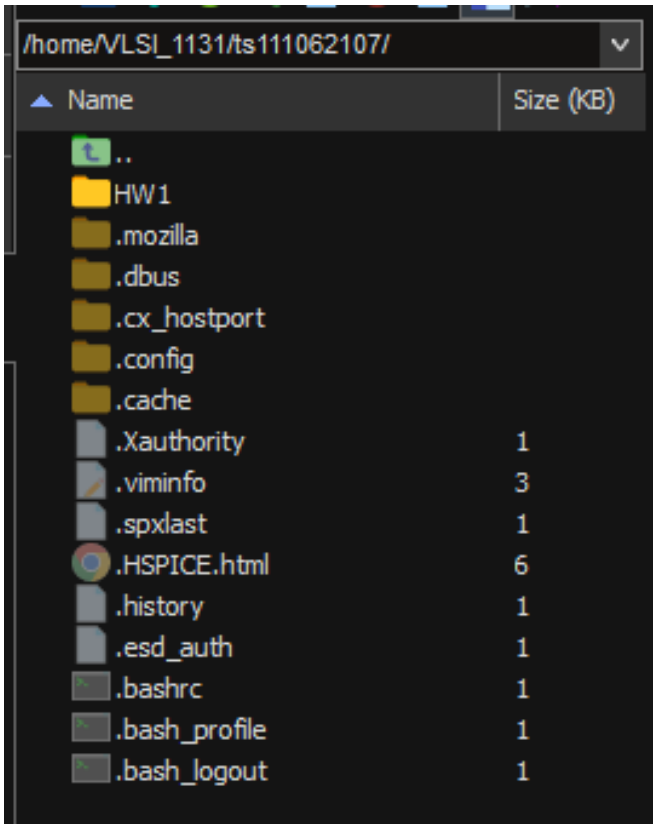
HW1 Report

ID：111062107

姓名：鄧弘利

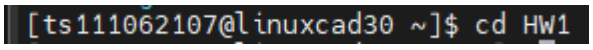
1. How to perform the simulation. (You can use a screenshot to explain)

1. mkdir HW1



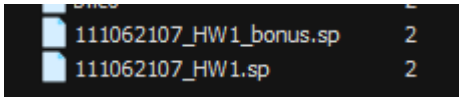
首先我先在我的學號資料夾下面創建了”HW1”資料夾。

2. cd HW1 或是 直接點擊旁邊的 HW1 資料夾 來進入 HW1 資料夾。



3. 進入之後，再透過上傳的按鈕將我的兩個 homework sp 檔案 template 放進

去。



4. 接下來將檔案寫完之後就可以跑模擬了。

按順序輸入以下指令，

source /usr/cad/synopsys/CIC/hspice.cshrc

hspice –i 111062107_HW1.sp -o a.lis

(for basic part)

hspice –i 111062107_HW1_bonus.sp -o a.lis

(for advanced part)

```
[ts111062107@linuxcad30 ~/HW1]$ source /usr/cad/synopsys/CIC/hspice.cshrc
set hspice version: 2020.12 (default)
[ts111062107@linuxcad30 ~/HW1]$ hspice -i 111062107_HW1.sp -o a.lis
Using: /home/tools/synopsys/hspice/2020.12/hspice/linux64/hspice  -i '111062107_HW1.sp' -o a.lis

>info:         ***** hspice job concluded
```

(當跑完”hspice –i 111062107_HW1.sp -o a.lis” 指令之後 最底下顯示 job concluded 就表示有成功，如果是顯示 aborted 就是有 errer，可以透過指令 vim a.lis 去看是哪邊發生 error)

接下來是看波型的部分，

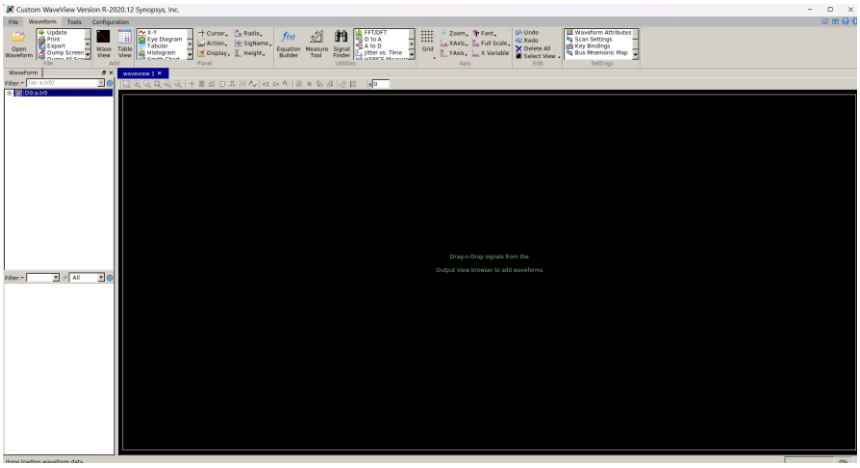
source /usr/cad/synopsys/CIC/customexplorer.cshrc

wv a.tr0 &

```
[ts111062107@linuxcad30 ~/HW1]$ source /usr/cad/synopsys/CIC/customexplorer.cshrc
set customexplorer version: 2020.12 (default)
[ts111062107@linuxcad30 ~/HW1]$ wv a.tr0 &
[1] 15472
```

