

For this exercise, the MIF file that contained the contents of the ROM had to be manipulated so that it would also include the keys for the upper case letters coming from the PS2 keyboard. The address for each of the keys were taken from the PS2 make code while the LCD provided the code to be put in the file at that specific address. The letter "A"s PS2 make-code was 1C, therefore the address was 11C because it was an uppercase and the lowercase letters have the most significant bit of the address as 0. The LCD code that was put inside the address was 41. In the code, add logic needs to be included. Depending on whether the left shift or right shift is pressed, the add logic would either be 100 hex or 000 hex. The difference between the upper case letter and lowercase letter is 100 hex.

For the output line to have 16 characters per line, thus the data register needs to be increased to 16 bits. Following the same method as shown, we initialize all 16 data\_reg as 8'h0. As for data\_counter, it now needs to be a 4 bit counter, counting up to 15.

The seven-segment display was told to be off the whole time unless the first letter after the sixteen characters were typed. The display would then show the number of times that character would show up in the sixteen characters. To create this, there needed to be a new register to be created called seven\_segment\_display to get the frequency of the letter at the top line in the bottom line. It was 4 bits of hexadecimal and had 1 bit added to it every time when another register of 16 bits called pre\_data (contains the data from previous shift registers) equalled the current PS2 code. This check was only initiated when the data counter was at 1 because that had meant the bottom line was full and a new character was pushed to the top line so that it can be compared to the bottom line. The seven\_segment\_display only gets displayed when the data\_counter equals 1 in the seven segment display.

In simulation at 4000 us, the data line changed from line 0 to line 1. Thus in the board\_events file, line 11 "PSQ 4000 11", needs to be changed to "PSQ 4015 11". The 4015 makes sure that 15  $\mu$ s have elapsed and the full line 16 characters gets sent to the LCD controller.