**Lab7\_1**

**Design Specification**

* For a stopwatch:

Input:

start\_in // 開始倒數

rst\_n // control rst\_n button

clk

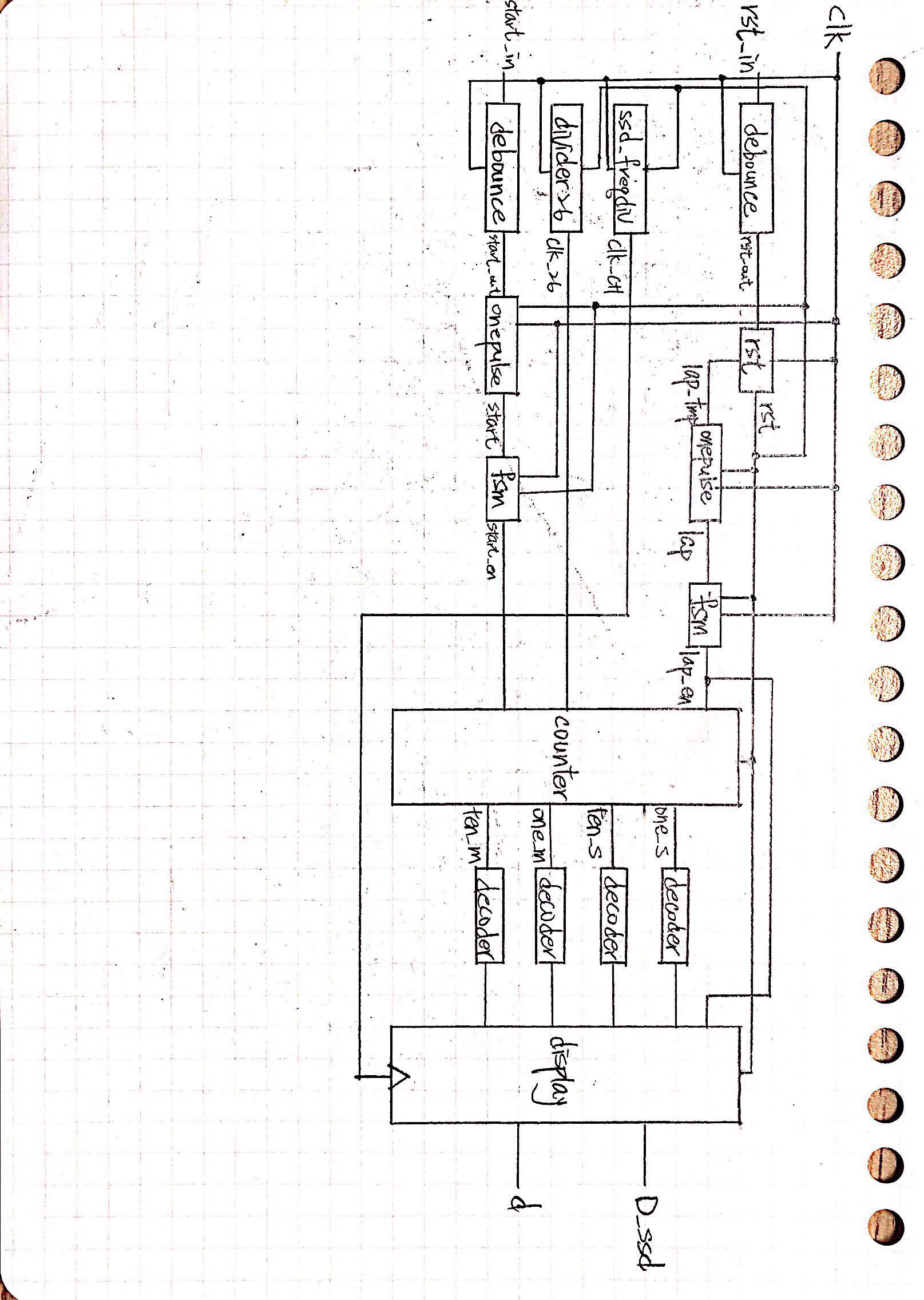
clk\_c // 控制秒的頻率(非必要，因此無在block diagram中顯示)

Output:

D\_ssd[7:0] // 7-segment display

d[3:0]

* Draw the block diagram of the design.

