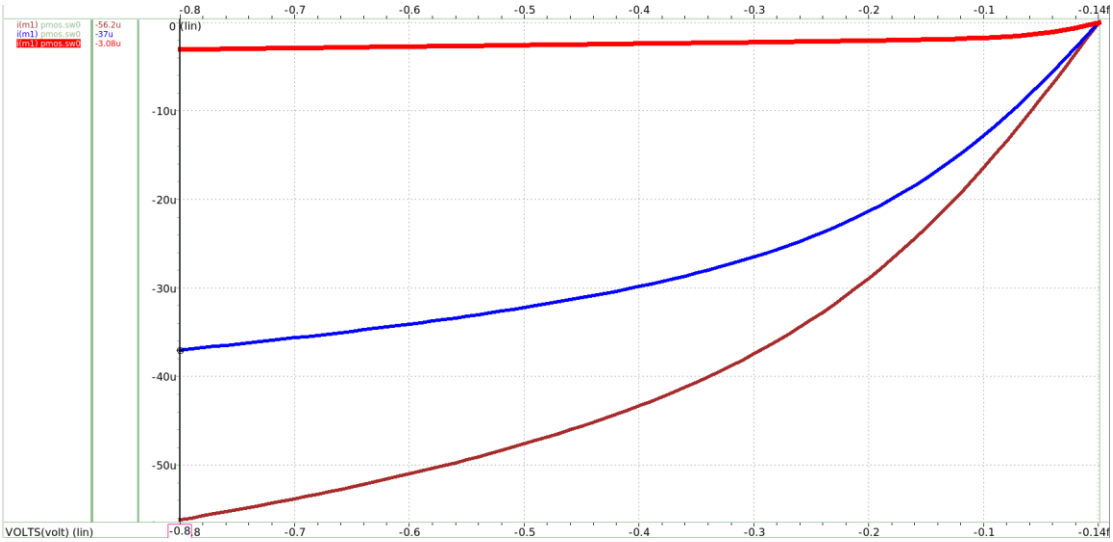
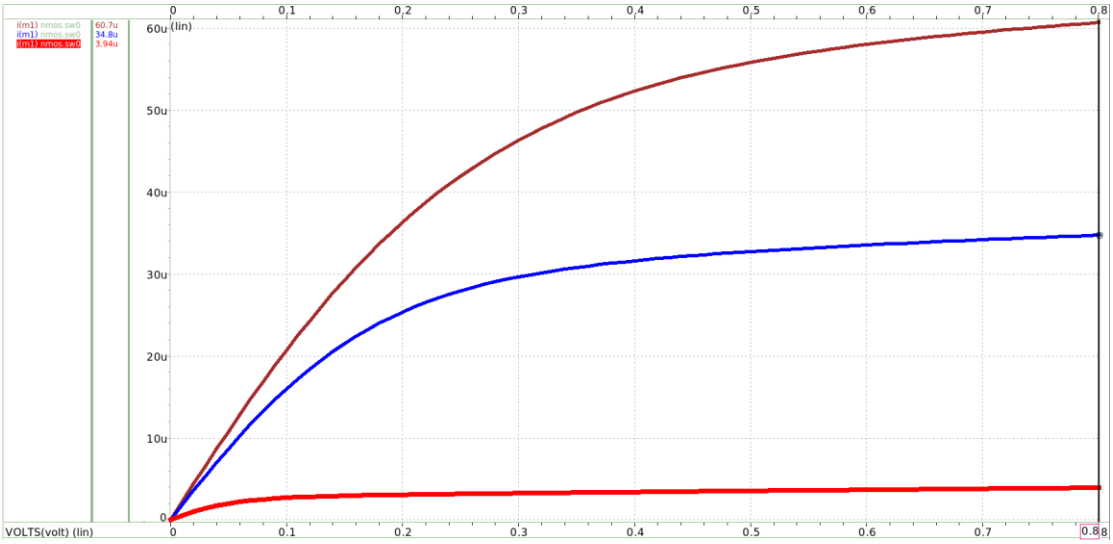
(1) MOS and Inverter (30%)

A. Run SPICE to draw I-V DC curve (like the figure below with V_{gs} = 0.8, 0.6, 0.35) for PMOS and NMOS with Fin n=1. Mark maximum I (V_{gs}=V_{ds}=VDD) in table form. Discuss the results.

