

1. Briefly describe how you converted the hex input to ASCII for your program in Design Task 3.

I used the table to convert the hex input to ASCII, so when every the key is pressed, the program looks for a certain ASCII value in the table and load it in the buffer

2. What addressing mode is used to store blank spaces in the display buffers in the subroutine

`clr_dsp_buffs`? How is the value for a blank space represented in the assembly language code? What is the actual value, in hexadecimal, that is written into the display buffer to represent a blank space?

The ASCII value for blank is 0x20.

3. In Task 3, if you clear the flip-flop as soon as you detect that it has been set, before displaying the character on the LCD, and the pushbutton bounces when pressed, what might happen that would make your program operate incorrectly?

If I clear the flip flop as soon as I detect that it has been set, then the program might take the same value (from the key) two times even though when it is pressed one time.

4. In Task 3, if you clear the flip-flop after displaying the character on the LCD, and the pushbutton is pressed for a long time and bounces when released, what might happen that would make your program operate incorrectly?

The next time you press another key, it might update display with one more extra character which has been loaded into the buffer due to bounce from the previous button pushed.

5. In Task 3, if do not clear the flip-flop after displaying the character on the LCD, what would happen that would make your program operate incorrectly?

If you do not clear the flip flop at all, then the program will not detect any next key pressed value.

6. Since you are not using the PRE input of the 74HC74 in this design, what, if anything, should be done with it.

I assume that I should preset the flipflop after input detect in order to debounce better.