COSC 2329- **EXAM #2 Review**

You may use your yellow reference sheet of Intel opcodes, and any notations you may have put on it. No other resources are permitted.

1. Be able to draw all the logic gates.
2. Be able to show how a full adder can be constructed from logic gates.
3. Draw the gate diagram for an RS flipflop.
4. Draw the gate diagram for a D flip-flop.
5. Show how to implement a 4-bit ripple-carry adder from four 1-bit full adders
6. Suppose a program contains the lines

**call proc1**

**mov ax,bx**

and: (a) instruction **mov ax,bx** is stored at **7300**h; (b) **proc1** is a procedure that begins at **address ABCh**; (c) **sp** = **00F2**h. What are the contents of **ip** and **sp** just after **call proc1** is executed? What word is on top of the stack?

1. Translate the following into Intel assembly language:
   1. a = b \* c
   2. d = e/f; g = e % f;
   3. h = i – j\*k/m;
2. Write a section of code that will read a character from the keyboard, and test it to see if it is a lowercase letter. If it is, convert it to uppercase and display it. Otherwise, display it unchanged.
3. Write a section of code that will read an octal number from the keyboard and store its value in **bx**.
4. Be able to show how procedures are declared and documented, and how the call and ret instructions work.
5. Show how one and two dimensional arrays are stored.
6. Know how the following addressing modes work: register indirect, based, indexed, and based-indexed.
7. Show how the various addressing modes work with arrays.
8. Be familiar with the basic string instructions (movsb,movsw,stosb,stosw,lodsb,lodsw,scasb,scasw,cmpsb,cmpsw).
9. Show how the string instructions may be used to perform basic word-processing tasks such as search and replace.
10. For each of the following instructions show what happens if:

**ax** contains 0500h, **bx** contains 1000h, **si** contains 1500h, **di** contains 2000h

[1000h ] = 0100h [1500h] = 0300h [2000h] = 0600h [3000h] = 0200h [4000h] = 0400h  
 and **kats** is a word variable whose offset address is 1000h.

* 1. **mov** **di**, **si**
  2. **mov ax,[si]**
  3. **lea si,[bx+kats]**
  4. **mov [si],[di]**
  5. **mov ah,[bx]**
  6. **mov dh,[si]**
  7. **mov** **cx,[bx+di+kats]**

1. Assuming the following declarations write code which will sum in ax all the values in the 20 word array A using register-indirect addressing mode:

A resw 20

1. Suppose the following declarations have been made. Write instructions to append string1 to the end of string2, producing the string "ABCDEFGHIJ".

string1 db "FGHIJ"

string2 db "ABCDE"

db “ “2

1. An ASCIIZ string is a string that ends with a 0 byte; for example

str db "THIS IS AN ASCIIZ STRING",0

Write a procedure **length** that receives the address of an ASCIIZ string in **dx**, and returns its length in **cx**.

1. Suppose **si** contains 0100h, **di** contains 200h, **ax** contains 4142h, **df** = 0, byte 100h contains 10h, byte 101h contains 15h, byte 200h contains 20h, and byte 201h contains 25h. Give the source, destination, and value moved for each of the following instructions. Also give the new contents of **si** and **di**.
   1. **movsb** source = \_\_\_\_\_\_\_\_ dest = \_\_\_\_\_\_\_\_ val. moved = \_\_\_\_\_\_\_\_   
       **si** = \_\_\_\_\_\_\_\_ **di** = \_\_\_\_\_\_\_\_
   2. **movsw** source = \_\_\_\_\_\_\_\_ dest = \_\_\_\_\_\_\_\_ val. moved = \_\_\_\_\_\_\_\_   
       **si** = \_\_\_\_\_\_\_\_ **di** = \_\_\_\_\_\_\_\_
   3. **stosb** source = \_\_\_\_\_\_\_\_ dest = \_\_\_\_\_\_\_\_ val. moved = \_\_\_\_\_\_\_\_   
       **si** = \_\_\_\_\_\_\_\_ **di** = \_\_\_\_\_\_\_\_
   4. **lodsb** source = \_\_\_\_\_\_\_\_ dest = \_\_\_\_\_\_\_\_ val. moved = \_\_\_\_\_\_\_\_   
       **si** = \_\_\_\_\_\_\_\_ **di** = \_\_\_\_\_\_\_\_
   5. **lodsw** source = \_\_\_\_\_\_\_\_ dest = \_\_\_\_\_\_\_\_ val. moved = \_\_\_\_\_\_\_\_   
       **si** = \_\_\_\_\_\_\_\_ **di** = \_\_\_\_\_\_\_\_