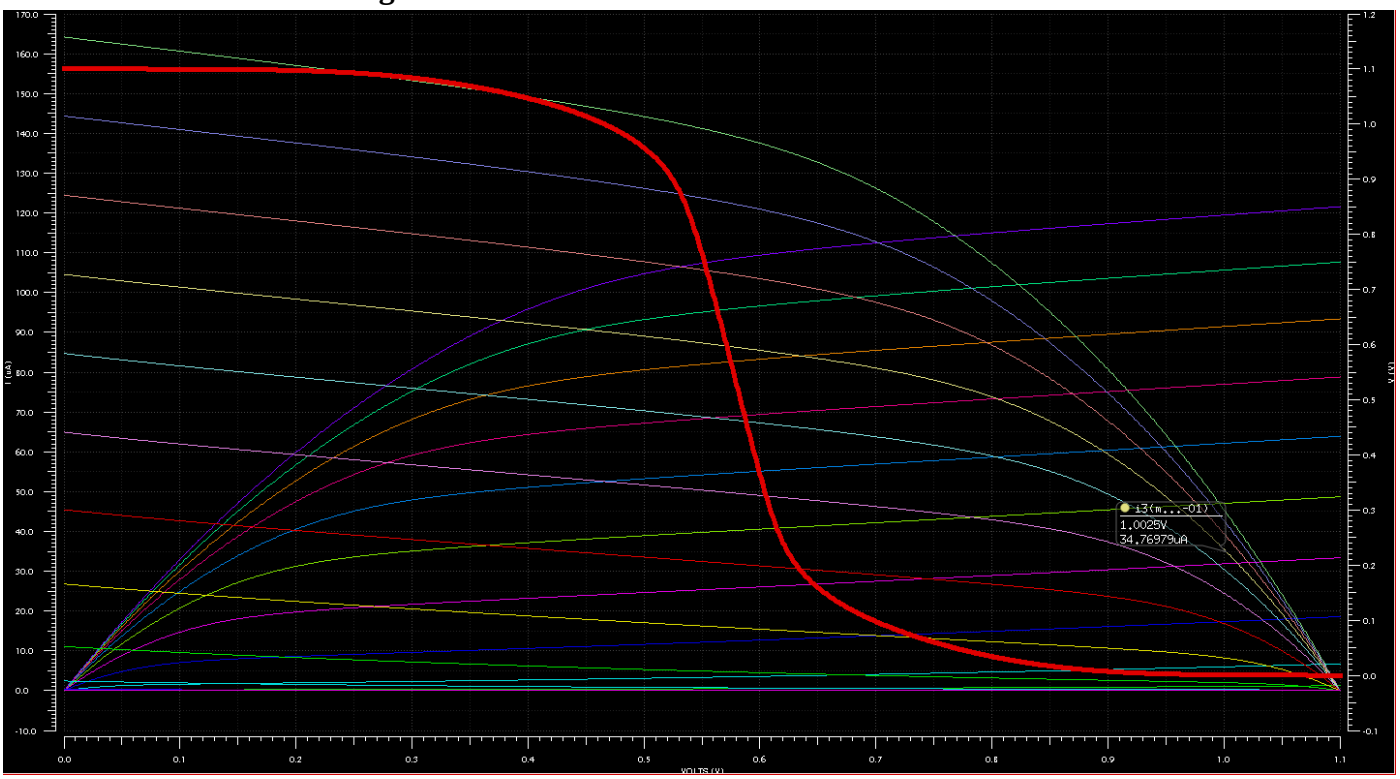
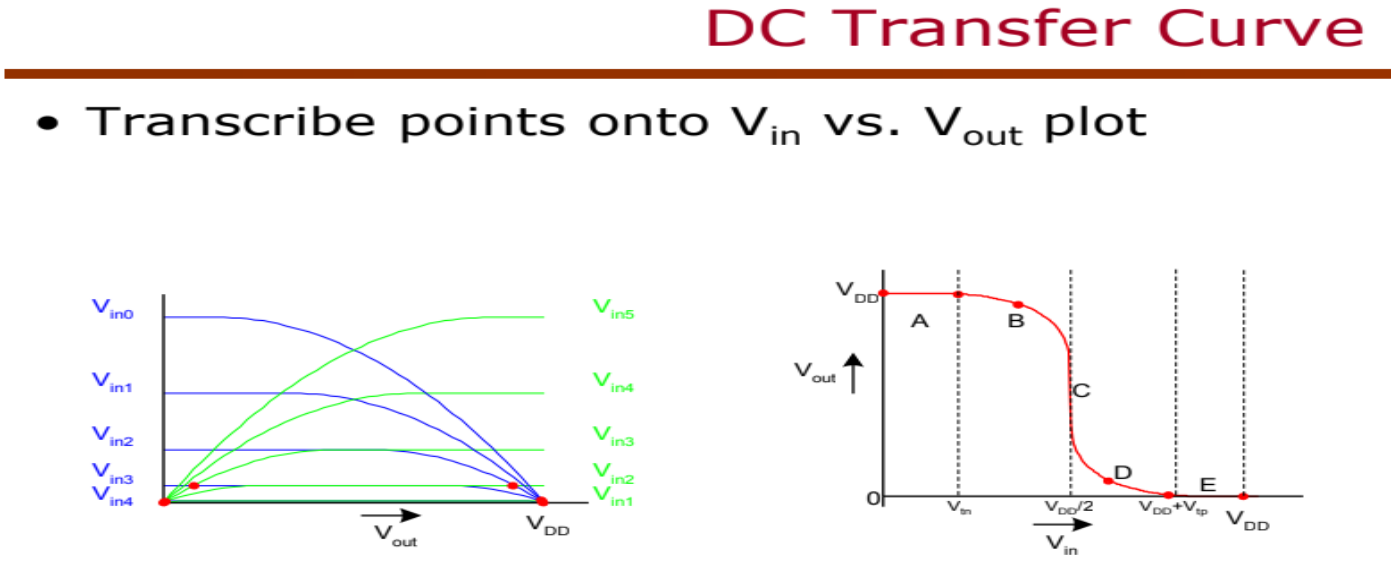
120/100, Super!!

1. (15pts) Design and simulate a matched inverter. Please add corner simulations.

I-V and DC simulation using HSPICE



<Fig 1.1> I-V Curve of nMOS and pMOS along with VTC of the Inverter



<Fig 1.2> Theoretical approach of DC Transfer Curve

As you can see in <Fig 1.1> and <Fig 1.2>, the simulation result is reasonable. Analysis of the current equation of MOSFET is omitted.

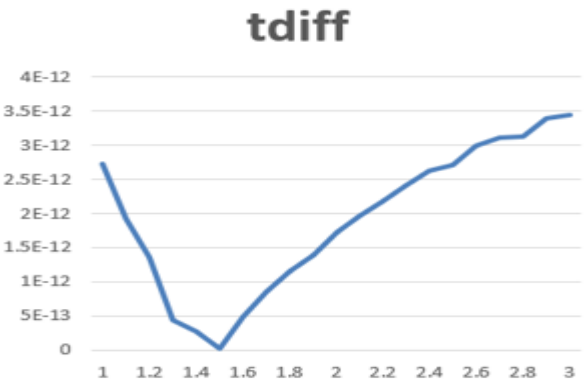
Noise Margin

```
**********************************
******  option summary
******
runlvl   = 3          bypass   = 2

 Opening plot unit= 15
file=output3/stimulus_3.pa0

vil= 4.6857e-01
voh= 9.9730e-01
vih= 6.9116e-01
vol= 1.0609e-01
nml= 3.6248e-01
nmh= 3.0614e-01
```

<Fig 1.3> Simulation result of Noise margin



<Fig 1.4> The result of minimum t_{diff}

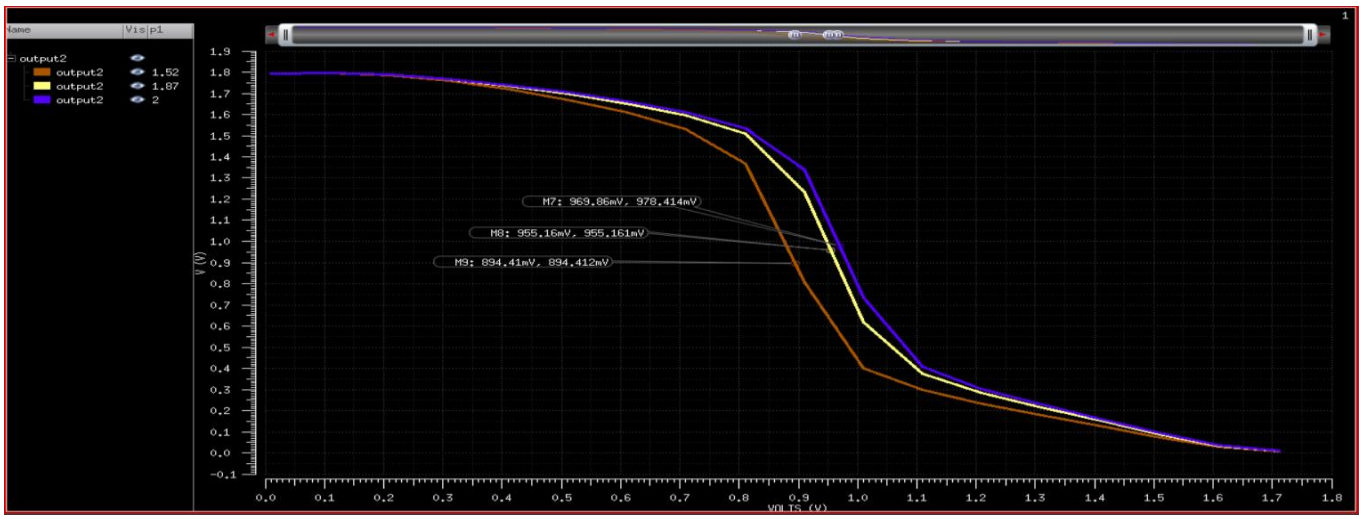
Using HSPICE arithmetic function (derivation), I computed $V_{OL}, V_{OH}, V_{IL}, and\ V_{IH}$, where the Curve's slope is -1. The Beta ratio $(\frac{\beta_p}{\beta_n})$ of the inverter was 1, and of course, the width sizing ratio $(\frac{W_p}{W_n})$ was 2.

