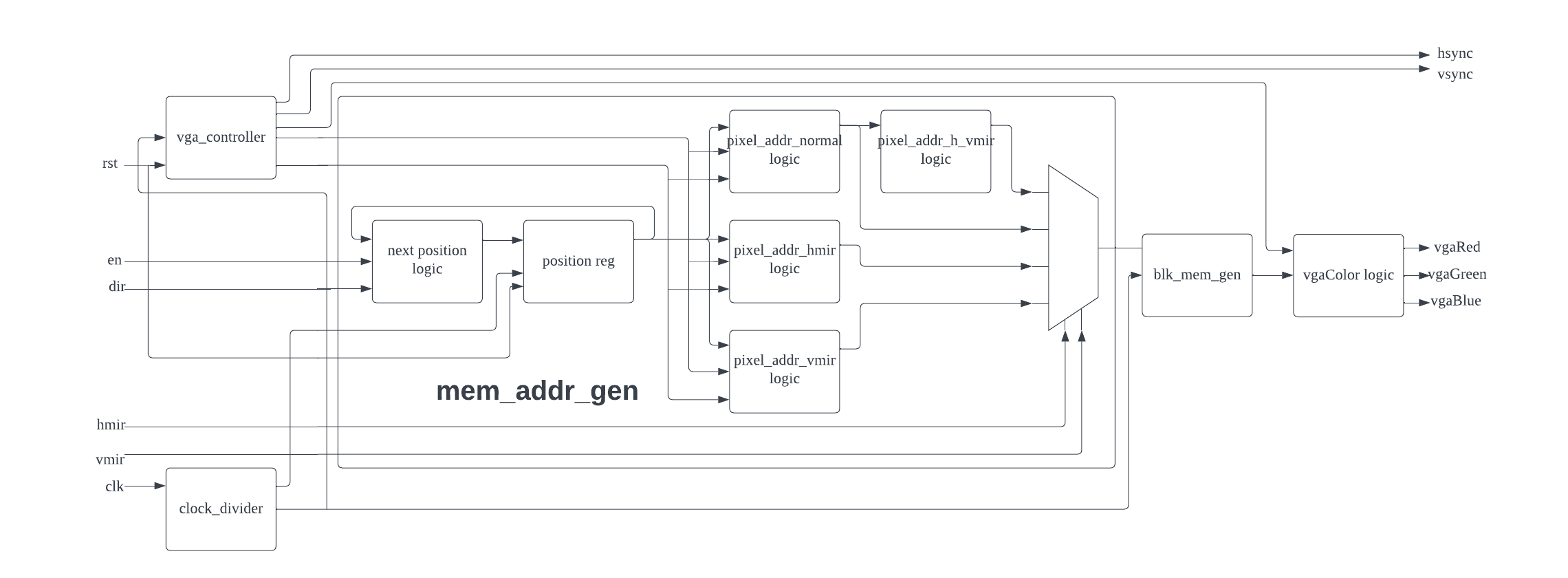
|  |  |
| --- | --- |
| **Lab 7** | |
| 學號: 110062131 | 姓名: 馬毓昇 |

1. **Lab Implementation**
   * + 1. Lab7\_1: Block diagram of the design with explanation



* + 1. Lab7\_1 module裡包含了vga\_controller、blk­\_mem\_gen、mem\_addr\_gen三個主要module，其中mem\_addr\_gen是主要邏輯的地方，能透過數學運算出v\_cnt, h\_cnt對應到要顯示的pixel的address為何。
    2. 因為要設計滾動，所以在mem\_addr\_gen裡面有一個position reg當作offset，最後用mux來決定要輸出哪種行為。
       1. Lab7\_1: Partial code screenshot with the explanation: you don’t need to paste the entire code into the report. Just explain the kernel part.

a. Lab7\_1 module裡面都跟demo的template長一樣，重點是在mem\_addr\_gen裡面。

一張含有 文字 的圖片

自動產生的描述一張含有 桌 的圖片

自動產生的描述

b. Position: 當en被按的時候更新position，不然position就不變。如果dir沒被按，position就+1形成往上滾動的效果，不然position就-1形成往下滾動的效果。額外有用if做邊界判斷，如果超過0~239就要從重置position。

