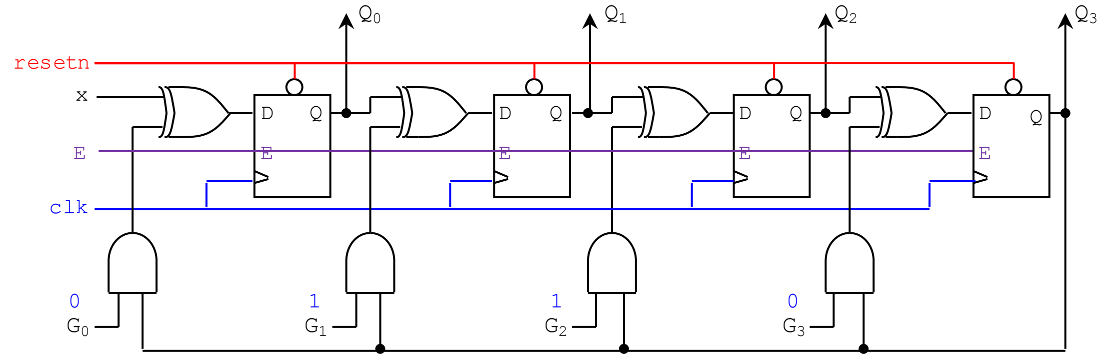
Solutions - Final Exam

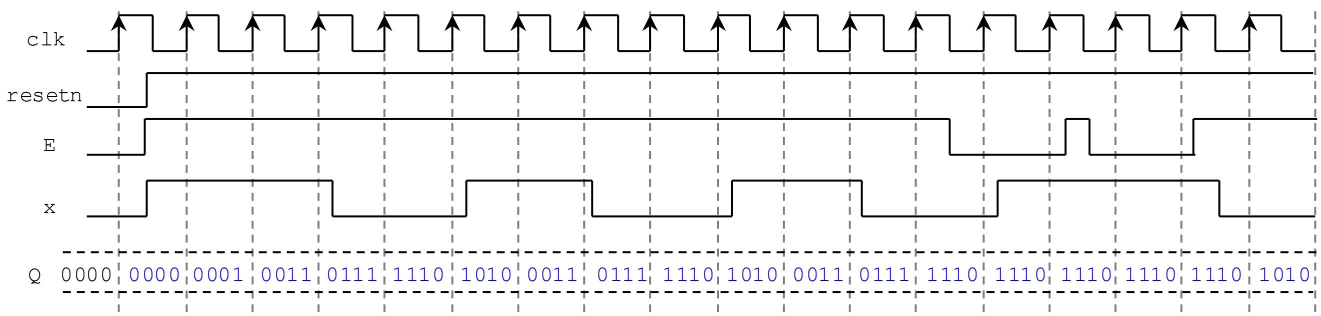
(December 13th @ 7:00 pm)

Presentation and clarity are very important! Show your procedure!

PROBLEM 1 (12 PTS)

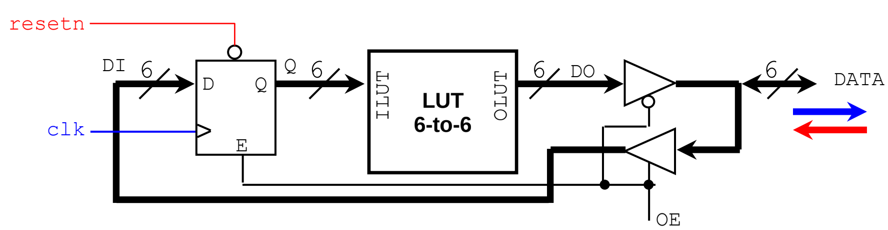
- Complete the timing diagram of the following circuit. $G = G_3G_2G_1G_0 = 0110$, $Q = Q_3Q_2Q_1Q_0$





PROBLEM 2 (12 PTS)

