

Exercise 1:

Find and show the **sum of minterms** ($F1$) of the following truth table

Find and show the **product of maxterms** ($F2$) of the following truth table

Verify $F1 = F2$

(using **Dataflow modeling**)

w	x	y	z	F
0	0	0	0	0
0	0	0	1	1
0	0	1	0	1
0	0	1	1	1
0	1	0	0	0
0	1	0	1	0
0	1	1	0	1
0	1	1	1	0
1	0	0	0	0
1	0	0	1	0
1	0	1	0	0
1	0	1	1	1
1	1	0	0	1
1	1	0	1	0
1	1	1	0	1
1	1	1	1	1

Exercise 2:

Derive $(x \downarrow y) \downarrow z = (x + y)z'$, $z \downarrow (y \downarrow x) = z'(y + x)$ and

$x \downarrow (y \downarrow z) = x'(y + z)$ (applying DeMorgan's theorem)

Verify $(x \downarrow y) \downarrow z = z \downarrow (y \downarrow x)$ (using **Structural level modeling**)

Verify $(x \downarrow y) \downarrow z \neq x \downarrow (y \downarrow z)$ (using **Structural level modeling**)

Exercise 3:

XOR is an **odd** function

Complete the truth table of four-variable XOR function

$F(w, x, y, z)$

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

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

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

Verify $F1 = F2 = F3$

(using **Structural level modeling**)

實驗內容

一、Exercise 1:

Find and show the **sum of minterms** ($F1$) of the following truth table

Find and show the **product of maxterms** ($F2$) of the following truth table

Verify $F1 = F2$

(using **Dataflow modeling**)

w	x	y	z	F
0	0	0	0	0
0	0	0	1	1
0	0	1	0	1
0	0	1	1	1
0	1	0	0	0
0	1	0	1	0
0	1	1	0	1
0	1	1	1	0
1	0	0	0	0
1	0	0	1	0
1	0	1	0	0
1	0	1	1	1
1	1	0	0	1
1	1	0	1	0
1	1	1	0	1
1	1	1	1	1

sum of minterms = $m_1 + m_2 + m_3 + m_6 + m_{11} + m_{12} + m_{14} + m_{15}$

product of maxterms = $M_0 * M_4 * M_5 * M_7 * M_8 * M_9 * M_{10} * M_{13}$

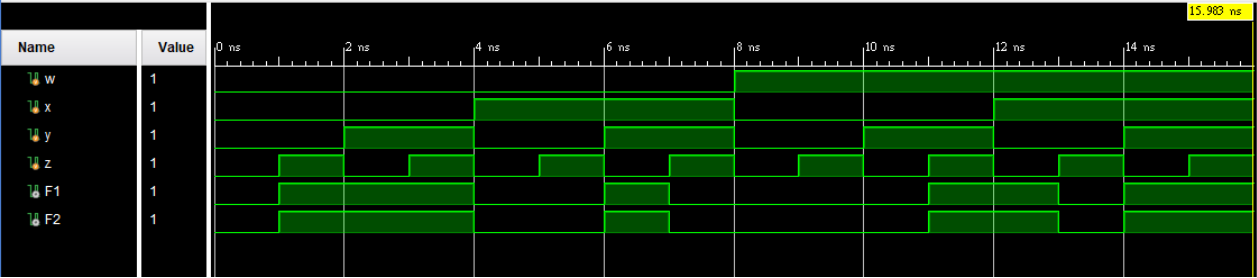
minterm: 取真值表為 1 的輸出

maxterm: 取真值表為 0 的輸出

Dataflow modeling 程式如下:

```
testbench.v x exercise_1.v x
C:/Users/ncpc/project_1/srcs/sources_1/new/exercise_1.v
1 timescale 1ns / 1ps
2
3 module exercise_1(F1,F2,w,x,y,z);
4     output F1,F2;
5     input w,x,y,z;
6     assign F1=(~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);
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);
8 endmodule
9
```

因為程式碼太寬文字略小



但電壓圖完全符合(上到下: 輸入:w,x,y,z 輸出:F1,F2)

二、Exercise 2:

Derive $(x \downarrow y) \downarrow z = (x + y)z'$, $z \downarrow (y \downarrow x) = z'(y + x)$ and
 $x \downarrow (y \downarrow z) = x'(y + z)$ (applying DeMorgan's theorem)

Verify $(x \downarrow y) \downarrow z = z \downarrow (y \downarrow x)$ (using Structural level modeling)

Verify $(x \downarrow y) \downarrow z \neq x \downarrow (y \downarrow z)$ (using Structural level modeling)

$$(一) : (x \downarrow y) \downarrow z = (x'y') \downarrow z = (x'y')' z' = (x+y)z'$$

$$: z \downarrow (y \downarrow x) = z \downarrow (y'x') = z' (y'x')' = z'(y+x)$$

$$: x \downarrow (y \downarrow z) = x \downarrow (y'z') = x' (y'z')' = x'(y+z)$$

(二) Verify $(x \downarrow y) \downarrow z = z \downarrow (y \downarrow x)$ (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。

