

1. 組別：第 22 組
2. 組員：404415073 電機二 蔡孟勳 404415055 電機二 劉恩瑞
3. 題目名稱：實驗 5 除頻器
4. 功能說明：

這次的實驗為撰寫一個擁有計數功能的除頻器

輸入訊號為一個 1kHz 的 clock 和一個脈波；

輸出訊號為以三個 LED 燈顯示的計數結果。

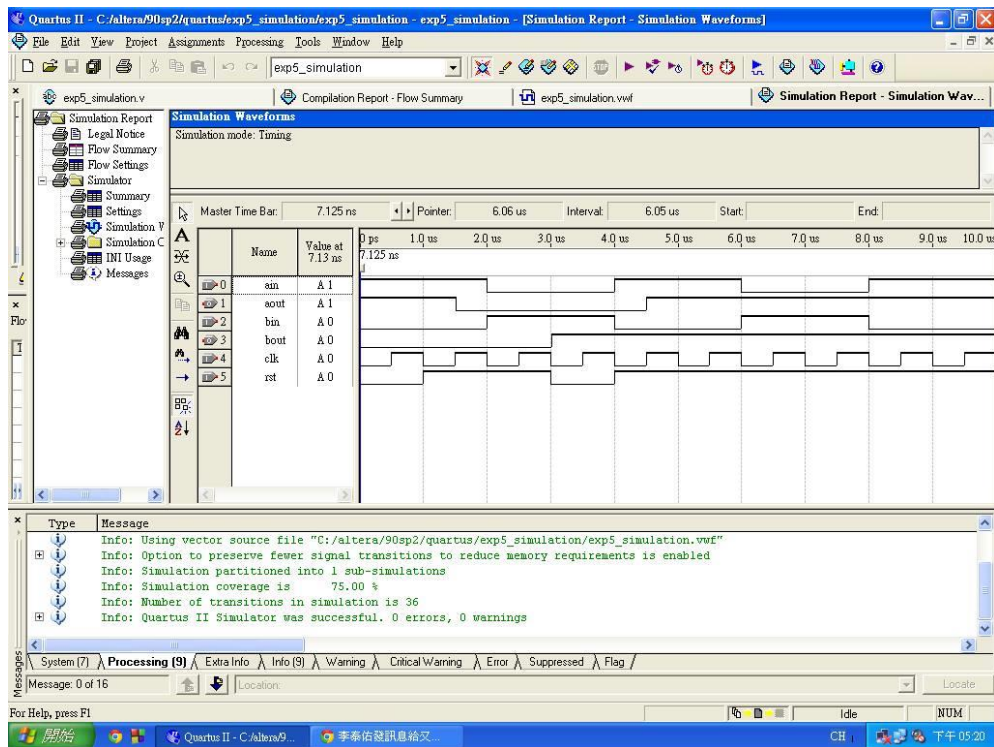
整個電路的功能為：可以消除彈跳現象並計數的除頻器

PS.

