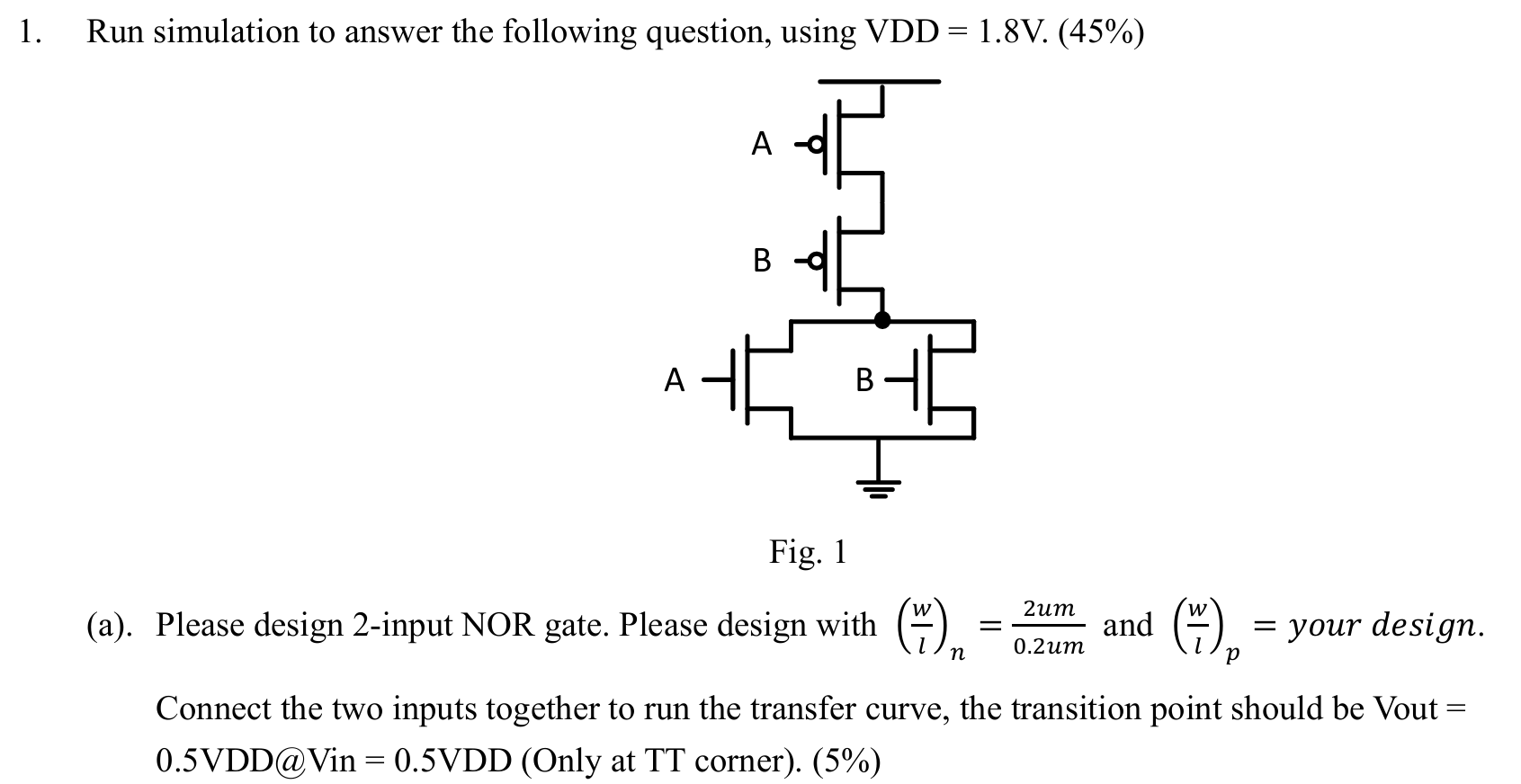
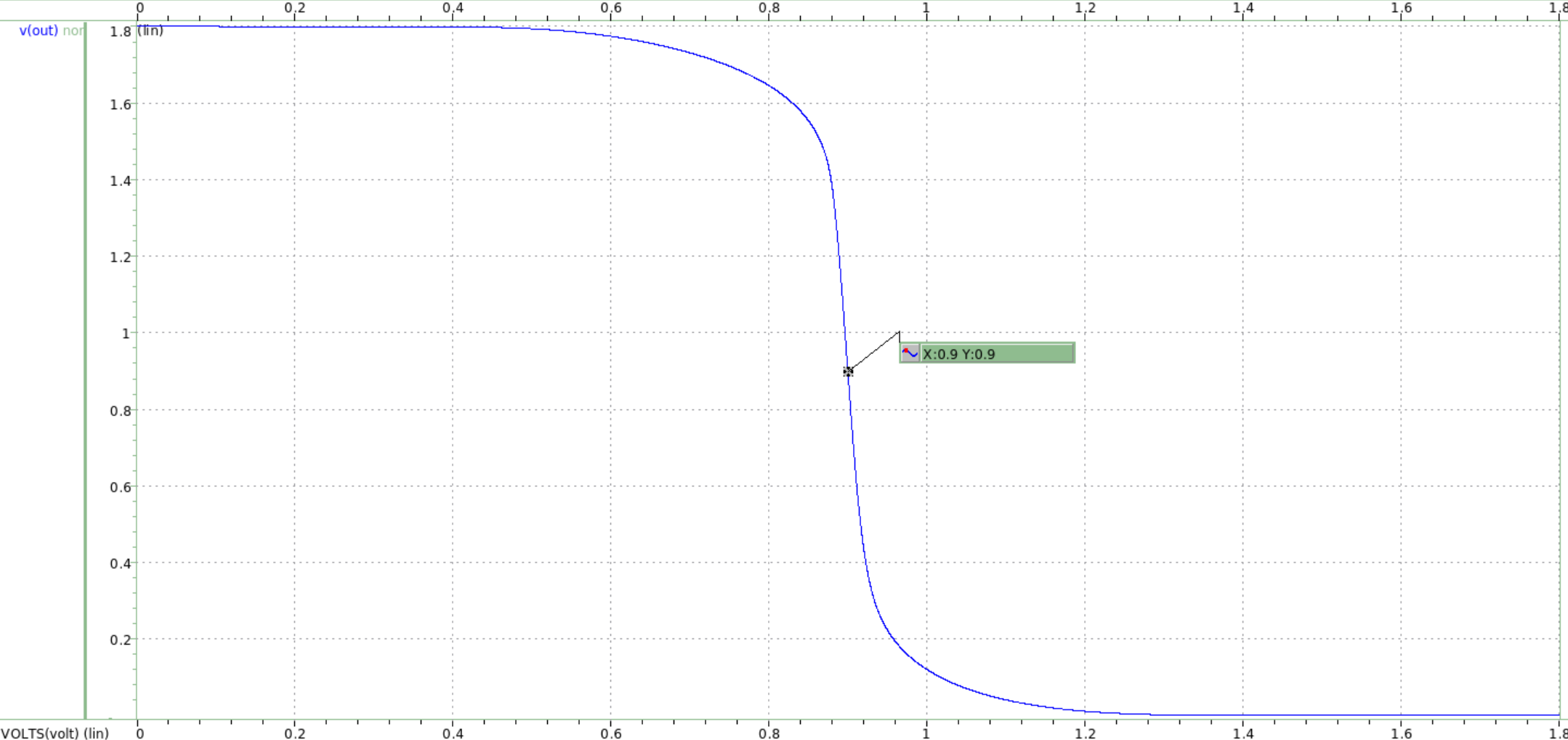
HW2

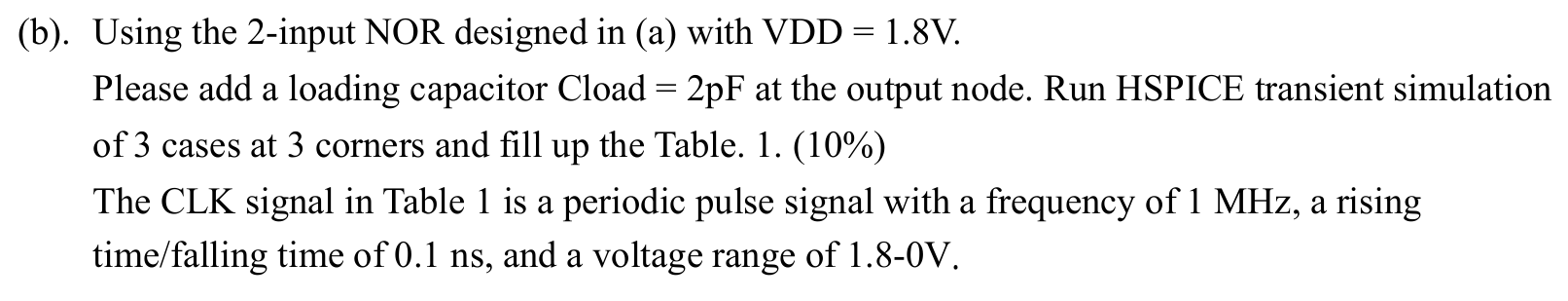
Student ID: 113063572

Name: 王品然



使用.dc掃過Wp參數，並使A, B接Vin固定在0.9V，找到Vout=0.9V對應之Wp數值。找到的。transfer curve如下圖所示：



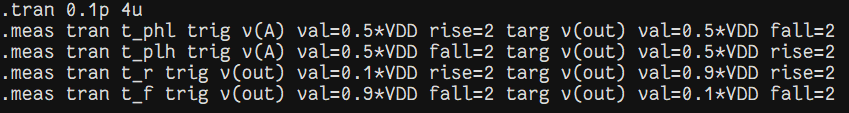


運用hspice中的.meas功能測量delay time與rise/fall time，過程與結果如下：

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | Case 1 | | | Case 2 | | | Case 3 | | |
| A | CLK | | | 0V | | | CLK | | |
| B | 0V | | | CLK | | | CLK | | |
| Corner | TT | SS | FF | TT | SS | FF | TT | SS | FF |
|  | 1.7516n | 3.1835n | 1.5271n | 1.6758n | 3.0197n | 1.4646n | 856.6141p | 1.5396n | 750.1360p |
|  | 637.1964p | 1.1650n | 548.9481p | 621.9445p | 1.1322n | 536.9697p | 635.1035p | 1.1539n | 550.5755p |
|  | 1.3035n | 2.3541n | 1.1360n | 1.3031n | 2.3542n | 1.1357n | 1.3038n | 2.3547n | 1.1358n |
|  | 3.0885n | 5.8297n | 2.686n | 2.9922n | 5.6339n | 2.6007n | 1.4984n | 2.8236n | 1.3018n |

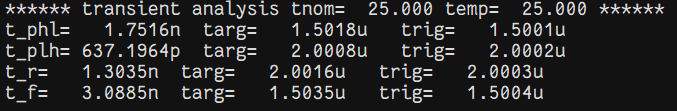
**Case 1:**

Measure command:

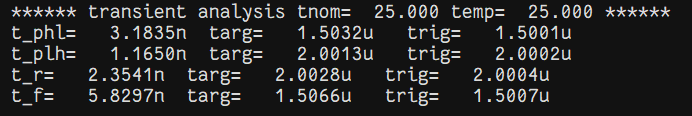


Results:

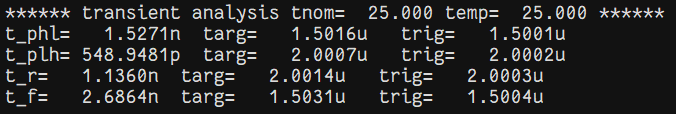
TT



SS

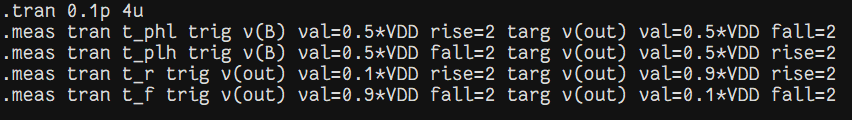


FF



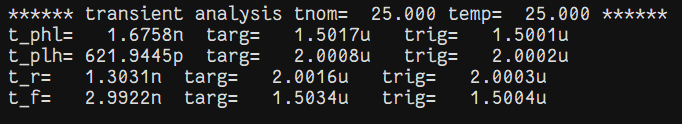
**Case 2:**

Measure command:

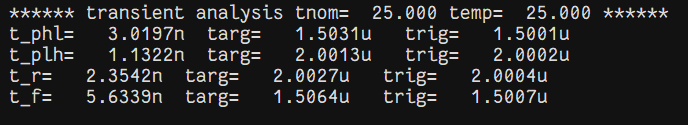


Results:

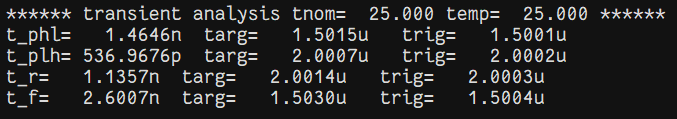
TT



SS

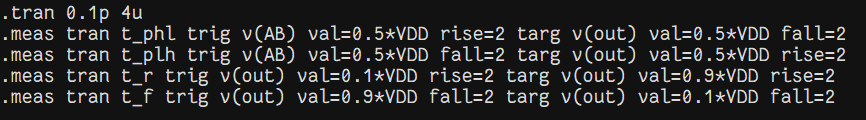


FF



**Case 3:**

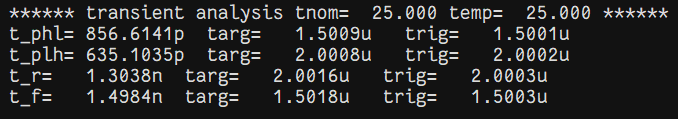
Measure command:



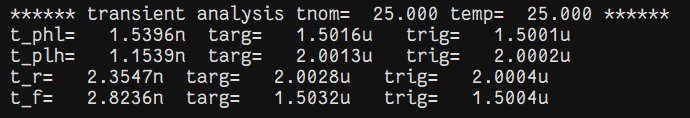
備註：把A, B黏在一起，叫做並稱此節點為”AB”

Results:

