SAAD DAHLAB UNIVERSITY OF BLIDA

Faculty of Sciences

Physic's Department

MASTER THESIS

Option : Materials and Components

LINEAR REGION PARAMETER EXTRACTION

USING FAST SIMULATED DIFFUSION METHOD

FOR DEEP SUBMICROMETER MOSFET

By

Naas CHARRAK

In front of the jury made up of :

K. Ferdjani	M. C.	U. S. D. B.	Chairman
M. T. Belaroussi	M. R.	C. D. T. A.	Director
M. Boumaour	M. R.	U. D. T. S.	Examiner
D. Ouadjaout	M. R.	U. D. T. S.	Examiner
Y. Harabi	C. R.	C. D. T. A.	Co –Director

Blida, March 2005

ABSTRACT

As devices are scaled down into sub-micrometer regime, many approaches have proposed in the linear region to determine MOSFET parameters such as threshold voltage, effective mobility, effective channel length, effective channel width, parasitic series resistance, and saturation velocity,

These parameters are required during the design phase of an integrated circuit for simulation and tuning of the circuit's electrical behavior. but these methods works under such considerations which may not valid for other models.

The new in this thesis, linear region parameters of deep sub-micrometer MOSFET for BSIM3v3 model are extracted simultaneously using a multi-dimensional global optimization method named “ Fast Simulated Diffusion method “.

ملخص

عند تكامل العناصر الالكترونية الى جزء من الميكرون، أقترحت عدة طرق في المنطقة الخطية لتعيين وسائط الــMOSFET مثل جهد العتبة، الحركية، طول القناة الفعال،عرض القنال الفعال، تشويش مقاومة التسلسل، و سرعة التشبع.
هذه الوسائط لازمة لإتمام طور التصميم للدارات المتكاملة للمحاكاة و الضبط الدقيق للسلوك الكهربائي للدارة. لكن هذه الطرق و تحت إعتبارات معينة لا يمكن أن تكون صالحة لنماذج أخرى.
الجديد في هذا العمل، وسائط المنطقة الخطية للـMOSFET ذو الجزء من الميكرون العميق لنموذجBSIM3v3. تستخرج في نفس الوقت بإستعمال طريقة محسنة متعددة الأبعاد الشاملة تدعى : "طريقة محاكات الإنتشار السريعة "

RÉSUME

Comme les dispositifs sont réduits en régime sub-micrométrique, différentes approches ont proposé la détermination des paramètres du MOSFET dans la région linéaire, la tension de seuil, la mobilité effective, la longueur effective du canal, les résistances séries parasites, ainsi que la vitesse de saturation.

Ces paramètres sont requis pendant la phase de conception du circuit intégré, pour la simulation et l’ajustement du comportement électrique de ce dernier. Toutefois ces méthodes sont intrinsèques au modèle étudié et ne peuvent être appliquées à d’autres modèles.

Cette étude concerne l’extraction des paramètres du modèle BSIM3v3 submicronique profond du MOSFET en considérant une optimisation multi-dimensionnelle globale et en utilisant la méthode appelée « Diffusion simulée rapide »

ACKNOWLEDGMENTS

First of all , I would like to express my sincere gratitude and appreciation to my advisor, Dr. M. T. Belaroussi for his invaluable guidance, advice, and encouragement.

Also many thanks to my co-advisor Mr. Y. Harabi for his guidance and many helpful discussions.

I would also like to express my gratitude for the members of the jury Dr. K. Ferdjani, Dr. M. Boumaour, and Dr. D. Oudjauot, for their careful reading of the thesis and valuable suggestions.

I would like to thank all the faculty members and all my friends for their excellent classes and support, especially Bencherif Mohamed, and Bouchenafa Mohamed.

TABLE OF CONTENTS

ABSTACT

AKNOWLEDGEMENTS

TABLE OF CONTENTS

LIST OF ILLUSTRATIONS, GRAPHICS AND TABLES

INTRODUCTION 7

1. DEVELOPMENTS OF THE COMPACT MODELS 10

 1.1.Introduction 10

 1.2. History of MOS modeling 11

 1.3. Review of MOSFET operation 19

2. MODEL EQUATIONS 26

 2.1. Introduction 26

 2.2. Effective Channel Width and Length 27

 2.3. Effective Voltages 27

 2.4. Threshold Voltage Modeling 28

 2.5. Mobility 29

 2.6. Source/Drain Resistance 30

 2.7. Bulk Charge 30

 2.8. Saturation Field 31

 2.9. Saturation Voltage 31

 2.10. Drain Current Modeling 31

 2.11. Early voltage modeling 33

Conclusion 34

3. OPTIMIZATION METHODS 35

 3.1 Introduction 35

 3.2 Optimization problems 35

 3.3 A Taxonomy of optimization problems 36

 3.4 Gradient following methods 40

 3.5 Direct searching methods 45

 3.6 Combined Methods 52

Conclusion 53

4. FAST SIMULATED DIFFUSION METHOD 54
4.1 Introduction 54
4.2 Parameter extraction 55
4.3 Simulated Diffusion 56
4.4 Fast Simulated Diffusion method 56
Conclusion 61
5. RESULTS AND DISCUSSION 62
5.1 Introduction 62
5.2. Test functions 64
5.3. BSIM3v3 results 65
Conclusion 71
CONCLUSION 72
APPENDICES 73
REFERENCES 80

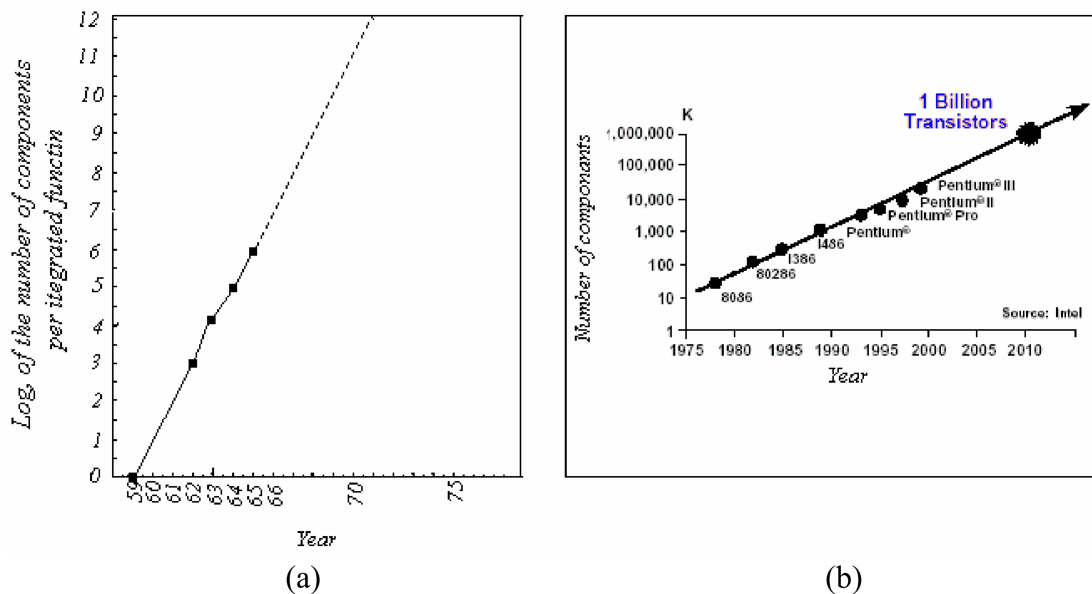
LIST OF ILLUSTRATIONS, GRAPHICS AND TABLES

Figure 1.1.	Moor's law, (a) the number of transistors on a chip	7
Figure 1.2.	The three generations of compact models	12
Figure 1.3.	Schematic of LEVEL 1, 2 and 3 MOSFET models	13
Figure 1.4.	A basic n-MOSFET structure	20
Figure 1.5.	Top view of an n-type MOSFET	21
Figure 1.6.	I-V characteristics of an n-type MOSFET	22
Figure 1.7.	Schematic illustration of the induced n-channel	23
Figure 1.8.	Illustrates the extraction of VT=V_{th}	24
Figure 1.9.	I_{ds} versus V_{ds} characteristics for different values of V_{gs}	25
Figure 3.1.	Classification of optimization problems	37
Figure 3.2.	Illustration of the geometrical operations in the simplex direct searching optimization method	46
Figure 3.3.	Illustration of the search path for Powell's conjugate direction method.	48
Figure 3.4.	The vector X' is always accepted as the new parameter vector X_{k+1}.	49
Figure 3.5.	The structure of the simulated annealing algorithm	50
Figure 4.1.	Search path toward the minimum	57
Figure 4.2.	Boltzmann's distribution curve	57
Figure 4.3.	X_{next} generation using gradient information in one dimension	59
Figure 4.4.	A rough flowchart of the Fast Simulated Diffusion	60
Figure 5.1.	Simulation of MOSFET I_{ds} versus V_{ds} characteristic using Level-1, Level-3, and the BSIM3v3 models	63
Figure 5.2.	Ids versus V_{ds} for V_{gs}=1.6v, 2.2v, 2.8v, and 3.0v	66
Figure 5.3.	Ids versus V_{gs} for V_{ds}=0.1v	66
Figure 5.4.	Ids versus V_{ds} for different V_{gs} values	67
Figure 5.5.	I_{ds} versus V_{ds} for different V_{gs} values	67
Figure 5.6.	I_{ds} versus V_{ds} for different V_{gs} values	68
Figure 5.7.	I_{ds} versus V_{ds} for different V_{gs} values	68
Figure 5.8.	I_{ds} versus V_{ds} for V_{gs} = 3v	69
Figure 5.9.	I_{ds} versus V_{ds} for V_{gs} = 5v	70
Figure 5.10.	I_{ds} versus V_{ds} for V_{gs} = 2v	70

Table 1.1.	Parameters for MOS Level 1, 2 & 3	13
Table 1.2.	Parameters for MOS Level 1, 2 & 3 continued	14
Table 3.1.	A few default parameters built into the FITDRF optimizer	43
Table 5.1.	Number of local & global minima in the test functions	64
Table 5.2.	A comparison results of test functions	64
Table 5.3.	A comparison of W/L=200/0.35 BSIM3v3 results	69

INTRODUCTION

As the semiconductor industry has progressed over the past thirty years, integrated circuit densities have increased tremendously. In fact, the number of transistors on a chip has been doubling every 18 to 24 months Figure 1.1(a), an observation that has come to be known as “Moore’s Law” after Gordon Moore, the man who first noted it [1]. Which states that the number of transistors on a chip doubled every 18 months. Intel’s first processor, the 4004, contained 2,300 transistors whereas today’s complex microprocessors incorporate close to 40 million transistors and are up to more than a quarter million times faster Figure1.1(b). Unfortunately, such aggressive decrease in device size and increase in circuit density have not been possible without bringing along a variety of limitations [2].



(a) (b)

Figure 1.1 : Moor's law ,(a) the number of transistors on a chip doubled every 18 to 24 months[1].
(b) Intel's microprocessors developments[2]

With CAD tools like SPICE (Simulation Program with Integrated Circuit Emphasis) [3] it is possible to simulate circuits containing thousands of devices. The success of these simulations is determined by the ability of the device models to describe the device behavior properly. Therefore, accurate device models and accurate means of characterizing real devices are of great importance. The device models usually consist of a set of model

equations that are either empirical or derived from device physics, or a combination of both. No matter how the model equations are derived they contain a number of model parameters. For transistors, the model equations can be quite complicated and the number of model parameters can be rather high. For instance, The BSIM3v3 MOSFET transistor models have about 70 model parameters, [4]. With the complicated transistor models used in the circuit simulators today, device characterization with respect to circuit models becomes a non-trivial task.

To obtain a complete device model suitable for circuit simulations, methods for obtaining "proper" values for the model parameters are crucial. These methods are often referred to as parameter extraction techniques or algorithms. The parameter extraction is to be thought of as an integral part of the device model. A fancy set of model equations without techniques for calculating the model parameters from measurement data is of no use when considering circuit simulation. The parameter extraction algorithm should consist of not only the proper manipulations of the model equations and calculation techniques but also instructions on how to dispose the measurement scheme to perform the best extraction using the least measurement effort.

As devices are scaled down into sub-micrometer regime, many approaches have proposed in the linear region to determine MOSFET parameters such as **threshold voltage**, **effective mobility**, **effective channel length**, **effective channel width**, and parasitic **series resistance**, but these methods works under such considerations which may not valid for another models.

The most important parameter to model the operation of a MOSFET is its threshold voltage, *Vth*. There are several definitions of threshold voltage and many methods have been developed over time to represent this parameter [5][6]

The channel length is a key parameter in CMOS technology used for circuit models, which is a broad description of three different channel lengths. The first one is the metallurgical gate length Lmet. The second is the mask channel length Lm, which denotes the physical length of the gate mask. And the last one is the electrical effective channel length Leff, which defines the length of a region near the Si-SiO2 interface in which the inversion free-carrier density is controlled by the gate voltage[7].

Almost twenty years ago, the most widely used methods to extract the total drain and source series resistance (RD + RS) are the reciprocal transconductance method and gate-voltage shift method. Several other methods are proposed recently, but as devices scaled

down, into 100nm regime Khaled et al[8] have demonstrated that Leff and Rds-gate-bias dependence extracted using modified shift-and-ratio method may not give accurate results.

The effective channel mobility is mainly extracted from the linear region Ids-Vgs characteristics, and since it is one of the most important process characterization parameters in CMOS technology. The accuracy of effective channel mobility is crucial to both compact modeling in circuit design and numerical simulation in device design. G. Niu et al [9] have proposed a method that can calculate the effective channel mobility without involving the effective channel length.

In the deep submicron MOSFETs regime and especially in the linear region W. Fikry et al[10] have proposed a method which takes into consideration the lateral field effect, and by which the threshold voltage, mobility degradation factor and low field mobility, saturation velocity, and series resistance can be extracted.

Finally, the channel width has an effect on threshold voltage as it is scaled. The width scaling effect is not as severe as the length scaling effects.

The aim of this work is to develop a parameter extraction tool that uses as input data just the I-V MOSFET characteristics and *Pmin* , *Pmax* for the parameters to be extracted.

In chapter 1 we will present a brief description of the different MOSFET compact models developed by different laboratories, and a review of the basic MOSFET operation.
Chapter 2, is devoted to model equations used in deep submicron MOSFET BSIM3v3.
Chapter 3, begins by describing several optimization methods that have been used for parameter extraction. The deficiencies presented by these existing methods during MOSFET developments, will be mentioned.
Chapter 4, deals with the FAST SIMULATED DIFFUSION method used for parameter extraction of deep submicron MOSFETs.
Chapter 5, is devoted to results and discussion. Then at the end an overall conclusion of this thesis, with a several pointers for future work.

CHAPTER 1
DEVELOPMENTS OF THE COMPACT MODELS

1.1 Introduction

Ever since the invention of transistors in 1947, nearly all kinds of electronic appliances have found their use as logic gates, switches or amplifiers. The first transistor was about half an inch high, mammoth by today's standards. As time passed since its invention, the need for downsizing the transistor arose due to its increased application in complex circuits. The interest of scientists together with competition between firms implementing transistors in their products spurred the downsizing of these devices at the same time balancing its functionality and manufacturing costs.

In the past 30 years, the IC semiconductor industry has been propelled from SSI (small scale integration) of less than 30 devices per chip in the 1970s, to MSI (medium-scale integration) of 30 to 10^3 devices per chip, to LSI (large-scale integration) of 10^3 to 10^5 devices per chip, to VLSI (very large-scale integration) of 10^5 to 10^7 devices per chip, and now to ULSI (ultra large-scale integration) of 10^7 to 10^9 devices per chip. As the device is being scaled down to the sub-micron region, the fabrication parameters, circuit design constraints and process information become critical to the device performance. Therefore there is a need to consider these factors before the transistor is fabricated. TCAD (Technology Computer-Aided Design) is an important tool in the simulation of wafer processing and device characterization. It enables users to obtain at little expense, realistic device structures, process information such as doping profiles and I-V curves through 2-D device simulation before actual transistor fabrication. With the help of TCAD, optimum device performance can be obtained from simulated process variations.

In this project, TCAD was used for parameter extraction, which is a critical stage(step) required for circuit design and simulation of deep submicron MOSFET technology. However, with the continued size reduction and changes in device structures, Several methods have been proposed for parameter extraction of a MOS transistor, but none have proved to be genuinely accurate. Each one has its own advantages and its drawbacks.

1.2. History of MOS modeling

FET modeling have evolved considerably since the past 30 years. The earlier models had very simple and basic equations for C-V and I-V characteristics. They also had very few parameters to describe the equations. These parameter values either represented particular and quantifiable physical values, or had a strong physical meaning. The values that had physical meaning were obtained directly from process information and those that could not be obtained from process information were obtained from electrical data and parameter extraction. Hence a basic SPICE FET model consists of a tabular set of model parameters which describes the technology and equations describing the characteristics of the device. These parameters are then plugged into the device equations and are used to solve the set of equations that describe the device characteristics during circuit simulation.

As the technology has evolved, the models that describe FET's have become more complex. The equations have become more involved to include various effects brought about due to shorter channels and higher field strengths. This has also led to the use of more parameters, which has led to increasing the workload in terms of extracting the model parameters. This has shifted the focus, somewhat, from analytic derivation of model equations to new and innovative parameter extraction techniques. In an ideal situation, a model would have a set of physical parameters with easily measurable and quantifiable values. *Physical* parameters, such as gate-oxide thickness have a direct and quantifiable meaning whereas *Electrical* parameters have to be derived with parameter extraction. As newer models have been developed, the number of electrical parameters has increased considerably.

The first generation models, Level 1, Level 2 and Level 3, represent the early efforts in modeling of FET's and they were all described by simple, physically based parameters. Second generation models, BSIM1, BSIM2, HSPICE level28, introduced a large number of empirical electrical parameters which required a large number of parameter extraction effort. The improvement of these models over their predecessors was that the derivatives for the current equations were continuous which is essential for analog circuit design as it provides greater stability. Third generation models, the latest version of BSIM3 and BSIM4 contain only one current and charge equation that describe the working of the device across all regions of operation of the device.

The development of BSIM compact models and others with respect to the year of their introduction is shown in Figure 1.2 , the principal third generation models are BSIM3v3,

BSIM4v2, MM9, EKV2.6, MM11. All these models are dedicated to the submicronic, and deep submicronic transistors. A brief description of these models is as follows:

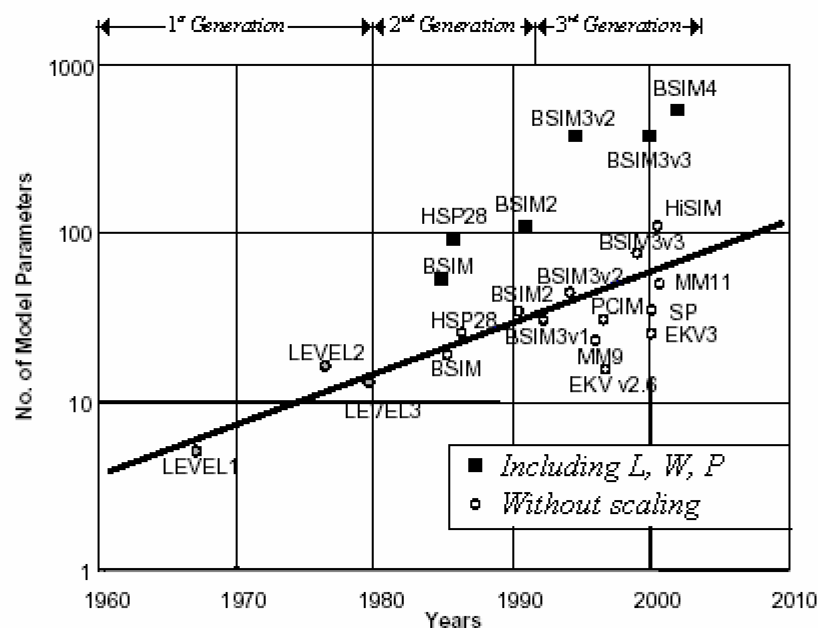


Figure 1.2: The three generations of compact models. Shows the number of model parameters vs the year's model introduction[11].

1.2.a. First generation

a.1 Level 1, 2, 3[12][13]

The parameter list for the level 1, 2 and 3 MOS models is shown in the Table1.1 and Table 1.2. A generalized equivalent circuit is shown in Figure 1.3 “Shichman-Hodges”, MOS1 (The MOS1 model).

This model was the first SPICE MOSFET model and was developed in 1968. It is an elementary model and has a limited scaling capability. It assumes simplifications such as gradual channel approximation and the square law for the saturated drain current. The only small geometry effect is the inclusion of a simple *lambda* model for channel length modulation, which leads to a finite value of output conductance. No subthreshold conduction model is included. It is applicable to fairly large devices with gate lengths greater than 4µm. Its main attribute is that only a few parameters need be specified and so it is good for preliminary analyses.

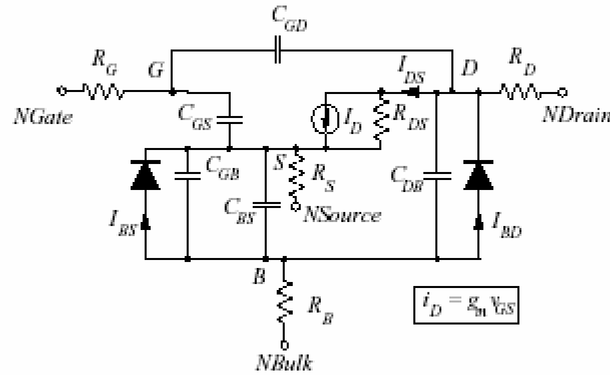


Figure 1.3: Schematic of LEVEL 1, 2 and 3 MOSFET models. VGS, VDS, VGD, VGB, VDB and VBS are voltages between the internal gate, drain, bulk and source terminals designated G, D, B and S respectively.

