Solutions - Quiz 4 (Online Section)
(Due Date: December 6th by 11:00 am)

Problem 1 (30 pts)
- Complete the timing diagram of the following FSM (represented in ASM form):

Problem 2 (20 pts)
- Provide the state diagram (in ASM form) of the FSM whose VHDL description is listed below:

library ieee;
use ieee.std_logic_1164.all;

entity circ is
  port ( clk, rstn: 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 (rstn, clk, a, b)
    begin
      if rstn = '0' then
        y <= S1;
      elsif (clk'event and clk = '1') then
        case y is
          when S1 =>
            if a = '1' then y <= S2;
            else if b = '1' then y <= S3;
            else y <= S1; end if;
          end if;
        when S2 =>
          if a = '1' then y <= S2;
          else y <= S1; end if;
        when S3 =>
          if b = '1' then y <= S3;
          else y <= S1; end if;
        end case;
      end if;
    end process;

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

PROBLEM 3 (30 pts)

- **Basic Processor:**
  
  *Available Registers:* R0 (register 0, 4 bits), R1 (register 1, 4 bits), PC (program counter, 4 bits), OUT (output register, 4 bits)
  
  *Instruction Memory:* Stores up to 16 8-bit instructions.
  
  *Instruction Set:* Instructions are specified on the Instruction Register (IR):

  ![IR Diagram]

  
  \[ \text{IR:} \]

  
  \[ \text{OPCODE} \quad \text{DR} \quad \text{SR} \]

  
  \[ \text{IMMEDIATE DATA} \]

  
  DR=0 \( \Rightarrow \) R0 is the destination register, DR=1 \( \Rightarrow \) R1 is the destination register.
  
  SR=0 \( \Rightarrow \) R0 is the source register, SR=1 \( \Rightarrow \) R1 is the source register.

  
<table>
<thead>
<tr>
<th>OPCODE [IR[7..5]]</th>
<th>Instruction</th>
<th>Operation</th>
</tr>
</thead>
<tbody>
<tr>
<td>000</td>
<td>MOV DR, SR</td>
<td>DR ( \leftarrow ) SR</td>
</tr>
<tr>
<td>001</td>
<td>LOADI DR, DATA</td>
<td>DR ( \leftarrow ) DATA, DATA = IR[3..0]</td>
</tr>
<tr>
<td>010</td>
<td>ADD DR, SR</td>
<td>DR ( \leftarrow ) DR + SR</td>
</tr>
<tr>
<td>011</td>
<td>ADDI DR, DATA</td>
<td>DR ( \leftarrow ) DR + DATA, DATA = IR[3..0]</td>
</tr>
<tr>
<td>100</td>
<td>SR0 DR, SR</td>
<td>DR ( \leftarrow ) 0&amp;SR[3..1]</td>
</tr>
<tr>
<td>101</td>
<td>IN DR</td>
<td>DR ( \leftarrow ) IN</td>
</tr>
<tr>
<td>110</td>
<td>OUT DR</td>
<td>OUT ( \leftarrow ) DR</td>
</tr>
<tr>
<td>111</td>
<td>JNZ DR, ADDRESS</td>
<td>PC ( \leftarrow ) PC + 1 if DR=0</td>
</tr>
</tbody>
</table>

  \* ADDRESS = IR[3..0]

  
  Write an assembly program for a counter from 1 to 12: 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 1, 2, 3, .... The count must be shown on the output register (OUT). Note that you can only have up to 16 instructions.

  
  start: loadi R0,1
  
  loadi R1,4
  
  loop: out R0 \( \rightarrow \) OUT: shows the count
  
  addi R0,1
  
  addi R1,1
  
  jnz R1, loop
  
  loadi R0,1
  
  jnz R0, start

PROBLEM 4 (20 pts)

- **Sequence detector (with overlap):** Draw the state diagram (in ASM form) of a circuit that detects the following sequence: 0110. The detector must assert an output z = ‘1’ when the sequence is detected.