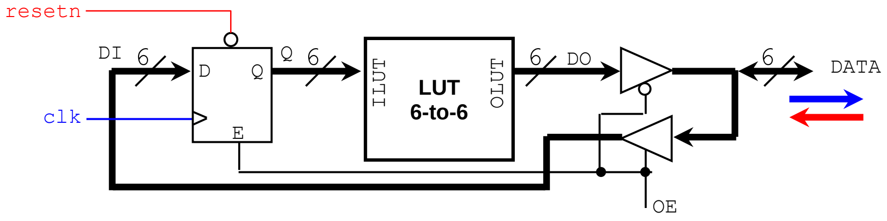
Final Exam

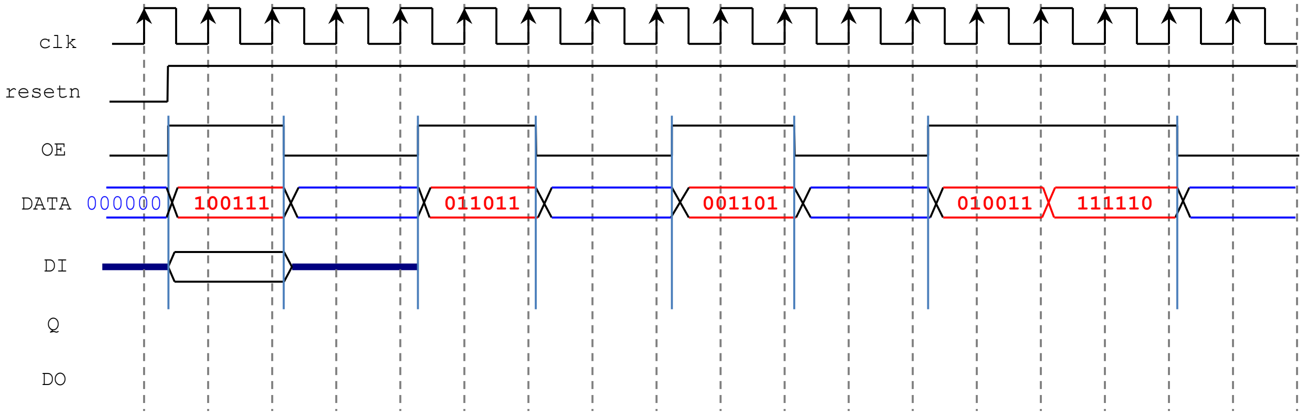
(December 13th @ 7:00 pm)

Presentation and clarity are very important! Show your procedure!

PROBLEM 1 (10 PTS)

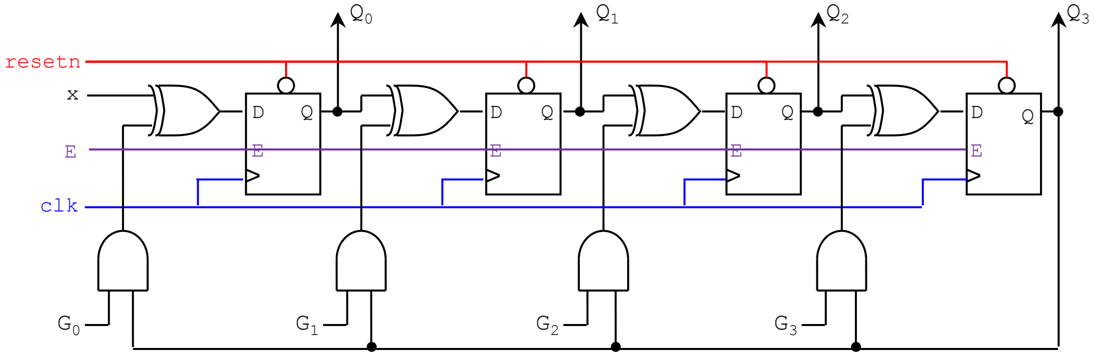
- Given the following circuit, complete the timing diagram.
  The LUT 6-to-6 implements the following function: $OLUT = \lceil sqrt(ILUT) \rceil$, where $ILUT$ is a 6-bit unsigned number.
  For example, $ILUT = 53\ (110101_2) \rightarrow OLUT = \lceil sqrt(53) \rceil = 8\ (001000_2)$

