邏輯設計實驗Lab9結報

105060012張育菘

**1 VGA displaying functions.**

**1.1 Inputs of the VGA controller are clk, reset, en and outputs of the VGA controller are hsync, vsync, vga\_red[3:0], vga\_green[3:0], vga\_blue[3:0].**

**1.2 At the beginning or when reset (button) is pressed, the VGA display shows the image (e.g. amumu.jpg). The VGA image stay still until en (button) is pressed.**

**1.3 Pressing odd times en button to start/resume scrolling. Pressing even times en button to pause scrolling. Counter for en press is reset to zero when reset is pressed.**

Design Specification

Input：clk,rst,en,;

Output：led,hsync,vsync,vgaRed[3:0],vgaGreen[3:0],vgaBlue[3:0];

Inout：PS2\_DATA,PS2\_CLK;

\*W19控制rst \*T17控制state

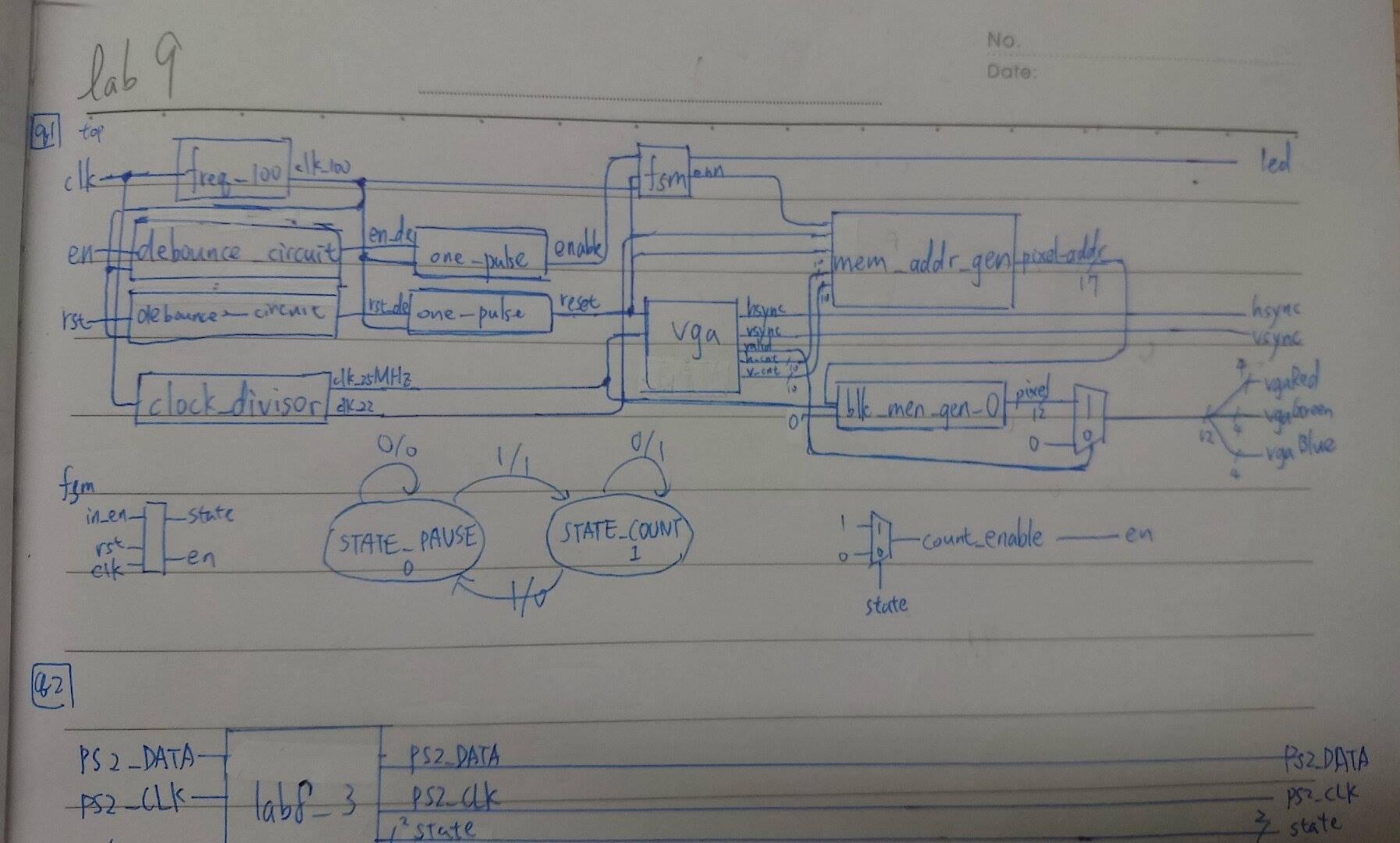
Design Implementation

Logic function :

1. top：

此top module，專門拿來呼叫其他小moudule的。

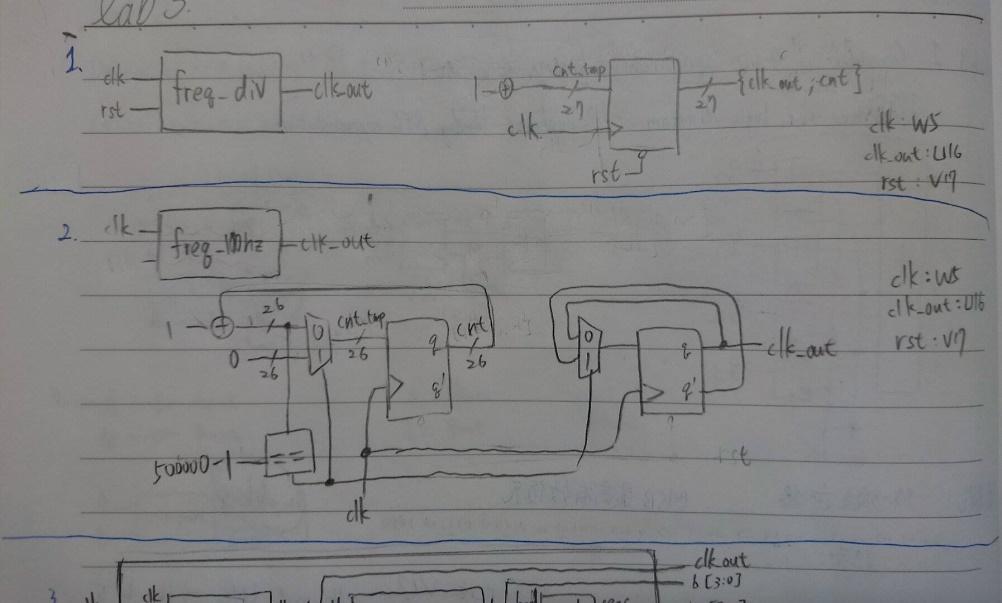
還有給定vgaRed, vgaGreen, vgaBlue 的值{vgaRed, vgaGreen, vgaBlue} = (valid==1'b1) ? pixel:12'h0;



1. freq\_100hz：

此為除頻的module，輸出100hz的clk\_out當作debounce\_circuit、one\_pulse以及fsm的clk。

* 這裡不能接top的input rst，因為那裡的rst是希望count\_up重新開始數。當rst=0時，freq\_100hz輸出的clk持續為0🡪freq\_100hz不做事。

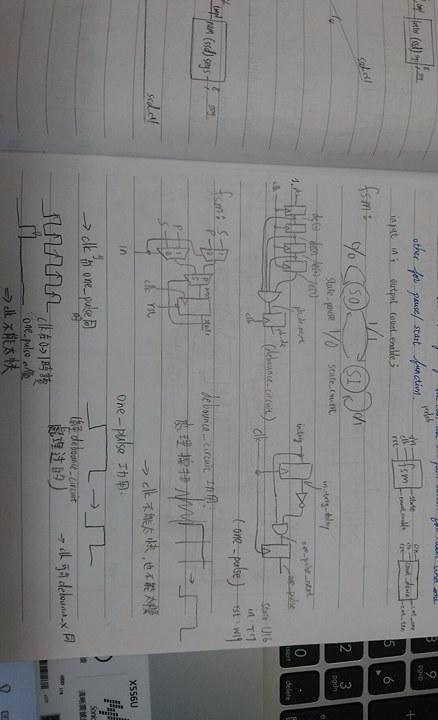


1. debounce\_circuit：

此為**除按鈕雜訊**的module，原理是，當debounce\_circuit收到連續4個1時才產生一個0🡪1的訊號，這樣可以避免收到前段起伏不定的雜訊，達到除雜訊的效果。

且要接100hz的clk，因為若接此的clk頻率太快就無法達到除雜訊的效果，太慢可能會超過按鈕維持穩定值的時間，因此我選擇100hz作為debounce\_circuit的clk。

* 這裡不能接top的input rst，因為那裡的rst是希望count\_up重新開始數。當rst=0時，freq\_100hz輸出的clk\_100持續為0🡪 debounce\_circuit不做事。

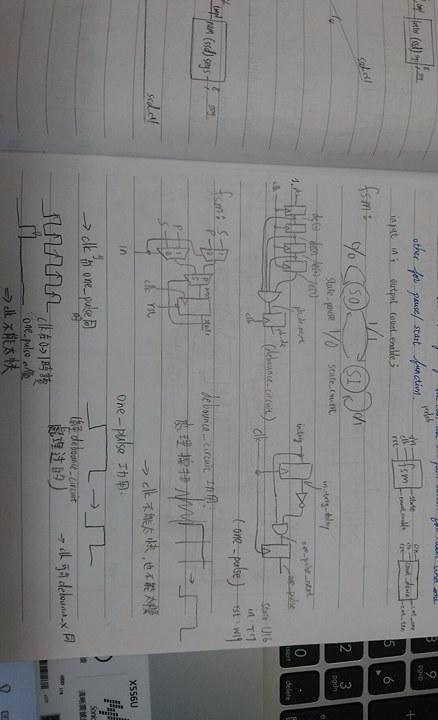


1. one\_pulse：

輸入訊號經過debounce\_circuit處理後，其週期可能會超過1個clk的時間，因此需要此module使訊號的週期為1個clk的時間。

這裡的clk我也是接與debounce\_circuit同的100hz。

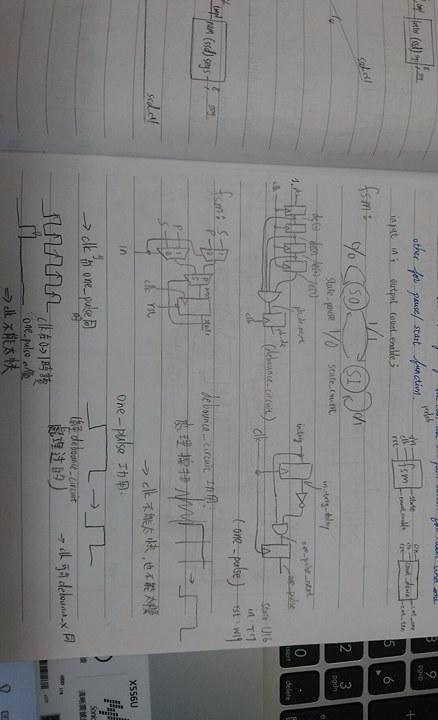
* 這裡不能接top的input rst，因為那裡的rst是希望count\_up重新開始數。當rst=0時，freq\_100hz輸出的clk\_100持續為0🡪 one\_pulse不做事。



1. fsm：

此module做為finite state machine，以接出state(在display內)決定要顯示lap抑或是顯示正在數的狀態。

這裡的clk我是接100hz，因為clk在0🡪1時讀one\_pulse(其clk為100hz)處理過的值，因此top的clk不能太快。



1. Clock\_divisor：

此module是用來產生clk，分別產生clk22(約24hz)以及clk\_25MHz。

1. vga\_controller：

此module是用來更新畫面的，產生的h\_cnt和v\_cnt可以用來讀圖片。

1. mem\_addr\_gen：

此module是用來讀圖片檔的，由於要產生往下滑的感覺，因此我設計position\_next = position + 1;讓y軸每經一個clk就讀不一樣的位置

Result

1. 圖二：當按下T17時圖片暫停往上移。
2. 圖三，當按下”W19”(rst)時，回歸至上移前的畫面。

圖一

 圖二



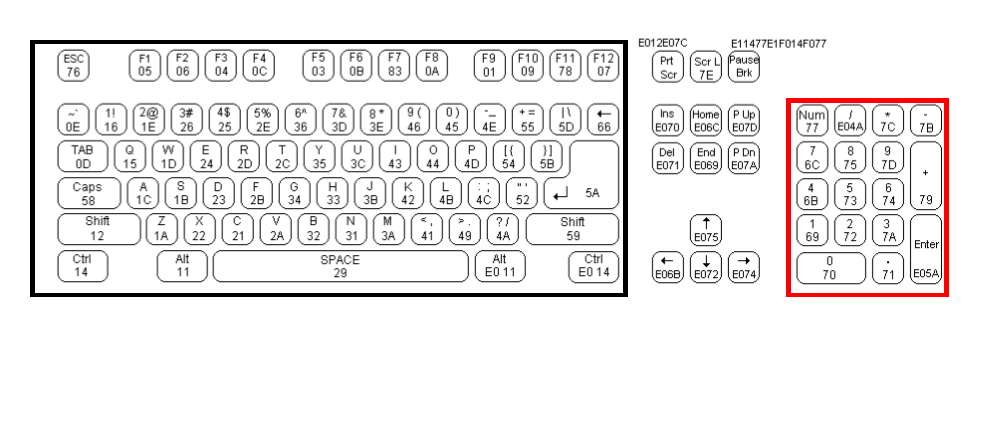
Discussion

1. adder

當{in1+in2}>=4'd16時會有overflow的情形產生，我是利用"規律性"發現當有overflow情形產生時，需加4’d6並可以解決此問題。

1. 設計key\_down[last\_change] && key\_valid去讀值，因為key\_valid是一當按下或放下才會產生的訊號，因此拿他與key\_down[last\_change]做交集可判斷last\_change是否有被按下去，最後至mux轉換成BCD(Binary-Coded Decimal)的形式。
2. 設計mem\_addr\_gen 時，position\_next = position + 1;讓y軸每經一個clk就讀不一樣的位置

