Traffic Signal Controller
Final Project

[Haiyi Zhang]
CSE241 Summer 2021

Introduction

The main function of the traffic signal light controller is to control the signal lights on the two crossing highways so that the vehicles can pass through the intersection in an orderly manner. This project is coded and tested by SystemVerilog, and combines with D flip-flops and the style of gate level to achieve function.

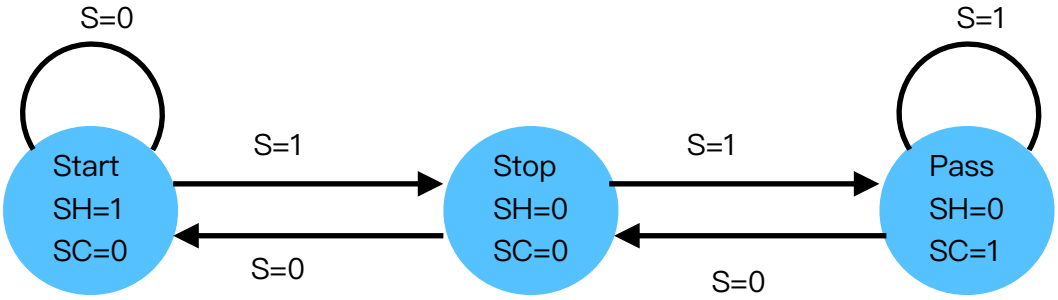
Problem Summary

Congestion and traffic accidents are prone to occur on two intersecting roads, so it is necessary to design a traffic signals controller to effectively manage the roads. There are now two intersecting one-way roads and a signal light on each road. However, because of the fault, the signals can only display red and green, which means that when designing the controller, it should be considered that both signal lights need to be temporarily set to red every time the status of the signal light is changed to avoid traffic accidents. In addition, the highway is green and the country road is red in the default state, but currently there is only one sensor on the country road, so the status of the signal light should be changed according to whether there are vehicles on the country road.

Design Methodology

The purpose of the project is to use the sensor and the signal lights to design a system that enables vehicles on the crossing road to pass in an orderly manner. Therefore the system has one input—sensor(S) and two outputs—-Highway road Signal(SH) and Country road Signal(SC).

There is a state diagram showing what I expect the light to achieve.



In the initial state, highway road signal is green and country road signal is red,SH=1,SC=0. When there is no car on the country road, S=0, the signal lights maintain the current state. When there is a car on the country road, the sensor triggers, S=1. The two signal lights will be temporarily set to red lights,SH=0,SC=0, then the highway road light turns to red, and the country road light turns to green,SH=0,SC=1. When there are still cars on the country

road,S=1, the signals maintain current state. If there is no car on the country road, S=0, the signal light will restore the default setting. So the two signal lights will temporarily be set to red lights,SH=0,SC=0, then the highway lights are green, and the country roads are red lights,SH=1,SC=0.

All the states that appear in the state machine need to be tested, which ensure that the controller can work effectively under any situations.

Present State	Input	Next State
SH=1 SC=0	S=0	SH*=1 SC*=0
SH=1 SC=0	S=1	SH*=0 SC*=1
SH=0 SC=1	S=0	SH*=1 SC*=0
SH=0 SC=1	S=1	SH*=0 SC*=1
SH=0 SC=0	S=0	SH*=1 SC*=0
SH=0 SC=0	S=1	SH*=0 SC*=1

Implementation Preparation

State Table

Present	x = 0	x = 1	SH	SC
	Next		Out	
Start	Start	Stop	1	0
Stop	Start	Pass	0	0
Pass	Stop	Pass	0	1

