CSC 258 Lab 2

You are to design and build a simple controller for the traffic light at a pedestrian crosswalk.

The operation of the crosswalk is as follows. The traffic light is normally green. When the pushbutton (PB) is pressed, a pedestrian-waiting (PW) signal is set and latched. The PW signal will cause the traffic light to turn yellow for one clock cycle, red for two clock cycles, and then green again. The PW signal is cleared when the traffic light turns red. After the traffic light turns green again, it stays green for a minimum of three cycles before another PW signal can restart this sequence.

This circuitry will be driven by a *state machine*. There are six states, numbered zero through five. The traffic light is usually in state zero, in which it is green, but in which pedestrians can call for a crossing any time. When a pedestrian calls for a crossing, we start cycling through states 1 through 5, then back to 0. In state 1, the traffic light is yellow; in state 2 and 3, the traffic light is red. In states 4 and 5, the traffic light is also green, but distinguishing these states from state zero is how we enforce the rule that the red light is followed by a minimum of three cycles of green.

If the pedestrian presses the button after state 2 (which is when the traffic light turns red and pedestrians are allowed to begin to cross), the button-press does not affect the traffic light immediately; but it is remembered until the next time we come around to state zero. This is why we need the separate PB and PW values. PB is an input switch; PW is a non-clocked latch.

Part A

First, build the counter which cycles through the states. It will count 0, 1, 2, 3, 4, 5, 0, 1, 2, 3, 4, 5, ... To add 1 repeatedly, use a 4-bit full adder chip (74LS283) and a 4-bit edge-triggered register (74LS175), in a similar configuration to that in part 3 of lab 1; however, the number being added will not be an input, but rather, the constant 1 (power the low bit and ground the other inputs as well as the carry-in).

Rather than using the register's ‘‘reset’’ input, each of the data bits going into the register should be ANDed with the output of a combinational circuit you devise which yields true iff the output is not equal to five. This will cause the counter to go back to zero after state 5 (but not immediately; in the subsequent cycle). Using this design instead of a traditional counter design will facilitate later parts of the lab, in which another line will also be ANDed in there.

Use an input switch for the clock, and indicate the value from 0 to 5 using the left three output LEDs.

Part B

Write a chart in which the meaning of each state is clearly delineated, including the colour of the traffic light. State how the PW signal affects the transition from state 0 to 1.

You will use an SR-latch to implement the PW signal. PW goes on when PB is on (pressed); PW goes off at state 2. Note that wiring PB into PW's ‘S’ input and a state-two-detector into PW's ‘R’ input means that we might have a situation where S=R=1 — and this is ok! Be sure to understand why this is ok.

If no pedestrians are around, the traffic light should stay green. Describe and present a formula to be ANDed in with your non-five-detector in part A to cause the state to stay at zero when no pedestrians are around.

(over)

Part C

Design, build, and test the circuit to implement the entire pedestrian crosswalk controller. Your circuit should have two inputs (clock and PB), each of which will be an input switch. Your circuit's outputs should still include the three-bit state value from part 1, and further LEDs towards the right should represent the green, yellow, and red traffic lights in that order from left to right.