Report: Traffic Light Controller

CS4362 – Hardware Description Languages

180022T – Lahiru Akmeemana

Problem Statement

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. There is a sensor on the Side Street which detects if there are vehicles in the Side Street.

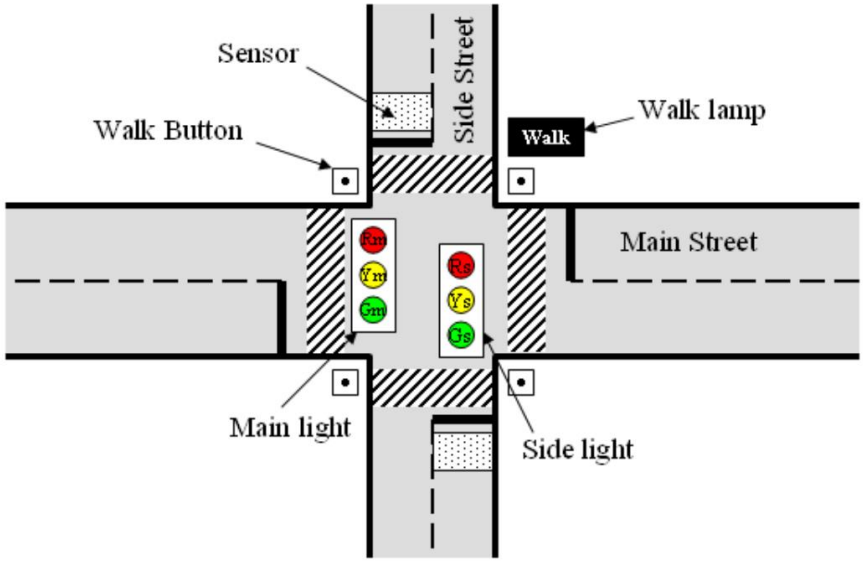

Figure 1: Overview of the system

At each corner there is a walk button placed and they are connected to the system using a wired OR. The traffic lights are timed on three parameters (in seconds): the base interval (tBASE), the extended interval (tEXT), and the yellow light interval (tYEL), and resetting and reprogramming on demand using switches and buttons on Basys3 board with the Time_Parameter_Selector, Time_Value, and Reprogram signals.

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

Figure 2: Default Timing Parameters

Normal process:-

- Main Street green light on for 2 tBase seconds.
- Main Street yellow light on for tYel seconds.
- Main Street red light on and Side Street green light on for tBase seconds.
- Side Street yellow light on for tYel seconds.
- Again, Main Street green light on. This occurs repeatedly.

Deviations

1. Walk request:- Walk button allows pedestrians to submit a walk request. The internal Walk Register should be set on a button press and the controller services 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 return to their usual routine by turning the Side Street green. The Walk Register is cleared at the end of a walk cycle.
2. Traffic Sensor:- If the traffic sensor is high at the end of the first tBASE length of the Main street green, the light remain green only for an additional tEXT seconds. Additionally, if the traffic sensor is high during the end of the Side Street green, it should remain green for an additional tEXT seconds.

The task is to implement a controller system using a finite state machine.

- Implementation of the system can be found following Git repository
    - https://github.com/lahiruakmeemana/TrafficLightController.git

Design

- In each state timer(t) is set accordingly
- One of the tBase, tExt, tYel time periods are set in each state. Additional states are added to handle rest of the cases.
- Reset or Reprogram(R) leads to initial state of the system.
- System outputs several signals for visualization purposes
- LED sequence as follows,
    - Main-Red, Main-Yellow, Main-Green, Side-Red, Side-Yellow, Side-Green, Walk

