實驗主題(Lab05)

B083022053

黃啟桓

Exercise 1:

- Find and show the sum of minterms (*F2*) of *F1*(*w*,*x*,*y*,*z*) = *w'y'*+*w'z'*+*xz'*+*wy'z'*+*wy'z*
- Draw the Karnaugh map and find the simplest sum-of-products of *F2*
- If the simplest sum-of-products of *F1* is *F3*, verify *F3* = *F2* = *F1* (using Verilog Dataflow modeling)

Exercise 2:

- Draw the Karnaugh map and find all the simplest sum-of-products of $F(w,x,y,z) = \sum(2,6,7,8,9,13,15)$
- If the simplest sum-of-products of *F* are *F1*, *F2* and *F3*, verify *F1* = *F2* = *F3* (using Verilog Structural level modeling)

Exercise 3:

- Draw the Karnaugh map of $F1(v, w, x, y, z) = \sum (1, 2, 3, 6, 7, 9, 10, 11, 14, 15, 17, 18, 19, 22, 25, 26, 27, 30)$ and find the simplest sum-of-products of this Boolean function.
- If the simplest sum-of-products of *F1* is *F2*, verify *F1* = *F2* (using Verilog Structural level modeling)

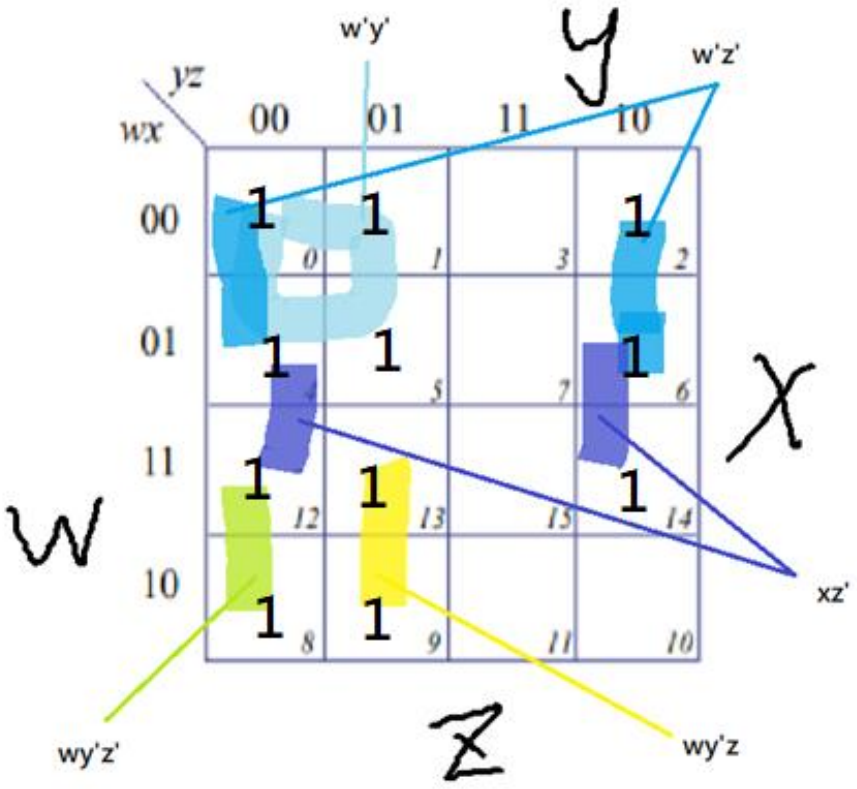
實驗內容

一、Exercise 1:

- Find and show the sum of minterms (*F2*) of *F1*(*w*,*x*,*y*,*z*) = *w'y'+w'z'+xz'+wy'z'+wy'z*

  F2 =$m_0+m_1+m_2+m_4+m_5+m_6+m_8+m_9+m_{12}+m_{13}+m_{14}$

- Draw the Karnaugh map and find the simplest sum-of-products of *F2*



  minterm: 取真值表為 1 的輸出

- If the simplest sum-of-products of *F1* is *F3*, verify *F3* = *F2* = *F1* (using Verilog Dataflow modeling)

