

ECE 3150: VLSI & Advanced Digital Design

Project 3

Part 1: 1 Bit Full Adder

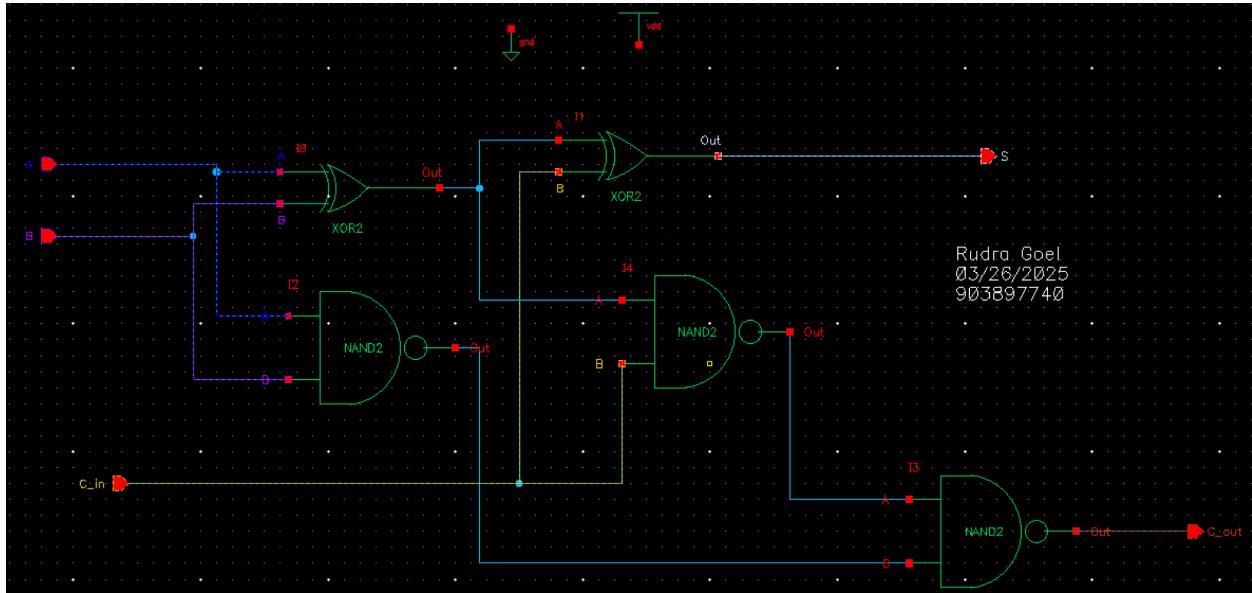


Figure 1. 1-bit Full Adder Schematic

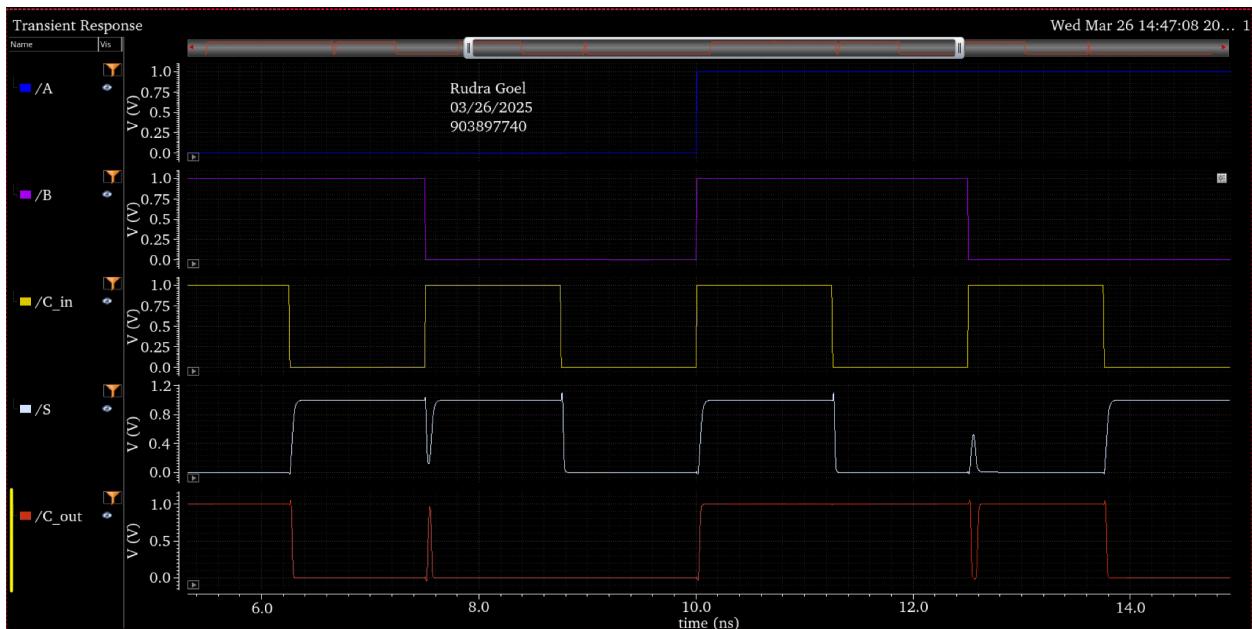


Figure 2. Transient Simulation For All 8 Input Combination of 1-bit Full Adder

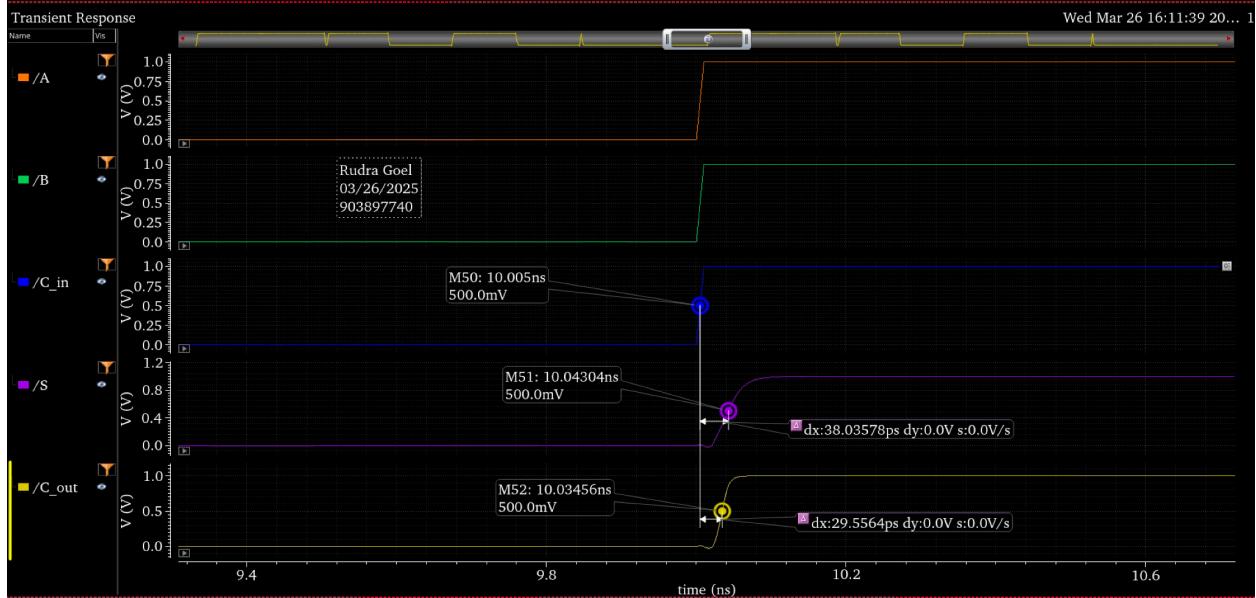


Figure 3. Propagation Delay Measurements of 1-bit Full Adder at 1.0V V_{DD}

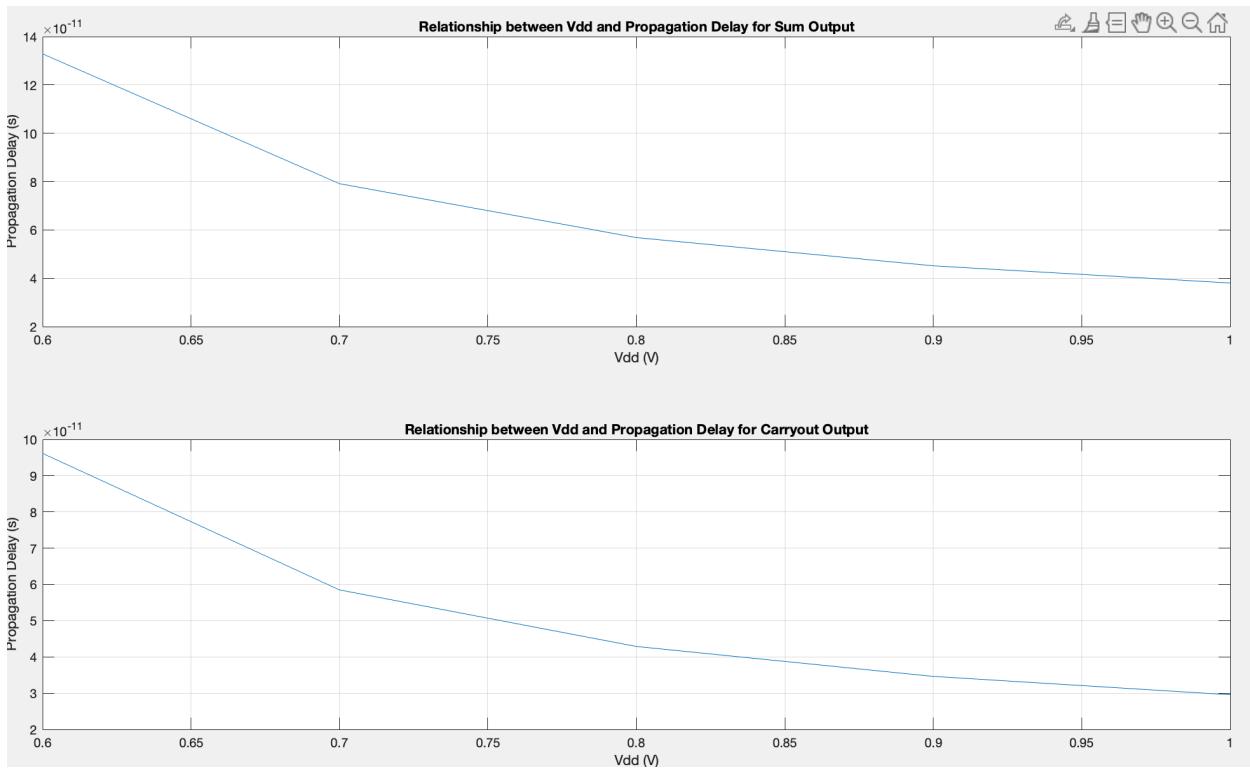


Figure 4. Graphs Illustrating Relationship Between Supply Voltage and Propagation Delay for Carryout and Sum Outputs

Part 2: 4 Bit Ripple Carry Adder (RCA)

