

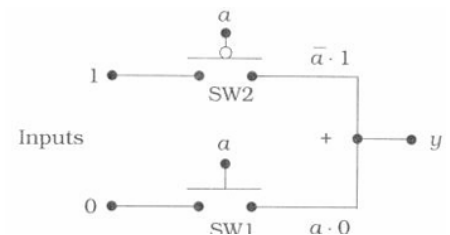
學號：111062307 姓名：陳大佑

一、 How to perform the simulation

我使用了 inverter、OR、AND 三個邏輯閘來完成，其中 OR 跟 AND 是用 NOR+inverter 以及 NAND+inverter 來完成，最後在化簡原本的 f ，從原本的 $(d \& \sim(b \mid \sim c)) \mid (c \& (\sim b \mid a))$ 得到了 $c \& (\sim b \mid a)$ ，然後再用做出來的 and or inverter 完成實作。

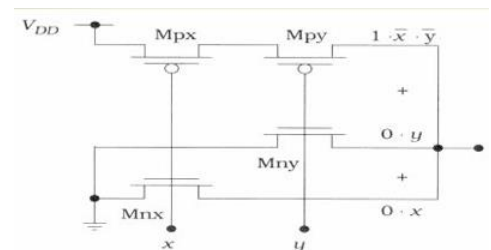
最開始的時候做出了 inverter，使用了 1 個 pmos 及 1 個 nmos 完成

```
*** Inverter ***
.subckt INV in1 inv_out vdd vss
** Your code **
mp1 inv_out in1 vdd vdd P_18 w=0.5u l=0.18u
mn1 inv_out in1 vss gnd N_18 w=0.25u l=0.18u
.ends
```



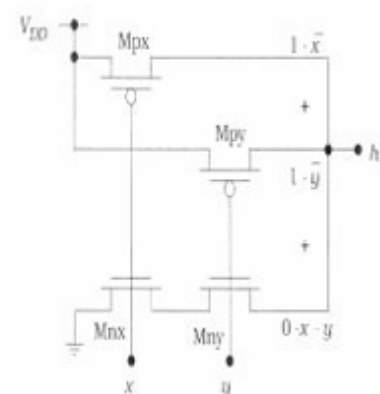
再來實作 NOR 使用了 2 個 pmos 及 2 個 nmos，為了方便辨別，將其標為語課本相同的變數名稱，而 Mpx 傳到 Mpy 那段導線則為 out。

```
*** NOR ***
.subckt NOR2 x y NOR vdd vss
** Your code **
mpx1 out x vdd vdd P_18 w=0.5u l=0.18u
mpy1 NOR y out vdd P_18 w=0.5u l=0.18u
mnx1 NOR x vss gnd N_18 w=0.25u l=0.18u
mny1 NOR y vss gnd N_18 w=0.25u l=0.18u
.ends
```



接著實作 NAND，共使用了 2 個 pmos 及 2 個 nmos，為了方便辨別，一樣標為課本變數名稱，其中 out1 是 Mnx 傳到 Mny 那一段導線

```
*** NAND ***
.subckt NAND2 x y NAND vdd vss
** Your code **
mpx2 NAND x vdd vdd P_18 w=0.5u l=0.18u
mpy2 NAND y vdd vdd P_18 w=0.5u l=0.18u
mnx2 out1 x vss gnd N_18 w=0.25u l=0.18u
mny2 NAND y out1 gnd N_18 w=0.25u l=0.18u
.ends
```



完成 NOR、NAND 後，開始實作 OR 跟 AND。

OR 只需要用 NOR + inverter 即完成、AND 只需要 NAND + inverter 即完成。

```
*** OR ***
.subckt OR2 in2 in3 OR vdd vss
** Your code **
Xnor1 in2 in3 out_nor vdd vss NOR2
Xinv1 out_nor OR vdd vss INV
.ends
```

```
*** AND ***
.subckt AND2 in4 in5 AND vdd vss
** Your code **
Xnand1 in4 in5 out_nand vdd vss NAND2
Xinv2 out_nand AND vdd vss INV
.ends
```

