Keyboard Input to Seven Segment Display

Chris Mathewson, Jason Mansour, Kristine Miller

Electrical and Computer Engineering Department
School of Engineering and Computer Science
Oakland University, Rochester, MI
e-mails:cmathewson@oakland.edu, kamiller234567@oakland.edu, jmmansour@oakland.edu

***Abstract*—We will be constructing a keyboard to seven segment display. In this report we will be covering the tools used to build this project and why it was used.**

I. Introduction

This project will cover all the material learned in throughout the course. This is also the motivation behind the creation of this project, to show all the knowledge retained throughout this course. This project will contain topics from the course such as state machines, shift registers, as well as all the VHDL covered in this course. This project's applications can be found as an example of the tools learned in practice for all student studying VHDL.

II. Methodology

A. Keyboard PS/2 input

The first task of the project was to be able to convert a keyboard input into usable data. Using a microcontroller on the board, PIC24FJ128, the USB signal from the keyboard was separated into two signals ps2c and ps2d emulating a PS/2 device. To capture the input sent from the ps2d signal, two finite state machines had to be created, a falling edge detector and the main FSM. When a falling edge was detected from the ps2c, a modulo-10 counter was used to count every time there was another falling edge, collecting data into a 10-bit shift register. However, prior to this, a filter using a shift register collects the ps2c signal every clock cycle. This filter checks to see if 8 consecutive bits are the same, and then passes the ps2c value as ps2cf. Once the modulo 10-bit is finished counting, the main FSM is used to send a done signal to a D-Flip-Flop that holds the done signal for an extra clock cycle. The done signal then allows for the data to be collected.

B. PS/2 Decoder and 16-bit Shift Register

Once the data was collected, the data was run through an FSM to check to see if it was a key-up signal. If the key-up signal was detected, the FSM sent an enable signal to the 16-shift register when the next done signal was obtained. At the same time, a PS/2 decoder takes the 8 bits of data needed and decodes it using the PS/2 key codes into a 4-bit value. If a value obtained was not a number 0-9, when received by the 16-bit shift register, it is ignored. To preserve the BCD values of the data, the 16-bit shift register shifts four spaces to the left.

C. Seven-Segment Display

A serializer is used to alternate the inputs and enables for the seven-segment displays. Using a pulse generator, the time to switch each display is around 1 ms. An array is used to coordinate each BCD value to the corresponding display, and multiplexers are used to determine the proper enables and input values for the displays. A basic modulo-8 counter was used to change the priority of the displays.

III. Experimental Setup

The setup used to verify the functionality of the project was the Nexys-4 DDR Artix-7 FPGA and Vivado Design Suite from Xilinx. On the board, the microcontroller PIC24FJ128 was used along with the USB port as an input from a keyboard, and the seven-segment displays were used as outputs.

IV. Results

The results for the keyboard to seven-segment display initially turned out more difficult than anticipated. On the first stage of debugging, the keyboard input would pick up random zeros and numbers depending on the key pressed. It wasn't until much later into development that a key-up FSM was implemented along with a ps2c filter that the issue became issue. Another issue that was very minor was the serializer was not utilizing all eight display; it was only using the first four. After delving into the code, the error turned out to be a simple logical error in the modulo-8 counter. Having a completely functional project, the eight seven-segment displays show the values of the keys pressed while being shifted to the left with each new value. The keyboard input is taken when the key is released, not when it is pressed.

