

EECS 2070 02 Digital Design Labs 2019

Lab 7

學號：104021215 姓名：熊磊

1. 實作過程

lab07_1

- 先利用 clock_divider 做出想要的二種頻率，然後將 input 中的 reset, en 做 debounce 的處理。之後改寫 mem_addr_gen 這個 module。
- 除了 mem_addr_gen 原本的 input, output，現在加入 en, dir 這兩個 input。在這個 module 中用了一個 always block，觸發條件為 clk 和 rst。

```
20 always @ (posedge clk, posedge rst) begin
21     if(rst)
22         position <= 0;
23     else if(en == 1)
24         if(dir == 0)
25             if(position < 319)
26                 position <= position + 1;
27             else
28                 position <= 0;
29         else
30             if(position > 0)
31                 position <= position - 1;
32             else
33                 position <= 319;
34     else
35         position <= position;
36 end
```

- always block 來控制一個 9 bits 的變數 position，需要 9 bits 的原因在於 $2^8 < 320$ ，所以需要 9 個 bits 才夠。這個 always block 觸發的條件為 posedge clk, posedge rst。在 rst 時，將 position 設為 0，其他時候看 en 是不是等於 1，若不是，則將 position 維持住；若是，再判斷方向為左或右來決定 position 加 1 或減 1。
- 接下來算要 output 的 pixel_addr，將水平位置除以 2 加上剛剛算出來的 position 一起取除以 320 的餘數，這樣可以使水平位置介於 0 到 319 之間，再加上垂直位置除以 2 再乘以 320，就可以控制垂直位置。

lab07_2

- 先利用 clock_divider 做出想要的二種頻率，然後將 input 中的 reset, shift, split 做 debounce 的處理，接著修改 mem_addr_gen2 這個 module。
- 這個 lab 要做 Shift 和 Split 因此可以分兩個部分來做。共有三個 state：INIT、SHIF、SPLI。

```
2 `define INIT 2'b00
3 `define SHIF 2'b01
4 `define SPLI 2'b10
```

```

109 // Disappear from Right to Left
110 if(count == 2'b00) begin
111     if((h_cnt >> 1) >= target) next_valid = 1'b0;
112     else next_valid = 1'b1;
113
114     if(target > 0) begin
115         next_target = target - 1;
116         next_count = count;
117     end
118     else begin
119         next_count = 2'b01;
120         next_target = 0;
121     end
122     next_done = done;
123 end
124
125 // Appear from Up to Down
126 else if(count == 2'b01) begin
127     if((v_cnt >> 1) <= target) next_valid = 1'b1;
128     else next_valid = 1'b0;
129
130     if(target <= 239) begin
131         next_target = target + 1;
132         next_done = done;
133         next_count = count;
134     end
135     else begin
136         next_count = 2'b10;
137         next_target = target;
138         next_done = 1'b1;
139     end
140 end

```

Shift

- 當從右往左消失的時候，如果 h_cnt 也就是 水平座標/2 比 target 還大時，就消失。Target 會從 319 往下降到 0。
- 當從上往下出現時，如果 v_cnt 也就是 垂直座標/2 比 target 還小時，就出現。Target 會從 0 往上升到 240。

Split

- 分成四個部分 top_left, bottom_right, top_right, bottom_left。
分裂的方法就是先控制哪裡要呈現黑的，然後根據上面四個 register 紀錄的數據，來呈現要顯示的區域的圖案。

```

181 // Top Left
182 if((h_cnt >> 1) <= 159 && (v_cnt >> 1) <= 119) begin
183     if((v_cnt >> 1) >= top_left) next_valid = 1'b0;
184     else begin
185         next_valid = 1'b1;
186         pixel = ((h_cnt >> 1) + 320 * ((v_cnt >> 1) + position)) % 76800;
187     end
188 end
189
190 // Bottom Left
191 else if((h_cnt >> 1) <= 159 && (v_cnt >> 1) <= 239) begin
192     if((h_cnt >> 1) >= bottom_left) next_valid = 1'b0;
193     else begin
194         next_valid = 1'b1;
195         pixel = (((h_cnt >> 1) + position) + 320 * (v_cnt >> 1)) % 76800;
196     end
197 end
198
199 // Top Right
200 else if((h_cnt >> 1) <= 319 && (v_cnt >> 1) <= 119) begin
201     if((h_cnt >> 1) <= top_right + 1) next_valid = 1'b0;
202     else begin
203         next_valid = 1'b1;
204         pixel = (((h_cnt >> 1) - position) + 320 * (v_cnt >> 1)) % 76800;
205     end
206 end
207
208 // Bottom Right
209 else if((h_cnt >> 1) <= 319 && (v_cnt >> 1) <= 239) begin
210     if((v_cnt >> 1) <= bottom_right) next_valid = 1'b0;
211     else begin
212         next_valid = 1'b1;
213         pixel = ((h_cnt >> 1) + 320 * ((v_cnt >> 1) - position)) % 76800;
214     end
215 end

```

2. 學到的東西與遇到的困難

以前很少用>>或<<來除以或乘以 2 的次方，這次作業用了這個寫法，可以省去一些力氣。這次也學會如何控制圖片的上下左右移動了。

3. 想對老師或助教說的話

謝謝老師與助教!