TRAFFIC LIGHT CONTROLLER

BY: AASHI SRIVASTAVA

M.Sc. (TECH) ELECTRONICS, NIT WARANGAL

DATE: 28.01.24

SPECIFICATIONS:

- As soon as there are no cars on the country road, the country road traffic signal turns YELLOW and then RED and the traffic signal on the main highway turns GREEN again.
- There is a sensor (x), to detect the cars waiting on the country road. The sensor sends a signal x as input to controller:

  X=1, if there are cars on the country road, otherwise x=0.
- There are delays on transition from s1 to s2, from s2 to s3 and from s4 to s0. The delays must be controllable. (These s0, s1, s2, s3, s4 are the states described below)
- The traffic signal for the main highway gets highest priority because cars are continuously present on the main highway. Thus, the main highway signal remains GREEN by default. Occasionally, cars from the country road arrive at the traffic signal.

Built a Verilog model for the traffic signal control using state machine diagram.

SOLUTION:

SENSOR CONTROL:

X=1: cars are present on the country road

X=0: cars are not present on the country road

LIGHTS CONTROL (same for both highway lights and country lights) :

RED: Represented by 0 (binary=00)

YELLOW: Represented by 1 (binary=01)

GREEN: Represented by 2 (binary=10)

STATE TABLE:

INPUT	PRESENT STATE	NEXT STATE	HIGHWAY LIGHT	COUNTRY LIGHT
X=0	S0	S0	GREEN	RED
X=1	S0	S1	GREEN	RED
Nil	S1	S2	YELLOW	RED
Nil	S2	S3	RED	RED
X=1	S3	S3	RED	GREEN
X=0	S3	S4	RED	GREEN
Nil	S4	S0	RED	YELLOW

Nil: The transition is not input dependent.

NOTE: Delay of 3 positive edge clock is applied btw s1 to s2 and s4 to s0. Also delay of 2 positive edge clock is applied btw s2 to s3 transition.

DESIGN CODE:

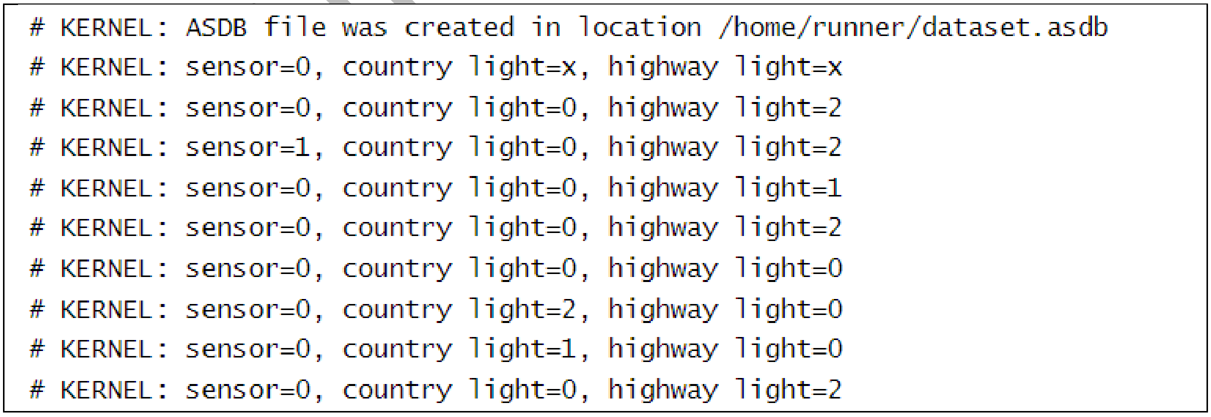
```
// TITLE: TRAFFIC LIGHT CONTROLLER
// DEVELOPED BY: AASHI SRIVASTAVA
// DATE: 28.01.24
//-------------------------------------------------------------

module traffic_light(input clk,rst, input x, output reg [1:0] highway_light, country_light);

 reg [2:0] present_state, next_state;
 parameter red=2'b00;
 parameter yellow=2'b01;
 parameter green=2'b10;
 parameter s0=3'd0,s1=3'd1,s2=3'd2,s3=3'd3,s4=3'd4,s5=3'd5;

 always @(posedge clk)begin
  if(rst)
   present_state<=s0; //default state
  else
   present_state<=next_state;
 end

 always @(*)begin
  case(present_state)
   s0:
    begin
     if(x==1)
      next_state<=s1;
     else
      next_state<=s0;
    end

   s1:
    begin
     repeat(3) @(posedge clk);
     next_state<=s2;
    end

   s2:
    begin
     repeat(2) @(posedge clk);
     next_state<=s3;
    end

   s3:
    begin
     if(x==1)
      next_state<=s3;
     else
      next_state<=s4;
    end

   s4:
    begin
     repeat(3) @(posedge clk);
     next_state<=s0;
    end

   default: next_state<=s0;

  endcase
 end


 always @(*)begin
  case(present_state)
   s0:
    begin
     highway_light<=green;
        country_light<=red;
    end
   s1:
    begin
     highway_light<=yellow;
        country_light<=red;
    end
   s2:
    begin
     highway_light<=red;
        country_light<=red;
    end
   s3:
    begin
     highway_light<=red;
        country_light<=green;
    end
   s4:
    begin
     highway_light<=red;
        country_light<=yellow;
    end
  endcase
 end

endmodule
```