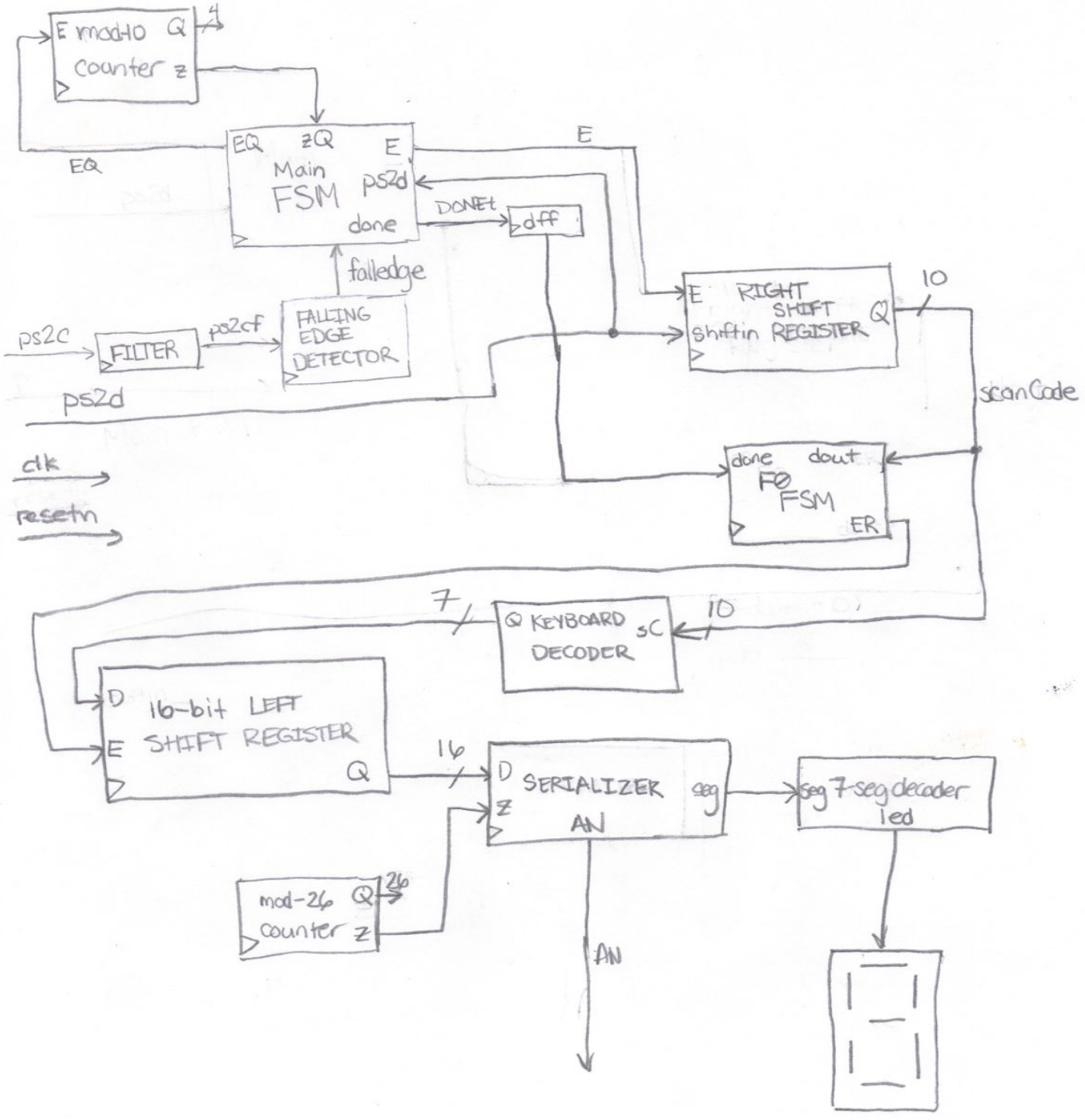
Conclusions

The main current take-away is how incredibly advanced a simple keyboard to seven segment display is. The sheer amount of work that goes into programming an FPGA to do a simple task such as take an input from a keyboard is overwhelming. Progress was slow and did not go as planned; however, the knowledge gained was worth it.