As you can see in <Fig 1.3>, N_{ML} was greater than N_{MH}. This result indicates that our assumption of the mobility difference between hole and electron is wrong. Due to the mobility difference, we set the width sizing ratio 2. However, taking account of the computed noise margin, the inverter's sizing is not perfectly matched.

Thus, I tried to find a perfectly matched sizing to some extent. The delay difference between rising and falling was used since the delay is related to current and sizing. To be specific, using optimization and sweep function of HSPICE, the width that $t_{diff} = |t_{pdr} - t_{pdf}|$ was minimum. One thing to be aware of when performing SWEEP is that if the rising and falling times of the input signal are set too long (e.g., 20ns), it is difficult to compare the MOSFET operation properly. Therefore, the simulation was made with an input signal having a rise delay and fall delay of 20ps, and a PARAM SWEEP was performed with a delta (ΔW) of 0.01 from 1.00 to 4.00 in width.

Python visualized the result of the simulation from HSPICE. I used "ExcelWriter", which is one of the Python libraries. <Fig 1.4> shows the result.

As a result, it was observed that t_{diff} was smaller when $\frac{W_p}{W_n}$ was set to 1.52 than 2. However, this result cannot indicate that the real mobility difference between nMOS and pMOS is 1.52 times, since the BSIM4 (Berkeley Short-channel IGFET model) (LEVEL=54) model, which is the MOSFET mathematical model used in this project, considered a variety of non-ideal effects.
<Figure 1.5> shows the Voltage Transfer Curve (VTC) according to $\frac{\beta_p}{\beta_n}$ using the DC simulation function of HSPICE. The plot makes us remember our previous assignment, Quiz #1-6. As the Beta ratio is getting lower, VTC is getting skew to the left side. Furthermore, Switching voltage is also getting smaller, indicating that nMOS in the lower beta ratio will operate faster. Also, as the same with Quiz #1-6, N_{ML} will be decreased, and N_{MH} will be increased as the lower Beta ratio.

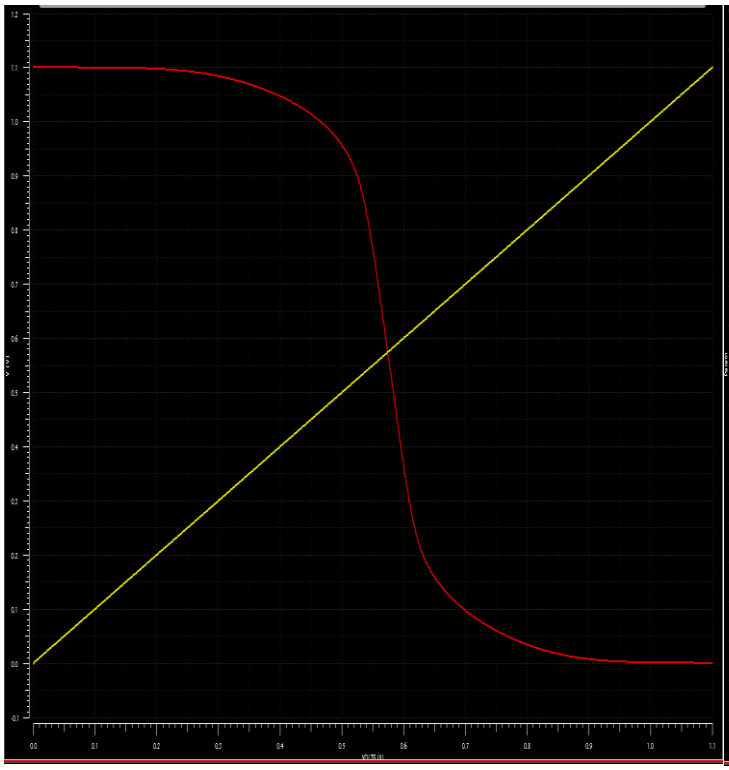


<Fig 1.5> VTC according to the Beta ratio

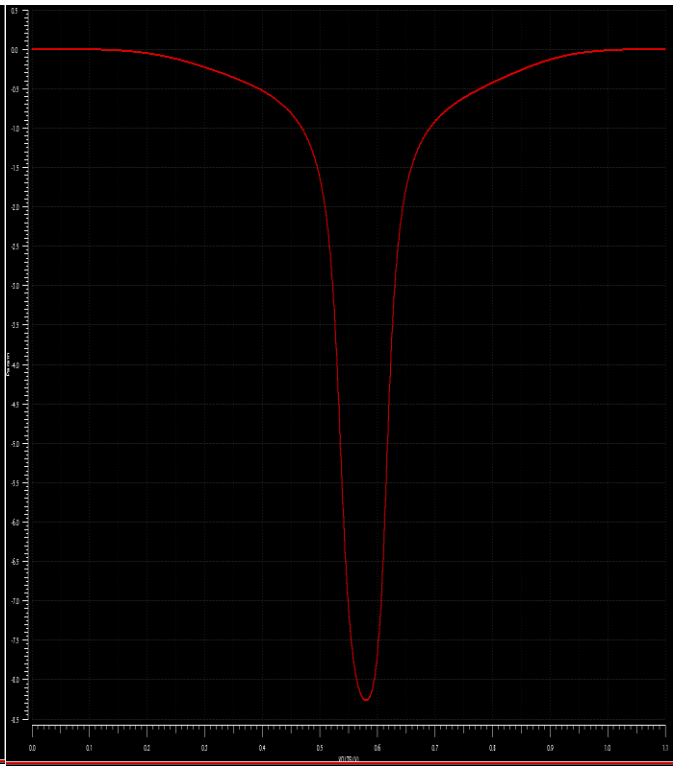
However, for convenience, the inverter with the Beta ratio as one was used in this lab.

Appendix of VTC

The intersect point which indicates the switching voltage



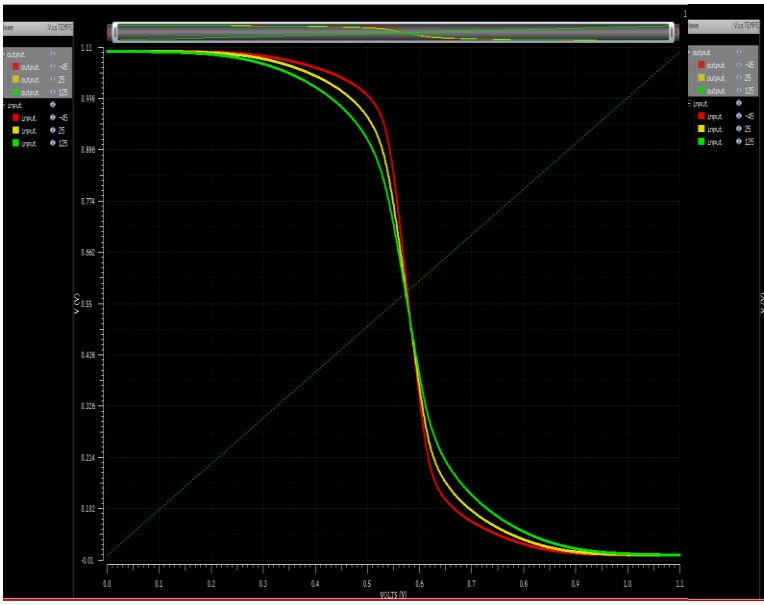
<Fig 1.6> The point indicating the Switching Voltage



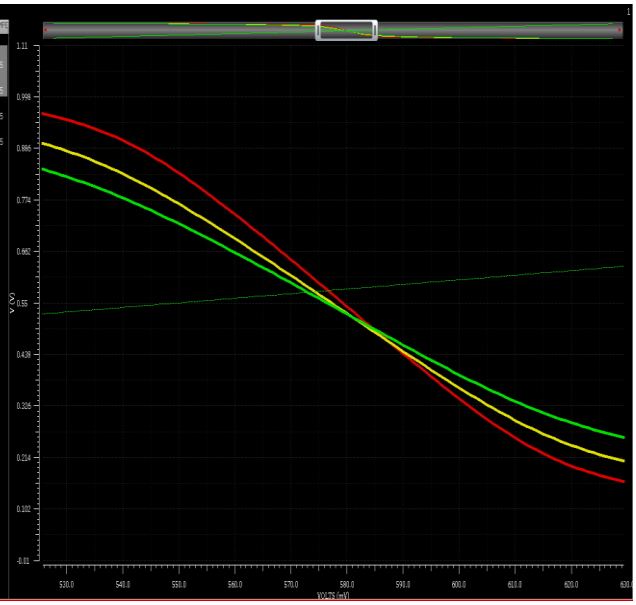
<Fig 1.7> Gain of the inverter

Corner simulation

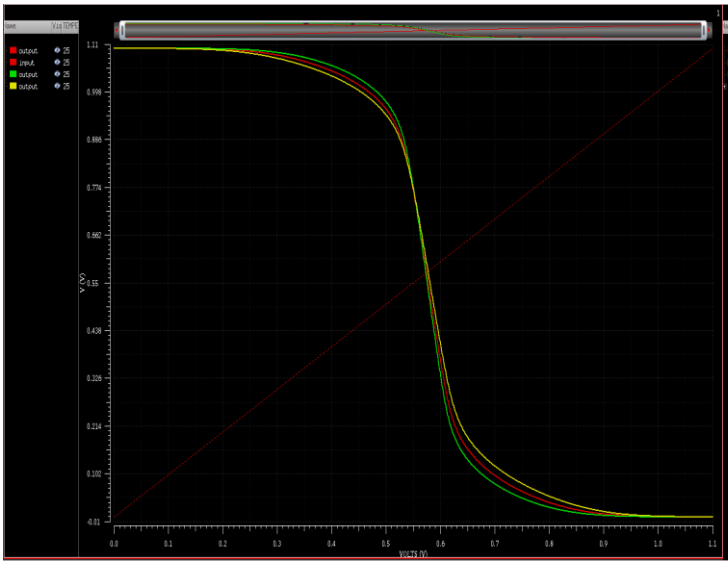
First, I simulated VTC with the Variation. In this section, 4 letters indicate Temperature, Voltage, nMOS, pMOS in order. (e.g. TTTT : 25℃, 1.1V, Typical nMOS, Typical pMOS)



