**Lab9\_1**

**Design Specification**

* For a key board display:

Input:

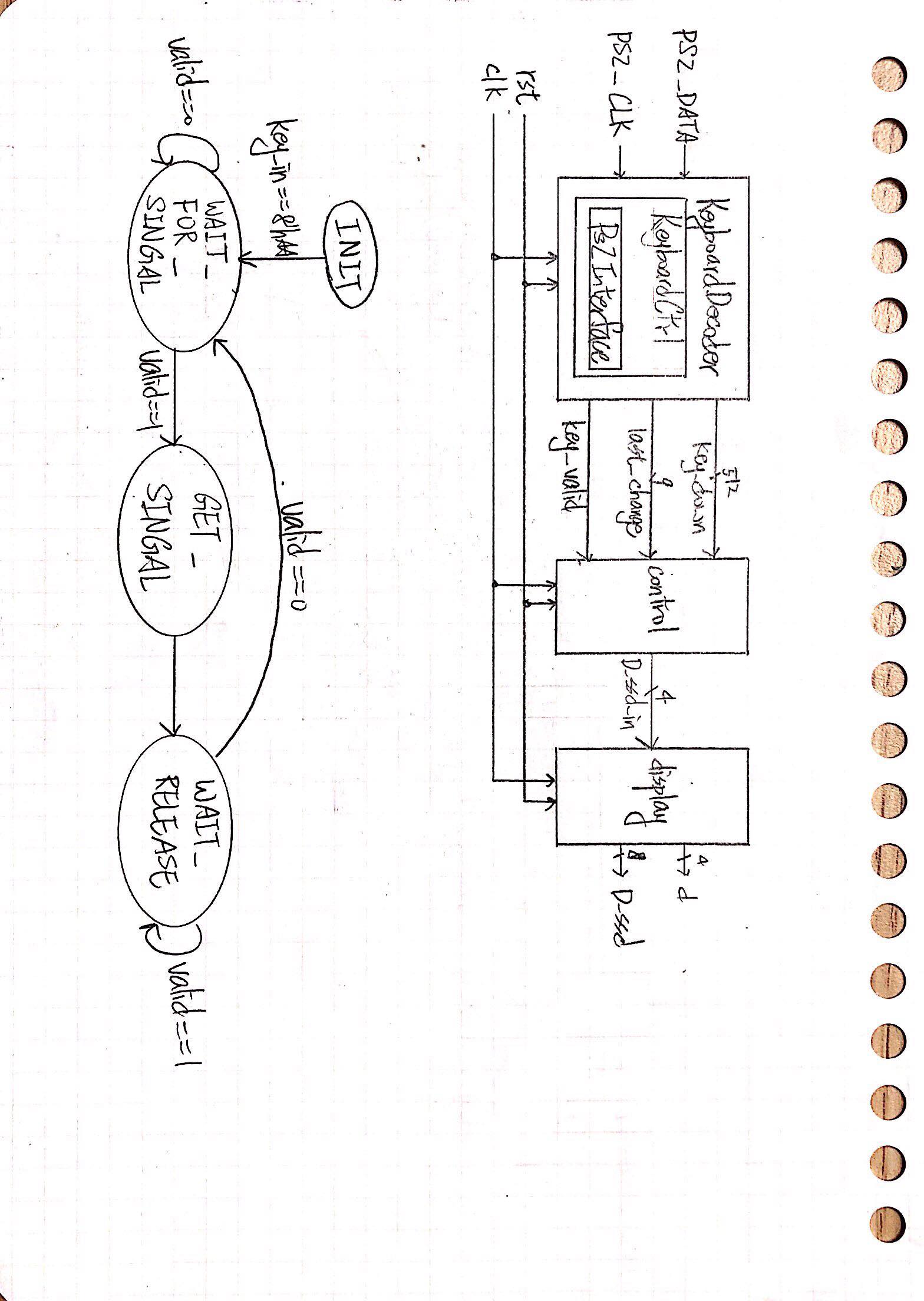
clk, rst\_n

Inout: PS2\_CLK, PS2\_DATA

Output:

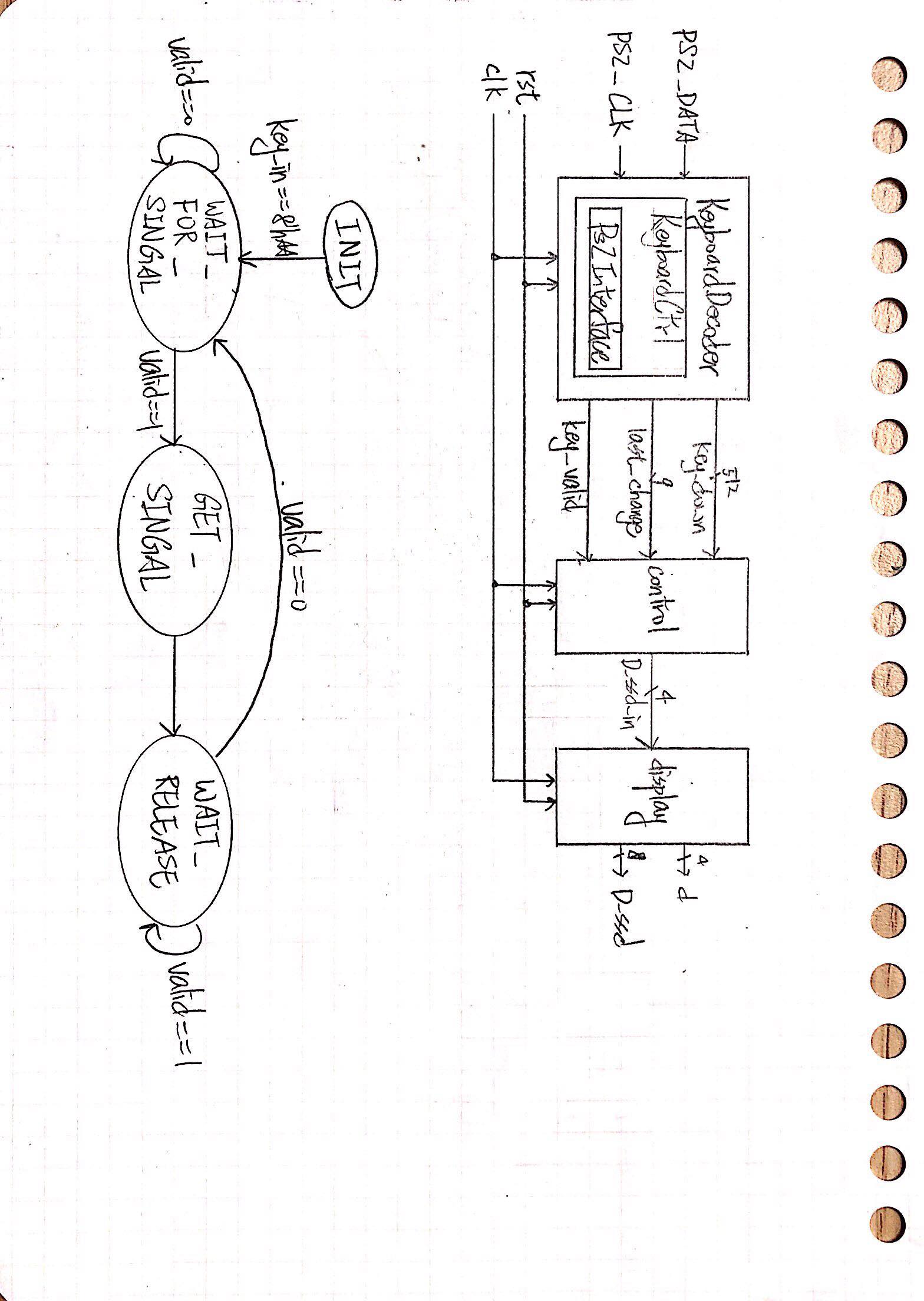
[3:0]d, [7:0]D\_ssd

* Draw the block diagram of the design.

