Homework 2
(Due date: October 9th @ 5:30 pm)
Presentation and clarity are very important! Show your procedure!

PROBLEM 1 (10 pts)
- Given a 24 MHz bus clock, provide a set of instructions to generate:
  ✓ A time delay of 40 ms.
  ✓ A time delay of 60 ms.

PROBLEM 2 (30 pts)
- The following directives store a bunch of numbers in memory that represent degrees in Celsius.
- Complete the program (provide a printout) that converts those numbers to Fahrenheit degrees. Use a subroutine for the Celsius to Fahrenheit conversion \(F = \frac{C}{5} + 32\). Use a loop to convert every number in the array. Store the result in the arrayF array.

```assembly
; Include derivative-specific definitions
INCLUDE 'derivative.inc'

ROMStart EQU $4000 ;
N EQU 10

; variable/data section
ORG RAMStart ; Originate data at address

ROMStart
; variables definition:
arrayC dc.w -21, 32, 45, 1120, 41, 13, -39, 100, 123, 0;
arrayF ds.w N;

; For the division by 5, only consider the integer part of the division.
Also, keep in mind that the input array contains 16-bit signed numbers.
```

PROBLEM 3 (20 pts)
- For the following code snippets, complete the value of the register when the last instruction is executed:

```
<table>
<thead>
<tr>
<th>Instruction</th>
<th>Instruction</th>
<th>Instruction</th>
<th>Instruction</th>
</tr>
</thead>
<tbody>
<tr>
<td>ldaa $58C</td>
<td>ldaa $59</td>
<td>ldd $F3FE</td>
<td>movw $F40FF, $F1</td>
</tr>
<tr>
<td>idx $04</td>
<td>brs $0F,$3A,next</td>
<td>bmi next</td>
<td>ldd $7122</td>
</tr>
<tr>
<td>loop: asra</td>
<td>inc $F0</td>
<td>add $F10F0</td>
<td>add $F next</td>
</tr>
<tr>
<td></td>
<td>next: asr $F0</td>
<td>bvs next</td>
<td>bvs next</td>
</tr>
<tr>
<td></td>
<td>ldd $F0</td>
<td>inca</td>
<td>inca</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

A = A = D = D =
```

- For the following code snippets, specify a value of B that makes the branch instruction branch to 'next':

```
<table>
<thead>
<tr>
<th>Instruction</th>
<th>Instruction</th>
<th>Instruction</th>
<th>Instruction</th>
</tr>
</thead>
<tbody>
<tr>
<td>ldb $34</td>
<td>clc</td>
<td>ldb $34</td>
<td>add $F1</td>
</tr>
<tr>
<td>cmpb $F5</td>
<td>rolb</td>
<td>clc</td>
<td>bpl next</td>
</tr>
<tr>
<td>bhs next</td>
<td>bcs next</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

B = B = B = B =
```

PROBLEM 4 (20 pts)
- Create an Assembly program (provide a printout) that reads the DIP Switch of the Dragon12-Light Board and displays the hexadecimal value present on the 4 LSBs. Utilize the 4 MSBs of the DIP Switch to determine which 7-segment displays to display: Bit 7 (MSB) controls display 3 (rightmost), bit 6 controls the display 2, bit 5 controls display 1, bit 4 controls display 0 (leftmost).

**Examples:**
- If DIP Switch: 11001001, we display the character ‘9’ on the two rightmost 7-segment displays.
- If DIP Switch: 00011110, we display the character ‘E’ on the leftmost 7-segment display.
- If DIP Switch: 00001001, no character is displayed.
PROBLEM 5 (20 PTS)
- Given the following Assembly code, specify the SP and the Stack Contents at the given times (right after the colored instruction has been executed). SP and the Stack Contents (empty) are specified for the first instruction (LDS #$4000).

```
ROMStart EQU $4000
; code section
ORG ROMStart
Entry:
_Startup: LDS #$4000
mainLoop: movb #$F1,1,-SP
           movb #$A3,1,-SP
           movb #$97,1,-SP
           ldd #$FACE
           bsr myfun
           leas 3,SP
           forever: bra forever
           leas 3,SP
           ... ; Subroutine
           myfun:  psha
                   pshb
                   leas -2,SP;
                   movw #$568A, 0,SP
                   leas 2,SP;
                   pulb
                   pula
                   rts
```

SP: $4000

0x4000
0x4001
...