Group 41 3DQ5 Lab 3 Exercise Report

To start this lab, we created a register called *PS2_reg* to hold the first 15 characters of data coming in from the PS2 Keyboard. To make sure that only numerical data was included in the register, we created a series of *else/if* statements which are used to check if the incoming PS2 Code is a number or not. If it is not, then we assign it to the make code for a space. The method used to track the numerical data uses a register, *numkey_presses,* which relates closely with the way a hash table may be used. In this scenario though, instead of an index which is comprised from a hash function, we use the index to tell us what the numerical value of the incoming PS2 Code is. The value at these indices will track the number of occurrences that each value is pressed. For example, let's say the number 4 is pressed 3 times on the PS2 keyboard. If so, *numkey_pressed[4]* is addressing the numerical value "4" from the PS2 Code. The value that is located at the register at the index 4, will be the number of times it is pressed. So in this case *numkey_pressed[4]=3.* This makes the task of keeping track which number is pressed the most very simple, as we keep a counter, *max_presses,* to keep track of this. Whenever the count of a certain number exceeds the current value of *current_count,* we re-assign this counter to the new value, and also store the make code of the most pressed number in *char_temp*. In the case of a tie break, the *if/else* statements are ordered addressing the numerical values in ascending order. By doing this, if there is a tie between the number of characters pressed, than we simply just store the greater value, which will always be the new discovered tie. For the case where more than 9 presses occur for the same number, we handle it by having two separate registers which hold the 1's column, and 0's column of the value at hand in *count_characater1,* and *count_character0* respectively. Then we use a switch case to form the BCD equivalent which what we want displayed if this happens. To display the data in the PPM file we used a series of if statements, with case statements to determine which number should go in the index based on the corresponding number in the register. The way we display the number of times a key is pressed, is by using the previous registers that were used to store all the information. Here we simply display it by using the values and assigning them to character addresses to show up in the PPM file. From our code, it can be seen that we have 191 registers. From the compilation report, it is seen that there are 191 registers reported for the "experiment4.rtl" exercise file with 18 removed due to being grounded, resulting in a total of 173. The grand total is 246 registers including the 27 registers from the PS2_Controller and the 46 registers from the VGA_Controller.