LAB #3: Traffic Light Controller

Lab writeup is due to your TA at the beginning of your next scheduled lab. Don’t put this off to the last minute! There is pre-lab work to complete before the start of the next lab. **NO LATE LAB REPORTS WILL BE ACCEPTED.**

1 Objectives

- Use the parallel I/O functions of the 68HC11.
- Learn about usage of FSM abstraction in assembly.
- Design a traffic light controller.

2 Reading

- Read section 2.4 about FSM abstraction
- Read section 3.4.2 about accurate time delays.

3 Background

In this lab, you will use the parallel I/O ports on the 68HC11 to design a traffic light controller. Consider the intersection shown in Figure 1. The streets are labeled North/South and East/West. There are three inputs. Two are car sensors, and one is a pedestrian walk button. There meanings are as follows:

- S1 ON Car is waiting on North/South Road
  - OFF No car is waiting on North/South Road
- S2 ON Car is waiting on East/West Road
  - OFF No car is waiting on East/West Road
- S3 ON Pedestrian would like to cross
  - OFF There is no pedestrian

There are six LED outputs with the following meanings:

- LED1, LED2, LED3 North/South traffic light green, yellow, and red
- LED4, LED5, LED6 East/West traffic light green, yellow, and red

Traffic should not be allowed to crash (i.e., there should not be green or yellow in both directions at the same time). When a pedestrian would like to cross, use the old Boston walk signal by lighting up yellow and red in all directions. Implement appropriate waiting times for each state in your state machine.

4 Tasks

Note: In order to use lab time efficiently, you should complete the first 3 tasks before your lab section.

1. Design the FSM for your traffic light controller.

2. Prepare a schematic for your design including all chips, switches, LEDs, resistors, capacitors, etc.
3. Design the assembly language program for your traffic light controller. Do not use loop delays, but instead use the TCNT timer to implement the time delays in your FSM. Be sure that it is designed using the software issues discussed in class. For example, it should consist of separate modules (ex., subroutine for waiting, another for reading the switches, another for setting the LEDs, etc.). It should also be well documented.

4. Connect your circuit and debug your software.

5 **Writeup**

Include the following items. In this lab, only one writeup per team is required.

1. Your FSM diagram.

2. Your hardware schematic.

3. A printout of your assembly code.