# Abstract

KRIPLANI NIKHIL, M. Transistor Modeling using advanced circuit simulator technology. (Under the direction of Michael B. Steer)

The advanced MOSFET model based on the Berkeley Short Channel IGFET Model (BSIM) version 4 is implemented in the circuit simulator Transim. The model is implemented as a charge controlled model using object-oriented programming and automatic differentiation. The result is a dramatically simplified approach to implementing the BSIM4 model in a simulator. The modeling technique does not use the associated discrete modeling approach commonly used in circuit simulators with the result that off-the-shelf numerical solvers can be used. The model is a simulator independent model and the same model code can be used for DC, transient and harmonic balance analysis. Implementation of the model was completed in 7 months with 17 pages of C++ code compared to the original code for the model implemented in SPICE that was 200 pages long. Results for an NMOS circuit are presented for DC and transient analysis.

# Transistor Modeling using Advanced Circuit Simulator Technology

by

**NIKHIL M. KRIPLANI**

A dissertation submitted to the Graduate Faculty of
North Carolina State University
in partial fulfillment of the
requirements for the Degree of
Master of Science

**ELECTRICAL ENGINEERING**

Raleigh

2002

**APPROVED BY:**

Chair of Advisory Committee

# Biographical Summary

Nikhil M. Kriplani was born on 30th August, 1977 in Mumbai, India. He received a degree in Electronics and Telecommunications engineering in 1999 from the Maharashtra Institute of Technology in Pune, India. From July 1999 to May 2000 he worked with Philips, India where he spent some time with the Philips Speaker Systems business group and was actively involved in loudspeaker design for the Indian audio market. He also worked on a team involved in a joint project with Philips and PriceWaterHouse Coopers in the analysis and re-design of the Philips supply chain. He was admitted to the Master's program at North Carolina State University in the Fall of 2000. His interests are in the fields of analog circuit design and computer-aided analysis of circuits.

# Acknowledgments

I would like to express my sincere gratitude to my advisor, Dr. Michael Steer for giving me an opportunity to work with his research group and supporting me through the most part of my graduate studies. It has provided me with a great chance to explore and learn, for which I am grateful. I would also like to thank Dr. Griff Bilbro and Dr. Paul Franzon for serving on my thesis committee. I would also like to thank Dr. Carlos Christofferson, whose knowledge and patience have been instrumental in my work. His knowledge of Transim and modelling have helped provide me with valuable tips along the way. I would also like to thank my father and mother whose foresight and sacrifice have put me in a position to achieve and excel. Lastly, thanks to Ludwig Van Beethoven for composing the most sublime fifth symphony.

# Contents

**List of Figures** vii

**List of Tables** ix

**1 Introduction** 1

1.1 Motivation . . . 1

1.2 Thesis Overview . . . 3

**2 Literature Review** 4

2.1 Introduction . . . 4

2.2 History of MOS modeling . . . 6

2.2.1 Level 1, 2, 3 . . . 7

2.2.2 BSIM . . . 13

2.3 Transim . . . 16

2.3.1 Introduction . . . 16

2.3.2 Support Libraries . . . 18

2.4 Looking Ahead . . . 23

**3 Implementation of BSIM4** 24

3.1 Introduction . . . 24

3.2 Parameter Table . . . . . 24
3.3 Channel Width and Length . . . . . 26
3.4 Threshold Voltage . . . . . 26
3.4.1 Effective Bulk-Source Voltage . . . . . 33
3.4.2 Effective Gate Source voltage . . . . . 36
3.4.3 Effective $V_{\mathrm{GS}}$ - $V_{\mathrm{TH}}$ Smoothing Function . . . . . 37
3.5 Mobility characteristics . . . . . 38
3.6 Current characteristics . . . . . 39
3.6.1 Effective Internal Source-Drain Resistance . . . . . 39
3.6.2 Bulk-Charge coefficient . . . . . 39
3.6.3 Drain Saturation Voltage . . . . . 39
3.6.4 Effective Drain to Source Voltage . . . . . 40
3.6.5 Effective oxide calculation . . . . . 41
3.7 Current Calculations . . . . . 41
3.7.1 Substrate Currents . . . . . 43
3.7.2 Gate Currents . . . . . 44
3.8 Charge computation and Conservation . . . . . 46
3.8.1 Basic Formulation . . . . . 46
3.8.2 Effective $V_{\mathrm{BS}}$, $V_{GB}$ . . . . . 48
3.8.3 Effective $V_{\mathrm{GS}} - V_{\mathrm{T}}$ . . . . . 48
3.8.4 Modified Bulk Charge coefficient . . . . . 48
3.8.5 The Terminal Charges . . . . . 49

**4 Simulations and Results** **53**

4.1 DC analysis . . . . . 55

4.2 Transient analysis . . . . . . . . . . 58
4.2.1 Sine wave input . . . . . . . . . . 58
4.2.2 Square wave input . . . . . . . . . . 60
4.3 Conclusions . . . . . . . . . . 61

**5 Conclusions and Future Research 66**
5.1 Conclusions . . . . . . . . . . 66
5.2 Future Research . . . . . . . . . . 67

**A MOSFET Model Source code 69**
A.1 C++ code . . . . . . . . . . 69
A.2 Header file . . . . . . . . . . 95

**Bibliography 97**

# List of Figures


2.1 Schematic of `LEVEL` 1, 2 and 3 MOSFET models . . . . . . . . 9

3.1 Schematic of the gate current distribution in the FET . . . . . 45

3.2 Schematic of the dc equivalent circuit . . . . . . . . . . . . . . . 47

4.1 Common Source Amplifier . . . . . . . . . . . . . . . . . . . . . 54

4.2 DC Analysis family of curves . . . . . . . . . . . . . . . . . . . . 55

4.3 DC Analysis comparison at $V_{\mathrm{GS}} = 1.3\mathrm{V}$ . . . . . . . . . . . . . 56

4.4 DC Analysis comparison at $V_{\mathrm{GS}} = 1.4\mathrm{V}$ . . . . . . . . . . . . . 56

4.5 DC Analysis comparison at $V_{\mathrm{GS}} = 1.5\mathrm{V}$ . . . . . . . . . . . . . 57

4.6 DC Analysis comparison at $V_{\mathrm{GS}} = 2.0\mathrm{V}$ . . . . . . . . . . . . . 57

4.7 Transient Analysis comparison at $V_{\mathrm{GS}} = 1.3\mathrm{V}$ . . . . . . . . . . 58

4.8 Transient Analysis comparison at $V_{\mathrm{GS}} = 1.5\mathrm{V}$ . . . . . . . . . . 59

4.9 Transient Analysis comparison at $V_{\mathrm{GS}} = 1.7\mathrm{V}$ . . . . . . . . . . 59

4.10 Square wave input 0.1 GHz . . . . . . . . . . . . . . . . . . . . 60

4.11 Transient Analysis SPICE3 result for a square pulse input at 0.1GHz . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 61

4.12 Transient Analysis Transim result for a square pulse input at 0.1 GHz . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 62

4.13 Transient Analysis SPICE3 result for a square pulse input at 1 GHz . . . . . . . . . . . . . . . . . . . . . . . . . . . . 62
4.14 Transient Analysis Transim result for a square pulse input at 1 GHz . . . . . . . . . . . . . . . . . . . . . . . . . . . . 63
4.15 Transient Analysis Transim result excluding the drain charge derivative . . . . . . . . . . . . . . . . . . . . . . . . . . . 63

# List of Tables


2.1 Parameters for MOS Level 1,2,3 . . . . . . . . . . . . . . . . . . 8

2.2 Parameters for MOS Level 1,2,3 contd . . . . . . . . . . . . . . 9

3.1 MOS Model Parameter table 1 . . . . . . . . . . . . . . . . . . . . 25

3.2 MOS Model Parameter table 2 . . . . . . . . . . . . . . . . . . . . 27

3.3 MOS Model Parameter table 3 . . . . . . . . . . . . . . . . . . . . 28

3.4 MOS Model Parameter table 4 . . . . . . . . . . . . . . . . . . . . 29

3.5 MOS Model Parameter table 5 . . . . . . . . . . . . . . . . . . . . 30

3.6 MOS Model Parameter table 6 . . . . . . . . . . . . . . . . . . . . 31

3.7 MOS Model Parameter table 7 . . . . . . . . . . . . . . . . . . . . 32

3.8 MOS Model Parameter table 8 . . . . . . . . . . . . . . . . . . . . 33

# Chapter 1

# Introduction

## 1.1 Motivation

Semiconductor technology continues to evolve rapidly with an increased demand for performance with a continuous reduction in physical dimensions of semiconductor devices. For high performance MOSFET's the critical dimension continues to reduce and head below $0.1\mu m$ barrier. These trends, however, introduce many subtle mechanisms that govern the properties of sub-micrometer FET's. These mechanisms must be incorporated into models used in modern circuit simulators to enable circuit designers to utilize the potential of modern technology.

The scaling of FET's to the sub-micrometer regime has given rise to various short-channel effects that are related to a weakening of gate control over channel charge. Typical short-channel phenomena include an increase of leakage currents, shifts in threshold voltages and effects such as DIBL.
For models in circuit simulators, it becomes essential to use unified device models, that is, models that are continuous in their functional values and their derivatives through all the regions of operation. This allows avoidance

of discontinuities which are the major problems of obtaining convergence in circuit simulations.

For the modeling of short channel devices and in the formulation of the single equation approach, the sub-threshold region of operation must be included along with the linear and saturation regions. The advantage of the use of a continuous equation for the I-V and C-V characteristics of a device are increased convergence and stability and reduction in computation time in circuit simulations.

Transim is a circuit simulator that has the capability to support many types of elements, like electrical, electromagnetic and thermal. It supports various types of analysis such as dc, transient and harmonic balance. The greatest difference between Transim and other modern commercial simulators is that it uses an object-oriented approach for analyzing circuits. Elements, be it electrical thermal or electromagnetic, can be considered as objects and all these are elements are linked or connected to each other at nodes and by edges, just like classes. Hence the concepts of OO (Object-oriented) programming maps cleanly onto circuit simulation. This makes the process of writing models for Transim relatively straightforward and less cumbersome as it involves the addition of a member to a class, whose properties can be used based on inheritance but is still isolated due to encapsulation of public and private members. This, in effect, simplifies the process of model maintenance and adding new functionality, thus being able to keep abreast with emerging trends in technology.

## 1.2 Thesis Overview

Chapter 2 deals with MOS modeling and gives a brief history about modeling of MOS transistors. It also gives a brief overview about Transim, the circuit simulator for which the MOS model had been written, and some of it's features that simplifies the model writing process.

Chapter 3 deals with the concept of Associated Discrete Modeling and shows by using an example, it's implementation in Transim.

Chapter 4 presents a review about the qualities of a good transistor model and some of the features that are incorporated in the current MOS model for Transim.

Chapter 5 presents some of the results of testing the model with a simple common source configuration. It shows results of a dc analysis and transient analysis at 1GHz and compares the results with Berkeley SPICE.

# Chapter 2

# Literature Review

## 2.1 Introduction

During the 1960's, there was the emergence of certain trends that hastened the process of development of circuit simulation techniques, such as the increase of size and complexity of an electronic circuit which made simulating a circuit on a computer a more attractive option than on a breadboard, along with an increase in computational power. Computer chips were getting smaller and faster, and as the number of people who used them increased, they became less expensive. This made it possible to simulate an electronic circuit with a simulation program that would provide reasonably accurate results in a reasonably small amount of time. Thus, the trend shifted toward the designing, evaluating and re-designing of complex electronic circuits entirely on a computer, using intelligent circuit simulation techniques. **S**imulation **P**rogram with **I**ntegrated-**C**ircuit **E**mphasis or SPICE was a result of this new approach to circuit design.

In SPICE, any circuit is treated in a node/element fashion, i.e. the circuit is visualized as a collection of various elements connected together at nodes.

The entire circuit can be described by the number of nodes and elements that it contains and the orientation of these nodes and elements with each other. If there are $n$ nodes in a circuit, SPICE creates a $n \times n$ matrix and by defining the values of voltages and/or currents at the external nodes, the values of the internal voltages and currents, or state variables, can be solved for each node using matrix manipulation techniques. With routines written for each model, the circuit calls on each model (routine) for evaluation of these state variables.

In addition to SPICE there are many other circuit simulator programs that use the same basic approach. The most commons ones include HSPICE developed by Meta-Software for use with Unix Workstations and PSPICE developed by Microsim for use with PC's. These and a few other popular simulators use element models and/or different mathematical models for solving the set of linear and non-linear equations, but use the same general technique for solving circuits. Because of the easy availability of SPICE which coincided with a growth in the IC industry, it became the industry standard for circuit simulation.

This chapter attempts to give a brief history about MOS modeling and how the various models have evolved up to their current state. It also talks about the path that the science of MOS modeling will take in the near future.

This chapter also presents a brief overview about Trasim, the circuit simulator and some of it's features that enable the ease of model writing and model maintenance.

## 2.2 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.

### 2.2.1 Level 1, 2, 3

The parameter list for the level 1, 2 and 3 MOS models is shown in the Table 2.1 and Table 2.2. A generalized equivalent circuit is shown in Figure 2.1

**"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 in-

| Parameter | Description | Units |
|---|---|---|
| AF | flicker noise exponent | - |
| CBD | zero-bias B-D junction capacitance | F |
| CBS | zero-bias B-S junction capacitance | F |
| CGBO | gate-bulk o/v cap/m of channel length | F/m |
| CGDO | gate-drain o/v cap/m of channel width | F/m |
| CGSO | 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 | $V^{\frac{1}{2}}$ |
| 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 | - |
| KP | transconductance parameter | $A/V^2$ |
| LAMBDA | channel-length modulation (LEVEL=1, 2 only) | 1/V |
| 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 |

Table 2.1: Parameters for MOS Level 1,2,3

| Parameter | Description | Units |
|---|---|---|
| `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) | - |
| `UO` | surface mobility (U-oh) | $cm^2/V - s$ |
| `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 |

Table 2.2: Parameters for MOS Level 1,2,3 contd

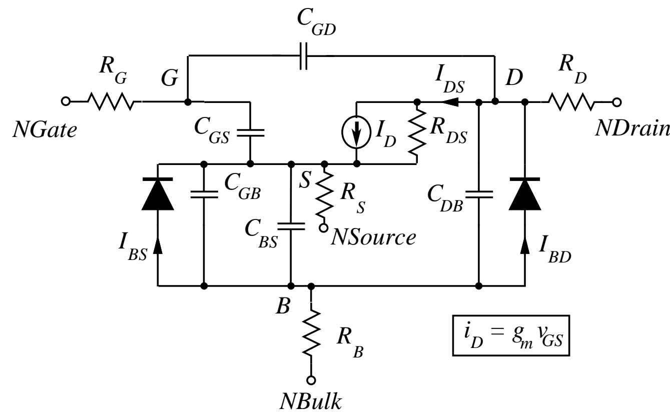


Figure 2.1: Schematic of `LEVEL` 1, 2 and 3 MOSFET models. $V_{GS}$, $V_{DS}$, $V_{GD}$, $V_{GB}$, $V_{DB}$ and $V_{BS}$ are voltages between the internal gate, drain, bulk and source terminals designated $G$, $D$, $B$ and $C$ respectively.

cluded. It is applicable to fairly large devices with gate lengths greater than 4 $\mu$m. Its main attribute is that only a few parameters need be specified and so it is good for preliminary analyses.

**Drain Current Model**

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:

$$I_{DS,Lin} = \frac{\mu W_{EFF} C_{OX}}{L_{EFF}}[(V_{GS} - V_T)V_{DS} - \frac{V_{DS}^2}{2}](1 + \lambda V_{DS}) \qquad (2.1)$$

and in the saturation region :

$$I_{DS,Sat} = \frac{\mu W_{EFF} C_{OX}}{2L_{EFF}}(V_{GS} - V_T)^2(1 + \lambda V_{DS}) \qquad (2.2)$$

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

**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

$$Q_{DEPL} = W_{EFF} L_{EFF} C_{OX} \gamma \sqrt{2\phi_f - V_{BS}} \quad (2.3)$$

The inversion charge $Q_{INV}$ is given by

$$Q_{INV}(y) = -C_{OX}(V_{GS} - V_T - V(y)) \quad (2.4)$$

The charge at the gate terminal, $Q_{GATE}$ can be found [1] using equations 2.3 and 2.4.

For the Linear Region:

$$Q_{GATE} = \frac{2}{3} W_{EFF} L_{EFF} C_{OX} [\frac{(V_{GD} - V_T)^3 - ((V_{GS} - V_T)^3}{(V_{GD} - V_T)^2 - (V_{GS} - V_T)^2}] - Q_{DEPL} \quad (2.5)$$

where $Q_{DEPL}$ is given by equation 2.3

For the saturation region:

$$Q_{GATE} = [\frac{2}{3} W_{EFF} L_{EFF} C_{OX} (V_{GS} - V_T)] - Q_{DEPL} \quad (2.6)$$

where $Q_{DEPL}$ is given by Equation 2.3

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. Detailed analysis can be found in [3], [4].

**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 $\mu$m 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. Detailed analysis can be found in [1]

**MOS3**

This is a semi-empirical model developed in 1980. It is also used for gate lengths of 2 $\mu$m 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 it's 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 appli-

cations. For more details refer to [1].
The parameters used in the Level 1, 2 and 3 models are given in table 2.1. 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 ($C_{GD}$, $C_{GS}$ and $C_{GB}$) 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.

### 2.2.2 BSIM

#### BSIM

The **B**erkeley **S**hort-Channel **I**GFET **M**odel 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.

**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.

**BSIM3**

This model eliminates the discontinuity in the derivatives if 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. The latest version at the time of this writing is BSIM3v3. This model has found to have produced accurate results at $0.18\mu m$ technologies. Information, new releases and code can be found at http://www-device.eecs.berkeley.edu/ bsim3/.

**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 sub-micron technologies and shows good stability during circuit simulation. More can be found out about this model at http://www-us2.semiconductors.philips .com/Philips_Models/introduction/mosmodel9/

**EKV**

This model was first proposed by Christian Enz, Franois Krummenacher and Eric Vittoz [12]. 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 http://legwww.epfl.ch/ekv/ model.html.

**BSIM4**

This model was first made public in the year 2000 and offers several improvements over BSIM3 in terms of I-V modeling, noise modeling and parasitics. Some features about BSIM4 can be found in Chapter 4 along with the equations used in Transim. The C++ code used will be found in Appendix A.

## 2.3 Transim

### 2.3.1 Introduction

The rapid rate of innovation of microwave and millimeter wave systems requires the development of an easily extensible and modifiable computer aided engineering (CAE) environment. While great strides have been made in the flexibility of commercial CAE tools, these sometimes prove inadequate in modeling advanced systems. As with virtually all aspects of electronic engineering the abstraction level of RF and microwave theory and techniques has increased dramatically. In particular, large systems are being designed with attention given to the interaction of components at many levels. One of the most significant developments relevant to computer aided engineering is the rise of object oriented (OO) design practice. While it is normal to think of OO-specific programming languages as being the main technology for implementing OO design, good OO practice can be implemented in more conventional programming languages such as C. However OO-specific languages foster code reuse and have constructs that facilitate object manipulation. The OO abstraction is well suited to modeling electronic systems, for example, circuit elements are already viewed as discrete objects and at the same time as an integral part of a (circuit) continuum. The OO view is a unifying concept that maps extremely well onto the way humans perceive the world around them.

Non-OO circuit simulators always become complicated with many layers of special cases. Referring to circuit elements again, traditional simulation implementations have many if-then like statements and individually identify

every element in many places for special handling. An integral part of the various high performance computing initiatives is the separation of the core components embodying numerical methods from the modeling and solver formulation process with the result that numerical techniques developed by computer scientists and mathematicians can be formulated using formal correctness procedures. Thus, what is adopted, is that the circuit abstraction is adapted so that highly reliable and efficient pre-developed libraries can be used. C++ was once considered slow for scientific applications. Advances in compilers and programming techniques, however, have made this language attractive and in some benchmarks C++ outperforms Fortran. Several OO numerical libraries have been developed . Of great importance to the work described here is the incorporation of the standard template library (STL). The STL is a C++ library of container classes, algorithms, and iterators; it provides many of the basic algorithms and data structures of computer science. The STL is a generic library, meaning that its components are heavily parameterized: almost every component in the STL is a template. The current ISO/ANSI C++ standard has not been fully implemented and C++ compilers support a variable subset of the standard. The biggest areas of noncompliance being the templates and the standard library. The goal in design was to obtain speed in development, to use off the shelf advanced numerical techniques, and to allow easy expansion and testing of new models and numerical methods. The circuit simulator implementing these ideas is Transim. Transim is the first circuit simulator to use recent OO techniques. The design intent was to to combine the advantages of previous OO circuit simulators with these new developments as well as expanding capability.

Transim uses C++ libraries and some written in C or Fortran.

### 2.3.2 Support Libraries

#### Solution of Sparse Linear Systems

Sparse 1.3 2 is a flexible package of subroutines written in C used to numerically solve large sparse systems of linear equations. The package is able to handle arbitrary real and complex square matrix equations. Besides being able to solve linear systems, it is also able to quickly solve transposed systems, find determinants, and estimate errors due to ill-conditioning in the system of equations and instability in the computations. Sparse also provides a test program that is able to read matrix equation from a file, solve it, and print useful information (such as condition number of the matrix) about the equation and its solution. Sparse was originally written for use in circuit simulators and is well adapted to handling nodal- and modified-nodal admittance matrices.

SuperLU 3 is used in the wavelet and time marching transient analyses. It contains a set of subroutines to numerically solve a sparse linear system Ax = b. It uses Gaussian elimination with partial pivoting (GEPP). The columns of A may be pre-ordered before factorization; the pre-ordering for sparsity is completely separate from the factorization. SuperLU is implemented in ANSI C. It provides support for both real and complex matrices, in both single and double precision.

#### Vectors and Matrices

Most of the vector and matrix handling in Transim uses MV++4. This is a small set of vector and simple matrix classes for numerical computing written in C++. It is not intended as a general vector container class but rather designed specifically for optimized numerical computations on RISC and pipelined architectures which are used in most new computer architectures. The various MV++ classes form the building blocks of larger user-level libraries. The MV++ package includes interfaces to the computational kernels of the Basic Linear Algebra Subprograms package (BLAS) which includes scalar updates, vector sums, and dot products. The idea is to utilize vendor-supplied, or optimized BLAS routines that are fine-tuned for particular platforms.

The Matrix template Library (MTL)5 is a high-performance generic component library that provides comprehensive linear algebra functionality for a wide variety of matrix formats. It is used in the wavelet and time marching transient analyses. As with the STL, MTL uses a five-fold approach, consisting of generic functions, containers, iterators, adaptors, and function objects, all developed specifically for high performance numerical linear algebra. Within this framework, MTL provides generic algorithms corresponding to the mathematical operations that define linear algebra. Similarly, the containers, adaptors, and iterators are used to represent and to manipulate matrices and vectors.

**Solution of Non-Linear systems**

Nonlinear systems of equations in Transim are solved using the NNES6 library. This package is written in Fortran and provides Newton and quasi-

Newton methods with many options including the use of analytic Jacobian or forward, backwards or central differences to approximate it, different quasi-Newton Jacobian updates, or two globally convergent methods, etc. This library is used through an interface class (NLSInterface), so it is possible to install a different routine to solve nonlinear systems if desired by just replacing the interface (four different nonlinear solvers have already been used). The Fortran routine NLEQ1 (Numerical solution of nonlinear (NL) equations (EQ)7) can also be used as a compile option.

**Fourier Transform**

Fourier transformation is implemented in Transim using the FFTW8 library. FFTW is a C subroutine library for computing the Discrete Fourier Transform (DFT) in one or more dimensions, of both real and complex data, and of arbitrary input size. Benchmarks, performed on a variety of platforms show that FFTW's performance is typically superior to that of other publicly available FFT software. Moreover, FFTW's performance is portable: the program performs well on most computer architectures without modification.