**2 Calculator display.**

**2.1 Combine the key board controller and VGA displaying controller to design a calculator with 2-digit addition/subtraction/multiplication. The display function should be the same as usual calculator or APP in the smartphone.**

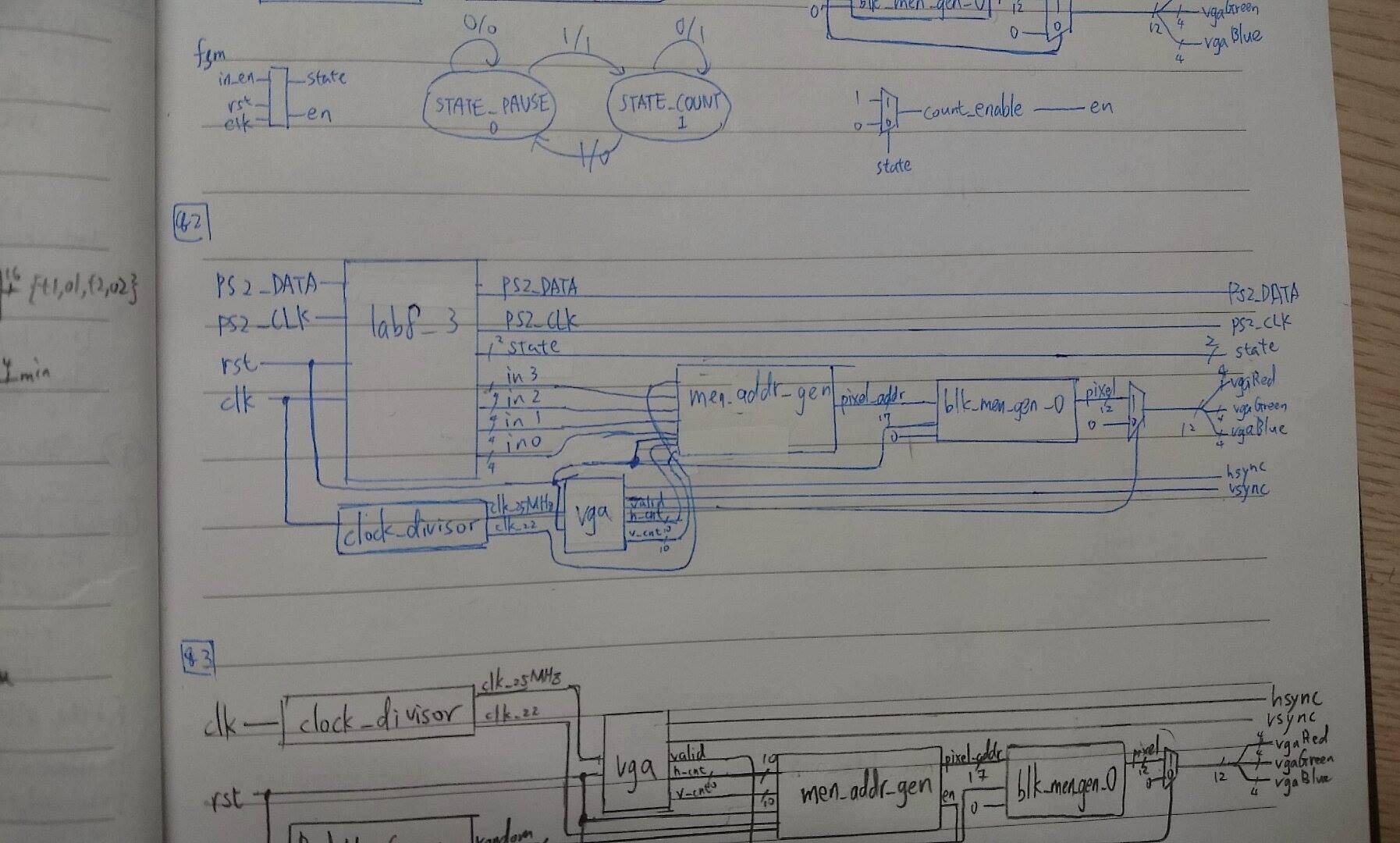
| state[1] | state[0] | clk | rst |
| --- | --- | --- | --- |
| P1 | N3 | W5 | V17 |

Design Specification

Input：clk,rst;-\*

Output：hsync,vsync,vgaRed[3:0],vgaGreen[3:0],vgaBlue[3:0],state[1:0];

Inout：PS2\_DATA,PS2\_CLK;

\*V17控制rst，當rst==0時會產生rst的訊號

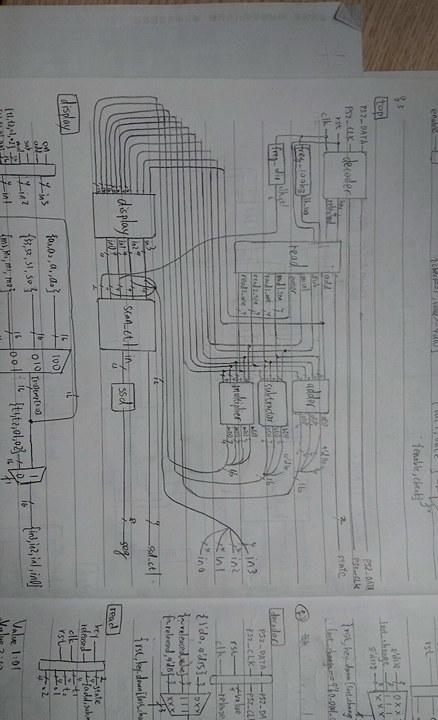
Design Implementation

Logic function :

1. top：

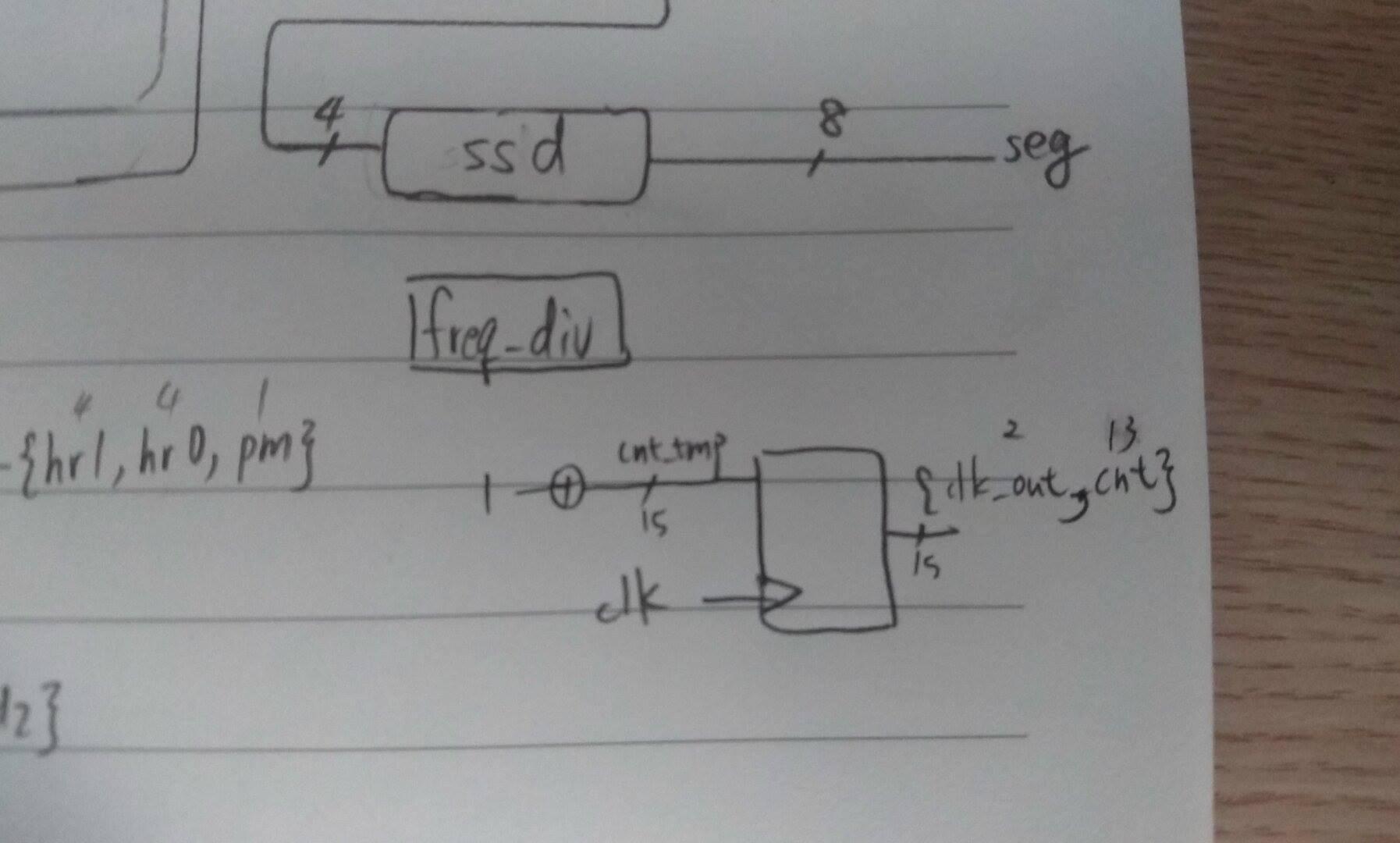
此top module，專門拿來呼叫其他小moudule的。

1. lab8\_3：



1. freq\_div：

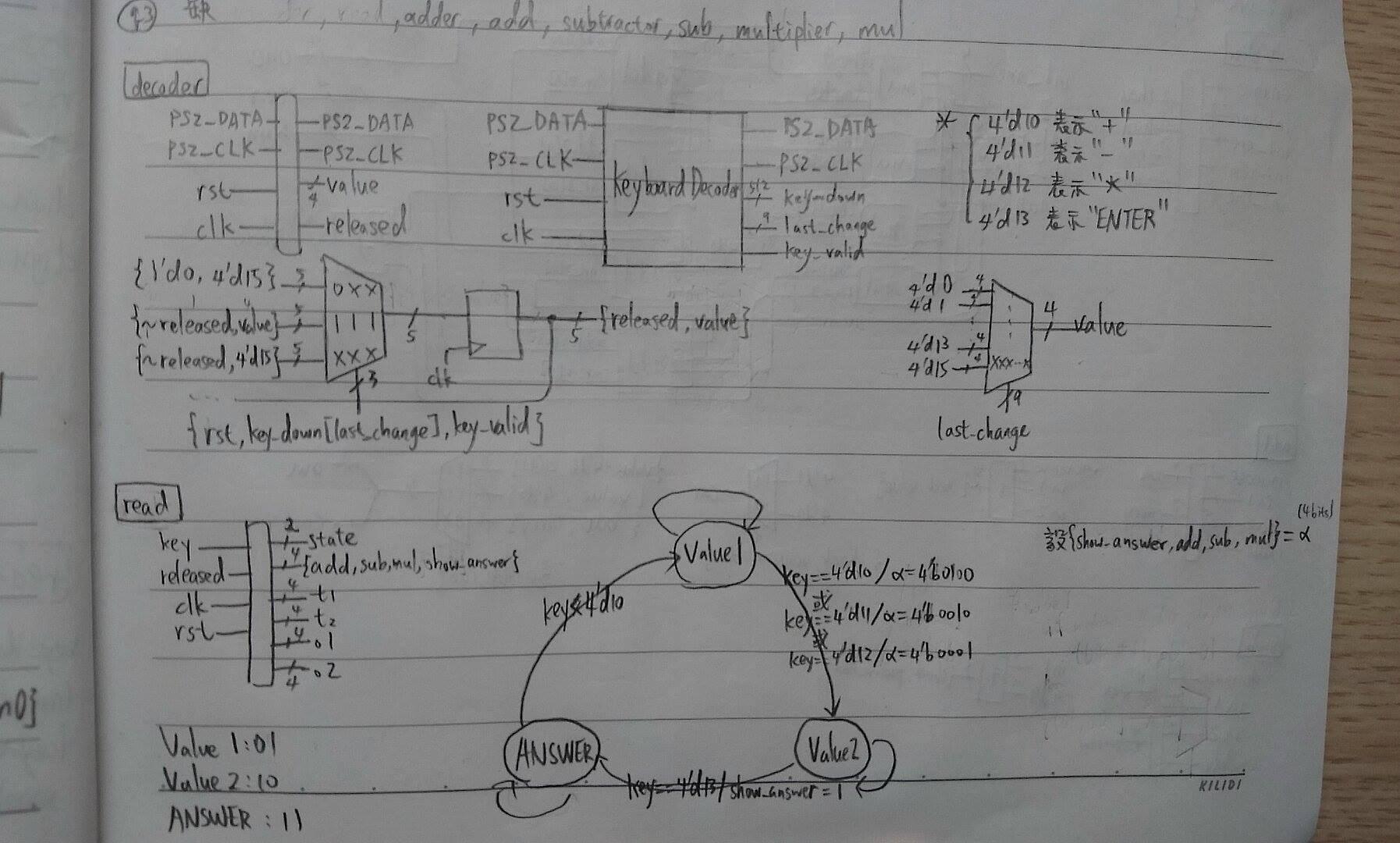
此為除頻的module，輸出的clk\_ctl當作scan\_ctl的clk，其頻率極高。



