

Lab11 report

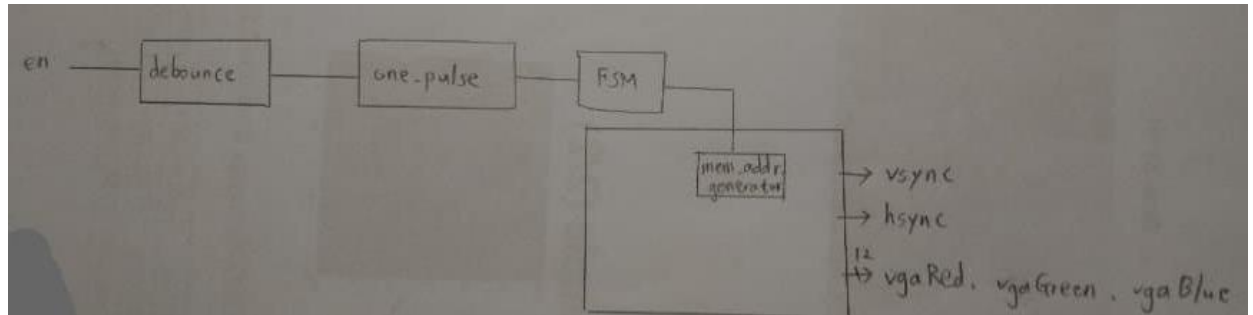
107061218 謝霖泳

1.

➤ Design Specification

Input: clk, rst, en

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



➤ Design Implementation

這題並不難，只要將所提供的 demo2 中，多加上一個按鈕的 input，名為 en，用來控制要不要捲動圖片，因此，我還需要 debounce 以及 one pulse 和 FSM 的 module，並將 FSM 的 output 傳入 memory address generator 的 module 中，判斷 position 是否要加 1'b1，就大功告成了。

IO pin assignment:

vgaBlue[3]	vgaBlue[2]	vgaBlue[1]	vgaBlue[0]
J18	K18	L18	N18

vgaGreen[3]	vgaGreen[2]	vgaGreen[1]	vgaGreen[0]
D17	G17	H17	J17

vgaRed[3]	vgaRed[2]	vgaRed[1]	vgaRed[0]
N19	J19	H19	G19

clk	rst	en	hsync	vsync
W5	U17	T18	P19	R19

➤ Discussion

這一題比較簡單，並沒有花很多時間便完成了。

2.

