```
-------------------------------------------------------------------------------
-- Design: VHDL Class Example 5 : design of sequential logic                 --
--                                                                           --
-- Author : Roland Hoeller                                                   --
-- Date   : 27 04 2000                                                       --
-- File   : sh_reg.vhd                                                       --
-------------------------------------------------------------------------------

library IEEE;
use IEEE.std_logic_1164.all;

entity sh_reg is
port (d_i :       in   std_logic_vector(3 downto 0);
      en_i :      in   std_logic;
      sh_i :      in   std_logic;
      clk   :     in   std_logic;
      reset   :   in   std_logic;
      q_o :       out  std_logic_vector(3 downto 0));
end sh_reg;

architecture rtl of sh_reg is

  signal s_q : std_logic_vector(3 downto 0);

begin
  -- The following process describes a simple D-FF with enable. Furthermore a
  -- shift operation can be performed depending on the control signals.
  p_sh_reg : process (clk, reset)
  begin
    if reset = '1' then
      s_q <= "0000";
    elsif clk'event and clk = '1' then
      if en_i = '1' then
        if sh_i = '1' then
          s_q(3 downto 1) <= s_q(2 downto 0);
          s_q(0) <= '0';
        else
          s_q <= d_i;
        end if;
      end if;
    end if;
  end process p_sh_reg;

  -- This line simply connects the register s_q to the output of the design
  q_o <= s_q;

end rtl;
```

```
-------------------------------------------------------------------------------
-- Design: VHDL Class Example 5 : design of sequential logic, testbench      --
--                                                                           --
-- Author : Roland Hoeller                                                   --
-- Date   : 27 04 2000                                                       --
-- File   : tb_sh_reg.vhd                                                    --
-------------------------------------------------------------------------------

library IEEE;
use IEEE.std_logic_1164.all;

entity tb_sh_reg is end tb_sh_reg;

architecture sim of tb_sh_reg is

  -- Declaration of the component under test
  component sh_reg
    port (
      d_i   : in  std_logic_vector(3 downto 0);
      en_i  : in  std_logic;
      sh_i  : in  std_logic;
      clk   : in  std_logic;
      reset : in  std_logic;
      q_o   : out std_logic_vector(3 downto 0));
  end component;

  signal d_i   : std_logic_vector(3 downto 0);
  signal en_i  : std_logic;
  signal sh_i  : std_logic;
  signal clk   : std_logic;
  signal reset : std_logic;
  signal q_o   : std_logic_vector(3 downto 0);

begin

  -- Instantiate the design under test
  i_sh_reg : sh_reg
    port map (
      d_i   => d_i,
      en_i  => en_i,
      sh_i  => sh_i,
      clk   => clk,
      reset => reset,
      q_o   => q_o);

  -- Generate clock
  p_clk : process
  begin
    clk <= '0';
    wait for 50 ns;
    clk <= '1';
    wait for 50 ns;
  end process p_clk;

  -- Generate reset
  p_reset : process
  begin
    reset <= '1';
    wait for 120 ns;
    reset <= '0';
    wait;
  end process p_reset;

  p_stim : process
  begin
    d_i <= "0000";
    en_i <= '0';
    sh_i <= '0';
    wait for 310 ns;
```

```
      d_i <= "0000";
      en_i <= '1';
      sh_i <= '0';
      wait for 400 ns;
      d_i <= "1111";
      en_i <= '1';
      sh_i <= '0';
      wait for 400 ns;
      d_i <= "0101";
      en_i <= '0';
      sh_i <= '0';
      wait for 100 ns;
      d_i <= "0111";
      en_i <= '1';
      sh_i <= '1';
      wait for 400 ns;
      d_i <= "1010";
      en_i <= '1';
      sh_i <= '0';
      wait for 400 ns;
      -- stop simulation
      assert false report "END OF SIMULATION" severity error;
    end process p_stim;

  end sim;
```