Traffic Light Controller

We have eight states in this FSM. Note that we begin at s0/IDLE, and end at S10. We managed to reduce redundancies in code by combining some setting states with enable states for when lights switch and timers turn on. We implemented a Mealy machine, primarily exemplified through the use of last_green and next_ped, where the present state was determined as a result of both the prior state and its input values, those values being last_green and next_ped.

Without going into the logic of the machine (it is exactly as prescribed on the spec), here are some key design choices we made aside from implementing a Mealy machine. First, we made sure to preserve the value of outputs throughout states. This ensures that we don’t forget a last_green, which ensures that we keep the alternating cadence of our system in check. This was our alternative since we didn’t truly have a “memory” device— our memory implantation was holding those values to the prior clock cycle value (unless it needed to be changed) for every clock cycle. Another design choice was the amount of time the timer was to be held. We saw that there was an extra clock cycle between a timer’s enable and the timer’s finishing, and this is because it takes one clock cycle for enable to go high and load to go low after having loaded in the proper values to the timer. To solve this, since we know we had 1 elapsed clock cycle, we made our timers elapse for one second less, e.g. the pedestrian state sends 4’d14 instead of 4’d15 to the timer, since the extra second shouldn’t be double counted.

The testbench simulates the main inputs that can be expected for the FSM. A 1-Hz clock feeds into instantiated timer and traffic_light modules, while an always block checks that the state the FSM moves to is the correct one (ensuring that the cur state matches the prev state). After a quick reset, different combinations of pedestrian and car signals are tested (and held) to ensure that the normal functioning is correct. There's a problem with verilog not comparing things correctly in the warnings at the bottom.