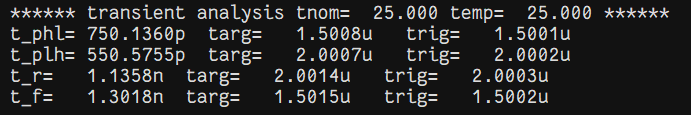
TT



SS



FF

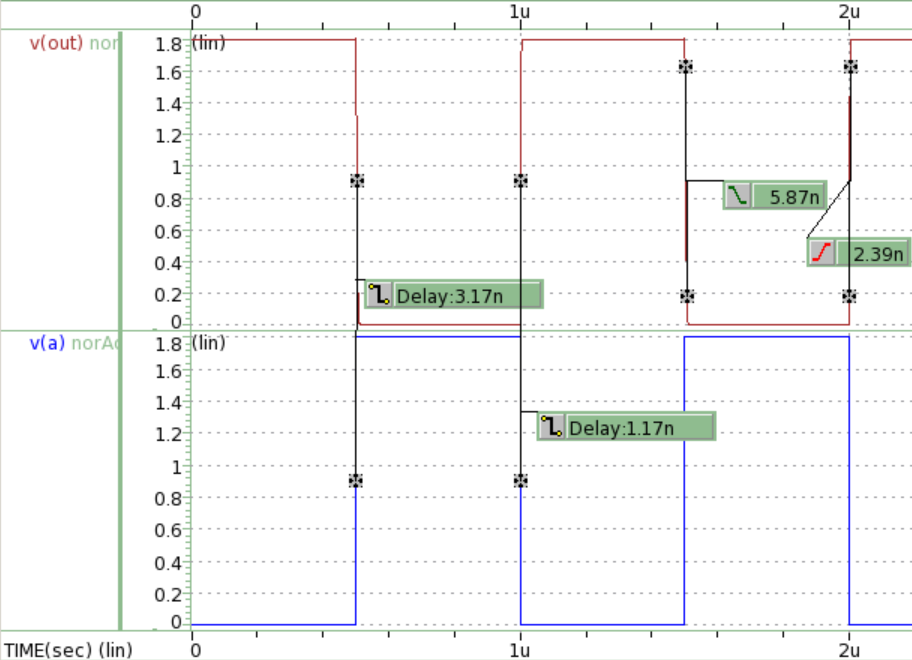
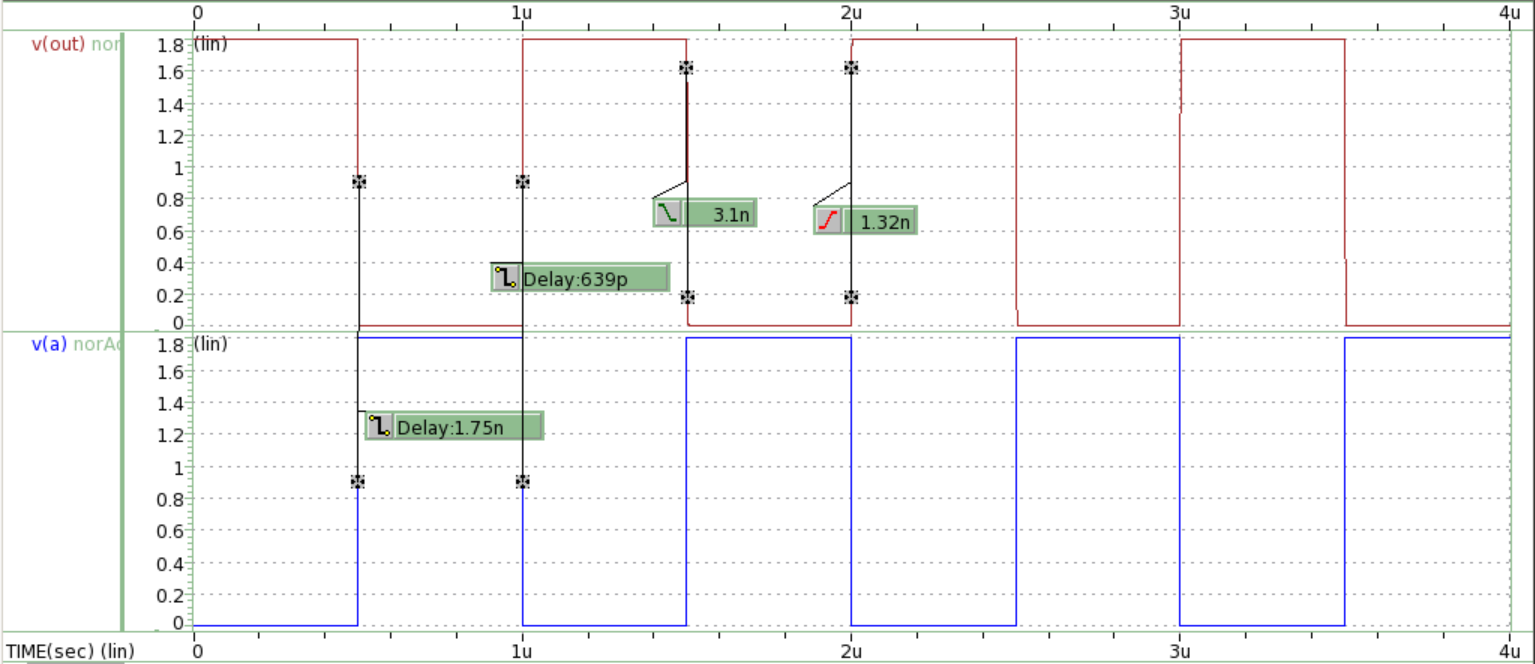
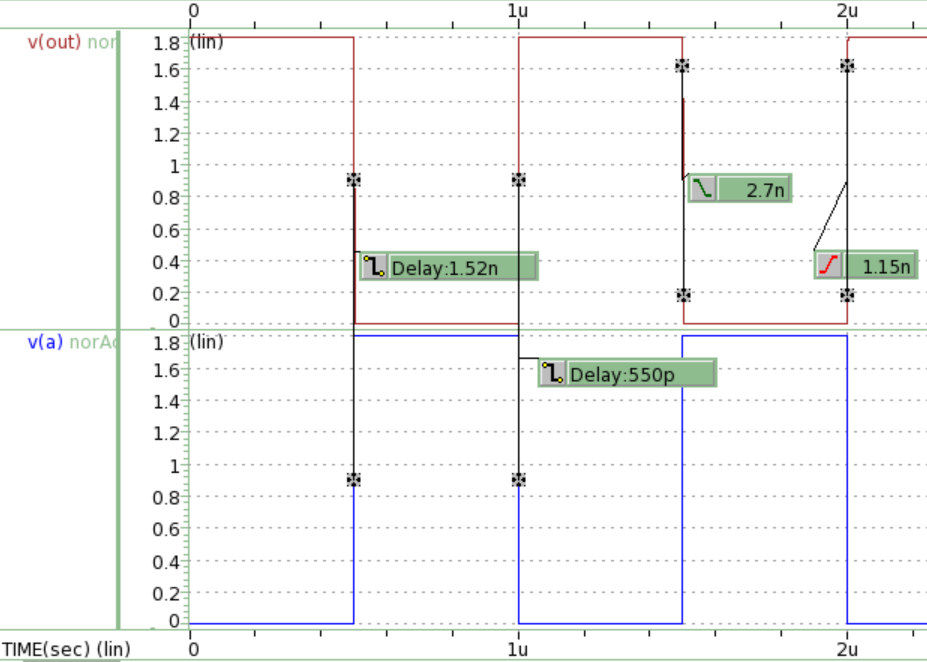


另外，我們也能運用custom waveview的測量工具，直接在圖上測量delay time 與rise/fall time做一個雙重確認，過程與結果如下：

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | Case 1 | | | Case 2 | | | Case 3 | | |
| A | CLK | | | 0V | | | CLK | | |
| B | 0V | | | CLK | | | CLK | | |
| Corner | TT | SS | FF | TT | SS | FF | TT | SS | FF |
|  | 1.75n | 3.17n | 1.52n | 1.66n | 2.98n | 1.45n | 848p | 1.52n | 743p |
|  | 639p | 1.17n | 550p | 627p | 1.15n | 542p | 642p | 1.17n | 556p |
|  | 1.32n | 2.39n | 1.15n | 1.35n | 2.46n | 1.17n | 1.36n | 2.49n | 1.18n |
|  | 3.1n | 5.87n | 2.7n | 3.03n | 5.72n | 2.63n | 1.52n | 2.88n | 1.32n |

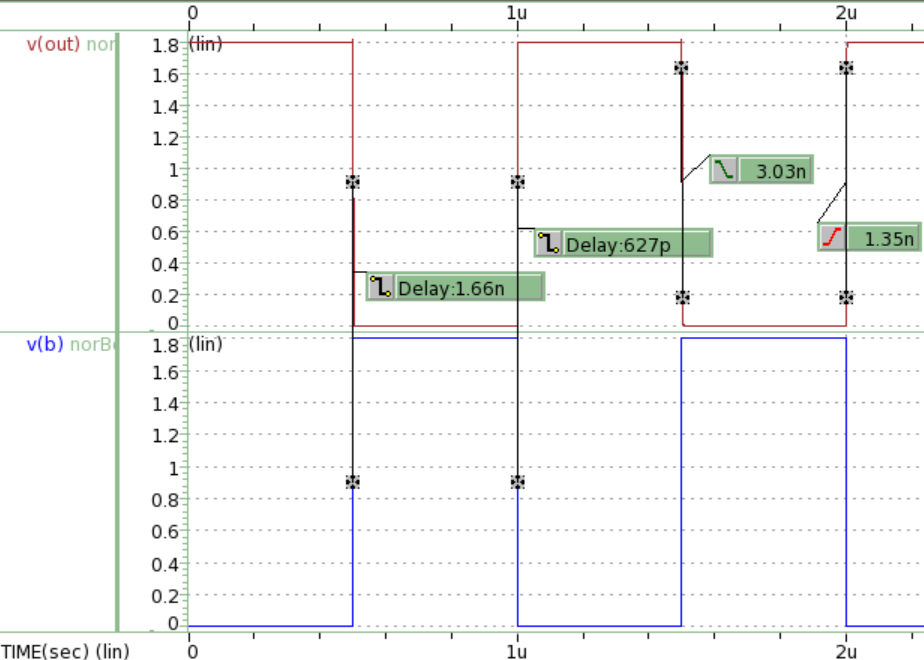
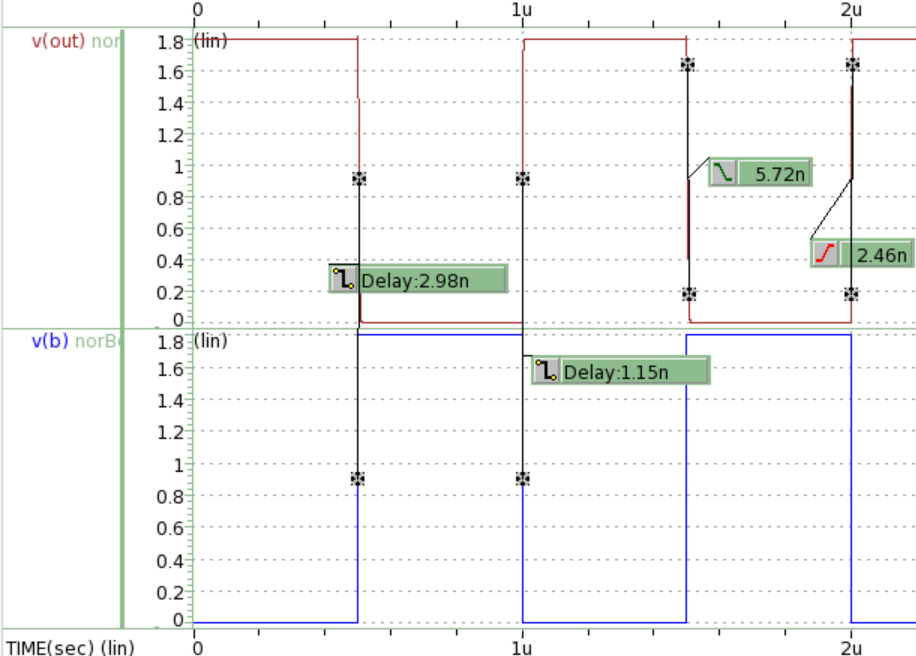
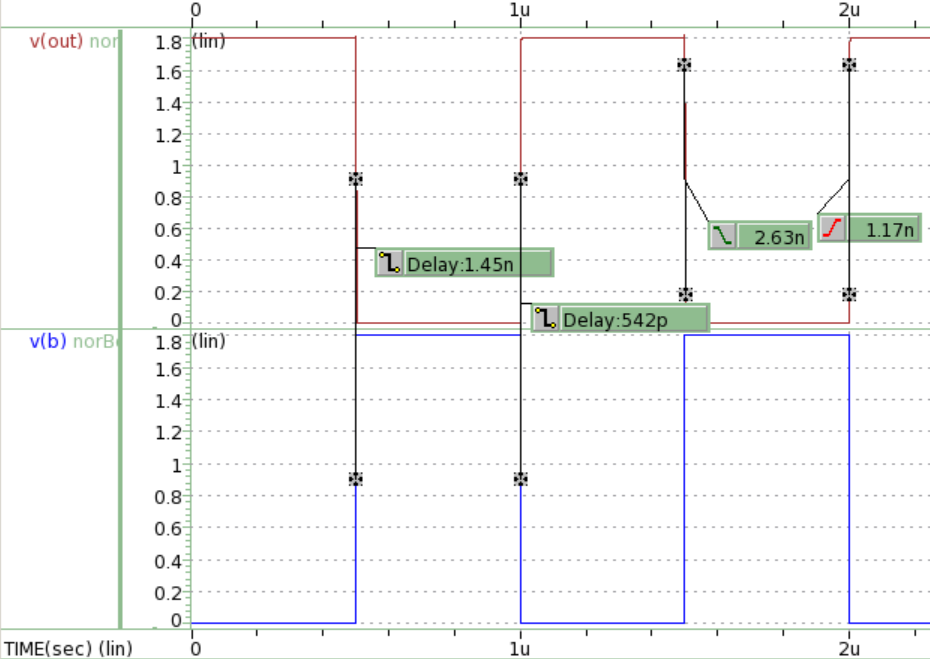
Case 1:

TT SS FF

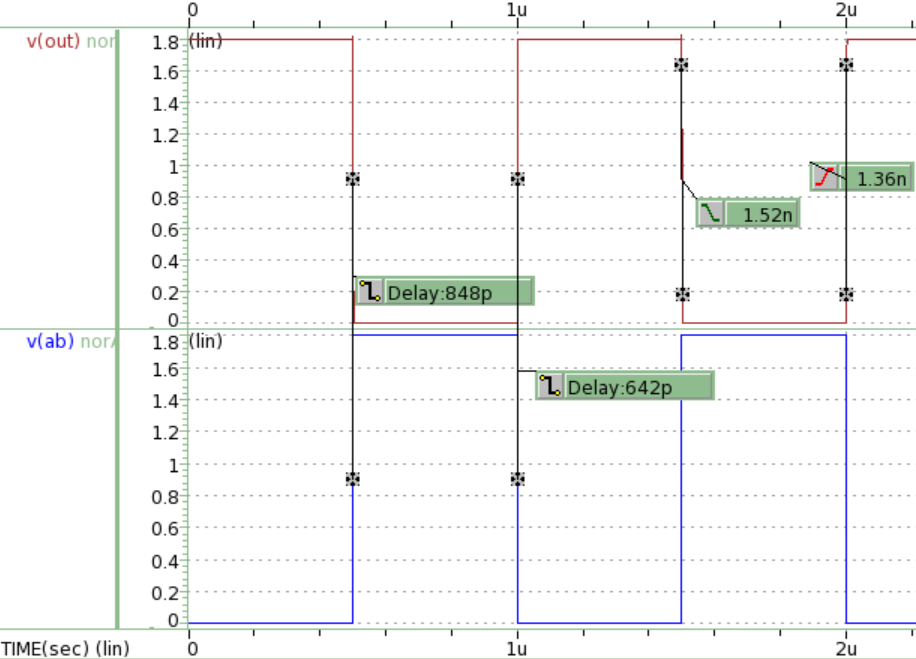
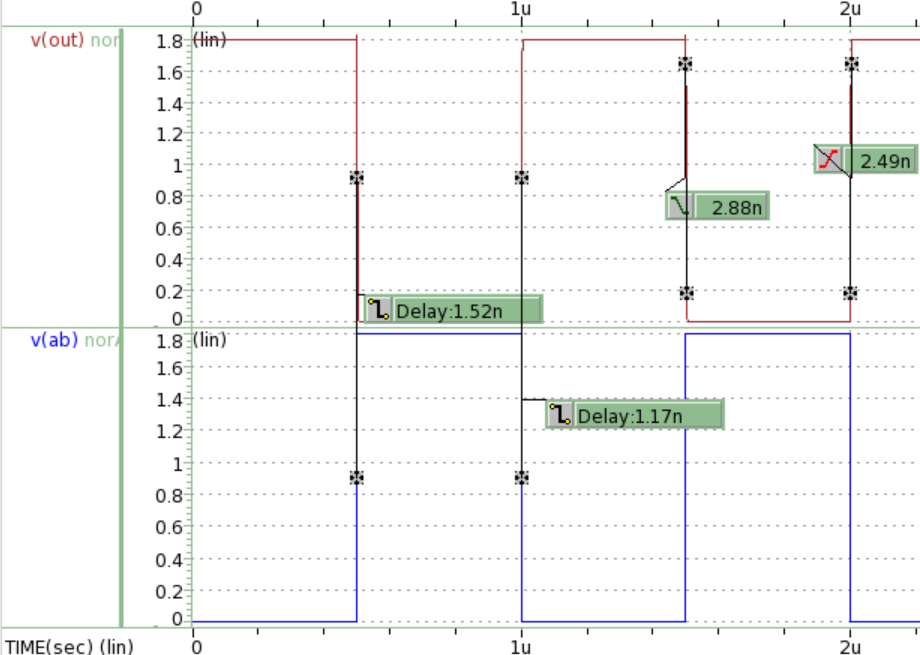
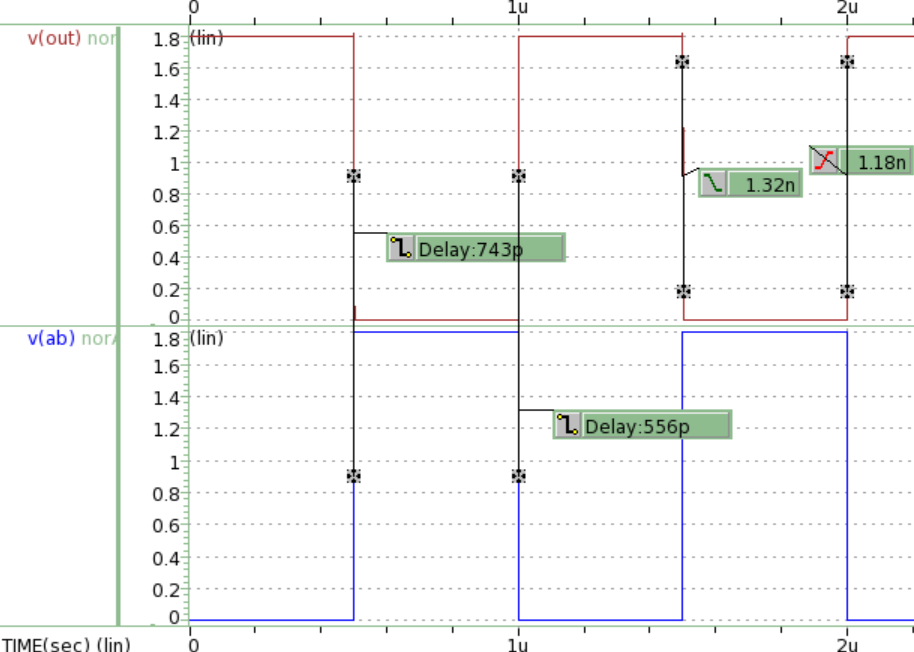
Case 2:

TT SS FF

Case 3:

TT SS FF

可以發現其實兩種測量方法的數值不盡相同，但誤差都很小，主要源於測算方法的不同，兩個結果都是設定.tran 0.1p 4u，但waveview給的數值只到1e-12的order，而相反的.meas給出的結果直接到了1e-16的order，測算的精確度也有差異。

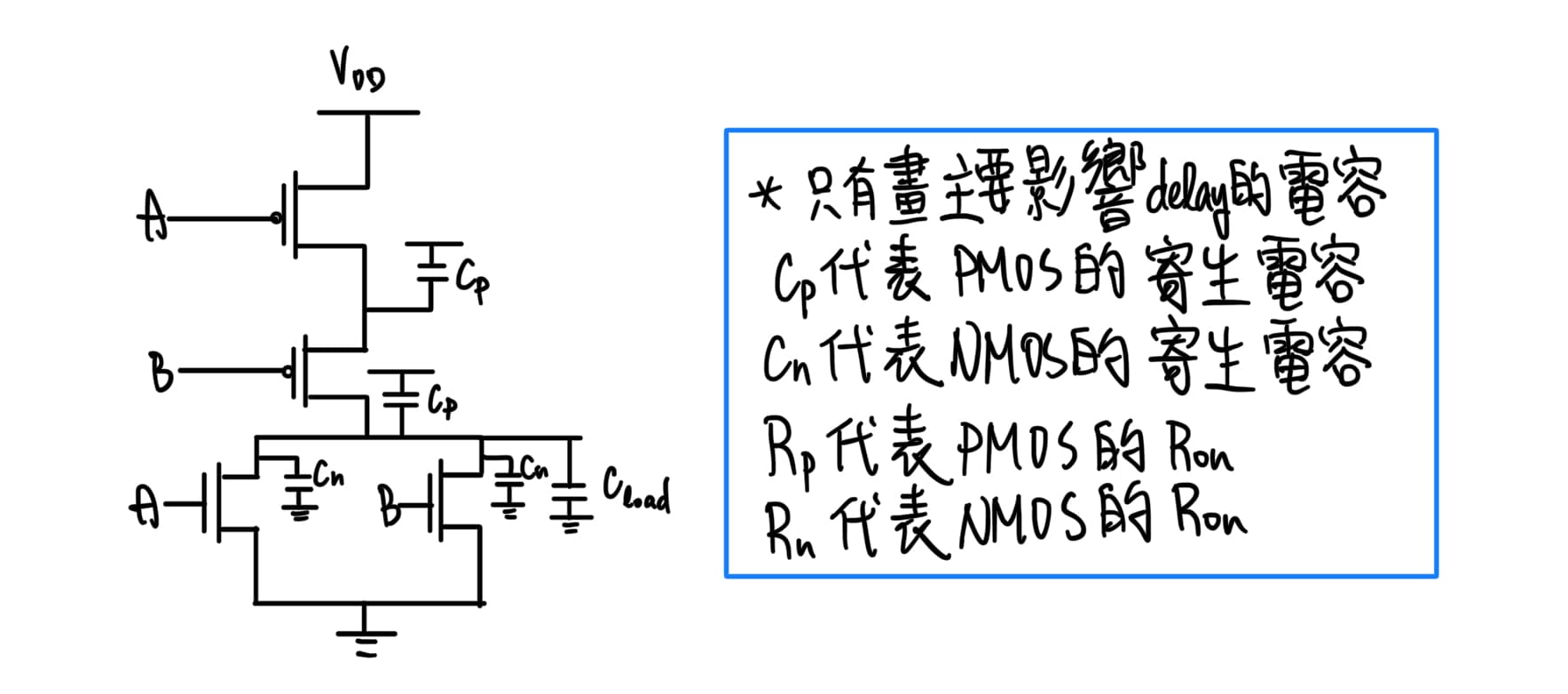


1. 同corner不同case
2. 和

在Case 1, 2中和則幾乎相同，因為充放電機制相同。至於Case 1, 2與Case 3的不同之處在於Case 1, 2是兩個訊號源，一個固定一個改變，而Case 3是兩個訊號源一起同向改變。Case 3的比起Case 1, 2快了差不多一倍，但卻沒有甚麼變化。造成不同case之間, 差距的原因可以藉由trace在訊號源改變時電路的行為得知。以Case 1作為Case1, 2的代表，當output從高電位被拉回低電位(fall)時，是因A從0變1.8V，會使得原本開啟的PMOS(A)關閉，並同時間開啟原本關閉的NMOS(A)，使得輸出被開啟的NMOS拉到0，注意此時只有一顆NMOS是開啟的(NMOS(B)被固定成關閉狀態)，因此只有一條電流在為負載和寄生電容放電；另一方面，Case 3是A, B訊號同時變化，使得在(A, B)=(0, 0)轉變到(A, B)=(1.8, 1.8)時，會有兩顆NMOS同時開啟，出現差不多兩倍的電流同時為負載和寄生電容放電，因此時間也差不多只要一半就可以完成，所以Case 3的比Case 1, 2的短了大約一倍。但當output從低電位要充電到高電位時，則是不管在Case 1, 2或Case 3中都是一樣的導通情況，都必須是兩顆PMOS同時開啟對電容充電，因此並無顯著的差別。

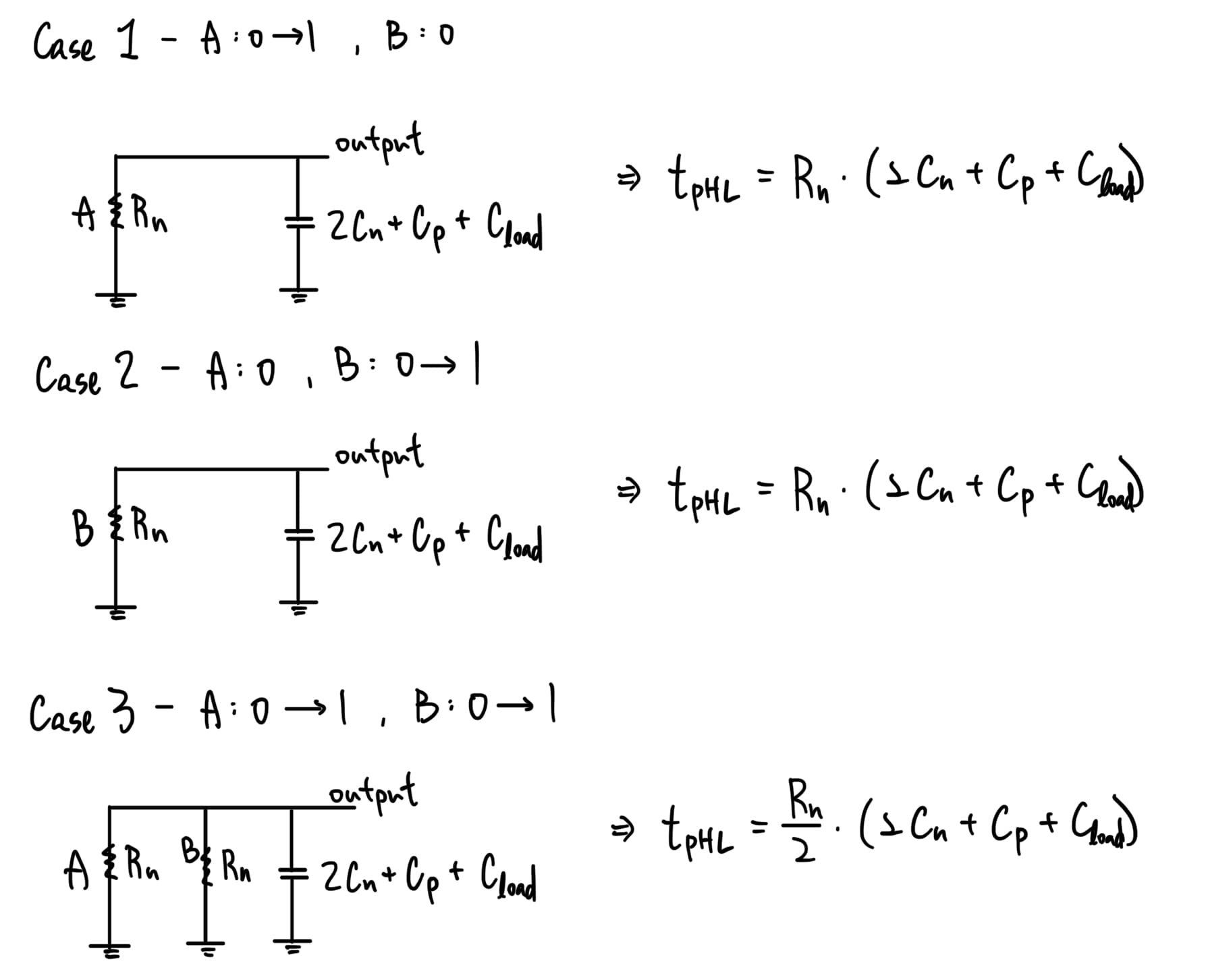
1. 和

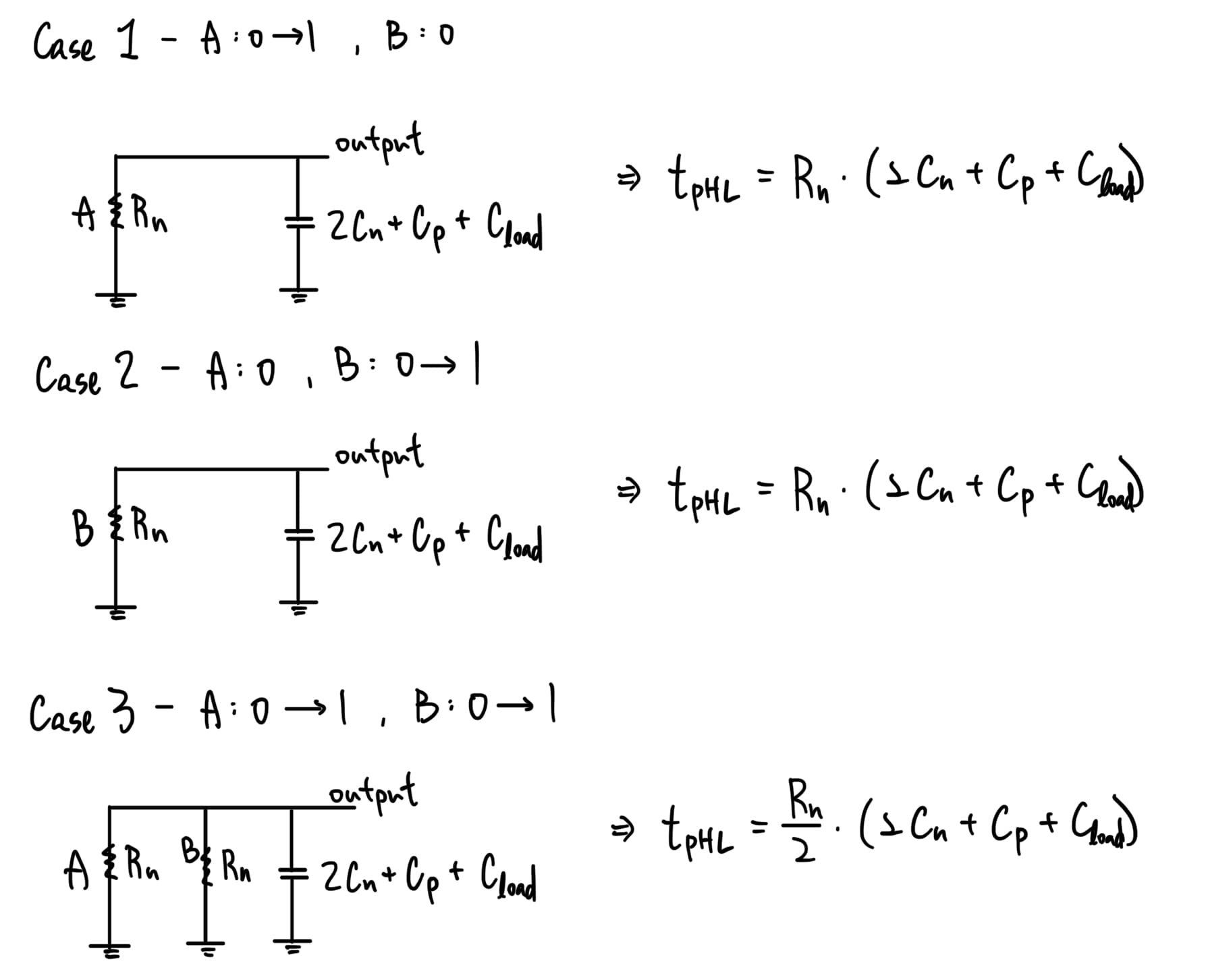
至於, 可以藉由估算路徑上的delay time得到，先把nor的負載與寄生電容畫出來如下圖：



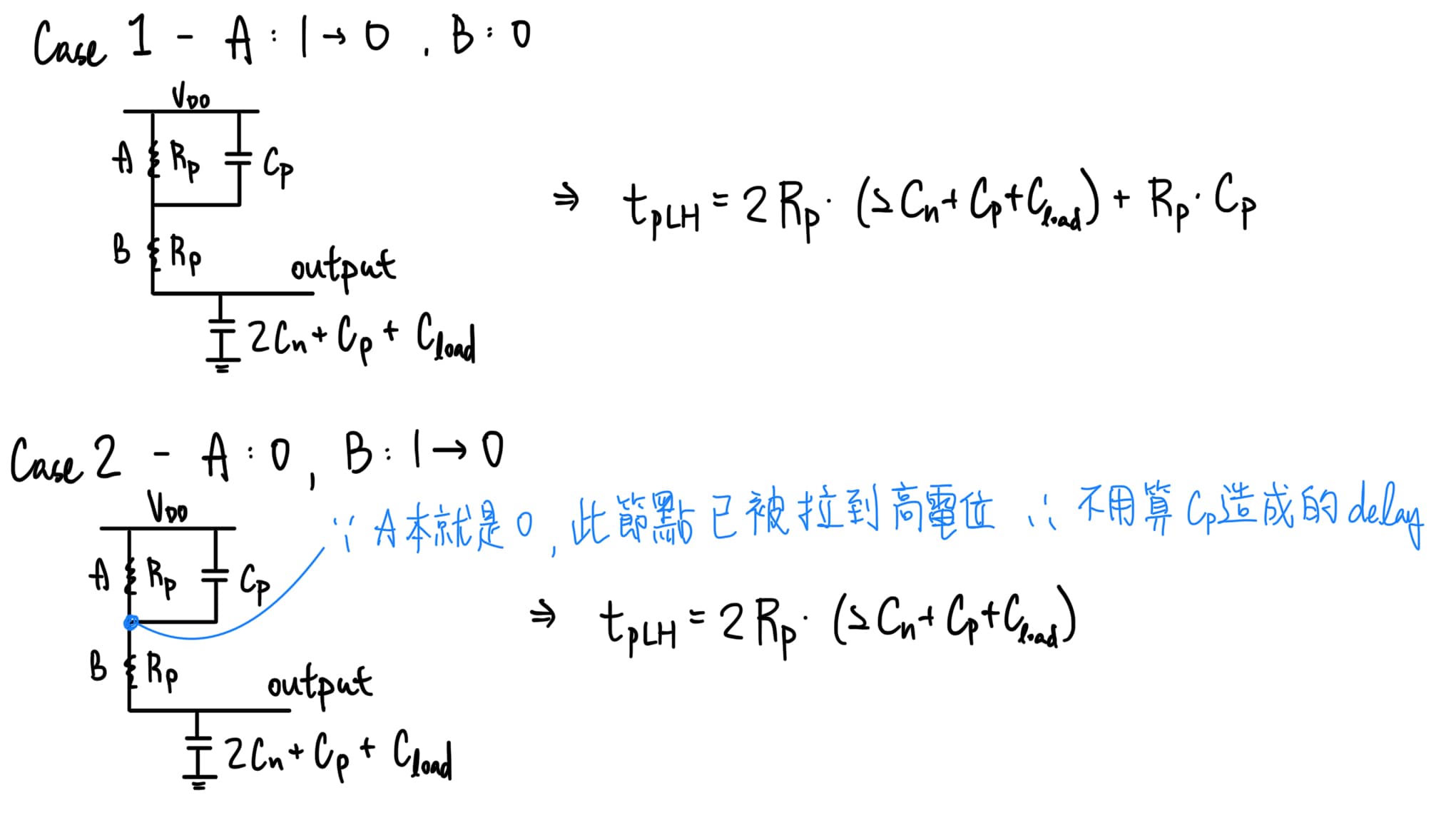
此圖沒有考慮NMOS在layout時可以合併drain的情況，不過不影響估算結論

