Computer Engineering Program

College of Engineering and Computer Science

Spring 2020

Final Project: Single Bit Comparator

EGCP 441 – 05/10

Prepared by: Levi Randall and James Samawi

Abstract

In modern day logic, one of the best ways to improve performance is scaling down logic to a smaller size. This offers less power consumption, better switching speed, and less resources being consumed in production. Our design for the 1-bit digital comparator was based around minimal power consumption and minimal size. All circuit creation and testing was done through Ledit and LTSpice.

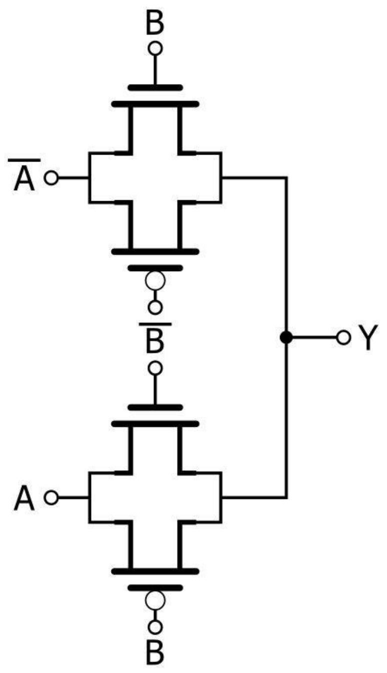
Design Approach

In order to accomplish our goals, we focused on minimizing the number of transistors the comparator/XNOR circuit required. At first, we found an XNOR circuit that used two inverter gates, two NAND gates, and a single NOR gate which together consists of 16 transistors. We used this as a foundation and kept reducing the number of transistors used.

At first, the obvious method to reduce the number of transistors is to use dynamic logic instead of static logic. This guarantees a reduction of transistors in most circuits; the reason being that only two transistors (that are controlled by the clock) are used to invert the circuit instead of typically doubling the number of transistors. This dynamic logic method comes with many complexities, such as its implementation as a layout in the Ledit program and the use of the clock to invert, so we decided to try something else.

We ran across a method that uses 10 transistors which used a “pull-up and pull-down network” technique ([¬A+¬B][A+B] and AB+¬A¬B, respectively) in order to create the XNOR CMOS equivalent. This consisted of 4 transistors for pull-up, 4 for pull-down, and 2 for the final inverter on the output line in the middle. This is much easier to implement on a layout compared to the dynamic logic method. But we can still do better.

Research has led us to a technique using “transmission gates” that is used to create analog CMOS logic to which we can use by observing the given comparator truth table. 4 transistors are required for the transmission gate and 4 are used to make the inverters for “A” and “B” inputs. This technique is used for analog CMOS designs, but since the input voltage signal is a step voltage that is either 5v or 0v, it is safe to assume it will work for this digital system too. The benefit of using the transmission gate is the efficient use of space, as seen in figure 1(a) below, compared to a digital CMOS design. It is also worth noting that a digital design of the XOR also has 4 transistors. In figure 1(b) below, we created the circuit below which would then be implemented in Ledit.



