

Lab 7

學號: 109062202

姓名: 陳禹辰

1. 實作過程

這次 lab 分為兩個部分，lab7_1 是要讓圖片上下移動跟變色、lab7_2 則是要讓圖片分區旋轉的小遊戲。

LAB7_1：

因為是第一次實作螢幕輸出的東西，所以一開始花了蠻多時間看懂 demo1, 2 是怎麼運作的，不過看懂之後就比較簡單了，因為 lab7_1 只要讓圖片上下移動跟變色，所以我是直接拿 demo2 讓他移動的算式來，然後更改 position 就可以了，如果 position 增加就會向上，反之就會向下。

因為 pixel_addr 是把每次要運算的 h_cnt 跟 v_cnt 進行運算，然後算出他會用一個一維陣列中存的哪一個當作要輸出在這個 h_cnt 跟 v_cnt 的顏色。

```
assign pixel_addr = ((h_cnt>>1)+320*(v_cnt>>1)+ position*320)% 76800; //640*480 --> 320*240

always @(posedge clk or posedge rst) begin
    if(rst)begin
        position <= 0;
    end else begin
        if (en) begin
            if (!dir) begin
                if(position < 239) position <= position + 1;
                else position <= 0;
            end else begin
                if(position > 0) position <= position - 1;
                else position <= 239;
            end
        end else begin
            position <= position;
        end
    end
end
```

知道要怎麼讓圖片上下移動之後剩下的就比較簡單了，只要根據要求的判斷來決定什麼時候向上移動，什麼時候向下移動或者不動就好。我讓他判斷在只有 en 是 1 的時候會更改 position 的值，所以 en = 0 時就不會動了。

至於變色的部分，因為是要變成 complementary color 所以我就是將輸出後的顏色反過來，也就是在前面加個~就可以了。

```
always @(*) begin
    if (valid) begin
        if (nf == 0) begin
            {vgaRed, vgaGreen, vgaBlue} = pixel;
        end else begin
            {vgaRed, vgaGreen, vgaBlue} = ~pixel;
        end
    end else begin
        {vgaRed, vgaGreen, vgaBlue} = 12'h0;
    end
end
```

LAB7_2 :

這個部分也是只要搞定分區旋轉的部分剩下的就是之前做過的東西了。

要做到分區旋轉的話，我也是從 pixel_addr 上面做，就是要以讀進來的 h_cnt 跟 v_cnt 算出原本他所在的位址所該用的 pixel_addr，我是先將圖片分成 12 區，水平分成四個垂直分成三個，然後要運算的話，以旋轉一次 90 度為例，就是先把 h_cnt 跟 v_cnt 扣掉 160*h 跟 160*v 讓他變成那一個正方形的第幾個然後 h 跟 v 換位子就是旋轉，然後再把 160*h 跟 160*v 加回去變成他所在的那一格。

```
always @(*) begin
    case(rotate_times)
        0: pixel_addr = ((h_cnt>>1)+320*(v_cnt>>1))% 76800;
        1: pixel_addr = (((v_cnt - 160 * pv) + 160 * ph) >> 1) + 320 * ((159 - (h_cnt - 160 * ph) + 160 * pv) >> 1);
        2: pixel_addr = ((159 - (h_cnt - 160 * ph) + 160 * ph) >> 1) + 320 * ((159 - (v_cnt - 160 * pv) + 160 * pv) >> 1);
        3: pixel_addr = ((159 - (v_cnt - 160 * pv) + 160 * ph) >> 1) + 320 * (((h_cnt - 160 * ph) + 160 * pv) >> 1);
    endcase
end
```

然後要讓鍵盤對應到各個區塊，所以我就設了 12 個變數紀錄各個區塊所旋轉的次數，然後在按鍵盤的時候去對應旋轉幾次所要使用的 pixel_addr 算式。

```
if (been_ready && key_down[last_change] == 1'b1) begin
  case (last_change)
    KEY_CODES[00] :begin...
    end
    KEY_CODES[01] :begin...
    end
    KEY_CODES[02] :begin...
    end
    KEY_CODES[03] :begin...
    end
    KEY_CODES[04] :begin...
    end
    KEY_CODES[05] :begin...
    end
    KEY_CODES[06] :begin...
    end
    KEY_CODES[07] :begin...
    end
    KEY_CODES[08] :begin...
    end
    KEY_CODES[09] :begin...
    end
    KEY_CODES[10] :begin...
    end
    KEY_CODES[11] :begin...
    end
  endcase
end
```

然後要讓他全部轉成正確時停下來我是根據 12 個的選轉次數去判斷，如果全部都是旋轉 0 次，就讓他按鍵盤沒反應，然後顯示 pass = 1。

```
assign pass = (rotate_times_1 == 0 && rotate_times_2 == 0 && rotate_times_3 == 0
  && rotate_times_4 == 0 && rotate_times_5 == 0 && rotate_times_6 == 0
  && rotate_times_7 == 0 && rotate_times_8 == 0 && rotate_times_9 == 0
  && rotate_times_10 == 0 && rotate_times_11 == 0 && rotate_times_12 == 0 ) ? 1'b1 : 1'b0;
```

2. 到的東西與遇到的困難

這次的東西其實就是多一個螢幕的東西要特別想一下要怎麼用，其他部分其實就是之前學到的東西，所以把那個 pixel_addr 的算式想出來之後就沒設麼特別大的問題了。

3. 想對老師或助教說的話

Final project 差不多要開始進行了，然後期末又有好多期末考跟 project，真不知道到時候來不來得及完成 final project，然後希望期中考可以簡單一點讓大家都有得寫，不然就超級難好了...大家都 0 分就是大家都 100 分的意思了。