UNIVERSITY of GUELPH

Department of Computing and Information Science

Digital Systems
(CIS*3120)

Laboratory Exercise 10

Description

In this lab you will design a traffic-light controller to control an intersection. The intersection consists of two streets, one running north-south (called NS) and the other running east-west (called EW).

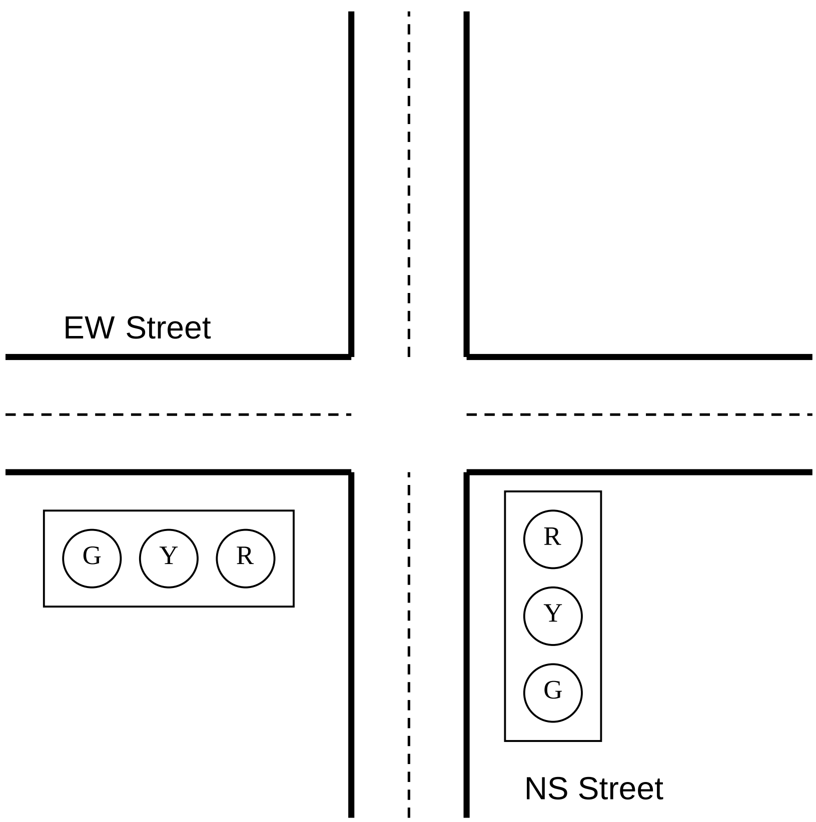

Figure 1: *Intersection controlled by traffic-light controller*

The circuit has to control the traffic light on the NS street and the traffic light on the EW street as shown in Fig 1. Each traffic signal consists of a red, yellow, and green light. Each signal will cycle through red, green yellow, and back to red. When one signal is green, the other will be red; when yellow, the other will be red; and when red, the other will be green. To simplify the design, set the green, yellow, and red time periods to 4, 1, and 5 seconds (clock cycles), respectively.

Part A

Design your traffic-light controller as a clocked synchronous state machine with 10 states, and use one state for each of the 10 seconds required in the traffic-light cycle. The number of flip-flops needed to code the 10 states is the smallest integer greater than or equal to $\log_2 10$, which is 4. Use D-type flip-flops when implementing your circuit.

You will not require any inputs, apart from the clock and a reset line, to initialize the circuit. You will, however, have to provide three outputs to control the traffic light on the NS street and three outputs to control the EW light.

Begin by drawing an appropriate state diagram for the problem.

STATE DIAGRAM

Next, create a state-transition table that lists each of the possible current states along with the outputs and next states. You should then construct an excitation table that shows the input values needed for each of the four D-type flip-flops to move to the desired next state. From this table you can derive the excitation equations and simplify them using Karnaugh maps.

STATE-TRANSITION TABLE

Finally, from the state-transition table you can derive the output equations to drive the six LEDs (use appropriate colors, i.e., red, yellow, and green). These equations can also be simplified using Karnaugh maps. Implement and test your circuit, then demonstrate your circuit to the TA. (Note: The TA will also ask to see your state diagram, state-transition table, minimized expressions, etc.)