Table 1.1: Parameters for MOS Level 1, 2 & 3

Parameter	Description	Units
AF	flicker noise exponent	-
CBD	zero-bias B-D junction capacitor	F
CBS	zero-bias B-S junction capacitor	F
CBG0	gate-bulk o/v cap/m of channel length	F/m
CGD0	gate-drain o/v cap/m of channel width	F/m
CGS0	gate-source o/v c/m of channel width	F/m
CJ	zero-bias bulk junction bottom cap/sq.m of junction area	F/m
CJSW	zero-bias bulk junction sidewall cap/m of junction perimeter	F/m
DELTA	width effect on threshold voltage (LEVEL=2)	
ETA	static feedback(LEVEL=3 only)	-
FC	coefficient for forward –bias depletion capacitance formula	-
GAMMA	bulk threshold parameter	$V^{½}$
IS	bulk junction saturation current	A
JS	bulk junction saturation current/sq.m of junction area	A/m^2
KAPPA	saturation field factor (LEVEL=3 only)	-
KF	flicker noise coefficient	-

Table 1.2 : Parameters for MOS Level 1, 2 & 3 Continued

Parameter	Description	Units
KP	transconductance parameter	A/V^2
LAMBDA	channel length modulation (LEVEL=1, 2 only)	$1/V^2$
LD	lateral diffusion	m
LEVEL	model index	-
MJ	bulk junction bottom grading coefficient	-
MJSW	bulk junction sidewall grading coefficient	-
NSUB	substrate doping	cm^{-3}
NSS	surface state density	cm^{-2}
NFS	fast surface state density	cm^{-2}
NEFF	total channel charge (fixed and mobile) coefficient. (LEVEL= 2 only)	-
PB	bulk junction potential	V
PHI	surface inversion potential	V
RD	drain ohmic resistance	Ω
RS	source ohmic resistance	Ω
RSH	drain and source diffusion sheet resistance	Ω/square
THETA	mobility modulation (LEVEL= 3 only)	1/V
TOX	oxide thickness (not used for Level 1)	m
TPG	type of gate material	-
UCRIT	critical field for mobility degradation (Level 2 only)	V/cm
UEXP	critical field exponent in mobility degradation (Level =2 only)	-
U0	surface mobility (U-oh)	$Cm^2/V\text{-}s$
UTRA	transverse field coefficient (mobility) (Level =1 and 3 only)	-
VMAX	maximum drift velocity of carriers	m/s
VTO	zero-bias threshold voltage (VT-oh)	V
XJ	metallurgical junction depth	m

a.1.1 Drain Current Model (See table 1.2)

There are two basic regions of operation for Level 1:

linear region: $V_{GS} > V_T$ and $V_{DS} < V_{GS} - V_T$

saturation region: $V_{GS} > V_T$ and $V_{DS} > V_{GS} - V_T$

where V_{GS} is the applied signal at the gate terminal with respect to the source,

V_{DS} is the bias at the drain terminal with respect to the source,

and V_T is the threshold voltage, below which the transistor is cut-off and the drain current is zero.

The current in the linear region is given by:

$$I_{DS,Lin}=\frac{\mu W_{EFF} C_{OX}}{L_{EFF}}\left[(V_{GS}-V_T)V_{DS}-\frac{V^2{}_{DS}}{2}\right](1+\lambda V_{DS}) \tag{1.1}$$

and in the saturation region:

$$I_{DS,Sat}=\frac{\mu W_{EFF} C_{OX}}{2L_{EFF}}(V_{GS}-V_T)^2(1+\lambda V_{DS}) \tag{1.2}$$

μ is the mobility coefficient, W_{EFF} and L_{EFF} are the effective channel width and length respectively, C_{OX} is the gate-oxide capacitance

a.1.2 Charge Model

Charge neutrality dictates that $Q_{GATE} + Q_{INV} + Q_{DEPL} = 0$.

This model makes the assumption that the depletion charge does not vary along the channel. It is given by

$$Q_{DEPL}=W_{EFF}L_{EFF}C_{OX}\gamma\sqrt{2\phi_f - V_{BS}} \tag{1.3}$$

The inversion charge Q_{INV} is given by

$$Q_{INV}(y)=-C_{OX}(V_{GS}-V_T-V(y)) \tag{1.4}$$

The charge at the gate terminal, Q_{GATE} can be found using equations (1.3) and (1.4).

For the Linear Region:

$$Q_{GATE} = \frac{2}{3} W_{EFF} L_{EFF} C_{OX} \left[\frac{(V_{GD} - V_T)^3 - (V_{GS} - V_T)^3}{(V_{GD} - V_T)^2 - (V_{GS} - V_T)^2} \right] - Q_{DEPL} \quad (1.5)$$

where Q_{DEPL} is given by equation (1.3)

For the saturation region:

$$Q_{GATE} = \left[\frac{2}{3} W_{EFF} L_{EFF} C_{OX} (V_{GS} - V_T) \right] - Q_{DEPL} \quad (1.6)$$

where Q_{DEPL} is given by Equation (1.3)

The utility of this model is now purely instructive. The derivations demonstrate a basic approach to analytic models of a MOS transistor. Also the process of parameter extraction is uncomplicated and mathematical.

a.2 MOS2

This is an analytical model which uses a combination of processing parameters and geometry. The major development over the LEVEL 1 model is improved treatment of the capacitances due to the channel charge. The model dates from 1980 and is applicable for channel lengths of 2 μm and higher. The LEVEL 2 model has convergence problems and is slower and less accurate than the LEVEL 3 model. The Level 2 model was the first attempt to describe the behavior of small geometry MOS transistors. However, the LEVEL 2 model is more mathematically complex. The charge model takes into account only the overlap of the source and drain depletion regions and ignores charge sharing between the source and drain. Also, there are many choices for the saturation voltage model. This tends to give rise to convergence problems with a discontinuous first derivative. The Level 3 is considered more robust and usable.

a.3 MOS3

This is a semi-empirical model developed in 1980. It is also used for gate lengths of 2μm and more. The parameters of this model are determined by experimental characterization and so it is more accurate than the LEVEL 1 and 2 models that use the

more indirect process parameters. This model, on account of its simplicity and operational reliability made it a popular choice for digital design.

However, there is an abrupt change from the linear to saturation regions which leads to a first derivative discontinuity of current because of which the model provides a poor fit to data. It also does not provide an accurate subthreshold model which is essential for analog applications. For more details refer to [12] [13].

The parameters used in the Level 1, 2 and 3 models are given in table 1.1. It is assumed that the model parameters were determined or measured at the nominal temperature.

The LEVEL 1, 2 and 3 models have much in common. These models evaluate the junction depletion capacitances and parasitic resistances of a transistor in the same way. They differ in the procedure used to evaluate the overlap capacitances (*CGD*, *CGS* and *CGB*) and that used to determine the current-voltage characteristics of the active region of a transistor. The overlap capacitances model charge storage as nonlinear thin-oxide capacitance distributed among the gate, source drain and bulk regions. These capacitances are important in describing the operation of MOSFETs. The LEVEL 1, 2 and 3 models are intimately intertwined as combinations of parameters can result in using equations from more than one model. The LEVEL parameter resolves conflicts when there is more than one way to calculate the transistor characteristics with the parameters specified by the user. The MOSFET LEVEL 1, 2 and 3 parameters fall into three categories: absolute device parameters, scalable and process parameters and geometric parameters. In most cases the absolute device parameters can be derived from the scalable and process parameters and the geometry parameters. However, if specified, the values of the device parameters are used.

1.2.b Second generation

The second generation of MOSFET SPICE models includes BSIM and its two descendants: HSPICE Level 28, and BSIM2.

BSIM

The Berkeley Short-Channel IGFET Model is an advanced empirical model and is referred to as level 4. It relies on polynomial equations to enable handling of various effects. Although it performs better than the earlier MOS models, it shows a degradation in

performance in sub-micron FETs because the polynomial equations can behave poorly and can cause negative output conductance leading to convergence problems [14].

LEVEL = 28 HSPICE

This is a proprietary model developed by Meta-Software and is similar to BSIM. It is often used in analog circuit design.

BSIM2

After many modifications, it became an extension to BSIM to be used in analog circuit design. It scored over BSIM in terms of model accuracy and convergence during runtime, but under certain conditions there are discontinuities in the I-V and the C-V characteristics, which can cause numerical errors during simulation.

1.2.c Third generation

Models BSIM3/4[15] and MM9 represent the emergence of the third generation of MOSFET SPICE models.

BSIM3

This model eliminates the discontinuity in the derivatives of the I-V and C-V characteristics by using a single equation to describe device characteristics such as current and charge across all regions of operation.. This model has found to have produced accurate results at 0.18 µm technologies. More details, new releases and code can be found at [16].

BSIM3 itself evolved through three versions:

-BSIM3v1 forms the original base of the model, but suffers of severe mathematical problems.

-BIM3v2 introduces strong corrections to solve the mathematical difficulties of BSIM3v1, and new parameters are added.

-BSIM3v3 significantly changes the form of the model to guarantee smoothed and continuous equations; a number of empirical expressions significant is also introduced, through many additional parameters.

Today the series of model BSIM is with its version 4v2 [17], and even 4v3 recently.

Model 9

This model was developed at Philips Laboratories and is the primary non-Berkeley model that is available for public use. This model is accurate in submicron technologies and shows good stability during circuit simulation. More can be found out about this model at [18].

EKV

This model was first proposed by Christian Enz, Franois Krummenacher and Eric Vittoz. It employs bulk-referencing which is a different approach compared to other models which use source-referencing. This fundamental change eliminates symmetry problems which is unavoidable in other models. The manuals and code can be found online at [19].

1.3. Review of MOSFET operation

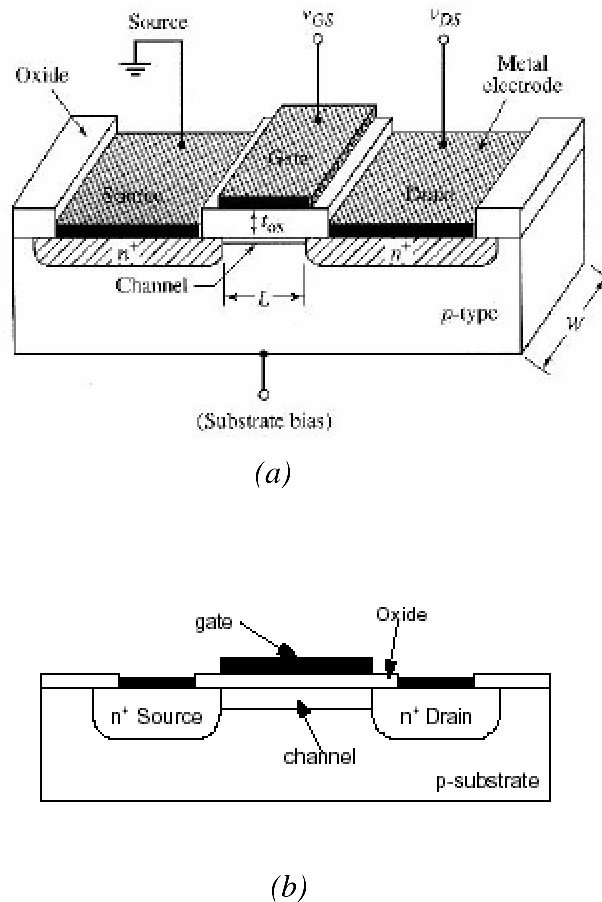
The origin of the field effect transistor (FET) goes back as far as 1926. The basic idea was that it might be possible to make a 'voltage-controlled resistor' by varying the resistance between two contacts on the surface of a semiconductor (the source and the drain contacts) with the aid of a third electrode called the gate.

The first metal-oxide-semiconductor field-effect transistor (MOSFET) was fabricated in the year 1960, which is only a year after the beginning of the integrated circuit in 1959. With the progress of technology in these days, since early 1980s, the MOSFET has become the basic building block of very large-scale integrated (VLSI) circuit, and has also been the most widely used device. This is possibly because it has a simpler structure, costs much lesser to fabricate, and consumes lesser power compared to other devices, such as bipolar and junction field-effect transistors [20]. In the following sections a brief description of MOSFET structure and operation for long channel will be studied. More details for MOSFET's model equations BSIM 3v2 will be discussed in chapter2.

1.3.1. MOSFET Structure

The n-type Metal-Oxide-Semiconductor Field-Effect-Transistor (nMOSFET) consists of a source and a drain, two highly conducting n-type semiconductor regions, which are isolated from the p-type substrate by reversed-biased p-n diodes. A metal or poly-crystalline gate covers the region between source and drain. The gate is separated from the

semiconductor by the gate oxide. The basic structure of an n-type MOSFET is shown in Figure 1.4.



(a)

(b)

Figure 1.4: A basic n-MOSFET structure. (a) cross section structure Showing the length L, and width W[20].(b) cross section of the structure.

Basically, the MOSFET consists of 4 terminals, namely: the ***source**, **drain***, ***gate***, and ***substrate*** terminals. These are also the electrical designations of the device, which determines the electrical characteristics. By convention, the drain is the terminal where the output current is measured, while the source is the common terminal to input and output in a grounded source configuration [21]. The voltage input is on the gate and the bulk is usually connected to a reference (either the source or the ground potential in the circuit).

1.3.2. Principle of operation

A top view of the same MOSFET is shown in Figure1.5, where the gate length, *L*, and gate width, *W*, are identified. Note that the gate length does not equal the physical dimension of the gate, but rather the distance between the source and drain regions underneath the gate. The overlap between the gate and the source/drain region is required to ensure that the inversion layer forms a continuous conducting path between the source and drain region. Typically this overlap is made as small as possible in order to minimize its parasitic capacitance.

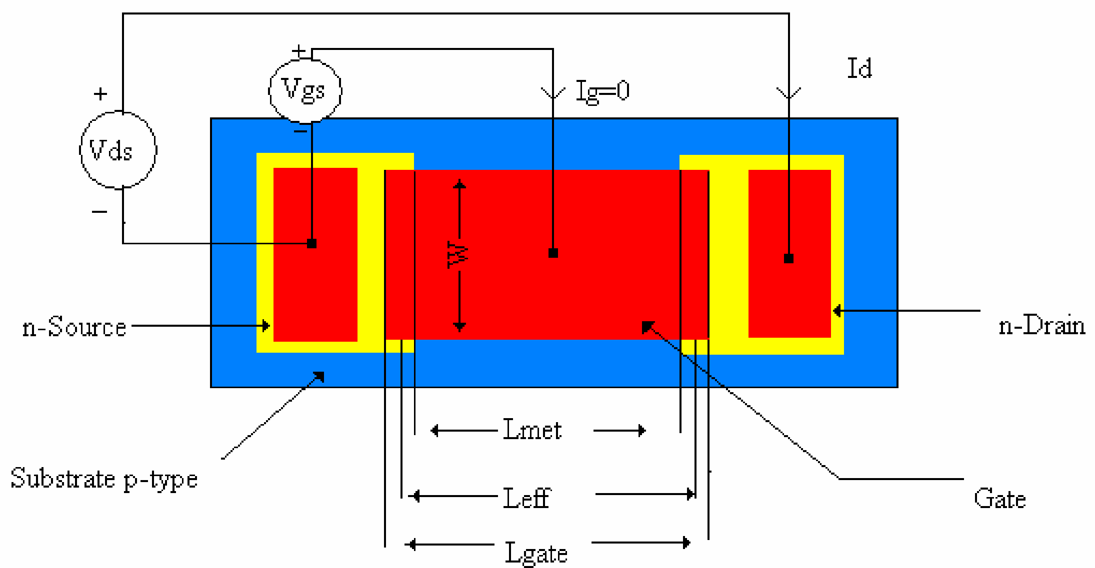


Figure 1.5: Top view of an n-type Metal-Oxide–Semiconductor–Field–Effect–Transistor (MOSFET).Where Lmet is the metallurgical channel length, Leff is the electrical effective channel length, and Lgate is the physical length of the gate mask.

The flow of electrons from the source to the drain is controlled by the voltage applied to the gate. A positive voltage applied to the gate attracts electrons to the interface between the gate dielectric and the semiconductor. These electrons form a conducting channel between the source and the drain called the inversion layer. No gate current is required to maintain the inversion layer at the interface since the gate oxide blocks any carrier flow. The net result is that the current between drain and source is controlled by the voltage, which is applied to the gate. The typical current versus voltage (*I*-*V*) characteristics of a MOSFET are shown in Figure1.6.

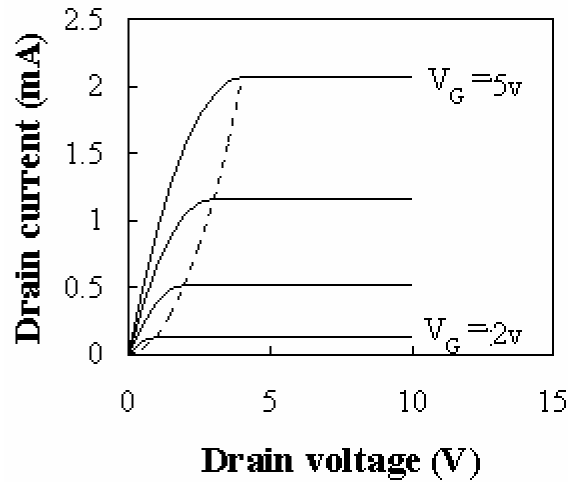


Figure 1.6: I-V characteristics of an n-type MOSFET

1.3.3. Formation of the Channel Region in the MOSFET

As mentioned earlier, the voltage is applied to the gate that controls the state of the silicon surface underneath. Negative gate voltages attract the holes from the p-type silicon to the surface (accumulation), while positive voltage larger than the threshold voltage creates a layer of electrons at the surface (inversion) [21].

The existence of this layer of electron corresponds to the ON state of the transistor, as the electrons channel virtually short circuits the source and the drain regions together. This creates a channel region Figure 1.7, adjacent to the oxide-semiconductor interface, and allows current to flow. When the gate voltage is below the threshold voltage, the electron layer (or channel region) disappears from the surface, and the source and drain regions are isolated by the p-substrate. This corresponds to the OFF state of the transistor, as no current flows from the drain to the source without a conducting n-channel between them.

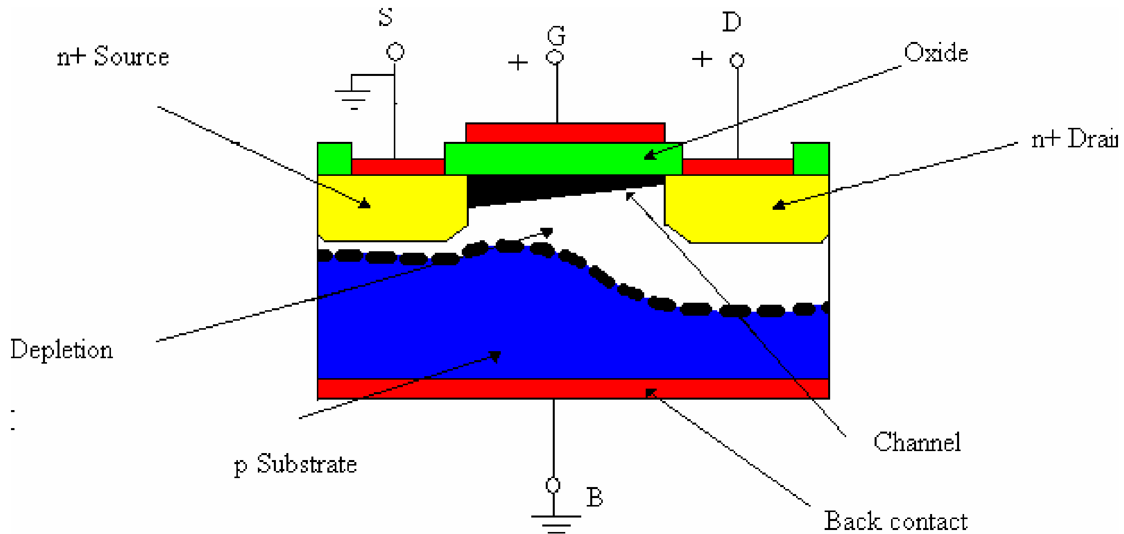


Figure 1.7: Schematic illustration of the induced n-channel [20].

1.3.4. MOSFET I-V Characteristic

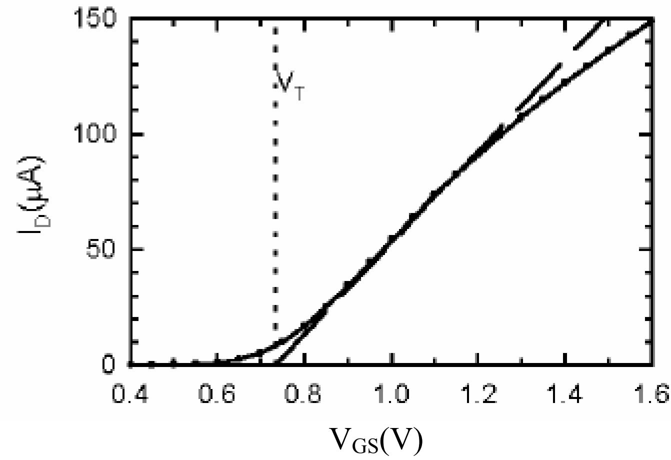
Under normal operating conditions, the source and drain voltages are always such that the source and drain-to-substrate pn junctions are reverse bias. The simplest bias arrangement that can be used to illustrate the operation of a MOSFET is when both the source and the bulk are at ground potential, i.e. $V_b = V_s = V_{sb} = 0$. Even at $V_{gs}=V_{ds}=0$, a depletion region is formed around n+ source and drain regions (see dashed lines of Figure 1.7 due to the n+p junction formed with the p-type substrate.

