Billy Koech and Nicolas Weinger

CS 141: Computing Hardware

Lab 3 part1 : Traffic Light Controller

Design, Testing and Simulation methodology

**Timer**

The timer is implemented with an asynchronous reset. A current state variable set to 4’b1111 through a nonblocking assignment when the reset is asserted; otherwise it is set to the next state variable. The next state variable is calculated through a blocking assignment. This is implemented through a sequence of if else statement that prioritizes the load (if load is asserted), then setting the next state to zero (if the current state is zero), then decreasing the counter (if enable is asserted) and finally setting the next state to the current state(in case none of the able conditions are met).

**Timer Testbench**

The testbench tests for the following functionality:

1. Load: Load is asserted then the init variable is set to a 4 bit value. The testbench waits for one clock cycle to ensure that the init value is sampled then the output is checked to see if it is equal to init.
2. Reset: 4’b0100 is loaded then reset is asserted. The output is then checked to see if it is equal to 4’b1111
3. Test down counter. Reset is asserted to ensure that the timer starts at 15. Enable is then asserted and the testbench waits for 10 clock cycles to make the timer count down by 10. It then checks that the output is equals 5.
4. Test if timer stops at zero: Reset is asserted to ensure that the timer starts at 15. Enable is asserted and the testbench runs for 20 clock cycles. The output is then checked to see whether or not it is equal to zero.
5. Test if timer holds value when enable is not asserted: Reset is first asserted then the clock runs for 5 clock cycles to make the timer count down by 5. Enable is then disabled and the clock runs for 5 more clock cycles. The output is then checked to see if it’s still equal to 15-5 and not 15-10.
6. Test if reset takes precedence over enable: Reset is asserted while enable is asserted. The output is then checked to see if it is equal to 15.
7. Test if reset takes precedence over load: Reset is asserted while load is asserted. The output is then checked to see if it is equal to 15.
8. Test if load takes precedence over enable: Load is asserted while enable is asserted. The output is then checked to see if it is equal to the init variable.

**Clock Divider Testbench**

The clock divider testbench simply pulses the clock input of a parameterized clock divider. We then observed the waveforms to check that the divider was indeed behaving as expected.