

DESIGN AND ANALYSIS OF N-PATH FILTER FOR RADIO FREQUENCIES

A Thesis

presented to

the Faculty of California Polytechnic State University,

San Luis Obispo

In Partial Fulfillment

of the Requirements for the Degree

Master of Science in Electrical Engineering

by

Nathan Bergen

June 2022

© 2022

Nathan Bergen

ALL RIGHTS RESERVED

## COMMITTEE MEMBERSHIP

TITLE: Design and Analysis of N-path Filter for Radio  
Frequencies

AUTHOR: Nathan Bergen

DATE SUBMITTED: June 2022

COMMITTEE CHAIR: William Ahlgren, Ph.D.  
Associate Professor of Electrical Engineering

COMMITTEE MEMBER: Jane Zhang, Ph.D.  
Professor of Electrical Engineering

COMMITTEE MEMBER: Dennis Derickson, Ph.D.  
Professor of Electrical Engineering

## ABSTRACT

### Design and Analysis of N-path Filter for Radio Frequencies

Nathan Bergen

Due to the growth of wireless communication many communication frequencies have grown increasingly dense. This density requires higher Q-factor to receive only the signal of interest. With the rise of smaller integrated circuits previous solutions used for filtering have become viable again. This paper explores whether the N-path filter is viable in the modern day for radio frequency receiver purposes.

A non-differential N-path filter was created by utilizing Cadence Virtuoso with a working center frequency range of 750MHz to 1GHz while using TSMC technology. The desired quality factor of over 1,000 was reached while maintaining a total area of 800 by 800 micrometers. Through the analysis of the N-path filter new techniques for mixed signal analysis were used for simulation. This included parametric analysis in Cadence ADE-L and additional analysis in MATLAB, and the addition of a bootstrapping circuit to decrease simulation time. Future applications regarding analyzing mixed signals could use these methods to provide frequency response data and automated processing.

## ACKNOWLEDGMENTS

I would like to extend my thanks to my family and friends who have supported me and encouraged me throughout this project. I am grateful for Chrissy as without her support and assistance I would not have made it as far as I have. I would also like to extend my thanks to my advisor Dr. Smilkstein for her feedback, assistance, and guidance throughout this project.

## TABLE OF CONTENTS

	Page
LIST OF TABLES .....	vii
LIST OF FIGURES.....	viii
CHAPTER	
1. INTRODUCTION.....	1
1.1 History.....	1
1.2 N-path Filter Operation .....	2
1.3 Requirements .....	9
2. DESIGN AND SETUP .....	10
2.1 Design and Justification.....	10
2.2 Testbench Setup .....	13
2.3 Test procedure .....	15
3. ANALYSIS AND RESULTS .....	16
3.1 Analysis .....	16
3.2 Size Analysis .....	20
3.3 Final Results .....	21
4. CONCLUSION .....	22
4.1 Future work.....	22
BIBLIOGRAPHY .....	24
APPENDICES	
A. MATLAB Code to generate N-path diagrams: .....	25
B. Parametric Analysis Example .....	27
C. Processed Data.....	28
D. MATLAB Code to Process Parametric Analysis .....	32

## LIST OF TABLES

Table	Page
1.1 Requirements for N-path Filter Design .....	9
3.1 Results for N-path Filter Design .....	21
C.1 Processed Data Frequency Response .....	28
C.2 Processed Data for Frequency Response to Various Clock and Vin Frequencies .....	30

## LIST OF FIGURES

Figure	Page
1.1 Early N-path Filter Design .....	1
1.2 Basic N-path Circuit Schematic .....	3
1.3 N-path Filter Single Capacitor Demonstration .....	3
1.4 All 8 Input Clocks .....	4
1.5 Input and Output Signal When Clock Matches Input Frequency .....	5
1.6 Input Frequency is Half of Clock Frequency .....	5
1.7 Input Frequency is Double of Clock Frequency .....	6
1.8 Input is White Noise .....	6
1.9 Block Diagram of N-path Filter .....	7
1.10 Lowpass to Bandpass Conversion by N-path Filter .....	8
2.1 Transmission Gate .....	10
2.2 Transmission Gate Inverter Chain .....	11
2.3 N-path Filter Full Schematic .....	12
2.4 N-path Filter Full Schematic Zoomed In .....	12
2.5 N-path Filter Symbol .....	13
2.6 N-path Filter Testbench .....	13
2.7 N-path Filter Bootstrap in Schematic .....	14
2.8 N-path Filter Simulation 1GHz Input without Bootstrap .....	14
2.9 N-path Filter Simulation 1GHz Input with Bootstrap .....	14
3.1 Steady State Transient Output .....	16
3.2 PMOS and NMOS Overlapping .....	17
3.3 Frequency Response for 1GHz Clock .....	17
3.4 Frequency Response about Center Frequency 1GHz .....	18
3.5 Frequency Response for Various Clock and Input Frequencies .....	19
3.6 Layout showing Size of N-path Filter .....	20
B.1 Example of Parametric Analysis Setup to Measure Frequency Response 2GHz Input Signal for a 1GHz Clock .....	27
B.2 Example of Parametric Analysis Setup to Measure Frequency Response 2GHz Input Signal for a 1GHz Clock .....	27



Chapter 1  
INTRODUCTION

1.1 History

The first design for an N-path filter was introduced by L. E. Franks and I. W. Sandberg in their 1960 paper entitled, "An Alternative Approach to the Realization of Network Transfer Functions: The N-path Filter". In the paper, they described both the math and design of an N-path filter. The main benefit of this circuit was that the center frequency was defined by an input clock instead of being set by circuit components [1]. However, at this time the N-path filter was not yet defined or explored for RF capabilities.

Figure 1.1 is an example of an early N-path filter from 1960. In this case they leave it as N amount of capacitors. The wheel would spin at a rate of  $1/T$  rotations per second and an external switch would be closed every  $N/T$  interval. This effectively created an N-path circuit for lower frequencies.

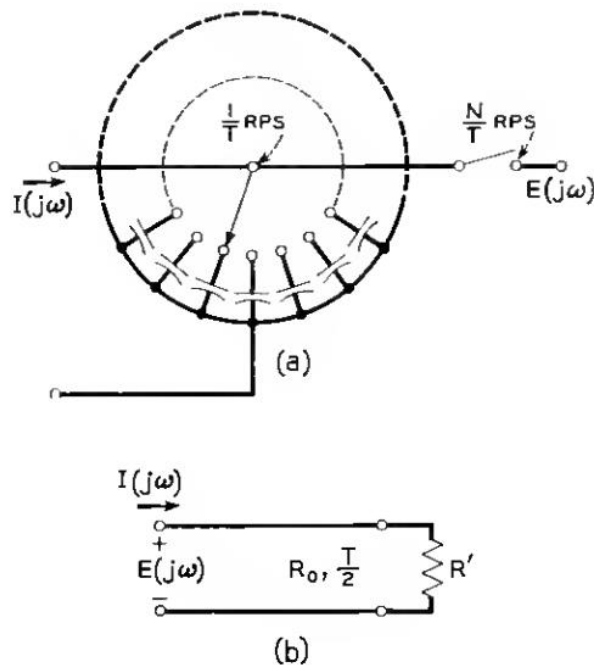


Figure 1.1: Early N-path Filter Design

The N-path filters began to die out in the 1980s due to the rise of popularity of digital filters. The N-path filters during this time were referred to as switch-capacitor filters and used for discrete time purposes. However, there has been a resurgence in the use of N-path filters for

continuous time systems [2]. One such design attempted to create RF CMOS based N-path filters, which ultimately lead to a N-path filter with a tunable center frequency of 240-530MHz using switched capacitors [3]. Recently N-path filters have been increasing in popularity because of the reduction of size in CMOS technology. The main purpose has been to create a large tunable range at increasingly higher frequencies some going up to 1.2GHz [4].

While the results of these filters have continued to increase in range the Q-factor of these filters have been from 100-300. The Q-factor of the of a bandpass filter is the measure of the sharpness of the frequency response. The higher the Q-factor the smaller of a range of frequencies that are passed through the filter [5]. RF communications today have been becoming increasingly dense, this density requires sharper filtering in order to not face interference between signals on the receiver end. To accommodate this need, the purpose of this project will be using capacitors to create a non-differential N-path filter using current CMOS technology to achieve a Q-factor of over 1000.

## 1.2 N-path Filter Operation

N-path filters operate on the basic level of capturing and storing the sections of the input signals onto N capacitors in sequence. These capacitors are connected to transmission gates that allow the signal to propagate when the clock signal for that capacitor goes high. When this happens the input voltage begins to charge the capacitor. If the input signal frequency matches that of the clock, then the capacitor will see the same section of the input signal at each period. This leads the capacitor to charge up to the average voltage of that section over the course of the RC constant of the system. Figure 1.2 shows the simplest model of the N-path filter. Each switch is turned on then off sequentially so that only one capacitor is connected to the line above at a time. The input signal would be connected to the left of the resistor and the output would be connected to the right of the resistor. This causes the output signal to match the capacitor voltage in the same sequential order as the capacitors turning on.

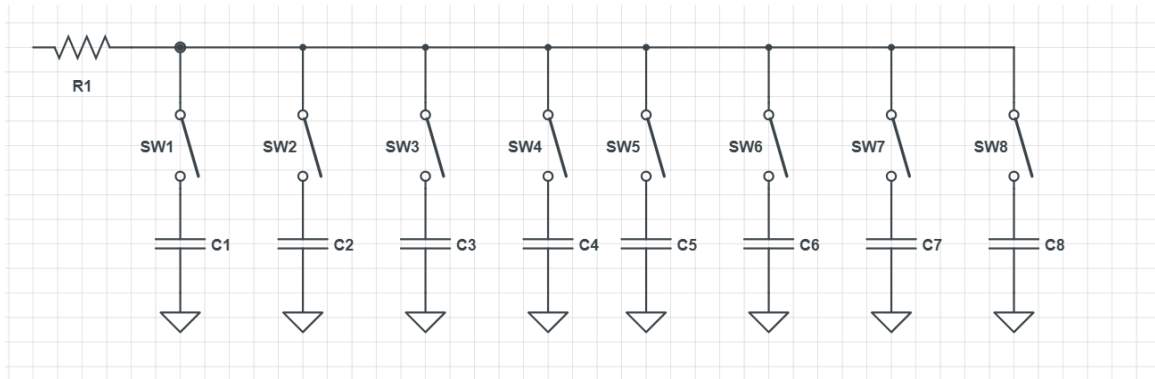


Figure 1.2: Basic N-path Circuit Schematic

The transient signal reaching steady state can be modeled by looking at single capacitor and its switch. The charging of the first capacitor is shown in Figure 1.3. The clock and input frequency match both with a period of  $2\pi$ . The blue line represents the clock. The red line is the input signal. The green line is the first capacitor of the filter. From 0 to  $\pi/4$  the clock is high. The clock turns on the switch and allows the first capacitor to start charging up. As the clock goes low the switch goes into the open position. The charge and therefore voltage on the capacitor remains constant. At the beginning of the next clock cycle at  $2\pi$  the capacitor continues to charge from the last voltage it was at. This process repeats until the voltage on the capacitor is equivalent to the average voltage of the input frequency from 0 to  $\pi/4$ .

