YET ANOTHER SIMULATION BASED SENSITIVITY ANALYSIS TOOL FOR ANALOG LAYOUT GENERATION

by

Taşkın ŞEN

BS, in Electronics and Communication Engineering, Istanbul Technical University

2003

Submitted to the Institute for Graduate Studies in
Science and Engineering in partial fulfillment of
the requirements for the degree of
Master of Science

Graduate Program in Electrical and Electronic Engineering

Boğaziçi University

2007

YET ANOTHER SIMULATION BASED SENSITIVITY ANALYSIS TOOL FOR ANALOG LAYOUT GENERATION

APPROVED BY:

Prof. Günhan Dündar .....
(Thesis Supervisor)

Prof. Avni Morgül .....

Prof. A.C. Cem Say .....

DATE OF APPROVAL: 25.01.2007

ACKNOWLEDGEMENTS

I would like to express my deep gratitude to my thesis supervisor Prof. Günhan Dündar, who initially introduced me the subject of this thesis, and provided invaluable motivation and guidance.

I would like to express my gratitude to Prof. Avni Morgül and Prof. A. C. Cem Say for their corrections and being a member of my thesis jury.

Special thanks to my friend Emre Kaya, for all your help.

I would like to express my special thanks to Özsun Sönmez his valuable inspiration and the contributions.

Finally, I would like to thank to my dear family for their continuous support and encouragement during my education.

ABSTRACT

YET ANOTHER SIMULATION BASED SENSITIVITY ANALYSIS TOOL FOR ANALOG LAYOUT GENERATION

In this thesis, a high performance and flexible analysis tool is presented for computing the sensitivities of performance measures of analog integrated circuits to the parasitic effects that are introduced during layout synthesis and manufacturing. The proposed sensitivity analyzer YASAv2 is based on YASAv1 developed by Mehmet Selçuk Ataç which makes use of its own circuit simulators for computing values of performance measures. YASAv2 is a general purpose tool which can analyze any CMOS circuit provided in SPICE netlist format. YASAv2 supports LEVEL2, LEVEL3, and BSIM3 mosfet parameters. YASAv2 can carry out simplifications according to the computed sensitivity values and the performance specifications provided by the user, to get significant set of parasitic effects which have a critical impact on overall circuit performance. In YASAv2, a new AC analyzer, new parasitic set models, substrate coupling model and inductance as a circuit element are added. Like YASAv1, YASAv2 is coded in C++ using object-oriented programming method. A MFC interface is added for easy usage. It is aimed to integrate YASAv2 into performance driven analog layout synthesis tools (ALG).

ÖZET

ANALOG TÜMDEVRE SERİM KISITLAMALARI ÜRETİMİ İÇİN BENZETİM TABANLI BİR HASSASİYET ANALİZ ARACI

Bu tezde, analog tümleşik devre performans değerlerinin, tümdevre serim, üretim, çalısma aşamaları sırasında oluşan parazitik etkilere karşı hassasiyetlerinin hesaplanmasında kullanılmak üzere tasarlanmış esnek ve yüksek performanslı bir hassasiyet analiz aracı geliştirilip sunulmuştur. Sunulmakta olan ve ilk versiyonu Mehmet Selçuk Ataç tarafından geliştirilen hassasiyet analiz aracı YASAv2 performans değerlerinin hesaplanmasında bünyesinde barındırdığı devre simülatörlerini kullanmaktadır ve diğer herhangi bir sisteme bağımlı değildir. YASAv2 genel amaçlı bir araç olarak tasarlanmış olup, SPICE devre tanım dosya biçimine uygun olarak sağlanan herhangi bir CMOS devre üzerinde hassasiyet analizi yapabilmektedir. Hesaplanan hassasiyet değerleri ve kullanıcı tarafından sağlanan performans kriterleri doğrultusunda önemsiz parazitik etkileri eleyebilmekte ve performans düşüşünde kritik öneme sahip etkileri belirleyebilmektedir. YASAv2 de, AC simülatörü, parazitik kümeler yenilenip taban kuplaj modeli ve devre elemanı olarak endüktans modeli eklenmiştir. YASAv2, YASAv1 gibi C++ da nesne yönelimli programlama yöntemiyle geliştirilmiştir. Kullanıcıya kolaylık sağlaması için MFC arayüzü eklenmiştir. Gelecekte YASAv2'nin performans güdümlü tümleşik devre serim araçları ile entegre edilmesi hedeflenmektedir.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF SYMBOLS/ABBREVIATIONS

$\mathbf{A}$	Augmented nodal matrix
$\mathbf{B}$	Vector of independent sources
$Gain@f_g\_\#o\_\#i$	Voltage gain from the input node i to the output node o at frequency f_g
K_i	i^{th} performance measure
$K_i^{(0)}$	Nominal value of the i^{th} performance measure
ΔK_i	The difference between the nominal and parasitic values of the i^{th} performance measure
$K_i(\mathbf{p})$	i^{th} performance measure as a function of all circuit parameters
$K_i^{(min)}$	The minimum value that the i^{th} performance measure is allowed to take performance measure
$K_i^{(max)}$	The maximum value that the i^{th} performance measure is allowed to take performance measure
$N_{i,j}$	Normalized sensitivity of the i^{th} performance measure to the j^{th} circuit parameter
N_k	The number of all performance measures
N_p	The number of all circuit parameters
$\mathbf{p}$	The vector of all circuit parameters
$\mathbf{p}^{(0)}$	The vector consisting of the nominal values of all circuit parameters
$Phase@f_g\_\#o\_\#i$	Phase margin from the input node i to the output node o at frequency f_g
p_j	j^{th} circuit parameter
$p_j^{(0)}$	The nominal value of the j^{th} circuit parameter
$p_j^{(min)}$	The minimum value that the j^{th} circuit parameter can take
$p_j^{(max)}$	The maximum value that the j^{th} circuit parameter can take
P_i	The set of all non-critical parasitic effects according to K_i
Δp_j	The difference between the ideal and parasitic values of the j^{th} circuit parameter

$S_{i,j}$	Sensitivity of the i^{th} performance measure to the j^{th} circuit parameter
$\mathbf{S}$	The matrix containing the sensitivities of all performance measures to all circuit parameters
$V_{off,i}$	Offset voltage at node i^{th}
V_i	The voltage at node i^{th}
V_{ref}	Reference voltage used for offset calculation
$v_o(f)$	Output voltage at frequency f
$v_i(f)$	Input voltage at frequency f
$\mathbf{x}$	Vector of unknown circuit variables
α	Threshold for parasitic effect simplification
ALG	Analog Layout Generator
BW	Bandwidth
IC	Integrated Circuit
MNA	Modified Nodal Analysis
MR	Magnitude ratio target of frequency search
PCA	Principal Component Analysis
UGBW	Unity-gain bandwidth
YASA	Yet Another Sensitivity Analyzer

1. INTRODUCTION

The main objective of analog CAD engineering is the creation of the methodologies and tools for the design automation of analog circuits, helping designers build functionality while satisfying intended performance specifications. The development of computer aids for design of electronic systems has been one of the fastest growing areas of activity. Physical realization of analog electronic circuits has become one of the largest research areas in this field.

Analog ICs (integrated circuit) have rapidly evolved from the relatively low complexity of the early days to the high sophistication of today [1]. The task of designing an analog circuit requires extremely large amount of computation so that more advanced design software has become a necessity.

In analog circuits, the most important challenge comes from performance degradation due to non-idealities like parasitic effects that are caused by the physical implementation of the circuits or that arise due to the statistical fluctuations of the circuit. These non-idealities are in-circuit parasitics, process variations, mismatches in devices, and substrate effects.

The degradation of circuit performances caused by such parasitics are becoming more critical with decreasing device sizes [2]. To control the non-idealities encountered during the layout phase is important for obtaining the desired performance specifications. Named as constraint driven layout synthesis, this approach translates high-level specifications into a set of physical constraints [3].

1.1. Theoretical Aspects of Constraint - Based Approach

In traditional layout design approach, after the layout is designed, the layout parasitics are extracted, included in the schematic and the circuit is simulated to check if performance specifications are met. However, much more parasitics are seen in the

extracted layout according to circuit netlist. Therefore, the circuit may not meet the performance specification. Besides, it is not obvious who causes which performance degradation in the layout. Therefore, designers go back to layout, make some changes, and extract the layout again. This process requires iteration and too much time. In another way, this technique is not efficient in terms of design time.

A constraint-based approach for analog layout design automation tries to define the bound of parasitics to prove the desired performance parameters for the designed circuit. For meeting the performance specifications of a circuit after physical design, some degradation can always be allowed in the performance functions during layout design as long as they are below certain thresholds. The performance constraints during layout design are the maximum changes allowed in performance functions because of the layout parasitics. A linear performance model on sensitivity analysis, was first proposed and formulated by Choudhury [4]. All performance functions can be represented in a compact and rigorous way as long as they are continuous and sufficiently regular in an interval around their nominal value. This approach involves computing performance sensitivities. Performance sensitivities are calculated using deviation of circuit parameters or parasitic perturbation of circuit. All possible combinations of parasitics which meet performance constraints define a feasible region in the space described by the parasitics. These parasitic constraints can then be used to drive the layout tools. The aim is to give information to layout design automation how circuit performances can be obtained after the extraction of circuit.

1.2. Theoretical Aspects of Substrate Coupling

Substrate coupling in integrated circuits has significant degradation on circuit performance. Continuous technology scaling and high density circuit design cause challenges on circuit performance. The most important one is substrate interference. In IC world, CMOS circuits use same substrate. Because analog circuits are noise sensitive, noise coming from one part of the circuit effects the other part. Therefore, substrate of the circuit must be modeled well to satisfy the performances of the circuits.

In YASAv1, simple resistive substrate coupling model is used. In YASAv2, this model is improved to simple resistive-capacitive substrate coupling model. LEVEL2 and LEVEL3 mosfet model parameters have no substrate parameters. Therefore addition of substrate model parameters is a must for these models. However, BSIM3 mosfet parameters have substrate coupling models which defines mosfet much more realistically compared to LEVEL2 or LEVEL3 parameters. YASAv2 can take circuits having BSIM3 model parameters.

1.3. Organization Of The Thesis

The thesis organized as follows. Chapter 2 provides some background required for a better understanding of this thesis. In this chapter, basic concepts such as performance sensitivities and interconnect parasitics are defined and methods for their numerical computation described. This part of thesis is based on works of Selçuk Ataç [5]. Chapter 3 provides a survey of substrate coupling effects which have important degradation on circuit performances. Some substrate models are given in this chapter. Then, a substrate coupling model to be used in YASAv2 is generated. In chapter 4, generating bounding constraints is presented. Algorithms for generating parasitic constraints from the performance constraints of the circuit are described. In chapter 5, analysis types used for performance computation are presented. Efficient analysis methods are described. Implementation details of YASAv2 are presented in chapter 6. Some example circuits for desired performances are analyzed also in this chapter. Chapter 7 summarizes the contribution of this thesis and provides suggestions for future work.

2. THESIS BACKGROUND

2.1. Constraint Generation

In a constraint-based design or system, performance specifications are enforced by translating them onto a format that can be handled directly by tools responsible for the design. The process of format translation is known as constraint-generation problem. The origin of constraint-based approach dates back to the research of Choudhury about parasitic-aware channel routing [6]. This typical approach consists of two phases. First, according to performance criteria, sensitivities of parasitics or parasitic sets are defined. Then using these sensitivities, parasitic bounds are generated. These bounds guarantee the satisfaction of all performance specifications degraded by parasitics such as symmetry problem, capacitive coupling, etc [7].

Performance values are the parameters of a circuit to meet the desired function. A perturbed circuit parameter in the sensitivity analysis can be any parameter, which affects circuit performance when its value is shifted from nominal. On the other hand, the parasitic effects triggered by the layout have to be modeled by adding new components that are not included in the original design. A capacitor coming from capacitive coupling, which result from finite spacing between two different nets of a circuit, can be given as an example to this parasitic. It has to be modeled and simulated by introducing a capacitor between the circuit nodes representing those nets.

Besides, there are some parasitic effects like substrate coupling which exist in the structure of the wafer but not modeled by design software. Parasitic set represents a set of parasitic effects to be considered and handled together. These parasitic effects are not modeled in circuit by model parameters. But they exist in the real circuit and degrade performance. Therefore, the values of the performance specifications deviate from nominal. All parasitic models have to be added and simulation must be run for the perturbed circuit. The question is in mind "how can sensitivity be calculated for parasitic sets?". All added parasitic effects are deviated from nominal value to calculate

performance sequentially and sensitivity values considering the effects on circuit are calculated. Substrate coupling, for instance, is the one of these parasitic sets. It must be modeled using circuit elements such as resistors-capacitors. Using substrate model, values of the new elements (resistors-capacitors) are determined.

2.2. Performance Functions

Performance functions of a circuit are the functions that describe its behavior. Depending on the defined performance functions, desired performances can be DC or AC performance functions. Example of a performance function of an opamp circuit is gain bandwidth product, which is an AC performance function.

The performance of a circuit depends on several parameters of a circuit. Examples of such parameters are length and width of MOSFETs, process parameters (threshold voltage and oxide thickness of a MOSFET), biasing voltages.

In YASAv1 and YASAv2, the defined performance functions are power consumption, offset voltage, gain bandwidth product, unity-gain bandwidth, gain and phase margin.

