STUDY OF INTERCONNECT VARIATION ON CIRCUIT PERFORMANCE

by

Zhihao Jeff Lin

Memorandum No. UCB/ERL M97/80

17 October 1997

STUDY OF INTERCONNECT VARIATION ON CIRCUIT PERFORMANCE

by

Zhihao Jeff Lin

Memorandum No. UCB/ERL M97/80

17 October 1997

ELECTRONICS RESEARCH LABORATORY

College of Engineering
University of California, Berkeley
94720

Abstract

Study of Interconnect Variation on Circuit Performance

by
Zhihao Jeff Lin

Electrical Engineering and Computer Sciences
University of California at Berkeley

Professor Costas J. Spanos, Advisor

Deep submicron technology makes interconnect one of the main factors determining the circuit performance. Previous work shows that interconnect parameters exhibit a significant amount of spatial variation. In this work, we developed approaches to study the influence of the interconnect variation on circuit performance and to evaluate the circuit sensitivity to interconnect parameters. First, an accurate interconnect modeling technique is presented, and an interconnect model library is developed. Then, we explore an approach using parameterized interconnect models to study circuit sensitivity via a ring oscillator circuit. Finally, we present another approach using statistical experimental design techniques to study the sensitivity of a large and complicated circuit to interconnect variations.

Table of Contents

List of Figures

List of Tables

Acknowledgments

First, I would like to express my most heartfelt gratitude and appreciation to my research advisor, Professor Costas J. Spanos, for his encouragement, guidance and lasting support in every way. I am grateful of my study experience here in the Berkeley Computer-Aided Manufacturing (BCAM) group. I would also like to thank Professor Andrew Neureuther for his interest in this work and serving as the second reader of this thesis.

I would like to give special thanks to Dr. Linda Milor for her guidance and help throughout this work.

Thanks also give to Dr. Jue-Hsien Chern and TMA Inc. for supplying the interconnect extraction software Raphael.

A warm "thanks" goes out to the past and present members of the Berkeley Computer-Aided Manufacturing (BCAM) group who helped make the experience worthwhile: Roawen Chen, Anna Ison, Nickhil Jakatdar, Greg Luurtesema, Xinhui Niu, Manolis Terrovitis, Haolin Zhang, and Dongwu Zhao.

This work was supported by the Semiconductor Research Corporation (SRC), the state of California Micro program, and participating companies (Advanced Micro Devices, Atmel Corporation, Lam Research, National Semiconductor, Silicon Valley Group, Texas Instruments and AME.).

Chapter 1 Introduction

1.1. Background and Motivation

The continuously increasing scale of integration used in the design and processing of integrated circuits has drawn special attention toward interconnect effects. As the minimum feature size in VLSI systems drops to 0.25 micron and below, interconnect characteristics have become limiting factors on performance, since the time constant associated with interconnect is scaled by a smaller factor compared to those of devices. Future chip complexity and speed advances will depend on the ability to model the electrical behavior of interconnect in an accurate and efficient fashion.

Critical path delays in circuits depend upon the interconnect as well as on the device parameters. The effects of device parameter variations have been widely studied [12] [14] [15] [16]. However, these simulations currently do not take into account the effects of interconnect parameter variations. As a result, the yield estimation and circuit optimization based on these studies may not be able to provide accurate results in current and future technologies, where more and more significant portions of path delays will result from interconnect.

An IC process is a series of steps used to manufacture a semiconductor product which turns a bare silicon wafer into packaged ICs. These steps include the introduction and redistribution of impurities into the silicon, the growth or deposition of layers on the wafer and the patterning of these layers. Finally, the wafer is tested and the die on the wafer is separated and put into packages. Each step of this process requires specific materials and settings on the manufacturing equipment in order to make the circuit functional and meet the performance specifications to be sold by the manufacturer. Process parameters refer to

measurable quantities which characterize the results of these steps. These performance specifications will be referred to as the circuit performances.

However, as with any manufacturing process, there are uncontrolled variations in the process which cause product performance to vary. Furthermore, process or equipment variability is not necessarily falling at the same pace as the minimum feature size. This is because while decreasing nominal critical dimensions, processing technologies are being pushed to the limit. So, as device and interconnect dimensions continue to shrink and wafer and die size increase, process uniformity and consistency become a big concern. In [2], it is shown that both layout and process parameters exhibit a significant amount of spatial variation. Most of this variation is deterministic, so there is hope to compensate for it using proper modeling and design techniques.

With current technology, the impact of interconnect parameter variations on signal delays may already be quite significant. Thus, it becomes necessary to comprehend and anticipate the effects of interconnect parameter variation in the design process. Specifically, a methodology to asses the impact of radom and systematic variations in interconnect parameters to circuit performance must be developed.

A modeling framework to study the sensitivity of circuit performance to interconnect parameter variations will allow circuit designers to meet timing targets while taking into account the random and systematic source of interconnect parameter variations. It will also help the process designers to design new technologies while taking the sensitivity information into consideration. Finally, the sensitivity study results will help make the circuit more robust against the variation.

The success of the IC industry has in part been due to the use of a design style, first formalized by Mead and Conway [11], which isolates IC designers from detailed consideration of the technology and the manufacturing process. However, the increasing cost and complexity of a modern IC manufacturing line are necessitating increased interaction among design, manufacturing and technology to deliver profitable products in a timely fashion. Design for Manufacturability (DFM) techniques strive to impact the design and manufacture of an IC product in light of a specific technology and manufacturing processes in order to improve the manufacturability of the product. Indeed, the term "Manufactura-

bility" implies the integration of process design, circuit design and manufacturing activities.

The focus of DFM has traditionally been on yield prediction and optimization. The yield of an IC is the fraction of manufactured parts which are functional and meet the specifications. Yield can be further decomposed into the fractions of parts sold at different performance levels, reflecting the distribution of product performance caused by process variation. Previous work in DFM analyzed the distribution of circuit performance, and optimized the circuit design or fabrication process for maximum yield [18] [19] [20]. This is also the underlying motivation for this work.

Also, DFM activity, and in particular enhanced interaction between process, design and manufacturing, can decrease the time required to reach high yield levels, or improve the learning curve. This results into significant and competitive advantages.

Overall, the goal of this thesis is to address the problem of interconnect variation, look for a methodology to model interconnect wires, and develop DFM approaches to quantify and investigate interconnect parameter variations on circuit performance under current and future technologies. The ultimate objective is to facilitate optimal circuit and process design, reduce time-to-yield, and improve the final yield.

1.2. Thesis Overview

Two approaches to study the circuit sensitivity to interconnect parameter variations are developed in this thesis. The first approach is based on a parameterized interconnect model library. The parameterized interconnect models allow us to manipulate interconnect parameters, and to generate a circuit description that is suitable for performance sensitivity study. The second approach uses statistical experimental design techniques to analyze complicated circuits via simulation experiments. The first approach is illustrated with the help of ring oscillation circuits, and the second approach is illustrated on a large multiplier circuit.

1.3. Thesis Organization

Chapter 2 introduces some background on interconnect modeling techniques, and discusses the related issues of interconnect modeling for the purpose of sensitivity study.

A methodology to model interconnect wires and generate the model library was developed based on numerical simulation of interconnect structures. With the models developed in Chapter 2, Chapter 3 shows how a parameterized circuit description can be generated and used for statistical circuit performance simulation. The purpose of the sensitivity study and its related aspects are discussed in detail. The results from the case study of a ring oscillator are presented. In Chapter 4, the advantages and disadvantages of the approach developed in Chapter 3 are discussed and a complementary approach using statistical design techniques is proposed, which was explored using a shift-and-add multiplier. Chapter 5 concludes the thesis with a summary and suggestions for future work.

Chapter 2 Interconnect Modeling

2.1. Introduction

The design and development of next-generation electronic products are driven by an increasing demand for greater functionality, higher performance, and shorter design-to-manufacturing cycle time. So the integrated circuit (IC) design trend is toward reduced line widths, larger die size, greater number of interconnect layers and higher clock frequencies. As a consequence, the electrical characteristics of the interconnections are becoming important factors in the behavior of integrated circuits. By mid 1997, the feature size was as small as 0.25 micron and it has been predicted that it would decrease to 0.18 micron soon afterwards. Shrinking silicon geometries affect the electrical properties of the interconnect, and this has a corresponding effect on the IC performance. As a result, factors which have an insignificant effect when the feature size was at one micron or larger, become significant impediments to performance at 0.25 micron and below.

Because transistor sizes are shrinking faster than interconnect distances between transistors, wiring delays dominate the total gate to gate delay. This factor is getting more pronounced as technologies change. For instance, at 2 micron, 80% of the delay is due to transistor or gate delay, and only 20% of the delay is attributed to the wires, that is, the speed of a chip was largely determined by the inherent capacitance of the transistors on the chip, while the effect of interconnections was relatively unimportant. In deep submicron designs, however, interconnect delay may account for 80% to 90% of the total delay for some very high performance circuits, and it plays an ever greater role in the timing of IC chips. Therefore, one must be able to assess the impact of interconnections on signal delay in order to accurately evaluate the timing of current and future integrated circuits.

In order to understand and account for interconnect effects in the design process, it is necessary to model the interconnection and extract its parasitic parameters. It is essential that the electrical behavior of interconnect is modeled accurately. The accuracy of interconnect models is the very basis of achieving meaningful predictions of circuit behavior and obtain reliable sensitivity evaluations. The models should also be suitable for statistical circuit simulation and sensitivity analysis, which is the purpose of this work.

One approach to interconnect modeling is to construct an equivalent electrical circuit representation. The equivalent circuits that represent the interconnections can be combined with the equivalent circuits that describe the active devices, and the behavior of the entire circuit can be analyzed with a circuit simulator such as HSPICE.

There are two steps to the process of constructing an equivalent circuit model for an interconnect wire. The first step is to determine the nature of the equivalent circuit, that is, what kinds of circuit elements are important, and how many degrees of freedom are required for the level of accuracy desired. A simple example of such an equivalent circuit is shown in Figure 2.1. The second step is to determine the value associated with each element in the equivalent circuit.

2.2. Background

2.2.1. Interconnect Trends

2.2.1.1. Scaling Effects on Interconnect

At maximum wiring density, each wire is capacitively coupled to its nearest neighboring wires on the same layer, as well as wires above and below it, as shown in Figure 2.2. As device sizes have been scaled for improved performance and increased density, the interconnect sizing, spacing, and conductor thickness have been reduced as well.

If all of the dimensions of interconnect are scaled in a complete die shrink, then the total RC for the interconnects would remain unchanged. However, as device sizes are

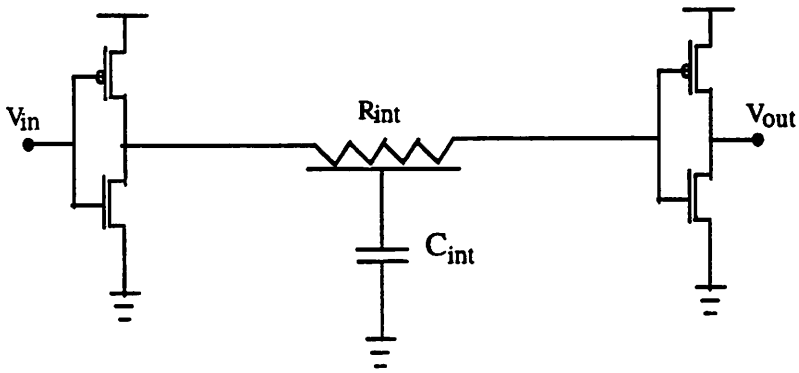


Figure 2.1.
A simple interconnect model

reduced, the trend is also to use larger die. Therefore, the average interconnect lengths not only do not scale, but also become longer from one generation to the next.

In a simplified form, the 50 percent delay[1] from the first to second gate in Figure 2.1 can be expressed as

$$t_{50\%} = R_{tr} \bullet C_{gate} + R_{int} \bullet C_{int} \tag{2.1}$$

where R_{tr} is the effective on-resistance of the driving gate, C_{gate} is the input capacitance the receiving gate, and R_{int} and C_{int} are the distributed interconnection resistance and capacitance, respectively.

Let S be the scaling factor of the minimum feature size, and S_c be the scaling factor of the chip size. We define S to be the linear reduction factor and S_c to be the linear magnification factor. In this way, both are larger than one as technology progress. Table 2.1 shows how the device and interconnect parameters are scaled in the situation of ideal scaling and quasi-ideal scaling, respectively. Ideal scaling is proposed by Dennard [22] in 1974. In quasi-

1. Fifty percent delay is defined as the delay from the time when the input potential reaches the midpoint between Vdd and ground to the time when the output reaches the same midpoint.

ideal scaling, the vertical dimensions of interconnect are scaled by approximately the square root of the lateral scaling factor S.

From Table 2.1, the scaling properties of the delay components for a critical path that extends across a chip can be found. For the case of ideal scaling, they are listed below:

$$R_{tr} = \frac{1}{\frac{W}{L}\mu C_{gox}(V_{DD} - V_T)} \propto 1 \tag{2.2}$$

$$C_{gate} = \varepsilon_{ox}\frac{(W_n L_n + W_p L_p)}{T_{ox}} \propto 1/S \tag{2.3}$$

$$R_{int} = \rho\frac{l_{int}}{W_{int}H_{int}} \propto 1/S \tag{2.4}$$

$$C_{int} = \varepsilon_{ox}\frac{W_{int}l_{int}}{T_{ILD}} \propto S_C \tag{2.5}$$

Here W_n, W_p, L_n and L_p are the width and length of a n and p transistor, respectively, μ and C_{gate} are the surface mobility of the carriers and the gate oxide capacitance per unit area, V_{DD} and V_T are the power supply and threshold voltage, T_{ox} and T_{ILD} are the gate and field oxide thicknesses, respectively. H_{int} is interconnect thickness, and l_{int} and W_{int} are the length and width of an interconnection.

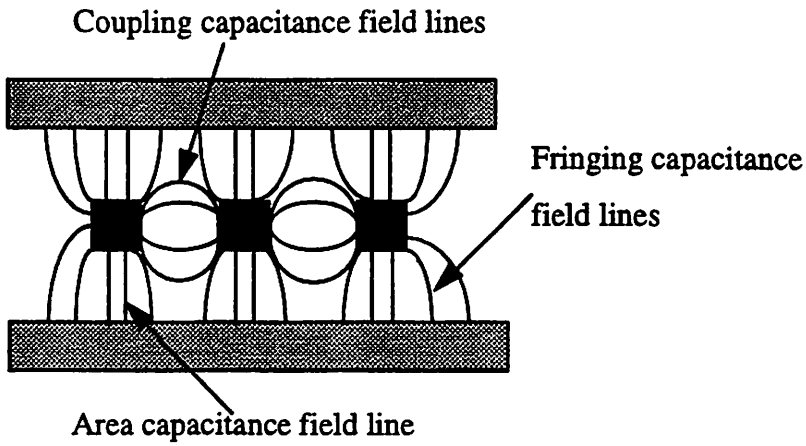


Figure 2.2.
Multi-level metal interconnect cross-section
wires are in orthogonal directions from layer to layer.

Parameter	Ideal Scaling Result	Quasi-ideal Scaling Result	Trend with $S>1$, $S_c>1$
W (Gate width)	W/S	W/S	↘
L (Gate length)	L/S	L/S	↘
T_{ox} (Gate oxide thickness)	T_{ox}/S	T_{ox}/S	↘
T_{ILD} (ILD thickness)	T_{ILD}/S	$T_{ILD}/S^{1/2}$	↘
l_{int} (Interconnect wire length)	$l_{int}\,S_c$	$l_{int}\,S_c$	↗
W_{int} (Interconnect wire width)	W_{int}/S	W_{int}/S	↘
H_{int} (Interconnect wire thickness)	H_{int}/S	$H_{int}/S^{1/2}$	↘

Table 2.1.
Scaling Properties of Device and Interconnect Parameters

Parameter	Ideal Scaling	Quasi-ideal Scaling	Trend with $S>1$ and $S_c>1$
R_{tr}	1	1	$\leftrightarrow$
C_g	1/S	1/S	$\searrow$
R_{int} (global)	S^2S_c	$S^{3/2}S_c$	$\nearrow$
C_{int} (global)	S_c	$S^{-1/2}S_c$	$\nearrow$
$R_{int}C_{int}$	$S^2S_c^2$	SS_c^2	$\nearrow$

Table 2.2.

Effects of Scaling on Interconnect

Table 2.2 shows the effects of scaling on FET transistors and interconnects. Given that the resistance of transistors stays the same for a given scale factor, the gate capacitance of the next generation is reduced by 1/S. The gate delay ($R_{tr}*C_{gate}$) is thus reduced by 1/S, so the device speed is improved, as expected. However, for global interconnect lines, the interconnect resistance R_{int} increases by S^2*S_c. S^2 comes into consideration because the wire thickness and width are reduced by 1/S. On the other hand, the interconnect capacitance is only increased by S_c, because the wire capacitance per unit length stays the same and wire length increases by S_c. The RC time constant of the global interconnect is thus increased by $S^2*S_c^2$.

Both S and S_c have approximately the same values for each new generation. Therefore, the delay of the global line is increased by S^4 with ideal scaling. In practice, ideal scaling is difficult to implement rigorously. Quasi-ideal scaling has been proposed and is followed more or less today by IC industry. Still, the delay of the global line is increased by S^3. This clearly illustrates why interconnect delay is so dramatically increased with scaling and how it becomes an important issue in submicron technology.

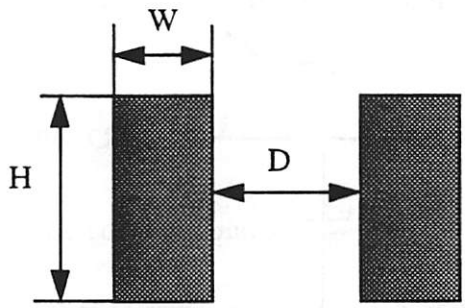


Figure 2.3.
Tighter pitch and higher height-to-width ratio of new technology

2.2.1.2. Coupling Capacitance and Its Effects

The interconnection capacitance has three components as indicated in Figure 2.2: the area component (also referred to as parallel plate capacitance component), the fringing field component, and the wire-to-wire capacitance component.

In current 0.25 micron technology, the conductor height-to-width ratio is about 1.4, and this ratio is expected to increase to 2.5 in the future, as illustrated in Figure 2.3. At the same time, conductor spacing is comparable to conductor thickness. Because of tighter pitch and the increasing conductor height-to-width ratio, the coupling capacitance between wires is becoming significant, and the fringing field is also responsible for a large portion of the overall capacitance. Combined fringing field and coupling capacitances are usually larger than the parallel plate component. To improve packing density while maintaining a relatively small interconnect RC constant, it is desirable to fix the conductor and oxide thickness, while reducing the wire width and spacing.

Assuming that the wiring layers in Figure 2.2 represent upper and lower level metal layers, it should be noted that the majority of the total capacitance will be between signal

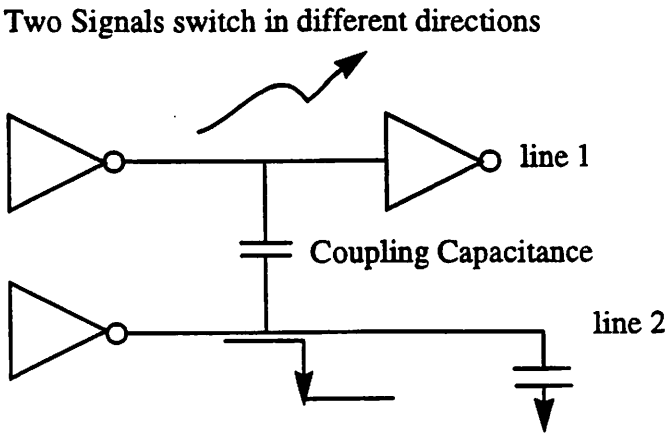


Figure 2.4.
The coupling capacitance effect

wires for multi-level technologies, with little coupling capacitance to the substrate. One has to be concerned with the coupling between signal wires, as it degrades the switching speed and causes cross talk between neighboring wires, which can result in faulty operation. For example, two coupled lines are illustrated in Figure 2.4. If one line is switching high, while the other is switching low, the waveform on line one may become non-monotonic, thus increasing the switching delay.

2.2.2. Constructing Equivalent Circuits to Interconnect Structures

2.2.2.1. The Nature of the Equivalent Circuit

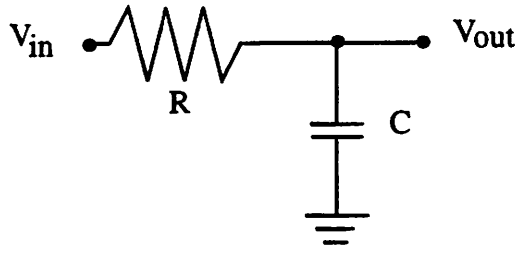
There is a wide range of equivalent circuits which can be used to model on-chip interconnections. The detailed character of the equivalent circuit for an on-chip interconnect depends on, among other things, the type of technology, the length of the interconnect wire, the amount of current on the interconnect, and the switching speed of the chip.

One fundamental quantity that characterizes the relative importance of interconnection capacitance and inductance for a particular technology is the lossless transmission line impedance level, $Z_g = \sqrt{L/C}$,where L and C are inductance and capacitance per unit length. [13] claims that the equivalent circuit for an average length on-chip interconnection in CMOS chips can be constructed using capacitors and resistors. In modern lossy on-chip interconnect, the inductive voltage drop is negligible compared to resistive voltage drop up at clock frequencies of 1-2 GHz. Thus, on-chip interconnect lines may be approximated by an RC line. So, throughout this thesis, all on-chip interconnects are modeled as RC networks.

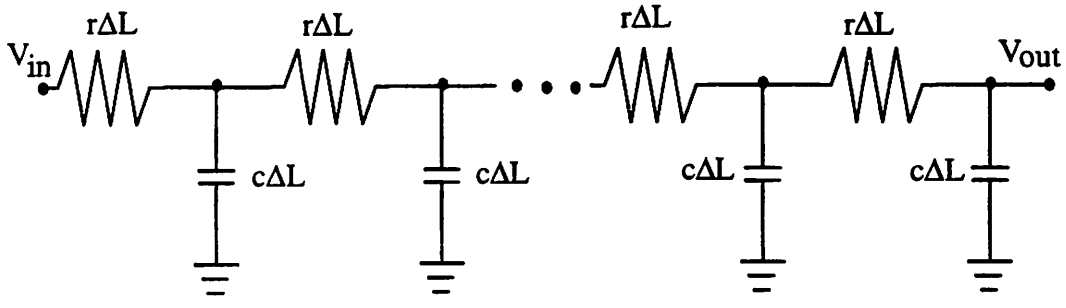
Because of the distributed nature of interconnect wires (see Figure 2.5(b)), once it has been determined what types of circuit elements are required to model a particular class of on-chip interconnections, one must then decide how many circuit elements of each type are needed. For example, it may be determined that the on-chip interconnections are to be modeled as RC ladder networks. Then, one must decide how many resistors and capacitors are needed to accurately model each piece of interconnection. The process of breaking each piece of interconnection into a fixed number of lumped circuit elements will be called the subdivision process. The sub-division into these partial elements is similar to the allocation of elements in a finite-element approach to the solution of partial differential equations. However, since one is typically interested in the overall delay, and not the exact shape of the waveform, the partial elements considered here can be assigned on a relatively coarse scale.

