**LAB11.1** 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.

1.Design specification:

功能: 按奇數次按鈕，圖片開始向上捲動，偶數次則圖片停止捲動。

輸入: clk (石英震盪器輸入)

rst (重置全部模組)

pb\_en (開始/停止向下捲動)

輸出: [3:0] vgaRed (紅色輸出)

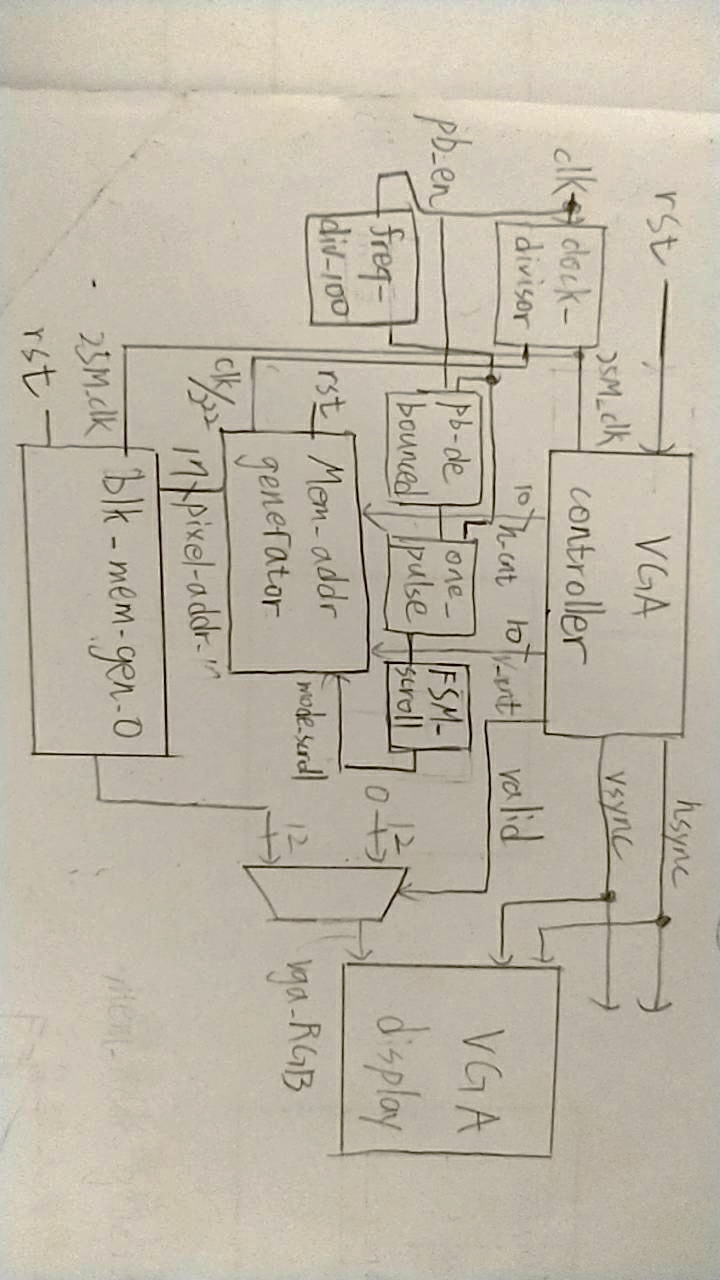
[3:0] vgaGreen (綠色輸出)

[3:0] vgaBlue (藍色輸出)

hsync (水平重置)

vsync (垂直重置)