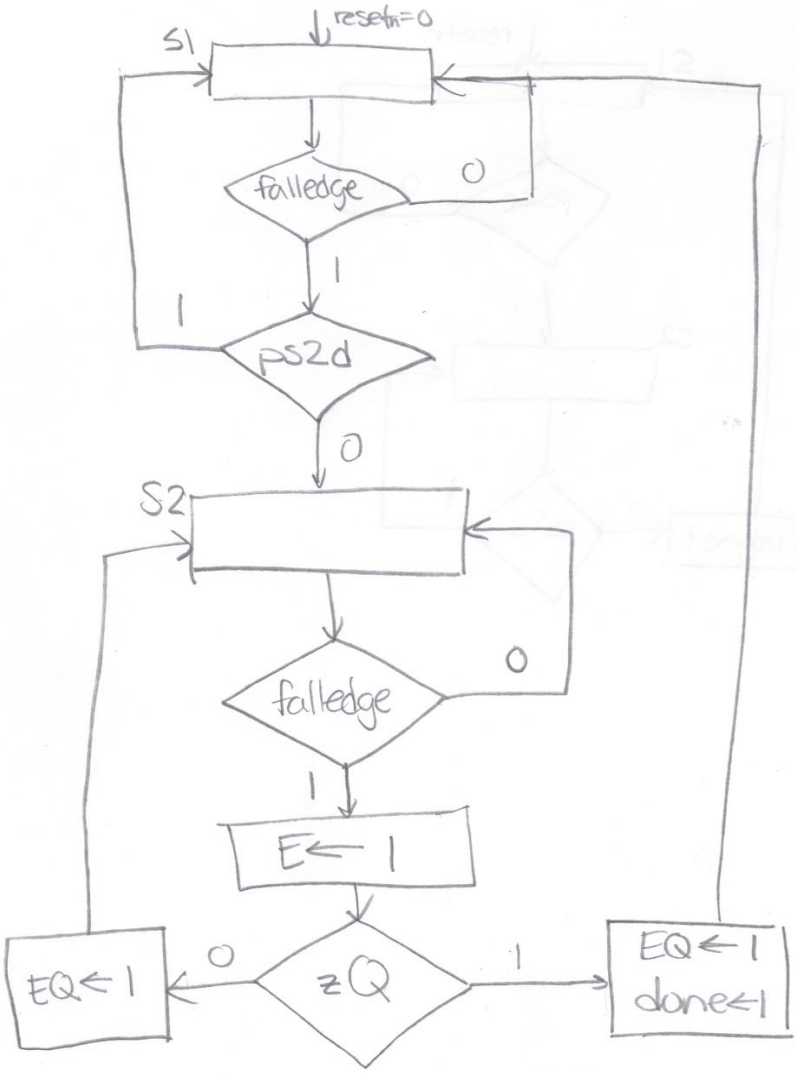
References

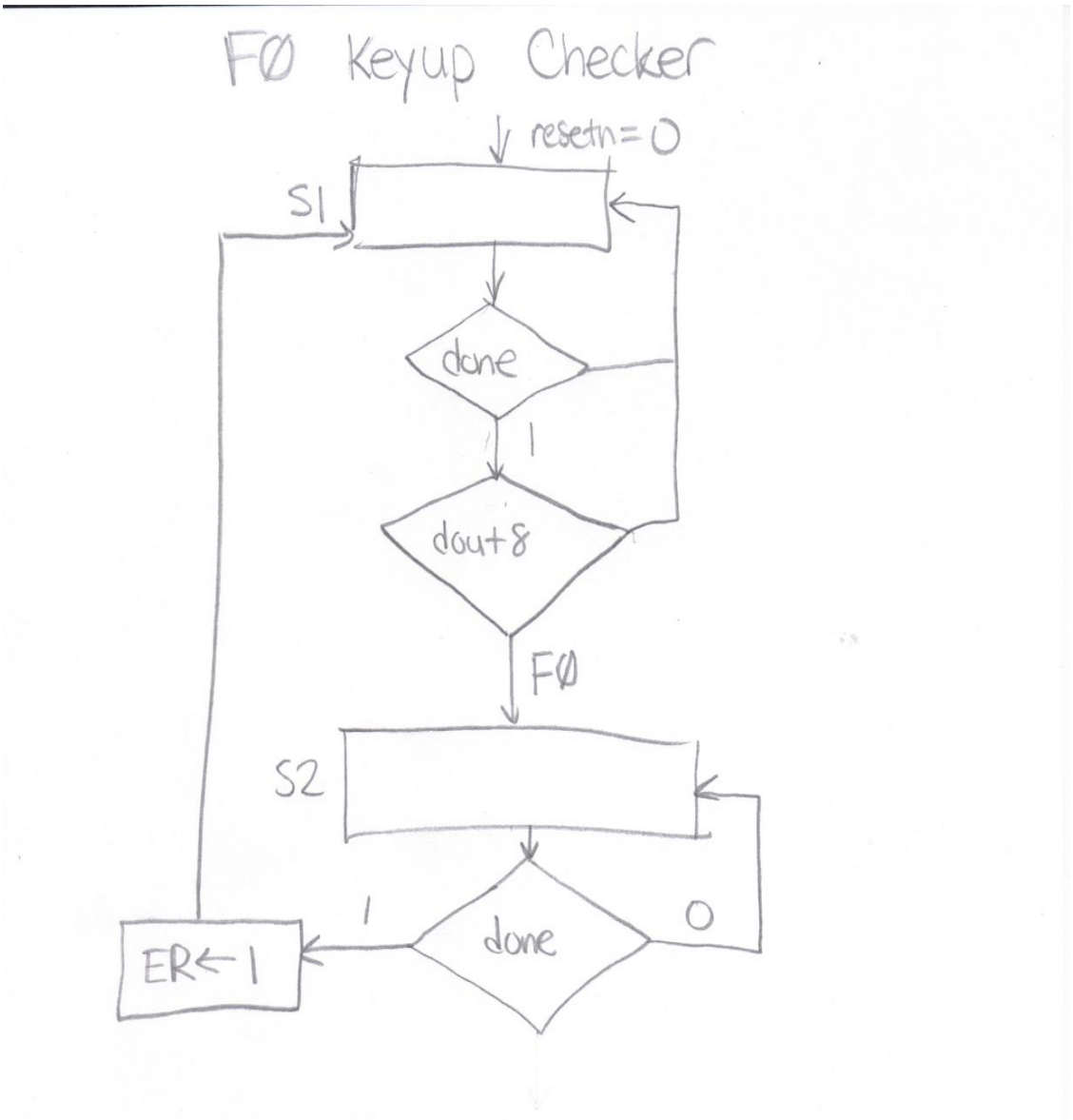
[1] Daniel Llamocca. Assistant Professor, Electrical and Computer Engineering Department, School of Engineering and Computer Science, Oakland University.
http://www.secs.oakland.edu/~llamocca/VHDLforFPGAs.html

