MIXED CIRCUIT AND DEVICE SIMULATION FOR ANALYSIS, DESIGN, AND OPTIMIZATION OF OPTO-ELECTRONIC, RADIO FREQUENCY, AND HIGH SPEED SEMICONDUCTOR DEVICES

A DISSERTATION
SUBMITTED TO THE DEPARTMENT OF ELECTRICAL
ENGINEERING AND THE COMMITTEE ON GRADUATE STUDIES
OF STANFORD UNIVERSITY
IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE DEGREEE OF
DOCTOR OF PHILOSOPHY

Francis M. Rotella

April 2000

© Copyright 2000
by Francis M. Rotella
All Rights Reserved

I certify that I have read this dissertation and that in my opinion it is fully adequate, in scope and quality, as a dissertation for the degree of Doctor of Philosophy.

Robert W. Dutton (Principal Adviser)

I certify that I have read this dissertation and that in my opinion it is fully adequate, in scope and quality, as a dissertation for the degree of Doctor of Philosophy.

S. Simon Wong (Associate Advisor)

I certify that I have read this dissertation and that in my opinion it is fully adequate, in scope and quality, as a dissertation for the degree of Doctor of Philosophy.

Arogyaswami J. Paulraj

I certify that I have read this dissertation and that in my opinion it is fully adequate, in scope and quality, as a dissertation for the degree of Doctor of Philosophy.

Zhiping Yu

Approved for the University Committee on Graduate Studies:

Abstract

Technology Computer Aided Design tools have played a vital role in the development of next generation semiconductor devices. Unfortunately, modern high speed devices can no longer be studied without considering parasitic components and biasing circuitry. These factors are especially significant for opto-electronic, radio frequency, and high speed devices whose performance depends on their interconnect, layout, and parasitic components.

This work describes the application of device simulation to opto-electronic and high frequency device design. It begins with a discussion of the tool development for such simulations. Two approaches are addressed. In the first approach, the device simulation is included in the circuit simulation as a numerical device model. The previous work of others is extended, addressing larger problems in terms of the number of devices, complexity of the devices, and parallel execution algorithms. A second approach reduces a linear circuit to a set of boundary conditions applied to the device simulator and thus allows for the inclusion of parasitic components during dc, ac, and transient analysis. In addition to circuit boundary conditions, an harmonic balance solver is integrated with the device simulator to allow for large signal RF simulation.

With these improvements to a device simulator such as PISCES, examples are provided to demonstrate the modeling methodology and tool capabilities. LED's for optical communication systems are studied and optimized. The impact of the layout is

characterized for a MESFET used in RF communication circuits. The physical phenomena of poly-depletion in digital CMOS scaling is studied through mixed circuit and device simulation.

An in-depth example explores the analysis, modeling, design, and optimization of RF LDMOS transistors. A brief review of the device operation is provided and key modeling regions and methodologies are explored. The model is verified using measured I-V data and C-V characteristics and the RF simulation accuracy is evaluated by generating curves for gain, efficiency, and linearity. With a proven RF model, the transistors's RF performance is studied, the effect of parasitic components is analyzed, and design optimization is considered.

Acknowledgments

I would like to take this opportunity to acknowledge those who have provided the help and guidance to complete my Ph.D. at Stanford. First and foremost I would like to thank my adviser Dr. Robert Dutton for his support and guidance throughout my stay at Stanford. Dr. Zhiping Yu provided many fascinating and intricate technical discussions on device simulation and I appreciate his time and patience. Dr. Simon Wong taught some of the first classes that I attended and from there became my associate adviser for which I'd like to thank him. I'd also like to thank Dr. Kunle Olukotun for chairing my Oral's defense committee and appreciate his time on short notice. I'd like to thank Dr. Arogyaswami Paulraj for reviewing my dissertation and appreciate his feedback.

This work would not have been possible without industrial contacts and interactions. I'd like to thank Chun Lei and K.L. Chen of Hewlett Packard, Kaoru Inoue of Matsushita Electric Industrial Co., Ron Knepper, Steve Furkay, and Jeff Johnson of IBM, and Gordon Ma of Motorola. All provided examples and support to drive this work and I hope that they have benefited from the results as much as I have benefited from our discussions. Funding for this work came from AASERT contract DAAH04-93-G-0178 on the parent Army contract DAAL03-91-G-0152 and Semiconductor Research Corporation under contract SRC 93-SJ-116.

Many in the TCAD group provided technical discussions, help, and support throughout my stay at Stanford. I'd like to especially acknowledge Boris Troyanovsky for his help in working with his harmonic balance solver. In addition, I'd like to thank Richard Williams,

Aon Mujtaba, Zakir Sahul, Xin-Yi Zhang and Edward Chan for the many discussions we had in the lab. Outside the TCAD group, Sunderarajan Sunderesan Mohan, besides being a good friend, provided many interesting discussion on RF circuits. Most importantly, I couldn't end this section without acknowledging Rex Lowther of Harris Semiconductor who introduced me to TCAD over 12 years ago.

Others at Stanford who provided support include Fely Barrera and Maria Perea both of whom were incredibly helpful in negotiating the mine field that is the Stanford bureaucracy. I'd like to thank Steve Hansen, Chris Quinn, and Todd Atkins for their support of the computer systems and would like to especially acknowledge Dan Yergeau. My goal in life should be to find a computer problem that he can't fix.

Finally, I'd like to mention some of the people whom I've met at Stanford and who provided the emotional support to get through the tough times. Specifically, I'd like to thank Mitch Oslick for being willing to listen when I needed someone with whom to talk. I'd like to acknowledge Peter and Nancy Smeys who have been good friends since my first week at Stanford and Steve Jurichich who always had a very calming effect when I talked with him. I'd also like to acknowledge my friends John Pennington and Laurie Addleman who I met outside the confines of Stanford University and appreciate them putting up with a poor graduate student in their lives.

There are many others who have had an impact on me since coming to Stanford. Rather than trying to list them all and potentially forgetting someone, let me just say I have many wonderful memories from this time period.

To all the teachers in the world . . .

especially

Carole Elder

Nazif Tepedelenlioglu

Table of Contents

Abstract iv

Acknowledgments vi

Table of Contents ix

List of Tables xiii

List of Figures xiv

Chapter 1: Introduction 1

1.1 Description 1

1.2 Motivation 2

1.3 Summary of Previous Work 3

1.4 New Contributions 4

1.5 Overview of Chapters 5

Chapter 2: Mixed Circuit and Device Simulation 7

2.1 Generic Mixed Circuit and Device Simulation 7

2.2 The Two-Level Newton Approach 7

2.2.1 Description 7

2.2.2 Circuit Equations 8

2.3 Computing Device Currents for Two-Level Newton 10

2.3.1 Semiconductor Equations 10

2.3.2 Discretization and Solution Techniques 11

2.3.3 Physical Models 11

2.4 Computing dI/dV and dQ/dV for Two-Level Newton 12
2.4.1 The Low Frequency Approximation 12
2.4.2 Solution for ΔI During DC Analysis 15
2.4.3 Solution for ΔI during Transient Analysis 17
2.4.4 Verification 19
2.5 Modular Interface for Two-Level Newton 23
2.6 Parallel Execution for Two-Level Newton 26
2.7 Full-Newton Approach 27
2.8 Computational Cost of Mixed Circuit and Device Simulation 27
2.9 Benchmark Circuits 29
2.10 Mixed Circuit and Device Simulations for Large Problems 31
2.11 Conclusions 34

Chapter 3: Boundary Conditions for Device Simulation 35
3.1 Description 35
3.2 Device Simulation with Parasitic Components 35
3.3 Types of Boundary Conditions 37
3.3.1 Voltage Boundary Conditions 38
3.3.2 External Resistance and Capacitance 41
3.3.3 Distributed Contact Resistance 42
3.3.4 Contact Resistance with External Resistances 43
3.3.5 Current Boundary Condition 44
3.4 Generic Circuit Boundary Condition Equations 45
3.4.1 Formulating the Boundary Condition Equations 45
3.4.2 Implementing Circuit Boundary Conditions 47
3.4.3 Multiple Circuit Boundary Conditions 47
3.5 Small Signal AC Analysis 48
3.5.1 Description 49
3.5.2 Boundary Conditions 49
3.6 Transient Analysis 50
3.6.1 Description 50
3.6.2 Boundary Conditions 51
3.7 Harmonic Balance Analysis 56
3.7.1 Description 57
3.7.2 Boundary Conditions for Harmonic Balance Analysis 58
3.7.3 Integration of HB Solver and Circuit BC 59
3.8 Conclusions 60

Chapter 4: Opto-Electronic, RF, and High Speed Component Optimization 61

4.1 Introduction 61

4.2 Design of LED with Floating Layer 62

4.2.1 Analysis of Original Structure 62

4.2.2 Design Optimization 66

4.2.3 LED Communication Circuit 68

4.3 Impact of Layout on RF MESFET Performance 70

4.3.1 Design Requirements 70

4.3.2 Device and Layout Optimization 72

4.4 Switching Speed in CMOS Inverters 75

4.4.1 Problem Description 75

4.4.2 Impact of Non-Degenerately Doped Poly Silicon 75

4.4.3 Design Optimization 77

4.5 Conclusions 81

Chapter 5: Modeling and Simulation of RF MOS Transistors 82

5.1 Motivation 82

5.2 Device Structure and Operation 82

5.3 Device Model and Parasitic Component Analysis 85

5.3.1 Intrinsic Device Structure 86

5.3.2 Layout Parasitic Components 88

5.3.3 Contact Pads 90

5.3.4 Packaging 91

5.4 DC and AC Analysis 92

5.4.1 I-V Characteristics 92

5.4.2 C-V Characteristics 94

5.4.3 S-Parameters 98

5.5 RF Characterization 98

5.5.1 Large Signal Distortion 100

5.5.2 Impact of Parasitic Components 102

5.5.3 Inter-modulation Distortion 104

5.5.4 Matching Networks 106

5.6 Conclusions 108

Chapter 6: Conclusions 109

6.1 Summary of Contributions 109

6.2 Future Work 110

Appendix A: Users Manual for Mixed-Mode Simulation 112
A.1 Introduction 112
A.2 New SPICE Cards 112
A.3 Running a Mixed-Mode Simulation 123
A.3.1 Files Required and Generated 123
A.3.2 Method and Model Parameters 125
A.3.3 Types of Execution 126
Appendix B: Examples from Mixed-Mode Manual 128
B.1 Introduction 128
B.2 Single Stage CMOS Inverter (DC Analysis) 128
B.3 High Frequency GaAs MESFET (AC Analysis) 129
B.4 SRAM Cell (Transient Analysis) 130
B.5 GaAs/AlGaAs LED (Transient Analysis) 133
Appendix C: Users Manual for PISCES2-HB 138
C.1 Introduction 138
C.2 PISCES Card Additions 138
C.3 The Circuit File 148
C.4 Node Numbers 167
C.5 Scaling Factors 167
Appendix D: Examples from PISCES2-HB Manual 168
D.1 Description 168
D.2 Circuit Boundary Conditions 168
D.2.1 Diode with External Circuit Components and Distributed Contact Resistance 168
D.2.2 Transient Response of a BJT Inverter 170
D.2.3 Gain vs. Bias Condition for a BJT Amplifier 172
D.3 Harmonic Balance Analysis 173
D.3.1 Large Signal Analysis of an AM Demodulator 173
D.3.2 BJT Mixer to Down Covert a 2MHz Signal to 100KHz 175
D.3.3 Analyzing Gain and Efficiency in an MOS Power Amplifier 176
References 179

List of Tables

Table 2.1: Resource statistics for a parallel mixed-mode simulation of an SRAM cell using the full-Newton algorithm. 30

Table 2.2: CODEC simulation comparison for two-level Newton and full-Newton simulations. 32

Table 3.1: Boundary condition equations for a generic circuit surrounding a PISCES device. 48

Table 4.1: Simulated experiments to study the impact of design parameters on the n-type AlGaAs layer in a new LED structure. 68

Table A.1: Model Types for Mixed-Mode Simulations. 117

Table A.2: Device Configurations for Mixed-Mode Simulation. 118

List of Figures

Figure 2.1: Comparison of matrix formation for mixed circuit and device simulation using loosely and tightly coupled algorithms. 8

Figure 2.2: Evaluation of dI_i/dV_j for i equal to 1 through 4 (drain, gate, source, and bulk of a (MOSFET) and j equal to 1 or 2 during a transient step from t_o = 50ps to $t_o + \Delta t$ = 55ps. . All contacts except the swept contact are held at constant bias. 21

Figure 2.3: Evaluation of dQ/dV for a diode during a transient step from t_o = 150ps to $t_o + \Delta t$ = 155ps during which the diode is in the process of being turned "on". 22

Figure 2.4: Block diagram for interfacing a numerical device simulator like PISCES into a circuit simulator like SPICE. It becomes another block in similar manner to an analytic model. 23

Figure 2.5: The modular interface for the two-level Newton algorithm for mixed circuit and device simulation. 24

Figure 2.6: Parallel execution algorithm for two-level Newton mixed-mode simulations. 26

Figure 2.7: Circuit diagram and a read/write cycle for a SRAM cell and its surrounding control circuitry. 29

Figure 2.8: CPU Time for simulating a read/write cycle of an SRAM cell. 30

Figure 2.9: CPU time comparison of the two-level Newton and full-Newton algorithms for (a) a one inverter and (b) a two inverter chain for different number of nodes per inverter.. 33

Figure 2.10: CPU time comparison of the two-level Newton and full-Newton algorithms for a generic mixed circuit and device simulation of an inverter chain.. . 34

Figure 3.1: Types of problems involving mixed circuit and device simulation.. 36

Figure 3.2: Methods for solving a mixed circuit and device problem for a single device surrounded by linear elements. 36

Figure 3.3: Resistive boundary conditions for a numerical device structure. (a) Plain voltage boundary condition. (b) External resistance. (c) Distributed resistance. (d) Combined distributed and external resistance. 38

Figure 3.4: (a) Linear circuit surrounding a numerical device. (b) For the purpose of finding the boundary conditions, the numerical device is assumed to consist of ideal current sources of unknown values.. 45

Figure 3.5: Coupling methods for Y-parameters of the device simulator with the small signal parameters of the circuit by (a) combining Y-parameters of each simulation block and (b) using the PISCES Y-parameters in a small signal simulation.. 50

Figure 3.6: Time discretization of a capacitor using backward Euler. 51

Figure 3.7: Time discretization of an inductor using Backward Euler.. 53

Figure 3.8: Local linearization of capacitors and inductors using TR/BDF2 time discretization. 55

Figure 3.9: Multi-tone response of diode with an input of two closely spaced frequencies [41]. 57

Figure 3.10: Integration of PISCES with a harmonic balance library and a circuit boundary condition library.. 59

Figure 4.1: Cross section of original LED structure.. 63

Figure 4.2: Measured and simulated I-V characteristics for heterostructure LED with a semiconductor floating layer to improve carrier confinement.. 65

Figure 4.3: Measured optical profile for LED structure.. 65

Figure 4.4: Integrate radiative recombination in the InGaAs region of the simulated LED structure.. 66

Figure 4.5: Cross section of the redesign of the LED structure. 67

Figure 4.6: Current flow lines for two structures of a re-design of an LED structure for (a) low resistance n-AlGaAs layer and (b) high resistance n-AlGaAs layer. 68

Figure 4.7: (a) Simulated recombination profile in active layer of new LED structure compared to the original structure and (b) the experimental optical output of the new structure. 69

Figure 4.8: Fiber optic transmitter circuit 69

Figure 4.9: LED response of a transmitter circuit: (a) I-V characteristics and (b) summed radiative response. 70

Figure 4.10: Cross-section of MESFET used in RF communication circuits. 71

Figure 4.11: Layout of MESFET used in RF communication circuits.. 71

Figure 4.12: 3D solid model of the gate region of an RF MESFET. 72

Figure 4.13: Model for MESFET layout with intrinsic device and external parasitic components 72

Figure 4.14: Comparison of MesFet performance (a) with and (b) without parasitic components. 73

Figure 4.15: Comparison of the (a) input impedance of a MESFET (s11) and (b) the gain of MESFET (s21) for a constant gate width of 100mm using a different number of fingers.. 74

Figure 4.16: Anomalous I-V characteristics caused by poly silicon depletion in the gate of a MOSFET for a channel doping of 5×10^{17} and various silicon gate doping levels (N_p). 76

Figure 4.17: Physical representation of poly-depletion in a MOS structure. 77

Figure 4.18: Circuit diagram of ring oscillator used for simulation.. 78

Figure 4.19: Cross sectional doping profile and mesh used for a simplified transistor to analyze poly-silicon depletion effects. 79

Figure 4.20: SImulated response of 5 stage ring oscillator in structure with different doping in the poly-silicon 80

Figure 4.21: Simulated poly-silicon gate doping profiles for the (a) NMOS and (b) PMOS FET. 81

Figure 5.1: Typical cross-section of an LDMOS device. 83

Figure 5.2: Comparison of the longitudal electric field in the channel of (a) an LDMOS device versus (b) a standard MOS device. The vertical lines indicate the edges of the channel region and the horizontal line is the approximate electric field (2 x 10^4 V/cm) when electrons reach velocity saturation. . . 84

Figure 5.3: Top view of a typical layout for long gate LDMOS device. 85

Figure 5.4: Model for an LDMOS device with parasitic components. 86

Figure 5.5: Current flow lines through the P^+ sinker and the substrate of an LDMOS device with a backside contact.. 88

Figure 5.6: Source resistance as a function of device width. 89

Figure 5.7: Top view and side view of pad structures along with a representative model. 91

Figure 5.8: Design of experiments to determine optimum 2D doping profile. 92

Figure 5.9: Comparison of simulated and measured I-V characteristics for the modeled LDMOS device. 93

Figure 5.10: Measured and simulated capacitance for Cdg and Cds. 95

Figure 5.11: Measured and simulated values for the input capacitance (C_{gs}) with the drain floating, grounded, and included in the measurement. The numbers correspond to different states of the channel. 96

Figure 5.12: Plots of accumulation, depletion, and inversion as each tracks across the channel from the low doped drain side to a high doped source side of the channel region. 97

Figure 5.13: LDMOS impedances looking into the device (a) with and (b) without parasitic components. 99

Figure 5.14: LDMOS gain (a) with and (b) without parasitic components. 99

Figure 5.15: Configuration of a power amplifier. The biasing networks set the load lines and the matching networks provide for transfer of RF power. 100

Figure 5.16: Simulated and measured RF responses for gain and efficiency of a power amplifier. 101

Figure 5.17: (a) Time domain signal and (b) frequency spectrum of the drain voltage in an harmonic distortion large signal sinusoidal simulation. The larger magnitude of the signal indicates increasing power along the gain curve. 102

Figure 5.18: Simulated effects of parasitic components on the gain of the LDMOS device. The "+" and "-" indicate whether the variable is at it's manufacturing high or low value. 103

Figure 5.19: Simulated effects of parasitic components on the efficiency of the LDMOS device. The "+" and "-" indicate whether the variable is at it's manufacturing high or low value. 104

Figure 5.20: Frequency domain response of a non-linear system given an input of two tightly spaced tones. 105

Figure 5.21: Simulated and measured inter-modulation distortion of the LDMOS power amplifier. 105

Figure 5.22: Experimental load-pull analysis. The white point represents the matching network used in the earlier RF response. 107

Figure 5.23: Simulated load-pull analysis. The white point represents the matching network used in the earlier RF response. 107

Figure B.1: CMOS inverter circuit schematic and netlist. 129

Figure B.2: Two port network and netlist for ac analysis to find the Y-parameters 130

Figure B.3: S-parameters for GaAs MESFET in the frequency range of 1GHz to 40GHz. 131

Figure B.4: Simplified circuit diagram of SRAM cell including control circuitry. . . 131

Figure B.5: SPICE netlist for SRAM simulation. 132

Figure B.6: Continuation of SPICE netlist for SRAM simulation. 133

Figure B.7: Read/Write cycle for SRAM cell. 134

Figure B.8: Cross section of LED structure that is used in a fiber optic transmitter circuit. 134

Figure B.9: Circuit schematic and netlist for LED transmitter simulation. 135

Figure B.10: Time transient response of the current flowing through LED in a transmitter circuit. 137

Figure C.1: One T-section of a transmission line defined with N sections.. 152

Figure D.1: (a) Circuit diagram for diode with extrinsic bulk resistance and distributed contact resistance. (b) The IV response of the structure shows a large leakage current generated by the shunt resistance. 169

Figure D.2: Current flow lines for a diode with (a) a low contact resistance and (b) with a large contact resistance. 169

Figure D.3: (a) PISCES input deck for dc sweep of diode surrounded by resistive network with a distributed contact resistance. (b) Netlist describing circuit diagram. 170

Figure D.4: Circuit diagram of a simplified BJT inverter and the response of that inverter given a pulse input. 171

Figure D.5: (a) PISCES input deck for transient simulation of a BJT inverter. (b) Netlist describing circuit diagram. 171

Figure D.6: (a) Circuit diagram for a BJT amplifier with (b) the small signal response for different bias conditions on the gate. 172

Figure D.7: (a) PISCES input deck for determining small signal gain versus bias condition of BJT amplifier. (b) Netlist for circuit and analysis limits.. . 173

Figure D.8: Circuit diagram of an AM demodulator. A 5KHz signal on a 5MHz carrier is fed into the demodulator. 174

Figure D.9: (a) PISCES input deck for analyzing an AM demodulator. (b) Netlist describing circuit diagram and analysis limits.. 174

Figure D.10: (a) A BJT Mixer that down converts a 200MHz to a 100KHz signal. (b) The noise in the 100kHz output signal. 175

Figure D.11: (a) PISCES input deck for analyzing the BJT Mixer. (b) Netlist describing circuit diagram and analysis limits. 176

Figure D.12: Cross section of an LDMOS transistor with parasitics represented by lumped elements. 177

Figure D.13: Gain and power added efficiency (PAE) for LDMOS power amplifier. 177

Figure D.14: (a) Time domain voltage and (b) spectrum of the voltage on the drain of the LDMOS transistor in a power amplifier as the device enters gm compression. 178

Chapter 1: Introduction

1.1 Description

Over the past decade and a half, device simulation has proven to be an important design tool for new semiconductor technology. It provides an environment for design engineers to "experiment" with different structures by providing insight into the performance of the physical device structure. It allows one to look inside the device to examine the physical response of the device as opposed to only looking at the external electrical response. In addition to helping the design engineer, research in device simulation has lead to studies and modeling of advance physical phenomenon in order to obtain better simulation results which in turn result in a better understanding of device performance [1].

This work extends the capabilities of device simulation by studying, characterizing, and improving the boundary conditions around the physical device simulation so that advance opto-electron, radio frequency (RF), and high speed components can be accurately modeled. The boundary conditions determine the solution for the device and are required to model the outside electronic world accurately in order to provide the designer with a reliable real world response. The information presented in this work will address these boundary conditions in a mixed circuit and device simulation where device physics play a crucial role in high frequency effects in radio frequency devices, radiative recombination in opto-electronic devices, and switching speeds in high speed digital CMOS.

1.2 Motivation

The motivation behind this research is to provide the design engineer with the capability to analyze, characterize, and develop a new or novel device in the circuit environment in which it will operate. Hence, a smaller design space is established before expensive wafers (in terms of cost and time) are manufactured.

Advances physical effects in semiconductor devices are not easily modeled in circuit simulation or require non-standard configurations so as to be able to model the effects [2]. These physical effects often include things related to niche devices targeted for a specific market. For example, the key component of opto-electronic devices is the photon generation due to direct recombination mechanisms. Some models do exist for such processes, yet they are of no benefit to the device engineer because they are often based upon experimental measurements. Device simulation itself is more useful because the actual recombination process can be tracked while simultaneously solving for the electrical response.

Another key area where device physics play a major role is that of RF devices. RF simulation is a relatively new field and circuit models are generally derived from analog counter parts [3]. The Root model is one of the common models, yet it is only an empirically based, requiring experimental data in order to properly calibrate and use [4] [5]. The device engineer interested in developing next generation technology, does not have silicon from which to work. Hence, a mixed circuit and device simulation capability allows the designer to place the device in the targeted RF circuit application and optimize based upon the required configuration. In addition, coupling harmonic balance simulation with circuit components increases the analysis capabilities.

Physical effects can change the performance of standard digital CMOS technology. For example, as geometry shrinks the electrically active doping in the poly-silicon gate can decrease because of the reduction in the thermal budget while at the same time the channel doping has been increasing [6]. These changes result in poly-depletion which in turn affects the high speed performance of the silicon devices. Device trade-offs must be accessed in the circuit domain yet process simulation is required to ascertain doping profiles.

Parasitic components around the device have a very strong impact on its performance and thus are needed in the device simulation to provide an accurate evaluation of the device

performance. The device engineer can no longer work solely on the intrinsic device because the scaling of such devices has increased the contributions of parasitic components that ultimately degrade the performance. Hence, the device engineer has to design the structure in such a way that the parasitic components are minimized while the geometry of the device is optimized.

During circuit analysis, a design engineer can study the internal structure of the device to evaluate the physics and thus be provided with information on how physical changes in the device affect circuit performance. This ability is one of the powerful aspects of the mixed circuit and device environment. For example, the design engineer can examine charge generation points, charge distribution, and electric fields to determine where hot carriers problems can originate and then make changes in doping profiles or the structure to improve performance [7].

As devices become smaller and more complex, previously neglected physical effects become more prevalent. Initiatives, like BSIM3, have focused on developing models to incorporate these scaling effects, but these models are intricately complex [8]. BSIM3 contains over 100 parameters which all must be extracted in order to be used in a circuit simulation. This extraction process can be prohibitively expensive in order to accurately characterize an important physical effect even if device simulation is used to generate the I-V curves. Since the concerns of the device engineer are limited to the circuit block level (*i.e.* never involving more than a handful of devices) a mixed circuit and device simulation provides a reasonable alternative to optimize a new structure or limit the design space for a new technology.

Given the many advantages of a mixed circuit and device analysis environment, this work endeavors to develop a methodology to model, analyze, and optimize devices that are affected by physical effects that are not well modeled in standard circuit simulation.

1.3 Summary of Previous Work

A early version of mixed circuit and device simulation was developed in the mid 1970's at the University of Aachen where Engl, developed a simulator for modular circuits: MEDUSA [9]. MEDUSA solved modules of sub-systems which in turn were used to solve the entire system; thus allowing large circuits to be simulated with the limited computational power of the times. Each subsystem represented subcircuits of varying complexity from a description by algebraic equations (*i.e.* Ohm's Law) to that of the

partial differential equations (*i.e.* Poison's Equation). MEDUSA has been used to solve a mixed circuit and device problem by coupling the Gummel-Poon compact model with that of a one dimensional physical simulation for a bipolar transistor.

In the 1980's Greg Rollins used a tightly coupled matrix approach to include the circuit equations in the matrix system of the device simulator [10]. This approach is discussed in Chapter 2 and has been used industrially by the TCAD vendors Silvaco and Technology Modeling Associates (now part of Avant!) [11] [12]. In his dissertation, Rollins applied mixed circuit and device simulation to single event upset in a static bipolar RAM circuit [13].

Mayaram made a comparison of different methods to solve the mixed circuit and device problem [14]. He compared and contrasted three methods (including that of Rollins) based on the mixed circuit and device simulator called CODEC which was later expanded into the design system CIDER by Gates [15]. Mayaram's results are discussed in detail in Chapter 2 and compared with those in this work for the complexity of problems encountered with opto-electronic, RF, and high speed devices.

1.4 New Contributions

The work in this dissertation expands upon the ideas developed over the last two decades. A modular mixed circuit-device simulator is developed and demonstrated to work with a variety of different device simulators. Previous work and those tools provided by vendors only allow one type of device simulator and circuit simulator to be coupled.

With a generalized simulator, a parallel version becomes possible. Namely, each module can be relegated to a node of a parallel machine or as demonstrated in this work, to one computer in a network of computers. Hence, the scope of the simulation capabilities are expanded to larger dimension problems.

Another major contribution of this work is that the types of problems presented are more realistic, larger, and complex. Mayaram did comparisons of different methods, but if the problems are scaled up in terms of the number of devices, nodes per device, and complexity of the circuits, the conclusions from Mayaram's analysis can vary markedly. To demonstrate this impact, some critical real world examples are presented.