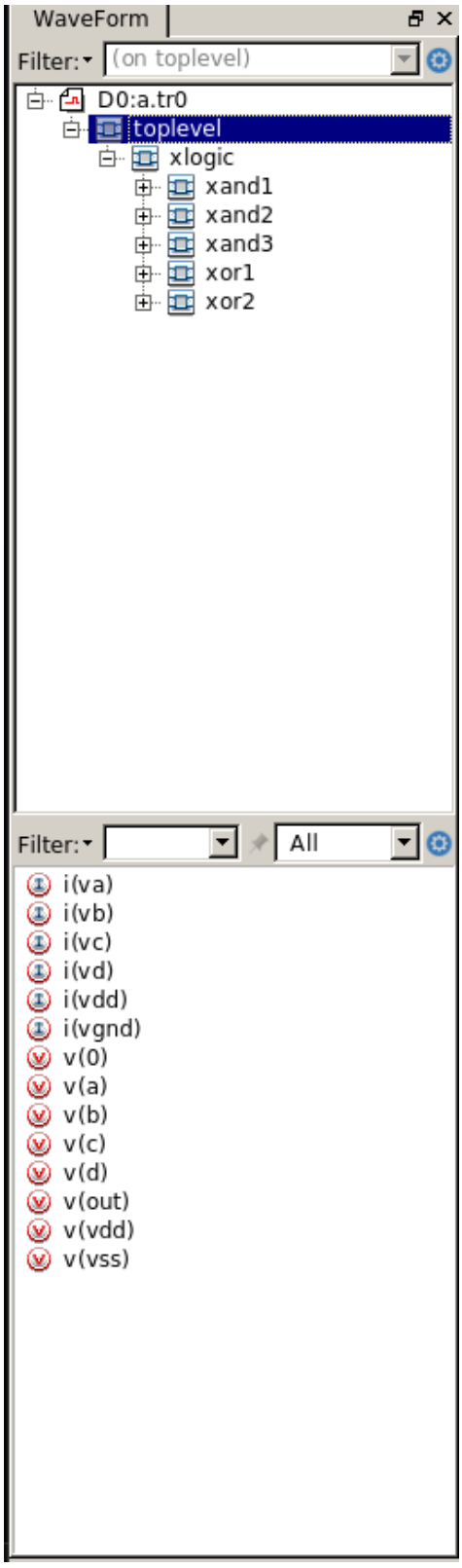
會需要花些時間才會開啟看波型的程式。

5. 看波型



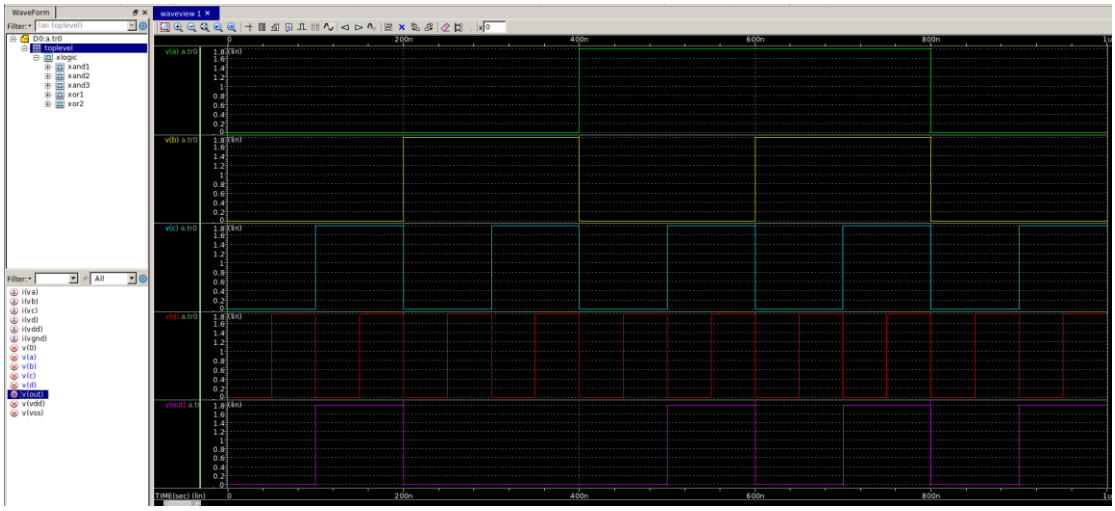
當開啟成功之後，畫面會如圖所示。

我們可以將左邊的檔案點開，會展開出底下的小電路。以這張圖為例，就是先



將 D0:a.tr0 旁邊的+號按鈕按下後，就會出現 toplevel，再點 toplevel 旁邊的+號按鈕就會再出現他底下用到的子電路，以此類推。

底下的那些 i(va)等等的就是對應的訊號名稱，我們接著可以將他們透過拉動到右邊的面板或是點擊等方式來將他們加入到右邊的面板以讓我們觀察。



我依序將 a, b, c, d(logic function 的輸入訊號) 還有 out(function 的輸出)拉到畫面後，我們就可以看到他們的波型。

2. The completion of the assignment. (If you complete all requirements, just specify all)

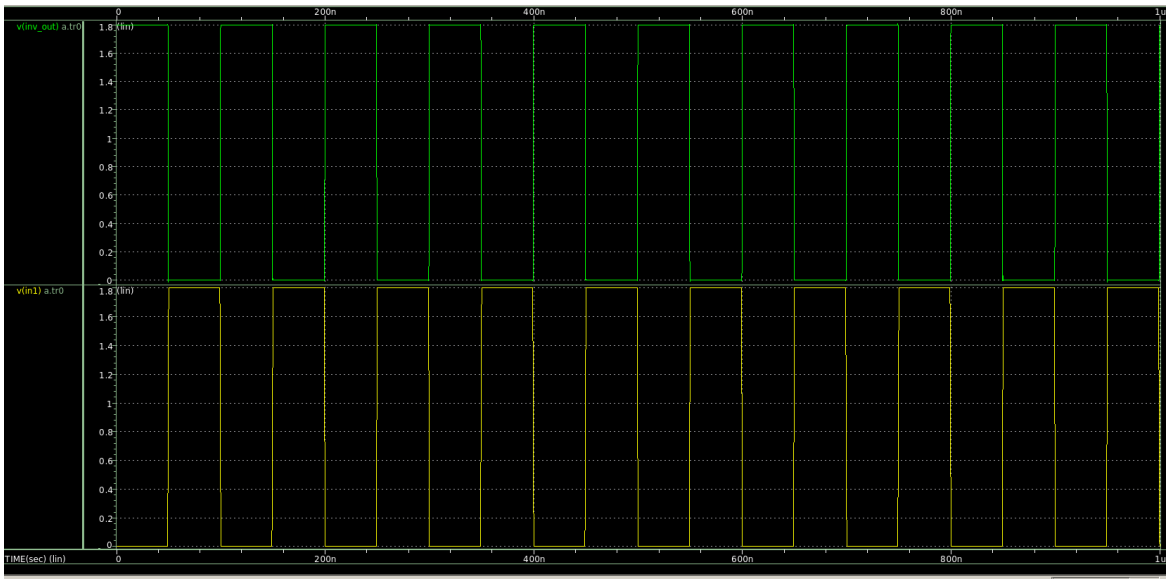
I complete all requirements.

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

1. Inverter

```
.subckt INV in1 inv_out vdd vss
mp1     vdd     in1     inv_out     vdd     p_18     w=0.5u     l=0.18u
mn1     inv_out     in1     vss     vss     n_18     w=0.25u     l=0.18u