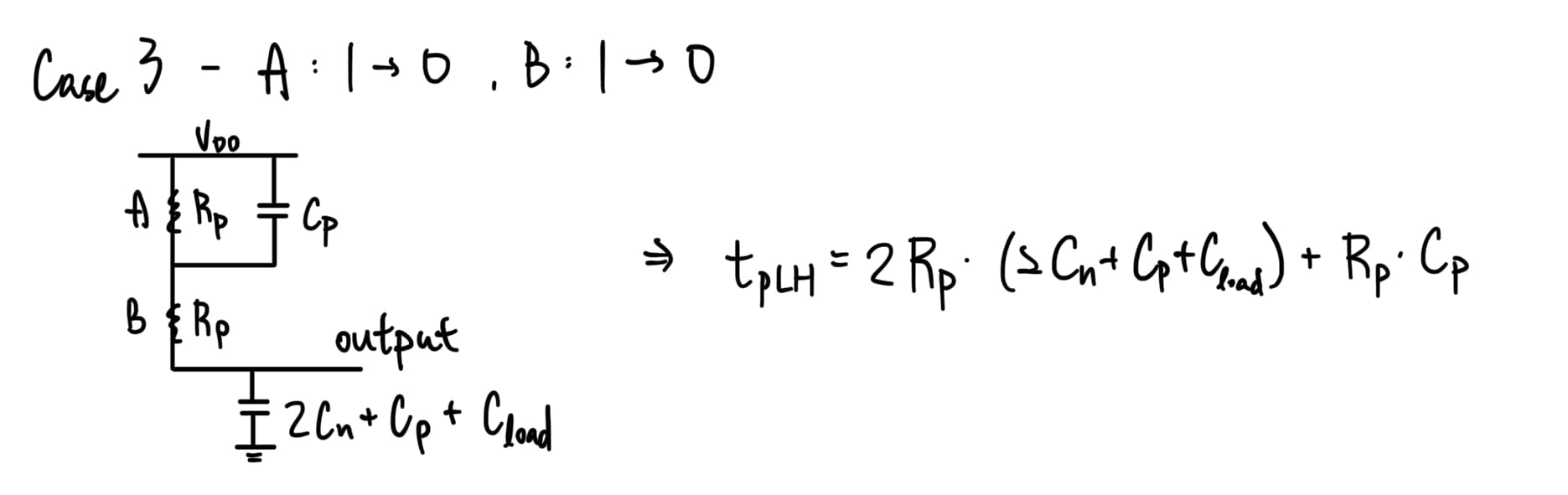
接著用Elmore model的方式估算delay：





可以由估算出來的結果看出Case 3的會是Case 1, 2的一半，因為Case 3同時有兩個NMOS開啟，使得電阻減半，delay減半，因此也可以說Case 3的是best case，也就是。這個估計結果與pre-sim結果相符。





可以由估算出來的結果看出Case 2的會略小於Case 1, 3的，原因如圖中藍筆所注釋的，所以儘管Case 2會是最短的delay，但也只短一點點而已，因為。因此也可以說Case 2的即為此電路的best case，也就是。這個估計結果與pre-sim結果相符。

1. 同corner同case
2. 和比較

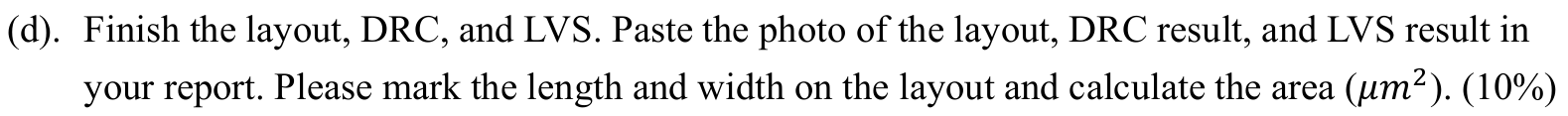
如上圖的手寫估算以及pre-sim的結果，可以清楚比較同case同corner下，皆大於，尤其是在Case 1, 2時最為明顯，因為遠小於(大約差14倍)，造成儘管會是的數倍(理論上是兩倍左右)，仍然小於數倍，又因為由I-V curve斜率倒數決定，而斜率倒數與成反比，所以最終會大數倍於，這最終導致即使多了一項(在Case 1的時候才有，Case 2沒有) 仍會遠小於。Case 3時，的因為並聯而相當於被砍半了，因此更接近，但仍舊大於，因為依照上面的推論不只是的兩倍而已。

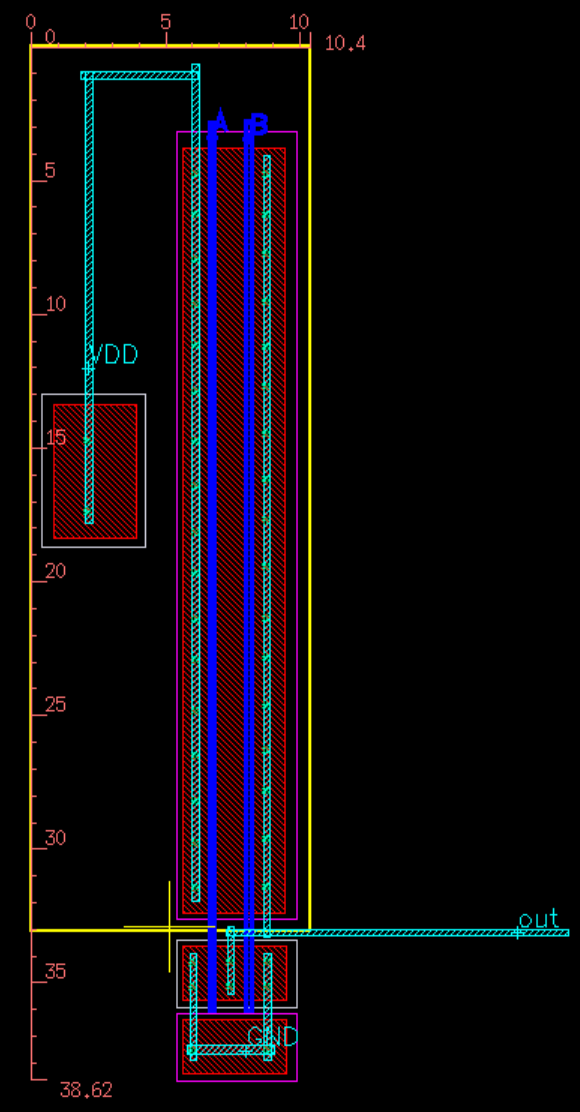
1. 和比較

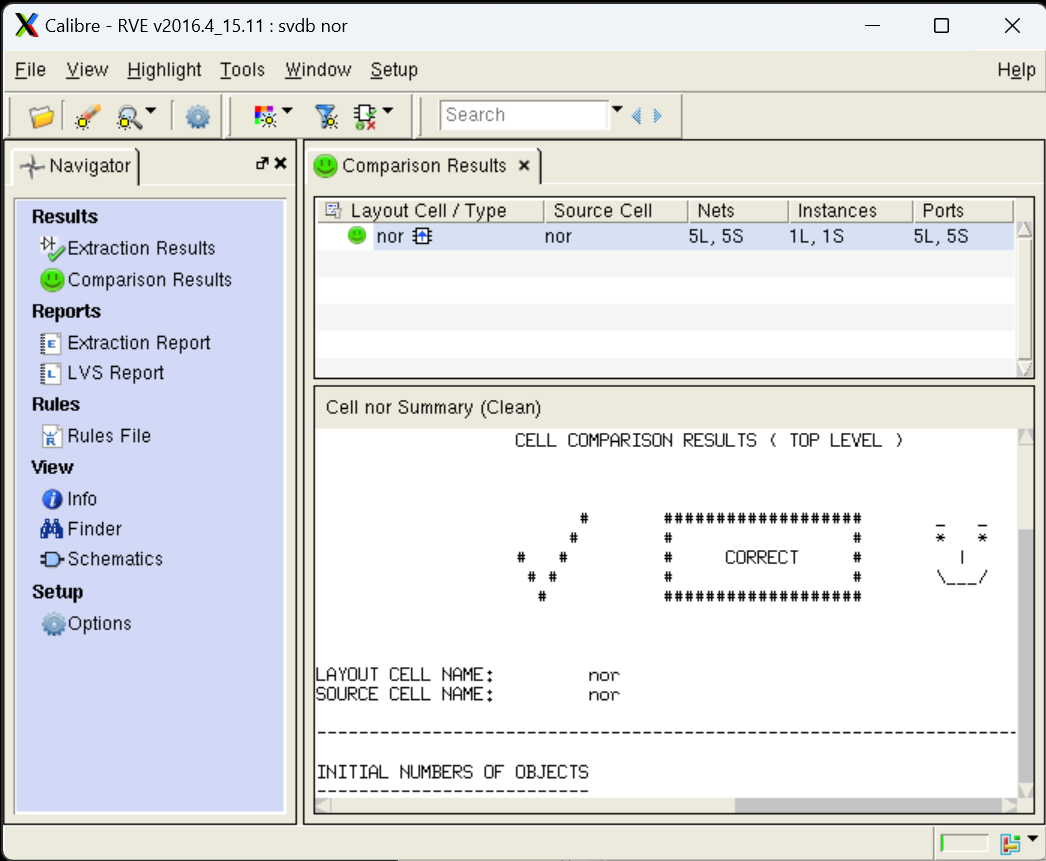
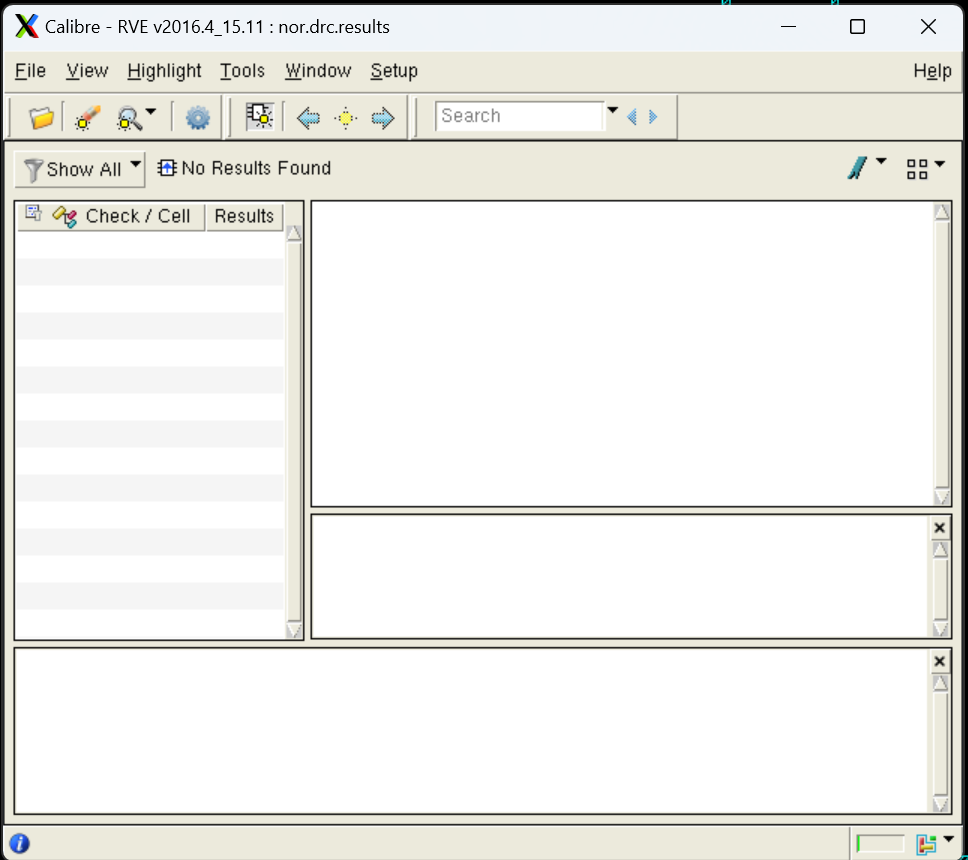
從pre-sim的結果可以清楚比較同case同corner下，皆大於，尤其是在Case 1, 2最為明顯。藉由，其中。I(t)與正相關，由上一段落的解釋可知，因此，由此可知，fall時的會大於rise時的，代表著充放電的速度，與充放電的時間呈負相關，因此最終可知大於。Case 3時，因為並聯的兩顆NMOS同時開啟，因此相較於Case 1, 2的等效增大兩倍，因此等效增大兩倍，增大兩倍，但仍然不及，就如同上面所討論的，小了不只兩倍，因此fall時的仍然大於rise時的，仍然大於，但會比Case 1, 2更為接近。

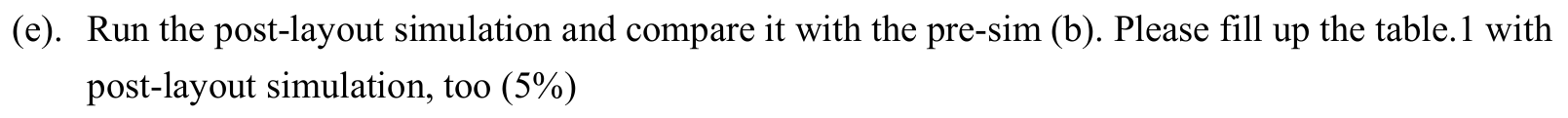
1. 不同corner同case

同個Case底下，SS時，因為PMOS和NMOS的載子移動速率都下降，所以delay time還有rise/fall time皆上升；反之FF時，因為PMOS和NMOS的載子移動速率都上升，所以delay time還有rise/fall time皆下降。至於TT則介於兩者之間。