In the later chapters, a more compact approach to mixed circuit and device simulation is presented for linear circuits. A linear circuit can be reduced to a set of equations which are

applied as boundary conditions in the device simulator. This approach allows for the simulation of devices with parasitic components extracted from the layout of the device. Using three dimensional solid modelers, the component values can be determined and the entire device and layout can be optimized [16].

Coupling device simulation with an harmonic balance solver [17] opens an important class of problems for device analysis, especially when coupled with circuit boundary conditions. The modeling of an LDMOS power device is presented from the analysis of the basic structure, the parasitic component analysis, and the full RF large signal performance. Such an approach allows for the optimization of the device for RF power applications.

1.5 Overview of Chapters

This work is divided into four chapters and four appendices. The first two chapters address algorithmic changes to a device simulator like PISCES [18] in order to meet the requirements for mixed circuit and device simulation. In Chapter 2, algorithms for mixed circuit and device simulation are described and improvements are explained. Comparisons are made between the two common approaches for a fully coupled Newton solution versus a two-level Newton approach. Chapter 3 focuses on boundary conditions for device simulation. It begins with a review of the standard boundary conditions and then builds on this basis to include generic linear circuit boundary conditions. In addition, it describes how the boundary conditions change for different types of analysis such as dc, ac, transient, and harmonic balance.

Chapter 4 and Chapter 5 include examples for coupled mixed circuit and device simulation. The three examples in Chapter 4 address: the design of an opto-electronic device containing a floating layer that requires special boundary conditions; the optimization of an RF device whose layout adds parasitic components that affect performance; and the performance limitation of high speed digital CMOS through the physical effect of poly-silicon depletion. An in-depth RF example is provided in Chapter 5. This example explores the modeling, analysis, and optimization of a LDMOS structure used for power amplifier applications.

The four appendices contain selected material from the user's manual provided with the prototype computer code developed for mixed circuit and device simulation. Appendix A and Appendix B contain the user's manual and example chapter from the mixed circuit

and device simulation tool [19]. Appendix C and Appendix D contain the user's manual and example chapter from the harmonic balance simulator with generic linear circuit boundary conditions [20].

Chapter 2: Mixed Circuit and Device Simulation

2.1 Generic Mixed Circuit and Device Simulation

Mixed circuit-device analysis allows for the simulation of advanced device and physical effects not available in standard circuit simulations [21]. Such effects include short channel effects in MOSFET's, high frequency (*i.e.* RF) effects in MOSFET's and MESFET's, and temperature effects in high power devices. In addition, during device design phases of development, mixed-mode simulation can be used to test simple circuits to verify that the design meets the required specifications without actually extracting model parameters.

This chapter provides a detailed discussion of the full-Newton approach to mixed circuit and device simulation. The approach is described in generic terms and then dissected into its main components. A modular interface is described and is developed into a parallel execution method. The two-level Newton approach is compared to the full-Newton approach and shown to be better for large problems.

2.2 The Two-Level Newton Approach

2.2.1 Description

Unlike the current commercial mixed-mode simulators which use tightly coupled device and circuit matrices [11] [12], this work has taken the approach of a loosely coupled algorithm. Figure 2.1 graphically shows the difference between two approaches. In the

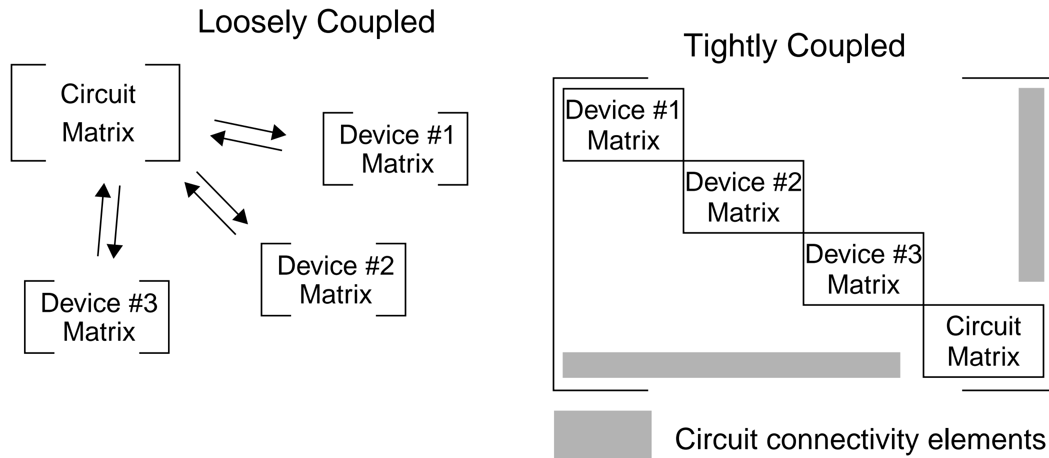


Figure 2.1: Comparison of matrix formation for mixed circuit and device simulation using loosely and tightly coupled algorithms.

tightly coupled approach the circuit equations are coupled with the device equations and the entire system is solved simultaneously. In a loosely coupled approach, the circuit matrix is self contained and the device matrix is evaluated at a lower iteration level for each circuit iteration. This chapter explores the details of the loosely coupled algorithm and contrasts that algorithm with the tightly coupled matrix approach.

2.2.2 Circuit Equations

A typical circuit simulator solves the non-linear circuit equations by a modified form of Newton-Raphson (NR) [22]. The non-linear system of equations for the circuit can be represented as shown in the following equation according to the Kirchoff's current law:

$$F(V) = 0 \tag{2.1}$$

where V is the vector of node voltages and F represents the sum of the currents into each node in the circuit, and both V and F have dimension of N, the number of nodes in the circuit.

Applying Newton-Raphson (NR) to the above equation yields the linear matrix system

$$V^{(i+1)} = V^i - J^{-1}(V^i)F(V^i) \tag{2.2}$$

where $J(V)$ is the Jacobian given as

$$J(V) = \begin{bmatrix} \frac{\partial F_1}{\partial V_1} & \frac{\partial F_1}{\partial V_2} & \cdots & \frac{\partial F_1}{\partial V_N} \\ \cdots & & & \cdots \\ \frac{\partial F_N}{\partial V_1} & \frac{\partial F_N}{\partial V_2} & \cdots & \frac{\partial F_N}{\partial V_N} \end{bmatrix} \tag{2.3}$$

and the index i is the iteration count during NR iterations.

For each Newton iteration, the previous voltage V^i is known and hence, $V^{(i+1)}$ can be computed. A circuit interpretation of Equation 2.2 and the Jacobian matrix (Equation 2.3) can be made as follows. Because each Jacobian element has units of the conductance, hereafter G replaces J. By multiplying G on both sides of Equation 2.2 from the left, one obtains the following set of linear equations at $(i+1)$-th iteration where i starts from 0:

$$G^i V^{(i+1)} = G^i V^i - F^i \tag{2.4}$$

The matrix G^i and vector F^i are evaluated at V^i. The above equation represents a linear circuit as far as node voltages $V^{(i+1)}$ because both the coefficient G^i and the source term (*i.e.* the right hand side (RHS) of the above equation) are independent from $V^{(i+1)}$. Furthermore, G^i can be interpreted as the linear conductance components, including both the linear components in the original circuit and differential conductance at V^i, in the equivalent circuit. $G^i V^i - F^i$ can be viewed as the current sources.

The NR iteration is terminated when convergence is reached meaning that the change in V between two consecutive iterations is smaller than a predefined tolerance. In SPICE, it is required that the current change in each circuit branch is also below a certain criterion when the convergence is considered to be reached [23].

Transient analysis is performed in a similar manner. For each time step, Newton-Raphson iterations are performed until convergence is met. In addition, the truncation error due to the time discretization is checked to determine if the time step is acceptable in terms of accuracy. If this error is too large, the time step is reduced and the computation is repeated.

For ac analysis, the dc solution is first sought and then the non-linear devices are linearized at the solution. This small signal equivalent circuit is used to find the ac response given an ac excitation vector.

2.3 Computing Device Currents for Two-Level Newton

According to the algorithm for the circuit iterations, two pieces of information are required from the device simulator. One is the current flowing through the device for a set of specified boundary conditions (V^i) and the second is the derivative of the current and charge with respect to those voltage changes on the electrodes (G^i). In this section, the basics of device simulation are discussed and characterized for the purpose of mixed circuit-device simulation. There have been a large number of works written about device simulation and this work has no intention of duplicating the completeness presented in those discussions [24] [25]. In the subsequent section, a detailed discussion for the calculation of the derivatives is given since a complete description is not provided in the literature.

2.3.1 Semiconductor Equations

The basic form of the semiconductor equations are

$$\nabla \cdot (-\varepsilon \nabla \Psi) = q(p - n + N_D^+ - N_A^-) \ , \tag{2.5}$$

$$\frac{\partial n}{\partial t} = \frac{1}{q} \nabla \cdot J_n - U \ , \tag{2.6}$$

$$\text{and } \frac{\partial p}{\partial t} = \left(-\frac{1}{q}\right) \nabla \cdot J_p - U \tag{2.7}$$

where the current densities for holes and electrons, J_p and J_n, are given by

$$J_n = qD_n \nabla n - q\mu_n n \nabla \Psi \ \text{ and} \tag{2.8}$$

$$J_p = -qD_p \nabla p - q\mu_p p \nabla \Psi \ . \tag{2.9}$$

In this set of equations, Ψ is the electrostatic potential, n is the electron concentration, and p is the hole concentration. N_D and N_A are the active doping concentrations for donor atoms and acceptor atoms respectively and the ionization charge of those atoms are denoted with a '+' or '-' sign. The permittivity of the materials in the structure is represented by ε. The recombination within the device is modeled and represented in the variable U. The diffusivity and mobilities of electrons and holes in the device are modeled and represented as the variables D_n, D_p, μ_n, and μ_p.

2.3.2 Discretization and Solution Techniques

The semiconductor equations can be solved numerically using either finite difference, finite element or finite box techniques on a mesh defined in one, two, or three dimensions [24]. The differential equations are discretized on a mesh and a Newton-Raphson algorithm or a iterative algorithm is used to solve the generated matrix of equations. There are many sources that discuss the varied solution methods; therefore this work will only provide a generic overview for the case of a two dimensional device simulator PISCES from Stanford University [18]. For an in depth discussion, refer to the book by Selberherr or the dissertation by Pinto [24] [25].

2.3.3 Physical Models

In solving the semiconductor equations, complex equations and models represent the properties of the material (*i.e.* mobility and recombination) [26]. Unfortunately, these terms are neither constant nor simple to determine. For the most part, they are spatially dependent upon the type of material, doping of the material, electron and hole concentrations, electric field, interfaces, etc. Hence, in order for the simulation to provide accurate results, correct physical phenomena must be included in the simulation model.

A device simulator like PISCES contains a significant number of models contributed by different people and universities [27]. The reader is referred to these works to gain a more in depth understanding of the advantages and disadvantages of each model type. This work will address specific physical modeling issues for specific applications and examples (Chapter 4 and Chapter 5). For each application of mixed circuit and device simulation or RF device simulation, the key models are described and purpose for selecting that specific model is explained.

2.4 Computing dI/dV and dQ/dV for Two-Level Newton

During each circuit iteration, the simulator requires derivative information to describe how current and charge varies in response to the voltage changes on the electrodes. By applying a small ac perturbation $v = \Delta V e^{-j\omega t}$ to a linearized device solution, a current $i = \Delta I e^{-j\omega t}$ response is computed. Because the device is linearized and follows the basic circuit laws (Kirchoff's Current Law and Kirchoff's Voltage Law), it can be expressed as

$$i = Yv = \left(\frac{dI}{dV}\Delta V + j\omega\frac{dQ}{dV}\Delta V\right)e^{-j\omega t} \quad (2.10)$$

Taking the ratio of the real current to the applied voltage yields $dI/dV = \Delta I_R/\Delta V$ and likewise, using the imaginary current yields $dQ/dV = \Delta I_I/(\omega\Delta V)$; the required derivatives for each circuit iteration.

2.4.1 The Low Frequency Approximation

An algorithm has been developed for small signal ac analysis in device simulation [28]. In contrast to that approach, the frequency dependent effects are not desired. Therefore, in this section the algorithm is modified for the limit as the frequency goes to zero; thereby, eliminating the frequency dependence in Equation 2.10.

The small signal equations are represented as

$$\begin{bmatrix} J & -D \\ D & J \end{bmatrix}\begin{bmatrix} X_R \\ X_I \end{bmatrix} = \begin{bmatrix} B \\ 0 \end{bmatrix} \quad (2.11)$$

where J is the Jacobian of the device solution, X_R and X_I are the real and imaginary perturbed solutions representing $\Delta\Psi$, Δn, and Δp, and B is the magnitude of the perturbation applied to the boundary nodes (*i.e.* ΔV).

D is a diagonal matrix given by

$$D = \omega \begin{bmatrix} 0 & 0 & 0 \\ 0 & j\frac{\partial G_n}{\partial n} & 0 \\ 0 & 0 & j\frac{\partial G_p}{\partial p} \end{bmatrix} = \omega K \tag{2.12}$$

where the frequency dependent effects are factored out of the expression. F and G are real nonlinear vector functions representing Poissons equation and the two continuity equations such that

$$\begin{aligned} F_\Psi(\Psi, n, p) &= 0 \\ F_n(\Psi, n, p) &= \frac{\partial}{\partial t} G_n(n) \\ F_p(\Psi, n, p) &= \frac{\partial}{\partial t} G_p(p) \end{aligned} \tag{2.13}$$

The matrix (Equation 2.11) can be separated into the real and imaginary sub-matrices and written as

$$\begin{aligned} JX_R - DX_I &= B \\ DX_R + JX_I &= 0 \end{aligned} \tag{2.14}$$

where the two equations are coupled and can be most efficiently solved by an iterative method since a simultaneous solution doubles the matrix size.

Substituting for D and re-arranging yields

$$\begin{aligned} X_R &= J^{-1}(B + \omega K X_I) \\ X_I &= -J^{-1}(\omega K X_R) \end{aligned} \tag{2.15}$$

If a “zero” frequency limit is imposed (*i.e.* $\omega \to 0$) the $\omega K X_I$ term may be ignored in the evaluation of X_R yielding the direct solution

$$X_R = \lim_{\omega \to 0}(J^{-1}(B + \omega K X_I)) = J^{-1}B \tag{2.16}$$

which then leads to the computation of X_I as

$$X_I = -J^{-1}(\omega K X_R) = \omega \tilde{X}_I \tag{2.17}$$

where $\tilde{X}_I$ is X_I with ω factored out.

X_R and X_I contain the perturbed solutions of Ψ, n, and p. This result is used to compute the current (ΔI) at each electrode. Each operation in computing the currents ΔI_n, ΔI_p, and $\Delta I_{\text{displacement}}$ from $\Delta\Psi$, Δn, and Δp is linear because of the small signal assumption. Hence, the real and imaginary parts of the currents can be computed as

$$\Delta I = F(X_R) + j\omega F(\tilde{X}_I) \tag{2.18}$$

where $F(X)$ is a real linear function operating on X_R and X_I.

The values of the derivatives (Equation 2.10) may be computed by dividing the real and imaginary current by the perturbing voltage given by

$$\frac{dI}{dV} = \frac{Re(\Delta I)}{\Delta V} = \frac{F(X_R)}{\Delta V} \quad \text{and}$$
$$\frac{dQ}{dV} = \frac{Im(\Delta I)}{\omega \Delta V} = \frac{\omega F(\tilde{X}_I)}{\omega \Delta V} = \frac{F(\tilde{X}_I)}{\Delta V} \tag{2.19}$$

where the radial frequency cancels leaving the expressions independent of its value.

From this derivation, the following observations are apparent:

1. Because of the linearization of the entire problem, ΔV can take on any value without affecting the final solution as long as it is divided out in the final step. For simplicity, ΔV is chosen as 1.0 V.

2. Like the value for the perturbing function, ω cancels in the final step and hence, its value becomes unimportant. Letting ω go to zero removes any frequency dependence by de-coupling Equation 2.15.

The next two sections describe the specific cases of computing dI/dV and dQ/dV during dc and transient analysis, respectively.

2.4.2 Solution for ΔI During DC Analysis

The determination of dI/dV and dQ/dV in a dc simulation is presented so that a basis may be established for the evaluation of these terms during transient analysis. The operating point is first determined in order to find the Jacobian J. A perturbation, as described in Section 2.4.1, is applied to this solution in order to find X_R and X_I. From this solution the current components are computed.

Solution for Conduction Current

The perturbed current consists of a real conductive component and an imaginary displacement component for electron and holes. The calculation of the conductive component from ΔI_{nr}, ΔI_{pr}, ΔI_{ni}, and ΔI_{pi} is based upon the dc solution given as

$$I_n = A(qD_n\nabla n - q\mu_n n\nabla\Psi) \quad (2.20)$$

$$\text{and} \quad I_p = A(-qD_p\nabla p - q\mu_p p\nabla\Psi) \; . \quad (2.21)$$

The current due to the small perturbation can be computed from the real functions

$$\Delta I_n = \Delta\Psi\frac{\partial I_n}{\partial\Psi} + \Delta n\frac{\partial I_n}{\partial n} \quad \text{and}$$
$$\Delta I_p = \Delta\Psi\frac{\partial I_n}{\partial\Psi} + \Delta p\frac{\partial I_n}{\partial p} \quad (2.22)$$

for both the real and imaginary electron current and hole current respectively. The real part of ΔI is computed from the real parts of $\Delta\Psi$, Δn, and Δp found in the solution vector X_R while the imaginary part is found in X_I. Each can be computed independently by superposition because of the linearity obtained through the small signal assumption.

Solution for Displacement Current

Displacement current is proportional to the time rate of change of the electric field as

$$I_{disp} = \varepsilon\frac{\partial E}{\partial t} = \varepsilon\frac{\partial}{\partial t}(-\nabla\Psi) \qquad (2.23)$$

The original form of the small signal perturbation is required to compute the displacement currents. A small signal perturbation source of $\Delta V e^{-j\omega t}$ is applied to the contacts. Because of the linearity assumption, $e^{-j\omega t}$ carries through all the calculations and divides out when i/v is evaluated. However, the displacement current is computed from the derivative with respect to time and thus, the $e^{-j\omega t}$ term must be included when that expression is evaluated.

Applying the perturbing source to the linear system yields a value for Ψ at each node k given by

$$\Psi_k = \Psi_{k(dc)} + \Delta\Psi_k e^{-j(\omega t - \phi_k)} = \Psi_{k(dc)} + (X_{Rk} + jX_{Ik})e^{-j\omega t} \qquad (2.24)$$

where the real part of the solution can be found in the vector X_R and the imaginary part can be found in the vector X_I by solving Equation 2.15.

In order to compute E between two nodes, the difference between the solution for Ψ at node k and the solution at node l is divided by the distance between the two nodes (ΔL_{kl}) and multiplied by the flux area (A_{kl}) between the two nodes as determined by the discretization method [24]. The derivative with respect to time is then computed as

$$I_{kl(disp)} = \varepsilon\frac{\partial}{\partial t}\left(-\frac{\partial\Psi}{\partial x}\right) = -\varepsilon A_{kl}\frac{\partial}{\partial t}\left(\frac{\Psi_k - \Psi_l}{\Delta L_{kl}}\right) \qquad (2.25)$$

$$= -\varepsilon A_{kl}\frac{\partial}{\partial t}\left(\frac{(\Psi_{k(dc)} - \Psi_{l(dc)}) + (X_{Rk} - X_{Rl}) + j(X_{Ik} - X_{Il})}{\Delta L_{kl}} e^{-j\omega t}\right) \qquad (2.26)$$

$$= -\varepsilon A_{kl}\left(\frac{-j\omega(X_{Rk} - X_{Rl}) + \omega(X_{Ik} - X_{Il})}{\Delta L_{kl}}\right)e^{-j\omega t} \qquad (2.27)$$

There is no displacement current from the original dc solution and thus, the displacement current term is generated only from the perturbation.

From this internal calculation of the displacement current, the electrode currents are computed by summing over the boundary nodes. This summing process is a real linear vector operation represented by the vector operator H.

The real and imaginary portions are summed with their real and imaginary counter parts for the conduction currents for I_n and I_p. In order to compute dI/dV, the real part of the current is divided by the small signal perturbing source. Taking only the real part of Equation 2.27 and summing over the boundary nodes yields

$$Re(\Delta I_d) = -\varepsilon H\left(\frac{\omega A_{kl}(X_{Ik} - X_{Il})}{\Delta L_{kl}}\right) = -\varepsilon\omega^2 H\left(\frac{A_{kl}(\tilde{X}_{Ik} - \tilde{X}_{Il})}{\Delta L_{kl}}\right) \quad (2.28)$$

Because ω is approaching zero, ω^2 approaches zero even more quickly; hence, the real part of the displacement current goes to zero relative to the real part of I_n and I_p. As a result, the real part of the displacement current can be ignored in the computation of dI/dV.

The imaginary solution is given by

$$Im(\Delta I_d) = \varepsilon\omega H\left(\frac{A_{kl}(X_{Rk} - X_{Rl})}{\Delta L_{kl}}\right) \quad (2.29)$$

and contains a first order frequency component; however, in the evaluation of $(dQ)/dV$, ω cancels out as it does in the conduction terms.

2.4.3 Solution for ΔI during Transient Analysis

The calculation of ΔI during transient analysis is somewhat more complicated conceptually, but rather simple numerically. Suppose that the solution at t_o is known and the simulation time t_s is advanced by a small Δt. A solution at $t_s = t_o + \Delta t$ is desired. With each circuit iteration, a guess is made for the circuit node voltages (V^i). The projection for the next guess uses the derivatives of current and charge with respect to

voltage in a similar manner to the dc solution method. These derivatives represent the change in current or charge under the limited situation where time moves from t_o to $t_o + \Delta t$ and is not valid for any other time step or initial condition at t_o.

Like the dc case, the Jacobian from the device solution is used in the determination of X_R and X_I where the time function of the perturbation is separate from the time function of the simulation. ΔI is evaluated from the perturbed values of Ψ, n, and p.

Solution for Conduction Currents

The calculation of the conduction currents I_n and I_p is analogous to that for the dc analysis since there is no perturbation time dependence in the carrier current equations. Hence, I_n and I_p in the transient case is given as provided in Equation 2.22.

Solution for Displacement Current

The displacement current calculation is a bit more difficult since there is a displacement current term from the transient solution and one from the perturbation. The potential is given as

$$\begin{aligned} \Psi_k(t_o + \Delta t) &= \Psi_{k(t_o + \Delta t)} + \Delta\Psi_k e^{-j(\omega t - \phi_k)} \\ &= \Psi_{k(t_o + \Delta t)} + (X_{Rk} + jX_{Ik})e^{-j\omega t} \end{aligned} \tag{2.30}$$

for the simulation time t_s equal to $t_o + \Delta t$ and

$$\Psi_k(t_o) = \Psi_{k(t_o)} \tag{2.31}$$

for t_s equal to t_o. There is no perturbing function at time equal to t_o since the perturbing function is only applied at $t_s = t_o + \Delta t$.

Using Equation 2.23 to calculate the displacement current, given the time discretization and the solutions for Ψ at each time step, yields

$$I_{dk} = \varepsilon\frac{\partial}{\partial t}\left(-\frac{\partial}{\partial x}(\Psi)\right) = -\varepsilon A_{kl}\frac{\partial}{\partial t}\left(\frac{\Psi_k - \Psi_l}{\Delta L_{kl}}\right) \tag{2.32}$$

$$= -\varepsilon A_{kl}\left(\frac{\left(\frac{\Psi_k(t_o+\Delta t)-\Psi_l(t_o+\Delta t)}{\Delta L_{kl}}\right)-\left(\frac{\Psi_k(t_o)-\Psi_l(t_o)}{\Delta L_{kl}}\right)}{\Delta t}\right) \quad (2.33)$$

$$= I_{do(kl)} - \varepsilon A_{kl}\left(\frac{(X_{Rk}-X_{Rl})+j(X_{Ik}-X_{Il})}{\Delta t \Delta L_{kl}}\right) \quad (2.34)$$

There are two terms generated. $I_{do(kl)}$ comes from the solution to the transient step in the simulation itself. The second expression is due to the "zero" frequency perturbation and can be computed at the electrodes by applying the real vector function H.

All the currents are summed and then divided by the magnitude of the perturbing function to generate the real and imaginary contributions from the displacement current

$$Re(\Delta I_d) = -\varepsilon H\left(\frac{A_{kl}(X_{Rk}-X_{Rl})}{\Delta t \Delta L_{kl}}\right) \quad (2.35)$$

$$Im(\Delta I_d) = -\varepsilon H\left(\frac{(X_{Ik}-X_{Il})}{\Delta t \Delta L_{kl}}\right) = -\varepsilon \omega H\left(\frac{(\tilde{X}_{Ik}-\tilde{X}_{Il})}{\Delta t \Delta L_{kl}}\right) \quad (2.36)$$

The real part of the displacement current does not depend upon the value of the frequency and thus contributes to the expression of dI/dV. The imaginary portion does depend on frequency, but that frequency cancels when dQ/dV (Equation 2.10) is evaluated.

2.4.4 Verification

The computation of a zero frequency evaluation of dI/dV and dQ/dV is implement in Stanford's device simulator PISCES and in IBM's simulator FIELDAY [29] [30]. This section verifies the previously derived expressions by showing that the "zero" frequency simulation produces a correct representation of how I and Q change.

To verify the calculation of dI/dV and dQ/dV during dc analysis, a simulation with zero frequency is compared to one with low frequency (*i.e.* 0.1 to 100Hz). If the "zero" frequency assumptions are correct, the results between the two simulations should be

identical. As expected, using a low frequency simulation which fully employs Laux's algorithm [28] agreed with the "zero" frequency simulation across many device structures.

On the other hand, the transient case is a bit more difficult to verify. A "zero" simulation is compared with a divided difference estimate for dI/dV and dQ/dV. A four terminal device is used to verify the computation of dI/dV; however, because it is not possible to associate charge with a specific electrode of a semiconductor device, only a two terminal device is used in the verification of dQ/dV. In this case, positive charge can be associated with one electrode and negative charge can be associated with the other under the appropriate conditions.

To verify the computation of dI/dV, a MOSFET (metal oxide semiconductor field effect transistor) is biased with 5V on the drain and the gate is switched from 0V to 5V in 100ps. The simulation is stopped at t_s = 50ps to set the t_o state of the device during a transient analysis step. The device is stepped forward in time by a value of Δt = 5ps. During a mixed circuit and device simulation, the circuit simulator needs information on how the current changes with respect to the voltage on the gate. A "zero" frequency simulation is capable of providing this data, but it can also be estimated by applying V to the gate and $V + \Delta$ to the gate to find I and $I + \Delta$. Taking the ratio of the difference in the currents and the difference in the voltages yields an estimate of dI/dV. For verification purposes, an estimate is calculated over the range of gate voltages from 2.0V through 3.0V in 0.01V increments. In an analogous simulation experiment, the voltage on the drain is perturbed. Therefore, the gate is kept constant at 2.45V and the drain voltage is swept from 4.5V to 5.5V for a Δt of 5ps.

From these simulations, dI/dV can be calculated as $\Delta I/\Delta V$ where ΔI is the difference in the currents for two adjacent values of V and $V + \Delta V$. This divided difference result is compared with the differentiation result obtained via a "zero" frequency small signal simulation. The top four graphs of Figure 2.2 show the comparison for dI_i/dV_1 and the bottom four graphs show the comparison for dI_i/dV_2. The index "i" represents the four nodes of the device: drain, gate, source, and bulk respectively. The lines represent the "zero" frequency simulation and the points are from the divided difference calculation. With the exception of noise in the divided difference estimates, the zero frequency simulation agree quite well with those results.

