Solutions - Homework 4

(Due date: March 31st @ 5:30 pm)
Presentation and clarity are very important! Show your procedure!

PROBLEM 1 (20 pts)

- Design a counter using a Finite State Machine (FSM):
  Counter features:
  ✓ Count: 000, 010, 111, 011, 110, 100, 001, 101, 000, 010, 111, ...
  ✓ resetn: Asynchronous active-low input signal. It initializes the count to “000”
  ✓ Input E: Synchronous input that increases the count when it is set to ‘1’.
  ✓ output z: It becomes ‘1’ when the count is 101.

  Provide the State Diagram (any representation), State Table, and the Excitation Table. Is this a Mealy or a Moore machine? Why? (10 pts)
  Provide the excitation equations (simplify your circuit using K-maps or the Quine-McCluskey algorithm) (5 pts)
  Sketch the circuit. (5 pts)

- State Diagram and State Table:

  ![State Diagram](image)

  The output ‘z’ only depends on the present state ⇒ Moore FSM.

  State Assignment and Excitation Table:

<table>
<thead>
<tr>
<th>PRESENT STATE</th>
<th>NEXT STATE</th>
<th>E</th>
<th>Q0Q1Q2(t)</th>
<th>Q0Q1Q2(t+1)</th>
<th>z</th>
</tr>
</thead>
<tbody>
<tr>
<td>0 0 0</td>
<td>0 0 0</td>
<td>0</td>
<td>0 0 0</td>
<td>0 0 0</td>
<td>0</td>
</tr>
<tr>
<td>0 0 1</td>
<td>0 0 1</td>
<td>0</td>
<td>0 1 0</td>
<td>0 1 0</td>
<td>0</td>
</tr>
<tr>
<td>0 1 1</td>
<td>0 1 1</td>
<td>0</td>
<td>1 1 1</td>
<td>1 1 1</td>
<td>0</td>
</tr>
<tr>
<td>0 0 1</td>
<td>0 1 1</td>
<td>0</td>
<td>0 1 1</td>
<td>0 1 1</td>
<td>0</td>
</tr>
<tr>
<td>0 1 0</td>
<td>1 0 0</td>
<td>0</td>
<td>1 1 0</td>
<td>1 1 0</td>
<td>0</td>
</tr>
<tr>
<td>0 1 0</td>
<td>1 0 0</td>
<td>0</td>
<td>1 0 0</td>
<td>1 0 0</td>
<td>0</td>
</tr>
<tr>
<td>0 0 1</td>
<td>0 0 1</td>
<td>0</td>
<td>0 0 1</td>
<td>0 0 1</td>
<td>0</td>
</tr>
<tr>
<td>0 1 1</td>
<td>1 0 1</td>
<td>1</td>
<td>1 0 1</td>
<td>1 0 1</td>
<td>1</td>
</tr>
<tr>
<td>1 0 0</td>
<td>0 1 0</td>
<td>1</td>
<td>0 1 0</td>
<td>0 1 0</td>
<td>0</td>
</tr>
<tr>
<td>1 0 1</td>
<td>1 1 0</td>
<td>1</td>
<td>1 0 1</td>
<td>1 1 0</td>
<td>0</td>
</tr>
<tr>
<td>1 1 0</td>
<td>1 1 0</td>
<td>1</td>
<td>1 1 0</td>
<td>1 1 0</td>
<td>0</td>
</tr>
<tr>
<td>1 1 0</td>
<td>1 1 0</td>
<td>1</td>
<td>1 0 0</td>
<td>1 0 0</td>
<td>0</td>
</tr>
<tr>
<td>1 0 1</td>
<td>1 0 1</td>
<td>1</td>
<td>0 1 0</td>
<td>0 1 0</td>
<td>0</td>
</tr>
<tr>
<td>1 1 0</td>
<td>0 0 1</td>
<td>1</td>
<td>1 0 0</td>
<td>0 0 1</td>
<td>0</td>
</tr>
<tr>
<td>1 0 1</td>
<td>1 0 1</td>
<td>1</td>
<td>0 1 0</td>
<td>0 1 0</td>
<td>0</td>
</tr>
<tr>
<td>1 1 0</td>
<td>0 0 0</td>
<td>1</td>
<td>1 0 0</td>
<td>0 0 0</td>
<td>1</td>
</tr>
</tbody>
</table>
- **Excitation equations and minimization:**

