The practice final is one hour, ten minutes long, closed book, closed notes, calculators allowed. To receive full credit on a question show all your work. Incorrect answers may receive partial credit only if the work is shown. The exam must be turned in to the instructor before he leaves the classroom at the end of the exam period.

1. 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. 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. int \( x = -2; \)

b. unsigned char \( x = 23; \)

3. 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 (\( \text{NM}_H \) and \( \text{NM}_L \)) for Family 1?

|       | \( |I_{OH}| \) | \( |I_{OL}| \) | \( |I_{IH}| \) | \( |I_{IL}| \) |
|-------|----------|----------|----------|----------|
| Family 1 | 360 \( \mu A \) | 10 \( mA \) | 40 \( \mu A \) | 2 \( mA \) |

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

c. True / False: An exception is any event that causes a break in normal program execution.
5. **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. **Counter/Timers.** The Arduino Uno is to be used to generate *50% duty cycle squarewave pulses* on Port D, bit 3 with a frequency of 80Hz. The Arduino’s 16-bit TIMER1 module will be used and the processor clock frequency is 16Mhz. Each time TIMER1 generates an interrupt the ISR can change the output bit (0 to 1 or 1 to 0) as needed to create the signal. In the timer mode to be used, the 16-bit value in register OCR1A determines the period of the pulses.

a. 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.
   
   Hint: Find the counter value that would have to be used if the prescaler was set to 1. Then if the value is too big for a 16-bit register that has a maximum value of 65536, adjust them by trying different prescaler values until it fits in 16 bits.

b. Fill in the ISR below with code to generate the pulse.

```cpp
ISR(TIMER1_COMPA_vect)
{
    // Code to generate the pulse
}
```