- Given the following circuit, complete the timing diagram.
  The LUT 6-to-6 implements the following function: $OLUT = |ILUT|$, where $ILUT$ is a 6-bit signed (2C) number, and $OLUT$ is a 6-bit unsigned number.
  For example: $ILUT = -27 = 100101_2 = \rightarrow\ OLUT = |-27| = 27 = 011011_2$

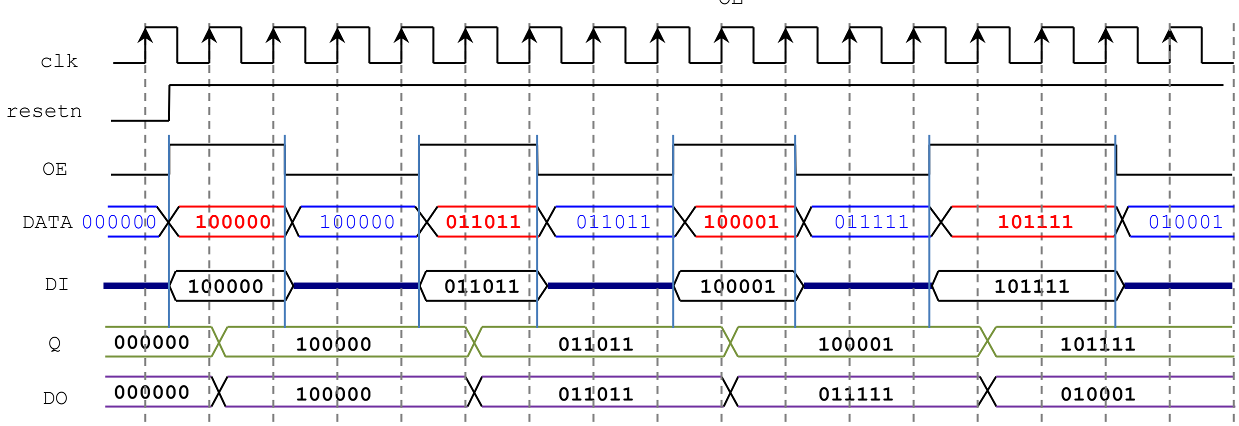


$|-32| = 32$

$|27| = 27$

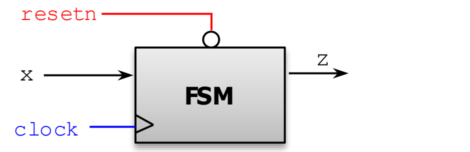
$|-31| = 31$

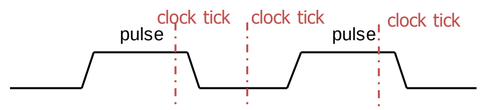
$|-17| = 17$

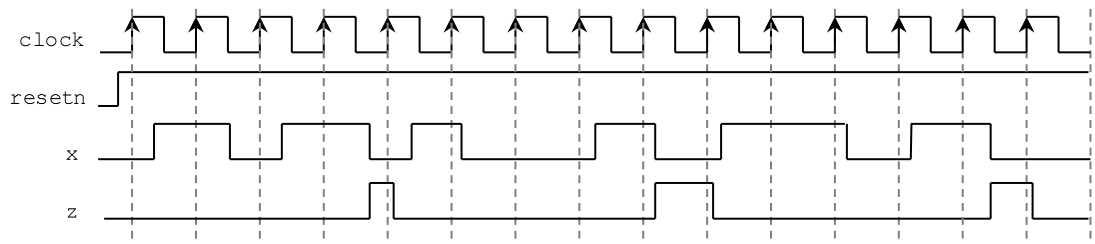


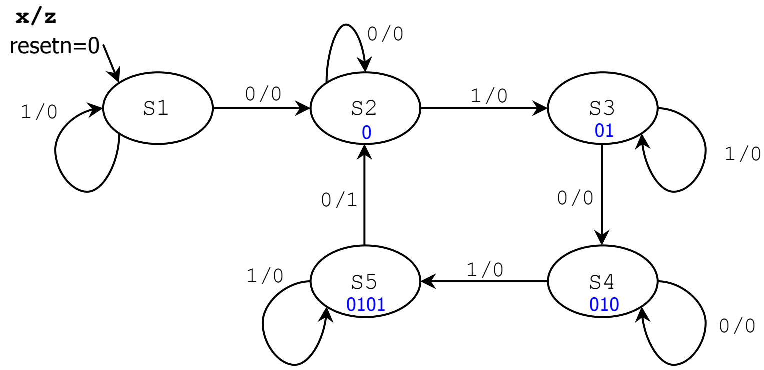
PROBLEM 3 (20 PTS)

- Two-pulse Detector: The timing diagram shows the behavior of the circuit. The FSM generates $z = 1$ when it detects two pulses. Note how in this design, the output z is 1 as soon as the second $1 \rightarrow 0$ transition is detected. Once the two pulses are detected, the FSM looks for a new pair of pulses. Assumption: For the circuit to detect a '1' or a '0' on x, this value needs to happen when a rising edge occurs.
