TR 3567 S

Stellingen
behorende bij het proefschrift
INTO DEEP SUBMICRON:
a simulation perspective
door
Serban Bruma

1. De "event-driven" aanpak hoort tot de essentie van de natuur, terwijl "global equations" alleen in onze geest bestaan.

2. Naarmate de afmetingen van transistors verder in het submicron gebied doordringen, stagneert de performance verbetering als gevolg van de verzadiging van de carrier velocity.

3. "Level transparency" is de sleutel tot system-on-chip simulatie.

4. "Het ikoon van de dode ster
   Komt langzaam aan de hemel op.
   Zij was, en was toch niet te zien,
   Nu we haar zien, is ze verdwenen."
   <<M. Eminescu, Aan de ster>>

5. In dichtbevolkte systemen tellen de interacties tussen de individuen veel zwaarder als de individuele deugden.

6. Als we ideeën konden communiceren in plaats van woorden, zou elk cultureel onderscheid verdwijnen.

7. Software is voor computers wat onderwijs is voor mensen. Wat een netwerk is voor computers is de omgeving voor mensen.

8. Het grootste probleem dat het informatietijdperk met zich brengt, is het filteren van de ruis.

9. Het belangrijkste gevolg van het bestuderen van razend ingewikkelde zaken is dat we inzicht verwerven in eenvoudige, fundamentele problemen.

10. Het laten maken van "stellingen" is een onderzoek naar de geestesgesteldheid van de promovendus na voltooiing van het promotiewerk.

Propositions
accompanying the thesis
INTO DEEP SUBMICRON:
a simulation perspective
by
Serban Bruma

1. The "event-driven" approach is in the very essence of nature, while "global equations" exist only in our minds.

2. When down-scaling into the deep submicron regime, the saturation of carrier velocity brings performance improvement to a halt.

3. "Level transparency" is the key to system-on-a-chip simulation.

4. "The icon of the now dead star
Slow in the sky it rises;
She was, while we could not see her,
Now that we see, she's vanished."
<<M. Eminescu, To the star>>

5. In densely populated systems what matters are the interactions between individuals rather than the particular virtues of each individual.

6. If we could communicate through ideas instead of words, cultural borders would disappear.

7. Software is for computers like education for human beings. What a network is for computers, is the society for the people.

8. The main problem brought by the information age is filtering the informational noise.

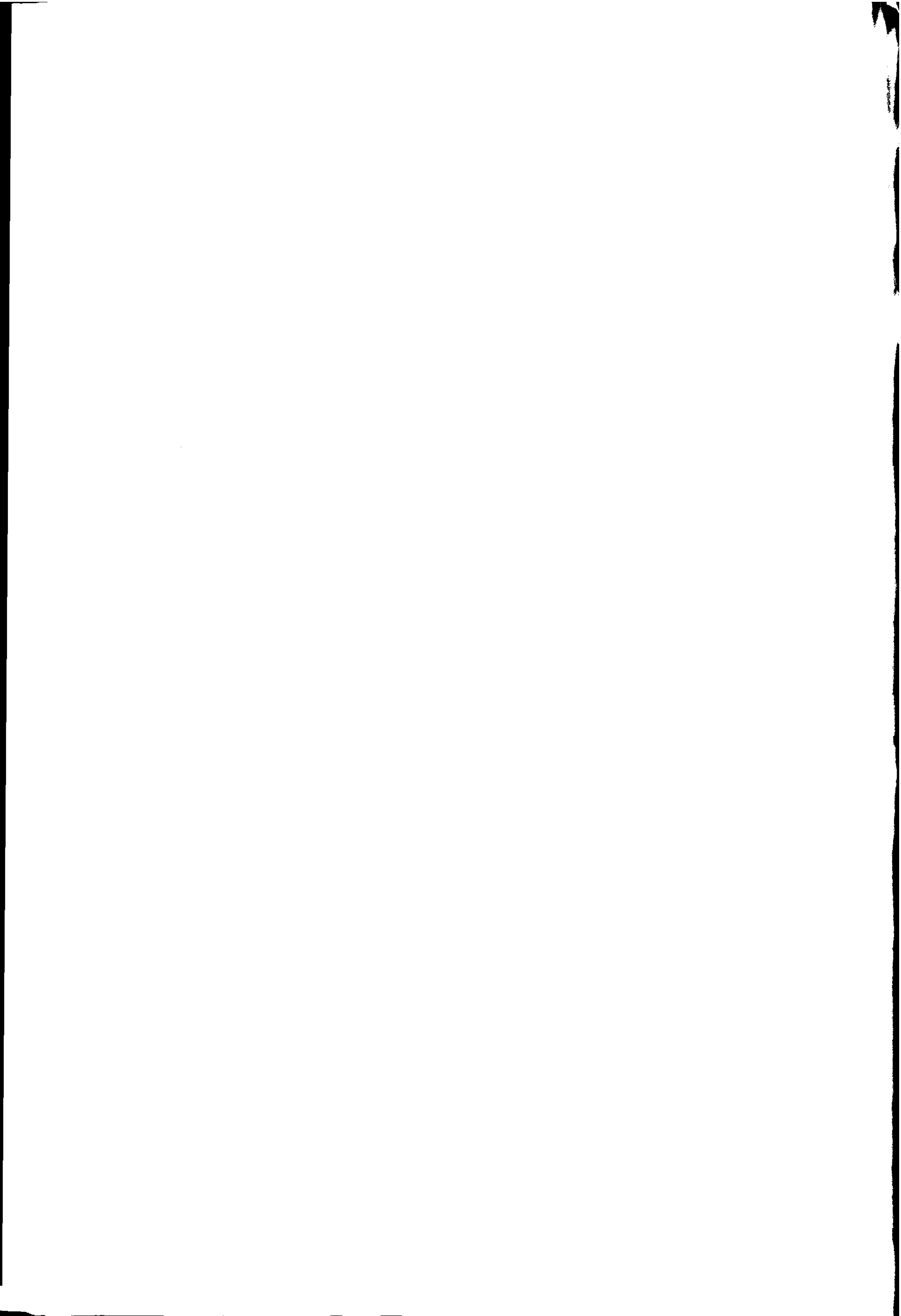
9. The main consequence of studying intricate matters is that we will be able to handle simple, fundamental matters.

10. The "propositions" are a sanity check of what is left of the spirituality of the Ph.D. candidate after years of research.

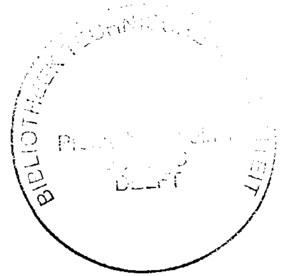
TR 3567

INTO DEEP SUBMICRON:
a simulation perspective

Serban BRUMA



INTO DEEP SUBMICRON:
a simulation perspective



PROEFSCHRIFT

ter verkrijging van de graad van doctor
aan de Technische Universiteit Delft,
op gezag van de Rector Magnificus prof. ir. K.F. Wakker,
voorzitter van het College voor Promoties,
in het openbaar te verdedigen op maandag 25 september 2000 om 16:00 uur

door

Serban BRUMA

inginer - Universitatea Tehnică din Iaşi, Roemenië
geboren te Iaşi, Roemenië

Dit proefschrift is goedgekeurd door de promotor:
Prof. dr. ir. R.H.J.M. Otten

Samenstelling promotiecommissie:

Rector Magnificus, voorzitter	
Prof. dr. ir. R.H.J.M. Otten, promotor	Technische Universiteit Delft
Prof. dr. ir. A.H.M. van Roermund	Technische Universiteit Delft
Prof. dr. H. Wallinga	Universiteit Twente
Prof. dr. ir. W.M.G. van Bokhoven	Technische Universiteit Eindhoven
Dr. ir. R. Nouta	Technische Universiteit Delft
Dr. ir. N.P. van der Meijs	Technische Universiteit Delft

This book was set using the TEX typesetting system (and related packages) and the `dvips` DVI to PostScript translator. The fonts used were 10 points Times Roman. Part of the illustrations were created by using XFIG, XMGR, `xdvi` and `ghostview`. All these programs were made freely available to the public by their authors.

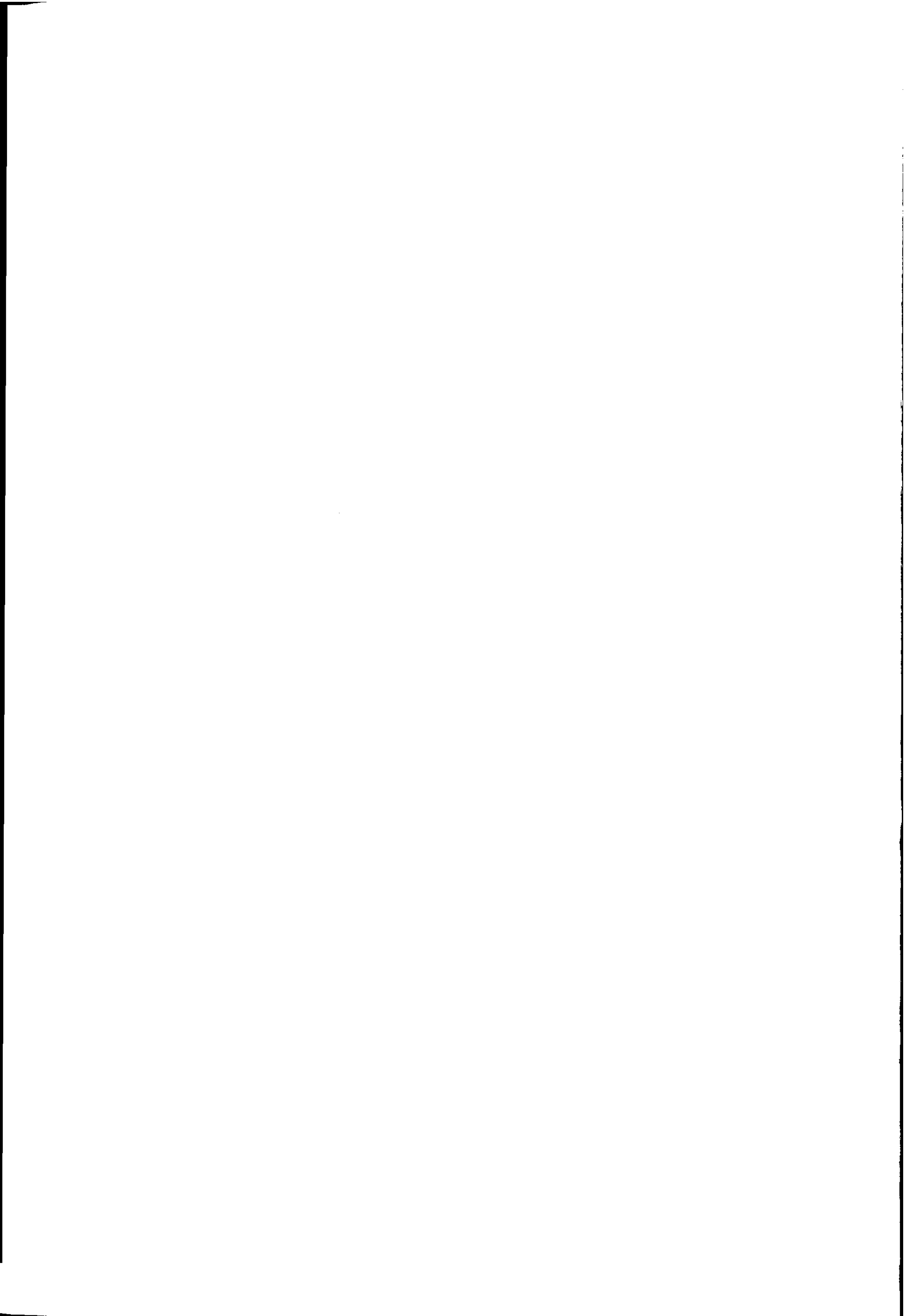
Copyright © 2000 by Serban Bruma. All rights reserved. No part of this book may be reproduced in any form or by any electronic or mechanical means (including photocopying, recording, or information storage and retrieval) without prior permission in writing from the author. Correspondence about this thesis and related subjects can be directed via email to: `serban@cas.et.tudelft.nl` .

ISBN 90-9014127-8

"E pur si muove."

Galileo Galilei

To my parents and my brother.



Contents

Abstract v

Acknowledgments vii

1 Deep submicron 1

1.1 Evolution of technology . . . 1

1.2 Devices . . . 3

1.3 Interconnect . . . 5

1.4 Complexity . . . 7

1.5 This thesis . . . 8

2 Level transparency 11

2.1 System description . . . 12

2.2 Direct methods . . . 14

2.3 Local linearization . . . 15

2.4 Locally linear systems . . . 17

2.5 A piecewise linear approach . . . 18

2.6 One algorithm . . . 23

3 Transistors in the limit 25

3.1 Impact of velocity saturation . . . 27

3.2 Model equations . . . 29

3.3 Scaling down . . . 34

3.4 Saturated velocity . . . 38

3.5 Piecewise linear transistor models . . . 40

4 Collapsible current sources **43**

4.1 Definition 44

4.2 Algorithmic requirements 49

4.3 Transistors as collapsible current sources 52

4.4 Modified collapsible current sources 52

4.5 Choosing a simulator 56

5 Introducing time **59**

5.1 Time discretization 61

5.2 Multirate integration 63

5.3 Event-driven simulation 64

5.4 Multi-level simulation capabilities 66

6 Transistor inertia **71**

6.1 Quasi-static approximations 72

6.2 Modeling large-signal charge variations 76

6.3 Evaluating large-signal charge variations 81

6.4 Dynamic characterization of an inverter 84

6.5 Average capacitors 86

7 One language, one algorithm **89**

7.1 Voltage and current events 90

7.2 Constitutive processes 92

7.3 Topological processes 93

7.4 Model calibration 95

7.5 Using the VHDL-MOS package 98

8 Perspectives **101**

8.1 Libraries 102

8.2 Complex digital circuits 103

8.3 Mixed analog/digital circuits 106

8.4 Conclusions 108

9 A deep submicron vanishing point **113**

A The linear complementarity problem **119**

A.1 Path followers 120

A.2 Complementary pivoting 122

A.3 PLATO 124

B Long channel static operation **125**

B.1 Transistor structure 125

B.2 Capacitors 127

B.3 Diodes 134

B.4 Transistors 135

B.5 Remarks 143

C VHDL summary **145**

C.1 Elements of behavior 145

C.2 Elements of structure 148

C.3 Mixed structural and behavioral models 149

D Charge pump PLL **153**

Bibliography **161**

Glossary **167**

About the author **173**

Samenvatting **175**

Abstract

The increasing demand for more processing capability has brought the CMOS technology into the deep submicron era. To integrate complex systems on the same chip designers have to manage the diversity and the complexity of the deep submicron circuits during the design process.

This thesis describes research on efficient analysis of complex, large-size deep submicron CMOS circuits. The *diversity* aspect of the problem calls for the usage of one single simulation algorithm for the multiple abstraction levels involved in system description. The *complexity* aspect of the problem calls for a minimal, compact model for the components.

We show that the event-driven approach, based on local modeling and global signaling, is a better alternative for efficient multi-level simulation than an equation-based approach. The piecewise linear approach, which is based on diode states and thus satisfies the one-algorithm demand for multi-level simulation, falls short in what concerns the computational efficiency.

We use a compact, piecewise linear model for the static behavior of the deep submicron MOS transistor. This collapsible current source model, is minimal to explain the transistor level operation of the CMOS circuits. Beside this, we use a compact model based on average capacitors to capture the terminal charge variations in the MOS structure for large-signal dynamic operation.

We prove that this compact way of modeling the static and dynamic operation of a deep submicron MOS transistor is an analytically sound method as far as the waveform estimations and power consumption for a digital CMOS circuit are concerned.

We achieve efficient analysis of complex, large-size deep submicron CMOS circuits by using this compact model of the MOS transistor in the framework of an event-driven simulator. We present a special VHDL package which we developed to capture continuous aspects of CMOS circuit operation in an event-driven simulator.

We practically use this approach to verify large circuits. We obtain a high level of accuracy, comparable with the one achieved in the classical transistor level modeling, however at considerably lower price in terms of computation. Our approach opens the way to multi-level simulation, i.e., digital gate-level descriptions can coexist with analog-like transistor-level descriptions inside the same event-driven simulation kernel.

Keywords: deep submicron, multi-level simulation, level transparency, event-driven, piecewise linear, collapsible current source model, average capacitors model, CMOS

Acknowledgments

The research presented in this thesis was carried out in the Circuits and Systems group at the Delft University of Technology.

At this occasion I would like to express my gratitude to a number of people that were of great help during my research. First of all, my supervisor Ralph Otten, for support and guidance towards the completion of this thesis. I do appreciate the freedom and trust that I was granted, as indispensable ingredients for my accomplishments. Reinder Nouta encouraged me with the VHDL-MOS approach. Nick van der Meijs and Arjan van Genderen gave their advice on various aspects of the CAD tools. Patrick Groeneveld and Viorica Simion generously offered their support in the early stages of my research. Jack Glas, with his rigorous time planning, acted as an efficiency example. Jan Nieuwstad helped me to set-up a Linux machine for typesetting this thesis. Aard Wiersma, as student, thoroughly exercised with the final approach. My gratitude goes also to Marion de Vlieger for ensuring the smooth functioning and the good spirit of the group.

I would also like to thank all my colleagues from Philips Research for support and help towards finishing this thesis.

Finally, I would like to thank my parents for continuous encouragement and Cezar Bruma for fraternal support. This book is dedicated to them.

Serban Bruma
Netherlands, September 2000

Chapter 1

Deep submicron

Contents

1.1 Evolution of technology 1
1.2 Devices 3
1.3 Interconnect 5
1.4 Complexity 7
1.5 This thesis 8

Three decades ago the digital CMOS emerged as a robust technology. At that time the theoretical framework was ripened: boolean algebra and sequential processing of digital data were matured concepts. The CMOS technology offered fertile ground for the implementation of the digital abstraction. It was a robust technology due to the large built-in noise margin.

The design effort was considerably lower because designers could much easier manipulate the digital abstraction than operate with a concept that would take into account the analog operation of the circuits.

1.1 Evolution of technology

During the last three decades these technological and economical advantages of the CMOS technology had a major contribution to the boost of the semiconductor industry, making it following the predictions of the Moore's [1] law.

In the last few years the pressure put by the demands of the new information age made people doubt of the capacity of the once so-good CMOS technology. The ever increasing demand for more processing power translated into two issues: (1) higher processing rates, and (2) more processing capability per silicon unit area.

Engineers believed that by down-scaling the dimensions of the MOS transistor they would reach both goals, therefore it would be possible to integrate more functionality on a given silicon area and at the same time improve the performance[1].

This can be seen from the following line of reasoning. By scaling-down the MOS transistor length, the current capability per unit width improves with some factor. The gate area reduces and as a consequence the capacitive load of the driver may decrease. Moreover, by scaling-down to a certain extent the transistor width the performance improves still considerably. Therefore, with the length and the width scaled-down, one can obtain both a more efficient utilization of the silicon area and a better performance.

The delay of the gate is roughly defined by:

$$t_d \approx \frac{(C_{GATE}/W) \cdot V_{DD}}{I_{D,SAT}/W} \tag{1.1}$$

where C_{GATE} is the switched capacitive load, W is the MOS transistor width, $I_{D,SAT}$ is the saturation current of the transistor, and V_{DD} is the supply voltage. By scaling down the transistor length, an improvement in the current capability $I_{D,SAT}/W$ can be obtained. In the overall scaling effort, the specific gate capacitance $C_{GATE}/W = C_{ox}L$ turns out to a first approximation to be constant, as C_{ox}, the gate oxide capacitance per unit area, increases by reducing the thickness of the gate oxide and L, the effective channel length, decreases. The supply voltage V_{DD} must decrease with the down scaling, based on power considerations. Concluding, the gate delay diminishes, i.e., improves, by shrinking the technology.

One might believe that down sizing can infinitely go on, however, as explained in the next sections, there are essential limiting factors.

In the situation of ideal down scaling, the same area of silicon would sell at a higher price due to more added value, which stems from: (1) the possibility to physically integrate more processing capability, which actually was the initial aim, and (2) the increased `complexity` of the systems that could be integrated on one

[1] In this context performance is equivalent to speed.

chip. The latter was the direct consequence of the possibility to have hundreds of millions of transistors on a deep submicron chip. Therefore the deep submicron technology marks the dawning of the system-on-a-chip era.

However, nothing comes from free. To extend the validity of the Moore's law further into future engineers have to solve a lot of problems. What could be considered just side-effects in the beginning proved to be the limiting factors in deploying new technologies.

The challenges that have to be dealt with in the deep submicron technology can be classified in three categories:

- effects on devices,
- effects on interconnect,
- induced consequences at system design level.

The first category directly relates to the shrinking of the dimensions of the active devices. The second category includes effects which are indirect consequences of shrinking the feature size all over the chip, including the interconnect. The last category came into reality when designers later realized that in their methodology for system-on-a-chip design they will have to deal with the increased system complexity, as well.

1.2 Devices

The deep submicron effects at device level arise from the fact that all the three dimensions of the transistor have been shrunk-down.

1. In our view the most important limiting factor[2] in the down-scaling is the saturation of the **drive current** as a consequence of the saturation of the carrier velocity [2]. The immediate implication is that the device performance, measured as delay time, does not improve anymore with the down-scaling.

At very short channel length, most carriers travel at a maximum saturated velocity, v_{sat}, through the channel, which nearly eliminates the influence of the

[2]We do not deny another important limiting factor, namely the poor scaling of the interconnect performance. While the down-scaling of the active device dimensions is the originally desired action, the down-scaling of the interconnect appeared as a consequence of the former.

channel length on the saturation current. A more suitable expression of the drain saturation current for this limiting case is:

$$I_{D,SAT} = W v_{sat} C_{ox} (V_{GS} - V_{th}) \tag{1.2}$$

This expression makes clear that the drain saturation current is independent of the channel length. Moreover it shows a linear dependence on the gate voltage drive. These characteristics contrast with the behavior of the classical long channel MOS, where the drain saturation current does depend on the channel length and the dependence is proportional to the square of the gate voltage drive. Apart from showing the forthcoming limitation of the deep submicron technology, equation (1.2) hints at how to build a compact static model for the deep submicron device.

We present the theoretical background and the derivation of the equation (1.2) in Chapter 3 and the implication on the limitation of the performance increase of the deep submicron devices in Chapter 9. We explain how we exploited the linear dependency to obtain computationally efficient model implementations for the deep submicron transistor in Chapter 4 and in Chapter 7. □

2. Another issue that requires special attention in deep submicron circuits is the **power dissipation** [3]. Even if the total active capacitance to be switched per unit of area stays constant with the down-scaling, it turns out that the power dissipation of the chip increases because of the increased operating frequency. The most efficient method to reduce the power consumption is to reduce the supply voltage, as

$$P = \alpha C V_{DD}^2 f \tag{1.3}$$

where α is the switching activity factor, C is the switched capacitance, and f is the operating frequency. V_{DD} cannot be reduced anymore as to preserve the already fragile noise margin.

One can find an accurate estimator for the switching activity factor, provided that he has an efficient model and an efficient method to simulate the power consumption. We present such a model incorporating the static as well as the dynamic behavior of the deep submicron transistor in Chapter 3 and in Chapter 6. We state the need for level transparency when simulating complex circuits and consider one such simulation method in Chapter 2. □

3. At short channel length another phenomenon looms large in the deep submicron technology. Its effect is the decrease of the MOS transistor threshold voltage as the distance between source and drain becomes shorter. This effect requires special consideration during device design, because it makes visible the subthreshold current contribution to the total power consumption. Process tolerances cause statistical deviations of the channel length across different dies, therefore the designer must ensure that the threshold voltage does not become too low for the device with the shortest channel length on the chip.

To keep this short-channel effect under control, gate **oxide thickness** is reduced nearly in proportion to channel length. This reduction is necessary for the gate to retain more control over the channel than the drain does. Thin oxide raises then serious concerns about leakage due to direct tunneling of the current through such an extremely thin layer [4, 5]. □

1.3 Interconnect

Interconnect wires are responsible for the communication on the chip. To take advantage of the scaled-down dimensions of the active devices, wiring pitches are dropping rapidly and at about the same rate as the gate length. This means that both the wire width and the wire spacing are scaled down. Deep submicron interconnect effects, among which the most important are the RC delay, the cross-coupling noise, and the electromigration, rise a variety of problems to process engineers, circuit designers, and CAD tool developers.

1. The most commonly cited deep submicron interconnect problem is the rising **RC delay** [6], or the long-wire delay problem [7]. The increased line resistance is the main reason behind the increased wire delay in deep submicron.

Wiring resistance scales inversely with the wire's cross sectional area. To keep resistance from increasing too quickly, two approaches can be taken: (1) scale the line height at a slower rate than the wire width, and (2) use better conductors for the on-chip interconnect.

Wiring capacitance is also increasing in scaled down processes due to the higher interconnect densities in modern chips. As a reduced packing density is not an option in deep submicron, the only method to reduce the wiring capacitance is to use a material with a lower dielectric constant than the silicon dioxide, material

commonly known as a low-k material.

Apart from the technological solutions, i.e., an increased interconnect aspect ratio and materials with higher conductivity, also circuit level solutions are possible. Various techniques have been developed to improve the communication on the chip, such as buffer insertion, shielding, differential signalling. Yet the interconnect tends to dominate the speed performance on a deep submicron chip. □

2. As mentioned earlier, one way to reduce the resistance has been to slowly scale line thickness, resulting in taller, thinner wires. These high aspect ratio interconnect tracks have a detrimental side effect in that they result in a large amount of coupling capacitance. With an aspect ratio greater than one, lines tend to have more coupling capacitance to neighboring wires in the same layer than to upper and lower wiring layers, which effectively serve as ground planes.

This increased coupling capacitance is what makes the **cross-coupling noise** an issue for deep submicron interconnects. It comes in two distinct forms: cross-coupling delay and crosstalk. The cross-coupling delay means a delay deterioration, as the total load capacitance of a gate is no longer a constant value, due to the Miller effect [6].

For the deep submicron technology, the increased coupling capacitance between adjacent wires and the reduced supply voltage are erosion factors for the fragile noise margin. They enable the undesired analog aspects to permeate the digital circuitry operation, endangering the originally robust digital abstraction.

Simulating deep submicron circuits such that to capture this effect, calls for a transparent way to model both digital and analog behavior and also compact way so that devices and interconnect can be included in the same analysis. We believe that the modeling and simulation approach presented in this thesis is very suitable for this purpose. □

3. Beside the beneficial effect on the wire resistance, better conductors also have the advantage of increased resistance to **electromigration** effects. Electromigration is a reliability problem associated with large current densities. The large current physically moves metal ions down the wire, resulting in opens in the line or shorts to adjacent wires. By using copper in place of the traditional aluminum, the electromigration lifetime until failure can be increased up to two orders of magnitude [8].

1.4 Complexity

An inherent implication of the reduced feature size in the deep submicron technologies is the availability of hundreds of millions transistors on one chip. This opens the possibility to integrate very complex systems on the same die, but also rises the challenge of managing the complexity of such systems during the design process. In the last years engineers realized that the growth enabled by the deep submicron technology is slowed down by the `design productivity gap`. This means that the pace at which systems-on-a-chip (SOC) can be designed is not quick enough to exploit all the integration capabilities of the new technologies.

Lately engineers talk about system-on-a-chip design methodologies as being the only way to develop huge systems on a single silicon substrate in a short design cycle. Nowadays, such a methodology begins at system level through planned **reuse** [9] of functional components including programmable processors, mixed signal functions, and software. In long term designers will continue to move their design efforts to higher levels of abstraction. Designing better architectures is what will generate the next large advance in systems-on-a-chip design [10].

Designers must make architectural decisions at system level: IP[3] selection, hardware/software partitioning, performance analysis, and functional verification. At the moment there are several EDA tools dealing with these issues. The evaluation of power and timing at system level is preferred to transistor level. The current transistor level tools are indeed more accurate, but not fast enough to handle large designs.

Managing the complexity in an efficient way calls for (1) compact models, and (2) one single algorithm for multi-level simulation.

Our approach to compact modeling for the deep submicron device allows us to verify large circuits at a high level of accuracy, comparable with the one achieved in the classical transistor level modeling, but at considerably lower price in terms of computation. Yet, the most important contribution that our work brings to system design is the possibility to co-simulate, co-verify mixed circuits. The key is the level transparency *in modeling and simulation of complex systems-on-a-chip, notion which we introduce in Chapter 2, we reconsider in Chapter 5, and then extend in an original manner in Chapter 7.*

[3]Intellectual Property

1.5 This thesis

This thesis describes our research work on modeling and analysis of large size deep submicron CMOS circuits. From all the challenges that have been mentioned towards the end of Section 1.1 we give an answer only to those which can be approached from circuit level.

Such an approach would incorporate the most important limiting factor in the down-scaling – the saturation of the drive current. It would allow us to accurately estimate the power dissipation of the deep submicron circuits. It would provide a way to include both digital and analog behavior under the same paradigm, so that devices and interconnect can be included in the same analysis; this might be used in dealing with cross-coupling noise. It would certainly provide the means to deal with part of the problems raised by the increased complexity of the systems-on-a-chip, i.e., the efficient verification of large circuits at high level of accuracy and the co-simulation/co-verification of mixed digital/analog circuits.

In Chapter 2 we state the need for level transparency for the efficient simulation of systems-on-a-chip, i.e., circuits with large size and with descriptions across different abstraction levels, as digital and analog. We introduce a candidate modeling and simulation framework based on a classical simulation approach.

In Chapter 3 we analyze the fundamental implications of reducing the MOS transistor length into the deep submicron regime. We contrast the static behavior of the short channel transistor to the one of the long channel transistor, presented for reference in Appendix B. We propose a compact static model for the deep submicron transistor, model which is minimal for the explanation of the transistor level operation of the CMOS circuits.

We realize that this MOS transistor model, as implied by the deep submicron reality, is well fitted with the simulation framework introduced in Chapter 2. We show how the model is implemented in this simulation framework in Chapter 4.

In Chapter 5 we address modalities for dynamic simulation. Reconsidering the need for efficient simulation we point out another modeling and simulation framework, which comes out to be best fitted with our need.

We complete the modeling of the deep submicron MOS transistor in Chapter 6 where we discuss the dynamic operation and we derive an analytically sound model for the large-signal terminal charge variations.

In Chapter 7 we discuss the implementation of the compact models, static

and dynamic, in the simulation framework from Chapter 5. Here we propose a simulation package with a larger applicability area than for CMOS. We devise a methodology to calibrate the models for accurate power and waveform estimation.

In Chapter 8 we indicate from a pragmatic perspective how our modeling and simulation approach would have to be used for the efficient analysis of large, complex systems. On a couple of test designs, we assess the accuracy and the efficiency of our approach. In this chapter we also summarize our findings and final thoughts regarding efficient modeling and analysis of large deep submicron circuits.

In Chapter 9 we comment on the future trend of down-scaling, from the point of view of the achievable performance increase.

In Appendix A we present some methods used for solving the linear complementarity problem that we mentioned in Chapter 2. In Appendix B we shortly present the classical concept of the long-channel MOS transistor operation. Here the reader can find some prerequisites for Chapter 3. In Appendix C we recall some basic VHDL language concepts that we use for the presentation in Chapter 7. In Appendix D we recall the main design and implementation issues of the phase-lock loop, used as a demonstration vehicle in Chapter 8.

The book also contains a list of bibliographic references regarding the physics and operating principles of the deep submicron MOS transistor, piecewise linear and event-driven simulation paradigms, modeling languages, and related subjects in the Bibliography chapter. Further, the Glossary explains the abbreviations, acronyms, symbols and a couple of terms frequently used throughout this book.

Chapter 2

Level transparency

Contents

2.1	System description	12
2.2	Direct methods	14
2.3	Local linearization	15
2.4	Locally linear systems	17
2.5	A piecewise linear approach	18
2.6	One algorithm	23

Deep submicron technologies promise the possibility to integrate very complex systems on a single chip. The equivalent transistor count for such systems-on-a-chip is huge, in the order of one hundred million. The manipulation of a system of such size during various design phases, such as simulation or verification, cannot always be performed at transistor level as a whole, because of the overwhelming complexity. Moreover, the amount of detail produced by simulations at that level is rarely manageable, and therefore hardly useful.

As a result, large parts of the system have to be represented at several levels of abstraction, i.e., gate level, register level, behavioral or higher. However, some smaller parts of the system, where continuous aspects of the operation cannot be abstracted into a logic description, remain to be represented at transistor level only.

Consequently, a system-on-a-chip simulation approach has to accommodate multiple levels of abstraction. In this way:

- the simulation becomes feasible, as the more abstract representations are more compact, demanding less computational resources;
- the degree of detail of a transistor level representation is provided at the places where needed;
- the simulation is precise enough, as the more abstract levels ensure a realistic signal environment for the transistor level.

A straight solution would be to use separate level-specific simulation engines, and to have these engines communicate with each other. The computational efficiency of such a solution depends heavily on the communication speed between processes at different levels. Hence, it would be best if we could have a common modeling language for all descriptions and a single algorithm to manipulate these descriptions. In this way, one engine would be sufficient for the system-on-a-chip simulation, provided that acceptable efficiency can be achieved. The capability of a simulation engine to use one single algorithm for manipulating multiple levels of description is called *level transparency.*

2.1 System description

At a specified hierarchical level a system is defined as a union of components which interact according to some topological and numerical rules. The system communicates with the environment through stimuli and responses. Each component has an intrinsic operation, reflected in a set of constitutive relations. In general, each component "inherits" the structure of the system it is part of, in the sense that each component may be regarded as a union of subcomponents which are interacting according to a set of topological rules, and that for each component one may define an environment to communicate with (see Figure 2.1). At the lowest level of hierarchy the system consists of atomic components that are no further decomposed. Atomic components are primitive systems which are described completely by constitutive relations.

The rules of interaction between components together with the individual behavior of each component give the functionality of the system. In the general case of a non-linear system both the components and the interactions between them are described by non-linear relations.

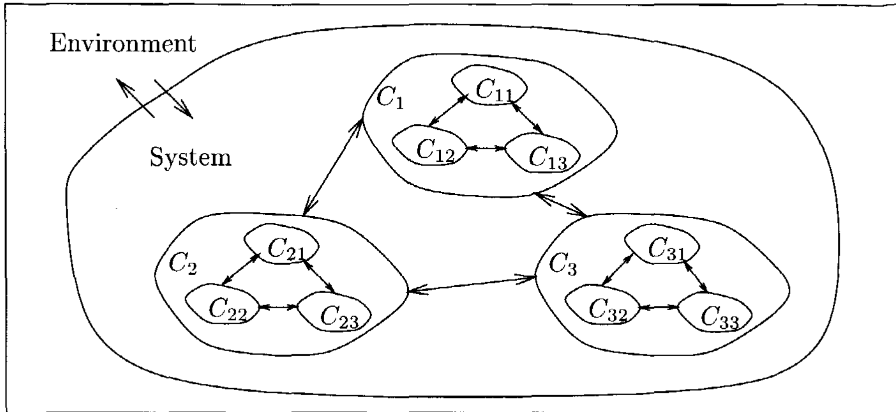


Figure 2.1: Component interaction inside a system. Each component C_i is a union of subcomponents C_{ij}. The arrows indicate interactions.

When the lumped circuit approximation is valid[1] the topology of an electrical system is well described by Kirchhoff's voltage and current laws. The branch voltages are related with the node voltages via the Kirchhoff voltage law. The Kirchhoff current law is an electric charge conservation law: the exchange of charge between the components connected to a node is done in a conservative way, i.e., the amount of charge which enters a node is the same as that which leaves it. Associating the flow of charge with the electric current, Kirchhoff current law states that the sum of currents which enter a node must be equal to the sum of currents leaving that node. Both, Kirchhoff current law and Kirchhoff voltage law, are of a linear nature and thus the topological equations for an electrical system are linear. In the case of a lumped electric circuit each atomic component is described by a dependency, the constitutive equation, between its terminal currents and voltages. This dependency is often of a non-linear nature.

[1]This depends on the size of the atomic components, their physical distance and the transients (frequencies) in the circuit.

2.2 Direct methods

It appears that a lumped electric circuit is completely specified if we know the constitutive equations of each component and the topology of the system (how the components interact). In order to simulate such a circuit, *direct methods* seek to solve two problems:

1. aggregate the constitutive equations according to the topology into some sort of global description, and
2. find a way to manipulate this global description, as for any input stimuli one should be able to compute the system response.

The latter task is performed by solving for a sequence of time points a set of equations that represent the behavior of a non-linear resistive network. This set has the form

$$\mathbf{f}(\mathbf{x},\mathbf{y}) = \mathbf{0} \quad \equiv \quad \begin{cases} f_1(\mathbf{x}, y_1, y_2, \ldots, y_n) = 0 \\ f_2(\mathbf{x}, y_1, y_2, \ldots, y_n) = 0 \\ \vdots \\ f_n(\mathbf{x}, y_1, y_2, \ldots, y_n) = 0 \end{cases} \tag{2.1}$$

where $\mathbf{f}$ is a vector of functions, $\mathbf{x}$ is the stimuli vector, and $\mathbf{y}$ is the response vector. The contents of the system response, $\mathbf{y}$, is dependent on the particular method (nodal, tableau, modified nodal) used to build (2.1). Each function $f_i(\mathbf{x},\mathbf{y})$ is a linear combination (topological equations) of possibly non-linear functions (constitutive equations), thus f_i is in general non-linear in $\mathbf{x}$ and $\mathbf{y}$. The set (2.1) describes a non-linear algebraic set of equations. Apart from special cases, usually of small size, the solution of this set cannot be obtained in a closed form and computer techniques are used to solve for $\mathbf{y}$.

This poses two problems, namely how to represent the non-linearities in a computer, and how to solve the set of equations. The two are of course not unrelated. Methods using the analytic properties of the non-linear functions need access to the Jacobian matrix or have the capability of approximating all partial derivatives. An elementary method for multidimensional problems that works well for finding a root is the *Newton-Raphson method* provided a sufficiently close guess of the solution is available. Otherwise one has to resort to globally more convergent methods,

which are without exception much more complicated. Fortunately, the way time-continuous circuit simulation is organized, that is from time point to sufficiently close time points, makes it easy to satisfy that condition, and therefore Newton-Raphson procedures are in the core of the the most popular transistor level circuit simulators since the seventies.

2.3 Local linearization

For a given stimuli vector $\mathbf{x}$ the Newton-Raphson method produces in an iterative way a sequence of vector values $\mathbf{y^0}$, $\mathbf{y^1}$, ... , $\mathbf{y^k}$, $\mathbf{y^{k+1}}$, ... which converges to the exact solution $\mathbf{y}^*$. $\mathbf{y^0}$ is the initial guess and is usually taken to be the solution of the previous stimuli set. As the computing resources are limited, infinite sequences cannot be afforded; a certain $\mathbf{y^a}$ will be declared the approximate solution if it satisfies given accuracy requirements (absolute and/or relative error less than some imposed values).

