

-- **** STUDENT: 64000225.....	2
-- KOMENTARJI K OCENI NALOGE -- Matej Možek: Ni pripomb.....	2
-- **** STUDENT: 64200100.....	5
-- KOMENTARJI K OCENI NALOGE -- Matej Možek: Ni pripomb.....	5
-- **** STUDENT: 64200163.....	8
-- KOMENTARJI K OCENI NALOGE -- Matej Možek: Ni pripomb.....	8
-- **** STUDENT: 64200238.....	11
-- KOMENTARJI K OCENI NALOGE -- Matej Možek: Ni pripomb.....	11
-- **** STUDENT: 64200288.....	14
-- KOMENTARJI K OCENI NALOGE -- Matej Možek: Ni pripomb.....	14
-- **** STUDENT: 64200296.....	17
-- KOMENTARJI K OCENI NALOGE -- Matej Možek: Ni pripomb.....	17
-- **** STUDENT: 64200385.....	21
-- KOMENTARJI K OCENI NALOGE -- Matej Možek: Ni pripomb.....	21
-- **** STUDENT: 64210113.....	24
-- KOMENTARJI K OCENI NALOGE -- Matej Možek: Ni pripomb.....	24
-- **** STUDENT: 64210132.....	28
-- KOMENTARJI K OCENI NALOGE -- Matej Možek: Ni pripomb.....	28
-- **** STUDENT: 64210290.....	32
-- KOMENTARJI K OCENI NALOGE -- Matej Možek: Ni pripomb.....	32
-- **** STUDENT: 64210382.....	35
-- KOMENTARJI K OCENI NALOGE -- Matej Možek: Ni pripomb.....	35
-- **** STUDENT: 64210384.....	38
-- KOMENTARJI K OCENI NALOGE -- Matej Možek: Ni pripomb.....	38
-- **** STUDENT: 64210386.....	42
-- KOMENTARJI K OCENI NALOGE -- Matej Možek: Ni pripomb.....	42
-- **** STUDENT: 64210445.....	46
-- KOMENTARJI K OCENI NALOGE -- Matej Možek: Ni pripomb.....	46
-- **** STUDENT: 64210455.....	50
-- KOMENTARJI K OCENI NALOGE -- Matej Možek: Ni pripomb.....	50
-- **** STUDENT: 64210457.....	54
-- KOMENTARJI K OCENI NALOGE -- Matej Možek: Ni pripomb.....	54
-- **** STUDENT: 64240430.....	57
-- KOMENTARJI K OCENI NALOGE -- Matej Možek: Ni pripomb.....	57

```

-- *****
-- **** STUDENT: 64000225
-- *****
-- KOMENTARJI K OCENI NALOGE
-- Matej Možek: Ni pripomb
-- *****

```

```

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 shift_reg_tb IS

```

```

    generic( reg_size: natural := 4 );

```

```

END shift_reg_tb;

```

```

ARCHITECTURE behavior OF shift_reg_tb IS

```

```

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

```

```

    COMPONENT shift_reg
        generic( reg_size: natural := 4 );
    PORT(
        clk : IN std_logic;
        nCLR : IN std_logic;
        sr_in : IN std_logic;
        sl_in : IN std_logic;
        s : IN std_logic_vector( 1 downto 0 );
        x : IN std_logic_vector( 3 downto 0 );
        Q : OUT std_logic_vector( 3 downto 0 )
    );
END COMPONENT;

```

```

    -- Inputs

```

```

    signal      clk : std_logic := '0';

```

```

signal          nCLR : std_logic := '0';
signal          sr_in : std_logic := '0';
signal          sl_in : std_logic := '0';
signal          s : std_logic_vector( 1 downto 0 ) := ( others => '0' );
signal          x : std_logic_vector( 3 downto 0 ) := ( others => '0' );

-- Outputs
signal          Q : std_logic_vector( reg_size-1 downto 0 );

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

BEGIN

    -- Instantiate the Unit Under Test ( UUT )
    uut: shift_reg
        generic map ( reg_size => reg_size )
        PORT MAP (
            clk => clk,
            nCLR => nCLR,
            sr_in => sr_in,
            sl_in => sl_in,
            s => s,
            x => x,
            Q => Q
        );

    -- 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;
```

```
wait for clk_period*10;
```

```
    x    <= "1011";  
    s    <= "11";  
    wait for clk_period;  
    s    <= "00";  
    wait for clk_period;  
    s    <= "10";  
    wait for clk_period;  
    sl_in <= '1';  
    wait for clk_period;  
    sl_in <= '0';  
    s    <= "01";  
    wait for clk_period;  
    sr_in <= '1';  
    wait for clk_period;  
    s    <= "00";  
    wait for clk_period;  
    nCLR <= '0';  
    wait for clk_period/10;  
    nCLR <= '1';
```

```
wait;  
end process;
```

```
END;
```

```

-- *****
-- **** STUDENT: 64200100
-- *****
-- KOMENTARJI K OCENI NALOGE
-- Matej Možek: Ni pripomb
-- *****
LIBRARY ieee;
USE ieee.std_logic_1164.ALL;

ENTITY shift_reg_tb IS
    generic( reg_size: natural := 4 );
END shift_reg_tb;

ARCHITECTURE behavior OF shift_reg_tb IS
    COMPONENT shift_reg
        generic( reg_size: natural := 4 );
    PORT(
        clk : IN std_logic;
        nCLR : IN std_logic;
        sr_in : IN std_logic;
        sl_in : IN std_logic;
        s : IN std_logic_vector( 1 downto 0 );
        x : IN std_logic_vector( reg_size-1 downto 0 );
        Q : OUT std_logic_vector( reg_size-1 downto 0 );
    END COMPONENT;

    signal      clk : std_logic := '0';
    signal      nCLR : std_logic := '1';
    signal      sr_in : std_logic := '0';
    signal      sl_in : std_logic := '0';
    signal      s : std_logic_vector( 1 downto 0 ) := ( others => '0' );
    signal      x : std_logic_vector( reg_size-1 downto 0 ) := ( others => '0' );
    signal      Q : std_logic_vector( reg_size-1 downto 0 );
    constant    clkP : time := 10 ns;

BEGIN

    uut: shift_reg
        generic map ( reg_size => reg_size )
        PORT MAP (

