KATHMANDU UNIVERSITY

SCHOOL OF ENGINEERING

DEPARTMENT OF ELECTRICAL & ELECTRONICS ENGINEERING

LAB REPORT 04

EEEG-321

Department of Electrical and Electronics Engineering

By:

Samyam Shrestha (31056)

To:

Santosh Shaha Sir

Date:

14th May, 2024

Title: Write VHDL code for Shift / Rotate

Requirements: Vivado Software, Laptop

1) 1 bit shift/ Rotate

VHDL Code

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

entity shiftrotate_56 is
    Port ( D : in STD_LOGIC_vector(7 downto 0);
           sel : in STD_LOGIC_vector(1 downto 0);
           y : out STD_LOGIC_vector(7 downto 0));
end shiftrotate_56;

architecture Behavioral of shiftrotate_56 is

begin
process (sel, D)
begin
case sel is
when "00" => y <= d(7 downto 0);
when "01" => y <= d(7 downto 1) & '0';
when "10" => y <= '0' & d(6 downto 0);
when "11" => y <= d(6 downto 0) &d(7);
when others => y <= "00000000";
end case;
end process;
end Behavioral;
```

Test Bench code:

```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
entity shiftrotate_56_tb is
end shiftrotate_56_tb;

architecture Behavioral of shiftrotate_56_tb is
component shiftrotate_56
Port( D : in STD_LOGIC_vector(7 downto 0);
 sel : in STD_LOGIC_vector(1 downto 0);
   y : out STD_LOGIC_vector(7 downto 0));
 end component;
 signal D: STD_LOGIC_vector(7 downto 0);
 signal sel:  STD_LOGIC_vector(1 downto 0);
 signal y : STD_LOGIC_vector(7 downto 0);
begin
uut: shiftrotate_56 port map( D => D,
sel => sel,
```

```
y => y);
stimuli: process
begin
wait for 10ns;
D <= "10101011";
sel <= "00";
wait for 10ns;
sel <= "01";
wait for 10ns;
sel <= "10";
wait for 10ns;
sel <= "11";
wait for 10ns;
end process;
end Behavioral;
```

RTL Schematic Diagram:

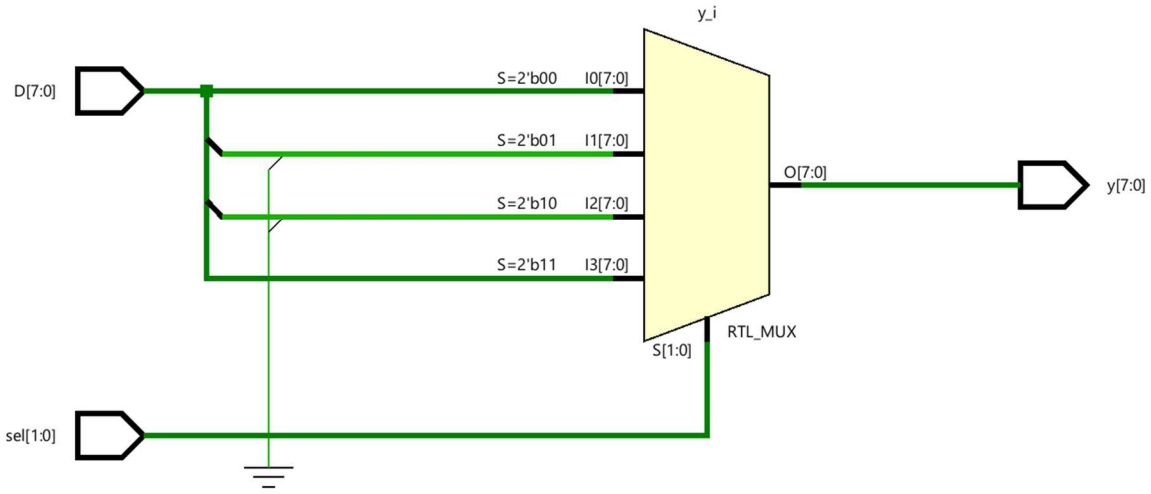

Figure 1: RTL Schematic of 1bit shit/rotate.

Simulation Model:

Here, two selection lines are given through sel variable. The output is shown in variable y.

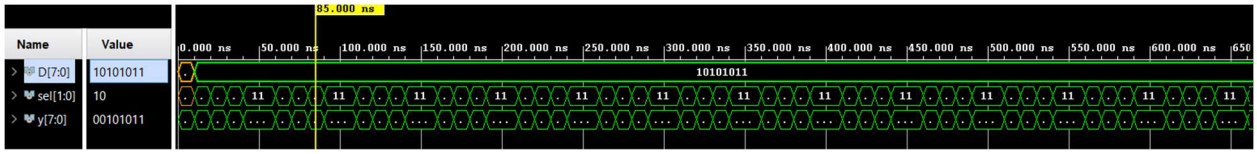

Figure 2: Simulation Model of 1bit shifter/ rotater

XDC File

