Lab 3 Digital Circuit Simulations ECE334

Objective:

To design/simulate some CMOS logic gates, using SUE/HSPICE.

Preparation:

For the following questions, assume $V_{tn} = 0.43V$, $V_{tp} = -0.616V$, this was obtained from the 0.25μm CMOS process model file. Also assume that $\mu_n C_{ox} = 88\mu A/V^2$, $\mu_p C_{ox} = 30.67\mu A/V^2$, which was obtained by simulating a diode connected transistor and using the drain current to extract the value of the transistor transconductance parameters. For the rise and fall time calculations, you can approximate the transistors by their equivalent resistance.

IMPORTANT NOTE 1: Please pay attention to the distinction between values for **W** and **W/L ratio**. For the 0.25μm process used by SUE, the minimum length for both PMOS and NMOS is 0.25u (equivalent to "lp_min" and "ln_min" in the software). As a result, for ex. in the inverter below, the PMOS width is equal to 6*0.25um =1.5um. **The software takes in the value for W and L separately and not their ratio.**

IMPORTANT NOTE 2: Instructions on how to increase simulation time are given in part 3 of the FAQ section.

IMPORTANT NOTE 3: Instructions on how to measure setup time and clock-Q to propagation delay for the flip-flops are given at the Appendix section.

P1) For the unit-sized inverter shown in figure **Error! Reference source not found.**, find analytically the approximate rise and fall times (10% to 90%) assuming the inverter derives a **0.3pF** load capacitor.

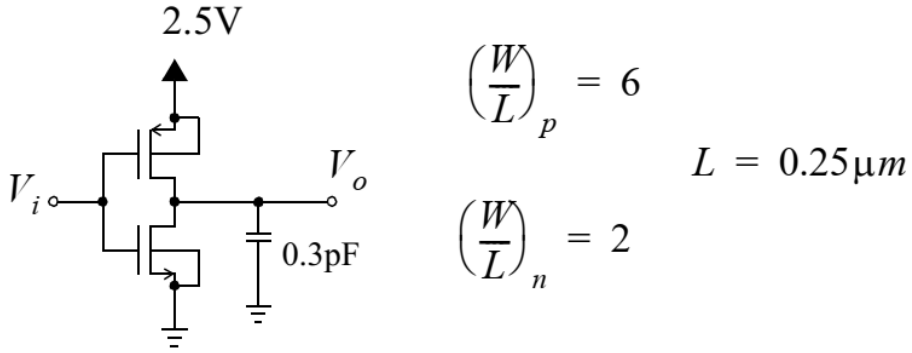


Figure 1: CMOS Inverter

P2) Draw the schematic of the CMOS logic gate that implements the function $Y = \overline{A + BC}$.

(a) Identify the input conditions for the worst case rise/fall times and the best case rise/fall times.
(b) Size your transistors, such that for the worst case the gate achieves the same rise and fall times of the unit-sized inverter of P1 when your gate is driving a load capacitance of **1.5pF**.
(c) Using the equivalent resistance to model the transistors and neglecting the diffusion capacitances, estimate the worst case rise and fall time of your gate and compare them to that of the unit-sized inverter in P1.
(d) Using the equivalent resistance to model the transistors and neglecting the diffusion capacitances, estimate the best case rise and fall time of your gate.

P3) Using a truth-table, briefly describe the operation of this flip-flop.