```

```
clk => clk,  
nCLR => nCLR,  
sr_in => sr_in,  
sl_in => sl_in,  
s => s,  
x => x,  
Q => Q );
```

```
clk_process :process  
begin  
clk    <= '0';  
wait for clkP/2;  
clk    <= '1';  
wait for clkP/2;  
end process;
```

```
sproc: process  
begin  
wait for 100 ns;  
wait for clkP*10;
```

```
x      <= "1011";  
s      <= "11";  
wait for clkP;  
s      <= "00";  
wait for clkP;  
s      <= "10";  
wait for clkP;  
sl_in  <= '1';  
wait for clkP;  
sl_in  <= '0';  
s      <= "01";  
wait for clkP;  
sr_in  <= '1';  
wait for clkP;  
s      <= "00";  
wait for clkP;  
nCLR   <= '0';  
wait for clkP/10;  
nCLR   <= '1';
```

```
wait;  
end process;  
END;
```

```

-- *****
-- **** STUDENT: 64200163
-- *****
-- KOMENTARJI K OCENI NALOGE
-- Matej Možek: Ni pripomb
-- *****

library ieee;
use ieee.std_logic_1164.all;
use ieee.numeric_std.ALL;

entity shift_reg_tb is
    generic( reg_size : natural := 4 );
end shift_reg_tb;

architecture behavior of shift_reg_tb is
    component shift_reg
        generic( reg_size : natural := 4 );
        port(
            clk : in std_logic;
            nCLR : in std_logic;
            sr_in : in std_logic;
            sl_in : in std_logic;
            s : in std_logic_vector( 1 downto 0 );
            x : in std_logic_vector( reg_size - 1 downto 0 );
            Q : out std_logic_vector( reg_size - 1 downto 0 )
        );
    end component;

    signal          clk : std_logic := '0';
    signal          nCLR : std_logic := '1';
    signal          sr_in : std_logic := '0';
    signal          sl_in : std_logic := '0';
    signal          s : std_logic_vector( 1 downto 0 ) := ( others => '0' );
    signal          x : std_logic_vector( reg_size - 1 downto 0 ) := ( others => '0' );
    signal          Q : std_logic_vector( reg_size - 1 downto 0 );
    constant        clk_period : time := 10 ns;
begin
    uut: shift_reg port map(
        clk => clk,
        nCLR => nCLR,

```



```

sr_in => sr_in,
sl_in => sl_in,
s => s,
x => x,
Q => Q
);
clk_process : process
begin
    clk    <= '0';
    wait for clk_period/2;
    clk    <= '1';
    wait for clk_period/2;
end process;
stim_proc : process
begin
    wait for clk_period*5;
    x      <= "1110";
    wait for clk_period;
    s      <= "11";
    wait for clk_period;
    s      <= "00";
    wait for clk_period;
    s      <= "10";
    wait for clk_period;
    sl_in  <= '1';
    wait for clk_period;
    sl_in  <= '0';
    wait for clk_period;
    s      <= "01";
    wait for clk_period;
    sr_in  <= '1';
    wait for clk_period;
    sr_in  <= '0';
    wait for clk_period;
    s      <= "00";
    wait for clk_period;
    nCLR   <= '0';
    wait for clk_period*5;
    nCLR   <= '1';

wait;

```

```
end process;  
end;
```

```

-- *****
-- **** STUDENT: 64200238
-- *****
-- KOMENTARJI K OCENI NALOGE
-- Matej Možek: Ni pripomb
-- *****

LIBRARY ieee;
USE ieee.std_logic_1164.all;
USE ieee.numeric_std.all;

ENTITY shift_reg_tb IS
generic( reg_size: natural := 4 );
END shift_reg_tb;

ARCHITECTURE behavior OF shift_reg_tb IS

    COMPONENT shift_reg
    GENERIC (
        reg_size : natural := 4
    );
    PORT (
        clk : in std_logic;
        nCLR : in std_logic;
        sr_in : in std_logic;
        sl_in : in std_logic;
        s : in std_logic_vector( 1 downto 0 );
        x : in std_logic_vector( reg_size-1 downto 0 );
        Q : out std_logic_vector( reg_size-1 downto 0 )
    );
    END COMPONENT;

    -- Signali za povezavo s testirano komponento
    signal      clk : std_logic := '0';
    signal      nCLR : std_logic := '1';
    signal      sr_in : std_logic := '0';
    signal      sl_in : std_logic := '0';
    signal      s : std_logic_vector( 1 downto 0 ) := "00";
    signal      x : std_logic_vector( 3 downto 0 ) := ( others => '0' );
    signal      Q : std_logic_vector( 3 downto 0 );