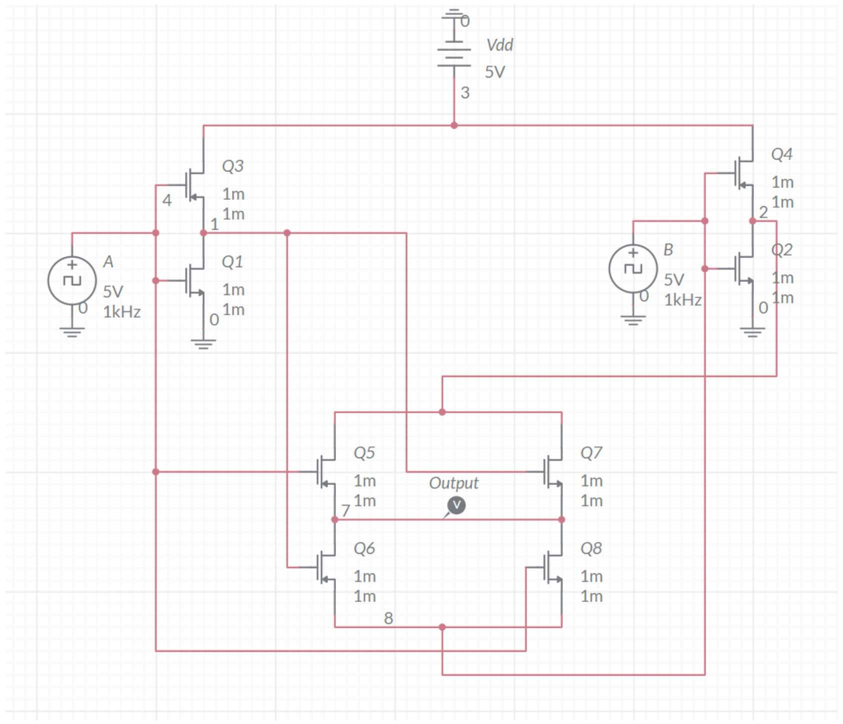


Figure 1(a) analog CMOS XOR transmission gate

Figure 1(b) digital CMOS XNOR Circuit Layout

Explanation of Theory

In figure 1(b), the upper left two transistors represent the inverter for signal A, and the upper right transistors are the inverter for signal B. Both the non-inverted and inverted A are fed into the gates of the transistor network on the bottom. And similarly, for B, both inverted and non-inverted are fed into the transistor network in the places of drains and sources of the transistor network on the bottom of our circuit.

When the signal for A is logic ‘0,’ the top two MOSFET are ON letting B’ through to the output and turning the bottom two MOSFETS off, meaning if both B and A are logic ‘0’ output is ‘1.’ Likewise, on the bottom, if A is logic ‘1,’ the bottom set of MOSFETS are ON and let through B, meaning if both A and B are logic ‘1’ output is ‘1’ as shown by *Table 1*. All other scenarios result in logic ‘0’ for the output, perfectly mimicking an XNOR gate in behavior.

A	B	F
0	0	**1**
0	1	**0**
1	0	**0**
1	1	**1**

Table 1

After implementing the circuit into Ledit, we then moved to simulating our circuit in LTspice by programming the two circuits in Fig 2. And Fig 3. below. The waveform from the simulation will allow us to determine if the layout behaves as an XNOR gate.

