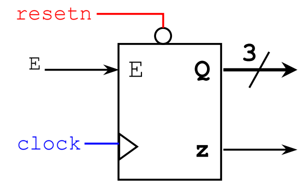
Solutions - Homework 4

(Due date: March 31st @ 5:30 pm)

Presentation and clarity are very important! Show your procedure!

PROBLEM 1 (20 PTS)

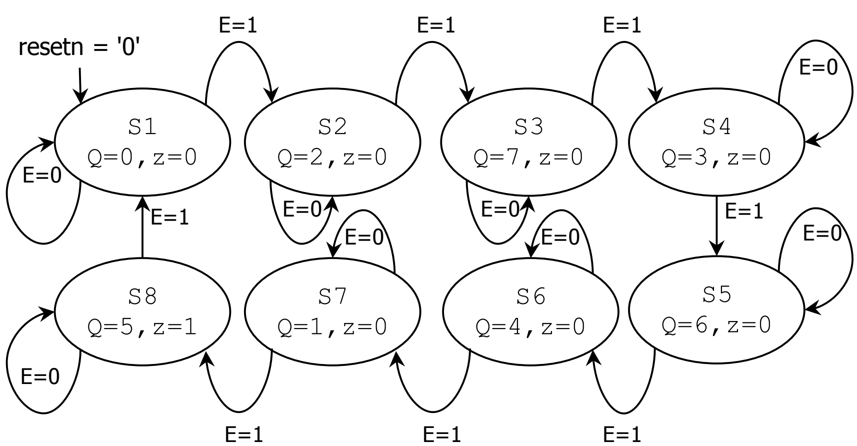
- Design a counter using a Finite State Machine (FSM):
  Counter features:
  - ✓ Count: **000**, 010, 111, 011, 110, 100, 001, 101, **000**, 010, 111, ...
  - ✓ $resetn$: Asynchronous active-low input signal. It initializes the count to "000"
  - ✓ Input E: Synchronous input that increases the count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



Counter

- 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*:



E	PRESENT STATE	NEXT STATE	z
0	S1	S1	0
0	S2	S2	0
0	S3	S3	0
0	S4	S4	0
0	S5	S5	0
0	S6	S6	0
0	S7	S7	0
0	S8	S8	1
1	S1	S2	0
1	S2	S3	0
1	S3	S4	0
1	S4	S5	0
1	S5	S6	0
1	S6	S7	0
1	S7	S8	0
1	S8	S1	1

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

- *State Assignment and Excitation Table*:
  - ✓ S1: Q = 000
  - ✓ S2: Q = 010
  - ✓ S3: Q = 111
  - ✓ S4: Q = 011
  - ✓ S5: Q = 110
  - ✓ S6: Q = 100
  - ✓ S7: Q = 001
  - ✓ S8: Q = 101

	PRESENT STATE	NEXTSTATE	
E	$Q_2Q_1Q_0$(t)	$Q_2Q_1Q_0$(t+1)	z
0	0 0 0	0 0 0	0
0	0 1 0	0 1 0	0
0	1 1 1	1 1 1	0
0	0 1 1	0 1 1	0
0	1 1 0	1 1 0	0
0	1 0 0	1 0 0	0
0	0 0 1	0 0 1	0
0	1 0 1	1 0 1	1
1	0 0 0	0 1 0	0
1	0 1 0	1 1 1	0
1	1 1 1	0 1 1	0
1	0 1 1	1 1 0	0
1	1 1 0	1 0 0	0
1	1 0 0	0 0 1	0
1	0 0 1	1 0 1	0
1	1 0 1	0 0 0	1

- *Excitation equations and minimization:*

$$Q_2(t+1) = \bar{E}Q_2 + E\overline{Q_2}Q_0 + EQ_1\overline{Q_0}$$
$$Q_1(t+1) = Q_1Q_0 + \bar{E}Q_1 + E\overline{Q_2}\,\overline{Q_0}$$
$$Q_0(t+1) = \bar{E}Q_0 + Q_0(\overline{Q_1 \oplus Q_2}) + E\overline{Q_0}(Q_1 \oplus Q_2)$$
$$z = Q_2\overline{Q_1}Q_0$$

Q_2(t+1)

Q_1Q_0 \ EQ_2	00	01	11	10
00	0	1	0	0
01	0	1	0	1
11	0	1	0	1
10	0	1	1	1

Q_1(t+1)

