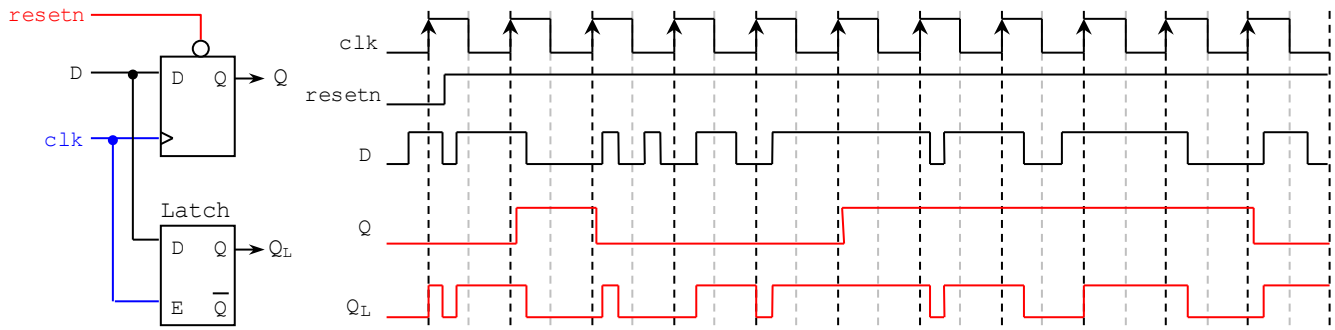
Solutions Homework 3

(Due date: March 19th @ 5:30 pm)

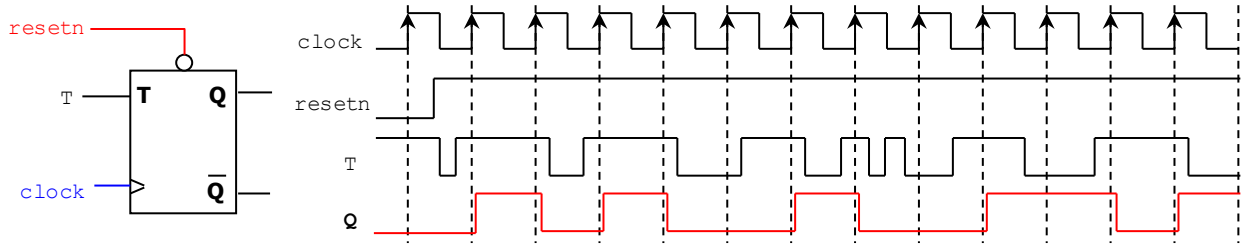
Presentation and clarity are very important! Show your procedure!

PROBLEM 1 (12 PTS)

- Complete the timing diagram of the circuit shown below. (7 pts)



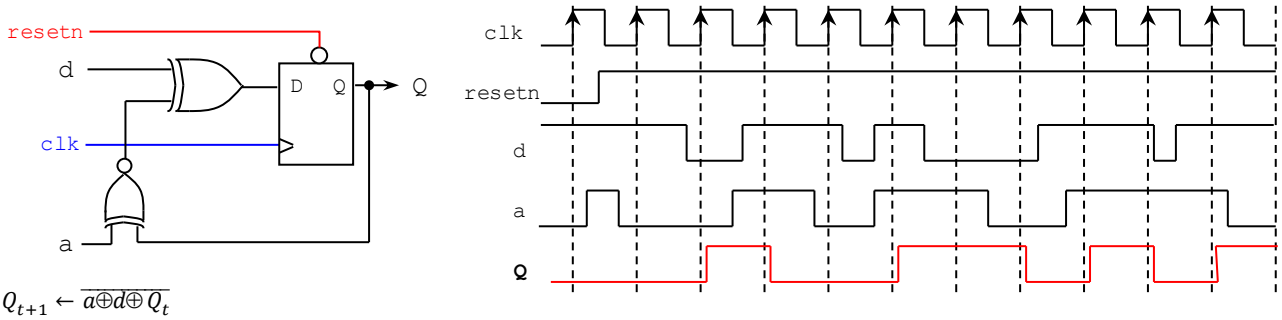
- Complete the timing diagram of the circuit shown below: (5 pts)



