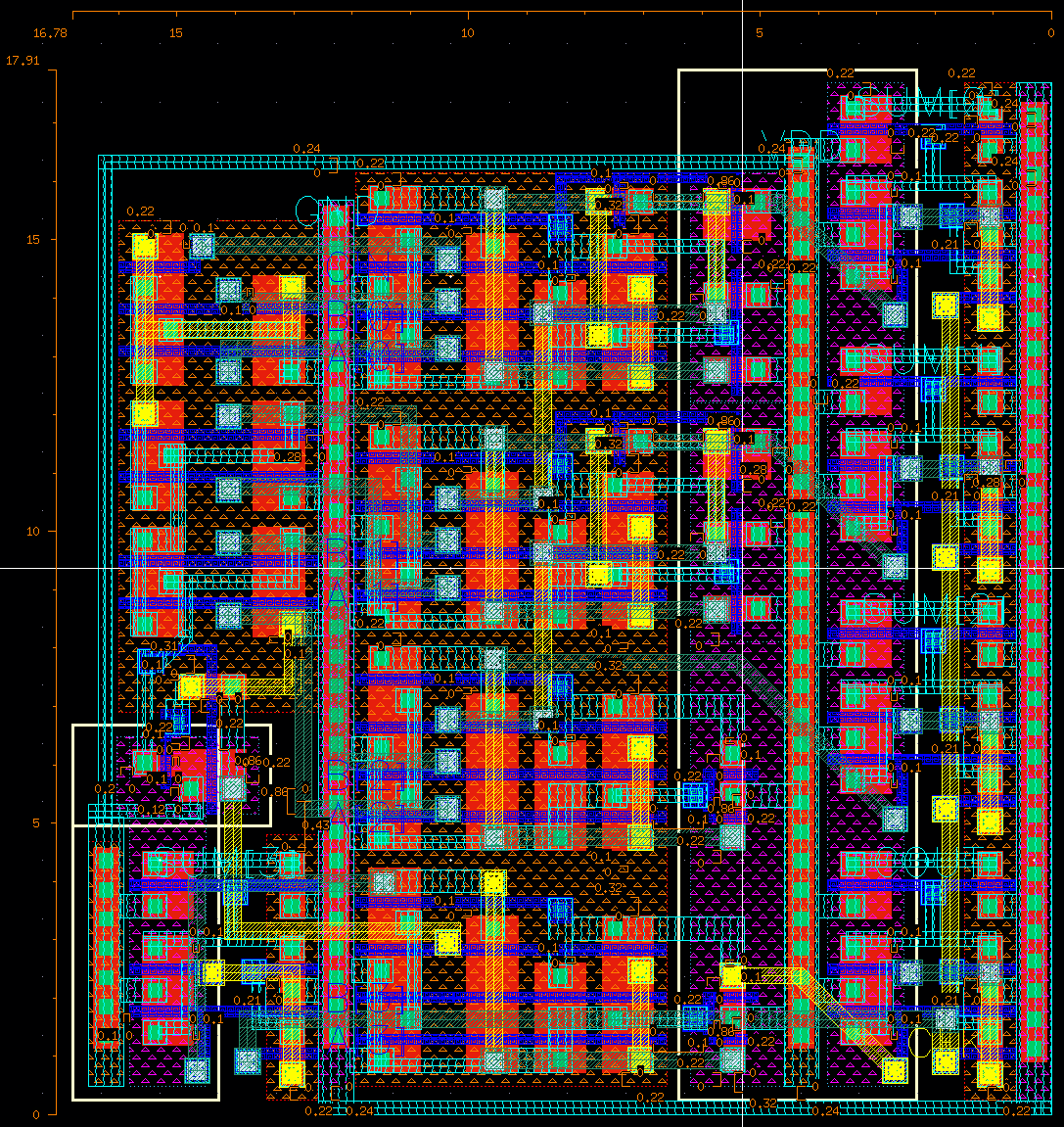
2022 NYCU EE VLSI Lab Report

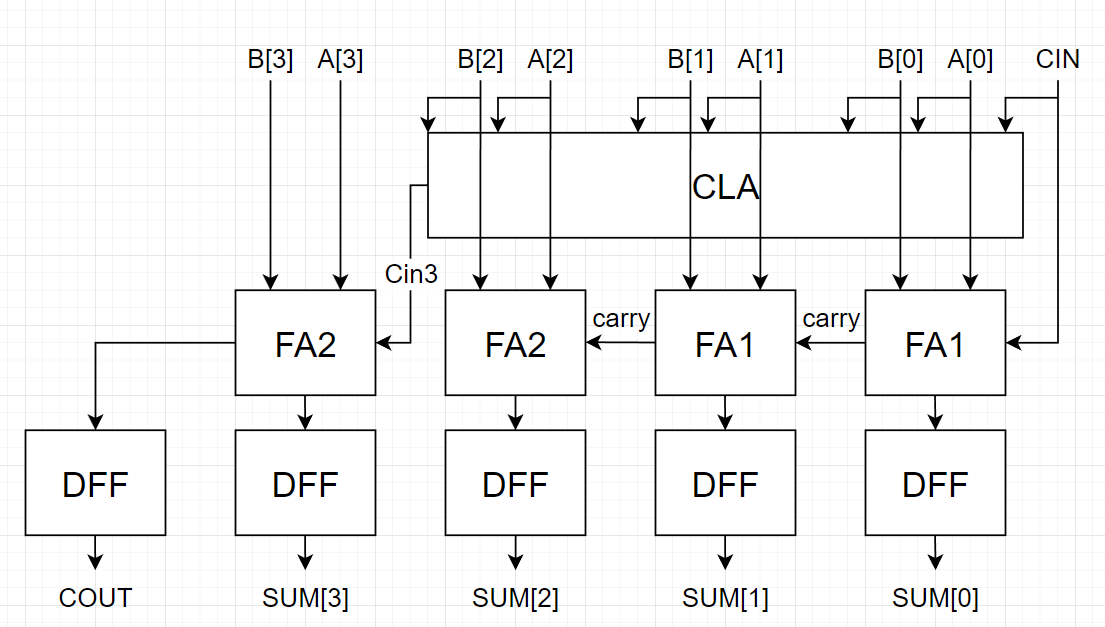
Lab03 4-Bit Full Adder **+** DFF

Student ID: 109611070 Name: 郭家均 Date: 2022/11/19

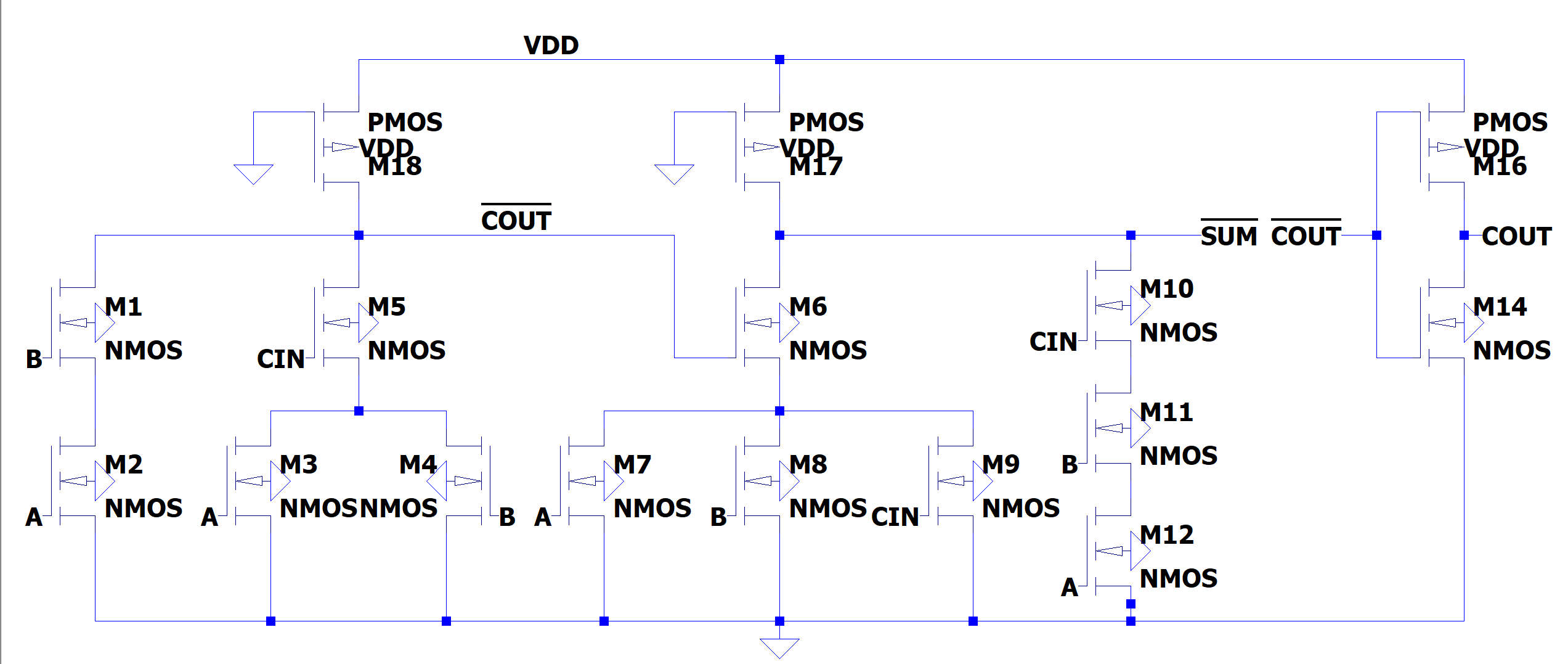
1. **Layout result**
2. **Layout picture with ruler** (Height: 17.91um, Width: 16.78um)