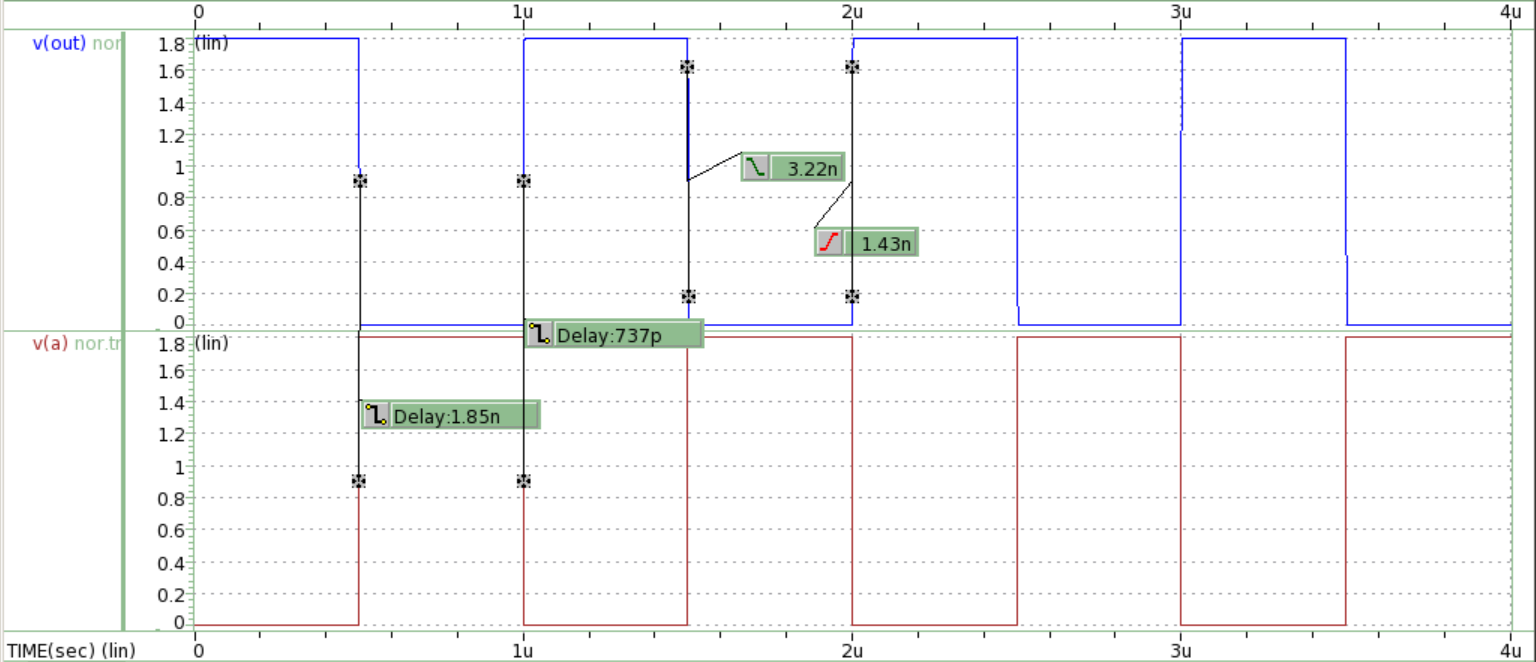




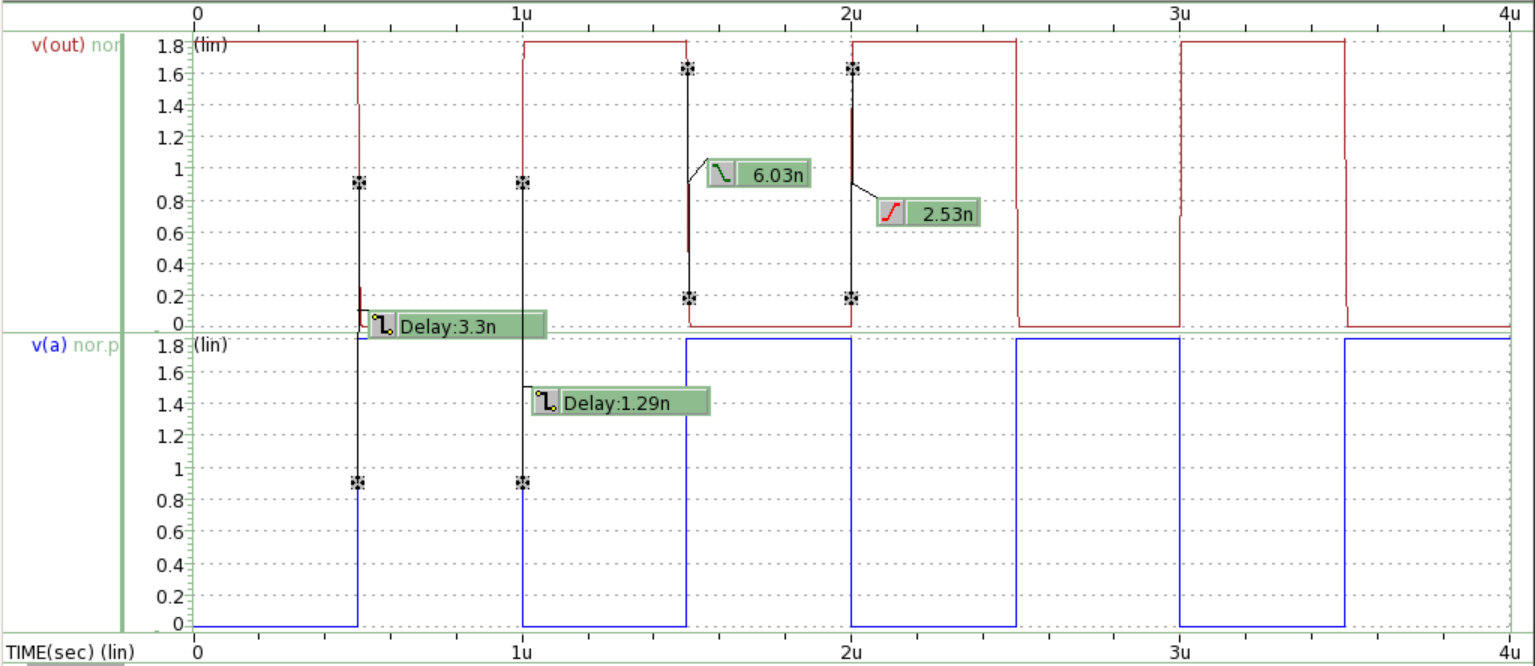
Post-sim後，以waveview重新測量3個corner，3個case分別的*、、*和，並填入表中如下：