.ends
```

```
XtestInv in1 inv_out vdd vss INV
V1 in1 vss PULSE(0v 1.8v 49ns 1ns 1ns 49ns 100ns)
```

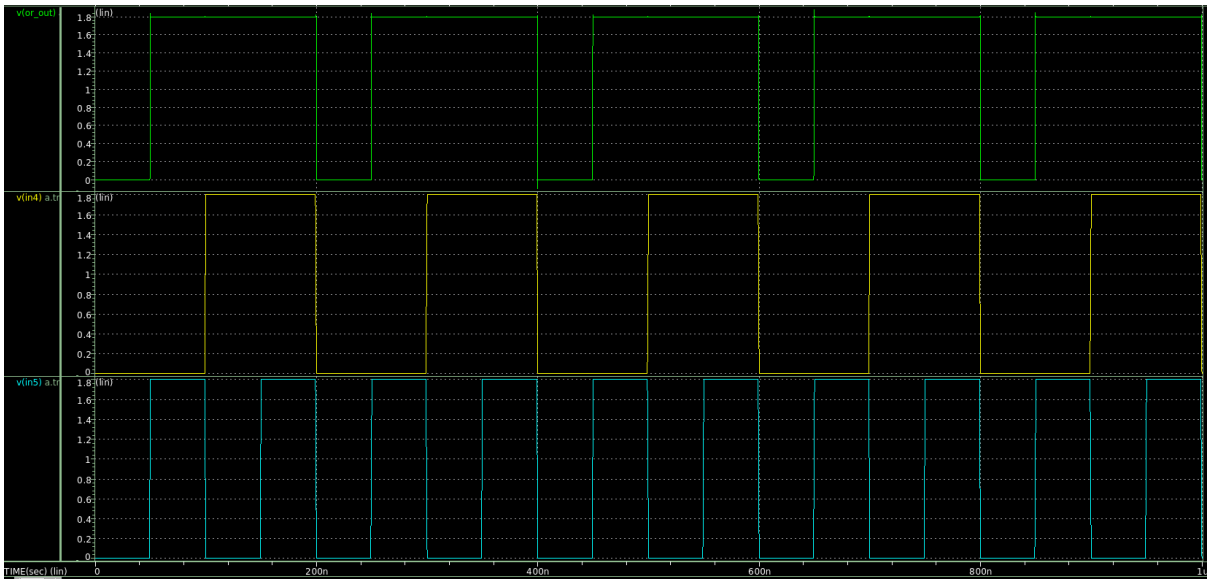


2. Or gate

I use Nor and an inverter to get a Or gate.

```
*** NOR Gate ***
.subckt NOR2 in1 in2 NOR_out vdd vss
mp1    vdd    in1    net2    vdd    p_18    w=0.5u     l=0.18u
mp2    net2   in2    NOR_out vdd    p_18    w=0.5u     l=0.18u
mn1    NOR_out  in1    vss    vss    n_18    w=0.25u    l=0.18u
mn2    NOR_out  in2    vss    vss    n_18    w=0.25u    l=0.18u
.ends
*** OR Gate ***
.subckt OR2 in4 in5 or_out vdd vss
XNOR in4 in5 net_nor vdd vss NOR2
XINV net_nor OR_out vdd vss INV
.ends
```

```
XtestOr in4 in5 or_out vdd vss OR2
V4 in4 vss PULSE(0v 1.8v 99ns 1ns 1ns 99ns 200ns)
V5 in5 vss PULSE(0v 1.8v 49ns 1ns 1ns 49ns 100ns)
```