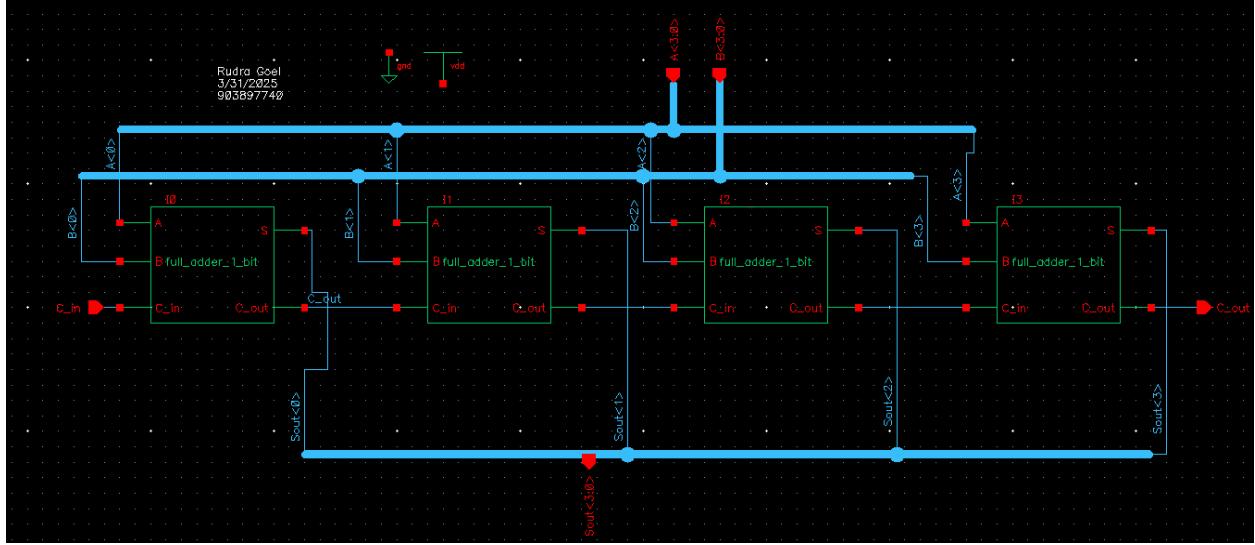


Figure 5. 4 Bit Ripple Carry Adder Schematic

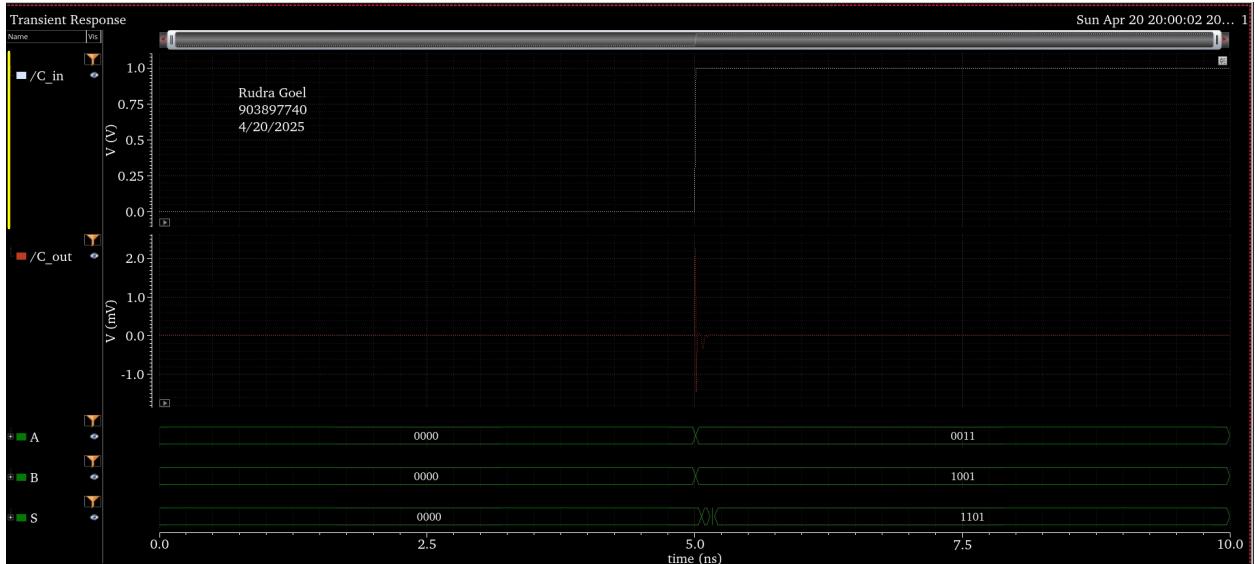


Figure 6. Transient Simulation of 4 Bit Ripple Carry Adder

Overall Delay of 4 Bit Ripple Carry Adder: ~ 155.75 ps



Figure 7. Overall Delay Measurements of 4-bit RCA

Part 3: 16 Bit RCA

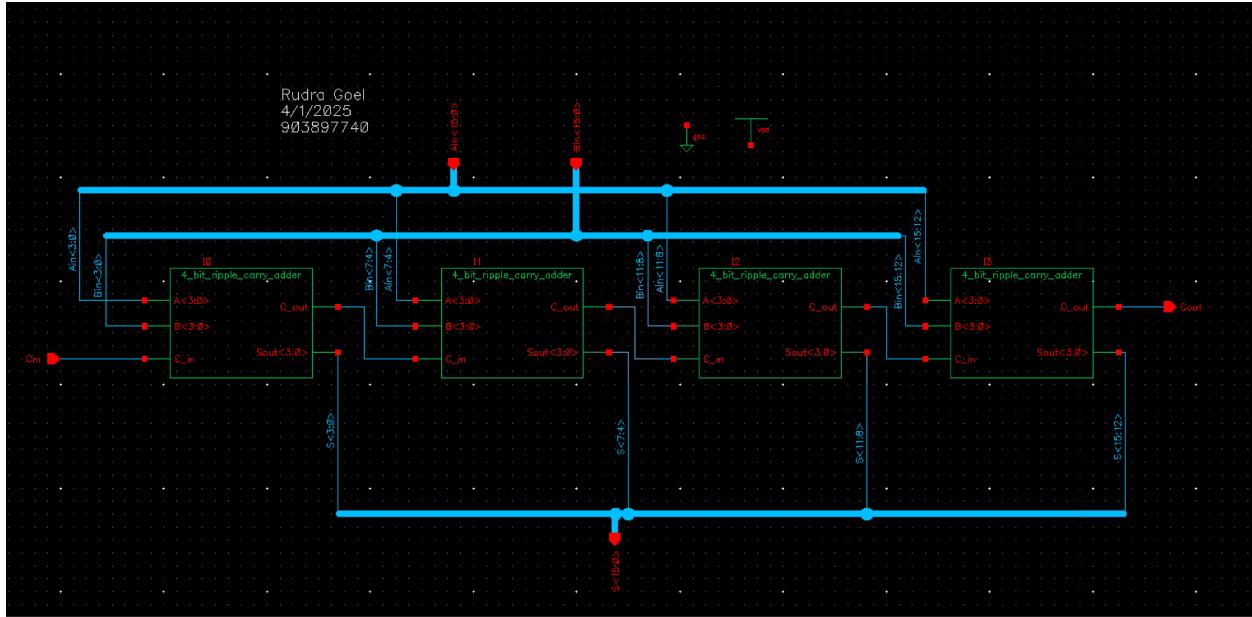


Figure 8. Schematic of 16-bit RCA

Part 4: D-Flip Flop

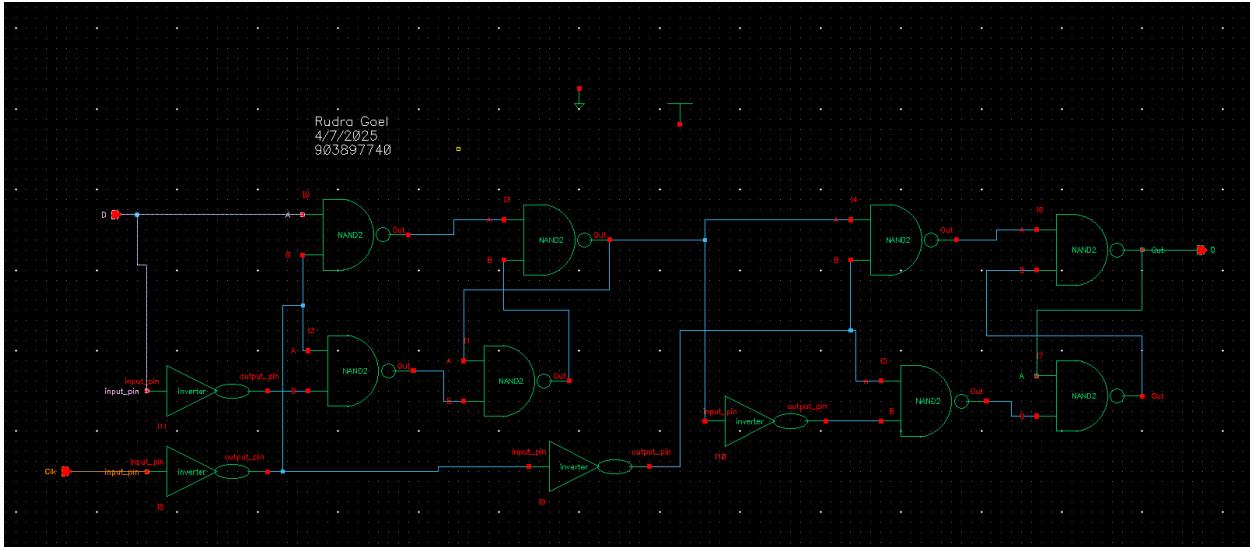


Figure 9. Schematic of D-Flip Flop

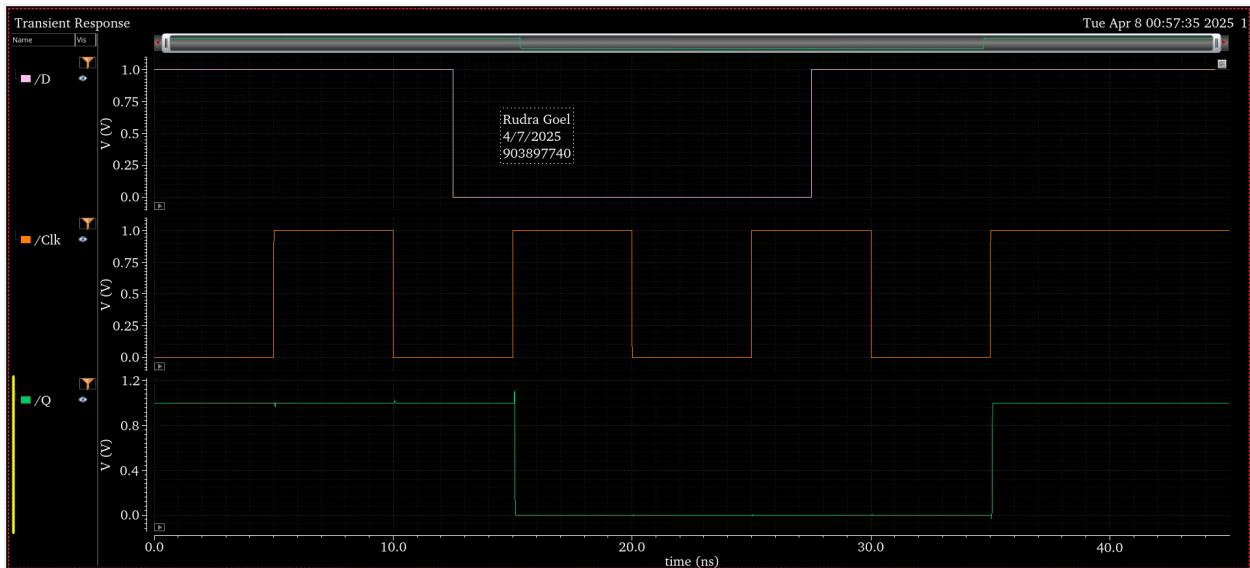


Figure 10. Transient Simulation of D-Flip Flop

Part 5: 16 Bit Register File

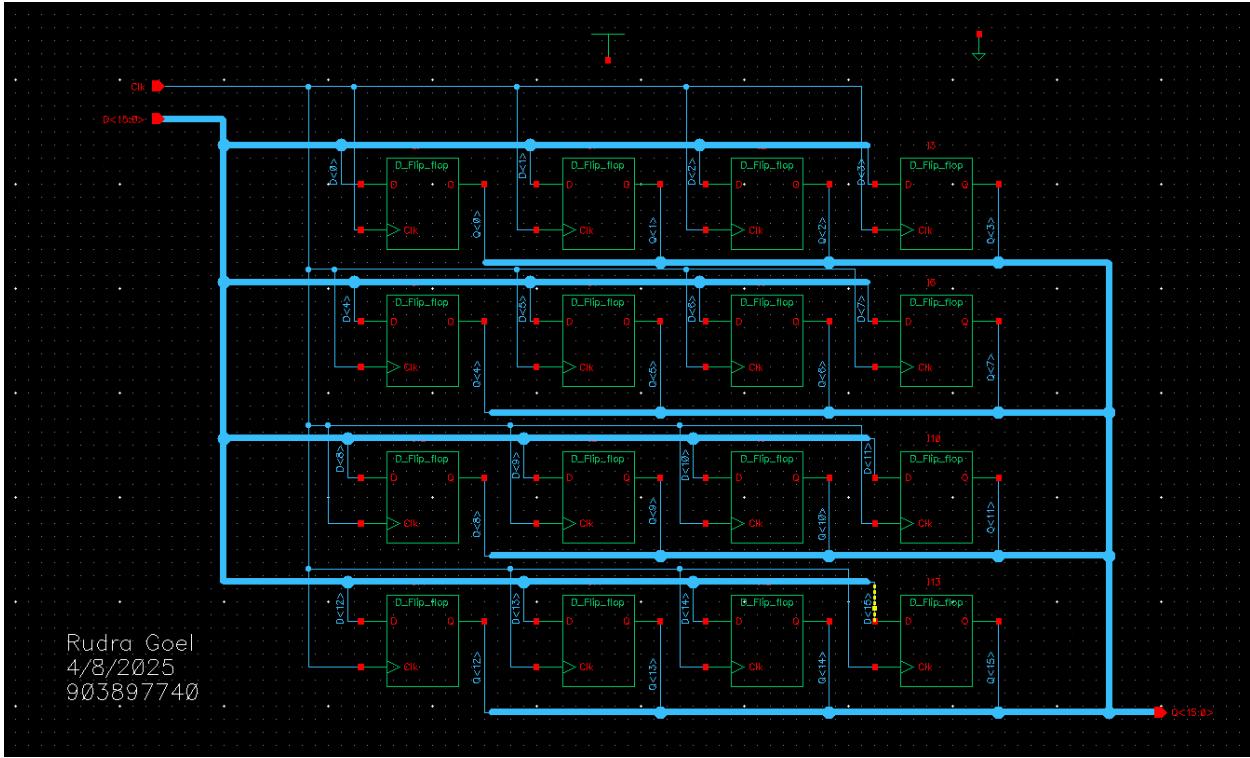


Figure 11. Schematic of 16 bit Register File

Part 6: Synchronous Programmable 16 Bit Add/Sub Computer

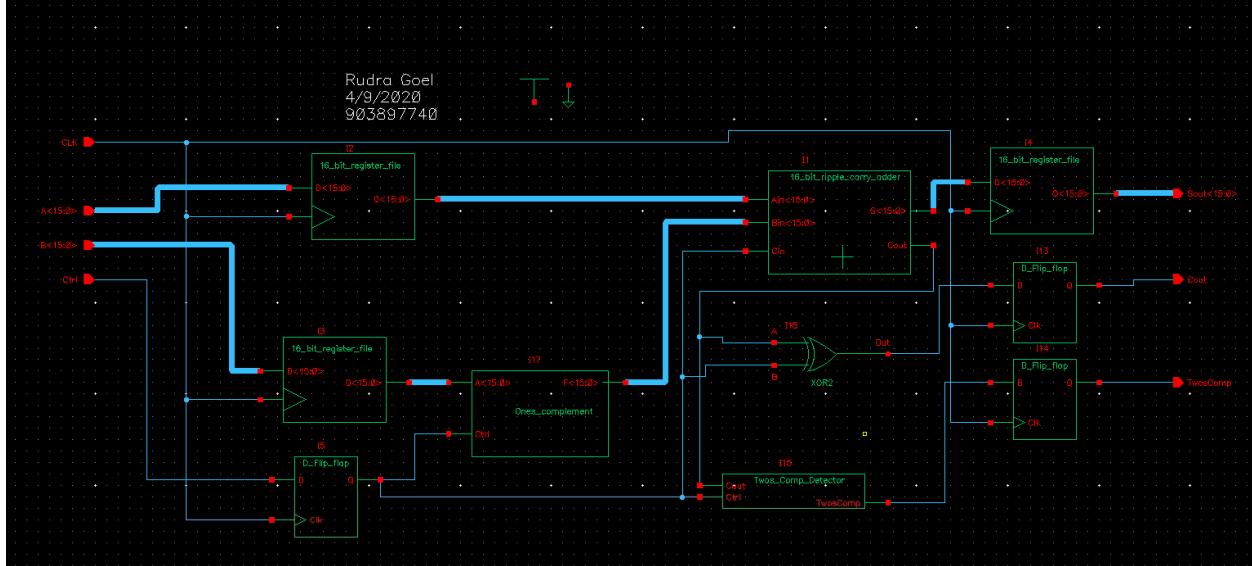


Figure 12. Block Level Schematic of 16-bit Add/Sub Computer [CHANGE]

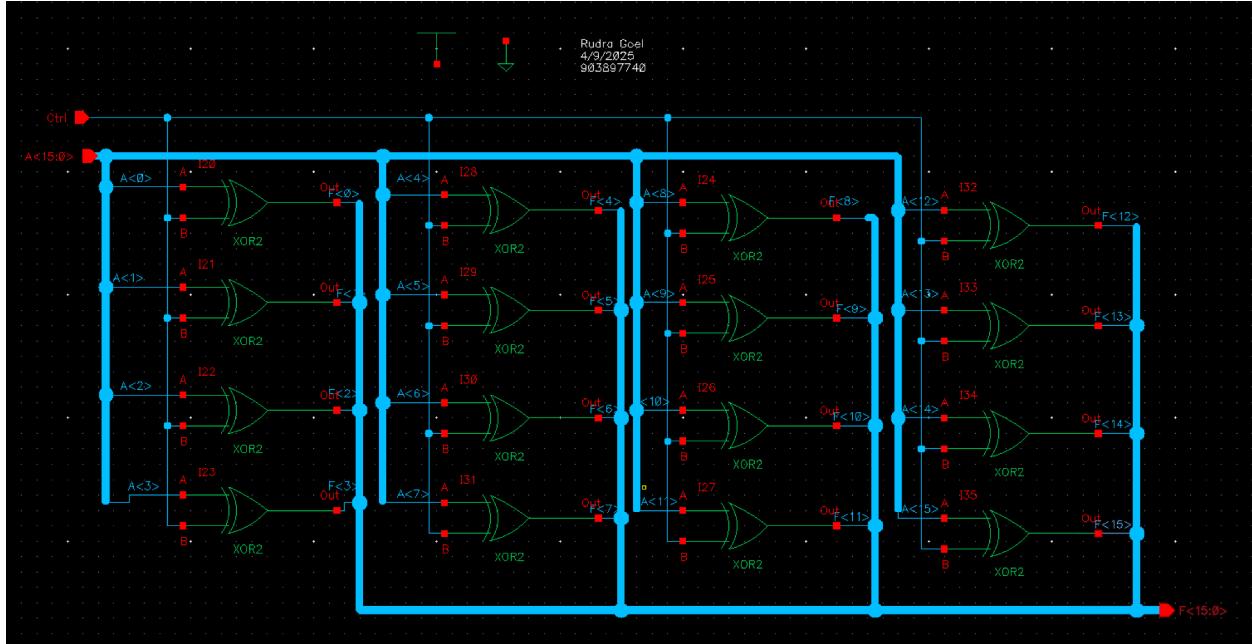


Figure 13. One's Complement Circuit