State machine diagram

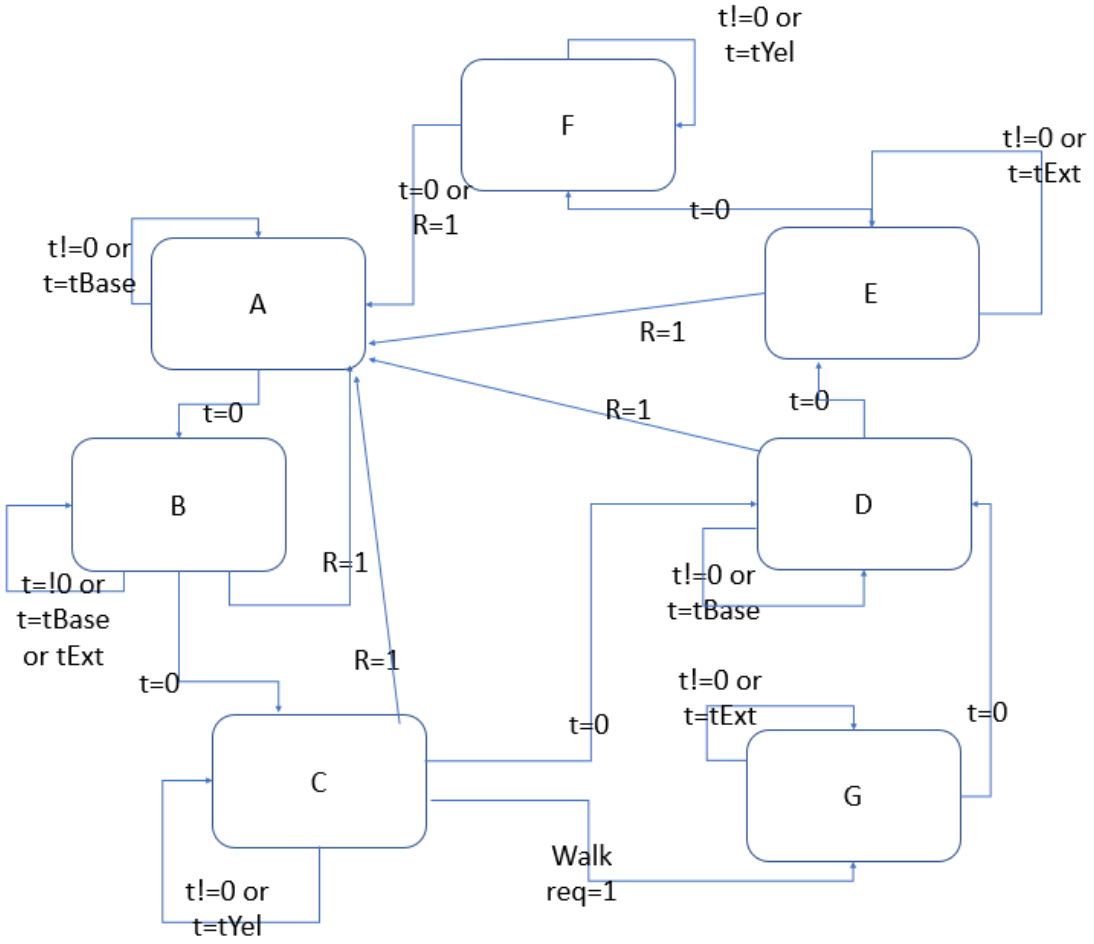

Figure 3: Finite State Machine

State	LEDs ON	State timer value
A	Gm, Rs	tBase
B	Gm, Rs	tExt if Sensor else tBase
C	Ym, Rs	tYel
D	Rm, Gs	tBase
E	Rm, Gs	tExt if Sensor else 0
F	Rm, Ys	tYel
G	Rm, Rs, Walk	tExt

