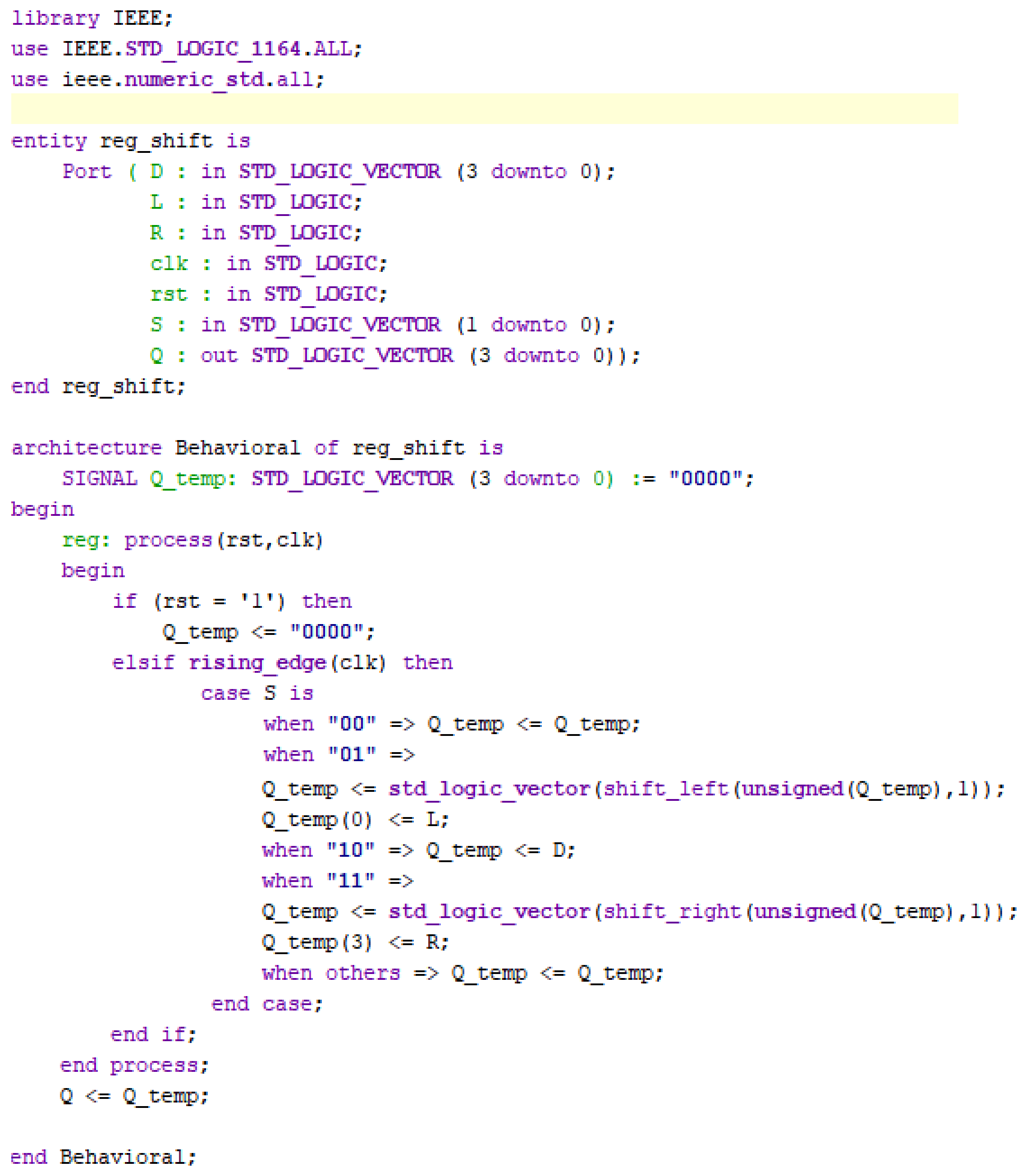
PRACTICA 4 – SHIFT REGISTER

CODIGO



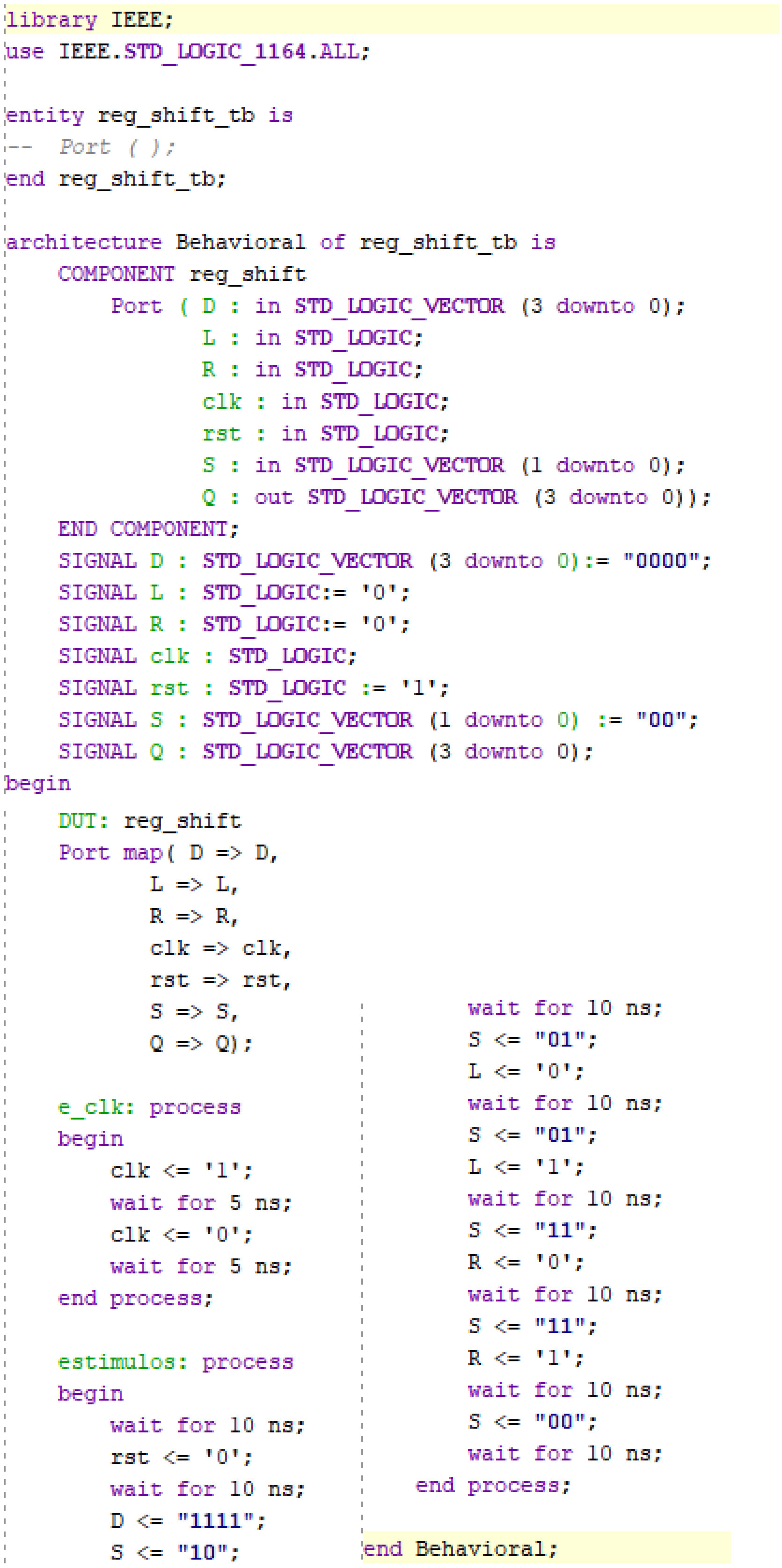
```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
use ieee.numeric_std.all;

entity reg_shift is
    Port ( D : in STD_LOGIC_VECTOR (3 downto 0);
           L : in STD_LOGIC;
           R : in STD_LOGIC;
           clk : in STD_LOGIC;
           rst : in STD_LOGIC;
           S : in STD_LOGIC_VECTOR (1 downto 0);
           Q : out STD_LOGIC_VECTOR (3 downto 0));
end reg_shift;

architecture Behavioral of reg_shift is
    SIGNAL Q_temp: STD_LOGIC_VECTOR (3 downto 0) := "0000";
begin
    reg: process(rst,clk)
    begin
        if (rst = '1') then
            Q_temp <= "0000";
        elsif rising_edge(clk) then
               case S is
                    when "00" => Q_temp <= Q_temp;
                    when "01" =>
                    Q_temp <= std_logic_vector(shift_left(unsigned(Q_temp),1));
                    Q_temp(0) <= L;