Q_1Q_0 \ EQ_2	00	01	11	10
00	0	0	0	1
01	0	0	0	0
11	1	1	1	1
10	1	1	0	1

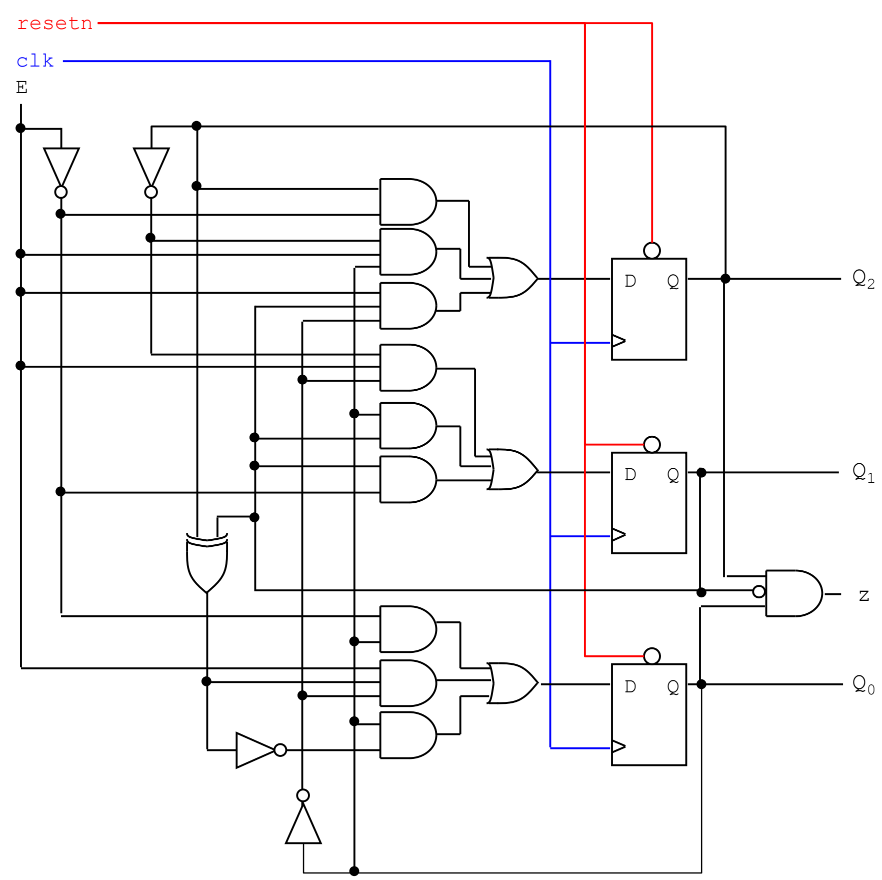
Q_0(t+1)

Q_1Q_0 \ EQ_2	00	01	11	10
00	0	0	1	0
01	1	1	0	1
11	1	1	1	0
10	0	0	0	1

z

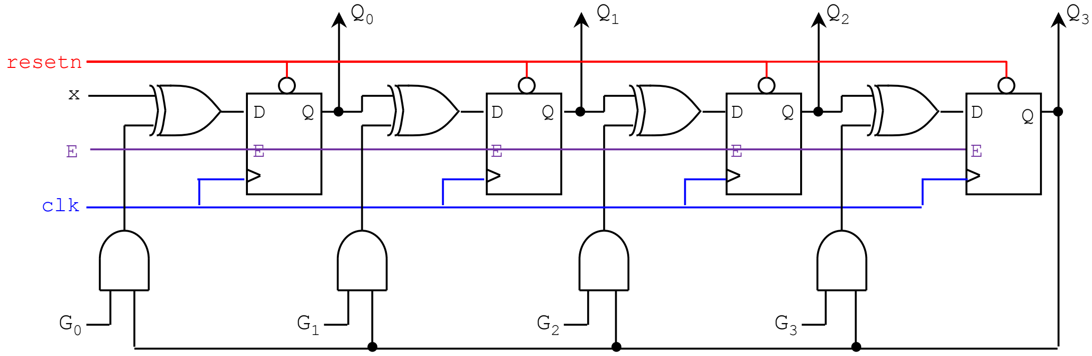
Q_1Q_0 \ EQ_2	00	01	11	10
00	0	0	0	0
01	0	1	1	0
11	0	0	0	0
10	0	0	0	0

