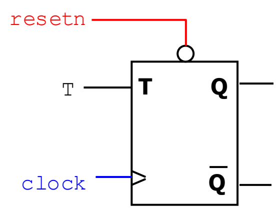
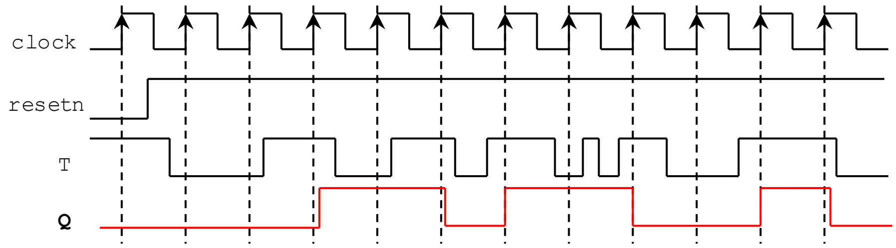
Solutions - Homework 3

(Due date: October 31st (001), October 30th (006) @ 5:30 pm)
Presentation and clarity are very important! Show your procedure!

PROBLEM 1 (12 PTS)

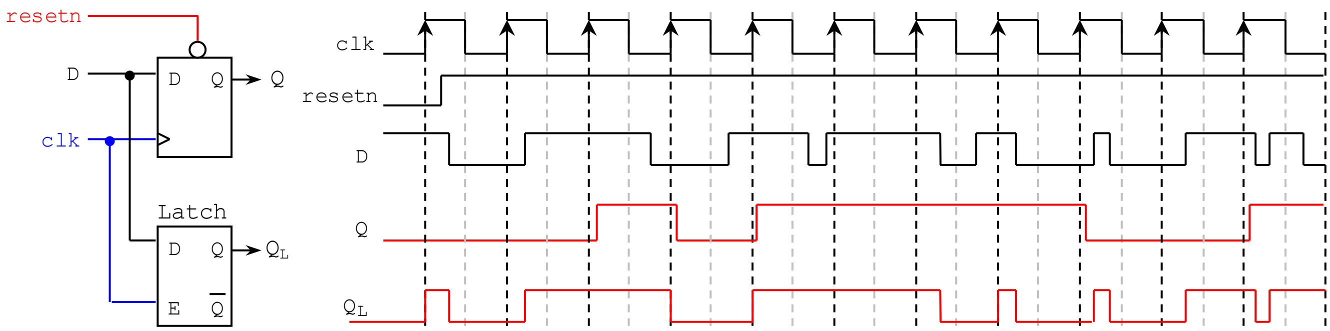
- Complete the timing diagram of the circuit shown below. (5 pts)





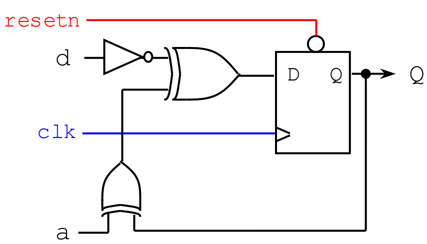
$Q_{t+1} \leftarrow T \oplus Q_t$

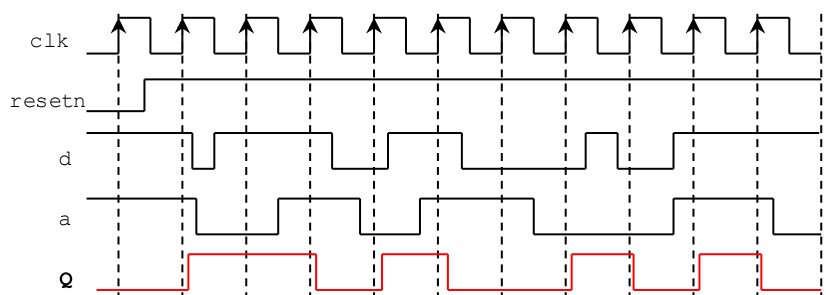
- Complete the timing diagram of the circuits shown below: (7 pts)



PROBLEM 2 (17 PTS)

