ECE 304

Lab #3: Design & Analysis of True-Single-Phase-Clock Flip-Flop

Abstract:

In this lab we first develop an inverted True-Single-Phase-Clock (TSPC) Flip-Flop with optimal sizing to minimize delay. The first step is to scale the transistors of each stage such that they will have equal pull-up and pull-down strength for their relative ratios (**Figure 2**). The scale of these transistors will then be multiplied by constants that will be resolved through logical effort analysis providing the smallest possible delays (**Figure 4**). After testing the functionality of it, "process corners" are used to model our circuit under different conditions to simulate varying properties that appear in an imperfect fabrication process.

These process corners include different modelling variants and there is also a range of temperatures that affect the transistor speeds; all of these different factors were simulated. Using the calculator tool, we found the propagation rising and falling delays (**Figure 11 & 14**). Finally, by using parametric sweeps to find the most efficient delay values, the setup and hold times of our normal conditions were calculated to be 0.003ns and 0.004ns, respectively (**Figure 18 & 20**).

Design:

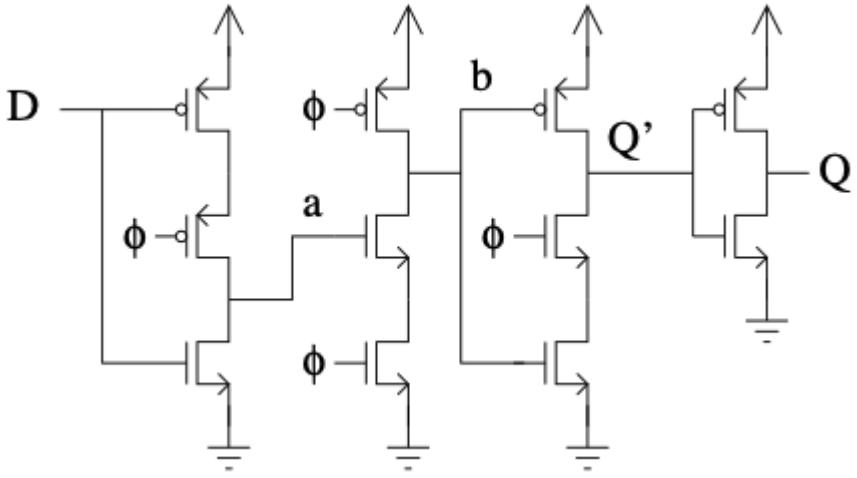

Figure 1 - TSPC flip-flop with inverter added

Since the mobility of holes in PMOS is twice that of electrons in NMOS ($\mu_n = 2\mu_p$), we must scale the transistors such that the pull-up and pull-down networks of each stage consist of equal strength. This is achieved by equalizing the resistances between them. Knowing that resistances of the transistors scales inversely with size, base PMOS resistance is 2R, and base NMOS resistance is R, the ratios below result in each stage having a resistance R.

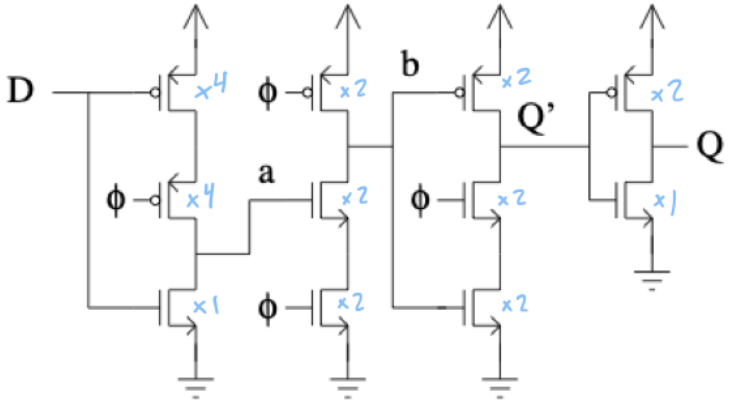

Figure 2 - Scaled TSPC gates

The actual size of the gates will be determined by working backwards through logical effort calculations that minimize delay. The size of each gate can be seen as a variable that is the total input capacitance of that gate relative to $C_{min.}$ The relative ratios for the transistors of each stage will be multiplied by the constants found in logical effort analysis.

The minimal delay results from each stage having equivalent effort delay ‘f’. Therefore the optimum effort delay of each stage is found from

$$f = F^{1/N} \tag{1}$$

where ‘F’ is the overall path effort. As shown by the critical path below we can see that ‘N’ - the number of stages, is 3.

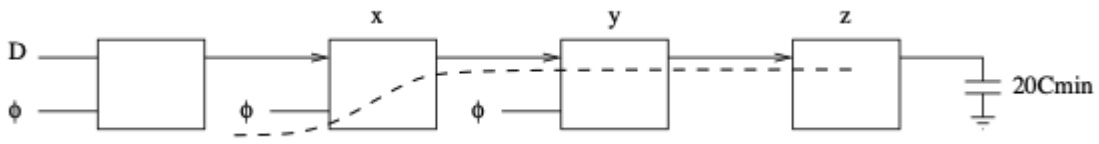

Figure 3 - TSPC flip-flop critical path

Overall path effort is given by

$$F = GHB \tag{2}$$

where G is the path logical effort, H is the path electrical effort, and B is the path branching effort.