(a) PMOS I-V curve:



(b) NMOS I-V curve:



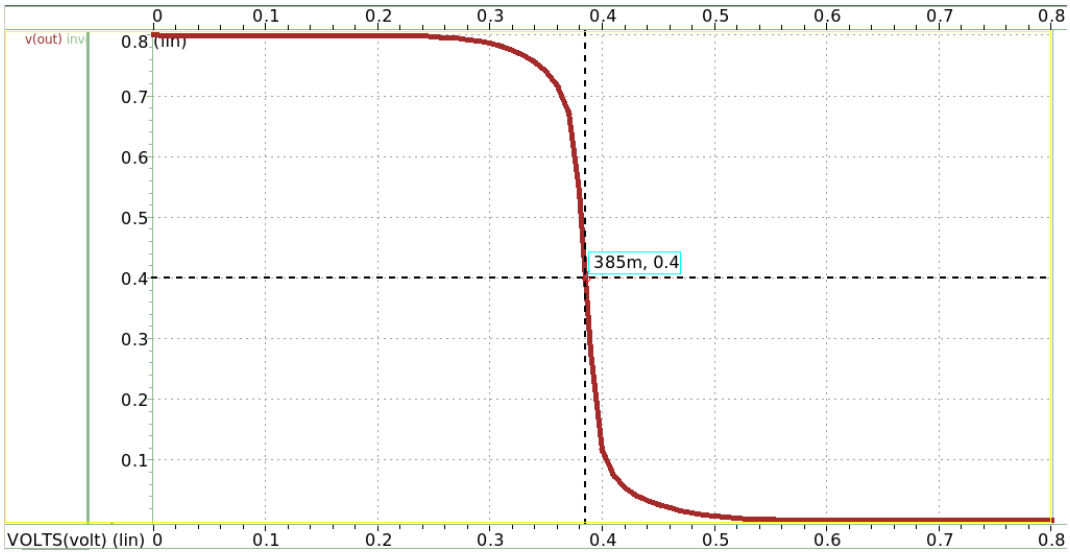
(c) Maximum I_{ds} in table form:

	I_{ds}		
	$\lvert V_{gs} \rvert$ = 0.8v	$\lvert V_{gs} \rvert$ = 0.6v	$\lvert V_{gs} \rvert$ = 0.35v
PMOS	-56.2uA	-37uA	-3.08uA
NMOS	60.7uA	34.8uA	3.94uA

(d) Discuss the result:
由上表可以發現 7nm FinFET 製程的 NMOS 與 PMOS 的電流大小非常接近，不像在 32nm 製程中 NMOS 的電流會是 PMOS 的兩倍之多。我們以往會習慣將 PMOS 大小調成 NMOS 的兩倍左右以平衡電流大小，得到一個 rise 和 fall 速度一致的元件，但在 7nm FinFET 製程中我們只需要把大小設的差不多。

