Digital Logic Design, Final Lab Project

“Design of Traffic Light Controller with Pedestrian Crossing”

ERIC NGUYEN (*009726509*)

Navyashree Chandraiah

11/26/2019

EE 118, SECTION 5

Abstract:

The purpose of this lab was to design a four way traffic stop using Verilog code in Xilinx ISE Design Suite. After creating a state table (*Figure 1)*, we can get a sense of how the traffic controller should behave based on data input.

Introduction/Background:

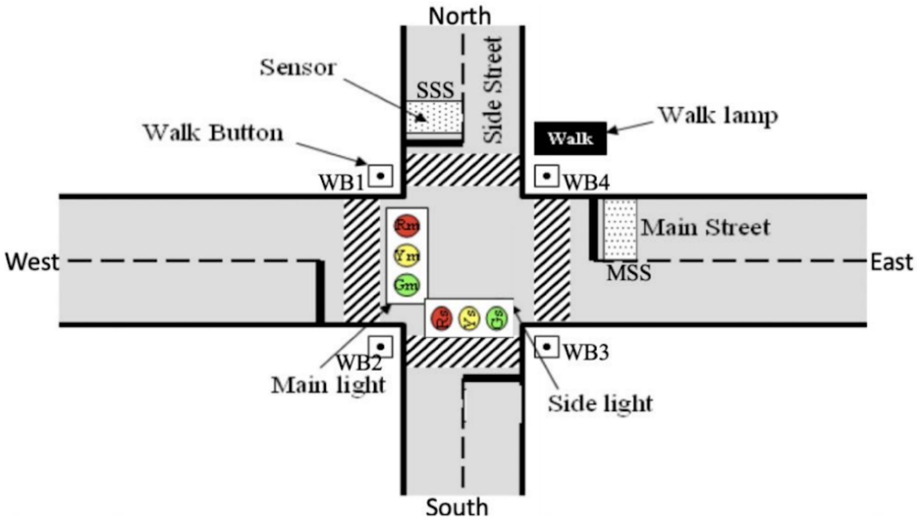


Figure: Traffic flow and overall view of the system

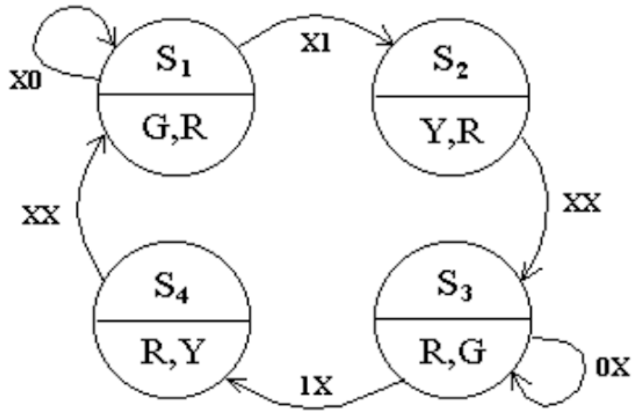
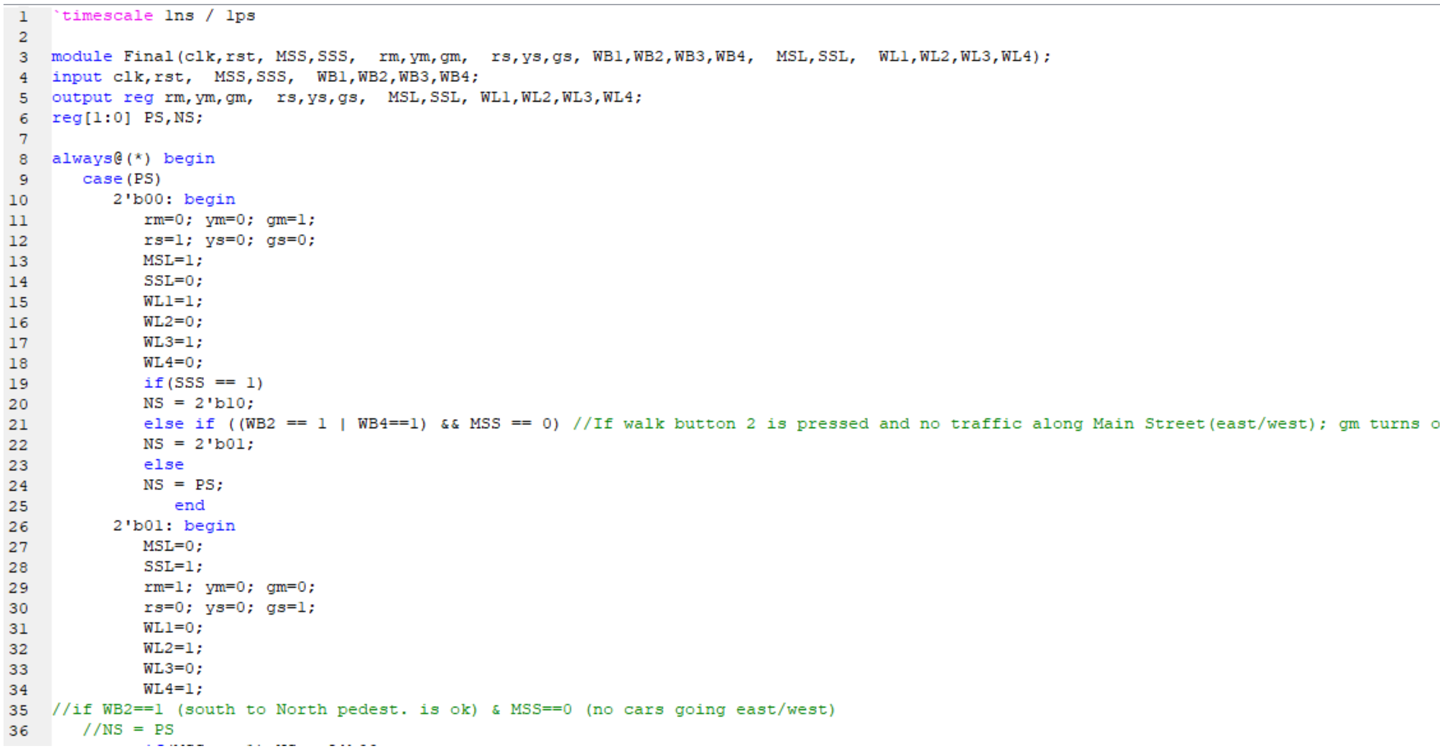


