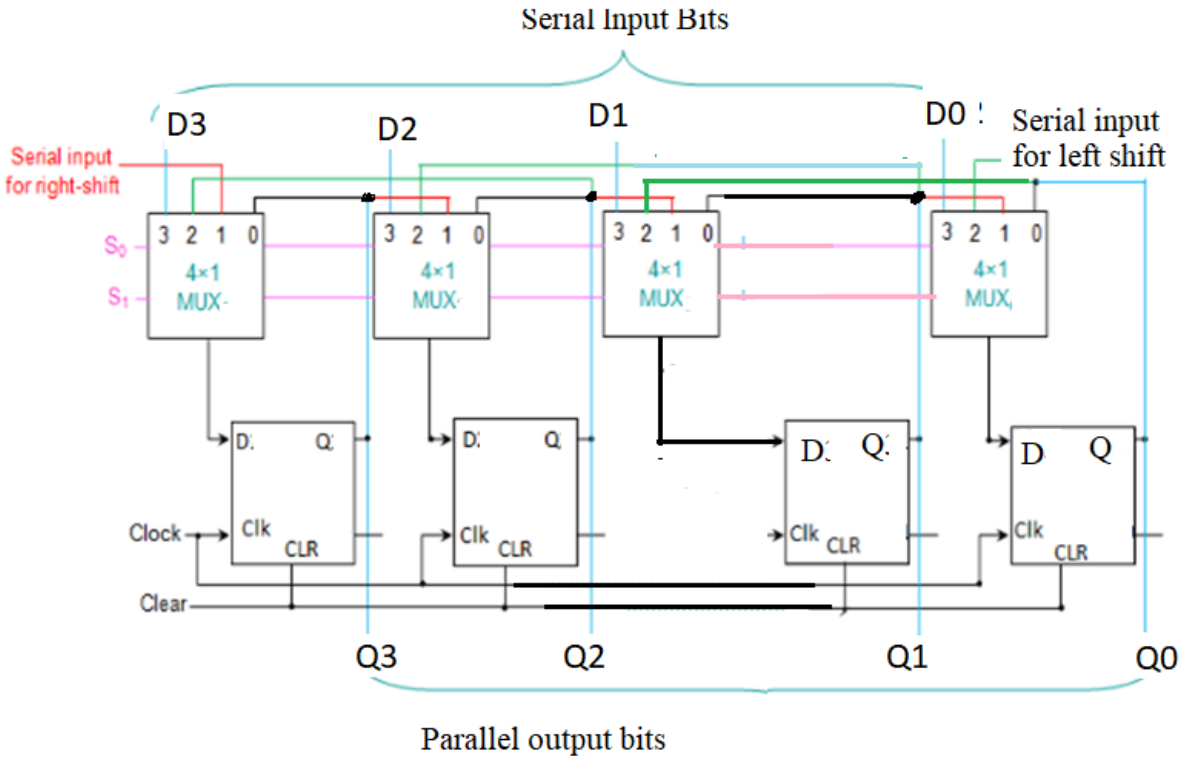
EXPERIMENT NO. 8

Aim: Write the VHDL code for the universal shift register. Perform the simulation and synthesis of the register.



Perform the following activities.

Activity 1: **Write the VHDL Code for D flip flop which should have synchronous clear input.**

```
------- Pos Edge triggered DFF
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;

entity Dff is
 port( D, clk,clr: in std_logic;
       q: out std_logic);
end dff;

architecture dff_arch of dff is

begin

process (clk,clr)
 begin
  if(clk='1' AND clk'event) then
     if (clr='1') then
        q<='0';
        else
         q<=d;
        end if;
   end if;
 end process;
end dff_arch;
```

Activity 2: Write the VHDL code for universal shift register using D flip flop and 4-to-1 multiplexer as the components. Hint: Refer the block level diagram of the universal shift register.

```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;

entity Exp7_USR is
 port( clk,clr: in std_logic;
       d: in std_logic_vector(3 downto 0);
       si_rs,si_ls: in std_logic;
       sel: in std_logic_vector(1 downto 0);
       q: out std_logic_vector(3 downto 0));
end Exp7_USR;

architecture Behavioral of Exp7_USR is
--component declarations
component mux_4to1 is
    Port ( i0,i1,i2,i3 : in  STD_LOGIC;
           s : in  STD_LOGIC_VECTOR (1 downto 0);
           y : out  STD_LOGIC);
end component;
```