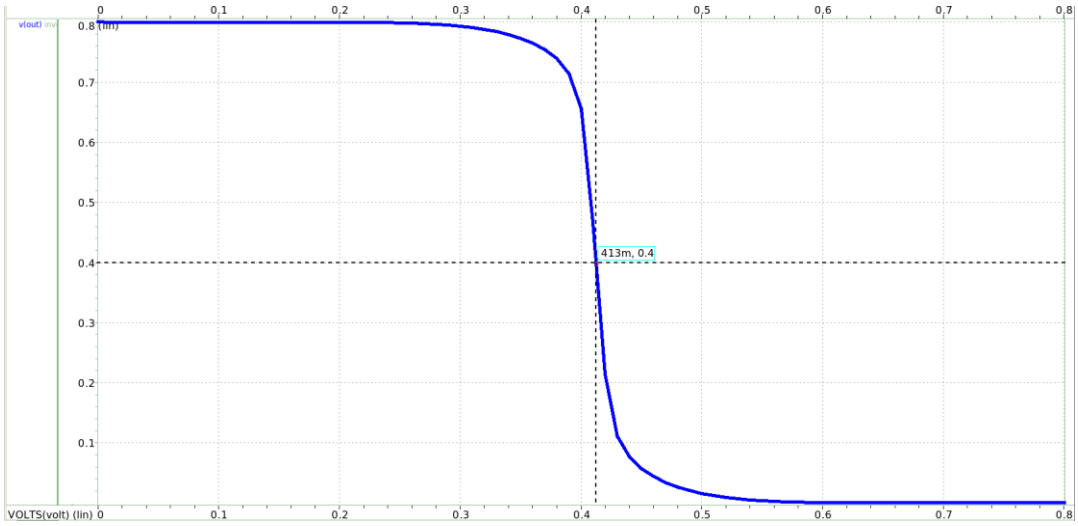
B. Keep a unit size inverter with NMOS n=1 and choose the n of PMOS for n=1 and 2 to show the logic threshold voltage. Run SPICE to verify your results by showing simulated waveforms.

(a) The waveform of n = 1:



According to the figure above, when input = 0.385V, the output = 0.5VDD = 0.4V.

(b) The waveform of n = 2:



According to the figure above, when input = 0.413V, the output = 0.5VDD = 0.4V.

(c) Discuss the result:

The following table shows the logic threshold voltage for n = 1 and n = 2.

	n = 1	n = 2
Logic threshold voltage	0.385V	0.413V

According to the table, 0.413V is closer to 0.4V. Therefore, I choose n = 2 as my unit size inverter.

(2) Ring Oscillator (20%)

According to the results of (1)B, design a 3-stage inverter-based ring oscillator with unit size inverter. Set the initial voltage of each node so that it can oscillate. Show the SPICE simulation results of oscillation frequency and power consumption in table form.

Here's my code of oscillator:

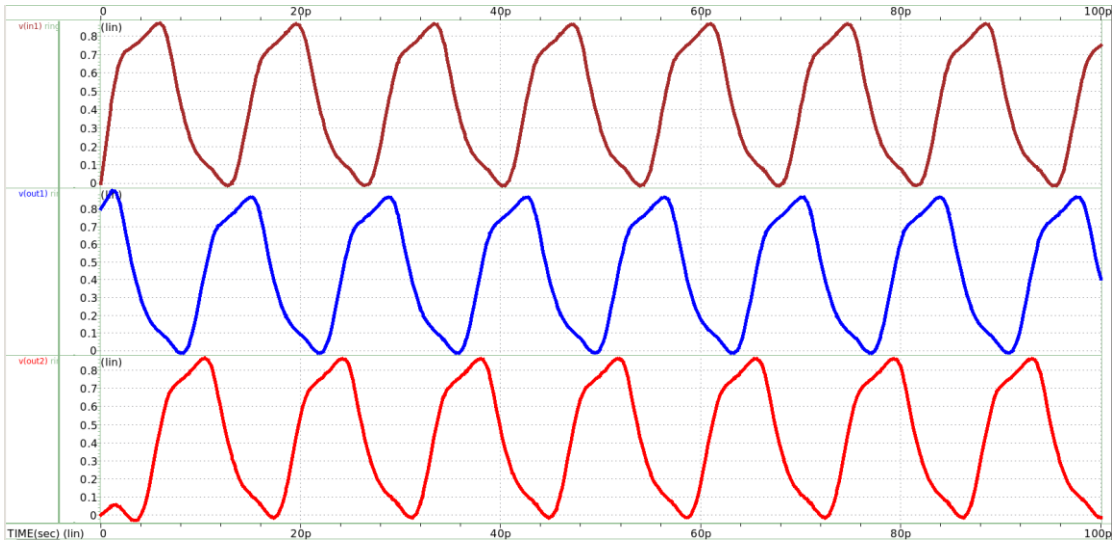
```
*********************************
**      Circuit description     **
*********************************
.subckt inverter A Y
MP Y A vdd! vdd! pmos_rvt nfin = 2
MN Y A gnd! gnd! nmos_rvt nfin = 1
.ends

xinv1  in1 out1 inverter
xinv2 out1 out2 inverter
xinv3 out2  in1 inverter
```

