RELATÓRIO- PROJETO 6

Questão 1

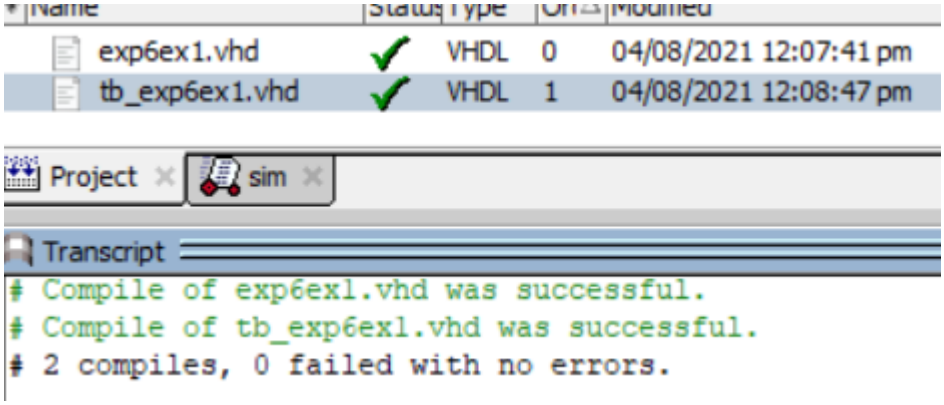
Arquivo exp6ex1.vhd:

```
2   --  Circuito: flip-fLop JK
3   --  gatilhado pela borda de subida
4   --          PR, CLR - Entradas assíncronas
5   --          CLK - Clock
6   --           JK - Entrada síncrona
7   --            Q - Saída
8   -- ******************************************
9   ------------------------------------------------
10  library IEEE;
11  use IEEE.std_logic_1164.all;
12  ------------------------------------------------
13  entity EXP6EX1 is port (
14      PR, CLR, CLK :in std_logic;
15      JK :in std_logic_vector(1 DOWNTO 0);
16      Q :out std_logic);
17  end EXP6EX1;
18  ------------------------------------------------
19
20  architecture flipflopJK of EXP6EX1 is
21  begin
22      process(PR, CLR, CLK)
23          variable Qbuf: std_logic;
24      begin
25          if (PR = '1') then
26              Qbuf := '1';
27          elsif (PR = '0' and CLR = '1') then
28              Qbuf := '0';
29          elsif rising_edge(CLK) then
30              case JK is
31                  when "00" => Qbuf := Qbuf;
32                  when "01" => Qbuf := '0';
33                  when "10" => Qbuf := '1';
34                  when "11" => Qbuf := not(Qbuf);
35                  when others => Qbuf := Qbuf;
36              end case;
37          else
38              Qbuf := Qbuf;
39          end if;
40          Q <= Qbuf;
41      end process;
42  end flipflopJK;
```

