

T.R.
GEBZE TECHNICAL UNIVERSITY
GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES

**ANALYSIS OF THE UNCERTAINTY MEASURE OF A CIRCUIT
OR A SYSTEM USING MONTE CARLO METHOD**

ISMAIL HAKKI SONALCAN
A THESIS SUBMITTED FOR THE DEGREE OF
MASTER OF SCIENCE
DEPARTMENT OF ELECTRONIC ENGINEERING

GEBZE
2019

T.R.
GEBZE TECHNICAL UNIVERSITY
GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES

**ANALYSIS OF THE UNCERTAINTY
MEASURE OF A CIRCUIT OR A SYSTEM
USING MONTE CARLO METHOD**

İSMAİL HAKKI SONALCAN
A THESIS SUBMITTED FOR THE DEGREE OF
MASTER OF SCIENCE
DEPARTMENT OF ELECTRONIC ENGINEERING

THESIS SUPERVISOR
ASSIST. PROF. DR. ÖNDER ŞUVAK

GEBZE
2019

T.C.
GEBZE TEKNİK ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

**BİR DEVRE VEYA SİSTEMİN MONTE
CARLO YÖNTEMİ İLE BELİRSİZLİK
ANALİZİNİN YAPILMASI**

İSMAİL HAKKI SONALCAN
YÜKSEK LİSANS TEZİ
ELEKTRONİK MÜHENDİSLİĞİ ANABİLİM DALI

DANIŞMANI
DR. ÖĞR. ÜYESİ ÖNDER ŞUVAK

GEBZE
2019



GTÜ *Fen Bilimleri* Enstitüsü Yönetim Kurulu'nun *29* / *01* / *2019* tarih ve *2019* / *07* sayılı kararıyla oluşturulan jüri tarafından *30* / *01* / *2019* tarihinde tez savunma sınavı yapılan *ismail Hakkı SONALCAN*'ın tez çalışması *Elektronik Mühendisliği* Anabilim Dalında YÜKSEK LİSANS tezi olarak kabul edilmiştir.

JÜRİ

ÜYE

(TEZ DANIŞMANI) : Dr. Öğr. Üyesi Önder ŞUVAK

ÜYE

: Doç. Dr. Serdar Süer ERDEM

ÜYE

: Dr. Öğr. Üyesi Anıl ÇELEBİ

ONAY

Gebze Teknik Üniversitesi Fen Bilimleri Enstitüsü Yönetim Kurulu'nun
...../...../..... tarih ve/..... sayılı kararı.

SUMMARY

The expected operation of an electrical circuit depends on the correct preparation of the circuit as well as the accuracy of the circuit element parameters used in the circuit. For example, if the gain value of a transistor that is intended to be used is significantly different from the nominal value specified by the manufacturer, the circuit may not operate correctly.

Within the scope of the study, theoretical information of six circuits is given. The results of the simulations applied for these circuits were shared. In the simulations applied for circuits, the characteristic tasks of the circuits are examined and the results are shared. In this study, the model information prepared by ASU in 22nm technology was used for the MOSFETs used in circuits.

In the section where there is uncertainty calculation study, the critical parameters of the important elements of the dynamic NAND gate and Class D power amplifier circuits are determined as random variables. In this context, uncertainty measurements were made for the prepared circuits using simulation techniques based on Monte Carlo and Polynomial Chaos methods.

As a result, the technology levels of electronic devices are constantly evolving. The capabilities of engineering programs are increasing. Furthermore, there is always the possibility that the methods of uncertainty modeling will improve. For these reasons, it is inevitable that more detailed and comprehensive studies will be carried out.

Key Words: Monte Carlo Method, Polynomial Chaos Method, D Class Amplifier, Buck converter, Gilbert cell, Inverter chain circuit.

ÖZET

Elektriksel bir devrenin beklenen şekilde çalışması, devrenin doğru hazırlanmasına bağlı olduğu kadar devrede kullanılan devre eleman parametrelerinin doğruluğunda da bağlıdır. Mesela, kullanılması düşünülen bir transistörün kazanç değeri üreticisinin belirlediği nominal değerden önemli ölçüde farklı ise devre çalışmayabilir.

Çalışmada kapsamında, altı adet devrenin teorik bilgileri verilmiştir. Bu devreler için uygulanan simülasyonların sonuçları paylaşıldı. Devreler için uygulanan simülasyonlarda, devrelerin karakteristik özellikleri incelenmiş ve devrelerin simülasyon sonuçları paylaşılmıştır. Çalışmada devrelerdeki MOSFETler için, ASU (Arizona State University) 22nm teknolojiye hazırladığı model bilgisi kullanılmıştır.

Belirsizlik hesaplama çalışmasının olduğu bölümde, bu devrelerden dinamik NAND kapısı ile D sınıfı güç yükselteç devrelerinin önemli görülen elemanlarının kritik parametreleri rastgele değişken olarak belirlenmiştir. Bu bağlamda, Monte Carlo ve Polinom Kaos yöntemlerine dayalı simülasyon teknikleri kullanılarak, hazırlanan devreler için belirsizlik ölçümü yapılmıştır.

Sonuçta, elektronik elemanların teknoloji seviyelerinin sürekli gelişmektedir. Mühendislik programlarının kabiliyetlerinin artmaktadır. Ayrıca, belirsizlik modelleme metotlarının iyileşme ihtimali her zaman bulunmaktadır. Bu sebeplerden ötürü, bu konu ile ilgili daha detaylı ve kapsamlı çalışmaların yapılması kaçınılmazdır.

Anahtar Kelimeler: Monte Carlo Yöntemi, Polinom Kaos Yöntemi, D sınıfı yükselteç, Buck dönüştürücü, Gilbert cell, Inverter zinciri.

ACKNOWLEDGEMENTS

I would like to express my deep and sincere gratitude to my supervisor, Assist. Prof. Dr. Önder ŞUVAK, who not only shared his profound scientific knowledge with me but also taught me great lessons of life. His support, suggestions and encouragement gave me the drive and will to complete this work.

I wish to express my warm and sincere thanks to Mecit Emre DUMAN for his collaboration to this research.

I am grateful to my son Yavuz Selim SONALCAN for his love and support.

Finally, I want to thank my wife Vildan SONALCAN, who showed me the beauty of love.

TABLE of CONTENTS

	<u>Page</u>
SUMMARY	v
ÖZET	vi
ACKNOWLEDGMENTS	vii
TABLE of CONTENTS	viii
LIST of ABBREVIATIONS and ACRONYMS	x
LIST of FIGURES	xi
LIST of TABLES	xv
1. INTRODUCTION	1
1.1. Background	1
1.2. Motivation	5
1.3. Contribution	6
2. RANDOM DISTRIBUTIONS AND UNCERTAINTY METHODS	7
2.1. Probability Density Functions	7
2.2. Probability Distributions Used in the Study	10
2.2.1. Uniform Distribution	11
2.2.2. Truncated Distribution	12
2.3.3. Beta Distribution	14
2.3. Uncertainty Methods	15
2.3.1. Monte Carlo Method	15
2.3.2. Generalized Polynomial Chaos Method	17
3. CIRCUITS USED IN THE THESIS	21
3.1. Buck Converter	21
3.2. Dynamic NAND Gate	25
3.3. Gilbert Cell Mixer	30
3.4. Transmission Gate Chain	40
3.5. Inverter Chain	51
3.6. Class D Amplifier	82
4. NUMERICAL RESULTS	88
4.1. Results of Monte Carlo Method	88

4.2. Results of gPC Method	95
5. CONCLUSION AND FUTURE WORK	98
REFERENCES	101
BIOGRAPHY	105

LIST of ABBREVIATIONS and ACRONYMS

<u>Abbreviations</u>	<u>Explanations</u>
<u>and Acronyms</u>	
ASU	: Arizona State University
CDF	: Cumulative Distribution Function
Cext	: Extrinsic Capacitance
CFD	: Computational Fluid Dynamics
Cgin	: Input Gate Capacitance
Cint	: Intrininsic Capacitance
C _L	: Load Capacitor
CMOS	: Complementary Metal-Oxide Semiconductor
COMP	: Comparator
CV	: Coefficient of Variation
F	: Fanout
FFT	: Fast Fourier Transform
GP	: Galerkin Projection
gPC	: Generalized Polynomial Chaos
JM	: Jacobian Matrix
KVL	: Kirchoff's Voltage Law
LPF	: Low Pass Filter
NAND	: Not And
NMOS	: N Channel Mosfet
PC	: Polynomial Chaos
PDF	: Probability Distribution Function
PMOS	: P Channel Mosfet
PWM	: Pulse Width Modulation
RC	: Resistor Capacitor
Req	: Equivalent Resistance
RV	: Random Variable
SC	: Stochastic Collection
tdf	: Falling Time Delay
tdr	: Rising Time Delay

LIST of FIGURES

<u>Figure No:</u>	<u>Page</u>
2.1: Probability functions of a discrete random variable for a dice tossing experiment. a) PDF. b) CDF.	7
2.2: PDF for continuous random data a) PDF. b) Field representation of the probability value.	7
2.3: CDF according to the PDF in Figure 2.2.	8
2.4: Representation of the link between significant probability distribution functions.	11
2.5: A standard normal distribution which is lower truncated.	12
2.6: Truncated normal distribution PDF.	14
2.7: Using Monte Carlo method to find square and quadrant area and pi number.	16
3.1: Buck converter schematic.	21
3.2: Buck converter schematic on phase.	22
3.3: Buck converter schematic off phase.	23
3.4: Buck converter schematic in LTSPICE tool.	23
3.5: Buck converter output for switch state, supply voltage and load voltage.	24
3.6: Buck converter load voltage output.	25
3.7: Dynamic NAND gate.	26
3.8: Charge leakage in dynamic CMOS	27
3.9: Charge sharing in dynamic CMOS	27
3.10: Dynamic NAND gate circuit schematic.	28
3.11: Dynamic NAND gate simulation result with all signals.	28
3.12: Dynamic NAND gate charge sharing result.	29
3.13: Time calculation of 1mV voltage difference.	30
3.14: Emitter coupled pair.	30
3.15: The DC transfer characteristic of the emitter-coupled pair.	31
3.16: Two quadrant analog multiplier.	32
3.17: Gilbert cell (Four quadrant multiplier).	33
3.18: Gilbert multiplier with predistortion circuits.	35

3.19:	Inverse hyperbolic tangent circuit	36
3.20:	Complete four quadrant analog multiplier.	37
3.21:	Gilbert cell Q3-Q5 turn on.	38
3.22:	Gilbert cell Q4-Q6 turn on.	38
3.23:	Gilbert cell mixer circuit schematic.	39
3.24:	Gilbert cell mixer input waveforms.	39
3.25:	Gilbert cell mixer output waveforms.	40
3.26:	Transmission gate chain diagram.	41
3.27:	Transmission gate chain simulations without buffer and with buffer.	41
3.28:	Single transmission gate circuit schematic.	42
3.29:	Resistance plot of a single transmission gate.	42
3.30:	Two transmission gates schematic for measuring capacitance.	43
3.31:	Tau and node capacitance result for single transmission gate.	43
3.32:	Node capacitance and node resistance of single transmission gate.	44
3.33:	The two transmission gates simulation plot.	44
3.34:	Buffer gate circuit schematic.	45
3.35:	Buffer delay result for rising and falling edge.	46
3.36:	Buffer gate simulation plot.	46
3.37:	Transmission gate single and chain schematic without buffer.	47
3.38:	Delay measurement of the transmission gate chain without buffer.	48
3.39:	The single transmission gate delay on the chain without buffer.	49
3.40:	The simulation plot of the transmission gate chain without buffer.	49
3.41:	The schematic of the transmission gate chain circuit with buffer.	50
3.42:	Delay result on the transmission gate chain with buffer.	50
3.43:	Simulation output of the transmission gate chain with buffer.	51
3.44:	The interior scheme of a MOSFET.	52
3.45:	An inverter circuit diagram.	52
3.46:	Switch models of CMOS inverter.	53
3.47:	Voltage transfer characteristic of static CMOS inverter.	53
3.48:	Switch model of dynamic behavior of static CMOS inverter.	54
3.49:	Impact of device variations on static CMOS inverter VTC.	55
3.50:	The schematic of single inverter with $W_p/W_n=1$.	56
3.51:	The plot of a single inverter with $W_p/W_n=1$.	56

3.52:	The delay results of a single inverter with $W_p/W_n=1$.	57
3.53:	The schematic of single inverter with $W_p/W_n=2$.	57
3.54:	The plot of a single inverter with $W_p/W_n=2$.	58
3.55:	The delay results of a single inverter with $W_p/W_n=2$.	58
3.56:	The schematic of single inverter with $W_p/W_n=3$.	59
3.57:	The plot of a single inverter with $W_p/W_n=3$.	60
3.58:	The delay results of a single inverter with $W_p/W_n=3$.	60
3.59:	The transistor as a switch.	61
3.60:	Propagation delay as a function of W_p/W_n .	63
3.61:	Intrinsic vs Extrinsic vs Parasitic load capacitance.	64
3.62:	Isolated inverter sizing graph.	65
3.63:	Inverter chain with two inverters.	65
3.64:	Inverter chain with N number inverter gates.	66
3.65:	Optimum sizing summary for an inverter chain.	68
3.66:	Single inverter gate schematic without load.	68
3.67:	Single inverter output without load capacitor.	69
3.68:	Single inverter output (zoom) without load capacitor.	69
3.69:	Delay measurement of the single inverter without load capacitor.	70
3.70:	RC charging circuit.	70
3.71:	RC charging circuit curves.	72
3.72:	Single inverter schematic for input capacitance.	73
3.73:	The plot of the single inverter simulation.	73
3.74:	Tau and input gate capacitance result for single inverter.	74
3.75:	Node capacitance result for single inverter.	74
3.76:	The schematic of a single inverter with 40fF load capacitor.	75
3.77:	The plot of the single inverter with 40fF load capacitor.	75
3.78:	Delay measurement of the single inverter with load.	76
3.79:	Inverter chain delay results.	76
3.80:	Inverter chain schematic with four inverters.	77
3.81:	The output of inverter chain with four gates.	78
3.82:	Delay measurement of the inverter chain with four gates.	78
3.83:	Inverter chain delay results.	79
3.84:	The output of inverter chain with two gates.	79

3.85:	The output of inverter chain with three gates.	80
3.86:	The output of inverter chain with five gates.	81
3.87:	Inverter chain all delay results.	82
3.88:	Class D amplifier circuit scheme.	82
3.89:	Class D amplifier circuit stages.	83
3.90:	Triangle and PWM signals.	83
3.91:	Major cause of degradation.	84
3.92:	Class D amplifier schematic 1.	84
3.93:	Subcircuit netlist for U1 and U2 comparators.	85
3.94:	Class D amplifier schematic 2.	85
3.95:	Class D amplifier schematic 3.	86
3.96:	Class D amplifier plot.	86
3.97:	Class D amplifier output on the NGSPICE Tool.	87
3.98:	Class D amplifier input and output signals.	87
4.1:	Monte Carlo analysis diagram for Class D circuit.	89
4.2:	CMOS RS latch circuit used for Class D amplifier.	91
4.3:	PDF for total harmonic distortion of Class D output signal.	94
4.4:	PDF for total harmonic distortion of Class D average power.	94
4.5:	The flow diagram of the gPC application.	96
4.6:	PDF of time for intersection on dynamic NAND simulation.	97

LIST of TABLES

<u>Table No:</u>	<u>Page</u>
3.1: Dynamic NAND gate symbol and Truth Table.	26
3.2: Voltage and current percentages on the capacitor	72
4.1: Random parameters configuration of Class D amplifier circuit.	90
4.2: Random parameters configuration of Buck converter circuit.	93
4.3: Random parameters configuration of dynamic NAND gate circuit.	95

1. INTRODUCTION

1.1. Background

Today, electronic devices such as tablets, mobile phones, and computers have become indispensable parts of our lives. Since interest in these devices is increasing day by day, producers are showing great efforts at the development of these products. In this way they are trying to reach more users.

There are several ways to increase the performance and capabilities of electronic devices. One of these paths goes by reducing the size of elements such as transistors and MOSFETs. By reducing the size of the transistor or MOSFET, more transistors or MOSFETs can fit into the processor or into the processor. In comparison with the old, the processor with more elements inside will be more capable. The ability of the integrated and processors will also increase the capabilities of electronic devices.

The dimensions of transistors and MOSFETs have shrunk sharply in a very short time. Today, the dimensions of the elements have fallen to nanometer levels. These rapidly shrinking elements quickly developed electronic devices, but at the same time it was difficult to control the production standard of the technology.

Considering the parameters of these shrinking elements, it is very difficult to control them on the computer. It is becoming rough to forecast the behavior of the elements and to realize them in the computer environment by changing the physical properties of the computer. In addition, there may be some uncertainties in the production of these small elements.

In short, as the elements become smaller, uncertainties increase. For this reason, in order to predict how the shrinking elements will affect the circuits, it is necessary to apply the stochastic methods besides the deterministic methods. In this way, circuit or system complex can be treated to determine how the system will behave. As a result, the modeling of the element produced by these studies can be done correctly.

Until now, the Monte Carlo Method has been used in the context of conventional stochastic studies [1]. Monte Carlo Method is used for commercial circuit and electromagnetic simulations.

Stanislaw Ulam was a Polish-American mathematician and nuclear physicist [2]. Ulam has evolved the Monte Carlo Method to perform complicated mathematical operations [3, 4]. The method is random sampling. Proper algorithms can be evaluated by random sampling. Today is used in computer software.

Ulam proposed the Monte Carlo Method to evaluate complex mathematical integrals in the theory of nuclear chain reactions. This proposal has been contributed by von Neumann, Metropolis and others to the development of Monte Carlo in a more systematic way, which has greatly contributed to solving most of the complex problems of atom bombing. President Truman explained that the US will develop a hydrogen bomb [5, 6].

Monte Carlo operation generates random variables to form the uncertainty of a system. This is a mathematical process. The random variables or inputs are shaped on the axis of some probability distribution types. A variety of repetitions and simulations are made so that a certain path can occur. The result is reached after various numerical calculations.

Monte Carlo Simulation is the highest approximation which is used when a model has ambiguous parameters or analysis of a dynamic complex system. It is a probable method for modeling the riskier in a system. Because of business, finance, physics, computational biology, statistics, artificial intelligence, and random variables, Monte Carlo Simulations have a wide range of potential applications in these areas. In large projects they are used for prediction of the cost overruns probability and the probability that an asset's price will settle into a certain way. Telecommunication uses network performance in different situations and uses them to help optimize the network. Analysts use them to analyze derivatives, such as options, when an operator evaluates the risk of default. They also use insurance companies and oil well machines. Monte Carlo Simulations, meteorology, astronomy and particle physics have numerous applications outside of business and finance.

Monte Carlo estimates the uncertainty of a system and is not deterministic. But, it is a practical tool to bring the uncertainty of the system closer to reality.

The Monte Carlo Method emerged along World War II. It can be based on to dynamic factors that can affect an output. With the Monte Carlo Method, it can help to reveal the results of decisions made under uncertainty. It also makes it easy to make the right choices under uncertainty. Besides the results, it can also produce the probabilities of the outputs. It uses the probability distribution to model the random

variable. Different types of distribution can be used to model the random variable assigned to the input. The Monte Carlo Method is a better uncertainty and risk simulator than the deterministic method. Not only results, but also the probability values of the results can be reported. Monte Carlo Method can model the corresponding input variable. Monte Carlo Method can be useful for finding the risk in a portfolio. This application estimates that a portfolio brings about the worst return in certain circumstances. For a portfolio of many entities, we can use Monte Carlo Simulation to generate the associated asset prices.

In this study, some of the digital and analog circuit structures were measured by Monte Carlo Simulation. For this purpose, some parameters with critical prespecification of Class D power amplifier, Buck converter and similar circuits are determined as random variables with certain distributions and simulations of the circuits have been realized by using the generated sets. Using the obtained simulated data, the variation of the quantities characteristic of each of these circuits (the value of the average voltage of the output voltage for the Buck converter, and the harmonic distortion of the output voltage for the Class D amplifier) was investigated. Thanks to these analyzes, the critical parameters of the circuits can be given as model inputs together with probability distributions, and output can be generated including the best and worst scenarios along with certain probability distributions as the analysis result. Although this method has nice sides, it has caused the search for alternative methods due to slow convergence and heavy computational burden. For this reason, there has been interest in exploring different ways of statistically analyzing the performance of electronic systems or elements.

As a result of this growing interest, a new approximation based upon the concept of generalized Polynomial Chaos (PC) has been developed which offers more advantages than the Monte Carlo Method [7]. By means of this method statistical properties of different types of circuits were estimated. This makes the process and benchmark load more advantageous than the Monte Carlo Method.

Previous studies have used PC, but the interconnections are concentrated on characterizing the variability analysis of multi-conductor transmission lines in the presence of process conductors [8, 9].

The PC method has been used in other areas as well as these studies. For example, the problem of variability in generic multiport linear circuits has been

addressed [10]. Differently, the method has been extended to nonlinear circuits [11] and transmission line circuits terminating with non-linear loads [12].

The Generalized Polynomial Chaos (gPC) approximation method is based on the calculation of a coefficient in a certain sequence of orthogonal polynomials of the Askey-Wiener type. That is, a numerical method is used to expand the circuit response and calculate the coefficients of this sequence in the Hermite polynomial, which is a sequence of polynomials of the Askey-wiener type. By obtaining these coefficients, the results such as mean, variance are obtained and statistical information about the circuit behavior is obtained.

Several techniques have recently been introduced to calculate PC series. To classify these techniques as basic, they can be grouped under two headings: intrusive and non-intrusive. To explain the intrusive method, in this method, the circuit equations are extended to a larger system of equations and the resulting PC coefficients of the solution are obtained directly. In the non-intrusive method, the coefficient calculation is performed in two steps. In the first step, the behavior of the electronic circuit is calculated on the grid points in the space emitted by the random parameters via a typical commercial simulator.

In the second phase of the non-intrusive approach, the solutions at these grid points are brought together to calculate the PC series for a given circuit behavior state. For example, Galerkin projection is a stochastic technical intrusive technique. Stochastic collection-based technique is non-intrusive approach.

The stochastic Galerkin (SG) technique has a significant disadvantage. In this technique, there is an intense Jacobian matrix, and with this matrix a larger system of equations has to be solved. This disadvantage causes the technique to work with a small number of random variables on the problems. That is, the number of variables to be applied is limited.

This is not the case in non- intrusive SCs. In the first stage of the SC, the original circuit equations must be solved on independent grid points in the space constraints. This gives us an advantage. This is because it can work effectively on parallel platforms. In addition, this approach also allows factoring in a sparse Jacobian matrix typically coming from sparse circuitry. Moreover, recent developments have suggested increasing the effectiveness of the collocation approach by using the sparse grid in the random space [13].

However, the intrusive approach provides a significant advantage over the non-intrusive collocation based approach in determining the statistical characterization of the response is an output of circuit variables. These responses are in the time domain. This statistical characterization can only be calculated in discrete time-point sets. This fact is a challenge to the non-intrusive collocation approach. Indeed, the time step length with the simulator in said first stage can be adjusted to be adaptable with each new time point.

It is highly probable that the selected discrete time domain points for a grid point in a random field, it is different from those of other grid points. It is difficult to specify which discrete time points will be taken in the second phase. And these time points are required for the coefficient calculation. The interpolation approach is used for the discrete time domain point pack. However, this method can be a cause of failure for high-order PC coefficients [14].

There are advantages of the two methods. Combining the advantages of two methods is beneficial. This merge operation is given as the first suggestion in the collocation formulation [15]. This equation causes a larger system of equations. The Jacobian matrix can be converted. This transformation is becoming block diagonal form. Factorization operation can be applied for these diagonal blocks.

A Galerkin based intrusive method, which allows separating the Jacobian matrix, has been recently evaluated [16, 17]. This is intended to provide a formula for describing the reinforced Jacobian matrix. It also shows that this structure can parse independently into smaller matrices that can be factored. However, this approach is limited to PC problems that are limited by the simulation if they are defined by Hermite polynomials and only have a Gaussian distribution of indeterminate parameters.

1.2. Motivation

The main motivation of this work is based on observing the effect of critical circuit elements on electronic circuits. The types of uncertainty also vary because of the reduction in circuit element dimensions. To investigate these uncertainties, the Monte Carlo Method was applied to basic simulation methods. In addition, the Polynomial chaos method, which has some advantages over the Monte Carlo Method and has recently been published, has been another motivation for studying. The

uncertainty analysis study in this study should not only benefit the specific areas of the circuit simulation, but should also be applicable to the uncertainty measurement in other areas.

1.3. Contribution

This work provides details for the situation of very random parameters and provides the necessary evidence. The study also contributes to the use of simulation programs.

- **Organization of the Thesis**

The content of the thesis is given as follows: Chapter 2 provides the mathematical framework and necessary indications for this thesis. Chapter 2 also contains the background of the amount of uncertainty, including the theory of probability and the background of the random variable. General information and operating principles for the circuits studied can be found in Chapter 3. Numerical examples and numerical results are given in Chapter 4. And finally, conclusions and future work can be found in Chapter 5.

2. RANDOM DISTRIBUTIONS AND UNCERTAINTY METHODS

2.1. Probability Density Functions

Random variables take random values. The distribution of a random variable indicates the probability at which that variable will be taken. For example, when a fair die is tossed, it takes between 1 and 6 values. There is a possibility of taking one of these values. This value is $1/6$. In the light of these values, the column graph of the data distribution of the dice event and the probability distribution function are created.

The probability density and cumulative density functions of this dice event is shown in Figure 2.1 [18].

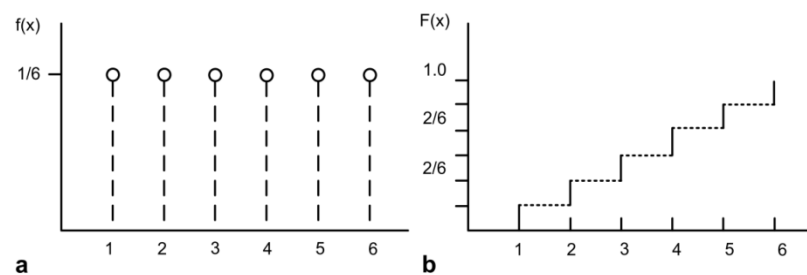


Figure 2.1: Probability functions of a discrete random variable for a dice tossing experiment. a) PDF. b) CDF.

Continuous random variable value data is shown below. Figure 2.2 (a) shows the continuous change in the hourly temperature values [18]. These temperature values were measured by a thermometer not affected by moisture. Temperature is the result of dry air. These values are the values of Philadelphia, PA.

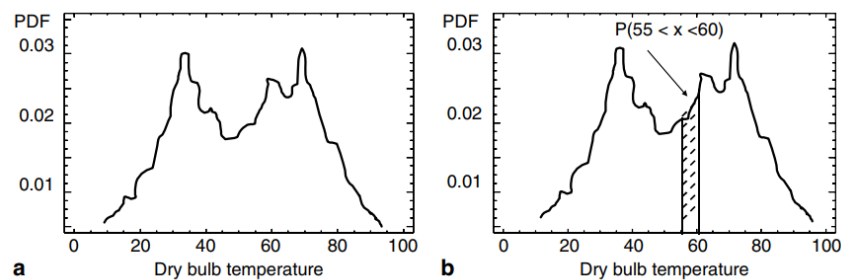


Figure 2.2: PDF for continuous random data a) PDF. b) Field representation of the probability value .

In the graph of continuous variable values, the hatched area gives the possibility that the temperature will remain within a certain temperature range. Figure 2.2 (b) stands for the possibility that the temperature is between 55F and 60F.

Figure 2.3 gives the CDF display according to the PDF results shown in Figure 2.2 [18]. From this graph, the probability that the temperature is below 60F can be read.

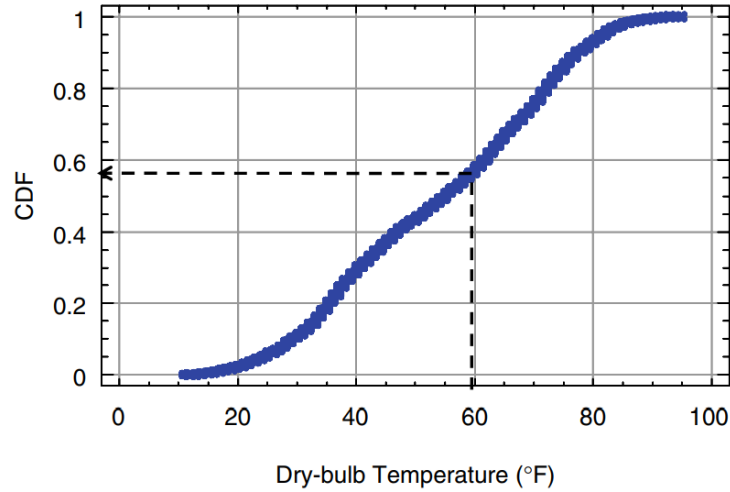


Figure 2.3: CDF according to the PDF in Figure 2.2.

It is not very useful to find a value for a single value in continuous variables. It would be more useful to find the possibility of certain intervals instead. Also, when the CDF result is extracted according to the PDF, the probability value of a certain value is determined. In Figure 2.3, the probability that the temperature is less than 60F is 58 percent.

Random variables can be discrete or continuous. In the experiments, the data were generally obtained as discrete. However, probability theories are constantly expressed in variable terms. Equations of probability distributions are as follows.

- The probability distribution function is always positive.

$$f(x) \geq 0 \quad -\infty < x < \infty \quad (2.1)$$

- The sum of the probabilities of all outputs must be one.

$$\int_{-\infty}^{\infty} f(x)dx = 1 \quad (2.2)$$

The area under $f(x)$ is (CDF) or $F(a)$ for $-\infty < x < a$:

$$F(a) = p\{X \leq a\} = \int_{-\infty}^a f(x)dx \quad (2.3)$$

The inverse state or relationship between $f(x)$ and $F(a)$ is as follows.

$$f(x) = \frac{dF(x)}{dx} \quad (2.4)$$

This causes the possibility shown below for an output between $a \leq X \leq b$ interval.

$$\begin{aligned} p\{a \leq X \leq b\} &= \int_a^b f(x)dx \\ &= \int_{-\infty}^b f(x)dx - \int_{-\infty}^a f(x)dx = F(b) - F(a) \end{aligned} \quad (2.5)$$

Since the PDF is defined in discrete values, the CDF increases step by step. In addition, for continuous variables, the CDF increases in a monotonic manner. There is a need to multiply a multiple random variable simultaneously. These functions can be expanded to evaluate this possibility.

The following shows the joint density function for continuous variables. This occurs as follows. It can be thought that two different random variables work together. The temperature variable and the form change variable can work together.

In other words, it can be thought of as the melting of the lead together with the change of temperature. If these variables are called x and y , then the probability function is expressed as $f(x, y)$.

$$f(x, y) \geq 0 \text{ for all } (x, y) \quad (2.6)$$

$$\int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f(x, y)dx dy = 1 \quad (2.7)$$

$$p[(X, Y) \in A] = \iint_A f(x, y) dx dy \quad (2.8)$$

A is any area in the xy plane for formula 2.8.

If the x and y variables are independent random variables, the joint PDF values are as follows. The margins of these variables are multiplied as shown below.

$$f(x, y) = f(x)f(y) \quad (2.9)$$

Variables are continuous and that gives joint probability. The x and y random variables are two jointly distributed variables. The marginal distribution of x is basically the probability distribution of x ignoring y. X is as shown below.

$$g(x) = \int_{-\infty}^{\infty} f(x, y) dy \quad (2.10)$$

X and Y were reported to be 2 jointly distributed random variables. Where X is equal to x, the conditional probability distribution of X is as follows.

$$f\left(\frac{y}{x}\right) = \frac{f(x, y)}{g(x)} \quad g(x) > 0 \quad (2.11)$$

2.2. Probability Distributions Used in the Study

The probability distributions describe the possibility of experiencing a situation. There are various types of probability distribution. The binomial distribution that physicists often use is an example. Gauss or Normal distributions are often used to produce the result distribution against small errors that may occur when taking measurements.

In this study, truncated, beta and uniform distribution types were studied. Section 4 contains application details. Most of the important and widely used species among the distribution types are shown in Figure 2.4 with a practical representation.

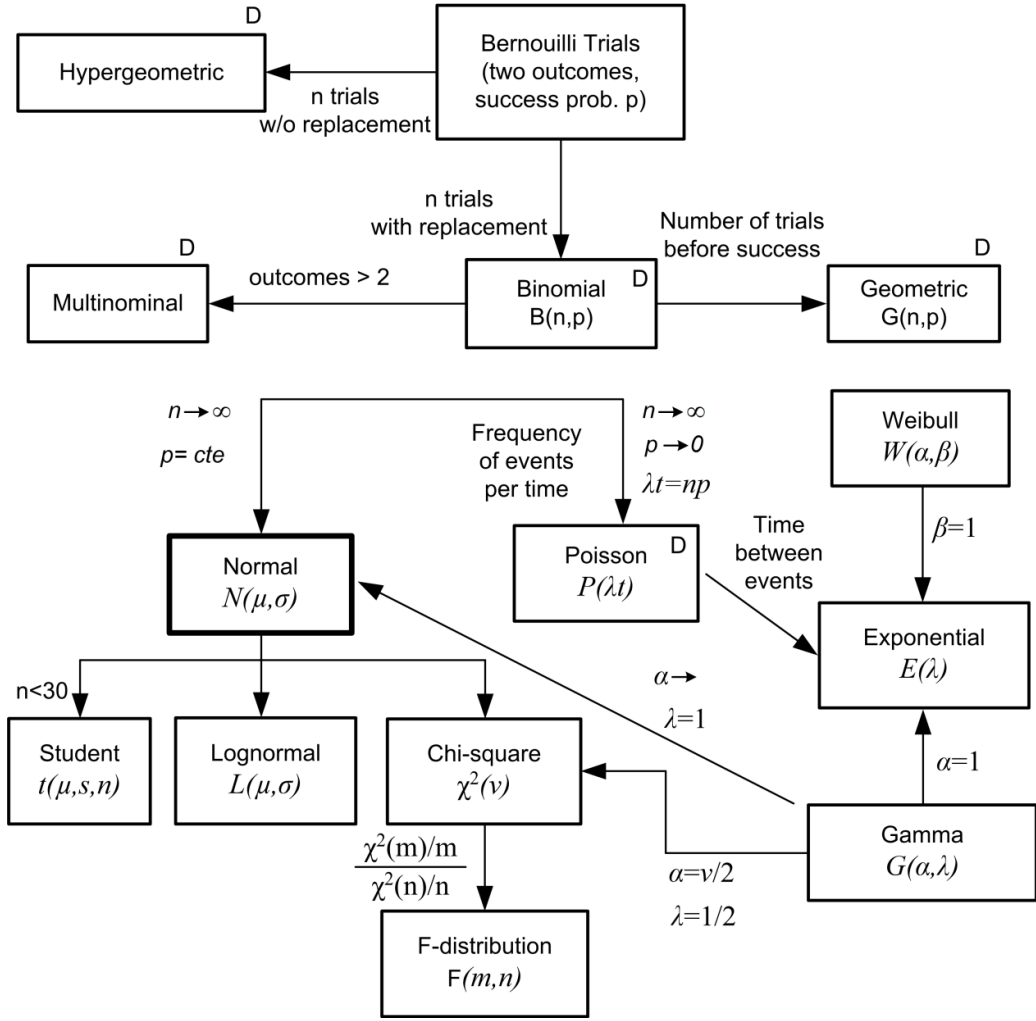


Figure 2.4: Representation of the link between significant probability distribution functions .