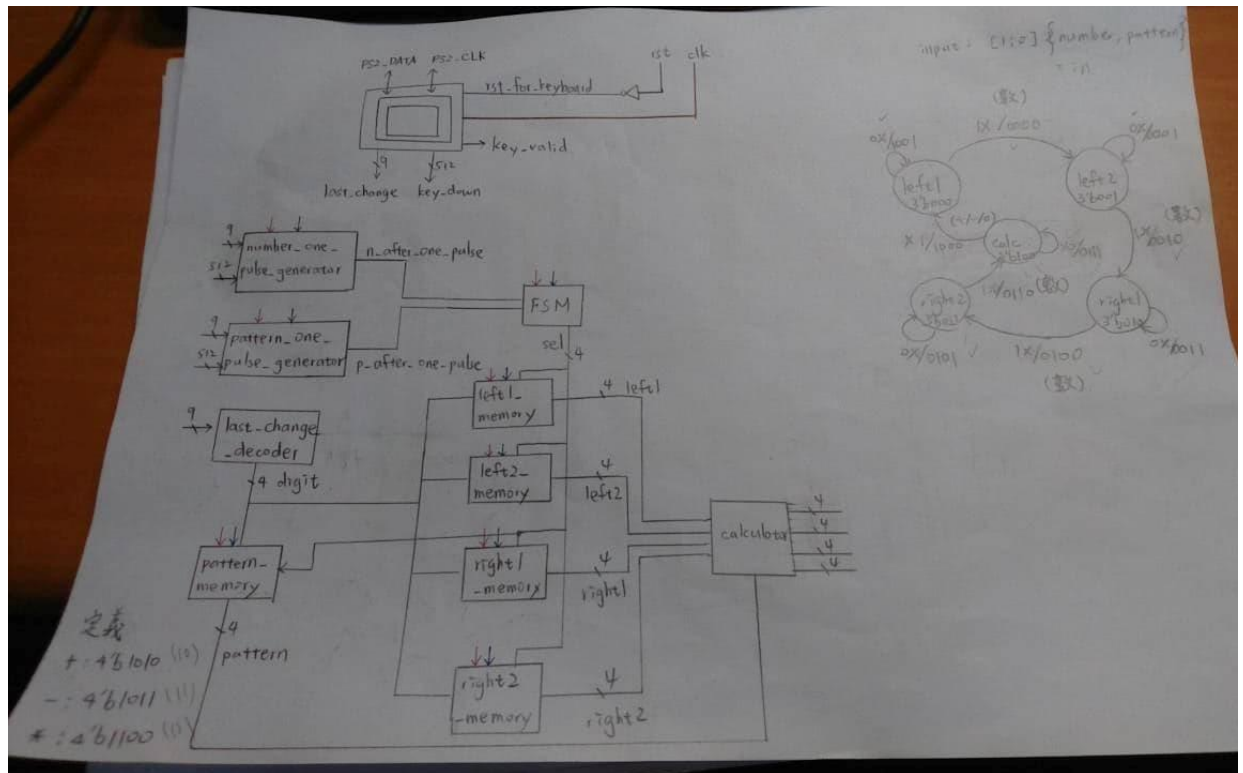
➤ Design Specification

Inout: PS2_CLK, PS2_DATA

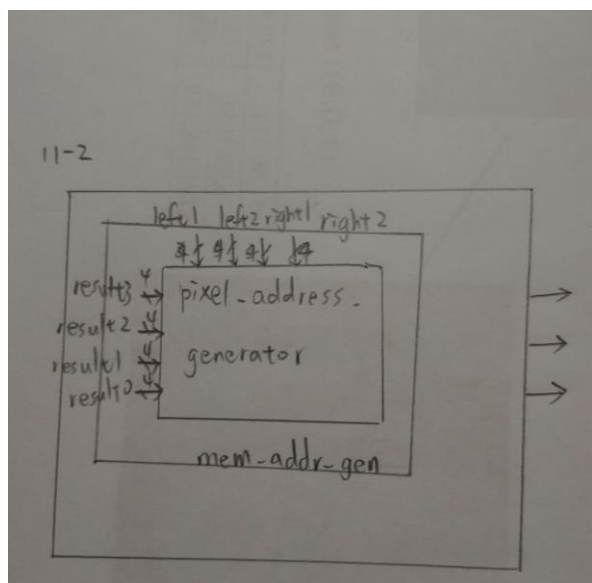
Input: clk, rst

Output: hsync, vsync, [3:0] ssd_ctrl, [7:0] show,
[3:0] vgaRed, [3:0] vgaGreen, [3:0] vgaBlue

Lab9-3:



Vga:



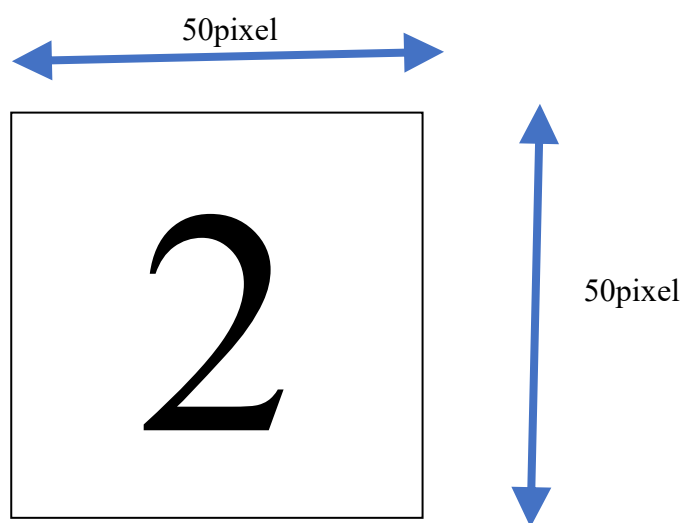
➤ Design Implementation

為了避免不必要的錯亂，這題我以之前的 keyboard calculator 為基底，因此，該 lab 有的功能將不會消失，比如七段顯示器，不一樣的是將運算前的四個位數與運算結果的四個位數額外拉出來，進入一個名為 pixel address generator 的 module 之中。

pixel address generator 是被包在 memory address generator 中的一個 module，功能為以不同位置的數字去選擇不同的 memory address 的輸出。於此之前，我用小畫家畫了一張數字 0~9 和 +-= 的一張.jpg 檔（如下圖），而每個字元皆是 50*50 的大小，因此，以加法為例，當我的 v_cnt 與 h_cnt 都 < 50 時，代表我現在螢幕掃到的位置是左上角那一格，應該顯示被加數的十位數，因此，以被加數的十位數作為選擇，去選擇我要顯示.jpg 檔中哪一個區塊，比如說，如果現在被加數的十位數=3，那麼，我就應該顯示.jpg 檔中的"3"，而 3 在圖中的位置就是 $h_cnt + 50 * v_cnt + 7500$ ，依此類推。

