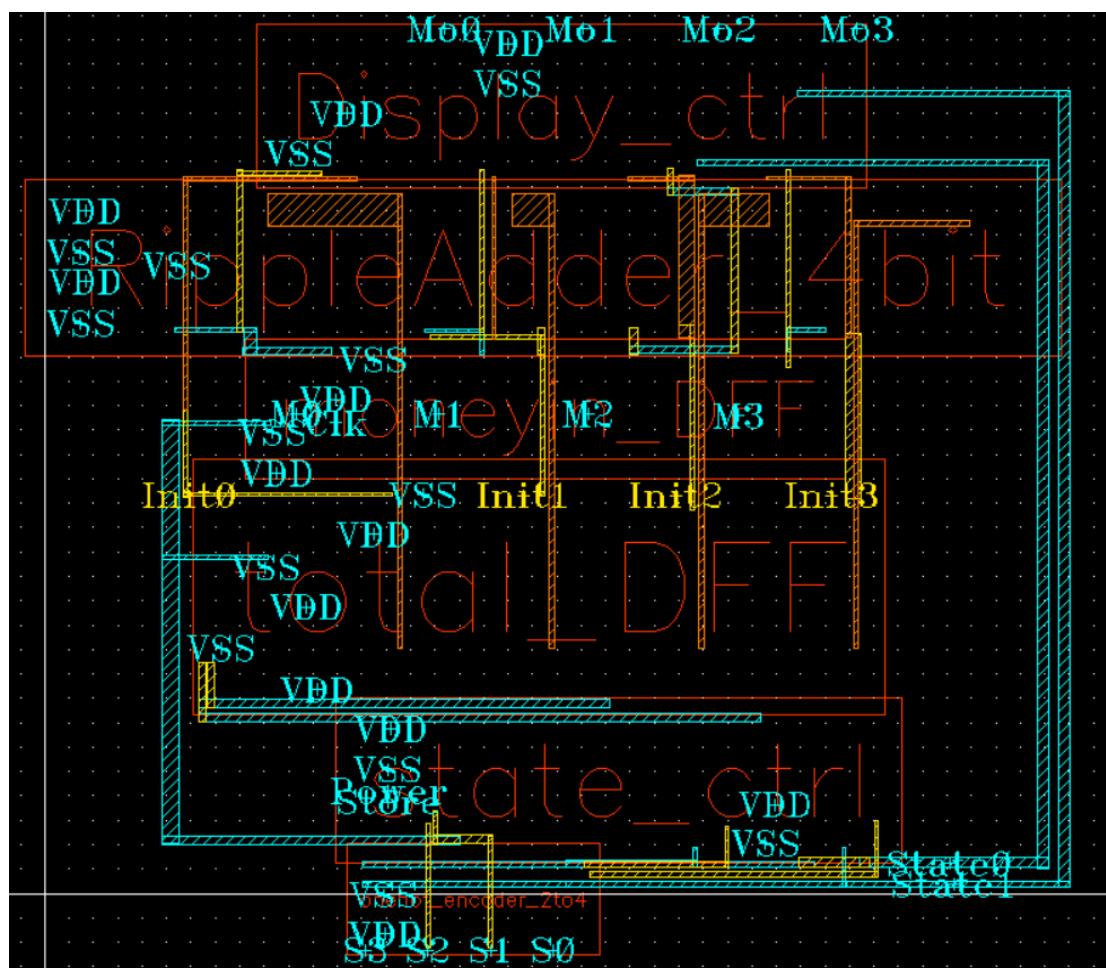


積體電路設計概論 Final Project report

1. The circuit diagram of your design and explaining your design (You can use screenshot to explain)

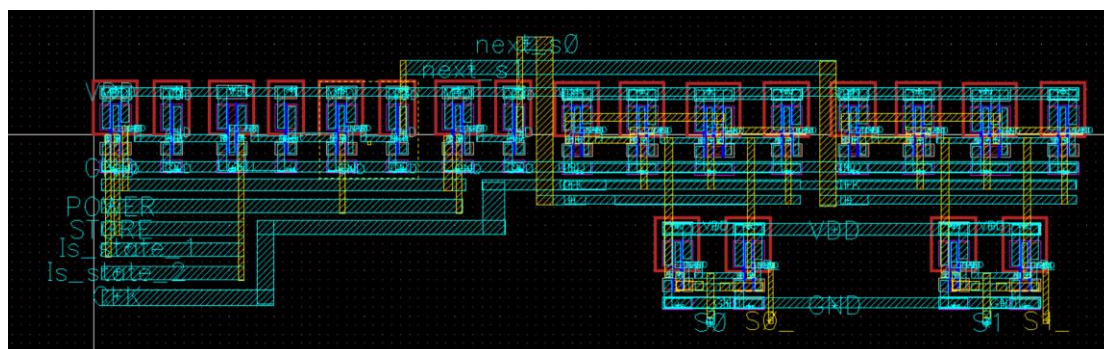


我的 coin_bank 主要有六個模組：

- ✧ State_ctrl:由兩個 DFF 以及數個 MUX 組成，根據 store 和 Power 值進行 state 的轉換。負責 state1,state0 output。State 轉換的邏輯為：若 power 為 0，則 S0 S1 皆為 0。