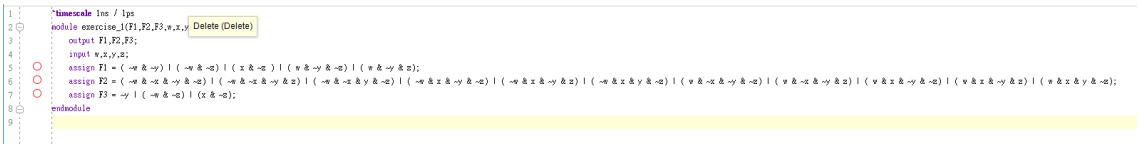
  F3 = y' + w'z' + xz'

$m_0+m_1+m_4+m_5+m_8+m_9+m_{12}+m_{13}$ = y'

$m_4+m_6+m_{12}+m_{14}$ = xz'

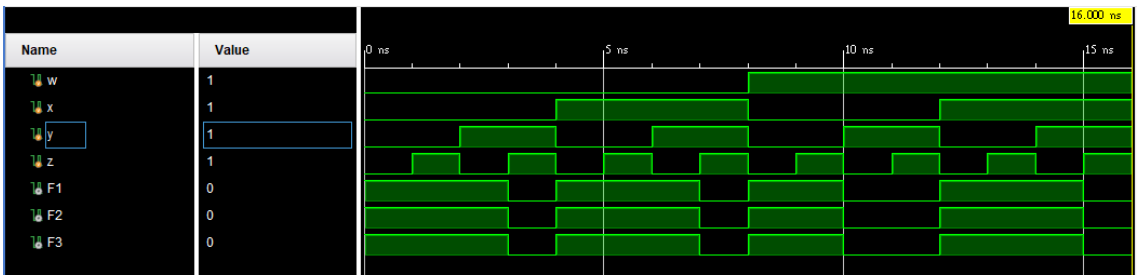
$m_0+m_2+m_4+m_6$ = w'z'

Dataflow modeling 程式如下:



放大後如下:

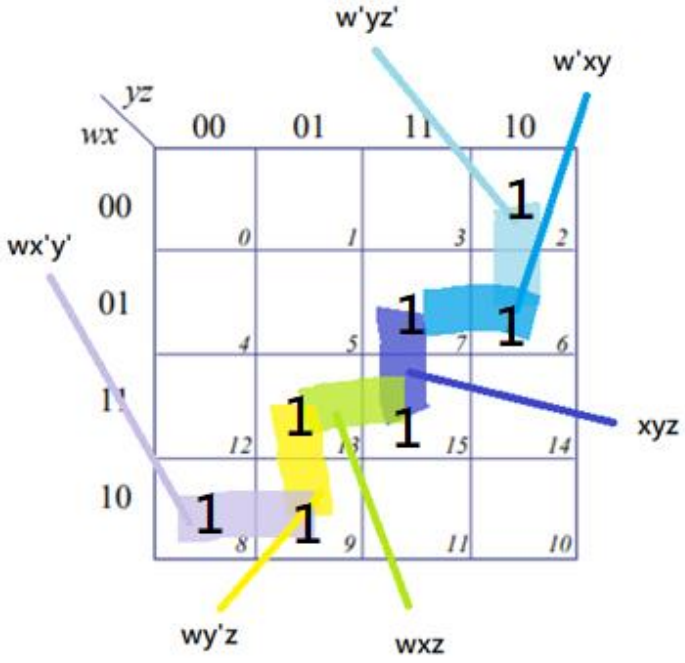
```
1.  * timescale 1ns / 1ps
2.
3.  module exercise_1 (F1,F2,F3,w,x,y,z);
4.      output Fl, F2, F3;
5.      input w, x, y, z;
6.      assign Fl = (~w & ~y)|(~w & ~z)|(x & ~z)|(w & ~y & ~z)|(w & ~y & z);
7.      assign F2 = (~w & ~x & ~y & ~z)|(~w & ~x & ~y & z)|(~w & ~x & y & ~z)|(~
    w & x & ~y & ~z)|(~w & x & ~y & z)|(~w & x & y & ~z)|(w & ~x & ~y & ~z)|(w &
     ~x & ~y & z)|(w & x & ~y & ~z)|(w & x & ~y & z)|(w & x & y & ~z);
8.      assign F3 = ~y |(~w & ~z)|(x & ~z);
9.
10. endmodule
```



電壓圖完全符合(上到下: 輸入:w,x,y,z 輸出:F1,F2,F3)

二、Exercise　2:

- Draw the Karnaugh map and find all the simplest sum-of-products of $F(w,x,y,z) = \sum(2,6,7,8,9,13,15)$



- If the simplest sum-of-products of *F* are *F1*, *F2* and *F3*, verify *F1* = *F2* = *F3* (using Verilog Structural level modeling)

已知 m_2 和 m_8 只有一種組合，即(m_2+m_6)與(m_8+m_9)，也就是說必須優先選擇 w'yz'和 wx'y'。只要 Karnaugh map 分得越小塊且多個重疊時，就會有越多組合。

F1 = $(m_2+m_6)+(m_8+m_9)+(m_6+m_7)+(m_{13}+m_{15})$
= w'yz' + wx'y' + w'xy + wxz
F2 = $(m_2+m_6)+(m_8+m_9)+(m_7+m_{15})+(m_9+m_{13})$
= w'yz' + wx'y' + xyz + wy'z
F3 = $(m_2+m_6)+(m_8+m_9)+(m_7+m_{15})+(m_{13}+m_{15})$
= w'yz' + wx'y' + xyz + wxz

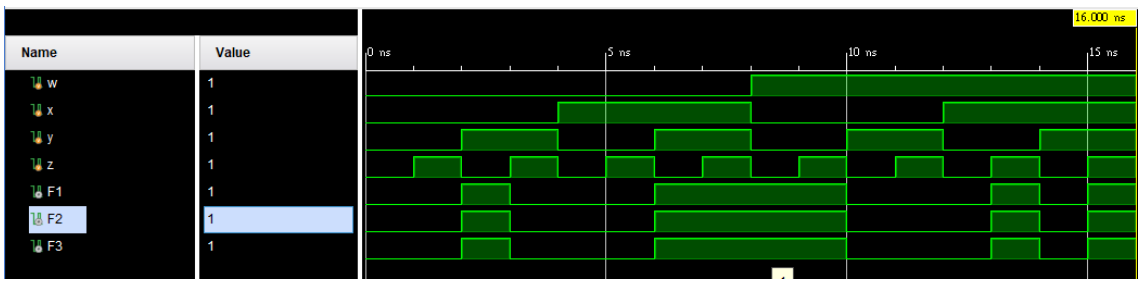
Structural level modeling 程式如下:

```
`timescale 1ns / 1ps
module exercise_2(F1,F2,F3,w,x,y,z);
    output F1,F2,F3;
    input w,x,y,z;
    wire n1,n2,n3,n4,n5,n6;
    wire notw,notx,noty,notz;

    not(notw,w);
    not(notx,x);
    not(noty,y);
    not(notz,z);

    and(n1,notw,y,notz);     //2,6
    and(n2,w,notx,noty);     //8,9
    and(n3,notw,x,y);        //6,7
    and(n4,w,x,z);           //13,15
    and(n5,x,y,z);           //7,15
    and(n6,w,noty,z);        //9,13

    or(F1,n1,n2,n3,n4); //2,6 8,9 6,7 13,15
    or(F2,n1,n2,n5,n6); //2,6 8,9 7,15 9,13
    or(F3,n1,n2,n4,n5); //2,6 8,9 7,15 13,15

endmodule
```

```
`timescale 1ns / 1ps
module exercise_2(F1,F2,F3,w,x,y,z);
    output F1,F2,F3;
    input w,x,y,z;
    wire n1,n2,n3,n4,n5,n6;

    and(n1,~w,y,~z);     //2,6
    and(n2,w,~x,~y);     //8,9
    and(n3,~w,x,y);      //6,7
    and(n4,w,x,z);       //13,15
    and(n5,x,y,z);       //7,15
    and(n6,w,~y,z);      //9,13

    or(F1,n1,n2,n3,n4); //2,6 8,9 6,7 13,15
    or(F2,n1,n2,n5,n6); //2,6 8,9 7,15 9,13
    or(F3,n1,n2,n4,n5); //2,6 8,9 7,15 13,15

endmodule
```