0
1
2
3
4
5
6
7
8
9
+
-
*
=

以數字 2 為例：



IO pin assignment:

vgaBlue[3]	vgaBlue[2]	vgaBlue[1]	vgaBlue[0]
J18	K18	L18	N18

vgaGreen[3]	vgaGreen[2]	vgaGreen[1]	vgaGreen[0]
D17	G17	H17	J17

vgaRed[3]	vgaRed[2]	vgaRed[1]	vgaRed[0]
N19	J19	H19	G19

show[7]	show[6]	show[5]	show[4]	show[3]	show[2]	show[1]	show[0]
W7	W6	U8	V8	U5	V5	U7	V7

ssd_ctrl[3]	ssd_ctrl[2]	ssd_ctrl[1]	ssd_ctrl[0]
W4	V4	U4	U2

clk	rst	hsync	vsync	PS2_CLK	PS2_DATA
W5	U17	P19	R19	C17	B17

➤ Discussion

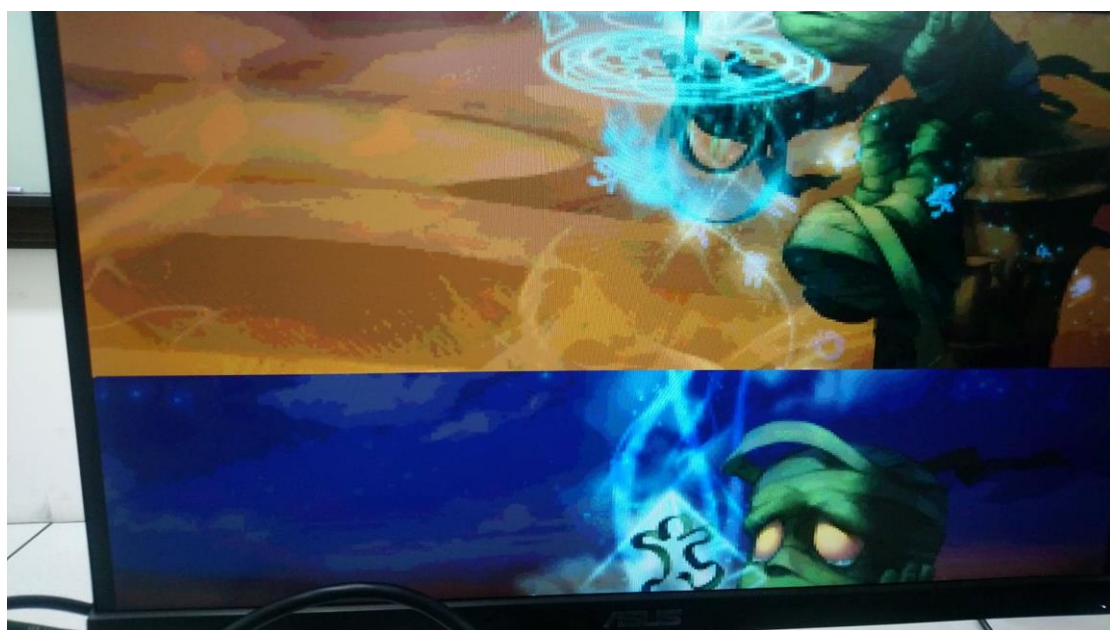
這題花費了我不少時間，主盡是因為我一開始想不到怎麼在對應區域顯示出對應的數字，後來才使用這個方法，就是把所有數字畫在同一張圖之中，並依照我運算的數字結果顯示相對應之區域即可。

3. conclusion

這次 lab 一開始還卡在一個窘境，就是我不知道要如何在小畫家中，將每個數字都設定成一樣大小，還好後來上網查有查到方法。

此外，我有發現在所提供之 vga_controller 的 module 之中，真正掃到的 pixel_cnt 會刻意慢兩個 clk，所以，在本 lab 的第一題當中，如果粉仔細觀察螢幕，便會發現圖片捲動時，左下角其實有一段不平整區，大概佔兩個 pixel 的大小，推測其可能原因就是因為這兩個 clk 的延遲。

乍看之下並無異狀：



仔細看就會發現有鬼：



這次的 lab 是這學期最後一次，謝謝教授這學期辛苦的教導和助教們認真地幫我們 demo 和批改結報。說實話，邏輯設計實驗應該是大一下我收穫最豐富的課程，也是我花費第二多時間的課程（花最多時間是微積分，很難 qq），總而言之，老師和助教都辛苦了！