**Automatic Differentiation**

Most nonlinear computations require the evaluation of first and higher derivatives of vector functions with m components in n real or complex variables. Often these functions are defined by sequential evaluation procedures involving many intermediate variables. By eliminating the intermediate variables symbolically, it is theoretically always possible to express the m dependent

variables directly in terms of the n independent variables. Typically, however, the attempt results in unwieldy algebraic formulae, if it can be completed at all. Symbolic differentiation of the resulting formulae will usually exacerbate this problem of expression swell and often entails the repeated evaluation of common expressions.

An obvious way to avoid such redundant calculations is to apply an optimizing compiler to the source code that can be generated from the symbolic representation of the derivatives in question. Given a code for a function $F : \Re^n \rightarrow \Re^m$, automatic differentiation (AD) uses the chain rule successively to compute the derivative matrix. AD has two basic modes, forward mode and reverse mode. The difference between these two is the way the chain rule is used to propagate the derivatives.

A versatile implementation of the AD technique is Adol-C 9, a software package written in C and C++. The numerical values of derivative vectors (required to fill a Jacobian for solving non-linear elements using Newton's method) are obtained free of truncation errors at a small multiple of the run time required to evaluate the original function with little additional memory required. It is important to note that AD is not numerical differentiation and the same accuracy achieved by evaluating analytically developed derivatives is obtained. The eval() method of the nonlinear element class is executed at initialization time and so the operations to calculate the currents and voltages of each element are recorded by Adol-C in a tape which is actually an internal buffer. After that, each time that the values or the derivatives of the nonlinear elements are required, an Adol-C function is called and the values are calculated using the tapes. This implementation is efficient because the

taping process is done only once (this almost doubles the speed of the calculation compared to the case where the functions are taped each time they are needed). When the Jacobian is needed, the corresponding Adol-C function is called using the same tape. In the case of Harmonic Balance simulations, the program has been tested with large circuits with many tones, and the function or Jacobian evaluation times are always very small compared with the time required to solve the matrix equation (typically some form of Newton's method) that uses the Jacobian. The conclusion is that there is little detriment to the performance of the program introduced by using automatic differentiation. However the advantage in terms of rapid model development is significant. The majority of the development time in implementing models in simulators, is in the manual development of the derivative equations. Unfortunately the determination of derivatives using numerical differences is not sufficiently accurate for any but the simplest circuits and in any event, is computationally intensive. With Adol-C full 'analytic' accuracy is obtained and the **implementation of nonlinear device models is dramatically simplified**. From experience the average time to develop and implement a transistor model is an order of magnitude less than deriving and coding the derivatives manually. Note that time differentiation, time delay and transformations are left outside the automatic differentiation block. The calculation speed achieved is approximately ten times faster than the speed achieved by including time differentiation, time delay and transformations inside the block.

## 2.4 Looking Ahead

At present, the third generation models that are being developed are changing constantly and evolving even more. There is also a tremendous growth in analog and mixed-signal circuits as parts of digital IC's. The direction that third generation models are taking now are to use smoothing functions to ensure mathematical stability near the transition points in the operation of a device. This means the description of a device with a single equation for current and charge (capacitance) across all regions of operation. This approach is superior than having region based equations with separate curves pieced together at the transition points. Hence, an increase in stability by using a single continuous equation across all regions of operation and accounting for various effects and mechanisms that are now becoming important as device dimensions are shrinking, is being sought.

# Chapter 3

# Implementation of BSIM4

## 3.1 Introduction

The BSIM4 model takes a lot of it's characteristics from it's predecessor, BSIM3 but also adds enough functionality to name it with a new model number. It uses many new parameters and replaces some old BSIM3 parameters. It uses a newly formulated smoothing function for gate-source voltage. It uses more than 200 parameters, uses charge conserving equations for calculation of various capacitances, has a single equation for modeling current in all regions of transistor operation, has better modeling for gate currents, for external parasitics, noise, temperature and mobility. This chapter deals with the equations and features of this advanced transistor model that have been modeled in Transim.

## 3.2 Parameter Table

The parameters used in the model are listed in the tables 3.1 to 3.8.

| Parameter | Description | Default | Units |
|---|---|---|---|
| TOXE | Electrical gate equivalent oxide thickness | 3.0e-9 | $m$ |
| TOXP | Physical gate equivalent oxide thickness | TOXE | $m$ |
| EPSROX | Gate dielectric constant relative to vacuum | 3.9 | - |
| VFB | Flat-band voltage | -1.0 | $V$ |
| VTH0 | Long-channel threshold voltage | 0.7 | $V$ |
| NGATE | Poly Si gate doping concentration | 0.0 | $cm^{-3}$ |
| XL | Channel length offset due to mask/etch effect | 0.0 | $m$ |
| XW | Channel width offset due to mask/etch effect | 0.0 | $m$ |
| NF | Number of device fingers | 1.0 | - |
| W | Width of the device | 5.0e-6 | $m$ |
| L | Length of the device | 5.0e-6 | $m$ |
| DWG | Coefficient of gate bias dependence of Weff | 0.0 | $m/V$ |
| DWB | Coefficient of body bias dependence of Weff | 0.0 | $m/V$ |
| WINT | Channel-width offset parameter | 0.0 | $m$ |
| WLN | Power of length dependence of width offset | 1.0 | $m$ |
| WL | Coefficient of length dependence for width offset | 0.0 | $m$ |
| WWN | Power of width dependence of width offset | 1.0 | $m$ |
| WW | Coefficient of width dependence for width offset | 0.0 | $m$ |
| WWL | Coefficient of length and width cross term dependence for width offset | | $m$ |
| LINT | Channel-length offset parameter | 0.0 | $m$ |
| LLN | Power of length dependence for length offset | 0.0 | $m$ |

Table 3.1: MOS Model Parameter table 1

## 3.3 Channel Width and Length

The effective channel lengths and widths are less than the values of `L` and `W` on account of diffusion effects. `XL` and `XW` are parameters that account for the channel length/width offset due to mask/etch effects and process nonuniformity. The terms `dL` and `dW` are provided for user convenience. They are turned off by default. The effective length $L_{\mathrm{EFF}}$ is represented as

$$L_{\mathrm{EFF}} = \mathtt{L} + 2\mathtt{XL} - 2\Delta L_{\mathrm{geom}} \tag{3.1}$$

where

$$\Delta L_{\mathrm{geom}} = \frac{\mathtt{LL}}{L^{\mathtt{LLN}}} + \frac{\mathtt{LW}}{W^{\mathtt{LWN}}} + \frac{\mathtt{LWL}}{L^{\mathtt{LLN}} W^{\mathtt{LWN}}} \tag{3.2}$$

The effective width is represented as

$$W_{\mathrm{EFF}} = \frac{W}{\mathtt{NF}} + \mathtt{XW} - 2\Delta W_{\mathrm{geom}} - 2\Delta W_{\mathrm{biasdep}} \tag{3.3}$$

where

$$\Delta W_{\mathrm{geom}} = \frac{\mathtt{WL}}{L^{\mathtt{WLN}}} + \frac{\mathtt{WW}}{W^{\mathtt{WWN}}} + \frac{\mathtt{WWL}}{L^{\mathtt{WLN}} W^{\mathtt{WWN}}} \tag{3.4}$$

$$\Delta W_{\mathrm{biasdep}} = \mathtt{DWG}(V_{\mathrm{GSTeff}}) + \mathtt{DWB}(\sqrt{2\phi_f - V_{\mathrm{BSeff}}} - \sqrt{2\phi_f}) \tag{3.5}$$

## 3.4 Threshold Voltage

This model attempts to accurately model threshold voltage and include various channel effects such as DIBL (Drain Induced Barrier Lowering), Non-uniform vertical doping, body-effect, charge sharing between the source and drain, short-channel and pocket implant effects.

| Parameter | Description | Default | Units |
|---|---|---|---|
| LL | Coefficient of length dependence for length offset | 0.0 | $m$ |
| LW | Coefficient of width dependence for length offset | 0.0 | $m$ |
| LWN | Power of width dependence for length offset | 1.0 | $m$ |
| LWL | Coefficient of length and width cross term dependence for length offset | 0.0 | $m$ |
| K1 | First-order body bias coefficient | 0.0 | $V^{-0.5}$ |
| K2 | Second-order body bias coefficient | 0.0 | - |
| LPEB | Lateral non-uniform doping effect on K1 | 0.0 | $m$ |
| LPE0 | Lateral non-uniform doping parameter at Vbs=0 | 1.74e-7 | $m$ |
| K3 | Narrow width coefficient | 80.0 | - |
| K3B | Body effect coefficient of K3 | 0.0 | $V^{-1}$ |
| W0 | Narrow width parameter | 2.5e-6 | $m$ |
| DVT0W | First coefficient of narrow width effect on threshold voltage for small channel length | 0.0 | - |
| DVT0 | First coefficient of short channel effect on threshold | 2.2 | - |
| DVT1W | Second coefficient of narrow width effect on threshold voltage for small channel length | 5.3e6 | - |
| DVT1 | Second coefficient of short channel effect on threshold | 0.53 | - |
| DSUB | DIBL coefficient exponent in sub-threshold region | 0.56 | - |
| ETA0 | DIBL coefficient in sub-threshold region | 0.56 | - |
| ETAB | Body-bias coefficient for the sub-threshold region | -0.07 | - |
| TOXM | $T_{\text{ox}}$ at which parameters are extracted | TOXE | $m$ |

Table 3.2: MOS Model Parameter table 2

| Parameter | Description | Default | Units |
|---|---|---|---|
| T | Temperature | 300.0 | $^oK$ |
| NDEP | Channel doping concentration at depletion edge for zero body bias | 1.7e17 | $cm^{-3}$ |
| PHIN | Non-uniform vertical doping effect on surface potential | 0.0 | $V$ |
| VBM | Maximum applied body bias in VTH0 calculation | -3.0 | $V$ |
| NSUB | Substrate doping concentration | 6.0e16 | $cm^{-3}$ |
| DVT2W | Body-bias coefficient of narrow width effect for small channel length | -0.032 | - |
| NSD | Source/drain doping concentration | 1.0e20 | $cm^{-3}$ |
| DVT2 | Body-bias coefficient of short-channel effect on threshold | -0.032 | - |
| MINV | $V_{gsteff}$ fitting parameter for moderate inversion condition | -0.0 | - |
| NFACTOR | Subthreshold swing factor | 1.0 | - |
| CDSC | Coupling capacitance between source/drain and channel | 1.0 | $F/m^2$ |
| CDSCD | Drain-bias sensitivity of CDSC | 2.4e-4 | $F/Vm^2$ |
| CDSCB | Body-bias sensitivity of CDSC | 0.0 | $F/Vm^2$ |
| CIT | Interface trap capacitance | 0.0 | $F/m^2$ |
| KETA | Body-bias coefficient of bulk charge effect | -0.047 | $V^{-1}$ |
| B0 | Bulk charge effect coefficient for channel width | 0.0 | $m$ |
| B1 | Bulk charge effect width offset | 0.0 | $m$ |
| A0 | Coefficient of channel-length dependence bulk charge effect | 1.0 | - |
| AGS | Coefficient of $V_{\mathrm{gs}}$ dependence of bulk charge effect | 0.0 | $V^{-1}$ |
| XJ | S/D junction depth | 1.5e-7 | $m$ |

Table 3.3: MOS Model Parameter table 3

| Parameter | Description | Default | Units |
|---|---|---|---|
| `U0` | Low-field mobility | 0.067 | $m^2/Vs$ |
| `UA` | Coefficient of first-order mobility degradation due to vertical field | 1.0e-15 | $m/V$ |
| `UB` | Coefficient of second-order mobility degradation due to vertical field | 1.0e-19 | $m^2/V^2$ |
| `UC` | Coefficient of mobility degradation due to body-bias effect | -0.0465e-9 | $m/V$ |
| `EU` | Exponent for mobility degradation | 1.67 | - |
| `DELTA` | Parameter for DC $V_{dseff}$ | 0.01 | $V$ |
| `PDITS` | Impact of drain-induced threshold shift on $R_{out}$ | 0.0 | $V^{-1}$ |
| `FPROUT` | Effect of pocket implant on Rout degradation | 0.0 | $V/m^0.5$ |
| `PDITSL` | Channel-length dependence of drain-induced $V_{th}$ shift for $R_{out}$ | 0.0 | $m^{-1}$ |
| `PDITSD` | $V_{ds}$ dependence of drain-induced $V_{th}$ shift for Rout | 0.0 | $V^{-1}$ |
| `PSCBE2` | Second substrate current induced body-effect parameter | 1.0e-5 | $m/V$ |
| `PSCBE1` | First substrate current induced body-effect parameter | 4.24e8 | $V/m$ |
| `PDIBLCB` | Body bias coefficient of DIBL effect on Rout | 0.0 | $V^{-1}$ |
| `PVAG` | Gate-bias dependence of Early voltage | 0.0 | - |
| `PDIBL1` | Parameter for DIBL effect on $R_{out}$ | 0.0 | - |
| `PDIBL2` | Parameter for DIBL effect on $R_{out}$ | 0.0 | - |
| `AGS` | Coefficient of $V_{gs}$ dependence of bulk charge effect | 0.0 | $V^{-1}$ |

Table 3.4: MOS Model Parameter table 4

| **Parameter** | **Description** | **Default** | **Units** |
|---|---|---|---|
| XJ | S/D junction depth | 1.5e-7 | $m$ |
| DROUT | Channel-length dependence of DIBL effect on $R_{out}$ | 0.56 | - |
| PCLM | Channel length modulation parameter | 1.3 | - |
| A1 | First non-saturation effect parameter | 0.0 | $V^{-1}$ |
| A2 | Second non-saturation factor | 1.0 | - |
| RDWMIN | Lightly-doped drain resistance per unit width at high $V_{\rm gs}$ and zero $V_{bs}$ | 0.0 | Ω |
| RDSW | Zero bias lightly-doped drain resistance per unit width | 200.0 | Ω |
| PRWG | Gate-bias dependence of LDD resistance | 1.0 | $V^{-1}$ |
| PRWB | Body-bias dependence of LDD resistance | 0.0 | $V^{-0.5}$ |
| WR | Channel-width dependence parameter of LDD resistance | 1.0 | $m$ |
| WLC | Coefficient of length dependence for CV channel width offset | WL | $m$ |
| WWC | Coefficient of width dependence for CV channel width offset | WW | $m$ |
| WWLC | Coefficient of length and width cross term dependence for CV channel width offset | WWL | $m$ |
| DWJ | Offset of the S/D junction width | WINT | $m$ |
| CLC | Constant term for the short channel model | 1.0e-7 | $m$ |
| CLE | Exponential term for the short channel model | 0.6 | - |
| NOFF | CV parameter in $V_{gsteffCV}$ for weak to strong inversion | 1.0 | - |
| VOFFCV | CV parameter in $V_{gsteffCV}$ for weak to strong inversion | 0.0 | $V$ |

Table 3.5: MOS Model Parameter table 5

| Parameter | Description | Default | Units |
|---|---|---|---|
| CF | Fringing field capacitance | 0.0 | $F/m$ |
| CKAPPAD | Coefficient of bias-dependent overlap capacitance for the drain side | 0.6 | $V$ |
| CKAPPAS | Coefficient of bias-dependent overlap capacitance for the source side | 0.6 | $V$ |
| LLC | Coefficient of length dependence on CV channel length offset | 0.0 | $m$ |
| LWC | Coefficient of width dependence on CV channel length offset | 0.0 | $m$ |
| LWLC | Coefficient of length and width cross term dependence on CV channel length offset | 0.0 | $m$ |
| WWLC | Coefficient of length and width cross term dependence on CV channel width offset | 0.0 | $m$ |
| VOFF | Offset voltage in the subthreshold region for large W and L | -0.08 | $V$ |
| VOFFL | Channel length dependence of VOFF | 0.0 | $V$ |
| POXEDGE | Factor for the gate oxide thickness in S/D overlap regions | 1.0 | - |
| TOXREF | Nominal gate oxide thickness for gate dielectric tunnelling current model | 3.0e-9 | $m$ |
| NTOX | Exponent for gate oxide ratio | 1.0 | - |
| DLCIG | Source/drain overlap length for $I_{\mathrm{gs}}$ and $I_{\mathrm{gd}}$ | LINT | $m$ |
| AIGSD | parameter for $I_{\mathrm{gs}}$ and $I_{\mathrm{gd}}$ | 0.43 | $(Fs^2/g)^{0.5}m^{-1}$ |
| BIGSD | parameter for $I_{\mathrm{gs}}$ and $I_{\mathrm{gd}}$ | 0.054 | $(Fs^2/g)^{0.5}m^{-1}$ |
| CIGSD | parameter for $I_{\mathrm{gs}}$ and $I_{\mathrm{gd}}$ | 0.075 | $(Fs^2/g)^{0.5}m^{-1}$ |
| MOIN | Coefficient for gate-bias dependent surface potential | 15.0 | - |

Table 3.6: MOS Model Parameter table 6

| Parameter | Description | Default | Units |
|---|---|---|---|
| VSAT | Saturation velocity | 8.0e4 | $m/s$ |
| PDITSD | $V_{ds}$ dependence of drain induced $V_{th}$ shift for Rout | 0.0 | $V^{-1}$ |
| AIGC | Parameter for $I_{\mathrm{gcs}}$ and $I_{\mathrm{gcd}}$ | 0.43 | $(Fs^2/g)^{0.5}m^{-1}$ |
| BIGC | Parameter for $I_{\mathrm{gcs}}$ and $I_{\mathrm{gcd}}$ | 0.054 | $(Fs^2/g)^{0.5}m^{-1}$ |
| CIGC | Parameter for $I_{\mathrm{gcs}}$ and $I_{\mathrm{gcd}}$ | 0.075 | $(Fs^2/g)^{0.5}m^{-1}$ |
| NIGC | Parameter for $I_{\mathrm{gcs}}$ $I_{\mathrm{gcd}}$, $I_{\mathrm{gs}}$ and $I_{\mathrm{gd}}$ | 1.0 | $(Fs^2/g)^{0.5}m^{-1}$ |
| PIGCD | $V_{ds}$ dependence of $I_{\mathrm{gcs}}$ and $I_{\mathrm{gcd}}$ | 1.0 | - |
| DVTP0 | First coefficient of drain induced $V_{th}$ shift due to long channel pocket devices | 0.0 | $m$ |
| DVTP1 | First coefficient of drain induced $V_{th}$ shift due to long channel pocket devices | 0.0 | $V^{-1}$ |
| PRT | Temperature coefficient for RDSW | 0.0 | $\Omega - m$ |
| AT | Temperature coefficient for saturation velocity | 3.3e-4 | $m/s$ |
| XT | Doping Depth | 1.55e-7 | $m$ |
| ALPHA0 | First parameter of impact ionization current | 0.0 | $Am/V$ |
| ALPHA1 | $I_{\mathrm{sub}}$ parameter for length scaling | 0.0 | $A/V$ |
| BETA0 | Second parameter of impact ionization current | 30.0 | $V$ |
| AGIDL | Pre-exponential coefficient for GIDL | 0.0 | $A/V$ |
| BGIDL | Exponential coefficient for GIDL | 2.3e9 | $V/m$ |
| CGIDL | Parameter for body-bias effect on GIDL | 0.5 | $V^3$ |
| EGIDL | Fitting parameter for band bending for GIDL | 0.8 | $V$ |
| ACDE | Exponential coefficient for charge thickness | 1.0 | $m/V$ |
| DLC | Channel length offset parameter | LINT | $m$ |
| DWC | Channel width offset parameter | WINT | $m$ |

Table 3.7: MOS Model Parameter table 7

| Parameter | Description | Default | Units |
|---|---|---|---|
| AIGBACC | Parameter for $I_{gb}$ in accumulation | 0.43 | $m^{-1}$ |
| BIGBACC | Parameter for $I_{gb}$ in accumulation | 0.054 | $m^{-1}V^{-1}$ |
| CIGBACC | Parameter for $I_{gb}$ in accumulation | 0.075 | $V^{-1}$ |
| NIGBACC | Parameter for $I_{gb}$ in accumulation | 1.0 | - |
| AIGBINV | Parameter for $I_{gb}$ in inversion | 0.35 | $m^{-1}$ |
| BIGBINV | Parameter for $I_{gb}$ in inversion | 0.03 | $m^{-1}V^{-1}$ |
| CIGBINV | Parameter for $I_{gb}$ in inversion | 0.006 | $V^{-1}$ |
| EIGBINV | Parameter for $I_{gb}$ in inversion | 1.1 | $V$ |
| NIGBINV | Parameter for $I_{gb}$ in inversion | 3.0 | - |
| KT1 | Temperature coeff for $V_{\mathrm{TH}}$ | -0.11 | $V$ |
| KT1L | Channel length for KT1 | 0.0 | $Vm$ |
| KT2 | Body bias coeff for $V_{\mathrm{TH}}$ temp effect | 0.022 | - |

Table 3.8: MOS Model Parameter table 8

### 3.4.1 Effective Bulk-Source Voltage

$V_{\mathrm{BSeff}}$ is calculated in order to prevent the body bias from taking unreasonably high values during simulation. It provides an upper limit on the value of body bias.

$$V_{\mathrm{BSeff}} = V_{\mathrm{bc}} + \frac{(V_{\mathrm{BS}} - V_{\mathrm{bc}} - 0.001) + \sqrt{(V_{\mathrm{BS}} - V_{\mathrm{bc}} - 0.001)^2 - 4\, V_{\mathrm{bc}}\, 0.001}}{2} \tag{3.6}$$

where $V_{\mathrm{bc}}$, which represents the maximum allowable $V_{\mathrm{BS}}$ is given by

$$V_{\mathrm{bc}} = 0.9.(2\phi_f - \frac{\mathtt{K1}^2}{4\mathtt{K2}^2}) \tag{3.7}$$

The threshold voltage is evaluated as

$$\begin{aligned} V_{\mathrm{TH}} &= \mathtt{VTH0} + \delta_{\mathrm{NP}}.(\Delta V_{\mathrm{T,BodyEffect}} - \Delta V_{\mathrm{T,ChargeSharing}} - \Delta V_{\mathrm{T,DIBL}} \\ &\quad + \Delta V_{\mathrm{T,ReverseShortChannel}} + \Delta V_{\mathrm{T,NarrowWidth}} + \Delta V_{\mathrm{T,SmallSize}} \\ &\quad - \Delta V_{\mathrm{T,PocketImplant}}) \end{aligned} \tag{3.8}$$

In certain cases, devices are operated with a positive value of $V_{\mathrm{BS}}$. In these cases, the threshold voltage reduces and drive current increases. The parameters K1 and K2 control the value of the body effect term and it is modeled by

$$\begin{aligned}\Delta V_{\mathrm{T,BodyEffect}} &= [\mathtt{K1}\,\frac{\mathtt{TOXE}}{\mathtt{TOXM}}\,\sqrt{2\phi_f - V_{\mathrm{BSeff}}} - \mathtt{K1}\sqrt{2\phi_f}]\,\sqrt{1+\frac{\mathtt{LPEB}}{L_{\mathrm{EFF}}}} \\ &\quad -\mathtt{K2}\,V_{\mathrm{BSeff}}\,\frac{\mathtt{TOXE}}{\mathtt{TOXM}} \end{aligned} \tag{3.9}$$

In modern technologies, the threshold voltage first increases as the effective length decreases before it takes on it's expected trend of decrease as effective length decreases. To correctly model the temporary increase of threshold voltage the term used is

$$\Delta V_{\mathrm{T,ReverseShortChannel}} = \mathtt{K1}\,\frac{\mathtt{TOXE}}{\mathtt{TOXM}}\,(\sqrt{1+\frac{\mathtt{LPE0}}{L_{\mathrm{EFF}}}}-1)\,\sqrt{2\phi_f} \tag{3.10}$$

