Traffic Management System

Navneet Agarwal - 140100090
Sanat Anand - 140040005
Sohum Dhar - 140070001
Tanmay Parekh - 140100011

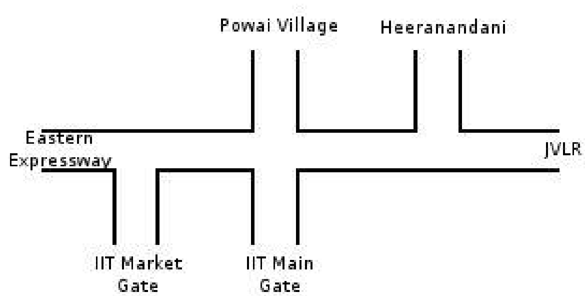
February 2016

1 Objective

We aim to design a traffic management system for the three traffic lights which impact IITB. We will control the three traffic lights : YP gate, Main Gate and the Pizza Hut junction. Using the average traffic observed on different roads connected to these junctions, we can figure out the optimal timings for the different lights in order to prevent infinite queue at any of the lights.

2 High Level Architecture

We have three cross roads to manage :



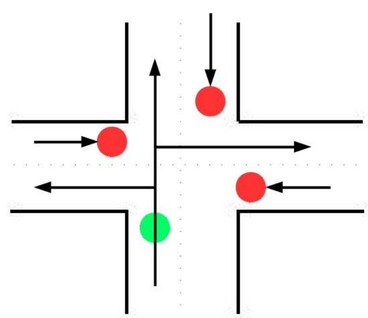
- YP junction - This joins the YP gate road entering IIT to the main JVLR-Eastern Expressway road. So this has essentially three roads meeting each other. Thus there can be a total of 6 types of flows of traffic (from every road to every other road). There will be a total of three traffic lights one for each of the roads.

- Main gate junction - This connects the JVLR-Eastern Expressway road to the road entering IIT main gate and the road going to Powai village. So this has essentially four roads meeting each other. Thus there can be a total of 12 types of flows of traffic (from every road to every other road).There will be a total of four traffic lights one for each of the roads.

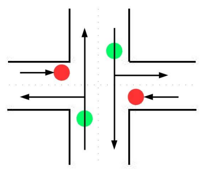
- Pizza Hut junction - This joins the road entering Heeranandani to the main JVLR-Eastern Expressway road. So this has essentially three roads meeting each other. Thus there can be a total of 6 types of flows of traffic (from every road to every other road). There will be a total of three traffic lights one for each of the roads.

To manage one crossroad we can adopt several strategies. We must ensure that two traffic flows must not cut each other at the same time. Broadly we can do two kind of things :

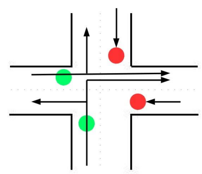
1. At one time we allow traffic flows incoming from one particular road and block all the other incoming roads i.e we allow all the three flows of traffic (left, straight, right) from one of the roads and the other roads are allowed only the default left traffic flow (which does not block any other flow). The following diagram illustrates it :



2. The next strategy is to allow multiple flows from different incoming roads in such a way that the heavy flows are given more time as compared to the lighter flows. Although this would be a little more involved than the previous strategy, it can give better results because it can allocate more time to heavy flows. Example : Say there is heavy traffic going from south to north and north to south but very little going into east and west roads. As per the previous strategy here we cannot let both the N-S traffic



(a) Straight N-S flow



(b) Right flow from S and straight from E

and the S-N traffic flow at the same time because only one incoming road would be open.

3 Description of functionality of the blocks

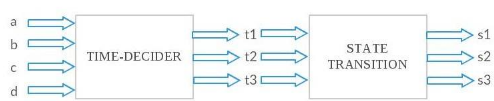
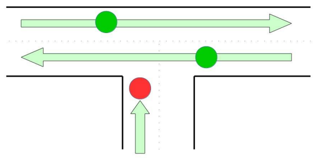


Figure 1: Block Diagram for three way junction

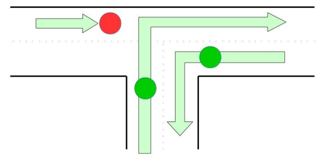
We can implement the traffic management system for three way junctions by implementing two modules. One to compute the time for multiplexing and the other to compute the state transitions of that traffic light based on the output(time slots) generated by the previous block.