Figure 1: *State Diagram for the Traffic Controller*

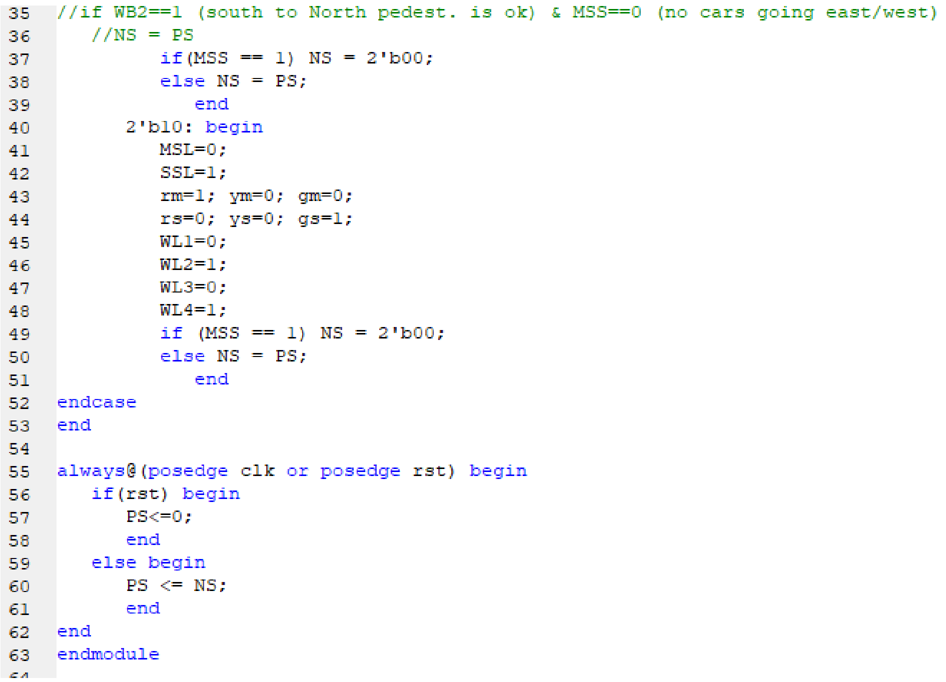
The traffic light controller system has two street sensors (MSS & SSS), corresponding to Main Street which goes horizontally East-West and Side Street which goes vertically North-South. There are four pedestrian walk buttons that should inform the traffic controller that pedestrians need to cross; the walk button input values being WB1, WB2, WB3, WB4. The walk buttons have matching pedestrian lamps in the same place. According to the map figure above, outputs WL1, WL2, WL3, WL4 are stationed on the same matching corners as their button input counterparts WB1, WB2, WB3, WB4. The pedestrian lights WL1 and WL3 tell pedestrians when they're allowed to cross horizontally across Main Street. Similarly, WL2 and WL4 alert pedestrians when they can cross vertically across Side Street.

The system design and state diagram shown in *Figure 2* is fashioned in the similar form of a Moore Machine; meaning that the output is dependent on the 'present state' of the machine. In the following code, the traffic control system output is dependent on the behaviour of the street sensor inputs: MSS and SSS. The sensors tell us what the present state of the traffic system is and what the output response will be. In S_1(state 1), the Main Street Light reads green while the Side Street Light reads red. In S_2, Main Street Light is yellow while Side Street is red. The transition from $S_1 \rightarrow S_2$occurs if the Side Street Sensor detects vehicles approaching from the North and/or the South; this can be understood by the '$x1$' written along the arrow path in *Figure 1*. S_3has Main Street light displaying the color red while the Side Street light is green for cars and pedestrians to travel vertically North-South. S_4 transitions from S_3if the main street sensor (MSS) is activated; it doesn't matter if SSS detects any cars. This can be interpreted as a 'don't care' when simplifying Karnaugh maps; signified as '$1x$' written along the arrow path in *Figure 1.*

<u>Implementation</u>:



```verilog
`timescale 1ns / 1ps

module Final(clk,rst, MSS,SSS,  rm,ym,gm,  rs,ys,gs, WB1,WB2,WB3,WB4,  MSL,SSL,  WL1,WL2,WL3,WL4);
input clk,rst,  MSS,SSS,  WB1,WB2,WB3,WB4;
output reg rm,ym,gm,  rs,ys,gs,  MSL,SSL, WL1,WL2,WL3,WL4;
reg[1:0] PS,NS;

always@(*) begin
   case(PS)
      2'b00: begin
         rm=0; ym=0; gm=1;
         rs=1; ys=0; gs=0;
         MSL=1;
         SSL=0;
         WL1=1;
         WL2=0;
         WL3=1;
         WL4=0;
         if(SSS == 1)
         NS = 2'b10;
         else if ((WB2 == 1 | WB4==1) && MSS == 0) //If walk button 2 is pressed and no traffic along Main Street(east/west); gm turns o
         NS = 2'b01;
         else
         NS = PS;
            end
      2'b01: begin
         MSL=0;
         SSL=1;
         rm=1; ym=0; gm=0;
         rs=0; ys=0; gs=1;
         WL1=0;
         WL2=1;
         WL3=0;
         WL4=1;
//if WB2==1 (south to North pedest. is ok) & MSS==0 (no cars going east/west)
   //NS = PS
```