```
set_property IOSTANDARD LVCMOS33 [get_ports {D[7]}]
set_property IOSTANDARD LVCMOS33 [get_ports {D[6]}]
set_property IOSTANDARD LVCMOS33 [get_ports {D[5]}]
set_property IOSTANDARD LVCMOS33 [get_ports {D[4]}]
set_property IOSTANDARD LVCMOS33 [get_ports {D[3]}]
set_property IOSTANDARD LVCMOS33 [get_ports {D[2]}]
set_property IOSTANDARD LVCMOS33 [get_ports {D[1]}]
set_property IOSTANDARD LVCMOS33 [get_ports {D[0]}]
set_property IOSTANDARD LVCMOS33 [get_ports {sel[1]}]
set_property IOSTANDARD LVCMOS33 [get_ports {sel[0]}]
set_property IOSTANDARD LVCMOS33 [get_ports {y[7]}]
set_property IOSTANDARD LVCMOS33 [get_ports {y[6]}]
set_property IOSTANDARD LVCMOS33 [get_ports {y[5]}]
set_property IOSTANDARD LVCMOS33 [get_ports {y[4]}]
set_property IOSTANDARD LVCMOS33 [get_ports {y[3]}]
set_property IOSTANDARD LVCMOS33 [get_ports {y[2]}]
set_property IOSTANDARD LVCMOS33 [get_ports {y[1]}]
set_property IOSTANDARD LVCMOS33 [get_ports {y[0]}]
set_property PACKAGE_PIN R13 [get_ports {D[7]}]
set_property PACKAGE_PIN U18 [get_ports {D[6]}]
set_property PACKAGE_PIN T18 [get_ports {D[5]}]
set_property PACKAGE_PIN R17 [get_ports {D[4]}]
set_property PACKAGE_PIN R15 [get_ports {D[3]}]
set_property PACKAGE_PIN M13 [get_ports {D[2]}]
set_property PACKAGE_PIN L16 [get_ports {D[1]}]
set_property PACKAGE_PIN J15 [get_ports {D[0]}]
set_property PACKAGE_PIN V10 [get_ports {sel[1]}]
set_property PACKAGE_PIN U11 [get_ports {sel[0]}]
set_property PACKAGE_PIN U16 [get_ports {y[7]}]
set_property PACKAGE_PIN V17 [get_ports {y[5]}]
set_property PACKAGE_PIN R18 [get_ports {y[4]}]
set_property PACKAGE_PIN N14 [get_ports {y[3]}]
set_property PACKAGE_PIN J13 [get_ports {y[2]}]
set_property PACKAGE_PIN K15 [get_ports {y[1]}]
