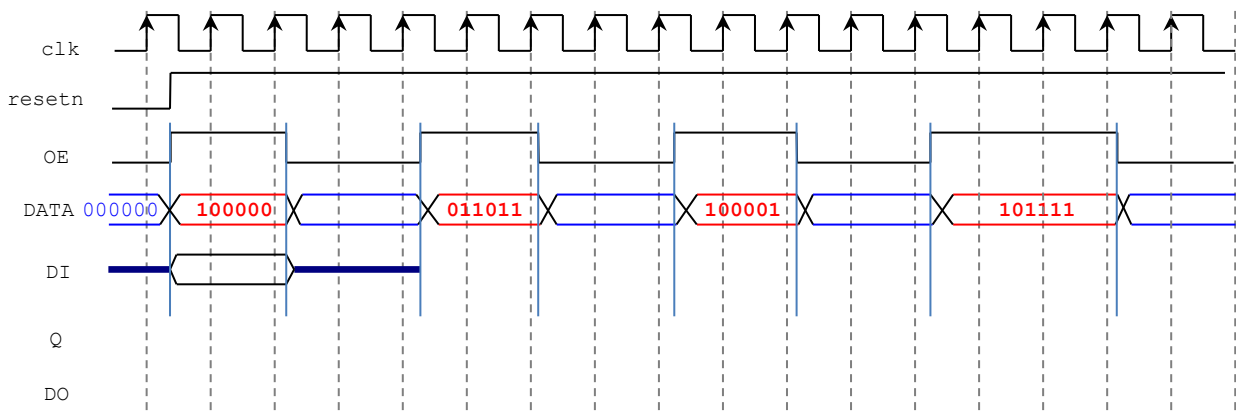


(December 13<sup>th</sup> @ 7:00 pm)

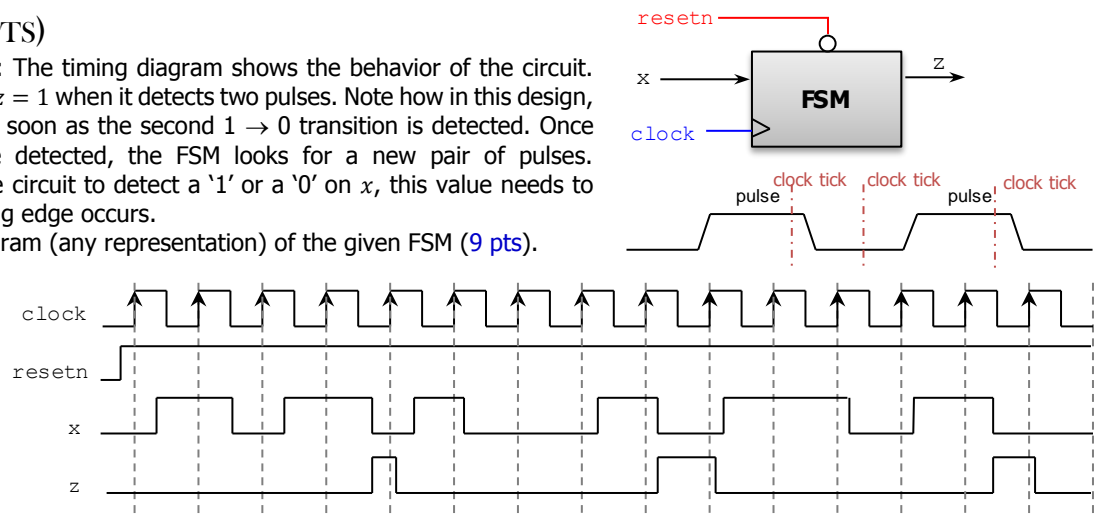
### PROBLEM 1 (12 PTS)

- 
- The timing diagram shows the signals over time, with vertical dashed lines indicating clock edges. The signals are:
- clk**: A periodic square wave clock signal.
  - resetrn**: An active-low reset signal that is high for most of the time and goes low at the first clock edge.
  - E**: An enable signal that is high from the first clock edge to the fourth, and then goes low.
  - x**: The input data signal, which is high from the first clock edge to the second, low from the second to the third, high from the third to the fourth, and low from the fourth to the fifth.
  - Q**: The output signals  $Q_0, Q_1, Q_2, Q_3$ . They all start at 0. At the first clock edge,  $Q_0$  becomes 1. At the second clock edge,  $Q_1$  becomes 1. At the third clock edge,  $Q_2$  becomes 1. At the fourth clock edge,  $Q_3$  becomes 1. When **E** goes low at the fifth clock edge, all **Q** signals return to 0.

