

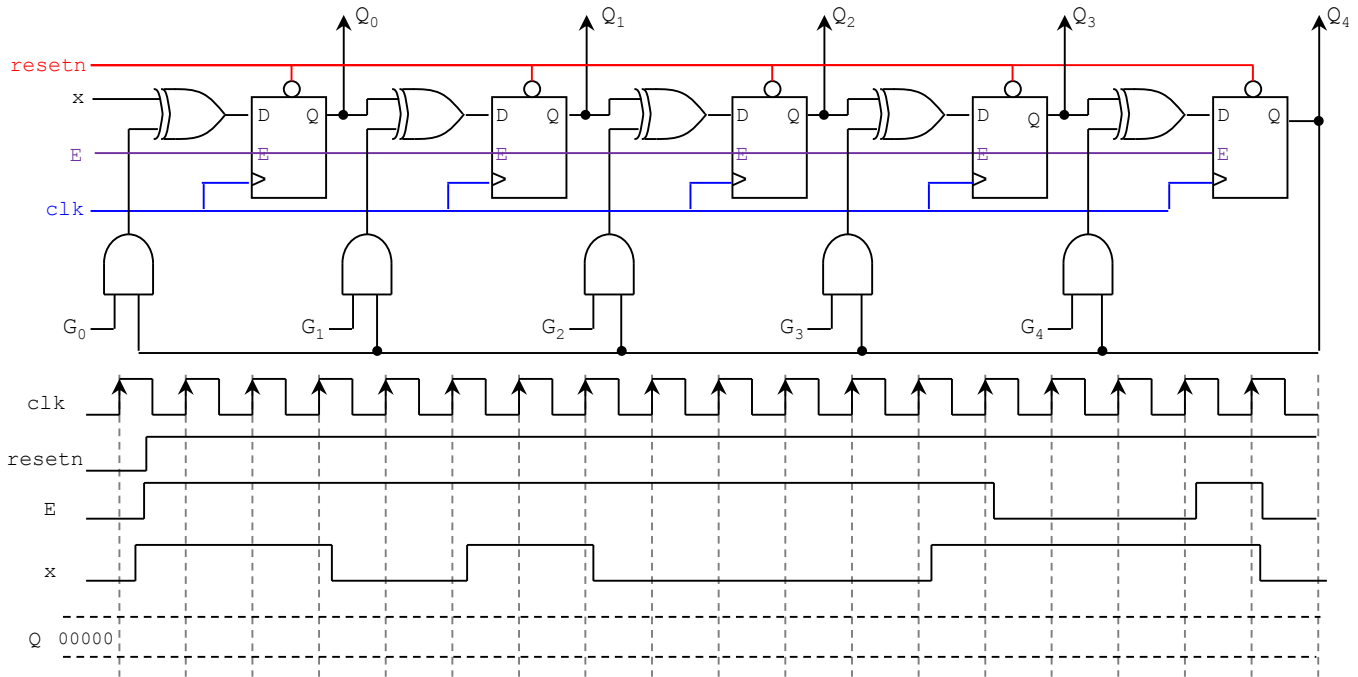
Final Exam

(April 23rd @ 7:00 pm)

Presentation and clarity are very important! Show your procedure!