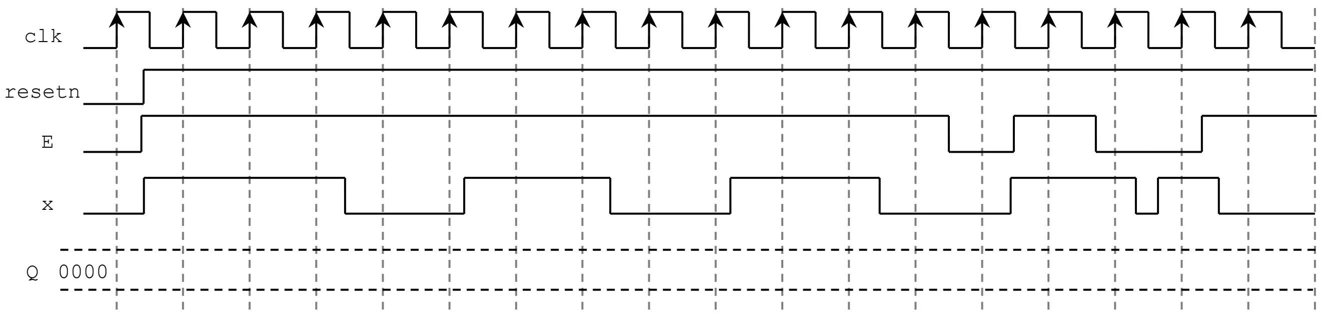




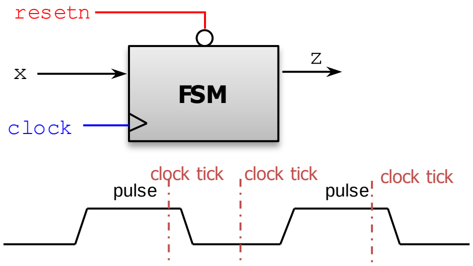
PROBLEM 2 (12 PTS)

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

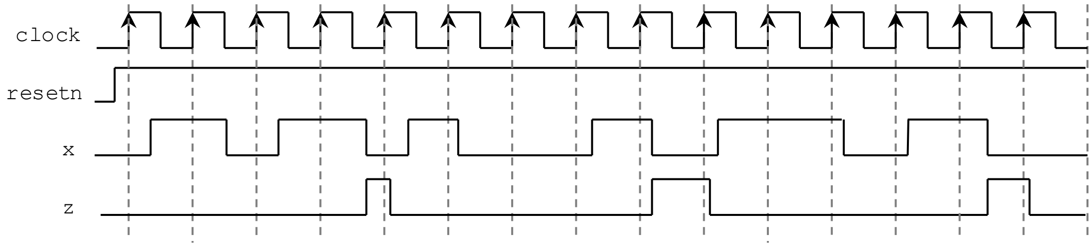


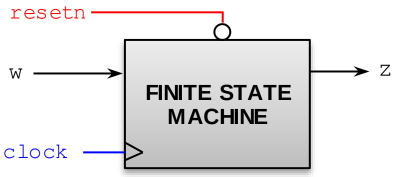


PROBLEM 3 (21 PTS)



- 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: The circuit detects a '1' or a '0' on x, this happens on the rising clock edge.
  - ✓ Draw the State Diagram (any representation) of this FSM (9 pts).

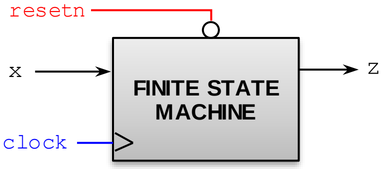


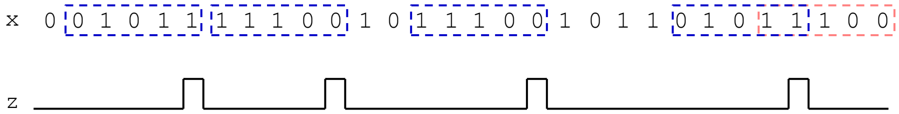


- The following FSM has 4 states, one input w and one output z. (12 pts)
  - ✓ The excitation equations are given by:
    - $Q_1(t+1) \leftarrow Q_0(t)$
    - $Q_0(t+1) \leftarrow \overline{Q_1(t)} \oplus w$
  - ✓ The output equation is given by: $z = Q_0(t) + (Q_1(t) \oplus w)$
  - ✓ Provide the State Diagram (any representation) and the Excitation Table.
  - ✓ Sketch the Finite State Machine circuit.

PROBLEM 4 (11 PTS)

- Sequence detector: This FSM has to generate $z = 1$ when it detects the sequence 01011 or 11100. 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).