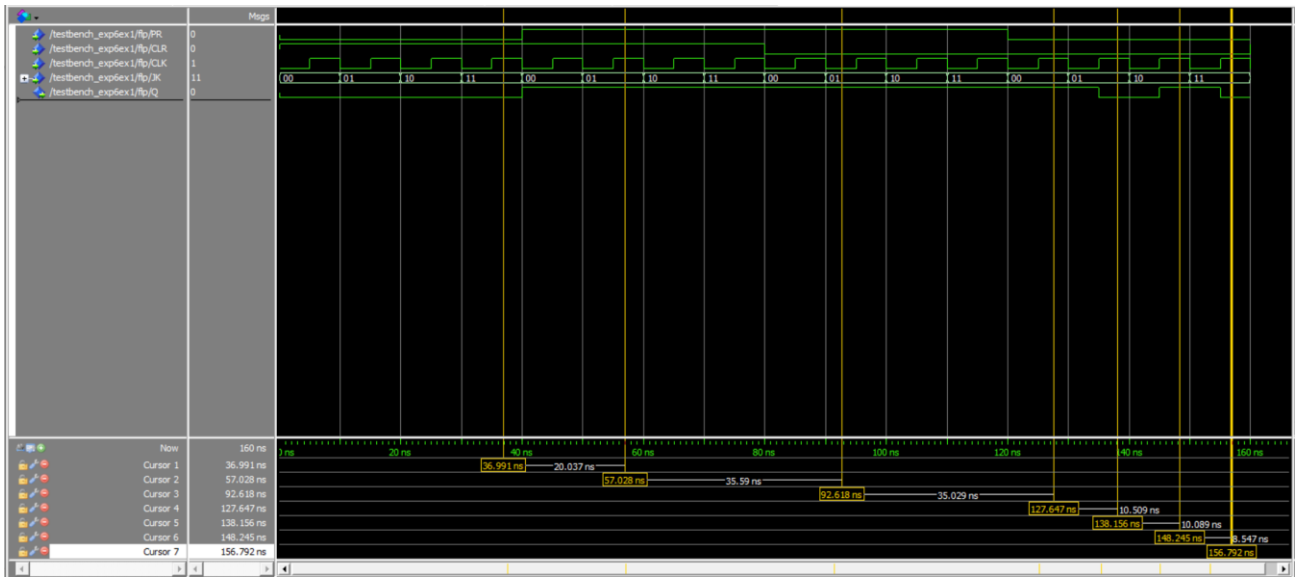
Arquivo tb_exp6ex1.vhd:

```vhdl
-- ************************************************
--  Testbench para simulacao Funcional do
--  Circuito: flip-fLop JK
--  gatilhado pela borda de subida
--          PR, CLR - Entradas assíncronas
--          CLK - Clock
--           JK - Entrada síncrona
--            Q - Saída
-- ************************************************
-------------------------------------------------------
library IEEE;
use IEEE.std_logic_1164.all;
use std.textio.all;
use ieee.numeric_std.all;
-------------------------------------------------------

entity testbench_exp6ex1 IS END;

architecture tb_exp6ex1 of testbench_exp6ex1 is

    component EXP6EX1

    port (
        PR, CLR, CLK :in std_logic;
        JK :in std_logic_vector(1 DOWNTO 0);
        Q :out std_logic);
    end component;
-------------------------------------------------------
    signal pr, clr: std_logic;
    signal clk : std_logic := '0';
    signal jk : std_logic_vector(1 DOWNTO 0);

    Begin

    flp: EXP6EX1 PORT MAP (PR => pr, CLR => clr, CLK => clk, JK => jk, Q => open);
    clk <= not clk after 5 ns;

    estimulo: PROCESS
    begin
        pr <= '0'; clr <= '1';
        for i in 0 to 3 loop
            jk <= std_logic_vector(to_unsigned(i, 2));
            wait for 10 ns;
        end loop;

        pr <= '1';
        for i in 0 to 3 loop
            jk <= std_logic_vector(to_unsigned(i, 2));
            wait for 10 ns;
        end loop;

        clr <= '0';
        for i in 0 to 3 loop
            jk <= std_logic_vector(to_unsigned(i, 2));
            wait for 10 ns;
        end loop;

        pr <= '0';
        for i in 0 to 3 loop
            jk <= std_logic_vector(to_unsigned(i, 2));
            wait for 10 ns;
        end loop;

    end PROCESS estimulo;

end tb_exp6ex1;
```

Compilação:



Simulação:



	PR	CLR	CLK	J	K	Q
Cursor 1	0	1	SUBIDA	1	1	0
Cursor 2	1	1	SUBIDA	0	1	1
Cursor 3	1	0	DESCIDA	0	1	1
Cursor 4	0	0	SUBIDA	0	0	1
Cursor 5	0	0	SUBIDA	0	1	0
Cursor 6	0	0	SUBIDA	1	0	1
Cursor 7	0	0	SUBIDA	1	1	0