$Q_{t+1} \leftarrow T \oplus Q_t$

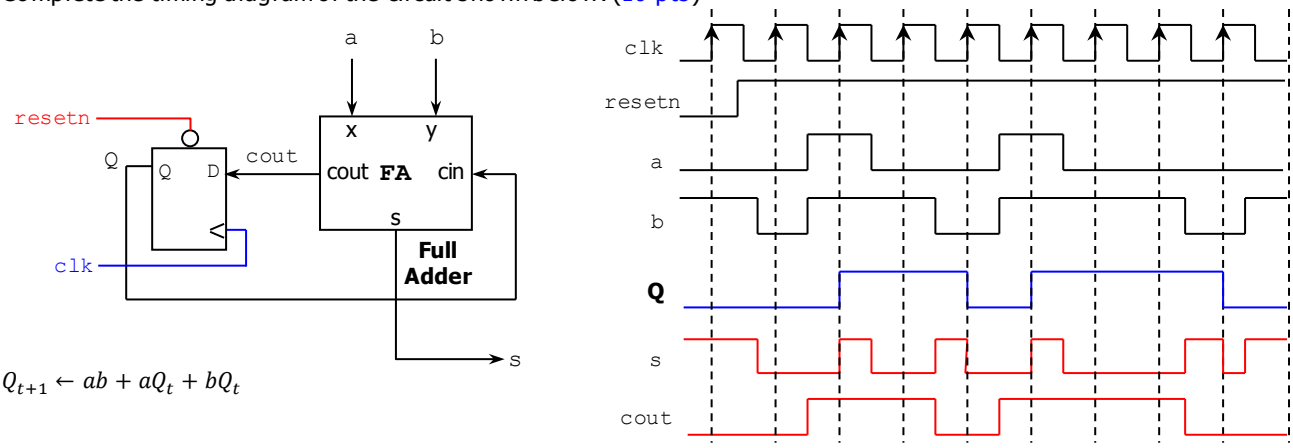
PROBLEM 2 (17 PTS)

- Complete the timing diagram of the circuit shown below: (7 pts)



$Q_{t+1} \leftarrow \overline{a \oplus d \oplus Q_t}$

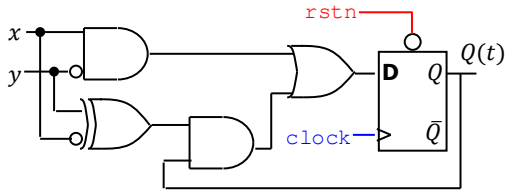
- Complete the timing diagram of the circuit shown below: (10 pts)



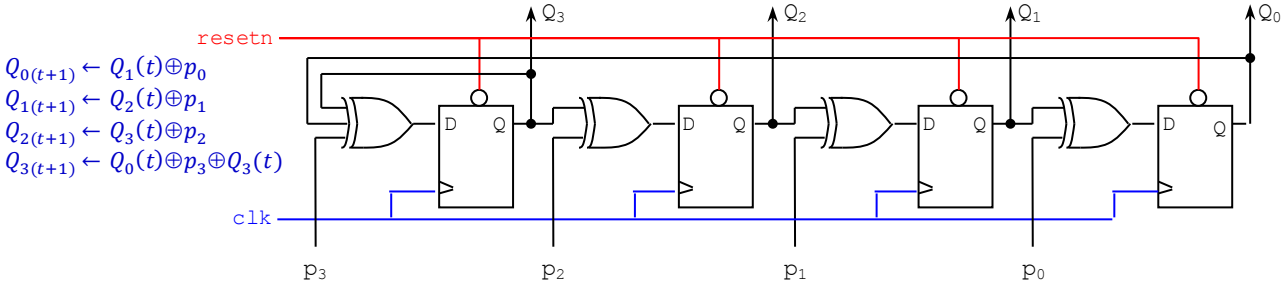
$Q_{t+1} \leftarrow ab + aQ_t + bQ_t$

PROBLEM 3 (16 PTS)

- With a D flip flop and logic gates, sketch the circuit whose excitation equation is given by:
  ✓ $Q(t+1) \leftarrow x\bar{y} + Q(t)(\bar{x} \oplus y)$ (4 pts)