The problem of deciding upon a proper subdivision of interconnect wires is still one whose solution involves more art than science. It is obvious that if the interconnection is

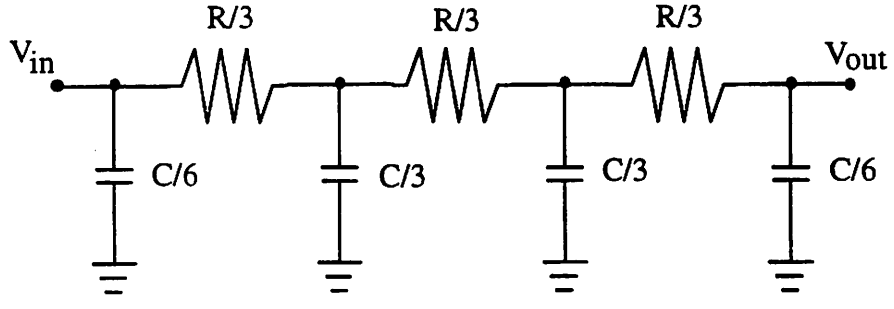
broken up into very many pieces, then the description will be more than adequate. It is equally obvious that if the interconnection is broken up into too few pieces, then the description obtained will be inadequate. In the former case, computational resources are wasted. In the latter case, the accuracy of the results will be questionable. Even if one has an appropriate subdivision scheme, it is difficult to determine its accuracy. Only on intuitive grounds can one make a priori arguments about what degree of subdivision is sufficient. These intuitive arguments can be based partly on the wavelength of propagating signals of the relevant frequency in whatever materials and geometry are under consideration. Portions of the interconnection that are well within one wavelength of each other can likely be treated as part of the same lumped element (Figure 2.5(a)). For lower speed circuits operated at relatively low frequencies (long wavelength), interconnect models are very simple. Each wire is much shorter than the wavelength, and can thus be treated as an individual lumped element. As the operating frequency (switching speed) of a circuit is increased, the wavelength decreases, requiring more distinct lumped elements. In practice, some simplified version of distributed model is used in simulation, such as the $\pi 3$ model shown as Figure 2.5(c).



(a) Lumped model



(b) Distributed Model



(c) π3 Model

Figure 2.5.
Simulation models of interconnect wires

2.2.2.2. The Values of the Circuit Elements

Once a procedure has been chosen for the sub-division of the distributed interconnections, the effective values of resistance and capacitance for each element must be calcu-

lated. The process of obtaining these values is called interconnect parasitics extraction.

There are many ways to extract the parasitics. However, the applications of empirical formulae to general submicron interconnects are rather limited because of the complexity of interconnect configurations in multi-level submicron technology, especially in extracting the coupling capacitances. These formulae can not cover layout configurations having multiple dielectric and metal layers, and most importantly, they are not accurate enough to capture the variations of the layout and technology parameters of interconnect. Furthermore, they are obtained by fixing the values of the technology parameters. So their applicability in our work of sensitivity study over the technology parameters is not appropriate.

In "exact" computations of electrical circuit parameters, one appeals to the theory of electromagnetic fields; that is, "exact" computations involve the numerical analysis of two or three dimensional integral or partial-differential equations for the values of an electromagnetic field. The accuracy of such solutions is limited in theory only by the number of grid points, the availability of computing resources, and by human patience in describing the geometry of actual interconnect structures.

Since multi-level interconnect technologies use multiple conductors with different thicknesses and multiple insulators with possibly different dielectric constants, numerical simulations are mandatory for accurate resistance and capacitance modeling [23] [24] [25]. Numerical techniques have been developed for rigorous interconnect capacitance extractions. They fall into the following three categories [3]:

(1) finite-difference method [4] [5];

(2) finite-element method [3]

(3) Green's function method [4].

All these methods use the quasi-transverse electromagnetic approximation and divide the space surrounding the object into meshes, and use local equations at each mesh point.

There are several ready-to-use or commercial extraction tools available, such as Fastcap [23], Space[29], and Raphael [26]. Among these, we found that Raphael is the easiest one to use because of its good user interface, and its capability to perform batch-mode simulation and two-dimensional simulations. So, Raphael is chosen as the extraction tool in this work.

2.3. Interconnect Model Library

We have concluded the necessity of numerically based simulation to perform interconnect parasitics extraction. However, numerical simulation is computationally intensive and real-time simulation is too time-consuming. Furthermore, in our approach to perform sensitivity study, all the interconnect wires are to be modeled using closed-form analytical models, which requires parameterized interconnect models. To cope with this problem, a realistic approach is to construct a parameterized interconnect model library based on numerical simulation. Then the circuit description can be generated with the help of the model library which contains models of typical two-dimensional interconnect structures. The circuit description will thus become the basis of sensitivity study and statistical circuit simulation.

Figure 2.5 shows the flow of this interconnect modeling approach and model library building. First, the possible and typical interconnect configurations are identified. The most common configurations encountered are listed as follows:

a. One conductor above a ground plane.

b. Two conductors above a ground plane.

c. Three conductors above a ground plane.

d. More than three conductors above a ground plane.

e. One conductor between two ground planes.

f. Two conductors side by side between two ground planes.

g. Three conductors side by side between two ground planes.

h. More than three conductors side by side between two ground planes.

These interconnect structures are two-dimensional structures, so they are easier to define in terms of both layout and technology parameters than three-dimensional structures.

The sensitivity analysis is performed under the assumption that the variation of technology parameters, interlayer dielectric thickness, conductor thickness, is within ±20% of its nominal value for a given technology, while the ranges of the layout parameters, such as metal width, inter-wire spacing, are set according to the design rules and their possible design ranges. Each parameter is divided into several levels, and a full factorial design is used to generate the simulation points for each structure. The input file for the numerical simulator which contains all the simulation points is generated, and two dimensional simulations are performed in batch-mode using a numerically based extractor (Raphael) to evaluate the unit length capacitance and resistance values. The numerical data are then fitted with an analytical expression using a special curve-fitting technique which is discussed in more detail in Section 2.5.

In the following sections, we will use the structure in Figure 2.6 as an example to illustrate the approach in more detail.

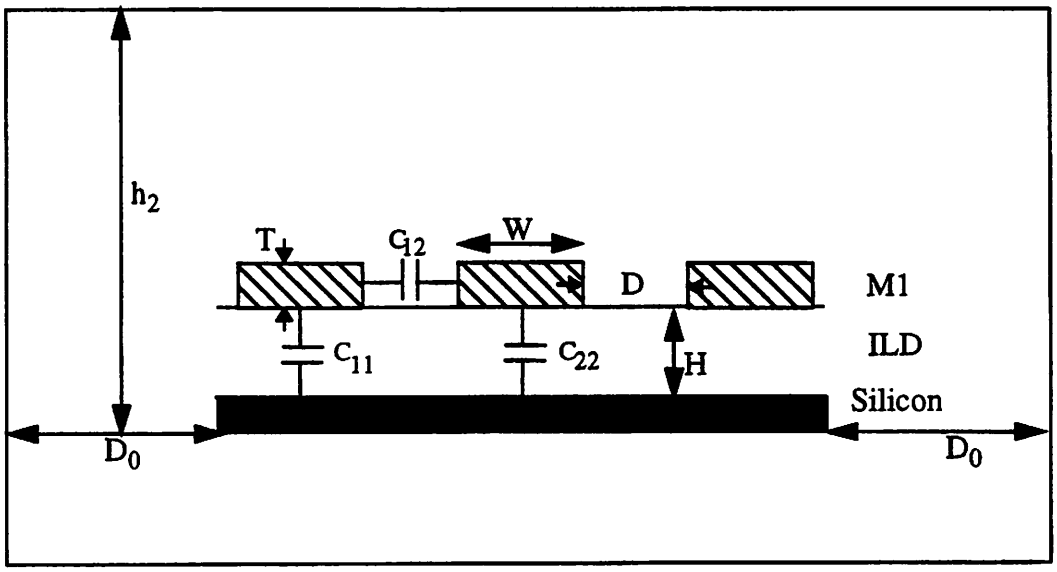


Figure 2.6.
The cross section of an interconnect structure

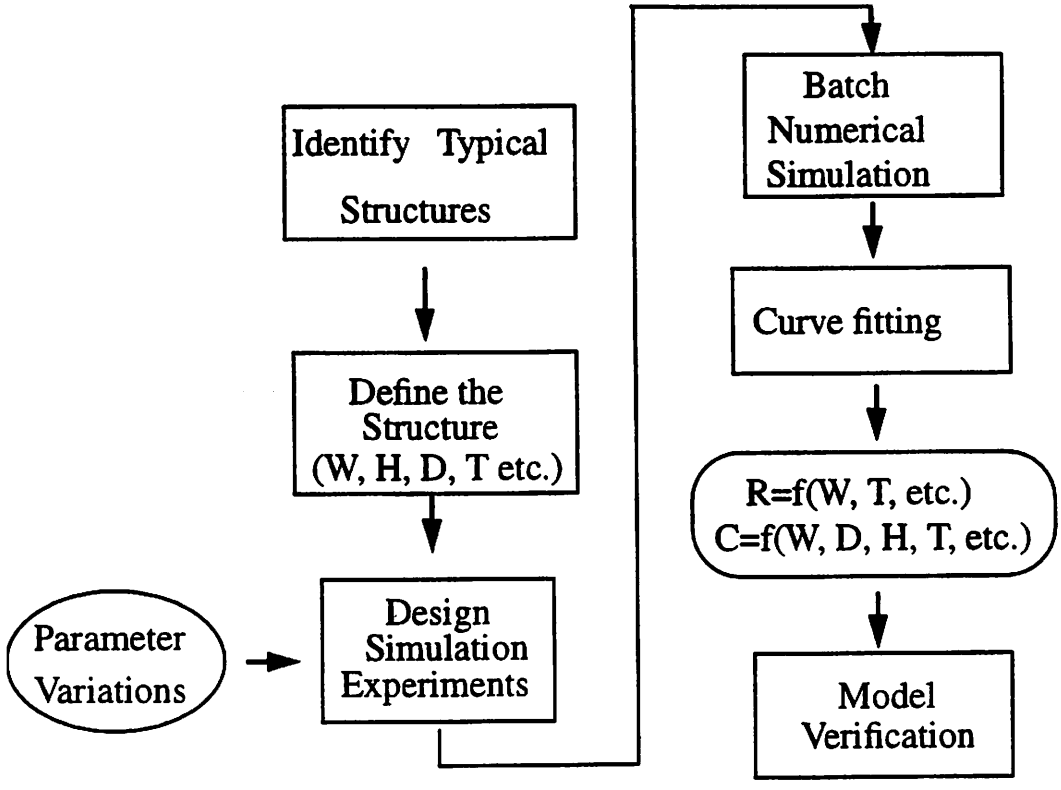


Figure 2.7.
Interconnect Modeling Flow

2.4. Numerical Simulation

The interconnect structure shown in Figure 2.6 is defined in terms of four parameters, metal width W, inter-wire distance D, metal thickness T and ILD thickness H. The three capacitances, C_{22}, C_{11}, and C_{12} are of interest and their models are constructed. Notice that all inter-wire spacings are the same in this structure.

The input file to the numerical extractor Raphael is generated as follows: each parameter, W, D, T and H, is set at 6, 6, 7 and 7 levels respectively. Using a full factorial design, a total number of 1764 simulation points (combination of different levels of the four

parameters) are generated, and the input file for Raphael is thus created based on these points.

The created input file which contains one simulation point at each row looks like:

H	T	W	D	h_2	s_0	ε_1	ε_2	ρ
0.955	0.51	0.9	0.5	10	12	3.9	3.9	0.01
0.955	0.53	0.9	0.5	10	12	3.9	3.9	0.01
0.955	0.55	0.9	0.5	10	12	3.9	3.9	0.01
...	...	...	...	...	...	...	...	...

Table 2.3.

Input File to Numerical Extractor for the structure depicted in Fig.2.6.

In the above table, h_2 and s_0 defines the simulation boundaries as indicated in Figure 2.6, ε_1 and ε_2 are the dielectric permitivities, while ρ is the resistivity of the metal. Each row of the table stands for one simulated experiment.

For this structure, it takes about 8 hours for Raphael to finish the 1764 numerical simulations on a Sparc 20 workstation. The next step is to construct an analytical model based on these simulation results. This is discussed in the next section.

2.5. Curve-fitting Technique

2.5.1. Linear least square regression analysis

The objective is to create a model $\boldsymbol{y} = \boldsymbol{F(x)}$ that maps the relationship between the set of parameters defining the physical interconnect and the values of the parasitics of the interconnect. Here $\boldsymbol{y}$ is the n dimensional vector representing the capacitances and resistances to be modeled, and $\boldsymbol{x}$ is the m dimensional input vector containing all the interconnect parameters.

This is implemented using simple polynomial expressions and linear regression. Linear regression postulates a model of the form:

$$y = \beta_0 + \sum_{i=1}^{M-1} \beta_i x_i + \varepsilon \tag{2.6}$$

(2.7)

where x_i stands for the independent variables. The terms that should be included in the model can be of first, second or higher order, or even interaction terms made of the group of parameters which define the interconnect structure. The final expression is determined using a technique discussed in Section 2.5.2. The least squares criteria chooses the β_i values that minimize the sum of squared residuals:

$$\sum_{j=1}^{N} \left\{ y_j - \left[\beta_0 + \sum_{i=1}^{M-1} \beta_i x_{i,j} \right] \right\}^2 \tag{2.8}$$

where j indexes each of the N simulation points.

The equations can be expressed more compactly in matrix form. Let $\mathbf{X}$ be the N by M data matrix; i.e. $\mathbf{X}$ looks like

$$\begin{bmatrix} 1 & x_{11} & x_{12} & \cdots & x_{1M} \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ 1 & x_{N1} & x_{N2} & \cdots & x_{NM} \end{bmatrix}$$

Also, let $\mathbf{y}$ be the N by 1 column of observed responses and let $\hat{\beta}$ be the M by 1 column of estimated coefficients, with the first element being the intercept. Then any $\hat{\beta}$ satisfying the normal equations:

$$(\mathbf{X}^T\mathbf{X})\hat{\beta} = \mathbf{X}^T\mathbf{y} \tag{2.9}$$

gives a least squares fit, assuming that $\mathbf{X}^T\mathbf{X}$ is invertible.

Linear regression implies that the model is linear in the coefficients being estimated, that is, linear in β. The models built in this work are linear in this sense. Another

common use of the term "linear" is to describe whether the model is linear in the variables. Better referred to as the order of the model, this work does include non-linear terms such as quadratic or interaction terms of the input variables ***X***.

. As stated above, the capacitance models include quadratic and higher order terms of the parameters, together with their interaction terms. In order to achieve an accurate model, the first step would be to determine the terms to be included in the final model. After the model terms have been determined, the coefficient of each term can be estimated using the least squares technique. However, choosing the terms of the model is a difficult task. In the following section, we present a systematic way to address this problem.

2.5.2. Determining the Necessary Model Terms

It is obvious that if the model terms are incorrect or inappropriate, one can not get an accurate capacitance model. Most of the time, the models built by the trial-and-error approach are not good enough to be used for sensitivity analysis. However, one can find an efficient and systematic solution to this problem, guided by the simple physical relationships between the input variables and the resulting capacitance.

First, we select the data points that are obtained by varying one parameter with the other parameters fixed, then these data points are fitted over this parameter using step-wise regression. This is easy since only one variable or parameter is involved. In this way, one will get a separate model related to each of the parameters. These models are simple polynomial functions. In some cases, non-linear data transformations are necessary in order to apply a linear model. Combining these separate models, the final model terms are easy to identify. This is illustrated as follows:

Suppose that capacitance C is an unknown polynomial function of two variables, W and D. That is, $C = f(W, D)$, and the goal is to choose the proper model terms based on discrete data points.

Let us assume that by curve-fitting over W with D fixed, one finds that C is a linear function of W, that is,

$$C = aW+b \qquad (2.10)$$

In the above equation, a and b are constants, and they will take different values when S is fixed at different points. So a and b are only functions of D, such as

$$a = f_1(D) \text{ and } b = f_2(D) \tag{2.11}$$

then (2.10) can be rewritten as:

$$C = f_1(D) \cdot W + f_2(D) \tag{2.12}$$

Suppose that C is fitted over D with W fixed, and the fitting result shows C is a second order polynomial function of D. Following the same argument as above, one can conclude that C can also be expressed as:

$$C = g_1(W) \cdot D^2 + g_2(W) \cdot D + g_3(W) \tag{2.13}$$

Note that (2.12) and (2.13) are equivalent expressions. Since both equations represent capacitance, which must be continuous functions of both D and W, we can take the derivative of these two equations over W, which leads to the following expression:

$$f_1(D) = \frac{\partial}{\partial W} g_1(W) \cdot D^2 + \frac{\partial}{\partial W} g_2(W) \cdot D + \frac{\partial}{\partial W} g_3(W) \tag{2.14}$$

Thus, the model terms of $C(W, D)$ can be decided. Given the fact that the left-hand side (LHS) of (2.14) does not depend on W, the right-hand side (RHS) should also be independent of W. Then we can conclude that $g_1(W)$, $g_2(W)$, *and* $g_3(W)$ are linear functions of W, i.e.,

$$g_1(W) = k_1 W + k_2; \tag{2.15}$$

$$g_2(W) = k_3 W + k_4; \tag{2.16}$$

$$g_3(W) = k_5 W + k_6; \tag{2.17}$$

Here, k_i (i = *1* to *6*) are constants. By substituting (2.15), (2.16) and (2.17) into (2.13), the equation can be rewritten as:

$$C(W, D) = (k_1 W + k_2) \cdot D^2 + (k_3 W + k_4) \cdot D + (k_5 W + k_6) \tag{2.18}$$

To simplify the situation, the above illustration assumes that C is a linear function of W when D is fixed. If C is a second or higher order function of W, the above explanation still applies on taking higher order derivatives, which will lead to similar results.

In practice, the above technique is used to choose the terms of the model, while the coefficients of the terms are determined using least squares fitting.

Though only two variables are discussed in the above technique, the approach can be easily generalized to three and more variables. Also, note that even though the above discussion is not a strict mathematical proof, it does provide us with some insight and guidance on data fitting in order to get an accurate capacitance model.

2.6. Example

We now apply this technique to model the results of an array of simulations performed on the example depicted in Figure 2.6.

Figure 2.8, Figure 2.9, Figure 2.10, and Figure 2.11 show how the capacitance C_{22} changes with metal width, inter-wire spacing, metal thickness and ILD thickness, respectively. It is easy to see from Figure 2.8 that C_{22} is linear with the metal width W and metal thickness T, respectively. The relationships between C_{22} and the other two variables also agree with physical intuition.

It is also found that C_{22} can be modeled as a third order polynomial function of inter-wire spacing S. The plot of Figure 2.11 indicates that data transformation will help the modeling of C_{22} against ILD thickness H. So C_{22} is modeled as a second order polynomial function of H_i^{-1}.

Based on the four individual models, the complete analytical model terms are thus chosen, and the coefficients are determined by the least squares criterion. Table 2.4 summarizes the modeling results:

Table 2.4 indicates that multiple R^2 is 0.9999, which means 99.99% of the variation can be explained by the model. F-ratio is the ratio of the mean square of the regression to the estimated variance, and the zero p-value means the ratio is very significant. However, one can not conclude that the model fits the data well just by looking at this table. Further analysis is necessary to assess the model.

Source	Sum of Squares	Degree of Freedom	Mean Squares	F-ratio	Prob>F
Model	91.1037	47	1.938376	613700	0
Error	0.00542	1716	3.158eE-6	$R^2 = 0.9999$	
Total	91.109	1763		$\sigma = 0.00177$	

Table 2.4.

Statistics of Modeling Result of C_{22} Model

The simplest and most informative method for assessing the fit is to plot the response against the fitted values, and also examine the residuals. Figure 2.14 shows the predicted values versus the simulation data. The straight line indicates good fitting of data. Figure 2.15 is the normal plot of the residual, and it gives no reason to doubt that the residuals are normally distributed. This is further justified by residual plots shown in Figure 2.12 and Figure 2.13. Since the minimum value of C_{22} is 0.33 (scaled by 1E-16), the ratio of the residual standard error, σ in Table 2.4, over the minimum of simulation data C_{22} is just 0.5% (0.0017/0.33).

The above analysis shows that the model fits the data very well, the regression is significant, and the residuals appear normally distributed. This underscores the usefulness of the technique.

Figure 2.16, Figure 2.17, Figure 2.18 and Figure 2.19 depict the plot of C_{12} against the four interconnect parameters. The clear patterns of their relationships indicate that we can also use the same fitting technique to find the form of the analytical model of C_{12}. Since the process is similar, it will not be repeated here.

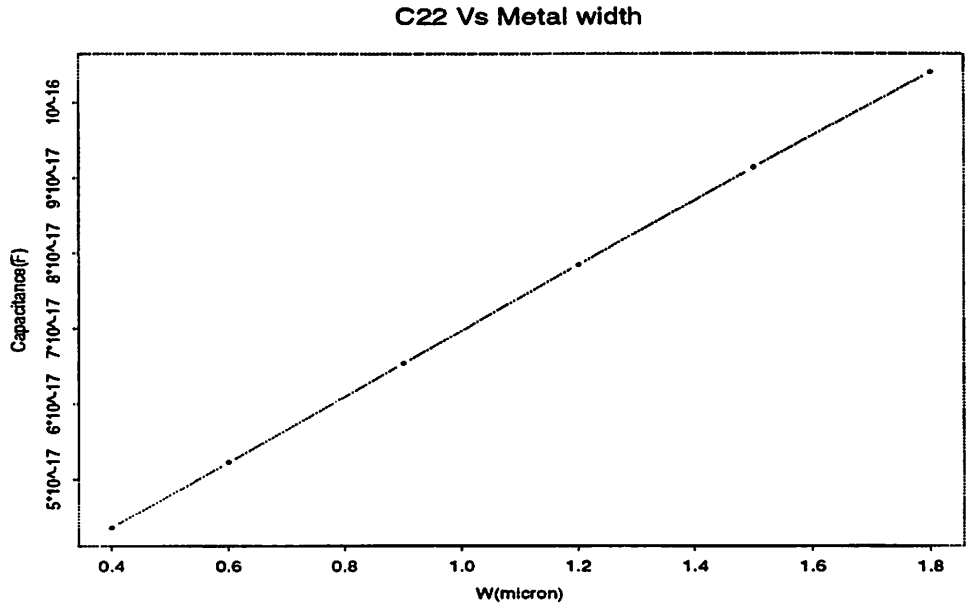


Figure 2.8.
Capacitance C_{22} versus metal width W with H, T, and S fixed.