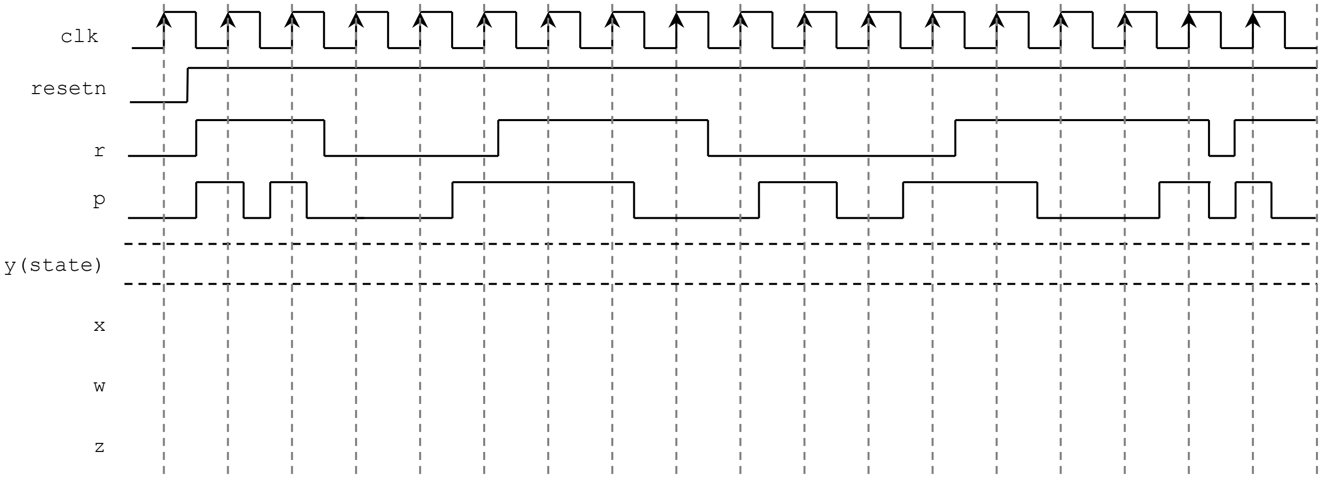
- ✓ Draw the State Diagram (any representation) and provide the State Table of this circuit with input x and output z.
- ✓ Which type is this FSM? (Mealy) (Moore) Why? ____________________