We will describe briefly the k-th iteration step: build $\mathbf{y^{k+1}}$ starting from $\mathbf{y^k}$, for fixed input stimuli $\mathbf{x}$. For the duration of the current iteration process the contribution of the fixed $\mathbf{x}$ can easily be absorbed in the constant part of $\mathbf{f}$. Consider the Taylor expansion of $\mathbf{f}(\mathbf{y})$ around $\mathbf{y^k}$:

$$\begin{aligned}
f_1(\mathbf{y}) &= f_1(\mathbf{y^k}) + \frac{\partial f_1}{\partial y_1}\bigg|_{\mathbf{y^k}}(y_1 - y_1^k) + \cdots + \frac{\partial f_1}{\partial y_n}\bigg|_{\mathbf{y^k}}(y_n - y_n^k) + \ldots \\
f_2(\mathbf{y}) &= f_2(\mathbf{y^k}) + \frac{\partial f_2}{\partial y_1}\bigg|_{\mathbf{y^k}}(y_1 - y_1^k) + \cdots + \frac{\partial f_2}{\partial y_n}\bigg|_{\mathbf{y^k}}(y_n - y_n^k) + \ldots \\
&\vdots \\
f_n(\mathbf{y}) &= f_n(\mathbf{y^k}) + \frac{\partial f_n}{\partial y_1}\bigg|_{\mathbf{y^k}}(y_1 - y_1^k) + \cdots + \frac{\partial f_n}{\partial y_n}\bigg|_{\mathbf{y^k}}(y_n - y_n^k) + \ldots
\end{aligned} \tag{2.2}$$

With the assumption that the terms which contain higher order partial derivatives can be neglected, the Taylor expansion (2.2) can be written in terms of only the first order differences:

$$\mathbf{f}_{\mathrm{L}|\mathbf{y^k}}(\mathbf{y}) = \mathbf{f}(\mathbf{y^k}) + \mathbf{M}\big|_{\mathbf{y^k}}(\mathbf{y} - \mathbf{y^k}) \tag{2.3}$$

The function $\mathbf{f}_{\mathbf{L}|\mathbf{y}^\mathbf{k}}$ represents the local linearization of $\mathbf{f}$ in the vicinity of $\mathbf{y}^\mathbf{k}$. The matrix $\mathbf{M}$ contains the first order partial derivatives of each component of the vector function $\mathbf{f}$ and is called the *Jacobian* matrix of the system.

$$\mathbf{M}\Big|_{\mathbf{y}} = \begin{bmatrix} \frac{\partial f_1}{\partial y_1}\Big|_{\mathbf{y}} & \frac{\partial f_1}{\partial y_2}\Big|_{\mathbf{y}} & \cdots & \frac{\partial f_1}{\partial y_n}\Big|_{\mathbf{y}} \\ \frac{\partial f_1}{\partial y_1}\Big|_{\mathbf{y}} & \frac{\partial f_1}{\partial y_2}\Big|_{\mathbf{y}} & \cdots & \frac{\partial f_1}{\partial y_n}\Big|_{\mathbf{y}} \\ \vdots & \vdots & & \vdots \\ \frac{\partial f_n}{\partial y_1}\Big|_{\mathbf{y}} & \frac{\partial f_n}{\partial y_2}\Big|_{\mathbf{y}} & \cdots & \frac{\partial f_n}{\partial y_n}\Big|_{\mathbf{y}} \end{bmatrix} \tag{2.4}$$

$\mathbf{y}^{\mathbf{k+1}}$ is found as the formal solution of $\mathbf{f}_{\mathbf{L}|\mathbf{y}^\mathbf{k}}(\mathbf{y}) = 0$:

$$\mathbf{y}^{\mathbf{k+1}} = \mathbf{y}^\mathbf{k} - \mathbf{M}^{-1}\Big|_{\mathbf{y}^\mathbf{k}}\mathbf{f}(\mathbf{y}^\mathbf{k}) \tag{2.5}$$

The solution (2.5) is not practical for computer manipulation because it involves the inversion of the Jacobian. A better way to find $\mathbf{y}^{\mathbf{k+1}}$ is to consider the Newton-Raphson equation:

$$\mathbf{M}|_{\mathbf{y}^\mathbf{k}} \cdot \Delta\mathbf{y}^\mathbf{k} = -\mathbf{f}(\mathbf{y}^\mathbf{k}) \tag{2.6}$$

Using the LU-factorization method one solves (2.6) for the correction $\Delta\mathbf{y}^\mathbf{k}$. The result of the k-th iteration step becomes:

$$\mathbf{y}^{\mathbf{k+1}} = \mathbf{y}^\mathbf{k} + \Delta\mathbf{y}^\mathbf{k} \tag{2.7}$$

Thus, one Newton-Raphson iteration step consists essentially of two actions:

1. solve the Newton-Raphson equation (2.6) using linear algebra methods, and

2. update the approximate solution according to (2.7).

The calculation of $\mathbf{f}(\mathbf{y}^\mathbf{k})$ requires virtually no effort once the constitutive equations and topological equations are known. By contrast, the calculation of the Jacobian matrix $\mathbf{M}|_{\mathbf{y}^\mathbf{k}}$ requires the existence of the partial derivatives $\frac{\partial f_i}{\partial y_j}|_{\mathbf{y}^\mathbf{k}}$ and then the capability of manipulating and calculating them. The Newton-Raphson procedure involves at each iteration step the *local linearization* of the function $\mathbf{f}$ in the vicinity of $\mathbf{y_k}$.

2.4 Locally linear systems

One important limitation of the classical Newton-Raphson algorithm as recognized above is that information about the system's Jacobian $\mathbf{M}$ is not directly embedded in the system function $\mathbf{f}$. Overhead is added during the simulation process in order to make this information available.

A substantial simplification of the Newton-Raphson algorithm becomes possible if the constitutive equations of the system components are locally linear, i.e., inside each operating region the relation between stimuli and response is linear, where an operating region is defined by certain ranges for the terminal currents and branch voltages. The union of the operating regions has to cover the whole operating range of the component. When gathering the linear topological equations with the locally linear constitutive equations one obtains a piecewise linear system function $\mathbf{f_{PL}}(\mathbf{x}, \mathbf{y})$. For a specific global operating region l the system function is given by:

$$\mathbf{f^l_{PL}}(\mathbf{y}) = \mathbf{M_l} \cdot \mathbf{y} - \mathbf{w_l} - \mathbf{w} \tag{2.8}$$

The system matrix $\mathbf{M_l}$ and the source vector $\mathbf{w_l}$ are constant throughout a linear operating region l, but change with the operating region; $\mathbf{w}$ is the independent source vector (here enters $\mathbf{x}$). $\mathbf{M_l}$ and $\mathbf{w_l}$ are assembled from the system Kirchhoff current law and Kirchhoff voltage law and from the component piecewise linear constitutive equations.

The application of the Newton-Raphson algorithm to find the solution of $\mathbf{f_{PL}}(\mathbf{y}) = 0$ follows the same main steps as before, but now with the additional advantage that during the iteration step k the Jacobian of the system is identical with the system matrix:

$$\mathbf{M^k} = \frac{\partial \mathbf{f_{PL}}}{\partial \mathbf{y}} = \mathbf{M_l} \tag{2.9}$$

For a piecewise linear system the Jacobian is implicitly stored with the topological equations and the piecewise linear constitutive equations, and is updated at each iteration step together with the system matrix.

Let us consider in some detail iteration step k: starting from $\mathbf{y^k}$ with the representative point situated in a piecewise linear region l, solve the Newton-Raphson

equation which now reads

$$\mathbf{M_l} \cdot \Delta \mathbf{y^k} = -\mathbf{f^k_{PL}} \quad \text{with} \quad \mathbf{f^k_{PL}} = \mathbf{M_l} \cdot \mathbf{y^k} - \mathbf{w^k_l} - \mathbf{w} \tag{2.10}$$

and find out the correction $\Delta \mathbf{y^k}$.

If the calculated next iteration point $\hat{\mathbf{y}}^{\mathbf{k+1}} = \mathbf{y^k} + \Delta \mathbf{y^k}$ stays within the same linear region l, then we are done: $\hat{\mathbf{y}}^{\mathbf{k+1}}$ is the exact solution of the piecewise linear equation system.

If $\hat{\mathbf{y}}^{\mathbf{k+1}}$ exceeds the limits of the current linear region, then the step has to be reduced by weighting $\Delta \mathbf{y^k}$ with $t^k < 1$, such that the point

$$\mathbf{y^{k+1}} = \mathbf{y^k} + t^k \cdot \Delta \mathbf{y^k} \tag{2.11}$$

is on the border between the region l and a neighbor region l'. This is the result of the k-th iteration step.

The next iteration step starts from the neighboring linear region l' with $\mathbf{y^{k+1}}$ as the starting point and is concerned with the solution of the updated Newton-Raphson equation: $\mathbf{M_{l'}} \cdot \Delta \mathbf{y^{k+1}} = -\mathbf{f^{k+1}_{PL}}$. $\mathbf{M_{l'}}$ is the updated system matrix in the new linear region and $\mathbf{f^{k+1}_{PL}}$ is the updated system function in the new linear region.

Where the general Newton-Raphson algorithm performs a local linearization on the system equation at each iteration step, the piecewise linear variant of the Newton-Raphson algorithm runs on an already *locally linear* equation. This means that for the piecewise linear case no extra information about the system Jacobian is needed, as the Jacobian is implicitly stored with the topological equations and the constitutive equations. The Newton-Raphson algorithm finds in the general form an approximate solution, while in the piecewise linear form it finds the exact solution of the equations; there is however the error due to the linearization during modeling.

2.5 A piecewise linear approach

As in the general case (see Section 2.1) each circuit entity is described by a mapping from the space of input variables (stimuli) to the space of output variables (responses). The core idea of the PL concept is to approximate this application by a concatenation of linear segments. The input space is divided into a number

of regions: inside each region the output variables depend linearly on the input variables. Changing of region is reflected in changing coefficients in the linear dependency. The *linear* topological equations and the *linear* constitutive equations inside each region are combined with a *linear* description of the regions. Hyperplanes are used as boundaries for regions. In this way we obtain what is called *polytopes* - regions expressed in terms of linear equations. This technique based on a linear dependency inside each linearly characterized region is called the *piecewise linear technique.*

The piecewise linear modeling problem for a generic entity can be formulated in the following manner. Given:

1. a partition of the input space $\mathbf{X}$ into a set of adjacent convex polytopes $\mathbf{P_i}$,
2. a mapping $\mathcal{F}$ from the input space $\mathbf{X}$ to the output space $\mathbf{Y}$ continuous along every boundary which separates the polytopes, and
3. a linear dependence of the output vector $\mathbf{y}$ on the input vector $\mathbf{x}$ inside each polytope $\mathbf{P_i}$

find a global analytical representation for $\mathcal{F}$ suitable for computer manipulation.

An elegant solution to this problem is the implicit global piecewise linear technique based on diode states [11], that we adopt in the remainder of this section. It is a mechanism to keep track of the operating region for the piecewise linear Newton-Raphson approach, with the aid of the so-called diode variables. The approach based on diode states emerged from a network perspective, with the ideal diode as representation for the basic nonlinearity. An ideal diode (see Figure 2.2a for notations and characteristic) operates in one of two states:

- conducting (any positive current implies a zero bias voltage), when it represents a closed connection between its terminals, or
- blocking (a zero current is prescribed for any reverse voltage), when it represents an open circuit between its terminals.

A circuit composed of linear components (sources and resistors) and ideal diodes, is suggested in Figure 2.2b. When the input is changed the *diodes* might switch *state* and the topology of the circuit changes accordingly. For each possible combination of diode states the circuit has a certain topology with linear components, and therefore a linear transfer function.

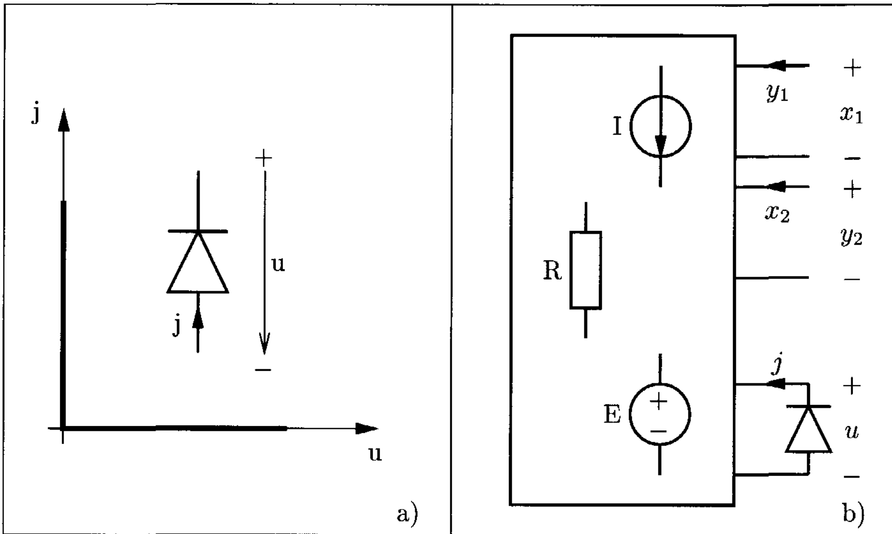


Figure 2.2: Ideal diode a) current-voltage convention and characteristic b) relation to a PL circuit

More precisely, each polytope $\mathbf{P_i}$ is associated with a diode state $\mathbf{S_i}$. The state of the diodes depends (not necessarily uniquely) on the input. The output is a function of the input and the diode state: $\mathbf{y} = \mathcal{F}(\mathbf{x}, \mathbf{S}(\mathbf{x}))$. When $\mathbf{x}$ crosses the boundary from $\mathbf{P_i}$ to $\mathbf{P_j}$ the diode state changes from $\mathbf{S_i}$ to $\mathbf{S_j}$ and the coefficients of the linear mapping should change as well.

The problem can thus be decomposed into:

1. find a state $\mathbf{S}(\mathbf{x})$ of the diodes corresponding to the input vector, and

2. find the output vector $\mathbf{y}$ by using the linear output mapping corresponding with the solution of $\mathbf{S}(\mathbf{x})$.

This formulation is called the **diode state model** of the piecewise linear system [12] and is represented in Figure 2.3. The diode state $\mathbf{S}$ may be encoded with the *diode vectors* $\mathbf{u}$ and $\mathbf{j}$. Diode vectors are orthogonal and nonnegative. The diode state $\mathbf{S}$ becomes a function of diode variables $\mathbf{u}$ and $\mathbf{j}$, and remains an implicit

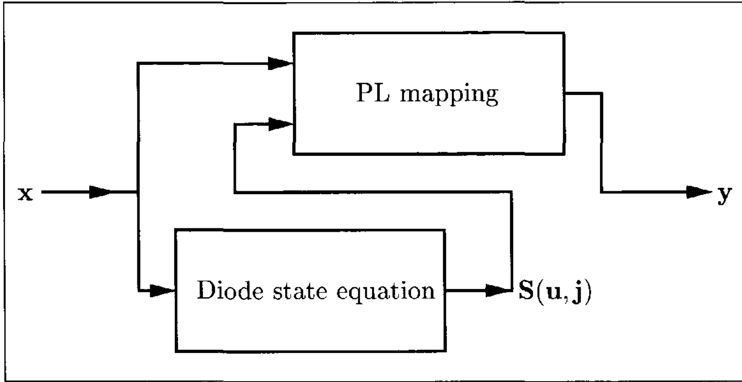


Figure 2.3: Diode state model for a piecewise linear system

function of the input vector $\mathbf{x}$.

$$\begin{cases} \mathbf{S} = \mathbf{S}(\mathbf{u},\mathbf{j}) \quad \text{with} \quad \mathbf{u} = \mathbf{u}(\mathbf{x}), \quad \mathbf{j} = \mathbf{j}(\mathbf{x}) \\ \mathbf{u}^{\mathrm{T}} \cdot \mathbf{j} = \mathbf{0}, \mathbf{u} \geq \mathbf{0}, \mathbf{j} \geq \mathbf{0} \end{cases} \tag{2.12}$$

The introduction of the additional variables $\mathbf{u}$ and $\mathbf{j}$ enables the compact analytical representation:

$$\mathbf{y} = \mathbf{A} \cdot \mathbf{x} + \mathbf{B} \cdot \mathbf{u} + \mathbf{f} \tag{2.13a}$$

$$\mathbf{j} = \mathbf{C} \cdot \mathbf{x} + \mathbf{D} \cdot \mathbf{u} + \mathbf{g} \tag{2.13b}$$

$$\mathbf{u}^{\mathrm{T}} \cdot \mathbf{j} = \mathbf{0}, \mathbf{u} \geq \mathbf{0}, \mathbf{j} \geq \mathbf{0} \tag{2.13c}$$

where $\mathbf{A}$, $\mathbf{B}$, $\mathbf{C}$, and $\mathbf{D}$ are matrices and $\mathbf{f}$ and $\mathbf{g}$ are column vectors. This is the most general piecewise linear description and was proposed by Van Bokhoven [13].

Relation (2.13a) is called the *linear input-output mapping*, equation (2.13b) is the *diode state equation*, and the set (2.13c) represents the *complementary constraints.* Relations (2.13b) and (2.13c) form together the so called *linear complementarity problem.* This problem [14] has been identified in a broader context: for a given source vector $\mathbf{q}$ find the orthogonal nonnegative vectors $\mathbf{w}$ and $\mathbf{z}$ that satisfy:

$\mathbf{w} = \mathbf{M} \cdot \mathbf{z} + \mathbf{q}$.

With $\mathbf{x}$ in the polytope $\mathbf{P}^*$, for which $\mathbf{u} = \mathbf{0}$, the input-output relation reads simply:

$$\mathbf{y} = \mathbf{A} \cdot \mathbf{x} + \mathbf{f} \tag{2.14}$$

The description (2.13) is then called the *natural representation* for the specific polytope $\mathbf{P}^*$. When the input vector leaves that specific polytope $\mathbf{P}^*$ at least one component of vector $\mathbf{u}$, say u_k, becomes positive and relation (2.14) is not valid anymore. However, the piecewise linear mapping (2.13a) is valid across the entire input space. If we now interchange j_k and u_k in relation (2.13b) - an operation called *pivoting* – and we transform (2.13a) to reflect this change, we obtain another natural representation valid in the newly entered polytope. This is the core idea of the so-called "path-following algorithms" for solving the linear complementarity problem. Various algorithms have been developed to solve the linear complementarity problem (see Appendix A).

In the representation (2.13) the diode state equation reflects the geometry of the input space. The hyperplanes are defined only in terms of the input variables. Such a hyperplane equation reads as $\mathbf{C_{k*}} \cdot \mathbf{x} + \mathbf{g_k} = \mathbf{0}$. This description is suited when the input space is divided into polytopes by full hyperplanes only and the mapping is continuous over changing the polytope. In those cases the diode state equation (2.13b) is easily obtained by inclusion of hyperplane equations.

From a modeling point of view it may be more convenient to consider the input-output space for the boundary equations. In terms of input space variable we may end up with half-hyperplanes or piecewise hyperplanes. If the result is a full hyperplane in terms of input and output variables, then we can use a slightly modified version of (2.13):

$$\mathbf{0} = \mathbf{I} \cdot \mathbf{y} + \mathbf{A} \cdot \mathbf{x} + \mathbf{B} \cdot \mathbf{u} + \mathbf{f} \tag{2.15a}$$

$$\mathbf{j} = \mathbf{D} \cdot \mathbf{y} + \mathbf{C} \cdot \mathbf{x} + \mathbf{I} \cdot \mathbf{u} + \mathbf{g} \tag{2.15b}$$

$$\mathbf{u^T} \cdot \mathbf{j} = \mathbf{0}, \mathbf{u} \geq \mathbf{0}, \mathbf{j} \geq \mathbf{0} \tag{2.15c}$$

In (2.15b) the diode state equation is written in the input-output space, or

in other words the boundary hyperplanes are defined in terms of both $\mathbf{x}$ and $\mathbf{y}$. The equation of such a hyperplane is $\mathbf{D_{k*}} \cdot \mathbf{y} + \mathbf{C_{k*}} \cdot \mathbf{x} + \mathbf{g_k} = \mathbf{0}$. j_k and u_k are the complementary variables supposed to change when the representative point in the $\mathbf{x}$-$\mathbf{y}$ space crosses this boundary.

2.6 One algorithm

Any network that can be modeled by linear components and ideal diodes, can be handled by simulators that can solve the ensuing linear complementarity problem. Once inside the polytope the problem is reduced to solving a set of linear equations. It is not important what level of abstraction has been modeled: the components can be switches, transistors, gates, etc. As long as they conceptually fit in the framework of (2.13), the same algorithm can be used to find an operating point that satisfies the constraints and the input-output mapping.

In that sense piecewise linear simulators can be considered level transparent, although the requirement to obtain suitable piecewise linear models for every non-linear component at every level should not be deemed simple and facile. Also, automatically "zooming" in and out by changing local levels is far from simple. But, once every component is represented in the framework at the adequate level of detail, the same simulation engine can do the job. Unlike methods with Newton-Raphson iterations in the kernel storage or calculating the Jacobian matrix is not necessary, and global convergence to an existing operating point is much better.

Chapter 3

Transistors in the limit

Contents

3.1 Impact of velocity saturation 27
3.2 Model equations 29
3.3 Scaling down 34
3.4 Saturated velocity 38
3.5 Piecewise linear transistor models 40

Any simulator for electronic circuits should be capable of including and handling amplifying devices, i.e., a signal imposed at the input of that device should appear amplified (or restored) at the output. This is the so-called *triode operation*, invented at the beginning of last century and catalyzing the creation of the largest industry ever during that century.

A triode has at least three terminals: an input terminal, the *gate*, an output terminal, the *drain*, and a common terminal, the *source*[1]. They come in two types, n-type and p-type triodes. The operational principle is to create and sustain mobile charge carriers in the current path between drain and source by bringing an opposite charge on the gate, close to that path. The number of carriers can be controlled by the amount of charge. Under the influence of an electric field created

[1]The naming conventions depend on the triode realization: vacuum tubes have *grid*, *anode* and *cathode*, while bipolar junction transistors have *base*, *collector* and *emitter*.

by potential difference between drain and source, carriers can form a current between those terminals. In n-type triodes the carriers are predominantly negatively charged (electrons), while in p-type triodes they are positively charged. Triodes can be classified according by the way they prevent carriers from flowing into the gate [15].

The discussion in this thesis is limited to triodes that are built by a three layer structure: a conducting gate, separated from an extrinsic semiconductor channel by a thin isolator. The well-established theory of these so-called *MOS-transistors* for relatively long distance between drain and source is summarized in Appendix B. Here we are interested in the behavior for relatively short distances between those terminals, but still long enough to maintain the same operational principles.

In the n-type MOS transistor positive charge should be moved to the gate electrode to attract negative mobile electrons in the channel. This requires a minimum amount of charge on the gate, but from then on, the more charge is on the gate, the more mobile electrons are available for the device current. Under ideal operating conditions the gate is perfectly isolated from the conducting channel and the positive charge moves in and out of the gate without losing any charge to elsewhere in the device. The gate charge attracts an equal amount of mobile charge of opposite sign, that can be used for current under the influence of the drain-source field, which cycles through the closed loop of load circuitry and power supply. The device current flows in a thin surface *sheet* of the silicon.

In a scaled-down technology the MOS transistor suffers from both short-channel and narrow-channel effects [16, 17]. Complex models, exhaustive in capturing the MOS transistor operation, are presented elsewhere in the literature [18]. Here we will focus on a modeling approach which is compact and still reflects the essential physical operation.

The channel length scales-down at a faster rate than the supply voltage. As a consequence, the longitudinal electric field in the channel increases with the technology generation and the charge carriers travel at a velocity closer to the saturation velocity. It is an hypothesis of this thesis that velocity saturation is the overwhelming effect on the current drive of transistor, an hypothesis that has to be tested.

If not further specified: "transistor" means n-type MOS-transistor. The names of the channel contacts, drain and source, is with reference to the circuit configuration, and not as usual, with reference to the flow direction of the carriers.

They are denoted D and S respectively. G denotes the gate. B stands for the bulk, the fourth terminal of the real device.

3.1 Impact of velocity saturation

To study the relative impact of the drift velocity saturation effect [19] on transistor behavior we consider two modeling experiments.

1. Suppose that the charge carriers in the transistor channel are allowed to approach a higher saturation velocity than imposed by the actual material of the channel.

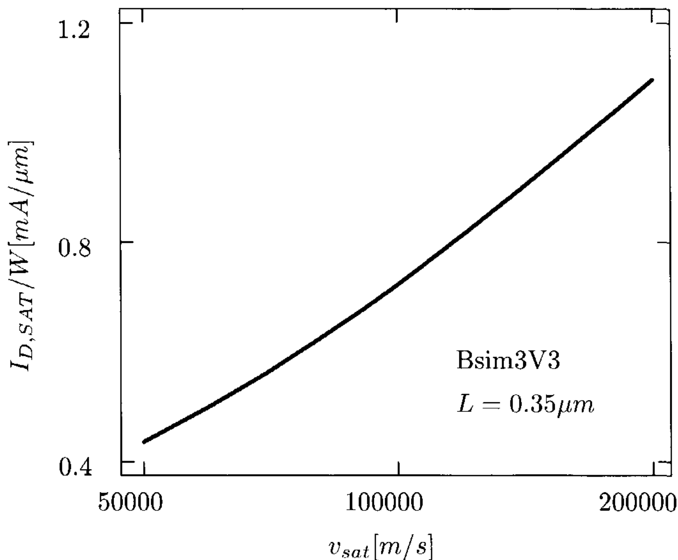


Figure 3.1: Current drive vs. saturation velocity ($v_{sat} \approx 100000[m/s]$).

We would obtain this situation by increasing the saturation velocity parameter v_{sat} in the detailed deep submicron transistor model from [18]. The simulation results in Figure 3.1 show that the current drive $I_{D,SAT}/W$ is affected by velocity saturation, and in such a way the smaller v_{sat} the smaller the current drive.

2. We take two transistor models:

- the same detailed deep submicron transistor model, complete with all the effects which may have an influence on the current drive, including the velocity saturation effect, and
- the first-order charge sheet model enhanced with only the velocity saturation as short-channel effect.

Comparing the current drive $I_{D,SAT}/W$ at various channel lengths L for both models (Figure 3.2) shows that the velocity saturation effect is dominant over other short-channel effects, in what concerns the current drive of a deep submicron transistor.

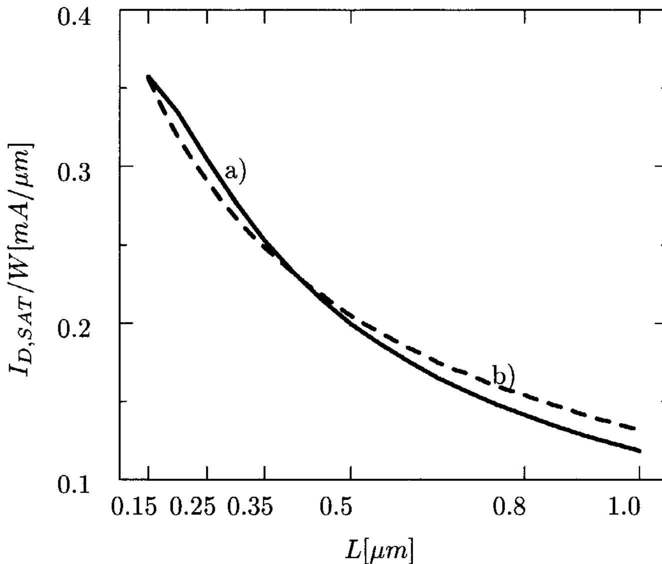


Figure 3.2: Current drive vs. channel length a) detailed model b) compact model

These two experiments show not only that velocity saturation has a significant effect on the transistor current drive, but that it is the dominant factor affecting the transistor current drive when the channel length gets shorter. We therefore

re-derive the model equations of the transistor, but not with constant mobility as in Appendix B, but with the commonly used dependency on the electrical field.

3.2 Model equations

In a channel with high longitudinal electric field the drift velocity (v_d) of the carriers no longer obeys the simple linear dependency expressed by:

$$|v_d| = \mu|E_y| = \mu\frac{d\psi_s}{dy} \cong \mu\frac{dV_{CS}}{dy} \qquad \text{for } V_{DS} \geq 0 \qquad \text{(B.20a)}$$

$$= -\mu\frac{d\psi_s}{dy} \cong -\mu\frac{dV_{CS}}{dy} \qquad \text{for } V_{DS} \leq 0 \qquad \text{(B.20b)}$$

where C is an arbitrary point with strong inversion in the channel. As the electric field is increased, $|v_d|$ exhibits a saturation tendency, i.e., there is a $|v_d|_{max}$ barrier which cannot be passed regardless how strong the electric field. A commonly used empirical approximation [20] for the drift velocity is:

$$|v_d| = |v_d|_{max}\frac{|E|/|E_c|}{1+|E|/|E_c|} = \mu\frac{|E|}{1+|E|/|E_c|} \tag{3.1}$$

where μ is the low field surface mobility of the carriers, $|E|$ is the magnitude of the longitudinal electric field, and $E_c = \frac{|v_d|_{max}}{\mu}$ is the critical field. The saturation of the drift velocity with the electric field as predicted by (3.1) is presented in Figure 3.3. The saturated drift velocity is given by:

$$v_{sat} = |v_d|_{max} = \mu|E_c| \tag{3.2}$$

As in the long-channel case we state that the current conduction through the channel is possible if at least one end of the channel is in strong inversion.

The transistor is said to be in forward conduction when the drain is biased to a higher potential than the source, i.e., $V_{DS} \geq 0$. Normal operation requires that both contact-to-substrate n^+p junctions to be reverse biased: $V_{SB} \geq 0$ and $V_{DB} \geq 0$.

When both ends of the channel are in strong inversion (B.19) for forward

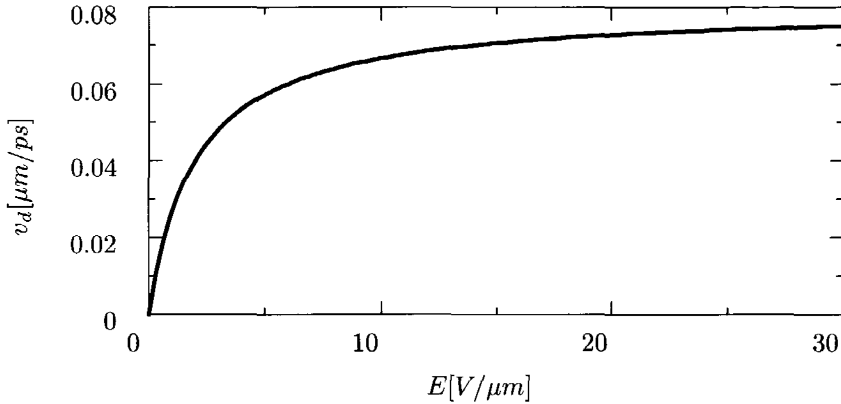


Figure 3.3: Drift velocity vs. longitudinal electric field

conduction with velocity saturation effects becomes:

$$I_D = W|Q_I|\mu\frac{|E|}{1+|E|/|E_c|}, \quad \text{with } |E| \cong \frac{dV_{CS}}{dy} \text{ for } V_{DS} \geq 0 \tag{3.3}$$

After some manipulations we find:

$$I_D dy = \big(\mu W|Q_I| - \frac{I_D}{E_c}\big) dV_{CS}(y) \tag{3.4}$$

With the inversion layer charge given by:

$$Q_I(y) = -C_{ox}\big(V_{GS} - V_T(V_{SB}) - (1+\delta)V_{CS}(y)\big) \tag{B.22}$$

the drain current in forward triode operation for a short-channel transistor becomes:

$$I_D = \mu\frac{W}{L}C_{ox}\frac{(V_{GS}-V_T)V_{DS} - (1+\delta)\frac{V_{DS}^2}{2}}{1+\frac{V_{DS}}{V_0}} \quad \text{for } V_{DS} \geq 0 \text{ and } V_{DS} \leq V_{DS,\,SAT} \tag{3.5}$$

where $V_0 = \dfrac{Lv_{sat}}{\mu}$ is a parameter which accounts for velocity saturation effects. Expression (3.5) is valid as long as I_D keeps increasing with V_{DS}, i.e., for V_{DS} below a certain value $V_{DS,\,SAT}$. The validity limit of (3.5) is found by setting the first derivative of the drain current with respect to V_{GS} equal to zero:

$$V_{GS} = V_T + (1+\delta)V_{DS} + \frac{1+\delta}{V_0}\frac{V_{DS}^2}{2} \quad \text{or} \tag{3.6a}$$

$$V_{DS,\,SAT} = V_0\left(\sqrt{1+\frac{2}{1+\delta}\frac{V_{GS}-V_T}{V_0}}-1\right) \tag{3.6b}$$

The value of the saturation current is found to be:

$$I_{D,\,SAT} = \mu\frac{W}{L}C_{ox}\frac{1+\delta}{2}V_{DS,\,SAT}^2 \tag{3.7}$$

Equation (3.6a) can be seen as the separation between the forward saturation and the forward triode regions of a short-channel transistor. The V_{DS} value at which the current saturation occurs for a given V_{GS} is lower than the value in the long channel situation.

Reverse conduction is possible when V_{SB} is assigned a positive value. Similar to forward conduction, reverse conduction of a short channel transistor shows triode and saturation operating regions when V_{DS} is decreased from zero down to a sufficiently large negative value, with V_{GS} fixed above the threshold voltage. The triode covers a part of the situation with strong inversion at both ends of the channel, namely the part with small V_{DS} magnitudes. The saturation covers the part with higher V_{DS}. For $V_{GS} > V_T$ the source end of the channel is in strong inversion, as appears from the discussion preceding (B.16) about the threshold voltage. Expression (B.22) for the inversion layer charge at the strong inverted points C in the channel is also valid for $V_{CS} < 0$. For reverse drain-to-source bias $V_{DS} < 0$ the drain end of the channel will be in deeper strong inversion than the source end. For reverse triode conduction with velocity saturation effects expression (B.19) becomes:

$$I_D = -W|Q_I|\mu\frac{|E|}{1+|E|/|E_c|} \quad \text{with } |E| \cong -\frac{dV_{CS}}{dy} \text{ for } V_{DS} \le 0 \tag{3.8}$$

Integrating over the channel length one may find the drain current in reverse triode

operation for a short-channel transistor:

$$I_D = \mu \frac{W}{L} C_{ox} \frac{\left((V_{GS} - V_T)V_{DS} - (1+\delta)\frac{V_{DS}^2}{2}\right)}{1 - \frac{V_{DS}}{V_0}} \tag{3.9}$$

The validity of (3.9) extends up to the separation between reverse triode and reverse saturation ($V_{DS} \leq 0$ and $V_{DS} \geq -V_{SD,\,SAT}$). In the (V_{SD}, V_{GD}) plane the separation equation is:

$$V_{GD} = V_T(V_{DB}) + (1+\delta)V_{SD} + \frac{1+\delta}{V_0}\frac{V_{SD}^2}{2} \tag{3.10}$$

$V_{SD,\,SAT}$ is found as the solution of (3.10). $V_T(V_{DB})$ defines the onset of the strong inversion at the drain end relative to V_{GD}. According to (B.16), $V_T(V_{DB})$ relates to $V_T(V_{SB})$ by

$$V_T(V_{DB}) = V_T(V_{SB}) + \delta V_{DS} \tag{3.11}$$

and the expression for $V_{SD,\,SAT}$ becomes:

$$V_{SD,\,SAT} = V_0\left(\sqrt{1 + \frac{2}{1+\delta}\frac{V_{GS} - V_T - (1+\delta)V_{DS}}{V_0}} - 1\right) \tag{3.12}$$

The separation equation (3.10) between reverse triode and reverse saturation may be rewritten in the (V_{DS}, V_{GS}) plane as:

$$V_{GS} = V_T + \frac{1+\delta}{V_0}\frac{V_{DS}^2}{2} \tag{3.13}$$

For $V_{DS} < -V_{SD,\,SAT}$ the transistor is in reverse saturation and the drain current is given by:

$$I_{D,\,SAT}^{REV} = -\mu \frac{W}{L} C_{ox} \frac{1+\delta}{2} V_{SD,\,SAT}^2 \tag{3.14}$$

For a short channel transistor there are two flavors of reverse conduction, depending on the V_{GS} magnitude: (1) if $V_{GS} > V_T$ first reverse triode and then

Operating region	Delimitation in the (V_{DS}, V_{GS}) plane	Drain current expression
Forward saturation	$V_{DS} \geq 0$ and $V_{GS} \geq V_T$ and $V_{GS} \leq V_T + (1+\delta)V_{DS} + \frac{1+\delta}{2}\frac{V_{DS}^2}{V_0}$	$I_D = \mu \frac{W}{L} C_{ox} \frac{1+\delta}{2} V_{DS,\,SAT}^2$
Forward triode	$V_{DS} \geq 0$ and $V_{GS} \geq V_T$ and $V_{GS} \geq V_T + (1+\delta)V_{DS} + \frac{1+\delta}{2}\frac{V_{DS}^2}{V_0}$	$I_D = \mu \frac{W}{L} C_{ox} \frac{(V_{GS}-V_T)V_{DS} - \frac{1+\delta}{2}V_{DS}^2}{1 + \frac{V_{DS}}{V_0}}$
Reverse triode	$V_{DS} \leq 0$ and $V_{GS} \geq V_T + (1+\delta)V_{DS}$ and $V_{GS} \geq V_T + \frac{1+\delta}{2}\frac{V_{DS}^2}{V_0}$	$I_D = \mu \frac{W}{L} C_{ox} \frac{(V_{GS}-V_T)V_{DS} - \frac{1+\delta}{2}V_{DS}^2}{1 - \frac{V_{DS}}{V_0}}$
Reverse saturation	$V_{DS} \leq 0$ and $V_{GS} \geq V_T + (1+\delta)V_{DS}$ and $V_{GS} \leq V_T + \frac{1+\delta}{2}\frac{V_{DS}^2}{V_0}$	$I_D = -\mu \frac{W}{L} C_{ox} \frac{1+\delta}{2} V_{SD,\,SAT}^2$
Nonconducting	$V_{GS} \leq V_T$ and $V_{GS} \leq V_T + (1+\delta)V_{DS}$	$I_D = 0$

$$V_{DS,\,SAT} = V_0\left(\sqrt{1 + \frac{2}{1+\delta}\frac{V_{GS}-V_T}{V_0}} - 1\right),\ V_{SD,\,SAT} = V_0\left(\sqrt{1 + \frac{2}{1+\delta}\frac{V_{GS}-V_T-(1+\delta)V_{DS}}{V_0}} - 1\right),\ \text{and}$$
$$V_T = V_T(V_{SB}).$$

Table 3.1: Velocity saturation effect on the large signal operation of a short channel transistor. Operating region delimitations in the (V_{DS}, V_{GS}) plane and expressions for the drain current.

reverse saturation is encountered when decreasing V_{DS} from zero down to negative values, and (2) if $V_{GS} < V_T$ first non-conduction and than reverse saturation is encountered for a similar excursion of V_{DS}.

Table 3.1 presents a summary of the equations derived in this section.

3.3 Scaling down

In Section 3.1 we showed the dramatic impact of velocity saturation on the saturation current when devices get very short, and consequently V_0 very small. Figure 3.2 showed the increase in current drive capability with shorter channel length, which is repeated in Figure 3.4a. Other effects can, of course, be read from the relations (3.6b) and (3.7), but one aspect becomes very explicit if we look at the mathematical limit of the latter equation:

$$\lim_{V_0 \to 0} I_{D,\,SAT} = W v_{sat} C_{ox} (V_{GS} - V_T). \tag{3.15}$$

The quadratic dependence of the saturation current on $V_{GS} - V_T$ observed (and derived) for long-channel transistors, supposedly changes into a linear dependence when the channel is short. This can already be observed with devices for $V_{GS} > 2V_T$ that are still $1\mu m$.

A semi-empirical expression has been proposed in [21] to capture that phenomenon. It is the so-called α-power law for the drain saturation current:

$$I_{D,\,SAT} \propto (V_{GS} - V_T)^{\alpha}. \tag{3.16}$$

The drain saturation current is considered proportional to the α-power of the gate drive $V_{GS} - V_T$, where both, α and the proportionality constant, depend on L. This implies for a given length a linear relation between the logarithm of the relative saturation current and the logarithm of the relative gate voltage above the threshold:

$$log \frac{I_{D,\,SAT}}{I_{D,\,SAT}^{REF}} = \alpha \; log \frac{V_{GS} - V_T}{V_{GS}^{REF} - V_T}. \tag{3.17}$$

This makes it easy to obtain evidence for the validity of the formula (3.16), and to establish the value of α graphically, and observe how it evolves with decreasing channel length. The latter is presented in Figure 3.4b where the slope of the charac-

teristic on a log-log plot equals α: when scaling a long channel down to a very-short one, the α coefficient moves from two towards one.

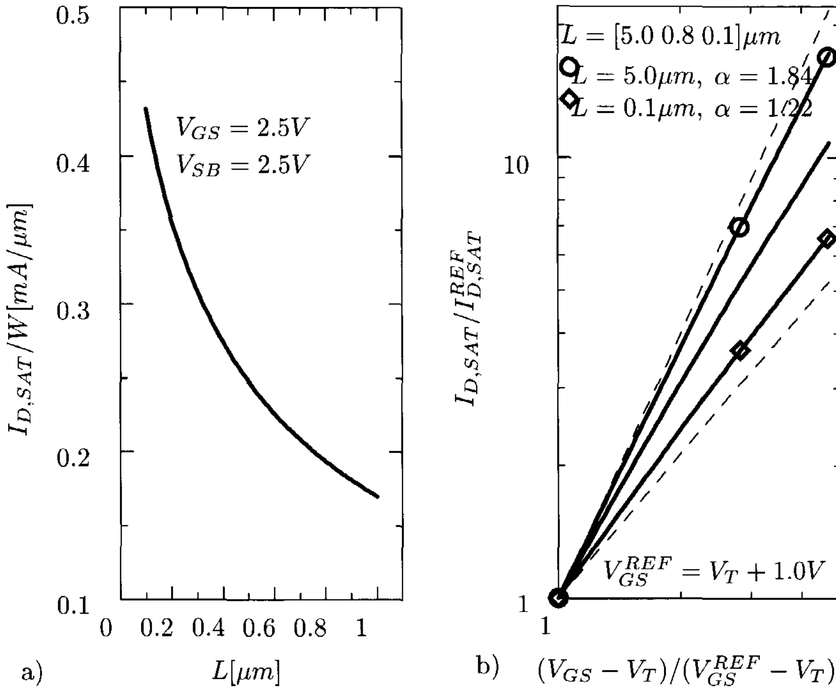


Figure 3.4: The forward saturation current: a) current drive improvement with down-scaling b) variation of the α coefficient with the channel length. The extreme cases of a long ($\alpha = 2.0$) and short ($\alpha = 1.0$) channel are given with thin dashed lines.

Yet another aspect can be uncovered by the study of Figure 3.5 where the operating regions of forward conduction are given in the (V_{DS}, V_{GS}) plane next to I_D vs. V_{DS} characteristic. In the (V_{DS}, V_{GS}) plane we show the separation line between the forward saturation and triode operating regions for three values of the V_0 parameter, 0.1V, 1V, and 10V, values which correspond to a very-short, short, and long channel, respectively. Note the relative positions of point S and of point L, to see that in the case of shorter channels the current saturates at smaller $V_{DS,\,SAT}$. `fsat+` denotes the region with the additional pairs (V_{GS}, V_{DS}) for short-channel devices as compared with the long-channel device. In the I_D vs. V_{DS} plane we see that the onset of saturation occurs the earlier the shorter the channel length.