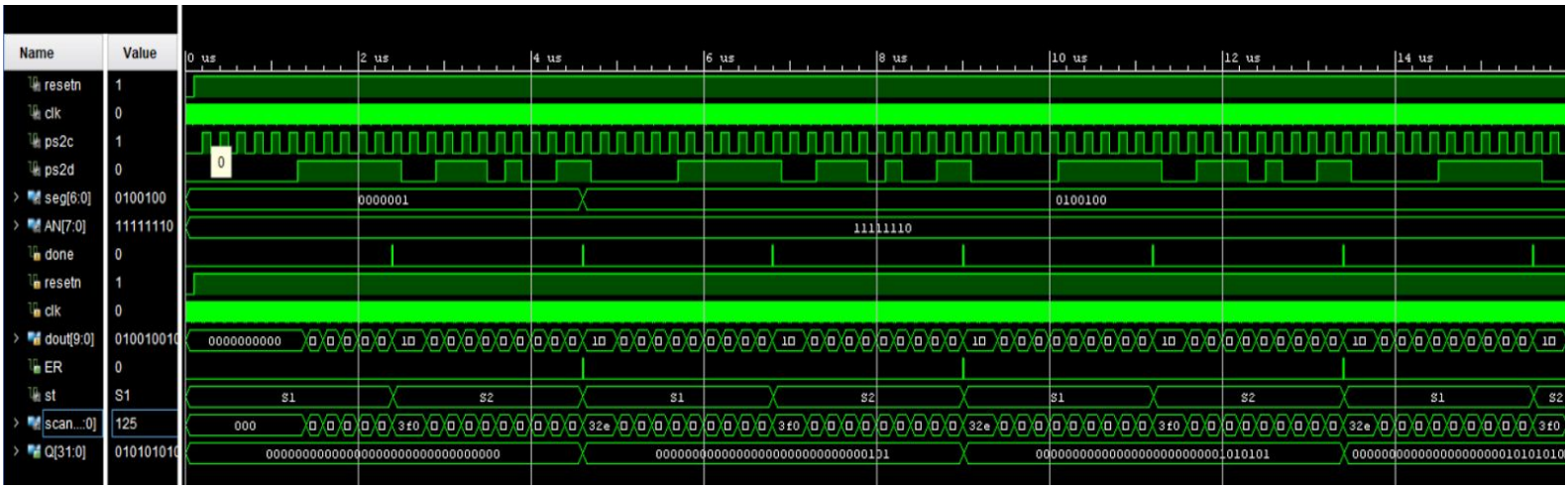
PS/2 Keyboard to Seven-Segment Display



MAIN FSM


reset_n=0
S1
falledge
0
1
1
ps2d
0
S2
falledge
0
1
E⇐1
EQ⇐1
0
zQ
1
EQ⇐1
done⇐1


FØ Keyup Checker
resetn=0
S1
done
1
dout+8
FØ
S2
done
1
0
ER←1


Name
Value
resetn
clk
ps2c
ps2d
seg[6:0]
AN[7:0]
done
resetn
clk
dout[9:0]
ER
st
scan...:0]
Q[31:0]
0 us
2 us
4 us
6 us
8 us
10 us
12 us
14 us
0000001
0100100
11111110
S1
S2