1. (4 points) In general, what are static hazards in a combinational network.

2. (5 points) Obtain a minimum sum of products expression for the following function:
   \[ f(A, B, C, D) = \Sigma m(0, 1, 2, 5, 8, 9, 10, 13) \]

3. (7 points) Write a short VHDL description of a negative edge-triggered D flip-flop using a VHDL process.

   ```vhdl
   entity DFF is
   port (D, CLK : in bit;
         Q, QBAR : out bit);
   end DFF;

   architecture DFF of DFF is
   begin

   end DFF;
   ```
4. (15 points) In the following VHDL process A, B, C, and D are all integers that have a value of 0 at time – 10 ns. If E changes from ‘0’ to ‘1’ at time 20 ns, specify the time(s) at which each signal will change and the value to which it will change. List these changes in chronological order. List any delta delays as a separate entry.

P1: process
begin
  wait on E;
  A <= 1 after 5 ns;
  B <= A + 1;
  wait for 0 ns;
  C <= B after 10ns;
  D <= A after 3 ns;
  A <= A + 5 after 15 ns;
  B <= B + 7;
end process P1;

<table>
<thead>
<tr>
<th>Time</th>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
<th>E</th>
</tr>
</thead>
<tbody>
<tr>
<td>10 ns</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

5. (6 points) What are the differences between signals and variables in VHDL? Where can signals and variables be used within a typical VHDL model?
6. **(15 points)** A synchronous sequential network has one input and one output. If the input sequence 1001 or 0101 occurs, an output of two successive 1s will occur. The first of these 1s should occur coincident with the last input of the 1001 or 0101 sequence. The network should reset when the second 1 output occurs. For example,

Input sequence: 011011000100101101010
Output sequence: 0000000000110000011

Derive a Mealy state graph and table with a minimum number of states (7 states)

8. **(10 points)** Find a minimum-row PLA table to implement the following sets of functions.

\[
\begin{align*}
    f_1(A, B, C) &= \Sigma m(0, 1, 2, 5) \\
    f_2(A, B, C) &= \Sigma m(0, 2, 4, 6) \\
    f_3(A, B, C) &= \Sigma m(5, 6) \\
    f_4(A, B, C) &= \Sigma m(1, 4, 5, 6, 7)
\end{align*}
\]
7. (10 points) Reduce the following state table to a minimum number of states. Show all your work in determining the state equivalents.

<table>
<thead>
<tr>
<th>Present State</th>
<th>Next State</th>
<th>X = 0</th>
<th>X = 1</th>
<th>Output</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>D</td>
<td>C</td>
<td></td>
<td>0</td>
</tr>
<tr>
<td>B</td>
<td>F</td>
<td>H</td>
<td></td>
<td>0</td>
</tr>
<tr>
<td>C</td>
<td>E</td>
<td>D</td>
<td></td>
<td>1</td>
</tr>
<tr>
<td>D</td>
<td>A</td>
<td>E</td>
<td></td>
<td>0</td>
</tr>
<tr>
<td>E</td>
<td>C</td>
<td>A</td>
<td></td>
<td>1</td>
</tr>
<tr>
<td>F</td>
<td>F</td>
<td>B</td>
<td></td>
<td>1</td>
</tr>
<tr>
<td>G</td>
<td>B</td>
<td>H</td>
<td></td>
<td>0</td>
</tr>
<tr>
<td>H</td>
<td>C</td>
<td>G</td>
<td></td>
<td>1</td>
</tr>
</tbody>
</table>

9. (5 points) Write out the truth table for the following equation.

\[ F = (A + B') \cdot (C + D) \]
10. (5 points) What is the difference between combinational and sequential logic?

11. (5 points) What is the purpose of the architecture and entity sections of a VHDL model? How do they differ?

12. (1 point) An entity X, when used in another entity, becomes a _________________ for the entity Y.
13. (12 points) Draw the state diagram for the following state machine. Is it a Moore machine or a Mealy machine?

```vhdl
ENTITY state_machine IS
    PORT (sig_in : IN BIT; clk : IN BIT;
         sig_out : OUT BIT);
END state_machine;

ARCHITECTURE state_machine OF state_machine IS
    TYPE state_type IS (a, b, c, d, e);
    SIGNAL current_state, next_state : state_type;
BEGIN
    PROCESS (sig_in, current_state)
    BEGIN
        sig_out <= '0';
        next_state <= b;
        CASE current_state
            WHEN a =>
                IF sig_in = '0' THEN
                    next_state <= a;
                ELSE
                    next_state <= d;
                END IF;
                sig_out <= '1';
            WHEN b =>
                IF sig_in = '0' THEN
                    next_state <= b;
                ELSE
                    next_state <= c;
                END IF;
            WHEN c =>
                IF sig_in = '1' THEN
                    next_state <= a;
                ELSE
                    next_state <= d;
                END IF;
                sig_out <= '1';
            WHEN d =>
                IF sig_in = '0' THEN
                    next_state <= e;
                END IF;
            WHEN e =>
                IF sig_in = '1' THEN
                    next_state <= c;
                END IF;
        END CASE;
    END PROCESS;
    PROCESS (clk)
    BEGIN
        IF (clk'EVENT AND clk = '1') THEN
            current_state <= next_state;
        END IF;
    END PROCESS;
END state_machine;
```