

MATRIX KEYPAD SCANNER

Overview

This reference design illustrates the implementation of a keypad matrix scanner using a Silicon Blue iCE65 FPGA. The keypad scanner HDL code is a generic implementation, which supports any number of rows and columns. This document describes 7x7 matrix keypad scanner, commonly used to accept inputs from a QWERTY keypad, generally found in PDAs and other hand held devices. The design includes functionalities such as user definable key debounce time setting and auto sleep feature for conserving power.

Description

A keypad forms an important part of the user interface of any electronic device. It is generally implemented using a grid of rows and columns, as shown in Figure 1. An event such as a key press or a key release can be decoded by driving and reading these rows and columns in a specific manner.

In this application example, the columns have been configured as outputs and the rows as inputs. All the row lines have been pulled up using pull-up resistors near the iCE65 FPGA inputs. On power-up, or on reset, all the columns are driven low and the status of all the seven rows is constantly monitored. This state is also known as the keypad scanner's "Sleep Mode", in which the dynamic scanning of every line is suspended to conserve power.

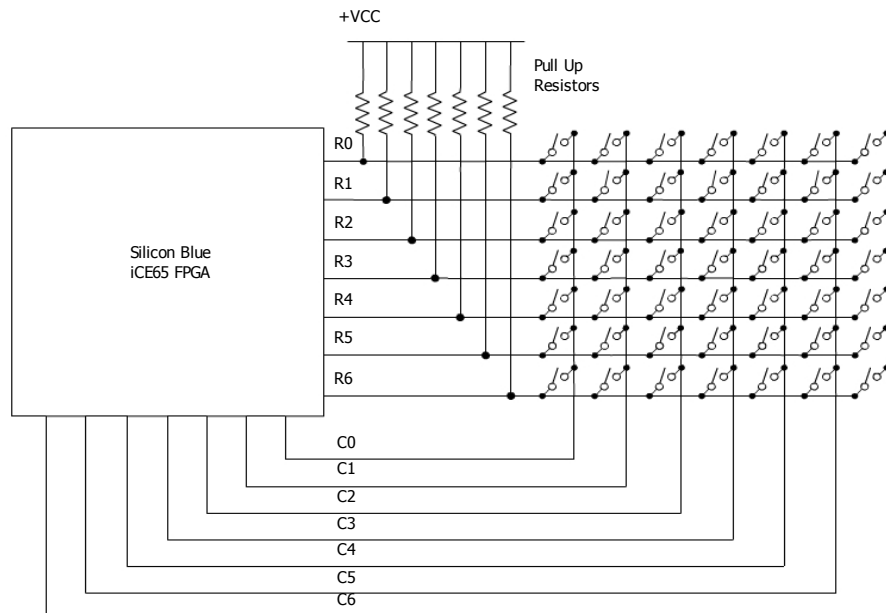


Fig 1: Keypad Matrix Arrangement

Implementation

Whenever a key is pressed, the row and column corresponding to that key get connected, causing its row input to go low. The keypad scanner exits its sleep mode and starts actively scanning each column one by one to determine the location of the key that was pressed. The column which is being scanned is made low while others are tri-stated. If the key which is pressed lies in this column, one of the row inputs will go low, revealing its location. When the required column is detected, the scanner maintains it at ground and checks whether the same row is still low after a small delay. This is done in order to debounce the key press.

After the location of the key has been rightly determined and the press has been debounced, the keypad scanner outputs its location in terms of its row and column (3-bits each in the case of a 7x7 matrix) address along with a valid bit. The valid bit remains high as long as the key is pressed. Upon release of the key, this bit returns to zero. The key release event is debounced using a small delay, similar to the case of a key press.

After waking up, the keypad scanner continues to scan each column repeatedly for sometime before it goes back into its sleep mode. The debounce delay and the autosleep time (after which the scanner enters the sleep mode) can be modified as per design requirements in the VHDL files of the scanner by changing the value of the constants "debounce_time" and "sleep_time". The interface diagram of keypad scanner is shown in Figure 2 and Table 1. provides the port description.

This example is designed to accept a single key press at a time. If a key is pressed while another is being held down, it will only be recognized after the key that was pressed first is released. Also, the parameters "row_size" and "col_size" need to be specified in the VHDL design files to suitably adapt the code for the required keypad matrix needed to be scanned.

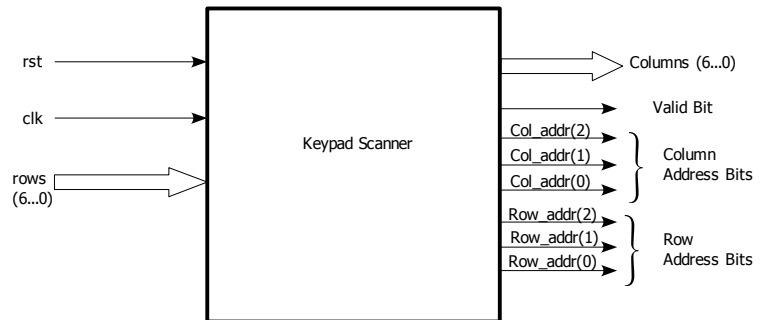


Fig 2: Keypad Scanner Interface Diagram

Port	Direction	Description
rst	Input	Active high reset signal. A high pulse to this input resets the keypad scanner module.
clk	Input	Keypad scanner system clock input.
rows	Input	7 bit wide row input bus with pull-ups
columns	Output	7 bit wide column output bus
valid	Output	Active high valid bit. Indicates that a key whose address is available on the col_addr and row_addr buses is pressed. It goes low once before a new pressed key address is made available.
col_addr	Output	3 bit wide column address bus. Gives the column address of the key which is being pressed whenever the valid line is high
row_addr	Output	3 bit wide row address bus. Gives the row address of the key which is being pressed whenever the valid line is high

Table 1: Pin Description

Before pressing any key, initialize keypad scanner by supplying a high pulse to its 'rst' input. A key press is indicated whenever the valid line is high. This bit will remain high as long as the key remains pressed. During this time, the row_addr and col_addr buses provide the matrix address of the pressed key in terms of its row and column number.

The keypad scanner design listed in this application note when implemented using Silicon Blue device iCE65L04-UCB284C for a keypad matrix of size 7x7, runs at a frequency of 32 kHz utilizing 50 Flip-Flops and 153 LUTs (Post P&R data).

Conclusion

This example design demonstrates the implementation of a keypad scanner on iCE FPGAs. iCE FPGAs, due to their very low power capabilities, are ideal for implementing applications requiring a keypad, such as hand held and mobile devices where power saving is of paramount importance.

Disclaimer

SiliconBlue is providing this document, design example, or information "as is." SiliconBlue does not ensure that this implementation or information is void of any claims of infringement. As a reader / implementer, you are responsible for obtaining any rights you may require for your implementation. SiliconBlue also does not warrant the fitness of this design example to be readily implemented in any specific context. It is your sole responsibility to validate this design example and its correctness while implementing it for your requirement.