

CS 2104 02 Hardware Design and Labs 2018

Lab 7

學號：105072123 姓名：黃海茵

1. 實作過程

這次的 lab 我用了 5 個 state 來實作，先說明每個變數的意義

```
always @(posedge clk_22 or posedge rst_one) begin
    if (rst_one) begin
        count <= 0;
        position <= 0;
        clk27 <= 0;
    end
    else begin
        count <= next_count;
        position <= next_position;
        clk27 <= next_clk27;
    end
end

always @(posedge clk_sel) begin
    if (rst_one) state <= BLACK;
    else state <= next_state;
end

assign clk_sel = (state == DISPLAY)? clk_22 : clk;
```

count: 控制螢幕顯示圖片與黑屏的範圍

position: 控制圖片的移動

clk27: 計算圖片顯示的 $\text{clk}/2^{27}$

```
BLACK: begin
    {vgaRed, vgaGreen, vgaBlue} = 12'h0;
    next_state = btnC_one? RISINGCURTAIN : btnR_one? SLIDINGIN : btnL_one? BOX : btnU_one? SPLIT : BLACK;
end
```

BLACK: 按下 reset 或顯示完 $\text{clk}/2^{27}$ 時，螢幕顯示全黑，然後按下按鈕時進入各個不同轉場的 state。(每個按鈕都有做 one pulse)

```
RISINGCURTAIN: begin
    next_count = (count >= 480)? 480 : count + 2;
    {vgaRed, vgaGreen, vgaBlue} = (valid == 1'b1 && v_cnt >= (480 - count))? pixel : 12'h0;
    next_state = (count == 480)? DISPLAY : RISINGCURTAIN;
end
```

RISINGCURTAIN: 黑屏由下往上減少，但圖片本身並沒有移動，所以 position 為 0。當 $v_cnt \geq (480 - \text{count})$ 時顯示圖片，其餘顯示黑色。

```

SLIDINGIN: begin
    next_count = (count >= 640)? 640 : count + 2;
    next_position = (count >= 638)? 0 : (position > 0)? position - 1 : 319;
    {vgaRed, vgaGreen, vgaBlue} = (valid == 1'b1 && h_cnt < count)? pixel : 12'h0;
    next_state = (count == 640)? DISPLAY : SLIDINGIN;
end

```

SLIDINGIN: 黑屏由左向右減少，圖片也從左向右移動。所以當 $h_cnt < count$ 時顯示圖片，其餘顯示黑色。而 $position$ 從 319（圖片最右邊移到最左邊）開始往下減。

```

BOX: begin
    next_count = (count >= 80)? 80 : count + 1;
    {vgaRed, vgaGreen, vgaBlue} = (valid == 1'b1 && v_cnt <= (240 + count*3) && v_cnt >= (240 - count*3) &&
                                   h_cnt <= (320 + count*4) && h_cnt >= (320 - count*4))? pixel : 12'h0;
    next_state = (count == 80)? DISPLAY : BOX;
end

```

BOX: 黑屏依長寬比 4:3 減少，但圖片本身並沒有移動，所以 $position$ 為 0。我讓圖片從(320, 240)這一點開始顯示，然後當 $(240 - count*3) \leq v_cnt \leq (240 + count*3)$ 且 $(320 - count*4) \leq h_cnt \leq (320 + count*4)$ 時顯示圖片，其餘顯示黑色。

```

SPLIT: begin
    next_count = (count >= 640)? 640 : count + 2;
    next_position = (count >= 638)? 0 : (position > 0)? position - 1 : 319;
    if(valid == 1'b1 && h_cnt < count && (v_cnt < 120 || v_cnt >= 240 && v_cnt < 360))
        {vgaRed, vgaGreen, vgaBlue} = pixel;
    else if(valid == 1'b1 && h_cnt >= 640 - count && (v_cnt >= 120 && v_cnt < 240 || v_cnt >= 360))
        {vgaRed, vgaGreen, vgaBlue} = pixel;
    else {vgaRed, vgaGreen, vgaBlue} = 12'h0;
    next_state = (count == 640)? DISPLAY : SPLIT;
end

```

```

assign pixel_addr = (state == 3'b100 && (v_cnt >= 360 || v_cnt >= 120 && v_cnt < 240))?
    ((h_cnt>>1)+320*(v_cnt>>1)-position)% 76800 : ((h_cnt>>1)+320*(v_cnt>>1)+position)% 76800;

```

SPLIT: 黑屏在 $v_cnt < 120$ 和 $240 \leq v_cnt < 360$ 時由左向右減少，圖片由左向右移動。所以當 $h_cnt < count$ 時顯示圖片，其餘顯示黑色，而 $position$ 從 319（圖片最右邊移到最左邊）開始往下減。其餘黑屏則由右向左減少，圖片由右向左移動。所以當 $h_cnt \geq 640 - count$ 時顯示圖片，其餘顯示黑色，而此時的 $pixel_addr$ 和由左向右移動的不同，是減 $position$ （向右是加）。

```

DISPLAY: begin
    next_clk27 = clk27 + 1;
    {vgaRed, vgaGreen, vgaBlue} = (valid == 1'b1)? pixel : 12'h0;
    next_state = (clk27 == 31)? BLACK : DISPLAY;
end

```

DISPLAY: 螢幕顯示完整圖片，用 `posedge clk_22 trigger`，讓 $clk27$ 從 0 跑到 31(也就是 2^5)來做 $clk/2^5$ 。

2. 學到的東西與遇到的困難

iLms 上的 VGA sample code 是把 position 放在 mem_addr_gen module 裡面做，但這次的 lab 會用到四個轉場效果，不像 sample code 只有一個向上捲動的效果，所以我把 position 拉回 top module 裡的 state 一起做。

初步完成後，我發現我的圖片會顯示一些顏色錯誤的雜點，但移動位置都是正常的，而且雜點還會越來越多。檢查 code 好幾次，甚至一度越改越糟 ☹ 最後才找到是因為顯示的範圍出現負數（沒有給 count 上限），改正後終於 OK 了！

3. 想對老師或助教說的話

希望下次上機考可以再簡單一點，上次最後一題來不及打完 (；_；)

感恩老師 & 助教～～～