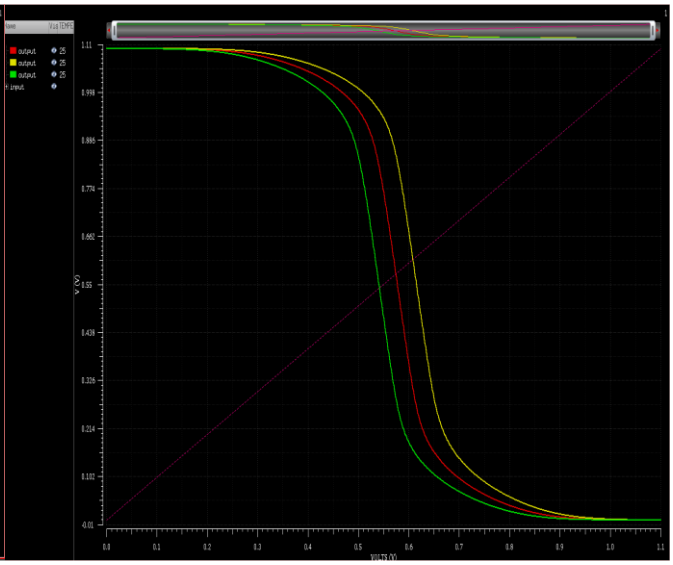
<Fig 1.8> VTC with Temperature Variation



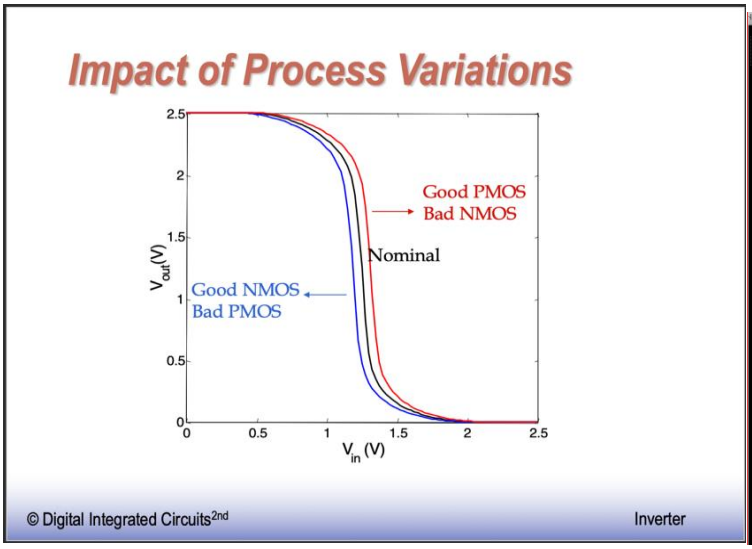
<Fig 1.9> Zoom of <Fig 1.8>

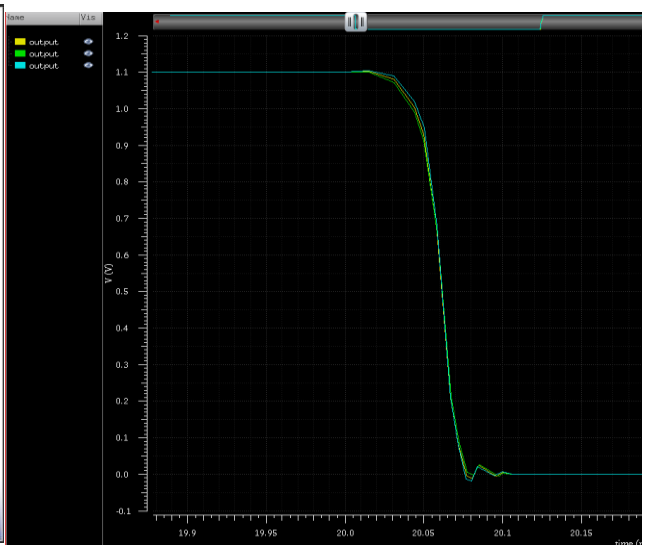


<Fig 1.10> VTC with Process Variation
(TTFF vs TTTT vs TTSS)



<Fig 1.11> VTC with Process Variation
(TTSF vs TTTT vs TTFS)





<Fig 1.12> VTC with Process Variation (TTSF vs TTTT vs TTFS)[1] <Fig 1.13> t_{pdf} of the inverter with the process variation

In <Fig 1.8> and <Fig 1.9>, the red, yellow, and green lines are the VTC with $-45℃$, $25℃$, $125℃$, respectively. From the corner simulation, the VTC Curve is steep in the order of Fast> Typical> Slow. In other words, it means switching faster. Like the <Fig 1.8> and <Fig 1.9>, <Fig 1.10> shows that the slope of the VTC Curve is steep in the order of TTFF> TTTT> TTSS (red: TTTT, yellow: TTSS, green: TTFF). As a result of <Fig 1.11> (red: TTTT, yellow: TTSF, green: TTFS), a graph in the form of <Fig 1.12> appeared.

<Table 1.1> The Delay of the inverter with the process variation

	TT	FF	SS
$t_{pdf}(ps)$	10.544	10.538	10.76
$t_{pdr}(ps)$	3.3338	4.9938	5.5981
$t_{pd}(ps)$	6.9391	7.7659	8.1789
$t_r(ps)$	17.126	20.548	16.436
$t_f(ps)$	15.228	15.935	14.564
$P_{avg}(nW)$	72.255	111.86	52.457
$P_{peak}(\mu W)$	41.628	48.971	34.217

Power consumption can be confirmed in the order of FF > TT > SS, along with Peak Power. Delay time is also generally FF> TT > SS.

Appendix:

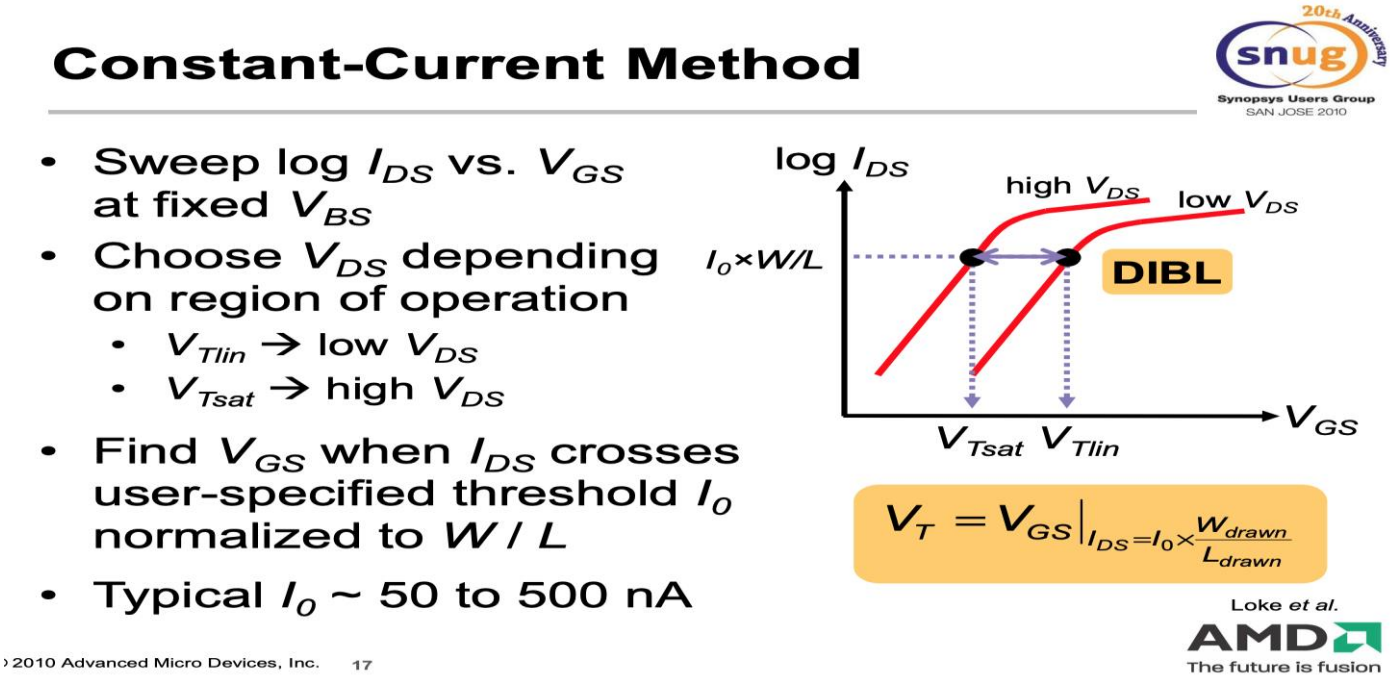
Corner Analysis

<Table 1.2> The Corner Specifications of FreePDK_45nm

	nMOS			pMOS		
	Slow	Typical	Fast	Slow	Typical	Fast
$I_{on}(\mu A/\mu m)$	1266	1350.8	1428.3	859.66	912.07	967.03
$I_{gate}(pA/\mu m)$	11881	21836	42843	9376.8	18590	36380
$I_{off}(pA/\mu m)$	1374.5	2280.4	4337.9	2122.3	3831.7	6888.4
$iV_{Tsat}(V)$	0.14762	0.11265	0.079866	0.14419	0.11017	0.075155
$iV_{Tlin}(V)$	0.2338	0.20644	0.17878	0.25617	0.22847	0.20054
$L_{eff}(nm)$	23.55	22.50	21.45	23.55	22.50	21.45

iV_{Tsat}, iV_{Tlin} are obtained using the constant-current method. As the MOSFET became shorter, non-ideal characters, including the short-channel effect, increased, and the existing extrapolation method showed a difference from the V_T measured in the actual die. This is because the square law is no longer applied as the influence of velocity saturation increases. Besides, while the MOSFET structure changes rapidly over time, the extrapolative model took much time to rewrite the formula to calculate the V_T of the changed model. So, the CC Method (Constant-Current Method), which is a method of obtaining V_T corresponding to the I_{con} determined by the circuit designer, becomes mainstream, and the method of obtaining V_T can grow like MOSFET technology.

This method's disadvantage is that I_{con} is user/process-specific, so V_T is strongly dependent on I_{con}, as shown in <Fig 1.14>. In this lab, I used $High\ V_{DS} = V_{DD}$, $Low\ V_{DS} = 50mV$, $I_{con} = 100nA$, which is the most commonly used values.