All operating regions are considered in Figure 3.6, including reverse conduc-

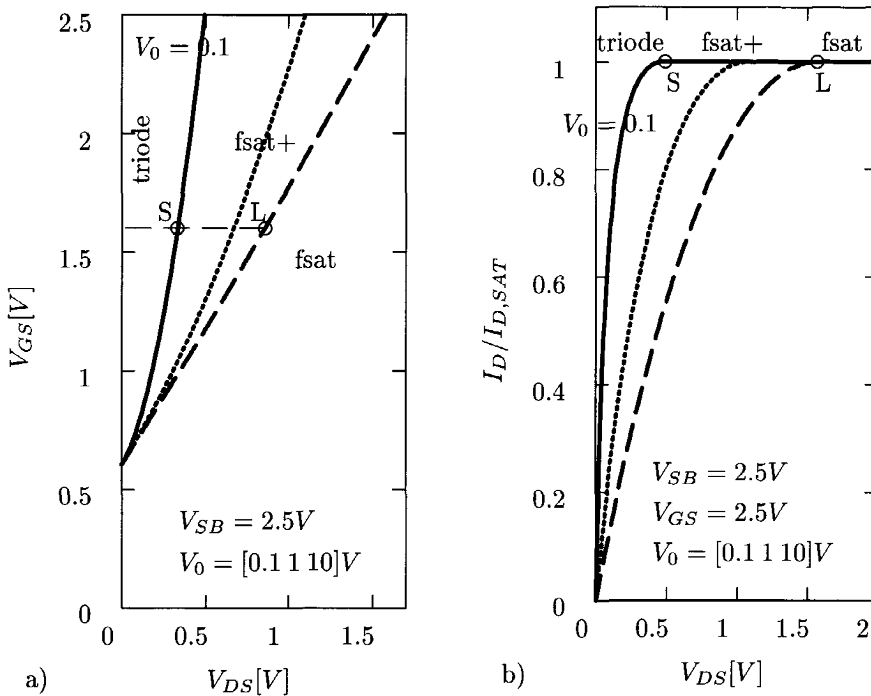


Figure 3.5: Forward conduction for long ($V_0 = 10V$), short ($V_0 = 1V$) and very short ($V_0 = 0.1V$) devices with dashed, dotted and full lines respectively: a) boundary between triode and saturation b) I_D vs. V_{DS} characteristic

tion and no conduction at all. While the separation between non-conducting and forward/reverse saturation remains invariant under V_0-variation, the triode region is getting smaller with decreasing V_0. Both, forward and reverse saturation moves further to smaller voltages as the transistor length is scaled down [22], taking the `fsat+/rsat+` regions away from the triode regime. Eventually the triode contribution to the large signal conduction of a transistor may be disregarded for very-short channel length.

The above discussions and illustrations highlight three effects of down-scaling transistors on their $I_{D,\,SAT}$:

1. the saturation current gets larger,
2. the dependence of $I_{D,\,SAT}$ on $V_{GS} - V_T$ becomes more and more linear ,
3. saturation voltages get lower and lower, reducing the triode region to almost

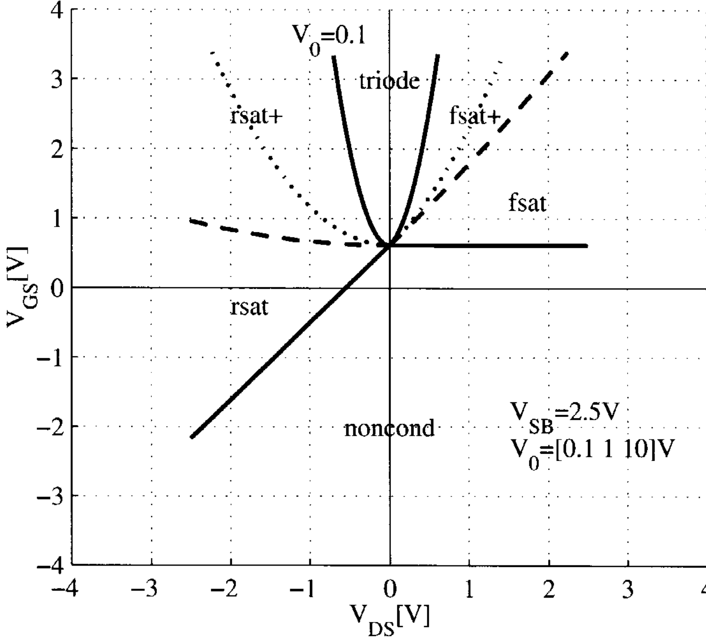


Figure 3.6: Operating regions in the (V_{DS}, V_{GS}) plane for long ($V_0 = 10V$), short ($V_0 = 1V$) and very short ($V_0 = 0.1V$) devices. The separating curves are given with dashed, dotted and full lines respectively. Only the separation between triode and saturation is affected by down-scaling.

nothing.

Two effects are clearly seen in the I_D vs. V_{DS} characteristics for various channel lengths in Figure 3.7. The forward drain saturation current becomes larger, and is reached at lower voltages, while in reverse saturation the drain current depends almost linearly on the drain-to-source voltage[2], as the length of the transistor goes deeper into the submicron region. The other effect appears in the characteristic family for a very-short transistor (Figure 3.8). The gate-source voltage is varied with steps of $0.5V$ which produces equidistant parallel lines in both saturation regions.

[2]The change of convexity for the reverse part of the characteristic appears more clearly on a normalized characteristic.

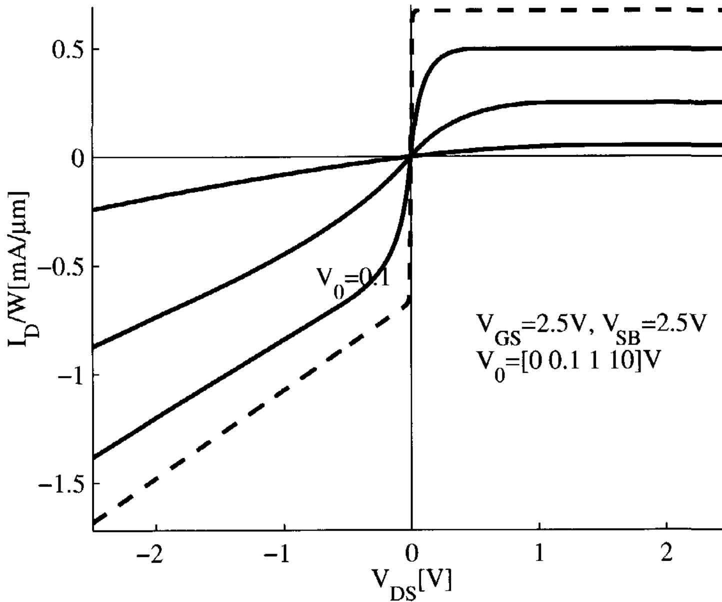


Figure 3.7: I_D vs. V_{DS} characteristic for various lengths including the limiting case

The current saturation for a deep submicron transistor is caused by two concurrent mechanisms: (1) the pinch-off saturation mechanism, which is inherited from the long-channel operation, and (2) the drift velocity saturation mechanism, which is specific to the short-channel operation. By scaling down the transistor length, the second mechanism becomes dominant in the settlement of the current saturation and the characterization of the operating regions suffers important changes.

3.4 Saturated velocity

For the situation when the drift velocity of the charge carriers in the channel gets rapidly into saturation with the longitudinal electric field, we may deduce in a fast manner the equation governing the transistor model. The basic assumption here is that the drift velocity v_d is equal to the saturation velocity v_{sat} for all situations

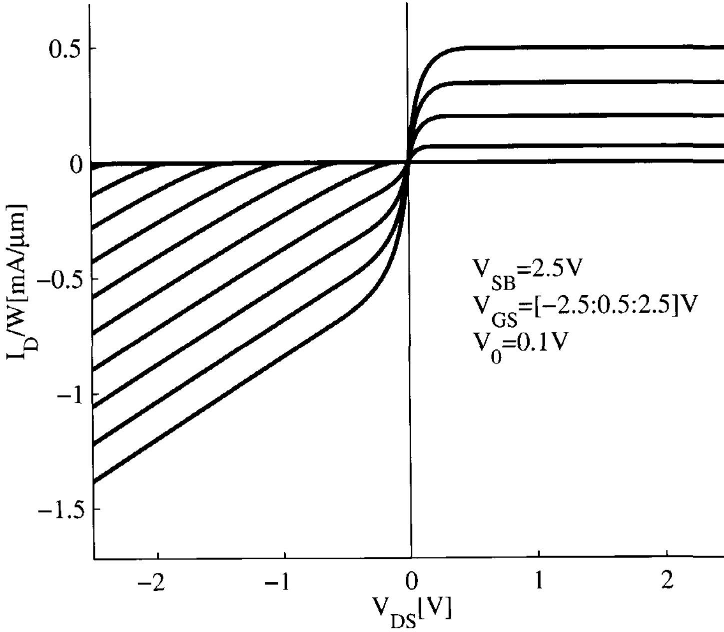


Figure 3.8: I_D vs. V_{DS} characteristic family for a very-short channel transistor

when conduction is possible (strong inversion present at least at one end of the channel).

By substituting $v_d = v_{sat}$ and $|Q_I| = |Q_I^S| = C_{ox}[V_{GS} - V_T(V_{SB})]$ in the general expression for the drain current

$$\mathbf{I_D} = -W|Q_I|\mathbf{v_d} \tag{B.19}$$

we obtain directly the current in forward conduction:

$$I_D^{(F)} = v_{sat}WC_{ox}[V_{GS} - V_T(V_{SB})], \tag{3.18}$$

identical to the mathematical limit in equation (3.15). In reverse conduction the strongest inversion is at the drain end of the channel. With $|Q_I| = |Q_I^D| = C_{ox}[V_{GS} - V_T(V_{SB}) - (1 + \delta)V_{DS}]$ it follows from (B.19) that the drain current

in reverse conduction is:

$$I_D^{(R)} = -v_{sat} W C_{ox}[V_{GS} - V_T(V_{SB}) - (1+\delta)V_{DS}] \tag{3.19}$$

For the case when neither the source end nor the drain end of the channel is strongly inverted ($V_{GS} \leq V_T(V_{SB})$ and $V_{GD} \leq V_T(V_{DB})$) no current will flow through the channel ($I_D = 0$). We gather the above three considerations into one equation:

$$I_D = \begin{cases} v_{sat} W C_{ox}(V_{GS} - V_T) & \text{if } V_{GS} \geq V_T \text{ and } V_{DS} \geq 0 \\ 0 & \text{if } V_{GS} \leq V_T \text{ and } V_{DS} \geq \dfrac{V_{GS} - V_T}{1+\delta} \\ -v_{sat} W C_{ox}[V_{GS} - V_T - (1+\delta)V_{DS}] & \text{if } V_{DS} \leq min(0, \dfrac{V_{GS} - V_T}{1+\delta}) \end{cases} \tag{3.20}$$

where $V_T = V_T|_{V_{SB}=0}$ is considered a constant[3].

3.5 Piecewise linear transistor models

The main feature of the model derived in the previous section is that the essential non-linearities of the device are localized: $V_{DS} = 0$, and $V_{GS} = V_T, V_{DS} \geq 0$, and $V_{GS} = V_T + V_{DS}, V_{DS} \leq 0$. Elsewhere the device behaves completely linear: a current source with linear dependency on $V_{GS} - V_T$ in forward saturation, and a current source with linear dependency on $V_{GS} - V_T$ parallel with a constant resistance in reverse saturation. The triode region is reduced to a vertical line segment in the I_D vs. V_{DS} characteristic on the axis where it follows any forced current between the two saturation currents. A family of characteristics is given in Figure 3.9.

The deep submicron transistor is thus reduced to a current source that collapses when in triode operation [15]. In essence, with suitable (traditional) reference changes, there are only two different conduction regimes:

[3] For the deep submicron technologies, as the circuits are operated at reduced supply voltages, it is legitimate to neglect the body effect; the threshold voltage V_T is considered independent on the source-to-substrate bias voltage V_{SB}.

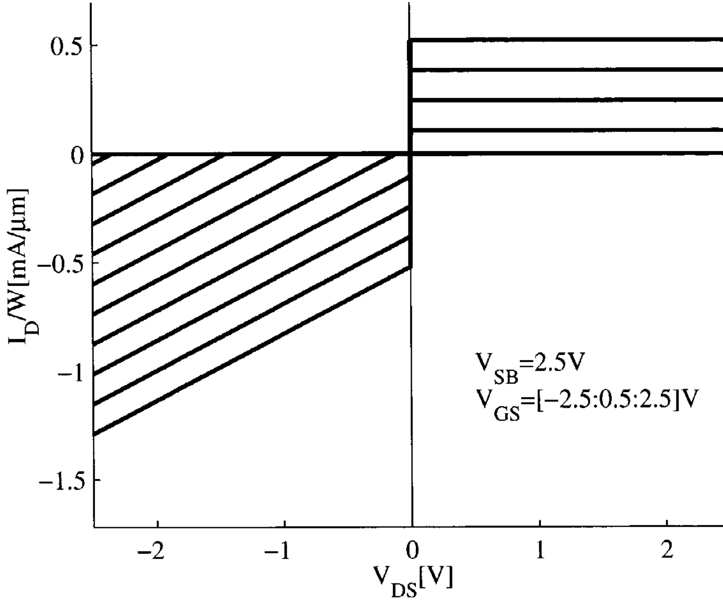


Figure 3.9: I_D vs. V_{DS} characteristic family for $V_0 = 0$

1. a *current source regime* where $V_{DS} > 0$ and $I_{DS} = \max\{k(V_{GS} - V_T), 0\}$
2. a *collapsed source regime* when the current imposed on the device is less than $I_{DS} = \max\{k(V_{GS} - V_T), 0\}$; the drain-source voltage is zero then,

where D is the channel terminal with the higher voltage.

Such a device is called a `collapsible current source` and simulators of deep submicron circuits better support such a device.

Chapter 4

Collapsible current sources

Contents

4.1 **Definition** 44

4.2 **Algorithmic requirements** 49

4.3 **Transistors as collapsible current sources** 52

4.4 **Modified collapsible current sources** 52

4.5 **Choosing a simulator** 56

In Chapter 3 we derived, based on the analysis of the device operation with velocity saturation as dominant effect, a *compact* transistor model. It is known as the collapsible current source model, because in the triode region the drain and the source terminals are considered to be fused together, i.e., the equivalent current source is collapsed.

The collapsible current source model is the *minimal* model necessary to explain the transistor level operation of the CMOS circuits. Being minimal and compact, it is exactly what we need for the *efficient* simulation of *large size* circuits.[1]

This model is a large signal piecewise linear model for the transport current of a deep submicron MOS transistor. In the following we derive its implementation in the piecewise linear equations framework from Section 2.5.

[1] With a huge number of devices, one needs compacts models in order to keep the circuit analysis tractable.

4.1 Definition

The *collapsible current source* is a device with three terminals G, D, and, S and characterized by its transconductance k, and its threshold V_T. The operating regions are characterized in a compact form as

NC	$I_D = 0$	if $V_{GS} < V_T$ and $V_{GD} < V_T$;
FS	$I_D = k(V_{GS} - V_T)$	if $V_{DS} > 0$ and $V_{GS} \geq V_T$;
RS	$I_D = -k(V_{GD} - V_T)$	if $V_{DS} < 0$ and $V_{GD} \geq V_T$;
T	$V_{DS} = 0$	if $V_{GS} \geq V_T$ and $-k(V_{GS} - V_T) \leq I_D \leq k(V_{GS} - V_T)$.

V_{DS} represents the drain-to-source voltage, V_{GS} represents the gate-to-source voltage, and I_D represents the current in the drain terminal (identical to the transport current I_T). Figure 4.1a shows how the input vector **x** and the output vector **y** are assigned. **x** has two components $x_1 = V_{DS}$ and $x_2 = V_{GS}$. **y** is one-dimensional and represents the current in the drain terminal I_D.

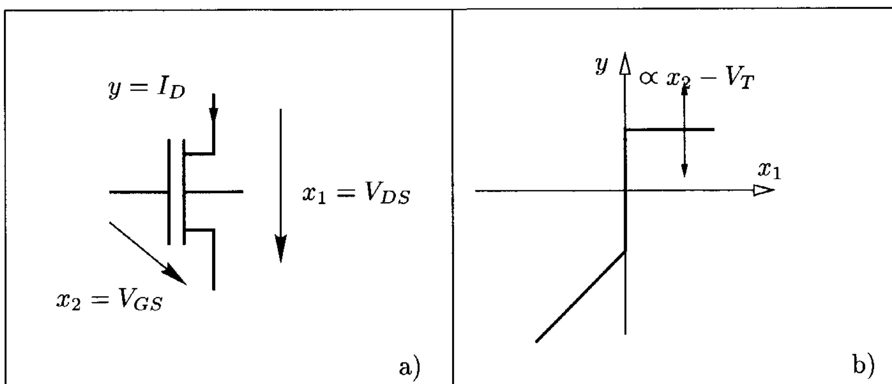


Figure 4.1: a) Symbol and notations for the collapsible current source b) Plot of y vs. x_1 with x_2 as parameter, according to Eq. (4.1) for the collapsible current source

The relation between y, x_1, and x_2 can be written as:

$$\begin{cases} \text{y}= \begin{cases} 0 & \text{if } x_2 - V_T \leq 0 \text{ and } -x_1 + x_2 - V_T \leq 0, \\ k(x_2 - V_T) & \text{if } x_2 - V_T \geq 0 \text{ and } x_1 > 0, \\ -k(-x_1 + x_2 - V_T) & \text{if } -x_1 + x_2 - V_T \geq 0 \text{ and } x_1 < 0, \end{cases} \\ x_1 = 0 \quad \text{if } -k(x_2 - V_T) < y < k(x_2 - V_T) \text{ and } x_2 - V_T > 0. \end{cases} \tag{4.1}$$

The input space $\mathbf{X}$ is divided into three polytopes as shown in Figure 4.2. The

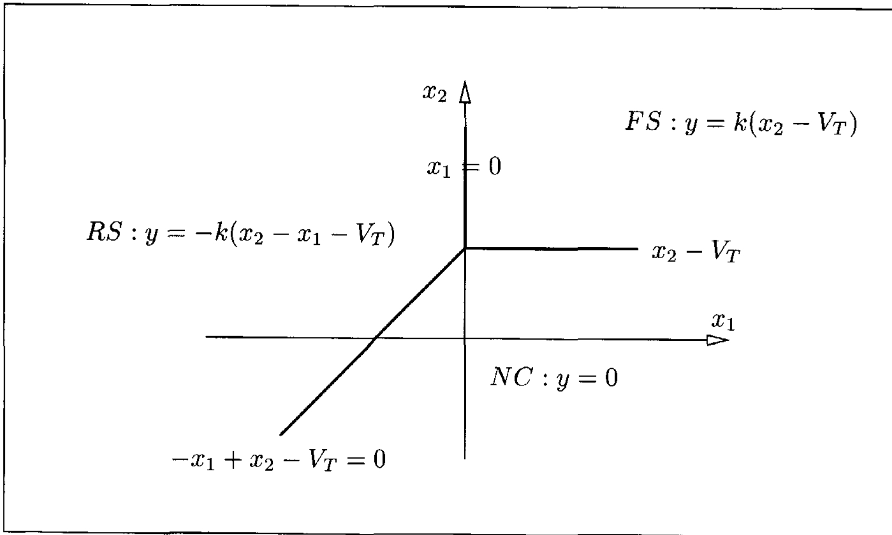


Figure 4.2: Partition of the input space (control voltage plane) in polytopes (operating regions)

boundaries of these polytopes are composed of halflines. In Figure 4.3 we show how the $\mathbf{x}$ vs. $\mathbf{y}$ space is partitioned into polytopes.

The hyperplane H_1 is defined by the concurrent lines l_1 and l_2 given by:

$$l_1 \begin{cases} y = 0 \\ -x_1 + x_2 - V_T = 0 \end{cases} \quad \text{and} \quad l_2 \begin{cases} x_1 = 0 \\ y - k(x_2 - V_T) = 0 \end{cases}$$

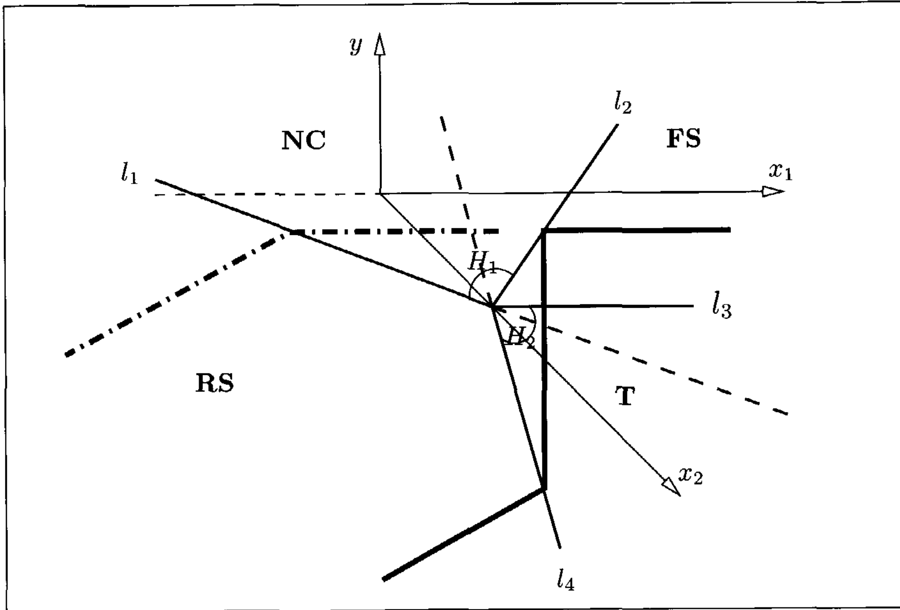


Figure 4.3: Partition of the input-output space in polytopes

and has the equation:

$$H_1 : \quad y + kx_1 - kx_2 + kV_T = 0 \tag{4.2}$$

Similarly, H_2 is defined by the lines l_3 and l_4

$$l_3 \begin{cases} y = 0 \\ x_2 - V_T = 0 \end{cases} \quad \text{and} \quad l_4 \begin{cases} x_1 = 0 \\ y + k(x_2 - V_T) = 0 \end{cases}$$

and has the equation:

$$H_2 : \quad -y - kx_2 + kV_T = 0 \tag{4.3}$$

The polytope indices are binary coded according to the rule: $\mathbf{P}_{\overline{\sigma(u_1)\sigma(u_2)}}$, where $\sigma(u) = 1$ for $u > 0$ and $\sigma(u) = 0$ for $u \leq 0$. The polytopes (operating regions) are

described by the relations:

$$\mathbf{P_0} \equiv \mathbf{NC} \qquad \begin{cases} y + kx_1 - kx_2 + kV_T > 0 \\ -y - kx_2 + kV_T > 0 \end{cases} \tag{4.4a}$$

$$\mathbf{P_1} \equiv \mathbf{FS} \qquad \begin{cases} y + kx_1 - kx_2 + kV_T > 0 \\ -y - kx_2 + kV_T < 0 \end{cases} \tag{4.4b}$$

$$\mathbf{P_2} \equiv \mathbf{RS} \qquad \begin{cases} y + kx_1 - kx_2 + kV_T < 0 \\ -y - kx_2 + kV_T > 0 \end{cases} \tag{4.4c}$$

$$\mathbf{P_3} \equiv \mathbf{T} \qquad \begin{cases} y + kx_1 - kx_2 + kV_T < 0 \\ -y - kx_2 + kV_T < 0 \end{cases} \tag{4.4d}$$

We choose the diode state variables such that $\mathbf{u} = \mathbf{0}$ in $\mathbf{P_0}$. In the form of the equation set (2.15) the collapsible current source is described by:

$$\begin{cases} 0 = \begin{pmatrix} 1 \end{pmatrix} y + \begin{pmatrix} 0 & 0 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} + \begin{pmatrix} b_1 & b_2 \end{pmatrix} \begin{pmatrix} u_1 \\ u_2 \end{pmatrix} + \begin{pmatrix} 0 \end{pmatrix} \\ j_1 = \begin{pmatrix} 1 \end{pmatrix} y + \begin{pmatrix} k & -k \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} + \begin{pmatrix} 1 & 0 \end{pmatrix} \begin{pmatrix} u_1 \\ u_2 \end{pmatrix} + \begin{pmatrix} kV_T \end{pmatrix} \\ j_2 = \begin{pmatrix} -1 \end{pmatrix} y + \begin{pmatrix} 0 & -k \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} + \begin{pmatrix} 0 & 1 \end{pmatrix} \begin{pmatrix} u_1 \\ u_2 \end{pmatrix} + \begin{pmatrix} kV_T \end{pmatrix} \end{cases} \tag{4.5}$$

In the polytope $\mathbf{P_2}$ the diode state variables take the values $j_1 = 0$, $u_2 = 0$; the PL segment $y = -k(-x_1 + x_2 - V_T)$ has, according to (4.5), the equivalent representation:

$$\begin{cases} 0 = y + b_1 u_1 \\ u_1 = -y - kx_1 + kx_2 - kV_T \end{cases} \tag{4.6}$$

It follows that $b_1 = 1/2$.

In the polytope $\mathbf{P}_1$ the diode state variables take the values $u_1 = 0$, $j_2 = 0$;

the PL segment $y = k(x_2 - V_T)$ is then equivalent to:

$$\begin{cases} 0 = y + b_2 u_2 \\ u_2 = y + kx_2 - kV_T \end{cases} \tag{4.7}$$

It follows that $b_2 = -1/2$.

The representation of the collapsible current source is therefore:

$$\begin{cases} 0 = \begin{pmatrix} 1 \end{pmatrix} y + \begin{pmatrix} 0 & 0 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} + \begin{pmatrix} \frac{1}{2} & -\frac{1}{2} \end{pmatrix} \begin{pmatrix} u_1 \\ u_2 \end{pmatrix} + \begin{pmatrix} 0 \end{pmatrix} \\ \begin{pmatrix} j_1 \\ j_2 \end{pmatrix} = \begin{pmatrix} 1 \\ -1 \end{pmatrix} y + \begin{pmatrix} k & -k \\ 0 & -k \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} + \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} u_1 \\ u_2 \end{pmatrix} + \begin{pmatrix} kV_T \\ kV_T \end{pmatrix} \\ \mathbf{u}^{\mathbf{T}} \cdot \mathbf{j} = \mathbf{0}, \mathbf{u} \geq \mathbf{0}, \mathbf{j} \geq \mathbf{0} \end{cases} \tag{4.8}$$

In Table 4.1 the correspondence is given between the current polytope, the name of the region according to the *collapsible current source*, the pivoting events, and the device equation. The polytope $\mathbf{P_0}$ is considered the natural representation.

Polytope/Region	Pivoting event	Device equation
$\mathbf{P_0}$/**N**on**C**onducting	none	$y = 0$
$\mathbf{P_1}$/**F**orward**S**aturation	$u_2 = pl.2$	$y = k(x_2 - V_T)$
$\mathbf{P_2}$/**R**everse**S**aturation	$u_1 = pl.1$	$y = -k(x_2 - x_1 - V_T)$
$\mathbf{P_3}$/**T**riode	$u_1 = pl.1$ and $u_2 = pl.2$	$x_1 = 0$

Table 4.1: Operating regions of the collapsible current source

The set (4.8) is further translated in Table 4.2 in terms of nodal voltages (V_D, V_G, V_S) and terminal currents (I_D, I_G, I_S).

	D.i	G.i	S.i	D.v	G.v	S.v	pl.1	pl.2	1
zero.1		1							
zero.2	1		1						
zero.3	1						1/2	-1/2	
pl.1	1			k	$-k$		1		kV_T
pl.2	-1				$-k$	k		1	kV_T

Table 4.2: Tabular representation of a collapsible current source

4.2 Algorithmic requirements

With the diode state model of Section 2.5 in mind it is pretty straightforward to devise algorithms that switch diodes one at the time very much like you expect such a network to behave. In a description according to the equations (2.13) this corresponds with pivoting on a diagonal element in **D**. We will illustrate this with the circuit of Figure 4.4a. The input vector $\mathbf{x} = (x_1 = v_{DS}, x_2 = v_{GS})$ is varied

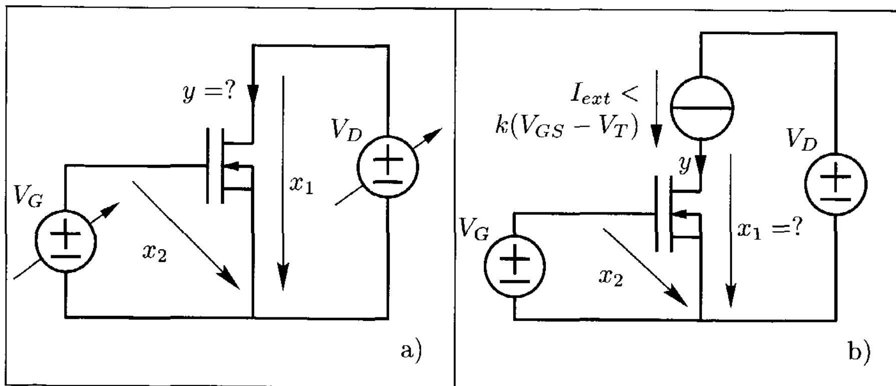


Figure 4.4: Test circuits for pivoting a) **NC** $\rightarrow$ **FS** $\rightarrow$ **RS** $\rightarrow$ **NC** b) **NC** $\rightarrow$ **T**

according to the path: $(V_+, 0) \rightarrow (V_+, V_+) \rightarrow (-V_+, V_+) \rightarrow (-V_+, -V_+)$ with $V_+ > V_T$. This means that the collapsible current source will be non-conducting initially, i.e., the collapsible current source is in **NC**. The equations written as in

the set (2.13) read:

$$
\begin{array}{rccccc}
y = & & & -\frac{1}{2}u_1 & +\frac{1}{2}u_2 & \\
j_1 = & kx_1 & -kx_2 & +\frac{1}{2}u_1 & +\frac{1}{2}u_2 & +kV_T \\
j_2 = & & -kx_2 & +\frac{1}{2}u_1 & +\mathbf{\frac{1}{2}u_2} & +kV_T
\end{array}
$$

where $\mathbf{u} = \mathbf{0}$. This remains so when increasing x_2 until it reaches the threshold voltage V_T. At that moment j_2 becomes zero. The operating region of the collapsible current source becomes then **FS** with corresponding natural representation:

$$
\begin{array}{rccccc}
y = & & kx_2 & -u_1 & +j_2 & -kV_T \\
j_1 = & kx_1 & & & +\mathbf{j_2} & \\
u_2 = & & 2kx_2 & -\mathbf{u_1} & +2j_2 & -kV_T
\end{array}
$$

In terms of diode states the diode associated with the complementary pair (u_2, j_2) "switched" and has now $j_2 = 0$, while initially $u_2 = 0$. Increasing x_2 further does not move the collapsible current source into a different operation region.

Unfortunately, not all changes in $\mathbf{x}$ can be handled by an algorithm that searches for single pivots on the diagonal, and thus allows only one diode to change state. We will see this when we decrease x_1 from its current value of V_+. When $x_1 = 0$, j_1 becomes zero, but the diagonal element in the corresponding row is zero as well. So we have to take a non-diagonal pivot, interchange j_1 with j_2 so that the latter can take positive values to compensate for the decreasing x_1. But with j_2 increasing, u_2 has to be 0, because of the complementarity constraint. So, also the pivoting exchanging u_2 with u_1 has to be performed. The coefficient of u_1 in the bottom row is non-zero. The collapsible current source changes abruptly from region **FS** to **RS** which is to be expected when the drain-source voltage is forced to change sign. This newly entered region corresponds with the natural representation:

$$
\begin{array}{rccccc}
y = & kx_1 & -kx_2 & -j_1 & +u_2 & +kV_T \\
j_2 = & -kx_1 & & & +j_1 & \\
u_1 = & -2kx_1 & +2kx_2 & -u_2 & +\mathbf{2j_1} & -2kV_T
\end{array}
$$

The transition from **FS** to **RS** is thus only possible if we perform two pivoting operations in the same step. Decreasing x_1 further to $-V_+$ does not require any pivoting. If we now decrease x_2, while keeping x_1 fixed at $-V_+$ another pivoting operation is required when $-2kx_1 + 2kx_2 - 2kV_T$ becomes 0, which implies that x_2

has reached $x_1 + V_T$. To decrease x_2 any further u_1 has to be exchanged with j_1, variables associated with the same diode. The collapsible current source is in **NC** again.

Another difficulty that straightforward piecewise linear simulators might encounter is when the current through the drain terminal of a collapsible current source is forced to be below the value of the saturation current corresponding with the present value of the gate-source voltage, i.e., $i_D < k(V_{GS} - V_T)$. This can occur in the circuit of Figure 4.4b. The variable y can hardly be the output variable, since its value is forced by the fixed current source. So, for the moment we do not distinguish input and output, and we start from the **NC**-region:

$$\begin{array}{rcccccc} 0 = & I_{ext} & & & +\frac{1}{2}u_1 & -\frac{1}{2}u_2 & \\ j_1 = & I_{ext} & +kx_1 & -kx_2 & +u_1 & & +kV_T \\ j_2 = & -I_{ext} & & -kx_2 & & +\mathbf{u_2} & +kV_T \end{array}$$

If u_2 were zero, than its complementary variable j_2 cannot be positive, because $-I_{ext} - kx_2 + kV_T < 0$. Interchanging j_2 and u_2 assumes the collapsible current source in the **FS**-region:

$$\begin{array}{rcccccc} 0 = & I_{ext} & & -kx_2 & +u_1 & -j_2 & +kV_T \\ j_1 = & I_{ext} & +kx_1 & -kx_2 & +\mathbf{u_1} & & +kV_T \\ u_2 = & I_{ext} & & +kx_2 & & +j_2 & -kV_T \end{array}$$

This cannot be true with u_1 and j_2 both zero, even if x_1 takes a positive value to balance the second equation. So, we also interchange j_1 and u_1, to obtain the set:

$$\begin{array}{rcccccc} 0 = & & -kx_1 & & +j_1 & -j_2 & \\ u_1 = & -I_{ext} & -kx_1 & +kx_2 & +j_1 & & -kV_T \\ u_2 = & I_{ext} & & +kx_2 & & +j_2 & -kV_T \end{array}$$

Now x_1 can be zero, while u_1 and u_2 take positive values. This is the triode-region, where V_{DS} is zero and the drain current can have any value up to $k(V_{GS} - V_T)$, and thus also I_{ext}. Two diagonal pivots yield the natural representation of the feasible region **T**.

4.3 Transistors as collapsible current sources

As pointed out in Chapter 3, a transistor becomes more and more a collapsible current source when the channel length approaches zero. Well before this limit other assumptions underlying the modeling will no longer be sustainable. The very concept of *mobility* and the mechanism of *drift* are just two examples. Yet, the graphs in Figure 3.7 and 3.8 show already behaviors close to that of a collapsible current source. Note however that these graphs are obtained by reducing the channel length while keeping the drain-to-source voltage quite high. This makes the field strength quite high, hence the high speed of the carriers. At more realistic voltage differences the characteristic would exhibit the limit behavior at much smaller channel lengths, lengths that are not manufacturable yet. At finite length saturated carrier velocities will not be reached before V_{DS} reaches $L \times E_{cr}$, and at $L = 0.1\mu m$ this might still come close to $0.1V$ in *p*-type transistors.

None of this is of much concern as long as interconnect dominates delay. As technology is scaled down this will only come closer to reality, but simulating where interconnect is not taken into account or indeed negligible, using the pure collapsible current source model for transistors will result in noticeable deviations in timing when compared to simulators using detailed models such as in [18].

Apart from being the mathematical limit of a model derived from physics, the collapsible current source model had the salient feature of compactness and linearity. The former is an absolute must for simulating systems-on-a-chip, and the latter gave us access to a one-algorithm approach and therefore level transparency. We therefore would like to retain these features, when modifying the model for time accuracy in today's circuits.

The modification consists of considering the triode region represented on the I-V characteristic by an oblique linear segment instead of a vertical one.

As we will see, this modification brings no additional computational complexity with respect to the pure model.

4.4 Modified collapsible current sources

This modified model has the operating regions characterized by:

- $I_D = 0$ if both V_{GS} and V_{GD} are less than V_T;

- I_D is a linear function of $V_{GS} - V_T$ or $V_{GD} - V_T$ whichever of them is greater than the other and positive, and the transistor is in the saturation region;
- I_D is a linear function of V_{DS} if $V_{GS} - V_T$ or $V_{GD} - V_T$ is positive, and the transistor is in the triode region;

The analytical formulation is given in (4.9) and corresponds to the I-V characteristic from Figure 4.5. One extra model parameter has been introduced, k_1, which has the significance of the large signal output conductance in the triode region. The parameter k_2 has the same meaning as k in Section 4.1.

$$y=\begin{cases} 0 & \text{if } -x_1 + x_2 - V_T < 0 \text{ and } x_2 - V_T < 0 \\ k_2(x_2 - V_T) & \text{if } x_2 - V_T > 0 \text{ and } -k_1x_1 + k_2(x_2 - V_T) < 0 \\ k_1x_1 & \text{if } -k_1x_1 + k_2(x_2 - V_T) > 0 \text{ and} \\ & \quad -k_1x_1 - k_2(-x_1 + x_2 - V_T) < 0 \\ -k_2(-x_1 + x_2 - V_T) & \text{if } -k_1x_1 - k_2(-x_1 + x_2 - V_T) > 0 \text{ and} \\ & \quad -x_1 + x_2 - V_T > 0 \end{cases} \tag{4.9}$$

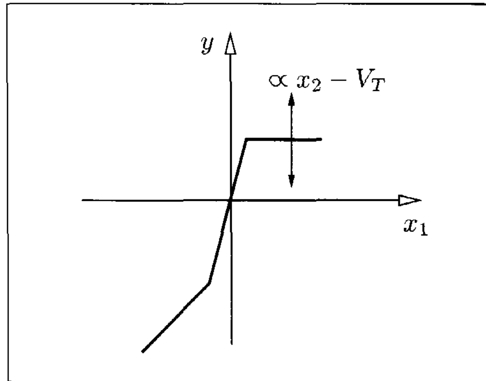


Figure 4.5: Plot of y vs. x_1 with x_2 as parameter, according to Eq. (4.9)

The input space $\mathbf{X}$ is divided into four polytopes as shown in Figure 4.6. From the same considerations as before the representation in the $\mathbf{x} - \mathbf{y}$ space is useful to

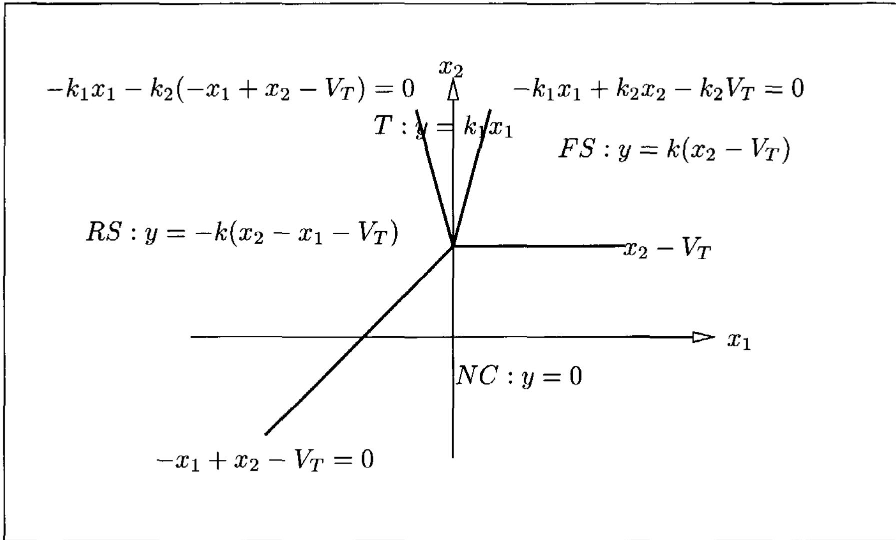


Figure 4.6: Partition of the input space (voltage plane) in polytopes (operating regions)

derive the diode state equations. The lines that define the polytope boundaries are presented in Figure 4.7.