2.2.1. Power Consumption

Power consumption of a circuit is an important parameter, especially when low power is desired. It can be measured using DC analyzer. In YASAv1, the overall power consumption of the target circuit is calculated by calculating the individual power consumptions of all the resistors and mosfets that exist by using the voltage and current data that are made available through DC simulation [5].

2.2.2. Offset Voltage

Offset voltage is the second DC performance measure realized in YASAv1. The entity named offset calculates the offset voltage of the i^{th} node in the target circuit

according to:

$$V_{off,i} = V_i - V_{ref} \tag{2.1}$$

The user can define as many offset entities as necessary, by supplying a target node and an associated reference voltage for each one.

2.2.3. Gain

Gain entity calculates the gain of a given output node o with respect to a given input node i at a given frequency f_g as in Equation 2.2:

$$Gain@f_{g\_}\#o\_\#i = \frac{v_o(f)}{v_i(f)}|_{f=f_g} \tag{2.2}$$

In YASAv1, this is achieved by first performing an AC simulation for the given frequency and then dividing the magnitudes of the node voltages. A variety of gain calculations can be configured using this entity by setting the node and frequency parameters accordingly.

2.2.4. Bandwidths and Unity Gain Bandwidths

BW entity searches for the frequency at which gain falls down to 3dB below the DC gain. UGBW entity searches for the frequency where the gain drops to unity.

2.2.5. Phase Margin

Phase margin is calculated using UGBW frequency. This function is calculated erroneously in YASAv1 as in Equation 2.3:

$$Phase@f_{g\_}\#o\_\#i = 180 - [ph(f)]_{f=f_u} \tag{2.3}$$

It is corrected in YASAv2 as in Equation 2.4 if phase margin lower than 0 degree:

$$Phase@f_{g\_}\#o\_\#i = 180 + [ph(f)]_{f=f_u} \tag{2.4}$$

2.3. Parasitics

The parasitics are the effects, which are not desired in the circuit. In YASA, all the parasitic effects are added, simulated, and removed sequentially. Application of parasitics may involve addition of a new device to the circuit, or modifying the value of a component or some geometrical, electrical or physical parameter of a device that already exists in the circuit. The parasitic effect types are explained in this chapter.

2.3.1. Interconnect Parasitics

When placement and routing phases of the circuit are realized in the layout, parasitics are shown in the form of resistance, capacitance, and inductance.

2.3.1.1. Resistances. Any interconnect in the circuit has a series resistance associated with it. Also, interconnect vias and bends cause interconnect parasitics resistances. In YASA, there is no default parasitic resistance calculation. Therefore, the user must give information about parasitic resistances, like value, node1, node2 in the configuration file.

2.3.1.2. Capacitances. Capacitive parasitic is an effect that is caused by the finite spacing of the nets of a circuit. Capacitive coupling that exists between any two nets can be modeled by inserting a capacitor that connects those two nets. Insertion of a capacitor into the circuit is an easier task compared to simulating many of the other parasitic effects because it does not affect the nodal structure of the circuit. Also, the cost of analyzing the effects of capacitive coupling is very low because insertion of a capacitor does not affect the operating point of a circuit, allowing the use of a DC solution over and over. The user has two different options for configuring capacitive coupling effects. The first one is to define all the capacitive coupling instances of interest manually by supplying the all the net pairs of interest individually. The second one is to allow YASA to automatically create the capacitive coupling entities by inspecting the circuit and determining all binary combinations of nets.

2.3.1.3. Inductances. Inductance effect was not included in YASAv1. The inductance model is added in YASAv2.

2.3.2. Parasitic Technology Parameters

In a typical semiconductor process, there are two kinds of parameter variation or deviation, which are special for circuit and general for process. A matching effect in the circuit depends on the geometric placement. Besides, a threshold deviation depend on process condition.

2.3.2.1. Model Parameters. In YASA, mosfet model parameter deviations can be analyzed using configuration file. For example, oxide thickness differences causes performance degradation on circuit. Using YASA how much degradation occurs from the oxide thickness deviation can be calculated.

2.3.2.2. Geometric Parameters. Geometric parasitic are caused by mismatches in the geometry of devices. Groups of mosfets having two or more transistors, such as in the case of differential pairs and current mirrors, may be designed identically or meant to

have properties linked to each other. Non-uniformities of the fabrication process can cause variations in the properties of these transistors, resulting in mismatch. YASA provides the parasitic entity, namely mosfet geometry parasitic for flexible analysis of the effect caused by transistor mismatch.

2.3.3. Automatic Parasitic Generation

YASA supports automatic parasitic generation. Automatic parasitic generation is realized for capacitive coupling and substrate coupling. A capacitance for each node pair is added and performance analysis realized. Then, the parasitic is removed from the circuit. The process is repeated until all of node pairs are analyzed according to performance criteria. The process for capacitive coupling is realized also for substrate coupling.

2.4. Performance Sensitivities

Performance specification is defined as the maximum tolerated performance degradation from nominal caused by process variance and the parasitics coming from layout. Using sensitivities performance measures of concern can be expressed as functions of parasitic effects, therefore the bound on parasitics can be calculated using these high-level performance specifications.

At this point, it is necessary that the defined performance measure as in Equation 2.7 must be continuously differentiable functions of all circuit parameters that can have an effect of the performance. Using sensitivities is an efficient way of expressing the dependence of performance measures on circuit parameters, provided that performance functions are sufficiently regular in an interval around their nominal values[7].

The array of all circuit parameters affecting performance can be denoted as

$$\mathbf{p} = \begin{pmatrix} p_1 & \dots & p_{N_p} \end{pmatrix}^T \tag{2.5}$$

and the array of their nominal values as

$$\mathbf{p}^{(0)} = \begin{pmatrix} p_1^{(0)} & ... & p_{N_p}^{(0)} \end{pmatrix}^T \tag{2.6}$$

where N_p represents the number of circuit parameters. The sensitivity of the i^{th} performance function to the j^{th} circuit parameter p_j can be defined as

$$S_{i,j} = \frac{\partial K_i(\mathbf{p})}{\partial p_j}|_{\mathbf{p}^{(0)}} \tag{2.7}$$

The array of all sensitivities is

$$\mathbf{S} = \begin{pmatrix} S_{1,1} & ... & S_{1,N_p} \\ ... & ... & ... \\ S_{N_k,1} & ... & S_{N_k,N_p} \end{pmatrix} \tag{2.8}$$

2.4.1. Sensitivity Analysis

YASA is a simulation-based sensitivity analysis tool that computes sensitivities using perturbation method by performing circuit simulation, and determines the physical constraints.

The sensitivity of the i^{th} performance function K_i to the j^{th} circuit parameter p_j defined in Equation 2.7 can be computed by performing circuit simulations according to

$$S_{i,j} = \frac{\Delta K_i}{\Delta p_j} = \frac{K_i - K_i^{(0)}}{p_j - p_j^{(0)}} \tag{2.9}$$

where $p_j^{(0)}$ and $K_i^{(0)}$ denote the nominal values of the j^{th} layout parameter and the i^{th} performance measure respectively. To calculate the sensitivities of concern, it is required to iterate once for each circuit parameter, and calculate performance values of the each nominal and perturbed circuit for each nominal and introduced parameter,

respectively.

In the sensitivity calculation process, the parasitic effect and its value are defined. To calculate sensitivity, a performance value and definition of performance function is necessary. In YASA, different performance functions and definitions exist. Using configuration file sensitivities of all parasitic effects for all performance criteria can be obtained.

Computation of sensitivities does not give enough information about circuit parasitic effect importance. Using allowed amount of deviation of performance and parasitic effects contribution based normalized sensitivity can be defined as follows:

$$N_{i,j}^{C} = |\frac{p_j^{(max)} - p_j^{(min)}}{K_i^{(max)} - K_i^{(min)}} \cdot S_{i,j}| \tag{2.10}$$

where $p_j^{(max)} - p_j^{(min)}$ and $K_i^{(max)} - K_i^{(min)}$ denote the maximum possible amount of deviation of the j^{th} layout parameter and the i^{th} performance measure respectively. This form of normalized sensitivity is an index of the relative criticality of a parasitic effect in meeting performance specifications [5].

The pseudo-code of sensitivity analysis [8] given in Figure 2.1 is improved in Figure 2.2 in YASAv1.

In the proposed pseudo-code Figure 2.1 first, nominal performances are calculated, then a parasitic is added and performance values are calculated. After that, the parasitic is removed. For each parasitic effect this procedure must be repeated again and again. This algorithm is first proposed by Charbon [9].

Given in Figure 2.2, the pseudo code written by Selçuk Ataç [5] is the base of YASAv1, and YASAv2. First, the circuit is perturbed using the defined parasitics. Then, parasitic and nominal performances of the circuit are calculated. Finally, the parasitic is removed and the circuit is taken back to normal. Using performance and

```
for EACH Performance do
  Performance.ComputeNominalValue();
  for EACH Parasitic do
    Parasitic.InfectCircuit();
    Performance.ComputeParasiticValue();
    Sensitivity(Performance,Parasitic)=Performance.Compute-
    Sensitivity(Parasitic);
    Parasitic.RestoreCircuit();
  end for
end for
```

Figure 2.1. The pseudo-code of sensitivity analysis

parasitic parameters, sensitivity and normalized sensitivity are calculated. All of the outputs are preserved in matrix sets.

2.4.2. Determining Significant Sets of Sensitivities

Parasitics added manually or automatically to the analysis environment generally do not cause significant performance degradation. In YASA, there is a mechanism which eliminates insignificant parasitic effects [5]. All parasitic effects are sorted incrementally. Defined cumulative contribution threshold $\alpha < 1$ and eliminate first n_i parasitics in each sorted list such that.

$$\sum_{j=1}^{n_i} N_{i,j}^{C} \leq \alpha \tag{2.11}$$

2.5. Circuit Analysis Types

In YASA, there are two types of simulation, DC and AC. All performance measurements are made according to these simulations.

```
for EACH ParasiticSetINAnalysisEnvironment do
  ParasiticSet.InitializeCircuit;
  if Circuit.HasBeenModified = True then
    for EACH PerformanceINAnalysisEnvironment do
      Performance.NominalValue = Performance.CalculateValue;
    end for
  end if
  for EACH ParasiticINParasiticSet do
    Parasitic.Apply();
    for EACH PerformanceINAnalysisEnvironment do
      Performance.ParasiticValue = Performance.CalculateValue;
      SensitivityValue = Performance.DeltaPerformance / Parasitic.Deltaparasitic;
      ParasiticSet.SensitivityData.Sensitivity (Performance,Parasitic) = Sensitivi-
      tyValue;
      ParasiticSet.SensitivityData.NormalizedSensitivity(Performance,Parasitic)=
      Parasitic.MaxDeviation / Performance.MaxAllowedDeviation * Sensitivity-
      Value;
    end forParasiticRemove
  end for
end for
```

Figure 2.2. The pseudo code of YASA

2.5.1. Nonlinear DC Analysis

Nonlinear DC analysis module of YASA was realized by İsmet Bayraktaroğlu [10]. In 2003, the module was inserted by Mehmet Selçuk Ataç into YASA [5]. The simulator uses Newton Raphson method to find the DC solution of the given circuit.

2.5.2. AC Analysis

In YASA, AC performance functions are calculated using AC simulator module. The realized AC simulator in YASAv1, modified nodal analyzer, does not converge

and give wrong information about circuit performance. In YASAv2, AC simulator is renewed.

2.6. Summary

General terms and information about YASA are given in this chapter. Performance functions, parasitic perturbed sensitivity calculation method are explained. Other modules of YASA such as automatic parasitic generator and performance analyzers, nonlinear DC and AC are studied.

3. SUBSTRATE COUPLING

This chapter presents an overview and comparison of several prevalent techniques for substrate noise coupling in integrated circuits. A brief review of the literature on substrate noise and its effects is provided to show the importance of the techniques for substrate noise coupling. The earliest technique, finite mesh method is presented. Other current techniques are also analyzed. Eventually, the simple resistive and capacitive technique is presented.

As the complexity of analog designs increases, and the area of the current technologies decreases, substrate noise coupling in integrated circuits becomes a significant consideration in the analog design [11]. Since the substrate coupling between on-chip analog circuits can corrupt low-level signals, it can impair the performance of signals in the integrated circuit. In order to determine the amount of coupling between the sensitive nodes and noisy nodes, modeling techniques for substrate noise coupling should be generated. During the last decade, several substrate noise-coupling techniques have been developed. There is no perfect modeling technique existing for this problem. Therefore, it is in order to analyze the models in the literature, and present the model used in YASAv2.

3.1. Substrate Noise Fundamentals

3.1.1. Parasitic Effect of Substrate

The inner structure of a substrate influences the behavior of a circuit. For example, bulk structure of the substrate causes late signal transmissions. The current flowing to the ground through the substrate causes a voltage drop, which affects the device operation. In addition, the substrate is not a perfect isolator between devices, leading to unwanted "cross-talk" in integrated circuits. As illustrated in Figure 3.1, a parasitic RLC circuit on a capacitor is introduced when the substrate is connected with bonding wires. This affects integrated devices in the circuit. For instance, a typical

bonding inductance of 4nH together with 10pF capacitance has a resonant frequency of 800MHz, which is a source of instabilities and oscillations [12].