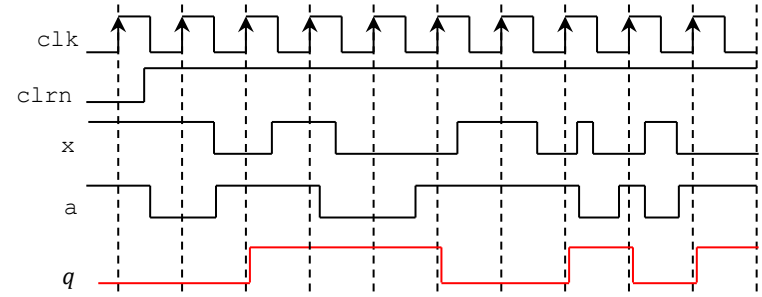
- Given the following circuit, get the excitation equations for each flip flop output $Q = Q_3Q_2Q_1Q_0$ (6 pts)



$Q_{0(t+1)} \leftarrow Q_1(t) \oplus p_0$
$Q_{1(t+1)} \leftarrow Q_2(t) \oplus p_1$
$Q_{2(t+1)} \leftarrow Q_3(t) \oplus p_2$
$Q_{3(t+1)} \leftarrow Q_0(t) \oplus p_3 \oplus Q_3(t)$

- Complete the timing diagram of the circuit whose VHDL description is shown below. Also, get the excitation equation for q.

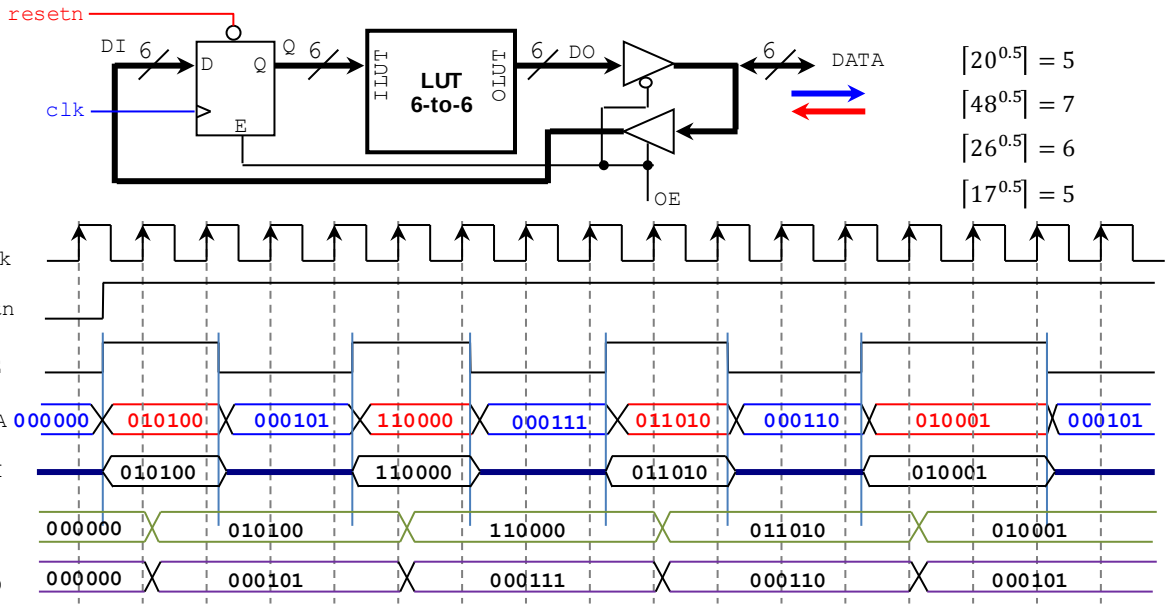
```
library ieee;
use ieee.std_logic_1164.all;

entity circ is
  port (clrn, clk, a: in std_logic;
        q: out std_logic);
end circ;

architecture t of circ is

   signal qt: std_logic;

begin
   process (clrn, clk, x, a)
   begin
     if clrn = '0' then
        qt <= '0';
     elsif (clk'event and clk = '1') then
        if x = '0' then
           qt <= not(a) xor not(qt);
        end if;
     end if;
   end process;
   q <= qt;
end t;
```



