Matrix Keypad Decoder

Introduction

In this lab you will design a circuit to scan the matrix keypad and display the digit that is being pressed (0 through 9) on the rightmost digit of the LED display.

You will use the same components as in the previous lab, connected the same way.

Matrix Keypad

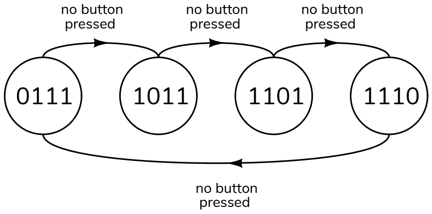
Review the description of the matrix keypad in a previous lab.

Design

In order to determine which button, if any, is being pressed your design must set one row output low and check to see if any of the column inputs is low. If any are low, then the button at the intersection of the low row and the low column is being pressed. If not, then no button along that row is being pressed and your circuit should proceed to the test the next row.

Your circuit should continuously test each of the rows in the order specified below and stop scanning when it detects that a button is being pressed. It should resume scanning when no button is being pressed.

A possible state transition diagram is:



where the state is the binary value of the `row[3:0]` output.

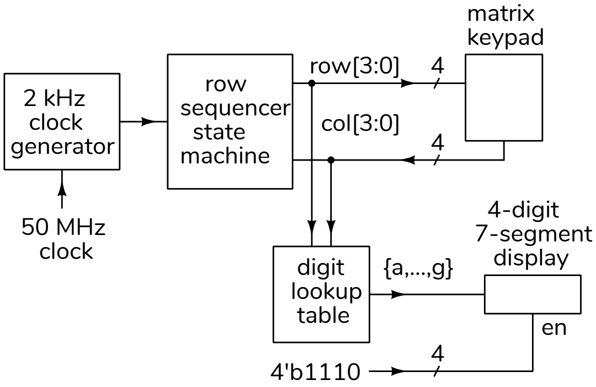
The scanning rate should be 2 kHz, determined by a clock generated as in the previous lab.

The order in which you must scan the rows will depend on the last (rightmost) digit of your student ID as shown in the following table:

last digit of ID	row testing order
0, 1, 2	3, 2, 1, 0
3, 4, 5	3, 1, 2, 0
6, 7	3, 0, 2, 1
8, 9	3, 2, 0, 1

For example, if the last digit of your ID were 0, 1 or 2 then the sequence of states would be as shown in the diagram above.

A possible block diagram for your design is:



Component Connections

The FPGA board, keypad and LED display should be connected as in the previous lab.

Procedure

Follow the general procedure in the Software Installation and Use document on the course website to create a project, compile it and configure the FPGA.

You can import the pin assignments as described in the previous lab. Remember to enable the internal pull-up resistors on the column inputs.

You can instantiate a 2 kHz clock generator as described in the previous lab or, if it's still available, you can add the `.qip` file from the previous lab (**Project / Add/Remove Files in Project... / File name: / `...`** and select the `.qip` file from the previous lab's project directory).

Configure SignalTap for your design as described in the Software Installation and Use document and the accompanying video. Use the 2 kHz clock when configuring the Clock: section of the Signal Configuration tab. Add at least the **`row`** and **`col`** signals in the Setup tab. You need not set a trigger condition for this lab – the logic analyzer will trigger continuously and you'll be able to see the effect of pushing buttons on the keypad.

Connect the keypad and LED to the FPGA board. Compile your design and program the FPGA. Test your design and fix any errors.

Hints

Your design will consist of:

1. A state machine that sequences through four states in the required order.

2. A combinational logic circuit that sets the LED segments corresponding to the row and column that are low. For example if the second column from the left (2) is low when the bottom row (0) is set low then 0 should be displayed on the rightmost digit of the LED display.

Include a SignalTap logic analyzer in your project. It will help you troubleshoot your design by allowing you to view the input and output values and is needed to show that your state machine is testing the rows in the required order for your student ID.

By default the FPGA registers will power up with a value of zero. Make sure your state machine will work properly if it starts at the all-zero state (e.g. design it so that it recovers from any invalid state).

Submissions

Pre-Lab Report

Submit the following to the appropriate Assignment folder on the course website:

1. A PDF document containing:

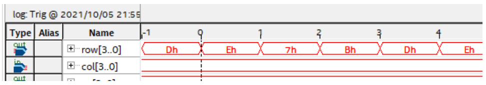
   - A state transition diagram appropriate for your ID. The state transition conditions should be Verilog expressions rather than the phrase "no button pressed" as in the diagram above.
   - A listing of your Verilog code.

Post-Lab Report

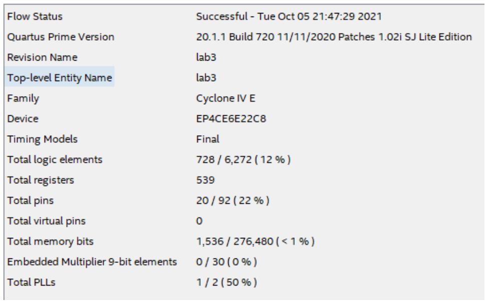
Submit the following to the appropriate Assignment folder on the course website:

1. A PDF document containing:

   - A listing of your Verilog code.
   - A screen capture of the SignalTap window showing a sequence of at least five values on the **`row`** output. The order the rows are tested should match the order required for your ID as shown in the table above. Display the row values in hexadecimal (right-click on a row to get a menu). For example:



   - A screen capture of your compilation report. For example:



Flow Status	Successful - Tue Oct 05 21:47:29 2021
Quartus Prime Version	20.1.1 Build 720 11/11/2020 Patches 1.02i SJ Lite Edition
Revision Name	lab3
Top-level Entity Name	lab3
Family	Cyclone IV E
Device	EP4CE6E22C8
Timing Models	Final
Total logic elements	728 / 6,272 (12 %)
Total registers	539
Total pins	20 / 92 (22 %)
Total virtual pins	0
Total memory bits	1,536 / 276,480 (< 1 %)
Embedded Multiplier 9-bit elements	0 / 30 (0 %)
Total PLLs	1 / 2 (50 %)

2. If you were not able to demonstrate your solution to the lab instructor during your scheduled lab period, submit a video showing the keypad and the LED display as you push the keys 0 through 9 and the A, D and "*"' keys.

Follow the *Report and Video Guidelines* and *Coding Guidelines* documents on the course website.