H_1 is given by the concurrent lines

$$l_1 \begin{cases} y = 0 \\ -x_1 + x_2 - V_T = 0 \end{cases} \quad \text{and} \quad l_2 \begin{cases} k_1x_1 = k_2(x_2 - V_T) \\ y = k_2(x_2 - V_T) \end{cases} .$$

The equation of H_1 is:

$$\frac{k_1 - k_2}{k_1k_2}y + x_1 - x_2 + V_T = 0 \tag{4.10}$$

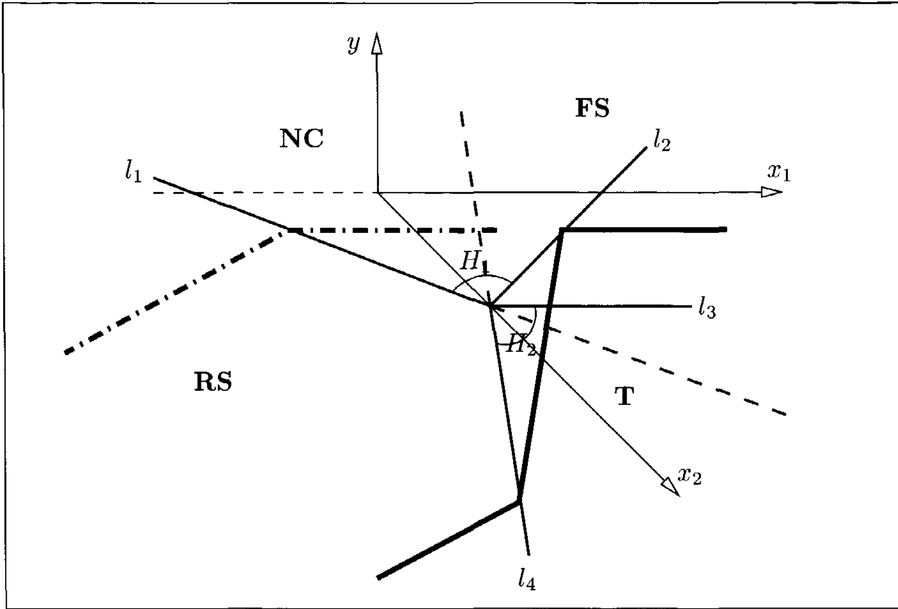


Figure 4.7: Partition of the input-output space in polytopes

H_2 is given by the concurrent lines

$$l_3 \begin{cases} y = 0 \\ x_2 - V_T = 0 \end{cases} \quad \text{and} \quad l_4 \begin{cases} y = k_1 x_1 \\ (k_1 - k_2)x_1 + k_2 x_2 - k_2 V_T = 0 \end{cases} .$$

The equation of H_2 is:

$$-\frac{k_1 - k_2}{k_1 k_2} y - x_2 + V_T = 0 \tag{4.11}$$

The two hyperplanes H_1 and H_2 divide the input-output space into four polytopes. We use the same convention for the polytope indices of Section 4.1. The

polytope equations are:

$$\mathbf{P_0} \equiv \mathbf{NC} \qquad \begin{cases} \frac{k_1-k_2}{k_1k_2}y + x_1 - x_2 + V_T > 0 \\ -\frac{k_1-k_2}{k_1k_2}y - x_2 + V_T > 0 \end{cases} \tag{4.12a}$$

$$\mathbf{P_1} \equiv \mathbf{FS} \qquad \begin{cases} \frac{k_1-k_2}{k_1k_2}y + x_1 - x_2 + V_T > 0 \\ -\frac{k_1-k_2}{k_1k_2}y - x_2 + V_T < 0 \end{cases} \tag{4.12b}$$

$$\mathbf{P_2} \equiv \mathbf{RS} \qquad \begin{cases} \frac{k_1-k_2}{k_1k_2}y + x_1 - x_2 + V_T < 0 \\ -\frac{k_1-k_2}{k_1k_2}y - x_2 + V_T > 0 \end{cases} \tag{4.12c}$$

$$\mathbf{P_3} \equiv \mathbf{T} \qquad \begin{cases} \frac{k_1-k_2}{k_1k_2}y + x_1 - x_2 + V_T < 0 \\ -\frac{k_1-k_2}{k_1k_2}y - x_2 + V_T < 0 \end{cases} \tag{4.12d}$$

We write the natural implicit representation for the polytope $\mathbf{P}_0$, where $y = 0$.

The PL representation of the modified collapsible current source is:

$$\begin{cases} 0 = \begin{pmatrix} 1 \end{pmatrix} y + \begin{pmatrix} 0 & 0 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} + \begin{pmatrix} \frac{k_1k_2}{2k_1-k_2} & -\frac{k_1k_2}{2k_1-k_2} \end{pmatrix} \begin{pmatrix} u_1 \\ u_2 \end{pmatrix} + \begin{pmatrix} 0 \end{pmatrix} \\ j_1 = \begin{pmatrix} \frac{k_1-k_2}{k_1k_2} \end{pmatrix} y + \begin{pmatrix} 1 & -1 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} + \begin{pmatrix} 1 & 0 \end{pmatrix} \begin{pmatrix} u_1 \\ u_2 \end{pmatrix} + \begin{pmatrix} V_T \end{pmatrix} \\ j_2 = \begin{pmatrix} -\frac{k_1-k_2}{k_1k_2} \end{pmatrix} y + \begin{pmatrix} 0 & -1 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} + \begin{pmatrix} 0 & 1 \end{pmatrix} \begin{pmatrix} u_1 \\ u_2 \end{pmatrix} + \begin{pmatrix} V_T \end{pmatrix} \\ \mathbf{u^T \cdot j = 0, u \geq 0, j \geq 0} \end{cases} \tag{4.13}$$

We observe that the representation (4.13) of the modified model has the same number of diode state variables as the representation (4.8) of the pure model.

4.5 Choosing a simulator

Circuits that contain *collapsible current sources* pose special demands on simulators. Although they fit perfectly in the piecewise linear approach to modeling, being the most compact formulation of the triode operational principle, the matrices they produce in the diode state model (2.13) can disqualify many renowned algorith-

mic circuit analysers. *Diagonal pivoting* and *principal pivoting*, in spite of several modifications, were found to be inadequate, whereas *complementary pivoting* as conceived by Lemke, and variants thereof processed all cases we confronted them with (see Appendix A). The conclusion is the ability to use the pure collapsible current source in circuits to be simulated hinges on the availability of such a solver for the linear complementarity problem.

The *modified collapsible current source* retains much of the salient features of the pure collapsible current source: it is as compact and equally suited for an approach based on piecewise linear modeling. Its main diminution is that it is not backed by a rigourous mathematical or physical derivation, and the additional parameters have to be established by careful calibration. This, of course, is not of any practical significance, since it is always wise to determine model parameters for every new technology by extensive measurements and simulations. Besides, the above objections against simpler path followers to solve the linear complementarity problem may well have vanished largely.

In circuitry where transistors are connected with negligeable interconnect resistance it even turned out to be necessary to use well-calibrated modified collapsible current sources to get the timing results close to what well-proven simulators with full-blown transistor models predict.

Chapter 5

Introducing time

Contents

5.1 **Time discretization** 61

5.2 **Multirate integration** 63

5.3 **Event-driven simulation** 64

5.4 **Multi-level simulation capabilities** 66

The discussion up to now has been limited to resistive circuits, meaning that every change in stimuli will be translated instantaneously into changes in the node voltages and branch currents, and their values only depend on the momentaneous values of the stimuli. Previous values may be helpful for fast convergence, but the new values are independent of what happened before. This is not realistic, because there will always be storage of energy in practical circuits. A change in excitation can cause a redistribution of the stored and added energy. The new *state* of the circuit therefore does not only depend on the new values of the stimuli, but also on the distribution of stored energy. In circuit models with lumped elements this energy is assumed to be stored in capacitors and inductors. A capacitor can hold electric charge. The branch current satisfies

$$i = \frac{dq}{dt}$$

when the branch holds a capacitor. The relation with the branch voltage for a

voltage-controlled capacitor is through the *incremental capacitance*:

$$C(v) = \frac{dq(v)}{dv}.$$

The state of an inductor is determined by the flux $\phi(t)$ and the voltage over an inductor obeys

$$v = \frac{d\phi}{dt},$$

while we use the incremental inductance to obtain the branch current:

$$L(i) = \frac{d\phi(i)}{di}.$$

This is how the *time* variable enters the description of lumped electric circuits. A change in the input stimuli will generally require an update in the response of the system with a certain time delay. This inertia of the system is due to the above memory-like elements. Lumped circuits can be described by a combination of non-linear algebraic and ordinary differential equations. The latter can be given in the form

$$\dot{\mathbf{x}} = \mathbf{F}(\mathbf{x}) \tag{5.1}$$

where $\mathbf{x}$ stands for an unknown voltage or current. To solve a set of ordinary differential equations we need to know the initial conditions. For the above set of first-order equations the values of all components of $\mathbf{x}$ at $t = 0$ must be given.

Numerical methods have to be applied to solve such a system (approximately) by computer. From the available methods, circuit analysis programs apply almost invariably the approximation of the derivative through integration rules.

5.1 Time discretization

An obvious approximation for the derivative is

$$\frac{dx}{dt} \approx \frac{x(t_{n+1}) - x(t_n)}{t_{n+1} - t_n}.$$

The quantity $t_{n+1} - t_n$ is called the *time-step* and often shorted to h. Substitution in the equation set (5.1) leads to

$$\mathbf{F}\left(\mathbf{x}(t_{n+1})\right) - \frac{1}{h}\mathbf{x}(t_{n+1}) + \frac{1}{h}\mathbf{x}(t_n) = 0 \tag{5.2}$$

which is a non-linear algebraic equation in the components of $\mathbf{x}_{n+1}$, because $\mathbf{x}(t_n)$ are constants. Focussing on a single branch with a dynamic element, for example a capacitor, shows the meaning of this reduction more clearly:

$$\frac{dv}{dt} = \frac{1}{C}i \rightarrow \frac{v(t_{n+1}) - v(t_n)}{h} = \frac{1}{C}i(t_{n+1}) \rightarrow i(t_{n+1}) = \frac{C}{h}v(t_{n+1}) - \frac{C}{h}v(t_n).$$

Since $v(t_n)$ is known at time t_{n+1} the dynamic branch is transformed by time discretization into a current source of intensity $-\frac{C}{h}v(t_n)$ and an internal conductance of $\frac{C}{h}$.

That is the conceptually simple end sound idea behind time discretization: reducing the simulation of a dynamical circuit to a sequence of simulations of resistive circuits. The formula (5.2) is just an example of how such a reduction can be carried out. There are numerous other methods, each with its own advantages and disadvantages. Choosing should be done on the basis of accuracy, efficiency and stability. The former two are largely a trade-off: more accurate methods often require more computational effort. Stability imposes more subtle constraints, often limiting the possibilities for the other two.

Accuracy is to measure the difference between the calculated solution and the exact one. With the latter generally unknown this criterion becomes unfeasible. One therefore resorts to local truncation errors: the difference between the calculated solution and the exact next value if the values up to the previous time-point were exact. Of course, this difference depends on the method, the circuit and the stimulus. We say that a method is of the *order* q if the local truncation error is zero when the solution is a polynomial of the order q.

High order can be achieved by using the values and derivatives at several

time-points, current and previous ones, i.e., by using *multistep methods*, among which linear multistep integration methods are the most popular (formula (5.2) is an example). In general they read like

$$\sum_{j=0}^{p} (a_j x(t_{n-j+1}) + h \cdot b_j \cdot \dot{x}(t_{n-j+1})) = 0.$$

Circuit simulators that use direct methods favour implicit methods ($b_0 \neq 0$), because, beside transforming the circuit at a time-point into a non-linear circuit, they have other desirable properties like charge conserving and exhaustive domain of dependence[1].

Putting it together for linear multistep formulae: the local truncation error is in general

$$LTE_{n+1} \approx \frac{C \cdot h^{k+1}}{(k+1)!} x_n^{(k+1)} \tag{5.3}$$

where C depends on the method, k is the order of the formula and $x^{(k+1)}$ denotes the $(k+1)$ derivative. The order constrains however for a fixed p the choice of coefficients. The higher the required order, the more constraints leaving less freedom for other quality aspects such as stability. Once the constraints become conflicting, only increasing the number of steps, and thus the complexity of the formula, can achieve the required order.

So in order to achieve accuracy, we have to choose small time steps and consequently solve the whole set of algebraic equations many times, or use complex formulae which use several values from the past and create more complex resistive networks. On top of that stability demands attention.

Stability roughly implies that differential equation with bounded solutions should yield bounded values at the different time points when subjected. Analysing a simple equation, such as the test equation

$$\dot{x} = -\lambda x; \quad x(0) = 1; \quad \lambda \in \mathbb{C} \quad Re(\lambda) \geq 0$$

learns that this depends on the modulus of $h \cdot \lambda$. If for every size of h the series of values produced by a given integration method stays bounded, then the method

[1]For small enough h the formula affects the same set of variables as the exact solution.

is called A-stable. Unfortunately, there are no linear multistep formulae of order higher than 2 that are A-stable so that truncation errors of in the order of h^3 cannot be avoided. Weakening the requirement by bounding $|arg(\lambda)|$, i.e., $|arg(\lambda)| < \alpha$ allows formulas that might have order 6. These methods are called A(α)-stable or stiffly stable. All other methods have at best an additional upper bound on the step size.

Stiff stability is important in circuit simulators because circuits lead in practice to stiff sets of differential equations, if not by design, then because of relatively small parasitics. The size of uniform time step should be small enough in order to correctly capture the evolution of the system variables which have the fastest time variation. This means that all system variables, including those latent or with slow variation in time, will be updated at the same high rate given by the size of the uniform time step. These "redundant" updates make the uniform time step integration method computationally inefficient.

Simulators therefore apply dynamic time-step control. It is aimed at keeping the accuracy within a given tolerance while minimizing the number of time points, that is the number of resistive circuits to build and to evaluate. This is usually achieved by monitoring the estimate for the local truncation error (5.3). Under some assumptions about how errors accumulate, a maximal step size can be derived. Stability may require a smaller stepsize though. Increasing the step-size after fast "parasitic" phenomenae have died out, may wake them up because $h \cdot \lambda$ with λ indicating the rate of change of these phenomenae, is moved outside the stability region. Another reason why stiffly stable integration methods are required in a simulator based on direct methods.

5.2 Multirate integration

Application of multistep methods, with or without accuracy-driven time-step control, discretize all system equations identically, and the time-step must be small enough so that the fastest changing variable can be represented with the required accuracy. The efficiency would often be greatly improved, if it were possible to have different discretizations for several subsets of the system equations, using the largest time-step that can accurately handle the circuit variables associated with each subset. Large-scale circuits often exhibit a considerable amount of *latency*, i.e., many circuit variable are inactive for most of the time-points. Numerical integra-

tion methods that allow for different time-steps for the variables in the equations are called *multirate integration methods*[2].

The more successful multirate method in circuit simulation is to decompose the circuit and solve the subsystems independently. The integration method can use the preferred time-step for each part, thereby achieving full multirate behavior. Transistors that are constructed to have the current into the gate independent from the voltages at the other terminals are particularly helpful in decomposing the system.[3] It is then possible to break the equation set into blocks so that if the blocks are solved in the proper order, a good approximating solution for the entire system can be obtained. An acyclic network of gates, solved in topological order from primary inputs to primary outputs, each time using the earlier solutions, is a standard example for successful application of this approach. The observation that this kind of single sweep through the equation system is in essence one iteration of a Gauss-Seidel relaxation algorithm directly applied to the set of differential equations, led to a new approach to circuit simulation, called *waveform relaxation* [23]. The unknowns, however, are rather elements of a function space (*waveforms*) than real variables.

5.3 Event-driven simulation

The variants of time discretization discussed up to now have been developed for the time domain simulation of transistor-level circuits. Event-driven time-step control emerged for time domain simulation of gate-level circuits [24]. The concept is based on scheduling changes in signals for specified time points. These changes are called *events* and their time of occurrence is determined by the inner operation of the system components, i.e., the time step between two consecutive evaluations of a component depends on the local activity in the system. Parts with slow time variation will operate with a large time step, while parts with fast time variation will operate with a small time step. If there is no activity to propagate, then the time step can be virtually infinite.

[2]One way to achieve multirate behavior is to abandon implicit multistep integration methods that inherently require the global solution of the ensuing set, and use semi-implicit methods. These take an A-stable method and use a relaxation scheme to solve the generated algebraic equations.

[3]Ideal MOS transitors have this property. Ideal means the absence of the so-called *Miller* capacitances, usually caused by gate-drain overlap, so that no feedback from drain to gate occurs.

In an event-driven simulator[4] each component of the system is regarded as a standalone *process* which interfaces with the other components and/or with the environment through *signals*. What is specific to the signals from an event-driven perspective is that each change in value for such a signal is associated with an event at a certain simulation time. Another specific characteristic of such a signal is then by necessity its discrete nature: the fact that the values the signal takes are from a finite and discrete set of values.

Transactions are scheduled (new values are put on the driver of a signal) for future simulation times according to a specific delay mechanism. At the current simulation time the transactions which are still on the signal driver and which predict a different value for the signal than the current one will become events. Because of the discrete nature of the signals and the event scheduling mechanism, activity in the system propagates to sharply specified time instants. Besides, activity propagates only to parts of the system which are fed by active signals.

Communication inside a system is realized by passing signal events between components. The functionality of each component is described according to its constitutive equation just as in the approaches using direct methods, but now employing the event scheduling mechanism.

A generic event-driven simulation flow is given in Figure 5.1 [27]. The simulation time is advanced to the next time for which transactions are pending; this becomes the current simulation time. Next, the simulator retrieves from the event list the events scheduled to occur at the current time and updates the values of the active signals. The fanout list of the active signals is then traced to locate the activated elements. This process parallels the propagation of changes in the behavioral conception of the circuit. The evaluation of the activated elements may result in new transactions. These are scheduled to occur in the future according to the delays associated with the operation of the elements. The simulator inserts the newly generated transactions in the event list. The simulation continues as long as there is logic activity in the circuit; that is, until the event list becomes empty.

[4]Event-driven simulators use input description languages such as the VHDL language [25] or the Verilog language [26]. The discussion in this section makes use of the VHDL terminology. The general setup for the simulation of a non-linear electrical system from Section 2.1 is maintained here.

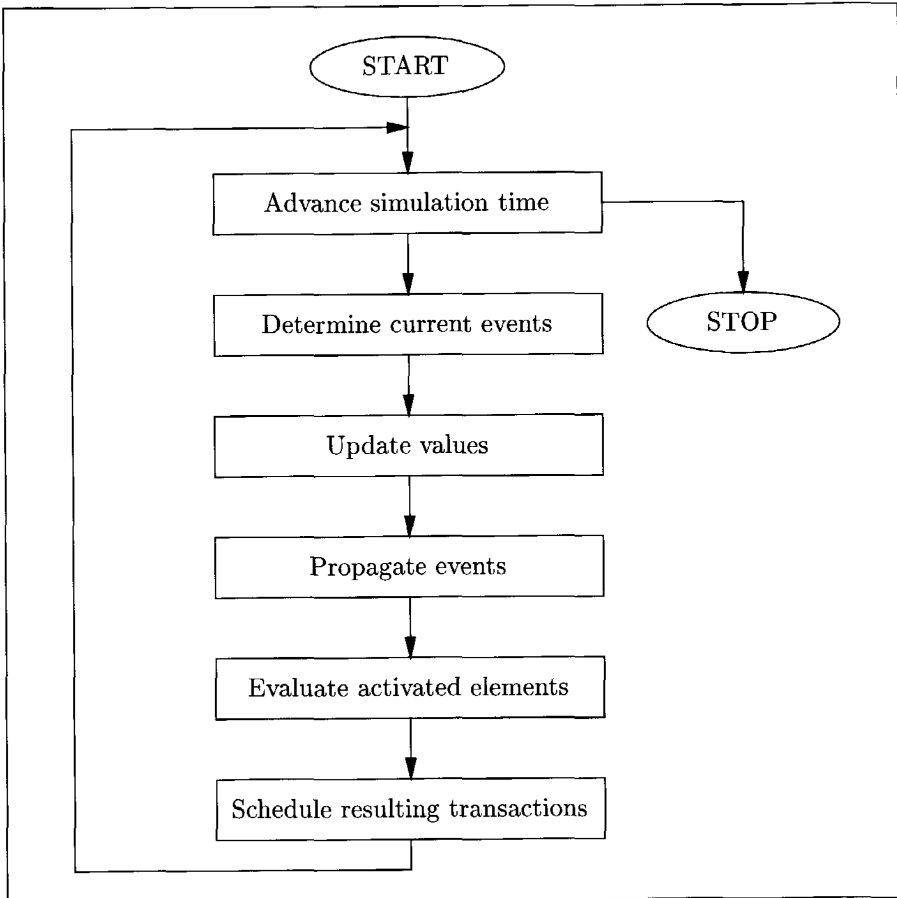


Figure 5.1: A generic event-driven simulation flow

5.4 Multi-level simulation capabilities

Assume that the network contains components described at various abstraction levels L_1, L_2, etc. Traditional simulators based on direct method have to gather functionally related components from one level, say L_1, into a larger component at a higher hierarchical level, say L_2. By applying in a recursive way the above technique, one finally gets the topological and constitutive equations of the global network.

By using this setup the steps needed to simulate the operation of a component positioned lowest in the hierarchy are:

1. build in a bottom-up manner the system description,
2. solve the global system equation, and
3. get the results for the specified (atomic) component.

These three steps are presented in Figure 5.2.

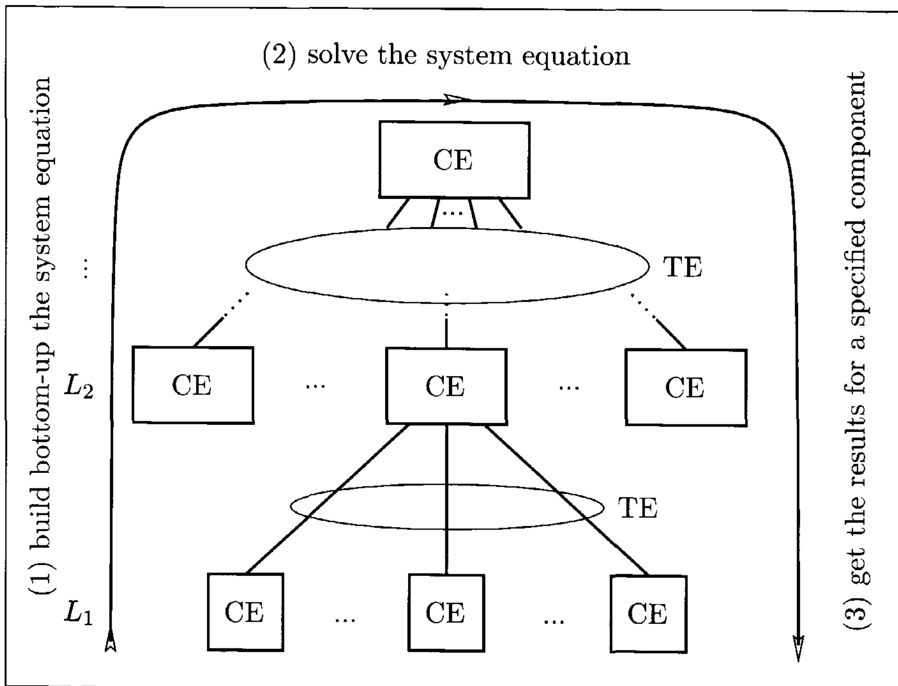


Figure 5.2: Simulation over levels of hierarchy from the perspective of flat direct method

Such an approach is based on flattening the hierarchy of the system under consideration in one equation set (Kirchhoff laws and constitutive equations). Some drawbacks of this all-in-one-equation approach are emphasized hereafter. Firstly, the manipulation of one single equation becomes prohibitively expensive for large

systems, in terms of memory consumption and especially in terms of run time. Secondly, for each update of the system status the *whole* system equation must be solved once again. The locality of the activity, which is to be encountered at large in digital systems, is not at all exploited because of the flattened hierarchy. Thirdly, and not least important, the equation-based approach, although it offers the utmost accurate solution, is a singular approach, in the sense that it cannot easily communicate with higher abstraction level simulation approaches (gate, behavioral, or algorithmic).

Figure 5.3 shows conceptually how components from different hierarchical

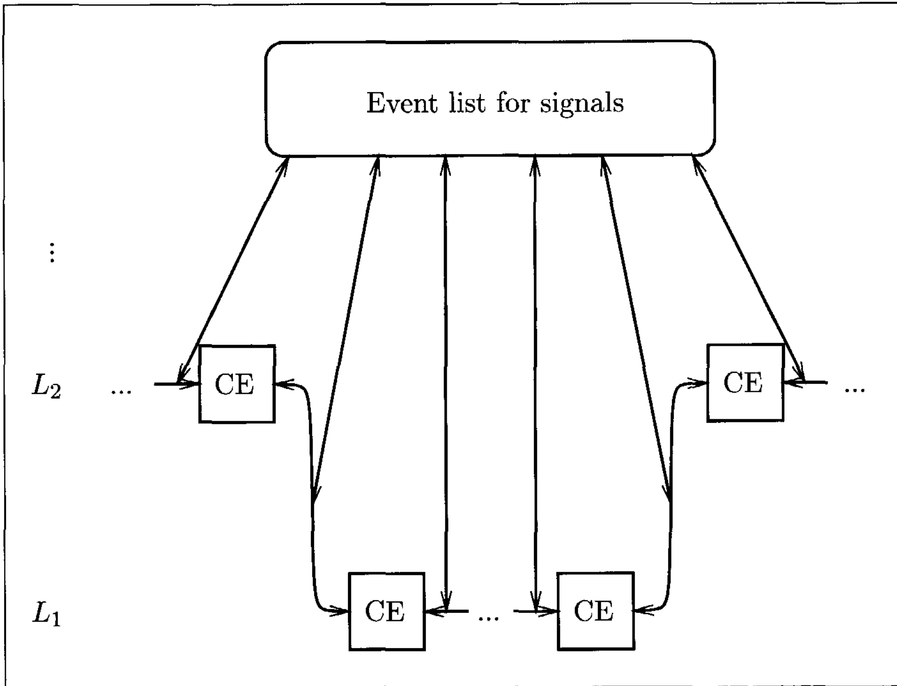


Figure 5.3: Simulation through many levels of hierarchy from the perspective of the event-driven approach

levels interact with each other in the event-driven paradigm. Signal events are exchanged between components situated at any level in the hierarchy via the universal

event list for signals. In this way not only the communication between components at the same level, but also the communication between components at any different levels in the hierarchy, becomes possible while keeping the representations of the components separated.

For monitoring the time evolution of a system, an event-driven simulator replaces the numerical integration methods of the direct methods with an event scheduling mechanism.

In the event-driven simulation approach, the constitutive equations of the components are kept by construction separated from the topological equations. The latter are distributed over the signals which interconnect the components.

In summary, the event-driven approach offers the following:

- Level transparency is enabled in a natural way. Components at various levels of abstraction (transistor, gate, and register transfer level) described in a single description language are accommodated within an event-driven simulator. This allows the usage of a single simulation algorithm.
- Simulation efficiency is considerably enhanced, because a single algorithm provides for enhanced simulation speed, and the locality of the signal activity is now exploited: instead of re-evaluating the overall system state at each update, only one part, the activity channel, is re-evaluated.
- Simulation of large and complex systems becomes possible. The problems related to the manipulation of large equation sets are eliminated.

Chapter 6

Transistor inertia

Contents

6.1 Quasi-static approximations 72

6.2 Modeling large-signal charge variations 76

6.3 Evaluating large-signal charge variations 81

6.4 Dynamic characterization of an inverter 84

6.5 Average capacitors 86

The theory of transistor operation is the theory of charge distribution and redistribution in the transistor structure. The power supply terminals determine fixed energy levels for the charge carriers, the logic circuitry modifies through the control signals the energy levels of the carriers in its nodes. If a conducting path exists between two regions with different energetic levels, then a net flow of charge carriers will occur from the region with higher energy towards the region with lower energy. Incorporating all consequences of semiconductor physics into the models for simulation will definitely kill the compactness and efficiency that we set as a goal for simulation of systems-on-a-chip. Compromising between model compactness and simulation accuracy is a delicate task that required a lot of effort in the past decades, an effort that has been amply recorded in literature and is adequately summarized in [16].

In spite of the incredible switching speeds feasible today, the lumped circuit approximation is still justified: the sizes are of course important, but assuming that

all electromagnetic phenomena are isolated from other parts of the system except through terminal connections. This means that the charge conservation law applies to the device, i.e., total charge neutrality.

6.1 Quasi-static approximations

For the static operation the constant terminal voltages determine the charge distribution inside the transistor structure. Flow of charge due to potential and/or concentration gradient along a conducting path, is of course possible, but the total charge inside a volume is constant in time, i.e., the charge amount that enters the volume is equal to the charge amount that leaves the volume. This static flow of charge is called transport current (I_T) and has been the object of study in Chapter 3. For the static case the terminal voltages completely determine the charge distribution and the transport current.

When the terminal voltages vary in time, transistors will exhibit inertia, because charges cannot be redistributed instantaneously. In theory, we have to know the exact charge distribution at a certain moment and the imposed terminal voltages from then on, to determine the momentary charge distribution. This is however a fairly complicated process, as detailed studies have shown in the past [28], too complicated to be used in simulation except for the smallest configurations in todays systems-on-a-chip. The devices on these chips are relatively fast, so it might be that a very common assumption in transistor modeling is still justifiable, namely that the charge distribution at any moment is completely determined by the actual terminal voltages at that moment. This assumption is called the *quasi-static assumption* [16]. It can only be valid when the terminal voltage variations are slow enough, and with the impact of interconnect in deep submicron circuits, this is likely to be so.

In the following treatment we assume quasi-static operation, meaning that the charge distribution is a function of the terminal voltages. Total charge inside a volume can vary when a net flow is exchanging charge between the volume and its exterior. A *charge current* is associated with this charge flow. Beside the transport current (derived by analyzing the static situation), the quasi-static analysis of the transistor has to deal with these charge currents that redistribute charge in response to variations in the terminal voltages. The charge current through a terminal is a measure for the rate of change of the charge associated with that terminal over the

time.

The operation of the transistor inside a digital circuit can be regarded as a large-signal operation. Usually the terminal voltages make full transitions between the negative and the positive supply rail voltages and consequently the operation point of the transistor crosses the borders between non-conducting, triode, and saturation regions. Even if at the end of a switching cycle the voltage on a terminal is clamped one threshold voltage away from a supply rail voltage, large-signal operation is still present. The large-signal operation implies that the charges associated with the terminals show large and strong non-linear variations with the terminal voltages when changing the operating region or even inside the same operating region. (Consider only the case of the drain inversion layer charge in triode region versus the gate-to-bulk voltage.) The attempt to model the charge variations with capacitors is laborious in order to be successful; the result is non-linear capacitors (small-signal capacitors dependent on the operating point) and is incorporated in very detailed models [29].

Our aim here is to propose a compact dynamic characterization for the transistor operating under full swing conditions, suited for the fast and accurate manipulation of large transistor circuits. Recalling the premises of the quasi-static operation we proceed to an algebraic formulation:

$$q_D = \mathbf{q_D}(v_{DB}, v_{GB}, v_{SB}) \tag{6.1a}$$

$$q_G = \mathbf{q_G}(v_{DB}, v_{GB}, v_{SB}) \tag{6.1b}$$

$$q_S = \mathbf{q_S}(v_{DB}, v_{GB}, v_{SB}) \tag{6.1c}$$

$$q_B = \mathbf{q_B}(v_{DB}, v_{GB}, v_{SB}) \tag{6.1d}$$

In (6.1) all the variables reflect large-signal (total) quantities with time variation. The charge associated with each terminal depends on the voltage drops across the MOS structure; the set of three independent voltages v_{DB}, v_{GB}, and v_{SB}, with the bulk as the common reference, has been chosen to enact this dependency. The functions $\mathbf{q_D}$, $\mathbf{q_G}$, $\mathbf{q_S}$, and $\mathbf{q_B}$ are identical to those which give the static charge distribution.

The charge conservation law is enforced by imposing

$$q_D + q_G + q_S + q_B = 0 \tag{6.2}$$

at any time instant during the device operation. It is important to recognize the three causes of charge inside a fabricated transistor structure: (1) intrinsic, related to the mechanism of inverting the semiconductor surface; (2) stray overlap, due to the technological overlaps between the gate contact, on one side, and the drain, source, and bulk regions, on the other side; and (3) stray reverse-biased junction, due to the *pn* junctions from the diffused contacts on the inverted channel to the bulk material.

From these three charge contributions, only the second one, the overlap, may be modeled with constant linear capacitors [28]. The other two contributions are strongly dependent on the terminal voltages. The charges contributed through the intrinsic operation mechanism of the transistor may be accurately calculated starting from the following integrals:

$$q_I = W\int_0^L q_I^{'} dy, \quad q_D = W\int_0^L \frac{y}{L} q_I^{'} dy, \quad q_S = W\int_0^L (1-\frac{y}{L}) q_I^{'} dy, \tag{6.3}$$

$$q_G = W\int_0^L q_G^{'} dy, \quad \text{and} \quad q_B = W\int_0^L q_B^{'} dy. \tag{6.4}$$

As the charge densities per unit area, $q_X^{'}$, are strongly dependent on the operating region, it appears that the result of the integration will inherit the same dependency. The small-signal intrinsic (trans)capacitances, each defined as the charge sensitivity to a terminal voltage, appear to be dependent on the operating point. The charges contributed by the reverse-biased junctions are obeying the basic theory of the abrupt *pn* junction: they depend on the physical parameters of the junction and have a polynomial variation with the reverse bias.

The time derivative of the total charge of a terminal defines the charge current associated with that terminal. The charge currents associated with the transistor

terminals are spelled out as:

$$i_{DC} = \frac{\mathrm{d}q_D}{\mathrm{d}t} \qquad i_{GC} = \frac{\mathrm{d}q_G}{\mathrm{d}t} \tag{6.5a}$$

$$i_{SC} = \frac{\mathrm{d}q_S}{\mathrm{d}t} \qquad i_{BC} = \frac{\mathrm{d}q_B}{\mathrm{d}t} \tag{6.5b}$$

From the static operation[1] of the transistor it was recognized that only for the drain and source terminals we have transport current.

Charge-wise the transistor terminals may be divided into two classes:

- terminals which only accumulate charge, G and B;
- terminals which accumulate charge *and* pass charge over to the other terminal from the same class, D and S.

Charge accumulation is related to the charge current and charge passing is related to the transport current. The total terminal currents appear to be:

$$i_D = i_T + i_{DC} \qquad i_G = i_{GC} \tag{6.6a}$$

$$i_S = -i_T + i_{SC} \qquad i_B = i_{BC} \tag{6.6b}$$

i_T is the large-signal transport current and is equal under the quasi-static assumption to the static transport current I_T calculated using the static formulation with the instant terminal voltages instead of the static voltages.

In Figure 6.1 we indicate where the terminal charges are located and how the charge and transport currents combine to give the total drain/source current. The drain/source charge current is tributary to the charge in the corresponding charge region. According to the actual tendency of the terminal voltages, the amount of charge inside a charge region is increased or decreased.

[1]The transistor is considered to operate under normal biasing conditions: both source-to-bulk and drain-to-bulk junctions are reverse biased.

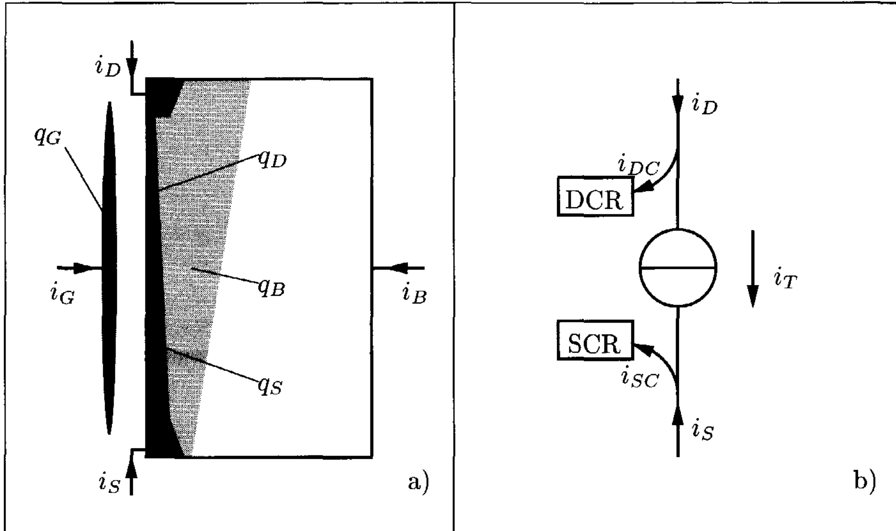


Figure 6.1: a) Localization of the terminal charges in the transistor structure; b) Channel region representation with transport current i_T and charge currents i_{DC}, i_{SC}. The charge currents may be regarded as contributing charge to the drain/source charge regions (DCR/SCR).

6.2 Modeling large-signal charge variations

Consider an arbitrary CMOS circuit and mark all the devices connected with at least one terminal to the positive supply rail (Figure 6.2). Assume that for each terminal i from the accumulation-only class the charge variation $\Delta q_i^{(1)}$ over a certain time interval $[0, T]$ is known, and also that for each terminal j from the accumulation-and-pass class the charge variation $\Delta q_j^{(2)}$ and the transport current $i_{T,j}^{(2)}(t)$ over the time interval $[0, T]$ are known. With this information one may calculate the charge supplied by the power source to the digital circuit in the time interval $[0, T]$ as:

$$\Delta q_{sup} = \sum_i \Delta q_i^{(1)} + \sum_j \Delta q_j^{(2)} + \sum_j \int_0^T i_{T,j}^{(2)}(t)dt \tag{6.7}$$

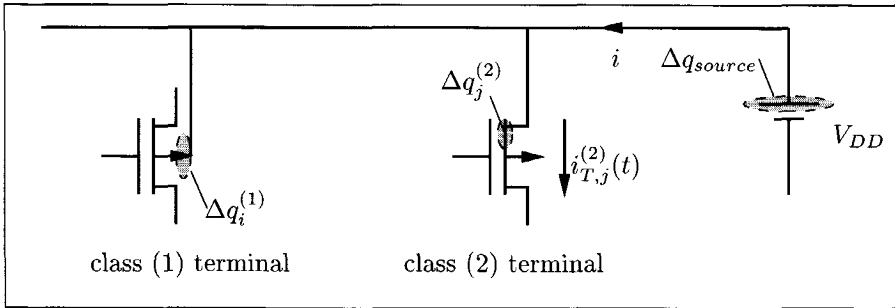


Figure 6.2: Representative p transistors connected to the positive supply rail (1) with an accumulation-only class terminal (2) with an accumulation-and-pass class terminal

The energy consumption of the digital circuit may be further calculated as:

$$E = \int_0^T (u\, i)dt = V_{DD} \int_0^T i dt = V_{DD} \cdot \Delta q_{sup} \tag{6.8}$$

It is sufficient to know the charge variation and the transport current associated with each of the device terminals connected to the positive supply rail to obtain the energy consumption over a time interval.

Take the case of a p-transistor connected only with the bulk to the positive supply rail[2] depicted in Figure 6.3a. Assume that for each switching event (generated by the switching of at least one input) the other terminals are large-signal active, i.e., they undergo well-defined large-signal voltage swings: Δv_{DB}, Δv_{GB}, and Δv_{SB}. Assume also that a method is available to evaluate the corresponding terminal charge variations: Δq_D, Δq_G, and Δq_S (each Δq_X is the effect of the large-signal voltage variations on *all* the active terminals).

We may model the large-signal charge variation associated with an active terminal with an average (large-signal) capacitor connected between that terminal

[2]Just think of p-transistor in a *nor*2 cell, connected on the drain-end to the output terminal of the cell. First is switched off the other p-transistor is switched off, then the one under consideration.

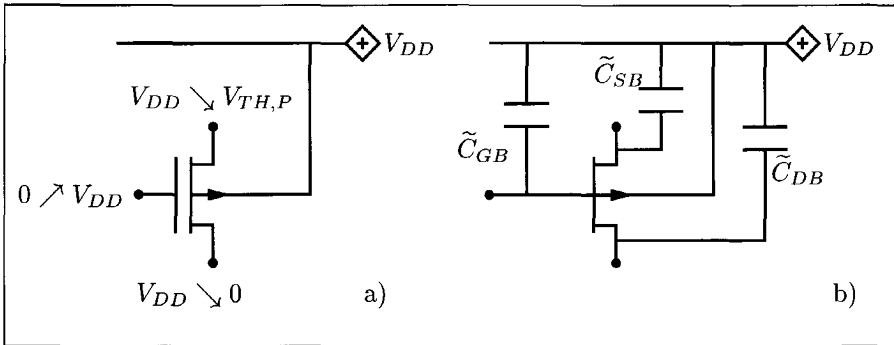


Figure 6.3: *p*-transistor operating under large-signal conditions a) A complex switching event b) Dynamic modeling using average capacitors around the static model

and the bulk. The value of the average capacitor is given by:

$$\widetilde{C}_{XB} = \frac{\Delta q_X}{\Delta v_{XB}}, \quad \text{with } X = [D, G, S]. \tag{6.9}$$

In Figure 6.3b we show how the *p*-transistor operating in full swing conditions has been modeled with average capacitors (accounting for the dynamic charge variations) added to the static model (accounting for the transport current controlled by the terminal voltages). It appears that the model satisfies inherently the charge conservation law (6.2), as each average large-signal capacitor has one terminal connected to the bulk.

Two particular situations may occur: (1) besides the bulk there is an additional terminal (usually the source) connected to the supply rail, and (2) during a switching event the voltage of a floating terminal does experience none or insignificant variation.

In both situations there is charge variation, but no voltage swing. The latter situation is considered not of interest for the large-signal capacitance calculation and a different switching event has to be selected for this calculation.

