CST 345 Midterm

By: Cal Barkman

Conclusion maybe?:

In both problems, I used a 1hz clock attached to a synchronizer attached to the Picoblaze interrupt input. That allows me to have a 1 second counter for turning on LEDS in increments of 1 second. For the keypad input, I took the keypad I had from lab 04 and concatenated the interrupt line with the data and fed it as a 5 bit value to the in port of the Picoblaze. Using this value, I made a small checker in software that would detect when a key was pressed, and when it was released, making it easy to process keypresses in just software. This simple machine allows for a large range of configuration for timer and keypad control measures, but is not extendable much beyond that without making an interrupt controller or altering the timer counter to be in hardware that raw values are read from. It also made it relatively easy to do both problems as once I had one mostly complete the other was much easier to finish since the hardware didn’t have to change at all.

**High level hardware block diagram**



**Design of hardware/software interfaces**

7 seg display interface diagram:



Keypad controller interface diagram:



**Software documentation**

Problem 1

Main loop and interrupt service routine:



Flag changed routine:



Problem 2

Main loop and reset routine:



Flag changed routine:

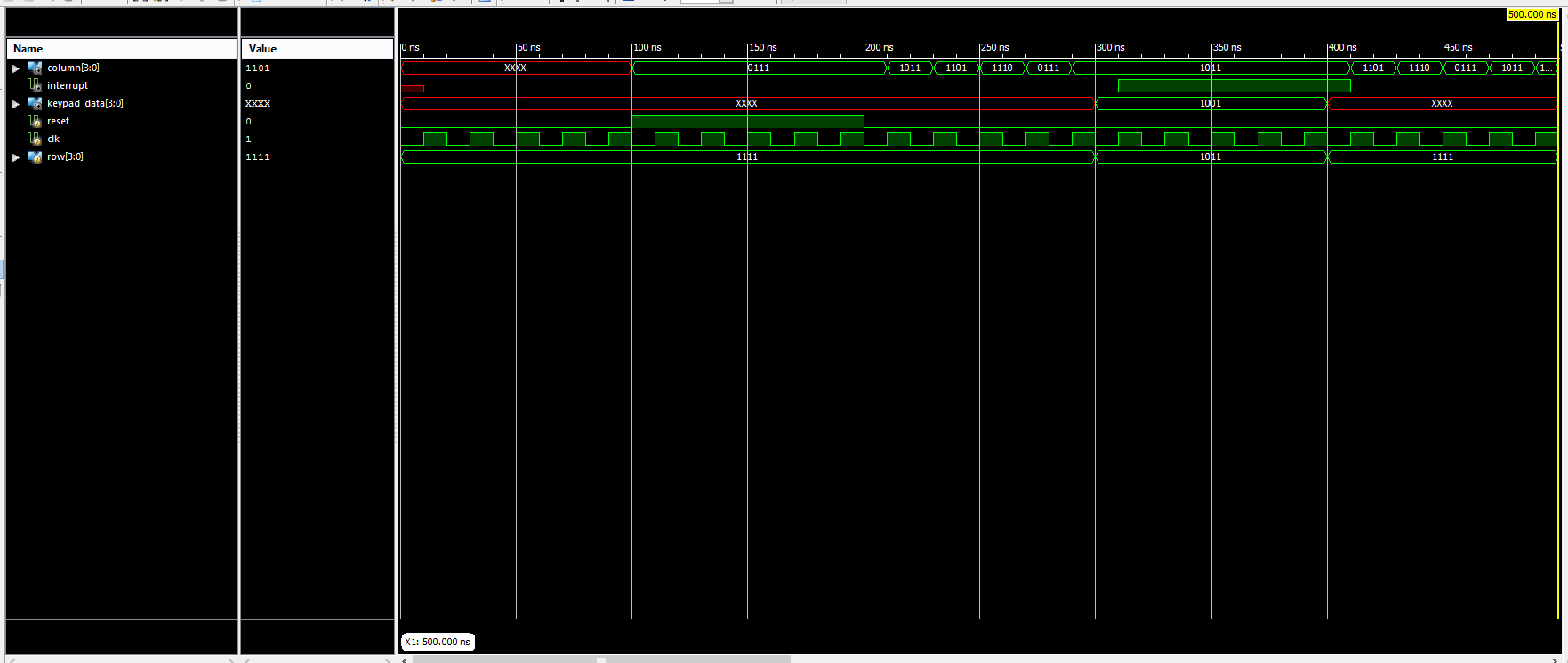


Interrupt service routine



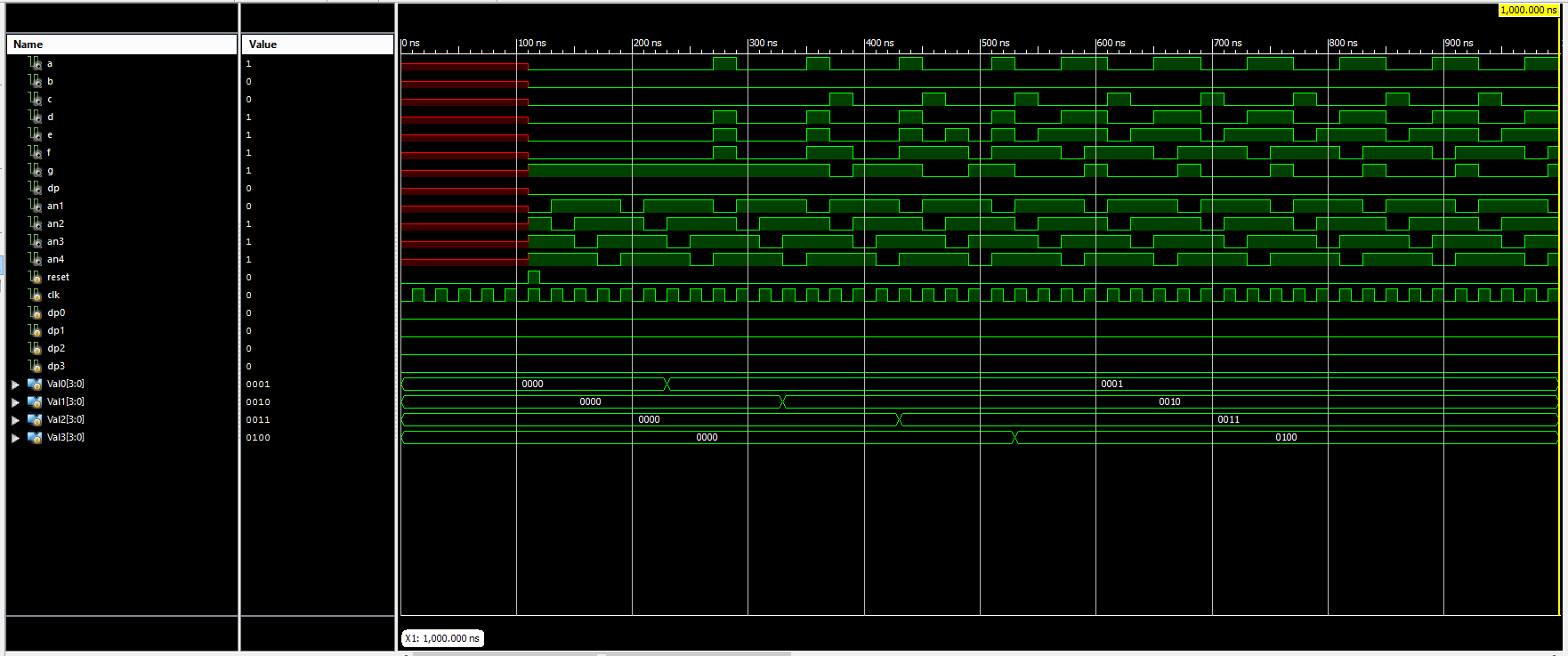
**Results of hardware and software simulation**

**Hardware simulations:**

Keypad simulation:

Higher res version included in zip as KeypadSimulation.PNG

Muxed 7 seg display simulation:



Higher res version included in zip as KeypadSimulation.PNG

**Software test steps:**

The software test videos are included in the zip.