****

**Design Implementation**

* 本題由ssd\_freqdiv、divider26、debounce、rst、onepulse、fsm、decoder、counter、display九個module組成。由於這次是用按鈕控制，因此有debounce、rst、onepulse三個module。Debounce的功能是產生穩定的波型；rst的功能是區分長按與短按；onepulse的功能是製造一個button的訊號，讓按下 button後可以一直保持在下個state。以上皆在之前的lab有說明過，因此不再贅述。ssd\_freqdiv、divider26則是做出需要的頻率；decoder是將binary轉為BCD以呈現再7-segment display上；counter的功能為計算秒與分，主要組成為BCD upcounter；fsm則是控制長按短按的狀態，以上在之前的lab皆有做過，因此此次重點將放在display中的lab。
* Display

這次實驗重點為碼錶分圈。Input 的lab和rst\_n為同一個按鈕控制，其中lab為短按。和之前的display不同之處為lab7的display中多加了一個mux，控制輸出為分圈的數字還是正在數的數字。若lab = 1，則輸出分圈數字；lab = 0，則輸出正在數的數字。每當lab再次為1時，分圈的數字皆會改變。

* I/O pin

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 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] |
| VOC | W7 | W6 | U8 | V8 | U5 | V5 | U7 | V7 |
| I/O | d[3] | d[2] | d[1] | d[0] | clk | rst\_n | start\_in |  |
| VOC | W4 | V4 | U4 | U2 | W5 | W19 | T17 |  |

**Lab7\_2**

**Design Specification**

* For a downcounter:

Input: clk,

rst\_n,

start, // 開始倒數

pause, // 暫停倒數

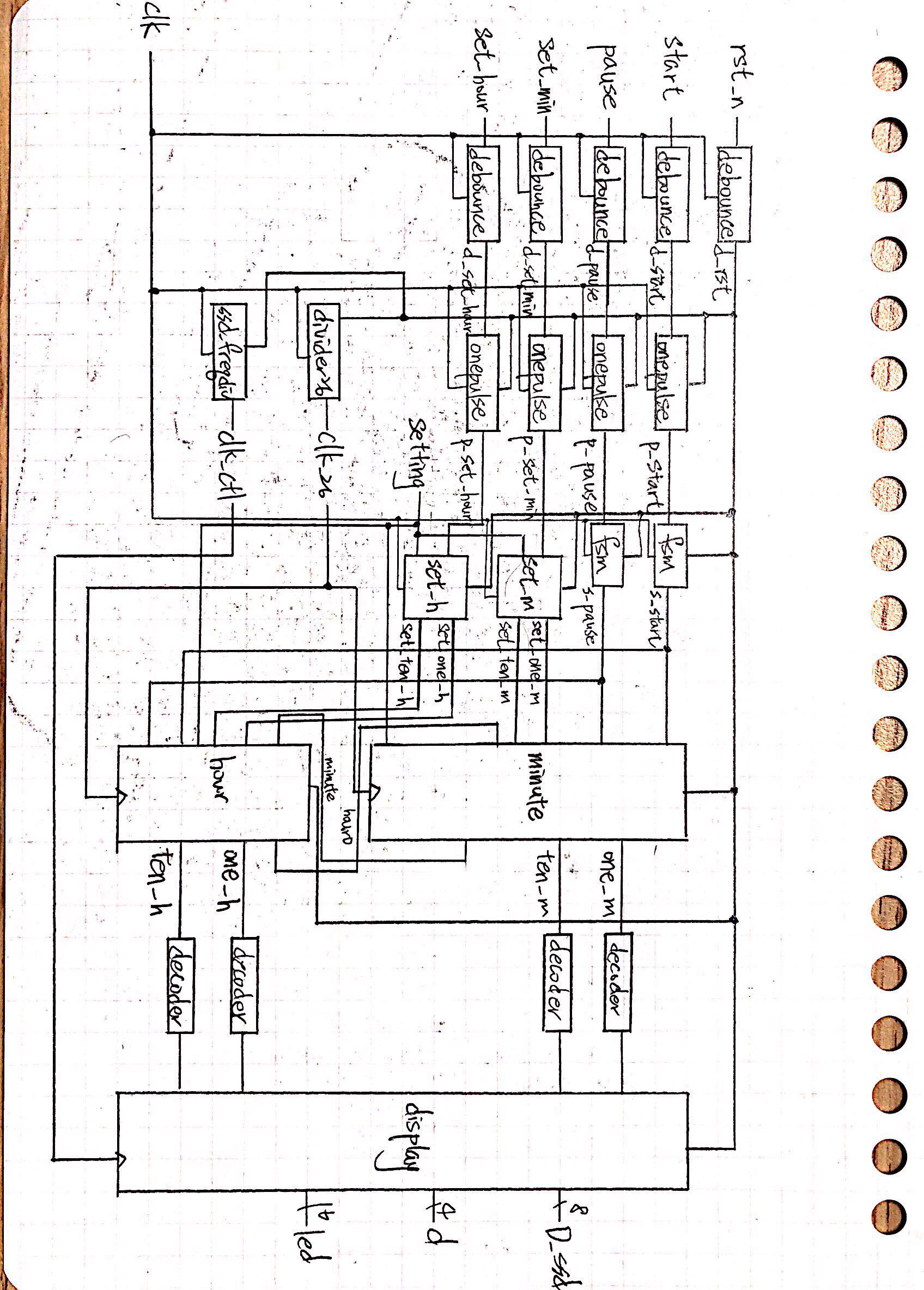
setting, // 允許設置時間

set\_min, // 設置分

set\_hour // 設置時

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

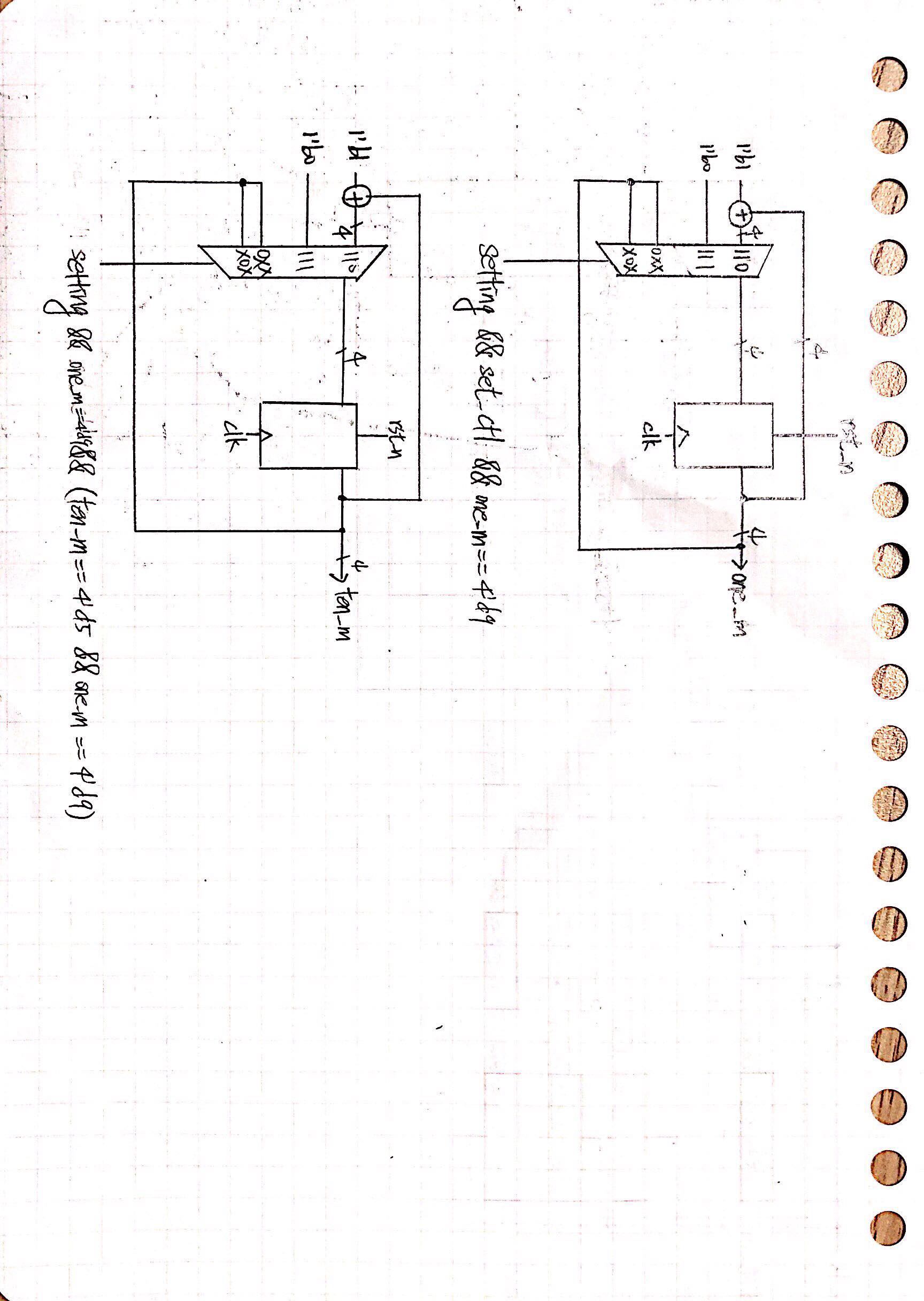
* Draw the block diagram of the design.