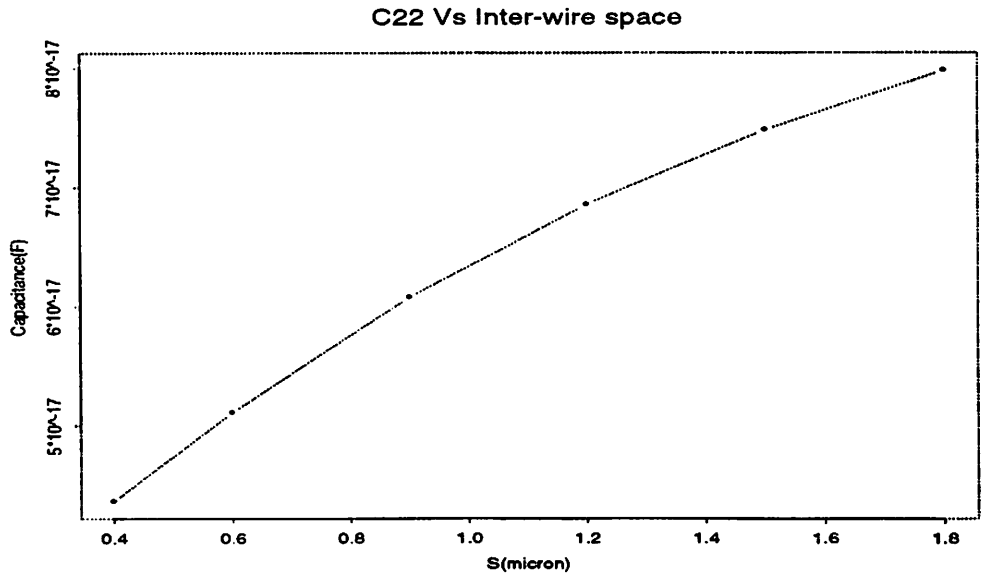


Figure 2.9.
Capacitance C_{22} versus inter-wire spacing S with W, H, and T fixed.

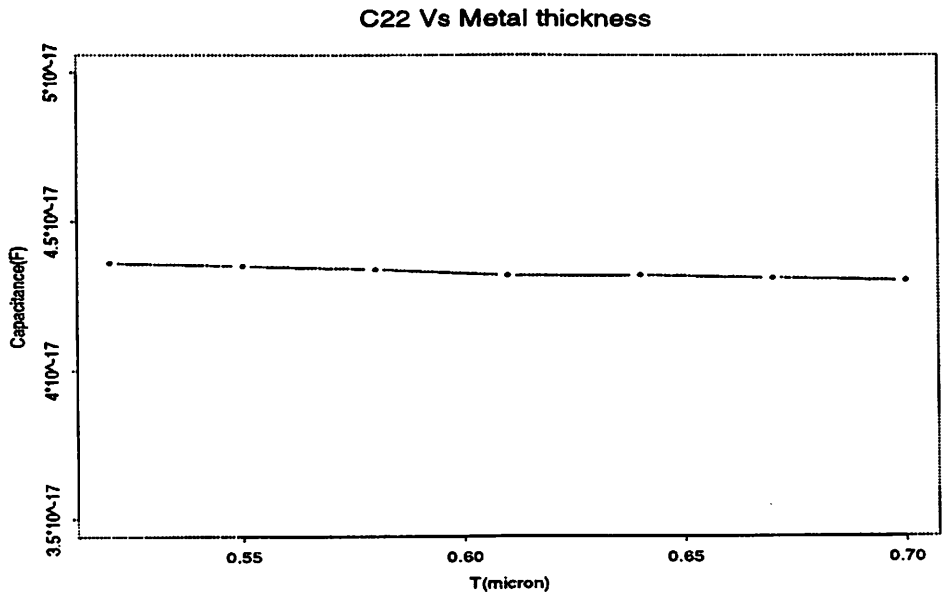


Figure 2.10.
C_{22} versus metal thickness T with W, H, and S fixed.

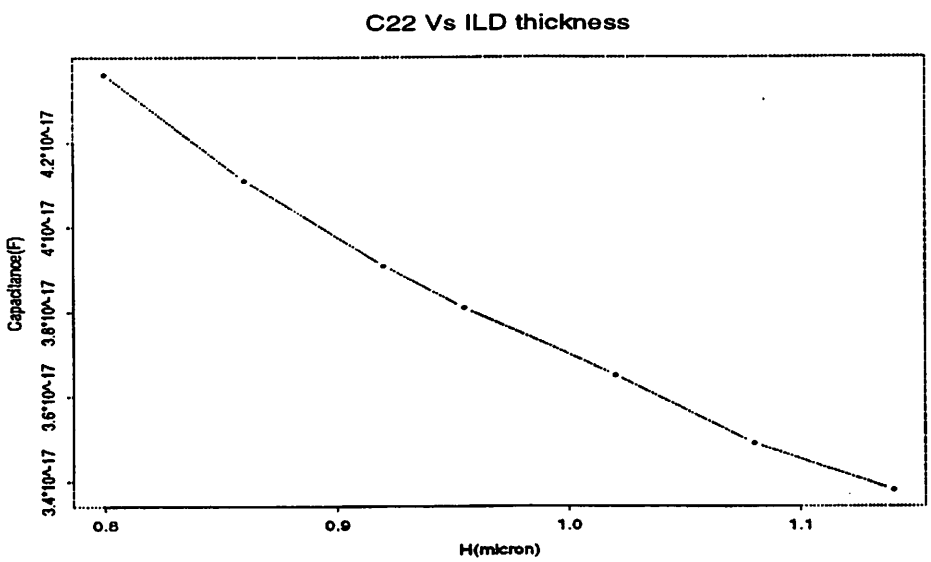


Figure 2.11.
C_{22} versus ILD thickness H with W, T, and S fixed.

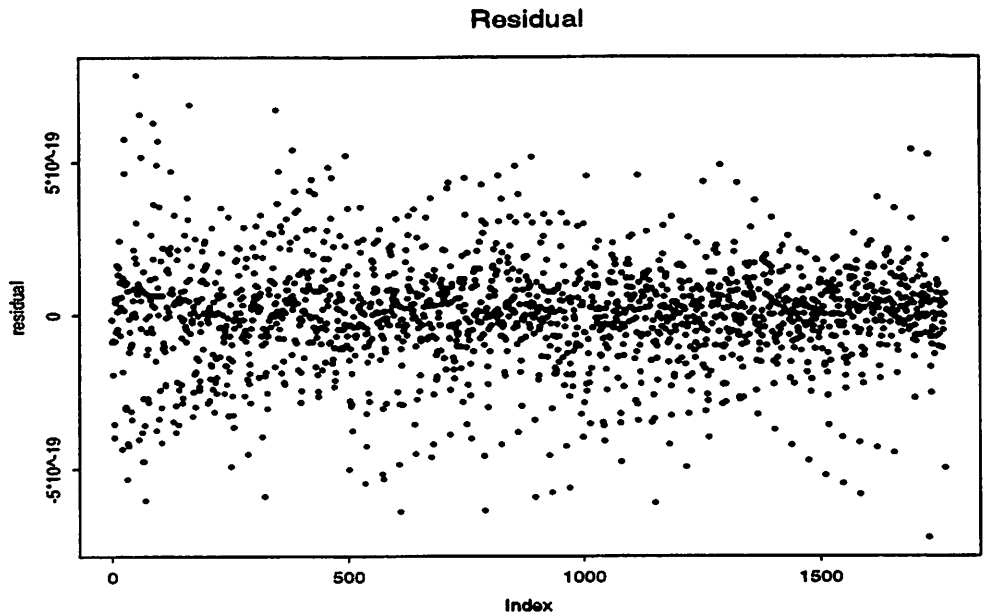


Figure 2.12.
Residual of regression model of C_{22} as a function of H, W, T, S.

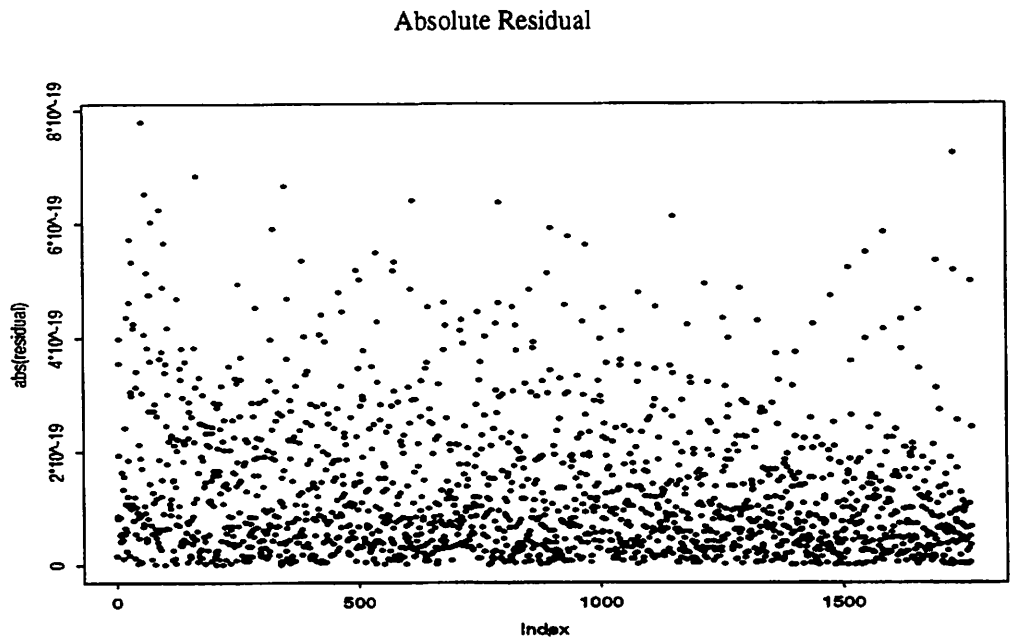


Figure 2.13.
Absolute residual of regression model of C_{22}

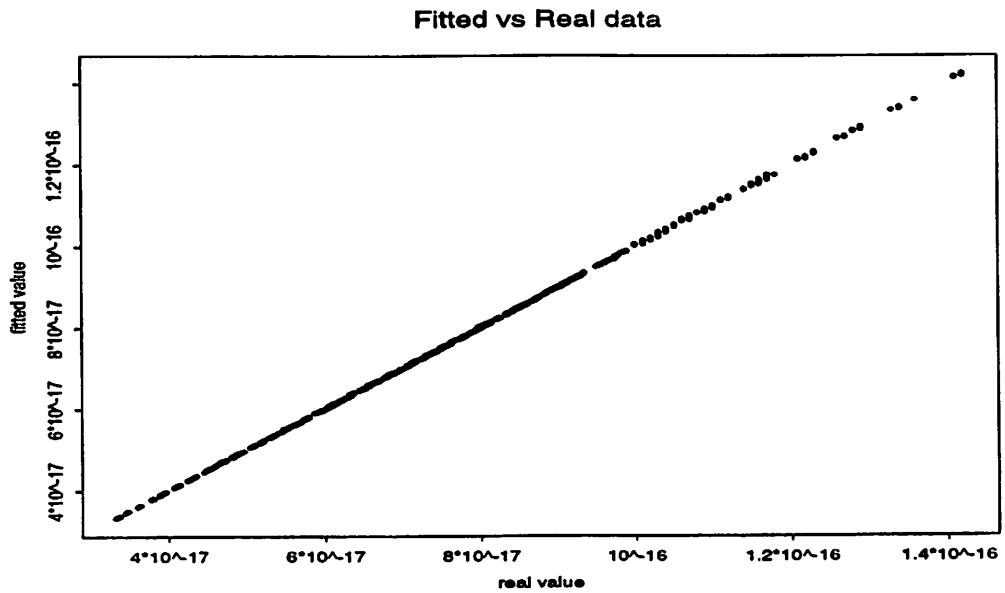


Figure 2.14.
Fitted values against simulated data of C_{22}

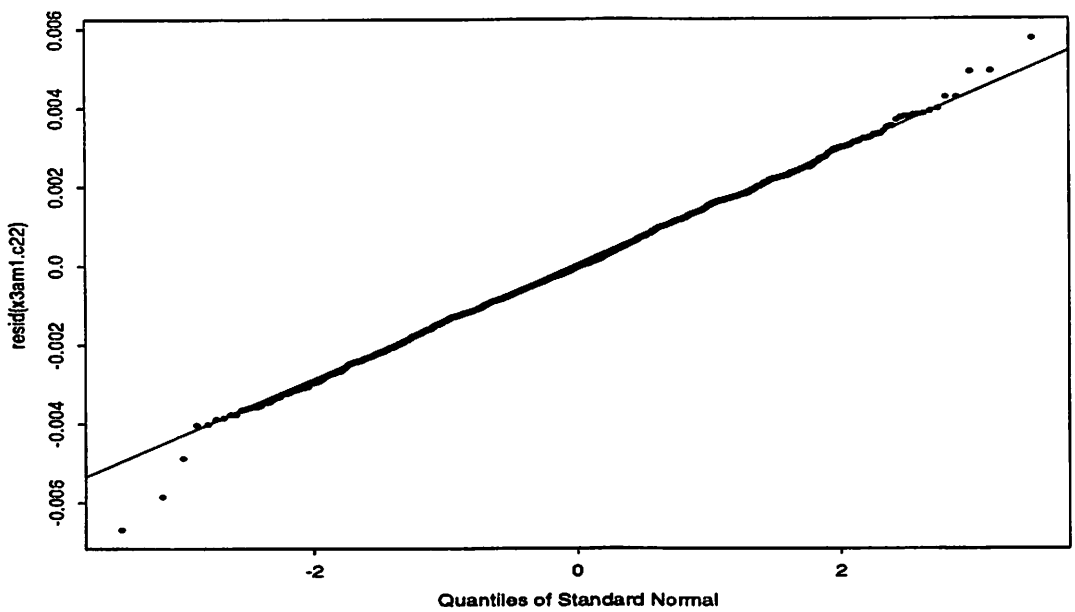


Figure 2.15.
Normal plot for residuals.

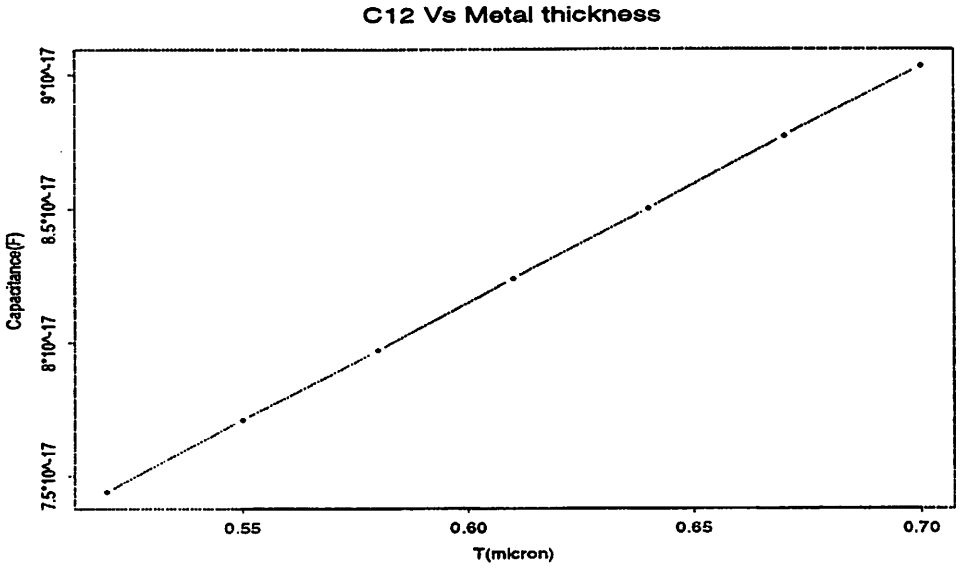


Figure 2.16.
C_{12} versus metal thickness

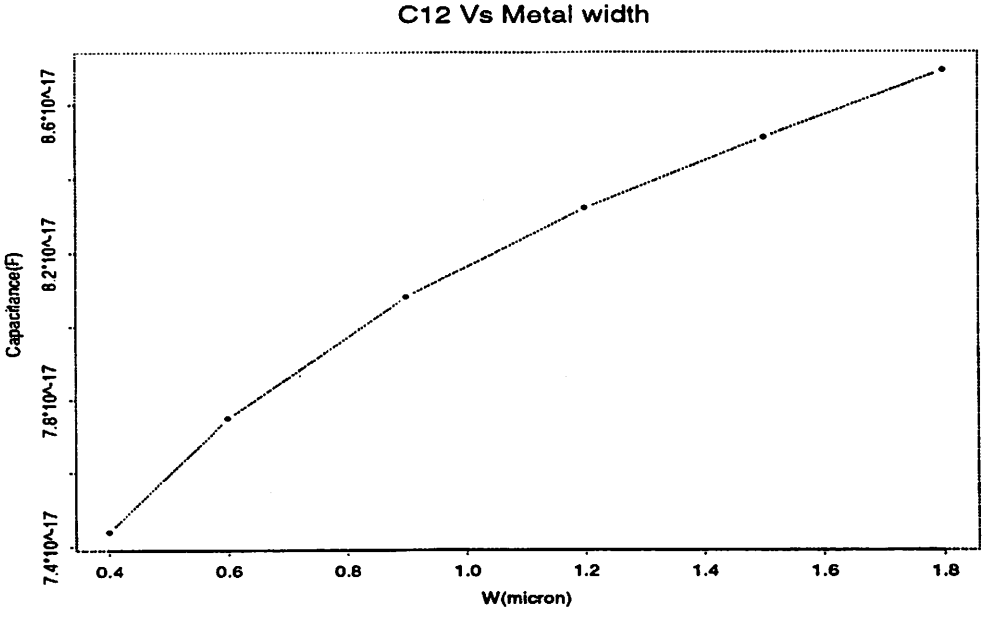


Figure 2.17.
C_{12} versus metal width

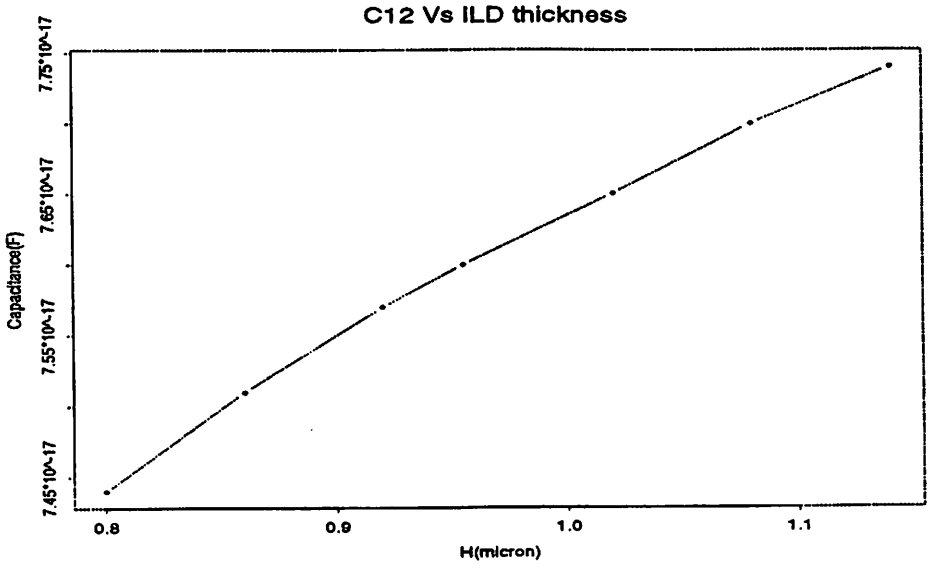


Figure 2.18.
C_{12} versus ILD thickness

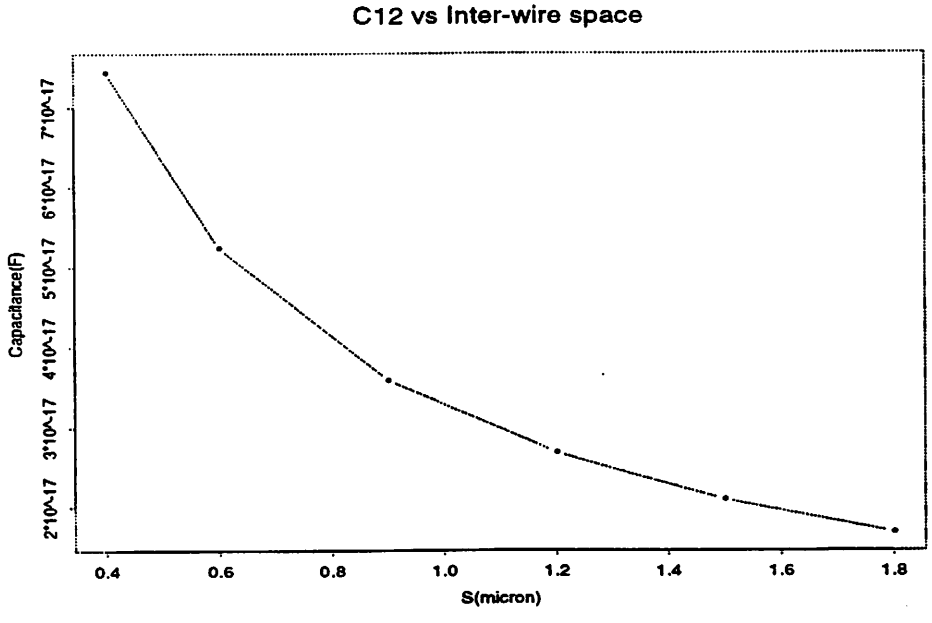


Figure 2.19.
C_{12} versus Inter-wire Spacing

2.7. Summary and Conclusion

In this chapter we discussed the issues of interconnect modeling for the purpose of the sensitivity study, which will be discussed in next chapter. We concluded the necessity of numerical simulation in order to extract accurate interconnect parasitics. An efficient technique of curve fitting is discussed in detail, and a specific fitting problem is solved as an example. We also presented a methodology to build a parameterized interconnect model library.

Based on the interconnect model library built, we are ready to develop an approach to study the circuit sensitivity to interconnect variations. This is discussed in the following chapter.

Chapter 3 Sensitivity Study

3.1. Introduction

In Chapter 2, we discussed the problem of interconnect modeling. We built an interconnect model library, so we are now ready to develop an approach to study the effects of interconnect on circuit performance. The goal is to anticipate the effects of processing variation of an IC product, and then characterize and control this variation during production.

The models developed in the last chapter, and the established range values for interconnect parameters are the essential ingredients for the evaluation of the impact on circuit performance. In this chapter, an approach to accomplish this evaluation will be explored. A representative ring oscillator test circuit will be used as a means to test the interconnect models, and the relationships between interconnect parameter variations and circuit performance will be developed.

3.2. Previous Work

The goal of statistical circuit design is to model and improve parametric yield [15]. The underlying concept is that variations in the manufacturing process change the performance of the integrated circuit and therefore cause the performance yield fluctuations seen in the final test. However, previous work is based on the fact that circuit performance is mainly determined by transistors or devices [12] [14] [15] [16], which is not the case in the era of deep submicron. The work only studies the effects of manufacturing line variations on fabricated device variations, and on circuit performance. Most of previous work does not take the interconnect variations into account.

For example, early work at Texas Instruments proposed characterizing CMOS processes with four statistical parameters: device width, device length, oxide capacitance, and flat-band voltage [12]. These parameters were identified because they explained most of the observed variations in the process and were determined to be nearly statistically independent. Equations for delay and power dissipation were then computed as a linear function of these parameters, and the resulting equations were then used for yield optimization [17].

Obviously, these efforts can not be applied to the modern submicron CMOS processes if the interconnect dominates the performance of the circuit. Since devices and interconnect wires are different in many aspects, a new approach needs to be developed.

To incorporate the interconnect into the framework, there are four components in the statistical circuit design, as shown in Figure 3.1. The manufacturing process must be described in a way which characterizes the process variations responsible for the yield. The manufacturing line variations must be mapped into the variations of devices and interconnect wires, and then be mapped into the performance variation of circuits, which is achieved by circuit simulation. Finally, an algorithm is used to optimize the yield.

