Final Exam (Online Section)
(Due Date: December 11th by 10:00 am)
Clarity is very important! Show your procedure!

**Problem 1 (12 pts)**
- Complete the timing diagram of the following circuit. $Q = Q_4Q_3Q_2Q_1Q_0$

**Problem 2 (10 pts)**
- We want to design a counter modulo-9 with enable using a State Machine. The counter must assert an output $z='1'$ when the maximum count is reached.
- Provide the State Diagram (any representation) and the Excitation table $[\text{Inputs}|\text{Present State}|\text{Next State}|\text{Outputs}]$. Is this a Moore or a Mealy machine?

**Problem 3 (15 pts)**
- Given the following system, complete the Timing Diagram.
- The LUT 8-to-8 implements the following function: $OLUT = [sqrt(ILUT)]$
**Problem 4 (18 pts)**
- Provide the State Diagram (any representation) of the FSM whose VHDL description is shown below.
- Complete the Timing Diagram.

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

**Problem 5 (10 pts)**
- Sequence detector (with overlap): Draw the state diagram (in ASM form) of a circuit (with an input 'x') that detects the following sequence: 1011011. The detector must assert and output z='1' when the sequence is detected.
**Problem 6 (15 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)

  IR (instruction register, 8 bits)

  Instruction Memory: Stores up to 16 8-bit instructions.

  **Instruction Set:** Instructions are specified on the Instruction Register (IR):

  - DR = 0 ⇒ R0 is the destination register, DR = 1 ⇒ R1 is the destination register.
  - SR = 0 ⇒ R0 is the source register, SR = 1 ⇒ R1 is the source register.

  **OPCODE (IR[7..5])**

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

- Write an assembly program for a counter from 2 to 13: 2, 3, ..., 13, 2, 3, ... The count must be shown on the output register (OUT). Use labels to specify any address where your program jumps.

- Provide the contents of the Instruction Memory.
PROBLEM 7 (20 PTS)

- Complete the timing diagram of the following digital circuit that includes an FSM (in ASM form) and a datapath circuit.