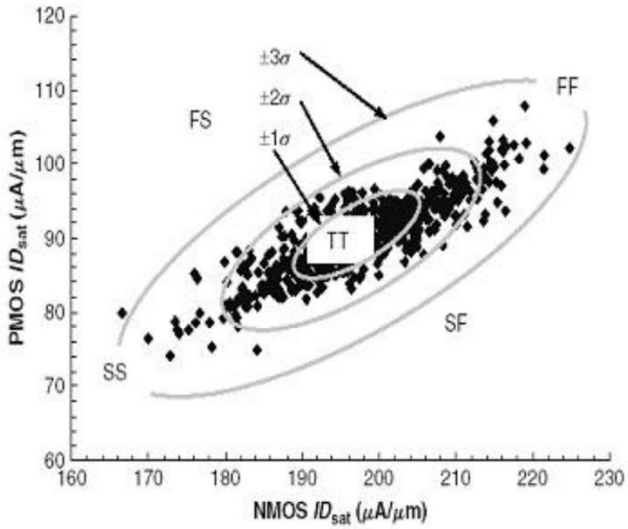


<Figure 1.14> The Constant-Current Method [2]

As follows <Table 3.1>, $I_{on}, I_{gate}, I_{off}$ is larger in order of Fast > Typical > Slow. Through the large value of I_{on}, we can estimate that the Fast process's delay time would be the fastest. Furthermore, through the considerable value of I_{gate}, I_{off}, the leakage power would be the largest in the Fast process (Of course, I_{on} plays a crucial role in large power consumption). The threshold voltage is larger in order of Slow > Typical > Fast. Through this, the MOSFET in the Fast process would flow a current the fastest, and it indicates that the MOSFET in the Fast process would react to transition rather than other processes.

Monte Carlo Simulation

In actual fabrication, the manufacturing MOSFET is not statically produced to fit the corner value. A specification of the MOSFET is estimated through Monte Carlo Simulation, which is based on statistics.



Process variation map for PMOS and NMOS devices.

<Fig 1.15> The example of Monte Carlo Simulation of MOSFET [3]

Another reason for proceeding with Monte Carlo is that Fixed Corner Analysis is pessimistic so that it can be overdesigned with an unrealistic, unphysical extreme combination.

A statistical model is required for Monte Carlo Simulation, but most foundry companies provide this model. However, even if there is no, it can be done by building a model, and you need to know the parameter value to be entered in the following equation.

$$P_j = agauss(P_{jTT}, \frac{1}{2}(P_{jFF} - P_{jSS}), \sigma)$$

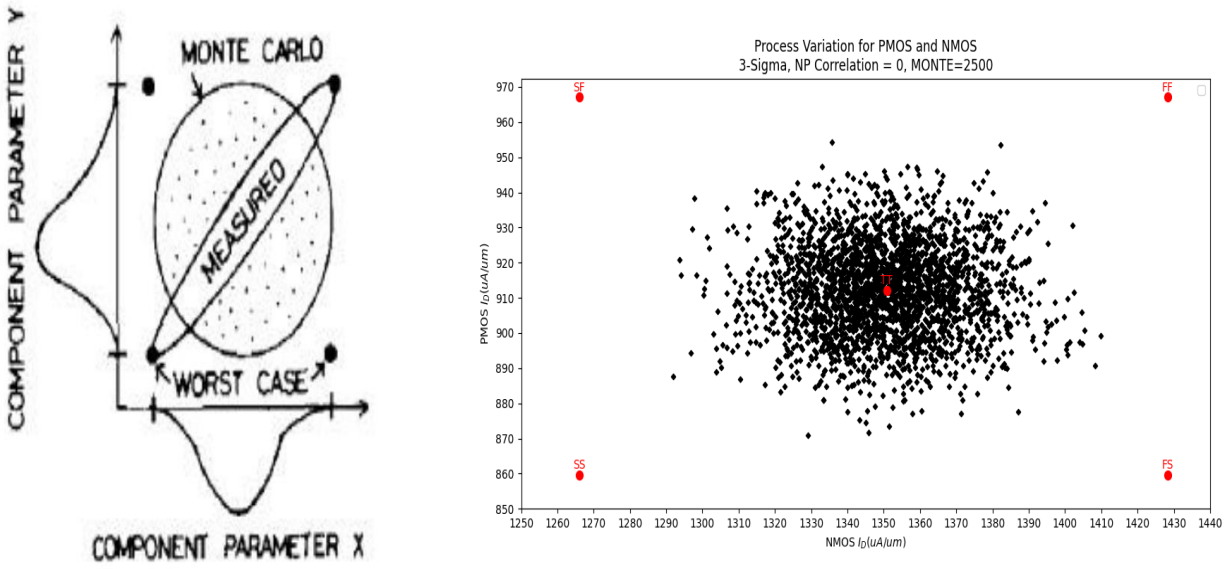
$agauss$: Hspice absolute gaussian[6]
P_{jTT} : Typical Library's parameter value
P_{jFF} : Fast Library's parameter value
P_{jSS} : Slow Library's parameter value
σ : Sigma Difference (typically, 3 σ)

Among the Slow, Fast, and Typical model parameters, the values that change according to each corner are summarized in <Table 1.3>.

<Table 1.3> the MOSFET values according to the process in FreePDK 45nm

	nMOS			pMOS		
	Slow	Typical	Slow	Typical	Slow	Typical
TOXREF	1.17E-09	1.14E-09	1.1E-09			
TOXE	1.17E-09	1.14E-09	1.1E-09	1.3E-09	1.26E-09	1.22E-09
TOXM	1.17E-09	1.14E-09	1.1E-09	1.3E-09	1.26E-09	1.22E-09
XJ	2.05E-08	1.98E-08	1.9E-08			
NDEP	3.6E+18	3.4E+18	3.4E+18			
VTH0	0.347	0.322	0.297	-0.327	-0.302	-0.277
CF	1.283E-10	1.289E-10	1.297E-10	1.26E-10	1.267E-10	1.274E-10
LINT	3.225E-10	3.75E-10	4.275E-10	3.225E-10	3.75E-10	4.275E-10

HSPICE can simulate Monte Carlo, but unfortunately, if the NP Correlation value is unknown, the unphysical combination result is derived, as shown in <Figure 3.2> [3]. <Figure 3.3> shows the Monte Carlo Simulation plot of the FreePDK 45nm with 3σ Variation and without correlation.

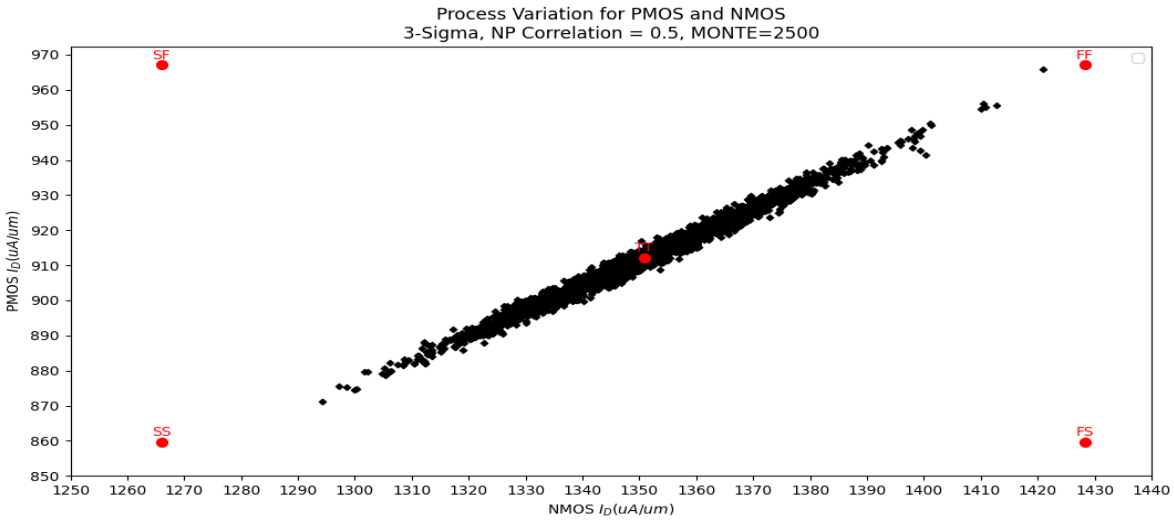


<Figure 1.16> The example of the Monte Carlo with 3σ Variation and without correlation (left) [3]
<Figure 1.17> The Monte Carlo with 3σ Variation and without correlation using HSPICE (right)

As I mentioned above, the foundry companies usually give the NP Correlation value, but we cannot know this value because we use the educational library. However, if you know, you can run Monte Carlo Simulation through the following equation.

$$P_j = P_{jTT} + \frac{1}{2}(P_{jFF} - P_{jSS})(\sqrt{c} \cdot agauss(0,1,\sigma) + \sqrt{1-c} \cdot agauss(0,1,\sigma))$$

In this lab, NP Correlation = 0.5 was arbitrarily set, and the Monte Simulation was conducted. <Figure 1.18> shows the resulting plot.