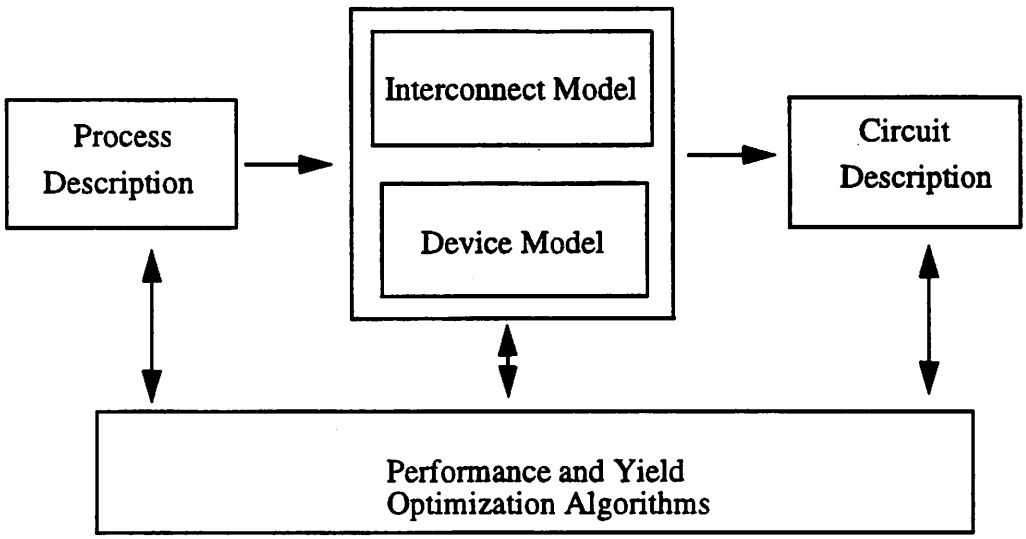


Figure 3.1.
Elements of statistical circuit design

3.3. Methodology

The methodology to perform the sensitivity study will be discussed in detail in this section. Some elements of the simulation will also be discussed.

An overview of our understanding of circuit sensitivity to interconnect variations is shown in Figure 3.2. The basic idea is to model each interconnect wire of a circuit using the parameterized interconnect models developed in Chapter 2, and then generate the circuit description based on a Spice file. The generated circuit description contains closed-form analytical expressions for each interconnect capacitance and resistance elements, and it is the basis of the statistical circuit simulation.

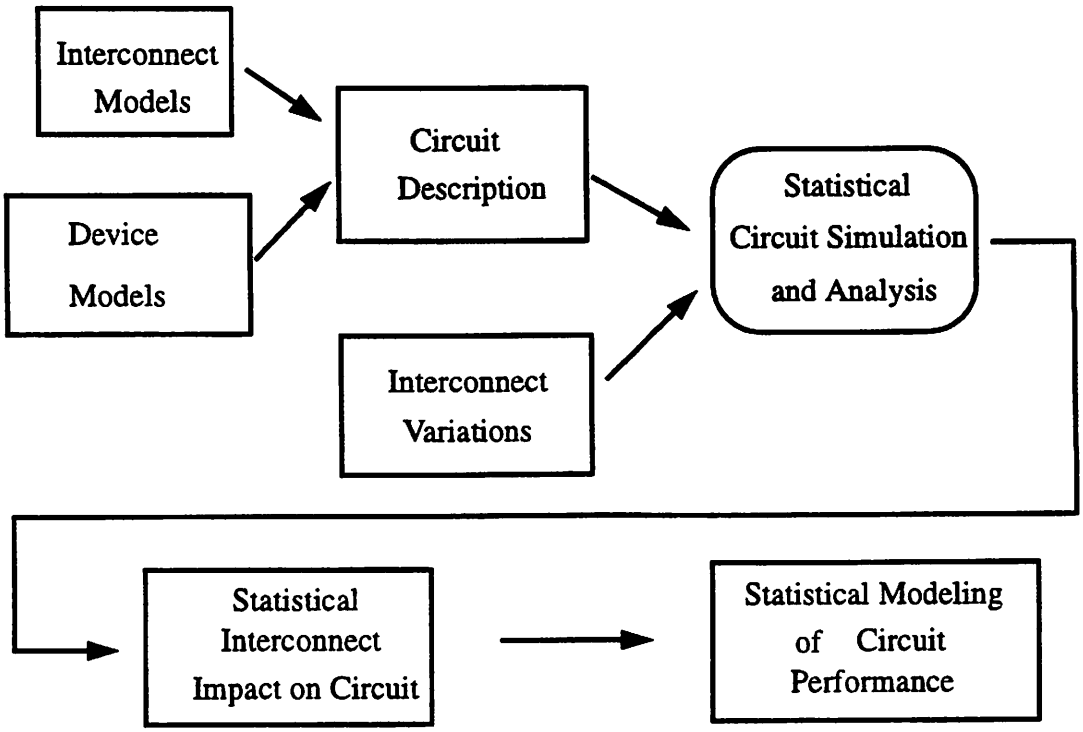


Figure 3.2.
Overview of sensitivity study

3.3.1. Interconnect Wire Model

To simulate the effect of process variation on a circuit, the connection between the process parameters and the input file to circuit simulator must be established. So the RC model for each interconnect wire should be expressed in terms of the interconnect parameters. With the help of the interconnect model library developed in the last chapter, the total capacitance and resistance of each interconnect wire can be easily described given the length of each wire, thus a RC model of each wire is built. The resulted description of the interconnect wires in a circuit usually take a form of an RC mesh because of the coupling capacitances among neighboring wires.

3.3.2. Device Model

In this work, the BSIM level 27 MOSFET model is used, which mapped the fabrication line parameters to device characteristics. The model parameters can be separated into three groups. The first group contains physical constants, the second group includes the measurable process parameters, such as gate oxide thickness, and the last group consists of the fitting parameters, which can be extracted to match the measurable process parameters.

3.3.3. Circuit Description

The circuit description is generated for use in circuit simulation. It contains both the transistors and interconnect wires description with the help of the interconnect and device models discussed above.

3.3.4. Circuit Simulation and Optimization

We use the SPICE circuit simulator to estimate circuit performance. Gradient and computer-based experimental design methods have been used to increase yield by optimizing the transistor sizes or the topology of a circuit for a process. The gradient method is a standard nonlinear optimization technique which can be implemented within a circuit simulator for use in yield optimization [17]. Experimental design techniques have focused on the use of Taguchi Robust Design methods to select optimum transistor sizes [18] [19].

In contrast, our work does not use formal optimization techniques to improve yield as the focus of this work is on the sensitivity analysis. More specifically, our goal is to determine the impact of interconnect related process parameters on performance.

3.3.5. Statistical Circuit Simulation

The variation ranges of interconnect parameters form a multidimensional region which is referred to as a *parameter space*. This parameter space will be mapped to the variation ranges of the performance which is referred to as the *performance space*.

3.3.5.1. Monte Carlo Simulation

The goal is to map the parameter space into the performance space, that is, to determine what will be the corresponding performance for each point in the parameter space.

The experiment involves selecting a set of points in parameter space. For each point selected, a circuit simulation is run to find the circuit performance. Monte Carlo simulation is a technique [20] which can cover the parameter space with a reasonable number of experiments. Often, statistical screening experiments, such as orthogonal arrays or factorial design are utilized to screen a large number of potential parameters.

The advantage of Monte Carlo convergence on an estimate of response lies in the fact that its cost is independent of the dimensionality of the problem, depending only on the desired precision of the analysis.

3.4. Case study: Ring Oscillator Circuit

A ring oscillator was used to explore the sensitivity analysis approach. Figure 3.3 shows part of the circuit diagram of the ring oscillator, which emphasizes the interconnect wires between stages. The loading of the circuit is dominated by interconnect wires, as indicated in Figure 3.3. The interconnect length for each stage is 180 μm, and is divided into 6 fingers as shown in the figure. Three of the fingers are next to previous stage fingers and the other three are next to next stage fingers, so there is a heavy capacitive coupling effect between neighboring stages.

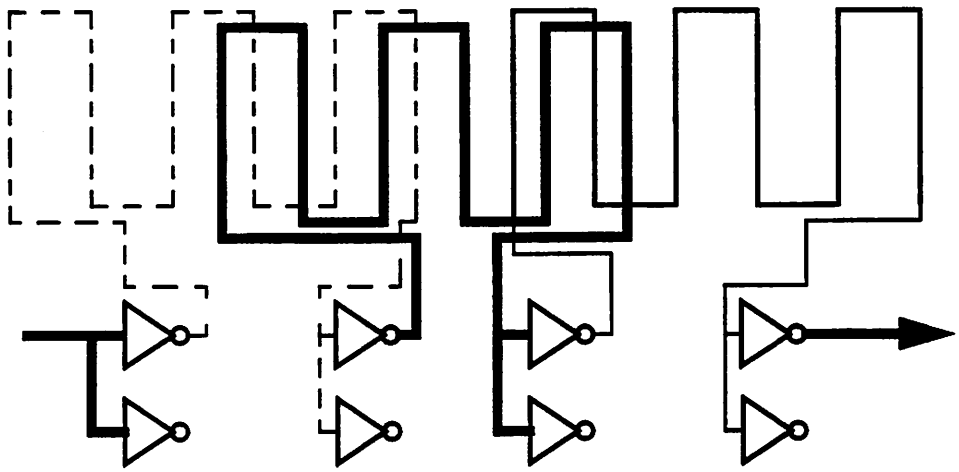


Figure 3.3.
Circuit diagram of a ring oscillator

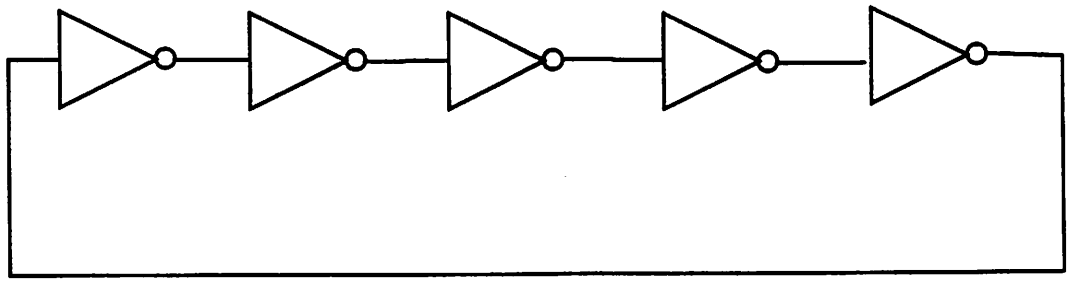


Figure 3.4.
Ring oscillator circuit is composed of odd number of inverters

The ring oscillator is a standard circuit for delay measurements. It consists of an odd number of inverters connected in a circular chain as shown in Figure 3.4. Due to the odd number of inversions, the circuit does not have a stable operating point and oscillates. The period T of the oscillation is determined by the propagation time of a signal transition through the complete chain, or

$$T = 2 \cdot t_p \cdot N \ , \tag{3.1}$$

with N the number of stages in the chain and t_p the propagation delay of each stage. The factor 2 results from the observation that a full cycle requires both a low-to-high and a high-to-low transition.

The ring oscillator circuit used in this study has nine stages, with fan-out of 2. However, the design is such that significant loading is contributed by interconnect wires. In this way, the signal delay t_p between each stage is mainly determined by the interconnect capacitance and resistance.

3.4.1. Circuit Model

To study the circuit performance sensitivity to interconnect parameters, we generate the Spice netlist file of the ring oscillator. The netlist generated from the extraction tool is modified so that the interconnect wires of the circuit are modeled in terms of the interconnect parameter. For example, coupling capacitance is modeled explicitly in terms of the length and distance of the wires. The regularity of this relatively simple ring oscillator circuit makes it easier to accomplish this modification. The fingers are parallel and have the same width and the same inter-wire space. By generating the circuit description in this way, a direct link between the circuit performance and interconnect parameters is established. The final circuit model is listed in Appendix C.

3.5. Results and Analysis

The ring oscillator circuit is simulated using HSPICE. The sensitivity of the delay to a particular parameter is evaluated by varying it over a reasonable range with the other parameters fixed. For example, the delay sensitivity to metal thickness is obtained by fixing the ILD thickness, metal width and metal spacing and varying the metal thickness over ±20 % variation range.

Figures 3.5 to 3.8 show the simulation results of the delay sensitivity to the wire width, inter-wire spacing, ILD thickness and metal thickness, respectively. The roughness of the curves are caused by the reading error and the limited numerical resolution of HSPICE. Table 3.1 summarizes these results. It indicates that inter-wire spacing is the most sensitive parameter. 20% variation of the inter-wire spacing from its nominal value will lead to 8.8% deviation of the delay. On the other hand, the circuit is not sensitive to the variation of the ILD thickness in the range of the simulation.

The lack of sensitivity of ILD thickness is because the delay is not sensitive to the plate capacitance. In fact, for this circuit, the delay is mostly sensitive to inter-wire coupling capacitance as can be seen in Figure 3.14. Also, it is because of the fact that resistance does not change along with ILD thickness, and that changing of inter-wire capacitance due to ILD thickness is almost cancelled by that of plate capacitance.

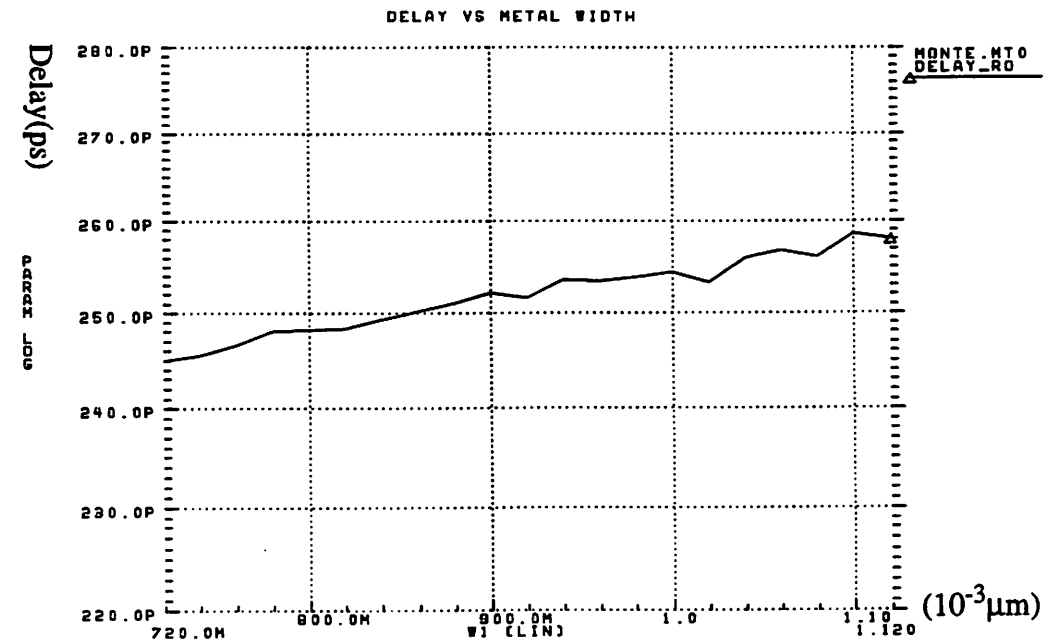


Figure 3.5.
Delay versus metal width

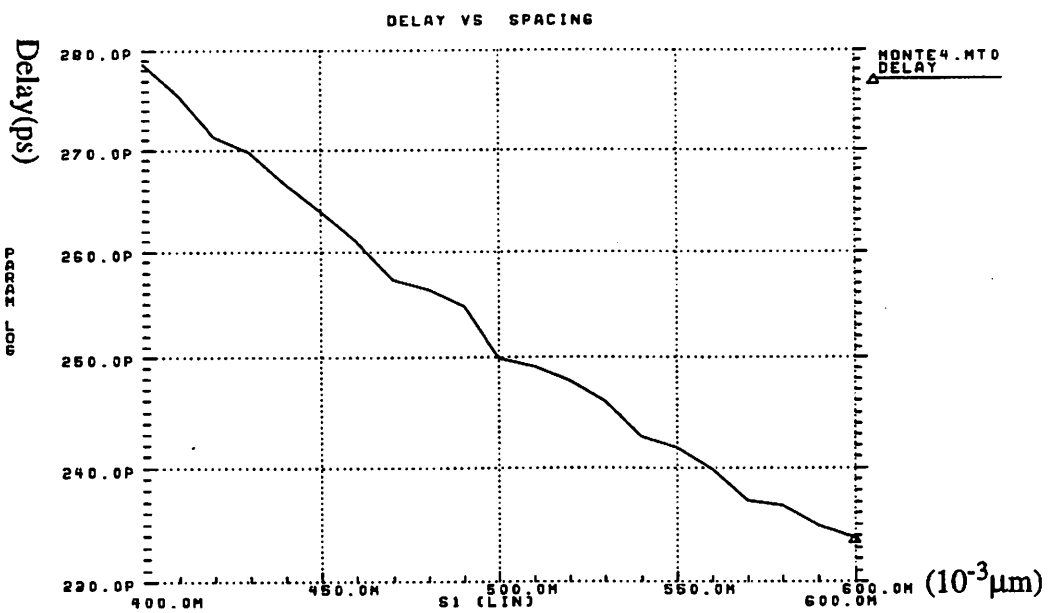


Figure 3.6.
Delay versus inter-wire spacing

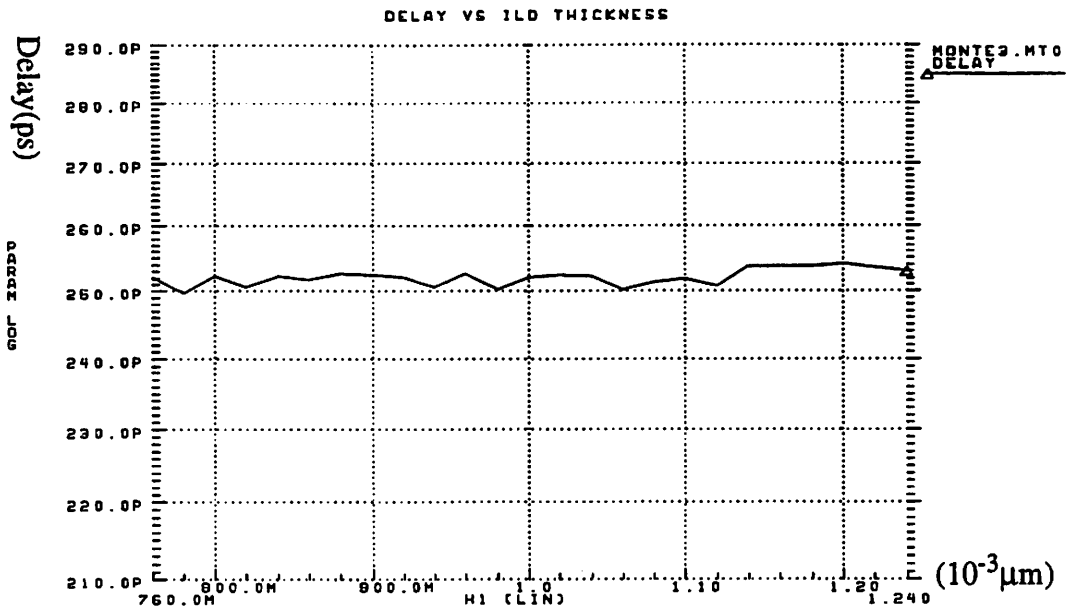


Figure 3.7.
Delay versus ILD thickness

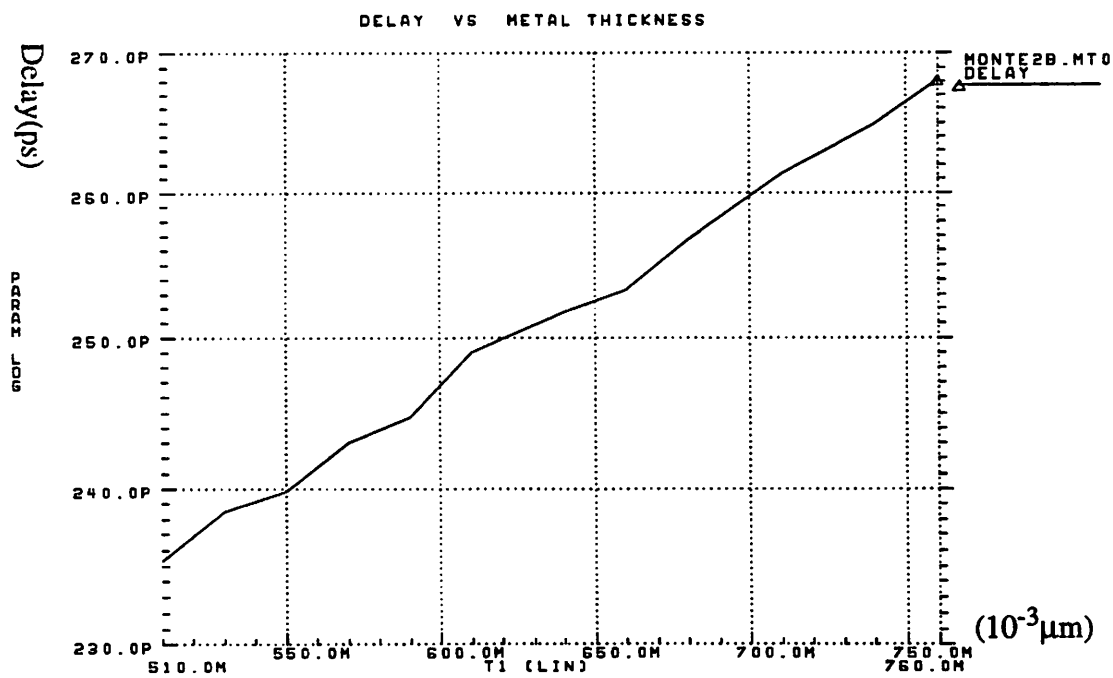


Figure 3.8.
Delay versus metal thickness

Parameter	Parameter range (μm) (20% variation)	Impact on Delay
Metal width	0.810-0.990	2.6%
ILD thickness	0.842-1.029	0.0%
Metal thickness	0.572-0.699	4.6%
Inter-wire spacing	0.450-0.550	8.8%

Table 3.1.
Sensitivity Simulation Results

To get further insight and generalize the methodology, Monte Carlo simulations are also set up to perform statistical analysis. These statistical simulations closely reflect what happens in the real world. Monte Carlo analysis is an effective way to provide the information for improving circuit robustness to interconnect variation. The results of simulation establish a connection between performance spread and the variation of parameters.

The Monte Carlo simulation is performed based on the assumption that all interconnect parameters (there are four parameters in this study) are normally distributed with 3 sigma equal to ±20 % of their nominal values. Figure 3.9 is the scatter plot of delay which provides the information about the worst and best corners and can be used for centering the design. Figures 3.10 to 3.13 show the sensitivity of delay to metal width, inter-wire spacing, ILD thickness and metal thickness respectively. These results are consistent with the previous deterministic analysis. Particularly, Figure 3.11 shows the significant sensitivity of inter-wire spacing with respect to the delay.

Interconnect variations lead to the change of interconnect resistance and capacitance, including both plate capacitance and coupling capacitance, and affect the delay of the circuit. Figures 3.14, 3.15, and 3.16 show how sensitive the delay is against unit-length inter-wire coupling capacitance, plate capacitance and resistance, respectively. Particularly, Figure 3.14 demonstrates the extreme importance of the coupling capacitance with regard to the delay.