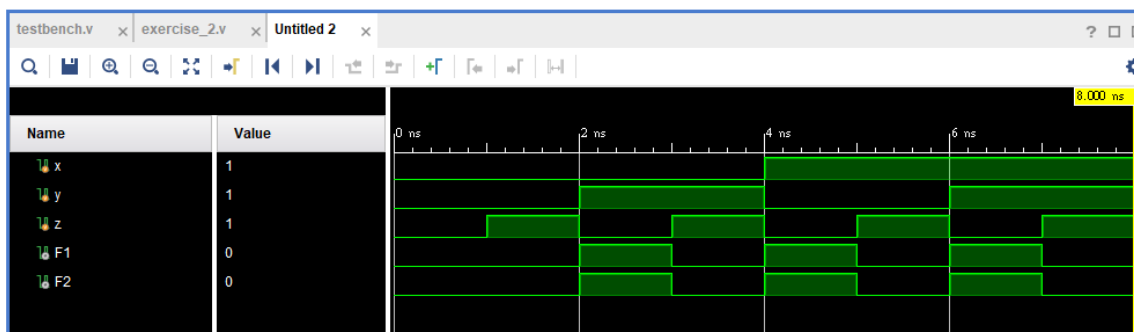
nor(n,a,b,c); : 為 nor 邏輯閘的函數，n (第一個變數)為輸出，僅能存在一個。而 a、b、c 為輸入 nor 邏輯閘的變數，幾個變數代表幾個輸入，且至少 2 輸入。

```
module exercise_2(F1,F2,x,y,z);
    output F1,F2;
    input x,y,z;
    wire n1,n2;

    nor (n1,x,y);
    nor (F1,n1,z);

    nor (n2,x,y);
    nor (F2,z,n2);

endmodule
```



由電壓圖 F1=F2,可推得 nor 具有交換律(上到下: 輸入:x,y,z 輸出:F1,F2)

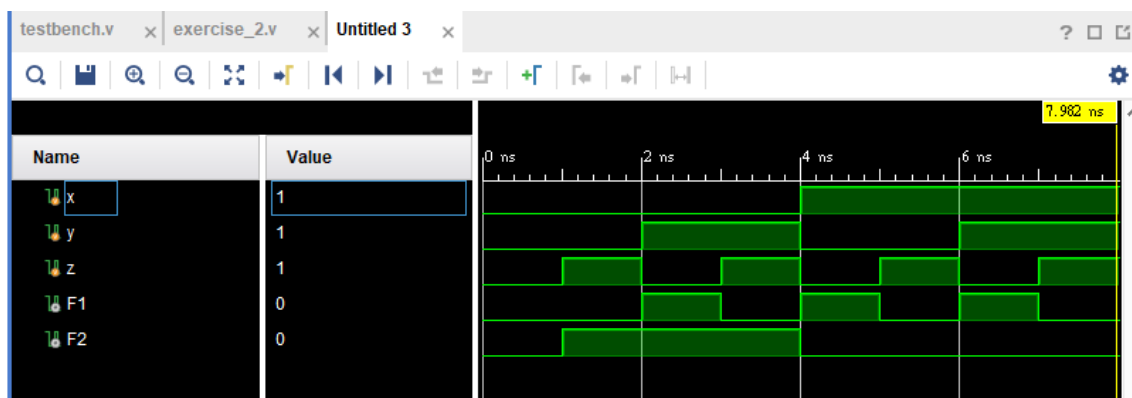
(三) Verify $(x \downarrow y) \downarrow z \neq x \downarrow (y \downarrow z)$ (using Structural level modeling),

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。

nor(n,a,b,c); : 為 nor 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 nor 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_2(F1,F2,x,y,z);
4          output F1,F2;
5          input x,y,z;
6          wire n1,n2;
7
8          nor (n1,x,y);
9          nor (F1,n1,z);
10
11         nor (n2,z,y);
12         nor (F2,x,n2);
13
14
15
16     endmodule
17
```



由電壓圖 $F1 \neq F2$, 可推得 nor 不具有結合律(上到下: 輸入:x,y,z 輸出:F1,F2)

三、Exercise 3:

(一) **XOR is an odd function** 意思是當輸入 n 個值時，其中 m 有 m 個值為 1 ($m \leq n$)。若 m 為奇數時，輸出 1，反之，則輸出 0。這段敘述中也能得知：XOR 有交換性、結合性，因為輸出結果與輸入值的順序無關

(二) Complete the truth table of four-variable XOR function $F(w, x, y, z)$

w	x	y	z	F
0	0	0	0	0
0	0	0	1	1
0	0	1	0	1
0	0	1	1	0
0	1	0	0	1
0	1	0	1	0
0	1	1	0	0
0	1	1	1	1
1	0	0	0	1
1	0	0	1	0
1	0	1	0	0
1	0	1	1	1
1	1	0	0	0
1	1	0	1	1
1	1	1	0	1
1	1	1	1	0

(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 $F1 = F2 = F3$

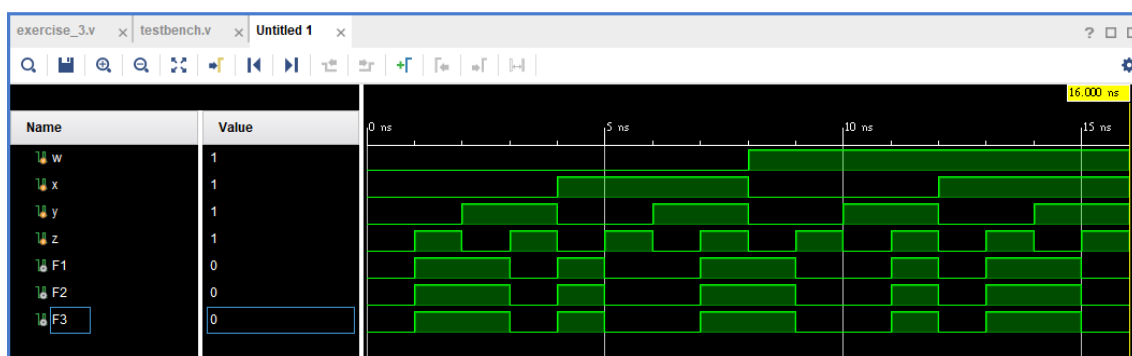
(using Structural level modeling)

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