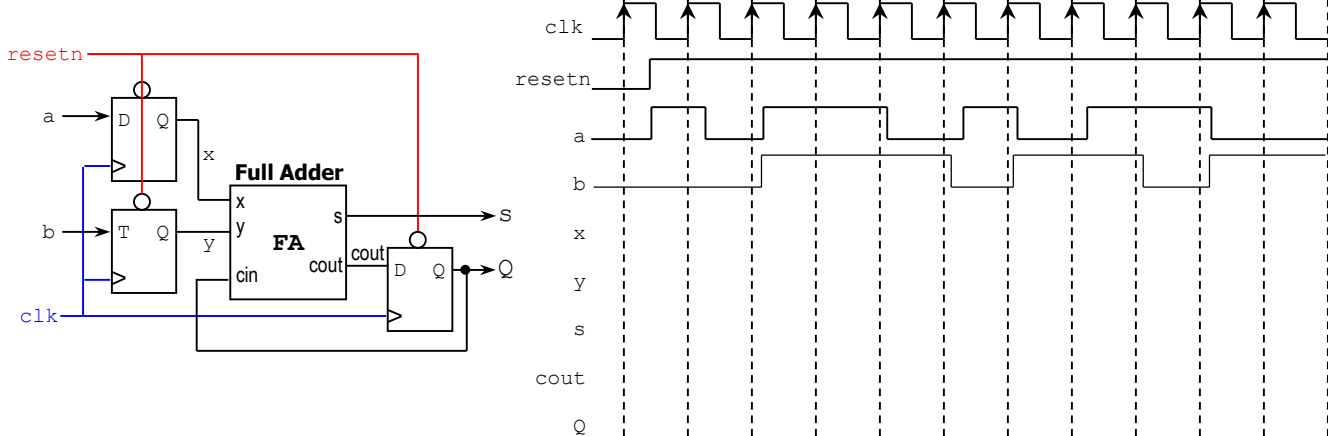
PROBLEM 1 (16 PTS)

- Complete the timing diagram of the following circuit. $G = G_4G_3G_2G_1G_0 = 10101$. $Q = Q_4Q_3Q_2Q_1Q_0$

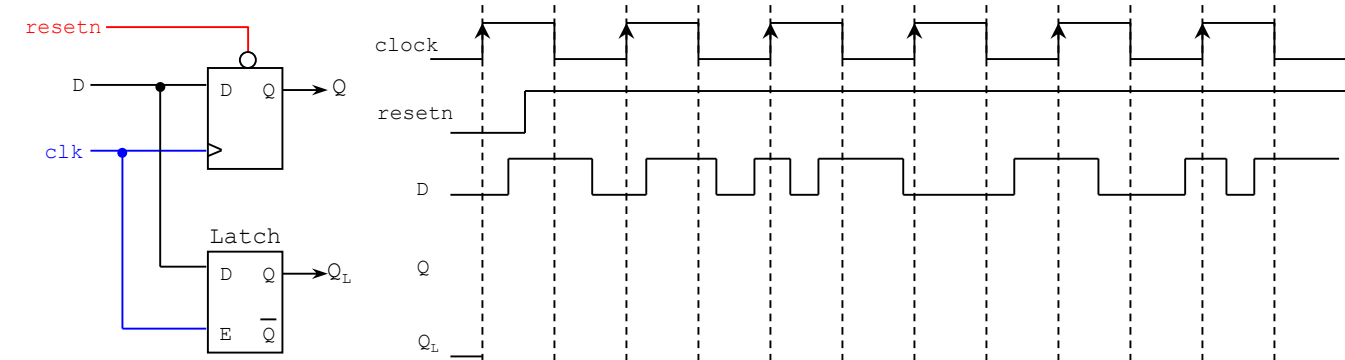


PROBLEM 2 (16 PTS)

- Complete the timing diagram of the following circuit that includes a T-type flip flop. (8 pts)