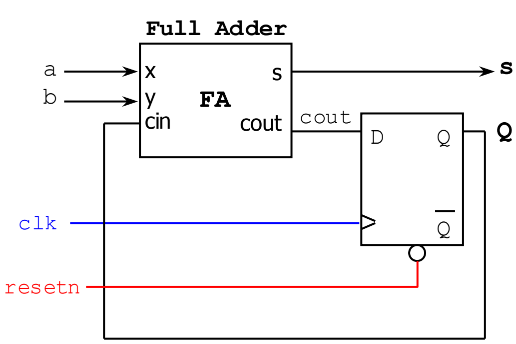
- Complete the timing diagram of the circuit shown below: (7 pts)

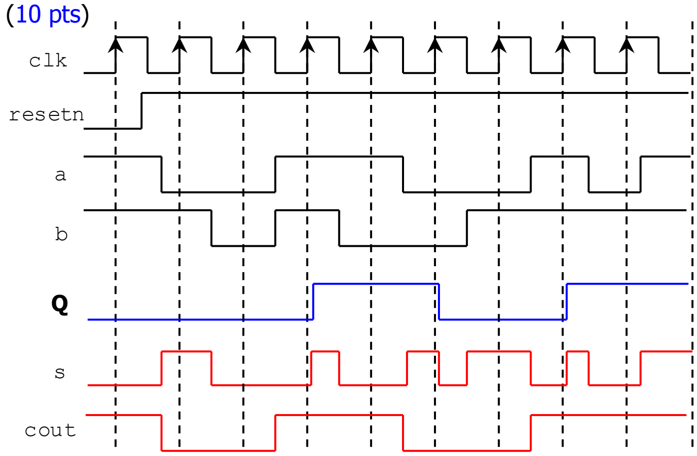




$Q_{t+1} \leftarrow \overline{a \oplus d \oplus Q_t}$

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

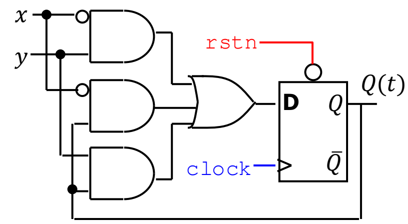




$Q_{t+1} \leftarrow ab + aQ_t + bQ_t$

PROBLEM 3 (16 PTS)

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



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

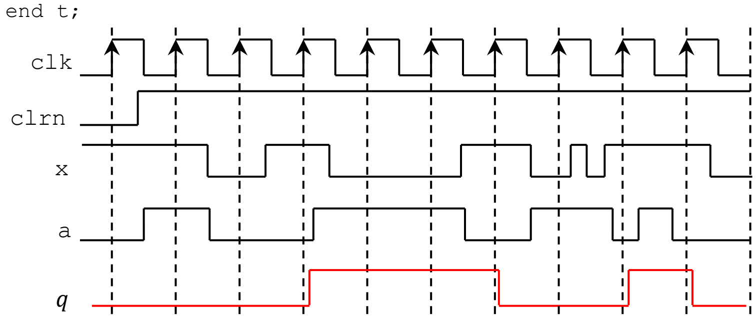
```
library ieee;
use ieee.std_logic_1164.all;

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 = '1' then
            qt <= a xor not(qt);
        end if;
     end if;
   end process;
   q <= qt;
end t;
```



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

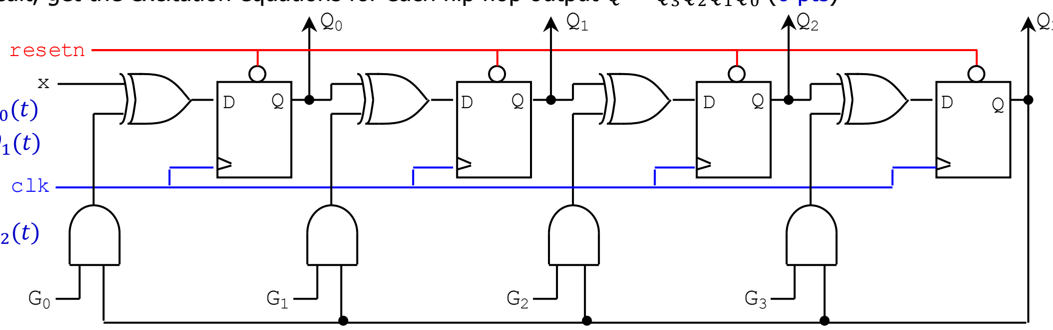
- Given the following circuit, get the excitation equations for each flip flop output $Q = Q_3Q_2Q_1Q_0$ (6 pts)

