Lab04_counter_106061146_陳兆廷

Pre-labs: 同 exp1。

Experiments:

1. exp1 題目要求做一個 8 位元的 ring counter，output 是 01010101。

```
q[0]<=q[7];
q[1]<=q[0];
q[2]<=q[1];
q[3]<=q[2];
q[4]<=q[3];
q[5]<=q[4];
q[6]<=q[5];
q[7]<=q[6];
```

圖(一)

首先需要一個除頻器，輸出的新頻率當作 shifter 的頻率，而 shifter 要做的事情就是每一次 clk 的啟動就會個個位元移動一次。再用一個 module 將除頻器和 shifter 合在一起。

因此需要兩個輸入值，clock 和 reset，以及一個 8 位元的輸出值 q。當 clk 啟動時做右圖的動作。如此一來一個 ring counter 就完成了。

而將此 pre-lab 用在 FPGA 上就只需要寫個 constraints，把 q 的每一位元輸入在 8 個 led 燈上，再將 rst 輸入再一個按鈕上即可。

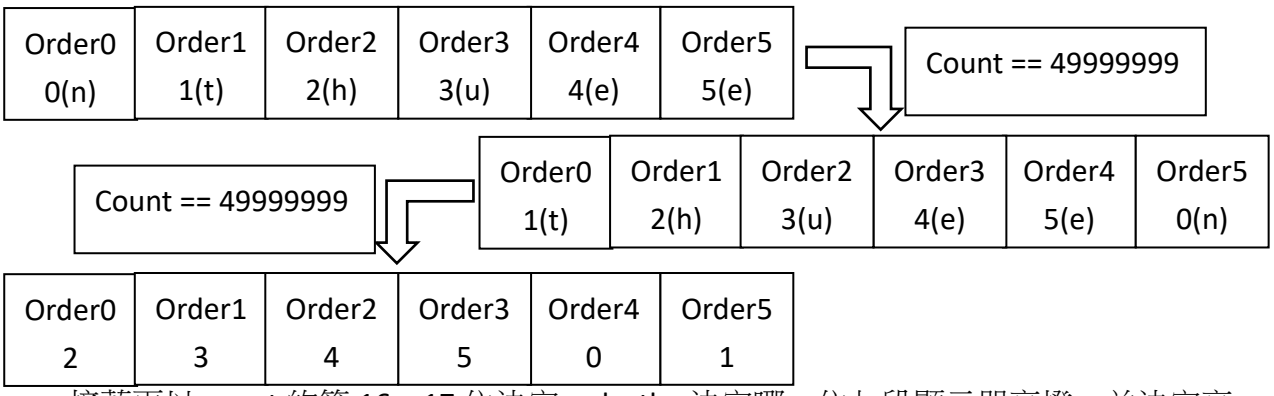
心得：exp1 應該是讓我們熟悉 shifter 的動作如圖(一)，還有將除頻器跟 shifter 合在一起的方法。

2. Exp2 題目要求將 exp1 的 ring counter 變成可以隨意設定初始值(initial value can be set randomly)。因此我多加一個 8 位元的輸入值，將輸出值同圖(一)方法 shift 下一位，再將 constraints 加入 8 個 I/O 的 switch，用來代表輸入值的 8bits，就可以了。

心得；這題跟 exp1 沒有甚麼不同。

3. Exp3 這次要用 7 段顯示器，且顯示 4 種不同的字樣，因此一定有個輸出值是 8 位的 D_ssd 輸出，還有 4 位的 ssd_ctl，然後輸入值是 clk 及 rst。首先要有 reg [26:0]counter 來計算 shifter 的頻率，當達到 50M 時(假設頻率為 1HZ)，shift 到下一個字元。