```
Xnand1 Is_state_1 STORE s0_tmp VDD GND NAND2
XINV s0_tmp s0sel VDD GND INV

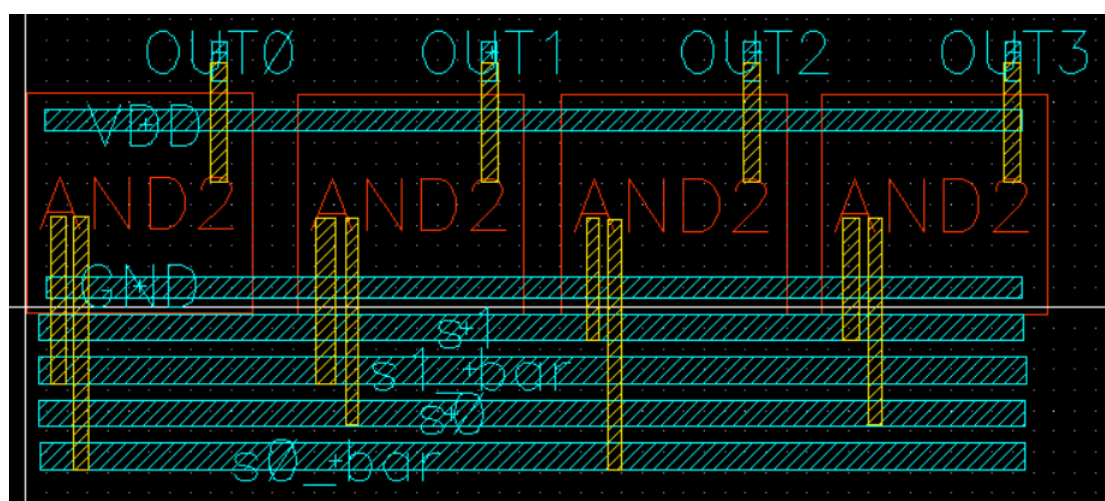
*Xand1 Is_state_1 STORE s0sel VDD GND AND2
Xor1 s0sel Is_state_2 s1sel VDD GND OR2
```



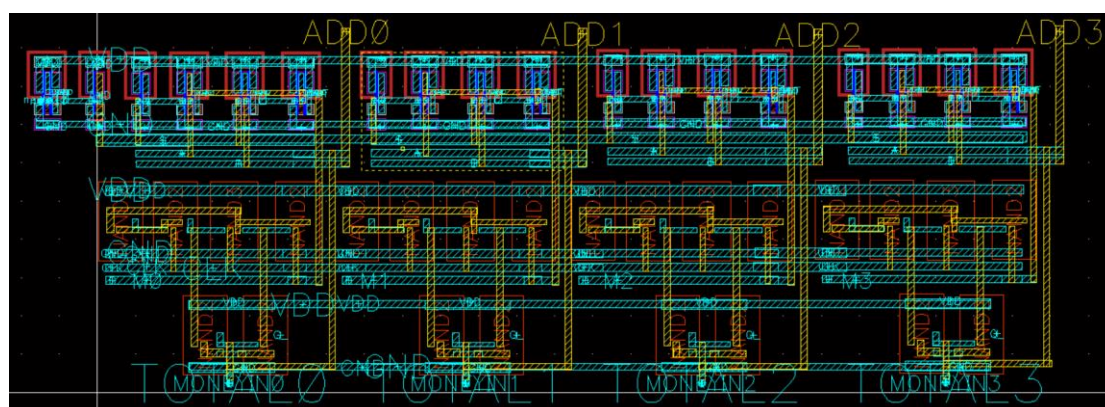
右邊是兩個 DFF，左方是一連串邏輯電路，計算 next state 傳輸至 DFF。