$Q_{0(t+1)} \leftarrow (G_0Q_3(t)) \oplus x$

$Q_{1(t+1)} \leftarrow (G_1Q_3(t)) \oplus Q_0(t)$

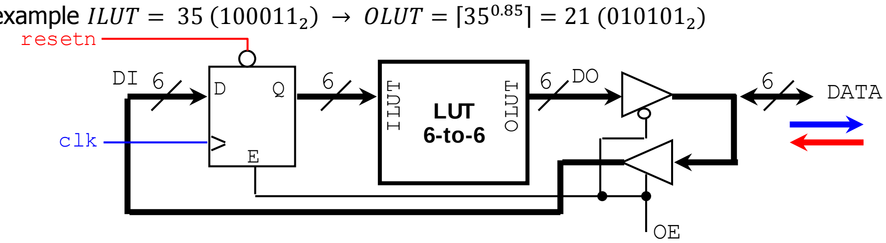
$Q_{2(t+1)} \leftarrow (G_2Q_3(t)) \oplus Q_1(t)$

$Q_{3(t+1)} \leftarrow (G_3Q_3(t)) \oplus Q_2(t)$



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: $OLUT = \lceil ILUT^{0.85} \rceil$, where $ILUT$ is an unsigned number.
  For example $ILUT = 35\ (100011_2) \rightarrow OLUT = \lceil 35^{0.85} \rceil = 21\ (010101_2)$

