# Introduction:

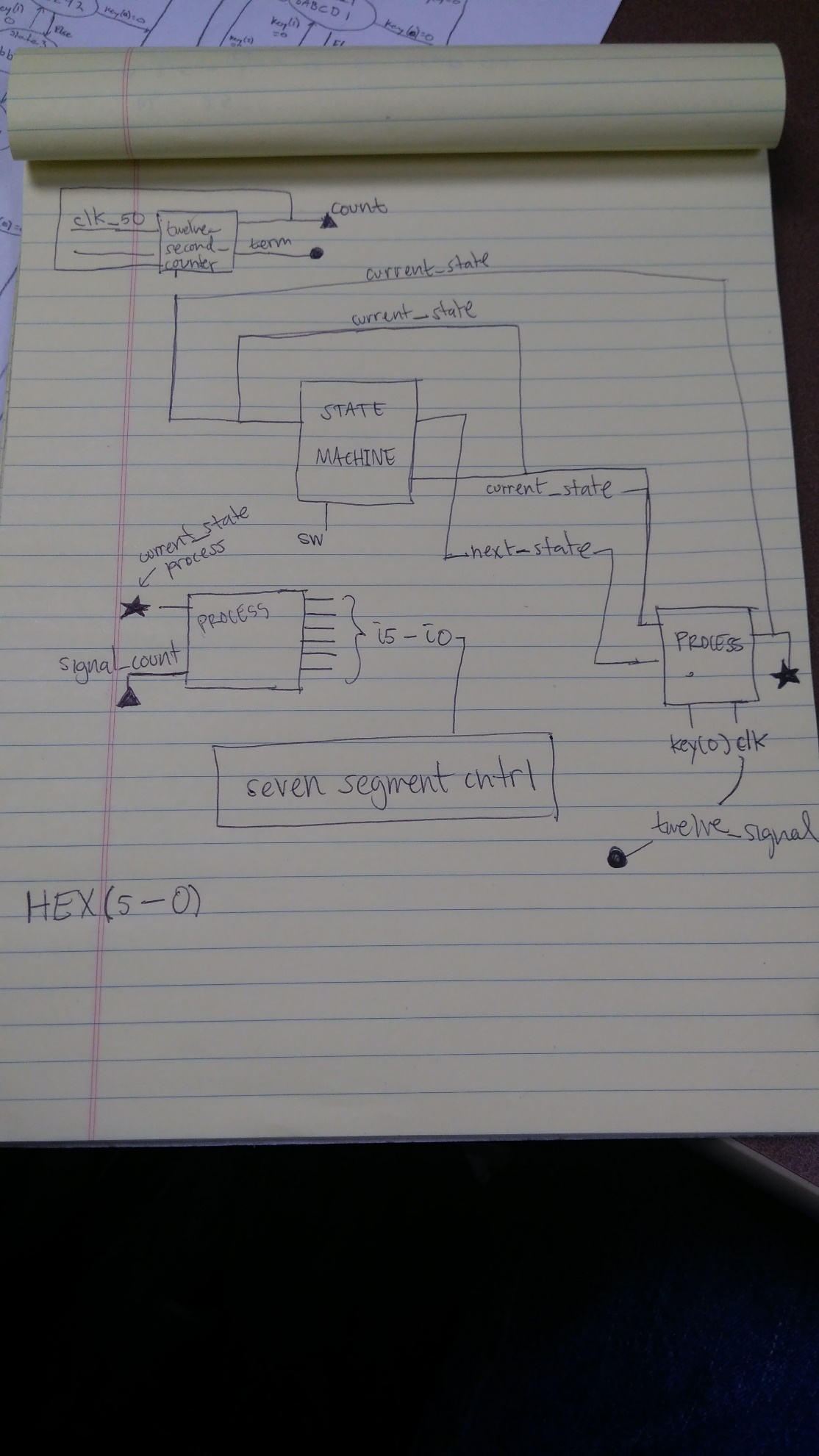
For this lab we will be programming the FPGA to simulate a two-road intersection.

# Theory of operation:

1. For this design there are multiple requirements.
   1. Each direction will have a GYR (green, yellow, red) sequence and a left turn signal
   2. State transitions must be safe
   3. There must be a night mode, with flashing yellow and red lights.
   4. KEY(0) must reset the design
   5. All switch and key inputs should use a 2 register pipeline clocked at 50mHz, except KEY(0).

# Procedure:

1. These are the details of my implementation
   1. The Hex displays, from HEX(5) to HEX(0) are as follows:
      1. HEX(5): Northbound Light
      2. HEX(4): Southbound Light
      3. HEX(3): Northbound/Southbound Left turn signal
      4. HEX(2): Eastbound Light
      5. HEX(1): Westbound Light
      6. HEX(0): Eastbound/Westbound left turn signal
   2. The left turn signals will be for both the NS and EW traffics, driving straight during these signals will not be allowed
   3. Drive SW(6) to high to indicate that there is a car Northbound or Southbound waiting to turn left. Similarly, SW(5) indicated for Eastbound/Westbound.
   4. Drive SW(0) high to infer the night-mode. During this time the intersection is composed of two flashing red lights normal to flashing yellow lights.
   5. There are six states used to implement the traffic behavior. Each state (except break state) represents 10 seconds of movement, followed by an additional 2 seconds of standstill traffic to facilitate state transition.
   6. The break\_state exists for system resets and transitions into and out of night mode.
2. Block Diagram



* 1. Some of the limitations of the design
     1. The current design does not allow traffic to go straight when there are left turn signals, Ideally, if there is an open oncoming turn lane in traffic during a turn signal the ongoing lane is safe to pass.
     2. The break state does not need to last as long.
  2. Ways the design can be improved:
     1. The seven-segment display component instantiation could be delegated to a function instead, with the result being passed directly to the HEX displays.

# Verification:

1. Test Plan:

We will verify the design by creating a test bench file with predetermined signals. We will create use that file to create a simulation waveform to use in ModelSim. For this simulation the clock-based signals are always enabled and interdependent. We will verify the state transitions and the HEX displays by altering the key, and switch signals.   
  
  
Scenarios objectives included:

* Making sure the reset works. (key(0))
* Making sure the state transitions worked correctly
* Making sure the hex displays displayed the right characters.

# Conclusion:

The most important things taken away from this lab was learning how to design logic around real-life application scenarios. The implementation had to take into consideration the demand of the user/consumer. The technique of creating states to organize signal responses is also demonstrated in this lab. Most of the difficulties stemmed from the lack of flexibility of components of the FPGA that could be used to represent the traffic light.). If prompted to start over, I would implement additional logic that would consider left turn lanes in all four lanes.