```

```
    -- Signal          za simulacijo ure  
    constant    clk_period : time := 10 ns;
```

```
BEGIN
```

```
    uut: shift_reg  
        generic map ( reg_size => reg_size )  
        PORT MAP (  
    clk => clk,  
    nCLR => nCLR,  
    sr_in => sr_in,  
    sl_in => sl_in,  
    s => s,  
    x => x,  
    Q => Q  
    );
```

```
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
```

```
wait for clk_period*10;
```

```
    -- insert stimulus here
```

```
    -- nCLR    <= '0';  
    x        <= "0101";  
    s        <= "00";  
    wait for clk_period*3;  
    s        <= "01";  
    wait for clk_period*3;  
    s        <= "10";  
    wait for clk_period*3;  
    s        <= "11";
```

```
wait for clk_period*3;
nCLR  <='0';
wait for clk_period*3;
nCLR  <='1';
wait for clk_period*3;
nCLR  <='0';
x      <= "1010";
wait for clk_period*3;
s      <= "00";
wait for clk_period*3;
s      <= "01";
wait for clk_period*3;
s      <= "10";
wait for clk_period*3;
s      <= "11";
wait for clk_period*3;
nCLR  <='0';
wait for clk_period*3;
nCLR  <='1';
wait for clk_period*3;
nCLR  <='0';
```

```
wait;
```

```
end process;
```

```
END behavior;
```

```

-- *****
-- **** STUDENT: 64200288
-- *****
-- KOMENTARJI K OCENI NALOGE
-- Matej Možek: Ni pripomb
-- *****
LIBRARY ieee;
USE ieee.std_logic_1164.all;
USE ieee.numeric_std.all;

ENTITY shift_reg_tb IS
generic( reg_size: natural := 4 );
END shift_reg_tb;

ARCHITECTURE behavior OF shift_reg_tb IS
  COMPONENT shift_reg
    GENERIC (
      reg_size : natural := 4
    );
    PORT (
      clk : in std_logic;
      nCLR : in std_logic;
      sr_in : in std_logic;
      sl_in : in std_logic;
      s : in std_logic_vector( 1 downto 0 );
      x : in std_logic_vector( reg_size-1 downto 0 );
      Q : out std_logic_vector( reg_size-1 downto 0 )
    );
  END COMPONENT;

  signal          clk : std_logic := '0';
  signal          nCLR : std_logic := '1';
  signal          sr_in : std_logic := '0';
  signal          sl_in : std_logic := '0';
  signal          s : std_logic_vector( 1 downto 0 ) := "00";
  signal          x : std_logic_vector( 3 downto 0 ) := ( others => '0' );
  signal          Q : std_logic_vector( 3 downto 0 );

  constant        ClockPeriode : time := 10 ns;

```

```

BEGIN
  uut: shift_reg
    generic map ( reg_size => reg_size )
    PORT MAP (
      clk => clk,
      nCLR => nCLR,
      sr_in => sr_in,
      sl_in => sl_in,
      s => s,
      x => x,
      Q => Q
    );

  clk_process :process
  begin
    clk    <= '0';
    wait for ClockPeriode/2;
    clk    <= '1';
    wait for ClockPeriode/2;
  end process;

  s_proc: process
  begin

    wait for ClockPeriode*5;
    x    <= "0101";
    s    <= "00";
    wait for ClockPeriode*5;
    s    <= "10";
    wait for ClockPeriode*5;
    s    <= "01";
    wait for ClockPeriode*5;
    s    <= "11";
    wait for ClockPeriode*5;
    nCLR <= '0';
    wait for ClockPeriode*5;
    nCLR <= '1';
    wait for ClockPeriode*5;
    nCLR <= '0';
    x    <= "1010";

```

```
wait for ClockPeriode*5;  
s      <= "00";  
wait for ClockPeriode*5;  
s      <= "01";  
wait for ClockPeriode*5;  
s      <= "10";  
wait for ClockPeriode*5;  
s      <= "11";  
wait for ClockPeriode*5;  
nCLR   <='0';  
wait for ClockPeriode*5;  
nCLR   <='1';  
wait for ClockPeriode*5;  
nCLR   <='0';  
  
wait;  
  
end process;  
  
END behavior;
```



```

-- *****
-- **** STUDENT: 64200296
-- *****
-- KOMENTARJI K OCENI NALOGE
-- Matej Možek: Ni pripomb
-- *****
--      --      --      --      --      --      --      --      --      --      --      --      --      --      --      --      --
--      --      --      --      --      --      --      --      --      --      --      --      --      --      --      --      --
--      --      --      --      --      --
-- Company:
-- Engineer:
--
-- Create Date: 15:58:41 12/21/2024
-- Design Name:
-- Module Name: C:/Users/Tim/Downloads/FIFO/FIFO/shift_reg_tb.vhd
-- Project Name: FIFO
-- Target Device:
-- Tool versions:
-- Description:
--
-- VHDL Test Bench Created by ISE for module: shift_reg
--
-- 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 shift_reg_tb IS
    generic( reg_size: natural := 4 );
END shift_reg_tb;
```

```
ARCHITECTURE behavior OF shift_reg_tb IS
```