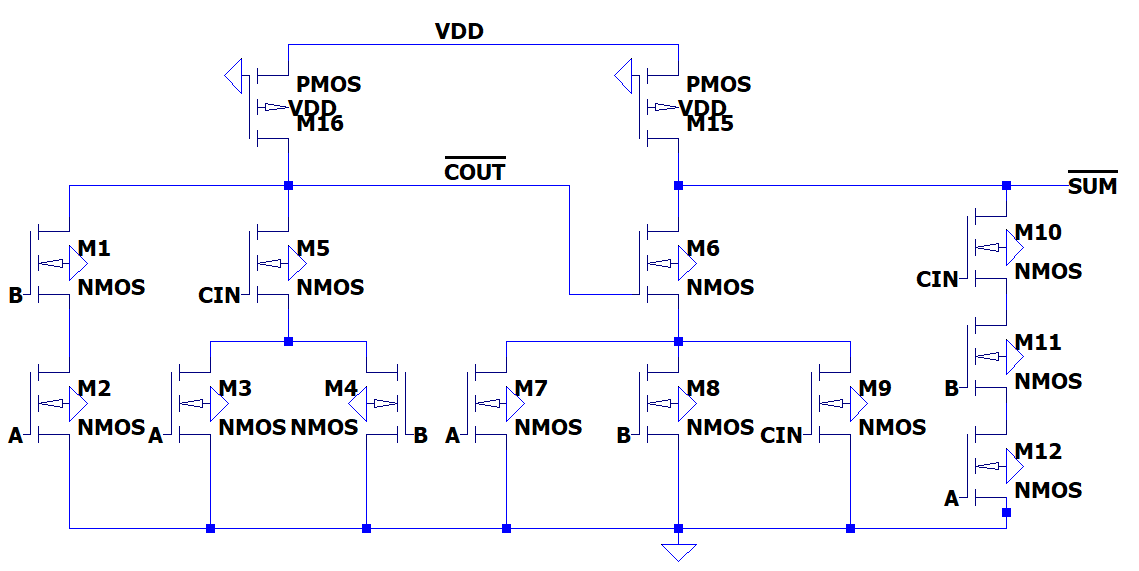
1. **Design concept**
2. Circuit Schematic

Block diagram :

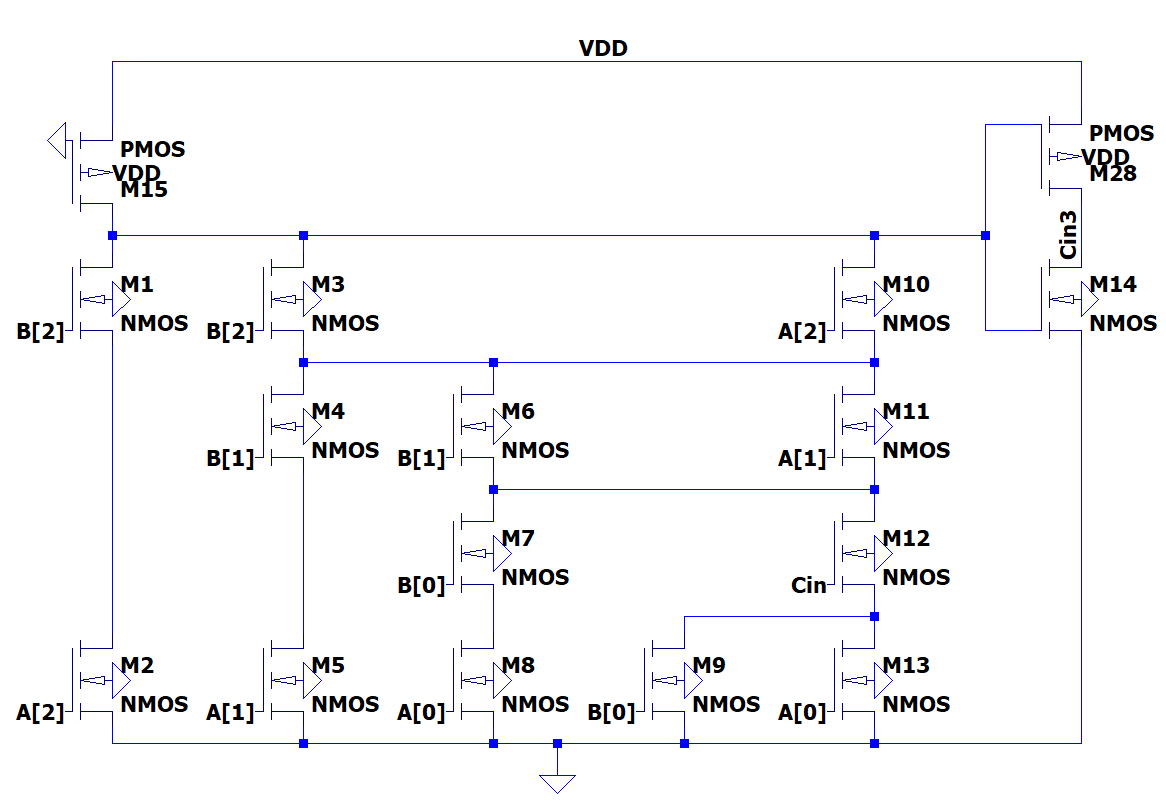
Sub-circuit : FA1, FA2, CLA, DFF

**FA1** (since DFF have inverse output, SUM doesn’t need inverter)

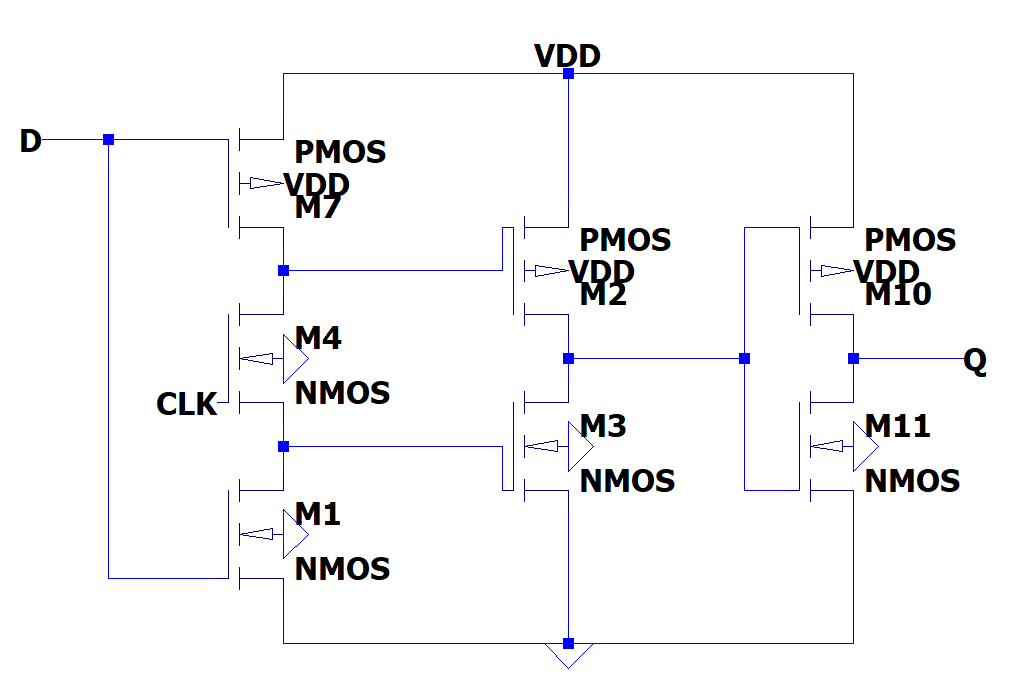
**FA2** (since implementing CLA circuit, 2FA don’t need COUT)