(a) Linear Region

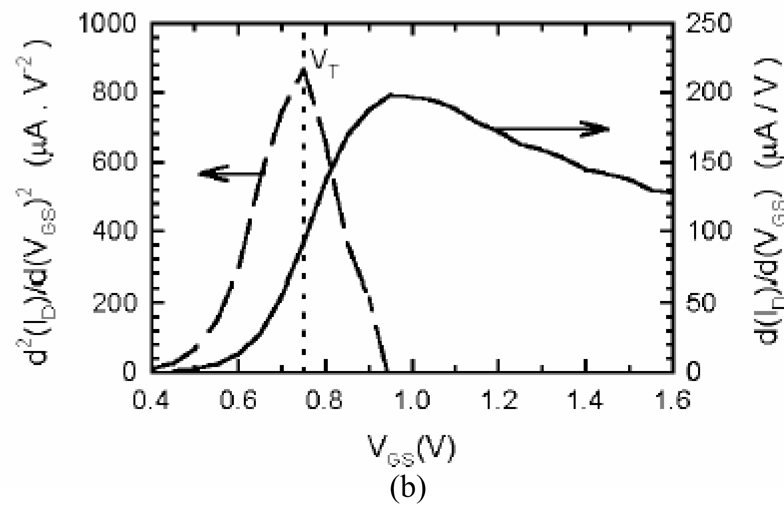
When V_{ds} is small, this is a region in which I_{ds} increases linearly with V_{ds} for a given V_{gs} ($> V_T$). I_{ds} in the linear region is given by:

$$I_{ds} = \mu C_{ox}\left(\frac{W}{L}\right)(V_{gs} - V_T - 0.5V_{ds})V_{ds} \tag{1.7}$$

Where μ is mobility of the carriers (electrons for nMOSFET) in the channel region, C_{ox} is the gate oxide capacitance per unit area, W/L is device width to length ratio, and V_T is the threshold voltage. The threshold voltage, V_T, can be determined by plotting I_{ds} versus V_g at low drain voltages, as shown in Figure 1.8. The extrapolated intercept of the linear portion of the $I_{ds}(V_g)$ curve with the V_g-axis gives the approximate value of V_T. Notice that the $I_{ds}(V_g)$ curve is not linear near the threshold voltage.



(a)



(b)

Figure 1.8: Illustrates the extraction of VT=Vth.(a)the linear extrapolation at the point of maximum slope method.(b) the second derivative method [22].

(b) Saturation Region

In this region, I_{ds} no longer increases as V_{ds} increases, therefore it saturates. I_{ds} in the saturation region is given by:

$$I_{ds} = \frac{1}{2}\mu C_{ox}\left(\frac{W}{L}\right)(V_{gs} - V_T)^2 \tag{1.8}$$

showing that I_{ds} does not depend on V_{ds}. This is evident from Figure 1.9. However, it should be pointed out that this complete current saturation occurs only for MOSFETs with long channel lengths. As L decreases the saturation behavior degrades rapidly, causing an increase in I_{ds} when V_{ds} is increased.

(c) Breakdown Region

With further increase of V_{ds} beyond saturation, the transistor enters a region in which I_{ds} suddenly increases until breakdown of the drain-to-substrate pn junction occurs, and is caused by high electric field at the drain end. Dashed line (b) of Figure 1.9 shows the boundary between the saturation and breakdown region.

(d) Cut-Off Region

This is the region in which $V_{gs} < V_T$ so that no channel exist between the source and the drain, resulting in $I_{ds} = 0$.

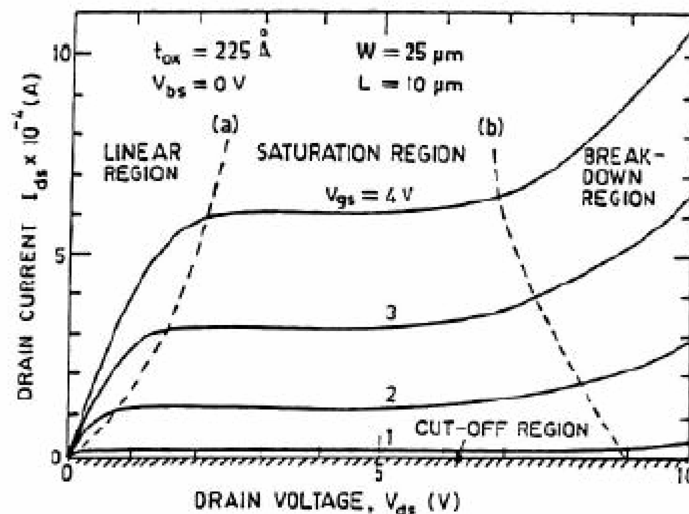


Figure 1.9: Ids vs Vds characteristics for different values of Vgs showing different regions of mosfet operation (a) linear (b) saturation (c) cut-off and (d) breakdown regions[20].

CHAPTER 2
MODEL EQUATIONS

2.1. Introduction

To cope with the continuous evolution of VLSI technology, many short-channel MOSFET I-V models for circuit simulation have been developed. Most of these models, however, are either not adequately covering the small-size effects that become significant at the deep-submicron level, or are highly empirical. Empirical models can have the advantages of easy formulation, because of the use of a large number of empirical parameters. They may provide good accuracy in fitting a single device from a wide range of technologies. However, their drawbacks are many: generating size-independent process files is a very difficult task. Extrapolating a process file for a present technology to a future one is virtually impossible, and, perhaps most important, circuit designers may lose the intuition which is vital in achieving high performance analog and digital circuits. BSIM3 is developed to address these drawbacks.

BSIM3 is a physical model with extensive built-in dependencies of important dimensional and processing parameters such as channel length (L), width (W), gate oxide thickness (T_{ox}), junction depth, (X_j) , substrate doping concentration, and LDD structures, and so forth. It allows users to accurately model, upon parameter extraction on existing technology, or predict, based on the default or extracted technologies, MOSFET behavior over a wide range of existing and future technologies. Using a coherent pseudo 2D formulation, such major short-channel effects and high field effects as threshold voltage reduction, non-uniform doping effect, mobility reduction due to vertical field [4], carrier velocity saturation, channel-length modulation (CLM), drain induced barrier lowering (DIBL), substrate current-induced body effect (SCBE), sub-threshold conduction, parasitic resistance effect, and LDD effect are properly included. Meticulous care has been taken to retain the physical functional forms while improving model accuracy and computational efficiency. The model is compact, and time-consuming functions are excluded. The ease of parameter extraction was also a major consideration. The number of parameters is small (~ 28) and every parameter has a physical meaning; the effects of parameters on output characteristics are very predictive. This feature of BSIM3 makes statistical study of the device fabrication process possible. Drain current and its first order derivative in all

operation regions are continuous, which removes all kinks and glitches at the boundaries between the regions. BSIM3 has been implemented into SPICE3 and the divergence problem has been greatly improved. In the following sections BSIM3v2/3 model equations will be investigated [23].

2.2. Effective Channel Width and Length (see appendix A)

The calculation of W_{eff} and L_{eff} is more complex (compared to the previous versions of BSIM), with the inclusion of dependencies for different values of W_{drawn} and L_{drawn}. W_{eff} is allowed to vary with gate and substrate bias. This improves the ability of the model to fit a variety of *W/L* ratios with a single set of parameters. These parameters are given by:

$$L_{eff}=L_{drawn}-2dL \qquad (2.1)$$

$$W_{eff}=W_{drawn}-2dW \qquad (2.2)$$

$$W'_{eff}=W'_{drawn}-2dW' \qquad (2.3)$$

Where *dL, dW* and *dW'* are modeled by the following:

$$dL=L_{int}+\frac{L_l}{L_{drawn}^{L\ln}}+\frac{L_w}{W_{drawn}^{Lwn}}+\frac{L_{wl}}{L_{drawn}^{L\ln}W_{drawn}^{Lwn}} \qquad (2.4)$$

$$dW = dW' + dW_{gsteff} + dW_b\left(\sqrt{\phi_s - V_{bseff}} - \sqrt{\phi_s}\right) \qquad (2.5)$$

$$dW' = W_{\text{int}} + \frac{W_l}{L_{drawn}^{W\ln}}+\frac{W_w}{W_{drawn}^{Wwn}}+\frac{W_{wl}}{L_{drawn}^{W\ln}W_{drawn}^{Wwn}} \qquad (2.6)$$

The length and width parameters in the above formulas have already been defined in appendix A. The parameters dW_g and dW_b account for the influence of gate and substrate bias on the effective channel width. *The* W'_{eff} represents the zero biased effective width.

2.3. Effective Voltages

The effective gate voltage is defined as

$$V_{gsteff}=\frac{2nv_t\ln\left[1+\exp\left(\frac{V_{gs}-V_{th}}{2nv_t}\right)\right]}{1+2nC_{ox}\sqrt{\frac{2\phi_s}{q\varepsilon_{si}N_{ch}}}\exp\left(-\frac{V_{gs}-V_{th}-2V_{off}}{2nv_t}\right)} \qquad (2.7)$$

Where n=1+N$_{factor}\dfrac{C_d}{C_{ox}}+\dfrac{C_{it}}{C_{ox}}+\dfrac{C_{dsc_term}}{C_{ox}}$ (2.8)

$$C_{dsc_term}=\left(C_{dsc}+C_{dscd}V_{ds}+C_{dscb}V_{bseff}\right)\left[\exp\left(-D_{VT1}\frac{L_{eff}}{2l_t}\right)+2\exp\left(-D_{VT1}\frac{L_{eff}}{l_t}\right)\right] \quad (2.9)$$

and

$$\text{Cd}=\frac{\varepsilon_{si}}{X_{dep}} \quad (2.10)$$

The effective bulk voltage is defined as

$$V_{bseff}=V_{bc}+0.5\left(V_{bs}-V_{bc}-\delta_1+\sqrt{(Vbs-Vbc-\delta_1)^2-4\delta_1V_{bc}}\right) \quad (\delta_1=0.001) \quad (2.11)$$

Where $$V_{bc}=0.9\left(\phi_s-\frac{K_1{}^2}{4K_2{}^2}\right) \quad (2.12)$$

The effective drain voltage is defined as

$$V_{dseff}=V_{dsat}-\frac{1}{2}\left(V_{dsat}-V_{ds}-\delta+\sqrt{(V_{dsat}-V_{ds}-\delta)^2+4\delta V_{dsat}}\right) \quad (2.14)$$

2.4. Threshold Voltage Modeling

The threshold voltage at nominal temperature in BSIM3v3 is modeled according to the equation below.

$$V_{th}=V_{th0}+K_1\left(\sqrt{\phi_s-V_{bseff}}-\sqrt{\phi_s}\right)-K_2V_{bseff}-\Delta V_{th-SC}+\Delta V_{th-RSC}$$
$$-\Delta V_{th_DIBL}+\Delta V_{th-NC}-\Delta V_{th_WL} \quad (2.15)$$

Where :

$$\Delta V_{th-SC}=D_{VT0}\left[\exp\left(-D_{VT1}\frac{L_{eff}}{2l_t}\right)+2\exp\left(-D_{VT1}\frac{L_{eff}}{l_t}\right)\right](V_{bi}-\phi_s) \quad (2.16)$$

$$\Delta V_{th_RSVC}=K_1\left(\sqrt{1+\frac{Nlx}{L_{eff}}}-1\right)\sqrt{\phi_s} \quad (2.17)$$

$$\Delta V_{th_DIBL}=\left[\exp\left(-D_{sub}\frac{L_{eff}}{2l_{t0}}\right)+2\exp\left(D_{sub}\frac{L_{eff}}{l_{t0}}\right)\right](E_{tab}+E_{tab}V_{bseff})V_{ds} \quad (2.18)$$

$$\Delta V_{th_NC} = \left(K_3 + K_{3b}V_{bseff}\right)\frac{T_{ox}}{W'_{eff} + W_0}\phi_s \tag{2.19}$$

$$\Delta V_{th_WL} = D_{VT0w}\left[\exp\left(-D_{VT1w}\frac{W'_{eff}L_{eff}}{2l_{tw}}\right) + 2\exp\left(-D_{VT1w}\frac{W'_{eff}L_{eff}}{l_{tw}}\right)\right]\left(V_{bi} - \phi_s\right) \tag{2.20}$$

$$Vbi = \frac{K_B T}{q}\ln\left(\frac{N_{ch}N_{DS}}{n_i^2}\right) \tag{2.21}$$

$$l_t = \sqrt{\frac{\varepsilon_{si}X_{dep}}{C_{ox}}\left(1 + D_{vt2}V_{bseff}\right)} \tag{2.22}$$

$$l_{tw} = \sqrt{\frac{\varepsilon_{si}X_{dep}}{C_{ox}}\left(1 + D_{vt2w}V_{bseff}\right)} \tag{2.23}$$

$$l_{t0} = \sqrt{\frac{\varepsilon_{si}X_{dep0}}{C_{ox}}} \tag{2.24}$$

$$X_{dep} = \sqrt{\frac{2\varepsilon_{si}\left(\phi_s - V_{bseff}\right)}{qN_{ch}}} \tag{2.25}$$

$$X_{dep0} = \sqrt{\frac{2\varepsilon_{si}\phi_s}{qN_{ch}}} \tag{2.26}$$

2.5. Mobility

The expression for effective mobility is based on V_{gsteff} and has different options depending on the mobility selector mobmod.

For mobmod=1:

$$\mu_{eff} = \frac{\mu 0}{1 + \left(U_a + U_c V_{bseff}\right)\left(\frac{V_{gsteff} + 2V_{th}}{T_{ox}}\right) + U_b\left(\frac{V_{gsteff} + 2V_{th}}{T_{ox}}\right)^2} \tag{2.27}$$

For mobmod=2:

$$\mu_{eff} = \frac{\mu 0}{1 + \left(U_a + U_c V_{bseff}\right)\left(\frac{V_{gsteff}}{T_{ox}}\right) + U_b\left(\frac{V_{gsteff}}{T_{ox}}\right)^2} \tag{2.28}$$

For mobmod=3 :

$$\mu_{eff} = \frac{\mu 0}{1 + \left[U_a\left(\frac{V_{gsteff} + 2V_{th}}{T_{ox}}\right) + U_b\left(\frac{V_{gsteff} + 2V_{th}}{T_{ox}}\right)^2\right]\left(1 + U_c V_{bsteff}\right)} \tag{2.29}$$

2.6. Source/Drain Resistance

Source/Drain resistance, Rds, is modelled according to :

$$R_{ds} = \frac{R_{dsw}\left[1 + P_{rwg}V_{gsteff} + P_{rwb}\left(\sqrt{\phi_s - V_{bseff}} - \sqrt{\phi_s}\right)\right]}{\left(10^6 W'_{eff}\right)^{Wr}} \tag{2.30}$$

Here a V_{gs} dependence has been added through P_{rwg}, a V_{bs} dependence has been added through P_{rwb}, the width dependence has been changed to a power law (Wr) and the R_{ds0} (resistance offset parameter) of the previous version has been removed . Allowing the source drain resistance to be a function of both gate and bulk biases is expected to improve model accuracy. W'_{eff} is the zero biased effective width.

2.7. Bulk Charge

There is only one expression for modeling bulk charge:

$$A_{bulk} = \left(1 + \frac{K_1}{2\sqrt{\phi_s - V_{bseff}}} A_{bulkterm}\right)\frac{1}{1 + ketaV_{bseff}} \tag{2.31}$$

$$A_{bulkterm} = \frac{A_0 L_{eff}}{L_{eff} + 2\sqrt{X_j X_{dep}}}\left[1 - A_{gs}V_{gsteff}\left(\frac{L_{eff}}{L_{eff} + 2\sqrt{X_j X_{dep}}}\right)^2\right] + \frac{B_0}{W'_{eff} + B_1} \tag{2.32}$$

2.8. Saturation Field

The saturation field is modeled as:

$$E_{sat} = \frac{2V_{sat}}{\mu_{eff}} \tag{2.33}$$

2.9. Saturation Voltage

The saturation voltage depends on the value of Rds

R_{ds}=0 or $\lambda = 1$:

$$V_{ds} = \frac{E_{sat} L_{eff} \left(V_{gsteff} + 2v_t\right)}{A_{bulk} E_{sat} L_{eff} + \left(V_{gsteff} + 2v_t\right)} \tag{2.34}$$

R_{ds}>0 or $\lambda \neq 1$:

$$Vdsat = \frac{-b - \sqrt{b^2 - 4ac}}{2a} \tag{2.35}$$

$$a = A^2{}_{bulk} W_{eff} R_{ds} C_{ox} v_{sat} + \left(\frac{1}{\lambda} - 1\right) Abulk \tag{2.36}$$

$$b = -\left(\left(V_{gsteff} + 2v_t\right)\left(\frac{2}{\lambda} - 1\right) + A_{bulk} E_{sat} L_{eff} + 3A_{bulk}\left(V_{gsteff} + 2v_t\right) R_{ds} C_{ox} W_{eff} v_{sat}\right) \tag{2.35}$$

$$c = \left(V_{gsteff} + 2v_t\right) E_{sat} L_{eff} + 2\left(V_{gsteff} + 2v_t\right)^2 R_{ds} C_{ox} W_{eff} v_{sat} \tag{2.36}$$

$$\lambda = A_1 Vgsteff + A_2 \tag{2.37}$$

2.10. Drain Current Modeling

This model uses different expressions for drain current depending on the region of operation. For large gate voltages the device operates in strong inversion which can be subdivided into linear and saturation regions depending on the drain voltage. For small voltages the device operates in weak inversion. Finally, for a range of gate voltages near the threshold the device is considered to operate in a transition region and the current is calculated using interpolation formulas. The various equations used in the different operating regions are as follows:

2.10.1. Strong Inversion Current (V_{gs}-V_{th} > $V_{ghigh}+\theta_{DIBL}V_{ds}$)

-Linear Region (V_{ds} < V_{dsat}):

$$I_{ds} = \frac{I_{dslin0}}{1+R_{ds}I_{dslin0}/V_{ds}} \tag{2.38}$$

Where

$$I_{dslin0} = \mu_{eff}C_{ox}\frac{W_{eff}}{L_{eff}}\frac{1}{1+V_{ds}/(E_{sat}L_{eff})}\left(V_{gs}-V_{th}-A_{bulk}\frac{V_{ds}}{2}\right)V_{ds} \tag{2.39}$$

-Saturation Region (Vds > Vdsat):

$$I_{ds} = I_{dsat}\left(1+\frac{V_{ds}-V_{dsat}}{V_A}\right)\left(1+\frac{V_{ds}-V_{dsat}}{V_{ASCBE}}\right) \tag{2.40}$$

$$I_{dsat} = W_{eff}v_{sat}C_{ox}\left(V_{gs}-V_{th}-A_{bulk}V_{dsat}\right) \tag{2.41}$$

2.10.2. Weak Inversion Current (V_{gs}-V_{th} < ($V_{glow}-\theta_{DIBL}V_{ds}$)

There are various options for weak inversion current depending on the parameter subthmod. For subthmod=2:

$$I_{ds} = I_{s0}\left(1-\exp\left(-\frac{V_{ds}}{v_t}\right)\right)\exp\left(\frac{V_{gs}-V_{th}-V_{off}-\theta_{DIBL}(L)V_{ds}}{nv_t}\right) \tag{2.42}$$

$$vt = \frac{kT}{q} \tag{2.43}$$

$$I_{s0} = \mu_0\frac{W_{eff}}{L_{eff}}\sqrt{\frac{\varepsilon_{si}qN_{peak}}{2\phi_s}}v_t^2 \tag{2.44}$$

$$n = 1+Nfactor\frac{Cd}{Cox}+\frac{\theta_n(L)(C_{dsc}+C_{dsc1}V_{bs})}{Cox}+\frac{C_{it}}{C_{ox}} \tag{2.45}$$

$$C_d = \frac{\varepsilon_{si}}{X_{dep}} \tag{2.46}$$

$$\theta_n(L) = \exp\left(-\frac{D_{vt1}L_{eff}}{2l_t}\right)+2\exp\left(-\frac{D_{vt1}L_{eff}}{l_t}\right) \tag{2.47}$$

$$\theta_{DIBL}(L) = (eta0 + etaV_{bs})\left[\exp\left(-\frac{D_{sub}L_{eff}}{2l_0}\right) + 2\exp\left(-\frac{D_{sub}L_{eff}}{l_0}\right)\right] \tag{2.48}$$

2.10.3. Transition Region Modeling

The transition region modeling for gate voltages near threshold is defined as follows:

$$V_{glow} - \theta_{DIBL}V_{ds} \le V_{gs} - V_{th} \le V_{ghigh} + \theta_{DIBL}V_{ds} \tag{2.49}$$

Here, an interpolation formula is used to calculate the current :

$$I_{ds} = (1-t)^2 I_{dslow} + 2(1-t)tI_p + t^2 I_{dshigh} \qquad t \in (0,1) \tag{2.50}$$

Where:

I_{dslow} : is the current predicted from the subthreshold formula at the lower edge (with respect to Vgs) of the transition region.

I_{dshigh}: is the current predicted by the strong inversion formula at the higher edge (with respect to Vgs) of the transition region.

..I_p and t are quantities used by the interpolation scheme, the parameter t varies from 0 to 1 as V_{gs} varies from the lower bound to the upper bound of the transition region.