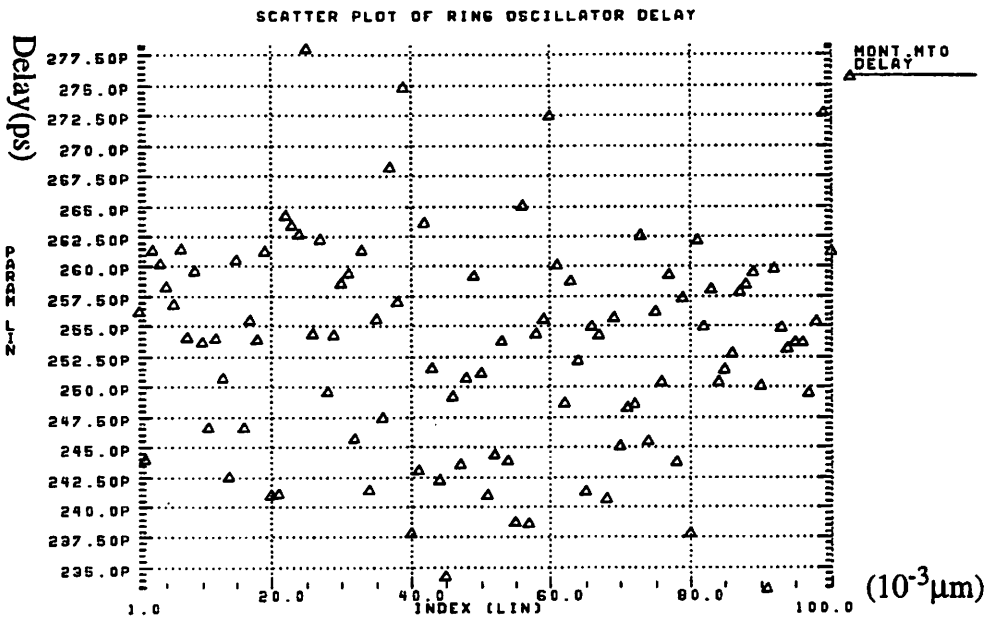


Figure 3.9.
Scatter plot of Monte Carlo simulation

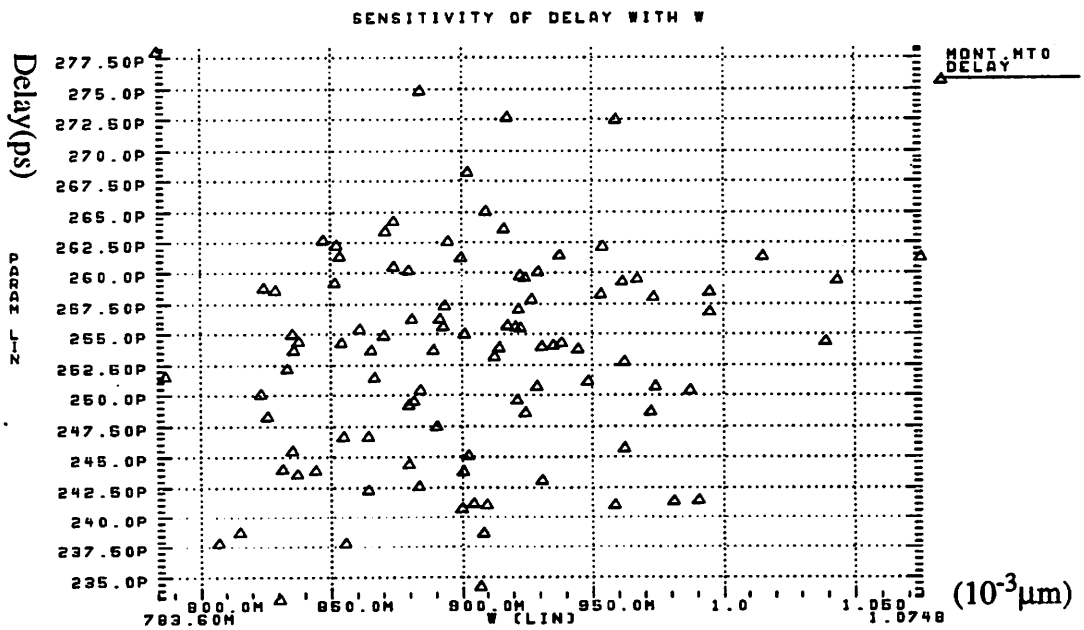


Figure 3.10.
Delay sensitivity to metal width

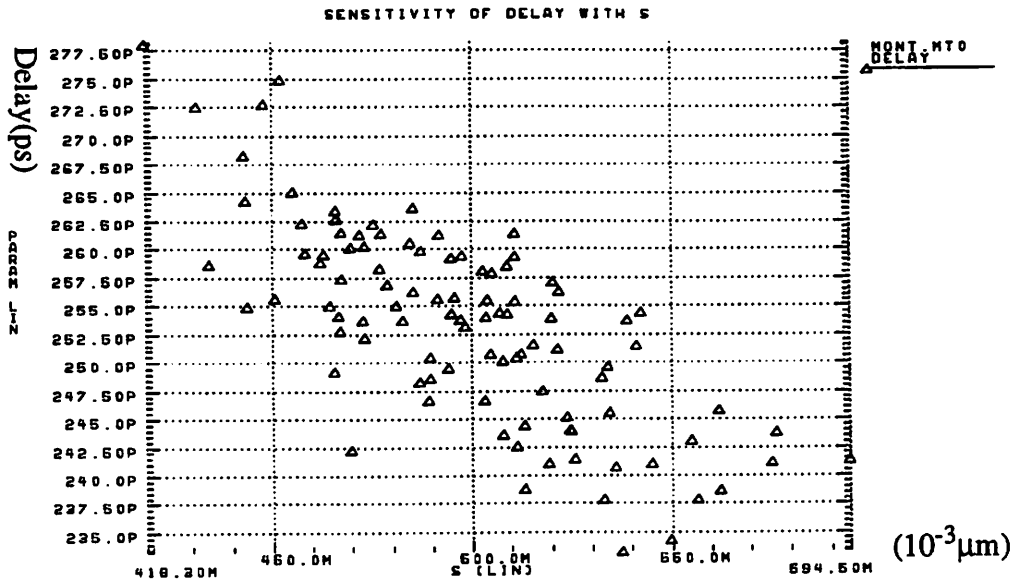


Figure 3.11.
Delay sensitivity to inter-wire spacing

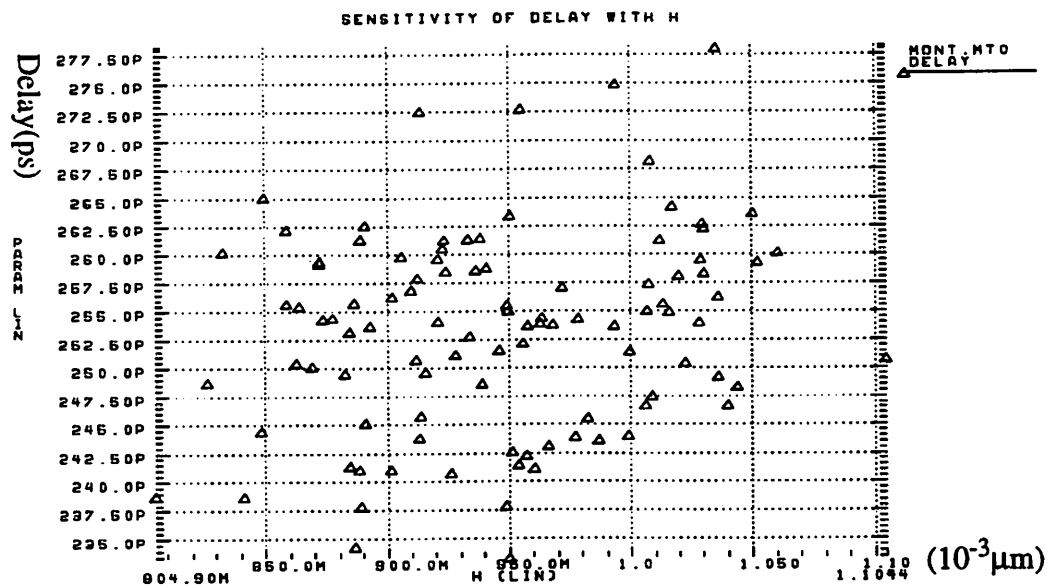


Figure 3.12.
Delay sensitivity to ILD thickness

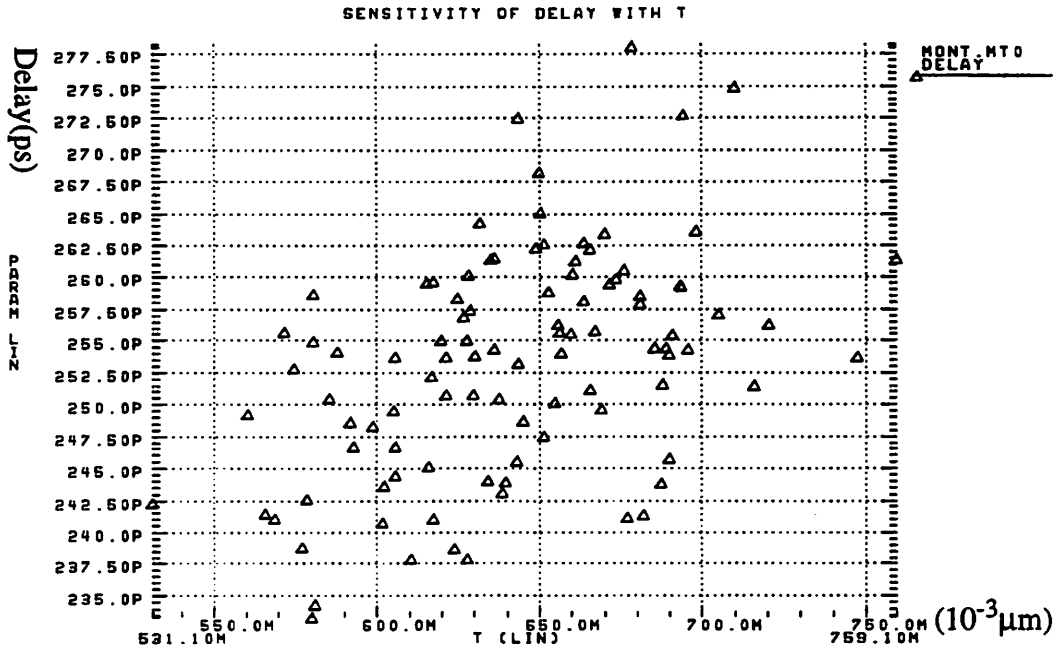


Figure 3.13.
Delay sensitivity to metal thickness

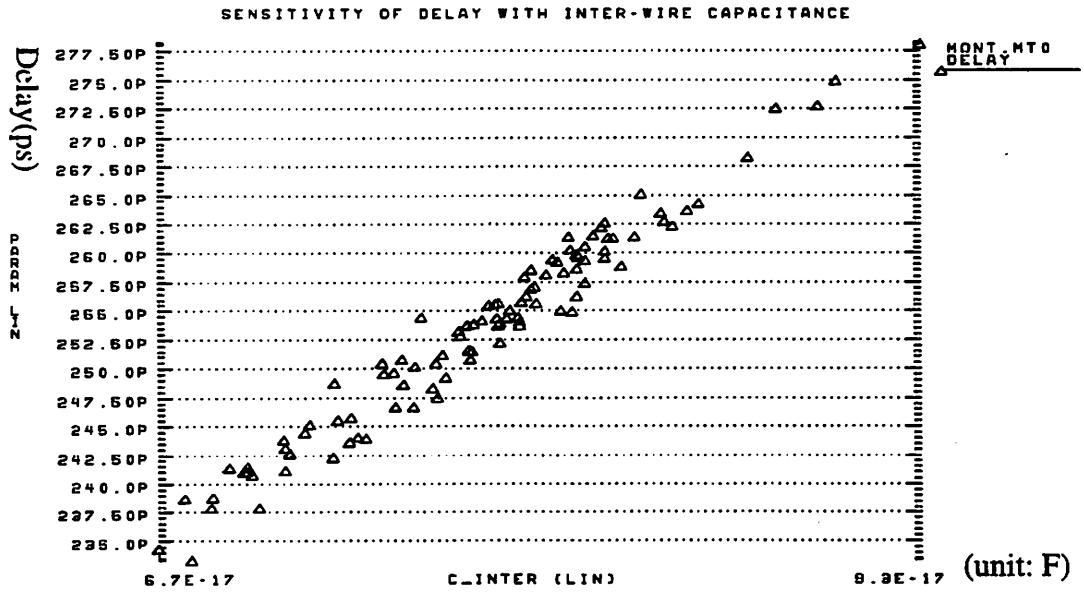


Figure 3.14.
Delay sensitivity to unit-length inter-wire coupling capacitance

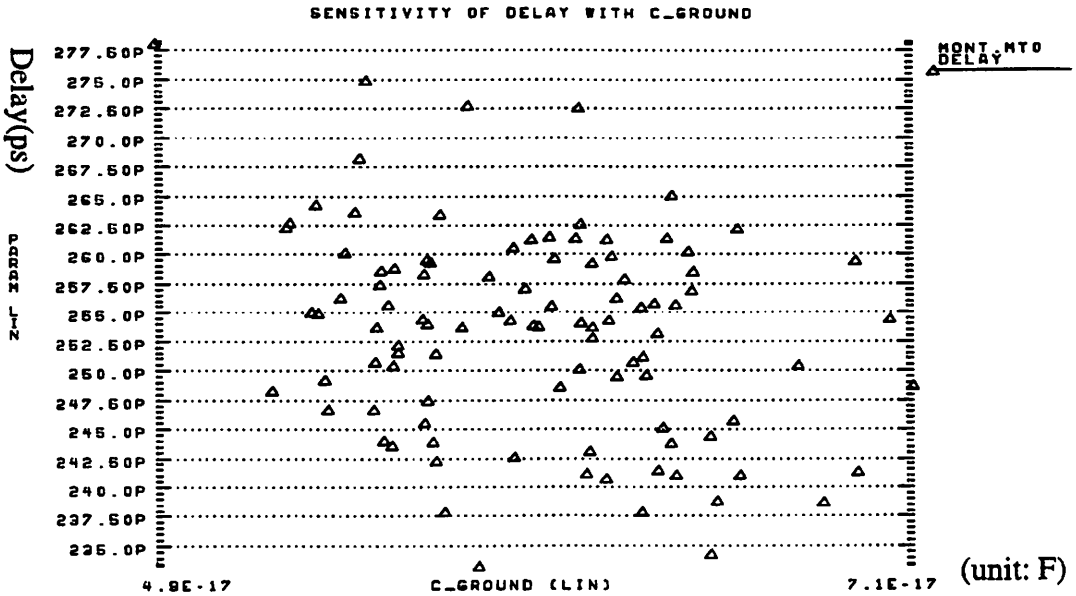


Figure 3.15.
Sensitivity of delay to unit-length plate capacitance

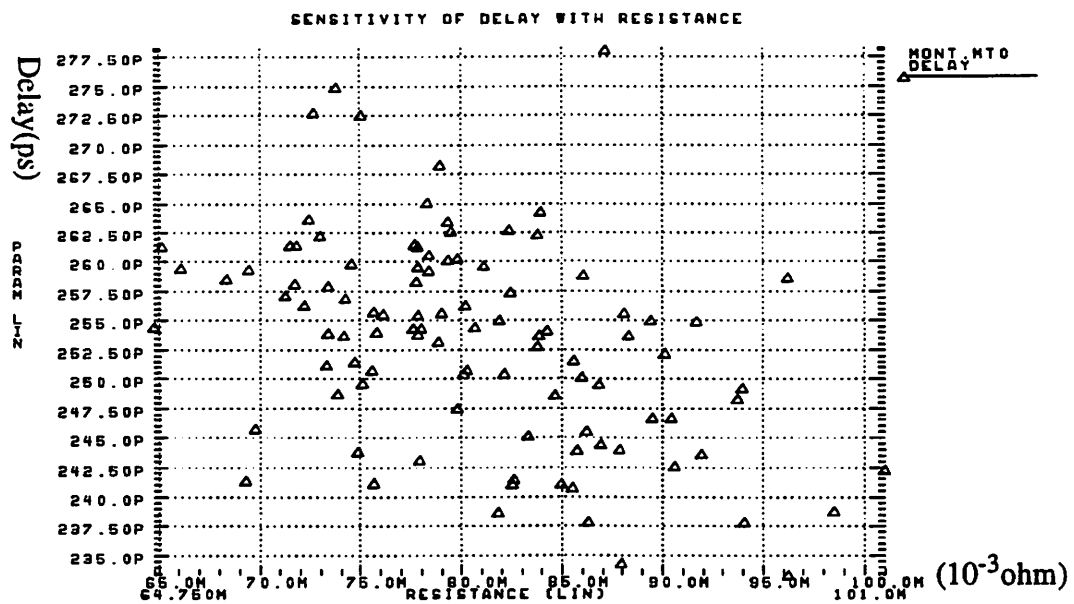


Figure 3.16.
Sensitivity of delay to unit-length wire resistance

3.6. Summary and Discussion

In this chapter, the issues related to statistical circuit design are discussed, and an approach to study circuit sensitivity to interconnect parameter variations is developed using parameterized interconnect model library. The circuit netlist is modified to include explicit parameterized expressions of interconnect parasitics as a function of layout parameter. The results from the study of a ring oscillator circuit reveal that the delay of this ring oscillator is most sensitive to inter-wire spacing while least sensitive to ILD thickness.

The sensitivity study results can not be generalized simply. More circuits of similar type need to be studied in order to collect enough data to reach a general conclusion. The emphasis of this work is to develop an approach rather than to look for a general sensitivity conclusion for a particular type of circuits. However, this does not limit the value of this approach, since very often we may only be interested in a particular circuit during the design process. In the next chapter we show how such an analysis could be carried out for a complex circuit that does not have the regularity of a ring oscillator.

Chapter 4 Sensitivity Study Using Statistical Experimental Design

4.1. Introduction

In Chapter 3, we developed an approach which uses parameterized interconnect models to study the circuit sensitivity to interconnect parameter variations. A ring oscillator circuit whose performance is mainly determined by interconnect wires was studied using this approach. All the interconnect wires of this circuit were modeled via closed form analytical models, that is, all the parasitic capacitors and resistors were described in terms of both interconnect technology parameters and layout parameters.

There were several advantages to this approach. Firstly, it made the sensitivity study much easier without going through the time-consuming and error-prone process of on-line whole chip circuit extraction. Secondly, when studying the effect of the spatially distributed variations, this approach will be a good candidate since interconnect wires can be modeled separately using different models at different positions. Thirdly, the sensitivity to circuit design or layout parameters can be evaluated easily via this approach. Fourthly, when studying a complicated large circuit such as a microprocessor, some simple circuits that closely resemble the statistics of a microprocessor circuit can be analyzed using the above approach. In such a way, we can evaluate and forecast the performance spread of the microprocessor resulting from interconnect parameter variations before the manufacturing of the product die.

However, there are some limitations to this approach. It requires manual construction of an RC model for each interconnect wire, so it is not very suitable for studying a complicated and irregular, circuit directly. It is inefficient to manually model the whole circuit.

So, an alternative approach is developed in this chapter to study the impact of process variations of interconnect technology parameters on circuit performance.

4.2. Methodology

The main idea is that: based on the variation ranges of the technology parameters, the technology file is revised, and different circuit description files are generated from the revised technology files. The circuit description files are HSPICE decks. They are fed into the circuit simulator to evaluate the performance of the circuit.

Since there are many parameters of interest in multi-layer interconnect technology, we use statistical experiment design techniques to carry out the computer simulation experiments. More specifically, the flow of this approach is shown in Figure 4.1 and is listed below:

a. Design the experiments with variables of interconnect parameters, and construct the design matrix.

b. Revise the technology file based on the design matrix for each designed experiment.

c. Extract the parasitics of the circuit from the layout with each revised technology file, and thus generate an HSPICE deck.

d. Convert the HSPICE deck to Epic compatible input file, and run Pathmill[1] to identify the critical paths and evaluate the delay of each critical path.

e. Perform statistical analysis based on the simulation results of the extracted critical paths.

This approach is suitable for studying large, complicated circuits. In this work, a 32-bit shift-and-add multiplier circuit is used as a study case.

1. Pathmill is a CAD tool from Epic Inc. The tool can identify the critical paths of a circuit given the source and sink nodes, and evaluate the delay of each path. The script file to generate the input file, a config file, and a script file to run Pathmill are attached in the Appendix.

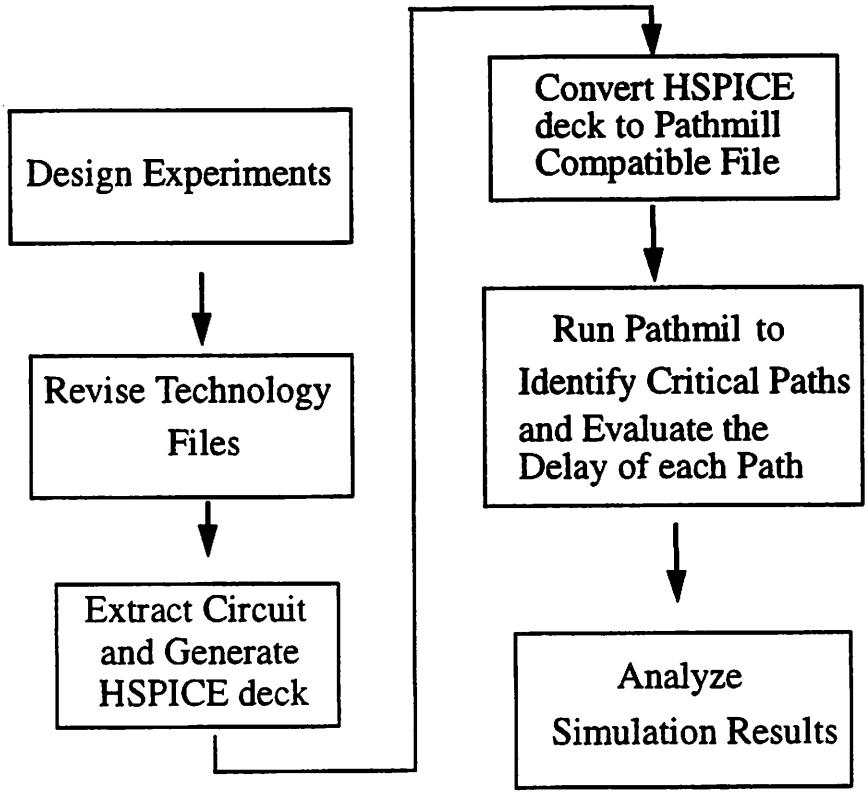


Figure 4.1.
The methodology of sensitivity study using statistical experimental design

From the layout and the Hspice deck, it is found that this multiplier circuit contains about 8000 transistors and occupies an area of about 230k micron^2. The technology for this circuit is 0.3 micron, and the interconnect wires have three metal layers and one poly layer, as shown in Figure 4.2. The variables of interest are listed below:

a. t_1, t_2 t_3, t_4 ------ thickness of poly, metal1, metal2 and metal3, respectively.

b. h_1------- Field oxide thickness.

c. h_2------- ILD thickness between poly and metal1.

d. h_3------- ILD thickness between metal1 and metal2.

e. h_4------- ILD thickness between metal2 and metal3.

It is also possible to examine the impact of layout parameters using this approach. To do this, however, will include a costly circuit extraction step for each of the simulated experiments. With today's CAD technology this will add several days of CPU time to a reasonably sized experiment.