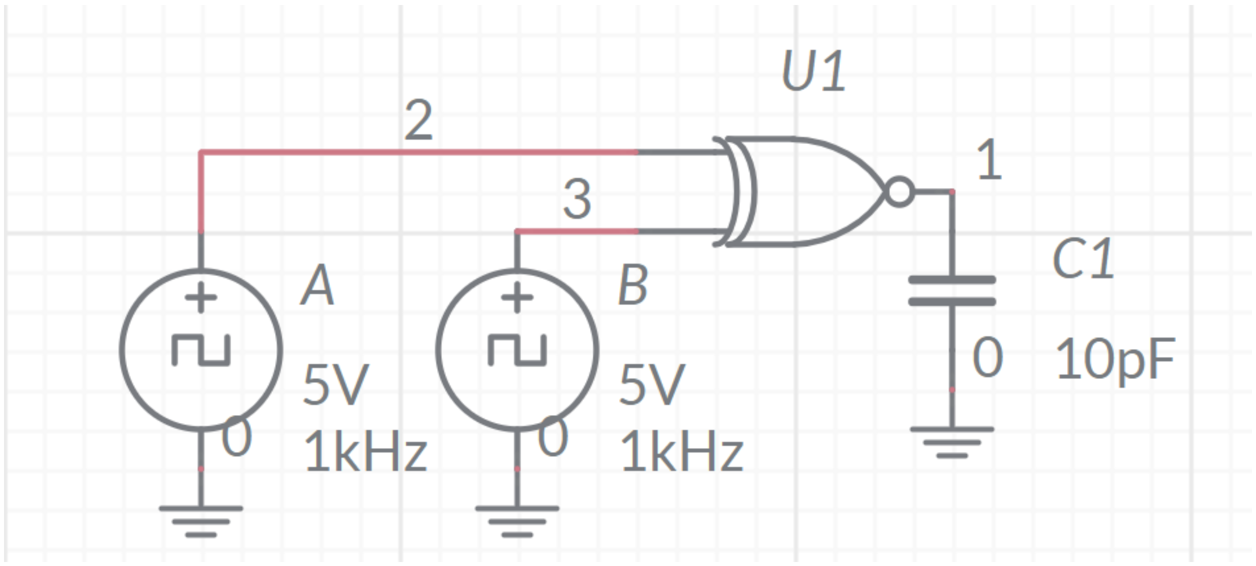


Figure 2 XNOR Circuit w/ 10pf Cap

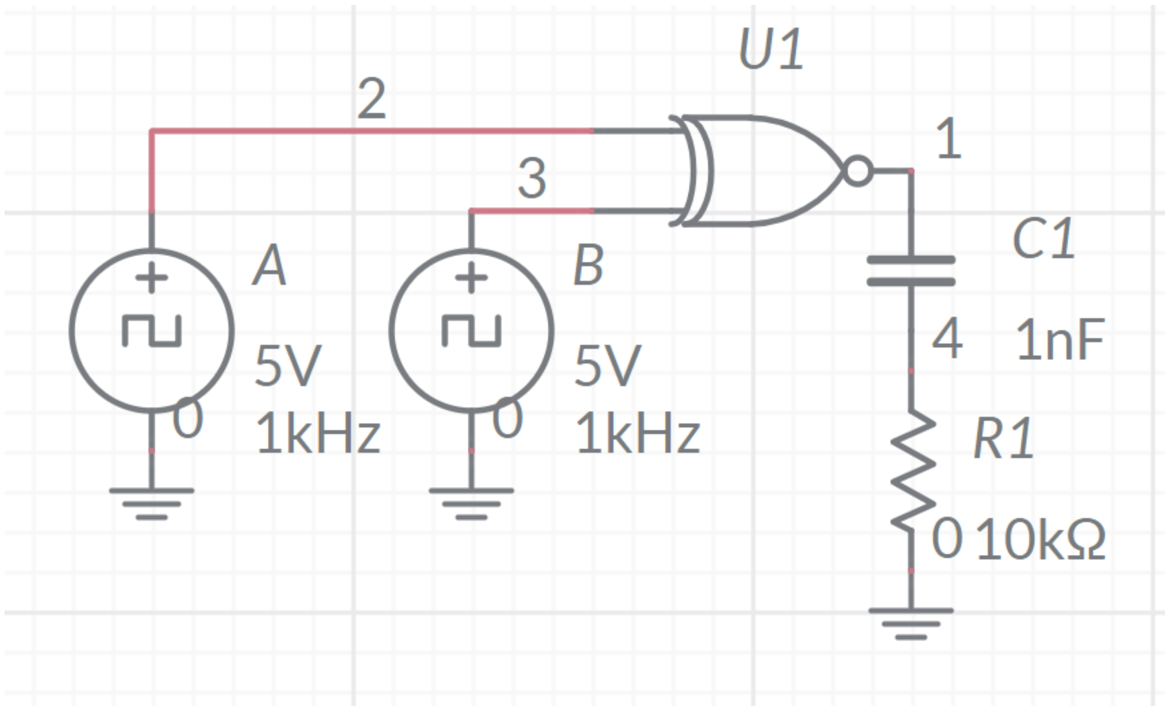


Figure 3 XNOR Circuit w/ 1nf Cap and 10kΩ Load

Testing and Results

Firstly, we modified the circuits above and are not using the correct valued capacitor. When using the 10pF capacitor, we found that our results were hard to distinguish as shown by Fig 4. below.

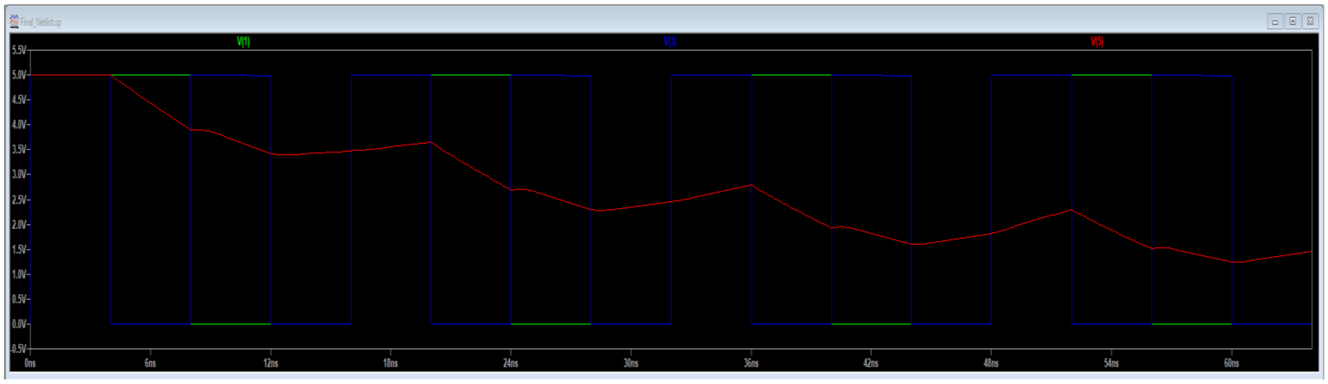


Figure 4 Wave Simulation of XNOR Output, w/ 10pF Cap

To combat this, all further measurements, until the load power measuring later, are done with a 10fF capacitor in place of the 10pF as it gave much clearer performance of the circuit.