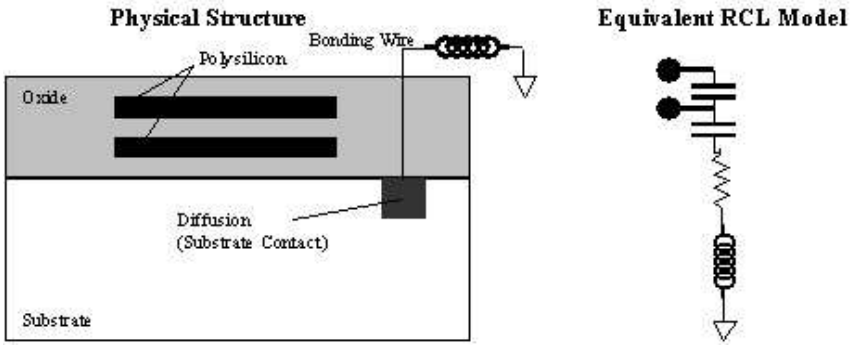

Figure 3.1. Parasitic effect on a capacitor made of two polysilicon

In the common silicon substrate, the phenomenon of cross talk occurs if a sensitive circuit portion of the integrated circuits is presented along this parasitic path. It is perturbed by the noisy signal, so that the substrate is used as a parasitic return path for signals carrying relevant information shown in Figure 3.2. Also, the cross-talk problem arises in any substrate coupled regions such as the drain of an n-channel mosfet or a bonding pad changes.

Furthermore, the substrate can also give AC noise to the ground when the least resistive path is followed and noise flow is determined by the distribution of substrate contacts to ground. In Figure 3.3, the presence of a backside contact produces a vertical current. Besides this, there are numerous parasitic capacitances exist at every node in a circuit. C_{ws} is the capacitance between the substrate and a well, which can provide

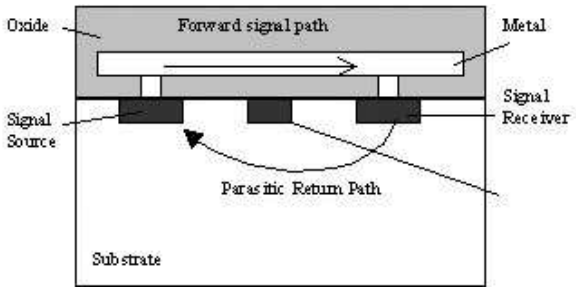

Figure 3.2. Substrate as a parasitic return path

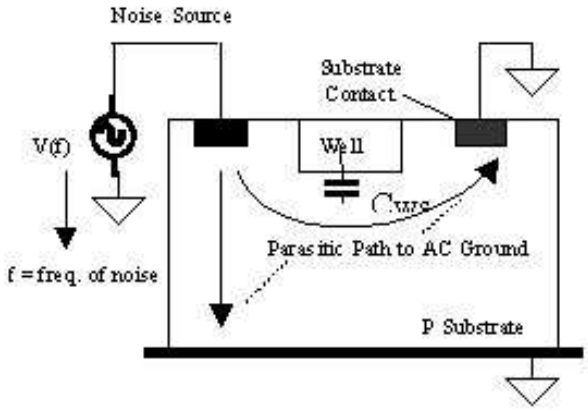

Figure 3.3. Substrate as a parasitic path to AC ground

a channel for noise to go into the substrate. Consequently, the substrate behaves as a noise vehicle and channel.

3.2. Modeling Techniques for Substrate Noise Coupling

Considerable research efforts have been invested for modeling the substrate. There are four main methods for substrate coupling, mesh-based extraction, boundary-based extraction, preprocessing, and macromodeling [12].

3.2.1. Finite Difference Mesh Method

The finite difference mesh method is the earliest technique developed for substrate noise coupling. It employs the discretization technique to model the substrate as layers of uniformly doped semiconductor of varying doping density outside the diffusion regions. As illustrated in Figure 3.4, using a finite difference operator, nodes are defined across the entire substrate volume [13].

The electric field vector between adjacent nodes is also approximated. Discretizing on the substrate volume results in a mesh circuit consisting of nodes interconnected by branches of capacitors and resistors in parallel, shown in Figure 3.5, the values of which are determined from process parameters - dielectric constant, sheet resistivity or doping density. Ignoring magnetic fields and using the identity $\nabla \cdot (\nabla \times a) = 0$,

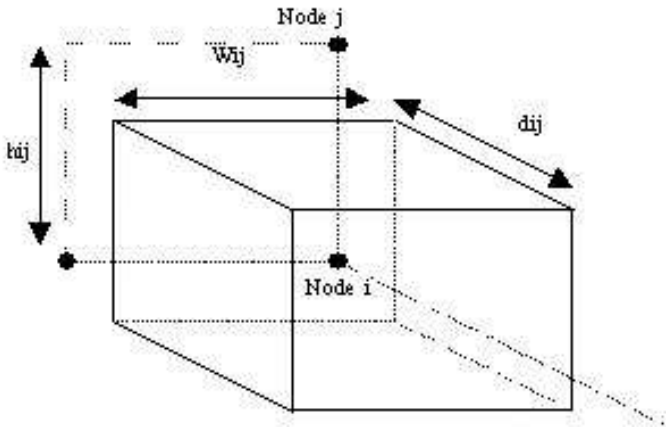

Figure 3.4. A control volume in the box

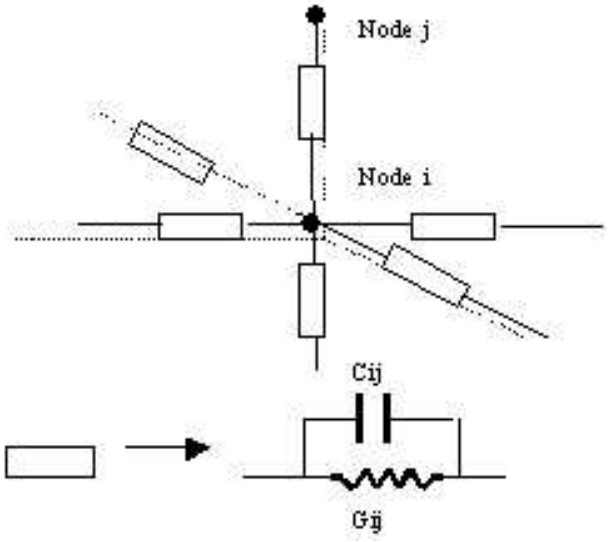

Figure 3.5. Capacitances and Resistances

Maxwell's equations can be written as:

$$\nabla \cdot J + \nabla \cdot \frac{\partial D}{\partial t} = 0 \tag{3.1}$$

where $D = \varepsilon E$; and $J = \frac{1}{\rho} E$; and it gives,

$$\frac{1}{\rho}(\nabla \cdot E) + \varepsilon \frac{\partial}{\partial t}(\nabla \cdot E) = 0 \tag{3.2}$$

$$E_{ij} = \frac{V_i - V_j}{h_{ij}} \tag{3.3}$$

where ρ is the sheet resistivity, ε is the dielectric constant, and E is the electric field intensity vector. In this stage, a simple box integration technique should be utilized to solve the above equation, since the substrate is spatially discretized. From Gauss' law, it gives

$$\nabla \cdot E = k \tag{3.4}$$

and $k = \frac{p'}{\varepsilon}$ where p' is the charge density of the material. From the divergence theorem,

$$\int_{Si} E dS = \int_{\Lambda i} k d\Lambda \tag{3.5}$$

where Si is the surface area of the cube and Λ_i is the volume of the cube shown in Figure 3.4. The left hand side of the Equation 3.5 can be approximated as

$$\int_{Si} E dS \approx \sum_j E_{ij} \cdot S_{ij} = \sum_j E_{ij} \cdot w_{ij} d_{ij} = k \cdot \Lambda_i \tag{3.6}$$

Modifying Equations 3.4 and 3.5,

$$\nabla \cdot E = k = \frac{1}{\Lambda i} \sum_j E_{ij} \cdot w_{ij} d_{ij} \tag{3.7}$$

Substituting Equation 3.7 and 3.3 into 3.2, it results

$$\sum_j [G_{ij}(V_i - V_j) + C_{ij}(\frac{\partial}{\partial t}V_i - \frac{\partial}{\partial t}V_j)] = 0 \tag{3.8}$$

where $G_{ij} = \frac{(w_{ij} \times d_{ij})}{\rho h_{ij}}$ and $C_{ij} = \frac{\varepsilon(w_{ij} \times d_{ij})}{h_{ij}}$ as modeled with RC circuit elements shown in Figure 3.5. The above result shows that the areas of contact and diffusion are represented as equipotential regions in the resulting three-dimensional RC mesh and treated as ports in the multiport network [14].

The accuracy of the Finite Difference Mesh Method's solution depends highly on the resolution of discretization. In addition, it is necessary to use fine grids to accurately approximate the non-linearity of the electric field intensity. Consequently, the size of the resulting finite difference mesh matrix, with an increasing number of ports and fine grids, becomes too large to solve. To reduce the size of RC model network there are two types of methods:

1. use coarse grids to reduce the overall number of grids,
2. ignore substrate capacitances, and consider the substrate as a purely resistive mesh.

Even though the above methods may reduce RC matrix, the finite difference method generally has a huge sparse matrix because it consists of discretizing the entire substrate and applying different equations at each node, due to the usage of a purely numerical calculation technique. Consequently, the Finite Difference Mesh method can be utilized to determine reduced order substrate models.

3.2.2. Boundary Element Methods

Gharpurey and Meyer have developed Boundary Element Methods using Green's function [15], [16] for efficient calculation of substrate macromodels in the last decade. The macromodels can be included in circuit simulators such as SPICE, in order to

predict the effects of substrate noise coupling and to allow optimization of the layout to minimize these effects [16].

In the substrate medium, Poisson's equation holds under the quasi-static assumption

$$\nabla^2\phi = -\frac{\rho}{\varepsilon} \tag{3.9}$$

A Green's function, $G(x_0, y_0, z_0, x, y, z)$, is defined as the potential induced at position (x, y, z) in the medium due to a unit charge source located at position (x_0, y_0, z_0). Green's function can be expressed analytically for stratified, layered substrates [16]. With the introduction of Green's function, the solution to Equation 3.9 can be determined by

$$\phi = \int_V \rho(x^{'}, y^{'}, z^{'})G(x^{'}, y^{'}, z^{'}, x, y, z)dx^{'}dy^{'}dz^{'} \tag{3.10}$$

The volume integral above can be discretized assuming a uniform potential and uniform charge density on the substrate contact:

$$[\Phi] = [P][Q] \tag{3.11}$$

$$[Q] = [c][\Phi] \tag{3.12}$$

where $[\Phi]$ and $[Q]$ are the contact potential and charge vectors and $[P]$ is the potential coefficient matrix with its entries determined by

$$p_{ij} = \frac{1}{V_i}V_j \int_{V_i}[\int_{V_j} G(x^{'}, y^{'}, z^{'}, x, y, z)dx^{'}dy^{'}dz^{'}]dxdydz \tag{3.13}$$

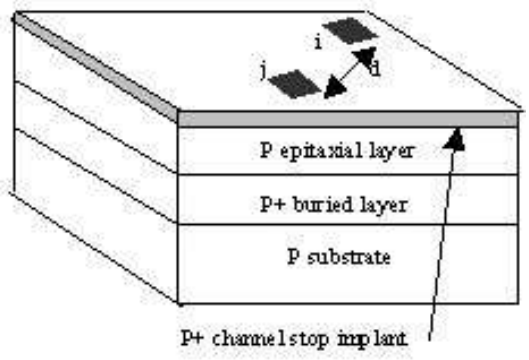

Figure 3.6. Two points substrate impedance ports separated by distance,d

Finally, the capacitance matrix is determined by

$$[C] = [P]^{-1} \tag{3.14}$$

The method using Green's function discussed above can be found in [16], [17], [18]. This method is enhanced in [19], [20], [21], and [22], resulting in reduced matrix size and improved computation efficiency. A mixed Finite Element Method (FEM) and Boundary Element Method (BEM) approach was presented in [23], allowing a combination between a 2-D FEM method for the layout-dependent doping patterns and a 3-D BEM method for the bulk substrate. To overcome the large matrix size problem in the conventional Green's function-based methods due to the uniform current distribution requirement, an analytical formulation of the nonuniform current distribution, mainly as a function of contact size and shape, was proposed in [23], where a speed up of orders of magnitude was reported.

