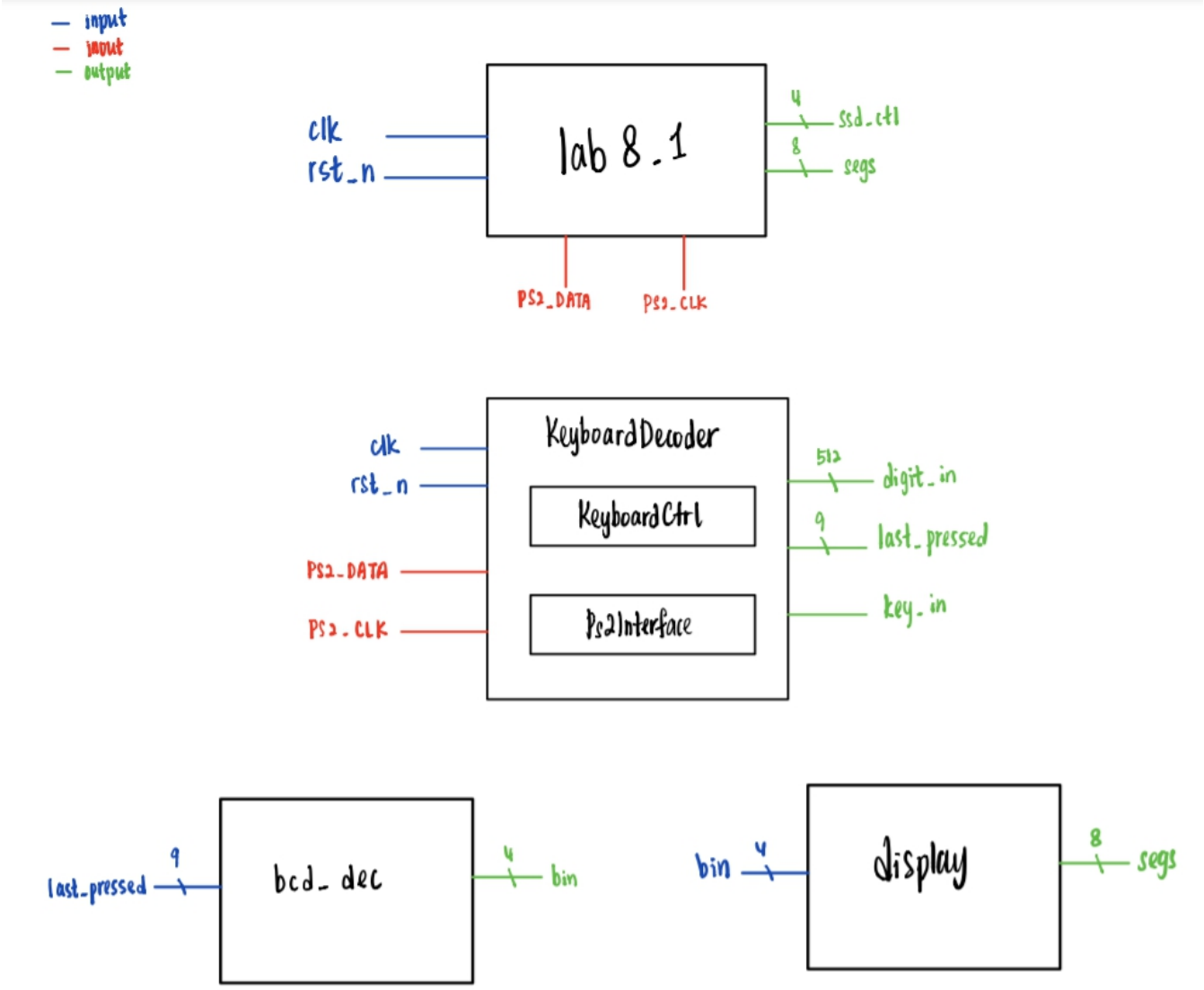
Lab 8 Report

Design Specification

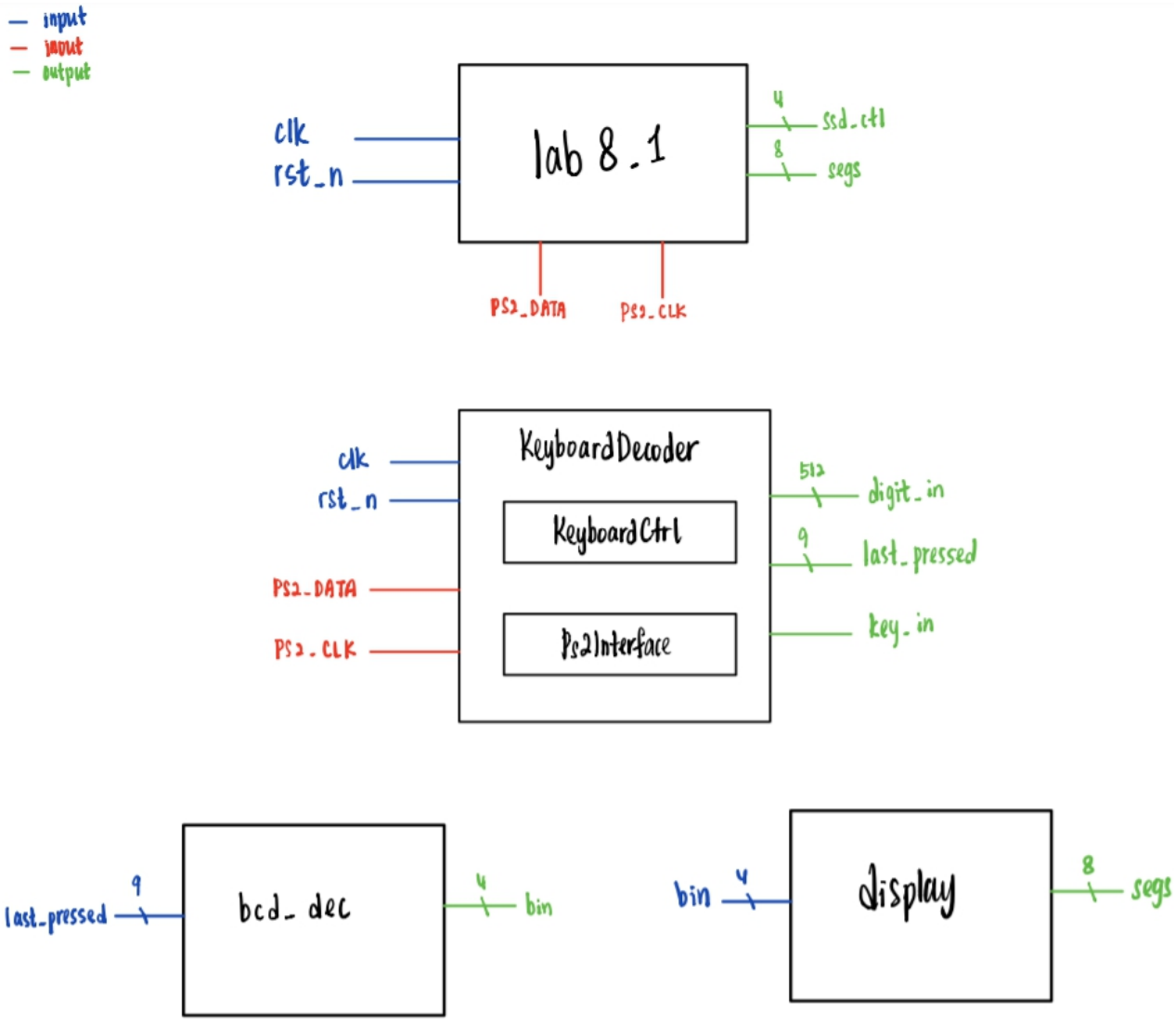
Lab8_1:

input clk;
input rst_n;
inout PS2_CLK, PS2_DATA;
output[7:0] segs;
output [3:0] ssd_ctl;



