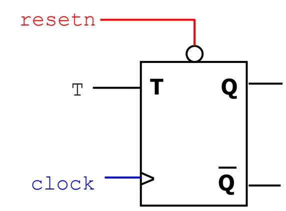
Solutions - Homework 3

(Due date: March 18th @ 11:59 pm)

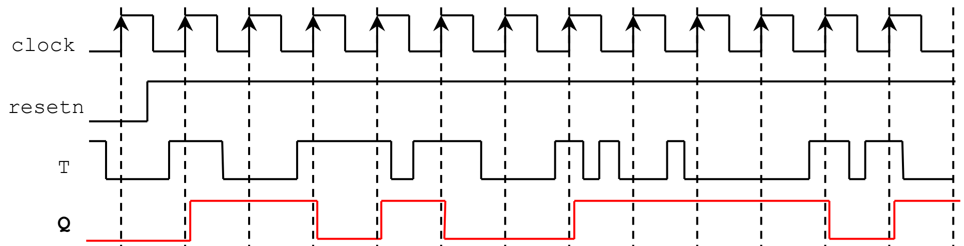
Presentation and clarity are very important! Show your procedure!

PROBLEM 1 (11 PTS)

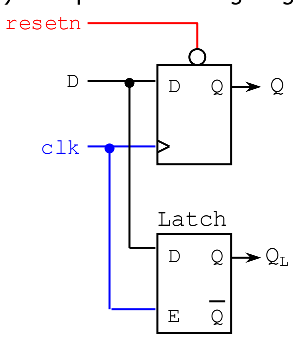
a) Complete the timing diagram of the circuit shown below. (5 pts)

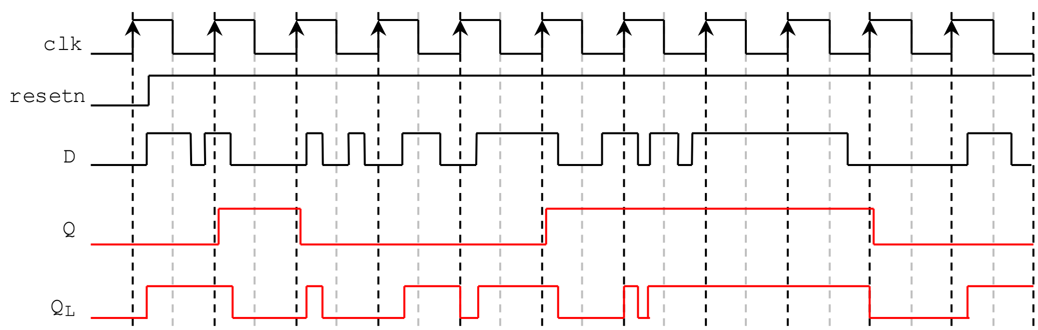


$Q_{t+1} \leftarrow T \oplus Q_t$



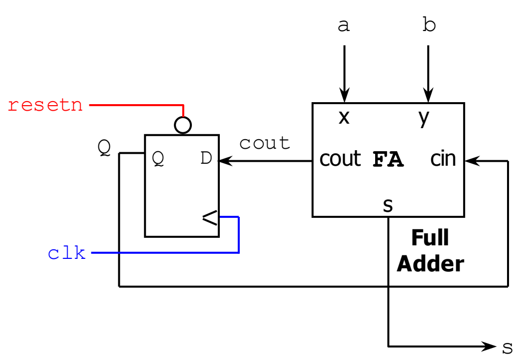
b) Complete the timing diagram of the circuits shown below: (6 pts)

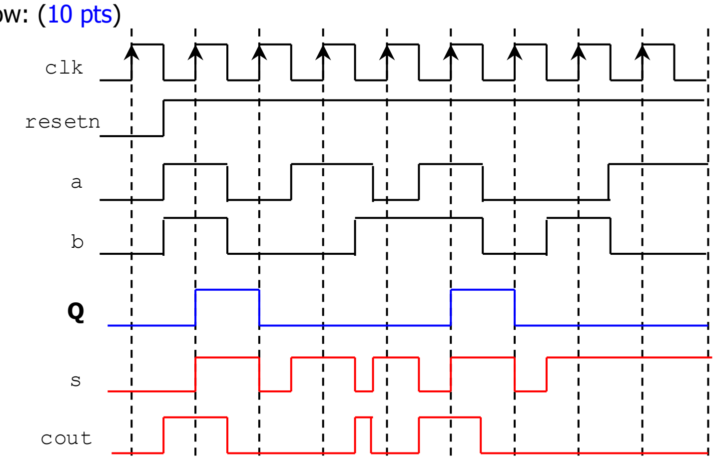




PROBLEM 2 (17 PTS)

