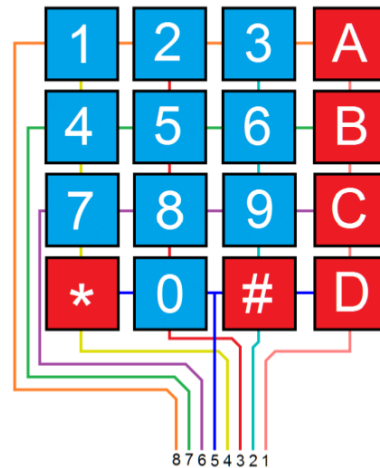
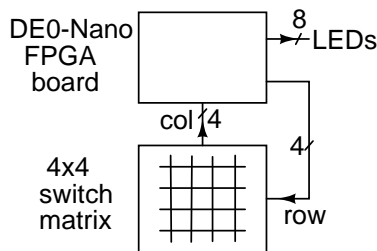


Keypad Decoder

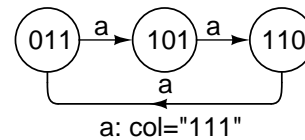
Introduction

In this lab you will design a decoder for a 3x3 switch-matrix numeric keypad. Your decoder will show the binary value of the key being pressed on the FPGA board's LEDs:



The three leftmost column conductors, keypad pins 4 through 2, are connected to FPGA inputs that are configured with pull-up resistors so they will read high (1) unless pulled low.

A simple¹ decoder can be designed as a state machine that cycles through three states. In each state a different row is set low (0). Pressing a switch will drive one of the column inputs low. If any of the columns is low then the state machine stays in the current state. Otherwise it goes to the next state.



We can design combinational logic to obtain the binary value of the key being pressed from the current row and column values.

Requirements

All LEDs should be off if no key is being pressed. When a key is pressed the binary value of the key should be displayed on LED(3) through LED(0).

You need only decode the keys between 1 and 9 and you may assume that only one key at a time will be pressed.

Decoder Operation

The keypad contains 16 switches arranged in a 4x4 grid. We will only use the 3x3 portion consisting of the digits 1 through 9. Each switch is connected to one row and one column. Pressing a switch makes a connection between the corresponding row and column conductors:

Keypad-FPGA Interface

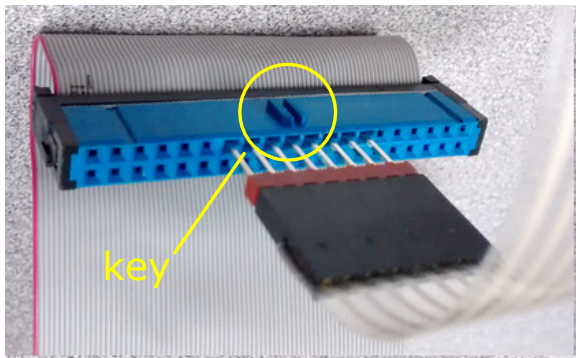
A ribbon cable cable connects the 8 pins of the keypad ribbon cable to 8 general purpose I/O (GPIO) pins on the DE0-Nano FPGA board.

¹A more complete design would have to debounce the switches and register key presses.

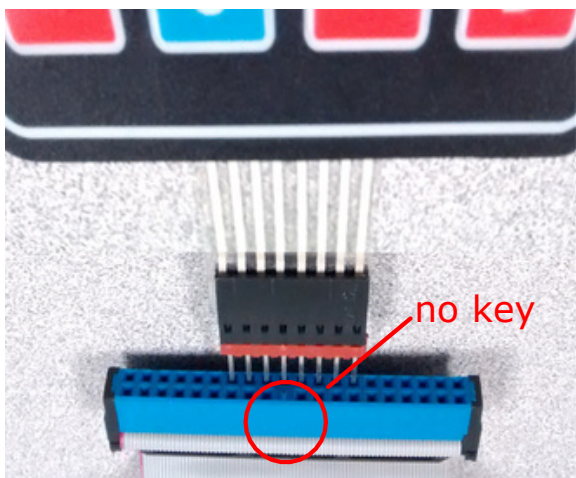
The lab5.qsf settings file assigns the names row(3 downto 1) and col(3 downto 1) to the 6 upper left row/column pins and configures col(3 downto 1) with internal pull-up resistors.

Connector Details

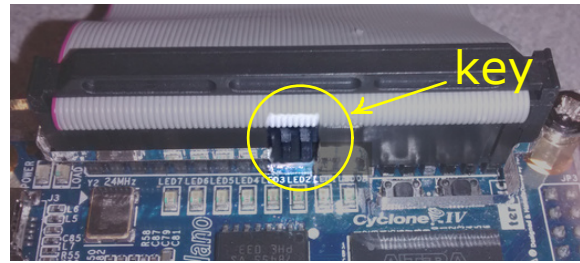
Connect the keypad to the blue ribbon cable connector using the row of connectors on the side with the “key.” Center the pins so that there are 6 unused positions on either side:



The keypad should be facing away from the connector key:



Connect the black ribbon cable connector to the DE0-Nano board connector on the side with the USB connector. Ensure the ‘key’ on the black ribbon cable connector fits in the slot on the Plexiglass cover on the board:



The connections to the ribbon cable can be confusing. See the diagrams and tables at the end of this document for more details.

