Increment counter: The counter variable has 6 bits [5:0] where each bit is set to each key. When each key is clicked, a 1’b1 is added to the counter and is stored there where it is checked in other blocks. When the reset button (switch 17) is toggled to low, all counters are reset to 0.

To create a second line, we created a new variable that is 20 pixels lower than the first line and changed both lines to spell out KEY _ PRESSED _ _ TIMES, where _ are a variable that changes as the user clicks on the keyboard, and other conditionals are met.

To determine the priority on which key is displayed when they have been clicked the same number of times, a series of else if statements have been used. To prioritize the higher numbers, they are placed first in the series of else if statements so they are checked first, and the rest can be ignored for that cycle until the next key is pressed. This also functions as the method to flip the lines, as the priority is checked from highest to lowest, so the higher number would be entered first.

To check which number is placed in the first blank of the line, a variable called temp_highest_key is changed in the else if statements which is then pushed through a case block that sets the first_counter_variable to the correct address to be displayed. The same occurs for the second_counter_address by using temp_highest_key_2 variable.

Once the system is booted up, it is asked that key 5 and key 4 are first displayed on screen when no keys are clicked. This is done by sending a value of 6’o65 to first_counter_address and 6’o64 to second_counter_address on reset.

In terms of how many registers are in the experiment, there are about 170 registers in the whole file but only 97 in the main experiment file. In the compilation report, it stated that the system had 176 registers while only 100 were in the main experiment file.