MECH 540B

Assignment 3:

Traffic Light Controller

Jonas Chianu 31298391

Aashkaran Dhillon 63377956

Scaling Factor: Aashkaran (1.0)

Jonas (1.0)

Prof: Christopher Sielman

UBC

March 19th, 2021

Abstract

A traffic light controller was designed for managing the traffic lights at the intersection with the goal of maximizing the traffic throughput given the vehicular traffic flow per minute equations for the Eastbound/Westbound for Road A and/Southbound/Northbound for Road B. The traffic system is also required to handle the pedestrian/bike crossing using the crossing lights that toggle between Walk/Stop. To formulate and simulate the logic, a state machine was built and implemented in Simulink using Stateflow to allow the operation of traffic lights while following the criterias provided by project stakeholders. 74HC4040-series counters with one second period input along with combinatorial logic involving MUX blocks were applied to seed calculated timer values for traffic light control for a predetermined schedule, which were determined from application of transportation theory. The results were presented and recommendations made.

Table of Contents

1. RCG

1.1. Goals:

The goal is to design the logic behind the new control scheme to maximize traffic throughput at the intersection. The control algorithm and sensor system must minimize the total amount of time that people wait at the intersection (assume a vehicle = 2 people).

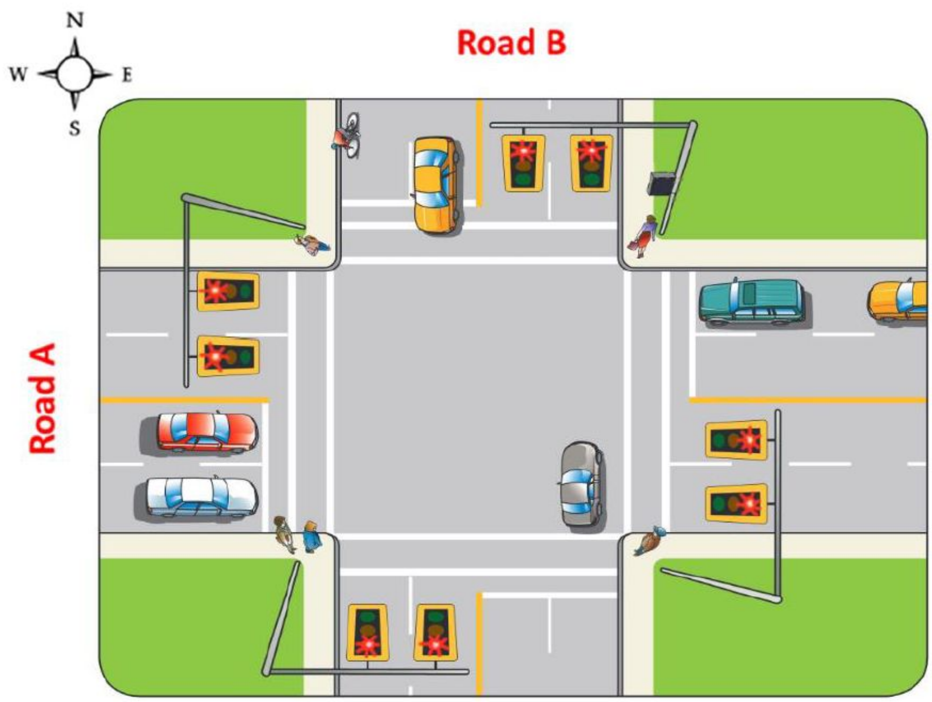


Figure 1: Intersection in consideration.

1.2. Requirements:

1. Roads A and B require a standard Red/Yellow/Green light in both directions and there are no left turn lights.
2. Pedestrians and bikes cannot wait more than 3 minutes for their crossing.

3. Pedestrian crossing lights at each corner that toggle between walk/stop must have at least 30 seconds to cross the street.
4. Once a traffic light turns green, it must remain green for at least 20 seconds.
5. The control algorithm and sensor system must minimize the total amount of time people wait at the intersection i.e maximize the traffic throughput at the intersection.
6. The design is failsafe. It is impossible for both a red and green light to be on at the same time, even if the controller is in an unknown state

1.3. Constraints:

1. All lights must be individually controlled (turned on/off).
2. The pretimed signal plan drive the traffic lights instead of
3. Limited by the bike/pedestrian push buttons being the only available sensor inputs.
4. Restricted to 12-bit counter 74HC4040 with 1 second period clock for the timing of logic.

2. Technical Solution

2.1. Traffic Management:

Vehicle traffic flow across the intersection can be modeled and simulated using the equations specified below, where h is hour of the day day and λ is average traffic per minute.

For Road A:

Eastbound: $\lambda(h)$= 2*max(sin(2*pi*(h−3)/12) + 2*sin(2*pi*(h−5)/36),0.1)

Westbound: $\lambda(h)$= 2*max(sin(2*pi*(h−4)/12) + 2*sin(2*pi*(h−2)/36),0.1)

For Road B:

Northbound: $\lambda(h)$= 0.8*max(sin(2*pi*(h−3)/12) + 2*sin(2*pi*(h−5)/36),0.1)

Southbound: $\lambda(h)$= 0.8*max(sin(2*pi*(h−4)/12) + 2*sin(2*pi*(h−2)/36),0.1)