Case 1

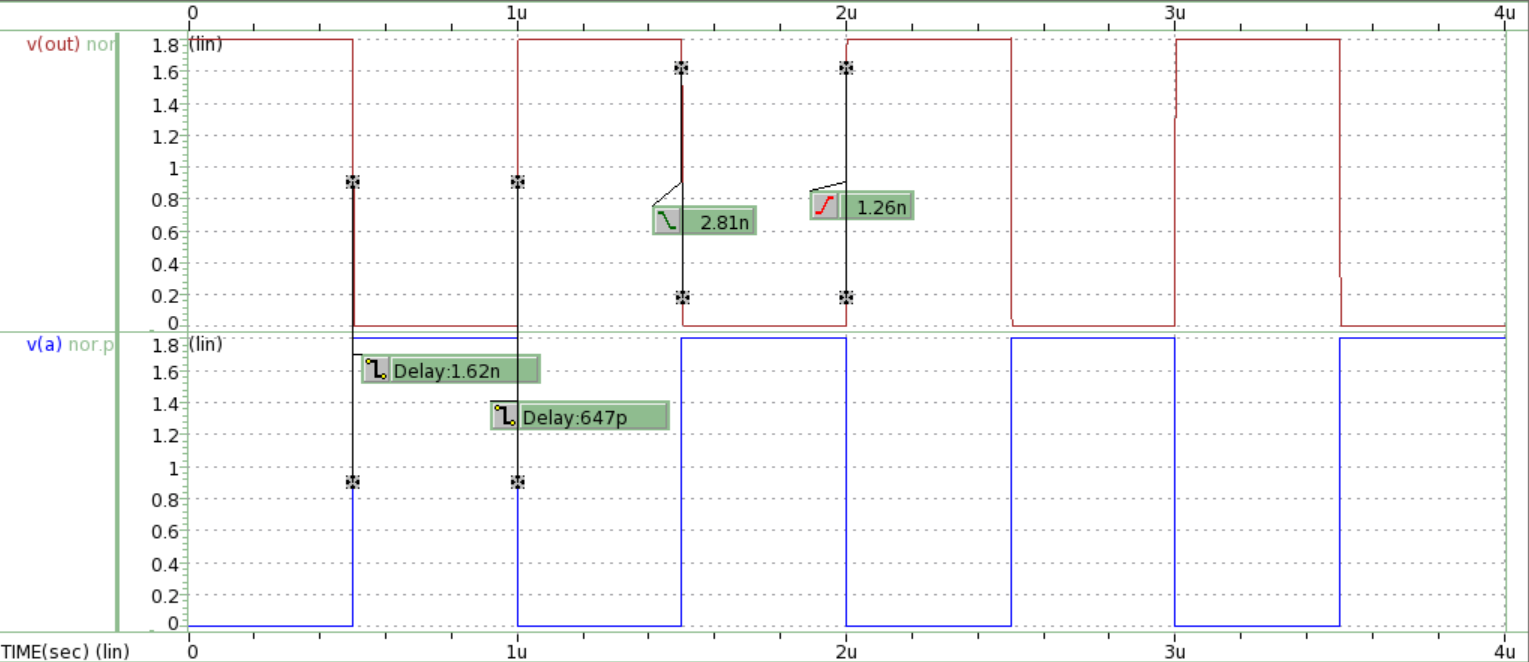
TT



SS

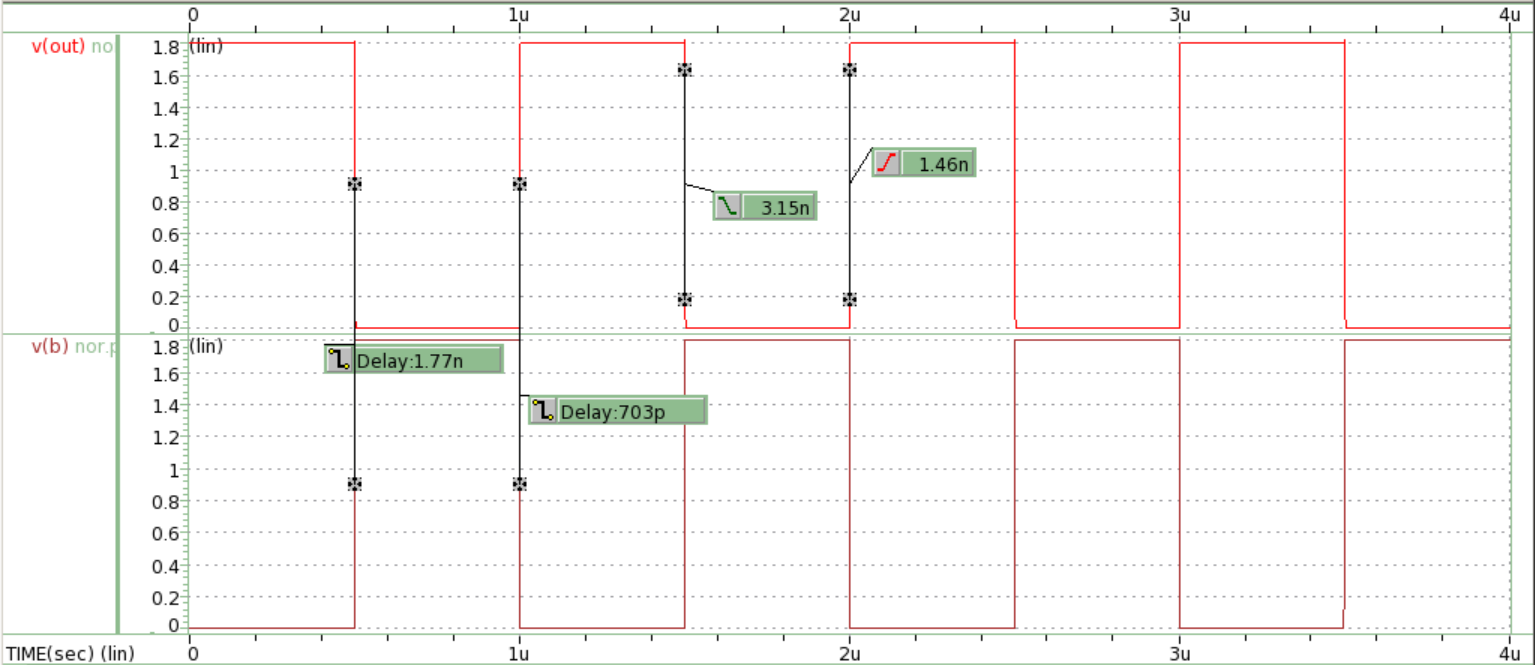


FF

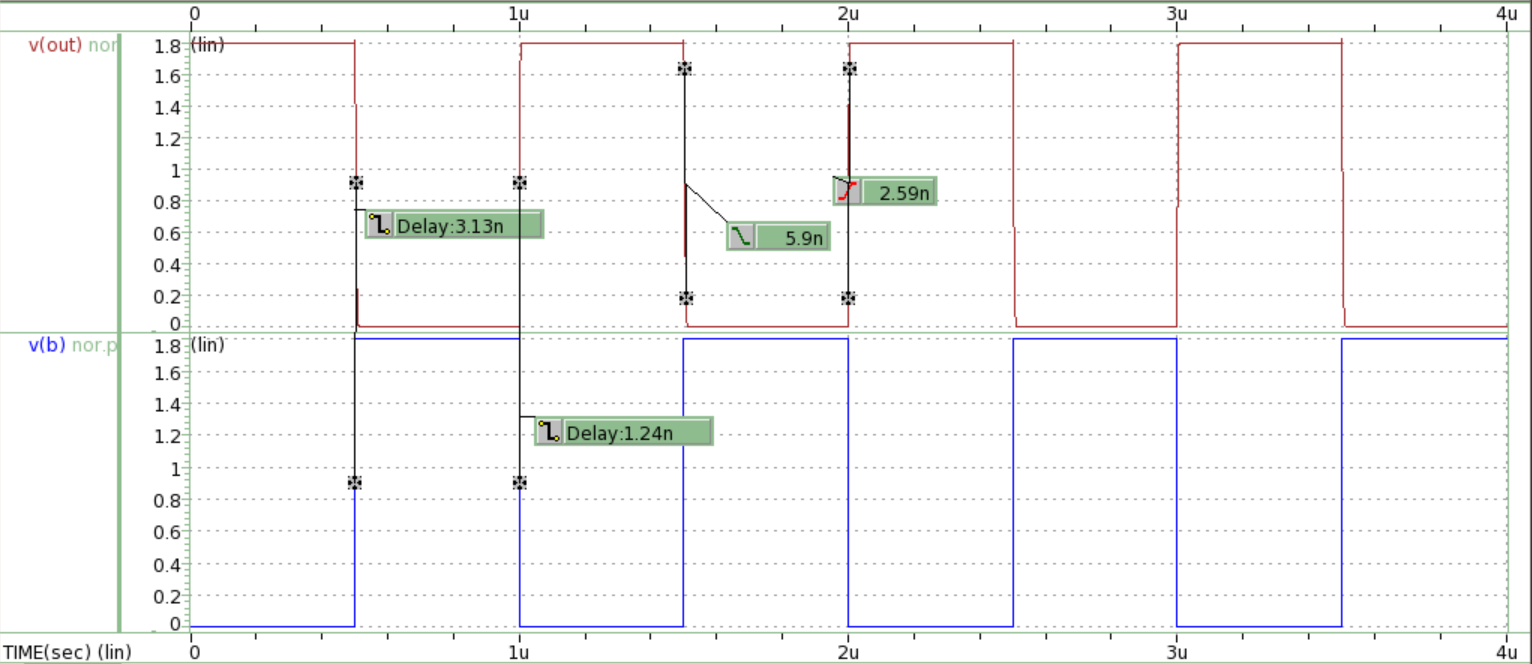


Case 2

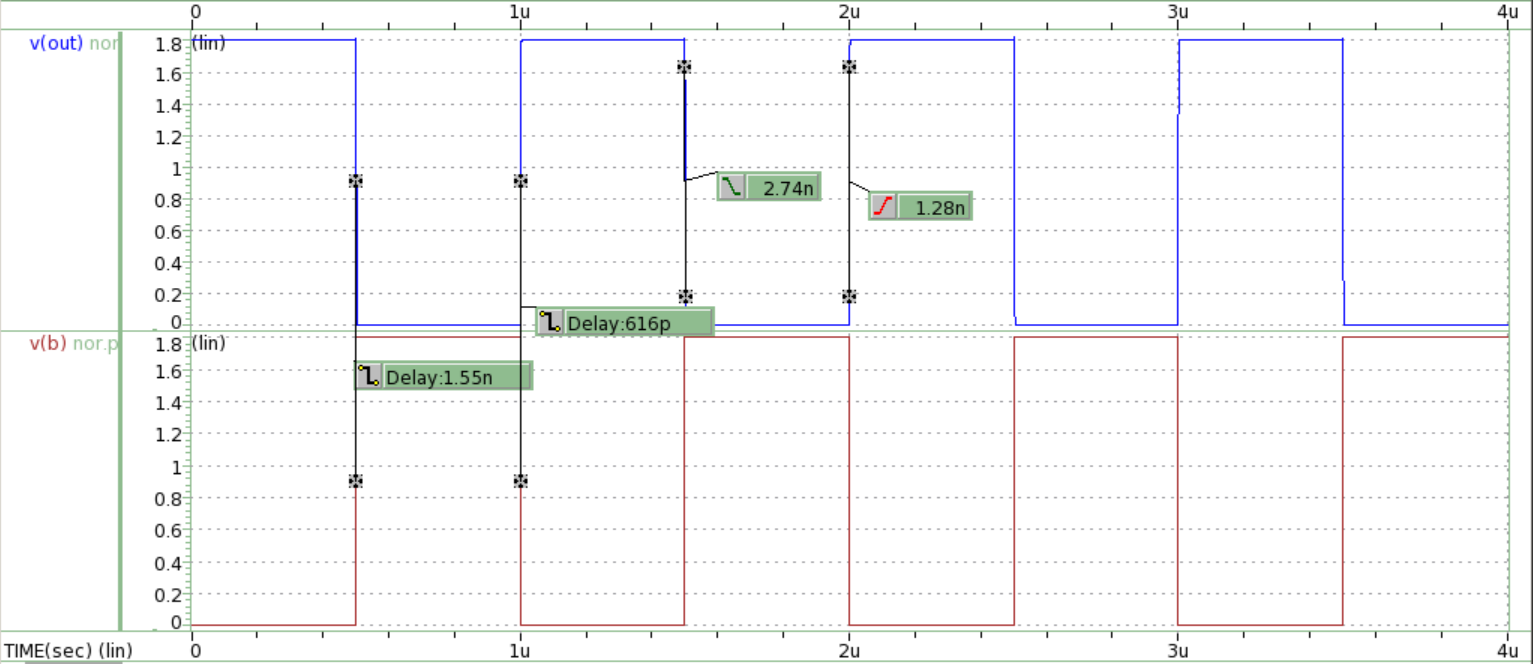
TT



SS

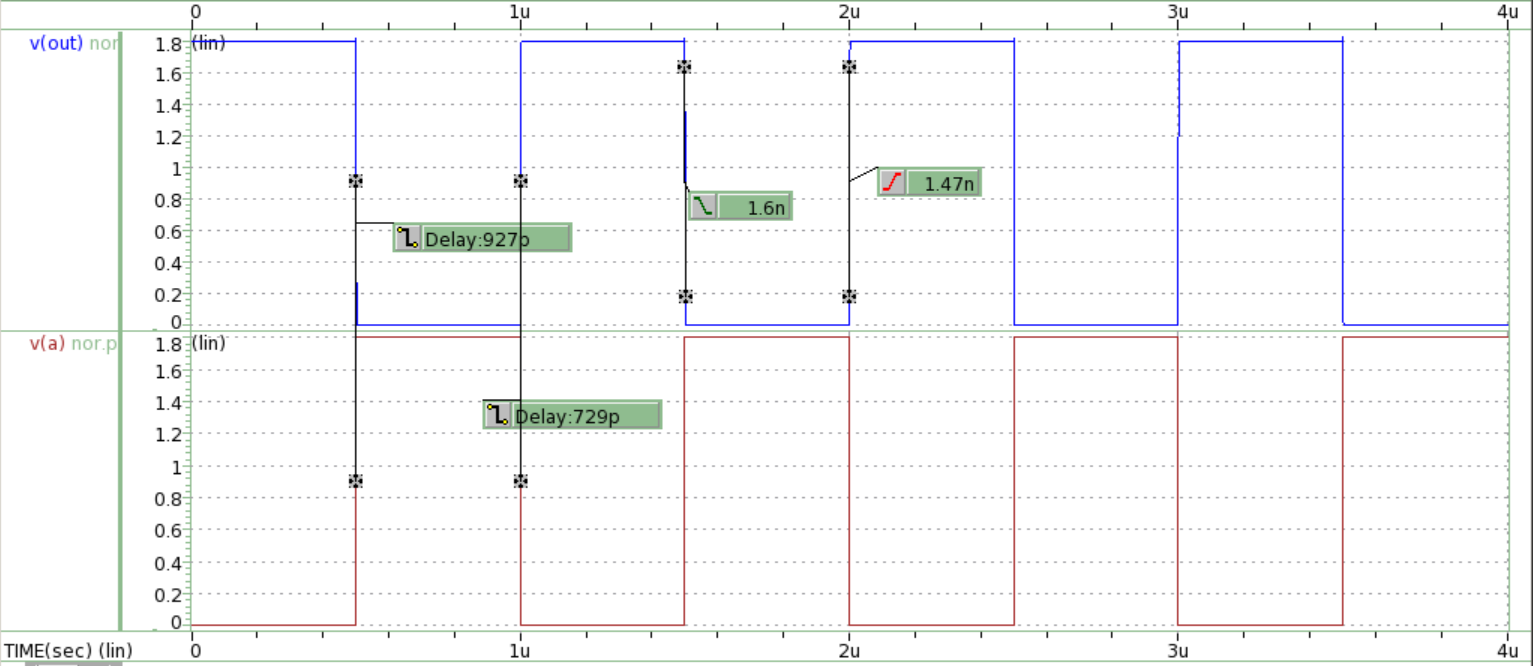


FF

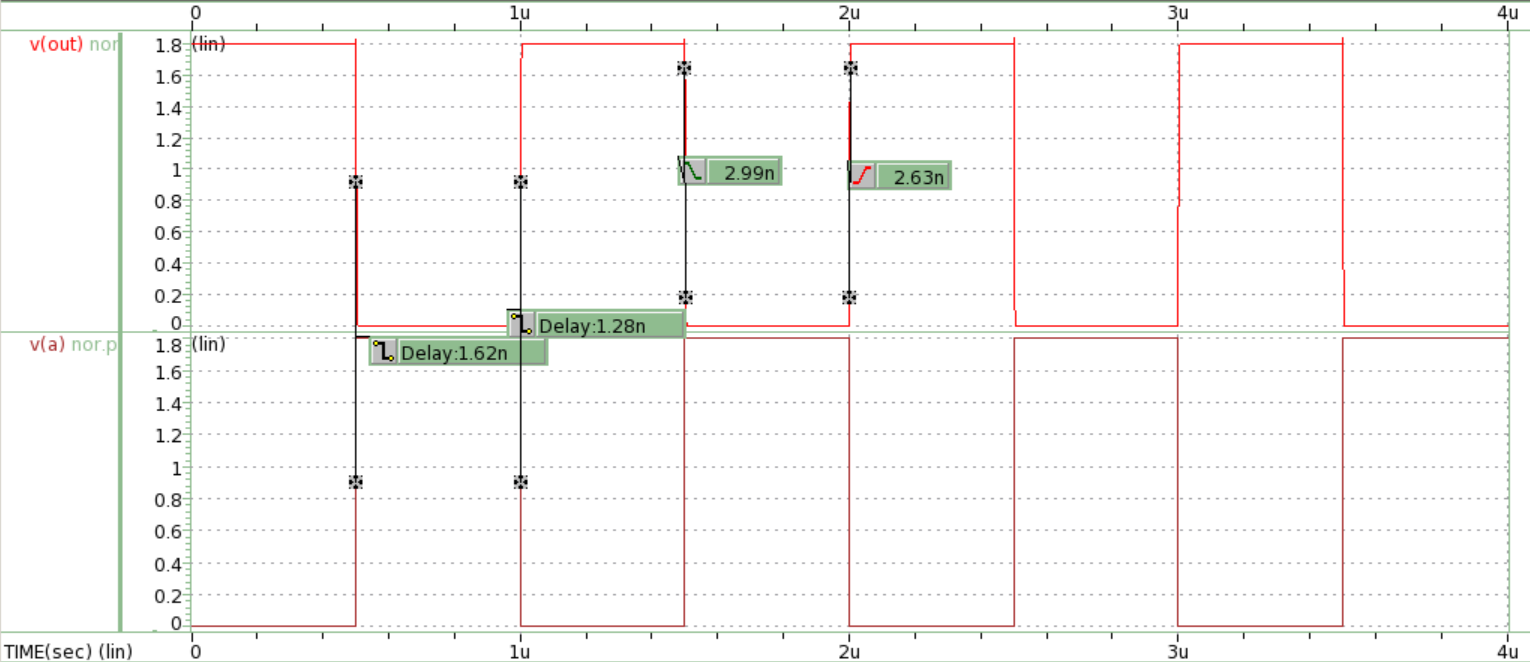


Case 3

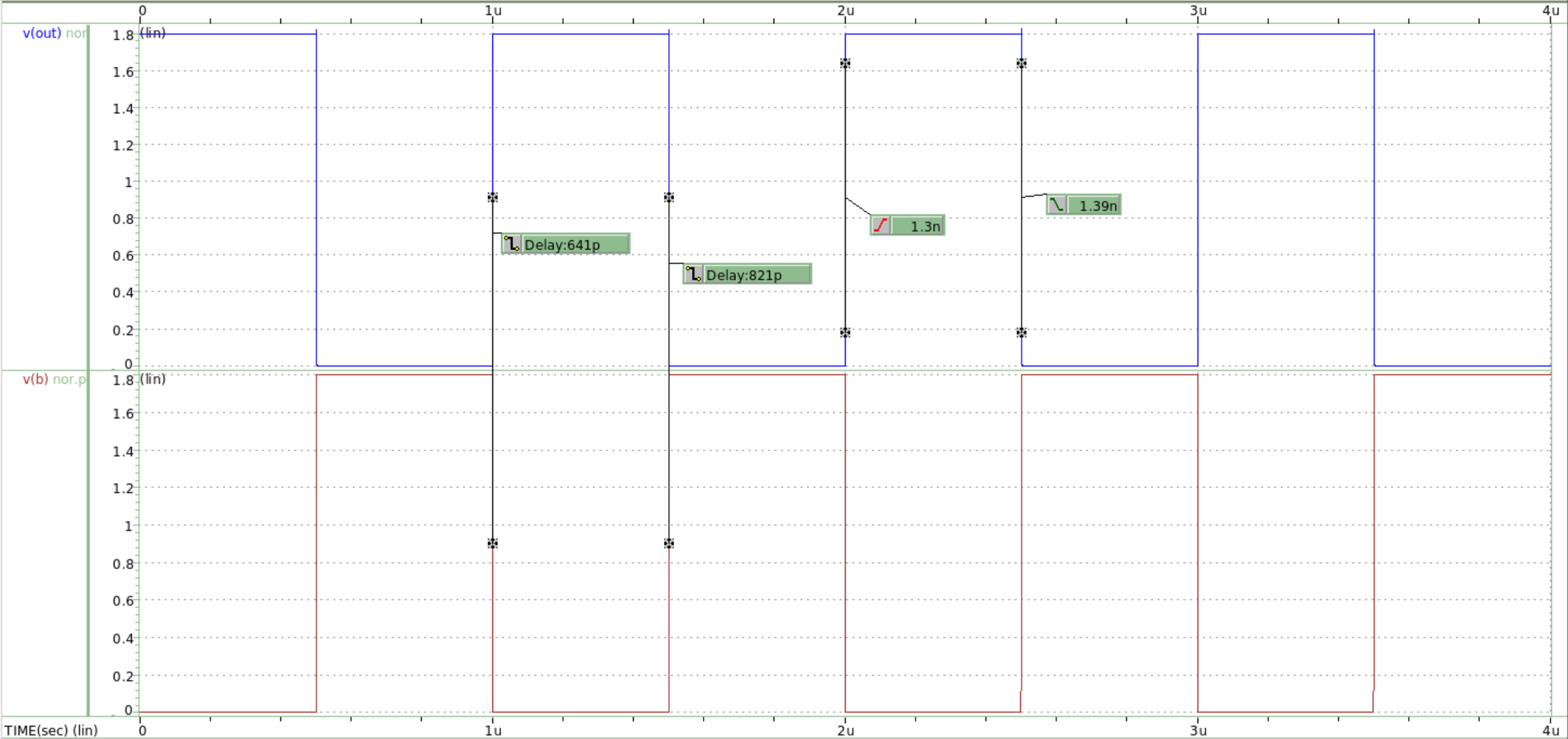
TT



SS



FF



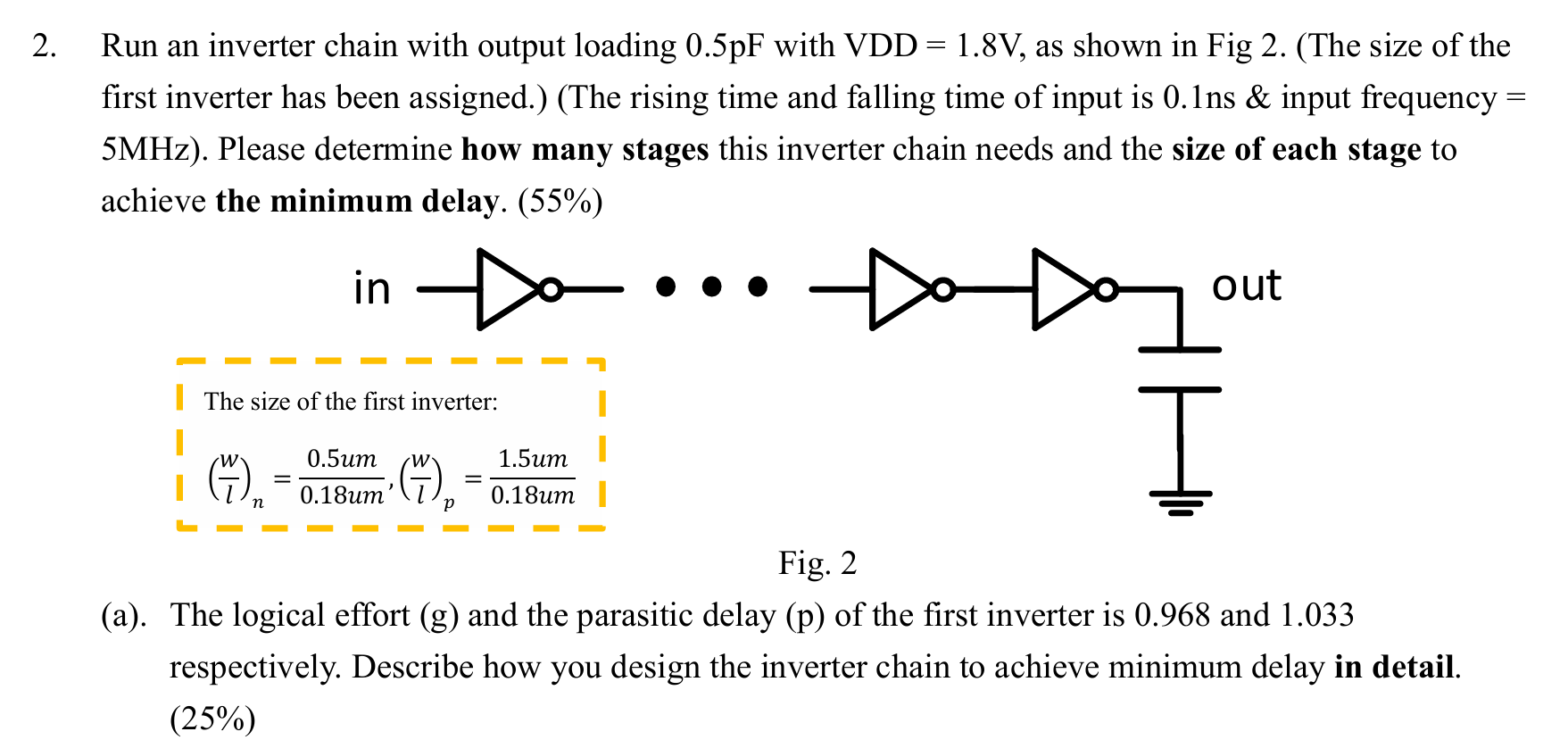
Post-sim測量的結果(用waveview測量)：

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | Case 1 | | | Case 2 | | | Case 3 | | |
| A | CLK | | | 0V | | | CLK | | |
| B | 0V | | | CLK | | | CLK | | |
| Corner | TT | SS | FF | TT | SS | FF | TT | SS | FF |
|  | 1.85n | 3.3n | 1.62n | 1.77n | 3.13n | 1.55n | 927p | 1.62n | 821p |
|  | 737p | 1.29n | 647p | 703p | 1.24n | 616p | 729p | 1.28n | 641p |
|  | 1.43n | 2.53n | 1.26n | 1.46n | 2.59n | 1.28n | 1.47n | 2.63n | 1.3n |
|  | 3.22n | 6.03n | 2.81n | 3.15n | 5.9n | 2.74n | 1.6n | 2.99n | 1.39n |

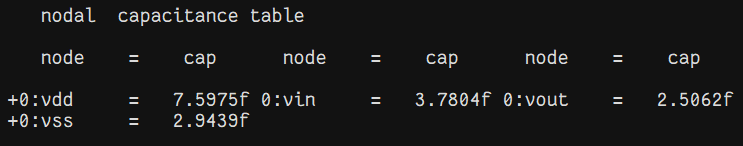
Pre-sim測量的結果(用waveview測量)：

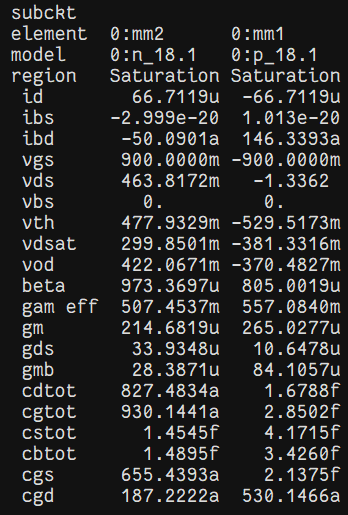
|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | Case 1 | | | Case 2 | | | Case 3 | | |
| A | CLK | | | 0V | | | CLK | | |
| B | 0V | | | CLK | | | CLK | | |
| Corner | TT | SS | FF | TT | SS | FF | TT | SS | FF |
|  | 1.75n | 3.17n | 1.52n | 1.66n | 2.98n | 1.45n | 848p | 1.52n | 743p |
|  | 639p | 1.17n | 550p | 627p | 1.15n | 542p | 642p | 1.17n | 556p |
|  | 1.32n | 2.39n | 1.15n | 1.35n | 2.46n | 1.17n | 1.36n | 2.49n | 1.18n |
|  | 3.1n | 5.87n | 2.7n | 3.03n | 5.72n | 2.63n | 1.52n | 2.88n | 1.32n |

可以清楚看到post-sim的delay和rise/fall time都是略大於pre-sim的結果，因為post-sim多考慮了實際layout後多出的寄生電容。



想要得到最小的delay可以先藉由上課所教的手算delay的方法去評估。首先我們先設定H的大小，因此我們需要估計。根據題目給定的第一個inverter的尺寸，建立一個sp檔，並用.op和.option captable量測當時，各節點的電容值，結果如下：





藉由時的模擬結果得出以及題目給定，可以進一步得出

另外題目也提供和，假設每個inverter的g, p相同，可以由此設和，inverter chain沒有branch所以，因此

假設是F的N次方根(為使最小)，，代入後可得

想找到D的極值，須先找到的解，因此用python解出，code與結果如下圖：



N必為整數，因此取N=4。為了確認此數值解的正確，再將N的數值從1開始直接依序帶入看看D的極小值出現在哪，結果如下表：

|  |  |
| --- | --- |
| N | D |
| 1 | 129.0617 |
| 2 | 24.33093 |
| 3 | 17.89488 |
| 4 | 17.26287 |
| 5 | 18.02177 |
| 6 | 19.30788 |
| 7 | 20.84655 |

可以看到N=4真的是極小值出現的地方，因此取N=4。此時

此值接近上課所推導的數值(ch4-1 p.52-55)。接著推導出

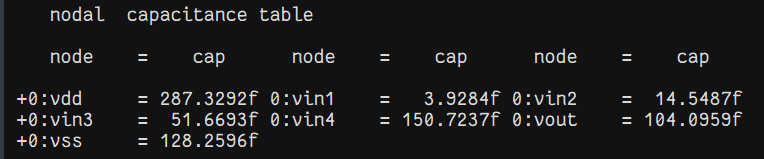
並以此求每一階的，從最後一階求到第一顆驗證答案，過程如下：

驗證與最初設置的相同，因此過程無誤，假設C正比於W/L，則

又因下級的輸入電容與上級的輸入電容之比值即為，因此可以推算出每個inverter的尺寸是以3.391倍為公比等比增加，NMOS尺寸依序為：

PMOS尺寸依序為：

依據上面計算好的各元件尺寸寫進sp檔並用hspice模擬，先用.option captable量測當時，各節點的電容值，為了確認是否與當初只有一顆時的狀況一樣，結果如下：