As the channel becomes shorter, the threshold voltage becomes more dependent on the channel length (SCE, short channel effects) and on DIBL. As the product of effective length and width reduces, the exponents in Equation 3.11 reduce and assume a finite value. This suggests that there is a shift in threshold voltage for smaller devices. The value can be controlled by the parameters DVT0W and DVT1W. SCE are represented as

$$\begin{aligned}\Delta V_{\mathrm{T,SmallSize}} &= \mathtt{DVT0W}\,[\exp(-\mathtt{DVT1W}\,\frac{W_{\mathrm{EFF}}\,L_{\mathrm{EFF}}}{2L_{\mathrm{tw}}}) \\ &\quad +2\,\exp(-\mathtt{DVT1W}\,\frac{W_{\mathrm{EFF}}\,L_{\mathrm{EFF}}}{2L_{\mathrm{tw}}})]\,(V_{\mathrm{bi}}-2\phi_f)\end{aligned} \tag{3.11}$$

As $V_{\mathrm{DS}}$ increases in short channel devices, there is a non-trivial change in the surface potential. As a result, the barrier blocking the carriers in the drain

from entering the channel diminishes and the device turns on sooner. Since this barrier lowering is induced by drain source voltage, this effect is called Drain Induced Barrier Lowering. To model DIBL, the following term is used.

$$\begin{aligned}\Delta V_{\mathrm{T,DIBL}} &= [\exp(-\texttt{DSUB}\,\frac{L_{\mathrm{EFF}}}{2L_{\mathrm{t0}}}) + 2\exp(-\texttt{DSUB}\,\frac{L_{\mathrm{EFF}}}{2L_{\mathrm{t0}}})] \\ &\quad \times(\texttt{ETA0} + \texttt{ETAB}\,V_{\mathrm{BSeff}})\,V_{\mathrm{DS}} \end{aligned} \tag{3.12}$$

The actual depletion region in the channel is larger than what is usually assumed because of fringing fields. Thus, as the channel width decreases, there is a net increase in the threshold voltage. This is modeled by

$$\Delta V_{\mathrm{T,NarrowWidth}} = (\texttt{K3} + \texttt{K3B}\,V_{\mathrm{BSeff}})\,\frac{\texttt{TOXE}}{W_{\mathrm{EFF}} + \texttt{W0}} \tag{3.13}$$

The influence of charge sharing effects between the source and drain depends greatly on the size of the channel. It's value increases as the channel lengths reduce. The effect of charge sharing on threshold voltage is controlled by parameters `DVT0`, `DVT1` and `DVT2`. When the effective channel length is small, the exponents assume a finite value and have a direct bearing on the value of threshold voltage. Increased charge sharing tends to reduce the value of threshold voltage and it is represented as

$$\Delta V_{\mathrm{T,ChargeSharing}} = \texttt{DVT0}\frac{0.5}{\cosh(\texttt{DVT1}\;L_{\mathrm{EFF}}/L_t) - 1}(V_{\mathrm{bi}} - 2\phi_f) \tag{3.14}$$

$\Delta V_{\mathrm{T,PocketImplant}}$ is defined after the calculation of ideality factor $n$. The built in potential is given as

$$V_{\mathrm{bi}} = \frac{k[T + 273.15]}{q}\ln\frac{(\texttt{NDEP NSD})}{n_i^2} \tag{3.15}$$

The characteristic length is given by:

$$L_t = \begin{cases} \sqrt{\epsilon_s X_{\text{dep}}/C_{\text{oxe}}}\,(1 + \texttt{DVT2}\ V_{\text{BSeff}}) & \texttt{DVT2}\ V_{\text{BSeff}} \geq -0.5 \\ \sqrt{\epsilon_s X_{\text{dep}}/C_{\text{oxe}}}\,(1 + 3\ \texttt{DVT2}\ V_{\text{BSeff}}) \times (3 + 8\,\texttt{DVT2}\,V_{\text{BSeff}})^{-1} & \texttt{DVT2}\ V_{\text{BSeff}} < 0.5 \end{cases} \tag{3.16}$$

$$L_{\text{t0}} = \sqrt{\frac{\epsilon_s\ X_{\text{dep0}}}{C_{\text{oxe}}}} \tag{3.17}$$

where

$$C_{\text{oxe}} = \frac{\epsilon_{\text{ox}}}{\texttt{TOXE}} \tag{3.18}$$

$$L_{\text{tw}} = \begin{cases} \sqrt{\epsilon_s X_{\text{dep}}/C_{\text{oxe}}}\,(1 + \texttt{DVT2W}\ V_{\text{BSeff}}) & \texttt{DVT2W}\ V_{\text{BSeff}} \geq -0.5 \\ \sqrt{\epsilon_s X_{\text{dep}}/C_{\text{oxe}}}\,(1 + 3\ \texttt{DVT2W}\ V_{\text{BSeff}}) \times (3 + 8\,\texttt{DVT2W}\,V_{\text{BSeff}})^{-1} & \texttt{DVT2W}\ V_{\text{BSeff}} < 0.5 \end{cases} \tag{3.19}$$

$$X_{\text{dep}} = \sqrt{\frac{2\epsilon_s(2\phi_f - V_{\text{BSeff}})}{q\ \texttt{NDEP}}} \tag{3.20}$$

$$X_{\text{dep0}} = \sqrt{\frac{2\epsilon_s(2\phi_f)}{q\ \texttt{NDEP}}} \tag{3.21}$$

### 3.4.2 Effective Gate Source voltage

Care is taken in Equation 3.23 to make sure that the voltage across the poly-silicon gate does not exceed the silicon band gap voltage.

$$V_{\text{poly}} = \frac{q\ \epsilon_s\ \texttt{NGATE}\ C_{\text{oxe}}^2\ 10^6}{2}[\sqrt{1 + \frac{2(V_{\text{GS}} - V_{\text{FB}} - 2\phi_f)}{q\ \epsilon_s\ \texttt{NGATE}\ C_{\text{oxe}}^2\ 10^6}} - 1]^2 \tag{3.22}$$

$$V_{\text{PolyEff}} = 1.12 - 0.5\ (1.12 - V_{\text{poly}} - \delta + \sqrt{(1.12 - V_{\text{poly}} - \delta)^2 + 4\ \delta\ 1.12}) \tag{3.23}$$

$$V_{\mathrm{GSeff}} = V_{\mathrm{GS}} - V_{\mathrm{PolyEff}} \tag{3.24}$$

### 3.4.3 Effective $V_{\mathrm{GS}}$ - $V_{\mathrm{TH}}$ Smoothing Function

This function smoothes out the characteristics between the subthreshold and the strong inversion operating regions. It is approximately equal to $V_{\mathrm{GS}} - V_{\mathrm{T}}$ in strong inversion but becomes proportional to $\exp[q(V_{\mathrm{GS}} - V_{\mathrm{T}})/nkT]$ in the subthreshold region.

$$V_{\mathrm{GSTeff}} = \frac{n\frac{kT}{q}\ln(1+\exp(\frac{m}{n}\frac{V_{\mathrm{GSeff}}-V_{\mathrm{T}}}{nkT/q}))}{m + n\frac{C_{\mathrm{oxe}}}{C_{\mathrm{dep0}}}\exp[-\frac{(1-m)(V_{\mathrm{GSeff}}-V_{\mathrm{T}})-V_{\mathrm{off}}}{nkT/q}]} \tag{3.25}$$

where

$$m = \frac{1}{2} + \frac{\arctan(\mathtt{MINV})}{\pi} \tag{3.26}$$

$$V_{\mathrm{off}} = \mathtt{VOFF} + \frac{\mathtt{VOFFL}}{L_{\mathrm{EFF}}} \tag{3.27}$$

$$C_{\mathrm{dep0}} = \frac{\epsilon_s}{X_{\mathrm{dep0}}} \tag{3.28}$$

$$C_{\mathrm{dep}} = \frac{\epsilon_s}{X_{\mathrm{dep}}} \tag{3.29}$$

The ideality factor $n$ is

$$n = 1 + \mathtt{NFACTOR}\,\frac{C_{\mathrm{dep}}}{C_{\mathrm{oxe}}}\frac{\mathtt{CDSC} + \mathtt{CDSCD}\,V_{\mathrm{DS}} + \mathtt{CDSCB}\,V_{\mathrm{BSeff}}}{C_{\mathrm{oxe}}} \tag{3.30}$$

$$\times\frac{0.5}{\cosh(\mathtt{DVT1}\,L_{\mathrm{EFF}}/L_t - 1)} + \frac{\mathtt{CIT}}{C_{\mathrm{oxe}}} \tag{3.31}$$

As mentioned earlier, the $\Delta V_{\mathrm{T}}$ correction due to pocket implant requires the knowledge of the ideality factor $n$. Pocket implants near the source and drain regions increase the drive currents. They also increase the drain conductance. The pocket implant correction is modeled as

$$\Delta V_{\mathrm{T,PocketImplant}} = n\frac{kT}{q}\ln[\frac{L_{\mathrm{EFF}}}{L_{\mathrm{EFF}} + \mathtt{DVTP0}\ (1+\exp(-\mathtt{DVTP1}\ V_{\mathrm{DS}}))}] \quad (3.32)$$

## 3.5 Mobility characteristics

The mobility equations are based on the universal-mobility theorem which predicts the mobility degradation with increasing $V_{\mathrm{GS}}$. As an electron moves along the channel due to the lateral electric field, it is also attracted to the gate due to the normal electric field. This causes the electron to drift toward the gate and results in mobility degradation. This effect is modeled in the equations for mobility. We first define a variable TEMP.

$$\mathtt{TEMP} = (\mathtt{UA} + \mathtt{UC}\ V_{\mathrm{BSeff}})\ (\frac{V_{\mathrm{GSTeff}} + V_{\mathrm{T-fb-}\phi}}{\mathtt{TOXE}})^{\mathtt{EU}} \quad (3.33)$$

where EU is set to zero if the user suppiled value is negative and

$$V_{\mathrm{T-fb-}\phi} = \begin{cases} 2\ (\mathtt{VTH0} - V_{\mathrm{FB}} - 2\phi_f) & \text{for NMOS} \\ 2.5\ (\mathtt{VTH0} - V_{\mathrm{FB}} - 2\phi_f) & \text{for PMOS} \end{cases} \quad (3.34)$$

The effective mobility of the model takes the generalized form

$$\mu_{\mathrm{eff}} = \frac{\mathtt{U0}}{\mathtt{Denom}} \quad (3.35)$$

where

$$\mathtt{Denom} = \begin{cases} 1 + \mathtt{TEMP} & \text{if } \mathtt{TEMP} \geq -0.8 \\ (0.6 + \mathtt{TEMP})/(7 + 10\ \mathtt{TEMP}) & \text{if } \mathtt{TEMP} < -0.8 \end{cases} \quad (3.36)$$

## 3.6 Current characteristics

### 3.6.1 Effective Internal Source-Drain Resistance

This model only models an internal drain to source resistance, $R_{\rm DS}$ given by

$$R_{\rm DS} = \frac{\texttt{RDSWMIN} + \texttt{RDSW} \times 0.5\,[\texttt{TEMP} + \sqrt{\texttt{TEMP}^2 + 0.01}]}{(10^6 \times W_{\rm EFFCJ})^{WR}} \tag{3.37}$$

### 3.6.2 Bulk-Charge coefficient

The bulk-charge coefficient should always have a real physical value, i.e. greater than zero. BSIM4 ensures that this value is always greater than zero. It is an intermediate variable that aids in the evaluation of current and saturation voltage.

$$\begin{aligned} A_{\rm bulk} \;=\; & [1 - F_{\rm doping} \times [\frac{\texttt{A0}\, L_{\rm EFF}}{L_{\rm EFF} + 2\sqrt{\texttt{XJ}\, X_{\rm dep}}} \\ & (1 - \texttt{AGS}\, V_{\rm GSTeff}(\frac{L_{\rm EFF}}{L_{\rm EFF} + 2\sqrt{\texttt{XJ}\, X_{\rm dep}}})^2) + \frac{\texttt{B0}}{W_{\rm EFF} + \texttt{B1}}]] \\ & \times \frac{1}{1 + \texttt{KETA}\, V_{\rm BSeff}} \end{aligned} \tag{3.38}$$

where

$$\begin{aligned} F_{\rm doping} \;=\; & \sqrt{1 + \frac{\texttt{LPEB}}{L_{\rm EFF}}} \times \frac{\texttt{K1}}{2\sqrt{2\phi_f - V_{\rm BSeff}}}\, \frac{\texttt{TOXE}}{\texttt{TOXM}} \\ & + \texttt{K2}\, \frac{\texttt{TOXE}}{\texttt{TOXM}} - \texttt{K3} \times \frac{\texttt{TOXE}}{W_{\rm EFF} + \texttt{W0}}\, 2\,\phi_f \end{aligned} \tag{3.39}$$

### 3.6.3 Drain Saturation Voltage

The value of the drain voltage in the saturation region is given by

$$\epsilon_{\rm sat} = \frac{2\,\texttt{VSAT}}{\mu_{\rm eff}} \tag{3.40}$$

If $R_{\mathrm{DS}} = 0$, then

$$V_{\mathrm{DSsat}} = \frac{\epsilon_{\mathrm{sat}}\ L_{\mathrm{EFF}}\ (V_{\mathrm{GSTeff}} + 2kT/q)}{A_{\mathrm{bulk}}\ \epsilon_{\mathrm{sat}}\ L_{\mathrm{EFF}} + (V_{\mathrm{GSTeff}} + 2kT/q)} \tag{3.41}$$

If $R_{\mathrm{DS}} \neq 0$, then,

$$V_{\mathrm{DSsat}} = \frac{-b - \sqrt{b^2 - 4ac}}{2a} \tag{3.42}$$

$$a = A_{\mathrm{bulk}}^2\ W_{\mathrm{EFF}}\ \mathtt{VSAT}\ C_{\mathrm{oxe}}\ R_{\mathrm{DS}} + (\frac{1}{\lambda} - 1)\ A_{\mathrm{bulk}} \tag{3.43}$$

$$\begin{aligned} b \quad = \quad & -[(V_{\mathrm{GSTeff+2kT/q}})\ (\frac{2}{\lambda} - 1) + A_{\mathrm{bulk}}\ \epsilon_{\mathrm{sat}}\ L_{\mathrm{EFF}} \\ & +3\ A_{\mathrm{bulk}}\ (V_{\mathrm{GSTeff+2kT/q}})\ W_{\mathrm{EFF}}\ \mathtt{VSAT}\ C_{\mathrm{oxe}}\ R_{\mathrm{DS}}] \end{aligned} \tag{3.44}$$

$$c = (V_{\mathrm{GSTeff+2kT/q}})\ \epsilon_{\mathrm{sat}}\ L_{\mathrm{EFF}} + 2(V_{\mathrm{GSTeff+2kT/q}})^2 W_{\mathrm{EFF}}\ \mathtt{VSAT}\ C_{\mathrm{oxe}}\ R_{\mathrm{DS}} \tag{3.45}$$

where $\lambda$ is determined by parameters `A1` and `A2`.

### 3.6.4 Effective Drain to Source Voltage

$V_{\mathrm{DSeff}}$ varies between 0 (when $V_{\mathrm{DS}} = 0$) and $V_{\mathrm{DSsat}}$ in the saturation region, where $V_{\mathrm{DS}}$ is fairly high. It is a smoothing function defined to smooth out the transition between the linear and saturation regions. The parameter `DELTA` can be varied between 0.1 and 0.001 to get greater control of the transition between the linear and saturation region.

$$\begin{aligned} V_{\mathrm{DSeff}} \quad = \quad & V_{\mathrm{DSsat}} - 0.5\ (V_{\mathrm{DSsat}} - V_{\mathrm{DS}} - \mathtt{DELTA} \\ & + \sqrt{(V_{\mathrm{DSsat} - V_{DS} - \mathtt{DELTA})^2 + 4\ \mathtt{DELTA}\ V_{DSsat}}}) \end{aligned} \tag{3.46}$$

### 3.6.5 Effective oxide calculation

The drain current, in BSIM4, has a significant amount of drain current in the strong inversion region. It calculates the maximum probability of carrier distribution occurs at a distance $X_{\mathrm{DC}}$ away from the interface. The oxide capacitance for the inversion calculation is given by

$$C_{\mathrm{ox}} = \frac{\epsilon_{\mathrm{ox}}}{\mathtt{TOXP}} \parallel \frac{\epsilon_s}{X_{\mathrm{DC}}} \tag{3.47}$$

where

$$X_{\mathrm{DC}} = \begin{cases} 1.9 \times 10^{-9} \\ \times[1 + \frac{V_{\mathrm{GSTeff}} + 4(\mathtt{VTH0} - V_{\mathrm{FB}} - 2\phi_f)}{2 \times 10^8\ \mathtt{TOXP}}]^{-0.7} & (V_{\mathrm{TH0}} - V_{\mathrm{FB}} - 2\phi_f) \geq 0 \\ \\ 1.9 \times 10^{-9} \\ \times[1 + V_{\mathrm{GSTeff}}/(2 \times 10^8\ \mathtt{TOXP})]^{-0.7} & (V_{\mathrm{TH0}} - V_{\mathrm{FB}} - 2\phi_f) < 0 \end{cases} \tag{3.48}$$

## 3.7 Current Calculations

This is the dominant portion of drain current. It flows from the drain to the source through the channel.

$$\begin{aligned} I_{\mathrm{DS}} &= \frac{I_{\mathrm{DS0}}}{1 + \frac{R_{\mathrm{DS}} I_{\mathrm{DS0}}}{V_{\mathrm{DSeff}}}} (1 + \frac{1}{C_{\mathrm{clm}}} \ln \frac{V_A}{V_{\mathrm{Asat}}}) \times (1 + \frac{V_{\mathrm{DS}} - V_{\mathrm{DSeff}}}{V_{\mathrm{ADIBL}}}) \\ & \times (1 + \frac{V_{\mathrm{DS}} - V_{\mathrm{DSeff}}}{V_{\mathrm{ADITS}}}) \times (1 + \frac{V_{\mathrm{DS}} - V_{\mathrm{DSeff}}}{V_{\mathrm{ASCBE}}}) \times \mathtt{NF} \end{aligned} \tag{3.49}$$

where the ideal long channel current in the absence of channel length modulation and DIBL effects is given by

$$I_{\mathrm{DS0}} = \frac{W_{\mathrm{EFF}}\ \mu_{\mathrm{eff}}\ C_{\mathrm{ox}}\ V_{\mathrm{GSTeff}}}{L_{\mathrm{EFF}}\ [1 + V_{\mathrm{DSeff}}/(\epsilon_{\mathrm{sat}} L_{\mathrm{EFF}})]} [1 - \frac{A_{\mathrm{bulk}}\ V_{\mathrm{DSeff}}}{2(V_{\mathrm{GSTeff}} + 2kT/q)}]\ V_{\mathrm{DSeff}} \tag{3.50}$$

There Early voltage is given as

$$V_A = V_{\text{Asat}} + V_{\text{ACLM}} \tag{3.51}$$

where $V_{\text{Asat}}$ calculates the ideal early voltage in the absence of short channel effects and $V_{\text{ACLM}}$ models channel length modulation.

$$\begin{aligned} V_{\text{Asat}} &= \frac{\epsilon_{\text{sat}}\, L_{\text{EFF}} + V_{\text{DSsat}} + 2R_{\text{DS}}\, \texttt{VSAT}\, C_{\text{oxe}} W_{\text{EFF}}\, V_{\text{GSTeff}}}{2/\lambda - 1 + R_{\text{DS}}\, \texttt{VSAT}\, C_{\text{oxe}}\, W_{\text{EFF}}\, A_{\text{bulk}}} \\ & \times [1 - \frac{A_{\text{bulk}}\, V_{\text{DSsat}}}{2\, (V_{\text{GSTeff}} + 2kT/q)}] \end{aligned} \tag{3.52}$$

The degradation factor due to pocket implantation is given by

$$F_p = \begin{cases} (1 + \texttt{FPROUT}\, \sqrt{L_{\text{EFF}}}/(V_{\text{GSTeff}} + 2kT/q))^{-1} & \texttt{FPROUT} > 0 \\ 1 & \texttt{FPROUT} \leq 0 \end{cases} \tag{3.53}$$

To account for the effects of gate-bias on the slope of $I_{\text{DS}}$ in the saturation region, we use

$$F_{\text{VG}} = \begin{cases} 1 + \texttt{PVAG}\, V_{\text{GSTeff}}/(\epsilon_{\text{sat}}\, L_{\text{EFF}}) & \texttt{PVAG}\, V_{\text{GSTeff}}/(\epsilon_{\text{sat}}\, L_{\text{EFF}}) > -0.9 \\ \\ 0.8 + \texttt{PVAG}\, V_{\text{GSTeff}}/(\epsilon_{\text{sat}}\, L_{\text{EFF}}) & \\ \times(17 + 20\, \texttt{PVAG}\, V_{\text{GSTeff}}/(\epsilon_{\text{sat}}\, L_{\text{EFF}}))^{-1} & \end{cases} \tag{3.54}$$

$$C_{\text{clm}} = \begin{cases} \texttt{TEMP} & \texttt{PCLM} > 0 \textit{ and } (V_{\text{DS}} - V_{\text{DSeff}}) > 10^{-10} \\ 5.834617425 \times 10^{14} & \end{cases} \tag{3.55}$$

where

$$\texttt{TEMP} = \frac{F_p}{\texttt{PCLM}\, L_{\text{itl}}} F_{\text{VG}} \times (1 + \frac{R_{\text{DS}}\, I_{\text{DS0}}}{V_{\text{DSeff}}}) \times (L_{\text{EFF}} + \frac{V_{\text{DSsat}}}{\epsilon_{\text{sat}}}) \tag{3.56}$$

$$\theta_{\text{rout}} = \texttt{PDIBLC1} \times [\exp(-\texttt{DROUT}\, L_{\text{EFF}}/2L_{\text{t0}}) + 2\exp(-\texttt{DROUT}\, L_{\text{EFF}}/2L_{\text{t0}})] + \texttt{PDIBLC2} \quad (3.57)$$

The effect of DIBL on Early voltage is modeled as:

$$V_{\text{ADIBL}} = \begin{cases} V_{\text{GSTeff}} + 2kT/q/(\theta_{\text{rout}}(1 + \texttt{PDIBLCB}\, V_{\text{BSeff}})) \times F_{\text{VG}} \\ \times[1 - A_{\text{bulk}}V_{\text{DSsat}}/(A_{\text{bulk}}V_{\text{DSsat}} + V_{\text{GSTeff}} + 2kT/q)] & \theta_{\text{rout}} \geq 0 \\ 5.834617425 \times 10^{10} & \theta_{\text{rout}} < 0 \end{cases} \quad (3.58)$$

The effect of DITS (Drain-induced Threshold Shift) due to pocket implant is modeled by

$$V_{\text{ADITS}} = \begin{cases} \frac{F_p}{\texttt{PDITS}}[1 + (1 + \texttt{PDITSL}\, L_{\text{EFF}}) \times \exp(\texttt{PDITSD}\, V_{\text{DS}})] & \texttt{PDITS} > 0 \\ 5.834617425 \times 10^{14} & else \end{cases} \quad (3.59)$$

Substrate current has an effect on Early voltage and it is modeled by SCBE (Substrate Current Induced Body Effect).

$$V_{\text{ASCBE}} = \begin{cases} \frac{L_{\text{EFF}}}{\texttt{PSCBE2}} \exp(\texttt{PSCBE2}\, L_{\text{itl}}/(V_{\text{DS}} - V_{\text{DSeff}})) & \texttt{PSCBE2} > 0 \\ 5.834617425 \times 10^{10} & else \end{cases} \quad (3.60)$$

where

$$L_{\text{itl}} = \sqrt{\epsilon_s/\epsilon_{\text{ox}} \times \texttt{TOXE}\,\texttt{XJ}} \quad (3.61)$$

### 3.7.1 Substrate Currents

The substrate currents comprise of two parts, one due to Impact Ionization and the other due to GIDL. When $V_{\text{DS}}$ is high, a large voltage is dropped

across the depletion region near the drain. This field accelerates the electrons as they are moving in the channel. When they generate sufficient energy, the collide with the semiconductor crystal and generate electron-hole pairs. This current that is generated flows towards the substrate. This forms the Impact Ionization current and is denoted by $I_{\text{sub}}$.

$$\begin{aligned} I_{\text{sub}} &= \texttt{NF} \times (\frac{\texttt{ALPHA0}}{L_{\text{EFF}}} + \texttt{ALPHA1})(V_{\text{DS}} - V_{\text{DSeff}}) \exp[-\frac{\texttt{BETA0}}{V_{\text{DS}} - V_{\text{DSeff}}}] \\ & \frac{I_{\text{DS0}}}{1 + \frac{R_{\text{DS}} I_{\text{DS0}}}{V_{\text{DSeff}}}} (1 + \frac{1}{C_{\text{clm}}} \ln \frac{V_A}{V_{\text{Asat}}}) \times (1 + \frac{V_{\text{DS}} - V_{\text{DSeff}}}{V_{\text{ADIBL}}}) \\ & \times (1 + \frac{V_{\text{DS}} - V_{\text{DSeff}}}{V_{\text{ADITS}}}) \end{aligned} \tag{3.62}$$

The contribution due to GIDL is given by

$$\begin{aligned} I_{\text{gidl}} &= \texttt{NF} \times \texttt{AGIDL}\, W_{\text{EFFCJ}} [\frac{V_{\text{DS}} - V_{\text{GSeff}} - \texttt{EGIDL}}{3\ \texttt{TOXE}}] \\ & \times \exp(\frac{-3\ \texttt{TOXE} \times \texttt{BGIDL}}{V_{\text{DS}} - V_{\text{GSeff}} - \texttt{EGIDL}}) \\ & \frac{V_{\text{DB}}^3}{\texttt{CGIDL} + V_{\text{DB}}^3} \end{aligned} \tag{3.63}$$

### 3.7.2 Gate Currents

As the oxide layer becomes progressively thinner, the tunnelling currents flowing through the oxide become more significant. This model considers four tunnelling currents as shown in Fig 3.1.