The calculation of dQ/dV using divided difference is not as straight forward. There is no obvious method to associate charge with an electrode in most multi-terminal

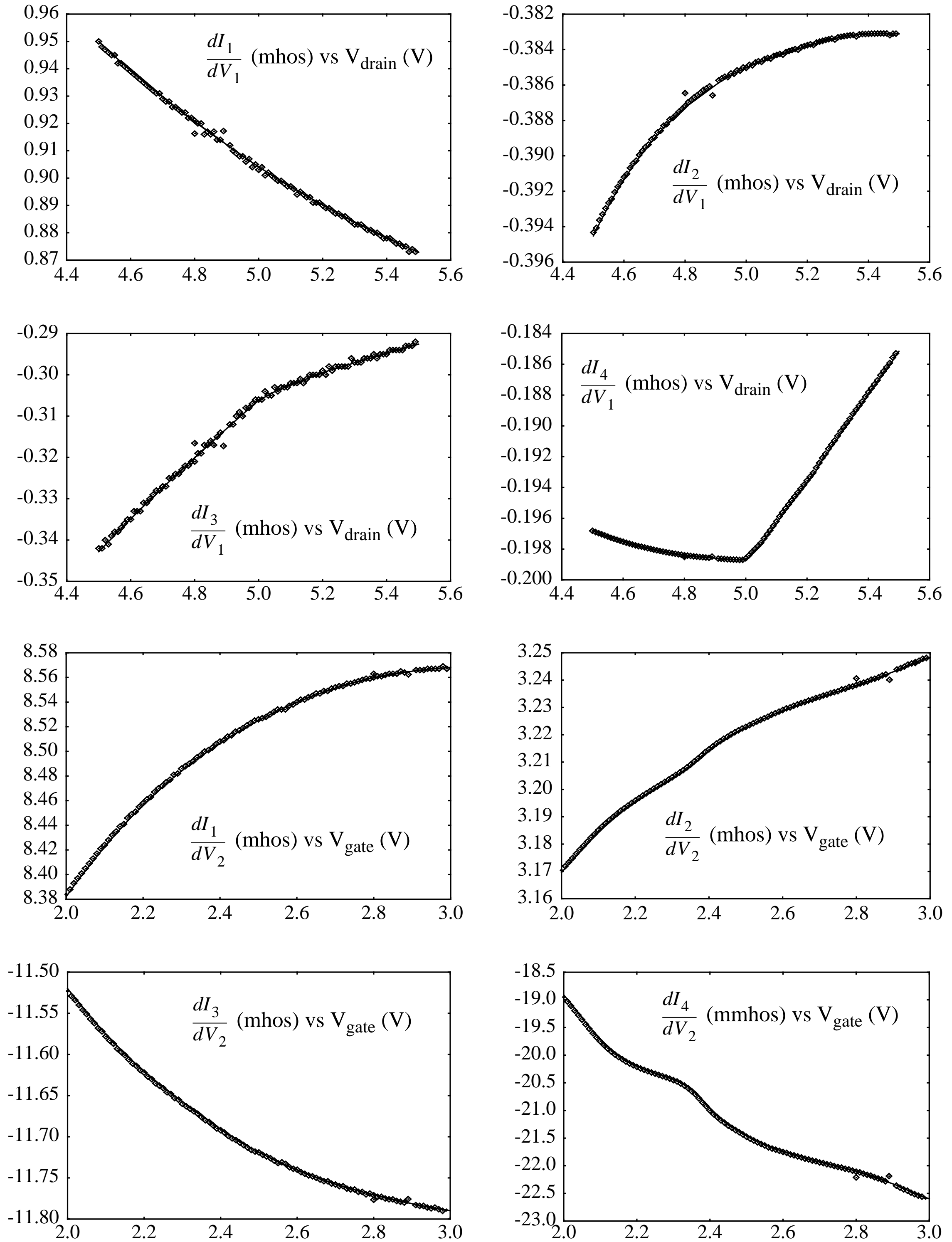


Figure 2.2: Evaluation of dI_i/dV_j for i equal to 1 through 4 (drain, gate, source, and bulk of a (MOSFET) and j equal to 1 or 2 during a transient step from $t_o = 50$ps to $t_o + \Delta t$ = 55ps. All contacts except the swept contact are held at constant bias.

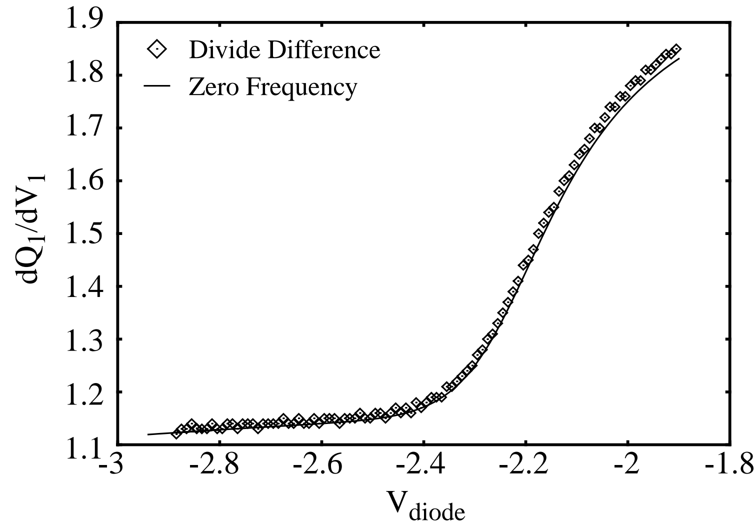


Figure 2.3: Evaluation of dQ/dV for a diode during a transient step from $t_o = 150$ps to $t_o + \Delta t = 155$ps during which the diode is in the process of being turned “on”.

semiconductor device simulations; however, it is possible to do so in a two terminal device like a diode. Under reverse bias, a diode can be modeled as a variable capacitor; hence, the anode and cathode can have charges associated with them.

For the verification of dQ/dV, a diode is switched from -10V to 0.7V in 1000ps. The transient simulation is stopped at t_o=150ps and restarted for a single time step of 5ps at various voltages. The charge at t_o and $t_o + \Delta t$ is computed by summing the positive and negative charge (which are equal for charge neutrality) and associating the negative charge with the p-side of the device and the positive charge with the n-side of the device under reverse bias. By divided difference, the value of $\Delta Q/\Delta V$ is estimated and compared with the value provided by a zero frequency simulation.

Figure 2.3 shows the comparison between the divided difference calculation of dQ/dV and the “zero” frequency simualtion.The agreement is good to within 5%. The relatively large difference for dQ/dV can be traced to the error introduced by summing the charge in a device and the divided difference routine for estimating the derivatives. In the simulations, the positive and negative charge (which should sum to zero) tended to be in agreement to about 1%. With round off errors in the divided difference approach, it is reasonable to get the differences shown in Figure 2.3.

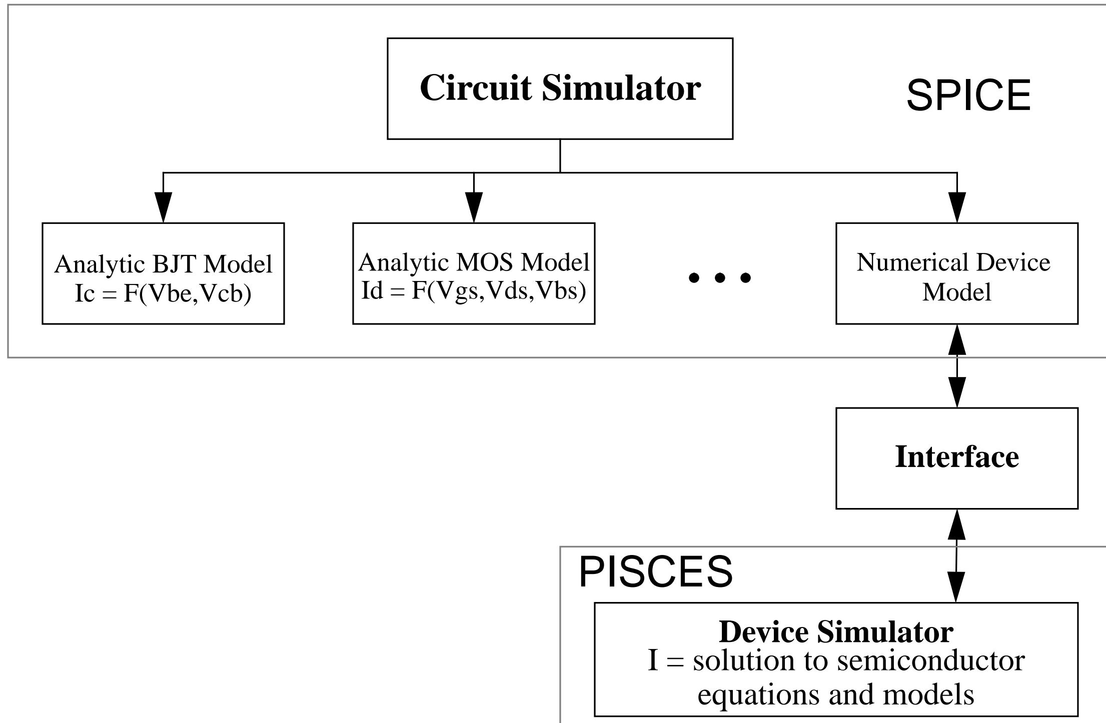


Figure 2.4: Block diagram for interfacing a numerical device simulator like PISCES into a circuit simulator like SPICE. It becomes another block in similar manner to an analytic model.

2.5 Modular Interface for Two-Level Newton

The mixed-mode interface is the code written to provide the communication between a circuit simulator like SPICE [31] and a numerical device simulator like PISCES [18]. Figure 2.4 shows how the two simulators are configured to communicate with each other. Similarly to an analytic model (like a BJT or MOSFET), a numerical device represents another block in circuit simulator [32] [33]. This block communicates to the numerical device simulator through an interface routine. Like other analytic models, the numerical device provides terminal currents (and conductances) for given node voltages the difference being that the latter finds the solution numerically.

This interface utilizes a two level Newton scheme to solve a circuit that includes numerical devices. On the upper level is the circuit iterations to determine the node voltages. For each one of these iterations, the device simulator solves for the terminal currents and conductance matrix using another Newton iteration within the device simulator (the lower level). The advantage of this method is the modularity created in the code.

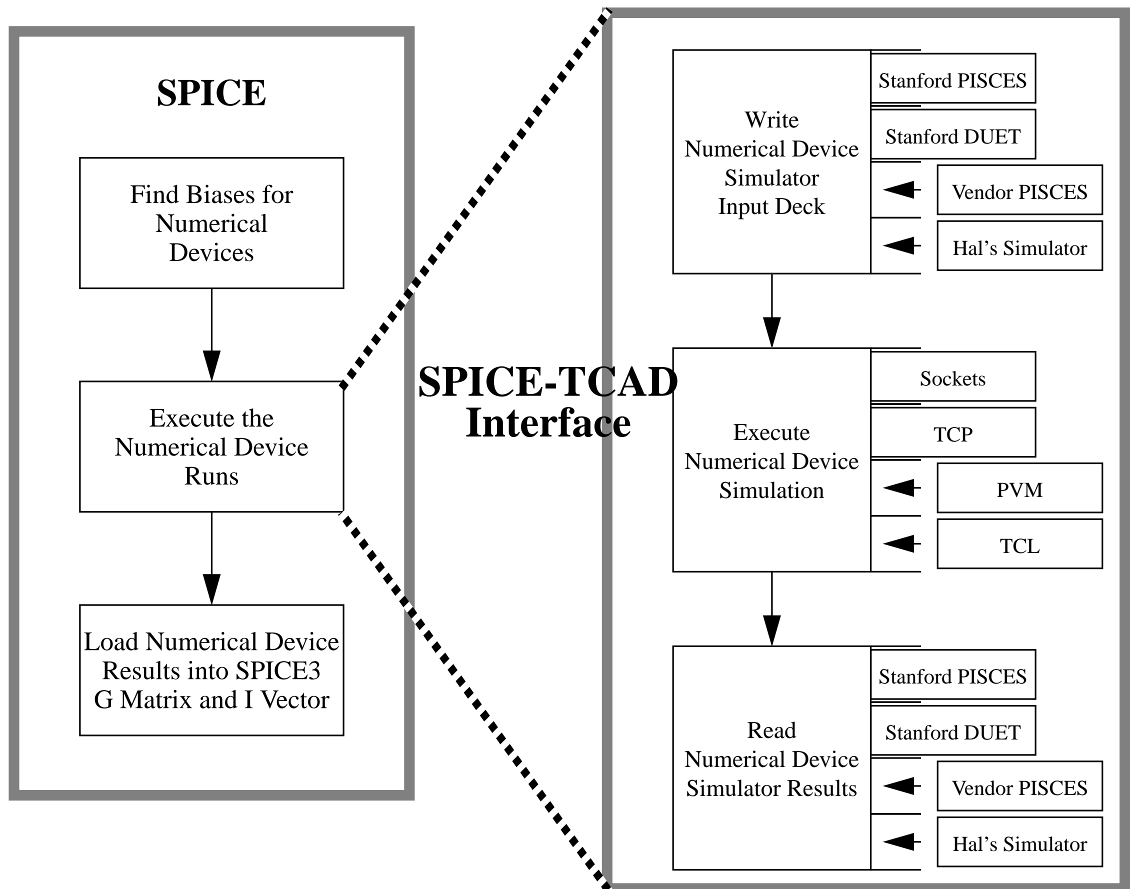


Figure 2.5: The modular interface for the two-level Newton algorithm for mixed circuit and device simulation.

Figure 2.5 shows the flow diagram for the execution of a single iteration step at the upper level. The block on the far right hand side represents different modules for the device simulator and/or method of execution. The sequence of steps is as follows:

1. Determine the voltage boundary conditions, time step, and/or frequency that the circuit simulator requires for the current iteration/solution. The exact information depends on the type of analysis but is independent of the individual device model.

2. Create the numerical device input deck that solves for the current and derivatives given the boundary conditions supplied by the circuit simulator. A separate module exists for each device simulator such that a new simulator may be added easily.

3. Execute the numerical device simulations on the current host or on a remote host by some specified protocol. This execution method can be sequential or some parallel algorithm (Section 2.6). The execution method is independent of the simulator and solution requested.

4. Read the results from the simulation based upon the simulator output file format.

5. Based upon the analysis type, load the circuit simulator's conductance (admittance) matrix and current vector in order to complete another circuit iteration.

6. Repeat for each device and each circuit iteration.

Since a device simulation is required for each circuit iteration, the interface algorithms are given some intelligence to limit the computations required by the numerical device. The most important aspects of those algorithms are:

1. Maximize the use of predictor corrector routines to decrease the calls to the device simulator [22]. These routines also reduce the number of calls to other models in addition to that of the numerical device model, but since the numerical device model requires significantly more CPU time it will dominate.

2. Do not iterate on devices that are not changing state; thus, taking advantage of latency in the circuit.

3. Use a-priori knowledge of the device behavior to provide a better guess of node voltages and to limit changes across semiconductor junctions.

4. Maintain files containing previous device solutions and utilize those solutions as an initial starting point for the requested bias.

Implementing each of these options provides an incremental decrease in the circuit simulation time.

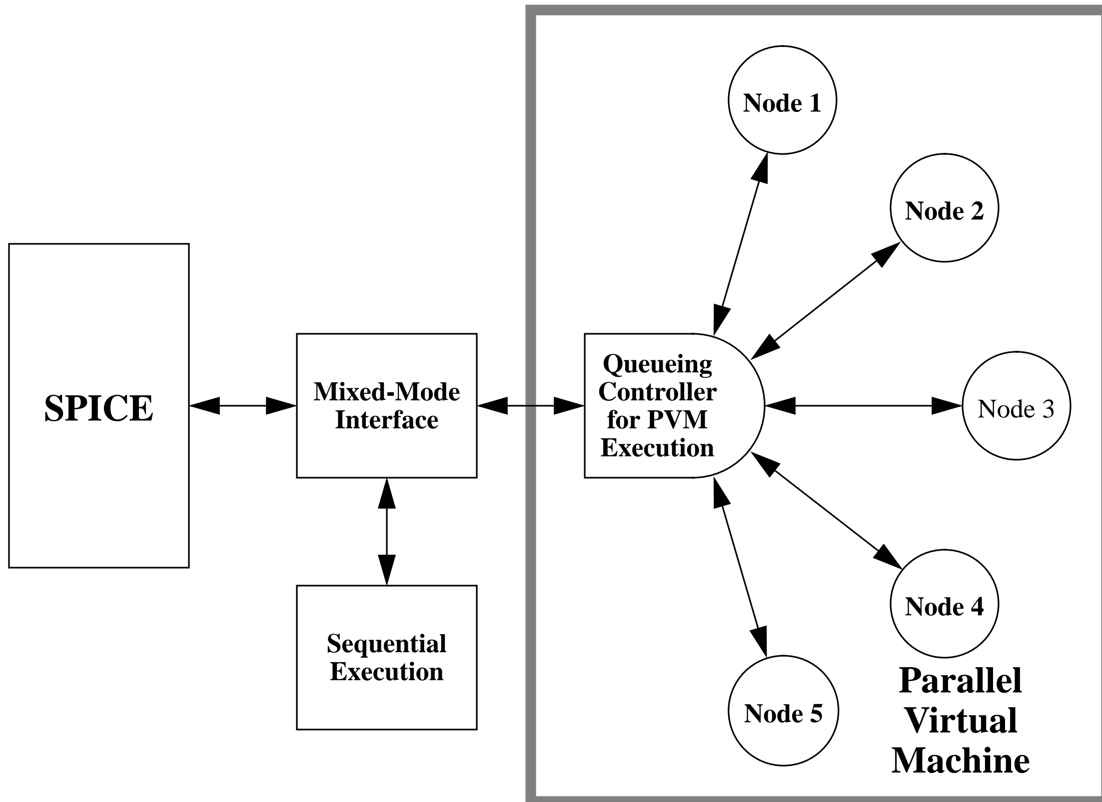


Figure 2.6: Parallel execution algorithm for two-level Newton mixed-mode simulations.

2.6 Parallel Execution for Two-Level Newton

In order to reduce the overall user time per circuit iteration, the numerical devices (*i.e.* the lower level Newton iterations) can be solved on different nodes on a network of machines. Because each device simulation is an independent unit, creating a parallel version of the code is relatively trivial.

The mixed-mode interface implemented in this work contains three mechanisms by which the parallel simulations are executed. The most basic of these algorithms utilizes standard Unix sockets [34] [35]. A more elegant mechanism involves PVM (Parallel Virtual Machine) which allows a heterogeneous network of computers to function like a parallel machine [36]. On a parallel machine, **M**essage **P**assing **I**nterface can be used to communicate between nodes [37]. Figure 2.6 shows a block diagram of how PVM is utilized. Likewise, MPI on a parallel machine employs a similar approach.

The mixed-mode interface spawns a queueing process on a node of the virtual machine. This process ascertains the number of available nodes that can be used for a device simulation. Whenever the interface program needs solutions from a number of numerical devices, it sends a request to the queueing program which then executes the device simulations on one of the available nodes. Once the device simulations are complete, the queueing program sends back the solutions to the interface and ultimately back to the circuit simulator.

2.7 Full-Newton Approach

In the full-Newton approach, the device and circuit matrix are solved simultaneously as shown graphically in Figure 2.1. The device matrices are assembled based upon the spatial discretization scheme. The circuit equations are assembled through modified nodal analysis as describe in Section 2.2.2. In addition, information about the connectivity in the circuit is provide by off diagonal terms in the device matrices.

The mixed circuit and device matrix system is solved using an iteration scheme optimized for the numerical devices since they dominate the solution [14] [38]. During each iteration, the derivative information is computed directly since the equations are known already. Iterations continue until the difference between the solutions for (Ψ^{i+1}, n^{i+1}, p^{i+1}, V^{i+1}) and (Ψ^{i}, n^{i}, p^{i}, V^{i}) is less than some tolerance.

2.8 Computational Cost of Mixed Circuit and Device Simulation

The two-level Newton method for mixed circuit-device simulation may incur a significant computational cost since multiple device simulations are required for each circuit solution point. Mayaram had investigated that cost for a variety of algorithms including the two-level Newton algorithm [14]. The later of those algorithms had been implemented by Rollins [10] and a variation of it had been implemented by Gates with the later developing a parallel execution method [15].

The two level Newton approach has the following advantages:

1. It has better convergence for dc analysis if node voltages are unknown. The full Newton requires all circuit node voltages to be specified within a certain percentage; otherwise, it fails to converge.

2. The modularity in two level Newton allows many device simulators to be used simultaneously. For example, PISCES may be used for standard devices in the circuit whereas a specialized device simulator may be used for novel devices.

3. It is easier to implement a parallel version of the two level Newton algorithm. Each numerical device simulation can be relegated to a node of a parallel machine whereas the matrix for the full-Newton algorithm has to be partitioned to each node and is much more dependent on the message passing capabilities.

4. By utilizing a standard simulator like SPICE, any improvements or modifications to the circuit simulator automatically benefit the mixed circuit-device simulation.

Likewise, there are a number of disadvantages to the two level Newton algorithm which Mayaram discusses in detail:

1. Given a good estimate of all the circuit node voltages, the full-Newton algorithm converges more quickly than the two-level Newton.

2. Similarly, since transient analysis involves small voltage changes at each time step making the problem well behaved, the full-Newton method converges much more quickly for this case as well. Mayaram determined a factor of 1.7 times as fast.

A full-Newton approach may not provide as strong a computational improvement as suggested by Mayaram [14]. Suppose there are 10 numerical devices with 2500 nodes each in a unit cell. In the full-Newton method an equation system with 3 x 10 x 2500 variables has to be solved for each circuit iteration. On the other hand for a two-level Newton algorithm, 10 matrices of 3 x 2500 will need to be solved for each circuit iteration. Given the size of the matrices and the memory size needed to solve the matrices, the two-level Newton is faster. Mayaram's work only addresses circuits with devices containing a couple of hundred nodes maximum and simple topographies.

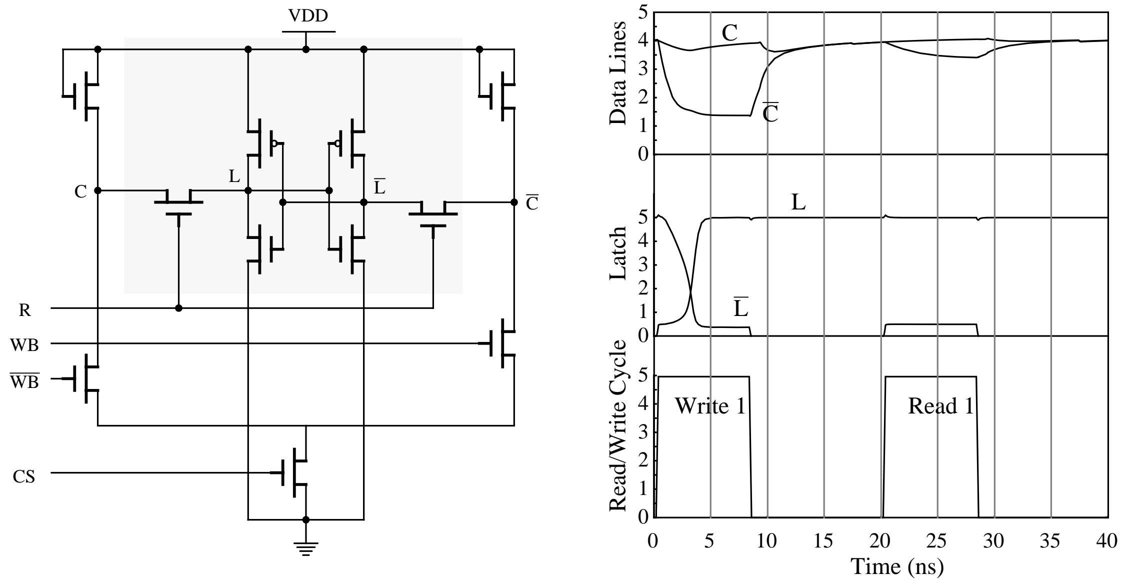


Figure 2.7: Circuit diagram and a read/write cycle for a SRAM cell and its surrounding control circuitry.

2.9 Benchmark Circuits

An SRAM cell is used to analyze the computational time of a two-level Newton mixed-mode simulation. Figure 2.7 contains a circuit diagram of the simulated structure and the response. The six transistor SRAM cell is simulated with a set of numerical devices containing a wide number of nodes per device with three circuit configurations. One circuit has only the pass transistor as a numerical device, another has half the cell (3 transistors) as numerical devices, and the third has all six transistors as numerical devices. The circuit was simulated to do a read/write cycle of a one and a zero so that each half of the cell is exercised in the same manner for cases of non-symmetry. The non-numerical device transistors are modeled to approximate the numerical device characteristics [39].

The simulations were executed on a single node of an IBM SP1 with 33Mhz RS6000 microprocessors. Figure 2.8 shows the total CPU time for all the combination of simulation runs. The first line (at 5×10^5 or 14 hours) is the time for an overnight run. At 33Mhz, the older SP1 is much slower than the 200+ Mhz machines available now. Multiplying by a 6X factor yields an overnight run time shown by the second line at 3×10^6. Thus it is possible to run an SRAM cell of 6 devices each containing 2000 nodes in a reasonable amount of time for design verification.

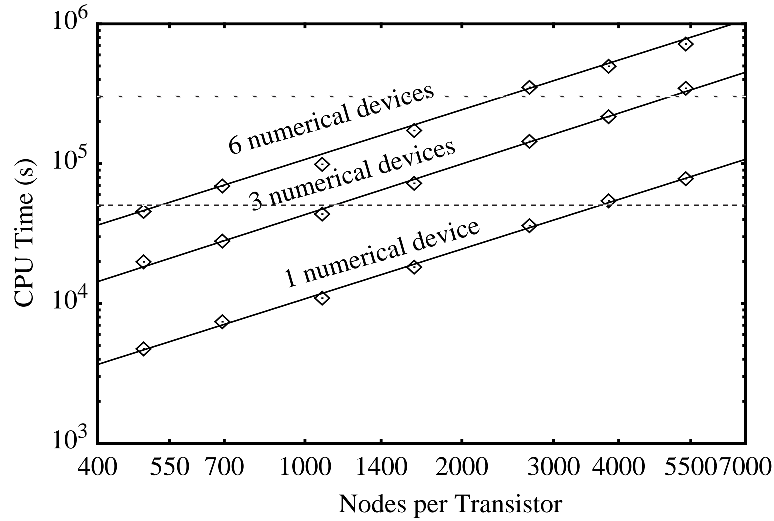


Figure 2.8: CPU Time for simulating a read/write cycle of an SRAM cell.

Nodes	User Time	%CPU	%Idle	Improve
1	51734s	90.9%	0.3%	1.00
2	27696s	90.7%	5.6%	1.87
3	20081s	90.5%	10.3%	2.58
4	18850s	90.3%	27.8%	2.74
5	17260s	89.8%	35.9%	3.00
6	12415s	88.3%	24.7%	4.17

Table 2.1: Resource statistics for a parallel mixed-mode simulation of an SRAM cell using the full-Newton algorithm.

Table 2.1 contains a listing the resource usage for a parallel execution of a 6 transistor SRAM cell with 500 nodes per device on an SP1. The single node simulation is provided strictly for reference. The first column is the total user time for those results. The sum CPU time for all cases is the same, hence, the important figure is how long the user has to wait for a solution. The second column contains the percentage of the processing time spent on CPU usage on a per device average. The average sum CPU time for the device simulations is about 45000s (12.5 hours). The average actual user time required is 52000s (14.4hrs).

The important consideration is how efficient is the parallel execution. That information is summed up in the percent idle and the improvement figure. The percent idle is the time the nodes are idle (not doing any processing) divided by the total time for the simulation results. The improvement factor is the amount of improvement gained by using a parallel simulation. For an N node parallel simulation, the maximum expected improvement is N.

The best performance is obtained for 2 and 3 node simulations. Poor performance is obtained for 4 and 5 node simulations and mediocre performance is obtained for a 6 node simulation.

These performance levels are a result of the parallelization scheme. In the parallel execution algorithm for two-level Newton, the entire device simulation is performed on a single node. Once that simulation is finished, another is executed on that node if needed. For 2, 3, and 6 nodes simulations, the 6 numerical devices in the SRAM cell can be evenly distributed over all the nodes. On the other hand, for the 4 node simulations, 4 numerical devices are executed followed by two more. Assuming each device simulation takes approximately the same, 2 nodes are idle while 2 are processing. Hence, very little advantage is gained over the 3 node parallelization. A similar condition occurs for a 5 node parallel execution.

The question arises as to the large idle time for the 6 node parallelization scheme over the 2 and 3 node. In each circuit iteration, it is not always necessary to simulate all the numerical devices. This is part of the computational savings aspect of the code. Many times, only one part of the circuit is changing (*i.e.* latency) and hence, only those devices need to be iterated upon. As a result, all 6 nodes are not being used in every single iteration.

