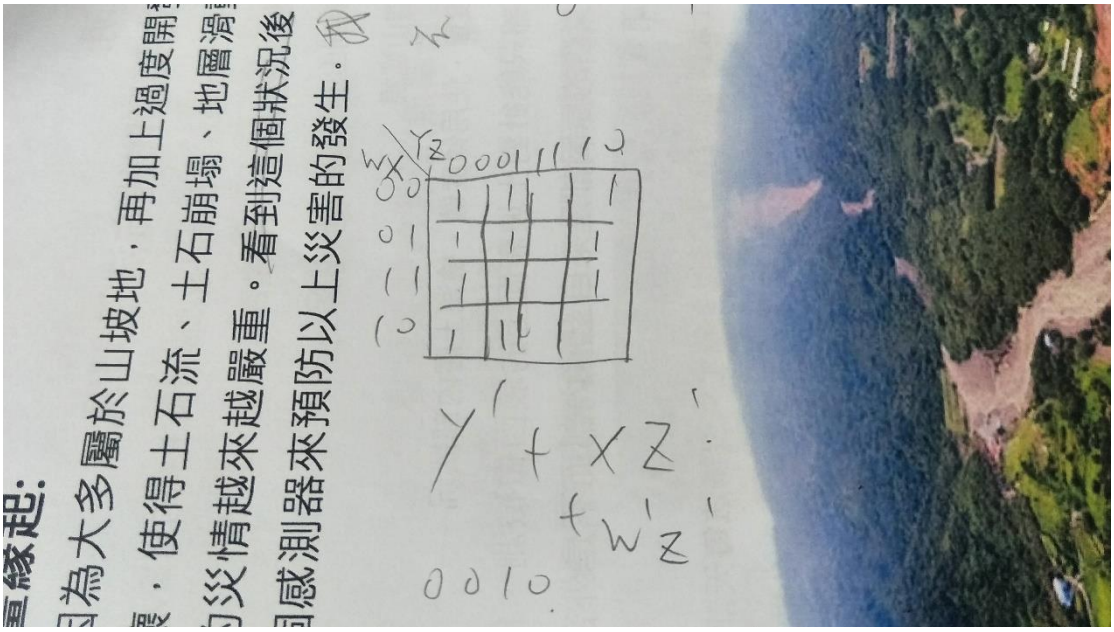
這次的實驗為卡諾圖化簡電路後，證實與原式相等

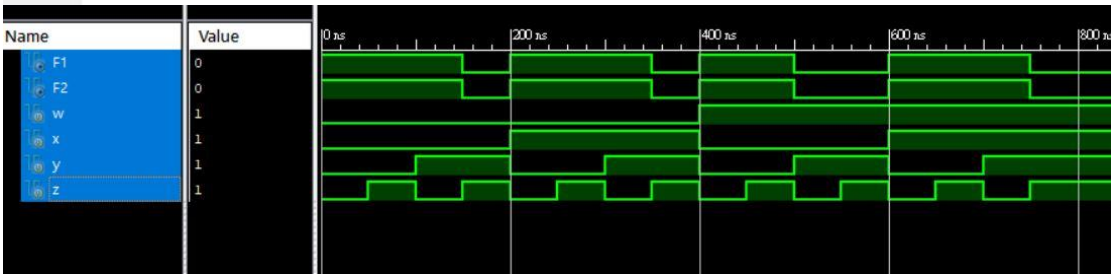
第一個實驗

題目的卡諾圖:



化簡成為 y’ +xz’

```
19  //
20  //////////////////////////////////////////////////////////////////////////////////
21  module DataFlow(output F1 ,output F2,input w,input x,input y ,input z);
22     assign F1=((~w)&(~y))|((~w)&(~z))|((x)&(~z))|((w)&(~y)&(~z))|((w)&(~y)&(z));
23     assign F2=(~y)|((x)&(~z))|((~w)&(~z));
24
25
26  endmodule
27
```



證實原式(F1)相等於卡諾圖化簡後式(F2)