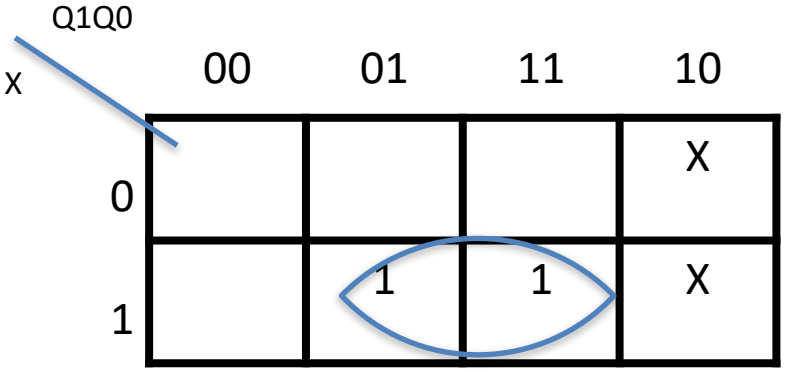
The state table is drawn according to the state diagram and using the name Start, Stop and Pass to represent the states.

Transition Table

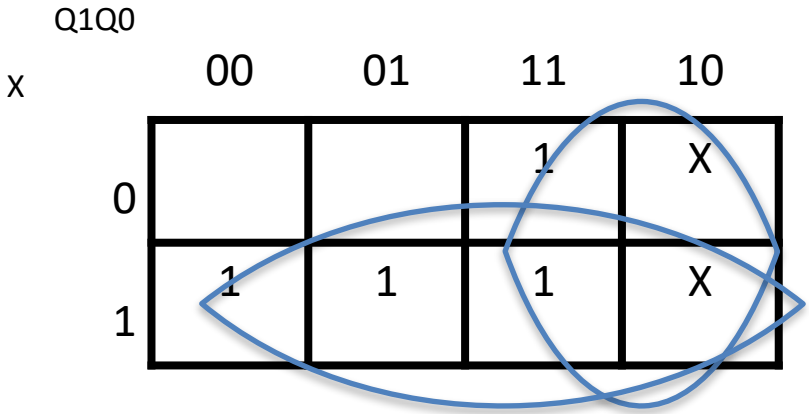
Present	x = 0	x = 1	SH	SC
	Next		Out	
00	00	01	1	0
01	00	11	0	0
11	01	11	0	1
10	xx	xx	x	x

Translation Table is created by using Gray Code Encoding
00 represents Start, 01 represents Stop, 11 represents Pass and we do not care the state 10.

K-Maps



It is the K–map of Q1



It is the K–map of Q0

Q1 / Q0	0	1
0	1	X
1		

It is the K-map of SH

Q1 Q0	0	1
0		X
1		1

It is the K-map of SC

Transition Equations

Q1* = Q0X
Q0* = Q1 + X
SH = Q0'
SC = Q1

SystemVerilog Implementation

```
module dff (clk,set, reset,d, q, qn); // D-Flip Flop
  input      clk,set,reset,d; //declare inputs clk, reset and d
  output     q,qn;        //declare outputs q and qn
  reg        q;           //declare q is reg
  assign qn = ~q;                    //assign qn equal not q

  always @(posedge clk or posedge reset)
  begin
    if (reset) begin  // asynchronous reset when reset goes high

      q <= 0;    //non-blocking
    end else begin    // assign D to Q on positive clock edge

      q <= d;    //non-blcking
    end
  end
endmodule
```