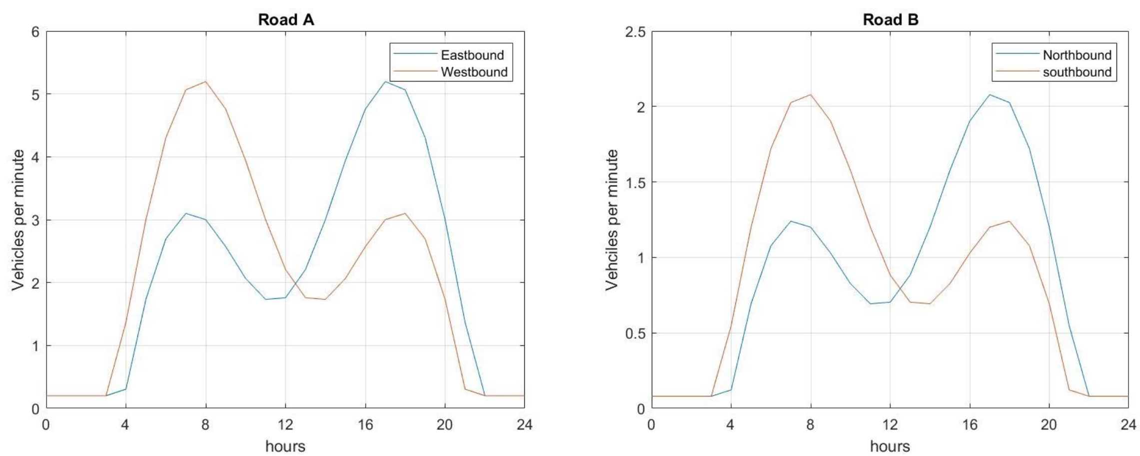


Figure 2: Model for Traffic flow across Road A and Road B at the intersection.

Similarly, pedestrian and bike traffic (north and south) is modeled by the equation:

$\lambda(h)$= 0.5*max(sin(2*pi*(h−7)/30),0)

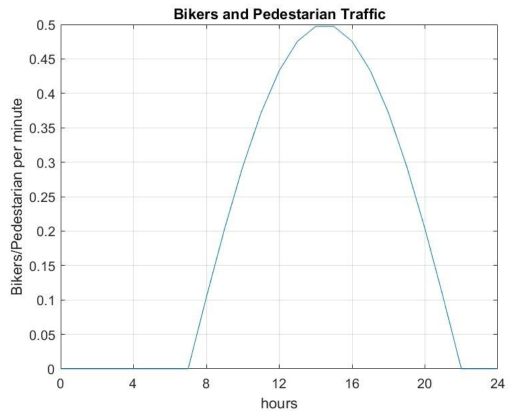


Figure 3: Model for Bike and pedestrian flow at the intersection

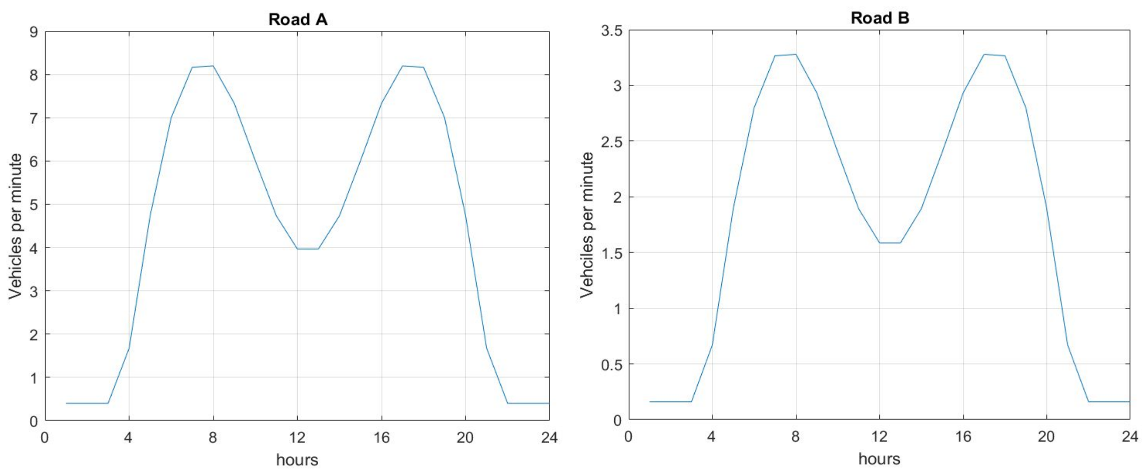


Figure 4: Model for Total flow for Road A and Road B

2.1. Timing Signal Analysis:

Example Calculation (for h=8:00):

C_{gr} =110 sec (cycle length for green and red)

(Green and red cycle length is chosen to minimize green light time as much as possible)

Saturation_flow = 2000 veh/h (Standard for most roads)

μ = saturation_flow / 3600 s/h = 0.5556 veh/s

Direction	Rate(λ) [veh/s]	Intensity (ρ=λ/μ)	X=λ/2(1-ρ)
EB	3 veh/min / 60 s/min	0.09	0.02747
WB	5.196 veh/min / 60 s/min	0.1559	0.05130
NB	1.2 veh/min / 60 s/min	0.036	0.01037
SB	2.078 veh/min / 60 s/min	0.06234	0.0184679

$$D_T(Total\ Delay) = X_{EB}r_{EB}^2 + X_{WB}r_{WB}^2 + X_{NB}r_{NB}^2 + X_{SB}r_{SB}^2, \quad r = red\ light\ time$$

$$r_{EB} = r_{WB}$$

$$r_{NB} = r_{SB}$$

$$r_{NB} = C_{gr} - r_{EB}$$

$$D_T = X_{EB}r_{EB}^2 + X_{WB}r_{EB}^2 + X_{NB}(C_{gr} - r_{EB})^2 + X_{SB}(C_{gr} - r_{EB})^2$$

$$D_T = (X_{EB} + X_{WB} + X_{NB} + X_{SB})r_{EB}^2 - 2C_{gr}(X_{NB} + X_{SB})r_{EB} + (X_{NB} + X_{SB})C_{gr}^2$$

To minimize total delay:

$$\frac{\partial D_T}{\partial r_{EB}} = 2(X_{EB} + X_{WB} + X_{NB} + X_{SB})r_{EB} - 2C_{gr}(X_{NB} + X_{SB}) = 0$$

