Course Name:	**Hardware Description Language Lab (2UXL401)**	**Semester:**	**IV**
Date of Performance:	**07 / 04 / 2021**	**Batch No:**	**B2**
Faculty Name:	**Prof. Bhargavi Kaslikar**	**Roll No:**	**1912060**
Faculty Sign & Date:		**Grade/Marks:**	

Experiment No: 5

Title: Use of sequential statements:

a) Shift Registers using Flip flops
b) Shift Registers using IC 74194

Aim and Objective of the Experiment:

1. Write a VHDL code for implementing a 4-bit universal shift register with synchronous reset. It has 2 select inputs which control the operation as follows
   00: pause

   01: shift left

   10: shift right

   11: Rotate right

2. Write a VHDL code to implement IC 74194
3. Write a test bench to verify your results.
   Also, generate a programming file and download the code on CPLD kit and verify the results.

   Write a VHDL code

To study basic sequential statements of VHDL and to understand use of test bench for simulation.

COs to be achieved:

CO 1: Use basic Concurrent and Sequential statements in VHDL and write codes for simple applications

CO 2: Test a VHDL code and verify the circuit model.

CO 3: Synthesize and Implement the designed circuits on CPLD/ FPGA.

Work to be done

Upload VHDL codes for 4 bit universal shift register with synchronous reset. Also upload test bench and simulation for the same.

Upload scanned copy of post lab questions.

Shift Register Using JK FlipFlop:

```
library ieee;
use ieee.std_logic_1164.all;
use ieee.std_logic_unsigned.all;
use ieee.std_logic_arith.all;

entity shift_reg_ff_naik is
port(
        rst,clk : in std_logic;
        output : out std_logic_vector(3 downto 0)
        );
end shift_reg_ff_naik;

architecture shift_reg_ff_naik_arch of shift_reg_ff_naik is

        component jk_naik is
                port (
                        JK: IN STD_LOGIC_VECTOR(1 downto 0);
                        clock: IN STD_LOGIC;
                        reset: IN STD_LOGIC;
                        q: out STD_LOGIC
                );
        end component;
        signal temp_output:std_logic_vector(3 downto 0);
        signal temp1,temp2,temp3: std_logic_vector(1 downto 0);

        begin
                FF1 : jk_naik port map("11"  , clk , rst , temp_output(0));

                temp1 <= temp_output(0) & (not temp_output(0));
```

```
                        FF2 : jk_naik port map(temp1 , clk , rst , temp_output(1));

                        temp2 <= temp_output(1) & (not temp_output(1));
                        FF3 : jk_naik port map(temp2 , clk , rst , temp_output(2));

                        temp3 <= temp_output(2) & (not temp_output(2));
                        FF4 : jk_naik port map(temp3 , clk , rst , temp_output(3));
                        output <= temp_output;
end shift_reg_ff_naik_arch;
```

JK Flip Flop:

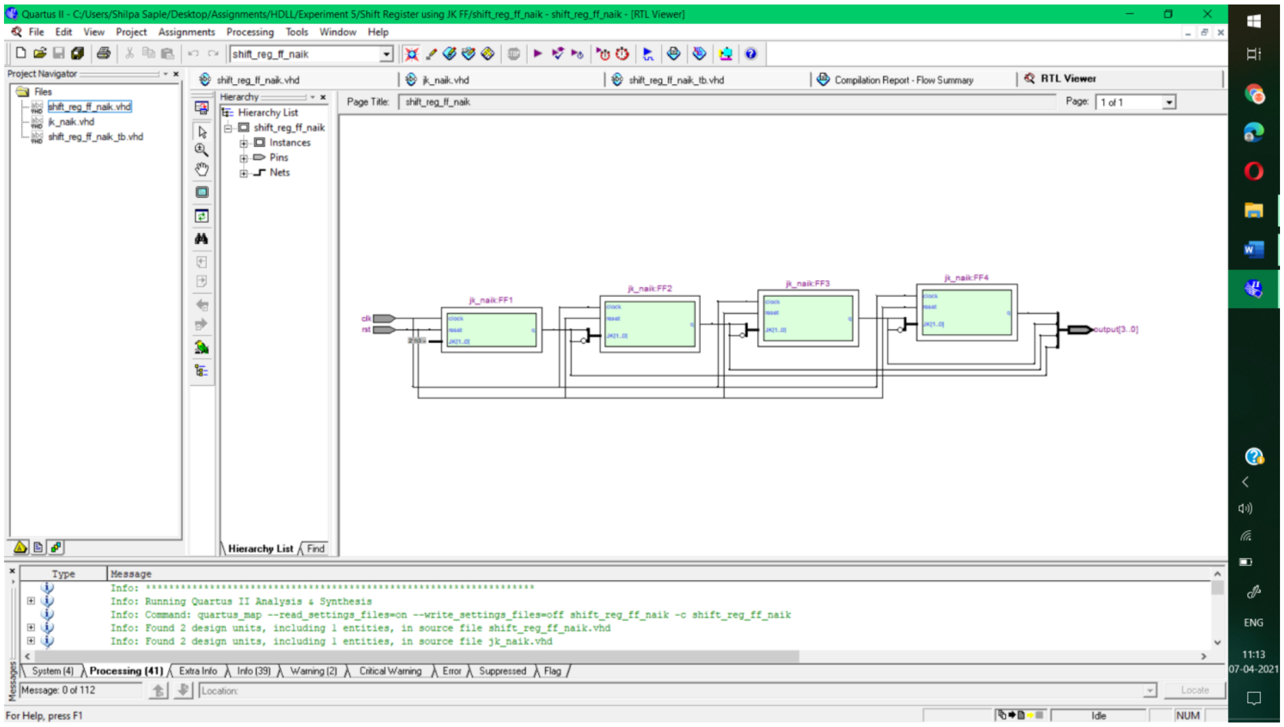
```
library ieee;
use ieee.std_logic_1164.all;
use ieee.std_logic_arith.all;
use ieee.std_logic_unsigned.all;

entity jk_naik is
            port (
                        JK: IN STD_LOGIC_VECTOR(1 downto 0);
                        clock: IN STD_LOGIC;
                        reset: IN STD_LOGIC;
                        q: out STD_LOGIC
            );
end jk_naik;

architecture jk_naik_arch of jk_naik is
            signal q_s  : std_logic := '0';
            begin
            process(reset,clock)
                        begin
                         if (reset = '1')then
                                    q_s <='0';
                         elsif (clock'event and clock = '1')then
                                    case (JK) is
                                                when "00" => q_s <= q_s;
                                                when "01" => q_s <= '0';
                                                when "10" => q_s <= '1';
                                                when others => q_s <= not q_s;
                                    end case;
```

```
                end if;
        end process;
        q <= q_s;
end jk_naik_arch;
```

Shift Register using JK FlipFlop Testbench:

```
library ieee;
use ieee.std_logic_1164.all;
use ieee.std_logic_unsigned.all;
use ieee.std_logic_arith.all;

entity shift_reg_ff_naik_tb is
end entity;
architecture shift_reg_ff_naik_tb_arch of shift_reg_ff_naik_tb is
        component shift_reg_ff_naik is
                port(
                        rst,clk : in std_logic;
                        output : out std_logic_vector(3 downto 0)
                );
        end component;
        signal rst,clk : std_logic;
        signal output : std_logic_vector(3 downto 0);
        begin
                SR : shift_reg_ff_naik port map(rst , clk , output);
                process begin
                        clk <= '1';
                        wait for 10ns;
                        clk <= '0';
                        wait for 10ns;
                end process;

                process
                        begin
                                rst <= '0';
                                wait for 2000ns;
                                rst <= '1';
                                wait for 20ns;
                end process;
end shift_reg_ff_naik_tb_arch;
```