2.10 Mixed Circuit and Device Simulations for Large Problems

In his investigation, Mayaram compared the computation time for the fully coupled and two-level Newton algorithms [14]. He found for transient analysis that the full-Newton method was faster than the two level Newton algorithm. Table 2.2 list some of his results for transient analysis. According to the table, only two basic problems are addressed. Four simulations are executed with multiple devices each containing only 61 nodes. Three simulations are executed with a single device containing 589 nodes and one with 61 nodes. There is no data for multiple devices with many nodes as one may find in a CMOS ring oscillator or SRAM cell. Therefore, the question posed in the previous section has not

Simulation Description			CPU Time (s)	
Circuit Name	Devices	Nodes per Device	2 Level Newton	Full Newton
Oscillator	1 BJT	61 (1D)	3636	2352
VCO	6 BJT	61 (1D)	5440	2911
Invchain	4 BJT	61 (1D)	965	514
Astable	2 BJT	61 (1D)	2538	1230
MECLgate	11 BJT's	61 (1D)	3931	2121
Pass	1 MOS	19 x 31 (2D)	1955	1059
MOSinv	1 MOS	19 x 31 (2D)	2194	1155
Charge Pump	1 MOS	19 x 31 (2D)	8045	4039

Table 2.2: CODEC simulation comparison for two-level Newton and full-Newton simulations.

been answered: Is it better to solve a large matrix system N times larger for N devices (full-Newton) or solve the device matrix N times for each circuit iteration (two-level Newton)?

To answer this question, PISCES is internally configured to allow simple circuit simulations using a full-Newton algorithm. Using the re-configured device simulator and the two-level Newton approach described in this work, a dc simulation of inverter strings is executed to compare the algorithms. A preferred comparison would be for the transient case, but because each simulator uses a different time stepping algorithm and time step control, it is not be possible to differentiate these algorithms based on the actual CPU time. Because the dc simulation is sweeping voltages by a small differential amount, it closely approximates a transient step and therefore provides a good reference for a time domain simulation.

The circuit consists of CMOS inverters connected in sequence. The input to the first inverter is swept and the output level at the last inverter is the desired response. The number of nodes in the entire inverter (number of nodes in the NMOS transistor plus the number of nodes in the PMOS transistor) is varied from about 800 to 8000. The inverter

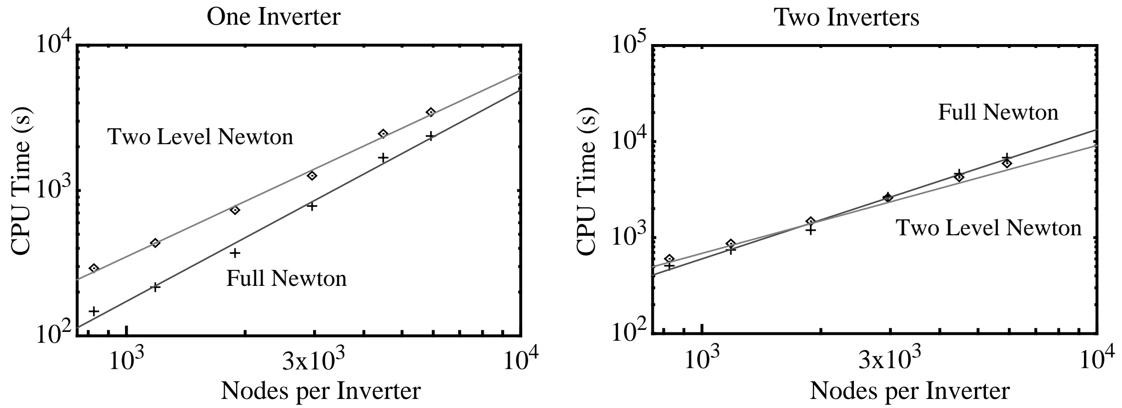


Figure 2.9: CPU time comparison of the two-level Newton and full-Newton algorithms for (a) a one inverter and (b) a two inverter chain for different number of nodes per inverter.

chain is biased at 3.3 volts and the input is swept from 0 volts to 5 volts in steps of 0.1. The total CPU time is recorded.

Figure 2.9 shows plots of the CPU time for the two-level Newton and the full-Newton algorithms on a node of an SP1. One plot is generated with a single inverter, while the second is generated on two inverters in sequence. For the small devices, both graphs show that the full Newton method outperforms the two-level Newton approach. However, as the number of nodes increases, the two-level Newton catches up to the full Newton and eventually outperforms it. This conclusion was also qualitatively verified by IBM using a similar approach with their simulator FIELDAY [29].

Looking at the problem from another perspective, a coupled PISCES simulation can be consider solely a PISCES simulation since the same solution algorithm is used regardless of the actual number of active devices regions. Thus the performance can be examine by the total size of the matrix independent of the number of devices. Figure 2.10 shows such a plot that consolidates the one and two inverter results and includes three and five inverter results. Unfortunately, the size of the three and five inverter PISCES simulations are limited by the memory on the computer (a major limitation of the full-Newton approach). Thus, the plot only contains those for which a simulation was possible.

Figure 2.10 shows that the better performing algorithm depends upon the size of the simulation problem. For a relatively small system (*i.e.* many devices with few nodes or a few devices with many nodes) the full-Newton algorithm out performs the two-level Newton. Mayaram's study focused on this region [14]. For a large system (*i.e.* many

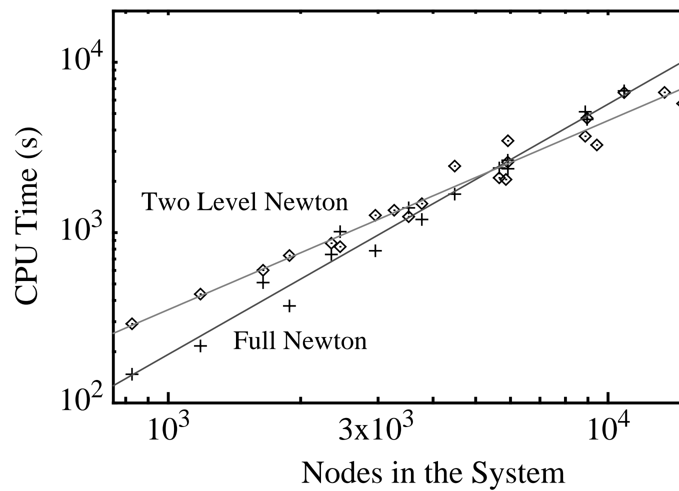


Figure 2.10: CPU time comparison of the two-level Newton and full-Newton algorithms for a generic mixed circuit and device simulation of an inverter chain.

device with a moderate number of nodes or a few devices with a large number of nodes), the two-level Newton out performs the full-Newton algorithm.

2.11 Conclusions

This chapter presents a detailed discussion of mixed circuit and device simulation. It expands upon the previous work of Mayaram and Rollins and investigates issues not addressed previously. A parallel version of the code is implemented and analyzed for its performance. The simulation of large problems is investigated for both the full-Newton and two-level Newton algorithm thus addressing an area not previously examined.

Chapter 3: Boundary Conditions for Device Simulation

3.1 Description

This chapter's discussion focuses on improvements to a device simulator like PISCES in order to meet the requirements of RF device simulation. The previous chapter discusses how circuit information is linked with a numerical device simulation and thus, devices can be optimized for their circuit application. In this chapter, physical device simulation is discussed for applications involving discrete transistors. Two important improvements to PISCES are implemented in order to meet the requirements for RF device simulation: generic circuit boundary conditions [40] and harmonic balance simulation [41].

In RF design, some simple yet important circuits consist of only a non-linear device and linear circuit components surrounding it. A full mixed circuit and device simulation is one option for solving this problem, yet a better approach would be to use more advanced boundary conditions in the device simulator. This approach provides an advantage in the computational time and ease of solution with a harmonic balance device solver. Harmonic balance simulation is used to find the large signal steady state solution for a sinusoidal input signal [42]. Instead of solving in the time domain using transient analysis, harmonic balance solves the system in the frequency domain as discussed later in this chapter.

3.2 Device Simulation with Parasitic Components

Figure 3.1 shows the different mechanisms to integrate circuit components with a device simulation. In Figure 3.1a, a simple device contains resistive components on each

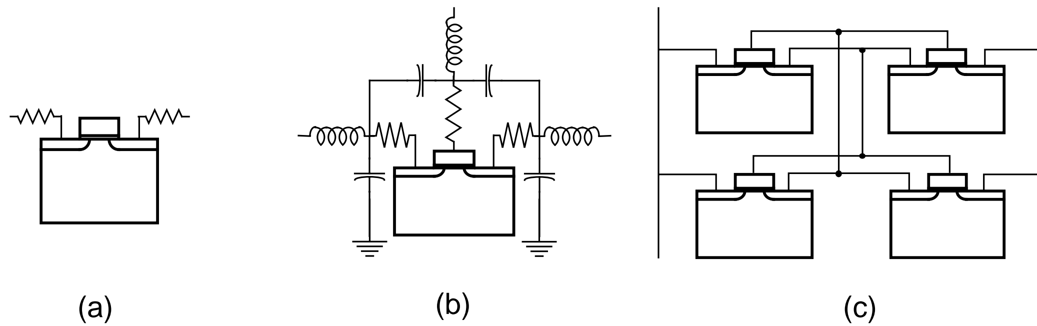


Figure 3.1: Types of problems involving mixed circuit and device simulation.

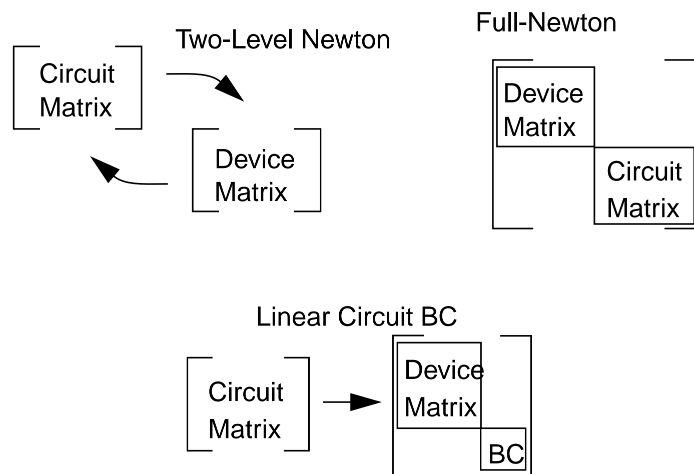


Figure 3.2: Methods for solving a mixed circuit and device problem for a single device surrounded by linear elements.

electrode. By applying a simple boundary condition equation, those resistances are coupled with the device simulation. In Figure 3.1c, a fully coupled circuit simulation is performed. The circuit contains independent devices such that the circuit nodes define the voltage on each electrode. Figure 3.1b is a combination of both techniques. It has circuit components and nodes not connected to the device electrodes, thus requiring a circuit level solution; yet it can be reduced to set a boundary equations in the device simulator.

Figure 3.2 shows the different approaches for solving a numerical device surrounded by linear components (Figure 3.1b). In the two-level Newton algorithm the circuit solution is iterated upon and for each iteration a device simulation is executed. Since the circuit matrix is linear except for the numerical device, the circuit converges quickly; however, that still implies at least three circuit iterations and consequently three device simulations.

The full-Newton approach includes the circuit matrix with the device matrix. Since the circuit matrix is linear, it doesn't adversely affect convergence of the numerical device. Since it is included with the device matrix, multiple device solves are not required. However, consider the case where the circuit consists of a resistive mesh to model a substrate or contains a model for a transmission line that uses a large number of T-sections or Π-sections. Both of these cases result in 100's of circuit nodes. As a result, the circuit matrix can compose a significant portion of the matrix system. Even though that portion of the circuit is linear, it must be factored every device iteration thus impacting the total computational time.

A boundary condition approach combines the full-Newton and two-level Newton algorithms. Since the circuit matrix is linear with the exception of the numerical device, it is reduced to a boundary condition equation in one iteration. This boundary condition equation adds only one equation for each electrode in the device matrix; thus, it does not drastically increase the size of the device matrix. After the device simulation is completed, the device results are incorporated into the circuit solution. Since the circuit is linear and the circuit effects are incorporated in the device simulations through boundary conditions, no further iterations are required.

This chapter provides an analysis of the boundary conditions by reviewing the standard approaches in device simulation. Upon building a foundation, a technique is described for incorporating a generic linear circuit into a device simulation through boundary conditions equations.

3.3 Types of Boundary Conditions

In order to solve the semiconductor equations numerically, appropriate boundary conditions must be applied. for the three unknown variables Ψ (potential), n (electron concentration), and p (hole concentration). The boundary conditions for these variables are set at the material interfaces, edges of the simulation space, and at the contact electrodes. The material interface and edge boundary conditions are discussed in detail in other works and are not addressed in this work [25]. The electrode boundaries are the regions where extrinsic effects impact the device's performance are discussed in detail.

For each boundary condition, an equation describing the boundary condition is provided. That equation represents the right hand side (RHS) expression or residual in the device

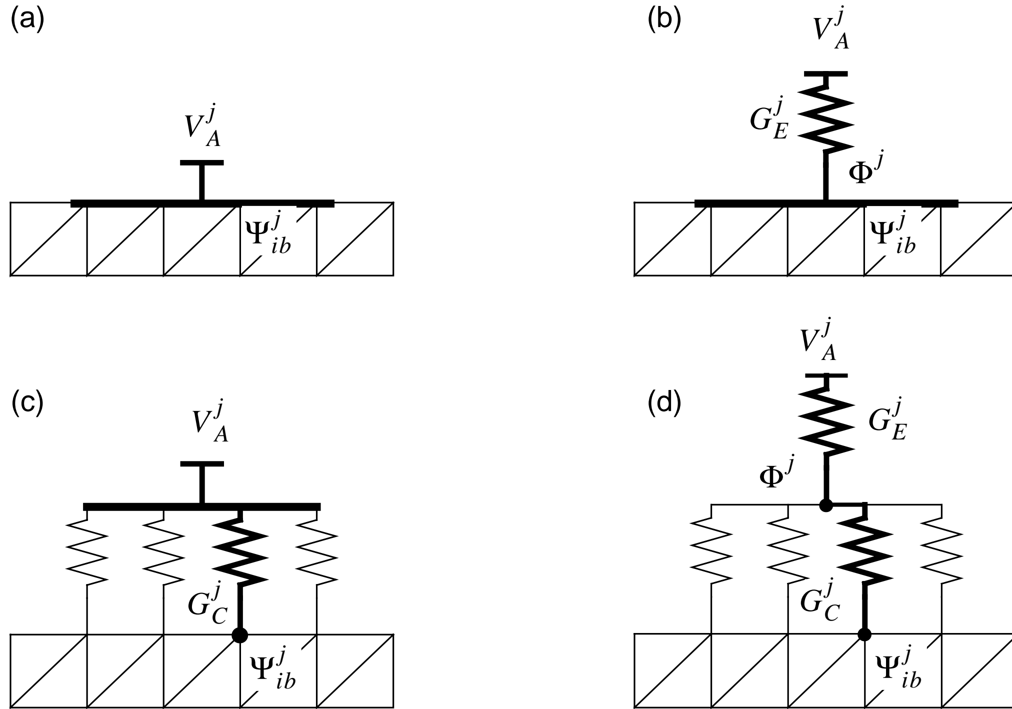


Figure 3.3: Resistive boundary conditions for a numerical device structure. (a) Plain voltage boundary condition. (b) External resistance. (c) Distributed resistance. (d) Combined distributed and external resistance.

simulator's matrix system. The derivative of that equation is used to load the Jacobian during each Newton-Raphson iteration [24].

3.3.1 Voltage Boundary Conditions

The simplest boundary condition is an applied voltage to an electrode on the numerical device with an ohmic contact (Figure 3.3a). Under this condition, the potential, electron concentration, and hole concentrations are fixed assuming zero space charge at the surface (*i.e.* no depletion region is formed). This assumption is valid only for cases where the semiconductor material is heavily doped such that any depletion region formed is assumed infinitesimally thin, there are no surface states at the interface, and there is infinite surface recombination [25]. Under this condition charge neutrality requires

$$n_{si}^j + N_{Ai}^{j\,-} = p_{si}^j + N_{Di}^{j\,+} \qquad (3.1)$$

where n_{si}^{j} is the electron surface concentration of node i in electrode j, N_{Ai}^{j-} is the ionized acceptor impurity concentration for node i in electrode j, p_{si}^{j} is the hole surface concentration of node i in electrode j, and N_{Di}^{j+} is the ionized donor impurity concentration for node i in electrode j.

The ohmic boundary condition assumes a single Fermi level and thus

$$n_{si}^{j} p_{si}^{j} = n_{ii}^{j} \tag{3.2}$$

where n_{ii}^{j} is the intrinsic carrier concentration at node i in electrode j and can be dependent upon models such as band gap narrowing as well as temperature [43].

Solving the two equations simultaneously yields

$$n_{si}^{j} = \frac{N_{Di}^{j+} - N_{Ai}^{j-}}{2} + \left[\left(\frac{N_{Di}^{j+} - N_{Ai}^{j-}}{2}\right)^{2} + n_{ii}^{j}\right]^{1/2} \quad \text{and} \quad p_{si}^{j} = \frac{n_{ii}^{j}}{n_{si}^{j}} \tag{3.3}$$

for the carrier surface concentrations.

When the semiconductor and metal are brought into contact the Fermi levels must align such that

$$\Psi_{ib}^{j} = -E_{FM}^{j} + \varphi_{T} \ln\left(\frac{n_{si}^{j}}{n_{si}^{j}}\right) \tag{3.4}$$

where E_{FM}^{j} is the Fermi level of the metal for electrode j, φ_{T} is the thermal voltage and no space charge region is present (*i.e.* no band bending or built-in potential). When a bias is applied to the metal, its Fermi level changes. Alternatively, by using the zero bias equilibrium Fermi level as reference, Equation 3.4 can be written as the following boundary condition equation in the simulator matrix system:

$$F_{i} \equiv V_{A}^{j} - \Psi_{ib}^{j} + \varphi_{T} \ln\left(\frac{n_{si}^{j}}{n_{si}^{j}}\right) = 0 \tag{3.5}$$

where V_A^j is the applied voltage on electrode j. The right hand side of the simulator matrix equation is the evaluation of F_i during each Newton-Raphson iteration based upon the previous solution. The Jacobian is loaded with the derivatives of F_i with respect to the unknowns and is given by

$$\frac{\partial F_i}{\partial \Psi_{ib}} = -1 \tag{3.6}$$

Since the impurity concentrations at the surface are constant because of the ohmic contact assumption.

A Schottky contact requires a more complex boundary condition given by

$$\Psi_{ib}^j = -E_{FM}^j - \Phi_{bi}^j + \frac{1}{2}\left[E_g + \varphi_T \ln\left(\frac{N_C}{N_V}\right)\right] \tag{3.7}$$

where Φ_{bi}^j is the contact barrier height, E_g is the semiconductor bandgap, and N_C and N_V are the effective density of states. The barrier height is dependent upon the built-in potential and space charge region which in turn depends upon the solution to the internal variables and doping profiles. It can also be affected by barrier lowering due to image forces [44]. Pinto provides an analysis for a uniform doping profile [25].

Finite surface recombination limits the current flowing across the electrode [45]. As a result, when current is computed internally based upon the solution for Ψ, n, and p, it is limited by surface recombination such that

$$\begin{aligned} J_{sni}^j &= q v_{sn}^j (n_{si}^j - n_{ei}^j) \\ J_{spi}^j &= q v_{sp}^j (p_{si}^j - p_{ei}^j) \end{aligned} \tag{3.8}$$

where J_{sni}^j and J_{spi}^j are the electron and hole current density for node i in electrode j, v_{sn}^j and v_{sp}^j are the electron and hole surface recombination velocities for electrode j, and n_{ei}^j and p_{ei}^j are the electron and hole equilibrium concentrations for node i in electrode j.

3.3.2 External Resistance and Capacitance

The next level of complexity for a boundary condition involves placing a parallel capacitor and resistor combination on an electrode. In the dc case such that the capacitor is an open circuit (Figure 3.4b), Kirchoff's Current Law (KCL) is applied to obtain

$$H_j \equiv G_E^j(V_A^j{}' - \Phi^j) - I^j(\Psi, n, p) = 0 \tag{3.9}$$

where Φ^j is the potential of the electrode boundary nodes defined such that

$$F_i \equiv \Phi^j - \Psi_{ib}^j = 0. \tag{3.10}$$

G_E^j is the conductance of the external resistor on electrode j and $I^j(\Psi, n, p)$ is the current flowing through the node computed from the solution for $\Psi, n,$ and p based upon the discretization scheme. Since Φ^j is unknown it is added to the matrix system through equation H_j which forces all surface potentials of electrode j to be equal.

The Jacobian is evaluated from the derivatives of H_j and F_i with respect to the unknowns and is given by

$$\frac{\partial H_j}{\partial \Phi^j} = -G_E^j , \tag{3.11}$$

$$\frac{\partial F_i}{\partial \Psi_{ib}} = -1 \quad \text{and} \quad \frac{\partial F_i}{\partial \Phi^j} = 1 . \tag{3.12}$$

During transient analysis, an external capacitance C_E^j in parallel with the resistor adds a displacement current component. The displacement current term I_D^j is computed from the capacitance value C_E^j, the time step Δt, and the potential across the capacitance. A simple evaluation of the capacitance using backward Euler time discretization step is

$$I_D^j = \frac{\partial E}{\partial t} = C_E^j\left(\frac{V_A^j{}' - \Phi^j}{\Delta t}\right) \tag{3.13}$$

but, a more complex time stepping routines can result in more complex expressions.

The displacement current adds to the expression for H_j as:

$$H_j \equiv G_E^j(V_A^j{}' - \Phi^j) - I^j(\Psi, n, p) + C_E^j\left(\frac{V_A^j{}' - \Phi^j}{\Delta t}\right) = 0 \quad (3.14)$$

The new derivative (assuming backward Euler discretization) is

$$\frac{\partial H_j}{\partial \Phi^j} = -\frac{C_E^j}{\Delta t} - G_E^j. \quad (3.15)$$

3.3.3 Distributed Contact Resistance

A distributed contact resistance models the resistance between the metal contact and silicon. To model this resistance (Figure 3.3c), an appropriately scaled resistance is placed at each boundary node of the electrode [25]. As a result, the boundary potentials Ψ_{ib}^j for a given electrode are no longer equal. The boundary condition for each individual node is given by

$$F_i \equiv G_{Ci}^j(V_A^j{}' - \Psi_{ib}^j) - I_i^j(\Psi_i^j, n_i^j, p_i^j) = 0 \quad (3.16)$$

where G_{Ci}^j is the contact resistance of electrode j scaled for node i and $I_i^j(\Psi_{ib}^j, n_i^j, p_i^j)$ is the current component of electrode j flowing into node i.

For a distributed contact resistance, only the equation of the potential is modified and no new additional equations are required. The derivative of F_i is required in order to compute the Jacobian and is given by

$$\frac{\partial F_i}{\partial \Psi_i} = -G_{Ci} \quad (3.17)$$

3.3.4 Contact Resistance with External Resistances

Integration of boundary conditions at the next level involves using contact and external resistances simultaneously (Figure 3.3d). The distributed contact resistance equations for F_i is first modified to account for the unknown potential Φ^j as

$$F_i \equiv G^j_{Ci}(\Phi^j - \Psi^j_i) - I^j_i(\Psi^j_i, n^j_i, p^j_i) = 0 \tag{3.18}$$

where the terms have been previously defined. The unknown potential is then related to the applied potential through the external resistance as given by

$$H_j \equiv G^j_E(V^j_A - \Phi^j) - I^j(\Psi, n, p) = 0 \tag{3.19}$$

The total current flowing through the electrode $I^j(\Psi, n, p)$ is the sum of the individual currents flowing through each contact resistor $I_i(\Psi, n, p)$ and can be summed as:

$$I^j(\Psi, n, p) = \sum_{n_b} I_i(\Psi, n, p) = \sum_{n_b} G_{Ci}(\Phi^j - \Psi_i) \tag{3.20}$$

Substituting the expression for $I^j(\Psi, n, p)$ into the equation for H_j yields

$$H_j \equiv G^j_E(V^j_A - \Phi^j) - \sum_{n_b} G_{Ci}(\Phi^j - \Psi_i) = 0 \tag{3.21}$$

where H_j is an additional equation added to the matrix system for the unknown variable Φ^j. Since both H_j and F_i contain the unknowns for Φ^j and Ψ^j_i the derivative with respect to these terms are required in the Jacobian and are given by

$$\frac{\partial F_i}{\partial \Psi_i} = -G_{Ci} \ , \tag{3.22}$$

$$\frac{\partial H_j}{\partial \Phi^j} = -G^j_E - \sum_{n_b} G_{Ci} , \tag{3.23}$$

$$\frac{\partial F_i}{\partial \Psi_i} = G_{Ci} \quad , \tag{3.24}$$

$$\text{and } \frac{\partial H_j}{\partial \Psi_i} = G_{Ci} \tag{3.25}$$

3.3.5 Current Boundary Condition

All boundary conditions thus far have been based upon an applied voltage, an alternative is to force a current into the boundary electrode. Current boundary conditions are very similar to external resistance boundary conditions. The current flowing into the electrode $I^j(\Psi, n, p)$ is set equal to the applied source I_A^j such that

$$H_j \equiv I_A^j - I^j(\Psi, n, p) = 0 \tag{3.26}$$

In addition, the potential on each of the boundary nodes of the electrode must be equal to each other (assuming no contact resistance). As a result, an additional variable Φ^j and its associated expression is included in the system of equations as

$$F_i \equiv \Phi^j - \Psi_{ib}^j = 0 \tag{3.27}$$

The derivatives of Equation 3.26 is determined by the discretization scheme and are available during the assembly of the Jacobian for the variables internal to the structure. In addition, the derivatives with respect to F_j yields similar equations to that for an electrode with an external resistance and are given by

$$\frac{\partial F_i}{\partial \Psi_{ib}} = -1 \quad \text{and} \quad \frac{\partial F_i}{\partial \Phi^j} = 1 \, . \tag{3.28}$$

A current boundary condition can be integrated with a contact resistance in the same manner as an external resistor.

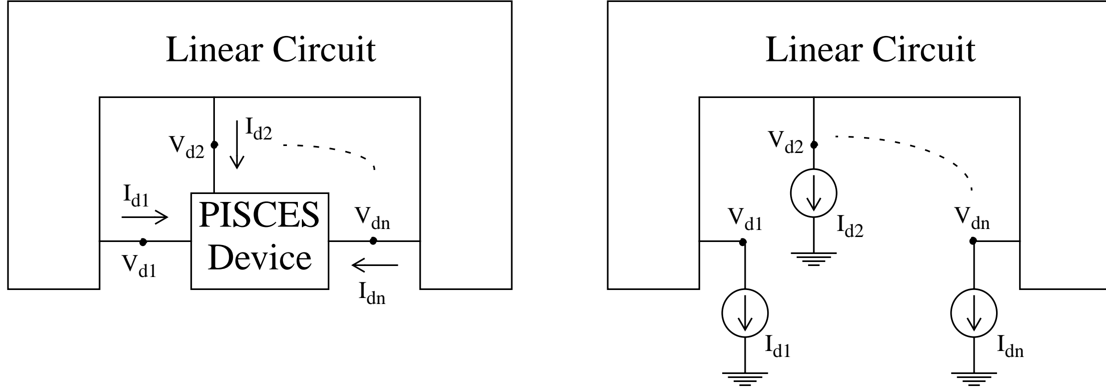


Figure 3.4: (a) Linear circuit surrounding a numerical device. (b) For the purpose of finding the boundary conditions, the numerical device is assumed to consist of ideal current sources of unknown values.

3.4 Generic Circuit Boundary Condition Equations

As discussed in Section 3.3, an external resistance may be added to a boundary electrode. A simplification of that expression is:

$$F_d \equiv G_E(E_d - V_d) - I_d = 0 \tag{3.29}$$

where G_E is the external conductance, E_d is the applied voltage, V_d is the device electrode potential, and I_d is the current flowing into the electrode.

For a generic linear circuit surrounding a device, Equation 3.29 can be expressed in matrix format such that the voltages and currents become vectors and the conductance becomes a matrix. The matrix form of the boundary condition equation is implemented in the same manner as the external resistance with another physical boundary condition (*i.e.* contact resistance and limited surface recombination). An additional variable is added for the unknown value of the electrode potential V_d and the derivative of the matrix equations F_d are computed with respect to the unknown V_d in the Jacobian.

