

**Lab9\_1****Design Specification**

- ✓ For a key board display:

Input:

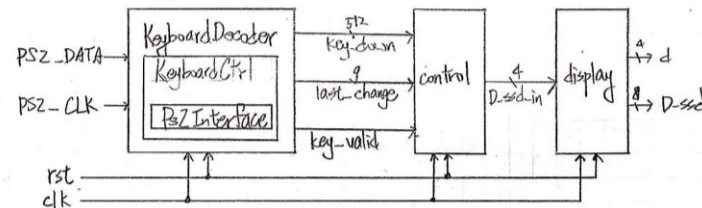
clk, rst\_n

Inout: PS2\_CLK, PS2\_DATA

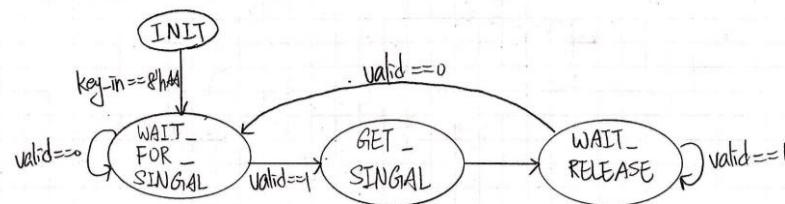
Output:

[3:0]d, [7:0]D\_ssd

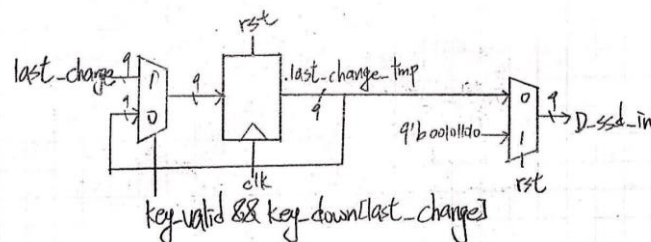
- ✓ Draw the block diagram of the design.

**Design Implementation**

- ✓ 本題由 KeyboardDecoder、control、display 組成
- ✓ KeyboardDecoder 功用為將鍵盤輸入的訊號 decode 成 last\_change



- ✓ Control 控制每當新的鍵盤輸入後，display 將顯示新的輸出。其中須給 key\_down、last\_change、key\_valid 三者一個 AND gate，以確保按下鍵盤輸入。此處輸出為 D\_ssd\_in。

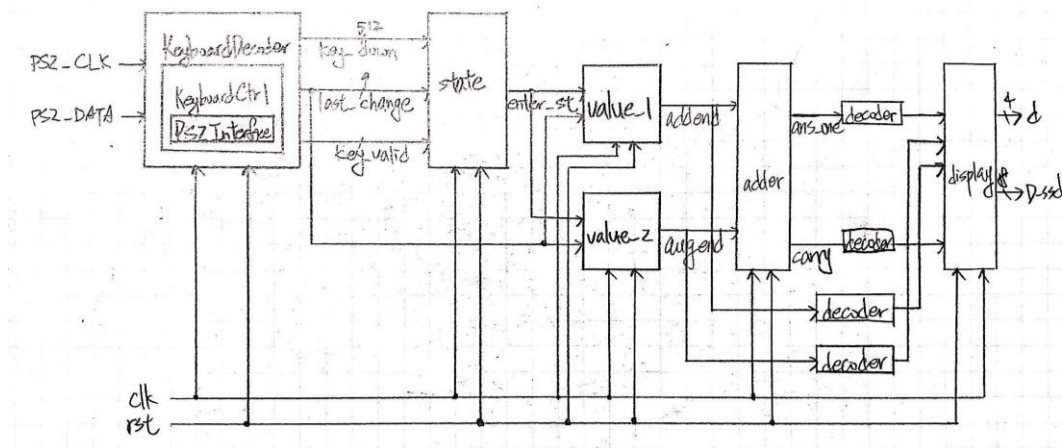


- ✓ Display 為將 D\_ssd\_in 的輸入轉為 8 bits 的 7-segment display，並控制只亮最右邊的七段式顯示器。此處輸出為 [7:0]D\_ssd(控制七段式顯示器數字)與 [3:0]d(控制亮最右邊)。
- ✓ I/O pin

