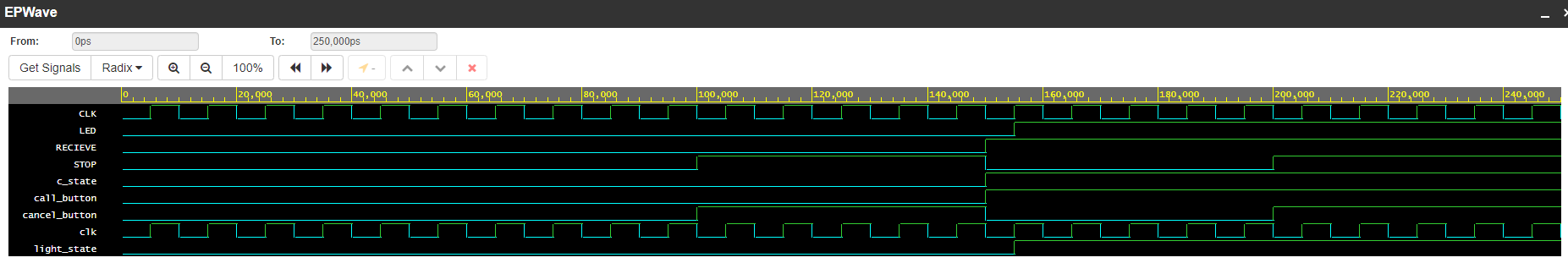
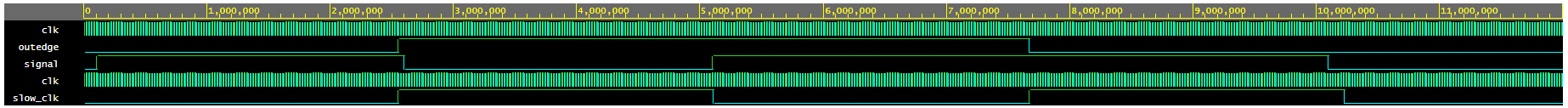
| // Code your testbench here  // or browse Examples  `timescale 1ns / 1ps  module lab4\_part1\_schematic\_tb\_lab4\_part1\_schematic\_sch\_tb;    //inputs  reg RECIEVE;  reg STOP;  reg CLK;    //output  wire LED;    //Instantiate the UUT  fasystem\_bh UUT (  .call\_button(RECIEVE),  .cancel\_button(STOP),  .light\_state(LED),  .clk(CLK)  );    initial CLK = 0;  always #5 CLK = ~CLK;  initial begin  $dumpfile("dump.vcd"); $dumpvars;  $monitor("At t = %0d, CLK = %0b" , $time, CLK);  #250 $finish;  end    initial begin  RECIEVE = 0;  STOP = 0;  end      initial begin  #50  RECIEVE = 0;  STOP = 0;  $display("TC1");  if (LED != 1'b0) $display ("result is wrong");    #50  RECIEVE = 0;  STOP = 1;  $display("TC2");  if (LED != 1'b0) $display ("result is wrong");    #50  RECIEVE = 1;  STOP = 0;  $display("TC3");  if (LED != 1'b1) $display ("result is wrong");    #50  RECIEVE = 1;  STOP = 1;  $display("TC4");  if (LED != 1'b1) $display ("result is wrong");    end    endmodule |
| --- |



Here, in our waveform, we can see that things are as they should be because the light state, and therefore the LED is only transitioned on a rising edge. Also, the light only transitions a wavelength after the call state is activated.

| // Code your testbench here  // or browse Examples  `timescale 1ns / 1ps  module lab4\_part2\_testbench;    reg clk;  reg signal;    wire outedge;  //instantiating the unit under test  lab4\_part2\_bh uut (  .clk(clk),  .signal(signal),  .outedge(outedge)  );  initial begin  signal = 'd0;  clk = 'd0;  end  initial clk = 0;  always #10 clk = ~clk;  initial begin  $dumpfile("dump.vcd"); $dumpvars;  $monitor("at t = %0d, clk = %0b", $time, clk);  #12000 $finish;  end  initial begin  #100  signal = 'd1;  #2500  signal = 'd0;  #2500  signal = 'd1;  #5000  signal = 'd0;    End  endmodule |
| --- |



When clk\_out transitions to 1, so does outedge, and when it transitions back down to 0, it corresponds with the signal