**Design Implementation**

* 此設計類似先前做過的時鐘與倒數計時器的結合，其中的minute與hour類似lab6中的時鐘，但為down counter。本次的重點為時間的設置，因此以下將說明set\_m、set\_h的設計。
* set\_m、set\_h

其原理即為up counter，當setting = 1時，允許設置時間，藉由set\_ctl的按紐，每按一下時間變加一。



上圖為分的設置，若是時(set\_h)只要將進位的條件更改即可。

set\_m:

|  |  |  |  |
| --- | --- | --- | --- |
| MUX條件 | | | MUX output |
| setting | set\_ctl | one\_m == 4'd9 | one\_m |
| 1 | 1 | 0 | one\_m + 1 |
| 1 | 1 | 1 | 0 |
| 0 | X | X | one\_m |
| X | 0 | X | one\_m |

|  |  |  |  |
| --- | --- | --- | --- |
| MUX條件 | | | MUX output |
| setting | one\_m == 4'd9 | one\_m == 4'd9 && ten\_m == 4'd5 | ten\_m |
| 1 | 1 | 0 | ten\_m + 1 |
| 1 | 1 | 1 | 0 |
| 0 | X | X | ten\_m |
| X | 0 | X | ten\_m |

set\_h:

|  |  |  |  |
| --- | --- | --- | --- |
| MUX條件 | | | MUX output |
| setting | set\_ctl | one\_h == 4'd9 | one\_h |
| 1 | 1 | 0 | one\_h + 1 |
| 1 | 1 | 1 | 0 |
| 0 | X | X | one\_h |
| X | 0 | X | one\_h |

|  |  |  |  |
| --- | --- | --- | --- |
| MUX條件 | | | MUX output |
| setting | one\_h == 4'd9 | one\_h == 4'd3 && ten\_h == 4'd2 | ten\_h |
| 1 | 1 | 0 | ten\_h + 1 |
| 1 | 0 | 1 | 0 |
| 0 | X | X | ten\_h |
| X | 0 | X | ten\_h |