Questão 2

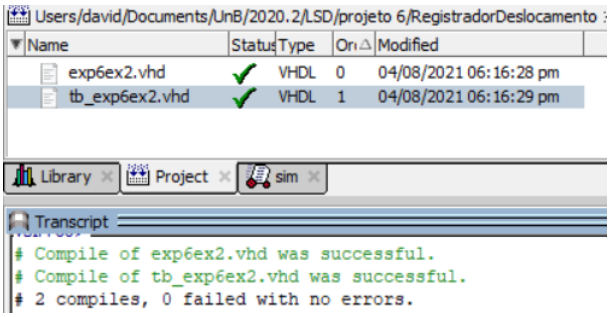
Arquivo exp6ex1.vhd:

```
1  -- ***********************************************
2  --  Circuito: Registrador de deslocamento
3  --  bidirecional de 4 bits
4  --          RST, LOAD - Entradas assíncronas
5  --          CLK - Clock
6  --          D, DIR, L e R - Entradas síncronas
7  --          Q - Saída
8  -- ***********************************************
9  -------------------------------------------------------
10 library IEEE;
11 use IEEE.STD_LOGIC_1164.ALL;
12 -------------------------------------------------------
13 entity EXP6EX2 is port (
14     CLK, RST, LOAD, DIR, L, R : in std_logic;
15     D : in std_logic_vector(3 DOWNTO 0);
16     Q : out std_logic_vector(3 DOWNTO 0));
17 end EXP6EX2;
18 -------------------------------------------------------
19 architecture registradorDeslocamento of EXP6EX2 is
20 
21     signal currentState, nextState : std_logic_vector(3 DOWNTO 0);
22 
23 begin
24 
25     sync_proc: process(CLK)
26     begin
27         if rising_edge(CLK) then
28             currentState <= nextState;
29         end if;
30     end process sync_proc;
31 
32 
33     comb_proc: process(currentState, RST, LOAD, DIR, D, L, R)
34     begin
35         if (RST = '1') then
36             nextState <= "0000";
37         elsif (LOAD = '1') then
38             nextState <= D;
39         elsif (DIR = '0') then
40             nextState <= currentState(2) & currentState(1) & currentState(0) & L;
41         elsif (DIR = '1') then
42             nextState <= R & currentState(3) & currentState(2) & currentState(1);
43         else nextState <= currentstate;
44         end if;
45     end process;
46     Q <= currentState;
47 end registradorDeslocamento;
```

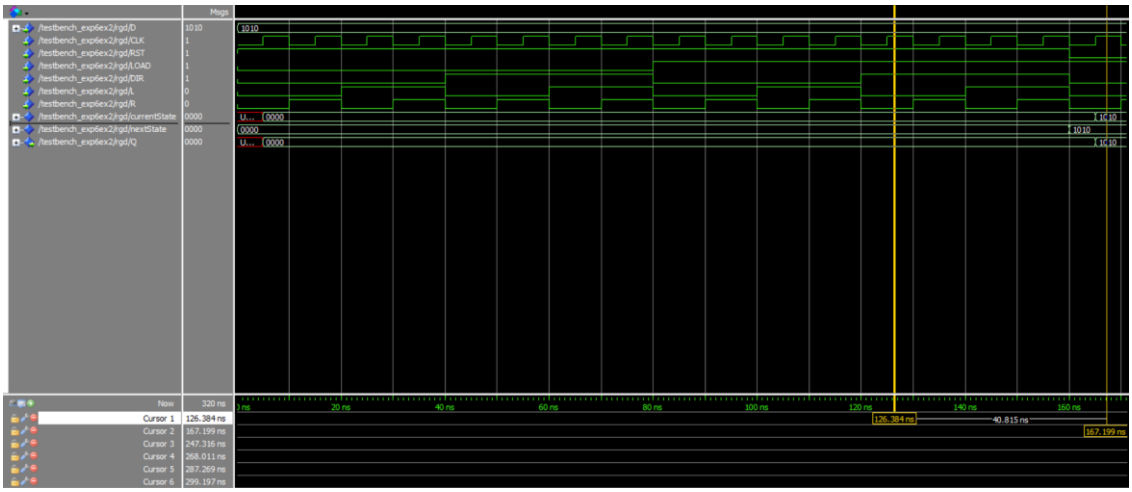
Arquivo tb_exp6ex1.vhd:

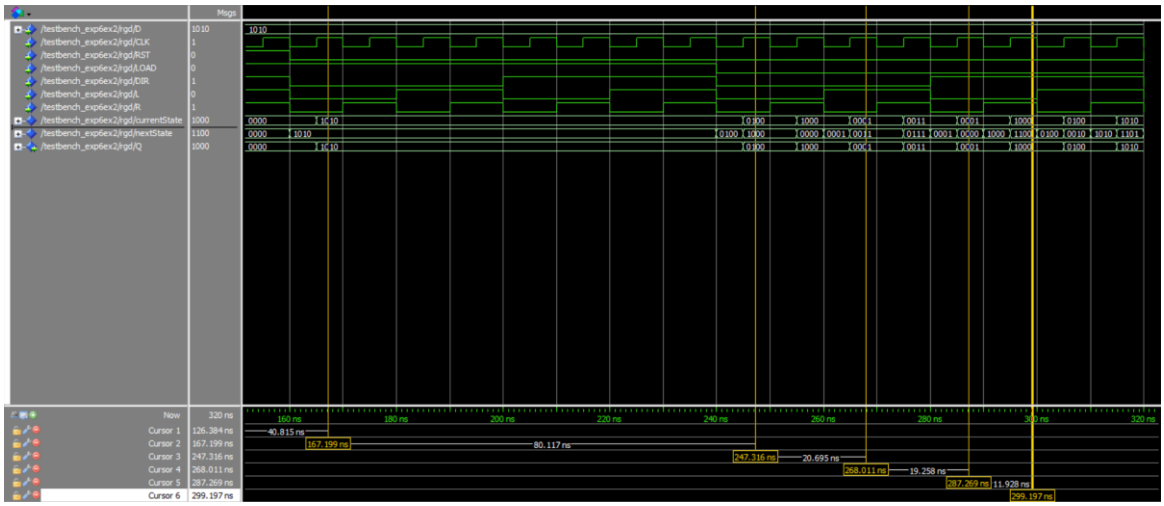
```vhdl
-- ***********************************************
--  Testbench para simulacao Funcional do
--  Circuito: Registrador de deslocamento
--  bidirecional de 4 bits
--          RST, LOAD - Entradas assíncronas
--          CLK - Clock
--          D, DIR, L e R - Entradas síncronas
--          Q - Saída
-- ***********************************************
-------------------------------------------------------
library IEEE;
use IEEE.std_logic_1164.all;
use std.textio.all;
use ieee.numeric_std.all;
-------------------------------------------------------

entity testbench_exp6ex2 IS END;

architecture tb_exp6ex2 of testbench_exp6ex2 is

    component EXP6EX2

    port (
        CLK, RST, LOAD, DIR, L, R : in std_logic;
        D : in std_logic_vector(3 DOWNTO 0);
        Q : out std_logic_vector(3 DOWNTO 0));
    end component;
-------------------------------------------------------
    signal load, rst : std_logic;
    signal clk : std_logic := '0';
    signal d : std_logic_vector(3 DOWNTO 0);
    signal DIR_L_R : std_logic_vector(2 DOWNTO 0);

    Begin

    rgd: EXP6EX2 PORT MAP (LOAD => load, RST => rst, DIR => DIR_L_R(2), L => DIR_L_R(1), R => DIR_L_R(0), CLK => clk, D => d, Q => open);
    clk <= not clk after 5 ns;

    estimulo: PROCESS
    begin
        -- Código com D fixado em um valor qualquer


        d <= "1010";

        rst <= '1'; load <= '0';
        for j in 0 to 7 loop
            DIR_L_R <= std_logic_vector(to_unsigned(j, 3));
                wait for 10 ns;
        end loop;

        load <= '1';
        for j in 0 to 7 loop
            DIR_L_R <= std_logic_vector(to_unsigned(j, 3));
                wait for 10 ns;
        end loop;

        rst <= '0';
        for j in 0 to 7 loop
            DIR_L_R <= std_logic_vector(to_unsigned(j, 3));
                wait for 10 ns;
        end loop;

        load <= '0';
        for j in 0 to 7 loop
            DIR_L_R <= std_logic_vector(to_unsigned(j, 3));
                wait for 10 ns;
        end loop;
    end PROCESS estimulo;
end tb_exp6ex2;
```

Compilação:



Simulação (Feita com D fixo em 1010):





	CLK	RST	LOAD	DIR	L	R	Q	nextState
Cursor 1	SUBIDA	1	1	1	0	0	0000	0000
Cursor 2	SUBIDA	0	1	0	0	0	1010	1010
Cursor 3	SUBIDA	0	0	0	0	0	0100	1000
Cursor 4	SUBIDA	0	0	0	1	0	0001	0011
Cursor 5	SUBIDA	0	0	1	0	0	0001	0000
Cursor 6	SUBIDA	0	0	1	0	1	1000	1100