  Draw the State Diagram (any representation) of the given FSM (9 pts).

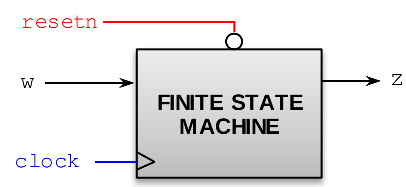






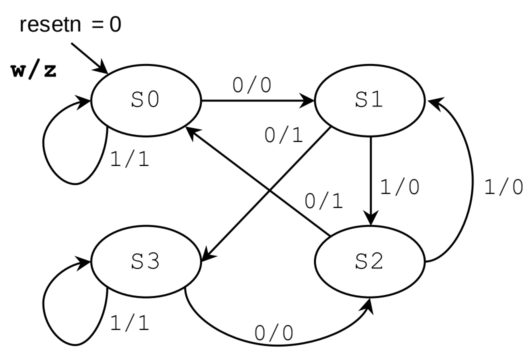


- The following FSM has 4 states, one input w and one output z. (11 pts)
  - ✓ The excitation equations are given by:
    - $Q_1(t+1) \leftarrow Q_0(t)$
    - $Q_0(t+1) \leftarrow Q_1(t) \oplus \overline{w}$
  - ✓ The output equation is given by: $z = Q_1(t) \oplus Q_0(t) \oplus w$
  - ✓ Provide the State Diagram (any representation) and the Excitation Table.
  - ✓ Sketch the Finite State Machine circuit.

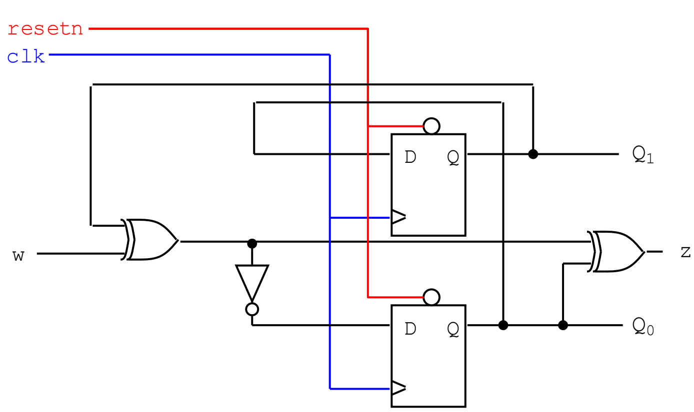


PRESENT STATE w	$Q_1Q_0(t)$	NEXTSTATE $Q_1Q_0(t+1)$	z
0	0 0	0 1	0
0	0 1	1 1	1
0	1 0	0 0	1
0	1 1	1 0	0
1	0 0	0 0	1
1	0 1	1 0	0
1	1 0	0 1	0
1	1 1	1 1	1