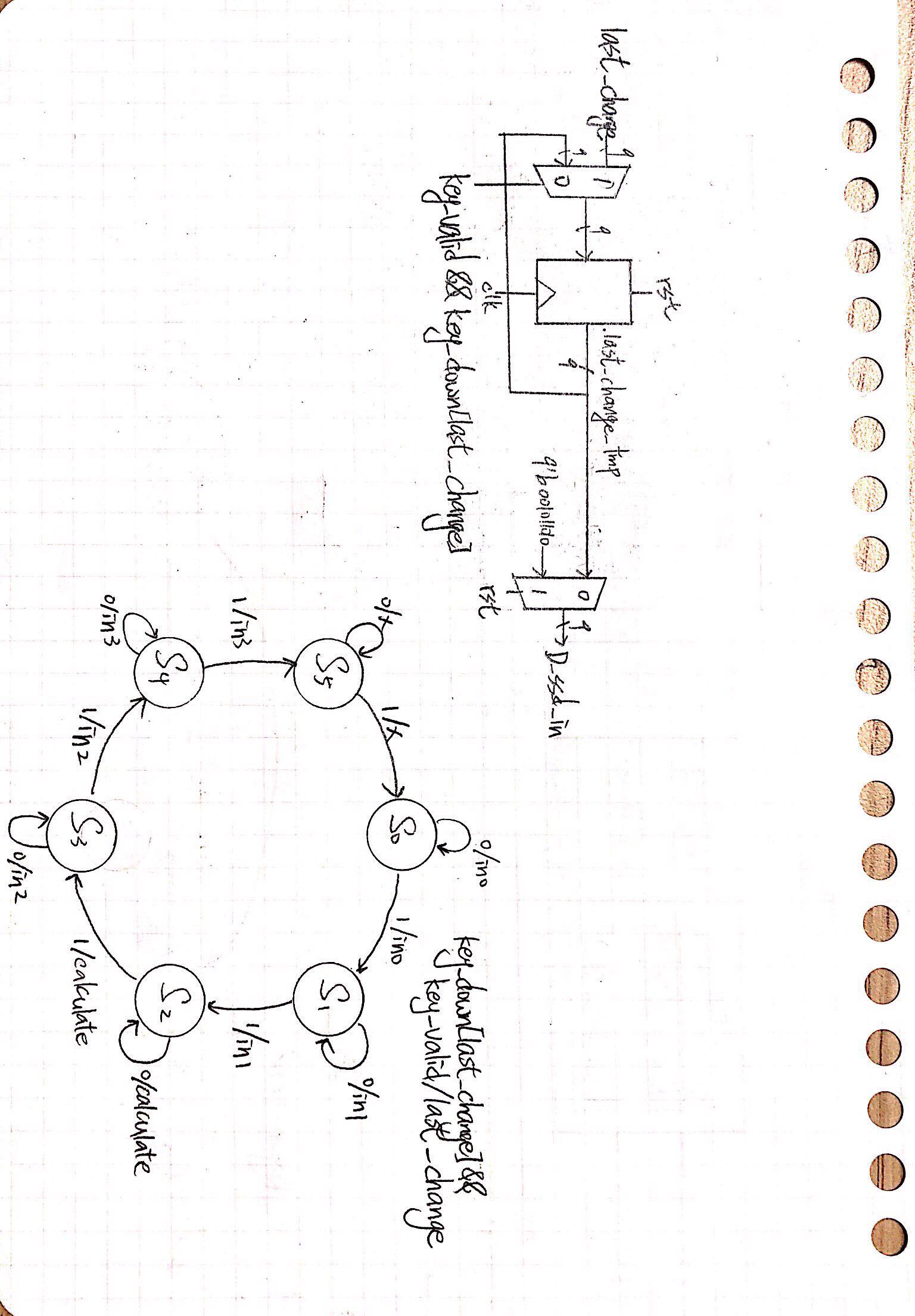
****

**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:

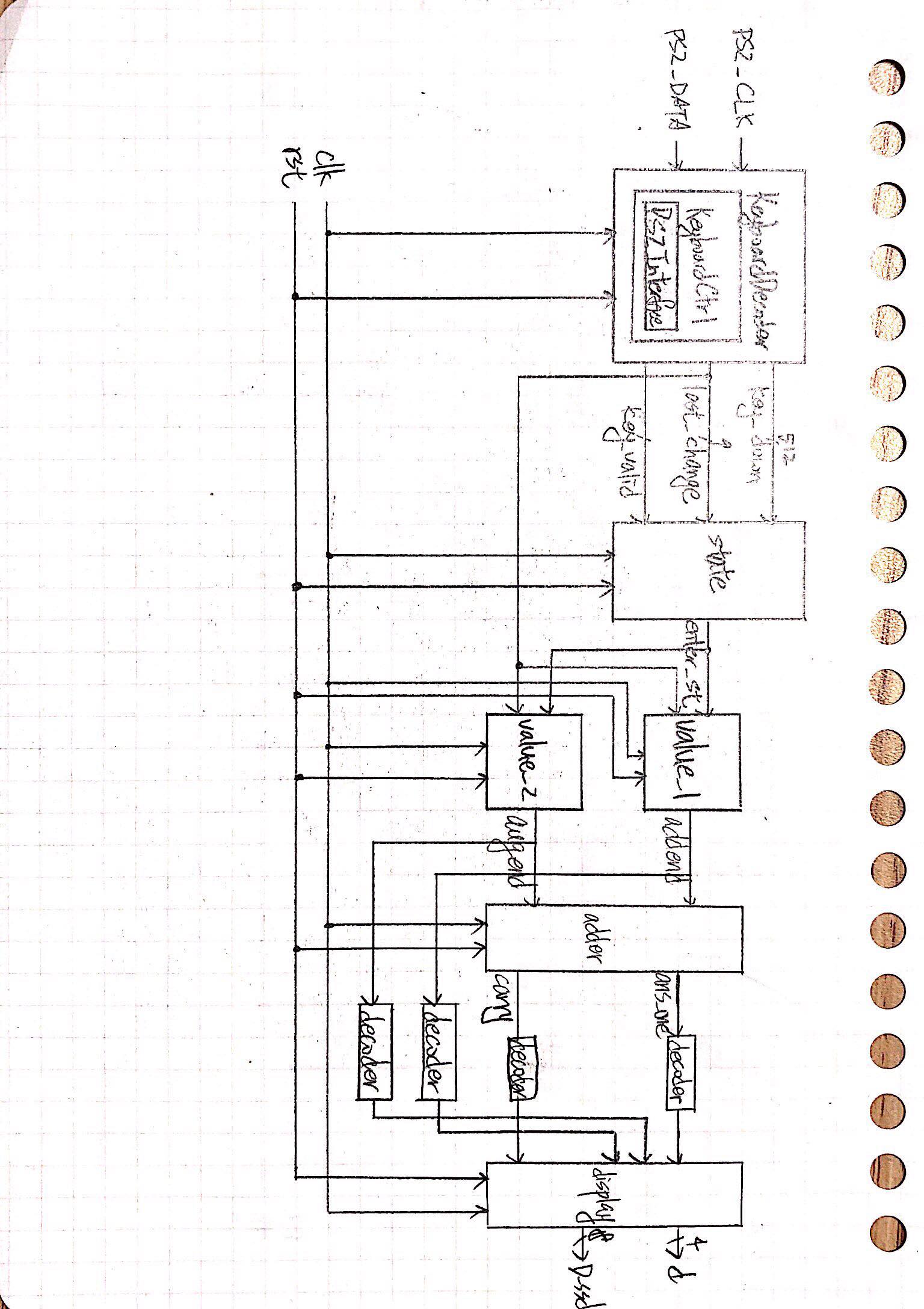
clk, rst\_n

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augdend。
* 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:

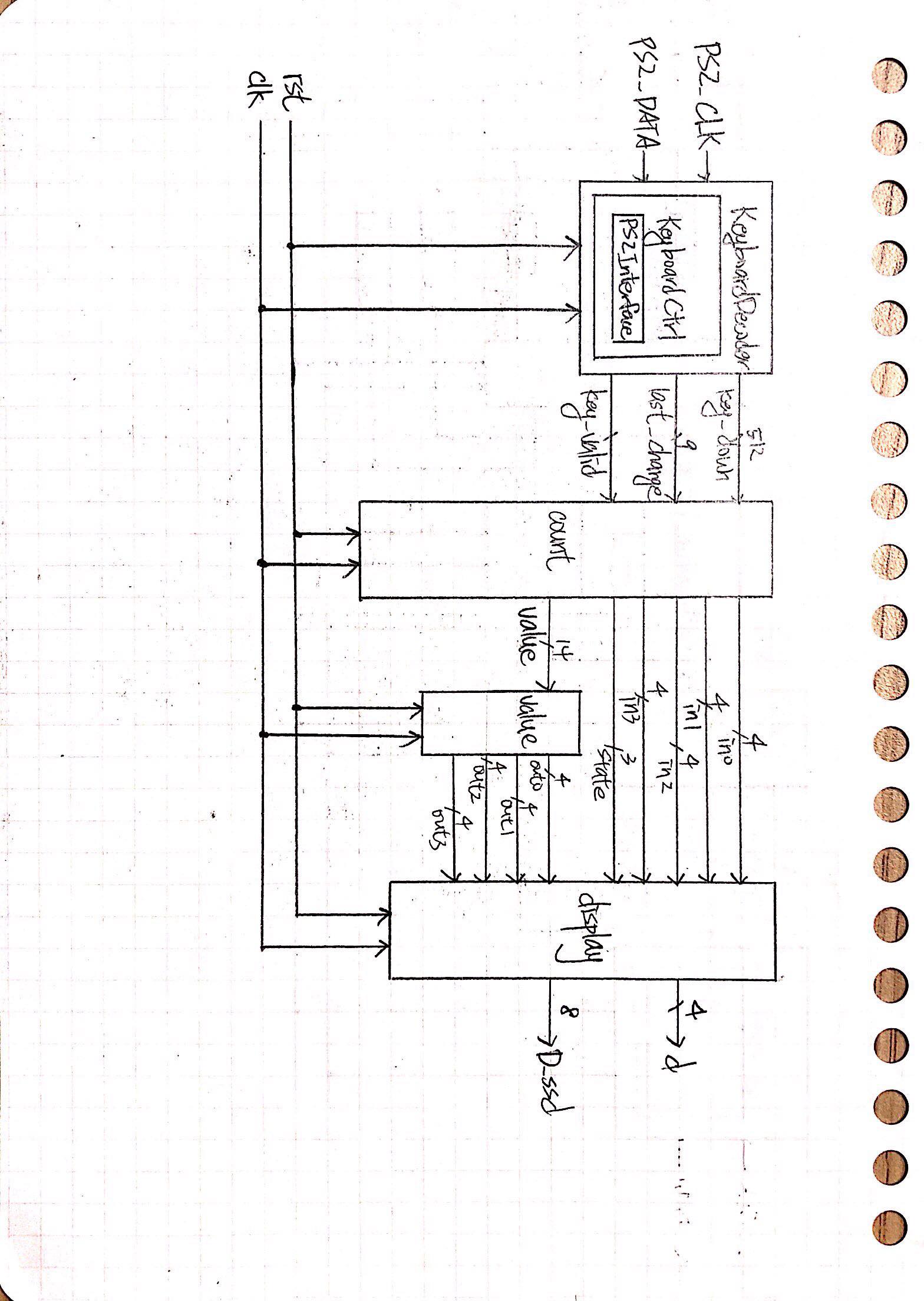
clk, rst\_n

Inout: PS2\_CLK, PS2\_DATA

Output:

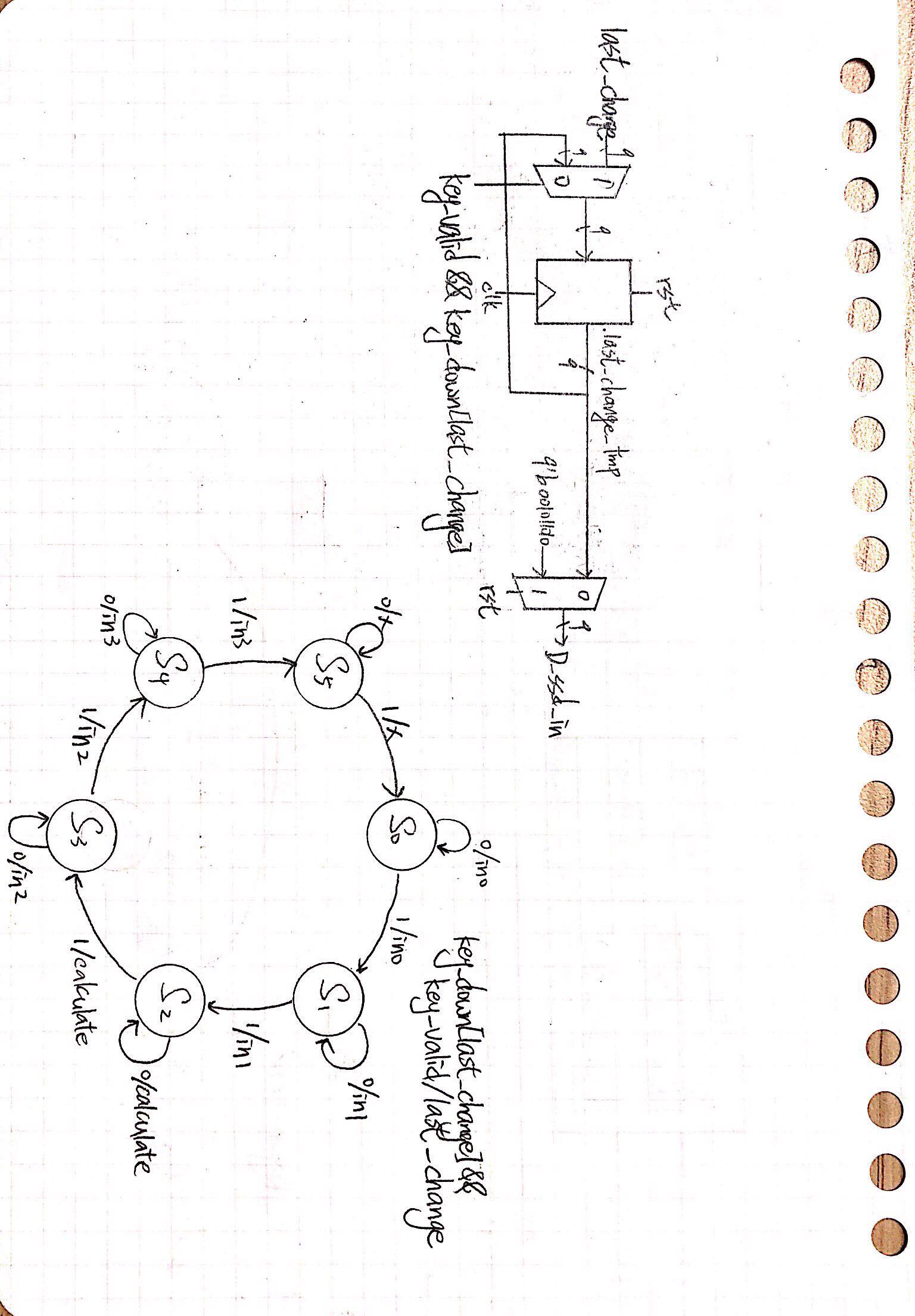
[3:0]d, [7:0]D\_ssd

* 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 diplay:
* For a key board display:

Input:

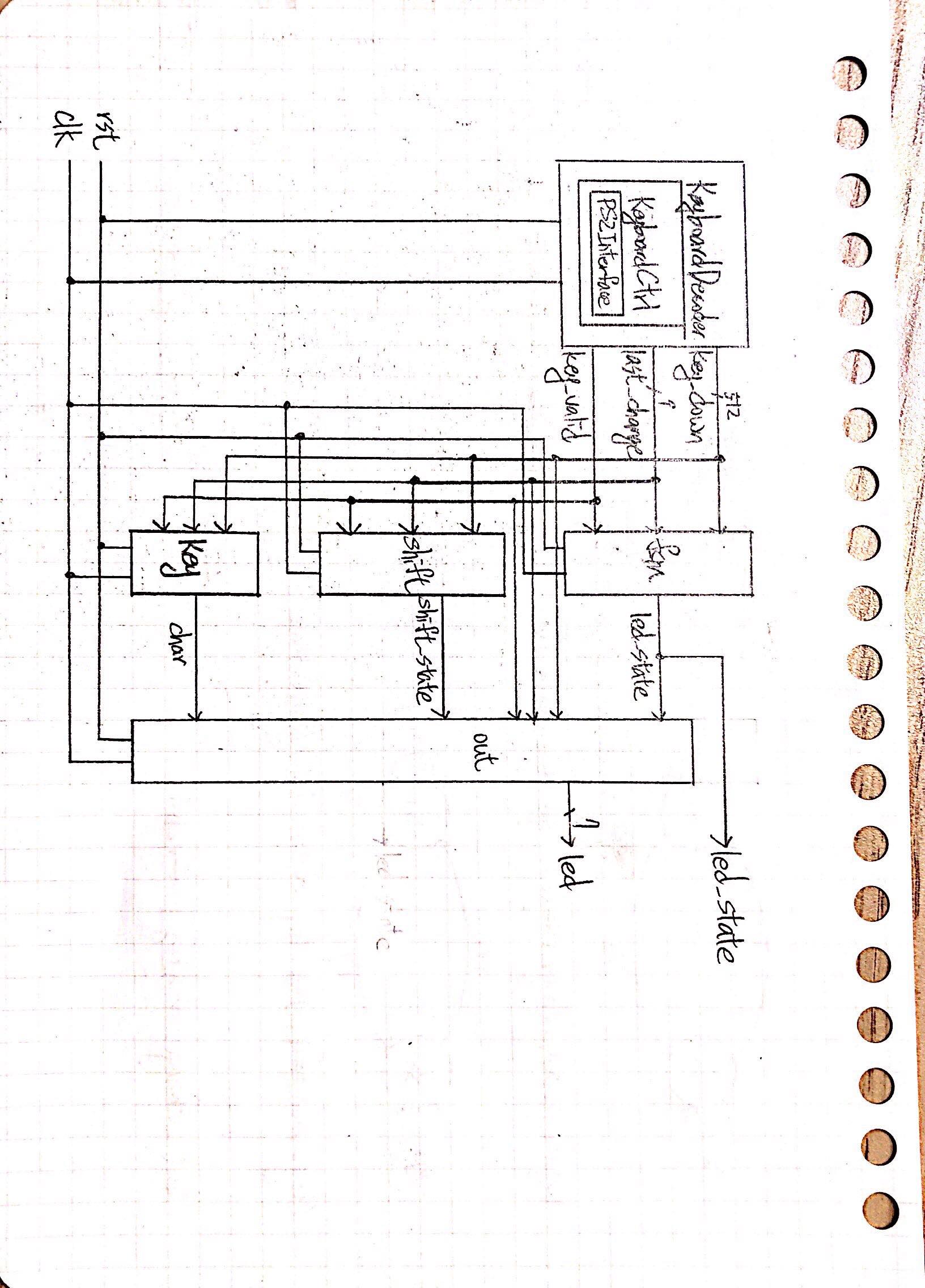
clk, rst\_n

Inout: PS2\_CLK, PS2\_DATA

Output:

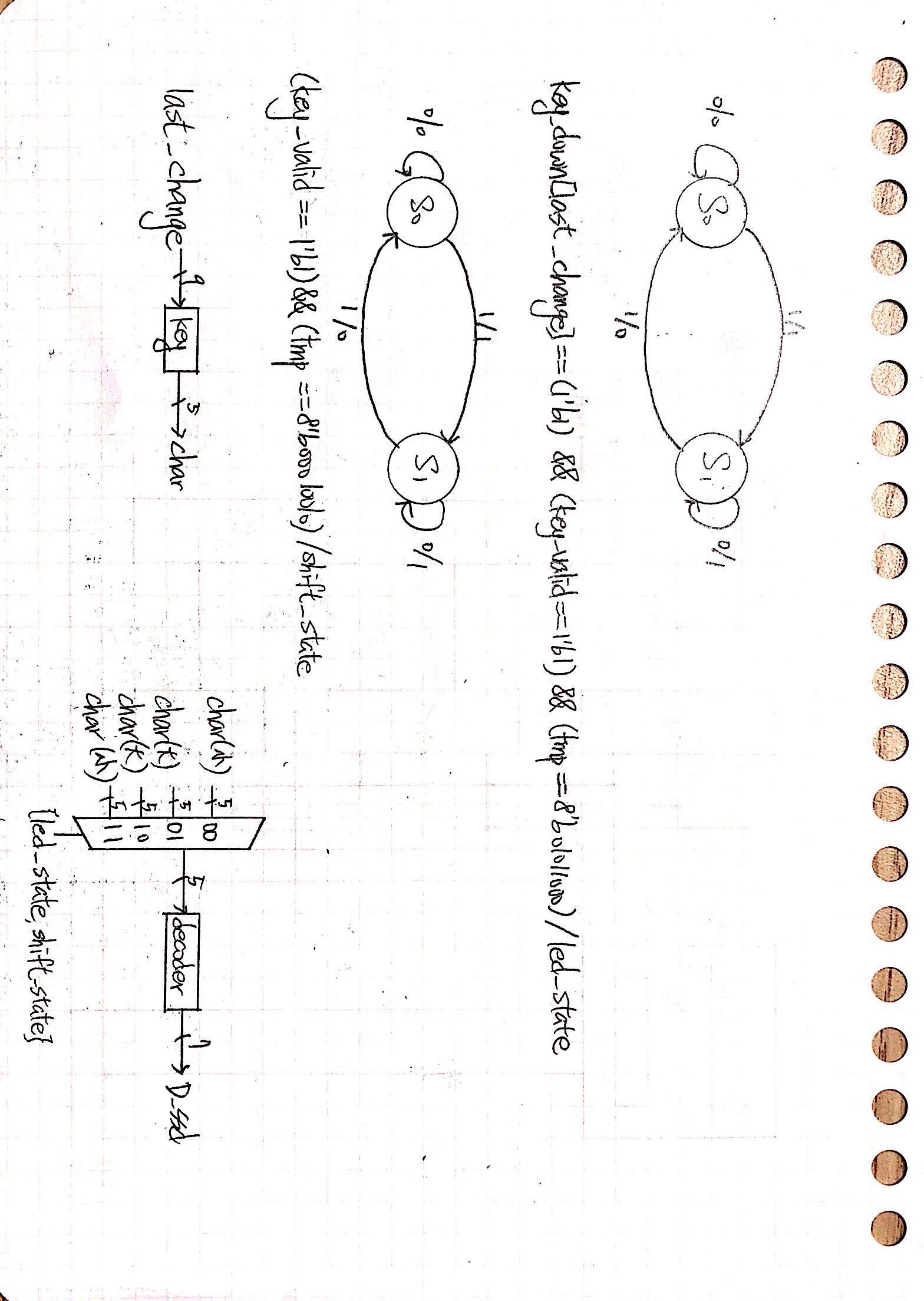
Led\_state, [6:0]led

* Draw the block diagram of the design.