RTL View:


Quartus II - C:/Users/Shilpa Saple/Desktop/Assignments/HDLL/Experiment 5/Shift Register using JK FF/shift_reg_ff_naik - shift_reg_ff_naik - [RTL Viewer]
File Edit View Project Assignments Processing Tools Window Help
shift_reg_ff_naik
Project Navigator
Files
shift_reg_ff_naik.vhd
jk_naik.vhd
shift_reg_ff_naik_tb.vhd
Hierarchy
Hierarchy List
shift_reg_ff_naik
Instances
Pins
Nets
Hierarchy List
Find
Compilation Report - Flow Summary
RTL Viewer
Page Title: shift_reg_ff_naik
Page: 1 of 1
jk_naik:FF1
jk_naik:FF2
jk_naik:FF3
jk_naik:FF4
clk
rst
output[3..0]

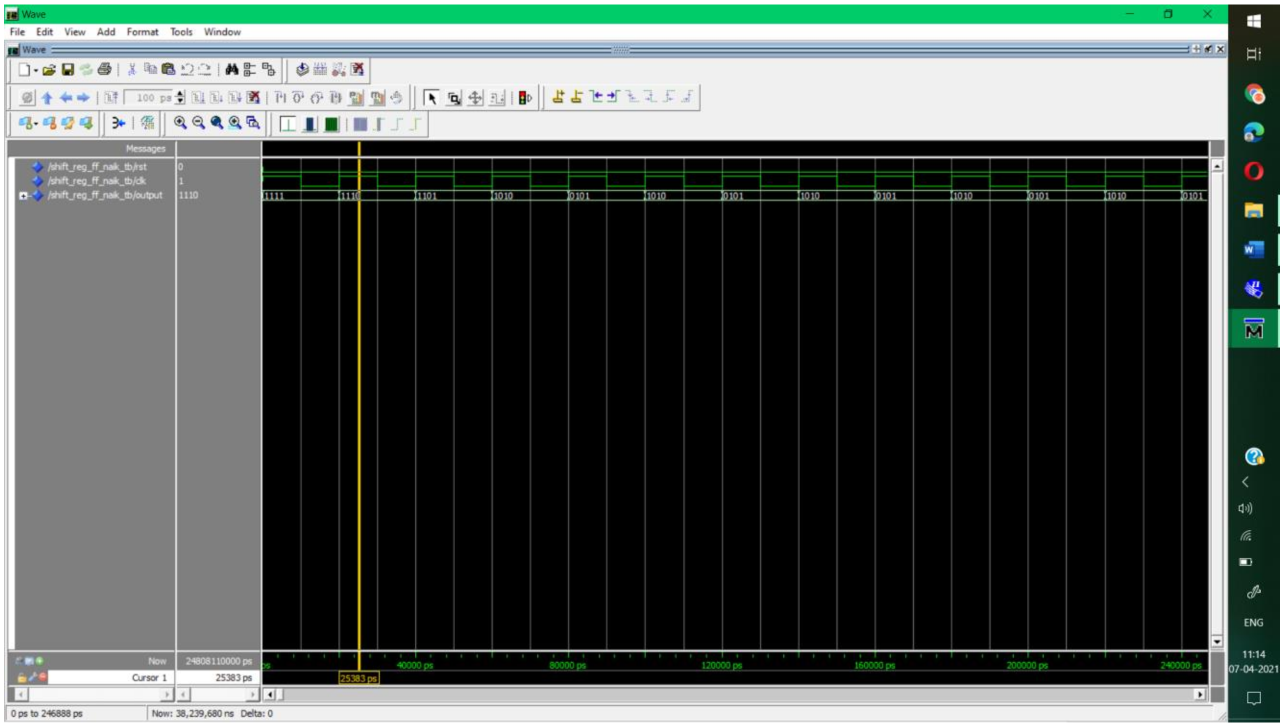
```
Type    Message
        Info: ************************************************************
        Info: Running Quartus II Analysis & Synthesis
        Info: Command: quartus_map --read_settings_files=on --write_settings_files=off shift_reg_ff_naik -c shift_reg_ff_naik
        Info: Found 2 design units, including 1 entities, in source file shift_reg_ff_naik.vhd
        Info: Found 2 design units, including 1 entities, in source file jk_naik.vhd
```

