

traffic controller



spring 2019

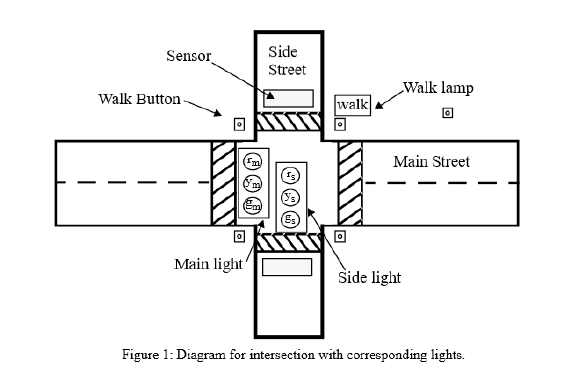
the american university in cairo

Eman Asem – Nada Badawy

900172070 – 900171975

This project is a simulation for a traffic light controller for an intersection between a main road and a side road. The main road has more traffic than the side road. However, at rush times both roads must work together.

The main idea is to design a traffic light controller which controls the cross roads and allowing people to pass at the same time.



# The design of the project:

**Time parameter:**

The time parameter is made to save the three values of the time to be used later in the program. The three times are tBASE , tEXT, and tYEL. Where they are in order as the tBASE = 6 which is equivalent to 0110 , tEXT = 3 which is equivalent to 0011, and tYEL= 2 which is equivalent to 0010.

**Clock divider:**

The timer is working as a clock divider and apply the values of the time on them. First it takes the time of the clock which is 100MHz and then divides it by the n which is 50MHz which will be later divided be 2 giving a final value of 1 Hz which is 1 second. Then apply that clock on the whole program which will control the traffic light. The expired signal is used when the counting is finished so the signal of the expired will be high making the finite state machine go to the next state.

**Debouncer and Synchronizer:**

The FSM is controlled by the asynchronous inputs which change according to the inputs of the user. Meanwhile, the time is changed too which result in a problem of sustaining the state of the clock with the input of the user so the debouncer and the synchronizer are made to resist the fluctuation of the rising edge and sync the clock with the inputs and by that the each state will work properly regardless of the inputs. all input are passed to the synchronizer before going to other functions. The main function of the synchronizer is to make sure that the inputs are synced with the system clock.

**Traffic state:**

The traffic state translates the number of the case into action and that was done by the FSM. As the first state was:

1. The start main green : at this state the main traffic light is green, the side traffic light is red, the time is set to tBASE and the walk request is off.

* The main green when the sensor reads no traffic : at this state the main traffic light is green, the side traffic light is red, the time is set to tBASE and the walk request is off.
* The main green when the sensor reads traffic : at this state the main traffic light is green, the side traffic light is red, the time is set to tEXT and the walk request is off.

1. The main yellow : at this state the main traffic light is yellow, the side traffic light is red, the time is set to tYEL and the walk request is off.
2. The pedestrian walk when the walker pushes: at this state the main traffic light is red, the side traffic light is red, the time is set to tEXT and the walk request is off.
3. The side green when there is no walker pushes: at this state the main traffic light is red, the side traffic light is green, the time is set to tBase and the walk request is off.

* The cont side green when there is traffic: at this state the main traffic light is red, the side traffic light is green, the time is set to tEXT and the walk request is off.
* The side yellow when there is no traffic: at this state the main traffic light is red, the side traffic light is yellow, the time is set to tYEL and the walk request is off.

1. Once again the first state which is Main green.

**Walk register:**

The walk register allows the walker to make a request in order to pass but only for a specific time. By pushing the button the main and side traffic are red and the walker is allowed to pass

**Traffic controller:**

Is the connecter of all the modules in the program.

A close up of a whiteboard

Description automatically generated

To implement this program we first designed a FSM to be able to decide the states which will control everything.

Then we started to implement the program. Both members contributed in this coding and testing the program using FPGA which was challenging as there were problems from the FPGA which hindered us from discovering the real problem of the project. After that we fixed the problem of the clock as the push buttons were not synced with the inputs.

We used 7 LEDs , 1 switch and 2 push buttons to be able to simulate the traffic light.