We tested our circuit with many various square wave input signals and measured the load across our capacitors. The first two measurements were done with a peak voltage of 5V and a low of 0V, where the only variable was the rise and fall times of the input signals. Later, we lowered the peak voltage and compared it to the original. the first test is shown below in Fig 5.

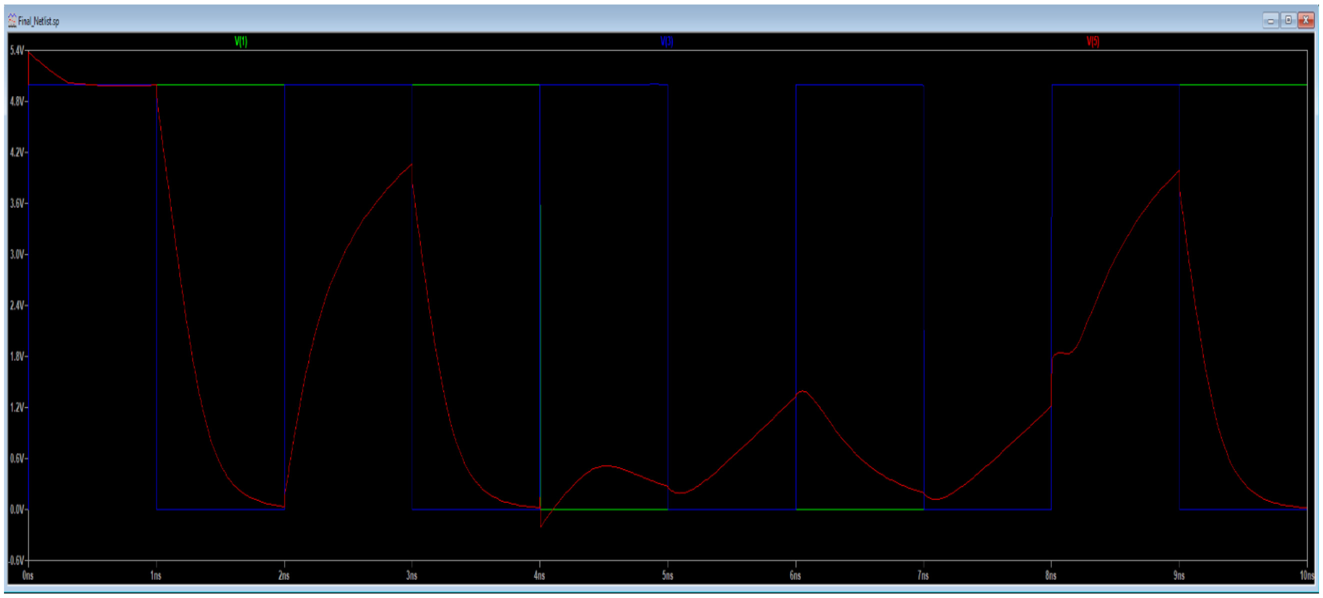


Figure 5 Wave Simulation of XNOR Output, Short Wave

First to give some clarity, the blue wave is representative of our input B which oscillates every 2ns while the green wave is our input A and has an period of 4ns. Lastly, the red wave is our output and should only be high or low when both our input is either high or low.

Now, the most noticeable attribute of this graph is the rise time of the output. Most consider the output to only be considered logic high when the output is at least 50% of its peak voltage. Unfortunately, our rise time is so slow that it only rises to about 25% of the original peak voltage in this time. So, for this circuit to work properly we changed the rise and fall time of the input signals and discover a much better result in the graphs below in Fig 6. by changing the rise and fall times to be 4 times longer than the first set.