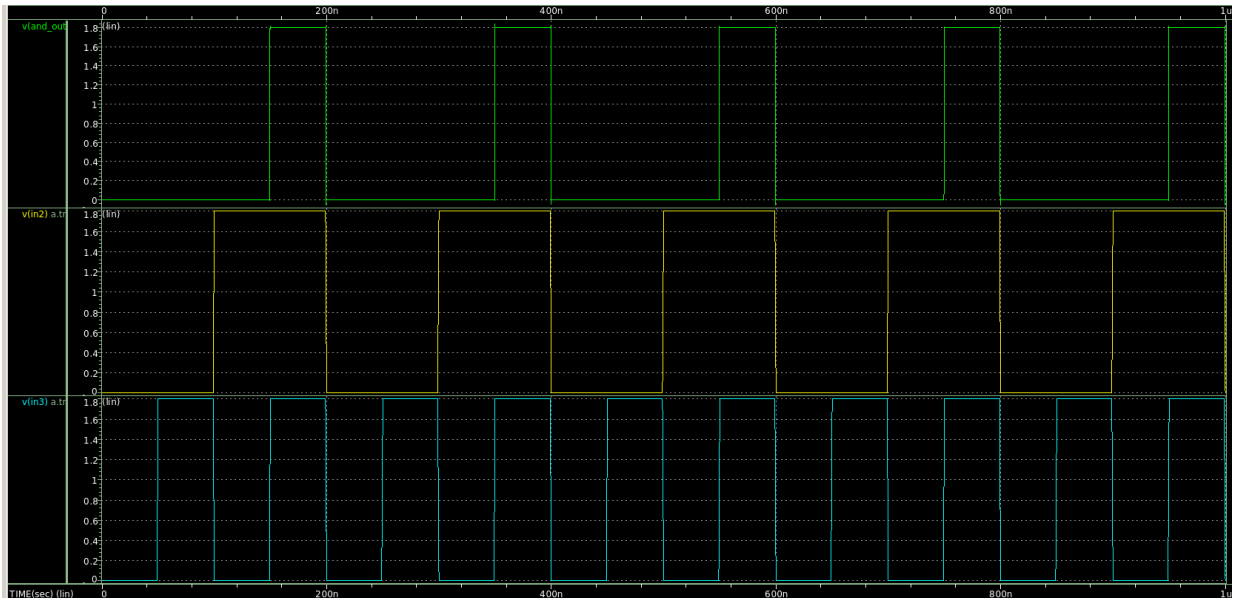


3. And gate

I use Nand and an inverter to get a And gate.

```
*** NAND Gate ***
.subckt NAND2 in1 in2 NNAD_out vdd vss
mp1    vdd    in1    NNAD_out    vdd    p_18    w=0.5u     l=0.18u
mp2    vdd    in2    NNAD_out    vdd    p_18    w=0.5u     l=0.18u
mn1    NNAD_out    in1    net1    vss    n_18    w=0.25u    l=0.18u
mn2    net1    in2    vss    vss    n_18    w=0.25u    l=0.18u
.ends
*** AND Gate ***
.subckt AND2 in2 in3 and_out vdd vss
XNAND in2 in3 net_nand vdd vss NAND2
XINV net_nand AND_out vdd vss INV
.ends
```

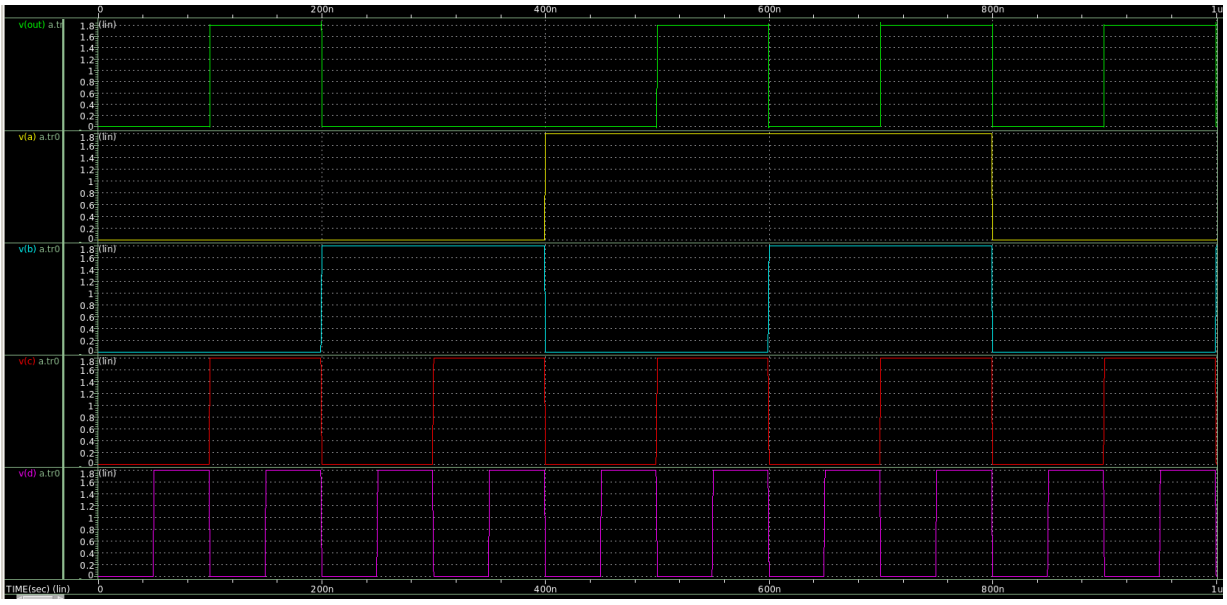
```
XtestAnd in2 in3 and_out vdd vss AND2
V2 in2 vss PULSE(0v 1.8v 99ns 1ns 1ns 99ns 200ns)
V3 in3 vss PULSE(0v 1.8v 49ns 1ns 1ns 49ns 100ns)
```