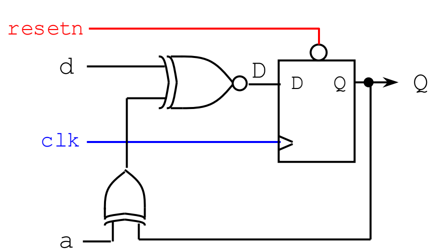
- Complete the timing diagram of the circuit shown below: (10 pts)

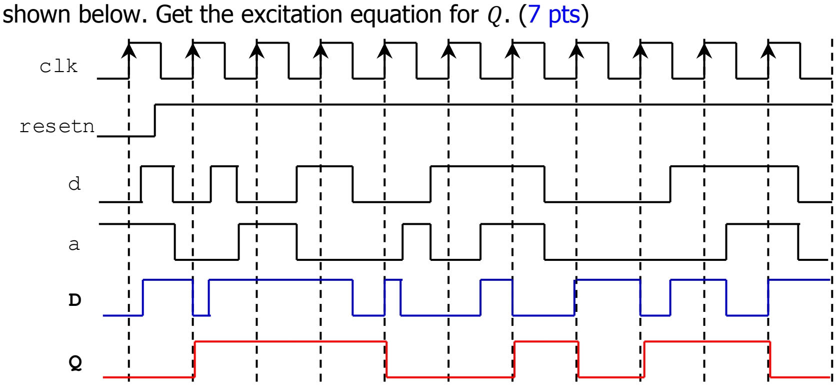




$Q_{t+1} \leftarrow ab + aQ_t + bQ_t$

- Complete the timing diagram of the circuit shown below. Get the excitation equation for Q. (7 pts)





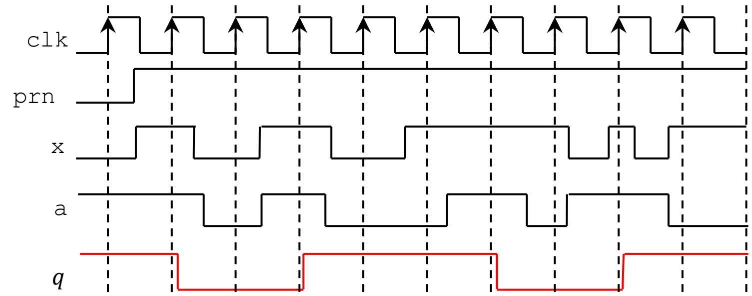
$Q_{t+1} \leftarrow \overline{a \oplus d \oplus Q_t}$

PROBLEM 3 (10 PTS)

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

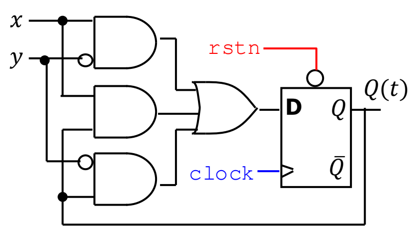
```
library ieee;
use ieee.std_logic_1164.all;

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

architecture a of circ is

   signal qt: std_logic;

begin
   process (prn, clk, x, a)
   begin
     if prn = '0' then
         qt <= '1';
     elsif (clk'event and clk = '1') then
        if x = '1' then
           qt <= a xor qt;
        end if;
     end if;
   end process;
   q <= qt;
end a;
```