2.11. Early voltage modeling

This model uses the concept of an Early voltage for output conductance modeling. The equations for the Early voltage terms are given below:

$$VA = V_{Asat} + \left(1 + \frac{P_{vag}V_{gsteff}}{E_{satLeff}}\right)\left(\frac{1}{V_{ACLM}} + \frac{1}{V_{ADIBLC}}\right)^{-1} \tag{2.51}$$

$$V_{Asat} = \frac{E_{sat}L_{eff} + V_{dsat} + 2R_{ds}v_{sat}C_{ox}W_{eff}V_{gsteff}\left[1 - \frac{A_{bulk}V_{dsat}}{2(V_{gsteff} + 2v_t)}\right]}{\frac{2}{\lambda} - 1 + R_{ds}v_{sat}C_{ox}W_{eff}A_{bulk}} \tag{2.52}$$

$$V_{ACLM} = \frac{A_{bulk}E_{sat}L_{eff} + V_{gsteff}}{P_{clm}A_{bulk}E_{sat}}(V_{ds} - V_{dseff}) \tag{2.53}$$

$$V_{ADIBLC} = \frac{V_{gsteff} + 2v_t}{\theta_{rout}(L)(1 + P_{DIBLCB})}\left(1 - \frac{A_{bulk}V_{dsat}}{A_{bulk}V_{dsat} + V_{gsteff} + 2v_t}\right) \qquad (2.54)$$

$$\theta_{rout}(L) = P_{DIBLC1}\left[\exp\left(-D_{rout}\frac{L_{eff}}{2l_{t0}}\right) + 2\exp\left(-D_{rout}\frac{L_{eff}}{l_{t0}}\right)\right] + P_{DIBLC2} \qquad (2.55)$$

$$\frac{1}{V_{ASCBE}} = \frac{P_{scbe2}}{L_{eff}}\exp\left(-\frac{P_{scbe1}l}{V_{ds} - V_{dseff}}\right) \qquad (2.56)$$

$$l = \sqrt{\frac{\varepsilon_{si}T_{ox}X_j}{\varepsilon_{ox}}} \qquad (2.57)$$

Where V_{ACLM} and V_{ASCBE} are the Early voltages associated with channel length modulation and substrate current induced body effect respectively. The Early voltage associated with DIBL is known as V_{ADIBLC}, and V_{Asat} is the Early voltage of the I-V curves at the onset of saturation.

Conclusion

BSIM3 was developed in an effort to solve the problems of semi-empirical models and as a complement to BSIM 1-2. It has extensive built-in dependencies of important dimensional and process parameters such as channel length, width, gate oxide thickness, junction depth, doping concentration, and so on [24].

The model has evolved through three different versions. The first version forms the original basis for the model but had some severe mathematical problems. The second version was largely a correction of these mathematical difficulties, and several new parameters were introduced.

The third version that we have seen, has become an industry standard for modeling deep-submicron MOS technologies.

CHAPTER 3
OPTIMIZATION METHODS

3.1. Introduction

Many application problems in engineering, decision sciences, and operations research are formulated as optimization problems. Such applications include digital signal processing, structural optimization, engineering design, neural networks and computer aided design for VLSI etc….

Optimization problems are made up of three basic components: a set of unknowns or variables, an objective function to be minimized or maximized, and a set of constraints that specify feasible values of the variables. The optimization problem entails finding values of the variables that optimize (minimize or maximize) the objective function, while satisfying the constraints.

In this chapter, we address an important issue for different methods in solving nonlinear optimization problems (overcoming local minima). We first review existing nonlinear local and global optimization methods, such as gradient following methods, direct searching methods, and Combined methods, and identify their advantages and disadvantages. Then, we deduce that the so called FAST SIMULATED DIFFUSION method used by T. Sakurai et al. which combines global and local searches can be proposed to solve multi-minimum problems.

In recent decades, as computers become more powerful, many optimization methods have been developed and numerical optimization algorithms have been proposed for many applications especially in MOS transistor parameter extraction.

3.2. Optimization problems

A general minimization problem is defined as follows:

Given a set D and a function $f : D \rightarrow P$, find at least $X^{*} \in D$ that satisfies

for all $X \in D$, or show the non-existence of such a point. $f\left(X^{*}\right) \leq f\left(X\right)$

A mathematical formulation of a minimization problem is as follows:

$$\text{Minimize} \quad f(X) \tag{3.1}$$

$$\text{Subject to} \quad X \in D.$$

In this formulation, $X=(x_1, x_2, \ldots x_n)$ is an n-dimensional vector of unknowns. The function f is the objective function of the problem, and D is the feasible domain of X specified by constraints.

A vector, $X^* \in D$, satisfying $f(X^*) \leq f(X)$ for all $X \in D$ is called a *global minimizer* of f over D. The corresponding value of f is called a *global minimum*. A vector $X^* \in D$ is called a local minimizer of f over D if $f(X^*) \leq f(X)$ for all $X \in D$ close to X*. The corresponding value of f is called a *local minimum*.

The objective function of an optimization problem may or may not have a closed-form formula. Some objectives are evaluated deterministically and return the same value for the same set of variables every time. Other objectives are evaluated probabilistically, and could have different values every time they are evaluated.

3.3. A Taxonomy of optimization problems

A taxonomy of optimization problems is shown in Figure 3.1, in this figure optimization problems are classified according to the attributes of variable type, presence of constraints, and complexity

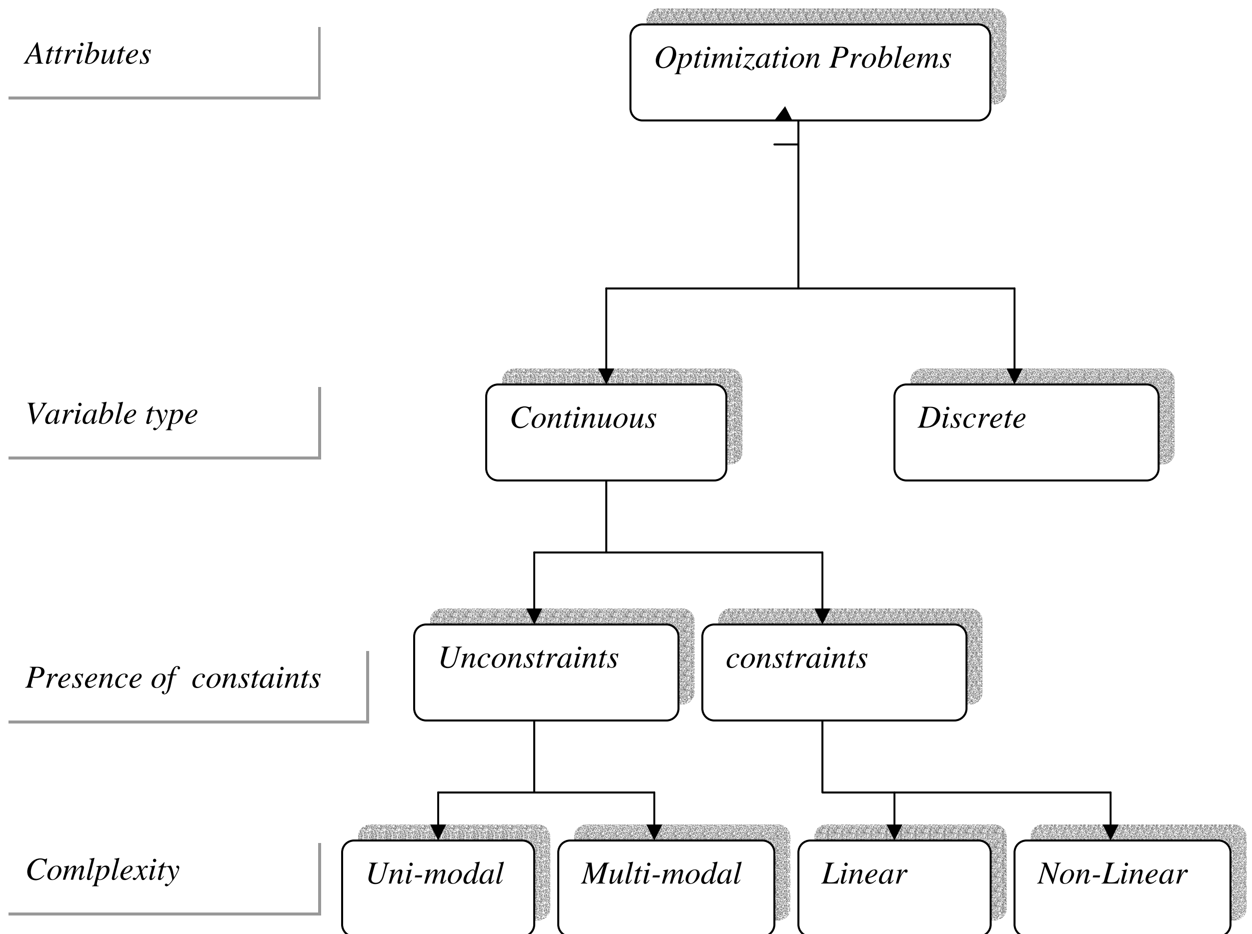


Figure 3.1: Classification of optimization problems [25].

- Continuous optimization problems

As indicated in Figure 3.1, optimization problems are classified into continuous and discrete problems. A problem is continuous if the unknowns (variables) take on continuous real values, for example *D* in eq.(3.1) consists of real numbers. A problem is discrete if the unknowns take on discrete, usually integer values.

Continuous optimization problems are classified into *constrained* and *unconstrained* optimization based on the presence of constraints.

In order to

$$\text{minimize } f(X) \tag{3.2}$$
$$\text{subject to } X \in R^n$$

There are two types of optimal points of an optimization problem :

-Local minimum which has the smallest value in a local feasible region surrounding itself.

-Global minimum which has the smallest value in the whole feasible domain.

In a continuous unconstrained optimization problem, an objective function is minimized in the real domain. An unconstrained optimization problem is *uni-modal* if its objective function is convex. A uni-modal problem has one local minimum, which is the global minimum at the same time. A problem is multi-modal if its objective function has more than one local minimum.

If each dimension of D in eq.(3.1) consists of real values constrained by simple lower and upper bounds, the corresponding optimization problem is called a *simple-bounded* continuous optimization problem.

$$\begin{aligned} &\text{minimize} \quad f(X) \\ &\text{subject to} \quad l \leq X \leq u \\ &\qquad\qquad\quad X \in R^n \end{aligned} \tag{3.3}$$

where l and u are constants.

Because simple-bounded constraints are easy to handle, and algorithms for problems without constraints and with simple-bounded constraints are similar, we put simple-bounded constrained problems in the class of unconstrained optimization. Hence, unconstrained nonlinear optimization problems and simple-bounded constrained problems are usually solved in similar ways.

An example of simple-bounded optimization problem is the *Rastrigin* function[25]. The minimization problem is formulated as follows:

$$\begin{aligned} &\min f(X) = x_1^{\,2} + x_2^{\,2} - \cos 18x_1 - \cos 18x_2 \\ &x \in R^2 \\ &\qquad -1 \leq x_i \leq 1, \qquad i = 1,2 \end{aligned} \tag{3.4}$$

Its global minimum is equal to -2 and the minimum point is at (0,0). There are approximately 50 local minima in the region bounded by the two constraints.

It should be noted that optimization methods are general and can be used for many different kinds of optimization problems. In this work we focus on the parameter extraction

techniques for CMOS, Peter R. K[26] has chosen to divide the algorithms into three categories:

- Gradient following methods.
- Direct searching methods
- Combined methods (combination of gradient following and direct searching methods)

The extraction algorithms based on optimization techniques try to minimize an objective function, also called error function or cost function. The value of the objective function is a measure of the distance between measured and calculated data points. Even after the minimization of the objective function there will be a difference between measured and calculated data points, this is because the model always is an approximation, and the data points contain measurement noise.

A general form commonly used objective function is as follows:

$$\|f(p_k)\|^2 = \sum_{i=1}^{N} f_i^2(p_k) = \sum_{i=1}^{N}\left(\frac{I_i(p_k) - I_i^*}{\max\left(I_i^*, I_{\min}\right)}\right)^2 \tag{3.5}$$

Where:

- *p_k is the parameter vector during the kth iteration.*
- *$I_i(p_k)$ is the calculated drain current at data point i.*
- *I_i^* is the measured drain current.*
- *I_{min} is the current limit above which the relative error is used and below which the absolute error (scaled by I_{min}) is used in the objective function.*
- *N is the number of data points.*

With this objective function, the relative differences between measured and simulated currents are minimized. The absolute differences between measured and simulated currents can also be minimized. These two different objective (error) functions have slightly different behaviors. While the absolute objective function favors large currents at the expense of the smaller currents, the relative objective function behaves in the opposite way. Therefore the current differences in equation (3.5) is divided by the maximum value of I_i^* and I_{min} to prevent the differences in the sub-threshold region for dominating.

3.4. Gradient following methods

Many researchers have proposed gradient following methods for extraction of model parameters. The most commonly used are:

- -The steepest descent method
- -The Gauss-Newton method
- -The Levenberg-Marquardts method

3.4.1 Steepest descent:

Steepest descent is one of the oldest and simplest methods. It is actually more important as a theoretical, rather than practical, reference by which to test other methods. However, `steepest descent' *steps* are often incorporated into other methods (e.g., Conjugate Gradient, Newton) when round-off destroys some desirable theoretical properties, progress is slow, or regions of indefinite curvature are encountered.

At each iteration of SD, the search direction is taken as $-g_k$, the negative gradient of the objective function at X_k. Recall that a descent direction P_k satisfies $g_K{}^T P_K \langle 0$.

The simplest way to guarantee the negativity of this inner product is to choose P_K=$-g_K$. This choice also minimizes the inner product - $g_K{}^T P_K$ for unit-length vectors and, thus gives rise to the name *Steepest Descent*. Its great advantage is its stability for points far from the minimum but its drawback requires modest storage, and progress toward a minimum may be very slow, especially near a solution.

P.Yang and P. K.chatterjee.[27][28] have combined the steepest descent and the Gauss-Newton method for parameter extraction, the first one is used only if the Gauss-Newton fails to converge. They proposed it as more efficient implementation than The Levenberg-Marquardt algorithm.

3.4.2. The Gauss-Newton method

In the Gauss-Newton method, $\left\| f(P_{k+1}) \right\|^2$ is approximated by a truncated Taylor series

$$\left\| f(P_k + \Delta P \right\|^2 \approx \left\| f(p_k) \right\|^2 + \Delta P^T g_k + \frac{1}{2} \Delta P^T G_k \Delta P \qquad (3.6)$$

Where:

$$g_k = \nabla\left(\left\| f(p_k) \right\|^2 \right)$$

$$G_k = 2 J_k{}^T J_k$$

J_k : is the Jacobian matrix, where its elements are defined as $J_{k,ij} = \partial fi(P_k) / \partial x_j$

and x_j are the elements of P.

T : stands for the transpose of a vector.

The difference between the updated parameter vector P_{k+1}, and the known parameter vector P_k is found from

$$\Delta P = +\alpha_k G^{-1} g_k \tag{3.7}$$

Where α_k is selected to minimize $\|f(p_{k+1})\|^2$. In this method, not only the slope of the error function but also its curvature are used to determine the direction of the search. This method doesn't need to compute the second derivatives, and converges rapidly for points close to the minimum, but can perform poorly for points far from the minimum, since a poor approximation to the Hessian is used.

3.4.3. The Levenberg-Marquardts method[2][12]:

The Levenberg-Marquardts method is a combination of the steepest descent and the Gauss-Newton methods.

The Levenberg-Marquardt nonlinear optimization algorithm finds values of model parameters such that the mean square error between simulated and experimental data is minimized:

$$\xi(P) = \frac{1}{N}\sum_{i=1}^{N} w_i^2 \left[y_i^{data} - y_i^{sim}(P) \right]^2 = \sum_i^N h_i^2(P) = \|h(P)\|^2 \tag{3.8}$$

Where N is the total number of points, P is the vector of model parameters, y_i^{data} is the value of the i-th experimental data point, y_i^{sim} is the i-th simulated data point, the wi's represent weights that can be assigned to each term and $h(P)$ is the error vector. In general, mean square error minimization is achieved by starting with an initial estimate of the parameter vector P^0 and iteratively updating it by taking a sequence of steps in error space

$$P^{k+1} = P^k + \delta P^k \tag{3.9}$$

such that the new mean square error $\xi(P^{k+1})$ is minimal along the chosen search direction. The vector δP^k represents the incremental update vector at each iteration k.

In the vicinity of a minimum in parameter space, $\xi(P)$ can be approximated by a second order multi-dimensional Taylor expansion around $P = P^k$:

$$\xi(P) - \xi(P^k) = g(P) \cdot (P - p^k) + \frac{1}{2}(P - P^k)^T H(P)(P - P^k) \qquad (3.10)$$

Where $g = \nabla\xi$ is the gradient vector according to $g_j = \partial\xi / \partial p_j$ and H is the symmetric Hessian matrix of second derivatives given by

$$Hij = \frac{\partial^2 \xi}{\partial p_i \partial p_j} \qquad (3.11)$$

The gradient can also be written as $g = 2J^T h$ where J is the Jacobian matrix given by $J_{ij} = \partial h_i(P) / \partial(p_j)$. Because the Hessian matrix H involves second derivatives which are computationally expensive, it is usually approximated using only first derivatives in the Gauss-Newton approach:

$$H \approx 2J^T J \qquad (3.12)$$

The parameter vector P must be found in order minimize the error sum $\xi(P)$. For each iteration step the incremental parameter update vector can be obtained by solving the linear system of equations:

$$H(P^k)\partial P^k = -2J^T h(P) \qquad (3.13)$$

The Levenberg-Marquardt method was introduced to regularize Newton's method because in practice the Hessian matrix often tends to be near-singular. The scalar parameter λ is added:

$$[H(P_k) + \lambda D_k]\delta P_k = -2J^T h(P) \qquad (3.14)$$

Where D is a diagonal matrix with $D_{ii} = H_{ii}$ and λ must be chosen such that $H + \lambda D$ is no longer near-singular. The Levenberg-Marquardt method reduces to the Gauss-Newton method for $\lambda \to 0$ and to the method of steepest descent for $\lambda \to \infty$.

The Levenberg-Marquardt algorithm is summarized below, with user-definable parameters in boldface:

1. start with λ=***lambda_init*** and an initial estimate for P^0
2. solve the system in equation (3.14) and let $P^{k+1} = P^k +$ ***damping factor*** $* \delta P^k$

3. if $\xi(P^{k+1})\langle\xi(P^{k})$ then

$$\lambda = \lambda / \textbf{\textit{lambda_scale}}$$

else

$$\lambda = \lambda * \textbf{\textit{lambda_fail}}$$

4. if not converged go to step 2.

A common values of the parameters used in FITDRF (is a general-purpose optimizer) at compile-time are shown in Table3.1. Finally, to determine whether the iterative procedure has found a minimum,

Table 3.1: A few default parameters built into the FITDRF optimizer[2].

Parameters	Value
Lambda_init	0.01
Damping_factor	1.0
lambda_ scale	8.0
Lambda_fail	10.0

Or if the iteration should be stopped for other reasons, several convergence criteria can be applied in practice:

- if the value of the error sum $\xi(P^{k})$ is small, the algorithm may become limited by machine accuracy and should be stopped. The iteration may be stopped even sooner if the desired accuracy is reached.
- if the relative change in the value of $\xi(P^{k})$ is small, the iteration should be stopped.
- if the value of the parameter λ becomes greater than some λ_{max}, then the algorithm is likely not to be able to find further improvement and the method fails.
- if λ has decreased below some minimal λ_{min} then the algorithm is likely to be wandering at the bottom of some valley in parameter space and it must be stopped.

It should also be noted that in practice the iteration should generally not be stopped on a step where $\xi(P^k)$ increases: that only shows that λ has not yet adjusted itself optimally [29].

The levenberg-Marquardt minimization algorithm commonly employed in MOSFET model parameter extraction has several known deficiencies, such as poor convergence characteristics without a good initial guess, low convergence characteristics to the globally optimal solution, and difficulty with simultaneous multiobjective optimizations [30].

The gradient following methods have been shown a deficiency that are trapped easily in local minima of the objective function. Another drawback of these methods is the existence of redundant parameters in the vector. Redundant parameters exist if the value of the objective function is independent or almost independent of the value of a certain parameters. Two or more parameters can also be redundant when considered jointly [30]. Ward and Doganis[31], suggested that the problem of redundant parameters can be detected by analyzing the curvature of the objective function at its minimum. One or more of the redundant parameters are then fixed to some value, for instance the theoretical value, and the parameter extraction is repeated.

The gradient following algorithms have converged when any of the following conditions are satisfied:

- The error at each point is less than some value.
- The relative change in $\|f(P)\|^2$ from one iteration to the next is less than some value.
- The norme of the gradient of the sum of squares with respect to the parameter vector $(\|\nabla_P(\|f(P)\|^2)\|)$ is less than some value.
- The change in each parameter from one iteration to the next is within some specified value.
- No adjustment of *P* can be found which reduces the error.

In order to overcome the difficulties mentioned above, these algorithms require an expert user with a detailed understanding of the MOSFET model and optimization methods to guide the parameter extraction process [31].

3.5. Direct searching methods

With the development of MOSFETs, a large number of the direct searching optimization methods have been widely used for parameter extraction. Since they require a great deal with the model to be used and the accuracy desired, several approaches have been developed according to the model evolution, in order to overcome the deficiencies presented in the other methods. A few of these methods are reviewed below.

3.5.1. The downhill simplex method

The *downhill simplex method* is due to Nelder & Mead (1965). The method requires only function evaluations, not derivatives. It is not very efficient in terms of the number of function evaluations that it requires.