1. KeyboardDecoder：

這個module主要的目的是讀取鍵盤的值。

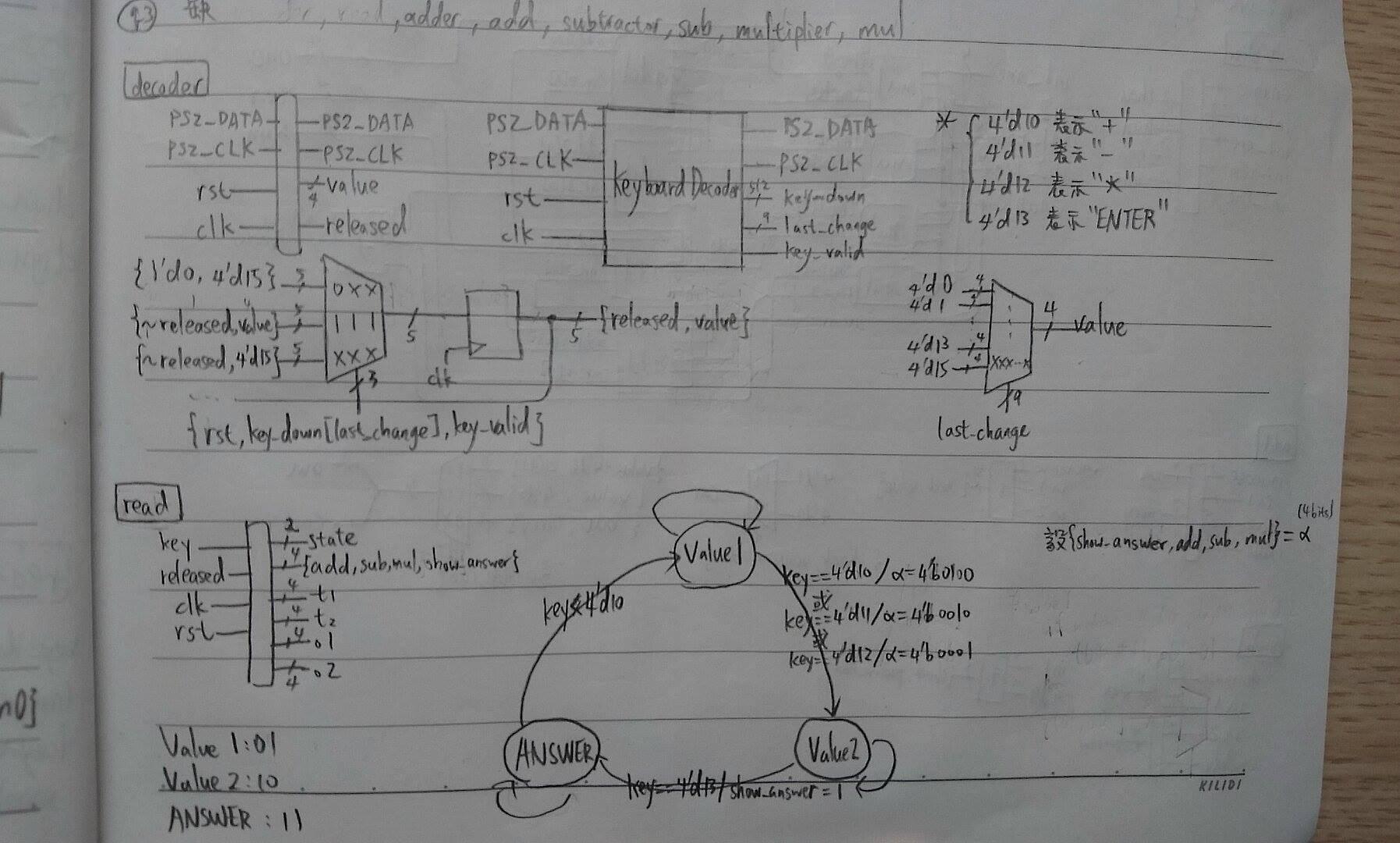
* key\_down是表示那一個鍵有沒有被按下去；last\_change是表示按鍵按下去的值；key\_valid在按鍵被按下去或放開的當下方會產生一個突波(理論上)。



1. decoder：

此module是用來讀取從key\_board輸入值，並轉成BCD的形式。

* 取值的方法與第一題同，都是用key\_down[last\_change] && key\_valid的方式判斷last\_change為何值的被按下去。
* 把KeyboradDecoder包在這個module的目的在於，使KeyboradDecoder影響範圍只限在這個module內，不會因產生不規則的訊號而破壞結果。
* 此輸出的released是在read module內讀值用。
* 4'd10表示"+"；4'd11表示"-"；4'd12表示"\*"；4'd10表示"ENTER"



1. read：

此module是用來讀取兩個數的值。

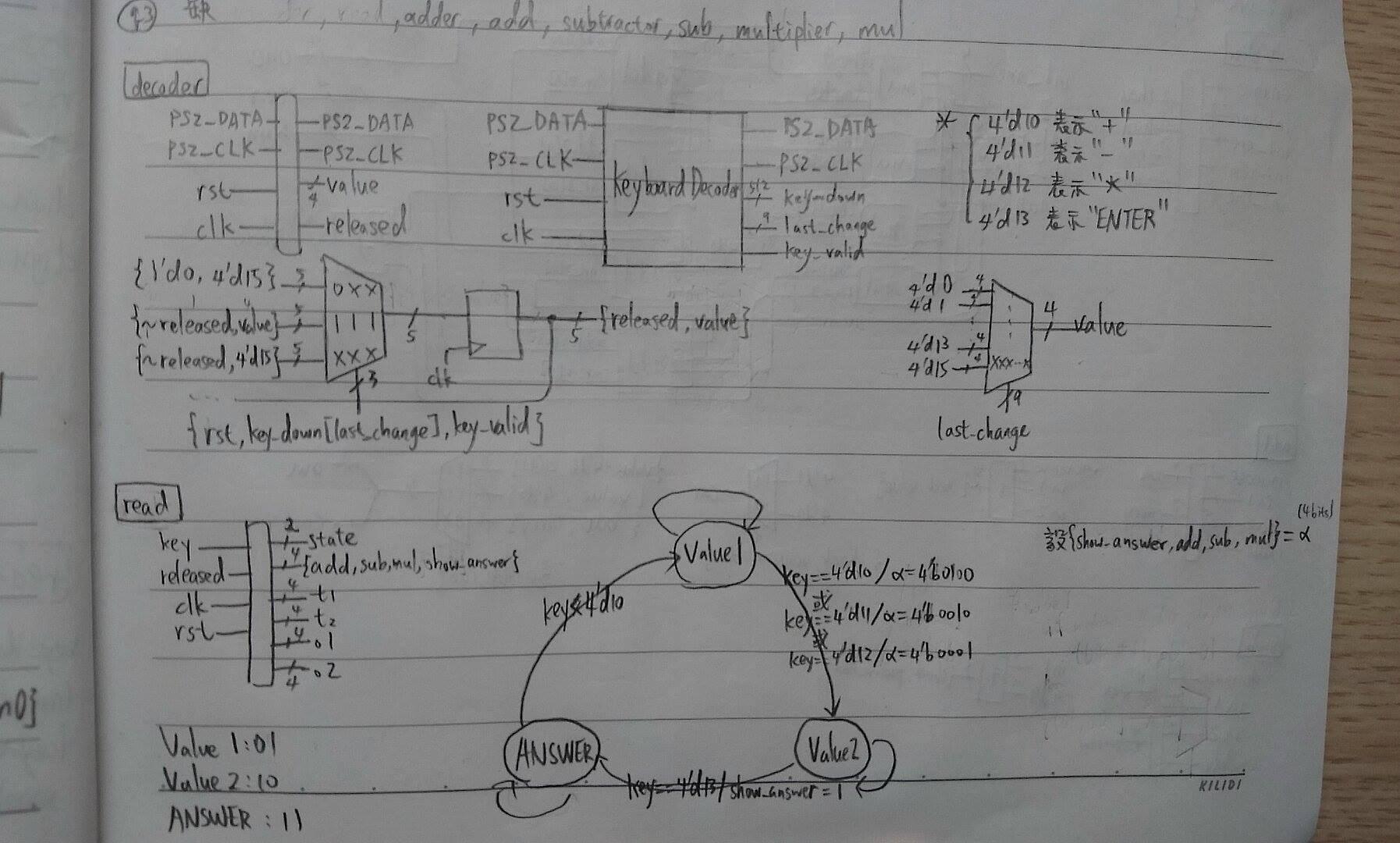
* 特別在此設計切換每一個state的條件都不同，目的在於能避免彼此間互相影響，以及避免key\_valid產生多的突波使state跳得太快。
* 從`Value1 🡪`Value2是當讀取到"+","-"or"\*"的鍵盤訊號才會切換