$r_{EB} = \frac{C_{gr}(X_{NB}+X_{SB})}{(X_{EB}+X_{WB}+X_{NB}+X_{SB})} \approx 30\ sec, \quad g_{EB} = C_{gr} - r_{EB} \approx 80\ sec$

$r_{NB} = C_{gr} - r_{EB} \approx 80\ sec, \quad g_{NB} = r_{EB} \approx 30\ sec$

$D_T = 255.4\ veh - sec$

As it turns out when there are pedestrians at the intersection (7 < h < 22), the traffic times are around the same values as calculated above, so we can assume constant traffic timings between the duration of the day. The reason for this is due to the fact that the flow rate of traffic is much less than the saturation flow rate of 2000 veh/hr. We also made an assumption that there are always pedestrians at the intersection between 7<h<22. And so the minimum green light must be 30 sec in order to allow enough time for the pedestrian to cross the road.

However, during the times when there are no pedestrians (h <=7 and h>=22), the traffic times are less because the cycle times can be reduce without having to worry about the pedestrian requirements:

$r_{EB} \approx 20\ sec$

$g_{EB} \approx 55\ sec$

$r_{NB} \approx 55\ sec$

$g_{NB} \approx 20\ sec$

$D_T = 118.74\ veh - sec$

The minimum green time still needs to be 20 sec as per the requirements.

Note: Yellow lights time is based on the time required for an approaching car to safely stop based on its speed. The time is based on the ITE-proposed formula/chart for 60km/hr i.e. 5 secs

Note: The provided 30s time length for pedestrian/bike lights is assumed to include the actual crossing time plus the additional headroom for pedestrian safety, where pedestrians are warned before crosswalk lights toggle to STOP state. This pedestrian crossing time by rule of thumb in transportation theory is also assumed to be the minimum red light time for the crossing road (or minimum green light time for the adjacent road).

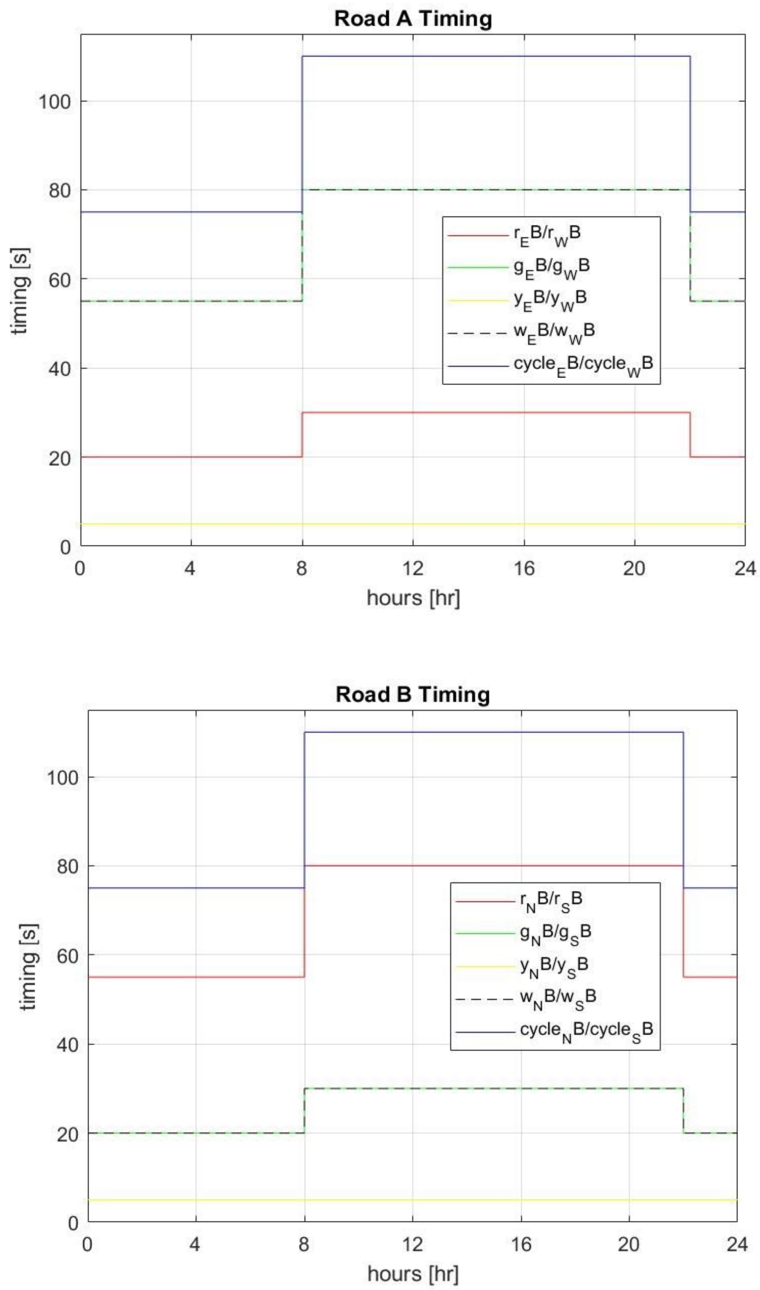


Figure 5: Traffic Light timing plan for Road A and Road B.

