```vhdl
--------------------------------------------------------------------------------
-- Company: 
-- Engineer:
--
-- Create Date:   15:42:20 08/24/2023
-- Design Name:   
-- Module Name:   C:/Users/ETC-VLSI LAB/Downloads/Xilinx_ISE_DS_14.7_1015_1_2/shiftregister/shiftregisterclk.vhd
-- Project Name:  shiftregister
-- Target Device:  
-- Tool versions:  
-- Description:   
-- 
-- VHDL Test Bench Created by ISE for module: shiftregistrer
-- 
-- Dependencies:
-- 
-- Revision:
-- Revision 0.01 - File Created
-- Additional Comments:
--
-- Notes: 
-- This testbench has been automatically generated using types std_logic and
-- std_logic_vector for the ports of the unit under test.  Xilinx recommends
-- that these types always be used for the top-level I/O of a design in order
-- to guarantee that the testbench will bind correctly to the post-implementation 
-- simulation model.
--------------------------------------------------------------------------------
LIBRARY ieee;
USE ieee.std_logic_1164.ALL;
 
-- Uncomment the following library declaration if using
-- arithmetic functions with Signed or Unsigned values
--USE ieee.numeric_std.ALL;
 
ENTITY shiftregisterclk IS
END shiftregisterclk;
 
ARCHITECTURE behavior OF shiftregisterclk IS 
 
    -- Component Declaration for the Unit Under Test (UUT)
 
    COMPONENT shiftregistrer
    PORT(
         s : IN  std_logic_vector(3 downto 0);
         m : IN  std_logic_vector(1 downto 0);
         y : OUT  std_logic_vector(3 downto 0);
         clk : IN  std_logic;
         rst : IN  std_logic
        );
    END COMPONENT;
    

   --Inputs
   signal s : std_logic_vector(3 downto 0) := (others => '0');
   signal m : std_logic_vector(1 downto 0) := (others => '0');
   signal clk : std_logic := '0';
   signal rst : std_logic := '0';

   --Outputs
   signal y : std_logic_vector(3 downto 0);

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

BEGIN

   -- Instantiate the Unit Under Test (UUT)
   uut: shiftregistrer PORT MAP (
          s => s,
          m => m,
          y => y,
          clk => clk,
          rst => rst
        );

   -- Clock process definitions
   clk_process :process
   begin
      clk <= '1';
      wait for clk_period/2;
      clk <= '0';
      wait for clk_period/2;
   end process;


   -- Stimulus process
   stim_proc: process
   begin
      -- hold reset state for 100 ns.
      rst<='1';
     wait for 10 ns;
      rst<='0';
       m<="00";
       s<="0001";
       wait for 80 ns;
       s<="1110";
       wait for 80 ns;
         s<="0000";
         wait for 80 ns;

         m<="01";
       s<="0001";
       wait for 80 ns;
       s<="1110";
       wait for 80 ns;
         s<="0000";
         wait for 80 ns;

         m<="10";
       s<="0001";
       wait for 80 ns;
       s<="1110";
       wait for 80 ns;
```

```
            s<="0000";
            wait for 80 ns;

             m<="11";
          s<="0001";
          wait for 80 ns;
          s<="1110";
          wait for 80 ns;
            s<="0000";
            wait for 80 ns;


         --wait for clk_period*10;

         -- insert stimulus here

         wait;
      end process;

   END;

```