EXAM #3

- Per MSU Academic Operating Policy 10.08, you **MUST** turn your cell phone off – not on vibrate, but completely off. If you have a pager, you must also turn it off.
- You may use only the provided reference materials.
- Unless stated otherwise, all multi-byte data values are stored in *little-endian* ordering.
- Absolutely NO cheating is allowed. If you are caught in the attempt of, the act of, or the past action of academic dishonesty, you will receive the maximum punishment allowed by University policy.

Pinout (shaded pins are 5V tolerant):

Reference for problem #1:

<table>
<thead>
<tr>
<th>Flag Bit</th>
<th>Set by:</th>
<th>Cleared by:</th>
</tr>
</thead>
<tbody>
<tr>
<td>TRAPR (RCON&lt;15&gt;)</td>
<td>Trap conflict event</td>
<td>POR, BOR</td>
</tr>
<tr>
<td>JOPUWR (RCON&lt;14&gt;)</td>
<td>Illegal opcode or initialized W register access</td>
<td>POR, BOR</td>
</tr>
<tr>
<td>CM (RCON&lt;9&gt;)</td>
<td>Configuration Mismatch</td>
<td>POR, BOR</td>
</tr>
<tr>
<td>EXTR (RCON&lt;17&gt;)</td>
<td>MCLR# Reset</td>
<td>POR</td>
</tr>
<tr>
<td>SWR (RCON&lt;6&gt;)</td>
<td>reset instruction</td>
<td>POR, BOR</td>
</tr>
<tr>
<td>WDTO (RCON&lt;4&gt;)</td>
<td>WDT time-out</td>
<td>pwrsvy instruction, cliwait instruction, POR, BOR</td>
</tr>
<tr>
<td>SLEEP (RCON&lt;3&gt;)</td>
<td>pwrsvy $0 instruction</td>
<td>POR, BOR</td>
</tr>
<tr>
<td>IDLE (RCON&lt;2&gt;)</td>
<td>pwrsvy $1 instruction</td>
<td>POR, BOR</td>
</tr>
<tr>
<td>BOR (RCON&lt;1&gt;)</td>
<td>BOR</td>
<td>n/a</td>
</tr>
<tr>
<td>POR (RCON&lt;0&gt;)</td>
<td>POR</td>
<td>n/a</td>
</tr>
</tbody>
</table>

Note: All Reset flag bits may be set or cleared by the user software.

As a Mississippi State University student I will conduct myself with honor and integrity at all times. I will not lie, cheat, or steal, nor will I accept the actions of those who do.

Signature: ___________________________ Date: _______________

Solution

sol@msstate.edu
(3 points each) A PIC24 programmed with the following code. See the previous page for reference information. What is printed on the screen? Enter X is nothing was printed.

<table>
<thead>
<tr>
<th>Problem</th>
<th>Time Period</th>
<th>Event (occurs at the beginning of the time period)</th>
<th>Message printed on screen during this time period</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>0.0 – 1.0 sec.</td>
<td>Power applied.</td>
<td>X</td>
</tr>
<tr>
<td>2.</td>
<td>1.0 – 2.0 sec.</td>
<td></td>
<td>EF</td>
</tr>
<tr>
<td>3.</td>
<td>2.0 – 3.0 sec.</td>
<td>Master clear button pressed.</td>
<td>C</td>
</tr>
<tr>
<td>4.</td>
<td>3.0 – 4.0 sec.</td>
<td></td>
<td>D</td>
</tr>
</tbody>
</table>

```c
int main() {
    // Config code (not shown)
    _POR = 0;
    if (_POR) {
        outString("A");
    }
    if (_BOR) {
        outString("B");
        _BOR = 0;
    }
    if (_EXTR) {
        outString("C");
        _EXTR = 0;
        DELAY_MS(1200);
        RESET(); // Execute the reset assembly instruction.
    }
    if (_SWR) {
        outString("D");
        _SWR = 0;
    }
    DELAY_MS(1200);
    outString("E");
    DELAY_MS(500);
    outString("F");
    DELAY_MS(400);
    RESET(); // Execute the reset assembly instruction.
    outString("G");
}
```
(3 points each) A PIC24 programmed with the following code is powered on. What is printed on the screen, and at what times? Assume the watchdog timer is programmed to expire in 2 seconds. Enter X is nothing was printed.

