Problem

Implement a shift register with load enable LD and shift enable S both connected to switches. The input X is connected to the switches. The output Z is to the LEDs.

```
Entity shift_reg is
generic(n : natural);
Port ( X : in std_logic_vector(n-1 downto 0);
Z : out std_logic_vector(n-1 downto 0);
CK, LD, S : in std_logic);
End shift_reg;
```

The register shifts left by one bit position and the leftmost bit wraps around to the rightmost bit.

Use two-buttons single-step to demonstrate the correctness for 4-bit register.

```
Entity shift_reg_tb is
Port ( X : in std_logic_vector(3 downto 0);
       Z : out std_logic_vector(3 downto 0);
CK, LD, S, btn0, btn1 : in std_logic);
End shift_reg_tb;
library ieee;
use ieee.std_logic_1164.all;
```

Solution

```
Entity shift_reg is
generic(n : natural);
Port ( X : in std_logic_vector(n-1 downto 0);
Z : out std_logic_vector(n-1 downto 0);
CK, LD, S : in std_logic);
End shift_reg;

architecture beh of shift_reg is
signal temp : std_logic_vector(n-1 downto 0);
begin
-- architecture body
process(ck)
begin
if ck='1' and ck'event then
  if LD = '1' then
    temp <= x;
  elsif s = '1' then
    for i in n-1 downto 0 loop
      if i > 0 then
      temp(i) <= temp(i-1);
      else
      temp(i) <= temp(n-1);
      end if;
    end loop;
  else null;
 end if;
end if;
end process;

-- wire to out ports
z <= temp;

end beh;

-----------------------------------
-- Test the shift_reg with n = 4
-- Add single-step to demonstrate
-- correctness
-----------------------------------

library ieee;
use ieee.std_logic_1164.all;

Entity shift_reg_tb is
Port ( X : in std_logic_vector(3 downto 0);
       Z : out std_logic_vector(3 downto 0);
CK, LD, S, btn0, btn1 : in std_logic);
End shift_reg_tb;

architecture struc of shift_reg_tb is

-- component declare

component  shift_reg
generic(n: natural);
Port ( X : in std_logic_vector(n-1 downto 0);
Z : out std_logic_vector(n-1 downto 0);
CK, LD, S : in std_logic);
End component;

-- signal for single step

signal en: std_logic;

begin

U: shift_reg generic map(4) port
map(x => x, ck => en, s => s, ld => ld, z => z);
```

```
-- single step, debounce (db)
-- btn0 to enter and btn1 to reset

process(ck)
type db_state is (not_rdy, rdy, pulse);
variable db_ns: db_state;

begin

if ck='1' and ck'event then
  case db_ns is
   when not_rdy => en <= '0';
   if btn1 = '1' then db_ns := rdy; end if;
   when rdy => en <= '0';
   if btn0 = '1' then db_ns := pulse; end if;
   when pulse => en <= '1';
   db_ns := not_rdy;
   when others => null;
  end case;
end if;

end process;

end struc;
```

Design Constraint XDC file

```
## Clock signal

set_property PACKAGE_PIN W5 [get_ports CK]

        set_property IOSTANDARD LVCMOS33
[get_ports CK]
        #create_clock -add -name sys_clk_pin -period
10.00 -waveform {0 5} [get_ports clk]

## Switches
# sw[0] – sw[3]

set_property PACKAGE_PIN V17 [get_ports {X[0]}]

        set_property IOSTANDARD LVCMOS33
[get_ports {X[0]}]
set_property PACKAGE_PIN V16 [get_ports {X[1]}]

        set_property IOSTANDARD LVCMOS33
[get_ports {X[1]}]
set_property PACKAGE_PIN W16 [get_ports {X[2]}]

        set_property IOSTANDARD LVCMOS33
[get_ports {X[2]}]
set_property PACKAGE_PIN W17 [get_ports {X[3]}]

        set_property IOSTANDARD LVCMOS33
[get_ports {X[3]}]

## sw[14] and sw[15]
set_property PACKAGE_PIN T1 [get_ports S]

        set_property IOSTANDARD LVCMOS33
[get_ports S]
set_property PACKAGE_PIN R2 [get_ports LD]

        set_property IOSTANDARD LVCMOS33
[get_ports LD]

## LEDs
set_property PACKAGE_PIN U16 [get_ports {Z[0]}]

        set_property IOSTANDARD LVCMOS33
[get_ports {Z[0]}]
set_property PACKAGE_PIN E19 [get_ports {Z[1]}]

        set_property IOSTANDARD LVCMOS33
[get_ports {Z[1]}]
set_property PACKAGE_PIN U19 [get_ports {Z[2]}]

        set_property IOSTANDARD LVCMOS33
[get_ports {Z[2]}]
set_property PACKAGE_PIN V19 [get_ports {Z[3]}]

        set_property IOSTANDARD LVCMOS33
[get_ports {Z[3]}]

##Buttons
#set_property PACKAGE_PIN U18 [get_ports btnC]

        #set_property IOSTANDARD LVCMOS33
[get_ports btnC]
#set_property PACKAGE_PIN T18 [get_ports btnU]

        #set_property IOSTANDARD LVCMOS33
[get_ports btnU]
set_property PACKAGE_PIN W19 [get_ports btn0]

        set_property IOSTANDARD LVCMOS33
[get_ports btn0]
set_property PACKAGE_PIN T17 [get_ports btn1]

        set_property IOSTANDARD LVCMOS33
[get_ports btn1]
#set_property PACKAGE_PIN U17 [get_ports btnD]

        #set_property IOSTANDARD LVCMOS33
[get_ports btnD]
```