Each of these is calculated as follows:

$$G = \prod_i g_i$$
$$H = \frac{C_{out,path}}{C_{in,path}}$$
$$B = \prod_i b_i$$

First we can calculate the logical effort for each stage by

$$g = \frac{C_{in,gate}}{C_{in,inv}} = \frac{The\ input\ capacitance\ of\ the\ gate}{Input\ capacitance\ of\ an\ inverter\ that\ delivers\ the\ same\ output\ current}$$

The input capacitance of the first stage in the critical path is 4C, second stage is 4C, and third stage is 3C. The input capacitance of an inverter delivering the same output current is 3C for all stages.

The final path logical effort is given by

$$G = g_1 g_2 g_3 = (4/3)(4/3)(3/3) = 16/9$$

Since the output load capacitance is assumed to be 20C, and the input capacitance of the first gate in the critical path is 4C, the final path electrical effort is given by

$$H = 20C/4C = 5$$

Finally, since each stage only has a single branch, the path branching effort is given by

$$B = 1 * 1 * 1 = 1$$

Now using (2), the overall path effort is

$$F = (16/9) * 5 * 1 = 8.89$$

And using (1), the effort delay for each stage is

$$f = 8.89^{1/3} = 2.07$$

Now we can solve for the constant that each stage's input capacitance must be multiplied by in order to satisfy the effort delay. Starting from the final stage and working backwards:

$$f_z = g_z b_z h_z = 2.07 = 1 * 1 * (20C/3ZC) => Z = 3.22$$

$$C_{in-z} = C_{out-y} = 3 * 3.22 = 9.66$$

$$f_y = g_y b_y h_y = 2.07 = (4/3) * 1 * (9.66C/4YC) => Y = 1.55$$

$$C_{in-y} = C_{out-x} = 4 * 1.55 = 6.22$$

$$f_x = g_x b_x h_x = 2.07 = (4/3) * 1 * (6.22C/4XC) => X = 1$$

Multiplying these constants by the transistor size ratios shown in **Figure 2**, gives us the following final sizes used to build our circuit.