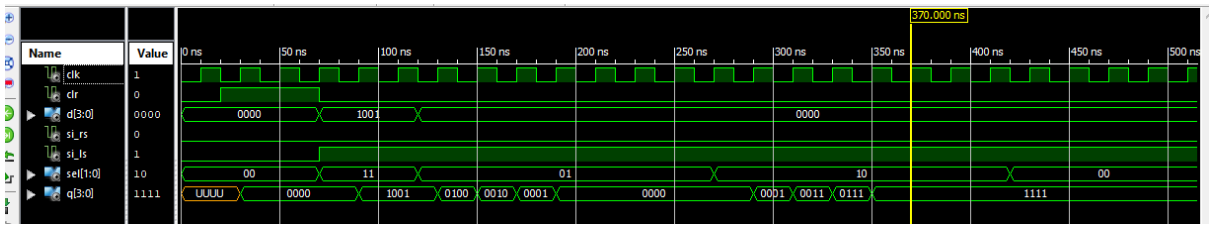
```
component Dff is
 port( D, clk,clr: in std_logic;
       q: out std_logic);
end component;

--- Signal declaration

signal mx: std_logic_vector(3 downto 0); --- mux outputs
signal qx: std_logic_vector(3 downto 0); --- FFs outputs

begin

DFF1: dff port map(mx(3),clk,clr,qx(3));
DFF2: dff port map(mx(2),clk,clr,qx(2));
DFF3: dff port map(mx(1),clk,clr,qx(1));
DFF4: dff port map(mx(0),clk,clr,qx(0));

Mux1: mux_4to1 port map(qx(3), si_rs,qx(2),d(3),sel,mx(3));
Mux2: mux_4to1 port map(qx(2), qx(3),qx(1),d(2),sel,mx(2));
Mux3: mux_4to1 port map(qx(1), qx(2),qx(0),d(1),sel,mx(1));
Mux4: mux_4to1 port map(qx(0), qx(1),si_ls,d(0),sel,mx(0));

--outputs
q<=qx;
end Behavioral;
```

Activity 3: Verify the operation of universal shift register using VHDL test bench.

```
LIBRARY ieee;
USE ieee.std_logic_1164.ALL;


ENTITY USR_tb IS
END USR_tb;

ARCHITECTURE behavior OF USR_tb IS

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

    COMPONENT Exp7_USR
    PORT(
         clk : IN  std_logic;
         clr : IN  std_logic;
         d : IN  std_logic_vector(3 downto 0);
         si_rs : IN  std_logic;
         si_ls : IN  std_logic;
         sel : IN  std_logic_vector(1 downto 0);
         q : OUT  std_logic_vector(3 downto 0)
        );
    END COMPONENT;
```

```
   --Inputs
   signal clk : std_logic := '0';
   signal clr : std_logic := '0';
   signal d : std_logic_vector(3 downto 0) := (others => '0');
   signal si_rs : std_logic := '0';
   signal si_ls : std_logic := '0';
   signal sel : std_logic_vector(1 downto 0) := (others => '0');

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

   -- Clock period definitions
   constant clk_period : time := 20 ns;

BEGIN

   -- Instantiate the Unit Under Test (UUT)
   uut: Exp7_USR PORT MAP (
          clk => clk,
          clr => clr,
          d => d,
          si_rs => si_rs,
          si_ls => si_ls,
          sel => sel,
          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;


   stim_proc: process
   begin
      -- hold reset state for 100 ns.
      wait for 20 ns;

      --  test inputs
      clr<= '1'; sel<="00"; wait for 50 ns;
      clr<= '0'; sel<="00";

      sel<="11";  d<="1001";  si_ls<='1'; si_rs<='0'; wait for 50 ns;
      sel<="01";  d<="0000";  si_ls<='1'; si_rs<='0'; wait for 150 ns;
      sel<="10";  d<="0000";  si_ls<='1'; si_rs<='0'; wait for 150 ns;
      sel<="00";  d<="0000";  si_ls<='1'; si_rs<='0'; wait for 50 ns;

      wait;
   end process;

END;
```



Activity 4: Perform the synthesis of universal shift register design.

Note: Set the FPGA device before the synthesis (FPGA Family- Spartan 3E, Device-XC3S100E).

Number of Slices	
Number of Slice FFs	
Number of LUTs	
IOBs	
Minimum Clock period	