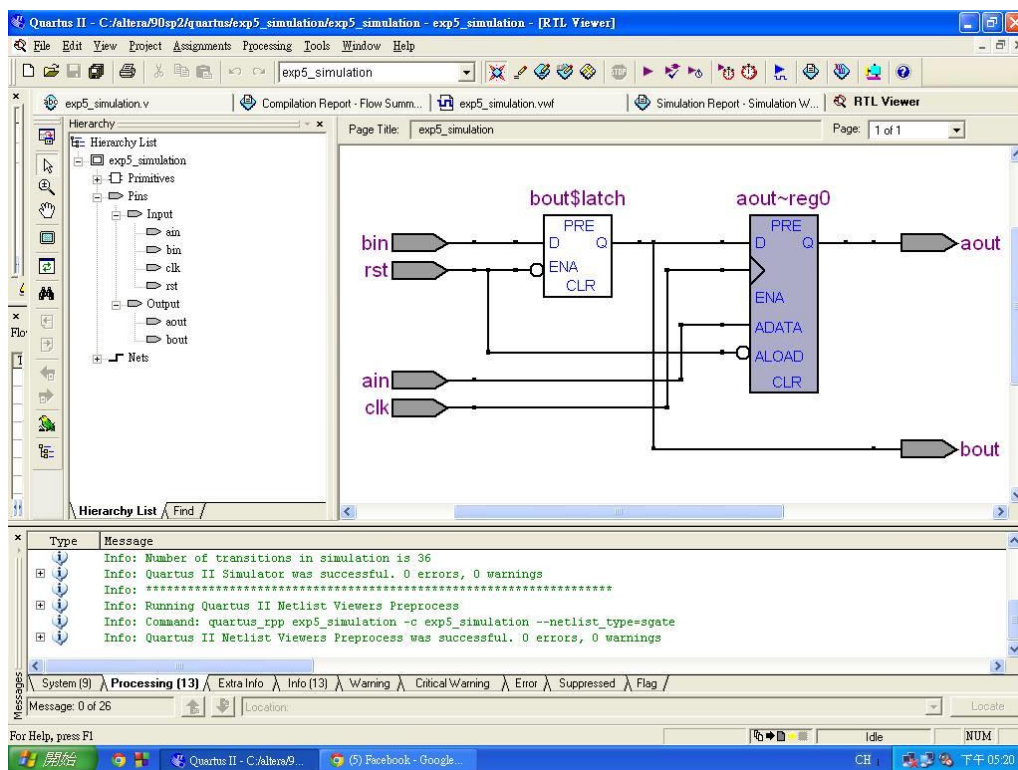
因為實作板只能提供 10MHz 的 clock，因此我們需要「除頻器」將之降至 1kHz。除頻的方式為另設一個 kHz，在 MHz 經過 5000 次的週期時，讓 kHz 反相，因此 MHz 每經過 10000 個週期，kHz 就會經過 1 個週期，則 10MHz 的 clock 就能降至 1kHz 的 clock 了。