<table>
<thead>
<tr>
<th>Problem</th>
<th>Time</th>
<th>Messages printed on screen during this time period</th>
<th>Explanation (not graded)</th>
</tr>
</thead>
<tbody>
<tr>
<td>5.</td>
<td>0 – 1 sec.</td>
<td>AB</td>
<td>Program start; Delay 1 finishes at 0.9s; WDT enabled</td>
</tr>
<tr>
<td>6.</td>
<td>1 – 2 sec.</td>
<td>C</td>
<td>Delay 2 finishes at 1.4s; CPU asleep</td>
</tr>
<tr>
<td>7.</td>
<td>2 – 3 sec.</td>
<td>D</td>
<td>WDT wakes up system at 2.9s</td>
</tr>
<tr>
<td>8.</td>
<td>3 – 4 sec.</td>
<td>X</td>
<td>System in Delay 3</td>
</tr>
<tr>
<td>9.</td>
<td>4 – 5 sec.</td>
<td>A</td>
<td>WDT resets program before Delay 3 finishes</td>
</tr>
<tr>
<td>10.</td>
<td>5 – 6 sec.</td>
<td>B</td>
<td>Delay 1 finishes at 5.8s; WDT enabled</td>
</tr>
<tr>
<td>11.</td>
<td>6 – 7 sec.</td>
<td>C</td>
<td>Delay 2 finishes at 6.3s; CPU asleep</td>
</tr>
</tbody>
</table>

```c
int main() {
    // Config code (not shown).
    _SWDTEN = 0;
    outString("A");
    DELAY_MS(900); // Delay 1.
    outString("B");
    _SWDTEN = 1;
    DELAY_MS(500); // Delay 2.
    outString("C");
    SLEEP();
    outString("D");
    DELAY_MS(3000); // Delay 3.
    outString("E");

    while (1) {
        _SWDTEN = 1;
    }
}
```
(4 points each) Use the following code snippet to answer questions 12-14.

```c
int main() {
    CONFIG_RB13_AS_DIG_OUTPUT();          // Line A
    _LATB13 = 1;                           // Line B
    DELAY_MS(1);                           // Line C

    while(1) {
        DELAY_MS(100);                     // Line D
    }
}
```

12. Assume Line A contains the code: `ENABLE_RB13_PULLUP();`
What line of code needs to be changed (A, B, C, or D) in order for the PIC to output Z (a high-impedance value) on pin RB13? If no lines need to change, your answer should be “no change”. If this cannot be done by changing a single line, your answer should be “not possible”.

   Line letter: A

   New line of code: `ENABLE_RB13_OPENDRAIN();`

13. Assume Line A contains the code: `ENABLE_RB13_OPENDRAIN();`
What line of code needs to be changed (A, B, C, or D) in order for the PIC to output 0 V on pin RB13? If no lines need to change, your answer should be “no change”. If this cannot be done by changing a single line, your answer should be “not possible”.

   Line letter: B

   New line of code: `_LATB13 = 0;`

14. Assume RB13 is connected to ground by a wire and Line A contains the code: `ENABLE_RB13_OPENDRAIN();`
What line of code needs to be changed (A, B, C, or D) in order for the PIC to output 3.3 V on pin RB13? If no lines need to change, your answer should be “no change”. If this cannot be done by changing a single line, your answer should be “not possible”.

   Line letter: not possible

   New line of code: not possible
15. (5 points) Given PR2 = 3, what is the time between timer interrupts, assuming Fcy = 10 MHz and T2CON = 0x8010? Show your work.

3.2 us. With PR2 = 3, convert 4 ticks to seconds given a prescale of 1:8 (bits 5:4 of 0x8010 are 01). So, $4\times8/10\times10^6 = 3.2\times10^{-6}$.
(3 points each) A pushbutton attached between RB13 and ground is pressed at 500 ms, released at 1500 ms, pressed at 2500 ms, and so on. Fill in the table below. X = nothing printed.