一張含有 文字 的圖片

自動產生的描述

c. Pixel\_addr：用一堆三元運算子來判斷要輸出哪個狀態，HV都mirror、H mirror、V mirror或是正常(normal)。

Pixel\_addr\_normal: 把h\_cnt/2 + v\_cnt/2\*320相當於把2的h\_cnt/2, v\_cnt/2座標轉換成一維的address(h\_cnt/2橫向每多1代表要多1 pixel，v\_cnt/2直向每多1代表要多一橫行也就是320 pixels)，再加上position\*320(position代表直向offset，每多1就要多一橫行也就是320 pixels)，最後模除76800(640\*480共76800 pixels，可以取得圖上的相對address)，而因為position會一直更新，這樣就可以達到轉動的效果。

Pixel\_addr\_h\_vmir: 用76799減去pixel\_addr\_normal，就可以讓整張圖pixel順序reverse，達成同時上下左右翻轉。

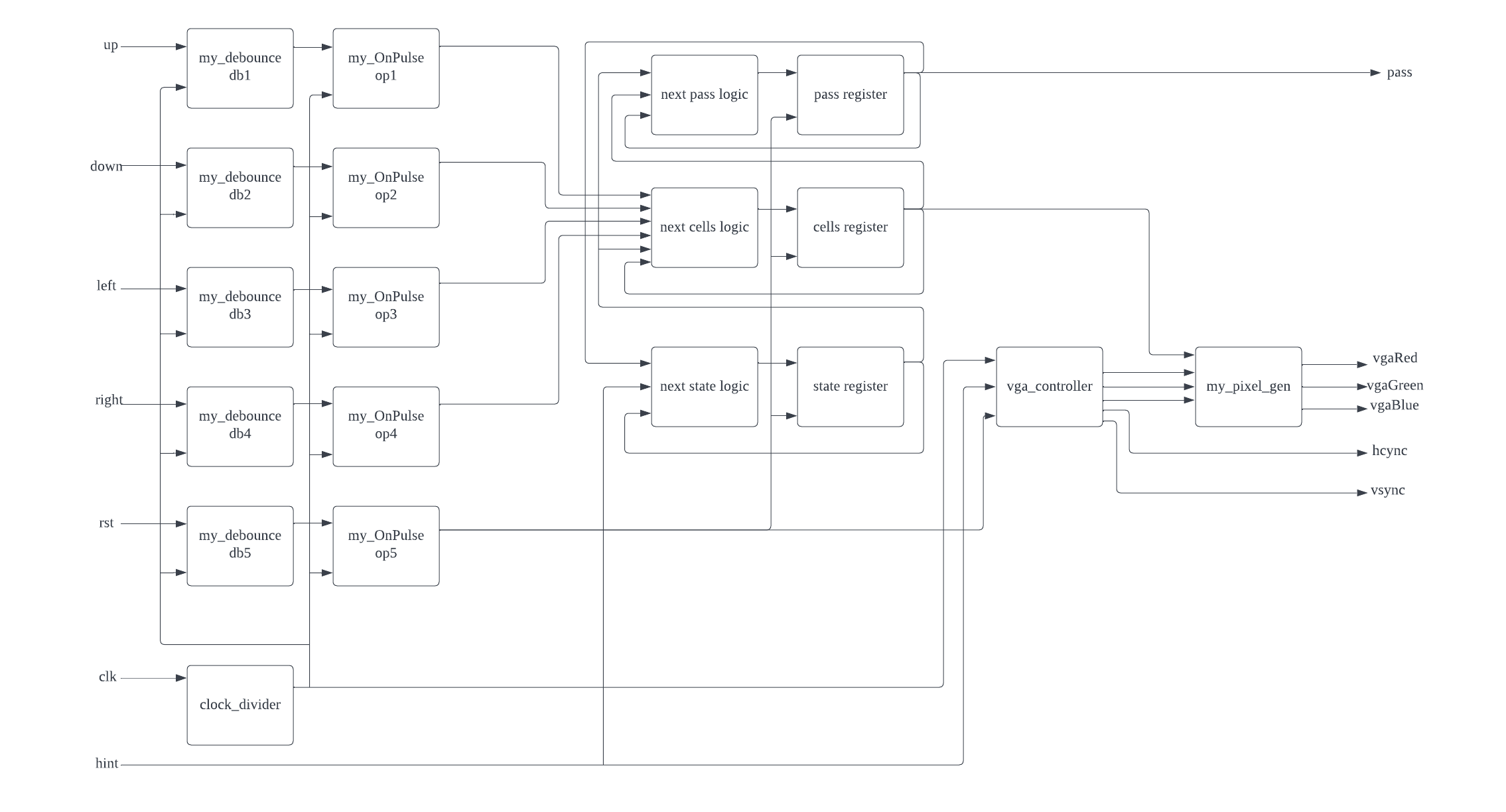
Pixel\_addr\_hmir: 用319減去h\_cnt/2就可以讓每一橫行的pixel順序reverse過來，達成左右翻轉。