$I_{\text{gd}}$ is the tunnelling current between the gate and the heavily-doped drain. $I_{\text{gcd}}$ denotes the current that flows from the gate to the channel and then to the drain. Likewise, $I_{\text{gs}}$ and $I_{\text{gcs}}$ are similar tunnelling currents, but associated with the source junction. The current $I_{gb}$ represents the current flowing from

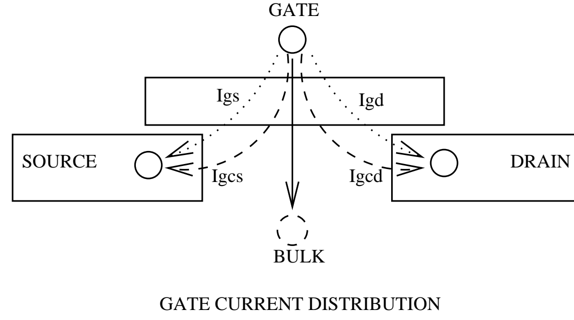


Figure 3.1: Schematic of the gate current distribution in the FET

the gate to the bulk.

The voltage drop across the oxide is given by

$$V_{\text{ox}} = V_{\text{FB}} - V_{\text{FBeff}} + \texttt{K1}\sqrt{\phi_s} + V_{\text{GSTeff}} \tag{3.64}$$

The first two terms of the above equation represent voltage dropped in the accumulation region or $V_{oxacc}$ and the depletion/inversion region or $V_{\text{oxdepinv}}$. The two channel tunnelling components are given by

$$I_{\text{gcs}} = I_{\text{gc}} \times \frac{-1 + \texttt{PIGCD}\, V_{\text{DS}} + \exp(-\texttt{PIGCD}\, V_{\text{DS}} + 10^{-4})}{(\texttt{PIGCD}\, V_{\text{DS}})^2 + 2 \times 10^{-4}} \tag{3.65}$$

$$I_{\text{gcd}} = I_{\text{gc}} \times \frac{1 - (1 + \texttt{PIGCD}\, V_{\text{DS}}) + \exp(-\texttt{PIGCD}\, V_{\text{DS}} + 10^{-4})}{(\texttt{PIGCD}\, V_{\text{DS}})^2 + 2 \times 10^{-4}} \tag{3.66}$$

Both these currents have dependencies on the drain-source voltage $V_{\text{DS}}$. Generally, these currents do not sum up to $I_{\text{gc}}$. However, when $V_{\text{DS}}$ is zero, they are identical to each other and equal to half of $I_{\text{gc}}$, which is given by

$$\begin{aligned} I_{\text{gc}} &= \texttt{NF}\, W_{\text{EFF}}\, L_{\text{EFF}} \frac{A}{(\texttt{TOXE})^2} (\frac{\texttt{TOXREF}}{\texttt{TOXE}})^{\texttt{NTOX}}\, V_{\text{GSeff}} \\ & \times \texttt{NIGC} \frac{kT}{q} \ln[1 + \exp(\frac{q(V_{\text{GSeff}} - \texttt{VTH0})}{kT.\texttt{NIGC}})] \\ & \exp[-B.\texttt{TOXE}(\texttt{AIGC} - \texttt{BIGC}V_{\text{oxdepinv}}).(1 + \texttt{CIGC}V_{\text{oxdepinv}})] \end{aligned} \tag{3.67}$$

The coefficients used in the above equations are

$$A = 4.97232 \times 10^{-7} \quad (3.68)$$

$$B = 7.45669 \times 10^{11} \quad (3.69)$$

The currents associated with the gate and source/drain regions is given by

$$\begin{aligned} I_{\mathrm{gs}} &= \mathtt{NF}\ W_{\mathrm{EFF}}\ \mathtt{DLCIG} \frac{A}{(\mathtt{TOXE\ POXEDGE})^2} \left(\frac{\mathtt{TOXREF}}{\mathtt{TOXE\ POXEDGE}}\right)^{\mathtt{NTOX}} V_{\mathrm{GS}} \times V'_{\mathrm{GS}} \\ & \exp[-B\ \mathtt{TOXE\ POXEDGE}(\mathtt{AIGSD} - \mathtt{BIGSD}\ V_{\mathrm{GS}}) \\ & (1 + \mathtt{CIGSD}\ V'_{\mathrm{GS}})] \end{aligned} \quad (3.70)$$

$$V'_{\mathrm{GS}} = \sqrt{(V_{\mathrm{GS}} - V_{\mathrm{fbsd}})^2 + 10^{-4}} \quad (3.71)$$

$$\begin{aligned} I_{\mathrm{gd}} &= \mathtt{NF}\ W_{\mathrm{EFF}}\ \mathtt{DLCIG} \frac{A}{(\mathtt{TOXE\ POXEDGE})^2} \left(\frac{\mathtt{TOXREF}}{\mathtt{TOXE\ POXEDGE}}\right)^{\mathtt{NTOX}} V_{\mathrm{GD}} \times V'_{\mathrm{GD}} \\ & \exp[-B\ \mathtt{TOXE\ POXEDGE}(\mathtt{AIGSD} - \mathtt{BIGSD}\ V_{\mathrm{GD}}) \\ & (1 + \mathtt{CIGSD}\ V'_{\mathrm{GD}})] \end{aligned} \quad (3.72)$$

$$V'_{\mathrm{GD}} = \sqrt{(V_{\mathrm{GD}} - V_{\mathrm{fbsd}})^2 + 10^{-4}} \quad (3.73)$$

The resultant dc equivalent circuit for the transistor is shown in Figure 3.2

## 3.8 Charge computation and Conservation

### 3.8.1 Basic Formulation

To ensure charge conservation, terminal charges are used as state variables along with terminal voltages. $Q_g$, $Q_s$, $Q_d$ and $Q_b$ are the charges associated

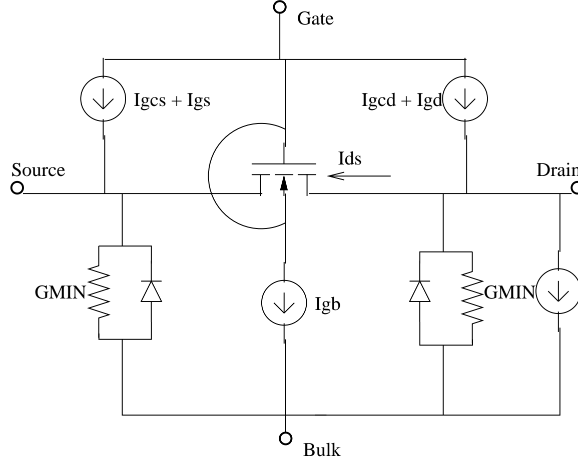


Figure 3.2: Schematic of the dc equivalent circuit

with the gate, source, drain and bulk terminals respectively. The gate charge comprises of the inversion charge $Q_{\rm inv}$, the accumulation charge $Q_{\rm acc}$ and the substrate depletion charge $Q_{\rm sub}$.

The channel charge comes from the source and drain terminals while the accumulation and substrate charge is associated with the substrate.

$$\begin{aligned} Q_g &= -(Q_{\rm sub} + Q_{\rm inv} + Q_{\rm acc}) \\ Q_b &= Q_{\rm acc} + Q_{\rm sub} \\ Q_{\rm inv} &= Q_d + Q_s \end{aligned} \tag{3.74}$$

The substrate charge can be divided further into two components: the substrate charge at zero source-drain bias ($Q_{\rm sub0}$) and a non-uniform substrate charge in the presence of a drain bias ($\delta Q_{\rm sub}$). The gate charge now becomes

$$Q_g = -(Q_{\rm sub0} + \delta Q_{\rm sub} + Q_{\rm inv} + Q_{\rm acc}) \tag{3.75}$$

The total charge is computed by integrating the charge along the channel. SPICE3 provides three options in BSIM4 whereby a user can select the per-

centage of charge distribution between the source and drain. The options are a 0/100 distribution which implies that no channel charge is associated with the source and it is all assigned to the drain, a 50/50 partition which divides the charges equally between the source and drain and a 40/60 partition wherein the total charge in the channel is divided in a 40:60 ratio between the source and drain. Transim uses a 40/60 charge partitioning scheme between the source and drain terminals because that is the closest to a physical situation in the channel.

### 3.8.2 Effective $V_{\mathrm{BS}}$, $V_{GB}$

This is a smoothing function required for C-V calculations.

$$V_{\mathrm{BSeffCV}} = \begin{cases} V_{\mathrm{BSeff}} & V_{\mathrm{BSeff}} < 0 \\ \phi_s - \frac{\phi_s}{\phi_s + V_{\mathrm{BSeff}}} & V_{\mathrm{BSeff}} \geq 0 \end{cases} \tag{3.76}$$

$$V_{\mathrm{GBeffCV}} = V_{gse} - V_{\mathrm{BSeffCV}} \tag{3.77}$$

### 3.8.3 Effective $V_{\mathrm{GS}} - V_{\mathrm{T}}$

This is also a smoothing function used in the C-V calculation.

$$V_{\mathrm{GSTeffCV}} = \mathtt{NOFF}\,\frac{nkT}{q}\,\ln[1 + \exp(\frac{V_{\mathrm{GSeff}} - V_{\mathrm{T}} - \mathtt{VOFFCV}}{\mathtt{NOFF}\,nkT/q})] \tag{3.78}$$

### 3.8.4 Modified Bulk Charge coefficient

For C-V calculations, the reduced bulk charge coefficient is used instead of the coefficient used in the DC calculations.

$$A_{\mathrm{bulk0}} = [1 - F_{\mathrm{doping}} \times [\frac{\mathtt{A0}\,L_{\mathrm{EFF}}}{L_{\mathrm{EFF}} + 2\sqrt{\mathtt{XJ}\,X_{\mathrm{dep}}}} + \frac{\mathtt{B0}}{W_{\mathrm{EFF}} + \mathtt{B1}}]]$$

$$\times \frac{1}{1 + \texttt{KETA}\ V_{\mathrm{BSeff}}} \tag{3.79}$$

Using this value of reduced bulk-charge coefficient, the bulk-charge coefficient for C-V calculations is given by

$$A_{\mathrm{bulkCV}} = A_{\mathrm{bulk0}}[1 + (\frac{\texttt{CLC}}{L_{\mathrm{EFF}}})^{\texttt{CLE}}] \tag{3.80}$$

### 3.8.5 The Terminal Charges

The effective oxide thickness

$$C'_{\mathrm{oxeff}} = \frac{\epsilon_{\mathrm{ox}}}{\texttt{TOXP}} \| \frac{\epsilon_s}{X_{\mathrm{DCeff}}} \tag{3.81}$$

where

$$X_{\mathrm{DCeff}} = X_{\mathrm{DCmax}} - \frac{X_{\mathrm{DCmax}} - X_{\mathrm{DC}} - \delta_x + \sqrt{(X_{\mathrm{DCmax}} - X_{DC} - \delta_x)^2 + 4\ \delta_x\ X_{\mathrm{DCmax}}}}{2} \tag{3.82}$$

The various terms inside the above equation are given by

$$X_{\mathrm{DC}} = \frac{L_{\mathrm{Deb}}}{3} \exp[\texttt{ACDE}(\frac{\texttt{NDEP}}{2 \times 10^{16}})^{-0.25}\ \frac{V_{\mathrm{GBeff}} - V_{\mathrm{fbzb}}}{10^8 \times \texttt{TOXP}}] \tag{3.83}$$

$$L_{\mathrm{Deb}} = \sqrt{\frac{\epsilon_s\ k[T + 273.15]/q}{q\texttt{NDEP}10^6}} \tag{3.84}$$

$$X_{\mathrm{DCmax}} = \frac{L_{\mathrm{Deb}}}{3} \tag{3.85}$$

$$\delta_x = 10^{-3}\texttt{TOXP} \tag{3.86}$$

The effective oxide is re-calculated for evaluation of the accumulation charge.

$$C_{\mathrm{oxeff}} = C'_{\mathrm{oxeff}} \times W_{\mathrm{EFF}} \times L_{\mathrm{EFF}} \times \texttt{NF} \tag{3.87}$$

The accumulation charge is given by

$$Q_{\text{acc}} = C_{\text{oxeff}}\ (V_{\text{FBeffCV}} - V_{\text{FBCV}}) \tag{3.88}$$

The substrate charge is given by

$$Q_{\text{sub0}} = C_{\text{oxeff}}\ (\texttt{K1}\frac{\texttt{TOXE}}{\texttt{TOXM}})\ \sqrt{\phi_{\text{s,dep}}} \tag{3.89}$$

When the transistor enters the sub-threshold region, another value of $X_{\text{DC}}$ is required. This value is used in the evaluation of $Q_{\text{inv}}$ and $\delta Q_{\text{sub}}$.

$$C'_{\text{oxinv}} = \frac{\epsilon_{\text{ox}}}{\texttt{TOXP}} \| \frac{\epsilon_s}{X_{\text{DCinv}}} \tag{3.90}$$

where

$$X_{\text{DCinv}} = \begin{cases} 1.9 \times 10^{-9} \\ \times[1 + \frac{V_{\text{GSTeffCV}} + 4(\texttt{VTH0} - V_{\text{FB}} - 2\phi_f)}{2\times 10^8\ \texttt{TOXP}}]^{-0.7} & (V_{\text{TH0}} - V_{\text{FB}} - 2\phi_f) \geq 0 \\ \\ 1.9 \times 10^{-9} \\ \times[1 + V_{\text{GSTeffCV}}/(2 \times 10^8\ \texttt{TOXP})]^{-0.7} & (V_{\text{TH0}} - V_{\text{FB}} - 2\phi_f) < 0 \end{cases} \tag{3.91}$$

The above equation is identical to $X_{\text{DC}}$ used for I-V calculations, except that $V_{\text{GSTeff}}$ is replaced by $V_{\text{GSTeffCV}}$.

$$C_{\text{oxinv}} = C'_{\text{oxinv}} \times W_{\text{EFF}} \times L_{\text{EFF}} \times \texttt{NF} \tag{3.92}$$

In addition to the new $X_{\text{DC}}$, the surface potential is not constant as in the I-V case and needs to be re-calculated.

$$\phi_\delta = \begin{cases} \frac{kT}{q} \ln[1 + V_{\text{GSTeffCV}}(V_{\text{GSTeffCV}} + 2\texttt{K1}\ (\texttt{TOXE/TOXM})\ (\phi_s))] \\ -\frac{kT}{q}\ \ln[\texttt{MOIN}\ \texttt{K1}^2(\texttt{TOXE/TOXM})^2\ (kT/q)] & \texttt{K1} > 0 \\ \\ \frac{kT}{q} \ln[1 + V_{\text{GSTeffCV}}(V_{\text{GSTeffCV}} + \sqrt{\phi_s})/(0.25 \times \texttt{MOIN}(kT/q))] & \texttt{K1} \leq 0 \end{cases} \tag{3.93}$$

$$V_{\mathrm{DSsatCV}} = \frac{V_{\mathrm{GSTeffCV}} - \phi_\delta}{A_{\mathrm{bulkCV}}} \tag{3.94}$$

$$\begin{aligned} V_{\mathrm{DSeffCV}} &= V_{\mathrm{DSsatCV}} - \frac{V_{\mathrm{DSsatCV}} - V_{\mathrm{DS}} - 0.02}{2} \\ &\quad - \sqrt{\frac{(V_{\mathrm{DSsatCV}} - V_{\mathrm{DS}} - 0.02)^2 + 4\;0.02\; V_{\mathrm{DSsatCV}}}{2}} \end{aligned} \tag{3.95}$$

Based on these calculations, the inversion charge can be written as

$$\begin{aligned} Q_{\mathrm{inv}} &= -C_{\mathrm{oxinv}}[(V_{\mathrm{GSTeffCV}} - \phi_\delta - \frac{A_{\mathrm{bulkCV}} V_{\mathrm{DSeffCV}}}{2}) \\ &\quad \frac{A_{\mathrm{bulkCV}}^2 V_{\mathrm{DSeffCV}}^2}{12(V_{\mathrm{GSTeffCV}} - \phi_\delta - A_{\mathrm{bulkCV}} V_{\mathrm{DSeffCV}}/2 + 10^{-20})}] \end{aligned} \tag{3.96}$$

The factor of $10^{-20}$ exists to mainly prevent the denominator from going to a negative value when the rest of the terms go close to zero.
The substrate charge in the presence of a drain bias is given by

$$\delta Q_{\mathrm{sub}} = C_{\mathrm{oxinv}}[\frac{1 - A_{\mathrm{bulkCV}}}{2} V_{\mathrm{DSeffCV}} \tag{3.97}$$

$$- \frac{(1 - A_{\mathrm{bulkCV}})\; A_{\mathrm{bulkCV}}\; V_{\mathrm{DSeffCV}}^2}{12\,(V_{\mathrm{GSTeffCV}} - \phi_\delta - A_{\mathrm{bulkCV}}\; V_{\mathrm{DSeffCV}}/2 + 10^{-20})}] \tag{3.98}$$

Finally, the four charges at the respective terminals are given by

$$Q_g = -Q_{\mathrm{inv}} - \delta Q_{\mathrm{sub}} + Q_{\mathrm{acc}} + Q_{\mathrm{sub0}} \tag{3.99}$$

$$Q_b = \delta Q_{\mathrm{sub}} - Q_{\mathrm{acc}} - Q_{\mathrm{sub0}} \tag{3.100}$$

For a 40/60 charge partition scheme, the charge at the source and drain regions is

$$Q_s = -\frac{C_{\mathrm{oxinv}}}{2(V_{\mathrm{GSTeffCV}} - \phi_\delta - \frac{A_{\mathrm{bulkCV}} V_{\mathrm{DSeffCV}}}{2})^2}$$

$$
\begin{aligned}
&\times[(V_{\text{GSTeffCV}} - \phi_\delta)^3 \\
&-\frac{4}{3}(V_{\text{GSTeffCV}} - \phi_\delta)^2 A_{\text{bulkCV}} V_{\text{DSeffCV}} \\
&+\frac{2}{3}(V_{\text{GSTeffCV}} - \phi_\delta)^2 A_{\text{bulkCV}} V_{\text{DSeffCV}} \\
&-\frac{2}{15} A_{\text{bulkCV}}^3 V_{\text{DSeffCV}}^3 \qquad (3.101)
\end{aligned}
$$

$$
\begin{aligned}
Q_d &= -\frac{C_{\text{oxinv}}}{2(V_{\text{GSTeffCV}} - \phi_\delta - \frac{A_{\text{bulkCV}} V_{\text{DSeffCV}}}{2})^2} \\
&\times[(V_{\text{GSTeffCV}} - \phi_\delta)^3 \\
&-\frac{5}{3}(V_{\text{GSTeffCV}} - \phi_\delta)^2 A_{\text{bulkCV}} V_{\text{DSeffCV}} \\
&+(V_{\text{GSTeffCV}} - \phi_\delta)^2 A_{\text{bulkCV}} V_{\text{DSeffCV}} \\
&-\frac{1}{5} A_{\text{bulkCV}}^3 V_{\text{DSeffCV}}^3 \qquad (3.102)
\end{aligned}
$$

The net currents at the gate, source and drain is given by

$$
i_G(t) = I_{\text{gcs}} + I_{\text{gcd}} + I_{\text{gs}} + I_{\text{gd}} + \frac{dQ_g}{dt} \qquad (3.103)
$$

$$
i_S(t) = -I_{\text{DS}} - I_{\text{gs}} - I_{\text{gcs}} + \frac{dQ_s}{dt} \qquad (3.104)
$$

$$
i_D(t) = I_{\text{DS}} + I_{\text{sub}} + I_{\text{gidl}} - I_{\text{gcd}} - I_{\text{gd}} + \frac{dQ_d}{dt} \qquad (3.105)
$$

# Chapter 4

# Simulations and Results

The purpose of this chapter is to present some of the results of the model implemented in Transim. The circuit used for the test is an NMOS common source transistor used as a small signal amplifier. The drain is biased at 5 Volts DC and a sinusoidal signal is applied at the gate terminal at a frequency of 1 GHz. This is indicated in Figure 4.1. The output is measured at the drain terminal with a drain resistance of 1000 Ω.

The circuit is also driven with a square pulse with a finite rise and fall time, using both SPICE3 and Transim. The pulse has a peak value of 1.5 Volts and a rise and fall time of 0.2 nano-seconds and a frequency of 0.1 GHz.

In Section 4.1, a DC analysis is performed on the implementation of the transistor in Transim. Graphs of drain current versus drain-source voltage at different values of gate-source voltage have been generated. The figures show a DC family of curves as well as comparisons of simulations of the same model used in SPICE3 under the same operating conditions.

In Section 4.2, a Transient analysis has been performed on the implementation of the transistor in Transim. The type of integration method used

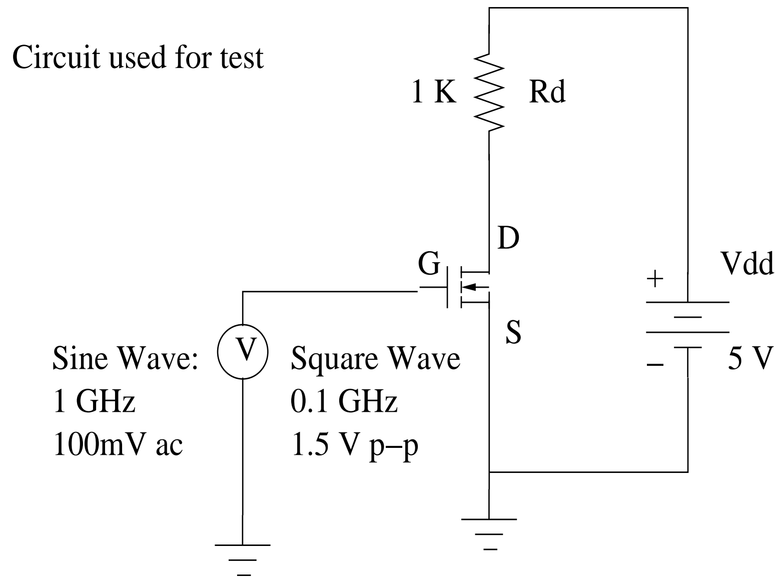


Figure 4.1: Common Source Amplifier

is Backward Euler. Graphs of drain-source voltage versus time have been generated using a 1 GHz time varying sinusoidal signal at the gate and comparisons have been made using the same model in SPICE3 under the same operating conditions.

