Lab 9/10: Decomposing State machines: Traffic Light Controller

PURPOSE

Just like large procedures or subroutines in a programming language, large state machines are difficult to conceptualize, design, and debug. Therefore, when faced with a large state-machine problem, digital designers often look for opportunities to solve it with a collection of smaller state machines. The purpose of this lab is to design and build a clocked synchronous state machine of moderate complexity. This design can be approached using basic design techniques, but such an approach would be a significant effort. In this lab, students will simplify the design process by decomposing the overall design into two sub-designs: a data unit and a control unit. This is a two-week lab project. The goal of the first week is to design the data unit for the project. The goal of the second week is to design the control unit for the project.

Traffic Light Controller

Consider the intersection of a busy highway and a little used farmroad, shown below. There is a sensor on both sides of the farmroad that detects the presence of a car waiting to cross the highway. Our goal is to design a traffic light control for this intersection to grant right of way to the roads as defined below.

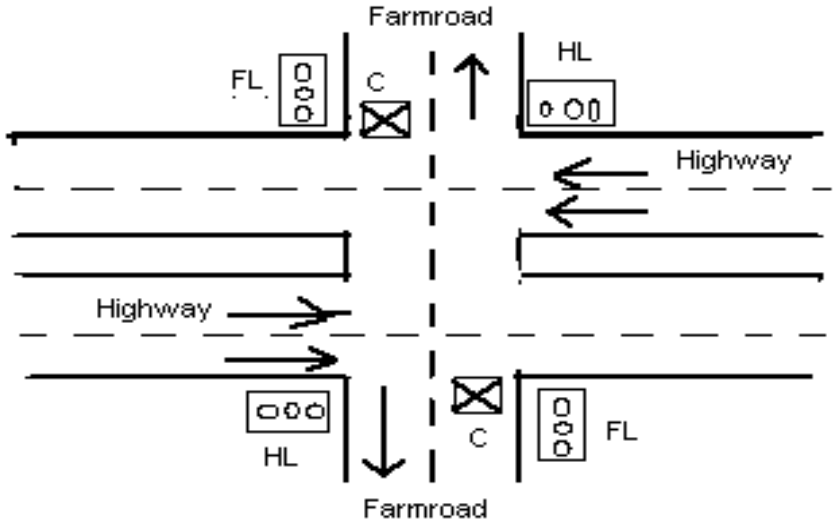

Figure 1: (a) Highway/Farmroad intersection – Placement of lights and sensors.

The traffic light control is a state machine with two *synchronous* input signals (CARDETECT and RESET) and six *synchronous* output signals (HLG, HLY, HLR, FLG, FLY, FLR) that control the traffic signal lights. The state machine to be designed must realize the following operational characteristics described below:

- The assertion of the CARDETECT input signal (denoted C in Figure 1) indicates the detection of a vehicle on the farm road. As long as CARDETECT is not asserted, the highway has right of way. When CARDETECT is asserted, the farmroad is granted right of way (subject to limitations below).
- The farmroad may not retain right of way for longer than $t_{rightofway}$ (even if CARDETECT is asserted).
- Once relinquished, the farmroad may not be granted right of way again for at least $t_{rightofway}$.
- Before a road can gain right of way, the road which currently has right of way (indicated with a Green traffic signal) must relinquish right of way (indicated by changing to a Yellow traffic signal for some period of time t_{yellow} and then changing to a Red signal). One road should always have right of way.
- $t_{rightofway}$ shall be defined to be 16 clock ticks, t_{yellow} shall be defined to be 3 ticks.
- Right of way should **never** be granted to both roads at once – even if your circuit somehow gets into an unexpected state. We must guarantee that the controller is risk-free – virtual lives are on the line!
- If you have any questions about the proper operation of this device, ask your TA. If you must make any assumptions, note them in your lab notebook.

LAB 9 – DATAPATH DESIGN

[6 points] Data units are used for storing, combining and processing data. Usually, a data unit consists of multi-bit counters, shift-registers, and ALUs which increment, shift, or modify register contents according to the control signals applied to the unit. The inputs to the data unit consist of primary inputs (usually) and commands from the control unit (**control signals**). The outputs of the data unit consist of the primary outputs (usually) and **status signals** that provide feedback to the control unit from the data unit.

You will need to consider how to best decompose this design - it may take several attempts to get something that works well. The most **difficult** part of this assignment may very well be in determining exactly HOW to decompose the design. Which unit will do what work? What controls signals will produce the work. In what order to the control signals have to be sent? What status signals have to be sent to the control unit in order for it to make appropriate decisions? Give this a good deal of thought and be willing to revise your design if you find opportunities for improvement later on. Refactoring is a solid design tool – both at the software and hardware level!