接著利用 .ic v(in1) = 0 (如下圖)，將 in1 的電壓初始化為 0，這樣輸入訊號就會通過 3 級 inverter 開始震盪。我個人是模擬 100 ps:

```
*********************************
**      Analysis setting       **
*********************************
.ic v(in1) = 0
.tran 1ps 100ps
```

The waveform of the three nodes, respectively:



(a) Show the SPICE simulation results of oscillation frequency:

Here's my code measuring the period of clock:

```
.meas tran period trig V(out1) val = 0.4 rise = 2 targ V(out1) val = 0.4 rise = 3
```

Here's the simulation result:

```
period=   13.7750p   targ=   37.8365p   trig=   24.0615p
```

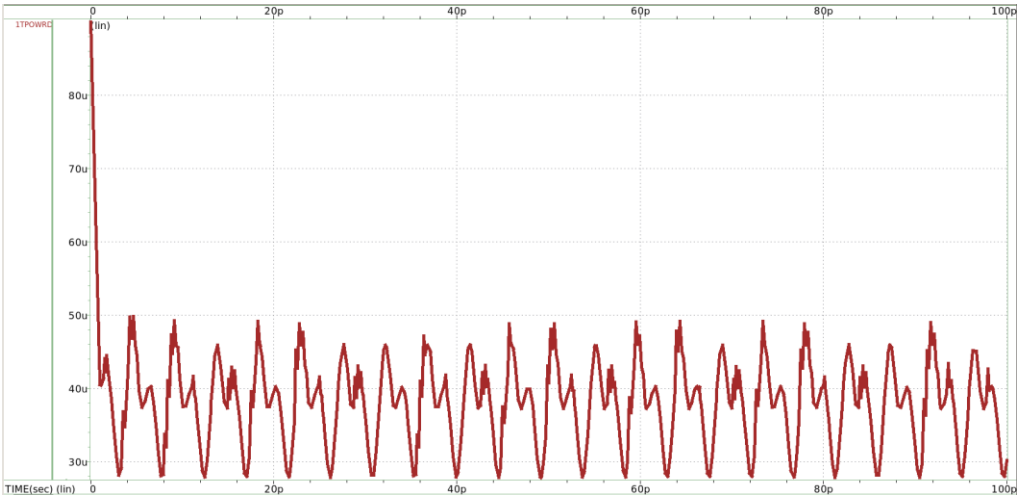
Therefore, the frequency is $\frac{1}{13.7750\times10^{-12}} = 72.5953$ GHz.

(b) Show the SPICE simulation results of power consumption

下圖是我量 power 的 code，我量了 average power 以及 peak power。此外，我看 power 的波型圖在最一開始衝得很高，我覺得那算離群值，所以我從 10ps 才開始取 data。

```
.tran 1ps 100ps
.probe power = par('-p(vvdd)')
.meas tran avgpwr AVG power from = 10p to = 90p
.meas tran peakpwr MAX power from = 10p to = 90p
```

Here's the power waveform:



Here's the simulation result of power:

```
avgpwr=   37.9873u   from=   10.0000p      to=   90.0000p
peakpwr=   49.1717u     at=   73.3988p
               from=   10.0000p     to=   90.0000p
```