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



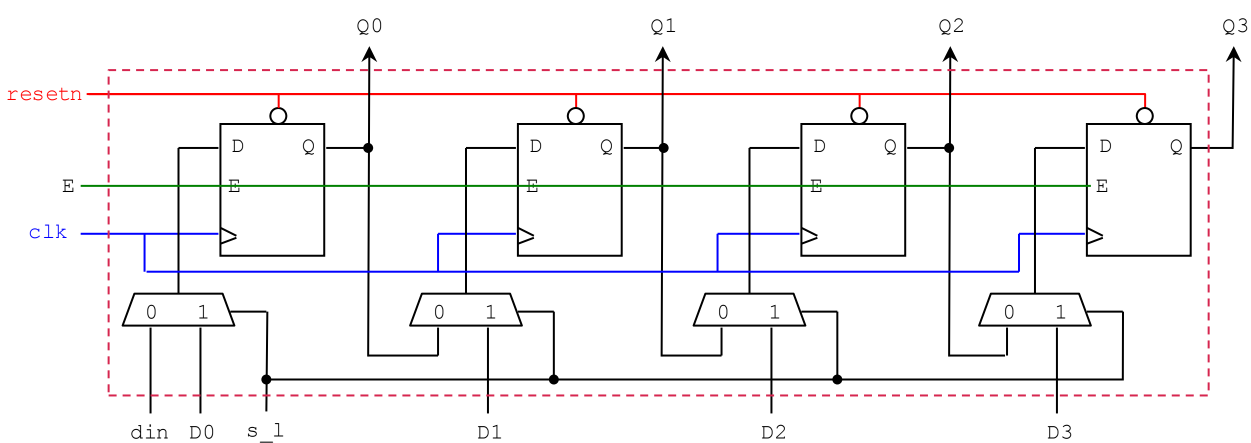
b) With a flip flop and logic gates, sketch the circuit whose excitation equations is given by (4 pts):

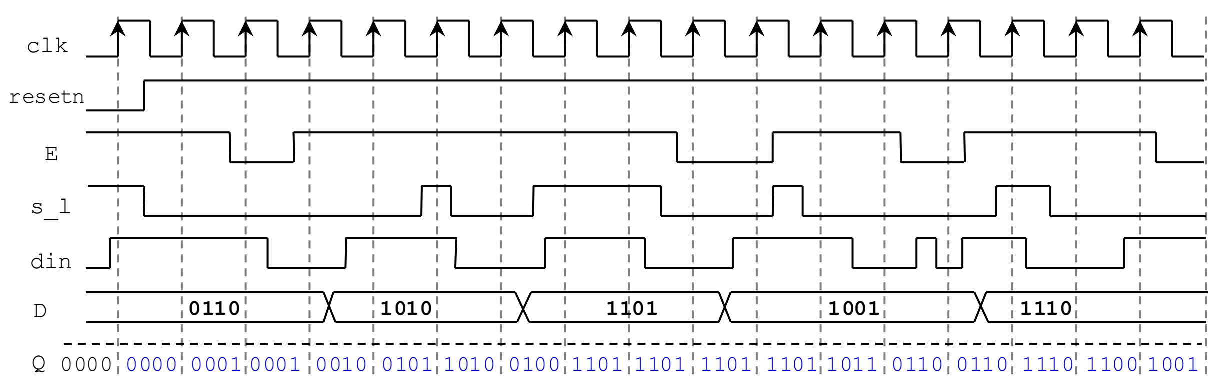
$$Q(t+1) \leftarrow x\bar{y} + xQ(t) + \bar{y}Q(t)$$



PROBLEM 4 (10 PTS)

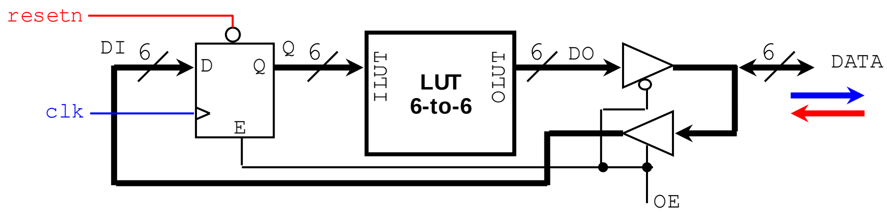
- Complete the timing diagram of the following 4-bit parallel access shift register with enable input.
  When `E=1`: If `s_l=0` (shifting operation). If `s_l=1` (parallel load) Note that $Q = Q_3Q_2Q_1Q_0$. $D = D_3D_2D_1D_0$





PROBLEM 5 (12 PTS)

- Given the following circuit, complete the timing diagram (signals DO, Q, and $DATA$).
  The LUT 6-to-6 implements the following function: $OLUT = \lceil ILUT^{0.95} \rceil$, where $ILUT$ is an unsigned number.
  For example: $ILUT = 54\ (110110_2) \rightarrow OLUT = \lceil 54^{0.95} \rceil = 45\ (101101_2)$