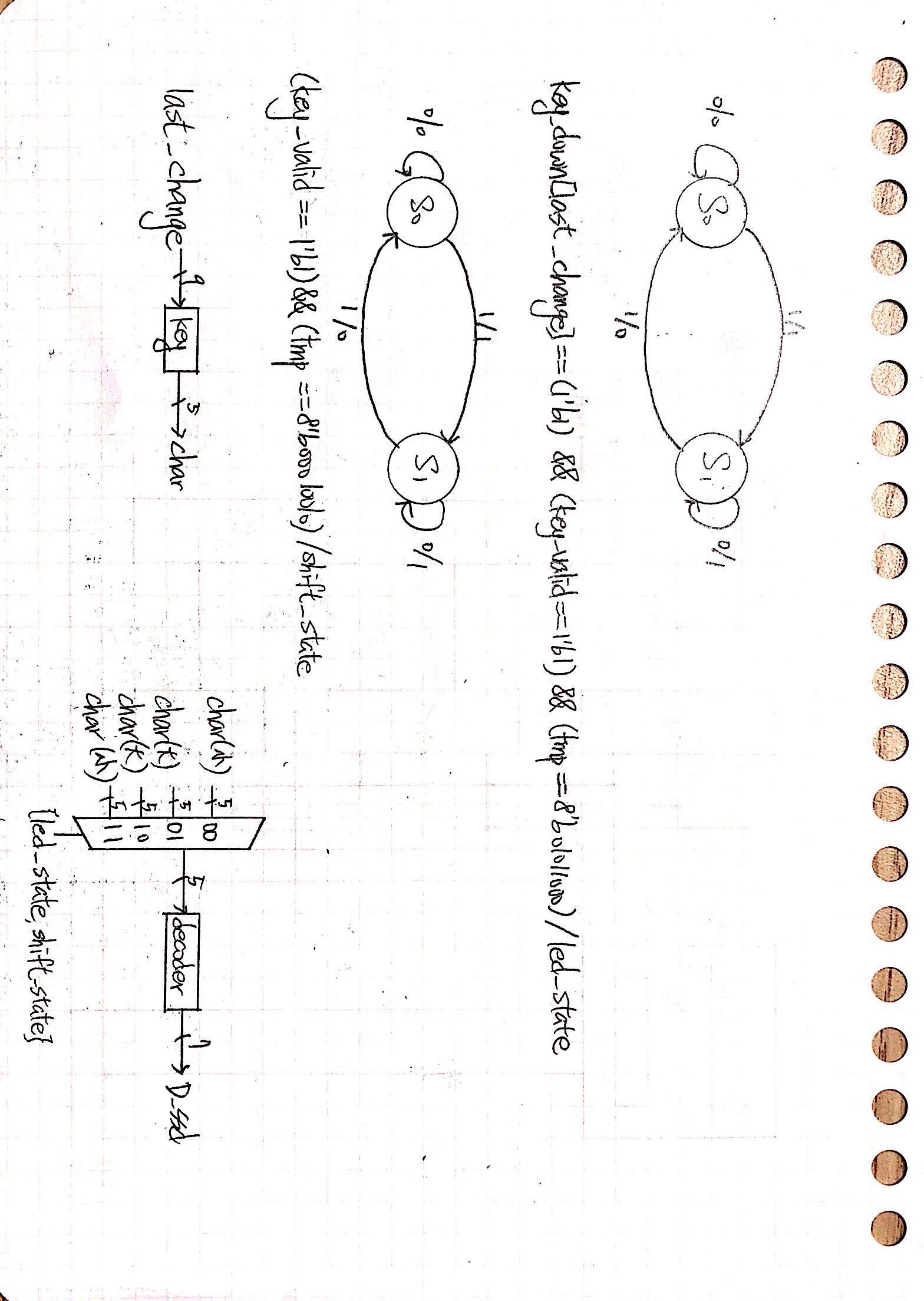


**Design Implementation**

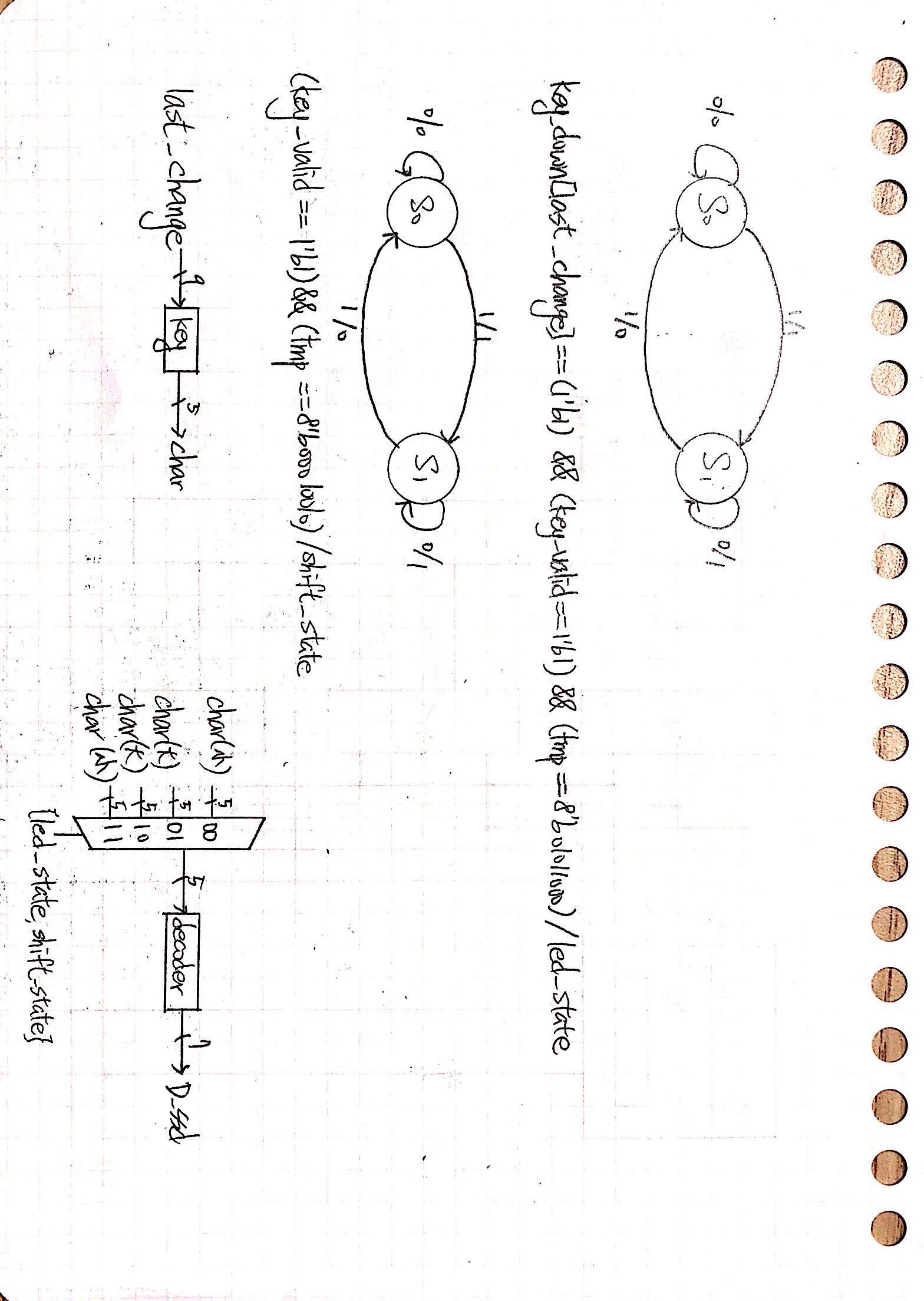
* 本題由keyboardDecoder、fsm、shift、key、out組成
* Fsm為判斷Caps鍵的輸入，若其輸入為1時，切換為大寫模式，led\_state輸出為1。