Transient analysis results of the circuit driven with a 0.1 GHz square pulse are also compared with SPICE3 under the same conditions and presented.

The source code, which comprises of a C++ file and a header file, used to describe the model in Transim can be found in Appendix A.

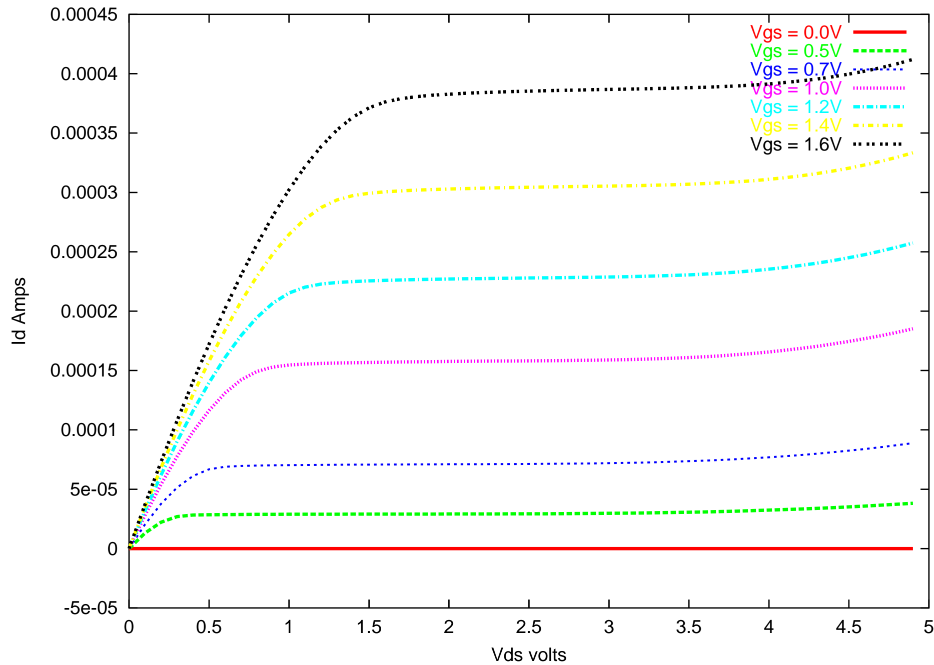


Figure 4.2: DC Analysis family of curves

## 4.1 DC analysis

The gate source voltage has been varied from 0 to 1.5 volts keeping the drain bias set at 5 volts. A family of curves of drain current versus drain-source voltage are plotted with varying values of gate-source voltage. The results are shown in Figure 4.2.

A comparison is made using the same circuit in SPICE3 under the same conditions at a gate bias of $V_{\mathrm{GS}} = 1.3\mathrm{V}$ with the comparison shown in Figure 4.3. A comparison with a gate bias of $V_{\mathrm{GS}} = 1.4\mathrm{V}$ is shown in Figure 4.4. A comparison using a gate bias of $V_{\mathrm{GS}} = 1.5\mathrm{V}$ is shown in Figure 4.5 and finally using a gate bias of $V_{\mathrm{GS}} = 2.0\mathrm{V}$ in Figure 4.6.

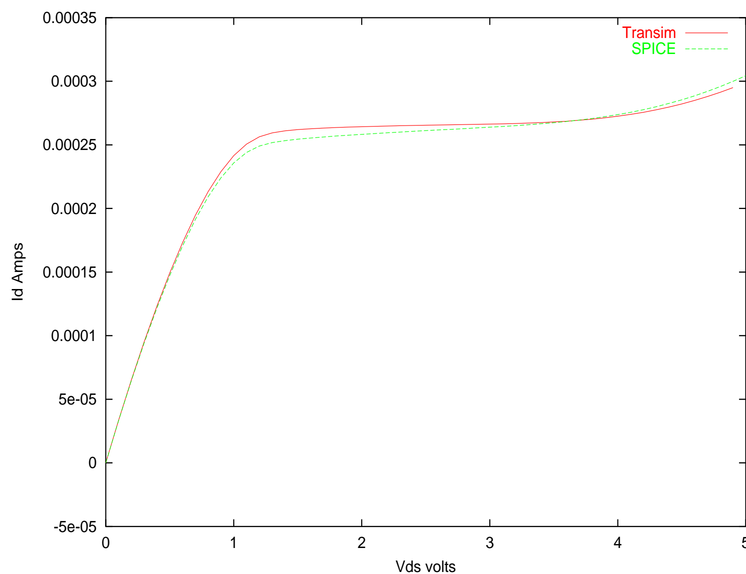


Figure 4.3: DC Analysis comparison at $V_{\mathrm{GS}} = 1.3\mathrm{V}$

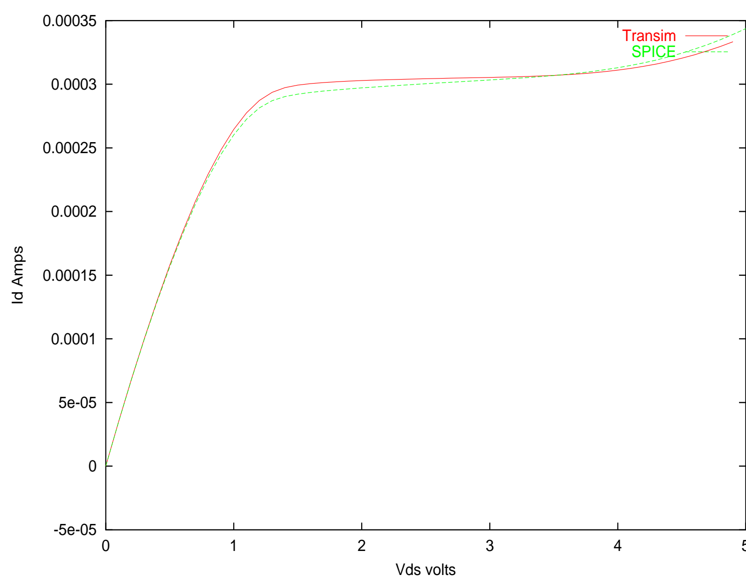


Figure 4.4: DC Analysis comparison at $V_{\mathrm{GS}} = 1.4\mathrm{V}$

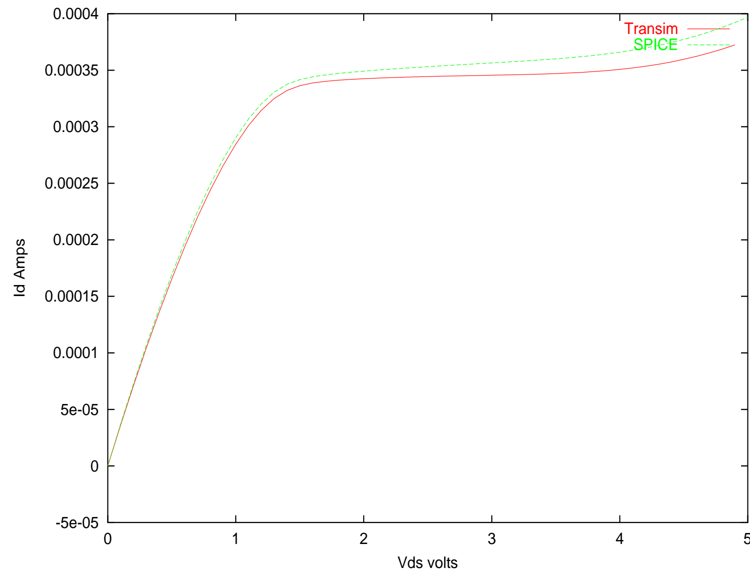


Figure 4.5: DC Analysis comparison at $V_{\mathrm{GS}} = 1.5\mathrm{V}$

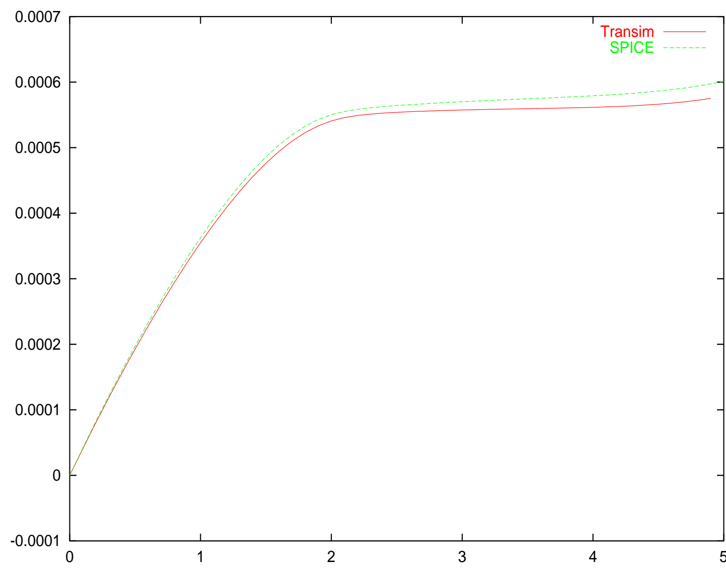


Figure 4.6: DC Analysis comparison at $V_{\mathrm{GS}} = 2.0\mathrm{V}$

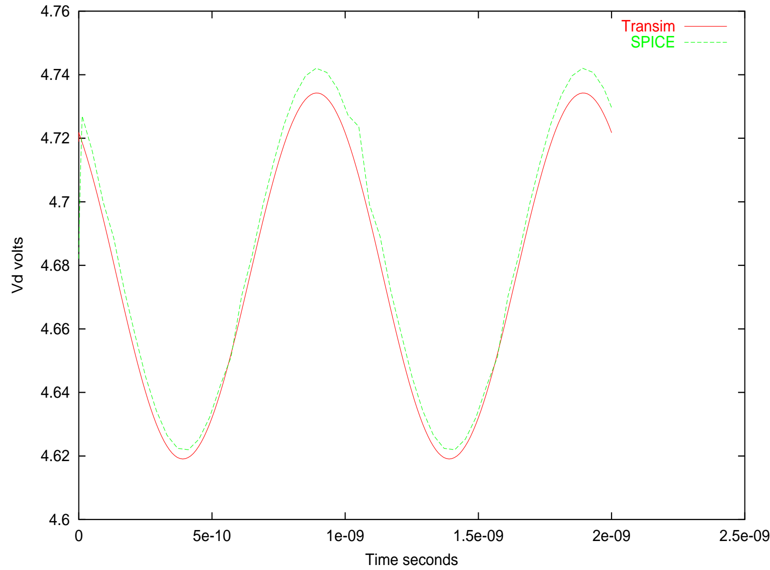


Figure 4.7: Transient Analysis comparison at $V_{\mathrm{GS}} = 1.3\mathrm{V}$

## 4.2 Transient analysis

### 4.2.1 Sine wave input

Transient analysis is carried out at a frequency of 1GHz in both SPICE3 and TRANSIM. A gate-source sinusoidal voltage of 100 mV with different values of DC bias have been applied. A comparison using a gate-source bias of $V_{gs}$ = 1.3 V DC is shown in Fig 4.7. A comparison using a gate-source bias of $V_{gs}$ = 1.5 V DC is shown in Fig 4.8 and finally a comparison using a gate-source bias of $V_{gs}$ = 1.7 V DC is shown in Fig 4.9. The integration method used in Transim is Backward Euler and it uses a variable time-stepping algorithm. The duration of the time step is calculated automatically and varies during run time. It is similar to the routine used in SPICE3.

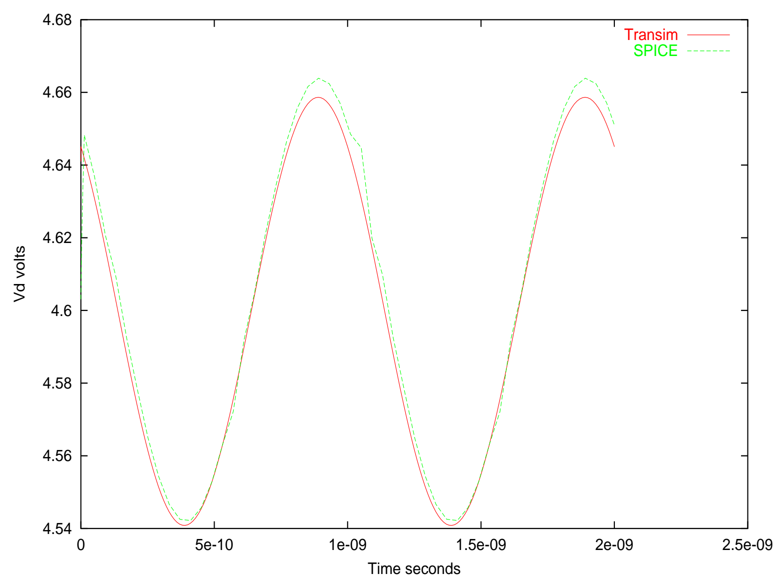


Figure 4.8: Transient Analysis comparison at $V_{\mathrm{GS}} = 1.5\mathrm{V}$

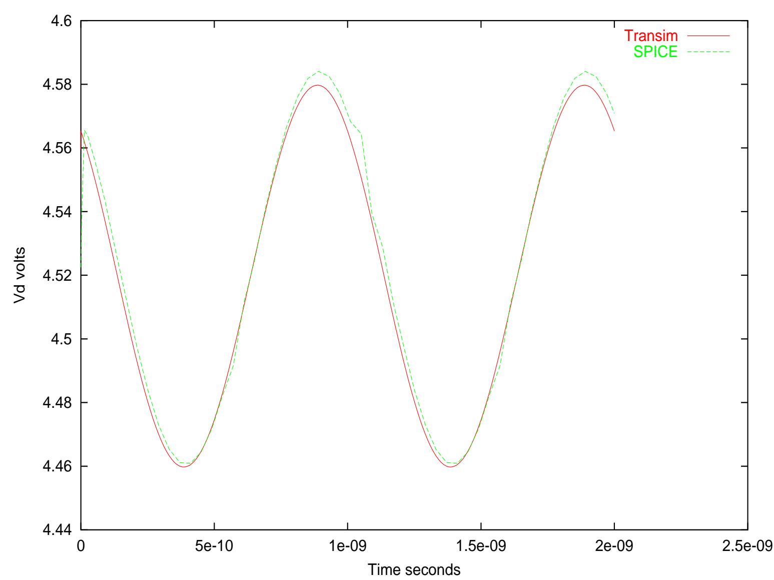


Figure 4.9: Transient Analysis comparison at $V_{\mathrm{GS}} = 1.7\mathrm{V}$

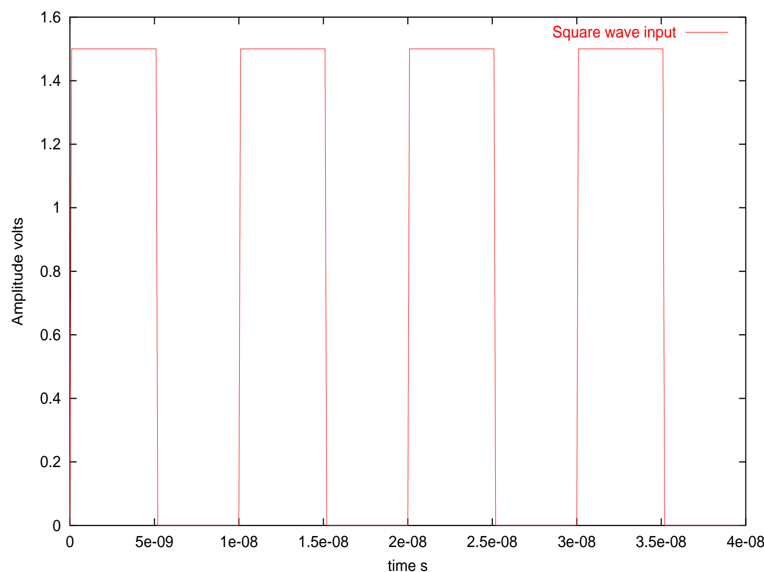


Figure 4.10: Square wave input 0.1 GHz

### 4.2.2 Square wave input

A square wave at a frequency of 0.1 GHz is used to drive the transistor. The square wave, as mentioned earlier, has a rise and fall time of 0.2 nsec and a peak value of 1.5 volts DC. It is shown in Figure 4.10. The SPICE3 output is shown in Figure 4.11.

Using the same circuit and same operating conditions, Transim's output is displayed in Figure 4.12. For both the simulators, the time step used is 0.01 nano-seconds.

The output in both the case of SPICE3 and Transim exhibits spikes which correspond to the voltage transitions of the input waveform. The magnitude of these spikes exceeds the value of the supply voltage which is unphysical and incorrect. At higher frequencies, the BSIM4 model in SPICE3 exhibits wider spikes at the input voltage transition points. When the input frequency

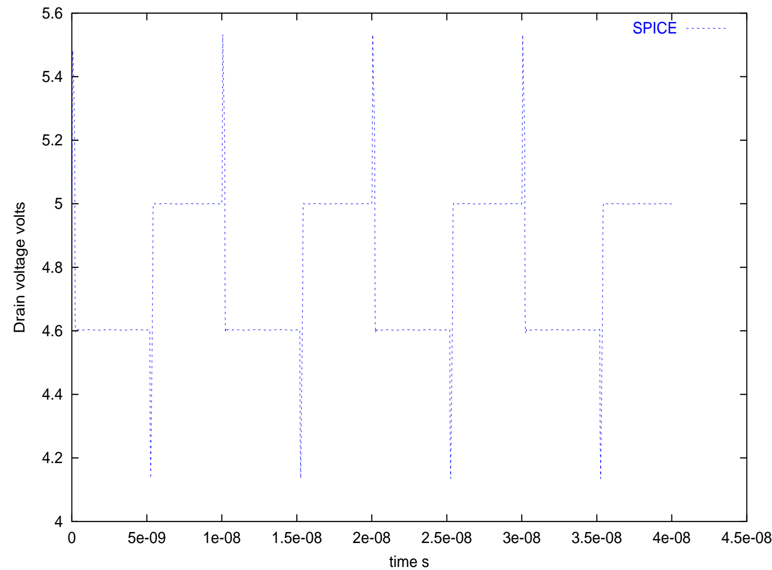


Figure 4.11: Transient Analysis SPICE3 result for a square pulse input at 0.1GHz

is increased to 1 GHz, the output produced by SPICE3 is shown in Figure 4.13. The corresponding result in Transim is shown in Figure 4.14.
A slight modification is made in the model in Transim wherein the derivative of the drain charge was removed and the result obtained is shown in Figure 4.15. This simulation is run using a 1 GHz square wave and the result is without spikes. It correctly predicts the clipping of the output at the supply voltage.

## 4.3 Conclusions

Though the comparison curves for the DC analysis and the transient analysis match fairly closely, there are still some discrepancies between the results in SPICE3 and Transim. SPICE3 uses Associated Discrete Modeling (ADM)

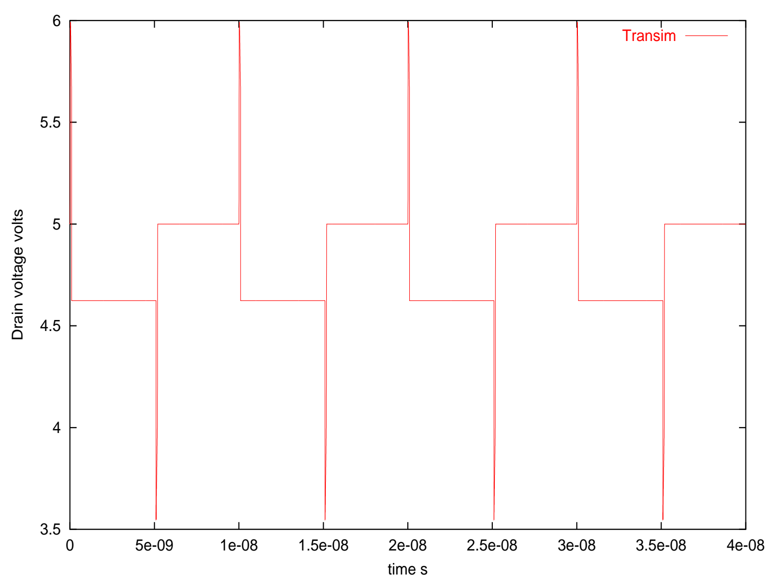


Figure 4.12: Transient Analysis Transim result for a square pulse input at 0.1 GHz

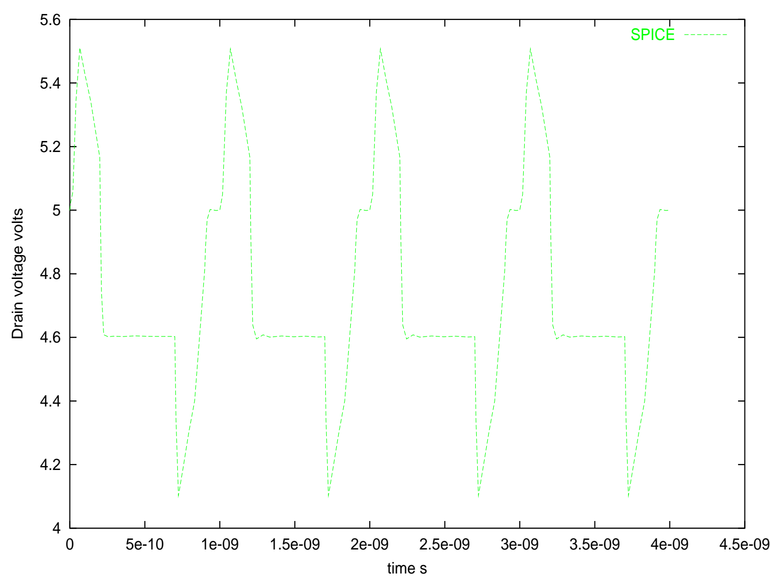


Figure 4.13: Transient Analysis SPICE3 result for a square pulse input at 1 GHz

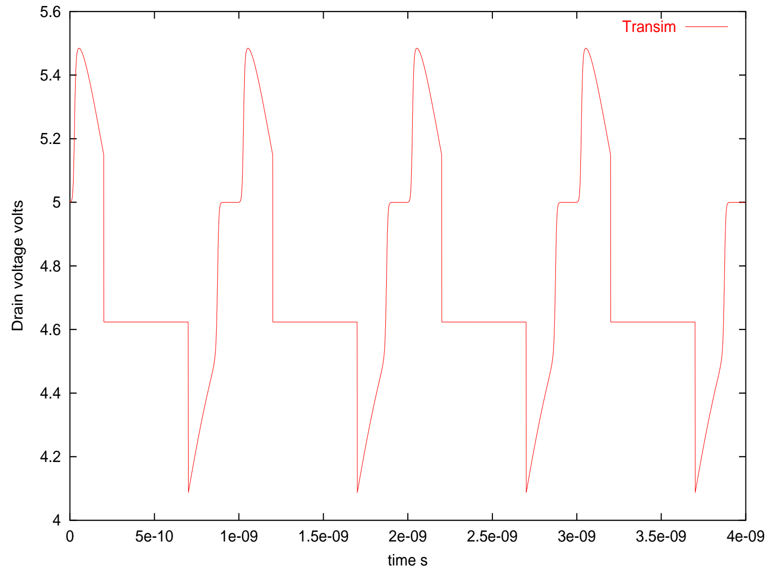


Figure 4.14: Transient Analysis Transim result for a square pulse input at 1 GHz

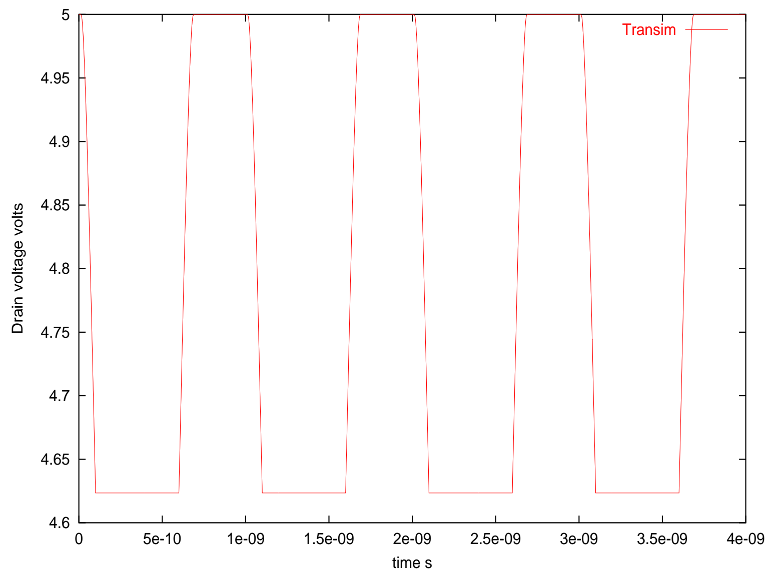


Figure 4.15: Transient Analysis Transim result excluding the drain charge derivative

where a time discretization step is applied to the equations describing the element characteristics. The non-linear differential equations are converted to non-linear algebraic equations. These non-linear algebraic equations are solved iteratively using Kirchoff's voltage and current laws. These associated discrete models are also called companion models. The differential equations describing the circuit elements (constitutive relations) are modeled as finite difference equations and solved iteratively at each time point. Thus, the differential equations describing circuit elements are approximated by resistive circuits associated with the numerical integration algorithm.

