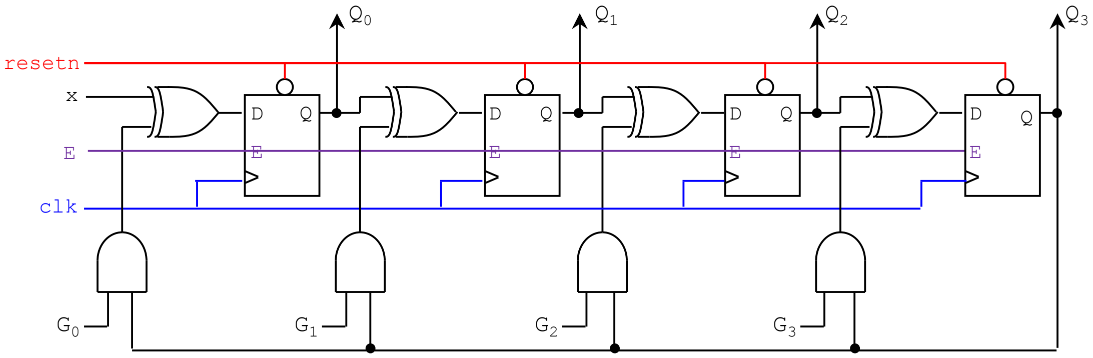
Solutions - Final Exam

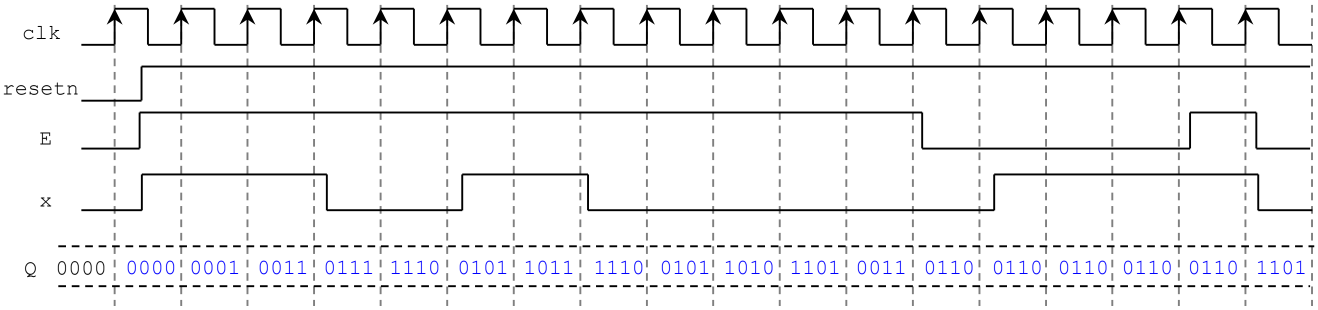
(April 21st @ 7:00 pm)

Presentation and clarity are very important! Show your procedure!

PROBLEM 1 (12 PTS)

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

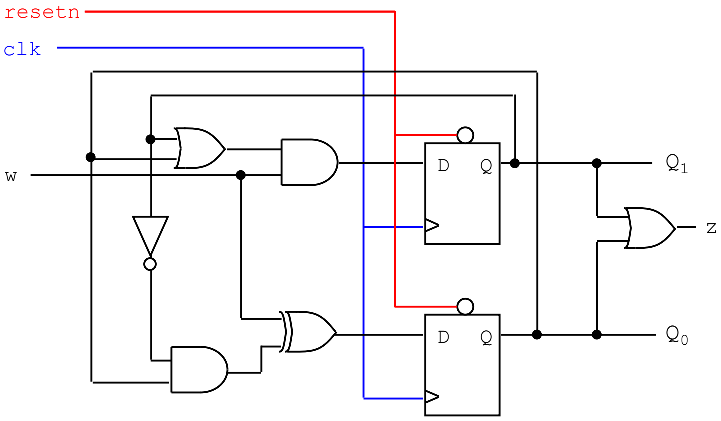




PROBLEM 2 (20 PTS)

- Provide the State Diagram (any representation), the Excitation Table, and the Excitation equations of the following Finite State Machine: (12 pts)

