

EECS 2070 02 Digital Design Labs 2019	
Lab 3	
學號：104021215	姓名：熊磊

1. 實作過程

- 這次的 Lab 終於要 implement 在板子上了，覺得很有趣也很期待，第一個要做的是除頻器，其實這個比較容易，就是把 Basys3's clock 進來的 bit，用 always 找 posedge 就可以了。

```

23 module clock_divider(clk, clk_div);
24     parameter n = 26;
25     input clk;
26     output clk_div;
27
28     reg [n-1:0] cnt;
29     wire [n-1:0] cnt_next;
30
31     always@(posedge clk) begin
32         cnt <= cnt_next;
33     end
34
35     assign cnt_next = cnt + 1'b1;
36     assign clk_div = cnt[n-1];
37 endmodule

```

- 第一個要做的是簡單的 LED Controller，首先 SPEC 要求要先把 Basys3 的 clock，用 2^{26} 來除頻，因此我把除頻後的頻率放在 clk_d 裡面，然後 DFF 直接接 clk_d。

控制 LED 的 always block，是額外再寫一個 always block，我的作法是每次 posedge 就把整條 led[15:0]都設成 0，再把要亮的燈設成 1。

而要亮哪個燈我是用 cnt、cnt_next 來控制，根據 dir 的方向來決定要±1

```

30 wire clk_d;
31 reg [3:0] cnt;
32 wire [3:0] cnt_next;
33 reg [15:0] led;
34 clock_divider c1(.clk(clk), .clk_div(clk_d));
35
36 always@(posedge clk_d, posedge rst) begin
37     if(rst == 1) begin
38         cnt = 15;
39     end
40     else if(en == 1) begin
41         cnt = cnt_next;
42     end
43     else begin
44         cnt = cnt;
45     end
46 end
47
48 always@(posedge clk_d) begin
49     led[15:0] = 15'b0;
50     led[cnt] = 1'b1;
51 end
52
53 assign cnt_next = (dir) ? (cnt + 1'b1)%16 : (cnt - 1'b1)%16;

```

- 第二個要做的是，不同速度和大小的 Led controller，一個Mr.1 一個 Mr.3，因為要用 dir 來換速度跟方向，因此Mr.1 和 Mr.3 的 clock，我也直接用 dir 來做選擇。
其他控制 Led 亮的方式，跟上一個一樣，每次 posedge 就把整條 led[15:0]都設成 0，再把要亮的燈設成 1。

```

38  always@(*) begin
39      led[15:0] = 16'b0;
40      led[mr1] = 1'b1;
41      led[(mr3 + 1'b1)%16] = 1'b1;
42      led[(mr3 - 1'b1)%16] = 1'b1;
43      led[mr3] = 1'b1;
44  end
45
46  assign mr1_next = (dir) ? (mr1 + 1'b1)%16 : (mr1 - 1'b1)%16;
47  assign mr3_next = (dir) ? (mr3 + 1'b1)%16 : (mr3 - 1'b1)%16;
48  assign mr1_clk = (dir) ? clk_slow : clk_fast;
49  assign mr3_clk = (dir) ? clk_fast : clk_slow;

```

- 第三個要做的是比較有挑戰性的設計，Mr.3 是一個胖子一直往右跑，Mr.1 碰到 Mr.3 就會換方向，而且Mr.1 跑得比較快。
這邊要考慮的就是，因為Mr.1 和 Mr.3 速度不一樣，所以會有 latch 發生，造成Mr.1 由右邊往左撞到 Mr.3 的時候，Mr.1 換方向，但是 Mr.3 這時候往右前進一格，所以 Mr.1 在換方向往右跑的時候，又再次撞到 Mr.3 的右邊，又在換了一次方向，使得Mr.1 看起來就好像直接穿過 Mr.3。
為了考慮這個情況進去，我在設計 Mr.1 的 dir 的時候，也用了一個 dir_next，他也是用跟 Mr.1 的 clock。

```

47  assign mr1_next = (dir) ? (mr1 + 1'b1)%16 : (mr1 - 1'b1)%16;
48  assign mr3_next = (mr3 + 1'b1)%16;
49  assign dir_next = (dir == 1 && (mr1 + 1'b1)%16 == (mr3 - 1'b1)%16) ? ~dir :
50                  (dir == 0 && (mr1 - 1'b1)%16 == (mr3 + 1'b1)%16) ? ~dir :
51                  (dir == 1 && (mr1 + 1'b1)%16 == mr3) ? ~dir : dir;

```

2. 學到的東西與遇到的困難

這次的 Lab，我在考量 lab03_3 的時候一直想不通要怎麼解決，Demo 的時候助教給我一些提示，讓我大概清楚是遇到什麼問題，也才能順利做完，謝謝助教解惑！

3. 想對老師或助教說的話

謝謝助教的幫忙☺