<table>
<thead>
<tr>
<th>Problem</th>
<th>Time</th>
<th>Messages printed on screen during this time period</th>
<th>Explanation (not graded)</th>
</tr>
</thead>
<tbody>
<tr>
<td>16.</td>
<td>0 – 1 sec.</td>
<td>BA</td>
<td>Both interrupt flags are initially set, but INT1 has a higher priority. CN ISR runs after INT1.</td>
</tr>
<tr>
<td>17.</td>
<td>1 – 2 sec.</td>
<td>A</td>
<td>CN responds to a rising edge.</td>
</tr>
<tr>
<td>18.</td>
<td>2 – 3 sec.</td>
<td>AB</td>
<td>Both interrupt flags are initially set, but CN has a higher priority. INT1 ISR runs after CN.</td>
</tr>
<tr>
<td>19.</td>
<td>3 – 4 sec.</td>
<td>A</td>
<td>CN is triggered by either edge.</td>
</tr>
<tr>
<td>20.</td>
<td>4 – 5 sec.</td>
<td>A</td>
<td>INT1 has priority of 0 so it is disabled; CN is triggered by either edge</td>
</tr>
<tr>
<td>21.</td>
<td>5 – 6 sec.</td>
<td>A</td>
<td>CN is triggered by either edge.</td>
</tr>
</tbody>
</table>

```c
void _ISR_CNInterrupt(void) {
    _CNIF = 0;
    outString("A");
    if (_CNIP < 5)
        _CNIP = _CNIP + 2;
}

void _ISR_INT1Interrupt(void) {
    _INT1IF = 0;
    outString("B");
    if (_INT1IP > 0)
        _INT1IP = _INT1IP - 1;
}

void main(void) {
    CONFIG_RB13_AS_DIG_INPUT();
    ENABLE_RB13_PULLUP();
    DELAY_US(1);
    ENABLE_RB13_CN_INTERRUPT();
    _CNIF = 0;
    _CNIE = 1;
    _CNIP = 1;
    CONFIG_INT1_TO_RP(RB13_RB);
    _INT1EP = 1; // Falling edge triggered.
    _INT1IF = 0;
    _INT1IE = 1;
    _INT1IP = 2;
}```
while (1) { // Do nothing.
}

(4 points each) Fill in the blanks in the timing diagram below, assuming a PIC is programmed with the following code.

```c
void _ISR _T2Interrupt(void) {
    _T2IF = 0;
    _LATB12 = !_LATB12;
    PR2 = PR2 + u16_1_ms;
}

void main(void) {
    // Config RB12 as a digital output.
    // Config Timer 2 to produce interrupts.
    u16_1_ms = msToU16Ticks(1, getTimerPrescale(T2CONbits));
    PR2 = u16_1_ms - 1;
    _LATB12 = 0;

    while (1) {
    }
}
```

This diagram is NOT drawn to scale.

22. First time interval: 1

23. Second time interval: 2

24. Last time interval: 3
(4 points each) In the program below, a pushbutton switch is connected to the PIC; it pressed from 125 ms to 175 ms, then pressed at 225 ms to 325 ms, but released at all other times. A timer interrupt is configured to call update_state() every 50 ms, with the first call occurring at 50 ms. Fill in the table; assume no switch bounce.

e_state_t e_state = STATE_A;
uint16_t u16_count = 0;

void update_state(void) {
  switch (e_state) {
    case STATE_A:
      ++u16_count;
      outString("A");
      if (PB_PRESSED() || u16_count == 2) {
        outString("Z");
        e_state = STATE_B;
      }
      break;

    case STATE_B:
      outString("B");
      if (PB_RELEASED()) {
        outString("Y");
        e_state = STATE_A;
        u16_count = 0;
      }
      break;
  }
}

<table>
<thead>
<tr>
<th>Problem</th>
<th>Time</th>
<th>Messages printed on screen during this time period</th>
</tr>
</thead>
<tbody>
<tr>
<td>25.</td>
<td>0 – 75 ms</td>
<td>A</td>
</tr>
<tr>
<td>26.</td>
<td>75 – 125 ms</td>
<td>AZ</td>
</tr>
<tr>
<td>27.</td>
<td>125 – 175 ms</td>
<td>B</td>
</tr>
<tr>
<td>28.</td>
<td>175 – 225 ms</td>
<td>BY</td>
</tr>
<tr>
<td>29.</td>
<td>225 – 275 ms</td>
<td>AZ</td>
</tr>
</tbody>
</table>