3.4.1 Formulating the Boundary Condition Equations

In order to determine the equations for the boundary conditions, consider Figure 3.4 where a linear circuit surrounds a numerical device. Substituting ideal current sources for the

numerical device connections to represent the $I^j(\Psi, n, p)$ computation during each Newton-Raphson device iteration, modified nodal analysis (MNA) [46] may be applied to the system to yield the following circuit equation:

$$\begin{bmatrix} F_c \\ F_d \end{bmatrix} \equiv \begin{bmatrix} G_{cc} & G_{cd} \\ G_{dc} & G_{dd} \end{bmatrix} \begin{bmatrix} \boldsymbol{V_c} \\ \boldsymbol{V_d} \end{bmatrix} + \begin{bmatrix} I_c \\ \boldsymbol{I_d} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}. \tag{3.30}$$

F_c and F_d represent the system of equations for the circuit and device nodes where G_{cc} is the conductance matrix for the independent circuit nodes, G_{dd} is the conductance matrix the components connected to a device electrode, and G_{cd} (which equals G_{dc} since the circuit is linear) represents the interaction between the circuit and device nodes. $\boldsymbol{V_c}$ and $\boldsymbol{V_d}$ are the unknown circuit node voltages and device node voltages as defined by the modified nodal analysis technique. I_c and $\boldsymbol{I_d}$ are the known circuit forcing functions and the unknown device current sources respectively.

For the purposes of this derivation, the circuit forcing functions are represented by current sources. By applying standard circuit simulation techniques [22], a voltage forcing function is implemented in a similar manner.

Because there are more unknowns than equations in this system, a unique solution is not possible until the device currents $\boldsymbol{I_d}$ are computed. The device currents are computed by solving the device system of equations with the boundary conditions provided by the expression F_d. As given in Equation 3.30, this expression contains the circuit node voltages which are not available in the device equation system; however, the circuit system can be transformed to

$$\begin{bmatrix} \boldsymbol{V_c} \\ \boldsymbol{V_d} \end{bmatrix} + \begin{bmatrix} G_{cc} & G_{cd} \\ G_{dc} & G_{dd} \end{bmatrix}^{-1} \begin{bmatrix} I_c \\ \boldsymbol{I_d} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}. \tag{3.31}$$

Evaluating the inverse of the conductance matrix results in

$$\begin{bmatrix} \boldsymbol{V_c} \\ \boldsymbol{V_d} \end{bmatrix} + \begin{bmatrix} M^{-1} & -G_{cc}^{-1} G_{cd} N^{-1} \\ -G_{dd}^{-1} G_{dc} M^{-1} & N^{-1} \end{bmatrix} \begin{bmatrix} I_c \\ \boldsymbol{I_d} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \tag{3.32}$$

where $M = G_{cc} - G_{cd}G_{dd}^{-1}G_{dc}$ and $N = G_{dd} - G_{dc}G_{cc}^{-1}G_{cd}$.

The resulting boundary conditions for the numerical device are now given by

$$\boldsymbol{V_d} - G_{dd}^{-1}G_{dc}M^{-1}I_c + V^{-1}\boldsymbol{I_d} = 0 \tag{3.33}$$

which can be transformed to

$$-N(G_{dd}^{-1}G_{dc}M^{-1}I_c - \boldsymbol{V_d}) - \boldsymbol{I_d} = 0 \tag{3.34}$$

such that $G_E = -N$ and $E_d = G_{dd}^{-1}G_{dc}M^{-1}I_c$ as given in Equation 3.29.

3.4.2 Implementing Circuit Boundary Conditions

The expression for G_E and E_d require a number of matrix inversions and multiplications. A more direct method involves only LU decomposing the full conductance matrix once and then using that factored version to solve some special cases of the equation system to evaluate G_E and E_d directly.

By letting $\boldsymbol{I_d} = 0$ and solving the system of equations, E_d is obtained as

$$\boldsymbol{V_d} - G_{dd}^{-1}G_{dc}M^{-1}I_c = 0 \quad \rightarrow \quad \boldsymbol{V_d} = E_d = G_{dd}^{-1}G_{dc}M^{-1}I_c \tag{3.35}$$

By letting $\boldsymbol{I_d} = 1$ and $I_c = 0$ then solving the system, G_E is obtained directly as

$$\boldsymbol{V_d} + N^{-1}\boldsymbol{I_d} = 0 \quad \rightarrow \quad G_E = \boldsymbol{I_d}\boldsymbol{V_d}^{-1} = -N \tag{3.36}$$

Upon applying the boundary conditions (Equation 3.29 given Equation 3.35 and Equation 3.36) to the device simulation and solving for the unknown currents $\boldsymbol{I_d}$, the nodal solution for the circuit and device is determined by solving Equation 3.31.

3.4.3 Multiple Circuit Boundary Conditions

The implementation of the generic boundary conditions assumes the conductance matrix can be factored. In general, not all circuit MNA matrices can be factored, but usually the

cause can be traced to a non-physical circuit configuration [47]. Even more troublesome is the fact that symbolic current sources for the device are introduced without regard to whether the circuit can by realized or the physical form of the circuit boundary conditions equations. As a result, there exist valid configurations where the determinant of the conductance matrix goes to zero or the determinant of G_{dd} and G_{cc} go to zero.

The boundary condition computation becomes indeterminate when a voltage or a current is applied directly to an electrode. A voltage source connected to an electrode only adds a term to G_{cd} and G_{dc} in the MNA matrix and leaves zeros along a diagonal in G_{dd} and G_{cc}. The MNA conductance matrix is invertible, but the individual components are not. A current source connected to an electrode can not be physically realized by this approach since this approach results in two current sources back to back.

In order to accommodate the directly applied voltage and current sources, the boundary equations for those electrodes must be handled separately as described in Section 3.3.1 and Section 3.3.5. Therefore, boundary conditions are separated into three types for which a summary is provided in Table 3.1.

3.5 Small Signal AC Analysis

Small signal analysis is a powerful capability to characterize both circuits designed for small signal applications and large signal applications. It provides information on the frequency performance of the device (*i.e.* F_t), extraction techniques can be used to determine intrinsic and extrinsic parasitic capacitance, and large signal performance can be estimated through S-parameters. For an RF and analog design engineer, such information becomes critical to their circuit and need to be understood.

Boundary Type	Boundary Equation
Constant Voltage	$F_d \equiv E_d - V_d = 0$
Constant Current	$F_d \equiv I_D - I_d = 0$
Generic Linear Circuit	$F_d \equiv G_E(E_d - V_d) - I_d = 0$

Table 3.1: Boundary condition equations for a generic circuit surrounding a PISCES device.

3.5.1 Description

In small signal analysis, the entire system is first solved for the dc steady state solution. Upon obtaining a solution, the system is linearized about the operating point. Each nonlinear element assumes the form of a model which is derived from the derivatives of the I-V (current-voltage) relationship and C-V (capacitance-voltage) curves of the device. With a linearized circuit, a perturbation can be applied at the desired node(s) and only a linear solution of the circuit equations is required.

The linearization of a numerical device is a bit more difficult. In device simulation, the small signal response of a device is characterized through Y-parameters from which other parameters are calculated. The Y-parameters are determined by perturbing the dc solution, measuring the changes in Ψ, n, and p, and computing the change in current with respect to the change in voltage. As described in Section 2.4.1, Laux developed a methodology for determining the frequency dependent change in the semiconductor variables ($\Delta\Psi$, Δn, Δp) by perturbing Ψ about the dc solution [28] (Equation 2.11 through Equation 2.15). From the solution for $\Delta\Psi$, Δn, and Δp, the perturbed current (ΔI) is computed and divide by the perturbing voltage.

3.5.2 Boundary Conditions

Boundary conditions add another layer of complexity to small signal analysis. Laux's approach only determines the impact of perturbations in Ψ on the semiconductor equation solutions from which Y-parameters are computed. This approach ignores any external circuit components since the solution is perturbed directly on the electrodes (*i.e.* Ψ_{ib}^{j}).

There are two approaches for incorporating extrinsic parasitic components into the small signal device solution as shown in Figure 3.5. Based upon the topology of the circuit (*i.e.* an external resistor in parallel with a capacitor), the Y-parameters of the entire system can be computed by connecting external blocks of Y-parameters through simple circuit theory techniques (Figure 3.5a). Alternatively, if the topology is generic, a better approach is to use small signal circuit simulation where the Y-parameters from the device simulator solution are substituted for the device structure. A circuit simulation is used to find the desired small signal results.

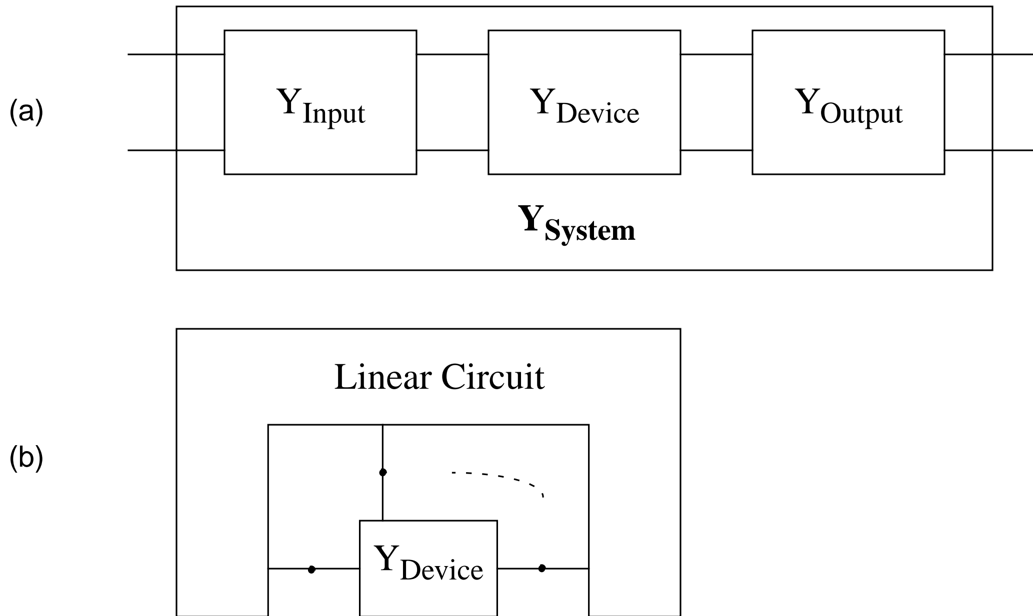


Figure 3.5: Coupling methods for Y-parameters of the device simulator with the small signal parameters of the circuit by (a) combining Y-parameters of each simulation block and (b) using the PISCES Y-parameters in a small signal simulation.

3.6 Transient Analysis

Transient analysis (simulation in the time domain) allows for the large signal characterization of a device in small and simple circuits. The response to a step function characterizes the speed of the device in terms of switching capabilities. Large signal sinusoidal simulation determines the performance of the device for high power RF applications.

3.6.1 Description

Transient analysis requires another level of discretization in time. The desired simulation time is partitioned into intervals for which a solution is sought at each step. Typically, to reduce CPU time, these intervals are determined based upon truncation error (*i.e.* estimated numerical error due to a time step) rather than using a set interval. As a result, if the circuit does not change its state over a time interval, then that time interval does not need to be subdivided into smaller intervals.

Both circuit and device simulation tools are required to evaluate the time derivative of currents, voltages, and concentrations in their respective equations. There are many

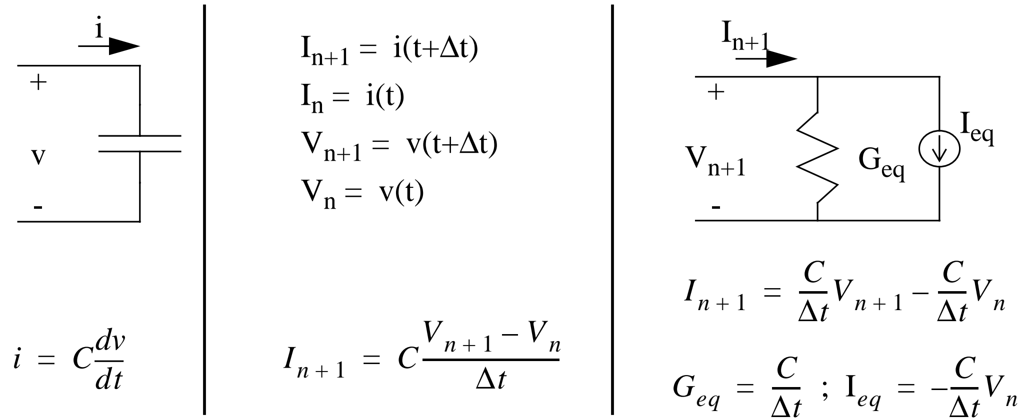


Figure 3.6: Time discretization of a capacitor using backward Euler.

approaches to the derivative approximation, but the two common methods used in device simulation are backward Euler (BE) and the combination method of a Trapezoidal step followed by a second order backward difference step (TR/BDF2). The backward Euler step is the simplest to implement, but lacks the ability for efficient time step control. The TR/BDF2 discretization technique allows for better time step control and has the advantage of re-using the Jacobian matrix decreasing the time for a solution [48].

3.6.2 Boundary Conditions

Because the circuit (and consequently any derivatives) is linear, the implementation of transient analysis with boundary conditions is straightforward. The simple case of a parallel resistor and capacitor on a electrode is discussed in Section 3.3.2 where the additional displacement current must be included in the equation formulation. The generic linear circuit requires a similar adjustment to the boundary condition equations, but the simulation approach remains the same.

For each time step, the evaluation of the derivatives of a circuit element (*i.e.* capacitor and inductor) can be represented using time independent elements (*i.e.* current sources and resistors). Referring to Figure 3.6, the relationship between the voltage across a capacitor and the current flowing through a capacitor is given by

$$i = C\frac{dv}{dt}. \qquad (3.37)$$

Using a backward Euler time step from time t_n (solution known) to time t_{n+1} (solution unknown) yields

$$I_{n+1} = C\frac{V_{n+1} - V_n}{\Delta t}. \tag{3.38}$$

Upon reorganizing the equation, it becomes obvious that it can be represented as an equivalent current source and equivalent conductance in parallel such that

$$I_{n+1} = \frac{C}{\Delta t}V_{n+1} - \frac{C}{\Delta t}V_n , \tag{3.39}$$

where

$$G_{eq} = \frac{C}{\Delta t} \text{ , and } \mathrm{I}_{eq} = -\frac{C}{\Delta t}V_n . \tag{3.40}$$

Hence, for each time step the capacitors can be reduced to two linear elements and the boundary condition equations are computed for the new linear circuit as described in Section 3.4.

In a similar approach, an inductor can be reduced to a set of linear elements as shown in Figure 3.7. In the expression for the equivalent current source, the inductor current from the previous iteration is required. The inductor current is typically not available in the circuit solution matrix. There are two approaches for adding this current into the circuit solution [22]. The first approach is to break the circuit and insert a voltage source through which the current is computed. An alternative is to added a sub-circuit from which the inductor current can be determined (Figure 3.7).

In order to have time step control in device simulation, Bank developed a two step method using a trapezoidal step (TR) followed by a second order backward difference step (BDF2) [48]. Given a time dependent differential equation

$$\frac{d}{dt}q(z(t)) + f(t, z(t)) = 0 \tag{3.41}$$

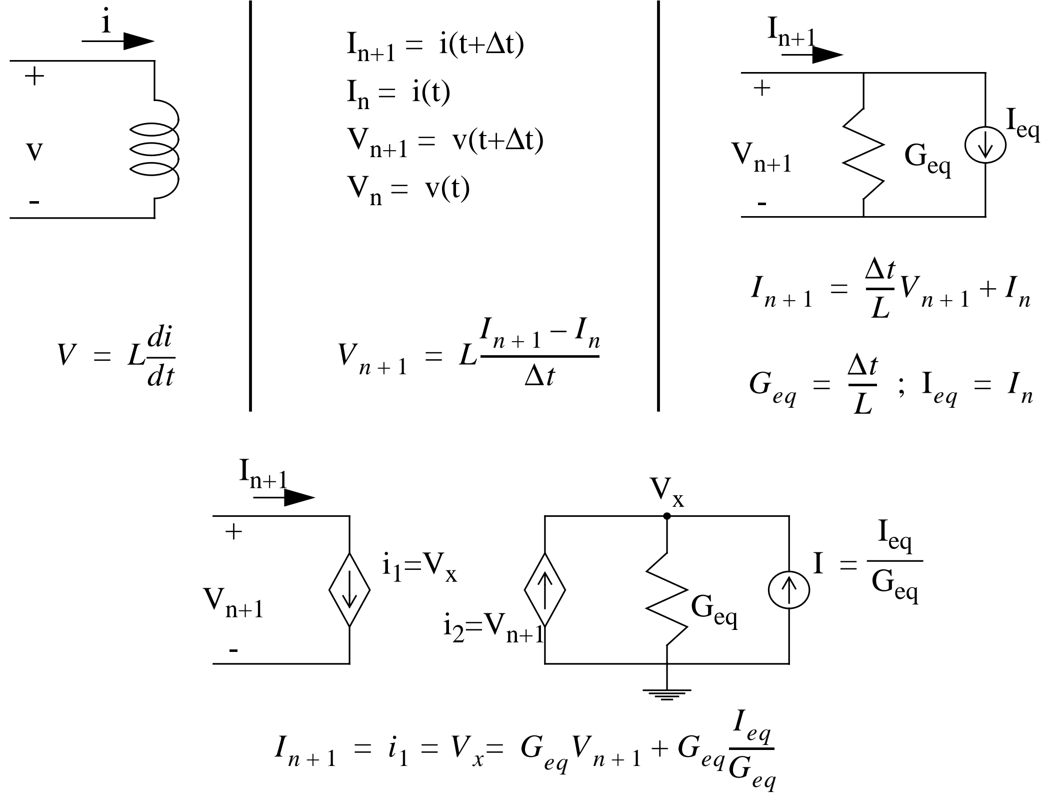


Figure 3.7: Time discretization of an inductor using Backward Euler.

and a time step h_n, a TR step is taken for a fraction of the time step (γh_n) such that

$$2q_{n+\gamma} + \gamma h_n f_{n+\gamma} = 2q_n - \gamma h_n f_n \tag{3.42}$$

and is followed by a BDF2 step of $(1-\gamma)h_n$ such that

$$(2-\gamma)q_{n+1} + (1-\gamma)h_n f_{n+1} = \gamma^{-1} q_{n+\gamma} - \gamma^{-1}(1-\gamma)^2 h_n q_n \tag{3.43}$$

where $0 < \gamma < 1.0$. The advantage of this approach becomes apparent when $\gamma = 2 - \sqrt{2} \approx 0.586$ which forces the device Jacobian (and G_{eq}) to be the same value for both the TR and BDF2 step.

The TR/BDF2 method can be applied to the simple differential equations for the inductor and capacitor to determine the values for the equivalent conductance and equivalent current source during each time step (Figure 3.8).

In order to get an accurate transient simulation, truncation error must be monitored at each time step. In addition, the next time step should be selected so as to minimize truncation error without unnecessarily small time steps which lead to excessive simulation times and potential numerical round-off error. To estimate local truncation error, a Taylor series expansion of the exact solution is compared to the solution obtained by taking a backward Euler or trapezoidal followed by a backward difference time step.

The backward Euler technique leaves an error term dependent upon the second derivative of the solution for which a solution is sought such that

$$\tau_{n+1} = \frac{h_n^2}{2} q''(\xi) \tag{3.44}$$

where $q''(\xi)$ is the second derivative of the solution which can be represented as

$$= \frac{h_n^2}{4}\left(\frac{q_{n+1}}{h_{n-1}(h_{n-1}+h_n)} + \frac{q_n}{-h_{n-1}h_n} + \frac{q_{n-1}}{-h_n(-h_{n-1}-h_n)}\right). \tag{3.45}$$

The TR/BDF method, on the other hand, has an error term proportional to the third derivative of the variable for which a solution is sought such that

$$\tau_{n+1} = C(\gamma)\frac{h_n^3}{2} q'''(\xi) \tag{3.46}$$

where $q'''(\xi)$ is the third derivative of the solution which can be represented as

$$\approx 2C(\gamma)h_n[\gamma^{-1}f_n - \gamma^{-1}(1-\gamma)^{-1}f_{n+\gamma} + (1-\gamma)^{-1}f_{n+1}] \tag{3.47}$$

and $C(\gamma)$ is a constant that depends on the value of γ.

Upon evaluating τ_{n+1} for each variable in the device and circuit, the factor r is computed over all device variables such that

Capacitors:

Diff. Eq: $\frac{dV}{dt} - \frac{I}{C} = 0$

TR Step: $I_{n+\gamma} = \frac{2C}{\gamma h_n} V_{n+\gamma} + \left(-\frac{2C}{\gamma h_n} V_n - I_n\right)$

BDF2 Step: $I_{n+1} = \frac{(2-\gamma)C}{(1-\gamma)h_n} V_{n+1} + \frac{C}{(1-\gamma)\gamma h_n}((1-\gamma)^2 V_n - V_{n+\gamma})$

$$G_{eq}(TR) = G_{eq}(BDF2) = \frac{2C}{\gamma h_n} = \frac{(2-\gamma)C}{(1-\gamma)h_n} \quad \text{for} \quad \gamma = 2 - \sqrt{2}$$

$$I_{eq}(TR) = \left(-\frac{2C}{\gamma h_n} V_n - I_n\right) \;;\; I_{eq}(BDF2) = \frac{C}{(1-\gamma)\gamma h_n}((1-\gamma)^2 V_n - V_{n+\gamma})$$

Inductors:

Diff. Eq: $\frac{dI}{dt} - \frac{V}{L} = 0$

TR Step: $I_{n+\gamma} = \frac{\gamma h_n}{2L} V_{n+\gamma} + \left(\frac{\gamma h_n}{2L} V_n + I_n\right)$

BDF2 Step: $I_{n+1} = \frac{(1-\gamma)h_n}{(2-\gamma)L} V_{n+1} + \frac{1}{(2-\gamma)\gamma}(I_{n+\gamma} - (1-\gamma)^2 I_n)$

$$G_{eq}(TR) = G_{eq}(BDF2) = \frac{\gamma h_n}{2L} = \frac{(1-\gamma)h_n}{(2-\gamma)L} \quad \text{for} \quad \gamma = 2 - \sqrt{2}$$

$$I_{eq}(TR) = \left(\frac{\gamma h_n}{2L} V_n + I_n\right) \;;\; I_{eq}(BDF2) = \frac{1}{(2-\gamma)\gamma}(I_{n+\gamma} - (1-\gamma)^2 I_n)$$

Figure 3.8: Local linearization of capacitors and inductors using TR/BDF2 time discretization.

$$r = \sqrt{\frac{1}{N}\sum_{i=0}^{N}\left(\frac{\tau_{n+1,i}}{e_{n+1,i}}\right)^2} \quad (3.48)$$

and all circuit variables such that

$$r = \frac{\|\tau_{n+1}\|_2}{\|e_{n+1}\|_2} . \quad (3.49)$$

The factor e_{n+1} is computed as

$$e_{n+1} = \varepsilon_R|q_{n+1}| + \varepsilon_A \quad (3.50)$$

where ε_R is a relative error parameter and ε_A is the absolute error parameter.

The value of r determines whether the time step is adequate and is used to predict the next time step. In the backward Euler discretization, if r is greater than or equal to one then the time step is too large and needs to be repeated with a smaller value (typically $\frac{1}{2}h_n$); otherwise the current time step is deemed adequate. Backward Euler provides no time step estimation and thus the next time step cannot be determined. In the TR/BDF2 approach, if r is greater than or equal to 1 then the time step is repeated with a new time step given by

$$h_{n+1} = h_n r^{-\frac{1}{3}} . \quad (3.51)$$

If r is less than 1.0 then Equation 3.51 gives a prediction of the next time step. Typically a "paranoia" factor can be included reducing the predicted time step by about 10% [48].

3.7 Harmonic Balance Analysis

Harmonic balance provides for large signal sinusoidal solutions of a circuit or device. It differs from transient analysis in that the solution is found in the frequency domain rather than the time domain [42].

The harmonic balance solver module for PISCES is provided by the EEsof Division of Hewlett Packard Co. in Santa Rosa, CA. The solver is designed to interface with

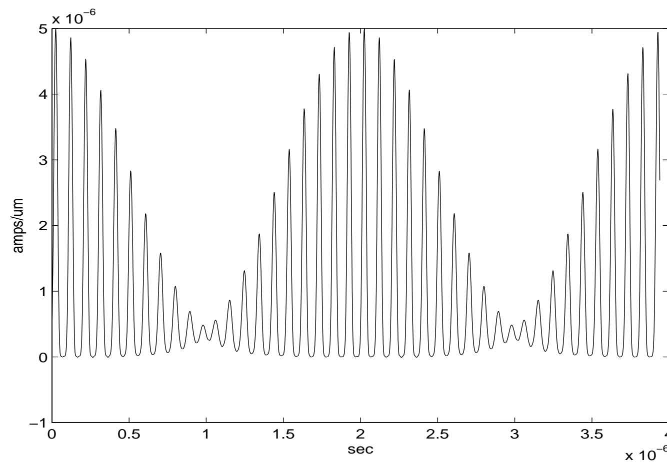


Figure 3.9: Multi-tone response of diode with an input of two closely spaced frequencies [41].

Stanford's PISCES-2HB to provide the numerical capabilities to solve the harmonic balance (HB) system of equations generated in a device simulator. This section discusses the basics of the solver and the integration with PISCES. For a full description of algorithms and techniques for solving the set of HB device equations, please refer to the works by Boris Troyanovsky [17] [49] [50] or his dissertation [41].

3.7.1 Description

Harmonic balance analysis offers computational improvements over transient analysis. It captures the steady state performance of a device in the presence of potentially long time constant phenomena. To get a steady-state solution, time domain analysis requires that the simulation time must progress far enough for the transients to die out; thus leading to long simulation times. For multi-tone analysis, harmonic balance avoids the excessive number of time steps required by transient analysis. As shown in Figure 3.9, the time steps need to be small enough to resolved the high frequency components while progressing far enough to encompass the low frequencies. Harmonic balance allows for power sweeping from low power to high power by utilizing previous solutions for the next solution whereas time domain simulation requires a restart from time equal to zero.

The harmonic balance approach takes the semiconductor equations and assumes periodic solutions (Equation 3.52) given that the input signal is periodic or quasi-periodic. Since the device is nonlinear, integer-multiple combination of harmonics to the input

frequency(ies) are created. In simulation, the frequency list is truncated to a subset of frequencies such that the excluded values are insignificant.

The independent semiconductor variables, Ψ, n, and p at each grid point are represented by the variable X such that

$$X_k(t) = X_{k,0} + \sum_{i=1}^{N} [X_{k,i}\cos(\omega_i t) + X_{k,i+N}\sin(\omega_i t)] \tag{3.52}$$

where k, refers to the solution at a specific grid point in the spatial discretization and i represents the i-th frequency of the total N frequencies included in the simulation.

The goal of the analysis is to find the quasi Fourier coefficients Ψ_{ki}, n_{ki}, and p_{ki} such that the semiconductor equations are satisfied. Once those coefficients (*i.e.* harmonics) for the basic variables are obtained, the conduction current and displacement current can be computed at the device terminals. The solution techniques for the generated system of equations becomes quite complex since the size of the matrix system increase by $2N + 1$. Troyanovsky uses knowledge of the device simulation matrix structure to exploit solution techniques to optimize simulation time and improve convergence [41].

3.7.2 Boundary Conditions for Harmonic Balance Analysis

Harmonic balance simulation presents another special case for external boundary conditions. The boundary equations have three distinctive forms to accommodate the different harmonics generated in the circuit.