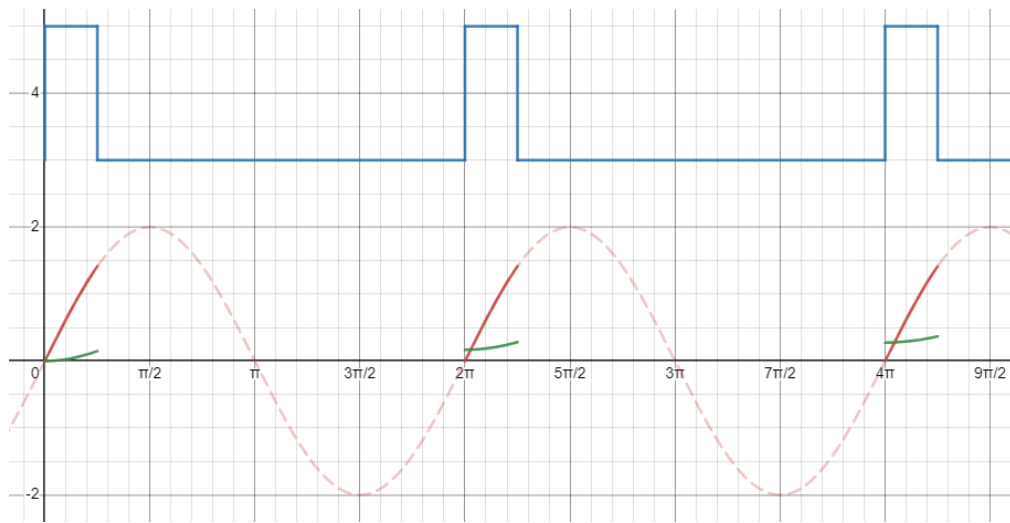
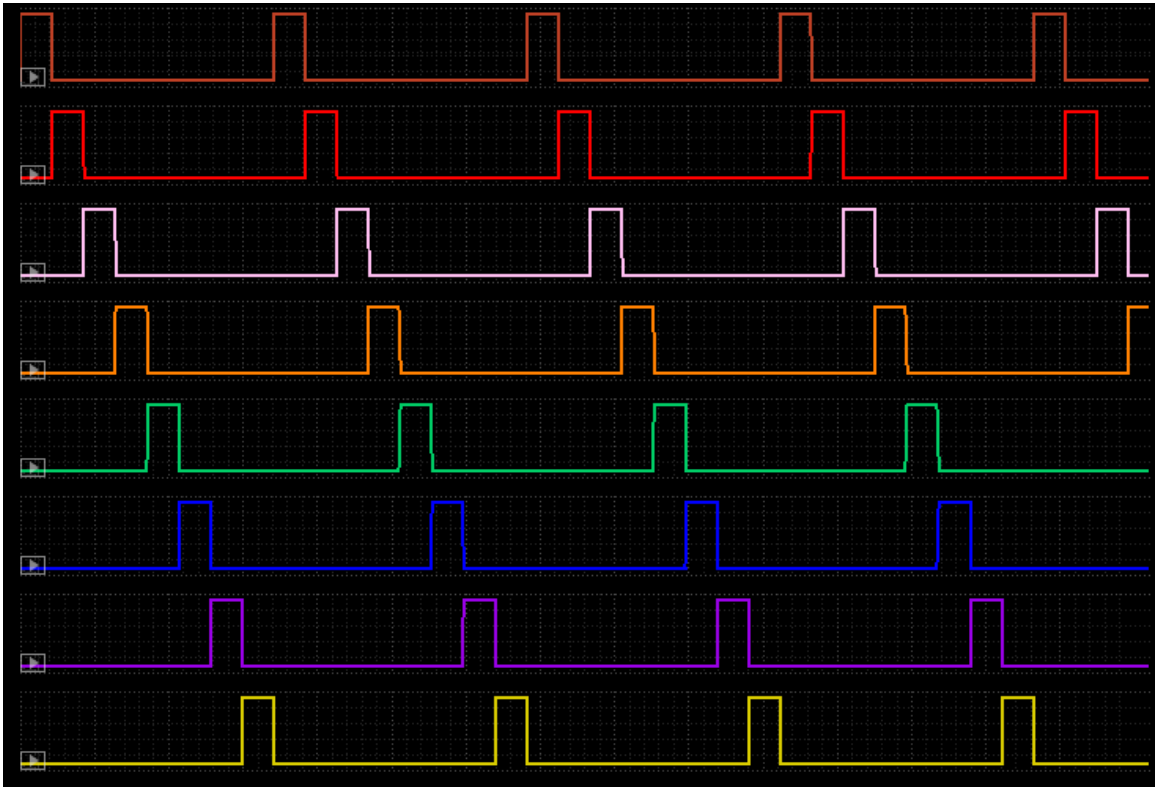


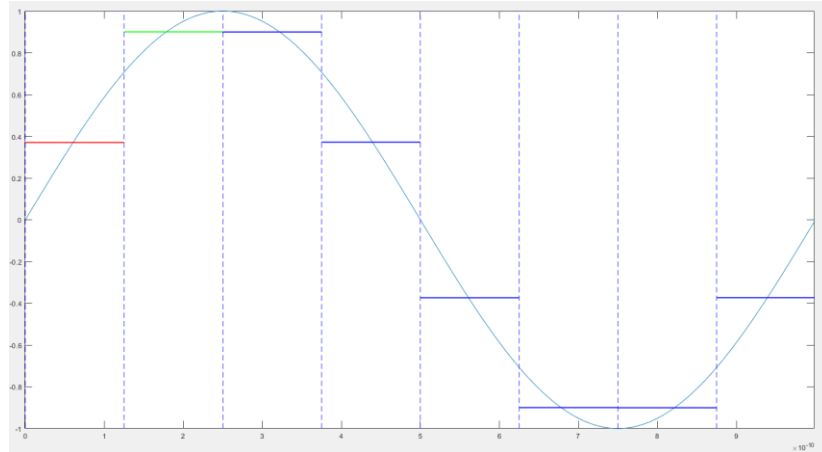
Figure 1.3: N-path Filter Single Capacitor Demonstration

The procedure for charging a single capacitor can be extended to all 8 as each clock is in sequence so the second clock turns on right after the first turns off. All 8 clocks are shown in Figure 1.4.



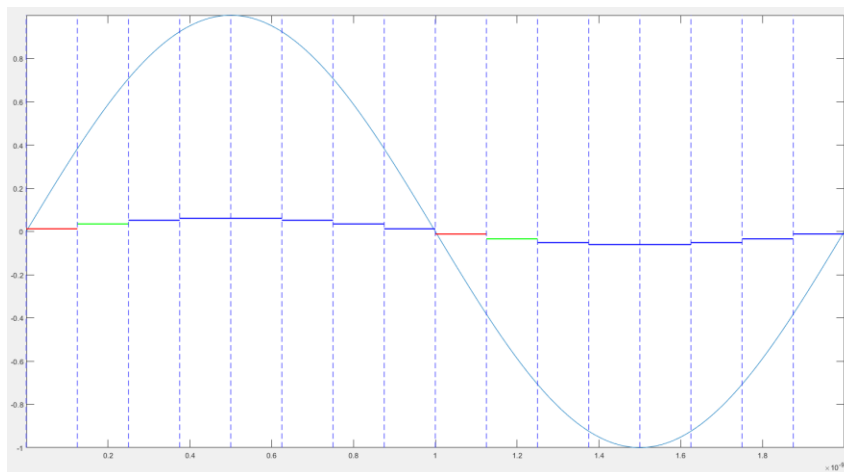
*Figure 1.4: All 8 Input Clocks*

These 8 clocks yield 8 averages taken of 8 different sections of the period of input signal. An example circuit of an N-path filter steady state, where the input frequency matches the clock frequency is shown in Figure 1.5. The input signal travels through R1 and then for the first 1/8 of the period, SW1 becomes closed, and the input signal starts to charge up C1. The vertical dotted lines represent when the switch change occurs, and each horizontal line in between represents the capacitors 1-8. This process repeats for each switch and capacitor. Eventually the capacitors reach steady state where their voltage matches the average voltage of their section of the passed input signal.



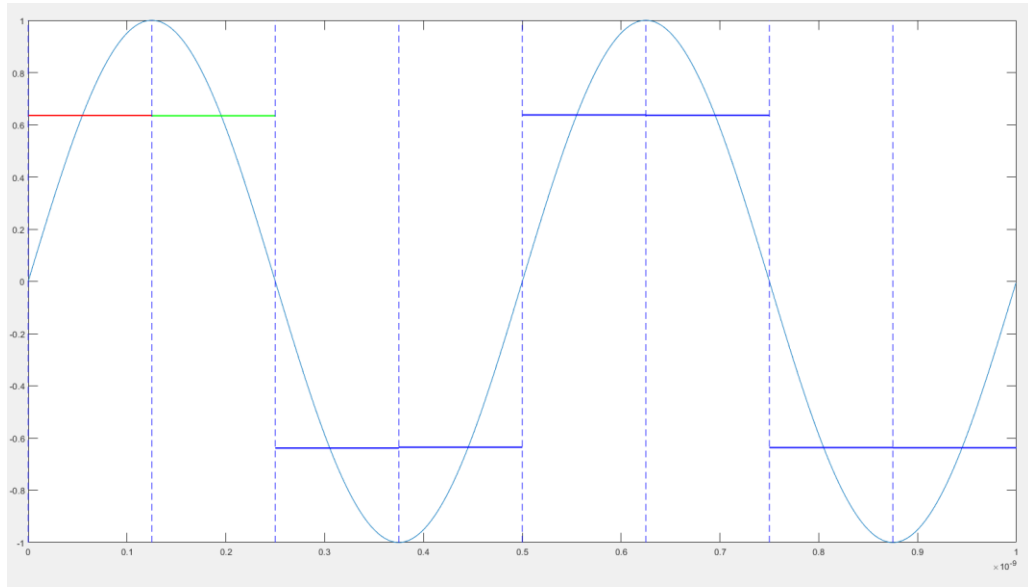
*Figure 1.5: Input and Output Signal When Clock Matches Input Frequency*

The system output behaves differently when the input frequency does not match the frequency of the clock. For a signal at half the frequency of the input signal, the output of the system would be close to 0. This is because during the first half of the period of the signal, all the capacitors would start charging up to the positive value of the signal as shown in Figure 1.6. As shown by the red line representing the first capacitor in the chain, it sees the positive value of the sine wave during its first clock cycle. Then on its second clock cycle starting at 1ns, it only sees the negative half of the sine wave. The capacitors only charge up slightly each clock period, so by seeing the positive voltage then the negative voltage the capacitors charge slightly then discharge. This repeats and settles into a steady state with an output at a low amplitude. This process holds true for all input frequencies outside of the clock frequency and its harmonics



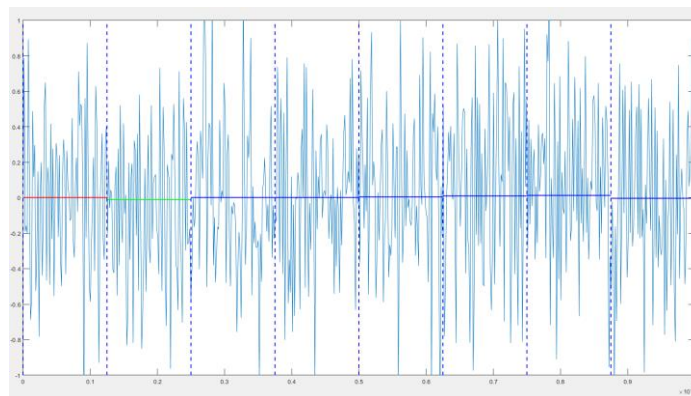
*Figure 1.6: Input Frequency is Half of Clock Frequency*

One disadvantage to the non-differential N-path filter is that for even harmonics, the signal propagates almost fully through the filter as shown in Figure 1.7. In this case, the first four capacitors capture the full period of the signal. The second four capacitors capture the same full period, despite the period not being within the ideal bandpass. This is true as the harmonics increase.



*Figure 1.7: Input Frequency is Double of Clock Frequency*

Given random noise as an input as shown in Figure 1.8, none of the capacitors can charge up or down due to the average of the noise being zero. This effectively makes the N-path a filter as shown from the previous figures through propagating select signals but not others. The N-path filter chooses to propagate signals near the center frequency and its harmonics.



*Figure 1.8: Input is White Noise*

The block diagram of an N-path filter is shown in Figure 1.9. The first stage of the block diagram takes the input signal and splits it into n paths by mixing the input with a sequence of pulse trains  $\varphi$  with periods  $T_s$ . In the second stage each signal goes through a low pass filter. Since the splitting occurred in the first stage, the lowpass filter only passes signals where the input signal's period matches the clock's period. The third stage then recombines the signal's output from the lowpass filter. On a block level, this is how the N-path filter operates. The conversion of the lowpass filter to the bandpass filter is shown in Figure 1.10. Due to the n paths, the lowpass filters effectively gets shifted over to the clock's frequency. This creates the unique qualities of the bandpass in the N-path filter.

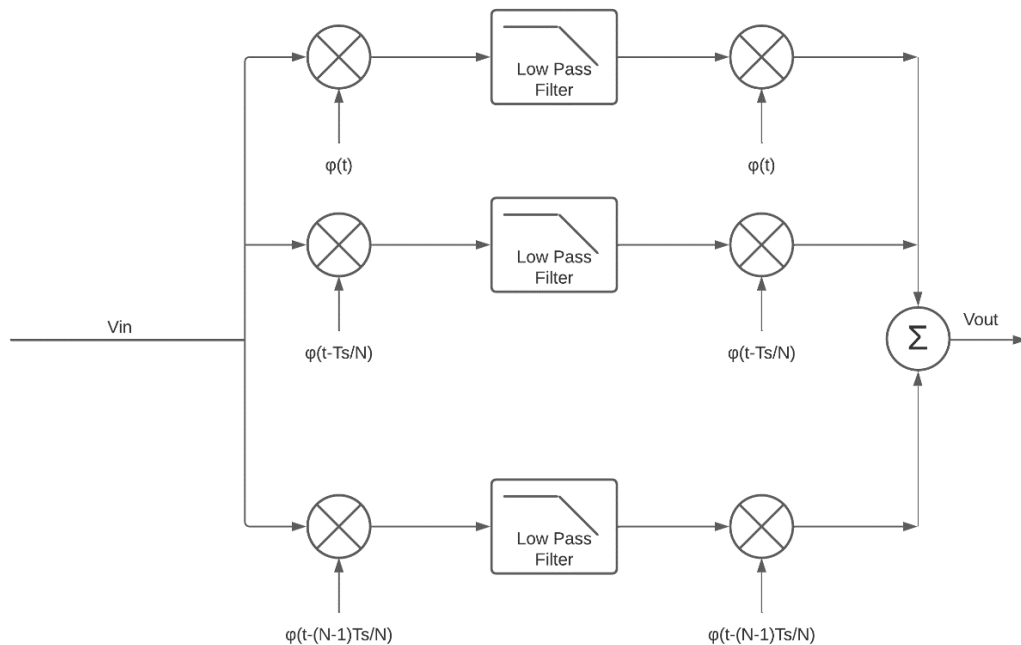


Figure 1.9: Block Diagram of N-path Filter

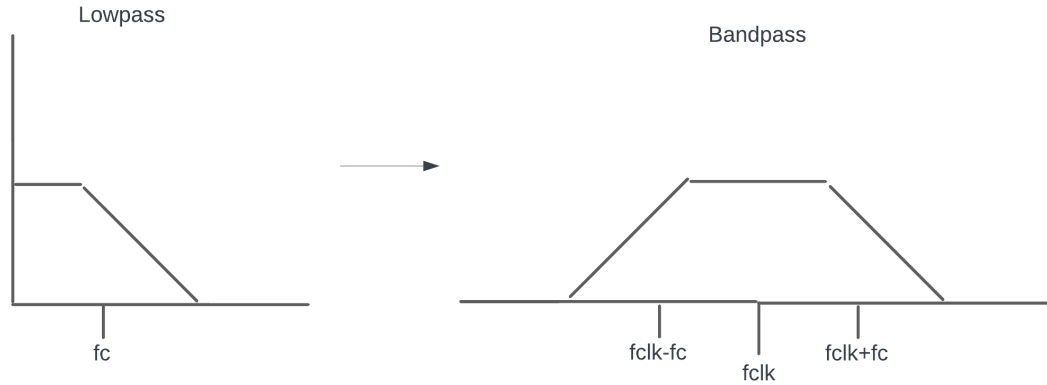


Figure 1.10: Lowpass to Bandpass Conversion by N-path Filter

The theoretical bandwidth can be derived by taking the lowpass filter response and shifting it up to the N-path filter's center frequency.

The cutoff frequency of a standard RC lowpass filter is given by

$$f_c = \frac{1}{2\pi RC_{total}}$$

The N-path filter has  $N$  capacitors all contributing to the total capacitance of the filter so the cutoff frequency for lowpass filter in terms of the N-path filter elements is given by.

$$f_c = \frac{1}{2\pi NRC}$$

Now the lowpass filter can be shifted up to the center frequency of the N-path filter. One aspect that does need to be considered is that the lowpass filter passes both positive and negative frequencies ranging from  $-f_c$  to  $+f_c$ . Normally this aspect is ignored. However due to shifting of the lowpass filter the negative aspect enters the positive frequency domain. This shifting means that that frequency range of the N-path filter becomes  $f_{clk} - f_c$  to  $f_{clk} + f_c$ . The bandwidth of the N-path filter would then be

$$BW_{N-path} = 2f_c = \frac{1}{\pi NRC} \quad [6]$$

The Q factor of this system is given by

$$Q_{factor} = \frac{f}{BW} = \pi NRC f_{clock} \quad [5]$$



Since this Q factor scales with the total capacitance, the resistance, and the frequency of the clock, the system can obtain Q values in the 1,000's.

### 1.3 Requirements

The goal of this project is to create a CMOS based N-path filter using TSMC technology. The requirements are defined in Table 1.1. The frequency range was taken from the 1.2GHz N-path filter [4] although with a slightly lower range to accommodate this projects main goal of Q-factor to The basis for the Q-factor of the project is that of the tuning fork, with a Q-factor of 1000 on the lower end. [7] The size of the N-filter is based on the 1mm by 1mm size of an 200MHz LC bandpass filter that had a Q-factor of 100 [8].

*Table 1.1: Requirements for N-path Filter Design*

<b>Requirement</b>	<b>Value</b>
Clock range	100MHz to 1GHz
Quality factor	Greater than 1000
Stopband attenuation	More than 10dB
Total size	1mm by 1mm on an IC



assumed to be generated externally. The external source is simulated by using an ideal voltage source pulse train. However, if this signal were to be directly connected to the transmission gate, the results would be more accurate than an actual implementation. The clock generates an ideal pulse with every harmonic present. However, in actual implementations, such as a ring oscillator, the signal will not be a perfect pulse. To decrease these idealities the input clock is sent through a chain of inverters as shown in Figure 2.2. The inverters take the ideal signal and add the missing idealities such as rounding the signal and some additional phase noise.

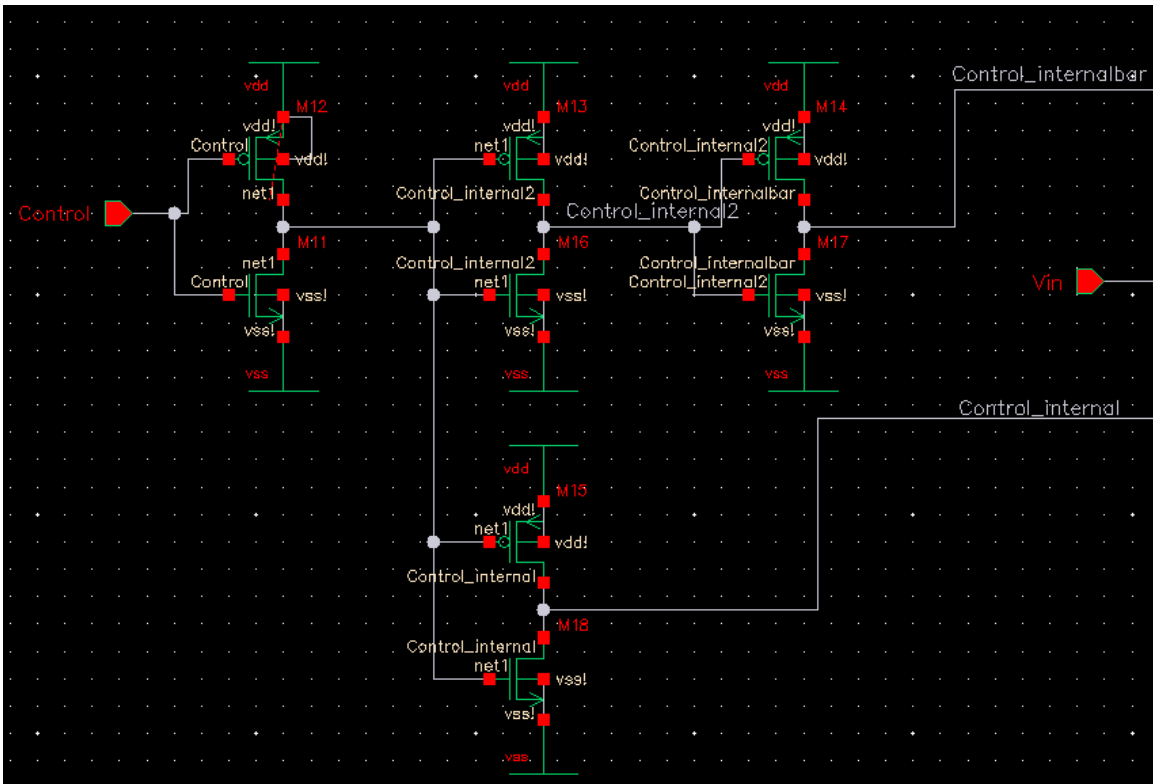


Figure 2.2: Transmission Gate Inverter Chain

Using the transmission gate, the full N-path filter can be constructed. Shown in Figure 2.3 and Figure 2.4 are the completed N-path filter consisting of 8 sets of transmission gates, 8 capacitors and a single resistor. The values chosen for these parts were based around both the filter parameters and the space required for each component. For the N-path filter, a majority of the space is taken by the capacitors. In order to mitigate the large capacitor size, the resistor was increased in size and the capacitor decreased to maintain the proper RC constant. The values

used for the resistor and capacitors were 558k and 8.96pF respectively. In order to fully get 8.96pF for each capacitor in the layout.

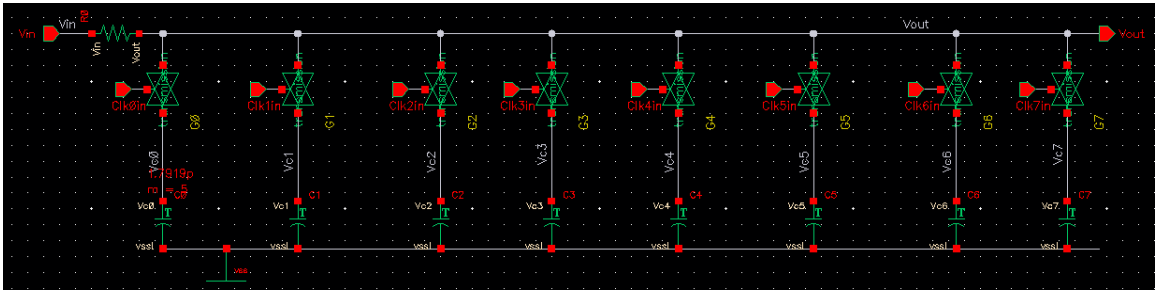


Figure 2.3: N-path Filter Full Schematic

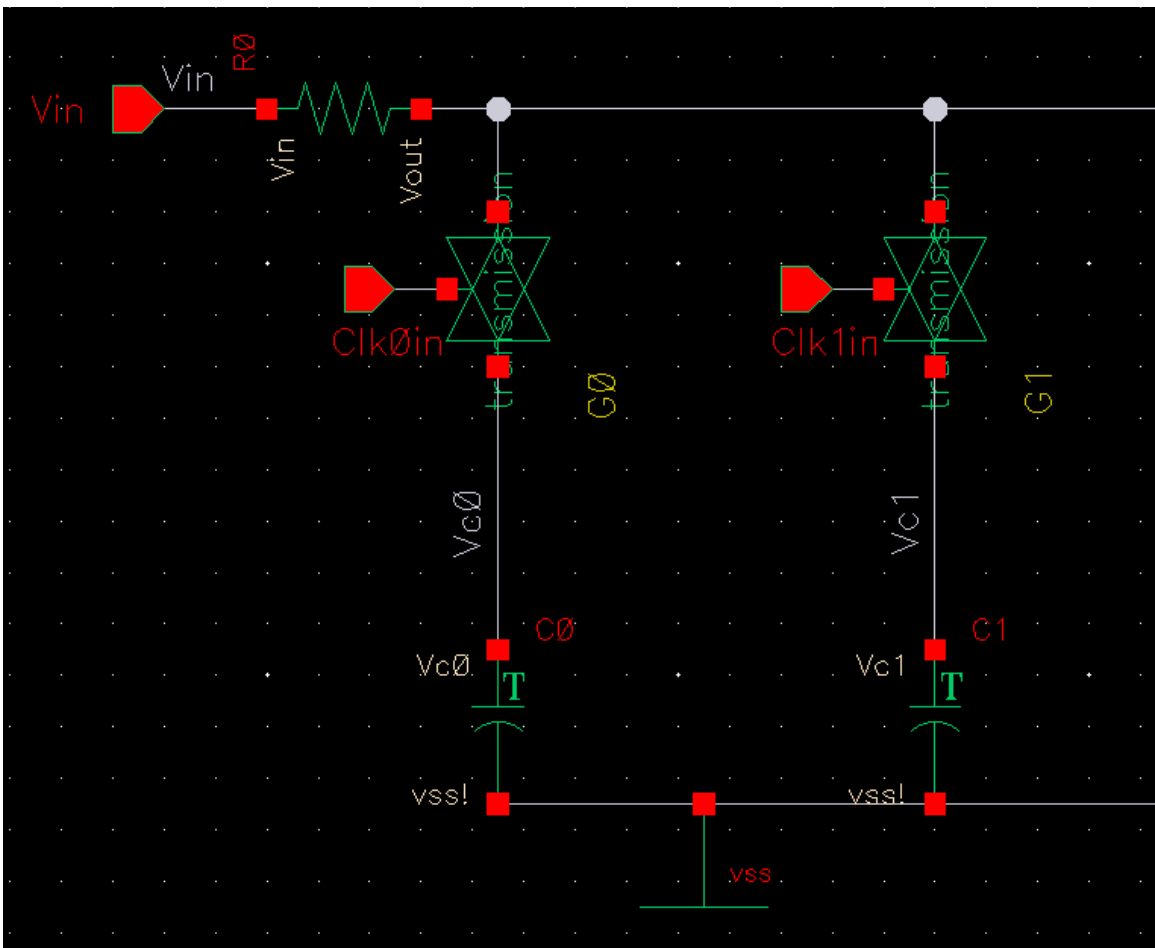


Figure 2.4: N-path Filter Full Schematic Zoomed In

For ease of testing a symbol was made to represent the N-path filter as shown in Figure 2.5. There are 9 inputs to the N-path filter. One input is for the input signal that is to be filtered

and the other 8 inputs are for the clocks used to turn on and off the transmission gates. The single output is located on the right for the filter output.

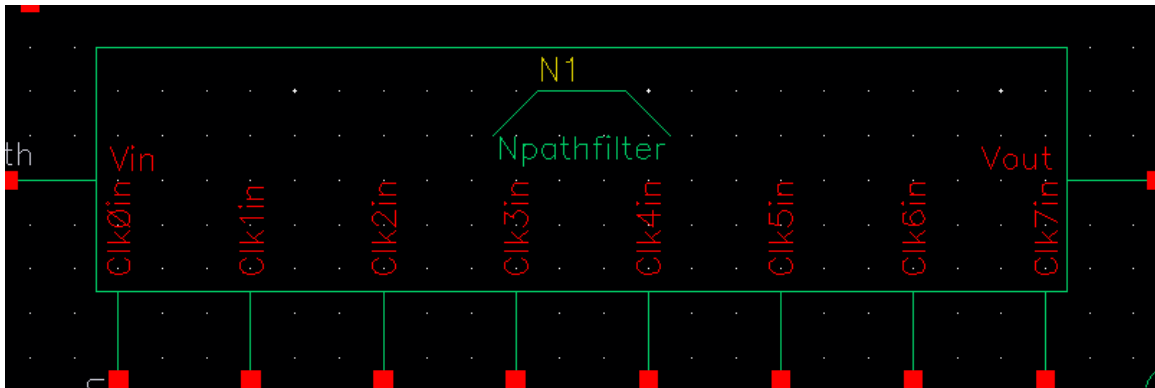


Figure 2.5: N-path Filter Symbol

## 2.2 Testbench Setup

The testbench shown in Figure 2.6 of the N-path filter consists of voltage sources that represent the input signals and clock. The testbench allows for more automated testing and increased clarity for what inputs go into the system. Additionally, a bootstrapping circuit is used on the right of the output.

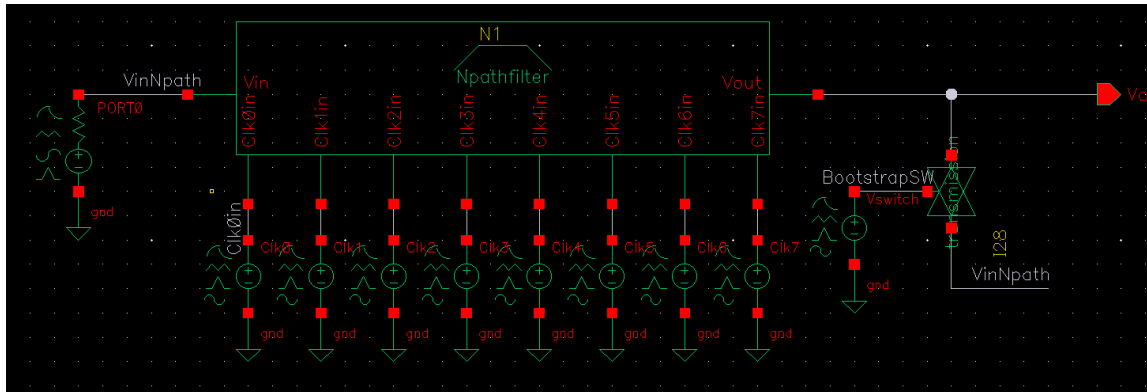


Figure 2.6: N-path Filter Testbench

The bootstrapping circuit shown in Figure 2.6 consists of the voltage source and transmission gate that allows the input signal to directly pass into the output of the N-path filter. The output of the circuit is directly connected to the internal transmission gates and capacitors of the filter. This effectively bypasses the resistor and reduces the RC charge time of the circuit drastically as the only resistance seen from this connection is the internal resistance of the transmission gate. The bypassing allows all the capacitors to charge at extremely fast rates. After

400 clock cycles of the N-path filter, the bootstrap transmission gate is turned off as shown in Figure 2.9. From there, the N-path filter quickly reaches its normal steady state operation. Without the bootstrapping circuit, as shown in Figure 2.8, the filter takes 2us to reach steady state. The bootstrapping circuit allows for the N-path filter to reach steady state in 0.5us. This decrease in time allows for decreased simulation times by a factor of 4 to reach steady state for the N-path filter. The simulation time per transient analysis decreased from 15 minutes to 4 minutes.

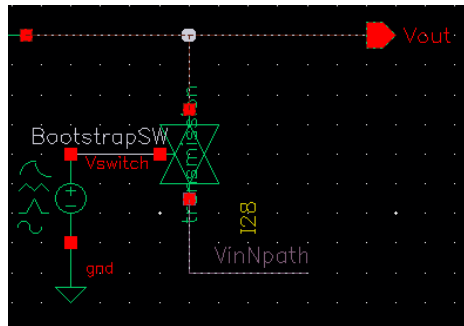


Figure 2.7: N-path Filter Bootstrap in Schematic

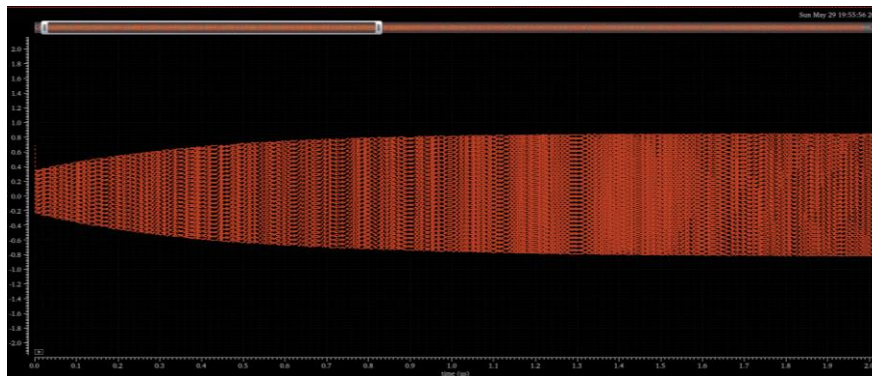


Figure 2.8: N-path Filter Simulation 1GHz Input without Bootstrap

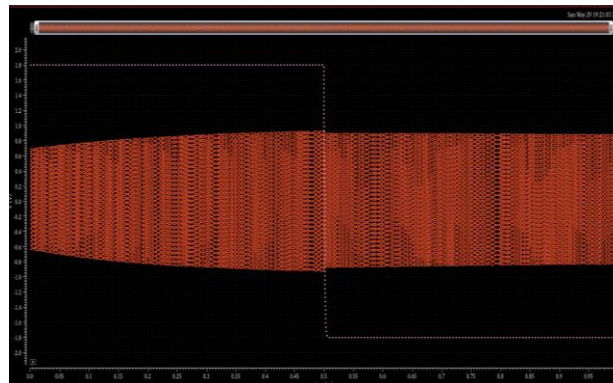


Figure 2.9: N-path Filter Simulation 1GHz Input with Bootstrap

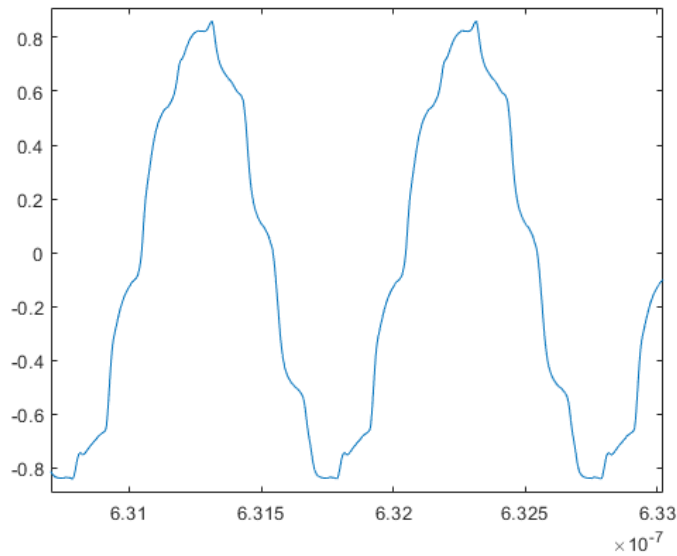
### 2.3 Test procedure

Due to the mixed circuit properties, different test procedures had to be done in order to obtain the desired properties of the N-path filter. Traditionally, a frequency sweep would be run on a normal RC filter. However, a frequency sweep cannot be performed due to the transmission gates and the filter working through a method more similar to sampling rather than a simple transfer function. This requires transient simulations for each test to be run. To get a full picture, tests had to be run en masse. This was accomplished through parametric analysis an example of which is shown in Appendix C. After the parametric analysis, MATLAB was used to manage the large amount of raw transient data.

Chapter 3  
ANALYSIS AND RESULTS

3.1 Analysis

The upper goal of 1GHz clock frequency was chosen for most of the tests. Shown in Figure 3.1 is the transient steady state response of the filter. The input for this system was a 1V peak to peak sine wave. Due to the N-path filter itself there are 8 discrete regions per period. The N-path filter output has a amplitude 0.2 V less than the input 1V peak to peak input signal. This 0.2V drop shows that the filter passes through the center frequency.



*Figure 3.1: Steady State Transient Output*

Additionally, there is some distortion as shown at the peaks of the sine waves. This distortion is caused by the speed of the inverter chains of the transmission gates. The NMOS and PMOS turn on and off at a delay from each other. At each switch there is 50ps of overlap where both the PMOS of the previous cycle and NMOS of the current transmission gate are on at the same time as shown in Figure 3.2. This overlap is the source of the distortion as both transmission gates are fighting to output their voltages.



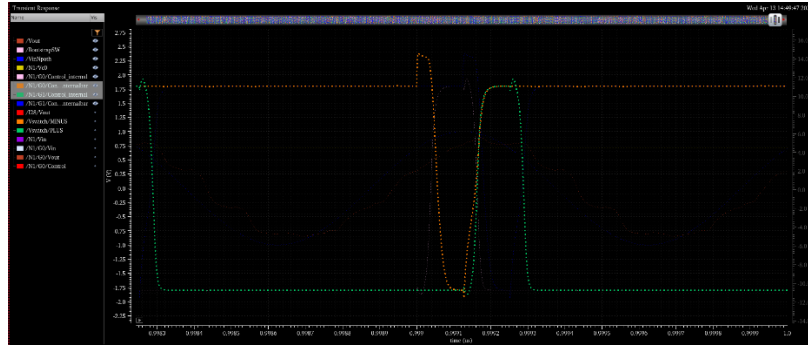


Figure 3.2: PMOS and NMOS Overlapping

The most critical parameter is the frequency response of the data because this defines the operation of the N-path filter. At 1GHz clock frequency, the following plot response is produced in Figure 3.3. This analysis was done using ADE-L where each point represents a separate transient analysis. The bandpass region located at 1GHz is narrow appearing a straight line in this current plot. As shown by the data, the harmonics of the system have very large magnitudes. This is expected based on the operation of the non-differential N-path filter as discussed previously. From this plot the stopband attenuation can be determined to be 11dB based on the passband magnitude and the stopband magnitude.

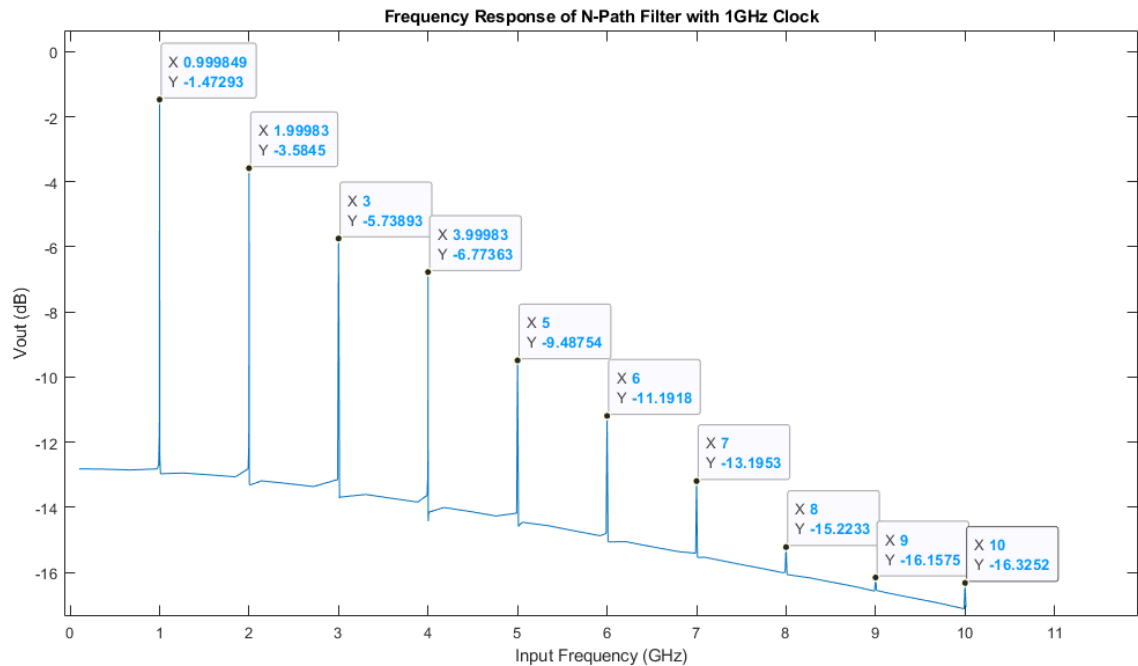


Figure 3.3: Frequency Response for 1GHz Clock

To better understand the bandpass a parametric analysis with 50 points consisting of 5GB of transient data was run. Figure 3.4 shows the frequency response at 8MHz window around the center frequency. At 1GHz the loss is about -3.3dB. The cutoff frequency on the left is 999,516,000Hz and on the right side is 1000,350,000Hz. The total bandwidth of the N-path filter is 833,300 Hz about the center frequency. This results in a Q factor of 1200.

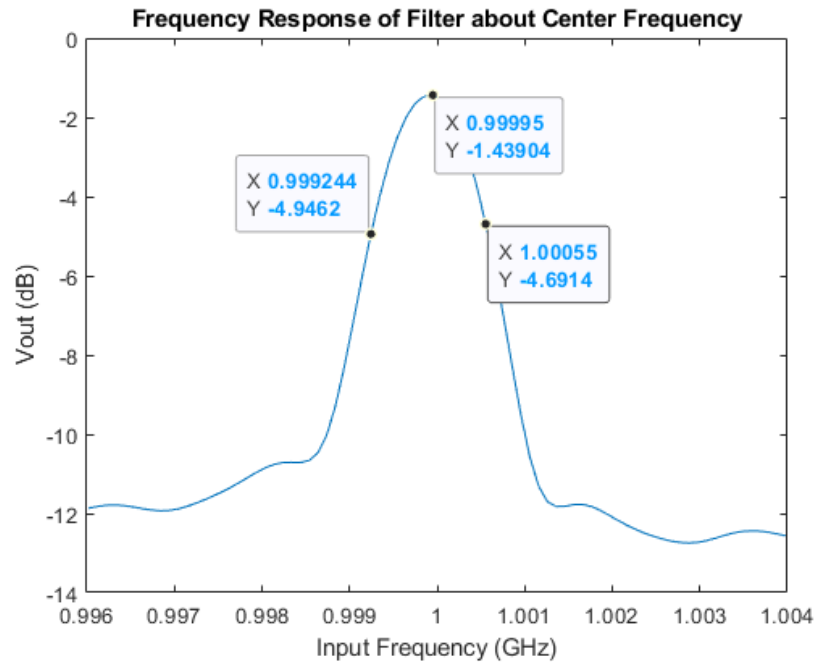


Figure 3.4: Frequency Response about Center Frequency 1GHz

The magnitudes of the output for different clock and input frequency combinations are shown in Figure 3.5. The three plots shown allow for the total range of the filter to be determined. The blue line signifies when the clock and the input frequency match. Ideally, the N-path filter should always respond with fairly low loss. The plots shows that from 10MHz to 750MHz the output magnitude remains constant at -1.2dB. After 750MHz, the magnitude of the output does begin to drop off. This drop off defines the upper range of the filter to be around 1GHz because after this point, the filter loses more of the signal that should be passed through. This upper range found defines the upper range of the N-path filter to be 1GHz. The orange and red lines represent signals that should ideally be filtered out by the N-path filter. The values of the input frequency of these lines are 0.8 and 1.2 times the clock frequency. For example, at 1GHz the frequencies of 800MHz and 1.2GHz should be filtered out. Looking at the

plot at 1GHz there is a loss of -10.1dB and -12.6dB respectively. However, at frequencies below 750MHz, the N-path filter passes through out of band signals at -5dB. This defines the main lower band of the N-path filter to be 750MHz.

These sets of plots define the main operation frequencies of the N-path filter to be 750MHz to 1GHz, however some filtering does occur outside of this band. At frequencies below 750MHz the signals outside of the bandpass only have a -5dB magnitude. Above 1GHz the passed frequencies begin to attenuate more. This -5dB comes from the RC constant being too small at lower frequencies which results in the capacitors having enough time to charge up to the input voltage through a single period of the clock. The passband signals are still preferred at they don't spend time charging at steady state unlike the stopband. This leads to the slight stopband attenuation shown at the lower frequencies.

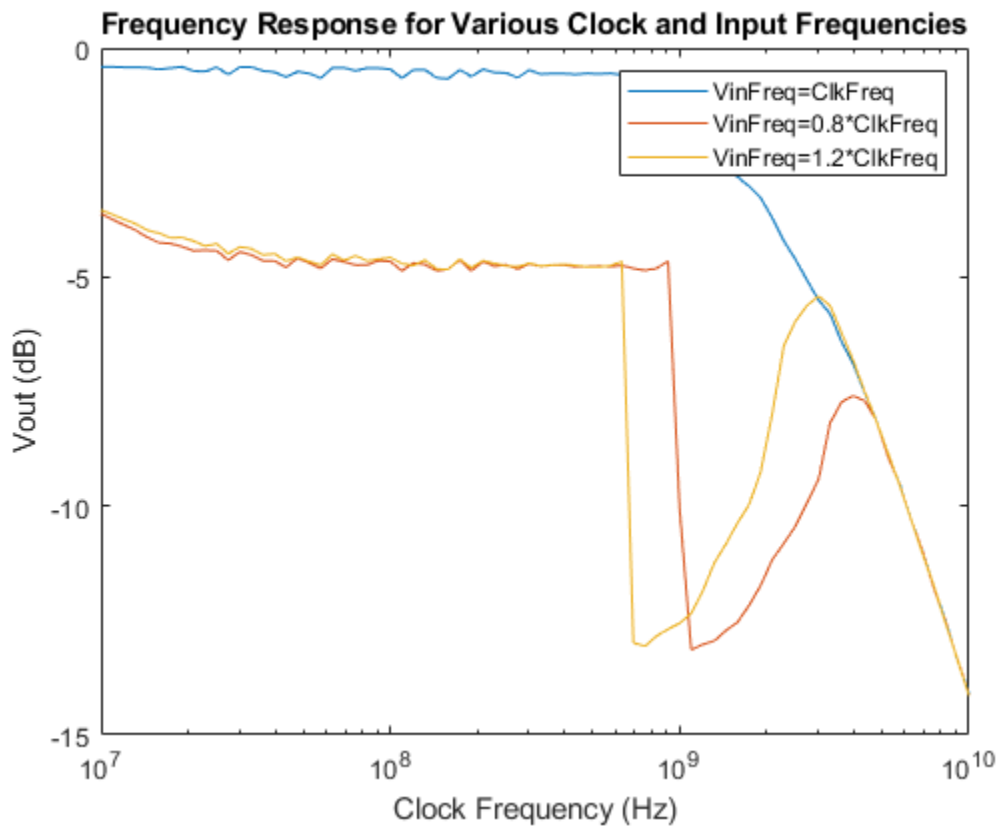


Figure 3.5: Frequency Response for Various Clock and Input Frequencies

### 3.2 Size Analysis

The size of the N-path filter is mainly comprised of the capacitors, which accounts for around 85% of the total layout area. The total area required for the N-path was  $800\mu\text{m}$  by  $800\mu\text{m}$ , which is smaller than the goal of  $1\text{mm}$  by  $1\text{mm}$ . The full layout of the N-path filter is shown in Figure 3.6. The large areas of red squares are the metal plates used to create the capacitors. The rectangle at the top of the image is a line of polymer that snakes around to form the  $558\text{k}\Omega$  resistor for the filter. The total area of the capacitors was reduced by increasing the resistor size as the main importance of the filter is to maintain the same RC constant. However, increasing the resistance does put more load on the filter input.  $558\text{k}\Omega$  was chosen for the maximum value because it allowed for multiple capacitors of maximum sizing without sacrificing the performance of the filter.

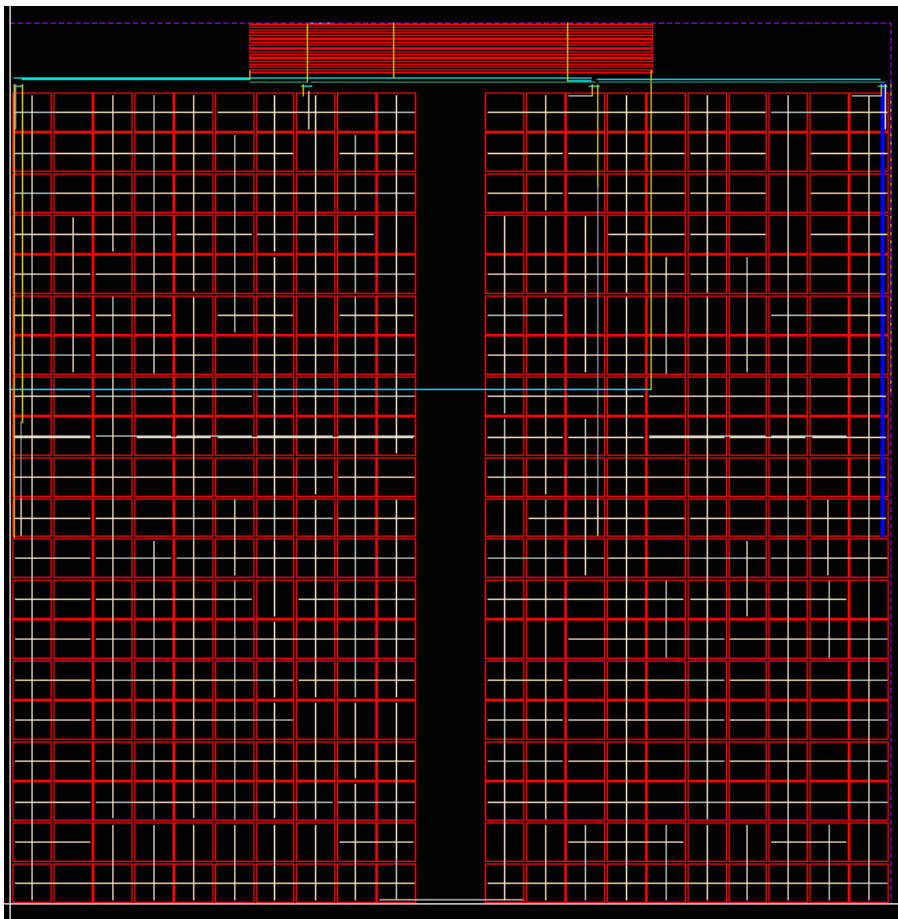


Figure 3.6: Layout showing Size of N-path Filter

### 3.3 Final Results

*Table 3.1: Results for N-path Filter Design*

<b>Requirement</b>	<b>Goal</b>	<b>Actual</b>
Clock range	100MHz to 1GHz	750MHz to 1GHz
Quality factor	Greater than 1000	1200
Stopband attenuation	More than 10dB	11dB
Total size	1mm by 1mm on an IC	800 $\mu$ m by 800 $\mu$ m

## Chapter 4

### CONCLUSION

An N-path filter was successfully designed, tested, and analyzed using TSMC technology. This process took a different approach to analyzing data due to the limits of frequency responses and the complexity of mixed circuit analysis. The analysis consisted of hundreds of transient simulations using ADE-L and its parametric analysis. The simulation process was greatly reduced in complexity and time requirements by introducing a novel bootstrapping circuit. Through the analysis, the frequency response and other tests were able to be compiled and analyzed in MATLAB. Through this mixed signal analysis, other projects may be able to use this information for their own simulations.

From the N-path data collected, many of the goals of the system were met. The bandwidth of the filter was narrow with only an 8.33 kHz bandwidth for a 1GHz center frequency. This calculated to a Q-factor of 1,200 which is above the original goal of 1,000. This Q-factor is equivalent to that of a tuning fork which shows that N-path filter have the potential in sharpness and tunability to be used as RF receivers in the future.

One goal not met by the N-path filter was the tunable center frequency of 100MHz to 1GHz. The N-path filter did meet this upper range however it failed to attenuate the out of band frequencies when the center frequency was reduced below 750MHz. The tunable center frequency tested for the N-path filter designed was 750MHz to 1GHz.

The size requirement of the filter was also met as the area of the components for the IC was 800 $\mu$ m by 800 $\mu$ m. This reached the intended goal of reducing the size below 1mm by 1mm which was the size of a comparable LC filter.

#### 4.1 Future work

N-path filters have the potential to continue to reach higher center frequencies with an extremely high Q-value. This lends itself well to RF filters, especially having a tunable center frequency without the need for different circuitry. For the future it would be beneficial to truly manufacture an N-path filter IC to properly test its effectiveness.

There are also many aspects of this current work that could be expanded upon such as increasing the range, increasing the Q-factor through optimization and potentially other circuit designs. One such way the even harmonics could be eliminated would be through turning the non-differential N-path filter into a differential N-path filter.

One final path to explore would be through the clock generation. The N-path filter used in this project had ideal clock inputs. However there are many ways these clock sequences could be generated. Some potential designs include Delay or Phase locked loops, ring oscillators and more. These options should be explored to increase the research into the viability of N-path filters.

## BIBLIOGRAPHY

- [1] L. E. Franks and I. W. Sandberg, "An Alternative Approach to the Realization of Network Transfer Functions: The  $iV$ -Path Filter," *THE BELL SYSTEM TECHNICAL JOURNAL*, pp. 1321–1350, Sep. 1960.
- [2] E. A. M. Klumperink, H. J. Westerveld and B. Nauta, "N-path filters and mixer-first receivers: A review," 2017 IEEE Custom Integrated Circuits Conference (CICC), 2017, pp. 1-8, doi: 10.1109/CICC.2017.7993643.
- [3] A. el Oualkadi, M. el Kaamouchi, J. Paillot, D. Vanhoenacker-Janvier and D. Flandre, "Fully Integrated High-Q Switched Capacitor Bandpass Filter with Center Frequency and Bandwidth Tuning," *2007 IEEE Radio Frequency Integrated Circuits (RFIC) Symposium*, 2007, pp. 681-684, doi: 10.1109/RFIC.2007.380974.
- [4] M. Darvishi, R. van der Zee and B. Nauta, "A 0.1-to-1.2GHz tunable 6th-order N-path channel-select filter with 0.6dB passband ripple and +7dBm blocker tolerance," *2013 IEEE International Solid-State Circuits Conference Digest of Technical Papers*, 2013, pp. 172-173, doi: 10.1109/ISSCC.2013.6487686.
- [5] M. Steer, "9.2: Q factor," Engineering LibreTexts, 22-May-2022. .
- [6] E. Klumperink, H. Westerveld, and B. Nauta, "N-path filters and mixer-first receivers: A review - utwente.nl." .
- [7] R. H. M. Smit, R. Grande, and B Lasanta, "A low temperature scanning tunneling microscope for electronic and ...," 09-Nov-2007. [Online]. Available: [https://www.researchgate.net/publication/5796566\\_A\\_low\\_temperature\\_scanning\\_tunneling\\_microscope\\_for\\_electronic\\_and\\_force\\_spectroscopy](https://www.researchgate.net/publication/5796566_A_low_temperature_scanning_tunneling_microscope_for_electronic_and_force_spectroscopy). [Accessed: 10-Jun-2022].
- [8] W. B. Kuhn, F. W. Stephenson and A. Elshabini-Riad, "A 200 MHz CMOS Q-enhanced LC bandpass filter," in *IEEE Journal of Solid-State Circuits*, vol. 31, no. 8, pp. 1112-1122, Aug. 1996, doi: 10.1109/4.508258.



APPENDIX A  
MATLAB Code to generate N-path diagrams:

```

figure(1)
freq=1E9
w=freq*2*pi
t=(0:0.01:2*pi)/w;
a=sin(t*w);
plot(t,a)

clock_freq=1E9;
cap_ontime=(1/clock_freq)*(1/8);
capval=[0,0,0,0,0,0,0,0];
count=[0,0,0,0,0,0,0,0];
linecolors=["r", "g", "b", "bl", "bl", "bl", "bl", "bl"]
for i=1:8
    offset=(i-1)*cap_ontime;
    for j=1:length(t)
        if offset<=t(j) && t(j)<cap_ontime+offset
            capval(i)=capval(i)+a(j);
            count(i)=count(i)+1;
        end
    end
    capval(i)=capval(i)/count(i);
    line([0+offset,cap_ontime+offset],[capval(i),capval(i)],'Color',linecolors(i),'LineWidth', 1.5)
    line([offset,offset],[-1,1],'Color','bl','LineWidth', 0.5,'LineStyle',"--")
end
%%
figure(2)
freq=0.5E9
w=freq*2*pi
t=(0:0.01:2*pi)/w;
a=sin(t*w);
plot(t,a)

clock_freq=1E9;
cap_ontime=(1/clock_freq)*(1/8);
capval=[0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0];
count=[0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0];
linecolors=["r", "g", "b", "bl", "bl", "bl", "bl", "bl", "r", "g", "b", "bl", "bl", "bl", "bl", "bl"]
for i=1:16
    offset=(i-1)*cap_ontime;
    for j=1:length(t)
        if offset<=t(j) && t(j)<cap_ontime+offset
            capval(i)=capval(i)+a(j);
            count(i)=count(i)+1;
        end
    end
    capval(i)=capval(i)/count(i)/16;
    line([0+offset,cap_ontime+offset],[capval(i),capval(i)],'Color',linecolors(i),'LineWidth', 1.5)
    line([offset,offset],[-1,1],'Color','bl','LineWidth', 0.5,'LineStyle',"--")
end
%%
figure(3)
freq=2E9
w=freq*2*pi
t=(0:0.01:4*pi)/w;
a=sin(t*w);
plot(t,a)

clock_freq=1E9;
cap_ontime=(1/clock_freq)*(1/8);
capval=[0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0];
count=[0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0];
linecolors=["r", "g", "b", "bl", "bl", "bl", "bl", "bl", "r", "g", "b", "bl", "bl", "bl", "bl", "bl", "bl"]
for i=1:8
    offset=(i-1)*cap_ontime;
    for j=1:length(t)
        if offset<=t(j) && t(j)<cap_ontime+offset

```

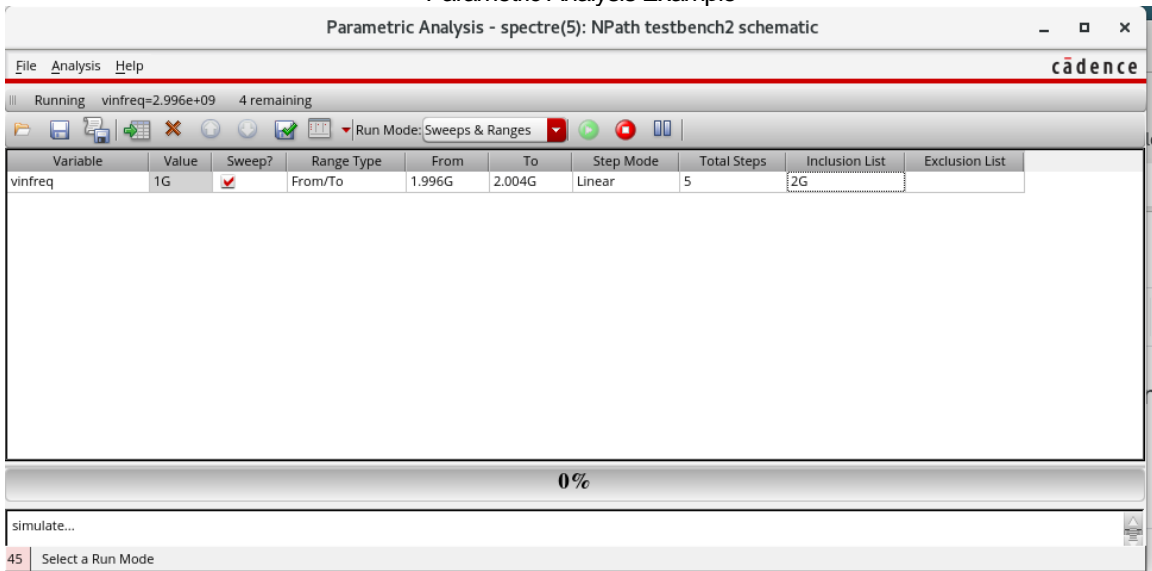
```

        capval(i)=capval(i)+a(j);
        count(i)=count(i)+1;
    end
end
capval(i)=capval(i)/count(i);
line([0+offset,cap_ontime+offset],[capval(i),capval(i)],'Color',linecolors(i),'LineWidth', 1.5)
line([offset,offset],[-1,1],'Color','b','LineWidth', 0.5,'LineStyle',"--")
end
%%
figure(4)
freq=1E9
w=freq*2*pi
t=(0:0.01:2*pi)/w;
x=0*t;
a=0.5*awgn(x,1);
plot(t,a,'LineWidth', 0.1)

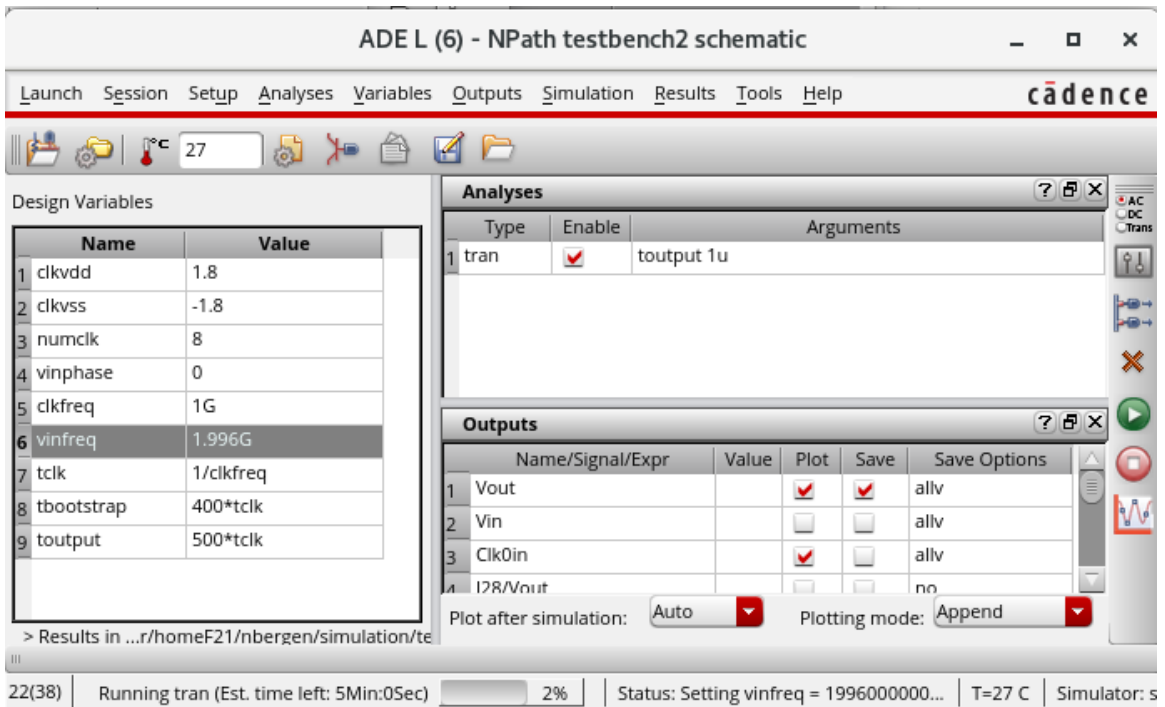
clock_freq=1E9;
cap_ontime=(1/clock_freq)*(1/8);
capval=[0,0,0,0,0,0,0,0];
count=[0,0,0,0,0,0,0,0];
linecolors=["r", "g", "b", "bl", "bl", "bl", "bl", "bl"]
for i=1:8
    offset=(i-1)*cap_ontime;
    for j=1:length(t)
        if offset<=t(j) && t(j)<cap_ontime+offset
            capval(i)=capval(i)+a(j);
            count(i)=count(i)+1;
        end
    end
end
capval(i)=capval(i)/count(i)/8;
line([0+offset,cap_ontime+offset],[capval(i),capval(i)],'Color',linecolors(i),'LineWidth', 1.5)
line([offset,offset],[-1,1],'Color','b','LineWidth', 1.5,'LineStyle',"--")
end
ylim([-1,1])

```

## APPENDIX B Parametric Analysis Example



*Figure B.1: Example of Parametric Analysis Setup to Measure Frequency Response 2GHz Input Signal for a 1GHz Clock*



*Figure B.2: Example of Parametric Analysis Setup to Measure Frequency Response 2GHz Input Signal for a 1GHz Clock*

APPENDIX C  
Processed Data

*Table C.1: Processed Data Frequency Response*

Frequency	Vout(dB)	Frequency	Vout(dB)	Frequency	Vout(dB)	Frequency	Vout(dB)
9.9602E+08	-11.87	7.0882E+09	-15.54	1.0015E+09	-11.82	4.0012E+09	-14.06
9.9612E+08	-11.82	7.3794E+09	-15.70	1.0016E+09	-11.78	4.0016E+09	-14.14
9.9622E+08	-11.79	7.6706E+09	-15.85	1.0017E+09	-11.77	4.0019E+09	-14.32
9.9632E+08	-11.79	7.9618E+09	-16.01	1.0018E+09	-11.83	4.0023E+09	-14.43
9.9642E+08	-11.80	8.0000E+09	-15.22	1.0019E+09	-11.92	4.0026E+09	-14.40
9.9652E+08	-11.84	8.2529E+09	-16.16	1.0020E+09	-12.05	4.0030E+09	-14.39
9.9663E+08	-11.88	8.5441E+09	-16.32	1.0021E+09	-12.18	4.0033E+09	-14.42
9.9673E+08	-11.91	8.8353E+09	-16.48	1.0022E+09	-12.30	4.0037E+09	-14.35
9.9683E+08	-11.93	9.0000E+09	-16.16	1.0023E+09	-12.40	4.0040E+09	-14.28
9.9693E+08	-11.93	9.1265E+09	-16.63	1.0024E+09	-12.50	1.9960E+09	-12.13
9.9703E+08	-11.89	9.4176E+09	-16.78	1.0025E+09	-12.58	1.9963E+09	-12.03
9.9713E+08	-11.83	9.7088E+09	-16.94	1.0026E+09	-12.64	1.9967E+09	-11.91
9.9723E+08	-11.76	1.0000E+10	-16.33	1.0027E+09	-12.70	1.9970E+09	-11.75
9.9733E+08	-11.67	1.9960E+09	-12.13	1.0028E+09	-12.73	1.9974E+09	-11.50
9.9743E+08	-11.57	1.9963E+09	-12.03	1.0029E+09	-12.74	1.9977E+09	-11.19
9.9753E+08	-11.47	1.9967E+09	-11.91	1.0030E+09	-12.73	1.9981E+09	-10.78
9.9763E+08	-11.36	1.9970E+09	-11.75	1.0031E+09	-12.69	1.9984E+09	-10.20
9.9773E+08	-11.23	1.9974E+09	-11.50	1.0032E+09	-12.63	1.9988E+09	-9.21
9.9783E+08	-11.10	1.9977E+09	-11.19	1.0033E+09	-12.56	1.9991E+09	-7.48
9.9793E+08	-10.97	1.9981E+09	-10.78	1.0034E+09	-12.50	1.9995E+09	-5.10
9.9804E+08	-10.85	1.9984E+09	-10.20	1.0035E+09	-12.46	1.9998E+09	-3.58
9.9814E+08	-10.76	1.9988E+09	-9.21	1.0036E+09	-12.44	2.0000E+09	-3.91
9.9824E+08	-10.72	1.9991E+09	-7.48	1.0037E+09	-12.45	2.0002E+09	-5.01
9.9834E+08	-10.71	1.9995E+09	-5.10	1.0038E+09	-12.47	2.0005E+09	-8.04
9.9844E+08	-10.71	1.9998E+09	-3.58	1.0039E+09	-12.52	2.0009E+09	-10.51
9.9854E+08	-10.67	2.0000E+09	-3.91	1.0040E+09	-12.57	2.0012E+09	-11.91
9.9864E+08	-10.46	2.0002E+09	-5.01	1.0000E+08	-12.82	2.0016E+09	-12.58
9.9874E+08	-10.01	2.0005E+09	-8.04	3.9118E+08	-12.83	2.0019E+09	-12.89
9.9884E+08	-9.27	2.0009E+09	-10.51	6.8235E+08	-12.85	2.0023E+09	-13.09
9.9894E+08	-8.28	2.0012E+09	-11.91	9.7353E+08	-12.82	2.0026E+09	-13.21
9.9904E+08	-7.16	2.0016E+09	-12.58	1.2647E+09	-12.95	2.0030E+09	-13.26
9.9914E+08	-6.02	2.0019E+09	-12.89	1.5559E+09	-13.00	2.0033E+09	-13.27
9.9924E+08	-4.95	2.0023E+09	-13.09	1.8471E+09	-13.07	2.0037E+09	-13.27
9.9935E+08	-3.98	2.0026E+09	-13.21	2.1382E+09	-13.19	2.0040E+09	-13.28
9.9945E+08	-3.16	2.0030E+09	-13.26	2.4294E+09	-13.27	9.9000E+08	-12.68
9.9955E+08	-2.50	2.0033E+09	-13.27	2.7206E+09	-13.36	1.0100E+09	-12.97

9.9965E+08	-1.99	2.0037E+09	-13.27	3.0000E+09	-5.74	1.9900E+09	-12.82
9.9975E+08	-1.65	2.0040E+09	-13.28	3.0118E+09	-13.69	2.0100E+09	-13.32
9.9985E+08	-1.47	3.9960E+09	-13.11	3.3029E+09	-13.61	2.9900E+09	-13.15
9.9995E+08	-1.44	3.9963E+09	-13.03	3.5941E+09	-13.73	3.0100E+09	-13.73
1.0001E+09	-1.58	3.9967E+09	-12.98	3.8853E+09	-13.84	3.9900E+09	-13.63
1.0002E+09	-1.88	3.9970E+09	-12.86	4.1765E+09	-14.01	4.0100E+09	-14.15
1.0003E+09	-2.33	3.9974E+09	-12.71	4.4676E+09	-14.13	4.9900E+09	-14.18
1.0004E+09	-2.95	3.9977E+09	-12.50	4.7588E+09	-14.27	5.0100E+09	-14.59
1.0005E+09	-3.74	3.9981E+09	-12.30	5.0000E+09	-9.49	5.9900E+09	-14.80
1.0006E+09	-4.69	3.9984E+09	-11.92	5.0500E+09	-14.46	6.0100E+09	-15.06
1.0007E+09	-5.78	3.9988E+09	-11.32	5.3412E+09	-14.57	6.9900E+09	-15.41
1.0008E+09	-7.00	3.9991E+09	-10.93	5.6324E+09	-14.72	7.0100E+09	-15.54
1.0009E+09	-8.28	3.9995E+09	-9.29	5.9235E+09	-14.87	7.9900E+09	-15.99
1.0010E+09	-9.52	3.9998E+09	-6.77	6.0000E+09	-11.19	8.0100E+09	-16.07
1.0011E+09	-10.57	4.0000E+09	-9.51	6.2147E+09	-15.06	8.9900E+09	-16.58
1.0012E+09	-11.31	4.0002E+09	-10.56	6.5059E+09	-15.22	9.0100E+09	-16.55
1.0013E+09	-11.70	4.0005E+09	-10.30	6.7971E+09	-15.36	9.9900E+09	-17.12
1.0014E+09	-11.82	4.0009E+09	-12.55	7.0000E+09	-13.20	1.0010E+10	-17.06

Table C.2: Processed Data for Frequency Response to Various Clock and Vin Frequencies

clkfreq	Vout (dB)		
	vinfreq=clkfreq	vinfreq=0.8*clkfreq	vinfreq=1.2*clkfreq
1.00E+07	-0.411	-3.616	-3.537
1.10E+07	-0.412	-3.743	-3.636
1.20E+07	-0.418	-3.863	-3.741
1.32E+07	-0.421	-3.980	-3.837
1.45E+07	-0.429	-4.135	-3.982
1.58E+07	-0.466	-4.250	-4.041
1.74E+07	-0.436	-4.268	-4.144
1.91E+07	-0.411	-4.335	-4.138
2.09E+07	-0.500	-4.427	-4.212
2.29E+07	-0.512	-4.416	-4.325
2.51E+07	-0.420	-4.426	-4.273
2.75E+07	-0.578	-4.631	-4.491
3.02E+07	-0.414	-4.451	-4.341
3.31E+07	-0.414	-4.517	-4.382
3.63E+07	-0.497	-4.644	-4.517
3.98E+07	-0.527	-4.650	-4.492
4.37E+07	-0.633	-4.788	-4.654
4.79E+07	-0.514	-4.585	-4.570
5.25E+07	-0.555	-4.699	-4.664
5.75E+07	-0.658	-4.811	-4.736
6.31E+07	-0.429	-4.609	-4.501
6.92E+07	-0.430	-4.668	-4.634
7.59E+07	-0.496	-4.737	-4.539
8.32E+07	-0.437	-4.736	-4.646
9.12E+07	-0.439	-4.643	-4.610
1.00E+08	-0.461	-4.660	-4.574
1.10E+08	-0.656	-4.867	-4.708
1.20E+08	-0.474	-4.693	-4.751
1.32E+08	-0.473	-4.727	-4.630
1.45E+08	-0.651	-4.865	-4.826
1.58E+08	-0.669	-4.826	-4.834
1.74E+08	-0.475	-4.632	-4.609
1.91E+08	-0.621	-4.867	-4.799
2.09E+08	-0.463	-4.661	-4.644
2.29E+08	-0.535	-4.765	-4.704
2.51E+08	-0.542	-4.733	-4.757
2.75E+08	-0.649	-4.827	-4.776
3.02E+08	-0.479	-4.712	-4.699

3.31E+08	-0.567	-4.765	-4.761
3.63E+08	-0.557	-4.740	-4.738
3.98E+08	-0.558	-4.738	-4.725
4.37E+08	-0.574	-4.773	-4.748
4.79E+08	-0.554	-4.772	-4.778
5.25E+08	-0.565	-4.762	-4.780
5.75E+08	-0.553	-4.771	-4.756
6.31E+08	-0.586	-4.749	-4.659
6.92E+08	-0.603	-4.812	-12.989
7.59E+08	-0.683	-4.857	-13.065
8.32E+08	-1.190	-4.813	-12.842
9.12E+08	-1.259	-4.653	-12.691
1.00E+09	-1.479	-10.075	-12.562
1.10E+09	-1.858	-13.142	-12.344
1.20E+09	-2.016	-13.026	-11.856
1.32E+09	-2.217	-12.947	-11.241
1.45E+09	-2.530	-12.721	-10.831
1.58E+09	-2.814	-12.539	-10.372
1.74E+09	-3.020	-12.171	-9.974
1.91E+09	-3.271	-11.736	-9.256
2.09E+09	-3.709	-11.175	-7.993
2.29E+09	-4.212	-10.820	-6.501
2.51E+09	-4.614	-10.443	-5.971
2.75E+09	-5.067	-9.938	-5.625
3.02E+09	-5.506	-9.421	-5.416
3.31E+09	-5.805	-8.185	-5.627
3.63E+09	-6.423	-7.728	-6.229
3.98E+09	-6.898	-7.596	-6.826
4.37E+09	-7.533	-7.707	-7.515
4.79E+09	-8.132	-8.137	-8.149
5.25E+09	-8.900	-8.926	-8.846
5.75E+09	-9.514	-9.523	-9.543
6.31E+09	-10.297	-10.306	-10.298
6.92E+09	-10.991	-10.995	-11.045
7.59E+09	-11.793	-11.826	-11.778
8.32E+09	-12.527	-12.550	-12.580
9.12E+09	-13.363	-13.353	-13.355
1.00E+10	-14.114	-14.123	-14.139

## APPENDIX D

### MATLAB Code to Process Parametric Analysis

```

data=Test1GAllData;

data_array=table2array(data(:,2:2:end));
data_array_time=table2array(data(:,1:2:end));
rmsvals=rms(data_array);
dboutput=20*log10(rmsvals)-20*log10(0.707);
names=insertAfter(erase(erase(data(:,2:2:end)).Properties.VariableNames, 'Voutvinfreq'), '_Y
'),1, '.');
freqs=str2double(names);
figure(1)
plot(freqs,dboutput)
xlabel('Input Frequency (GHz)')
ylabel('Vout (dB)')
title('Frequency Response of Filter about Center Frequency')
figure(10)
plot(data_array_time(:,41),data_array(:,41))
figure(11)
plot(data_array(:,41))
figure(15)
freqs_total=freqs;
dboutput_total=dboutput;
scatter(freqs,dboutput)
xlabel('Input Frequency (GHz)')
ylabel('Vout (dB)')
title('Frequency Response of Filter about Center Frequency')
hold on
%%
data=rmmissing(Test1GSmaller);

data_array=table2array(data(:,2:2:end));
rmsvals=rms(data_array);
dboutput=20*log10(rmsvals)-20*log10(0.707);

names=insertAfter(erase(erase(data(:,2:2:end)).Properties.VariableNames, 'Voutvinfreq'), '_Y
'),1, '.');
freqs=str2double(names);
figure(2)
plot(freqs,dboutput)
xlabel('Input Frequency (GHz)')
ylabel('Vout (dB)')
title('Frequency Response of Filter about Center Frequency')
%%
data=rmmissing(clksweepexact);

data_array=table2array(data(:,2:2:end));
rmsvals=rms(data_array);
dboutput_1=20*log10(rmsvals)-20*log10(0.707);

names=erase(erase(data(:,2:2:end)).Properties.VariableNames, 'Voutvinfreq'), '_Y');
for i=1:length(names)
    if contains(names(i), 'e')
        names(i)=insertAfter(names(i),1, '.');
    end
end
freqs_1=str2double(names);
figure(3)
semilogx(freqs_1,dboutput_1)
xlabel('Clock Frequency (GHz)')
ylabel('Vout (dB)')
title('Center Frequency Response for Vin Frequency = Clock Frequency')
hold off
%%
data=rmmissing(clksweepexact08);

data_array=table2array(data(:,2:2:end));
rmsvals=rms(data_array);
dboutput_08=20*log10(rmsvals)-20*log10(0.707);

```



```

names=erase(erase(data(:,2:2:end).Properties.VariableNames,'Voutvinfreq'),'_Y');
for i=1:length(names)
    if contains(names(i),'e')
        names(i)=insertAfter(names(i),1, '.');
    end
end
freqs_08=str2double(names);
figure(4)
semilogx(freqs_08,dboutput_08)
xlabel('Clock Frequency (GHz)')
ylabel('Vout (dB)')
title('Frequency Response for Vin Frequency = 0.8*Clock Frequency')
hold off
%%
data=rmmissing(clksweepexact12);

data_array=table2array(data(:,2:2:end));
rmsvals=rms(data_array);
dboutput_12=20*log10(rmsvals)-20*log10(0.707);

names=erase(erase(data(:,2:2:end).Properties.VariableNames,'Voutvinfreq'),'_Y');
for i=1:length(names)
    if contains(names(i),'e')
        names(i)=insertAfter(names(i),1, '.');
    end
end
freqs_12=str2double(names);
figure(5)
semilogx(freqs_12,dboutput_12)
hold off
xlabel('Clock Frequency (GHz)')
ylabel('Vout (dB)')
title('Frequency Response for Vin Frequency = 1.2*Clock Frequency')
figure(6)
semilogx(freqs_1,dboutput_1,freqs_08,dboutput_08,freqs_12,dboutput_12)
hold off
legend('VinFreq=ClkFreq','VinFreq=0.8*ClkFreq','VinFreq=1.2*ClkFreq')
xlabel('Clock Frequency (GHz)')
ylabel('Vout (dB)')
title('Frequency Response for Various Clock and Input Frequencies')
figure(7)
plot(freqs_1,dboutput_1,freqs_08,dboutput_08,freqs_12,dboutput_12)
hold off
legend('VinFreq=ClkFreq','VinFreq=0.8*ClkFreq','VinFreq=1.2*ClkFreq')
xlabel('Clock Frequency (GHz)')
ylabel('Vout (dB)')
title('Frequency Response for Various Clock and Input Frequencies')
%%
data=rmmissing(Test1G2G);

data_array=table2array(data(:,2:2:end));
rmsvals=rms(data_array);
dboutput=20*log10(rmsvals)-20*log10(0.707);

names=insertAfter(erase(erase(data(:,2:2:end).Properties.VariableNames,'Voutvinfreq'),'_Y'),1, '.');
freqs=str2double(names);
figure(8)
semilogx(freqs,dboutput)
hold off
xlabel('Input Frequency (GHz)')
ylabel('Vout (dB)')
title('Frequency Response of Filter about 2*Center Frequency')

%%
data=rmmissing(freqresponse);

data_array=table2array(data(:,2:2:end));
rmsvals=rms(data_array);
dboutput=20*log10(rmsvals)-20*log10(0.707);

```

```

names=insertAfter(erase(erase(data(:,2:2:end)).Properties.VariableNames,'Voutvinfreq'),'_Y
'),1,'.');
freqs=str2double(names);
figure(11)
plot(freqs,dboutput)
xlabel('Input Frequency (GHz)')
ylabel('Vout (dB)')
title('Frequency Response of Filter about Center Frequency')
freqs_total=cat(2,freqs_total,freqs);
dboutput_total=cat(2,dboutput_total,dboutput);
figure(15)
scatter(freqs,dboutput)
%%
data=rmmissing(Test2G);

data_array=table2array(data(:,2:2:end));
rmsvals=rms(data_array);
dboutput=20*log10(rmsvals)-20*log10(0.707);
names=insertAfter(erase(erase(data(:,2:2:end)).Properties.VariableNames,'Voutvinfreq'),'_Y
'),1,'.');
freqs=str2double(names);
figure(12)
plot(freqs,dboutput)
xlabel('Input Frequency (GHz)')
ylabel('Vout (dB)')
title('Frequency Response of Filter about Center Frequency')
freqs_total=cat(2,freqs_total,freqs);
dboutput_total=cat(2,dboutput_total,dboutput);
figure(15)
scatter(freqs,dboutput)
%%
data=rmmissing(Test4G);

data_array=table2array(data(:,2:2:end));
rmsvals=rms(data_array);
dboutput=20*log10(rmsvals)-20*log10(0.707);
names=insertAfter(erase(erase(data(:,2:2:end)).Properties.VariableNames,'Voutvinfreq'),'_Y
'),1,'.');
freqs=str2double(names);
figure(12)
plot(freqs,dboutput)
xlabel('Input Frequency (GHz)')
ylabel('Vout (dB)')
title('Frequency Response of Filter about Center Frequency')
freqs_total=cat(2,freqs_total,freqs);
dboutput_total=cat(2,dboutput_total,dboutput);
figure(15)
scatter(freqs,dboutput)
%%
data=rmmissing(freqcorner);
data_array=table2array(data(:,2:2:end));
rmsvals=rms(data_array);
dboutput=20*log10(rmsvals)-20*log10(0.707);
names=insertAfter(erase(erase(data(:,2:2:end)).Properties.VariableNames,'Voutvinfreq'),'_Y
'),1,'.');
freqs=str2double(names);
figure(20)
scatter(freqs,dboutput)
xlabel('Input Frequency (GHz)')
ylabel('Vout (dB)')
title('Frequency Response of Filter about Center Frequency')
figure(15)
freqs_total=cat(2,freqs_total,freqs);
dboutput_total=cat(2,dboutput_total,dboutput);
scatter(freqs,dboutput)
xlabel('Input Frequency (GHz)')
ylabel('Vout (dB)')
title('Frequency Response of N-path Filter with 1GHz Clock')
hold off

```

```
figure(16)
scatter(freqs_total,dboutput_total)
[xsort,isrt] = sort(freqs_total);
ysort = dboutput_total(isrt);
plot(xsort,ysort)
xlabel('Input Frequency (GHz)')
ylabel('Vout (dB)')
title('Frequency Response of N-path Filter with 1GHz Clock')
hold off

frequencyresponsedata=transpose([xsort;ysort])
```