**CLA**



**High level trig D-latch + Inverter** 以下全部簡稱DFF



1. Summary of structure (number of transistor / logic gate is used)

此實驗我主要選用pseudo C-CMOS Adder + pseudo compound gate CLA + DFF來設計4bit-FA。全部用了111個Transistors。

簡單來說，前面三級FA是ripple adder架構，第四級FA的carry-in  
 由一個combinational CLA circuit去運算。縮短critical path的時間。

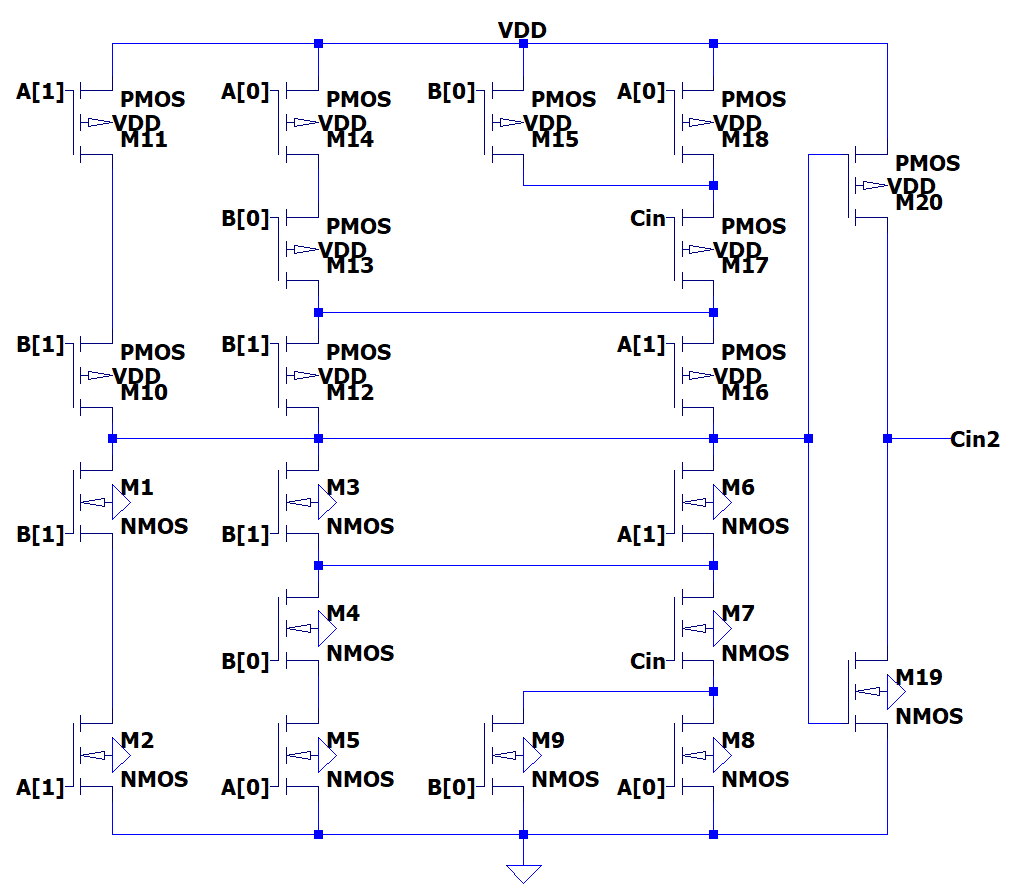
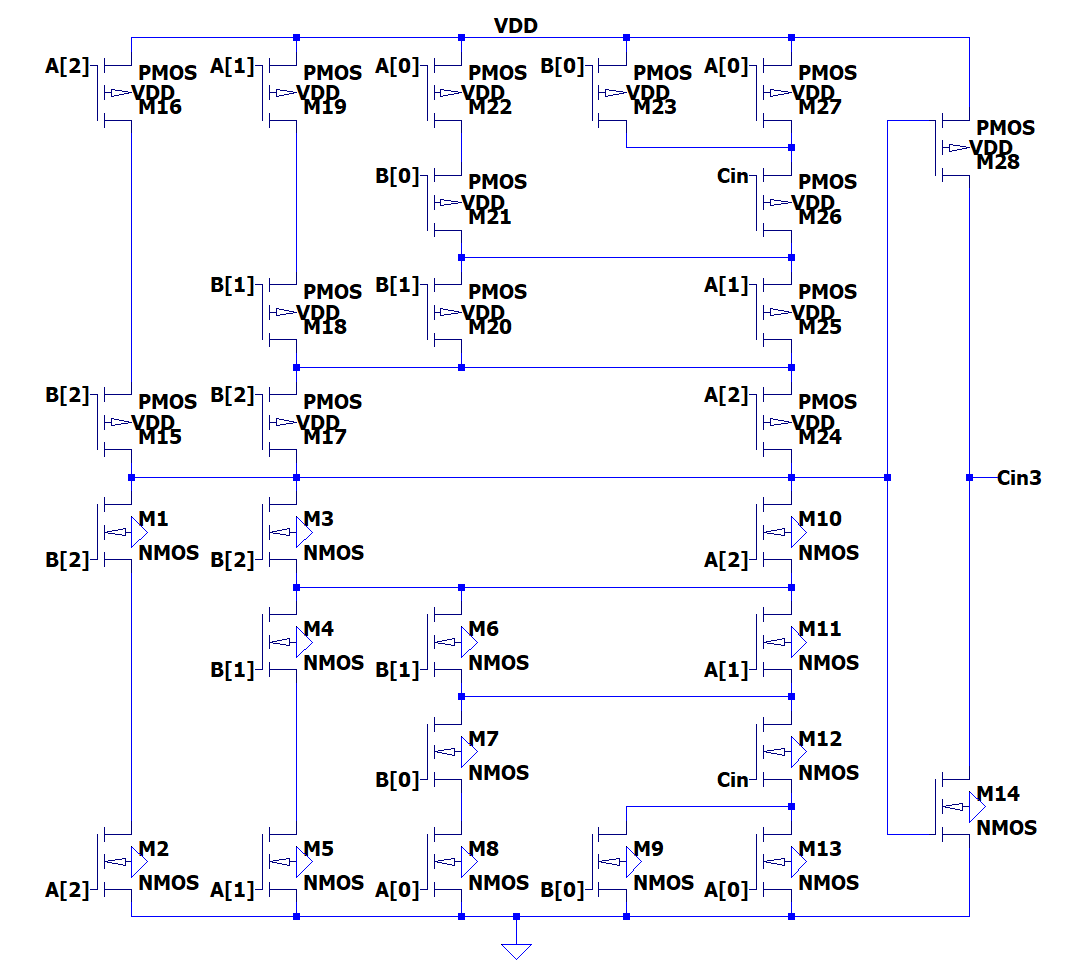
我們必須在CLK posedge前讓FA的output stabilize，意即運算速 度 越快越好，因此在猜想不同架構時去比較不同critical path的長短、 可以在沒有err0的狀況下使CLK最快為主要考量。