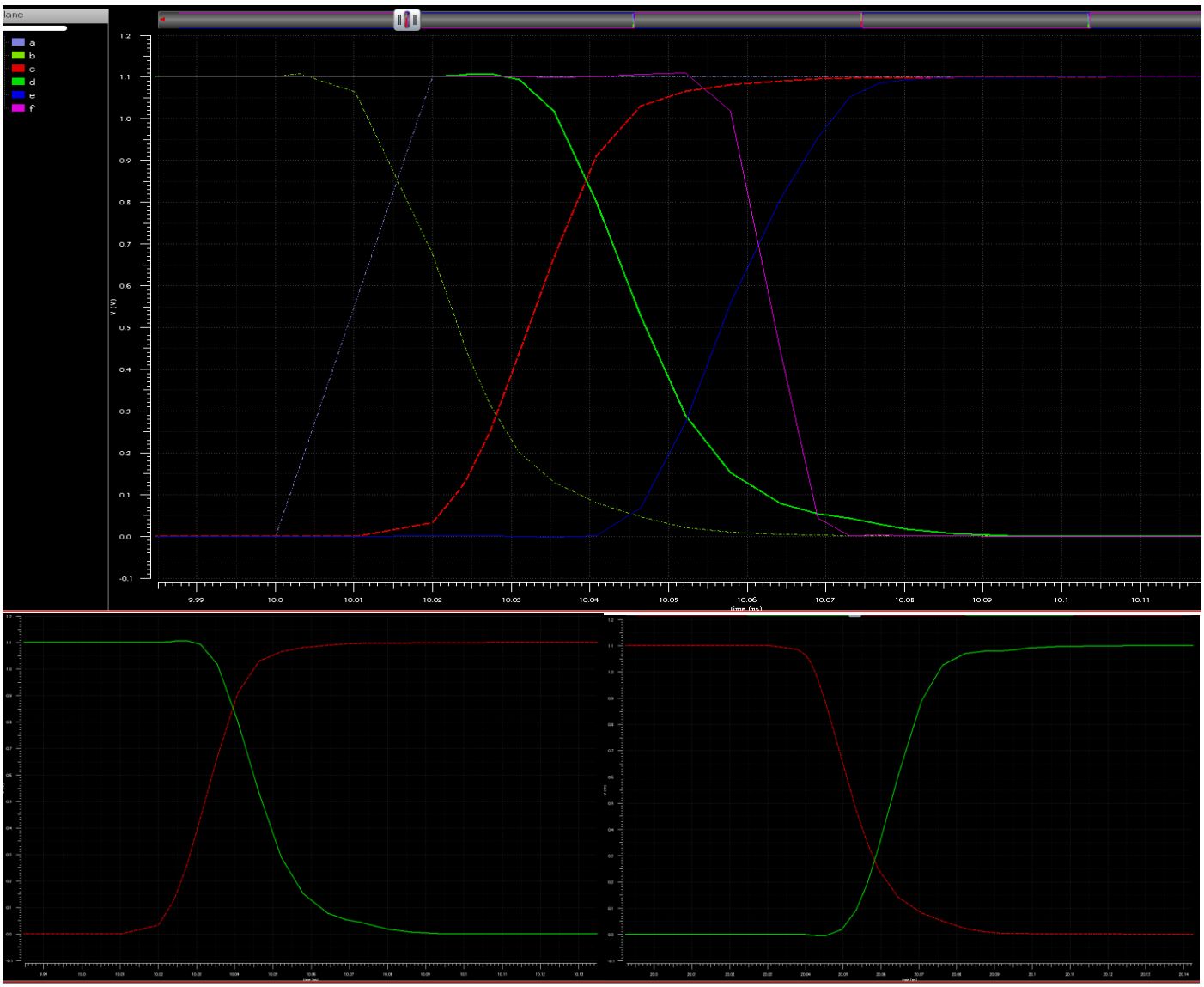


<Figure 1.18> A Monte Carlo with 3σ Variation and NP correlation as 0.5 using HSPICE

2. (15pts) Measure FO4 delay and the FO4 power. For this experiment, please refer to the HSPICE lecture note or the textbook.

HSPICE Program (Refers to attachment: Lab01/HSpice/Q2/)

.



<Fig 2.1> All nodes result of FO4 Simulation which refers to our handbook (Top)
<Fig 2.1> DUT's t_{pdf} of FO4 Simulation which refers to our handbook (Bottom Left)
<Fig 2.1> DUT's t_{pdr} of FO4 Simulation which refers to our handbook (Bottom Right)

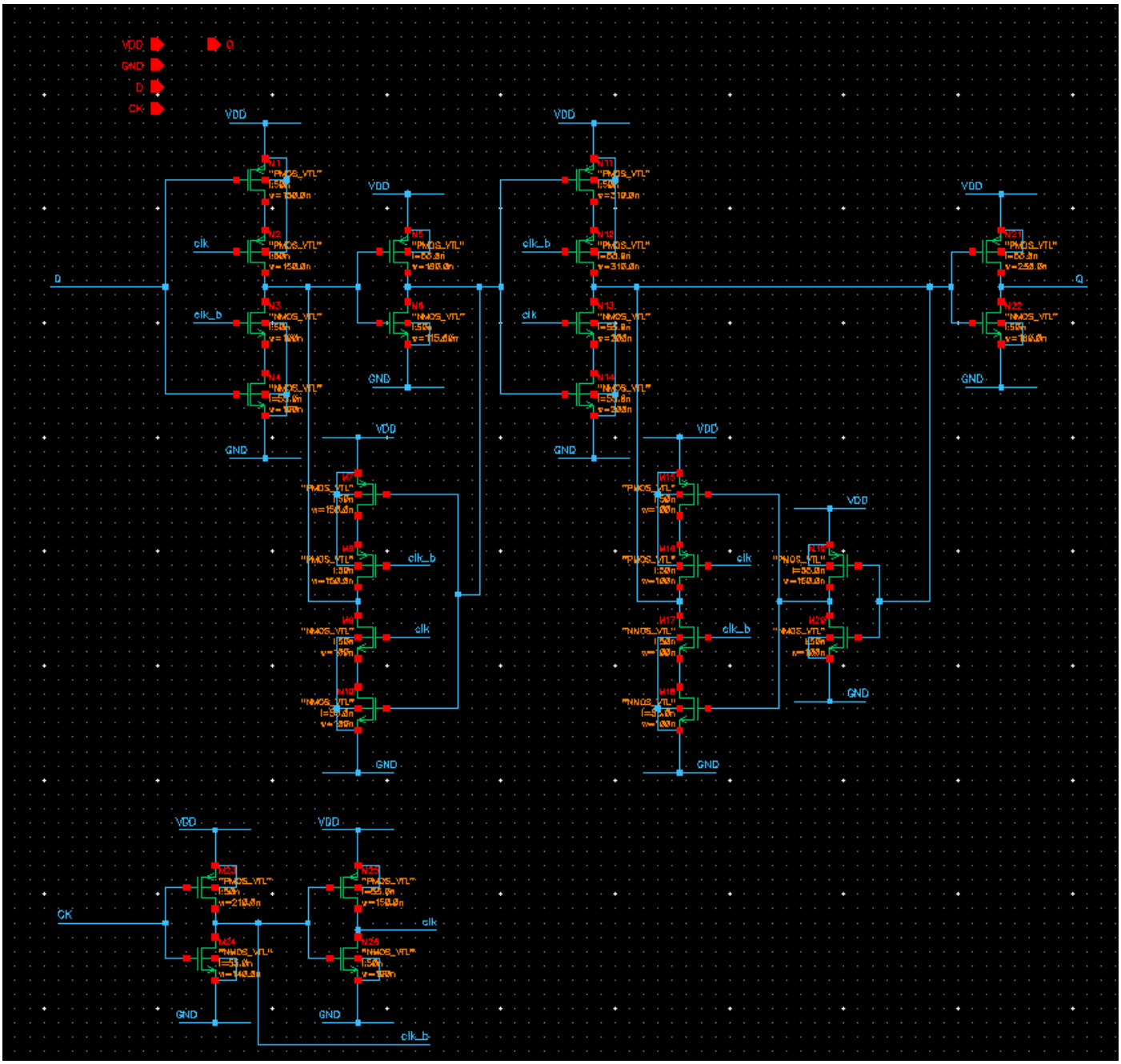
As we learned in 10/20 lecture, the formula, $\hat{N} = \log_{\rho} F,$ achieves the least delay in a multistage path. The theoretical best value of ρ is in the range of 2.4 to 6. Using a stage effort of 4 is a rule of thumb. Thus, FO4 is deemed as a representative logic gate delay. To illustrate how to obtain FO4's realistic delay, first, two inverters connected to input in series are connected since the delay of DUT (inverter) depends on its input slope. The two inverters make a realistic input slope and drive node c. Moreover, because the fourth inverter's drain capacitance, which is directly connected to the DUT's output, would be increased twice due to the miller effect, DUT's load on load, which makes load inverter switches at a slower, more realistic rate, is needed. <Table 2.1> shows the result of the FO4 simulation.

<Table 2.1> A result of the FO4 simulation

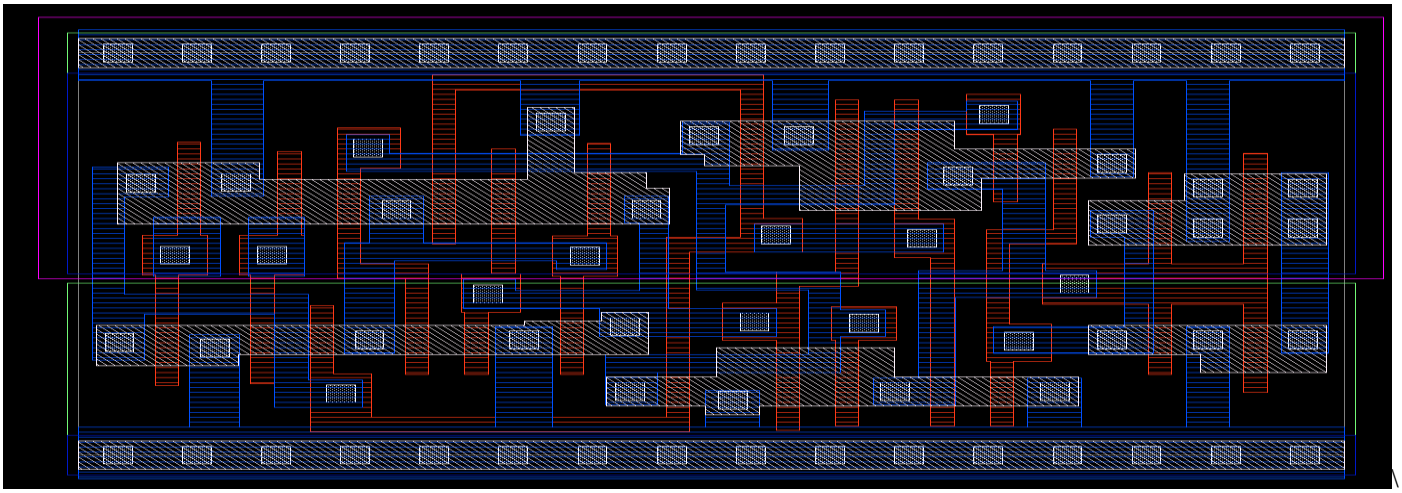
$t_{pdf}(ps)$	12.83
$t_{pdr}(ps)$	11.55
$t_{pd}(ps)$	12.19
$t_{r}(ps)$	13.50
$t_{f}(ps)$	16.15

3. (20pts) Simulate CMOS D Flip-Flop. Also, try to analyze the setup time and the hold time.