3.2.3. Preprocessing Analytical Method

The previous two substrate coupling techniques, the Finite Difference Mesh Method and Boundary Element Method can only be utilized after layout extraction and do not provide a priori insight to the designer. The following three methods, the Preprocessing Analytical Method, Simple Resistive Method, and Simple Resistive-Capacitive Method can provide some insights to circuit designers in the early stage of design [14]. In a pre-processing stage, precomputed z parameters are used to develop a preprocessing

analytical method for substrate noise coupling. This approach is then used in an extraction stage to find out point-to-point impedance. First of all, three assumptions are made. Current density across ports is uniform, ports are equipotential, and the effects of chip edge are ignored. As derived in Figure 3.6, two square ports are on top of the substrate separated by a distance, d. Characterizing the electrical interaction, the matrix equation between two ports is given as follows:

$$\begin{pmatrix} z_{ii} & z_{ij} \\ z_{ji} & z_{jj} \end{pmatrix} \begin{pmatrix} i_i \\ i_j \end{pmatrix} = \begin{pmatrix} v_i \\ v_j \end{pmatrix} \tag{3.15}$$

where z_{ii} is the potential observed at point i when a unit current is injected into point i, while point j is floating due to zero current. z_{jj} is the potential at point j due to a unit current injected at point j. $z_{ij} = z_{ji}$ is the potential at one point when a unit current is injected at the other. z_{ii} and z_{jj} are constant because of ignoring the effects of the edges in the lateral plane. z_{ij} is a function of only the distance d between the two points. As z_{ij} is inversely proportional to d, it gives:

$$z_{ij}(d) = k_0 + \frac{k_1}{d} + \frac{k_2}{d^2} + ... + \frac{k_m}{d^m} \tag{3.16}$$

$$z_{ii} = K_1 \tag{3.17}$$

$$z_{jj} = K_2 \tag{3.18}$$

$$z_{ij} = z_{ji} \tag{3.19}$$

where the constants, K_1, K_2, k_i, and the polynomial order, m can be determined by first precomputing the actual parameters using curve fitting techniques on data points obtained by a 3-D numerical simulator. For multiple ports on the surface, large ports should be discretized into smaller ports, illustrated in Figure 3.6, due to the assumption

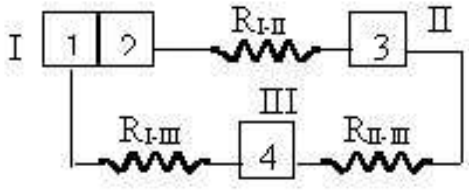

Figure 3.7. Determining resistive coupling between ports using point-to-point impedance

of uniform current density across the port. Using the above preprocessing analytical model, an admittance and impedance matrix can be formed. The impedances for the ports I, II and III shown in Figure 3.7 can be calculated as follows:

$$\begin{pmatrix} y_{11} & y_{12} & y_{13} & y_{14} \\ y_{21} & y_{22} & y_{23} & y_{24} \\ y_{31} & y_{32} & y_{33} & y_{34} \\ y_{41} & y_{42} & y_{43} & y_{44} \end{pmatrix} = \begin{pmatrix} z_{11} & z_{12} & z_{13} & z_{14} \\ z_{21} & z_{22} & z_{23} & z_{24} \\ z_{31} & z_{32} & z_{33} & z_{34} \\ z_{41} & z_{42} & z_{43} & z_{44} \end{pmatrix}^{-1} \tag{3.20}$$

$$R_{I-II} = (y_{13} + y_{23})^{-1}; R_{II-III} = (y_{34})^{-1}; R_{I-III} = (y_{14} + y_{24})^{-1} \tag{3.21}$$

The preprocessing analytical method is the simple substrate noise coupling technique that can be developed in preprocessing stage. It is used in the extraction process to evaluate point-to-point impedances rapidly. The coefficients need to be computed only once for a given process because the resulting data can be stored in libraries and then used in the real-extraction of different integrated circuit again. On the other hand, only the ports that connect the substrate to the wells, contacts, or devices have to be discretized so that the resulting matrix in the network is much smaller. The computation of this approach is faster than the Finite Difference Mesh Method and Boundary Element Method. The trade-off of this approach is that the accuracy is relatively low.

3.2.4. Simple Resistive Macromodel Method

Recently, a similar idea of a scalable macromodel for substrate noise coupling in heavily doped substrates is explored [24]. It is called the Simple Resistive Macromodel method. Based on a physical understanding of the current flow paths, this approach requires only three parameters which can be extracted from simulations or measurements. The Simple Resistive Macromodel method is a simple but not accurate method, and can provide a clear picture about the substrate noise coupling to IC designers in the early stages of the design. This method is also used by Selçuk Ataç in YASAv1. The drawback in YASAv1 is that the component values are taken as constants. Therefore, the results are independent of the characteristics of the substrate and highly inaccurate.

According to current flow lines between a source point and a sensor point from a device simulator, a typical heavily doped substrate can be modeled as a resistive network as depicted in Figure 3.8 [25]. In this circuit, $G_2(= 1/R_2)$ is the cross coupling conductance, while $G_{1A}(= 1/R_{1A})$ and $G_{1B}(= 1/R_{1B})$ are conductances from the source and the sensor, respectively. From the experimental results, R_2 decreases as the separations between the source and the sensor decreases; otherwise, R_2 increases rapidly for larger separations. R_{1A} and R_{1B} are independent of the separation and remain constant. Note that the back plane is either grounded, connected to ground through an impedance, or left floating.

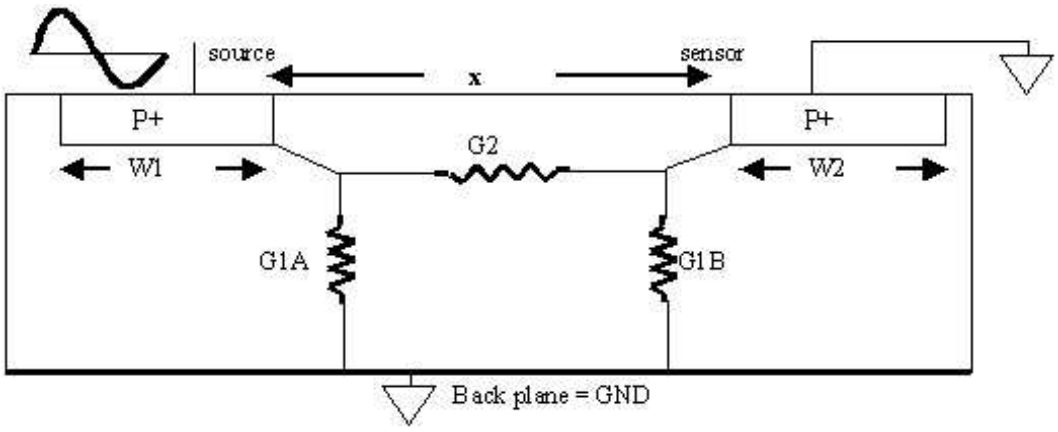

Figure 3.8. Macromodel for the substrate when the backplane is ground

In order to obtain a Y-parameter matrix for the equivalent circuit of the heavily doped substrate, an AC voltage is applied at one port and the currents are measured with the other port connected to ground. Assume that sizes and shapes of the contacts are the same, then it gives: $G_2 = G_{1A} = G_{1B}$. The following two-port Y-parameters for the substrate macromodel is shown:

$$Y = \begin{pmatrix} y_{11} & y_{12} \\ y_{21} & y_{22} \end{pmatrix} = \begin{pmatrix} G_1 + G_2 & -G_2 \\ -G_2 & G_1 + G_2 \end{pmatrix} \tag{3.22}$$

In order to determine G_1 and G_2, a Z-parameter matrix is used, and it gives:

$$Z = \begin{pmatrix} z_{11} & z_{12} \\ z_{21} & z_{22} \end{pmatrix} = Y^{-1} = \frac{1}{G_1^2 + 2G_1G_2} \begin{pmatrix} G_1 + G_2 & G_2 \\ G_2 & G_1 + G_2 \end{pmatrix} \tag{3.23}$$

Let $\Delta = (G_1^2 + 2G_1G_2)$ be the determinant of the Y-parameter matrix. z_{11} is a constant ξ because it is the impedance of contact one to the substrate, with all other contacts flowing. Thus, the constant ξ can be extracted from simulations. This gives

$$z_{11} = \frac{G_1 + G_2}{G_1^2 + 2G_1G_2} = \xi \Rightarrow G_1^2 + 2G_1G_2 - \frac{1}{\xi}(G_1 + G_2) = 0 \tag{3.24}$$

Rearranging the above equation,

$$G_1(x) = \frac{1}{2\xi} - G_2(x) + \frac{1}{2\xi}\sqrt{1 + 4\xi^2 G_2^2(x)} \tag{3.25}$$

where G_1 and G_2 can be determined from simulators and measurements, and they are dependent of the separations between the source and sensors. Based on the linear dependence on the semi-plot of G_2, it provides a relationship between G_2 and the separation x [14].

$$G_2(x) = \alpha e^{-\beta x} \tag{3.26}$$

where α and β are constants determined from simulations and measured data.

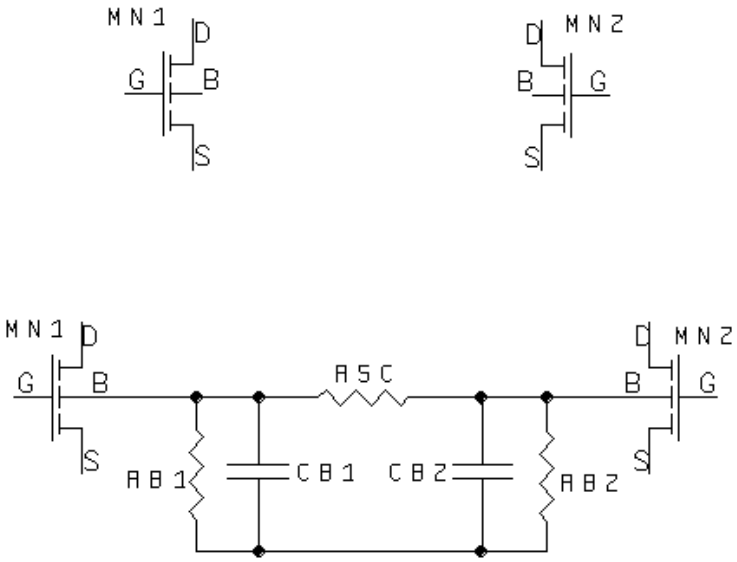

Figure 3.9. Simple Parasitic and Capacitive Substrate Coupling

Consequently, there are only two contact points required to obtain relatively accurate results, so the accuracy of α and β can be improved with more data and a nonlinear least-square fit. The main shortcoming of this method is that it can only be used on the heavily doped substrate [11]. Moreover, the operating frequency must be below 2-3 GHz because the substrate can only be modeled as the equivalent circuit as illustrated in Figure 3.8. In general, the Simple Resistive Macromodels Method is a simple technique and provide fairly accurate results. Significantly, it can provide good insights regarding the substrate noise coupling to IC designers in the early stages of the design.

3.2.5. Simple Resistive and Capacitive Macromodel Method

This model is similar to the simple resistive model, and is novel to the best of our knowledge. The substrate is modeled using resistive and capacitive parasitics. It is a simple but approximate model. In YASA, the aim is to calculate the parasitic effects, and not to extract the exact model for substrate modeling. The model given in Figure 3.9 is used for all substrate types. R_{s1} and R_{s2} models are given below [24]:

$$R_{si} = \frac{K_{n,p}}{K_1 \cdot Area_i + K_2 \cdot Perimeter_i + K_3} \tag{3.27}$$

K_1, K_2, K_3 are empirical parameters. K_n and K_p are also empirical parameters which is defined for nwell or pwell.

R_{sij} model [12] is

$$R_{sij} = \Omega_{/\mu m} \cdot \varphi \cdot [\ln(d_{ij} + 1)^{\alpha_1} \cdot (s_i + s_j)^{\alpha_2} \cdot (p_i + p_j)^{\alpha_3}] \tag{3.28}$$

where $\Omega_{/\mu m}$ is a resistance having unit [ohms/micrometer], φ, a_1, a_2,a_3, are empirical fitting parameters, d_{ij} is the minimum technology space between mosfets, s_i is the area of $mosfet_i$, p_i is the perimeter of the $mosfet_i$.

The capacitance model of substrate is given below:

$$C_{si} = (AD + AS + W \cdot L) \cdot C_{silicon} \cdot \beta \tag{3.29}$$

$$C_{silicon} = \frac{\varepsilon_{Si}}{t_{Si}} \tag{3.30}$$

where AD is the area of drain, AS is the area of source, W is the width of mosfet, L is the length of mosfet, β is a empirical fitting parameter.

The most important stage after deciding of the model is the calculation of model parameters. There are not many programs or simulators calculating substrate coupling. These tools extract substrate parasitics after the layout phase. The most known is Cadence SubstrateStorm which is a commercial product. Another known tool is SPACE which is an accurate and efficient layout-to-circuit extractor for deep submicron technologies being actively developed at Delft University of Technology [26].

For the generation of TSMC0.18 process parameters, inverter gate layouts are

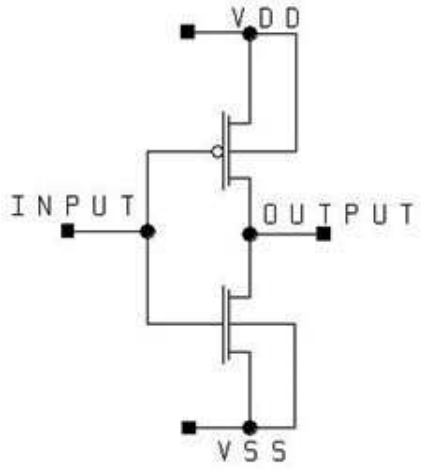

Figure 3.10. Inverter Circuit

used. The circuits given their schematic in Figure 3.10 are drawn in MAGIC layout tool and exported to .gds files. Then, using Space substrate extractor tool which takes .gds layout file, substrate coupling elements are generated. These generated values and geometric information of the circuit are given in Table 3.1.

Table 3.1. Space Extracted and Calculated Resistor Values

	Wn	Wp	R_1^s	R_2^s	R_3^s	R_1^y	R_2^y	R_3^y
inv2a	0.72u	1.68u	451k	341k	1.5meg	452k	452k	1.535meg
inv4a	1.44u	3.36u	394k	262k	1meg	421k	421k	1.116meg
inv6a	2.16u	5.04u	326k	243k	823k	394k	394k	900k
inv8a	2.88u	6.72u	286k	215k	695k	370k	370k	765k
inv10a	3.6u	8.4u	260k	193k	613k	349k	349k	671k

Using Space the capacitor model of substrate is tried to be extracted. But the tool extracts only, drain-source, drain-gate, gate-source capacitors of a mosfet. Therefore, the parameters for the calculation of the capacitor parasitic sets are defined based on our knowledge. Besides, in YASA user can define his/her special process parameter in the configuration file and take the desired outputs.