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

```
    COMPONENT shift_reg
    PORT(
        clk : IN std_logic;
        nCLR : IN std_logic;
        sr_in : IN std_logic;
        sl_in : IN std_logic;
        s : IN std_logic_vector( 1 downto 0 );
        x : IN std_logic_vector( reg_size-1 downto 0 );
        Q : OUT std_logic_vector( reg_size-1 downto 0 )
    );
    END COMPONENT;
```

```
    -- Inputs
```

```
    signal      clk : std_logic := '0';
    signal      nCLR : std_logic := '1';
    signal      sr_in : std_logic := '0';
    signal      sl_in : std_logic := '0';
    signal      s : std_logic_vector( 1 downto 0 ) := ( others => '0' );
    signal      x : std_logic_vector( reg_size-1 downto 0 ) := ( others => '0' );
```

```
    -- Outputs
```

```
    signal      Q : std_logic_vector( reg_size-1 downto 0 );
```

```
    -- Clock period definitions
```

```
    constant    clk_period : time := 10 ns;
```

```
BEGIN
```

```
    -- Instantiate the Unit Under Test ( UUT )
```

```
uut: shift_reg PORT MAP (  
  clk => clk,  
  nCLR => nCLR,  
  sr_in => sr_in,  
  sl_in => sl_in,  
  s => s,  
  x => x,  
  Q => Q  
);
```

```
    -- 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;  
  
    wait for clk_period*10;
```

```
    -- insert stimulus here
```

```
    x    <= "1010";  
    s    <= "11";  
    wait for clk_period;  
    s    <= "00";  
    wait for clk_period;  
    s    <= "10";  
    wait for clk_period;  
    sl_in <= '1';  
    wait for clk_period;  
    sl_in <= '0';
```

```
s      <= "01";  
wait for clk_period;  
sr_in <= '1';  
wait for clk_period;  
s      <= "00";  
wait for clk_period;  
nCLR   <= '0';  
wait for clk_period/10;  
nCLR   <= '1';
```

```
wait;  
end process;
```

```
END;
```

```

-- *****
-- **** STUDENT: 64200385
-- *****
-- KOMENTARJI K OCENI NALOGE
-- Matej Možek: Ni pripomb
-- *****
LIBRARY ieee;
USE ieee.std_logic_1164.ALL;
USE ieee.numeric_std.ALL;

ENTITY shift_reg_tb IS
    generic( reg_size: natural := 4 );
END shift_reg_tb;

ARCHITECTURE behavior OF shift_reg_tb IS

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

    COMPONENT shift_reg
        generic( reg_size: natural := 4 );
    PORT(
        clk : IN std_logic;
        nCLR : IN std_logic;
        sr_in : IN std_logic;
        sl_in : IN std_logic;
        s : IN std_logic_vector( 1 downto 0 );
        x : IN std_logic_vector( reg_size - 1 downto 0 );
        Q : OUT std_logic_vector( reg_size - 1 downto 0 )
    );
    END COMPONENT;

    -- Inputs
    signal      clk : std_logic := '0';
    signal      nCLR : std_logic := '0';
    signal      sr_in : std_logic := '0';
    signal      sl_in : std_logic := '0';
    signal      s : std_logic_vector( 1 downto 0 ) := ( others => '0' );
    signal      x : std_logic_vector( reg_size - 1 downto 0 ) := ( others => '0' );

    -- Outputs

```

```

signal          Q : std_logic_vector( reg_size - 1 downto 0 );

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

BEGIN

    -- Instantiate the Unit Under Test ( UUT )
    uut: shift_reg
        PORT MAP (
            clk => clk,
            nCLR => nCLR,
            sr_in => sr_in,
            sl_in => sl_in,
            s => s,
            x => x,
            Q => Q
        );

    -- 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;

        wait for clk_period*10;

        -- insert stimulus here

        x      <= "1001";
        sl_in  <= '1';

```

```
sr_in <= '1';  
nCLR  <= '1';  
wait for clk_period;  
for i in 0 to 3 loop  
    s      <= std_logic_vector( to_unsigned( i, s'length ) );  
    wait for clk_period*3;  
end loop;  
nCLR  <= '0';
```

```
wait;  
end process;
```

```
END;
```

```

-- *****
-- **** STUDENT: 64210113
-- *****
-- KOMENTARJI K OCENI NALOGE
-- Matej Možek: Ni pripomb
-- *****
--      --      --      --      --      --      --      --      --      --      --      --      --      --      --      --
--      --      --      --      --      --      --      --      --      --      --      --      --      --      --      --
--      --      --      --      --      --
-- Company:
-- Engineer:
--
-- Create Date: 11:43:47 12/18/2024
-- Design Name:
-- Module Name: C:/Users/Mihaheh/Downloads/Trash/FIFO/FIFO/shift_reg_tb.vhd
-- Project Name: FIFO
-- Target Device:
-- Tool versions:
-- Description:
--
-- VHDL Test Bench Created by ISE for module: shift_reg
--
-- 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;
USE ieee.numeric_std.ALL;

```



```

ENTITY shift_reg_tb IS
    generic( reg_size: natural := 4 );
END shift_reg_tb;

ARCHITECTURE behavior OF shift_reg_tb IS

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

    COMPONENT shift_reg
        generic( reg_size: natural := 4 );
    PORT(
        clk : IN std_logic;
        nCLR : IN std_logic;
        sr_in : IN std_logic;
        sl_in : IN std_logic;
        s : IN std_logic_vector( 1 downto 0 );
        x : IN std_logic_vector( reg_size-1 downto 0 );
        Q : OUT std_logic_vector( reg_size-1 downto 0 )
    );
    END COMPONENT;

    -- Inputs
    signal      clk : std_logic := '0';
    signal      nCLR : std_logic := '1';
    signal      sr_in : std_logic := '0';
    signal      sl_in : std_logic := '0';
    signal      s : std_logic_vector( 1 downto 0 ) := ( others => '0' );
    signal      x : std_logic_vector( reg_size-1 downto 0 ) := ( others => '0' );

    -- Outputs
    signal      Q : std_logic_vector( reg_size-1 downto 0 );

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

BEGIN

    -- Instantiate the Unit Under Test ( UUT )
    uut: shift_reg PORT MAP (

```

```

clk => clk,
nCLR => nCLR,
sr_in => sr_in,
sl_in => sl_in,
s => s,
x => x,
Q => Q
);

```

*-- 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

nCLR <= '0';
wait for clk_period * 2;
nCLR <= '1';
wait for clk_period * 2;

s    <= "11";
x    <= "1010";
wait for clk_period;

s    <= "10";
sl_in <= '1';

wait for clk_period* 2;

s    <= "01";
sr_in <= '0';

wait for clk_period* 2;

```

```
s    <= "00";  
wait for clk_period * 2;  
  
wait;  
end process;  
  
END;
```

```

-- *****
-- **** STUDENT: 64210132
-- *****
-- KOMENTARJI K OCENI NALOGE
-- Matej Možek: Ni pripomb
-- *****
--      --      --      --      --      --      --      --      --      --      --      --      --      --      --      --      --
--      --      --      --      --      --      --      --      --      --      --      --      --      --      --      --      --
--      --      --      --      --      --
-- Company:
-- Engineer:
--
-- Create Date: 03:19:47 12/23/2024
-- Design Name:
-- Module Name: C:/Users/blazk/Desktop/NDV/DN/fifo/shift_reg/shift_reg_tb.vhd
-- Project Name: shift_reg
-- Target Device:
-- Tool versions:
-- Description:
--
-- VHDL Test Bench Created by ISE for module: shift_reg
--
-- 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 shift_reg_tb IS
END shift_reg_tb;
```

