ADARSH KUMAR CHAUDHARY

1701CS01

Codes

1……..

//// listing all inputs and outputs, by convention outputs go first

```
module p1(out, in);
input[3:0] in; // binary code
output [3:0] out; // gray code


assign out[0] = in[1] ^ in[0];
assign out[1] = in[2] ^ in[1];
assign out[2] = in[3] ^ in[2];
assign out[3] = in[3];


endmodule


// test bench
module tb_p1();
  wire [3:0] out1;
  reg [3:0] in1;
  integer i;
  p1 uut(out1, in1);
  initial
  begin
    for(i = 0;i <= 15;i = i+1) // varying in1
```

```
    begin
      in1 = i;
      #10
      $monitor("in = %b",in1," | out = %b", out1);
      $display("----------------------------");
    end
  end
endmodule
```

2……..

```
// listing all inputs and outputs, by convention outputs go first
module p2(Output,opcode,A,B);
        output[15:0] Output;
        input [15:0] A,B;
        input [2:0] opcode;
        reg [15:0] Output;
        always @ (opcode)
        begin
                case (opcode) //ALU implementation
                        3'd0: Output <= A+B;
                        3'd1: Output <= A-B;
                        3'd2: Output <= A&B;
                        3'd3: Output <= A^B;
                        3'd4: Output <= A|B;
```

```
                3'd5: Output <= A+1;
                3'd6: Output <= A<<1;
                3'd7: Output <= A>>1;
                default: Output <= 16'd0;
        endcase
    end
endmodule


module tb_p2(); //test_bench
    wire[15:0] Output;
    reg [15:0] A,B;
    reg [2:0] opcode;
    integer i,j;
    p2 UUT (Output, opcode, A, B);
    initial
    begin //checking for random values of A,B
    for( j = 0; j <= 3; j = i + 1)
    begin
        A=$urandom%2**15;
        B=$urandom%2**15;
        for( i = 0; i <= 7; i = i + 1)
        begin
            opcode=i;
            $monitor("Opcode = %d", i," | A = %d", A, " | B = %d", B, " | Output = %d",
Output);
```

```
                        #10;
                        $display("----------------------------------------");
                end
        end
        end
endmodule
```

3……

```
module p3(out1, in, sel);
output out1;
input[15:0] in;
input[3:0] sel;

reg out1;
integer i;
always @(in or sel)
case(sel)
  4'b0000:  out1 <= in[0];
  4'b0001:  out1 <= in[1];
  4'b0010:  out1 <= in[2];
  4'b0011:  out1 <= in[3];
  4'b0100:  out1 <= in[4];
  4'b0101:  out1 <= in[5];
  4'b0110:  out1 <= in[6];
```

```
  4'b0111:  out1 <= in[7];
  4'b1000:  out1 <= in[8];
  4'b1001:  out1 <= in[9];
  4'b1010:  out1 <= in[10];
  4'b1011:  out1 <= in[11];
  4'b1100:  out1 <= in[12];
  4'b1101:  out1 <= in[13];
  4'b1110:  out1 <= in[14];
  4'b1111:  out1 <= in[15];
endcase

endmodule

module tb_p3();
wire out1;
reg[15:0] in;
reg[3:0] sel;
integer i,j;

p3 uut(out1, in, sel);

initial
  begin
  in = 0;
  for(i = 0; i <=15; i= i+1)
```

```
        begin
        sel = i;
        for(j = 0; j <=1; j= j+1)
                begin
                in[i] = j;
                $monitor("in  = %b",in, " | sel = %b",sel, "| out = %b", out1);
                $display("------------------------");
                #10;
                end
        end
        end
endmodule
```

