Chiao Lu 204848946

Yuhai Li 104844760

Shen Teng 104758168

CS152B Lab 2 Report

# Introduction

In this project, we will be implementing a traffic light controller that controls a main street, a side street, and walk lamps using finite state machine.

# Description

The traffic light controller is for an intersection between a Main Street and a Side Street. Both streets have a red, yellow, and green signal light. Pedestrians have the option of pressing a walk button to turn all the traffic lights red and cause a single walk light to illuminate. Moreover, there is a sensor on the Side Street which tells the controller if there are cars still on the Side Street.

This is summarized below:

The side street sensor is placed near the intersection to tell the controller when there are cars passing over the sensor. We assumed that the sensor will remain constantly high if several cars pass over the sensor, rather than quick pulses, provided the cars are close enough together. You do not need to implement this specific functionality. This input is named Sensor.

The traffic lights are timed on three parameters (in seconds): the base interval (6 seconds), the extended interval (3 seconds), and the yellow light interval (2 seconds).

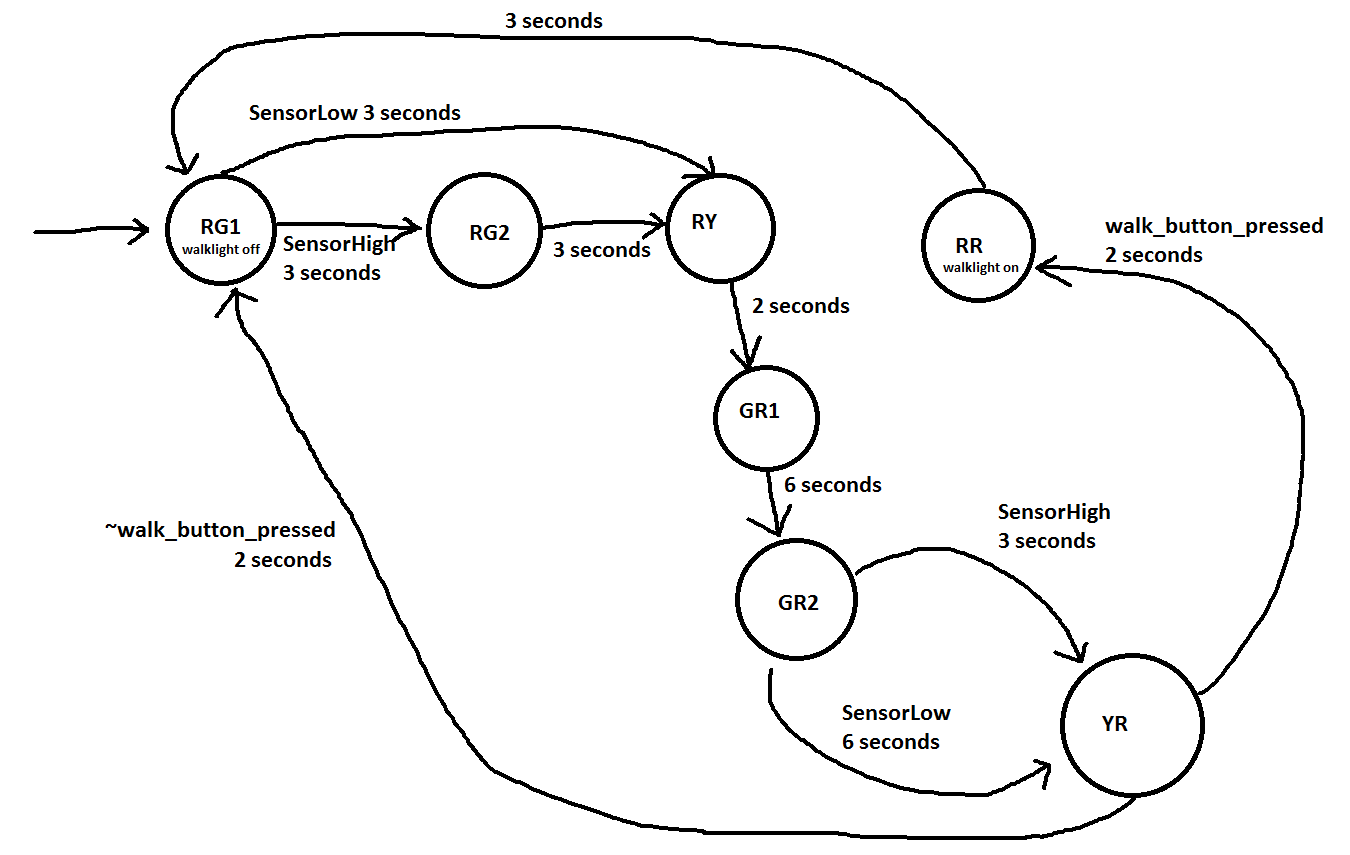
The operating sequence of this intersection begins with the Main Street having a green light for 12 seconds. Next, the Main lights turn to yellow for 2 seconds and then turn red while simultaneously turning on the Side Street green light. The Side Street is green for 6 seconds, and yellow is held for 2 seconds. Whenever a stoplight is green or yellow, the other street’s stoplight is red. Under normal circumstances, this cycle repeats continuously.

