(5) **Question 1.** Using the basis elements, \( \%11010100 = -128+64+16+4 = -128+84 = -44 \)

(5) **Question 2.** RAM is volatile (possibly one could say registers are volatile)

(5) **Question 3.** What will be the value of the carry (C) bit?
Step 1: make sure both numbers are in unsigned 8-bit format, they are in format already
Step 2: perform subtraction, \( 110 - 140 = -30 \)
Step 3: check to see if the result fits into 8-bit unsigned format, it doesn’t fit, so \( C=1 \)

(5) **Question 4.** What will be the value of the overflow (V) bit?
Step 1: make sure both numbers are in signed 8-bit format, they are in format already
Step 2: perform addition, \(-90 + -100 = -90-100 = -190\)
Step 3: check to see if the result fits into 8-bit signed format, it doesn’t fit, so \( V=1 \)

(5) **Question 5.** \( $C2 = \%1100,0010. \)
Part a) Shift right maintaining the sign bit. Register A=\( \%1110,0001 = \$E1 \)
Part b) The C bit contains the least significant bit that is shifted out, \( C=0 \)

(5) **Question 6.** The reset vector is a 16-bit address at \( \$FFE \).

<table>
<thead>
<tr>
<th>org</th>
<th>$FFFE</th>
</tr>
</thead>
<tbody>
<tr>
<td>fdb</td>
<td>$5000</td>
</tr>
</tbody>
</table>

(10) **Question 7.** The PC will be \( \$4212 \) after the opcode and operand are fetched. \( \$E0 \) will be sign-extended to \( \$FFE0 \). This means \(-32\). \( \$4212 + \$FFE0 = \$41F2 \)

| $4210 07E0 | bsr MyFunction |
| R/W Addr Data | Changes to A,B,X,Y,S,PC,IR,EAR |
| 1 $4210 $07 | IR=$07, PC=$4211 |
| 1 $4211 $E0 | PC=$4212 (return address) |
| 0 $3FF5 $12 | Push return address, SP=$3FF5 |
| 0 $3FF4 $42 | SP=$3FF4, PC=$41F2 |

(20) **Question 8.**
Part a) (friendly)

```assembly
bset DDRT,#$20   ; Set bit 5 DDRT so PT5 is an output
bclr DDRT,#$04   ; Clear bit 2 DDRT so PT2 is an input
```

or (friendly)

```assembly
ldaa DDRT
oraa #$20     ; Set bit 5 DDRT so PT5 is an output
anda #$FB     ; Clear bit 2 DDRT so PT2 is an input
staa DDRT
```

or (not friendly)

```assembly
ldaa #$20     ; Set bit 5 DDRT so PT5 is an output
staa DDRT     ; Clear bit 1 DDRT so PT2 is an input
```
Part b) (friendly)

```
brset PTT,#$04,next ; skip over if PT2==1
bset  PTT,#$20       ; set bit 5 of PTT so PT5=1
next
```

or (friendly)

```
ldaa PTT   ; read PTT
anda #$04  ; test bit 1, skip over if PT2==1
bne  next  ; not equal to zero, if PT2==1
ldaa #$20
ora  PTT   ; friendly
staa PTT   ; set bit 5 of PTT so PT5=1
next
```

or (not friendly)

```
ldaa PTT   ; read PTT
anda #$04  ; test bit 2, skip over if PT2==1
bne  next  ; not equal to zero, if PT2==1
ldaa #$20
staa PTT   ; set bit 5 of PTT so PT5=1
next
```

(20) Question 9.

;*****Sub1 subroutine****************
;Inputs: RegA is the first number
;        RegB is the second number
;Outputs: RegA is the difference of the first-second numbers
;         RegA is returned as 0 if an unsigned overflow occurs
Sub1 sba          ;RegA=RegA-RegB
 bcc done
 clra          ;unsigned overflow
done rts

(20) Question 10. This is an if-then-else structure.

```
ldd Input       ;bring in first number
cpd #-100        ;subtract second number
blt less         ;skip if Input < -100
movw #0,Output   ;Input >= -100
bra done
less  movw #200,Output ;Input < -100
done
```

Jonathan W. Valvano       September 26, 2008       10:00am-10:50am