~x :為 x 經過 not 運算的值。為 not 邏輯閘的簡化 EXP: not(notx,x); (其中 notx 即等同於 ~x

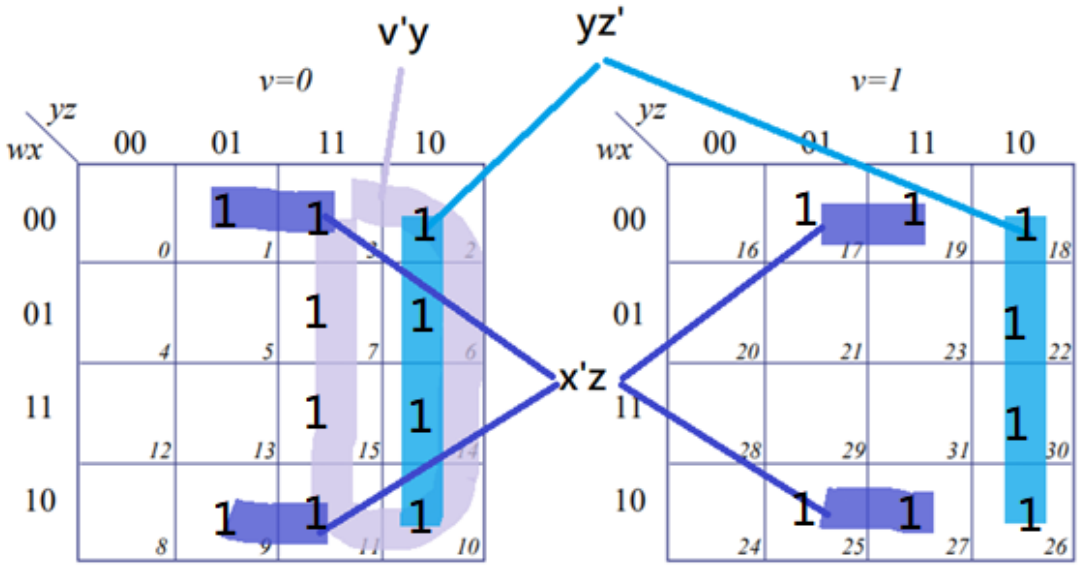
同理:and or 也可以如此，但是題目限定 Structural level modeling



電壓圖完全符合(上到下: 輸入:w,x,y,z 輸出:F1,F2,F3)

三、Exercise　3:

- Draw the Karnaugh map of *F1*(*v*, *w*, *x*, *y*, *z*) = ∑ (1, 2, 3, 6, 7, 9, 10, 11, 14, 15, 17, 18, 19, 22, 25, 26, 27, 30) and find the simplest sum-of-products of this Boolean function.