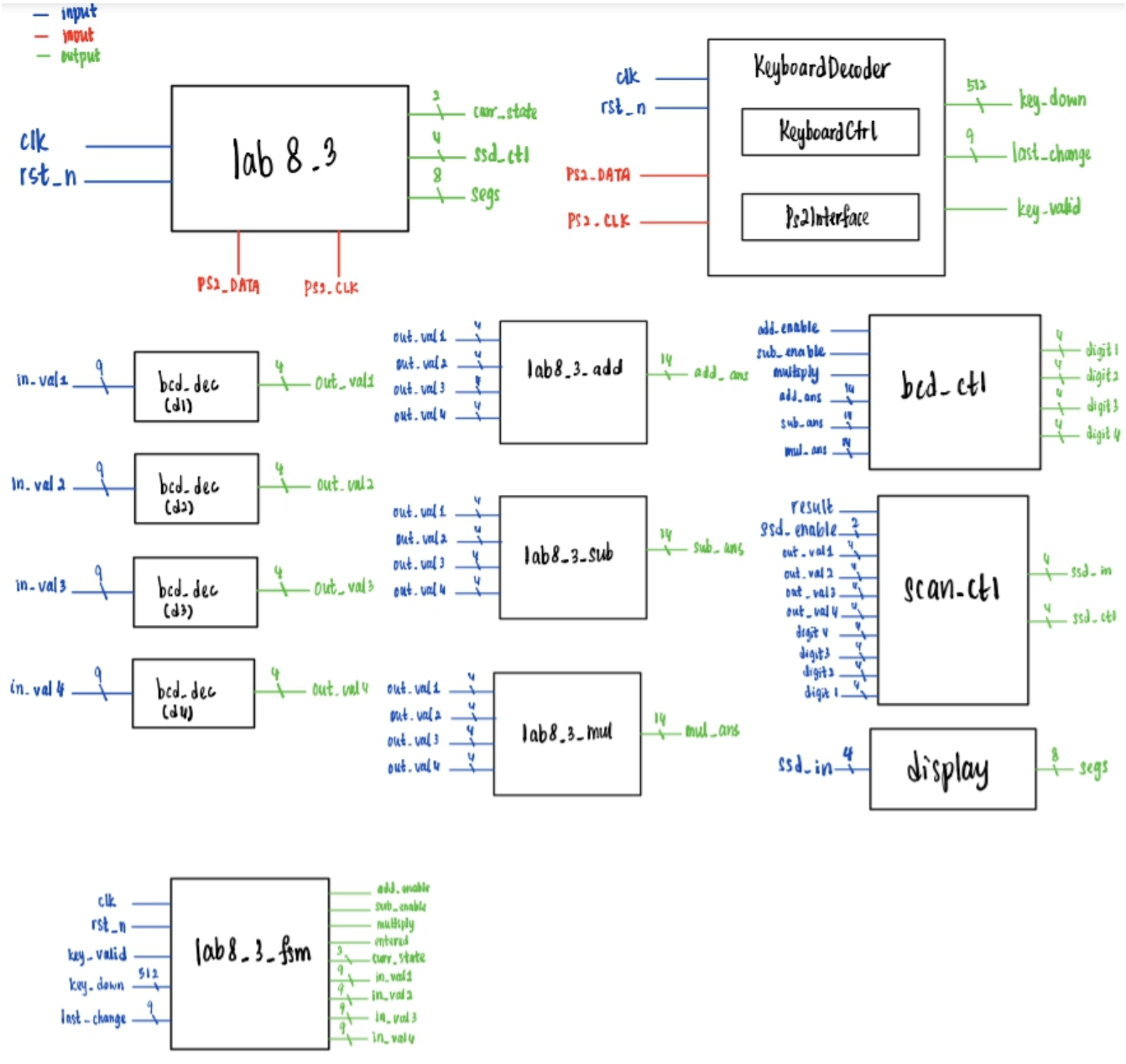
Lab8_2:

input clk;
input rst_n;
inout PS2_DATA, PS2_CLK;
output [3:0] ssd_ctl;
output [7:0] segs;
output [1:0] curr_state;