There are several ways the controller can deviate from the typical loop:

1. A walk button allows pedestrians to submit a walk request. The internal Walk Register should be set on a button press and the controller should service the request after the Main Street yellow light by turning all street lights to red and the walk light to on. After a walk of 3 seconds, the traffic lights should return to their usual routine by turning the Side Street green. The Walk Register should be cleared at the end of a walk cycle.
2. The second deviation is the traffic sensor. If the traffic sensor is high at the end of the first 6 second length of the Main street green, the light should remain green only for an additional 3 seconds, rather than the full 6 seconds. Additionally, if the traffic sensor is high during the end of the Side Street green, it should remain green for an additional 3 seconds.

It is required to debounce the input signals in order to get the buttons to work properly.

# State Diagram



The first letter of each state name represents the color of main street light while the second letter represents the color of side street light. The followed number of each state name is used to digtingurish different states with same name.

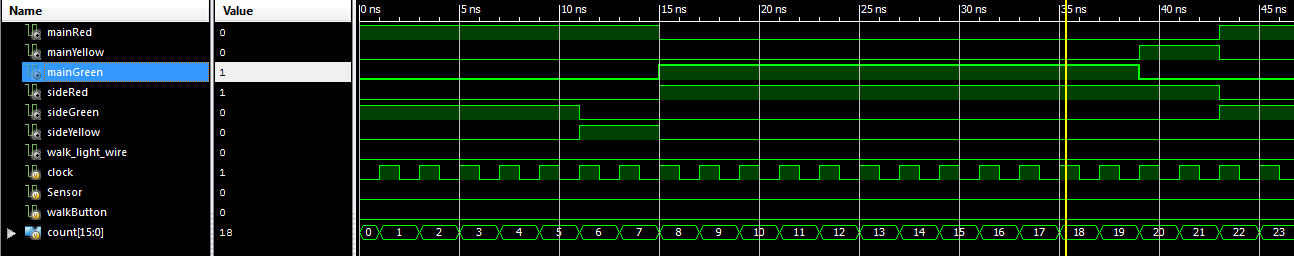
The walk\_button can be pressed any time.

We determine whether we should jump to RR state and turn on the walk light at the end of YR state.

We turn the walk light off at state RG1.

