HW5: VLSI Design (CMPE 222)

Carlos Isaac Espinosa Ramirez

Due Friday March 15th 6:00PM (March 17 6:00AM)
I am using my last 12 hour late period as well as the 24 hour extra time. Thank you!

1 Characterizing the Flip-Flop

Introduction

To characterize the Flip-Flop I have included the proportioned ideal model of Flip-Flop within the ngspice simulation. i.e. I have included the spice circuit in the same spice file as a .subcrkt instance. I did this only for the hold time and setup time characterization.

What is the clock-to-q delay?

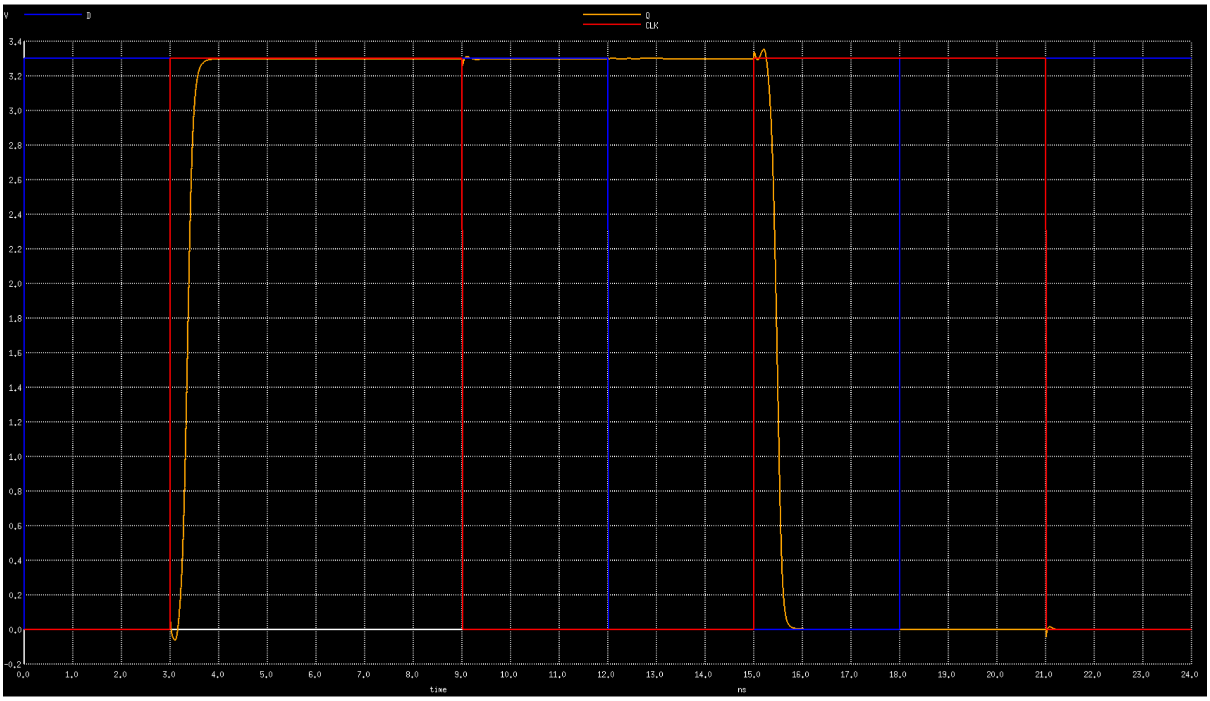
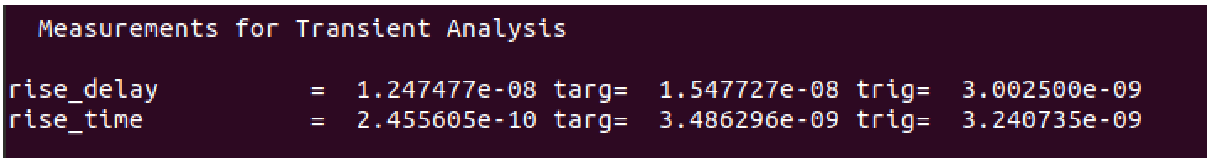


Figure 1: Clock-to-Q delay.

The figure above shows the results for the simulation to characterize the Clock-to-Q which is the time delay between the clock signal and the output Q. The the terminal screenshot bellow shows the results obtained form simulation.



```
 Measurements for Transient Analysis

rise_delay          =  1.247477e-08 targ=  1.547727e-08 trig=  3.002500e-09
rise_time           =  2.455605e-10 targ=  3.486296e-09 trig=  3.240735e-09
```

Figure 2: Clock-to-Q delay simulation results.

What is the hold time?

To characterize the hold time I included the simulation for the ideal model to have a visual appreciation between the implemented model and the ideal. I used the .sp model provided, however i had to modify the file to warp the top circuit into a subcircuit to be able to do the simulation simultaneously. The figure bellow shows the results of the simulation.

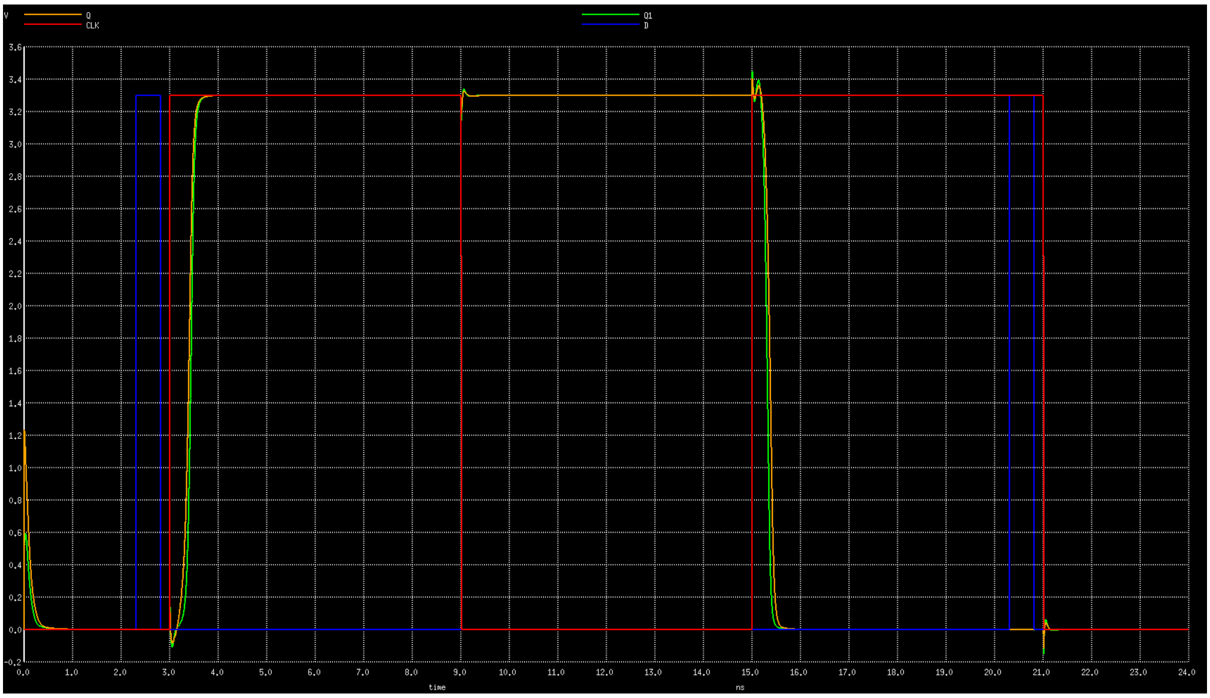
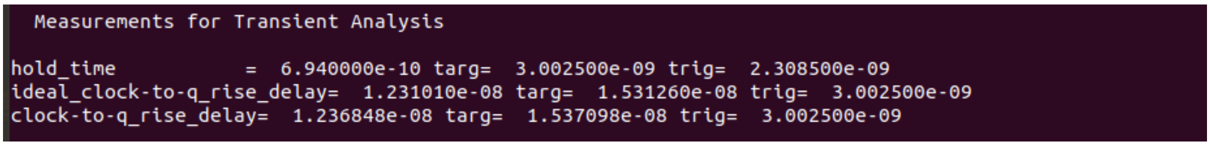


Figure 3: Ideal Flip-Flop and standard cell implementation response to the stimulus.