For the dc components of the Fourier series, the boundary condition equation is determined as defined in Table 3.1 and repeated for the standard case:

$$G_{d0}(E_{d0} - V_{d0}) - I_{d0}(\Psi_0, n_0, p_0) = 0 \tag{3.53}$$

For frequencies at which there is a source (*i.e.* the fundamental(s)) the equation is given by a similar format except that it is complex where

$$Y_{d1}(E_{d1} - V_{d1}) - I_{d1}(\Psi_1, n_1, p_1) = 0 \tag{3.54}$$

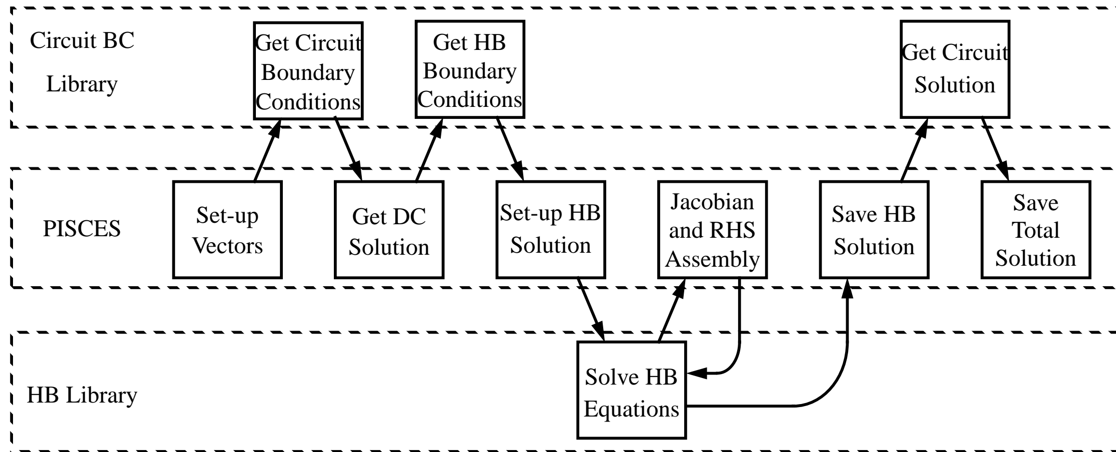


Figure 3.10: Integration of PISCES with a harmonic balance library and a circuit boundary condition library.

where the conductance matrix becomes an admittance matrix and the sources have magnitude and phases associated with them.

The non-linear PISCES devices has harmonics generated within its structure. These harmonics propagate to the external circuity where they must meet the boundary conditions imposed. The boundary conditions for these harmonics are given as

$$-Y_{dn}V_{dn} - I_{dn}(\Psi_n, n_n, p_n) = 0 \qquad (3.55)$$

and do not contain any sources since the external circuit does not generate any new harmonic tones.

These three variations of boundary conditions establish the solution for the harmonic balance problem being solved.

3.7.3 Integration of HB Solver and Circuit BC

Circuit boundary conditions and harmonic balance simulation are integrated with a device simulator such as PISCES through a set of libraries (Figure 3.10). PISCES calls the circuit boundary condition routines to generate the boundary equations from a SPICE-like netlist. It then calls the harmonic balance solver, passing the boundary conditions and PISCES state variables. The harmonic balance solver uses the PISCES assembly routines to generate the Jacobian and RHS at each frequency in the Fourier series expansion and for

each iteration. Once the convergence criteria are met, the solution is passed back to PISCES which calls a circuit solver to get the complete solution.

3.8 Conclusions

This chapter addresses the necessary components for RF and device simulation of discrete structures. The standard boundary conditions for PISCES are described. These boundary conditions are expanded to include a generalized linear circuit and can be used with the analysis most important to high speed, RF, and opto-electronic device simulation. In addition, a new device simulation analysis technique (harmonic balance simulation) is described and references are provided on its implementation. The inclusion of circuit boundary conditions and harmonic balance allows for in-depth analysis of devices and the parasitic components associated with them.

Chapter 4: Opto-Electronic, RF, and High Speed Component Optimization

4.1 Introduction

There are many cases where device engineers need to improve the performance of a structure in its application. This chapter describes three examples addressing opto-electronic, RF, and high speed components optimized in the environment in which they function.

Three examples demonstrate different types of optimization utilizing mixed circuit-device simulation. This work does not address the designs in full detail because of the complexity involved, but rather examines a part of a design for which a single engineer may be responsible.

- Design for Optimal Optical Performance
  The next generation of an LED structure used in an optical communication circuit requires an analysis of the internal physics with surrounding drive circuitry. The doping and band-gap of the layers in the device structure are optimized for optical performance.

- Design for Optimal Layout
  The design of high frequency MESFET devices is highly dependent upon the layout and the parasitic components associated with that layout. The optimization of the device examines different layouts and the impact on high frequency performance.

- Design for Optimal Process Flow
  Deep sub-micron development has consistently pushed the physical limits of semiconductor technology. With each new generation, new physical limitations become apparent and thus require analysis and optimization. In this example, the impact of poly-depletion is address and process optimization is used to minimize the effect.

For each example, the important physics of the device are explained. A simulation approach is defined followed by some sample optimization results. Each example will ultimately need to be tailored to the process flow and structure of alternative designs.

4.2 Design of LED with Floating Layer

In collaborative work with the Optical Components Division of Hewlett Packard in San Jose, a redesign of an LED was completed with the PISCES device simulator. The purpose of the design change involved creating thermally stable AlGaAs by oxygen implantation [51]. The approach began by first characterizing the original structure. Upon verifying that the simulation model accurately reflected the performance of the device, variations based on the new design are studied. From the simulation of the new designs, an optimized structure is obtained.

4.2.1 Analysis of Original Structure

A cross section of the cylindrical device structure is provided in Figure 4.1. The device is created in a III-IV system of GaAs with various mole fractions of Al or In to obtain the desired band gap profile. The InGaAs layer sandwiched between two AlGaAs layers is designed to create a potential well to collect carriers. The carriers then recombine through radiative recombination producing photons with a wavelength of 820nm.

The cylindrical structure lends itself to application as a transmitter for a fiber optic communication system. Current is confined by the N^+ floating layers on the p-GaAs side of the junction. The confinement increases the concentration of electrons in the center part of the InGaAs layer (thus producing more radiative recombination) and establishes the spot size of the optical output. The cathode contact of the device is offset from the center of the device to prevent reflections of photons back into the structure.

The simulation of this structure requires some special considerations for the physics of the device and the boundary conditions required for convergence. The recombination

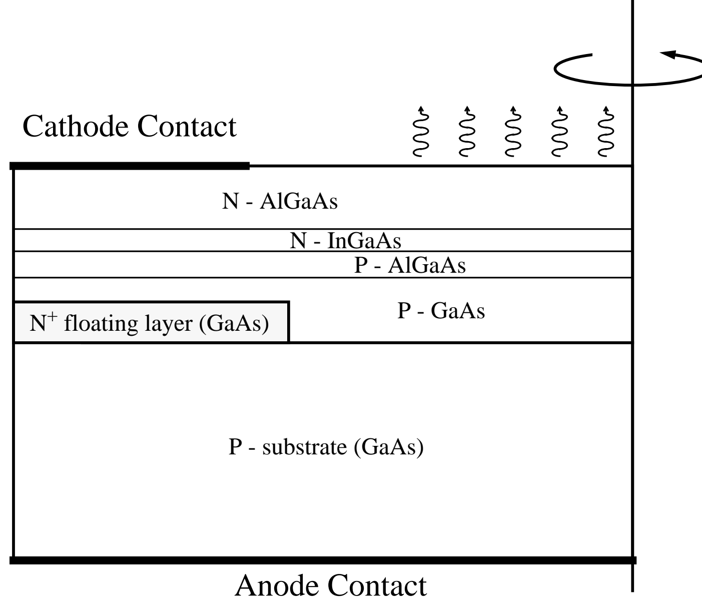


Figure 4.1: Cross section of original LED structure.

mechanisms play an important role in the device functionality. Three physical mechanisms contribute to the recombination of carriers: Shockley-Read-Hall (SRH), Auger, and radiative [43]. Shockley-Read-Hall recombination occurs through traps in the middle of the band gap. The SRH model is

$$U_{SRH} = \frac{np - n_i^2}{\tau_n[p + n_i e^{(E_i - E_t)/(k_B T_L)}] + \tau_p[n + n_i e^{(E_t - E_i)/(k_B T_L)}]} \tag{4.1}$$

where n and p are the electron and hole concentrations, n_i is the intrinsic carrier concentration, E_i is the intrinsic Fermi level, E_t is the trap energy, T_L is the lattice temperature, and τ_n and τ_p are the carrier lifetimes which are doping dependent [52] [53].

Auger recombination for electrons occurs when a high energy electron interacts with another electron to release its energy and cause the second electron to recombine with a hole [54] and likewise by a similar process with holes. The model for Auger recombination is

$$U_{Aug}(n, p) = (c_n n + c_p p)(np - n_i^2) \quad \text{(Eq 4.2)}$$

where c_n and c_p are empirical parameters dependent upon the material.

The most important recombination mechanism in this application (*i.e.* opto-electronic devices) is that for the radiative recombination which creates a photon through the process [55]. The model is given by

$$U_{rad} = B(np - n_i^2) \qquad \text{(Eq 4.3)}$$

and is dependent upon the product of the carrier concentrations (n and p) in excess of the intrinsic carrier concentration. The lifetime of the carriers, doping, and material properties are incorporated in the empirically determined factor B.

The boundary conditions for the LED structure are more complex than applying constant voltage sources. The structure is designed to conduct high current in order to supply the large number of carriers necessary to produce an adequate optical output, Thus, it is important to include the extrinsic resistances in the dc simulation in order to account for potential drops. The primary resistive component comes from the substrate to the backside anode contact.

The N^+ floating region presents a difficult problem for numerical simulation. Convergence is hindered because the Fermi levels in this region do not settle easily. To provide better stability, an additional boundary condition is imposed on the floating layer by attaching a current source with a value of zero. Since the current source is set to zero, it becomes an open circuit thereby not affecting the solution but does help settle the Fermi potentials.

There are three key parameters associated with this LED structure. The "on" voltage (V_f) is the forward bias required for the device to conduct 60mA of current for the cylindrical structure with a set diameter. The "on" resistance (R_{on}) determines how much power is dissipated during the operation of the device. The optical spot size determines the dispersion of photon output and is related to the ability to transfer the signal to a fiber optic cable.

Before a redesign is initiated, the accuracy of the device simulations are calibrated to measured data. The comparison allows for the tuning of the simulator and model, but more importantly determines the relationship between measured and simulated results. Once this relationship is understood, the simulation of the new structure can be used to predict trends in performance and ultimately an optimized new structure.

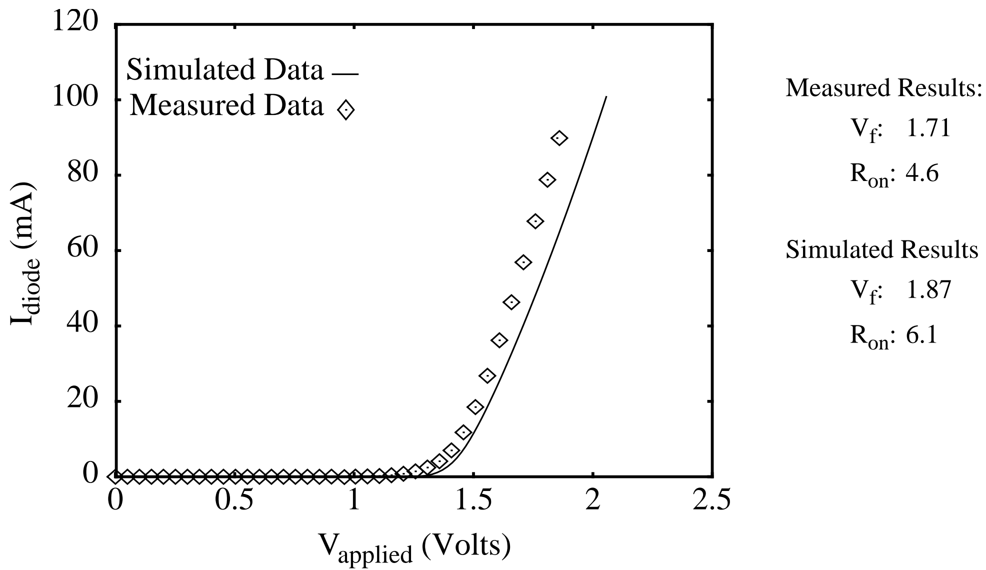


Figure 4.2: Measured and simulated I-V characteristics for heterostructure LED with a semiconductor floating layer to improve carrier confinement.

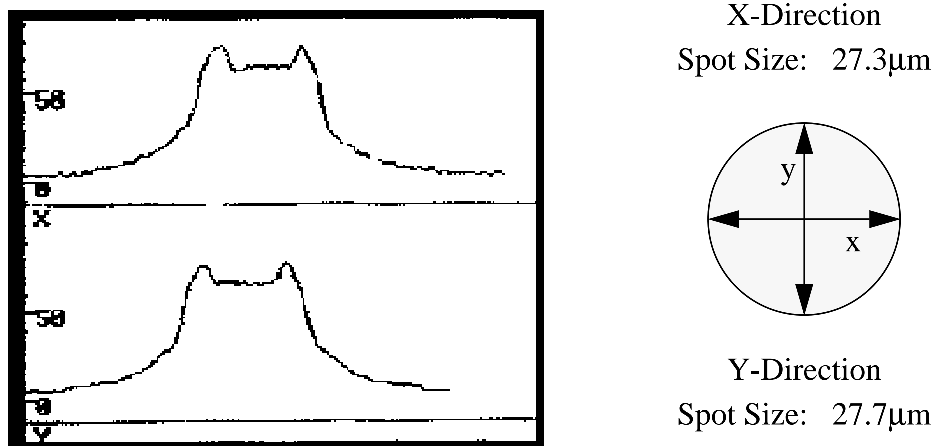


Figure 4.3: Measured optical profile for LED structure.

The determination of the forward voltage and resistance is computed from the I-V characteristics (Figure 4.2). The simulated results slightly over predict both V_f and R_{on}, but the error is within reasonable bounds for performing the analysis necessary in this work. Improvements to the model accuracy need to come from the material characterization and especially the impact of the etch which can affect the interface traps [56] [57].

The key design characteristic is the optical profile of the device as shown from measurements in Figure 4.3. Since PISCES does not contain an optics solver, the optical

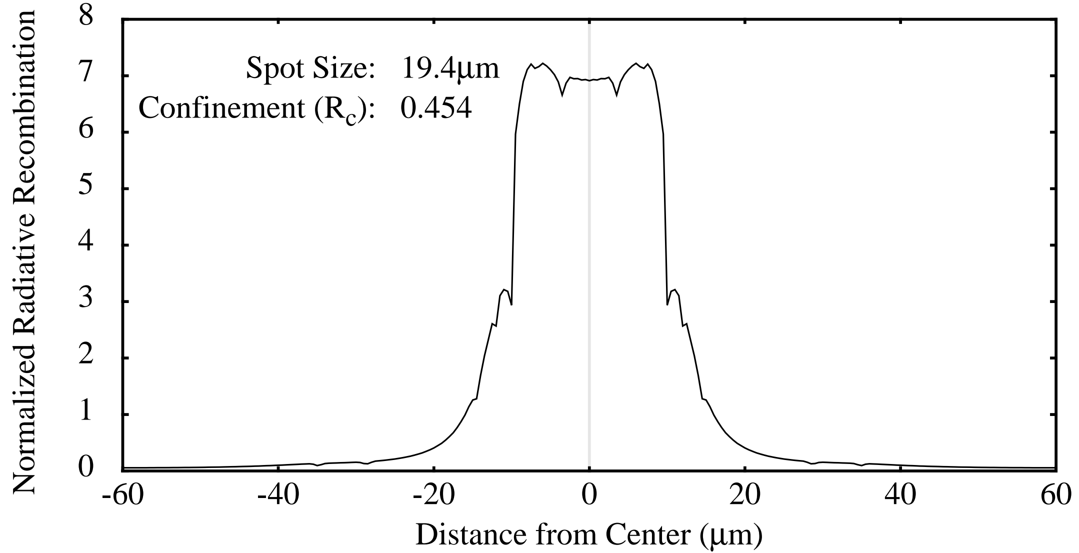


Figure 4.4: Integrate radiative recombination in the InGaAs region of the simulated LED structure.

profile can not be determined directly. To estimate the optical performance of the device, the amount of radiative recombination can be computed in the InGaAs photon generation layer as shown in Figure 4.4. The profile of the recombination across that layer correlates well with the measured spot size.

Since the new structure is a re-design of the original, other characterization factors of quality can be ascertained by studying the recombination in the structure. Therefore, a figure of merit termed the confinement ratio and represented by the variable R_c relates the ratio of radiative recombination in the InGaAs region to the total recombination:

$$R_c = \frac{\text{Radiative Recombination in InGaAs Layer}}{\text{Total Recombination in Structure}} \qquad \text{(Eq 4.4)}$$

and is related to the efficiency of the device.

4.2.2 Design Optimization

Having defined and characterized the original LED structure, trade-offs in the new structure are studied. The new structure moves the InGaAs photon generating layer to a position below the N^+ floating layer as shown in Figure 4.5 and attempts to maintain similar characteristics. Hence, many questions arises as to the layer thicknesses, doping,

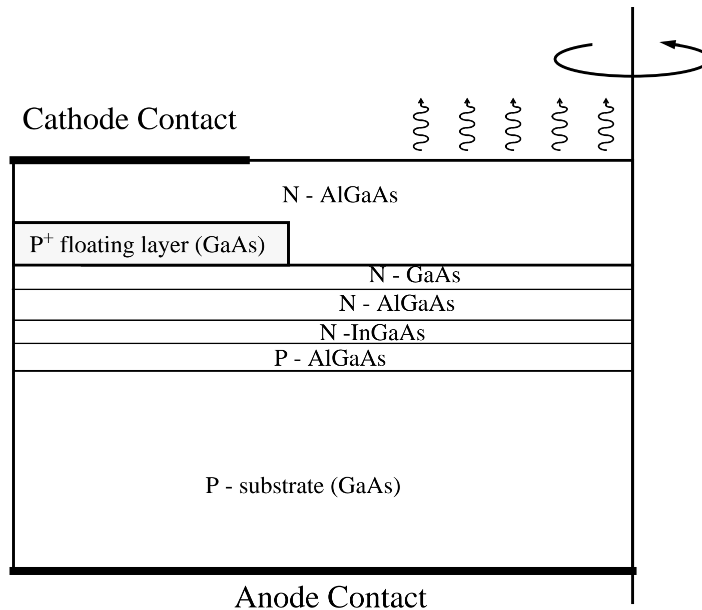


Figure 4.5: Cross section of the redesign of the LED structure.

and mole ratios of Al to Ga. A design matrix is established and multiple simulations are performed to determine the sensitivity to the structure parameters.

One layer of significant interest is the n-AlGaAs layer below the P^+ floating layer. Because an etch is required, it is best to have the InGaAs layer as far as possible from the floating layer. Therefore, variations on the layer are studied (Table 4.1) where the actual values have been removed for confidentiality.

Through statistical analysis, the best new structures can be selected although the variable interactions are quite complex. The most important parameter is the spot size and confinement ratio for the device. A lower doping (Doping Y), a lower thickness (Thickness B), and a higher mole ratio of aluminum (%Al N) tended to offer better performance. The reason for this performance enhancement can be associated with the current spreading in the device as shown in Figure 4.6. A higher doping (*i.e.* lower resistance) with a thicker region and lower mole fraction (higher mobility) allows the current to spread more.

The best design though, is determined by manufacturing considerations and turns out to be structure #3 with a thickness of A, a doping of Y, and mole fraction M. Figure 4.7a shows the simulated structure of the best design and Figure 4.7b shows the experimental result demonstrating good agreement.

No.	Thick	Doping	%Al	V_f	R_{on}	W_{spot}	R_c
1	A	X	L	1.81	6.1	28.2	0.421
2	A	Y	L	1.91	6.9	25.9	0.478
3	A	Y	M	2.06	8.5	24.0	0.451
4	A	Y	N	2.13	8.6	25.9	0.478
5	A	X	N	1.88	6.6	26.6	0.480
6	B	Y	L	1.89	6.5	26.7	0.514
7	B	Y	M	2.00	7.7	24.4	0.467
8	B	Y	N	2.05	7.8	24.3	0.466
9	B	X	N	1.87	6.4	27.3	0.525
Simulated Old Design				1.87	6.1	19.4	0.454
Measured Old Design				1.71	4.6	27.0	n/a

Table 4.1: Simulated experiments to study the impact of design parameters on the n-type AlGaAs layer in a new LED structure.

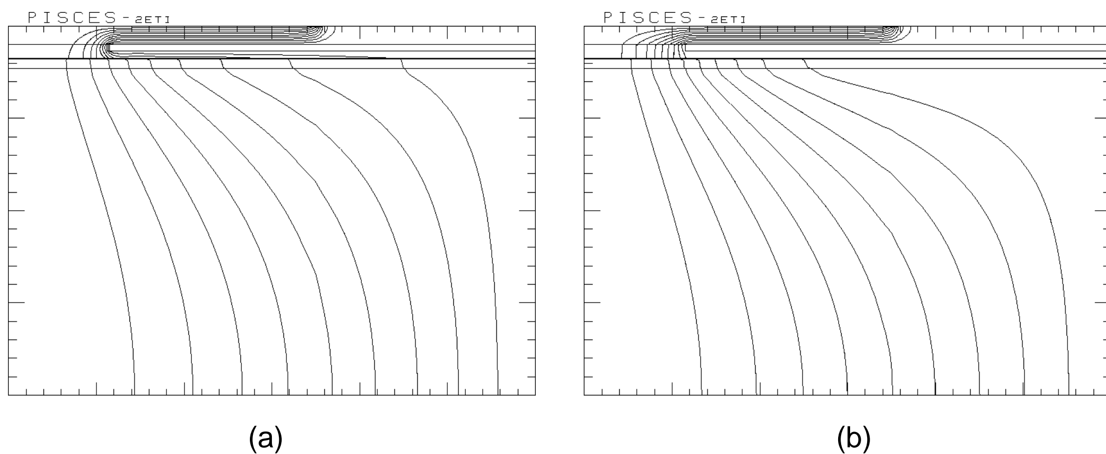


Figure 4.6: Current flow lines for two structures of a re-design of an LED structure for (a) low resistance n-AlGaAs layer and (b) high resistance n-AlGaAs layer.

4.2.3 LED Communication Circuit

The LED design is targeted for use in an fiber optic system as a digital transmitter shown in Figure 4.8. The circuit takes a TTL digital circuit output and buffers it in ECL gates

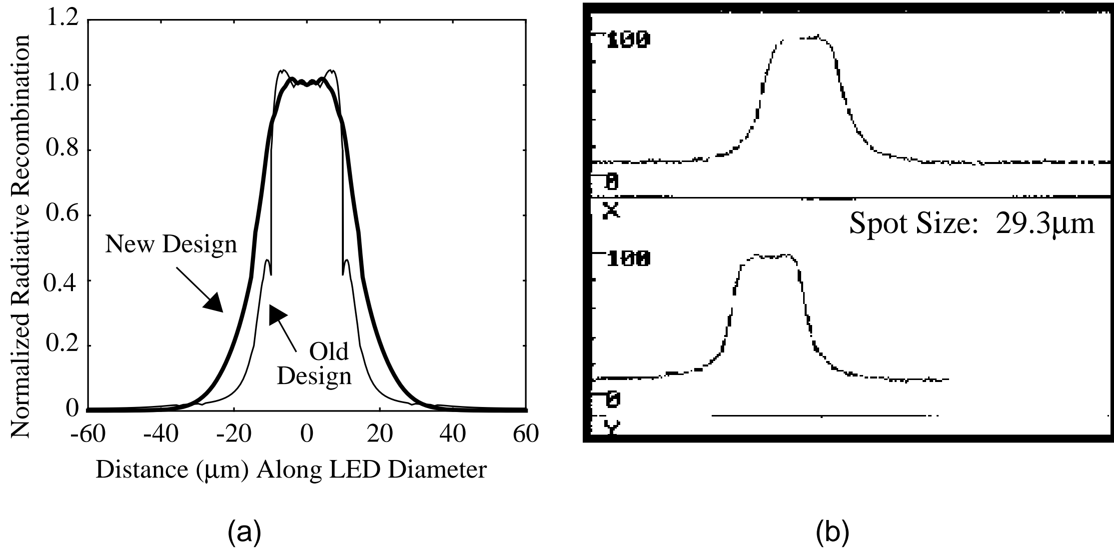


Figure 4.7: (a) Simulated recombination profile in active layer of new LED structure compared to the original structure and (b) the experimental optical output of the new structure.

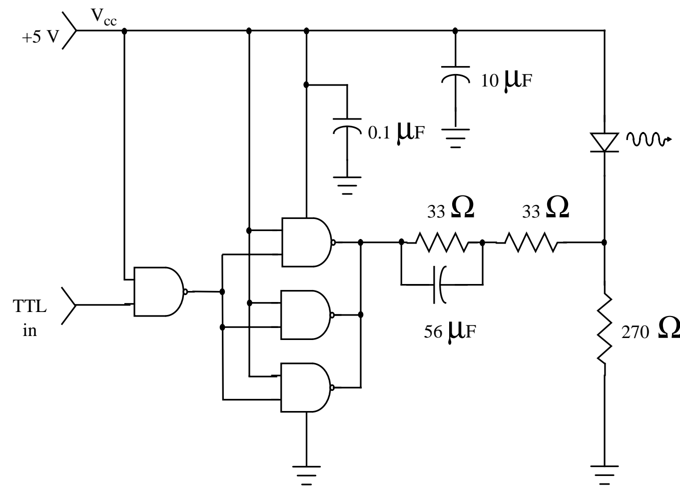


Figure 4.8: Fiber optic transmitter circuit

which are capable of supplying/sinking the current required by the LED [58] [59]. The important characteristics for the device is the switching speed and settling time for the optical output.

Mixed circuit-device simulation can characterize the switching behavior [60]. Since the optical output of the device can not be directly characterized, an alternative figure of merit is the current and summed radiative recombination. The current is correlated with the

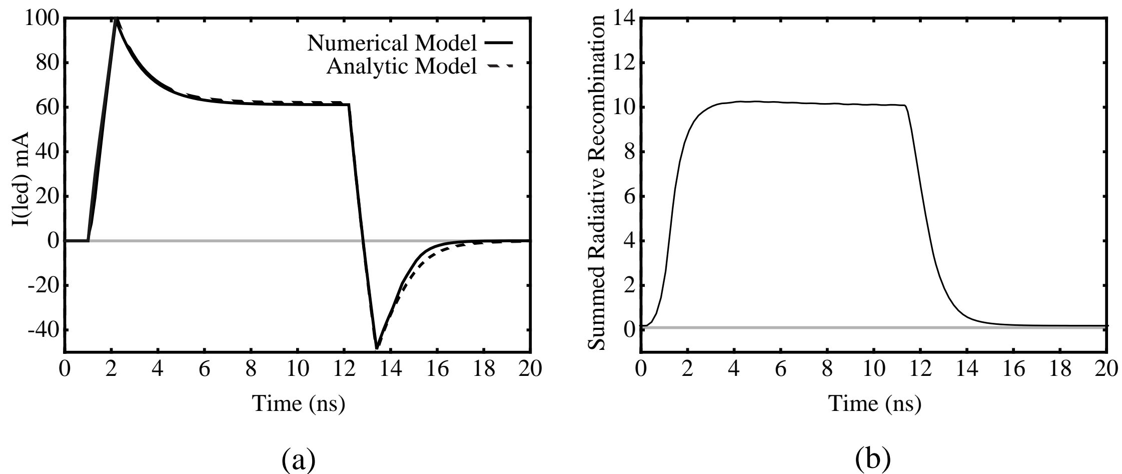


Figure 4.9: LED response of a transmitter circuit: (a) I-V characteristics and (b) summed radiative response.

optical output since it determines how many carriers are injected into the device. The summed radiative recombination is the number of recombining carriers that could potentially create a photon. Figure 4.9a shows the transient response of the current flowing through the original structure for the LED. The physics based simulation result is compared with an empirically extracted circuit model for the device. The circuit model is well tuned to the measured data and thus accurately represents the response. More importantly, an estimate of the optical performance is obtainable from the physics based simulation as shown in Equation 4.9b.

4.3 Impact of Layout on RF MESFET Performance

As devices shrink and operating frequencies increase, no longer is the device speed itself the limiting factor. All the surrounding interconnect and packaging parastics must be included in order to accurately model and study the device design. Hence, linear circuit boundary conditions become critical for obtaining meaningful results.