Warning

Ask the instructor to verify that your connections are correct before connecting the USB cable. Incorrect connections can cause damage to the board and result in you receiving a mark of zero for failure to follow instructions.

Pre-Lab

Write a VHDL entity and architecture that meets the requirements above.

To obtain a clock your top-level entity should use the lab5pkg package:

```
use work.lab5pkg.all;
```

which declares the same clock component you used in Lab 3:

```
component clock is
  port ( inclk0 : in std_logic ; c0 : out std_logic );
end component ;
```

This is a PLL that outputs a 10 kHz clock, c0, if supplied a 50 MHz clock input, inclk0.

Submit a report to the appropriate dropbox with a listing of your VHDL code.

Procedure

Follow the instructions in Lab 1 but download lab5.zip from the course web site instead of DE0_NanoX3330.qsf.zip and import the assignments from lab5.qsf instead of DE0_NanoX3330.qsf.

Also add the following instruction before compiling your code:

- select Project > Add/Remove Files in Project... and add the file `clock.vhd` to your project.

Test your design and demonstrate it to the instructor to get credit for completing the lab.

Lab Report

Submit a report to the appropriate dropbox with your working VHDL code.

Hints and FAQs

Q: How do I reset the state machine?

A: This is up to you. You can use the fact that the FPGA's registers power up in the all-zeros state, you can detect invalid states, or you can use one of the pushbuttons on the FPGA board (e.g. `KEY(0)`) as a reset input.

Q: How do I implement the “combinational logic”?

A: One way is to use a selected assignment with 9 choices plus a default. Perhaps something like:

```
with row_q & col select
  led(3 downto 0) <=
    x"1" when "011011",
    ...
    x"0" when others ;
```

where `row_q` and `col` are the row output and column input respectively.

Q: How do I detect if a button is not pushed?

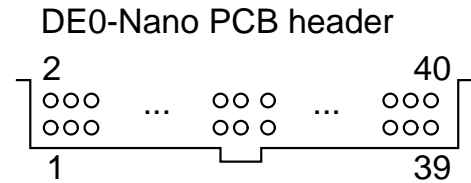
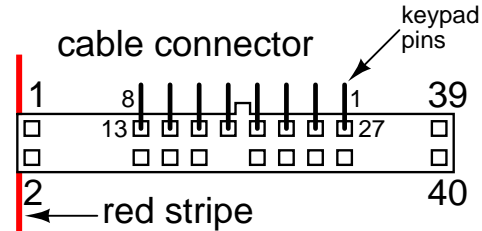
A: `col = "111"?`

Q: How do I troubleshoot my design?

A: You can (temporarily) connect signals to the LEDs. You can also use the SignalTap II logic analyzer (see Appendix).

Connector Pinouts

The following diagrams show the connector pinouts:



The following table shows the correspondence between the keypad row or column, the ribbon cable connector pin and the corresponding GPIO-0 connector pin:

GPIO name	FPGA pin	keypad pin	cable pin	VHDL signal
GPIO_08	A5	8	13	row(3)
GPIO_010	B6	7	15	row(2)
GPIO_012	B7	6	17	row(1)
GPIO_016	C8	4	21	col(3)
GPIO_018	E7	3	23	col(2)
GPIO_020	E8	2	25	col(1)

Using the SignalTap II Logic Analyzer

This tool adds logic to your design that allows you to capture the real-time behaviour of signals inside the FPGA.

- select Tools -> Signal Tap II Logic Analyzer
- under JTAG Chain Configuration -> Setup... select USB Blaster
- under SOF Manager click on ... and select the correct .sof (FPGA serial programming) file in the output_files folder
- click on the download (“Program Device”) icon to make sure the device is recognized and can be programmed.

Type	Alias	Name	-4	-3	-2	-1	0	1	2	3	4	5	6
in		col[3..1]			111b								011b
out		led[3..0]			0000b								0001b
out		row[3..1]	101b	011b	110b	101b							011b

- under Signal Configuration: Clock: click on ... to use the node finder to list the signals that can be monitored. Click on the “Show more search options” button beside List if necessary and select SignalTapII:pre-synthesis as the filter option. Add your clock signal as the clock.
- in the Setup tab of the main window double-click in the indicated area to add signals. As before, list the pre-synthesis signals and add the one you want to view to the list of signals being monitored in the Setup tab.
- in the Trigger Condition column enter the signal values that you want to trigger on. For example, if you want to see what happens when you press button '0', set the col(3) trigger condition to “Falling Edge” and row(3) to “Low.” There are many other ways to configure trigger conditions.
- recompile and reprogram the FPGA (answer yes when asked to save the .stp file and add it to the project)
- click on “Autorun Analysis” to start the logic analyzer
- the logic analyzer should display the signals being monitored in the Data tab each time the trigger condition is met
- you can right-click on a trace to and change the display format (e.g. to binary) to make it easier to interpret the traces

For example, the screen capture above shows the row, column and LED signals for a few clock cycles before and after the '1' button is pushed.