set_property PACKAGE_PIN H17 [get_ports {y[0]}]
```

Device Utilization Report:

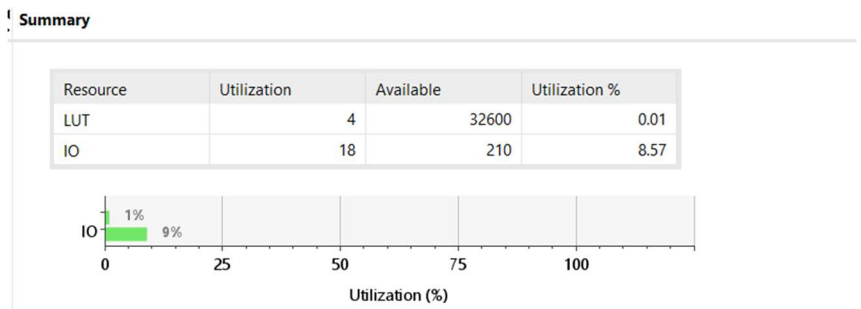

Figure 3: Device Utilization Report

Timing Analysis Report:

Design Timing Summary

Setup		Hold		Pulse Width	
Worst Negative Slack (WNS):	inf	Worst Hold Slack (WHS):	inf	Worst Pulse Width Slack (WPWS):	NA
Total Negative Slack (TNS):	0.000 ns	Total Hold Slack (THS):	0.000 ns	Total Pulse Width Negative Slack (TPWS):	NA
Number of Failing Endpoints:	0	Number of Failing Endpoints:	0	Number of Failing Endpoints:	NA
Total Number of Endpoints:	8	Total Number of Endpoints:	8	Total Number of Endpoints:	NA

There are no user specified timing constraints.

Figure 4: Timing Summary

2) 2 bit shifter/rotator

VHDL Code

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

entity shiftrotate_56 is
   Port ( D : in STD_LOGIC_VECTOR(7 downto 0);
          sel : in STD_LOGIC_VECTOR(2 downto 0);
          y : out STD_LOGIC_VECTOR(7 downto 0));
end shiftrotate_56;

architecture Behavioral of shiftrotate_56 is
begin
   process (sel, D)
   begin
      case sel is
         when "000" => y <= D; -- No shift
         when "001" => y <= D(6 downto 0) & D(7); -- Right shift by 1
         when "010" => y <= D(5 downto 0) & D(7 downto 6); -- Right shift by 2
         when "011" => y <= D(4 downto 0) & D(7 downto 5); -- Right shift by 3
         when "100" => y <= D(3 downto 0) & D(7 downto 4); -- Right shift by 4
         when "101" => y <= D(2 downto 0) & D(7 downto 3); -- Right shift by 5
         when "110" => y <= D(1 downto 0) & D(7 downto 2); -- Right shift by 6
         when "111" => y <= D(0) & D(7 downto 1); -- Right shift by 7
         when others => y <= (others => '0');
      end case;
   end process;
end Behavioral;
```