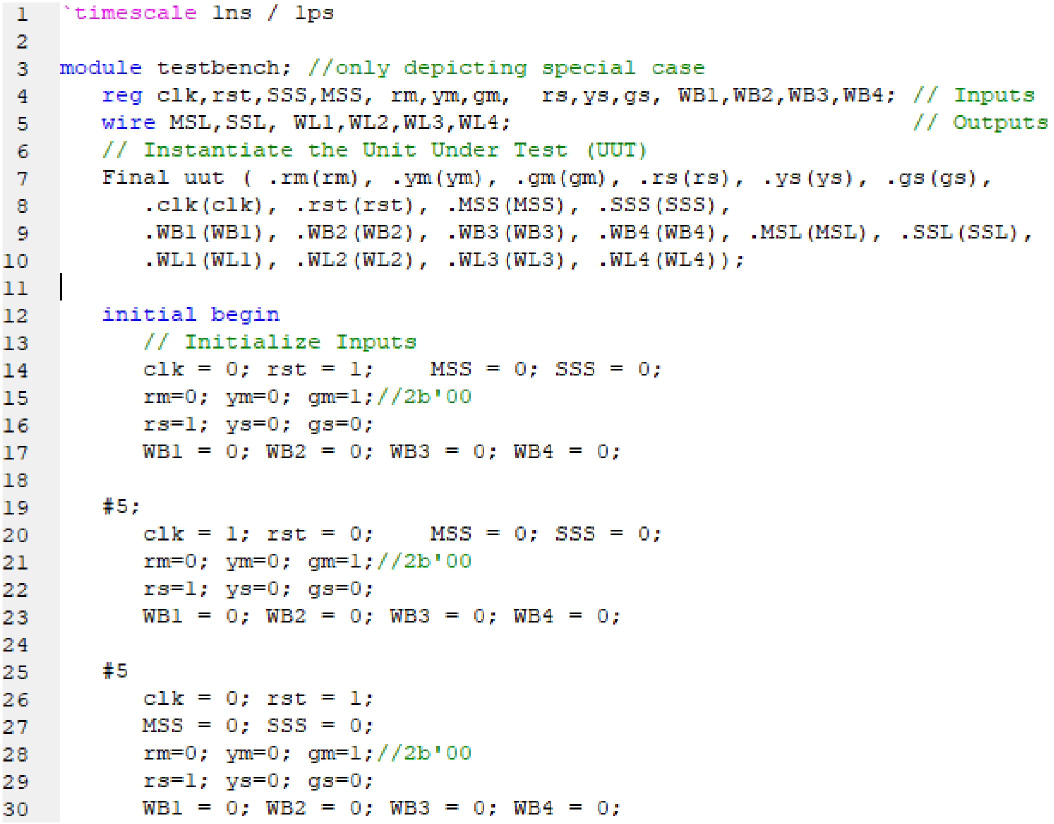
```verilog
//if WB2==1 (south to North pedest. is ok) & MSS==0 (no cars going east/west)
   //NS = PS
         if(MSS == 1) NS = 2'b00;
         else NS = PS;
            end
      2'b10: begin
         MSL=0;
         SSL=1;
         rm=1; ym=0; gm=0;
         rs=0; ys=0; gs=1;
         WL1=0;
         WL2=1;
         WL3=0;
         WL4=1;
         if (MSS == 1) NS = 2'b00;
         else NS = PS;
            end
endcase
end

always@(posedge clk or posedge rst) begin
   if(rst) begin
      PS<=0;
      end
   else begin
      PS <= NS;
      end
end
endmodule
```