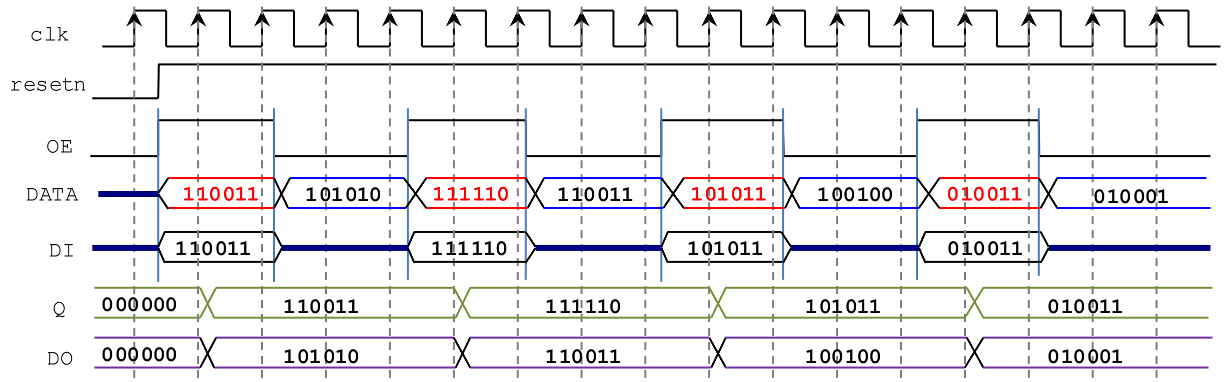


$\lceil 51^{0.95} \rceil = 42$

$\lceil 62^{0.95} \rceil = 51$

$\lceil 43^{0.95} \rceil = 36$

$\lceil 19^{0.95} \rceil = 17$



PROBLEM 6 (22 PTS)

a) For the following circuit, complete the timing diagram and get the excitation equations of the flip flop outputs. $Q = Q_3Q_2Q_1Q_0$.

$Q_3(t+1) \leftarrow \bar{E}Q_3(t) + E(Q_3(t) \oplus Q_0(t))$

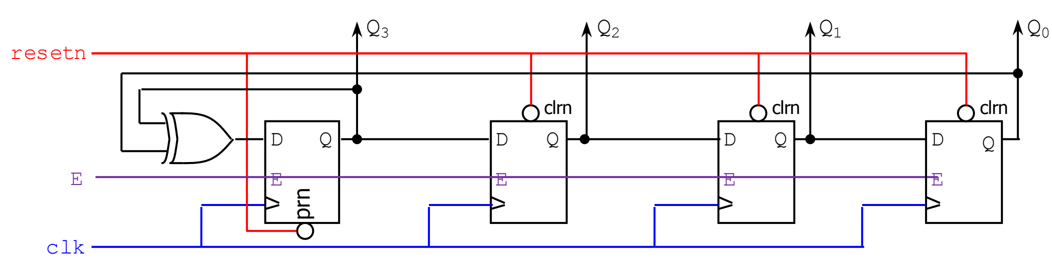
$Q_2(t+1) \leftarrow \bar{E}Q_2(t) + EQ_3(t)$

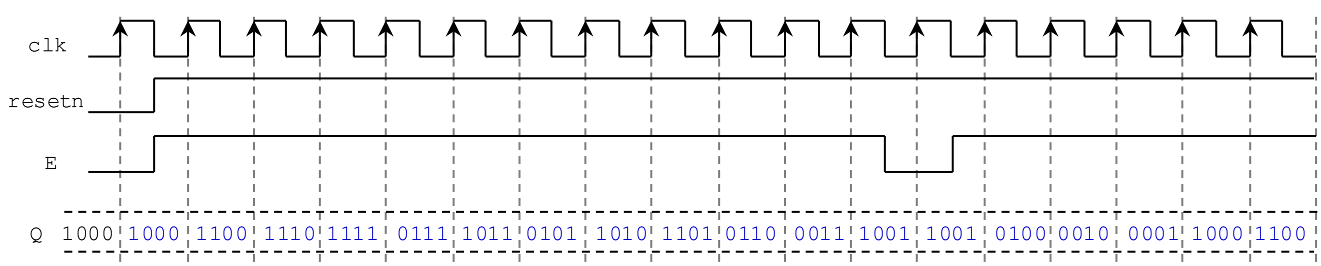
$Q_1(t+1) \leftarrow \bar{E}Q_1(t) + EQ_2(t)$

$Q_0(t+1) \leftarrow \bar{E}Q_0(t) + EQ_1(t)$

b) Write the VHDL for the given circuit and simulate your circuit.
- ✓ Write structural VHDL code. Create two files: i) flip flop, ii) top file (where you will interconnect the flip flops and the logic gates). (10 pts)
  - ▫ resetn input: It is connected to the presetn input of flip flop Q_3, and to the resetn input of flip flops Q_2, Q_1, Q_0.
- ✓ Write a VHDL testbench according to the timing diagram shown below. Run the simulation (Behavioral Simulation) and verify the results by comparing them with the simulation you completed manually. The clock frequency must be 100 MHz with 50% duty cycle. (8 pts)

c) Upload (as a .zip file) the following files to Moodle (an assignment will be created):
- ✓ VHDL code files and testbench.
- ✓ A screenshot of your Vivado simulation results (it should show the values for Q).