K-MAPS and MINIMIZED EXPRESSIONS

Part B

You will now extend the traffic-light controller in Part A to include a left turn traffic sensor and signal for the NS street as shown in Figure 2. Assume that the sensor (implemented as a binary switch) will produce a logic 1 signal anytime there is a car or cars waiting to turn left on the NS street. If no cars are detected, the traffic controller will simply cycle the NS and EW lights as in part A. At the end of the EW street's green and yellow period, when the NS street's green phase would normally start, the left-turn detector is checked to see if a car is waiting to turn left. If a car is waiting, the left-turn light should be turned on for two seconds, indicating the car can now turn left. During these two seconds the EW and NS lights will be at red. Once the two seconds have elapsed then the left turn light should be switched off and the NS green light set to on. The cycle then continues as before.

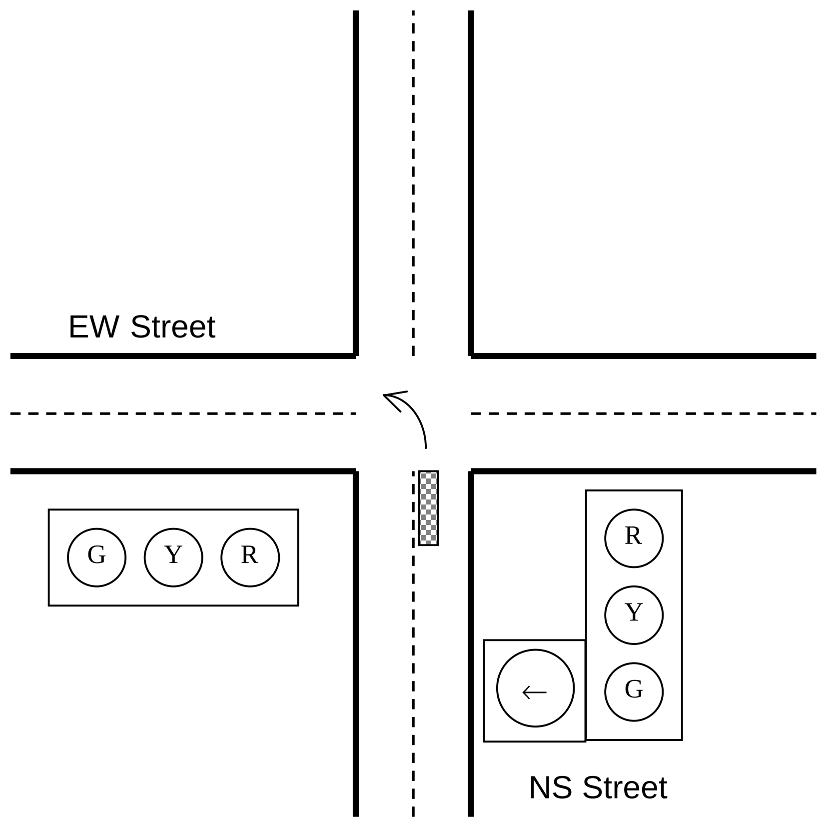

Figure 2: *Intersection with a left turn traffic sensor and light.*

The design process for this controller is very similar to the previous one. The state machine will now have 12 states instead of 10 to support the two-second left-turn condition. The previous design did not have any conditional inputs and so the excitation equations and resulting logic were simply a function of the current state. We now have an input representing the left-turn car detector and if this input is active (indicating that a car is waiting) during state 10, the state machine will go into states 11, then 12, then back to 1. If, on the other hand, the input is inactive (no cars waiting on the left turn), the state machine must to directly from state 10 to state 1.

Follow the **same procedure** described in Part B to construct this new controller.

STATE DIAGRAM

STATE-TRANSITION TABLE

K-MAPS and MINIMIZED EXPRESSIONS

Implement and test your circuit, then demonstrate your circuit to the TA. (Note: The TA will also ask to see your state diagram, state-transition table, minimized expressions, etc.)