ECE 590.03 PRACTICE Midterm Exam 1

This is a full length practice midterm exam. If you want to take it at exam pace, give yourself 75 minutes to take the entire test. Just like the real exam, each question has a point value. There are 75 points in the exam, so that you can pace yourself to average 1 point per minute (some parts will be faster, some slower).

Questions:

1. Combinatorial Logic: 15 pts
2. Sequential Logic: 10 pts
3. Finite State Machines: 10 pts
4. MIPS Assembly: 20 pts
5. Datapaths: 10 pts
6. Memory Hierarchy: 10 pts
Question 1: Combinatorial Logic [15 pts]

Given the following truth-table:

<table>
<thead>
<tr>
<th>a</th>
<th>b</th>
<th>c</th>
<th>x</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>

1. Write the sum-of-product formula

2. Simplify the formula

3. Write VHDL that implements the formula

```vhdl
entity q1 is
    port (
        a : in std_logic;
        b : in std_logic;
        c : in std_logic;
        x : out std_logic);
end q1;
architecture basic of q1 is
begin

end basic;
```
Question 2: Sequential Logic [10 pts]

Consider the following VHDL fragment, in which x, and y are inputs, z is an output, and there are two DFFs (a and b) whose d inputs are a_d and b_d respectively, and whose q outputs are a_q and b_q respectively:

\[
\begin{align*}
  b_d &\leq a_q \oplus x; \\
  a_d &\leq (b_q \land a_q) \oplus y; \\
  z  &\leq a_q \oplus b_q;
\end{align*}
\]

Complete the waveform below (assume that the DFFs are triggered by the rising edge of clk): Note that a_q is initially 1, and b_q is initially 0:

```plaintext
clk ______ ______ ______ ______

x ______ ______ ______ ______

y ______ ______ ______ ______

a_d ______

a_q ______

b_d ______

b_q ______

z ______
```
Question 3: FSMs [10 pts]
Draw a state machine diagram for a finite state machine which accepts a single bit input (either 0 or 1—you can just label each edge with 0 or 1). This state machine also has a single bit of output, which is initially 0.

- Whenever the FSM receives an input bit of 1 followed by anything other than two 0s (so 11, or 101) the output goes to 1.
- The output remains at 1 until 3 consecutive 0s are received by the FSM.
- Once three consecutive 0s are received by the FSM, the output returns to 0.
- The output now remains 0 until it again sees a 1 which is not followed (immediately) by two 0s.

Note that when the FSM sees a 1 not followed by two 0s, the output goes to 1 as soon as the situation is detected (e.g., on the second 1 of 11 or 101). Label each state with the bit it outputs. Be sure to indicate your start state (with an arrow to it from nowhere).
**Question 4: Asm Programming [20 pts]**

Write MIPS assembly for the following C function.

```c
int countMatching(int * ptr, int n) {
    int count = 0;
    while (n > 0) {
        int x = testFn(*ptr);
        if (x != 0) {
            count++;
        }
        ptr++;
        n--;
    }
    return count;
}
```
.globl countMatching
.ent countMatching
.text
countMatching:

.end countMatching
Question 5: Datapaths [10 pts]

Consider the following single-cycle data-path:

Fill in the following table of control signals to correctly control the datapath for each of these instructions. For any control signal that does not matter, write X for “don’t care.” RW is the register file write-enable. BS and V are mux selectors (0 = top input, 1 = bottom input), DW is the data-memory write-enable. Br is the second input of the AND-gate pictured.

<table>
<thead>
<tr>
<th>Insn</th>
<th>RW</th>
<th>BS</th>
<th>DW</th>
<th>V</th>
<th>Br</th>
</tr>
</thead>
<tbody>
<tr>
<td>add</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>sw</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Question 6: Memory Hierarchy [10 pts]

1. SRAM access latency is roughly proportional to \( \text{bits} \times \text{ports}^2 \). Explain why.

2. Explain the motivation for a memory hierarchy (as opposed to just one single memory element).