Assignment 5
HDL

Name:Akshat Saxena

Roll:S20200020243

Q1.4 bit right shift register using behavioral modeling non blocking statement

Test Operator code:

```verilog
// 4- bit right shift register using behavioral modeling
// a

module right_shift( in,out,reset,clk);
    input in;
    input clk;
    output reg [3:0] out;
    input reset;
    always @(posedge clk)
    begin
        if(reset)
            out <= 4'b0000;
        else
            out <= {out[2:0],in};
    end


endmodule
```

Q2.4 bit right shift register using behavioral modeling blocking statement

```verilog
// 4- bit right shift register using behavioral modeling
// a
```

```verilog
module right_shift( in,out,reset,clk);
    input in;
    input clk;
    output reg [3:0] out;
    input reset;
    always @(posedge clk)
    begin
        if(reset)
            out = 4'b0000;
        else
            out = {out[2:0],in};
    end


endmodule
```

Testbench code:

```verilog
// Test bench
`timescale 1ns/1ps
`include "right_shift.v"

module right_shift_tb;
    // Inputs
    reg in;
    reg reset;
    reg clk;
    // Outputs
    wire [3:0] out;
    // Instantiate the Unit Under Test (UUT)
    right_shift uut (
        .in(in),
        .out(out),
        .clk(clk),
        .reset(reset)
    );
    initial begin
        // Initialize Inputs
```

```verilog
    $dumpfile("right_shift_tb.vcd");
    $dumpvars(0,right_shift_tb);
        in = 1;
        reset = 1;
        clk = 0;
        // Add stimulus here
         clk = ~clk;
         #20;
        $display("out = %b",out);
        reset=0;
         clk = ~clk;
         #20;
        $display("out = %b",out);
         clk = ~clk;
         #20;
        $display("out = %b",out);
         clk = ~clk;
         #20;
        $display("out = %b",out);
         clk = ~clk;
         #20;
        $display("out = %b",out);
         clk = ~clk;
         #20;
        $display("out = %b",out);
         clk = ~clk;
         #20;
        $display("out = %b",out);
         clk = ~clk;
         #20;
        $display("out = %b",out);
         clk = ~clk;
         #20;
        $display("out = %b",out);




        #100 $finish;

    end

endmodule
```

```
(base) akshatsaxena@GOURIs-MacBook-Air assign5 % iverilog -o right_shift.vvp right_shift_tb.v
(base) akshatsaxena@GOURIs-MacBook-Air assign5 % vvp right_shift.vvp
VCD info: dumpfile right_shift_tb.vcd opened for output.
out = 0000
out = 0000
out = 0001
out = 0001
out = 0011
out = 0011
out = 0111
out = 0111
out = 1111
out = 1111
```