In Figure 2.4, those marked with D indicate discrete functions [18].

2.2.1. Uniform Distribution

The probability of occurrence of events occurring in a certain interval (a,b) is equal to this distribution. A random number by people can be the analogous analogy to this distribution of things called.

The PDF of this distribution (U (a, b)) is below.

$$f(x) = \frac{1}{b-a}; \quad a \leq x \leq b \quad (2.12)$$

CDF of this:

$$F(x) = \begin{cases} 0 & x < a \\ \frac{x-a}{b-a} & a \leq x \leq b \\ 1 & x > b \end{cases} \quad (2.13)$$

The first two moments:

$$E[X] = \frac{a+b}{2}; \quad Var(X) = \frac{(b-a)^2}{12}; \quad SD(X) = \sqrt{\frac{(b-a)^2}{12}} \quad (2.14)$$

2.2.2. Truncated Distribution

This distribution is similar to normal distribution. The mean value is indicated by μ . standard deviation is indicated by σ . range values can be selected. Take and upper limits can be set. This distribution has 4 parameters.

- μ : mean.
- σ : standard deviation.
- a: lower x-value (to min $-\infty$).
- b: upper x-value (to max ∞).
-

The following graph gives the probability density of only half of the normal curve. In addition, this distribution is limited to 0 to infinity and x values are in this range.

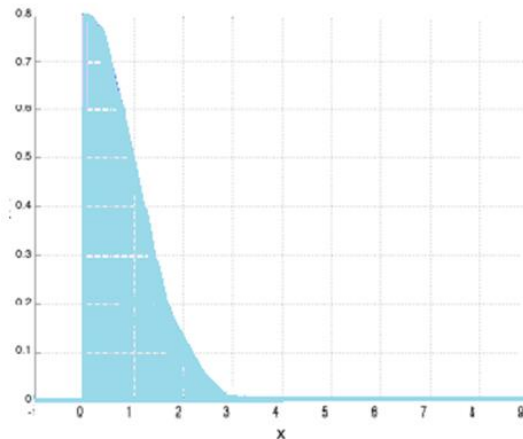


Figure 2.5: A standard normal distribution which is lower truncated.

Below is the mean formula of the truncated distribution. The following formula applies where the standard normal distribution is ϕ .

$$\mu + \frac{\phi(\alpha) - \phi(\beta)}{Z} \sigma \quad (2.15)$$

$$\psi(\bar{\mu}, \bar{\sigma}, a, b; x) = \begin{cases} 0 & \text{if } x \leq a \\ \frac{\phi(\bar{\mu}, \bar{\sigma}; x)}{\Phi(\bar{\mu}, \bar{\sigma}^2; b) - \Phi(\bar{\mu}, \bar{\sigma}^2; a)} & \text{if } a < x \leq b \\ 0 & \text{if } b \leq x \end{cases} \quad (2.16)$$

a ve b are truncate area, $\bar{\mu}$, $\bar{\sigma}$ are parameters of the distribution.

Output Point: The probabilities for values outside area and the rest are updated. By using formula 2.17, expected value (μ) and variance (σ) are found with formula 2.18 and 2.19 respectively.

$$\alpha = \frac{a - \bar{\mu}}{\bar{\sigma}}; \quad \beta = \frac{b - \bar{\mu}}{\bar{\sigma}} \quad (2.17)$$

$$\mu = \bar{\mu} - \bar{\sigma} \frac{\phi(0, 1; \beta) - \phi(0, 1; \alpha)}{\Phi(0, 1; \beta) - \Phi(0, 1; \alpha)} \quad (2.18)$$

$$\sigma^2 = \bar{\sigma}^2 \left(1 - \frac{\beta \phi(0, 1; \beta) - \alpha \phi(0, 1; \alpha)}{\Phi(0, 1; \beta) - \Phi(0, 1; \alpha)} - \left(\frac{\phi(0, 1; \beta) - \phi(0, 1; \alpha)}{\Phi(0, 1; \beta) - \Phi(0, 1; \alpha)} \right)^2 \right) \quad (2.19)$$

Expected value and variance of two distributions are given in Figure 2.6. For example, expected value of the distribution with red line is 2 and variance of this distribution is 1.5.

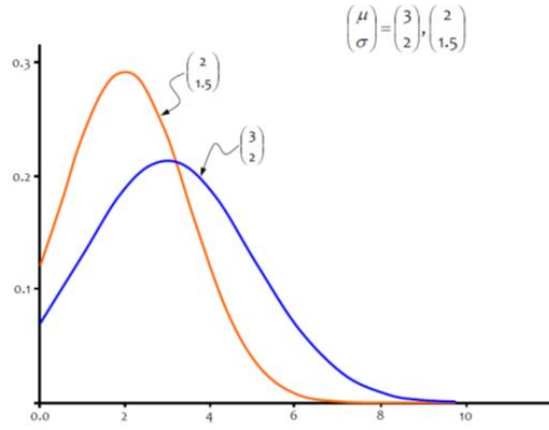


Figure 2.6: Truncated normal distribution PDF.

2.2.3. Beta Distribution

These types of distribution are useful when the data is finite range and constant. Below is the PDF of Beta distribution.

$$f(x) = \frac{x^{a-1}(1-x)^{b-1}}{\beta(a,b)}; \quad 0 \leq x \leq 1 \quad (2.20)$$

The function $\beta(a,b)$ is the Beta function. This function is below in equation 2.21.

$$\begin{aligned} \beta(a,b) &= \int_0^1 x^{a-1}(1-x)^{b-1} dx \\ &= \frac{\Gamma(a)\Gamma(b)}{\Gamma(a+b)} \end{aligned} \quad (2.21)$$

The CDF result of this distribution is not in proper form. CDF notation can be achieved using other methods. The CDF result is not uniform in other continuous distributions. The first two moments of this distribution are below in equation 2.22.

$$E[X] = \frac{a}{a+b}; \quad \text{Var}(X) = \frac{ab}{(a+b+1)(a+b)^2} \quad (2.22)$$

2.3. Uncertainty Methods

2.3.1. Monte Carlo Method

A significant number of simulation models are probabilistic models. Analytical solution of systems consisting of random variables defined by probability distributions becomes difficult. The Monte Carlo technique is a technique for generating a sample from a probability distribution with the help of random numbers in the operation of a simulation model. So the Monte Carlo technique is not a type of simulation. It is a mathematical process used in simulations.

A variable that characterizes a demand, a supply, or an operation in the Monte Carlo process is grouped by a determined probability distribution. The determination of which group a random number falls into is the Monte Carlo process. Depending on this determination, various interpretations and conclusions can be obtained for the system or event being considered. So when such studies are named, for example; the expression "number of pi according to Monte Carlo Method" is used.

The Monte Carlo method is a simulation method that uses random samples to predict the outcome of an experiment. The Monte Carlo Method provides a good indication of the statistical direction of the simulation technique.

It is used especially in the numerical and random acquisitions of the integrals of functions where direct analytical approaches are not possible.

- Finding the number of pi by Monte Carlo Method

Let's draw a square whose edges are in unit length ($R = 1$). The area of the square is (1x1). The base of the square is x-axis, and the left-hand side is the y-axis. Place a circle quadrangle with a radius of unit length inside the square.

The equation of the circle in Figure 2.7 is $x^2 + y^2 = R^2 = 1$. Assume that x and y are the numbers of the points found by generating N random numbers and count them in the circle and denote it by the number s. This is the radius of the random number pair.

$$R_{xy} = \sqrt{x^2 + y^2} \quad (2.23)$$

This value is compared to the value of the defined field $R = 1$; It is understood that the minor is inside the circle, the bigger is outside the circle.

Let's show the quarter circle area with A . The resulting s/N ratio equals the ratio of $s/N = A/(R \times R)$ = area of the quarter circle to area of the square.

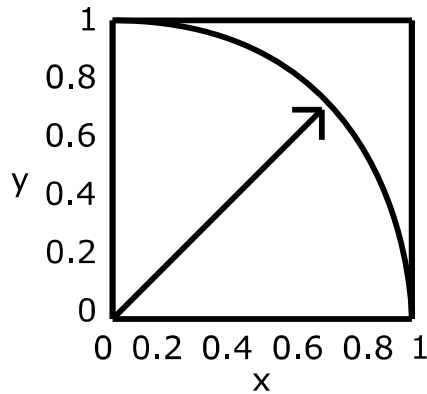


Figure 2.7: Using Monte Carlo Method to find square and quadrant area and pi number.

From this it can be estimated that the number of $\pi = 4A/(R \times R) = 4s/N$ if the full circle area ($4A$) and hence $4A = \pi(R \times R)$

- Finding the number of π in MATLAB:

Create a square with $R = 1$ edges and $N = 1000$ random (x, y) points for a quarter circle with radius $R = 1$ drawn into it.

$$R_{xy} = \sqrt{x^2 + y^2}$$

Let's count the circles of these points, that is, those that satisfy the condition $R_{xy} < R$, and denote it by the number s . From here

$$\text{Pi number} = 4A / (R \times R) = 4s / N \quad (2.24)$$

Let's find it. The larger the N value, the closer it can be to the π number. The corresponding MATLAB code is below.

```
% pi number calculation with Monte Carlo methods
% n number is repating number, circle radius is r=1, full circle area=pi
N=10000000000; c=0; r=1;
```

- for k=1:N
- x=rand;y=rand;
- Z=sqrt(x^2+y^2);
- if Z<r
- c=c+1;
- else
- c=c;
- end
- end

- A=(r*r)*(c/N);% this gives quarter circle area. we know $A=\pi*r*r/4$
- pi_number=4*A

2.3.2. Generalized Polynomial Chaos Method

Polynomial Chaos was first introduced in 1938 by Wiener [19]. This method is in the field of uncertainty measurement and is a spectral expansion technique with practice. The stochastic process uses Hermite polynomials with Gaussian variables to separate deterministic and nondeterministic fragments. In 1947, Cameron and Martin demonstrated that this expansion came together in a mean square for square integrable random processes [20]. Ghanem and Spanos [21] initiated the Polynomial Chaos application in 1991 with a Galerkin projection in the light of the finite element method in the field of solid mechanics. In the next years, the projection method became more popular, leading to a wider application area such as CFD (eg, [22, 23]). Xiu and Karniadakis [8, 24] proposed a generalization of Hermitian Chaos for other probability distribution classes by establishing a relationship between the probability density function and the hyperfunctional series and the weight function of orthogonal polynomials in 2002. In 2010, Ernst et al. [25] have demonstrated the convergence of the generalized Polynomial Chaos (gPC) for specific probability distributions.

In 2006, Wan and Karniadakis [26, 27] were able to cope with strong nonlinear dependencies on random inputs. They developed a multi-headed polynomial Chaos method (MEgPC) to come up from the weaknesses of GPC in terms of accuracy. For example, when the application involves long-term integration or stochastic discontinuity. The basic idea is to separate the probability area of the stochastic entropy and solve independent local problems. ME-gPC proved effective in reducing the degree of non-linear dependencies. But there are still some restrictions on long-term integration. For balanced limit cycles, Le Maître et al. [28, 29] proposed

improved asynchronous time integration for gPC to capture solution dynamics. [30] developed dynamic orthogonal field equations. These equations derive equations to calculate an optimal basis with an orthogonality criterion in a deterministic and a stochastic field. In his articles in 2010, Gerritsma et al. [31] applies a discrete time-dependent approach (TD-gPC) for ordinary differential equations subject to evenly spaced random inputs. This made it possible to obtain good accuracy with a possible overhead for numerical costs. The goal of this method is to reduce nonlinear dependencies by a series of new random variable base transformations defined by the solution itself at certain discrete time intervals. The authors led something in 2011. It was a combination leader. Here both the ME-gPC and TD-gPC were combined in a hybrid manner [32, 33]. The key idea is to reduce the numerical cost by exploiting the insignificant parallelization construction of ME-gPC. And usage of time-dependent basis functions in each element arising from the decomposition of the probability of the random.

The basic idea in the gPC starts by expanding $\mathbf{X}(\xi; s)$ in a set of orthogonal polynomials. To facilitate presenting the core idea of the gPC, we assume first that the underlying random space is uni-dimensional, i.e., that $d = 1$, where ξ , the vector of random variable is replaced by the scalar RV, ξ . In this case, the expansion of $\mathbf{X}(\xi; s)$ in the orthogonal polynomial takes the following form (2.25)

$$\mathbf{X}(\xi, s) = \sum_{i=0}^{\infty} \hat{X}_i(s) \phi_i(\xi) \quad (2.25)$$

where $\phi_i(\xi)$ is an i -th degree polynomial in ξ , and $\hat{X}_i(s)$ is an s -dependent coefficients vectors with dimension N that need to be computed.

The set of polynomials $\phi_i(\xi)$ are orthogonal polynomials with respect to a certain weighting function, $w(\xi)$, over the domain of the RV, ξ , i.e.

$$\langle \phi_i(\xi), \phi_j(\xi) \rangle = \int_{\Omega} \phi_i(\xi) \phi_j(\xi) w(\xi) d\xi = \delta_{ij} \theta_i \quad (2.26)$$

where Ω is the support of the polynomial, θ_i is the so-called inter-product, which is a constant whose value depends on the degree i and type of the polynomials and δ_{ij} is the Kronecker delta function defined by (2.27):

$$\delta_{i,j} = \begin{cases} 1, & i = j \\ 0, & \text{otherwise} \end{cases} \quad (2.27)$$

The key principle in the gPC is that basic statistical information of $\mathbf{X}(\xi; s)$ can be extracted from the coefficients, $\hat{X}_i(\xi; s)$, directly. To explain this principle, assume that ξ is standard Gaussian RV, where its probability density function is given by (2.28);

$$p(\xi) = \frac{1}{\sqrt{2\pi}} e^{-\frac{\xi^2}{2}} \quad (2.28)$$

and let $\phi_i(\xi)$ be given by the probabilistic Hermite polynomial of the i th degree, i.e.

$$\phi_i(\xi) \equiv H_i(\xi) = (-1)^i e^{\frac{\xi^2}{2}} \frac{d^i}{d\xi^i} e^{-\frac{\xi^2}{2}} \quad (2.29)$$

It is well-known [34] that the Hermite polynomials are orthogonal with respect to the weighting function $w(\xi)$ that is given by

$$w(\xi) = e^{-\frac{\xi^2}{2}} = \sqrt{2\pi} p(\xi) \quad (2.30)$$

As probability theory states, the mean value of $\mathbf{X}(\xi, s)$ is obtained from (2.31);

$$E\{X(\xi, s)\} = \int_{\Omega} X(\xi, s) p(\xi) d(\xi) \quad (2.31)$$

Now substituting from the expansion in (2.25) and (2.30) into (2.31) and manipulating, using the fact that $H_0(\xi) \equiv 1$, shows that the mean value is given by (2.32):

$$\begin{aligned}
E\{X(\xi, s)\} &= \int_{\Omega} \sum_{i=0}^{\infty} \hat{X}_i(s) \phi_i(\xi) \frac{1}{\sqrt{2\pi}} w(\xi) d\xi \\
&= \frac{1}{\sqrt{2\pi}} \sum_{i=0}^{\infty} \hat{X}_i(s) \langle \phi_i(\xi), \phi_0(\xi) \rangle \\
&= \frac{\theta_0}{\sqrt{2\pi}} \sum_{i=0}^{\infty} \hat{X}_i(s) \delta_{0,i} \\
&= \frac{\theta_0}{\sqrt{2\pi}} \hat{X}_0(s)
\end{aligned} \tag{2.32}$$

where the orthogonality of the Hermite polynomial, with respect to the weight $e^{-\frac{\xi^2}{2}}$ has been used to obtain the last equality. In a similar manner, the variance of $X(\xi, s)$ is extracted from the coefficients $\hat{X}_i(s)$ as can be seen from the following manipulation (2.33)

$$\begin{aligned}
\text{var}\{X(\xi, s)\} &= E\{X^2(\xi, s)\} - E^2\{X(\xi, s)\} \\
&= E\left\{\left(\sum_{i=0}^{\infty} \hat{X}_i(s) \phi_i(\xi)\right)^2\right\} - \mu^2 \\
&= \frac{1}{\sqrt{2\pi}} \int_{\Omega} \left(\sum_{i=0}^{\infty} \sum_{j=0}^{\infty} \hat{X}_i(s) \hat{X}_j(s) \phi_i(\xi) \phi_j(\xi) w(\xi) d\xi\right) - \mu^2 \\
&= \frac{1}{\sqrt{2\pi}} \sum_{i=0}^{\infty} \sum_{j=0}^{\infty} \hat{X}_i(s) \hat{X}_j(s) \langle \phi_i(\xi), \phi_j(\xi) \rangle - \mu^2 \\
&= \frac{1}{\sqrt{2\pi}} \sum_{i=0}^{\infty} \sum_{j=0}^{\infty} \hat{X}_i(s) \hat{X}_j(s) \delta_{i,j} \theta_i - \mu^2 \\
&= \frac{1}{\sqrt{2\pi}} \sum_{i=0}^{\infty} \hat{X}_i^2(s) \theta_i - \mu^2 \\
&\approx \frac{1}{\sqrt{2\pi}} \sum_{i=0}^k \hat{X}_i^2(s) \theta_i - \mu^2
\end{aligned} \tag{2.33}$$

The main point to take out from the above discussion is that the coefficients $\hat{X}_i(s)$ are the key to computing the statistical properties of $X(\xi; s)$.

There are three main questions that need to be answered to fully describe the gPC approach.

- How can one deal with other standard RVs which are not Gaussian in their pdf?
- How can the gPC be generalized to multi-random parameter, i.e., for the case where $d > 1$?
- How can we compute the coefficients $\hat{X}_i(s)$ in a computationally efficient manner?

3. CIRCUITS USED IN THE THESIS

This section provides information about buck converter, dynamic NAND gate, Gilbert cell, Transmission gate chain, inverter chain and Class D amplifier circuits. After giving theoretical information about the related circuits, simulation studies have been done with circuits. Simulation results were performed for each circuit and the simulation results were shared.

First of all, the netlist of the circuit to be simulated are prepared in the spice language which will be understood by NGSPICE simulation tool. In some places, the prepared netlists were sent directly to the NGSPICE tool and the necessary calculation results were obtained. In some cases, netlist has been sent to NGSPICE via OCTAVE or MATLAB. Plot results were obtained as MATLAB or OCTAVE figures and made clearer. LTSPICE tool was used to prepare the schematic representations of the circuits. LINUX operating system was used as the working environment.

In the other part, numerical analysis studies were performed for Class D amplifier and dynamic NAND gate circuits. These studies will be discussed in detail in the other chapter.

3.1. Buck Converter

Switched-mode buck converter sends energy to the output periodically. In a part of the period, the inductor and capacitor store this energy. In the other part of the period it uses this stored energy. The buck converter only allows the output voltage to be less than the input voltage.

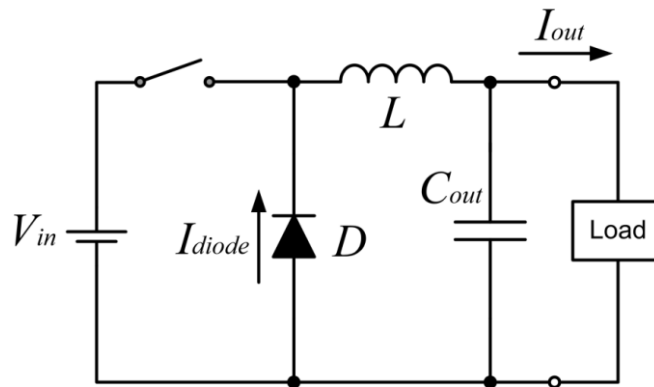


Figure 3.1: Buck converter schematic.

The switch in the circuit in Figure 3.1 opens and closes periodically. The switch is active during the t_{on} , inactive during the t_{off} time. One period of the circuit is $T = t_{on} + t_{off}$. The duty cycle of the circuit is $D = t_{on}/T$. D indicates the circuit's on-rate. There is ripple in the output of the circuit. This ripple is called small-ripple approximation and is generally simplified and the calculations are simplified. Let's examine the behavior of the circuit in the open and closed state.

- On Phase:

When the switch is activated, the power supply is connected to the inductor, the diode is reversed polarized. The circuit in Figure 3.2 works as follows.

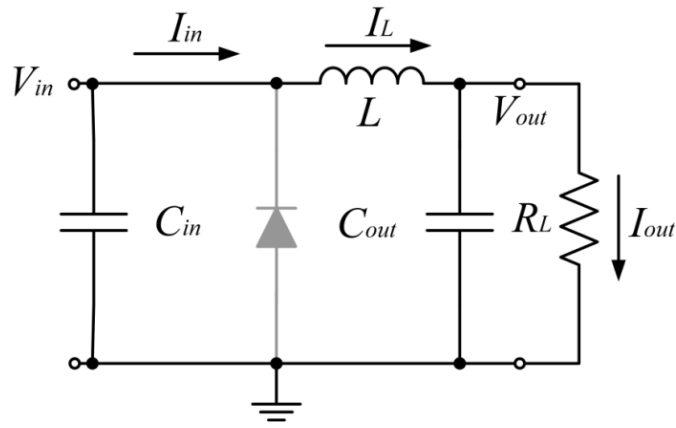


Figure 3.2: Buck converter schematic on phase.

The following formula (3.1) gives an increment slope of the inverter when the switch is active.

$$\frac{dI_L}{dt} = \frac{V_{in} - V_{out}}{L} \quad (3.1)$$

The following formula (3.2) gives the amount of change of the current of the inductor when the switch is active.

$$\Delta I_{L+} = \frac{V_{in} - V_{out}}{L} t_{on} = \frac{V_{in} - V_{out}}{L} DT \quad (3.2)$$

- Off Phase:

When the switch is deactivated in Figure 3.3, the connection has been changed and the inductor is connected to the ground with the diode.

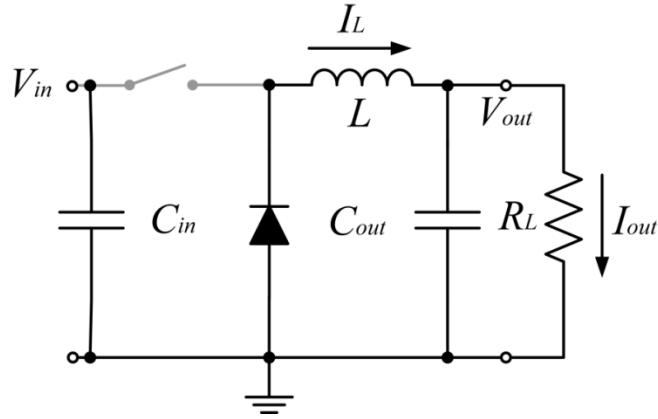


Figure 3.3: Buck converter schematic off phase.

The current level through the inductor begins to decrease. The slope of this decline is in the formula (3.3) below.

$$\frac{dI_L}{dt} = \frac{0 - V_{out}}{L} \quad (3.3)$$

Two different conditions may occur in the light of the above. One of them is continuous; the other is the discontinuous mode. When the circuit is closed the inductor current never drops to zero; this mode is persistent.

However, if the inductor current is reduced to zero at certain points during the period when the circuit is off, this is the discontinuous mode.

Buck converter simulation:

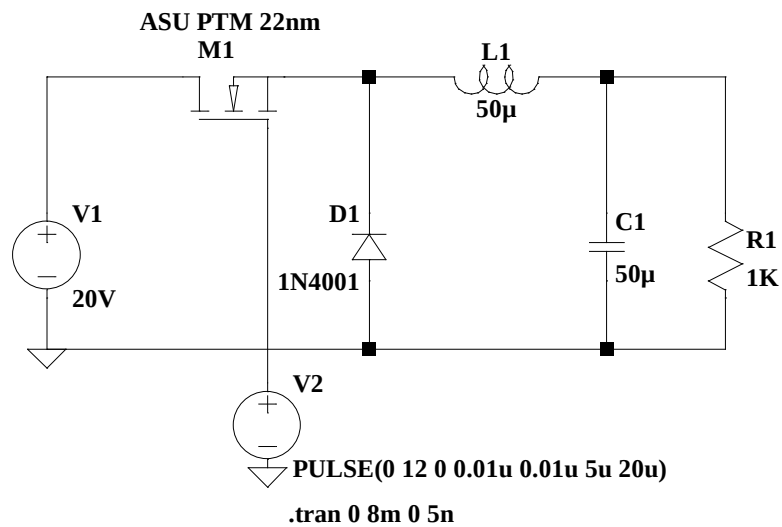


Figure 3.4: Buck converter schematic in LTSPICE tool.

The spice netlist of the circuit was created to simulate the buck converter circuit shown in Figure 3.4. The necessary arrangements have been made for circuit elements, supply voltage, trigger signal in the netlist. Arizona State University (ASU) PTM 22nm MOSFET model was used as the switching element in the netlist. In transient analysis, the step interval was determined as 5ns and the analysis was terminated at 8ms.

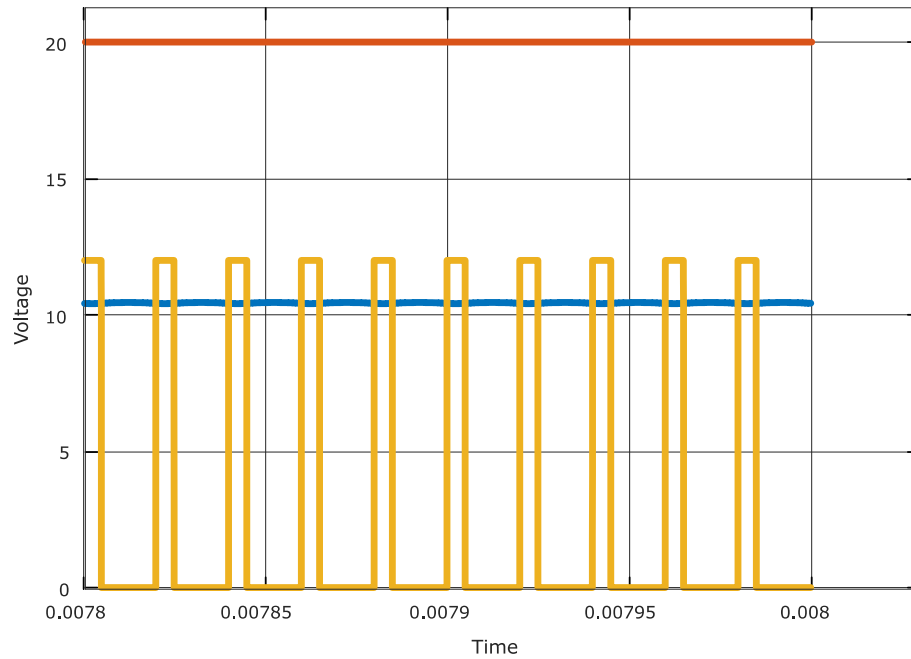


Figure 3.5: Buck converter output for switch state, supply voltage and load voltage.

Figure 3.5 shows the simulation results of netlist for buck converter. The output with yellow indicates the status of the switch. The red output is the supply voltage. The blue signal is the load voltage at the output. As can be seen from Figure 3.5, when the supply voltage is around 20V, the output voltage is reduced to about 10.5V.

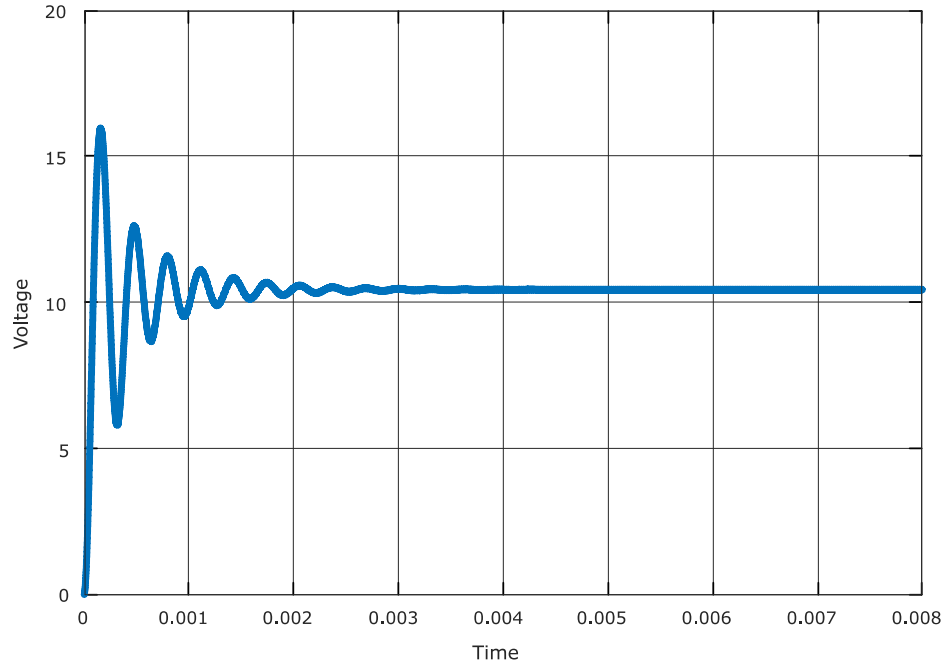


Figure 3.6: Buck converter load voltage output.

Figure 3.6 only shows the voltage on the load. The output voltage can be seen in more detail in Figure 3.6. The output voltage was first oscillated and then fixed at 10.5 V.

3.2. Dynamic NAND Gate

Dynamic NAND gate has problems such as charge sharing, charge leakage backgate coupling, and clock feedthrough. For example, if the MN1 is ON and the MN2 is in the OFF state of the desired function, it will cause the charge stored in the parasitic capacitor in the ndp node to be shared with the parasitic capacitor in node ns1 as seen in Figure 3.7. However, if the time between two consecutive pre-charge phases is not short enough, the charges stored in these capacitors will start to decrease due to the leakage current. In this study, how the effects of charge sharing and charge leakage vary depending on random parameters is analyzed. Before the analysis, the changes in the nominal parameters ndp and ns1 voltages were plotted and the time difference was calculated as 1mV.

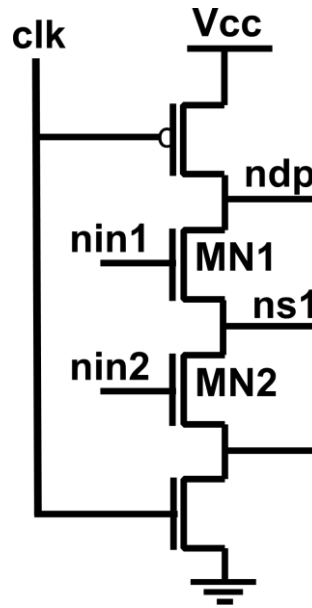
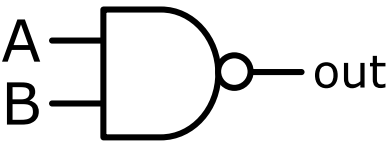


Figure 3.7: Dynamic NAND gate.

In this section, the charge sharing status occurring in the dynamic NAND gate circuit will be examined. The truth table of the dynamic NAND gate circuit and its symbol are given in Table 3.1.

Table 3.1: Dynamic NAND gate symbol and truth table.

Symbol	Truth Table		
	B	A	Q
	0	0	1
	0	1	1
	1	0	1
	1	1	0
Boolean Expression $Q = \overline{A \cdot B}$	Read as A AND B, gives NOT Q		

- **Charge Leakage:** Figure 3.8 shows the charge leakage status [35]. Here, it is seen that the current passes through the gate A without being desired. It is also seen that the circuit operates in two phases according to the clock state. These sections are called pre-charge and evaluation phases. In the pre-charge section, the clock is zero and the V_{out} output reaches the V_{DD} supply voltage. Then, when the clock is turned to high, V_{out} is disconnected from the V_{dd} and the voltage value of

V_{out} is slightly reduced even though the element A is off. The reason for this drop is the charge is leaked.

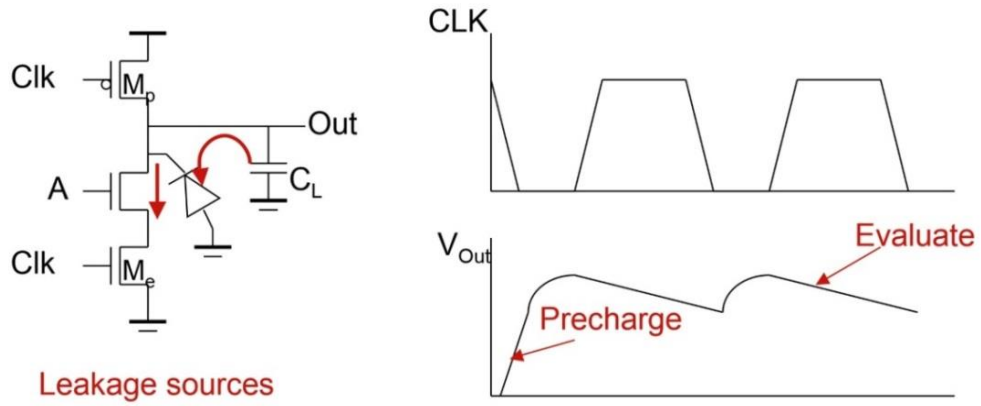


Figure 3.8: Charge leakage in dynamic CMOS .