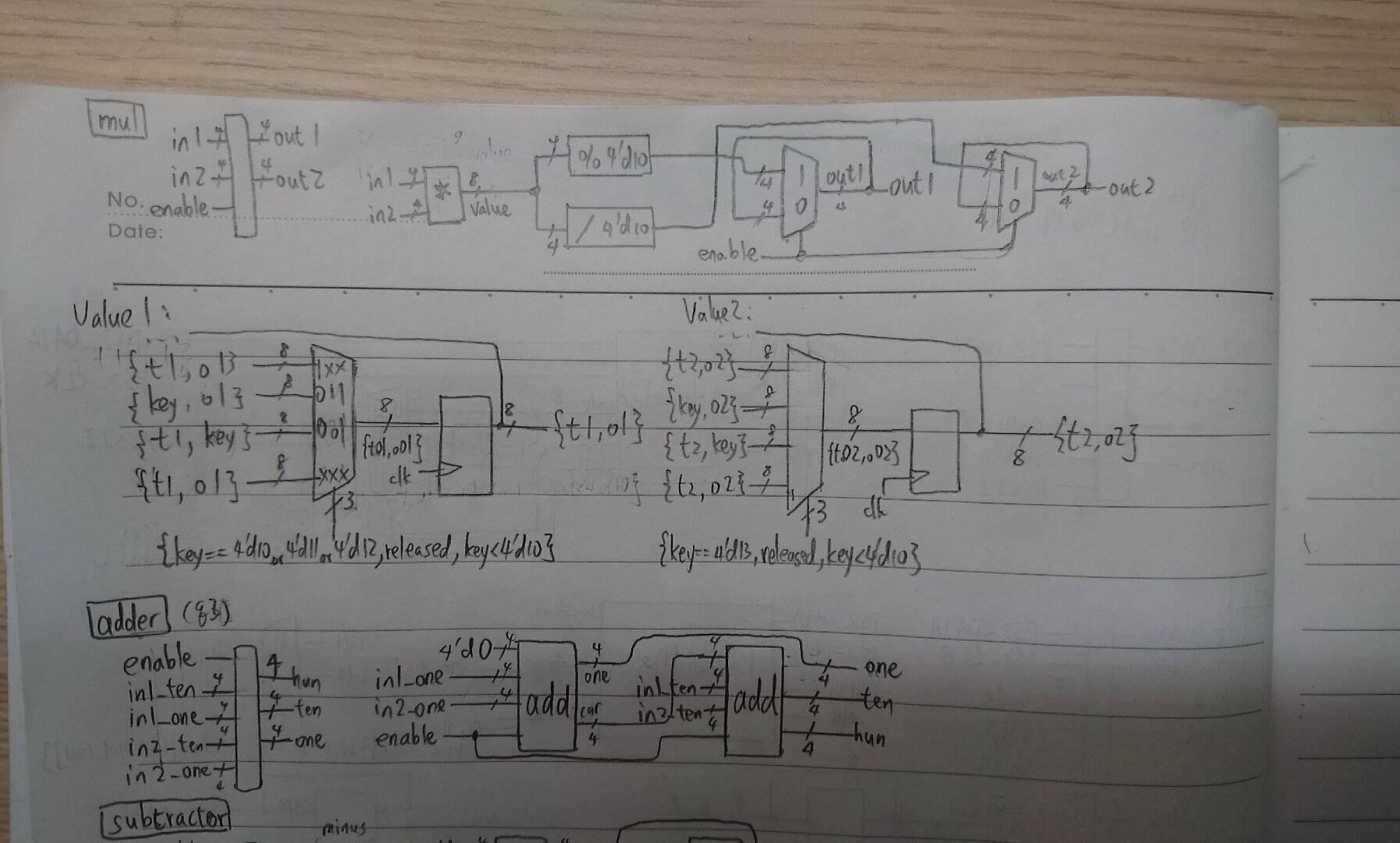
從`Value2🡪`ANSWER是當讀取到"ENTER"的鍵盤訊號才會切換

從`ANSWER 🡪`Value1是當讀取到”數字”的鍵盤訊號才會切換

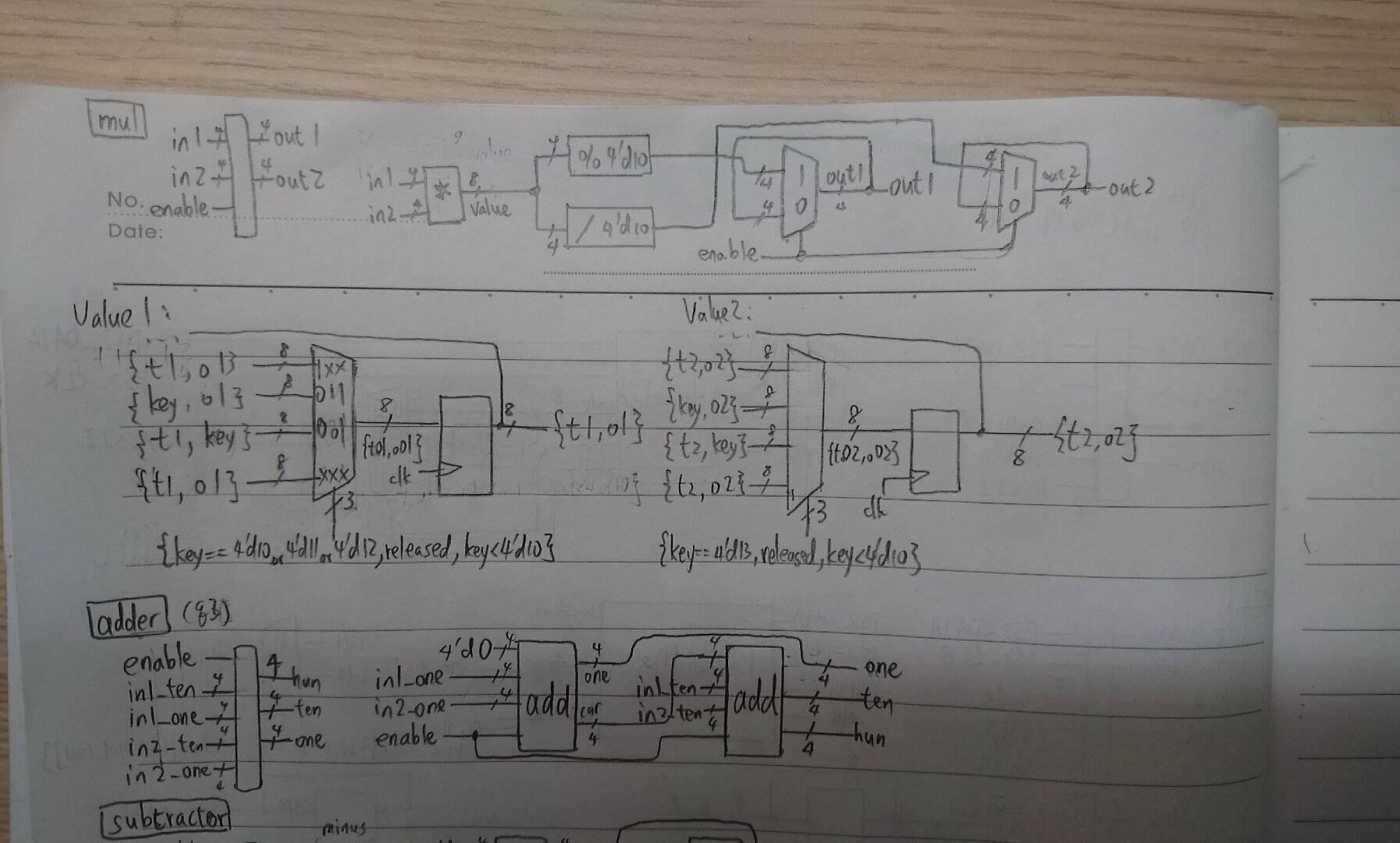
* state==`Value1表示讀取第一個數；state==`Value2表示讀取第二個數

state==`ANSWER表示顯示運算結果。





1. adder：

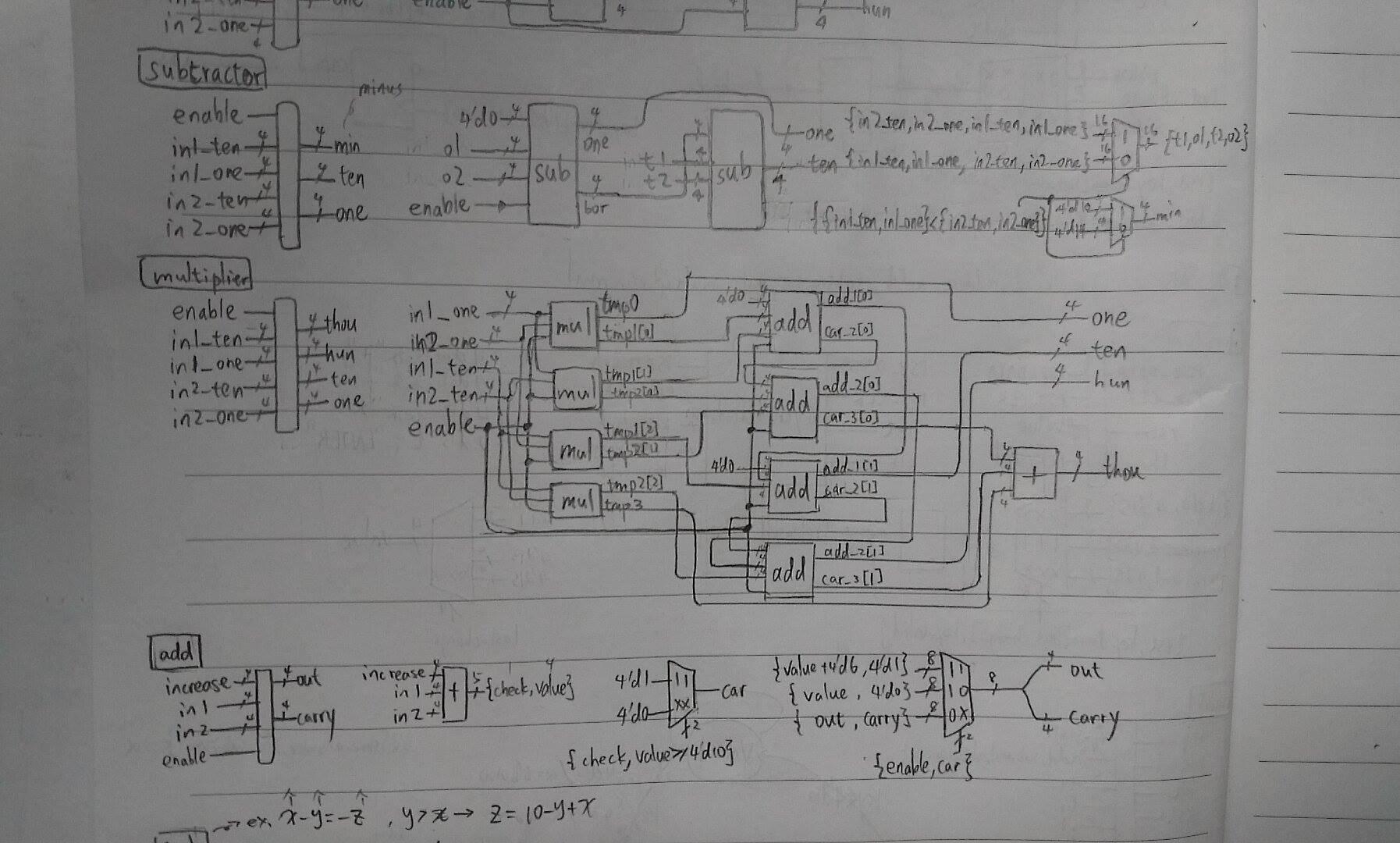
此module是用來做二位數的加法，內含add(做一位數的加法)。

* enable是從read module接出來的add訊號。

1. subtractor：

此module是用來做二位數的減法，內含sub(做一位數的減法)。

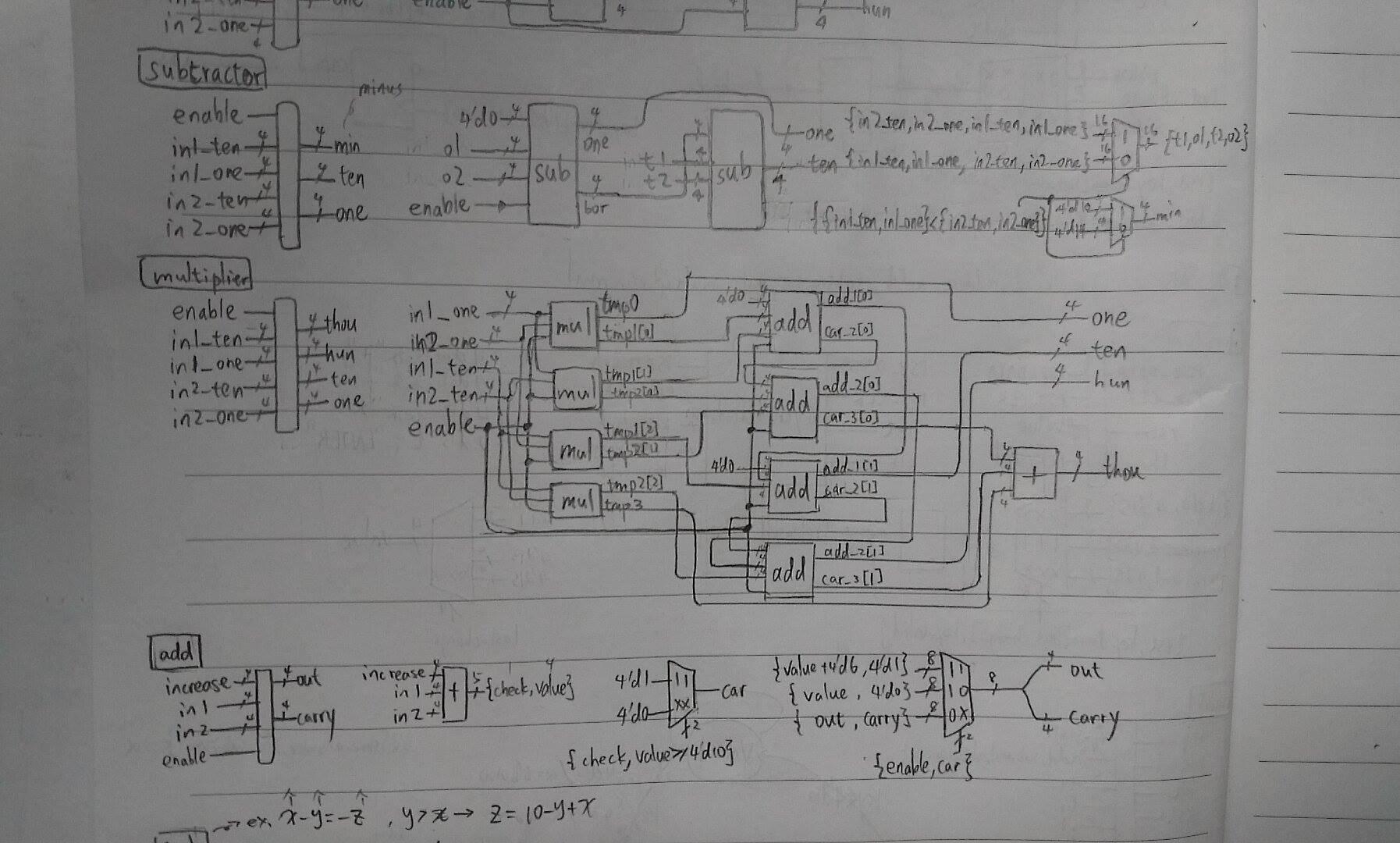
* min是當"(較小的數) - (較大的數)"產生的"負號"，在七段顯示器上顯示。
* enable是從read module接出來sub訊號。



1. multiplier：

此module是用來做二位數的乘法，內含add(做一位數的加法)以及mul(做一位數的乘法)。

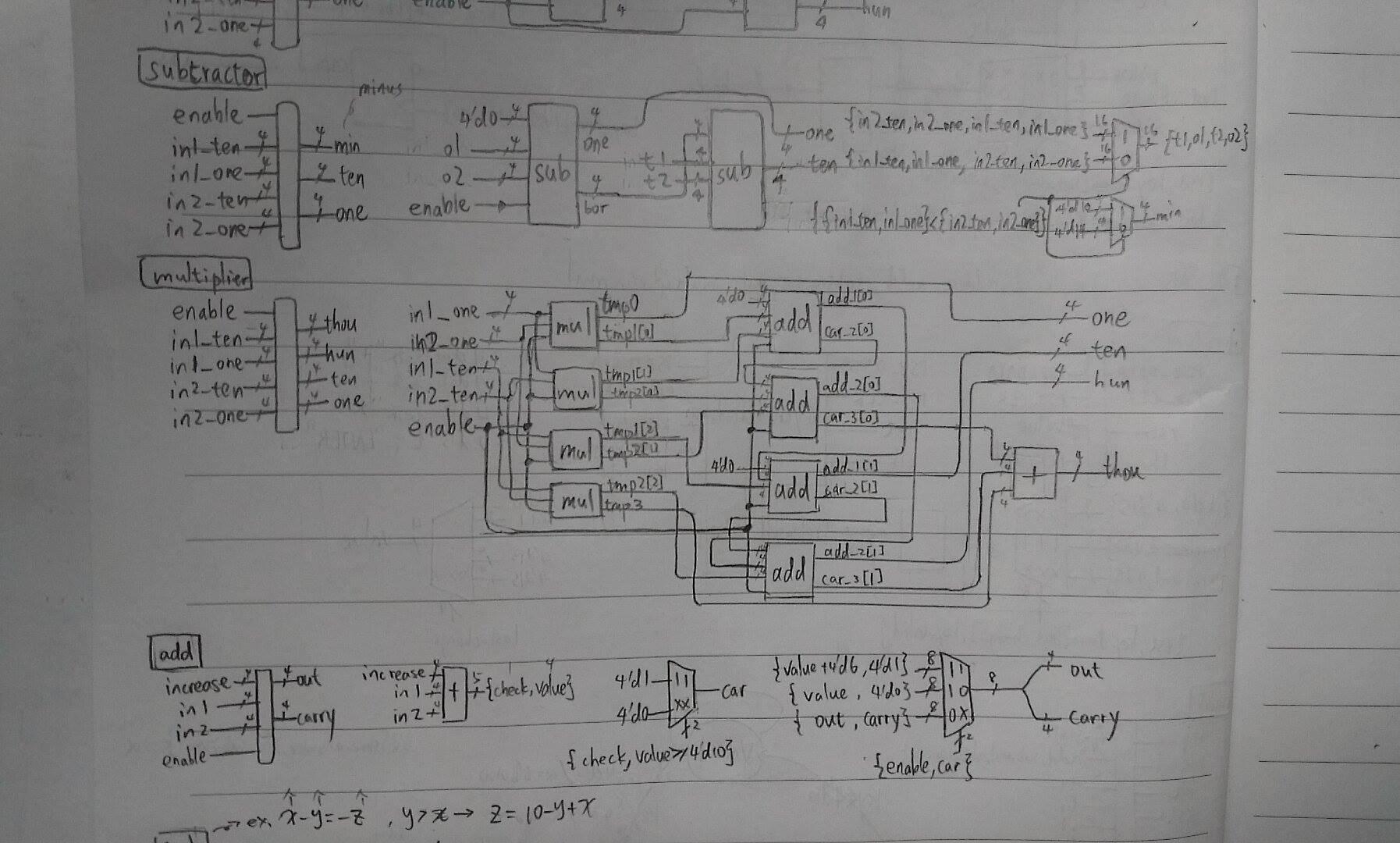
* 由於二位數的乘法是4個一位數乘法產生的值，依照位數一一做相加，因此才會包add以及mul module。
* enable是從read module接出來的mul訊號。