4……

```
module p4(out1, D, set);

output out1;
input [15:0] D;
input [3:0] set;

wire [7:0] t;
wire [3:0] t2;
wire [1:0] t3;
reg out1;

mux_2_to_1 m1 (t[7], D[15], D[14], set[0]);
```

```
mux_2_to_1 m2 (t[6], D[13], D[12], set[0]);

mux_2_to_1 m3 (t[5], D[11], D[10], set[0]);

mux_2_to_1 m4 (t[4], D[9], D[8], set[0]);

mux_2_to_1 m5 (t[3], D[7], D[6], set[0]);

mux_2_to_1 m6 (t[2], D[5], D[4], set[0]);

mux_2_to_1 m7 (t[1], D[3], D[2], set[0]);

mux_2_to_1 m8 (t[0], D[1], D[0], set[0]);


mux_2_to_1 m9  (t2[3], t[7], t[6], set[1]);

mux_2_to_1 m10 (t2[2], t[5], t[4], set[1]);

mux_2_to_1 m11 (t2[1], t[3], t[2], set[1]);

mux_2_to_1 m12 (t2[0], t[1], t[0], set[1]);


mux_2_to_1 m13 (t3[1], t2[3], t2[2], set[2]);

mux_2_to_1 m14 (t3[0], t2[1], t2[0], set[2]);


mux_2_to_1 m15 (out1, t3[1], t3[0], set[3]);


endmodule



module tb_p4();


reg [3:0] S;

reg [15:0] D;
```

```
wire out;
integer i,j;
p4 UUT(out, D, S);

initial
begin
	D = 16'b0;
	for (i = 0; i <= 15; i = i + 1)
	begin
		for(j=0; j<=1; j = j + 1)
		begin
			S = i;
			D[i] = j;
			#10
			$monitor("D = %b", D, " & S = %b", S);
		end
	end
end
endmodule
```

5……

```
module p5(O, I);

output [63:0] O;
input [5:0] I;
```

```
reg [63:0] O = 64'b0;


always @ (I)
begin
        O = 64'b0;
        O[I] = 1;
end
endmodule


module tb_p5();


wire[63:0] O;
reg[5:0] I;


p5 UUT(O, I);


integer i;


initial
begin
        for(i=0;i<=63;i=i+1)
        begin
                I=i;#10
                $monitor("O = %d", O, " & I = %b", I);
        end
```

```
end

endmodule
```

6……

```
module p6(data_out,data_in);

        output [63:0] data_out;

        input [5:0] data_in;

        decoder_3to8 maingate (a,b,c,d,e,f,g,h,data_in[3],data_in[4],data_in[5],1);

        decoder_3to8 decode1
(data_out[0],data_out[1],data_out[2],data_out[3],data_out[4],data_out[5],data_out[6],data_out[7],dat
a_in[0],data_in[1],data_in[2],a);

        decoder_3to8 decode2
(data_out[8],data_out[9],data_out[10],data_out[11],data_out[12],data_out[13],data_out[14],data_out[
15],data_in[0],data_in[1],data_in[2],b);

        decoder_3to8 decode3
(data_out[16],data_out[17],data_out[18],data_out[19],data_out[20],data_out[21],data_out[22],data_o
ut[23],data_in[0],data_in[1],data_in[2],c);

        decoder_3to8 decode4
(data_out[24],data_out[25],data_out[26],data_out[27],data_out[28],data_out[29],data_out[30],data_o
ut[31],data_in[0],data_in[1],data_in[2],d);

        decoder_3to8 decode5
(data_out[32],data_out[33],data_out[34],data_out[35],data_out[36],data_out[37],data_out[38],data_o
ut[39],data_in[0],data_in[1],data_in[2],e);

        decoder_3to8 decode6
(data_out[40],data_out[41],data_out[42],data_out[43],data_out[44],data_out[45],data_out[46],data_o
ut[47],data_in[0],data_in[1],data_in[2],f);

        decoder_3to8 decode7
(data_out[48],data_out[49],data_out[50],data_out[51],data_out[52],data_out[53],data_out[54],data_o
ut[55],data_in[0],data_in[1],data_in[2],g);

        decoder_3to8 decode8
(data_out[56],data_out[57],data_out[58],data_out[59],data_out[60],data_out[61],data_out[62],data_o
ut[63],data_in[0],data_in[1],data_in[2],h);

endmodule
```

```
//test_bench for binary to grey
module tb_p6();
	wire [63:0]out;
	reg [5:0]in;
	integer i;
	p6 UUT (out,in);
	initial
		begin
		for( i = 0; i <= 63; i = i + 1)
			begin
			in = i;#10
			$monitor("input_code = %b", in, "  |  output_code = %b", out);
			$display("----------------------------------------");
			end
		end
endmodule
```