Figure 2: *Verilog Code*

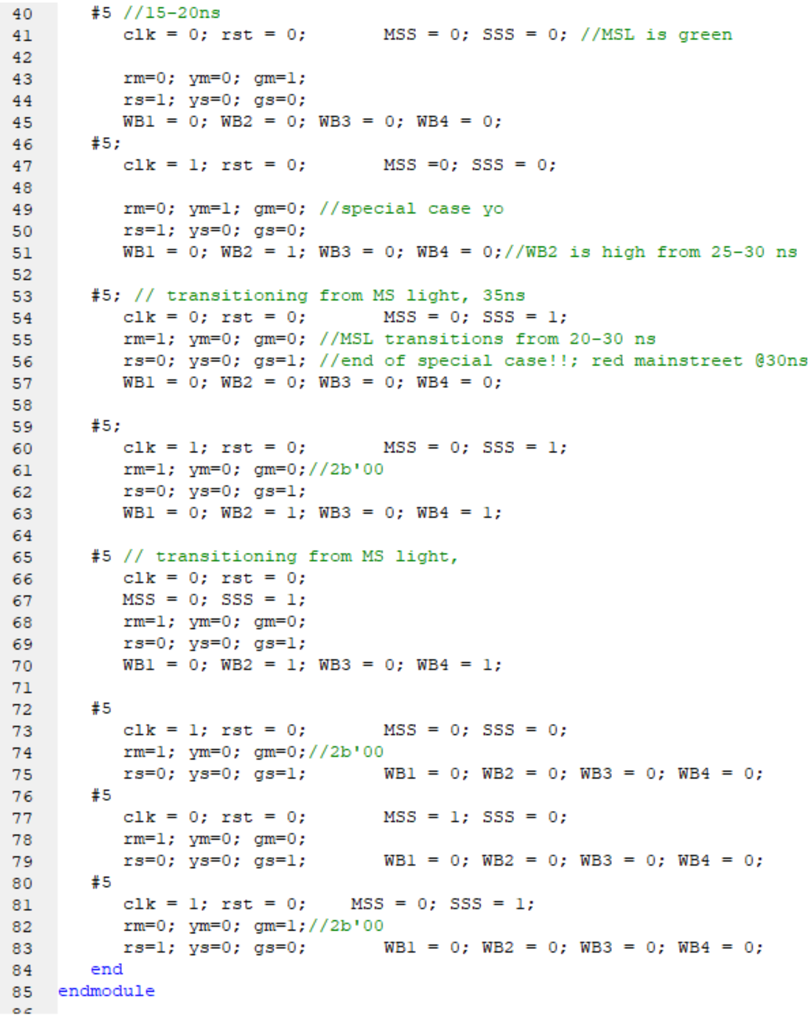
The Verilog code used in *Figure 2* mirrors a Moore Machine, meaning that its outputs MSL and SSL are dependent upon the two inputs MSS and SSS. The traffic light control system will always begin at ‘ $2'b00$.’ With traffic lights being: green along Main Street and red along Side Street; allowing traffic to more easily flow along the more congested Main Street. This will always occur if the Side Street Sensor (SSS) is high; SSS has a value of 1. It will transition to ‘$2'b01$ ’ if pedestrian walk buttons WB2 or WB4 are pressed and the Main Street Sensor doesn’t detect any cars traveling East-West (MSS=0). If there is no change to SSS and MSS, there will be no change in the traffic light control system output. In other words, the ‘next state’ is equal to the present state.

If the traffic light control system begins with the MSS being activated on the other hand, it enters a state in the traffic light control system where the Main Street Light is red and the Side Street Light (SSL) is green; noted by the ‘$2'b10$ ’ state in *Figure 2*. In this case, pedestrians stationed at the top right and bottom left are allowed to cross vertically North-South as well once the SSL is green; noted in code by WB2=1 and WB4=1.