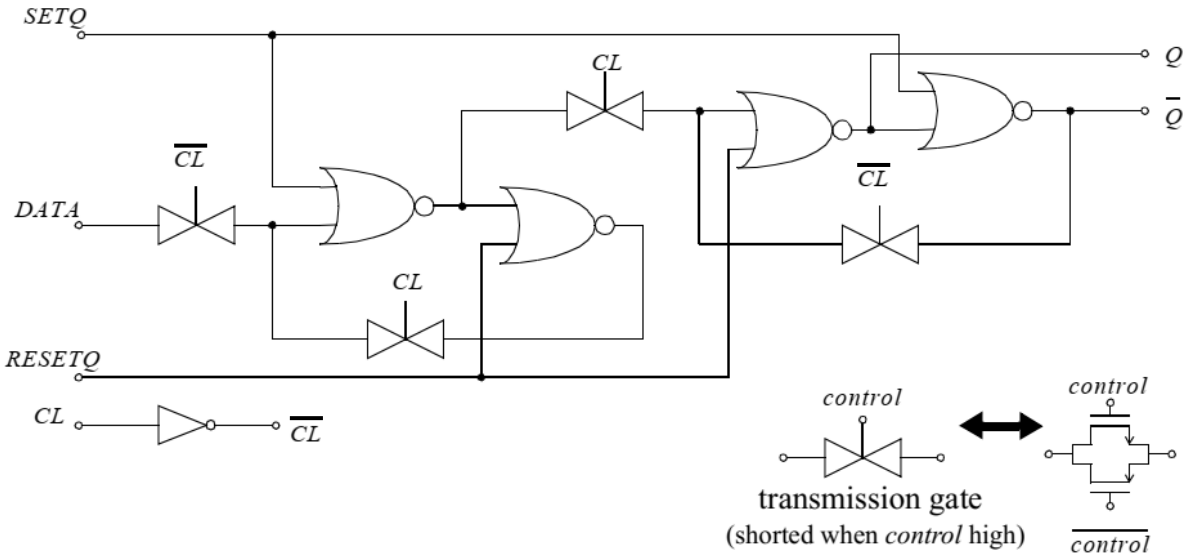


Figure 2: Logic Diagram of "D-type" flip-flop [ref. Section 10.3.4 of textbook – 4th edition]

Lab Work

L1) Simulate the inverter of Figure **Error! Reference source not found.** using a 0.3pF load capacitance. Find the rise and fall times, and compare your simulation results with the analytic results.

L2) Simulate the CMOS gate you designed in P2 assuming it drives a load capacitance of **1.5pF**. Find the worst/best case rise and fall times, and compare your simulation results with the analytic results. How does your gate's worst and best case rise/fall times compare to that of the inverter of part L1?

L3) Simulate the D-type flip-flop in figure 2 and show that it operates as expected. Use minimum length for all the transistors, and assume $W_n = 3\mu$m and $W_p = 9\mu$m for all the switches and the clock inverter. But size the NOR gates to achieve a worst case rise/fall times equal to that of the clock inverter.

You will now need to increase the transient simulation time. To do that, follow the instructions in **part 3 of the FAQ section**, on page 5.

Divide your plot into 4 panels, and plot the following:

panel 1: CL, CL
panel 2: SETQ, RESETQ
panel 3: DATA
panel 4: Q, Q.

Hint: To demonstrate your flip-flop operation, it is recommended to use the following settings for your input signal sources:
CL: pulse(t_d=0ns, t_r=0.5ns, t_f=0.5ns, pulse-width=10ns, pertiod=20ns)
SETQ: pulse(t_d=55ns, t_r=0.5ns, t_f=0.5ns, pulse-width=40ns, pertiod=1000ns)
RESETQ: pulse(t_d=175ns, t_r=0.5ns, t_f=0.5ns, pulse-width=40ns, pertiod=1000ns)
DATA: pulse(t_d=5ns, t_r=0.5ns, t_f=0.5ns, pulse-width=20ns, pertiod=40ns).

Your input signals should look similar to Figure 3.