- Charge Sharing: Dynamic NAND gate circuit is given in Figure 3.9. The pre-charge and evaluation phases mentioned in the previous section are also present. It reaches the voltage supply voltage at the out point with pre-charge.

In Figure 3.9, charge sharing starts when the circuit returns to the evaluation stage, that is, when the clock returns to high and the gate A is set [35]. In other words, when $A = 1$ and $B = 0$, there is a charge sharing between the C_L and the C_A parasitic capacitances. This sharing is undesirable and reduces the robustness of the circuit.

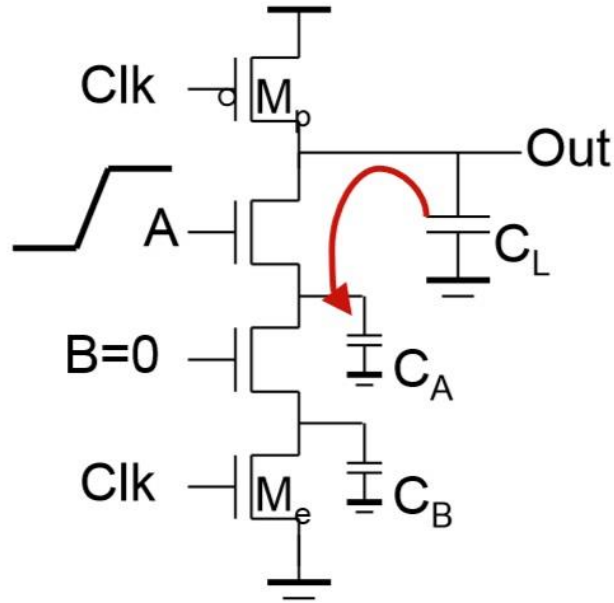


Figure 3.9: Charge sharing in dynamic CMOS.

Simulation 1: Charge sharing simulation in dynamic CMOS

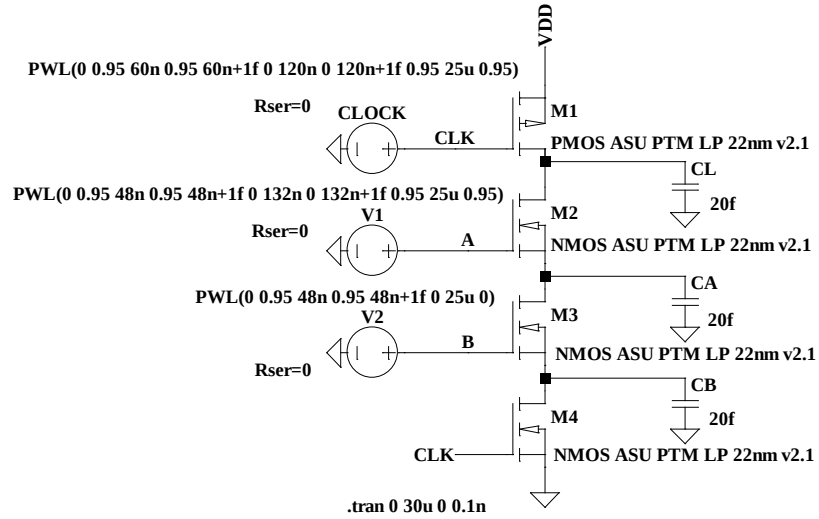


Figure 3.10: Dynamic NAND gate circuit schematic.

The required netlist was created in accordance with the circuit diagram in Figure 3.10. ASU PMOS and NMOS models were used for four MOSFET. NMOS width was determined as 88nm. The width of PMOS was 132nm. Clock and input1 (A) signal were determined by taking into account the precharge and evaluation conditions. The calculation command for finding the time between the output load

and the parasitic capacitance is 1mV was added to the netlist. The simulation was run after the spice netlist file was prepared. Figure 3.11 shows the dynamic NAND gate simulation result.

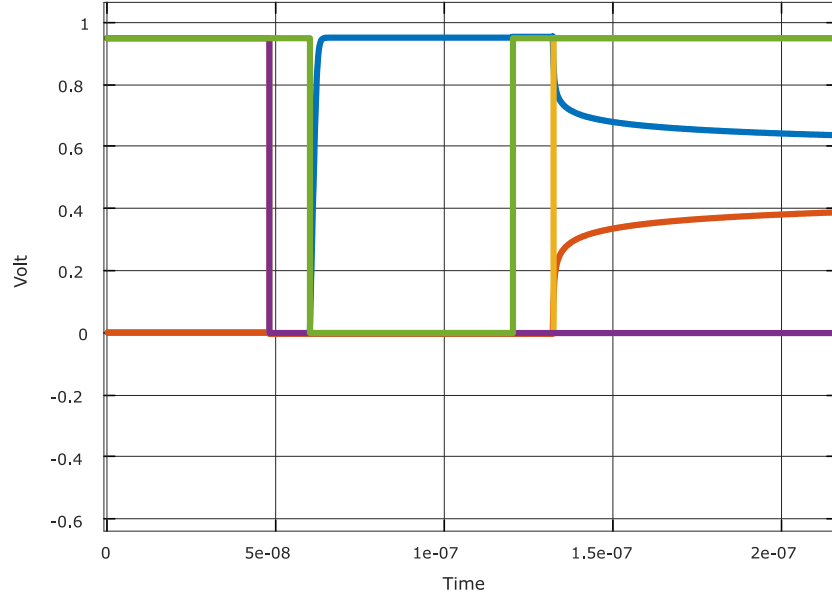


Figure 3.11: Dynamic NAND gate simulation result with all signals.

Figure 3.11 shows five different signals. The green signal is clock, the yellow is the first input, and the purple is the second input signal. The blue signal is the voltage curve on the C_L in Figure 3.10 and the red signal is the voltage curve on the C_A in Figure 3.10. The green clock signal is low in the period of precharge. In this period, the C_L voltage was maximized because PMOS was active. When the green clock signal is high, the evaluation period starts, and when yellow first input signal is high, charge sharing between C_L and C_A has started. While the amount of blue C_L voltage decreases, the red C_A voltage increases. In other words, the topmost PMOS in Figure 3.10 is not active, the NMOS below is active. But the second NMOS below it is not active. It is seen in Figure 3.11 that the second input purple signal is constantly low during this period.

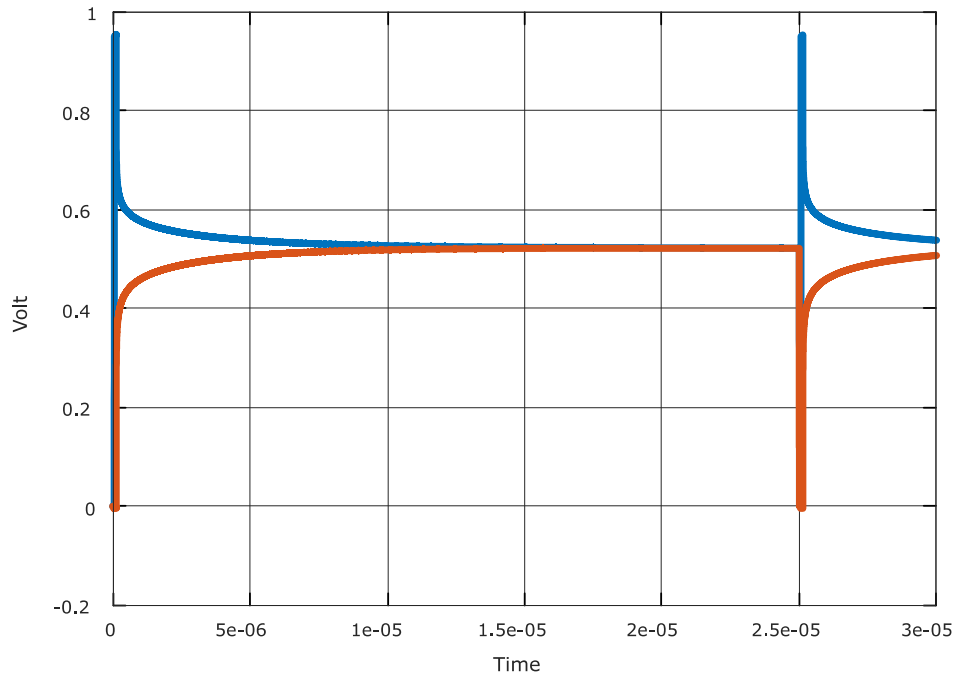


Figure 3.12: Dynamic NAND gate charge sharing result.

Figure 3.12 shows the alteration of the voltage on the load at the output of the dynamic NAND gate. The blue signal is C_L output voltage in Figure 3.12. The other red signal is the voltage curve above the parasitic capacitance C_A on the dynamic NAND gate in Figure 3.12. From the Figure 3.12, it is understood that the voltage at the output C_L load is absorbed by the parasitic capacitance of the NAND gate. This is a condition that can impair the behavior and accuracy of the circuit. One way to avoid such situations is to select the frequency of the clock signal.

```
Measurements for Transient Analysis
teval3          = 1.96405e-05
ngspice 2 -> █
```

Figure 3.13: Time calculation of 1mV voltage difference.

Figure 3.13 shows the time value when the voltage difference between C_L and C_A is 1mV. It was observed that the voltage on the C_L decreased and the voltage on the C_A increased during the charge sharing process. In the study, it was assumed that the load sharing ended when the voltage difference between C_L and C_A was 1mV. For this reason, as shown in Figure 3.13, when the voltage difference is 1mV, the time is calculated and the result is 1.96us. Also a gPC analysis study was performed

on dynamic NAND. This study is shared in Section 2.9 and Section 2.10 where Monte Carlo and gPC calculations are performed.

3.3. Gilbert Cell Mixer

Using analog signals to perform operations, instrumentation, communication, and control systems is much needed. Some of these processes are rectification, modulation, demodulation, frequency translation, multiplication and division. In this section, multiplication and phase detection is given.

In analogue signal processing, circuits having two analog input signals and having output proportional to the products of these inputs are often used. Such circuits are called analogue multiplier circuits. In the following sections, multiplier circuits will be mentioned starting from emitter coupled pair circuit.

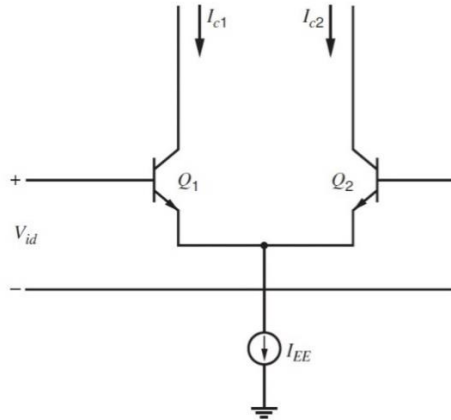


Figure 3.14: Emitter coupled pair.

The emitter coupled circuit is shown in Figure 3.14 [36]. The difference in the current flow through the transistors varies according to the input voltage as seen in formula (3.4).

$$I_{C1} = \frac{I_{EE}}{1 + \exp(-V_{id}/V_T)} \quad I_{C2} = \frac{I_{EE}}{1 + \exp(V_{id}/V_T)} \quad (3.4)$$

$$\Delta I_C = I_{C1} - I_{C2} = I_{EE} \tanh(V_{id}/2V_T)$$

The input voltage and the change curve of the currents passing through the transistors are given below in Figure 3.15 [36]. An emitter coupled circuit can be seen to be a simple multiplier.

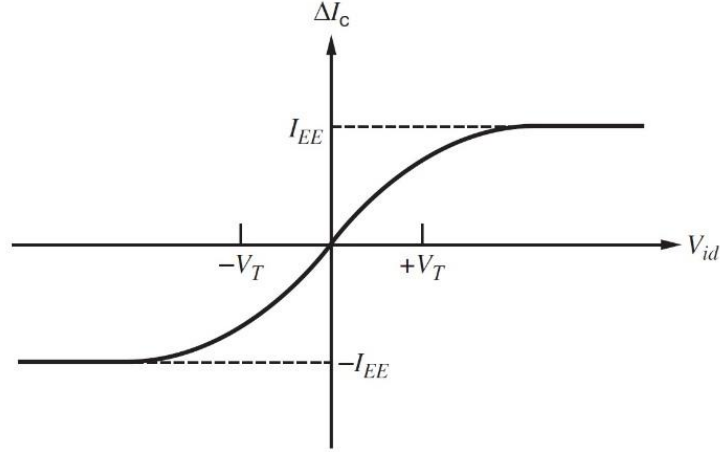


Figure 3.15: The DC transfer feature of the emitter-coupled pair.

Considering $V_{id}/2V_T \ll 1$, the formula (3.5) will be as follows.

$$\Delta I_C = I_{EE} \times (V_{id}/2V_T) \quad (3.5)$$

The I_{EE} current is a bias current for this circuit. With the addition of an additional circuit, the I_{EE} current can be proportional to the second input signal. The circuit prepared with the additional circuit is shown below. The following circuit given in Figure 3.16 is called as two quadrant analog multiplier [36].

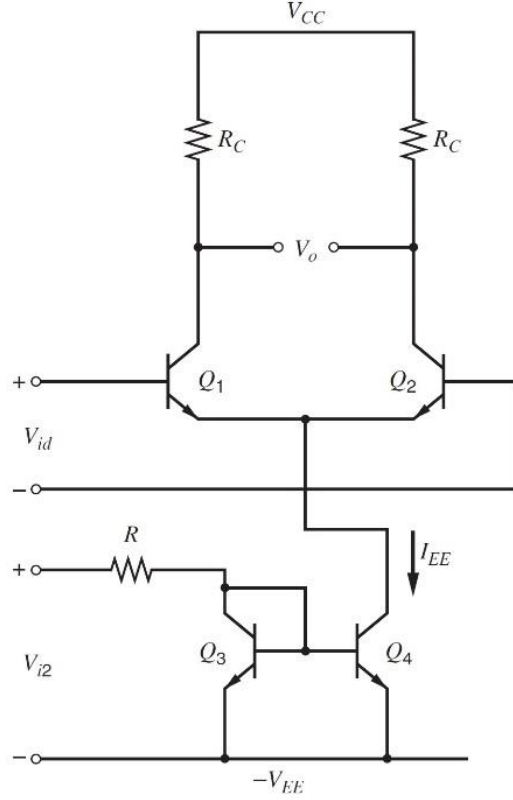


Figure 3.16: Two quadrant analog multiplier.

With the above circuit, the above formula can be formed as follows (3.6).

$$I_{EE} \cong K_0 (V_{i2} - V_{BE(on)}) \quad (3.6)$$

The current change in the emitter couple is found in the following formula (3.7).

$$\Delta I_C \cong \frac{K_0 V_{id} (V_{i2} - V_{BE(on)})}{2V_T} \quad (3.7)$$

A multiplier circuit was created under conditions where V_{id} is small and V_{i2} is greater than $V_{BE(on)}$. However, these two quadrant analog multiplier circuits operate as multiplier in the V_{id} - V_{i2} plane in only two quarters. The function of the circuit in two quarters is an important limitation for most applications. Therefore, the circuits used as multiplier are practically four-quadrant. The circuit shown in Figure 3.17 is a Gilbert cell circuit that works as a four-quadrant multiplier and is an alteration of the emitter coupled circuit [36].

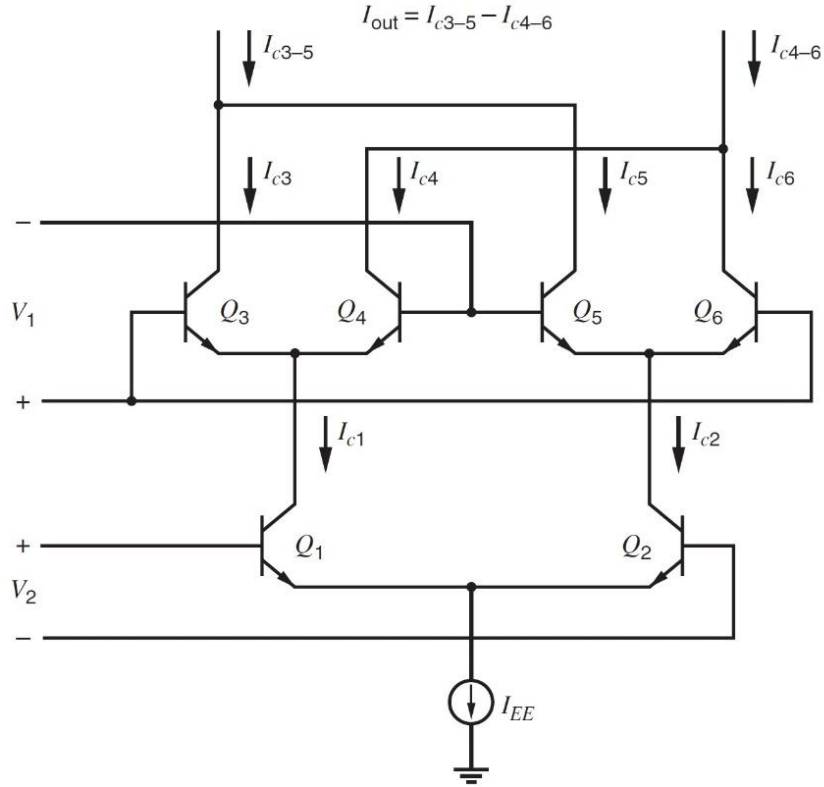


Figure 3.17: Gilbert cell (Four quadrant multiplier).

The Gilbert cell forms the basis of most integrated circuit balanced multiplier systems. A cross-coupled emitter pair is connected in series with one emitter coupled circuit. The formulas (3.8-11) for the currents that occur in the following circuit are shown.

$$I_{C3} = \frac{I_{C1}}{1 + \exp(-V_1/V_T)} \quad (3.8)$$

$$I_{C4} = \frac{I_{C1}}{1 + \exp(V_1/V_T)} \quad (3.9)$$

$$I_{C5} = \frac{I_{C2}}{1 + \exp(V_1/V_T)} \quad (3.10)$$

$$I_{C6} = \frac{I_{C2}}{1 + \exp(-V_1/V_T)} \quad (3.11)$$

The I_{C1} and I_{C2} currents relate to V_2 . The formulas (3.12) and (3.13) are as follows.

$$I_{C1} = \frac{I_{EE}}{1 + \exp(-V_2/V_T)} \quad (3.12)$$

$$I_{C2} = \frac{I_{EE}}{1 + \exp(V_2/V_T)} \quad (3.13)$$

If formulas for I_{C3} , I_{C4} , I_{C5} and I_{C6} are rearranged, the formulas (3.14-17) are as follows.

$$I_{C3} = \frac{I_{EE}}{\left[1 + \exp\left(-\frac{V_1}{V_T}\right)\right] \left[1 + \exp\left(-\frac{V_2}{V_T}\right)\right]} \quad (3.14)$$

$$I_{C4} = \frac{I_{EE}}{\left[1 + \exp\left(\frac{V_1}{V_T}\right)\right] \left[1 + \exp\left(-\frac{V_2}{V_T}\right)\right]} \quad (3.15)$$

$$I_{C5} = \frac{I_{EE}}{\left[1 + \exp\left(\frac{V_1}{V_T}\right)\right] \left[1 + \exp\left(\frac{V_2}{V_T}\right)\right]} \quad (3.16)$$

$$I_{C6} = \frac{I_{EE}}{\left[1 + \exp\left(-\frac{V_1}{V_T}\right)\right] \left[1 + \exp\left(\frac{V_2}{V_T}\right)\right]} \quad (3.17)$$

Current changes are as follows (3.18).

$$\begin{aligned} \Delta I &= I_{C3-5} - I_{C4-6} = I_{C3} + I_{C5} - (I_{C4} + I_{C6}) = (I_{C3} + I_{C6}) - (I_{C4} + I_{C5}) \\ &= I_{EE} \tanh(V_1/2V_T) \tanh(V_2/2V_T) \end{aligned} \quad (3.18)$$

As understood from the above formula (3.18), dc transfer characteristic is the result of the hyperbolic tangent product of the V_1 and V_2 voltages. According to the state of these two voltages, Gilbert cell works in three different ways.

- $V_1 < V_T$ and $V_2 < V_T$ In this case, the circuit operates as a multiplier and the formula (3.19) is the following.

$$\tanh(V_{1,2}/2V_T) \cong V_{1,2}/2V_T \quad (3.19)$$

- If one of the inputs is too large according to V_T , it will act as a modulator by multiplying the small signal applied by a square wave.
- If both of the inputs are greater than V_T , all transistors work as switches that do not enter saturation. This is useful for determining the phase differences of two signals with limited amplitude values. Since the circuit operates in this way, it can sometimes be called a phase detector. This mode of operation is required for phase-locked loops.

Considering the first case, that is, when $V_1 < V_T$ and $V_2 < V_T$ statuses occur:

$$\tanh(x) = x + x^3/3 + \dots \cong x$$

As mentioned earlier, when there are low amplitude signals, the circuit performs a multiplication. Generally, however, the amplitudes of the input signals are much larger than that of V_T . Therefore, it is necessary to make up for the hyperbolic tangent transfer characteristics of the basic circuit. This leads to a non-linear application that causes pre-distortion in the input signals.

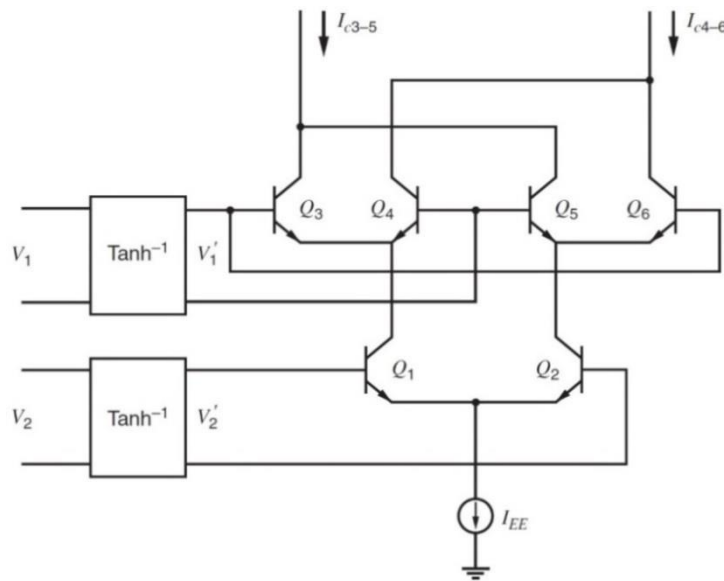


Figure 3.18: Gilbert multiplier with predistortion circuits.

hyperbolic tangent character as shown above in Figure 3.18 [36].

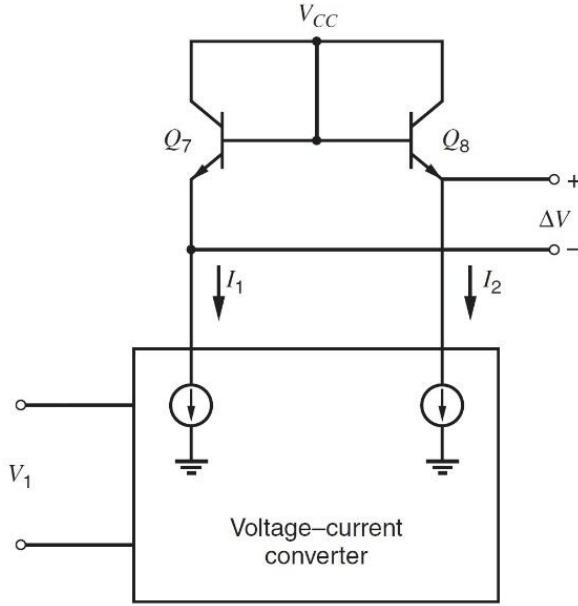


Figure 3.19: Inverse hyperbolic tangent circuit.

The above circuit (Figure 3.19) is a detailed illustration of the predistortion section shown as Tanh^{-1} box.

It was assumed in the study that this circuit produces a differential output current that is linearly related with the V_1 input. The following are the formulas (3.20) for currents in relation to V_1 .

$$I_1 = I_{o1} + K_1 V_1 \text{ and } I_2 = I_{o1} - K_1 V_1 \quad (3.20)$$

The current I_{01} is the dc current flowing through each output line when V_1 is zero. K_1 is the transconductivity of the voltage current converter. The differential voltage formula (3.21) resulting from the above two diode-connected transistors is given below.

$$\Delta V = V_T \ln \left(\frac{I_{o1} + K_1 V_1}{I_s} \right) - V_T \ln \left(\frac{I_{o1} - K_1 V_1}{I_s} \right) = V_T \ln \left(\frac{I_{o1} + K_1 V_1}{I_{o1} - K_1 V_1} \right) \quad (3.21)$$

$$\tanh^{-1} = \ln\left(\frac{1+x}{1-x}\right)/2 \text{ from as,}$$

The above formula can be written as formula (3.22).

$$\Delta V = 2V_T \tanh^{-1} \left(\frac{K_1 V_1}{I_{o1}} \right) \quad (3.22)$$

As a result, the following formula (3.23) is obtained.

$$\Delta I = I_{EE} \left(\frac{K_1 V_1}{I_{o1}} \right) \left(\frac{K_2 V_2}{I_{o2}} \right) \quad (3.23)$$

In the following section, the full circuit of an analog multiplier is given in Figure 3.20 [36].

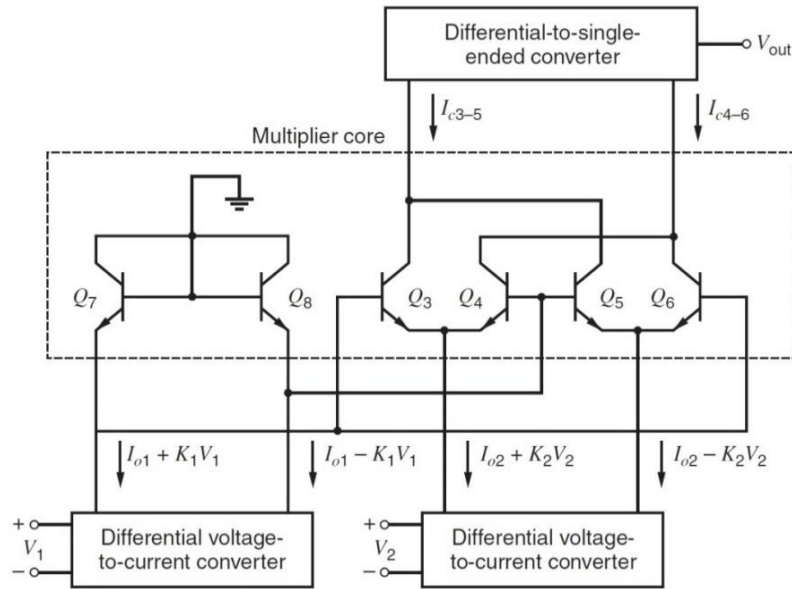


Figure 3.20: Complete four quadrant analog multiplier.

The output voltage formula (3.24) of the above full multiplier circuit is given below.

$$V_{out} = I_{EE} K_3 \left(\frac{K_1}{I_{o1}} \right) \left(\frac{K_2}{I_{o2}} \right) V_1 V_2 = 0.1 V_1 V_2 \quad (3.24)$$

In communication systems, it is often necessary to multiply a continuously changing signal by a square wave signal. In the case of such a need, this multiplication can be performed by multiplier circuit directly by applying a

sufficiently large signal directly to the cross-coupled pair circuit. In order to be an example of this situation, the simulation results are given below.

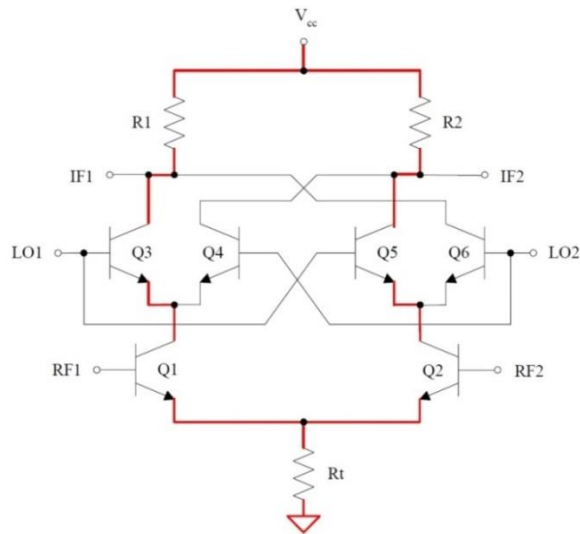


Figure 3.21: Gilbert cell Q3-Q5 turn on.

In Figure 3.21 V_{L01} is large enough to activate $Q3$ and $Q5$, and v_{L02} is small enough to keep $Q4$ and $Q6$ off. And the circuit works as in Figure 3.21, so the current flows through the red areas. $Q1$ transistor is connected to $R1$ and $Q2$ is connected to $R2$.

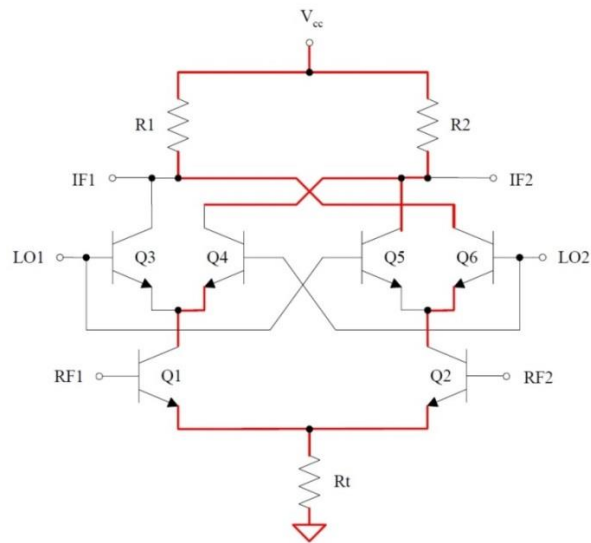


Figure 3.22: Gilbert cell Q4-Q6 turn on.

In Figure 3.22 V_{L02} is large enough to activate Q_4 and Q_6 , and V_{L01} is small enough to keep Q_3 and Q_5 off. And the circuit works as in Figure 3.22, so the current flows through the red areas. Q_1 transistor is connected to R_2 and Q_2 is connected to R_1 .

Simulation: Gilbert cell mixer

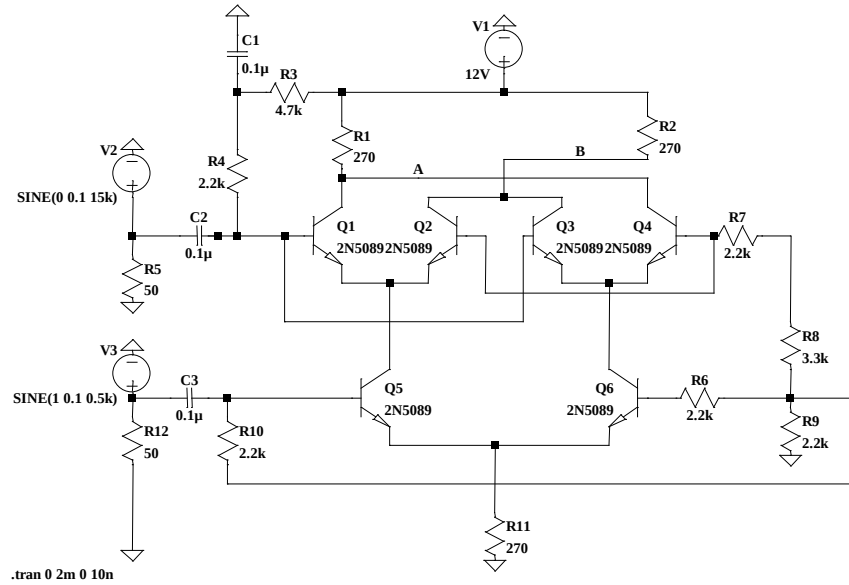


Figure 3.23: Gilbert cell mixer circuit schematic.

Figure 3.23 is the schematic created for Gilbert cell mixer simulation. Netlist file was created in accordance with this schematic. The frequency of the input signal indicated by V_2 is higher than that of V_3 . V_2 was designated as local oscillator and V_3 was treated as radio frequency. After Netlist was created, simulation results were obtained by running NGSPICE and OCTAVE programs jointly. The points to be considered in the simulation output are the points A and B.

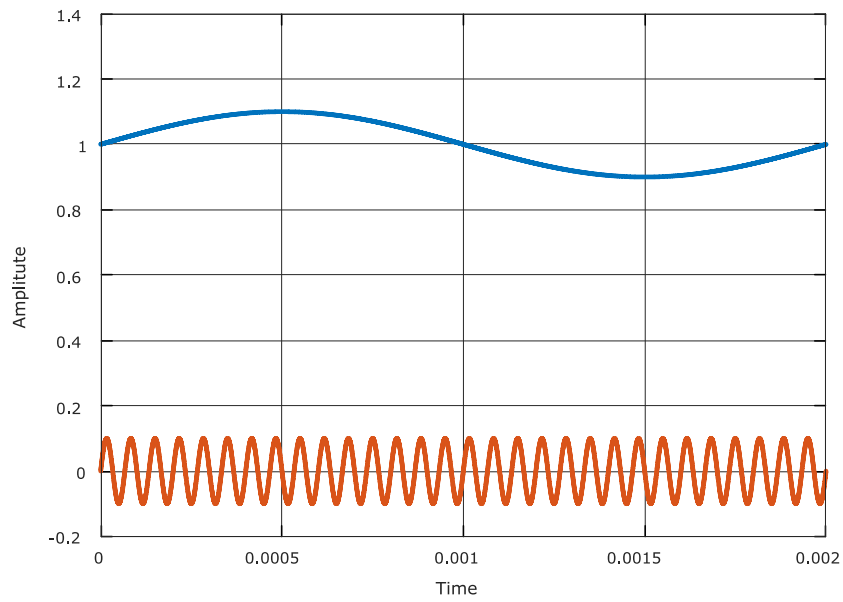


Figure 3.24: Gilbert cell mixer input waveforms.