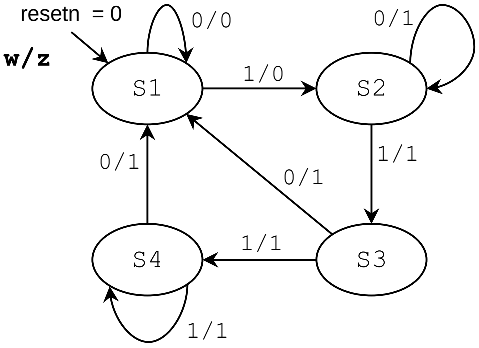
$$Q_1(t+1) = (Q_1 + Q_0)w$$
$$Q_0(t+1) = (\overline{Q_1}Q_0) \oplus w$$
$$z = Q_1 + Q_0$$



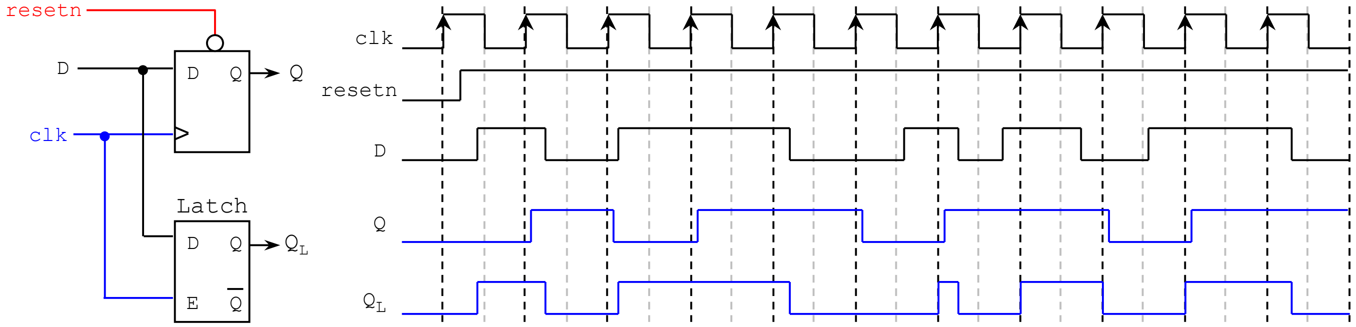
- *State Assignment:*
  - ✓ S1: Q = 00
  - ✓ S2: Q = 01
  - ✓ S3: Q = 10
  - ✓ S4: Q = 11

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

w	PRESENT STATE	NEXT STATE	z
0	S1	S1	0
0	S2	S2	1
0	S3	S1	1
0	S4	S1	1
1	S1	S2	0
1	S2	S3	1
1	S3	S4	1
1	S4	S4	1

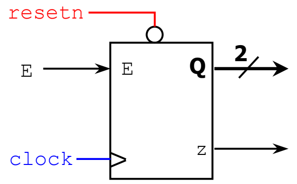


- Complete the timing diagram of the circuit shown below: (8 pts)



PROBLEM 5 (18 PTS)

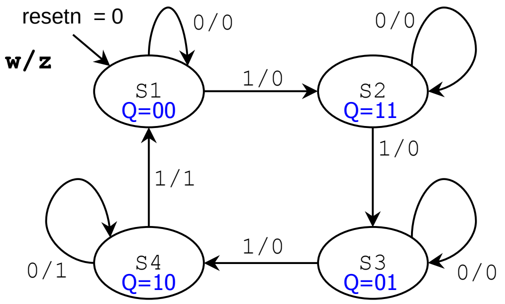
- Design a counter using a Finite State Machine (FSM):
  Counter features:
  ✓ Count: **00**, 11, 01, 10, **00**, 11, 01, …
  ✓ *resetn*: Asynchronous active-low input signal. It initializes the count to "00"
  ✓ Input E: Synchronous input that increases the count when it is set to '1'.
  ✓ output z: It becomes '1' when the count is 10.



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. (3 pts)

