1. (10 points) Basic Microcontroller Concepts
   
a. When writing C programs that use interrupts, variables that are used both inside an interrupt service routine and in other parts of the program should be declared with the “volatile” keyword. For example: volatile int my_flag; Why is this necessary? Discuss what could potentially happen in a C program if the variable is not declared as volatile.

b. Data sent using an asynchronous serial data protocol like RS-232 is sent from a transmitter to a receiver without a separate clock signal as might be used with other protocols. Explain how the receiver is able to correctly interpret the bits in the incoming data stream without having a separate clock signal to refer to.
2. (10 points) **Binary Representation Systems.** Each C declaration of the variable \( x \) is initialized to a value in decimal. Show that value represented in hex using the appropriate size indicated by the variable type (e.g. 1-byte = 2 hex digits).

   a. \( \text{int } x = -2; \)

   b. \( \text{unsigned char } x = 23; \)

   c. Assuming the 12-bit shortened IEEE FP format, what is the decimal equivalent of the number:

   \[
   \begin{array}{c}
   1 \\
   10011 \\
   010110
   \end{array}
   \]

3. (12 points) **Circuit Interfacing.**

   Given two families of logic with the given Voltage levels.

<table>
<thead>
<tr>
<th></th>
<th>( V_{OH} )</th>
<th>( V_{OL} )</th>
<th>( V_{IH} )</th>
<th>( V_{IL} )</th>
</tr>
</thead>
<tbody>
<tr>
<td>Family 1</td>
<td>4.8 V</td>
<td>1.2 V</td>
<td>3.6 V</td>
<td>2.0 V</td>
</tr>
<tr>
<td>Family 2</td>
<td>4.2 V</td>
<td>0.2 V</td>
<td>3.2 V</td>
<td>0.8 V</td>
</tr>
</tbody>
</table>

   a. What is the noise margin high and low (NM\(_H\) and NM\(_L\)) for Family 1?

   \[
   \begin{array}{c|c|c|c|c|c}
   |          | \( |I_{OH}| \) | \( |I_{OL}| \) | \( |I_{IH}| \) | \( |I_{IL}| \) |
   |----------|------------|------------|------------|------------|
   | Family 1 | 360 \( \mu \)A | 10 mA      | 40 \( \mu \)A | 2 mA       |
   \end{array}
   \]

   b. According to the specifications above, how many other logic gate inputs can the output of a single gate drive successfully? Briefly justify your answer.
4. (17 points) Short Answer

a. A state machine transition function (i.e. to determine the next state) depends on
   i. Current state
   ii. Input value
   iii. Output values
   iv. i and ii
   v. ii and iii

b. The IEEE standard for floating point representation uses special code values to represent which of
   the following:
   i. 0
   ii. Infinity
   iii. PI
   iv. i and ii
   v. All of the above

c. Instructions are fetched from memory by using the contents of the PC (Program Counter) as the
   ___________ (address of the next instruction / actual instruction).

d. The MIPS instruction set uses jal to call a subroutine (i.e. function). It jumps to a new location by
   updating the ___________ (PC / $ra) register and also stores the return address into ___________
   ($ra / the stack)

e. To support nested subroutines, the return address should be ___________ (pushed / popped) onto
   the ___________ (stack / heap) before another subroutine call is made.

f. True / False: An exception is any event that causes a break in normal program execution.

g. An example of an exception is (circle all that apply):
   i. A subroutine call using ‘jal’
   ii. A HW interrupt
   iii. Error conditions such as divide-by-0 or illegal address
   iv. Asking for and receiving an extension on your assignment from your instructor

h. A Turing machine model consists of a controller, a read/write ‘head’, and a ___________ (finite /
   infinite) tape of squares that can contain symbols. If a problem is not computable on a Turing
   Machine, it ___________ (can / cannot / may ) be computable on a modern electronic computer.
5. (12 points) State Machines. The state diagram to the right is for a state machine with three states (S1, S2 and S3) and two inputs (A and B). Write the C code for the loop that implement the state changes for this machine. You don’t have to write any other parts of the program such as dealing with where the inputs come from and updating them each time through the loop. Assume the input variables are available for use inside the loop. The machines stays in its current state unless the input conditions are such to move it to another state. If only one variable is shown on a transition, that means the other variable can be of any value.

```c
int a, b;
int state = 1; // Use 1, 2 and 3 for states S1, S2 and S3

while (1) {

}
```
6. (10 points) Memory Usage. A student writes a C function to calculate the value of a variable raised to some power \( x^n \) where \( n \) is an integer greater than zero by using recursion.

```c
long power(int x, int n)
{
    if (n == 1)
        return(x);
    else {
        return(x * power(x, n-1));
    }
}
```

If the processor has 100 bytes available for the stack space, what is the largest exponent value (argument \( n \)) that can safely be used without potentially causing corruption of data in RAM? Assume `int` variables are two bytes and the arguments are passed to the function by pushing them on the stack. The processor also stores a 16-bit value for the program counter return address and any local variables on the stack each time the function is called. You can also assume the function return value is not passed back on the stack but is passed in registers.

7. (12 points) Counter/Timers. The Arduino Uno is to be used in a pulse width modulation (PWM) system with the following specifications:

- PWM pulse frequency = 40 Hz (period = 25msec)
- Pulse width can range from 2 msec to 8 msec

The Arduino’s 16-bit TIMER1 module will be used and the processor clock frequency is 16Mhz. In the PWM mode to be used, the 16-bit value in register OCR1A determines the period of the pulse (25 msec), and the 16-bit value in register OCR1B determines the pulse width (2 msec to 8 msec). Find a set of values for the following that will make the system work as specified above.

- Prescaler setting (1, 8, 64, 256 or 1024)
- The value for the OCR1A register that determines the pulse frequency.
- The value for the OCR1B register for generating the maximum width pulse (8 msec).
- The value for the OCR1B register for generating the minimum width pulse (2 msec).

Hint: Find the three counter values (period, max width, min width) that would have to be used if the prescaler was set to 1. Then if the values are too big for a 16-bit register that has a maximum value of 65536, adjust them by trying different prescaler values until they fit in 16 bits.
8. (17 points) **Assembly Programming and Instruction Sets.** Given the following C language code snippets provide an equivalent MIPS assembly implementation. You may assume that for each of your three responses the following common set of directives and initial instructions has been executed.

<table>
<thead>
<tr>
<th>C Initialization</th>
<th>Assembly Initialization</th>
</tr>
</thead>
<tbody>
<tr>
<td>int x, dat[10], z;</td>
<td>X: .space 4</td>
</tr>
<tr>
<td></td>
<td>DAT: .space 40</td>
</tr>
<tr>
<td></td>
<td>Z: .space 4</td>
</tr>
<tr>
<td></td>
<td>la $s0, X</td>
</tr>
<tr>
<td></td>
<td>la $s1, DAT</td>
</tr>
<tr>
<td></td>
<td>la $s2, Z</td>
</tr>
</tbody>
</table>

To implement the following code snippets in assembly you should only need the following instructions: **lw, sw, add, addi** (add-immediate), **slt, bne/beq** and **sll or mul**. Your first priority is to provide code that works, but then try to use as few instructions as possible.

<table>
<thead>
<tr>
<th>C Code</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>z = x + dat[0];</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>C Code</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>if (x &lt; 0)</td>
<td></td>
</tr>
<tr>
<td>++dat[1];</td>
<td></td>
</tr>
<tr>
<td>else</td>
<td></td>
</tr>
<tr>
<td>x = x - 1;</td>
<td></td>
</tr>
</tbody>
</table>