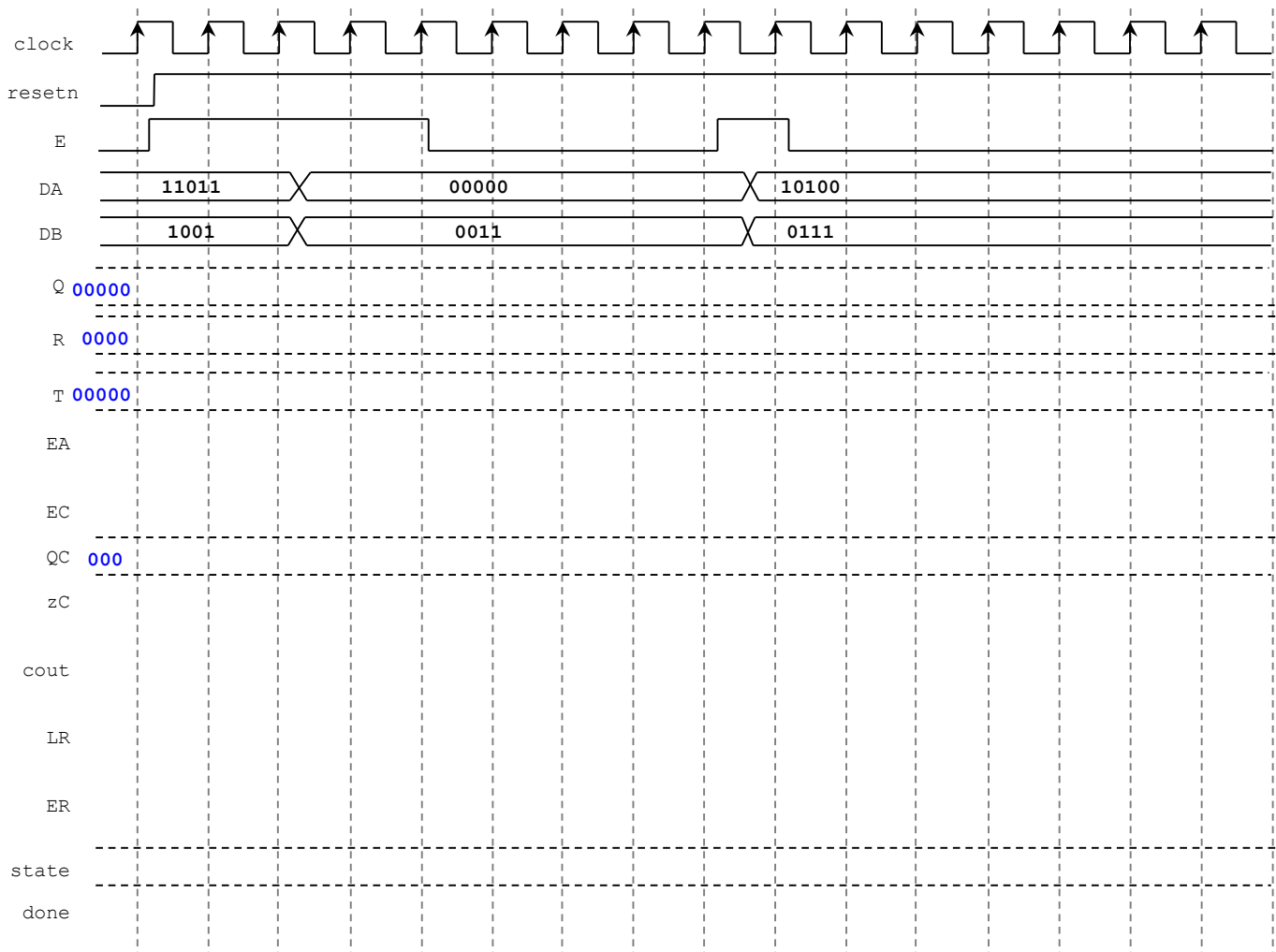
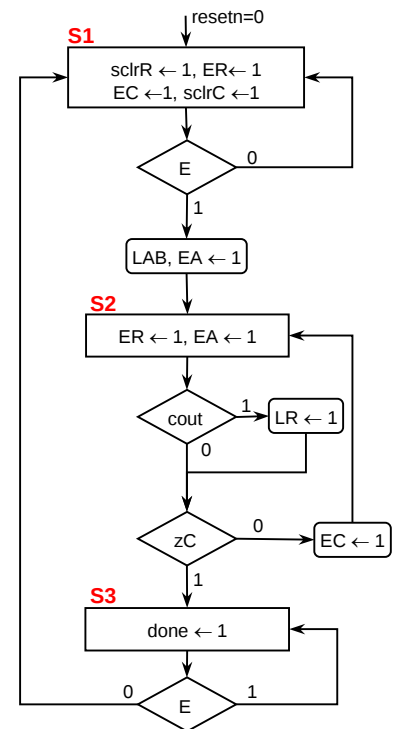