4. Logic function

```
.subckt logic_function a b c d out vdd vss
* Invert b to get !b
XINB b nb vdd vss INV
* First term: d(!b * c)
XAND1 nb c nb_A_c vdd vss AND2
XAND2 d nb_A_c first vdd vss AND2
* Second term: c(!b + a)
XOR1 nb a nb_O_a vdd vss OR2
XAND3 c nb_O_a second vdd vss AND2
* Final OR gate to combine the two terms: d(!b * c) + c(!b + a)
XOR2 first second out vdd vss OR2
.ends
```

```
* Logic function instantiation
Xlogic a b c d out vdd vss logic_function
*** Testbench with PULSE inputs ***
* Input signals with PULSE
Vd d vss PULSE(0v 1.8v 49ns 1ns 1ns 49ns 100ns)
Vc c vss PULSE(0v 1.8v 99ns 1ns 1ns 99ns 200ns)
Vb b vss PULSE(0v 1.8v 199ns 1ns 1ns 199ns 400ns)
Va a vss PULSE(0v 1.8v 399ns 1ns 1ns 399ns 800ns)
```



我有先自己化簡。

原本的應該是這樣

$$out = d\overline{(b + \bar{c})} + c(\bar{b} + a)$$

化簡後是 d(!b * c) + c(!b + a)

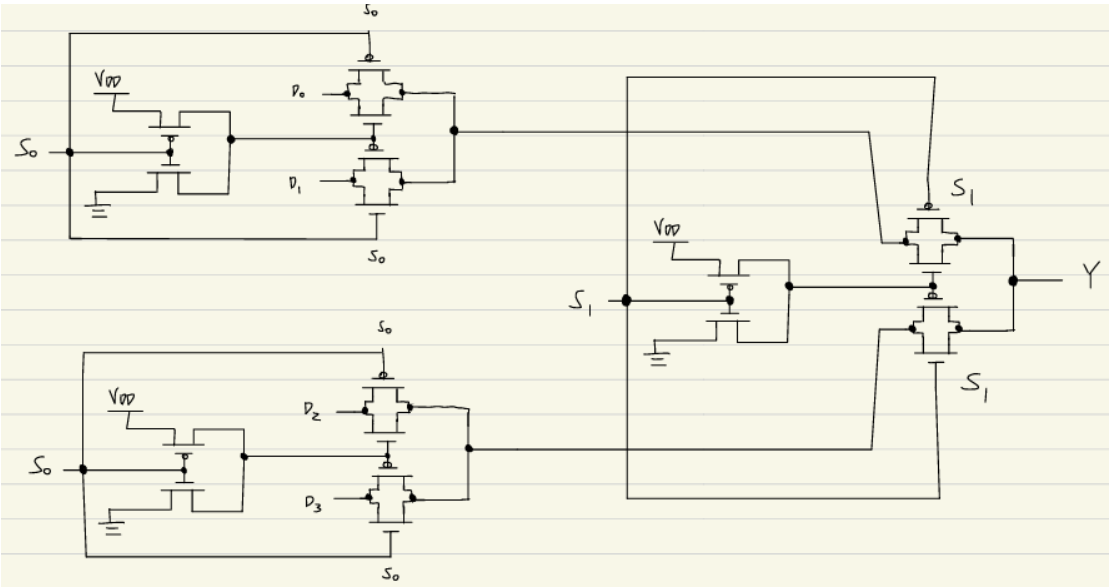
4. The hardness of this assignment and how you overcame it.

我覺得最難的部分是一開始寫 inverter nor nand gate 的時候，因為我其實對 nMos pMos 還不是很熟悉，所以需要花很多時間想。我透過看講義中的畫法，還有其他網路上的資料教學來了解確切該如何用 nMos pMos 這些基本的元件做出基本的 gate。