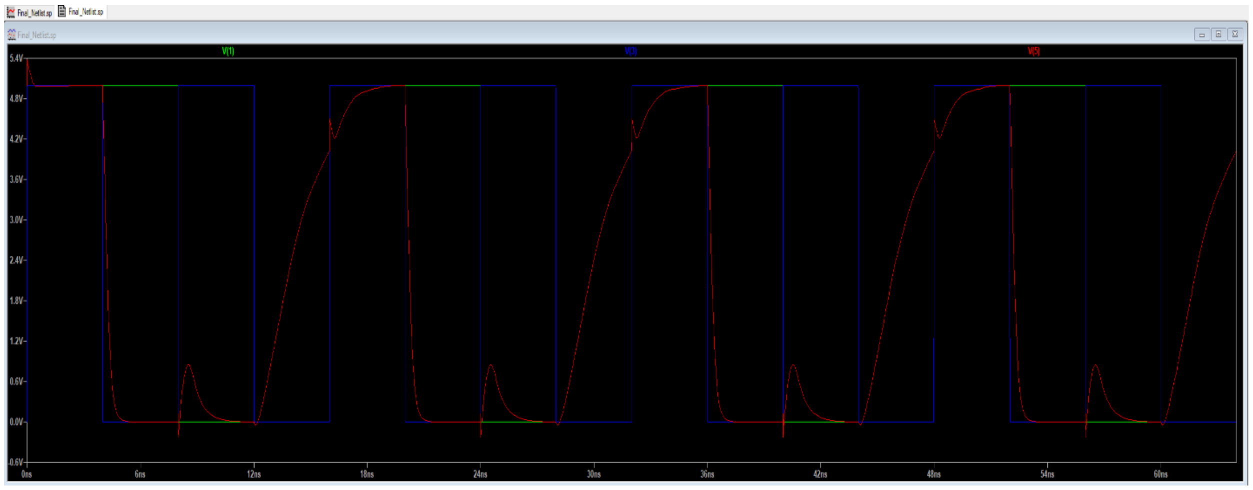


Figure 6 Wave Simulation of XNOR Output, Long Wave

Some other noteworthy occurrences happen during this waveform. First and most important, the circuit has a minor glitch when changing from the input of “01” to “10” where it believes both inputs are 0. This occurs in the very small picosecond gap that it takes for the graph to rise as seen in Fig 7.

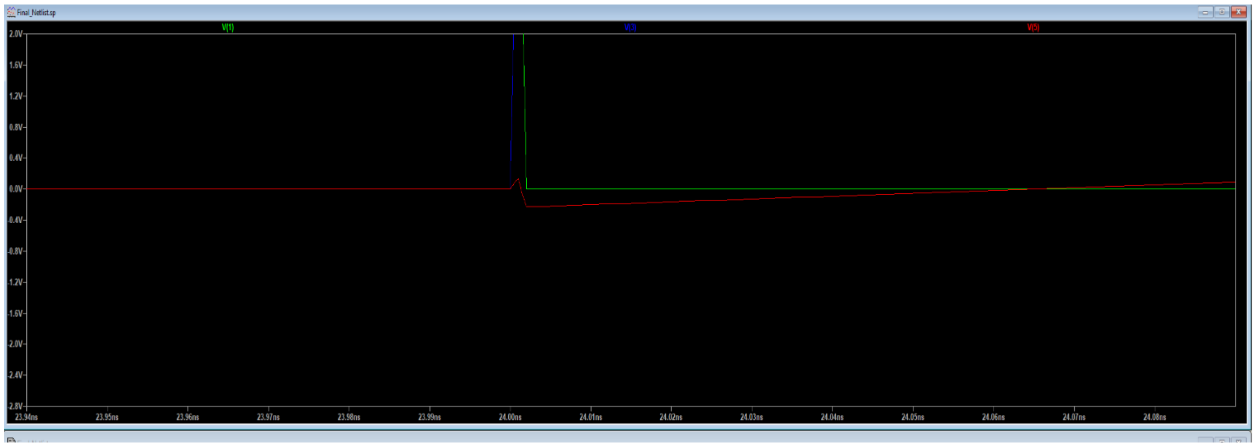


Figure 7 Wave Simulation of XNOR Output, Long Wave, Circuit Hiccup

This in turn creates a spike in voltage for the output right during the switch of the signal, enlarged in Fig 8.