I/O	PS2_CLK	PS2_DATA	clk	rst	d[3]	d[2]	d[1]	d[0]
LOC	C17	B17	W5	V17	W4	V4	U4	U2
I/O	D_ssd[7]	D_ssd[6]	D_ssd[5]	D_ssd[4]	D_ssd[3]	D_ssd[2]	D_ssd[1]	D_ssd[0]
LOC	W7	W6	U8	V8	U5	V5	U7	V7

**Lab9\_2****Design Specification**

- ✓ For a single digit decimal adder:  
Input:  
clk, rst\_n  
Inout: PS2\_CLK, PS2\_DATA  
Output:  
[3:0]d, [7:0]D\_ssd
- ✓ Draw the block diagram of the design.

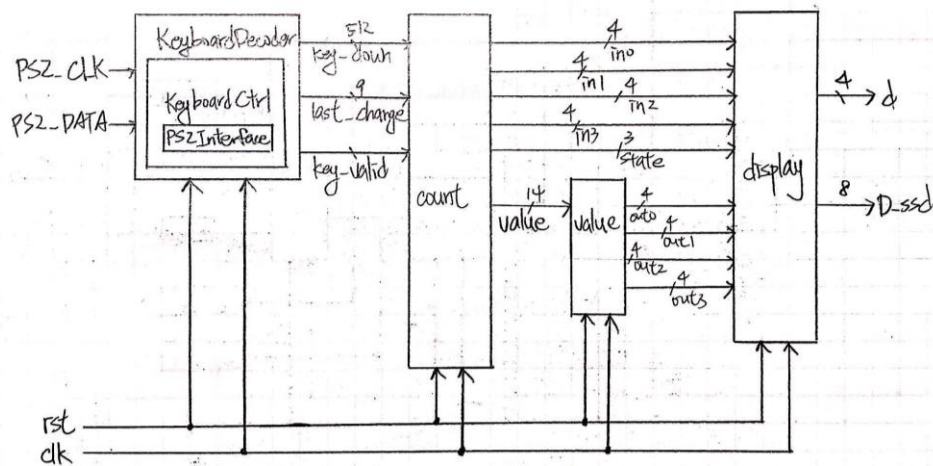
**Design Implementation**

- ✓ 本題由 keyboardDecoder、state、value\_1、value\_2、adder、display 組成
- ✓ State 為控制輸入第一個數與第二個數。當輸入第一個數後按下 enter 鍵，其中的 enter\_st 將變為 ~enter\_st，藉此將第一個數維持輸入的質，並讓下一個輸入成為第二個數的質。簡言之，一開始的 enter\_st == 0，此時鍵盤輸入的質將存進 value\_1 中的 addend；按下 enter 後，enter\_st == 1，此時 addend 的質將不會被改變，鍵盤輸入的質將存進 value\_2 中的 augend。
- ✓ Adder 為加法器，接收 addend 與 augend 做加法，並輸出合的個位數與進位。
- ✓ Decoder 將 9 bits 的 addend、augend、carry、ans\_one 轉為 8 bits 的 D\_ssd 作為輸出
- ✓ display 有 ssdfreq\_div 控制顯示造成視覺暫留，顯示 addend、augend、carry、ans\_one。
- ✓ I/O pin

I/O	PS2_CLK	PS2_DATA	clk	rst	d[3]	d[2]	d[1]	d[0]
LOC	C17	B17	W5	V17	W4	V4	U4	U2
I/O	D_ssd[7]	D_ssd[6]	D_ssd[5]	D_ssd[4]	D_ssd[3]	D_ssd[2]	D_ssd[1]	D_ssd[0]
LOC	W7	W6	U8	V8	U5	V5	U7	V7