其output為設定好的時間，將其作為minute與hour的input即為倒數計時器的時間設定。

* I/O pin

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| I/O | led[15] | led[14] | led[13] | led[12] | led[11] | led[10] | led[9] | led[8] |
| VOC | L1 | P1 | N3 | P3 | U3 | W3 | V3 | V13 |
| I/O | led[7] | led[6] | led[5] | led[4] | led[3] | led[2] | led[1] | led[0] |
| VOC | V14 | U14 | U15 | W18 | V19 | U19 | E19 | U16 |
| 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] |
| VOC | W7 | W6 | U8 | V8 | U5 | V5 | U7 | V7 |
| I/O | d[3] | d[2] | d[1] | d[0] | clk | start | pause | set\_min |
| VOC | W4 | V4 | U4 | U2 | W5 | T18 | U17 | T17 |
| I/O | set\_hour | setting | rst\_n |  |  |  |  |  |
| VOC | W19 | V17 | U18 |  |  |  |  |  |

**Lab7\_3**

**Design Specification**

* Input: clk,

rst\_n,

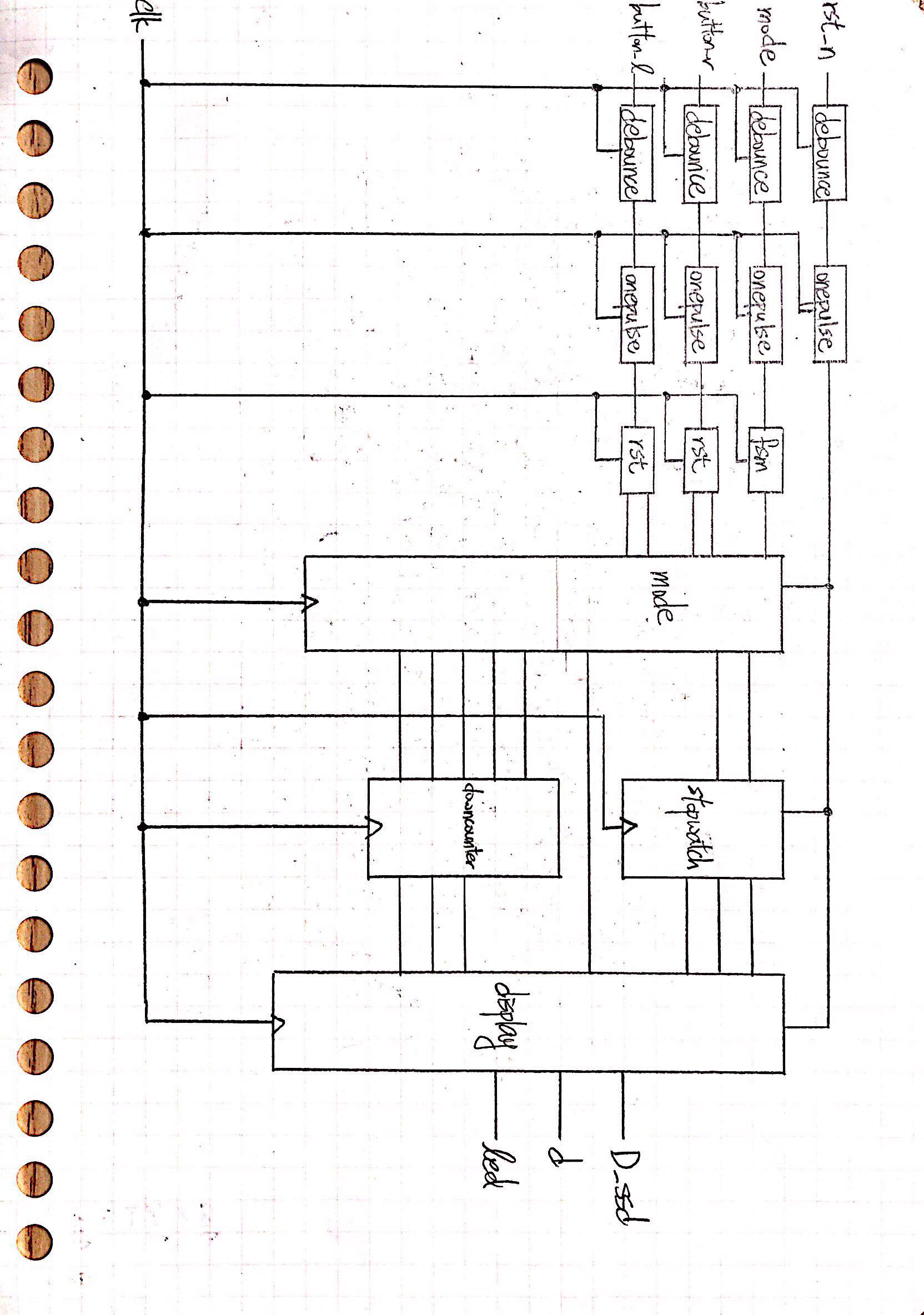
mode, // 模式切換

button\_r,

button\_l

Output: D\_ssd[7:0], d[3:0], led[15:0]

* Draw the block diagram of the design.



leap\_year

**Design Implementation**

* 第三題為第一題與第二題的結合，其中的mode功能為控制模式與按鈕的作用。其餘module皆與第一題與第二題差不多，因此不再討論。
* mode

input:

clk,

rst\_n,

mode,

button\_r\_l, // 右鍵長按

button\_r\_s, // 右鍵短按

button\_l\_l, // 左鍵長按

button\_l\_s, // 左鍵短按

output:

lap,

start\_in,

start,

pause,

setting,

set\_min,

set\_hour

一開始的時候output 皆為0，此處我設計當mode = 0時為stop watch模式，mode = 1時為down counter模式。

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| mode | setting | set\_min | set\_hour | start | pause | lap | start\_in |