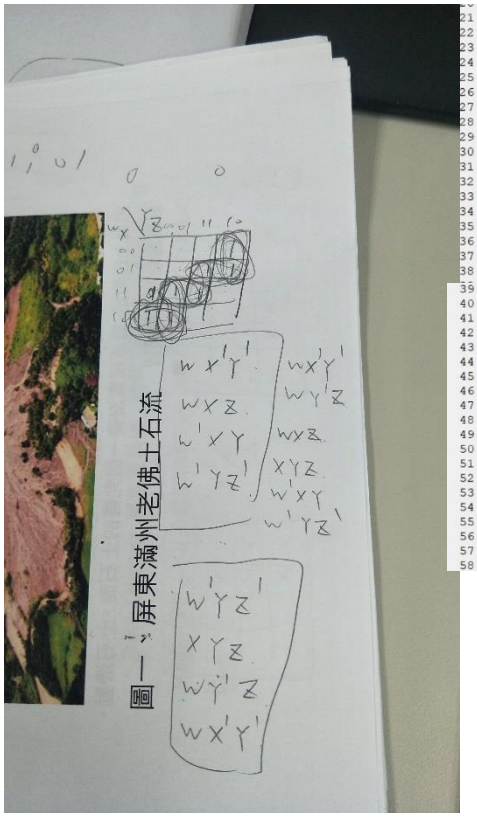
第二個實驗

化簡後理論上應該只有兩個最佳解

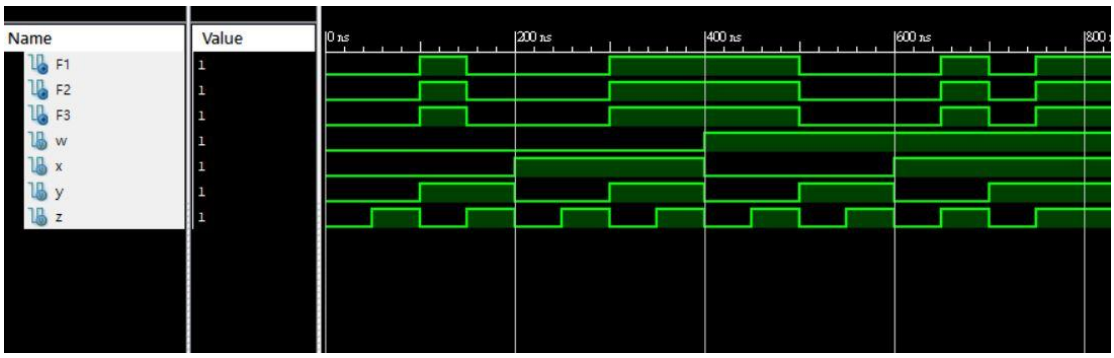
因此加上另一個組合

證明 wx′ y′ +wxz+w′ xy+w′ yz′ =w′ yz′ +xyz+wy′ z+wx′ y′ =

wx′ y′ +wy′ z+wxz+xyz+w′ xy+w′ yz′



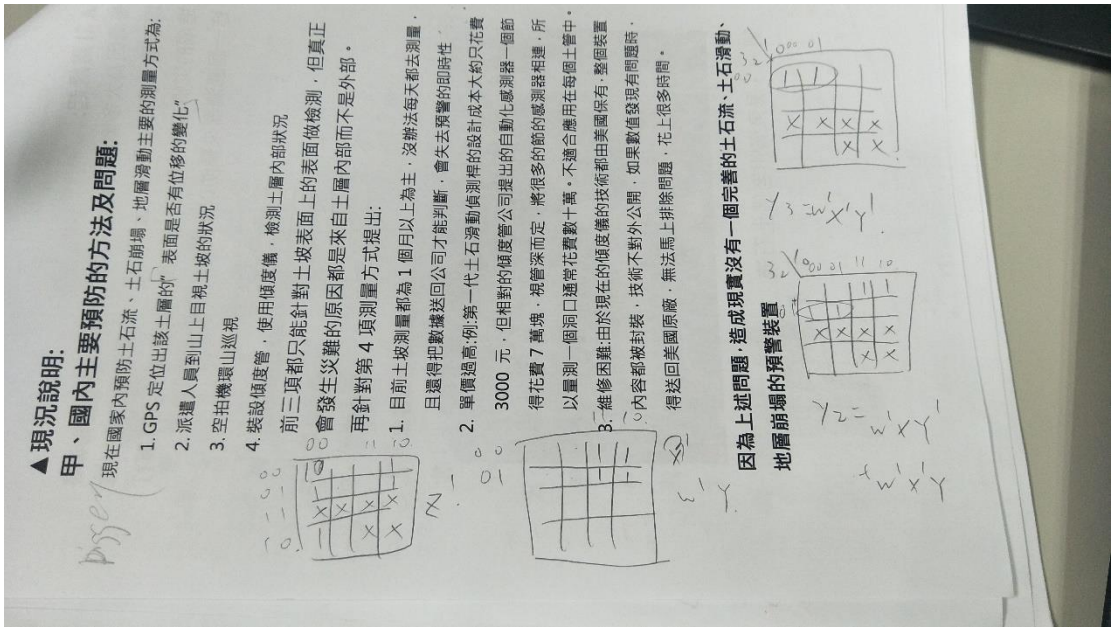

```
module modu(output F1,output F2,output F3,input w,input x,input y,input z);
   wire not_w;
   wire not_x;
   wire not_y;
   wire not_z;

   not(not_w,w);
   not(not_x,x);
   not(not_y,y);
   not(not_z,z);

   wire F1_A,F1_B,F1_C,F1_D;
   and(F1_A,w,not_x,not_y);
   and(F1_B,w,x,z);
   and(F1_C,not_w,x,y);
   and(F1_D,not_w,y,not_z);

   or(F1,F1_A,F1_B,F1_C,F1_D);

   wire F2_A,F2_B,F2_C,F2_D;
   and(F2_A,not_w,y,not_z);
   and(F2_B,x,y,z);
   and(F2_C,w,not_y,z);
   and(F2_D,w,not_x,not_y);

   or(F2,F2_A,F2_B,F2_C,F2_D);

   wire F3_A,F3_B,F3_C,F3_D,F3_E,F3_F;
   and(F3_A,w,not_x,not_y);
   and(F3_B,w,not_y,z);
   and(F3_C,w,x,z);
   and(F3_D,x,y,z);
   and(F3_E,not_w,x,y);
   and(F3_F,not_w,y,not_z);

   or(F3,F3_A,F3_B,F3_C,F3_D,F3_E,F3_F);
endmodule
```



