ECE 4250/7250
VHDL AND PROGRAMMABLE LOGIC DEVICES

LAB#7
Design and Simulation of Traffic Light Controller

I. Objective

The objective is to learn the VHDL modeling of a State Machine. A traffic light controller is selected as an example.

II. Description

Consider the scenario of students walking across 6th street between EBW and EBE. A sequential traffic light controller will help make crossing the street safer. It is suggested that a pushbutton control is installed for this purpose. A pedestrian who wants to cross the street needs to push the button once to get a "WALK" signal. The following describes the control specifications:

- Yellow light for cars will last 4 seconds.
- The green light for cars will last at least 32 seconds.
- The "WALK" sign for the pedestrian will last 20 seconds including 4 seconds of "WALK" flashing.
- There will be "NO WALK" sign when the yellow or green light is on.
- When no one intends to pass the street, the green light for cars should remain 'ON' even after 32 seconds.
- The clock period will be 4 seconds. Another clock with a period of 1 second should be available for flashing the "WALK" sign.

Notes:

✎ You will need to generate the clock for the 4 seconds and 1 second pulses. Lab 6 should help you work on this. i.e you need to include your GenClock.vhd file to your design.

✎ You can refer to the text book (Section 4.4 page 201) for example on Traffic Light Controller operation and design.

III. Instructions

You need to derive an adequate state transition diagram for the design. Since the clock period is 4 second, your diagram should have 14 states. Use the following notation:

- Sb=1: the pushbutton PB has been pressed; a person is waiting to cross the street.
- Ga, Ya, and Ra: drive the green, yellow, and red lights for cars.
- WALK, NOWALK: drive the corresponding lights for pedestrian.

Implement your own VHDL file and define your entity in the following form:

```
ENTITY traffic_light is
   PORT (clk, PB: in std_logic;
            Ga, Ya, Ra, WALK, NOWALK:  out std_logic);
END traffic_light;
```

IV. Deliverables

1. Demonstration is **required**

2. Pre- Lab: A state transition diagram describing your design for the operation above.

3. Report:

a. Objective

b. Lab work:

- Include the code you have implemented and the working state transition diagram. With the details about your implementation.
- Draw block diagram of the whole system which can show the connection of each module. Briefly describe the functionality of each module.

c. Conclusion: What went wrong? Why? , And how did you solve it?

V. Grading

This lab is worth 15 points including report submission.

1. Successful demonstration: 8 Points
2. Report: 7 Points neat and organized report.