Simulation Results and Discussion:

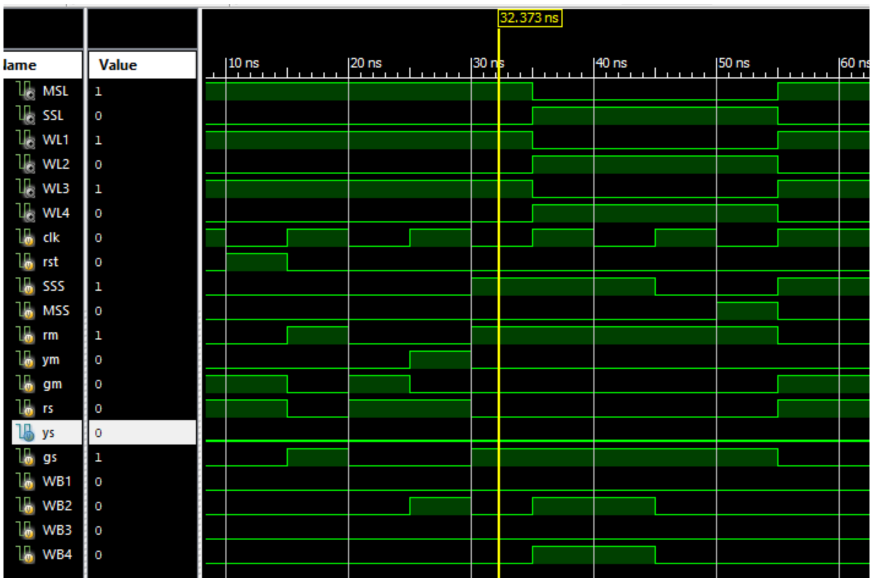


```
`timescale 1ns / 1ps

module testbench; //only depicting special case
   reg clk,rst,SSS,MSS, rm,ym,gm,  rs,ys,gs, WB1,WB2,WB3,WB4; // Inputs
   wire MSL,SSL, WL1,WL2,WL3,WL4;                             // Outputs
   // Instantiate the Unit Under Test (UUT)
   Final uut ( .rm(rm), .ym(ym), .gm(gm), .rs(rs), .ys(ys), .gs(gs),
      .clk(clk), .rst(rst), .MSS(MSS), .SSS(SSS),
      .WB1(WB1), .WB2(WB2), .WB3(WB3), .WB4(WB4), .MSL(MSL), .SSL(SSL),
      .WL1(WL1), .WL2(WL2), .WL3(WL3), .WL4(WL4));

   initial begin
      // Initialize Inputs
      clk = 0; rst = 1;     MSS = 0; SSS = 0;
      rm=0; ym=0; gm=1;//2b'00
      rs=1; ys=0; gs=0;
      WB1 = 0; WB2 = 0; WB3 = 0; WB4 = 0;

   #5;
      clk = 1; rst = 0;     MSS = 0; SSS = 0;
      rm=0; ym=0; gm=1;//2b'00
      rs=1; ys=0; gs=0;
      WB1 = 0; WB2 = 0; WB3 = 0; WB4 = 0;

   #5
      clk = 0; rst = 1;
      MSS = 0; SSS = 0;
      rm=0; ym=0; gm=1;//2b'00
      rs=1; ys=0; gs=0;
      WB1 = 0; WB2 = 0; WB3 = 0; WB4 = 0;
```