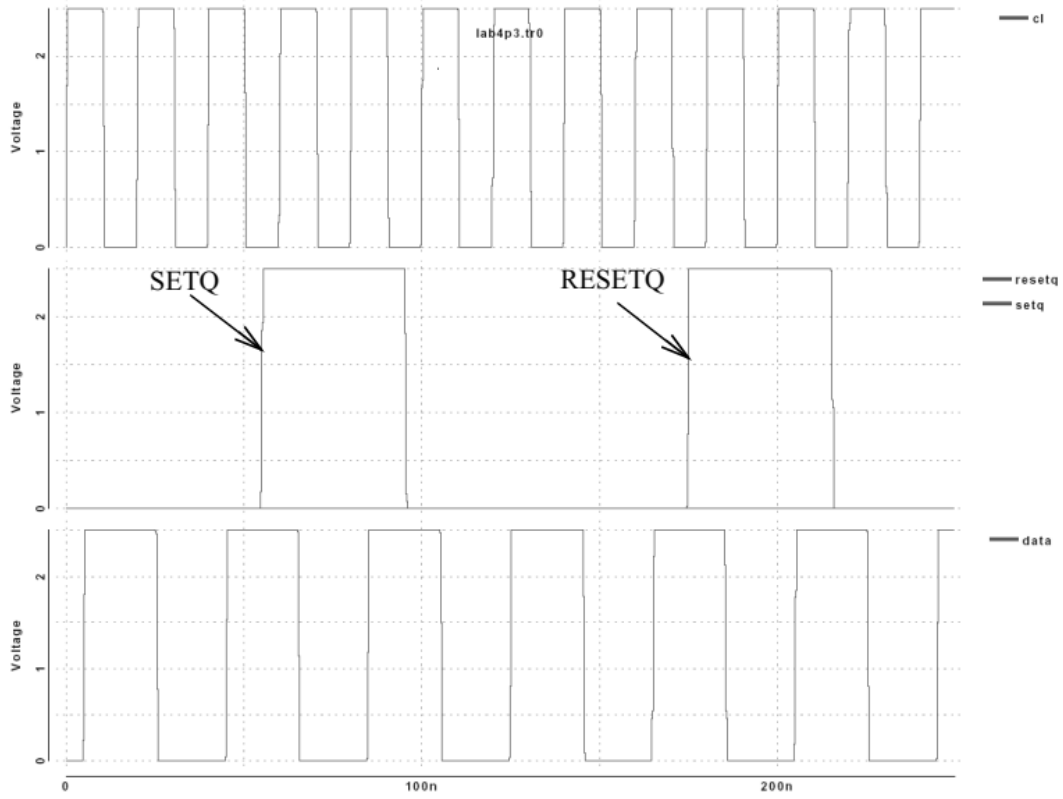


Figure 3: Input signals applied to verify the operation of the "D-type" flip-flop

L4) For the flip-flop of Figure 2., change the input signals to simulate and measure the **setup time**.

L5) For the flip-flop of Figure 2., measure the **clock-to-Q propagation delay**, or $\mathbf{t_{PCQ}}$.

L6) Create a sequential circuit consisting of two flip-flops from L3, and a chain of 4 identical inverters in between as shown in Figure 4. Apply the same input signals as from L1 and use the same clock signal for both flip-flops. Find the combined propagation delay of the 4 inverters. For the inverters, use the same sizes as L1.

• Simulate your synchronous circuit and confirm that it operates correctly.

• Compute the **maximum clock frequency** of the synchronous circuit using the max-delay constraint. Now, in your simulations, increase the clock frequency until your circuit no longer operates "correctly." Compare this simulated frequency to your maximum frequency.

FAQ:

1. In part L1, the output waveform looks weird. It does not behave like an inverter in simulation.

• Check the **pulse width of your input source**. Make sure enough time is given for the output rising or falling completely. (i.e., make sure your circuit is "fast" enough to follow the input.) You can increase the input pulse width--and period respectively--to address this issue.