w	PRESENT STATE	NEXT STATE	z
0	S0	S1	0
0	S1	S3	1
0	S2	S0	1
0	S3	S2	0
1	S0	S0	1
1	S1	S2	0
1	S2	S1	0
1	S3	S3	1

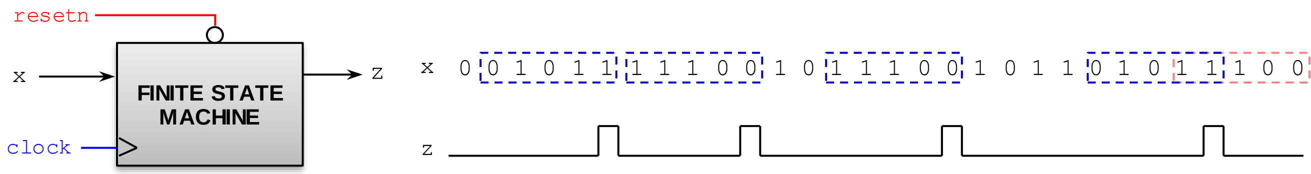


State Assignment:
S0: Q=00 S1: Q=01
S2: Q=10 S3: Q=11

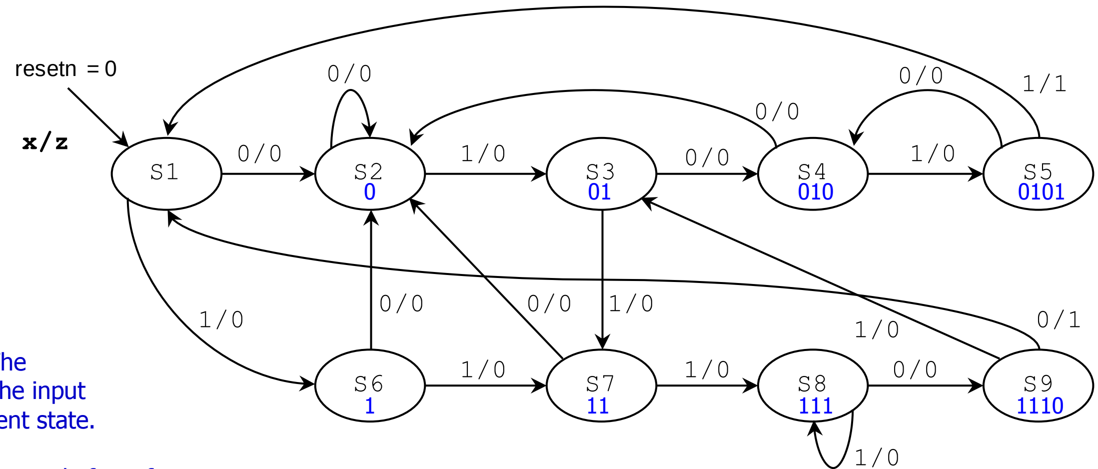


PROBLEM 4 (11 PTS)

- Sequence detector: This FSM generates $z = 1$ when it detects the sequence 11100 or 01011. Once the sequence is detected, the circuit looks for a new sequence. Note that once we start detecting a sequence, we prioritize the sequence that we have over the other (e.g.: last sequence inside a dotted red rectangle is not considered).



- ✓ Draw the State Diagram (any representation) and provide the State Table of this circuit with input x and output z.
- ✓ Which type is this FSM? ~~(Mealy)~~ (Moore) Why? ______________________



This is a Mealy FSM. The output z depends on the input as well as on the present state.

* Excitation Table shown only for reference.

