**Code:**

`timescale 1ns / 1ps

//////////////////////////////////////////////////////////////////////////////////

// Company:

// Engineer:

//

// Create Date: 11:24:52 03/14/2017

// Design Name:

// Module Name: rshift\_4

// Project Name:

// Target Devices:

// Tool versions:

// Description:

//

// Dependencies:

//

// Revision:

// Revision 0.01 - File Created

// Additional Comments:

//

//////////////////////////////////////////////////////////////////////////////////

module mux2x1(

input wire in0,

input wire in1,

input wire sel,

output reg m\_out

);

always@(\*) begin

if(sel)

m\_out = in1;

else

m\_out = in0;

end

endmodule

module rshift\_4(

input wire [3:0]data,

input wire [1:0]sel,

output wire [3:0]out,

//output reg [3:0]rshift\_4\_out,

wire y0,y1,y2,y3,

wire zero = 0

);

mux2x1 m10(.in0(data[0]),.in1(data[1]),.sel(sel[0]),.m\_out(y0));

mux2x1 m11(.in0(data[1]),.in1(data[2]),.sel(sel[0]),.m\_out(y1));

mux2x1 m12(.in0(data[2]),.in1(data[3]),.sel(sel[0]),.m\_out(y2));

mux2x1 m13(.in0(data[3]),.in1(zero),.sel(sel[0]),.m\_out(y3));

mux2x1 m20(.in0(y0),.in1(y2),.sel(sel[1]),.m\_out(out[0]));

mux2x1 m21(.in0(y1),.in1(y3),.sel(sel[1]),.m\_out(out[1]));

mux2x1 m22(.in0(y2),.in1(zero),.sel(sel[1]),.m\_out(out[2]));

mux2x1 m23(.in0(y3),.in1(zero),.sel(sel[1]),.m\_out(out[3]));

endmodule

**Testbench:**

`timescale 1ns / 1ps

////////////////////////////////////////////////////////////////////////////////

// Company:

// Engineer:

//

// Create Date: 11:47:45 03/14/2017

// Design Name: rshift\_4

// Module Name: F:/BTECH/Mini Project/Barell shifter/ISE/4 bit right shifter/rshift\_4/t\_rshift\_4.v

// Project Name: rshift\_4

// Target Device:

// Tool versions:

// Description:

//

// Verilog Test Fixture created by ISE for module: rshift\_4

//

// Dependencies:

//

// Revision:

// Revision 0.01 - File Created

// Additional Comments:

//

////////////////////////////////////////////////////////////////////////////////

module t\_rshift\_4;

// Inputs

reg [3:0] data;

reg [1:0] sel;

// Outputs

wire [3:0] out;

// Instantiate the Unit Under Test (UUT)

rshift\_4 uut (

.data(data),

.sel(sel),

.out(out)

);

initial begin

// Initialize Inputs

data = 0;

sel = 0;

// Wait 100 ns for global reset to finish

#100;

// Add stimulus here

data = 1000;

sel = 01;

$monitor("data[0] = %d \tdata[1] = %d \tdata[2] = %d \tdata[3] = %d \tsel[0] = %d \tsel[1] = %d, \tout[0]= %d \tout[1] =%d \tout[2] = %d \tout[3] =%d",data[0],data[1],data[2],data[3],sel[0],sel[1],out[0],out[1],out[2],out[3]);

#50;

sel = 10;

#50;

data = 1000;

sel=01;

#50;

sel=11;

end

endmodule