以下有一些subckt的代號解釋

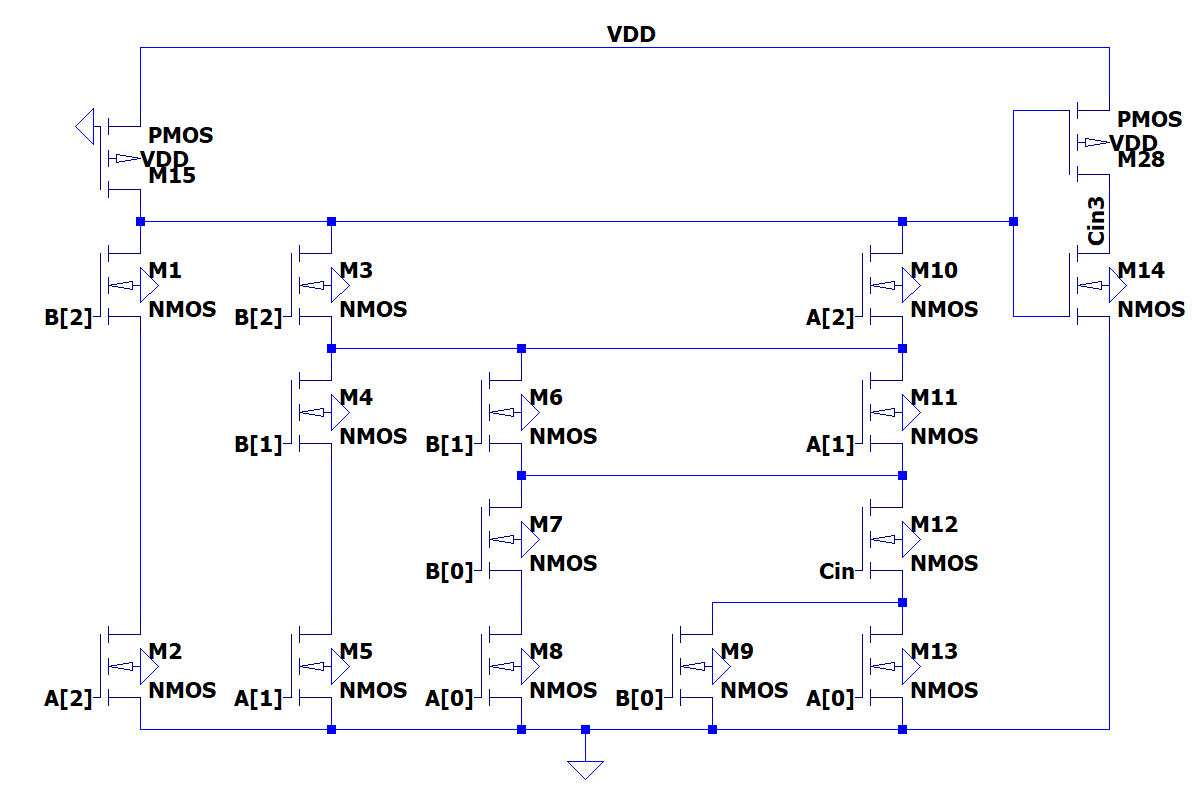
(carry 1邏輯運算因為和C-CMOS的算法相同因此不列入考慮)

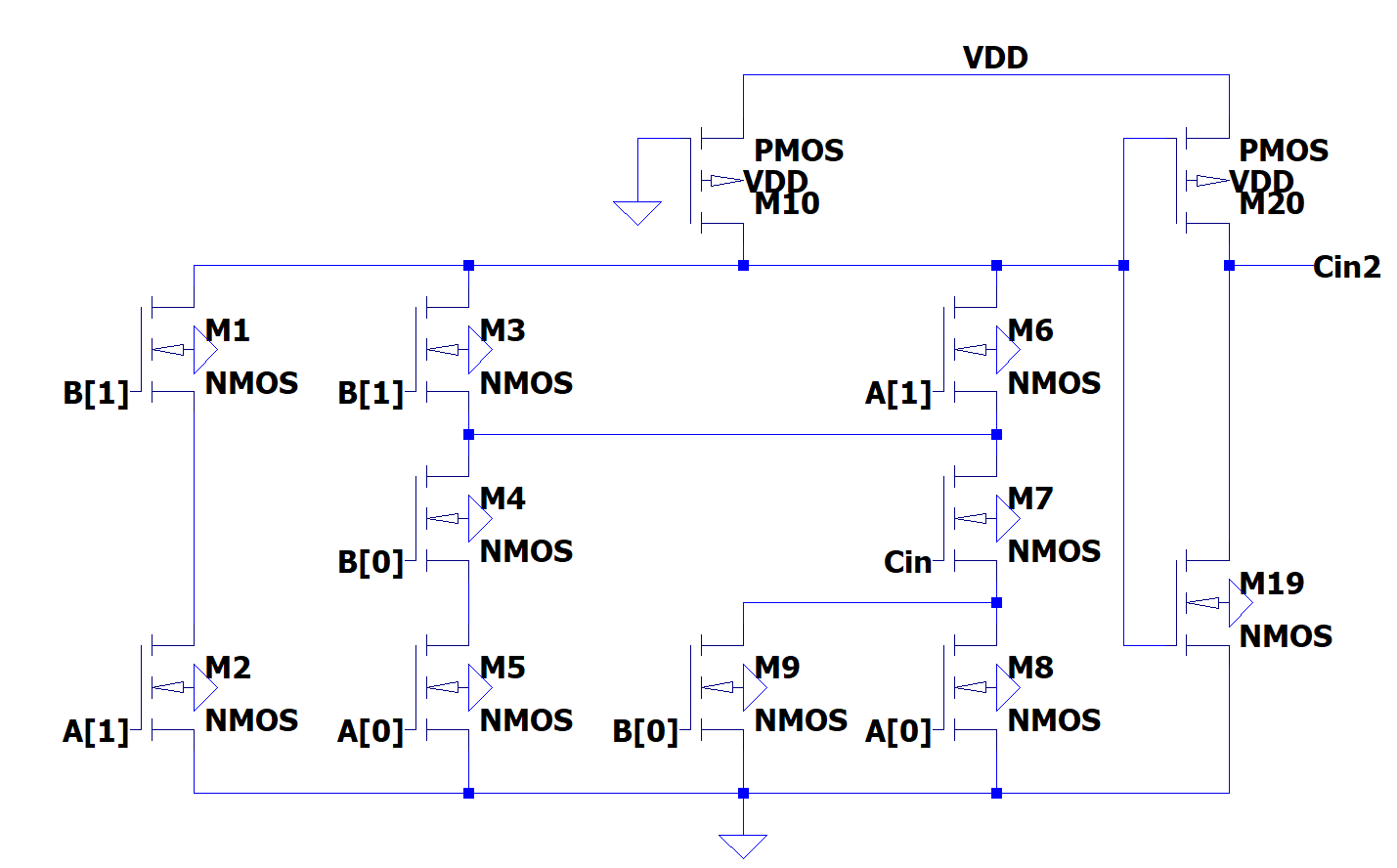
CLA2 : compound gate架構下output得到Cin2 (下左)

CLA3 : compound gate架構下output得到Cin3 (下右)



pCLA2 : CLA2改成pseudo NMOS形式(下左)

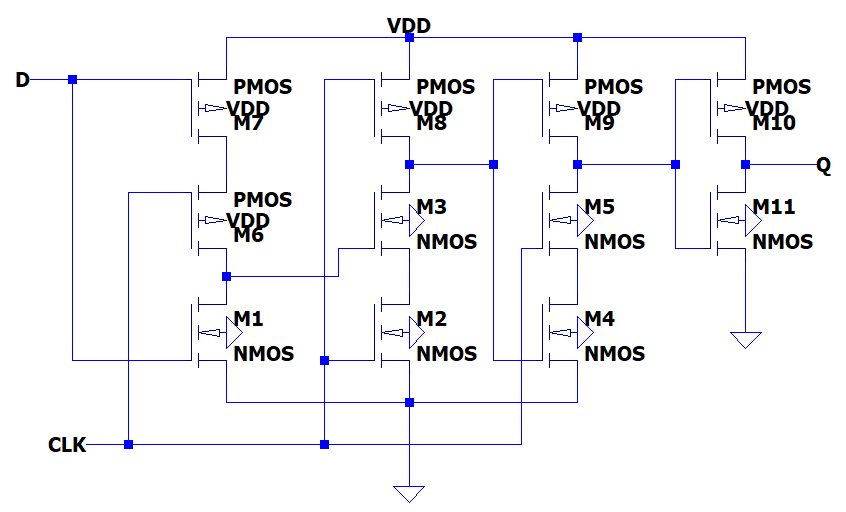
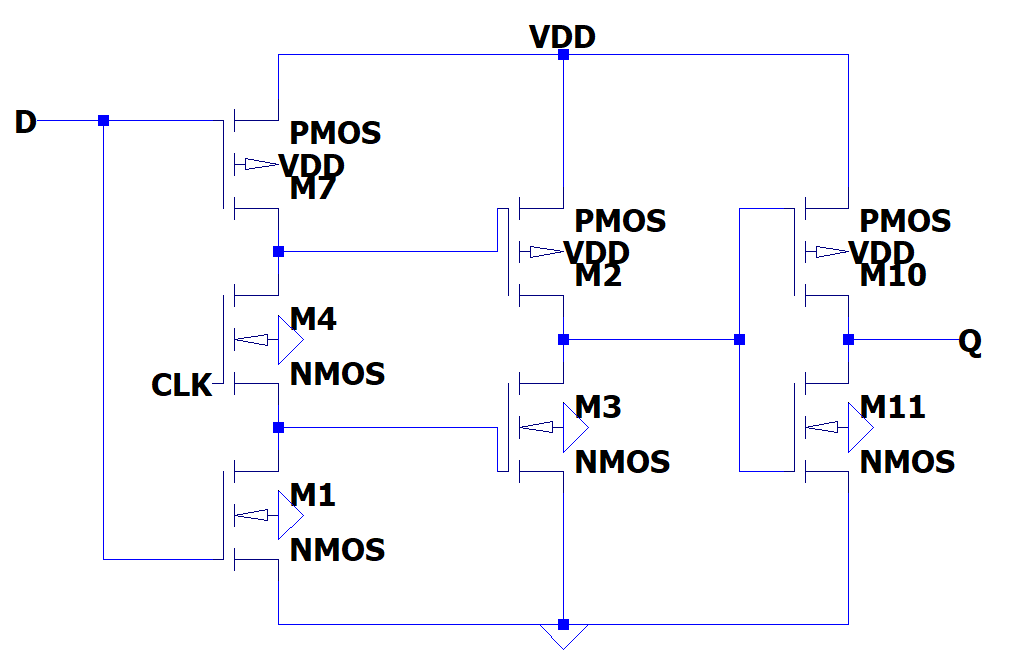
pCLA3 : CLA3改成pseudo NMOS形式(下右)