System (4) | Processing (41) | Extra Info | Info (39) | Warning (2) | Critical Warning | Error | Suppressed | Flag

Message: 0 of 112

For Help, press F1

Output Waveform:



```
/shift_reg_ff_naik_tb/rst      0
/shift_reg_ff_naik_tb/clk      1
/shift_reg_ff_naik_tb/output   1110    1111 1110 1101 1010 0101 1010 0101 1010 0101 1010 0101 1010 0101

Now      24808110000 ps
Cursor 1 25383 ps                25383 ps

0 ps to 246888 ps    Now: 38,239,680 ns  Delta: 0
```

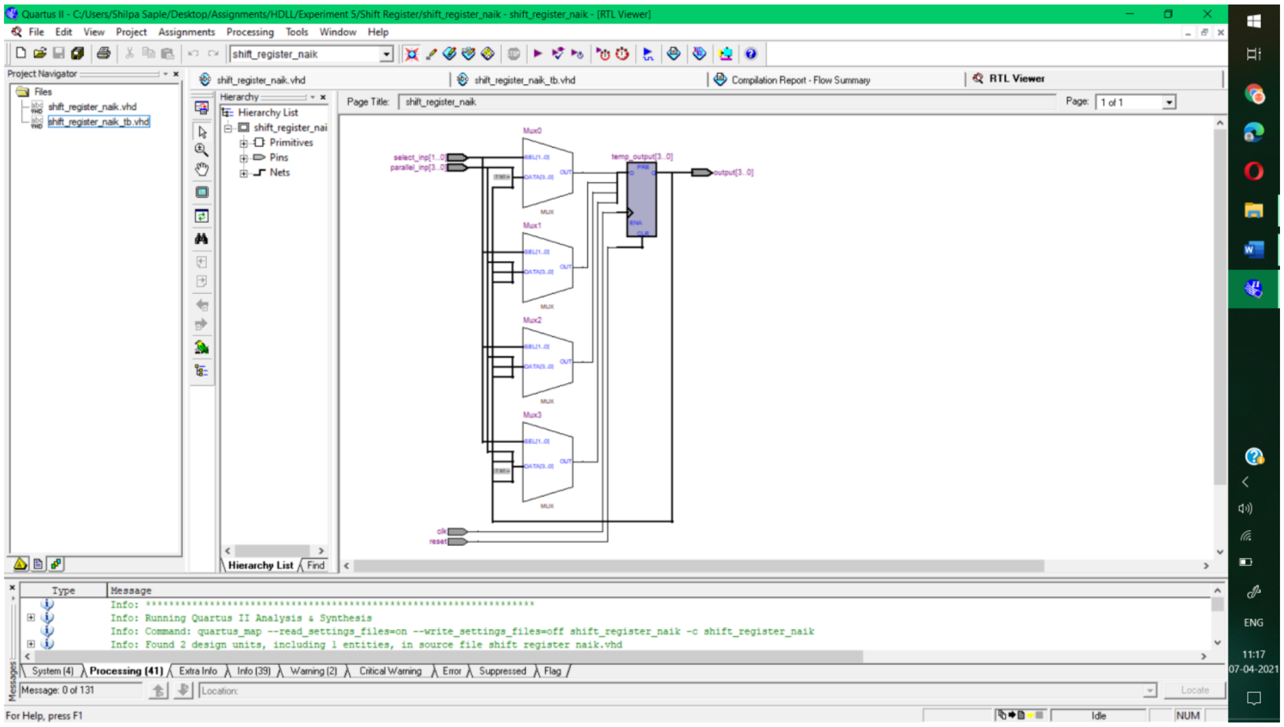
Shift Register

```
library ieee;
use ieee.std_logic_1164.all;
use ieee.std_logic_unsigned.all;
use ieee.std_logic_arith.all;

entity shift_register_naik is
  port ( clk, reset : in std_logic;
         select_inp : in std_logic_vector(1 downto 0);
         parallel_inp : in std_logic_vector(3 downto 0);
         output : out std_logic_vector(3 downto 0)
        );
end entity;

architecture shift_register_naik_arch of shift_register_naik is
signal temp_output : std_logic_vector(3 downto 0);
begin
 process(clk,reset,select_inp,parallel_inp)
   begin
            if(reset='1') then
                     temp_output <= "0000";
     elsif(clk'event and clk='1') then
       case select_inp is
          when "00" => temp_output <= temp_output;
          when "01" => temp_output <= temp_output(2 downto 0)&'1';
          when "10" => temp_output<= '1'&temp_output(3 downto 1);
          when "11" => temp_output <= parallel_inp;
          when others => temp_output <= "0000";
       end case;
     end if;
 end process;
 output <= temp_output;
end shift_register_naik_arch;
```