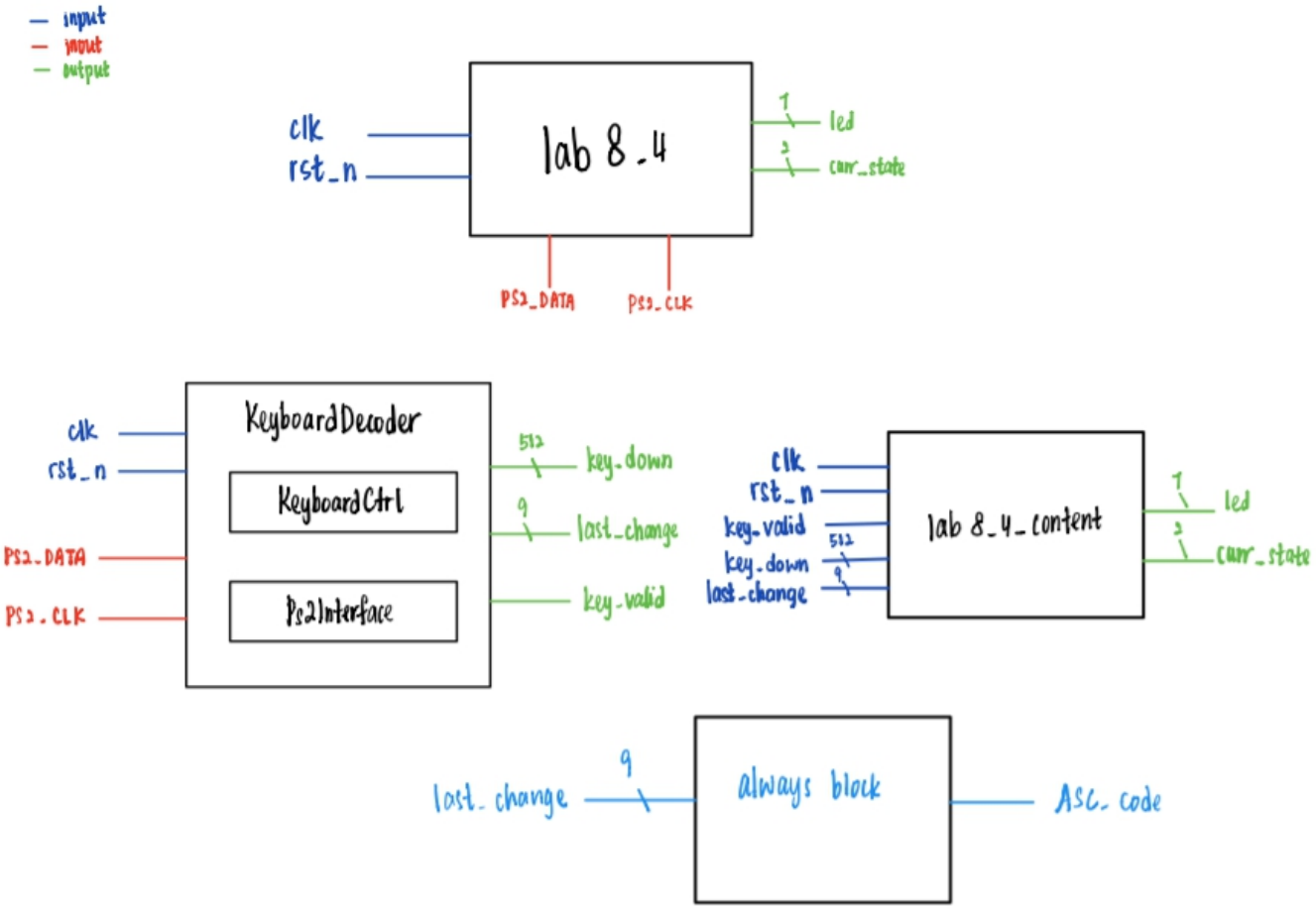
Lab8_3:

input clk;
input rst_n;
inout PS2_CLK, PS2_DATA;
output [2:0] curr_state;
output [3:0] ssd_ctl;
output [7:0] segs;



Lab8_4:

input clk;
input rst_n;
inout PS2_CLK, PS2_DATA;
output [6:0] led;
output curr_state;



Design Implementation

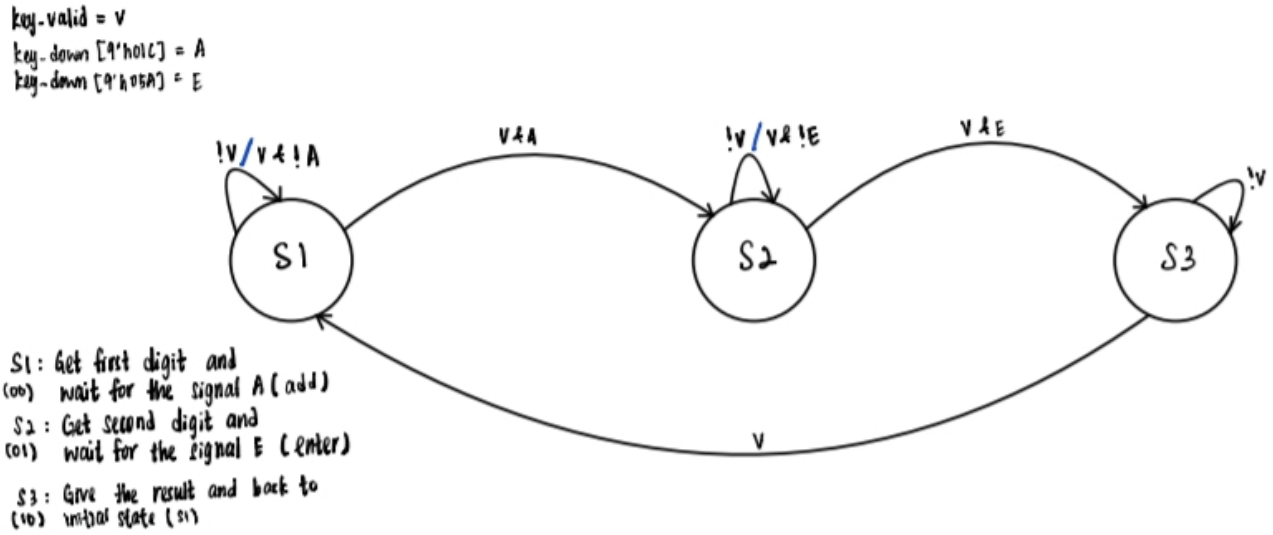
The module KeyboardDecoder used in all the lab was provided by the professor.

Lab8_1:

The module bcd_dec will “translate” the input from the keyboard and send to the module display. The input to the bcd_dec is a 9 bits input, which is the last_pressed from the KeyboardDecoder. The output of it is just a 4 bits bin, which will act as input to the module display.
The module display will display the segments according to the input.
For example, the user type on the keyboard with input of 3, the segments display on the seven segment decoder will be 3.
The 0-9 digit will just display 0-9, on the seven segment decoder.
While the A will display A on the seven segment decoder, since it will do the operation of addition.
The S will display U on the seven segment decoder, since it is doing the subtraction operation. The reason why is it using U is because, if use S it will be similar with 5, when display on the seven segment decoder.
The M will display L. on the seven segment decoder, since it is doing the operation of multiplication. The reason of using L. is because M cannot be display on the seven segment decoder, and the U is already set as the subtraction operation.