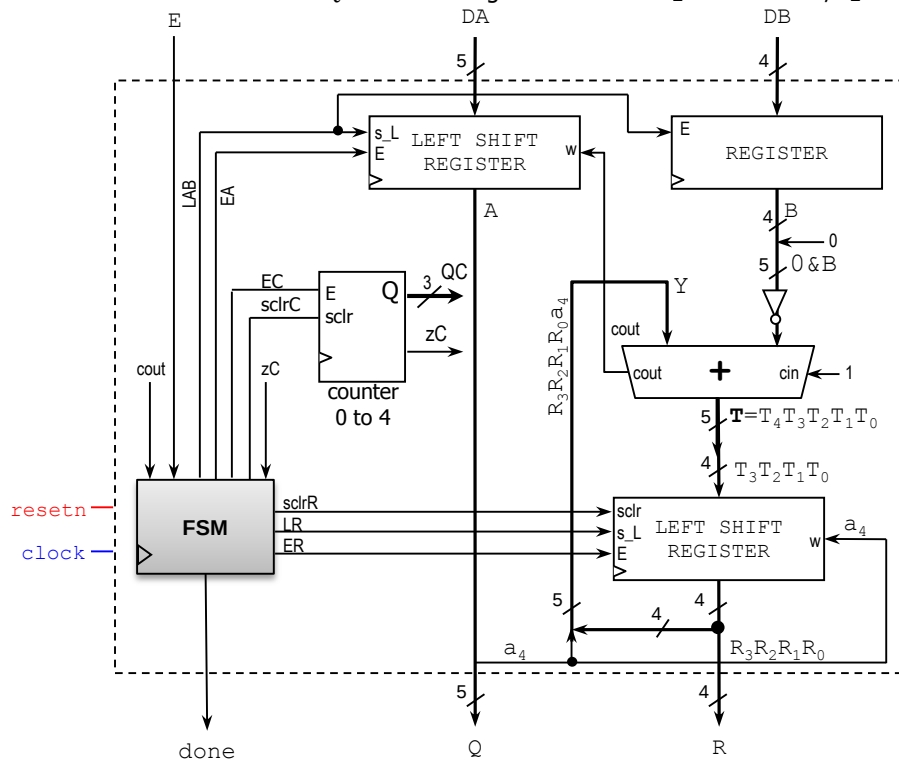


