(5) **Question 1.** The format is 8-bit signed. What is the hexadecimal representation of the value -60? First way using basis.
-128 needed -60 + 128 = 68, 64 needed 68-64 = 4, 4 needed. Binary is 11000200, which is $C4$. Second way, first calculate
+60 = 32 + 16 + 8 + 4, which in binary is 00111100. Next complement 11000111, then add one 11000100 = $C4$.

(5) **Question 2.** Which of the following techniques can be used to handle the problem of overflow?
- E) Implement ceiling and floor.
- G) Use promotion.

(5) **Question 3.** Consider the following two instructions
```assembly
ldab #250
subb #-2
```
To determine the overflow (V) bit, first convert both to signed -128 to +127
```assembly
ldab #=>-6
subb #=>-2
```
-6 – (-2) is -4, so V=0
To determine the carry (C) bit, first convert both to unsigned 0 to +255
```assembly
ldab #250
subb #254
```
250-254 is -4, so C=1

(10) **Question 4.** For the circuit, see Figure 2.17 (b). The desired operating point is 2.6V at 10 mA.
\[
R = \frac{5 - V_d - V_{OL}}{I_d} = \frac{5 - 2.6 - 0.4}{0.01} = 200\Omega
\]

(10) **Question 5.** $0008$ is pushed first, $4009$ is the return address. Both numbers are big endian
```assembly
$3FFC = $40
$3FFD = $09
$3FFE = $00
$3FFF = $08
```
Part b) The subroutine will be executed 4 times because Y is pulled, allowing the action caused by `dex`.

(5) **Question 6.** Fetch all machine bytes, then store D to memory. The effective address of 4,x is X+4.

<table>
<thead>
<tr>
<th>R/W</th>
<th>Addr</th>
<th>Data</th>
<th>Changes to D,X,Y,S,PC,IR,EAR</th>
</tr>
</thead>
<tbody>
<tr>
<td>R</td>
<td>$4000</td>
<td>$6C</td>
<td>IR = $6C, PC = $4001</td>
</tr>
<tr>
<td>R</td>
<td>$4001</td>
<td>$04</td>
<td>EAR = $2004, PC = $4002</td>
</tr>
<tr>
<td>W</td>
<td>$2004</td>
<td>$11</td>
<td>(RegD and RegX are not changed)</td>
</tr>
<tr>
<td>W</td>
<td>$2005</td>
<td>$22</td>
<td></td>
</tr>
</tbody>
</table>

(20) **Question 7.** Write an assembly code that waits until the switch at PP1 is pressed.
```assembly
Wait ldaa PTP
andda #0$02
beq Wait

Wait brclr PTP,#0$02,Wait
```

(20) **Question 8.** Write assembly code that increments all numbers of the buffer
```assembly
loop ldy 0,x
iny ;add 1 to value
sty 0,x
inx
inx
cpx #Buffer+200
bne loop
```
```assembly
loop ldy 0,x
iny ;add 1 to value
sty 2,x+
```
```assembly
dbne A,loop
```

(20) **Question 9.** If Reg A is greater than 100, turn on the LED at PP5
```assembly
LEDout cmpa #100
bss done
bset PTP,#20
done rts
```

Jonathan W. Valvano  October 1, 2010  10:00am-10:50am