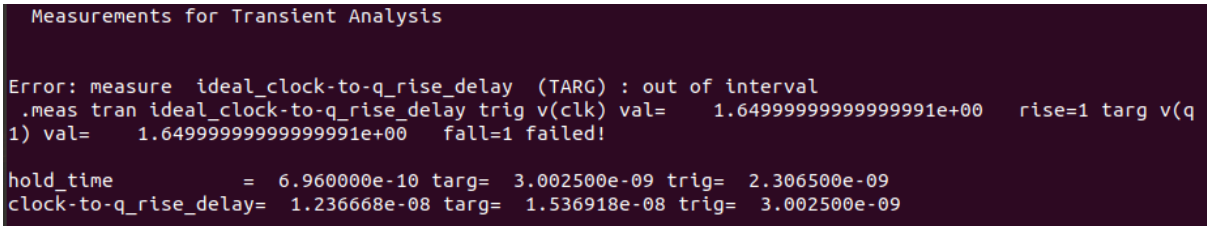
The signal in green is the ideal Flip-Flop response to the stimulus and the signal in yellow is the respose of the CMOS standard cell implementation. One important remark is that my results show a negative hold time. As it was mentioned in Piazza if there are delays to the D input or the CLK this can give the effect of a negative value. As a consequence I did not observed a drastic degradation in delay before the FF fails. The failing behaviour were very sudden, changing from a good performance as the shown in fig. 3 to the one shown in 6 with a very small variation in time. The results of the simulation are shown bellow.



```
 Measurements for Transient Analysis

hold_time           =  6.940000e-10 targ=  3.002500e-09 trig=  2.308500e-09
ideal_clock-to-q_rise_delay=  1.231010e-08 targ=  1.531260e-08 trig=  3.002500e-09
clock-to-q_rise_delay=  1.236848e-08 targ=  1.537098e-08 trig=  3.002500e-09
```

Figure 4: Simulation results for fig. 3.



```
  Measurements for Transient Analysis


Error: measure  ideal_clock-to-q_rise_delay  (TARG) : out of interval
 .meas tran ideal_clock-to-q_rise_delay trig v(clk) val=    1.64999999999999991e+00   rise=1 targ v(q
1) val=    1.64999999999999991e+00   fall=1 failed!

hold_time           =  6.960000e-10 targ=  3.002500e-09 trig=  2.306500e-09
clock-to-q_rise_delay=  1.236668e-08 targ=  1.536918e-08 trig=  3.002500e-09
```

Figure 5: Simulation results for fig. 6.

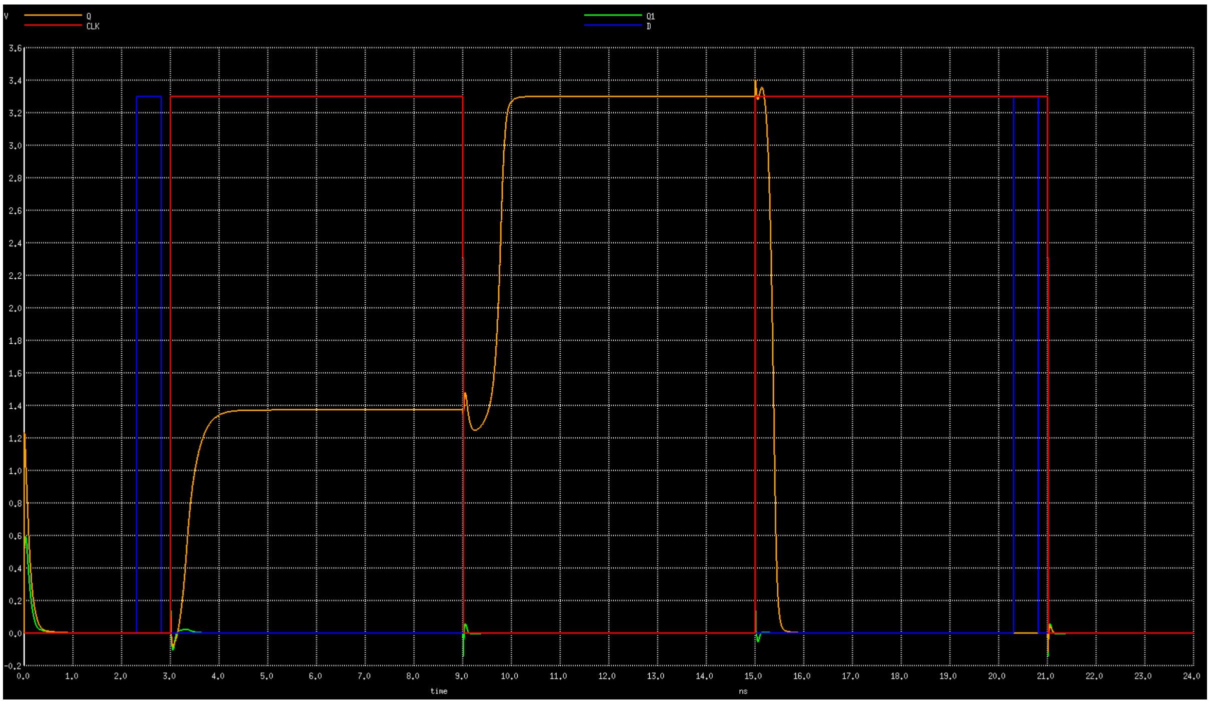
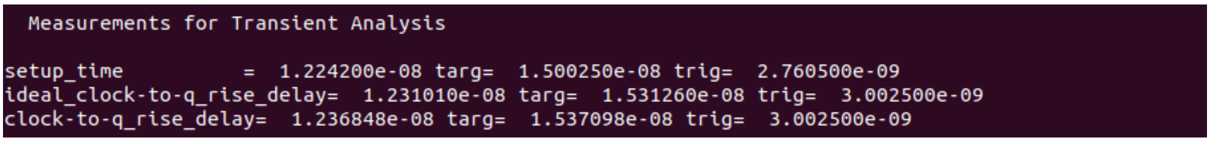


Figure 6: Flip-Flop falling.

What is the setup time?

For the setup time I had the same behaviour as the one described above for the hold time. Very dramatic changes of going from good performance to failing within a 1-2ps variation in time. The simulation results are presented in the following figures.



```
  Measurements for Transient Analysis

setup_time           =  1.224200e-08 targ=  1.500250e-08 trig=  2.760500e-09
ideal_clock-to-q_rise_delay=  1.231010e-08 targ=  1.531260e-08 trig=  3.002500e-09
clock-to-q_rise_delay=  1.236848e-08 targ=  1.537098e-08 trig=  3.002500e-09
```

Figure 7: Simulation results for fig. 8.