PROBLEM 5 (28 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;
```



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 behavior (on the clock tick) of the generic components is as follows:

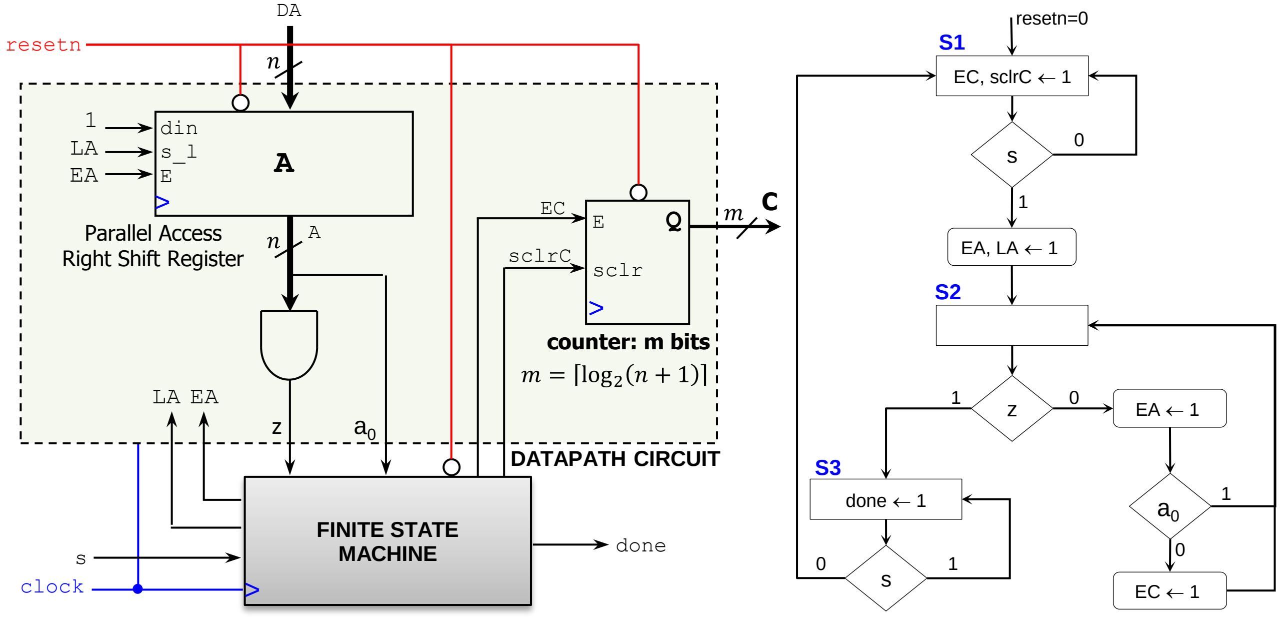
m-bit counter (modulo-n+1): If E=0, the count stays.

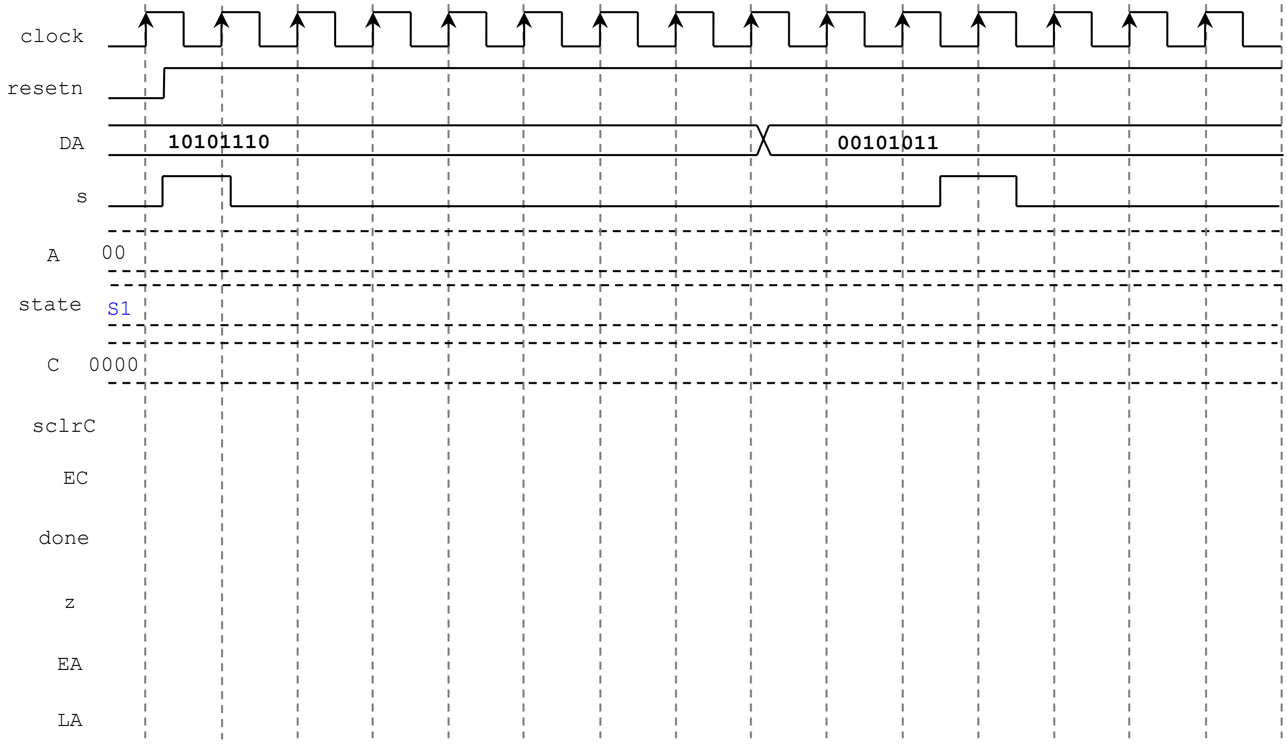
```
if E = 1 then
   if sclr = 1 then
      Q ← 0
   else
      Q ← Q+1
  end if;
end if;
```

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.


DA
resetn
n
1
LA
EA
din
s_l
E
A
Parallel Access
Right Shift Register
n
A
EC
E
Q
sclrC
sclr
m
C
counter: m bits
$m = \lceil \log_2(n+1) \rceil$
LA EA
z
a_0
DATAPATH CIRCUIT
FINITE STATE MACHINE
s
clock
done
resetn=0
S1
EC, sclrC ← 1
s
0
1
EA, LA ← 1
S2
z
1
0
EA ← 1
S3
done ← 1
a_0
1
0
0
s
1
EC ← 1


clock
resetn
DA
10101110
00101011
s
A
00
state
S1
C
0000
sclrC
EC
done
z
EA
LA