使用 power 分別和 s1sel 以及 s0_tmp and 即可完成 state 切換邏輯。

- ✧ Encoder：將兩 bit 表示的 state，變成 one-hot 形式。負責 s3,s2,s1,s0 的 output。分別使用 state_ctrl 的 s1,s1_,s0,s0_ 的訊號進行 and，即可完成。

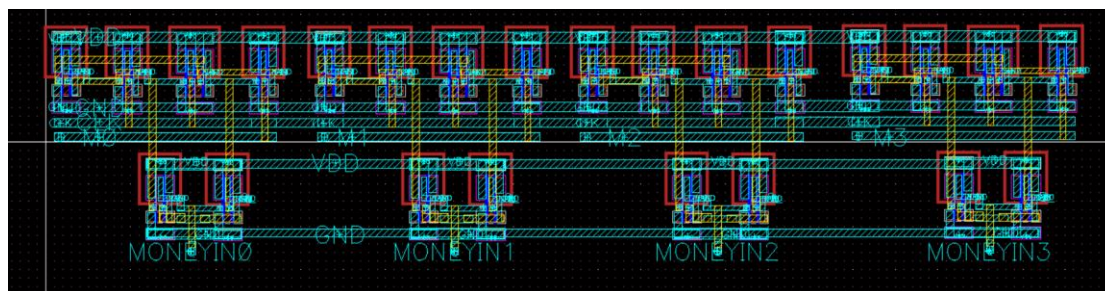


- ✧ Total_DFF：儲存總額，並附有 MUX，會根據 state_ctrl nextstate 決定是否 update 新的值。若 next_state 為 3，則選擇 added value 為 update 訊號。



(上排為 MUX，最左方有一個 AND，計算 nextstate)

- ✧ Money_DFF：暫存 m3 m2 m1 m0 的地方。

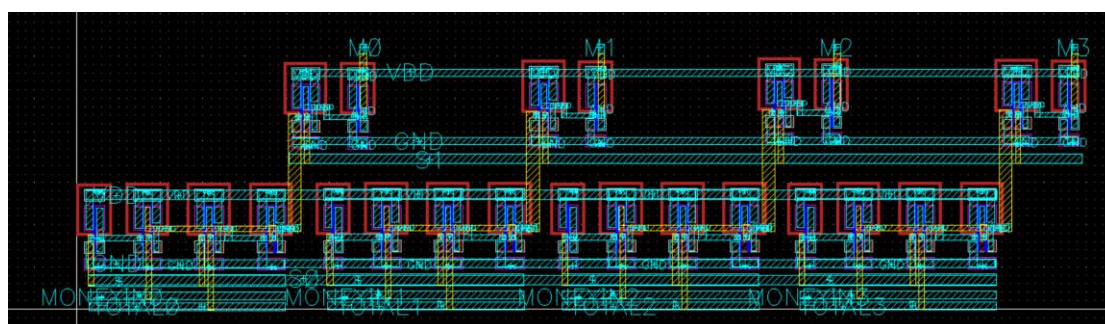


- ✧ Ripple_adder：4 bit 加法器，進行 money_in 和 total_DFF 的運算。我使用 XOR 和 AND 組成 halfadder，接著使用兩個 halfadder 變成 fulladder，接者串連 4 個 adder 製成 four bits adder。



(每一個小單位是一個 full adder)

- ✧ Dplay_ctrl：根據 state_ctrl 的 state 決定 output 內容。使用 s0 選擇要 moneyin 還是 total money 的訊號，使用 s1 去 and，即可完成此邏輯。

