Lab 3: Keyboard Scanner

Introduction

In this lab you will learn how to use an FPGA to scan inputs from a keypad.

Learning Objectives

By the end of this lab you will have...

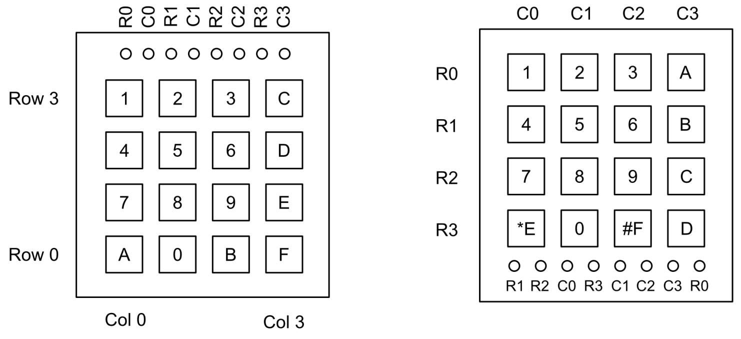
- Designed a circuit interface to read a matrix keypad.
- Understood and implemented a solution to deal with switch bouncing.
- Implemented robust digital design strategies to ensure appropriate behavior with a variety of different user inputs.

Requirements

Design and construct a circuit on your FPGA to read a 4-by-4 matrix keypad. Display the last two hexadecimal digits pressed on your dual seven-segment display, with the most recent entry appearing at the right. Each key press should be recorded exactly once, at the time the key is pressed, no matter how long or short a time the user holds the key down. If multiple keys are pressed, the system can do anything graceful, but should not lock up. The seven-segment displays should not flicker and should be the same brightness regardless of how many segments are displayed.

Discussion

A matrix keypad has four rows and four columns, connected to 8 pins, as shown below for two different models. When you press a key, the corresponding row and column are connected. Check this with a multimeter to confirm your pinout.



Mechanical switches are prone to a phenomenon called switch bounce, where the switch makes and breaks its connection repeatedly on a time scale of microseconds to a few milliseconds. You should design your system in such a way that a single keypress registers only once even if there is some momentary bounce.

This is a thinking person's lab. If you thoroughly understand the problem and design a simple scanner circuit, you can complete the lab fairly efficiently. If you go by trial and error, you may find yourself in lab indefinitely. You will need to generate rather complex stimuli to simulate your keypad scanner. Many students have been tempted to skip simulating and debug on the real hardware. Almost all have regretted it. Similarly, some students have been lured down the path of asynchronous design, at their peril.

What to Turn In

When you are done, have your lab checked off by the instructor. You should thoroughly understand how it works and what would happen if any changes were made. Turn in your lab writeup including the following information:

- Schematics of the breadboarded circuit.
- Your Verilog code and simulation results.
- What strategy did you use for debouncing?
- How many hours did you spend on the lab? This will not count toward your grade.

Hints

Look at your RTL schematic in your synthesis tool (Tools -> Netlist Analyzer). Understand why your code produces the hardware you see. Be sure your combinational logic doesn't have any registers. Be sure your logic has no latches or tristate buffers. The oscilloscope is handy for tracking down timing problems.