Design and Implementation of traffic Light Controller

Introduction:

Now a day's need to control traffic, which have junction places. For that purpose traffic departments are provide new European model. In generally many traffic signals are operates on a timing mechanism that changes the light after given time interval. The traffic signal system consists of three important parts, in that traffic light controller is first one. Because of it represent brain of the traffic system. Many traffic light systems operate on a timing mechanism that changes the lights after a given interval. An intelligent traffic light system senses the presence or absence of vehicles and reacts accordingly. The idea behind intelligent traffic systems is that drivers will not spend unnecessary time waiting for the traffic lights to change. An intelligent traffic system detects traffic in many different ways.

Traffic Light Control:

The traffic signal system consists of three important parts. The first part is the controller, which represents the brain of the traffic system.

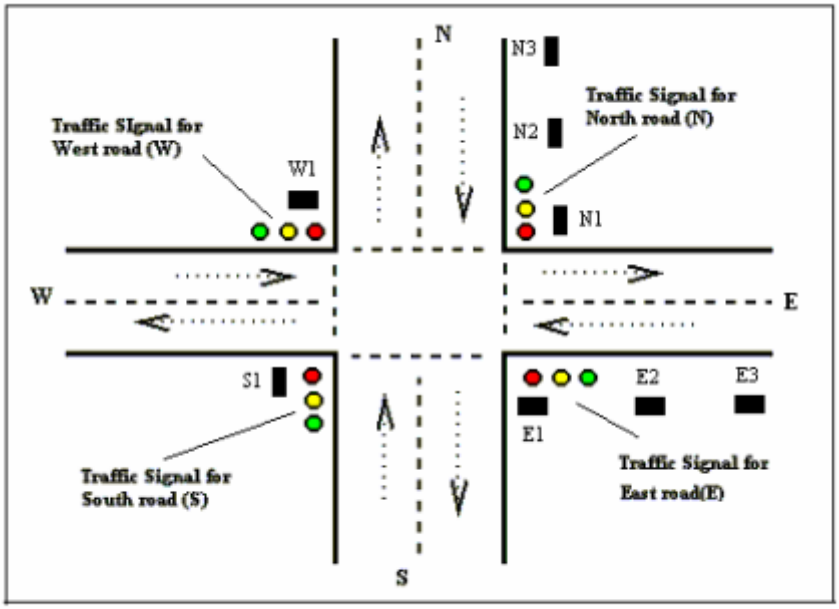


Figure 1: Traffic Light Control

It consists of a computer that controls the selection and timing of traffic movements in accordance to the varying demands of traffic signal as registered to the controller unit by sensors. The second part is the signal visualization or in simple words is signal face. Signal faces are part of a signal head provided for controlling traffic in a single direction and consist of one or more signal sections. These usually comprise of solid red, yellow, and green lights. The third part is the detector or sensor. The sensor or detector is a device to indicate the presence of vehicles. One of the technologies, which are used today, consists of wire loops placed in the pavement at intersections. They are activated by the change of electrical inductance caused by a vehicle passing over or standing over the wire loop. Recent technology utilization is video detection. A camera feeds a small computer that can "see" if a vehicle is present.

In order to implement the Intelligent Traffic Signal Simulator, one needs to setup and assemble the hardware components and write a program to control the intelligent traffic signal simulator. The layout of the Intelligent Traffic Signal Simulator is displayed in Figure 1. The blocks, which are labeled N1, N2, N3, E1, E2, E3, S1 and W1 are the infrared object detectors.

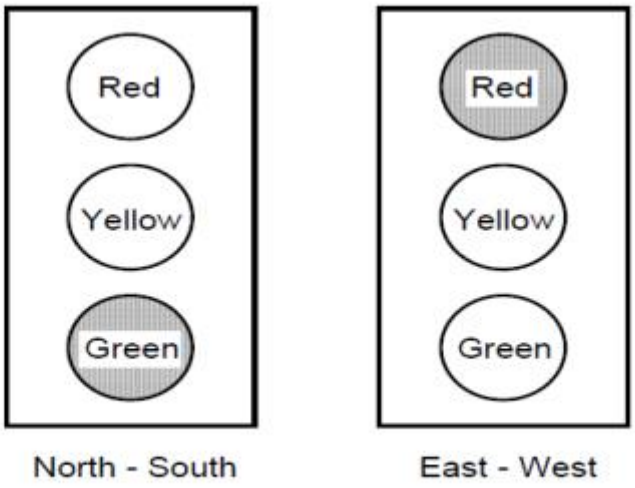


Figure 2: Six colored LEDs can represent a set of traffic lights

It is often useful to be able to sequence through an arbitrary number of states, staying in each state an arbitrary amount of time. For example, consider the set of traffic lights shown in Figure 2. The lights are assumed to be at a four-way intersection with one street going northsouth and the other road going east-west. The Lab VIEW programming is done in the diagram using graphical source code. In the block diagram the program runs from left to right. If the green light in the traffic model does not illuminate, the system goes into default since there is no input into the system. The signal from the sensor is acquired through the DAQ, which is connected, to the computer.