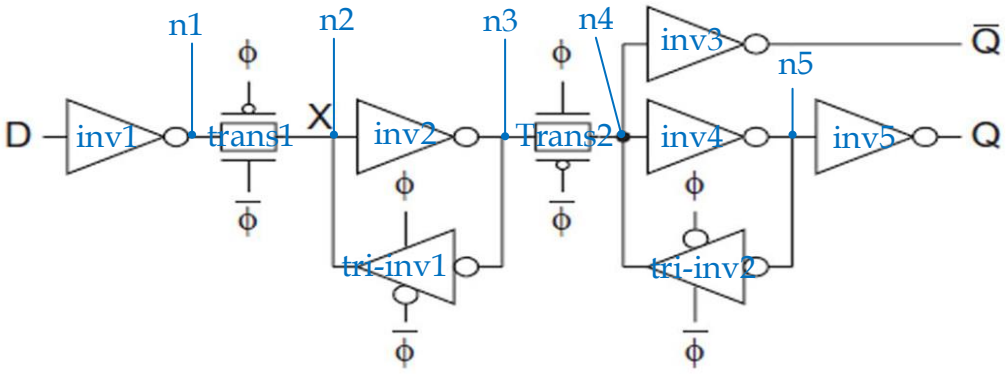
In conclusion, average power =37.9873 uW and peak power = 49.1717 uW.

(c) Table form:

Oscillation period	13.7750 ps
Oscillation frequency	72.5953 GHz
Average power	37.9873 uW
Peak power	49.1717 uW

(3) Assume both ϕ and $\bar{\phi}$ are available, design a static D register as shown in the figure below with proposed size of NMOS and PMOS to have better t_{setup}, t_{pcq}, t_{pdq}, and t_{hold}. The loading of Q and $\bar{Q}$ have 4 unit size inverters as loading. Both D and CLK has rise time and fall time of 0.1ns (0V-0.8V). (50%)

A. Explain your sizing principle of each MOS to have least t_{setup}, t_{pcq}, t_{pdq}, and t_{hold} for logic 1 and 0. (Explain that by changing which transistors would affect each timing respectively)



(a) t_{setup}:

Setup time 取決於前半部的 DFF，也就是 D-> inv1 -> trans1 -> inv2 -> tri-inv1，改變 n2 及 n3 並穩定所需的時間。因此:

- inv1 需要強一點讓 D 的值可以快速傳到下一級。
- inv2 及 tri-inv1 不用太強，因為這兩者的作用只是用來穩定 trans1 的 output 而已。
- trans1 要比 tri-inv1 強，否則會沒有能力把 n2 拉成新的值。

(b) t_{hold}:

當 clk 從 0 變 1 的時候，理想上 trans2 會開啟將 n3 的值傳到第二個 latch，而 trans1 會關閉以避免 D 的變化傳入 latch1。但實際上仍會有一小段時間 trans1 及 trans2 是同時開啟的，若此時 D 發生變化就會導致錯誤的值被一路傳到 Q 去，因此 D 必須在 posedge clk 之後一段時間保持穩定。因此 trans1 若關閉速度越快則所需的 hold time 越短。

(c) t_{pcq}:

clk 從 0 變 1，n3 經過 trans2 -> inv3 -> tri-inv2 -> inv3 -> inv4，得到穩定輸出 Q 所需的時間。因此:

- inv5 需要強一點，否則會推不動 Q 後面接的 loading。
- inv4 需要次強，用來推動 inv5 以及 tri-inv2。
- tri-inv2 弱一點，因為 tri-inv2 只是用來維持 trans2 的 output 而已。

無論是 logic 0 或 1，各個 MOS 的 size 對 timing 的影響都符合上述趨勢。除非有特定需求，比方說要把 Q 的 0 拉到 1，則會需要特別調大 PMOS。

(d) My MOS size:

Here's the table showing the size for each MOS.