消除彈跳現象的方式則與講義中的方式相同。

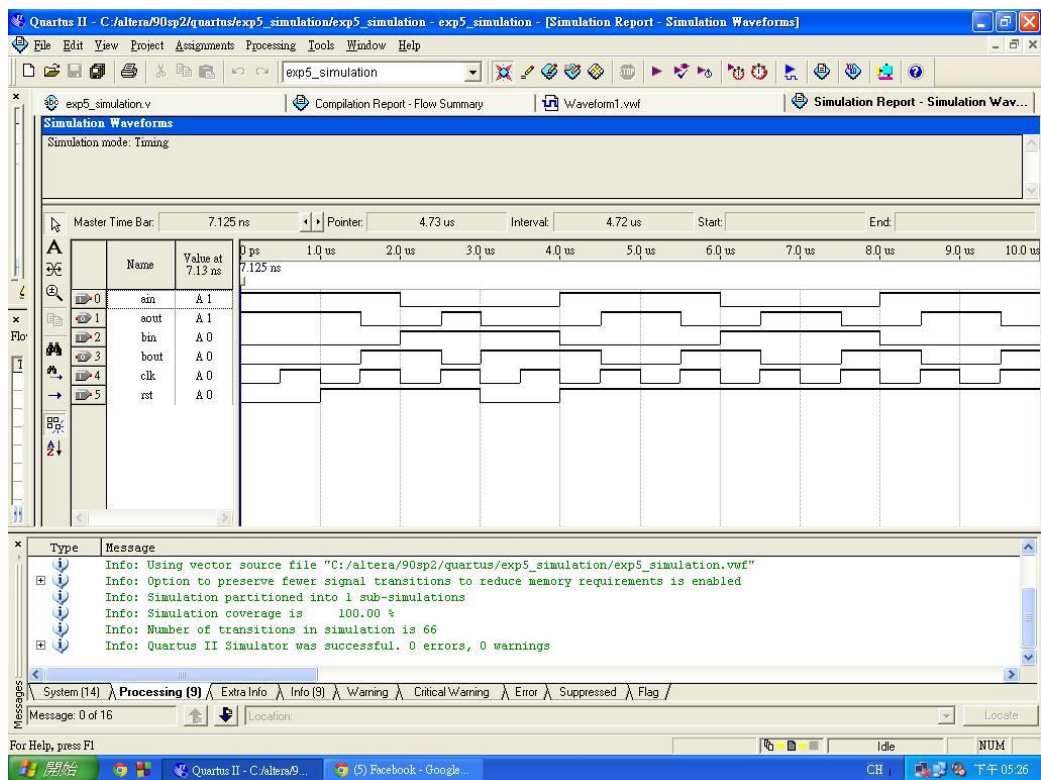
5. Blocking VS Non-blocking 模擬



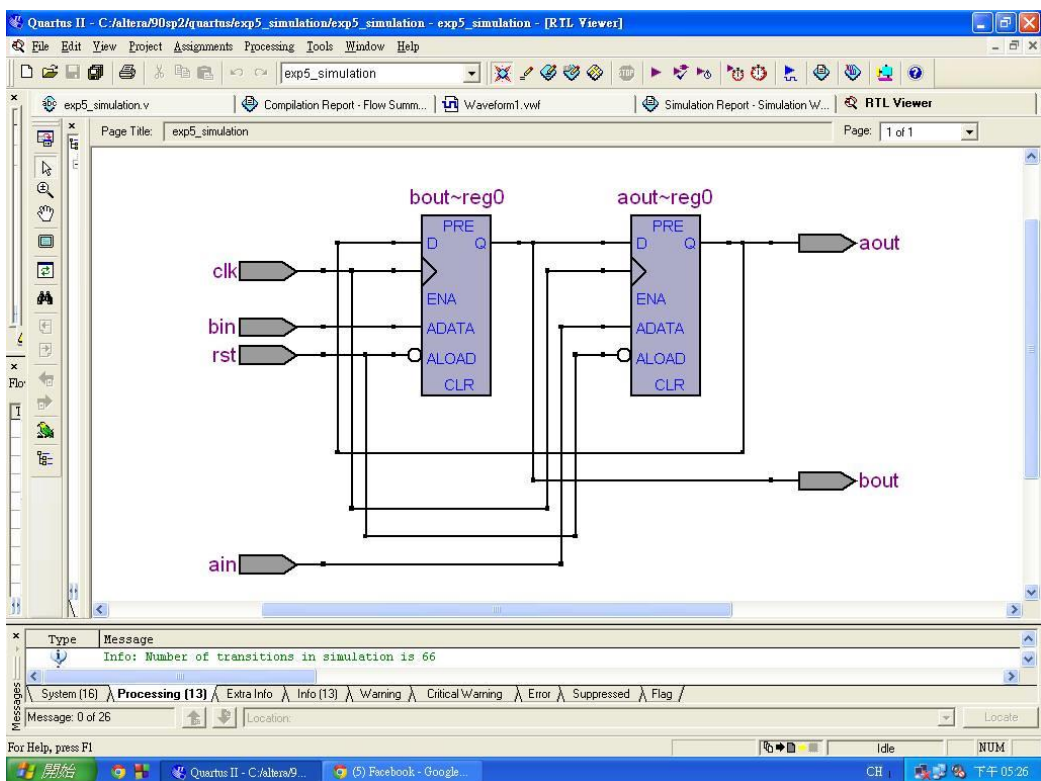
(Blocking 模擬圖)



(Blocking 接角圖)



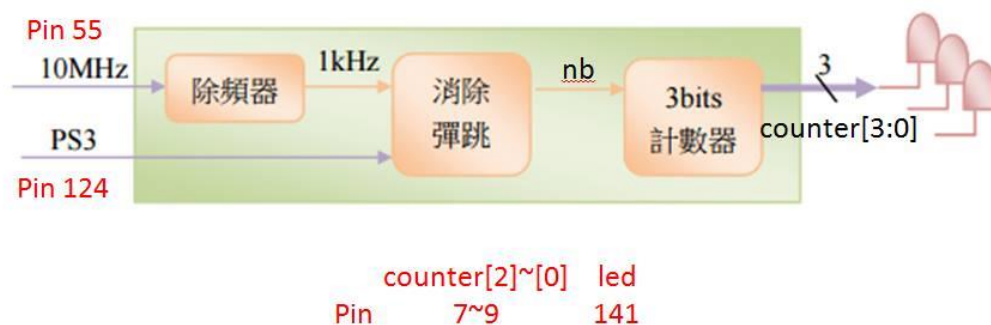
(Non-blocking 模擬圖)



(Non-blocking 接角圖)

由模擬圖可得知：clock 正緣觸發或 reset 負緣觸發且 reset=1 時，在 Blocking 的型式下，aout=bout；在 Non-blocking 的型式下，aout 和 bout 的值互換。如果由程式碼與接角圖判斷則可得知：Blocking 的程式碼雖然是同時進行，但結果仍是有順序性的；Non-blocking 的程式碼與結果則都是同時進行，因此 aout 與 bout 兩個值會互換。