Design Process of Traffic Light Controller:

A single 3-lamp traffic light is considered as a finite state machine. It has three states, Red, Yellow, and Green, which are also the outputs. A single input for the traffic light is defined, with values 0 for no change and 1 for change. This input is connected to the output of a countdown timer, which outputs a 1 when it reaches zero. Thus for a single light, we can draw the state transition diagram as shown in Figure 3.

To simulate these traffic lights we will use the red, yellow, and green LEDs connected to ld[7:2] on the BASYS board and cycle through the six states shown in Table 1.

Table 1: Traffic Light States

State	North - South	East - West	Delay (sec.)
0	Green	Red	5
1	Yellow	Red	1
2	Red	Red	1
3	Red	Green	5
4	Red	Yellow	1
5	Red	Red	1

A state diagram for controlling these traffic lights is shown in Figure 3. If we use a 3 Hz clock to drive this state diagram then a delay of 1 second is achieved by staying in a state for three clock cycles. Similarly, a delay of 5 second is achieved by staying in a state for fifteen clock cycles. The count variable in Figure 3 will be reset to zero when moving to the next state after a timeout.

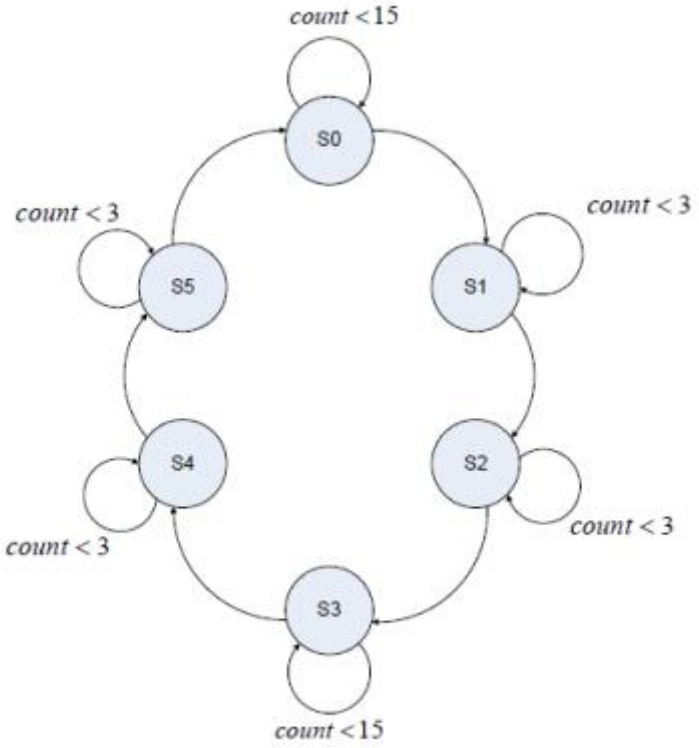


Figure 3: State diagram for controlling traffic lights.

Verilog Code:

```
module Intelligent_Traffic_Control(input clk, clr, X, output reg [1: 0]hwy, cntry, output reg [2: 0]State);

parameter S0=3'b000,S1=3'b001,S2=3'b010,S3=3'b011,S4=3'b100,S5=3'b101;

parameter RED=2'b00, YELLOW=2'b01, GREEN=2'b10;

initial

begin

State<=S0;

hwy=GREEN;

cntry=RED;

end


always @(posedge clk)

begin

case(State)

S0:begin

    hwy=GREEN;

    cntry=RED;

  end

S1:begin

    hwy=YELLOW;

    cntry=RED;

  end

S2:begin

    hwy=RED;

    cntry=RED;

  end

S3:begin

    hwy=RED;

    cntry=GREEN;

  end
```