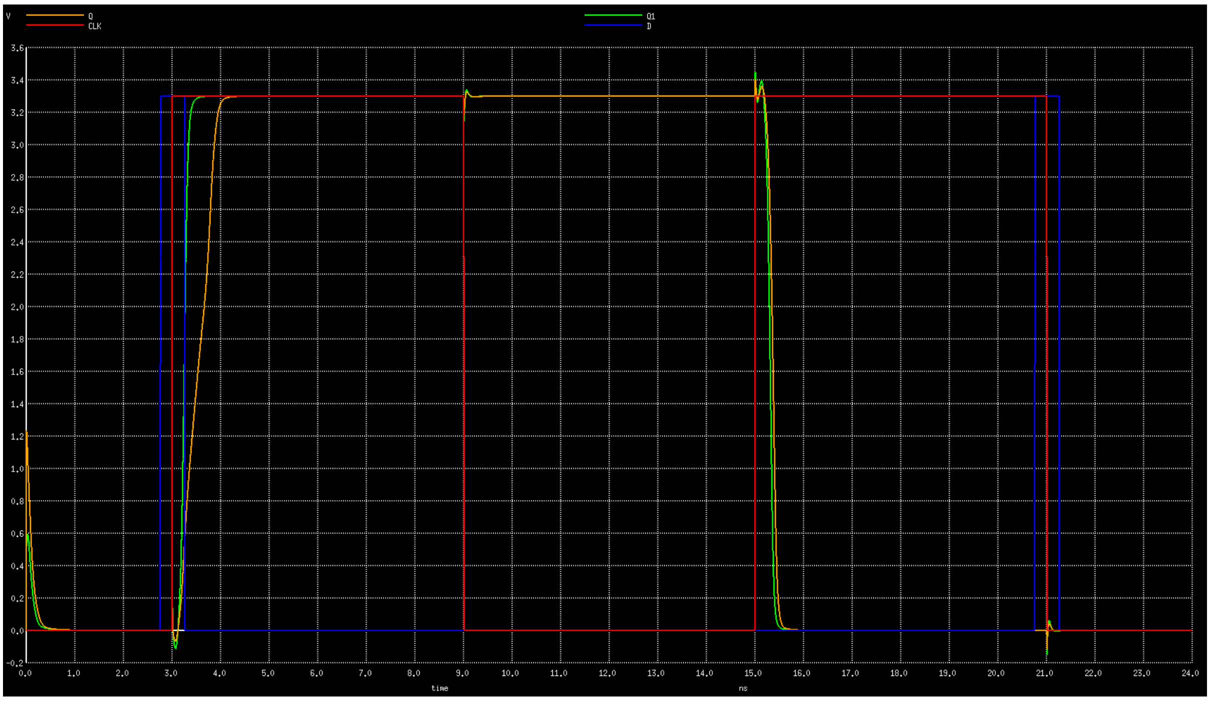


Figure 8: Ideal Flip-Flop and standard cell implementation response to the stimulus.

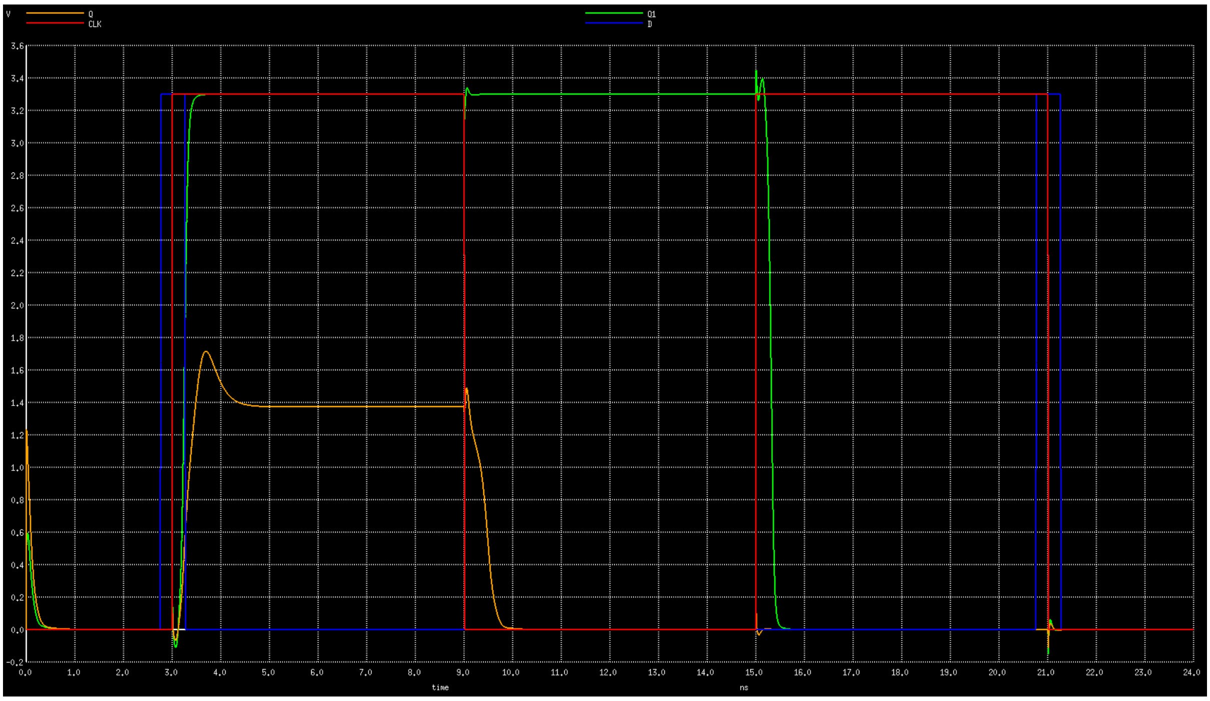
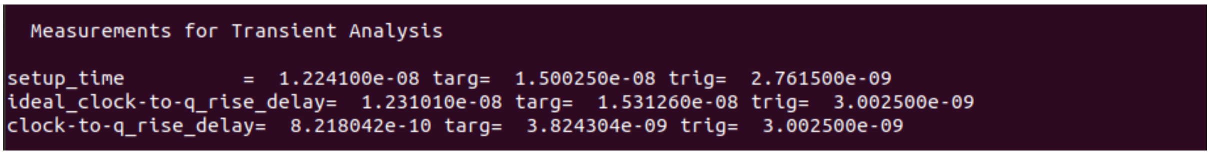


Figure 9: Flip-Flop for setup characterization.



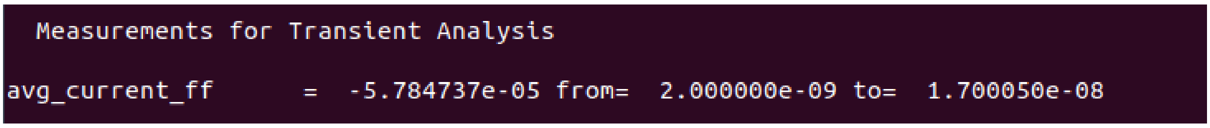
```
  Measurements for Transient Analysis

setup_time          =  1.224100e-08 targ=  1.500250e-08 trig=  2.761500e-09
ideal_clock-to-q_rise_delay=  1.231010e-08 targ=  1.531260e-08 trig=  3.002500e-09
clock-to-q_rise_delay=  8.218042e-10 targ=  3.824304e-09 trig=  3.002500e-09
```

Figure 10: Simulation results for fig. 9.