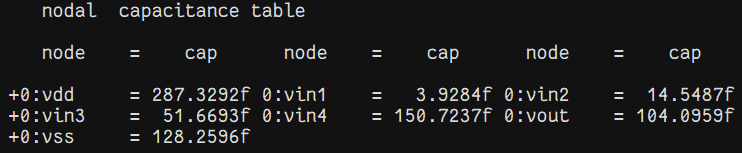
可以看到此時，與當初所設的有些微不同，因此再次代入並將估計delay的過程再做一次，但這一次我們直接代入N的數字，以便直接觀察D的極小值出現在哪，過程中的各項參數結果如下表：

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| N |  |  | H | G | B | F | fi | P | DF | D |
| 2 | 3.93fF | 0.5pF | 127.3 | 0.94 | 1 | 119.26 | 10.92 | 2.066 | 21.84 | 23.91 |
| 3 | 0.91 | 115.45 | 4.87 | 3.099 | 14.61 | 17.71 |
| 4 | 0.88 | 111.75 | 3.25 | 4.132 | 13.01 | 17.14 |
| 5 | 0.85 | 108.18 | 2.55 | 5.165 | 12.76 | 17.92 |

可以發現換成後，D的極小值仍出現在N=4的地方，因此估算其，得出各階W如下表：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | 第一顆inverter | 第二顆inverter | 第三顆inverter | 第四顆inverter |
|  | 0.5 | 1.68 | 5.64 | 18.95 |
|  | 1.5 | 5.04 | 16.92 | 56.84 |

以上表之尺寸去跑hspice後，也一樣先檢查的值，以確認的收斂，一樣是設成0.9V，其結果如下：



可以確認的值與開始估算D時所用的相同，因此可以開始來算delay time 實際上在hspice的模擬結果中有沒有最小，雖然估算結果告訴我們N=4時，delay會最小，但這只是手算的粗略估計，有很多假設及忽略在其中，因此實際跑hspice時，可以將N=3, 4, 5都試一次，以下是分別計算出來的與W：

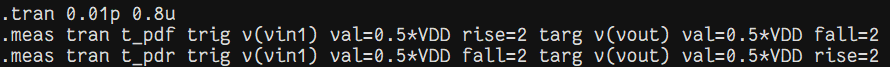
|  |  |  |  |
| --- | --- | --- | --- |
| N=3, |  |  |  |
|  | 0.5 | 2.52 | 12.65 |
|  | 1.5 | 7.55 | 37.95 |

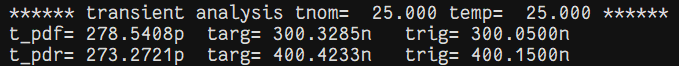
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| N=4, |  |  |  |  |
|  | 0.5 | 1.68 | 5.64 | 18.95 |
|  | 1.5 | 5.04 | 16.92 | 56.84 |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| N=5, |  |  |  |  |  |
|  | 0.5 | 1.32 | 3.47 | 9.16 | 24.14 |
|  | 1.5 | 3.95 | 10.42 | 27.48 | 72.43 |