A simplex is the geometrical figure consisting of N+1 points (or vertices) in N dimensions and all their interconnecting line segments, i.e., polygons, polygonal faces, etc. In two dimensions, a simplex is a triangle, in three dimensions it is a tetrahedron, and so forth. The simplex algorithm has been proposed for parameter extraction by Conway et al [32]. This algorithm consists of three movements of an original simplex.

1- Reflection.
2- Reflection and expansion.
3- Contraction.

To start the method we need to choose the first point to start. The algorithm is then supposed to make its own way downhill through the unimaginable complexity of an *N*-dimensional topography, until it encounters a minimum (at least local). The downhill simplex method must be started not just with a single point, but with *N*+1 points, defining an initial simplex. If there is one initial point $\mathbf{P}_0$ then the other *N* points can be expressed by:

$$\mathbf{P}_i = \mathbf{P}_0 + a_i\,\mathbf{e}_i \qquad (3.15)$$

Where $\mathbf{e}_i$ are *N* unit vectors, and a_i are constants that characterize the length scale for each vector direction. In the two dimension case Figure 3.2, $P^1{}_K$ and $P^2{}_K$ are generated by perturbing each component of the parameter vector , $P^0{}_K$. The objective function is evaluated for each parameter vector and the parameter vector with the largest value, $P^H{}_K$, and the smallest value, $P^L{}_K$, are identified.

The first attempt at a better vertex takes vertex $P^0{}_k$ (the worst one, which is $P^H{}_k$ in Figure 3.2 and reflects it through the centroid of the other vertices to try point $P^R{}_K$.

The reflected point is calculated form :

$$P^R{}_K = C + \alpha\,(C - P^H{}_K) \qquad (3.16)$$

Where α is a sizing constant ($\alpha \approx 1$) and C is the centroid of all points $P^i{}_k$ except $P^H{}_k$

$$C = \frac{1}{n} \sum_{i=0, i \neq H}^{n} P^i{}_K \qquad (3.17)$$

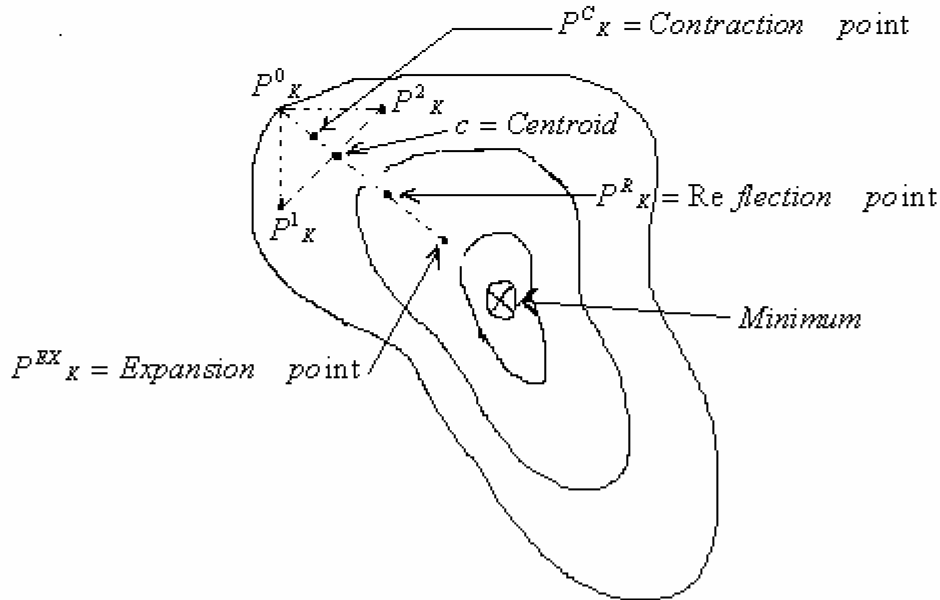


Figure 3.2: Illustration of the geometrical operations in the simplex direct searching optimization method.

The objective function at point $P^R{}_K$ is evaluated to face the following three cases:

a- The objective function at point $P^R{}_K$ is in the interval between that at point $P^L{}_K$ and $P^H{}_K$. In this case $P^R{}_K$ replaces $P^H{}_K$ giving a new simplex and a new $P^H{}_K$ is identified among the vectors $P^i{}_k$. and a new reflection is attempted.

b- If the objective function at point $P^R{}_K$ is smaller than that at point $P^L{}_K$, an expanded point is calculated through (in the same direction)

$$P^{EX}{}_K = C + \gamma (P^R{}_K - C) \qquad (3.18)$$

Where γ is a sizing constant ($\gamma \approx 2$). $P^L{}_K$ is replaced by the vector with smaller objective function of $P^R{}_K$ and $P^{EX}{}_K$. From this point , the perturbation process is repeated and a new reflection is attempted.

c- If the reflection results in a larger objective function at point $P^R{}_K$ than in at $P^i{}_K$, the minimum is probably between $P^H{}_K$ and C. A contraction is performed according to:

$$P^C{}_K = C + \beta(P^H{}_K - C) \tag{3.19}$$

Where β is a sizing constant ($\beta \approx 0.5$). Here $P^H{}_K$ is replaced with either $P^R{}_K$ or $P^C{}_K$ depending on which of them results in the smaller value of the objective function.

The method has converged when neither reflecting or contracting can find a better vector than $P^H{}_K$.

This method is slower than the gradient following methods, but according to Conway et al. this is the price to be paid for the increased flexibility of a robust algorithm [30].

3.5.2. Powell's conjugate direction method

This method used by Sugimoto[33] for parameter extraction of MOS transistors. Which is based on the reduction of the objective function in every step without using its derivatives. Powell's method for parameter extraction can be described as follows. Knowing the parameter vector P_0, and the N linearly independent directions ($\xi_1, \xi_2, ..., \xi_n$), the next four steps are repeated in each iteration until the algorithm has converged.

a- For r=1, 2,…,N calculate λ_r so that $\|f(p_{r-1} + \lambda_r \xi_r)\|^2$ has a minimum and define $p_r = p_{r-1} + \lambda_r \xi_r$.

b- Find the integer m, $1 \le m \le N$, so that $\left(\|f(p_{m-1})\|^2 - \|f(p_m)\|^2\right)$ has a maximum, and define $\Delta = \|f(p_{m-1}) - f(p_m)\|$.

c- Calculate, $f_3 = \|f(2p_N - p_0)\|^2$ and define $f_1 = \|f(p_0)\|^2$ and $f_2 = \|f(p_N)\|^2$.

d- If $f_3 \ge f_1$ or $(f_1 - 2f_2 + f_3)(f_1 - f_2 - \Delta)^2 \ge 0.5\Delta(f_1 - f_3)^2$ then use the old search directions $\xi_1, \xi_2, ..., \xi_N$ for the next iteration and use p_N for the next p_0 else define $\xi = p_N - p_0$, calculate λ so that $\|p_N + \lambda\xi\|^2$ has a minimum, use $\xi_1, \xi_2, ..., \xi_{m-1}, \xi_{m+1}, \xi_{m+2}, ..., \xi_N, \xi$ as the new search directions and $p_N + \lambda\xi$ as the starting point for the next iteration[30].

Figure 3.3 represents a two dimensional example of this optimization method, Assuming that ξ_1=(1,0), ξ_2=(0,1) and that P_0 is the initial parameter vector. According to :

Step # 1: $\|f(p)\|^2$ is minimized along ξ_1 resulting in P_1 and then along ξ_2 resulting in P_2.

Step # 2: m=1 is found.

Step # 3: Calculate f_1, f_2, and f_3

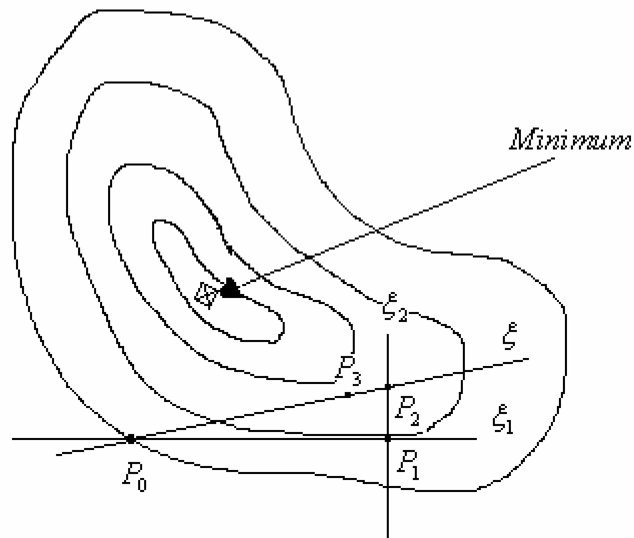


Figure 3.3: Illustration of the search path for Powell's conjugate direction method.

Step # 4: Assume that neither $f_3 \geq f_2$ nor

$(f_1 - 2f_2 + f_3)(f_1 - f_2 - \Delta)^2 \geq 0.5\Delta(f_1 - f_3)^2$ are satisfied. Next define $\xi = P_2 - P_0$ and minimize along this direction resulting in P_3. The search directions in this next iteration are equal to ξ_2 and ξ and the search for a minimum starts from P_3

This method requires $3N(N+1)$ function evaluations for N variables therefore is not so good for large size optimization problems. Also replacing the original directions by P_n-P_0 can lead to a set of directions that are linearly dependent, which results in only a subspace of the entire N dimensional space is explored for a minimum [34].

3.5.3 Simulated annealing method

All gradient following techniques can be trapped easily in local minima, and suffer from it. To overcome this problem, simulated annealing was developed in 1983 to deal with highly nonlinear problems. This method was introduced for parameter extraction by [30][35] [36].

Simulated annealing *(SA)* is a random-search technique which was inspired by a physical phenomenon. If we reduce the temperature of a liquid, the mobility of the molecules is lost. If the decrease is slow enough a pure crystal is formed, corresponding to a state of minimum energy (the annealing process). If the decrease is too fast a polycrystalline or an amorphous state with higher energy are reached. The analogy

between the reduction of the energy in the physical system and the reduction of the objective function in an optimization problem has lead to the definition of simulated annealing (SA) algorithms for combinatorial optimization problems.

In order to determine a new parameter vector, one of the extracted parameters is randomly chosen *X* and modified with a small random disturbance $Xnew = X + \Delta x$. Then the objective function is calculated with this new vector. If the value of the objective function is smaller for *Xnew* than for the old one, *Xnew* is accepted as in all optimization techniques Figure 3.4. Contrary to other methods, parameter vectors that results in larger values of the objective function can also be accepted. In this case, a random number in the interval [0, 1] is compared with a Boltzman-like probability distribution [30]

$$P = \exp(-\Delta\|f(X_k)\|/T) \qquad (3.20)$$

Where $\Delta\|f(X_k)\|$ the increase in *f* and *T* is a control parameter, which by analogy with the original application is known as the system *"temperature"* (often called pseudo-temperature) irrespective of the objective function involved. If *P* is larger than the random number, the new parameter vector is accepted, otherwise it is rejected and the old parameter is used again (regenerated).

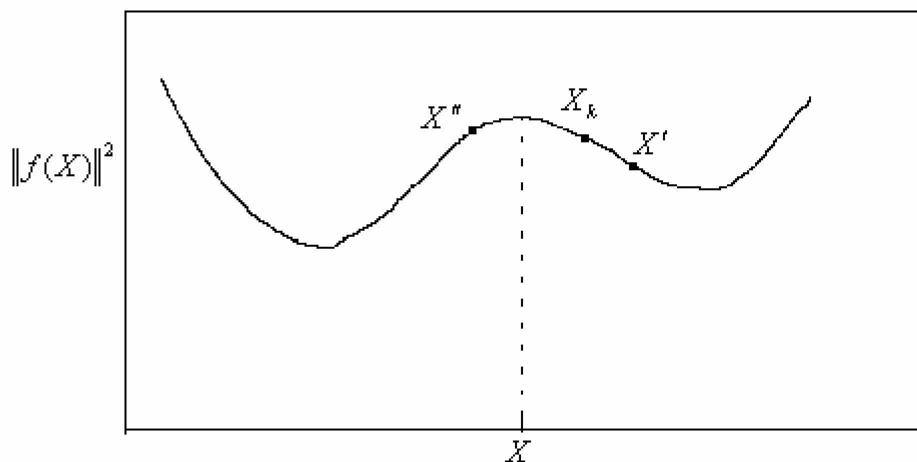


Figure 3.4: The vector X' is always accepted as the new parameter vector X_{k+1} since it results in a smaller value of the objective function $\|f(X)\|^2$ than X_k. If instead X'' is generated, this vector is accepted as the new vector X_{k+1} with the probability distribution given by Eq.(3.20).

The implementation of the basic SA algorithm is straightforward [37]. The following figure shows its structure Figure 3.5:

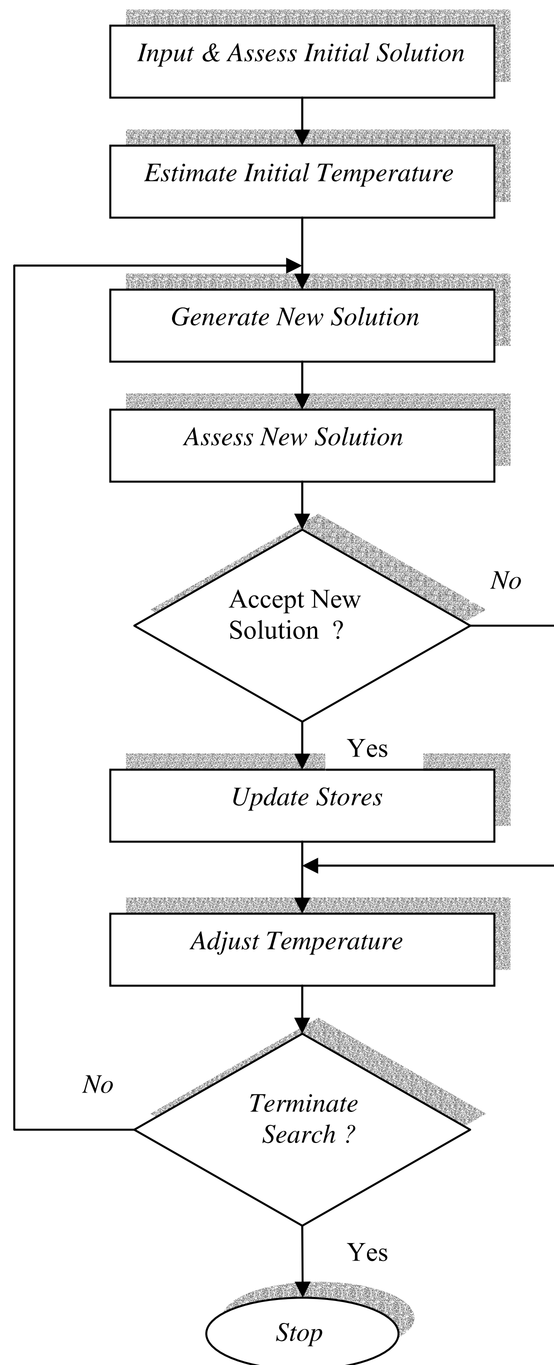


Figure 3.5: The structure of the simulated annealing algorithm.

To implement a simulated annealing, it is necessary to specify a certain number of ingredient compared to the diagram given above: how to determine the initial solution?, how to choose the initial pseudo-temperature?, how to choose the cooling system?, and how to choose the stop condition?[38].

- The initial solution: Basically, its choice is not of primary importance. It could thus be calculated in a way or another, for example by taking a configuration randomly, or a solution determined by an algorithm which gives a small value of the objective function.
- Initial value of the pseudo-temperature T: It must be sufficiently large so that many expensive transformations are accepted. Sakurai et al. [39] And White [40] proposed that the initial pseudo-temperature can be calculated from the standard deviation (σ) of the objective function as

$$T = k\sigma \tag{3.21}$$

Where k is a constant typically $k=0.2$ for Sakurai et al. And $k=20$ for White.
- Decrease of the temperature (cooling schedule) : A model often used for the decrease of T consists in choosing a function g defined by $T_{k+1}=g(T_k)=\mu T_k$ with $0<\mu<1$ (the series of the temperatures is thus a geometrical series tending towards 0). An alternative Sakurai et al.[39] and Huang et al.[41] proposed a formula such that the average error decreased in a uniform manner

$$T_{k+1} = T_k \exp(-\frac{\lambda T_k}{\sigma}) \tag{3.22}$$

Where λ is a constant smaller than one.

-Stop conditions: In some simple implementations of the SA algorithm the final temperature is determined by fixing

- the number of temperature values to be used, or
- the total number of solutions to be generated.

Alternatively, the search can be halted when it ceases to make progress. Lack of progress can be defined in a number of ways, but a useful basic definition is

- no improvement (i.e. no new best solution) at one temperature, combined with
- the acceptance ratio falling below a given (small) value P_f

More details about the stopping rules can be found in [42].

The main disadvantages of the simulated annealing method are computation-intensive, and weak method (does not use gradient information and makes relatively few assumptions about the problem being solved).

3.6. Combined Methods

In the literature, combined methods are used in parameter extraction in order to eliminate the problem of the local minima which most optimization techniques suffer from. Such as a combination of a gradient following algorithm and a direct searching process was presented by Wang et al [43], another combined method called simulated diffusion that uses of one part the gradient of the objective function and one stochastic part to determine the updated vector. The updated vector can be expressed mathematically as:

$$\Delta p = -\alpha \nabla (\| f(p_k) \|) + \sqrt{2T}\,dw \qquad (3.23)$$

where :

- α is a non-negative scalar
- T is the pseudo-temperature
- dw is the Gaussian random noise

When the temperature is high, the second term dominates which means that Δp is stochastic but when the temperature is low the first term dominates and Δp is in the direction of the gradient. The first term gives the tendency to minimize the objective function, and the second term introduces a random displacement which prevents getting trapped in a local minimum. There is a strict connection between simulated diffusion and simulated annealing. Equation (3.23) can be compared to the motion of a particle in a potential, which can be defined as follows

$$dx = -\nabla f(x)dt + \sqrt{2T}\,dw \qquad (3.24)$$

Where t is time

x is the space coordinate

$f(x)$ is a potential function

Therefore Δp is associated with *dx*, α is associated with *dt*, and $\| f(p_k) \|$ is associated with *f(x)*.

This method still too difficult to implement, as mentioned above is related to the simulated annealing method, and presents a CPU time consuming. That's why Sakurai et al [39] have proposed a modified algorithm of Simulated Diffusion method to accelerate its convergence, and they called it "Fast Simulated Diffusion."

They introduced two basic modifications which will be seen in more details in the next chapter.

Conclusion

In this chapter we have reviewed the optimization problems and the different optimization methods used in the parameter extraction. After an introduction about the optimization problems, and after a general description of the optimization methods, for each method we have described its algorithm or the way how it works and its drawbacks. We have concluded that all gradient following methods can be trapped in a local minimum, and the direct searching methods are slower or time consuming.

But, the combined methods are developed in such a way to overcome these problems, in this thesis we focus our work on *fast simulated diffusion* for its major advantage over other methods is the ability to avoid becoming trapped in local minima, its efficiency, and fastness.

CHAPTER 4
FAST SIMULATED DIFFUSION METHOD

4.1. Introduction

Since the early 1980s, the MOSFET has become the most widely used semiconductor device in very large scale integrated circuits. This is due mainly to the fact that the MOSFET has a simpler structure, costs less to fabricate, and consumes less power than its bipolar transistor counterpart.

Many compact models of the MOS transistor appeared with the development of CMOS technology. The first ones were very simple, based on physics, but used different equations for each operating region of the transistor. At the end of a long evolutionary process the family of BSIM models appeared. These models are made up by an extremely complex set of equations and often use more than one hundred parameters. These parameters are required during the design phase of an integrated circuit for simulation and tuning of the circuit's electrical behavior.

This leads to the so called "Parameter extraction process" which begins with a detailed measurement of the electrical characteristics of the devices. Typically, for MOSFETs the measured data is comprised of a series of Id-Vg (Drain current vs. Gate voltage) and Id-Vd (Drain current vs. Drain voltage) characteristics. And then appropriate values for the parameters of the corresponding models are found by fitting the measured data as closely as possible to simulated data. Many parameter extraction methodologies are currently employed, each for a specific device model (BSIM for MOSTEFs, Gummel-poon for BJTs,etc.). They are performed using several methods, most of them are local optimization, each of which requires a very specific set of data to be available, and requires experienced user guidance to succeed the parameter extraction process [30].

The Levenberg-Marquardt method minimization algorithm commonly employed in MOSFETs model parameter extraction has several known deficiencies, such as poor convergence characteristics without a good initial guess, low likelihood of convergence to the globally optimal solution, and difficulty with simultaneous multi-objective optimizations.

To overcome these deficiencies a new method called Simulated Diffusion has been used for global optimization. This method is conceived by the stimulus of simulated

annealing [41][44]. Many researches have been made to exploit this method, and it has been demonstrated that it can reach the global minimum. But it is very slow.

In this chapter, the FAST SIMULATED DIFFUSION optimization method is presented to ensure a faster way of convergence to the global minimum.

4.2. Parameter extraction

The simulations presented in this work were performed using the BSIM3v2 model, a physics-based, deep-submicron MOSFET model for digital and analog circuit designs from the Device Group at the University of California at Berkeley [45]. In order to fit a set of simulated curves to measured data, a large set of model parameters must be adjusted to appropriate values. Typically, one begins this process by setting each parameter to either a default value or a "best guess" of its appropriate value. The difficulty of the subsequent parameter extraction, as well as the quality of the solution found, depend heavily on the quality of these initial parameter settings. Once initial parameter values are specified, a series of optimizations is typically performed. Each successive optimization attempts to improve the quality of fit to some subset of the measured data by adjusting a subset of the model parameters. A good set of initial guesses, combined with a carefully and insightfully chosen sequence of optimizations can provide a set of parameters that fit the measured data as well as the model is able to. Unfortunately, this requires a very experienced individual to guide a lengthy, tedious process.