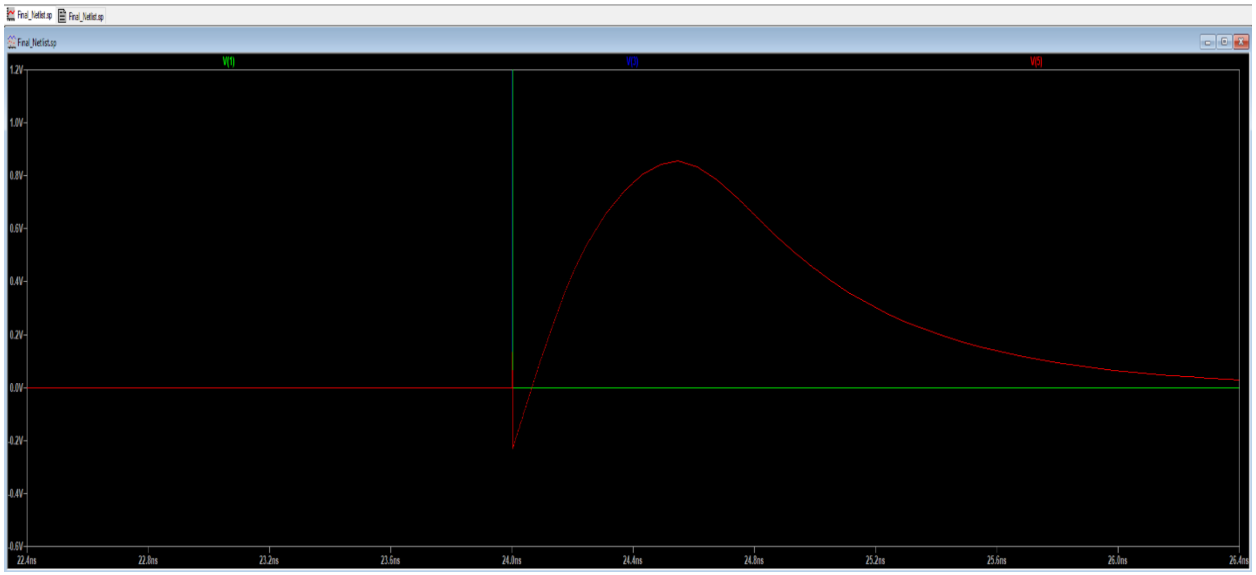
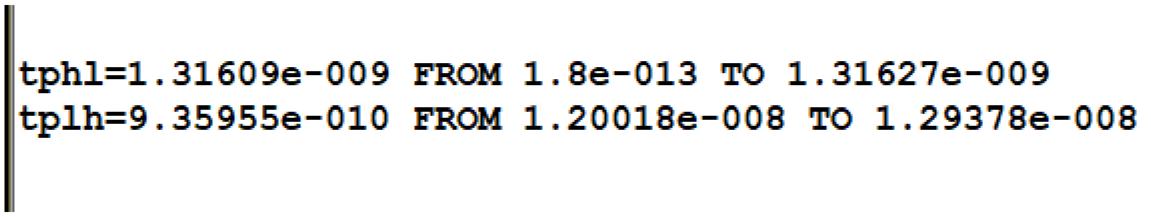


Figure 8 Wave Simulation of XNOR Output, Long Wave, Circuit Hiccup

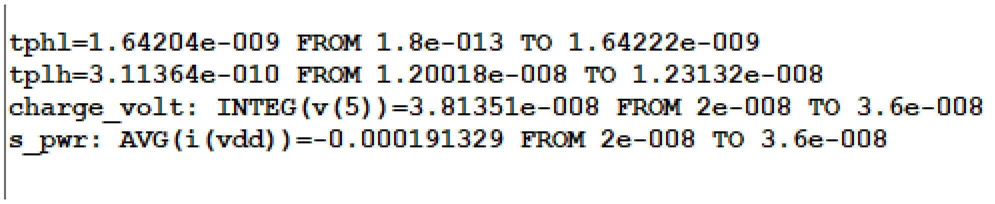
Moving forward, we found the worst case rise and fall time by shutting off one of the inputs and measuring the respective parameters. The result of those are shown below, Fig 9.



```
tphl=1.31609e-009 FROM 1.8e-013 TO 1.31627e-009
tplh=9.35955e-010 FROM 1.20018e-008 TO 1.29378e-008
```

Figure 9 Worst Case Rise and Fall Time

Then, we measured the voltage across the capacitor and the average power sent from the power supply to measure the output resistance, using Ohm's Law.



```
tphl=1.64204e-009 FROM 1.8e-013 TO 1.64222e-009
tplh=3.11364e-010 FROM 1.20018e-008 TO 1.23132e-008
charge_volt: INTEG(v(5))=3.81351e-008 FROM 2e-008 TO 3.6e-008
s_pwr: AVG(i(vdd))=-0.000191329 FROM 2e-008 TO 3.6e-008
```