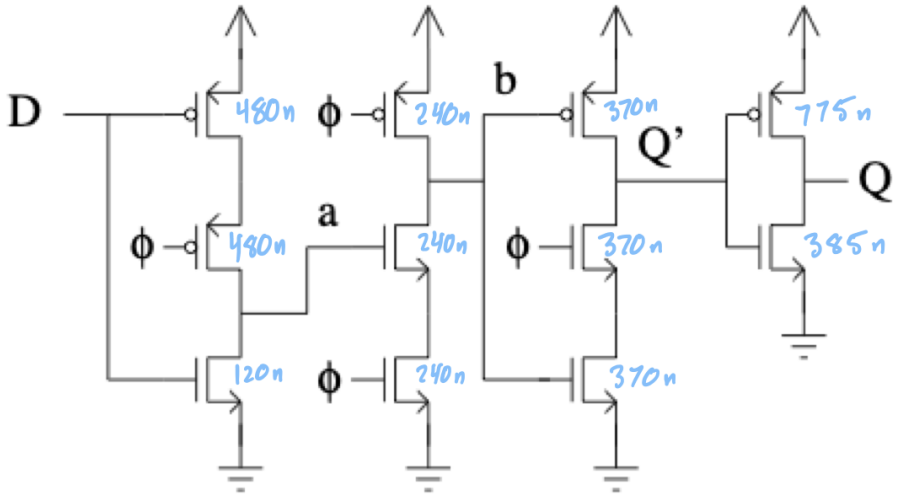

Figure 4 - TSPC min delay transistor sizing

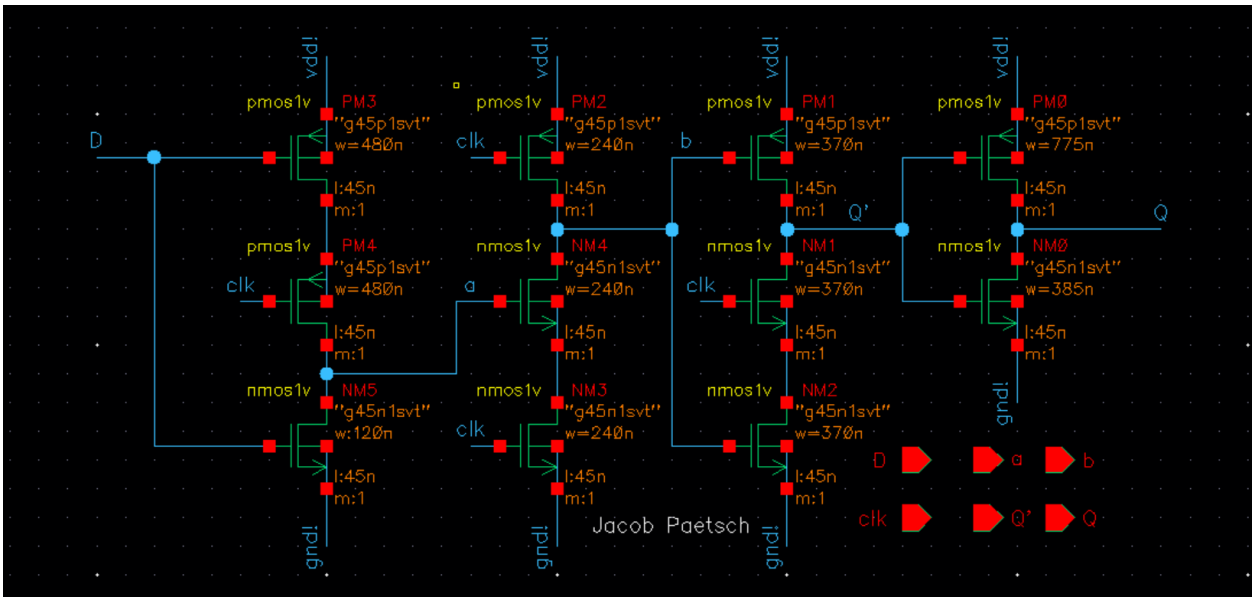

Figure 5 - TSPC Flip Flop Circuit

This TSPC Flip Flop implementation has sized transistors and includes internal signals (a, b, and Q'), inputs (D and clk), and final output signal Q. Note that the Virtuoso software automatically rounds the NMOS and PMOS width to the nearest 5 nm.

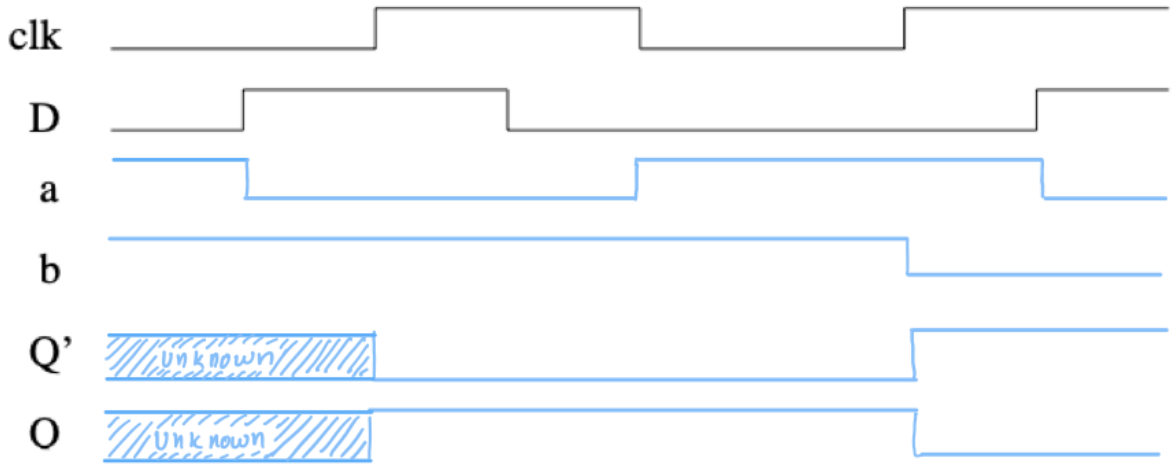

Figure 6 - Expected TSPC Flip Flop Timing Behaviour

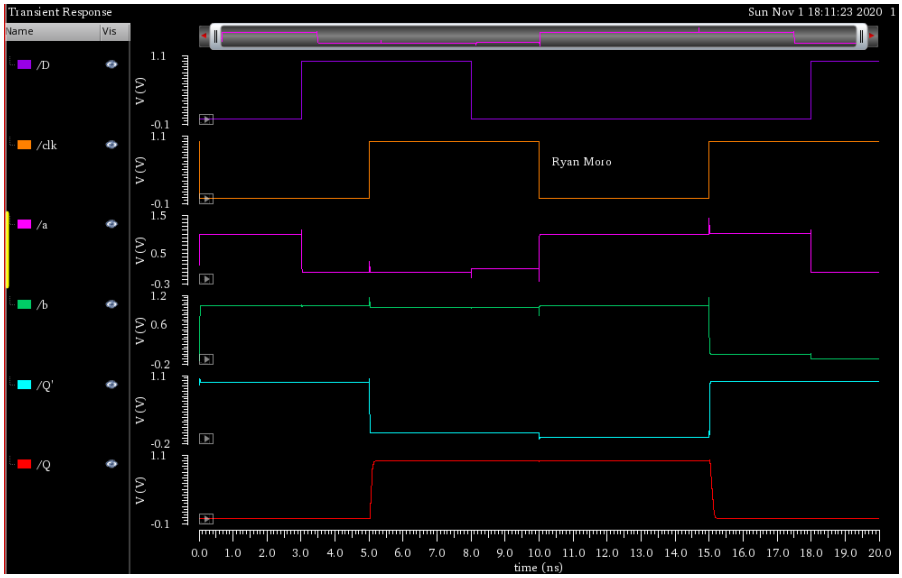

Figure 7 - TSPC Flip Flop Testbench Plotted Signals

Based on a testbench schematic set up using our TSPC Flip Flop circuit, we can see that the graph of the outputs is as expected. In reality, there are slight voltage dips for certain clock/input combinations as the capacitors in the circuit experience some leakage and other non-

ideal transistor behaviour. In addition, there are small propagation timing delays for capacitor charge up and discharge times.

