Problem 5, 3 Solutions

Problem 1

(a)

<table>
<thead>
<tr>
<th>Qn</th>
<th>Qn+1</th>
<th>X</th>
<th>Dn</th>
<th>Dn+1</th>
<th>Z</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<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>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

(b)

```
-- This is a sample solution to Problem 5

library ieee;
use ieee.std_logic_1164.all;
use work.std_arith.all;
entity cd_fsm is port(
  clk : in std_logic;
  reset : in std_logic;
  X : in std_logic;
  Z : out std_logic;
  D : out std_logic_vector(1 downto 0));
end cd_fsm;

architecture state_machine of cd_fsm is
  type StateType is (ZeroZero, ZeroOne, OneZero, OneOne);
  signal present_state, next_state : StateType;
begin
  state_machine :
    process(present_state, X)
    begin
      case present_state is
        when ZeroZero =>
          next_state <= ZeroZero;
          Z <= X;
        when ZeroOne =>
          if X = '1' then
            next_state <= OneZero;
          else
            next_state <= ZeroZero;
            end if;
            Z <= '1';
        when OneZero =>
          if X = '1' then
            next_state <= OneOne;
          else
            next_state <= ZeroZero;
            end if;
            Z <= X;
        when OneOne =>
          next_state <= OneOne;
          Z <= '1';
      end case;
    end process;
end state_machine;
```
when others =>
  next_state <= OneZero;
end case;
Z <= '0';
end process;

--state_comb;

state_clocked: process(clk, reset) begin -- state transitions
  if (reset = '1') then
    present_state <= ZeroOne;
  else if rising_edge(clk) then
    present_state <= next_state;
  end if;
end process state_clocked;
end state_machine;

(d)

<table>
<thead>
<tr>
<th>Qc</th>
<th>Qb</th>
<th>Qa</th>
<th>V/B</th>
<th>Dc</th>
<th>Dn</th>
<th>Da</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

On reset, Dc, Dn, Da should be set to 0.
(b) 

<table>
<thead>
<tr>
<th>Qa</th>
<th>Qb</th>
<th>Qc</th>
<th>Qd</th>
</tr>
</thead>
<tbody>
<tr>
<td>00</td>
<td>01</td>
<td>10</td>
<td>11</td>
</tr>
<tr>
<td>01</td>
<td>01</td>
<td>10</td>
<td>11</td>
</tr>
<tr>
<td>10</td>
<td>01</td>
<td>10</td>
<td>11</td>
</tr>
<tr>
<td>11</td>
<td>01</td>
<td>10</td>
<td>11</td>
</tr>
</tbody>
</table>

(Dc) 

(Db) 

(Da) 

(c) 

Problem 3

(a) 
Inputs should be connected as follows:
A = 1
B = C = D = 0
/LD = Qc = as and QD
/CLR = 1
(b) Inputs should be connected as follows:
\[
A = 0
\]
\[
B = \text{not } Q_B
\]
\[
C = Q_C \text{ or } (Q_A \text{ and not } Q_B)
\]
\[
D = (\text{not } Q_B \text{ and } Q_C \text{ and } Q_A) \text{ or }
(Q_B \text{ and not } Q_C) \text{ or }
(Q_C \text{ and not } Q_B) \text{ or }
(Q_B \text{ and not } Q_A)
\]
\[
/LD = 0 \quad /CLR = 1
\]

(c) Inputs should be connected as follows:
\[
A = Q_A \text{ and not } Q_b
\]
\[
B = (\text{not } Q_B \text{ and not } Q_C) \text{ or } (Q_B \text{ and } Q_C)
\]
\[
C = (\text{not } Q_B \text{ and } Q_B \text{ and } Q_A) \text{ or } (Q_B \text{ and } Q_B \text{ and } \text{not } Q_A)
\]
\[
D = Q_C
\]
\[
/LD = 0 \quad /CLR = 1
\]