Experiment 2: Introduction to FSM: Traffic Light Controller

Objective

- To become familiar with up/down synchronous counters
- To implement Finite State Machines (FSM) using Verilog

Background:

Finite State Machines are a useful abstraction for sequential circuits with centralized states of operation. At each clock edge, combinational logic computes the outputs and next states as a function of inputs and present state.

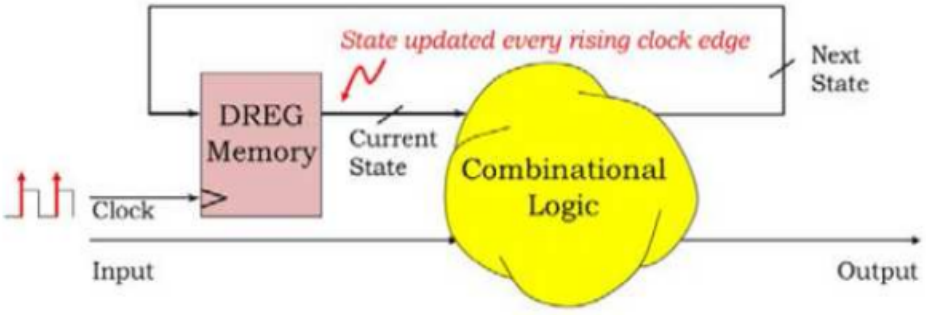

Figure 1: FSM model

Experimental Procedure:

1. Design a 4 bit up/down counter with enable and synchronous reset in Verilog using multiplexers, D register and an incrementer/decrementer. User can choose the mode of operation (up/down count) through a *mode* signal. If *mode* is 1, the counter counts up, and if *mode* is 0, counter counts down. Finally develop a simple test bench for verifying the same.

2. Implement a traffic light controller in Verilog. There is a highway and a crossroad, with a time counter, red and green traffic lights on each direction. When green light is on in one direction (say H:Highway), red is on in the other direction (say C:Crossroad), and the counter starts counting down from X to 0. Once the counter hits 0, the red traffic light should be on in H and green in C with the counter in C starting to count down from Y to 0. This should keep alternating. Come up with a nice state diagram and Verilog code using FSM abstraction. X and Y can be any number between 1 to 7. Use high counter value for highway counter and low counter value for crossroad counter to create more realistic scenario.

3. Modify the specification in qn. 2 such that the traffic light has an additional yellow light with a small timer value. Come up with a nice state diagram and Verilog code using FSM abstraction.