證實化簡(F1)=化簡(F2)=化簡(F3)

第三個實驗

為 9 補數電路實作，利用真值表及卡諾圖化簡(有關 don’t care)

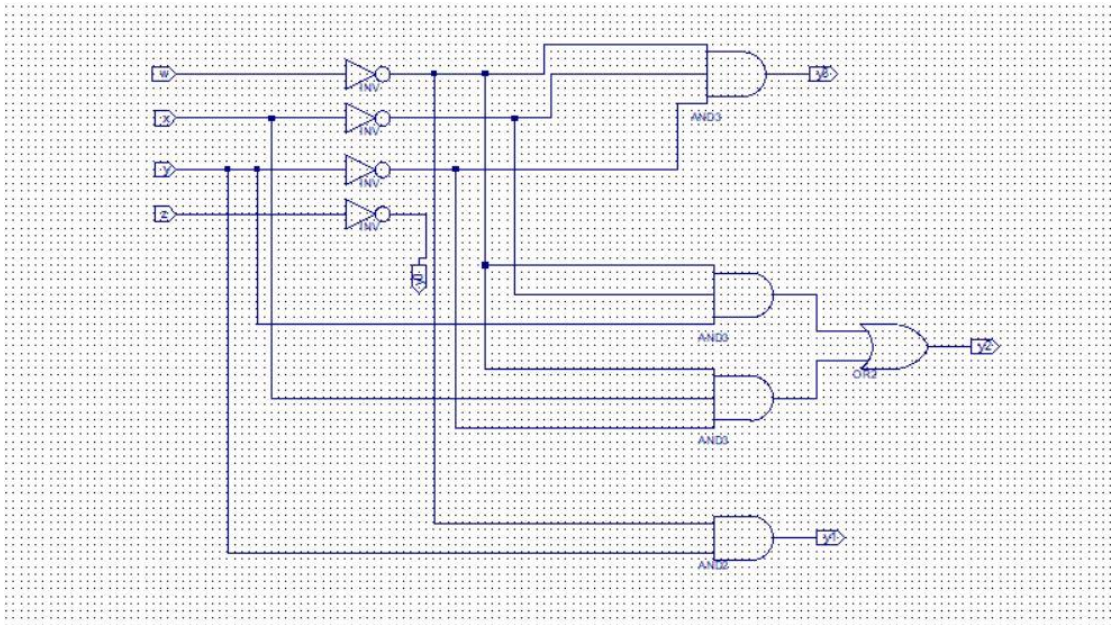


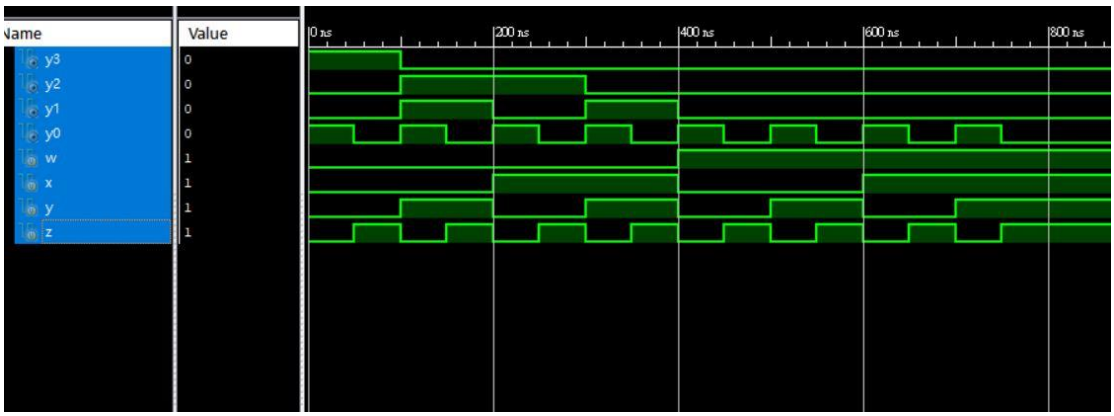
歸納出　y3=w’ x’ y’

y2=w’ xy’ +w’ x’ y

y1=w’ y

y0=z’ (有利用到 don’t care 原則)





經過檢查後，證實此 F 電路為 Input 9 補數電路