

Lab07

實驗主題: Karnaugh Map

實驗日期:2022/10/17

學號姓名:B103040009 尹信淳

實驗內容: Design and Verify the following circuits using Verilog HDL and Schematic

Exercise1

■ Derivation:

$$F = (AB' + A'B)(C + D')$$

F1->AND-OR gates implementation

F2->NAND gates implementation

F3->NOR gates implementation

■ Verification:

Verify $F1=F2=F3$

■ Code:

```
module exercisel(F1,F2,F3,a,b,c,d);
output F1,F2,F3;
input a,b,c,d;
wire n1,n2,n3,n4,n5,n6,n7,n8,n9,n10,n11,n12,n13,n14,n15,n16,n17;
not(n1,a);
not(n2,b);
not(n3,c);
not(n4,d);
and(n5,a,n2);
and(n6,n1,b);
or(n7,n5,n6);
or(n8,c,n4);
and(F1,n7,n8);
//////////
nand(n9,a,n2);
nand(n10,n1,b);
nand(n11,n9,n10);
nand(n12,n3,d);
nand(n13,n11,n12);
nand(F2,F2,n13);
//////////
nor(n14,n1,b);
nor(n15,a,n2);
nor(n16,n14,n15);
nor(n17,c,n4);
nor(F3,n17,n16);

endmodule
```

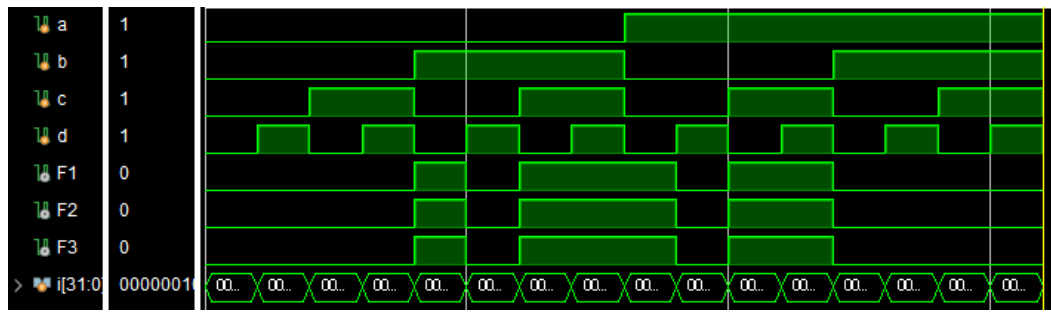
■ Testbench:

```
module tbt();
reg a,b,c,d;
wire F1,F2,F3;
integer i;

exercisel ul(.a(a), .b(b), .c(c), .d(d), .F1(F1), .F2(F2), .F3(F3));
initial begin

{a,b,c,d} = 0;
for(i = 0; i < 16; i = i + 1) begin
{a,b,c,d} = i;
#10;
end
$finish;
end
endmodule
```

■ 波型



■ Exercise 1 實驗結果與分析:

這些 implementations 有相同功能。

Exercise2

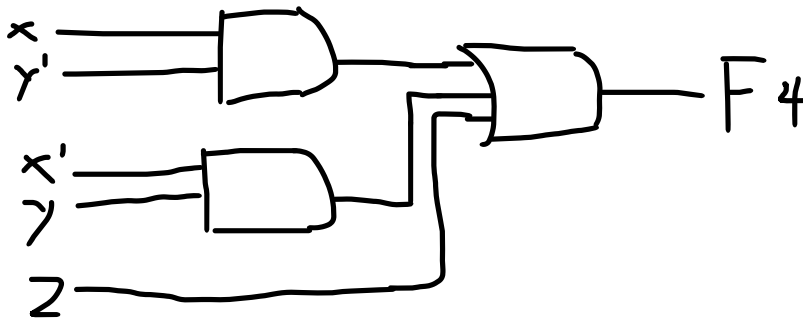
■ Simplify the following Boolean function using Karnaugh maps:

$$F(x,y,z) = S(1,2,3,4,5,7)$$

F	yz	00	01	11	10
x					
0		0	1	1	1
1		1	1	1	0

Simplest sum of products: $z+xy'+x'y$

Draw its diagram:



■ Derivation:

$$F4 = z+xy'+x'y$$

F5 不太需要推導

$$F6 = [(z+xy'+x'y)']' = [z'(x'+y)(x+y)']'$$

■ Verification:

Verify that $F4=F5=F6$

■ Code:

```
module exercise2(F4,F5,F6,x,y,z);
output F4,F5,F6;
input x,y,z;
wire n1,n2,n3,n4,n5,n6,n7,n8,n9,n10;
not(n1,x);
not(n2,y);
not(n3,z);
and(n4,x,n2);
and(n5,n1,y);
or(F4,z,n5,n4);
//////////
nand(n6,x,n2);
nand(n7,n1,y);
nand(F5,n3,n6,n7);
//////////
or(n8,x,n2);
or(n9,n1,y);
and(n10,n3,n8,n9);
not(F6,n10);

endmodule
```

■ Testbench:

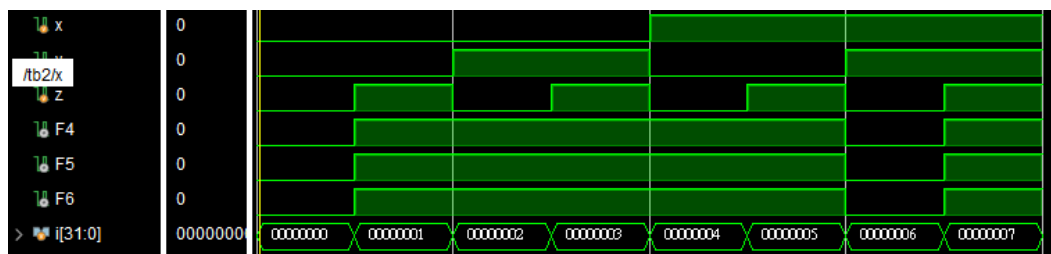
```
module tb2();
reg x, y, z;
wire F4,F5,F6;
integer i;

exercise2 u1(.x(x), .y(y), .z(z), .F4(F4),.F5(F5),.F6(F6));
initial begin

{x, y, z} = 0;
for(i = 0;i<8 ;i = i+1)begin
{x,y,z} = i;
#10;
end

$finish;
end
endmodule
```

■ 波型:



■ Exercise 2 實驗結果與分析:

這些 implementations 有相同功能。

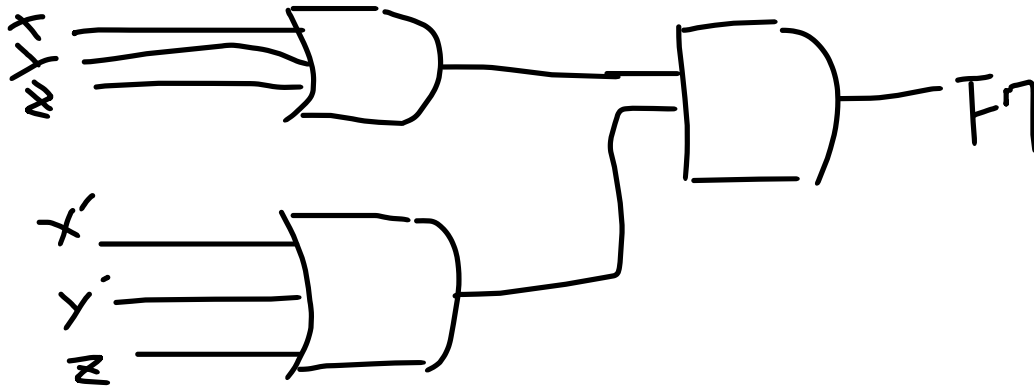
Exercise3

- Simplify the following Boolean function using Karnaugh maps: $F(x,y,z) = \sum (1,2,3,4,5,7)$