在實驗中我前後採用了兩種不同的DFF架構

DFF1 : 11 transistors , D in Q out (下圖左)

DFF2 : 7 transistors , D in Q\_bar out (下圖右, output名稱仍訂為Q)

實質上為High-level trigger D-latch + inverter



**有測試過以下的幾種架構** (藍色為使用DFF1, 橘色為使用DFF2) **:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Design Structure** | **Critical Path** | **CLK** | **Adder Type** |
| 4-bit RCA | A[0] => SUM[3] | >1.66n | C-CMOS |
| 2-bit RCA + CLA2 + 2-bit RCA | A[0] => CLA2 => SUM[3] | >1.66n | C-CMOS |
| 3-bit RCA + CLA3 + FA | A[0] => CLA3 => SUM[3] | >1.66n | C-CMOS |
| 2-bit RCA + pCLA2 + 2-bit RCA | A[0] => pCLA2 =>SUM[3] | 0.88n | Pseudo C-CMOS |
| 4-bit RCA | A[0] => SUM[3] | 0.68n | Pseudo C-CMOS |
| 2-bit RCA + pCLA2 + 2-bit RCA | A[0] => pCLA2 => SUM[3] | 0.62n | Pseudo C-CMOS |
| 2-bit RCA + pCLA2 + FA + pCLA3 + FA | A[0] => pCLA3 => SUM[3] | 0.52n | Pseudo C-CMOS |
| 3-bit RCA + pCLA3 + FA | A[0] => pCLA3 => SUM[3] | 0.52n | Pseudo C-CMOS |

經過多次驗證和實驗後，我歸納出影響電路速度的三大個因素。

DFF架構的差別、使用pseudo NMOS 架構、使用CLA架構

我們先對DFF1和DFF2進行分析 : (D-Latch vs DFF)

DFF1的電路因為是D in Q out，因此前方adder的電路接到DFF前必須先經由inverter stage拉為SUM、COUT的值。

DFF2的電路為D in Q\_bar out，因此adder的SUM output不用加上inverter stage，COUT的部分如果再RCA的架構下，除了最後一級不用加inverter stage外，其他級因為是互相傳遞不會經過DFF，所以都必須加上inverter stage。

不過因為adder電路都採用C-CMOS為原型，因此單看最後一級FA可以發現SUM的delay都會比COUT來的大，因此使用DFF2的架構速度上會優於DFF1，因為相當於critical path少了一級的inverter delay。DFF1的優點在前一級adder會經過inverter stage做restore，因此整體比較robust。

Pseudo NMOS的架構根據上表可以明顯看出差距，若是用一般C-CMOS架構，最基本的1.66ns無法達成。Pseudo NMOS大大改善了PMOS driving較小的問題，但相對上power的部分就會大很多。原因在於Pseudo NMOS的logic 0是在NMOS導通且driving 能力為PMOS四倍的情況下達成，會有電流從VDD 流到GND。

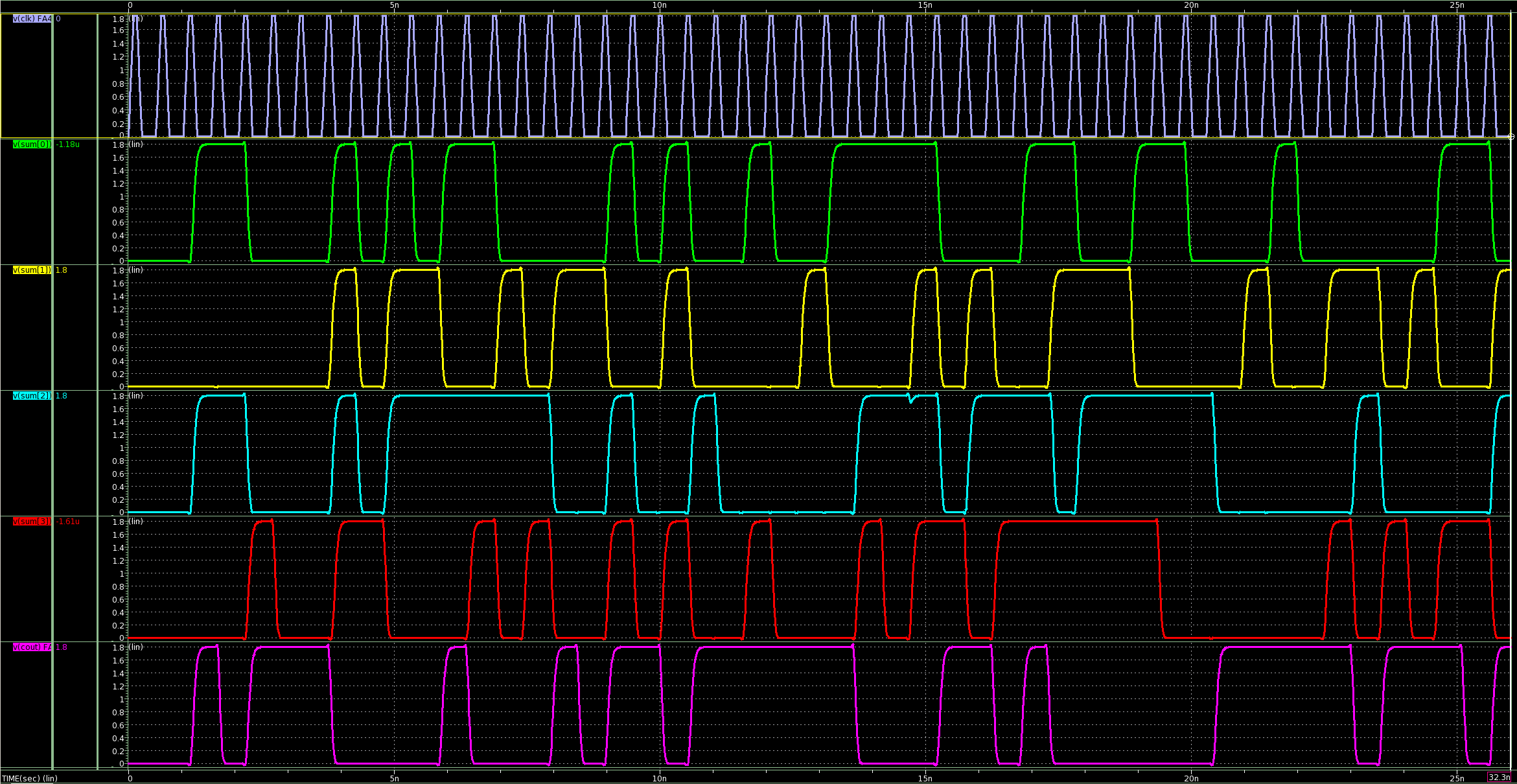
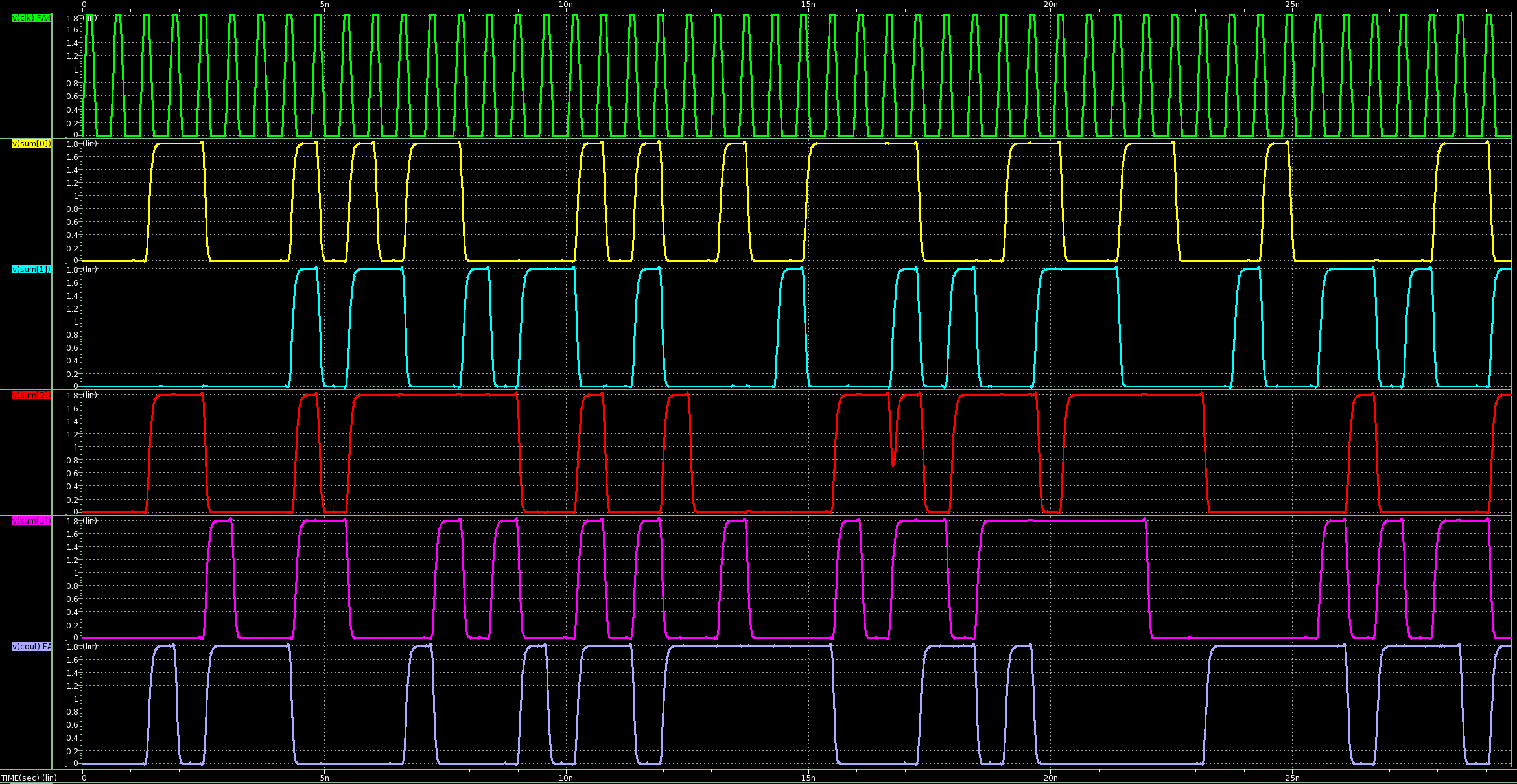
不過此次lab performance的部分不考慮power的問題，且設計的過程採full custom design，和stdcell design取向不同，因此最後使用pseudo來做。