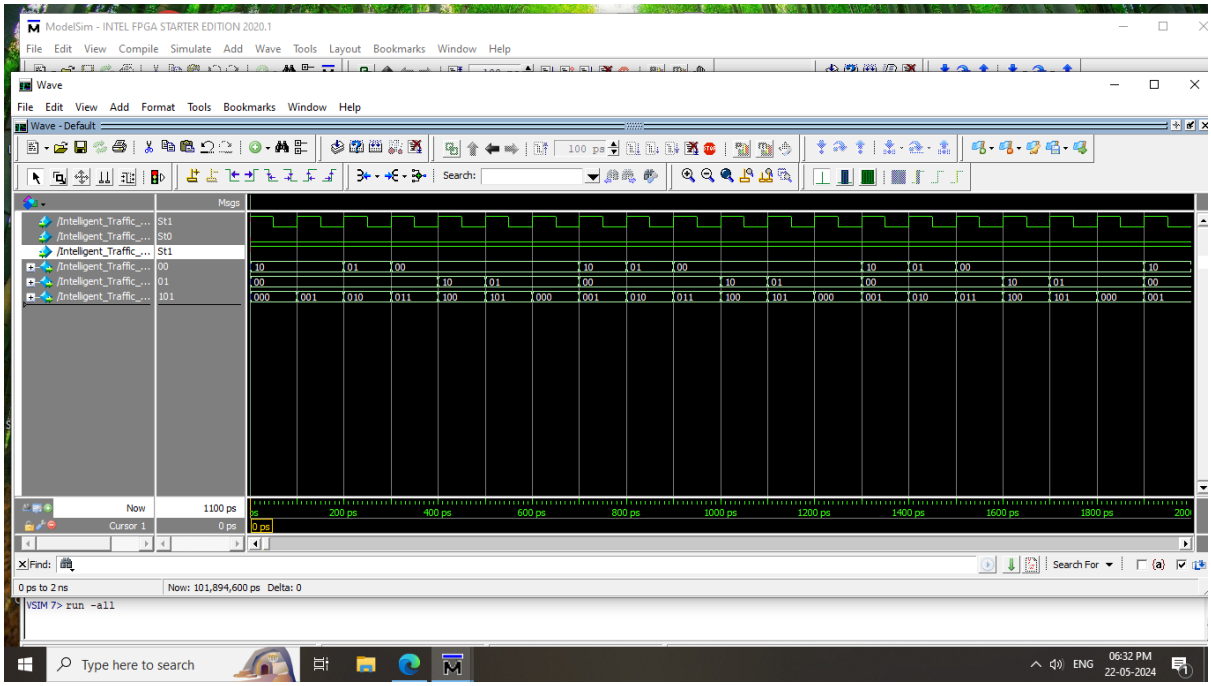
```
S4:begin
    hwy=RED;
    cntry=YELLOW;
  end
S4:begin
    hwy=RED;
    cntry=RED;
  end
endcase
end

always @(posedge clk)
begin
/*if(rst==0)
State<=S0;*/
if(clr)
State<=S0;
else
  begin
  case(State)

  S0:begin
      if(X==4'b1111)
          State<=S0;
      else
          State<=S1;
  end
  S1:begin
      if(X==4'b0011)
          State<=S1;
      else
```

```
            State<=S2;
    end
    S2:begin
        if(X==4'b0011)
            State<=S2;
        else
            State<=S3;
    end
    S3:begin
        if(X==4'b1111)
            State<=S3;
        else
            State<=S4;
    end
    S4:begin
        if(X==4'b0011)
            State<=S4;
        else
            State<=S5;
    end
    S5:begin
        if(X==4'b0011)
            State<=S5;
        else
            State<=S0;
    end
endcase
end
end
endmodule
```

Test Bench:

```
Module Intelligent_Traffic_Light_TB();

reg clk;

reg clr;

reg X;

wire [1: 0]hwy, cntry;

wire [2: 0]State;

Intelligent_Traffic_Control dut (clk, clr, X, hwy, cntry, State);

endmodule
```

Waveform:


ModelSim - INTEL FPGA STARTER EDITION 2020.1
File Edit View Compile Simulate Add Wave Tools Layout Bookmarks Window Help
Wave
File Edit View Add Format Tools Bookmarks Window Help
Wave - Default
Search:
Msgs
/Intelligent_Traffic_... St1
/Intelligent_Traffic_... St0
/Intelligent_Traffic_... St1
/Intelligent_Traffic_... 00
/Intelligent_Traffic_... 01
/Intelligent_Traffic_... 101
Now 1100 ps
Cursor 1 0 ps
200 ps 400 ps 600 ps 800 ps 1000 ps 1200 ps 1400 ps 1600 ps 1800 ps
Find:
Search For
0 ps to 2 ns
Now: 101,894,600 ps Delta: 0
VSIM 7> run -all
Type here to search
ENG
06:32 PM
22-05-2024