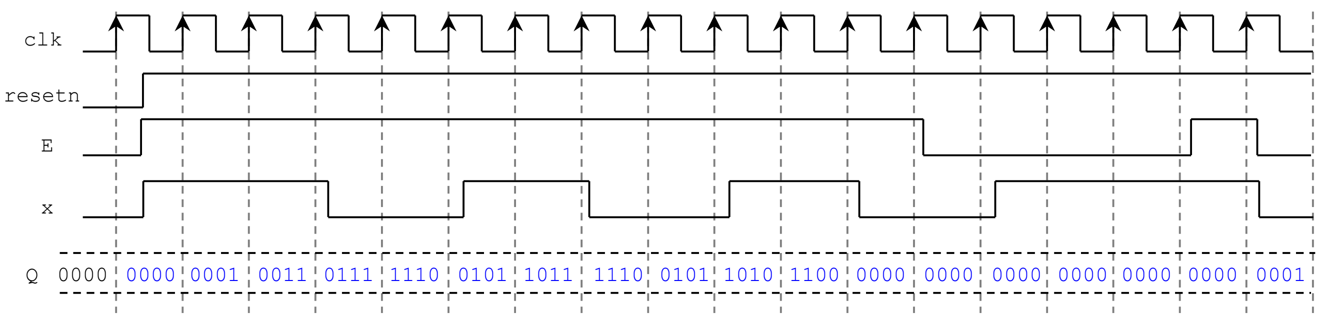
- *Circuit Implementation:*



PROBLEM 2 (15 PTS)

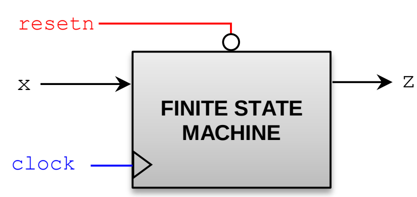
- Complete the timing diagram of the following circuit. $G = G_3G_1G_2G_0 = 1001$, $Q = Q_3Q_2Q_1Q_0$

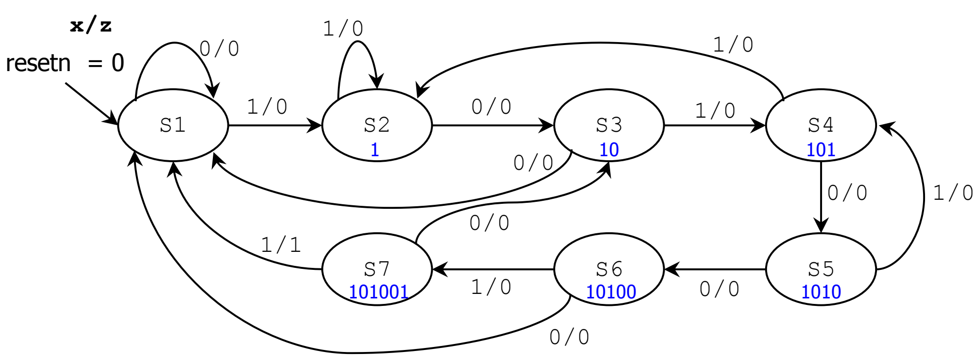




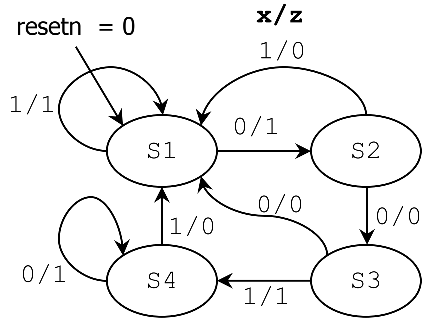
PROBLEM 3 (30 PTS)

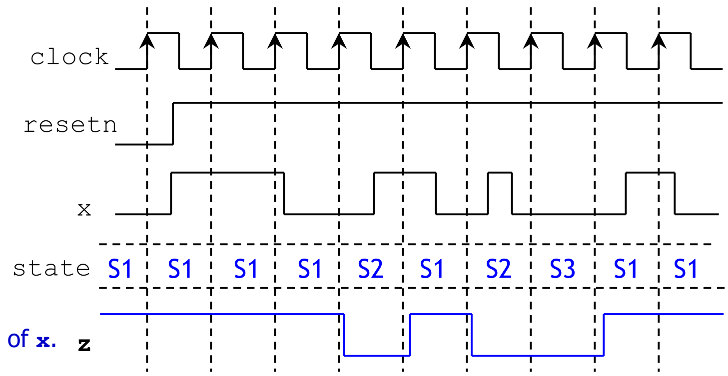
- Sequence detector (with overlap): (10 pts)
  Draw the state diagram (<u>any representation</u>) of a circuit (with an input x) that detects the following sequence: 1010011. The detector must assert an output z when the sequence is detected.