✓ **VHDL Code**: Top File

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

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

architecture structural of lfsr_prbsg 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);

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

 df: dffe port map (d => D(N-1), clrn => '1', prn => resetn, clk => clock, ena => E, q => Qt(N-1));

 g1: for i in N-2 downto 0 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;
```

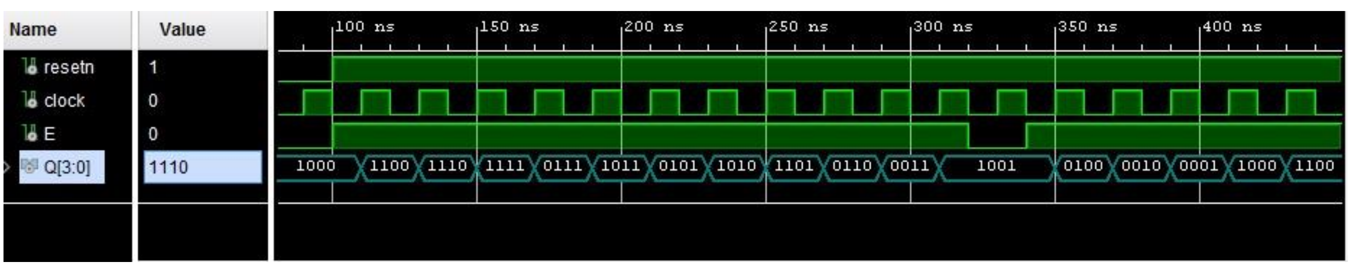
✓ **VHDL Code**: D-Type flip flop

```
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**:

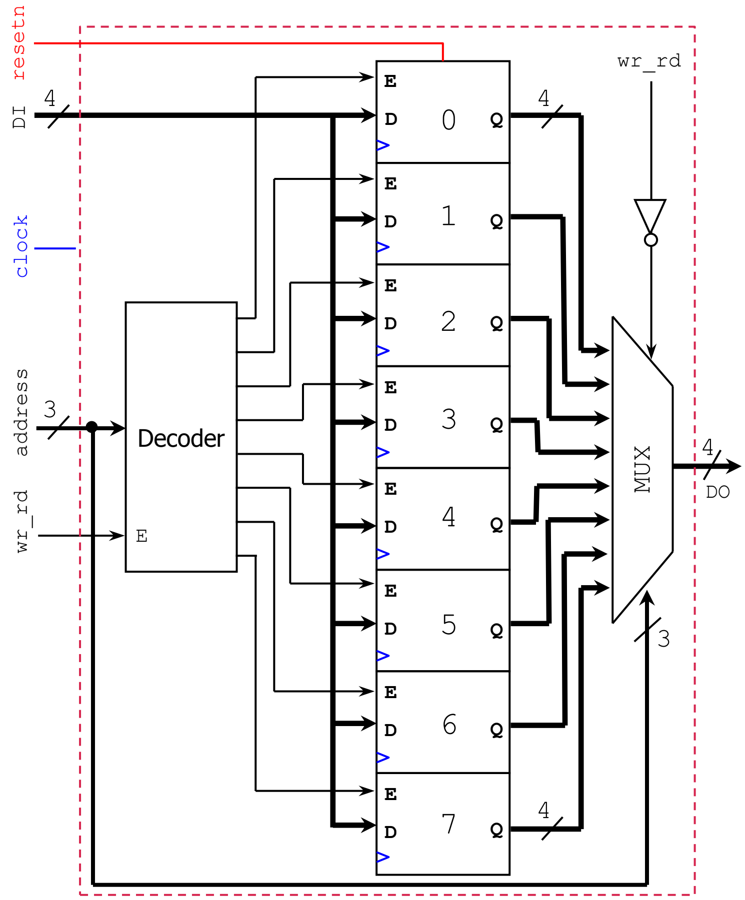
```
LIBRARY ieee;
USE ieee.std_logic_1164.ALL;

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

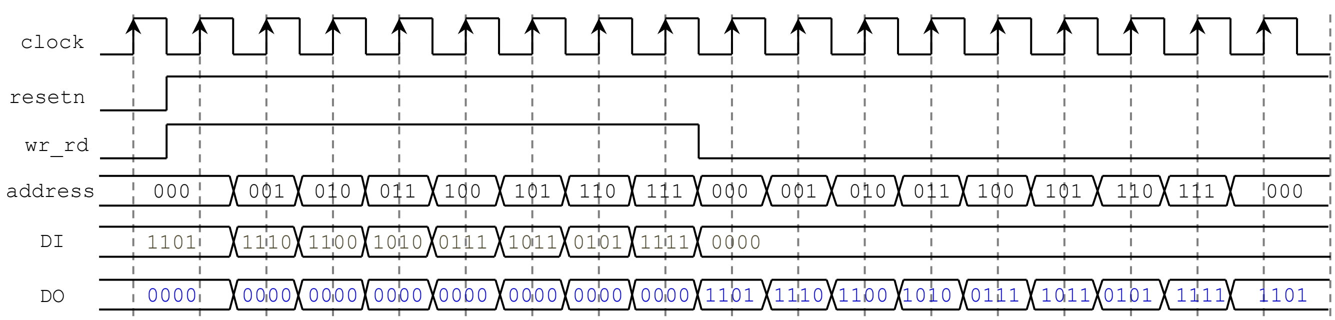
```
ARCHITECTURE behavior OF tb_lfsr_prbsg IS
    component lfsr_prbsg
        port (  resetn, clock: in std_logic;
                E: in std_logic;
                Q: out std_logic_vector (N-1 downto 0));
    end component;

