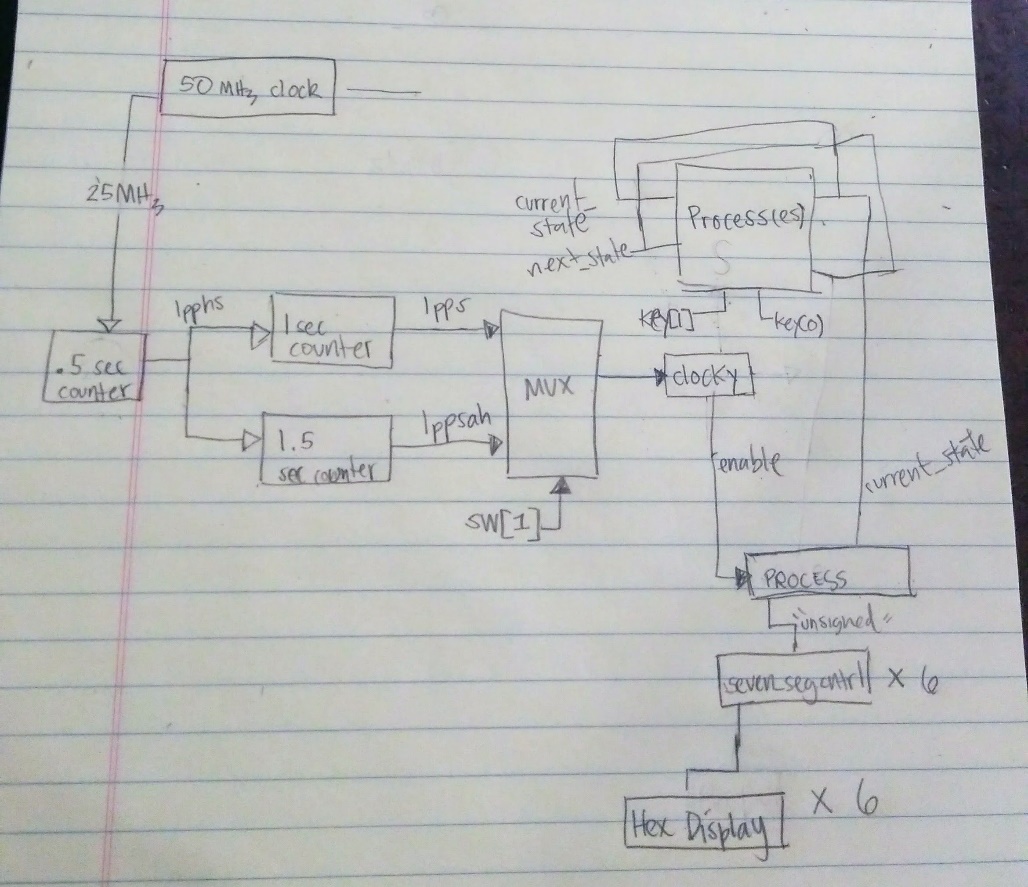
# Introduction:

1. For this lab we will be programming a finite state-machine with a sliding display using the hex displays on the do1-soC device.

# Theory of operation:

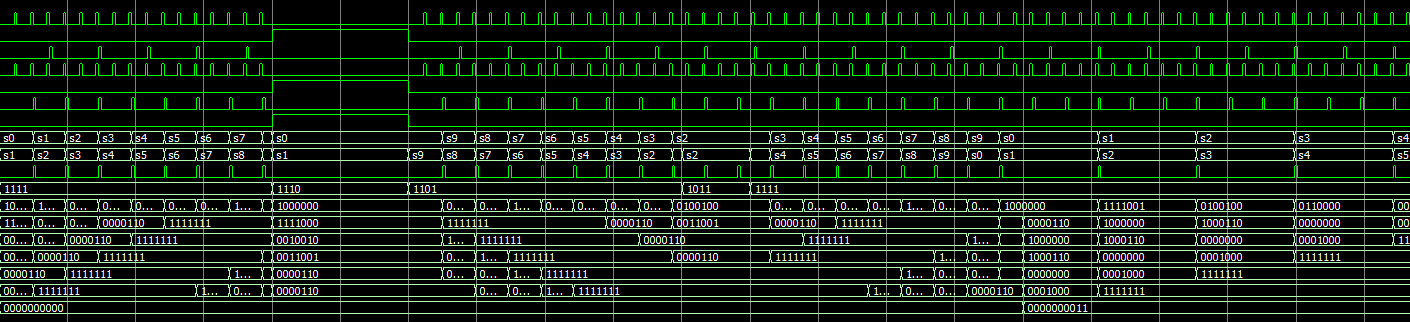
1. For this design there are multiple requirements.
   1. Minimum amount of 2 processes.
   2. Similar to lab 5 and state machine input signals are delayed by a shifter controlled by the KEY(0) and KEY(1) signals
   3. Block Diagram:



* 1. Some of the limitations of the design
     1. Ways the design can be improved:
        1. The following design uses 3 different clocks. The same functionality can be achieved with 2 clocks or 1 clock with a defined count signal.
        2. 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 are the flexibility and knowledge for implementing the hardware-time relationship. The technique of creating states to organize signal responses is also demonstrated in this lab. Most of the difficulties stemmed from my misinterpretation of the Hardware key symbols (being inverted). If prompted to start over, I would implement the design with 2 clocks to reduce the amount of registers and tidy up the implementation