State Assignment:

S1: Q=0000 S2: Q=0001
S3: Q=0010 S4: Q=0011
S5: Q=0100 S6: Q=0101
S7: Q=0110 S8: Q=0111
S9: Q=1000

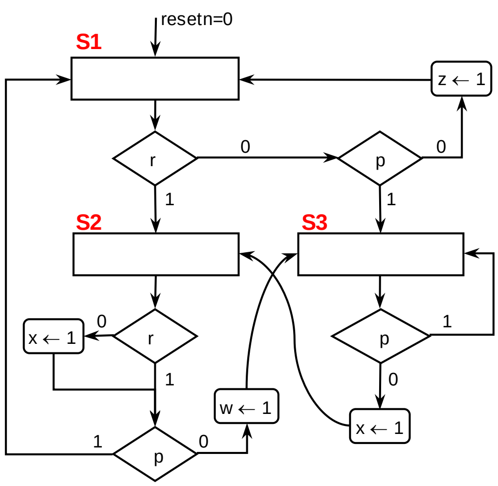
x	PRESENT STATE	NEXT STATE	z
0	S1	S2	0
0	S2	S2	0
0	S3	S4	0
0	S4	S2	0
0	S5	S4	0
0	S6	S2	0
0	S7	S2	0
0	S8	S9	0
0	S9	S1	1
1	S1	S6	0
1	S2	S3	0
1	S3	S7	0
1	S4	S5	0
1	S5	S1	1
1	S6	S7	0
1	S7	S8	0
1	S8	S8	0
1	S9	S3	0

	PRESENT STATE	NEXTSTATE	
x	$Q_3Q_2Q_1Q_0$ (t)	$Q_3Q_2Q_1Q_0$ (t+1)	z
0	0 0 0 0	0 0 0 1	0
0	0 0 0 1	0 0 0 1	0
0	0 0 1 0	0 0 1 1	0
0	0 0 1 1	0 0 0 1	0
0	0 1 0 0	0 0 1 1	0
0	0 1 0 1	0 0 0 1	0
0	0 1 1 0	0 0 0 1	0
0	0 1 1 1	1 0 0 0	0
0	1 0 0 0	0 0 0 0	1
0	1 0 0 1	X X X X	X
0	1 0 1 0	X X X X	X
0	1 0 1 1	X X X X	X
0	1 1 0 0	X X X X	X
0	1 1 0 1	X X X X	X
0	1 1 1 0	X X X X	X
0	1 1 1 1	X X X X	X
1	0 0 0 0	0 1 0 1	0
1	0 0 0 1	0 0 1 0	0
1	0 0 1 0	0 1 1 0	0
1	0 0 1 1	0 1 0 0	0
1	0 1 0 0	0 0 0 0	1
1	0 1 0 1	0 1 1 0	0
1	0 1 1 0	0 1 1 1	0
1	0 1 1 1	0 1 1 1	0
1	1 0 0 0	0 0 1 0	0
1	1 0 0 1	X X X X	X
1	1 0 1 0	X X X X	X
1	1 0 1 1	X X X X	X
1	1 1 0 0	X X X X	X
1	1 1 0 1	X X X X	X
1	1 1 1 0	X X X X	X
1	1 1 1 1	X X X X	X