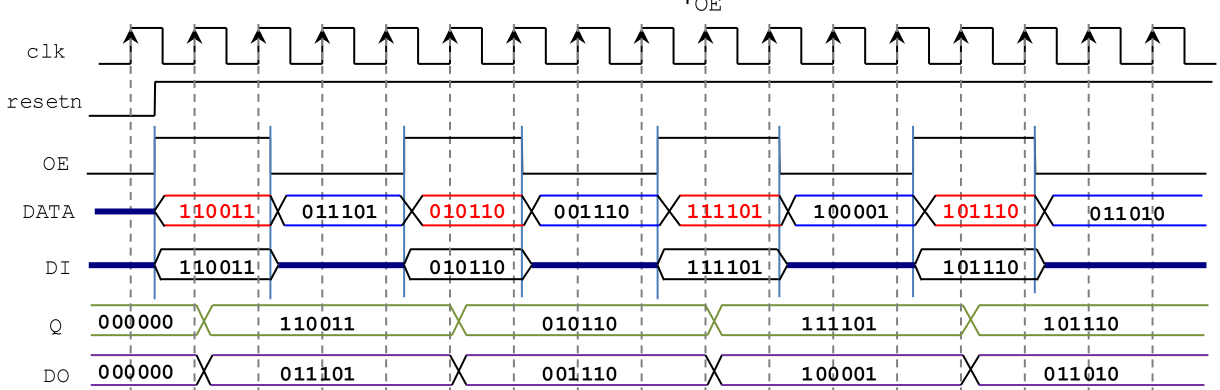


$\lceil 51^{0.85} \rceil = 29$

$\lceil 22^{0.85} \rceil = 14$

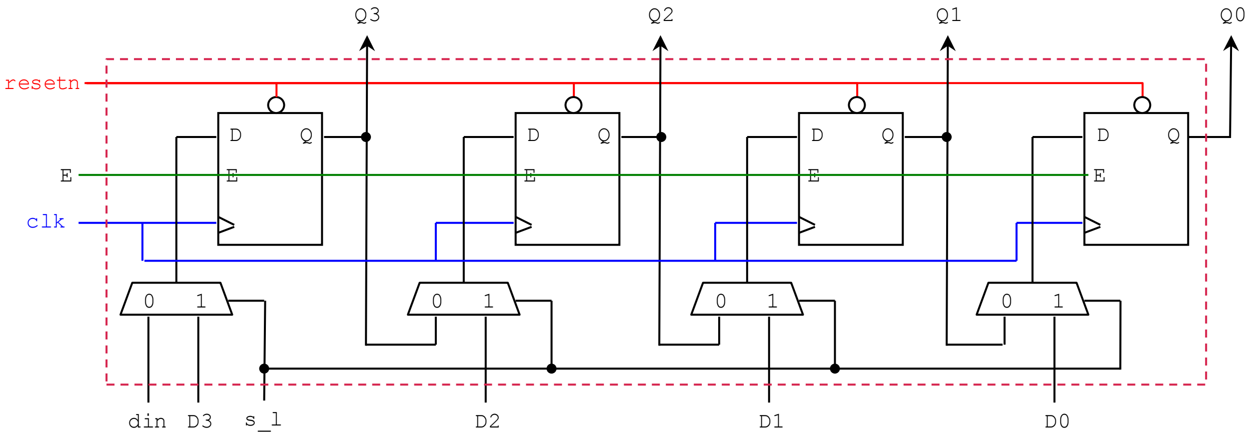
$\lceil 61^{0.85} \rceil = 33$

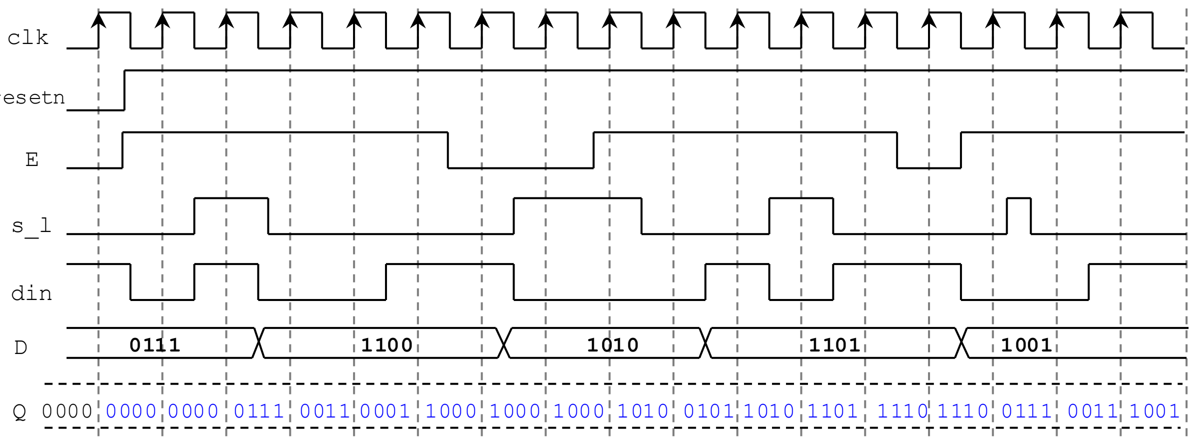
$\lceil 46^{0.85} \rceil = 26$



PROBLEM 5 (12 PTS)

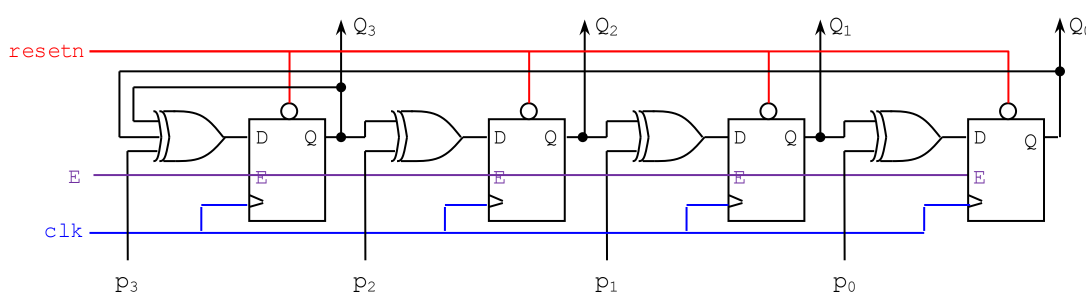
- Complete the timing diagram of the following 4-bit parallel access shift register with enable input. Shifting operation: `s_l=0`. Parallel load: `s_l=1`. Note that $Q = Q_3Q_2Q_1Q_0$. $D = D_3D_2D_1D_0$