The Levenberg-Marquardt minimization algorithm can be very effective in parameter extraction, and is commonly employed [12]. The method is effective if the number of parameters to be adjusted is kept relatively small and a good initial guess for each parameter is available. However, as the number of parameters being optimized is increased, or the quality of initial guesses is decreased, the method will either fail to converge to a meaningful solution or be trapped in whatever local minimum is nearest the given starting guesses.

In contrast to most of the commonly used optimization methods, fast simulated diffusion algorithm can optimize many parameters at once, are not bound by initial starting guess, all information needed beforehand is on the parameter intervals P_{min}, P_{max}. The fast simulated diffusion algorithm we have used for parameter extraction is presented in the following sections.

4.3. Simulated Diffusion

The simulated diffusion method is based on an analogy with a physical phenomenon. That is when a small particle is placed in a given potential, the particle is diffused toward the global minimum of the given potential profile. This movement is known as Brownian motion. The mathematical formulation of this process can be described by the following differential equation

$$dx = -\nabla f(x)dt + \sqrt{2T}\,dw \qquad (4.1)$$

Where *t* is time, *dw* is Gaussian random noise, *T* is the temperature, *x* is the space coordinate which indicates the location where the particle is, and *f(x)* is a potential function in which the particle is put. If we consider T= 0 in eq.(4.1), we note that the trajectory *x* moves at each instant along the antigradient direction thus reducing the objective function values *f(x)* as *t* increases. Unfortunately, there is no guarantee that x converges to the global optimum as *t* goes to infinity. It will be more likely trapped in a local minimum. This is accomplished by adding the Brownian motion term (second term in eq.(4.1)). This term introduces a random displacement which prevents getting trapped in a local minimum.

As *t* goes to infinity, and with a proper cooling schedule, the probability distribution of *x, P(x)* approaches

$$P(x) \propto \exp\{-f(x)/T\} \qquad (4.2)$$

This means that the limit distribution is independent of the initial value and is peaked around the global minimizers of *f(x)*. This means that if dx is integrated over a long period of time, *x* tends to converge to a global minimum of the function *f(x)*. This is the principle of simulated diffusion. But it is very slow.

4.4. Fast Simulated Diffusion method

In this method, and in order to make the simulated diffusion faster, instead of directly integrating eq.(4.1), two modifications are introduced by Sakurai et al[39]. The first one is the introduction of an accept/nonaccept function of the so-called Boltzmann probability distribution, which is widely used in simulated annealing [28][42].

If $x_{next}=x+dx$ gives smaller function value than the current *x*, ($f(x_{next}) < f(x)$) take x_{next}. This is generally used by the optimization methods figure 4.1.

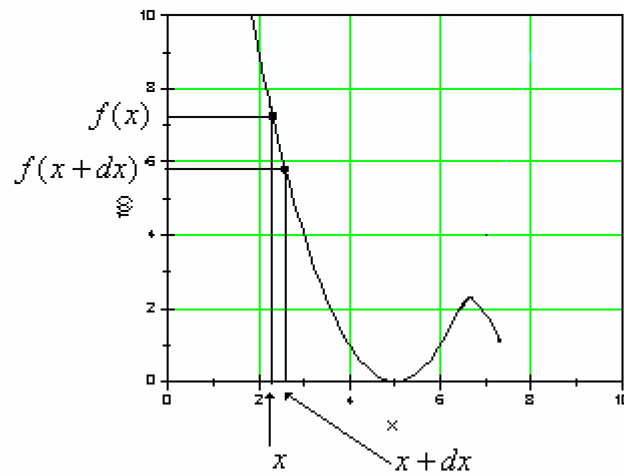


Figure 4.1: Shows the search path toward the minimum.

On the other hand, if x_{next} gives a larger function value than the current x, generate a random number R uniformly distributed in [0, 1] and calculate $P=exp[-\{f(x_{next})-f(x)\}/T]$. If $R<P$, then accept x_{next}.

Otherwise, discard x_{next} and regenerate x_{next}. That means if the object function value is considerably high, leads to the next move, and if it is less probably accept the move Figure 4.2. This point selection rule is mainly used in simulated annealing methods.

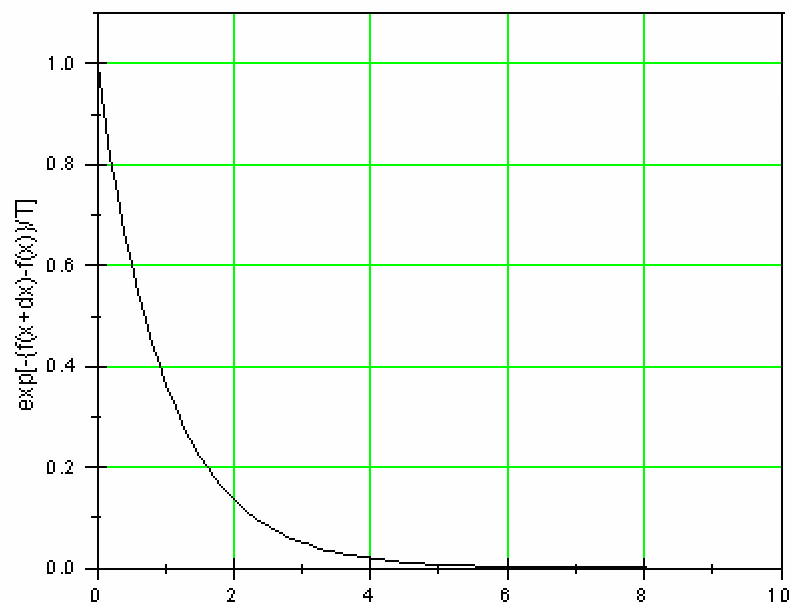


Figure 4.2: Show a rough curve of the Boltzmann distribution.

The other modification is how to generate the next move, Instead of the sum of the greedy hill-descending part (the first term in eq.(4.1)) and the random perturbation (the second term in eq.(4.1)) the generation of x is carried out alternately by a greedy search

and a random search. That is dx is calculated first as $-\nabla f(x)dt$ and is then calculated as $\sqrt{2T}dw$. At high temperature the hill-descending can be reached by adding the two terms, but the random term dominates and generates an ineffective moves and hence, all moves are possibly rejected.

To make the method more efficient, two major modifications are considered. First, since it is expensive to calculate the direction of $\nabla f(x)$ if the space has large dimensions, $\langle \nabla f(x) \bullet r \rangle$ r is used instead, where r is a unit vector of a randomly picked axis. This is because the expected direction of $\langle \nabla f(x) \bullet r \rangle$ r approaches $\nabla f(x)$ in the long run.

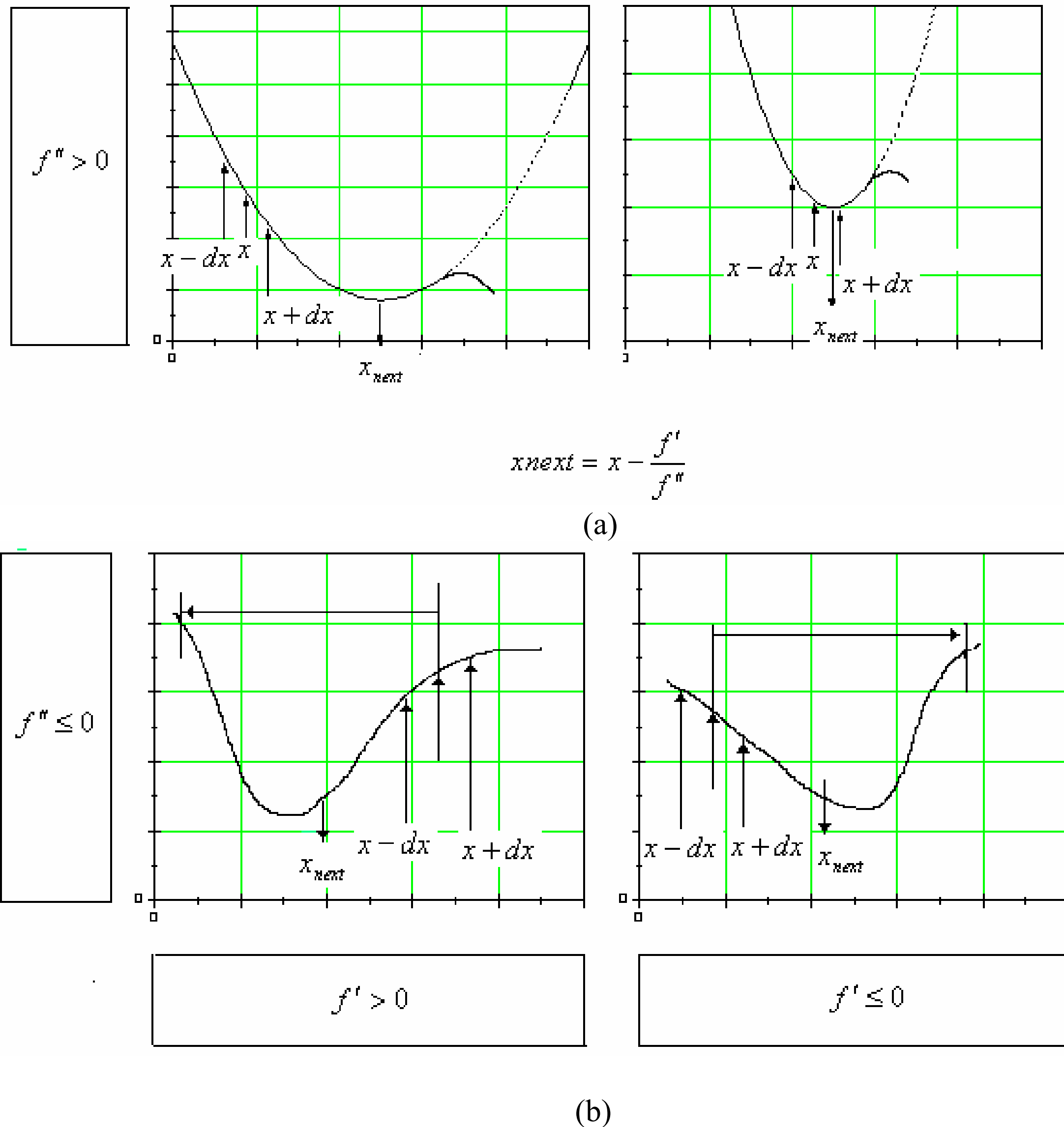
Second, since it is difficult to choose a good value of dt, a new hill-descending method is proposed and used[39]. The choice of dt is critical because if it is too small, the improvement of the solution is small, but if it is too big, $-\nabla f(x)$ dt does not always give an improvement.

Sakurai et al. have proposed a new hill-descending method using f' and f'' information figure 4.3. First pick a random axis direction.

If the function is concave at the point along the picked axis ($f''>0$), quadratic fitting is carried out and the minimum x in that direction is guessed and adopted as x_{next} figure 4.3a.

If the function is convex $(f(x) \leq 0)$, choose a small dx first and double the dx until $f(x+2^n dx)$ (where n is an integer) fails to decrease from $f(x)$, that means $f(x)< f(x+2^n dx)$ then take $x+2^{(n-1)}dx$ as a new point, and respecting the sign of f' the refining procedure is carried out to reach the minimum x which is adopted as x_{next} figure 4.3b.

The objective of this method is to provide an effective solution near the minimum, and is regarded as an inexpensive adaptive method for determining a good value of dt. But there is always a possibility that the random search can give rise to a big jump, in which case the previous hill-descending becomes wasteful.



(a)

(b)

Figure.4.3: Illustrates the generation of x_{next} using gradient information in one dimension.
In (a) f">0, quadratic fitting is carried out and x_{next} is guessed.
In (b) $f'' \leq 0$, shows the direction of the doubling process according to the sign of f' (convex or concave) and then x_{next} is guessed.

A rough flowchart of the Fast Simulated Diffusion method is shown in Figure 4.4. This algorithm is consists of two main loops, the first one is the outer loop with a varying temperature, and the second one is the inner loop with a constant temperature. In the inner loop the random and gradient search are carried out until the stop conditions met.

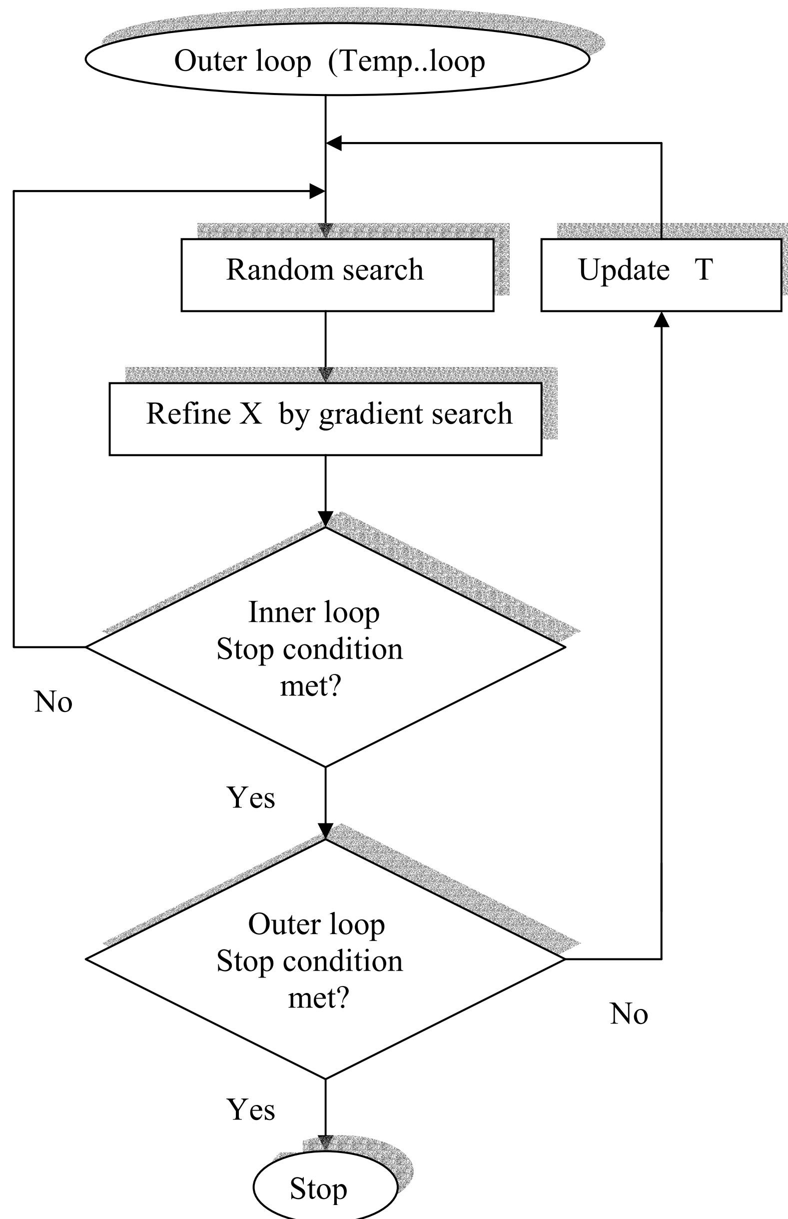


Figure 4.4: A rough flowchart of the Fast Simulated Diffusion

The detailed description of the FSD algorithm is shown in appendix C. The annealing schedule determines the degree of uphill movement permitted during the search and is thus critical to the algorithm's performance. The principle underlying the choice of a suitable annealing schedule is easily stated: the initial temperature should be high enough to *"melt"* the system completely and should be reduced towards its *"freezing point*" as the search progresses. To achieve this task the initialization scheme[40], and the temperature update[41] are adopted. The initial pseudo-temperature Tinit is calculated from the standard deviation of the objective function over randomly selected Ninit points. And the

cooling schedule is chosen to be slower than a geometric decrease to ensure that the global minimum will be reached.

The random search covers the feasible region [Xmin, Xmax] so that the randomly generated X falls in it. Otherwise it is regenerated. The multiplier S, controls the random search space volume. S should be shrunk in proportion to $\sqrt{T}$, in practice S can be reduced faster and is proportional to $\sqrt{T}^{n}$ (n=0.5~1.0).

For the external loop with constant temperature T, the stop criterion is chosen so that the counter iLast_Gasp is taken long enough and the temperature T gets low enough. Also, the iLast_Gasp counter gives the information when to decrease or to increase the temperature.

And for the inner loop, the first several (15~25 *dimension) loops, hill-descending research is not taken and only random research is carried out because at high temperature big jumps are accepted. This is controlled by the counter iINIT. This part is very similar to simulated annealing [42][44].

Conclusion

This method can be considered as a general global optimizer, it is not restricted to specific device models. The program source code is written in C++ and Matlab version 6.5 and is tested using many nonlinear test problems found in [46].

CHAPTER 5
RESULTS AND DISCUSSION

5.1. Introduction

The first generation of MOS transistor models has been used successfully when applied to 0.8 μm technologies and above. As geometries shrink below 0.8 μm, better models are required. Researchers in the Electrical Engineering and Computer Sciences department at The University of California at Berkeley have been leaders in the development of SPICE and the models used in it. In 1984 they introduced the BSIM1 model [14] to address the need for a better submicron MOS transistor model.

The BSIM1 model approached the modeling problem as a multi-parameter curve fitting exercise. The model contained 60 parameters covering the dc performance of the MOS transistor. There was some relationship to device physics, but in a large part, it was a non-physical model. Later, in 1991, UC Berkeley released the BSIM2 model that improved performance related to the modeling of output resistance changes due to hotelectron effects, source/drain parasitic resistance, and inversion-layer capacitance. This model contained 99 dc parameters, making it more unwieldy than the 60-parameter (dc parameters) BSIM1 model. In 1994, U.C. Berkeley introduced the BSIM3 model (version 2) which, unlike the earlier BSIM models, returned to a more device-physic based modeling approach. The model is simpler to use and only has 40 dc parameters. Moreover, the BSIM3 provides good performance when applied to analog as well as digital circuit simulation. In its third version, BSIM3v3 [4], it has become the industry standard MOS transistor model.

The BSIM3 model addresses the following important effects seen in deep-submicron MOSFET operation:

- Threshold voltage reduction
- Mobility degradation do to a vertical field
- Velocity saturation effects
- Drain-induced barrier lowering (DIBL)
- Channel length modulation
- Subthreshold (weak inversion) conduction
- Parasitic resistance in the source and drain

- Hot-electron effects on output resistance

The plot shown in Figure 5.1 shows a comparison of a 20/0.8 device using the Level 1, Level 3, and the BSIM3v3 models. The model parameters were adjusted to provide similar characteristics (given the limitations of each model). Assuming that the BSIM3v3 model closely approximates actual transistor performance, this figure indicates that the Level 1 model is grossly in error, the Level 3 model shows a significant difference in modeling the transition from non-saturation to linear region.

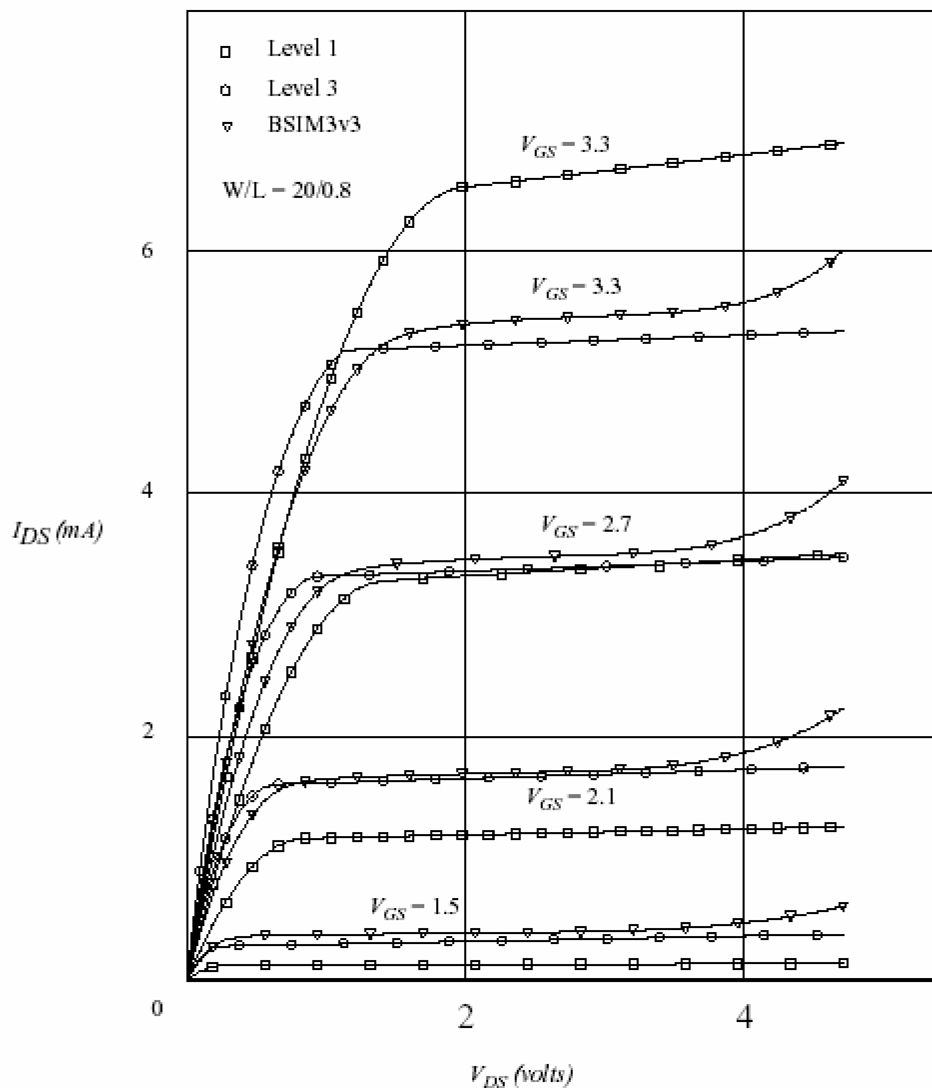


Figure5.1 Simulation of MOSFET Ids vs Vds characteristic using Level-1, Level-3, and the BSIM3v3 models.

