**Digital System Design Lab**

**CEL-442**

**Lab Journal no: 10**



Name: Omar

Class: BCE-6A

# Enrollment no: 01-132182-024

**LAB NO. 10**

**TITLE:**

**Basic Building Block Optimization – Barrel Shifter design**

**OBJECTIVE:**

The main objective of this lab is to design a barrel shifter, which is a basic building block in any digital system and used to optimize the design.

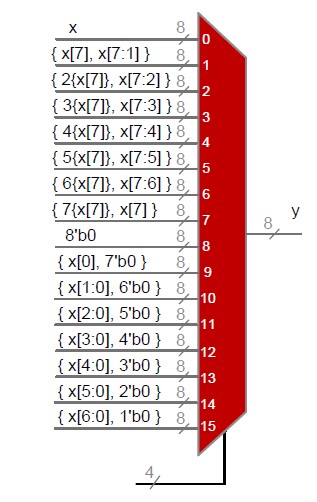
**Introduction:**

A barrel shifter is a [digital circuit](http://en.wikipedia.org/wiki/Digital_circuit) that can [shift](http://en.wikipedia.org/wiki/Bit_shift) a [data word](http://en.wikipedia.org/wiki/Word_(data_type)) by a specified number of [bits](http://en.wikipedia.org/wiki/Bit) in one [clock cycle.](http://en.wikipedia.org/wiki/Clock_cycle) It can be implemented as a sequence of [multiplexers](http://en.wikipedia.org/wiki/Multiplexer) (mux), and in such an implementation the output of one mux is connected to the input of the next mux in a way that depends on the shift distance.

**Task 1:**

**TASK 01:**

Implement an 8-bit barrel shifter as shown in the figure using Verilog HDL.



**Code:**

**Module:**

module Shift,(input [7:0]x, input [3:0]sel, output [7:0]y); reg [7:0]y; always @(sel or x)

begin case (sel)

4'd0: y = x;

4'd1: y = {x[7],x[7:1]};

4'd2: y = {{2{x[7]}},x[7:2]};

4'd3: y = {{3{x[7]}},x[7:3]};

4'd4: y = {{4{x[7]}},x[7:4]};

4'd5: y = {{5{x[7]}},x[7:5]};

4'd6: y = {{6{x[7]}},x[7:6]};

4'd7: y = {{7{x[7]}},x[7]};

4'd8: y = 8'b0;

4'd9: y = {x[0],7'b0};

4'd10: y = {x[1:0],6'b0};

4'd11: y = {x[2:0],5'b0};

4'd12: y = {x[3:0],4'b0};

4'd13: y = {x[4:0],3'b0};

4'd14: y = {x[5:0],2'b0}; 4'd15: y = {x[6:0],1'b0};

end endmodule

**Test bench**

module Shifttb\_Testbench; reg [7:0] x; reg [3:0] select;

wire [7:0] y;

QuestionTwo mux (.X(x), .Select(select), .Y(y));

initial begin

$monitor("Select = %b | x = %b | y = %b",select,x,y); x = 8'b00001110;

select = 4'd0;

#10;

x = 8'b00001110;

select = 4'd1;

#10;

x = 8'b00001101;

select = 4'd2;

#10;

x = 8'b00001101;

select = 4'd3;

#10;

x = 8'b00001011;

select = 4'd4;

#10;

x = 8'b00001011; select = 4'd5;

#10;

x = 8'b00000111; select = 4'd6;

#10;

x = 8'b00000111; select = 4'd7;

#10;

x = 8'b00001111; select = 4'd9;

#10;

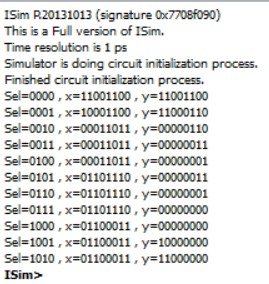
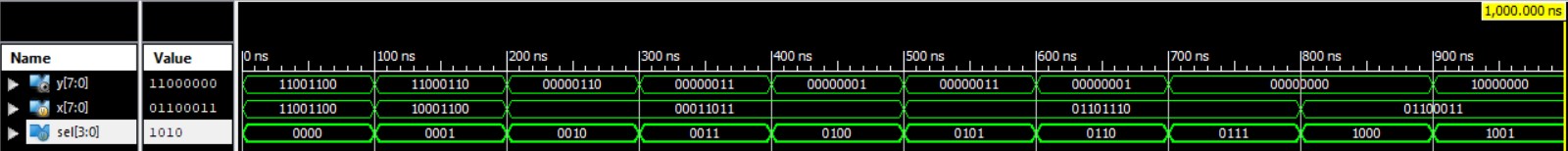
x = 8'b00001111;

select = 4'd10;

