Project Report

CS4362 Hardware Description Languages

Traffic Light Controller

Using Verilog HDL

R. M. K. V. Ratnayake

150533H

# Table of Contents

[Table of Contents 2](#_Toc12827745)

[Introduction 3](#_Toc12827746)

[System Design 3](#_Toc12827747)

[Finite State Machine (FSM) 4](#_Toc12827748)

[Timer 4](#_Toc12827749)

[Divider 4](#_Toc12827750)

[Synchronizer 4](#_Toc12827751)

[Time Parameters 4](#_Toc12827752)

[Walk Register 4](#_Toc12827753)

[Timing Diagrams 5](#_Toc12827754)

[FSM 5](#_Toc12827755)

[FSM general simulation 6](#_Toc12827756)

[FSM sensor simulation 6](#_Toc12827757)

[FSM walking simulation 7](#_Toc12827758)

[Timer 7](#_Toc12827759)

[Divider 7](#_Toc12827760)

[Synchronizer 8](#_Toc12827761)

[Time Parameters 8](#_Toc12827762)

[Walk Register 8](#_Toc12827763)

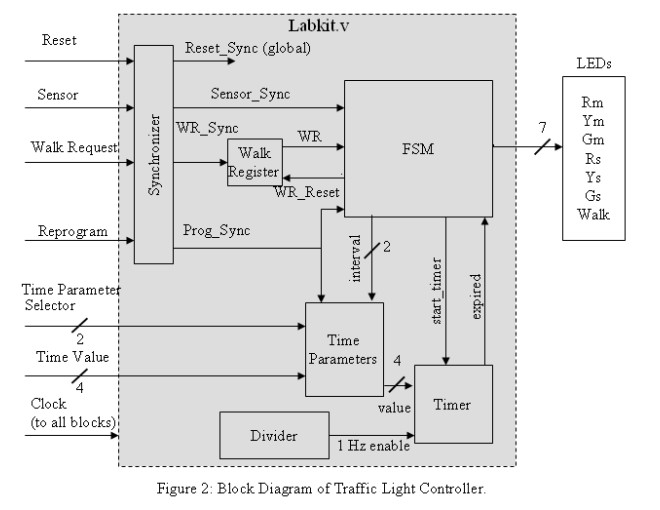
[Overall System 9](#_Toc12827764)

# 

# Introduction

The task is to develop a traffic light controlling system using Verilog HDL and simulate the functionality. The system should be able to detect the level of traffic using a sensor and adapt timing accordingly and to enable pedestrians to cross safely by triggering walk laps as required.

## System Design



## 

## Finite State Machine (FSM)

Holds the core logic of the system as a finite state machine. Each state corresponds to a combination of output light sequence. Therefore we can consider the FSM as a Mealy machine. State transition occur according to the “expired” signal generated by the timer. State transition interval is also assigned by the FSM.

## Timer

Since each output combination keep activated for different time periods, there need to have a separate component to fire interrupts whenever the timeout take place, so Timer takes start\_timer, 1Hz enable, and Time Parameter value as inputs and send expired signal accordingly.

## Divider

Divider generates a 1 Hz enable, which is sent to the timer. The signal generated is a pulse that is high for one clock cycle every 1sec.

## Synchronizer

Synchronize all the input signals with the internal clock. All user input values will hold by the synchronizer until the next rising edge of the clock and with the rising edge, it feeds the values to the system.

## Time Parameters

Allows the user to set different time intervals for output combinations to stay on. Those parameters are Tbase , Tyellow, and Text. Each of these values are 4 bits and is selected using a 2-bit address.

## Walk Register

Holds the value of walk request until it is being served by the FSM and get reset after the FSM is served the request.

# Timing Diagrams

## FSM

S0

Sensor = 0

Sensor = 1

S2

S1

S3

Walk request = 1

Walk request = 0

S4

S5

Sensor = 1

Sensor = 0

S6

S7

|  |  |  |  |
| --- | --- | --- | --- |
| **State** | **Output Pattern**  [Rm,Ym,Gm,Rs,Ys,Gs,WL] | **Affecting Input Signals** | **Time Interval/s** |
| S0 | [0 0 1 1 0 0 0] | Sensor\_Sync | Tbase |
| S1 | [0 0 1 1 0 0 0] |  | Text |
| S2 | [0 0 1 1 0 0 0] |  | Tbase |
| S3 | [0 1 0 1 0 0 0] | Walk Request | Tyellow |
| S4 | [1 0 0 1 0 0 1] |  | Text |
| S5 | [1 0 0 0 0 1 0] | Sensor\_Sync | Tbase |
| S6 | [1 0 0 0 0 1 0] |  | Text |
| S7 | [1 0 0 0 1 0 0] |  | Tyellow |

### 

### FSM general simulation

A screenshot of a video game

Description automatically generated

This is the simulation of the FSM under normal conditions. It moves in the S0, S2, S3, S5, S7 order.

### FSM sensor simulation

A picture containing screenshot

Description automatically generated

This is the simulation of the FSM when the sensor has been triggered. It moves in the S0, S1, S3, S5, S6, S7 order.

### FSM walking simulation

A picture containing screenshot

Description automatically generated

This is the simulation of the FSM when the walk request has been triggered. It moves in the S0, S2, S3, S4, S5, S7 order. Walk reset command can also be seen.

## Timer

A screenshot of a social media post

Description automatically generated

This shows the timer with simulation for 0110 value and 0010 value. Below image shows the timer with reset functionality.

A screenshot of a social media post

Description automatically generated

Reset has been done after two counts. From there on it counts new 6 counts.

## Divider

A picture containing sky, screenshot, indoor

Description automatically generated

This image shows the simulation of the divider. It has been assumed that outside clock and internal clock of divider has a 10: 1 ratio.

## Synchronizer

A picture containing screenshot, green

Description automatically generated

This shows the functionality of the synchronizer. It calibrates the input signals which are outside the clock with correct clock pulse

## Time Parameters

A screenshot of a video game

Description automatically generated

This shows the functionality of the time parameters block. First three values are internally stored defaults values. Then a value assigning occur and then later that value has been returned.

## Walk Register

A picture containing screenshot

Description automatically generated

This simulation results show the walk register. It shows the setting up of the value with wr\_sync as well as resetting it with wr\_reset.

## Overall System

A picture containing screenshot

Description automatically generated

Due to issues with 1000000ps simulation time, only a small number of iterations could be shown.