	PMOS	NMOS
inv1	5	5
trans1	1	1
inv2	2	2

	PMOS1	PMOS2	NMOS1	NMOS2
tri-inv1	1	1	1	1

	PMOS	NMOS
trans2	1	1
inv3	3	3
inv4	2	2
inv5	3	3

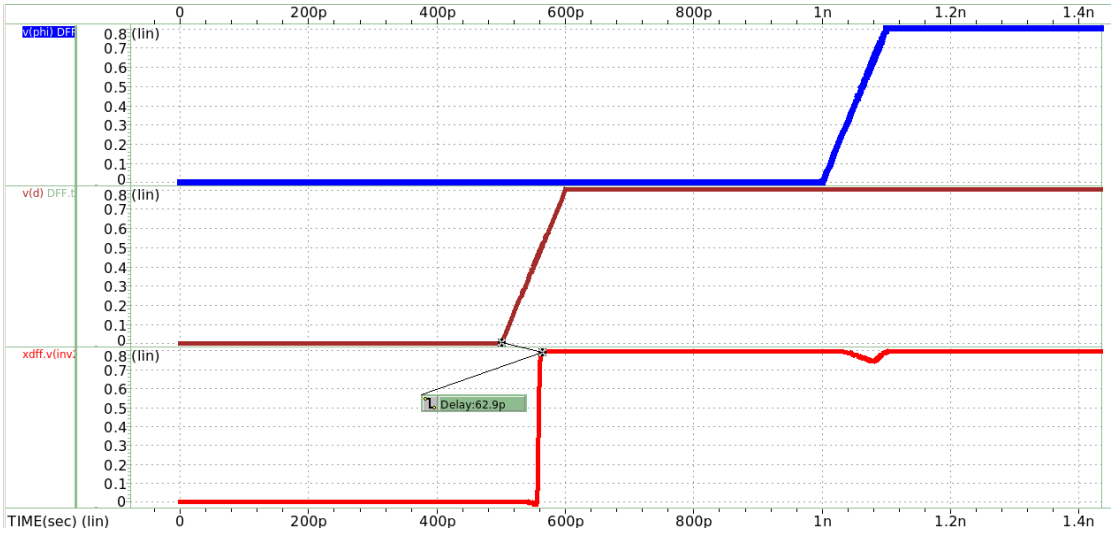
	PMOS1	PMOS2	NMOS1	NMOS2
tri-inv2	1	1	1	1

B. Run SPICE to verify your results and list the results (size of each transistor and four kinds of timing) in table form for part A.