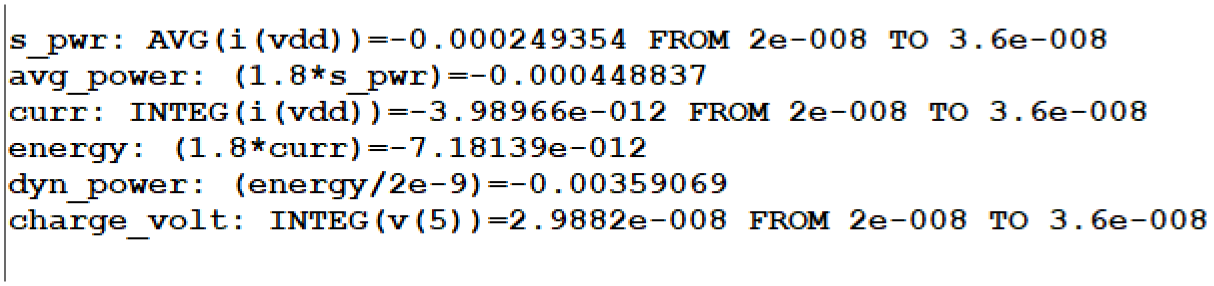
Figure 10 Average Rise and Fall Time w/ Avg Current and Voltage

Ohm's Law:

$$V = IR$$

$$R_{out} = \frac{V}{I} = \frac{3.81351}{0.000191329} = 19931.6\Omega$$

Next, we measured the switching energy, Fig 11.



```
s_pwr: AVG(i(vdd))=-0.000249354 FROM 2e-008 TO 3.6e-008
avg_power: (1.8*s_pwr)=-0.000448837
curr: INTEG(i(vdd))=-3.98966e-012 FROM 2e-008 TO 3.6e-008
energy: (1.8*curr)=-7.18139e-012
dyn_power: (energy/2e-9)=-0.00359069
charge_volt: INTEG(v(5))=2.9882e-008 FROM 2e-008 TO 3.6e-008
```

Figure 11 Switching Energy

Lastly, to measure the input Capacitance, we used a method from our lab 4, by measuring the delay and resistance as a function of Tao. Additionally, the circuit is parallel regardless of input, therefore Cin1 = Cin2. The circuit and equation below models the process we used,

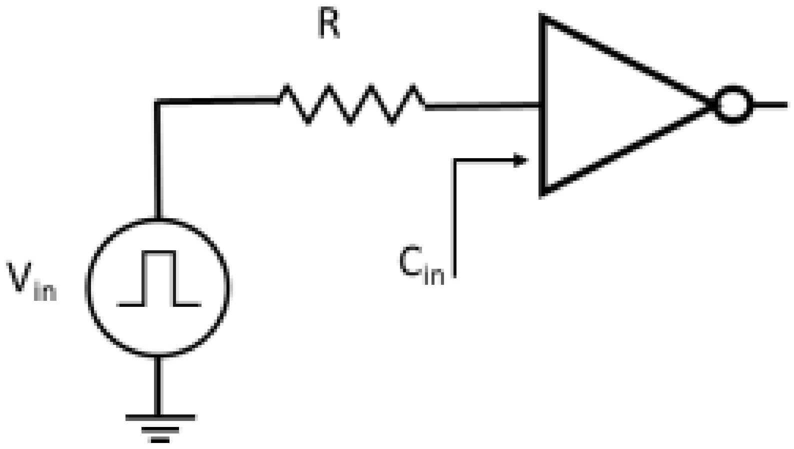
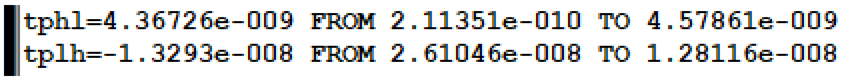


Figure 12 Input Capacitance



```
tphl=4.36726e-009 FROM 2.11351e-010 TO 4.57861e-009
tplh=-1.3293e-008 FROM 2.61046e-008 TO 1.28116e-008
```

Figure 13 Rise and Fall Time for Input Capacitance

$$T = \frac{(T_{PHL} + T_{PLH})}{2} = \frac{4.367 * 10^{-9} - 0.13293 * 10^{-9}}{2} = 2.117035 \times 10^{-9} ns$$

$$\tau = 0.69 * R * Cin$$

$$Cin1 = Cin2 = \frac{\tau}{0.69 * R} = \frac{2.117035 * 10^{-9}}{0.69 * 10 * 10^{3}} = 3.0681e - 13$$

We also included the layout dimensions of our circuit below and included a picture of the design of our circuit in Fig 14.