What is the leakage power of your FF?

The simulation results for the leakage power calculations are shown in fig. 11



```
  Measurements for Transient Analysis

avg_current_ff      =  -5.784737e-05 from=  2.000000e-09 to=  1.700050e-08
```

Figure 11: Simulation results for leakage power calculations.

As the fig.11 shows, the total current from 2n to 17n is: 5.784737×10^{-5}, therefore the power can be obtained as:

$$(3.3\mathrm{V})(5.784737 \times 10^{-5}\mathrm{A}) = 0.0000190896\mathrm{W} = 190896\mathrm{nW}$$

What is the dynamic power of your FF?

To obtain the dynamic power I generated the following stimulus wave.

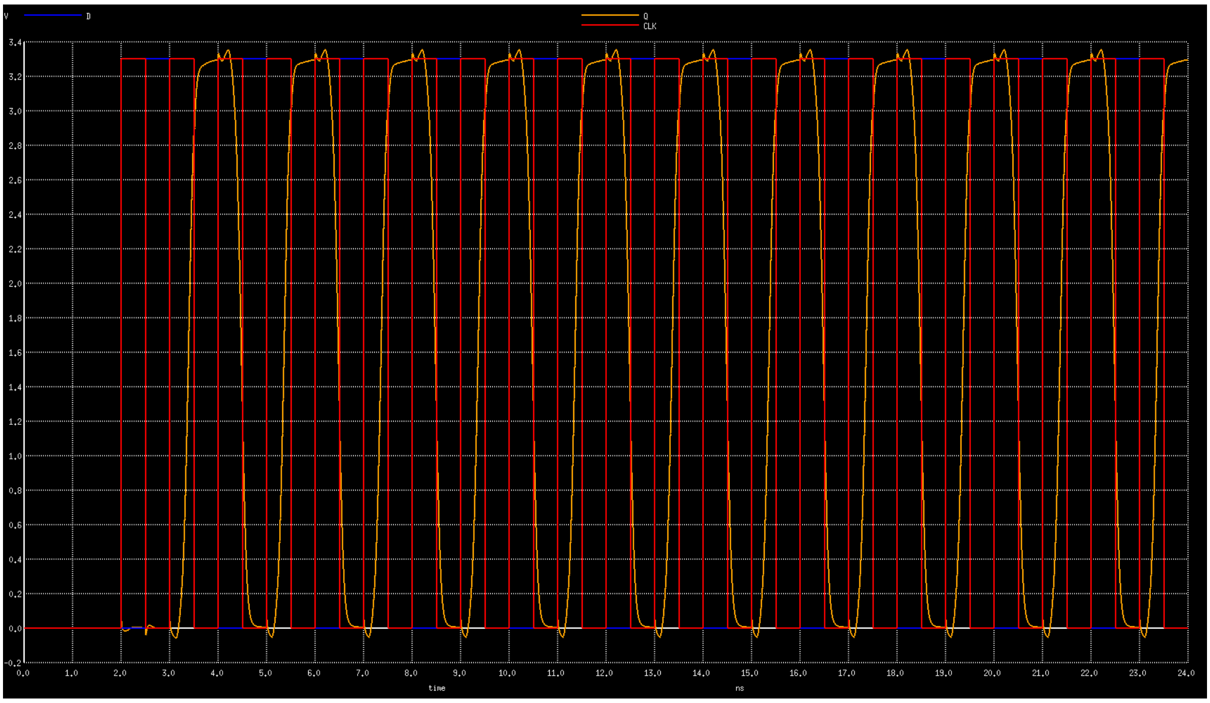
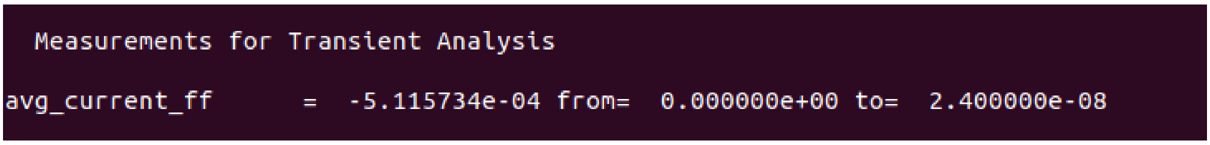


Figure 12: Dynamic power stimulus wave.

The simulation results for the dynamic power calculations are shown in fig. 13



```
  Measurements for Transient Analysis

avg_current_ff      =  -5.115734e-04 from=  0.000000e+00 to=  2.400000e-08
```

Figure 13: Simulation results for dynamic power calculations.

As the fig.13 shows, the total current from 0n to 2.4n is: 5.115734×10^{-4}, therefore the power can be obtained as:

$$(3.3\text{V})(5.115734 \times 10^{-4}\text{A}) = 0.0016688192\text{W} = 16688192\text{nW}$$

What is the clock power of your FF?

(Power when only the clock toggles, not the data.)

To generate the clock power measurement I used the same file as the dynamic power simulation but I set the D signal to do not oscillate (Please notice that this means that a line is commented out in the FF_dynamic.sp file). The results are shown bellow.

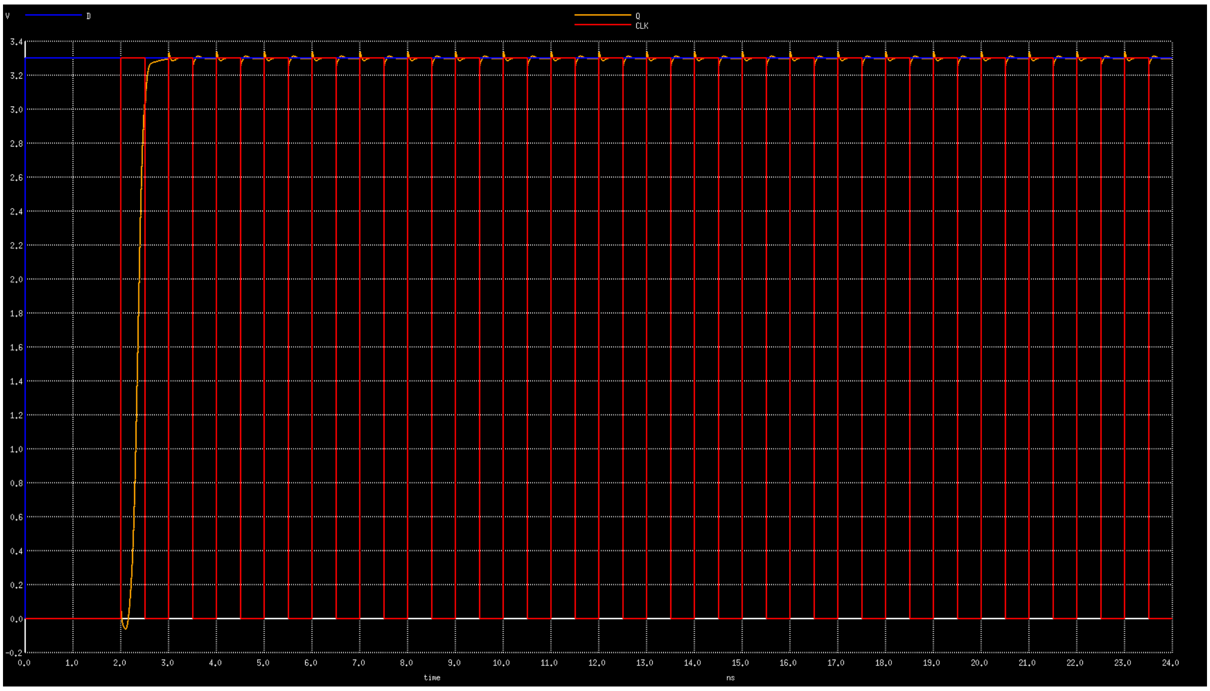
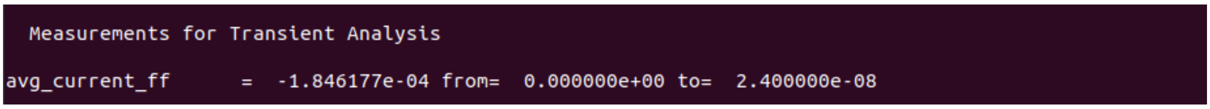


Figure 14: Clock power stimulus wave.