在此需要特別注意的是，一般數位電路的設計上會避免pseudo NMOS的架構。高度整合的晶片，例如processor裡面會有很多運算的單元，如果單位面積有大量電流持續導通，將造成功耗過 大、晶片過熱、甚至讓電路整個燒掉。因此EDA抓取的stdcell library應該不會出現pseudo的架構。

CLA在4bit adder下效果並不顯著，原因在於bit數實在太少。一般RCA的架構如果出現8bit、16bit等，其critical path會出現在carry chain，因此在bit數多的情況下CLA會大大改善電路速度。但這次lab只有4bit，CLA用一般complementary的架構並沒有比RCA好多少。即使改成pCLA架構，改善幅度也只有0.1~0.15ns左右。甚至pCLA3 + FA的delay經過測試仍比3bit RCA delay大，可以從上表最後一欄看到critical path為經過pCLA3的路徑，而非A[0] => SUM[2]。

根據上表和分析過程，我最終選用3-bit RCA + pCLA3 + FA+DFF2的Design。

上表最後兩個架構測出的CLK一樣快(critical path為同一條)，因此選用mos數較小的design，layout起來面積也會比較小。

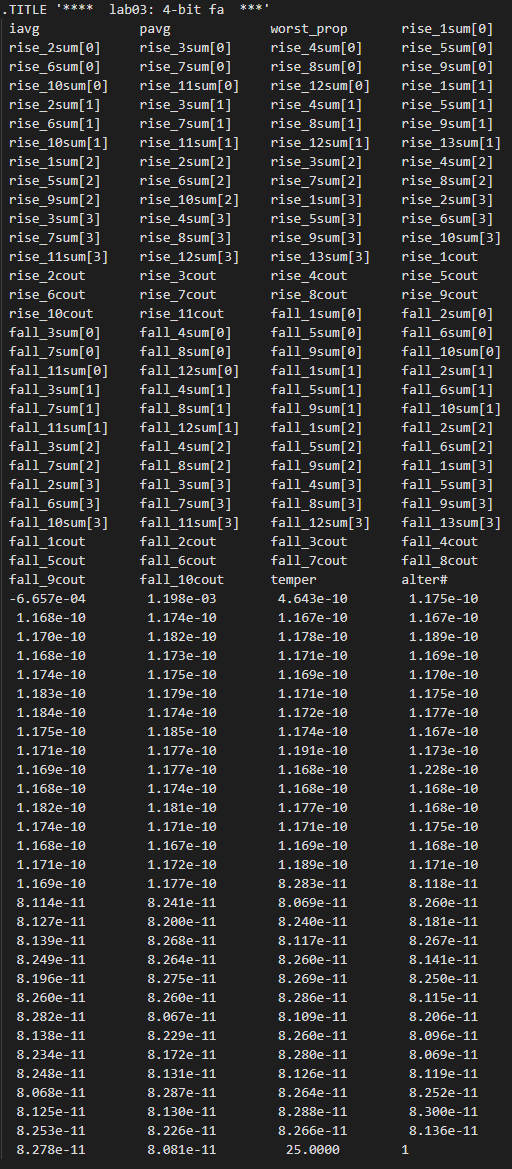
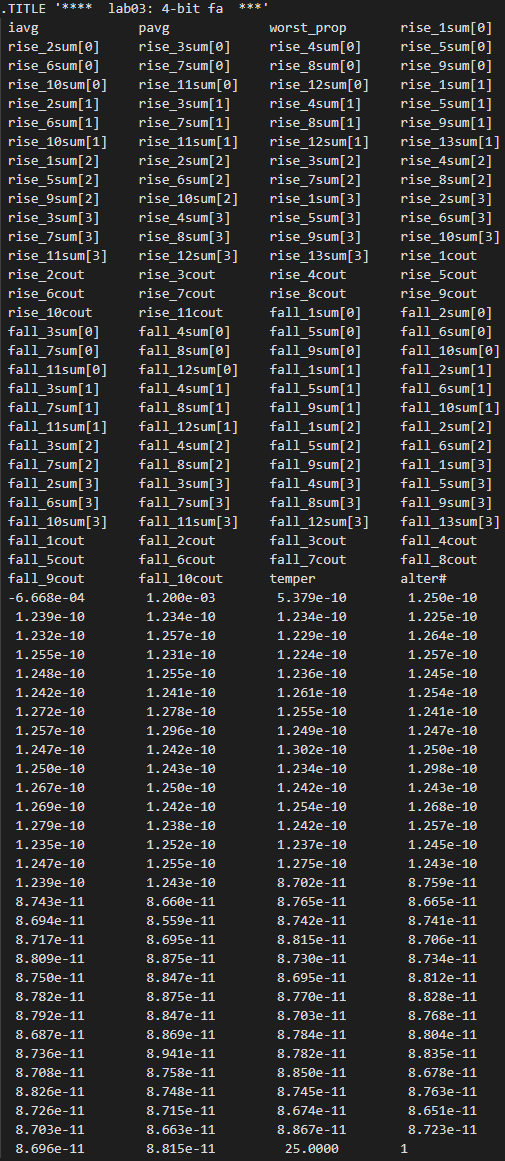
1. **Simulation result**
2. **Output waveform**
3. Pre-sim (Output waveform)
4. Post-sim (Output waveform)
5. Performance list (TT case under worst case input pattern)

|  |  |
| --- | --- |
| Maximum operation frequency | Pre-sim: 1.923 GHz |
| Post-sim: 1.695 GHz |
| Average power | Pre-sim: 1.1982m |
| Post-sim: 1.2002m |
| Layout area | 16.78 \*17.91 = 300.53 () |
| 4-bit full adder structure | Pseudo-NMOS C-CMOS Adder |
| Glitch control (Yes/No) | Presim Yes(< 0.2V) / Postsim No |