4.3.1 Design Requirements

This section discusses a simple RF application and addresses the impact of parasitic components on the performance of high frequency devices. For this example (a more detailed example is presented in Chapter 5), a MESFET (Figure 4.10) provided by Matsushita Electric Industrial Co. of Japan is considered. The design of the device is optimized by the recessed gate and a judiciously selected doping profile to improve RF

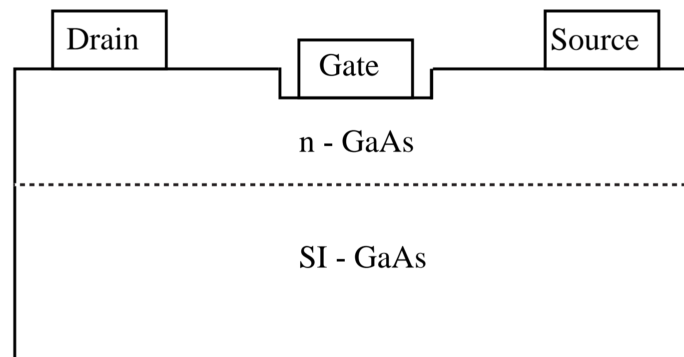


Figure 4.10: Cross-section of MESFET used in RF communication circuits.

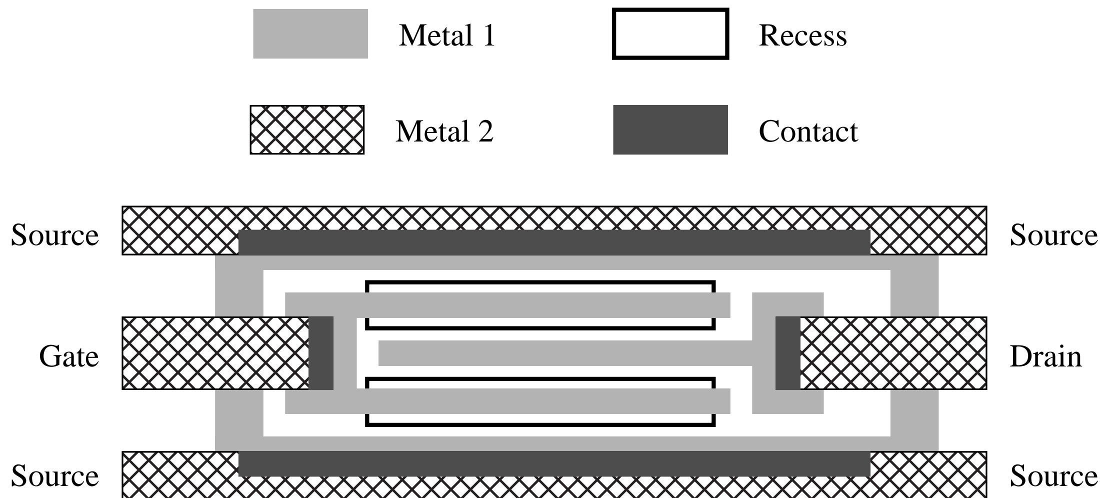


Figure 4.11: Layout of MESFET used in RF communication circuits.

performance [61]. In order to obtain the large gate width required to handle the high currents in RF applications, the layout is created with a multi-fingered structure using two levels of metal as shown qualitatively Figure 4.11 for a structure with 2 gate fingers. An important consideration is the impact of the layout and the intrinsic parasitic components in order to accurately characterize the device and layout.

The parasitic components can be determined by a variety of methods, but should be based upon the layout. For example, using 3D solid modeling and knowledge of the process technology, the layout can be represented in three spatial dimensions [62]. For example, Figure 4.12 shows a simplified structure as the gate metal drops into the recess. The geometric model can then be fed into an electro-magnetic solver (*i.e.* FastCap and

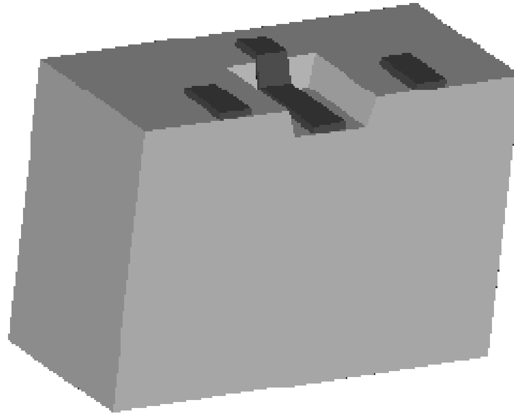


Figure 4.12: 3D solid model of the gate region of an RF MESFET.

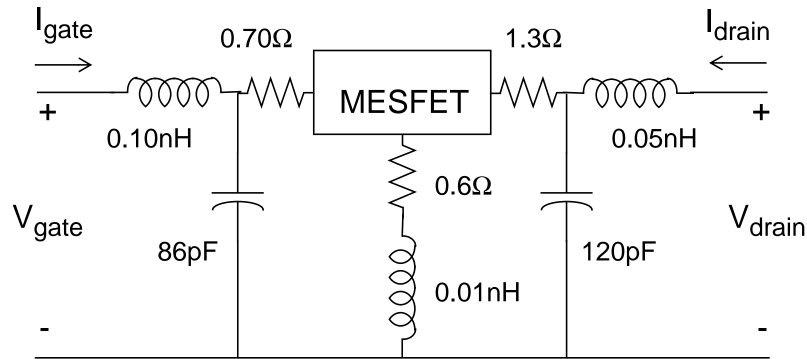


Figure 4.13: Model for MESFET layout with intrinsic device and external parasitic components

FastHenry) [63] [64] from which parasitic components are computed to fit the model in Figure 4.13. A mixed circuit-device simulation provides small signal RF characterization through S-parameter analysis.

4.3.2 Device and Layout Optimization

S-parameters are the best means to characterize the device structure; however, they can not be simulated directly. Y-parameters are easily simulated and with a simple transformation

$$\begin{bmatrix} S_{11} & S_{12} \\ S_{21} & S_{22} \end{bmatrix} = \begin{bmatrix} 1+y_{11} & y_{12} \\ y_{21} & 1+y_{22} \end{bmatrix}^{-1} \begin{bmatrix} 1-y_{11} & y_{12} \\ y_{21} & 1-y_{22} \end{bmatrix} \quad \text{(Eq 4.5)}$$

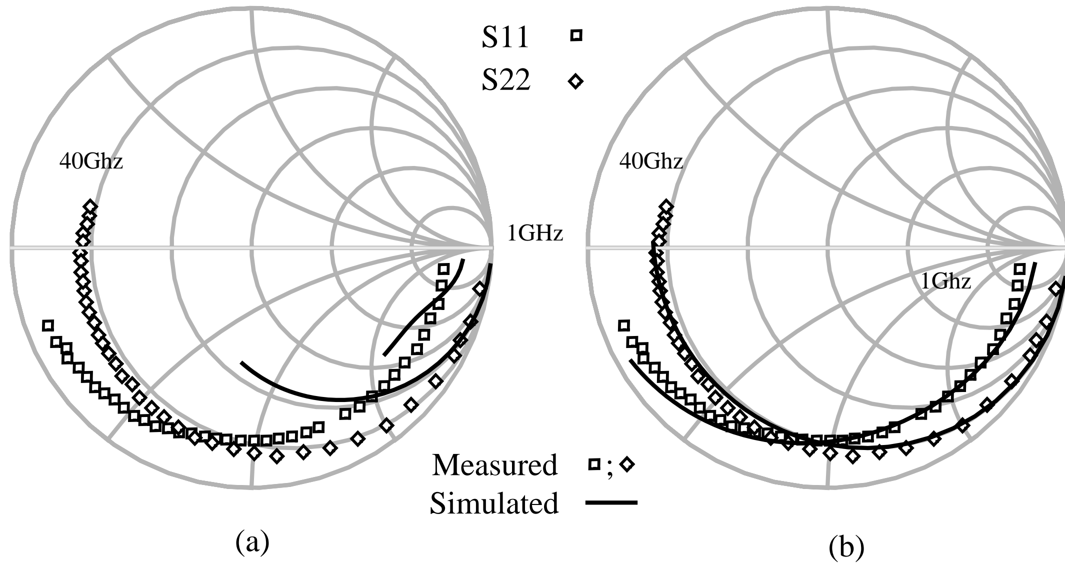


Figure 4.14: Comparison of MesFet performance (a) with and (b) without parasitic components.

yields the S-parameters. Using the definitions in Figure 4.13, the Y-parameters are determined by applying a small signal perturbation on the gate or drain while shorting the other. They are given by

$$\begin{bmatrix} Y_{11} & Y_{12} \\ Y_{21} & Y_{22} \end{bmatrix} = \begin{bmatrix} \frac{I_{gate}}{V_{gate}} & \frac{I_{gate}}{V_{drain}} \\ \frac{I_{drain}}{V_{gate}} & \frac{I_{drain}}{V_{drain}} \end{bmatrix} \qquad \text{(Eq 4.6)}$$

and normalized by the characteristic admittance of 0.02 mhos (*i.e.* an impedance of 50Ω) such that

$$y_{ij} = \frac{Y_{ij}}{Y_o} . \qquad \text{(Eq 4.7)}$$

The impact of the layout becomes apparent when simulations are performed with and without the parasitic components as shown in Figure 4.14. The two simulations clearly indicate that parasitic elements play an important role. Failure to include the appropriate parasitic elements yields incorrect results and consequently an incorrect evaluation of device performance.

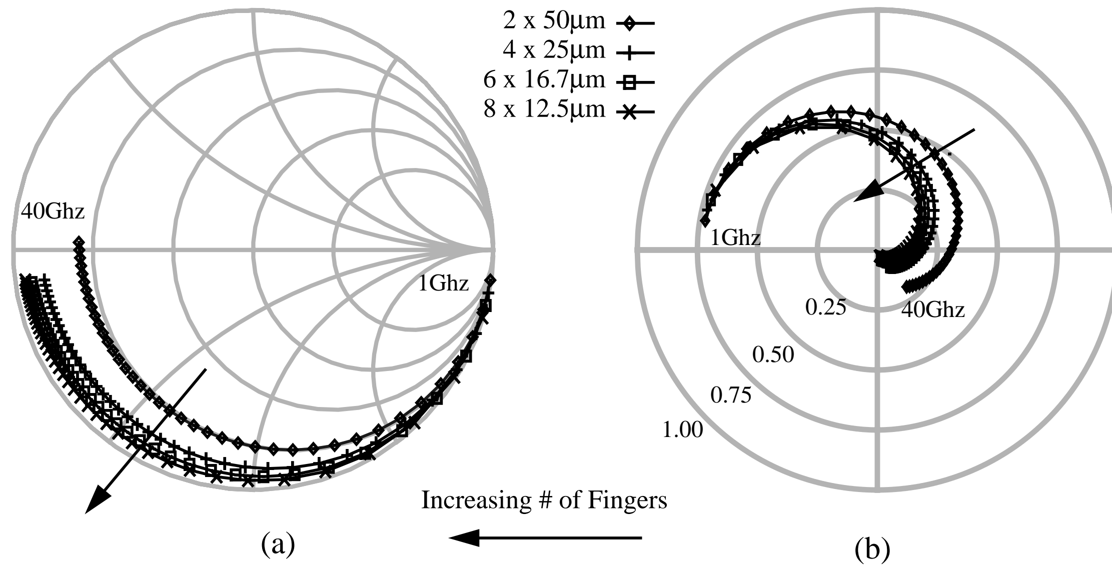


Figure 4.15: Comparison of the (a) input impedance of a MESFET (s11) and (b) the gain of MESFET (s21) for a constant gate width of 100µm using a different number of fingers.

Having a fully coupled layout extraction and device simulation tool allows a design engineer to develop and optimize the new structure. For example, one may want to optimized the number of fingers on the structure. For a constant gate width of 100µm, the device can have 2 fingers of 50µm each (2 x 50µm), 4 fingers of 25µ each (4 x 25µm), 6 fingers of 16.7µm each (6 x 16.7µm), or 8 fingers of 12.5µm each (8 x 12.5µm). The reduction in finger lengths lead to a reduction in the resistance and inductance associated with the drain and gate interconnect. The cost for such a change is an increases in the capacitance from the source to the gate and drain since a second level metal (source) must cross over the first level metal. In general, for a doubling of the number of fingers, the capacitance will double while the finger inductance and resistance is reduced by a factor of two.

The S-parameters for the different layouts are shown in Figure 4.15 for a frequency sweep of 1Ghz to 40Ghz. For this example, the input impedance (S11) is important for the matching network used in the application while the unmatched gain of the device (S21) is important for performance. As the number of fingers increase, the gain of the device is reduced at higher frequencies because of the losses associate with the capacitance to ground (*i.e.* source). Also, at higher frequencies, the input impedance increases which affects the design of the input matching network. As a result, a design trade-off has to be made to determine the best layout for the target application.

4.4 Switching Speed in CMOS Inverters

Digital CMOS circuit continue to shrink in order to obtain better performance, faster speeds, and higher packing densities. Current microprocessor technology operate in the 500 MHz clock frequency range and is already moving toward 800 MHz. With shrinking geometries, many intrinsic physical limitations and effects are beginning to dominate. As a result, the device design engineer is required to develop devices that meet circuit performance guidelines with ever more stringent physical limitations on the device.

The example in this section shows how mixed circuit and device simulation can be used to evaluate delay time and switching speed of CMOS inverters so that devices are optimized for digital performance. Specifically, a method for simulating ring oscillators at the physics based level allows for the proper characterization.

4.4.1 Problem Description

State of the art digital CMOS transistors typically use a heavily doped poly-silicon gate structure in order to reduce the resistance of the gate and to obtain an adequate work function. In order to dope the poly-silicon, a standard process sequence calls for an implant after the poly-silicon deposition or during the source and drain formation [65]. This implant has to be shallow enough so as not to go through the poly-silicon and into the thin gate oxide or even into the underlying silicon. The subsequent thermal steps are expected to activate the dopant atoms and diffuse them throughout the poly-silicon leading to degenerately doped the layer.

In reality, device changes make the doping of poly-silicon more difficult. In order to get the shallow junctions required for deep sub-micron devices, the number of thermal cycles are significantly limited. Without high temperature thermal cycles the dopant does not get activated nor does it diffuse throughout the poly-silicon. As a result, the poly silicon does not necessarily become degenerately doped near the poly-oxide interface [6].

4.4.2 Impact of Non-Degenerately Doped Poly Silicon

With the doping in the channel region increasing and the reduction in active dopant in the gate region, the poly-silicon can have a depletion layer near the poly-gate and gate oxide interface under positive gate bias (for NMOS). This phenomena was first observed in anomalous C-V characteristics by Lu [66] as shown in Figure 4.16. Lu investigate potential causes including a Schottky-like diode at the silicide/poly barrier, high density of

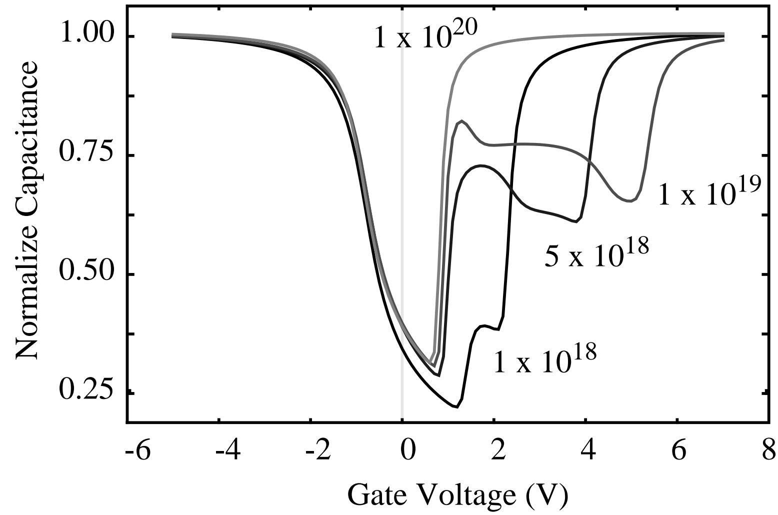


Figure 4.16: Anomalous I-V characteristics caused by poly silicon depletion in the gate of a MOSFET for a channel doping of 5 x 10^{17} and various silicon gate doping levels (N_p).

interface traps at the oxide/poly interface, non-uniform dopant distribution in the poly, and insufficient activation of dopant at the poly/oxide interface. He concluded with his analysis and experiments, that the primary cause could be attributed to the later of those hypotheses. Habas [67] later developed a physics based model using semiconductor device theory by assuming depletion layers in both the silicon channel and poly gate as shown in Figure 4.17. Other authors have characterized the impact of poly-depletion on the dc current-voltage relationship, overlap capacitance, and gate capacitance [68]-[72].

In order to determine the impact on circuit performance, a model can be developed and incorporated into the circuit MOSFET model as described by Arora [68]. In their model development, the expressions for the dc and ac parameters are modified in the PCIM model [73] to account for the additional depletion region. The dc expression for threshold voltage increases to account for the potential dropped in the poly silicon such that

$$V_{thp} = V_{fb} + 2\phi_f + \frac{Q_b}{C_{ox}} + \frac{1}{2a_v}\left(\frac{Q_b}{C_{ox}}\right)^2 \qquad \text{(Eq 4.8)}$$

where V_{fb} is the flat band voltage, ϕ_f is the bulk Fermi potential, Q_b is the bulk charge density, C_{ox} is the oxide capacitance, and

Figure 4.17: Physical representation of poly-depletion in a MOS structure.

$$a_v = \frac{q\varepsilon_o\varepsilon_{si}N_p}{C_{ox}^2} \qquad \text{(Eq 4.9)}$$

where N_p is the poly-silicon doping. With a higher threshold voltage, the drain current is lower because of the charge reduction in the channel region. In an analogous analysis, the ac expression of the capacitances is reduced to account for the additional depletion layer thickness in the poly-silicon region.

Using circuit simulation, Arora characterized the impact of poly-depletion on inverter delays for different bias conditions and loads. He found that even though the RC time constant was due to the reduction in the gate capacitance, the reduction in drain current had a stronger impact. Thus, it became important to have fully characterized circuit model in order to evaluate the impact of poly-depletion.

4.4.3 Design Optimization

The poly-depletion effect can be characterized at the device level through C-V and I-V measurements, but requires circuit simulation to truly evaluate its impact on performance. Hence, three levels of simulation are required for the CMOS technology development:

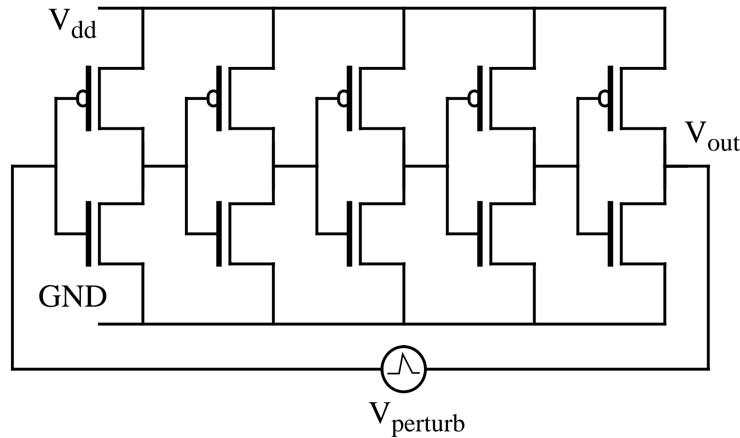


Figure 4.18: Circuit diagram of ring oscillator used for simulation.

process simulation to determine doping profiles, device simulation to determine electrical characteristics, and circuit simulation to determine performance characteristics. Since, the impact of poly-depletion is dependent upon the doping profile, it is best to use the structure generated in the process simulation for both the device and circuit simulation. This approach results in direct feedback to the process development team so that the device structure can be optimized within a process limitation (*i.e.* thermal budget).

To adequately simulate the poly-depletion effect, it is important to model the poly silicon layer in both the process and device simulator. Physics based modeling requires that the dopant activation is correctly computed in the process simulation based upon grain boundaries and traps [74]. The device simulator typically represents gate structures as a degenerately doped electrode with a specified work function, but for this case, a physics based model is required. Poly-silicon can be modeled as silicon where both the lifetime and mobility of carriers are significantly reduced thus providing an adequate physical representation of the material.

The characterization of the switching speed and delay time requires circuit simulation. Ring oscillators like that shown in Figure 4.18 are typically used in process development to characterize the speed and delay of gates. An odd number of inverters are connected in a ring where the output of one feeds the input of another. When biases are applied, a physical circuit oscillates because of slight non-uniformities in each device structure caused by process variations. In simulation, the oscillations needed to be induced by a perturbing source. This source causes a slight difference in potential between the input and

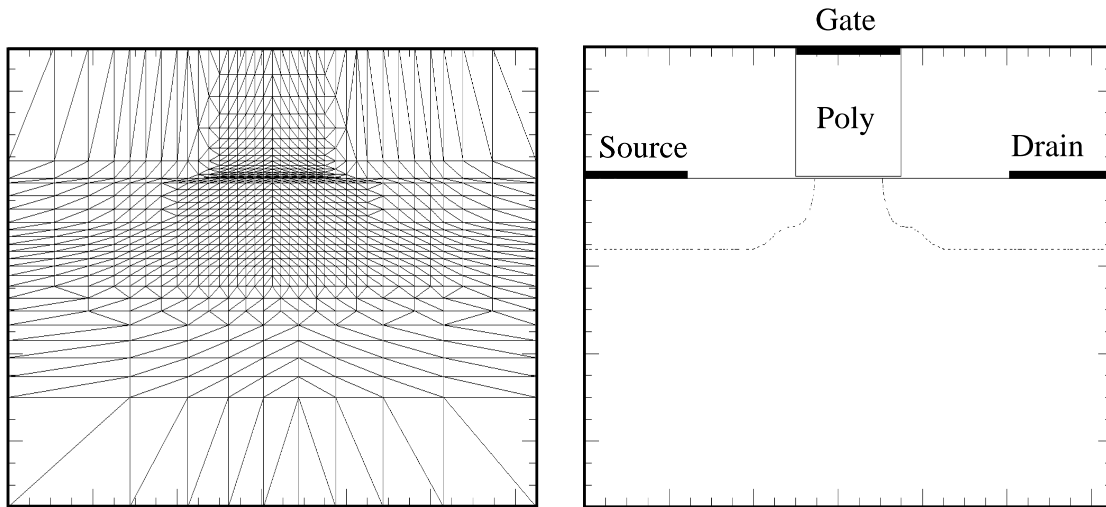


Figure 4.19: Cross sectional doping profile and mesh used for a simplified transistor to analyze poly-silicon depletion effects.

output of adjacent devices. The difference magnifies with each time step until steady state oscillations are reached after which the perturbing source acts as a short circuit.

Mixed circuit and device simulation of a ring oscillator is a challenging problem. Approximately 900 nodes per device is required to model the transistors effectively in the device simulator as shown in Figure 4.19. Grid is provided in the poly-silicon region (and especially near the channel region and thin gate oxide) in order to solve accurately for the electron and hole concentrations, thus leading to depletion layers that might form in the region. The poly-silicon region is modeled as silicon in the device simulator; however, the material characteristics are modified to account for the lower mobility and higher recombination rates. With 10 devices (*i.e.* 5 gates with 2 devices per gate) in the circuit, it is best to simulate this circuit using a parallel version of the two-level Newton algorithm. Using a network of ten RS6000 CPU's of various computation power as might be found in an industry CAD group, this simulation required an overnight run.

To demonstrate the impact of poly depletion and the feasibility of design optimization using mixed circuit and device simulation, some simulations are performed on a 0.25µm process using an oxide thickness of 8nm with a channel doping of 5×10^{17} cm^{-3} for the NMOS device and 3×10^{17} cm^{-3} for the PMOS device. As a basis, a simulation with degenerately doped poly-silicon at 1×10^{20} cm^{-3}, yields a delay time of 42.2ps per gate as shown in Figure 4.20. Dropping the doping by an order of magnitude increases the delay

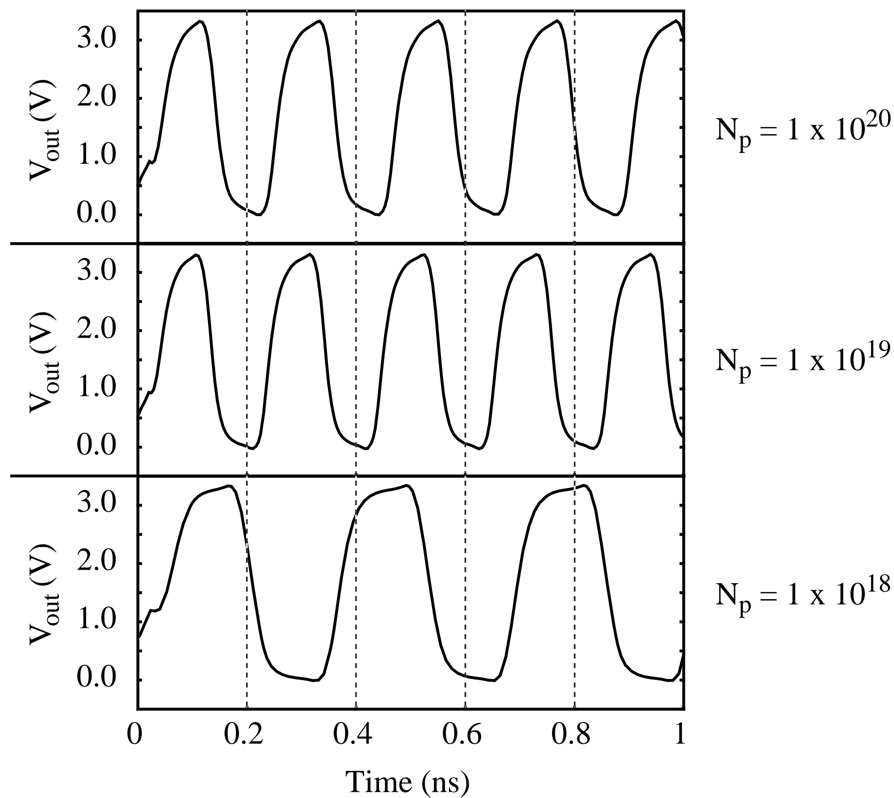


Figure 4.20: SImulated response of 5 stage ring oscillator in structure with different doping in the poly-silicon

time to 45.6ps and dropping it another order of magnitude to 1 x 10^{18} increases the delay time to 67.4ps per gate.

These simulations assumed uniform doping in the poly-silicon gate. In reality, the doping profile is going to be impacted by the process flow. In order to obtain the shallow junctions required by deep sub-micron processing, the thermal budget for the process is limited to rapid thermal processing (RTP). Using process simulation [75] with the appropriate models for poly-silicon grain growth [76] [77], grain diffusion [78], and activation of dopants in poly-silicon [79], the doping profiles in Figure 4.21 are obtain for an effective thermal budget of 1min at 1000C, 2.5min at 1000C, and 5min at 1000C. Using the process simulation results in a mixed circuit and device simulation, one can determine the minimum thermal cycles so as not to impact delay time significantly. Like the uniform doping example, the delay can drop significantly at the 1 min thermal budget since very few dopant atoms diffuse to the poly-silicon and oxide interface within that budget.

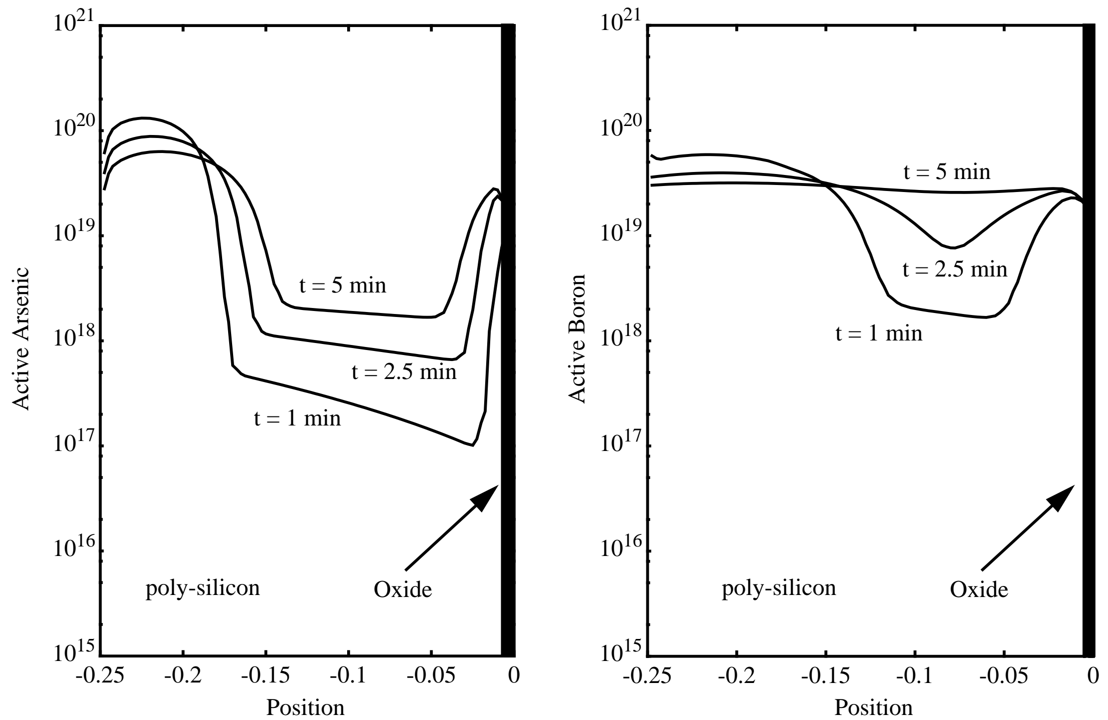


Figure 4.21: Simulated poly-silicon gate doping profiles for the (a) NMOS and (b) PMOS FET.