最後將邏輯閘合在一起，拼出 $c \& (\sim b \mid a)$ ，並且供給適當電壓，即可得出 f 。

```
*** logic function ***
.subckt logic A B C D F vdd vss
** Your code **
Xinv3 B negB vdd vss INV
Xor A negB out vdd vss OR2
Xand C out F vdd vss AND2
.ends

Xlogic A B C D F vdd gnd logic
```

```
**** vlotage source setting ***
*** Vname node1 node2 pulse (V1 V2 delay time_rise
Vin1 A gnd pulse(1.8v 0v 0 0.1n 0.1n 64n 128n)
Vin2 B gnd pulse(1.8v 0v 0 0.1n 0.1n 32n 64n)
Vin3 C gnd pulse(1.8v 0v 0 0.1n 0.1n 16n 32n)
Vin4 D gnd pulse(1.8v 0v 0 0.1n 0.1n 8n 16n)
```

二、The completion of the assignment.

(0, 0, 0, 0) -> 0、(0, 0, 0, 1) -> 0、(0, 0, 1, 0) -> 1、(0, 0, 1, 1) -> 1

(0, 1, 0, 0) -> 0、(0, 1, 0, 1) -> 0、(0, 1, 1, 0) -> 0、(0, 1, 1, 1) -> 0

(1, 0, 0, 0) -> 0、(1, 0, 0, 1) -> 0、(1, 0, 1, 0) -> 1、(1, 0, 1, 1) -> 1

(1, 1, 0, 0) -> 0、(1, 1, 0, 1) -> 0、(1, 1, 1, 0) -> 1、(1, 1, 1, 1) -> 1 (需求皆有完成)

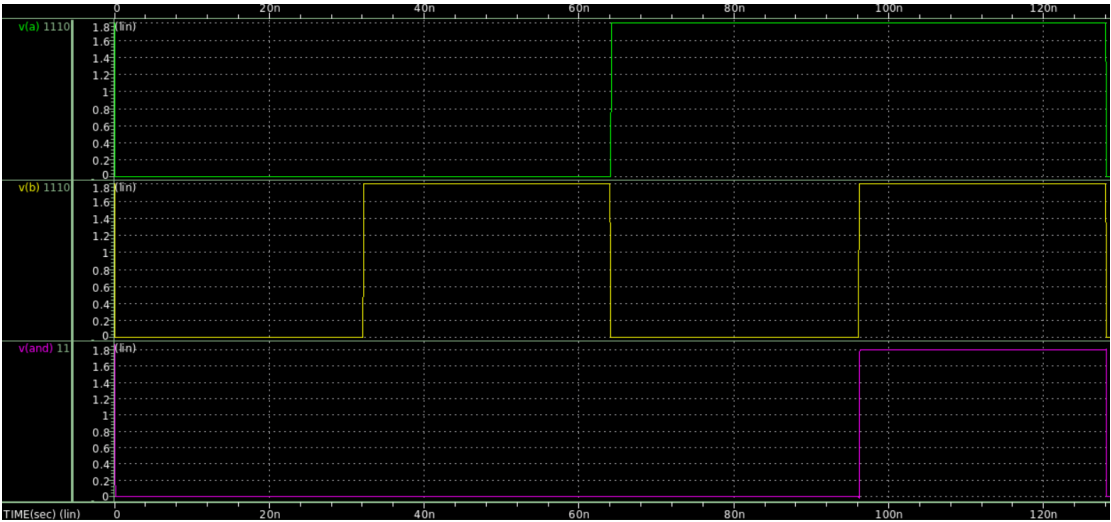
```
int d = 0; d <= 1; d++;
int f1 = -1;
int f2 = -1;
f1 = (d & ~(b | ~c)) | (c & (~b | a));
f2 = c & (~b | a);
cout << a << " " << b << " " << c << " " << d << " f1:" << f1 << " f2:" << f2
cout << endl;
```

三、The waveform of OR gate, AND gate, inverter, and the specific logic function for every combination of inputs

