STUDY OF STATE-MACHINE BY DESIGNING A CODE ENTRY SYSTEM

Ibrahim Benayad
Doctoral Degree Programme (1), FEEC BUT
E-mail:benaya00@stud.feec.vutbr.cz

Supervised: Vladislav Musil
E-mail: Musil@feec.vutbr.cz

ABSTRACT

The purpose of this paper is to introduce some of the basic circuits used in digital electronics, such as binary number, Boolean algebra, Karnaugh map, and state machine, and used them in the design of a practical application. The paper is centered on the design of a keypad entry system, and uses both combinatorial and sequential logics. The circuit to design is an entirely digital one and should read a sequence of digits entered on a four by three keypad matrix. It should detect the sequence of digits pressed and indicate when that sequence is identical to an internally stored sequence of digits. Additionally, this project will look at the concept of finite state machine and how those can be implemented in a digital design, in particular with regards to the way states are encoded in a digital form. Finally, this paper looks at how the design can be implemented in an FPGA.

1 INTRODUCTION

A 3x4 matrix keypad is often used others in mobile phone and building entry control.This project objective is to design an interface for such a keypad that could be used for example in building entry control system. The task is to detect whenever a sequence of key is pressed on the keypad, and check that it matches a pre-defined code. This code is four digits long.

The 3x4 matrix keypad has the following keys:

- Digits 0-9
- Key #
- Key *

A press on key * acts as a backspace to cancel the previous digit entered
A press on key # validates the code once four digits have been entered.
The outputs of the system are 6 LEDs, indicating the state of the system

- Two LEDS are used to indicate whether the code entered is correct or wrong
- Four LEDs are used by the system to indicate how many digits have been entered in the code: Each time a new digit is entered, one more LED light up.

2 DESIGN OF KEYPAD DECODING CIRCUIT:

In this section, the circuit used for decoding the keypad is described and explained. This circuit is mainly combinatorial, with only some counters being the synchronous part the design. First, we will explain how the 3x4 matrix keypad operates, and then we will explain how the decoding circuit operates.

2.1 KEYPAD OPERATION

In order to operate the keypad, a process called keypad decoding is used to determine the key that is pressed. As seen from the above figure it can be observed that the matrix keypad requires both outputs and inputs. Matrix keypads are made this way since they are easier to manufacture. Each keyboard operates in such a way that when a key is depressed, there is an open circuit in the connection between the row and column.The first part of my paper is to design a logic circuit that will scan the keypad and produce a 4-bits output that will indicate which key is pressed.

The output code corresponding to each key pressed is summarised in pod:

Key	Code
None	1111
0	0000
1	0001
2	0010
3	0011
4	0100
5	0101
6	0110
7	0111
8	1000
9	1001
#	1010
*	1011

Tab. 1: *Code corresponding to each key*

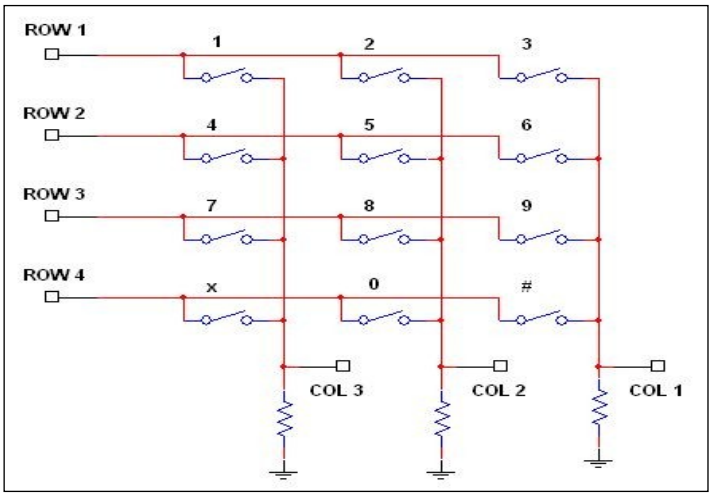

Fig. 1: *Schematic of a 4x3 matrix keypad*

2.2 KEY ENCODING

On its output, the keypad will produce a code corresponding to the column of the key pressed. This code for the columns and the binary number of the row being scanned is then used to determine which key is pressed. This is summarised in the following truth table where A1 and A2 indicates the binary number of the row being scanned (Output of binary counter) and B2, B1, B0 are the output from the keypad corresponding to the column of the key pressed. D3, D2, D1 and D0 make up the 4-bits code that correspond to the key pressed.

2.3 SIMULATION OF CIRCUIT IN VHDL

The logic circuits designed previously have been described using VHDL, and simulated with Modelsim to ensure the design performed according to the specified functionality. A brief description of the VHDL language follows.

