Final Project

BICYCLE LOCK

OBJECTIVE:

1. To design a finite state machine for a lock/alarm system.
2. To use VHDL to define a finite state machine.
3. To implement the finite state machine of the alarm on a FPGA,
4. To experimentally check the operation of the lock/alarm system

Problem:

Design an alarm system for a bicycle that has five input push-buttons: X4,X3, X2, X1 and X0. Assume that the buttons cannot be pressed simultaneously (an electromechanical interlock guarantees this). The alarm system should have the following features:

1. X4 represents the ENTER button.

2. The main three states are:
   A. Reset State
   B. Lock/Armed State
   C. Alarm state

3. A. The combination for locking (arming from the Reset state) is

   X2-X0-X3-X3-X1-X4

sequence:	X4	X3	X2	X1	X0	Note: One button at a time is pressed.
	0	0	1	0	0	press
	0	0	0	0	0	release
	0	0	0	0	1	press
	0	0	0	0	0	release
	0	1	0	0	0	
	0	0	0	0	0	
	0	1	0	0	0	
	0	0	0	0	0	
	0	0	0	1	0	
	0	0	0	0	0	
	1	0	0	0	0	ENTER key
	0	0	0	0	0	

B. Pressing X4 (Enter) early in the arming sequence should start the sequence over by returning to the RESET state.

4. The combination for unlocking (disarming from the Lock/Armed State) is

   X2-X0-X3-X3-X1-X4 (same sequence as Arming)

   In order to prevent tampering with the system, an ALARM signal will get activated if any wrong button was pressed. However, in order to make it harder to figure out the right sequence:

   A. We don't want the alarm signal to get activated after the first wrong button has been pressed. Instead, the alarm signal should get activated after pressing 6 buttons, as long as one of the 6 buttons pressed is a wrong one.

   B. While disarming, pressing the sequence X2-X2 during 2nd through 6th pushes should get the system back to the ARMED state.

   C. While disarming, pressing X4 during the first 5 pushes should get the system back to ARMED state.

5. The combination for getting out of the ALARM state is by pressing the
   X2-X0-X3-X3-X1-X4 (same sequence as Arm and disarm)
   sequence. (It goes back to the ARMED state.) If any wrong button or X4 is pressed during the first 5 push, the system should go back to the ALARM state and the sequence should be reentered.

6. Use as few states as possible without compromising the operation or security of the alarm system.

PreLab:

Part 1 and 2 are due at the beginning of Lab on April 15.

1. Draw the state diagram for the Alarm system design. Indicate what each state represents, the input conditions that cause the state transitions, and the corresponding outputs. Number the state S0, S1, etc.

2. Design an experiment to test your prototype. The experiment should include the list of test vectors. Provide a written justification of your experiments ability to test your prototype.

Note: As with any project specifications not all possibilities may be covered. Feel free to make reasonable assumptions and state them clearly. If you make assumptions because not everything was specified, include them in your pre-lab.

Part 3 and 4 are due at the beginning of Lab on April 22.

3. Implement your design using VHDL code.

4. Implement your test procedure as either a VHDL test bench or a do file to automate its execution.

Experiment:

1. Elaborate the VHDL file from your prelab and correct any possible error(s).

2. Your your test procedure to Simulate and check your simulation results to make sure your circuit is working according to your state table. (Behavioral Simulation)
   (For simulation: assign 40ns period to the input clk and run for 400ns.)

3. Make a constraints file for your design. Use the 7-segment display to indicate what state your lock is in: Unlocked/Reset - Armed - Alarm. Use the five push buttons as the inputs, make the top button X0, the right button X1 the bottom button X2, the left button X3 and the center button X4.

4. Generate a bit file and download your design to a Basys 3 board. Use you test procedure to verify your hardware works as designed.

Make sure report demonstrates all the work you have completed in this laboratory, including designs, simulations and hardware testing along with anything else you did or observed.

Grading for this project:

Pre-lab Parts 1 and 2	**April 15 - 9:00 AM**	**80 pts**
Pre-lab Parts 3 and 4	**April 22 - 9:00 AM**	**80 pts**
Implementation, simulation and hardware verification		**80 pts**
Lab report	**May 6 - 5:00 PM**	**80 pts**
Presentation	**April 29 - 9:00 AM**	**80 pts**

Points will be deducted for late delivery.