Pixel\_addr\_vmir: 用239減去v\_cnt/2就可以讓每一直行的pixel順序reverse過來，比較不一樣的後面要減position而不是加，這樣圖片才能按照原本的滾動方向滾動。而因為是減，所以有可能會小減大然後溢位，模除就會有問題，因此在模除前要加76800來避免。

一張含有 文字 的圖片

自動產生的描述

1. Lab7\_2: Block diagram of the design with explanation



a. up、down、left、right、rst有用debounce跟onepulse處理

b. FSM部分有state作狀態變化，pass判斷結束了沒，cells為記錄背後的puzzle(0~15)的register

c. vga\_controller控制vga，my\_pixel\_gen接受vga\_controller的v\_cnt跟h\_cnt還有cells的訊號來判斷要輸出哪個addr的pixel。

2. Lab7\_2: Partial code screenshot with the explanation: you don’t need to paste the entire code into the report. Just explain the kernel part

a. state： 本題我有用state來作，分別作了INIT、GAME、HINT、PASS四個state。當hint被按起時，state會從GAME跳到HINT，反之則會從HINT跳回GAME。如果pass了就進到PASS。

一張含有 文字 的圖片

自動產生的描述一張含有 文字 的圖片

自動產生的描述

b. puzzle邏輯：如果up, down, left, right被按起，就找出cell是0的位置，並判斷反方向是不是valid的cell，如果是就兩者交換。

一張含有 文字 的圖片

自動產生的描述

c. pass: 我有做一個pass\_cells來表示答案，過程中就一直檢查cells跟pass\_cells有沒有完全一樣，有的話pass就會變1。

一張含有 文字 的圖片

自動產生的描述

一張含有 文字 的圖片

自動產生的描述

c. my\_pixel\_gen： 核心module，會依據h\_cnt、v\_cnt、cells的模樣的輸出pixel。一張含有 文字 的圖片

自動產生的描述

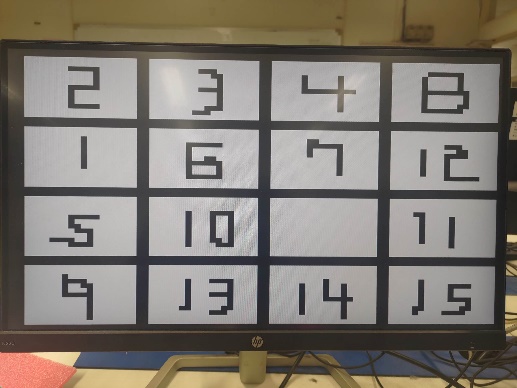
在這個module裡，我有放圖片的bitmap，大小是80\*60=4800 bits，然後還有一個reg標記出”每個cell數字對應到的小bitmap”的index起始點是多少，小bitmap就是把大bitmap切成4x4每個是20\*15 pixels。

一張含有 桌 的圖片

自動產生的描述一張含有 文字 的圖片

自動產生的描述

一張含有 文字, 電子用品, 鍵盤, 美工圖案 的圖片

自動產生的描述

至於我算出v\_cnt、h\_cnt要輸出的pixel的方式是：

1. 先用nested if-else statement判斷出這個v\_cnt、h\_cnt對應到的是cell[x : x+3] 現在是甚麼值(value: 0~15)