In SPICE3, the derivatives of the variables with respect to the input voltages are manually derived in the code for the element as are the solvers for the non-linear system of equations evaluating the companion models associated with the element. This means, that keeping track of the derivatives during a simulation run or searching for errors due to derivative discontinuities becomes a very strenuous task. Transim, on the other hand, as described in Chapter 2, has an object oriented structure which enables it to use off-the-shelf numerical calculators to evaluate these derivatives and solve the non-linear system of equations. These routines are distinctly separate from the element code making the process of distinguishing errors relatively simpler. Hence, the difference in the evaluation of these companion models could be a major reason for the discrepancy in the results between SPICE3 and Transim.

As was shown the transient analysis results with a square wave input, there were spikes in the output voltage waveform corresponding to the input voltage transitions. This suggests that there is a discontinuous variable or

derivative in the model. After making a small change to Transim by removing the effect of the first derivative of drain charge, the spikes disappear and the output is correctly clipped at the supply voltage level. It is possible that there is a discontinuity in the drain charge and it's first derivative.

# Chapter 5

# Conclusions and Future Research

## 5.1 Conclusions

SPICE3's BSIM4 captures many of the effects that show up due to smaller and narrower channels. It models body effects, channel length modulation, effects due to pocket implants, mobility degradation due to vertical fields, drain induced barrier lowering, impact ionization current, GIDL current, charge sharing between the source and drain, velocity saturation and bias dependent channel resistance, to name a few. It also uses the continuous equations for bias dependent current and charge that are valid across all regimes of operation. It models effects that have never been modeled in SPICE FETs before BSIM4 such as GIDL current and gate tunnelling currents that flow between the gate and source, drain and bulk regions respectively for narrow width devices.

The Transim implementation of the BSIM4 model is a faithful implementation of the BSIM4v3.2.1 model and models all of the effects mentioned

above. The entire source code in Transim is approximately 25 pages long with 1500 lines of code and the entire element is contained in a single C++ file. It took a little under seven months to develop. SPICE3, on the other hand, requires 23 files of source code in C and many thousands of line of code. A major advantage is that Transim uses ADOL-C [11], an off-the-shelf automatic differentiation package that works independently of the element and calculates and keeps track of the derivatives during run time with a minimum of truncation error. This enables easier model writing and maintenance. This was shown in the previous chapter in the case of the voltage spikes produced at the output, where it was easy to debug and isolate errors. In Transim, all the analysis types, elements and numerical solvers are treated as objects with a high level of abstraction. In contrast, SPICE3 includes the solvers, derivatives and the analysis types in the element code itself. This makes it more difficult to maintain and check.

## 5.2 Future Research

In spite of the above, there is still some work to be done to make it a truly complete model. Some of the steps that need to be taken include

- Formulation of the PMOS model. This will enable the testing of digital circuits such as inverters and gates. A thorough testing procedure using benchmark circuits can also be carried out.

- Including temperature as a state variable and making this an electro-thermal model. This will model the behavior of the transistor at various

temperatures. This is very important for various Analog/RF applications such as amplifier design.

- Providing support for Harmonic Balance. It can then become the first third-generation transistor to run with Harmonic Balance.

- Support for noise analysis needs to be included to enable accurate modeling of high frequency analog applications.

- Inclusion of reverse biased diode breakdown currents. On application of high reverse bias, the intrinsic diodes of the transistor break down and a very large current flows. That effect needs to be modeled as well.

# Appendix A

# MOSFET Model Source code

This section contains the C++ code of the representation of the model.