In the first situation the definition of an average capacitor associated with the extra terminal connected to the supply rail is no longer possible ($\Delta v_{XB} = 0$). Can

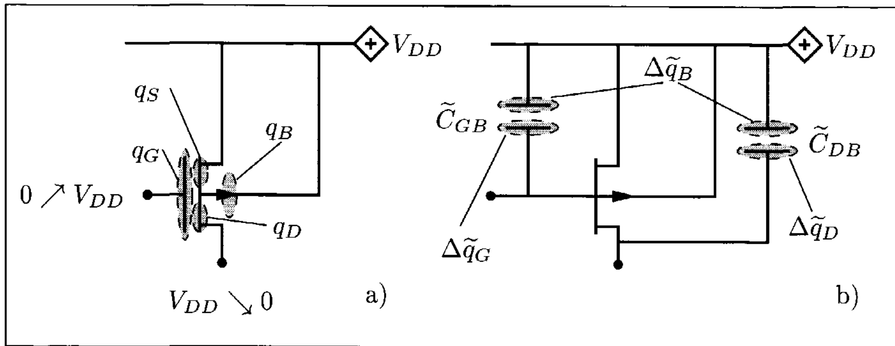


Figure 6.4: p-MOS transistor with both the bulk and the source terminals connected to the positive supply rail a) Large-signal digital operation b) Dynamic modeling with a reduced set of average capacitors

we model the other charge variations by average capacitors to estimate the charge supplied by the power source? With reference to Figure 6.4, the charge supplied to the circuit via the transistor under consideration is calculated for the case with exact modeling as:

$$\Delta q_{sup} = \Delta q_B + \Delta q_S + \int_0^T i_T(t)dt \tag{6.10}$$

and for the case with large-signal capacitors as:

$$\Delta \widetilde{q}_{sup} = \Delta \widetilde{q}_B + \int_0^T \widetilde{i}_T(t)dt \tag{6.11}$$

In both cases there is charge conservation: $\Delta q_B + \Delta q_S = -(\Delta q_D + \Delta q_G)$ and $\Delta\widetilde{q}_B = -(\Delta\widetilde{q}_D + \Delta\widetilde{q}_G)$. By construction the large-signal charge variations are identical in both situations, i.e., $\Delta\widetilde{q}_D = \Delta q_D$ and $\Delta\widetilde{q}_G = \Delta q_G$. Assuming that the transport current for the average capacitors model has the same time evolution as for the exact model, i.e., $\widetilde{i}_T(t) = i_T(t)$:

$$\Delta q_{sup} \equiv \Delta \widetilde{q}_{sup} \tag{6.12}$$

We have just proved the following:

Lemma. *The average capacitors approach gives the same estimate for the energy consumption as the exact approach, if the time-evolution of the terminal voltages and of the transport current are identical in both approaches.*

The assumption that the transport current for the average capacitors model has the same time evolution as for the exact model is violated for at least two reasons: (1) the static model we use with the average capacitors predicts a slightly different transport current compared with the exact static model, even when the terminal voltages are the same; and (2) the terminal voltages for the average capacitors model are slightly different from those for the exact model, even when the same static model is used. We can formally express this two-side error as:

$$i_T(t) = \mathbf{i_T}\big(v_{DB}(t), v_{GB}(t), v_{SB}(t)\big) \tag{6.13a}$$

$$\simeq \widetilde{\mathbf{i}}_\mathbf{T}\big(v_{DB}(t), v_{GB}(t), v_{SB}(t)\big) \tag{6.13b}$$

$$\approx \widetilde{\mathbf{i}}_\mathbf{T}\big(\widetilde{v}_{DB}(t), \widetilde{v}_{GB}(t), \widetilde{v}_{SB}(t)\big) = \widetilde{i}_T(t) \tag{6.13c}$$

The approximation in (6.13b) is due to the the collapsible current source model linearizations and the approximation in (6.13c) owes to the constant average capacitors modeling. We will claim that the errors introduced by these approximations stay within reasonable limits.

Claim 1. *The average capacitors approach is within 5% error for the energy consumption estimation when compared with the exact approach.*

Claim 2. *The average capacitors approach is within 5% error for the timing estimation when compared with the exact approach. Also the waveforms for the average capacitors approach are good approximations for those obtained with the exact approach.*

Both claims are supported by the experimental results to be presented in Chapter 8.

6.3 Evaluating large-signal charge variations

Up to now we assumed the availability of a method to evaluate the large-signal charge variations inside a transistor. Now we present such a method.

Consider a transistor connected with the bulk to the appropriate supply rail and with the drain, gate, and source to active nets. Let A be the initial state and B the final state of a digital switching event, i.e., at the times t_A and t_B the terminal voltages reach a steady state. For generality we accept that all the active terminals of the transistor under consideration experience large-signal voltage sweeps during the A→B transition.

Our goal is to find quantitative expressions for the total charge variations $\Delta q_D^{A\to B}$, $\Delta q_G^{A\to B}$, and $\Delta q_S^{A\to B}$. In the quasi-static assumption the charge variations depend *only* on the initial and final terminal voltages; the charge variations are independent on how the transition between A and B is performed.

By differentiating (6.1) we find the small-signal charge variations due to small-signal terminal voltage variations:

$$dq_D = \frac{\partial \mathbf{q_D}}{\partial v_{DB}} dv_{DB} + \frac{\partial \mathbf{q_D}}{\partial v_{GB}} dv_{GB} + \frac{\partial \mathbf{q_D}}{\partial v_{SB}} dv_{SB} \tag{6.14a}$$

$$dq_G = \frac{\partial \mathbf{q_G}}{\partial v_{DB}} dv_{DB} + \frac{\partial \mathbf{q_G}}{\partial v_{GB}} dv_{GB} + \frac{\partial \mathbf{q_G}}{\partial v_{SB}} dv_{SB} \tag{6.14b}$$

$$dq_S = \frac{\partial \mathbf{q_S}}{\partial v_{DB}} dv_{DB} + \frac{\partial \mathbf{q_S}}{\partial v_{GB}} dv_{GB} + \frac{\partial \mathbf{q_S}}{\partial v_{SB}} dv_{SB} \tag{6.14c}$$

Each quantity $\frac{\partial \mathbf{q_X}}{\partial v_{YB}}$ represents the sensitivity of the charge associated with the X terminal, q_X, with respect to the voltage on the Y terminal, v_{YB}. These sensitivities have the dimension of a capacitance and depend on the actual bias point:

$$\frac{\partial \mathbf{q_X}}{\partial v_{YB}} = \mathbf{f_{XY}}(v_{DB}, v_{GB}, v_{SB}).$$

$C_{XX} \triangleq \frac{\partial \mathbf{q_X}}{\partial v_{XB}}$ is named *capacitance* and $C_{XY} \triangleq -\frac{\partial \mathbf{q_X}}{\partial v_{YB}}$ with $X \neq Y$ is named *transcapacitance*. Relations (6.14) inherit the fact that the charge associated with one terminal is determined by all the terminal voltages.

We may calculate the total charge variations during the A→B state transition by integrating (6.14) between the generic points A and B:

$$\Delta q_D^{A\to B} = \int_A^B \frac{\partial \mathbf{q_D}}{\partial v_{DB}} dv_{DB} + \int_A^B \frac{\partial \mathbf{q_D}}{\partial v_{GB}} dv_{GB} + \int_A^B \frac{\partial \mathbf{q_D}}{\partial v_{SB}} dv_{SB} \quad (6.15a)$$

$$\Delta q_G^{A\to B} = \int_A^B \frac{\partial \mathbf{q_G}}{\partial v_{DB}} dv_{DB} + \int_A^B \frac{\partial \mathbf{q_G}}{\partial v_{GB}} dv_{GB} + \int_A^B \frac{\partial \mathbf{q_G}}{\partial v_{SB}} dv_{SB} \quad (6.15b)$$

$$\Delta q_S^{A\to B} = \int_A^B \frac{\partial \mathbf{q_S}}{\partial v_{DB}} dv_{DB} + \int_A^B \frac{\partial \mathbf{q_S}}{\partial v_{GB}} dv_{GB} + \int_A^B \frac{\partial \mathbf{q_S}}{\partial v_{SB}} dv_{SB} \quad (6.15c)$$

As we recognized earlier, $\Delta q_X^{A\to B}$ depends only on the initial and final states. To render the integral calculations easier, we may choose a particular path between A and B passing through the intermediate states C and D which satisfy $t_A < t_C < t_D < t_B$ and $\Delta v_{GB}^{A\to C} = \Delta v_{SB}^{A\to C} = 0$, $\Delta v_{DB}^{C\to D} = \Delta v_{SB}^{C\to D} = 0$, and $\Delta v_{DB}^{D\to B} = \Delta v_{GB}^{D\to B} = 0$. In this particular situation the terminal voltages execute their large-signal swings not overlapping in time throughout the interval $[t_A, t_B]$, but separately: v_{DB} in the interval $[t_A, t_C]$, v_{GB} in the interval $[t_C, t_D]$, and v_{SB} in the interval $[t_D, t_B]$. The set (6.15) is rewritten for this particular case as:

$$\Delta q_D^{A\to B} = \int_A^C \frac{\partial \mathbf{q_D}}{\partial v_{DB}} dv_{DB} + \int_C^D \frac{\partial \mathbf{q_D}}{\partial v_{GB}} dv_{GB} + \int_D^B \frac{\partial \mathbf{q_D}}{\partial v_{SB}} dv_{SB} \quad (6.16a)$$

$$\Delta q_G^{A\to B} = \int_A^C \frac{\partial \mathbf{q_G}}{\partial v_{DB}} dv_{DB} + \int_C^D \frac{\partial \mathbf{q_G}}{\partial v_{GB}} dv_{GB} + \int_D^B \frac{\partial \mathbf{q_G}}{\partial v_{SB}} dv_{SB} \quad (6.16b)$$

$$\Delta q_S^{A\to B} = \int_A^C \frac{\partial \mathbf{q_S}}{\partial v_{DB}} dv_{DB} + \int_C^D \frac{\partial \mathbf{q_S}}{\partial v_{GB}} dv_{GB} + \int_D^B \frac{\partial \mathbf{q_S}}{\partial v_{SB}} dv_{SB} \quad (6.16c)$$

For the transistor under consideration, we have:

Statement. *The charge sensitivities with respect to the terminal voltages (small-signal capacitances and transcapacitances) are easy to deduce once the* **y***-parameters at a certain frequency are known.*

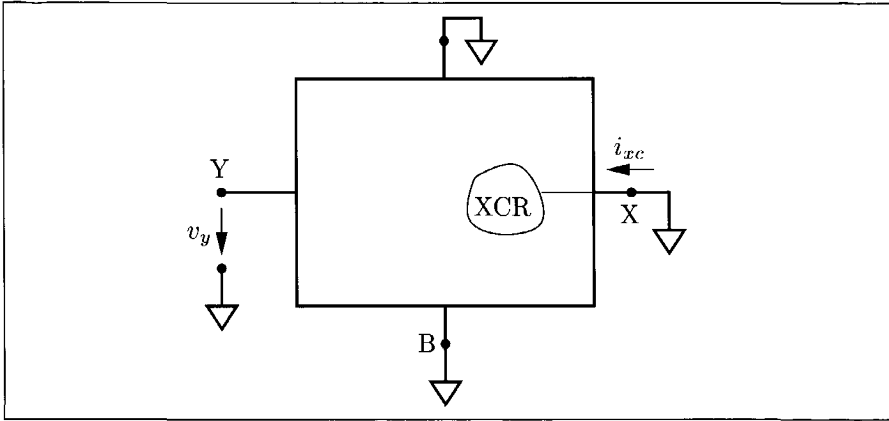


Figure 6.5: Experimental setup for the measurement of the $\mathbf{y_{xy}}$-parameter

Support for the statement. For the calculation of $\frac{\partial \mathbf{q_X}}{\partial v_{YB}}$ we consider the experimental setup of Figure 6.5 where all terminals except Y are AC-grounded. The necessary DC-biasing is not shown. A small-signal AC-voltage v_y with frequency f is applied at terminal Y and the small-signal short-circuit charge current i_{xc} is measured at terminal X. In the time domain we have the following dependency:

$$i_{xc} = \frac{dq_X}{dt} = \frac{\partial \mathbf{q_X}}{\partial v_{YB}} \cdot \frac{dv_{YB}}{dt} = \frac{\partial \mathbf{q_X}}{\partial v_{YB}} \cdot \frac{dv_y}{dt} \tag{6.17}$$

Relation (6.17) is spelled out in the frequency domain as:

$$\frac{\partial \mathbf{q_X}}{\partial v_{YB}} = \frac{Im(\mathbf{y_{xy}})}{2\pi f} \tag{6.18}$$

$\mathbf{y_{xy}}$ is the high-frequency complex parameter that relates the complex current $\mathbf{I_x}$ in terminal X to the complex voltage $\mathbf{V_y}$ at terminal Y, in the conditions of the above setup.

$$\left.\frac{\mathbf{I_x}}{\mathbf{V_y}}\right|_{\mathbf{V_k}=0, k \neq y} \triangleq \mathbf{y_{xy}} = Re(\mathbf{y_{xy}}) + j \cdot Im(\mathbf{y_{xy}}) \tag{6.19}$$

□

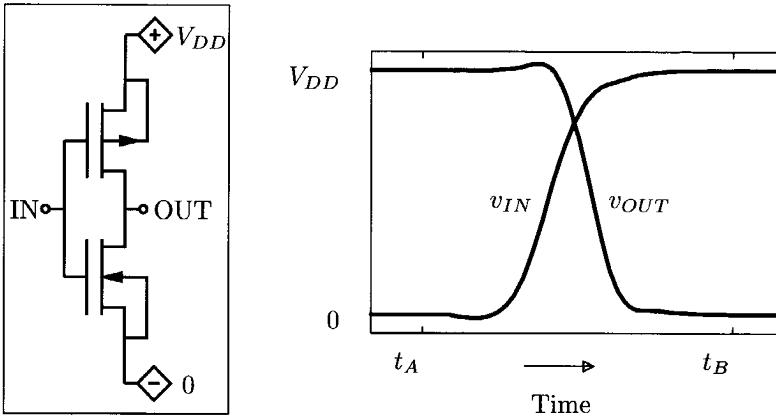


Figure 6.6: Static CMOS inverter and input/output waveforms during normal operation

With the **y**-parameters available from high frequency measurements, we have reduced the charge variation calculations to simple numerical integrations.

6.4 Dynamic characterization of an inverter

Figure 6.6 shows the connection of the p- and n-transistors in the static inverter configuration and the input/output voltage waveforms for a high-to-low switching event. At times t_A and t_B the voltage signals have stabilized at the 0 or V_+ levels. We are going to follow in some detail the calculation of $\Delta q_{D,n}^{A\to B}$ for the n-transistor.

Example. Δq_D calculation for the n-transistor in the inverter configuration.

As stated on page 81, the drain charge variation during the $A \to B$ transition does not depend on the series of intermediate states between A and B. We choose the intermediate state C as in Figure 6.7 in order to decouple the gate voltage variation from the drain voltage variation and thus render an easy integral calculation. Two

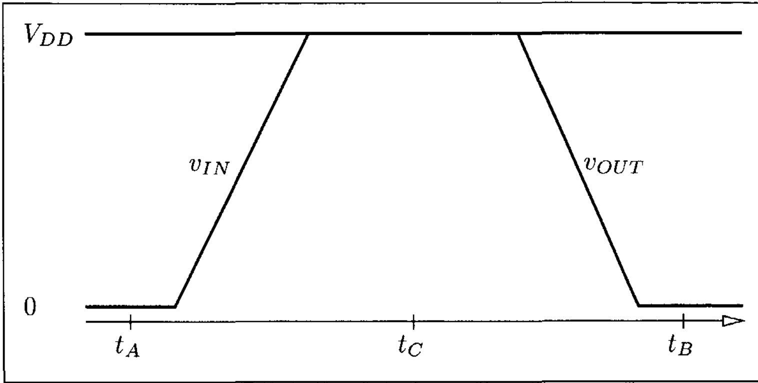


Figure 6.7: Decoupled input/output voltage waveforms used for the calculation of large signal charge variations

steps are taken for the integration:

$$A \to C \begin{cases} v_{DB} = V_{DD} \\ v_{GB} = (0 \nearrow V_{DD}) \end{cases} \quad \text{and} \quad C \to B \begin{cases} v_{DB} = (V_{DD} \searrow 0) \\ v_{GB} = V_{DD} \end{cases}$$

The drain charge variation becomes:

$$\begin{aligned} \Delta q_{D,n}^{A \to B} &= \int_A^C \frac{\partial \mathbf{q_D}}{\partial v_{GB}} dv_{GB} + \int_C^B \frac{\partial \mathbf{q_D}}{\partial v_{DB}} dv_{DB} \\ &= \int_A^C \frac{Im(y_{dg})}{2\pi f} dv_{GB} + \int_C^B \frac{Im(y_{dd})}{2\pi f} dv_{DB} \end{aligned}$$

f is the frequency at which the RF-parameters y_{dg} and y_{dd} have been tabulated as functions of the static point voltages V_{GB} and V_{DB}. □

Similar calculations may be carried out for $\Delta q_{G,n}^{A \to B}$, $\Delta q_{D,p}^{A \to B}$, and $\Delta q_{G,p}^{A \to B}$. As the voltage variations for each active terminal are known and well defined, the

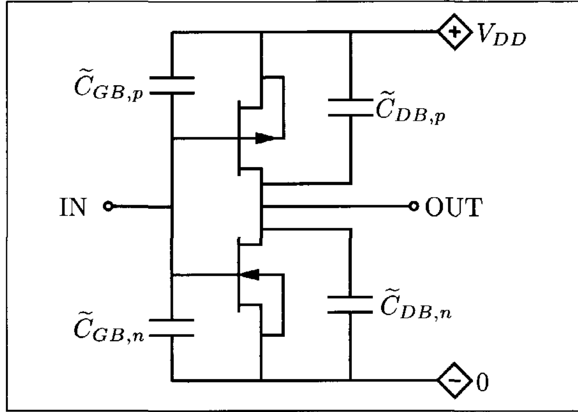


Figure 6.8: Complete model for a CMOS inverter using large-signal average capacitors for the charge variations and collapsible current source models for the transport currents

dynamic characterization of the inverter is completed by setting

$$\widetilde{C}_{DB,n} = \frac{\Delta q_{D,n}^{A\to B}}{\Delta v_{OUT}^{A\to B}} \qquad \widetilde{C}_{GB,n} = \frac{\Delta q_{G,n}^{A\to B}}{\Delta v_{IN}^{A\to B}}$$

$$\widetilde{C}_{DB,p} = \frac{\Delta q_{D,p}^{A\to B}}{\Delta v_{OUT}^{A\to B}} \qquad \widetilde{C}_{GB,p} = \frac{\Delta q_{G,p}^{A\to B}}{\Delta v_{IN}^{A\to B}}$$

In Figure 6.8 we show how the average capacitors are to be connected around the static models of the transistors in order to obtain accurate power consumption and waveform estimations.

6.5 Average capacitors

We considered the dynamic operation of the MOS transistor in digital circuits. We proposed to model the large-signal terminal charge variations with average capacitors. For this analysis we assumed quasi-static operation.

We observed that the charges associated with the MOS transistor terminals

have a non-linear dependency on the potentials on all four terminals. Therefore, to build a compact dynamic model for the MOS transistor we proposed to use average capacitors. We proved that modeling of charge currents with large-signal average capacitors is an analytically sound method in what concerns the power consumption and waveform estimations for a digital CMOS circuit.

We described a practical method to calculate the average capacitances starting from the RF-parameters of the MOS transistor.

Chapter 7

One language, one algorithm

Contents

7.1 Voltage and current events 90
7.2 Constitutive processes 92
7.3 Topological processes 93
7.4 Model calibration 95
7.5 Using the VHDL-MOS package 98

The investigations from the previous chapters for an efficient system-on-a-chip simulation approach have revealed the following necessary items: (1) a compact, minimal model for the transistor, consisting of the piecewise linear formulation for the static behavior and of the average capacitors formulation for the dynamic behavior, and (2) an efficient simulation method, based on the event-driven paradigm. The efficiency of the event-driven simulation method stems both from the usage of one single algorithm for manipulating descriptions at multiple levels of abstraction and from the combination of local modeling with global signaling. At present, event-driven simulators can accommodate system components described at gate, register, or higher levels. One more item is needed to reach a consistent approach for system-on-a-chip simulation: (3) a package to capture the transistor level operation of a system component, and accepting as input the same event-driven modeling language that is used for the higher abstraction levels.

In principle, the modeling concept that we will follow here should be feasible for implementation in VHDL [25], Verilog [26], or any other hardware description language accepted as input language by an event-driven simulator. Here we use the VHDL language (see Appendix C for some language concepts) because its feature of strong typing offers the user the freedom of defining own types. The VHDL package that we propose is targeted at capturing continuous aspects of circuit operation in an event-driven simulator, a concept originally meant for discrete circuits.

7.1 Voltage and current events

In general, the event-driven operation of a system is rendered by the sequential evolution of each system variable through values from a finite discrete set. A system variable changing from one to another value in the set is associated with an event at the calculated moment of time. The interaction between any two system components, from the same or from different abstraction levels, is modeled by event passing.

In order to capture the continuous behavior of an electrical system within an event-driven simulator, voltages and currents have to be taken into consideration. An enumeration type, named `VL`, is defined for replacing the continuous voltages, and a real type, named `MC`, is provided for the currents.[1]

The discrete voltage levels are equally spaced between the ground and the power supply limits. The spacing (`GAP`) between the voltage levels determines the voltage resolution. An example of the `VL` type declaration is given in the first part of Figure 7.1.

The currents are allowed to take values inside a real interval, by defining a real type for currents. The same goes for currents through a current controlled device. The currents through a voltage controlled device will however have discrete values, given the discrete `VL` type. An example of the `MC` type declaration is shown in the middle part of Figure 7.1.

The two types `VL` and `MC` are combined into a record type called `TERM`. Each port of an entity is of the type `TERM`. Because a terminal of a device is either voltage or current controlled, the mode of a port must be "inout". Consequently the types `VL` and `MC` are of a "resolved" kind. The type declaration for `TERM` is given in the

[1] `VL` stands for Voltage Levels, while `MC` stands for Mos transistor Current.

last part of Figure 7.1.

```
constant VDD : real := 2.7;
constant GAP : real := VDD/8.0;
--
type VL_unres is ('0','1','2','3','4','5','6','7','8');
subtype VL is resolve_VL VL_unres;
--
type MC_unres is range -1.0 to +1.0;
subtype MC is resolve_MC MC_unres;
--
type TERM is record
      v : VL;
      i : MC;
end record;
```

Figure 7.1: VHDL code used to declare the VL, MC, and TERM types

To keep the model implementation task manageable, we have to be aware of the controlled nature of the various circuit level components.

- The static MOS transistor is a voltage controlled device: the terminal voltages completely determine the terminal currents; or using event terminology: a voltage event on any terminal generates a transaction on each terminal current.

- The capacitor is a current controlled device: a current event on its variable terminal schedules a voltage transaction on that terminal.

- The electrical net is implemented as a stand-alone component: it has voltage controlled terminals to interface with voltage controlled devices, and current controlled terminals to interface with current controlled devices.

- The resistor is a voltage controlled device: voltage events on any terminal generate current transactions.

7.2 Constitutive processes

Transistor The modified collapsible current source model is characterizable by two parameters: the threshold voltage V_T and the width of the submicron transistor W, which modulate the strength of the equivalent current source. This model is voltage controlled, as the terminal voltages completely determine the terminal currents. The V_{GS} vs. V_{DS} control voltage plane is divided into four operating regions (non-conducting, forward-saturation, reverse-saturation, and triode) by four half-lines with a common intersection point, as shown in Figure 4.6 on page 54.

The VHDL implementation of the collapsible current source model is based on a table look-up method. The terminal voltages determine the operating region and the drain current is calculated accordingly. The behavior is confined inside one process which is sensitive to voltage events on the four terminals. Each voltage event triggers the calculation of the actual operating region and the process ends up assigning new values to the terminal currents. □

Linear capacitor Average linear capacitors were introduced in Chapter 6 for the compact modeling of large signal charge variations in the MOS transistor structure. Essentially, these linear capacitors are virtually grounded, i.e., from the two terminals only one can experience potential variations.

The voltage across the capacitor is proportional to the time integral of the current through the variable terminal. So the capacitor is a current controlled device. Yet, although the voltage across the capacitor is quantized due to the `VL` type the model keeps track internally of the continuous variation of the voltage on the capacitor. The integral is calculated incrementally. Current integration over time can be translated into event language as that at each integration step a transaction is put on the voltage driver for a future time (with a certain time delay).

Assume that during actual simulation the current through the variable terminal gets imposed a new value. The time delay needed for the voltage on the capacitor to reach the next voltage level is then calculated by:

$$time_delay = capacitance \times \frac{|next_voltage_level - continuous_voltage|}{|new_current|} \tag{7.1}$$

Note that if the current variation through the capacitor were to be precise, then the timing of the voltage events would be precise too.

What should the delay type be for the assignment to the voltage signal of the variable terminal? Transport delay is not the right choice, because an existing scheduled transaction is not removed when a new transaction is put on the driver of the voltage signal after the already existing one. Were the delay of transport type then the prediction of the capacitor voltage would be in error. The problem is less severe for the inertial type of delay, when an already existing transaction is removed by the new transaction scheduled for a later time, except for the case when the two transactions schedule the same voltage level. To completely solve the problem, the enumeration type `VL` has been enhanced with an exceptional item dedicated only to removal purposes. This item was baptized `'X'` and in Figure 7.2 we indicate the intended usage sequence. The time constant `infinity` has a relatively high

```
pin1.v <= 'X' after infinity;
-- other sequential statements
pin1.v <= next_level after delay;
```

Figure 7.2: Assigning values to the variable terminal voltage of a capacitor

value, usually not encountered for normal delays. It may be assigned the highest representable time value for the available simulator (`time'high`). The first inertial delay signal assignment says that all existing transactions will be removed from the driver of the signal, given that none has scheduled a value of `'X'` and all are scheduled earlier then `infinity` elementary time units from `now` on. The second inertial delay signal assignment finds only an `'X'` scheduled after `infinity` elementary time units. The `next_level` transaction scheduled after `delay` will remove the existing `'X'` transaction, as the former appears before the latter. After these two signal assignments have been executed, only the desired `next_level` transaction will be present on the driver of the signal. □

7.3 Topological processes

The individual components have to be glued together in order to obtain the electric circuit functionality. The abstraction for the "glue" is called electrical net. Its function is to enforce Kirchhoff's laws: voltages around a loop of nets add to zero

(Kirchhoff voltage law), and currents entering a net add to zero (Kirchhoff current law). Kirchhoff voltage law has the under-laying assumption that all the terminals connected to a net have the same potential.

Supply net The supply net is meant to model the connection of device terminals to constant voltage sources. It has a constant potential, imposed by one privileged terminal, where the voltage source is connected. A supply net has one current controlled terminal and one or more voltage controlled terminals.

The operation of the supply net, described in event terminology is:

- each voltage event on the current controlled terminal is replicated over all its voltage controlled terminals (Kirchhoff voltage law), and
- each current event on any of the voltage controlled terminals triggers the calculation of the current through the current controlled terminal as the sum of the currents through the voltage controlled terminals (Kirchhoff current law).

With this approach to modeling the power supply nets, it is straight forward to obtain the current consumption of a cluster of gates: each gate has its power supply net and there is one extra power supply net for the cluster, which adds the individual current consumptions of the gates. □

Capacitive net The capacitive net has one or more capacitor variable terminals connected to it and can be extended with other capacitive net(s). The potential on a capacitive net can float between the ground and the supply potential limits.

A capacitive net has one or more current controlled terminals and one or more voltage controlled terminals. The current controlled terminals are meant for connecting to capacitors or capacitor-like devices, while the voltage controlled terminals are meant for connecting to voltage controlled devices (e.g., transistors, resistors).

The normal operation resembles that of the supply net. A voltage event on any current controlled terminal is replicated over the voltage controlled terminals (Kirchhoff voltage law). A current event on any voltage controlled terminal triggers the summation of the currents through the voltage controlled terminals, followed by the weighted distribution of this sum current to the current controlled terminals (Kirchhoff current law).

A major difference between this capacitive net and the supply net is the behavior during the initialization phase. This has been adapted to allow for correct manipulation of multiple capacitors and capacitor-like devices connected to a capacitor net. This situation occurs when the user wants to fan-out the signal from the output of a gate to an arbitrary number of load gates. This is equivalent to distributing the total charge/discharge current available from the output to the various capacitances connected to the ground or to the power supply lines. But the output does not know a priori which one will be the external capacitive load. A solution based on the event driven paradigm was identified (see [30]); it allows a correct waveform estimation at the output of a gate for any loading conditions and the distribution of this waveform to all the supplied inputs. □

7.4 Model calibration

In Chapter 3 a compact PL model was presented for the transport current through the channel of a deep submicron MOS transistor, namely the collapsible current source model. Then, in Chapter 6 a compact model was introduced for the large signal charge variations in the MOS transistor structure, model which makes use of average linear capacitors. To obtain accurate simulation results (waveforms, delay, power) when using these models with any of the simulation approaches from Chapter 2 or from Chapter 5, the designer has to carefully calibrate the parameters of the models. For the modified collapsible current source model one should set adequate values for k_1, k_2, and V_T and for a MOS transistor operating in a certain configuration one should "fix" the corresponding average capacitors to appropriate values.

From the user point of view the modified collapsible current source model has one design parameter, W-the width of the transistor, and one process parameter, V_T-the threshold voltage. For a given technology the model-developer should determine the values of k_1 and k_2 based on W and V_T.

In Chapter 3 it was shown that in the saturation region as well as in the triode region the current scales linearly with the transistor width (the $I_D = f(V_{DS})|_{V_{GS}=ct}$ characteristic scales in amplitude linearly with W). Moreover, the separation between the operation regions in the V_{GS} vs. V_{DS} control voltage plane is independent of the transistor width (see Figure 4.6 on page 54). From either of these two facts one may see that the large signal conductances k_1 and k_2 are proportional with W.

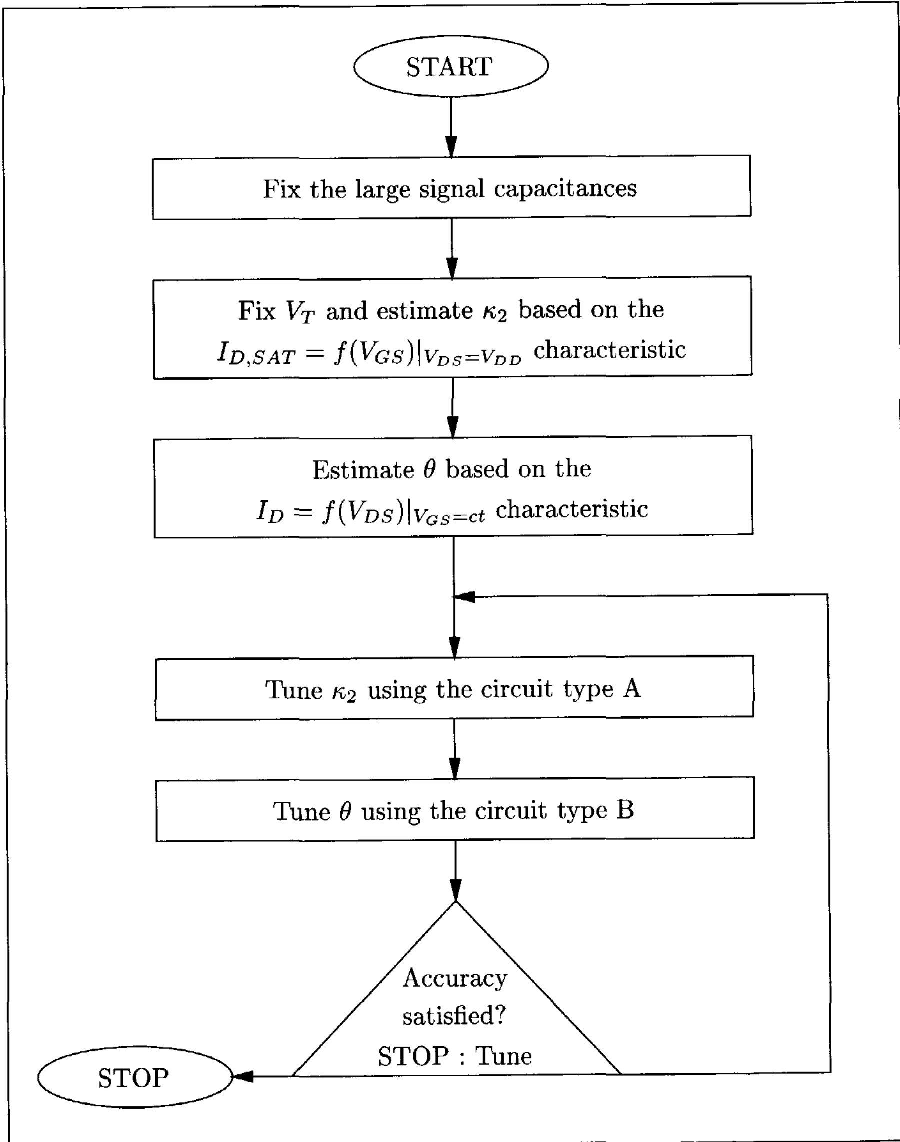


Figure 7.3: Iterative procedure for the collapsible current source model calibration

Consequently, the specific large signal conductances defined as

$$\kappa_1 = k_1/W \qquad \text{and} \qquad \kappa_2 = k_2/W \tag{7.2}$$

are technology constants. Their ratio is also a constant:

$$\kappa_1 = \theta\kappa_2, \qquad \text{with } \theta \text{ a technology constant.} \tag{7.3}$$

We remark that from the model-developer point of view it is convenient to use κ_2, θ, and V_T as model parameters. These parameters are to be deduced for a nominal width ($W = 1\mu m$) transistor. Transistors of any width are then calibrated automatically.

Figure 7.3 presents an iterative procedure for calibrating the static and the dynamic compact models associated with a MOS transistor. From charge conservation considerations, once the capacitor values for the transistors are calculated according to the methodology presented in Section 6.3, these values should be kept fixed during the calibration procedure.

Next, the V_T value is fixed based on the $I_{D,SAT} = f(V_{GS})|_{V_{DS}=V_{DD}}$ characteristic; using the same characteristic the value of κ_2 is estimated. The correlation parameter θ is then estimated based on the $I_D = f(V_{DS})|_{V_{GS}=ct}$ characteristic family.

The κ_2 parameter is responsible for the current in the saturation region. Its value is best tuned in a switching configuration where the effect of κ_2 is maximized with respect to the effect of κ_1. Such a configuration is a simple inverter driven by a fast input transition; we call such a configuration a `type A` circuit.

The κ_1 parameter is responsible for the current in the triode region. Its value is best tuned in a switching configuration where the effect of κ_1 is maximized with respect to the effect of κ_2. Such a configuration is where a series connection of transistor channels appears in the N- or the P-block of a CMOS gate, e.g., a NAND or a NOR gate; we call such a configuration a `type B` circuit.

With charge conservation satisfied by construction and accurate timing obtained by the above calibration procedure, one is assured of an accurate power consumption estimation when using the proposed approach.

The proposed calibration procedure may be performed having as reference either the simulation results obtained with complete, detailed models (as available with the SPICE simulator) or the direct measurements on fabricated test circuits.

In either situation the calibrator has to do static and transient measurements on a few conveniently chosen circuits. Then, the calibrated models are guaranteed to perform well in all situations.

7.5 Using the VHDL-MOS package

This section shows through a simple example how the VHDL-MOS components are to be put together. A static CMOS inverter is presented in Figure 7.4. At the

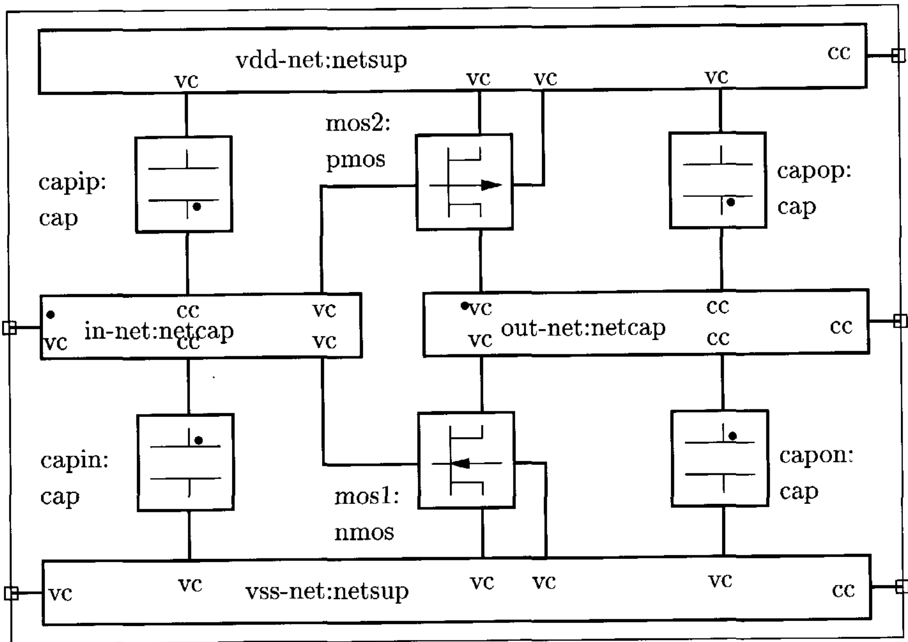


Figure 7.4: VHDL-MOS schematic of a CMOS inverter

schematic level the major difference when compared with traditional representations is the emphasis on electrical nets as separate components. A dot on a capacitor symbol indicates the variable terminal, while a dot on a capacitive net symbol represents the reference (first) voltage controlled terminal.

Figure 7.5 shows some simulated waveforms for the inverter. The input volt-

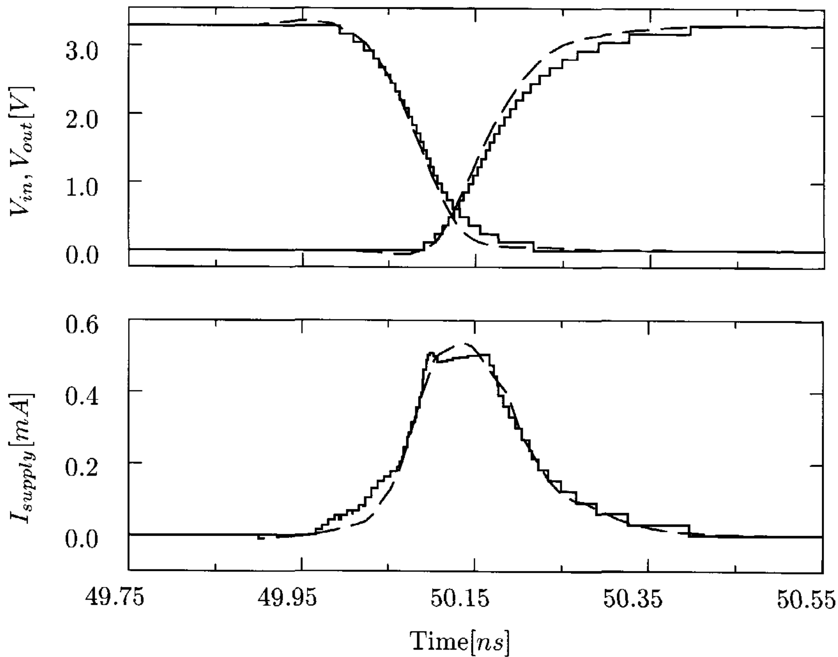


Figure 7.5: Simulated waveforms for a falling transition at the input of a static CMOS inverter obtained using the VHDL-MOS package; the SPICE reference appears in dashed line.

age is provided from the output of another gate and the output is loaded with a similar gate. The voltage waveforms have step transitions between the discrete voltage levels, hence the "analog-like" attribute. Both the input and the output voltage waveforms have clear analog features, e.g., finite and variable equivalent slope during the transitions.

The third waveform shows another important feature of the VHDL-MOS package: the capability to accurately simulate the power consumption of the MOS gates. The current through the power supply connection appears as a step-wise waveform also, because of the voltage discrete nature. But, as the current is of a continuous type, the width of the current step is not uniform.

Chapter 8

Perspectives

Contents

8.1 Libraries 102
8.2 Complex digital circuits 103
8.3 Mixed analog/digital circuits 106
8.4 Conclusions 108

The approach that we proposed so far in this thesis opens the perspective for efficient analysis of large systems that have components described at multiple levels of abstraction.

We will consider experimental results regarding the usage of the presented modeling and simulation approaches on a couple of submicron CMOS circuits.

The models that we used are the following:

1. for the static behavior of the deep submicron MOS transistor, the modified version of the collapsible current source model (see Section 4.4);

2. for the dynamic behavior of the deep submicron MOS transistor, the average capacitors model (see Section 6.2).

As a reference to compare our results with, we used the SPICE simulator [31], i.e., a circuit simulator with detailed models. We used the same generic circuit description to obtain first the SPICE description, and then the event-driven and

the piecewise linear ones, respectively. For the event-driven description we have used the VHDL language as proposed in the IEEE Std. 1076-1987 [25] and the VSYSTEM simulator [32]. For the piecewise linear description we have used the NDML language [33] and the PLATO simulator [34].