```
ARCHITECTURE behavior OF shift_reg_tb IS
```

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

```
    COMPONENT shift_reg
    PORT(
        clk : IN std_logic;
        nCLR : IN std_logic;
        sr_in : IN std_logic;
        sl_in : IN std_logic;
        s : IN std_logic_vector( 1 downto 0 );
        x : IN std_logic_vector( 3 downto 0 );
        Q : OUT std_logic_vector( 3 downto 0 )
    );
    END COMPONENT;
```

```
    -- Inputs
```

```
    signal      clk : std_logic := '0';
    signal      nCLR : std_logic := '0';
    signal      sr_in : std_logic := '0';
    signal      sl_in : std_logic := '0';
    signal      s : std_logic_vector( 1 downto 0 ) := ( others => '0' );
    signal      x : std_logic_vector( 3 downto 0 ) := ( others => '0' );
```

```
    -- Outputs
```

```
    signal      Q : std_logic_vector( 3 downto 0 );
```

```
    -- Clock period definitions
```

```
    constant    clk_period : time := 10 ns;
```

```
BEGIN
```

```
    -- Instantiate the Unit Under Test ( UUT )
```

```

uut: shift_reg PORT MAP (
  clk => clk,
  nCLR => nCLR,
  sr_in => sr_in,
  sl_in => sl_in,
  s => s,
  x => x,
  Q => Q
);

-- 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;

    wait for clk_period*10;

    -- insert stimulus here

    sr_in <= '1';
    s     <= "01";

    wait for clk_period*4;

    sr_in <= '0';

    wait for clk_period*4;

    sl_in <= '1';
    s     <= "10";

```

```
wait for clk_period*4;

sl_in <= '0';

wait for clk_period*4;

x      <= "1001";
s      <= "11";

wait for clk_period;

s      <= "00";
x      <= "0110";

wait for clk_period*2;

s      <= "11";

wait for clk_period*1;

s      <= "00";
x      <= "1001";

wait;
end process;

END;
```

```

-- *****
-- **** STUDENT: 64210290
-- *****
-- KOMENTARJI K OCENI NALOGE
-- Matej Možek: Ni pripomb
-- *****
LIBRARY ieee;
USE ieee.std_logic_1164.ALL;
USE ieee.numeric_std.ALL;

ENTITY shift_reg_tb IS
END shift_reg_tb;

ARCHITECTURE behavior OF shift_reg_tb IS

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

    COMPONENT shift_reg
    PORT(
        clk : IN std_logic;
        nCLR : IN std_logic;
        sr_in : IN std_logic;
        sl_in : IN std_logic;
        s : IN std_logic_vector( 1 downto 0 );
        x : IN std_logic_vector( 3 downto 0 );
        Q : OUT std_logic_vector( 3 downto 0 )
    );
    END COMPONENT;

    -- Inputs
    signal      clk : std_logic := '0';
    signal      nCLR : std_logic := '0';
    signal      sr_in : std_logic := '0';
    signal      sl_in : std_logic := '0';
    signal      s : std_logic_vector( 1 downto 0 ) := ( others => '0' );
    signal      x : std_logic_vector( 3 downto 0 ) := ( others => '0' );

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

```



```

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

BEGIN

    -- Instantiate the Unit Under Test ( UUT )
    uut: shift_reg PORT MAP (
    clk => clk,
    nCLR => nCLR,
    sr_in => sr_in,
    sl_in => sl_in,
    s => s,
    x => x,
    Q => Q
    );

    -- 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;

        wait for clk_period*10;

        nCLR    <= '1';
        wait for clk_period;

        s        <= "11";
        for idi in 0 to 15 loop
            x        <= std_logic_vector( to_unsigned( idi,x'length ) );
            wait for clk_period;

```

```
end loop;
```

```
sl_in <= '0';
```

```
sr_in <= '1';
```

```
s      <= "10";
```

```
wait until Q="0000";
```

```
s      <= "01";
```

```
wait until Q="1111";
```

```
x      <= "0000";
```

```
s      <= "00";
```

```
wait;
```

```
end process;
```

```
END;
```

```

-- *****
-- **** STUDENT: 64210382
-- *****
-- KOMENTARJI K OCENI NALOGE
-- Matej Možek: Ni pripomb
-- *****
LIBRARY ieee;
USE ieee.std_logic_1164.ALL;

USE ieee.numeric_std.ALL;

ENTITY shift_reg_tb IS
    generic( reg_size: natural := 4 );
END shift_reg_tb;

ARCHITECTURE behavior OF shift_reg_tb IS

    COMPONENT shift_reg
        generic( reg_size: natural := 4 );
    PORT(
        clk : IN std_logic;
        nCLR : IN std_logic;
        sr_in : IN std_logic;
        sl_in : IN std_logic;
        s : IN std_logic_vector( 1 downto 0 );
        x : IN std_logic_vector( reg_size-1 downto 0 );
        Q : OUT std_logic_vector( reg_size-1 downto 0 )
    );
    END COMPONENT;

    signal          clk : std_logic := '0';
    signal          nCLR : std_logic := '1';
    signal          sr_in : std_logic := '0';
    signal          sl_in : std_logic := '0';
    signal          s : std_logic_vector( 1 downto 0 ) := ( others => '0' );
    signal          x : std_logic_vector( reg_size-1 downto 0 ) := ( others => '0' );

    signal          Q : std_logic_vector( reg_size-1 downto 0 );

```

```

constant    clk_period : time := 10 ns;

BEGIN

 uut: shift_reg
      generic map ( reg_size => reg_size )
      PORT MAP (
clk => clk,
nCLR => nCLR,
sr_in => sr_in,
sl_in => sl_in,
s => s,
x => x,
Q => Q
);