Test Bench Code

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

entity shiftrotate_56_tb is
end shiftrotate_56_tb;
```

```
architecture Behavioral of shiftrotate_56_tb is
    -- Component declaration for the unit under test
    component shiftrotate_56
        Port ( D : in STD_LOGIC_VECTOR(7 downto 0);
               sel : in STD_LOGIC_VECTOR(2 downto 0);
               y : out STD_LOGIC_VECTOR(7 downto 0));
    end component;

    -- Inputs
    signal D : STD_LOGIC_VECTOR(7 downto 0) := (others => '0');
    signal sel : STD_LOGIC_VECTOR(2 downto 0) := "000";

    -- Outputs
    signal y : STD_LOGIC_VECTOR(7 downto 0);

begin
    -- Instantiate the Unit Under Test (UUT)
    uut : shiftrotate_56 port map (
        D => D,
        sel => sel,
        y => y
    );

    -- Stimulus process
    stimuli: process
    begin
        -- Test case 1: No shift
        D <= "10101010";
        sel <= "000";
        wait for 10 ns;

        -- Test case 2: Right shift by 1
        sel <= "001";
        wait for 10 ns;

        -- Test case 3: Right shift by 2
        sel <= "010";
        wait for 10 ns;

        -- Test case 4: Right shift by 3
        sel <= "011";
        wait for 10 ns;

        -- Test case 5: Right shift by 4
        sel <= "100";
        wait for 10 ns;

        -- Test case 6: Right shift by 5
        sel <= "101";
```

```
        wait for 10 ns;

        -- Test case 7: Right shift by 6
        sel <= "110";
        wait for 10 ns;

        -- Test case 8: Right shift by 7
        sel <= "111";
        wait for 10 ns;

        wait;
    end process;
end Behavioral;
```

RTL Schematic

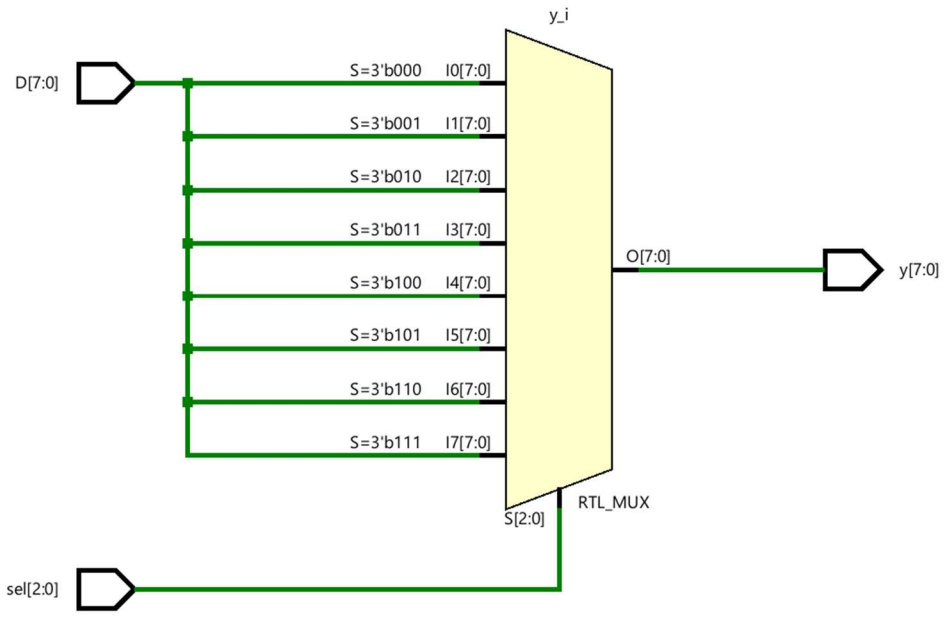

Figure 5: RTL Schematic of 2bit shifter/rotator

XDC File