- *State Diagram, State Table, and Excitation Table:*



E	PRESENT STATE	NEXT STATE	z
0	S1	S1	0
0	S2	S2	0
0	S3	S3	0
0	S4	S4	1
1	S1	S2	0
1	S2	S3	0
1	S3	S4	0
1	S4	S1	1

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

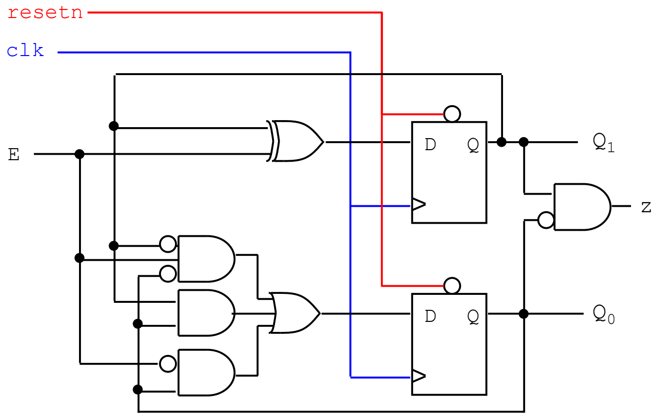
- *State Assignment:*

S1: Q = 00 S3: Q = 01
S2: Q = 11 S4: Q = 10

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

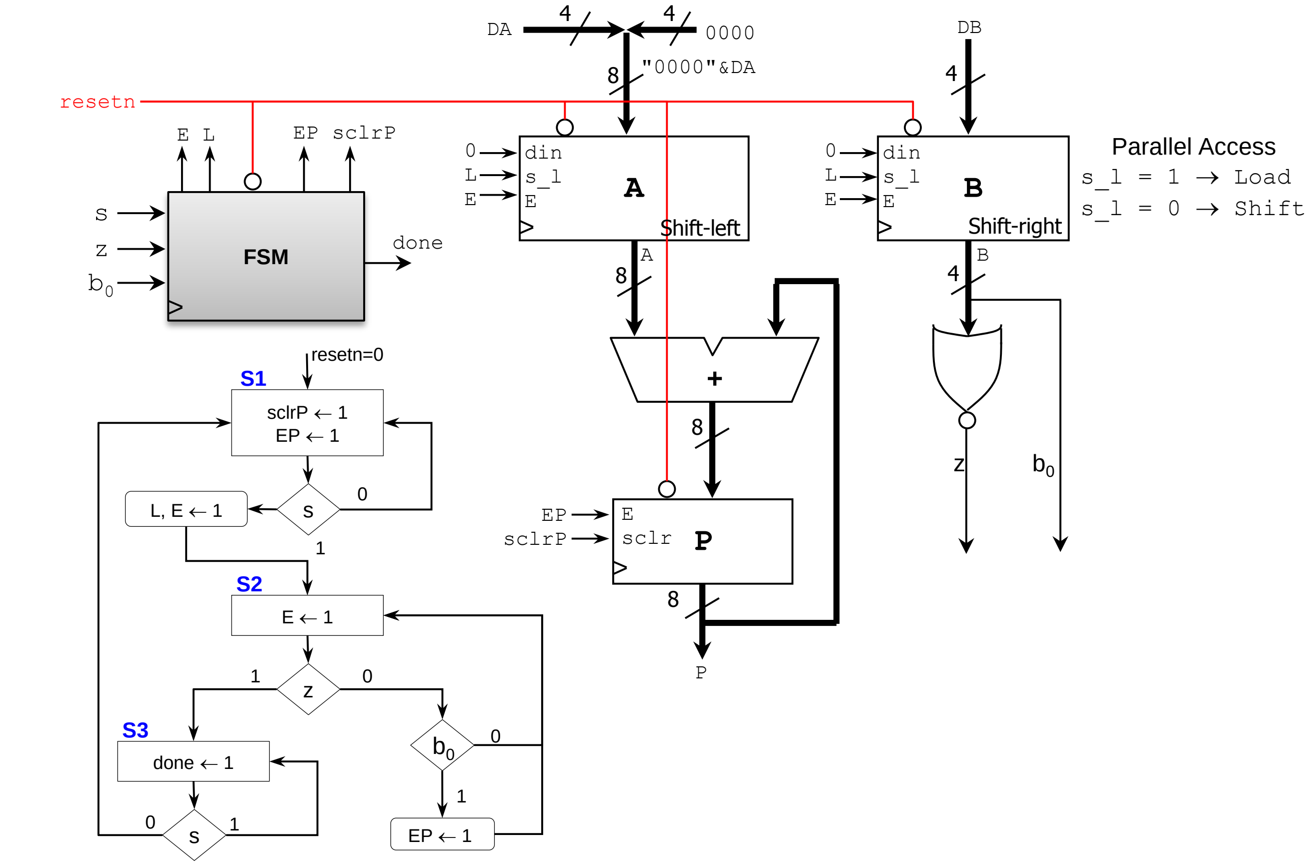
- *Excitation equations, minimization and circuit implementation:*