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

stim_proc: process
begin
wait for 100 ns;

wait for clk_period*10;

      x    <= "1011";
      s    <= "11";
      wait for clk_period;
      s    <= "00";
      wait for clk_period;
      s    <= "10";
      wait for clk_period;
      sl_in <= '1';
      wait for clk_period;
      sl_in <= '0';

```

```
s      <= "01";  
wait for clk_period;  
sr_in <= '1';  
wait for clk_period;  
s      <= "00";  
wait for clk_period;  
nCLR  <= '0';  
wait for clk_period/10;  
nCLR  <= '1';
```

```
wait;  
end process;
```

```
END;
```

```

-- *****
-- **** STUDENT: 64210384
-- *****
-- KOMENTARJI K OCENI NALOGE
-- Matej Možek: Ni pripomb
-- *****
--      --      --      --      --      --      --      --      --      --      --      --      --      --      --      --
--      --      --      --      --      --      --      --      --      --      --      --      --      --      --
--      --      --      --      --      --
-- Company:
-- Engineer:
--
-- Create Date: 20:02:31 12/20/2024
-- Design Name:
-- Module Name: C:/NDV/DN4/shift_reg_tb.vhd
-- Project Name: DN4
-- Target Device:
-- Tool versions:
-- Description:
--
-- VHDL Test Bench Created by ISE for module: shift_reg
--
-- 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 shift_reg_tb IS
    generic( reg_size: natural := 4 );
END shift_reg_tb;

ARCHITECTURE behavior OF shift_reg_tb IS

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

    COMPONENT shift_reg
        generic( reg_size: natural := 4 );
    PORT(
        clk : IN std_logic;
        nCLR : IN std_logic;
        sr_in : IN std_logic;
        sl_in : IN std_logic;
        s : IN std_logic_vector( 1 downto 0 );
        x : IN std_logic_vector( reg_size-1 downto 0 );
        Q : OUT std_logic_vector( reg_size-1 downto 0 )
    );
    END COMPONENT;

    -- Inputs
    signal      clk : std_logic := '0';
    signal      nCLR : std_logic := '1';
    signal      sr_in : std_logic := '0';
    signal      sl_in : std_logic := '0';
    signal      s : std_logic_vector( 1 downto 0 ) := ( others => '0' );
    signal      x : std_logic_vector( reg_size-1 downto 0 ) := ( others => '0' );

    -- Outputs
    signal      Q : std_logic_vector( reg_size-1 downto 0 );

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

BEGIN

```

```

    -- Instantiate the Unit Under Test ( UUT )
uut: shift_reg
    generic map ( reg_size => reg_size )
    PORT MAP (
clk => clk,
nCLR => nCLR,
sr_in => sr_in,
sl_in => sl_in,
s => s,
x => x,
Q => Q
);

    -- 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;

    wait for clk_period*10;

    -- insert stimulus here
    x    <= "1011";
    s    <= "11";
    wait for clk_period;
    s    <= "00";
    wait for clk_period;
    s    <= "10";
    wait for clk_period;
    sl_in <= '1';

```



```
wait for clk_period;  
sl_in <= '0';  
s     <= "01";  
wait for clk_period;  
sr_in <= '1';  
wait for clk_period;  
s     <= "00";  
wait for clk_period;  
nCLR  <= '0';  
wait for clk_period/10;  
nCLR  <= '1';
```

```
wait;  
end process;
```

```
END;
```

```

-- *****
-- **** STUDENT: 64210386
-- *****
-- KOMENTARJI K OCENI NALOGE
-- Matej Možek: Ni pripomb
-- *****
--      --      --      --      --      --      --      --      --      --      --      --      --      --      --      --
--      --      --      --      --      --      --      --      --      --      --      --      --      --      --
--      --      --      --      --      --
-- Company:
-- Engineer:
--
-- Create Date: 19:24:26 12/16/2024
-- Design Name:
-- Module Name: /home/ise/Xilinx_shared/domaca_naloga_4/shift_reg_tb.vhd
-- Project Name: domaca_naloga_4
-- Target Device:
-- Tool versions:
-- Description:
--
-- VHDL Test Bench Created by ISE for module: shift_reg
--
-- 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 shift_reg_tb IS
generic( reg_size: natural := 4 );
END shift_reg_tb;
```

```
ARCHITECTURE behavior OF shift_reg_tb IS
```

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

```
    COMPONENT shift_reg
        generic( reg_size: natural := 4 );
    PORT(
        clk : IN std_logic;
        nCLR : IN std_logic;
        sr_in : IN std_logic;
        sl_in : IN std_logic;
        s : IN std_logic_vector( 1 downto 0 );
        x : IN std_logic_vector( reg_size - 1 downto 0 );
        Q : OUT std_logic_vector( reg_size - 1 downto 0 )
    );
    END COMPONENT;
```

```
    -- Inputs
```

```
    signal      clk : std_logic := '0';
    signal      nCLR : std_logic := '1';
    signal      sr_in : std_logic := '0';
    signal      sl_in : std_logic := '0';
    signal      s : std_logic_vector( 1 downto 0 ) := ( others => '0' );
    signal      x : std_logic_vector( reg_size - 1 downto 0 ) := ( others => '0' );
```

```
    -- Outputs
```

```
    signal      Q : std_logic_vector( reg_size - 1 downto 0 );
```

```
    -- Clock period definitions
```

```
    constant    clk_period : time := 50 ns;
```

```
BEGIN
```

```

    -- Instantiate the Unit Under Test ( UUT )
uut: shift_reg
    generic map ( reg_size => reg_size )
    PORT MAP (
clk => clk,
nCLR => nCLR,
sr_in => sr_in,
sl_in => sl_in,
s => s,
x => x,
Q => Q
);

    -- 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;

    wait for clk_period*10;