7…….

```
module decade_counter(q, clk);
	output reg[3:0] q = 0;                    // telling the compiler which lines are inputs and outputs
	input clk;
	always @(posedge clk)
		q <= q == 9 ? 0 : q + 1;
endmodule
```

```
module decoded_counter(ctrl, clk);
	output ctrl;				// Tells the compiler which lines are inputs and outputs
	input clk;
	reg [3:0] count_value = 0;
	always @(posedge clk)
		count_value <= count_value + 1;
		assign ctrl = count_value == 4'b0111 || count_value == 4'b1011;
endmodule

module decade_counter(q, clk);
	output reg[3:0] q = 0;			// telling the compiler which lines are inputs and outputs
	input clk;
	always @(posedge clk)
		q <= q == 9 ? 0 : q + 1;
endmodule

module decoded_counter(ctrl, clk);
	output ctrl;				// Tells the compiler which lines are inputs and outputs
	input clk;
	reg [3:0] count_value = 0;
	always @(posedge clk)
		count_value <= count_value + 1;
		assign ctrl = count_value == 4'b0111 || count_value == 4'b1011;
endmodule
```

DECODER CODE

```
module decoder_3to8(Y0, Y1, Y2, Y3, Y4, Y5, Y6, Y7, C, B, A, en);
        output Y7, Y6, Y5, Y4, Y3, Y2, Y1, Y0;
        input A, B, C;
        input en;
        assign {Y7,Y6,Y5,Y4,Y3,Y2,Y1,Y0} = ( {en,A,B,C} == 4'b1000) ? 8'b0000_0001 :
        ( {en,A,B,C} == 4'b1001) ? 8'b0000_0010 :
        ( {en,A,B,C} == 4'b1010) ? 8'b0000_0100 :
        ( {en,A,B,C} == 4'b1011) ? 8'b0000_1000 :
        ( {en,A,B,C} == 4'b1100) ? 8'b0001_0000 :
        ( {en,A,B,C} == 4'b1101) ? 8'b0010_0000 :
        ( {en,A,B,C} == 4'b1110) ? 8'b0100_0000 :
        ( {en,A,B,C} == 4'b1111) ? 8'b1000_0000 :
        8'b0000_0000;
 endmodule
```

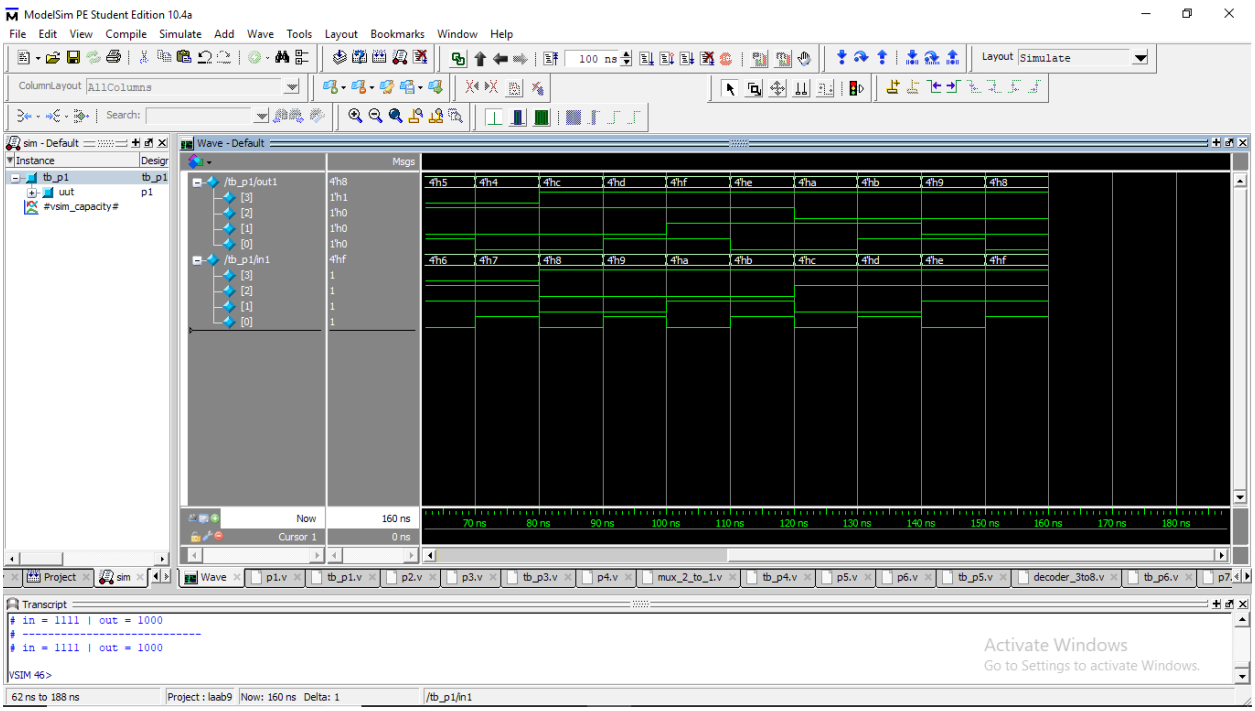
MULTIPLEXER CODE

```
module mux_2_to_1(out, a, b, s);

                input a;
                input b;
                input s;
                output reg out;
                always@(s or a or b)
                        case(s)
```