The simulation results for the clock power calculations are shown in fig. 15



```
  Measurements for Transient Analysis

avg_current_ff      =  -1.846177e-04 from=  0.000000e+00 to=  2.400000e-08
```

Figure 15: Simulation results for fig. 14.

As the fig.14 shows, the total current from 0n to 2.4n is: 1.846177×10^{-4}, therefore the power can be obtained as:

$$(3.3\text{V})(1.846177 \times 10^{-4}\text{A}) = 0.000609238\text{W} = 609238\text{nW}$$

2 Logic Style Comparison

Introduction

I decided to use Pseudo-NMOS as my logic family of choice for this assignment. The logic gates I choose are the inverter, the NAND gate and the NOR gate. In the following images I show the magic layout and the circutt diagram.

What are the pros and cons of using this logic style?
(compared to CMOS)

One of the main advantages of this logical family resides in the less amount of devices necessaries to implement a logical funtion. Pseudo-NMOS offer improved speed by removing the PMOS transistors from loading the input. Also pseudo-NMOS have advanges in the implementation of some gates, for example Pseudo-NMOS is a way to build fast wide NOR gates pseudo-NMOS and multiplexers are slightly better than CMOS multiplexers(considering logical effort). However, NAND gates are worse than CMOS NAND gates and since pseudo-NMOS logic consumes power even when not switching, it is best used for critical NOR functions where it shows greatest advantage.

What are the areas of your cells?
(compared to the CMOS cells)