2.3.1 VHDL DESCRIPTION

VHDL stands for Very high-speed integrated circuit Hardware Description Language. It is one of the programming language used to model a digital system by dataflow, behavioral and structural style of modeling. This language was first introduced in 1981 for the department of Defense (DoD) under the VHSIC program. In 1983 IBM, Texas instruments and Intermetrics started to develop this language. In 1985 VHDL 7.2 version was released. In1987IEEEstandardizedthelanguage.

2.3.2 KEYPAD DECODING CIRCUIT IN VHDL

Each part of the design has been described in VHDL. Each circuit is described as a separate component and then all components are brought together in a single component. Several components have been written, each corresponding to a part of the design:

- Component *counter*
- Component *counter_decoder*
- Component *keypad_encoder*

All those components are connected together in a single component, *key_decoder*.

2.3.3 SIMULATION IN VHDL

To simulate the design in VHDL, we need to create a test bench, and design a component to simulate the 3x4 matrix keypad.

Keypad simulation

A special component is created to simulate the 3x4 matrix keypad. This component has 16 inputs and 3 outputs**:**

- Four inputs corresponding to each individual rows of the keypad,
- Twelve inputs corresponding to each individual key. Whenever one of those inputs is high, the simulated design acts as if the corresponding key is pressed. Whenever one of those inputs is low, the simulated design acts as if the corresponding key is released.
- Three outputs corresponding to the three column outputs of the keypad.

Simulation of behavioural model

We can simulate the designing using the integrated ModelSim flow. We can run processes from within ISE which launches the installed ModelSim simulator.

3 CODE DETECTION USING FINITE STATE MACHINE:

In this section, we will give some introductory material about Finite State Machine and explain their operation, then explain how our system could be described using a Finite State Machine. Finally, we will explain how the Finite state Machine is described in VHDL, and will look at the advantages of different types of state encoding.

3.1 STATE MACHINE

The state machine, also called a sequential machine, is used in a system that can be described in terms of a set of states that the system may enter. Once in a particular state, the system must be capable of remaining in that state for some finite period of time even if the system inputs change. This requirement dictates memory capability for the state machine. Furthermore, the state machine must have a set of inputs and a set of outputs.
It was earlier stated that a sequential machine or state machine must possess memory capability. A large majority of practical state machines use clocked flip-flops as the storage elements. The code that defines each state then corresponds directly to the code contained by the flip-flops.Another important characteristic of the state machine depends on whether the system is clock-driven or not. In many digital systems a timing reference signal is required. Some sort of stable multi-vibrator is generally used to produce a continuous clock signal. We refer to a variable as clock driven if that variable changes value only at the time of a clock transition. It need not change at every clock transition to be clock-driven, but when it does change it must do so as the clock is changing values. Synchronous systems are the easiest to design and asynchronous systems are the most difficult.

3.2 FINITE STATE MACHINE USED FOR CODE DETECTION

In the paper to detect a sequence of digits entered on the keypad. We used a finite state machine; it will describe the whole operations for all steps where the finite state machine detected the numbers. so we designed the FSM code entry detection , it gives us all states further to get the number detected which selected on the first step in the FSM.

3.3 STATE ENCODING

Each state of state machine can be represented with unique pattern of high and low register output signals, a process called encoding. The two primary encoding methods
are binary and one hot encoding.

3.4 TESTING

Once the design is downloaded in the FPGA, it can be tested by connecting the keypad to connector B1. After testing, it appears that the design will not function properly. This was because the on board clock was running at 50 MHz, and this was too fast to operate the keypad. A modification was done to the design so that the clocked is slowed down to about 1 kHz by means of a counter. After this was performed, the design could be tested again. Some fine-tuning in the Finite State Machine had also to be carried out. However, the final design operates correctly with the keypad, and meets the required objectives of the design.

4 CONCLUSION

This paper dealt with the design of a keypad encoding circuit in a first part. In this part, we used mainly combinatorial circuit and applied the Karnaugh map to simplify complex logic equation. We used simulation tools to verify the correct operation of our design and make any changes before implementation. Having checked the design through simulation before designing the second part of the paper, we were relatively confident that this section would work inside an FPGA.In the second part of this paper, we designed a state machine to detect the code entered. It included many states to get all operations required by the project.We looked at the different encoding possible for implementing the finite state Machine. After connecting the keypad to the connector, we could verify the correct operation of the design. As an improvement to design, we notice that the keypad decoding function will not take into account when two keys are pressed at the same time. This could be an area for future improvement of the design.

REFERENCES

[1] PEDRONI, V. Circuit Design with VHDL (5th edition), The MIT Press, Aug 2004, pp. 25-90, 159-186

[2] COMER, D. Digital Logic and State Machine Design, Oxford University Press, USA, 3rd Edition, Dec 1994, pp. 208-305

[3] http://www.utdallas.edu/~shankars/teaching/ee5325/foils/lectures/lecture11.pd

[4] http://www.altera.com/literature/ab/ab131_01.pdf