Table 3.2. Parameter Values For TSMC 0.18 in YASA

Parameter Name	Value	Parameter Name	Value
K_n	25	d_{ij}	$0.1{\cdot}10^{-6}$
K_p	37.5	α_1	$10{\cdot}10^{6}$
K_1	$5.1{\cdot}10^{6}$	α_2	-1/12
K_2	1	α_3	-0.6
K_3	$50{\cdot}10^{-6}$	t_{Si}	$10{\cdot}10^{-6}$
$\Omega_{/\mu m} \cdot \varphi$	105	β	1

3.3. Summary

In this chapter, substrate coupling effect and models are explained. Because Finite Difference Mesh Method, Boundary Element Method, Preprocessing Analytical Method are used after the layout, they are not appropriate for pre-layout simulation. The simple resistive macromodel only includes resistive parasitics and is used in YASAv1. The simple resistive-capacitive macromodel is derived in this thesis. The model parameter is process dependent and extracted for TSMC 0.18um in SPACE extractor. Using these parameters, substrate analysis can be made. Also, users can define specific process parameters in a configuration file.

4. BOUNDING CONSTRAINTS ON PARASITICS

In general, the parasitic constraint generation problem seems to be extremely difficult, as the amount of parasitics is not decided or not known. In this chapter a methodology for parasitics constraint generation using sensitivity analysis is presented.

4.1. Limits For Parasitics

In sensitivity analysis, some of the sensitivities may be positive and some negative. However, when the layout is not available, one can not take advantage of the possible cancelation effects while generating bounds on the various parasitics. Hence, for modeling the constraints on a performance function in positive (negative) direction, only the parasitics which have positive (negative) sensitivities with respect to that performance function are considered.

Approximations to performance constraints can thus be modeled by the following inequalities:

$$\sum_{j=1}^{K_p} S_{ij}^{+} p_j \leq \Delta K_{i_{max}}^{+}, \nabla K_i \epsilon K^{+} \tag{4.1}$$

$$\sum_{j=1}^{K_p} S_{ij}^{-} p_j \leq \Delta K_{i_{max}}^{-}, \nabla K_i \epsilon K^{-} \tag{4.2}$$

where $S_{ij}^{+} = S_i$ if $S_{ij} \geq 0$; $S_{ij}^{+} = 0$ if $S_{ij} < 0$

$S_{ij}^{-} = -S_i$ if $S_{ij} \leq 0$; $S_{ij}^{-} = 0$ if $S_{ij} > 0$

In YASA, while generating sensitivities, maximum value of bound for each parasitic is also calculated for each sensitivity. These limits mean that when the maximum value of a parasitic is added to circuit, performance degradation reaches its bound.

The pseudo code of bound generation is given in Figure 4.1:

```
for EACH Performances do
  for EACH Parasitics do
    Sensitivity = CalculateSensitivity();
    if Sensitivity > 0 then
      Performance.Parasitics.Bound = (MaximumPerformanceValue - ParasiticPer-
      formanceValue) / Sensitivity;
    else
      Performance.Parasitics.Bound = (ParasiticPerformanceValue - MinimumPer-
      formanceValue) / Sensitivity;
    end if
  end for
end for
for EACH Performance.Parasitic.Bound do
  if Parasitic.Bound > Performance.Parasitic.Bound then
    Parasitic.Bound = Performance.Parasitic.Bound;
  end if
end for
```

Figure 4.1. The Pseudo Code of Bound Generation

4.2. Normalization Data

YASA generates not only sensitivity data, but it also calculates normalized sensitivity data. The pseudo code of normalized sensitivity generation is given in Figure 4.2.

4.3. Principal Component Analysis

In YASA, the output data sets are huge. Therefore, the idea to analyze the data sets in relationships between the individual points in that data set is a good way. In statistics, PCA is a way of identifying patterns in data, and express the data in such

```
for EACH Performance do
  for EACH Parasitic do
    Performance.Parasitic.Sensitivity = (Performance.Parasitic.Sensitivity - Performance.MeanSensitivity) / Performance.StandardDeviation;
  end for
end for
```

Figure 4.2. The Pseudo Code of Normalized Sensitivity Generation

a way as to highlight their similarities and differences. The main advantage of PCA is that once you find these patterns in the data, and you compress the data by reducing the number of dimensions, with considerably low loss of information. The steps of PCA process;

- Get sensitivity data
- Subtract the mean
- Calculate covariance matrix
- Calculate the eigenvectors and eigenvalues of the covariance matrix
- Choose components and forms a feature vector
- Derive the new data set

The PCA module of YASA is added using the code of F. Murtagh [27].

4.4. Summary

The method of bound parasitics calculation and generation of normalize sensitivity set was explained. Beside this, benefits of the PCA into sensitivity set and steps of the analysis are mentioned. All of these data sets are given as output files by YASA.

5. CIRCUIT ANALYSIS FUNCTIONS

The aim of this chapter is to give information about circuit analysis simulators and functions, such as AC simulator and frequency search algorithm. First, AC simulator is presented. Then general information about BSIM3 parameters is given. Third, a faster frequency search algorithm method is presented.

5.1. AC Simulator

As mentioned in Section 2.5.2, YASA uses an AC analyzer module. In YASAv1, the AC analyzer written by Selçuk Ataç has some drawbacks, like not having ability to solve modified nodal matrix or not converge the results. The AC module of YASA is upgraded to AC analyzer written by Özsun Sönmez. First, linear parameters of mosfets are calculated. These parameters are $g_{ds}, g_{mb}, g_{mbs}, i_{ds}, C_{gs}, C_{gd}, C_{gb}, C_{db}, C_{sb}$. In AC analyzer, area and perimeter of source-drain is calculated according to technology description In YASAv1, these parameters are given in the circuit file. Modifial nodal matrix is solved after linear parameters are generated. This solution gives the gain response of the circuit for the definite frequency.

5.1.1. Frequency Search Algorithm

In AC module, frequency search is realized using Newton-Raphson method. Frequency search process is run until acquiring the desired gain with a acceptable tolerance. The frequency search process can continue indefinitely. Therefore, it is necessary to define an iteration limit. Also, in frequency search algorithm it is required to define the left and right boundary of the frequency limit. The algorithm given in Figure 5.1 is used for frequency search:

In this algorithm, it is assumed that gain always decreases as the frequency increases. Therefore this algorithm gives wrong results for a positive tendency gain as the frequency increases. The user should pay attention to this method.

```
Frq = (FrqLeft + FrqRight);
Gain = Magnitude(AC(Devices, IN, OUT, Frq));
Iteration= 0;
while Absolute(Gain − GainWanted)/GainWanted > Tolerance do
  if GainWanted > Gain then
    Frq = (Frq + FrqLeft) / 2;
  else if GainWanted < Gain then
    Frq = (Frq + FrqUp) / 2;
  end if
  Iteration++;
  if Iteration > IterationMax then
    return 1;
  end if
  Gain = Magnitude(AC(Devices, IN, OUT, Frq));
end while
return Frq;
```

Figure 5.1. The Pseudo Code of Frequency Search Algorithm

5.1.2. Bandwidth Calculation

As mentioned before, YASA is a tool that finds sensitivities about designed circuit. It makes this decision by running many simulations and utilizing the results. Therefore, if simulation run faster, the results can be taken in less time. Bandwidth calculation algorithm given in Table 5.2 provides performance the YASA when finding the bandwidth of the circuit. First, for the nominal circuit bandwidth is found in a wide frequency space defined by the user. Then, bandwidth sensitivity process is run in a narrow frequency space which is also defined by the user. The pseudo-code of algorithm is given in Figure 5.2.

Similar algorithm is realized for unity gain-bandwith calculation process.

```
Frq = (FrqLeft + FrqRight);
Gain = Magnitude(AC(Devices, IN, OUT, Frq));
Iteration= 0;
while Absolute(Gain − GainWanted)/GainWanted > Tolerance do
  if GainWanted > Gain then
    Frq = (Frq + FrqLeft) / 2;
  else if GainWanted < Gain then
    Frq = (Frq + FrqUp) / 2;
  end if
  Iteration++;
  if Iteration > IterationMax then
    return 1;
  end if
  Gain = Magnitude(AC(Devices, IN, OUT, Frq));
end while
return Frq;
```

Figure 5.2. The Pseudo Code of Bandwidth Search Algorithm

5.2. BSIM3 MOSFET Model Parameters

In YASA, supported mosfet models are Level2, Level3, BSIM3. YASAv1 supports only Level2 and Level3 parameters. BSIM3 parameters are inserted in YASAv2.

BSIM3 is a physical mosfet model with extensive built-in dependencies of important dimensional and processing parameters such as channel length, width, gate oxide thickness, junction depth, substrate doping concentration, and so on [28]. It allows users to accurately model mosfet behavior over a wide range of channel lengths for present and future technologies. Using a coherent pseudo 2D formulation, major short-channel effects and high field effects such as threshold voltage roll-off, non-uniform doping effect, mobility reduction due to vertical field, carrier velocity saturation, channel-length modulation, drain induced barrier lowering, substrate current-induced body effect, subthreshold conduction, parasitic resistance effect, quantum effects, and LDD

effect are included [28]. BSIM3 parameters achieves a high level of model accuracy and computational efficiency. The effects of variations in these parameters on the device AC and DC characteristics are built into the model. More information can be found official source of BSIM3 [28]. It is not aimed to give detailed information about BSIM3 parameters in there.

The most important effect of addition of BSIM3 parameters into YASA is that trusted substrate coupling effect is inserted into tool. Because BSIM3 parameters makes much more realistic substrate model according to LEVEL2 or LEVEL3 parameters, much more realistic circuit can be simulated using mosfets modeled with BSIM3.

5.3. Summary

The new AC simulator gives trusted results for sensitivity analysis and constraint generation. In addition, consumed time for analysis is reduced using efficient algorithms. Using BSIM3 mosfet models, the physical characteristics of a mosfet are modeled better than that of Level3 or Level2. The simple but efficient frequency search algorithm provides more speed the simulator.

6. YASAV2

YASAv2 is written in C++ using object-oriented methodology. A MFC interface is used for easy usage. In this section, some results of YASA are presented using circuit examples.

6.1. Program Input

YASAv2 like YASAv1 needs a circuit netlist and a configuration file. The circuit netlist can include, MOSFETs, resistors, inductances, capacitors, voltage and current sources, sub-circuits, mosfet-model parameters. The circuit is put in a structure in YASA, and analysis steps are realized according to defined parameters such as bandwidth, phase margin, substrate coupling analysis, etc.

A configuration file contains performance and parasitic parameters, model definition parameters for substrate coupling. The commands for configuration file are given in Appendix A.

6.2. Program Output

For each parasitic two files are generated, one is a non-parasitic (nominal) performance criterion; the other one is parasitic sensitivity file. Besides this, normalized sensitivity and PCA lists are generated for each parasitic set. Also, using constraint algorithm bound definition file is generated for each parasitic.

6.2.1. Symmetric Operational Transconductance Amplifier (OTA)

The Symmetric OTA structure tested in YASA is given in Figure 6.1. The netlist of the circuit is given in Appendix D.

The circuit includes matched differential inputs, a current source path, and sym-

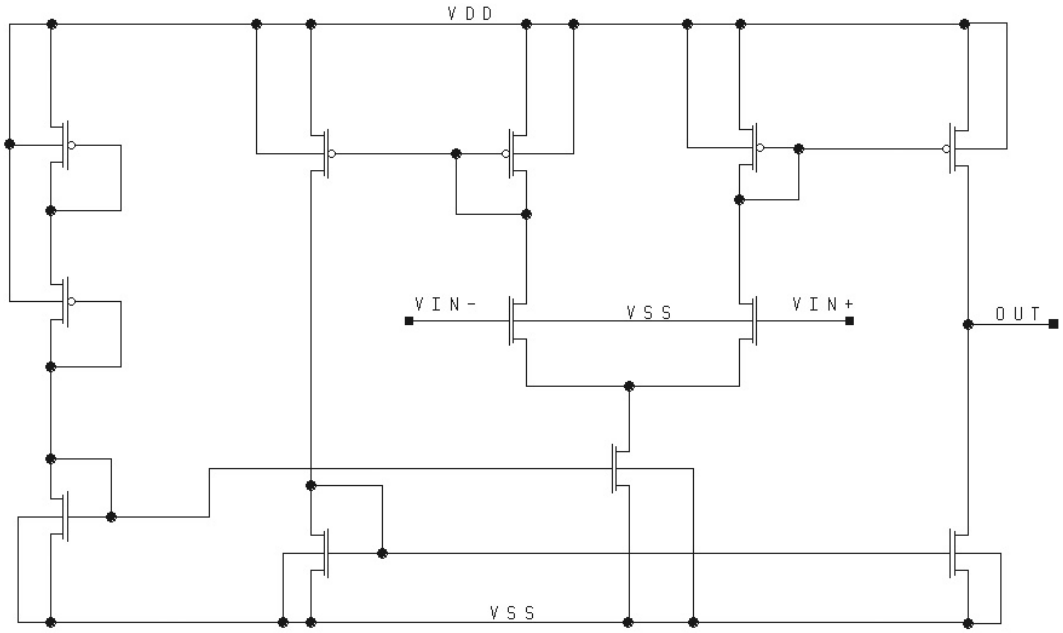

Figure 6.1. OTA

metric gain stages. The performance measure in the analysis file and nominal values for these measures generated by YASA are presented in Table 6.1.