Consider the 3 way case. The input to the compute-time-block are the backed up traffic at the 3 ways of the junction and the speed of vehicles at the junction. The output of the junction is valid only at the points of time when the second block is in the start/idle state, i.e. one cycle is completed(based on previous computation). The output of this block is the time slots for the various states corresponding to the states of the various traffic lights facing the 3 ways of the junction. The output of this block is fed into state-transition-block.

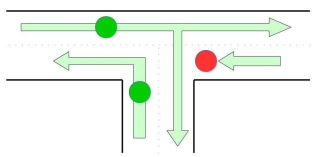
The state-transition-block simulates the states of the traffic lights for the 3 ways on the junction. For the traffic control system we have designed 3 states, such that traffic is allowed to move from any way to any other in atleast one state. The state-transition-block allots time to these states based on the inputs from the compute-time-block. After completion of every cycle the state-transition-block sets the time slots for the 3 states based on the input from the compute-time block. The state-transition-block gives output as the state of the traffic lights represented by 3 bit vectors(representing various states of signals)



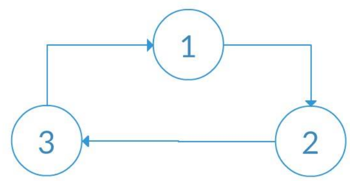
(a) State 1



(b) State 2



(c) State 3



(d) State Transition Diagram for three way

The compute-block solves the following inequalities:

$$X - 6v(t_1 + t_3) + 120(t_1 + t_2 + t_3) \leqslant 0$$

$$Y - 5v(t_1 + t_2) + V_x(t_1 + t_2 + t_3) \leqslant 0$$

$$Z - 3v(t_2 + t_3) + 15(t_1 + t_2 + t_3) \leqslant 0$$

Where X, Y, Z are the number of backed up cars in clockwise order from left in figure, t_1, t_2, t_3 are the time slots allotted to the 3 states, v is the speed of the vehicles on the junction and V_x is the expected flow of traffic from the east, for the 3 way junction. Here, V_x is calculated by the compute-block as the average flow rate of traffic from east, over previous cycle, i.e.,

$$V_x = 5v^{previous}\frac{t_1 + t_2}{t_1 + t_2 + t_3} + \frac{X^{present} - X^{previous}}{t_1 + t_2 + t_3}$$

Here t_1,t_2,t_3 were computed in the last iteration

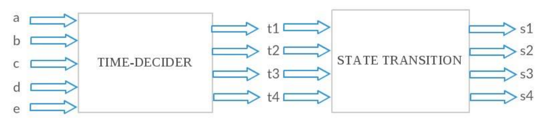


Figure 2: Block Diagram for four way junction

Similarly, we can implement the blocks for 4 way junction, where we have 5 states for the 4 traffic lights, one for each way.
We'll have inequalities for the 4 way junction with 5 states and 5 time slots. The compute-block for the 4 way junction solves the following inequalities:

$$X - 6v(t_1 + t_2) + V_x(t_1 + t_2 + t_3 + t_4 + t_5) \leqslant 0$$

$$Y - vt_4 + 5(t_1 + t_2 + t_3 + t_4 + t_5) \leqslant 0$$

$$Z - 5v(t_1 + t_5) + V_z(t_1 + t_2 + t_3 + t_4 + t_5) \leqslant 0$$

$$W - vt_3 + V_w(t_1 + t_2 + t_3 + t_4 + t_5) \leqslant 0$$

Where X, Y, Z, W are the number of backed up cars in clockwise order from left way in figure, t_1, t_2, t_3, t_4, t_5 are the time slots allotted to the 4 states, v is the speed of the vehicles on the junction and V_x, V_z, V_w, with $V_w \geqslant 8$, is the expected flow of traffic from west, east, south respectively, for the 4 way junction. (Here, west and east bound traffic, which are more important, are given more time and states to free up the traffic better.) Here, V_x, V_z, V_w are computed by the compute-block as average flow rates of traffic from west, east and south, over the previous cycle, i.e.,

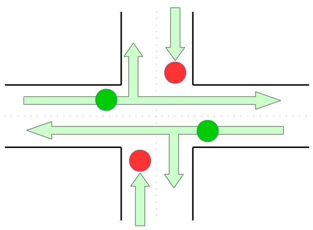
$$V_x = 6v^{previous}\frac{t_1 + t_2}{t_1 + t_2 + t_3 + t_4 + t_5} + \frac{X^{present} - X^{previous}}{t_1 + t_2 + t_3 + t_4 + t_5}$$

$$V_z = 5v^{previous}\frac{t_1 + t_5}{t_1 + t_2 + t_3 + t_4 + t_5} + \frac{X^{present} - X^{previous}}{t_1 + t_2 + t_3 + t_4 + t_5}$$

