CS4362 - Hardware Description Languages

Traffic Light Controller

Assignment 2

A.S.K.Jayasena

150256K

24/06/2019

# Task

Task is to develop a traffic light controller system to a intersection where a side street is crossing a main street. Both streets have usual traffic lights for vehicles and pedestrians.

A picture containing screenshot

Description automatically generated

Figure 1 Diagram for intersection with corresponding lights and sensors

All the walk request buttons are attached to the controller using a wired OR. There are two sensors on the side street to detect vehicles that are passing over them. (Assumption: Sensor remains high constantly when several cars pass over it).

The Traffic light controller is timed based on three parameters (in seconds) tBASE, tEXT, tYEL. These parameters can be changed using operation. Default values are as follows.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Interval Name** | **Symbol** | **Parameter Number** | **Default Time (sec)** | **Time Value** |
| Base Interval | tBASE | 00 | 6 | 0110 |
| Extended Interval | tEXT | 01 | 3 | 0011 |
| Yellow Interval | tYEL | 10 | 2 | 0010 |

Table 1 Default timing parameters

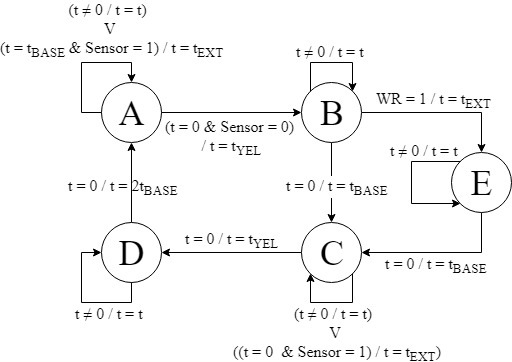
The operating sequence of this intersection begins with the Main Street having a green light for 2 lengths of tBASE seconds. Next, the Main lights turn to yellow for tYEL and then turn red while simultaneously turning on the Side Street green light. The Side Street is green for tBASE, and its yellow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

There are two ways the controller can deviate from the typical loop. First, a walk button allows pedestrians to submit a walk request. The internal Walk Register will set on a button press and the controller will service the request after the Main Street yellow light by turning all streetlights to red and the walk light to on. After a walk of tEXT seconds, the traffic lights will return to their usual routine by turning the Side Street green. The Walk Register will be cleared at the end of a walk cycle.

The second deviation is the traffic sensor. If the traffic sensor is high at the end of the first tBASE length of the Main street green, the light will remain green only for an additional tEXT second, rather than the full tBASE. Additionally, if the traffic sensor is high during the end of the Side Street green, it will remain green for an additional tEXT second.

# Design

State chart diagram



## Top Module (TrafficControllerMain)

# 

Figure 2 Top module block diagram

## Synchronizer

### Module description

On the block diagram, you see that all input signals pass through the synchronizer before going to other blocks. The purpose of the synchronizer is to ensure that the inputs are synchronized to the system clock.

### Module overview

Reset

Sensor

clk

Walk\_Request

Reprogram

Prog\_Sync

WR\_Sync

Sensor\_Sync

Reset\_Sync

## Synchronizer

Figure 3 Synchronizer block diagram

### Test bench output

## Walk Register

### Module description

The Walk Register allows pedestrians to set a walk request at any time. There is also a signal controlled by the finite state machine that will be able to reset the register at the end of the actual walk cycle.

### Module overview

WR\_Sync

WR\_Reset

WR

## Walk Register

Figure 4 Walk Register block diagram

### Test bench output

## Time Parameters

### Module description

The time parameters module stores the three different time parameter values, namely tBASE, tEXT, and tYEL on the FPGA. The module acts like a (small) memory from the FSM and Timer blocks, where the FSM addresses the three parameters and the timer reads the data. From the user's perspective, the three time parameter values can be modified. On a reset, the three parameters should be respectively set to 6, 3 and 2 seconds. However, at any time, the user may modify any of the values by manipulating Time\_Parameter\_Selector, Time\_Value, and Reprogram. Each of these values are 4 bits, and is selected using a 2 bit address. Whenever a parameter is reprogrammed, the FSM should be reset to its starting state.

### Module overview

Selector [2bit]

Time\_value [4bit]

clk

Prog\_Sync

Interval [2bit]

Value [4bit]

## Time Parameters

Figure 5 TimeParameters block diagram

### Test bench output

## Divider

### Module description

The divider is necessary for the timer to properly time the number of seconds for any traffic light state. Using only the clock as input, this module 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.

### Module overview

rst

clk

oneHz\_enable

## Divider

Figure 6 Divider block diagram

### Test bench output

1Hz enable requires 106 clock cycles at 100MHz. Since it is output cannot be visualized clearly 10MHz pulses were generated instead of 1Hz pulses.

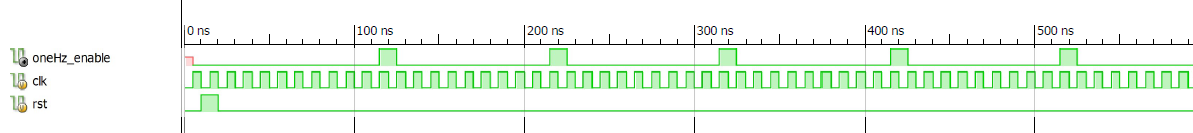


Figure 7 Divider at 10MHz

## Timer

### Module description

The timer is responsible for taking the start\_timer, 1Hz enable, and Time Parameter value to properly time the traffic light controller. When done counting a particular state, the expired signal will go high for one clock period to signal to the FSM that it should change states.

