COMPENG 3DQ5: Digital Systems Design Take-Home Exercise 3

10/6/2021

Thursday - Group 55

William Siddeley 400245905

Mohamed Al-Asfar 400262489

This take home exercise was completed in two parts. The first involved displaying the characters for the first 31 keys that were pressed while following the exercise requirements. These included only monitoring letters A - F and displaying a space when an unlisted key was pressed. In order to display the letters, we first check which letter was pressed on the PS/2 keyboard by checking the PS/2 code (e.g. 8'h1C for A). We then set the character to be added to the register as the corresponding octal code for that letter (e.g. 6'o01 for A) as well as increment the decimal and BCD counters for that letter. Additionally, the BCD counter must be manually rolled over in the case that a listed key is pressed more than 9 times. After that, the same logic process is executed for each of the individual listed keys. Then, during the vertical blanking period, when the pixel_X_pos and pixel_Y_pos is on the correct screen location, the letters are printed to the screen by setting the character_address variable to the corresponding register index (ie. character_address = data_reg[30]).

In the second part of the exercise, we utilize the counters and BCD counters that were incremented in part 1 of the exercise. Inside the always_comb block, the counters are compared to each other in order to find the most frequent letter in the data register, and stored in our "max_letter" variable. Then, we initialize two new variables "number_address" and set them equal to the octal code corresponding to the BCD values we previously incremented in part 1. We use two "number_address" variables for the tens and ones place of the number we are printing. We then check if all of the counters are 0, and if they are, we print the "No monitored keys pressed" message, and if not, then we proceed to print the message "Key _ pressed _ times" where the underscores are filled in with the information from our max_letter and number_address variables. The result is the message that tells you which key was pressed most frequently, and how often it was pressed.

In terms of resource usage relating to registers, there were a lot used in this lab. The data register used to hold PS/2 keystrokes was a 2 dimensional array of registers, and there were 31 × 6 registers for a total of 186. We also had registers to hold 2 BCD values of each monitored letter, for another 12 registers. Finally, we had registers for holding decimal values of each letter, so that is another 6 registers. In total, we estimate we used 204 registers in this lab. When comparing our estimate with Quartus' estimate of 279 registers, there is a large discrepancy. When looking at the netviewer and compilation report, we observed that the VGA controller uses 46 registers and the PS/2 controller uses 27. In total, when added to our estimate, we used 277 registers, which is 2 registers short of Quartus' estimate, and those 2 registers can be attributed to delay_X_pos and PS2_code_ready_buf, which is a part of the starter code provided to us at the start of the lab.