5. Any suggestions about this homework?

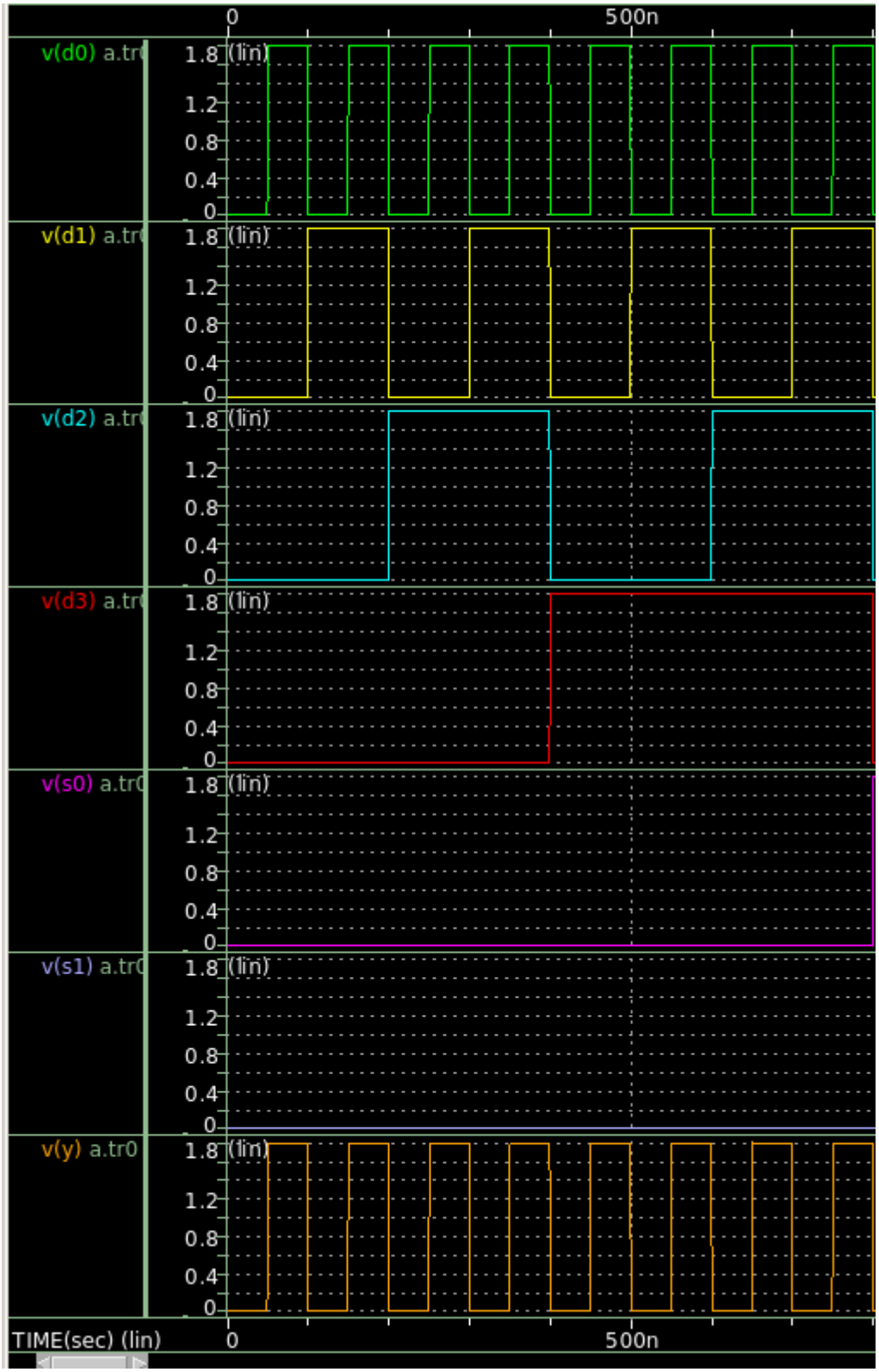
謝謝助教一開始的 introduction 影片，讓我更了解這項作業大致上的方向。