2.3. Sourcing Timings from Counter:

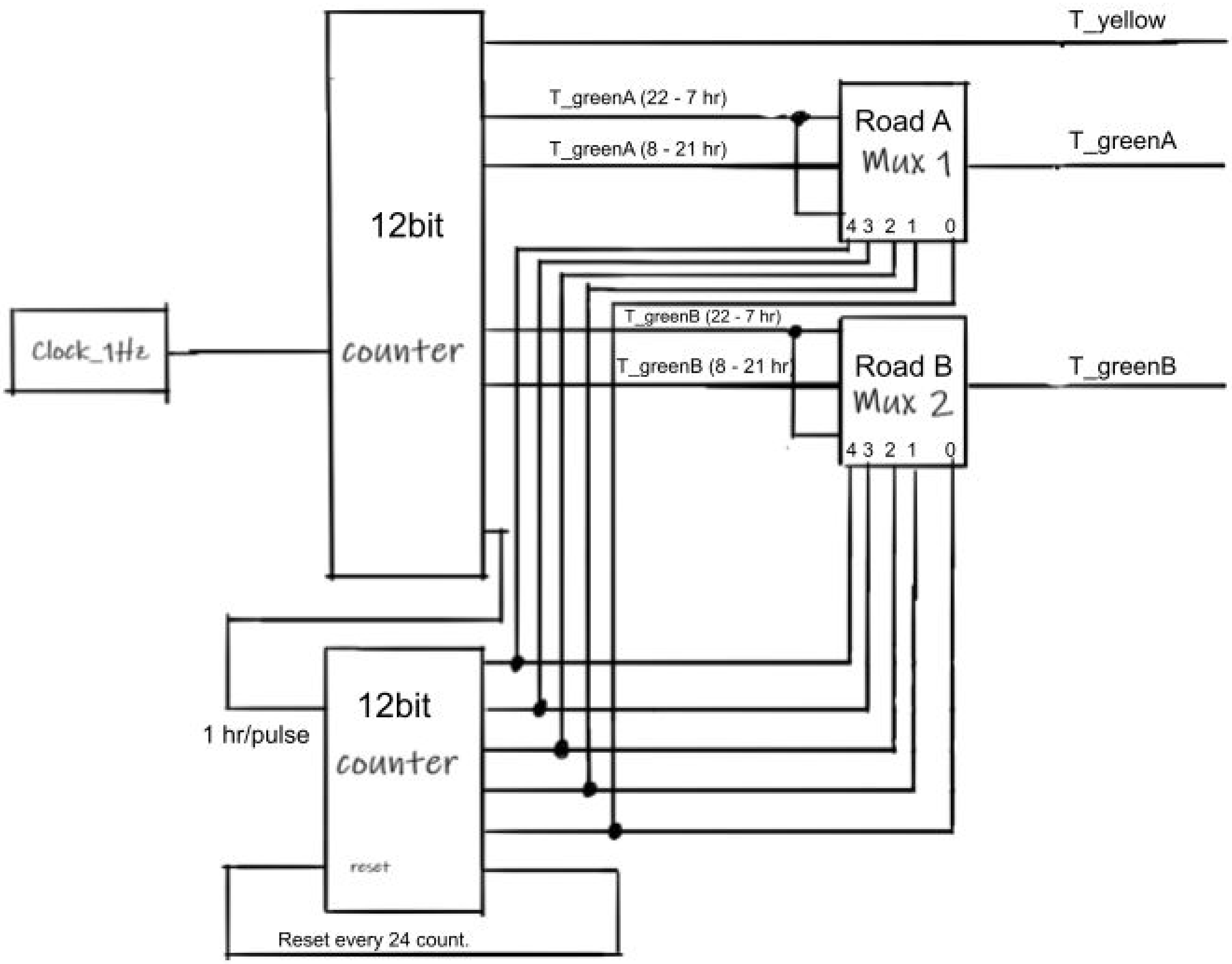


Figure 6: Layout for sourcing scheduled time values for the traffic controller by the hour

