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 9 以 Gate Level 設計有限狀態機

4. 功能說明：

這次的實驗為撰寫一個有限狀態機 (以 Gate Level 方式呈現)

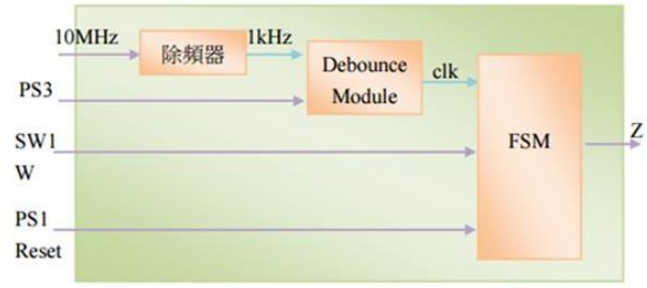
輸入訊號為任意 0 或 1 的訊號

輸出訊號為 1(相鄰兩個輸入含奇數個 1 時)

或 0(相鄰兩個輸入含偶數個 1 時)

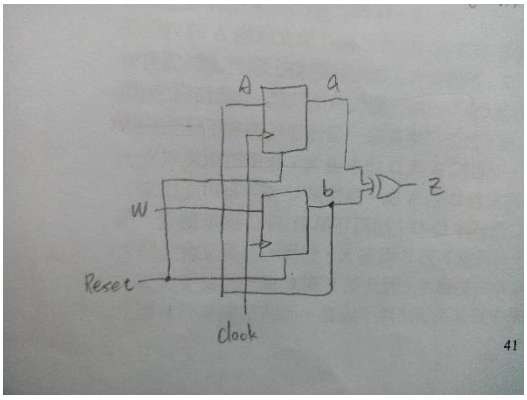
整個電路的功能為：可得知特定狀態的有限狀態機

5. 硬體架構圖：



(左圖為整體概念圖)

	MHz	PS3	Reset	Z	led	W
Pin	55	124	54	7	141	47

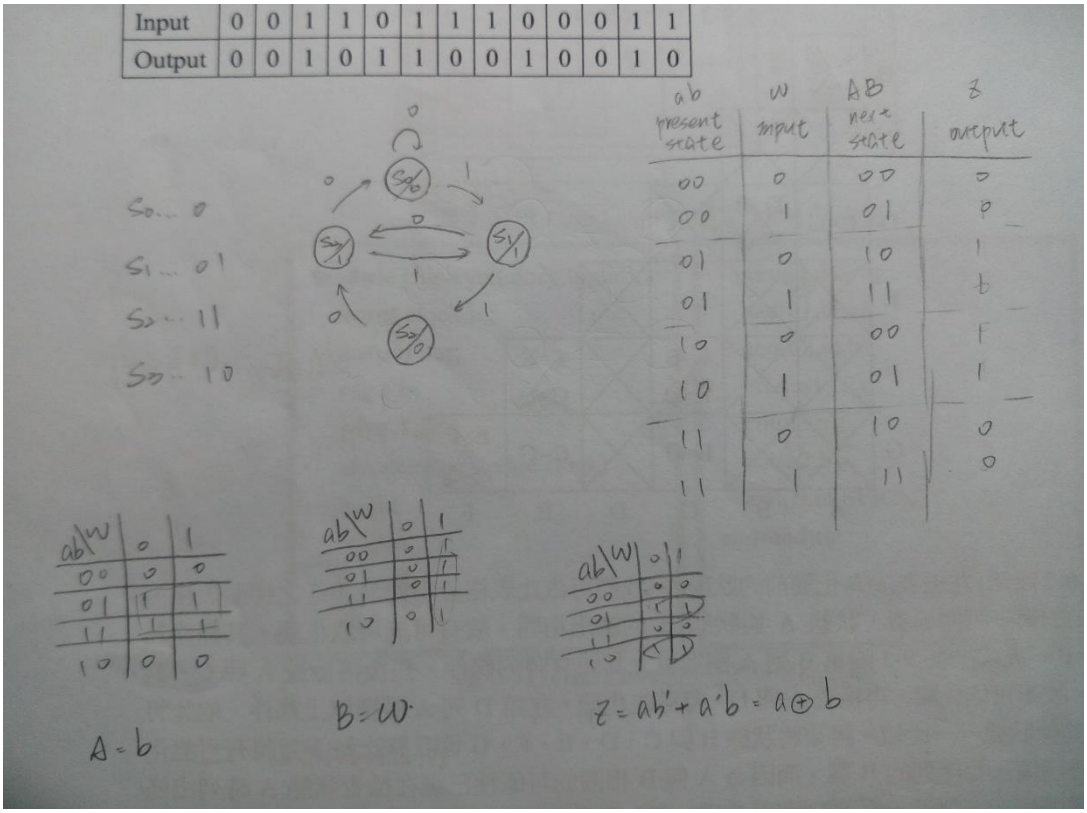


(右圖為 FSM 電路設計圖)