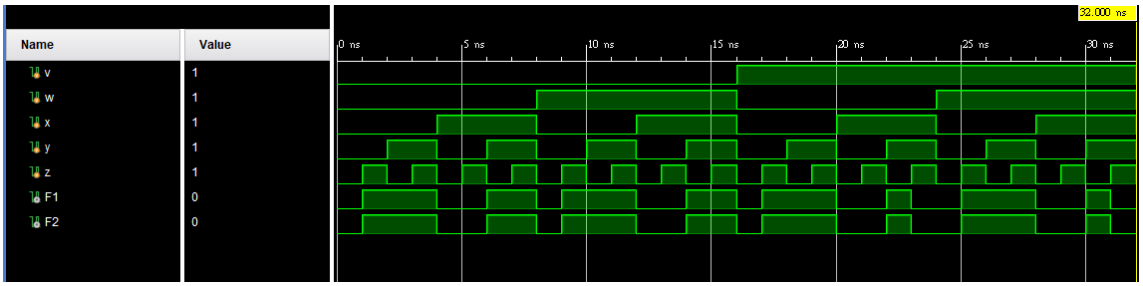


the simplest sum-of-products：要選最大塊的

the simplest sum-of-products (F2)= v'y + yz' + x'z

- If the simplest sum-of-products of *F1* is *F2*, verify *F1* = *F2* (using Verilog Structural level modeling)

```
`timescale 1ns / 1ps
module exercise_3(F1,F2,v,w,x,y,z);
    output F1,F2;
    input v,w,x,y,z;
    wire n1,n2,n3,n6,n7,n9,n10,n11,n14,n15,n17,n18,n19,n22,n25,n26,n27,n30;
    wire m1,m2,m3,m4;
    wire notv,notw,notx,noty,notz;

    not(notv,v);
    not(notw,w);
    not(notx,x);
    not(noty,y);
    not(notz,z);

    and (n1,notv,notw,notx,noty,z);
    and (n2,notv,notw,notx,y,notz);
    and (n3,notv,notw,notx,y,z);
    and (n6,notv,notw,x,y,notz);
    and (n7,notv,notw,x,y,z);
    and (n9,notv,w,notx,noty,z);
    and (n10,notv,w,notx,y,notz);
    and (n11,notv,w,notx,y,z);
    and (n14,notv,w,x,y,notz);
    and (n15,notv,w,x,y,z);
    and (n17,v,notw,notx,noty,z);
    and (n18,v,notw,notx,y,notz);
    and (n19,v,notw,notx,y,z);
    and (n22,v,notw,x,y,notz);
    and (n25,v,w,notx,noty,z);
    and (n26,v,w,notx,y,notz);
    and (n27,v,w,notx,y,z);
    and (n30,v,w,x,y,notz);
    or (F1,n1,n2,n3,n6,n7,n9,n10,n11,n14,n15,n17,n18,n19,n22,n25,n26,n27,n30);

    and (m1,notv,y);
    and (m2,y,notz);
    and (m3,notx,z);
    or (F2,m1,m2,m3);

endmodule
```



電壓圖完全符合(上到下: 輸入:v,w,x,y,z 輸出:F1,F2)

(三)

$F1 = (w \oplus x) \oplus (y \oplus z)$

$F2 = w \oplus (x \oplus (y \oplus z))$

$F3 = w \oplus x \oplus y \oplus z$

Verify $F1 = F2 = F3$

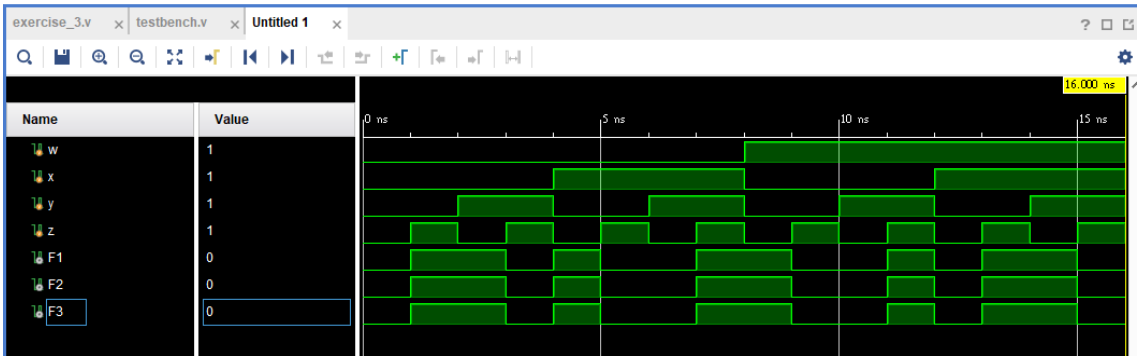
(using Structural level modeling)

程式碼如右:

wire ：為 Structural level modeling 非系統輸入也非系統輸出的線路宣告。

xor(n,a,b,c); ：為 xor 邏輯閘的函數，n (第一個變數)為輸出，僅能存在一個。而 a、b、c 為輸入 xor 邏輯閘的變數，幾個變數代表幾個輸入，且至少 2 輸入。

```
1   `timescale 1ns / 1ps
2
3   module exercise_3(F1,F2,F3,w,x,y,z);
4       output F1,F2,F3;
5       input w,x,y,z;
6       wire n1,n2;
7
8       xor (n1,w,x);
9       xor (n2,y,z);
10      xor (F1,n1,n2);
11
12      xor (n3,y,z);
13      xor (n4,x,n3);
14      xor (F2,w,n4);
15
16
17      xor (F3,w,x,y,z);
18
19
20
21  endmodule
22
```



由電壓圖 F1=F2=F3,可推得 xor 具有交換律和結合律

實驗心得

1. Structural level modeling 中邏輯閘使用函數表示，而 Dataflow modeling 中邏輯閘使用算式表示
2. Structural level modeling 可以很直觀的轉換成 Schematic
3. Structural level modeling 需要宣告邏輯閘之間的線路
4. Dataflow modeling 要特別注意優先權，尤其是程式越複雜時越容易出錯
5. 所有的 modeling 都需要注意該邏輯閘是否有交換律或結合律
6. 驗證等式是否成立時應將等式 2 邊同時輸出，比較容易對照電壓圖
7. minterm 和 maxterm 寫成程式碼太過攏長，雖然一定寫得出來，但太累了