- Complete the timing diagram of the following FSM. Is this a Mealy or a Moore machine? Why? (5 pts)

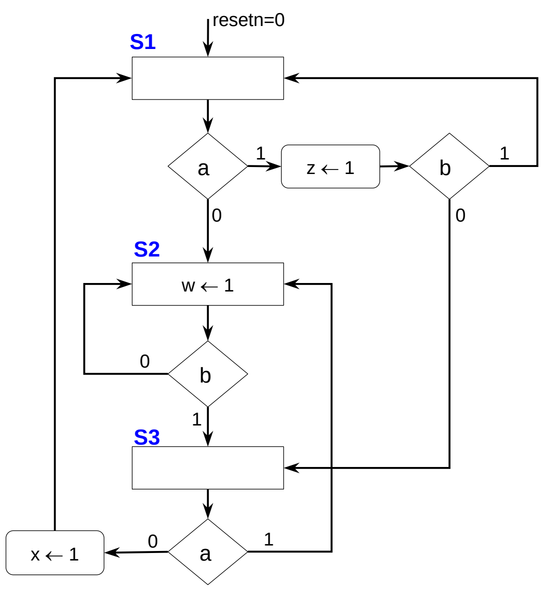




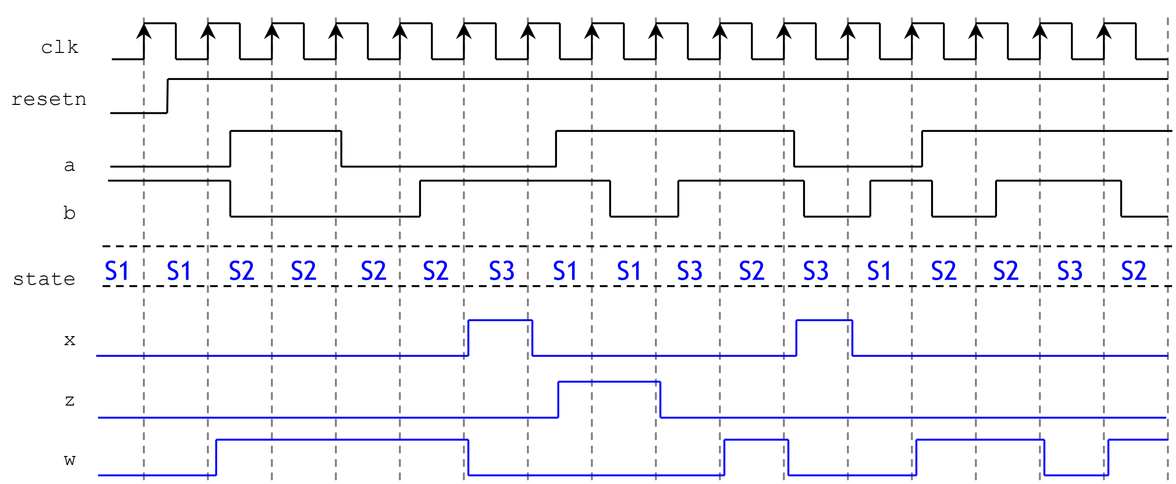
On S4 and S3, the output **z** depends on the value of **x**. Thus, it is a Mealy machine.