1. add：

此module是用來做一位數的加法。

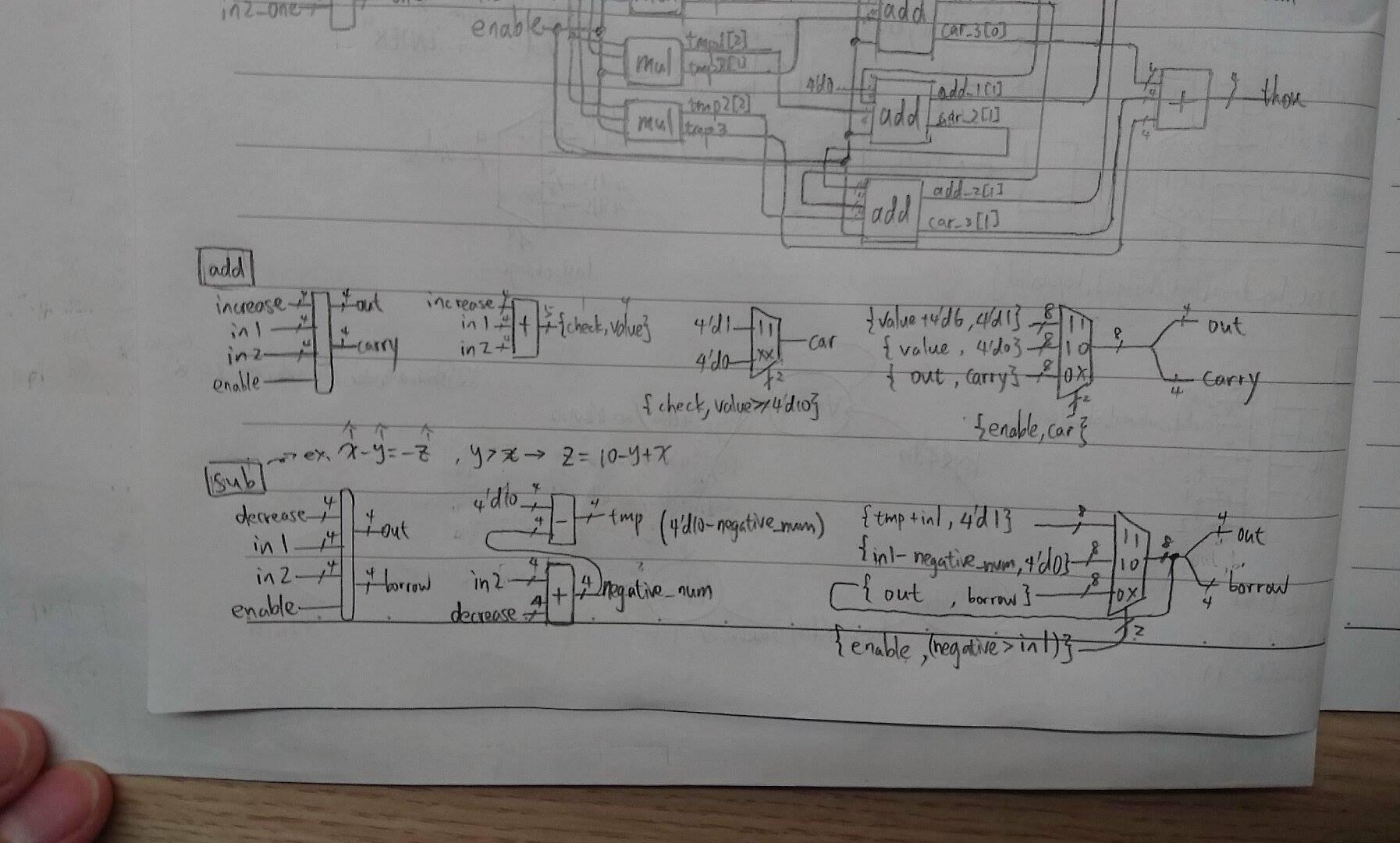
* 使用enable作為mux的判斷值，可以使在enable==1的當下才會運算，其餘時間則維持當下的值。
* 當increase + in1 + in2超過4'd15時會有overflow的產生，因此多接一個check訊號出來判斷是否有overflow的產生，若有overflow的產生，便使運算出來的值再加上4'd6(經驗使然)使結果產生應有的值。



1. sub：

此module是用來做一位數的減法。

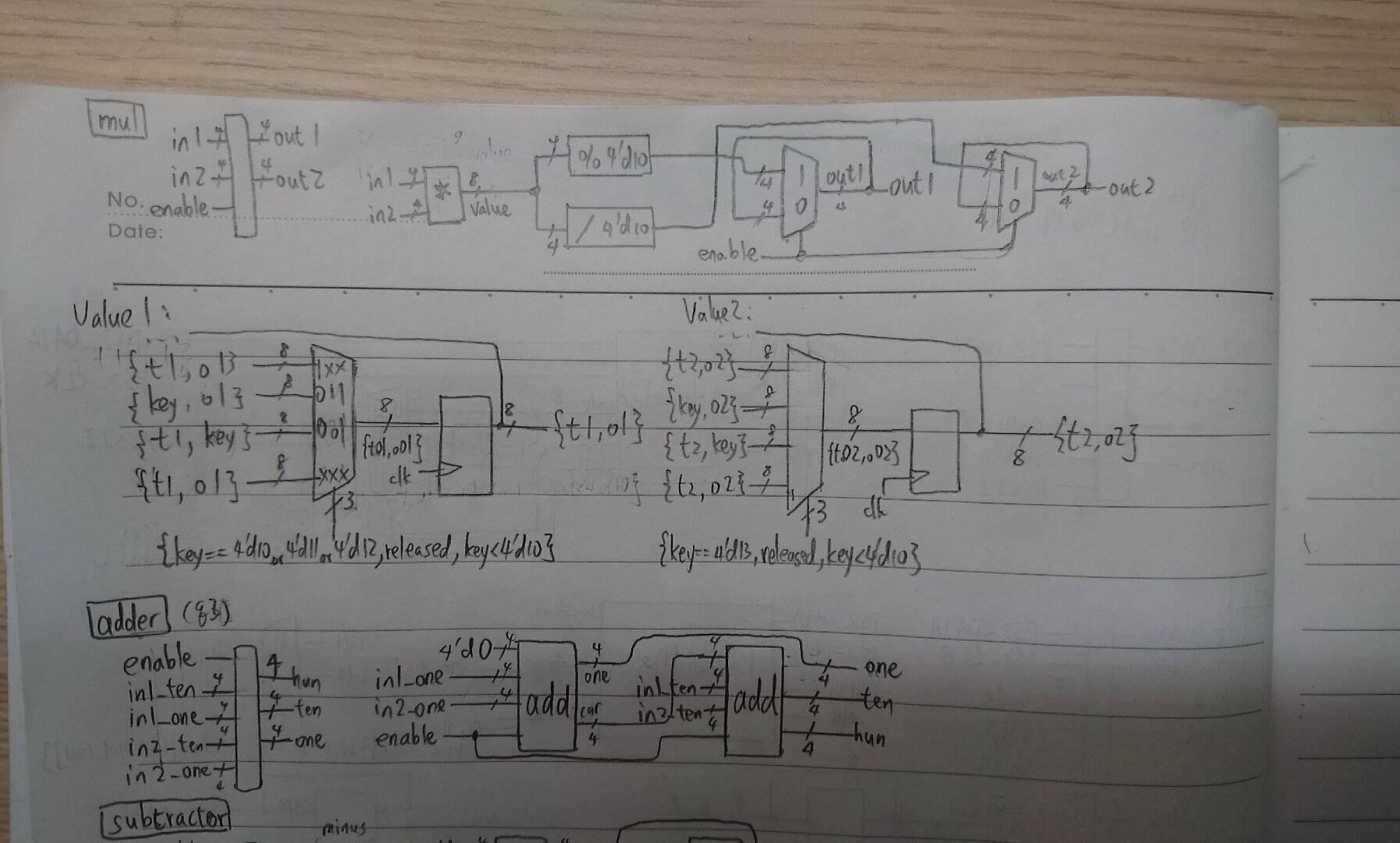
* 使用enable作為mux的判斷值，可以使在enable==1的當下才會運算，其餘時間則維持當下的值。
* 想法來源：ex. X-Y=-Z, Y>X 🡪 Z = 10 – Y + X



1. mul：

此module是用來做一位數的乘法。

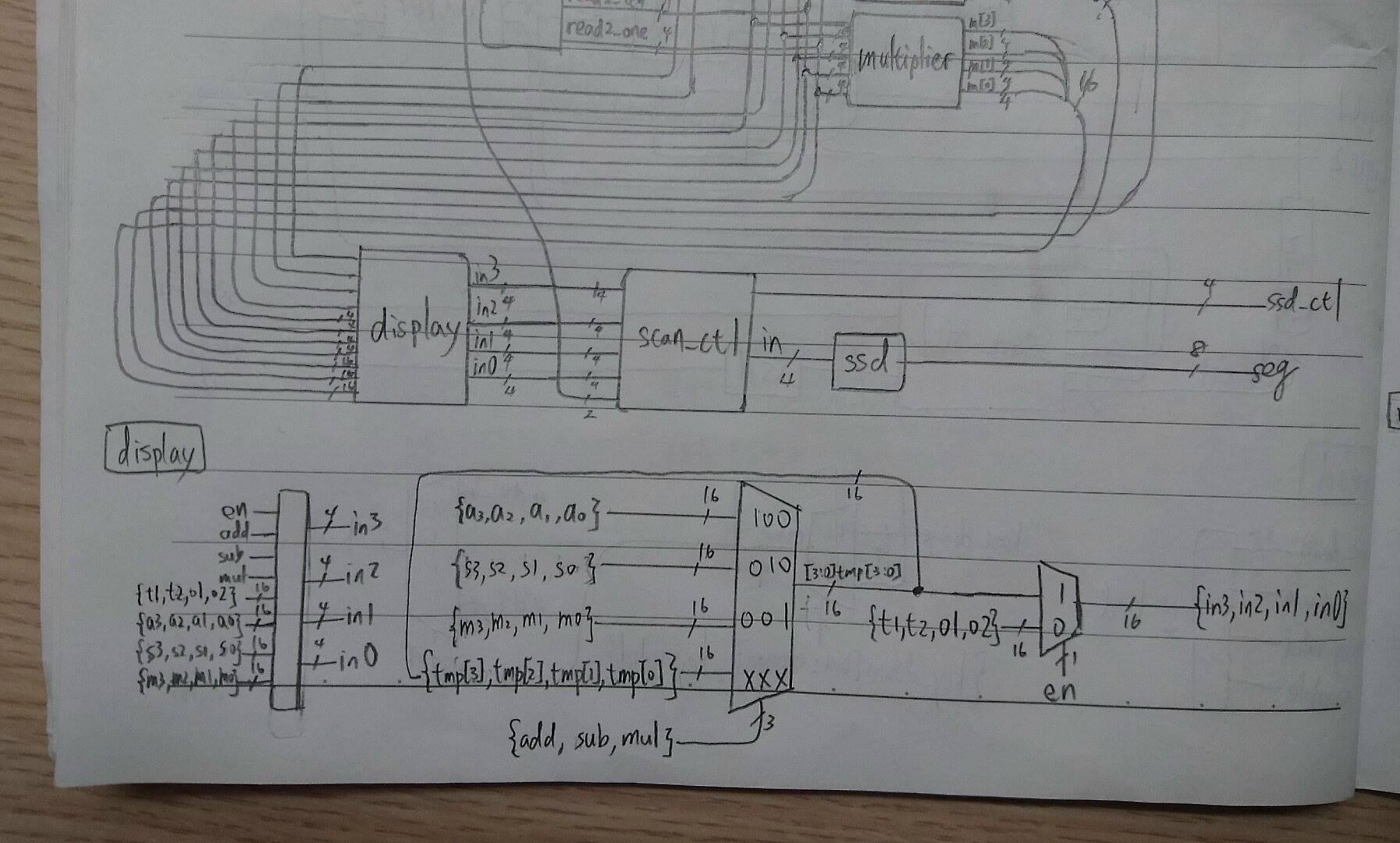
* 使用enable作為mux的判斷值，可以使在enable==1的當下才會運算，其餘時間則維持當下的值。



1. display：

此module是用來決定men\_addr\_gen的input :in3, in2,in1,in0。

* en是從read module接出來的enter訊號，目的是判斷要顯示的數是計算前抑或是計算後的值。
* 這裡接{add,sub,mul}作為mux訊號用意在判斷要顯示經個何種運算的結果。