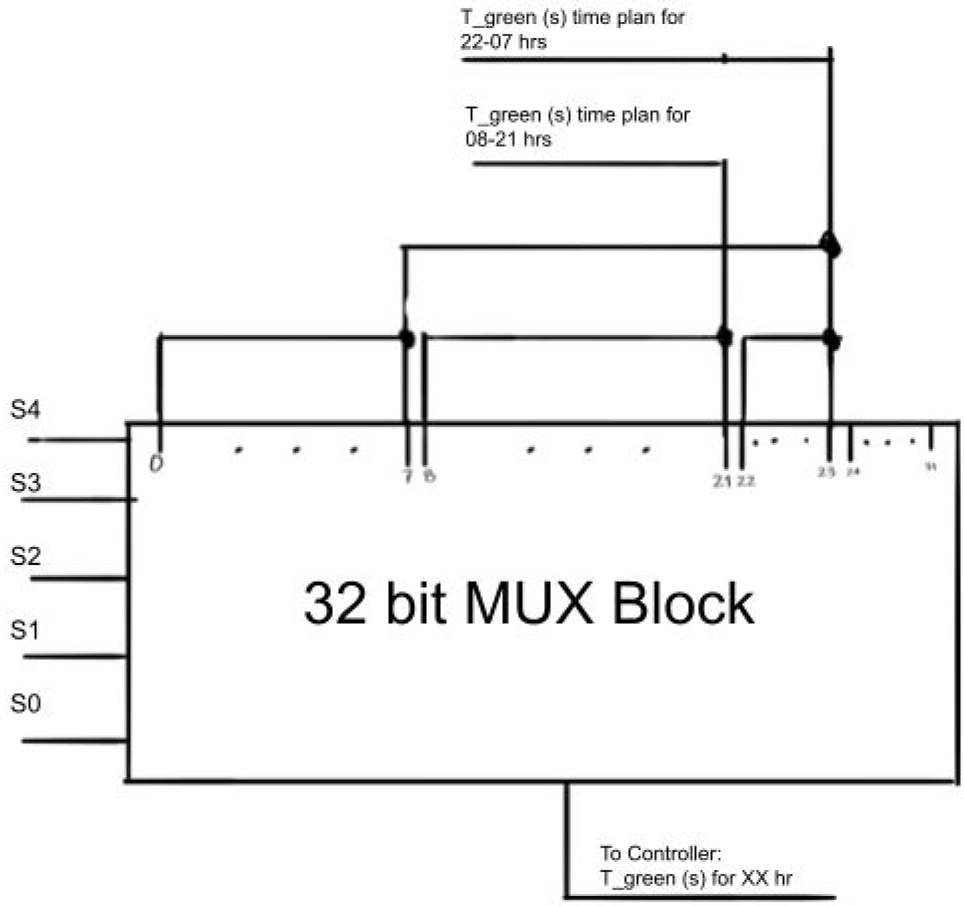


Figure 7: Implementation of 32bit - 5 bit select MUX block to output green light timer values for the pre-selected hours of the day.

To seed the fixed Yellow light timings of 5s and signal plan for green lights time for Road B and Road A at different hours of the day, we use a pair of counters and MUX logic. Through frequency division of 1Hz at first counter, we get T_yellow value of 5s.

The green light times for Road A & Road B are different at times slots 0-7hrs, 7-22hrs and 22-0hrs of the day. These timer values are sourced from the first 1Hz input counter as shown in the table below. Next the timings flow into the 32bit MUX blocks (as shown in figure) and occupy 0-23 bits, where the 5 bit Select inputs updates the signal time plan to be output for the given hour, at every hour, to go to the logic controller.

Bus Line	**Bits used**
T_greenA (22 - 7 hr)	$2^5 + 2^4 + 2^2 + 2^1 + 2^0$
T_greenA (8 - 21 hr)	$2^6 + 2^4$
T_greenB (22 - 7 hr)	$2^4 + 2^2$
T_greenB (8 - 21 hr)	$2^4 + 2^3 + 2^2 + 2^1$
T_yellow	$2^2 + 2^0$
1 hr/pulse counter	$2^{11} + 2^{10} + 2^9 + 2^4$
Reset	$2^4 + 2^3$

The select bits of MUX blocks (one each for RoadA/B traffic light) upcount every hour from 0 to 23 to allow selection of a corresponding signal plan for the hour. This is made possible with the 2nd counter whose clock is fed by the first counter's frequency division output that produces 1 clock pulse/hour. This allows the 2nd counter to use 0-4 bits to do the 24hour counting that helps select the hourly plan at MUX blocks. This counter self auto resets at the end of the day, when the 24hr count is reached, thereby allowing for automated functioning.

The issue with this approach is that the quartz, which seeds the times for master clock, tends to drift over time. This requires routine maintenance for a hard reset. A solution to mitigate this issue would be to subscribe to a 3rd party GPS run clock, as it is done by the civil bodies for reliable functioning of traffic lights.

2.3. Inputs and Outputs:

The following are the I/O

Type	Definition	Tag	Value/State	Port
Input	Timer Green Light Road A	T_greenA	55s/80s	1
Input	Timer Green Light Road B	T_greenB	20s/30s	2
Input	Time Yellow Light	T_yellow	5s	3
Input	Road A Push Button	walkA	1/0	4
Input	Road B Push Button	walkB	1/0	5
Output	Red Light Northbound	RNB	1/0	1
Output	Yellow Light Northbound	YNB	1/0	2
Output	Green Light Northbound	GNB	1/0	3
Output	Red Light Southbound	RSB	1/0	4
Output	Yellow Light Southbound	YSB	1/0	5
Output	Green Light Southbound	ESB	1/0	6
Output	Walk Road B	WB	1/0	7
Output	Red Light Eastbound	REB	1/0	8
Output	Yellow Light Eastbound	YEB	1/0	9
Output	Green Light Eastbound	GEB	1/0	10
Output	Red Light Westbound	RWB	1/0	11
Output	Yellow Light Westbound	YWB	1/0	12
Output	Green Light Westbound	GWB	1/0	13
Output	Walk Road A	WA	1/0	14

2.4. Finite State Machine:

The state machine for the behaviour of the control system is implemented below using Stateflow in Matlab, which also allows for implementation of logic directly using the states. There are a total of 6 states that take 5 inputs to drive 14 outputs as shown in the I/O table.