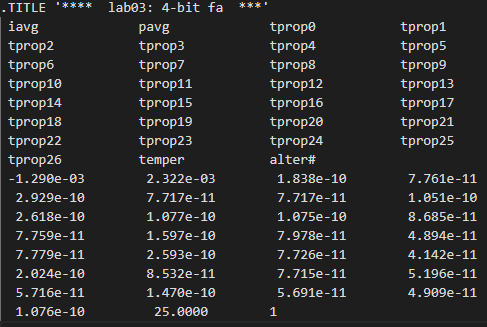
Post-Sim have Glitch Control( < 0.2V) when CLK period >= 0.62ns

**Table 1:** Output Simulation Summary

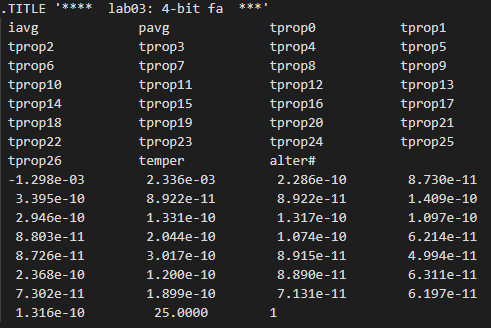
|  |  |  |  |
| --- | --- | --- | --- |
|  | **Spec.** | **Pre-sim** | **Post-sim** |
| **Worst Rise Time** | < 0.8ns | 122.8 ps | 130.2 ps |
| **Worst Fall time** | < 0.8ns | 83.0 ps | 89.4 ps |
| **Worst Propagation Delay** | N/A | 292.9 ps | 339.5 ps |
| **Average Power** | N/A | 1.1982 m | 1.2002 m |

Pre-sim rise/fall Post\_sim rise/fall

Pre\_sim tprop

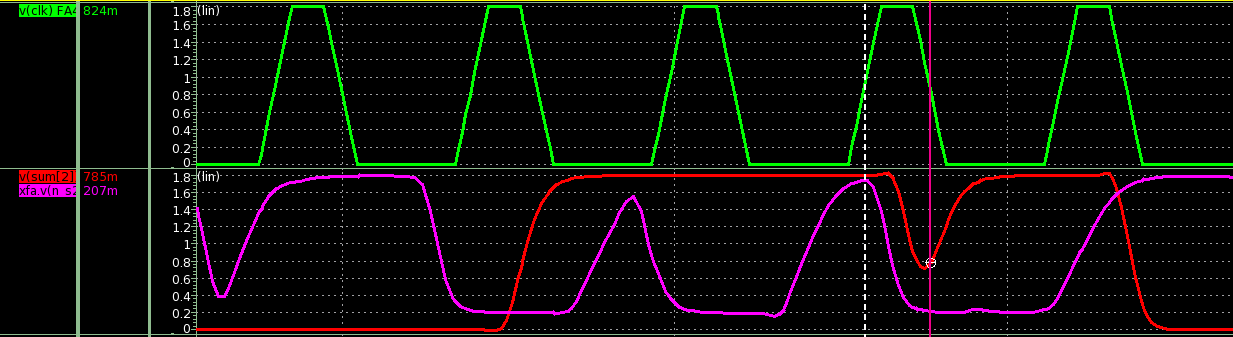


Post\_sim tprop



關於這次量測propagation delay的部分，理論上只需量測critical path的pro- -pagation delay，且若將clk period調到過小 (i.e. Period = 0.51ns for presim / 0.58ns for postsim)就可以看到是哪一組input產生error，進而找到worst pro- -pagation delay，不用每組input、output都量測。

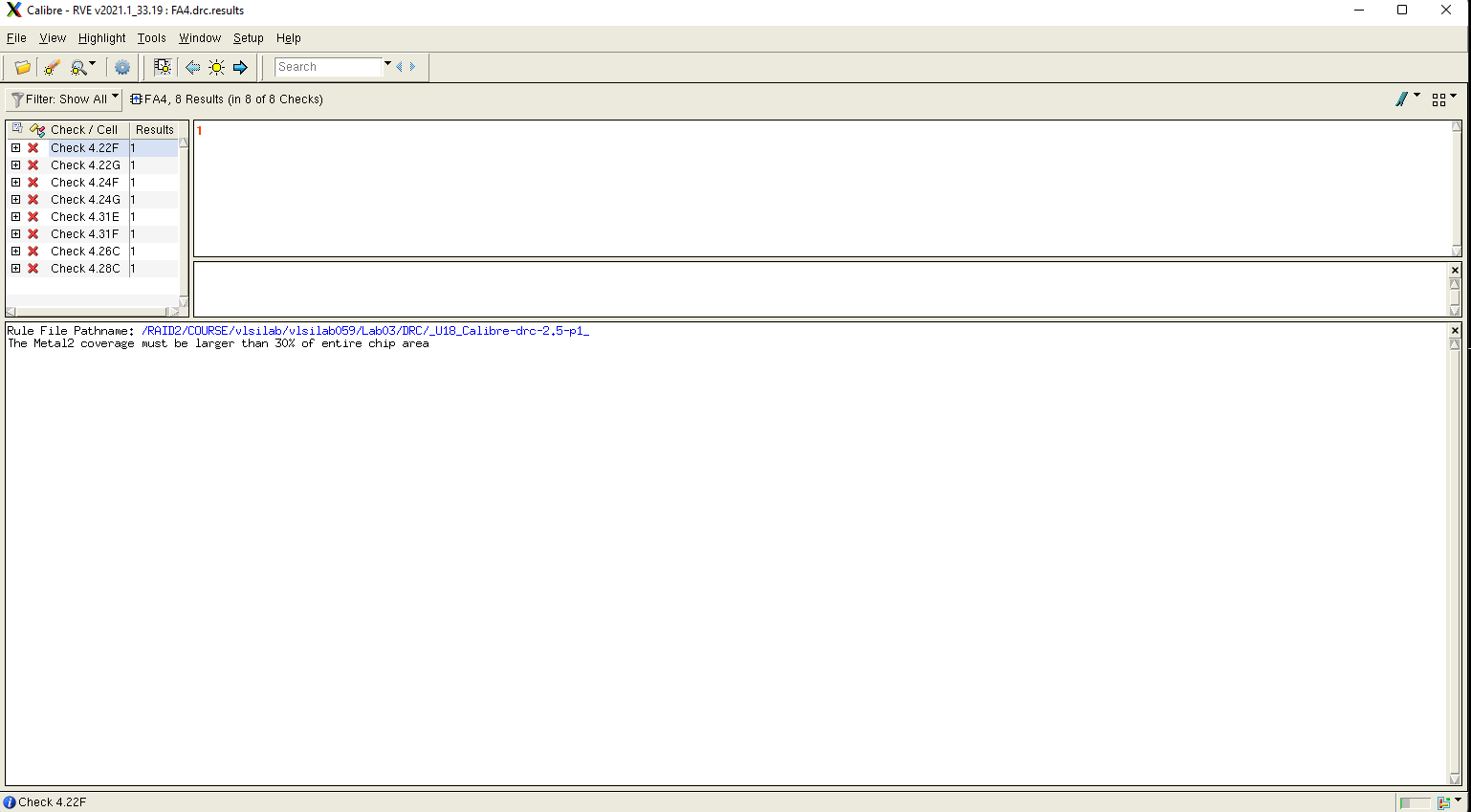
以上還是有附上每一組critical path tprop的量測，而最後也驗證了以上猜想是 正確的，worst propagation delay是在CLK 調過快出現的第一個err0處。