Shift Register Testbench:

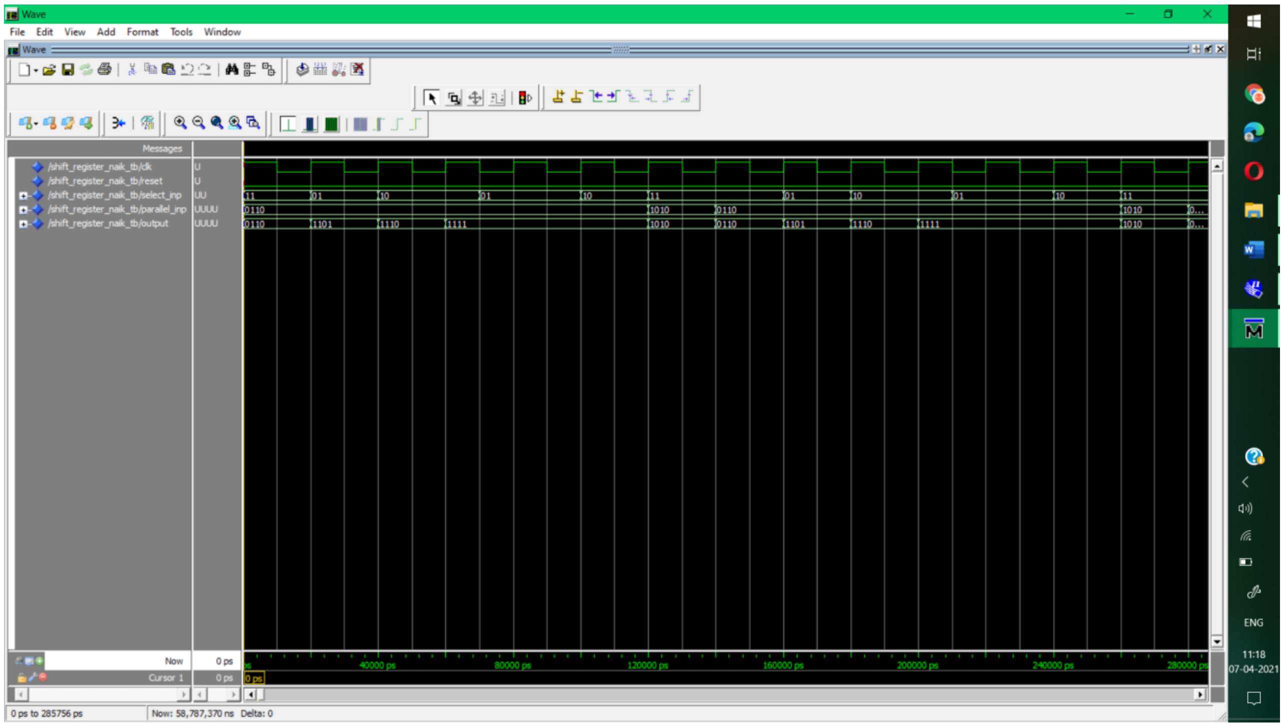
```
library ieee;
use ieee.std_logic_1164.all;
use ieee.std_logic_unsigned.all;
use ieee.std_logic_arith.all;

entity shift_register_naik_tb is
end entity;

architecture shift_register_naik_tb_arch of shift_register_naik_tb is
component shift_register_naik is
  port ( clk, reset : in std_logic;
         select_inp : in std_logic_vector(1 downto 0);
         parallel_inp : in std_logic_vector(3 downto 0);
         output : out std_logic_vector(3 downto 0)
        );
end component;

signal clk,reset : std_logic;
signal select_inp : std_logic_vector(1 downto 0);
signal parallel_inp : std_logic_vector(3 downto 0);
signal output : std_logic_vector(3 downto 0);

begin
            SR : shift_register_naik port map(clk , reset , select_inp , parallel_inp , output);

            process
                        begin
                                    clk <= '1';
                                    wait for 10ns;
                                    clk <= '0';
                                    wait for 10ns;
            end process;

            process
                        begin
                                    reset <= '0';
                                    wait for 2000ns;
                                    reset <= '1';
                                    wait for 20ns;
            end process;

            process
                        begin
                                    select_inp <= "11";
                                    parallel_inp <= "0110";
                                    wait for 20ns;

                                    select_inp <= "01";
                                    wait for 20ns;

                                    select_inp <= "10";
                                    wait for 30ns;

                                    select_inp <= "01";
                                    wait for 30ns;

                                    select_inp <= "10";
                                    wait for 20ns;

                                    select_inp <= "11";
                                    parallel_inp <= "1010";
                                    wait for 20ns;
            end process;
end shift_register_naik_tb_arch;
```