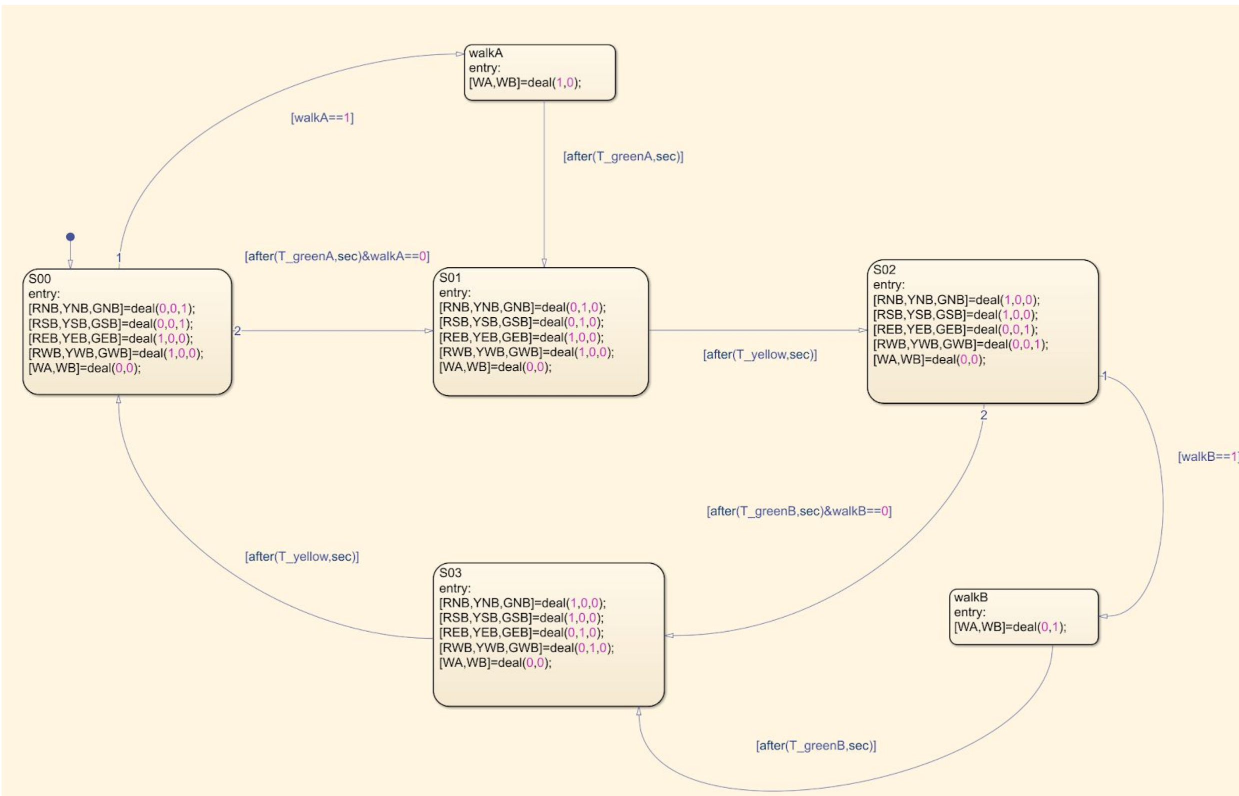
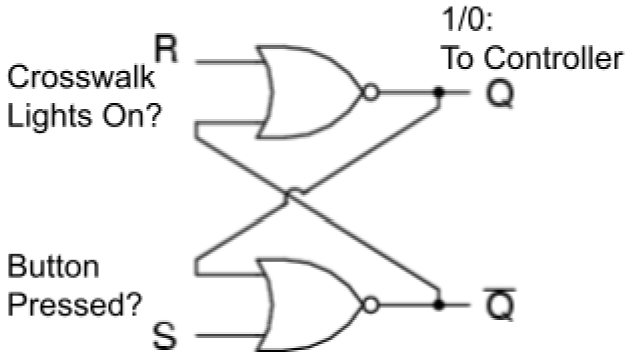


Figure 8: State Machine

Note: For convenience, the pedestrian/bike crossing push button will use SR latch to latch onto the TURN ON bit when button is pushed and stay there until a reset, which is triggered when the Walking Light for that crossing is turned on. This converts our pushbutton into an auto-resettable switch, whose status is read only once per traffic light cycle by the controller to (or not to) turn on the crossing lights when requested (set) by the pedestrian/biker.



S	R	Q	$\overline{Q}$
0	0	latch	latch
0	1	0	1
1	0	1	0
1	1	0	0

Figure 9:A typical SR latch

As shown in the FSM above, the default state (S00) for FSM is the one where Road B traffic lights turn GREEN to allow for Northbound and Southbound traffic flow. Since, the Road A traffic lights are by logic inverse of Road B, the state S00 also begins with RED Lights turned on at Eastbound and Westbound. At the state (S00), the controller detects if any push buttons on Eastbound or Westbound (Road A crossing) were pressed, if YES, then the walkA state i.e the two Road A walking lights are turned ON(WALK) to allow pedestrians/bike to cross Road A before heading into State S01 in which crosswalk lights turn OFF(STOP) and Northbound/Southbound Road B traffic lights go RED to YELLOW in T_greenB seconds. If the pedestrian/bike button isn't pressed, the S00 to S01 state is directly transverse after T_greenB seconds.

After time of T_yellow seconds in state S01 (i,e, 5s), the Northbound/Southbound (Road B) traffic lights go from YELLOW to RED and simultaneously the Eastbound/Westbound (Road A) traffic lights go RED to GREEN to initiate the traffic flow across Road A. Again, the controller checks if any of the push button was flagged high at Road B crossing, if YES, the system transverses to walkB state where pedestrian lights turn ON(WALK) at

Northbound/Southbound Road B crossing before heading to state S03 in T_greenA seconds. In state S03, the Eastbound/Westbound Road B traffic lights go RED to YELLOW and crosswalk lights turn OFF (STOP). If the pedestrian/bike cross push button was not pressed for Road B crossing within the traffic light cycle, the S02 to S03 state is directly transversed in T_greenA seconds.