2. Design implementation



首先，VGA\_controller的功用是產生hsync及vsync訊號，給予螢幕從哪裡開始數像素的參考點，並且輸出h\_cnt及v\_cnt表示現在正輸出哪個像素。h\_cnt及v\_cnt會被輸進mem\_addr\_generator，產生出這個像素對應的儲存圖片像素地址pixel\_addr。此外，為了判斷目前按下奇數次還是偶數次按鈕，所以使用FSM\_scroll，按第一次按紐時，按鈕訊號被處理之後，其輸出值會從零變成一，再按一次時會從一變成零，藉以知道現在的狀態mode\_scroll，且同樣輸到mem\_addr\_generator裡面。

blk\_mem\_gen\_0是一塊ROM，內存要顯示在銀幕上的圖片，當像素地址pixel\_addr被傳下來後，ROM會對應輸出應顯示的RGB三原色比例。之後再被送到顯示器前，為了確認現在數到的地方在螢幕裡，所以會有一個多工器控制，當在螢幕邊界裡時valid為一，多工器放行三原色比例，不然全部輸出零。

mem\_addr\_generator的構造很單純，pixel\_addr 是由((h\_cnt >> 1) + 320 \* (v\_cnt >> 1) + position \* 320 ) % 76800算出來的，h\_cnt及v\_cnt要向右移一位是因為儲存圖片解析度為320 \* 240，為螢幕解析度的一半(不存640 \* 480的圖片是因為basys板記憶體空間不足)。position是暫存器，其接clk\_22，在正緣時，如果FSM\_scroll為一，在position到239前會一直往上加一，到239時會歸零。FSM\_scroll為零時，position的值則不動。

3. I/O pin assignment

|  |  |  |
| --- | --- | --- |
| I/O變數 | 對應腳位 | 補充說明 |
| 輸入 | | |
| clk | W5 | 石英震盪器輸入 |
| rst | U18 | 重置模組 |
| pb\_en | U17 | 開始/停止向下捲動 |
| 輸出 | | |
| hsync | P19 |  |
| vsync | R19 |  |
| vgaBlue | 參考.xdc檔 |  |
| vgaGreen | 參考.xdc檔 |  |
| vgaRed | 參考.xdc檔 |  |

**LAB11.2** Combine the key board controller and VGA displaying controller to design a calculator with 2-digit addition/subtraction/multiplication.

1.Design specification:

功能: 二位元計算機，包含加法、減法、乘法

使用方式:扳起最右邊DIP後，先透過小鍵盤輸入一個二位數，在按下運算 符號，輸入第二個二位數，輸出同時輸出結果。

輸入: PS2\_DATA (鍵盤訊號輸入)

PS2\_CLK (鍵盤訊號輸入)

clk\_100mhz (石英震盪器輸入)

rst\_n (重置全部模組)

輸出: [3:0] vgaRed (紅色輸出)

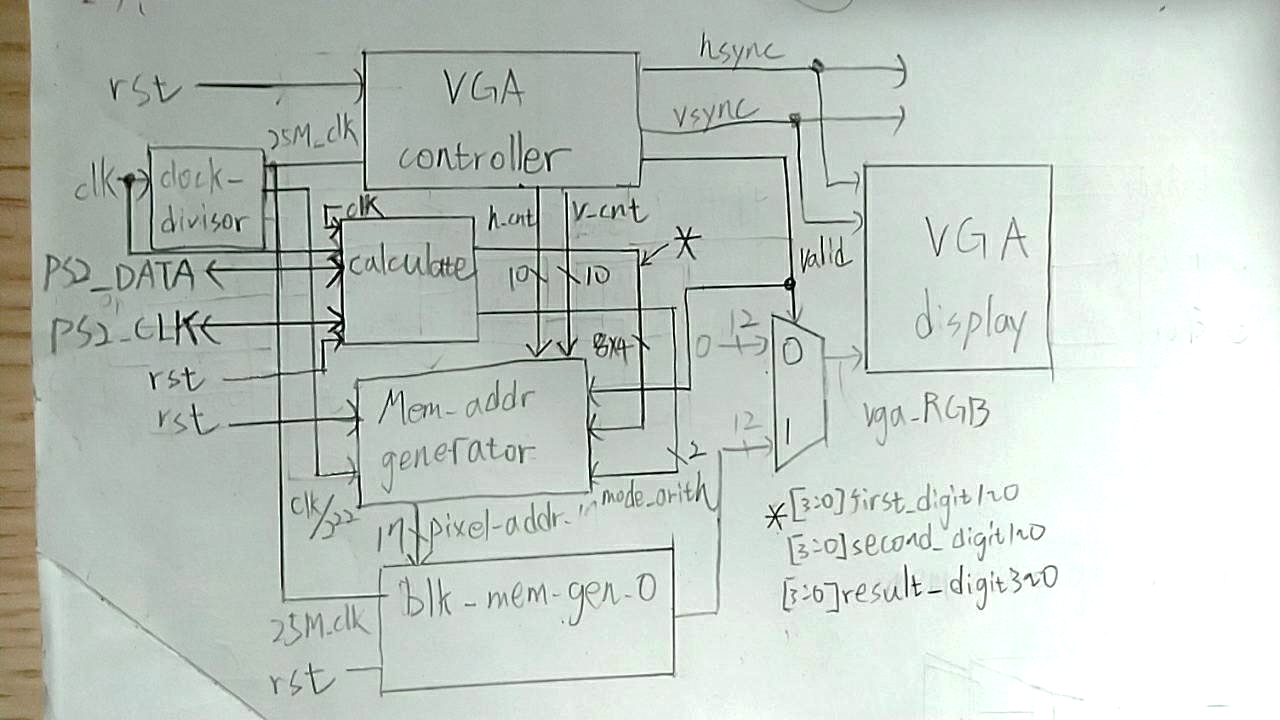
[3:0] vgaGreen (綠色輸出)