4.3. Screening Experiment

4.3.1. Experimental Design

We want to investigate the most sensitive and important factors among the eight parameters stated above. This is achieved via a screening experiment. The range of each

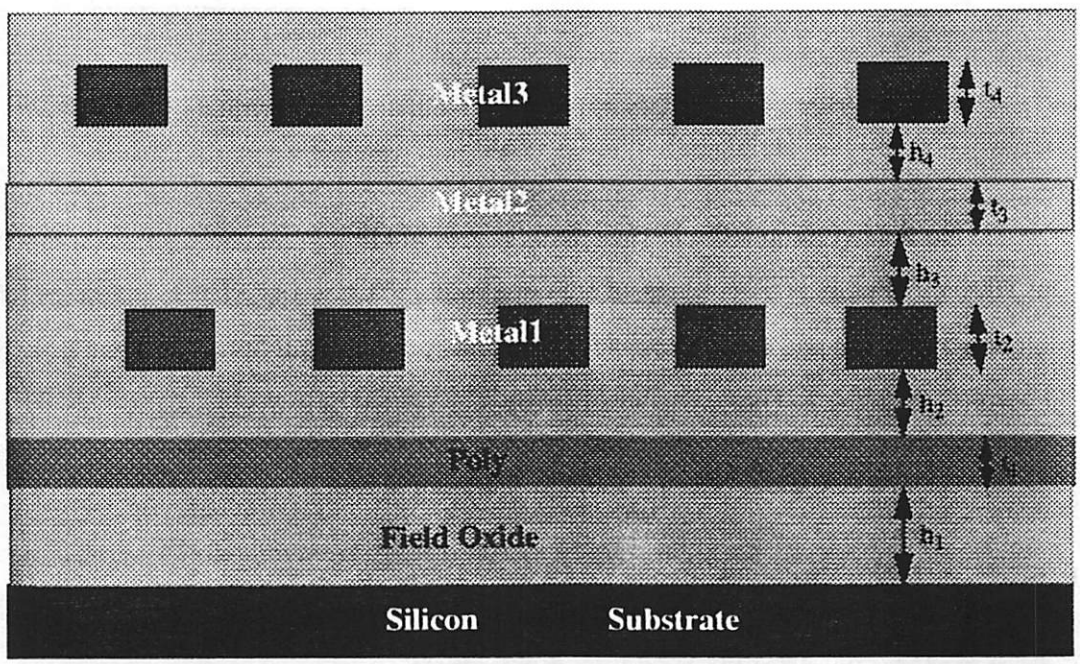


Figure 4.2.
Illustration of multi-layer interconnect structure

parameter was chosen to effectively encompass its possible variation range during regular production. A full factorial experiment to determine all effects and interactions for the eight factors would require 2^8, or 256 experiments. However, in order to reduce the experimental budget and simulation cost, the effects of higher order interactions were neglected and a 2^{8-4} fractional factorial design requiring only 16 runs was performed. The design matrix appears in Table 4.1.

Run	h_1	h_2	h_3	h_4	t_1	t_2	t_3	t_4
1	0.2520	0.4200	0.5600	1.0400	0.2600	0.8255	0.6545	1.2155
2	0.4680	0.4200	0.5600	0.5600	0.1400	0.8255	1.2155	1.2155
3	0.2520	0.7800	0.5600	0.5600	0.2600	0.4445	1.2155	1.2155
4	0.4680	0.7800	0.5600	1.0400	0.1400	0.4445	0.6545	1.2155
5	0.2520	0.4200	1.0400	1.0400	0.1400	0.4445	1.2155	1.2155
6	0.4680	0.4200	1.0400	0.5600	0.2600	0.4445	0.6545	1.2155
7	0.2520	0.7800	1.0400	0.5600	0.1400	0.8255	0.6545	1.2155
8	0.4680	0.7800	1.0400	1.0400	0.2600	0.8255	1.2155	1.2155
9	0.4680	0.7800	1.0400	0.5600	0.1400	0.4445	1.2155	0.6545
10	0.2520	0.7800	1.0400	1.0400	0.2600	0.4445	0.6545	0.6545
11	0.4680	0.4200	1.0400	1.0400	0.1400	0.8255	0.6545	0.6545
12	0.2520	0.4200	1.0400	0.5600	0.2600	0.8255	1.2155	0.6545
13	0.4680	0.7800	0.5600	0.5600	0.2600	0.8255	0.6545	0.6545
14	0.2520	0.7800	0.5600	1.0400	0.1400	0.8255	1.2155	0.6545
15	0.4680	0.4200	0.5600	1.0400	0.2600	0.4445	1.2155	0.6545
16	0.2520	0.4200	0.5600	0.5600	0.1400	0.4445	0.6545	0.6545

Table 4.1.
Design Matrix of Screening Experiment (unit: μm)

It can be seen from the table that the generators are: I = 1248, I = 2345, I = 1346, and I = 1237, so the resolution of the design is IV. The main effects are confounded with three-factor interactions, and two-factor interactions are confounded with another two-factor interaction. Since we expect that the three-factor interactions will not be significant,

the design should give us the correct results for the main effects. Also, since the experiments are actually computer simulations, randomization of the run sequence is not necessary.

4.3.2. Screening Experiment Results and Discussion

Analysis of the screening experiment results revealed that only two of the eight variables have large effects[2] on the circuit performance: ILD thickness between poly and metal 1 and ILD thickness between metal1 and metal2. Table 4.2 provides a summary of the results.

Factor	Degrees of Freedom	Sum of Squares	Mean Square	F value	Pr(F)
h_1	1	0.013748	0.013748	0.3471	0.574
h_2	1	8.822385	8.822385	222.7478	0.000
h_3	1	1.707596	1.707596	43.1134	0.000
h_4	1	0.000371	0.000371	0.0094	0.925
t_1	1	0.005148	0.005148	0.1300	0.729
t_2	1	0.190751	0.190751	4.8161	0.064
t_3	1	0.001580	0.001580	0.0399	0.847
t_4	1	0.034503	0.034503	0.8711	0.381
Residuals	7	0.277249	0.039607		

Table 4.2.
Results of Screening Experiment

2. Since this is a computer simulated experiment, lacking experimental error, it is meaningless to talk about statistical significance. We used traditional ANOVA techniques for the analysis with the understanding that the residuals are the result of under modeling. The ANOVA was used to help us identify the important factors.

The above table indicates that only h_2 and h_3 have large effects on the response. The result also shows that interconnect wires play an important role in determining the critical path delay of this multiplier. This will be further analyzed in the subsequent sections. At this point, we can isolate the two variables and check whether the interaction of h_2 and h_3 has a large effect on the response. Table 4.3 summarizes the results of the new model which eliminates all the other six variables. The result shows that the interaction term is significant, but it is not as important as the main effects.

Since variation of circuit performance is mainly due to the variation of capacitance, the results show that h_2 and h_3 can explain most of the variations of the capacitance. This can be explained partly by the fact that most routing wires of interconnect are metal1 and metal2, especially metal1, which can be inspected from the layout. Though detailed distributions of total lengths of different metal layers are not available (due to the complexity of the circuit), the circuit loading will be analyzed in the next section in order to get more insight.

Factor	Degrees of Freedom	Sum of Squares	Mean Square	F value	Pr(F)
h_2	1	8.822385	8.822385	359.3319	0.000
h_3	1	1.707596	1.707596	69.5496	0.000
h_2h_3	1	0.228723	0.228723	9.3158	0.010
Residuals	12	0.294626	0.024552		

Table 4.3.
Significance Test Results of the Interaction Term, h_2h_3.

4.3.3. Circuit Loading Analysis

It should be noted that the above result is much circuit dependent, and even layout dependent to some extent. So different categories of circuits will exhibit different sensitivities to interconnect parameters. Even for the same circuit, the sensitivity analysis results may be different with different technologies, or with the same technology but different layouts. This is because the routing layers and length of each layer may be much different. So

the results from the analysis of one circuit can not be simply generalized for even the same style of circuits without further analysis of the statistics of the circuit. The results of the sensitivity study will be more helpful and useful when linked to a detailed analysis of the circuit loading distribution, such as gate capacitance, diffusion capacitance, capacitance contributed by interconnect, and even the capacitance associated with different metal layers.

To compute the total gate capacitance, diffusion capacitance (includes junction capacitance and side-wall capacitance) and interconnect capacitance, we started from the HSPICE deck, computed the relevant geometry and multiplied it by the capacitance per unit area or unit length. The C code to implement this function is attached in Appendix. Specifically, we computed the following terms from the HSPICE deck:

$$WL_{total} = \sum_{i=1}^{n} W_i L_i \tag{4.1}$$

$$AD_{total} = \sum_{i=1}^{n} AD_i \tag{4.2}$$

$$AS_{total} = \sum_{i=1}^{n} AS_i \tag{4.3}$$

$$PD_{total} = \sum_{i=1}^{n} PD_i \tag{4.4}$$

$$PS_{total} = \sum_{i=1}^{n} PS_i \tag{4.5}$$

$$C_{int} = \sum_{k=1}^{N} C_k \tag{4.6}$$

In the above equations, n is the total number of the transistors in the circuit, and N is the total number of interconnect capacitance elements that appear in the HSPICE deck. W_i is the width of the ith transistor, and L_i is the length of the ith transistor. AD_i and AS_i are the drain and source diffusion area of the ith transistor, respectively. PD_i and PS_i are the perimeters of drain and source of the ith transistor, respectively. C_k is the kth interconnect capacitance element appearing in the HSPICE deck. So, WL_{total} is the summation of gate area of all the transistors. AD_{total} is the summation of AD_i, and AS_{total} is the summation of AS_i of all transistors. PD_{total} and PS_{total} are the summation of PD_i and PS_i of all transistors, respectively. C_{int} is the summation of interconnect capacitances.

Note that gate capacitance and diffusion capacitance are computed implicitly by the internal algorithm of HSPICE. Since unit-area gate capacitance and unit-area diffusion capacitance are explicitly shown in the device model, the total gate capacitance and diffusion can be easily computed as below:

$$C_{gate} = C_g \cdot WL_{total} \tag{4.7}$$

$$C_{diffusion} = C_j \cdot (AD_{total} + AS_{total}) + C_{jsw} \cdot (PD_{total} + PS_{total}) \tag{4.8}$$

Where C_{gate}, C_j and C_{jsw} are unit area gate capacitance, bottom junction capacitance and side-wall capacitance, respectively.

Using the above formulae, the loading distribution of this circuit in the nominal case was computed and listed in Table 4.4.

	Percentage of Loading
Interconnect Capacitance	75%
Gate Capacitance	13%
Junction Capacitance	12%

Table 4.4.
Capacitance loading distribution of the multiplier in the nominal case.

The above table indicates that the interconnect dominates the loading of the circuit, and is in agreement with the screening experimental result.

The above calculation of loading distribution is relatively rough, since some components of capacitance, such as miller capacitance, are not included. Also, the way of summation may not accurately reflect their effects on the circuit performance. However, it does provide us with some insight about the experiment results.

4.4. Second-phase Experiment Design

The loading distribution analysis does not provide accurate sensitivity information with regard to gate capacitance and diffusion capacitance. Based on the results of the screening experiment, a second experiment is designed which takes both device and interconnect variations into consideration. There are four variables in the design: C_{gate}, $C_{diffusion}$, h_2 and h_3. A full factorial design would need 16 runs, so a 2^{4-1} fractional factorial design with 8 runs was used. The design matrix appears in Table 4.5.

Note that $C_{diffusion}$ is the summation of bottom junction capacitance and side-wall capacitance. They were treated as a single factor since they are highly correlated.

Table 4.6 reveals the significant effect of h_2, h_3 and C_{gate} on circuit performance, among which the effect of h_2 is the most prominent. The result is consistent with that of the screening experiment and the circuit loading analysis.

4.5. Central Composite Design and Model Building

Recall that the goal is to understand the impact of the variations of interconnect related technology parameters on circuit performance. We are interested to investigate how these parameters will affect the interconnect capacitance, and how the interconnect capac-

Run	h_2 (μm)	h_3 (μm)	C_{gate} (F/m^2)	C_j (F/m^2)	C_{jw} (F/m)
1	0.42	0.56	0.0004485	3.277e-5	3.289e-08
2	0.78	0.56	0.0004485	1.764e-5	1.770e-08
3	0.42	1.04	0.0004485	1.764e-5	1.770e-08
4	0.78	1.04	0.0004485	3.277e-5	3.289e-08
5	0.42	0.56	0.0002415	1.764e-5	1.770e-08
6	0.78	0.56	0.0002415	3.277e-5	3.289e-08
7	0.42	1.04	0.0002415	3.277e-5	3.289e-08
8	0.78	1.04	0.0002415	1.764e-5	1.770e-08

Table 4.5.
Design Matrix of Second-phase Experiment

Factor	Degree of Freedom	Sum of Square	Mean Square	F value	Pr(F)
h_2	1	1.786995	1.786995	42.20239	0.007
h_3	1	1.079715	1.079715	25.49898	0.014
C_{gate}	1	0.627760	0.627760	14.82543	0.030
$C_{diffusion}$	1	0.193131	0.193131	4.56106	0.122
Residuals	3	0.127030	0.042343		

Table 4.6.
Screening of the Main Factors

itance relates to circuit performance. So in next section, we will build models to link the parameter variations with circuit performance.

In order to obtain the model, it is necessary to augment the data gathered with 7 additional runs which employed a Central Composite Design. In this design, the two-level factorial "box" was enhanced by further experiments at the center as well as symmetrically located "star" point [27]. These additional seven runs are listed in Table 4.7.

Combine the results of 15 runs(Table 4.5 and Table 4.8), the regression model is fitted:

$$Delay = 1.686975/h_2 + 1.309948/h_3 + 5400.86C_{gate}$$ (h_2, h_3: μm, C_{gate}: F/M^2)

Based on the above regression model, the sensitivity of delay to each parameter is calculated and listed in Table 4.7.

Parameter	h_2 (μm)	h_3 (μm)	C_{gate} (F/m^2)
Variation range	0.54 - 0.66	0.72 - 0.88	4.8607 - 59401.43
Impact on delay	8.99%	5.24%	5.9%

Table 4.7.
Delay Sensitivity to Main Factors

Run	h_2 (μm)	h_3 (μm)	C_{gate} (F/m^2)	C_j (F/m^2)	C_{jw} (F/m)
1	0.6	0.8	3.45e-4	2.5211e-05	2.5303e-08
2	0.6	0.32	3.45e-4	2.5211e-05	2.5303e-08
3	0.6	1.44	3.45e-4	2.5211e-05	2.5303e-08
4	0.24	0.8	3.45e-4	2.5211e-05	2.5303e-08
5	0.96	0.8	3.45e-4	2.5211e-05	2.5303e-08
6	0.6	0.8	1.035e-4	2.5211e-05	2.5303e-08
7	0.6	0.8	5.515e-4	2.5211e-05	2.5303e-08

Table 4.8.
Additional "Star Point" Recipes

The data transformation of h_2 and h_3 in the above model is suggested by physical intuition. The ANOVA table for the model is shown in Table 4.9.

Table 4.9 reveals the goodness of fit of the model, and it shows that the model can explain up to 99% of the variations of the delay. Table 4.10 shows that each term contributes to the good fit.

Source	Sum of Squares	Degree of Freedom	Mean Squares	F-ratio	Prob>F
Model	699.3331	3	1.938376	549.2	0
Error	0.0054	12	3.158eE-6	$R^2 = 0.9928$	
Total	91.109	15			

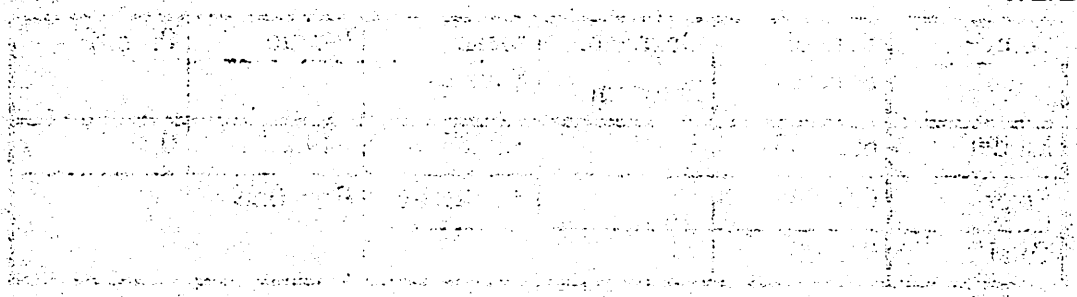
Table 4.9.
ANOVA table of regression model

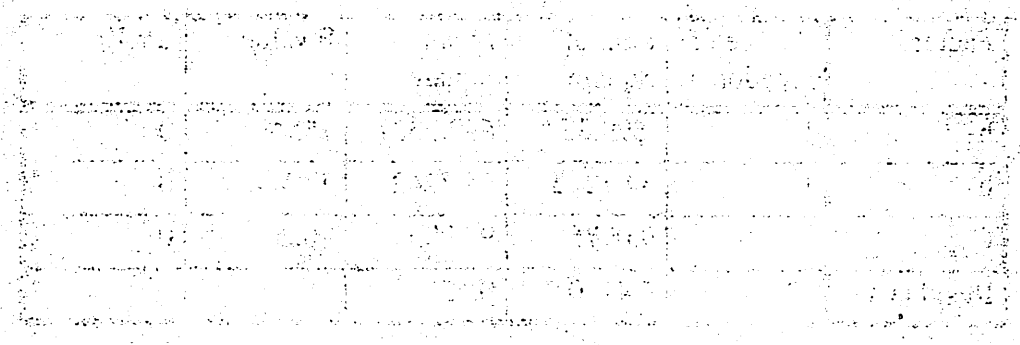
Factor	Degree of Freedom	Sum of Square	Mean Square	F value	Pr(F)
h_2	1	699.3331	699.3331	1554	0
h_3	1	34.7242	34.7242	77.164	0
C_{gate}	1	9.0975	9.0975	20.2	0
Residuals	12	5.4130	0.45		

Table 4.10.
Significance Test Result of Main Factors

4.6. Conclusion

In this chapter, we developed an approach using statistical design techniques to study the effects of interconnect parameter variations on the performance of a large, complicated circuit. With two experiments, the most significant factors are isolated, and the model is fitted via a Central Composite Design. The results from the case study of a shift-and-add multiplier revealed the significance of ILD thickness. The loading distribution of the circuit was also analyzed and correlated with the results.





Chapter 5 Conclusion

5.1. Summary

The main goal of this thesis is to present interconnect modeling techniques and develop approaches to study the circuit sensitivity to interconnect parameter variations. In this work, two different approaches are developed and are explored with a ring oscillator and a multiplier circuit, respectively.

In Chapter 2, we discussed interconnect modeling issues in detail and presented a methodology to build an interconnect model library.

The first approach presented in Chapter 3 is based on the parameterized interconnect model library. This approach can capture the effects of both layout and technology parameter variations. This approach is suitable for studying spatially distributed variation effects. A ring oscillator circuit was studied using this approach. The limitation of this approach is its inefficiency to study a complicated real circuit unless an automatic method can be found to pick up the right model for each interconnect wire.

In Chapter 4, we developed another approach which uses statistical design techniques. This approach is suitable for the sensitivity study of a large and complicated circuit. A multiplier circuit is studied using this approach. The disadvantage of this approach is that it requires multiple time-consuming circuit extraction steps.

An important point is that in order to make a general conclusion for one category of circuits, a reasonably large number of circuits must be studied. Since these circuits must have similar characteristics, one must attempt a meaningful taxonomy of like circuits. The

conclusion from the result of one circuit will be meaningful for the same family of circuits only if one is able to define such a family.

5.2. Future Work

A possible direction of future work would be to make the process of sensitivity study automatic without much manual work. This can be extended from the first approach discussed in Chapter 3. The main challenge is to generate the circuit description more efficiently, or even automatically, and describe the interconnect in a way suitable for sensitivity study.

Also, integration with variation models in the study will make the sensetivity study results more convincing.

Another direction of this work would be to study the problem from a higher level of the design flow, such as the logic level.

References

[1] Gail Anderson, Paul Anderson, *The UNIX™ C Shell Field Guide*, Englewood Cliffs, NJ: Prentice-Hall, 1986.

[2] Crid Yu, *Integrated Circuit Process for Manufacturability Using Enhanced Metrology*, Ph.D thesis, ERL, Univeristy of California at Berkeley, 1995.

[3] G. Coatache, "Finite element method applied to skin-effect problems of instrip transmission lines," *IEEE Tran. on Microwave Theory and Technology,* Nov. 1987.

[4] A. Zemanian, "A finite-difference procedure for the exterior problem inherent in capacitance computations for vlsi interconnections," *IEEE Tran. on Eectron Devices,* July 1988.

[5] A. Zemanian and R. Tewarson, " Three-dimensional capacitance computations for VLSI/ULSI interconnections," *IEEE Tran. on CAD,* CAD-8, pp. 1319-1326, Dec 1989.

[6] Brian W. Kernighan, Dennis M. Ritchie, *The C Programming Language*, 2nd ed., Englewood Cliffs, NJ: Prentice Hall, 1988.

[7] Douglas C. Montgomery, *Introduction to Statistical Quality Control*, 2nd. ed., NY: John Wiley & Sons, 1985.

[8] William H. Press, Saul A. Teukolsky, William T. Vetterling, Brian P. Flannery, *Numerical Recipes in C*, 2nd ed., Cambridge University Press, 1992.

[9] Bjarne Stroustrup, *The C++ Programming Language*, 2nd ed., Addison-Wesley.

[10] S. Y. Oh, "Interconnect Modeling and Design in High-Speed VLSI/ULSI Systems",

[11] C. Mead and L. Conway, *Introduction to VLSI Systems,* Addison-Wesley, 1980.

[12] P. Yang, D. Hocever, P.Cox, C.Machala and P. Chatterjee, "An Integrated and Efficient Approach for MOS VLSI Statistical Circuit Design," *IEEE Trans. on CAD*, Vol. CAD-5, pp 5-14, Jan. 1986.

[13] A. E. Ruehli, "Circuit Analysis, Simulation and Design", Elsevier Science Publishers, 1987.

[14] W. Maly, A.J.Strojwas and S.W. Director, "VLSI Yield Prediction and Estimation: A Unified Framework," *IEEE Trans. on CAD*, Vol. CAD-5, No. 1, Jan. 1986.

[15] A. J. Strojwas, editor, Selected Papers on Statistical Design of Integrated Circuits, *IEEE Press*, 1987.

[16] E. D. Boskin, *A Methodology For Modeling the Manufacturability of Integrated Circuits*, Ph.D thesis, ERL, Univeristy of California at Berkeley, 1995.

[17] D.E. Hocevar, P.F. Cox and P. Yang, "Computing parametric yield accurately and efficiently," *Proc. ICCAD*, 1990, pp. 116-119.

[18] J.C. Zhang and M.A. Styblinski, "Deisgn of Experiments Approach to Gradient Estimation and its Application to CMOS Circuit Stochastic Optimization," *Proc. ISCAS*, Singapore, pp 3098-3101, 1991.

[19] Z. Daoud, DORIC: *Design of Optimized and Robust Integrated Circuit*, Mater's Thesis, UC Berkeley, December 1993.

[20] R. Spence, R. S. Soin, *Tolerance Design of Electronic Circuits, Addison-Wesley, 1988.*

[21] H. B. Bakoglu, *Circuits, Interconnections, and Packaging for VLSI*, Addison-Wesley, 1990.