HSPICE Program (Refers to attachment: Lab01/HSpice/Q3/)



<Fig 3.1> The Schematic view of D-FF

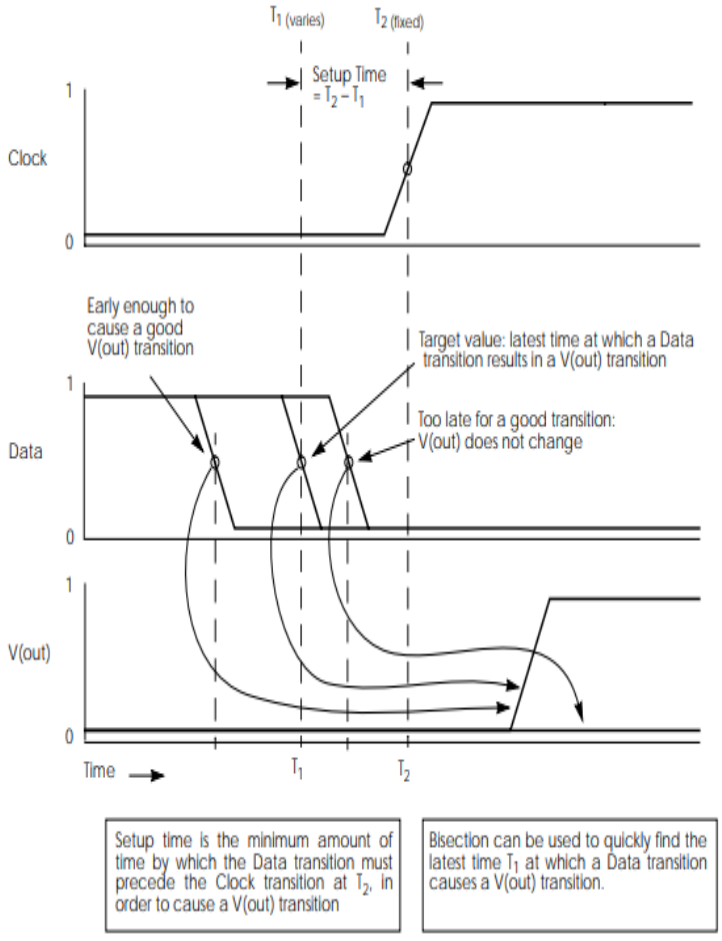


<Fig 3.2> The Layout view of D-FF [4]

I designed D-FF referred to Samsung Library. Unlike a conventional low-risk D-FF using transmission gates, the designed D-FF uses a tri-state inverter (clock gating). The tri-state D-FF reduces the sub-threshold leakage power due to staked MOSFET. However, stacked MOSFET weakens the drive strength so that D-FF's delay goes down. The D-FF Sizing was manipulated for the correct operation of writing and retention. Furthermore, referring to the dissertation titled Skewed Flip-Flop and Mixed-V_T Gates for minimizing leakage in sequential Circuits, some of the transistors in the D-FF have large gate length. The dissertation showed that the increase in gate length does not significantly reduce leakage in terms of a total cell, even though a 10% increase in gate length reduces 77% leakage in 45-nm technology. However, the dissertation divided groups of transistors that make up leakage for states of D and Q. It allocated mixed gate length according to divided groups. The proposed D-FF had an average 20% decrease in leakage and the same switching power than the original D-FF. [5]

In terms of timing, especially hold time, the designed D-FF has an advantage. The D-FF has a negative hold time, whereas the conventional D-FF has a positive hold time. To answer the question as to how the hold time is negative, we must review a definition of the hold time along with the setup time. The setup and hold times describe the limits relative to the active clock edge of a 'window' within which the input data must be valid for the data to be reliably recognized. In other words, the hold time is the minimum time that an input signal must remain stable after the active edge of the clock. On the contrary, the setup time is the minimum time that an input signal must stabilize to its logical level before the clock edge. Following this definition, the negative hold time can be readily imagined. When the hold time conduction, if the input signal passed from some gates in the D-FF reaches later than a first tri-state inverter or transmission gate actually off, the hold time would be negative. Thus, if the hold time is negative, the absolute earliest data is no longer valid before the active clock edge. In other words, if the hold time is negative, the absolute earliest data can be changed after the active clock edge. The negative hold time is a great advantage because the flip-flop can be connected by back-to-back following the min-delay equation ($t_{cd} + t_{ccq} \geq t_{hold}$). The negative hold time can also reap benefits when trying to increase in clock frequency.

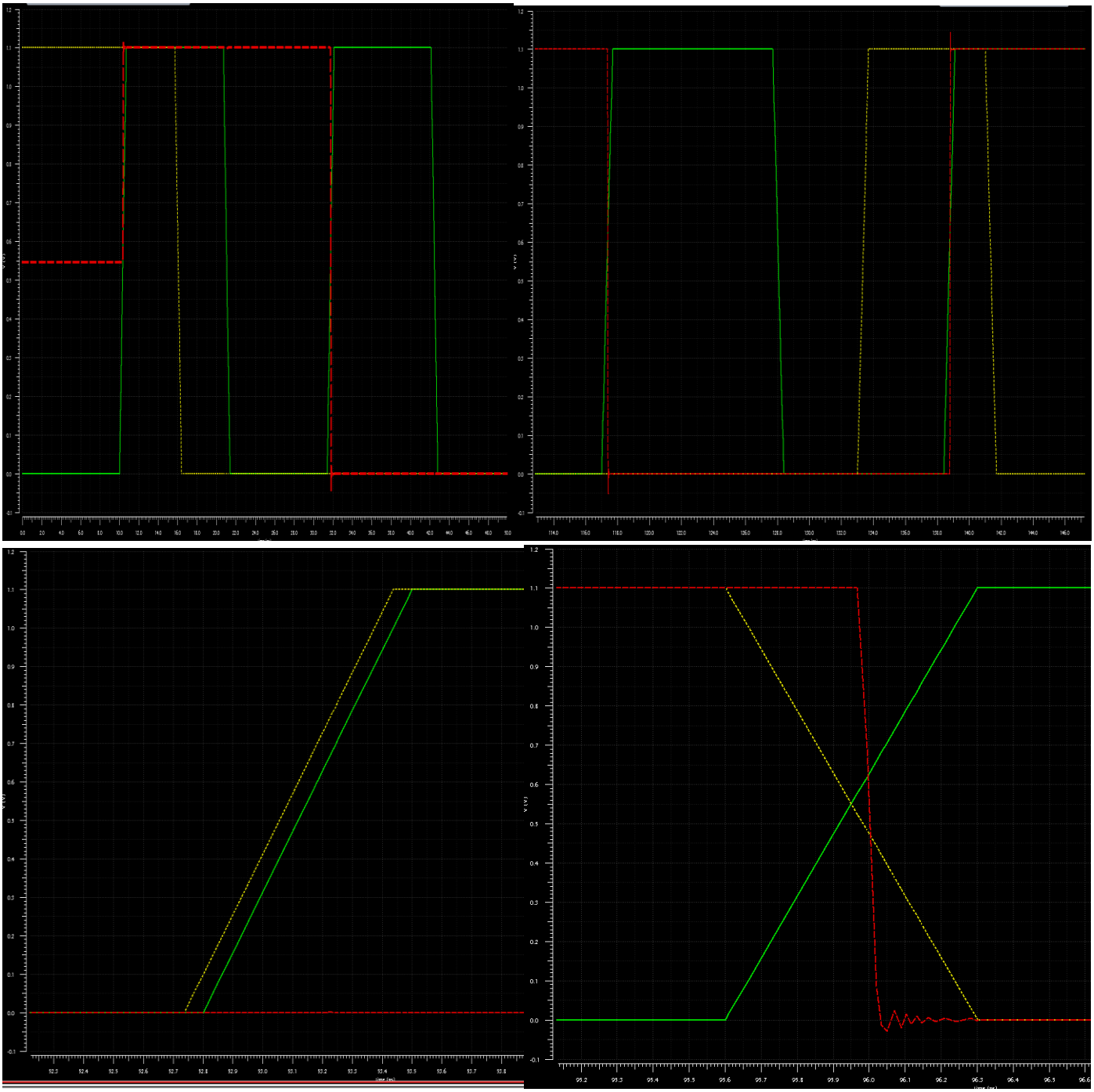
In HSPICE, to obtain the setup and hold time, the bisectional method was used. In brief, the method is separating two sections based on goal or root, generally positive and negative, and indicates the separated point. To illustrate how to find setup time, the HSPICE moves data signal to the fixed clock edge during probing Q. While moving the data, Q will be different when the data signal violates the D-FF's setup time. When Q is changed, the HSPICE computes the point, and we can know the setup time from the point. The hold time can be obtained in the same way.




```
******
dff_case11

 ******  transient analysis tnom=  25.000 temp=  25.000 *****
vout= 4.4702e-01  from= 5.4000e-09    to= 6.4000e-09
setup11= 6.4551e-11  targ= 5.7500e-09   trig= 5.6854e-09
avgpower= 3.4476e-06  from= 3.4000e-09    to= 1.1000e-08
peakpower= 2.5702e-04    at= 5.8060e-09
          from= 3.4000e-09    to= 1.1000e-08
```

<Fig 3.3> Determining Setup Time with Bisection Violation Analysis [6] (Left)
. <Fig 3.4> The result of the setup time measurement using Bisection Analysis (Right)



<Fig 3.5> A plot of Q transition 1 to 0 (Top Left)
<Fig 3.6> A plot of Q transition 0 to 1 (Top Right)
<Fig 3.7> A plot of the setup time violation (Bottom Left)
<Fig 3.8> A plot of the hold time violation (Bottom Right)