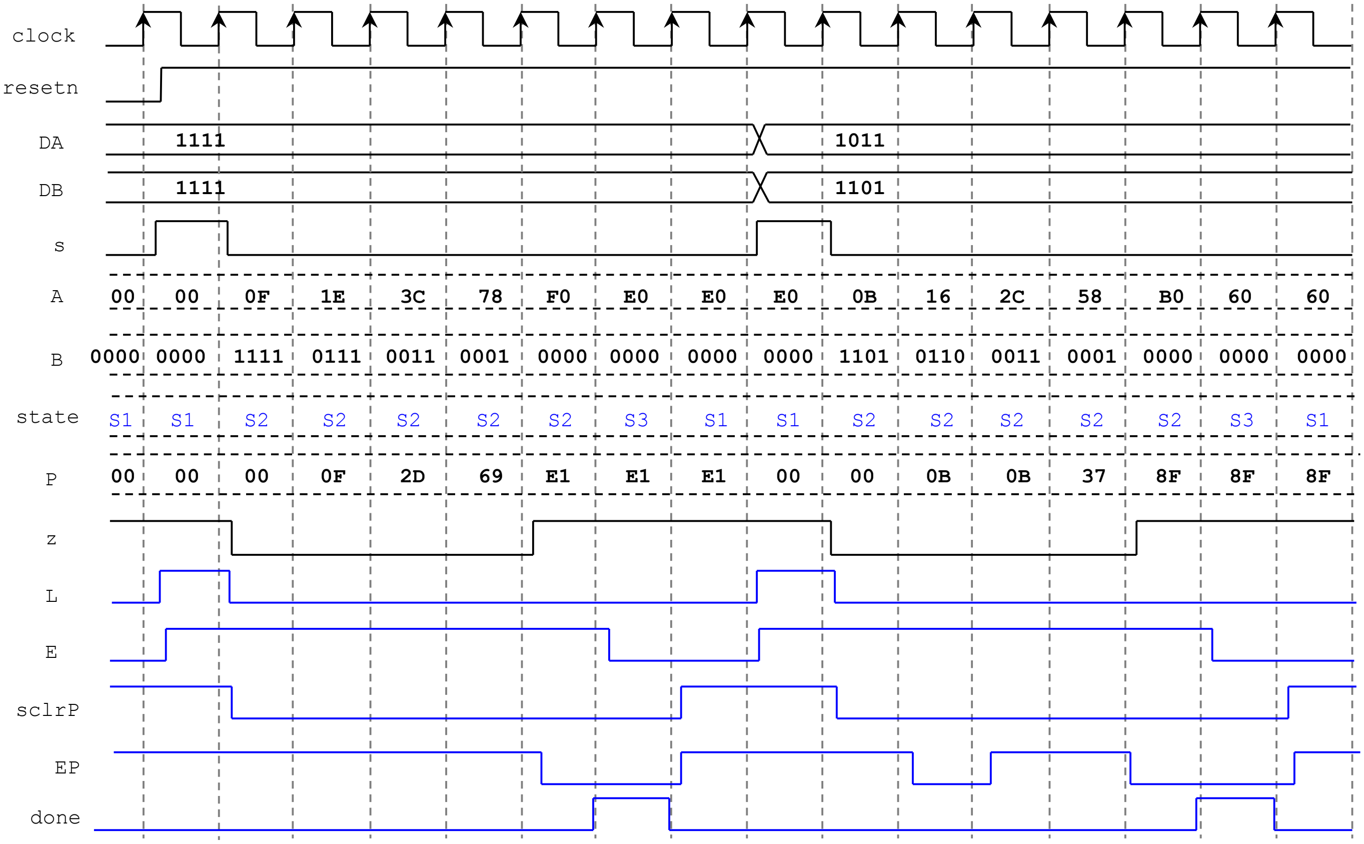
$Q_1(t+1) = \bar{E}Q_1 + E\overline{Q_1} = E \oplus Q_1$
$Q_0(t+1) = E\overline{Q_1}\,\overline{Q_0} + \bar{E}Q_0 + Q_1Q_0$
$z = \overline{Q_0}Q_1$



PROBLEM 3 (20 PTS)