In the schematic of the Gilbert mixer circuit, the transistor model 2N5089 was used. Figure 3.24 shows the input signals applied to the Gilbert mixer circuit. The frequency of the sine signal that appears as blue in RF (Radio Frequency), shown in Figure 3.24, is 0.5 kHz. The frequency of the other sine signal, which is displayed as LO (Local Oscillator), which appears red in Figure 3.24, is 15 kHz.

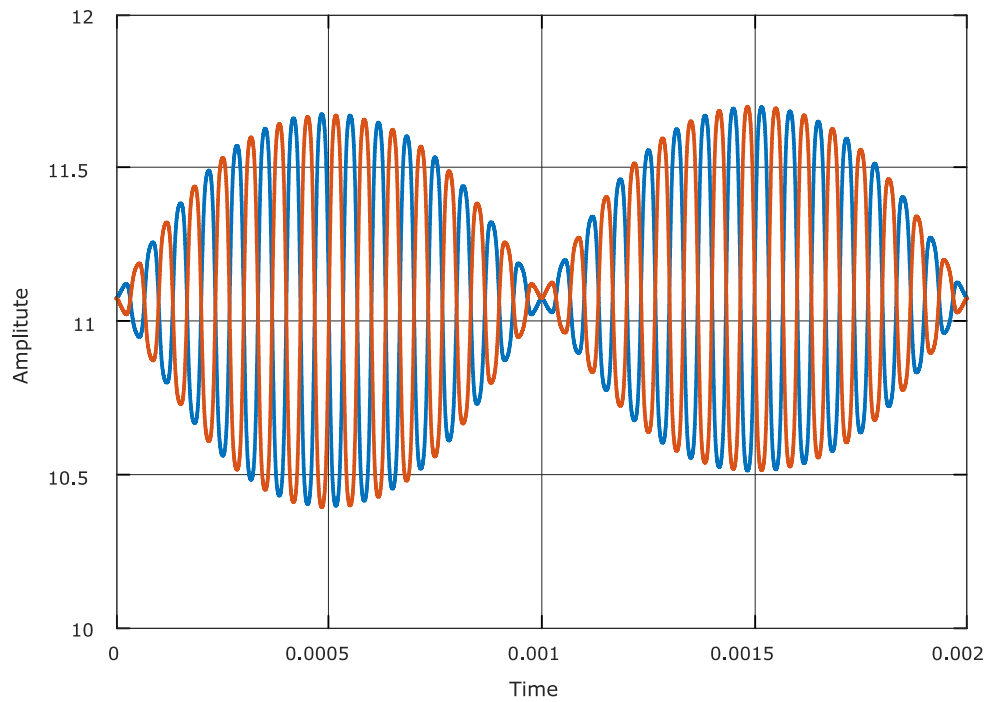


Figure 3.25: Gilbert cell mixer output waveforms.

As shown in Figure 3.23, the drain legs of the 4 transistors at the top are connected to each other to form a group of 2 and are named A and B. When the Gilbert cell Mixer circuit is simulated, two drain points A and B are selected as the measuring points. Figure 3.25 shows the simulation result at these two common ports. The red waveform in Figure 3.25 is the output of the A point, while the blue waveform in Figure 3.25 is the output of the B point. As a result, the correct mixing process was required in accordance with the two input signals and the result was determined in the simulation.

3.4. Transmission Gate Chain

Propagation delay in transmission gate chains (Figure 3.26) used in many digital applications is commensurate to the square of the number of gates. Within the scope of this study, delay optimization was achieved by adding buffer at certain frequency between the transmission gates, and how the propagation delay of a chain consisting of 16 transmission gates changed depending on the random parameters was analyzed. Delay values in buffered and unbuffered chain circuits were calculated in nominal parameters before analysis.

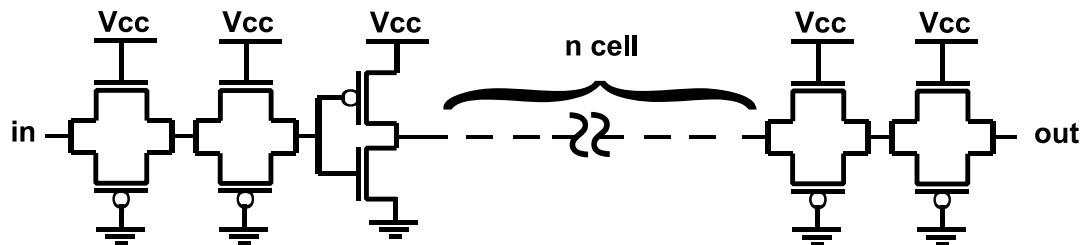


Figure 3.26: Transmission gate chain diagram.

Figure 3.27 shows the simulation outputs at the all output nodes of a transmission gate chain designed with and without buffer.

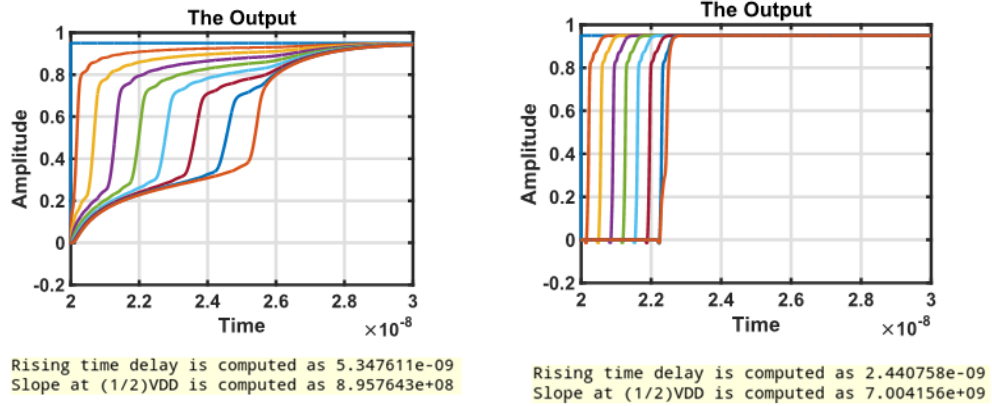


Figure 3.27: Transmission gate chain simulations without buffer and with buffer.

In the second figure in Figure 3.27, the simulation result of a transmission gate chain designed using buffer is seen. In addition, under the pictures there is a delay result for the last output signal at the end of the chain. In addition, there is also information on the slope of the signals. More detailed information on the simulation will be given later.

Simulation1: Achieving the resistance value of a single transmission gate

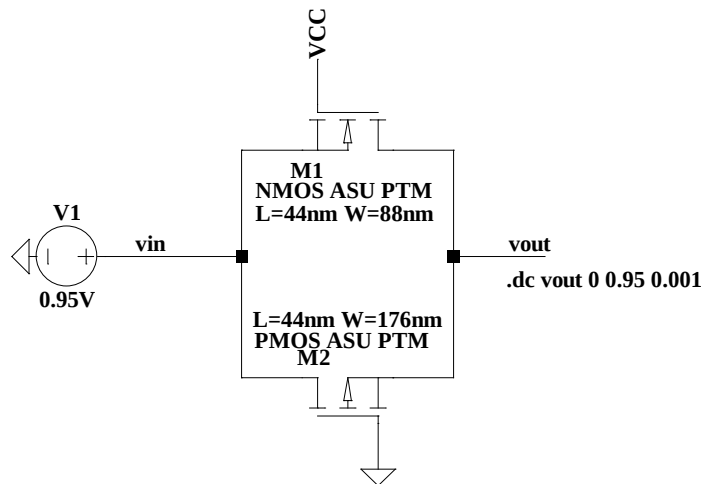


Figure 3.28: Single transmission gate circuit schematic.

Figure 3.28 indicates the schematic of the single transmission gate circuit. The circuit consists of a NMOS and a PMOS. At the end of the netlist of the Figure 3.28, the difference between the input voltage and the output voltage is divided by the

current flowing through the input. Here, however, it is desired to have the resistance of a transmission gate.



Figure 3.29: Resistance plot of a single transmission gate.

Figure 3.29 indicates the resistance change graph of a transmission gate on NGSPICE. The difference between the output voltage and the input voltage is divided by the current drawn from the input. During the simulation, the output voltage was increased from 0 to 0.95 V in 0.001V increments and the graph in Figure 3.29 was obtained. The average resistance value of a single transmission gate is approximately 30.5k.

Simulation 2: Finding the node capacitance between two transmission gates

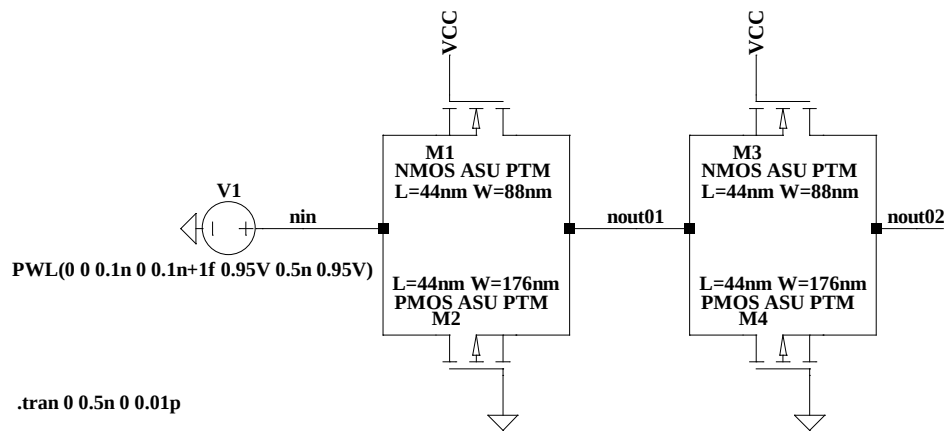


Figure 3.30: Two transmission gates schematic for measuring capacitance.

Figure 3.30 indicates the schematic of a circuit consisting of two transmission gates. The purpose of establishing this circuit is to obtain the node capacitance of a transmission gate. Two calculations are made in the netlist of the circuit. The first calculation finds the τ (Tau) time value. The τ value is the time at which the voltage value at the first transmission gate output is 0.63 times the input voltage. In the second calculation, the τ value is divided by the resistance of a transmission gate and as a result a transmission gate node capacitance is calculated.

```
ngspice 1 -> tg_cnode.cir
Circuit: * determine node capacitance of a transmission gate
Doing analysis at TEMP = 27.000000 and TNOM = 27.000000
Warning: This model supports BSIM4.2.1, 4.3.0 and 4.4.0; you specified a wrong version number.
Warning: (B1 + Weff) may be too small.
Warning: toxe, toxp and dtox all given and toxe != toxp + dtox; dtox ignored.
Warning: This model supports BSIM4.2.1, 4.3.0 and 4.4.0; you specified a wrong version number.
Warning: This model supports BSIM4.2.1, 4.3.0 and 4.4.0; you specified a wrong version number.
Warning: (B1 + Weff) may be too small.
Warning: toxe, toxp and dtox all given and toxe != toxp + dtox; dtox ignored.
Warning: This model supports BSIM4.2.1, 4.3.0 and 4.4.0; you specified a wrong version number.
Warning: vin: no DC value, transient time 0 value used

Initial Transient Solution
-----
Node          Voltage
----          -
nvdd          0.95
nvss          0
nin           0
nout01        2.53483e-06
nout02        3.37974e-06
vin#branch    3.42798e-10
vss#branch    3.95024e-14
vdd#branch    -3.42837e-10

Reference value : 4.85311e-10
No. of Data Rows : 50018

Measurements for Transient Analysis
tau           = 2.389154e-11 targ= 1.238915e-10 trig= 1.000000e-10
cnode         = 7.83329e-16
```

Figure 3.31: τ and node capacitance result for single transmission gate.

Figure 3.31 shows the calculated τ and node capacitance values. The output voltage of the first transmission gate has a time to reach $0.63V_{dd}$. This time to reach $0.63V_{dd}$ is τ value. Its value is 2.39×10^{-11} second. The Node capacitance was found by dividing the τ value by the previously calculated transmission gate resistance value. Node capacitance is 7.83×10^{-16} Farad.

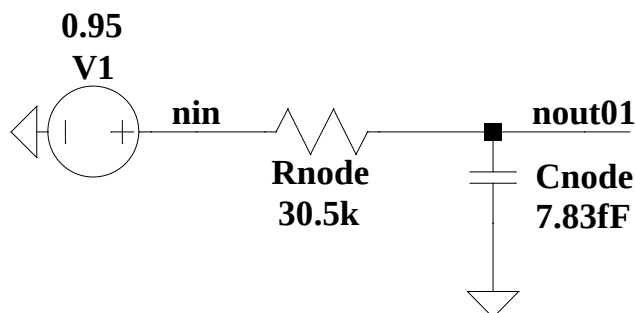


Figure 3.32: Node capacitance and node resistance of single transmission gate

Figure 3.32 shows the simple expression of a transmission gate in terms of resistance and capacitance. As a result of the calculations made, the resistance of a gate was 30.5k and its capacitance was 7.83fF.

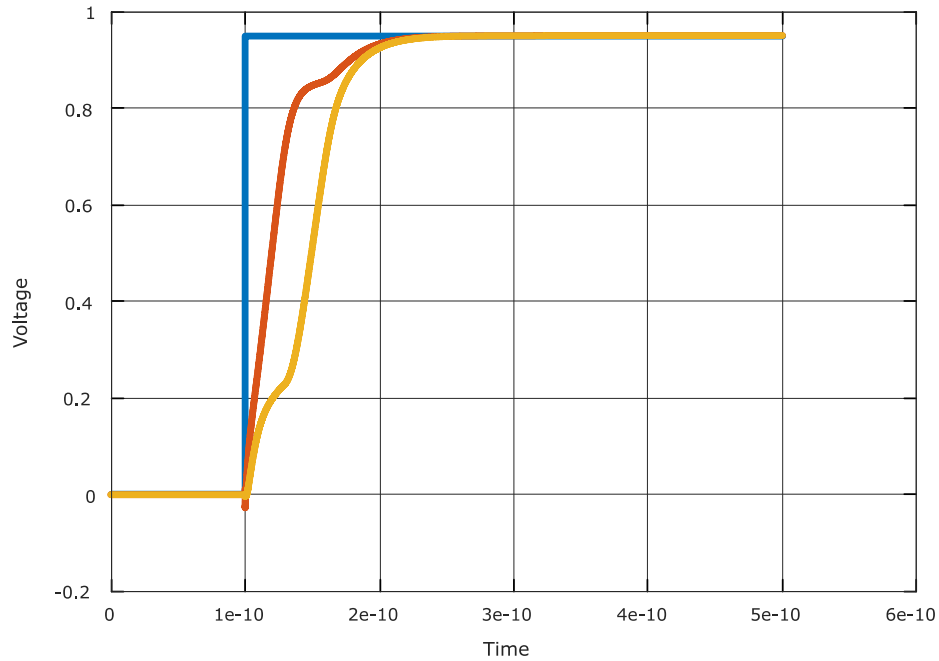


Figure 3.33: The two transmission gates simulation plot.

Figure 3.33 is the plot of two transmission gate circuits simulation. The blue signal is the input voltage applied to the circuit. The red signal is the output of the first transmission gate. The yellow signal refers to the output of the second transmission gate. Naturally, the final output of the circuit reached 0.95V later.

“Simulation 3: Buffer latency value and optimum buffer count for a transmission gate chain.”

In this simulation, the buffer gate was studied. The delay in the buffer gate was found. In the previous simulation, transmission gate capacitance and resistance values were found. These values will be used in this simulation to obtain the buffer number.

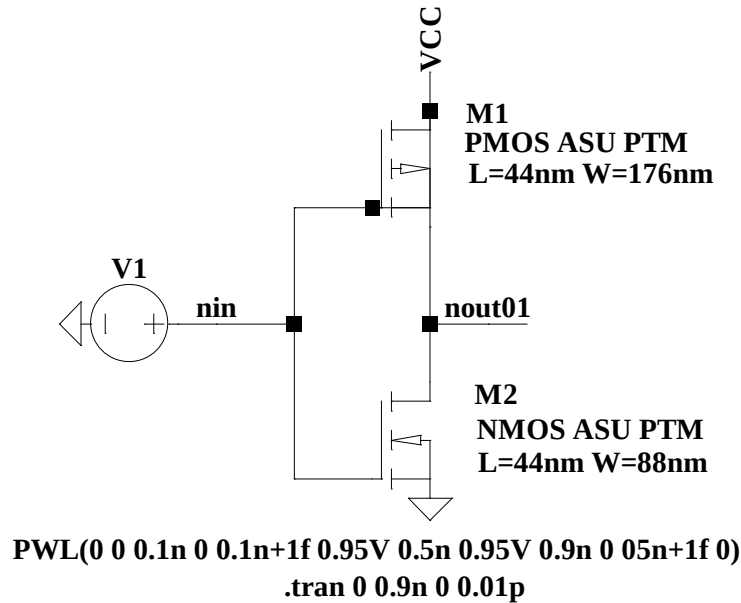


Figure 3.34: Buffer gate circuit schematic.

Figure 3.34 indicates the schematic of a buffer gate. It is formed by combining the gate ports of an NMOS and a PMOS. It has the same function as the inverter. The purpose of this simulation is to find the delay difference between the input signal and the output signal. For this purpose, when creating the netlist of the schematic in Figure 3.34, the commands for the delay account are added and then simulated.

Figure 3.35 shows the rising and falling delays of the generated buffer circuit. The rising delay is 1.14×10^{-11} seconds. The falling delay is 1.22×10^{-11} seconds.

```

ngspice 1 -> buffer.cir
Circuit:
Doing analysis at TEMP = 27.000000 and TNOM = 27.000000
Warning: This model supports BSIM4.2.1, 4.3.0 and 4.4.0; you specified a wrong version number.
Warning: (B1 + Weff) may be too small.
Warning: tox, toxp and dtox all given and tox != toxp + dtox; dtox ignored.
Warning: This model supports BSIM4.2.1, 4.3.0 and 4.4.0; you specified a wrong version number.
Warning: vin: no DC value, transient time 0 value used

Initial Transient Solution
-----
Node                Voltage
----                -
vout1                0.949999
vin1                 0
nvdd                 0.95
nvss                 0
vin#branch           2.06294e-10
vss#branch           0
vdd#branch           -2.0731e-10

Reference value : 8.80681e-10
No. of Data Rows : 90028

Measurements for Transient Analysis
tdr_single          = 1.141043e-11 targ= 5.114109e-10 trig= 5.000005e-10
tdf_single          = 1.219308e-11 targ= 1.121936e-10 trig= 1.000005e-10

```

Figure 3.35: Buffer delay result for rising and falling edge.

In Figure 3.36, delays in the rising and falling edge of a buffer were shown. The resulting delays can be used in the following formulas (3.25) to calculate $t_{p_{buff}}$, with the result that m can be found.

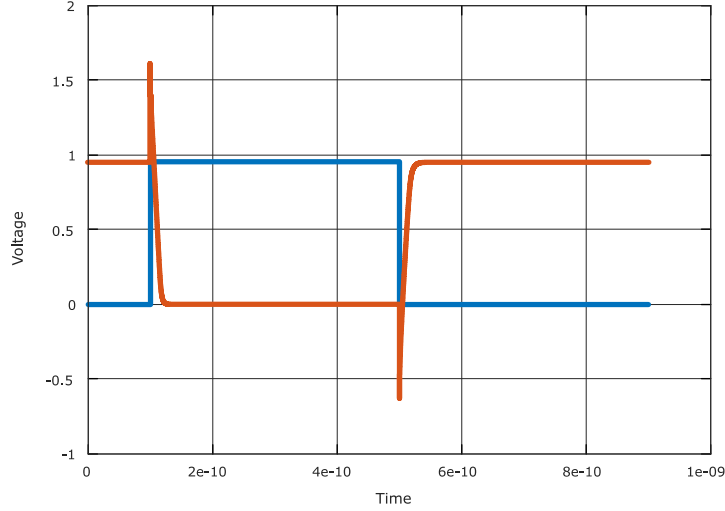


Figure 3.36: Buffer gate simulation plot.

After how many transmission gates, a buffer will be added chain, this is the name of the m information. In order to be used in this formula, there is a need for a transmission gate's capacitance and resistance value. These values were determined in Figure 3.31, previous simulation.

$$M_{opt} = 1.7 \sqrt{t_{p_{buf}} / C \times R_{eq}} \quad (3.25)$$

Calculation is done as follows:

$$C = 0.78 \text{ fF}$$

$$R_{eq} = 30.5 \text{ k}\Omega$$

$$t_{p_{buf}} = (t_{dr_buff} + t_{df_buff}) / 2 = ((1.14 + 1.21) / 2) 10^{-11}$$

From this, the optimum number of m is found as 1.2. So from here, a buffer should be added after each 1.2 gate. Since, the number 1.2 is not an integer, by adding a buffer after each transmission gate, all the transmission gate chain is simulated.

Simulation 4: Transmission gate chain without buffer

Figure 3.37 shows the schematic of a transmission gate single and chain circuit that is prepared without buffer.

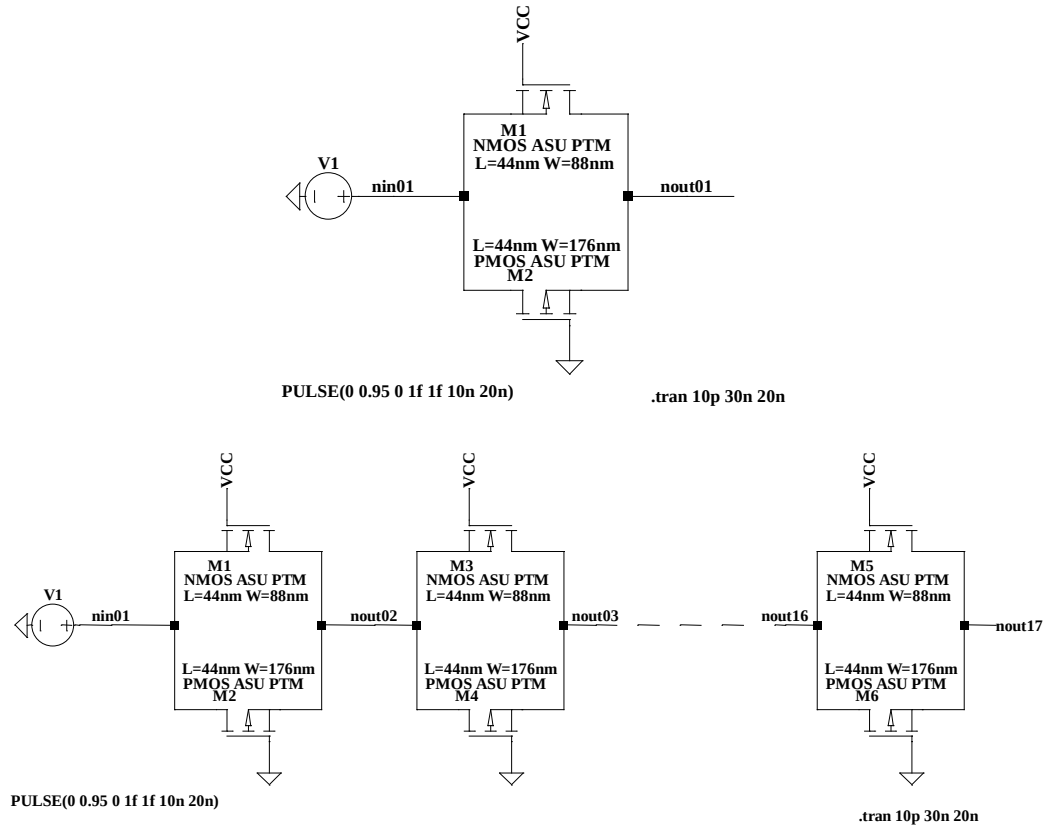


Figure 3.37: Transmission gate single and chain schematic without buffer.

These two schematics were collected in one netlist file for simulation. The required measure commands have been added to the netlist file for single gate and 16 gates delay. The difference between the time the input voltage reaches 0.475V (0.95/2) and the output voltage to 0.475V (0.95/2) gives the delay information.

```

Initial Transient Solution
-----
Node                               Voltage
----                               -
nin01                               0
nvss                                0
nout01                             8.4492e-07
nvdd                                0.95
nout02                             2.61668e-05
nout03                             5.06458e-05
nout04                             7.3437e-05
nout05                             9.45403e-05
nout06                             0.000113956
nout07                             0.000131683
nout08                             0.000147722
nout09                             0.000162073
nout10                             0.000174735
nout11                             0.00018571
nout12                             0.000194996
nout13                             0.000202594
nout14                             0.000208503
nout15                             0.000212724
nout16                             0.000215256
nout17                             0.0002161
vin#branch                          2.91092e-09
vss#branch                          3.40383e-13
vdd#branch                          -2.91127e-09

Reference value : 2.66705e-08

No. of Data Rows : 1021

Measurements for Transient Analysis

tdr_single      = 8.603222e-12 targ= 2.000860e-08 trig= 2.000000e-08
tdr_multi       = 4.256135e-09 targ= 2.425614e-08 trig= 2.000000e-08

```

Figure 3.38: Delay measurement of the transmission gate chain without buffer.

Figure 3.38 shows the rising delay on a single gate with the name tdr_single. In addition, the total rising delay in the chain is also called tdr_multi. The single gate delay is 8.6×10^{-12} second. The total latency of the chain is 4.3×10^{-9} second.

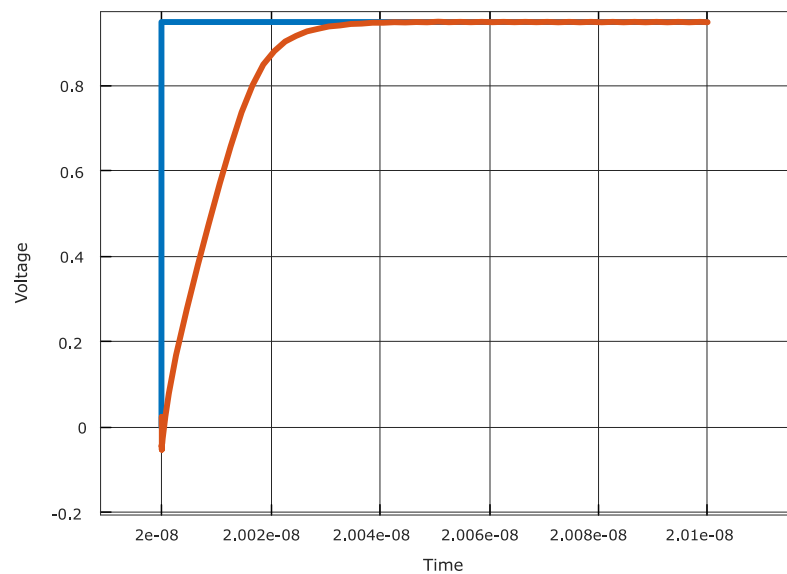


Figure 3.39: The single transmission gate delay on the chain without buffer.

Figure 3.39 indicates the attitude of a single transmission gate. The blue signal is the input and the red signal is the output voltage graph of the single transmission gate.

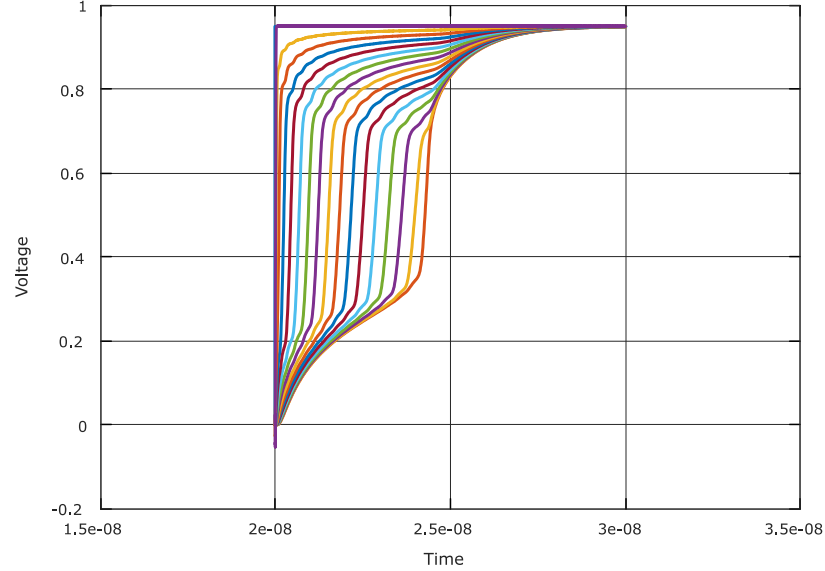


Figure 3.40: The simulation plot of the transmission gate chain without buffer.

Figure 3.40 gives the simulation output of the transmission Gate chain, which is prepared without buffer. Here the outputs of all the gates in the chain are seen.

As a result of this simulation, the total delay amount in the transmission gate chain prepared without using the buffer is 4.26×10^{-9} second.

Simulation 5: Transmission gate chain with buffer considering m value

The transmission gate chain circuit generated using the buffer is examined. The amount of buffer was previously studied.

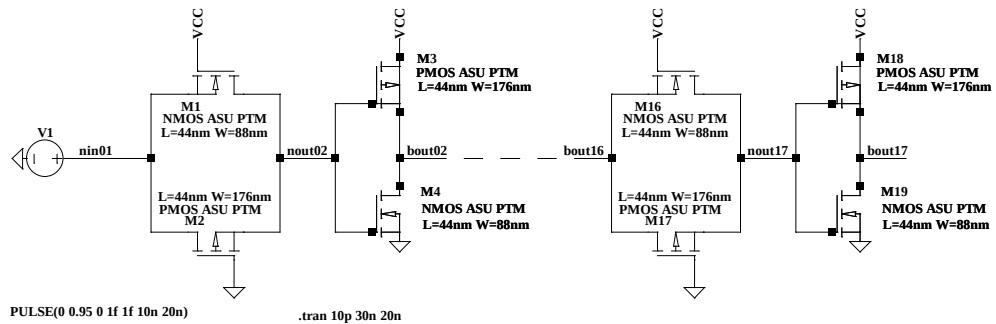


Figure 3.41: The schematic of the transmission gate chain circuit with buffer.

Figure 3.41 shows the schematic of the transmission gate chain circuit created using the buffer. One buffer was added at the end of each gate. And the necessary delay measurement commands have been added on the netlist of this schematic.

```
Initial Transient Solution
-----
Node          Voltage
-----
nin01         0
nvss          0
nout02        2.87889e-06
nvdd          0.95
bout02        0.949994
nout03        0.94999
bout03        4.56966e-06
nout04        7.44851e-06
bout04        0.949994
nout05        0.94999
bout05        4.56966e-06
nout06        7.44851e-06
bout06        0.949994
nout07        0.94999
bout07        4.56966e-06
nout08        7.44851e-06
bout08        0.949994
nout09        0.94999
bout09        4.56966e-06
nout10        7.44851e-06
bout10        0.949994
nout11        0.94999
bout11        4.56966e-06
nout12        7.44851e-06
bout12        0.949994
nout13        0.94999
bout13        4.56966e-06
nout14        7.44851e-06
bout14        0.949994
nout15        0.94999
bout15        4.56966e-06
nout16        7.44851e-06
bout16        0.949994
nout17        0.94999
bout17        8.4575e-07
vin#branch    3.77693e-10
vss#branch    1.66574e-09
vdd#branch    -6.05962e-09

Reference value : 2.82105e-08
No. of Data Rows : 1021

Measurements for Transient Analysis
tdr_multi      = 1.751200e-09 targ= 2.175120e-08 trig= 2.000000e-08
```

Figure 3.42: Delay result on the transmission gate chain with buffer.

Figure 3.42 shows the delay in the chain with the name tdr_multi. The total delay in the chain using Buffer is 1.75×10^{-9} seconds.

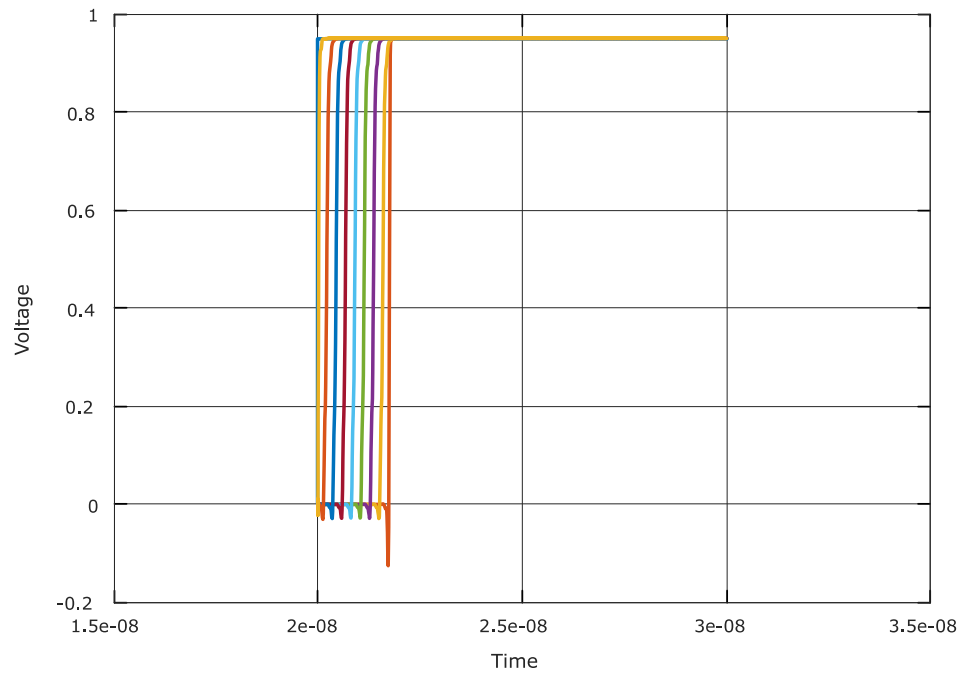


Figure 3.43: Simulation output of the transmission gate chain with buffer.

Figure 3.43 shows the simulation output of the transmission gate chain generated using the buffer. The circuit generated using the buffer from both graphical and numerical calculation results in a faster response. As a result, when the buffer is used, the total delay in the transmission gate chain is $1.75e^{-9}$. In the simulation made on the transmission gate chain circuit without using the buffer, the total delay was $4.26e^{-9}$. From this, it was concluded that the total delay when using the buffer was approximately 2.43 times faster than the delay in non-buffered circuit.