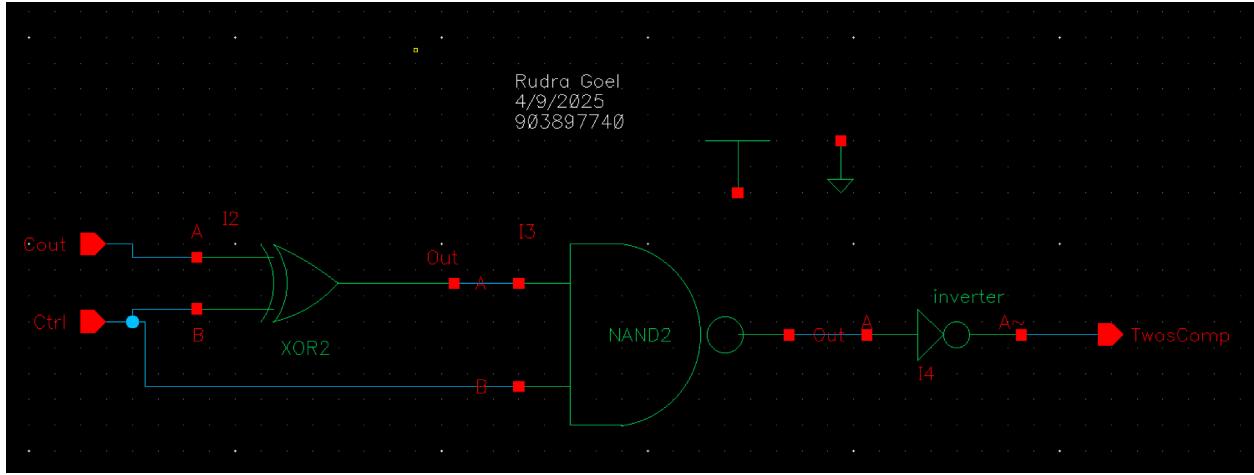


Figure 14. Two's Complement Detector Circuit

Explanation of Two's Complement Schematic:

The programmable aspect of our add/sub computer we want to achieve comes from the two's complement derivation of the input B. In order to correctly process $A - B$ while utilizing the 16 bit adder hardware previously developed, we can visualize the operation $A - B$ as $A + (-B)$ where $-B$ is represented as the two's complement of B.

The two's complement of any binary number is derived by flipping its bits and adding one to the result. In order to flip the bits based on the “Ctrl” signal, we can utilize a XOR gate that effectively flips the bits of one input when its other input is logic ‘1.’ By doing this in parallel for each bit, we can abut 16 XOR gates together to get the one's complement of B. By passing the “Ctrl” signal into the Carry-in of the 16 bit adder, we effectively derive the two's complement of B and can compute $A - B$.

The Two's Complement Detector circuit can be thought about logically with regards to the most significant bit's Carryout; if there is no carryout voltage and we wish to compute a subtraction operation, then we know the MSBs of A and B added together resulted in 1, meaning the signed bit is high implying that we need to take the Two's Complement in order to correctly interpret the result. If the Carryout bit was high, then the MSB resolved to zero and the answer is positive → Two's Complement should not be interpreted on the answer.

```

; this is the vector file simulation for the 16 bit programmable synchronous adder module

tunit ns
vih 1.0
vil 0.0
voh 0.9
vol 0.1
trise 0.01
tfall 0.01

vname CLK A<[15:12]> A<[11:8]> A<[7:4]> A<[3:0]> B<[15:12]> B<[11:8]> B<[7:4]> B<[3:0]> Ctrl

radix 1 4 4 4 4 4 4 4 4 4 1

;t period 5

;t    CLK    A4    A3    A2    A1    B4    B3    B2    B1    Ctrl
0     1      0      0      0      0      0      0      0      0      0
5     0      0      0      0      0      0      0      0      0      0
10    1      0      0      0      0      0      0      0      0      0
15    0      0      3      B      B      2      A      6      F      0
20    1      0      3      B      B      2      A      6      F      0
25    0      7      E      5      9      2      A      6      F      1
30    1      7      E      5      9      2      A      6      F      1
35    0      7      E      5      9      C      4      1      8      1
40    1      7      E      5      9      C      4      1      8      1
45    0      F      F      F      F      F      F      F      F      0
50    1      F      F      F      F      F      F      F      F      0
55    0      0      0      0      0      F      F      F      F      1
60    1      0      0      0      0      F      F      F      F      1
65    0      6      C      0      E      1      F      0      3      0
70    1      6      C      0      E      1      F      0      3      0
75    0      6      C      0      E      E      C      E      1      0
80    1      6      C      0      E      E      C      E      1      0
85    0      F      F      F      F      0      0      0      0      1
90    1      F      F      F      F      0      0      0      0      1
95    0      0      0      0      0      0      0      0      0      1
100   1      0      0      0      0      0      0      0      0      1
105   0      0      0      0      0      0      0      0      0      0
110   0      0      0      0      0      0      0      0      0      0

```

Figure 15. Vector File Script for Transient Simulation of 16-bit Computer



Figure 16. Transient Simulation Waveform of Vector File Script

Part 7: Determining F_{Min} and F_{Max}



Figure 17. Add/Sub Computer Operating at Maximum Frequency

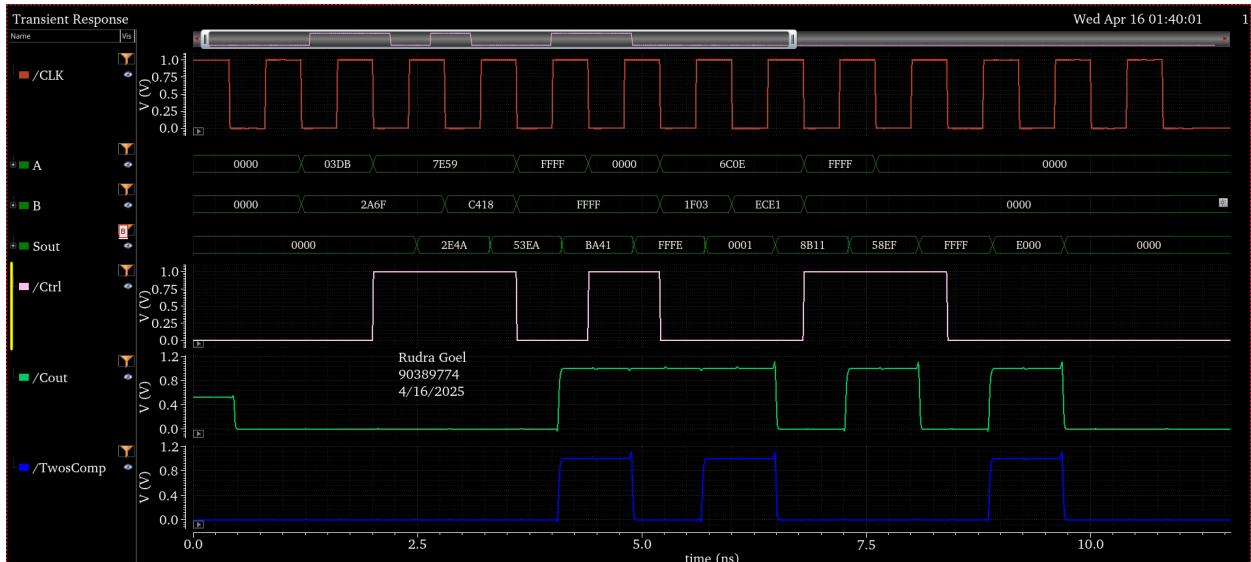


Figure 17. Add/Sub Computer Simulated at Minimum Failing Frequency

Maximum Operating Frequency: 1GHz

Minimum Failing Frequency: 1.1GHz

Part 8: Identifying Critical Path

The design of the 16 bit add/sub computer fails when operating at a clock frequency of 1.1GHz. Our design **fails when computing the correct value for both the TwosComp signal and the proper carry out bit (Cout)**. This implies that the critical path in our circuit is the path taken to output the correct signals. This makes logical sense as our design utilizes the carry out signal from the 16 bit RCA for the TwosComp detector and the Cout Correction circuit. The image below highlights the additional path that electrons must propagate through in order for the correct Twos Comp signal to be outputted. The waveform from the previous page highlights this error as the signal goes high on the 11th rising edge when it should not.

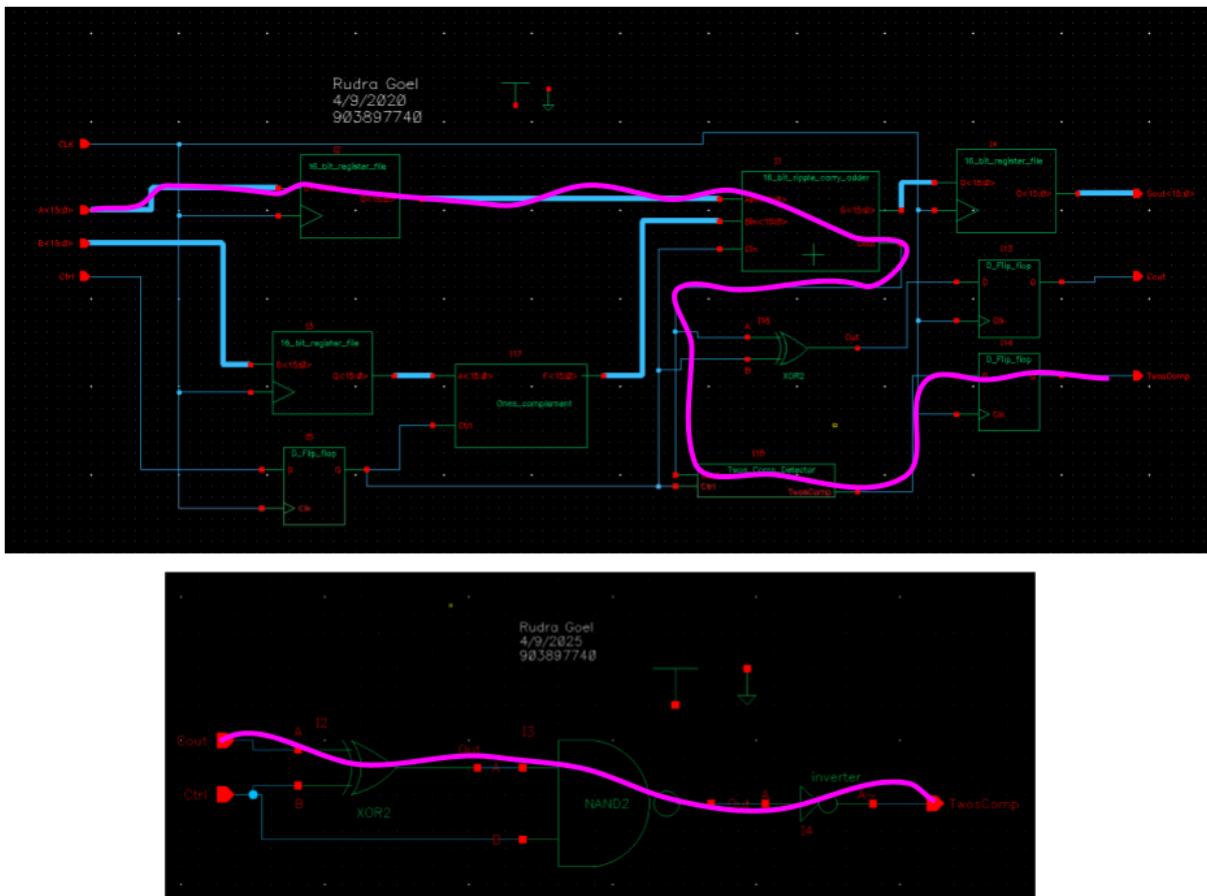


Figure 18. Add/Sub Computer Simulated at Minimum Failing Frequency

Part 9: Supply Voltage Effects on Max Operating Frequency

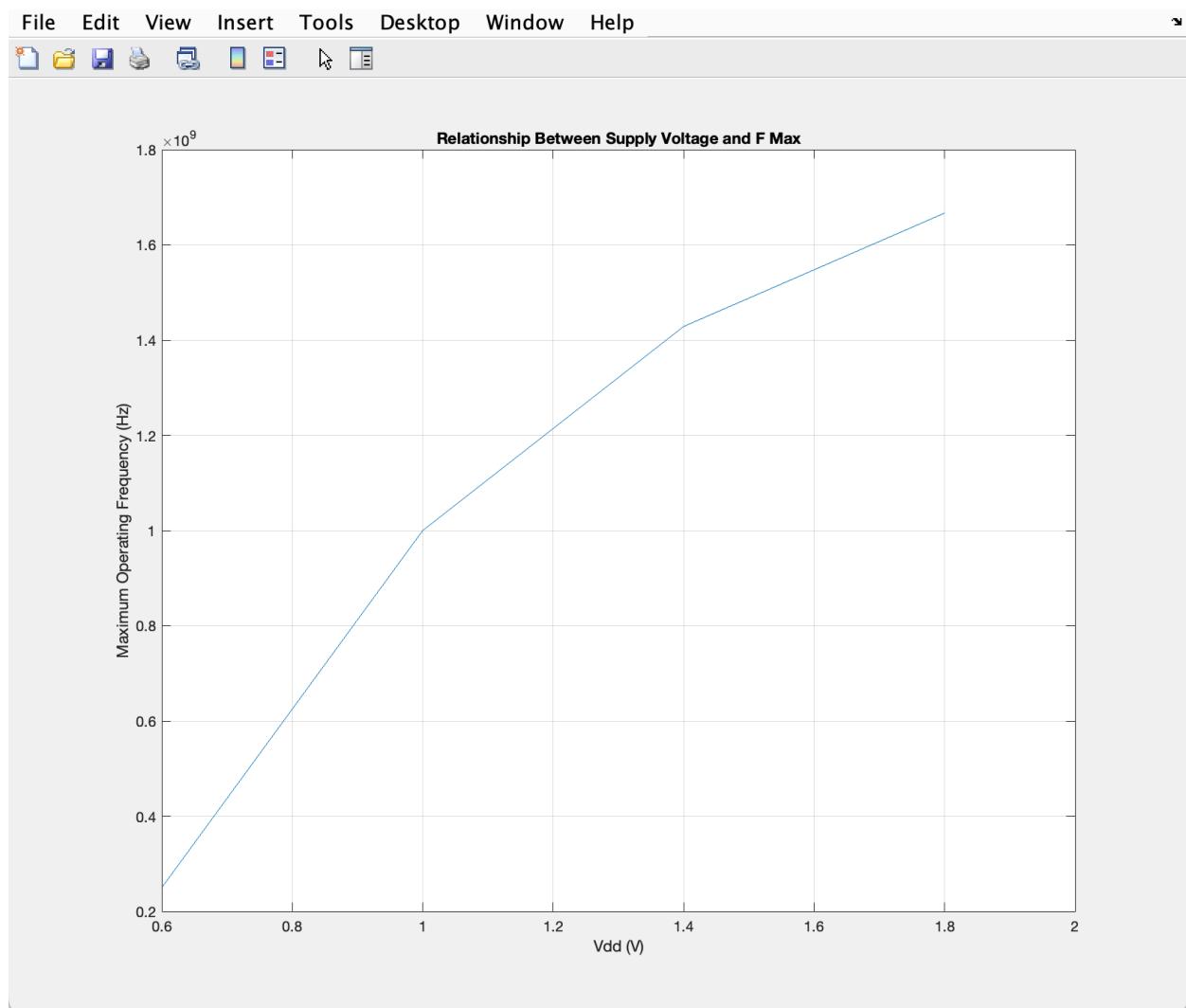


Figure 19. Maximum Operating Frequency Scales With V_{dd}

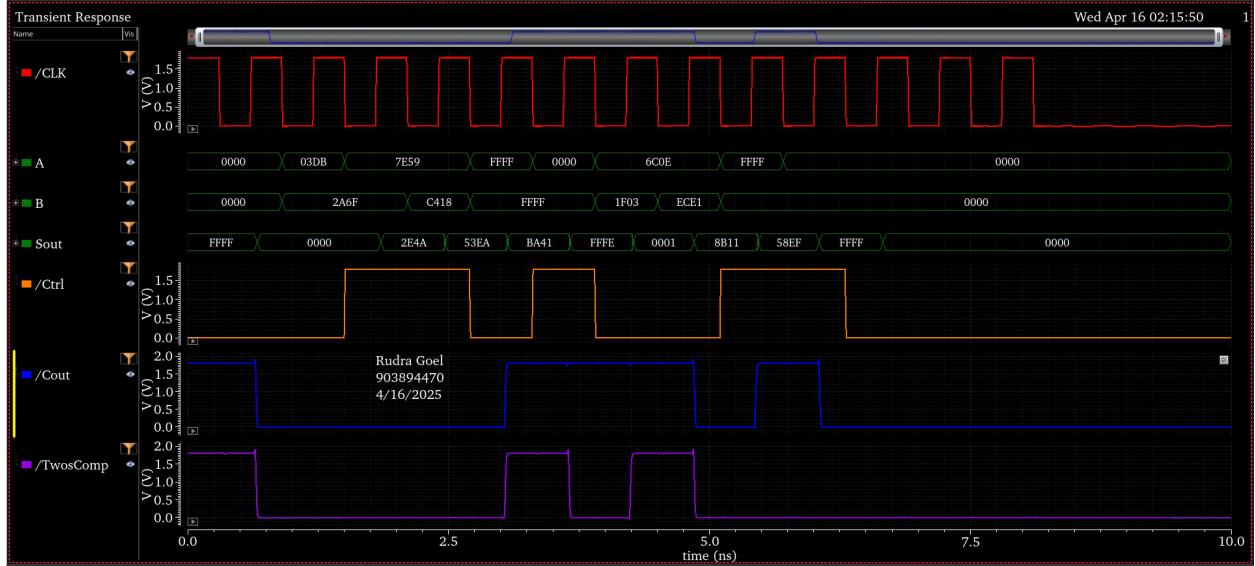


Figure 20. Add/Sub Computer Operating at Maximum Frequency With Vdd @ 1.8V

The maximum operating frequency follows a linear relationship with the supply voltage because as V_{dd} scales, the gate to the source voltage of each transistor in the circuit increases (in the case of nMOS, conversely with pMOS). This change in V_{gs} increases the drive current which in turn reduces the time it takes to charge the load capacitance on each device; this is significant as the charging time for each load capacitance is what affects the propagation delay of the circuit. By increasing supply voltage, we effectively reduce the delay and increase the operational frequency.

Part 10: Full Layout

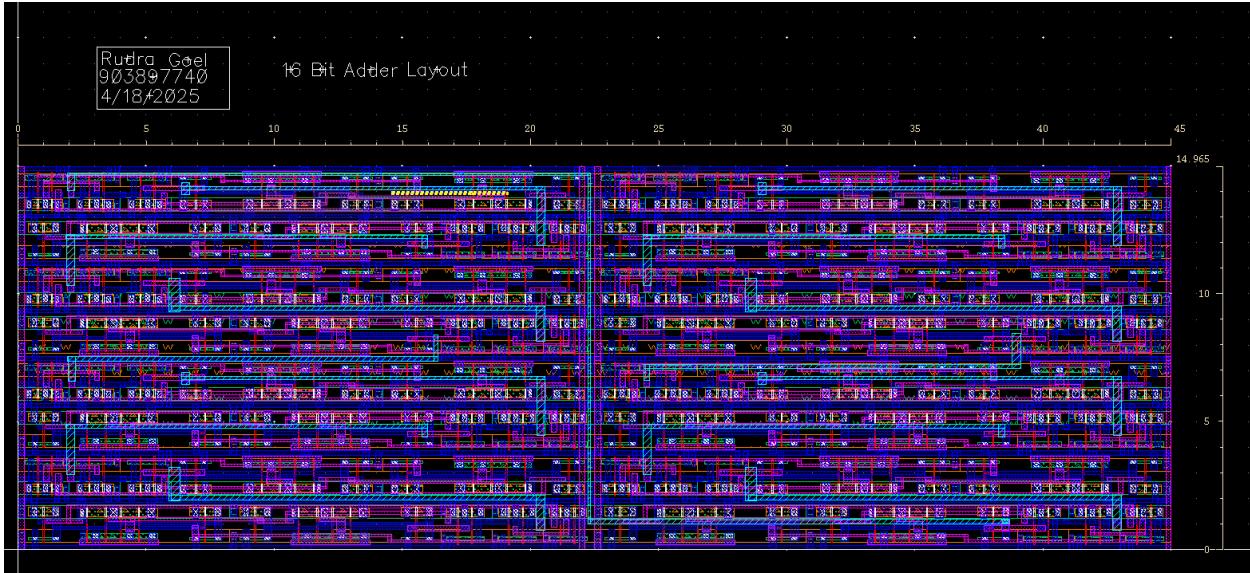


Figure 21. Layout of 16 Bit Ripple Carry Adder

Height: 14.965 μm

Width: 45 μm

H/W Aspect Ratio $\sim \frac{1}{3}$

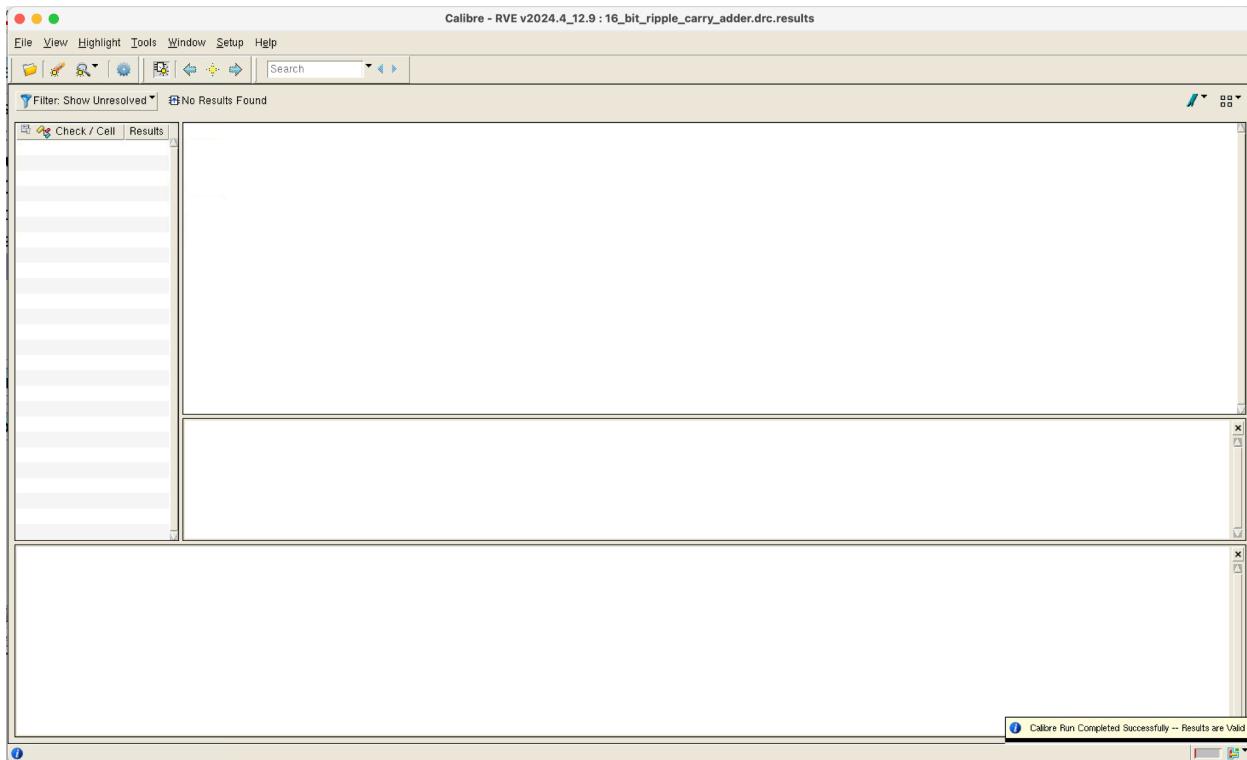


Figure 22. DRC Runs Clean

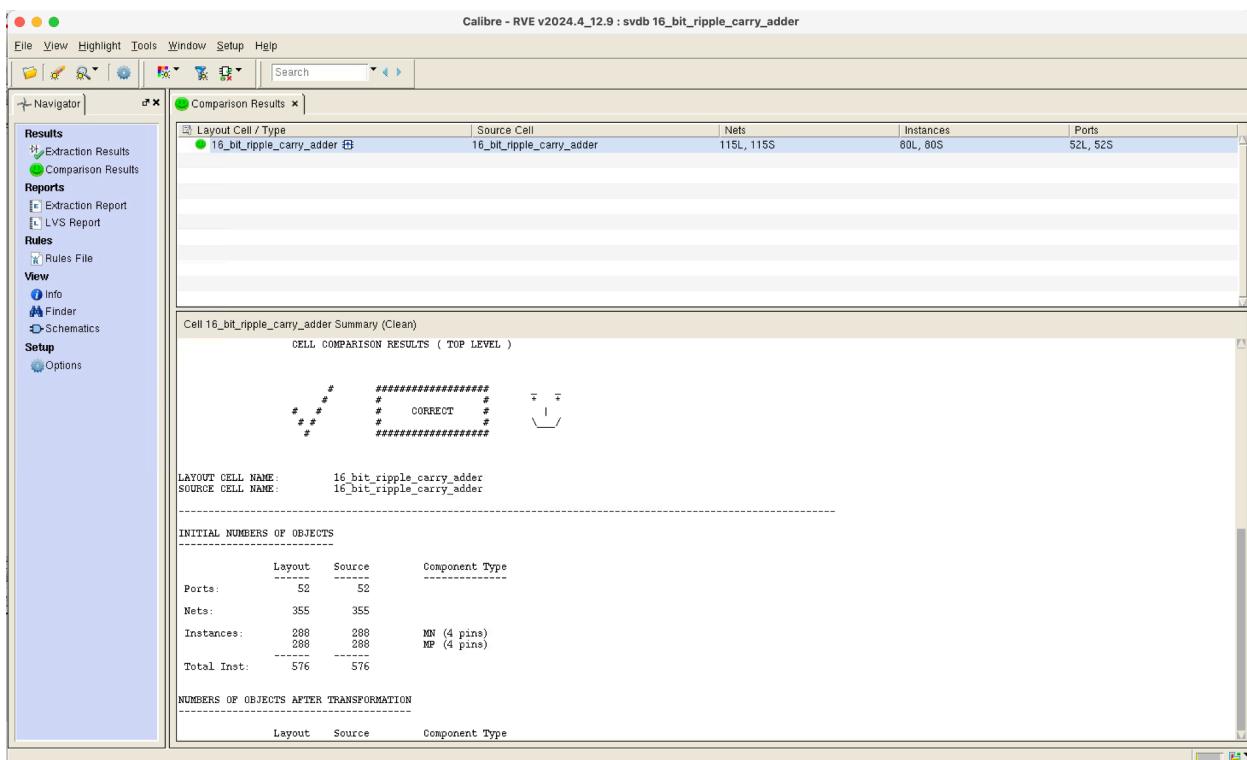


Figure 23. LVS Runs Clean

Part 11: Sweeping Setup and Hold Time for DFF

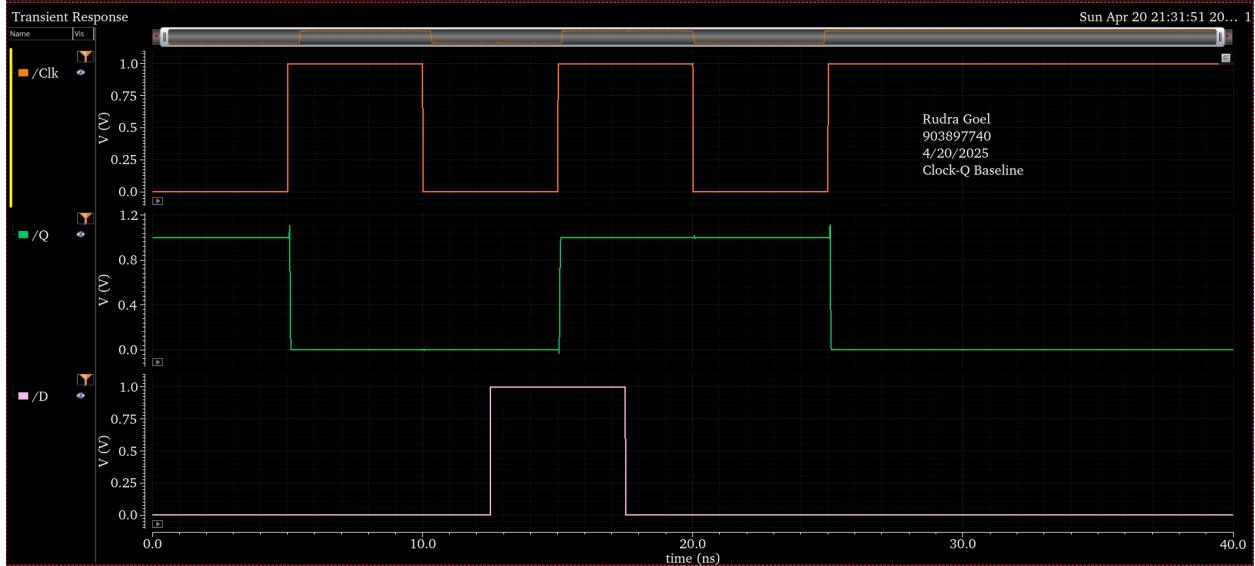


Figure 24. Transient Simulation for Baseline $T_1 = 0.5T$ & $T_2 = 0.5T$ for D-Flip-Flop

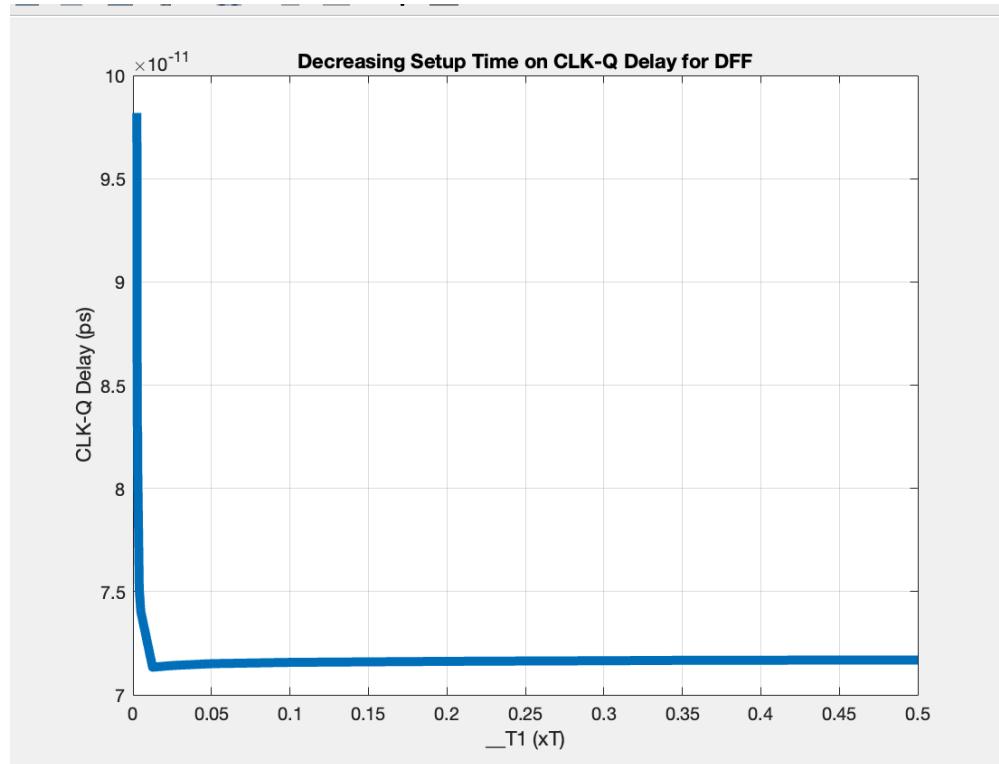


Figure 24. Graph Illustrating Setup Time Sweep From .5T to 0T and its Effects on CLK-Q Delay

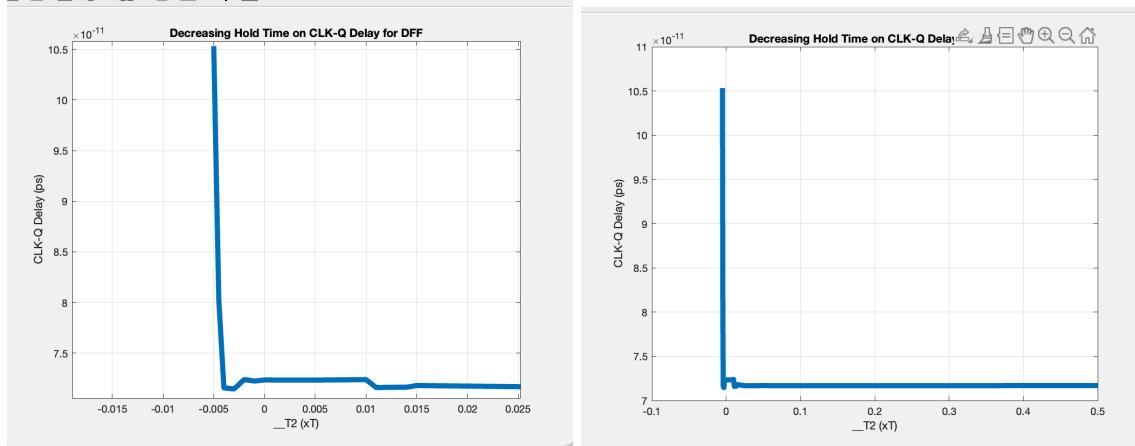


Figure 24. Graphs illustrating Hold Time Sweep from .5T to 0T and its Effects on CLK-Q Delay

Table 1. Values of T1/T2 and CLK-Q Delay Measurements (design fails for delays == 0)

T1	Delay (s)	T2	Delay (s)
0.5	71.687	0.5	71.688
0.45	71.682	0.4	71.687
0.4	71.68	0.3	71.685
0.35	71.672	0.2	71.681
0.3	71.652	0.1	71.684
0.25	71.639	0.05	71.686
0.2	71.622	0.025	71.685
0.15	71.599	0.015	71.8
0.1	71.573	0.014	71.633
0.05	71.509	0.013	71.625
0.025	71.42	0.012	71.616
0.0125	71.338	0.011	71.598
0.005	74.029	0.005	72.385
0.004	75.127	0.005	72.335
0.003	81.292	0	72.344
0.00295	82.809	-0.001	72.244
0.0027	83.161	-0.002	72.39
0.0026	91.359	-0.003	71.461
0.0025	98.186	-0.004	71.566
0.00225	103.24	-0.0045	80.213
		-0.005	116.33
		-0.009	0
		-0.25	0

Part 12: Determining Setup Time

Setup Time: $0.0033T \sim 16.5\text{ps}$

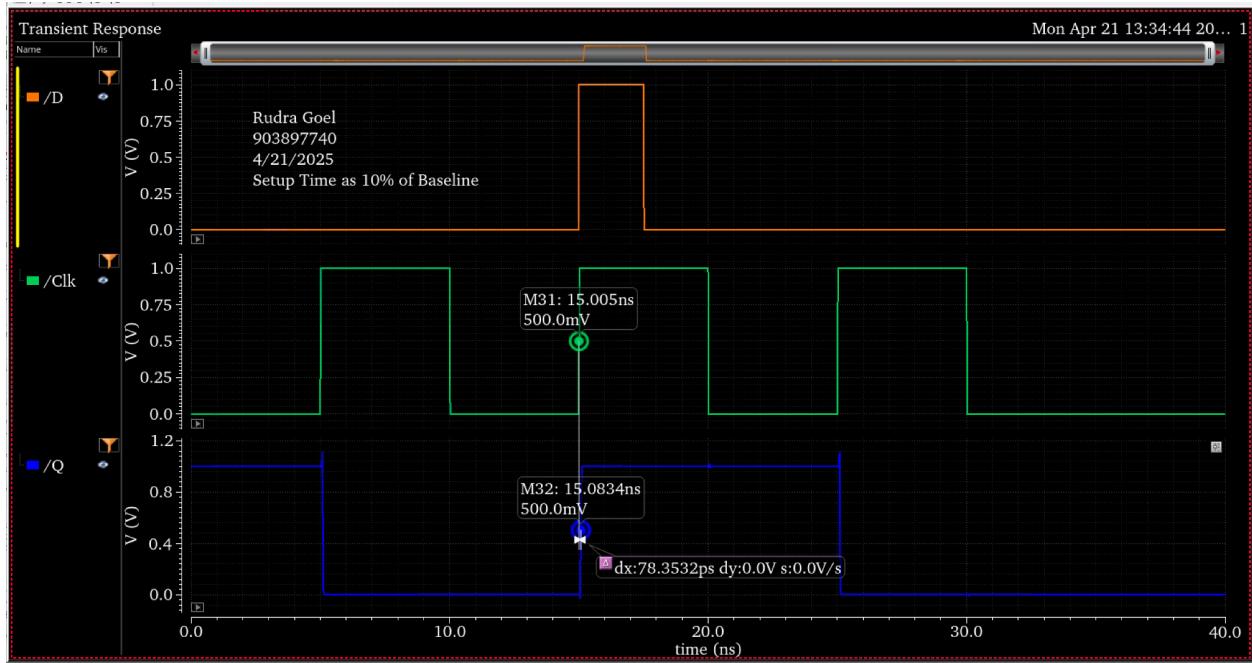


Figure 25. Transient Simulation of Setup Time Being 10% of Baseline (.5T)

Part 13: Determining Hold Time

Hold Time: $-0.004759T \sim -23.79\text{ps}$

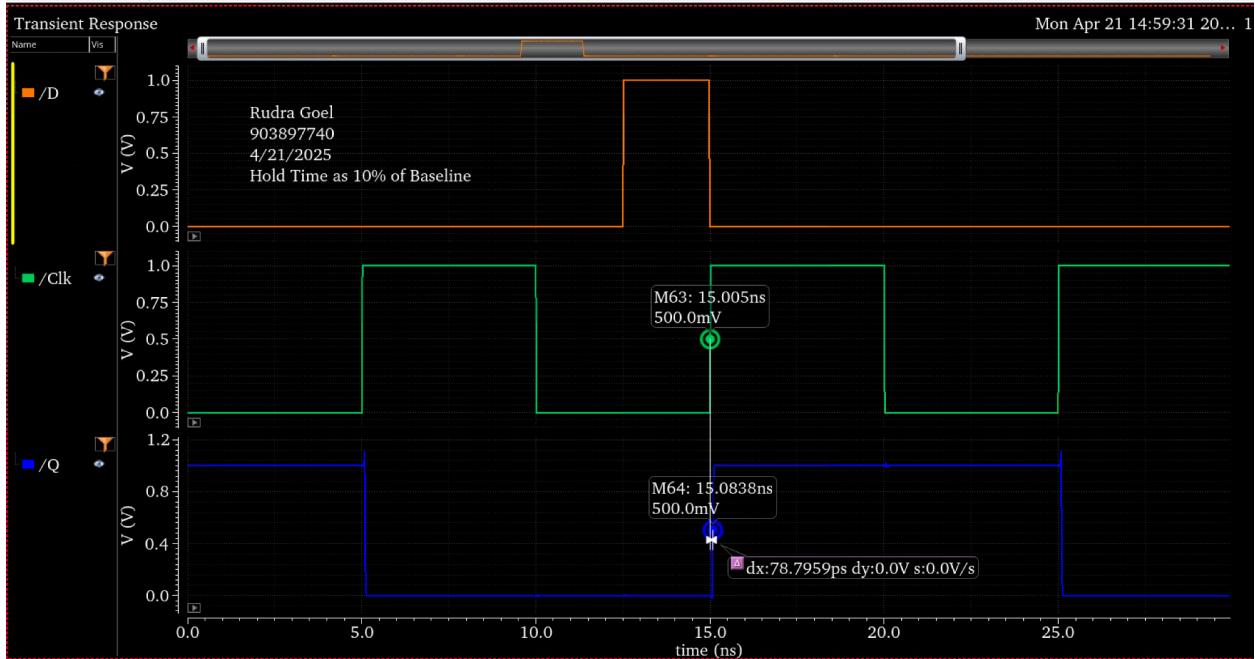


Figure 25. Transient Simulation of Hold Time Being 10% of Baseline (.5T)

Part 14: Explanations on Why CLK-Q Delay is Affected by Setup and Hold Times

Setup and hold times are crucial for proper use of flip flops. The setup time is the time required for the data at the input of the flip flop to be stable prior to the rising edge of the clock signal. Conversely, hold time is the time the input data signal must remain stable after the rising edge of a clock. Violating these constraints can lead to metastability issues where the D Flip Flop cannot latch onto the data correctly.

When the input signal violates the setup time constraint, the master latch of the D-Flip-Flop cannot latch onto the D input signal as the signal does not have enough time before the rising edge of the clock signal to propagate throughout the first latch and hold the data. Therefore, there needs to be sufficient time before the rising edge of the clock for the data input to be stable in order for that constant value to propagate through all the gates within the first latch and be stored within the flip flop properly. The flip flop would then require more time to propagate the signal (CLK-Q Delay) or may not produce the correct output at all.

Similarly, if the hold time is not met at the D input, then the latch requires more time to capture onto the signal to stabilize the output thus increasing the CLK-Q Delay.

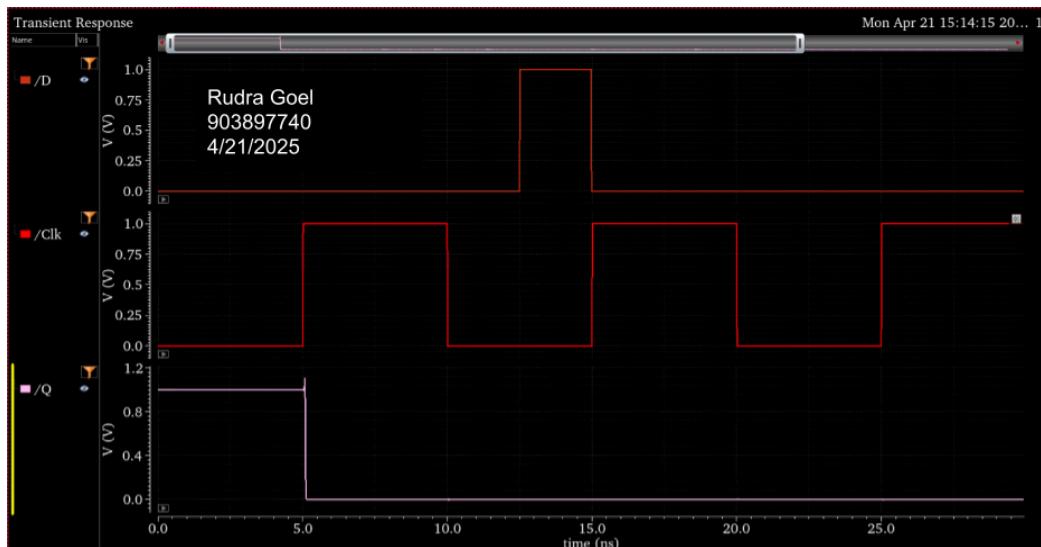


Figure 26. Insufficient Hold Time. Q Output is Invalid



Figure 27. Insufficient Setup Time. Q Output is Invalid

Part 15: Clock Skew Design

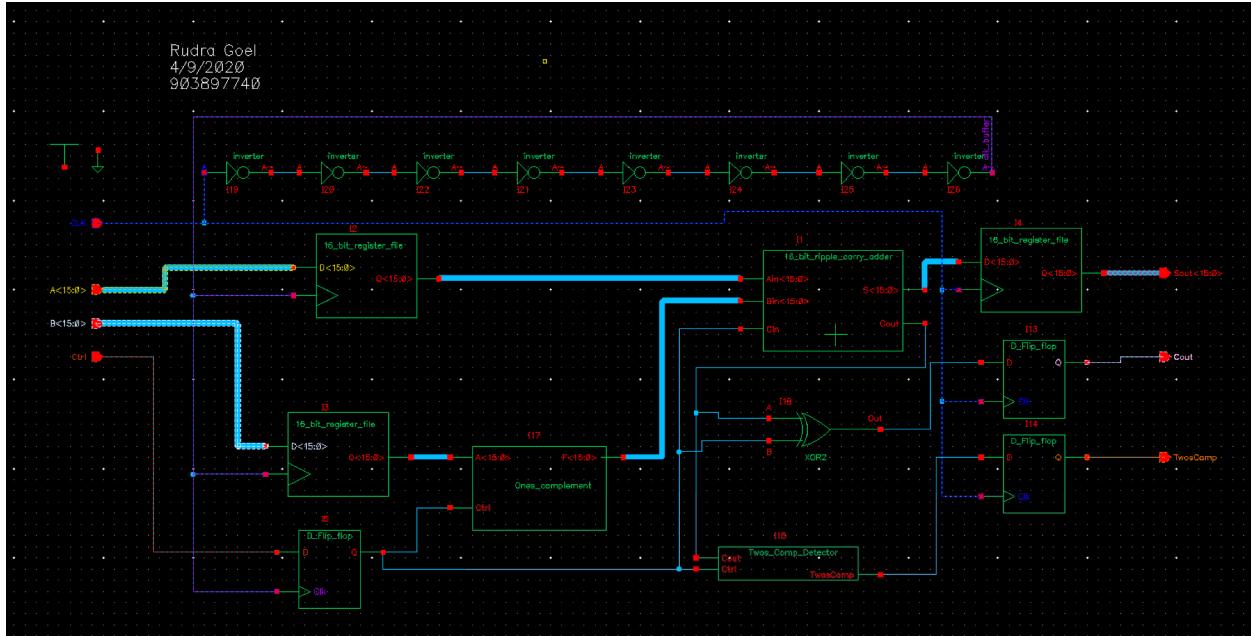


Figure 28. Synchronous Computer Schematic with Clock Buffer to Input Registers



Figure 29. Transient Simulation With Clock Buffer Illustrating Clock Skew

Clock Skew is -197.94 ps

Part 16: Determining Maximum Operating Frequency with Clk Buffer

$F_{\max} \sim 0.83\text{GHz}$

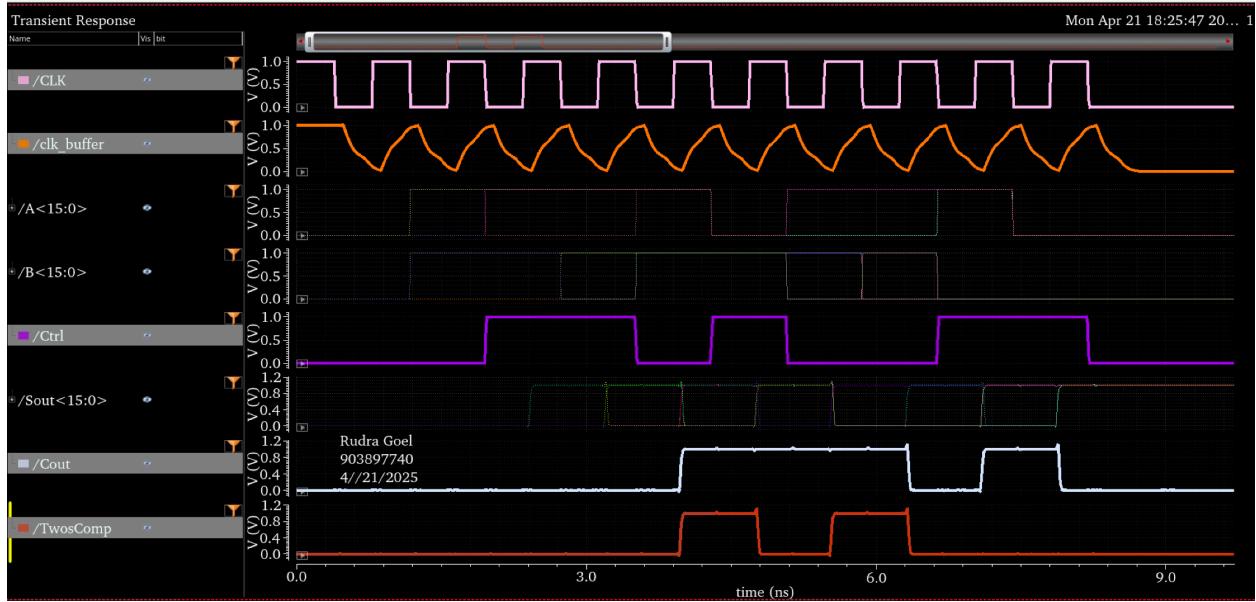


Figure 30. Clk Buffered Computer Operating at new F_{\max}

This new design which introduces a negative clock skew does change the maximum operating frequency of our computer; **it decreases the operating frequency by ~17%.**

The input registers have a larger delay due to the clock buffer in our design; this is the negative clock skew when compared with the registers on the output buffer. Because of this, our computer receives a delayed version of the inputs as the rising edge on the input registers are delayed when compared with the output registers. And because the combinational logic of our computer has an unchanged critical path, the propagation delay remains constant; as a result, when our computer operates at high frequencies and switches inputs fast enough, the computer cannot propagate correct values through from the input register to the output register without also violating the setup time of the D-Flip-Flops.