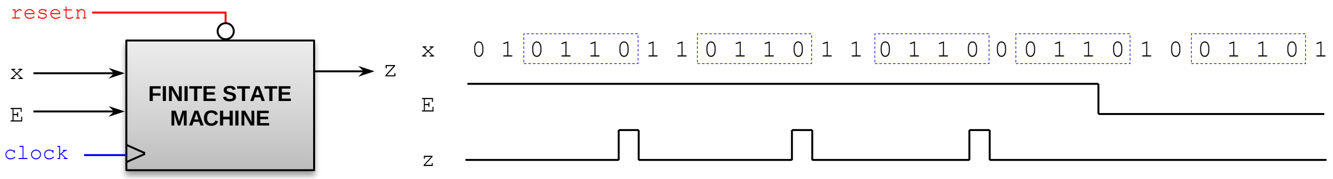
- Iterative unsigned multiplier: Complete the following timing diagram of the following digital circuit that includes an FSM (in ASM form) and a datapath circuit.
  Register (for P): $sclr$: synchronous clear. Here, if $sclr = E = 1$, the register contents are initialized to 0.
  Parallel access shift register (for A and B): If $E = 1$: $s\_l = 1 \rightarrow$ Load, $s\_l = 0 \rightarrow$ Shift

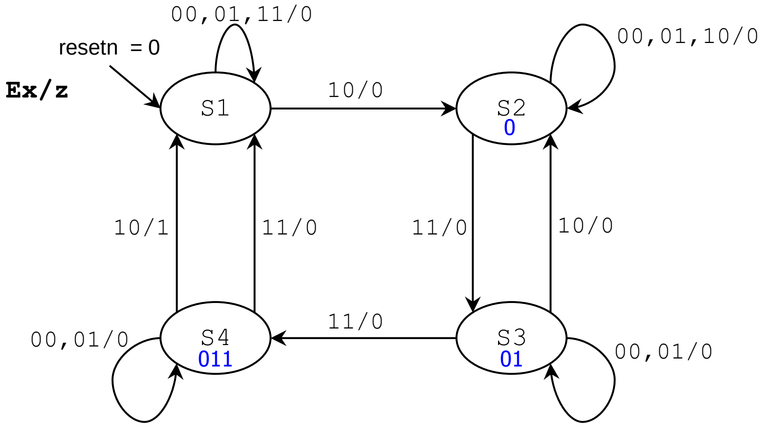




PROBLEM 4 (15 PTS)

- Sequence detector: Draw the State Diagram (any representation) and the Excitation Table of a circuit with an input x and output z. The machine has to generate $z = 1$ when it detects the sequence 0110. Once the sequence is detected, the circuits looks for a new sequence.
- The signal E is an input enable: It validates the input x, i.e., if $E = 1$, x is valid, otherwise x is not valid.


resetn
x
E
clock
FINITE STATE MACHINE
z
x 0 1 0 1 1 0 1 1 0 1 1 0 1 1 0 1 1 0 0 0 1 1 0 1 0 0 1 1 0 1
E
z



- ✓ S1: Q = 00
- ✓ S2: Q = 01
- ✓ S3: Q = 10
- ✓ S4: Q = 11

		PRESENT	NEXT	
E	x	STATE	STATE	z
0	0	S1	S1	0
0	0	S2	S2	0
0	0	S3	S3	0
0	0	S4	S4	0
0	1	S1	S1	0
0	1	S2	S2	0
0	1	S3	S3	0
0	1	S4	S4	0
1	0	S1	S2	0
1	0	S2	S2	0
1	0	S3	S2	0
1	0	S4	S1	1
1	1	S1	S1	0
1	1	S2	S3	0
1	1	S3	S4	0
1	1	S4	S1	0

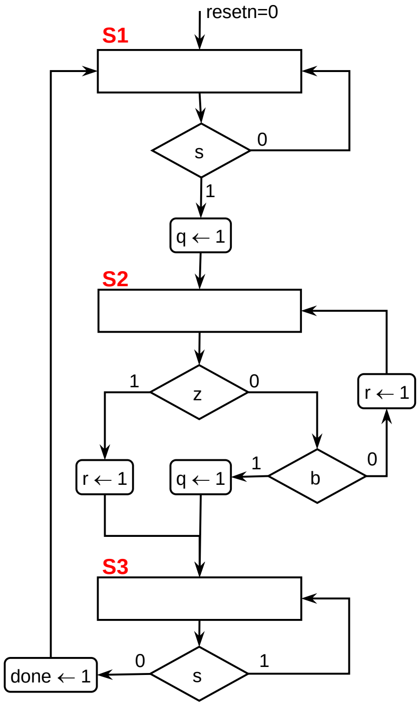
		PRESENT STATE	NEXTSTATE	
E	x	Q_1Q_0(t)	Q_1Q_0(t+1)	z
0	0	0 0	0 0	0
0	0	0 1	0 1	0
0	0	1 0	1 0	0
0	0	1 1	1 1	0
0	1	0 0	0 0	0
0	1	0 1	0 1	0
0	1	1 0	1 0	0
0	1	1 1	1 1	0
1	0	0 0	0 1	0
1	0	0 1	0 1	0
1	0	1 0	0 1	0
1	0	1 1	0 0	1
1	1	0 0	0 0	0
1	1	0 1	1 0	0
1	1	1 0	1 1	0
1	1	1 1	0 0	0