| 0 | 0 | 0 | 0 | 0 | 0 | button\_r\_s | button\_l\_s |
| 1 | 0 | 0 | 0 | button\_r\_s | button\_l\_s | 0 | 0 |
| 1 | 1 | button\_r\_s | button\_l\_s | 0 | 0 | 0 | 0 |

其中當mode = 1時，setting <= button\_r\_l。藉由mode的控制，此處將決定按鈕所輸出對應到的功能為何。

* I/O pin

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| I/O | led[15] | led[14] | led[13] | led[12] | led[11] | led[10] | led[9] | led[8] |
| VOC | L1 | P1 | N3 | P3 | U3 | W3 | V3 | V13 |
| I/O | led[7] | led[6] | led[5] | led[4] | led[3] | led[2] | led[1] | led[0] |
| VOC | V14 | U14 | U15 | W18 | V19 | U19 | E19 | U16 |
| 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] |
| VOC | W7 | W6 | U8 | V8 | U5 | V5 | U7 | V7 |
| I/O | d[3] | d[2] | d[1] | d[0] | clk | rst\_n | mode | button\_r |
| VOC | W4 | V4 | U4 | U2 | W5 | T18 | U18 | T17 |
| I/O | button\_l |  |  |  |  |  |  |  |
| VOC | W19 |  |  |  |  |  |  |  |

**Discussion**

　　在第一題的lap設置中，我原本想在counter中先做好分圈，再將其輸出到display中顯示，但遇到的問題是因為會延遲一個clock所以分圈的值會多一秒，且這樣要多接線，造成不必要的浪費。因此最後我將其做在display中，當lap = 1時即控制顯示為分圈時的數字。

　　第二題為將時鐘改為倒數，並可以自己設定時間。在這題沒遇到太大的問題，主要是最後當歸零時要全部的led燈全亮，由於我原本的控制跟時與分的設定寫在一起，一直無法讓他在正確的時間亮，之後將其單獨分開判斷便成功了。

　　第三題需要把第一題與第二題中的debounce、onepulse、fsm、rst接在外面再做為mode的input，無法直接藉由判斷mode的值來決定按鍵功能。此處須注意mode的output與down counter、stop watch 的關聯。

**Conclusion**

這次的實驗幾乎都是重組之前做過的module並加以小修改，比較困難的是第三題不能直接用mode的值來決定按鍵功能，跟軟體的想法很不一樣。