2. When you open the spice file, you should find the .tran command looks like:

^.tran 10p 16n

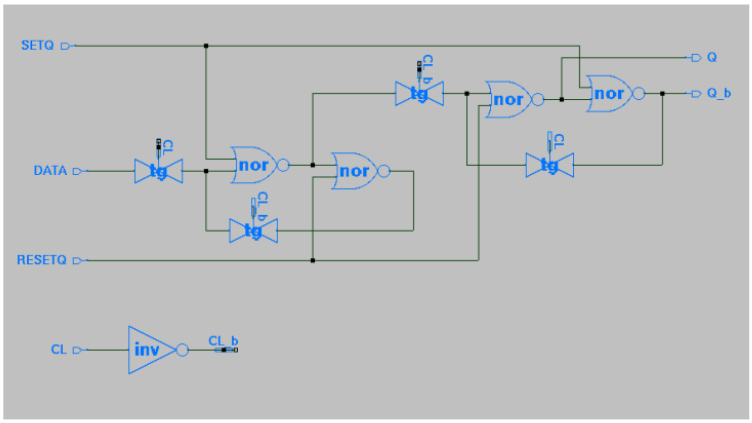
The first parameter value (10 ps) specifies the **step size**, which instructs the simulator how often it should sample one simulation data, and the second parameter value (16 ns) specifies the end time of simulation.

3. What's the procedure to **increase the transient simulation time** in part L3?
To do that, add a spice command inside your schematic diagram:

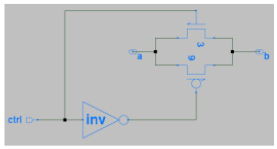
a) On SUE schematic viewer, press "t". That puts you into typing mode.
b) Click anywhere on the schematic and type the following:

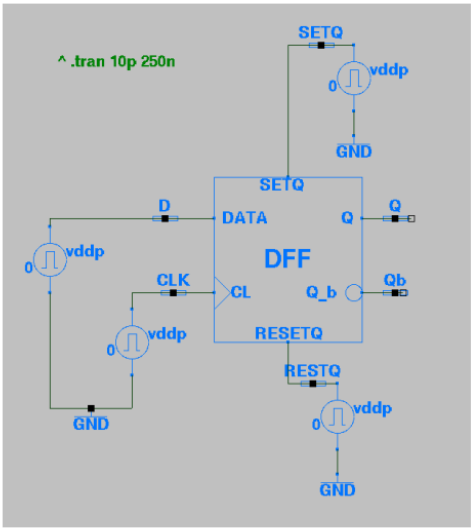
^.tran 10p 250n

c) Press enter. This command changes the total simulation time to 250n (or any other desired simulation time) with **time steps of 10p**.



transmission gate:





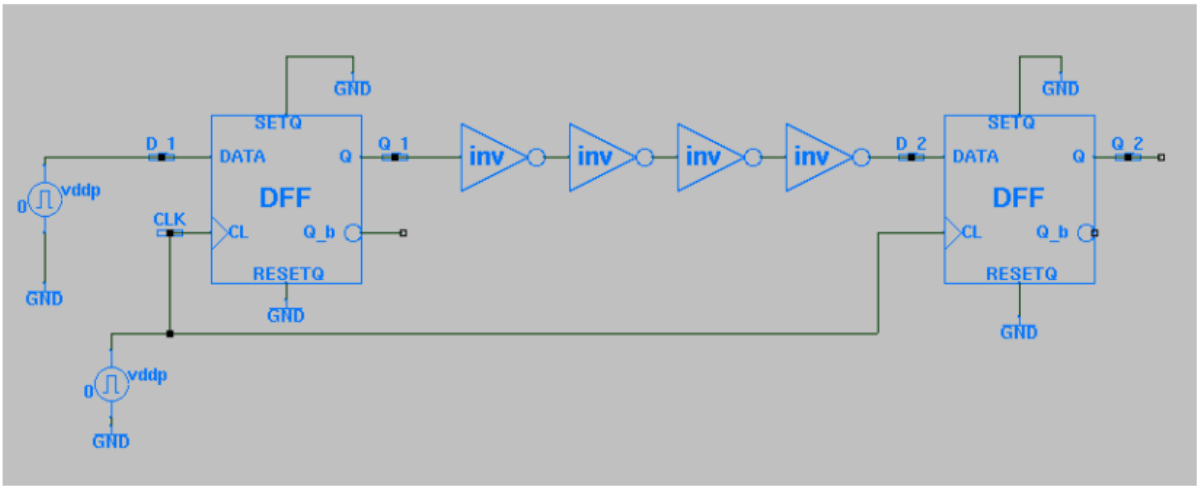


Figure 4: Circuit schematics for part L6.