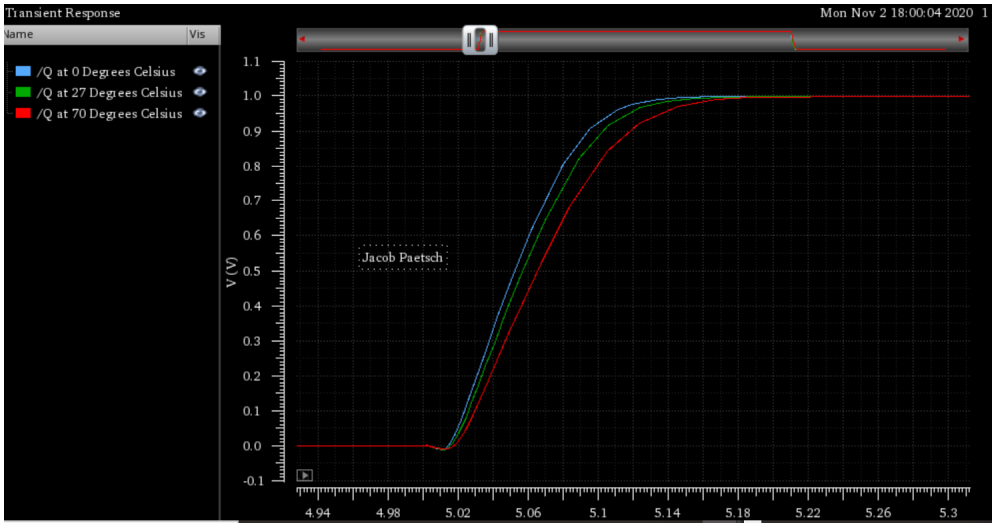

Figure 8 - TSPC Flip Flop Q Temperature Variation Response

Temperature variation can have a significant effect on the output signal and create variation, especially at more extreme temperatures (see the difference in rising edge of the plotted output signal Q as temperature increases).

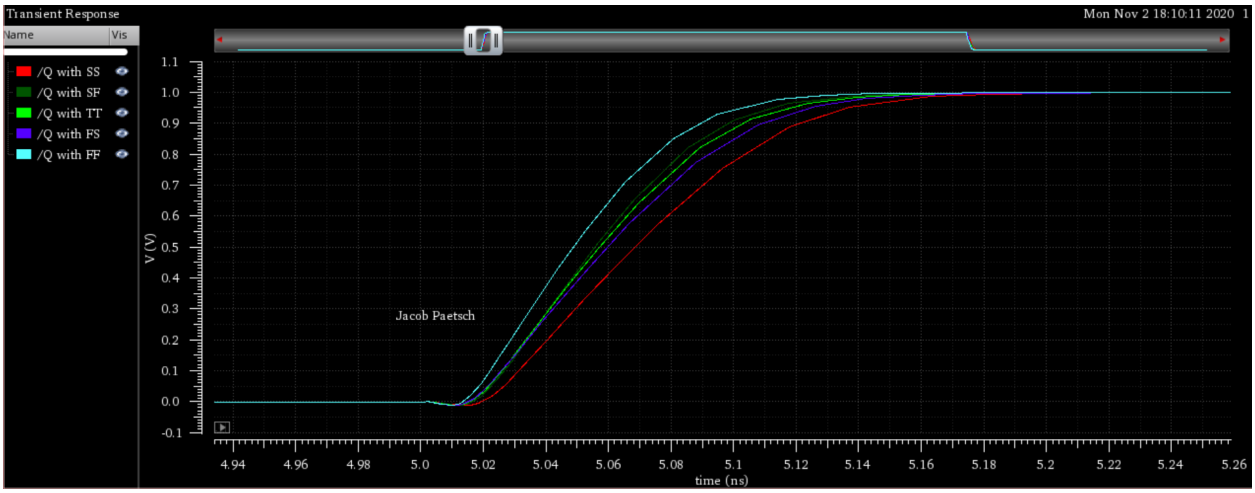

Figure 9 - TSPC Flip Flop Q Process Variation Response

The gdpk045 library kit has preset process corners that have an effect on our Q output signal as evidenced by the distinct rising edges. Each of these modes represents a different response time configuration for our circuit's NMOS and PMOS transistors as shown below.