You may use whichever control and status signals you wish (subject to the restrictions below). The number and type of control and status signals is up to you. Well thought-out selections may reduce the required design effort. Poorly thought-out designs may be amazingly difficult to implement. Consider carefully the actions that the control unit will need to request.

In designing the data unit for this project, you are subject to the following restrictions:

- The data unit must contain (at a minimum) two 3-bit registers (one register for each light). One register must be called HighwayLight (or HL). One register must be called FarmroadLight (or FL). The datapath may contain any additional state devices but MUST include at least these two registers.
- The three outputs of HL and FL registers must be used as the primary outputs of the traffic light signals controller. The outputs for the farmroad lights are FLG, FLY, and FLR. The outputs for the highway lights are HLG, HLY, and HLR. A value of 1 on these signals turns on the appropriate traffic signal lamp. HLG asserted, for example turns on the HighwayLightGreen lamp. Exactly one lamp of each signal should be asserted at any given time.

Design your datapath within these constraints. Clearly communicate your datapath design (and/or design revisions). Construct a table, diagram or other representation that clearly and unambiguously shows how the inputs to the data path (control signals and any relevant primary inputs) affect its state and outputs (status signals and any relevant primary outputs).

If you get stuck: consider beginning with a state diagram showing all possible state transitions. Examine your state table to identify all possible transitions required by this design. With this overall functionality in mind, construct a device of making those state transitions. Create a set of control signals that specify which transitions will take place. This should help you design a datapath that uses these control signals as inputs and implements the appropriate operations. Note that our decision to use two registers has an advantage in that the farmroad lights and the highway lights are two separate (small) sub-machines. Keep in mind that the data-path does not need to contain *any* next-state functionality (although it can – if you design requires this). The data path will certainly need to have controls signals that allow it to be able to transition between the necessary states.

Describe how your data unit will respond to various control signals. Include a complete set of circuit printouts and timing diagrams in your lab book. Construct your design using the laboratory simulation package and test it thoroughly. Implement your data-path as a user-defined circuit (macro) with appropriate inputs and outputs.

LAB 9 - DEMONSTRATION

[2 points] Demonstrate your overall design and the preliminary design and implementation of your data unit to your lab instructor. Be prepared to answer questions regarding the decisions you made as part of the design process. Also, be prepared to ask questions about any problems or implementation issues that you are encountering or anticipating.

Integrated Writing [2 points]: Refer to "Digital System Design: Engineering Journals & Lab Policies" for details. 0.5 points each for Completeness, Clarity, Organization, and Testing.

LAB 10 – CONTROL UNIT DESIGN

[6 points] Control units are sub-machines within a design that are used to control the operations of the data unit. The inputs to the control unit consist of primary inputs (usually) and feedback from the data unit (**status signals**). The outputs of the control unit consist (primarily) of **control signals** that select the appropriate data unit operation.

In order to complete our design, we must design a control unit to sequence our data unit appropriately (as determined by the current state and the values of the primary inputs). Design the next-state logic for the Traffic Light Controller using an implementation of your choice. You are free to implement the control unit as a state machine that sequences the operations or as combinational logic that reads/maintains all necessary state information in the state-coded outputs. Depending on your design, you may need to use D-type flip-flops, registers, counters, or possibly combinational logic alone in your control unit.

Document your design thoroughly! Include all attempts (even failed attempts!), tests, the results of tests, and how your design will change because of the results of those tests! As you develop your control unit, you may find that your preliminary data unit does not provide some necessary functionality. If so, you may need to add functionality or redesign your preliminary data unit with the overall functionality in mind.

SAFETY CONSIDERATIONS: Be certain to keep the safety of the users of this system in mind. Only one lamp should be asserted in each direction. It should be impossible for lights to allow traffic through in conflicting directions simultaneously. If something bad happens (like a brown out, lightning strike, or evil TA) that puts your device into an unexpected or unused state then your device MUST behave sanely. Discuss your safety goals and any circuitry, signals, or other modifications that you make to your design to guarantee that the traffic signal controller avoids situations that might lead to accidents!

Include a complete set of design notes, circuit printouts, and simulation results in your lab book. Your simulation results should show that your circuit meets all of the project specifications. Please write in additional comments describing your simulations, and be sure that output/tables/timing diagrams demonstrate the correction functionality of your device.

LAB 10 - DEMONSTRATION

[2 points] Demonstrate the design of your control unit to your lab instructor. Demonstrate the complete functionality of your Traffic Light Controller state machine. Be prepared to answer questions regarding the decisions you made as part of the design process. Also be prepared for your lab instructor to modify your state devices to simulate 'lightning strikes' and thus observe how your device deals with the unexpected and resumes safe operation.

Integrated Writing [2 points]: Refer to "Digital System Design: Engineering Journals & Lab Policies" for details. 0.5 points each for Completeness, Clarity, Organization, and Testing.