1. Clock\_divisor：

此module是用來產生clk，分別產生clk22(約24hz)以及clk\_25MHz。

1. vga\_controller：

此module是用來更新畫面的，產生的h\_cnt和v\_cnt可以用來讀圖片。

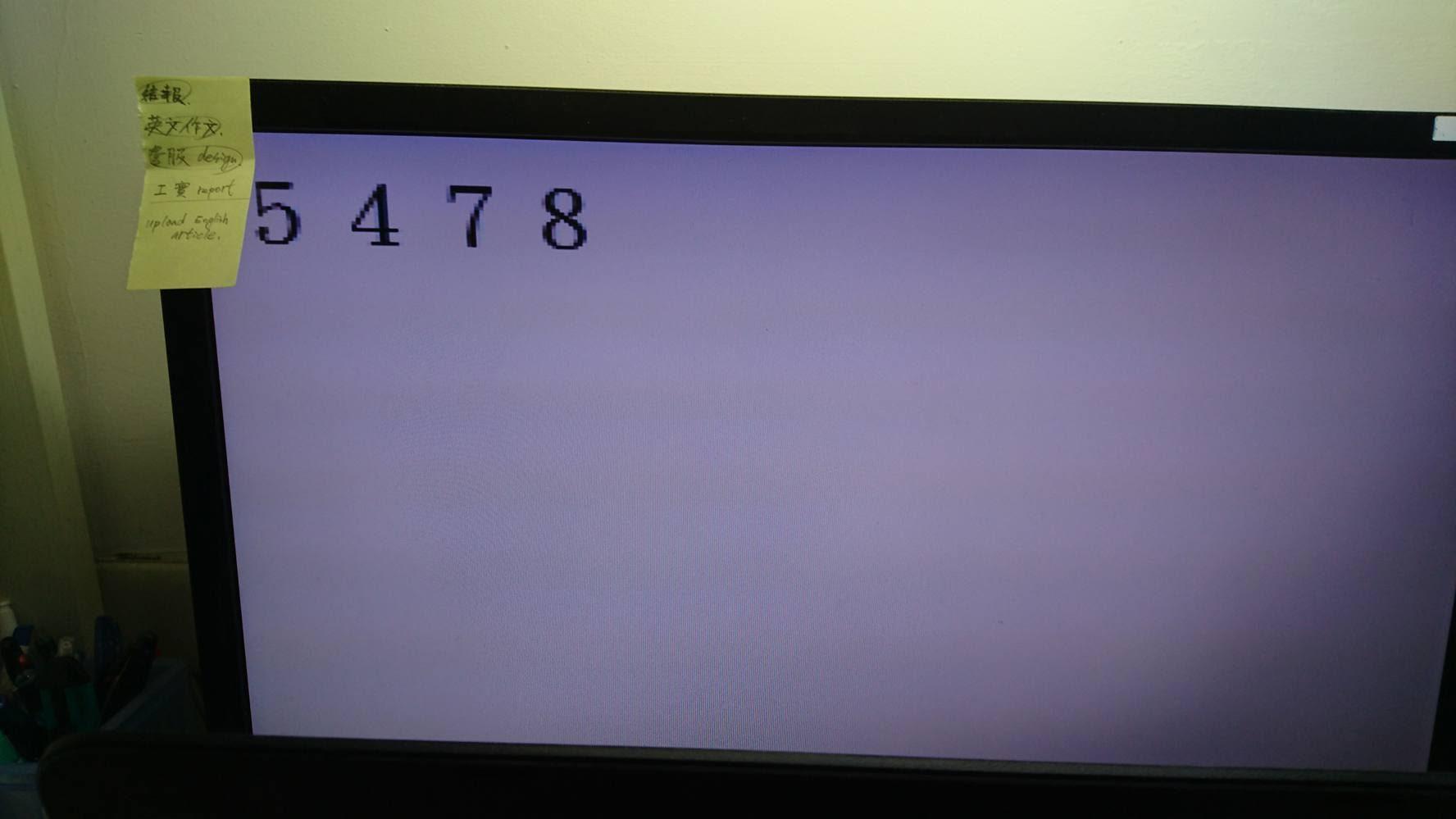
1. mem\_addr\_gen：

此module是用來讀圖片檔的。因此有從display接出in3, in2,in1,in0訊號，作為在某個位置要讀哪一個圖片的依據。

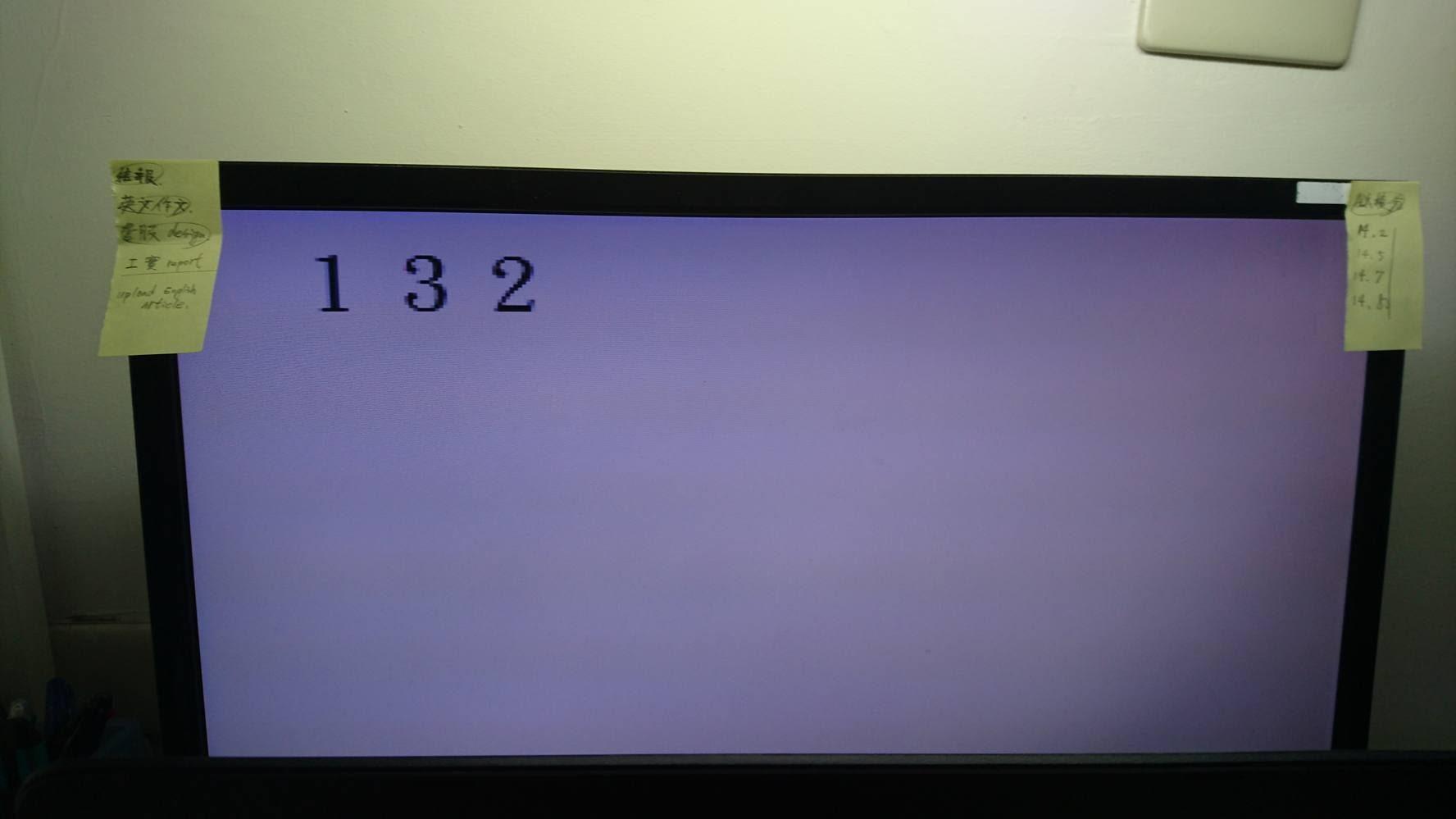
Result (state==`Value1(2’b01)表示讀取第一個數；state==`Value2(2’b10)表示讀取第二個數state==`ANSWER(2’b11)表示顯示運算結果。)

1. 圖一~圖四是在進行"19 + 59 = 78"的過程(加法)。

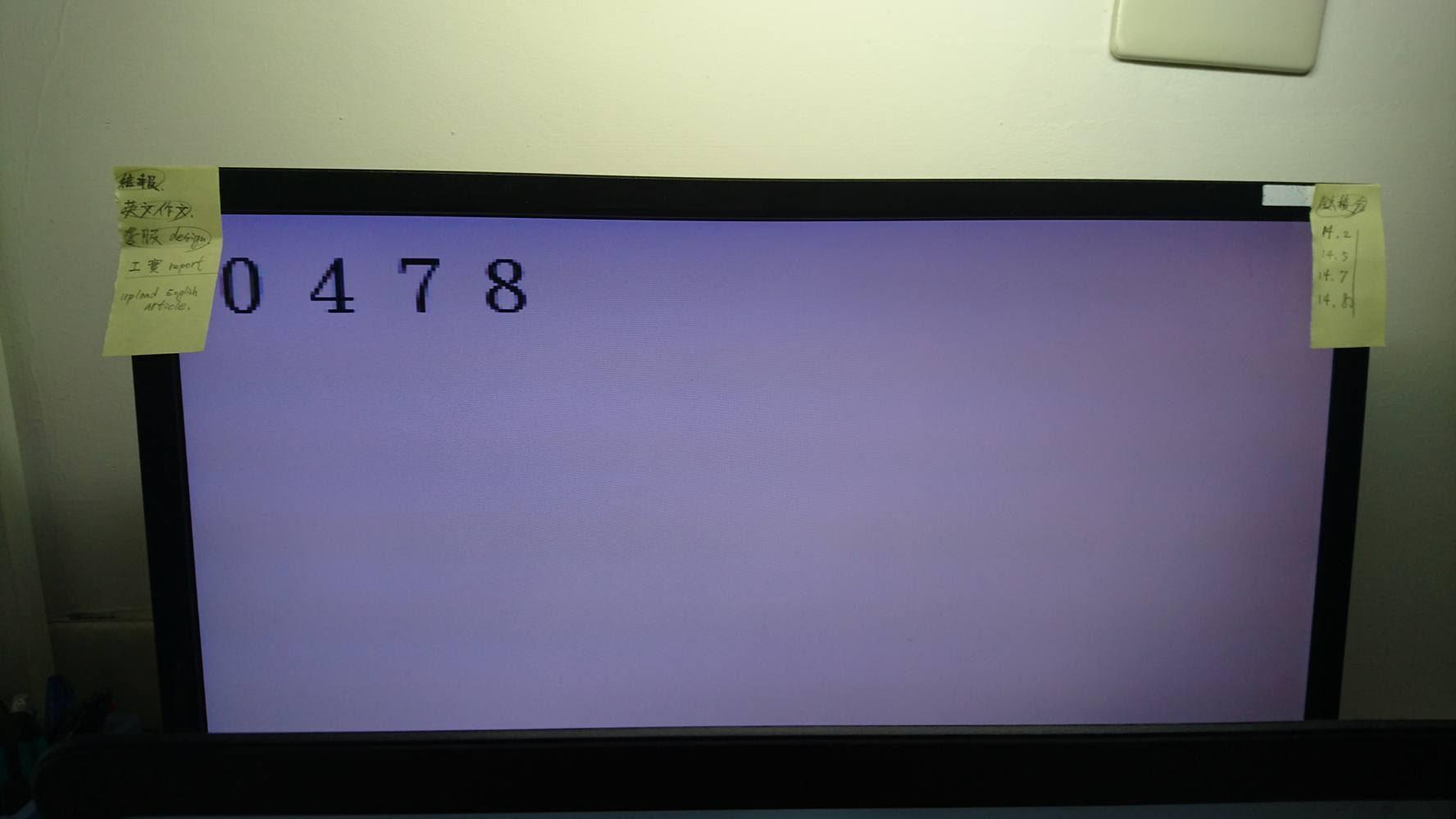
1. 圖一二”54+78=132”。
2. 圖三四”04\*78=312”。
3. 圖五六”05-78 = -73”。



