# Computer Architecture Project
# Traffic Controller

Larsen Friis
104127870

Dr. Imran Ahmad

03-60-265-01

November 29, 2016

## Design Procedure

### Specifications

For this project, we are to design a Simple Traffic Controller. This will consist of two roads, one running North-South (NS) , and one running East-West (EW). The controller with have four traffic lights:

- a traffic light controlling the NS Road for North bound traffic
- a traffic light controlling the NS Road for South bound traffic
- a traffic light controlling the EW Road for East bound traffic
- a traffic light controlling the EW Road for West bound traffic

Each light will cycle through:
RED -> GREEN ARROW -> YELLOW -> GREEN -> YELLOW -> RED

8 + 2 + 1 + 4 + 1 + reset state = 16 seconds

Time assumptions for lights:
RED = 8 seconds
GREEN = 4 seconds
GREEN ARROW = 2 seconds
YELLOW = 1 second

### Formulation

We have 10 potential states in this problem. Each road consists of two directions, but will be using the same lighting sequences. But as one lighting directions lighting sequences are executing, the other road also has a lighting sequence. Each sequence is a complement of one another. So when one direction is executing, the other will be red and vise-versa.

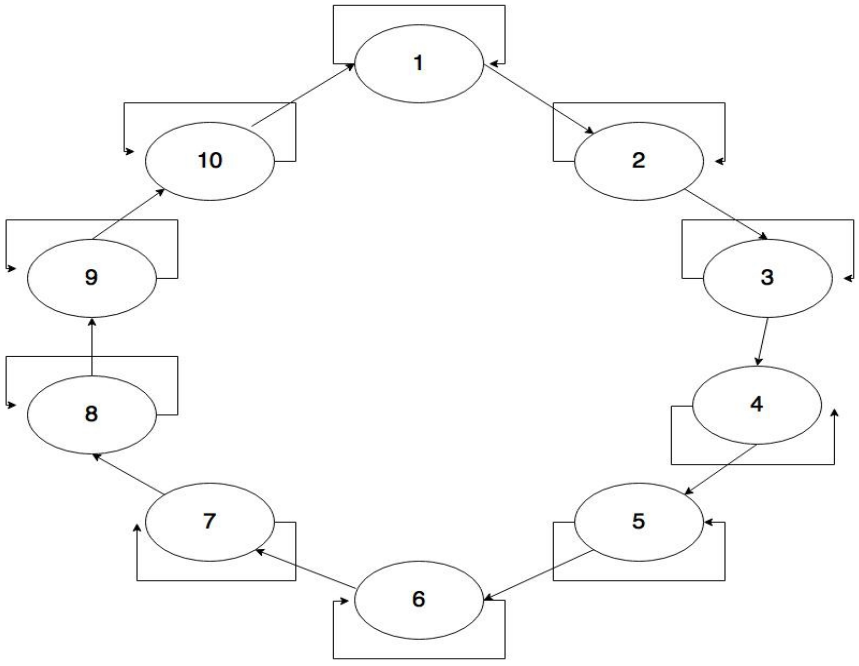
**Table 1:** Representation of States

| STATE | North-South (on) | East-West (on) | Delay |
|---|---|---|---|
| 1 | RNS | REW | 0 |
| 2 | RNS | GAEW | 2 |
| 3 | RNS | YEW | 1 |
| 4 | RNS | GEW | 4 |
| 5 | RNS | YEW | 1 |
| 6 | RNS | REW | 0 |
| 7 | GANS | REW | 2 |
| 8 | YNS | REW | 1 |
| 9 | GNS | REW | 4 |
| 10 | YNS | REW | 1 |

**Table 2:** Binary Representation of States

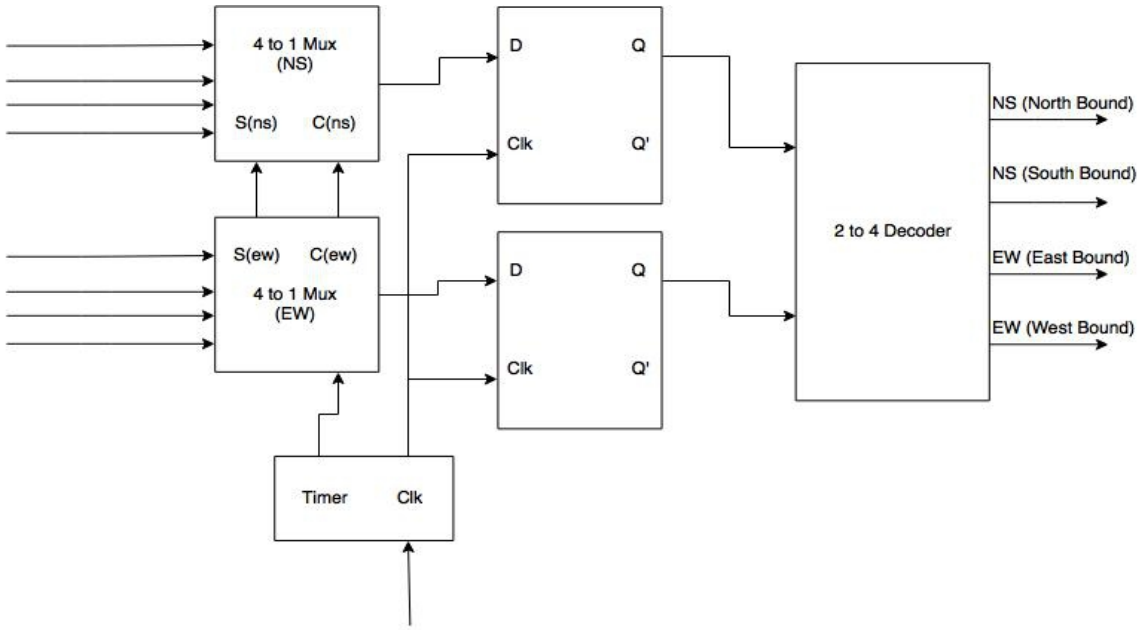
| STATE | RNS | YNS | GANS | GNS | REW | YEW | GAEW | GEW |
|---|---|---|---|---|---|---|---|---|
| 1 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 |
| 2 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 |
| 3 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 0 |
| 4 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |
| 5 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 0 |
| 6 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 |
| 7 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 0 |
| 8 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 0 |
| 9 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 |
| 10 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 0 |

At each state, there will only be two lights on, depending on which lights are on in the cycle will determine its complement. For example, when the EW light is going through the cycle, the NS light will remain red until the cycle is complete. When at a state, we stay in that specific state until the length of time we are delayed is reached. After, we move to the next state and so on. Once all the states have been visited, the states will be reset and the process will continue.



Now, we need a time to execute the proper lighting for traffic at each state for the proper length of time. We will need a 4-bit binary counter that counts for a total of 16 seconds (0000-1111) for the lighting to complete it's cycle. For the initial 8 seconds of the cycle red will be on for one roadway and the other will execute the traffic light cycle in that time frame. Once the cycle is finished, the states reset so the process can run again. The timer is counting (increment) each second to determine how long the circuit stays in the specified state. I chose a 4 – to – 1 multiplexer for the initialization of the circuit. Both the outputs of the D-flip-flops are passed to a decoder to determine the states the circuit is in. The four outputs determine light that will be illuminated at the each intersection. After all the states have been executed, the state will be reset and the process will execute.

**Circuit Diagram:**




**Timing Diagram of Circuit Cycle:**