[22] R. H. Dennard et al., "Design of ion implanted MOSFETs with Very Small Physical Dimensions," *IEEE Journal of Solid-State Circuits*, Vol. SC-9, Oct. 1974.

[23] K. Nabors, "FastCap: A multiple accelerated 3-D capacitance extraction program", *IEEE Trans. Computer-Aided Design*, vol 10, pp.1447-1459, Nov. 1991.

[24] U. Choudhury et al., " An Analytical-model generator for interconnect capacitances," in *Proc. IEEE CICC*, 1991.

[25] K. J. Chang, S.Y. Oh, and K. Lee, "HIVE: An efficient interconnect capacitance extractor to support submicron multilevel interconnect designs," in *Proc. ICCAD*, 1991.

[26] TMA Inc., *Raphael User's Manual*, 1996.

[27] G. E. P. Box, W.B. Hunter, and J.S.Hunter, *Statistics of Experimenters*, New York: Yiley, 1978.

[28] Using FrameMaker, Release 5, Frame Technology Corporation, May 1995.

[29] http://cas.et.tudelft.nl/~space/

Appendix A

Script to convert Hspice file to Epic compatible input file and the command to run pathmill:

```
#! /bin/csh -f
# runpm (to run pathmill)
set ckt=$2
spice2e -u m -i $ckt.sp.ext -o $ckt.ntl.ext -f hspice  >& /dev/null
set tech=/net/test0/disks/test0/a/cns/cad/epic/3.4.1/techfile/cmos14tb_3_3V.tech
switch ($1)
        case b:
                echo "pathmill -n $ckt.ntl.ext -o base -c $ckt.cfg -p $tech"
                     pathmill -n $ckt.ntl.ext -o base -c $ckt.cfg -p $tech
        breaksw
        case i:
                echo "pathmill -n $ckt.ntl.ext -o base -c $ckt.cfg -p $tech -i"
                     pathmill -n $ckt.ntl.ext -o base -c $ckt.cfg -p $tech -i
        breaksw
        case cl:
                rm base* full* pathmill*
                rm critical.*
                       echo "directory cleaned up"
        breaksw
        default:
                echo "Usage: runpm [option] file"
                echo ""
                echo "option choices:"
                echo "b - run batch pathmill timing"
                echo "i - run interactive pathmill timing"
                echo "cl - clears demo"
        breaksw
endsw


1.1 Config file:

 xfer_all_pair
search_mux
```

```
report_paths critical max 20
print_spice_paths critical 1
source_node  A31 A30 A29 A28 A27 A26 A25 A24 A23 A22 A21 A20 A19 A18 A17 A16
A15 A14 A13 A12 A1  A10 A9 A8 A7 A6 A5 A4 A3 A2 A1 A0  .....
sink_node  O31 O30 O29 O28 O27 O26 O25 O24 O23 O22 O21 O20 O19 O18 O17 O16 O15
O14 O13 O12 O11  O10 O9 O8 O7 O6 O5 O4 O3 O2 O1 O0
```

Appendix B

C code to calculate the loading of the circuit from Hspice deck

```
#include <stdio.h>
#include <fcntl.h>
#include <stdlib.h>
#include <unistd.h>
#include <ctype.h>
#include <iostream.h>
#define MAXSIZE_buf 50
#define MAXSIZE 20000
#define POS 4
 int deBug =0;
 long double sum_up(long double * buf_array, int num);
void print_array(long double * buf_array, int num);

void printf_usage(void) {
    printf("Usage: a.out <file_name> <choice>\n ");
    printf("choice =1 is to calculate sum of WL, AD, AS, PD and PS\n ");
    printf("choice =2 is to calculate capacitance\n");
}

main(int argc, char ** argv) {

    int f1;
    int l_num =0;
    int ad_num =0, as_num =0, pd_num=0, ps_num=0;
    char buf[MAXSIZE_buf]; //temp buffer
    char * bufptr = buf;
    long double l_val[MAXSIZE];
    long double ad_val[MAXSIZE];
    long double as_val[MAXSIZE];
    long double pd_val[MAXSIZE];
    long double ps_val[MAXSIZE];
    char lvaltmp[MAXSIZE_buf];
    char * lvaltmpptr = lvaltmp;

    if(argc != 3) {
       printf_usage();
       exit(1);
```

```
    }
    int choice = atoi(argv[2]);
    if(choice != 1 && choice != 2) {
       printf("Must choose either 1 or 2 for your choice\n");
       printf_usage();
       exit(1);
    }
    if((f1=open(argv[1], O_RDONLY, 0)) ==-1) {
       printf("Error: Sorry! can't open %s\n", argv[1]);
       exit(1);
    }

    if (choice ==1) {
       while(read(f1, bufptr, 1) >0) {
           lvaltmpptr = lvaltmp;
           //L:
           if(*bufptr == 'L') {
              if(read(f1, bufptr, 1) >0 && *bufptr == '=') {
                  while(read(f1, bufptr, 1)>0 && !isspace(*bufptr)) {
                     *lvaltmpptr = *bufptr;
                     lvaltmpptr++;
                  }
                  *lvaltmpptr ='\0';
                  if(deBug) printf("L string: %s\n", lvaltmp);
                  l_val[l_num]= atof(lvaltmp);
                  if(deBug) printf("%d\n", atoi(lvaltmp)); if(deBug)
printf("%e\n", atof(lvaltmp));
              }
           }
           //W:
           else if(*bufptr == 'W') {
              if(read(f1, bufptr, 1) >0 && *bufptr == '=') {
                  while(read(f1, bufptr, 1)>0 && !isspace(*bufptr)) {
                     *lvaltmpptr = *bufptr;
                     lvaltmpptr++;
                  }
                  *lvaltmpptr ='\0';
                  if(deBug) printf("W string: %s\n", lvaltmp);
                  l_val[l_num]= atof(lvaltmp) * l_val[l_num];
                  l_num++;
                  if(deBug) printf("%d\n", atoi(lvaltmp)); if(deBug)
printf("%e\n", atof(lvaltmp));
                  if(deBug) printf("product: %e\n", l_val[l_num-1]);
              }
           }

           else if(*bufptr == 'A') {
              //AD:
              if(read(f1, bufptr, 1) >0 && *bufptr == 'D' &&
                 read(f1, bufptr, 1) >0 && *bufptr == '=') {
                  while(read(f1, bufptr, 1)>0 && !isspace(*bufptr)) {
                     *lvaltmpptr = *bufptr;
                     lvaltmpptr++;
                  }
                  *lvaltmpptr ='\0';
                  if(deBug) printf("AD string: %s\n", lvaltmp);
                  ad_val[ad_num]= atof(lvaltmp);
                  ad_num++;
```

```
                if(deBug) printf("%d\n", atoi(lvaltmp)); if(deBug)
printf("%e\n", atof(lvaltmp));

            }
            //AS:
            else if(*bufptr == 'S' &&
                  read(f1, bufptr, 1) >0 && *bufptr == '=') {
               while(read(f1, bufptr, 1)>0 && !isspace(*bufptr)) {
                  *lvaltmpptr = *bufptr;
                  lvaltmpptr++;
               }
               *lvaltmpptr ='\0';
               if(deBug) printf("AS string: %s\n", lvaltmp);
               as_val[as_num]= atof(lvaltmp);
               as_num++;
               if(deBug) printf("%d\n", atoi(lvaltmp)); if(deBug)
printf("%e\n", atof(lvaltmp));

            }
         }
         else if(*bufptr == 'P') {
            //PD:
            if(read(f1, bufptr, 1) >0 && *bufptr == 'D' &&
               read(f1, bufptr, 1) >0 && *bufptr == '=') {
               while(read(f1, bufptr, 1)>0 && !isspace(*bufptr)) {
                  *lvaltmpptr = *bufptr;
                  lvaltmpptr++;
               }
               *lvaltmpptr ='\0';
               if(deBug) printf("PD string: %s\n", lvaltmp);
               pd_val[pd_num]= atof(lvaltmp);
               pd_num++;
               if(deBug) printf("%d\n", atoi(lvaltmp)); if(deBug)
printf("%e\n", atof(lvaltmp));

            }
            //PS:
            else if(*bufptr == 'S' &&
                  read(f1, bufptr, 1) >0 && *bufptr == '=') {
               while(read(f1, bufptr, 1)>0 && !isspace(*bufptr)) {
                  *lvaltmpptr = *bufptr;
                  lvaltmpptr++;
               }
               *lvaltmpptr ='\0';
               if(deBug) printf("PS string: %s\n", lvaltmp);
               ps_val[ps_num]= atof(lvaltmp);
               ps_num++;
               if(deBug) printf("%d\n", atoi(lvaltmp)); if(deBug)
printf("%e\n", atof(lvaltmp));

            }
         }
      }

      //calculate Sum of WL, AD, AS, PD, PS:
      if(deBug) print_array(l_val,l_num);
      if(l_num ==0)
          printf("No W or L values found.\n");
      else
```

```
        printf("Sum of WL is %e\n", sum_up(l_val, l_num));
    if(ad_num ==0)
        printf("No AD values found.\n");
    else
        printf("Sum of AD is %e\n", sum_up(ad_val, ad_num));
    if(as_num==0)
        printf("No AS values found.\n");
    else
        printf("Sum of AS is %e\n", sum_up(as_val, as_num));
    if(pd_num==0)
        printf("No PD values found.\n");
    else
        printf("Sum of PD is %e\n", sum_up(pd_val, pd_num));
    if(ps_num==0)
        printf("No PS values found.\n");
    else
        printf("Sum of PS is %e\n", sum_up(ps_val, ps_num));

    close(f1);
  }


  //Calculate capacitance:
  else {
    long double c_val[MAXSIZE];
    int c_num = 0;
    int position = 0;
    while(read(f1, bufptr, 1) >0) {
        position = 1;
        lvaltmpptr = lvaltmp;
        //read each line, extract POS th element:
        while(*bufptr!='\n') {
          if(position == POS) {

              //skip extra space in the beginning:
              while(read(f1, bufptr, 1)>0 && isspace(*bufptr))
                 ;
              if(!isspace(*bufptr))
                 *lvaltmpptr++ = *bufptr;
              while(read(f1, bufptr, 1)>0 && !isspace(*bufptr)) {
                 *lvaltmpptr = *bufptr;
                 lvaltmpptr++;
              }
              *lvaltmpptr ='\0';
              if(deBug) printf("C string: %s\n", lvaltmp);
              c_val[c_num]= atof(lvaltmp);
              if(deBug) printf("%d\n", atoi(lvaltmp)); if(deBug)
printf("%e\n", atof(lvaltmp));
              position++; c_num++; continue;
          }
          else{
              while(read(f1, bufptr, 1)>0 && !isspace(*bufptr)) ;
              position++;
          }
        }
    }
    if(deBug) print_array(c_val, c_num);
    if(c_num==0)
        printf("No capacitance found.\n");
```

```
        else
            printf("Sum of %d C is %e\n", c_num, sum_up(c_val, c_num));
        close(f1);
    }//else
}

void print_array(long double * buf_array, int num) {
    printf("array size: %d\n", num);
    for(int i = 0; i<num; i++) {
       printf("%e, ", buf_array[i]);
    }
    printf("\n");
}

long double sum_up(long double * buf_array, int num) {

    long double result =0.0;
    for(int i = 0; i<num; i++) {
       result +=buf_array[i];
    }
    return result;
}
```