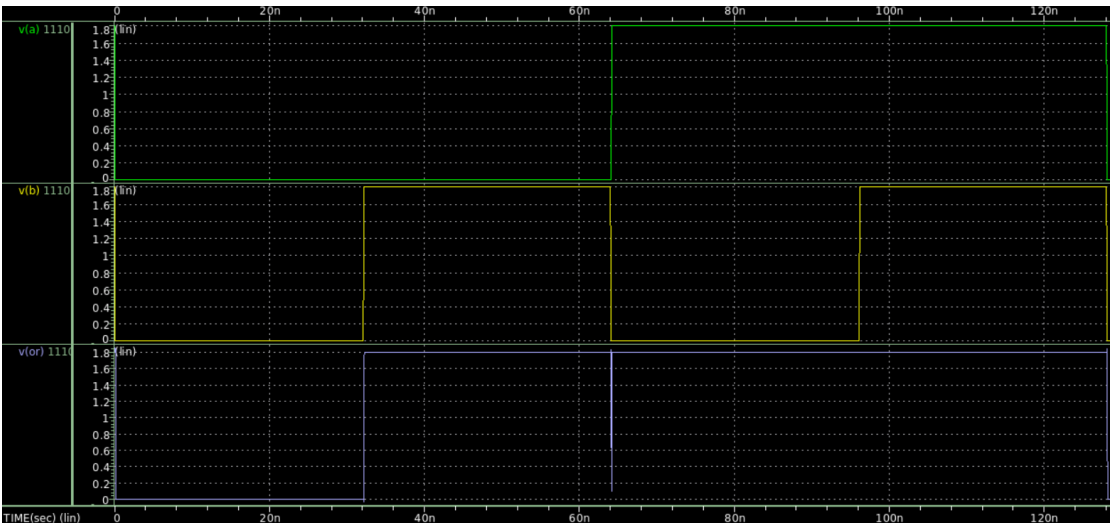
inverter waveform



AND waveform



OR waveform



specific logic function waveform



四、The hardness of this assignment and how you overcame it.

最難的地方就是 debug 的部分，因為對這個 code 不太熟，以為.end 跟.ends 是一樣的，找這個 bug 找了我 3 小時，這個 bug 他在 waveform 那邊才會出錯，前面的 lis 偵測不到，最後是受不了了重打才發現到這個錯誤。。。

五、Any suggestions about this homework?

助教在 tutorial 的 parameter 名稱跟給我們 code 的 parameter 名稱不一樣，感覺可以一樣，會比較容易懂。然後助教的影片有點模糊，就一點點。

助教回覆速度很快，十分感人，十分感謝，希望未來還能繼續！

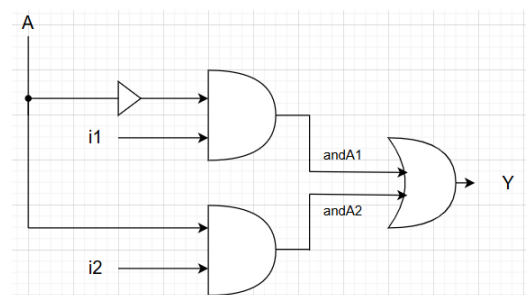
然後感覺可以在 reference 那邊，也放上幾部關於 Hspice 的影片，可以更加清楚這個 code 到底在幹嘛，感謝教授與助教。

六、bonus version

我使用前面用到的 3 個邏輯閘(inverter, OR, AND)。

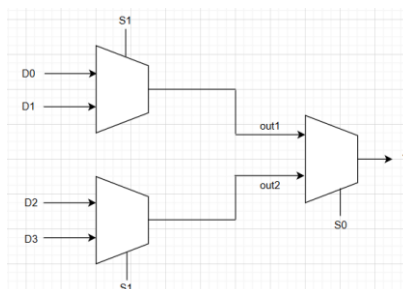
接下來做出 2to1 Mux，如旁邊 diagram

```
*** 2 to 1 MUX ***
.subckt MUX2 A i1 i2 Y vdd vss
** Your code **
Xinv A      invA      vdd vss INV
Xand1 invA i1      andA1 vdd vss AND2
Xand2 A     i2      andA2 vdd vss AND2
Xor  andA1 andA2 Y    vdd vss OR2
.ends
```