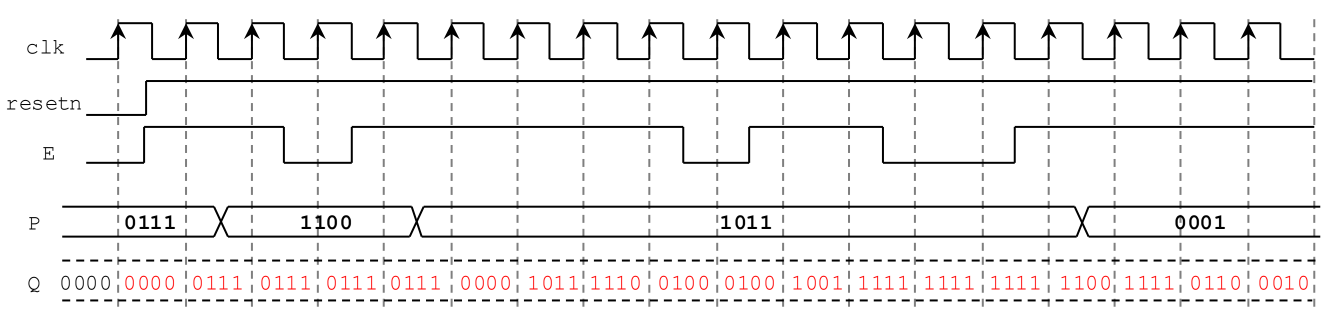




PROBLEM 6 (25 PTS)

- The following circuit is a multiple-input compressor circuit (MIC), a component in Built-in Self-Test systems. $Q = Q_3Q_2Q_1Q_0$. $P = P_3P_2P_1P_0$
  - ✓ 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_mic is
   generic (N: INTEGER:= 4);
   port ( resetn, clock: in std_logic;
          E: in std_logic;
          P: in std_logic_vector (N-1 downto 0);
          Q: out std_logic_vector (N-1 downto 0));
end lfsr_mic;

architecture structural of lfsr_mic 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 D, Qt: std_logic_vector (N-1 downto 0);

begin

  D(N-1) <= Qt(N-1) xor Qt(0) xor P(N-1);

  g0: for i in N-2 downto 0 generate
         D(i) <= Qt(i+1) xor P(i);
      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 => Qt(i));
      end generate;

  Q <= Qt;

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_mic IS
    generic (N: integer:= 4);
END tb_lfsr_mic;

ARCHITECTURE behavior OF tb_lfsr_mic IS
   component lfsr_mic
        port ( resetn, clock: in std_logic;
               E: in std_logic;
               P: in std_logic_vector (N-1 downto 0);
               Q: out std_logic_vector (N-1 downto 0));
   end component;

-- Inputs
  signal E : std_logic := '0';
  signal resetn : std_logic := '0';
  signal clock  : std_logic := '0';
  signal P: std_logic_vector (N-1 downto 0):= (others => '0');

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

-- Clock period definitions
  constant T : time := 10 ns;

BEGIN
   -- Instantiate the Unit Under Test (UUT)
   uut: lfsr_mic PORT MAP (resetn => resetn, clock => clock, E => E, P => P, Q => Q);

   -- 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
      resetn <= ‘0’; P <= "0111"; wait for 100 ns;
      resetn <= '1';
      P <= "0111"; E <= '1'; wait for T;
      P <= "1100"; E <= '1'; wait for T;
      P <= "1100"; E <= '0'; wait for T;
      P <= "1100"; E <= '1'; wait for T;
      P <= "1011"; E <= '1'; wait for 4*T;
      P <= "1011"; E <= '0'; wait for T;
      P <= "1011"; E <= '1'; wait for 2*T;
      P <= "1011"; E <= '0'; wait for 2*T;
      P <= "1011"; E <= '1'; wait for T;
      P <= "0001"; E <= '1'; wait for 4*T;
      E <= '0';
      wait;
   end process;

END;
```