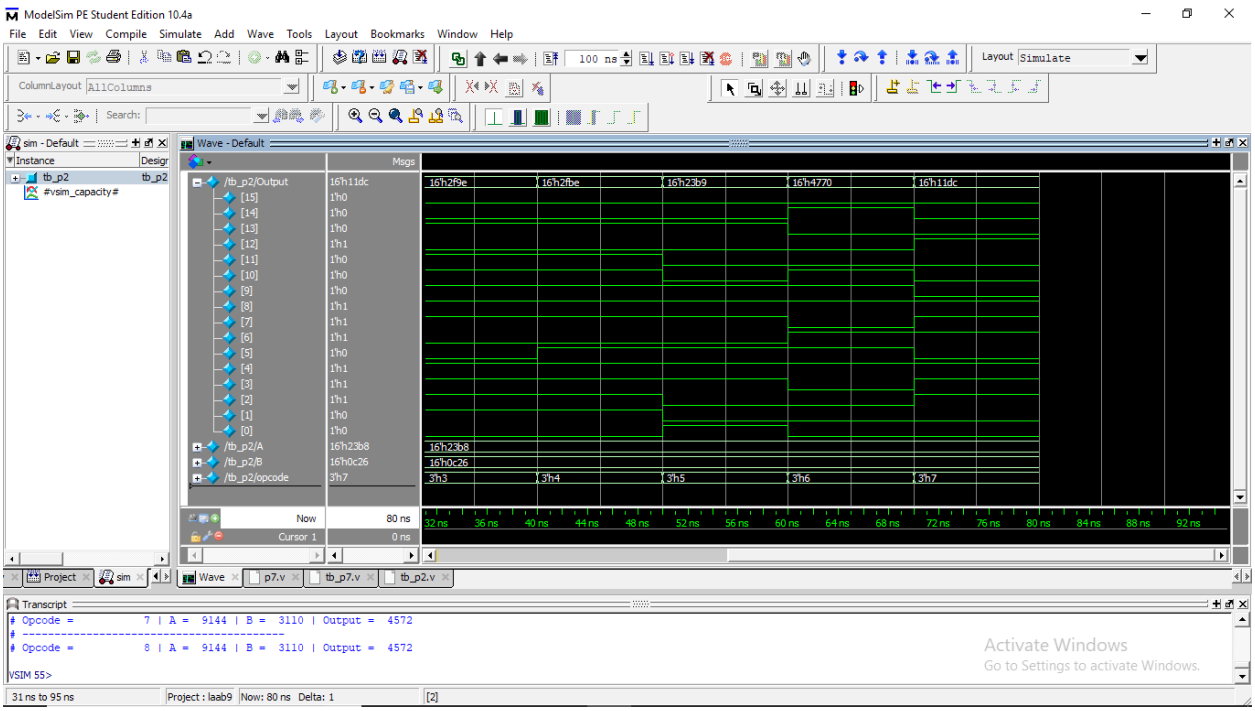
```
                    1'd0: out = a;

                    1'd1: out = b;

                endcase

endmodule
```