8.1 Libraries

We began our experiments by describing at circuit level a complete logic library, consisting of inverters, NAND gates, NOR gates, various types of latches and flip-flops, transmission gates, and tristatable buffers. For each element of this logic library we have created two new descriptions, apart from the SPICE description: one for the event-driven simulation using the VHDL-MOS package, and one for the piecewise linear simulation using the NDML language. This is presented in Figure 8.1. At this stage, after verifying the functionality of each element, we calibrated

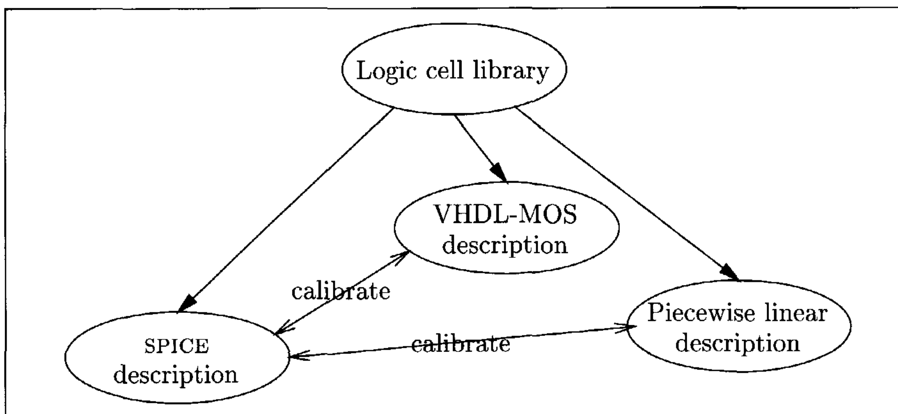


Figure 8.1: Various descriptions of each element of the logic library

the model parameters using the methodology that we devised and presented in Section 7.4. As a fit criteria we used the timing of the cells. We minimized the timing difference in comparison with the SPICE based models. Examples showing the difference between the waveforms obtained with our models and those obtained with the SPICE model are shown in Figure 7.5 and in Figure 8.2.

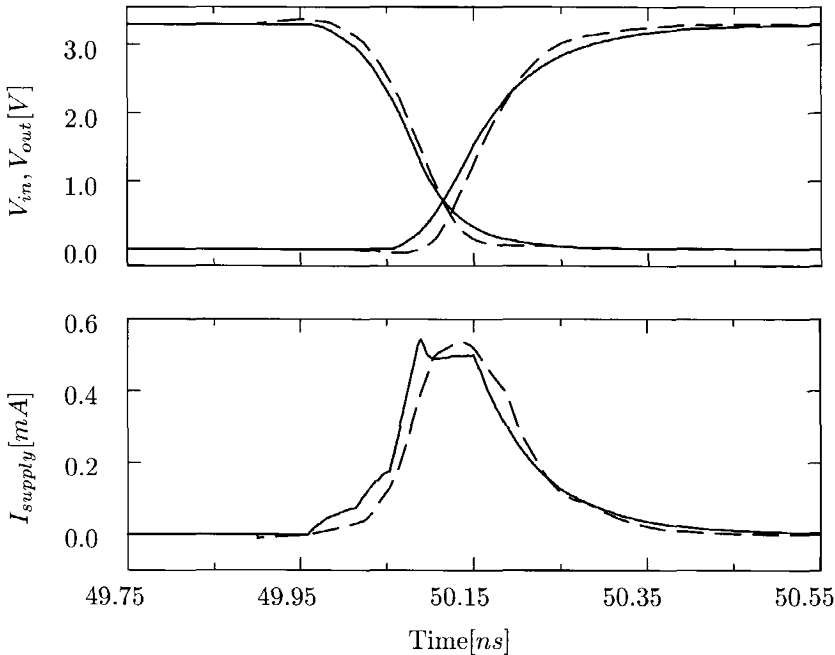


Figure 8.2: Simulated waveforms for a static CMOS inverter using the PLATO simulator; the SPICE reference appears in dashed line.

8.2 Complex digital circuits

To evaluate the performance of our modeling approaches we have built more complex test circuits based on this logic library. We translated each such test circuit to three descriptions corresponding to the three versions of the library. We performed the comparison in terms of timing and power consumption accuracy, and simulation efficiency (CPU run-time). The results of the former comparison are presented in Table 8.1 and Table 8.2, while those of the latter in Table 8.3 and Table 8.4.

We considered two classes of complex designs: adders and multipliers [35]. The `Csaxx` circuits are carry select adders on 4, 8, 16, and 32 bits. The `arrayxx` and `treexx` circuits are 4x4, 8x8, and 16x16 bits multipliers, with array and tree architectures. These tables present the accuracy of the delay and power estimations obtained with the event-driven and piecewise linear modeling approaches. As

	VHDL-MOS accuracy				PLATO accuracy	
	8 levels		28 levels			
Circuit	Delay	Power	Delay	Power	Delay	Power
Csa4	0.29%	1.21%	-2.98%	3.92%	0.54%	1.52%
Csa8	0.09%	1.36%	-2.63%	3.50%	0.23%	1.90%
Csa16	0.95%	1.74%	-1.93%	2.33%	0.49%	2.33%
Csa32	1.29%	1.39%	-1.36%	1.71%	0.33%	1.55%

Table 8.1: Delay and power estimation accuracy relative to SPICE for the carry select adder

	VHDL-MOS accuracy				PLATO accuracy	
	8 levels		28 levels			
Circuit	Delay	Power	Delay	Power	Delay	Power
Array4x4	7.85%	5.80%	4.51%	7.17%	3.42%	6.66%
Array8x8	10.12%	7.42%	5.73%	7.97%	4.39%	6.81%
Array16x16	10.69%	9.03%	5.85%	8.62%	4.45%	7.05%
Tree4x4	11.15%	5.38%	5.68%	7.25%	3.91%	6.86%
Tree8x8	9.17%	5.89%	6.67%	6.61%	4.38%	7.34%
Tree16x16	6.35%	7.77%	2.29%	7.59%	3.77%	6.80%

Table 8.2: Delay and power estimation accuracy relative to SPICE for various multiplier architectures

reference we used the estimations obtained with SPICE. We performed the delay measurements according to the 50/50 method and we considered the power consumption proportional with the time integral of the current through the voltage supply. We did the experiments with two different resolutions for the voltage variables: 8 and 28 levels. The simulation results show that the modeling approach that we took has a very good accuracy: the delay and power are estimated typically within a 5% error margin, independent of the size of the design.

Regarding the simulation efficiency we see that the transistor-level event-driven simulation approach is better, offering more than 10 times reduction in the CPU time compared to SPICE (at best 32 times). The speed-up factor increases with

the size of the circuit, as an event driven simulator takes advantage of the locality of the activity in a digital circuit. Furthermore, a reduction in the resolution of the VHDL-MOS package from 28 discrete voltage levels down to 8 brings a simulation speed-up factor of 3 to 4, as in the plot presented in Figure 8.3. The accuracy loss due to the reduction in the number of voltage levels is acceptable: less than 3%.

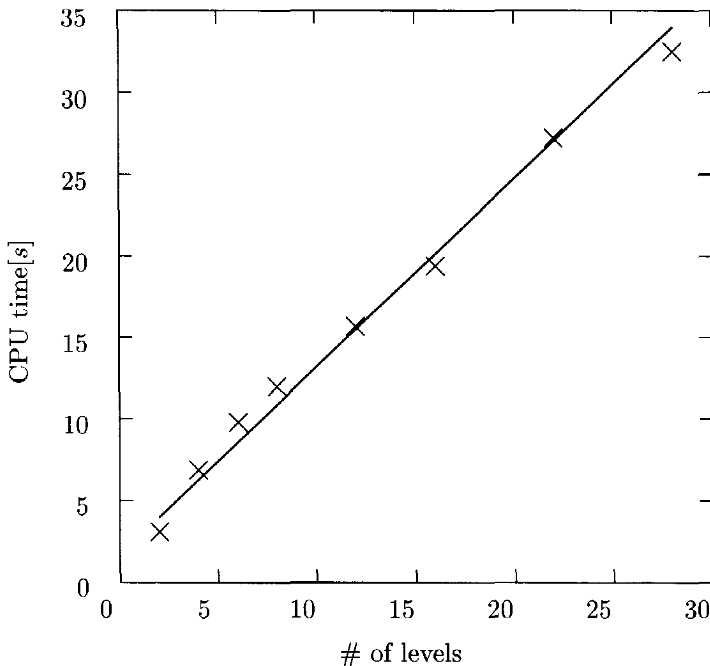


Figure 8.3: Influence of the number of discrete voltage levels in the VHDL-MOS package on the simulation time

A state-based piecewise linear simulator will be slower than the electrical level simulators on electrically modeled circuits and also slower than digital simulators on digitally modeled circuits [36].

	VHDL-MOS speed-up factor			
# of levels	Csa4	Csa8	Csa16	Csa32
28 levels	2.49	3.74	4.72	7.80
8 levels	8.13	8.47	12.60	25.44

Table 8.3: VHDL-MOS simulation speed-up factor relative to SPICE for the carry select adder

	VHDL-MOS speed-up factor					
	Array			Tree		
# of levels	4x4	8x8	16x16	4x4	8x8	16x16
28	4.07	7.76	14.49	3.78	4.35	13.02
8	10.17	21.28	31.95	11.44	16.88	29.93

Table 8.4: VHDL-MOS simulation speed-up factor relative to SPICE for various multiplier architectures

8.3 Mixed analog/digital circuits

In this subsection we will present the results of our experiments in the simulation of more analog-like structures using the VHDL-MOS package. We have chosen for a charge pump phase-lock loop [37] due to its good mix of digital and analog structures. An implementation of such a loop is presented in Appendix D. The operating principles of the charge pumps, the starved-current inverter ring oscillator, and the loop filter, which are true analog blocks, recommended it as a test case situated at the interface between digital and analog circuitry.

Our main goal was to observe the behavior of the loop, i.e., the output digital-like signal of the starved-current inverter voltage-controlled oscillator, in the two operation modes of the loop: acquisition and tracking.

We mapped each component block of the diagram presented in Figure D.1 on page 153 in terms of cells from the VHDL-MOS library. Then we simulated the loop using as simulation parameter the number of discrete voltage levels which defines the resolution of the VHDL-MOS package.

A remarkable difference noticeable in the output of the voltage-controlled oscillator in comparison with a SPICE simulation was in the tracking mode. A sample of the continuous time simulation is presented in Figure D.10 on page 159. This difference shows up as hops of the output frequency of the voltage-controlled oscillator between two discrete values. This is presented in Figure 8.4.

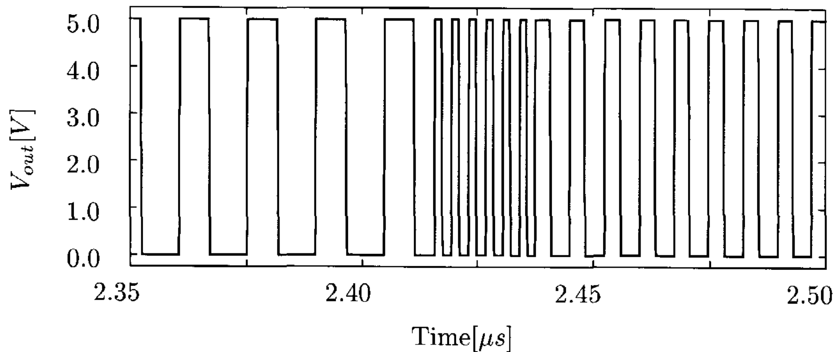


Figure 8.4: Output waveform of the voltage-controlled oscillator, as simulated with the VHDL-MOS package with 28 voltage levels

We explain this by the fact that in our modeling approach any node voltage is discretized on a finite number of levels. Hence, the control voltage of the voltage-controlled oscillator can only take values from a finite set, the one that we define when setting the resolution of the VHDL-MOS package (in the case presented in Figure 8.4 there is a resolution of 28 voltage levels). Accordingly, the output frequency of the voltage-controlled oscillator can take a finite number of discrete values.

Therefore, for any frequency of the input reference signal which is different from the discrete frequencies that could be generated via simulation by the voltage-controlled oscillator, the frequency of the output signal of the phase-lock loop hops between the two neighboring upper and lower discrete frequencies[1]. This is visible on the waveform in Figure 8.4 as the alternation of two different frequencies.

We can reduce the frequency error at the output of the simulated loop by

[1] When the loop is locked the phase difference should also be zero.

increasing the number of discrete voltage levels in the VHDL-MOS package. However, the price paid for this is a linear increase in the simulation time, as presented in Figure 8.3.

To simulate such analog behavior as in a phase-lock loop, the piecewise linear approach has been shown to be more suitable [38]. The reason lays in the continuous way of modeling the currents, as well as the voltages. □

Our experiments showed that our approach, i.e., event-driven simulation using the VHDL-MOS package, as well as the piecewise linear approach are suitable for simulating digital circuits with a good accuracy. The former showed a better performance in computation efficiency in comparison with the latter.

For large digital circuits we observed from the experiments that we carried out with the VHDL-MOS package that the run time improvement factor in comparison with a continuous-time SPICE simulation increases with the size of circuit. We can explain this through the fact that, due to the locality of signal propagation, only reduced portions of the circuit are activated for simulation. Furthermore, more analog behavior is encountered, slower the simulation runs, as more event scheduling has to take place.

Our approach turns out to capture very well the analog aspects of the large-signal digital operation of CMOS circuits. We could see that the errors introduced by it for modeling small-signal analog behavior cannot be always neglected. The accuracy in such a situation can be improved at an increased cost in the simulation time.

8.4 Conclusions

With this section we round up the suite consisting of exploration, modeling, simulation approach, calibration, and implementation, aiming at efficient transparent mixed-mode analysis of large deep submicron digital CMOS circuits.

⋆ ⋆ ⋆

With the research presented in this thesis we gave an answer to some of the challenges raised by the forthcoming deep submicron CMOS technologies. We were concerned with the modeling and analysis of large size, complex deep submicron circuits and focused on the circuit level aspects.

Our event-driven approach to deep submicron device modeling and analysis allows us to verify large circuits at a high level of accuracy, comparable with the one achieved in the classical transistor level modeling, but at considerably lower price in terms of computation. The simulation efficiency of our approach stems from local modeling combined with global signaling between the modeled entities. Yet, the most important contribution that our work brings to system design is the possibility to co-simulate, co-verify mixed circuits. The key is a transparent modeling approach between transistor level, gate level, or other higher levels of abstraction.

Approach We decided what framework has to be used for efficient multi-level simulation. We built up a large-signal compact, minimal model which captures the static as well as the dynamic behavior of the deep submicron MOS device.

We stated in Chapter 2 the need for level transparency when simulating multi-level deep submicron digital systems. Level transparency implies the usage of one single simulation algorithm for an improved simulation efficiency.

For the equation-based simulation approach we identified that the piecewise linearization improves the simulation efficiency. We found that the piecewise linear approach based on diode states satisfies the one algorithm demand for level transparency.

We addressed the dynamic simulation of large systems in Chapter 5. We analyzed what variant of the time discretization method is the most computationally efficient. In our view the event-driven simulation approach, based on local modeling and global signaling, is the most convenient framework for the multi-level simulation of large digital deep submicron CMOS circuits. It enables in a natural way the efficient transparent simulation of systems with components described at various levels of abstraction.

We analyzed the implications of scaling-down the MOS transistor length on the DC characteristic in Chapter 3. We showed that the most impact of scaling-down the channel length on the MOS transistor behavior is to be attributed to the drift velocity saturation effect. We proposed a compact model for the transport current of a MOS transistor operating in large-signal digital conditions.

We investigated the effects of velocity saturation on the static behavior of the MOS transistor. When the length comes down in the deep submicron range, the drain saturation current becomes rather linear with the gate-to-source voltage and the saturation current increases while the saturation voltage decreases. In the control-voltage plane there is loss in the extent of the triode region for gain in the extent of the saturation region. For a deep submicron MOS transistor the saturation of the drain current owes to the pinch-off mechanism, inherited from the long channel operation, and to the drift velocity saturation mechanism, specific for the short channel operation. For a very-short length transistor, the current saturation owes mostly to the velocity saturation and the triode region may be eventually disregarded.

We concluded Chapter 3 with the proposal of the Collapsible Current Source Model as the ultimate model for a deep submicron MOS transistor. This is a large-signal piecewise linear, compact model and thus suited for the simulation of large digital circuits.

We explored the modeling of the large-signal dynamic operation of digital CMOS circuits in Chapter 6. Here we proposed a compact model and then proved its analytical soundness. We modeled the large-signal terminal charge variations with average capacitors, assuming quasi-static operation.

We observed that the charges associated with the MOS transistor terminals have a non-linear dependency on the potentials on all four terminals. Therefore, to build a compact dynamic model for the MOS transistor we proposed to use average capacitors.

The charge accumulation inside the MOS transistor structure is associated with the charge currents. We proofed that the modeling of charge currents with large-signal average capacitors is an analytically sound method in what concerns the power consumption and waveform estimations for a digital CMOS circuit. Finally, we described a practical method to calculate the average capacitances starting from the RF-parameters of the MOS transistor.

Implementation Based on the proposed modeling concepts we then proceeded to the implementation of the compact global model of the deep submicron MOS transistor. We showed in Chapter 4 how the implementation was actually done in the piecewise linear framework. We presented in Chapter 7 our original way to implement circuit level models inside an event-driven simulator. We assessed then in Chapter 8 the accuracy and the efficiency of these implementations.

Firstly we have shown how the static model translates into the piecewise linear equations. Based on a geometrical interpretation of the input-output space partition in operating regions, we have solved for the coefficients of the piecewise linear equations.

Secondly, we developed a special package named VHDL-MOS, to model the interaction of the MOS transistor with other circuit-level components. The core idea was to declare dedicated signal types to mimic the analog voltages and currents. In this way we managed to integrate models capturing analog aspects of CMOS circuit operation in an event-driven simulator, originally meant for digital circuits.

We also presented a methodology to calibrate the composed static and dynamic model for the purpose of accurate power and waveform estimation.

At last we compared the two simulation approaches on a couple of test designs, which we considered representative. Both approaches appeared to have a good accuracy in estimating timing and power. The simulation approach based on the VHDL-MOS package turned out to be computationally more efficient. Furthermore, this package is meant to capture the *analog aspects of the large-signal digital* operation of CMOS circuits. Its accuracy when simulating small signal analog circuits is limited, unless traded off for CPU time. Still, the most important advantage of the VHDL-MOS package approach is the following: it opens the way to system-on-a-chip simulation, i.e., digital, gate or higher level, descriptions can coexist with analog-like transistor-level descriptions inside the same event-driven simulation kernel.

Chapter 9

A deep submicron vanishing point

By down-scaling the dimensions of the MOS transistors the engineers are aiming to integrate more functionality on the same chip area and at the same time improve the performance.

An important factor which contributed to the performance improvement along many technology generations is the current drive $I_{D,SAT}/W$. As the current drive improved with each technology generation, so did the performance (see expression (1.1) for the time delay).

As the drift velocity approaches the saturation velocity, its field dependence on the electric field will begin to depart from the linear relationship specific for the low-field situation. A semi-empirical formula for the drift velocity of the charge carriers is proposed in [39]:

$$v_d = \frac{v_{sat}}{\left[1 + \left(\frac{E_c}{E}\right)^{\eta}\right]^{1/\eta}} \tag{9.1}$$

The saturation velocity v_{sat} can be considered in a first approximation the same both for the holes and for the electrons. E_c is the critical electrical field and the

coefficient η varies with the type of charge carriers: for holes close to 1, while for electrons close to 2.

Based on this formula we are going to derive a general linear dependency between the drain saturation current and the drain-source saturation voltage of an MOS transistor, dependency valid for all transistor length L and for both p- and n-transistor types.

We recall here some starting point relations: (B.19) for the drain current,

$$\mathbf{I_D} = -W|Q_I|\mathbf{v_d} \tag{B.19}$$

and (B.22) for the inversion layer charge:

$$Q_I(y) = -C_{ox}\big(V_{GS} - V_T(V_{SB}) - (1+\delta)V_{CS}(y)\big) \tag{B.22}$$

For the drift velocity we use (9.1) with $E = \frac{dV_{CS}}{dy}$. The contact-to-source bias $V_{CS}(y)$ at an arbitrary point $C(y)$ in the channel is a monotonically increasing function of y. The solution at the two ends of the channel satisfies the boundary conditions: $V_{CS}(0) = 0$ and $V_{CS}(L) = V_{DS}$. By substituting (B.22) and (9.1) into (B.19) we obtain the following expression for the drain current in triode region:

$$I_D = WC_{ox}\big(V_{GS} - V_T(V_{SB}) - (1+\delta)V_{CS}(y)\big)\frac{v_{sat}}{\left[1+\left(\frac{E_c}{E}\right)^{\eta}\right]^{1/\eta}} \tag{9.2a}$$

$$= WC_{ox}(1+\delta)v_{sat}\left[\frac{V_{GS} - V_T(V_{SB})}{1+\delta} - V_{CS}(y)\right]\frac{E}{\left(E^{\eta} + E_c^{\eta}\right)^{1/\eta}} \tag{9.2b}$$

For notational simplicity let us introduce the following notations: $a = WC_{ox}(1+\delta)v_{sat}$, $b = \frac{V_{GS}-V_T}{1+\delta}$, $c = E_c$, $I = I_D$, $u = V_{CS}(y)$, and $V = V_{DS}$. As comes out from (9.2b) the drain current can be written as:

$$I = a(b-u)\frac{du/dy}{\left[(du/dy)^{\eta} + c^{\eta}\right]^{1/\eta}} \tag{9.3}$$

The above expression can be rewritten in the following form:

$$Ic\,dy = \left[a^{\eta}(b-u)^{\eta} - I^{\eta}\right]^{1/\eta} du \tag{9.4}$$

which would allow us to find an implicit relation between the drain current (I) and the drain-to-source voltage (V) in triode region by integrating (9.4) over the channel length:

$$\int_0^V \left[a^{\eta}(b-u)^{\eta} - I^{\eta}\right]^{1/\eta} du = cLI \tag{9.5}$$

When the transistor switches from triode to saturation regime, the first derivative of the drain current with respect to V_{DS} equals to zero, i.e., $\frac{\partial I}{\partial V} = 0$. Let us introduce additional notations:

$$f(u,I) = \left[a^{\eta}(b-u)^{\eta} - I^{\eta}\right]^{1/\eta} \qquad \alpha = cL \qquad F(V,I) = \int_0^V f(u,I)du \tag{9.6}$$

Relation (9.5) can be rewritten as:

$$F(V,I) = \alpha I \tag{9.7}$$

Note that $F(V,I)$ depends on V only, since I is uniquely determined by V: $I = I(V)$. If we now differentiate both sides of (9.7) we get:

$$\frac{\partial F}{\partial V} + \frac{\partial F}{\partial I} \cdot \frac{\partial I}{\partial V} = \alpha \frac{\partial I}{\partial V} \longrightarrow \frac{\partial I}{\partial V} = \frac{\partial F/\partial V}{\alpha - \partial F/\partial I} \tag{9.8}$$

We are looking for the curve Γ in the I-V plane such that $\frac{\partial I}{\partial V} = 0$ for any $(I,V) \in \Gamma$. From (9.8) it comes out that $\frac{\partial I}{\partial V}(I,V) = 0$ if and only if $\frac{\partial F}{\partial V}(I,V) = 0$ on Γ. As follows from the definition (9.6) of $F(V,I)$ we have

$$\frac{\partial F}{\partial V} = \frac{\partial}{\partial V} \int_0^V f(u,I)du = f(V,I) \tag{9.9}$$

Thus, the curve Γ is given by $f(V,I) = 0$. This means that

$$I = a(b - V) \tag{9.10}$$

on Γ, which follows from (9.6).

We found that for the general case the triode-saturation separation is given by the linear relation:

$$I = WC_{ox}v_{sat}[V_{GS} - V_T - (1+\delta)V] \quad (9.11)$$

Expression (9.11) can be seen as the separation between triode and saturation regions in the I-V plane.

The drain current in triode region is the implicit solution of equation (9.5). For a p-device the charge carriers in the channel are holes, and, as mentioned before the η coefficient takes values close to 1. In that case an explicit expression for the drain current is easily derived. For $\eta \in (1, 2]$ it leads to β-functions and it is better to use numerical software [40].

In Figure 9.1 we show for the p-transistor case the linear relation (9.11) between the saturation current and the saturation voltage together with the triode part of the I-V characteristic families for various technology generations. We observe that:

- for the long ($5\mu m$) channel technologies the saturation voltage $V_{DS,SAT}$ is bounded above by $\frac{V_{GS} - V_T}{1+\delta}$, while
- for the very-short ($0.05\mu m$) channel technologies the saturation current per unit width, or current drive, is bounded above by $v_{sat}C_{ox}(V_{GS} - V_T)$.

This maximum achievable current from a transistor is not directly dependent on the channel length L. We also remark that, in the quest for higher speed through the relative increase of the current drive by down-scaling of the transistor length, there is an inherent limitation.

We show in Figure 9.2 the situation for an n-transistor, i.e., $\eta = 2$. We additionally observe that, for a given technology generation, the current drive of an n-transistor comes closer to the $v_{sat}C_{ox}(V_{GS} - V_T)$ upper boundary than the current drive of a p-transistor.

We showed that due to the velocity saturation effect the current drive does not anymore improve significantly by scaling the transistor dimensions below a certain limit. Not only that the current drive improvement saturates, but also the capacitive load that a gate has to drive increases relative to the gate strength (as

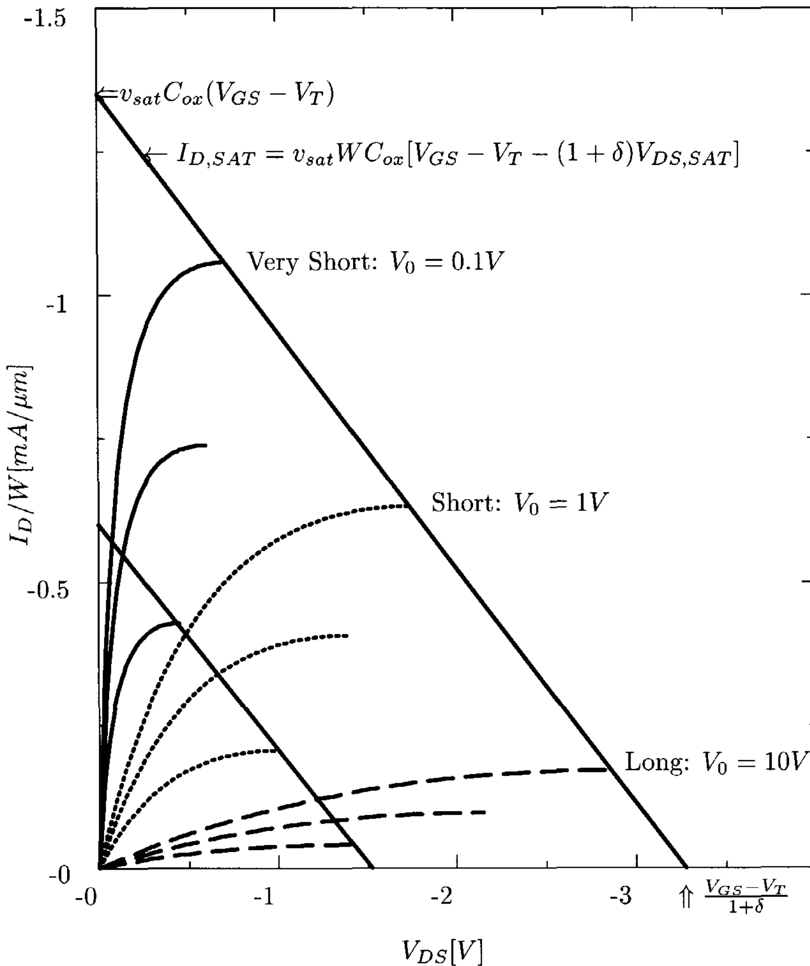


Figure 9.1: I-V characteristic families and the triode/saturation separation for various technology generations for the p-transistor case

another detrimental effect of the interconnect lateral capacitance). In an attempt to keep improving the performance with the down-scaling, one may scale the transistor width at a slower pace than its length, but this would affect adversely the transistor density on the chip.

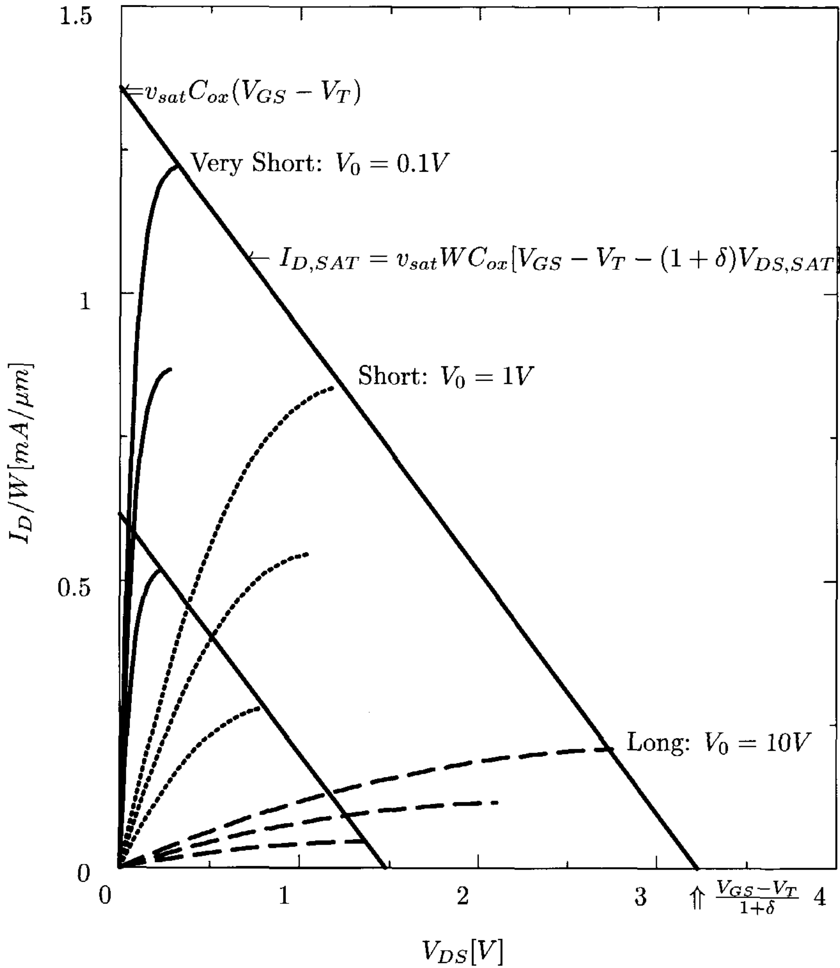


Figure 9.2: I-V characteristic families and the triode/saturation separation for various technology generations for the n-transistor case

It is in the sense of this discussion that we predict a "deep submicron vanishing point" on the technology road-map.

Appendix A

The linear complementarity problem

Contents

A.1 Path followers 120
A.2 Complementary pivoting 122
A.3 PLATO 124

The *linear complementarity problem* is, for a given $p \times p$-matrix $\mathbf{M}$ and vector $\mathbf{q}$ with p components, to find vectors $\mathbf{w}$ and $\mathbf{z}$ such that

$$\mathbf{w} - \mathbf{M}\mathbf{z} = \mathbf{q} \tag{A.1a}$$

$$\mathbf{w} \geq \mathbf{0},\ \mathbf{z} \geq \mathbf{0} \tag{A.1b}$$

$$\mathbf{w}^{\mathbf{T}} \cdot \mathbf{z} = \mathbf{0} \tag{A.1c}$$

if such a solution exists. A solution is $(\mathbf{w}, \mathbf{z})$ is called a *complementary feasible solution*, and it is a *complementary basic feasible solution* if one variable of each *complementary pair* (w_j, z_j) is in the basis.

The problem is $\mathcal{NP}$-complete in the strong sense [1], so that we cannot expect

[1] Proven apparently by S.J.Chung in 1979 "A note on the complexity of LCP: the LCP is NP-complete" Report No 79-2, Department of Industrial and Operations Engineering, University of Michigan, Ann Arbor,Mi, U.S.A.

a polynomial-time algorithm solving the problem for all $\mathbf{M}, \mathbf{q}$. Many methods from mathematical programming, however, can be applied. For the problem can be cast in a minimization problem with a concave objective function, into a zero-one bilinear programming problem and of course into linear integer programming problems. The latter opens up several solving techniques when $\mathbf{M}$ possesses particular properties. When $\mathbf{M}$ is positive semidefinite, it can be solved in polynomial time, but in the piecewise linear approach to simulation circuits almost never produce matrices from such a restricted class.

Simulators do not use any of the above formulations or techniques. They use almost invariably a simplex type of pivoting methods for solving linear complementarity problems. Since they follow a path from simplex to neighboring simplex they are called *path followers* [36]. Under certain non-degeneracy assumptions these algorithms are guaranteed to find a complementary feasible solution for matrices $\mathbf{M}$ that satisfy certain properties. In practice, they are known to work well even when these assumptions are violated.

A.1 Path followers

Path followers modify the set defined in A.1 by introducing an artificial parameter a fixed positive vector $\mathbf{e}$ and a scalar λ:

$$\mathbf{w} - \mathbf{M}\mathbf{z} - \lambda\mathbf{e} = \mathbf{q} \tag{A.2a}$$

$$\mathbf{w} \geq \mathbf{0},\ \mathbf{z} \geq \mathbf{0},\ \lambda \geq \mathbf{0} \tag{A.2b}$$

$$\mathbf{w}^{\mathbf{T}} \cdot \mathbf{z} = \mathbf{0} \tag{A.2c}$$

that initially gets the value $\max_i\{-q_i\}$ which makes $\mathbf{z} = \mathbf{0},\ \mathbf{w} = \mathbf{q} + \lambda\mathbf{e}$ a starting solution[2]. If $\mathbf{q}$ is nonnegative we immediately have a solution with $\lambda = 0$. Otherwise we try to drive λ to zero through a sequence of pivots while satisfying A.2 all the time. If successful we obtained a solution to A.1.

Path followers find at most one solution, and in a sense mostly one close to the simplex it started from. This is a desirable property, because we are mostly

[2] Although there are some efficiency arguments to make a smart choice for the vector $\mathbf{e} \geq 0$, in proofs $\mathbf{e}$ is assumed to be the vector of all ones.

interested in only that solution since perturbations from time-step to time-step are not expected to be very big.

Other facts that make path followers the preferred solvers in piecewise linear simulators are that

1. for a large class of matrices they either find a solution or indicate non-existence[3]
2. the pivoting is explicit and explainable in terms of circuit behavior,
3. although they might in theory perform a large part of the exponentially many pivot possibility, they usually finish in a very small number of steps.

Conceptually the simplest algorithm [41] works with only positive diagonal pivots. An interpretation of the algorithm is that it changes diode states one by one. It tries to do this in such a way that λ is decreased. λ is only so far decreased until a variable in the basis becomes zero. This means that a boundary of a simplex has been encountered. In order to decrease λ further a positive diagonal element should be found. In that case a pivot on that element is performed, and the path is continued. If λ reaches zero, a solution has been found. If λ is not zero and no positive diagonal pivot can be found the algorithm aborts. This does not happen when the matrix $\mathbf{M}$ has only positive principal minors: in that case this simple algorithm always finds the unique solution. For other matrices it may abort also when there is a solution. Several attempts to modify this *diagonal pivoting* algorithm so that negative pivots can somehow be used to allow λ to increase temporarily, extends its potential, but insufficiently to be practical for circuit simulation.

Another straightforward idea with the diode state model of Chapter 2 in mind is to extend the linear multiport of Figure 2.2b with ideal diodes one by one. During the process we try to keep the components of the initially 0-vector $\mathbf{Cx} + \mathbf{g}$ nonnegative to solve (2.13b) and (2.13c) for a given input vector $\mathbf{x}$. This is in essence equivalent to a path follower known as *principal pivoting* an algorithm due to Cottle [42]. It is guaranteed to find the solution when all principal minors are positive.

The algorithm of the next section, called *complementary pivoting* and originally due to Lemke [43], serves our purposes better and will therefore be described in more detail.

[3] In jargon: "they process the linear complementarity problem for that class".

A.2 Complementary pivoting

The best known path follower for linear complementarity problem is an algorithm due to Lemke, successively streamlined and improved into the following steps:

0. `if` $\mathbf{q} \geq \mathbf{0}$ `then`
   - $(\mathbf{w}, \mathbf{z}) = (\mathbf{q}, \mathbf{0})$ is a complementary basic feasible solution,
   - `stop`

   `else`,
   - organize the equation set A.2 in a tableau, columns: $\mathbf{w}, \mathbf{z}, \lambda$
   - let $-q_f := \max_i\{-q_i\}$
   - pivot at row f and column λ
   - let $y_f := z_f$
   - `goto` 1

1. Let $\mathbf{c}_f$ be the updated column under y_f

   `if` $\mathbf{c}_f \leq \mathbf{0}$ `then goto` 4.

   `else`, $\tilde{\mathbf{q}}$ is the updated right-hand side,

   i.e., the values of the variables in the basis
   - let $\frac{\tilde{q}_r}{c_{rf}} := \min_i \left\{ \frac{\tilde{q}_i}{c_{if}} | c_{if} > 0 \right\}$
   - `if` the basic variable at row r is λ `then goto` 3

     `else goto` 2

2. the basic variable at row r is w_l or z_l, $l \neq f$.
   - pivot at row r and the column of y_f
   - `if` z_l left the basis `then` $y_f := w_l$

     `else` $y_f := z_l$.

   return to 1

3.
   - pivot at row r and the column of y_f
   - return the complementary basic feasible solution
   - `stop`

4. *ray termination*: `stop`

When the algorithm ends in 4 there is no solution to the set (A.1) and it finds a *ray* $\{(\mathbf{w}, \mathbf{z}, \lambda) - \gamma\mathbf{c} | \gamma \geq 0\}$ of which all elements satisfy (A.2). Here is $(\mathbf{w}, \mathbf{z}, \lambda)$ an *almost* complementary basic feasible solution associated with the last tableau, and

$\mathbf{c}$ a vector of zeros, but -1 in the row of y_f and $\mathbf{c}_f$ in the rows of variables in the basis.

As stated before, *complementarity pivoting* cannot successfully handle every linear complementarity problem, but it is to date the method that can process the widest class of matrices $\mathbf{M}$. This class has been proven to include:

1. matrices of which the principal minors are positive [42],

   the algorithm will find the unique solution.

2. matrices that are copositive-plus[4] [43],

   the algorithm finds a complementary feasible solution if it exists;
   if no solution exists, the algorithm terminates in ray termination.

3. matrices that are the sum of a symmetric copositive-plus matrix $\mathbf{P}$ and a copositive matrix $\mathbf{C}$ [44]:

   if $\mathbf{q} + \mathbf{Px} - \mathbf{C^T y} \geq \mathbf{0},\ \mathbf{y} \geq \mathbf{0}$ has a solution $\mathbf{x} \in \mathbb{R}^\mathbf{n},\ \mathbf{y} \in \mathbb{R}^\mathbf{n}_+$
   the algorithm will terminate with a complementary basic feasible solution

Unfortunately it is not possible to characterize all matrices that arise in piecewise linear simulation, in such a way that conclusive statements can be made about the existence of solutions. Neither can be said for which class exactly complementarity pivoting processes the problem. The above statements are in fact only sufficient conditions. Assuming that only such matrices should be given to a complementary pivoting algorithm is overly pessimistic, though in the thesis, Chapter 4, we show that straightforward diagonal pivoting does not work for simple networks with collapsible current sources. We also experienced that a well-researched simulator such as PLANET [38] cannot handle pure collapsible current sources[5] It is therefore recommended to use a path following type of linear complementarity problem solver that can handle an as wide as possible class of matrices, and that is to the best of our knowledge the above complementary pivoting algorithm, primarily based on the work of Lemke, or one of its variants.

[4] A matrix $\mathbf{M}$ is *copositive* if $\mathbf{x^T M x} \geq \mathbf{0}$ whenever $\mathbf{x} \geq \mathbf{0}$. It is said to be *copositive-plus* if in addition $\mathbf{x} \geq \mathbf{0}$ and $\mathbf{x^T M x} = \mathbf{0}$ implies that $(\mathbf{M^T} + \mathbf{M})\mathbf{x0}$.

[5] PLANET uses a simpler algorithm as its core solver for resistive networks, a modified version [45] of the early result of Katzenelson [41].

A.3 PLATO

In our experiments we have chosen for PLATO [34], a simulator based on piecewise linear modeling. Its salient features are:

1. a variant of complementary pivoting to solve resistive networks,
2. multi-rate integration techniques to exploit the circuit's latency,
3. sparse matrix update techniques for efficiency.

The complementary pivoting algorithm is a variant of the one in the previous section and is due Van de Panne [46]. It solves the same class of linear complementarity problems, but it uses only block pivots, where a block is a principal submatrix of $\mathbf{M}$. The implementation in PLATO is such that either a block pivot is performed with pivots local to a leaf cell in the stored hierarchy, or all pivots are on the diagonal of separate leaf cells. The block matrix structure, reflecting the original hierarchy is thus preserved [47].