#10; end

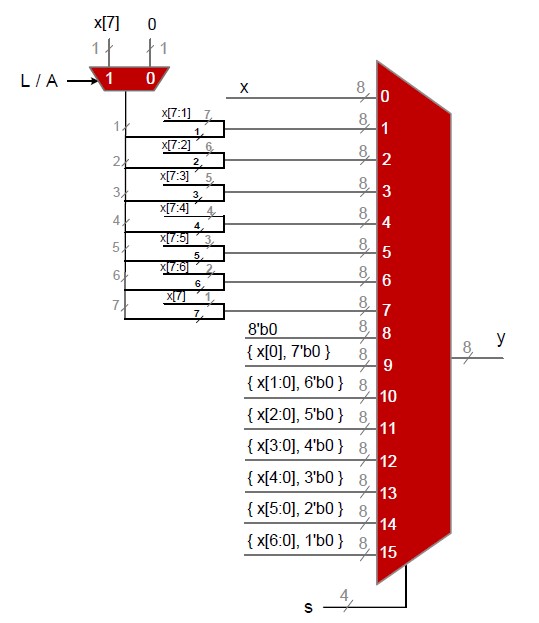
endmodule

**Output:**



**TASK 02:**

Implement an 8-bit barrel shifter as shown in the figure using Verilog HDL.



**Code:**

**Module:**

module Shifter(A,out); input [3:0]A; output reg [7:0]out; wire [7:0]w; assign w = 8'b11100111; always@(\*) begin if(A[3]==0) begin case(A[2:0])

3'b000: out = {w[0],w[7:1]};

3'b001: out = {w[1:0],w[7:2]};

3'b010: out = {w[2:0],w[7:3]};

3'b011: out = {w[3:0],w[7:4]};

3'b100: out = {w[4:0],w[7:5]};

3'b101: out = {w[5:0],w[7:6]};

3'b110: out = {w[6:0],w[7]};

3'b111: out = {w[7:0]};

default : out = 8'bxxxxxxxx;

endcase

end if(A[3]==1)

begin case(A[2:0]) 3'b000: out = {w[7:0]};

3'b001: out ={w[6:0],w[7]};

3'b010: out = {w[5:0],w[7:6]};

3'b011: out = {w[4:0],w[7:5]};

3'b100: out = {w[3:0],w[7:4]};

3'b101: out = {w[2:0],w[7:3]};

3'b110: out = {w[1:0],w[7:2]}; 3'b111: out = {w[0],w[7:1]};

default : out = 8'bxxxxxxxx; endcase end end

endmodule

**Test\_bench:**

module t3;

// Inputs reg [3:0] A;

// Outputs

wire [7:0] out;

// Instantiate the Unit Under Test (UUT)

Shifter uut (

.A(A),

.out(out)

);

initial begin // Initialize Inputs

A = 8'b00000001;

// Wait 100 ns for global reset to finish

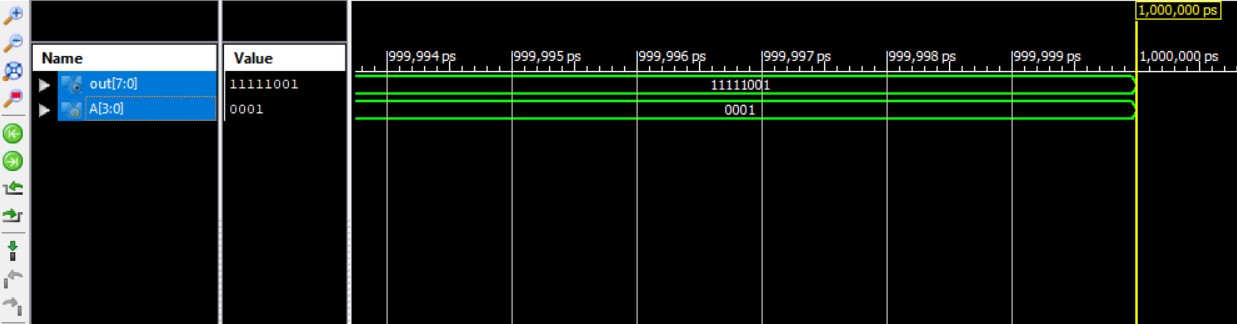
#100;

// Add stimulus here

end

endmodule

Output



**Conclusion:**

**In this lab we learned about how to shift bits using barrel shifters.**