Jonathan W. Valvano  
May 12, 2004, 9am - 12 noon  

<table>
<thead>
<tr>
<th>Question</th>
<th>Choice</th>
<th>Question</th>
<th>Choice</th>
<th>Question</th>
<th>Choice</th>
<th>Question</th>
<th>Choice</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Private in scope</td>
<td></td>
<td>unsigned long</td>
<td></td>
<td>ADC conversion result for channel 2</td>
<td>oscillates between state A and state B with the output low</td>
<td>clear all flag bits in the TFLG1 register. It is not friendly.</td>
</tr>
<tr>
<td>2.</td>
<td>E)</td>
<td>1000*100=100,000</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>After step 4</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3.</td>
<td>D)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Software crashes, interrupts are requested over and over.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4a.</td>
<td>E</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4b.</td>
<td>Minimum ( I_{CE} )</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>4.5V/50Ω = 90mA or 5V/50Ω = 100 mA</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6a.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>CPOL = 1 because clock is normally high</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6b.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>CPHA = 1 so it will sample input on rising edge</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>C code</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>// no overflow can occur</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>[ Z = (X*Y)/256; ]</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>[ (255*255)/256 = ]</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>[ 65025/256 = 254 ]</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>8.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>IEEE Code of Ethics</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>1. to accept responsibility in making engineering decisions consistent with the safety, health and welfare of the public, and to disclose promptly factors that might endanger the public or the environment;</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Example: If you build an embedded that fails, then you should responsibility for damages.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Example: If you deliver software with bugs, then you should implement a plan allowing customers to receive software patches to fix the errors.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
(4) Question 14. Yes or no
Yes

(4) Question 15. Yes or no
No

(1) Question 16a. synchronous or asynchronous
synchronous

(1) Question 16b. simplex, half-duplex, or full duplex
full duplex

(2) Question 16c. maximum bandwidth
8 bits/frame (both)

(2) Question 18. critical sections
yes, between LDAA time
LDAB time + 1

(16) Question 21.

```c
char *Pt; // 0 means idle
void Tx_Init(void){
    SC0BD = 208; // 500000/2400
    // baudRate = 2400 bits/sec
    SC0CR1 = 0x12;
    // M=1 start, PE=1, PT=0 even parity
    SC0CR2 = 0x08;
    // TE=1, enable transmitter
    Pt = (char *)0; // idle
    asm cli   /* enable interrupts */
}

short Tx_Out(char *data){
    if(Pt){
        return 1;  // busy
    }
    Pt = data;  // set pointer
    SC0CR2 |= 0x80;  // arm TDRE
    return 0;
}

interrupt 20 void SciHandler(void){
    char data;
    if(SC0SR1 & 0x80){  // TDRE?
        data = *Pt;
        if(data){
            Pt ++;
            SC0DRL = data;  // send, ack
        } else{
            Pt = (char *)0; // done
            SC0CR2 &= ~0x80;  // disarm
        }
    }
}
```

(2) Question 19a. maximum t_{su}
8 bits/frame (both)
2*1MHz/8 = 250k bytes/sec

(2) Question 19b. maximum t_{h}
500+t_{h}<500+[0,10]+20
$ t_{h} < 20$

(4) Question 17. Stack size
19 bytes

(2) Question 19. maximum t_{su}
106 < 500+[0,10]-t_{su}
t_{su} < 394

(2) Question 19b. maximum t_{h}
500+t_{h}<500+[0,10]+20
$t_{h} < 20$

(2) Question 20. Number of stretches
60+[100,200] < t_{l} \sim 30
290 < t_{l}
2 stretches

(4) Question 20. Number of stretches
2 stretches

(2) Question 22. Choose A-D
C) I/O bound, because spends time waiting for TDRE