$q(t+1) \leftarrow \bar{x}.a \oplus q(t) + xq(t)$

PROBLEM 4 (18 PTS)

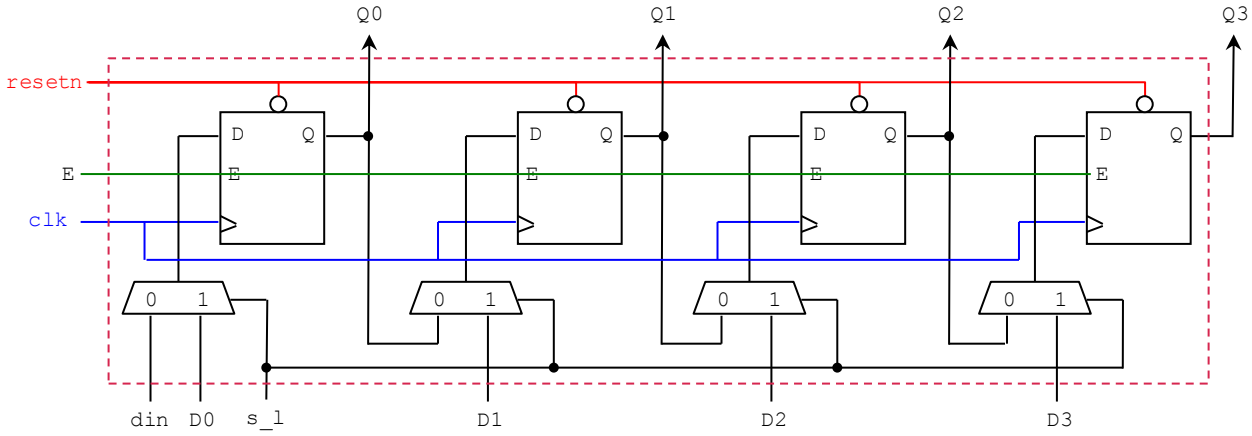
- Given the following circuit, complete the timing diagram (signals DO, Q and $DATA$).
  The LUT 6-to-6 implements the following function: $OLUT = \lceil ILUT^{0.5} \rceil$, where $ILUT$ is an unsigned number.
  For example: $ILUT = 35\ (100011_2) \rightarrow OLUT = \lceil 35^{0.85} \rceil = 6\ (000110_2)$

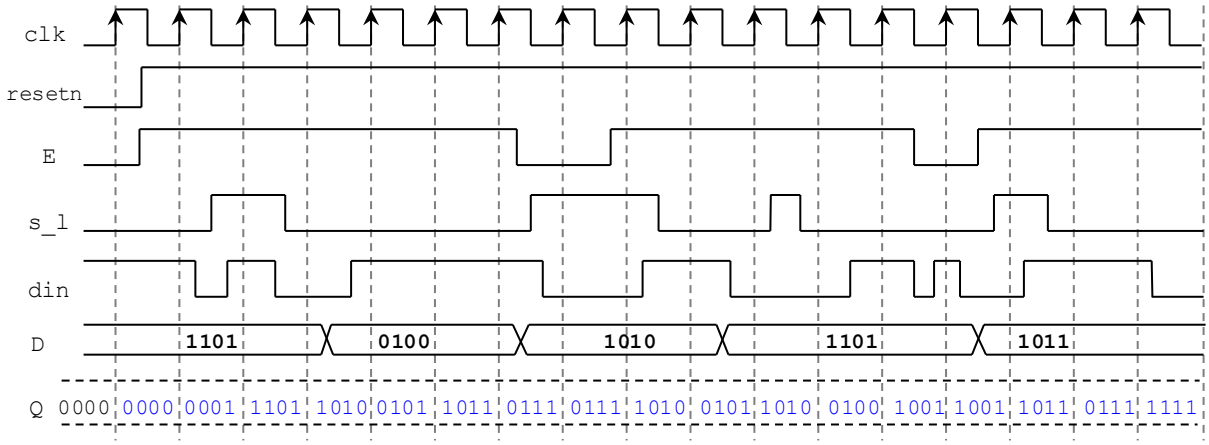


$\lceil 20^{0.5} \rceil = 5$
$\lceil 48^{0.5} \rceil = 7$
$\lceil 26^{0.5} \rceil = 6$
$\lceil 17^{0.5} \rceil = 5$