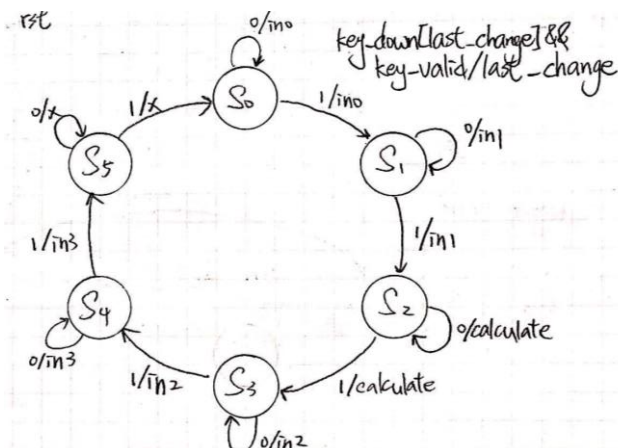
**Lab9\_3****Design Specification**

- ✓ For a two-digit decimal adder/subtractor/multiplier:
- ✓ For a key board display:  
Input:  
clk, rst\_n  
Inout: PS2\_CLK, PS2\_DATA  
Output:  
[3:0]d, [7:0]D\_ssd
- ✓ Draw the block diagram of the design.



### Design Implementation

- ✓ 本題由 keyboardDecoder、count、value、display 組成
- ✓ Count 的功能為將 last\_change decode 並做計算。其中有 6 個 state。第一與第二個 state 分別讀取被加數、被減數、被乘數的十位數與個位數；第三個 state 讀取要做加法、減法或乘法；第四與第五個 state 分別讀取加數、減數、乘數的十位數與個位數；第六個 state 為按下 enter 鍵後計算並輸出 value。



- ✓ Value 功能為將 count 輸出的四位數 value 的每一位轉成一個 4 bits 的 output。其作法為將 value 除以 10 取餘數得個位數(out3)；value 除以 100 取餘數減 value 除以 10 取餘數得十位數(out2)；value 除以 1000 取餘數減 value 除以 100 取餘數得百位數(out1)；value 減 value 除以 1000 取餘數得千位數(out0)。
- ✓ Display 顯示輸入的被(加減乘)數與(加減乘)數與答案。其中藉由 state 選擇顯示的數字，state == 0 時顯示 in0~in3；state == 1 時顯示 out0~out3。
- ✓ I/O pin

I/O	PS2_CLK	PS2_DATA	clk	rst	d[3]	d[2]	d[1]	d[0]
LOC	C17	B17	W5	V17	W4	V4	U4	U2
I/O	D_ssd[7]	D_ssd[6]	D_ssd[5]	D_ssd[4]	D_ssd[3]	D_ssd[2]	D_ssd[1]	D_ssd[0]
LOC	W7	W6	U8	V8	U5	V5	U7	V7

### Lab9\_4

#### Design Specification

- ✓ For a ASCII code display:
- ✓ For a key board display:

Input:

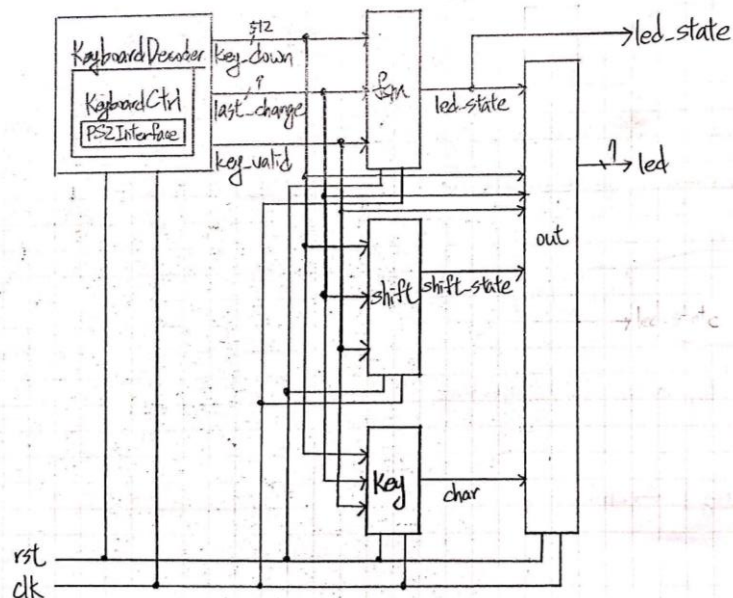
clk, rst\_n

Inout: PS2\_CLK, PS2\_DATA

Output:

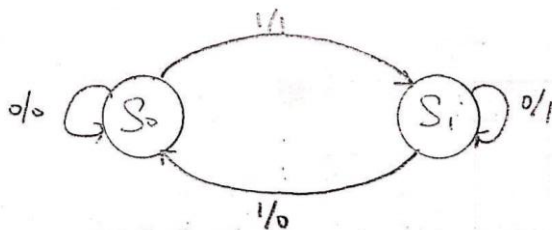
Led\_state, [6:0]led

- ✓ Draw the block diagram of the design.