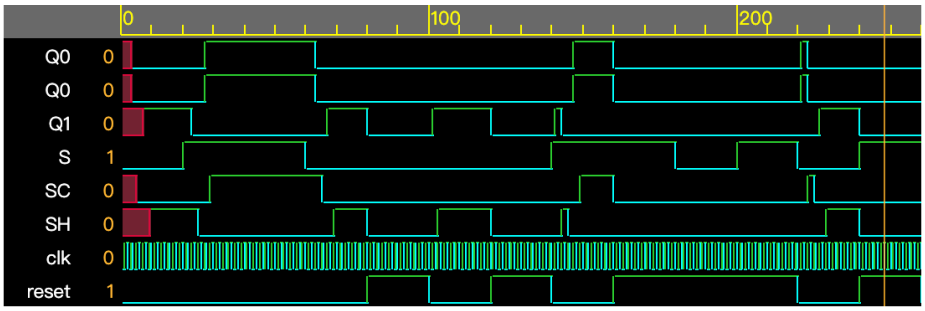
```
module Signals(clk,S,SH,SC,set,reset);
  input clk, S,set,reset; // declare input clk, S, set and reset
  output reg SH, SC;      // declare output reg SH and SC

  reg Q1, Q0;  //declare reg Q1 and Q0

  wire D1,D0,nS,nQ1,nQ0;  //create wires for gates
  not #1(nS,S);    // NOT gate for S
  not #1(nQ1, Q1); // NOT gate for Q1
  not #1(nQ0,Q0);  // NOT gate for Q0

  and #2(D1,nQ0,nS);  // and gate for nQ0 and nS
  and #2(D0,nQ1,S);   // and gate for nQ1 and S

  wire nSH,nSC,n1,n2;  //Create wires
  dff t1(.clk(clk),.d(D1),.q(Q1),.qn(nQ1),.set(set),.reset(reset));
  //Q1<=D1
  //instantiate clk with clk, d with D1, q with Q1, qn with nQ1, set with set, reset with reset
  dff t2(.clk(clk),.d(D0),.q(Q0),.qn(nQ0),.set(set),.reset(reset));
   //Q0<=D0
   //instantiate clk with clk, d with D0, q with Q0, qn with nQ0, set with set, reset with reset
  dff t3(.clk(clk),.d(Q1),.q(SH),.qn(nSH),.set(set),.reset(reset));
   //SH<=Q1
   //instantiate clk with clk, d with Q1, q with SH, qn with nSH, set with set, reset with reset
  dff t4(.clk(clk),.d(Q0),.q(SC),.qn(nSC),.set(set),.reset(reset));
   //SC<=Q0
  //instantiate clk with clk, d with Q0, q with SC, qn with nSC, set with set, reset with reset

endmodule
```

My purpose is to control the signal lights to let the vehicles pass in sequence, so I use a sequence logic, and the output of the signals depend on the current and last inputs. The code uses D flip flop to store the last state. Except for the FF module, everything is a gate-level style with delay time, which avoids propagation delay.

Evaluation and Analysis

Q1 is present state of Highway road signal.
Q0 is present state of Country road signal.
SH is next state of Highway road signal.
SC is next state of Country road signal.
S represent Sensor
Clk is a clock signal
Reset is the reset of D-FF.

When reset is 0 and S changes, SC and SH will change according to Q0 and Q1 and S values.
When reset is 1, Q1,Q0, SC and SH equal to 0/



Reflection on Design

If I redesign this system, I will consider the duration of the two signal lights being red at the same time. I want to set a counter for the system to set the duration of the signal lights being red at the same time. In addition, I want to change the outputs SH and SC to 2 bits vectorized variables, one bit represents current state, another one represents next state.

Reflection on Experience

It is true that I encountered a big problem in time management, because I am also doing internships during the summer. Therefore, during the day, I have almost no time to complete homework and review. So the time I study is after get off work and on weekends. For the problem of insufficient study time, I think the best solution is to reduce the time for entertainment and rest. So on weekends, I spent very concentrated time studying in the room and did not go outside or do other things. In fact, I like to spend very concentrated time studying problems, which can ensure the continuity of my thinking when thinking about problems.

If I had to do this project again, I would definitely not do other jobs. I would focus on the project and communicate more with professor to improve efficiency.

Conclusion

Setting the expected function, listing the test conditions, drawing the state diagram, and writing the transition equations. After the code pass the test cases, I get a signal light controller, which combines D-FF and gates to form a sequential logic circuit. The implementation is able to control the signal lights of the two roads to make the vehicles pass through the intersection in an orderly manner, which is helpful to avoid congestion and traffic accidents.