Figure 4: State details

Circuit design

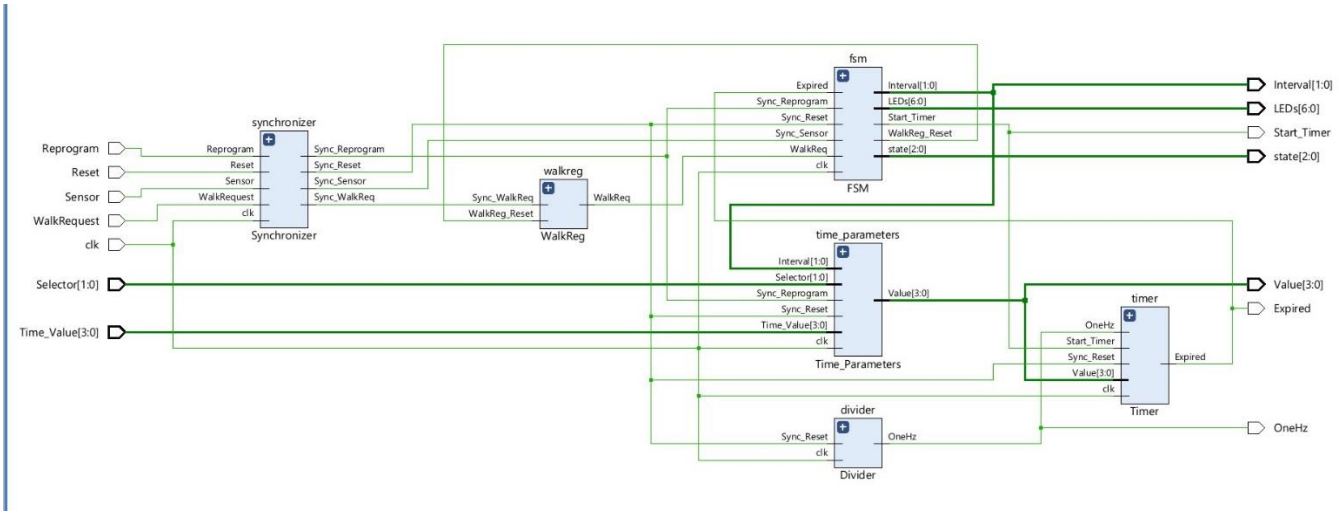

Figure 5: System block diagram

Modules and Simulations

Synchronizer

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.

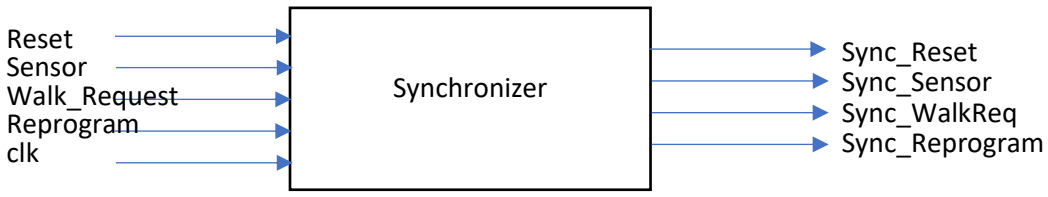

Figure 6: Synchronizer block diagram

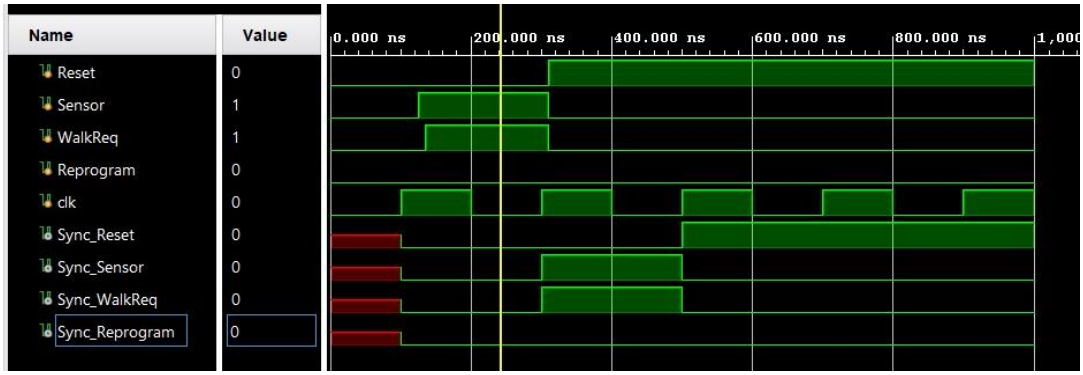

Figure 7: Synchronizer simulation

Walk Register

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.

Figure 8: Synchronizer block diagram

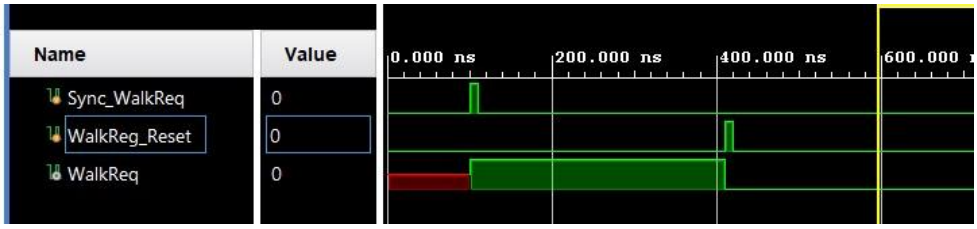

Figure 9: WalkReg simulation

Time Parameters

This module stores the three different time parameter values, namely tBASE, tEXT, and tYEL. On a reset, the three parameters 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.