In this work we have developed a source code of the FAST SIMULATED DIFFUSION method using C++ and Matlab to solve the multi-minimum global optimization problems. First we test our code using test optimization problems found in the literatures. And then we implement the BSIM3v2 model in the linear region using the experimental data underhand.

5.2. Test functions [46]

This section presents results of some traditional mathematical test problems for both local and global optimization. The code implementing Fast Simulated Diffusion method has been tested using these functions, and shows a good agreement with the results given below.

These test functions with different dimensions and different numbers of local and global minima that were used with our code, are known from the global optimization literatures. The dimension and the number of local and global minima are shown in Table 1.

Table 5.1: Number of local & global minimum in test functions

Function	Dimension	Number of local min.	Number of global min.
Roesenbrock	**2**	**1**	**1**
McCormic	**2**	**1**	**1**
Six Hump Camel	**2**	**6**	**2**

From these examples, the accuracy of our code can be tabulated as follows:

Table 5.2: A comparison of test functions results

Function	Error in Variables		Error in global min.
Roesenbrock	X1	0.18 %	≈ 0.0 %
	X2	0.36 %	
McCormic	X1	0.01 %	≈ 0.0 %
	X2	0.01 %	
Six Hump Camel	X1	0.87 %	0.03 %
	X2	0.85 %	
	X1	4.23 %	0.63 %
	X2	4.16 %	

For the Roesenbrock and McCormic functions the result were very good compared to the results given in the global optimization literatures, whereas for Six Hump Camel function the results were fairly good. Because the Six Hump Camel function has many oscillations, therefore the feasible region must be chosen carefully to get more accurate results.

5.3. BSIM3v3 results

The model parameter extraction problem is to minimize the object function

$$f(p) = \left[\frac{1}{n} \sum_{i=1}^{n} \left[\frac{I_{dmes} - I_{d\,\mathrm{mod}\,el}(p)}{I_{dmes}} \right]^2 \right]^{\frac{1}{2}} \quad (5.3.1)$$

With the model parameters , *p* as variables. In the above expression *Id* denotes the drain current. The BSIM3v2 model is used to extract model parameters in the linear region. These model parameters, *p*, that minimize *f(p)* are used for circuit simulation afterwards.

The following figures 5.2, 5.4, 5.5, 5.6, 5.7 show plots of measured and simulated characteristics of Ids vs Vds for Vgs=1.2v, 1.6v, 2.0v, and 2.4v. The devices which were used during the parameter extraction procedure have drawn W/L dimensions of 200/0.35µm, 200/0.40µm, 200/0.45µm, 200/0.50µm, and 200/0.55µm. The model predictions are relatively accurate. It can be seen that the modeling of the linear region IDS-VGS data for the shortest device at non-zero substrate biases is not very good. This was due to the absence of a parameter which would model the length dependence of the BSIM3 UC body-effect mobility degradation parameter, or a parameter which would model the dependence of the parasitic drain and source resistance on substrate bias. This model deficiency is not currently addressed in BSIM3v3 either.

The experimental data Idmes=f(Vds) for Vds<=0.1v (Linear region) for different Vgs with geometries specified above are obtained from [SPICE , station SUN] at CDTA. The following plots represent Ids vs Vds for different Vgs. The agreements between measured and simulated data are very good in this region. The average error is about 0.0899 %

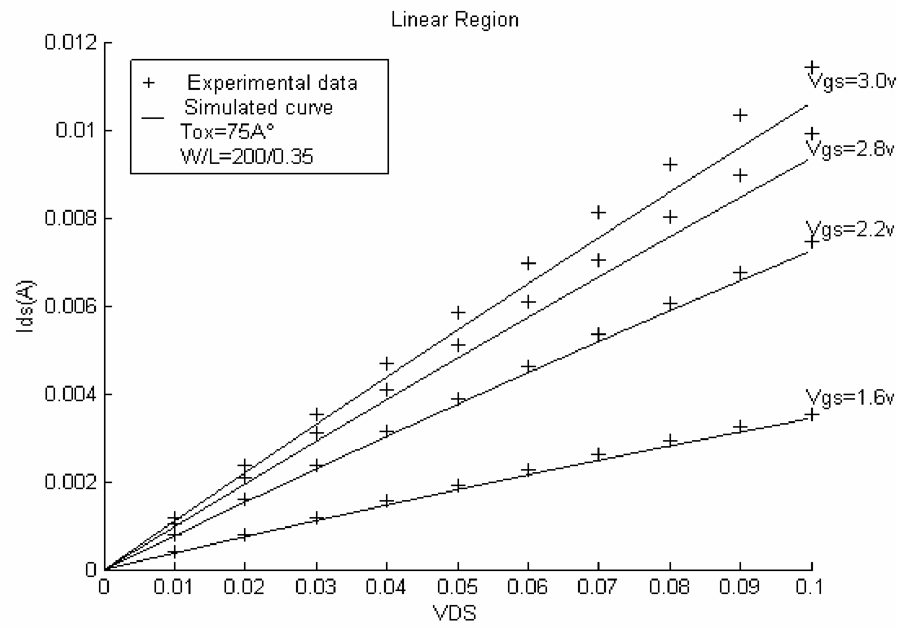


Figure 5.2 : Ids vs Vds for Vgs=1.6v, 2.2v, 2.8v, and 3.0v

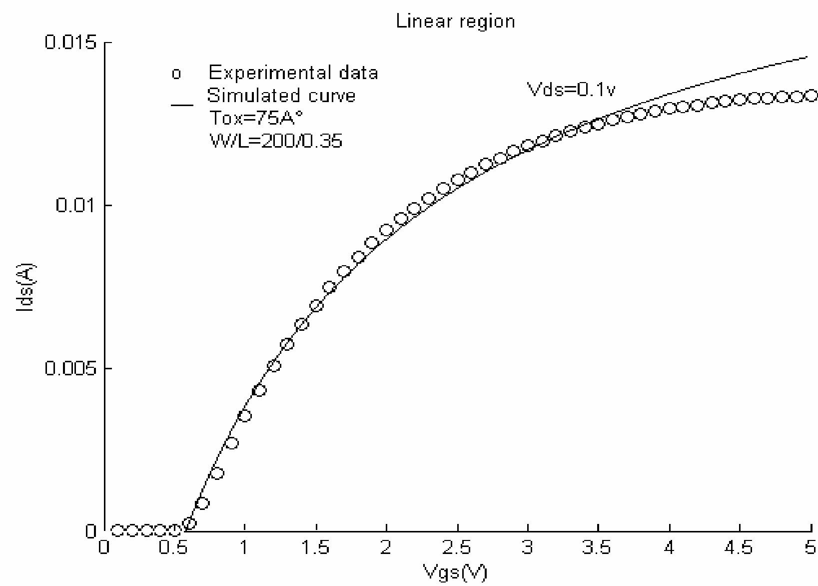


Figure 5.3 : Ids vs Vgs for Vds=0.1v

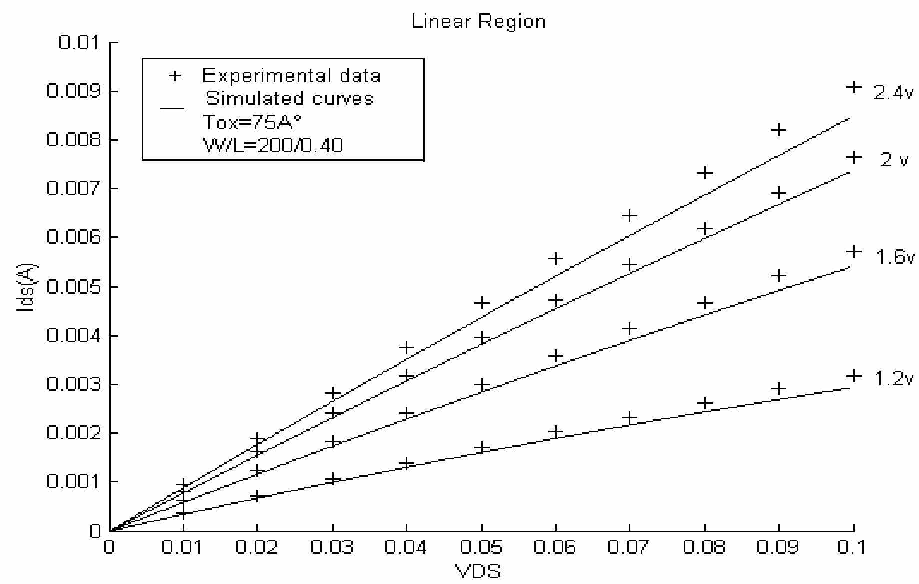


Figure 5.4 : Ids vs Vds for different Vgs values

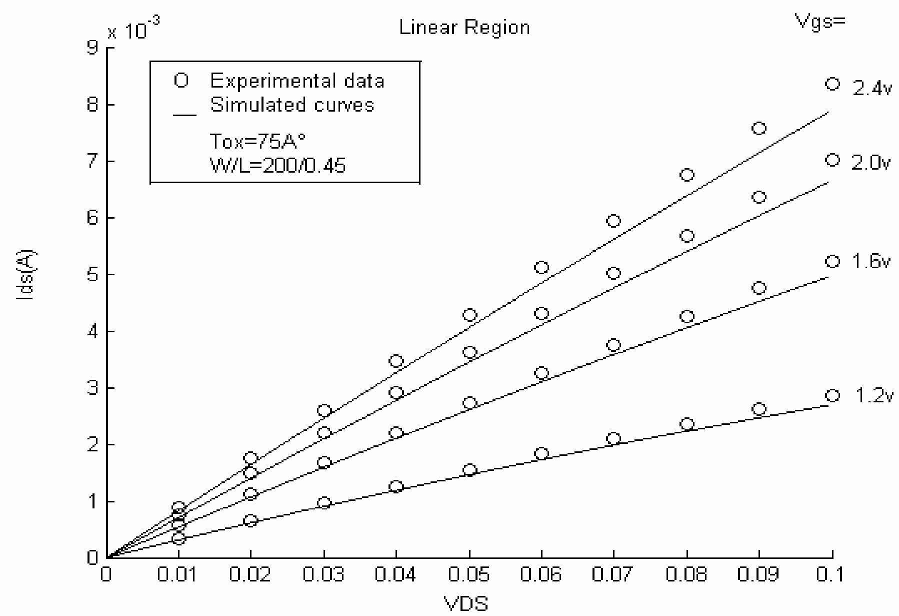


Figure 5.5 : Ids vs Vds for different Vgs values

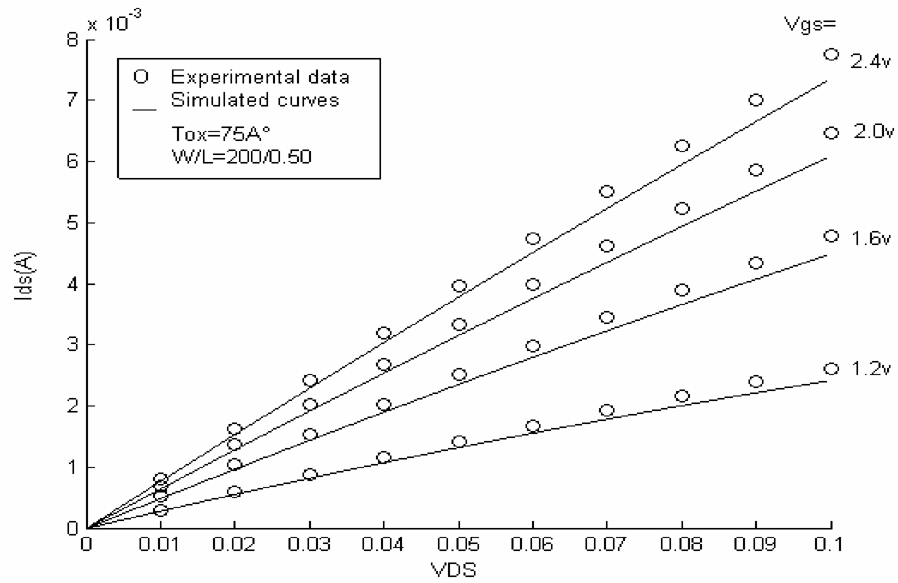


Figure 5.6 : Ids vs Vds for different Vgs values

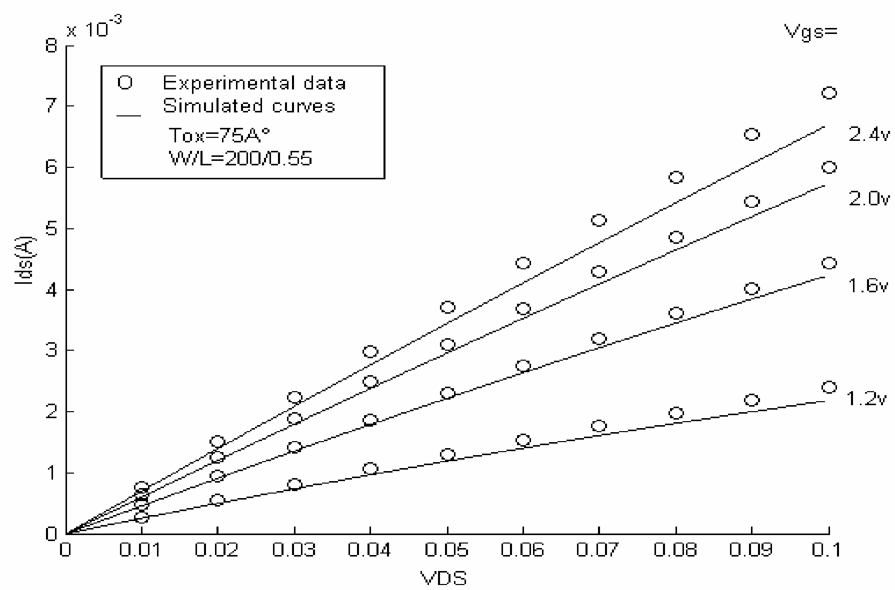


Figure 5.7 : Ids vs Vds for different Vgs values

Figure 5.8 shows Ids vs Vds for Vgs=3v, W/L=200/0.35 with Tox=75A°. Symbols represent experimental data from the extractor developed by Y. Harabi and M.T.Belaroussi, and solid line represent simulated curve. The error function found is 0.27%

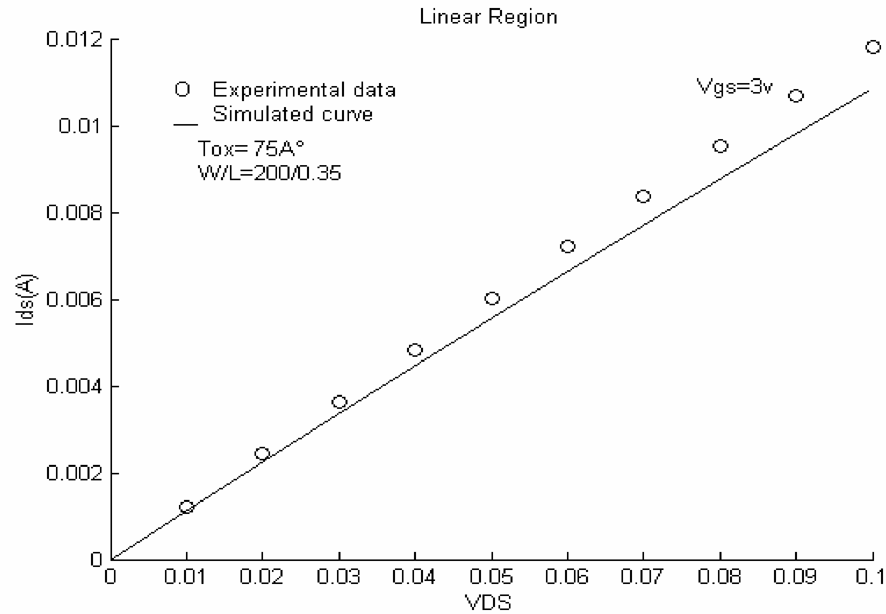


Figure 5.8 : Ids vs Vds for Vgs =3v

The errors in the extracted parameters which are the effective mobility μ_{eff}, the effective channel width W_{eff}, the effective channel length L_{eff}, the saturation velocity V_{sat}, the threshold voltage V_{th}, the charge charge A_{bulk}, and the source/drain series resistance R_{ds}, are presented in the following table :

Table 5.3 : A comparison of W/L=200/0.35 BSIM3v3 results

Parametrs	CDTA extractor	Our extractor	Error in %	Unit
μeff	0.03329	0.04485624	25.78	m^2/vs
Weff	200e-6	200.34e-6	-	m
Leff	0.34e-6	0.34e-6	-	m
Vsat	69500	66557.58	4.23	m/s
Vth	0.61	0.554	9.18	v
Abulk	1.1	1.0674	2.96	-
Rds	3.36	4.74	29.11	Ω

The experimental data used by T. Sakurai et al [39] for L=1um channel length and W=10um channel width is shown in Figure 5.9

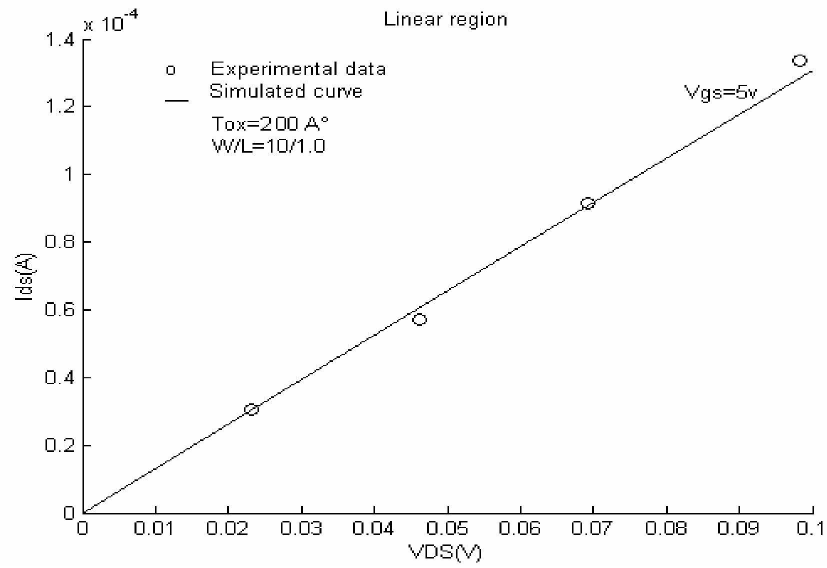


Figure 5.9: Ids vs Vds for Vgs =5v

And for L=0.25um , W=4um is shown in Figure 5.10

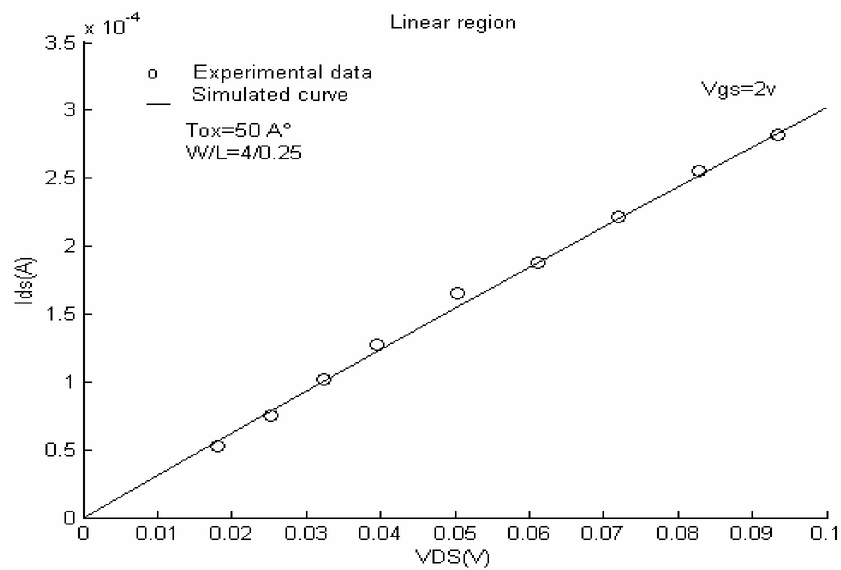


Figure 5.10: Ids vs Vds for Vgs =2v

These figures show good agreement between the measured (symbols) and simulated (solid line) data. This indicates the effectiveness of the BSIM3v2 model in the submicron region if the extracted model parameter set is used only for a narrow range of channel length. Usually the shortest channel length is used for almost all the MOSFET's in a VLSI and two or three sets of parameters are enough in designing a whole VLSI.

Conclusion

The proposed parameter extraction method for deep submicron MOSFETs, based on I-V characteristics, and the feasible region for each parameter, has been shown to be robust and accurate enough to stand on its own. Unlike, the classical methods the new in this work is the extraction of the source/drain series resistance R_{ds} and the saturation velocity v_{sat} from the I-V characteristics.