Screenshots

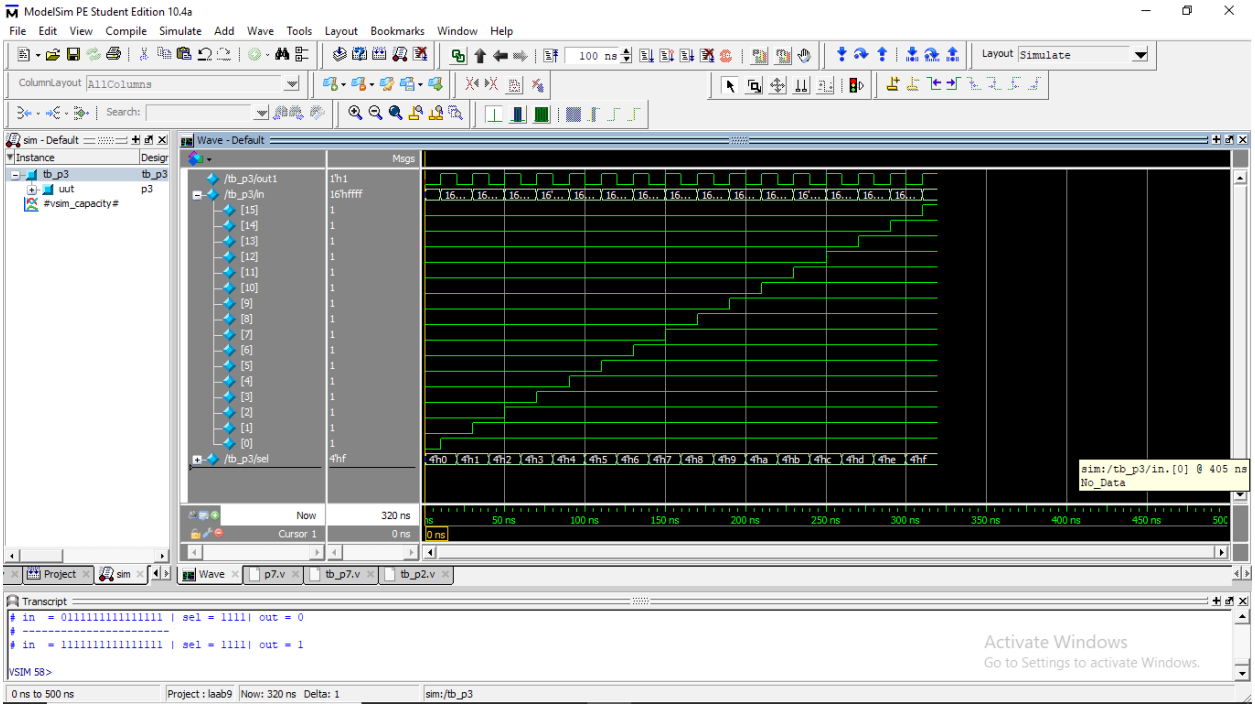
1.



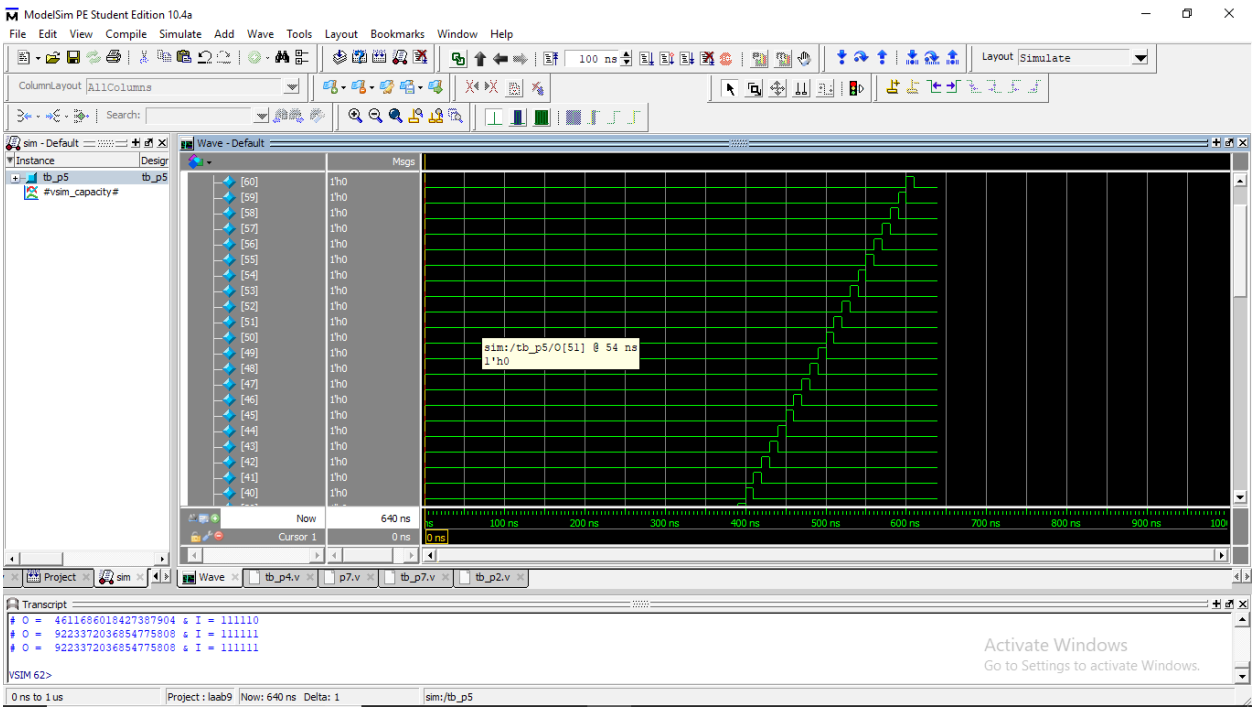
2.