PROBLEM 5 (12 PTS)

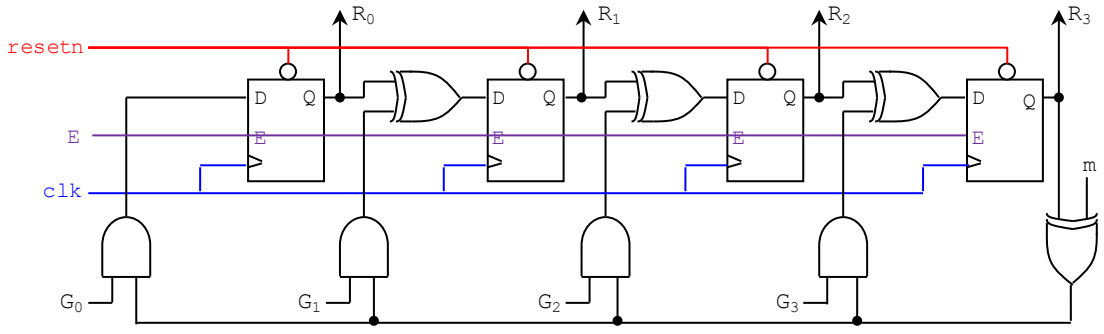
- Complete the timing diagram of the following 4-bit parallel access shift register with enable input. Shifting operation: `s_l=0`. Parallel load: `s_l=1`. Note that $Q = Q_3Q_2Q_1Q_0$. $D = D_3D_2D_1D_0$

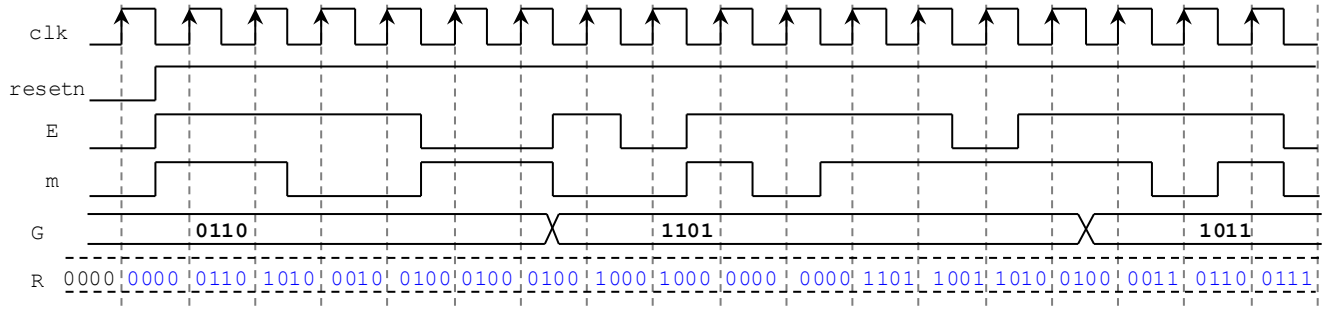




PROBLEM 6 (25 PTS)

- For the following circuit, we have $R = R_3R_2R_1R_0$. $G = G_3G_2G_1G_0$
  - ✓ Write structural VHDL code. Create two files: i) flip flop, ii) top file (where you will interconnect the flip flops and the logic gates). Provide a printout. (10 pts)
  - ✓ Write a VHDL testbench according to the timing diagram shown below. Complete the timing diagram by simulating your circuit (Behavioral Simulation). The clock frequency must be 100 MHz with 50% duty cycle. Provide a printout. (15 pts)