Appendix:

- Measurement of Clock-to-Q Delay of the D flip-flop:

  To measure Clock to Q Delay of the D- flipflop we can reuse the testbench used for testing the functionality of the D- flipflop. The Clock to Q delay is the delay from rising edge of the clock to rising/falling edge of D flip flop if the data changes, as shown in Figure 5.

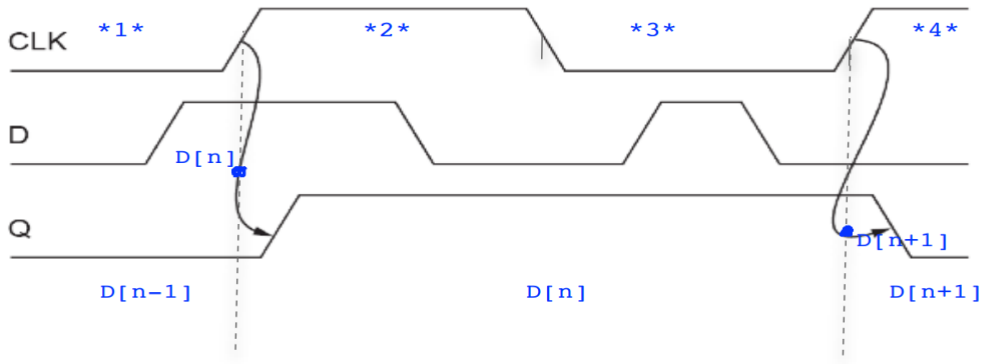


Figure 5: waveforms to define Clock-to-Q delay.

- Measurement of Setup time of the D flip-flop

  Setup time is defined as the minimum amount of time before the clock's active edge that the data must be stable for it to be latched correctly. Any violation may cause incorrect data to be captured, which is known as setup violation.

  In Figure 6, when D = 0 and CLK is LOW, input D is reflected at node Z so that W = 1, Y = 0, and Z = 1 and it will take some time to traverse the path D-W-X-Y-Z. The time that it takes data D to reach node Z is called the SETUP time. When the CLK is HIGH, T1 is switched OFF and T2 is switched ON. Therefore, the LHS 'latching circuit' kicks into action latching the value present at node Z and producing it in the output (Q=0 and Q'= 1).

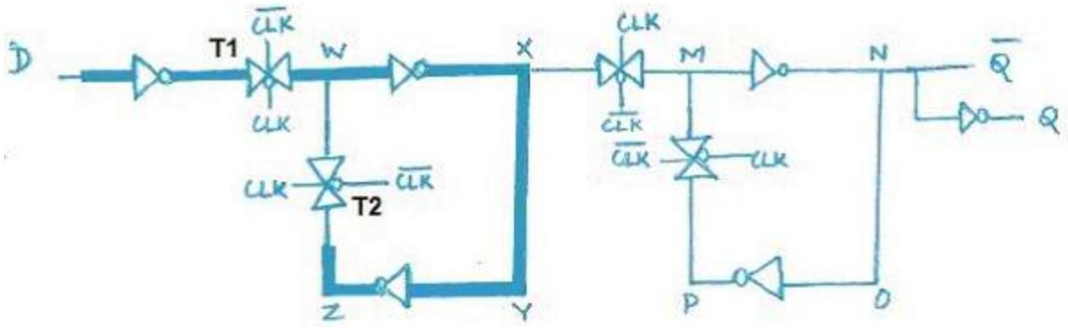


Figure 6: D flip-flop with highlighted nets used to explain setup time.