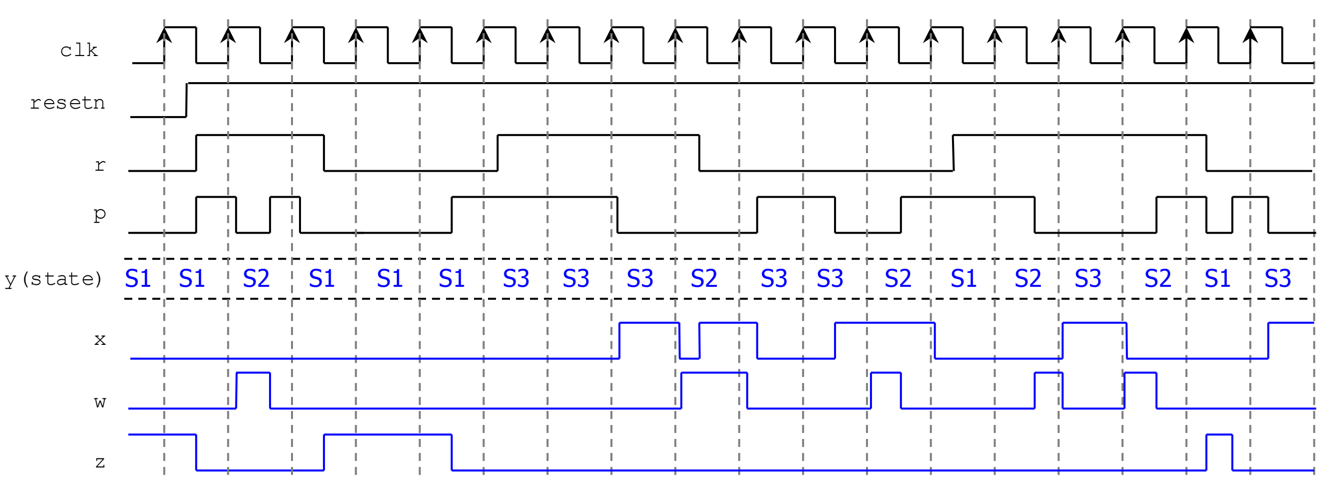
PROBLEM 5 (27 PTS)

- Draw the State Diagram (in ASM form) of the FSM whose VHDL description in shown below. (5 pts)
- Complete the Timing Diagram. (7 pts)
- Provide the State Table and the Excitation Table. Is it a Mealy or a Moore FSM? (6 pts).
- Provide the excitation equations and the Boolean output equations (simplify your circuit: K-maps or Quine-McCluskey).
- Sketch the circuit. (3 pts)

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

entity myfsm is
   port ( clk, resetn: in std_logic;
          r, p: in std_logic;
          x, w, z: out std_logic);
end myfsm;
```




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

          when S2 =>
            if p = '1' then y <= S1; else y <= S3; end if;

          when S3 =>
            if p = '1' then y <= S3; else y <= S2; end if;

        end case;
     end if;
  end process;

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

         when S2 => if r = '0' then x <= '1'; end if;
                    if p = '0' then w <= '1'; end if;

         when S3 => if p = '0' then x <= '1'; end if;
      end case;
  end process;
end behavioral;
```



- *State Table and Excitation Table*:

r	p	PRESENT STATE	NEXT STATE	x	w	z
0	0	S1	S1	0	0	1
0	0	S2	S3	1	1	0
0	0	S3	S2	1	0	0
0	1	S1	S3	0	0	0
0	1	S2	S1	1	0	0
0	1	S3	S3	0	0	0
1	0	S1	S2	0	0	0
1	0	S2	S3	0	1	0
1	0	S3	S2	1	0	0
1	1	S1	S2	0	0	0
1	1	S2	S1	0	0	0
1	1	S3	S3	0	0	0

PRESENT STATE				NEXTSTATE				
r	p	Q_1	Q_0 (t)	Q_1	Q_0 (t+1)	x	w	z
0	0	0	0	0	0	0	0	1
0	0	0	1	1	0	1	1	0
0	0	1	0	0	1	1	0	0
0	0	1	1	X	X	X	X	X
0	1	0	0	1	0	0	0	0
0	1	0	1	0	0	1	0	0
0	1	1	0	1	0	0	0	0
0	1	1	1	X	X	X	X	X
1	0	0	0	0	1	0	0	0
1	0	0	1	1	0	0	1	0
1	0	1	0	0	1	1	0	0
1	0	1	1	X	X	X	X	X
1	1	0	0	0	1	0	0	0
1	1	0	1	0	0	0	0	0
1	1	1	0	1	0	0	0	0
1	1	1	1	X	X	X	X	X

State Assignment:
S1: Q=00 S2: Q=01
S3: Q=10

This is a Mealy FSM. The outputs x, w, z depend on the input as well as on the present state.