6. If you implement the bonus version, you have to provide the transistor-based schematic diagram in your report.

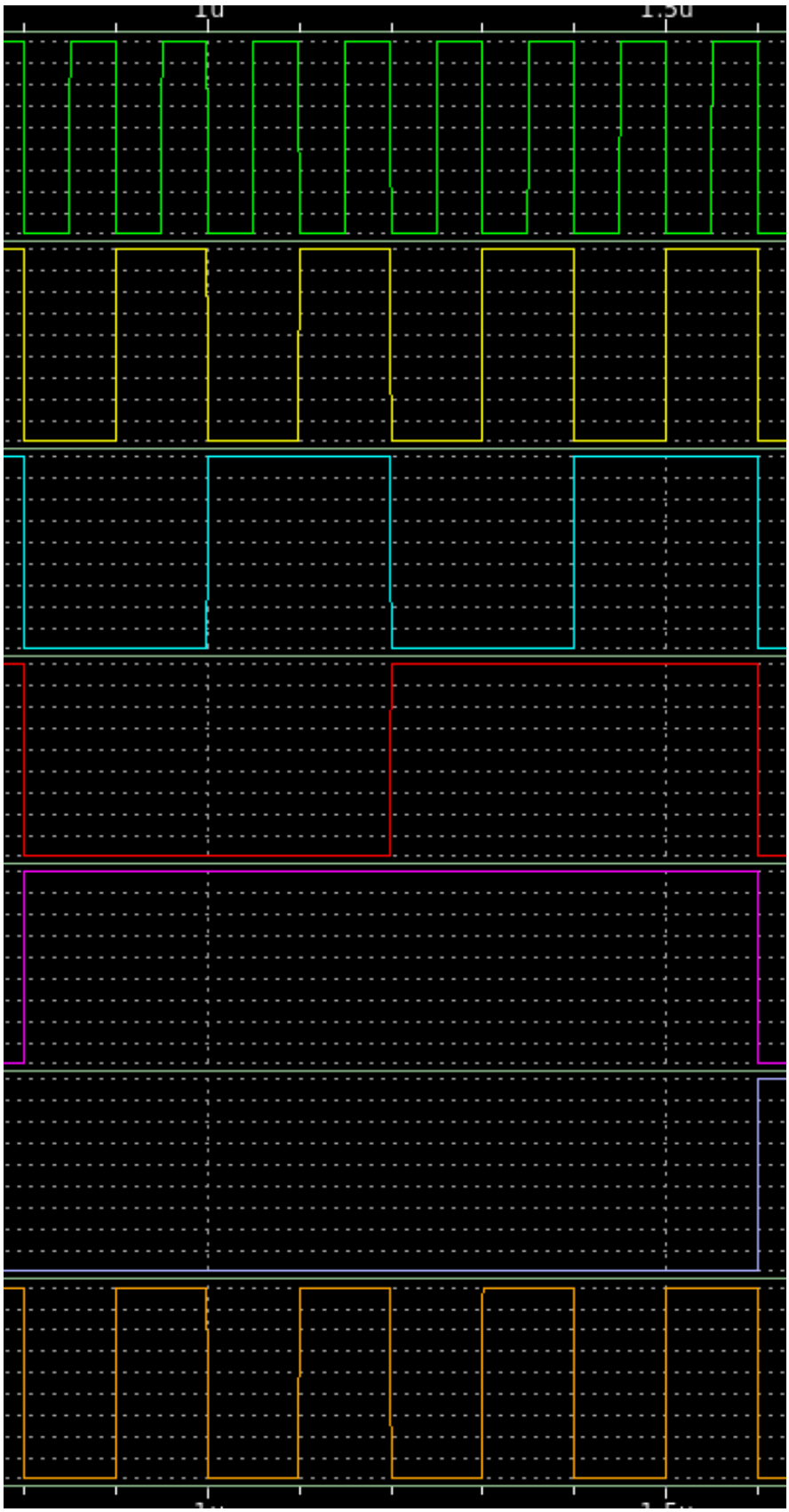


7. Waveform of bonus part

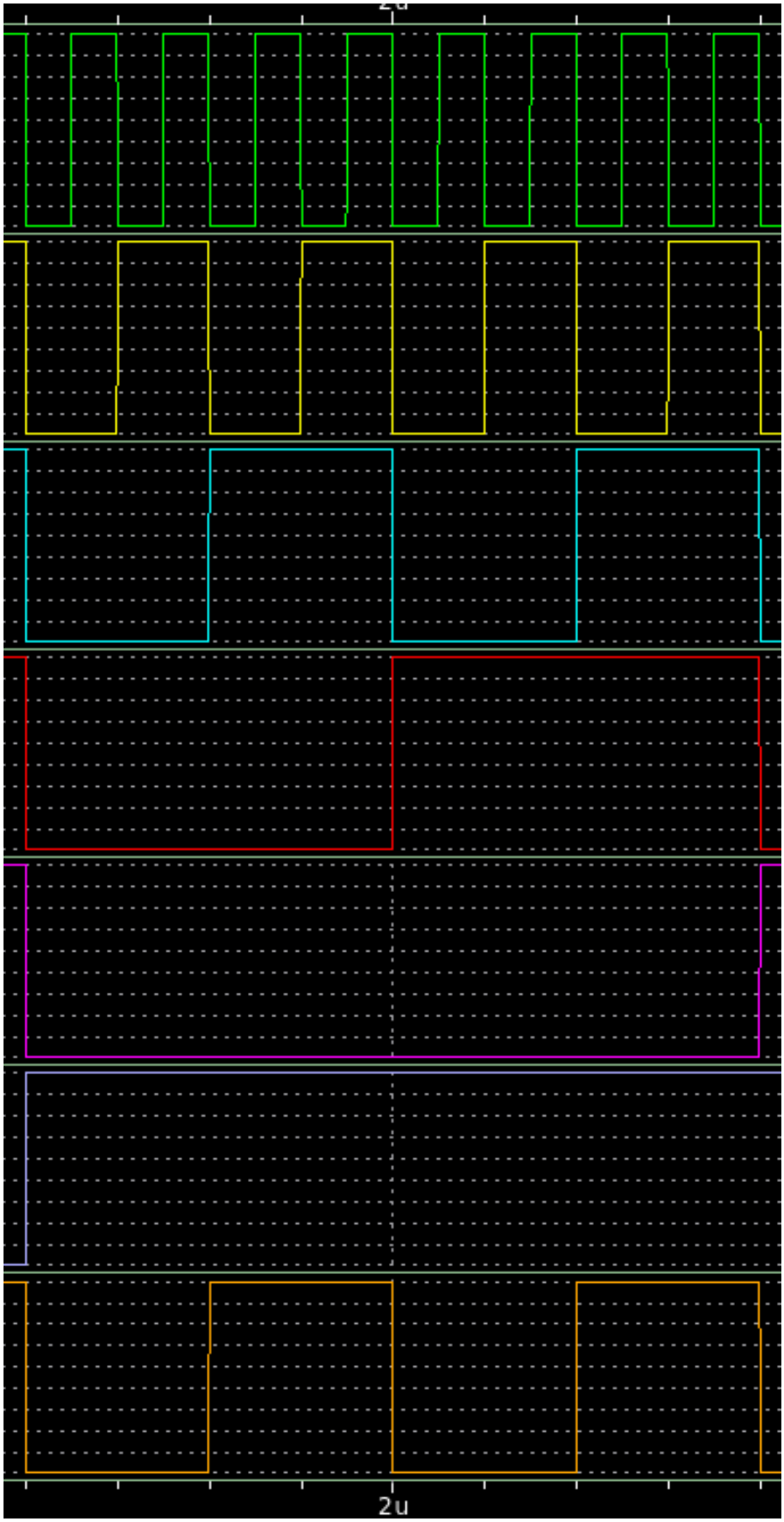
This is when s0 and s1 all equal to 0, then y will be the same with d0.