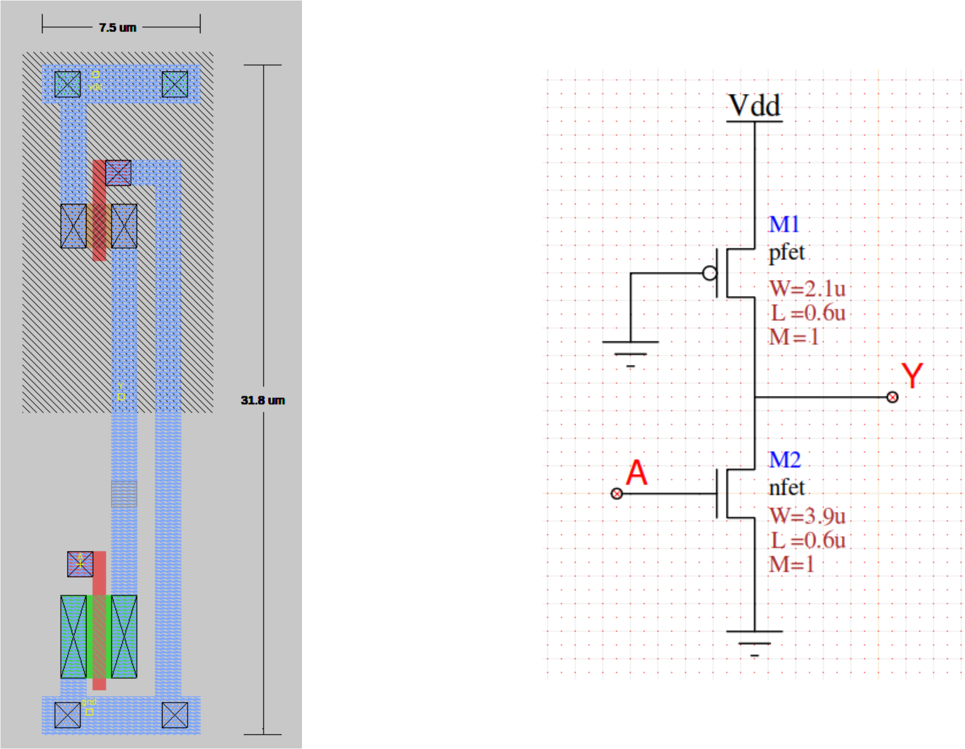


Figure 16: Pseudo-NMOS Inverter with Area=238.5µm^2

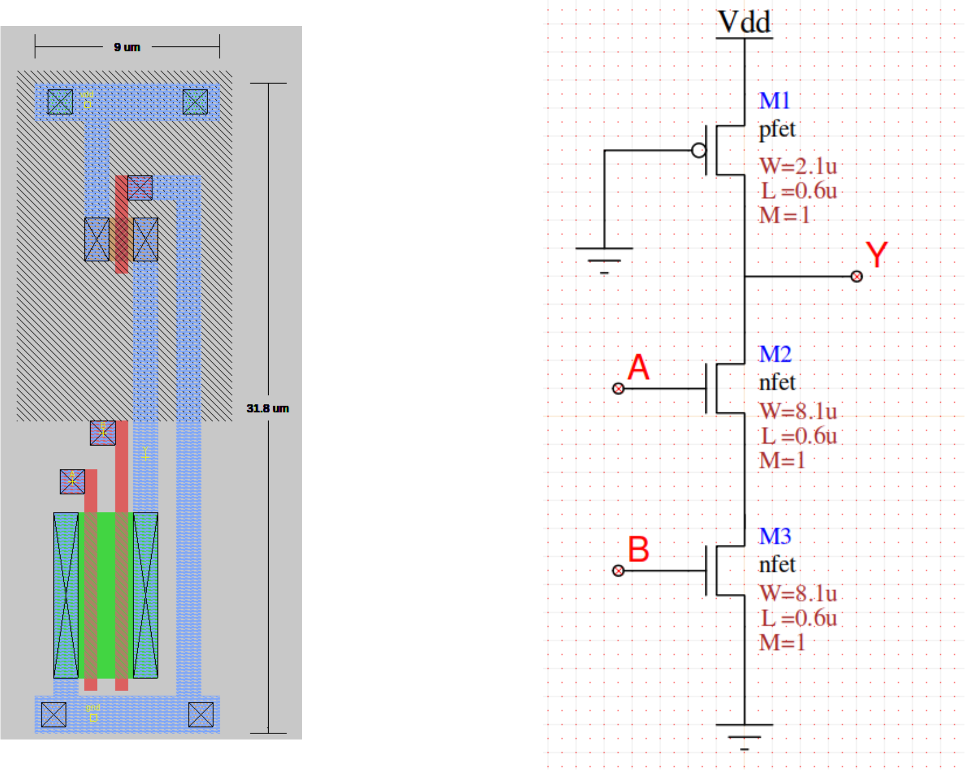


Figure 17: Pseudo-NMOS NAND with Area=286.2µm^2

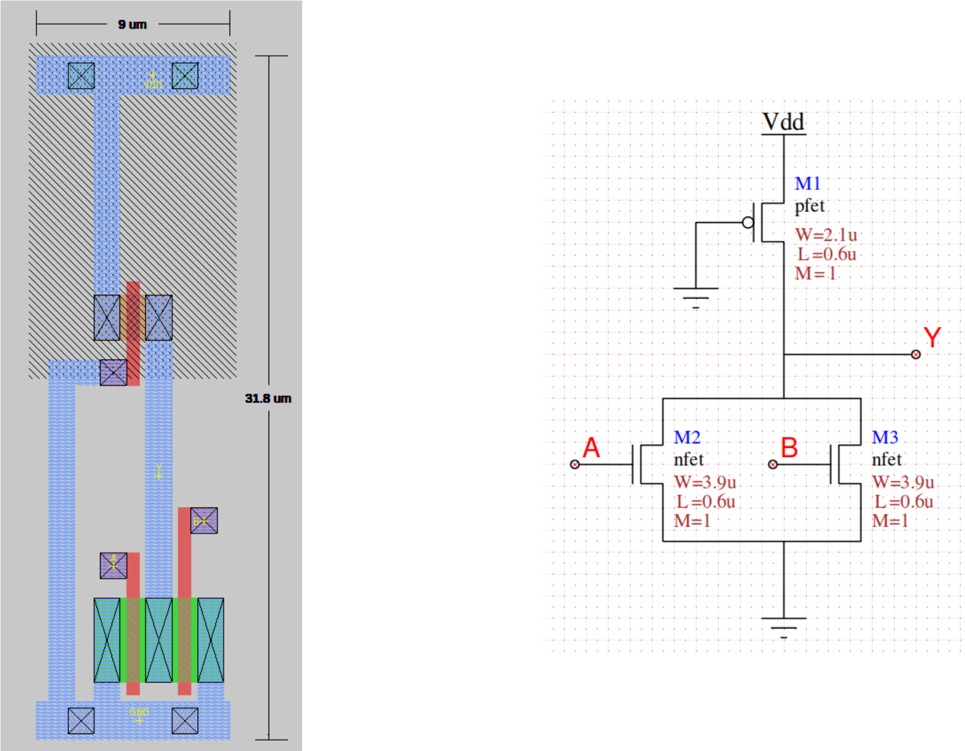


Figure 18: Pseudo-NMOS NOR with Area=286.2µm^2

What are the delays of your cells?
(compared to the CMOS cells)

Now I will present the results of the simulations.

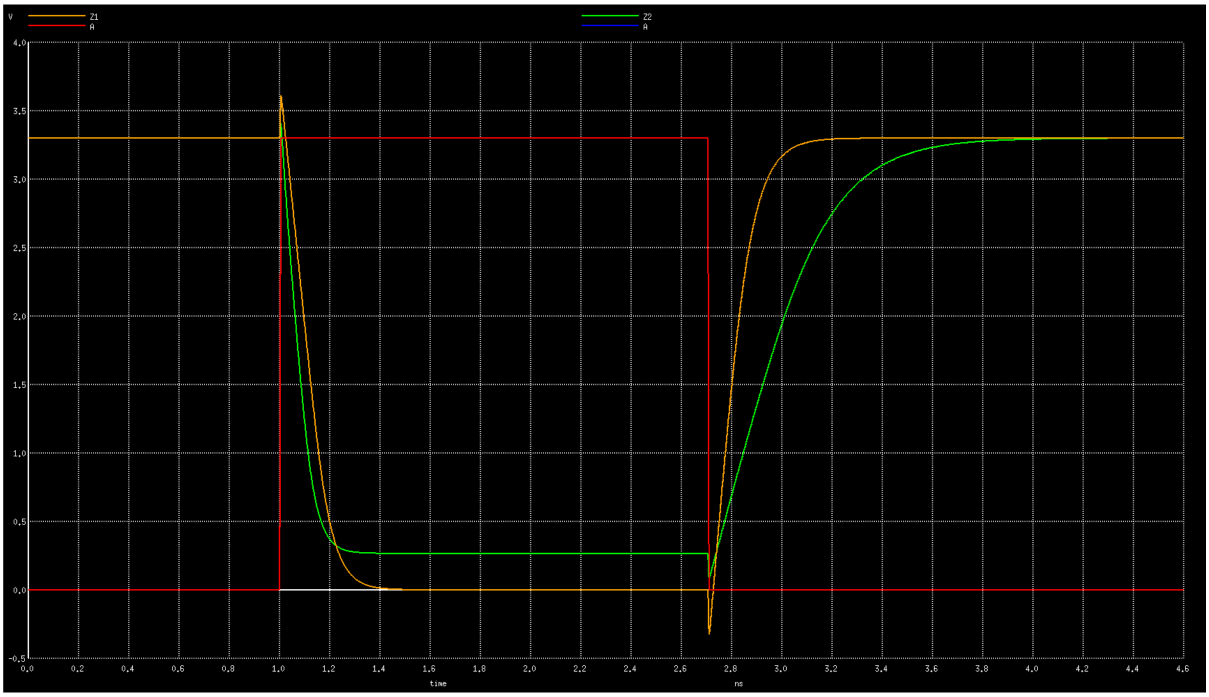
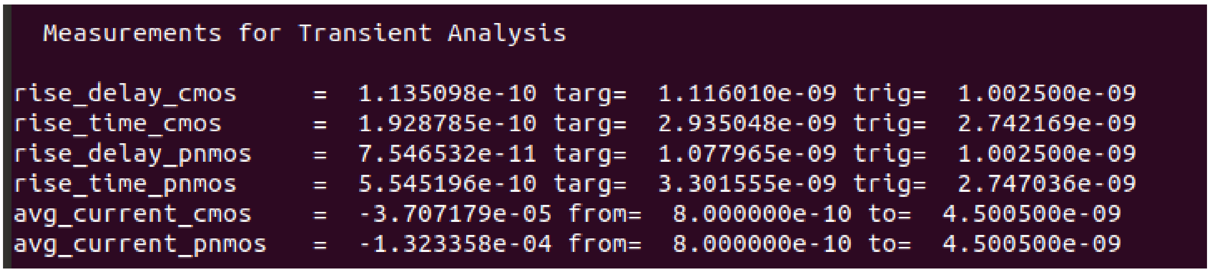


Figure 19: Inverter stimulus wave.

The simulation results for the inverters characterization are shown in fig. 20

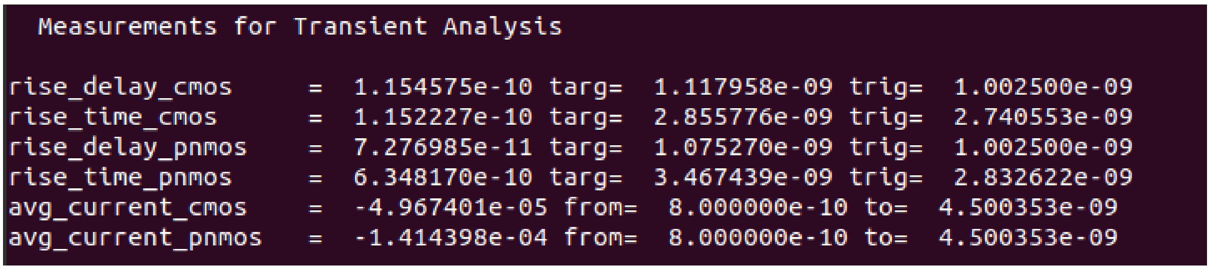


```
 Measurements for Transient Analysis

rise_delay_cmos     =  1.135098e-10 targ=  1.116010e-09 trig=  1.002500e-09
rise_time_cmos      =  1.928785e-10 targ=  2.935048e-09 trig=  2.742169e-09
rise_delay_pnmos    =  7.546532e-11 targ=  1.077965e-09 trig=  1.002500e-09
rise_time_pnmos     =  5.545196e-10 targ=  3.301555e-09 trig=  2.747036e-09
avg_current_cmos    =  -3.707179e-05 from=  8.000000e-10 to=  4.500500e-09
avg_current_pnmos   =  -1.323358e-04 from=  8.000000e-10 to=  4.500500e-09
```

Figure 20: Simulation results for fig. 19.