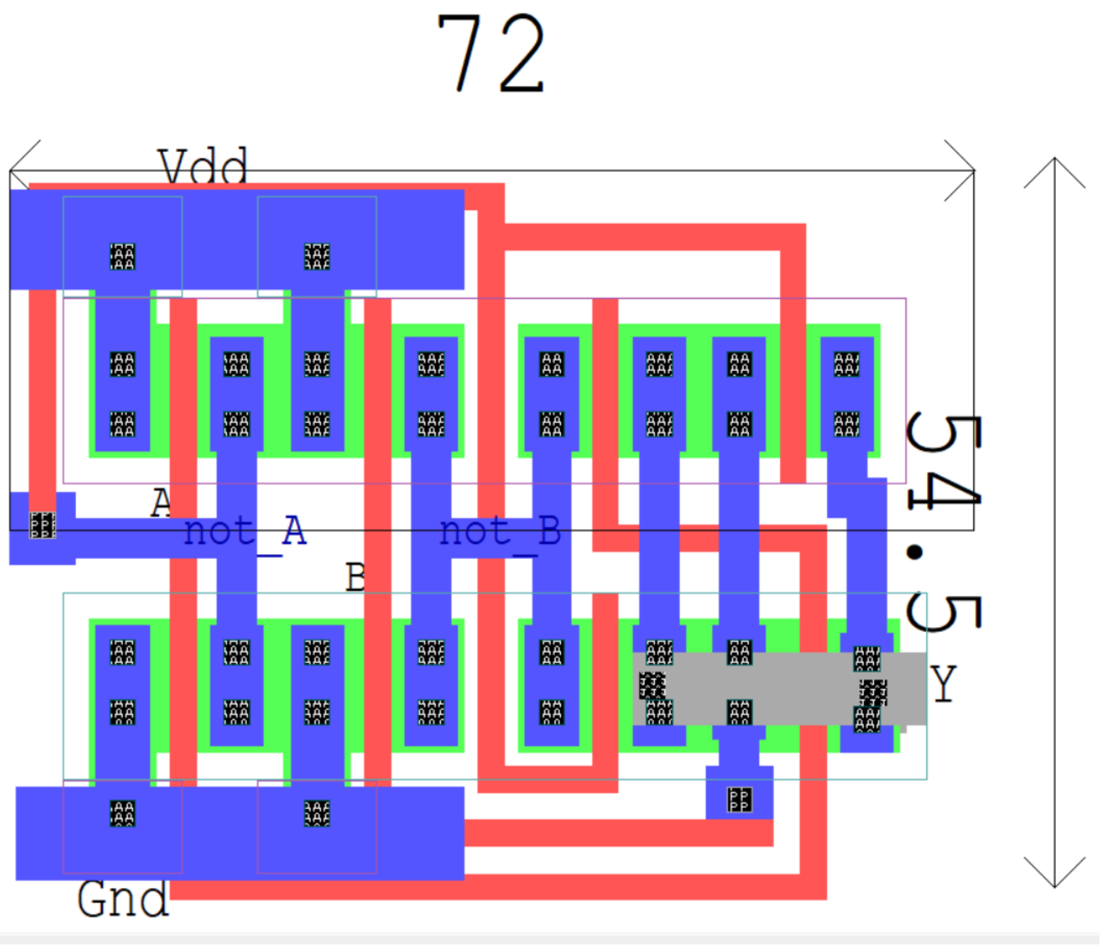


Figure 14 Ledit Design of Circuit

The area of the layout can be calculated by multiplying X = 72λ and Y = 54.5λ which is 3924λ*λ.

Important Param.	Value
CIN1	3.0681e-13
CIN2	3.0681e-13
Rout	19931.6Ω
Switching Energy	-7.18139e-12(W)
Layout Area	3924(λ·λ)
Delay(Rise/Fall)	3.1e-10/1.6e-9(s)

Conclusion

The circuit behaved appropriately but not with the numbers given by the instructor. It was able to function properly with a slower clock cycle and a lower output capacitor. So, this circuit may be good for low power IC sensors but is not the strongest circuit, which is the trade off for the very tiny switching energy of -7.181e-12 Watts. It also has a small delay with an average delay of 2.35ns for Rise and Fall, which is in my experience a very good performance. In all, an alternative, energy conserving, and fast circuit was shown.

Dynamic logic could possibly lower the number of transistors used, depending on how it is implemented into a layout. But given the circumstances, we tried our best and pushed to an extent the limitations of static CMOS design and received great results.

HOURS SPENT	Levi Randall	James Samawi
Researching/Designing Schematic	3	4
Writing Report	5	3
Coding	3	2
Modeling/Creation	2	4