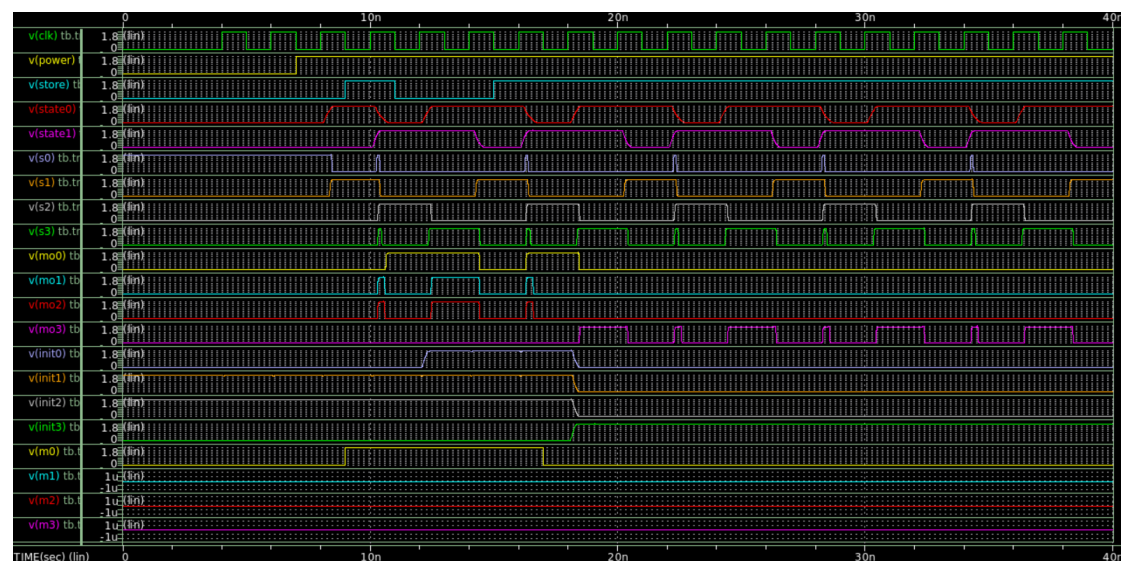


(下方為 MUX 上方為 AND)

2. Pre-sim waveform



(cycle:4ns)



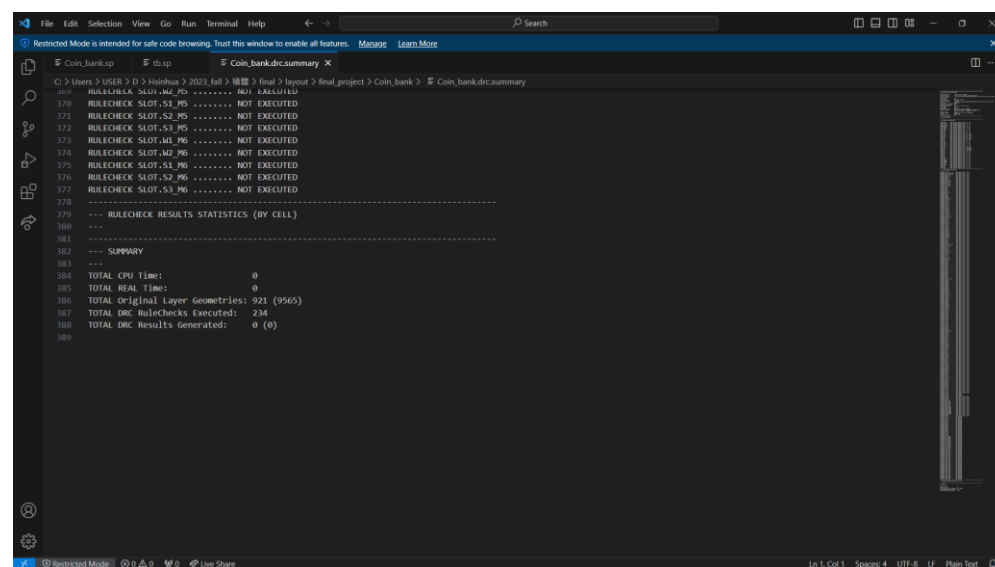
(cycle:2ns)

