**Semester Project – Traffic Light Control**

There are many applications for controller circuits. One of them is a highway intersection traffic light. In this project you will design and implement a 4-way traffic light controller that uses pressure activated sensors embedded into the roadway to determine the sequence of light changes used to optimize the flow of traffic.

**System requirements:**

The system has 2 input sensors and 6 output lights.

**Input sensors -** are devices embedded into the roadway and they are capable of detecting if a vehicle is over them. The output of a sensor is logic 0 when no vehicle is above and logic 1 otherwise. There are two sensors (will be simulated by switches), and they are:

**North South road vehicle sensor**

**East West road vehicle sensor**

**Output lights –** These lights are part of the traffic light assembly. Each light assembly contains three lights colored red, yellow and green, which are used to indicate stop, attention and go respectively (they will be simulated by LEDs).

**North South road red light**

**North South road yellow light**

**North South road green light**

**East West road red light**

**East West road yellow light**

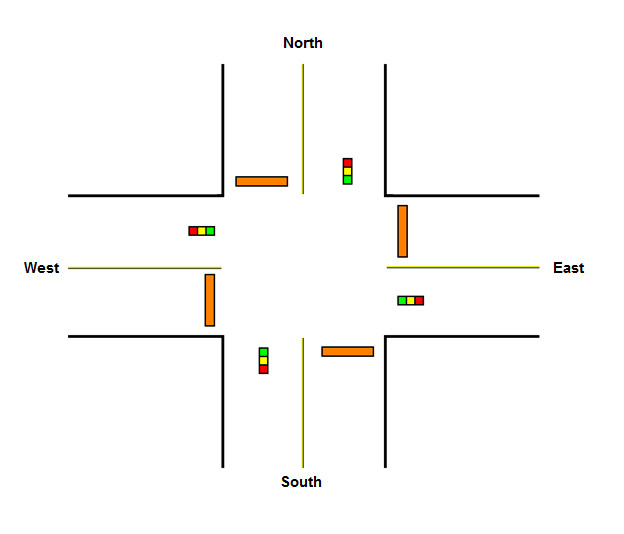
**East West road green light**

#### Details about the design:

* + - 1. The traffic light cycle is shown in the diagram given below:

Diagram 1

Traffic flow diagram



* + - 1. Even though the information provided below may seem obvious, it is always better to have all facts stated explicitly:
* At any given instance both directions, N-S and E-W, may not be red, i.e., assume one direction is red and the other green or yellow.
* When cars are detected flowing in only one direction, that direction should remain green.
* When cars are not detected in either direction, the lights should not change.
* If a car stops at a red light, then the clock count should start. Let’s assume, just for the sake of argument, that after 5 clock cycles it will take a right turn and leave the intersection. As soon as it leaves the intersection, then the clock count should stop at 5. When the next car reaches and stops at the red light, the clock count should start counting continuing from where it left off, aka 5.
* If any yellow light is ON, the clock count should continue even if there are no cars waiting.
* When cars are flowing in both directions the lights should toggle according to diagram 1 shown above.

Create a project called Your\_Last\_Name, then create the Verilog file describing the circuit discussed above. Create a test bench and simulate your circuit. Check the simulation for correctness of your design then synthesize, fit, and program the circuit into FPGA. Test circuitry for correct functionality.

The signals of the circuit should be connected to the following locations on the FPGA board:

Inputs

NS\_sensor – switch (SW3);

EW\_sensor – switch (SW0);

Clock – debounced button (North button BTN\_NORTH). Make sure the button is debounced;

Outputs

NS (R,Y,G) Leds (LED7 – LED5);

EW (R,Y,G) Leds (LED2 – LED0);