- Complete the timing diagram of the circuit shown below: (8 pts)



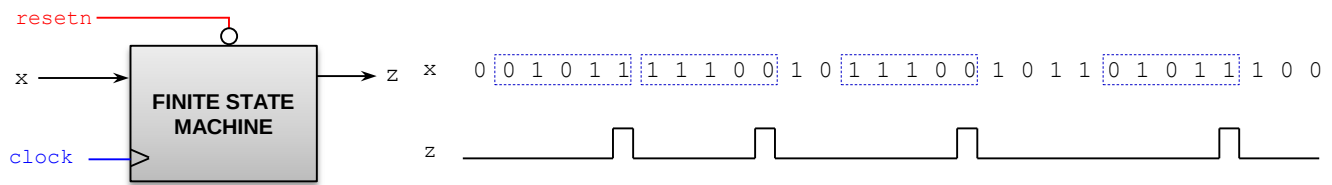
PROBLEM 3 (20 PTS)

- Complete the timing diagram of the following digital circuit that includes an FSM (in ASM form) and a datapath circuit. Counter: $E = sclr = 1 \rightarrow Q = 0$. Shift Register: If $E = 1$: $s_L = 0 \rightarrow \text{shift}$, $s_L = 1 \rightarrow \text{load}$.



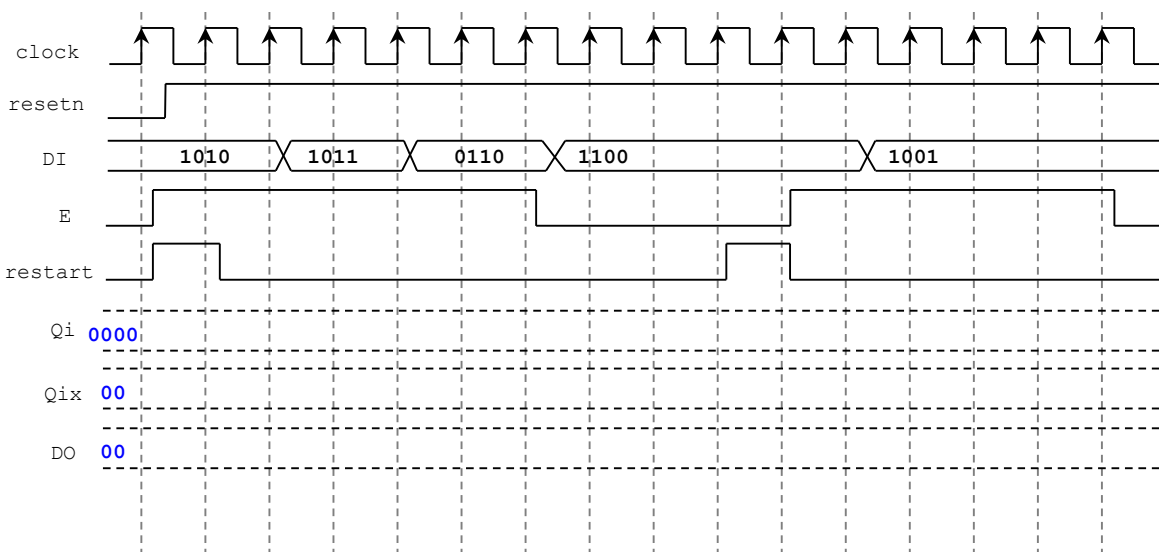
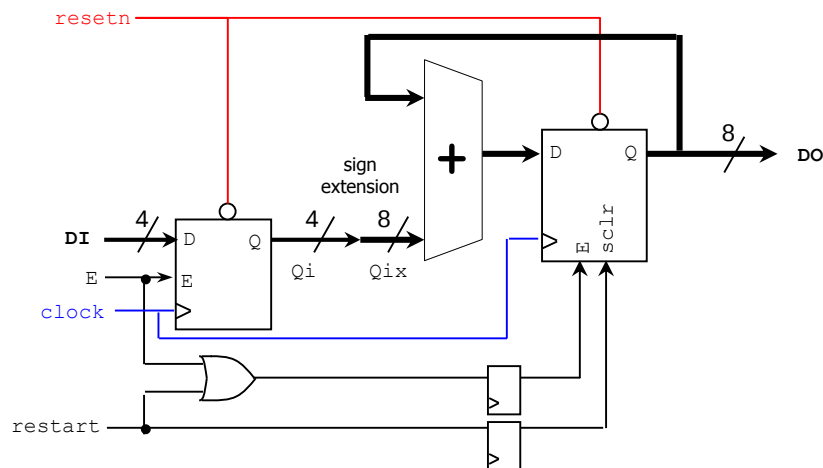
PROBLEM 4 (16 PTS)

- 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 01011 or 11100.



PROBLEM 5 (16 PTS)

- Complete the timing diagram of the following circuit. Provide DO and Q_{ix} as hexadecimal values.
- $sclr$: Synchronous clear. If $E = sclr = 1$, then the register output is cleared (set to 0).



PROBLEM 6 (16 PTS)

- Draw the State Diagram (in ASM form) of the FSM whose VHDL description is 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 = '1' then y <= S3; else y <= S2; 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 => q <= '1';
            when S2 => if z = '0' then
                            r <= '1';
                            if b = '1' then q <= '1'; end if;
                        end if;
            when S3 => done <= '1';
        end case;
    end process;
end behavioral;
```