```
set_property IOSTANDARD LVCMOS33 [get_ports {D[7]}]
set_property IOSTANDARD LVCMOS33 [get_ports {D[6]}]
set_property IOSTANDARD LVCMOS33 [get_ports {D[5]}]
set_property IOSTANDARD LVCMOS33 [get_ports {D[4]}]
set_property IOSTANDARD LVCMOS33 [get_ports {D[3]}]
set_property IOSTANDARD LVCMOS33 [get_ports {D[2]}]
set_property IOSTANDARD LVCMOS33 [get_ports {D[1]}]
set_property IOSTANDARD LVCMOS33 [get_ports {D[0]}]
set_property IOSTANDARD LVCMOS33 [get_ports {sel[2]}]
set_property IOSTANDARD LVCMOS33 [get_ports {sel[1]}]
set_property IOSTANDARD LVCMOS33 [get_ports {sel[0]}]
```

set_property IOSTANDARD LVCMOS33 [get_ports {y[7]}]
set_property IOSTANDARD LVCMOS33 [get_ports {y[6]}]
set_property IOSTANDARD LVCMOS33 [get_ports {y[5]}]
set_property IOSTANDARD LVCMOS33 [get_ports {y[4]}]
set_property IOSTANDARD LVCMOS33 [get_ports {y[3]}]
set_property IOSTANDARD LVCMOS33 [get_ports {y[2]}]
set_property IOSTANDARD LVCMOS33 [get_ports {y[1]}]
set_property IOSTANDARD LVCMOS33 [get_ports {y[0]}]
set_property PACKAGE_PIN R13 [get_ports {D[7]}]
set_property PACKAGE_PIN U18 [get_ports {D[6]}]
set_property PACKAGE_PIN T18 [get_ports {D[5]}]
set_property PACKAGE_PIN R17 [get_ports {D[4]}]
set_property PACKAGE_PIN R15 [get_ports {D[3]}]
set_property PACKAGE_PIN M13 [get_ports {D[2]}]
set_property PACKAGE_PIN L16 [get_ports {D[1]}]
set_property PACKAGE_PIN J15 [get_ports {D[0]}]
set_property PACKAGE_PIN V10 [get_ports {sel[2]}]
set_property PACKAGE_PIN U11 [get_ports {sel[1]}]
set_property PACKAGE_PIN U16 [get_ports {y[7]}]
set_property PACKAGE_PIN U17 [get_ports {y[6]}]
set_property PACKAGE_PIN V17 [get_ports {y[5]}]
set_property PACKAGE_PIN R18 [get_ports {y[4]}]
set_property PACKAGE_PIN N14 [get_ports {y[3]}]
set_property PACKAGE_PIN J13 [get_ports {y[2]}]
set_property PACKAGE_PIN K15 [get_ports {y[1]}]
set_property PACKAGE_PIN H17 [get_ports {y[0]}]

Simulation

The timing waveform graph for different possible combinations of input is given in the figure below. In a test-bench code, the different input combination was given and also the wait instruction for 10ns. So, while simulating, all the output can be seen in a single figure.

Figure 6: Timing Diagram for 2bit shifter/rotator

Device Utilization Report

Summary

Resource	Utilization	Available	Utilization %
LUT	12	32600	0.04
IO	19	210	9.05

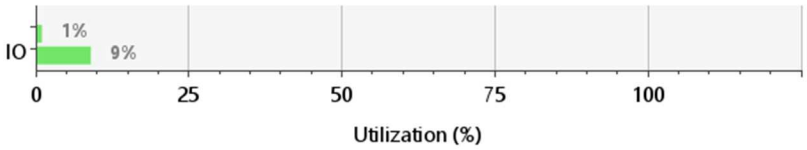

Figure 7: Device Utilization Report

Time Analysis Report

Design Timing Summary

Setup		Hold		Pulse Width	
Worst Negative Slack (WNS):	inf	Worst Hold Slack (WHS):	inf	Worst Pulse Width Slack (WPWS):	NA
Total Negative Slack (TNS):	0.000 ns	Total Hold Slack (THS):	0.000 ns	Total Pulse Width Negative Slack (TPWS):	NA
Number of Failing Endpoints:	0	Number of Failing Endpoints:	0	Number of Failing Endpoints:	NA
Total Number of Endpoints:	8	Total Number of Endpoints:	8	Total Number of Endpoints:	NA

There are no user specified timing constraints.

Figure 8: Timing Summary for Full Adder

Conclusion

In this way, one bit and two bit shifter and rotator was designed in VIVADO software in VHDL language. In VHDL file, under architecture, between begin and end, the obtained expression was assigned to the respective outputs. But before that, selection lines were assigned. Then, another file was added under “Add Sources” that contained the test bench file. So, the test bench file was created. In the test bench file, all the possible combinations for input was given with each input, a delay of 10ns. Then synthesis along with the implementation was done. After that, simulation was done and the result was obtained in the form of graph. Also, schematic from the RTL Analysis was taken. Then, the device utility report and time analysis report was obtained. Here in 2 bit selection, “00” was for passing the input, “01 was for shift left”, “10” for shift right and “11” was for rotating the bit left.