3.5. Inverter Chain

Inverter is a real building block of digital designs. If the inverter's structure, features and operation are understood, the design of various circuits is also easier. These circuits may be more complex circuits, such as multipliers, adders, NAND gates or microprocessors. The electrical behavior of more complex structures such as these complex gates can be solved by understanding the electrical behavior of the inverter [39].

In this inverter chapter, the dynamic CMOS inverter is analyzed with respect to the performance, robustness and integrity applying various simulations.

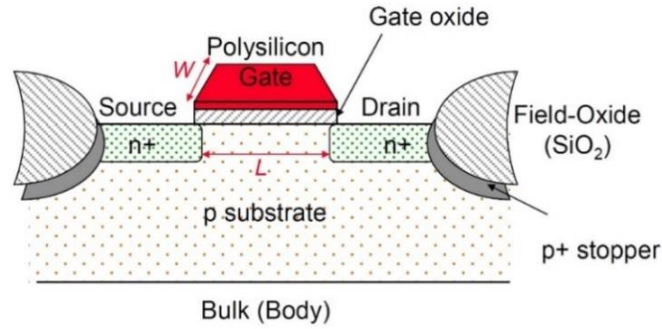


Figure 3.44: The interior scheme of a MOSFET.

The interior of a MOSFET is given in Figure 3.44. A MOSFET is the building block of the inverter and many other circuits.

- **Inverter Cell Schematic:** The Figure 3.45 shows the scheme of a CMOS inverter circuit. The inverter circuit is an infinite-resistant switching circuit.

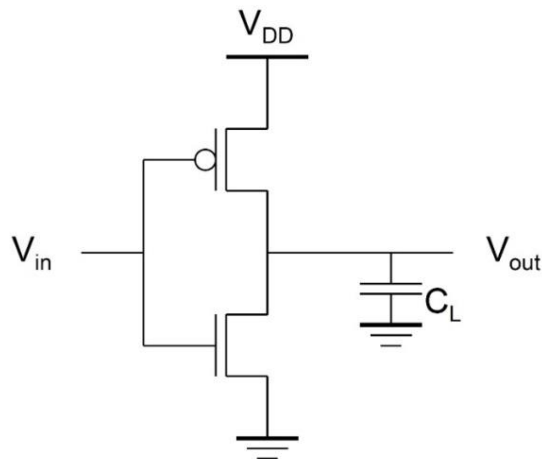


Figure 3.45: An inverter circuit diagram.

The Figure 3.45 below shows a different representation of the inverter circuit, which is an infinite-resistant switching circuit. In a transistor, V_{gs} has a finite resistance when $V_{gs} > V_t$, and when $V_{gs} < V_t$, it has infinite resistance. Thus, when the V_{in} value is high and V_{dd} , NMOS starts to operate and has a finite resistance and terminates the PMOS operation and has infinite resistance. If the value of V_{in} is low and 0, PMOS will have a finite resistance starting to work, and NMOS will have an infinite resistance to finish its operation. The Figure 3.46 summarizes this situation [37].

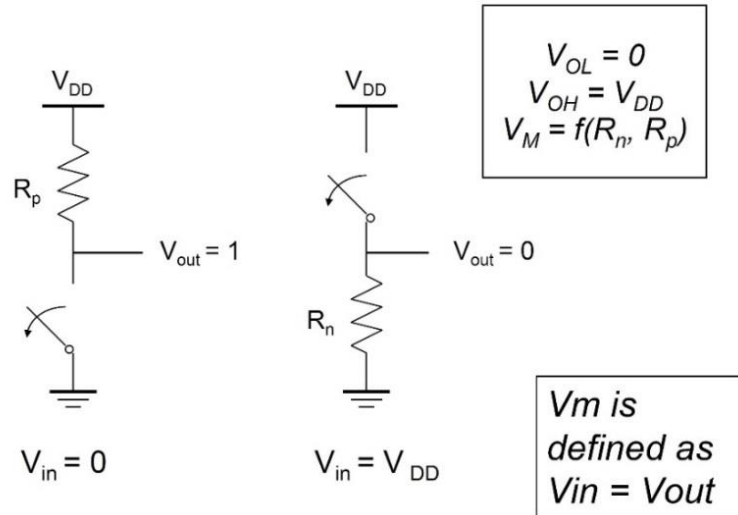


Figure 3.46: Switch models of CMOS inverter.

The voltage transfer characteristic graph of an inverter circuit is in Figure 3.47 below. That is, it reflects the voltage change value (V_{out}) at the output according to the input voltage value (V_{in}) change of an inverter circuit. As can be seen from the V_{tc} operation takes place in a narrow region. It causes a large change in the output, when there is a minor change in input voltage.

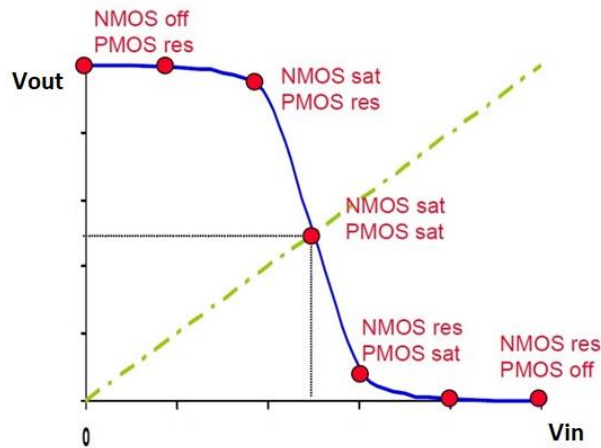


Figure 3.47: Voltage transfer characteristic of static CMOS inverter.

The graph also shows which regions (resistive, off, saturation) the NMOS and PMOS during the inverter operation. We will use 0.95 V as the supply voltage (V_{DD}) in the simulations we applied for the inverter. The reason of our 0.95 volts is that we

are using the ASU PTM 22nm MOSFET model. For this reason, V_{in} and V_{out} values will be between 0-0.95 V in the chart Figure 3.47.

Figure 3.48 illustrates the operation of an inverter and the movement of the current during operation. When the input of the circuit reaches from high to low, the PMOS operates and the C_L capacitance is recharged through R_p . Finally V_{out} reaches V_{DD} level. When the circuit is set from low to high, NMOS operates and the C_L capacitance is discharged over R_n . At the end, V_{out} drops to low.

The response time of the inverter will vary with the size of the C_L . A fast gate is made by keeping the output capacitance small or by reducing the resistance of the transistor or MOSFET. In the future, simulation results will show us this situation.

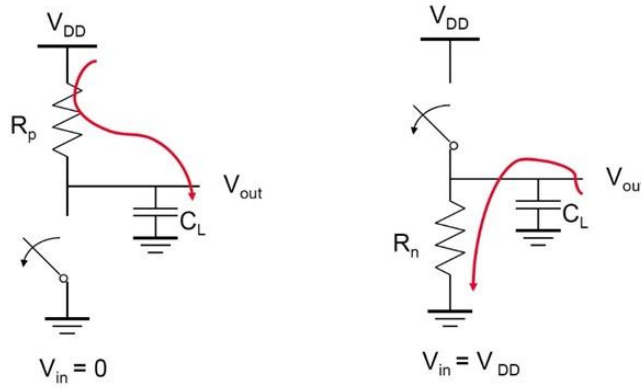


Figure 3.48: Switch model of dynamic behavior of static CMOS inverter.

It is important that the driving of the transistors is balanced during the design of the inverter circuit. Because the transistor drives balanced, noise margin maximizes and ensures the behavior of the circuit is symmetrical.

For the balanced and symmetrical driving of the PMOS and NMOS, the structural width of the PMOS element must be made larger than the NMOS. And this difference in width should be well established. By making these adjustments in the simulation stage, the results will be shown.

The V_m is called the switching threshold. The point at the moment when $V_i = V_{out}$ is the switching threshold. Using V_{tc} , V_m is the intersection point where V_{in} and V_{out} are equal. At this point, NMOS and PMOS are saturated region. V_m 's expression is as follows (3.26):

$$V_M \approx rV_{DD}/(1+r) \quad \text{where} \quad r = k_p V_{DSATp}/k_n V_{DSATn} \quad (3.26)$$

Switching threshold calibrate the ratio r (3.27), which crosscheck the relative driving resistance of the PMOS and NMOS transistors.

It is desirable for V_m to be $V_{dd}/2$. Therefore, r is desired to be close to 1.

$$\frac{(W/L)_p}{(W/L)_n} = \frac{k'_n V_{DSATn} (V_M - V_{Tn} - V_{DSATn}/2)}{k'_p V_{DSATp} (V_{DD} - V_M + V_{Tp} V_{DSATp}/2)} \quad (3.27)$$

It is possible that the post-production circuit parameters may move away from the nominal values when designing the door or circuit. Therefore, we should consider this situation. In this case, the behavioral variation of the inverter structure is shown in the graph below on the V_{TC} graph in Figure 3.49. Here we can see that the V_m (switching threshold) is shifted as a result of the parameters and elements moving away from the nominal value.

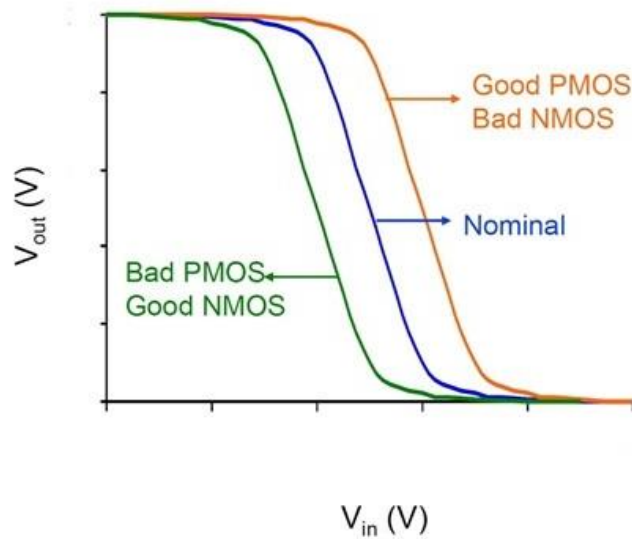


Figure 3.49: Impact of device variations on static CMOS inverter VTC.

As previously mentioned, in order for an inverter structure to exhibit symmetrical behavior, it was stated that the width ratios of NMOS and PMOS should be appropriate. Now, with the simulations we make, it is important to determine this width ratio correctly and accordingly to realize the future simulations correctly. Now, by simulating inverter structures with a different width ratio, it will be seen which width ratio provides symmetric behavior.

ASU PTM 22nm MOSFET model is used for NMOS and PMOS in simulations. According to this model, the supply voltage is 0.95 volts. In the first simulation below we observe the situation where the ratio of $(W/L)_p/(W/L)_n$ is 1.

Simulation 1: $(W/L)_p/(W/L)_n=1$

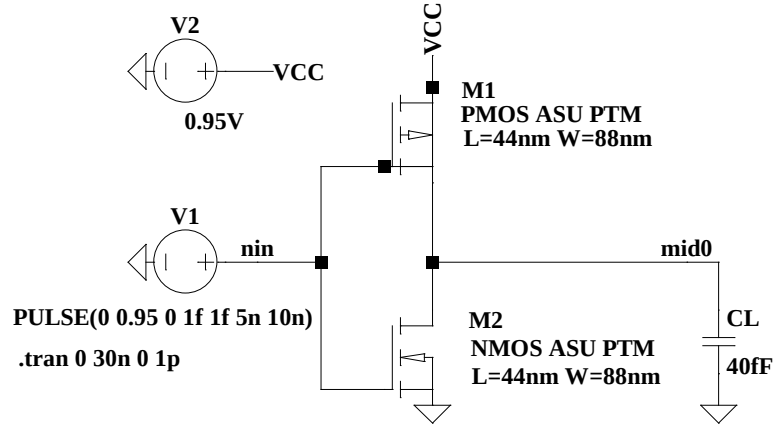


Figure 3.50: The schematic of single inverter with $W_p/W_n=1$.

Figure 3.50 shows the schematic of an inverter circuit where the width of the PMOS is equal to the width of the NMOS. The calculation commands required for the delay calculation are added to the schematic spice netlist. In this way, necessary conditions have been met to calculate the falling and rising delays.

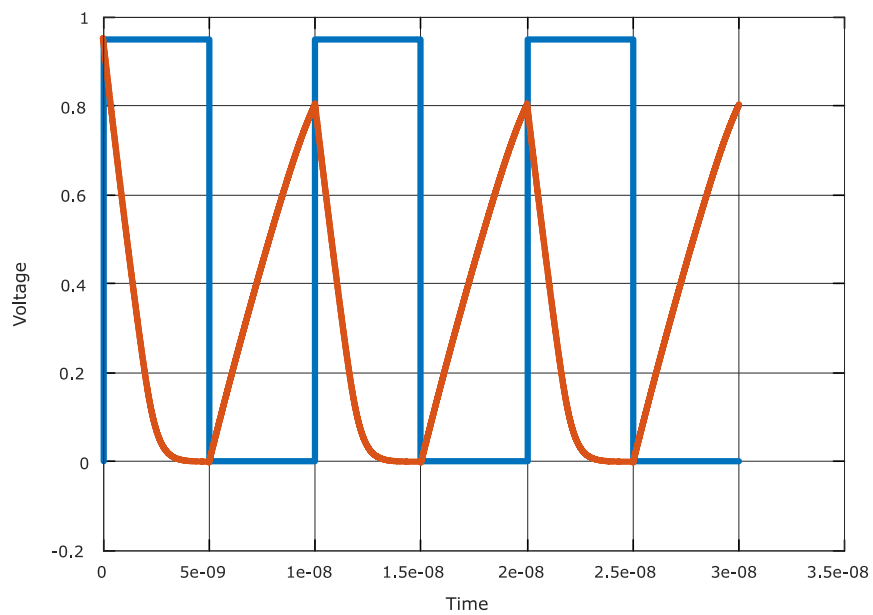


Figure 3.51: The plot of a single inverter with $W_p/W_n=1$.

Figure 3.51 shows the simulation output of an inverter circuit. The blue signal is the input signal applied to the inverter and the red signal is the voltage change graph on the inverter output.

```
ngspice 1 -> source inverter_single_ratio1.cir
Circuit: * testing propagation delay of an inverter
Doing analysis at TEMP = 27.000000 and TNOM = 27.000000
Warning: This model supports BSIM4.2.1, 4.3.0 and 4.4.0; you specified a wrong version number.
Warning: (B1 + Weff) may be too small.
Warning: tox, toxp and dtox all given and tox != toxp + dtox; dtox ignored.
Warning: This model supports BSIM4.2.1, 4.3.0 and 4.4.0; you specified a wrong version number.
Warning: (B1 + Weff) may be too small.
Warning: vin: no DC value, transient time 0 value used

Initial Transient Solution
-----
Node                Voltage
----                -
mid0                 0.949999
nin                  0
nvdd                 0.95
nvss                 0
vin#branch           9.69334e-11
vss#branch           0
vdd#branch           -9.7949e-11

Reference value : 2.79973e-08
No. of Data Rows : 30107

Measurements for Transient Analysis
tdr                  = 2.690954e-09 targ= 2.769096e-08 trig= 2.500000e-08
tdf                  = 8.429588e-10 targ= 2.084296e-08 trig= 2.000000e-08
```

Figure 3.52: The delay results of a single inverter with $W_p/W_n=1$.

Figure 3.52 shows the delay results of an inverter gate with a $W_p/W_n=1$ dimensioning ratio. The rise delay is named tdr and is 2.69×10^{-9} seconds. The fall delay is also called tdf and is 8.43×10^{-10} seconds.

Simulation 2: $(W/L)_p/(W/L)_n=2$

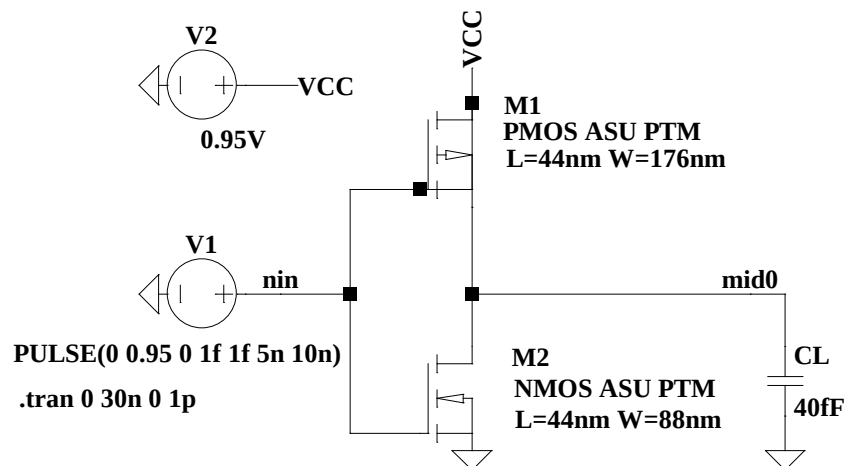


Figure 3.53: The schematic of single inverter with $W_p/W_n=2$.

Figure 3.53 shows the schematic of an inverter circuit in which the width of the PMOS is two times the width of the NMOS. Also the calculation commands required for the delay calculation are added to the schematic spice netlist. In this way, necessary conditions have been met to calculate the falling and rising delays.

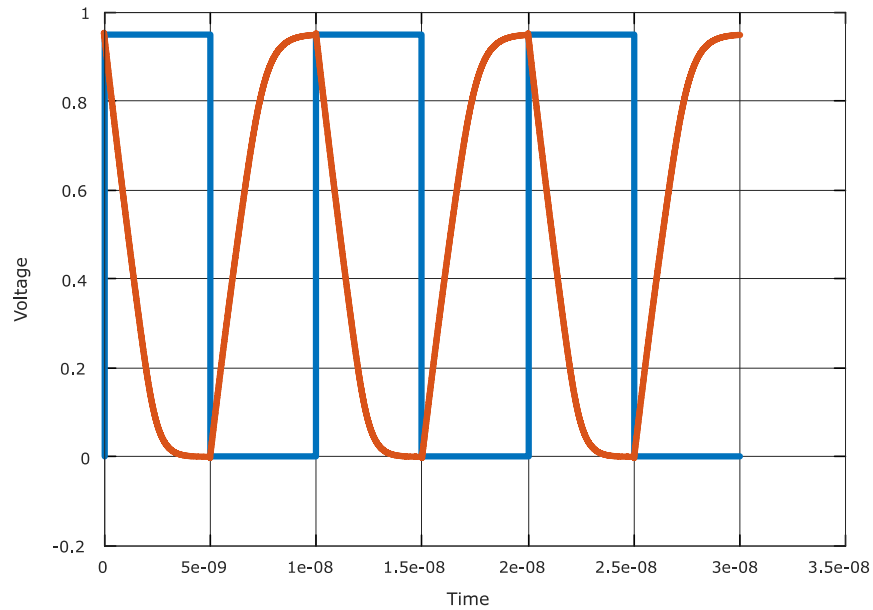


Figure 3.54: The plot of a single inverter with $W_p/W_n=2$.

Figure 3.54 shows the simulation output of an inverter circuit with $W_p/W_n=2$. The blue signal is the input signal applied to the inverter and the red signal is the voltage change graph on the inverter output.

```

ngspice 4 -> source inverter_single.cir
Circuit: * testing propagation delay of an inverter
Doing analysis at TEMP = 27.000000 and TNOM = 27.000000
Warning: This model supports BSIM4.2.1, 4.3.0 and 4.4.0; you specified a wrong version number
Warning: (B1 + Weff) may be too small.
Warning: tox, toxp and dtom all given and tox != toxp + dtom; dtom ignored.
Warning: This model supports BSIM4.2.1, 4.3.0 and 4.4.0; you specified a wrong version number
Warning: vin: no DC value, transient time 0 value used

Initial Transient Solution
-----
Node                                Voltage
----                                -
mid0                                0.949999
nin                                 0
nvdd                                0.95
nvss                                0
vin#branch                          2.06294e-10
vss#branch                          0
vdd#branch                          -2.0731e-10

Reference value : 2.67373e-08
No. of Data Rows : 30107

Measurements for Transient Analysis
tdr                                = 1.272273e-09 targ= 2.627227e-08 trig= 2.500000e-08
tdf                                = 1.200343e-09 targ= 2.120034e-08 trig= 2.000000e-08

```

Figure 3.55: The delay results of a single inverter with $W_p/W_n=2$.

Figure 3.55 shows the delay results of an inverter gate with a $W_p/W_n=2$ dimensioning ratio. The rise delay is named tdr and is 1.27×10^{-9} seconds. The fall delay is also called tdf and is 1.2×10^{-9} seconds.

The propagation delay is proportional to the time constant of the circuit consisting of resistance and capacity. In other words, the delay time is a function depending on the pull-up or pull-down resistance and the load capacity. In the following formulas the resistance value of the circuit is obtained by using this function.

$$tp_{HL} = \ln(2) Req_{nCL} = 0.69 Req_{nCL}$$

$$tp_{LH} = \ln(2) Req_{pCL} = 0.69 Req_{pCL}$$

$$tp = (tp_{HL} + tp_{LH})/2 = 0.69 CL (Req_n + Req_p)/2$$

$$1.272273 \times 10^{-9} = 0.69 \times 40 \times 10^{-15} (Req_p)$$

$$Req_p = 46.1 k\Omega$$

$$1.200343 \times 10^{-9} = 0.69 \times 40 \times 10^{-15} (Req_n)$$

$$Req_n = 43.5 k\Omega$$

$$(1.272273 \times 10^{-9} + 1.200343 \times 10^{-9})/2 = 0.69 \times 40 \times 10^{-15} (Req)/2$$

$$Req = 89.6 k\Omega$$

Simulation 3: $(W/L)_p/(W/L)_n=3$

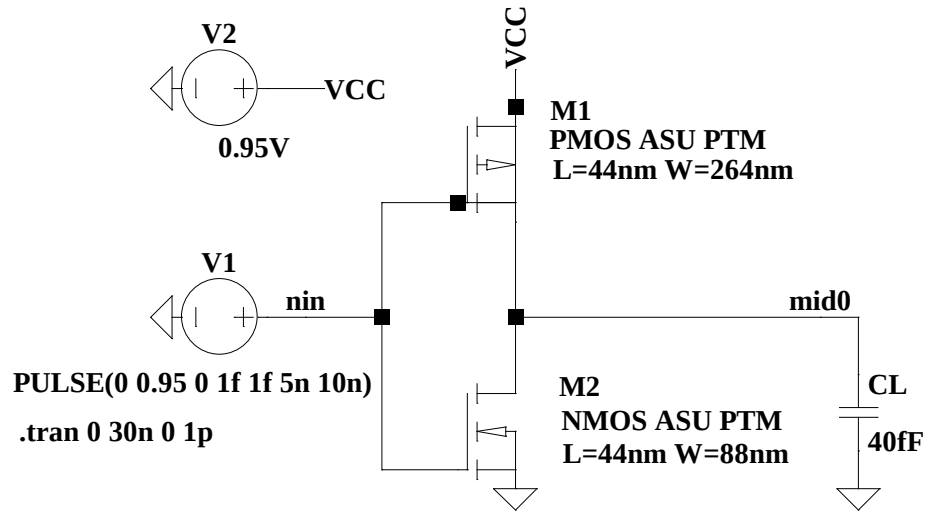


Figure 3.56: The schematic of single inverter with $W_p/W_n=3$.

Figure 3.56 indicates the schematic of an inverter circuit in which the width of the PMOS is three times the width of the NMOS. Also the calculation commands required for the delay calculation are added to the schematic spice netlist. In this way, necessary conditions have been met to calculate the falling and rising delays.

Figure 3.57 shows the simulation output of an inverter circuit with $W_p/W_n=3$. The blue signal is the input signal applied to the inverter and the red signal is the voltage change graph on the inverter output.

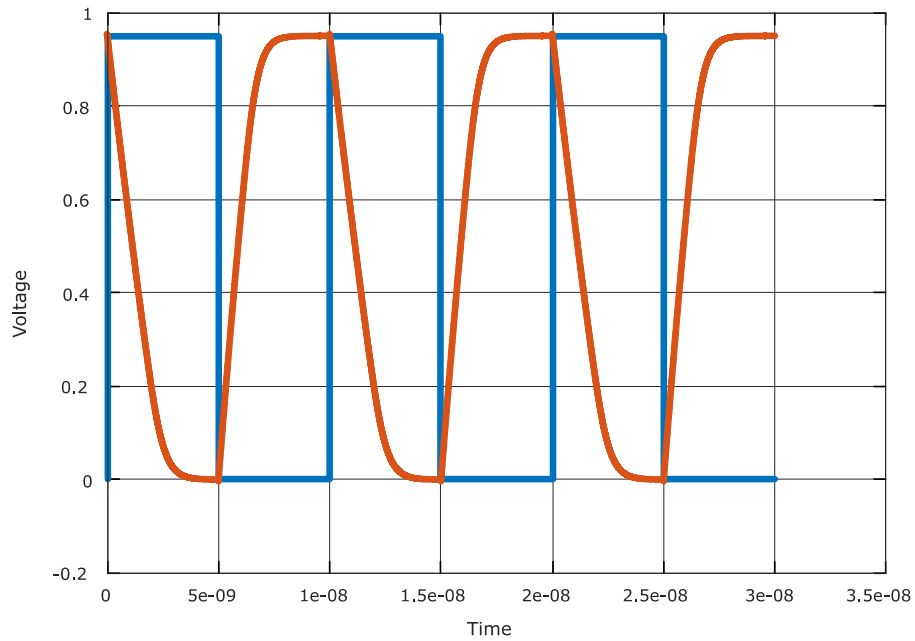


Figure 3.57: The plot of a single inverter with $W_p/W_n=3$.

```

*****
ngspice 1 -> source inverter_single_ratio2.cir
Circuit: * testing propagation delay of an inverter
Doing analysis at TEMP = 27.000000 and TNOM = 27.000000
Warning: This model supports BSIM4.2.1, 4.3.0 and 4.4.0; you specified a wrong version number.
Warning: (B1 + Weff) may be too small.
Warning: tox, toxp and dtoc all given and tox != toxp + dtoc; dtoc ignored.
Warning: This model supports BSIM4.2.1, 4.3.0 and 4.4.0; you specified a wrong version number.
Warning: vin: no DC value, transient time 0 value used

Initial Transient Solution
-----
Node                Voltage
----                -
mid0                0.949999
nin                 0
nvdd                0.95
nvss                0
vin#branch          3.15655e-10
vss#branch          0
vdd#branch          -3.16671e-10

Reference value : 2.98293e-08
No. of Data Rows : 30107

Measurements for Transient Analysis
tdr                  = 8.350327e-10 targ= 2.583503e-08 trig= 2.500000e-08
tdf                  = 1.208639e-09 targ= 2.120864e-08 trig= 2.000000e-08

```

Figure 3.58: The delay results of a single inverter with $W_p/W_n=3$.

Figure 3.58 shows the delay results of an inverter gate with a $W_p/W_n=3$ dimensioning ratio. The rise delay is named t_{dr} and is 8.35×10^{-10} seconds. The fall delay is also called t_{df} and is 1.21×10^{-9} seconds.

The results of the three simulations above are shared. Delays in the rise and fall in simulation were examined. Equivalent resistances of NMOS and PMOS were also calculated.

When we compare delays and equivalent resistances, we see that the second simulation is what we expect and give the result. In other words, when the width/length amount of PMOS is twice the width/length of NMOS, we found that the circuit is symmetrical. This was what we wanted and expected. In the simulations for the next MOSFET models, these dimensions will be used.

The work that we have done so far has been to identify what would be the optimum sizing for an inverter structure. In the next step, the optimum inverter chain will be designed to examine how it is designed. The delays in the chain and the appropriate sizing required for the chain will be determined.

Figure 3.59 describes the working principle of a transistor. The principle of operation has been tried to be expressed simply by the switch and resistance representation. From the moment the V_{GS} value is greater than the threshold voltage, the transistor has started to operate. In addition, the connection between the V_{DS} voltage and the current flow is seen. There was an increase in the current flowing through the transistor due to the change in voltage between the V_{DS} .

The speed of this increase started to decrease after a point as can be seen from the graph. Accordingly, the equivalent resistance formula of a transistor is indicated. We will use this equivalent resistance formula (3.28) in the next step.

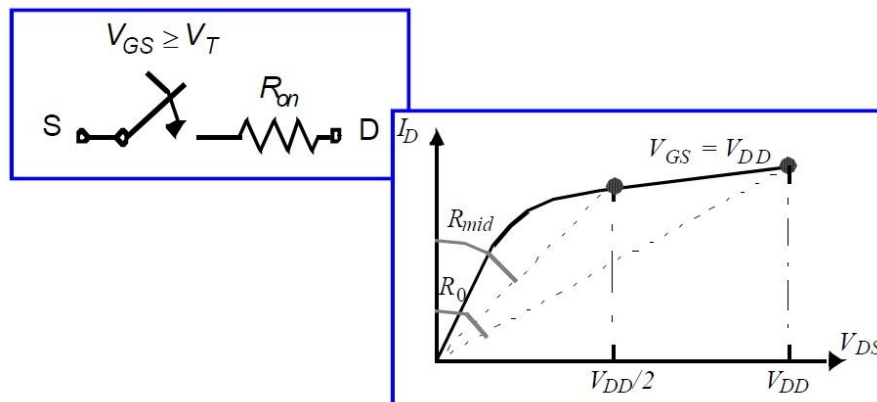


Figure 3.59: The transistor as a switch.

$$R_{eq} = \frac{1}{2} \left(\frac{V_{DD}}{I_{DSAT} (1 + \lambda V_{DD})} + \frac{V_{DD}/2}{I_{DSAT} (1 + \lambda V_{DD}/2)} \right) \approx \frac{3}{4} \frac{V_{DD}}{I_{DSAT}} \left(1 - \frac{5}{6} \lambda V_{DD} \right) \quad (3.28)$$

The delay formula (3.29) is given below. The delay of a transistor or a MOSFET can be determined according to the following formula. The delay varies depending on the resistance and capacity. In addition to the previous Resistance formula, the coefficient 0.69 and the capacity expression are present.

$$\left. \begin{aligned} t_{pHL} &\approx 0.69 \times \frac{3}{4} \frac{V_{DD}}{I_{DSAT_n}} \left(1 - \frac{5}{6} \lambda V_{DD} \right) C_L \\ I_{DSAT_n} &= k' \frac{W}{L} \left[(V_{GS} - V_T) V_{DSAT} - \frac{1}{2} V_{DSAT}^2 \right] \end{aligned} \right\} \quad \lambda = 0 \quad (3.29)$$

$$t_{pHL} \approx 0.52 \frac{C_L V_{DD}}{(W/L)_n k'_n V_{DSATn} (V_{DD} - V_{Tn} - V_{DSATn}/2)}$$

As can be seen from the formula (3.29), we can make some changes to reduce the delay time. For example, we can increase the V_{DD} to reduce the delay ($V_{DD} \square V_T + V_{DSAT}/2$). We can increase W to reduce the delay. We can also reduce C_L . To summarize, V_{DD} increase, W (width) increase, or capacity (C_L) drop improves the delay. However, doing all of these together may not bring optimum benefit. It may be necessary to make a trade off on the changes to be made to make an optimum design. This trade off study will be shown in the advanced stages.

The increase in V_{DD} has a significant effect on delay. However, according to the ASU PTM model we used in our simulations, our working voltage is 0.95V. In our study, we focused on capacity and W (width) to make the inverter chain design at the best level. We have made improvements on this issue. According to the formula, C_L and W will change at different locations but the C_L value will change with the change of W . That is, the C_L value is associated with the W value. However, by doing a good design study, the C_L is reduced to an optimum, resulting in optimum driving and delay.

Sizing of NMOS PMOS: As shown in the chart below, when W_p/W_n ratio changes, the amount of delay is changed. If we keep the W_p/W_n rate low from the graph, the passing from low to high is long and the delay from high to low is short. When we keep the W_p/W_n ratio higher than a certain value, the transition from high to low is long and the delay from low to high is short. But the point where the three curves

intersect is the point where the two delays are equal (Figure 3.60). The W_p/W_n value at this point is the appropriate sizing value. With this sizing ratio, the inverter gate performs its function with symmetrical behavior.

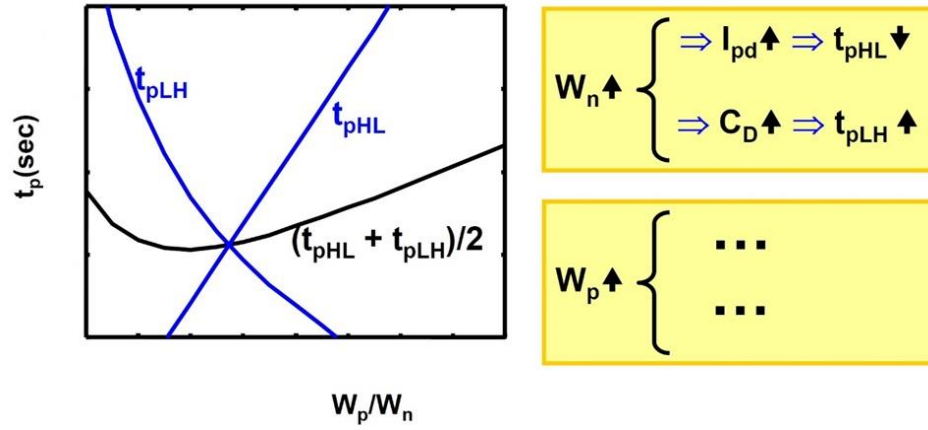


Figure 3.60: Propagation delay as a function of W_p/W_n .

On the other hand, preparing an inverter at this W_p/W_n ratio does not mean that the minimum delay can be achieved. But only delay when designing is not considered. It is also very important that the inverter performs its task correctly and reliable. And also the physical area that the gate occupies is another consideration.

In this study, the rate of W_p/W_n at the point where the rising and falling delay is closest to each other will be taken as the basis. Studies will be conducted to determine this point. The design and simulations made within the scope of the study were prepared in the light of these points and the results were shared in the following headings.

Intrinsic and Extrinsic Loads: In Figure 3.61 we see that two inverters are connected in succession. When such a connection is established, the capacity values are formed in different places on the circuit. The locations and nomenclature of these capacities are shown.

The capacity values are grouped under three headings: Intrinsic, extrinsic and parasitic. The parasitic capacity effect can be neglected or added to other capacities.