PROBLEM 6 (15 PTS)

- Draw the State Diagram (in ASM form) of the FSM whose VHDL description in shown below. Is it a Mealy or Moore FSM?
- Complete the Timing Diagram.

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

entity circ is
   port ( clk, resetn: in std_logic;
          s, b, z: in std_logic;
          done, r, q: out std_logic);
end circ;
```

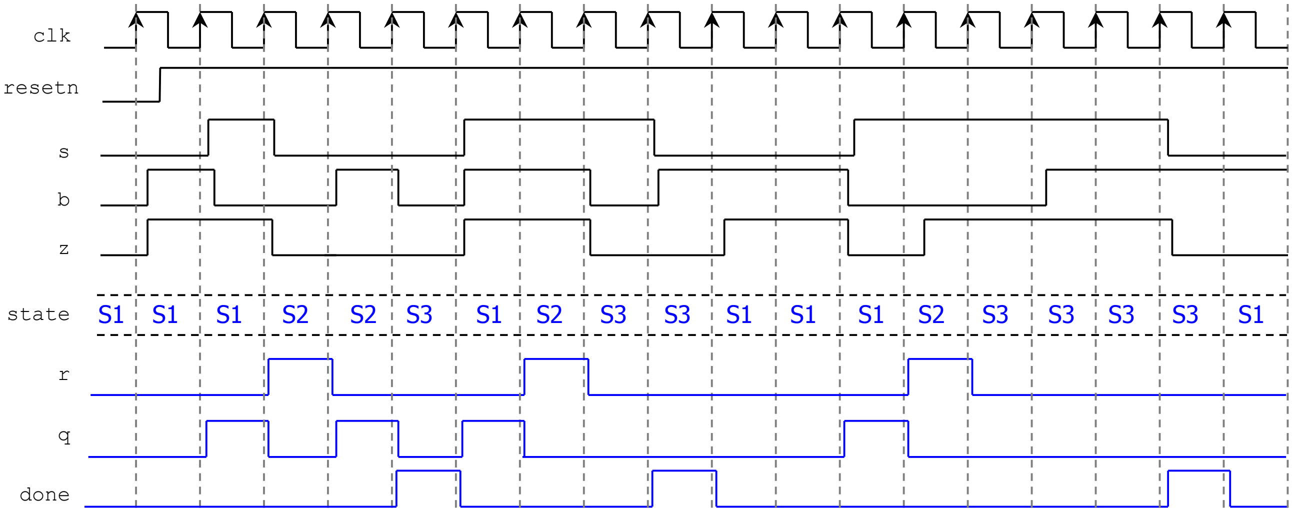


```
architecture behavioral of circ is
   type state is (S1, S2, S3);
   signal y: state;
begin
  Transitions: process (resetn, clk, s, b, z)
  begin
     if resetn = '0' then y <= S1;
     elsif (clk'event and clk = '1') then
        case y is
          when S1 =>
            if s = ‘1’ then y <= S2;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

          when S2 =>
            if z = '1' then
               y <= S3;
            else
               if b = ‘0’ then y <= S2; else y <= S3; end if;
            end if;

          when S3 =>
            if s = '1' then y <= S3; else y <= S1; end if;
        end case;
     end if;
  end process;

  Outputs: process (y, z, b)
  begin
      done <= ‘0’; q <= ‘0’; r <= ‘0’;
      case y is
         when S1 => if s = ‘1’ then q <= ‘1’; end if;
         when S2 => if z = ‘0’ then
                       if b = ‘1’ then
                          q <= ‘1’;
                       else
                          r <= ‘1’;
                       end if;
                    else
                       r <= ‘1’;
                    end if;
         when S3 => if s = ‘0’ then done <= ‘1’; end if;
      end case;
  end process;
end behavioral;
```



The outputs ‘done’, ‘q’, ‘r’ depend on the present state as well as on the inputs ⇒ Mealy FSM.