TESTBENCH:

```
module traffic_light_tb();
 reg clk=0,rst;
```

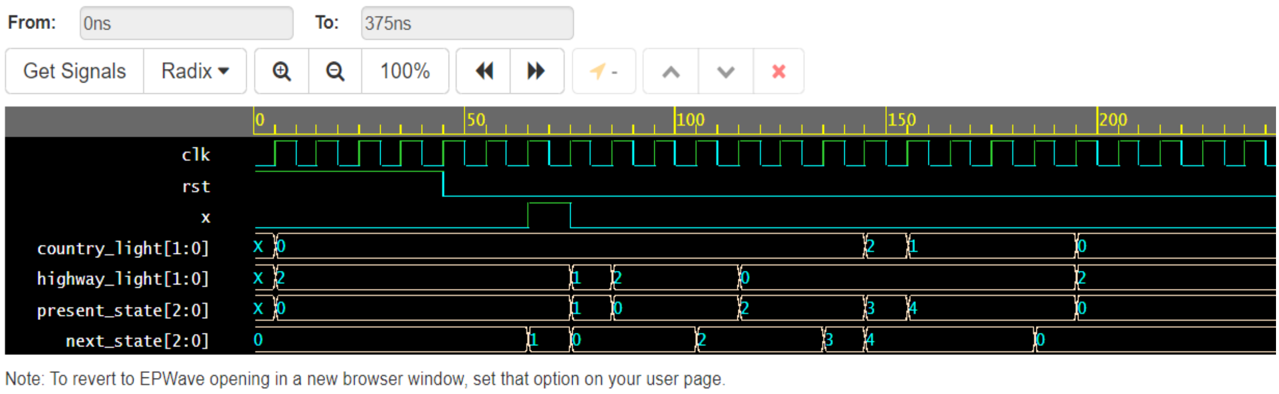
```
  reg x=0;
  wire [1:0] highway_light, country_light;

  traffic_light dut(clk,rst,x,highway_light, country_light);

  always begin
   #5 clk = ~clk;
  end

  initial begin
   rst=1;
   repeat(5) @(posedge clk);
   rst=0;
   #20;
   x=1;
   #10;
   x=0;
   #300;
   $finish;
  end

  initial begin
   $monitor("sensor=%d, country light=%0d, highway light=%0d", x,country_light,highway_light);
   $dumpfile("dump.vcd");
   $dumpvars();
  end

endmodule
```

OUTPUT:



```
# KERNEL: ASDB file was created in location /home/runner/dataset.asdb
# KERNEL: sensor=0, country light=x, highway light=x
# KERNEL: sensor=0, country light=0, highway light=2
# KERNEL: sensor=1, country light=0, highway light=2
# KERNEL: sensor=0, country light=0, highway light=1
# KERNEL: sensor=0, country light=0, highway light=2
# KERNEL: sensor=0, country light=0, highway light=0
# KERNEL: sensor=0, country light=2, highway light=0
# KERNEL: sensor=0, country light=1, highway light=0
# KERNEL: sensor=0, country light=0, highway light=2
```

GRAPH:



From: 0ns To: 375ns

Note: To revert to EPWave opening in a new browser window, set that option on your user page.