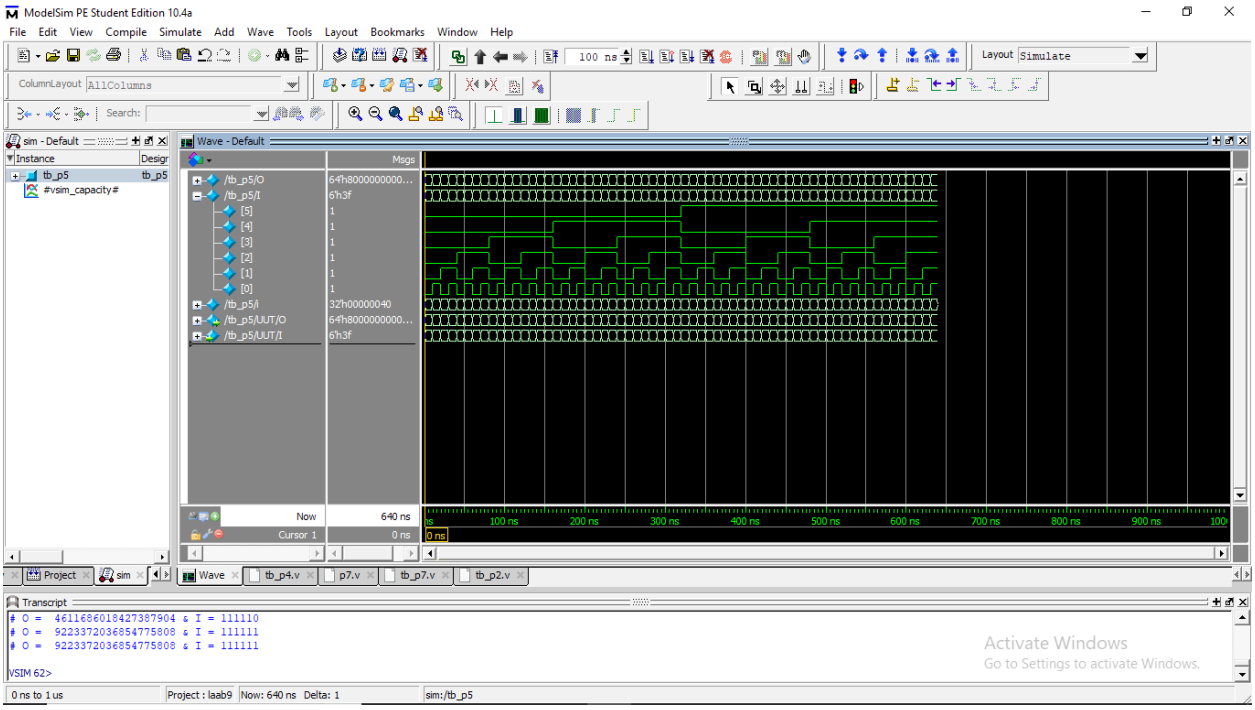

ModelSim PE Student Edition 10.4a
File Edit View Compile Simulate Add Wave Tools Layout Bookmarks Window Help
Wave - Default
/tb_p2/Output
/tb_p2/A
/tb_p2/B
/tb_p2/opcode
16'h2f9e
16'h2fbe
16'h23b9
16'h4770
16'h11dc
16'h23b8
16'h0c26
3'h3
3'h4
3'h5
3'h6
3'h7
Transcript
Opcode = 7 | A = 9144 | B = 3110 | Output = 4572
Opcode = 8 | A = 9144 | B = 3110 | Output = 4572
VSIM 55>

3.