Corner	NMOS	PMOS
SS	slow	slow
SF	slow	fast
TT	typical	typical
FS	fast	slow
FF	fast	fast

Figure 10 - Process corner effect on transistors

Test	Output	Nominal	Spec	Weight	Pass/Fail	Min	Max	Temp-0	Temp-27	Temp-70
lab3:tspcflipflop_tb:1	/Q									
lab3:tspcflipflop_tb:1	Tprop0-1	55.47p				50.87p	63.65p	50.87p	55.47p	63.65p
lab3:tspcflipflop_tb:1	Tprop1-0	87.93p				79.52p	102.4p	79.52p	87.93p	102.4p

Figure 11 - Calculator Temperature Rising and Falling Delays

Based on the Tprop0-1 (rising delay) and Tprop1-0 (falling delay) expressions that were set up, we can see our times for temperature corners comparison: For 0 degrees we have a 50.87ps rise & 79.52ps fall, for 27 degrees we have a 55.47ps rise & 87.93ps fall, and for 70 degrees 63.65ps rise & 102.4ps fall. See the temperature plots below in **Figure 12 & 13**.

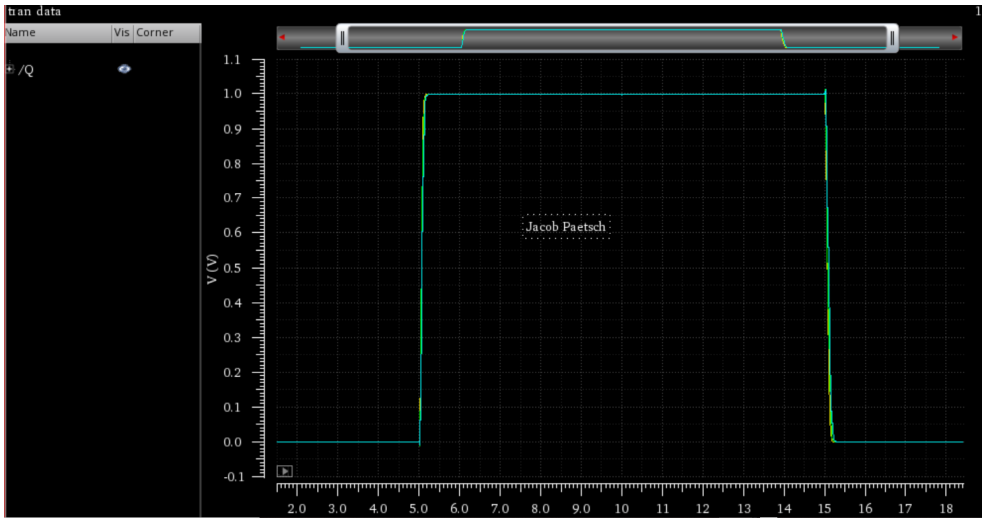

Figure 12 - Calculator Temperature Corners Plot

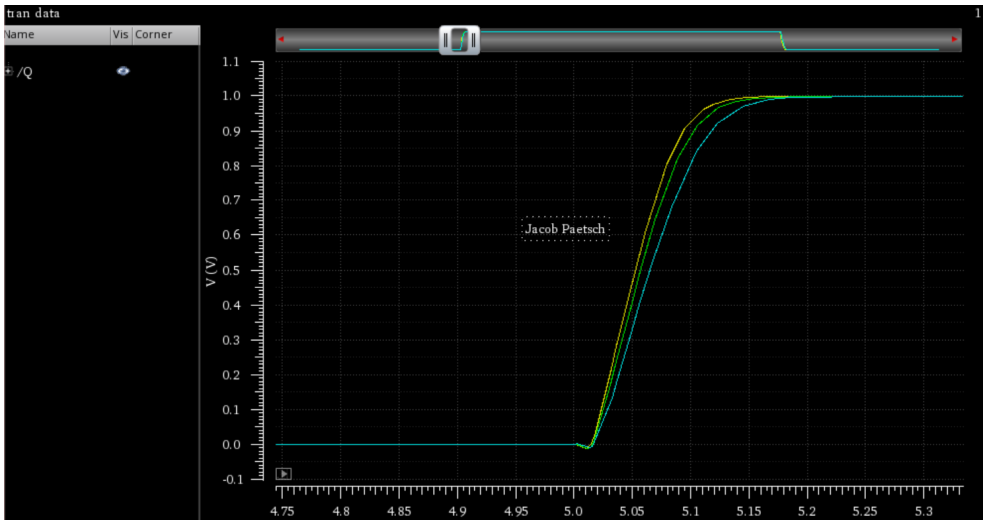

Figure 13 - Calculator Temperature Corners Plot Zoomed-In

t	Output	Nominal	Spec	Weight	Pass/Fail	Min	Max	SS	SF	TT	FS	FF
flop_tb:1	/Q											
flop_tb:1	Tprop0-1	55.47p				47.31p	66.79p	66.79p	54.22p	55.47p	58.14p	47.31p
flop_tb:1	Tprop1-0	87.93p				74.58p	107.3p	107.3p	94.86p	87.93p	83.18p	74.58p

Figure 14 - Calculator Process Corners Rising and Falling Delays

Based on the Tprop0-1 (rising delay) and Tprop1-0 (falling delay) expressions that were set up and the same process as before, we can see our times for process corners comparison: For mode SS we have a 66.79ps rise & 107.3ps fall, for mode SF we have a 54.22ps rise & 94.86ps fall, for mode TT we have a 55.47ps rise and 87.93ps fall, for mode FS we have a 58.14ps rise and 83.18ps fall, and for mode FF we have a 47.31ps rise and 74.58ps fall. See the process variation plots below in **Figure 15 & 16**.

