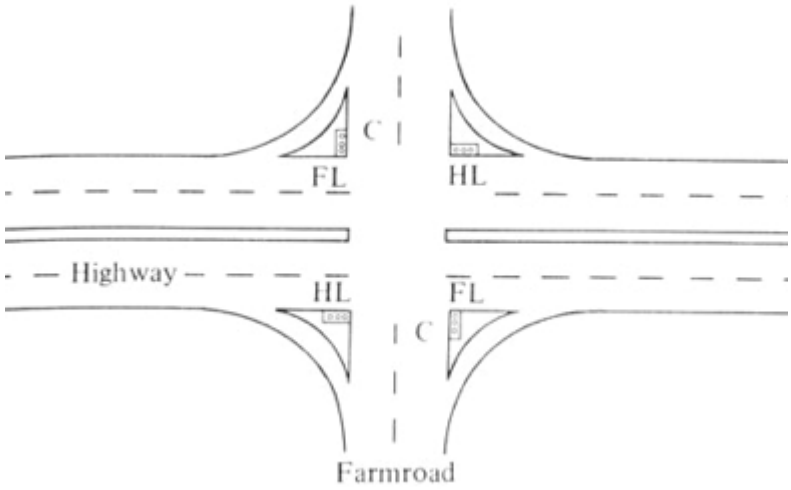
CS 310H **Computer Organization and Programming** **Spring 2008**
Professor Fussell **Due: Mar. 27, 2008**

Assignment #5

Instructions: The assignment is due on the date shown above. Tips to remember: give the assignments to your TA in section, remember your name, section number, TA name, and assignment number (5 points). Also, make sure your assignment is neat, stapled, and is entirely **your own work**.

1. P&P 4.5
2. P&P 4.7
3. P&P 4.8
4. P&P 4.16
5. **Problems of State II:** The most critical aspect of state problems is defining the states, the inputs, the outputs, and the state machine. In this question, you get to stop as soon as you have the state diagram - you don't need to worry about creating the actual logic. However, we need to standardize on a notation for the state diagram to reduce confusion.
   The Problem: In the 1950's, suburban sprawl hadn't filled in all the empty space yet - in fact, developments were small islands surrounded by oceans of farms with quiet superhighways carrying people across the countryside. This allowed a much different technique for controlling traffic lights.



The Setup: You have a super highway crossing a farm road. You wish to install traffic lights. You put two car sensors at the points marked "C" on the Farmroad. Here is logic for the traffic lights: Like a normal traffic light, one direction is red while the other is green. When it's time to switch, the green light turns yellow for a brief time, then red. (Just like the traffic lights you've seen.) The main difference is that we assume the Highway has priority and the Farmroad is almost never used, so...

- Normally, the Highway light is always green...
- ...until the sensors detect a car on the farm road. The the light immediately turns yellow, then the Farmroad Light goes green...
- The Farmroad Light stays green as long as you still see a '1' from the C sensor (meaning there are cars on the Farmroad)... EXCEPT THAT...
- ...there is a certain "long interval time". The Farmroad cannot have a green light longer than this time, even if cars are waiting. And once the Highway Light goes green again, the Farmroad Light can't turn green again until a second long interval is up, even if cars are present.

Got it?

Your hardware will include two timers, a Short Timer for yellow lights and a Long Timer for keeping the Farmroad traffic from getting too uppity. Both Timers have a 'Start' input, and an output which is normally '0'. When you assert 'Start' on a timer, it starts counting down. The entire time it's counting down it outputs a '1' value, and when the time is up, it reverts back to a '0' output. The Highway Light can be set to 'Red', 'Yellow', or 'Green', and also the Farmroad Light can be set to 'Red', 'Yellow', or 'Green'. Your job is to create a state machine (described by a state diagram) that will correctly control the traffic lights according to the rules above.

a. Clearly explain each of your machine's INPUTs and OUTPUTs. State what symbol or abbreviation you will use for each of these Inputs and Outputs. Also state what the states in your machine will actually represent and how they will be labeled.

b. Draw the state diagram that describes the operation of your state machine. There are Mealy and Moore state diagrams, and there are different ways to draw both of these. You have learned different forms in different classes, but they are all logically equivalent. In order to make things as clear as possible, this time we ask that you draw your state machine using a Mealy diagram with the following syntax: In each circle, draw only the state identifier. For each input or condition a state could react to, draw an arrow from that state to the next state. Along this arrow, summarize the input conditions (separated by commas), then a slash /, then the outputs (separated by commas). Make sure that ANY symbols you write in this diagram are described in part a.