RTL View:


Quartus II - C:/Users/Shilpa Saple/Desktop/Assignments/HDLL/Experiment 5/Shift Register/shift_register_naik - shift_register_naik - [RTL Viewer]
File Edit View Project Assignments Processing Tools Window Help
shift_register_naik
Project Navigator
Files
shift_register_naik.vhd
shift_register_naik_tb.vhd
Hierarchy
Hierarchy List
shift_register_nai
Primitives
Pins
Nets
Hierarchy List
Find
shift_register_naik.vhd
shift_register_naik_tb.vhd
Compilation Report - Flow Summary
RTL Viewer
Page Title: shift_register_naik
Page: 1 of 1
Mux0
Mux1
Mux2
Mux3
MUX
select_inp[1..0]
parallel_inp[3..0]
temp_output[3..0]
output[3..0]
clk
reset

```
Type    Message
        Info: *******************************************************************
        Info: Running Quartus II Analysis & Synthesis
        Info: Command: quartus_map --read_settings_files=on --write_settings_files=off shift_register_naik -c shift_register_naik
        Info: Found 2 design units, including 1 entities, in source file shift register naik.vhd
```

Output Waveform:



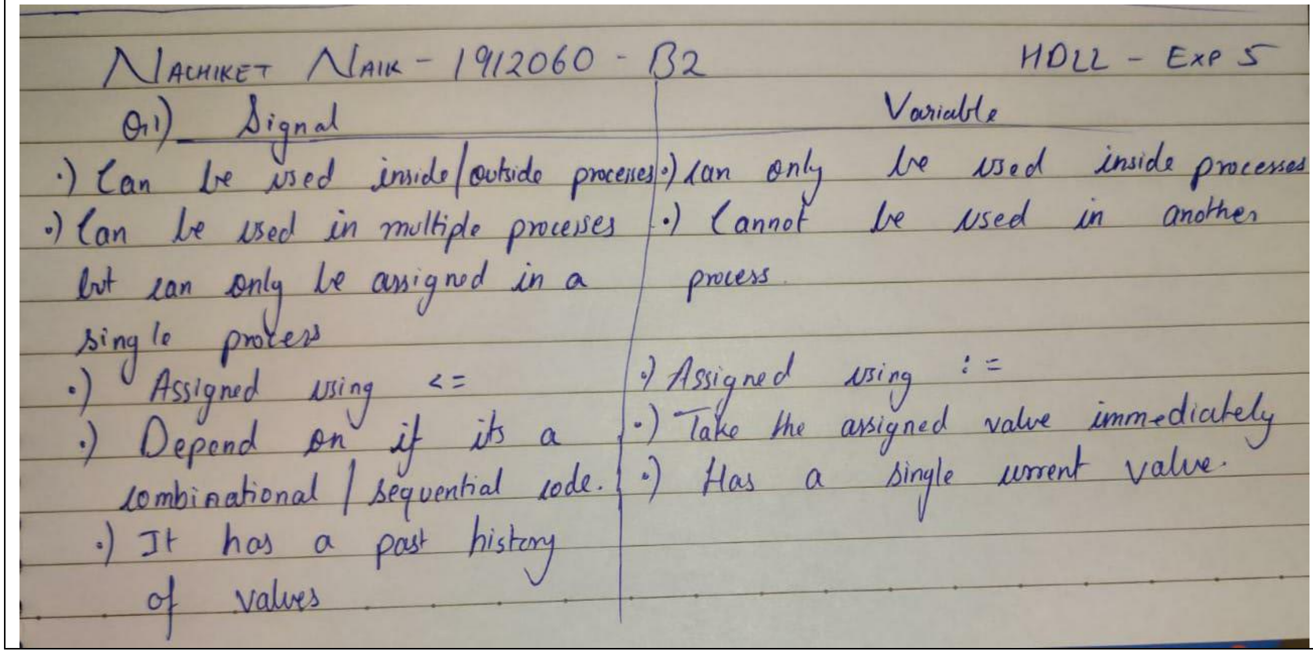
```
/shift_register_naik_tb/clk            U
/shift_register_naik_tb/reset          U
/shift_register_naik_tb/select_inp     UU     11    01    10    01    10    11    01    10    01    10    11
/shift_register_naik_tb/parallel_inp   UUUU   0110                          1010  0110                    1010  0...
/shift_register_naik_tb/output         UUUU   0110  1101  1110  1111        1010  0110  1101  1110  1111  1010  0...
```

```
0 ps to 285756 ps        Now: 58,787,370 ns  Delta: 0
```

Post Lab Subjective/Objective type Questions:

Upload Answer of following question before coming to next laboratory.

Q1 Differentiate the properties of signal and variable. Give suitable example



NACHIKET NAIK - 1912060 - B2 HDL2 - EXP 5

Q1)

Signal	Variable
•) Can be used inside/outside processes	•) Can only be used inside processes
•) Can be used in multiple processes but can only be assigned in a single process	•) Cannot be used in another process.
•) Assigned using <=	•) Assigned using :=
•) Depend on if its a combinational / sequential code.	•) Take the assigned value immediately
•) It has a past history of values	•) Has a single current value.

Conclusion:

In this experiment we have used structural architecture to make a left shift register using flip flop and process, case statements to make a bidirectional shift register.

The shift register takes a select input which decides the mode of operation of the shift register.

For 00 - hold the current value

For 01 - left shift the current bits and add a '1' to the right

For 10 - right shift the current bits and add a '1' to the left

For 11 - the register accepts the parallel load input and loads the output with the parallel input

If reset input is given high then the output is reset to "0000".

To shift the bits we use the concatenation operator : &

Signature of faculty in-charge with Date: