|  |
| --- |
|  |
| Analysis, Design, and Verification of a BJT NAND Gate |
| Digital Electronics (EEE3310.01) Spring 2019 |

|  |
| --- |
| Peter A. Dranishnikov and Jackson Purvis    ID: U0000005258 ID: U0000005324  4-24-2019  Professor: Dr. Zahra Amjadi |
|  |

Table of Contents:

[Table of Figures: 2](#_Toc7007634)

[Abstract: 3](#_Toc7007635)

[Introduction: 3](#_Toc7007636)

[Analysis: 3](#_Toc7007637)

[Results: 7](#_Toc7007638)

[Discussion: 13](#_Toc7007639)

[Conclusion: 13](#_Toc7007640)

[Appendix A: MATLAB Source code 15](#_Toc7007641)

[Appendix B: SPICE Model 16](#_Toc7007642)

[References 16](#_Toc7007643)

# Table of Figures:

[**Figure 1.** This diagram shows the NAND gate we modeled in MultiSim based on our design calculations. 6](#_Toc7007571)

[**Figure 2.** This image shows the various internal currents and the output current/voltage when the NAND gate’s input is 00. 7](https://flpoly-my.sharepoint.com/personal/pdranishnikov5258_floridapoly_edu/Documents/digel/project/project_BJT_NAND_gate_thesis.docx#_Toc7007572)

[**Figure 3.** This image shows the various internal currents and the output current/voltage when the NAND gate’s input is 11. 8](#_Toc7007573)

[**Figure 4.** In this image, the red line represents input A, the blue line represents input B, and the purple line represents the output. 9](#_Toc7007574)

[**Figure 5.** This image shows the current measured at the points labeled as XCP1 and XCP2 in Figure 1. 10](#_Toc7007575)

[**Figure 6.** This image shows the high to low propagation delay time. 11](#_Toc7007576)

[**Figure 7.** This image shows the low to high propagation delay time. 12](#_Toc7007577)

# Abstract:

The goal of this project is to design, model, and test a high-speed TTL NAND gate. This objective shall be accomplished utilizing such tools as MultiSim, physical transistors and resistors, and various literatures including datasheets and textbooks. This project shall involve calculating internal resistance and current values based on predetermined initial values such as specific voltage drops and current gains. After all calculations are finalized, the computed values shall be used to model a device in MultiSim, before physically constructing the device as a real-world circuit. The model and the circuit shall both be tested, and their final performances will be compared and analyzed. These tests will demonstrate the veracity of our calculations and the success of this project.

# Introduction:

We have chosen the high-speed TTL NAND gate as the subject of our project. We made this decision because of the useful advantages this device offers. For example, high-speed TTL gates have extremely short propagation delay times, as well as very high fan-outs. Our TTL NAND gate is composed of seven PN2222A BJT transistors. We chose this model because it is one of, if not the, most widely used BJTs in the industry. This high-speed NAND gate is also made up of six different resistors. In order to find the values for these resistors, we had to perform a series of calculations based a collection of initial parameters taken from the PN2222A’s datasheet.

# Analysis:

In order to design our NAND gate, we had to determine a number of parameters for the device to operate under. Most of these initial conditions were taken from the datasheet for the PN2222A BJT. Several were also taken from the MultiSim datasheet for the PN2222A datasheet. Finally, VCC, VOH, N, and IN1 were chosen by us as our intended outputs. These values are listed below:

We performed our calculations using MATLAB. The program we utilized can be seen in Appendix A. Our methodology is displayed below.

First, we calculated the values for R1 and R2:

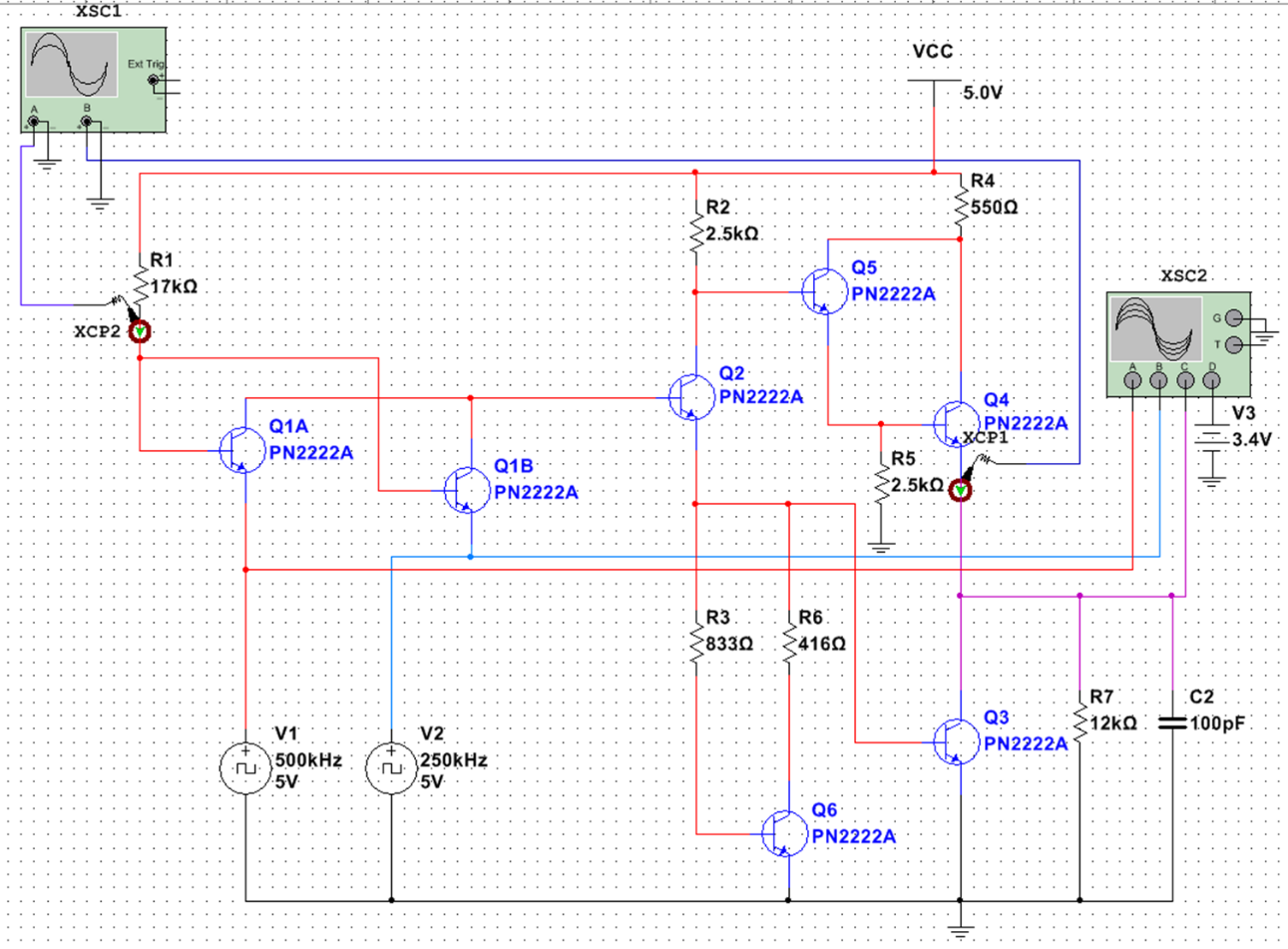
Next, we calculated a series of currents for high and low outputs. These calculations eventually led to the value for RB:

After this, we calculated more currents, which led us to the values of R4 and R5:

Then, we used the value of RB to find the values for R3 and R6:

Finally, we found the remaining values, including the load capacitance, the time delay, the input voltages, and the noise margins:

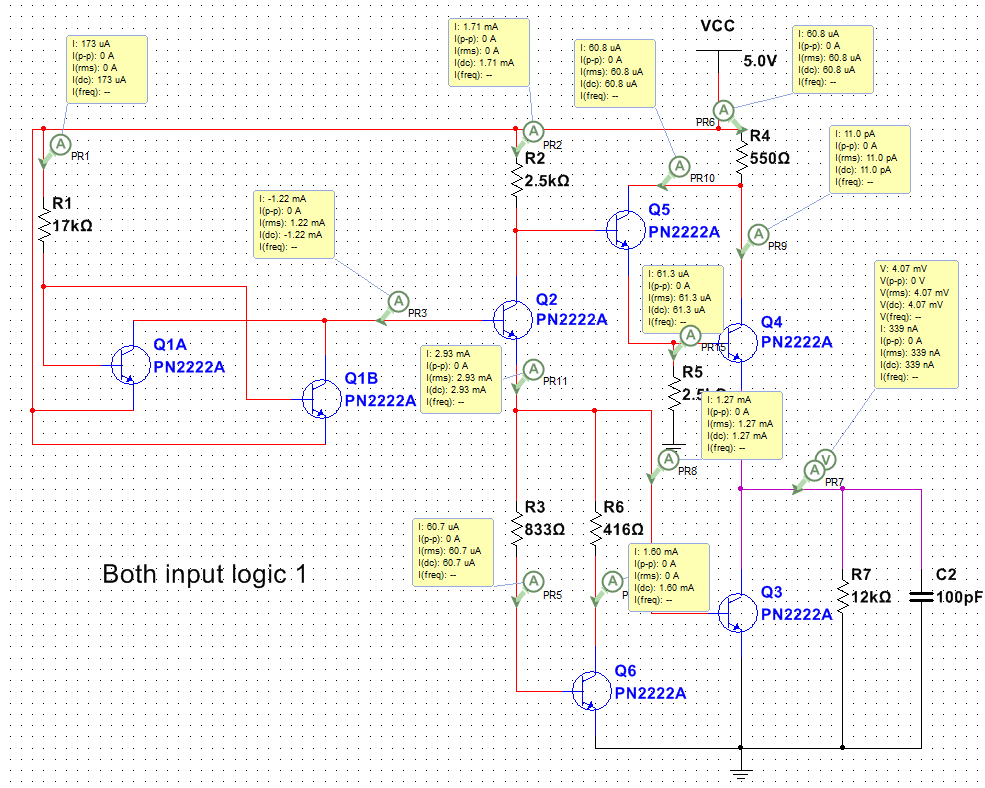
After finding these values, we proceeded to design the circuit in MultiSim. For this process, we simply drew up the circuit schematic in the software based the values we calculated above. The MultiSim circuit can be seen in the figure below.



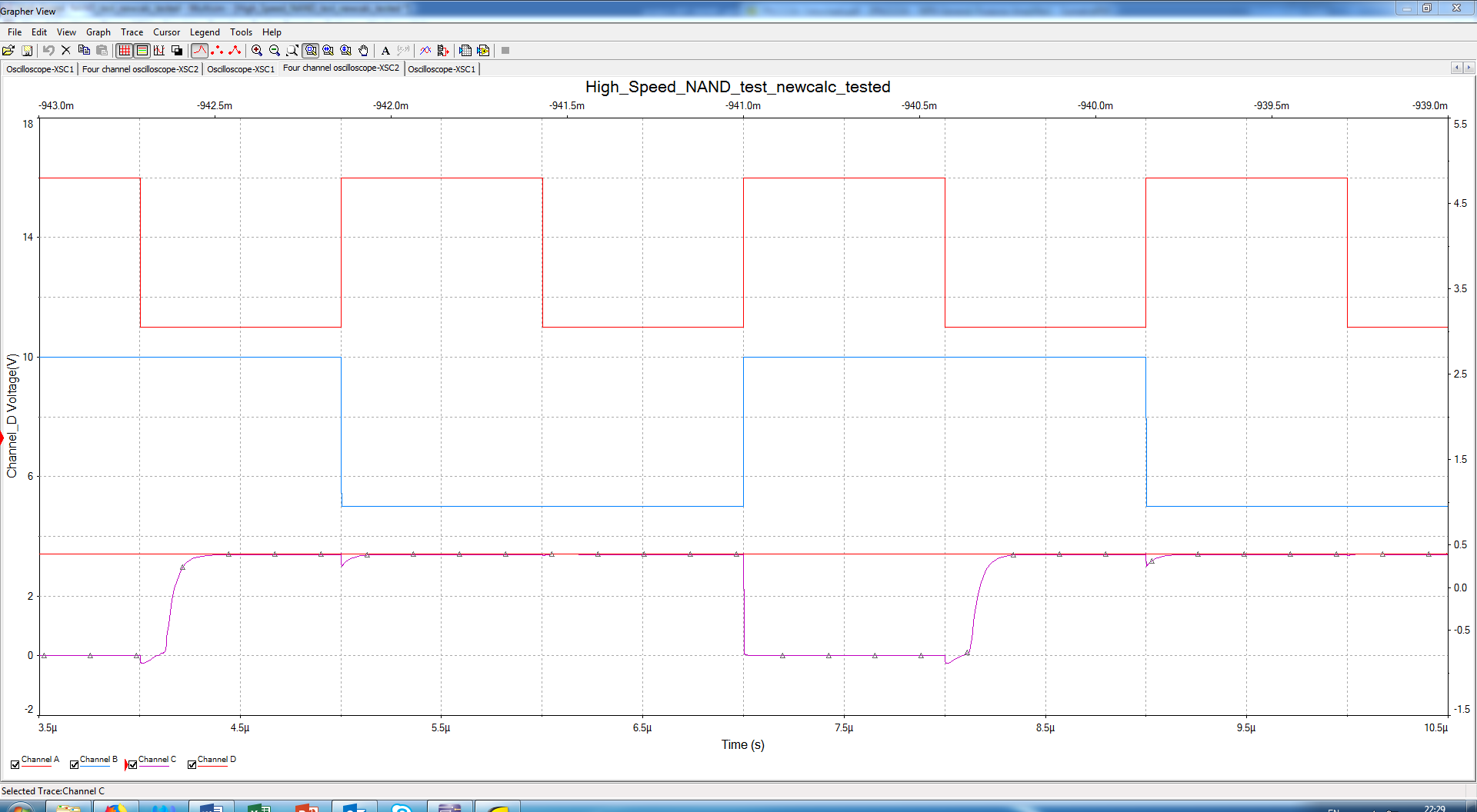
**Figure 1.** This diagram shows the NAND gate we modeled in MultiSim based on our design calculations.

# Results:

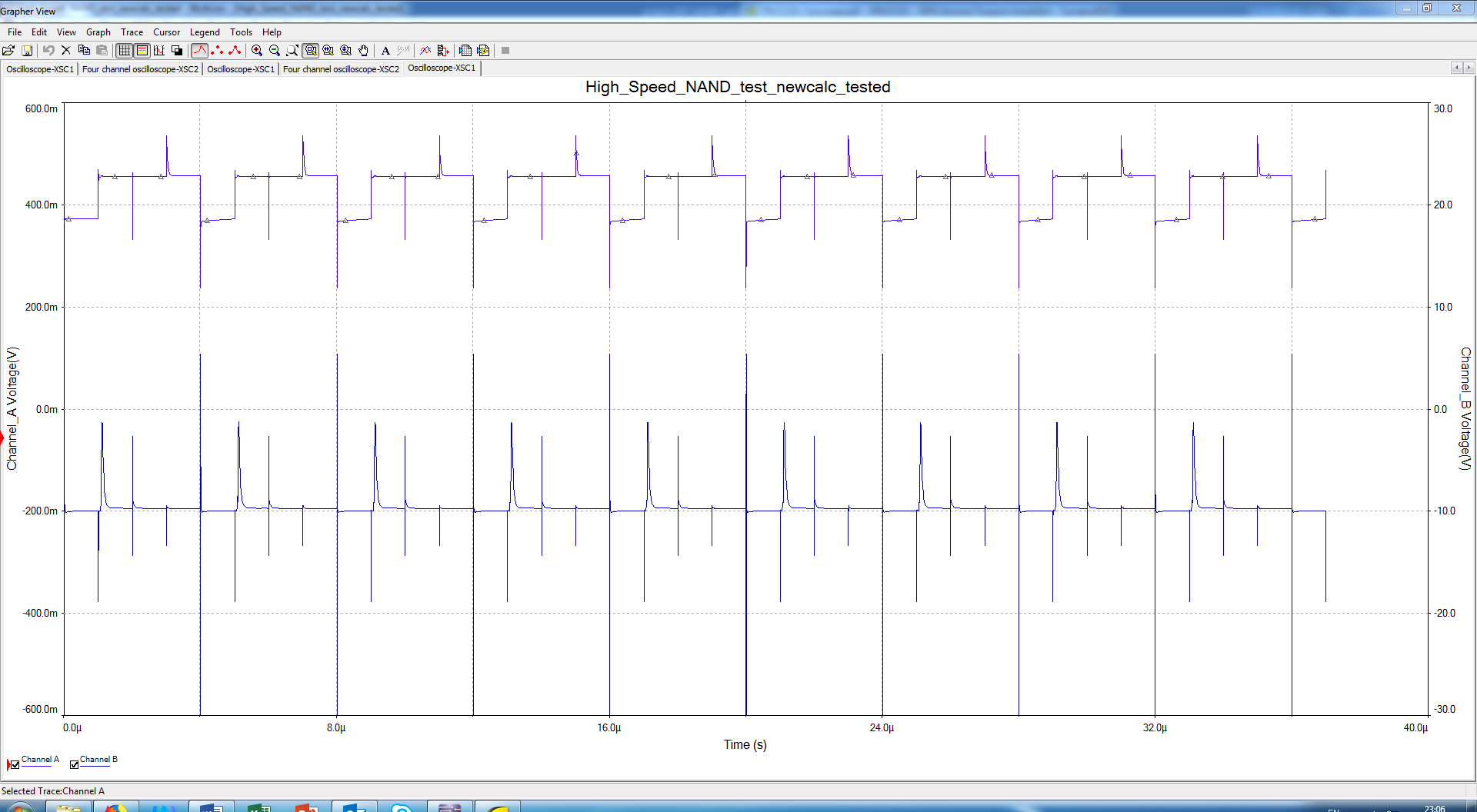
**Figure 2.** This image shows the various internal currents and the output current/voltage when the NAND gate’s input is 00.

****

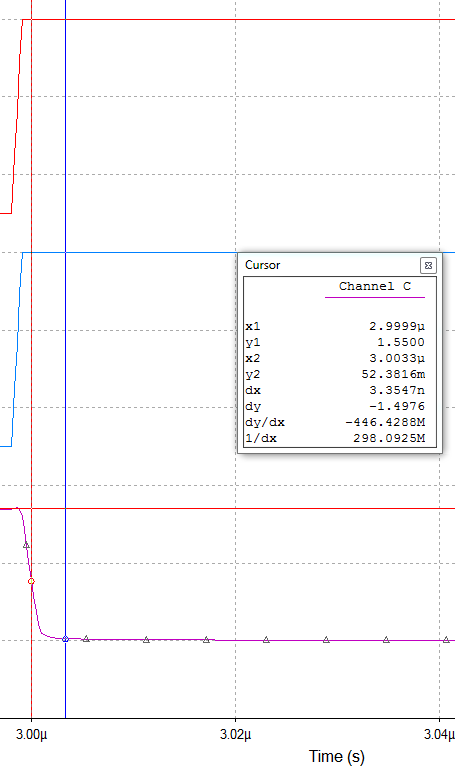
**Figure 3.** This image shows the various internal currents and the output current/voltage when the NAND gate’s input is 11.

****

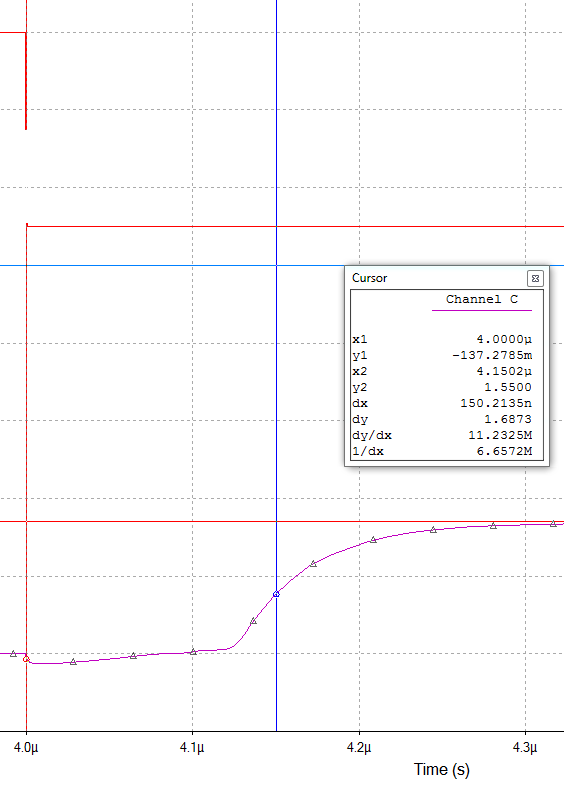
**Figure 4.** In this image, the red line represents input A, the blue line represents input B, and the purple line represents the output.

****

**Figure 5.** This image shows the current measured at the points labeled as XCP1 and XCP2 in Figure 1.

****

**Figure 6.** This image shows the high to low propagation delay time.

****

**Figure 7.** This image shows the low to high propagation delay time.

# Discussion:

The figures above display all the results we observed from our testing process in MultiSim. Figure 2 shows the state of the high-speed NAND gate when the inputs are both logic low. This, of course, leads to a high output. As our simulation shows, the output voltage is 3.39 V; this is equivalent to our predetermined VOH value of 3.4 V. In other words, the output successfully passed the threshold for a logic high output. This means the device operated correctly in Mode 2.

Figure 3 shows the state of the device when both inputs are logic high. This means the device is operating in Mode 1, and the output should be logic low. As you can see in the figure, our simulated results produced an output voltage of 4.07 mV. This well under our VOL value of 300 mV. Therefore, the device was proven to be operating successfully in this mode as well.

Figure 4 shows the output produced when we used a virtual function generator to test all possible inputs with our NAND gate. These inputs are 00, 01, 10, and 11. Every input produced a high output, except for 11. This entry generated a low output. Once again, this is exactly how a well-designed NAND gate should behave.

Figure 5 shows the combined base current (top) of the input transistors and the emitter current (bottom) of transistor Q4. The “voltage” is generated by the current clamps, both set to 1mA/V. The combined base current is relatively stable, with some sinking due to the fan-in current provided to the voltage source (for a real design, other TTL chips). The output current swings greatly on the output transition due to the base recovery circuit and Darlington pair switching.

Figure 6 shows the high to low propagation delay time. This diagram illustrates how long it takes for the device to transfer from output high state to output low state. As the measurements in the diagram illustrate, this transfer time meshes quite nicely with the value we found in our calculations.

Figure 7 shows the low to high propagation delay time. This diagram illustrates how long it takes for the device to transfer from output low state to output high state. As you can see from the measurements in the diagram, it takes less than 300 nanoseconds for the change of state to occur. This is exactly what our calculations predicted.

# Conclusion:

The high speed BJT NAND gate designed worked as expected in the simulation and in the physical circuit at static and high-frequency input up to 500khz. This result indicates that the manual and the computer-assisted computations are correct. However, this design has room for improvement in several ways. Of one, we can target a higher current to yield a greater βF and subsequently kODF, but at the cost of a higher-power design which scales poorly with transistor count. Another improvement would be to target lower currents for a scalable design, but the transistor chosen may not reliably drive the fan-out or transistors themselves. A final design modification would be to include Schottky diodes between the base and the collector to prevent the transistor from entering saturation. This modification comes at a significantly increased bill of materials cost. The current design achieves a balance in component cost, power, and fan-out.

# Appendix A: MATLAB Source code

%parameters from multisim

v\_cc = 5;

v\_be = 0.75;

v\_bc = 0.75;

i\_cmax = 1e-3;

v\_ce = 0.3; %saturated

v\_ol = v\_ce;

%v\_d = 0.75;

v\_oh = 3.4;

i\_n1 = 1e-3;

n = 4;

beta\_f = 100;

beta\_r = 6.092;

c\_f = 25e-12;

t\_smax = 225e-9;

t\_dmax = 10e-9;

%params from experimental verification

%v\_be = 0.677;

%v\_bc = ;

%v\_cesat = 8.1e-3;

%beta\_f = 100;

%computation steps

r\_1 = beta\_r \* ((v\_cc - 2\*v\_be - v\_bc)/i\_n1);

r\_2 = ((1+beta\_f)\*(v\_cc - 2\*v\_be - v\_oh)/(n\*i\_n1));

i\_b1 = ((v\_cc - 2\*v\_be - v\_bc)/r\_1); %output "low"

i\_no = beta\_r\*((v\_cc - v\_ce - v\_be)/r\_1); %output "low"

i\_b2 = (1 + beta\_r)\*i\_b1; %output "high"

i\_c2 = ((v\_cc - v\_be - v\_ce)/r\_2); %output "high"

i\_e2 = i\_c2 + i\_b2; %output "high"

i\_r = 0.4 \* i\_e2; %output "high"

r\_b = v\_be/i\_r; %while there's technically no resistor base, this is the equivalent

%resistance to the base recovery circuit

i\_b3 = i\_e2 - i\_r; %output "high"

i\_c3max = beta\_f \* i\_b3;

i\_Lo = n \* i\_no; %output "low"

isDesignValid = i\_c3max > i\_Lo; %if true, design can support the fanout current

k\_ODF = i\_c3max / i\_Lo; %if large, then can support much more fanout quantity

r\_4 = ((v\_cc - v\_oh - v\_ce - v\_be)/i\_cmax);

r\_5 = ((v\_oh + v\_be)/i\_b3);

i\_br = v\_be/r\_b;

r\_6 = ((v\_be - v\_ce)/(i\_br/(1+1/beta\_f)));

r\_3 = 2 \* r\_6;

%prop delay calcs

r\_l = v\_oh / (n\*i\_n1);

c\_l = n \* c\_f;

tao\_r = (r\_4 + r\_l) \* c\_l;

vo\_mid = (v\_oh - v\_ol)/2;

t\_r = tao\_r \* log((v\_cc - v\_ol)/(v\_cc - vo\_mid));%matlab log is eq. to ln()

tao\_f = r\_l \* c\_l;

t\_f = tao\_f \* log((2 \* v\_oh)/(v\_oh + v\_ol));

t\_plh = t\_smax + t\_r;

t\_phl = t\_dmax + t\_f;

t\_pd = (t\_plh + t\_phl)/2;

%noise margin calcs

v\_il = (2\*v\_be + v\_bc) - v\_be;

v\_ih = 2\*v\_be + v\_bc;

nm\_h = v\_oh - v\_ih;

nm\_l = v\_il - v\_ol;

# Appendix B: SPICE Model

.MODEL PN2222 NPN(Is=14.34f Xti=3 Eg=1.11 Vaf=74.03 Bf=255.9 Ne=1.307

+ Ise=14.34f Ikf=.2847 Xtb=1.5 Br=6.092 Nc=2 Isc=0 Ikr=0 Rc=1

+ Cjc=7.306p Mjc=.3416 Vjc=.75 Fc=.5 Cje=22.01p Mje=.377 Vje=.75

+ Tr=46.91n Tf=411.1p Itf=.6 Vtf=1.7 Xtf=3 Rb=10)

# References

[1] M. Rashid, *Microelectronic Circuits: Analysis and Design*, 3rd ed. Boston, MA. Cengage Learning, 2017.

[2] Fairchild, “PN2222A NPN General-Purpose Amplifier”, July 2014.

[3] National Semiconductor, *MultiSim*, “PN2222A”, April 2019