The simulation results for the NAND gates characterization are shown in fig. 21



```
 Measurements for Transient Analysis

rise_delay_cmos     =  1.154575e-10 targ=  1.117958e-09 trig=  1.002500e-09
rise_time_cmos      =  1.152227e-10 targ=  2.855776e-09 trig=  2.740553e-09
rise_delay_pnmos    =  7.276985e-11 targ=  1.075270e-09 trig=  1.002500e-09
rise_time_pnmos     =  6.348170e-10 targ=  3.467439e-09 trig=  2.832622e-09
avg_current_cmos    =  -4.967401e-05 from=  8.000000e-10 to=  4.500353e-09
avg_current_pnmos   =  -1.414398e-04 from=  8.000000e-10 to=  4.500353e-09
```

Figure 21: Simulation results for fig. 22.

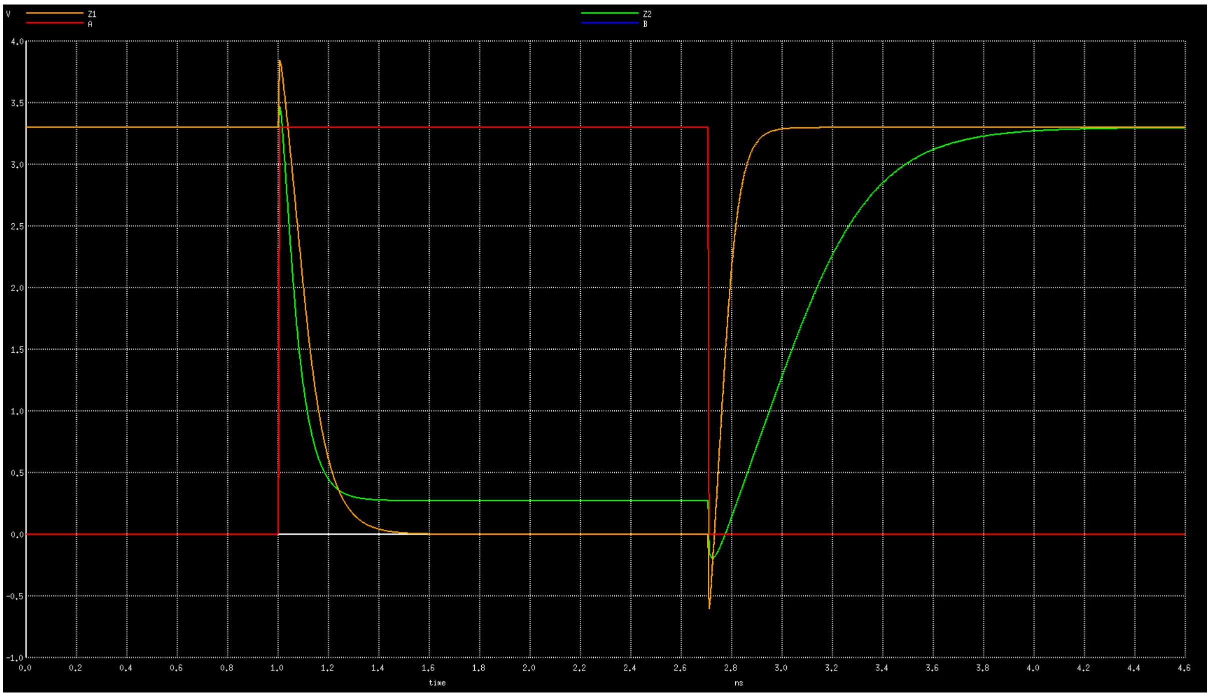


Figure 22: NAND stimulus wave.

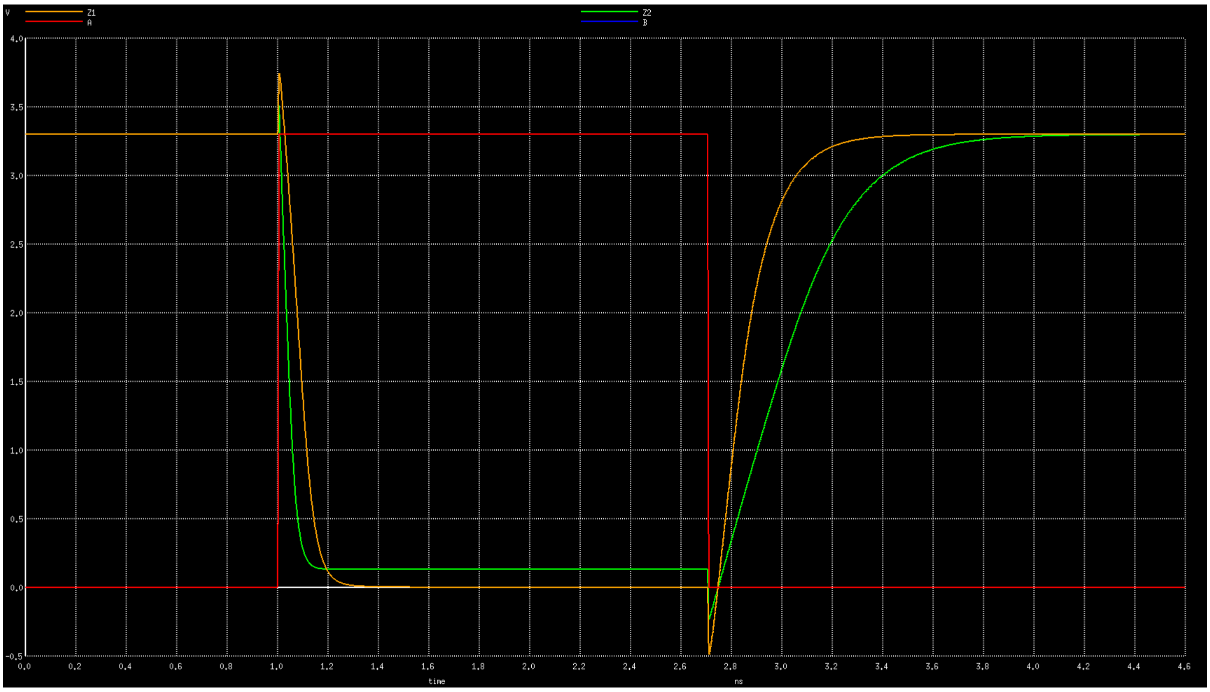
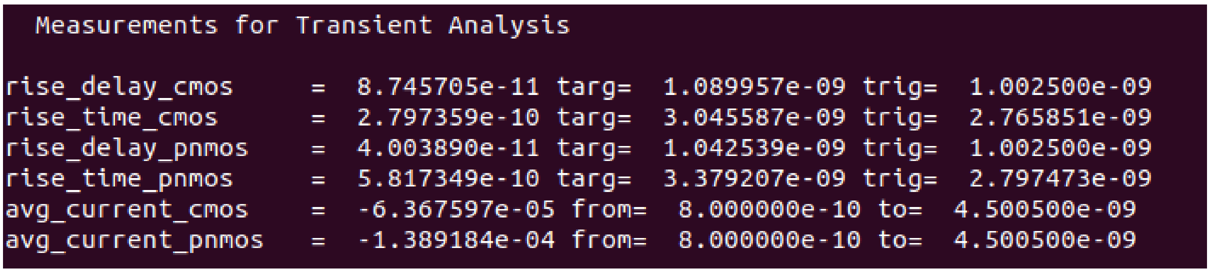