To see the effect in simulation:

- Set the time of simulation to 2 data periods (40ns in our case) with data flipping (in Figure 8, in my case data is going from high to low as shown in figure below).
- Add the delay in clock pulse, by adjusting the "delay" parameter in the "pulse" element, as show in Figure 7.

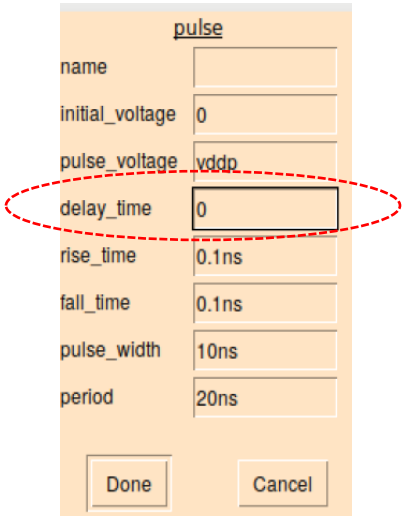


Figure 7: delay parameter highlighted in the "pulse" element configuration box.

- Manually sweep the number till the setup time violation occurs. (Hint: sweep the delay finer to see the effect). As soon as the setup time violates the output will not flip even though input flips, as shown in Figure 8. The setup time can be then defined as the minimum amount of time used in the delay parameter, so a violation occurs.

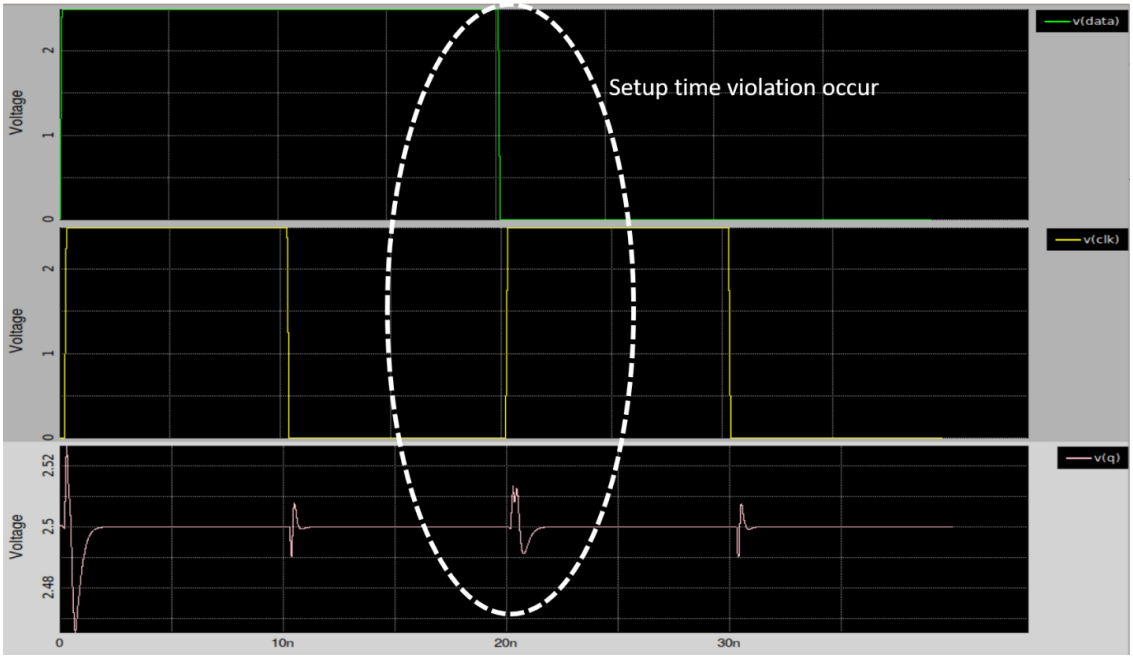


Figure 8: example case when a setup violation occurs.