EECS 2070 02 Digital Design Labs 2019
Lab 2

學號：107062115　姓名：陳博暐

1. 實作過程

1.

```
always @(posedge clk, negedge rst_n)begin
    if (rst_n == 1'b0) next_out = 4'b0000;
    else if (en == 1'b0) next_out = next_out;
    else begin
        if (in == 1'b0) begin
            if (dir == 1'b1) {carry_out, next_out} = {1'b0, next_out} + 5'b00001;
            else {carry_out, next_out} = {1'b0, next_out} - 5'b00001;
        end
        else next_out = data;
    end
end

assign out = temp_out;
```

我讓 counter 裡所有的東西都受 clk 控制，因此每過一個 clk 才會產出一個新的答案。而在加一或減一那部份，因為我不確定 overflow 會出現什麼狀況所以我用了一個 carry_out 將 overflow 的狀況去除，對於下一題的 carry out 也好解決。

Testbench 的部份我是參考助教所發的第二題的來改。

2.

這提示要設計一個 gray code counter 要先完成 1-digit 的 gray code counter 然後在湊成 2-digit。

在設計 1-digit 的 gray code counter 時，我採用的想法是：先造出一個正常的 counter ，再將 counter 的值轉換成 gray code 。因為這題有給定 gary code 的順序，因此可以當程式一個 counter ，再做值的轉換就行了。

因此設計細節，counter 基本上個前一題一樣，而轉換的過程，我用 case 來一一對應，因為數量不多就全部展開了。

```
always @(*) begin
    case (counter_out)
        4'b0000: next_out = 4'b0000;
        4'b0001: next_out = 4'b0001;
        4'b0010: next_out = 4'b0011;
        4'b0011: next_out = 4'b0010;
        4'b0100: next_out = 4'b0110;
        4'b0101: next_out = 4'b0111;
        4'b0110: next_out = 4'b0101;
        4'b0111: next_out = 4'b0100;
        4'b1000: next_out = 4'b1100;
        4'b1001: next_out = 4'b1101;
        4'b1010: next_out = 4'b1111;
        4'b1011: next_out = 4'b1110;
        4'b1100: next_out = 4'b1010;
        4'b1101: next_out = 4'b1011;
        4'b1110: next_out = 4'b1001;
        4'b1111: next_out = 4'b1000;
        default : next_out = 4'b0000;
    endcase
end
```

設計完 1-digit 之後，就要湊出 2-digit 了，具體想法是，個位數的 counter 就一直數，而十位數的 counter 在個位數 counter 產出 cout 為 1 時，就要加一。因此將個位數的 cout 接到十位數的 en。

3. (bonus)

按照題目給的電路圖寫，然後就完成了。

2. 學到的東西與遇到的困難

先說這次應該有什麼要改進的好了，這次我的寫法將很多東西都放在受 clk 控制的 always block 裡，雖然我不太確定這樣的寫法是正確或是建議不要這樣寫，但是在跟朋友討論的時候，我們得出的結論是應該在受 clk 控制的 always block 裡（flip flop）只有給值而已。而運算的部份用 combinational circuit 來做。下次寫的時候會往這個大方向做改進！

在寫第一題的 testbench 時，雖然是複製助教的 testbench，但是要自己設計波型時，馬上就遇到問題了，要用文字的方法是寫出波型非常的不直覺與人性化啊，因此寫了很久才將測試波型寫出來，不過想想好像也直有這種方法比較快速（？

在寫第二題的時候就真的有點崩潰了，看波型圖時常會發現一些很奇怪的 output ，比如說我的十位數 counter 會延遲一個 clk 才 + 1，或是 cout 突然變 1，後來是將 cout 分開來判斷而不是受 clk 控制。

第三題的 testbench 有點問號，因為是產生亂數的話，好像也不太知道要如何驗證答案是否正確，只能看著波形圖，然後說這沒規律啊。

3. 想對老師或助教說的話

來個笑話好了：

未達放假標準的颱風，猜一立委名？

.

.

.

.

.

.

.

.

.

.

.

.

.

費鴻泰（台語）