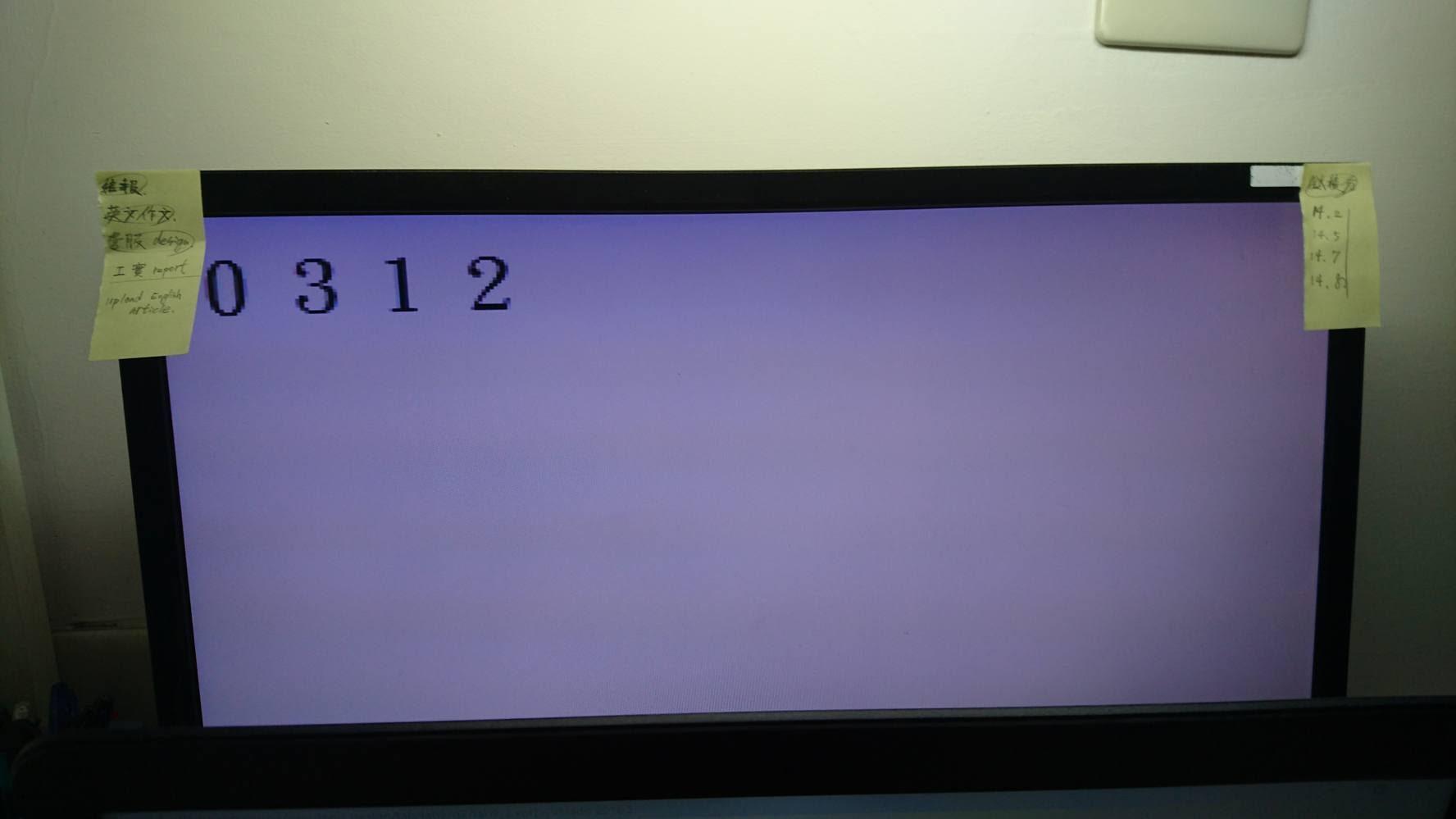
圖二



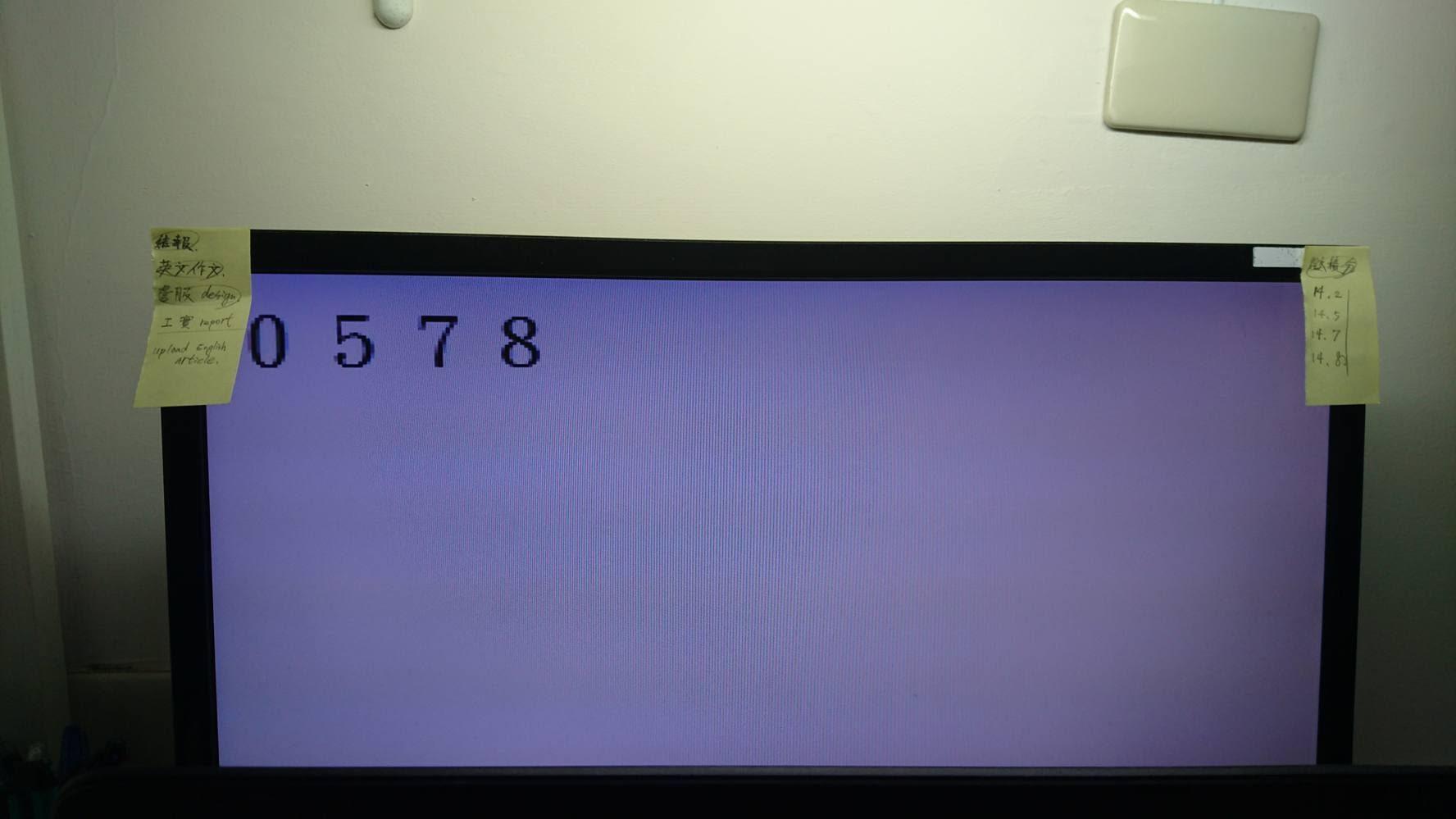
圖三



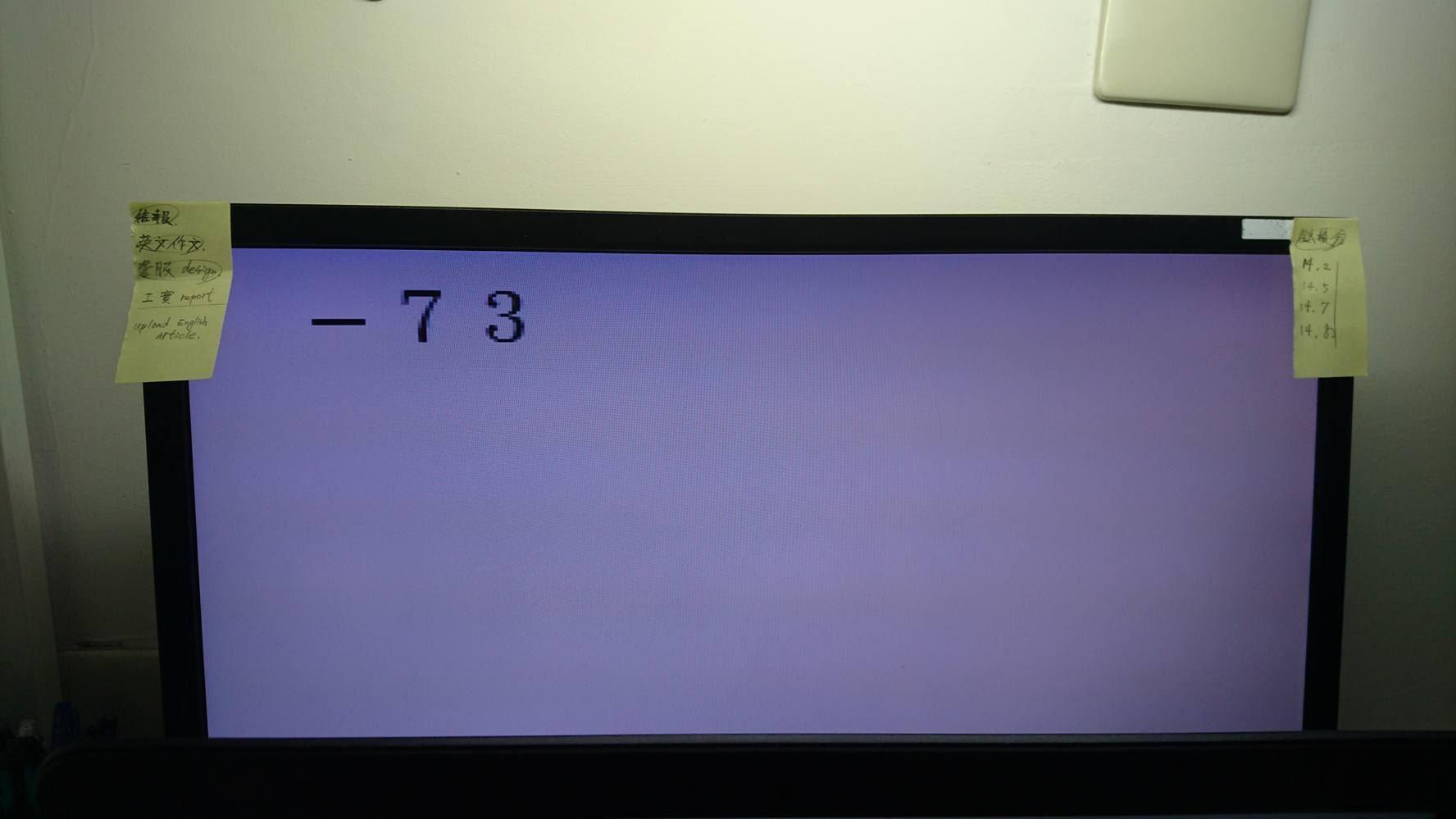
圖四



圖五



圖六



Discussion

1. decoder

把KeyboradDecoder包在這個module的目的在於，使KeyboradDecoder影響範圍只限在這個module內，不會因產生不規則的訊號而破壞結果。

1. read

特別在此設計切換每一個state的條件都不同，目的在於能避免彼此間互相影響，以及避免key\_valid產生多的突波使state跳得太快。

1. add

當increase + in1 + in2超過4'd15時會有overflow的產生，因此多接一個check訊號出來判斷是否有overflow的產生，若有overflow的產生，便使運算出來的值再加上4'd6(經驗使然)使結果產生應有的值。

1. sub

想法來源：ex. X-Y=-Z, Y>X 🡪 Z = 10 – Y + X。

1. mem\_addr\_gen

做法是先給定個數字圖片左上角的x軸和y軸的值，pixel\_addr在應其需要讀取正確的值。

**3 TETRIS element generator**

**3.1 Generate basic elements of TETRIC (as follows) randomly in the VGA monitor, and plot each of them in the center of the first row of the display, which is a 10 x 20 (WxH) square 2D playing space.**

**3.2 Each generated basic element moves down by the step of a square at the speed of 1Hz. Finally, they disappear below the playing space. When a basic element disappears, a new basic element is generated again and fall down again repeatedly.**

Design Specification

Input：clk,rst;

Output：hsync,vsync,vgaRed[3:0],vgaGreen[3:0],vgaBlue[3:0];

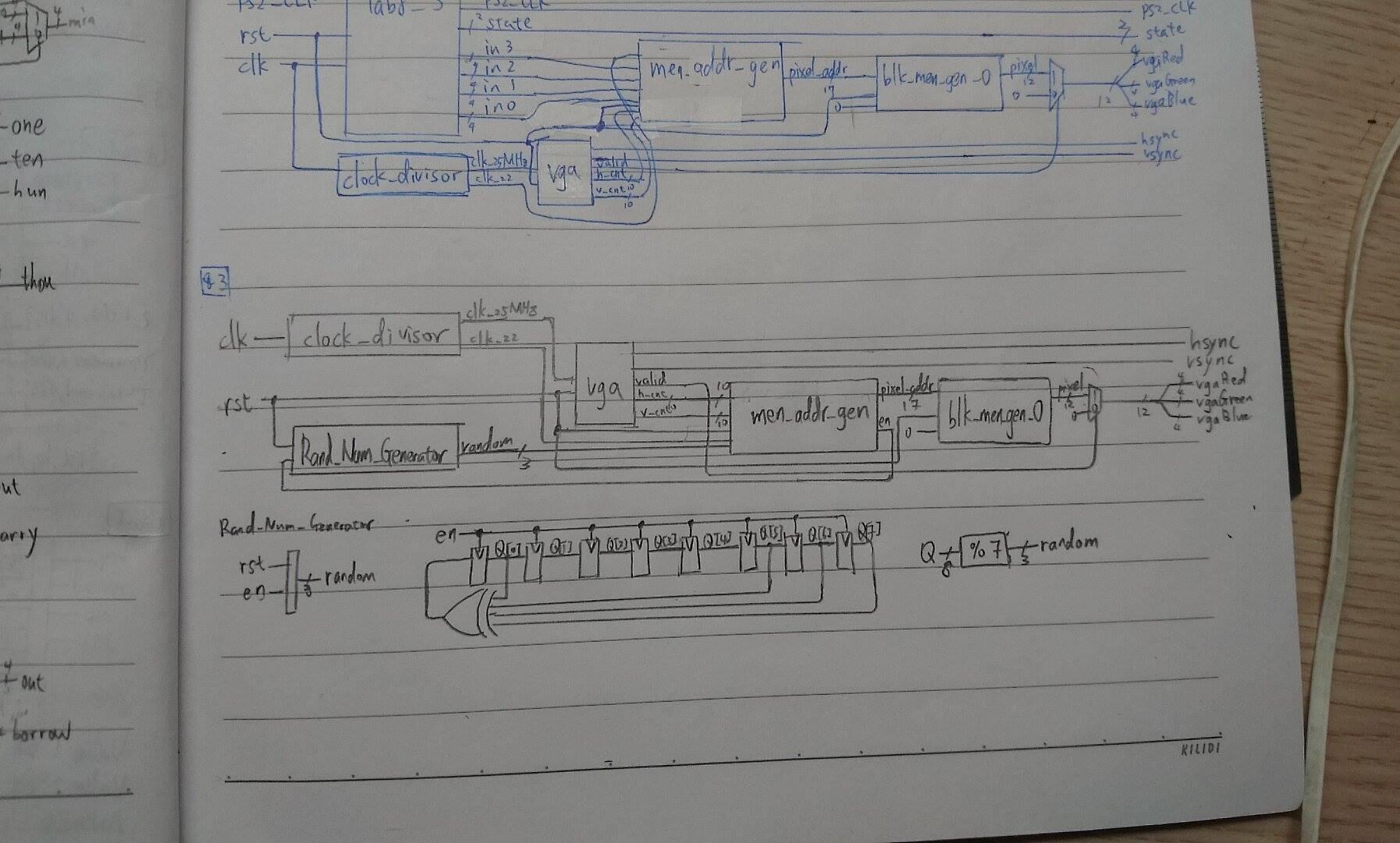
Inout：PS2\_DATA,PS2\_CLK; //V17控制rst，當rst==0時會產生rst的訊號