  \[
  Q_2(t+1) = E\overline{Q}_2Q_0 + EQ_1\overline{Q}_0
  \]

  \[
  Q_1(t+1) = Q_1Q_0 + E\overline{Q}_1 + EQ_2\overline{Q}_0
  \]

  \[
  Q_0(t+1) = EQ_2 + Q_0(Q_1\oplus Q_2) + EQ_2(Q_1\oplus Q_2)
  \]

  \[z = Q_2\overline{Q}_1Q_0\]

- **Circuit Implementation:**

![Circuit Diagram](image-url)
**PROBLEM 2 (15 pts)**

- Complete the timing diagram of the following circuit. \( G = G_3G_1G_2G_0 = 1001 \), \( Q = Q_3Q_2Q_1Q_0 \)

[Diagram of the circuit is shown here with timing signals and state transitions labeled.]

**PROBLEM 3 (30 pts)**

- Sequence detector (with overlap): (10 pts)
  Draw the state diagram (any representation) of a circuit (with an input \( x \)) that detects the following sequence: 101001. The detector must assert an output \( z \) when the sequence is detected.

[State diagram is shown here with states labeled and transitions indicated.]
• Complete the timing diagram of the following FSM. Is this a Mealy or a Moore machine? Why? (5 pts)

On S4 and S3, the output \( z \) depends on the value of \( x \). Thus, it is a Mealy machine.

• Provide the state diagram (in ASM form) and complete the timing diagram of the FSM whose VHDL description is listed below. (15 pts)

```
library ieee;
use ieee.std_logic_1164.all;

entity circ is
  port ( clk, resetn: in std_logic;
         a, b: in std_logic;
         x, w, z: out std_logic);
end circ;

architecture behavioral of circ is
  type state is (S1, S2, S3);
  signal y: state;
begin
  Transitions: process (resetn, clk, a, b)
  begin
    if resetn = '0' then
      y <= S1;
    elsif (clk'event and clk = '1') then
      case y is
        when S1 =>
          if a = '1' then
            if b = '0' then y <= S3; else y <= S1; end if;
            else y <= S2;
            end if;
        when S2 =>
          if b = '1' then y <= S3; else y <= S2; end if;
        when S3 =>
          if a = '1' then y <= S2; else y <= S1; end if;
          end case;
      end if;
    end if;
  end process;

  Outputs: process (y, a, b)
  begin
    x <= '0'; w <= '0'; z <= '0';
    case y is
      when S1 =>
        if a = '1' then z <= '1'; end if;
        when S2 => w <= '1';
        when S3 => if a = '0' then x <= '1'; end if;
      end case;
    end process;
  end behavioral;
```

```
resetn = 0
1/0
S1
0/1
S2
1/0
0/0
0/0
0/1
S4
1/1
S3

x/z
1/0
S1
1/0
S2
S4
S3

clock
resetn
x
state

S1
S1
S1
S1
S2
S2
S2
S2
S3
S1
S1
S2
S3
S1
S1
S2
S2
S3
S1
S1
```
**Problem 4 (20 pts)**
- Complete the timing diagram of the following digital circuit that includes an FSM (in ASM form) and a datapath circuit.

![Timing Diagram](image)

**Problem 5 (15 pts)**
- Attach a printout of your Project Status Report (no more than three pages, single-spaced, 2 columns). This report should contain the current status of the project. You **MUST** use the provided template (Final Project – Report Template.docx).