同 Exercise2

Simplest product of sums: $(x+y+z)(x'+y'+z)$

Draw its diagram:



- Derivation:

$$F7 = (x+y+z)(x'+y'+z)$$

F8 略

$$F9 = \{[(x+y+z)(x'+y'+z)]'\}' = (x'y'z' + xyz)'$$

- Code:

```
module exercise3(F7,F8,F9,x,y,z);
output F7,F8,F9;
input x,y,z;
wire n1,n2,n3,n4,n5,n6,n7,n8,n9,n10;
not(n1,x);
not(n2,y);
not(n3,z);
or(n4,x,y,z);
or(n5,n1,n2,z);
and(F7,n4,n5);
////////////////////////////////
nor(n6,x,y,z);
nor(n7,n1,n2,z);
nor(F8,n6,n7);
////////////////////////////////
and(n8,n1,n2,n3);
and(n9,x,y,n3);
or(n10,n8,n9);
not(F9,n10);
endmodule
```

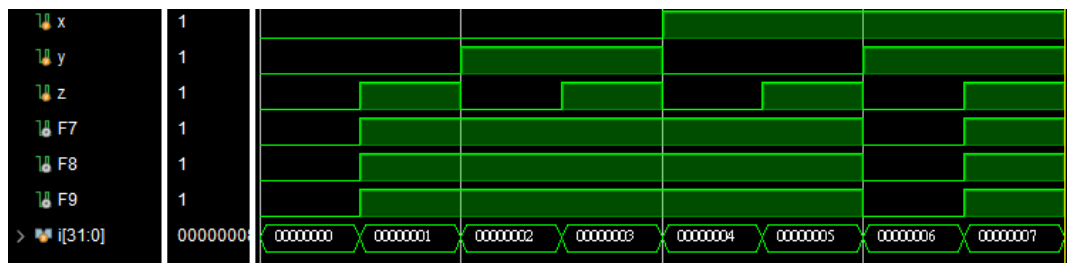
■ Testbench:

```
module tb3();
  reg x, y, z;
  wire F7,F8,F9;
  integer i;

  exercise3 u1(.x(x), .y(y), .z(z), .F7(F7),.F8(F8),.F9(F9));
  initial begin

    {x, y, z} = 0;
    for(i = 0;i<8 ;i = i+1)begin
      {x,y,z} = i;
      #10;
    end
    $finish;
  end
endmodule
```

■ 波型:



■ Exercise 3 實驗結果與分析:

這些 implementations 有相同功能。

實驗心得:

這次的三題練習了各種 boolean function 的 implementation。

第一題先是練習了 AND-OR 再練習把 AND-OR 的邏輯閘換成全部 NAND 跟全部 NOR。有些邏輯閘可以一起換成 NAND 或一起換成 NOR；剩下的邏輯閘再各自轉換。必要時要將輸入取 complement 或是在輸出加上 NOT。

第二題需要先做卡諾圖化簡，化簡之後得到 SOP (F4)並畫出其電路圖；F5 是將 F4 換成全部 NAND 閘的版本，轉換方式在第一題已經練習過（由於已經是 SOP，因此單純把所有邏輯閘都改成 NAND 就行）。至於 F6 要求的 OAI 版本則可以把 F4 取兩次反向，較內的反向用迪摩根展開，外面的反向留著，即可取得 F6。

第三題的卡諾圖與第二題相同，差別在於要先取得其 POS (F7)，POS 取法：取得原式反向後的 SOP 後再取反向即可得到原式的 POS。F8 為 F7 的 NOR 版本（由於已經有 SOP，因此把邏輯閘全部換成 NOR 即可）。F9 則是把 F7 取兩次反向，將內層反向展開，外層留下，即可取得 AOI 版本。

這次的實驗讓我更熟悉了這些 implementation 的轉換，頗有收穫。