LAB 4: FINITE STATE MACHINE

Part 1: (Prepare state diagram)

We wish to implement a finite state machine (FSM) that recognizes two specific sequences of applied input symbols, namely four consecutive 1s or four consecutive 0s. There is an input w and an output z. Whenever w = 1 or w = 0 for four consecutive clock pulses, the value of z has to be 1; otherwise, z = 0. ***Overlapping sequences are allowed,*** so that if w = 1 for five consecutive clock pulses the output z will be equal to 1 after the fourth and fifth pulses.

Use the toggle switch SW0 on the Altera DE2 board as the input w, the LEDG0 as the output z, and the push button KEY0 as the clock input which is applied manually. Simulate the behavior of your circuit and then test the functionality of your design on board DE2.

Part 2:

It is often useful to be able to sequence through an arbitrary number of states, staying in each state an arbitrary amount of time. For example, consider the set of traffic lights shown in Figure 2.1. The lights are assumed to be at a four-way intersection with one street going north-south and the other road going east-west.

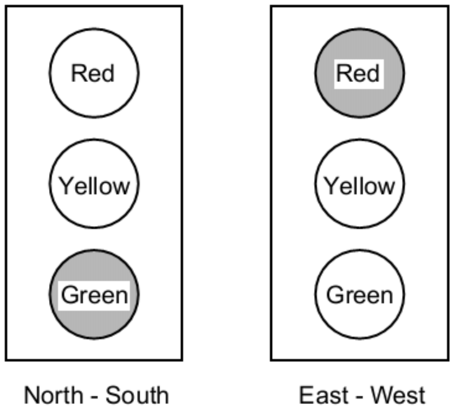

Figure 2.1 - Six colored LEDs can represent a set of traffic lights

A set of states for controlling these traffic lights is shown in Table 2.1 and a state diagram is illustrated in Figure 2.2. Use LEDs on board DE2 for displaying the lights Red, Green, Yellow; CLOCK_50 as input for controlling the timing of your circuit. Write the program and test the functionality of your circuit on board DE2.

Table 2.1 – Trafic light states

State	North - South	East - West	Delay (sec.)
0	Green	Red	5
1	Yellow	Red	1
2	Red	Red	1
3	Red	Green	5
4	Red	Yellow	1
5	Red	Red	1

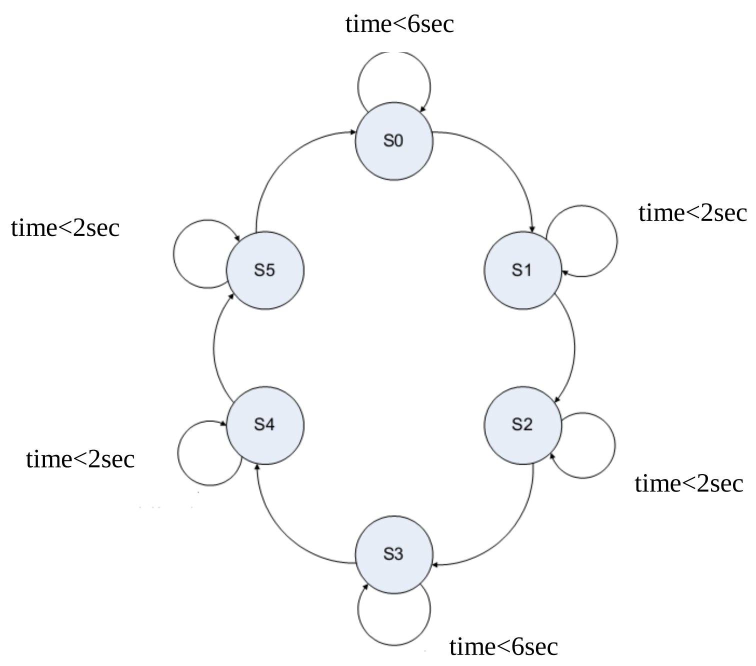

Figure 2.2 – State diagram for controlling traffic lights

Part 3: (Prepare state diagram)

In this part of the exercise you are to implement a Morse-code encoder using an FSM. The Morse code uses patterns of short and long pulses to represent a message. Each letter is represented as a sequence of dots (a short pulse), and dashes (a long pulse). For example, the first eight letters of the alphabet have the following representation:

A	• —
B	— • • •
C	— • — •
D	— • •
E	•
F	• • — •
G	— — •
H	• • • •

Design and implement a Morse-code encoder circuit using an FSM. Your circuit should take as input one of the first eight letters of the alphabet and **display the Morse code for it on a red LED**. Use switches SW_{2-0} and pushbuttons KEY_{1-0} as inputs. When a user presses KEY1, the circuit should display the Morse code for a letter specified by SW_{2-0} (000 for A, 001 for B, etc.), using 0.5-second pulses to represent dots, and 1.5-second pulses to represent dashes. Pushbutton KEY0 should function as an asynchronous reset.

- **Students prepare Verilog code for Part 1 to 3 and Run on Kit De2.**

Reference

Digital_Logic lab from **Altera**.