接著我用 6 個 3 位元的變數，設為[2:0]order0 ~ [2:0]order5，裡面裝著 0~5 的數字，個代表著 n、t、h、u、e、e，如下圖。




接著再以 count 的第 16、17 位決定 ssd_ctl，決定哪一位七段顯示器亮燈，並決定亮 order0~4 的哪一位，將那一位用 case 把個位數轉成英文字的 8 位元顯示器，再用 case 輸出至 D_ssd，完成那一瞬間的亮燈。

```
if(count == 26'd49999999)
begin
    count <= 26'd0;
    order0 <= order5;
    order1 <= order0;
    order2 <= order1;
    order3 <= order2;
    order4 <= order3;
    order5 <= order4;
end
```

```
case(count[17:16])
    2'b00 : ssd_ctl = 4'b1110;
    2'b01 : ssd_ctl = 4'b1101;
    2'b10 : ssd_ctl = 4'b1011;
    2'b11 : ssd_ctl = 4'b0111;
endcase
case(ssd_ctl)
    4'b1110: order = order1;
    4'b1101: order = order2;
    4'b1011: order = order3;
    4'b0111: order = order4;
endcase
```

```
case (order)
    3'd0: D_ssd = `SS_e;
    3'd1: D_ssd = `SS_u;
    3'd2: D_ssd = `SS_h;
    3'd3: D_ssd = `SS_t;
    3'd4: D_ssd = `SS_n;
    3'd5: D_ssd = `SS_e;
    default: D_ssd = `SS_f;
endcase
```

心得：這次需要應用到 shifter 裡 shift 的概念，並把上次 7 段顯示器的內容應用在上面，比較難。

4. Bonus 要做出三種不同的 shifter，以三位元的 one-hot 控制，加上一位元的 I/O 當作左移、右移。初始值為 1010，並用一個按鈕來控制 shifter 的開始。

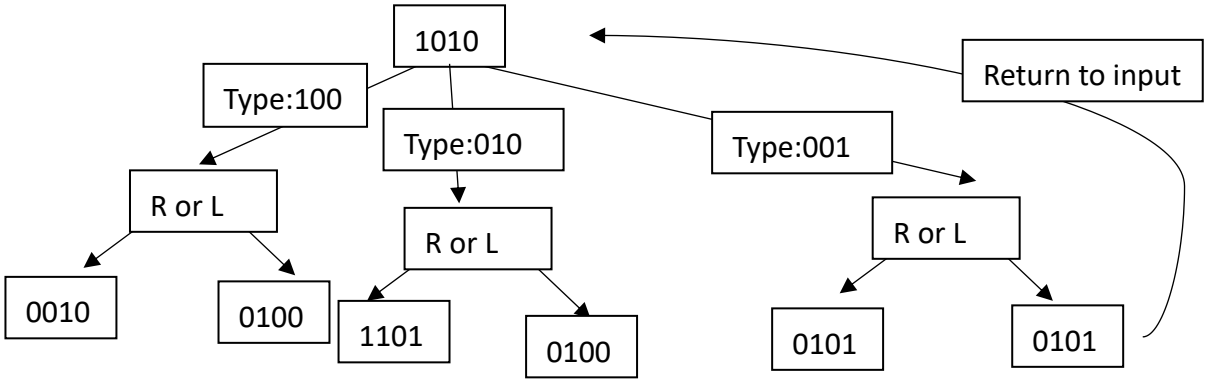
我用五個 input，clk、rst、還有一個 right-left-switch (rlsw)是控制左右移，一個[2:0]的 type 控制 shfiter 的種類，一個開始的按鈕(bot)。

有兩個 output，一個 D_ssd 是輸出 1 或是 0，ssd_ctl 是輸出 7 段顯示亮的位置。

再來要兩個 reg，一個是 count 來計時，一個是 4 位元的變數[3:0]order 來儲存當下四位的數值。

首先先用 if 當 rst==1 時把 count 歸零跟 order 歸回 1010，接著當 rst 不是 1 時，對 bot 做 if、else，在 bot 按下去時，開始對 type 做 if，else，在 100、010、001 時各做不同種類 shifter 該做的事，在裡面再寫一個 if、else 寫是左移或是右移，配合不同狀況寫出三種不同的 shifter，補 0 或是最高位或是無限巡迴。

接著再用 exp3 七段顯示器顯示的方式，顯示出各位的 order 當下的值，就大功告成了。




這樣寫出來的效果就是，撥好下面的 I/O switch 後，按住按鈕 bot，七段顯示器就會開始照下面的條件 shift，放開後停止，可以再修改下面的 I/O 改成其他種的 shifter，當要回到 1010 時就按一下 rst 即可。

心得：這個 bonus 是最有挑戰性的 bonus，我想了蠻久的，也發信件問了助教，我覺得寫出來後很有成就感。