Glitch control :在Post-Sim的waveform上可以看到出現明顯的glitch如下圖。

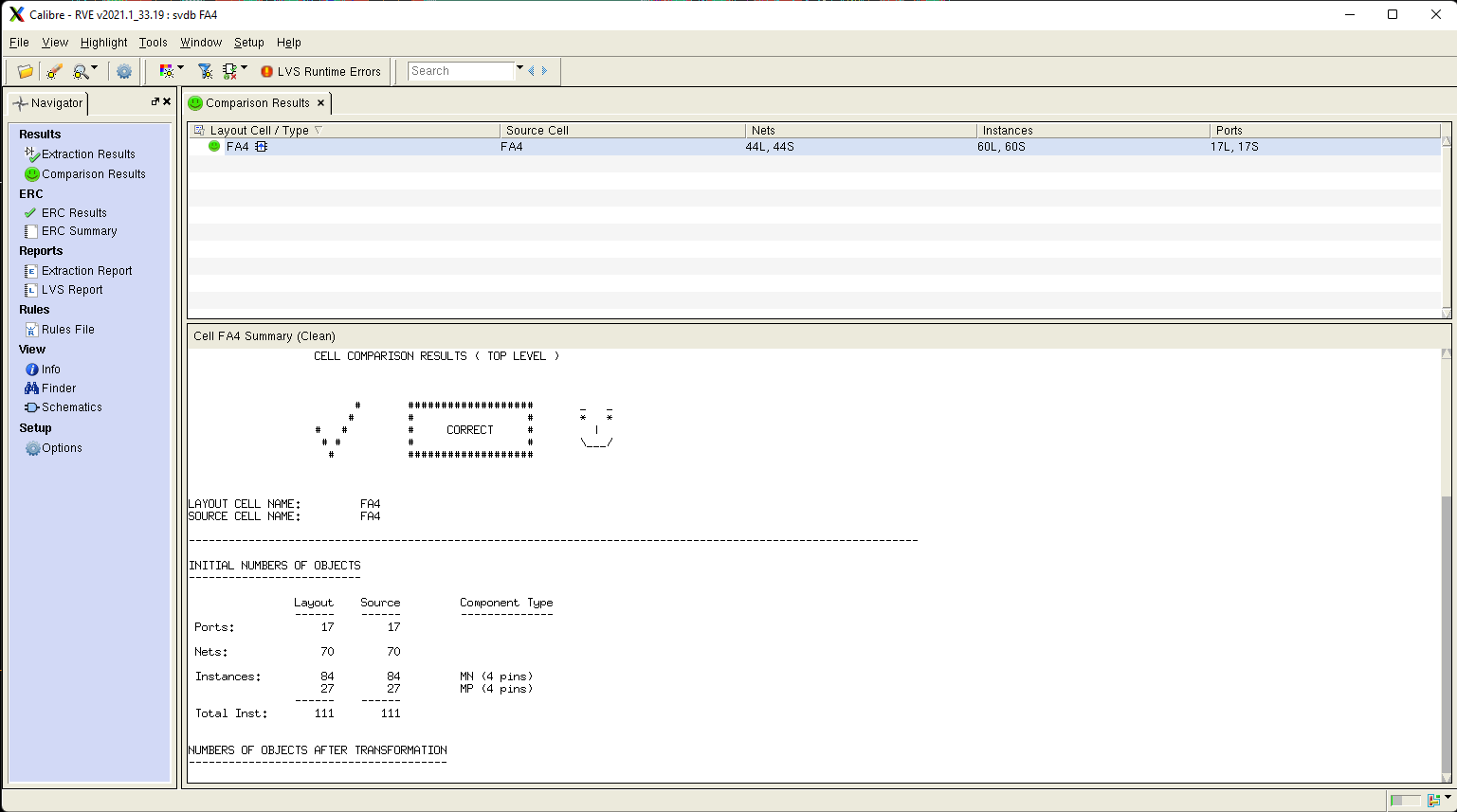
上圖紫色信號為SUM[2]\_bar，由於glitch剛好在clk posedge觸發的threshold， 因此紅色output被拉低，理論上如果架構為DFF，此結果會使output SUM[2] 在下一個clk posedge前都處於logic 0的狀態、導致err0。但由於今天使用的 是high level trigger的D-Latch架構，因此多了將近195ps的時間去使output 拉回正確的數值，才出現了有glitch但沒有error的情況。

如果將Post\_Sim CLK period調到0.62ns，glitch的情況就會消失。

1. **Verification result**
2. **DRC**



1. **LVS**

****

1. **Discussion**
2. **Compare Full Custom to Cell-based design and list their pros and cons.**

Full-custom design : Logic cells、circuit到最後layout全部由工程師設計。

Pros : 設計上有較高的彈性，可以達到最optimal的面積和performance

Cons: 設計流程繁瑣、成本高，大規模晶片整合time-to-market不切實際

Cell-based design :使用已經預先設計好的component (stdcell)作為building block。而layout的拉線部分是customize (大規模由EDA做routing)。

Pros : 設計速度快，較符合time-to-market。由於stdcell通常高度固定， 很方便由EDA tool去做place & route。因此對於大規模的晶片設 計會使用cell-based design + EDA。

Cons: 由於stdcell已固定，因此無法達到最優化的performance和面積。 整合度不會像full-custom那麼高。

1. **How to eliminate performance variation between Post-sim and Pre-sim.**

Post-Sim的netlist因為經過layout LPE後會將繞線的導線、poly等寄生電 容電阻值考慮進去，因此layout的一些準則如果達到整體post\_sim的 performance會離pre-sim小一些。以下列入幾點 :

盡量多打contact。diffusion、GND、VDD、metal間銜接都需要contact。 而打一個contact相當於銜接處接上一個寄生電阻。因此打越多contact代 表並聯的寄生電阻越多=>等效寄生電阻值會趨近於零，減少IR drop。

避免latch-up。P-substrate/N-substrate電位要拉到GND/VDD。因此有 VDD GND拉線的部分，應盡可能加入diffusion去使body的準位固定。

能共用的diffusion tap應盡量共用，如此減少中間連線使用metal帶來的 delay。metal拉線有過長、或多附載的情形應考慮加寬，可以降低訊號 在傳遞上的IR drop太明顯。

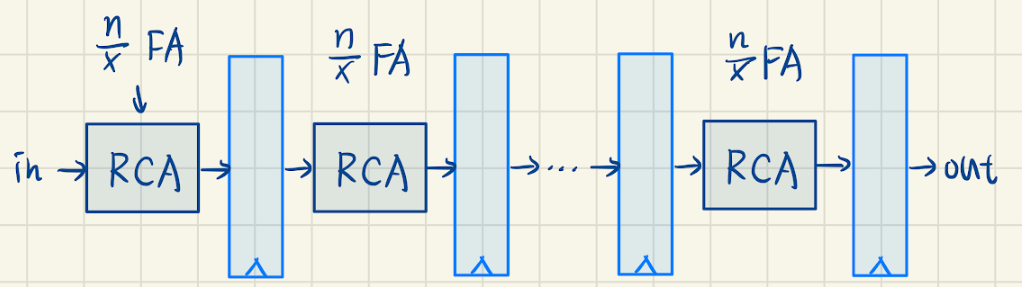
1. **Please discuss what has influence on speed while pipeline regs are placed at different location.**

如果考慮到要做出32bit、64bit以上的adder時，比起用非常龐大的CLA 電路，切pipeline可能會更實際一些。

假設今天做一個n-bit RCA，且做x級的pipeline :

可預想n-bit RCA的critical path就是由input經過n個FA到最後一級的 output。最好的pipeline可以將total Tdelay縮小x倍。意即每經過n/x個 FA後都在中間擋一層pipeline reg。每一級的pipeline時間會幾乎相等， critical path就被縮短了。(如下圖)

如果今天切pipeline十分不平均，有的大有的小，可以推測出critical path 會被pipeline最大的那一級latch住，如此反而在切同樣級數的pipeline下 得到較差的performance。



1. **Summary (Optional)**

此次實驗的設計過程相較Lab02來說順利許多。基本上設計的flow都是 先參閱很多不同的paper、講義。在比較過不同電路的架構後，可以算出 該電路會經過的critical path、和大概經過幾級的logic gate等。初步判斷 後選出認為最佳的幾個架構(不確定彼此間的performance情況下)利用方 便的HSpice 跑presim去驗證自己的想法和選擇最終的電路架構。

在layout的部分，因為面積的判定包含N-Well，因此雖為full-custom的 design，我認為在畫stdcell的一些概念可以引進，也有助於優化面積。不 同的subckt之間，應該要盡量使PMOS network對接，如此N-Well才可 以圈在一起。再來是GND、VDD的共用、加寬。基本上每個subckt我都 盡量使用metal1去繞線，這樣在銜接不同subckt的過程，才可以有很多 空間給metal2、metal3去做繞線。

這次lab比較可惜的是，一開始採用DFF (edge trigger)的架構。但後來對 DFF、D-latch的判定有些誤解，後來為了增加CLK frequency而不小心使 用了D-latch的結構。我認為在電路的design上(尤其現在是數位電路)， 不應該使用D-latch。在追求performance的同時，我認為要怎麼做出一個 很robust的電路更為重要。.18的製程下，如果今天我做出的4bit FA速度 極快、但會有glitch的發生，那我不如用較慢的CLK frequency，或是改 用更robust的電路架構，如此一來我設計出來的adder還會更實用一些。

不過今天如果只是考慮製作一個單純的4bit FA，沒有要拿來做成其他電 路的building block，單純以full-custom的角度來說，這樣pseudo-NMOS 大power的問題、使用latch還是DFF就會比較不那麼重要。單看今天電 路是扮演甚麼樣的一個角色吧。

總之，此次的Lab還是滿好玩的，尤其是最後看到自己猜想的架構能正常 運作，還是很有成就感的 !