Universal shift register simple code

```
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;

-- Uncomment the following library declaration if instantiating
-- any Xilinx primitives in this code.
--library UNISIM;
--use UNISIM.VComponents.all;

entity usreg is
    Port ( Din : in  STD_LOGIC_VECTOR (3 downto 0);
           Clk : in  STD_LOGIC;
           rst : in  STD_LOGIC;
           S : in  STD_LOGIC_VECTOR (1 downto 0);
           Dout : inout  STD_LOGIC_VECTOR (3 downto 0));
end usreg;

architecture Behavioral of usreg is
signal msbin,lsbin:std_logic;
begin
process(Clk,rst)
begin
if(rst='1') then
Dout<="0000";
elsif Clk' event and Clk='1' then
msbin<=din(3);
```

```
lsbin<=din(0);
case S is

when "00"=>Dout<=Dout; --hold
when "01"=>Dout<=msbin &Dout(3 downto 1); --right shift
when "10"=>Dout<=Dout(2 downto 0)& lsbin; --left shift
when "11"=>Dout<=Din; --parallel
when others=>Dout<="XXXX";
end case;
end if;
end process;
end Behavioral;
```

Universal shift regisister test bench code

```
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 usregtb IS
END usregtb;

ARCHITECTURE behavior OF usregtb IS

   -- Component Declaration for the Unit Under Test (UUT)

   COMPONENT usreg
   PORT(
       Din : IN  std_logic_vector(3 downto 0);
```

```
        Clk : IN  std_logic;
        rst : IN  std_logic;
        S : IN  std_logic_vector(1 downto 0);
        Dout : INOUT  std_logic_vector(3 downto 0)
       );
    END COMPONENT;
   --Inputs
   signal Din : std_logic_vector(3 downto 0) := (others => '0');
   signal Clk : std_logic := '0';
   signal rst : std_logic := '0';
   signal S : std_logic_vector(1 downto 0) := (others => '0');

            --BiDirs
   signal Dout : std_logic_vector(3 downto 0);

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

BEGIN
            -- Instantiate the Unit Under Test (UUT)
   uut: usreg PORT MAP (
        Din => Din,
        Clk => Clk,
        rst => rst,
        S => S,
        Dout => Dout
       );

   -- Clock process definitions
   Clk_process :process
   begin
```

```
                Clk <= '0';
                wait for Clk_period/2;
                Clk <= '1';
                wait for Clk_period/2;
  end process;

  -- Stimulus process
  stim_proc: process
  begin
    -- hold reset state for 100 ns.
    wait for 100 ns;
    rst<='1';
    assert Dout<="0000"report"error";
    wait for Clk_period*10;
                rst<='0';
                Din<="1000";
                S<="01";
                assert Dout<="1000"report"error";
    wait for Clk_period/2;
                assert Dout<="1100"report"error";
    wait for Clk_period/2;
                assert Dout<="1110"report"error";
    wait for Clk_period/2;
                assert Dout<="1111"report"error";
    wait for Clk_period/2;
    wait;
  end process;
END;
```