- 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;
```

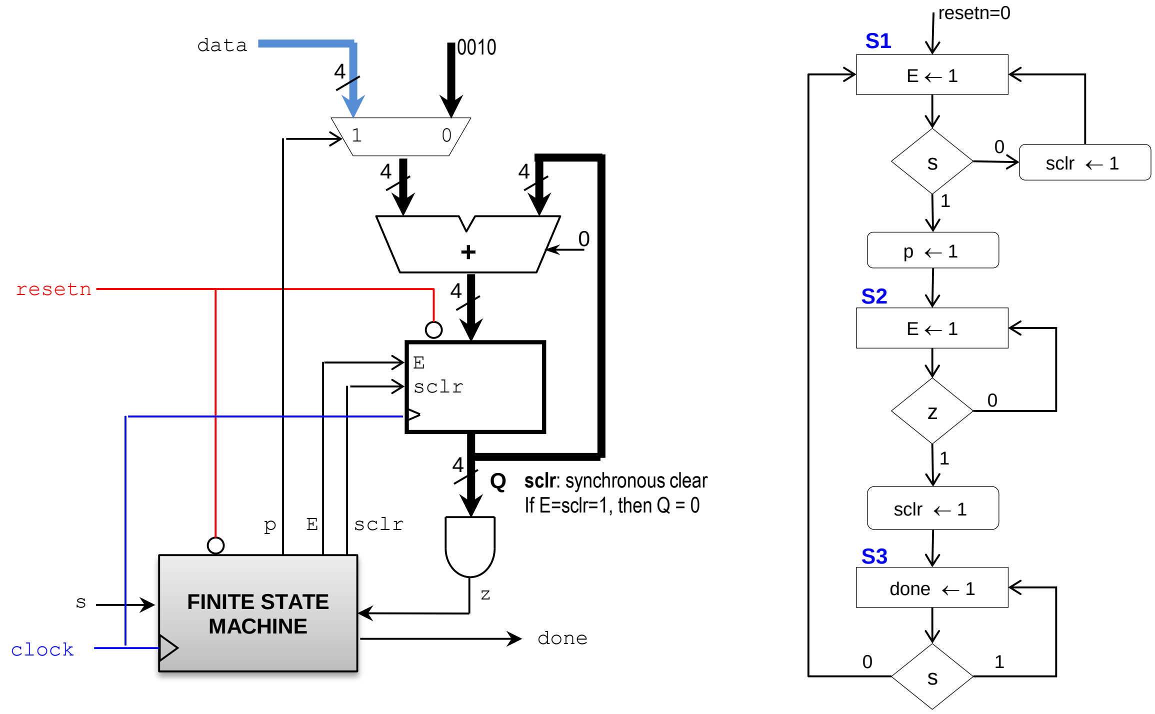


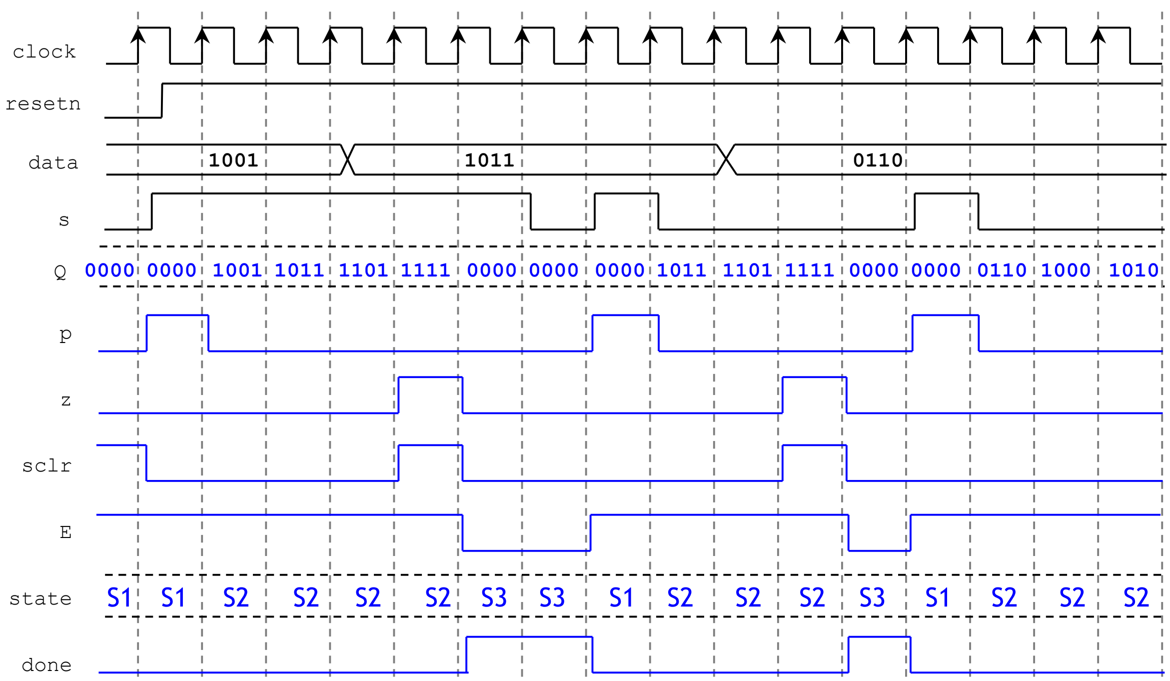

```
architecture behavioral of circ is
   type state is (S1, S2, S3);
   signal y: state;
begin
  Transitions: process (resetn, clk, a, b)
  begin
     if resetn = '0' then y <= S1;
     elsif (clk'event and clk = '1') then
        case y is
          when S1 =>
            if a = '1' then
         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



PROBLEM 4 (20 PTS)

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





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`).