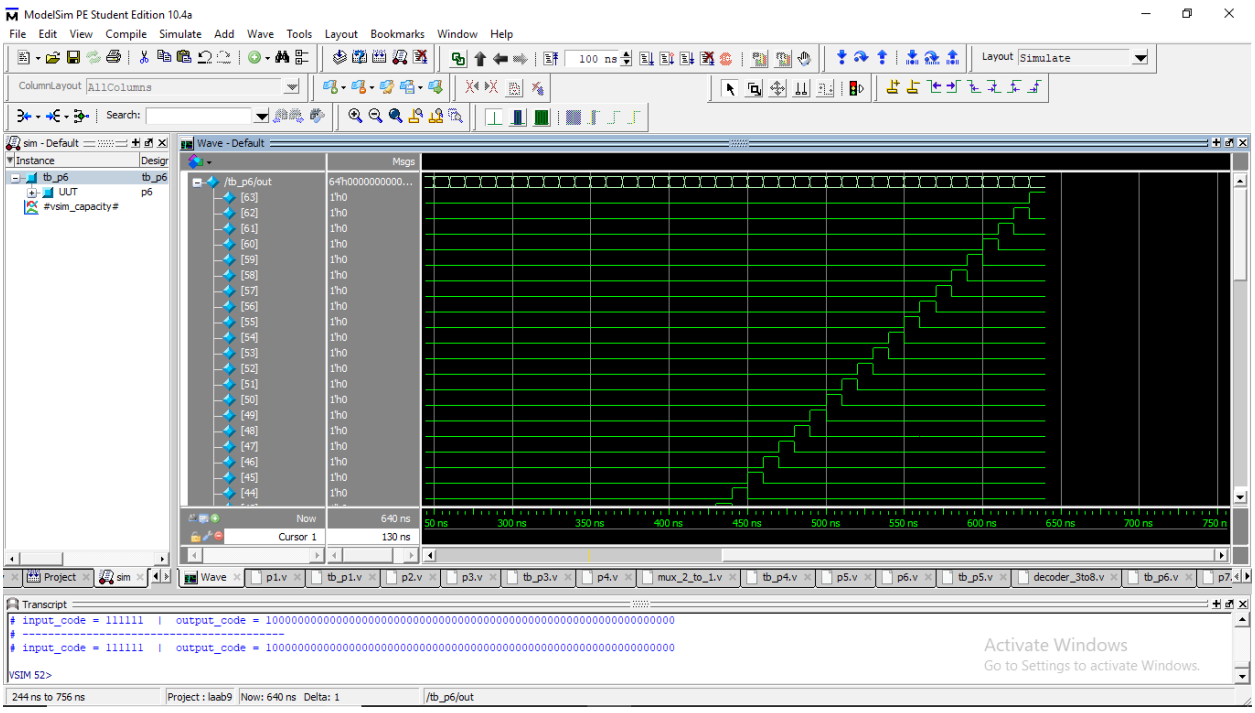

ModelSim PE Student Edition 10.4a
File Edit View Compile Simulate Add Wave Tools Layout Bookmarks Window Help
Wave - Default
/tb_p3/out1
/tb_p3/in
/tb_p3/sel
4'h0 4'h1 4'h2 4'h3 4'h4 4'h5 4'h6 4'h7 4'h8 4'h9 4'ha 4'hb 4'hc 4'hd 4'he 4'hf
sim:/tb_p3/in.[0] @ 405 ns
No_Data
Transcript
in = 0111111111111111 | sel = 1111| out = 0
in = 1111111111111111 | sel = 1111| out = 1
VSIM 58>