在 testbench 中，init 初始值為 0110，透過此 testbench，能夠檢查：

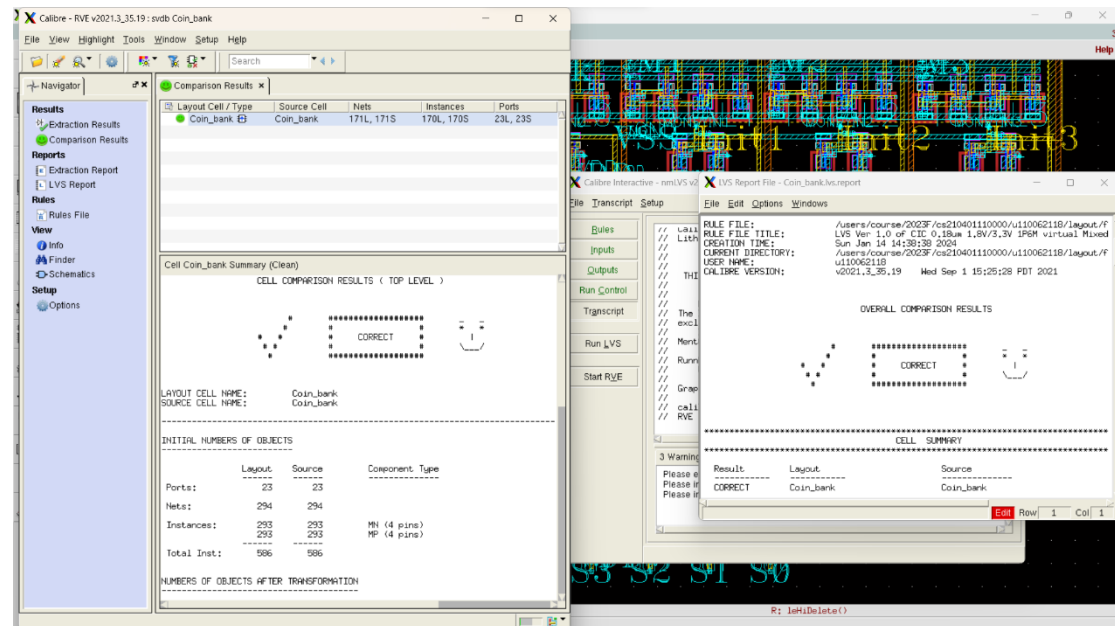
- ✧ Power 和 state 的關係
- ✧ Total money 是否會在正確的時間 update
- ✧ One hot encoder 是否正常運行
- ✧ State 是否正確轉換
- ✧ Monitor 是否正確運作
- ✧ Adder 是否正確運作

另外，可以看見某些訊號有嚴重 glitch，但皆在 clk 正緣發生，因此並不特別修正。在 state 相關波行將之和 postsim 重疊顯示，可以發現 postsim 有些許的延遲。但在負緣時數值正確。

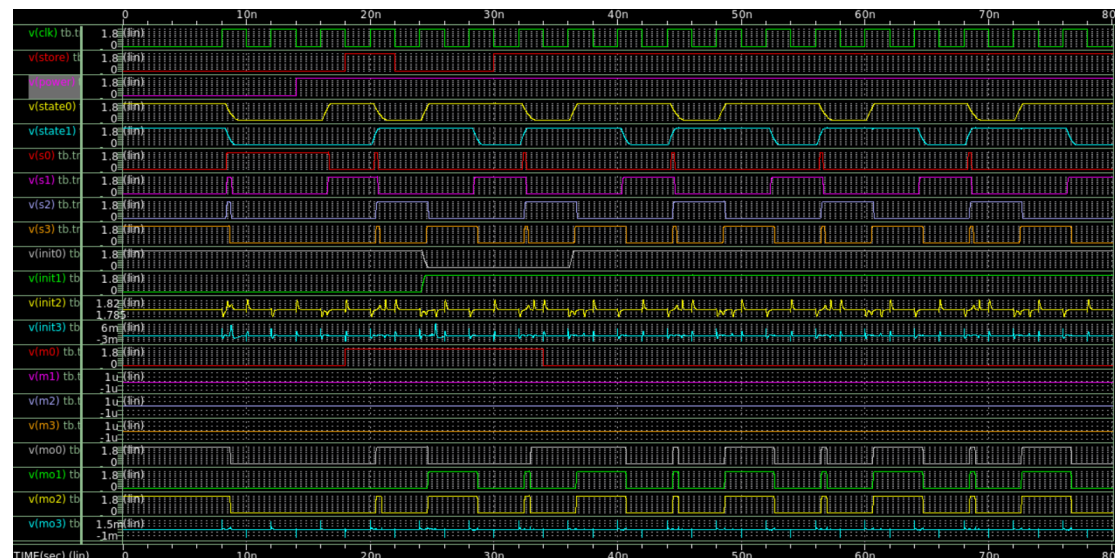
3. DRC summary report :)



4. The message of passing LVS



5. Post-sim waveform



Post_sim 的波型和 pre 不同，但主要是因為 Init 的值不同(0110->0101)的緣故，因此除了比對 pre_sim 外，我也自己對照了波行邏輯，確認無誤。