    -- insert stimulus here

    x    <= "1011";
    s    <= "11";
    wait for clk_period;
    s    <= "00";
    wait for clk_period;
    s    <= "10";
    wait for clk_period;

```

```
sl_in <= '1';  
wait for clk_period;  
sl_in <= '0';  
s      <= "01";  
wait for clk_period;  
sr_in <= '1';  
wait for clk_period;  
s      <= "00";  
wait for clk_period;  
nCLR  <= '0';  
wait for clk_period/10;  
nCLR  <= '1';
```

```
wait;  
end process;
```

```
END;
```

```

-- *****
-- **** STUDENT: 64210445
-- *****
-- KOMENTARJI K OCENI NALOGE
-- Matej Možek: Ni pripomb
-- *****
--      --      --      --      --      --      --      --      --      --      --      --      --      --      --      --      --
--      --      --      --      --      --      --      --      --      --      --      --      --      --      --      --      --
--      --      --      --      --      --
-- Company:
-- Engineer:
--
-- Create Date: 10:38:43 12/21/2024
-- Design Name:
-- Module Name: C:/UN-LJ/NDV/DVaja4/FIFO/shift_reg_tb.vhd
-- Project Name: FIFO
-- Target Device:
-- Tool versions:
-- Description:
--
-- VHDL Test Bench Created by ISE for module: shift_reg
--
-- 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 shift_reg_tb IS
END shift_reg_tb;
```

```
ARCHITECTURE behavior OF shift_reg_tb IS
```

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

```
    COMPONENT shift_reg
    PORT(
        clk : IN std_logic;
        nCLR : IN std_logic;
        sr_in : IN std_logic;
        sl_in : IN std_logic;
        s : IN std_logic_vector( 1 downto 0 );
        x : IN std_logic_vector( 3 downto 0 );
        Q : OUT std_logic_vector( 3 downto 0 )
    );
    END COMPONENT;
```

```
    -- Inputs
```

```
    signal      clk : std_logic := '0';
    signal      nCLR : std_logic := '0';
    signal      sr_in : std_logic := '0';
    signal      sl_in : std_logic := '0';
    signal      s : std_logic_vector( 1 downto 0 ) := ( others => '0' );
    signal      x : std_logic_vector( 3 downto 0 ) := ( others => '0' );
```

```
    -- Outputs
```

```
    signal      Q : std_logic_vector( 3 downto 0 );
```

```
    -- Clock period definitions
```

```
    constant    clk_period : time := 10 ns;
```

```
BEGIN
```

```
    -- Instantiate the Unit Under Test ( UUT )
```

```

uut: shift_reg PORT MAP (
clk => clk,
nCLR => nCLR,
sr_in => sr_in,
sl_in => sl_in,
s => s,
x => x,
Q => Q
);

-- 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;

    -- insert stimulus here
    x    <= "1011";
    s    <= "11";
    wait for clk_period;
    s    <= "00";
    wait for clk_period;
    s    <= "10";
    wait for clk_period;
    sl_in <= '1';
    wait for clk_period;
    sl_in <= '0';
    s    <= "01";
    wait for clk_period;
    sr_in <= '1';
    wait for clk_period;

```



```
s      <= "00";  
wait for clk_period;  
nCLR   <= '0';  
wait for clk_period/10;  
nCLR   <= '1';
```

```
wait;  
end process;
```

```
END;
```

```

-- *****
-- **** STUDENT: 64210455
-- *****
-- KOMENTARJI K OCENI NALOGE
-- Matej Možek: Ni pripomb
-- *****
--      --      --      --      --      --      --      --      --      --      --      --      --      --      --      --      --
--      --      --      --      --      --      --      --      --      --      --      --      --      --      --      --      --
--      --      --      --      --      --
-- Company:
-- Engineer:
--
-- Create Date: 19:18:18 11/28/2022
-- Design Name:
-- Module Name: C:/Users/Saso/Documents/Xilinx_Projects/dn4/FIFO/shift_reg_tb.vhd
-- Project Name: FIFO
-- Target Device:
-- Tool versions:
-- Description:
--
-- VHDL Test Bench Created by ISE for module: shift_reg
--
-- 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 shift_reg_tb IS
    generic( reg_size: natural := 4 );
END shift_reg_tb;
```

```
ARCHITECTURE behavior OF shift_reg_tb IS
```

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

```
    COMPONENT shift_reg
        generic( reg_size: natural := 4 );
    PORT(
        clk : IN std_logic;
        nCLR : IN std_logic;
        sr_in : IN std_logic;
        sl_in : IN std_logic;
        s : IN std_logic_vector( 1 downto 0 );
        x : IN std_logic_vector( reg_size-1 downto 0 );
        Q : OUT std_logic_vector( reg_size-1 downto 0 )
    );
    END COMPONENT;
```

```
    -- Inputs
```

```
    signal      clk : std_logic := '0';
    signal      nCLR : std_logic := '1';
    signal      sr_in : std_logic := '0';
    signal      sl_in : std_logic := '0';
    signal      s : std_logic_vector( 1 downto 0 ) := ( others => '0' );
    signal      x : std_logic_vector( reg_size-1 downto 0 ) := ( others => '0' );
```

```
    -- Outputs
```

```
    signal      Q : std_logic_vector( reg_size-1 downto 0 );
```

```
    -- Clock period definitions
```

```
    constant    clk_period : time := 10 ns;
```

```
BEGIN
```

```

    -- Instantiate the Unit Under Test ( UUT )
uut: shift_reg
    generic map ( reg_size => reg_size )
    PORT MAP (
clk => clk,
nCLR => nCLR,
sr_in => sr_in,
sl_in => sl_in,
s => s,
x => x,
Q => Q
);

    -- 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;

    wait for clk_period*10;

    -- insert stimulus here
    x    <= "1011";
    s    <= "11";
    wait for clk_period;
    s    <= "00";
    wait for clk_period;
    s    <= "10";
    wait for clk_period;
    sl_in <= '1';

```