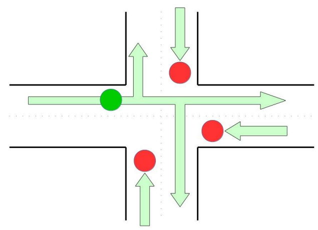
$$V_w = v^{previous}\frac{t_3}{t_1 + t_2 + t_3 + t_4 + t_5} + \frac{X^{present} - X^{previous}}{t_1 + t_2 + t_3 + t_4 + t_5}$$

Here t_1,t_2,t_3,t_4,t_5 were computed in the last iteration

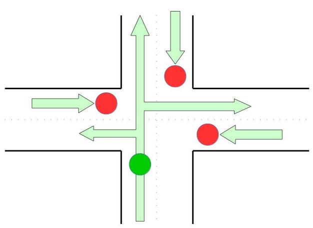
Also, pedestrian movements were factored in while construction of the states. The states are such that there is atleast one state allowing pedestrians to cross each road-way on the junction, as in 3 way case, or two or more states together allow the pedestrians to cross the roadway, as in the 4 way case. The pedestrian crosses the road in atmost 2 parts.



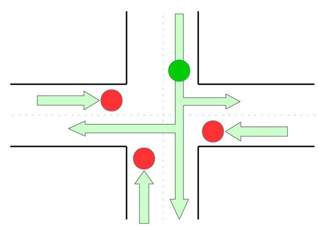
(a) State 1



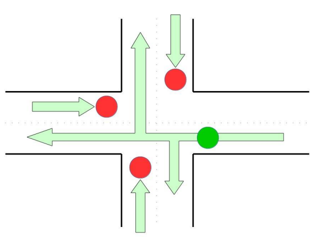
(b) State 2



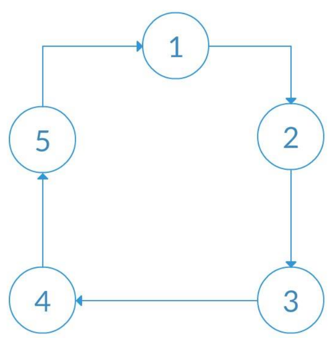
(c) State 3



(d) State 4



(e) State 5



(f) State Transition Diagram for four way

4 Plan for testing and verification

- We will create a test-bench which supplies the above described blocks with the backed up rows for each junction along with the average speed on each junction as input.
- The test bench will compute the backed up rows of traffic as the simulation proceeds based on the output of state of the signals at every junction and after every cycle this information is supplied as input to the above described traffic-light-control blocks.
- For testing we try out different traffic flows and see how the blocks behave and monitor the number of backed up rows on each junction. For example, we modulate the traffic flow directions and the traffic values(which are not specified) and run the system.
- Corner cases include the traffic flows where some way on the junction is assigned low traffic or very high traffic flow arbitrarily. Since, the traffic flow being high or low has direct impact on the number of backed up rows, they have been accounted for in the model. If number of backed up cars is more on any side that side is given more time to clear the traffic. Similarly, if the number of backed up rows is less and the traffic flow is less as well, then that way on the junction is allotted lesser time.
- We monitor the number of backed up rows as computed by the test bench after every cycle and make sure that it is under some bound. We expect the number of backed up rows to decrease with time implying good traffic management.

5 Assumptions and Constraints

- The net traffic going into the system equals net traffic going out. The test bench suitably assigns traffic flows within the system(based on lanes, etc.) such that the above constraint is met.
- The traffic going into Powai vilage, market gate, main gate and Heeranandani is taken to be negligible in comparison to the Eastern Expressway and JVLR traffic. And hence the calculations are done accordingly.
- For sides on junctions where number of cars coming in is not specified, in the test bench we assume some reasonable values, such that net traffic going in equals net traffic coming out.
- We assume that when traffic light is green all of the lanes are being used to free up the traffic.

6 Distribution of work among team members

Although all of the team members will contribute in every aspect of the project and all of us will take responsibility for all parts of the project, we have decided that these things will be majorly seen by these people:

- High Level Architecture - Sanat, Navneet
- VHDL coding - Sanat, Tanmay, Sohum
- Test bench - Sohum, Navneet
- Hardware Implementation - Sohum, Navneet
- Presentation and report making - Sanat, Tanmay