✓ **VHDL Code**: Top File

```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;

entity lfsr_crc2 is
     generic (N: INTEGER:= 4);
     port ( m_in, E: in std_logic;
            resetn, clock: in std_logic;
            G: in std_logic_vector (N-1 downto 0);
            R: out std_logic_vector (N-1 downto 0));
end lfsr_crc2;

architecture structural of lfsr_crc2 is

     component dffe
          port ( d : in  STD_LOGIC;
                 clrn: in std_logic:= '1';
                 prn: in std_logic:= '1';
                 clk : in  STD_LOGIC;
                 ena: in std_logic;
                 q : out  STD_LOGIC);
     end component;

     signal B, D, Q: std_logic_vector (N-1 downto 0);

begin

D(0) <= B(0);

g0: for i in 1 to N-1 generate
                    D(i) <= B(i) xor Q(i-1);
      end generate;

g1: for i in 0 to N-1 generate
        di: dffe port map (d => D(i), clrn => resetn, prn => '1', clk => clock, ena => E, q => Q(i));
        R(i) <= Q(i);
        B(i) <= G(i) and (Q(N-1) xor m_in);
    end generate;

end structural;
```

✓ **VHDL Code**: D-Type flip flop

```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
use IEEE.STD_LOGIC_UNSIGNED.ALL;

entity dffe is
    port ( d : in  STD_LOGIC;
           clrn, prn, clk, ena: in std_logic;
           q : out  STD_LOGIC);
end dffe;

architecture behaviour of dffe is

begin
     process (clk, ena, prn, clrn)
     begin
        if clrn = '0' then q <= '0';
        elsif prn = '0' then q <= '1';
        elsif (clk'event and clk='1') then
           if ena = '1' then q <= d; end if;
        end if;
     end process;
end behaviour;
```

✓ **VHDL Tesbench**:

```
LIBRARY ieee;
USE ieee.std_logic_1164.ALL;

ENTITY tb_lfsr_crc2 IS
    generic (N: integer:= 4);
END tb_lfsr_crc2;

ARCHITECTURE behavior OF tb_lfsr_crc2 IS
    component lfsr_crc2
    port( m_in, E : IN  std_logic;
          resetn : IN  std_logic;
          clock : IN  std_logic;
          G: in std_logic_vector (N-1 downto 0);
          R : OUT  std_logic_vector(N-1 downto 0));
   end component;

   --Inputs
   signal m_in, E : std_logic := '0';
   signal resetn, clock : std_logic := '0';
   signal G: std_logic_vector (N-1 downto 0);

    --Outputs
   signal R : std_logic_vector(N-1 downto 0);

   constant T : time := 10 ns;    -- clock period definition

BEGIN
    -- Instantiate the Unit Under Test (UUT)
   uut: lfsr_crc2 PORT MAP (m_in => m_in, E => E, resetn => resetn, clock => clock, G => G, R => R);

   -- Clock process definitions
   clock_process: process
   begin
      clock <= '0'; wait for T/2;
      clock <= '1'; wait for T/2;
   end process;

   -- Stimulus process
   stim_proc: process
   begin
      -- hold reset state for 100 ns.
        resetn <= '0'; m_in <= '0'; G <= "1001";
        wait for 100 ns;
        resetn <= '1';

      -- insert stimulus here
        G <= "0110"; E <= '1'; m_in <= '1'; wait for 2*T;
        G <= "0110"; E <= '1'; m_in <= '0'; wait for 2*T;
        G <= "0110"; E <= '0'; m_in <= '1'; wait for 2*T;
        G <= "1101"; E <= '1'; m_in <= '0'; wait for T;
        G <= "1101"; E <= '0'; m_in <= '0'; wait for T;
        G <= "1101"; E <= '1'; m_in <= '1'; wait for T;
        G <= "1101"; E <= '1'; m_in <= '0'; wait for T;
        G <= "1101"; E <= '1'; m_in <= '1'; wait for 2*T;
        G <= "1101"; E <= '0'; m_in <= '1'; wait for T;
        G <= "1101"; E <= '1'; m_in <= '1'; wait for T;
        G <= "1011"; E <= '1'; m_in <= '1'; wait for T;
        G <= "1011"; E <= '1'; m_in <= '0'; wait for T;
        G <= "1011"; E <= '1'; m_in <= '1'; wait for T;
        G <= "1011"; E <= '0'; m_in <= '0'; wait for T;

        wait;
   end process;

END;
```