                    when "10" => Q_temp <= D;
                    when "11" =>
                    Q_temp <= std_logic_vector(shift_right(unsigned(Q_temp),1));
                    Q_temp(3) <= R;
                    when others => Q_temp <= Q_temp;
                end case;
        end if;
    end process;
    Q <= Q_temp;

end Behavioral;
```

TEST BENCH



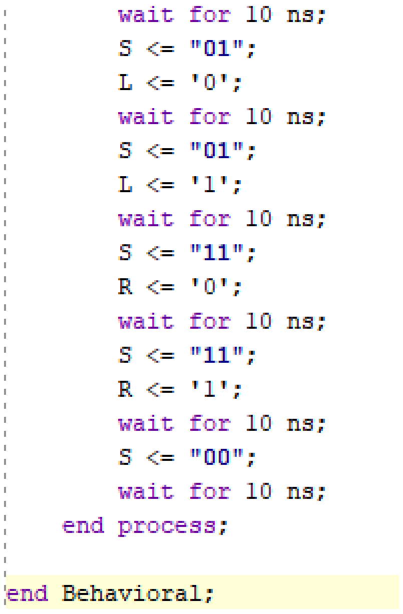
```vhdl
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;

entity reg_shift_tb is
--  Port ( );
end reg_shift_tb;

architecture Behavioral of reg_shift_tb is
    COMPONENT reg_shift
        Port ( D : in STD_LOGIC_VECTOR (3 downto 0);
               L : in STD_LOGIC;
               R : in STD_LOGIC;
               clk : in STD_LOGIC;
               rst : in STD_LOGIC;
               S : in STD_LOGIC_VECTOR (1 downto 0);
               Q : out STD_LOGIC_VECTOR (3 downto 0));
    END COMPONENT;
    SIGNAL D : STD_LOGIC_VECTOR (3 downto 0):= "0000";
    SIGNAL L : STD_LOGIC:= '0';
    SIGNAL R : STD_LOGIC:= '0';
    SIGNAL clk : STD_LOGIC;
    SIGNAL rst : STD_LOGIC := '1';
    SIGNAL S : STD_LOGIC_VECTOR (1 downto 0)  := "00";
    SIGNAL Q : STD_LOGIC_VECTOR (3 downto 0);
begin
    DUT: reg_shift
    Port map( D => D,
            L => L,
            R => R,
            clk => clk,
            rst => rst,
            S => S,
            Q => Q);

    e_clk: process
    begin
        clk <= '1';
        wait for 5 ns;
        clk <= '0';
        wait for 5 ns;
    end process;

    estimulos: process
    begin
        wait for 10 ns;
        rst <= '0';
        wait for 10 ns;
        D <= "1111";
        S <= "10";
```



```vhdl
        wait for 10 ns;
        S <= "01";
        L <= '0';
        wait for 10 ns;
        S <= "01";
        L <= '1';
        wait for 10 ns;
        S <= "11";
        R <= '0';
        wait for 10 ns;
        S <= "11";
        R <= '1';
        wait for 10 ns;
        S <= "00";
        wait for 10 ns;
    end process;

end Behavioral;
```