[3:0] vgaBlue (藍色輸出)

hsync (水平重置)

vsync (垂直重置)

2. Design implementation

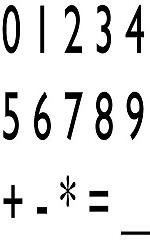


模組的功用都和上題都提到的都一樣，只是多了一個calculate模組。Calculate的功用是將鍵盤輸入的數值及運算符號儲存起來，並且運算結果。詳情請見LAB9.2。之後呢，這些訊息會被輸到mem\_addr\_generator，當然其構造被改動過，來應變不同的輸入，不過功能仍是輸出預存圖片的像素地址。

Mem\_addr\_generator:

在這題中，先將輸入的h\_cnt及v\_cnt先都除以二變h\_cnt\_small及v\_cnt\_small。然後根據處理後的h\_cnt及v\_cnt位置來判斷現在要讀取進來的數字。在螢幕上，我在上方一300 \* 60的長方形分割了十個小區域，當h\_cnt及v\_cnt座標指到該區域時，選擇現在應該要輸出的數字(ex.h\_cnt\_small和v\_cnt\_small皆為20時，則選擇一)。而這個該輸出的數字會對應預存圖片ROM中該數字的起始像素位置(如8的位置在90 \* 86)。那要如何知道現在要輸出預存圖片的哪個像素呢。公式是像素地址 = (v\_cnt\_small + 該數字起始垂直座標 – 螢幕長方形起始垂直座標) \* 150 + (該數字起始水平座標 + (h\_cnt\_small – 螢幕長方形起始水平座標) %長方形分割後小長方形水平長度)。

Blk\_mem\_gen\_0中預存的圖



3. I/O pin assignment

|  |  |  |
| --- | --- | --- |
| I/O變數 | 對應腳位 | 補充說明 |
| 輸入 | | |
| clk\_100MHz | W5 | 石英震盪器輸入 |
| rst | U17 | 重置模組 |
| PS2\_CLK | C17 |  |
| PS2\_DATA | B17 |  |
| 輸出 | | |
| hsync | P19 |  |
| vsync | R19 |  |
| vgaBlue | 參考.xdc檔 |  |
| vgaGreen | 參考.xdc檔 |  |
| vgaRed | 參考.xdc檔 |  |

**討論與心得:**

這一次的lab很有難度，第三題的bonus礙於時間壓力，選擇沒打。不過還好教授有先給模板，讓我們實際了解要如何輸出圖片到螢幕上，省去了不少需要自行摸索的部分。