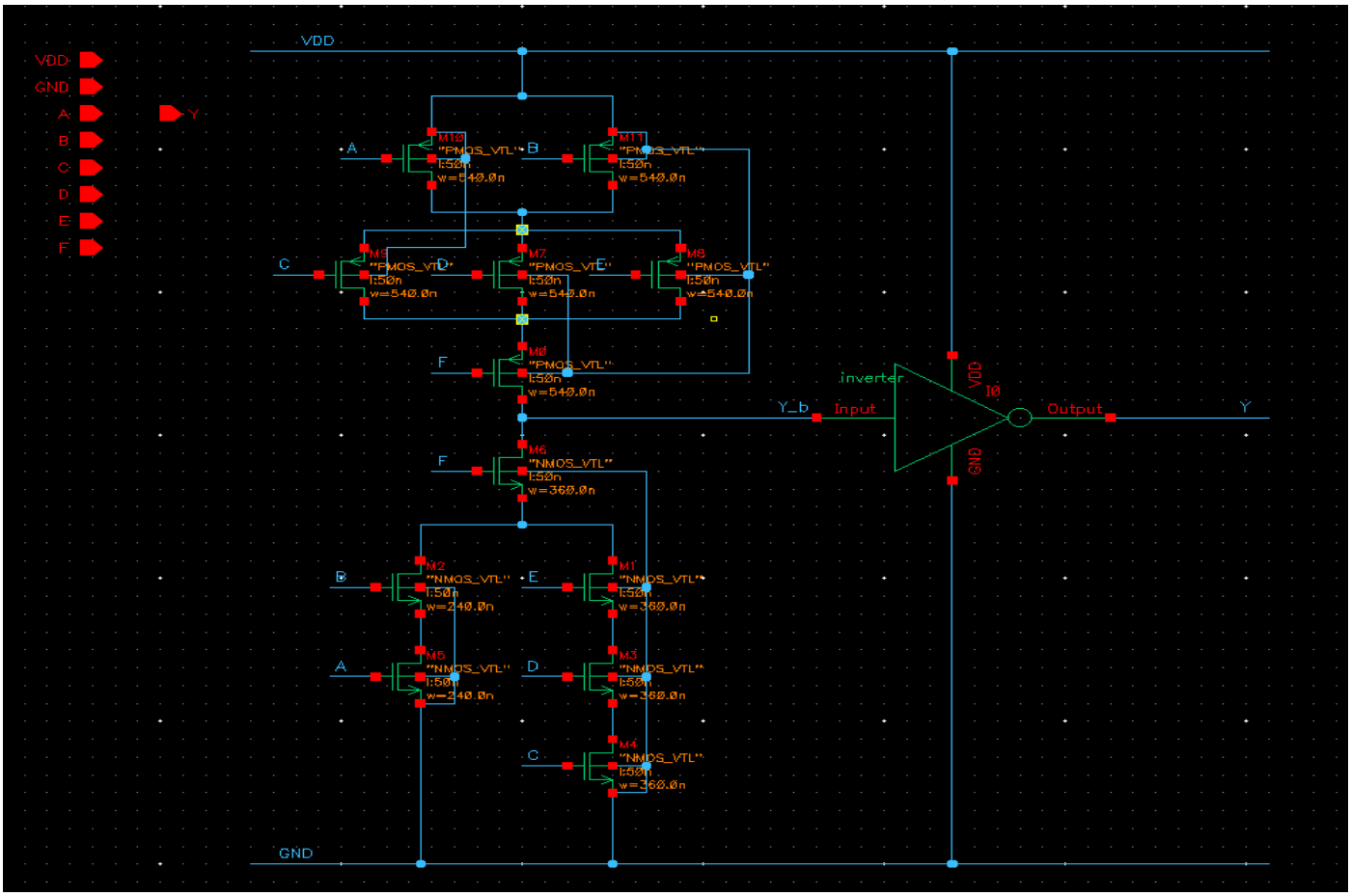
<Table 3.1> A Specification of the designed D-FF

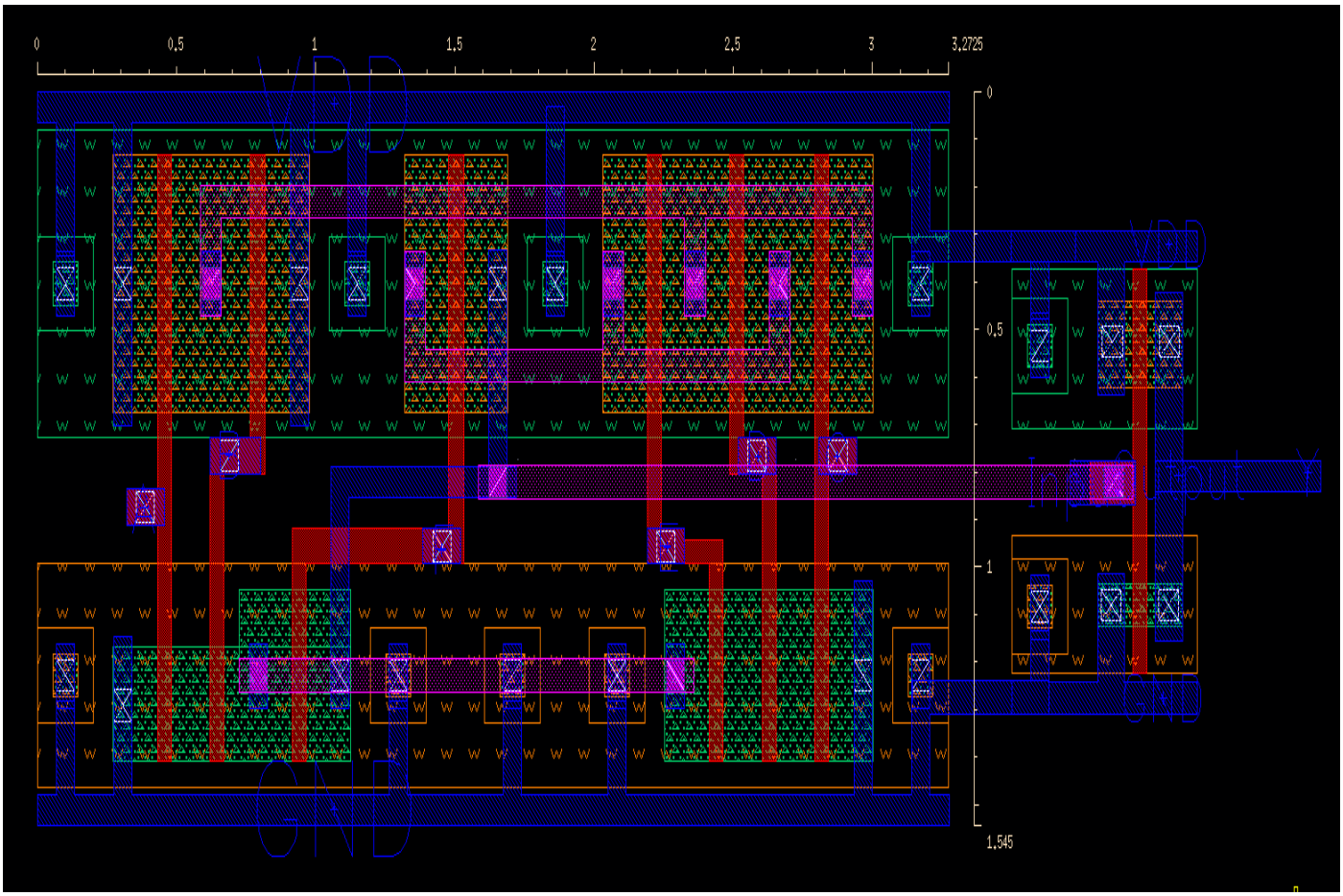
$t_{clk-q}(ps)$	56.37
$t_{setup}(ps)$	64.551
$t_{hold}(ps)$	−15.0

Before an analysis, in <Fig 3.5>, <Fig 3.6>, <Fig 3.7>, <Fig 3.8>, the red line is Q, the yellow line is D, the green line is CLK. D and clock have 0.7ns transition time.

First, in <Fig 3.5>, the initial output of D-FF was shown to the unknown-state. After that, D was changed, and Q was changed when the positive clock edge. Like <Fig 3.5>, also in <Fig 3.6>, the normal D-FF operation was observed. <Fig 3.7> shows that the input signal D violated the setup time. As you can see, the Q was not changed, even though the Data signal was changed before the clock edge. <Fig 3.8> shows the hold time violation. The input signal violated the D-FF hold time, and of course, it violated the setup time. Thus, Q should be 1 like in <Fig 3.7> situation, but Q did not retain its value. The abnormal Q output can be easily analyzed if considering which transistors are on-state depending on CLK.

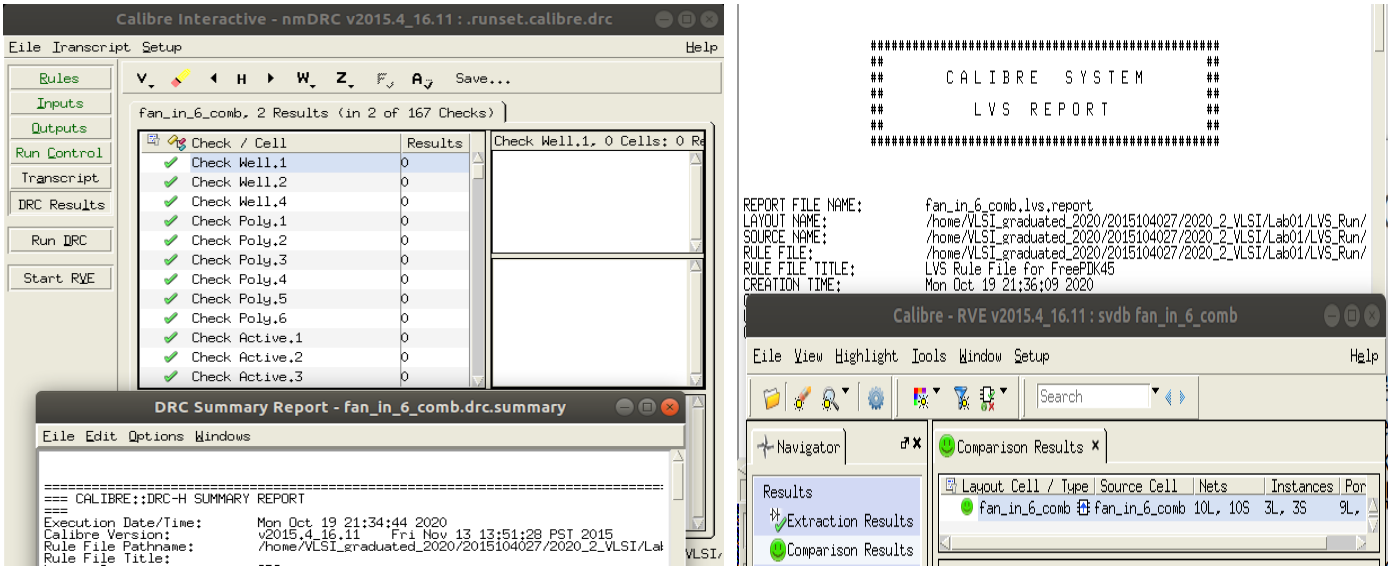
4. (50pts) Given Y=(AB+CDE)F (Complementary inputs are not available). Simulate and layout a two-stage CMOS circuit by adding an output inverter. Also, measure propagation delays and contamination delays