6. 硬體架構圖：



電路設計的想法：

這次的實驗其實可以概略分為三個部分：除頻、消除彈跳現象、計數。由於實作板只能提供 10MHz 的脈波，因次我們先設計一個除頻器讓 10MHz 降至題目要求的 1kHz，再利用講義中的方式消除按壓 PS3 造成的彈跳現象，最後再讓 counter 隨著 PS3 按壓的次數計數，這樣就可以完成這次的實驗了。

7. 程式碼&註解：

```
1 module exp5(counter,led,MHz,PS3); //三位元計數器模組宣告
2   input MHz,PS3; //輸入為MHz,PS3
3   output led; //輸出為led燈
4   output reg [2:0]counter; //輸出為3-bits共3個數counter[0]~[2]
5   reg kHz; //保留kHz的值到下一次指定新值
6   reg [12:0]count; //保留count[0]~[12]的值到下一次指定新值
7   wire nb; //三位元計數器的邏輯電路圖中會用到1條線連接不同的模組
8   always@(posedge MHz) //當MHz正緣觸發時，底下的Behavioral Model的敘述會被執行
9   begin //Behavioral Model開始
10     if (count==4999) //如果count的值等於4999時，執行以下敘述
11     begin //開始
12       count<=0; //count的值變為0
13       kHz<=~kHz; //kHz反相
14     end //結束
15     else //其他情況
16       count<=count+1; //count的值+1
17   end //Behavioral Model結束
18
19   count<=count+1; //count的值+1
20   end //Behavioral Model結束
21
22   debounce d1(nb,kHz,PS3); //使用消除彈跳的模組，輸出為nb，輸入為kHz,PS3
23
24   always@(posedge PS3) //當PS3正緣觸發時，底下的Behavioral Model的敘述會被執行
25   begin //Behavioral Model開始
26     if (nb==1) //如果nb的值=1時，執行以下敘述
27     begin //開始
28       if (counter==7) //如果counter的值等於7時，執行以下敘述
29       counter<=0; //counter的值變為0
30       else //其他情況
31       counter<=counter+1; //counter的值+1
32     end //結束
33   end //Behavioral Model結束
34   assign led=1'b1; //宣告LED燈為1位元，二進制，數值為1
35 endmodule //三位元計數器模組結束
36
37 module debounce(out,kHz,in); //消除彈跳的模組宣告
38   input kHz,in; //輸入為kHz,in
39   output out; //輸出為out
40   reg [10:0]d; //保留d[0]~d[10]的值到下一次指定新值
41   always@(posedge kHz) //當kHz正緣觸發時，底下的Behavioral Model的敘述會被執行
42   begin //Behavioral Model開始
43     d[10]<=d[9]; //d[10]的值變為d[9]的值
44     d[9]<=d[8]; //d[9]的值變為d[8]的值
45     d[8]<=d[7]; //d[8]的值變為d[7]的值
46     d[7]<=d[6]; //d[7]的值變為d[6]的值
47     d[6]<=d[5]; //d[6]的值變為d[5]的值
48     d[5]<=d[4]; //d[5]的值變為d[4]的值
49     d[4]<=d[3]; //d[4]的值變為d[3]的值
50     d[3]<=d[2]; //d[3]的值變為d[2]的值
51     d[2]<=d[1]; //d[2]的值變為d[1]的值
52     d[1]<=d[0]; //d[1]的值變為d[0]的值
53     d[0]<=in; //d[0]的值變為in的值
54   end //Behavioral Model結束
55   and(out,d[6],d[5],d[4],d[3],d[2],d[1],d[0]); //邏輯閘AND，輸入為d[0]~d[6]；輸出為out
56 endmodule //消除彈跳的模組結束
```

8. 心得：

404415073 蔡孟勳

這次的實驗其實沒有上次實驗來的複雜，但我們在打程式與燒錄時，遇到的問題卻比上次的實驗來的多，雖然計數器的 LED 燈有在閃爍，但並沒有如預期地一一計數上去，因此我們判斷是彈跳現象並未被消除，最後我們加長了消除彈跳現象的時間，結果就比較符合預期的了，往後的實驗為了提繩實作的成功率，或許消除彈跳現象的時間都應該家長一點。