Finally, after T_yellow seconds (5s) spent in state S03, S01 state is activated in which the Eastbound/Westbound traffic lights turn YELLOW to RED and simultaneously Northbound/Southbound traffic lights turn RED to GREEN. Thereby, completing one full traffic cycle time for the intersection. This goes on in a loop.

Note: We disregarded the state where all the traffic lights need to be Red for a brief time owing to the fact that the vehicle traffic per hour on the road is very small compared to the saturated flow capacity of the road, plus the speed limit is just 60km/hr with sufficient yellow light time. Also because, there are no left turns allowed, which mitigates any safety concert which causes the implementation of this brief delay time where all traffic lights are momentarily red. This causes the simplification of the logic and its implementation.

Note: The time allowed for pedestrians/bikers to cross the road is equal to time the other opposite road traffic lights are on green, This crosswalk time is required to be a minimum of 30seconds, and this is assumed to factor in the phase time in which the crosswalk lights warns users to stop crossing for safety, before changing to STOP.

2.5 Simulation Setup:

For simulating the state machine logic, we use Stateflow in Simulink. As shown in the figure below, we have our 5 inputs and 14 outputs consisting of push buttons, timers and traffic light outputs. For simulation, we have kept a common push button for each road, which at any of the crossing is set by the user and latched until an auto-reset from the pedestrian lights is turned on. We also use one traffic right each to represent Southbound/Northbound traffic at Road B and Eastbound/Westbound traffic at Road A, since the opposite traffic lights mirror each other.

From the 12bit counter, the state flow takes in optimized green light timer values for each Road along with the constant yellow light a time values to initiate the state machine transitions. The pedestrian crossing time (minimum 30s) is taken to be equal to the time the traffic light is green on the adjacent road. Here, we use switches to mimic the input from the push buttons at the four intersection crossings. The current screenshot in the image below represents the state when Eastbound/Westbound traffic is flowing across Road A and the Northbound/Westbound traffic is stopped, with requested pedestrian/bike crossing happening across it.

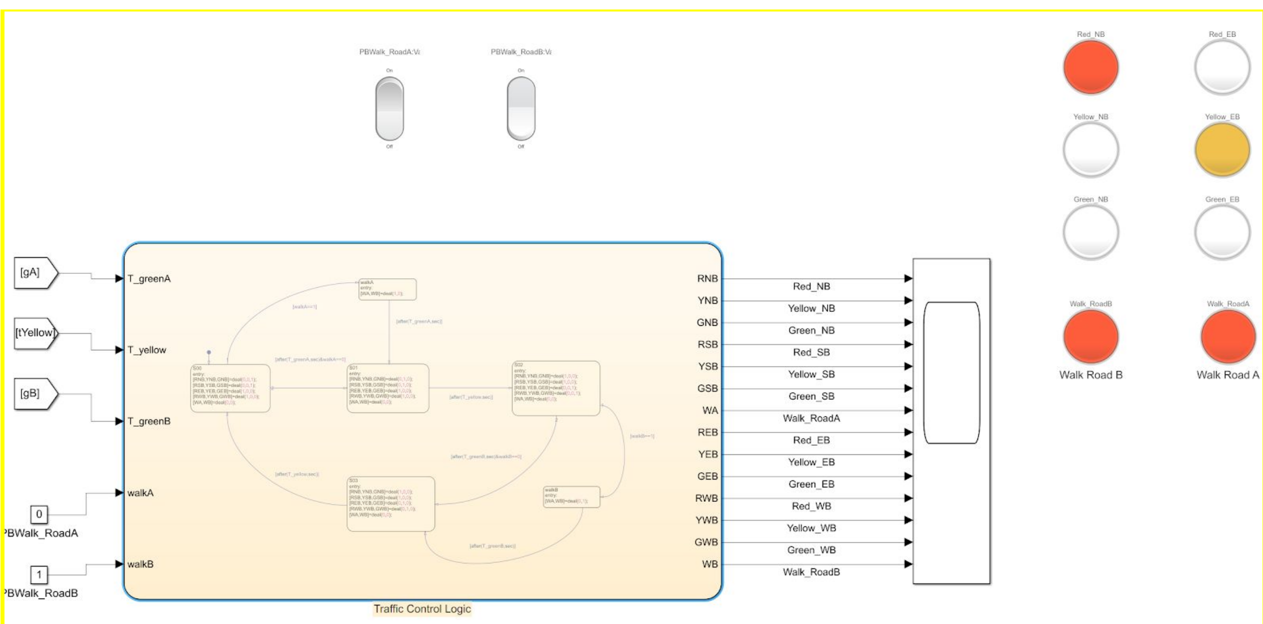


Figure 10: Implementation of State Machine for traffic light control

2.6. Simulation Results:

Presented below in the split figure 8 a) & 8 b) is the simulated scenario of Road A and Road B Traffic light control for 480s duration with T_greenA = 80s, T_greenB = 30s, T_yellow = 5s, which is scheduled during 08-21 hours of the day.