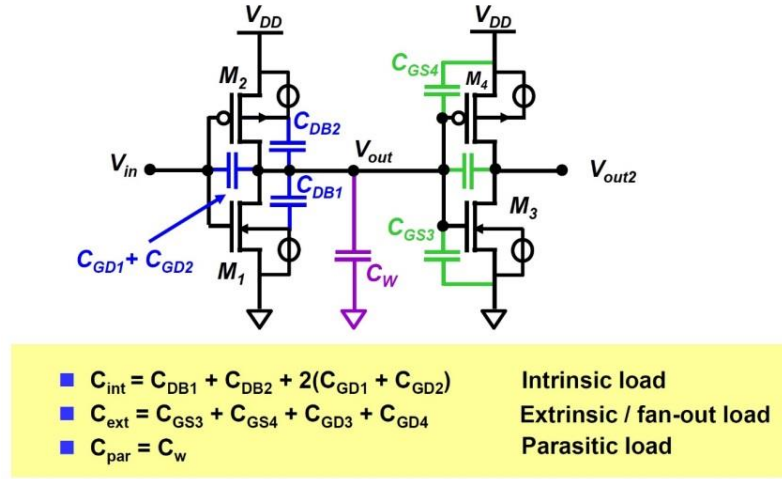


Figure 3.61: Intrinsic vs Extrinsic vs Parasitic load capacitance.

In the following, the previously mentioned delay formula (3.30) is given. Here the delay formula depends on the intrinsic and extrinsic capacities.

$$t_p = 0.69 R_{eq} (C_{int} + C_{ext}) = 0.69 R_{eq} C_{int} \left(1 + \frac{C_{ext}}{C_{int}} \right) \quad (3.30)$$

Below is an inverter with a minimum size, but with a suitable W_p/W_n ratio. This inverter formula is a delay (t_{p0}) formula for an inverter gate with no load and only intrinsic capacity.

R_0 : A resistance symbol of a minimum inverter gate with a specified W_p/W_n ratio.

C_0 : An intrinsic capacity symbol of an inverter of a minimum size.

Depending on the above values, the following formula (3.31) gives the delay value of an inverter with only the intrinsic capacity and without an output load.

$$T_{p0} = 0.69 \times R_0 \times C_0 \quad (3.31)$$

W_n and W_p values of an inverter in are minimum optimum size as shown below.

$$W_n = W_{min}, \quad W_p = \beta \times W_{min}$$

When the inverter is enlarged physically, the formula (3.32) is as follows. The S value below is an inverter's physical expansion factor.

$$W_n = S \times W_{min}, \quad W_p = S \times \beta \times W_{min} \quad (3.32)$$

When we revise the Inverter with S;

$R_{eq} = R_0/S$ and $C_{int} = S \times C_0$. Put them in the form of the above formula. In this way, the following general formula (3.33) is found.

$$t_p = t_{p0} \left(1 + \frac{C_{ext}}{SC_0} \right) \quad (3.33)$$

If the S value is increased until $SC_0 \gg C_{ext}$, the delay will decrease. The graph (Figure 3.62) below shows this decrease due to S.

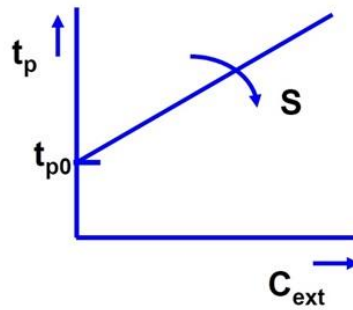


Figure 3.62: Isolated inverter sizing graph.

What should be done to drive a load with an optimum delay in the light of the information obtained so far? At this stage work on this subject will be done. Let's find out how to get the optimum number of inverter gates and inverter sizing to drive a load.

Figure 3.63 has 2 inverter gates. Suppose the first inverter is optimally dimensioned and fixed. When the size of the second inverter is increased, the delay caused by the second inverter decreases, but the extrinsic capacity of the first inverter increases.

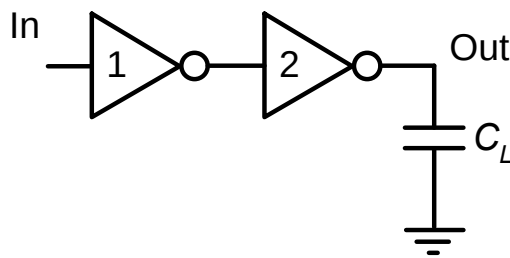


Figure 3.63: Inverter chain with two inverters.

When C_L is known, how many inverters should be used to minimize the delay and how should they be sized? We will try to find these values.

When looking at this formula from the fanout point, the following formula (3.34) is achieved.

C_{gin} : input gate capacitance

$C_{ext}/C_{gin} = SC_0/C_{gin}$: self loading coefficient property of technology, typically $\gamma \approx 1$

$f = C_{ext}/C_{gin}$: effective fanout

$$\begin{aligned} f/\gamma &= (C_{ext}/C_{gin}) \times (C_{gin}/SC_0) = C_{ext}/SC_0 \\ t_p &= t_{p0} \left(1 + \frac{C_{ext}}{SC_0} \right) = t_{p0} \left(1 + \frac{f}{\gamma} \right) \end{aligned} \quad (3.34)$$

Figure 3.64 shows the inverter chain diagram with an N number of Inverter Gates.

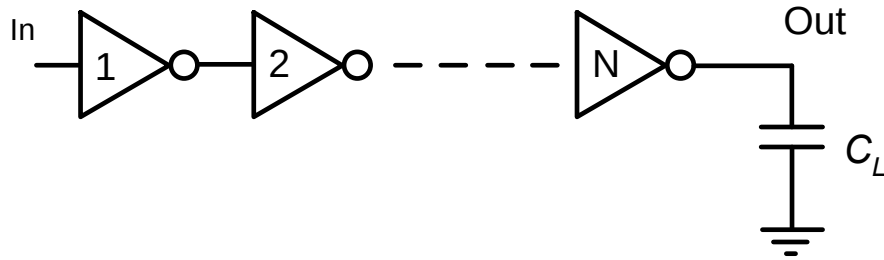


Figure 3.64: Inverter chain with N number inverter gates.

The following generalizations, simulations and formulas are formed from Figure 3.64.

$$\begin{aligned} t_p &= t_{p,1} + t_{p,2} + \dots + t_{p,N} \\ t_{p,j} &= t_{p0} \left(1 + \frac{f_j}{\gamma} \right) = t_{p0} \left(1 + \frac{C_{gin,j+1}}{\gamma C_{gin,j}} \right) \\ t_p &= \sum_{j=1}^N t_{p,j} = t_{p0} \sum_{j=1}^N \left(1 + \frac{C_{gin,j+1}}{\gamma C_{gin,j}} \right), \quad C_{gin,N+1} = C_L \end{aligned} \quad (3.35)$$

The delay formula (3.35) with the N-1 unknown $(C_{gin,2} \dots C_{gin,N})$ is indicated above. If the formula (3.36) above is derived from $C_{gin,j}$;

$$\frac{\partial t_p}{\partial C_{gin,j}} = t_{p0} \left(\frac{1}{\gamma C_{gin,j-1}} - \frac{C_{gin,j+1}}{\gamma (C_{gin,j})^2} \right) = 0, \quad j = 2, \dots, N-1 \quad (3.36)$$

$$\frac{1}{a} = \frac{b}{c^2} \Rightarrow c = \sqrt{ab}$$

Here we see that the optimum size for each inverter section must have the geometric mean of the two neighbors.

$$C_{gin,j} = \sqrt{C_{gin,j-1} \times C_{gin,j+1}}, \quad j = 2, \dots, N-1 \quad (3.37)$$

Square the above equation (3.38);

$$C_{gin,j}^2 = C_{gin,j-1} \times C_{gin,j+1} \quad (3.38)$$

When we write the above equation (3.38) as follows (3.39), it is seen that the load capacity and input capacity ratio in the inverter chain is the same in every section of the chain.

$$\frac{C_{gin,j}}{C_{gin,j-1}} = \frac{C_{gin,j+1}}{C_{gin,j}} \quad (3.39)$$

With the following statement (3.40), it can be seen that the same delay and the same fanout in each section. The geometric mean of the fanout of the whole path gives the fanout of a single section.

$$f_j = \frac{C_{gin,j+1}}{C_{gin,j}} = N \sqrt{\frac{C_L}{C_{gin,1}}} = N \sqrt{F}$$

$$t_{p,j} = t_{p0} \left(1 + \frac{f_j}{\gamma} \right) = t_{p0} \left(1 + \frac{C_{gin,j+1}}{\gamma C_{gin,j}} \right) = t_{p0} \left(1 + \frac{N \sqrt{F}}{\gamma} \right) \quad (3.40)$$

$$f_1 = f_2 = f_3 = \dots = F^{1/N} \quad f_1 \times f_2 \times f_3 \times \dots = F \quad F = C_L / C_{gin,1}$$

Figure 3.65 shows an inverter with a fixed number. In addition, input and output capacitances are shown. Fanout and dimensioning are also summarized. The previous theoretical calculations are simply explained by the Figure 3.65 representation.

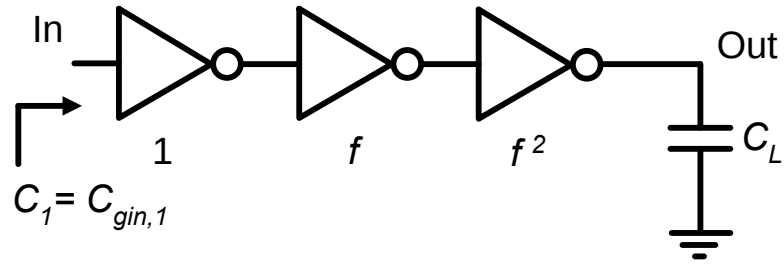


Figure 3.65: Optimum sizing summary for an inverter chain.

In this way, the way in which a particular inverter chain circuit should be installed is theoretically shared as summarized in Figure 3.65. After this point, some simulations were made to experience the theoretical knowledge described. The data obtained as a result of the simulations were shared.

Simulation 1: Rise and delay times of a single unloaded inverter

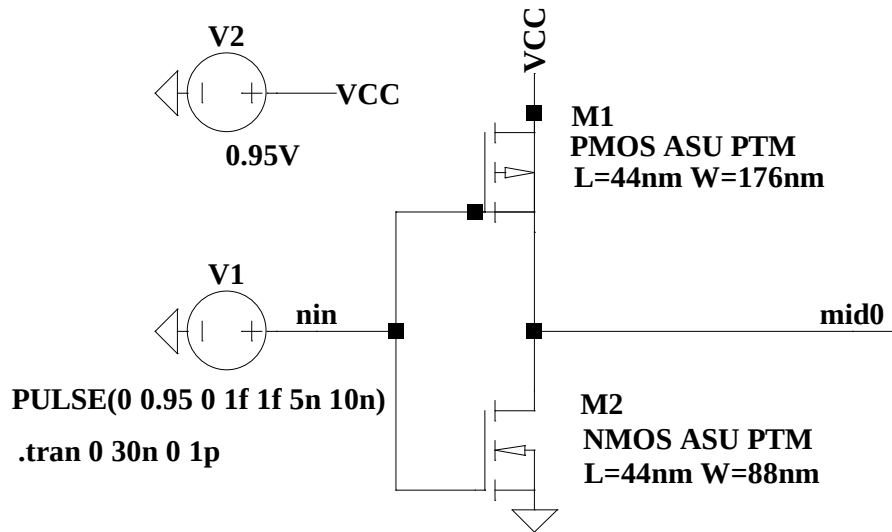


Figure 3.66: Single inverter gate schematic without load.

Figure 3.66 shows the schematic of an inverter circuit with $W_p/W_n = 2$. The output capacitor is not added to the circuit to be simulated. After the circuit was prepared in this way, delay calculation commands were added to the netlist of the schematic. In this way, the necessary conditions have been met to calculate the falling and rising delays.

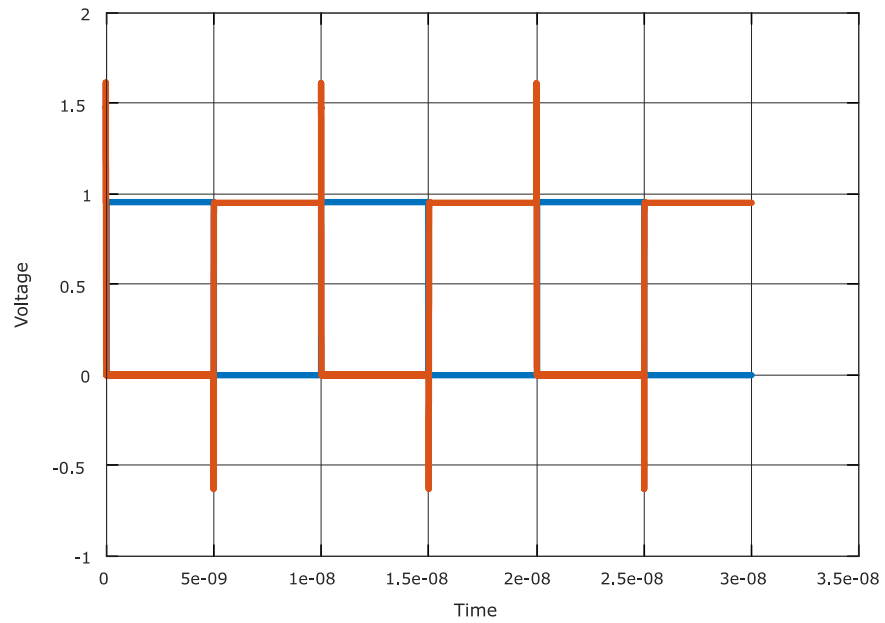


Figure 3.67: Single inverter output without load capacitor.

Figure 3.67 shows the simulation result of a single inverter circuit without load. Here the blue signal is the input signal and the red signal is the output signal.

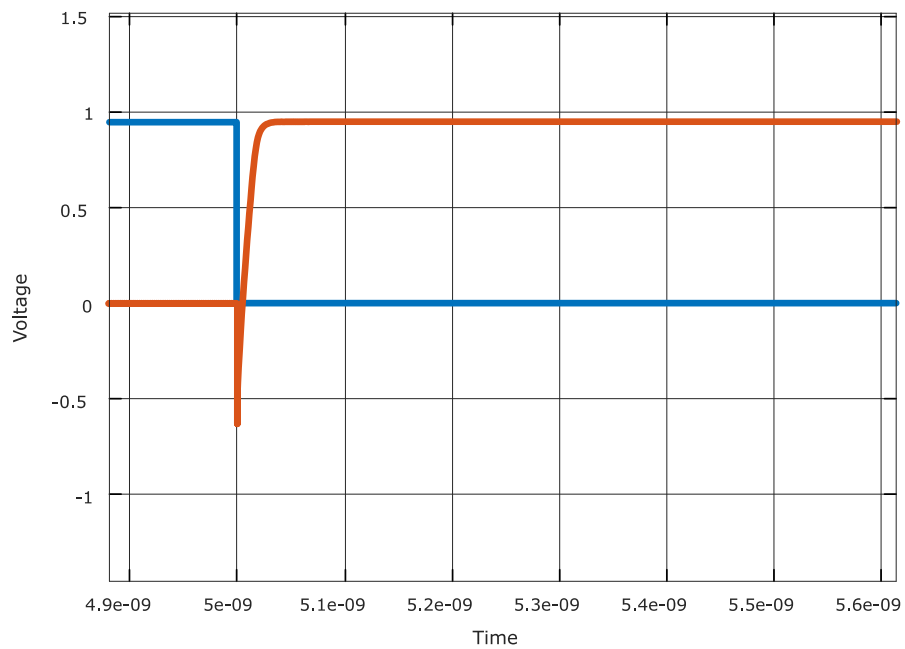


Figure 3.68: Single inverter output (zoom) without load capacitor.

Figure 3.68 shows the simulation result of a single inverter circuit without load. Here the blue signal is the input signal and the red signal is the output signal. Graphics are zoomed differently of Figure 3.67.

Figure 3.69 shows the delay results of a single inverter gate prepared without load capacitor. The rise delay is named t_{dr} and is 1.14×10^{-11} seconds. The fall delay is also called t_{df} and is 1.22×10^{-11} seconds.

```
ngspice 1 -> source inverter_single.cir
Circuit: * testing propagation delay of an inverter
Doing analysis at TEMP = 27.000000 and TNOM = 27.000000
Warning: This model supports BSIM4.2.1, 4.3.0 and 4.4.0; you specified a wrong version number.
Warning: (B1 + Weff) may be too small.
Warning: tox, toxp and dtox all given and tox != toxp + dtox; dtox ignored.
Warning: This model supports BSIM4.2.1, 4.3.0 and 4.4.0; you specified a wrong version number.
Warning: vin: no DC value, transient time 0 value used

Initial Transient Solution
-----
Node                Voltage
----                -
mid0                 0.949999
nin                  0
nvdd                 0.95
nvss                 0
vin#branch           2.06294e-10
vss#branch           0
vdd#branch           -2.0731e-10

Reference value : 2.93233e-08
No. of Data Rows : 30107

Measurements for Transient Analysis
tdr                  = 1.139261e-11 targ= 2.501139e-08 trig= 2.500000e-08
tdf                  = 1.216306e-11 targ= 2.001216e-08 trig= 2.000000e-08
```

Figure 3.69: Delay measurement of the single inverter without load capacitor.

According to the ASU PTM 22nm MOSFET model, the sizing of a minimum inverter was determined in the previous sections. In this simulation, the delay at a minimum size inverter with no load was examined. As a result, the rising delay was seen as 1.14×10^{-11} and the falling delay was seen as 1.22×10^{-11} . Delay results will be given in tabular form at the end of all simulations.

RC Charging Circuit: Below is a series of RC circuits (Figure 3.70) [38]. When the switch is closed, the current flows from the circuit and the capacitor starts to charge. The capacitor continues to be charged through the resistor so long as the voltage on the capacitor is equal to the supply voltage.

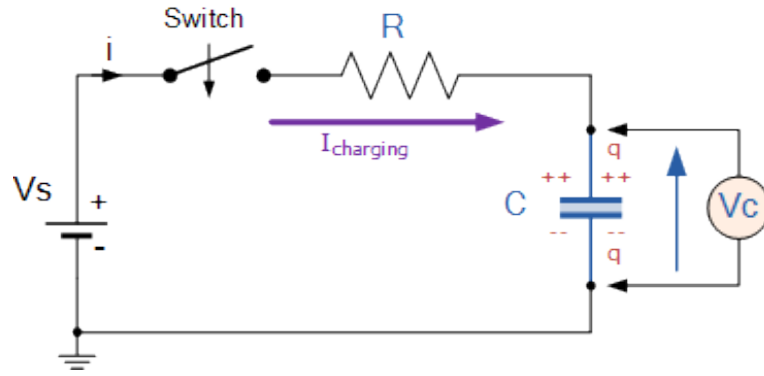


Figure 3.70: RC charging circuit.

Suppose the capacitor is completely discharged and the switch is on. At $t = 0$, $V_C = 0$ and $q = 0$. At the moment when the switch is closed, $V_C = 0$ and the capacitor acts as a short circuit. Immediately after the switch is closed, the maximum current from the circuit starts to flow. The current passage is only prevented by resistance. For this reason, the value of V_C increases rapidly at first.

Kirchhoff's voltage law (KVL), the voltage formula of the above Figure 3.70 circuit is in the expression 3.41.

$$V_s - R \times i(t) - V_C(t) = 0 \quad (3.41)$$

The current indicated by I_{charging} on Figure 3.70 is given by the following formula (3.42).

$$i = V_s / R \quad (3.42)$$

RC Charging Circuit Curves: In the first of the two charts above Figure 3.71, the capacitor has a charge curve. The capacitor is fully discharged at $t = 0$. When the switch is closed, it is seen that the capacitor is rapidly charged. The voltage on the capacitor at time $1T$ is $0.63V_s$. The T (τ) value is equal to $R \times C$. It will be seen that this T value is used in the simulation demonstrations to be shown in the future. In addition, at $5T$, the capacitor is fully charged.

After the capacitor is fully charged, the current will not flow through the circuit and the capacitor will act as open circuit. $t = Q$, $i = 0$, $q = Q = CV$. The period up to $4T$ is called the period of transient period, and after $4T$ it is called steady state period.

The following equation (3.43) is the equation at which the voltage on the capacitor can be found at any time during charging.

$$V_C = V_s (1 - e^{-t/RC}) \quad (3.43)$$

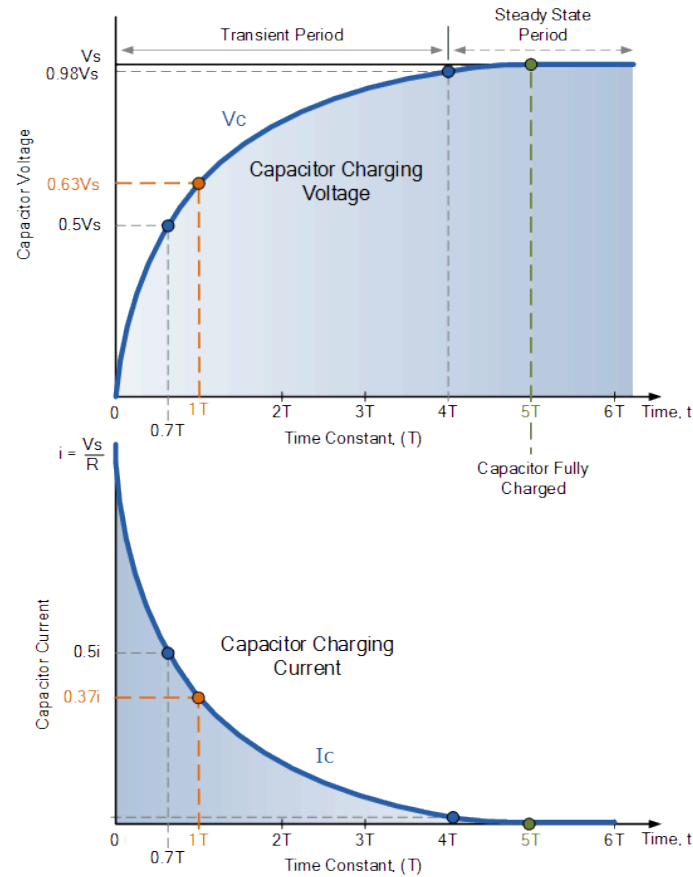


Figure 3.71: RC charging circuit curves [38].

The following Table 3.2 shows the voltage and current percentages on the capacitor at certain tie points in the RC circuit that is charging.

Table. 3.2: Voltage and current percentages on the capacitor.

Time Constant	RC Value	Percentage of Maximum	
		Voltage	Current
0.5	$0.5T = 0.5RC$	39.3%	60.7%
0.7	$0.7T = 0.7RC$	50.3%	49.7%
1.0	$1T = 1RC$	63.2%	36.8%
2.0	$2T = 2RC$	86.5%	13.5%
3.0	$3T = 3RC$	95.0%	5.0%
4.0	$4T = 4RC$	98.2%	1.8%
5.0	$5T = 5RC$	99.3%	0.7%

Simulation 2: Achieving the input capacity of a single inverter

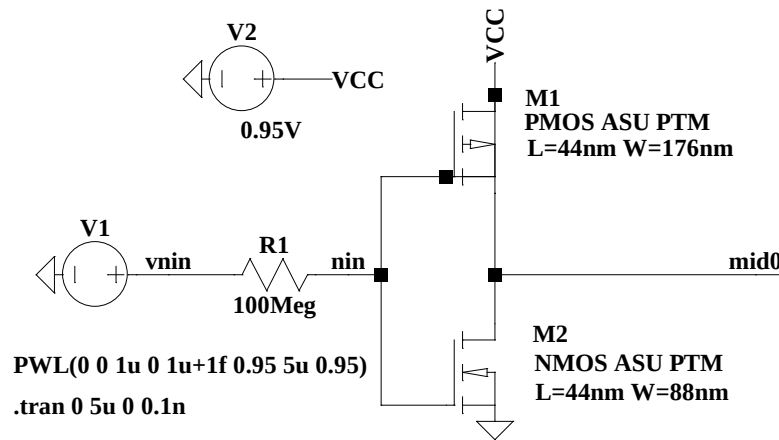


Figure 3.72: Single inverter schematic for input capacitance.

Figure 3.72 shows the schematic of a single inverter circuit. There is a great 100Meg resistance to the inverter input. The calculation commands for the τ value and inverter input capacitance are included in the prepared netlist for this schematic. τ time is the time to reach the output voltage of $0.63 \times V_{CC}$. When the τ value is divided by the added 100Meg resistor value, the capacitance of the gate can be found.

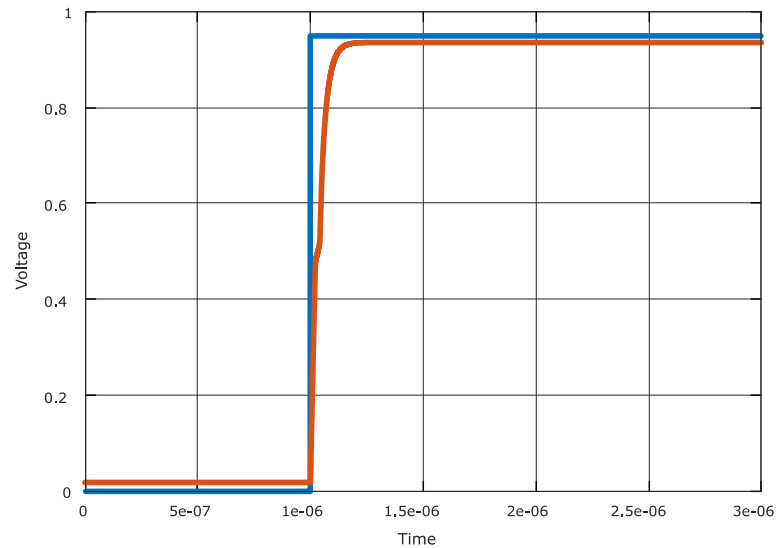


Figure 3.73: The plot of the single inverter simulation.

Figure 3.73 shows the result of a single inverter simulation with a 1×10^8 ohm resistor in front. The blue signal is the input signal and the red signal is the output signal.

```
ngspice 1 -> source inv_inp_cap_con_inp.cir
Circuit: * find the input capacitance of an inverter.
Doing analysis at TEMP = 27.000000 and TNOM = 27.000000
Warning: This model supports BSIM4.2.1, 4.3.0 and 4.4.0; you specified a wrong version number.
Warning: (B1 + Weff) may be too small.
Warning: tox, toxp and dtov all given and tox != toxp + dtov; dtov ignored.
Warning: This model supports BSIM4.2.1, 4.3.0 and 4.4.0; you specified a wrong version number.
Warning: vin: no DC value, transient time 0 value used

Initial Transient Solution
-----
Node          Voltage
----          -
out0           0.949999
nin            0.0183248
nvdd            0.95
nvin            0
nvss            0
vin#branch     1.83248e-10
vss#branch     0
vdd#branch     -1.84298e-10

Reference value : 4.94587e-06
No. of Data Rows : 50032

Measurements for Transient Analysis
tau            = 4.667069e-08 targ= 1.046671e-06 trig= 1.000000e-06
val cint       = 4.66707e-16
```

Figure 3.74: τ and input gate capacitance result for single inverter.

Figure 3.74 shows the calculated τ and input capacitance values. The input voltage of the inverter gate has a time to reach $0.63V_{dd}$. This time to reach $0.63V_{dd}$ is τ value. Its value is 4.66×10^{-8} seconds. The input capacitance was found by dividing the τ value by the 1×10^8 ohm resistance value. Input gate capacitance is 4.67×10^{-16} F.

In this simulation, a 1×10^8 ohm resistance was added before an inverter. No capacitor was added to the inverter input. The supply voltage was applied for a certain period of time before the resistance's input node. After the resistance, the voltage status at the inverter input was simulated. The time elapsed from the inverter input until the value of $0.63V_{dd}$ was calculated. When we divide this time (τ) value to the value of the resistance, we found the input capacity of an inverter. According to the results, the input capacitance of an inverter is approximately 0.5 fF as shown in Figure 3.75.

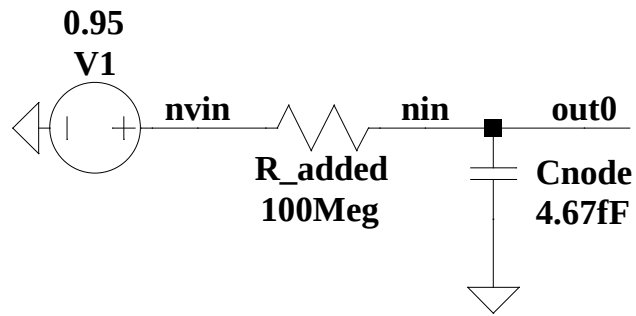


Figure 3.75: Node capacitance result for single inverter.

Simulation 3: Delay of a single inverter with 40fF load

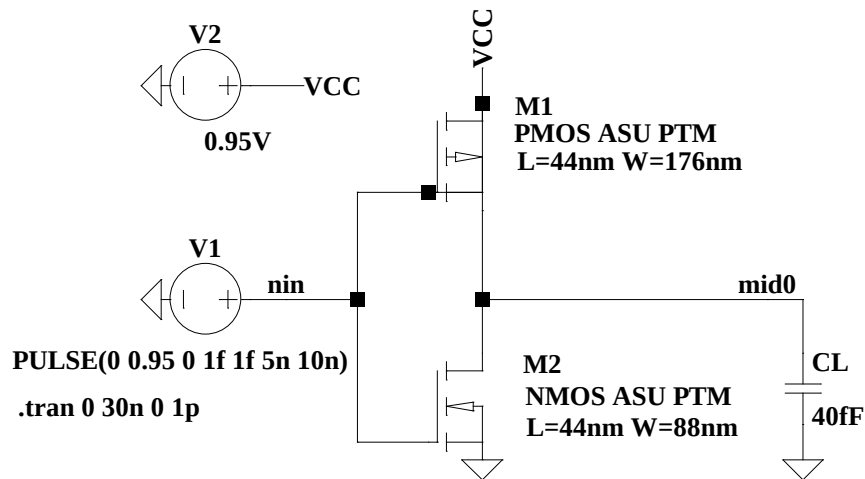


Figure 3.76: The schematic of a single inverter with 40fF load capacitor.

In the previous study, the input capacitance of the inverter gate was found. Output capacitance was determined as 40fF before starting inverter chain

simulations. Accordingly, Figure 3.76 shows the schematic of the inverter circuit with an output load of 40fF. The measurement command required to calculate the delay has also been added to the netlist of the circuit in the Figure 3.76.

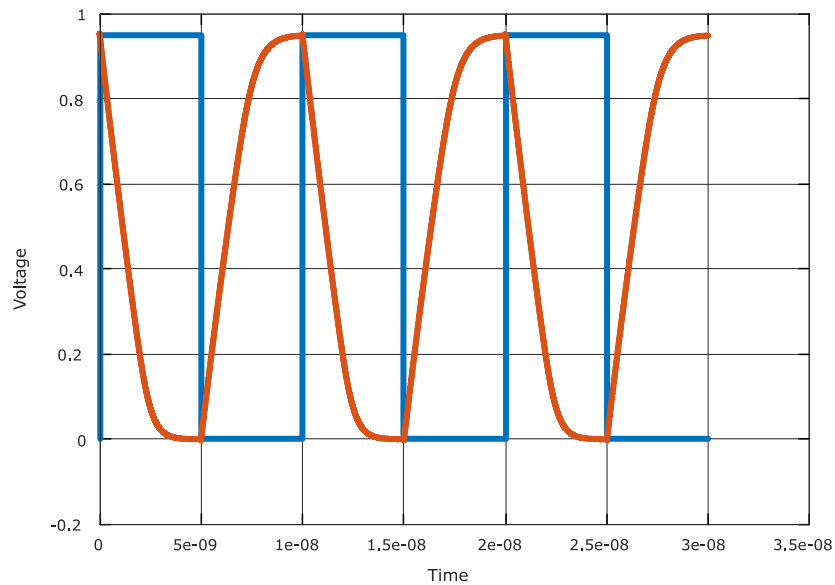


Figure 3.77: The plot of the single inverter with 40fF load capacitor.

Figure 3.77 shows the result of a single inverter simulation with a 40fF load. The blue signal is the input and the red signal is the output on the Figure 3.77.

Figure 3.78 shows the delay results of an inverter gate with a 40fF load. The rise delay is named tdr and is 1.27×10^{-9} seconds.

```
ngspice 1 -> inverter_single.cir
Circuit: * testing propagation delay of an inverter
Doing analysis at TEMP = 27.000000 and TNOM = 27.000000
Warning: This model supports BSIM4.2.1, 4.3.0 and 4.4.0; you specified a wrong version number
Warning: (B1 + Weff) may be too small.
Warning: toxe, toxp and dttox all given and toxe != toxp + dttox; dttox ignored.
Warning: This model supports BSIM4.2.1, 4.3.0 and 4.4.0; you specified a wrong version number
Warning: vin: no DC value, transient time 0 value used

Initial Transient Solution
-----
Node          Voltage
----          -
mid0          0.949999
nin           0
nvdd          0.95
nvss          0
vin#branch    2.06294e-10
vss#branch    0
vdd#branch    -2.0731e-10

Reference value : 2.61423e-08
No. of Data Rows : 30107

Measurements for Transient Analysis
tdr           = 1.272273e-09 targ= 2.627227e-08 trig= 2.500000e-08
```

Figure 3.78: Delay measurement of the single inverter with load.

In this simulation, a load of 40fF (Figure 3.79) was added to a single inverter output and the delay was evaluated. The delay according to the simulation result is $1.27e^{-9}$. The first of the 4 figures in the following illustration represents this simulation and indicates the result. A value of 0.5 is the input capacitance of the first inverter. The N value is the number of stages in a chain, f is the fanout value for an inverter, tdr is the increase delay.