Even though it requires detailed knowledge of the transistor equations in use, the extraction method has been shown to be relatively easy to implement for different transistor models. Almost the same accuracy as that provided by global optimization techniques is obtained.

CONCLUSION

Parameter extraction is a critical task required during circuit design and simulation of MOSFET in today's technology. However, with the continued size reduction and changes in device structures, the applicability of these methods becomes limited.

Several different approaches to parameter extraction have been proposed. They include general or specialized, and direct or iterative extraction methods. Specialized methods extract some subsets of model parameters, for example, model resistances, or capacitances, or dc parameters only, while general methods determine all parameters of the model. . Direct extraction methods approximate model equations by linear functions and determine the values of parameters graphically or by solving linearized equations, while iterative methods fit the model responses to a set of measured characteristics by minimizing an objective function that quantitatively characterizes the fit.

In this thesis, we have seen different methods used in parameter extraction for MOSFETs technology, and the Fast Simulated diffusion used as a fast method for finding the global minimum of a multi-minimal function on multi-dimensional continuous space. This method shows accurate results when applied to a set of standard test functions, and is successfully applied to MOSFET model parameter extraction in the deep submicron region. The new in this work is the extraction of the source/drain series resistance R_{ds} and the saturation velocity v_{sat} from the I-V characteristics.

The results of Fast Simulated diffusion method for W/L = 200/0.35um MOSFET in the linear region is compared to that obtained by the Levenberg Marquadt, the first one shows very good results, this can be seen when we plot the measured data and the simulated curve on the same axis.

This method is believed to be applicable to other optimization problems encountered in system and VLSI designs

APPENDIX

Appendix A

BSIM3v3 model parameters

This appendix lists all model parameters and references used in BSIM3 version 3.0 MOS model from UC Berkeley. The first table presents the Basic Model Parameters, the second one presents the Length and Width Parameters, and the last one presents the Process Parameters, with their default values used as they implemented as Level 49 in HSPICE[18].

Basic Model Parameters

Name	Units	Default	Comments
VTH0 (*VTHO*)	V	0.7	threshold voltage of long channel device at V_{bs} = 0 and small V_{ds} (typically 0.7 for n-channel, - 0.7 for p-channel)
K1	$V^{1/2}$	0.50	first-order body effect coefficient
K2		-0.0186	second-order body effect coefficient
K3		80.0	narrow width effect coefficient
K3B	1/V	0	body width coefficient of narrow width effect
W0	m	2.5e-6	narrow width effect coefficient
NLX	m	1.74e-7	lateral nonuniform doping along channel
DVT0W	1/m	0	narrow width coefficient 0, for Vth, at small L
DVT1W	1/m	5.3e6	narrow width coefficient 1, for Vth, at small L
DVT2W	1/V	-0.032	narrow width coefficient 2, for Vth, at small L
DVT0		2.2	short channel effect coefficient 0, for V_{th}
DVT1		0.53	short channel effect coefficient 1, for V_{th}
DVT2	1/V	-0.032	short channel effect coefficient 2, for V_{th}
VBM	V	-5.0	maximum substrate bias, for V_{th} calculation

Name	Units	Default	Comments
U0	$cm^2/V/sec$	670	low field mobility at T = *TREF* = *TNOM* (670 for n-channel, 250 for p-channel)
UA	m/V	2.25e-9	first-order mobility degradation coefficient
UB	m^2/V^2	5.87e-19	second-order mobility degradation coefficient
UC	1/V	-4.65e-11 or -0.0465	body bias sensitivity coefficient of mobility -4.65e-11 for MOBMOD=1,2 or, -0.0465 for MOBMOD = 3
VSAT	m/sec	8e4	saturation velocity of carrier at T = *TREF* = *TNOM*
A0		1.0	bulk charge effect coefficient for channel length
AGS	1/V	0.0	gate bias coefficient of Abulk
B0	m	0.0	bulk charge effect coefficient for channel width
B1	m	0.0	bulk charge effect width offset
KETA	1/V	-0.047	body-bias coefficient of bulk charge effect
A1	1/V	0	first nonsaturation factor
A2		1.0	second nonsaturation factor
RDSW	ohm · μm	0.0	parasitic source drain resistance per unit width
PRWG	1/V	0	gate bias effect coefficient of RDSW
PRWB	$1/V^{1/2}$	0	body effect coefficient of RDSW
WR		1.0	width offset from Weff for Rds calculation
WINT	m	0.0	width offset fitting parameter from I-V without bias
LINT	m	0.0	length offset fitting parameter from I-V without bias
DWG	m/V	0.0	coefficient of Weff's gate dependence
DWB	$m/V^{1/2}$	0.0	coefficient of Weff's substrate body bias dependence
VOFF	V	-0.1	offset voltage in subthreshold region
NFACTOR		1.0	subthreshold region swing
ETA0		0.08	subthreshold region DIBL (Drain Induced Barrier Lowering) coefficient
ETAB	1/V	-0.07	subthreshold region DIBL coefficient

Name	Units	Default	Comments
DSUB		DROUT	DIBL coefficient exponent in subthreshold region
CIT	F/m^2	0.0	interface state capacitance
CDSC	F/m^2	2.4e-4	drain/source and channel coupling capacitance
CDSCD	F/Vm^2	0	drain bias sensitivity of CDSC
CDSCB	F/Vm^2	0	body coefficient for CDSC
PCLM		1.3	coefficient of channel length modulation
PDIBLC1		0.39	DIBL (Drain Induced Barrier Lowering) effect coefficient 1
PDIBLC2		0.0086	DIBL effect coefficient 2
PDIBLCB	1/V	0	body effect coefficient of DIBL effect coefficients
DROUT		0.56	length dependence coefficient of the DIBL correction parameter in R_{out}
PSCBE1	V/m	4.24e8	substrate current induced body effect exponent 1
PSCBE2	V/m	1.0e-5	substrate current induced body effect coefficient 2
PVAG		0	gate dependence of Early voltage
DELTA	V	0.01	effective Vds parameter
TOX	m	150e-10	gate oxide thickness
XJ	m	0.15e-6	junction depth
NSUB	cm^{-3}	6.0e16	substrate doping concentration
NCH	cm^{-3} (see Note 6)	1.7e17	peak doping concentration near interface
NGATE	cm^{-3}	infinite	Poly gate doping concentration
ALPHA0	m/V	0	the first parameter of impact ionization current
BETA0	V	30	the second parameter of impact ionization current
RSH	0.0	ohm/square	source/drain sheet resistance in ohm per square
MOBMOD	1		mobility model selector

Length and Width Parameters

Name	Units	Default	Comments
WLN		1.0	power of length dependence of width offset
WW	m^{WWN}	0.0	coefficient of width dependence for width offset
WWN		1.0	power of width dependence of width offset.
WWL	m^{WWN} $*m^{WLN}$	0.0	coefficient of length and width cross term for width offset
LL	m^{LLN}	0.0	coefficient of length dependence for length offset
LLN		1.0	power of length dependence of length offset
LW	m^{LWN}	0.0	coefficient of width dependence for length offset
LWN		1.0	power of width dependence of length offset
LWL	m^{LWN} $*m^{LLN}$	0.0	coefficient of length and width cross term for length offset

Process Parameters

Name	Units	Default	Comments
GAMMA1	$V^{1/2}$	see Note 8	body effect coefficient near the surface
GAMMA2	$V^{1/2}$	see Note 9	body effect coefficient in the bulk
VBX	V	see Note 10	VBX at which the depletion region width equals XT
XT	m	1.55e-7	doping depth
VBI	V	see Note 11	drain and source junction built-in potential

Appendix B

Drain Current Expression

$$I_{ds} = \frac{I_{ds0}(V_{dseff})}{1 + \frac{R_{ds} I_{ds}(Vgseff)}{V_{dseff}}} \left(1 + \frac{V_{ds} - V_{dseff}}{V_A}\right)\left(1 + \frac{V_{ds} - V_{dseff}}{V_{ASCBE}}\right)$$

$$I_{ds0} = \frac{W_{eff}\,\mu_{eff}\,C_{ox} V_{gsteff}\left(1 - A_{bulk}\frac{V_{dseff}}{2\left(V_{gsteff} + 2vt\right)}\right)V_{dseff}}{L_{eff}\left[1 + V_{dseff} / \left(E_{sat} L_{eff}\right)\right]}$$

$$V_A = V_{Asat} + \left(1 + \frac{P_{vag} V_{gsteff}}{E_{sat} L_{eff}}\right)\left(\frac{1}{V_{ACLM}} + \frac{1}{V_{ADIBLC}}\right)^{-1}$$

$$V_{ACLM} = \frac{A_{bulk} E_{sat} L_{eff} + V_{gsteff}}{P_{CLM} A_{bulk} E_{sat} litl}\left(V_{ds} - V_{dseff}\right)$$

$$V_{ADIBLC} = \frac{\left(V_{gsteff} + 2vt\right)}{\theta_{rout}\left(1 + P_{DIBLCB} V_{bseff}\right)}\left(1 - \frac{A_{bulk} V_{dsat}}{A_{bulk} V_{dsat} + V_{gsteff} + 2vt}\right)$$

$$\theta_{rout} = P_{DIBLC1}\left[\exp\left(-D_{ROUT}\frac{L_{eff}}{2l_{t0}}\right) + 2\exp\left(-D_{ROUT}\frac{L_{eff}}{l_{t0}}\right)\right] + P_{DIBLC2}$$

$$\frac{1}{V_{ASCBE}} = \frac{P_{scbe2}}{L_{eff}}\exp\left(\frac{-P_{scbe1} litl}{V_{ds} - V_{dseff}}\right)$$

$$V_{Asat} = \frac{E_{sat} L_{eff} + V_{dsat} + 2R_{DS} v_{sat} C_{ox} W_{eff} V_{gsteff}\left[1 - \frac{A_{bulk} V_{dsat}}{2\left(V_{gsteff} + 2vt\right)}\right]}{1/\lambda - 1 + R_{DS} vsat C_{ox} W_{eff} A_{bulk}}$$

$$litl = \sqrt{\frac{\varepsilon_{si} T_{ox} X_j}{\varepsilon_{ox}}}$$

Appendix C

Detailed algorithm of Fast Simulated Diffusion method

```
Main {
    // Set initial temperature by using huristics. k=0.2, Ninit=200
    // σ is the standart deviation of f(x) over randomly selected Ninit points
        T=Tinit=k* σ ;
    // Set initial S to Sinit
        S=Sinit;
    // Set initial X to Xinit given by the user or one of those randomly selected Ninit
    // points whichever gives the minimum value of f
        Xopt=X=Xinit
    // External loop with varying T
    // While Last_Gasp loop is not taken long enough and T is not gets low enough
    While( iLast_Gasp ≤ iLast_Gasp_Max) and (T/Tinit ≥ T_Ratio_Min)
        {
         // Set counter to zero
         iINT=0;
         // Internal loop with constant T
         While(a certain times (ex. 15~25 * dimension)
             {
              // Increment counter
              iINT++;
              // Generate new X by simulated diffusion
              Generate_X( );
                //Calculate Δf
                Δf =f(Xnew)-f(X);
                // If cost decreases
                If ( Δf < 0)
                   { // adopt the new X
                     X=Xnew;
                     // Compare Xopt with X
                     If ( f(X)<f(Xopt))
                        //save best X
                        { Xopt = X }
                   }
                else  {
                         //even if cost increases, calculate the Boltzman
                         // distribution
                          P=exp(- Δf /T);
                          R=random number uniformly distributed in [1, 0]
                          If (R<P)
                             // adopt Xnew according to Boltzman distribution
                             { X=Xnew}
                      }
             }
              // Resume the best X
             If ( f(X)>f(Xopt))
                { X=Xopt }
```

```
        // If cost is not improved considerably, take Last_Gasp loop, where
        // T is increased a little and then decreased to freez.
        If (cost is not improved considerably)
            { iLast_Gasp ++; }
            else  { iLast_Gasp =0; }
         Update_T( );
         Update_S( );
       }
       Solution = Xopt;
     }

// Generate Xnew
Generate_X( )
  {
     // In the first mINT(ex.=15) loops the hill-descending method is not taken.
     // gradient_Flag=0, for gradient search. And gradient_Flag=1, for random
     // search.
      If (iINT<mINT)
          { gradient_Flag=0; }
     else
           { gradient_Flag=1- gradient_Flag; }
     If  (gradient_Flag = = 1 )
        {  Randomly select single variable Xi and move only in this axis. Generate
         Xnew with gradient information according to f' and f'' values. }
     else
    // Xnew is chosen randomly in the interval specified as
      { Xnew= Rand*(Xmax-Xmin)-Xmin;
    // or using the Gaussian or Lorentzian distribution
       Xnew= X+S* (n^th dimensional Gaussian or Lorentzian distribution) }
  }// Update the temperature
Update_T( )
   {
     If ( iLast_Gasp= 0)
          { // σ  is the standard deviation of accepted f(X), and  λ=0.7.
               T_Factor = exp(-λ T/σ);
               If (T_Factor < T_Factor_Min)
                   { T_Factor= T_Factor_Min(=0.5) }
                    T*= T_Factor
          }
     If (1≤ iLast_Gasp  ≤ n2)
         { T*=T_Factor2(T_Factor2>1,ex.1.3) } // n2=4
     If (n2 ≤ iLast_Gasp )
         { T*=T_Factor1(T_Factor1 < 1,ex.0.75) }

// Update the control parameter S
Update_S( )
     { S = Sinit * (T/Tinit)^a ; // a=0.5~ 1(ex. a=0.75) }
```

REFERENCES

1. Gordon E. Moore " Cramming more components onto integrated circuits" electronics, Volume 38, Number 8, April 19, (1965).
2. Eric Pop, "CMOS Inverse Doping Profile Extraction and Substrate Current Modeling", thesis, (June 1999).
3. Hspice Manual, Release (February 2001).
4. Y. Cheng, M. Chan, K. Hui, M. Jeng, Z. Liu, J Huang, K. Chen, J. Chen, R. Tu,P. K. Ko, C. Hu " BSIM3v3" (final version) (1996).
5. L. K. Yong "Design, Modelling, and Characterisation of Submicron MOSFETs" Phd Thesis, Nanyang Technological University, (2001).
6. J. He, X. Xi, M. Chan, K. Cao, C. Hu, Y. Li, X. Zhang, Ru Huang, and Y. Wang, " Normalized Mutual Integral Difference Method to Extract Threshold Voltage of MOSFETs ", IEEE Electron Device Letters, Vol. 23, No. 7, pp 428-430 (July 2002)
7. Y. Taur," MOSFET Channel length: Extraction and Interpretation" IEEE Electron Device Letters, Vol. 47, No. 1, pp 160-169 (January 2000)
8. K. Ahmed,I. De, C. Osburn, J. Wortman, and J. Hauser "Limitations of the Shift-and-Ratio Technique for Extraction of the Bias Dependent of Leff and Rsd of LDD MOSFETs" IEEE Trans.On Electron Devices, Vol. 47, No. 4, pp 891-893 (April 2000).
9. G. Niu, J. D. Cressler, S. J. Mathew, and S. Subbanna "A Total Resistance Slope-Based Effective Channel Mobility Extraction Method for Deep Submicrometer CMOS Technology " IEEE Trans.On Electron Devices, Vol. 46, NO. 9, PP 1912-1914 (September 1999)
10. W. Fikry, G. Ghibaudo, H. Haddara, S. Cristoloveanu and M. Dutoit " Method for extracting deep submicrometer MOSFET parameters " Eelectronics Letters Vol.31 No.9 (April 1995).
11. F. Prégaldiny, " Étude et modélisation du comportement électrique des transistors MOS fortement submicroniques", thèse, Univ. Louis Pasteur, (Dec. 2003).
12. Y. Harabi, M. T. Belaroussi, "Etude et Réalisation d'un Extracteur de Paramètres SPICE " CDTA (Juillet 1994).

13. N. M. Kriplani, " Transistor Modeling using Advanced Circuit Simulator Technology" thesis, Faculty of North Carolina State University, (2002).
14. "HSPICE User's Manual ", 16_mosfet_ models, Meta-Software, Inc. (1996)
15. D. P. Foty, " MOSFET Modeling and SPICE ", Tutorial at the Custom Integrated Circuits Conference Santa Clara, California, (May 1997)
16. BSIM3v3 MOSFET model – User's manual, Univ. Of California, Berkely,(1996) [Online]. Available : http://www-device .eecs.berkeley.edu/~bsim
17. BSIM4 MOSFET model – User's manual, Univ. Of California, Berkely,(2000) [Online]. Available : http://www-device .eecs.berkeley.edu/~bsim
18. MM9 Model, Philips Semiconductors, (2001) [Online]. Available : http://www.semiconductors.philips.com/philips_models/mos_models
19. M. Bucher, C. Lallement, C. C. Enz, F. Theodolz, and F. Krummenacher, " The EPFL -EKV MOSFET model, version 2.6," EPFL, Tech. Rep., (1999) [Online]. Available : http://legwww.epfl.ch/ekv/
20. M. L. L. Chng," Critical Comparison of a New MOSFET Channel Length Extraction Method With Other Existing Methods ", thesis, University of Queensland, (October 2000).
21. B. Van Zeghbrock " Principles of Semiconductor Devices", (2002).
22. F. J. G. Sanchez, A. O. Conde, G. D. Mercato, J. A. Salcedo. J. J. Liou, Y. Yue, J. Finol "Extracting the Threshold Voltage from the sub-threshold to Strong Inversion Transition Region of MOSFETS". (1998).
23. S. Healy, K. McCarthy, D. B. M. Klaassen " Comparison of BSIM3v3 and MOS MODEL 9 " (1997).
24. H. Abebe, V. C. Tyree, " BSIM3v3.1 Model Parameters Extraction and Optimization", (October 2000).
25. Y. Shang, " Global Search Methods For Solving Nonlinear Optimization Problems" thesis, Univ. of Illinois at Urbana-Champaign, (1997).
26. Peter R. Karlsson, "Direct Extraction of MOS Transistor Current Model Parameters". Goteborg (1994).
27. P. Yang and P.K. Chatterjee, " SPICE Modeling for small Geometry MOSFET Circuits ". IEEE Trans.Computer-Aided Design of integrated Circuits and Systems. Vol.CAD-1, No. 4, pp 169-182, (1982).
28. P. Yang and P.K. Chatterjee, " An Optimal Parameter Extraction Program for MOSFET Models". IEEE Trans.Electron Devices. Vol.ED-30, No. 9, pp.

1214-1219, (1983).

29. W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery. " Numerical Recipes in C ", 2nd. edition. Cambridge University Press, 1992.
30. Milan Keser, Kuntal Joardar "Genetic Algorithm Based MOSFET Model Parameter Extraction ". paper, (1999).
31. D. E. Ward and K. Doganis. "Optimized extraction of MOS model parameters", IEEE Trans. Computer-Aided Design of Integrated Circuits and Systems, Vol. CAD-1, No.4 pp 163-168, (1982).
32. P. Conway, C. Cahill, W. A. Lane, and S. U. Lidholm, " Extraction of MOSFET parameters Using the Simplex Direct Search Optimization Method ", IEEE Trans. Computer-Aided Design, Vol.CAD-4, No. 4, pp.694-698, (1985).
33. M. Sugimoto, " General-purpose model parameter extraction program with initial value exploration technique ", Proc. IEEE Trans 1986 Custom Integrated Circuits Conf., pp. 624-627, (1986).
34. Physics " Minimization Maximization of functions " .
35. M. K. Vai, S. Prasad, N. C. Li, and F. Kai, " Modeling of Microwave Semicoductor Devices Using Simulated Annealing Optimization ", IEEE Trans. Electron Devices, Vol. ED-36, No. 4, pp.761-762, (1989).
36. R. Niutanen, M. Valtonen, K. Mannersalo, Optimization Methods in APLAC, (December 1992).
37. Wei MI. " Extraction des paramètres et domaine de validité du modèle d'un composant de puissance ", pp. 67-69 Thèse Année (2002).
38. I. Charon, A. Germa, O. Hurdy, Methodes d'optimization combinatoire, Paris, Masson,(1996).
39. T. Sakurai, B. Lin, R. Newton, " Fast Simulated Diffusion: An Optimization Algorithm for Multiminimum Problems and Its Application to MOSFET Model Parameter Extraction ", IEEE Trans. On Computer-Aided Design, Vol. 11, No. 2 (February 1992).
40. S. R. White, " Concepts of scale in simulated annealing ", proc. IEEE Int. Computer Design, pp. 646-651, (1984).
41. M.D. Huang F.Romeo. and A.Sangiovanni-Vincentelli, " An Efficient General Cooling Schedule for simulated annealing ", Proc. IEEE 1986 Int. Conf. Computer-Aided Design (ICCAD86), pp. 381-384, (1986).
42. M. Locatelli, "Simulated Annealing Algorithms for Continuous Global Optimization"

Dipartimento di Informatica Universita de Torino – ITALY

43. S. J. Wang, J.Y. Lee, and C. Y. Chang, " An Efficient and Reliable Approach for Semiconductor Device Parameter Extraction ", IEEE Trans. Computer-Aided Design. Vol. CAD-5, No. 1, pp. 170-179, (1986).
44. S. Kirkparik, C. D. Gellatt, M. P. Vecchi, "Optimization by simulated annealing" Science, vol. 220, no. 4598, pp. 671-680, (May 1983).
45. S. Healy, K. McCarthy, D. B. M. Klaassen " Comparison of BSIM3v2 and BSIM3v3.95.1 MOSFET Models " (1997).
46. K. Mdsen, J. Zilinskas " Testing branch-and-bound methods for global optimization", (May 2000).