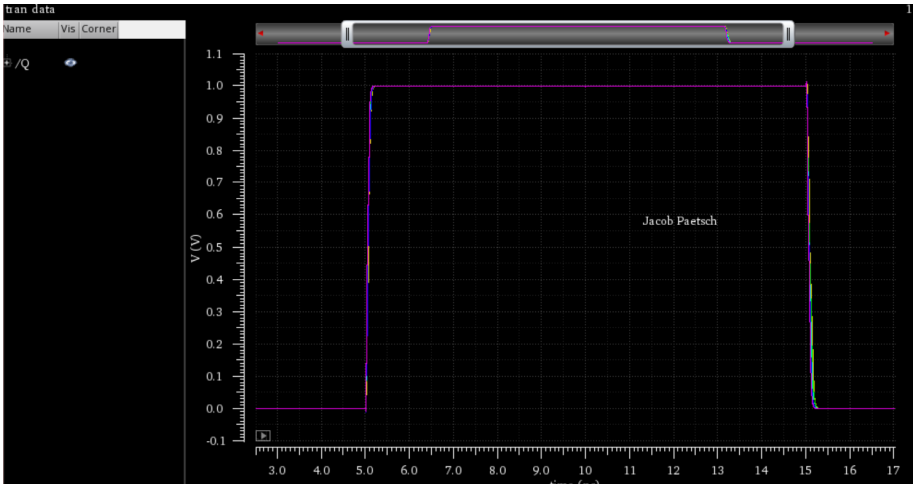

Figure 15 - Calculator Process Corners Plot

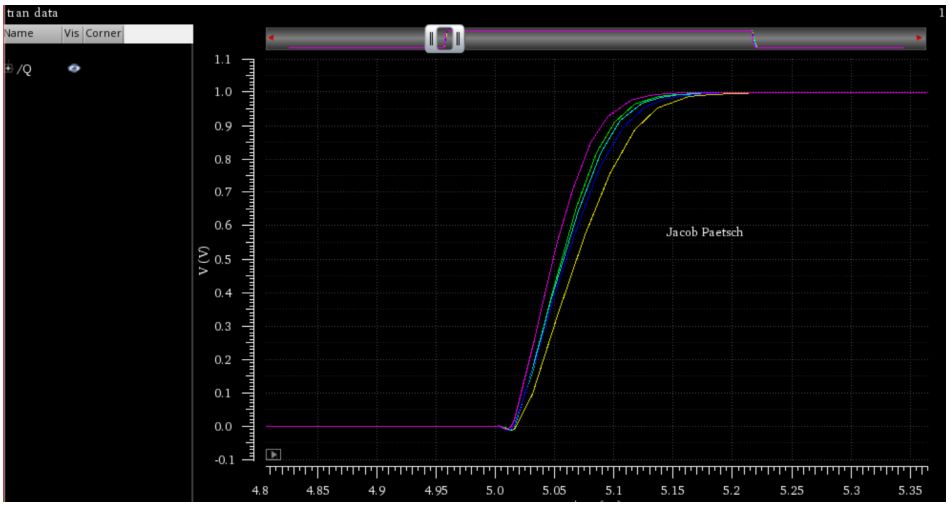

Figure 16 - Calculator Process Corners Plot Zoomed-In

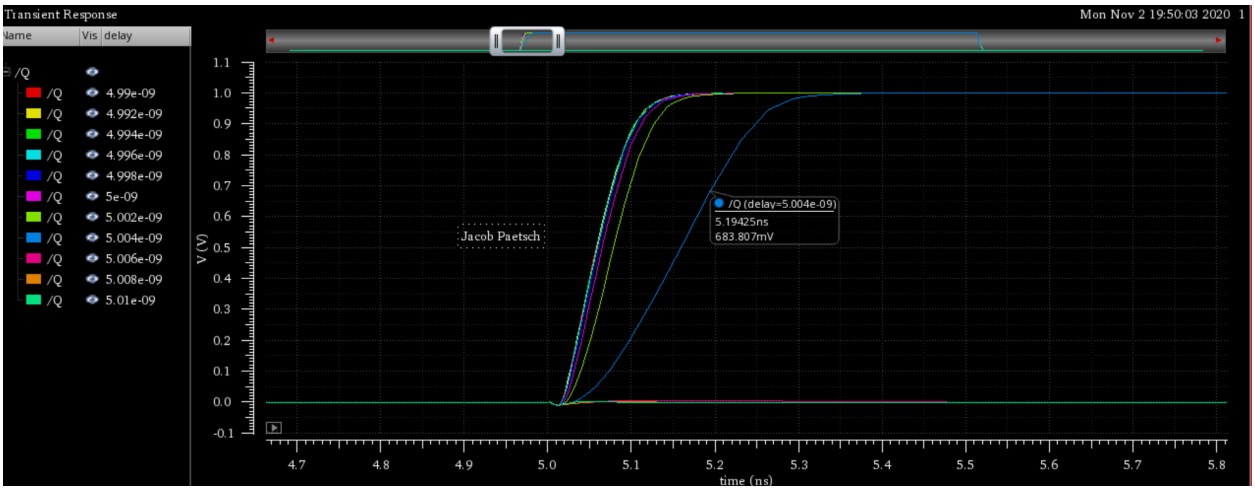

Figure 17 - Parametric Highest Delay Value Plot (For Setup Time)

By using Cadence's parametric analysis tool, we can test a range of delays (11 different values from 4.99ns to 5.01ns) to find the maximum value that still toggles the output to 1. As seen in the plot (although it is a bit of an outlier from the others), a maximum allowable delay value of 5.004ns is obtained; we will then use this as our delay variable