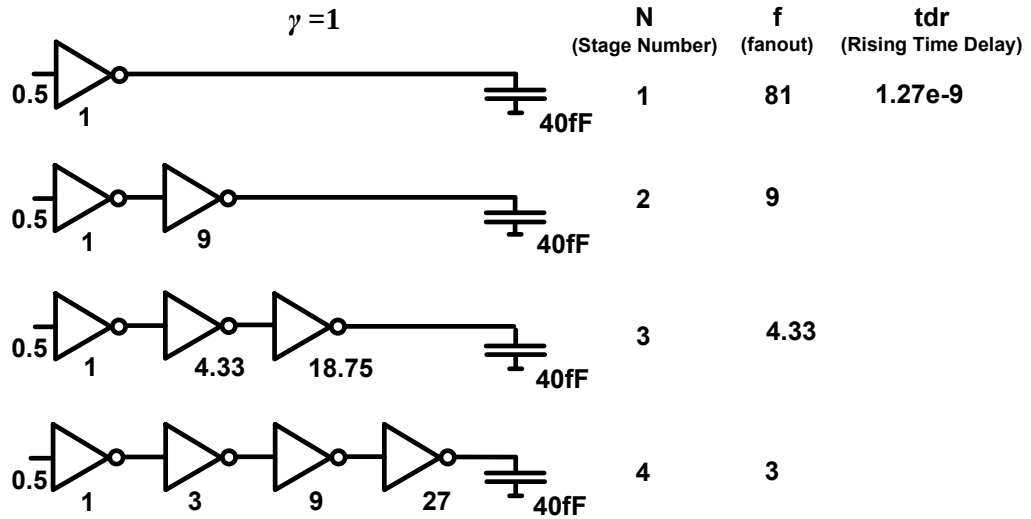


Figure 3.79: Inverter chain delay results.

Simulation 4: Delay of four inverters with 40fF load

We have found val_cint is 0.5fF. Let's assume f is 3. We determine stages number is 4.

Here;

$$f \times f \times f \times f = C_L / C_{int}$$

Let's assume the f is 3.

$$3 \times 3 \times 3 \times 3 = C_L / C_{int}$$

$$81 = C_L / 0.5 \text{ fF}$$

$$C_L = C_g + C_L$$

$$C_L = 40 \text{ fF approx.}$$

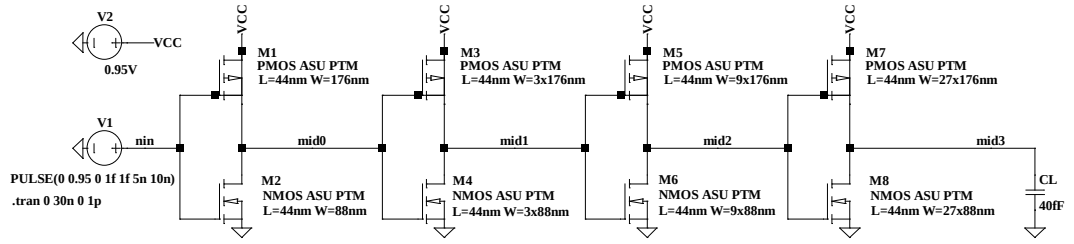


Figure 3.80: Inverter chain schematic with four inverters.

Figure 3.80 shows the schematic of an inverter chain circuit with 40fF output load consisting of four inverter gates.

The circuit is designed according to the optimum dimensioning and the necessary commands for delay are written.

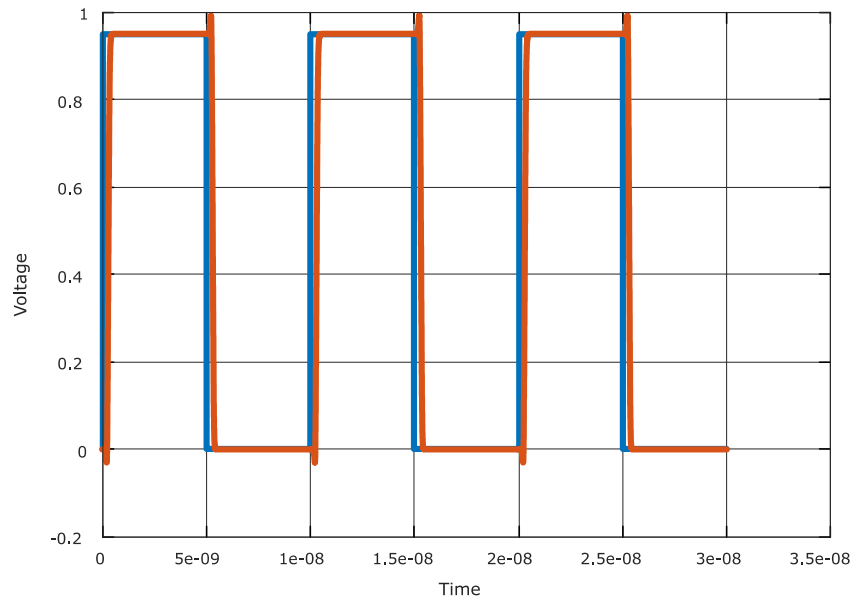


Figure 3.81: The output of inverter chain with four gates.

The Figure 3.81 shows the simulation result of the 4-gates chain. Blue refers to the input signal and red is the output at the end of the chain.

```

Circuit: * testing propagation delay of an inverter chain
Doing analysis at TEMP = 27.000000 and TNOM = 27.000000

Warning: This model supports BSIM4.2.1, 4.3.0 and 4.4.0; you specified a wrong version number.
Warning: toxe, toxp and dtox all given and toxe != toxp + dtox; dtox ignored.
Warning: This model supports BSIM4.2.1, 4.3.0 and 4.4.0; you specified a wrong version number.
Warning: This model supports BSIM4.2.1, 4.3.0 and 4.4.0; you specified a wrong version number.
Warning: toxe, toxp and dtox all given and toxe != toxp + dtox; dtox ignored.
Warning: This model supports BSIM4.2.1, 4.3.0 and 4.4.0; you specified a wrong version number.
Warning: toxe, toxp and dtox all given and toxe != toxp + dtox; dtox ignored.
Warning: This model supports BSIM4.2.1, 4.3.0 and 4.4.0; you specified a wrong version number.
Warning: toxe, toxp and dtox all given and toxe != toxp + dtox; dtox ignored.
Warning: This model supports BSIM4.2.1, 4.3.0 and 4.4.0; you specified a wrong version number.
Warning: (B1 + Weff) may be too small.
Warning: toxe, toxp and dtox all given and toxe != toxp + dtox; dtox ignored.
Warning: This model supports BSIM4.2.1, 4.3.0 and 4.4.0; you specified a wrong version number.
Warning: vin: no DC value, transient time 0 value used

Initial Transient Solution
-----
Node                                Voltage
----                                -
mid0                                0.949991
nin                                  0
nvdd                                0.95
mid1                                6.88652e-06
mid2                                0.949991
mid3                                8.58873e-07
nvss                                  0
vin#branch                          2.06296e-10
vss#branch                          0
vdd#branch                          -7.86263e-09

Reference value : 2.91683e-08
No. of Data Rows : 30107

Measurements for Transient Analysis
tdr                                = 3.042928e-10 targ= 2.030429e-08 trig= 2.000000e-08

```

Figure 3.82: Delay measurement of the inverter chain with four gates.

Figure 3.82 shows the delay results of an inverter gate chain. The rise delay is named tdr and is 3.04×10^{-10} seconds.

In this simulation, a chain consisting of four inverters (Figure 3.83) was created considering the load amount and fanout value. The rising delay was examined. And the result in the appropriate section in the table was expressed. The delay is 3.043×10^{-10} .

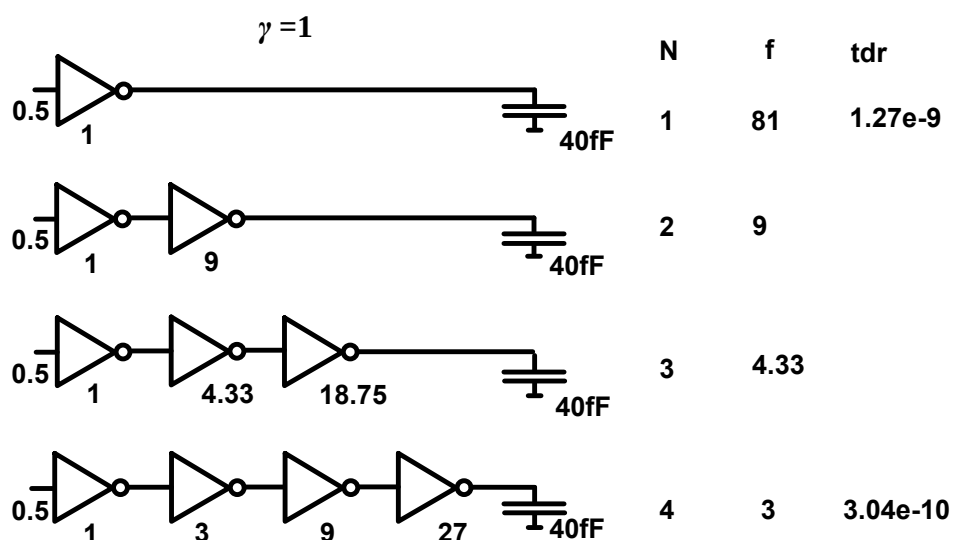


Figure 3.83: Inverter chain delay results.

Figure 3.83 shows the calculated delay values for a single inverter and the chain with four inverter gates. Let's expand the simulation and examine the chains with 2, 3 and 5 inverters.

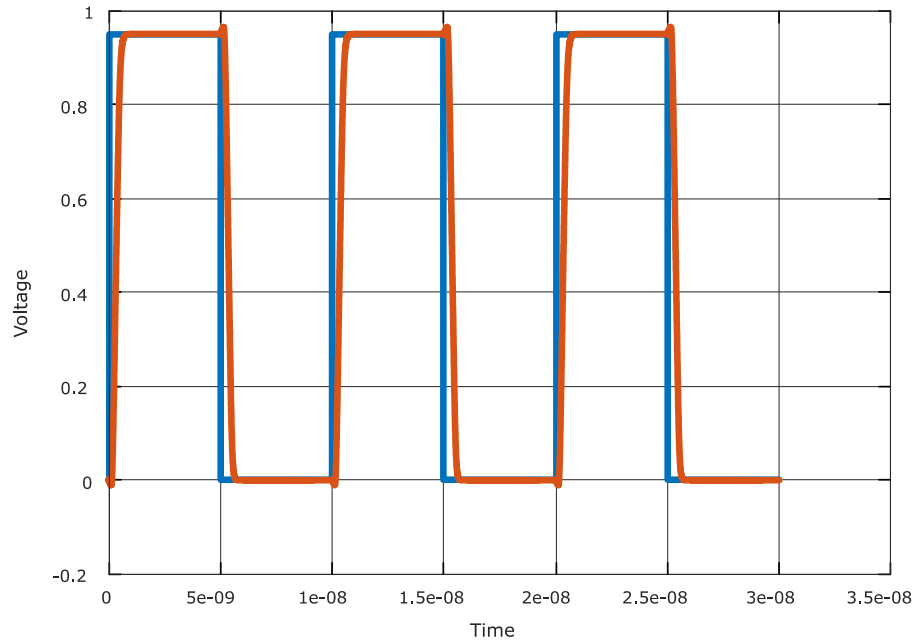


Figure 3.84: The output of inverter chain with two gates.

The Figure 3.84 shows the simulation result of the 2-gates chain. Blue refers to the input signal and red is the output at the end of the chain.

In this simulation, a chain consisting of two inverters was created considering the load amount and fanout value. The rising delay was examined. And the result in the appropriate section in the table was expressed. The delay is $3.44e^{-10}$.

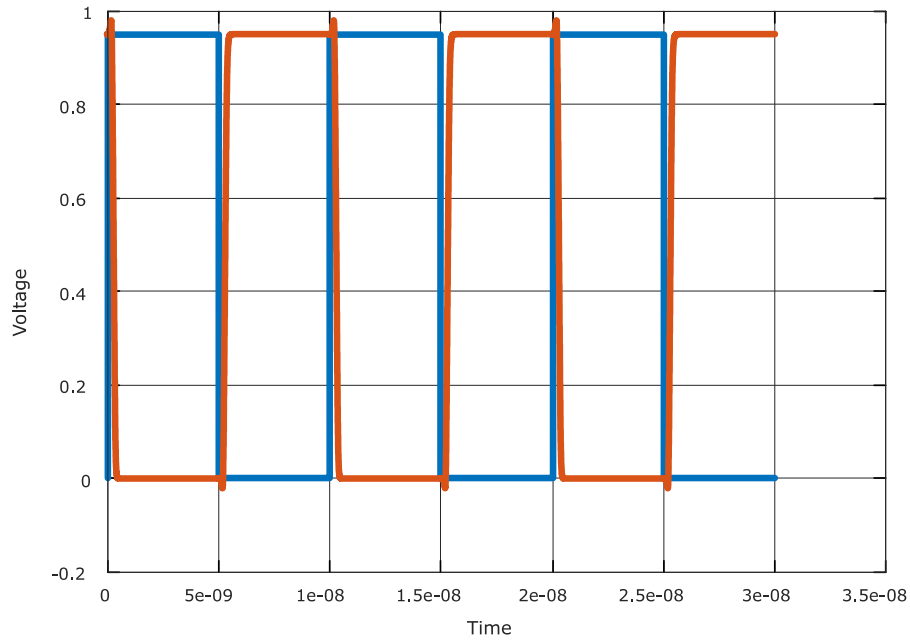


Figure 3.85: The output of inverter chain with three gates.

The Figure 3.85 shows the simulation result of the 3-gates chain. Blue refers to the input signal and red is the output at the end of the chain.

In this simulation, a chain consisting of three inverters was created considering the load amount and fanout value. The rising delay was examined. And the result in the appropriate section in the table was expressed. The delay is $3.042e^{-10}$.

The Figure 3.86 shows the simulation result of the 5-gates chain. Blue refers to the input signal and red is the output at the end of the chain.

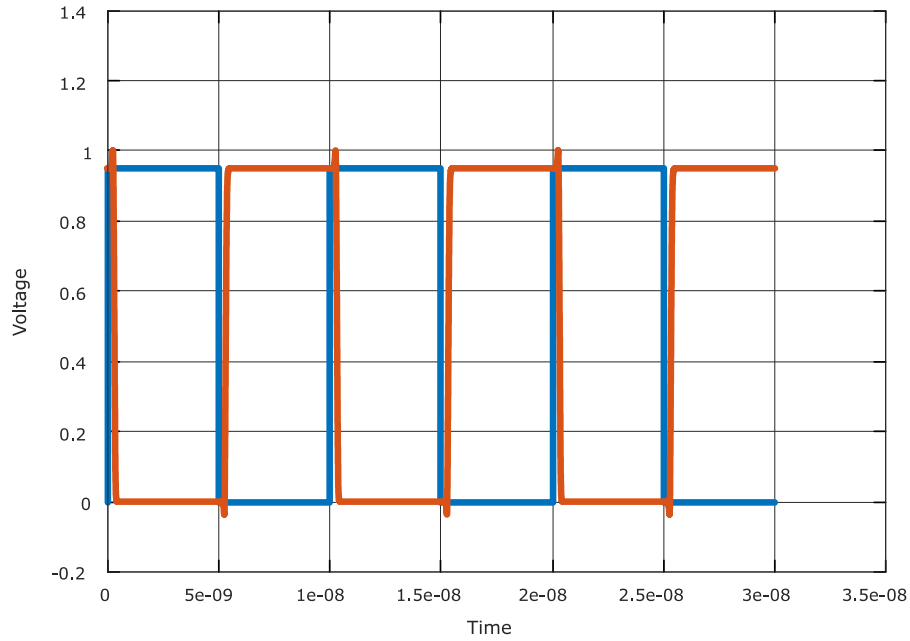


Figure 3.86: The output of inverter chain with five gates.

In this simulation, a chain consisting of five inverters was created considering the load amount and fanout value. The rising delay was examined. And the result in the appropriate section in the table was expressed. The delay is $3.34e^{-10}$.

In this section, inverter and inverter chain are examined. After the theoretical study, inverter structures were evaluated in simulation environment. NGSPICE environment was used in this simulation study. As MOSFET model, ASU PTM 22nm technology is used. As seen in the simulations, a circuit consisting of a single inverter was used at first. Here, the minimum and optimum sizing for a single inverter is determined. Then, the input capacitance was found for the single inverter. With this information, the basis of the simulation work for the inverter chain was created. Then, inverter chains with various inverter quantities were created. Five different inverter chains with 1, 2, 3, 4, 5 inverters were analyzed. And the results were expressed in the following illustration. According to the study, the chains of three and four inverters are the optimum inverter chains. Because the delay caused by these two chains is less than the others.

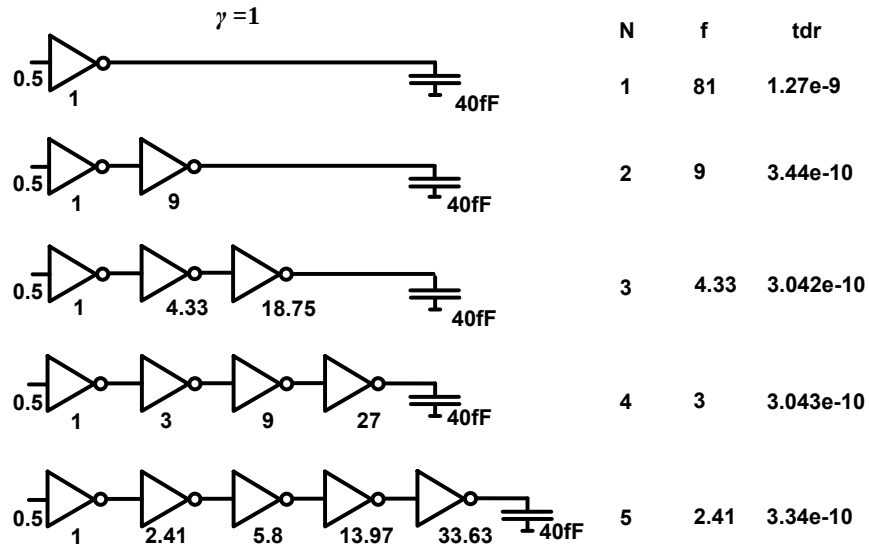


Figure 3.87: Inverter chain all delay results.

Figure 3.87 shows the calculated delay values for all five inverter chain circuits.

3.6. Class D Amplifier

In the D-class power amplifier (Figure 3.88), which is the other circuit examined in the study, the amplifying components are used as switching elements. In this circuit, the PWM modulated signal obtained at the comparator output is first amplified and then demodulated by applying it to a low pass filter inlet. In this study, it is analyzed how the parameter variations which may occur in the low pass filter components and the transistors in the triangle wave generator at the output of the related circuit lead to the disturbance of the output signal.

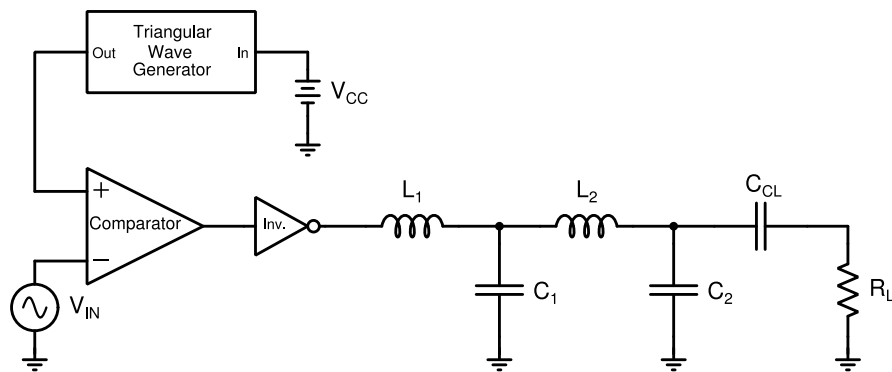


Figure 3.88: Class D amplifier circuit scheme.

For this purpose, the output data obtained for each parameter set given as input are sent to FFT tool and Fourier series coefficients are obtained and then total harmonic distortion (THD) is calculated. The respective circuit was also analyzed in terms of the average power spent.

In the block below (Figure 3.89), Class D amplifier is seen to be simpler.

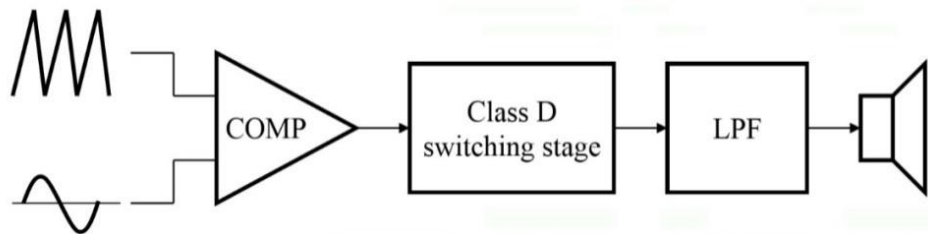


Figure 3.89: Class D amplifier circuit stages.

The following graph (Figure 3.90) shows the input signal, triangle signal and comparator circuit output signal. Where the sine wave amount is greater than the value of the triangular wave signal, the corresponding portion of the signal at the comparator circuit output is high. The comparator signal is switched to the PWM signal first by switching. Then, after the low pass filter, it is converted to the output signal which is the raised signal of the input signal.

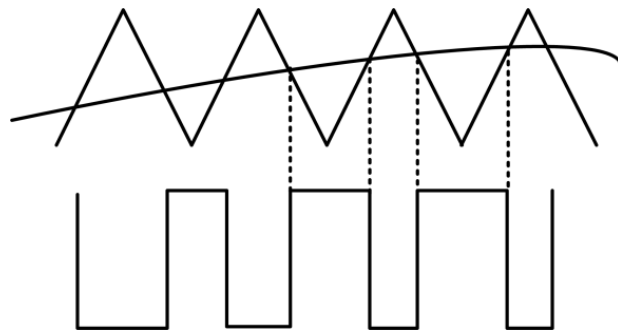


Figure 3.90: Triangle and PWM signals.

An ideal Class D amplifier does not generate noise and distortion on output, which ensures 100 percent efficiency. However, as shown below (Figure 3.91), in practice there are disturbances and noises caused by switching and different elements.

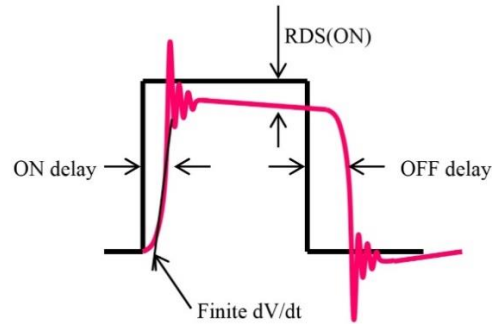


Figure 3.91: Major cause of degradation.

Simulation: Class D amplifier

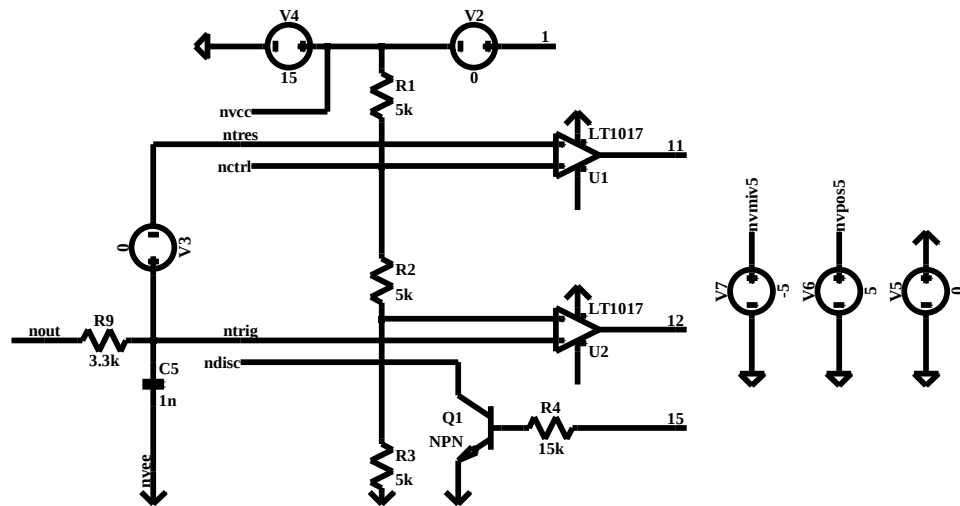


Figure 3.92: Class D amplifier schematic 1.

Figure 3.92 shows the first schematic of the Class D amplifier circuit. This schematic is a certain part of the internal structure of the 555 timer. 5k voltage divider resistors and comparator elements are internal components of the 555 timer chip. The purpose of using 555 is to produce the triangular wave required for the circuit. The components with U1 and U2 designators shown in the schematic are different in the spice netlist file with the extension ".cir" prepared for simulation. For U1 and U2 comparators, a separate subcircuit in netlist was written as in Figure 3.93.

```

.SUBCKT COMP_class_d 1 2 5 ngnd
* TERMINALS: 1-INPUT+, 2-INPUT-, 5-OUTPUT
* DIFF AMP WITH HYSTERESIS
EDIFF 3    ngnd    VALUE = { V(1) - V(2) + V(5)/500 }
* FREQUENCY RESPONSE
RP1  3    4    200
CP1  4    ngnd  100PF
* LIMITER
EOUT  5    ngnd  TABLE {V(4)} = (-1MV, -5V) (1MV, 5V)
.ENDS

```

Figure 3.93: Subcircuit netlist for U1 and U2 comparators.

The second sheet of the Class D amplifier circuit appears in Figure 3.94. Here, we see the RS Flip-Flop circuit consisting of 2 NOT and 2 NAND gates in the 555 timer.

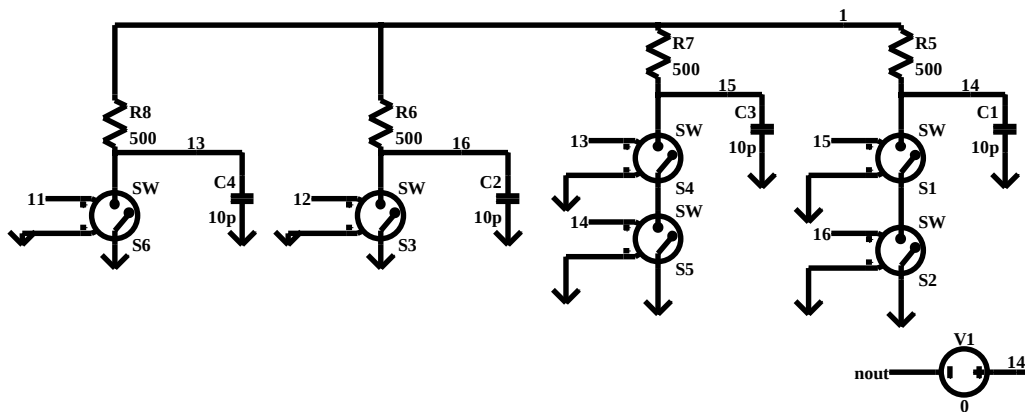


Figure 3.94: Class D amplifier schematic 2.

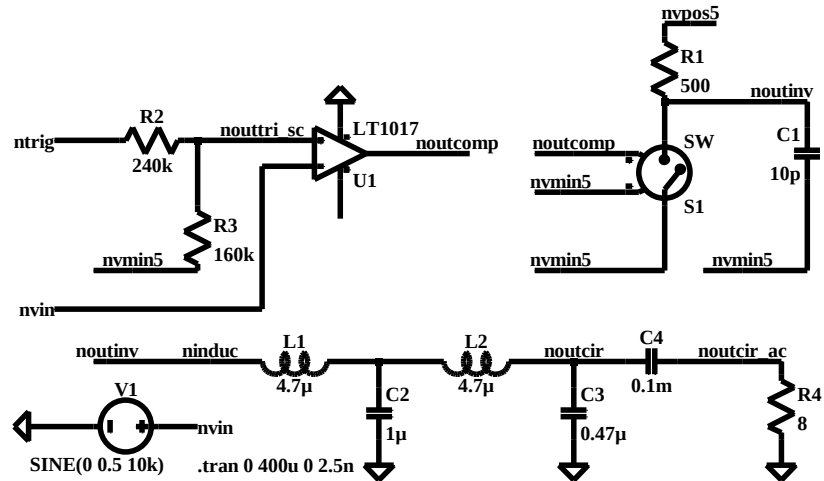


Figure 3.95: Class D amplifier schematic 3.

The third sheet of Class D amplifier appears in Figure3.95 above. In this sheet, it is seen that the sinus signal applied to the circuit with the triangular wave formed by 555 has passed through the comparator and reached to the NOT gate. After completing the crossing from the NOT gate, it also performs circuit switching. Then the signal passes through the low pass filter and the output signal is obtained. A suitable model for the switch components has been determined when creating the netlist of this circuit. In addition, a different subcircuit was created in netlist instead of the element which performs comparator in schematic. The simulation results obtained from the netlist of the Class D amplifier circuit will be given.

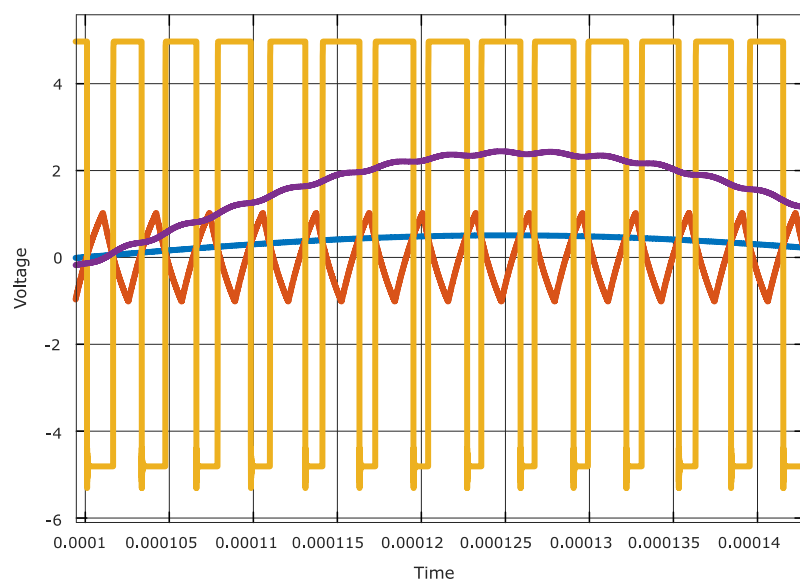


Figure 3.96: Class D amplifier plot.

Figure 3.96 shows the simulation result of Class D circuit via OCTAVE. All-important signals for Class D appear. Blue signal is input, purple signal is output. The yellow signal is a PWM signal and the red signal is the triangular wave signal. In the below plot (Figure 3.97), input, output and PWM signals are seen on NGSPICE environment.



Figure 3.97: Class D amplifier output on the NGSPICE Tool.

Figure 3.98 shows the input signal and the output signal. Blue signal is input, red signal is output. It is seen that the input signal is raised at the output. The Class D amplifier circuit works properly.

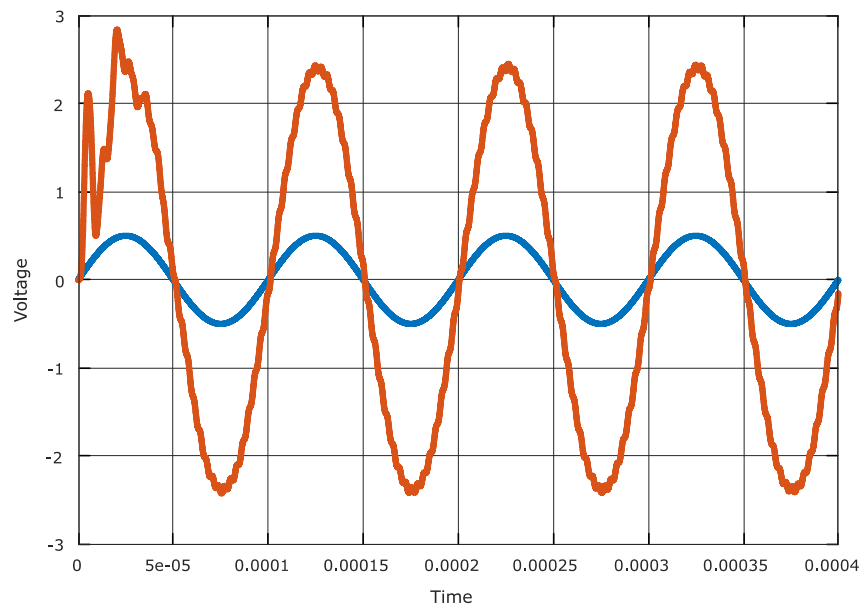


Figure 3.98: Class D amplifier input and output signals.

4. NUMERICAL RESULTS

In this section, information is given about Monte Carlo and gPC analysis applied for Class D amplifier and dynamic NAND circuits. The Monte Carlo application was implemented for the Class D amplifier circuit. The gPC method was also applied to the dynamic NAND gate. Technical information about the Monte Carlo analysis for the Class D amplifier circuit and the gPC method for dynamic NAND gate circuit will be given respectively.

4.1. Results of Monte Carlo Method

Firstly, the aim of the study on the Class D circuit is to try to determine the behavior change of the circuit as a result of certain deviations of some component parameters seen in circuit. Some MATLAB code files have been created to find the change that occurred as a result of variations in the component parameters. One of these files is the master code named mc_analyzer.

The mc_analyzer code is the main code that manages other pieces of code and files. This code first enters the Class D related file. It then pulls each value set of parameter 13 of the circuit from outside in a certain order. It sends this data set to the netlist of the circuit for simulation.

Mc_analyzer code also generates the power information from the input with the THD information of the output signal obtained as a result of the simulations and saves it to a specific file. In short, this code is the administrative code that controls the information to be sent out and the information coming from outside. Figure 4.1 shows the operation of the Monte Carlo method.