Design Implementation

Logic function :

1. top：

此module，專門拿來呼叫其他小moudule的。



1. Clock\_divisor：

此module是用來產生clk，分別產生clk22(約24hz)以及clk\_25MHz。

1. vga\_controller：

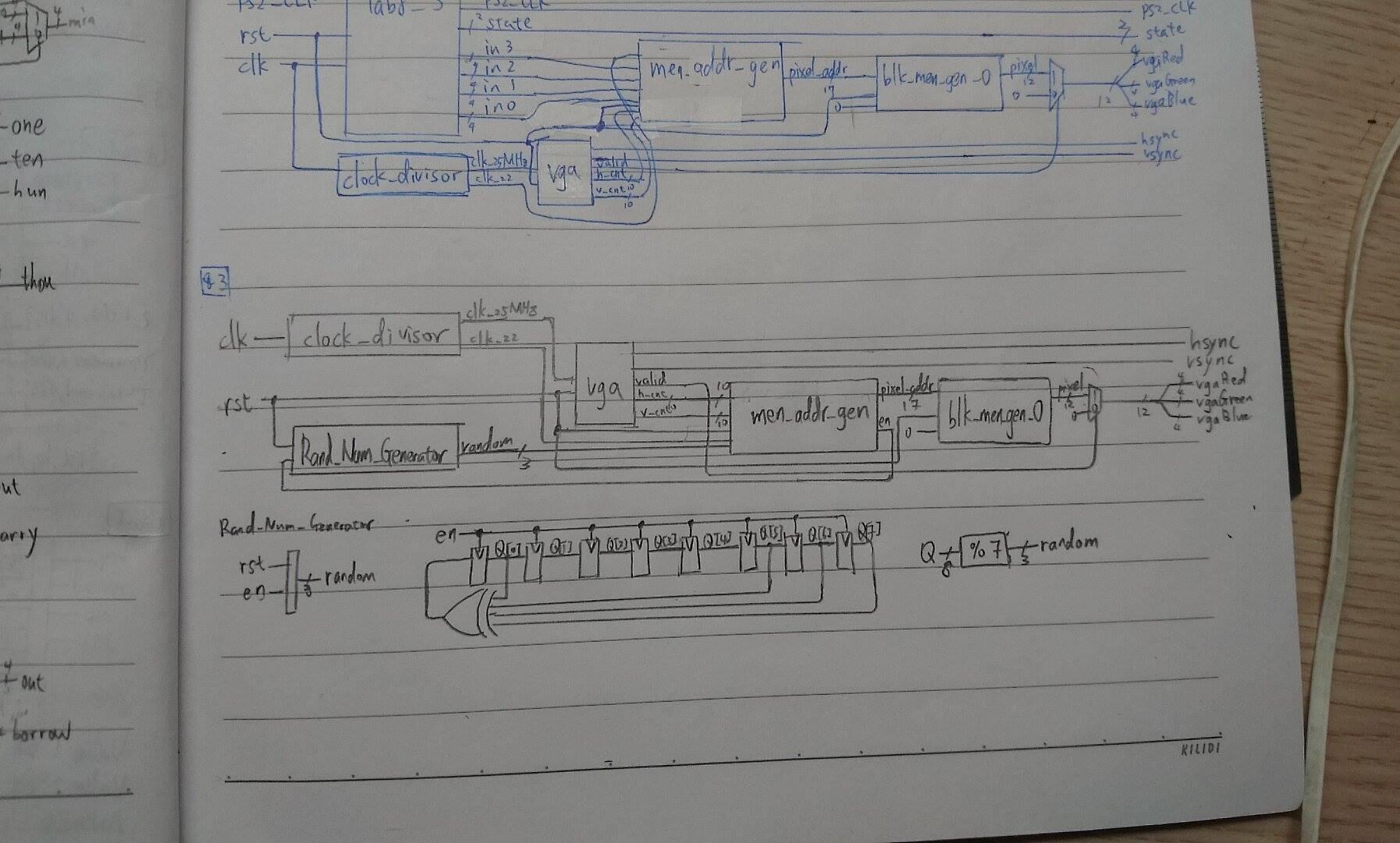
此module是用來更新畫面的，產生的h\_cnt和v\_cnt可以用來讀圖片。

1. mem\_addr\_gen：

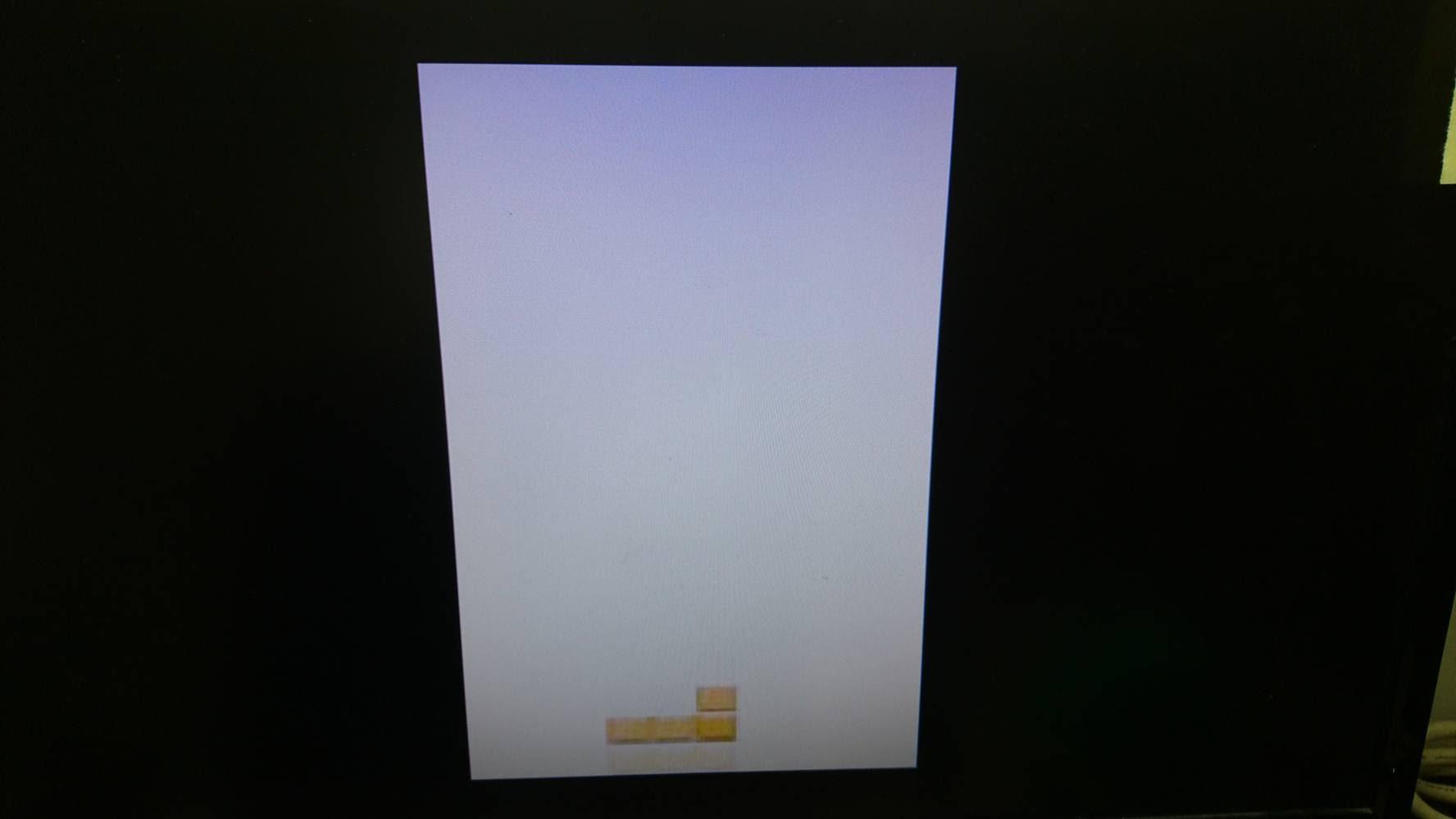
此module是用來讀圖片檔的，並從Rand\_Num\_Generator接出random訊號，作為決定顯示哪個方塊的依據，且這裡讀圖的方法是一排方格一排方格一起顯示。

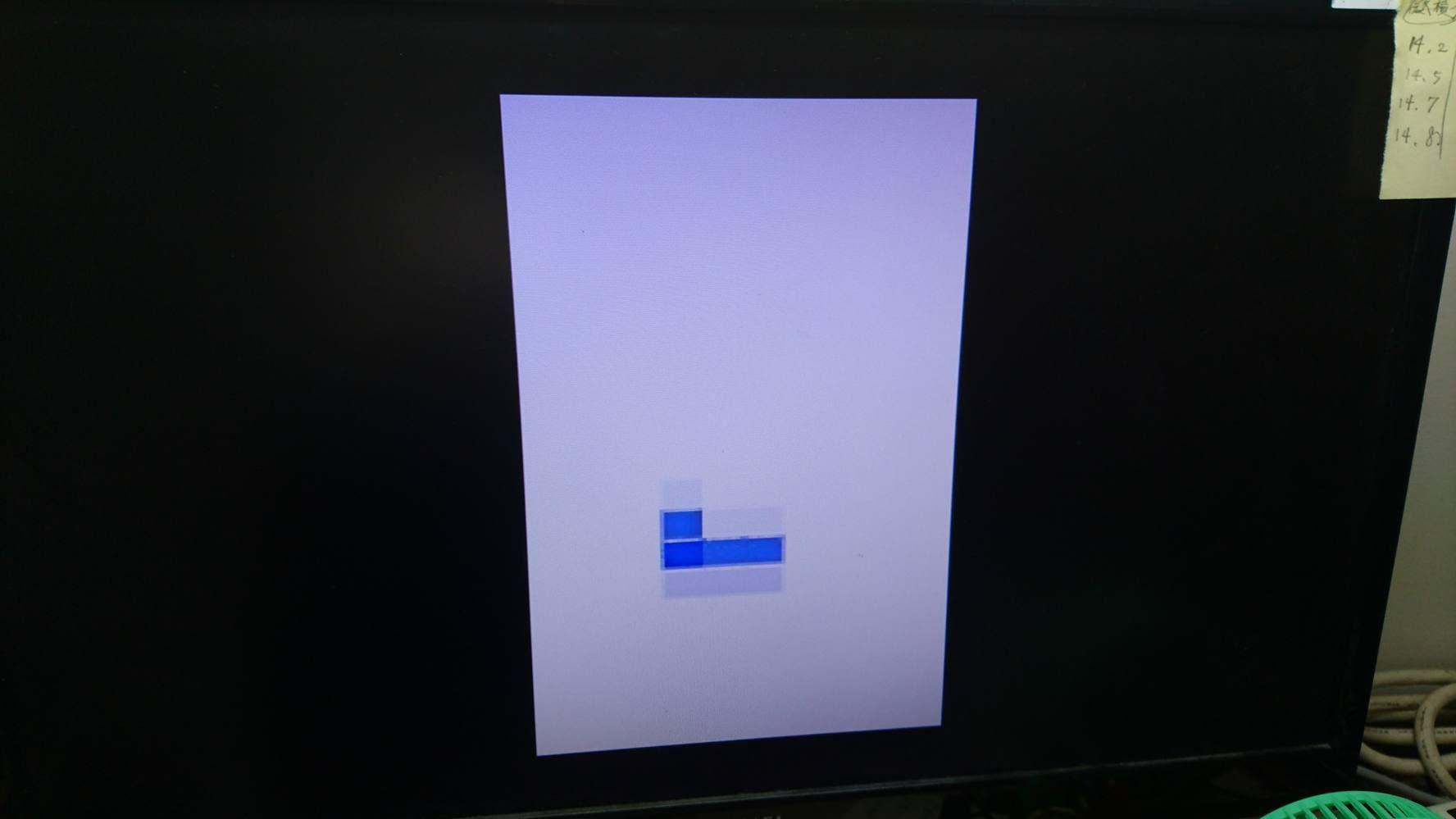
1. Rand\_Num\_Generator：

此module是用來產生亂數。



Result





Discussion

1. decoder

這設計以key\_down[9’b0\_0101\_1000]作為給caps值的DFF的clk。這樣可以使當按下”caps”鍵時就會進行存值。

1. 由於大寫與小寫字母的ASCII code都剛好相差8’d32(小寫>大寫)，因此在做給

ASCII code值時可以運用這個特性，不用把每個字母大小寫的ASCII code都解碼。

1. 設計men\_gen\_addr時，從Rand\_Num\_Generator接出random訊號，作為決定顯示哪個方塊的依據，且這裡讀圖的方法是一排方格一排方格一起顯示。
2. 設計men\_gen\_addr時，還設定邊界條件，因此並不會掉落到最底才會產生新的方塊掉落。

**Conclusion**：

這次的lab做的真的頗挫折的，光是在顯示螢幕上就下了不少工夫，真的很感謝一路上幫助我的人。