Table 6.1. Nominal Performance Values of OTA

Performance Measure	**Nominal Value**
Power Consumption	20.15 mW
Output Offset Voltage	0.942 V
Low Frequency Gain	5.98 V/V
Gain at 100MHz	4.48 V/V
3dB Bandwidth	112.8 MHz
Unity-Gain Bandwidth	647.5 MHz
Phase Margin	84°

The results presented below are that of several different sensitivity analysis for various performance measures-parasitic effects couples. In the first analysis, the effects of capacitive coupling on bandwidth, unitygain bandwidth, gain and phase margin is analyzed and sorted individually using YASA. Tables 6.2 and 6.3 contain the first eighteen node pairs that the corresponding performance measures are most sensitive

to. Normalized sensitivities, PCA outputs, generated bounds are presented in Table 6.4, 6.5, 6.6, respectively. Threshold value is chosen as 0.1.

Normalized sensitivity results give much simpler outputs according to sensitivity results. Beside this, generated PCA results is much more simple and usable to finf the sensitive parameters. The bounds of parasitics are the other usable results of YASA.

Table 6.2. Capacitive Sensitivities-1 of Symmetric OTA

Parasitics	**Gain@100MHz** $[(V/V)/pf]$	**BW** $[Hz/pf]$
C_{0-13}	$-76.0 \cdot 10^{-3}$	0
C_{1-13}	$-76.0 \cdot 10^{-3}$	0
C_{2-13}	$-76.0 \cdot 10^{-3}$	0
C_{11-13}	$-76.0 \cdot 10^{-3}$	0
C_{11-14}	-6.2	$-352.5 \cdot 10^{+6}$
C_{12-16}	$-112.6 \cdot 10^{-3}$	0
C_{12-13}	$-72.9 \cdot 10^{-3}$	0
C_{12-14}	-6.3	$-352.5 \cdot 10^{+6}$
C_{3-6}	$-164.5 \cdot 10^{-3}$	0
C_{3-13}	$-122.2 \cdot 10^{-3}$	0
C_{4-6}	$-186.7 \cdot 10^{-3}$	0
C_{4-13}	$-151.3 \cdot 10^{-3}$	0
C_{6-8}	$-382.5 \cdot 10^{-3}$	0
C_{6-14}	-9.74	$-528.7 \cdot 10^{+6}$
C_{8-13}	$-262.2 \cdot 10^{-3}$	0
C_{8-14}	-5.2	$-281.9 \cdot 10^{+6}$
C_{13-14}	-9.3	$-493.5 \cdot 10^{+6}$
C_{14-10}	-7.2	$-3.85 \cdot 10^{+6}$

Table 6.3. Capacitive Sensitivities-2 of Symmetric OTA

Parasitics	**UGBW** $[Hz/pf]$	**PM** $[degree/pf]$
C_{0-13}	$-404.6 \cdot 10^{+6}$	-167.4
C_{1-13}	$-404.6 \cdot 10^{+6}$	-167.4
C_{2-13}	$-404.6 \cdot 10^{+6}$	-167.4
C_{11-13}	$-404.6 \cdot 10^{+6}$	-167.4
C_{11-14}	$-1.92 \cdot 10^{+9}$	45.8
C_{12-16}	$-606.9 \cdot 10^{+6}$	-214.2
C_{12-13}	$404.7 \cdot 10^{+6}$	-247.9
C_{12-14}	$2.42 \cdot 10^{+9}$	232.8
C_{3-6}	$910.5 \cdot 10^{+6}$	139.9
C_{3-13}	$-606.9 \cdot 10^{+6}$	189.9
C_{4-6}	$-1.01 \cdot 10^{+9}$	162.3
C_{4-13}	$-708.2 \cdot 10^{+6}$	214.9
C_{6-8}	$-1.82 \cdot 10^{+9}$	338.1
C_{6-14}	$-3.84 \cdot 10^{+9}$	312.7
C_{8-13}	$-1.32 \cdot 10^{+9}$	417.3
C_{8-14}	$-1.72 \cdot 10^{+9}$	371.8
C_{13-14}	$-2.63 \cdot 10^{+9}$	-477
C_{14-10}	$-2.33 \cdot 10^{+9}$	138.8

After the capacitive coupling analysis, substrate degradation effects are included in YASA and their analysis results are given in Table 6.7, 6.10, 6.11, 6.12, 6.13, respectively. Sensitivity threshold is chosen as 0 which means that all of parasitic results are calculated. Mismatch parasitic analysis results between mosfet N1 and N2 are given in Table 6.8. Process variation parasitic analysis results between mosfet N1 and N2 are given in Table 6.9.

Table 6.4. Normalized Capacitive Sensitivities of Symmetric OTA

Parasitics	**Gain@100MHz**	**BW**	**UGBW**	**PM**
C_{0-13}	0.36183	0.37656	0.10513	-1.06982
C_{1-13}	0.36183	0.37656	0.10513	-1.06982
C_{2-13}	0.36183	0.37656	0.10513	-1.06982
C_{11-13}	0.36183	0.37656	0.10513	-1.06982
C_{11-14}	-2.22530	-2.33688	-1.82659	0.29305
C_{12-16}	0.34639	0.37656	-0.15244	-1.36890
C_{12-13}	0.36312	0.37656	0.10513	-1.58459
C_{12-14}	-2.26239	-2.33688	-2.47050	1.48760
C_{3-6}	0.32445	0.37656	-0.53878	-0.89451
C_{3-13}	0.34230	0.37656	-0.15244	-1.21362
C_{4-6}	0.31509	0.37656	-0.66756	-1.03767
C_{4-13}	0.33004	0.37656	-0.28122	-1.37333
C_{6-8}	0.23239	0.37656	-1.69781	-2.16112
C_{6-14}	-3.71996	-3.69360	-4.27344	-1.99898
C_{8-13}	0.28317	0.37656	-1.05391	-2.66712
C_{8-14}	-1.80887	-1.79419	-1.56903	2.37591
C_{13-14}	-3.52675	-3.42225	-2.72806	-3.04805
C_{14-10}	-2.65351	-2.60822	-2.34172	0.88718

6.3. ALG&YASA Interface

The constraint-driven analog layout generator is aimed to consist of three separate modules. The first module, which is circuit generator, produces the corresponding netlist based on the given specifications. The second module called YASA generates the sensitivity and boundaries of circuit parameters. The third module, ALG, realizes the physical implementation of the circuit from the given netlist.

The circuit generator module has been improved by Özsun Sönmez. The mod-

Table 6.5. PCAs of Capacitive Coupling of Symmetric OTA

Parasitics	**PCA0**
C_{0-13}	-0.095253
C_{1-13}	-0.095253
C_{2-13}	-0.095253
C_{11-13}	-0.095253
C_{11-14}	0.141632
C_{12-16}	-0.093839
C_{12-13}	-0.095371
C_{12-14}	0.145028
C_{3-6}	-0.091831
C_{3-13}	-0.093465
C_{4-6}	-0.090973
C_{4-13}	-0.092342
C_{6-8}	-0.083402
C_{6-14}	0.278487
C_{8-13}	-0.088051
C_{8-14}	0.103502
C_{13-14}	0.260797
C_{14-10}	0.180840

ule implemented by him creates a netlist from the performance parameters. Then, YASA takes the netlist and iterates to find sensitive parameters using sensitivity analysis results. Finally, ALG generates the layout synchronously with YASA. It makes placement and routing of the connections between modules. YASA can generate capacitive coupling, substrate coupling, transistor mismatches sensitivities, and their bound limits. YASA gives the proper information to ALG about modules (transistors or subcircuits) which must be far from each other. This will be realized by substrate coupling, and transistor mismatches analysis results. Using substrate coupling, the placement is realized by taking into consideration minimum coupling degradation. In addition to

Table 6.6. Bounds of Significant Capacitive Coupling of Symmetric OTA

Parasitics	**Maximum Value**
C_{0-13}	489 fF
C_{1-13}	489 fF
C_{2-13}	489 fF
C_{11-13}	489 fF
C_{11-14}	275 fF
C_{12-16}	380 fF
C_{12-13}	327 fF
C_{12-14}	215 fF
C_{3-6}	587 fF
C_{3-13}	430 fF
C_{4-6}	505 fF
C_{4-13}	379 fF
C_{6-8}	237 fF
C_{6-14}	132 fF
C_{8-13}	190 fF
C_{8-14}	168 fF
C_{13-14}	165 fF
C_{14-10}	225 fF

Table 6.7. Nominal Performance Values of OTA including Substrate Coupling Model

Performance Measure	**Nominal Value**
Power Consumption	20.15 mW
Output Offset Voltage	0.942 V
Low Frequency Gain	5.98 V/V
3dB Bandwidth	82.76 MHz
Unity-Gain Bandwidth	491.2 MHz

Table 6.8. Geometric Mismatch Sensitivity Results of OTA

Sensitivity Performance of $M_{N1} - M_{N2}$ Width	**Sensitivity Value**
Power Consumption	$0.70 \cdot 10^{-3}[Watt/\mu m]$
Output Offset Voltage	$34.6 \cdot 10^{-3}[V/\mu m]$
Low Frequency Gain	$-398.6 \cdot 10^{-3}[V/V/\mu m]$
Gain at 100MHz	$-176.875 \cdot 10^{-3}[V/V/\mu m]$
3dB Bandwidth	$0[Hz/\mu m]$
Unity-Gain Bandwidth	$0[Hz/\mu m]$
Phase Margin	$587.5 \cdot 10^{-3}[degree/\mu m]$

this, transistor mismatch analysis results give information about which modules must be near to each other. Then, ALG makes placement of modules. After placement of modules, it is required to route connections between modules. Capacitive coupling results give information about routing of wires. Highly coupled wires must be as far as possible to satisfy the desired performance values. Then, ALG routes the connections taking into consideration capacitive coupling analysis outputs of YASA. Finally, ALG produces extracted netlist and YASA makes analysis to verify if the generated layout satisfies the performance criteria, or not.

Table 6.9. Process Mismatch Sensitivity Results of OTA

Sensitivity Performance of $M_{N1} - M_{N2}$ Tox	**Sensitivity Value**
Power Consumption	$-673.2 \cdot 10^{-18}[Watt/\mu m]$
Output Offset Voltage	$-712.208 \cdot 10^{-18}[V/\mu m]$
Low Frequency Gain	$305.7 \cdot 10^{-15}[V/V/\mu m]$
Gain at 100MHz	$132.7 \cdot 10^{-15}[V/V/\mu m]$
3dB Bandwidth	$0[Hz/\mu m]$
Unity-Gain Bandwidth	$0[Hz/\mu m]$
Phase Margin	$-545.9 \cdot 10^{-15}[degree/\mu m]$

Table 6.10. Sensitivities-1 of Substrate coupling of Symmetric OTA

Parasitics	**PWR**$[W]/[pF, ohm]$	**Offset** $[V]/[pF, ohm]$
Rbs_{MN1}	256.3^{-15}	-6.8^{-12}
Rbs_{MN2}	203.0^{-15}	7.2^{-12}
Rbs_{MN3}	-635.6^{-15}	177.2^{-15}
Rbs_{MN4}	3.0^{-15}	293.5^{-15}
Rbs_{MN5}	3.7^{-15}	-679.9^{-15}
Rbs_{MN6}	44.9^{-15}	-303.3^{-15}
Rbb_{MN1-2}	24.4^{-21}	2.7^{-18}
Rbb_{MN1-3}	-45.6^{-18}	-202.7^{-18}
Rbb_{MN1-4}	-152.9^{-18}	5.8^{-15}
Rbb_{MN1-5}	-147.9^{-18}	-887.7^{-18}
Rbb_{MN1-6}	-468.9^{-18}	-283.5^{-18}
Rbb_{MN2-3}	-40.3^{-18}	21.8^{-18}
Rbb_{MN2-4}	-127.2^{-18}	-2.0^{-15}
Rbb_{MN2-5}	-122.4^{-18}	-8.7^{-15}
Rbb_{MN2-6}	-471.6^{-18}	-999.7^{-18}
Rbb_{MN3-4}	71.7^{-18}	210.6^{-18}
Rbb_{MN3-5}	78.5^{-18}	-9.2^{-15}
Rbb_{MN3-6}	-988.9^{-18}	-1.6^{-15}
Rbb_{MN4-5}	77.4^{-21}	-8.0^{-18}
Rbb_{MN4-6}	-618.0^{-18}	5.6^{-15}
Rbb_{MN5-6}	-610.8^{-18}	-5.1^{-15}
Cbs_{MN1}	0	0
Cbs_{MN2}	0	0
Cbs_{MN3}	0	0
Cbs_{MN4}	0	0
Cbs_{MN5}	0	0
Cbs_{MN6}	0	0

Table 6.11. Sensitivities-2 of Substrate coupling of Symmetric OTA

Parasitics	**UGBW** $[Hz]/[pF, ohm]$	**PM** $[degree]/[pF, ohm]$
Rbs_{MN1}	305.3^{-12}	81.8^{-6}
Rbs_{MN2}	305.3^{-12}	64.7^{-6}
Rbs_{MN3}	159.9^{-12}	-2.8^{-6}
Rbs_{MN4}	43.5^{-15}	1.4^{-6}
Rbs_{MN5}	43.5^{-12}	1.4^{-6}
Rbs_{MN6}	87.2^{-12}	-32.4^{-9}
Rbb_{MN1-2}	3.6^{-12}	40.1^{-9}
Rbb_{MN1-3}	3.8^{-12}	17.9^{-9}
Rbb_{MN1-4}	4.2^{-12}	57.6^{-9}
Rbb_{MN1-5}	4.2^{-12}	-12.0^{-9}
Rbb_{MN1-6}	4.0^{-12}	4.4^{-9}
Rbb_{MN2-3}	3.8^{-12}	5.0^{-9}
Rbb_{MN2-4}	4.2^{-12}	-310.8^{-12}
Rbb_{MN2-5}	4.17^{-12}	53.4^{-9}
Rbb_{MN2-6}	4.0^{-12}	18.1^{-9}
Rbb_{MN3-4}	4.9^{-12}	9.5^{-9}
Rbb_{MN3-5}	4.9^{-12}	9.5^{-9}
Rbb_{MN3-6}	4.6^{-12}	-4.8^{-9}
Rbb_{MN4-5}	6.2^{-12}	127.9^{-9}
Rbb_{MN4-6}	5.5^{-12}	35.7^{-9}
Rbb_{MN5-6}	5.5^{-12}	20.1^{-9}
Cbs_{MN1}	268.9^{-6}	34.9
Cbs_{MN2}	268.9^{-6}	9.8
Cbs_{MN3}	537.9^{-6}	-7.5
Cbs_{MN4}	2.7^{-3}	50.0
Cbs_{MN5}	2.7^{-3}	-89.2
Cbs_{MN6}	1.1^{-3}	170.9^{-3}

Table 6.12. Normalized Sensitivities of Substrate coupling of Symmetric OTA

Parasitics	**Power**	**Offset**	**UGBW**	**PM**
Rbs_{MN1}	1.86	−3.49	−0.38	0
Rbs_{MN2}	1.48	3.70	−0.38	0
Rbs_{MN3}	−4.49	0.09	−0.38	0
Rbs_{MN4}	0.06	0.15	−0.38	0
Rbs_{MN5}	0.06	−0.35	−0.38	0
Rbs_{MN6}	0.35	−0.15	−0.38	0
Rbb_{MN1-2}	0.03	0	−0.38	0
Rbb_{MN1-3}	0.03	0	−0.38	0
Rbb_{MN1-4}	0.03	0	−0.38	0
Rbb_{MN1-5}	0.03	0	−0.38	0
Rbb_{MN1-6}	0.03	0	−0.38	0
Rbb_{MN2-3}	0.03	0	−0.38	0
Rbb_{MN2-4}	0.03	0	−0.38	0
Rbb_{MN2-5}	0.03	0	−0.38	0
Rbb_{MN2-6}	0.03	0	−0.38	0
Rbb_{MN3-4}	0.03	0	−0.38	0
Rbb_{MN3-5}	0.04	0	−0.38	0
Rbb_{MN3-6}	0.03	0	−0.38	0
Rbb_{MN4-5}	0.03	0	−0.38	0
Rbb_{MN4-6}	0.03	0	−0.38	0
Rbb_{MN5-6}	0.03	0	−0.38	0
Cbs_{MN1}	0.03	0	-0.01	1.64
Cbs_{MN2}	0.03	0	-0.01	0.46
Cbs_{MN3}	0.03	0	0.35	-0.35
Cbs_{MN4}	0.03	0	3.29	2.35
Cbs_{MN5}	0.03	0	3.29	-4.18
Cbs_{MN6}	0.03	0	1.09	0

Table 6.13. PCM0 of Sensitivities of Substrate coupling of Symmetric OTA

Parasitics	**PCM0**
Rbs_{MN1}	$2.1 \cdot 10^{-3}$
Rbs_{MN2}	$1.7 \cdot 10^{-3}$
Rbs_{MN3}	$-5.09 \cdot 10^{-3}$
Rbs_{MN4}	$0.06 \cdot 10^{-3}$
Rbs_{MN5}	$0.07 \cdot 10^{-3}$
Rbs_{MN6}	$0.4 \cdot 10^{-3}$
Rbb_{MN1-2}	$0.04 \cdot 10^{-3}$
Rbb_{MN1-3}	$0.04 \cdot 10^{-3}$
Rbb_{MN1-4}	$0.04 \cdot 10^{-3}$
Rbb_{MN1-5}	$0.04 \cdot 10^{-3}$
Rbb_{MN1-6}	$0.04 \cdot 10^{-3}$
Rbb_{MN2-3}	$0.04 \cdot 10^{-3}$
Rbb_{MN2-4}	$0.04 \cdot 10^{-3}$
Rbb_{MN2-5}	$0.04 \cdot 10^{-3}$
Rbb_{MN2-6}	$0.04 \cdot 10^{-3}$
Rbb_{MN3-4}	$0.04 \cdot 10^{-3}$
Rbb_{MN3-5}	$0.04 \cdot 10^{-3}$
Rbb_{MN3-6}	$0.04 \cdot 10^{-3}$
Rbb_{MN4-5}	$0.04 \cdot 10^{-3}$
Rbb_{MN4-6}	$0.03 \cdot 10^{-3}$
Rbb_{MN5-6}	$0.03 \cdot 10^{-3}$
Cbs_{MN1}	$0.04 \cdot 10^{-3}$
Cbs_{MN2}	$0.04 \cdot 10^{-3}$
Cbs_{MN3}	$0.04 \cdot 10^{-3}$
Cbs_{MN4}	$0.04 \cdot 10^{-3}$
Cbs_{MN5}	$0.04 \cdot 10^{-3}$
Cbs_{MN6}	$0.04 \cdot 10^{-3}$

6.4. Summary

In this chapter, an example OTA circuit is analyzed using YASA. Using configuration commands, desired analysis results are acquired. PCA outputs of analysis are presented to decide the important parasitic set in an easy way. Finally, ALG&YASA association is presented.

7. CONCLUSIONS

In this thesis, a performance oriented analysis tool is presented. It gives information about how much the analyzed circuit is affected by the parasitics, which are either introduced during layout synthesis or the manufacturing process, on the performance of analog circuits.

The tool helps circuit designers by giving information about the most critical points in the circuits. The aim of this work is to prepare YASA to integrate with ALG. Then, a constraint-driven analog layout synthesizer can be built. Because the tool is written in a object-oriented way, it can be accessed and improved easily. Also, configurable structure of YASA provides easy access to the supported functions. Another important property of YASA is that, it can be used in different environment. Although this version is based on WINDOWS operating systems, by changing of only a few lines, it can be used in LINUX or other operating systems.

Because BSIM3 mosfet model is much more near the real behavior of the mosfet, designers want to work with BSIM3 models. Addition of BSIM3 mosfet parameter into YASA provides much more realistic results. In the future, BSIM4 parameter can be added to YASA.

Another important issue in a circuit is the substrate coupling problem. It causes different results between modeled and realized circuits. YASA adds simple capacitive and resistive substrate coupling effect into analyzed circuit.

In YASA, supported performance functions only DC and AC. In the future, transient analysis module can be added.

APPENDIX A: Analysis Configration Commands

The user can define or configure desired output parameters of YASA. The syntax of the commands are given below:

COMMAND Parameter1, Parameter2, ... , Parameter[N]

The set of commands provided below can be used to tailor the analysis setup to fit the requirements of any analog circuit.

FRQ_L	*Value* The left coefficient to search the frequency for the perturbated circuit
FRQ_H	*Value* The right coefficient to search the frequency for the perturbated circuit
FRQ_MAX	*Value* Maximum value of the frequency used in analysis
POWER	*MinVal, MaxVal*
OFFSET	*OffsetNode,* V_{ref}*, MinVal, MaxVal* Creates an instance of Performance-Offset for calculating the offset at *OffsetNode* with respect to the reference voltage V_{ref}.
GAIN	*NodeOut,NodeIn,Freq,MinVal,MaxVal* Creates an instance of Performance-GainAtFreq for calculating the magnitude of the transfer function from *NodeIn* to *NodeOut* evaluated at the frequency *Freq.*

BW	*NodeOut,NodeIn,MinVal,MaxVal* Creates an instance of Performance-BW for calculating the -3dB bandwidth of the transfer function from *NodeIn* to *NodeOut*.
UGBW	*NodeOut,NodeIn,MinVal,MaxVal* Creates an instance of Performance-UGBW for calculating the unity-gain bandwidth of the transfer function from *NodeIn* to *NodeOut*
PM	*NodeOut,NodeIn,MinVal,MaxVal* Creates an instance of Performance-PM for calculating the phase margin of the transfer function from *NodeIn* to *NodeOut*.
CC-AUTO	Automatically creates instances of parasitic-CapacitiveCoupling for all pairs of nets that exist in the circuit.
CC	*MinVal,MaxVal,Node1,Node2* Creates an instance of parasitic-CapacitiveCoupling for the nets defined by *Node1* and *Node2* with minimum and maximum values *MinVal* and *MaxVal*.
SC-AUTO	Automatically creates instances of substrate coupling parasitic sets for all the wells containing two or more mosfets.
SC	*SubstrateNode* Creates an instance of substrate coupling parasitic set for the well defined by *SubstrateNode*.

SC-CoeffN *Value*

Substrate coupling coefficient K_n.

SC-CoeffP *Value*

Substrate coupling coefficient K_p.

SC-Distance *Value*

d_{ij} is the minimum technology space unit in meter between mosfets.

SC-Coeff1 *Value*

Defines the K_1 for the substrate.

SC-Coeff2 *Value*

Defines the K_2 for the substrate.

SC-Coeff3 *Value*

Defines the K_3 for the substrate.

SC-Coeff4 *Value*

Defines the β for substrate coupling.

SC-GammaN *Value*

$\Omega_{/\mu m}\varphi$ Defines the unit resistance per micrometers of the substrate and φ product for N-Type.

SC-GammaP *Ohms*

$\Omega_{/\mu m}\varphi$ Defines the unit resistance per micrometers of the substrate and φ product for P-Type.

SC-Alfa1 *Value*

Defines the α_1 for the substrate coupling.

SC-Alfa2 *Value*

Defines the α_1 for the substrate coupling.

SC-Alfa3 *Value*

Defines the α_1 for the substrate coupling.

SiThickness *Value*

Defines the thickness of Si for substrate coupling.

WELL *WellType*

Chosen well type for substrate coupling (default is Nwell).

TSMC0.18

TSMC0.18 specific process parameter defined in YASA for substrate coupling

PR *ResistorIndex, MaxDeviation*

Creates an instance of parasitic-Resistance for the resistor defined by *ResistorIndex*.

GEOMISMATCH-AUTO

Automatically perturbes all transistor pairs in the same well in the circuit.

MGP *MOS1,MOS2,ParamIndex, MaxDeviation*

Creates an instance of parasitic-PMGP for the geometry parameters defined by *ParamIndex* of the pair of mosfets *MOS1* and *MOS2*.

MMP *MOS1,MOS2,ParamIndex, MaxDeviation*

Creates an instance of parasitic-PMMP for the model parameters defined by *ParamIndex* of the pair of mosfets *MOS1* and *MOS2*.

GND *GroundNode*

Defines the ground node of the circuit. (Overrides the default value of 0)

SMGP *MOS1,ParamIndex, MaxDeviation*

Creates an instance of parasitic-SMGP for the geometry parameters defined by *ParamIndex* of the mosfet *MOS1*.

SMMP *MOS1,ParamIndex, MaxDeviation*

Creates an instance of parasitic-SMMP for the model parameters defined by *ParamIndex* of the mosfet *MOS1*.

MGPF *Factor*

Overrides the default value 0.0005 of the factor used for calculating the perturbation amount of mosfet geometry parameters.

MMPF *Factor*

Overrides the default value 0.0005 of the factor used for calculating the perturbation amount of mosfet model parameters.

STH *Threshold*

Overrides the default value 0.001 of the simplification threshold used for eliminating insignificant parasitic effects.

APPENDIX B: Level3 mosfet model parameters

Table B.1: Level-3 model parameters of mosfets

Index	Parameter	Description
1	VTO	Zero-bias threshold voltage
2	KP	Transconductance parameter
3	GAMMA	Body-effect parameter
4	PHI	Surface inversion potential
5	LAMBDA	Channel-length modulation
6	RD	Drain ohmic resistance
7	RS	Source ohmic resistance
8	CBD	-
9	CBS	-
10	IS	Bulk-junction saturation current
11	PB	Bulk-junction potential
12	CGSO	Gate-source overlap capacitance per meter
13	CGDO	Gate-drain overlap capacitance per meter
14	CGBO	Gate-bulk overlap capacitance per meter
15	RSH	Source and drain sheet resistance
16	CJ	Zero-bias bulk capacitance per square-meter
17	MJ	Bulk-junction grading coefficient
18	CJSW	Zero-bias perimeter capacitance per square meter
19	MJSW	Perimeter capacitance grading coefficient
20	JS	Bulk-junction saturation current per meter-square
21	TOX	Thin oxide thickness
22	NSUB	Substrate doping
23	NSS	Surface state density
24	NFS	Surface-fast state density
25	TPG	Type of gate material

Continued on next page

Table B.1. – continued from previous page

Index	**Parameter**	**Description**
26	XJ	Metallurgical junction depth
27	LD	Literal diffusion
28	UO	Surface mobility
29	UCRIT	Critical electric field for mobility
30	UEXP	Exponential coefficient for mobility
31	UTRA	Transverse field coefficient
32	VMAX	Maximum drift velocity of carriers
33	NEFF	Total channel charge coefficient
34	XQC	Coefficient of channel charge share
35	KF	Flicker-noise coefficient
36	AF	Flicker-noise exponent
37	FC	Bulk-junction forward bias coefficient
38	DELTA	Width effect on threshold voltage
39	THETA	Mobility modulation
40	ETA	Static feedback on threshold voltage
41	KAPPA	-

APPENDIX C: BSIM3 mosfet model parameters

It is not given all of the BSIM3 parameters and their description. Only added parameters for sensitivity analysis are presented in Table C.1.