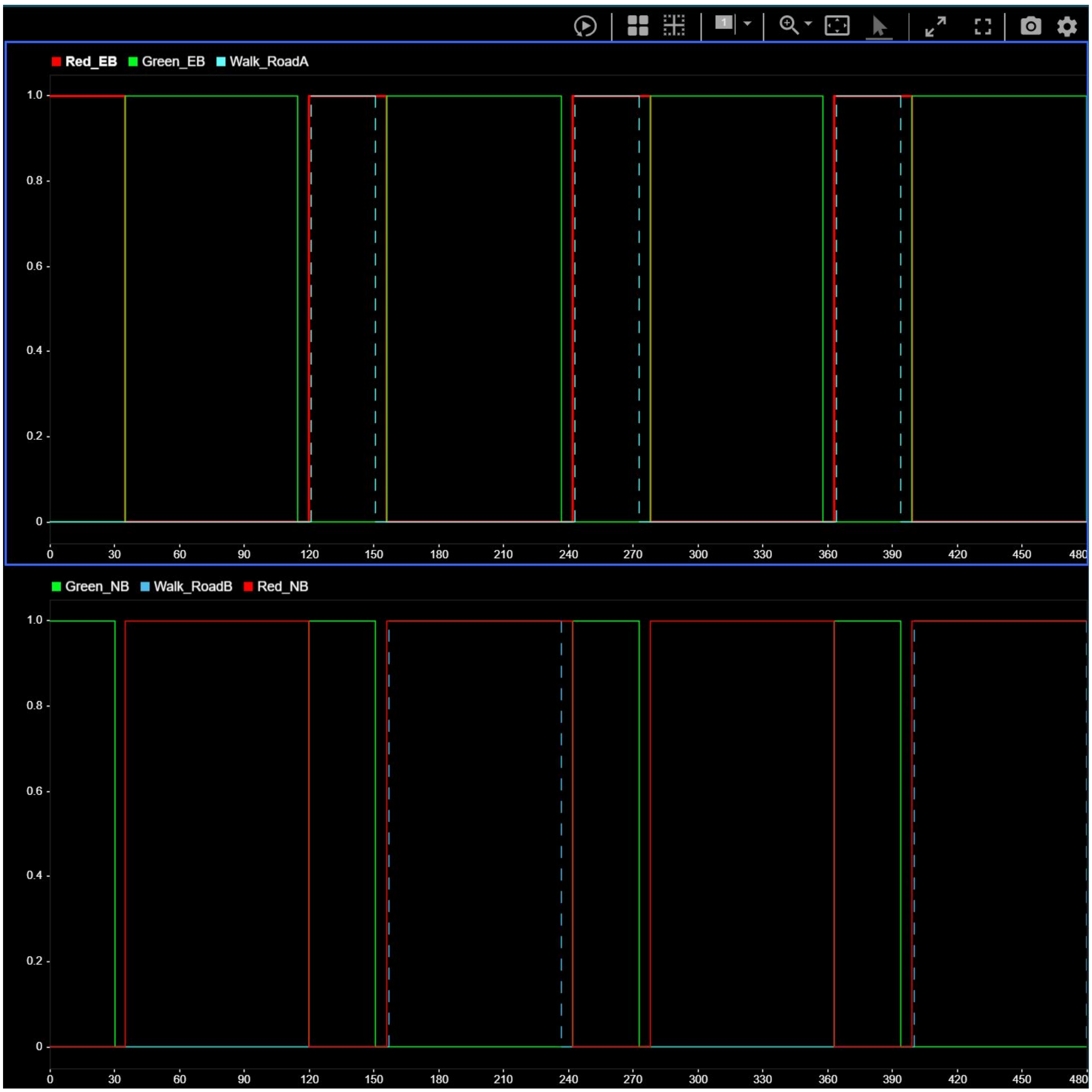


Figure 11: Traffic Light Controller Simulation 0-480s. Road B(Top) Road A(bottom).

As shown, the Traffic Light Controller behaviour works exactly as desired through the state machine design. It's responsive to the pedestrian inputs and the logic is robust for the real-life implementation for any desired traffic light cycle times.

Note: Since the traffic lights and pedestrian lights/push buttons at Northbound / Southbound (Road A) and Eastbound / Westbound (Road B) mirror each other, we used one traffic light from each road to represent simulated output of logic.

3. Conclusion

In this report, we analyzed the provided models for the pedestrian and vehicular traffic/min flow across south,north,east & westbound lanes to determine via application of Transport theory, the traffic light timings that allow for the most optimal flow of traffic, while following the client's requirements. It was deduced that we used Road A and Road B green traffic light values of 80 sec and 30 sec respectively during 08-21hours and Road A and Road B green traffic light values of 55 sec and 20 sec respectively during 22-07hours.

The desired traffic light timer values are seeded via frequency division from the 1hz input using the 12bit counter 74HC4040 and the hourly selection of the desired green light time values is made possible with the additional implementation of combinational logic involving an hour counter coupled with a couple of 32bit MUX blocks. Since, we schedule timer values dependent on the hour of the day, we highly recommend the client to invest in better options for the timing, such as 3rd party GPS timers as the quartz clock tends to drift over time and needs a manual reset, which accumulates to maintenance cost. Also, having GPS timers from a master clock is the commonly used method for modern day Traffic Lights and worth the additional costs.

Furthermore, a state machine was successfully developed and then implemented for the postulated operational logic of Traffic Light Controller at the intersection. The controller was designed in Stateflow in Simulink to simulate real life scenarios. Although the state machine works perfectly as desired in the software simulations, we recommend a thorough hardware implementation and testing of the control logic on a scaled down prototype using the 74HC4040 counter output to seed the timers.