## A.1 C++ code

```
#include "../network/CircuitManager.h"
#include"../network/Element.h"
#include"../network/NAdolcElement.h"
#include "Mos4.h"

//  Static members
const unsigned Mos4::n_par = 144;

//  Element information
ItemInfo Mos4::einfo = {
  "mos4",
  "MOSFET using BSIM4 model",
  "Nikhil Kriplani",
  DEFAULT_ADDRESS"elements/Mos4.h.html"
};

 ParmInfo Mos4::pinfo[] = {
  {"TOXE", "Electrical gate equivalent oxide thickness", TR_DOUBLE, false},
  {"TOXP", "Physical gate equivalent oxide thickness", TR_DOUBLE, false},
  {"EPSROX", "Gate dielectric constant relative to vacuum", TR_DOUBLE, false},
  {"VFB", "Flat-band voltage", TR_DOUBLE, false},
  {"VTH0", "Long-channel threshold voltage", TR_DOUBLE, false},
  {"NGATE", "Poly Si gate doping concentration", TR_DOUBLE, false},
  {"XL", "Channel length offset due to mask/etch effect", TR_DOUBLE, false},
  {"XW", "Channel width offset due to mask/etch effect", TR_DOUBLE, false},
  {"NF", "Number of device fingers", TR_DOUBLE, false},
  {"W", "Width of the device", TR_DOUBLE, false},
  {"L", "Length of the device", TR_DOUBLE, false},
  {"DWG", "Coefficient of gate bias dependence of Weff", TR_DOUBLE, false},
  {"DWB", "Coefficient of body bias dependence of Weff", TR_DOUBLE, false},
  {"WINT", "Channel-width offset parameter", TR_DOUBLE, false},
```

```
{"WLN", "Power of length dependence of width offset", TR_DOUBLE, false},
{"WL", "Coefficient of length dependence for width offset", TR_DOUBLE, false},
{"WWN", "Power of width dependence of width offset", TR_DOUBLE, false},
{"WW", "Coefficient of width dependence for width offset", TR_DOUBLE, false},
{"WWL", "Coefficient of length and width cross term dependence for width offset",
         TR_DOUBLE, false},
{"LINT", "Channel-length offset parameter", TR_DOUBLE, false},
{"LLN", "Power of length dependence for length offset", TR_DOUBLE, false},
{"LL", "Coefficient of length dependence for length offset", TR_DOUBLE, false},
{"LW", "Coefficient of width dependence for length offset", TR_DOUBLE, false},
{"LWN", "Power of width dependence for length offset", TR_DOUBLE, false},
{"LWL", "Coefficient of length and width cross term dependence for length
         offset", TR_DOUBLE, false},
{"K1", "First-order body bias coefficient", TR_DOUBLE, false},
{"K2", "Second-order body bias coefficient", TR_DOUBLE, false},
{"LPEB", "Lateral non-uniform doping effect on K1", TR_DOUBLE, false},
{"LPE0", "Lateral non-uniform doping parameter at Vbs=0", TR_DOUBLE, false},
{"K3", "Narrow width coefficient", TR_DOUBLE, false},
{"K3B", "Body effect coefficient of K3", TR_DOUBLE, false},
{"W0", "Narrow width parameter", TR_DOUBLE, false},
{"DVT0W", "First coefficient of narrow width effect on Vth for small channel
           length", TR_DOUBLE, false},
{"DVT0", "First coefficient of short-channel effect on Vth", TR_DOUBLE, false},
{"DVT1W", "Second coefficient of narrow width effect on Vth for small channel
           length", TR_DOUBLE, false},
{"DVT1", "Second coefficient of short-channel effect on Vth", TR_DOUBLE, false},
{"DSUB", "DIBL coefficient exponent in sub-threshold region", TR_DOUBLE, false},
{"ETA0", "DIBL coefficient in sub-threshold region", TR_DOUBLE, false},
{"ETAB", "Body-bias coefficient for the sub-threshold region", TR_DOUBLE, false},
{"TOXM", "Tox at which parameters are extracted", TR_DOUBLE, false},
{"T", "Temperature", TR_DOUBLE, false},
{"NDEP", "Channel doping concentration at depletion edge for zero body bias",
          TR_DOUBLE, false},
{"PHIN", "Non-uniform vertical doping effect on surface potential", TR_DOUBLE,
         false},
{"VBM", "Maximum applied body bias in VTH0 calculation", TR_DOUBLE, false},
{"NSUB", "Substrate doping concentration", TR_DOUBLE, false},
{"DVT2W", "Body-bias coefficient of narrow width effect for small channel length",
          TR_DOUBLE, false},
{"NSD", "Source/drain doping concentration fatal error if not positive",
         TR_DOUBLE, false},
{"DVT2", "Body-bias coefficient of short-channel effect on Vth", TR_DOUBLE, false},
{"MINV", "Vgsteff fitting parameter for moderate inversion condition",
          TR_DOUBLE, false},
{"NFACTOR", "Subthreshold swing factor", TR_DOUBLE, false},
{"CDSC", "Coupling capacitance between source/drain and channel", TR_DOUBLE,
         false},
{"CDSCD", "Drain-bias sensitivity of CDSC", TR_DOUBLE, false},
{"CDSCB", "Body-bias sensitivity of CDSC", TR_DOUBLE, false},
{"CIT", "Interface trap capacitance", TR_DOUBLE, false},
{"KETA", "Body-bias coefficient of bulk charge effect", TR_DOUBLE, false},
{"B0", "Bulk charge effect coefficient for channel width", TR_DOUBLE, false},
{"B1", "Bulk charge effect width offset", TR_DOUBLE, false},
{"A0", "Coefficient of channel-length dependence bulk charge effect",
        TR_DOUBLE, false},
{"AGS", "Coefficient of Vgs dependence of bulk charge effect", TR_DOUBLE, false},
{"XJ", "S/D junction depth", TR_DOUBLE, false},
```

```
{"U0", "Low-field mobility", TR_DOUBLE, false},
{"UA", "Coefficient of first-order mobility degradation due to vertical field",
        TR_DOUBLE, false},
{"UB", "Coefficient of second-order mobility degradation due to vertical field",
        TR_DOUBLE, false},
{"UC", "Coefficient of mobility degradation due to body-bias effect", TR_DOUBLE,
        false},
{"EU", "Exponent for mobility degradation", TR_DOUBLE, false},
{"DELTA", "Parameter for DC Vdseff", TR_DOUBLE, false},
{"PDITS", "Impact of drain-induced Vth shift on Rout", TR_DOUBLE, false},
{"FPROUT", "Effect of pocket implant on Rout degradation", TR_DOUBLE, false},
{"PDITSL", "Channel-length dependence of drain-induced Vth shift for Rout",
            TR_DOUBLE, false},
{"PDITSD", "Vds dependence of drain-induced Vth shift for Rout", TR_DOUBLE,
            false},
{"PSCBE2", "Second substrate current induced body-effect parameter", TR_DOUBLE,
            false},
{"PSCBE1", "First substrate current induced body-effect parameter", TR_DOUBLE,
           false},
{"PDIBLCB", "Body bias coefficient of DIBL effect on Rout", TR_DOUBLE, false},
{"PVAG", "Gate-bias dependence of Early voltage", TR_DOUBLE, false},
{"PDIBL1", "Parameter for DIBL effect on Rout", TR_DOUBLE, false},
{"PDIBL2", "Parameter for DIBL effect on Rout", TR_DOUBLE, false},
{"DROUT", "Channel-length dependence of DIBL effect on Rout", TR_DOUBLE, false},
{"PCLM", "Channel length modulation parameter", TR_DOUBLE, false},
{"A1", "First non-saturation effect parameter", TR_DOUBLE, false},
{"A2", "Second non-saturation factor", TR_DOUBLE, false},
{"RDSWMIN", "Lightly-doped drain resistance per unit width at high Vgs and
             zero Vbs", TR_DOUBLE, false},
{"RDSW", "Zero bias lightly-doped drain resistance per unit width", TR_DOUBLE,
          false},
{"PRWG", "Gate-bias dependence of LDD resistance", TR_DOUBLE, false},
{"PRWB", "Body-bias dependence of LDD resistance", TR_DOUBLE, false},
{"WR", "Channel-width dependence parameter of LDD resistance", TR_DOUBLE, false},
{"WLC", "Coefficient of length dependence for CV channel width offset", TR_DOUBLE,
         false},
{"WWC", "Coefficient of width dependence for CV channel width offset", TR_DOUBLE,
         false},
{"WWLC", "Coefficient of length and width crossterm dependence for CV channel width
         offset", TR_DOUBLE, false},
{"DWJ", "Offset of the S/D junction width", TR_DOUBLE, false},
{"CLC", "Constant term for the short channel model", TR_DOUBLE, false},
{"CLE", "Exponential term for the short channel model", TR_DOUBLE, false},
{"NOFF", "CV parameter in VgsteffCV for weak to strong inversion", TR_DOUBLE,
          false},
{"VOFFCV", "CV parameter in VgsteffCV for weak to strong inversion", TR_DOUBLE,
          false},
{"CF", "Fringing field capacitance", TR_DOUBLE, false},
{"CKAPPAD", "Coefficient of bias-dependent overlap capacitance for the drain side",
             TR_DOUBLE, false},
{"CKAPPAS", "Coefficient of bias-dependent overlap capacitance for the source side",
             TR_DOUBLE, false},
{"LLC", "Coefficient of length dependence on CV channel length offset", TR_DOUBLE,
        false},
{"LWC", "Coefficient of width dependence on CV channel length offset", TR_DOUBLE,
        false},
{"LWLC", "Coefficient of length and width cross term dependence on CV channel length
```

```
            offset", TR_DOUBLE, false},
  {"WWLC", "Coefficient of length and width cross term dependence on CV channel width
            offset", TR_DOUBLE, false},
  {"VOFF", "Offset voltage in the subthreshold region for large W and L", TR_DOUBLE,
            false},
  {"VOFFL", "Channel length dependence of VOFF", TR_DOUBLE, false},
  {"POXEDGE", "Factor for the gate oxide thickness in the S/D overlap regions",
               TR_DOUBLE, false},
  {"TOXREF", "Nominal gate oxide thickness for gate dielectric tunneling current
              model", TR_DOUBLE, false},
  {"NTOX", "Exponent for the gate oxide ratio", TR_DOUBLE, false},
  {"DLCIG", "Source/drain overlap length for Igs and Igd", TR_DOUBLE, false},
  {"AIGSD", "parameter for Igs and Igd", TR_DOUBLE, false},
  {"BIGSD", "parameter for Igs and Igd", TR_DOUBLE, false},
  {"CIGSD", "parameter for Igs and Igd", TR_DOUBLE, false},
  {"MOIN", "Coefficient for gate-bias dependent surface potential", TR_DOUBLE, false},
  {"VSAT", "Saturation velocity", TR_DOUBLE, false},
  {"PDITSD", "Vds dependence of drain-induced Vth shift for Rout", TR_DOUBLE, false},
  {"AIGC", "Parameter for Igcs and Igcd", TR_DOUBLE, false},
  {"BIGC", "Parameter for Igcs and Igcd", TR_DOUBLE, false},
  {"CIGC", "Parameter for Igcs and Igcd", TR_DOUBLE, false},
  {"NIGC", "Parameter for Igcs, Igcd, Igs and Igd", TR_DOUBLE, false},
  {"PIGCD", "Vds dependence of Igcs and Igcd", TR_DOUBLE, false},
  {"DVTP0", "First coefficient of drain induced Vth shift due to long channel pocket
             devices", TR_DOUBLE, false},
  {"DVTP1", "First coefficient of drain induced Vth shift due to long channel pocket
             devices", TR_DOUBLE, false},
  {"PRT", "Temperature coefficient for RDSW", TR_DOUBLE, false},
  {"AT", "Temperature cpefficient for saturation velocity", TR_DOUBLE, false},
  {"XT", "Doping Depth", TR_DOUBLE, false},
  {"ALPHA0", "First parameter of impact ionization current", TR_DOUBLE, false},
  {"ALPHA1", "Isub parameter for length scaling", TR_DOUBLE, false},
  {"BETA0", "Second parameter of impact ionization current", TR_DOUBLE, false},
  {"AGIDL", "Pre-exponential coefficient for GIDL", TR_DOUBLE, false},
  {"BGIDL", "Exponential coefficient for GIDL", TR_DOUBLE, false},
  {"CGIDL", "Parameter for body-bias effect on GIDL", TR_DOUBLE, false},
  {"EGIDL", "Fitting parameter for band-bending for GIDL", TR_DOUBLE, false},
  {"ACDE", "Exponential coefficient for charge thickness", TR_DOUBLE, false},
  {"DLC", "Channel length offset parameter", TR_DOUBLE, false},
  {"DWC", "Channel width offset parameter", TR_DOUBLE, false},
  {"AIGBACC", "Parameter for Igb in accumulation", TR_DOUBLE, false},
  {"BIGBACC", "Parameter for Igb in accumulation", TR_DOUBLE, false},
  {"CIGBACC", "Parameter for Igb in accumulation", TR_DOUBLE, false},
  {"NIGBACC", "Parameter for Igb in accumulation", TR_DOUBLE, false},
  {"AIGBINV", "Parameter for Igb in inversion", TR_DOUBLE, false},
  {"BIGBINV", "Parameter for Igb in inversion", TR_DOUBLE, false},
  {"CIGBINV", "Parameter for Igb in inversion", TR_DOUBLE, false},
  {"EIGBINV", "Parameter for Igb in inversion", TR_DOUBLE, false},
  {"NIGBINV", "Parameter for Igb in inversion", TR_DOUBLE, false},
  {"KT1", "Temperature coefficient for threshold voltage", TR_DOUBLE, false},
  {"KT1l", "Channel Length dependence of KT1", TR_DOUBLE, false},
  {"KT2", "Body-bias coefficient of Vth with temperature effects", TR_DOUBLE, false}
};

Mos4::Mos4(const string& iname) : NAdolcElement(&einfo, pinfo,
n_par, iname) {
  // The default parameter values
```

```
paramvalue[0] = &(TOXE = 3.0e-9);
paramvalue[1] = &(TOXP = TOXE);
paramvalue[2] = &(EPSROX = 3.9);
paramvalue[3] = &(VFB = -1.0);
paramvalue[4] = &(VTH0 = 0.7);
paramvalue[5] = &(NGATE = 0.0);
paramvalue[6] = &(XL = 0.0);
paramvalue[7] = &(XW = 0.0);
paramvalue[8] = &(NF = 1.0);
paramvalue[9] = &(W = 5.0e-6);
paramvalue[10] = &(L = 5.0e-6);
paramvalue[11] = &(DWG = 0.0);
paramvalue[12] = &(DWB = 0.0);
paramvalue[13] = &(WINT = 0.0);
paramvalue[14] = &(WLN = 1.0);
paramvalue[15] = &(WL = 0.0);
paramvalue[16] = &(WWN = 1.0);
paramvalue[17] = &(WW = 0.0);
paramvalue[18] = &(WWL = 0.0);
paramvalue[19] = &(LINT = 0.0);
paramvalue[20] = &(LLN = 1.0);
paramvalue[21] = &(LL = 0.0);
paramvalue[22] = &(LW = 0.0);
paramvalue[23] = &(LWN = 1.0);
paramvalue[24] = &(LWL = 0.0);
paramvalue[25] = &(K1 = 0.53);
paramvalue[26] = &(K2 = -0.0186);
paramvalue[27] = &(LPEB = 0.0);
paramvalue[28] = &(LPE0 = 1.74e-7);
paramvalue[29] = &(K3 = 80.0);
paramvalue[30] = &(K3B = 0.0);
paramvalue[31] = &(W0 = 2.5e-6);
paramvalue[32] = &(DVT0W = 0.0);
paramvalue[33] = &(DVT0 = 2.2);
paramvalue[34] = &(DVT1W = 5.3e6);
paramvalue[35] = &(DVT1 = 0.53);
paramvalue[36] = &(DSUB = 0.56);
paramvalue[37] = &(ETA0 = 0.08);
paramvalue[38] = &(ETAB = -0.07);
paramvalue[39] = &(TOXM = TOXE);
paramvalue[40] = &(T = 300.0);
paramvalue[41] = &(NDEP = 1.7e17);
paramvalue[42] = &(PHIN = 0.0);
paramvalue[43] = &(VBM = -3.0);
paramvalue[44] = &(NSUB = 6.0e16);
paramvalue[45] = &(DVT2W = -0.032);
paramvalue[46] = &(NSD = 1.0e20);
paramvalue[47] = &(DVT2 = -0.032);
paramvalue[48] = &(MINV = 0.0);
paramvalue[49] = &(NFACTOR = 1.0);
paramvalue[50] = &(CDSC = 2.4e-4);
paramvalue[51] = &(CDSCD = 0.0);
paramvalue[52] = &(CDSCB = 0.0);
paramvalue[53] = &(CIT = 0.0);
paramvalue[54] = &(KETA = -0.047);
paramvalue[55] = &(B0 = 0.0);
paramvalue[56] = &(B1 = 0.0);
```

```
paramvalue[57] = &(A0 = 1.0);
paramvalue[58] = &(AGS = 0.0);
paramvalue[59] = &(XJ = 1.5e-7);
paramvalue[60] = &(U0 = 0.067);
paramvalue[61] = &(UA = 1.0e-15);
paramvalue[62] = &(UB = 1.0e-19);
paramvalue[63] = &(UC = -0.0465e-9);
paramvalue[64] = &(EU = 1.67);
paramvalue[65] = &(DELTA = 0.01);
paramvalue[66] = &(PDITS = 0.0);
paramvalue[67] = &(FPROUT = 0.0);
paramvalue[68] = &(PDITSL = 0.0);
paramvalue[69] = &(PDITSD = 0.0);
paramvalue[70] = &(PSCBE2 = 1.0e-5);
paramvalue[71] = &(PSCBE1 = 4.24e8);
paramvalue[72] = &(PDIBLCB = 0.0);
paramvalue[73] = &(PVAG = 0.0);
paramvalue[74] = &(PDIBL1 = 0.0);
paramvalue[75] = &(PDIBL2 = 0.0);
paramvalue[76] = &(DROUT = 0.56);
paramvalue[77] = &(PCLM = 1.3);
paramvalue[78] = &(A1 = 0.0);
paramvalue[79] = &(A2 = 1.0);
paramvalue[80] = &(RDSWMIN = 0.0);
paramvalue[81] = &(RDSW = 200.0);
paramvalue[82] = &(PRWG = 1.0);
paramvalue[83] = &(PRWB = 0.0);
paramvalue[84] = &(WR = 1.0);
paramvalue[85] = &(WLC = WL);
paramvalue[86] = &(WWC = WW);
paramvalue[87] = &(WWLC = WWL);
paramvalue[88] = &(DWJ = WINT);
paramvalue[89] = &(CLC = 1.0e-7);
paramvalue[90] = &(CLE = 0.6);
paramvalue[91] = &(NOFF = 1.0);
paramvalue[92] = &(VOFFCV = 0.0);
paramvalue[93] = &(CF = 1.08e-10);
paramvalue[94] = &(CKAPPAD = 0.6);
paramvalue[95] = &(CKAPPAS = 0.6);
paramvalue[96] = &(LLC = 0.0);
paramvalue[97] = &(LWC = 0.0);
paramvalue[98] = &(LWLC = 0.0);
paramvalue[99] = &(WWLC = 0.0);
paramvalue[100] = &(VOFF = -0.08);
paramvalue[101] = &(VOFFL = 0.0);
paramvalue[102] = &(POXEDGE = 1.0);
paramvalue[103] = &(TOXREF = TOXE);
paramvalue[104] = &(NTOX = 1.0);
paramvalue[105] = &(DLCIG = LINT);
paramvalue[106] = &(AIGSD = 0.43);
paramvalue[107] = &(BIGSD = 0.054);
paramvalue[108] = &(CIGSD = 0.075);
paramvalue[119] = &(MOIN = 15.0);
paramvalue[110] = &(VSAT = 8.0e4);
paramvalue[111] = &(PDITSD = 0.0);
paramvalue[112] = &(AIGC = 0.43);
paramvalue[113] = &(BIGC = 0.054);
```

```
  paramvalue[114] = &(CIGC = 0.075);
  paramvalue[115] = &(NIGC = 1.0);
  paramvalue[116] = &(PIGCD = 1.0);
  paramvalue[117] = &(DVTP0 = 0.0);
  paramvalue[118] = &(DVTP1 = 0.0);
  paramvalue[119] = &(PRT = 0.0);
  paramvalue[120] = &(AT = 3.3e4);
  paramvalue[121] = &(XT = 1.55e-7);
  paramvalue[122] = &(ALPHA0 = 0.0);
  paramvalue[123] = &(ALPHA1 = 0.0);
  paramvalue[124] = &(BETA0 = 30.0);
  paramvalue[125] = &(AGIDL = 0.0);
  paramvalue[126] = &(BGIDL = 2.3e9);
  paramvalue[127] = &(CGIDL = 0.5);
  paramvalue[128] = &(EGIDL = 0.8);
  paramvalue[129] = &(ACDE = 1.0);
  paramvalue[130] = &(DLC = LINT);
  paramvalue[131] = &(DWC = WINT);
  paramvalue[132] = &(AIGBACC = 0.43);
  paramvalue[133] = &(BIGBACC = 0.054);
  paramvalue[134] = &(CIGBACC = 0.075);
  paramvalue[135] = &(NIGBACC = 1.0);
  paramvalue[136] = &(AIGBINV = 0.35);
  paramvalue[137] = &(BIGBINV = 0.03);
  paramvalue[138] = &(CIGBINV = 0.006);
  paramvalue[139] = &(EIGBINV = 1.1);
  paramvalue[140] = &(NIGBINV = 3.0);
  paramvalue[141] = &(KT1 = -0.11);
  paramvalue[142] = &(KT1l = 0.0);
  paramvalue[143] = &(KT2 = 0.022);

  // Set flags
  setFlags(NONLINEAR | ONE_REF | TR_TIME_DOMAIN);
}

  void Mos4::init() throw(string&) {

  // Set the number of terminals (drain, gate, source and bulk)
  setNumTerms(4);

  //Set number of state variables
  setNumberOfStates(3);

  //Set number of levels
  setNlevels(2);

  // create tape
  IntVector var(3);
  var[0] = 0;
  var[1] = 1;
  var[2] = 2;
  IntVector novar;
  DoubleVector nodelay;
  createTape1(var, novar, nodelay, 3, 6);
  createTape2(0, 6);
}
```

```
  void Mos4::eval1(adoublev& x, adoublev& xt, adoublev& y1, adoublev& z1)
{
  //state variables
  //vds x[0] (drain-source), vgs x[1] (gate-source), vbs x[2]
  //(bulk-source)

  adouble lt, Vbseff, Cdsc_term, n, Theta0, Vbc, Phis, Weff;
  adouble ltw, Vth, lambda, ueff, Esat, Xdep, sqrtPhis;
  adouble Vgse, Vgsteff, Vb, Abulk, F_doping, Igidl;
  adouble Rds, Vdsat, Vdseff, Vasat, Ids0, Ids, theta_rout;
  adouble Vadits, Vadibl, Cclm, Vaclm, Va, Vascbe;
  adouble T0, T1, T2, T3, T4, T5, T6, T7, T8, T9, T10, T11, T12, Isub;
  double t0, t1, t2, t3;

  #define MAX_EXP 5.834617425e14
  #define MIN_EXP 1.713908431e-15
  #define EXP_THRESHOLD 34.0


  double k = kBoltzman; //Boltzman's constant
  double q = eCharge; //electron charge

  //threshold voltage
  double vt = k*T/q;

  //band-gap energy
  double Eg0 = 1.16 - 7.02e-4 * T *T / (T + 1108.0);

  //Si intrinsic concentration
  double ni = 1.45e10 * (T/300.15) * sqrt(T/300.15)
              * exp(21.5565981 - Eg0/(2.0 * vt));

  double e0 = epsilon0;
  double esi = 11.7 * e0;

  //electrical oxide capacitance
  double coxe = EPSROX * (e0/TOXE);
  //physical oxide capacitance
  double coxp = EPSROX * (e0/TOXP);

  double Lnew = L + XL;
  double Wnew = W/NF + XW;
  t0 = pow(Lnew, LLN);
  t1 = pow(Wnew, LWN);
  double DL = LINT + LL/t0 + LW/t1 + LWL/(t0*t1);
  DLC += LLC/t0 + LWC/t1 + LWLC/(t0*t1);
  DLCIG += LLC/t0 + LWC/t1 + LWLC/(t0*t1);

  t2 = pow(Lnew, WLN);
  t3 = pow(Wnew,WWN);
  double DW = WINT + WL/t2 + WW/t3 + WWL/(t2*t3);
  DWC += WLC/t2 + WWC/t3 + WWLC/(t2*t3);
  DWJ += WLC/t2 + WWC/t3 + WWLC/(t2*t3);

  //Effective channel Length
  double Leff = Lnew - 2.0 * DL;
  //Effective channel width
```

```
Weff = Wnew - 2.0 * DW;

double Weffcj = Wnew - 2.0 * DWJ;

//surface potential
double phi_s = 0.4 + (k*T/q)*log(NDEP/ni) + PHIN;

double m = 0.5 + atan(MINV)/pi;

double Voff_prime = VOFF + VOFFL/Leff;

//Built - in potential
double Vbi = (k*T/q)*log(NDEP*NSD/pow(ni,2.0));

double Xdep0 = sqrt(2.0 * esi / (eCharge * NDEP * 1.0e6))
               * sqrt(phi_s);

double lt0 = sqrt(esi * TOXE * Xdep0 / (e0*EPSROX));

double litl = sqrt(3.0 * XJ * TOXE);

double gamma1 = 5.753e-12 * sqrt(NDEP)/coxe;
double gamma2 = 5.753e-12 * sqrt(NSUB)/coxe;

double VBX = phi_s - 7.7348e-4 * NDEP * XT * XT;
if (VBX > 0.0)
  VBX = -VBX;

t0 = gamma1 - gamma2;
t1 = sqrt(phi_s - VBX) - sqrt(phi_s);
t2 = sqrt(phi_s * (phi_s - VBM)) - phi_s;

K2 = t0 * t1 / (2.0 * t2 + VBM);
K1 = gamma2 - 2.0 * K2 * sqrt(phi_s - VBM);

VTH0 = VFB + phi_s + K1 * sqrt(phi_s);

double K1ox = K1 * TOXE / TOXM;
double K2ox = K2 * TOXE / TOXM;

//---------Calculation for Vbc--------
if (K2 < 0.0)
  {
    Vbc = 0.9 * (phi_s - pow(0.5 * K1/K2,2.0));
    if (Vbc > -3.0)
  Vbc = -3.0;
    else if (Vbc < -30.0)
  Vbc = -30.0;
  }
else
  Vbc = -30.0;
//--------end Vbc-----------------

//Effective Bulk Source voltage
//------Calculation for Vbseff-----------
T0 = x[2] - Vbc - 0.001;
T1 = sqrt(T0*T0 - 0.004 * Vbc);
```

```
if (T0 >= 0.0)
  Vbseff = Vbc + 0.5 * (T0 + T1);
else
  {
    T2 = -0.002 / (T1 - T0);
    Vbseff = Vbc * (1.0 + T2);
  }
if (Vbseff < x[2])
  Vbseff = x[2];
//------end Vbseff----------------------

//-----Calculation for Phis---------------
if (Vbseff > 0.0)
  {
    Phis = phi_s * phi_s / (phi_s + Vbseff);
    sqrtPhis = sqrt(phi_s) * phi_s / (phi_s + 0.5*Vbseff);
  }
else
  {
    Phis = phi_s - Vbseff;
    sqrtPhis = sqrt(Phis);
  }
//------end Phis--------------------------

//-----Calculation for Xdep---------------
Xdep = Xdep0 * sqrtPhis / sqrt(phi_s);
//-----end Xdep-------------------------

//Threshold voltage calculation
//-------Calculation for Vth------------
if ((DVT2W * Vbseff) >= -0.5)
  T1 = 1.0 + DVT2W * Vbseff;
else
  T1 = (1.0 + 3.0 * DVT2W*Vbseff) / (3.0 + 8.0
        * DVT0W * Vbseff);

ltw = sqrt(Xdep) * T1 * sqrt(esi/(EPSROX * e0) * TOXE);

if ((DVT2 * Vbseff) >= -0.5)
  T1 = 1.0 + DVT2 * Vbseff;
else
  T1 = (1.0 + 3.0 * DVT2 * Vbseff) / (3.0 + 8.0
        * DVT2 * Vbseff);

ltw = sqrt(Xdep) * T1 * sqrt(esi/(EPSROX * e0) * TOXE);

if ((DVT1 * Leff/lt) < EXP_THRESHOLD)
  {
    T1 = exp(DVT1 * Leff/lt);
    T2 = T1 - 1.0;
    T3 = T2 * T2;
    T4 = T3 + 2.0 * T1 * MIN_EXP;
    Theta0 = T1/T4;
  }
else
  Theta0 = 1.0 / (MAX_EXP - 2.0);
```

```
if ((DVT1W * Leff * Weff/ltw) < EXP_THRESHOLD)
  {
    T1 = exp(DVT1W * Weff * Leff/ltw);
    T2 = T1 - 1.0;
    T3 = T2 * T2;
    T4 = T3 + 2.0 * T1 * MIN_EXP;
    T5 = T1/T4;
  }
else
  T5 = 1.0/(MAX_EXP - 2.0);

T0 = DVT0W * T5;
T2 = (Vbi - phi_s) * T0;

T0 = sqrt(1.0 + LPE0/Leff);
T1 = K1ox * (T1 - 1.0) * sqrtPhis + (KT1 + KT1l/Leff + KT2*Vbseff)
     * (T/300.0 - 1.0);

T8 = TOXE * phi_s /(Weff + W0);

T6 = sqrt(esi / (EPSROX * e0) * TOXE * Xdep0);
T0 = DSUB * Leff / T6;
if (T0 < EXP_THRESHOLD)
  {
    T1 = exp(T0);
    T2 = T1 - 1.0;
    T3 = T2 * T2;
    T4 = T3 + 2.0 * T1 * MIN_EXP;
    T5 = T1 / T4;
  }
else
  T5 = 1.0 / (MAX_EXP - 2.0);

T3 = ETA0 + ETAB * Vbseff;

if (T3 < 1.0e-4)
  T3 = (2.0e-4 - T3) / (3.0 - 2.0e4 * T3);
else
  T3 = ETA0 + ETAB * Vbseff;

T7 = T3 * T5 * x[0];


Vth = VTH0
  + (K1ox * sqrtPhis - K1 * sqrtPhis) * sqrt(1.0 + LPEB/Leff)
  - K2ox * Vbseff
  - Theta0 * DVT0 * (Vbi - phi_s)
  - T2
  + (K3 + K3B * Vbseff) * TOXE * phi_s /(Weff + W0)
  + T1
  - T7;
//--------------end Vth-------------------------

//------Calculation for n-------------
Cdsc_term = (CDSC + CDSCD * x[0] + CDSCB * Vbseff) * Theta0;
T1 = (NFACTOR*esi/Xdep + Cdsc_term + CIT) / coxe;
```

```
if (T1 >= -0.5)
  n = 1.0 + T1;
else
  n = (1.0 + 3.0 * T1) / (3.0 + 8.0 * T1);
//------end n------------------------

//----Vth correction for pocket implant
if (DVTP0 > 0.0)
  {
    T0 = -DVTP1 * x[0];

    if (T0 < -EXP_THRESHOLD)
  T2 = MIN_EXP;
    else
  T2 = exp(T0);

    T3 = Leff + DVTP0 * (1.0 + T2);
    T4 = vt * log(Leff/T3);
    Vth -= n * T4;
  }

//----------Calculation for Vgse-------------
T0 = VFB + phi_s;

if((NGATE > 1.0e18) && (NGATE < 1.0e25) && (x[1] > T0))
  {
    T1 = 1.0e6 * eCharge * esi * NGATE / (coxe * coxe);
    T8 = x[1] - T0;
    T4 = sqrt(1.0 + 2.0 * T8 / T1);
    T2 = 2.0 * T8 / (T4 + 1.0);
    T3 = 0.5 * T2 * T2 / T1;
    T7 = 1.12 - T3 - 0.05;
    T6 = sqrt(T7 * T7 + 0.224);
    T5 = 1.12 - 0.5 * (T7 + T6);
  }
else
  Vgse = x[1];
//----------end Vgse-------------------------

//---------Calculation for Vgsteff------------
T2 = m * (Vgse - Vth) /(n * vt);

if (T2 > EXP_THRESHOLD)
  T0 = m * (Vgse - Vth);
else if (T2 < -EXP_THRESHOLD)
  T0 = vt * log(1.0 + MIN_EXP) * n;
else
  T0 = n * vt * log(1.0 + exp(T2));

T2 = (Voff_prime - (1.0 - m) * (Vgse - Vth)) / (n * vt);

if (T2 < -EXP_THRESHOLD)
  T1 = m + n * coxe * MIN_EXP / sqrt((phi_s * q * NDEP
      * esi * 1.0e6)/2.0);
else if (T2 > EXP_THRESHOLD)
  T1 = m + n * coxe * MAX_EXP / sqrt((phi_s * q * NDEP
      * esi * 1.0e6)/2.0);
```

```
else
  T1 = m + n * coxe * exp(T2) / sqrt((phi_s * q * NDEP
      * esi * 1.0e6)/2.0);

Vgsteff = T0/T1;
//--------end Vgsteff------------------------

//Effective width
//---Calculation for Weff------------------
T9 = sqrtPhis - sqrt(phi_s);
Weff += -2.0 * (DWJ * Vgsteff + DWB * T9);

if (Weff < 2.0e-8)
  {
    T0 = 1.0 / (6.0e-8 - 2.0*Weff);
    Weff = 2.0e-8 * (4.0e-8 - Weff) * T0;
  }
//-----end Weff---------------------------

//--------Calculation for Rds----------------
T0 = 1.0 + PRWG * Vgsteff;
T1 = PRWB * (sqrtPhis - sqrt(phi_s));
T2 = 1.0 / T0 + T1;
T3 = T2 + sqrt(T2 * T2 + 0.01);
T5 = RDSW + PRT * (T/300.0 - 1.0) * NF / (pow(Weffcj * 1.0e6,WR) * NF);
T4 = 0.5 * T5;
RDSWMIN += PRT * (T/300.0 - 1.0) * NF / (pow(Weffcj * 1.0e6,WR) * NF);
Rds = RDSWMIN + T3 * T4;
//------end Rds-----------------------------

//---------Calculation for lambda----
if (A1 == 0.0)
  lambda = A2;
else if (A1 > 0.0)
  {
    T0 = 1.0 - A2;
    T1 = T0 - A1 * Vgsteff - 0.0001;
    T2 = sqrt(T1 * T1 + 0.0004 * T0);
    lambda = A2 + T0 - 0.5 * (T1 + T2);
  }
else
  {
    T1 = A2 + A1 * Vgsteff - 0.0001;
    T2 = sqrt(T1 * T1 + 0.0004 * A2);
    lambda = 0.5 * (T1 + T2);
  }
//---------end lambda----------------

//-----------Calculation for F_doping--------------
F_doping = 0.5 * sqrt(1.0 + LPEB/Leff) * K1ox / (sqrt(phi_s - Vbseff))
  + K2ox - K3B * TOXE * phi_s / (Weff + W0);
//--------end of F-doping----------

//--------Calculation for Abulk----------
T0 = Leff/(Leff + 2.0 * sqrt(XJ * Xdep));
T1 = A0 * T0 + B0/(Weff + B1);
T2 = 1.0 + F_doping * T1;
```

```
adouble Abulk0 = T2;

Abulk = Abulk0 - Vgsteff * F_doping * AGS * A0 * pow(T0,3.0);

if (Abulk < 0.1)
  Abulk = (0.2 - Abulk) / (3.0 - 20.0 * Abulk);

if (Abulk0 < 0.1)
  {
    T9 = 1.0 / (3.0 - 20.0*Abulk0);
    Abulk0 = (0.2 - Abulk0) * T9;
  }

if (Abulk < 0.1)
  {
    T9 = 1.0 / (3.0 - 20.0*Abulk);
    Abulk = (0.2 - Abulk) * T9;
  }

T2 = KETA * Vbseff;

if (T2 >= -0.9)
  T0 = 1.0 / (1.0 + T2);
else
  T0 = (17.0 + 20.0 * T2) / (0.8 + T2);

Abulk *= T0;
Abulk0 *= T0;
//-------end Abulk-------------------

//--------Calculation for Vb----------
Vb = (Vgsteff + 2.0*vt)/Abulk;
//-----------end Vb-------------------

//The effective mobility is calculated using the universal
//mobility theorem which is valid in all regions of operation
//---------Calculation for ueff--------------
T1 = (Vgsteff + 2.0 * (VTH0 - VFB - phi_s))/TOXE;
T2 = exp(EU * log(T1)) * (UA + UC * Vbseff);

if (T2 >= -0.8)
  T3 = 1.0 + T2;
else
  T3 = (0.6 + T2) / (7.0 + 10.0 * T2);

if (U0 > 1.0)
  U0 = U0 / 1.0e4;

ueff = U0 / T3;
//---------end ueff---------------------------

//---------Calculation for Esat----------------
Esat = 2.0 * (VSAT - AT * (T/300.0 - 1))/ueff;
//----------end of Esat------------------------

//The saturation drain-source voltage
```

```
//---------Calculation for Vdsat-------------
if((Rds == 0.0) && (lambda == 1.0))
  {
    T0 = 1.0 / (Abulk + Esat * Leff + Vgsteff + 2.0 * vt);
    T3 = Esat * Leff * (Vgsteff + 2.0 * vt);
    Vdsat = T3 * T0;
  }
else
  {
    T6 = (Vgsteff + 2.0 * vt) * Weff * coxe * Rds
         * (VSAT - AT * (T/300.0 - 1.0));
    T9 = Abulk * Weff * coxe * Rds * (VSAT - AT * (T/300.0 - 1.0));
    T0 = 2.0 * Abulk * (T9 - 1.0 + 1.0/lambda);
    T1 = (Vgsteff + 2.0 * vt) * (2.0/lambda - 1.0) + Abulk * Esat * Leff
         + 3.0 * (Vgsteff + 2.0 * vt) * Abulk * Rds * Weff * coxe * (VSAT
         - AT * (T/300.0 - 1.0));
    T2 = (Vgsteff + 2.0 * vt) * (Esat * Leff + 2.0 * T6);
    T3 = sqrt(T1*T1 - 2.0 * T0 * T2);
    Vdsat = (T1 - T3)/T0;
  }
//---------end Vdsat------------------------

//The effective drain-source voltage
//----------Calculation for Vdseff---------
T1 = Vdsat - x[0] - DELTA;
T2 = sqrt(T1 * T1 + 4.0 * DELTA * Vdsat);
T0 = T1/T2;
T3 = 2.0 * DELTA / T2;

if (T1 >= 0.0)
  Vdseff = Vdsat - 0.5 * (T1 + T2);
else
  {
    T4 = 2.0 * DELTA / (T2 - T1);
    T5 = 1.0 - T4;
    Vdseff = Vdsat * T5;
  }

if (x[0] == 0.0)
  Vdseff = 0.0;

if (Vdseff > x[0])
  Vdseff = x[0];
//---------end Vdseff----------------------

//------------Calculation for Vasat-------
T0 = Esat*Leff + Vdsat + 2.0 * (Vgsteff * Rds * Weff * coxe
     * (VSAT - AT * (T/300.0 - 1.0))) * (1.0 - 0.5 * Abulk
     * Vdsat / (Vgsteff + 2.0 * vt));

T1 = 2.0/lambda - 1.0 + Rds * Vgsteff * Weff * coxe
     * (VSAT - AT * (T/300.0 - 1.0));

Vasat = T0 / T1;
//-----------end Vasat--------------------

//----------Calculation for Ids-----------
```

```
T1 = 4.0 * (VTH0 - VFB - phi_s);
T2 = (Vgsteff + T1) / (2.0e8 * TOXP);
T3 = 1.9e-9 / (1.0 + exp(0.7 * log(T2)));

T4 = esi * coxp /(esi + coxp * T3);
T5 = ueff * T4 * Weff/Leff;
T6 = Vgsteff * (1.0 - 0.5 * Vdseff * Abulk/(Vgsteff + 2.0 * vt));
T7 = T5 * T6 / (1.0 + Vdseff/(Esat * Leff));

Ids0 = T7 / (1.0 + T7 * Rds);

adouble F;

if (FPROUT <= 0.0)
  F = 1.0;
else
  F = 1.0 / (1.0 + FPROUT * (sqrt(Leff)/(Vgsteff + 2.0 * vt)));


//Channel length modulation effects
//------Calculation of Cclm and Vaclm---------
if ((Vgsteff * PVAG/(Esat * L)) > -0.9)
  T7 = 1.0 + Vgsteff * PVAG/(Esat * L);
else
  T7 = (0.8 + (Vgsteff * PVAG/(Esat * L)))
       / (17.0 + 20.0 * (Vgsteff * PVAG/(Esat * L)));

if ((PCLM >0.0) && ((x[0] - Vdseff) > 1.0e-10))
  {
    Cclm = F * T7 * (1.0 + Rds * Ids0) * (Leff + Vdsat/Esat)
          / (PCLM * litl);
    Vaclm = Cclm * (x[0] - Vdseff);
  }
else
  {
    Cclm = MAX_EXP;
    Vaclm = MAX_EXP;
  }
//--------end of Cclm and Vaclm----------

Va = Vasat + Vaclm;

//Drain induced barrier lowering effects DIBL
//----------Calculation for Vadibl-------
if ((DROUT * Leff/lt0) < EXP_THRESHOLD)
  T1 = exp(DROUT * Leff/lt0) / (2.0 * exp(DROUT * Leff/lt0)
       * MIN_EXP + pow(exp(DROUT * Leff/lt0)-1.0 ,2.0));
else
  T1 = 1.0 / (MAX_EXP - 2.0);

theta_rout = PDIBL1 * T1 + PDIBL2;


if (theta_rout > 0.0)
  {
    T2 = (Vgsteff + 2.0 * vt) * Abulk * Vdsat;
    T3 = Vgsteff + 2.0 * vt + Abulk * Vdsat;
```

```
    Vadibl = (Vgsteff + 2.0 * vt - T2/T3) / theta_rout;


    if ((PDIBLCB * Vbseff) >= -0.9)
  Vadibl *= (1.0 / (1.0 + (PDIBLCB * Vbseff)));
    else
  Vadibl *= (17.0 + 20.0 * (PDIBLCB * Vbseff)) / (0.8 + PDIBLCB * Vbseff);

    Vadibl *= T7;
  }
else
  Vadibl = MAX_EXP;
//-----------end Vadibl---------------

//Substrate current body effect SCBE
//-----Calculation for Vascbe----------
if (PSCBE2 > 0.0)
  {
    if ((x[0] - Vdseff) > PSCBE1 * litl/EXP_THRESHOLD)
  Vascbe = Leff * exp(PSCBE1 * litl/(x[0] - Vdseff)) / PSCBE2;
    else
  Vascbe = MAX_EXP * Leff / PSCBE2;
  }
else
  Vascbe = MAX_EXP;
//----------end Vascbe----------------

//Drain induced threshold shift due to pocket implants
//---------Calculation for Vadits------
if ((PDITSD * x[0]) > EXP_THRESHOLD)
  T2 = MAX_EXP;
else
  T2 = exp(PDITSD * x[0]);

if (PDITS > 0.0)
  Vadits = F * (1/PDITS) * (1.0 + (1.0 + PDITSL * Leff) * T2);
else
  Vadits = MAX_EXP;
//--------end Vadits-------------------

Ids = Ids0 * (1.0 + (x[0] - Vdseff)/Vadibl)
      * (1.0 + (x[0] - Vdseff)/Vadits)
      * (1.0 + log(Va/Vasat)/Cclm);

//--------Calculation for Subthreshold current------
T1 = ALPHA0 + ALPHA1 * Leff;
if ((T1 <= 0.0) || (BETA0 <= 0.0))
  Isub = 0.0;
else
  {   T2 = T1 / Leff;
  if ((x[0] - Vdseff) > BETA0 / EXP_THRESHOLD)
    {   T0 = -BETA0 / (x[0] - Vdseff);
    T1 = T2 * (x[0] - Vdseff) * exp(T0);
    T3 = T1 / (x[0] - Vdseff) * (T0 - 1.0);
    }
  else
    {   T3 = T2 * MIN_EXP;
```

```
    T1 = T3 * (x[0] - Vdseff);
    }
  T4 = Ids * Vdseff;
  Isub = T1 * T4;
  }
//------------end Isub-----------------------------

//-------Calculation for Igidl---------------------
T0 = 3.0 * TOXE;
T1 = (x[0] - Vgse - EGIDL) / T0;

if ((AGIDL <= 0.0) || (BGIDL <= 0.0) || (T1 < 0.0)
    || (CGIDL <= 0.0) || ((x[0] - x[2]) < 0.0))
  Igidl = 0.0;
else
  {T2 = BGIDL / T1;
  if (T2 < 100.0)
    {   Igidl = AGIDL * Weffcj * T1 * exp(-T2);
    T3 = Igidl * (1.0 + T2) / T1;
    }
  else
    Igidl = T1 * AGIDL * Weffcj * 3.720075976e-44;

  T4 = pow((x[0] - x[2]),2.0);
  T5 = (x[0] - x[2]) * T4;
  T6 = CGIDL + T5;
  T7 = T5 / T6;
  Igidl *= T7;
  }
//-------end Igidl--------------------------------

//Drain source DC
//----and finally.....
Ids =  NF * Vdseff * Ids * (1.0 + (x[0] - Vdseff)/Vascbe);
//--------end Ids--------------------------

//Equations for gate current
//They are Igs - Gate source current
//         Igd - Gate Drain current
//         Igcs - Gate source current through the channel
//         Igcd - Gate drain current through the channel
//         Igb - Gate bulk current

double A = 4.97232e-7;
double B = 7.45669e11;

//-----------Calculation for Voxacc--------------------
adouble Vfbzb, Voxacc;

T0 = DVT1W * Weff * Leff/(sqrt(esi*TOXE*Xdep0/e0));
if (T0 < EXP_THRESHOLD)
  {
    T1 = exp(T0);
    T2 = T1 - 1.0;
    T3 = T2 * T2;
    T4 = T3 + 2.0 * T1 * MIN_EXP;
    T8 = T1 / T4;
```

```
  }
else
  T8 = 1.0/(MAX_EXP - 2.0);

T0 = DVT0W * T8;
T8 = T0 * (Vbi - phi_s);

T0 = DVT1 * Leff /(sqrt(esi*TOXE*Xdep0/e0));
if (T0 < EXP_THRESHOLD)
  {
    T1 = exp(T0);
    T2 = T1 - 1.0;
    T3 = T2 * T2;
    T4 = T3 + 2.0 * T1 * MIN_EXP;
    T9 = T1 / T4;
  }
else
  T9 = 1.0/(MAX_EXP - 2.0);

double KT1 = -0.11;
double KT1L = 0.0;

T9 = DVT0 * T9 * (Vbi - phi_s);
T4 = TOXE * phi_s / (Weff + W0);
T0 = sqrt(1.0 + LPE0/Leff);
T5 = K1ox * (T0 - 1.0) * sqrtPhis + (KT1 + KT1L / Leff) * (T/300.0 - 1.0);
T6 = VTH0 - T8 - T9 + K3 * T4 + T5;

Vfbzb = T6 - phi_s - K1 * sqrtPhis;

T3 = Vfbzb - Vgse + Vbseff - 0.02;

if (Vfbzb <= 0.0)
  T0 = sqrt(pow(T3,2.0) - 4*Vfbzb*0.02);
else
  T0 = sqrt(pow(T3,2.0) + 4*Vfbzb*0.02);

Voxacc = Vfbzb - (Vfbzb - 0.5*(T3 + T0));

if (Voxacc < 0.0)
  Voxacc = 0.0;
//---------end Voxacc--------------------------------

//--------Calculation for Voxdepinv------------------
adouble Voxdepinv;

T1 = Vgse - (Vfbzb - 0.5*(T3 + T0)) - Vbseff - Vgsteff;

if (K1ox == 0.0)
  Voxdepinv = 0.0;
else if (T1 < 0.0)
  Voxdepinv = -T1;
else
  Voxdepinv = K1ox * (sqrt(T1 + 0.25*K1ox*K1ox) - 0.5*K1ox);

Voxdepinv += Vgsteff;
//-------end Voxdepinv------------------------------
```

```
//-------Calculation for Igc---------------------
adouble Igc, Vaux;

if ((Vgse-VTH0)/(vt*NIGC) > EXP_THRESHOLD)
  Vaux = Vgse - VTH0;
else if ((Vgse-VTH0)/(vt*NIGC) < -EXP_THRESHOLD)
  Vaux = vt*NIGC*log(1.0 + MIN_EXP);
else
  Vaux = vt*NIGC*log(1.0 + exp((Vgse - VTH0)/(vt*NIGC)));

T3 = AIGC * CIGC - BIGC;
T4 = BIGC * CIGC;
T5 = (-B*TOXE) * (AIGC + T3 * Voxdepinv - T4 * Voxdepinv * Voxdepinv);

if (T5 > EXP_THRESHOLD)
  T6 = MAX_EXP;
else if (T5 < -EXP_THRESHOLD)
  T6 = MIN_EXP;
else
  T6 = exp(T5);

Igc = A* Weff* Leff* exp(NTOX * log(TOXREF/TOXE)) * Vgse * Vaux * T6;
//--------end Igc---------------------------------

//----Calculation for Igcs and Igcd--------------
adouble Igcs, Igcd;

if ((-PIGCD*x[0]) > EXP_THRESHOLD)
  T1 = MAX_EXP;
else if ((-PIGCD*x[0]) < -EXP_THRESHOLD)
  T1 = MIN_EXP;
else
  T1 = exp(-PIGCD*x[0]);

T8 = (-PIGCD*x[0]) * (-PIGCD*x[0]) + 2.0e-4;
T0 = T8 * T8;
T2 = T1 - 1.0 + 1.0e-4;
T10 = (T2 - (-PIGCD*x[0])) / T8;

Igcs = Igc * T10;

T2 = T1 - 1.0 - 1.0e-4;
T10 = ((-PIGCD * x[0]) * T1 - T2) / T8;

Igcd = Igc * T10;
//-----end Igcs and Igcd-------------------------

//----Calculation for Igs and Igd---------------
adouble Igs, Igd;
double Vfbsd;

if (NGATE > 0.0)
  Vfbsd = (k*T/q)*log(NGATE/NSD);
else
  Vfbsd = 0.0;
```

```
T0 = x[1] - Vfbsd;
Vgse = sqrt(T0 * T0 + 1.0e-4);

T2 = x[1] * Vgse;
T3 = AIGSD * CIGSD - BIGSD;
T4 = BIGSD * CIGSD;
T5 = (-B * TOXE * POXEDGE) * (AIGSD + T3 * Vgse - T4 * Vgse * Vgse);

double  Toxratioedge = exp(NTOX * log(TOXREF / (TOXE * POXEDGE)))
                       / TOXE / TOXE / POXEDGE / POXEDGE;

if (T5 > EXP_THRESHOLD)
  T6 = MAX_EXP;
else if (T5 < -EXP_THRESHOLD)
  T6 = MIN_EXP;
else
  T6 = exp(T5);

Igs = A * Weff * Toxratioedge * DLCIG * T6 * T2;

T0 = (x[1] - x[0]) - Vfbsd;

adouble Vgde;
Vgde = sqrt(T0 * T0 + 1.0e-4);

T2 = Vgde * (x[1] - x[0]);
T5 = (-B * TOXE * POXEDGE) * (AIGSD + T3 * Vgde - T4 * Vgde * Vgde);

if (T5 > EXP_THRESHOLD)
  T6 = MAX_EXP;
else if (T5 < -EXP_THRESHOLD)
  T6 = MIN_EXP;
else
  T6 = exp(T5);

Igd = A * Weff * Toxratioedge * DLCIG * T6 * T2;
//------end Igs and Igd-----------------------------

//---------Calculations for Igb--------------------
T0 = vt * NIGBACC;
T1 = -Vgse + Vbseff + Vfbzb;

if (T1/T0 > EXP_THRESHOLD)
  Vaux = T1;
else if (T1/T0 < -EXP_THRESHOLD)
  Vaux = T0 * log(1.0 + MIN_EXP);
else
  Vaux = T0 * log(1.0 + exp(T1/T0));

T2 = (Vgse - Vbseff) * Vaux;
T11 = 4.97232e-7 * Weff * Leff * exp(NTOX * log(TOXREF/TOXE));
T12 = -7.45669e-11 * TOXE;
T3 = AIGBACC * CIGBACC - BIGBACC;
T4 = BIGBACC * CIGBACC;
T5 = T12 * (AIGBACC + T3 * Voxacc - T4 * Voxacc * Voxacc);

if (T5 > EXP_THRESHOLD)
```

```
  T6 = MAX_EXP;
else if (T5 < -EXP_THRESHOLD)
  T6 = MIN_EXP;
else
  T6 = exp(T5);

adouble Igbacc;
Igbacc = T11 * T2 * T6;

T0 = vt * NIGBINV;
T1 = Voxdepinv - EIGBINV;

if (T1/T0 > EXP_THRESHOLD)
  Vaux = T1;
else if (T1/T0 < -EXP_THRESHOLD)
  Vaux = T0 * log(1.0 + MIN_EXP);
else
  Vaux = T0 * log(1.0 + exp(T1/T0));

T2 = (Vgse - Vbseff) * Vaux;
T11 *= 0.75610;
T12 *= 1.31724;
T3 = AIGBINV * CIGBINV - BIGBINV;
T4 = BIGBINV * CIGBINV;
T5 = T12 * (AIGBINV + T3 * Voxdepinv - T4 * Voxdepinv * Voxdepinv);

if (T5 > EXP_THRESHOLD)
  T6 = MAX_EXP;
else if (T5 < -EXP_THRESHOLD)
  T6 = MIN_EXP;
else
  T6 = exp(T5);

adouble Igbinv;
Igbinv = T11 * T2 * T6;

adouble Igb;
Igb = Igbinv + Igbacc;

//Equations for capacitance. CAPMOD=2 with a 40/60 charge partition
//between the source and drain.
//The equations for charge at the respective transistor nodes begins here.
//The derivatives of charge with respect to time are evaluated in eval2.
//That corresponds to the current contribution at each node.

//------Calculation for LeffCV and WeffCV----------
Leff = Lnew - 2.0 * DLC;
Weff = Wnew - 2.0 * DWC;
//------end LeffCV and WeffCV---------------------

//-----Calculation for VbseffCV---------------------
adouble VbseffCV;
if (Vbseff < 0.0)
  VbseffCV = Vbseff;
else
  VbseffCV = phi_s - Phis;
//------end VbseffCV--------------------------------
```

```
//------Calculation for VgsteffCV------------------
T0 = n * NOFF * k * T / q;
T1 = (Vgse - Vth) / T0;
if (T1 > EXP_THRESHOLD)
  Vgsteff = Vgse - Vth - VOFFCV;
else if (T1 < -EXP_THRESHOLD)
  Vgsteff = T0 * log(1.0 + MIN_EXP);
else
  Vgsteff = T0 * log(1.0 + exp(T1));
//-------End VgsteffCV-----------------------------

//---------------Calculation for Vfbeff-------------
adouble V3;
V3 = Vfbzb - Vgse + VbseffCV - 0.02;
if (Vfbzb <= 0.0)
  T0 = sqrt(V3 * V3 - 4.0 * 0.02 * Vfbzb);
else
  T0 = sqrt(V3 * V3 + 4.0 * 0.02 * Vfbzb);

adouble Vfbeff;
Vfbeff = Vfbzb - 0.5 * (V3 + T0);
//-----------end Vfbeff-----------------------------

T0 = (Vgse - VbseffCV - Vfbzb) / TOXP;

//-------------Calculation for Tcen----------------
double LDEB = sqrt(esi * vt/(q * NDEP * 1.0e6)) / 3.0;

adouble Tcen;
T1 = T0 * ACDE;
if ((-EXP_THRESHOLD < T1) && (T1 < EXP_THRESHOLD))
  Tcen = LDEB * exp(T1);
else if (T1 <= -EXP_THRESHOLD)
  Tcen = LDEB * MIN_EXP;
else
  Tcen = LDEB * MAX_EXP;

V3 = LDEB - Tcen - 1.0e-3 * TOXP;

adouble V4;
V4 = sqrt(V3 * V3 + 4.0 * 1.0e-3 * TOXP * LDEB);

Tcen = LDEB - 0.5 * (V3 + V4);
//-------------end Tcen----------------------------

adouble Ccen;
Ccen = esi / Tcen;

adouble Coxeff;
Coxeff = Ccen * coxp / (Ccen + coxp);

//--------Calculation for QoverlapCox-------------
adouble QoverlapCox, Qac0, CoxWLcen, Qsub0;

CoxWLcen = coxp * Weff * Leff * NF * Coxeff / coxe;
Qac0 = CoxWLcen * (Vfbeff - Vfbzb);
```

```
QoverlapCox = Qac0 / Coxeff;

T0 = 0.5 * K1ox;
T3 = Vgse - Vfbeff - VbseffCV - Vgsteff;
if (K1ox == 0.0)
  { T1 = 0.0;
  T2 = 0.0;
  }
else if (T3 < 0.0)
  { T1 = T0 + T3 / K1ox;
  T2 = CoxWLcen;
  }
else
  { T1 = sqrt(T0 * T0 + T3);
  T2 = CoxWLcen * T0 / T1;
  }

Qsub0 = CoxWLcen * K1ox * (T1 - T0);
QoverlapCox = Qsub0 / Coxeff;

//--------Calculation for Delta_phis------------
if (K1ox <= 0.0)
  { T2 = 0.25 * MOIN * vt;
  T0 = 0.5 * sqrt(phi_s);
  }
else
  { T2 = MOIN * vt * K1ox * K1ox;
  T0 = K1ox * sqrt(phi_s);
  }
T1 = 2.0 * T0 + Vgsteff;

adouble Delta_phis;

Delta_phis = vt * log(1.0 + T1 * Vgsteff / T2);
//----------end Delta_phis-------------------

//The calculation for Tcen must be done once more
T0 = (Vgsteff + 4.0*(VTH0 - VFB - phi_s))/ (2.0 * TOXP);
T1 = 1.0 + exp(0.7 * log(T0));
T2 = 0.7 * exp(0.7 * log(T0)) / (T0 * 2.0 * TOXP);
Tcen = 1.9e-9 / T1;

Ccen = esi / Tcen;
Coxeff = Ccen * coxp / (Ccen + coxp);
CoxWLcen = coxp * Weff * Leff * Coxeff / coxe;

adouble AbulkCV;
AbulkCV = Abulk0 * (1.0 + pow((CLC/Leff),CLE));

adouble VdsatCV;
VdsatCV = (Vgsteff - Delta_phis) / AbulkCV;

T0 = VdsatCV - x[0] - 0.02;
T1 = sqrt(T0 * T0 + 4.0 * 0.02 * VdsatCV);

adouble VdseffCV;
```

```
  if (T0 >= 0.0)
    VdseffCV = VdsatCV - 0.5 * (T0 + T1);
  else
    {
      T3 = 0.04 / (T1 - T0);
      T4 = 1.0 - T3;
      T5 = VdsatCV * T3 / (T1 - T0);
      VdseffCV = VdsatCV * T4;
    }

  if (x[0] == 0.0)
    VdseffCV = 0.0;

  T0 = AbulkCV * VdseffCV;
  T1 = Vgsteff - Delta_phis;
  T2 = 12.0 * (T1 - 0.5 * T0 + 1.0e-20);
  T3 = T0 / T2;
  T4 = 1.0 - 12.0 * T3 * T3;
  T5 = AbulkCV * (6.0 * T0 * (4.0 * T1 - T0) / (T2 * T2) - 0.5);
  T6 = T5 * VdseffCV / AbulkCV;

  adouble qgate;
  qgate = CoxWLcen * (T1 - T0 * (0.5 - T3));

  adouble qbulk;
  qbulk = CoxWLcen * (1.0 - AbulkCV) * (0.5*VdseffCV - T0*VdseffCV/T2);

  QoverlapCox = qbulk / Coxeff;

  T2 = T2 / 12.0;
  T3 = 0.5 * CoxWLcen / (T2 * T2);
  T4 = T1 * (2.0 * T0 * T0 / 3.0 + T1 * (T1 - 4.0 * T0 / 3.0))
       - 2.0 * T0 * T0 * T0 / 15.0;

  adouble qsrc;
  qsrc = -T3 * T4;

  qgate += Qac0 + Qsub0 - qbulk;
  qbulk -= Qac0 + Qsub0;
  adouble qdrn;
  qdrn = -(qbulk + qgate + qsrc);
  qsrc = -(qbulk + qgate + qdrn);

  y1[0] = qdrn; //Drain charge
  y1[1] = qgate; //Gate Charge
  y1[2] = qsrc; //Source charge

  z1[0] = Ids + Isub + Igidl - Igcd - Igd; //Drain current
  z1[1] = Igs + Igcd + Igd + Igcs + Igb; //Gate current
  z1[2] = -Ids - Igs - Igcs; //Source current

  z1[3] = x[0] - x[2]; //Vdb
  z1[4] = x[1] - x[2]; //Vgb
  z1[5] = -x[2];//Vsb
}

void Mos4::eval2(adoublev& dy1, adoublev& z1, adoublev& y2,
```

```
adoublev& z2) {
  z2[0] = z1[3]; //Vdb
  z2[1] = z1[4]; //Vgb
  z2[2] = z1[5]; //Vsb

  z2[3] = z1[0] + dy1[0]; //Drain current  Drain charge derivative
  z2[4] = z1[1] + dy1[1]; //Gate current, gate charge derivative
  z2[5] = z1[2] + dy1[2]; //Source current, source charge derivative
}
```