# Testbench/Waveforms

## Sensor is Low:

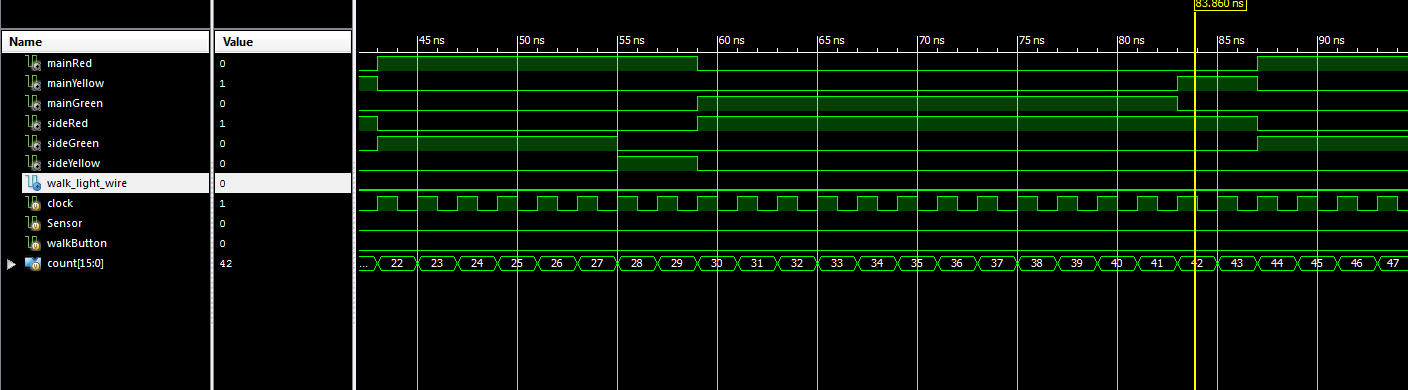


second 0 - 5: mainRed sideGreen

second 6 - 7: mainRed sideYellow

second 8 - 19: mainGreen sideRed

second 20 - 21: mainYellow sideRed

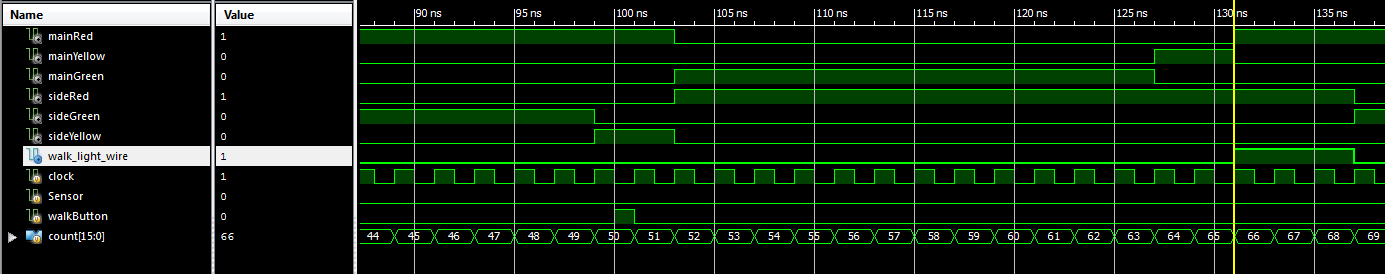


second 22 - 27: mainRed sideGreen

second 28 - 29: mainRed sideYellow

second 30 - 41: mainGreen sideRed

second 42 - 43: mainYellow sideRed



second 44 - 49: mainRed sideGreen

walk button pressed!

second 50 - 51: mainRed sideYellow

second 52 - 63: mainGreen sideRed

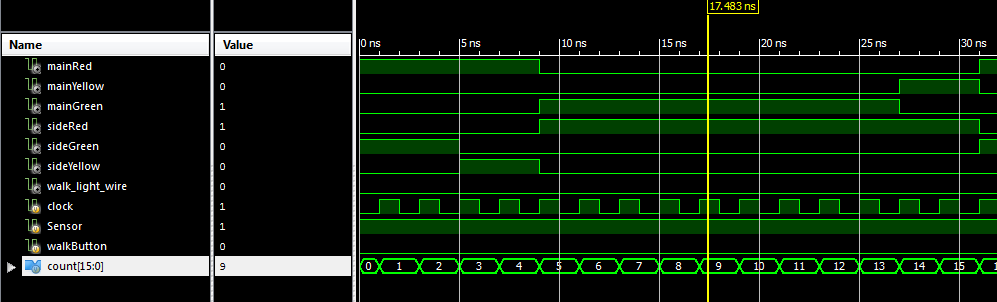
second 64 - 65: mainYellow sideRed

second 66 - 68: mainRed sideRed

walk light is on

second 69 - 74: mainRed sideGreen

## Sensor is High



second 0-2: mainRed sideGreen

second 3-4: mainRed sideYellow

second 5-13: mainGreen sideRed

second 14-15: mainYellow sideRed

second 16: mainRed sideGreen

second 17: mainRed sideGreen

# Encountered Problems

1. Assigning a variable in different always blocks  
   We originally plan to create a reg variable named walk\_light indicating the status of the walk light. Our design was that walk\_light is turned on when the walk button is triggered, and walk\_light is turned off when (Red, Red) state is over. However, we encountered a syntax error showing that assignments on same variable in different always blocks are not allowed. We eventually solved this problem by defining a new variable named indicating whether or not the walk light should be turned on so that we can access this new variable in one always block and walk\_light in another always block.
2. Walk light is triggered anyway even if button is not pressed  
   We solved this problem by fixing our finite state machine codes. Our code mistakenly changed the state to RR after GR1 at first.