Table C.1. BSIM3 model parameters of mosfet

Index	Parameter	Description
1	VTH0	Threshold voltage @Vbs=0
2	K1	First order body effect coefficient
3	K2	Second order body effect coefficient
4	GAMMA1	Body-effect coefficient near the surface
5	GAMMA2	Body-effect coefficient in the bulk
6	TOX	Gate oxide thickness
7	NSUB	Substrate doping concentration
8	NLX	Lateral non-uniform doping parameter
9	U0	Mobility at Temp = TNOM

APPENDIX D: Netlist of OTA

****

vdd 1 0 2.5

vss 2 0 -2.5

vinp 11 0 AC 1

vinn 12 0 DC 0

mP1 3 3 1 1 pfet w=14.8u l=0.5u

mP2 4 4 3 1 pfet w=15.2u l=0.5u

mP2 4 4 3 1 pfet w=15.2u l=0.5u

mP3 6 8 1 1 pfet w=20u l=0.5u

mP4 9 8 1 1 pfet w=20u l=0.5u

mP5 13 13 1 1 pfet w=20u l=0.5u

mP6 14 13 1 1 pfet w=20u l=0.5u

mN1 9 11 10 2 nfet w=140u l=0.8u

mN2 13 12 10 2 nfet w=140u l=0.8u

mN3 10 5 2 2 nfet w=20u l=0.5u

mN4 6 7 2 2 nfet w=12u l=0.5u

mN5 14 6 2 2 nfet w=12u l=0.5u

mN6 4 5 2 2 nfet w=10u l=0.5u

C14 14 0 300f

.MODEL nfet NMOS (LEVEL = 8 +VERSION = 3.1 TNOM = 27 TOX = 3.11E-8 +XJ = 3E-7 NCH = 7.5E16 VTH0 = 0.5226871 +K1 = 0.9213463 K2 = -0.0814952 K3 = 4.1421676 +K3B = -1.9875411 W0 = 1.685077E-6 NLX = 1E-8 +DVT0W = 0 DVT1W = 0 DVT2W = 0 +DVT0 = 0.8910709 DVT1 = 0.2700411 DVT2 = -0.2009103 +U0 = 669.3870114 UA = 1.764533E-9 UB = 1.085708E-18 +UC = 3.829457E-11 VSAT = 1.10563E5 A0 = 0.5636702 +AGS = 0.1074063 B0 = 2.428059E-6 B1 = 5E-6 +KETA = -4.173792E-3 A1 = 0 A2 = 1 +RDSW = 2.954897E3 PRWG = -0.0608257 PRWB = -0.0397134 +WR = 1 WINT = 6.852624E-7 LINT = 2.190496E-7 +XL = 0 XW = 0 DWG = -2.146929E-8 +DWB = 4.267813E-8 VOFF = 0 NFACTOR =

0.6186703 +CIT = 0 CDSC = 0 CDSCD = 0 +CDSCB = 5.3651E-6 ETA0 = -1 ETAB = -0.5327237 +DSUB = 1 PCLM = 1.3190106 PDIBLC1 = 0.0129789 +PDIBLC2 = 3.783518E-3 PDIBLCB = -0.1 DROUT = 0.0907944 +PSCBE1 = 5.501192E9 PSCBE2 = 1.260724E-9 PVAG = 0.2557599 +DELTA = 0.01 RSH = 53.2 MOBMOD = 1 +PRT = 0 UTE = -1.5 KT1 = -0.11 +KT1L = 0 KT2 = 0.022 UA1 = 4.31E-9 +UB1 = -7.61E-18 UC1 = -5.6E-11 AT = 3.3E4 +WL = 0 WLN = 1 WW = 0 +WWN = 1 WWL = 0 LL = 0 +LLN = 1 LW = 0 LWN = 1 +LWL = 0 CAPMOD = 2 XPART = 0.5 +CGDO = 1.79E-10 CGSO = 1.79E-10 CGBO = 1E-9 +CJ = 2.633758E-4 PB = 0.9823017 MJ = 0.5653097 +CJSW = 1.569584E-10 PBSW = 0.99 MJSW = 0.1 +CJSWG = 6.4E-11 PBSWG = 0.99 MJSWG = 0.1 +CF = 0)
.MODEL pfet PMOS (LEVEL = 8 +VERSION = 3.1 TNOM = 27 TOX = 3.11E-8 +XJ = 3E-7 NCH = 2.4E16 VTH0 = -0.8476404 +K1 = 0.4513608 K2 = 2.379699E-5 K3 = 13.3278347 +K3B = -2.2238332 W0 = 9.577236E-7 NLX = 7.534987E-7 +DVT0W = 0 DVT1W = 0 DVT2W = 0 +DVT0 = 1.1623904 DVT1 = 0.3238803 DVT2 = -0.0499006 +U0 = 236.8923827 UA = 3.833306E-9 UB = 1.487688E-21 +UC = -1.08562E-10 VSAT = 1.844017E5 A0 = 0.2402221 +AGS = 0.1463634 B0 = 3.616274E-6 B1 = 5E-6 +KETA = -1.069813E-3 A1 = 0 A2 = 0.364 +RDSW = 1.928149E3 PRWG = 0.1775431 PRWB = -0.1874418 +WR = 1 WINT = 7.565065E-7 LINT = 8.486462E-8 +XL = 0 XW = 0 DWG = -2.13917E-8 +DWB = 3.857544E-8 VOFF = -0.0877184 NFACTOR = 0.2508342 +CIT = 0 CDSC = 2.924806E-5 CDSCD = 1.497572E-4 +CDSCB = 1.091488E-4 ETA0 = 0.26103 ETAB = -0.0163174 +DSUB = 0.2873 PCLM = 0.015015 PDIBLC1 = 1.001279E-3 +PDIBLC2 = 1E-3 PDIBLCB = -1E-3 DROUT = 0.0237243 +PSCBE1 = 3.353579E9 PSCBE2 = 5.021337E-10 PVAG = 14.984985 +DELTA = 0.01 RSH = 77.9 MOBMOD = 1 +PRT = 0 UTE = -1.5 KT1 = -0.11 +KT1L = 0 KT2 = 0.022 UA1 = 4.31E-9 +UB1 = -7.61E-18 UC1 = -5.6E-11 AT = 3.3E4 +WL = 0 WLN = 1 WW = 0 +WWN = 1 WWL = 0 LL = 0 +LLN = 1 LW = 0 LWN = 1 +LWL = 0 CAPMOD = 2 XPART = 0.5 +CGDO = 2.33E-10 CGSO = 2.33E-10 CGBO = 1E-9 +CJ = 3.035635E-4 PB = 0.7392918 MJ = 0.4278001 +CJSW = 1.82197E-10 PBSW = 0.99 MJSW = 0.1406384 +CJSWG = 3.9E-11 PBSWG = 0.99 MJSWG = 0.1406384 +CF = 0) .END

REFERENCES

1. Charbon, E., *Constraint-Driven Analysis and Synthesis of High-Performance Analog IC Layout*, Ph.D. Thesis, EECS Department, University of California, Berkeley, 1995, http://www.eecs.berkeley.edu/Pubs/TechRpts/1995/2952.html.

2. Charbon, E., E. Malavasi and A. Sangiovanni-Vincentelli, "Generalized Constraint Generation for Analog Circuit Design", pp. 408–414, citeseer.ist.psu.edu/charbon93generalized.html.

3. Chang, H., E. Charbon, U. Choudhury, A. Demir, E. Felt, E. Liu, A. Malvasi, A. L. Sangiovanni-Vincentelli and I. Vassiliou, *A Top-Down, Constraint-Driven Design Methodology for Analog Integrated Circuits*, Kluwer Academic Publishers, second edition, 2000.

4. Choudhury, U. and A. Sangiovanni-Vincentelli, "Constraint Based-Channel Routing For Analog and Digital Circuits", *Proceeding IEEE ICCAD*, pp. 198–201, Novamber 1990.

5. Atac, M. S., *A Flexible and High Performance Simulation Based Sensitivity Analysis Tool For Analog Layout Constraint Generation*, M.S. Thesis, Boğaziçi University, 2003.

6. Choudhury, U. and A. Sangiovanni-Vincentelli, "Use of Performance Sensitivities In Routing of Analog Circuits", *Circuits and Systems IEEE International Symposium*, Vol. 1, pp. 348–351, May 1990.

7. Malavasi, E., E. Charbon, E. Felt and A. Sangiovanni-Vincentelli, "Automation of IC Layout with Analog Constraints", *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, Vol. 15, No. 8, pp. 923–942,, August 1996.

8. Dündar, G., S. Balkir and A. S. Öğrenci, *Analog VLSI Design Automation*, CRC Press, New York, 2003.

9. Chang, H., E. Charbon and A. Sangiovanni-Vincentelli, "Automatic generation of parasitic constraints for performance-constrained physical design of analog circuits", *IEEE Transaction on CAD*, Vol. 12, pp. 208–224, February 1993.

10. Bayraktoğlu, I., *Circuit simulation based training algorithms for analog neural networks*, M.S. Thesis, Boğaziçi University, 1996.

11. Su, D., M. Loinaz, S. Masui and B. Wooley, "Experimental results and modeling techniques for substrate noise in mixed-signal integrated circuits", *IEEE Journal in Solid-State Circuits*, Vol. 28, No. 4, pp. 420–430, April 1993.

12. Lan, H., *Synthsized compact models for substrate noise coupling in mixed-signal Ics*, Ph.D. Thesis, Stanford University, March 2006.

13. Verghese, N., D. Allstot and S. Masui, "Rapid Simulation of Substrate Coupling Effects in Mixed-Mode ICs", *IEEE Custom Integrated Circuits Conference*, pp. 18.3.1–18.3.4, 1993.

14. Vincent, M., "Analysis of Modeling Techniques for Substrate Noise Coupling", *ECE1352 Analog IC Lecture Notes University of Toronto*, 2001.

15. Roach, G., *Green's Functions*, Cambridge University Press, Cambridge, second edition, 1970.

16. Gharpurey, R. and R. Meyer, "Modeling and Analysis of Substrate Coupling in Integrated Circuits", *IEEE Journal of Solid State Circuits*, Vol. 31, No. 3, pp. 344–353, 1996.

17. Smedes, T., N. van der Meijs and A. van Genderen, "Boundary element methods for capacitance and substrate resistance calculations in VLSI layout verification package", *Proc. of ELEC THROSOFT'93*, pp. 337–344, July 1993.

18. Verghese, N., D. Allstot and M. Wolfe, "Verification techniques for substrate coupling and their application to mixed-signal IC design", *IEEE Journal of Solid-State Circuits*, Vol. 31, pp. 354–365, March 1996.

19. Verghese, N. and D. Allstot, "Computer-aided design consideration for mixed-signal coupling in RF integrated circuits", *IEEE Journal of Solid-State Circuits*, Vol. 33, pp. 314–323, March 1998.

20. Niknejad, A., R. Gharpurey and R. Meyer, "Numerically Stable Green Function for Modeling and Analysis of Substrate Coupling in ICs", *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, Vol. 17, No. 4, pp. 305–315, April 1998.

21. Charbon, E., R. Gharpurey, R. Meyer and A. Sangiovanni-Vincentelli, "Substrate optimization based on semi-analytical techniques", *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, Vol. 18, pp. 172–190, February 1999.

22. Aragones, X., J. Gonzalez and A. Rubio, *Analysis and solutions for switching noise coupling in mixed-signal ICs*, Kluwer Academic Publishers, Boston, MA, USA, 1999.

23. Schrik, E. and N. van der Meijs, "Combined bem/fem substrate resistance modeling", *Proc. of IEEE Design Automation Conf. (DAC'02)*, pp. 771–776, June 2002.

24. Ozis, D., T. Fiez and K. Mayaram, "A comprehensive geometry-dependent macromodel for substrate noise coupling in heavily doped CMOS processes", *in Proc. of IEEE Custom Integrated Circuits Conf. (CICC'02)*, pp. 497–500, September 2002.

25. Samavedam, A., A. Sadate and K. Mayaram, "A Scalable Substrate Noise Coupling Model for Design of Mixed-Signal IC's", *IEEE Journal of Solid-State Circuits*, Vol. 35, No. 6, June 2000.

26. Meijs, N., F. Beeftink and X. Burgerhout, "Space: a tool for VLSI Physical Design and Verification, Delft University", http://www.space.tudelft.nl.

27. Murtagh, F., "Principal Component Analysis Algorithm", Munich, June 1989.

28. Hu, C. and X. J. Xi, "BSIM3: a physics-based, accurate, scalable, robustic and predictive MOSFET SPICE model, BSIM Research Group in the EECS at the University of California, Berkeley", http://www-device.eecs.berkeley.edu/ bsim3/intro.html.