ECEN 429: Introduction to Digital Systems Design Laboratory

North Carolina Agricultural and Technical State University

Department of Electrical and Computer Engineering

Freddie Boadu (Reporter)

Mike Umelo (Partner)

April 23, 2020

Pre Lab #11

**Introduction**

This prelab deals with using a traffic light controller in the lab. The traffic light controller can be used to direct crosswalks and traffic daily. This lab will provide us with the technology used to perfect an ideal traffic flow.

**Background, Design Solution and Results**

This prelab provides use insight into how a traffic light would work in vhdl.

Problem 1:

The system would have around 8 states. For the 3 main states there would be a state for if there is a car sensed at that point. And a state for if there isn’t a car sensed at the point. There is a states that controls the north and south and all there types of lights. The same is happening for east and west and all the types of lights as well. The types of lights of green, yellow, and red.

Problem 2:

library ieee;

use ieee.std\_logic\_1164.all;

use ieee.std\_logic\_arith.all;

use ieee.std\_logic\_unsigned.all;

entity TL is

port(r,clk,censorew, censorns: in std\_logic; --This instruction will have the sensor ffor the east west and north south

light: out std\_logic\_vector(2 downto 0)); --Green, yellow and red for all the directions

end;

architecture beh of TL is

type state is (nsgreen,nsyellow,nsred,ewgreen,ewyellow,ewred);-- the state for each light color in each direction

signal cs,ns: state; -- current state, and next state

begin

process(r,clk)

begin

if(r = '1') then --reset

cs<=nsgreen;--to the original state

elsif(clk'event AND clk = '1') then

cs<= ns;

end if;

end process;

process(cs,censorns,censorew)

begin

case cs is

when nsgreen =>

if(censorew = '1' and (censorns = '0' or censorns = '1')) then

ns <= nsyellow;--doesnt change the light

else

ns <= nsgreen;-- stays at the same light

end if;

light<= "000";--the light will be green

when nsyellow =>--auto transition so input doesnt matter

ns<=nsred;

light<= "001";--light will be yellow

when nsred =>--input doesnt matter

ns<= ewgreen;

light<= "010";--light will be yellow

when ewgreen =>

if(censorns = '1' and (censorew = '0' or censorew = '1')) then

ns <= ewyellow;--doesnt change the light

else

ns <= ewgreen;-- stays at the same light

end if;

light<= "011";--light will be red

when ewyellow =>--input doesnt matter

ns<= ewred;

light<= "100";

when ewred =>---input doesnt matter

ns<= nsgreen;

light<= "101";

end case;

end process;

end beh;

**Conclusion**

After doing the prelab, I understand the relationship between the vending machine and its components. The counter and the vendingmachine fsm have to work together to do fully operate the desired outcome.