value. In a different figure (with horizontal markers), we then obtain the setup time by checking the first clock rising edge against the first input rising edge. The setup time is calculated to be (5.0045ns - 5.0015ns) = 0.003ns (**Figure 18**).

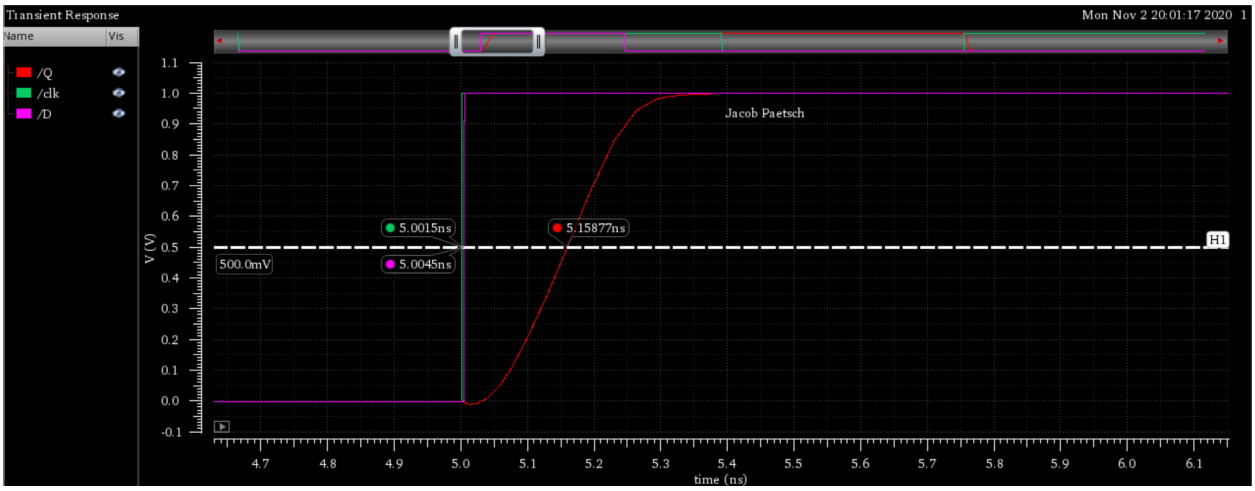

Figure 18 - Setup Time Plot

Note that although the horizontal marker is halfway through the plots at 500.0 mV, based on the tables in Cadence the difference remains constant (the clock and input rise time is negligible as they go from 0 to 1).

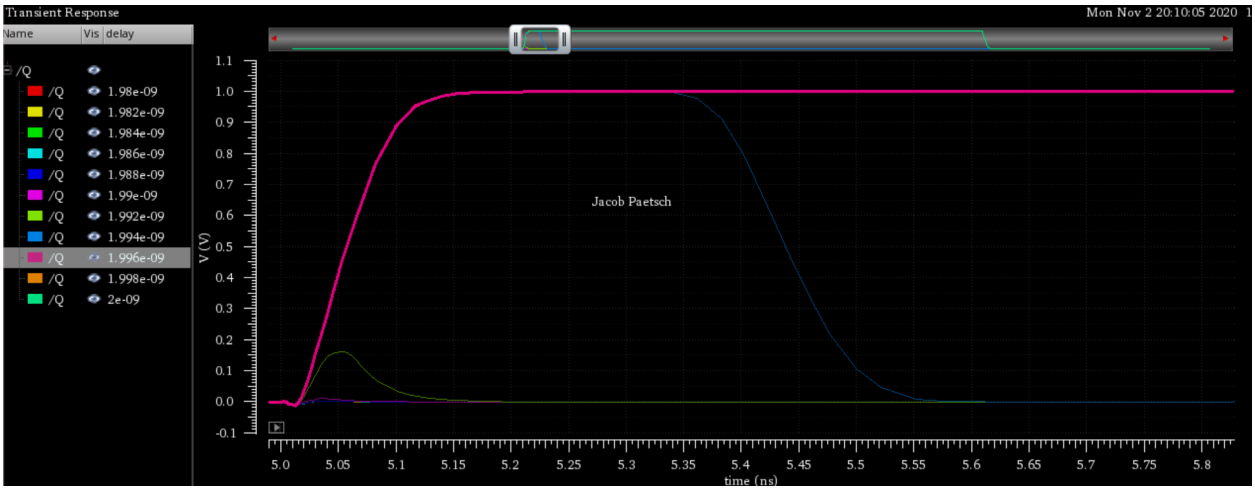

Figure 19 - Parametric Smallest Delay Value Plot (For Hold Time)

