ECPE 174 – Lab 3
Asynchronous Design

I. Objectives

Design and implement an asynchronous finite state machine that controls a combination lock.

II. Problem Statement

A combination lock has three input keys: A, B, and C. These input keys cannot be pressed simultaneously, but only one at a time. The lock has the following features:

- Opens lock by generating a *unlock* signal when a person enters the combination "BACB."
- Closes the lock by clearing *unlock* signal when any key is pushed and lock is open.
- Resets to initial state (locked with no keys entered) when a person enters any incorrect combination such as "AC," "BC," and "BACC." System does not remember past entries so "BB" would also send it back to an initial state.

Create the controller for this lock using an asynchronous Moore FSM. Connect the input keys to A=KEY2, B=KEY1, and C=KEY0. Output *unlock* on LEDG0.

III. Pre-Lab

Complete the following steps before lab and turn in by 2pm:

1. Generate a state machine diagram and state assignment table.
2. Implement the design using behavioral VHDL.
3. Simulate your code using the waveform editor and functional simulation. Ensure simulation covers 75% or greater of the test cases (hint - check out the report generated when performing the simulation) and use simulation to verify VHDL matches your state machine.

Submit electronically via Sakai if possible. Otherwise submit a clean, handwritten copy by 2pm in class.

IV. In-Lab

Complete the following steps in lab:

1. Implement the VHDL design on the Cyclone II.
2. Demonstrate it for check-off of the lab.

V. Post-Lab

Discuss the following your lab report:

- What does the Quartus state machine viewer generate? How does it compare to your state machine? Explain any discrepancies.
- What complexities arise in designing an asynchronous FSM compared to a synchronous FSM?