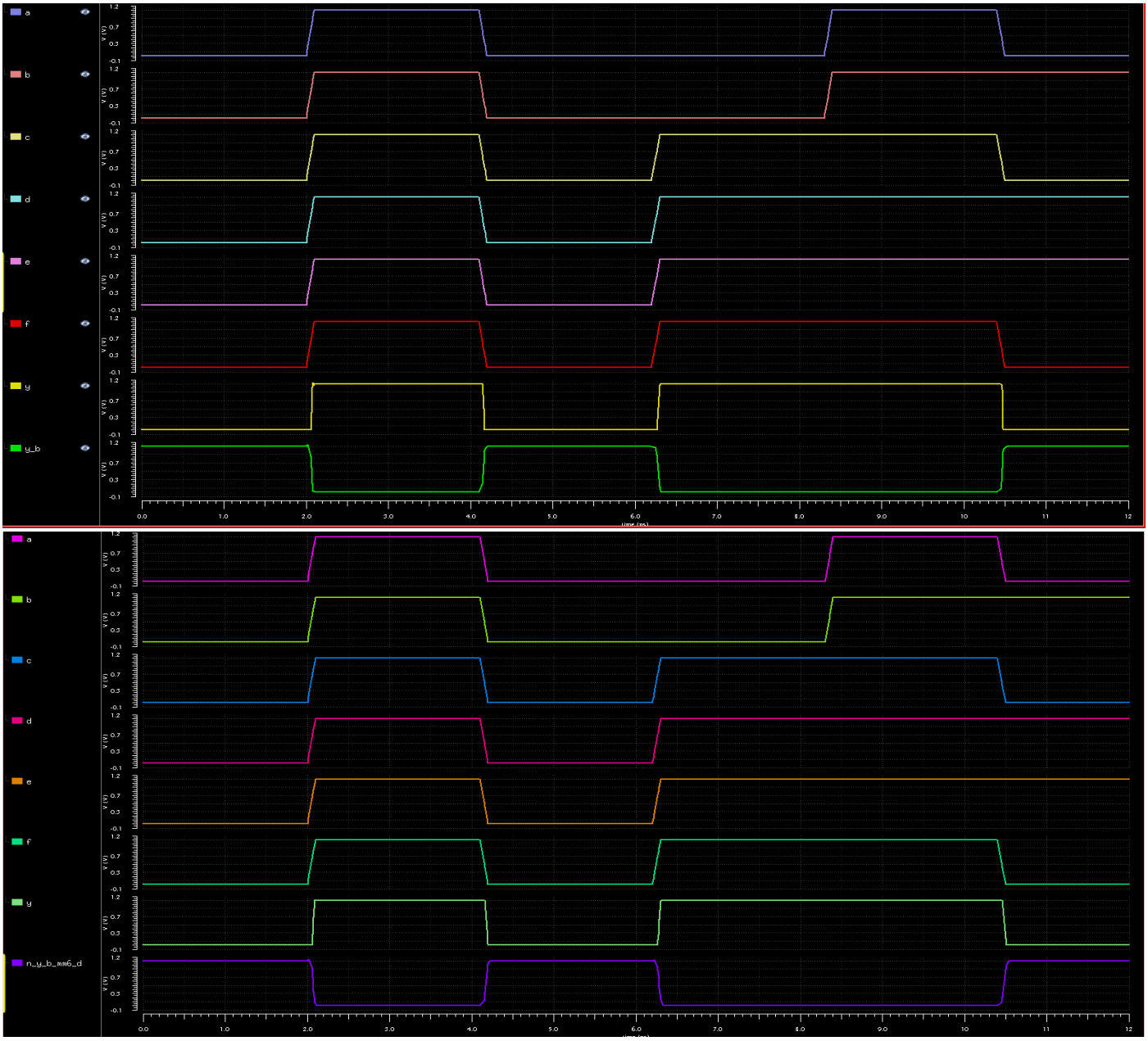
<Fig 4.1> A Schematic view of a Lab01_Q4's circuit

<Fig 4.2> A Layout view of a Lab01_Q4's circuit



<Fig 4.3> DRC result of the Lab01_Q4's layout

<Fig 4.4> LVS result of a Lab01_Q4's layout



<Fig 4.5> A plot of Pre-Simulation

<Fig 4.6> A plot of Post Layout Simulation

A, B, C, D, E, F, Y, Y_b in order of top to bottom, respectively.

<Table 4.1> The delay of the Lab01_Q4's circuit (input to Y_b, which is before the second stage)

	Pre-Sim	PLS
$t_{pdf}(ps)$	29.583	29.147
$t_{pdr}(ps)$	16.929	31.628
$t_{cdf}(ps)$	17.514	21.844
$t_{cdr}(ps)$	9.7082	23.279

A pMOS position of the F input is wrong. The position should modify to parallel like (A'+B')(C'+D'+E')+F'. (A pMOS of the F should be connected to VDD and a drain of an nMOS of the F.)

Through <Fig 4.1>, <Fig 4.2>, <Fig 4.3>, <Fig 4.4>, <Fig 4.5>, <Fig 4.6>, the Lab01_Q4's circuit was verified. Furthermore, <Table 4.1> shows the delay of the Lab01_Q4's circuit. I estimated t_{pdf} when only C, D, E, F are becoming 1, t_{pdr} when only A, C, F are becoming 0, t_{cdf} when all inputs are becoming 1, t_{cdr} when all inputs are becoming 0. The difference between the results and the calculated Elmore delay based on our previous assignment, Quiz #1-4, was not big. However, the question of why the little difference exists can be answered by limiting the Elmore delay computation. The Elmore delay is a linear delay model that makes delay estimation simple. However, the limitation is explicit.

First, the limitation is related to the problem of an input source. The input source used in the linear delay model assumes an ideal pulse or step function, and there is no rise or fall time in the input. However, in the simulation, there is no source with 0 rise and fall as in real, and it has a slope unconditionally. As a result, the propagation delay is measured longer than expected by the input's rise and fall times. Moreover, when one has switched on a multiple-input gate, the model assumes that others remain stable. However, in reality, and simulation, if the series transistors are turned ON simultaneously, the delay becomes longer than expected. The reason is that the series transistor is partially (sequentially) ON at the beginning of the transition. In the opposite case, if the parallel transistors are turned ON simultaneously, the delay becomes shorter than expected because current flows in both parallel transistors. Also, the delay varies according to the input pattern of the transistor.

Secondly, the model neglects the velocity saturation. In the long-channel, the current decreased linearly to L. However, when the transistor is fully velocity-saturated, I and R become independent of L. This means that R_{sum} in series transistors is smaller than R_{sum} in the calculation. This phenomenon is more dominated in nMOS than in pMOS because the mobility of nMOS is greater, and the degree of velocity saturation is more substantial. In conclusion, since the $R_{sum}$of the series transistor becomes smaller in reality and simulation, the delay comes out smaller than expected. Therefore, to drive the same current with the unit inverter, the size must be designed differently from the sizing calculation based on the logical effort.

$$\frac{I_{dsat-N-series}}{I_{dsat}} = \frac{(V_{DD}-V_t)+V_c}{V_{DD}-V_T+NV_c} \quad (V_c = E_cL, \text{ saturated voltage})$$

The series sizing ratio should be determined through the above equation, and this ratio will have different values for each process. Overall, to match the expected value, the series-connected part should be designed to be smaller than the size obtained when calculating the conventional logical effort.

Lastly, it is the difference in capacitance. Bootstrapping occurs when the gates are connected due to the coupling capacitor existing between the two gates' input and output. This phenomenon, also known as the Miller effect, is not a remarkable effect in digital, but it cannot be denied that it does. We calculated as C = C_{gs}, but in reality and simulation, the C_{gd} value is added (C = $C_{gs} + C_{gd}$). The reason is that when the transistor is partially ON, OFF, that is, in the linear region, C_{gd} is increased by the value multiplied by the gain of the circuit. Hence, the phenomenon can be observed that the C_{gd} value, which was initially ignored, becomes not negligible, and the delay becomes more extensive than expected.

In terms of PLS, the long delay results from not only parasitic RC but also the small number of Contact and Via. When all inputs switches, the strong current flows only through one Contact and Via so that their resistance can be substantial. Through the case of t_{cdr}, the explanation can be deemed as plausible.

Reference

[1] Jan M. Rabaey, Anantha Chandrakasan, Borivoje NiKolic, *Digital Integrated Circuits: A Design Perspective*, Prentice Hall India, 2002.

[2] Alvin Loke, Zhi-Yuan Wu, Reza Moallemi, Dru Cabler, Chad Lackey, Tin Tin Wee, Bruce Doyle "Constant-Current Threshold Voltage Extraction in HSPICE for Nanoscale CMOS Analog Design", *SNUG*, San Jose, 2010.

[3] Kerwin Khu, "Statistical Modeling for Monte Carlo Simulation using Hspice", *SNUG*, Singapore, 2006.

[4] The Primitive in the Samsung 130nm Library, Samsung, Yong-in, 1998.

[5] Jun Seomun, Jae-Hyun Kim, and Youngsoo Shin, Skewd Flip-Flop and Mixed-Vt Gates for Minimizing Leakage in Sequential Circuits, IEEE, 2008.

[6] HSPICE User Guide Simulation and Analysis, Synopsys Inc, Mountain View, CA, USA, 2008.