### Module overview

Value [4bit]

oneHz\_enable

start\_timer

expired

## Timer

Figure 8 Timer block diagram

### Test bench output

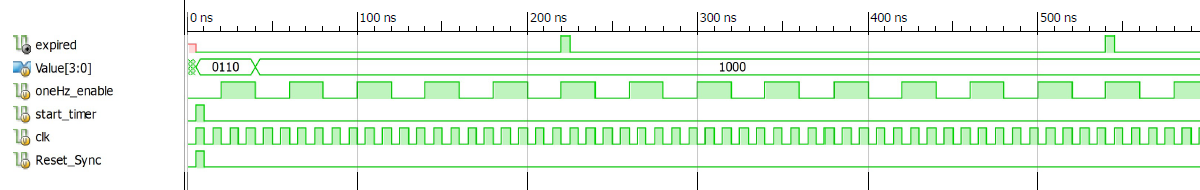


Figure 9 Timer simulation output

## Finite State Machine

### Module description

The finite state machine controls the sequencing for the traffic light. As previously described, it changes states based on the Walk Register and sensor signals, and with the expired signal.

### Module overview

Sensor\_Sync

WR

clk

expired

Prog\_Sync

WR\_Reset

LEDs [7bit]

Interval [3bit]

start\_timer

## FSM

Figure 10 FSM block diagram

### Test bench output

Normal routine

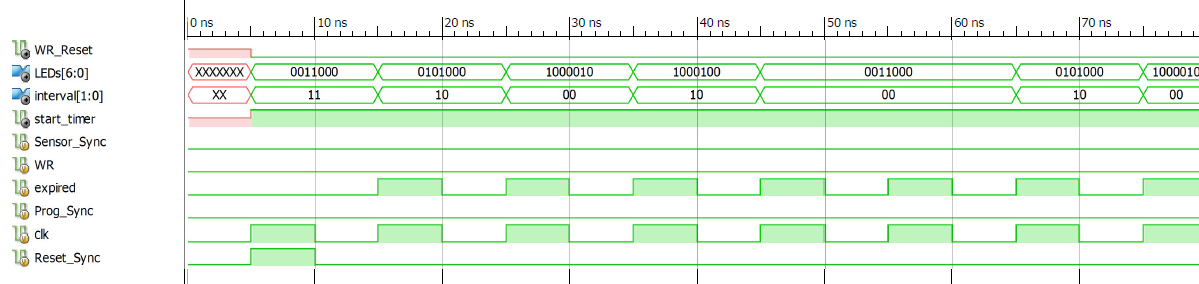


Figure 11 Normal operation

Walk request

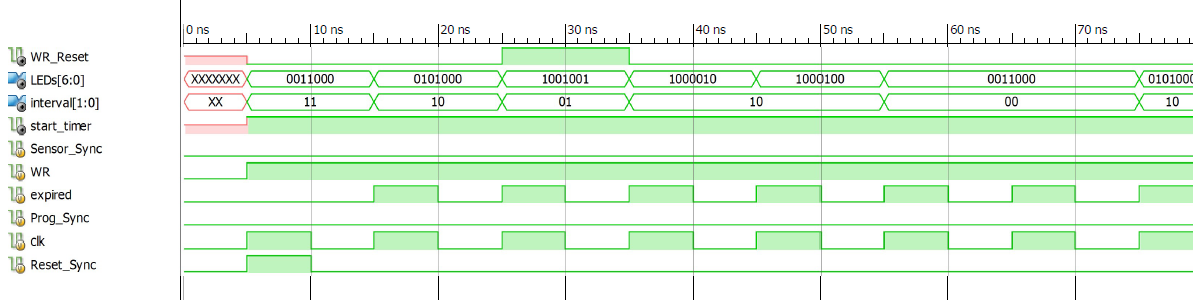


Figure 12 Walk request by pedestrian

Vehicle sensor

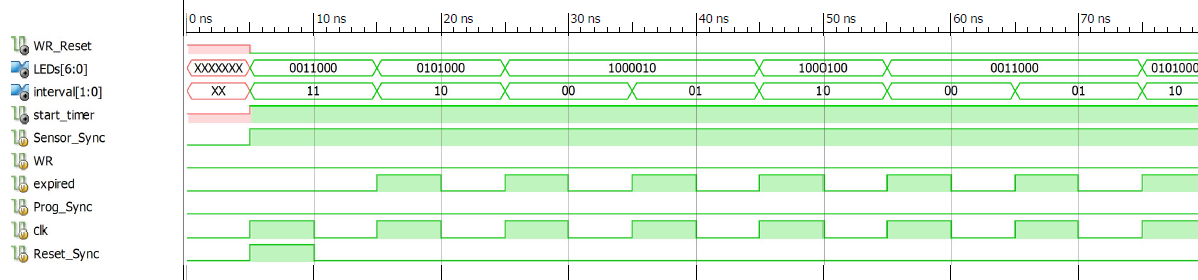


Figure 13 Vehicle sensor activated

LEDs values represent the Following states of the system

Bits of the LEDs represent lights in following manner

LED sequence: [ Redmain, Yellowmain, Greenmain, Redside, Yellowside, Greenside, Walk]

|  |  |  |
| --- | --- | --- |
| **State** | **Representation** | **Meaning** |
| A | 0011000 | Main green |
| B | 0101000 | Main yellow |
| C | 1000010 | Side green |
| D | 1000100 | Side yellow |
| E | 1001001 | walk |

Interval values represent following timer values

|  |  |
| --- | --- |
| **Representation** | **Timer value** |
| 00 | tBASE |
| 01 | tEXT |
| 10 | tYEL |
| 11 | 2\*tBASE |