```
wait for clk_period;  
sl_in <= '0';  
s     <= "01";  
wait for clk_period;  
sr_in <= '1';  
wait for clk_period;  
s     <= "00";  
wait for clk_period;  
nCLR  <= '0';  
wait for clk_period/10;  
nCLR  <= '1';
```

```
wait;  
end process;
```

```
END;
```

```

-- *****
-- **** STUDENT: 64210457
-- *****
-- KOMENTARJI K OCENI NALOGE
-- Matej Možek: Ni pripomb
-- *****

```

```

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 shift_reg_tb IS
    generic( reg_size: natural := 4 );
END shift_reg_tb;

```

```

ARCHITECTURE behavior OF shift_reg_tb IS

```

```

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

```

```

    COMPONENT shift_reg
        generic( reg_size: natural := 4 );
    PORT(
        clk : IN std_logic;
        nCLR : IN std_logic;
        sr_in : IN std_logic;
        sl_in : IN std_logic;
        s : IN std_logic_vector( 1 downto 0 );
        x : IN std_logic_vector( reg_size-1 downto 0 );
        Q : OUT std_logic_vector( reg_size-1 downto 0 )
    );
END COMPONENT;

```

```

    -- Inputs

```

```

    signal      clk : std_logic := '0';
    signal      nCLR : std_logic := '1';
    signal      sr_in : std_logic := '0';
    signal      sl_in : std_logic := '0';

```

```

signal          s : std_logic_vector( 1 downto 0 ) := ( others => '0' );
signal          x : std_logic_vector( reg_size-1 downto 0 ) := ( others => '0' );

    -- Outputs
signal          Q : std_logic_vector( reg_size-1 downto 0 );

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

BEGIN

    -- Instantiate the Unit Under Test ( UUT )
    uut: shift_reg
        generic map ( reg_size => reg_size )
        PORT MAP (
            clk => clk,
            nCLR => nCLR,
            sr_in => sr_in,
            sl_in => sl_in,
            s => s,
            x => x,
            Q => Q
        );

    -- 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;

        wait for clk_period*10;

```

```
-- insert stimulus here
x      <= "1011";
s      <= "11";
wait for clk_period;
s      <= "00";
wait for clk_period;
s      <= "10";
wait for clk_period;
sl_in  <= '1';
wait for clk_period;
sl_in  <= '0';
s      <= "01";
wait for clk_period;
sr_in  <= '1';
wait for clk_period;
s      <= "00";
wait for clk_period;
nCLR   <= '0';
wait for clk_period/10;
nCLR   <= '1';
```

```
wait;
end process;
```

```
END;
```



```

-- *****
-- **** STUDENT: 64240430
-- *****
-- KOMENTARJI K OCENI NALOGE
-- Matej Možek: Ni pripomb
-- *****

library ieee;
use ieee.std_logic_1164.all;
use ieee.numeric_std.all;

-- Uncomment the following library declaration if using
-- arithmetic functions with Signed or Unsigned values
-- USE ieee.numeric_std.ALL;

ENTITY shift_reg_tb IS
    generic ( reg_size : natural := 4 );
END shift_reg_tb;

ARCHITECTURE behavior OF shift_reg_tb IS

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

    COMPONENT shift_reg
        generic ( reg_size : natural := 4 );
    PORT(
        clk : IN std_logic;
        nCLR : IN std_logic;
        sr_in : IN std_logic;
        sl_in : IN std_logic;
        s : IN std_logic_vector( 1 downto 0 );
        x : IN std_logic_vector( reg_size - 1 downto 0 );
        Q : OUT std_logic_vector( reg_size - 1 downto 0 )
    );
    END COMPONENT;

    -- Inputs
    signal      nCLR : std_logic := '1';
    signal      sr_in : std_logic := '0';
    signal      sl_in : std_logic := '0';
    signal      s : std_logic_vector( 1 downto 0 ) := ( others => '0' );

```

```

signal          x : std_logic_vector( reg_size - 1 downto 0 ) := ( others => '0' );

    -- Outputs
signal          Q : std_logic_vector( reg_size - 1 downto 0 );

    -- Clock period definitions
signal          clock : std_logic;
constant        clock_period : time := 100 ns;

BEGIN

    -- Instantiate the Unit Under Test ( UUT )
    uut: shift_reg
        generic map( reg_size => reg_size )
        PORT MAP (
            clk => clock,
            nCLR => nCLR,
            sr_in => sr_in,
            sl_in => sl_in,
            s => s,
            x => x,
            Q => Q
        );

    -- Clock process definitions
    clock_process : process
    begin
        clock <= '0';
        wait for clock_period/2;
        clock <= '1';
        wait for clock_period/2;
    end process;

    -- Stimulus process
    -- en primer procesa

    stimulus_process: process
    begin
        -- primer za paralelni ulaz za direktno uèitavanje vrijednosti u registar
        x      <= "0101";

```

```

        nCLR    <= '0';
-- vrednosti sl_in in sr_in
        sl_in   <= '0';
        sr_in   <= '1';
-- uporaba trostruke for petlje, zaradi lazjega speminjanja spremljenivki
-- nakon cetrtre for petlje, spreminaju se vrednosti
        for i in 0 to 1 loop
            for j in 0 to 1 loop
                for k in 0 to 1 loop
                    for l in 3 downto 0 loop
-- najprej se testira 11, 10, 01, 00
                        s    <= std_logic_vector( to_unsigned( l, s'length ) );
                        wait for clock_period;
                    end loop;
-- spreminjanje vrednosti sl_in, sr_in
                        sl_in <= not sl_in;
                        sr_in <= not sr_in;
                    end loop;
-- sprememba nCLR
                        nCLR  <= not nCLR;
                    end loop;
-- sprememba x
                        x    <= not x;
                    end loop;
                wait;
            end process;

```

```

END;

```