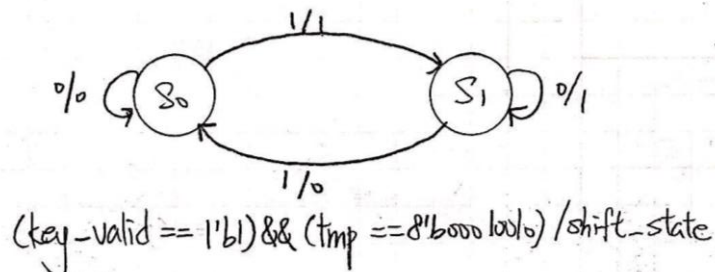
### Design Implementation

- ✓ 本題由 keyboardDecoder、fsm、shift、key、out 組成
- ✓ Fsm 為判斷 Caps 鍵的輸入，若其輸入為 1 時，切換為大寫模式，led\_state 輸出為 1。

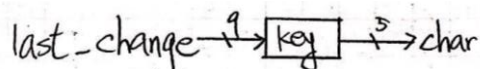


$\text{key\_down}[\text{last\_change}] == (1'b1) \ \&\& \ (\text{key\_valid} == 1'b1) \ \&\& \ (\text{tmp} == 8'b01011000) / \text{led\_state}$

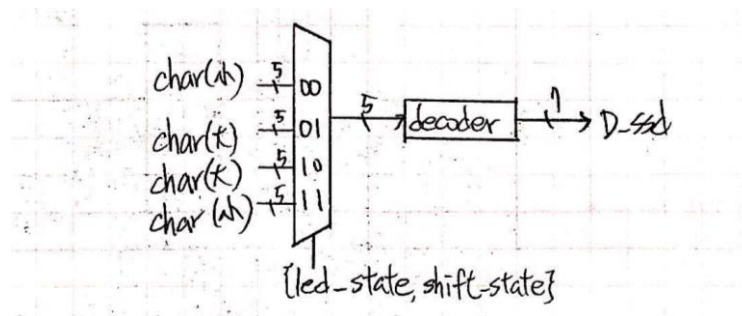
- ✓ Shift 為判斷 shift 鍵得輸入，在此需要按著 shift 鍵時，輸出才會是大寫。



- ✓ Key 為一個 decoder，將輸入的 last\_change 轉為 5 bits 的 char 輸出。



- ✓ Out 根據 led\_state 與 shift\_state 判斷輸出的 7 bits binary 訊號，為字母以 ASCII 的二進位表示法。圖中 char(大)與 char(小)實際上輸入皆為 char，在此只是為了表示其輸出為大寫或小寫。



✓ I/O pin

I/O	PS2_CLK	PS2_DATA	clk	rst	led_state	led[6]
LOC	C17	B17	W5	V17	L1	U14
I/O	led[5]	led[4]	led[3]	led[2]	led[1]	led[0]
LOC	U15	W18	V19	U19	E19	U16

### Discussion

在第一到第三題中，第一題與第二題其實可以直接將 last\_change 轉換成 8 bits 的 D\_ssd 輸出，但我一開始是依照之前的實驗方式去想，因此將數字從 9 bits 的 last\_change 先轉成 4 bits，再經過 BCD to 7-segment decoder 才輸出。到第三題才想到這個過程很多餘，因為其皆為 decoder。因此在第三題時便直接將 decoder 合在 count 裡。

在第三題中，我的設計會有個小問題是只要按下鍵盤任一鍵，state 皆會跳下一個 state。因此應該要像第二題一樣，在按 enter 後才可以輸入下一個數字，以免無法確定當下的 state 為何。或藉由 led 的幫助顯示當下的 state，將有助於改善此問題。

第四題主要要注意大小寫的切換，也就是 led\_state 與 shift\_state 的條件。

### Conclusion

這次利用鍵盤作為輸入，主要重點是將鍵盤輸入的訊號 decode 以及 state 的切換。雖然可以將 decoder 全部做在一起，但有可能會自己搞混出錯。因此適時的加入新的 module 也很重要。