Appendix C

Circuit Description of Ring Oscillator

```
*input and voltage sources
Vdd vdd 0 2.7
.options co=80 cptime=20000 ingold=2 aspec dv=10000 list=1 TNOM=25
.options NOMOD ABSMOS=1.0E-11A ABSI=1.0E-15A NOWARN
.option post
.option AUTOSTOP
.option MEASOUT
.option LIST
.option OPLIST=1
.option RMIN=1.0e-13
.option IMAX=80



.PC BRIEF

* C_finger is the 1/4 of total ground capacitance of one finger.
*C_couple is one fourth of total coupling capacitance. This is used in Pai-two
model.
*The units of re_m1 abd re_poly is ohm*micron=ohm*meter*e6

.DATA no_idea w1 s1 h1 t1

+ 0.8917563 0.5221378 0.9020851 0.7201508
+ 0.8314726 0.5248488 0.9991259 0.6342517
+ 1.0153535 0.4775299 0.9335106 0.6353313
+ 0.8799255 0.4731690 0.8332002 0.6602766
+ 0.9538009 0.4771178 1.0305190 0.6249106
+ 0.9954437 0.4790919 0.9102252 0.6271164
+ 0.9379111 0.4675249 0.9382028 0.6362858
+ 0.9355017 0.4667678 0.9210991 0.5881973
+ 0.9246677 0.4586467 0.9211457 0.6176872
+ 0.8658956 0.4730288 0.9935074 0.6057444
+ 0.8550219 0.5033504 1.0402076 0.6054858
+ 0.9310345 0.5036487 1.0284207 0.6569016
+ 0.9741649 0.5122836 1.1044229 0.6296159
+ 0.8840876 0.5114837 0.9513213 0.5787401
+ 0.8743103 0.4845691 0.9229710 0.6759014
+ 0.8641794 0.4891369 1.0061622 0.5931395
+ 0.9228918 0.5041463 0.9491762 0.6598040
```

```
+ 0.9149374 0.5200395 0.9627820 0.6900300
+ 1.0747631 0.4920658 1.0122807 0.6614316
+ 0.9589540 0.5192873 0.8887054 0.5688981
+ 0.9048981 0.5751155 0.9605021 0.6772162
+ 0.8741582 0.4435069 1.0175249 0.6318292
+ 0.8715277 0.4661557 0.9508814 0.6701363
+ 0.8476327 0.4665086 1.0300021 0.6637712
+ 0.7835601 0.4182697 1.0355751 0.6784084
+ 0.8379300 0.5070392 0.9637785 0.6855530
+ 0.8522240 0.4574727 1.0304158 0.6490815
+ 0.9218800 0.5339630 0.9157754 0.6691841
+ 0.8540682 0.5089209 0.8740384 0.6956623
+ 0.8295268 0.4343348 0.9367255 0.5805970
+ 1.0435429 0.4975167 0.8726432 0.6718035
+ 0.9627138 0.5617925 0.9138834 0.6901030
+ 0.8535260 0.5108785 0.8893709 0.7591139
+ 0.9907192 0.5258847 0.9538826 0.5660364
+ 0.9209197 0.4509142 1.0137399 0.5714684
+ 0.8908194 0.5180432 1.0088803 0.6516517
+ 0.9029767 0.4433561 1.0084647 0.6500955
+ 0.9224700 0.5203898 0.9720091 0.7052973
+ 0.8848372 0.4523477 0.9944207 0.7099907
+ 0.8069106 0.5329890 0.9487354 0.6106426
+ 0.9313617 0.5550781 0.9660230 0.6386648
+ 0.9165230 0.4852519 1.0507315 0.6983116
+ 0.7870019 0.5156851 0.9463055 0.6877613
+ 0.8643605 0.4696992 0.9573080 0.5311269
+ 0.9070147 0.5496496 0.8869324 0.5811470
+ 0.8800161 0.4655948 1.0361314 0.5603562
+ 0.8371379 0.5077688 0.9871336 0.6021125
+ 0.9292139 0.5047424 0.9117219 0.6215206
+ 0.8521266 0.4950489 0.8722474 0.6940449
+ 0.9488851 0.5216005 0.9281859 0.6659744
+ 0.9096822 0.5452756 0.9013380 0.6173544
+ 0.8803263 0.5132141 0.8486108 0.6058607
+ 0.9447274 0.4971294 0.9681247 0.6304846
+ 0.8443738 0.5249684 0.9772291 0.6398642
+ 0.8155145 0.5131178 0.8048713 0.5769195
+ 0.9094519 0.4552890 0.8497864 0.6507151
+ 0.9081614 0.5624789 0.8412818 0.6239283
+ 1.0391348 0.5423510 0.8777805 0.6889920
+ 0.8936000 0.4914890 0.8587014 0.6563577
+ 0.9593051 0.4314931 0.9139968 0.6436558
+ 0.9301022 0.4699993 1.0606242 0.6284102
+ 0.9727428 0.5325628 0.8268246 0.6452869
+ 0.8245712 0.4623502 0.9412781 0.6532268
+ 0.8336046 0.4733132 0.9560272 0.6168349
+ 0.9810727 0.5944854 0.8847276 0.6818505
+ 0.8358412 0.4646143 1.0069672 0.6202038
+ 0.9383882 0.4948397 0.9785246 0.6363633
+ 0.9000652 0.5358629 0.9262279 0.6020131
+ 0.9175903 0.4957370 0.8865573 0.6675443
+ 0.9027687 0.5237919 0.8912602 0.6159703
+ 0.8258893 0.4870396 1.0441089 0.5987448
+ 0.9247247 0.4895584 0.9391106 0.5918677
+ 0.8953915 0.4662171 0.8915096 0.6512865
+ 0.8358008 0.5345351 0.9825520 0.6428714
+ 0.8813148 0.4855859 1.0359075 0.6558892
+ 0.9878994 0.4894790 0.8627540 0.5855179
```

```
+ 0.9619740 0.5107840 1.0526355 0.6932790
+ 0.9006527 0.5764256 0.9135464 0.6876119
+ 0.8941355 0.4677227 1.0079439 0.6288488
+ 0.8554127 0.5567003 0.8893319 0.6277068
+ 0.9541563 0.4753967 0.8592374 0.6656648
+ 0.9013626 0.4814269 0.9498793 0.6278216
+ 0.9739542 0.5029272 1.0198768 0.6637317
+ 0.8847032 0.5113723 1.0226162 0.6379343
+ 0.8667802 0.5415473 0.9998916 0.7159989
+ 0.9623856 0.4672669 0.9341589 0.5749412
+ 0.9272194 0.5051652 0.9127402 0.6810382
+ 0.9952835 0.5092058 0.9241529 0.6808844
+ 0.9672866 0.4631770 1.0291962 0.6155057
+ 0.8233839 0.5079111 0.8691708 0.6547891
+ 0.8304894 0.5376769 0.9503627 0.5799626
+ 0.9227659 0.4871963 0.9059131 0.6735011
+ 0.8706225 0.4441442 1.0158954 0.5806382
+ 0.9131319 0.4982515 0.8844850 0.6435803
+ 0.8892915 0.4829204 0.8929878 0.6213515
+ 0.8362308 0.5390086 0.9581169 0.7475521
+ 0.8819314 0.4941663 0.8830290 0.6051975
+ 0.8613954 0.5109117 0.8641273 0.6908952
+ 0.9181492 0.4480691 0.9551777 0.6943885
+ 0.9004550 0.4721167 0.9236793 0.6612836


.ENDDATA

.PARAM


+l=180
+w_poly=0.6
+t_poly=0.2
+tre_m1=0.04636

+tre_poly=1.2
***********Parameters to manipulate:


+k='1/s1'
+j1='1/h1'

+tR_finger='tre_m1*l/6/(w1*t1)'
+tRt='tR_finger/2'
+tR_poly='tre_poly*s1/(t_poly*w_poly)'

+p1='(2.212801-4.135041*j1+1.975136*(j1**2)-2.139694*w1-
6.896431*s1+6.762089*(s1**2)-1.999438*(s1**3)-3.394265*t1+4.471293*j1*w1-
1.943877*w1*(j1**2)+13.9447*j1*s1-13.29183*j1*(s1**2)+3.903754*j1*(s1**3)-
6.473884*(j1**2)*s1)'

+p2='(6.252752*(j1**2)*(s1**2)-1.844696*(j1**2)*(s1**3)+7.151785*w1*s1-
7.102698*w1*(s1**2)+2.11662*w1*(s1**3)+6.58938*j1*t1-
3.157405*t1*(j1**2)+3.330826*w1*t1+11.07973*s1*t1-
10.80463*t1*(s1**2)+3.188786*t1*(s1**3)-13.78758*j1*w1*s1)'

+p3='(13.62611*j1*w1*(s1**2)-4.0375*j1*w1*(s1**3)+6.4706*w1*s1*(j1**2)-
6.349213*w1*(j1**2)*(s1**2)+1.868889*w1*(j1**2)*(s1**3)-
```

```
6.433998*j1*w1*t1+3.038343*(j1**2)*w1*t1-
21.68562*j1*s1*t1+21.14122*j1*(s1**2)*t1-6.221446*j1*(s1**3)*t1)'

+p4='(10.35775*(j1**2)*s1*t1-
10.0577*(j1**2)*(s1**2)*t1+2.953382*(j1**2)*(s1**3)*t1-
11.13048*w1*s1*t1+10.9072*w1*(s1**2)*t1-
3.208904*w1*(s1**3)*t1+21.42735*j1*w1*s1*t1-20.87907*j1*w1*(s1**2)*t1)'
+p5='(6.105417*j1*w1*(s1**3)*t1-
10.03706*(j1**2)*w1*s1*t1+9.69086*(j1**2)*w1*(s1**2)*t1-
2.810206*(j1**2)*w1*(s1**3)*t1)'


+C_finger='((p1+p2+p3+p4+p5)*1e-16*l/6/4)'
+C_finger='((p1+p2+p3+p4+p5)*1e-16*l/6/4)'
+C_finger2='C_finger*2'



+p6='(0.00688471-0.0724764*h1-0.1628133*t1-0.1313869*w1+0.05048325*(w1**2)-
0.1330233*k+0.1945126*(k**2)-0.04740452*(k**3)+0.1757653*h1*t1+0.1821481*h1*w1-
0.06756171*h1*(w1**2)+0.183505*t1*w1-0.05474535*t1*(w1**2))'
+p7='(0.3959186*h1*k-0.2886834*h1*(k**2)+0.06281918*h1*(k**3)+0.7639119*t1*k-
0.2998657*t1*(k**2)+0.06836239*t1*(k**3)+0.4785611*w1*k-
0.3360392*w1*(k**2)+0.07838072*w1*(k**3)-0.147569*(w1**2)*k)'
+p8='(0.11097*(w1**2)*(k**2)-0.02800627*(w1**2)*(k**3)-
0.2410461*h1*t1*w1+0.08525747*h1*t1*(w1**2)-
0.4407959*h1*t1*k+0.3279416*h1*t1*(k**2)-0.07611591*h1*t1*(k**3)-
0.4430397*h1*w1*k)'
+p9='(0.3715836*h1*w1*(k**2)-0.09052401*h1*w1*(k**3)+0.1679877*h1*(w1**2)*k-
0.1423609*h1*(w1**2)*(k**2)+0.0360401*h1*(w1**2)*(k**3)-
0.4905369*t1*w1*k+0.400199*t1*w1*(k**2)-0.1043287*t1*w1*(k**3))'
+p10='(0.1525136*t1*(w1**2)*k-
0.1345484*t1*(w1**2)*(k**2)+0.03861758*t1*(w1**2)*(k**3)+0.653568*h1*t1*w1*k-
0.5309238*h1*t1*w1*(k**2)+0.1355755*h1*t1*w1*(k**3)-0.2368426*h1*t1*(w1**2)*k)'
+p11='(0.2000196*h1*t1*(w1**2)*(k**2)-0.0538886*h1*t1*(w1**2)*(k**3))'


+C_couple='(p6+p7+p8+p9+p10+p11)*1e-16*l/6/4'
+C_couple2='2*C_couple'



.SUBCKT stage in f0 a0 a1 a2 b0 b1 b2 c0 c1 c2 d0 d1 d2 e0 e1 e2 f1 f2  vdd R=tRt
R2=tR_poly C=C_finger C2=C_finger2

Ma a0  in  0   0    tn W=3.05 L=0.40 AD=3.889 AS=1.982  PD=5.6 PS=1.3 NRD=0.175
NRS=0.279
Mb a0  in  vdd vdd  tp W=6.05 L=0.40 AD=7.714 AS=3.932  PD=8.6 PS=1.3 NRD=0.082
NRS=0.260

M1a  end1 in  0   0   tn W=3.05 L=0.40 AD=3.889 AS=3.889  PD=5.6 PS=5.6 NRD=0.178
NRS=0.178
M1b  end1 in  vdd vdd tp W=6.05 L=0.40 AD=7.714 AS=7.714  PD=8.6 PS=8.6 NRD=0.082
NRS=0.082

r11 a0 a1 R
r12 a1 a2 R
c10 a0 0  C
```

```
c11 a1 0  C2
c12 a2 0  C

r21 b0 b1 R
r22 b1 b2 R
c20 b0 0  C
c21 b1 0  C2
c22 b2 0  C

r31 c0 c1 R
r32 c1 c2 R
c30 c0 0  C
c31 c1 0  C2
c32 c2 0  C

r41 d0 d1 R
r42 d1 d2 R
c40 d0 0  C
c41 d1 0  C2
c42 d2 0  C

r51 e0 e1 R
r52 e1 e2 R
c50 e0 0  C
c51 e1 0  C2
c52 e2 0  C

r61 f0 f1 R
r62 f1 f2 R
c60 f0 0  C
c61 f1 0  C2
c62 f2 0  C

ra   a2 b2 R2
rb   b0 c0 R2
rc   c2 d2 R2
rd   d0 e0 R2
re   e2 f2 R2
.ENDS stage

X1 x9f0 x1f0  x1a0 x1a1 x1a2 x1b0 x1b1 x1b2 x1c0 x1c1 x1c2 x1d0 x1d1 x1d2 x1e0
x1e1 x1e2 x1f1 x1f2 vdd stage
c1d4 x1d0 x2a0 C_couple
c1d5 x1d1 x2a1 C_couple2
c1d6 x1d2 x2a2 C_couple

c1e1 x1e0 x2a0 C_couple
c1e2 x1e1 x2a1 C_couple2
c1e3 x1e2 x2a2 C_couple
c1e4 x1e0 x2b0 C_couple
c1e5 x1e1 x2b1 C_couple2
c1e6 x1e2 x2b2 C_couple

c1f1 x1f0 x2b0 C_couple
c1f2 x1f1 x2b1 C_couple2
c1f3 x1f2 x2b2 C_couple
c1f4 x1f0 x2c0 C_couple
c1f5 x1f1 x2c1 C_couple2
c1f6 x1f2 x2c2 C_couple
```

```
X2 x1f0 x2f0  x2a0 x2a1 x2a2 x2b0 x2b1 x2b2 x2c0 x2c1 x2c2 x2d0 x2d1 x2d2 x2e0
x2e1 x2e2  x2f1 x2f2 vdd stage

c2a1 x2a0 x1d0 C_couple
c2a2 x2a1 x1d1 C_couple2
c2a3 x2a2 x1d2 C_couple
c2a4 x2a0 x1e0 C_couple
c2a5 x2a1 x1e1 C_couple2
c2a6 x2a2 x1e2 C_couple

c2b1 x2b0 x1e0 C_couple
c2b2 x2b1 x1e1 C_couple2
c2b3 x2b2 x1e2 C_couple
c2b4 x2b0 x1f0 C_couple
c2b5 x2b1 x1f1 C_couple2
c2b6 x2b2 x1f2 C_couple

c2c1 x2c0 x1f0 C_couple
c2c2 x2c1 x1f1 C_couple2
c2c3 x2c2 x1f2 C_couple
c2c4 x2c0 x3a0 C_couple
c2c5 x2c1 x3a1 C_couple2
c2c6 x2c2 x3a2 C_couple

c2d1 x2d0 x3a0 C_couple
c2d2 x2d1 x3a1 C_couple2
c2d3 x2d2 x3a2 C_couple
c2d4 x2d0 x3b0 C_couple
c2d5 x2d1 x3b1 C_couple2
c2d6 x2d2 x3b2 C_couple

c2e1 x2e0 x3b0 C_couple
c2e2 x2e1 x3b1 C_couple2
c2e3 x2e2 x3b2 C_couple
c2e4 x2e0 x3c0 C_couple
c2e5 x2e1 x3c1 C_couple2
c2e6 x2e2 x3c2 C_couple

c2f1 x2f0 x3c0 C_couple
c2f2 x2f1 x3c1 C_couple2
c2f3 x2f2 x3c2 C_couple
c2f4 x2f0 x3d0 C_couple
c2f5 x2f1 x3d1 C_couple2
c2f6 x2f2 x3d2 C_couple

X3 x2f0 x3f0   x3a0 x3a1 x3a2 x3b0 x3b1 x3b2 x3c0 x3c1 x3c2 x3d0 x3d1 x3d2 x3e0
x3e1 x3e2  x3f1 x3f2 vdd stage

c3a1 x3a0 x2c0 C_couple
c3a2 x3a1 x2c1 C_couple2
c3a3 x3a2 x2c2 C_couple
c3a4 x3a0 x2d0 C_couple
c3a5 x3a1 x2d1 C_couple2
c3a6 x3a2 x2d2 C_couple

c3b1 x3b0 x2d0 C_couple
c3b2 x3b1 x2d1 C_couple2
c3b3 x3b2 x2d2 C_couple
```

```
c3b4 x3b0 x2e0 C_couple
c3b5 x3b1 x2e1 C_couple2
c3b6 x3b2 x2e2 C_couple

c3c1 x3c0 x2e0 C_couple
c3c2 x3c1 x2e1 C_couple2
c3c3 x3c2 x2e2 C_couple
c3c4 x3c0 x2f0 C_couple
c3c5 x3c1 x2f1 C_couple2
c3c6 x3c2 x2f2 C_couple

c3d1 x3d0 x2f0 C_couple
c3d2 x3d1 x2f1 C_couple2
c3d3 x3d2 x2f2 C_couple
c3d4 x3d0 x4a0 C_couple
c3d5 x3d1 x4a1 C_couple2
c3d6 x3d2 x4a2 C_couple

c3e1 x3e0 x4a0 C_couple
c3e2 x3e1 x4a1 C_couple2
c3e3 x3e2 x4a2 C_couple
c3e4 x3e0 x4b0 C_couple
c3e5 x3e1 x4b1 C_couple2
c3e6 x3e2 x4b2 C_couple

c3f1 x3f0 x4b0 C_couple
c3f2 x3f1 x4b1 C_couple2
c3f3 x3f2 x4b2 C_couple
c3f4 x3f0 x4c0 C_couple
c3f5 x3f1 x4c1 C_couple2
c3f6 x3f2 x4c2 C_couple


X4 x3f0  x4f0 x4a0 x4a1 x4a2 x4b0 x4b1 x4b2 x4c0 x4c1 x4c2 x4d0 x4d1 x4d2 x4e0
x4e1 x4e2  x4f1 x4f2 vdd stage

c4a1 x4a0 x3d0 C_couple
c4a2 x4a1 x3d1 C_couple2
c4a3 x4a2 x3d2 C_couple
c4a4 x4a0 x3e0 C_couple
c4a5 x4a1 x3e1 C_couple2
c4a6 x4a2 x3e2 C_couple

c4b1 x4b0 x3e0 C_couple
c4b2 x4b1 x3e1 C_couple2
c4b3 x4b2 x3e2 C_couple
c4b4 x4b0 x3f0 C_couple
c4b5 x4b1 x3f1 C_couple2
c4b6 x4b2 x3f2 C_couple

c4c1 x4c0 x3f0 C_couple
c4c2 x4c1 x3f1 C_couple2
c4c3 x4c2 x3f2 C_couple
c4c4 x4c0 x5a0 C_couple
c4c5 x4c1 x5a1 C_couple2
c4c6 x4c2 x5a2 C_couple

c4d1 x4d0 x5a0 C_couple
c4d2 x4d1 x5a1 C_couple2
```

```
c4d3 x4d2 x5a2 C_couple
c4d4 x4d0 x5b0 C_couple
c4d5 x4d1 x5b1 C_couple2
c4d6 x4d2 x5b2 C_couple

c4e1 x4e0 x5b0 C_couple
c4e2 x4e1 x5b1 C_couple2
c4e3 x4e2 x5b2 C_couple
c4e4 x4e0 x5c0 C_couple
c4e5 x4e1 x5c1 C_couple2
c4e6 x4e2 x5c2 C_couple

c4f1 x4f0 x5c0 C_couple
c4f2 x4f1 x5c1 C_couple2
c4f3 x4f2 x5c2 C_couple
c4f4 x4f0 x5d0 C_couple
c4f5 x4f1 x5d1 C_couple2
c4f6 x4f2 x5d2 C_couple

X5  x4f0 x5f0 x5a0 x5a1 x5a2 x5b0 x5b1 x5b2 x5c0 x5c1 x5c2 x5d0 x5d1 x5d2 x5e0
x5e1 x5e2  x5f1 x5f2 vdd stage

c5a1 x5a0 x4c0 C_couple
c5a2 x5a1 x4c1 C_couple2
c5a3 x5a2 x4c2 C_couple
c5a4 x5a0 x4d0 C_couple
c5a5 x5a1 x4d1 C_couple2
c5a6 x5a2 x4d2 C_couple

c5b1 x5b0 x4d0 C_couple
c5b2 x5b1 x4d1 C_couple2
c5b3 x5b2 x4d2 C_couple
c5b4 x5b0 x4e0 C_couple
c5b5 x5b1 x4e1 C_couple2
c5b6 x5b2 x4e2 C_couple

c5c1 x5c0 x4e0 C_couple
c5c2 x5c1 x4e1 C_couple2
c5c3 x5c2 x4e2 C_couple
c5c4 x5c0 x4f0 C_couple
c5c5 x5c1 x4f1 C_couple2
c5c6 x5c2 x4f2 C_couple

c5d1 x5d0 x4f0 C_couple
c5d2 x5d1 x4f1 C_couple2
c5d3 x5d2 x4f2 C_couple
c5d4 x5d0 x6a0 C_couple
c5d5 x5d1 x6a1 C_couple2
c5d6 x5d2 x6a2 C_couple

c5e1 x5e0 x6a0 C_couple
c5e2 x5e1 x6a1 C_couple2
c5e3 x5e2 x6a2 C_couple
c5e4 x5e0 x6b0 C_couple
c5e5 x5e1 x6b1 C_couple2
c5e6 x5e2 x6b2 C_couple

c5f1 x5f0 x6b0 C_couple
c5f2 x5f1 x6b1 C_couple2
```

```
c5f3 x5f2 x6b2 C_couple
c5f4 x5f0 x6c0 C_couple
c5f5 x5f1 x6c1 C_couple2
c5f6 x5f2 x6c2 C_couple


X6 x5f0  x6f0 x6a0 x6a1 x6a2 x6b0 x6b1 x6b2 x6c0 x6c1 x6c2 x6d0 x6d1 x6d2 x6e0
x6e1 x6e2  x6f1 x6f2 vdd stage

c6a1 x6a0 x5d0 C_couple
c6a2 x6a1 x5d1 C_couple2
c6a3 x6a2 x5d2 C_couple
c6a4 x6a0 x5e0 C_couple
c6a5 x6a1 x5e1 C_couple2
c6a6 x6a2 x5e2 C_couple

c6b1 x6b0 x5e0 C_couple
c6b2 x6b1 x5e1 C_couple2
c6b3 x6b2 x5e2 C_couple
c6b4 x6b0 x5f0 C_couple
c6b5 x6b1 x5f1 C_couple2
c6b6 x6b2 x5f2 C_couple

c6c1 x6c0 x5f0 C_couple
c6c2 x6c1 x5f1 C_couple2
c6c3 x6c2 x5f2 C_couple
c6c4 x6c0 x7a0 C_couple
c6c5 x6c1 x7a1 C_couple2
c6c6 x6c2 x7a2 C_couple

c6d1 x6d0 x7a0 C_couple
c6d2 x6d1 x7a1 C_couple2
c6d3 x6d2 x7a2 C_couple
c6d4 x6d0 x7b0 C_couple
c6d5 x6d1 x7b1 C_couple2
c6d6 x6d2 x7b2 C_couple

c6e1 x6e0 x7b0 C_couple
c6e2 x6e1 x7b1 C_couple2
c6e3 x6e2 x7b2 C_couple
c6e4 x6e0 x7c0 C_couple
c6e5 x6e1 x7c1 C_couple2
c6e6 x6e2 x7c2 C_couple

c6f1 x6f0 x7c0 C_couple
c6f2 x6f1 x7c1 C_couple2
c6f3 x6f2 x7c2 C_couple
c6f4 x6f0 x7d0 C_couple
c6f5 x6f1 x7d1 C_couple2
c6f6 x6f2 x7d2 C_couple

X7  x6f0 x7f0 x7a0 x7a1 x7a2 x7b0 x7b1 x7b2 x7c0 x7c1 x7c2 x7d0 x7d1 x7d2 x7e0
x7e1 x7e2  x7f1 x7f2 vdd stage

c7a1 x7a0 x6c0 C_couple
c7a2 x7a1 x6c1 C_couple2
c7a3 x7a2 x6c2 C_couple
c7a4 x7a0 x6d0 C_couple
c7a5 x7a1 x6d1 C_couple2
```

```
c7a6 x7a2 x6d2 C_couple

c7b1 x7b0 x6d0 C_couple
c7b2 x7b1 x6d1 C_couple2
c7b3 x7b2 x6d2 C_couple
c7b4 x7b0 x6e0 C_couple
c7b5 x7b1 x6e1 C_couple2
c7b6 x7b2 x6e2 C_couple

c7c1 x7c0 x6e0 C_couple
c7c2 x7c1 x6e1 C_couple2
c7c3 x7c2 x6e2 C_couple
c7c4 x7c0 x6f0 C_couple
c7c5 x7c1 x6f1 C_couple2
c7c6 x7c2 x6f2 C_couple

c7d1 x7d0 x6f0 C_couple
c7d2 x7d1 x6f1 C_couple2
c7d3 x7d2 x6f2 C_couple
c7d4 x7d0 x8a0 C_couple
c7d5 x7d1 x8a1 C_couple2
c7d6 x7d2 x8a2 C_couple

c7e1 x7e0 x8a0 C_couple
c7e2 x7e1 x8a1 C_couple2
c7e3 x7e2 x8a2 C_couple
c7e4 x7e0 x8b0 C_couple
c7e5 x7e1 x8b1 C_couple2
c7e6 x7e2 x8b2 C_couple

c7f1 x7f0 x8b0 C_couple
c7f2 x7f1 x8b1 C_couple2
c7f3 x7f2 x8b2 C_couple
c7f4 x7f0 x8c0 C_couple
c7f5 x7f1 x8c1 C_couple2
c7f6 x7f2 x8c2 C_couple

X8 x7f0  x8f0 x8a0 x8a1 x8a2 x8b0 x8b1 x8b2 x8c0 x8c1 x8c2 x8d0 x8d1 x8d2 x8e0
x8e1 x8e2  x8f1 x8f2 vdd stage

c8a1 x8a0 x7d0 C_couple
c8a2 x8a1 x7d1 C_couple2
c8a3 x8a2 x7d2 C_couple
c8a4 x8a0 x7e0 C_couple
c8a5 x8a1 x7e1 C_couple2
c8a6 x8a2 x7e2 C_couple

c8b1 x8b0 x7e0 C_couple
c8b2 x8b1 x7e1 C_couple2
c8b3 x8b2 x7e2 C_couple
c8b4 x8b0 x7f0 C_couple
c8b5 x8b1 x7f1 C_couple2
c8b6 x8b2 x7f2 C_couple

c8c1 x8c0 x7f0 C_couple
c8c2 x8c1 x7f1 C_couple2
c8c3 x8c2 x7f2 C_couple
c8c4 x8c0 x9a0 C_couple
c8c5 x8c1 x9a1 C_couple2
```

```
c8c6 x8c2 x9a2 C_couple

c8d1 x8d0 x9a0 C_couple
c8d2 x8d1 x9a1 C_couple2
c8d3 x8d2 x9a2 C_couple
c8d4 x8d0 x9b0 C_couple
c8d5 x8d1 x9b1 C_couple2
c8d6 x8d2 x9b2 C_couple

c8e1 x8e0 x9b0 C_couple
c8e2 x8e1 x9b1 C_couple2
c8e3 x8e2 x9b2 C_couple
c8e4 x8e0 x9c0 C_couple
c8e5 x8e1 x9c1 C_couple2
c8e6 x8e2 x9c2 C_couple

c8f1 x8f0 x9c0 C_couple
c8f2 x8f1 x9c1 C_couple2
c8f3 x8f2 x9c2 C_couple
c8f4 x8f0 x9d0 C_couple
c8f5 x8f1 x9d1 C_couple2
c8f6 x8f2 x9d2 C_couple

X9  x8f0 x9f0 x9a0 x9a1 x9a2 x9b0 x9b1 x9b2 x9c0 x9c1 x9c2 x9d0 x9d1 x9d2 x9e0
x9e1 x9e2  x9f1 x9f2 vdd stage

c9a1 x9a0 x8c0 C_couple
c9a2 x9a1 x8c1 C_couple2
c9a3 x9a2 x8c2 C_couple
c9a4 x9a0 x8d0 C_couple
c9a5 x9a1 x8d1 C_couple2
c9a6 x9a2 x8d2 C_couple

c9b1 x9b0 x8d0 C_couple
c9b2 x9b1 x8d1 C_couple2
c9b3 x9b2 x8d2 C_couple
c9b4 x9b0 x8e0 C_couple
c9b5 x9b1 x8e1 C_couple2
c9b6 x9b2 x8e2 C_couple

c9c1 x9c0 x8e0 C_couple
c9c2 x9c1 x8e1 C_couple2
c9c3 x9c2 x8e2 C_couple
c9c4 x9c0 x8f0 C_couple
c9c5 x9c1 x8f1 C_couple2
c9c6 x9c2 x8f2 C_couple

c9d1 x9d0 x8f0 C_couple
c9d2 x9d1 x8f1 C_couple2
c9d3 x9d2 x8f2 C_couple

.PARAM SH=1.0
.MODEL TN NMOS
+WMIN='0.399/SH' WMAX='1E+06/SH' LMIN='0.399/SH' LMAX='0.421/SH'
+WMLT=SH LMLT=SH ACM=3HDIF='.625/SH'
+NLEV=2        AF=1          KF=1.0E-24N=1.4
+CJ=0.8491f MJ=0.4560     PB=0.9759JS=3.00E-18
+CJSW=0.1404f MJSW=0.1669PHP=0.6          JSW=5.0E-17
+CTA=0.9008M    CTP=1.0E-15     PTA=1.0E-18      PTP=2.01E-22
```

```
+WDEL=0.08LDEL=-0.06LATD=0.06CAPOP=13
+XPART=0.6       TLEVC=1BULK=0RSH=.5

+LEVEL=47
+Tref=25.0
+Npeak=4.430637E17 Tox=7.0E-9 Xj=1.0E-07
+SatMod=2 SubthMod=2
+BulkMod=1

+Vth0=0.6965789 Phi=0.8936359 k1=.7774307 K2=-.1347269 K3=-4
+Dvt0=0 Dvt1=0 Dvt2=0
+N1x=0 W0=0
+K3b=.22

+Vsat=9438840 Ua=-2.235486e-09 Ub=3.14041e-18 Uc=1.990162e-02
+Rds0=0 Rdsw=570.4586 U0=232.1551
+A0=1.009278
+Keta=6.6505114e-03 Al=3.857901e-02 A2=0.9

+Voff=-.1090096 NFactor=1.023152 Cit=1.687067e-05
+Cdsc=0 Vglow=-.12 Vghigh=.12
+Cdscb=0
+Eta0=2.538968E-02 Etab=-3.37488E-03
+Dsub=0

+Pclm=1.67584 Pdibl1=0 Pdibl2=4.474354E-2
+Drout=0 Pscbe1=5.818927E+08 Pscbe2=3.021162E-5
+Pvag=0
+Eta=0 Litl=4.582576E-08
+Em=0 Ldd=0

+Kt1=-.262 kt2=-0.287
+At=18000
+Ute=-1.09
+Ual=1.39E-09 Ubl=-5.88E-19 Ucl=.0289
+Kt1l=0

.MODEL TP PMOS
+WMIN='0.399/SH' WMAX='1E+06/SH' LMIN='0.399/SH' LMAX='0.421/SH'
+WMLT=SH LMLT=SH ACM=3HDIF='.625/SH'
+NLEV=2        AF=1          KF=2.0E-24N=1.4
+CJ=0.5609f MJ=0.4570    PB=0.8469 JS=3.00E-17
+CJSW=0.08438f MJSW=0.128      PHP=0.9          JSW=3.0E-17
+CTA=0.8201M     CTP=1.0E-15     PTA=1.0E-20      PTP=2.01E-20
+WDEL=-0.02LDEL=-0.06LATD=0.06CAPOP=13
+XPART=0.6       TLEVC=1BULK=0RSH=3.5

+LEVEL=47
+Tref=25.0
+Npeak=3.868E+17 Tox=7.0E-9 Xj=1.0E-07
+SatMod=2 SubthMod=2
+BulkMod=1

+Vth0=-0.6312737 Phi=0.886609 k1=.7263917 K2=-5.926275E-02 K3=0
+Dvt0=20.70769 Dvt1=1.340933 Dvt2=-1.48616E-02
+N1x=0 W0=0
+K3b=0

+Vsat=1.1447902E+07 Ua=-3.241925E-10 Ub=1.739219E-18 Uc=1.029927E-02
```

```
+Rds0=0 Rdsw=1563.3325 U0=83.40553
+A0=0.4018775
+Keta=-0.0234127 Al=0.2235923  A2=0.4

+Voff=-.0772043 NFactor=1.073783 Cit=-1.815597E-04
+Cdsc=5.40629E-03 Vglow=-.12 Vghigh=.12
+Cdscb=0
+Eta0=0 Etab=-.1962199
+Dsub=0.9138862

+Pclm=3.419094 Pdibl1=0.8692119 Pdibl2=1.051596E-2
+Drout=0.9138862 Pscbe1=0 Pscbe2=1E-28
+Pvag=0
+Eta=0 Litl=4.582576E-08
+Em=0 Ldd=0

+Kt1=-.357 kt2=-0.0289
+At=-50000
+ute=-1.28
+Ual=-2.77E-10 Ubl=-6.75E-19 Ucl=.07
+Kt11=0

.tran 0.2ns 40ns start=20ns sweep DATA=no_idea    ----For data-driven analysis....

********.tran 0.02ns 60ns start=20ns sweep Monte=6

.MEAS TRAN TDELAYa TRIG V(x1f0) VAL=1.35 RISE=2
+                  TARG v(x1f0) VAL=1.35 FALL=2

.MEAS TRAN delay  PARAM='abs(TDELAYa/9)'

.MEAS TRAN W  PARAM='w1*1.0'
.MEAS TRAN T  PARAM='t1*1.0'
.MEAS TRAN H  PARAM='h1*1.0'
.MEAS TRAN S  PARAM='s1*1.0'

.MEAS TRAN C_ground   PARAM='(p1+p2+p3+p4+p5)*1e-16'
.MEAS TRAN C_inter PARAM='(p6+p7+p8+p9+p10+p11)*1e-16'
.MEAS TRAN C_total PARAM='(C_ground+2*C_inter)'
.MEAS TRAN Resistance  PARAM='(tre_m1/(w1*t1))'

.end
```