(4) Question 1. The basis elements are $1000 = 27$, $0100 = 9$, $0010 = 3$, and $0001 = 1$.

(3) Question 2. Answer true/false for each of the following three statements:

Part a) False, Flash EEPROM memory on the 9S12 is nonvolatile.

Part b) True, the order in which I add the numbers does affect the final value of the carry bit.

Part c) True, dropout error can occur on a logical right shift (e.g., $lsra$). Overflow can occur.

(4) Question 3. Consider $ldab \#-100$ $subb \#50$

Convert to signed (done), Subtract two signed: $-100 - 50 = -150$. Does not fit, so $V=1$.

Convert to unsigned: $-100 = -100 + 256 = 156$. Subtract unsigned: $156 - 50 = 106$. This fits, so $C=0$.

(4) Question 4. What is the binary representation of 8-bit signed number -10?

Method 1) $+10$ is 8+2 or 00001010. Negative is 2's complement. Complement 1111,0101, then add 1. $11110110$

Method 2) Look at basis elements, need -128,64,32,16,4,2, so $11110110$

Method 3) -11 is the same binary as $-10 + 256 = 246$. 246/16=15 remainder 6. So hex is $SF6$

(20) Question 5. The current through LED resistor 25mA = $(5 - 2.5)/R$. Solve for $R = 2.5V/25mA = 100\Omega$.

The pull down resistor on the switch could be 10k$\Omega$ or 100k$\Omega$. I will even count 1k$\Omega$ or 1M$\Omega$.

![9S12 circuit diagram]

(5) Question 6. The bus cycles occurring for $stx \$2000$

<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>$5200$</td>
<td>$7E$</td>
<td>PC=$5201$,IR=$7E$</td>
</tr>
<tr>
<td>R</td>
<td>$5201$</td>
<td>$20$</td>
<td>PC=$5202$</td>
</tr>
<tr>
<td>R</td>
<td>$5202$</td>
<td>$00$</td>
<td>PC=$5203$,EAR=$2000$</td>
</tr>
<tr>
<td>W</td>
<td>$2000$</td>
<td>$12$</td>
<td></td>
</tr>
<tr>
<td>W</td>
<td>$2001$</td>
<td>$34$</td>
<td></td>
</tr>
</tbody>
</table>

(20) Question 7. Mask the bits of interest, then compare.

; fastest execution

Check $ldaa$ PTT ;read all 8 bits

   anda #$45$ ;look at just bits 6,2,0
   cmpa #$01$ ;expected value
   bne done
   bset PTT,$#80$ ;PT0=1, PT2=0, and PT6=0 so make PT7=1

done rts

; simple to understand

Check $ldaa$ PTT ;read all 8 bits

   bita #$44$ ;look at bits 6,2
   bne done ;skip if either PT6 or PT2 are 1
   bita #$01$ ;look at bit 0
   beq done ;skip if PT0 is 0
   oraa #$80$ ;PT0=1, PT2=0, and PT6=0 so make PT7=1
   staa PTT

done rts

; fewest number of instructions

Check $brsset$ PTT,$#44$,done ;skip if either PT6 or PT2 are 1

   $brclr$ PTT,$#01$,done ;skip if PT0 is 0
   bset PTT,$#80$ ;PT0=1, PT2=0, and PT6=0 so make PT7=1

done rts
(20) Question 8. Write an assembly language subroutine that adds two unsigned 16-bit numbers.

```
: simple to understand
org $2000 ;RAM
yval rmb 2
org $4000
add sty yval ;save in variable
tfr x,d
addd yval ;add two inputs
bcc ok
ldd #65535 ;ceiling on overflow
ok rts
```

; uses stack, so no global is required
add pshy ;save Y on stack
tfr x,d
addd 2,sp+ ;add two inputs
bcc ok
ldd #65535 ;ceiling on overflow
ok rts

(20) Question 9. A subroutine that counts the number of binary bits that are zero.

```
: simple to understand
Count clrb ;result
ldx #8 ;loop counter
loop lsra ;bit into carry (could shift right or left)
 bcs skip
 incb ;found a zero
skip dbne x,loop
rts
```

; fastest to execute, does not require a loop counter
Count clrb ;result
coma ;will be counting 1’s now
loop bpl skip ;bit7=0, do not count
incb ;found a 1 (means found a 0)
lsla ;move bits into bit7
bne loop ;done when A=0
rts