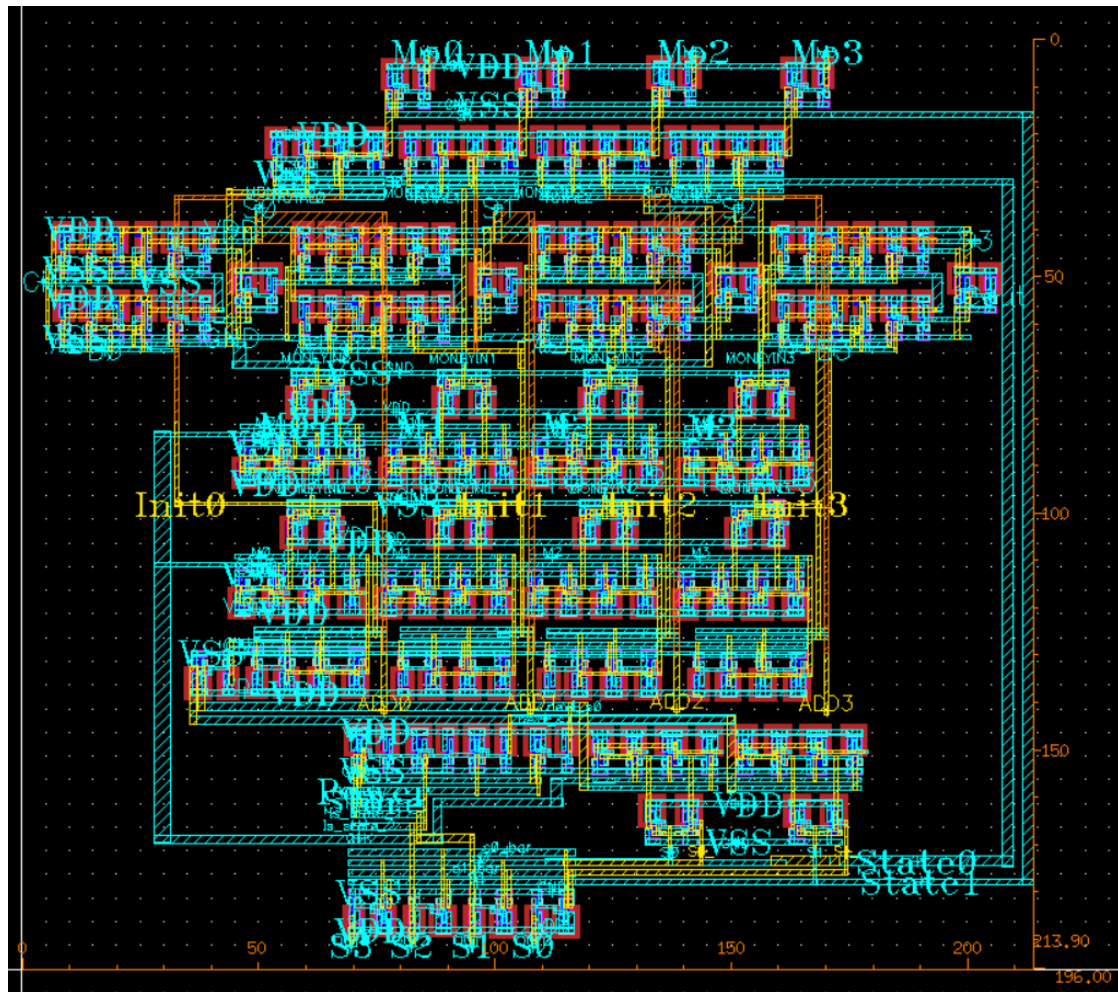
6. A form about the latency and area of the circuit

Area: 213.9um/196um(W/H)

Pre Latency = $20 \times 2n = 40n$

Post Latency = $20 \times 4n = 80n$

7. Image of your layout



8. The hardness of this assignment and how you overcome it

在進行小 module 的 layout 時，其實沒有面臨到很多困難，最難的是最後要將所有 module 拼湊起來時，畫面密密麻麻，錯綜複雜。有些擺法很好繞，但面積會很大，要在不同因素間考慮，我在繞線時會切換 display 模式，讓畫面變得簡潔。另外在最後合併階段，我花了很多的時間在解決 lvs error，我一度感到很絕望，還在想要不要重拉，幸好，透過 lvs 報的錯誤，逐步找出連錯的地方。過去的作業規模比較小，因此可以用肉眼檢查錯誤。這次的作業讓我學習到如何使用 lvs 的資訊幫助自己

9. Any suggestions about this programming assignment

Server 在 deadline 最後一天爆掉了，打報告時還沒好。(清寒大學)感謝教授願意延長時間。助教改作業辛苦了。