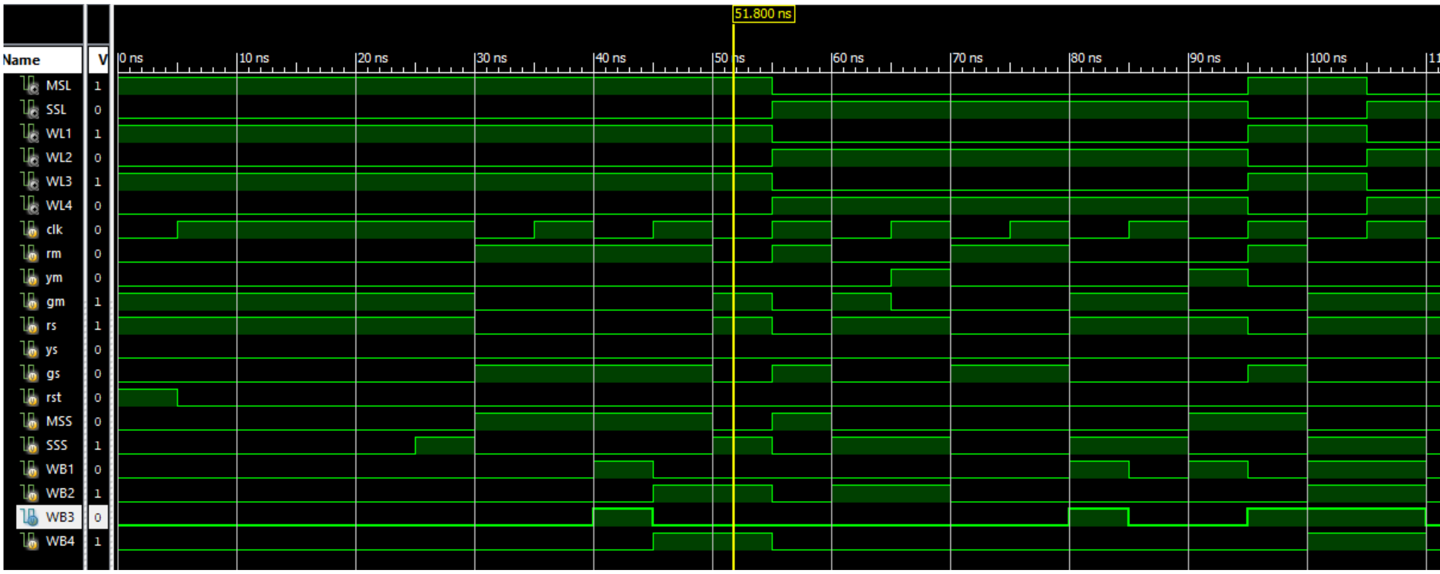
```
   #5 //15-20ns
      clk = 0; rst = 0;       MSS = 0; SSS = 0; //MSL is green

      rm=0; ym=0; gm=1;
      rs=1; ys=0; gs=0;
      WB1 = 0; WB2 = 0; WB3 = 0; WB4 = 0;
   #5;
      clk = 1; rst = 0;       MSS =0; SSS = 0;

      rm=0; ym=1; gm=0; //special case yo
      rs=1; ys=0; gs=0;
      WB1 = 0; WB2 = 1; WB3 = 0; WB4 = 0;//WB2 is high from 25-30 ns

   #5; // transitioning from MS light, 35ns
      clk = 0; rst = 0;       MSS = 0; SSS = 1;
      rm=1; ym=0; gm=0; //MSL transitions from 20-30 ns
      rs=0; ys=0; gs=1; //end of special case!!; red mainstreet @30ns
      WB1 = 0; WB2 = 0; WB3 = 0; WB4 = 0;

   #5;
      clk = 1; rst = 0;       MSS = 0; SSS = 1;
      rm=1; ym=0; gm=0;//2b'00
      rs=0; ys=0; gs=1;
      WB1 = 0; WB2 = 1; WB3 = 0; WB4 = 1;

   #5 // transitioning from MS light,
      clk = 0; rst = 0;
      MSS = 0; SSS = 1;
      rm=1; ym=0; gm=0;
      rs=0; ys=0; gs=1;
      WB1 = 0; WB2 = 1; WB3 = 0; WB4 = 1;

   #5
      clk = 1; rst = 0;       MSS = 0; SSS = 0;
      rm=1; ym=0; gm=0;//2b'00
      rs=0; ys=0; gs=1;       WB1 = 0; WB2 = 0; WB3 = 0; WB4 = 0;
   #5
      clk = 0; rst = 0;       MSS = 1; SSS = 0;
      rm=1; ym=0; gm=0;
      rs=0; ys=0; gs=1;       WB1 = 0; WB2 = 0; WB3 = 0; WB4 = 0;
   #5
      clk = 1; rst = 0;    MSS = 0; SSS = 1;
      rm=0; ym=0; gm=1;//2b'00
      rs=1; ys=0; gs=0;       WB1 = 0; WB2 = 0; WB3 = 0; WB4 = 0;
   end
endmodule
```