Lab8_2:

The lab8_2_add is doing the addition operation for the 2 inputs(2 digits) , and send the output to the scan_ctl and the display for the display on the seven segment decoder.
The current state is initially in the 00 state, and it will change to 01 when the A is being pressed. In this case, A means addition operation. When current state changed to 01, then if the next digit is inserted the output will be the first digit, second digit and result of the addition, and when the enter is pressed, the state will change to 10. At the state of 10, all the output will be turn to zero, and back to the initial state, to wait for the next input.

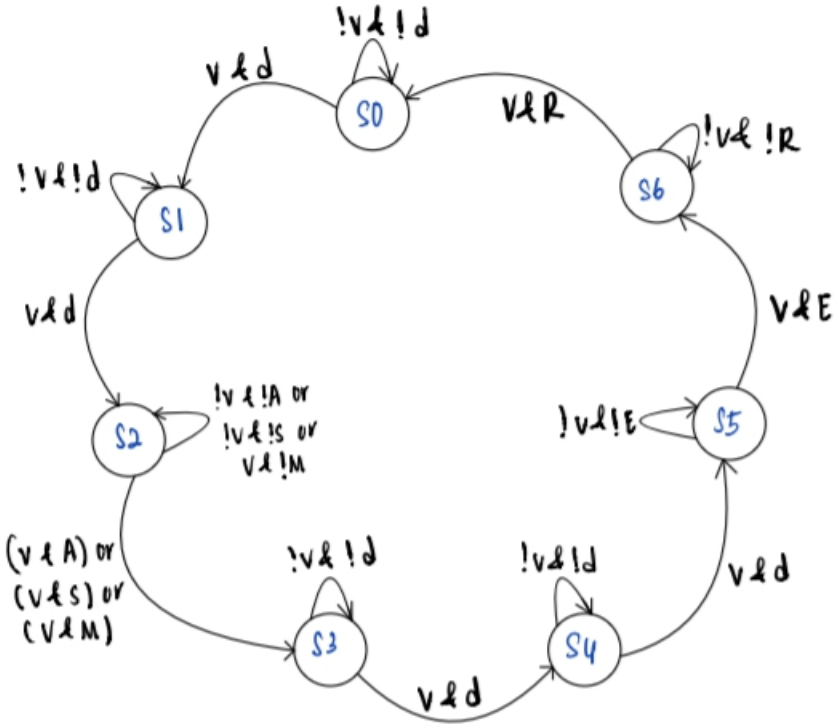


Lab8_3:

The operation of addition, subtraction and multiplication are separated to 3 different module, which are lab8_3_add, lab8_3_sub, lab8_3_mul.
The bcd_ctl will control the output of every operation, when the enable of each operation is triggered. The enable is just the A, S, M input from the keyboard.
The lab8_3_fsm, as shown in the figure below. The current state will change according to the inputs.
The scan_ctl will determine the what to display on the seven segment decoder. It will first decide to display the digits from the inputs, and if the Enter is pressed, it will decide to display the result. The digit to display and the result to display will be send to the display module.

