EE 308 – Homework 3

1. (E4.1 from Textbook) Assuming that we have the following instruction sequence to be executed by the HCS12, what will be the contents of the topmost 4 bytes of the stack after the execution of these instructions?

```
        lds  $#1500
        ldaa #$56
        staa  1,-SP
        ldab  #22
        staa  1,-SP
        ldy   #0
        sty   2,-SP
```

2. (E4.10) from Textbook) Draw the stack frame and enter the value of each stack slot (if it is known) at the end of the following instruction sequence:

```
        leas   -2,sp   ; reserve 2 bytes for local variables
        clrb
        ldaa  #20
        psha
        ldaa  #$E0
        psha
        ldx   #$7000
        pshx
        jsr    sub_abc
        ...
sub_abc: pshd
        leas  -12,sp     ; reserve 12 bytes for local variables
        ...
```

3. The Dragon12 Plus has several ways to display data. It has an LCD display, four seven-segment LED displays, and eight individual LEDs. Programming the LCD display is rather complicated, and will not be discussed at this time. The easiest display to start with is the individual LEDs. You will write some programs to display patterns on the LEDs.

You can display patterns on the LEDs by writing to the I/O port at address 0x0001 (called PORTB). Because of the way the LEDs are connected, it is necessary to do some other setup of I/O ports as well. The first five instructions in the program below set up the MC9S12 hardware so that you can write patterns to the LEDs. (For now, we will give you the code you need to set up the I/O system properly. Later, you will learn how to do the setup yourself.) The following program will flash the LEDs:

Program 1 Demo program

```
PORTB   equ $_____      ; Port B data register
DDRB    equ $_____      ; Port B direction register
PTP     equ $_____      ; Port P data register
DDRP    equ $_____      ; Port P direction register
PTJ     equ $_____      ; Port J data register
DDRJ    equ $_____      ; Port J direction register

        bset  DDRP,#$0F   ; Make PPO-PP3 outputs
        bset  PTP,#$0F    ; Turn off seven-sed LEDs
        bset  DDRJ,#$02   ; Make PJ1 output
        bclr  PTJ,#$02    ; Turn on individual LEDs
        bset  DDRB,#$FF   ; Activate control lines for LEDs
        movb #$55,PORTB   ; Turn on every other LED
loop:   com   PORTB       ; Toggle LEDs
        bra loop          ; Repeat
```

Make sure you have the programs written and clearly thought out. You should put all your code starting at memory location 0x2000.

Part 1. Complete Program 1 by adding the necessary addresses and assembler directives. You can find the addresses for the ports in Section 1.5 of the MC9S12DP256B Device User Guide.

Part 2. Write a program which will start with all the LEDs off, then increment the LEDs, implementing a binary counter.

Part 3. Write a program to determine the largest and smallest 16-bit number in memory locations 0x8000 to 0x80ff. Store the maximum in address 0x1000, and the minimum in address 0x1002. Treat the numbers as signed.

Part 4. Write a program which puts the exclusive OR of the eight-bit numbers from memory locations 0x8000 through 0x8FFF and display the result on the LEDs. (This operation is often used to generate a check sum to verify data transmission. The sending computer generates and transmits the check sum along with the data. The receiving computer calculates the check sum for the received data and compares it with the check sum sent by the sending computer. If the two values do not match, then there was an error in the transmission.)

4. An 9S12 executes the following set of instructions:

```
        org     $2000
        lds     #$1500
        ldaa    #56
        staa    1,-SP
        ldab    #94
        stab    1,-SP
        ldx     2,SP+
        jsr     sub1
        ...

sub1:   pshb
        psha
        ldy     0,SP
        ...
```

Show the value of A, B, X, Y, and SP registers (in Hex), as much as you can tell, after the 9S12 executes the set of instructions shown above.

A	
B	
X	
Y	
SP	