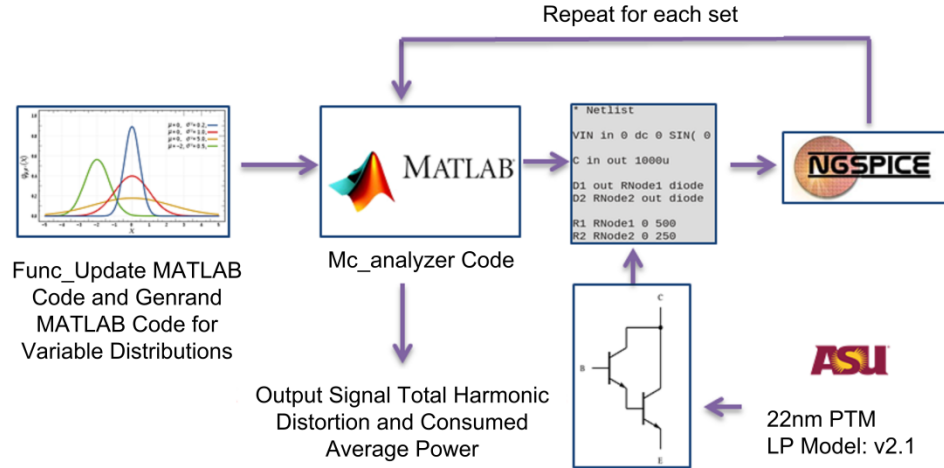


Figure 4.1: Monte Carlo analysis diagram for Class D circuit [39].

Firstly, component parameters that will be used as random variables are determined. 13 parameters for Class D circuit are used in the system. Random values for these 13 variables are assigned with the MATLAB code called genrand. The distribution characteristics of random variables were determined elsewhere. Each of these 13 parameters is assigned 50000 random values. 50000 different values assigned to each 13 variable were divided into 100 groups and 500 chunks were created in total. When the simulation starts, the analyzer code receives a set of 100 different values of the 13 variables in a chunk and sends it to the update_classd code one by one. The update_classd code writes this set of 13 values to the corresponding file to be sent in the netlist. The written file is received by the analyzer code and sent to the simulation.

At the end of the simulation, the information about the power and total harmonic distortion (THD) of interest is obtained by the analyzer code and saved to a specific file. This saved data is then sent to a different code to perform a probability density function (PDF) study on THD and power. This PDF-generating code, using the power and THD information obtained by the analyzer code extracts a PDF graph of them. This process is one of the 100 processes in a chunk. When 500 chunks are finished in total, the operation is completed. In short, 50000 values for 13 variables were not evaluated at one time and divided into 100x500.

There are two reasons why it is not performed at once. The first is that it is not desirable to disrupt the simulation due to unwanted external factors. The second

reason is to save time by transferring these 100s to other computers. Four computers were used to reduce simulation time. One of these four computers was divided into 4 virtual parts. Each of the other two computers was divided into two virtual parts. In this way, the simulation was made by 8 different computers. The disadvantage here is that the linux (opensuse) environment used during the work is virtual running with the virtual box program. The Linux environment running with the virtual drive is a waste of time because it cannot fully use the resources of the computer. If the computer had a fully installed linux system, the simulation would end in a shorter time.

More detailed information will be given about assigning different values to 13 variables and sending these assigned values to the appropriate locations in update_classd code.

Table 4.1 shows the component parameters that are selected as variables for the simulation and how these variables take values.

Table. 4.1: Random parameters configuration of Class D amplifier circuit.

%Random parameters of Class-D Amplifier circuit					
%LIMIT_L/R=3 for TRUN					
%PAR	TYPE	Ogpc.a	Ogpc.b	Arg.a	Arg.b
W_MN1	TRUN	2	0.2		
W_MN2	TRUN	2	0.2		
W_MN3	TRUN	2	0.2		
W_MN4	TRUN	2	0.2		
W_MP1	BETA	12	1.2	2	5
W_MP2	BETA	12	1.2	2	5
W_MP3	BETA	12	1.2	2	5
W_MP4	BETA	12	1.2	2	5
L1	UNIF	4.7×10^{-6}	4.7×10^{-7}		
C1	TRAN	1×10^{-6}	1×10^{-7}		
L2	BETA	4.7×10^{-6}	4.7×10^{-7}	2	5
C2	UNIF	4.7×10^{-7}	4.7×10^{-8}		
VAL_CAP	TRUN	6×10^{-10}	6×10^{-11}		

In Table 4.1, the column named par is the names of the component parameters selected as variables. The Type column contains the distribution type names used to assign a random value to the variables. In Ogpc.a column, the nominal values of the parameters used as variable are given. In Ogpc.b column, limits are given for variations of these nominal values. Change amount is 10% of nominal values. Arg.a and Arg.b columns are the distribution setting that should be determined for Beta distribution. The values 2 and 5 were used here.

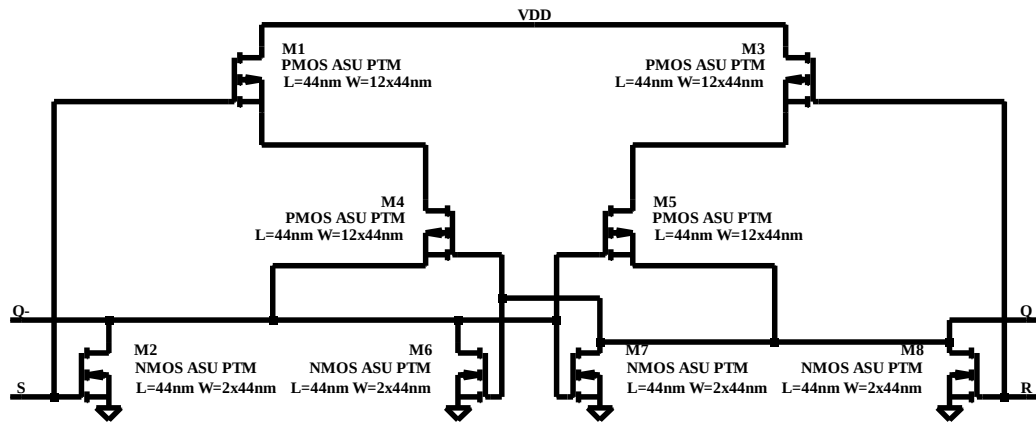


Figure 4.2: CMOS RS latch circuit used for Class D amplifier.

Figure 4.2 shows the circuit that is used instead of the flip flop circuit shown in Figure 3.94 and which RS latch the task. The netlist of the CMOS RS latch circuit in Figure 4.2 is prepared and added to the netlist of the Class D circuit. This RS latch consists of 4 NMOS and 4 PMOS MOSFET. And the length and width information of these MOSFETs are also shown in Figure 4.2.

8 of the 13 parameters selected as random variables constitute the channel widths of the MOSFETs in Figure 4.2. The parameter column in Table 4.1 shows the parameters W_MN1, W_MN2, W_MN3, W_MN4, W_MP1, W_MP2, W_MP3, W_MP4. These parameters are the channel widths of the rs latch MOSFETs used for Class D. Parameter L1, C1, L2, C2 and VAL_CAP are given in the other column.

The names L1, C1, L2, C2 from these parameters represent the low pass filter elements of the Class D circuit shown in Figure 3.95. The values of these elements are specified as random variables. Finally, the VAL_CAP parameter is the timing capacitor of the triangular wave generator section shown in Figure 3.92. The nominal

value of this component with C5 designator in the circuit diagram is 600pF in netlist. In this way, 13 parameters were transferred.

In the second column, it is stated that the parameters determined as random variables will take values according to which type of distribution. Truncated, beta and uniform distribution types were selected as distribution type. The data below the third column shows the nominal values of the relevant parameters. 2 and 12 values are used as coefficients. In fact, the nominal values for the width can be found by multiplying the numbers 2 and 12 by 44nm. When performing the simulation, the values 2 and 12 are multiplied by 44nm. Nominal values of other capacity and coil parameters are given in Table 4.1 is exactly as it looks.

In column 4, ogpc.b, the column shows which ranges of these nominal values will vary. As can be seen, the upper and lower limit of variation is stated as 10% of the nominal values. The arg.a and arg.b values in the last two columns are the configuration values of the beta distribution. Alpha and beta values should be indicated when beta distribution is performed. Therefore, these values are added.

As a result, thanks to this information in Table 4.1, the MATLAB code is able to understand random variables and how to assign random values to those variables. As mentioned previously, the genrand code generates 50000 values for each 13 parameters in accordance with this information. This 50000, created for 13 variables, has been divided into 500 chunk units, each with a set of 100 13 pack which chunk will be simulated is determined by the analyzer code. Update code is responsible for transferring these variable values to the appropriate places of the netlist.

At the end of the 50000 simulations performed, the THD of the output signal and the power information of the circuit were calculated. These calculations were then transferred to the other software. In this software, the probability distribution function of THD and power information was calculated. The PDF results obtained are shown in Figure 4.3 below.

For better understanding of the graph, the logarithmic version of the output was also shared. Information about random variables and their distribution for the Buck converter circuit has also been prepared. This information appears in the Table 4.2.

Table. 4.2: Random parameters configuration of Buck converter circuit.

%Random parameters of Buck converter circuit					
%LIMIT_L/R=3 for TRUN					
%PAR	TYPE	Ogpc.a	Ogpc.b	Arg.a	Arg.b
W_MN1	UNIF	2	0.2		
W_MN2	UNIF	2	0.2		
W_MN3	UNIF	2	0.2		
W_MN4	UNIF	2	0.2		
W_MP1	TRUN	12	1.2		
W_MP2	TRUN	12	1.2		
W_MP3	TRUN	12	1.2		
W_MP4	TRUN	12	1.2		
L1	BETA	5×10^{-5}	2×10^{-6}	2	5
C1	UNIF	2.5×10^{-5}	2.5×10^{-6}		
RL	TRUN	10	1		
IS(D1)	BETA	29.5×10^{-9}	29.5×10^{-10}	2	5

The first eight of the parameters in the Table 4.2 refer to the channel widths of PMOS and NMOS used for RS latch. Unlike Figure 3.4, RS latch was added to the netlist for analysis. These MOSFETs are therefore located here. The reason that the nominal values of the channel widths of the PMOS elements are chosen to be large is to operate the circuit at the desired accuracy level. The coil, capacitor and resistors in the last lines are the filter element. The last parameter is the I_s parameter, which is the diode's saturation current. For these 12 parameters, distribution types and variation limits are expressed. With this information, preliminary study has been performed for the analysis of buck circuit with 50000 repetitive Monte Carlo study. But, as in Class D, the analysis results were not shared.

In (Figure 4.3) the probability distribution function for the total harmonic distortion in the output signal is shown. Before this PDF result was obtained, the values of the low pass filter components, the channel lengths of the RS latch MOSFETs and the value of the timing capacitor of the triangle wave generator of the Class D power amplifier circuit were determined as random variables.

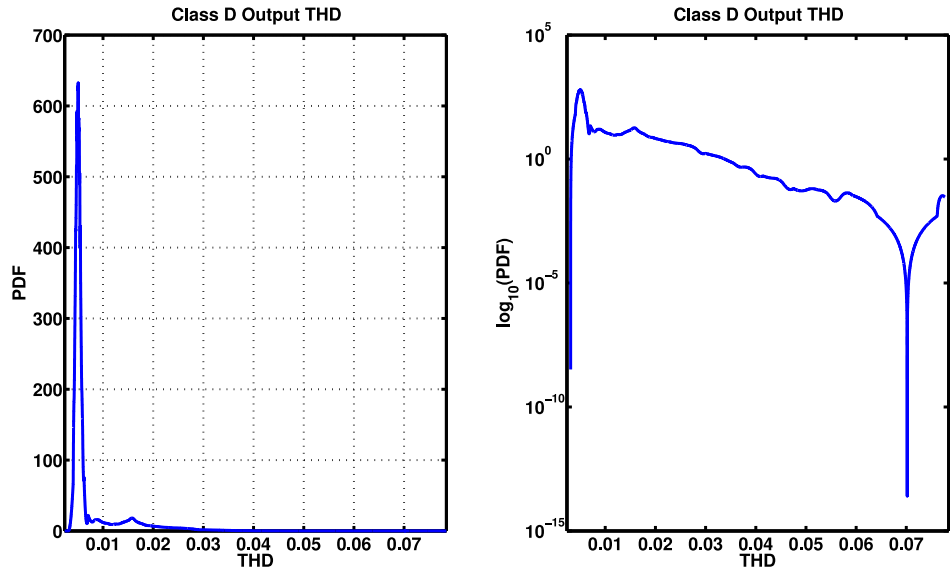


Figure 4.3: PDF for total harmonic distortion of Class D output signal.

As a result of variations of these parameters of the Class D power amplifier circuit, the probability distribution functions for the average power consumption (Figure 4.4).

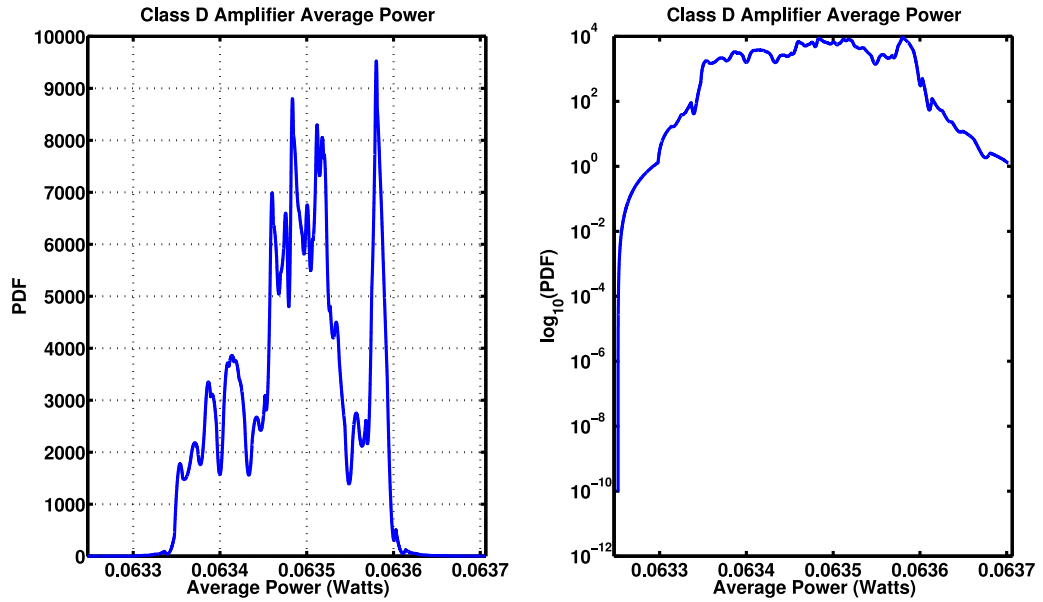


Figure 4.4: PDF for total harmonic distortion of Class D average power.

4.2. Results of gPC Method

gPC application was applied to dynamic NAND. Firstly, the parameters selected as random variables are the channel lengths of all MOSFET used in circuit. The distribution properties for these channel lengths are shown in Table 4.3 below.

Table. 4.3: Random parameters configuration of dynamic NAND gate circuit.

%Random parameters of Dynamic NAND gate circuit					
%LIMIT_L/R=3 for TRUN					
%PAR	TYPE	Ogpc.a	Ogpc.b	Arg.a	Arg.b
W_MNe	TRAN	88×10^{-9}	88×10^{-10}		
W_MPp	TRAN	132×10^{-9}	264×10^{-10}		
W_MN1	BETA	88×10^{-9}	88×10^{-10}	2	5
W_MN2	UNIF	88×10^{-9}	176×10^{-10}		

The W_MPp parameter in Table 4.3 is the width of the precharge PMOS first MOSFET in the circuit in Figure 3.9. The W_MNe parameter is the channel width of the NMOS MOSFET, which is the last in the circuit diagram. The parameters in the last lines are the parameters belonging to the MOSFETs in the middle of the circuit. Truncated, beta and uniform distribution types are selected as type of distribution. Distribution variation limits were 10%, 20%, 10% and 10%, respectively. It is seen that 2 and 5 values are given for Beta distribution.

Most of the codes used in the Monte Carlo application are also used in the gPC method. gPC implementation is started by determining the variables and their distribution characteristics.

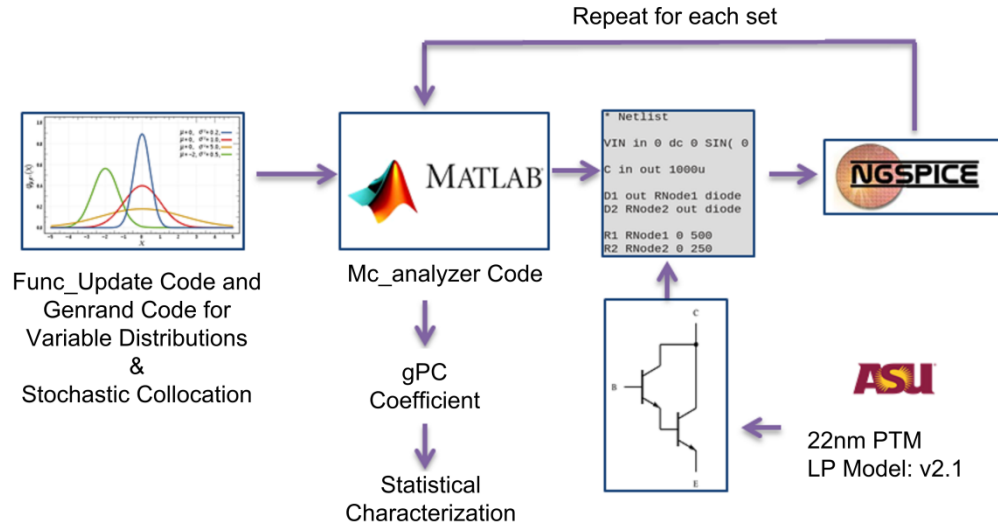


Figure 4.5: The flow diagram of the gPC application.

Figure 4.5 represents gPC application. Firstly, 4 random variables were determined as shown in Table 4.3. These variables were assigned random variables in accordance with the definitions in Table 4.3 with the genrand code. For each of the 4 parameters, 81 random variables were assigned. The number of random values assigned to each variable in the Monte Carlo application was 50000. However, 81 random variables are assigned here. This creates 81 sets of values, each with four values. In the gPC method, each of these data sets is called the collocation node. The number 81 is obtained according to a formula. Collocation node number is calculated using the number of random variables and the degree of polynomial used for the gPC method. First, one is added to the amount of polynomial degree and three are obtained.

As a result of 3 to the power 4 (random variable number), 81 is obtained. The necessity of using 81 data sets is determined in this way. When the application starts, the analyzer code 81 sends the data set individually to the update_classd code. The Update_classd code prepares this data set for the purpose of using it in the netlist of the circuit. The prepared file is received by the analyzer code and sent to the simulation. The simulation is performed to obtain information about charge sharing on the C_L and C_A represented in Figure 3.9 in the Dynamic NAND circuit. After 81 simulations are completed, the voltages of the ndp and ns1 nodes shown in Figure 3.7 are evaluated. As a result, a further post worker code is used to obtain 81 time data. These 81 times information keeps the time values when the voltage between the

two nodes is 1 mV. After these 81 time values gPC coefficients were obtained. The polynomials with the gPC coefficients were multiplied and another vector that converged to the result vector was obtained. Then, with the gPC details, base results, weight and node results and more repetitive simulation results were obtained and more than 81 results were obtained. These results were then sent to the PDF tool and the following Figure 4.6 graph was obtained. According to the PDF results in Figure 4.6, the standard deviation is 4.6×10^{-6} and the mean is 1.98×10^{-5} .

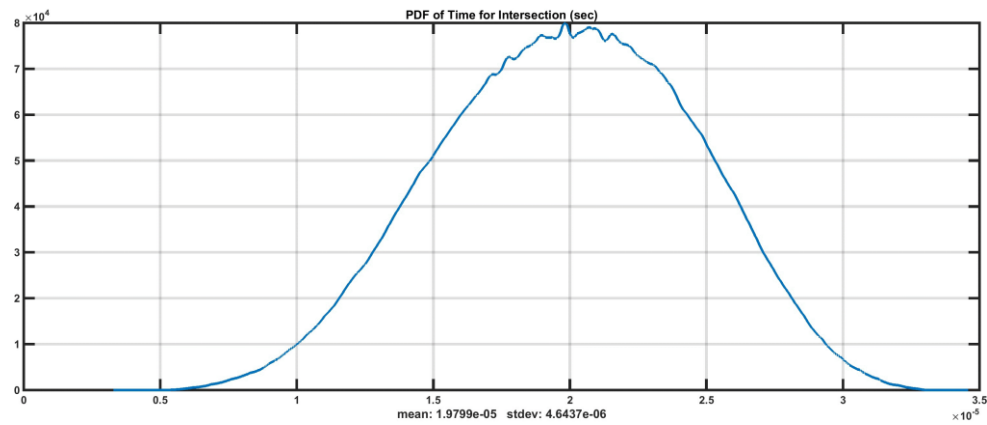


Figure 4.6: PDF of time for intersection on dynamic NAND simulation.

5. CONCLUSION AND FUTURE WORK

In the scope of the study, firstly gPC and Monte Carlo methods were investigated. The history of the Monte Carlo method and its developmental stages were mentioned. Then, information was given about the studies about gPC method. Issues such as gPC coefficients, Stochastic Galerkin were shared.

In the second part of the study, the necessary literature study related to the subject of probability was added to the study. Necessary information about permutation and combination was mentioned. Then, random variables, PDF and CDF titles were included. After that, some information about discrete and continuous distribution types were obtained and explained within the scope of the study. At the end of the second chapter, a sample of Monte Carlo was shared. Technical information was given about the gPC method.

In the third chapter of the study, technical information and application results for some digital and analog circuits were shared. In this section, Class D power amplifier, Buck, dynamic NAND, Gilbert cell, transmission gate chain, inverter chain circuits were included. For these 6 circuits, firstly, the literature study was performed and each of these information was included in the study. Then circuit diagrams and net lists of these circuits were created with suitable circuit element models. After the creation of circuit netlists, the circuits were simulated and the characteristic results of the circuit were shared.

In the simulation process, netlist is simulated with NGSPICE tool. Simulation outputs were taken by OCTAVE and MATLAB. In this way, the results are more visual and understandable. To run NGSPICE, OCTAVE and MATLAB tools synchronously, some software has been created. As a result, NGSPICE tool is not obliged to get the results and the application has become more flexible and rich with MATLAB and Octave programs. For example, the schematic of the Buck converter circuit is first drawn by the LTSPICE tool. Then the netlist of this circuit is created as a spice .cir file. And this file has been passed to the NGSPICE tool via the OCTAVE program. The results were simulated and the results were evaluated again in the OCTAVE tool. Thus, the characteristic of the circuit is understood from the simulation result. The result graph is reached with the help of the OCTAVE program. The theoretical information is taken into account while creating other circuits and simulations are provided. In most of the circuits, the MOSFET model of the ASU

PTM 22nm technology has been used for simulation. The charge sharing problem was discussed in the Dynamic NAND gate circuit. ASU PTM 22nm model was used for the 4 MOSFETs in the circuit. The channel widths of the MOSFETs are set accordingly. The time value between the two output nodes of the circuit created in this way was 1mV. The result graph of the required signals was shared. In order to make uncertainty calculations on this circuit, a gPC study was conducted.

For this study, the channel widths of the circuit are selected as random variables. As a result, the results were shared in Chapter 4. Gilbert cell circuit of the other circuit related to the necessary literature study was made and this information was shared. The circuit is prepared as a mixer circuit. This circuit was created with 2N5089 transistors. 15k and 0.5k sine signals were applied as input signal. The output results of this gilbert cell mixer circuit were shared. The other circuit is the transmission gate chain circuit. It is used in most digital circuits and is one of the signal improvement tasks. 16 transmission gates were used for this chain circuit. The problem in this chain circuit is the delay time. In order to improve this delay, the buffer was added to the chain by using the theoretical information. Simulations were made without the buffer and the optimum buffer was added and the results were compared. A significant delay advantage has been achieved by adding the buffer.

The other circuit, the inverter chain, is an important structure for digital designs. An inverter gate consists of 1 PMOS and 1 NMOS element. ASU PTM 22nm model was used as a model in MOSFETs used in this circuit. In the simulation, single door work was performed before the chain installation was started. The optimum W_p/W_n ratio was obtained by using various channel width ratios in these single gates. The next circuits were created by considering this ratio. Then the input capacitance required to create the chain was found. In addition, chain circuits with different inverter numbers were simulated with the appropriate fanout sizing settings. The delay results of the circuits with 1, 2, 3, 4, 5 inverter gates were shared.

Another circuit in the thesis, the Class D amplifier circuit, is a power amplifier that amplifies the signal applied to the circuit at the output. The circuit first compares the triangle signal with the input signal. The signal passing through the switching unit is then raised in the output after the low pass filter. In the study, the triangular wave signal was obtained by using the internal structure of the 555 integral. For the MOSFETs used for RS latch within the 555 device, ASU PTM 22nm model was used. Input, output, pwm and triangle signals graphs of the circuit were shared. In

addition, some parameters of some components of Class D circuit have been selected as random variables for Monte Carlo study. These parameters are the timing capacitance of the triangular waveform generator, the values of the coils and capacitors in the low-pass filter section, and the 8 MOSFETs channel widths used for RS latch. 13 random variables were used in the analysis of Monte Carlo uncertainty study. These 13 random variables were assigned 50000 values with specific distribution types and 50000 simulations were made with these assigned values.

As a result of the 50000 simulation, total harmonic distortions of the output signal of the circuit and the input power of the circuit have been calculated. Then, the PDF of these results was calculated.

In Chapter 4, the Monte Carlo method applied for the Class D circuit, the gPC method applied for the dynamic NAND and the results obtained were shared. This section discusses how these applications have been made and which codes have been executed. For Class D, a PDF result is given for the amount of THD in the output signal and for the amount of power spent. For the dynamic NAND, the uncertainty at the moment of the charge sharing of the two nodes was investigated in the gPC result. The PDF result of the time values at the moment when the charge sharing stopped was shared.

In gPC method, polynomial degree 2 was selected, four circuit parameters were chosen as random variables and 81 collocation nodes were formed. In Class D circuit, 13 parameters are selected as random variables. These variables are assigned 50000 different values with certain distribution types. As a result of the simulations made with these values, the uncertainty of the harmonic distortion of the output signal and the spent power value was shared.

In the future, when doing Monte Carlo work, instead of 50000 repeats, the results can be evaluated with more repetitions. Of course, both gPC and Monte Carlo studies on the same circuit can be compared and the two methods can be compared. The superiorities and weaknesses of these two methods can be listed. For all circuits used in the study, it is useful to do both gPC and Monte Carlo applications.

REFERENCES

- [1] Fishman G., (2003), “Monte Carlo: concepts, algorithms, and applications”, 1st Edition, Springer Science & Business Media.
- [2] Ulam S.M., (1991), “Adventures of a Mathematician”, 1st Edition, University of California Press.
- [3] Eckhardt R., (1987), “Stan ulam, john von neumann, and the monte carlo method”, Los Alamos Science, 15, 131-136.
- [4] Metropolis N. Ulam S., (1949), “The Monte Carlo Method”, Journal of the American Statistical Association, 44 (247), 335-341.
- [5] Herken G., (2014), “The Winning Weapon: The Atomic Bomb in the Cold War, 1945-1950”, 1st Edition, Princeton University Press.
- [6] Stimson H.L. Truman H.S., (1947), “The decision to use the atomic bomb”, Bulletin of the Atomic Scientists, 3 (2), 37-67.
- [7] Xiu D. Karniadakis G.E., (2002), “The Wiener-Askey polynomial chaos for stochastic differential equations”, SIAM Journal on Scientific Computing, 24 (2), 619-644.
- [8] Vruthula S., Wang J.M., Ghanta P., (2006), “Hermite polynomial based interconnect analysis in the presence of process variations”, IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, 25 (10), 2001-2011.
- [9] Stievano I., S Manfredi, P. Canavero F.G., (2011), “Stochastic analysis of multiconductor cables and interconnects”, IEEE Transactions on Electromagnetic Compatibility, 53 (2), 501-507.
- [10] Spina D., Ferranti F., Dhaene T., Knockaert L., Antonini G., and Ginste D.V., (2012), “Variability analysis of multiport systems via polynomial-chaos expansion”, IEEE Transactions on Microwave Theory and Techniques, 60 (8), 2329-2338.
- [11] Rufuie M.R., Gad E., Nakhla M., Achar R., (2014), “Generalized hermite polynomial chaos for variability analysis of macromodels embedded in nonlinear circuits”, IEEE Transactions on Components, Packaging and Manufacturing Technology, 4 (4), 673-684.
- [12] Biondi A., Ginste D.V., Zutter D.D., Manfredi P., Canavero F.G., (2013), “Variability analysis of interconnects terminated by general nonlinear loads”, IEEE Transactions on Components, Packaging and Manufacturing Technology, 3 (7), 1244-1251.

- [13] Tao J., Zeng X., Cai W., Su Y., Zhou D., Chiang C. (2007), "Stochastic sparse-grid collocation algorithm (SSCA) for periodic steady-state analysis of nonlinear system with process variations", Conference on Asia and South Pacific Design Automation, 474-479, Yokohama, Japan, 23-26 January.
- [14] Zhang Z., El-Moselhy T.A., Elfadel I.M., Daniel L. (2013), "Stochastic testing method for transistor-level uncertainty quantification based on generalized polynomial chaos", IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, 32 (10), 1533-1545.
- [15] Sandu A., Sandu C., Ahmadian M., (2006), "Modeling multibody systems with uncertainties. Part I: Theoretical and computational aspects", Multibody System Dynamics, 15 (4), 369-391.
- [16] Pham T.A., Emad G., Michel N., Ramachanda A., (2013), "Efficient hermite-based variability analysis using approximate decoupling technique", 17th IEEE Workshop on Signal and Power Integrity, 1-4, Paris, France, 12-15 May.
- [17] Rufuie M.R., Gad E., Nakhla M.S., Achar R., (2015), "Fast variability analysis of general nonlinear circuits using decoupled polynomial chaos", IEEE Transactions on Components, Packaging and Manufacturing Technology, 5 (12), 1860-1871.
- [18] Reddy T.A., (2011), "Applied Data Analysis and Modeling for energy Engineers and Scientist", 1st Edition, Springer Science & Business Media.
- [19] Wiener N., (1938), "The homogenous chaos", American Journal of Mathematics, 60 (4), 897-936.
- [20] Cameron R.H. Martin W.T., (1947), "The orthogonal development of non-linear functionals in series of Fourier-Hermite functionals", Annals of Mathematics, 385-392.
- [21] Ghanem R.G. Spanos P.D., (1991), "Stochastic finite element method: Response statistics in Stochastic finite elements", 1st Edition, Springer Science & Business Media.
- [22] Knio O.M., Najm H.N., Ghanem R.G., (2001), "A stochastic projection method for fluid flow: I. basic formulation", Journal of Computational Physics, 173 (2), 481-511.
- [23] Le Maitre, O.P., Reagan M.T., Najm H.N., Ghanem R.G., and Knio O.M., (2002), "A stochastic projection method for fluid flow: II. Random process", Journal of Computational Physics, 181 (1), 9-44.
- [24] Xiu D. Karniadakis G.E., (2003), "Modeling uncertainty in flow simulations via generalized polynomial chaos", Journal of Computational Physics, 187 (1), 137-167.

- [25] Ernst O.G., Mugler A., Starkloff H.J., Ullmann E., (2012), "On the convergence of generalized polynomial chaos expansions", *ESAIM: Mathematical Modeling and Numerical Analysis*, 46 (2), 317-339.
- [26] Wan X. Karniadakis G.E., (2005), "An adaptive multi-element generalized polynomial chaos method for stochastic differential equations", *Journal of Computational Physics*, 209 (2), 617-642.
- [27] Wan X. Karniadakis G.E., (2006), "Long term behavior of polynomial chaos method in stochastic flow simulations", *Computer Methods in Applied Mechanics and Engineering*, 195 (41-43), 5582-5596.
- [28] Le Maitre O., Mathelin L., Knio O.M., Hussaini M.Y., (2010), "Asynchronous time integration for polynomial chaos expansion of uncertain periodic dynamics", *Discrete & Continuous Dynamics Systems-A*, 28 (1), 199-226.
- [29] Le Maitre, O. Knio O.M., (2010), "Spectral methods for uncertainty quantification with applications to computational fluid dynamics", 1st Edition, Springer Science & Business Media.
- [30] Sapsis T.P. Lermusiaux P.F., (2009), "Dynamically orthogonal field equations for continuous stochastic dynamical systems", *Physica D: Nonlinear Phenomena*, 238 (23-24), 2347-2360.
- [31] Gerritsma M., van der Steen J.B., Vos P., and Karniadakis G., (2010), "Time-depended generalized polynomial chaos", *Journal of Computational Physics*, 229 (22), 8333-8363.
- [32] Heuveline V. Schick M., (2011), "Towards a hybrid numerical method using generalized polynomial chaos for stochastic differential equations", 3th Edition, Preprint Series of the Engineering Mathematics and Computing Lab.
- [33] Schick M., (2011), "Uncertainty quantification for stochastic dynamical systems: Spectral methods using generalized polynomial chaos", PhD Thesis, Karlsruher Institut für Technologie.
- [34] Dattoli G., Srivastava H., Zhukovsky K., (2005), "Orthogonality properties of the Hermite and related polynomials", *Journal of Computational and Applied Mathematics*, 182 (1), 165-172.
- [35] Web 1, (2018) <http://bwrc.eecs.berkeley.edu/IcBook/index.htm>, (Accessdate: 20/12/2018).
- [36] Gray P.R. Meyer R.G., (1984), "Analysis and Design of Analog Integrated Circuits", 2nd Edition, John Wiley & Sons.

- [37] Rabaey J.M., Chandrakasan A.P., Nikolic B., (2002), “Digital Integrated Circuits”, 2nd Edition, Englewood Cliffs: Prentice Hall.
- [38] Web 2, (2018) https://www.electronics-tutorials.ws/rc/rc_1.html, (Accessdate: 25/12/2018).
- [39] Duman M.E. Şuvak Ö., (2017), “C tabanlı elektronik devre benzetim programının MATLAB aracılığıyla kontrol edilmesi ve eşit olmayan zaman aralıklarında örneklemiş benzetim verilerinin frekans analizi için numerik bir FFT yöntemi”, GTU Lisansüstü Araştırmalar Sempozyumu & Tanıtım Günleri, 53, Kocaeli, Türkiye, 17-18 May.

BIOGRAPHY

İsmail Sonalcan was born in Adapazarı, Sakarya, in 1985. He received the B.Sc. degree in Electric and Electronics Engineering from Eskişehir Anatolian University, in 2010. He has served as a designer for a number of precious projects with valuable teammates for 7 years. He works project basis and design medical product. He is married and has a son.