Solutions - 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.

<table>
<thead>
<tr>
<th>40 ms delay</th>
<th>60 ms delay</th>
</tr>
</thead>
<tbody>
<tr>
<td>$n \times n_{times} \times \frac{1}{24 \times 10^6} = 40 = n \times n_{times} = 960000$</td>
<td>$n \times n_{times} \times \frac{1}{24 \times 10^6} = 60 = n \times n_{times} = 1440000$</td>
</tr>
<tr>
<td>$n_{times} = 64000 &lt; 65535, n = 15$</td>
<td>$n_{times} = 57600 &lt; 65535, n = 25$</td>
</tr>
<tr>
<td>ldx #64000</td>
<td>ldx #57600</td>
</tr>
<tr>
<td>loop: psha ; 2 cycles</td>
<td>loop: psha ; 2 cycles</td>
</tr>
<tr>
<td>pula ; 3 cycles</td>
<td>pula ; 3 cycles</td>
</tr>
<tr>
<td>psha ; 2 cycles</td>
<td>psha ; 2 cycles</td>
</tr>
<tr>
<td>pula ; 3 cycles</td>
<td>pula ; 3 cycles</td>
</tr>
<tr>
<td>nop ; 1 cycle</td>
<td>psha ; 2 cycles</td>
</tr>
<tr>
<td>nop ; 1 cycle</td>
<td>pula ; 3 cycles</td>
</tr>
<tr>
<td>dbne X, loop ; 3 cycles</td>
<td>nop ; 1 cycle</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>

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 ($F = \frac{\text{C} \times 9}{5} + 32$). Use a loop to convert every number in the array. Store the result in the arrayF array.

; Include derivative-specific definitions
INCLUDE 'derivative.inc'
ROMStart  EQU $4000  ;
N EQU 10
; variable/data section
ORG ROMStart  ; Originate data at address
ROMStart
; variables definition:
arrayC  ds.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.

ASM Code: hw2p2.asm
**PROBLEM 3 (20 pts)**

- For the following code snippets, complete the value of the register when the last instruction is reached:

<table>
<thead>
<tr>
<th>ldac $8C</th>
<th>ldac $59</th>
<th>ldac $F3FE</th>
<th>movw $40FF,$F1</th>
</tr>
</thead>
<tbody>
<tr>
<td>ldx $04</td>
<td>staa $F0</td>
<td>cba $F1</td>
<td>ldd $7122</td>
</tr>
<tr>
<td>loop: asra</td>
<td>brset $F0,$3A,next</td>
<td>bmi next</td>
<td>addh $F1</td>
</tr>
<tr>
<td>inc $F0 -&gt; {$F0} --&gt; $5A</td>
<td>bmi next</td>
<td>r0</td>
<td>bvs next</td>
</tr>
<tr>
<td>next: asr $F0</td>
<td>decs</td>
<td>$F0</td>
<td>inca</td>
</tr>
<tr>
<td>ldd $F0</td>
<td>decb</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

  - A = $FA
  - B = $2D
  - D = $F2FD
  - D = $B220

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

<table>
<thead>
<tr>
<th>ldab #__</th>
<th>ldab #__</th>
<th>ldab #__</th>
<th>ldab #__</th>
<th>ldab #__</th>
<th>ldab #__</th>
<th>ldab #__</th>
<th>ldab #__</th>
</tr>
</thead>
<tbody>
<tr>
<td>cmpb $E8</td>
<td>stab $FF</td>
<td>clc</td>
<td>rolb</td>
<td>rolb</td>
<td>asrb</td>
<td>cmpb $EB</td>
<td>cmpl next</td>
</tr>
<tr>
<td>bhs next</td>
<td>dec $FF</td>
<td>bcs next</td>
<td>bcs next</td>
<td>bcs next</td>
<td>bcs next</td>
<td>bcs next</td>
<td></td>
</tr>
</tbody>
</table>

  - B = $EB
  - B = $01
  - B = $00
  - B = $F6
  - B = $0F
  - B = $7F
  - B = $45

**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.

```
DDRB ← $FF, DDRP ← $FF, DDRH ← $00,
PTP ← $00
X ← sevsegdata

while (1)
  A ← PORTH
  B ← A
  B ← B and $F0
  B ← not(B) ; Display is ON with a 0 on the cathode
  Shift B to the right 4 positions
  PTP ← B ; This controls the displays that will be turned on

  A ← A AND $50F ; We are only interested in the 4 LSBs
  B ← [X] + [A] |
  PORTB ← B; Display hexadecimal value only on the specified displays

end

ASM Code: hw2p4.asm
```
**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`).

- HCS12 processor: When `SP` is incremented (by pulling values out of the Stack, or by executing `leas`), the values on the memory positions that used to be part of the Stack are considered unknown (i.e., the values are replaced by random data).