## A.2 Header file

```
// BSIM4 MOSFET MODEL
//
//               Drain 1
//                 o
//                 |
//                 |
//             |---+
//             |
// Gate 2 o-----|------o 4 Bulk
//             |
//             |---+
//                 |
//                 |
//                 o
//              Source 3
//
//
//    Author: Nikhil Kriplani

#ifndef Mos4_h #define Mos4_h 1

class Mos4:public NAdolcElement { public:

  Mos4(const string& iname);

  ~Mos4() {}

  static const char* getNetlistName()
  {
    return einfo.name;
  }

  // Do some local initialization
  virtual void init() throw(string&);

private:

virtual void eval1(adoublev& x, adoublev& xt, adoublev& y1,
adoublev& z1); virtual void eval2(adoublev& dy1, adoublev& z1,
adoublev& y2, adoublev& z2);

  // Element information
  static ItemInfo einfo;

  // Number of parameters of this element
  static const unsigned n_par;

  // Parameter variables
  double TOXE, TOXP, EPSROX, VFB ,VTH0, NGATE, XL, XW, NF, W, L;
  double DWG, DWB, WINT, WL, WLN, WW, WWN, WWL, LINT, LL, LLN, LW;
  double LWN, LWL, K1, K2, LPEB, LPE0, K3, K3B, W0, DVT0W, DVT0;
  double DVT1W, DVT1, DSUB, ETA0, ETAB, TOXM, T, NDEP, PHIN, VBM, NSUB;
  double DVT2W, NSD, DVT2, MINV, NFACTOR, CDSC, CDSCD, CDSCB, CIT, KETA;
  double B0, B1, A0, AGS, XJ, U0, UA, UB, UC, EU, DELTA, PDITS, FPROUT, PDITSL;
```

```
  double PDITSD, PSCBE2, PSCBE1, PDIBLCB, PVAG, PDIBL1, PDIBL2, DROUT, PCLM;
  double A1, A2, RDSWMIN, RDSW, PRWG, PRWB, WR, WLC, WWC;
  double DWJ, CLC, CLE, NOFF, VOFFCV, CF, CKAPPAD, CKAPPAS;
  double LLC, LWC, LWLC, WWLC, CGBO, VOFF, VOFFL, POXEDGE, TOXREF, NTOX, DLCIG;
  double AIGSD, BIGSD, CIGSD, MOIN, VSAT, AIGC, BIGC, CIGC, NIGC;
  double PIGCD, DVTP0, DVTP1, PRT, AT, XT, ALPHA0, ALPHA1, BETA0;
  double AGIDL, BGIDL, CGIDL, EGIDL, ACDE, DLC, DWC;
  double AIGBACC, BIGBACC, CIGBACC, NIGBACC, AIGBINV, BIGBINV, CIGBINV;
  double EIGBINV, NIGBINV, KT1l, KT1, KT2;

  // Parameter information
  static ParmInfo pinfo[];

};

#endif
```

# Bibliography

[1] Foty, *MOSFET modeling with SPICE: Principles and Practice*, Prentice Hall, 1997.

[2] Liu, *MOSFET Models for SPICE simulation including BSIM3v3 and BSIM4*, John Wiley and Sons, 2001.

[3] H. Shichman and D. Hodges, "Modeling and Simulation of Insulated-Gate Field-Effect Transistor Switching Circuits," *IEEE J. Sol. St. Circ.* vol. 3, pp. 285-289 (1968)

[4] *Field Effect Transistors*, (ed. by J. Wallmark and H. Johnson), Prentice-Hall, 1966.

[5] Lee, Shur, Fjeldy and Ytterdal, *Semiconductor Device Modeling for VLSI: with the AIM-spice circuit simulator*, Prentice Hall, 1993.

[6] Fjeldy, Ytterdal and Shur, *Introduction to device modeling and circuit simulation*, A Wiley-Interscience Publication, 1998.

[7] Ron M. Kielkowski, *SPICE Practical Device Modeling*, McGraw-Hill, Inc., 1995.

[8] M. B. Steer, "Transient and Steady-State Analysis of Nonlinear RF and Microwave Circuits," ECE 718 class notes, 2001.

[9] R. S. Bain, *NNES User's Manual*, 1993.

[10] Kundert and Songiovanni-Vincentelli, *Sparse User's Guide - A Sparse Linear Equation Solver*, Department of Engineering and Computer Sciences, University of California, Berkeley, Calif. 94720 Version 1.3a, Apr 1988.

[11] Griewank, Juedes and Utke, *Adol-C: A Package for the Automatic Differenciation of Algorithms Written in C/C++*, Version 1.7, Sep 1996.

[12] Enz, Krummenacher and Vittoz, "An analytical MOS transistor model valid in all regions of Operation and dedicated to low-voltage and low-current applications," Analog Integrated Circuits and Signal Processsing, Kluwer Academic Publishers, pp. 83-114, July 1995

[13] Vlach and Singhal, *Computer Methods for Circuit Analysis and Design*, Chapman and Hall, 1994.