   -- Inputs
   signal E : std_logic := '0';
   signal resetn : std_logic := '0';
   signal clock : std_logic := '0';

   -- 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_prbsg PORT MAP (resetn => resetn, clock => clock, E => E, 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
      E <= '0'; wait for 100 ns;       -- hold reset state for 100 ns.
      resetn <= '1';
      E <= '1'; wait for 11*T;
      E <= '0'; wait for T;
      E <= '1'; wait for 6*T;
      E <= '0';
      wait;
   end process;

END;
```



PROBLEM 7 (8 PTS)

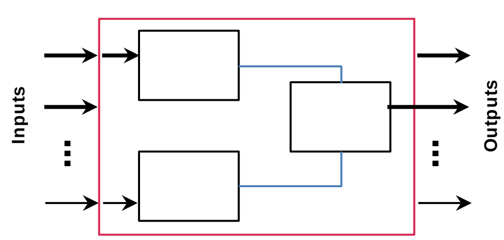
- Complete the timing diagram (output DO) of the following Random Memory Access (RAM) Emulator.
- RAM Emulator: It has 8 addresses, where each address holds a 4-bit data. The memory positions are implemented by 4-bit registers. The *resetn* and *clock* signals are shared by all the registers. Data is written or read onto/from one of the registers (selected by the signal address).
- Operations:
  - ✓ Writing onto memory (wr_rd='1'): The 4-bit input data (DI) is written into one of the 8 registers. The address signal selects which register is to be written.
    - For example: if address = "101", then the value of DI is written into register 5.
    - Note that because the BusMUX 8-to-1 includes an enable input, if wr_rd=1, then the BusMUX outputs are 0's.
  - ✓ Reading from memory (wr_rd='0'): The address signal selects the register from which data is read. This data appears on the BusMUX output.
    - For example: If address = "010", then data from register 2 appears on BusMUX output.





PROBLEM 8 (10 PTS)

- Attach your Project Status Report (no more than 1 page, single-spaced, 2 columns, only one submission per group). This report should contain the initial status of your project. For formatting, use the provided template (`Final Project - Report Template.docx`). The sections included in the template are the ones required in your Final Report. At this stage, you are only required to:
  - ✓ Include a (draft) project description and title.
  - ✓ Include a <u>draft</u> Block Diagram of your hardware architecture.



Block Diagram

- As a guideline, the figure shows a simple Block Diagram. There are input and output signals, as well as internal components along with their interconnection.
  - ✓ At this stage, only a rough draft is required. There is no need to go into details: it is enough to show the tentative top-level components that would constitute your system as well as the tentative inputs and outputs.
- Only student is needed to attach the report (make sure to indicate all the team members).