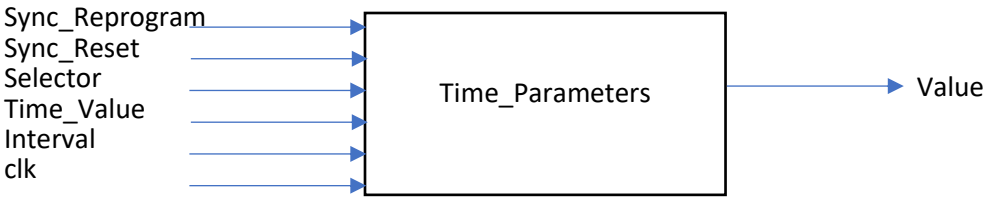

Figure 10: Time_Parameters block diagram

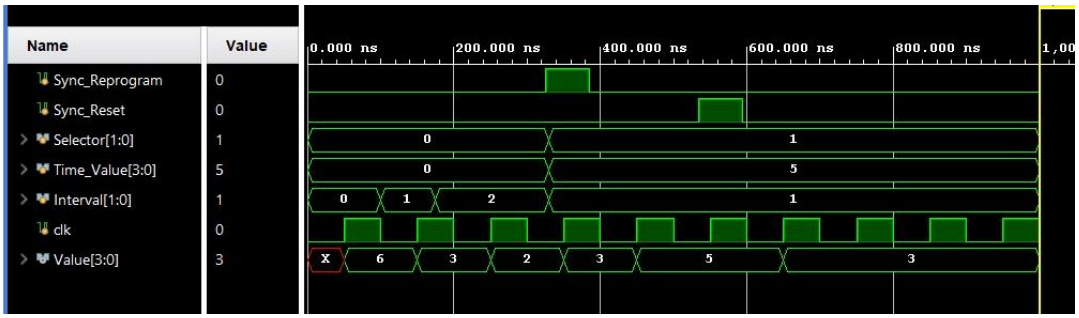

Figure 11: Time_Parameters simulation

Divider

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.

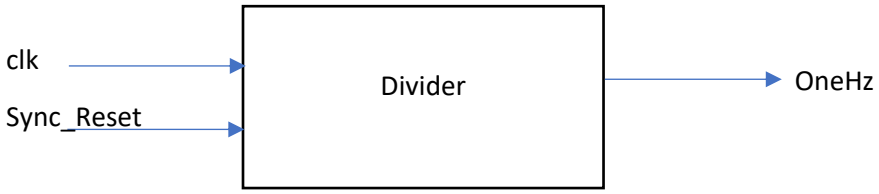

Figure 12: Divider block diagram

Figure 13: Divider simulation

Timer

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 go high for one clock period to signal to the FSM that it should change states.