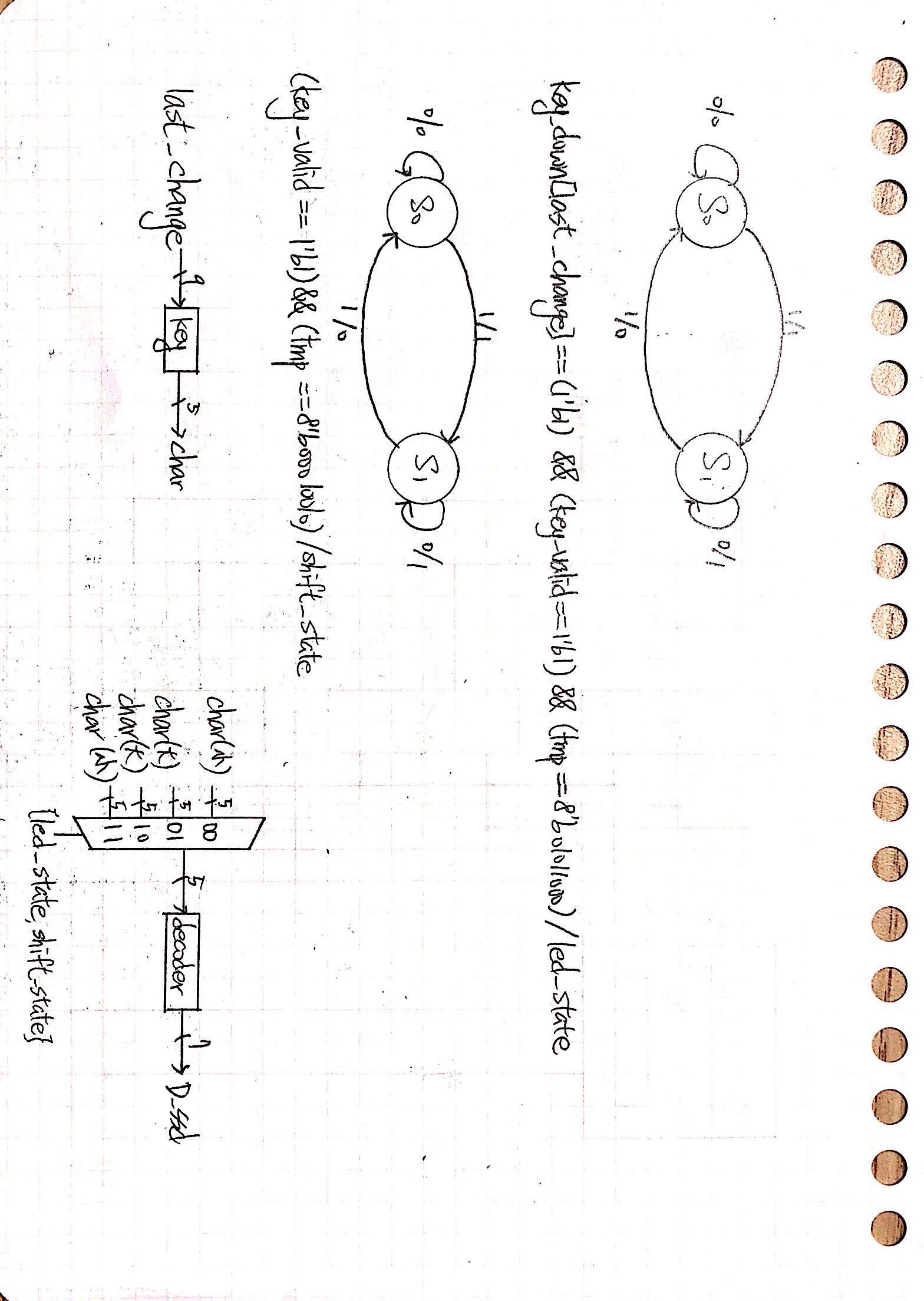
* Shift為判斷shift鍵得輸入，在此需要按著shift鍵時，輸出才會是大寫。



* 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。