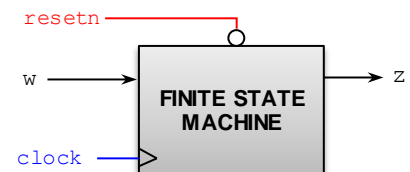
[illegible]

### 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).

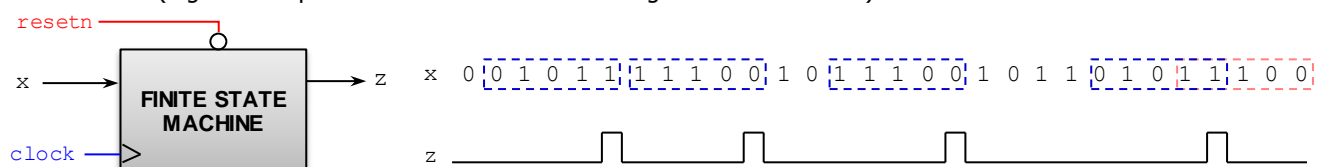


- The following FSM has 4 states, one input  $w$  and one output  $z$ . (11 pts)
  - ✓ The excitation equations are given by:
    - $Q_1(t+1) \leftarrow Q_0(t)$
    - $Q_0(t+1) \leftarrow Q_1(t) \oplus \bar{w}$
  - ✓ The output equation is given by:  $z = Q_1(t) \oplus Q_0(t) \oplus w$
  - ✓ Provide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



### 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? \_\_\_\_\_

### PROBLEM 5 (27 PTS)

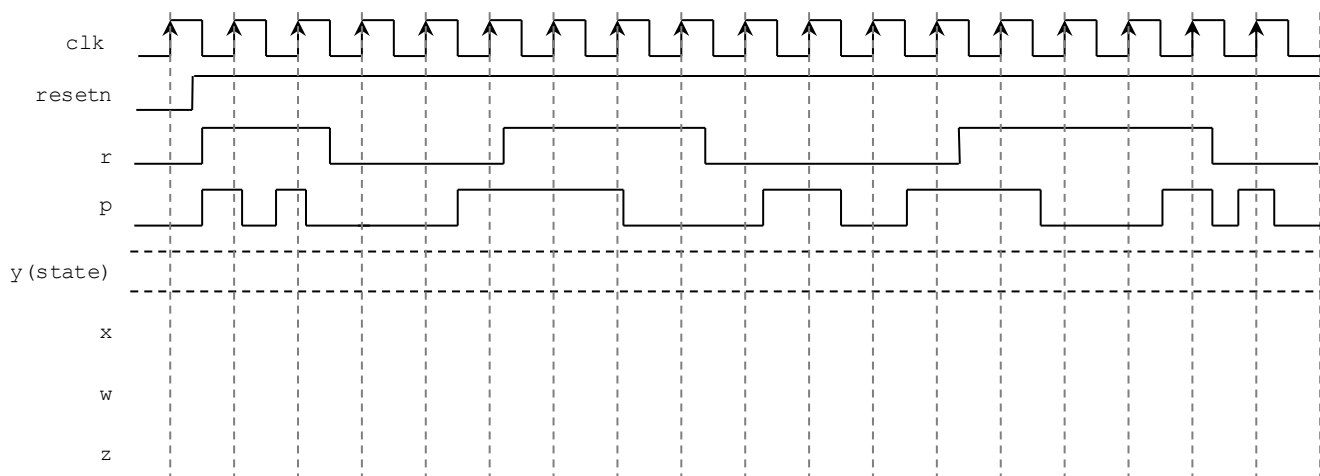
- Draw the State Diagram (in ASM form) of the FSM whose VHDL description is shown below. (5 pts)
- Complete the Timing Diagram. (7 pts)
- Provide the State Table and the Excitation Table. Is it a Mealy or a Moore FSM? (6 pts).
- Provide the excitation equations and the Boolean output equations (simplify 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;
```



## 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.