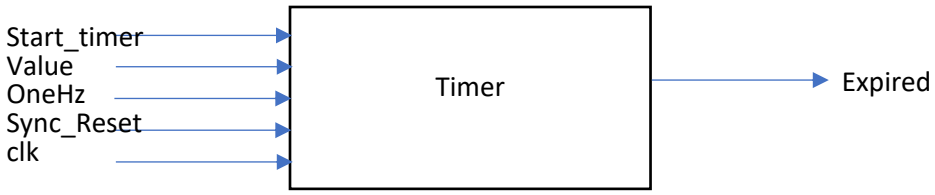

Figure 14: Timer block diagram

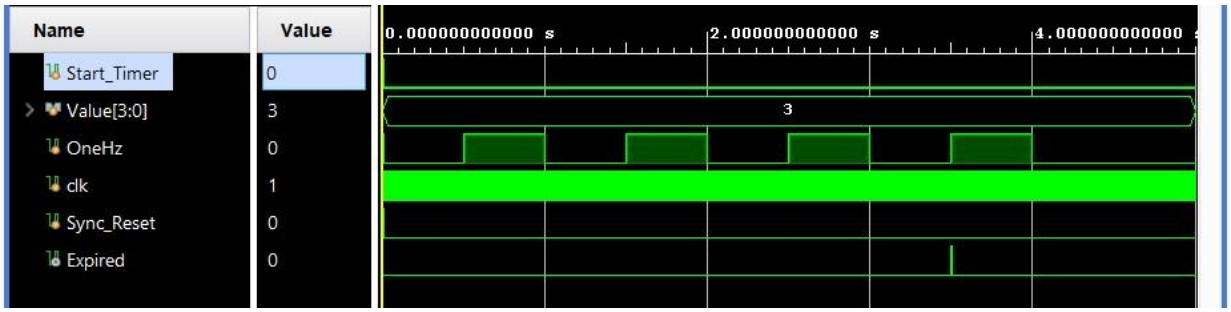

Figure 15: Timer simulation

Finite State Machine

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

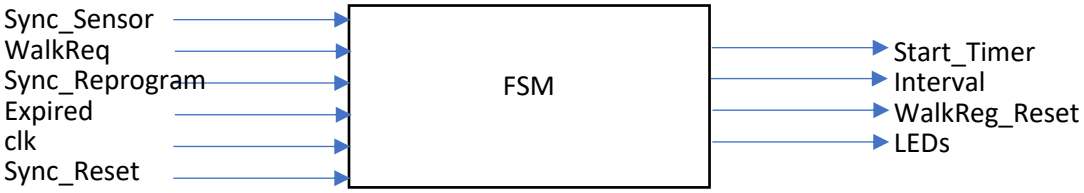

Figure 16: FSM block diagram

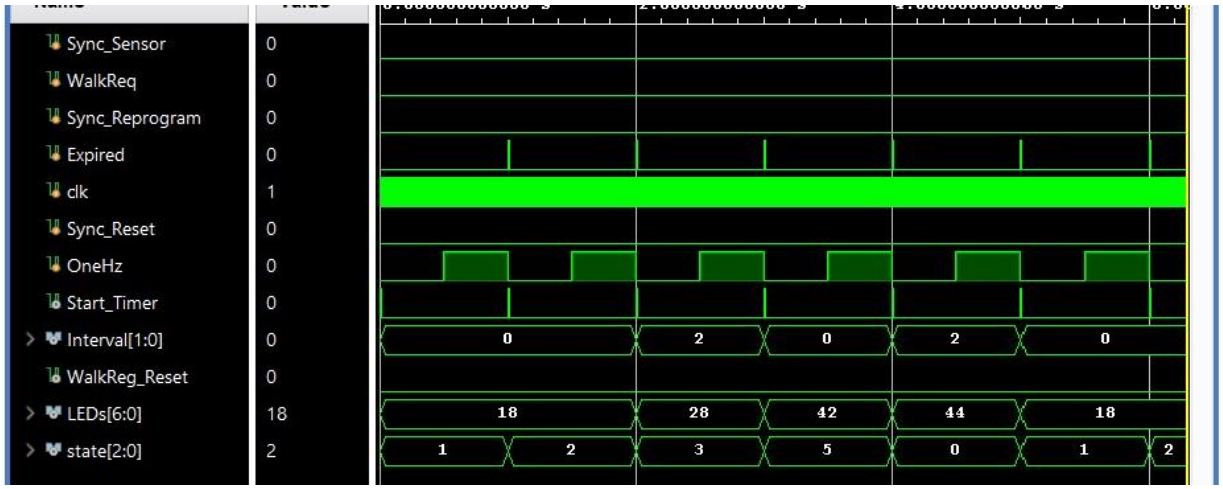

Figure 17: FSM simulation in normal behaviour

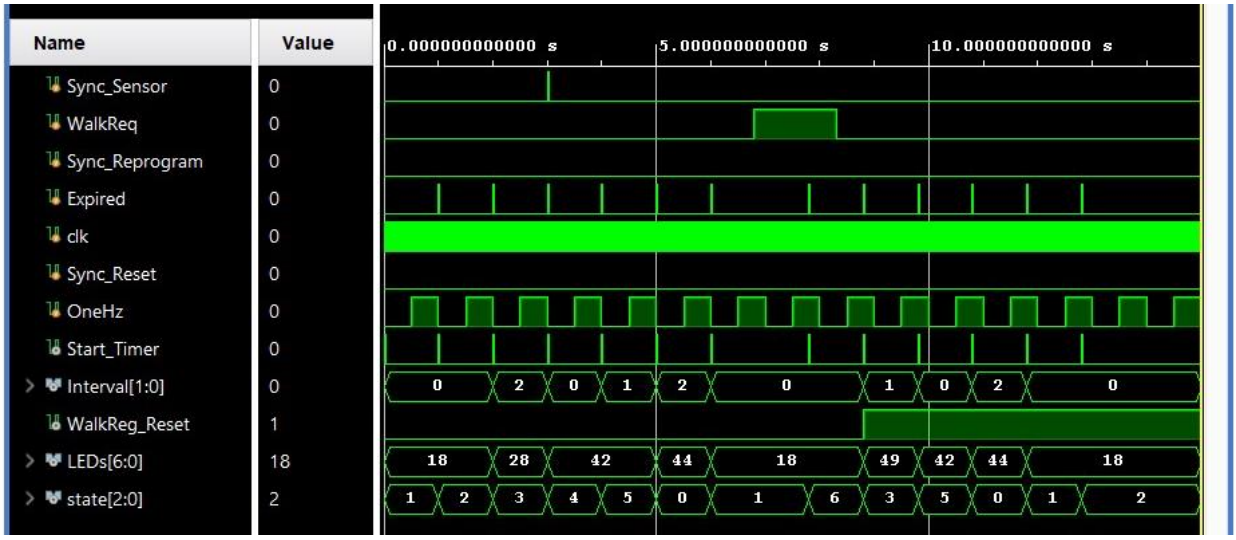