電路設計的想法:

這次的實驗有兩種做法：Mealy-type 和 Moore-type，我們這組選擇 Moore-type 的做法。題目要求為：以相鄰兩個輸入來判斷，如果有奇數個 1 時，輸出為 1，其餘情況則輸出為 0。我們先把可能的狀態列出來，然後決定目前輸入 0 的話應該要跳到哪個狀態，輸入 1 的話又該跳到哪個狀態，最後在題目要求的情況下使輸出為 1，即可畫出如下的 state graph 和 state table，但這次題目還要求要用 Gate Level 的方式呈現，由於狀態較少，因此我們決定要用 K-map 進行化簡 (化簡的過程如下)，化簡完後，以電路搭配 register 的方式即可完成本次實驗。

(下圖為 state graph , state table , K-map)



6. 程式碼&註解：

```
module exp9(Z,led,W,PS3,MHz,Reset); //以 Gate Level 設計有限狀態機模組宣告
 input W,PS3,MHz,Reset; //輸入為W,PS3,MHz,Reset
 output Z,led; //輸出為Z,led
 reg a,b; //保留a,b的值到下一次指定新值
 wire A,B,clk,kHz; //邏輯電路圖中會用到4條線連接不同模組
 div10000 d1(kHz,MHz); //使用除頻器1的模組，輸出為kHz，輸入為MHz
 debounce d2(clk,kHz,PS3); //使用消除彈跳模組，輸出為clk，輸入為kHz,PS3
 always@(posedge clk or negedge Reset) //當clk正緣觸發或Reset負緣觸發，底下的Behavioral Model的敘述會被執行
  if(~Reset) //如果Reset反相為true，執行以下敘述
   begin //開始
    a=0; //a的值變為0
    b=0; //b的值變為0
   end //結束
  else //其他情況
   begin //開始
    a=A; //a的值變為A的值
    b=B; //b的值變為B的值
   end //結束
 assign A=b; //宣告A的值變為b的值
 assign B=W; //宣告B的值變為W的值
 xor x1(Z,a,b); //邏輯閘XOR，輸入為a,b，輸出為Z
 assign led=1'b1; //宣告led的值為1
endmodule  //結束模組
```

```
module div10000(out,in); //除頻器除以10000的模組宣告
 input in; //輸入為in
 output reg out; //保留輸出out的值到下一次指定新值
 reg [12:0]counter; //保留輸出counter[0]~[12]的值到下一次指定新值
 always@(posedge in) //當in正緣觸發時，底下的Behavioral Model的敘述會被執行
  if(counter == 13'd4999) //如果counter的值等於4999時，執行以下敘述
   begin //開始
    counter<=13'd0; //counter的值變為0
    out<=~out; //out反相
   end //結束
  else //其他情況
   counter<=counter+13'd1; //conuter的值+1
endmodule //結束模組

module debounce(out,kHz,in); //消除彈跳的模組宣告
 input in,kHz; //輸入為in,kHz
 output out; //輸出為out
 reg [10:0]d; //保留d[0]~d[10]的值到下一次指定新值
 always@(posedge kHz) //當kHz正緣觸發時，底下的Behavioral Model的敘述會被執行
  begin //開始
   d[10]<=d[9]; //d[10]的值變為d[9]的值
   d[9]<=d[8]; //d[9]的值變為d[8]的值
   d[8]<=d[7]; //d[8]的值變為d[7]的值
   d[7]<=d[6]; //d[7]的值變為d[6]的值
   d[6]<=d[5]; //d[6]的值變為d[5]的值
   d[5]<=d[4]; //d[5]的值變為d[4]的值
   d[4]<=d[3]; //d[4]的值變為d[3]的值
   d[3]<=d[2]; //d[3]的值變為d[2]的值
   d[2]<=d[1]; //d[2]的值變為d[1]的值
   d[1]<=d[0]; //d[1]的值變為d[0]的值
   d[0]<=in; //d[0]的值變為in的值
  end //結束
  and(out,d[10],d[9],d[8],d[7],d[6],d[5],d[4],d[3],d[2],d[1],d[0]); //邏輯閘AND，輸入為d[0]~d[10]；輸出為out
endmodule  //消除彈跳的模組結束
```

7. 心得：

404415073 蔡孟勳

這次的實驗也滿簡單的，跟上次的差別就只是要用 Gate Level 的方式呈現，因此我們多了用 K-map 化簡的步驟，最後用電路圖描述出本次實驗的結果，希望之後的實驗也都可以像這次實驗一樣順利。