

**Steven Phan**

**014358810**

**CECS 460**

**Project 1 TramelBlaze Counter**

This lab allowed me to review the basics skills and coding practices I have learned over the semesters in computer hardware designing. Using a combination of eight modules in a top-level design and a single clock input, I was able to create a push button counter. I used two external push buttons and one slide switch as inputs for the design. The outputs would be displayed onto the 7-segment display in hexadecimal.

The TramelBlaze processor requires three memories to run. The main engine of the processor requires a 512x16 scratch pad ram and a 128x16 stack ram in order to handle the interrupts. This was created using IP corgen. The engine would use an external rom memory of about 4k x 16. This memory would be initialized with a coefficient file. Using the trambler assembler, I created a counter in assembly for a tba file. In the assembly code, I initialized register 9 to start off with 0’s. When an input from the button goes high, we going into the interrupt service routine. It first checks to see if the switch is either high or low to determine the direction of the count. The register increments by one if the return address is a one, or decrements if it’s a 0.

Using mechanical buttons create a debounce of signals that can oscillate for up to 20 ms. To accommodate for these oscillations, two modules used. The first module called debounce would debounce these signals. Using Pong Chu’s state transition diagram as a reference, I created the eight states that takes inputs of a switch and tick. The 10 ns tick is generated from the example shown in class. Its output would only go high when the signal from the button settles down. A wire would connect the debounce output to the input of the positive edge detector module. The edge detector’s job is to create a pulse signal one clock wide after receive input from the debounce.

A SRQ flop would take inputs from the positive edge detector and interrupt acknowledge. When the positive edge detector is high, the output from the SRQ flop is 1. However, when the interrupt acknowledge goes high, the output from the flop is a 0. The comparator checks and outputs a high load when the port id becomes a one and write strobe goes high. The load register passes the out port to be displayed on the 7-segment displayed.