Figure 18: FSM simulation with a WalkRequest

Traffic Light Controller

This module assembles all the other components.

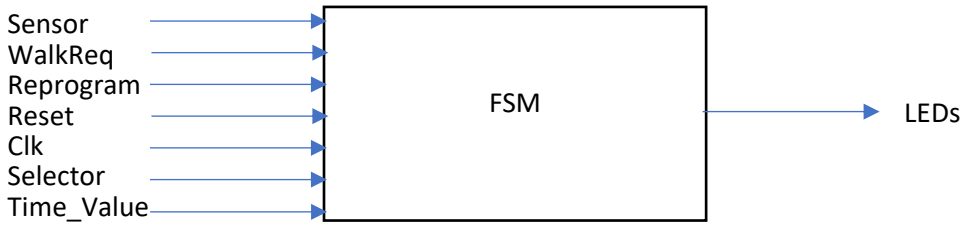

Figure 19: FSM block diagram

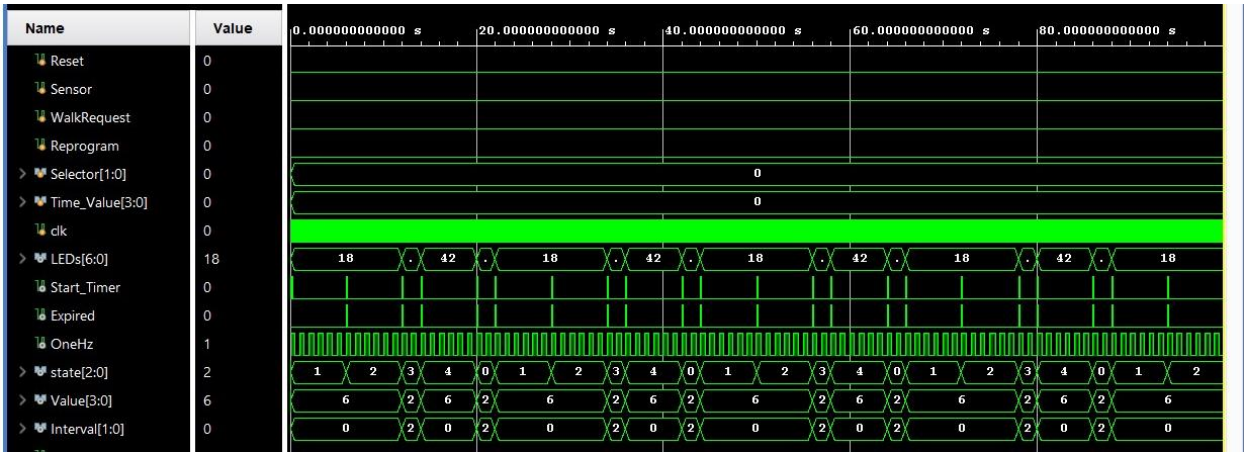

Figure 20: Normal process simulation

In Figure 21, a walk request is handled and after Main street green, Walk is set to on keeping other lights red.