The use of multi-rate integration can potentially save a lot of computation time, because the computational effort can be restricted to the active parts of the circuit. Latent parts can be left alone or integrated with large time steps.

Sparse updating schemes prevent solving large linear systems as happens in more conventional direct methods.

It shares with other simulators based on piecewise linear modeling the excellent global convergence properties and the level transparency which inherent to the uniform modeling concept and the flexible addition of macro models.

Appendix B

Long channel static operation

Contents

B.1 Transistor structure 125
B.2 Capacitors 127
B.3 Diodes 134
B.4 Transistors 135
B.5 Remarks 143

B.1 Transistor structure

The basic structure of an n-channel MOS transistor is shown in Figure B.1. From up to down we may identify: the gate (G) metalization and terminal, the gate oxide layer of thickness t_{ox}, the p-type bulk[1] semiconductor, the drain and source n^+-type diffusions, the drain (D) and source (S) metalizations and terminals, and the bulk (B) metalization and terminal. The voltage sources V_{SB}, V_{GS}, and V_{DS} shown with their polarities provide on the four terminals of the structure the necessary potentials for normal operation. B is the reference terminal, while G is the control terminal.

[1] As usually, the terms "bulk", "substrate", and "body" are used with the same meaning.

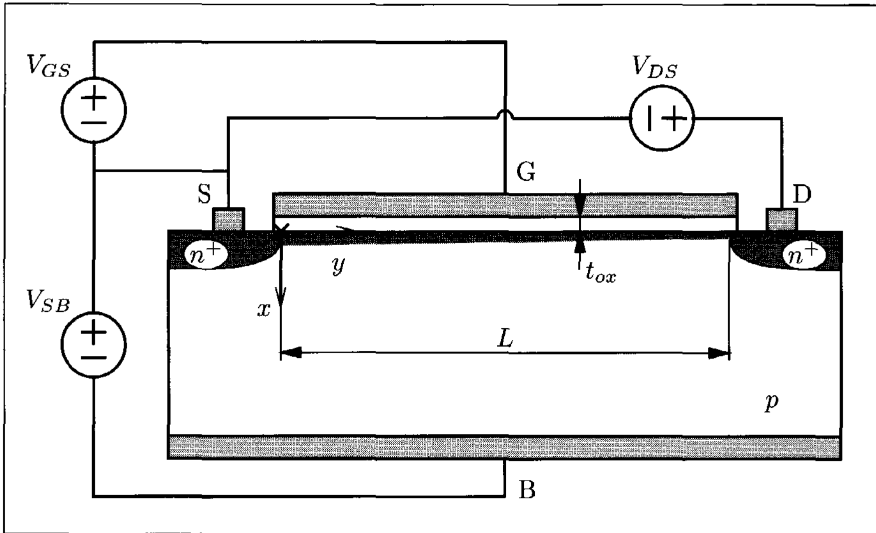


Figure B.1: Longitudinal cross section through an n-MOS transistor structure

In general, for a perfect structure and for a fixed V_{SB}, the operation is as follows. For zero V_{GB} there is no conducting channel between the drain and the source diffusions; the structure is equivalent with two pn junctions connected back-to-back. For positive V_{GB} values electrons will be attracted at the bulk surface and holes will be driven away from the surface deep into the bulk. For sufficiently high V_{GB} there will be an electron channel at the semiconductor surface, which makes possible the electric conduction between drain and source. It is customarily to consider that the channel comes into existence for V_{GS} values above a threshold V_T. The external V_{GB} controls the existence of a surface conducting n-channel between the drain and the source diffusions, and influences the channel properties as soon as this exists. A flow I_D of electrons from source to drain may be observed when the potential difference V_{DS} is applied.

The length L of the channel has an important impact on its electrical behavior. While for a long MOS transistor the channel may be modeled as a resistor, for a deep submicron length MOS transistor the channel may be assimilated with a current source. In the following section we will review the classical theoretical

approach for the static operation of the long channel MOS transistor.

Before proceeding to the study of the fairly complex four-terminal MOS transistor structure, we will examine the simpler two- and three-terminal structures. Basic operation principles are recognized for these simpler structures, and then they are combined to obtain the global four-terminal operation [4, 16, 17, 48]. The surface potential and inversion layer charge notions are introduced with the two-terminal structure, and the so called "body effect" is recognized through the study of the three-terminal structure. After these preliminaries, a closed form expression for the current through the four-terminal structure is deduced.

B.2 Capacitors

The two-terminal MOS structure (also called the MOS capacitor) is presented in Figure B.2; it consists of the gate metalization, the gate oxide, the substrate semiconductor, and the substrate metalization. ψ_{ox} denotes the voltage drop across the

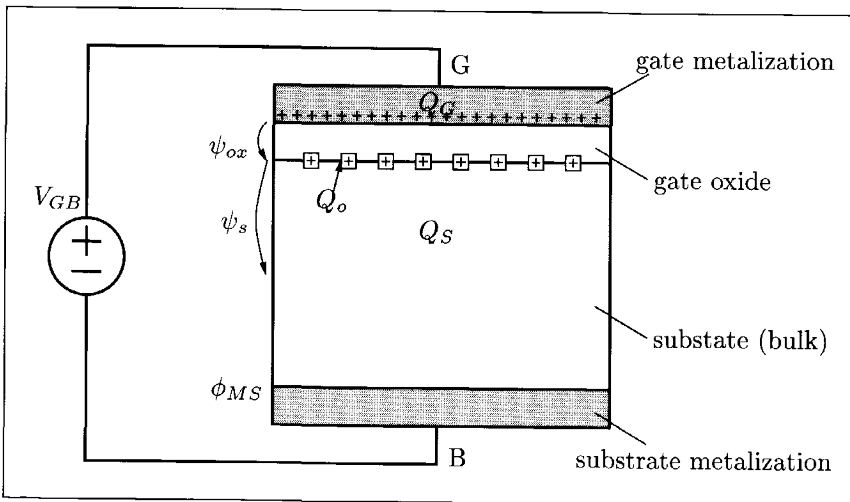


Figure B.2: Cross section through the two-terminal MOS structure

oxide and ψ_s is the voltage drop across the semiconductor surface.

Two "parasitic" effects will produce a voltage drop across the semiconductor

even when the externally applied V_{GB} is zero: (1) Because of technological imperfections, electric charge is trapped in the oxide and at the oxide-semiconductor interface. An effective interface charge, denoted by Q_o and positive in most situations, is assumed to be located at the oxide-semiconductor interface. Q_o may be assimilated with a preexisting voltage-drop across the oxide of value $-Q_o/C_{ox}$. (2) Due to different Fermi levels in the semiconductor and in the bulk metalization, a contact potential ϕ_{MS} will develop at the substrate contact. To bring back the semiconductor into neutrality, V_{GB} must be assigned the *flat-band voltage* expressed by:

$$V_{FB} = \phi_{MS} - \frac{Q_o}{C_{ox}} \tag{B.1}$$

For an arbitrary V_{GB} the potential balance across the two-terminal structure is:

$$V_{GB} = \psi_{ox} + \psi_s + \phi_{MS} \tag{B.2}$$

The voltages applied to this capacitor-like structure determine the accumulation of electric charges on both sides of the gate oxide. Q_G is the charge on the gate metalization and Q_S is the charge in the semiconductor under the oxide; together with Q_o these charges must add up to zero for overall charge neutrality:

$$Q_G + Q_o + Q_S = 0 \tag{B.3}$$

We will study how the substrate condition is influenced by the external gate-to-bulk voltage V_{GB}. Firstly a qualitative discussion is made, followed by quantitative expressions for the interest variables.

For V_{GB} equal to V_{FB} we have the *flat-band condition* as discussed above. $\psi_s = 0$ and no excess charge appears at the surface of the semiconductor, $Q_S = 0$. Consider V_{GB} values below V_{FB}: the negative change in V_{GB} determines a negative change in ψ_s. $\psi_s < 0$ and holes will accumulate at the surface of the semiconductor, $Q_S > 0$. The semiconductor surface is said to be in *accumulation.* Assume V_{GB} values above V_{FB}: ψ_s will have a positive change relative to the flat-band situation, $\psi_s > 0$ and holes will be driven away from the semiconductor surface, $Q_S < 0$. Depending on the ψ_s magnitude, the source of the negative charge at the semiconductor surface is different. (Keep in mind that in the neutral p-type substrate the

holes are the majority charge carriers, with reference to the intrinsic type of semiconductor where both electrons and holes are present in equal concentrations.) For V_{GB} not much above V_{FB}, at the surface the hole concentration will be still higher than the electron concentration; the semiconductor surface is said to be in *depletion* and the negative charge Q_S is mainly due to the uncovered ionized acceptor atoms with the concentration N_A.

The depletion region will reach an end for a sufficiently high V_{GB} when the surface hole concentration will become equal to the surface electron concentration; this marks the beginning of the *inversion* condition. For even higher V_{GB} the electron concentration at the surface will reach the impurity concentration N_A; the thin semiconductor sheet at the surface will behave like an n-type semiconductor with an equivalent impurity concentration $N_D^{eq} = N_A$. This marks the on-set of the *strong inversion*, with the corresponding situation captured in Figure B.3. In

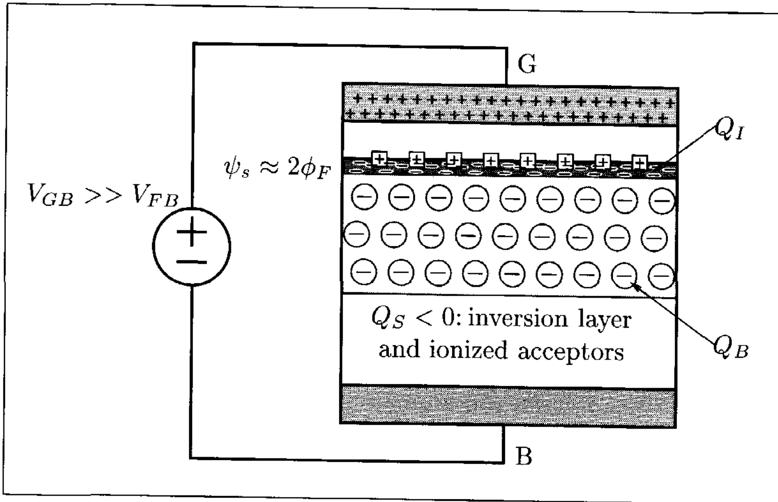


Figure B.3: Strong inversion condition at the substrate surface

inversion, the mobile inversion layer charge Q_I adds to the fixed depletion region charge Q_B:

$$Q_S = Q_I + Q_B \tag{B.4}$$

From a quantitative point of view, it is important to delimit in terms of ψ_s the on-set of the strong inversion condition at the substrate surface. We make use of the energy bands representation [49, 19] for the electrons in the semiconductor crystal, shown in Figure B.4a. In the intrinsic semiconductor crystal the electron

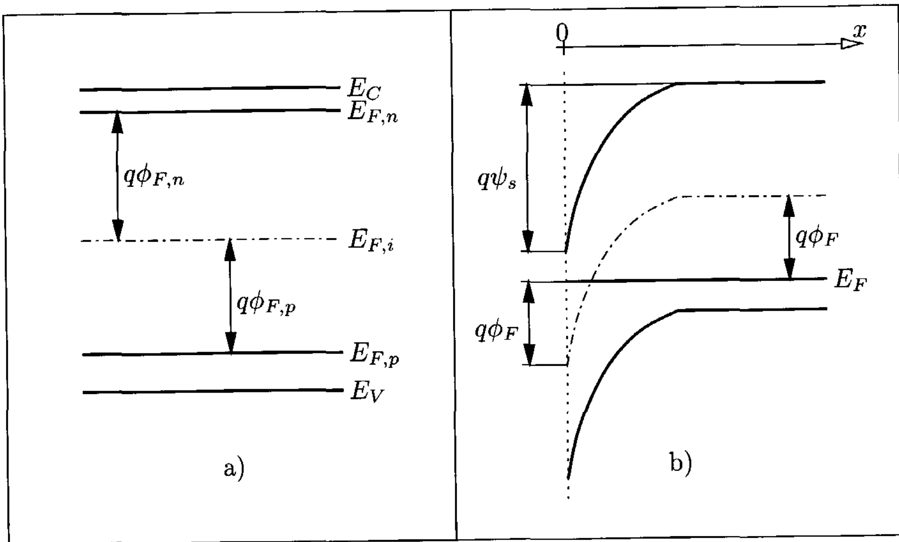


Figure B.4: Energy bands representation for the semiconductor crystal a) The position of the various energetic levels b) Energy bands bending at the p-type semiconductor surface due to the applied surface potential $\psi_s = 2\phi_F$

energy is confined inside energy bands. The last two populated bands are the valence band and the conduction band, which are separated by the so called band-gap. E_V designates the upper limit of the valence band, E_C the lower limit of the conduction band, and $E_{F,i}$ the middle of the band-gap. $E_{F,i}$ is called the intrinsic Fermi level. In the extrinsic p-type semiconductor the acceptor impurities give an energetic level E_A inside the band-gap, below $E_{F,i}$ and close to E_V. Conversely, in the extrinsic n-type semiconductor the donor impurities give an energetic level E_D above $E_{F,i}$ and close to E_C. The impurity energetic level gives the Fermi level for the extrinsic semiconductor.

If the surface potential is positive (with the neutral bulk as reference), then

the energetic levels at the surface will lower from the corresponding ones in the neutral bulk. Thus, for a positive ψ_s the bands structure will bend downwards, as shown in Figure B.4b; the amount of bending is proportional to the applied surface potential. One may observe that for $\psi_s = \phi_F$ the Fermi level at the surface is in the middle of the band-gap, situation equivalent with the intrinsic semiconductor; this marks the beginning of the inversion condition. For $\psi_s = 2\phi_F$ the Fermi level at the surface reaches the position that it would have in an n-type semiconductor with $N_D = N_A$ (i.e., with a $q\phi_F$ quantity above $E_{F,i}$); this defines the on-set of the strong-inversion condition.

We will approximate the inversion layer charge and the depletion region charge, as functions of the surface potential. The electron concentration at the surface $n_{surface}$ can be related to the surface potential ψ_s, and to the intrinsic concentration n_i or the impurity concentration N_A:

$$n_{surface} = n_i e^{(\psi_s - \phi_F)/\phi_t} \tag{B.5a}$$

$$\approx N_A e^{(\psi_s - 2\phi_F)/\phi_t} \tag{B.5b}$$

By solving the Poisson's equation for the substrate, a general closed form expression for Q_S is found:

$$Q_S = -(F\sqrt{N_A})\sqrt{\phi_t e^{-\psi_s/\phi_t} + \psi_s - \phi_t + e^{-2\phi_F/\phi_t}(\phi_t e^{\psi_s/\phi_t} - \psi_s - \phi_t)}$$

In inversion where $\psi_s > \phi_F$ the above equation is approximated by:

$$Q_S = -(F\sqrt{N_A})\sqrt{\psi_s + \phi_t e^{(\psi_s - 2\phi_F)/\phi_t}} \qquad \text{where} \quad F = \sqrt{2q\epsilon_{Si}} \tag{B.6}$$

The inversion charge Q_I is located in a very shallow layer at the semiconductor surface, where the inversion condition is met. The thickness of the inversion layer is very small when compared with the depth of the depletion region. A widely used approximation called the *charge sheet* approximation is then possible: the inversion layer has a negligible thickness and all the inversion charge is concentrated in a surface sheet[2]. The theory of abrupt n^+p junctions can be used here; the surface potential ψ_s plays the role of the junction bias. Q_B, the depletion region charge,

[2]The charges are reported as charge density per unit area; the capacitances are reported as capacitance per unit area.

can be approximated by:

$$Q_B = -F\sqrt{N_A}\sqrt{\psi_s} \tag{B.7}$$

Combining (B.4) with (B.6) and (B.7), an approximate closed form for the inversion layer charge is obtained:

$$Q_I = -F\sqrt{N_A}(\sqrt{\psi_s + \phi_t e^{(\psi_s - 2\phi_F)/\phi_t}} - \sqrt{\psi_s}) \tag{B.8}$$

$|Q_I|$, $|Q_B|$, and their sum $|Q_S|$ are plotted versus ψ_s in Figure B.5. Also the limits

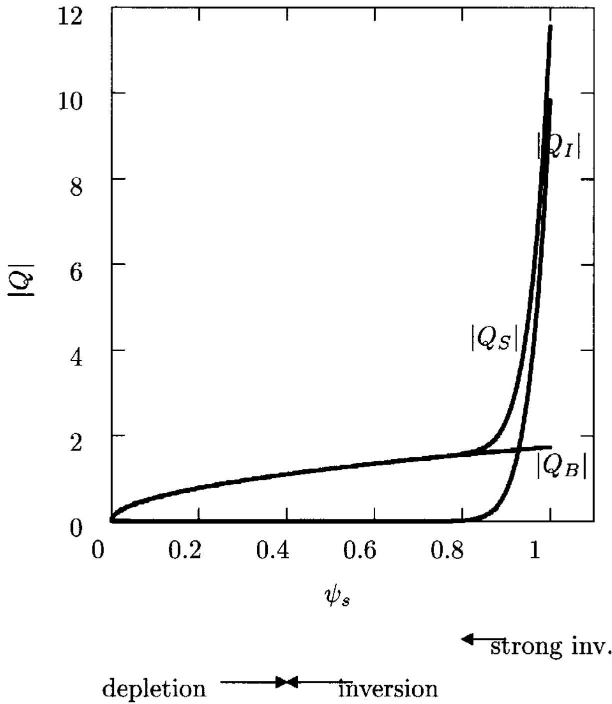


Figure B.5: The inversion layer charge Q_I, the depletion region charge Q_B, and their sum Q_S vs. the surface potential ψ_s for the two-terminal structure

for various conditions at the substrate surface are marked.

Remarks. *It appears that up to the on-set of the strong inversion condition, the inversion layer charge is negligible when compared to the depletion region charge. Once the strong inversion condition is entered, Q_I takes off exponentially.*

Further, by considering the voltage balance equation (B.2), the charge balance (B.3) and (B.4), and the charge-voltage relations (B.7), (B.8) and

$$Q_G = C_{ox}\psi_{ox} \tag{B.9}$$

it is possible to deduce a closed form relation between V_{GB} and ψ_s:

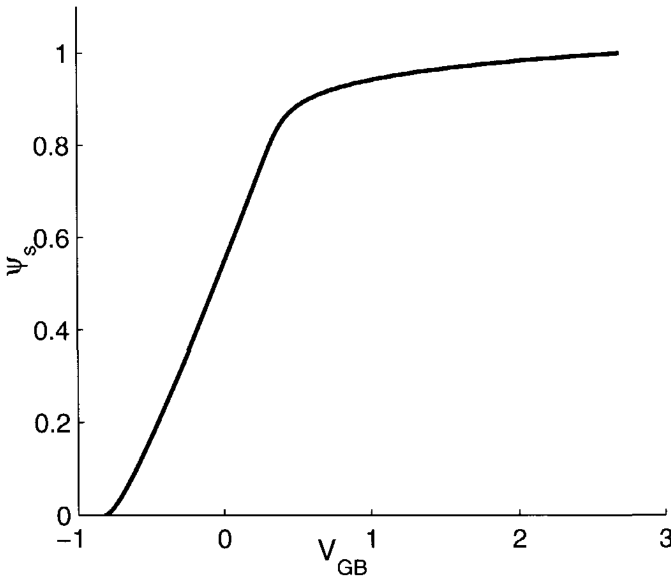


Figure B.6: The surface potential ψ_s vs. the gate-to-bulk voltage V_{GB} for the two-terminal structure

$$V_{GB} = V_{FB} + \psi_s - \frac{Q_B(\psi_s) + Q_I(\psi_s)}{C_{ox}} \tag{B.10}$$

which in inversion may be approximated by:

$$V_{GB} = V_{FB} + \psi_s + \gamma\sqrt{\psi_s + \phi_t e^{(\psi_s - 2\phi_F)\phi_t}} \qquad \text{where} \quad \gamma = \frac{F\sqrt{N_A}}{C_{ox}} \tag{B.11}$$

The dependency expressed by (B.11) is plotted in Figure B.6 on the preceding page.

Remarks. *Once the strong inversion condition is entered the surface potential ψ_s has a very little increase above the $2\phi_F$ on-set value, no matter how large the applied V_{GB} is. With relation to Figure B.5 it appears that in the strong inversion situation Q_B may be considered constant with V_{GB}.*

We resume the study of the two-terminal MOS structure with an useful formula which concerns the inversion layer charge. Q_I can be conveniently expressed as the charge on one plate of the oxide capacitor when the equivalent voltage on this capacitor is known:

$$Q_I = -C_{ox}(V_{GB} - V_{FB} - \psi_s - \gamma\sqrt{\psi_s}) \tag{B.12}$$

B.3 Diodes

The conductive channel, which consists of the inversion layer at the substrate surface, must be contacted in order to make use of its controllable electric properties. This is achieved by means of contact diffusions, of the same type as the induced channel (for a p-type substrate the extra diffusions are of n^+-type). By adding one more terminal to the two-terminal structure, the three-terminal structure (also called the MOS diode) is obtained. This rather academic situation is presented in Figure B.7; we will use this abstraction to examine how the condition of the substrate surface is influenced by a positive voltage V_{CB} applied between the contact diffusion and the bulk[3]. We make use of the energy bands diagram for electrons, presented in Figure B.8. At the substrate surface the Fermi level is set by the n^+-region to $E_{F,n}$. A positive V_{CB} value will establish the Fermi level of the n^+-diffusion with a quantity qV_{CB} lower than the Fermi level in the p-substrate. In

[3]Negative V_{CB} values are of no interest here, as they forward bias the n^+p diode, and are avoided during the normal operation.

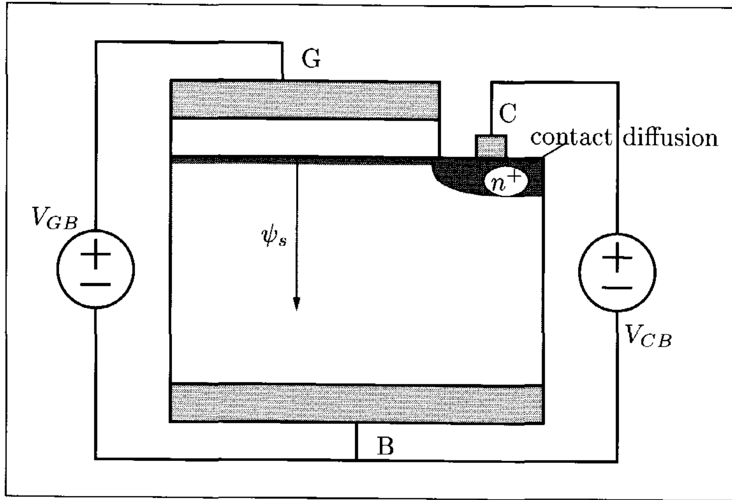


Figure B.7: Cross section through the three-terminal MOS structure

order to reestablish the equilibrium condition at the surface (similar to the $V_{CB} = 0$ case), the potential of the surface (ψ_s) must be raised at the level of the potential of the contact (i.e. with a V_{CB} quantity). Thus, the $\psi_s = V_{CB}$ situation for the three-terminal structure is equivalent with the $\psi_s = 0$ situation for the two-terminal structure, in what concerns the condition of the semiconductor surface. All the remarks from Section B.2 remain valid with the provision that a positive shift with a V_{CB} quantity is considered for ψ_s: $\psi_s = V_{CB} + \phi_F$ marks the beginning of the inversion condition ($n_{surface} = n_i$), while $\psi_s \approx V_{CB} + 2\phi_F$ marks the on-set of the strong inversion condition ($n_{surface} \approx n_A$).

The dependence of the condition at the substrate surface on the reverse contact-to-substrate bias V_{CB} is commonly called *body effect.* This effect stays central for the operation of the long-channel MOS transistor.

B.4 Transistors

We begin the study of the four-terminal structure by stating some important points which rely on the remarks from the previous sections. (1) For the large-signal

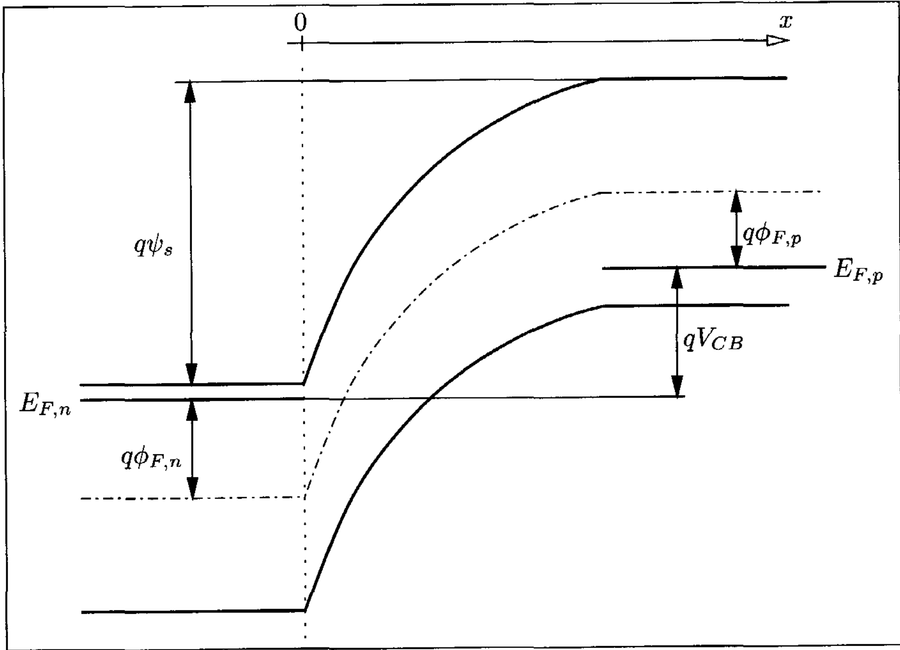


Figure B.8: Energy bands representation for the three-terminal MOS structure for the situation where surface potential $\psi_s = V_{CB} + \phi_{F,p} + \phi_{F,n}$ makes possible the conduction between the n^+-contact diffusion and the surface of the substrate

operation of the MOS transistor, as is the situation in digital circuits, only the strong inversion condition at the substrate surface is interesting in what concerns the existence of a conducting channel. This is based on the observation that only after ψ_s has been exceeded the $V_{CB} + 2\phi_F$ barrier the inversion charge really builds up at the semiconductor surface and establishes a fairly strong conducting path between the drain and source diffusions. (2) Prior to the on-set of the strong inversion, the inversion layer charge may be neglected when compared with the depletion region charge. (3) After the on-set of the strong inversion condition, the surface potential and the depletion region charge hardly increase with V_{CB}. We may consider for a strong inverted point C along the channel that $\psi_s \approx V_{CB} + 2\phi_F$ and $Q_B \approx -\gamma\sqrt{V_{CB} + 2\phi_F}$. (4) If none of the channel ends is in strong inversion there is no conducting path from source to drain and the drain current is zero.

Our objective is to deduce an approximate computationally efficient expression for the drain current. From the previous discussions we can infer that once the terminal voltages are known, the drain current is uniquely determined by these voltages. For the structure in Figure B.1 we consider an arbitrary fixed positive V_{SB} value and a V_{GS} value such as the source end of the channel is entered in strong inversion. By varying V_{DS} from zero towards positive and negative values we first study how the condition of the substrate surface changes near the drain and then how the drain current depends on the drain-to-source voltage.

The (V_{DS}, V_{GS}) voltage plane appears as a convenient place to delimit the operating regions of the transistor. The on-set of the strong inversion condition at the drain end of the channel is defined by:

$$V_{GB} = V_{FB} + \gamma\sqrt{V_{DB} + 2\phi_F} + V_{DB} + 2\phi_F \quad \text{or with } V_{DB} = V_{DS} + V_{SB} \quad \text{(B.13a)}$$

$$= V_{FB} + \gamma\sqrt{V_{DS} + V_{SB} + 2\phi_F} + V_{DS} + V_{SB} + 2\phi_F \quad \text{(B.13b)}$$

The condition for the on-set of the strong inversion at the drain end in terms of terminal voltages is then given by:

$$V_{GS} = V_{FB} + \gamma\sqrt{V_{DS} + V_{SB} + 2\phi_F} + V_{DS} + 2\phi_F \quad \text{(B.14)}$$

The on-set of the strong inversion at a point in the channel is modulated by V_{SB} through the γ coefficient; for this reason γ has been named the *body-effect coefficient.* We look now for a possibility to linearize the square-root term in (B.14). For `large-signal digital` circuit operation one may choose a first order expansion of the square-root term around large V_{DS} and small V_{SB}. Specifically, for $(V_{DD}, 0)$ as the origin point for the expansion we may approximate:

$$\sqrt{V_{DS} + V_{SB} + 2\phi_F} \cong \sqrt{V_{DD} + 2\phi_F} + \frac{\varphi_1}{2\sqrt{V_{DD} + 2\phi_F}}(V_{DS} - V_{DD})$$

$$+ \frac{\varphi_2}{2\sqrt{V_{DD} + 2\phi_F}}V_{SB}$$

The coefficients φ_1 and φ_2 are fitting coefficients[4] chosen for a low approximation error. The condition for the on-set of the strong inversion at the drain end can be linearized as:

$$V_{GS} = V_{FB} + \gamma\big(\sqrt{V_{DD} + 2\phi_F} + \frac{\varphi_1}{2\sqrt{V_{DD} + 2\phi_F}}(V_{DS} - V_{DD}) + \frac{\varphi_2}{2\sqrt{V_{DD} + 2\phi_F}} V_{SB}\big) + V_{DS} + 2\phi_F$$

At the source end the condition for the on-set of the strong inversion can be obtained by taking $V_{DS} = 0$ in the relation (B.14):

$$V_{GS} = V_{FB} + \gamma\sqrt{V_{SB} + 2\phi_F} + 2\phi_F \tag{B.15}$$

and further linearized in a similar manner to:

$$V_{GS} = V_{FB} + \gamma\big(\sqrt{V_{DD} + 2\phi_F} + \frac{\varphi_1}{2\sqrt{V_{DD} + 2\phi_F}}(-V_{DD}) + \frac{\varphi_2}{2\sqrt{V_{DD} + 2\phi_F}} V_{SB}\big) + 2\phi_F$$

The V_{GS} value at the on-set of the strong inversion at the source end is customarily named the *threshold voltage* of the MOS transistor and is denoted with V_T. The threshold voltage is defined in connection with the source terminal and depends on the source-to-bulk voltage V_{SB} through the body-effect coefficient γ:

$$V_T = V_{FB} + \gamma\big(\sqrt{V_{DD} + 2\phi_F} + \frac{\varphi_1}{2\sqrt{V_{DD} + 2\phi_F}}(-V_{DD}) + \frac{\varphi_2}{2\sqrt{V_{DD} + 2\phi_F}} V_{SB}\big) + 2\phi_F \tag{B.16}$$

The separation between the operating regions of the MOS transistor in the control-voltage plane (V_{DS}, V_{GS}) has been identified for the long channel case with the set

[4]The fitting coefficients have approximately the same value: $\varphi_1 \cong \varphi_2$.

(B.17) of two straight lines:

$$V_{GS} = V_T(V_{SB}) + (1+\delta)V_{DS} \quad \text{(B.17a)}$$

$$V_{GS} = V_T(V_{SB}) \quad \text{(B.17b)}$$

where $\delta = \gamma \frac{\varphi_1}{2\sqrt{V_{DD} + 2\phi_F}}$ is a constant coefficient for a given technology and $V_T(V_{SB})$ is given by (B.16).

Equation (B.17a) separates the region with- from the region without-strong inversion at the drain end, while equation (B.17b) defines a similar separation in what concerns the source end. The set (B.17) has been plotted with V_{SB} as parameter for the allowable V_{DS} and V_{GS} ranges in Figure B.9. The operating regions

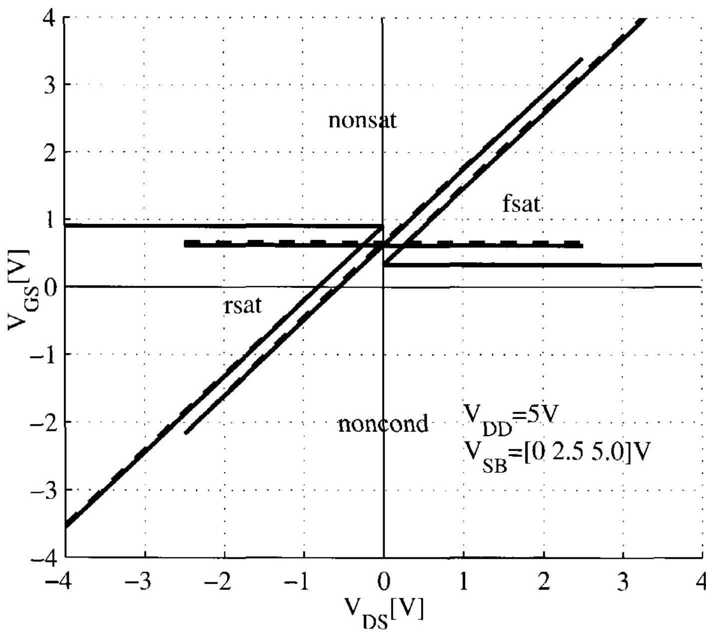


Figure B.9: Separation of the operating regions for a long-channel MOS transistor in the (V_{DS}, V_{GS}) plane

have been identified according to the presence and/or absence of the strong inversion condition at the ends of the channel. If only one end is in strong inversion the MOS transistor is said to be in the *saturation* region. If both ends are in strong inversion the MOS transistor is said to be in the *triode (non-saturation)* region. Finally, if neither end is in strong inversion the MOS transistor is said to be in the *non-conducting* region. These denominations come from the ability to conduct electric current in the specified regions.

We proceed now to the evaluation of the drain current under the assumption that a continuum of strong inversion exists all the way from source to drain. We aim for a general expression, valid for both positive and negative values of V_{DS}. Figure B.10 presents an electron moving from source to drain with the velocity $\overrightarrow{v}$ in an elemental transversal section of the channel; also shown are the local electric field $\overrightarrow{E(y)}$ and the electric current $\overrightarrow{I_D}$. Applying the charge conservation law one

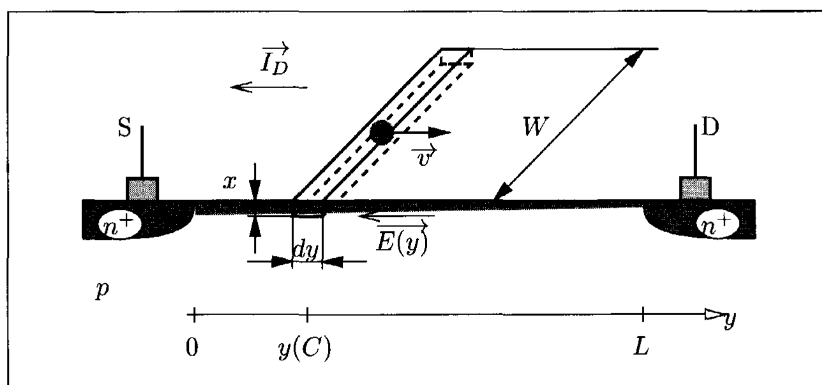


Figure B.10: An elemental section of the channel at the distance $y(C)$ from the source end

may find a global expression for the drain current:

$$I_D = WQ_Iv_d + WD\frac{dQ_I}{dy} \quad \text{with} \quad D = (kT/q)\mu \tag{B.18}$$

W is the transistor width, v_d is the drift velocity of the carriers, and D is the carrier diffusion constant. By neglecting the diffusion component of the current, an

approximate[5] formula for the drain current is found:

$$\mathbf{I_D} = -W|Q_I|\mathbf{v_d} \tag{B.19}$$

For a long channel we are in the low longitudinal field case and the drift velocity may be approximated as proportional to the magnitude of the longitudinal electric field:

$$|v_d| = \mu|E_y| = \mu\frac{d\psi_s}{dy} \cong \mu\frac{dV_{CS}}{dy} \qquad \text{for } V_{DS} \geq 0 \tag{B.20a}$$

$$= -\mu\frac{d\psi_s}{dy} \cong -\mu\frac{dV_{CS}}{dy} \qquad \text{for } V_{DS} \leq 0 \tag{B.20b}$$

μ is the low field surface mobility of the carriers in the channel. The approximations in (B.20) were possible due to a previous approximation on page 135 that for any strong inverted point C in the channel the surface potential satisfies $\psi_s(y) \approx V_{CB}(y) + 2\phi_F = V_{CS}(y) + V_{SB} + 2\phi_F$; it follows that $d\psi_s(y) \approx dV_{CS}(y)$. Recalling the expression (B.12) on page 134 for the inversion layer charge at point C in the channel we find that:

$$Q_I(y) = -C_{ox}\big(V_{GB} - V_{FB} - (V_{CS}(y) + V_{SB} + 2\phi_F) - \gamma\sqrt{V_{CS}(y) + V_{SB} + 2\phi_F}\big) \tag{B.21}$$

Using for the linearization of $\sqrt{V_{CS}(y) + V_{SB} + 2\phi_F}$ the same technique as used for the linearization of (B.14) on page 137 we obtain the following linearization for the inversion layer charge:

$$Q_I(y) = -C_{ox}\big(V_{GS} - V_T(V_{SB}) - (1+\delta)V_{CS}(y)\big) \tag{B.22}$$

The expression (B.19) for the drain current I_D becomes:

$$I_D = \mu W|Q_I|\frac{dV_{CS}}{dy} \tag{B.23}$$

with validity for both forward- and reverse-triode conduction. Integrating (B.23)

[5] This approximation is reasonable as long as the diffusion contributes a weak component to the total drain current, as is the case in the strong inversion.

over the channel length, with 0 and L as limits for y, and 0 and V_{DS} as limits for V_{CS}, we find the final formula for the drain current of a long channel MOS transistor in triode region:

$$I_D = \mu \frac{W}{L} C_{ox} \left((V_{GS} - V_T)V_{DS} - (1 + \delta)\frac{V_{DS}^2}{2} \right) \tag{B.24}$$

Equation (B.24) is valid as long as the strong inversion condition is satisfied at both ends of the channel, thus, according to (B.17) on page 139, for $V_{DS} \leq V_{DS,\,sat} = (V_{GS} - V_T)/(1 + \delta)$ and $V_{GS} \geq V_T$.

For $V_{DS} \geq V_{DS,\,sat}$ the drain end in not anymore in strong inversion and the drain current saturates at the value $I_{D,\,sat}$ corresponding to $V_{DS,\,sat}$. The situation when at the drain end the channel is not anymore in strong inversion is commonly called *pinch-off*; formula (B.22) on the preceding page predicts zero inversion layer charge for the points of the channel where $V_{CS} \geq V_{DS,\,sat}$.

For $V_{GS} \geq V_T$[6] the conduction of a long channel MOS transistor can be summarized in a compact manner:

$$I_D = \begin{cases} \mu \frac{W}{L} C_{ox} \left((V_{GS} - V_T)V_{DS} - (1 + \delta)\frac{V_{DS}^2}{2} \right) & \text{if } V_{DS} \leq V_{DS,\,sat} \text{ and } V_{GS} \geq V_T \\ \mu \frac{W}{L} C_{ox} \frac{(V_{GS} - V_T)^2}{2(1 + \delta)} & \text{if } V_{DS} \geq V_{DS,\,sat} \text{ and } V_{GS} \geq V_T \end{cases} \tag{B.25}$$

with $V_{DS,\,sat}$ as solution for (B.17a) on page 139 and V_T given by (B.16) on page 138.

The I_D vs. V_{DS} characteristic family for a long channel MOS transistor is presented in Figure B.11; one may remark that the separation between triode and

[6]For $V_{GS} \leq V_T$ and with sufficiently negative V_{DS} as the drain end of the channel is in strong inversion, the transistor is in reverse saturation; one may find that:

$$I_D = -\mu \frac{W}{L} C_{ox} \frac{(V_{GS} - V_T - (1 + \delta)V_{DS})^2}{2(1 + \delta)} \quad \text{if } V_{GS} \leq V_T \text{ and } V_{DS} \leq \frac{V_{GS} - V_T}{1 + \delta}.$$

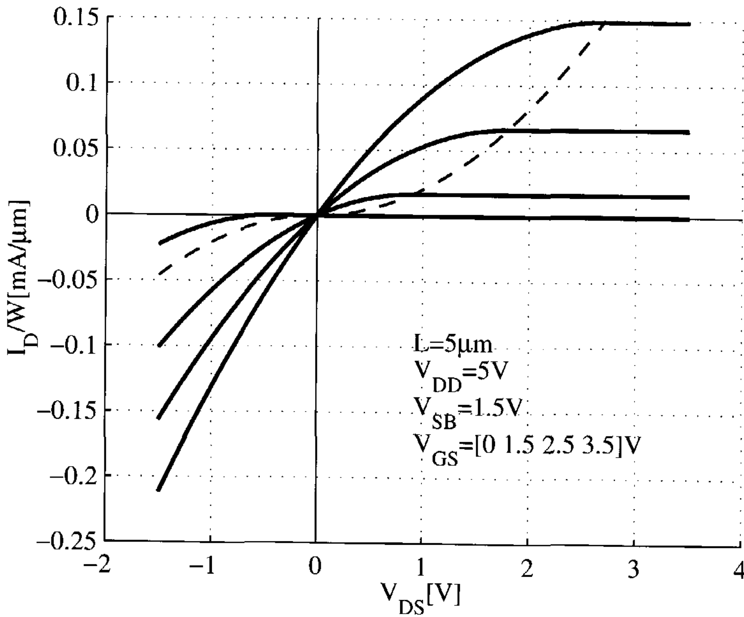


Figure B.11: I_D vs. V_{DS} characteristic family for a long channel MOS transistor

saturation in the I-V characteristic plane is a parabola with the equation:

$$I_{D,\,sat} = \mu \frac{W}{L} C_{ox} \frac{1+\delta}{2} V_{DS,\,sat}^2 \qquad \text{(B.26)}$$

B.5 Remarks

We conclude this appendix with two remarks concerning the conduction regime of a long channel MOS transistor.