Figure 23: NOR stimulus wave.



```
  Measurements for Transient Analysis

rise_delay_cmos     =  8.745705e-11 targ=  1.089957e-09 trig=  1.002500e-09
rise_time_cmos      =  2.797359e-10 targ=  3.045587e-09 trig=  2.765851e-09
rise_delay_pnmos    =  4.003890e-11 targ=  1.042539e-09 trig=  1.002500e-09
rise_time_pnmos     =  5.817349e-10 targ=  3.379207e-09 trig=  2.797473e-09
avg_current_cmos    =  -6.367597e-05 from=  8.000000e-10 to=  4.500500e-09
avg_current_pnmos   =  -1.389184e-04 from=  8.000000e-10 to=  4.500500e-09
```

Figure 24: Simulation results for fig. 23.

What are the leakage/dynamic powers of your cells?
(compared to the CMOS cells)

Leakage power for this family is worse as it consumes power even when not switching.

As the fig.20 shows, the total current for the CMOS device is: 3.707179×10^{-5}, therefore the power can be obtained as:

$$(3.3\text{V})(3.707179 \times 10^{-5}\text{A}) = 0.000122337\text{W} = 122337\text{nW}$$

As the fig.20 shows, the total current for the Pseudo-NMOS device is: 1.323358×10^{-4}, therefore the power can be obtained as:

$$(3.3\text{V})(1.323358 \times 10^{-4}\text{A}) = 0.000436708\text{W} = 436708\text{nW}$$

As the fig.21 shows, the total current for the CMOS device is: 4.967401×10^{-5}, therefore the power can be obtained as:

$$(3.3\text{V})(4.967401 \times 10^{-5}\text{A}) = 0.000163924\text{W} = 163924\text{nW}$$

As the fig.21 shows, the total current for the Pseudo-NMOS device is: 1.414398×10^{-4}, therefore the power can be obtained as:

$$(3.3\text{V})(1.414398 \times 10^{-4}\text{A}) = 0.000466751\text{W} = 466751\text{nW}$$

As the fig.24 shows, the total current for the CMOS device is: 6.367597×10^{-4}, therefore the power can be obtained as:

$$(3.3\text{V})(6.367597 \times 10^{-5}\text{A}) = 0.000210131\text{W} = 210131\text{nW}$$

As the fig.24 shows, the total current for the Pseudo-NMOS device is: 1.389184×10^{-5}, therefore the power can be obtained as:

$$(3.3\text{V})(1.389184 \times 10^{-4}\text{A}) = 0.000458431\text{W} = 458431\text{nW}$$

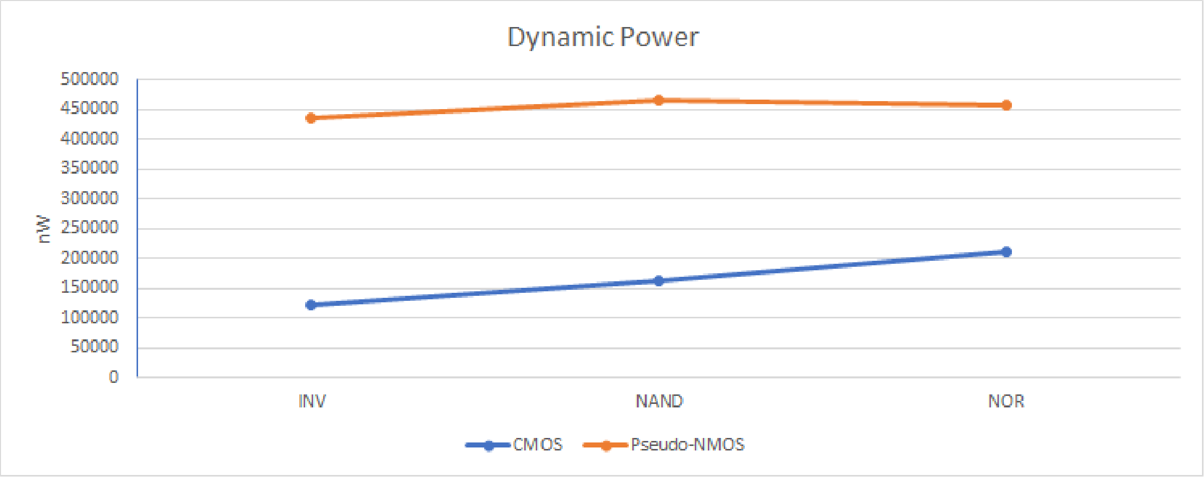


Figure 25: Cells dynamic power.

What assumptions did you need to make?
(This should make the comparison as fair as possible!)

To perform my comparison I used logical effort to size the transistor in order to normalize the logical efforts so that an inverter in the Pseudo-NMOS technology has an average logical effort of 1. Therefore the sizing of the gates ended up to be as follows:

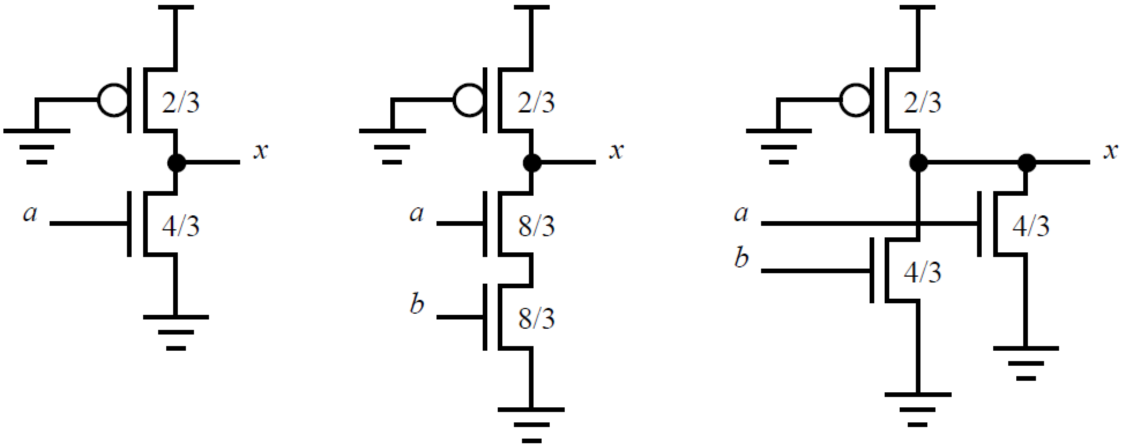


Figure 26: Pseudo-NMOS inverter, NAND and NOR gates sizing.