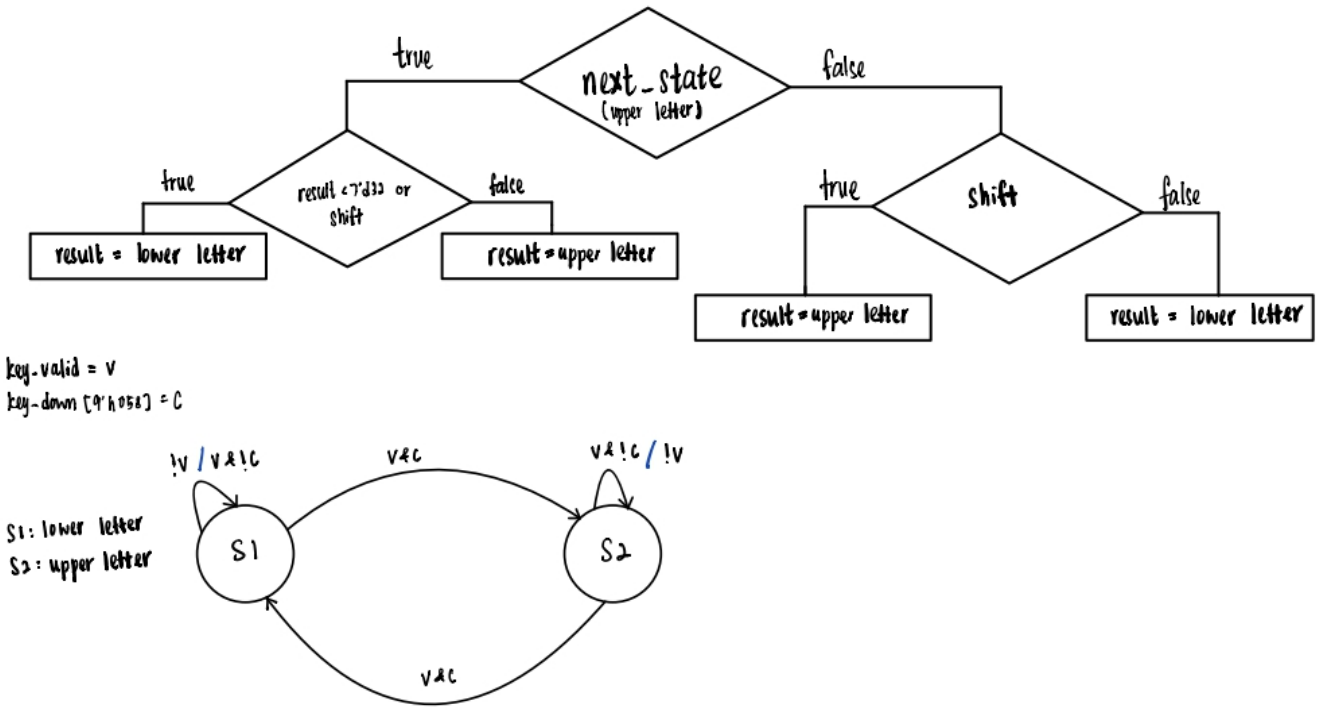
(000) S0: Get first number's first digit
(001) S1: Get first number's second digit
(100) S2: Get which operation to perform
(Addition, Subtraction, Multiplication)
(010) S3: Get second number's first digit
(011) S4: Get second number's second digit
(101) S5: Get signal for result (Enter)
(111) S6: Reset

key_valid = V
key_down [511:0] = d
key_down [9'h01C] = A
key_down [9'h01B] = S
key_down [9'h03A] = M
key_down [9'h05A] = E
key_down [9'h02D] = R



Lab8_4:

The module lab8_4_content will accept the input form the KeyboardDecoder and determine the current state of the input. The initial condition of the current state is 0, which means it is in the state of lower letter. If the input of Caps Lock is pressed or the shift is pressed, current state will be 1, the upper letter's ASCII code will be displayed by the led. If the Caps Lock is pressed and the shift is pressed, the output will be the lower case. If the current state is in 1, and the Caps Lock is pressed, current state will change to 0.
The always block in the lab8_4_content is doing the "translation" for the input of the alphabets, for both upper and lower letters. The result will be subtracted by a decimal of 32, since the upper latter of the alphabets have a difference of 32 with lower letter of alphabets when they are represented in ASCII code.



Discussion

For the question 3, the mistake I made is that I didn't consider the case for subtraction of small number by a big number.

Conclusion

I had learned about how to use the keyboard and display the results according to the different type of input, for example, the digits, the alphabets and the Caps Lock and Shift.