this sequence of instructions take to execute on an 8088 processor? Assume that DS is loaded with the correct segment address, so that no segment overrides are required to access the array ‘stuff’.

1b) (5 Points) Assuming that the clock frequency of the machine is 4.77 Mhz, how long (e.g. how many microseconds) will the above sequence of instructions take to execute?

Problem 2) When the assembler encodes the instruction: MOV FOO[BP+SI], DH, The resulting machine code will occupy 4 bytes of memory. The bytes will be as follows:

[opcode] [mod reg r/m] [low byte of offset of VAR] [high byte of offset of VAR]

2a) (5 Points) What will be the value encoded by the assembler for the opcode byte (i.e., what will be the value (in hex) of the first byte of the encoded instruction)?

2b) (10 Points) What will be the value encoded by the assembler for the [mod reg r/m] byte (i.e., what will be the value (in hex) of the second byte of the encoded instruction)?

2c) (5 Points) If the memory location for the label “FOO” is 3C90h, what will be the complete encoding for the above instruction in hex?
Problem 3) (15 Points) The following shows three different instructions which will set the DX register to 0. What is the difference in code size and execution time of the two instructions on an 8088 processor?

xor dx,dx
mov dx,0
sub dx,dx

Problem 4) (40 Points) Write a procedure in assembly language to count the number of times that the DOS Function Calls (INT 21h) are performed. You are to “hook” into the existing interrupt so that the operation of the DOS function call occurs prior to your update of the count. Include the lines of code at the beginning of the main routine required to modify the interrupt vector table to point to your procedure, and at the end of the main routine for restoring the interrupt vector table to its original status.

You may assume the following:

- There are memory locations labeled “fcncount” where the total may be stored, “intcs” and “intoff” where the code segment and offset of the original interrupt service routine may be stored.
- INT 21h will not be called more than $2^{16}-1$ times by the main routine.
- Calls to INT 21h for the getting/setting of the interrupt vector table at the beginning and end of the main routine may be ignored in the count.