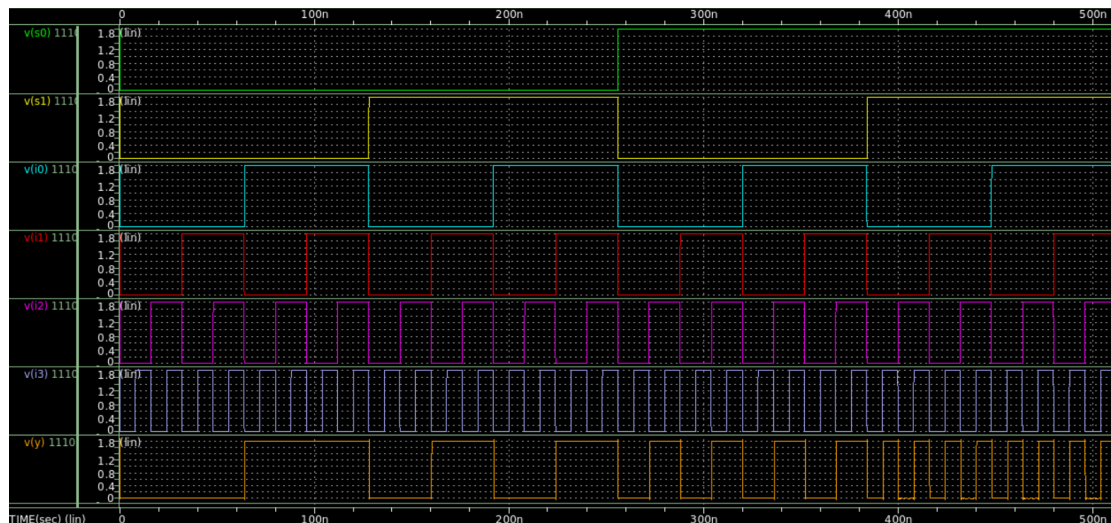


再根據 ppt 要求用 3 個 2to1 Mux 做出一個 4to1 Mux，如旁邊 diagram

```
*** 4 to 1 MUX ***
.subckt MUX S0 S1 D0 D1 D2 D3 Y vdd vss
Xmux1 S1 D0 D1 out1 vdd vss MUX2
Xmux2 S1 D2 D3 out2 vdd vss MUX2
Xmux3 S0 out1 out2 Y vdd vss MUX2
** Your code **
.ends
```