1. The separation between the triode and the saturation regions in the (V_{DS}, V_{GS}) control voltage plane is a line segment, as has been shown in Figure B.9 on page 139.

2. The drain saturation current $I_{D,\,sat}$ has a parabolical dependence on the gate drive $(V_{GS} - V_T)$, as comes out from relation (B.25).

Appendix C

VHDL summary

VHDL[1], as hardware description language, uses concurrency for modeling the way real circuits behave. VHDL can be used for structural specification and/or for behavioral description of hardware [50, 51]. It can also be used for hardware synthesis [52] if we use a subset of the language.

In the following we recap on some VHDL modeling concepts. As an example we look at ways of describing a 4-bit register, shown in Figure C.1.

Using VHDL terminology, we call the module `reg4` a design *entity*, and the inputs and outputs are *ports*. Figure C.2 shows a VHDL description of the interface to this entity. This is an example of an *entity declaration*. It introduces a name for the entity and lists its input and output ports, specifying that they carry bit values ('0' or '1') into and out of the entity. From this we see that an entity declaration describes the external view of the entity.

C.1 Elements of behavior

In VHDL, a description of the internal implementation of an entity is called an *architecture body* of the entity. There may be a number of different architectures bodies of one interface to an entity, corresponding to alternative implementations that perform the same function. We can write a *behavioral* architecture body of an

[1] VHDL stands for VHSIC Hardware Description Language, where VHSIC stands for Very High Speed Integrated Circuits.

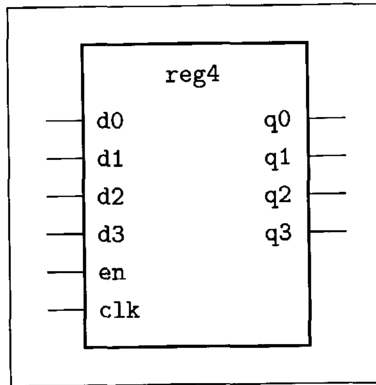


Figure C.1: A four-bit register module. The register is named **reg4** and has six inputs, d0, d1, d2, d3, en, and clk, and four outputs, q0, q1, q2, and q3.

```
entity reg4 is
   port(d0, d1, d2, d3, en, clk: in bit;
        q0, q1, q2, q3: out bit);
end entity reg4;
```

Figure C.2: A VHDL entity description of a four-bit register

entity, which describes the function in an abstract way. Such an architecture body includes only *process statements*, which are collections of actions to be executed in sequence. These actions are called *sequential statements* and are much like the kinds of statements we see in a conventional programming language. The types of actions that can be performed include evaluating expressions, assigning values to variables, conditional execution, repeated execution, and subprogram calls. In addition there is a sequential statement that is unique to hardware modeling languages, the *signal assignment* statement. This is similar to variable assignment, except that it causes the value on a signal to be updated at some future time.

To illustrate these ideas, let us look at a behavioral architecture body for the reg4 entity, shown in Figure C.3. In this architecture body, the part after the first

```
architecture behav of reg4 is
begin
    storage: process is
        variable stored_d0, stored_d1, stored_d2, stored_d3: bit;
    begin
        if en = '1' and clk = '1' then
            stored_d0 := d0;
            stored_d1 := d1;
            stored_d2 := d2;
            stored_d3 := d3;
        end if;
        q0 <= stored_d0 after 5 ns;
        q1 <= stored_d1 after 5 ns;
        q2 <= stored_d2 after 5 ns;
        q3 <= stored_d3 after 5 ns;
        wait on d0, d1, d2, d3, en, clk;
    end process storage;
end architecture behav;
```

Figure C.3: A behavioral architecture body of the `reg4` entity

begin keyword includes one process statement, which describes how the register behaves. It starts with the process name, `storage`, and finishes with the keywords **end process**.

The process statement defines a sequence of actions that are to take place when the system is simulated. These actions control how the values of the entity's ports change over time, i.e., they control the behavior of the entity. This process can modify the values of the entity's ports using signal assignment statements.

The way this process works is as follows. When the simulation is started, the signal values are set to '0', and the process is activated. The process's variables (listed after the keyword **variable**) are initialized to '0', then the statements are executed in order. The first statement is a condition that tests whether the values of the `en` and `clk` signals are both '1'. If they are, the statements between the keywords **then** and **end if** are executed, updating the process's variables using the values of the input signals. After the conditional if statement, there are four signal assignment statements that cause the output signals to be updated 5 ns later.

When all of these statements in the process have been executed, the process reaches the *wait statement* and *suspends*, i.e., it becomes inactive. It stays suspended until one of the signals to which it is *sensitive* changes value. In this case, the process is sensitive to the signals `d0, d1, d2, d3, en`, and `clk`, since they are listed in the wait statement. When one of these changes value, the process is resumed. The statements are executed again, starting from the keyword **begin**, and the cycle repeats. Notice that while the process is suspended, the values in the process's variables are not lost. This is how the process can represent the state of a system.

C.2 Elements of structure

An alternative way of describing the implementation of an entity is to specify how it is composed of subsystems. We can give a structural description of the entity's implementation. An architecture body that is composed only of interconnected subsystems is called a *structural* architecture body. Figure C.4 shows how the `reg4` entity might be composed of latches and gates. If we are to describe this in VHDL, we will need entity declarations and architecture bodies for the subsystems, shown in Figure C.5.

Figure C.6 is a VHDL architecture body declaration that describes the structure shown in Figure C.4. The *signal declaration*, before the keyword **begin**, defines the internal signals of the architecture. In this example, the signal `int_clk` is declared to carry a bit value ('0' or '1'). In general, VHDL signals can be declared to carry arbitrarily complex values. Within the architecture body the ports of the entity are also treated as signals.

In the second part of the architecture body, a number of *component instances* are created, representing the subsystems from which the `reg4` entity is composed. Each component instance is a copy of the entity representing the subsystem, using the corresponding `basic` architecture body. (The name **work** refers to the current working library, in which all of the entity and architecture body descriptions are assumed to be held.)

The *port map* specifies the connection of the ports of each component instance to signals within the enclosing architecture body. For example, `bit0`, an instance of the `d_latch` entity, has its port `d` connected to the signal `d0`, its port `clk` connected to the signal `int_clk`, and its port `q` connected to the signal `q0`.

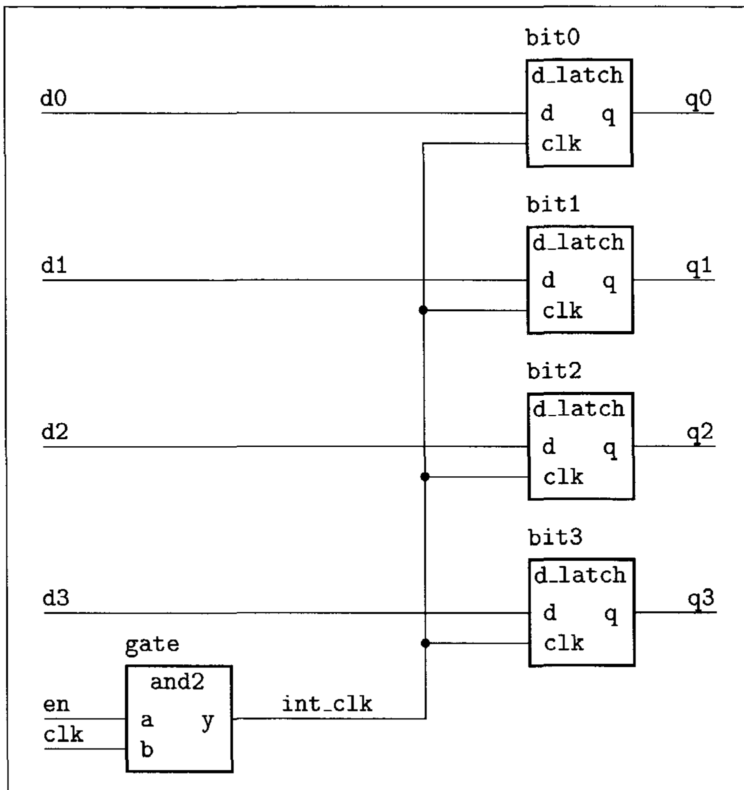


Figure C.4: A structural composition of the `reg4` entity

C.3 Mixed structural and behavioral models

Models need not be purely structural or purely behavioral. Often it is useful to specify a model with some parts composed of interconnected component instances, and other parts described using processes. We use signals as the means of joining component instances and processes. A signal can be associated with a port of a component instance and can also be assigned to or read in a process.

We can write such a hybrid model by including both component instances and process statements in the body of an architecture. These statements are collectively called *concurrent statements*, since the corresponding processes all execute

```
entity d_latch is
    port (d, clk: in bit; q: out bit);
end d_latch;

architecture basic of d_latch is
begin
    latch_behavior: process is
    begin
        if clk = '1' then
            q <= d after 2 ns;
        end if;
        wait on clk, d;
    end process latch_behavior;
end architecture basic;
--------------------------------
entity and2 is
    port (a, b: in bit; y: out bit);
end and2;

architecture basic of and2 is
begin
    and2_behavior: process is
    begin
        y <= a and b after 2 ns;
        wait on a, b;
    end process and2_behavior;
end architecture basic;
```

Figure C.5: Entity declarations and architecture bodies for the D-flipflop and two-input and2 gate

concurrently when the model is simulated. An outline of such a model is shown in Figure C.7. This model describes a multiplier consisting of a data path and a control section. The data path is described structurally, using a number of component instances. The control section is described behaviorally, using a process that assigns to the control signals for the data path.

```
architecture struct of reg4 is
    signal int_clk: bit;
begin
    bit0: entity work.d_latch(basic)
        port map (d0, int_clk, q0);
    bit1: entity work.d_latch(basic)
        port map (d1, int_clk, q1);
    bit2: entity work.d_latch(basic)
        port map (d2, int_clk, q2);
    bit3: entity work.d_latch(basic)
        port map (d3, int_clk, q3);
    gate: entity work.and2(basic)
        port map (en, clk, int_clk);
end architecture struct;
```

Figure C.6: A VHDL structural architecture body of the `reg4` entity

```
entity multiplier is
    port (clk, reset: in bit;
          multiplicand, multiplier: in integer;
          product: out integer);
end entity multiplier;

architecture mixed of multiplier is
    signal partial_product, full_product: integer;
    signal arith_control, result_en, mult_bit, mult_load: bit;
begin -- mixed
    arith_unit: entity work.shift_adder(behavior)
        port map (addend => multiplicand, augend => full_product,
                  sum => partial product,
                  add_control => arith_control);
    result: entity work.reg(behavior)
        port map (d => partial_product, q => full_product,
                  en => result_en, reset => reset);
    multiplier_sr: entity work.shift_reg(behavior)
        port map (d => multiplier, q => mult_bit,
                  load => mult_load, clk => clk);
    product <= full_product;
    ------------------------
    control_section: process is
        -- variable declarations for control_section
        -- ...
    begin -- control section
        -- sequential statements to assign values to control signals
        -- ...
        wait on clk, reset;
    end process control_section;
end architecture mixed;
```

Figure C.7: An outline of a mixed structural and behavioral model of a multiplier

Appendix D

Charge pump PLL

The phase-lock loop that we used in our experiments to test the simulation capabilities of our approach on mixed digital-analog circuits is taken from [53].

It is a charge pump type loop [37] used for clock signal generation and resynchronization on a digital Sea-of-Gates chip. It is suitable for this application due to the extended tracking range (useful during the testing and experimenting phase), the frequency-aided acquisition (which contributes to a short acquisition time over a wide range of reference frequencies), the low phase jitter of the output signal due to the *high-Z* state of the pump, the relative simple implementation in digital hardware, and the simple filter. The main blocks of such a loop are presented in Figure D.1.

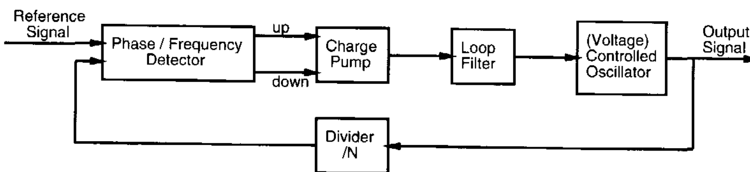


Figure D.1: Charge-pump PLL

The blocks of this loop with an essentially analog behavior are the voltage controlled oscillator (VCO), the loop filter, and the charge pump. For the oscillator, a ring oscillator type built with current starved inverters has been chosen. This is

presented in Figure D.2. The functioning of such an oscillator requires an analog approach because of the presence of current sources (the top row of p transistors and the bottom row of n transistors), and of the current mirror (left-top).

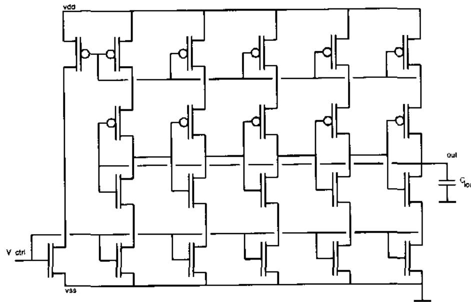


Figure D.2: Ring oscillator with 5 cascaded current-starved inverters

A simple analytical model for the VCO, i.e., the control characteristic frequency vs. control-voltage, was found through a simulation based approach. In the linear model of the VCO operation in the vicinity of the static point (loop locked), namely $\omega_{osc} = \omega_0 + K_o * V_{ctrl}$, the oscillator sensitivity (K_o) and the free-running frequency (ω_0) were found by performing the Taylor expansion of the polynomial expression of the VCO characteristic, up to the first order term, in the desired static point.

The charge pump is a voltage pump, presented in Figure D.3. A simplified

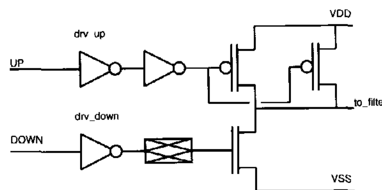


Figure D.3: Voltage charge-pump

physical model, equivalent to a current charge pump, is presented in Figure D.4, with

$I_{p_UP} = (V_{DD} - V_d)/R_{sw_UP}$ and

$I_{p_DOWN} = V_d/R_{sw_DOWN}$,

where I_{p_UP} is the pump charge current when the p transistors are in conduction, I_{p_DOWN} is the pump discharge current when the n transistor is in conduction,

V_d is the voltage at the input of the loop filter, V_{DD} the power source voltage (drain voltage), R_{sw_UP} and R_{sw_DOWN} are the resistances of the p and respectively n transistors when in conduction. These two equations show very well the analog approach needed for the charge pumps, impossible to mimic by a classical digital simulation.

The phase detector gain (including the charge pump) is $K_d = (\frac{V_d(s)}{\theta(s)}) = \frac{I_p}{2\pi}$.

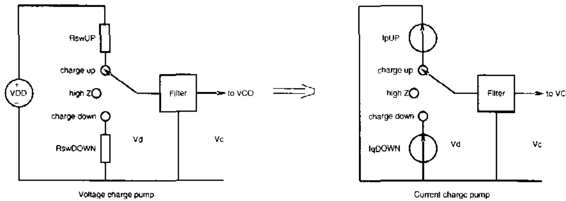


Figure D.4: Models of the charge-pump

The loop filter is a passive RC series circuit, as presented in Figure D.5. This also requires an analog approach when simulated. The loop was designed to operate at a VCO frequency of 110 MHz, which requires a control voltage of about 2.1 V.

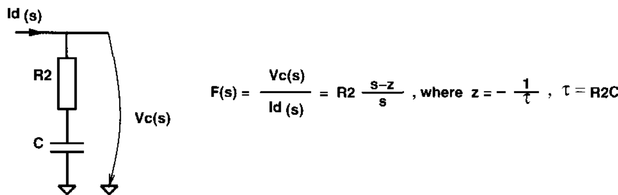


Figure D.5: Loop filter

The mathematical linear model (in the Laplace domain) of the PLL is presented in Figure D.6. Using the same notation as in [54], the transfer function of the closed loop is: $H(s) = \frac{\theta_o(s)}{\theta_i(s)} = \frac{\frac{K_o K_d}{N_d}\frac{F(s)}{s}}{1+\frac{K_o K_d}{N_d}\frac{F(s)}{s}}$.

With this type of filter, the loop is a type II, second order loop. It is stable when the continuous-time approximation is valid (narrow bandwidth compared to the input frequency), with zero static phase error.

The maximum value of the resistor is set by the condition that the −3 dB bandwidth of the loop should be much lower than the rate of the input signal (the reference):

$\omega_{-3dB} < \frac{\omega_{ref}}{10}$, where:

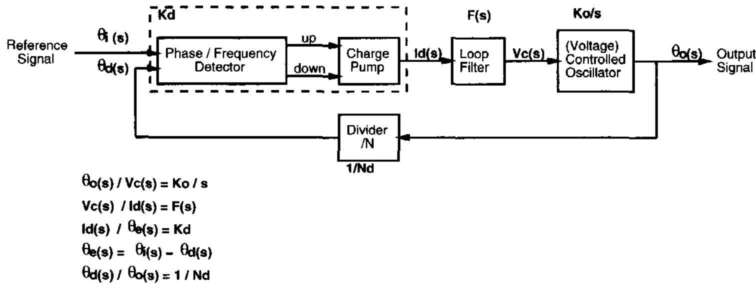


Figure D.6: Linear model of the loop.

$\omega_{-3dB} = \frac{K_o K_d}{N_d} K_h,$
$K_o = 77.37$ MHz/V,
$K_d = 7.96 \cdot 10^{-4}$ A/rad,
$N_d = 15$
$K_h = F(s \to \infty) = R_2,$
$\omega_{ref} = 7.33$ MHz,
which results in $R_2 < 180\Omega$.

The value of the capacitance C results from the condition $\zeta \approx 0.707$, which leads to $C = 9.565$ nF for $R_2 = 180\Omega$ ($\zeta = \frac{\tau_2}{2}\sqrt{\frac{\frac{K_o K_d}{N_d}}{C}}$, from [54]).

With this values, the natural frequency of the loop is: $\omega_n = \sqrt{\frac{\frac{K_o K_d}{N_d}}{C}} \Rightarrow f_n =$ 262 kHz. The closed loop gain is presented in Figure D.7.

The phase/frequency detector and the frequency divider are pure digital circuits. They are presented in Figure D.8 and Figure D.9. The role of the delay elements *del_1* and *del_2* in Figure D.8 is to avoid parasitic spikes at one output (UP/DOWN) when the other output (DOWN/UP) is switching.

The frequency divider is similar to a PN generator. The *dead lock detector* circuit eliminates the so-called *all zero-s* state of a PN generator (that is actually 0101 due to the buffers/inverters placed between flip-flops). The *pulse generator* circuit detects the *all ones* state (which is 1010 for the same reason). The divider factor is $2^N - 1$, N being the number of D flip-flops in the shift-register. Such a configuration allows a high clocking rate, higher than what can be achieved by a configuration using a ripple counter.

Figure D.10 presents several waveforms of the loop just reaching the steady state (tracking with zero phase error). The loop was designed to generate a 110

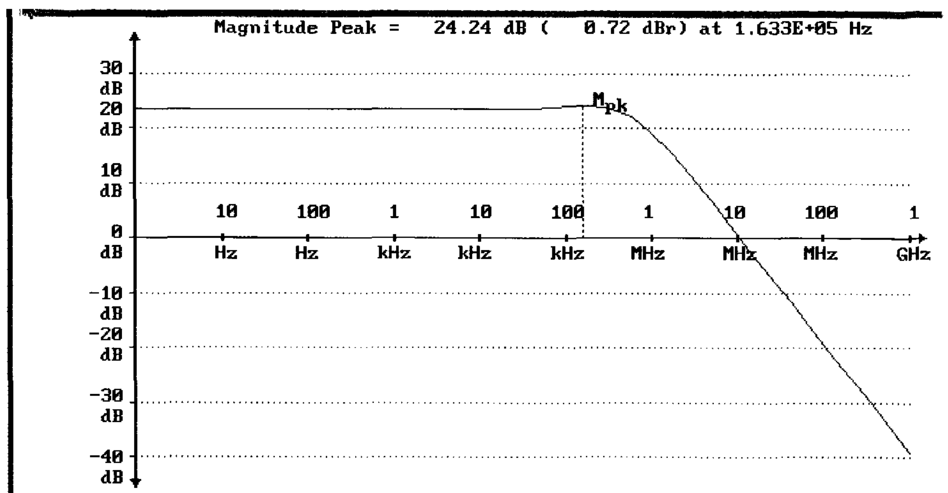


Figure D.7: Closed loop gain

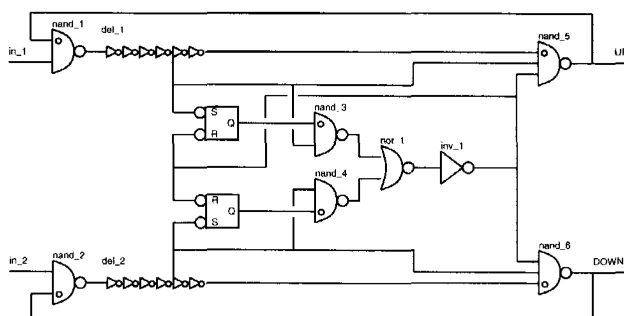


Figure D.8: The phase / frequency detector.

MHz signal at the output of the VCO. One can see how the loop builds-up the control voltage of the oscillator, in order to lock the VCO to the external reference.

When the loop is locked, the phase jitter due to the charge pulses may be too large for some applications, even if the pulses are very narrow. (The control voltage of the VCO presents ripple, instantaneous voltage jumps of $\Delta v_c = I_p R_2$, which produce frequency jumps of the VCO). This ripple can be reduced by reducing R_2 (and accordingly increasing C), or by using a higher order filter to mitigate this ripple. Such a filter is presented in Figure D.11. The loop is in this case a third

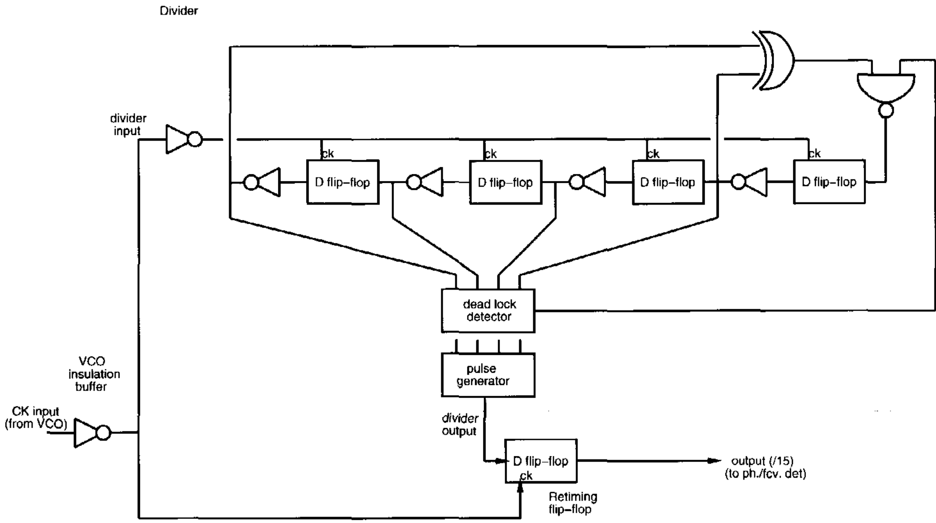


Figure D.9: Frequency divider, by 15.

order, type II loop.

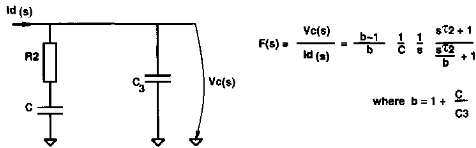


Figure D.11: Second order filter

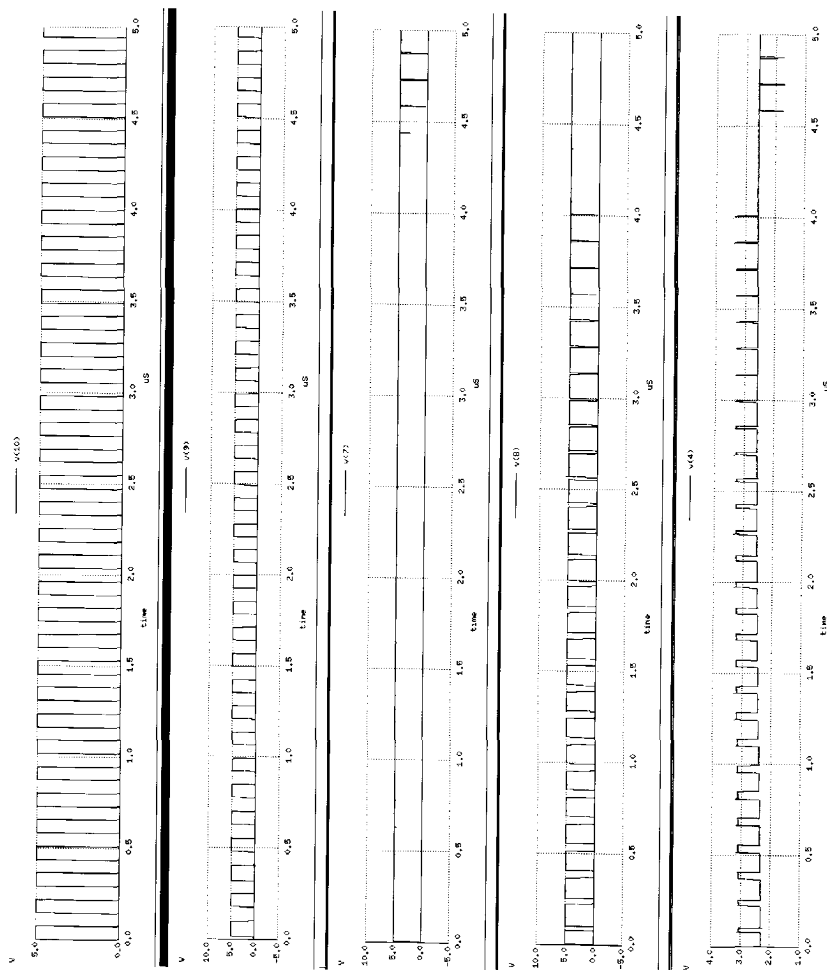


Figure D.10: Representative voltage waveforms in the loop v(10)=the reference signal v(9)=the output of the divider (/16) v(8)=the *ch_up* output of the phase detector v(7)=the *ch_dw* output of the phase detector v(4)=the control voltage of the oscillator

Bibliography

[1] SEMATECH, "National technology roadmap for semiconductor," tech. rep., Semiconductor Industry Association, 1997.

[2] S. M. Sze, *Physics of Semiconductor Devices.* New York : Wiley, 1981.

[3] A. P. Chandrakasan and R. W. Brodersen, *Low power digital CMOS design.* Kluwer, Boston, 1995.

[4] Y. Taur and T. H. Ning, *Fundamentals of modern VLSI devices.* Cambridge : Cambridge University Press, 1998.

[5] J. R. Brews, *High-Speed Semiconductor Devices*, ch. 3:The Submicron MOSFET. John Wiley & Sons, 1990.

[6] H. B. Bakoglu, *Circuits, interconnections, and packaging for VLSI.* Reading : Addison-Wesley, 1990.

[7] D. A. Kirkpatrick, *The implications of DSM technology on the design of high performance digital VLSI systems.* PhD thesis, University of California, Berkeley, 1997.

[8] A. Christou, *Electromigration and electronic device degradation.* New York : Wiley, 1994.

[9] M. Keating and P. Bricaud, *The reuse methodology manual for System-on-a-Chip designs.* Kluwer Academic Publishers, 1998.

[10] C. Bruma, *Systems-on-a-Chip architecting.* PhD thesis, Delft University of Technology, The Netherlands, 1999.

[11] D. M. W. Leenaerts and W. M. G. Bokhoven, *Piecewise linear modeling and analysis.* Kluwer Academic Publishers, 1998.

[12] W. M. G. van Bokhoven, *Circuit analysis, simulation and design*, vol. 3 of *Advances in CAD for VLSI*, ch. 9:PIECEWISE LINEAR ANALYSIS AND SIMULATION, pp. 129–166. Amsterdam: North-Holland, 1987.

[13] W. M. G. van Bokhoven, *Piecewise Linear Modelling and Analysis.* Deventer, The Netherlands: Kluwer Technische Boeken, 1981.

[14] R. W. Cottle, J. S. Pang, and R. E. Stone, *The linear complementarity problem.* Academic Press, 1992.

[15] M. Shoji, *Theory of CMOS Digital circuits and Circuit Failures.* Princeton University Press, Princeton, 1992.

[16] Y. P. Tsividis, *Operation and modeling of the MOS transistor.* New York : McGraw-Hill, 1987.

[17] L. A. Glasser and D. W. Dobberpuhl, *The design and analysis of VLSI circuits.* Addison-Wesley Publishing Company, 1985.

[18] University of California, Berkeley, *BSIM3v3.2.2 MOSFET Model Users' Manual*, 1999.

[19] S. M. Sze, *Physics of Semiconductor Devices.* John Willey & Sons, Inc., 1969.

[20] F. Trofimenkoff, "Field-dependent mobility analysis of the field-effect transistor," *Proceedings of the IEEE*, vol. 53, pp. 1765–1766, january 1965.

[21] T. Sakurai and A. R. Newton, "Alpha-power law mosfet model and its applications to cmos inverter delay and other formulas," *IEEE Journal of SOLID-STATE CIRCUITS*, vol. 25, pp. 584–594, april 1990.

[22] S. Bruma, "A simulation tool for deep submicron cmos circuits," tech. rep., Technical University Delft, 1996.

[23] E. Lelarasmee, A. Ruehli, and A. Sangiovanni-Vincentelli, "The waveform relaxation method for time domain analysis of large scale integrated circuits," *IEEE Transactions on CAD/ICAS*, 1982.

[24] S. A. Szygenda and E. W. Thompson, "Digital logic simulation in a time-based, table-driven environment; part 1, design verification," *IEEE Computer*, vol. 8, no. 3, pp. 24–49, 1975.

[25] The Institute of Electrical and Electronic Engineers, Inc., *IEEE Standard VHDL Language Reference Manual*, ieee-std. 1076-1987 ed., 1988.

[26] The Institute of Electrical and Electronic Engineers, Inc., *IEEE Standard Verilog Language Reference Manual*, ieee std. 1364-1996 ed., 1996.

[27] M. Abramovici, M. A. Breuer, and A. D. Friedman, *Digital Systems Testing and Testable Design.* Computer Science Press, 1990.

[28] T. Smedes, *Compact Modelling of the Dynamic Behaviour of MOSFETS.* PhD thesis, Eindhoven University of Technology, 1991.

[29] D. E. Ward and R. W. Dutton, "A charge-oriented model for mos transistor capacitances," *IEEE Journal of Solid State Circuits*, vol. 13, no. 5, pp. 703–707, 1978.

[30] S. Bruma and R. H. J. M. Otten, "The vhdl-mos package for transistor-level event-driven simulation," in *Proc. IEEE International ASIC/SOC Conference*, pp. 86–90, 1999.

[31] META-SOFTWARE, INC., *HSPICE User's Manual*, 1996.

[32] Model Technology, *V-System/PLUS Workstation User's Manual*, 1997.

[33] Q. J. A. van Gemert and J. T. J. van Eijndhoven, *The NDML Network Description and Modeling Language.* Eindhoven University of Technology, 1988.

[34] M. T. van Stiphout, *PLATO Users Guide.* Eindhoven University of Technology.

[35] N. H. E. Weste and K. Eshraghian, *Principles of CMOS VLSI Design, A Systems Perspective.* Addison-Wesley publishing company, 1993.

[36] H. W. Buurman, *From circuit to signal: development of a piecewise linear simulator.* PhD thesis, Eindhoven University of Technology, 1993.

[37] F. M. Gardner, "Charge-pump phase-lock loops," *IEEE Transactions on Communications*, vol. 28, pp. 1849–1858, November 1980.

[38] T. A. M. Kevenaar, *PLANET: a hierarchical network simulator.* PhD thesis, Eindhoven University of Technology, 1993.

[39] S. M. Sze, *Semiconductor devices: physics and technology.* New York : Wiley, 1985.

[40] Waterloo Maple Software, *Maple.* info@maplesoft.on.ca.

[41] J. Katzenelson, "An algorithm for solving nonlinear resistor networks," *Bell Syst. Tech. J.*, vol. 44, 1965.

[42] R. W. Cottle and G. Dantzig, "Complementary pivot theory of mathematical programming," *Linear Algebra and Applications*, 1968.

[43] C. E. Lemke, "Bimatrix equilibrium points and mathematical programming," *Management Science*, 1965.

[44] P. Jones, "Even more with the lemke complementary algorithm," *Mathematical Programming*, 1986.

[45] M. Chien and E. Kuh, "Solving piecewise linear equations for resistive networks," *International Journal of Circuit theory and Applications*, 1976.

[46] C. van de Panne, "A complementary variant and a solution algorithm for piecewise linear resistor networks," *SIAM J. Math. Anal.*, vol. 8, 1977.

[47] M. van Stiphout, *PLATO - a Piecewise Linear Analysis Tool for mixed-level circuit simulation.* PhD thesis, Eindhoven University of Technology, 1990.

[48] H. C. de Graaff and F. M. Klaassen, *Compact transistor modelling for circuit design.* Wien : Springer, 1990.

[49] A. S. Grove, *Physics and Technology of Semiconductor Devices.* John Wiley and Sons, Inc., 1967.

[50] P. J. Ashenden, *The Designer's Guide to VHDL.* Morgan Kaufmann Publishers, Inc., 1996.

[51] Z. Navabi, *VHDL Analysis and Modelling of Digital Systems.* McGraw Hill, Inc., 1993.

[52] Y.-C. Hsu, K. F. Tsai, J. T. Liu, and E. S. Lin, *VHDL Modeling for Digital Design Synthesis.* Kluwer Academic Publishers, 1995.

[53] C. Bruma, "An on-chip, high rate clock signal generator," tech. rep., Delft University of Technology, 1995.

[54] Floyd M. Gardner, *Phaselock Techniques*, ch. 2. John Wiley & Sons, 1979.

Glossary

A/D
: Analog to Digital (converter)

AC
: Alternating Current

ASIC
: Application Specific Integrated Circuit

CAD
: Computer Aided Design

CCSM
: Collapsible Current Source Model

CE
: Constitutive Equation

CMOS transistor
: Complementary Metal Oxide Semiconductor transistor

CPU
: Central Processing Unit

D/A
: Digital to Analog (converter)

DC
: Direct Current

DIMES

Delft Institute of MicroElectronics and Submicron technology

DSM

Deep Sub-Micron

DSP

Digital Signal Processor

EDA

Electronic Design Automation

FS

Forward Saturation (operating region of a MOST)

HDL

Hardware Description Language

HLD

High Level Description

HW

HardWare

I/O

Input or Output

$I_{D,\,SAT}$

Drain SATuration current (of a DSM MOST)

IEEE

Institute of Electrical and Electronic Engineers

IC

Integrated Circuit

IP

Intellectual Property

KCL

Kirchhoff's Current Law

KCVL

Kirchhoff's Current and Voltage Laws

KVL

Kirchhoff's Voltage Law

LCP

Linear Complementarity Problem

MC

MOS Current

MOS transistor

Metal Oxide Semiconductor transistor

MOS(FE)T

Metal Oxide Semiconductor (Field Effect) Transistor

N-R

Newton-Raphson

NC

Non-Conduction (operating region of a MOST)

NDML

Network Description and Modeling Language

NQS

Non Quasi-Static

PL

Piecewise Linear

PLL

Phase-Lock Loop

PLATO

Piecewise Linear Analysis TOol

QS

Quasi-Static

RAM

Random Access Memory

RC

Resistor-Capacitor (delay)

RF

Radio Frequency

RS

Reverse Saturation (operating region of a MOST)

RTL

Register Transfer Level

SEMATECH

SEmiconductor MAnufacturing TECHnology

SIA

Semiconductor Industry Association

SoG

Sea-of-Gates (image)

SPICE

Simulation Program with Integrated Circuit Emphasis

SW

SoftWare

TE

Topological Equation

ULSI

Ultra Large Scale Integration

V_{DD}

Supply Voltage of a CMOS IC

$V_{DS,\, SAT}$

Drain-Source SATuration Voltage (of a DSM MOST)

VCO

Voltage Controlled Oscillator

VHDL

Very (High Speed IC) Hardware Description Language

VL

Voltage Level

VLSI

Very Large Scale Integration

About the author

Serban Bruma was born on August 22, 1968 in Iaşi, Romania. From 1982 to 1986 he attended the Lyceum "Costache Negruzzi" in Iaşi, Romania, where he received the "Diploma de Bacalaureat".

In 1987 he began his study in Electrical Engineering at the Polytechnic Institute of Iaşi, Romania. In 1992, at the end of the five year study program, he graduated and received the "Diploma de Inginer" .

His first job was as telecommunication engineer with the digital switching division of ROM-TELECOM, the national telecommunications operator of the country. In parallel, he ran together with two other young enthusiastic engineers his private company producing microcomputers.

In 1994 he joined as a research assistant the Circuits and Systems Group of Prof. Ralph Otten at the Delft University of Technology, the Netherlands. Here he researched on the effects of down-scaling the minimum feature size of a CMOS technology, with emphasis on the compact modeling for the deep submicron devices. After completing a two year post-graduate study program, he received the "Chartered Designer in Microelectronics" diploma from the same university. In 1997 he visited Compass Design Automation in San Jose, California, as a summer intern. He continued his research with a focus on efficient mixed-mode analysis of large size deep submicron digital CMOS circuits. At the Delft University is where he did the research described in this thesis.

In the spring of 1999 he joined the Electronic Design & Tools group at Philips Research Eindhoven, the Netherlands.

Samenvatting

De stijgende vraag naar meer computerkracht heeft de CMOS technologie in het deep submicron tijdperk gebracht. Teneinde complexe systemen op een chip te integreren moeten de ontwerpers kunnen omgaan met de diversiteit en complexiteit van deep submicron schakelingen.

Dit proefschrift beschrijft het onderzoek naar de efficiente analyse van zeer grote en complexe deep submicron schakelingen. Het aspect van de *diversiteit* vraagt om het gebruik van een enkel simulatiealgoritme dat alle abstractieniveaus van een systeembeschrijving aan kan. Het aspect van de *complexiteit* vraagt om een eenvoudig en compact model van de deelschakelingen.

We laten zien dat de event-driven methode, die gebaseerd is op locale modellering en globale signalering, een beter alternatief is voor de efficiente simulatie van diverse abstractieniveaus dan methodes die gebaseerd zijn op vergelijkingen. De stuksgewijs-lineaire methode, die gebaseerd is op diodetoestanden en daarom voldoet aan de eis van één algoritme voor meerdere abstractieniveaus, blijkt niet erg efficiënt in z'n berekeningen te zijn.

Wij gebruiken een compact, stuksgewijs-lineair model voor het statische gedrag van de deep submicron MOS-transistor. Om de werking van schakelingen op transistorniveau te kunnen analyseren is minstens dit zogenamde ”collapsible current source” model nodig. Verder gebruiken we gemiddelde condensatorcapaciteiten om de ”terminal charge” variaties in de MOS-transistor te modelleren in het geval van dynamische groot-signaal werking.

We tonen aan dat deze compacte manier van modelleren van de statische en dynamische werking van een deep submicron MOS-transistor een betrouwbare en degelijke methode is voor signaalvormschattingen en vermogensopname van digitale deep submicron CMOS-schakelingen.

Onze analyse van zeer grote en complexe deep submicron schakelingen is efficiënt door het gebruik van dit compacte MOS-transistormodel in het raamwerk van een event-driven simulator. We introduceren een VHDL package dat we ontwikkeld hebben om de continue werking van een CMOS-schakeling te representeren in een event-driven simulator.

We hebben onze methode in de praktijk gebruikt om grote schakelingen te verifiëren. We verkrijgen zeer nauwkeurige uitkomsten, vergelijkbaar met de resultaten van klassieke modellering op transistorniveau, maar er zijn veel minder berekeningen nodig. Onze methode opent de deur naar simulatie op diverse abstractieniveaus, d.w.z. digitale schakelingen kunnen tegelijkertijd met bijna-analoge transistormodellen door een event-driven simulator worden verwerkt.

Keywords: deep submicron, multi-level simulation, level transparency, event-driven, piecewise linear, collapsible current source model, average capacitors model, CMOS