UNCC, Department of Electrical and Computer Engineering, ECGR4101/5101/6090, Fall 2004, Homework #7, Due: 10/22/04, at the beginning of class (20 points)

0. How long did this homework take you? (1 point)

1. Read the Jack Ganssle article “Interrupt Latency.” In four lines of correct English, summarize the general objective of the article. (4 points)

2. How far apart in time can two interrupts be yet still be considered simultaneous if an M16C26 processor’s clock is 4 MHz? SHOW YOUR WORK. (1 point)

3. For the M16C/262, list the following interrupts in order of decreasing priority: single-step, NMI, watchdog timer, reset, ADC conversion complete, UART1 transmit, Timer B2. (1 point)

4. Assume an enabled interrupt has occurred. Fill in the contents of the stack and update the registers listed when the processor is about to begin executing the ISR at address 0x0ff440. Recall that the FLG register is modified when the processor responds to an interrupt. Assume the following processor state immediately before the interrupt occurred. (8 points)
   - SP = 0x200,
   - PC = 0xff312
   - FLG = 0x3062
   - Stack contents as shown in table below.

<table>
<thead>
<tr>
<th>Address</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>0x01f9</td>
<td></td>
</tr>
<tr>
<td>0x01fa</td>
<td></td>
</tr>
<tr>
<td>0x01fb</td>
<td></td>
</tr>
<tr>
<td>0x01fc</td>
<td></td>
</tr>
<tr>
<td>0x01fd</td>
<td></td>
</tr>
<tr>
<td>0x01fe</td>
<td></td>
</tr>
<tr>
<td>0x01ff</td>
<td></td>
</tr>
<tr>
<td>0x0200</td>
<td></td>
</tr>
<tr>
<td>0x0201</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Register</th>
<th>Initial Value</th>
<th>Final Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>SP</td>
<td>0x0200</td>
<td></td>
</tr>
<tr>
<td>PC</td>
<td>0x0ff312</td>
<td></td>
</tr>
<tr>
<td>FLG</td>
<td>0x3062</td>
<td></td>
</tr>
</tbody>
</table>

5. Correct the three problems with the following interrupt service routine declaration. (2 point)

```c
#pragma new_isr INTERRUPT
int new_isr(int n) {
    ...  
}
```
6. Modify the following code from sect30_26skp.inc to register the C function raise_shields() as the interrupt service routine for external interrupt 0. (3 points)

```assembly
.section vector ; variable vector table
.org VECTOR_ADR
.1word dummy_int ; vector 0 (BRK)
.org (VECTOR_ADR +16)
.1word dummy_int ; INT3
.1word dummy_int ; Reserved
.1word dummy_int ; Reserved
.1word dummy_int ; Reserved
.1word dummy_int ; INT5
.1word dummy_int ; INT5
.1word dummy_int ; UART2 Bus collision detection iic
.1word dummy_int ; DMA0 (for user)
.1word dummy_int ; DMA1 (for user)
.1word dummy_int ; Key-on wakeup (for user)
.1word dummy_int ; AD Converter (for user)
.1word dummy_int ; UART2 transmit/NACK
.1word dummy_int ; UART2 receive/ACK
.1word dummy_int ; UART0 transmit (for user)
.1word dummy_int ; UART0 receive
.1word dummy_int ; UART1 transmit (for user)
.1word dummy_int ; UART1 receive:
.1word dummy_int ; TIMER A0 (for user)
.1word dummy_int ; TIMER A1 (for user)
.1word dummy_int ; TIMER A2 (for user)
.1word dummy_int ; TIMER A3 (for user)
.1word dummy_int ; TIMER A4 (for user) (vector 25)
.1word dummy_int ; TIMER B0 (for user) (vector 26)
.1word dummy_int ; TIMER B1 (for user) (vector 27)
.1word dummy_int ; TIMER B2 (for user) (vector 28)
.1word dummy_int ; INTO (for user) (vector 29)
.1word dummy_int ; INT1 (for user) (vector 30)
.1word dummy_int ; Reserved
```