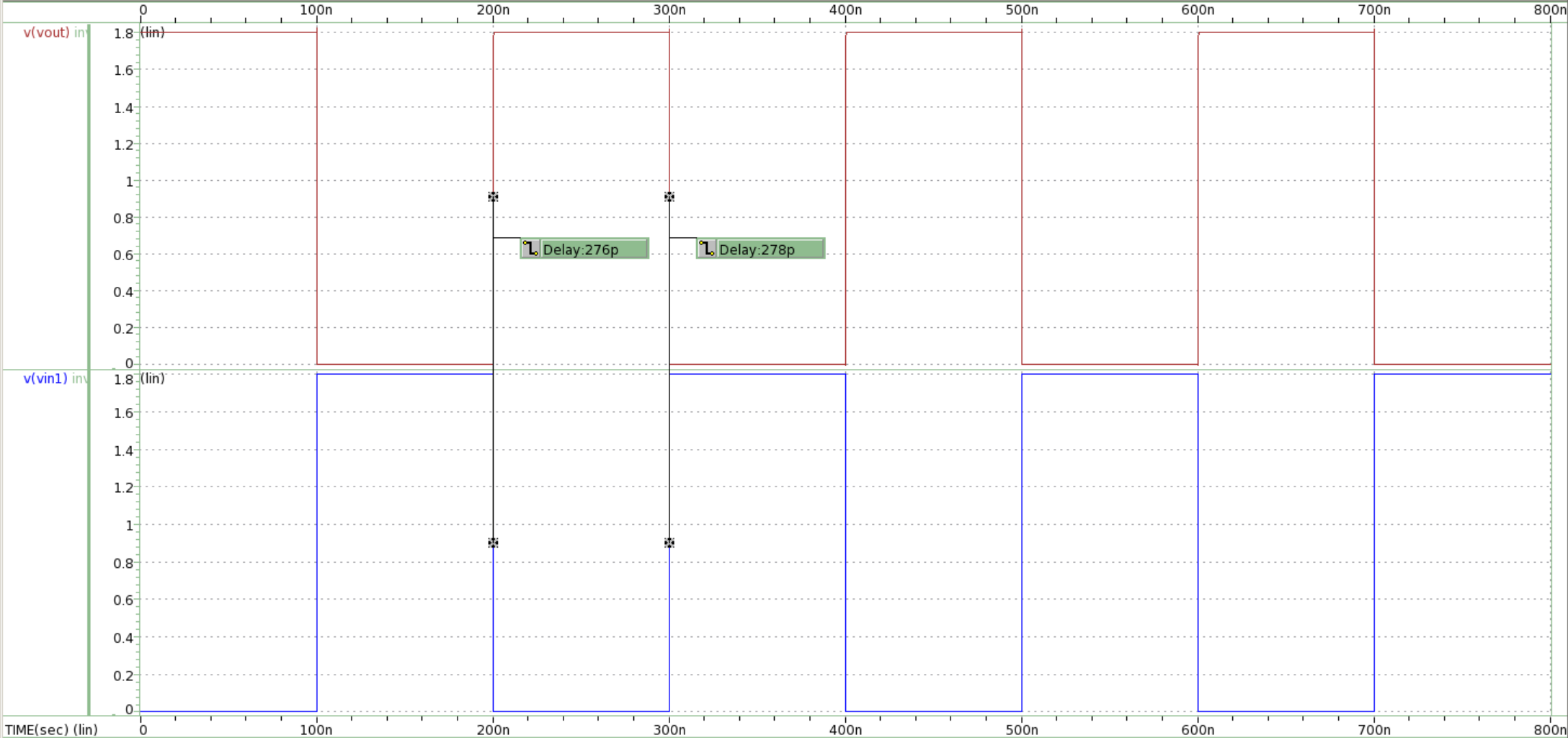
用.meas及waveview測量delay的結果如下()：

**N=3**



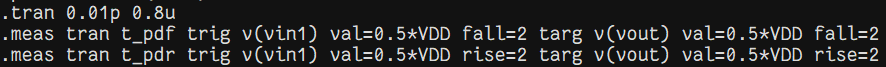


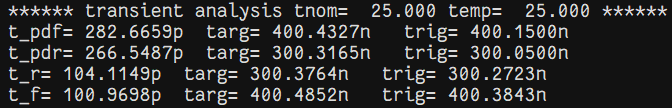
.meas量測的



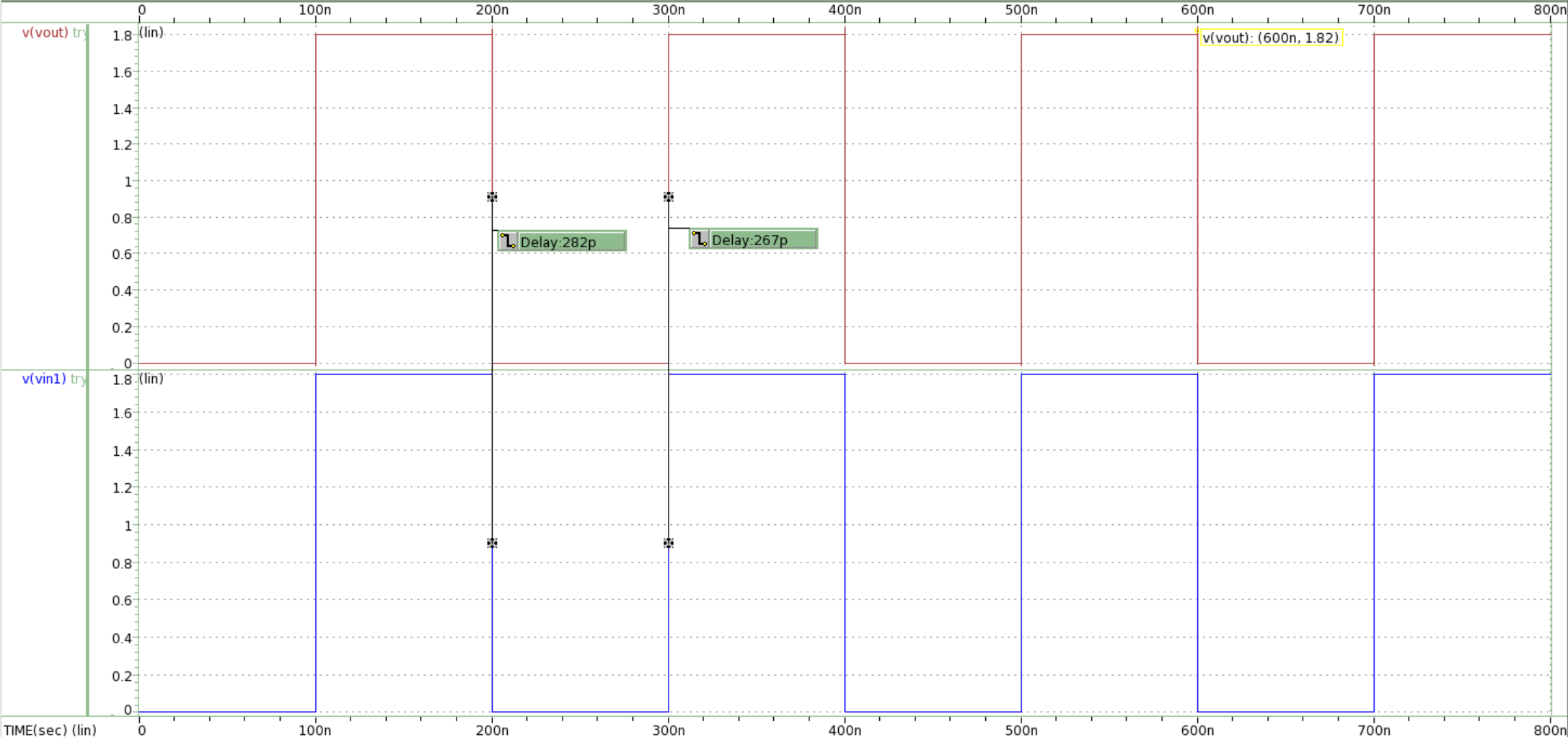
Waveview量測的

**N=4**



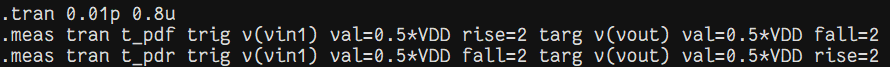


.meas量測的



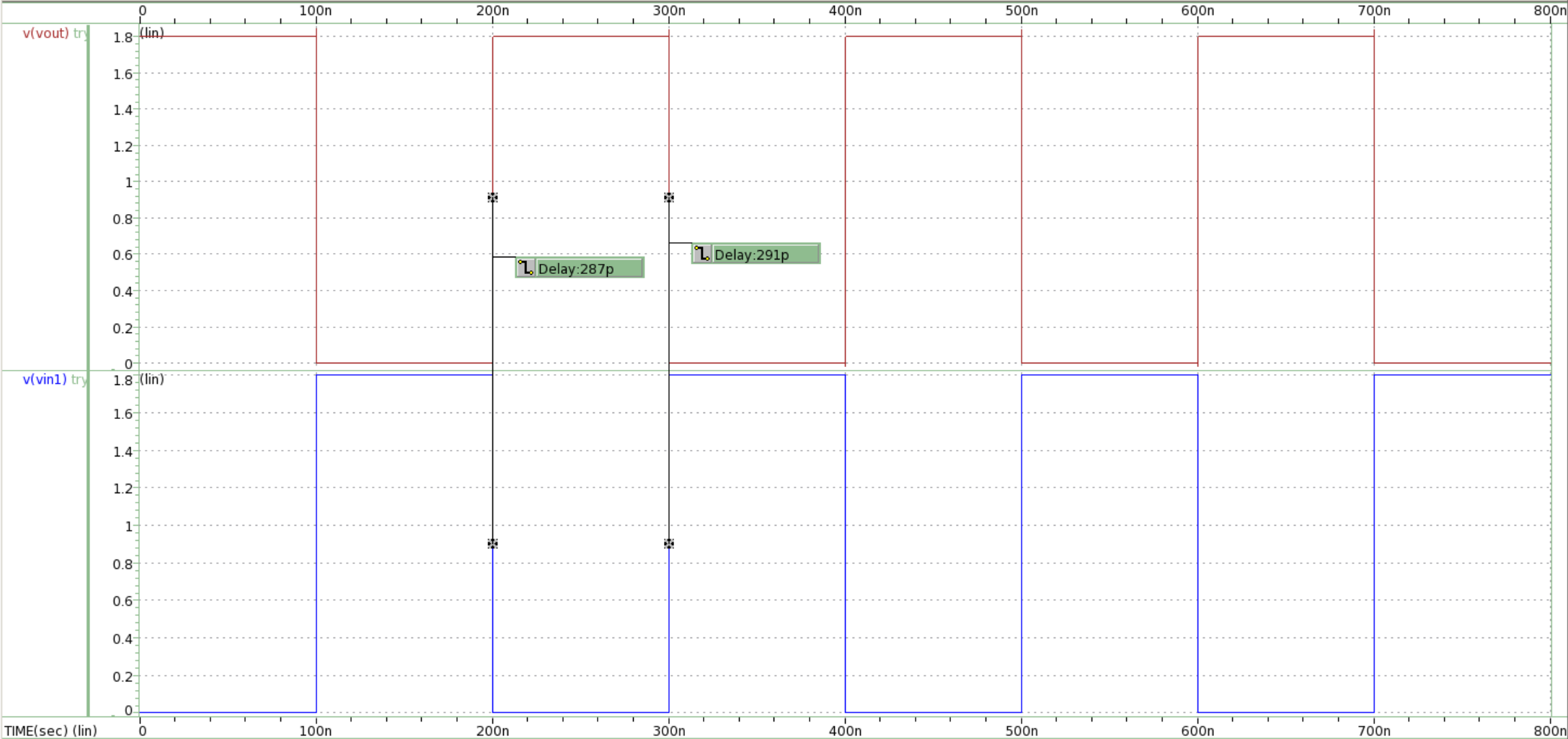
Waveview量測的

**N=5**



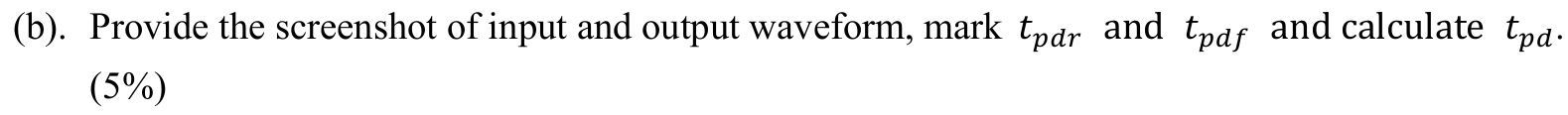


.meas量測的

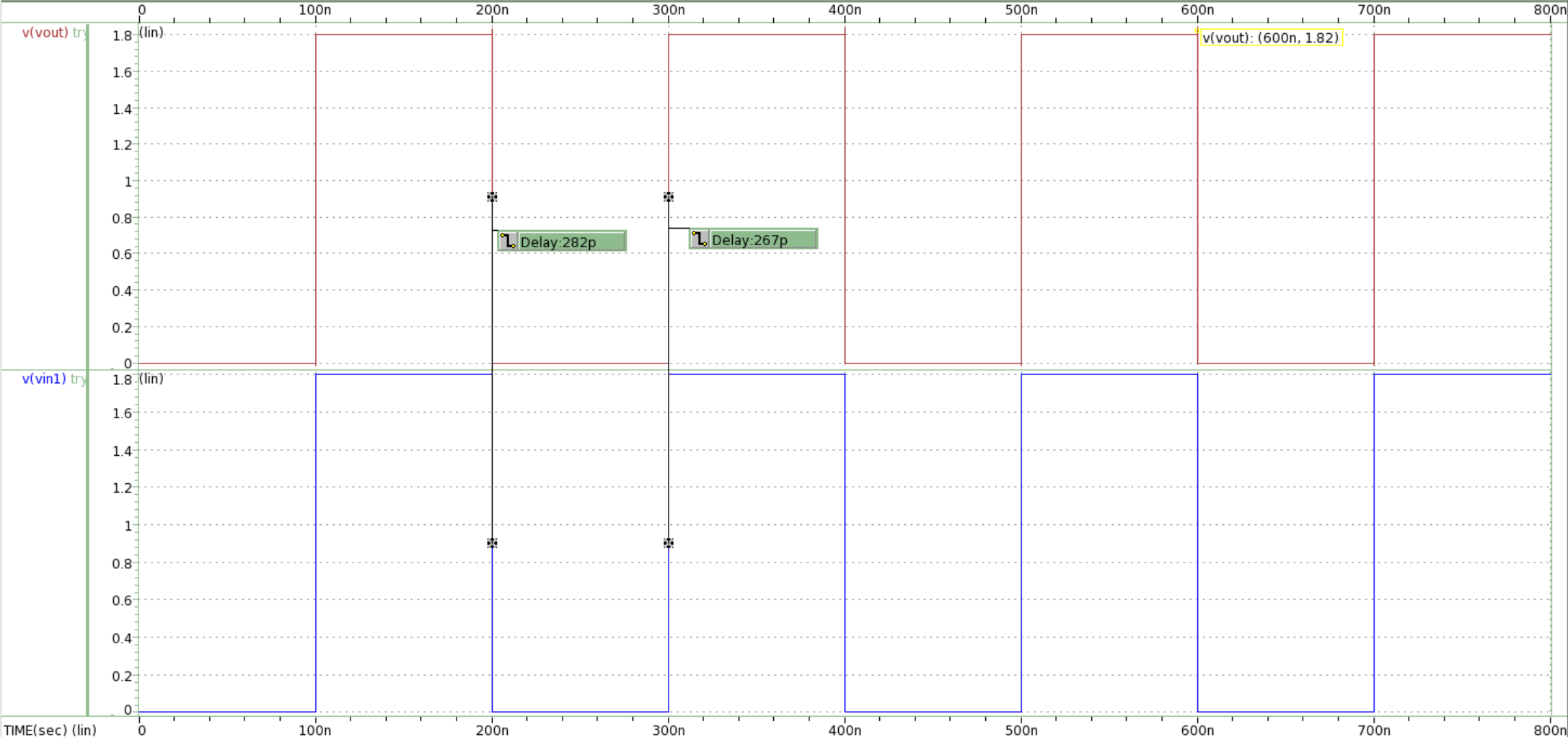


Waveview量測的

發現N=4的情況，測量出來的delay會最小，N=3的略大一點，而N=5的情況大的更多，與估算結果吻合，所以最終取用N=4時所推估的尺寸參數，以獲得最小的delay。備註：此時input/output為同向

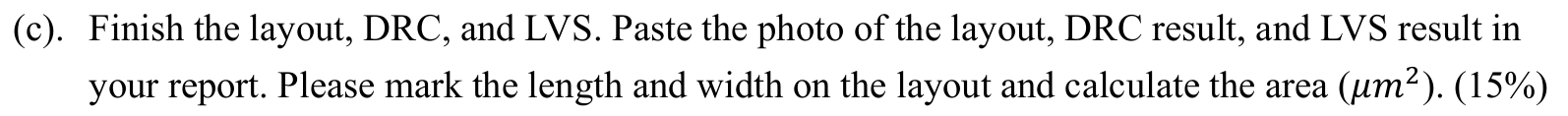


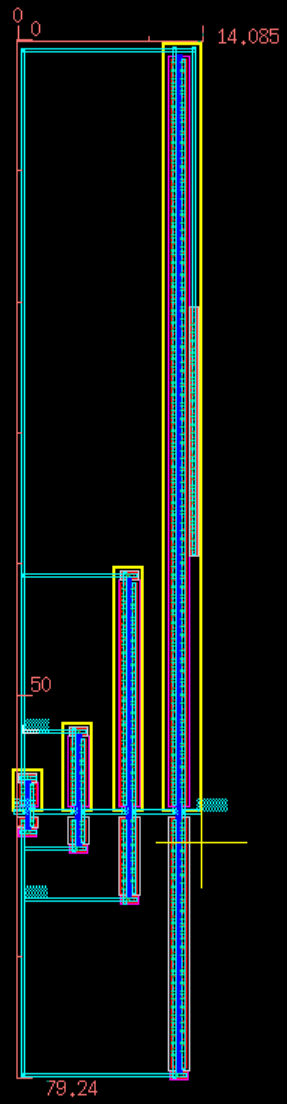
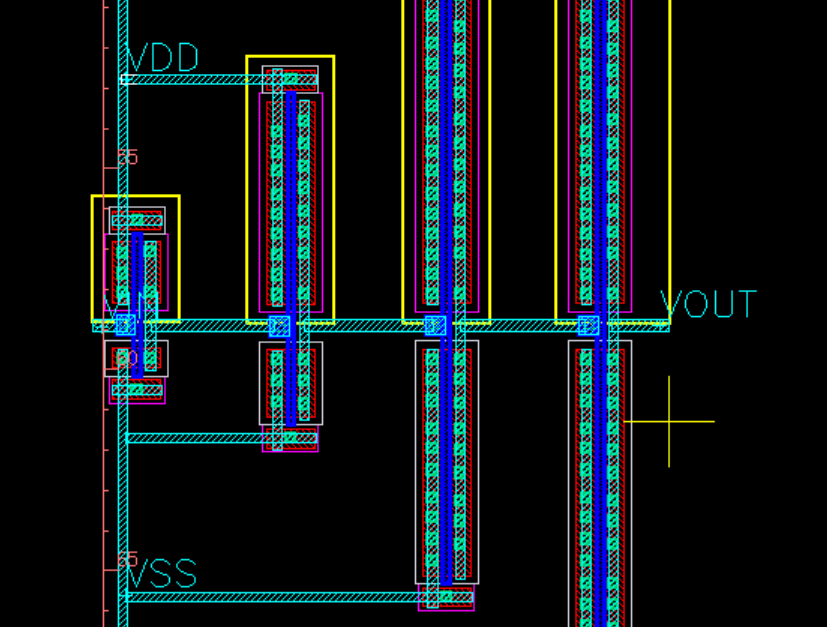
將上述尺寸交給hspice模擬，並用waveview測量結果如下：

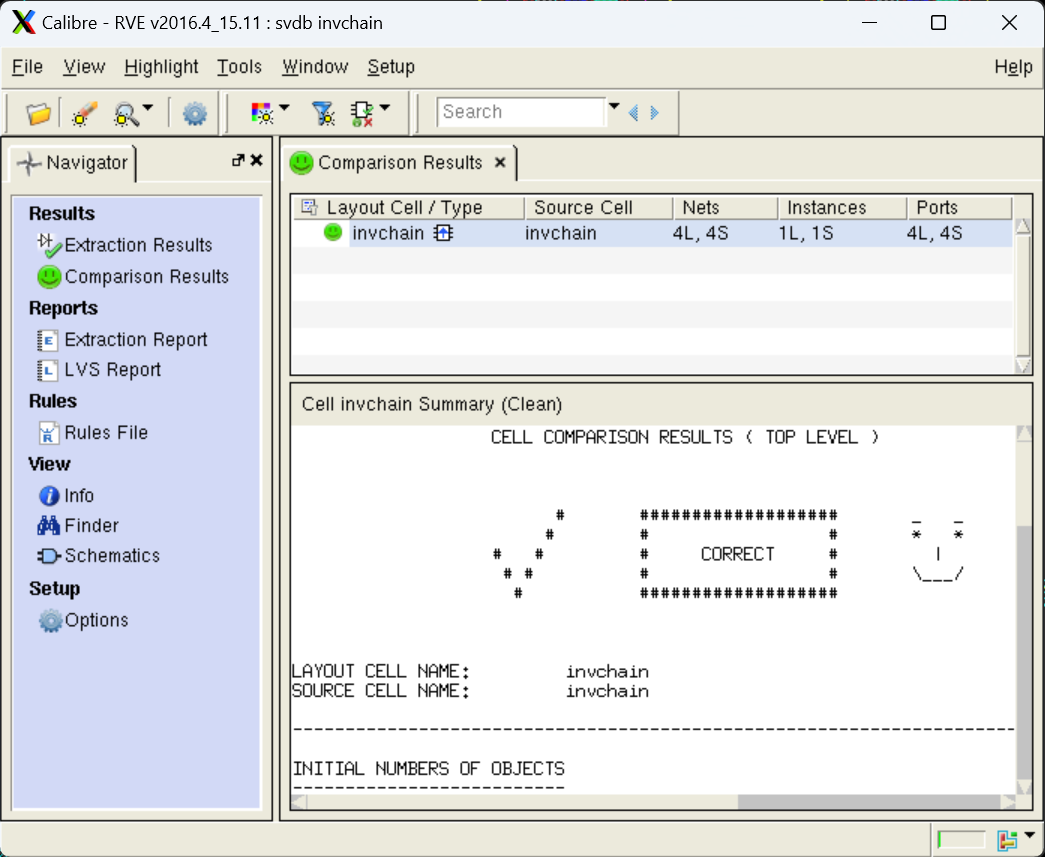
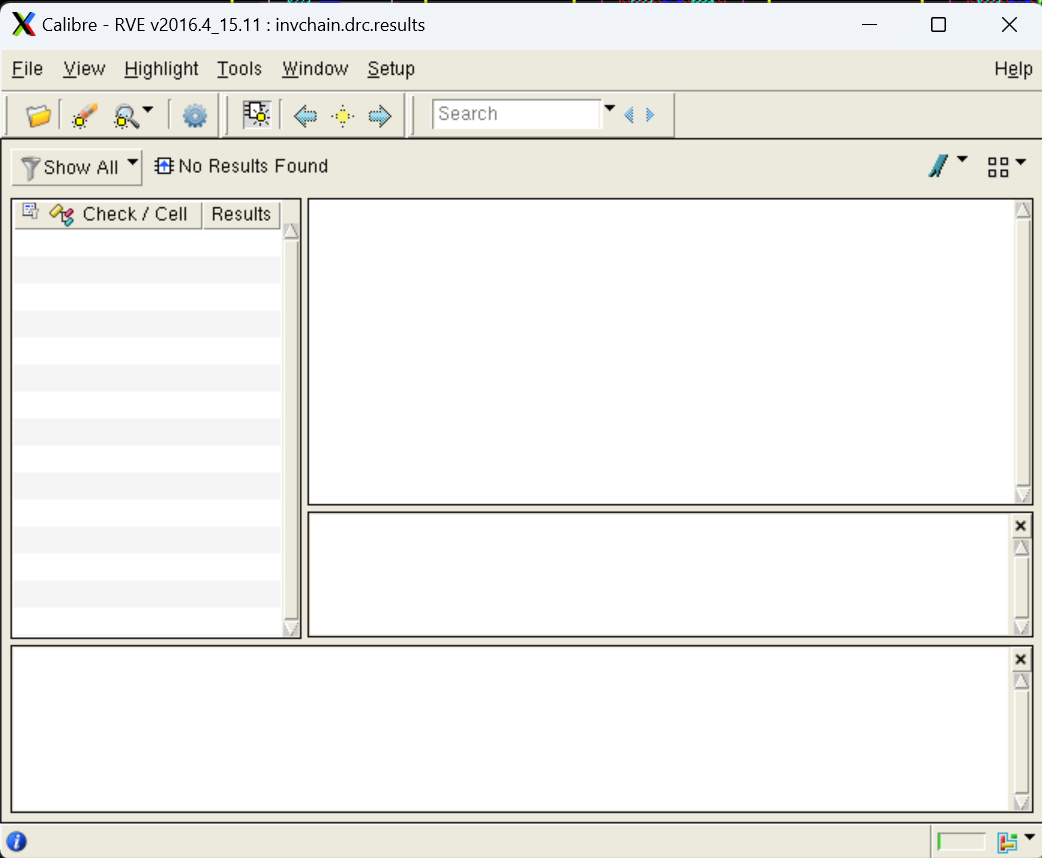


由圖可知並計算，結果如下：

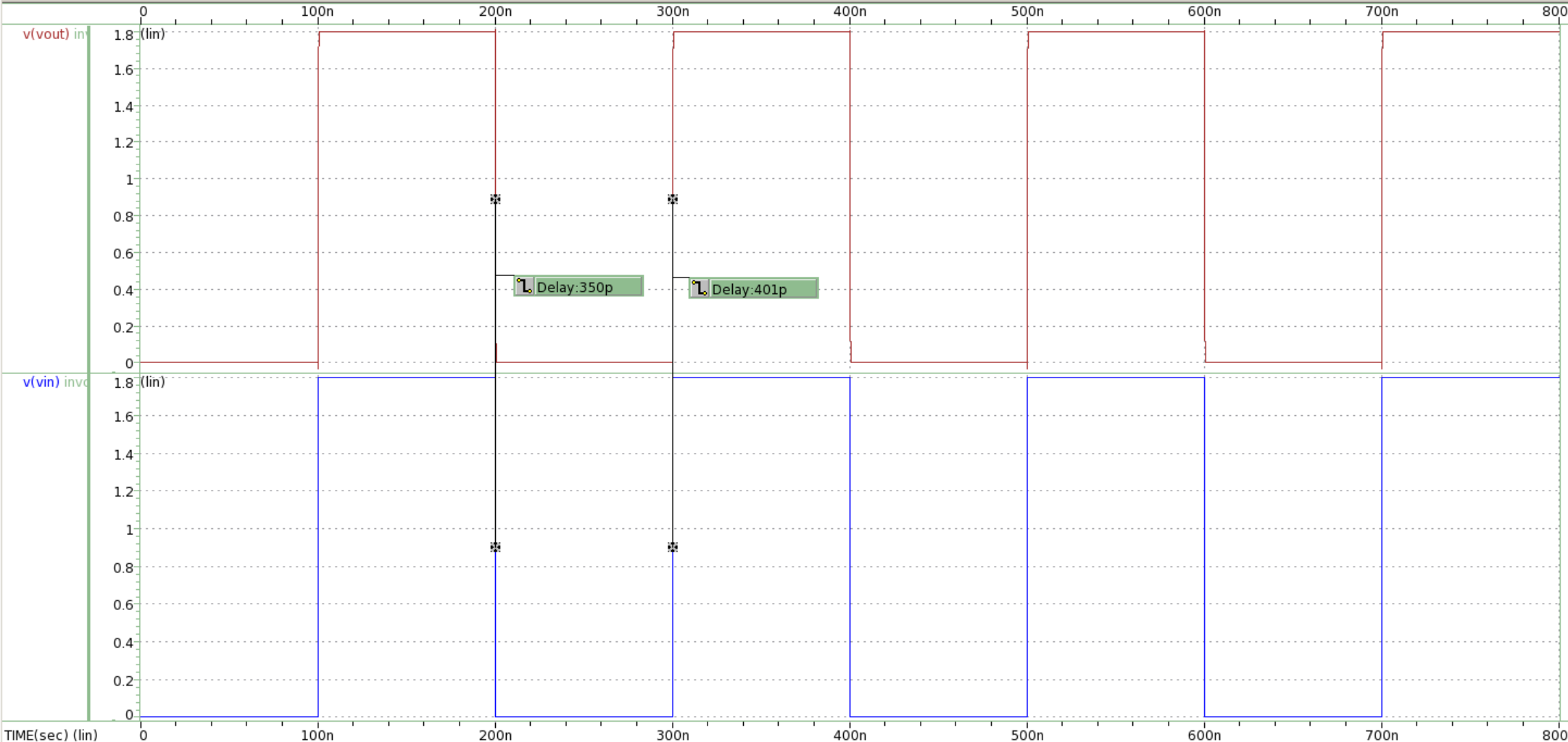
|  |  |  |
| --- | --- | --- |
|  |  |  |
| 276ps | 278ps | 277ps |



全圖：局部放大圖：







由圖可知並計算，結果如下：

|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  |  |
| Pre-sim | 276ps | 278ps | 277ps |
| Post-sim | 401ps | 350ps | 375.5ps |

可以清楚的看到post-sim的delay有明顯的比pre-sim的更大，這是合理的結果，因為post-sim考慮了實際layout所帶來額外的寄生電容。而會讓原本與幾乎相同甚至還略小的，在post-sim 後反而變得比還要更大很多的原因是因為在layout時，為了讓第四顆inverter的drain/source diffusion region與 body的diffusion region之最大距離不超過20以符合design rule，因此畫了一個很大的body給它。這個並排在drain/source diffusion region旁邊的大body額外加上了電容(因為間距小且面積大)，使得pull up 所需考慮的電容值上升，因此delay變得更大，而這是pull down那邊所沒有的。想要讓body回到正常大小的話，可以將第四顆inverter拆成四顆並聯，使其與第三顆的尺寸差不多。