- *Minimization, Excitation equations (including Boolean output equations), and circuit implementation*:

$$Q_1(t+1) \leftarrow \bar{r}p\overline{Q_0(t)} + pQ_1(t) + \bar{p}Q_0(t)$$
$$Q_0(t+1) \leftarrow r\overline{Q_1(t)}\,\overline{Q_0(t)} + \bar{p}Q_1(t)$$
$$x = \bar{p}Q_1(t) + \bar{r}Q_0(t)$$
$$w = \bar{p}Q_0(t)$$
$$z = \bar{r}\bar{p}\overline{Q_1(t)}\,\overline{Q_0(t)}$$

$Q_1(t+1)$

Q_1Q_0 \ rp	00	01	11	10
00	0	1	0	0
01	1	0	0	1
11	X	X	X	X
10	0	1	1	0

$Q_0(t+1)$

Q_1Q_0 \ rp	00	01	11	10
00	0	0	1	1
01	0	0	0	0
11	X	X	X	X
10	1	0	0	1

x

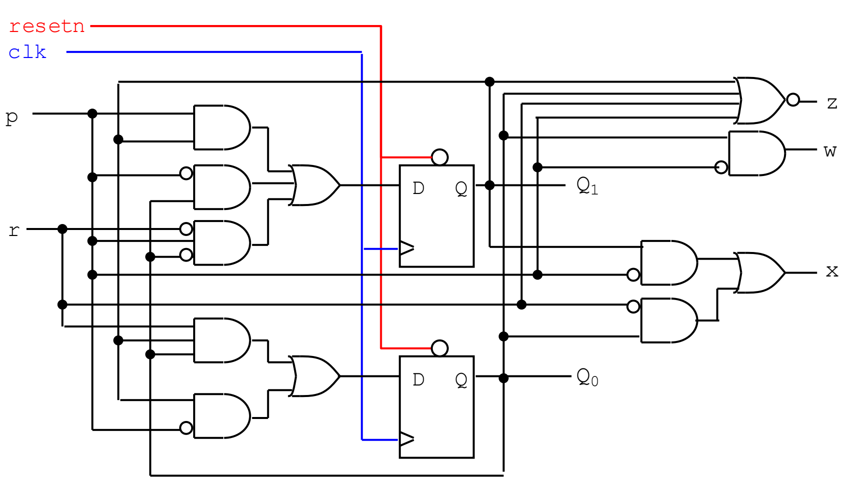
Q_1Q_0 \ rp	00	01	11	10
00	0	0	0	0
01	1	1	0	0
11	X	X	X	X
10	1	0	0	1

w

Q_1Q_0 \ rp	00	01	11	10
00	0	0	0	0
01	1	0	0	1
11	X	X	X	X
10	0	0	0	0

z

Q_1Q_0 \ rp	00	01	11	10
00	1	0	0	0
01	0	0	0	0
11	X	X	X	X
10	0	0	0	0



PROBLEM 6 (18 PTS)

- "Counting 0's" Circuit: It counts the number of bits in register A that has the value of '0'.
  - ✓ Example: for $n = 8$: if $A = 00110010$, then $C = 0101$.
  - ✓ The digital system (FSM + Datapath) is depicted below. The behavior (on the clock tick) of the generic components is as follows:

m-bit counter (modulo-n+1): If E=0, the count stays.

```
if E = 1 then
   if sclr = 1 then
      Q ← 0
   else
      Q ← Q+1
  end if;
end if;
```

n-bit Parallel access shift register: If E=0, the output is kept.

```
if E = 1 then
   if s_l = '1' then
      Q ← D
   else
      Q ← shift in 'din'  (to the right)
  end if;
end if;
```

- Complete the timing diagram where $n = 8, m = 4$. A is represented in hexadecimal format, while C is in binary format.