最後附上 waveform 。

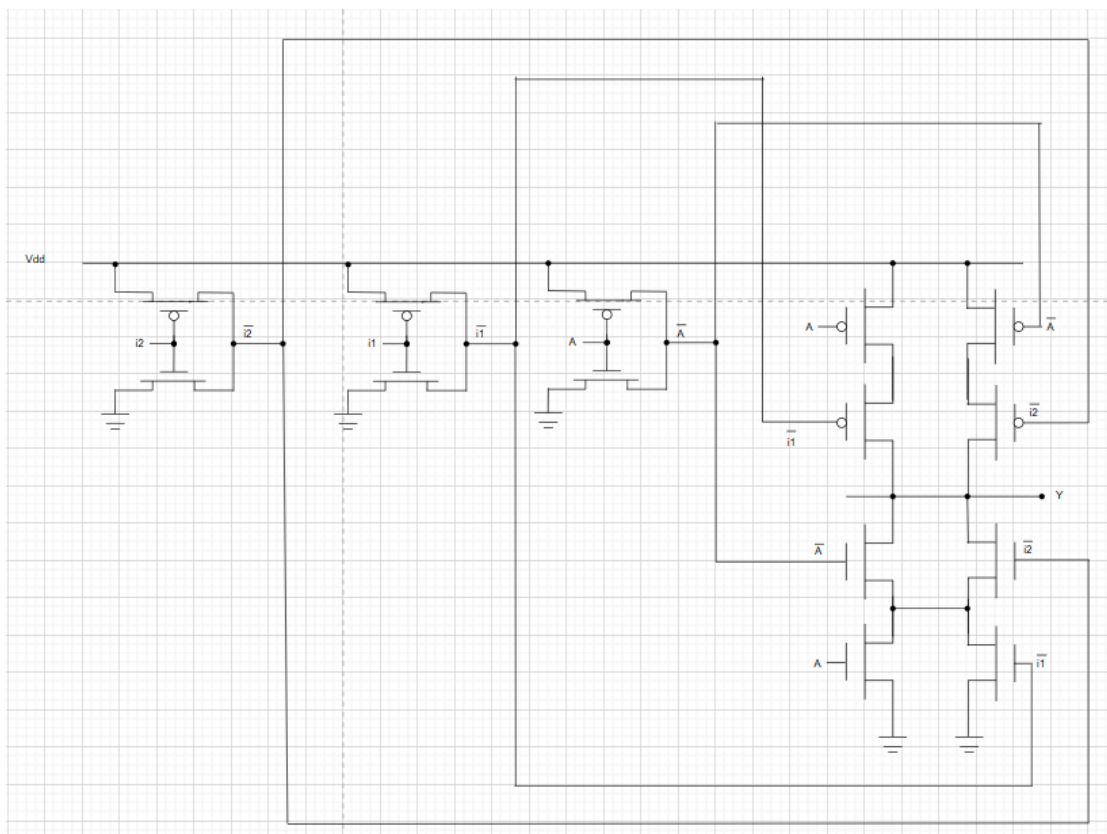


當 $S_0 = 0, S_1 = 0$ 的時候，output(y)的結果會跟 $D_0 (i_0)$ 一模一樣。

當 $S_0 = 0, S_1 = 1$ 的時候，output(y)的結果會跟 $D_1 (i_1)$ 一模一樣。

當 $S_0 = 1, S_1 = 0$ 的時候，output(y)的結果會跟 $D_2 (i_2)$ 一模一樣。

當 $S_0 = 1, S_1 = 1$ 的時候，output(y)的結果會跟 $D_3 (i_3)$ 一模一樣。



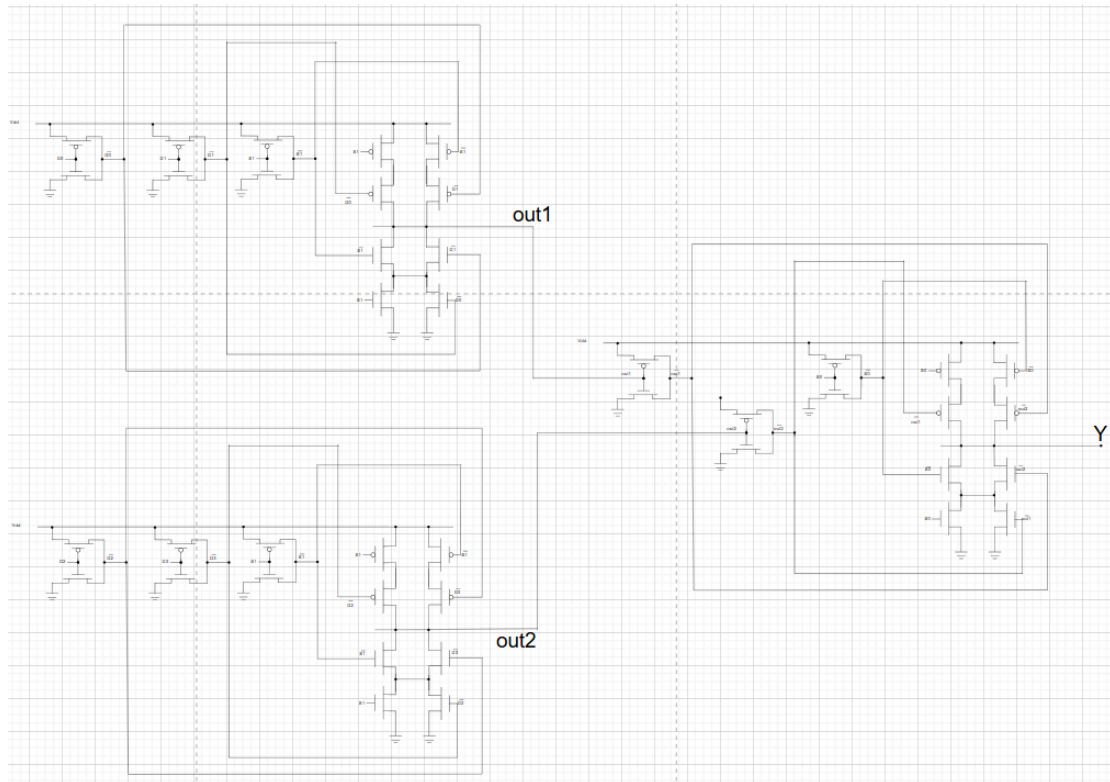
然後附圖為 transistor-base 2to1 mux 圖，

因為得知 2 to 1 Mux 的 Y 會 $= A'i1 + Ai2$

所以 $Y = ((A+i1')(A'i2'))'$ ，

再根據 $Y = ((A+i1')(A'i2'))'$ ，可以畫 nmos 圖

然後再用對稱畫出 pmos 圖



而上面的則是 4to1 mux 用 3 個 2to1 mux 所畫出來的 transition diagram。