2. position[ value ]取出這個值對應到bitmap上的起點

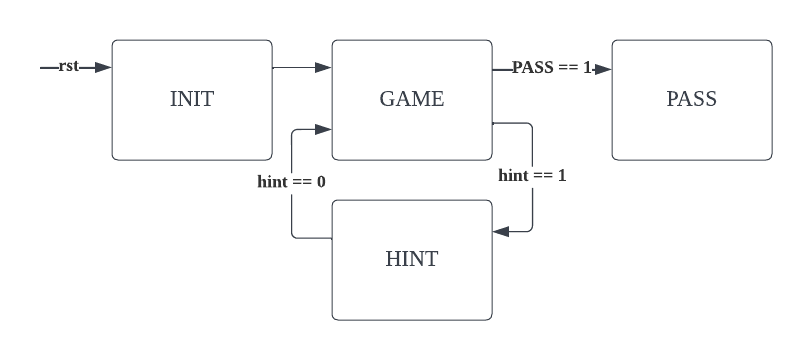
3. ((v\_cnt>>3)%15)\*80 + (h\_cnt>>3)%20 取出目前v\_cnt、h\_cnt相對於小bitmap起點的相對address差

4. 兩者相加：position[cells[0:3]] + ((v\_cnt>>3)%15)\*80 + (h\_cnt>>3)%20 即是pixel address

color[address] 便是 pixel

一張含有 桌 的圖片

自動產生的描述

 3. lab7\_2 FSM:

1. -> INIT: rst被按

2. INTI -> GAME: 進INIT下個cycle就進GAME

3. GAME -> HINT: hint被按

4. INIT->GAME: hint沒被按

5. GAME->PASS: 贏了

1. **Questions and Discussions**

A: wire pixel\_addr = ((h\_cnt>>1+position)%320 + 320\*(v\_cnt>>1))，讓向左捲動時，保持position遞增，向右捲動時，則保持position遞減，position在0~319之間更新。(h\_cnt>>1 + position)可以讓pixel水平移動，模除320則是找出pixel在這橫行的正確相對位置，最後在正常的補上320\*(v\_cnt>>1)，這樣一來就可以達成左右捲動的效果。

B: 可以用 reg [3:0] puzzle [0:3][0:3] 的方式來做出存puzzle狀態的4x4大小的register(0~15)。

C:

1. 如果圖片的顏色很少的話(黑白遊戲)，可以用Run-length encoding algorithm來存圖以節省空間。例如：有一黑白顏色列0001100000110000，0代表白1代表黑，可以不用{12'hFFF, 12'hFFF, …}來存，改用{4'd3, 4'd2, 4'd5, 4'd2, 4'd4}來存，就可以把12\*16=192個bits降到20個bits。

2. 如果是要做遊戲背景的話，可以選用幾何圖形重複的方式來製作，這樣的話，只需要儲存一個圖形的模樣，然後再依據h\_vnt、v\_cnt算出要輸出哪個pixel的address就好。假設一個640x480的畫面被分割成4x4個圖案，每一個圖案就只佔160x120，那麼實際上就只需要存19200\*12=230400 bits而不用到3686400 bits，便可以用更小的空間輸出同樣解析度的圖片。

1. **Problem Encountered**

1. 第一個問題是我在做lab7\_1時遇到的，就是我在弄上下顛倒的時候要減position才能達到遞增時向上滾動、遞減時向下滾動。但只減的話，就會觀察到說滾動的時候會破圖。想了想才發覺是因為負數取模會壞掉，解決辦法就是在取模前加回來就可以了。



2. 第二個問題是我在做lab7\_2時遇到的，就是怎麼算都算不好數字對應到的圖片address要怎麼取，後來發現問題是出在我定義每個數字所對應的圖片的最左上那點座標找錯了，最後就用reg暴力記左上座標，直接省去運算。

一張含有 文字 的圖片

自動產生的描述

3. systemverilog不支援 cell[i : i+3]兩邊有一邊不是constant的用法，查了才知道要改成 cell [i +: 4]，學到一個新語法。

1. **Suggestions**

這次的lab實作出來的15puzzle遊戲感覺完成度比上次的1A2B還要高，說實在我很喜歡作這類型很像遊戲的lab，希望之後可以多一點~