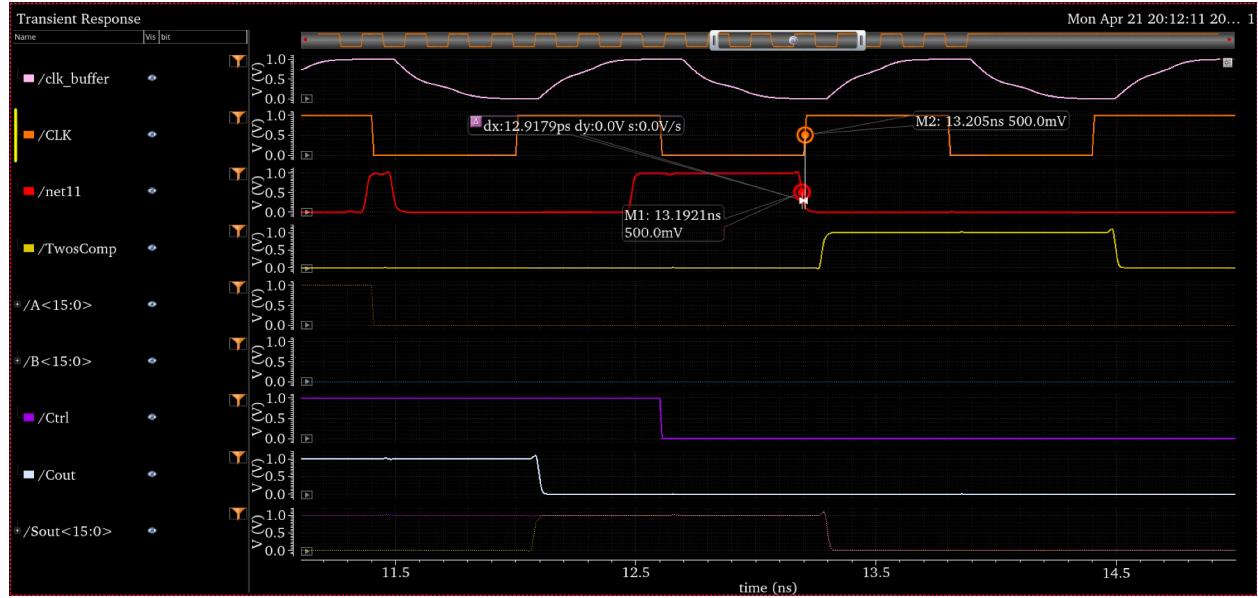


Figure 31. TwosComp Signal Violates DFF Setup Constraint to Latch Correct Signal

The figure above illustrates how our computer design received an input on the rising edge of the buffered clock signal which propagated through the computer; specifically, the TwosComp signal (named ‘net11’) becomes the correct value within the setup time of the DFF. In other words, the input signal to the output register TwosComp is not stable within the setup time of the DFF and thus the register latches onto the incorrect signal.

Part 17: Verifying Accuracy of Test Vector for 16b Computer in Part 6

In part 6, we tested our synchronous add/sub computer against various 16 bit vectors for A and B. When determining the maximum operating frequency for our add/sub computer, we must consider the critical path of the circuit; the critical path in our design is the path taken from the input vectors to resolving the correct value of signal TwosComp. This signal requires the Cout signal from the 16 bit adder module; given that this adder is a ripple carry adder, the Cout for each successive bit depends on the result of the previous bit operations. Additionally, the TwosComp signal must go through another circuit highlighted in Figure 14.

In order for us to determine the maximum frequency of the circuit, we need an input vector that transitions states and takes the longest path in the circuit to transition. Because we have two input vectors each 16 bits wide, there are 2^{32} possible inputs that can cause each carry-in bit from the previous full adder to switch states. This is the critical path in conjunction with the TwosComp detector circuit changing. As such, **the waveform online does not** test for the case where the carry-in bit has a ripple effect and changes for each subsequent full adder in the 16 bit RCA.

Part 18: Subthreshold Current Leakage Baseline Analysis

pMOS Leakage Current ~ 1.3986 nA

nMOS Leakage Current ~ 1.3471 nA

The leakage currents for both the pMOS and nMOS are roughly the same. This is primarily because we optimized the design of the transistors to have widths that ensure a V_m of $0.5V_{dd}$. Because hole mobility is typically $\frac{1}{3}$ or $\frac{1}{4}$ of electron mobility, we size the pMOS to have quadruple the width of the nMOS such that their overall currents in any region of operation are the same. This is why the leakage current for both the devices are roughly the same.

Part 19: Subthreshold Current Leakage Analysis on pMOS Width

Table 2. Effects of Varying pMOS Width On Leakage Current

Width	Leakage Current
100n	385.62 pA
200n	812.42 pA
300n	1.239 nA
400n	1.6652 nA
500n	2.092 nA

Table 3. Effects of Varying pMOS Channel Length On Leakage Current

Length	Leakage Current
100n	10.677 pA
200n	6.651 pA
300n	5.9104 pA
400n	5.594 pA
500n	5.415 pA

The data above illustrate a proportional relationship between width and leakage current and an inversely proportional relationship between gate length and leakage current. Considering the 3D shape of the pMOS device, if the width of the transistor grows, the source and drain regions of the device encounter more cross-sectional area for electrons to flow; in the case of leakage current, there is simply more area for electrons to propagate off the source to the drain.

Likewise, the gate length is inversely proportional to I_{DS} as there is less of a tendency for electrons to move even in the cut-off region of a device if the length it must travel from drain to source increases. As the gate length increases, the work each electron must use to move itself (ΔU) scales and thus the leakage current drops.

The data highlighted in the above two tables supports these claims.

Part 20: Characterizing High V_{th} on Leakage Current

pMOS Leakage Current for Transistor Under FreePDK45: 2.827 pA
nMOS Leakage Current for Transistor Under FreePDK45: 5.412 pA

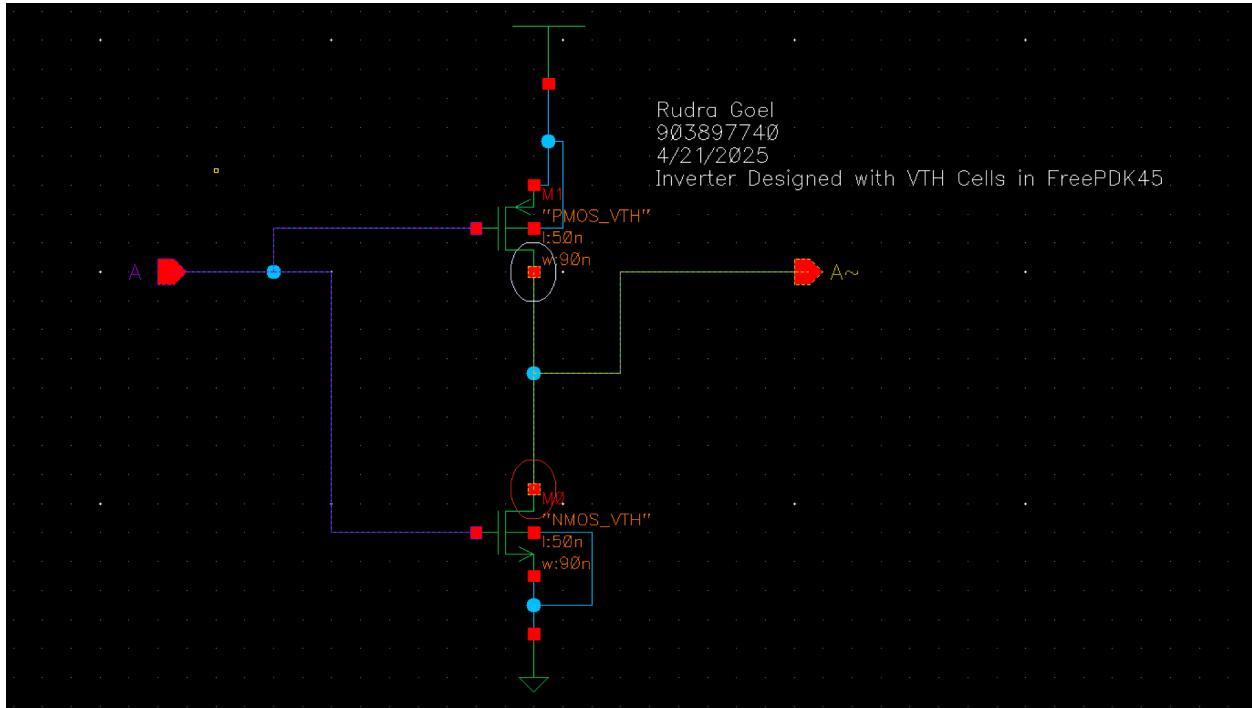


Figure 40. Inverter with V_{th} FreePDK properties.

The leakage current is dependent on the current I_{DS} for a MOS device operating in the linear or saturation region. I_{DS} is inversely proportional to the threshold voltage V_{th} , thus as V_{th} increases, the current in the linear or saturation region would decrease. As a result, the leakage current will decrease as well. This is illustrated in the data above and in part 18 where the leakage current is three orders of magnitude larger than the leakage current for the VTH MOS devices used in FreePDK45.

Although having a smaller leakage current is desirable, it comes at the cost of using a higher supply voltage V_{DD} in order to raise gate voltages high enough to switch the necessary transistors on or off. Transistors with a lower threshold voltage can use a smaller supply as the gate voltage only needs to surpass the threshold voltage to begin to operate.

Applications that switch devices back and forth will not benefit from a high V_{th} as the leakage current will not affect the overall design since the transistor itself is not staying in one constant state for too long. If there is a high supply voltage and the circuit is intended to stay powered on for a long time, then the device outlined in part 18 is good.

Feedback

Time spent on this project ~ 35hrs

Experienced challenges with linlabs remote servers crashing towards the end (no disk space etc.)
Both group and team based projects are good. We should have individual projects at the beginning of the semester and the final project should be group based as everyone would then have experience with Cadence on their own and wouldn't rely on their groupmate to finish the project completely.

It'd be nice to do a larger layout which would force us to use scripting.