(a) t_{setup} (logic 0): 當 D 的值改成 0 時，看 n3 的值會多快跟著改成 0。

t_{setup}=67.6ps

(b) t_{setup} (logic 1): 當 D 的值改成 1 時，看 n3 的值會多快跟著改成 1。

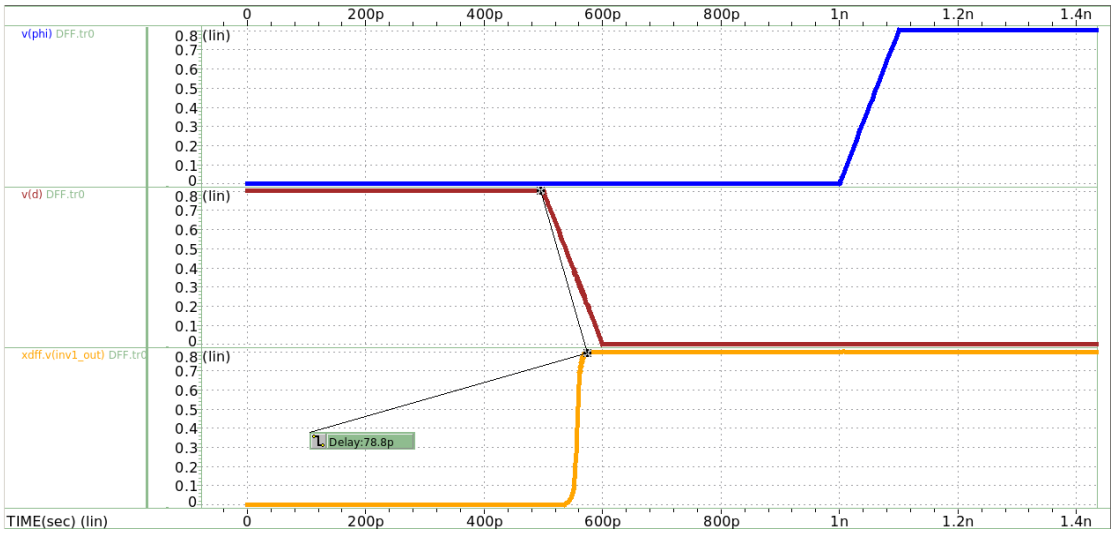


t_{setup}=62.9ps

(c) t_{hold} (logic 0): 下圖可以看出 D 經過 inv1 的 delay 有 62.9 ps。

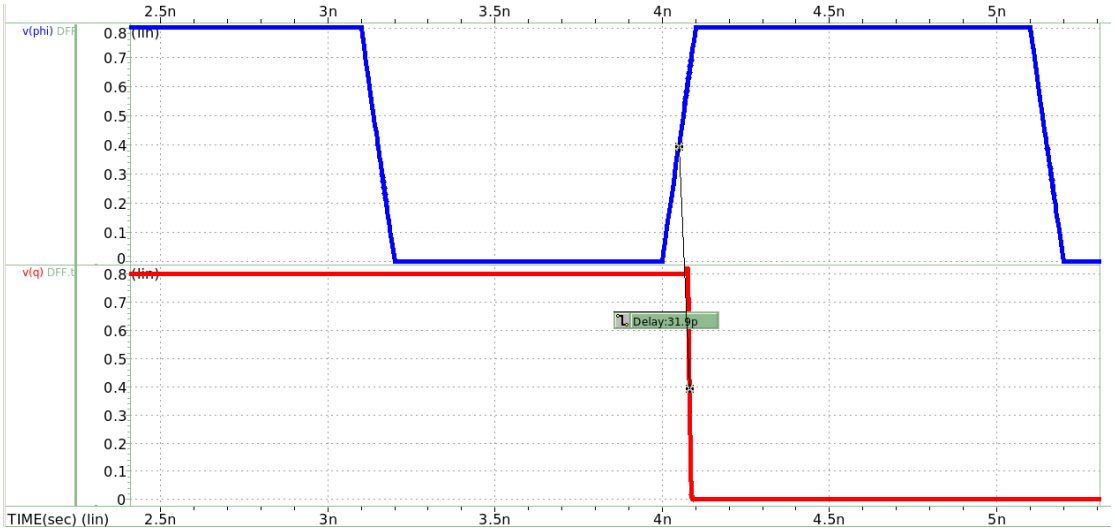
我在製程檔中看 nmos 的 Vt = 0.48 及 pmos 的 Vt = 0.4，也就是說當 phi = 0.48V 時 tran1 會關閉。又因為 rising time 是 0.1ns，所以 0.048 ns = 48 ps 時 trans1 會關閉。D 經過 inv1 的 delay 要 62.9ps，但 trans1 關閉只需 48ps。因此 D 其實在 posedge clk 後怎麼改值都沒關係，因為經過 62.9ps 的 delay 後 trans1 也早已關閉，因此 t_{hold} = 0ps。

(d) t_{hold} (logic 1): 下圖可以看出 D 經過 inv1 的 delay 有 62.9 ps。



同理(c)，因此 t_{hold} = 0ps。

(e) t_{pcq} (logic 0):



t_{pcq}=31.9ps

(f) t_{pcq} (logic 1):

t_{pcq}=36ps

(g) list the results in table form:

Here's the table showing the size for each MOS.

	PMOS	NMOS
inv1	5	5
trans1	1	1
inv2	2	2

	PMOS1	PMOS2	NMOS1	NMOS2
tri-inv1	1	1	1	1

	PMOS	NMOS
trans2	1	1

inv3	3	3
inv4	2	2
inv5	3	3

	PMOS1	PMOS2	NMOS1	NMOS2
tri-inv2	1	1	1	1

Here's the table of timing.

	t_{setup}	t_{hold}	t_{pcq}
Logic 0	67.6ps	0ps	31.9ps
Logic 1	62.9ps	0ps	36ps