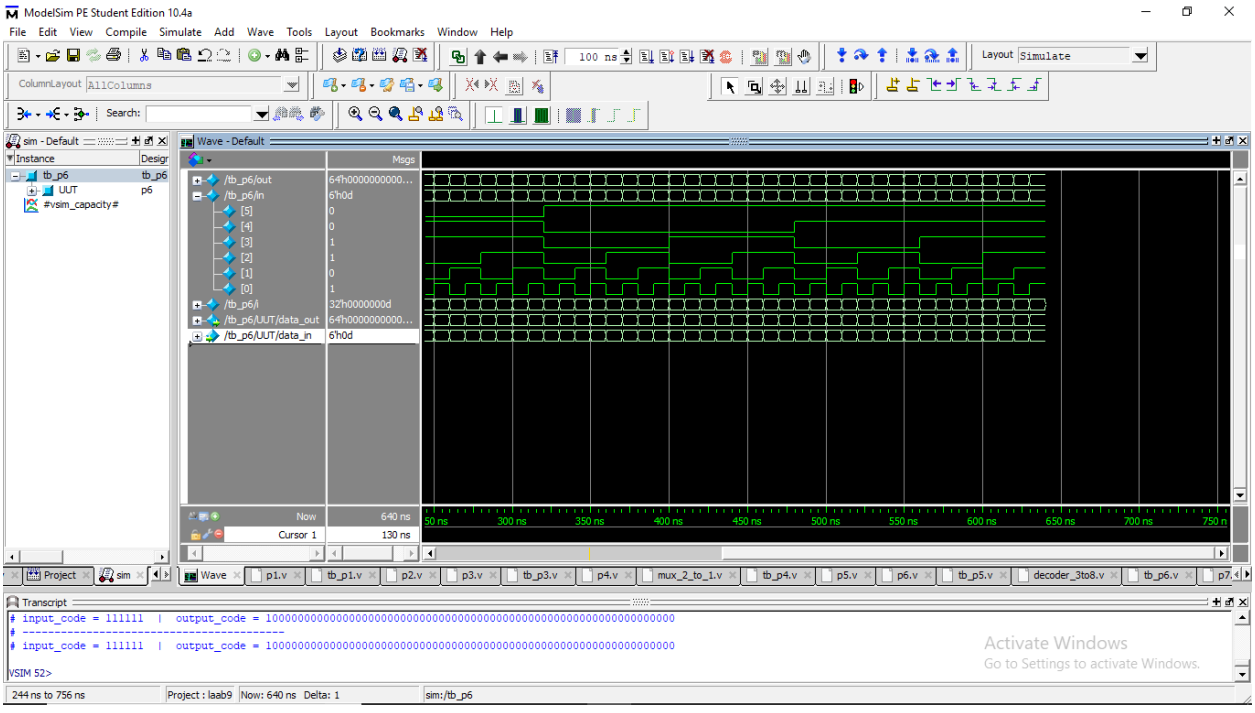
5.


ModelSim PE Student Edition 10.4a
File Edit View Compile Simulate Add Wave Tools Layout Bookmarks Window Help
sim:/tb_p5/O[51] @ 54 ns
1'h0
O = 4611686018427387904 & I = 111110
O = 9223372036854775808 & I = 111111
O = 9223372036854775808 & I = 111111
VSIM 62>
Activate Windows
Go to Settings to activate Windows.


ModelSim PE Student Edition 10.4a
File Edit View Compile Simulate Add Wave Tools Layout Bookmarks Window Help
O = 4611686018427387904 & I = 111110
O = 9223372036854775808 & I = 111111
O = 9223372036854775808 & I = 111111
VSIM 62>
Activate Windows
Go to Settings to activate Windows.

6.


ModelSim PE Student Edition 10.4a
File Edit View Compile Simulate Add Wave Tools Layout Bookmarks Window Help
Wave - Default
/tb_p6/out
Transcript
input_code = 111111 | output_code = 10000000000000000000000000000000000000000000000000000000000000000
input_code = 111111 | output_code = 10000000000000000000000000000000000000000000000000000000000000000
VSIM 52>
Activate Windows
Go to Settings to activate Windows.
244 ns to 756 ns
Project : laab9
Now: 640 ns Delta: 1
/tb_p6/out


ModelSim PE Student Edition 10.4a
File Edit View Compile Simulate Add Wave Tools Layout Bookmarks Window Help
Wave - Default
/tb_p6/out
/tb_p6/in
/tb_p6/i
/tb_p6/UUT/data_out
/tb_p6/UUT/data_in
Transcript
input_code = 111111 | output_code = 10000000000000000000000000000000000000000000000000000000000000000
input_code = 111111 | output_code = 10000000000000000000000000000000000000000000000000000000000000000
VSIM 52>
Activate Windows
Go to Settings to activate Windows.
244 ns to 756 ns
Project : laab9
Now: 640 ns Delta: 1
sim:/tb_p6

7.