Part 3 test cases

bcdto7led\_bh c1(sw0, sw1, sw2, sw3,

in0[0],in0[1],in0[2],in0[3], in0[4],in0[5],in0[6],in0[7] );

bcdto7led\_bh c2(sw0, sw1, sw2, sw3,

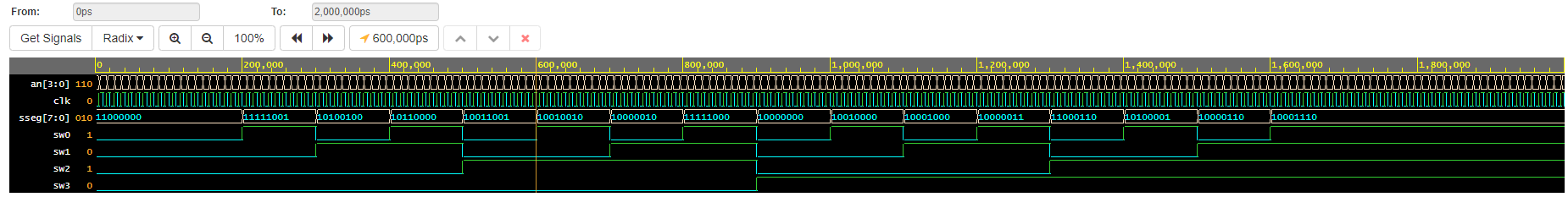
in1[0],in1[1],in1[2],in1[3], in1[4],in1[5],in1[6],in1[7] );

bcdto7led\_bh c3(sw0, sw1, sw2, sw3,

in2[0],in2[1],in2[2],in2[3], in2[4],in2[5],in2[6],in2[7] );

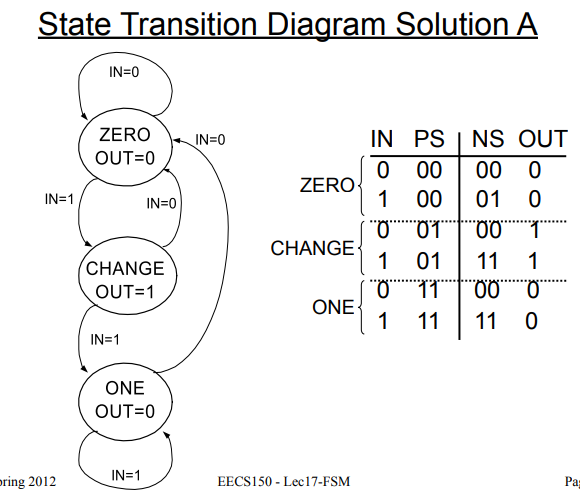
bcdto7led\_bh c4(sw0, sw1, sw2, sw3,

in3[0],in3[1],in3[2],in3[3], in3[4],in3[5],in3[6],in3[7] );



Here, we can see that all 16 of our test cases are visible and that they are all correct if we match the binary values to our test cases in our code.

| Part 3 Testbench  // Code your testbench here  // or browse Examples  `timescale 1ns / 1ps  module lab4\_part3\_testbench;    //inputs  reg clk;  reg sw0;  reg sw1;  reg sw2;  reg sw3;    //outputs  wire [3:0] an;  wire [7:0] sseg;    //instantiating the UUT  displaymux\_main\_bh uut (  .clk(clk),  .sw0(sw0),  .sw1(sw1),  .sw2(sw2),  .sw3(sw3),  .an(an),  .sseg(sseg)  );  initial begin  clk = 0;  sw0 = 0;  sw1 = 0;  sw2 = 0;  sw3 = 0;  end    always #5 clk = ~clk;  initial begin  $dumpfile("dump.vcd"); $dumpvars;  $monitor("At t = %0d, clk = %0b" , $time, clk);  #2000 $finish;  end    initial begin  //initializing inputs  #100  sw0 = 0;  sw1 = 0;  sw2 = 0;  sw3 = 0;    #100  sw0 = 1;  sw1 = 0;  sw2 = 0;  sw3 = 0;    #100  sw0 = 0;  sw1 = 1;  sw2 = 0;  sw3 = 0;    #100  sw0 = 1;  sw1 = 1;  sw2 = 0;  sw3 = 0;        #100  sw0 = 0;  sw1 = 0;  sw2 = 1;  sw3 = 0;    #100  sw0 = 1;  sw1 = 0;  sw2 = 1;  sw3 = 0;    #100  sw0 = 0;  sw1 = 1;  sw2 = 1;  sw3 = 0;    #100  sw0 = 1;  sw1 = 1;  sw2 = 1;  sw3 = 0;        #100  sw0 = 0;  sw1 = 0;  sw2 = 0;  sw3 = 1;    #100  sw0 = 1;  sw1 = 0;  sw2 = 0;  sw3 = 1;    #100  sw0 = 0;  sw1 = 1;  sw2 = 0;  sw3 = 1;    #100  sw0 = 1;  sw1 = 1;  sw2 = 0;  sw3 = 1;        #100  sw0 = 0;  sw1 = 0;  sw2 = 1;  sw3 = 1;    #100  sw0 = 1;  sw1 = 0;  sw2 = 1;  sw3 = 1;    #100  sw0 = 0;  sw1 = 1;  sw2 = 1;  sw3 = 1;    #100  sw0 = 1;  sw1 = 1;  sw2 = 1;  sw3 = 1;  end    endmodule |
| --- |



Out = PS0’ AND PS1

NS0 = IN

NS1 = IN AND PS0