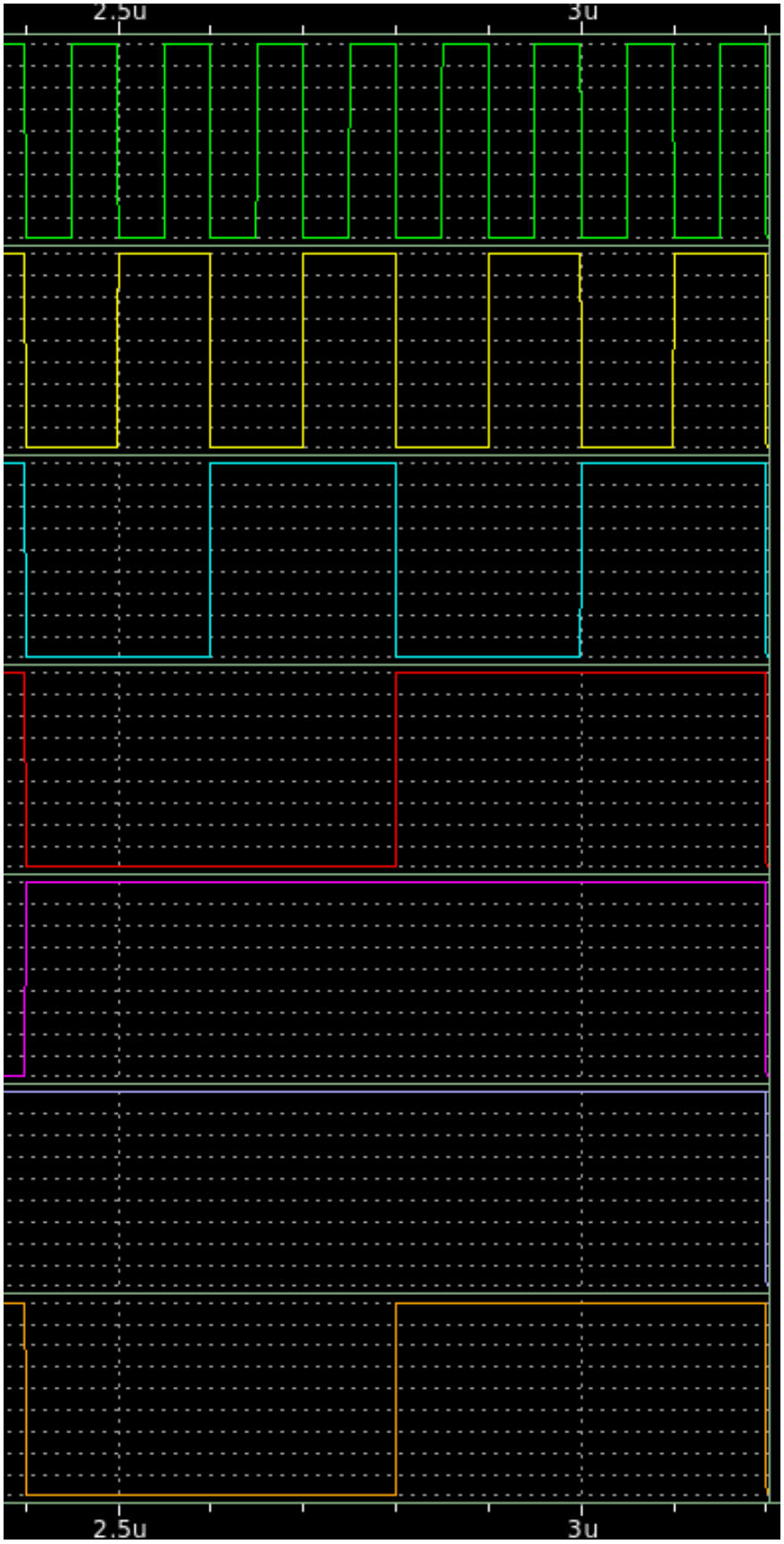
This is when s0 equals to 1 and s1 equals to 0, then y will be the same with d1.



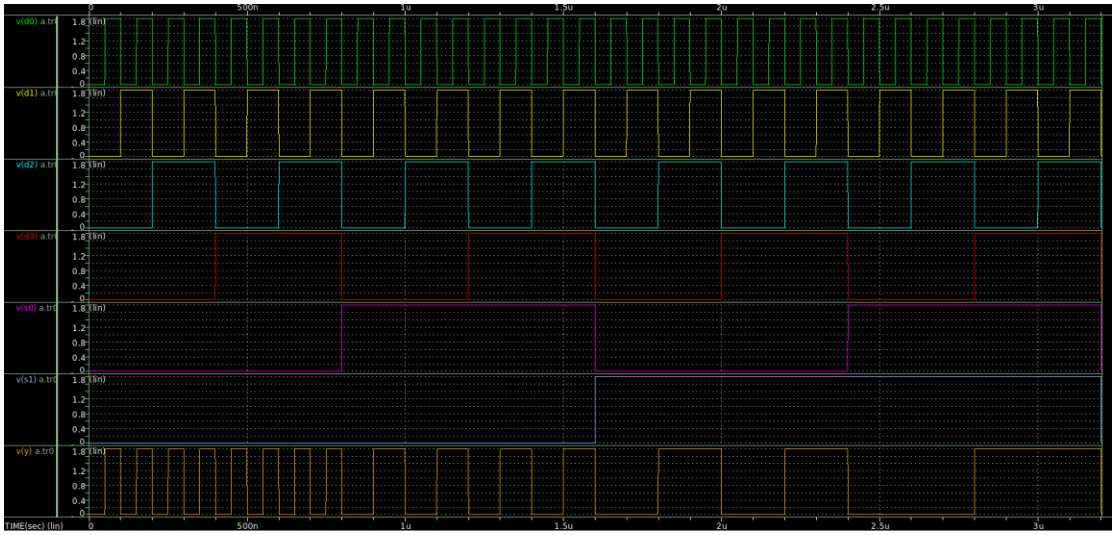
This is when s0 equals to 0 and s1 equals to 1, then y will be the same with d2.



This is when s0 equals to 1 and s1 equals to 1, then y will be the same with d3.



This is overview of the waveform.



S1	S0	Output should be the same with
0	0	d0
0	1	d1
1	0	d2
1	1	d3