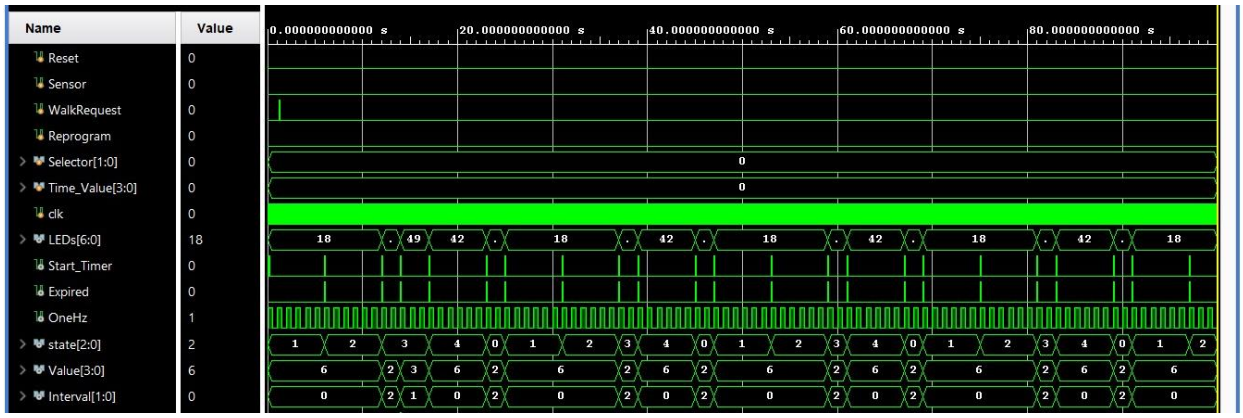

Figure 21: Walk Request

Sensor is detected to be high at 20 sec in Figure 22, and Side Street green is extended.

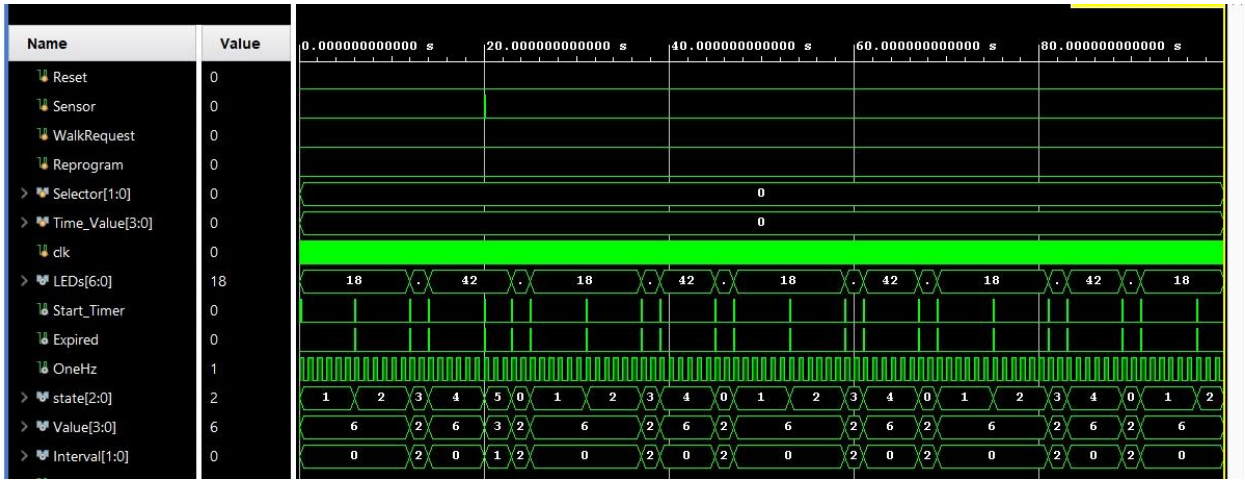

Figure 22: Vehicle sensor high, Side Street green extended