Figure 3: *Testbench code depicting only the special case*



lame	Value
MSL	1
SSL	0
WL1	1
WL2	0
WL3	1
WL4	0
clk	0
rst	0
SSS	1
MSS	0
rm	1
ym	0
gm	0
rs	0
ys	0
gs	1
WB1	0
WB2	0
WB3	0
WB4	0

Figure 4: *Waveform depicting only the special case*



Name	V
MSL	1
SSL	0
WL1	1
WL2	0
WL3	1
WL4	0
clk	0
rm	0
ym	0
gm	1
rs	1
ys	0
gs	0
rst	0
MSS	0
SSS	1
WB1	0
WB2	1
WB3	0
WB4	1

Figure 5: *Waveform output with all states*

The traffic light control system's 'next state' output is positively edge triggered; listening to the inputs MSS and SSS whenever the input clock (clk) variable transitions from $0 \rightarrow 1$. This can be seen in *Figure 5* around $55ns,\ 95ns,\ 105ns,\ and\ 120ns$. From $90ns \rightarrow 100ns$, the MSS is high (has a value of 1) and the SSS is low. At $95ns$, the clock transitions from $0 \rightarrow 1$, allowing the Main Street Light (MSL) and Side Street Light values to interchange.

When MSL is high, WSL 1 and WSL 3 are also high, meaning all traffic is allowed to flow East-West. Similarly when SSL is high WSL 2 and WSL 4 are also high; automobile and pedestrian traffic is allowed to flow North-South. The walk button inputs WB1, WB2, WB3, WB4 have no effect on changing the next state output of MSL and SSL. This can be seen throughout the entirety of *Figure 5*. This is due to the Verilog code being modeled in the fashion of a Moore Machine. The next state of the traffic light control system is dependent on the present state inputs. MSL and SSL interchange values when their sensors (MSS and SSS) experience some change at a positive clock edge.

Figures 3 and 4 depict a special scenario, wherein the MSL is green as this is its standard beginning state; usually it would be in response to the SSS being activated beforehand however. In this special scenario, a pedestrian has to cross South to North, parallel to Side Street; pressing WB2. MSS doesn't detect any incoming perpendicular traffic, allowing the MSL to transition from green to yellow and eventually red. Allowing the pedestrian to cross the street. From $20ns \rightarrow 30ns$, we can see the complete transition of the MSL; input rs (red light for Side Street Light) remains low while gm, ym, and rm (green light Main Street, yellow light Main Street, etc.) are sequentially activated. This can be seen in the kind of ascending cascade pattern in *Figure 4*. It can be seen that at the end of this transition and a $5ns$delay around $30 \rightarrow 35ns$, inputs gs (green light Side Street Light) and rm are high (red light Main Street) are both high; paired with SSL, WL2, and WL4 also all being high. This shows that pedestrians at WB2 and WB4 are allowed to cross at this point in time.

This changing of SSL being high and allowing pedestrians to cross is due to the positively edge clocked SSS response at $35ns$.

Conclusion:

In conclusion, for this final lab assignment we were tasked with building a Traffic Light Controller with Pedestrian Crossing; using Xilinx ISE Design Suite. This project incorporated a special situation where the Main Street Light was turned on and a pedestrian needed to cross along the perpendicular: Side Street. The transition states of our traffic light controller depended on the state of the sensors MSS and SSS. If SSS is high at a positive edge clock, SSL should be high and vice versa.