We do virtually the same process to find the hold time, but first we must isolate the minimized delay value. As can be seen in the plot, the minimum delay value that holds a stable output logic value is 1.996 ns. Using this as our input delay, we then check the first falling input edge against the first clock rising edge. This gives us the hold time to be (5.0015ns - 4.9975ns) = 0.004ns (**Figure 20**).

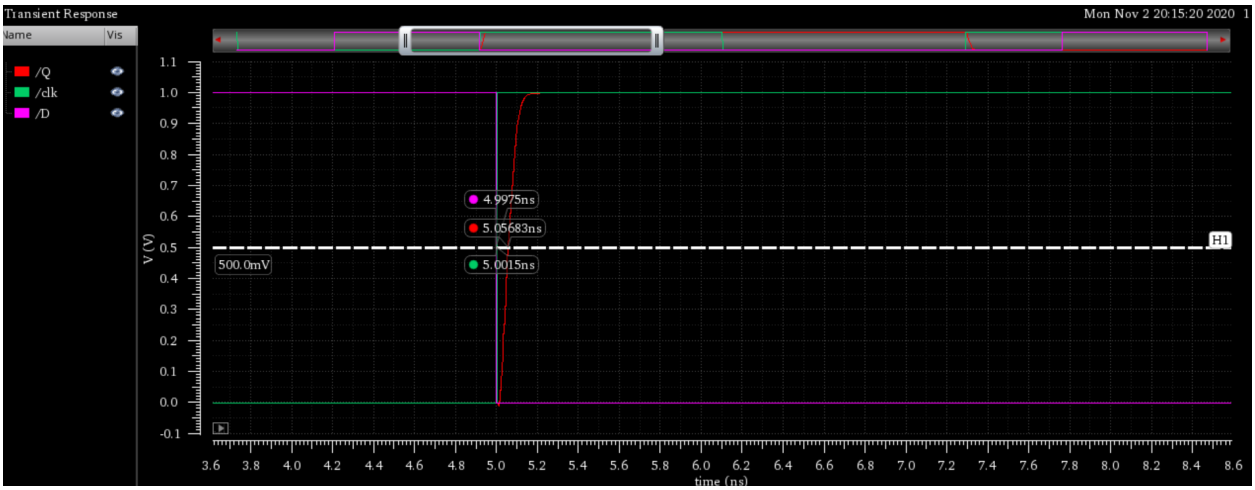

Figure 20 - Hold Time Plot

Questions:

1) Describe how the propagation delays change as temperature changes, and for different process corners.

As evidenced by the numerous plotted lab simulations, as the temperature increases both the rise and fall times increase. Temperature affects electron (and hole) mobilities, so transistor speed slows down when it gets hotter and the circuit performance is negatively impacted.

For the different process corners we see the general behaviour that should be expected from these models. For example in SS, both the NMOS and PMOS are 'slow' so proves to be the greatest rise and fall propagation delays. In comparison, SS < TT < FF for their corresponding rise/fall propagation delays. For the SF and FS variants, the rise and fall are affected separately. For example, SF means that NMOS is slow and PMOS is fast. Since the NMOS are pulling the circuit to ground (high to low or fall propagation delay), this will take longer. In comparison, the rise propagation delay time decreases as the PMOS transistors pulling the circuit to Vdd are 'fast'.

These different combinations of process corners give a range of propagation delay speeds and are important to take into account when designing for real-world applications, particularly for more precise or sensitive systems.

2) How does setup time affect the maximum clock frequency that you could run your circuit at?

The setup time is the minimum amount of time required before each clock edge for which the data about to be latched must be valid. This directly contributes to part of the minimum clock period (in addition to the time from clock to input change, and propagation delay) and therefore the maximum clock frequency.

If you increase the clock frequency of a circuit beyond the maximum you could have significant issues, with data not latching properly or latching on to previous/unstable values; computer crashes caused by overclocking are a first-hand example of why maximum clock frequency is a vital consideration in our circuits.

3) Why do you think the propagation delays for the falling output edge are always about twice as large as they are for the rising output edge? Hint: consider the internal nodes of the flip-flop and it's operation for high and low inputs. (see Fig.1 of the prelab; especially the last two stages)

In the critical path of the circuit, the major stages have twice as many NMOS (which go to ground). This means that on the way to 0, there are roughly twice as many capacitors to discharge. This increases the propagation delay when going from high to low voltage to roughly two times the rising edge.