4.5 Conclusions

This chapter describes some simple optimizations for device structures, layout, and process based upon using complex boundary conditions with mixed circuit and device simulation. An LED structure required special boundary conditions to obtain convergence on a structure with a floating semiconductor layer and is optimized for optical performance. Generic linear circuit boundary conditions are required in order to obtain an accurate representation of the impact of the layout which can then be optimized for RF performance. A process limitation results in some complex physical effects that are incorporated into a circuit simulation to provide a way to optimize the design of the process flow. Each design required mixed circuit and device simulation or advance boundary conditions in order to obtain the most robust solution.

Chapter 5: Modeling and Simulation of RF MOS Transistors

5.1 Motivation

The critical components of wireless communication system designs are the RF transistors used for power amplification in wireless communication. Most systems are designed around the performance of a handful of large transistors (*i.e.* extremely wide gates) used to send and receive signals. Hence, design cycle time focuses on the ability to develop and improve these critical individual RF devices.

Numerical device simulation provides an ideal environment to make design decisions and trade-offs before committing to manufacturing which is more expensive in terms of cost and time. PISCES with enhancements for RF simulations (generic linear circuit boundary conditions and harmonic balance simulation) provides a powerful tool for such analysis. This chapter considers the design of an RF transistor used as a power amplifier in a hand held cellular phone. It begins with a description of the device with its physical attributes, shows the development of a model to represent the device, and concludes with an RF analysis and optimization of performance.

5.2 Device Structure and Operation

LDMOS (laterally diffused metal oxide semiconductor) devices offer significant performance enhancements for RF applications. The device was first characterized for RF applications in 1972 by Sigg [80]. The double diffused design at that time offered advantages because it allowed for channel lengths of 1μm while the dominant technology

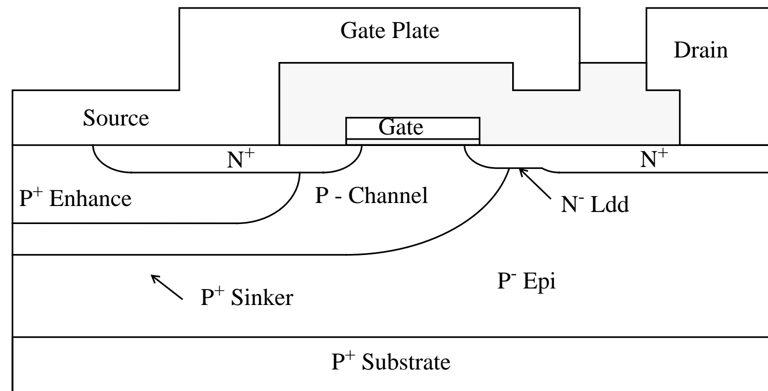


Figure 5.1: Typical cross-section of an LDMOS device.

was at 5µm due to lithography. Structures today use both the sub-micron lithography (0.6µm - 0.9µm) and a diffused channel as shown Figure 5.1 for a 0.8µm technology at Motorola [81] [82].

In spite of the advances since 1972, this device still offers significant improvements over other devices, even an equivalent uniformly doped channel MOSFET. The device provides excellent linearity performance compared to other technologies and slightly higher gain for its cost. Linearity is important for the currently preferred modulation schemes: CDMA in north America and GSM in Europe and Asia. The laterally diffused channel increases the speed of the device while at the same time maintaining a high breakdown voltage.

The laterally diffused graded channel enhances RF performance, prevents punch-through, and increases the transconductance. These improvements result from increasing the electric field in the channel region such that the electrons reach velocity saturation. Figure 5.2 shows a PISCES plot of the longitudal electric field in the channel of a LDMOS device and a standard MOSFET. The device structures are the same except in the channel region where device (a) has a graded channel and device (b) has uniform doping. The integrals of the net doping profiles along the interfaces are approximately equal. The devices are biased in inversion just above threshold with 6V on the drain. The vertical lines indicate the edge of the channel and the minimum electric field to obtain velocity saturation. Electrons reach velocity saturation throughout the entire channel region for the graded channel device, but not in the uniformly doped channel.

With carriers reaching velocity saturation, the change in the drain current versus the gate voltage becomes linear rather than quadratic. Hence, the transconductance is relatively flat over a large range of input values leading to good linearity performance.

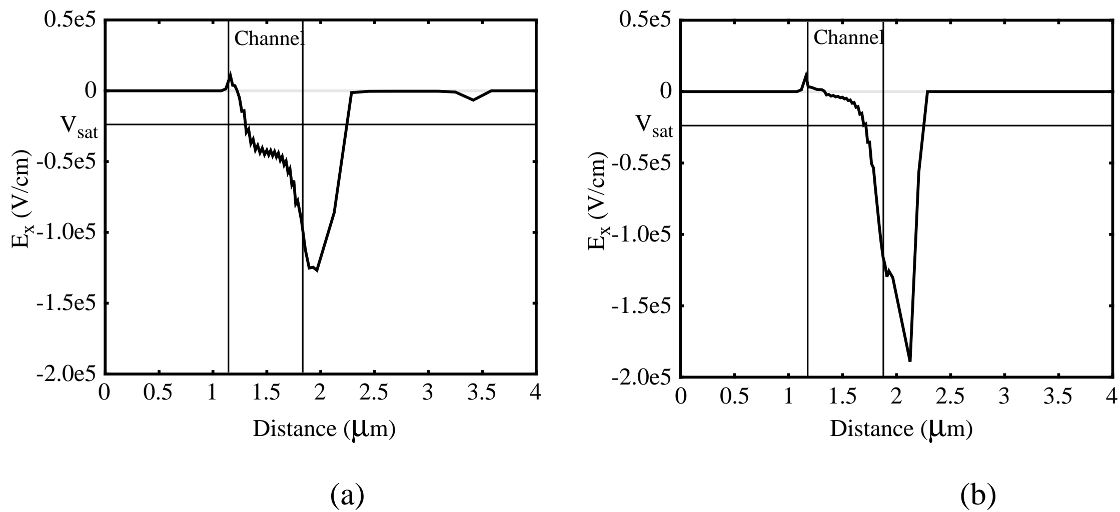


Figure 5.2: Comparison of the longitudal electric field in the channel of (a) an LDMOS device versus (b) a standard MOS device. The vertical lines indicate the edges of the channel region and the horizontal line is the approximate electric field (2×10^4 V/cm) when electrons reach velocity saturation.

Other enhancements to the device include a P^+ sinker to connect the source and substrate together and to eliminate extra surface bond wires leaving only the gate and drain with surface bond wires. The elimination of bond wires on the source leads to improved RF performance in a power amplifier configuration because of the reduced source inductance.

A metal field plate (*i.e.* Faraday shield) reduces the electric fields at the edge of the gate thereby increasing the breakdown voltage along with hot carrier generation and C_{dg}; however, there is a slight increase in C_{gs}. In a power amplifier application (common source), the input matching network can compensate for the additional C_{gs} while RF performance is improved by reducing the Miller capacitance (C_{dg}) between the input (gate) and the output (drain). The reduction in the cut-off frequency due to the increased input capacitance is not as critical as long as it is 5 to 10 times the operating frequency of the power amplifier.

Other important characteristics include an extended N^- LDD to decrease the electric field at the drain end of the channel and to optimize R_{ds}(on), BV_{dss}, and C_{dg}. As a result, the device is capable of handling the high voltages in RF power applications (45V for the device analyzed).

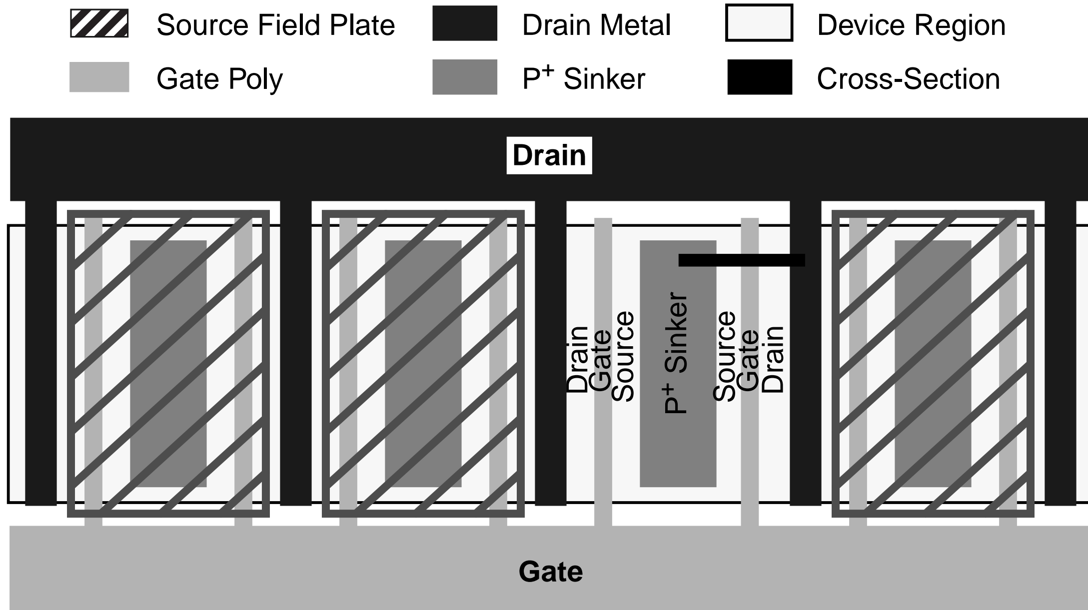


Figure 5.3: Top view of a typical layout for long gate LDMOS device.

The device must be capable of supplying large amounts of currents which implies a wide gate. To accommodate the gate, the device is created with interdigitated fingers for the drain and gate as shown in Figure 5.3 Trade-offs in the layout are made in order to optimize the RF feed network, to reduce parasitic capacitances (especially C_{dg}), and to minimize any resistances (especially R_{gate}).

5.3 Device Model and Parasitic Component Analysis

To use RF device simulation requires modeling and calibration of the intrinsic device and parasitic components. The calibration has three goals: (1) develop a model for the device that minimizes computational time while maintaining accuracy; (2) gain understanding of the device physics affecting the performance; (3) obtain better knowledge of the parasitic components and their effect on performance. Figure 5.4 shows the model used to represent the LDMOS device and to meet these goals.

Portions of the device have been eliminated from the physical device structure and replaced with lumped circuit components in simulation. As a result, the simulation time decreases because the number of mesh nodes is reduced, resulting in a smaller system of equations.

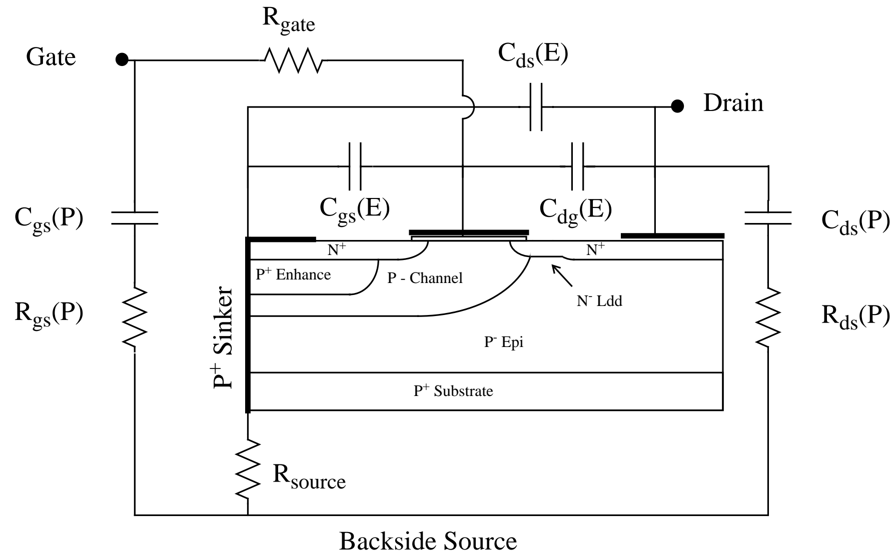


Figure 5.4: Model for an LDMOS device with parasitic components.

The criterion for replacing a section of the device is based upon the impact of physics surrounding that region. The major regions for model simplification in the LDMOS device are the P^+ sinker/backside contact and the surface contact structure. The P^+ sinker is replaced by an electrode along the source edge of the device to act as a low resistance path connecting the source and substrate. Rather than simulating the full substrate to the backside contact (~230 μm), a resistance is placed on this contact to represent the current path to the backside.

In addition to the backside contacts, the surface contacts are replaced with circuit components. Figure 5.4 shows two contributions: one from the actual device electrode vias (E) and the other from the surface pads (P). The electrode structure contributes capacitances while the pad structure adds a capacitor and resistor in series. The resistance under the pad adds to the impedance of the circuit branch and becomes important at higher frequencies as the capacitors become low impedance paths.

5.3.1 Intrinsic Device Structure

The modeling of the intrinsic device focuses on selecting the suite of models to represents the physics in the device. For the LDMOS structure the following basic models are required: concentration dependent Shockly-Read-Hall recombination, Auger recombination, and band gap narrowing. The recombination processes model the regions

that are heavily doped and the areas that can have a high concentration of carriers (*i.e.* the channel region). The band gap narrowing model addresses the energy bands in the heavily doped N^+ source and P^+ substrate [83].

The mobility models are the single most important parameters that affect the simulation accuracy. The Lombardi model is selected because it provides excellent modeling of the physical effects in the inversion layer of MOSFET's [84]. It is given by

$$\frac{1}{\mu} = \frac{1}{\mu_b} + \frac{1}{\mu_{ac}} + \frac{1}{\mu_{sr}} . \qquad (5.1)$$

It contains parameters for degradation due to surface acoustical phonon scattering:

$$\mu_{ac} = \frac{B}{E_\perp} + \frac{\alpha N^\beta}{T_L E_\perp^{1/3}} \qquad (5.2)$$

and surface roughness scattering:

$$\mu_{sr} = \frac{\delta}{E_\perp^2} \qquad (5.3)$$

as well as a bulk mobility value μ_b. Surface phonon scattering and surface roughness scattering are dependent upon the transverse electric field $E_\perp$, doping concentration N, lattice temperature T_L, and material parameters (α , β , B , δ). The transverse field component is important because of the high gate biases that occur in RF power applications.

In addition to the transverse fields, the longitudal field (*i.e.* parallel to current flow in the channel $E_{||}$) leads to a reduction in the mobility of carriers given by

$$\mu = \frac{\mu_0}{\left[1 + \left(\frac{\mu_0 E_{||}}{v_{sat}}\right)^\beta\right]^{1/\beta}} . \qquad (5.4)$$

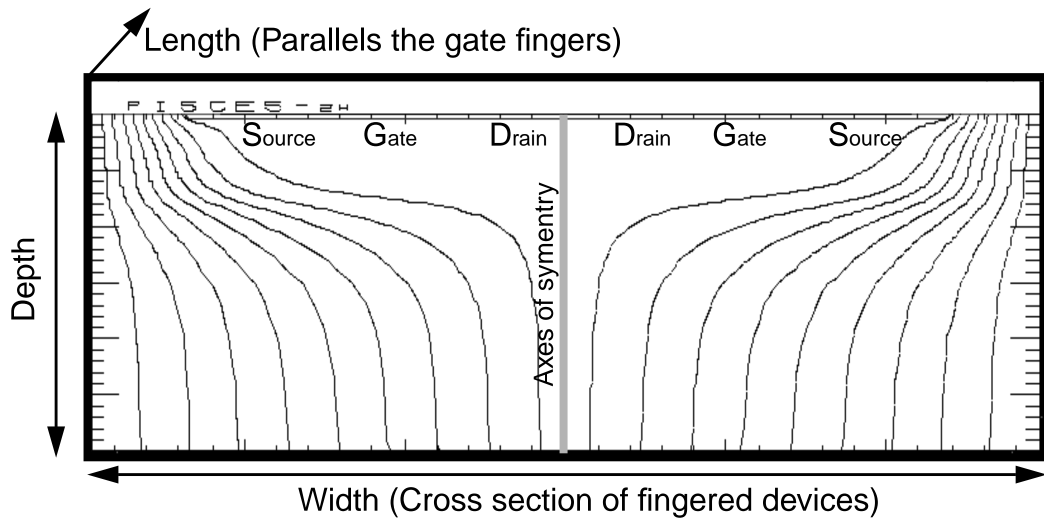


Figure 5.5: Current flow lines through the P^+ sinker and the substrate of an LDMOS device with a backside contact.

The principal component comes from saturation of the carrier velocities (v_{sat}) flowing from the source to drain as well as a material dependent factor β [85]. As demonstrated in the previous section, the carriers experience high longitudal electric fields throughout the channel because of the graded doping profile.

5.3.2 Layout Parasitic Components

Based on Figure 5.4, there are three groups of parasitic components that need to be determined: gate resistance, source resistance, and interconnect capacitance. The gate resistance is determined from the poly silicon silicide formations and is process dependent [86]. Therefore, experimental values for the sheet resistance or process development data must be used for the value of this resistance. In a latter section, sensitivity analysis is used to determine the impact of this parameter and hence, can be used to make process changes for improved RF performance.

The source resistance has components from the distributed contact resistance, sinker resistance, and substrate resistance. The contact resistance is insignificant compared to the other components for this device structure and thus will not be considered. In other structures this resistance could potentially play an important role and therefore, it would need to be characterized. A PISCES simulation of the sinker structure leads to the current flow lines shown in Figure 5.5. The figure contains a cross section of two adjacent devices due to the layout (Figure 5.3). The devices themselves are not included in the simulation

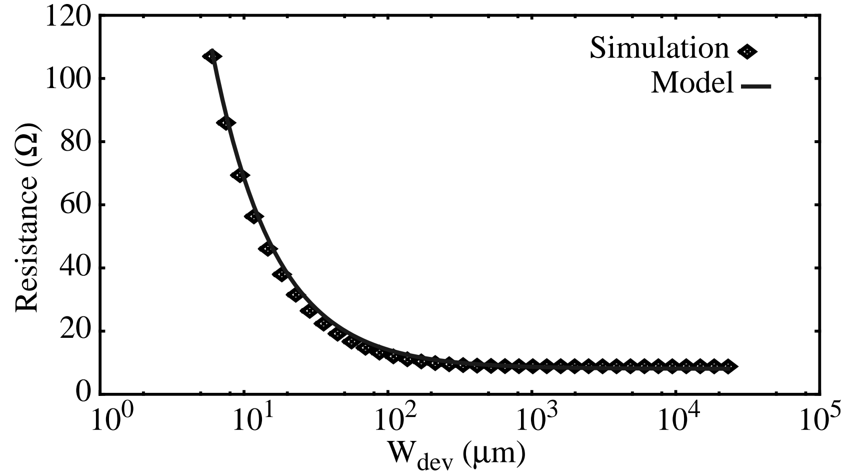


Figure 5.6: Source resistance as a function of device width.

since they do not directly affect the results. Instead, the source doping is extended all the way across both devices. The spreading of the current is limited by adjacent devices and has a significant effect on the actual resistance.

Figure 5.6 shows the resistance of one entire sinker as a function of the width of the device. The points represent simulation data and the line is a plot of

$$R(W_{dev}) = \frac{1}{L_{snk}}\left(\frac{D_{snk}}{W_{snk}} + \frac{D_{sub}}{W_{max}} + \frac{D_{sub}}{W_{dev}}\right)\frac{1}{\sigma} \tag{5.5}$$

where L_{snk} is the length of the sinker along the surface, W_{snk} is the width of the sinker along the surface, D_{snk} is the depth of the sinker (contacts the substrate through the p^- epi layer), D_{sub} is the thickness of the substrate, W_{max} is the maximum spreading of current in the substrate, W_{dev} is the width of one device (*i.e.* abscissa) and σ is the substrate conductances which is considered valid for the sinker (If the conductance is significantly different then the sinker and substrate would have to be divided by their respective individual conductance values).

Measuring one sinker as an independent unit, yields a resistance of 8Ω . The simulation model predicts a slightly higher resistance of 8.5Ω. When device confinement is included in the evaluation of the source resistance, one device sees a source resistance 46Ω . Thus,

as the size of the devices are scaled, the source resistance must be reduced appropriately so as not to degrade performance due to an increase of this parasitic component.

Important device parasitic components come from the metal contact capacitances and inductances. Using the Poisson solver in PISCES, small signal analysis can be performed on the interconnect structure to determine the values for the capacitances. The interconnect structure is created using the standard PISCES mesh generation tools which only allow very coarse representations and consequently some inaccuracies are introduced. An improved approach involves using an etching and deposition simulator to create a better replica of the structure [87]. In addition, a three dimensional interconnect simulator can be used to analyze the structure in even greater detail [88]. Fortunately, these parasitics only consist of about 15% of the total capacitances and slight inaccuracies will not impact the model accuracy. In addition, these capacitor values can be used for tuning purposes in the C-V measurements, thus providing another degree of freedom.

The inductances of the interconnect are small and at the frequencies used in this simulation, they are insignificant ($f < 1$Ghz). On the other hand, for higher frequency devices ($f > 1$Ghz), these inductances can play an important role and must be incorporated into the model. The inductances come from the length of wire along the gate and drain fingers. In addition, the metal of the Faraday shield can add a small amount of inductance to the source. Of much more importance is the inductance from the package.

5.3.3 Contact Pads

Another important parasitic component is generated by the bond wire pads. Figure 5.7 shows a top view and a side view of the pads. In the top view, the rectangular blocks represents the interdigitated devices and the unshaded labeled blocks represent the pads. From the side view, the pads are formed on thick field oxide over a high resistivity epi on top of the low resistance substrate. There are three contributions for parasitic components to the pad from the field oxide capacitance, epi resistance, and substrate resistance.

The lightly doped thin epi has a resistance much greater than that of the thick substrate. This high resistance adds to the series impedance of the capacitance to prevent the leakage of RF power especially at high frequencies when the capacitor becomes a low impedance pathway. The model for the resistance excludes current spreading because very little spreading occurs in the epi layer.

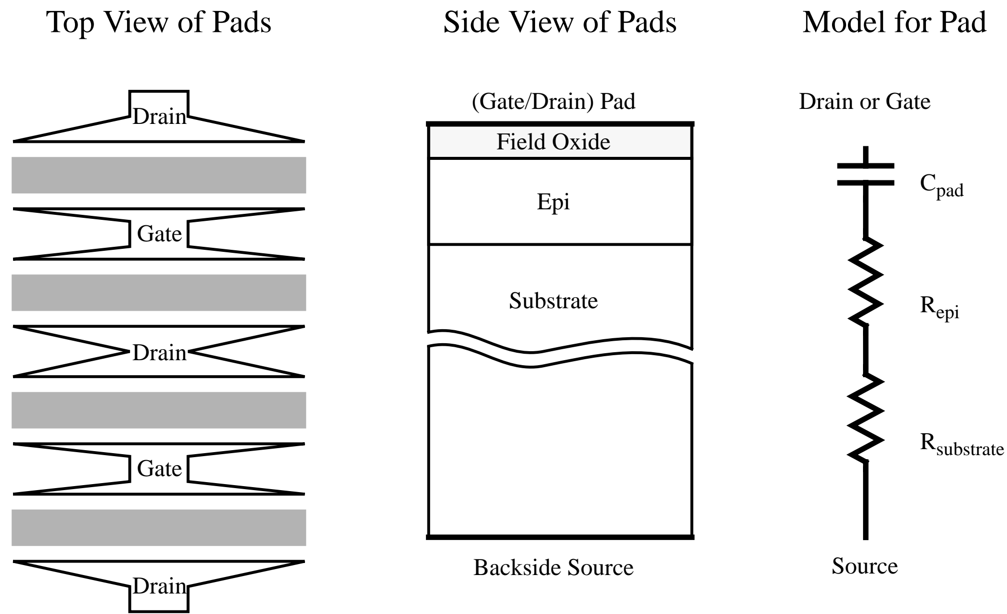


Figure 5.7: Top view and side view of pad structures along with a representative model.

At higher frequencies (> 10GHz) the substrate capacitance can become significant and must be incorporated into the pad model. The substrate capacitance results when the carriers in the substrate can not be modulated as a conduction current as effectively as the displacement current present from the pad to the back-side ground of the device. This capacitance can help high frequency performance because it effectively reduces the total impedance since C_{pad} is in series with $C_{substrate}$. For the relatively low frequencies used with this device (< ~6Ghz for the 7th harmonic), this additional capacitance is not needed.

5.3.4 Packaging

The packaging plays a very important role in the performance of the device. Bond wires connect the pads on the device and the pins of the package and can add a significant amount of inductance. Good package design takes advantage of the inductance to provide some internal matching. The simulation of the structure requires that the packaging be considered whether the additional parasitic components are intentional or not. The less well understood source inductance originates from the back side of the package and its experimentally measured value varies somewhere between 0.02nH and 0.15nH. Hence, this inductance becomes a prime candidate for sensitivity analysis.

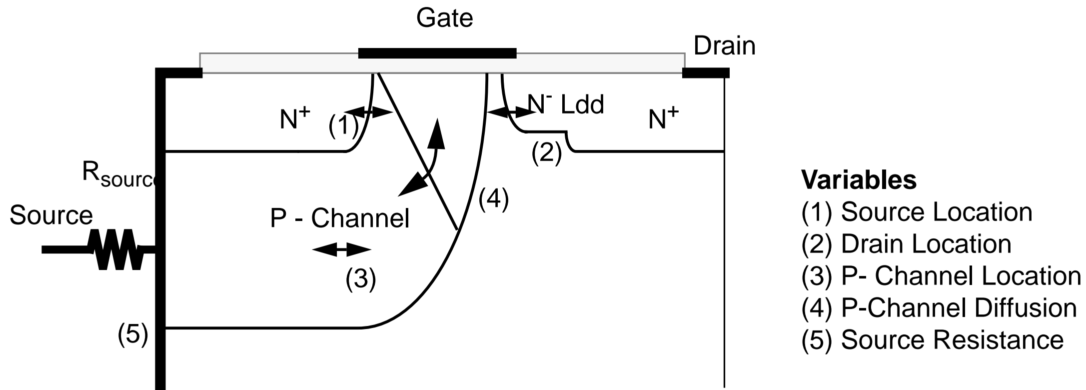


Figure 5.8: Design of experiments to determine optimum 2D doping profile.

5.4 DC and AC Analysis

Having defined the contributions of the parasitics, the simulated responses of the LDMOS model is compared to measured data. Current voltage relationships are used for tuning the dc aspects of the model and for understanding the operating sensitivities. Small signal analysis is used to determine the capacitance-voltage relationships which can give detailed information about junctions and doping profiles.

5.4.1 I-V Characteristics

One of the key modeling regions for the I-V characteristics is the graded channel. Only one dimensional doping profiles are simulated and calibrated with measured process data; however, the profiles must be expanded into two dimensions in order to generate the channel region of the device. The methodology chosen for this calibration is based upon a simulation design of experiments (DOE) [89].

Key PISCES parameters describing two dimensional spreading of doping profiles are varied based upon manufacturing tolerances as shown in Figure 5.8. The parameters are the inner edge of the source and drain relative to the gate poly (parameters 1 and 2 in Figure 5.8); the edge of the p-channel implant relative to the edge of the gate (parameter 3 in Figure 5.8); the curvature of the p-channel region under the gate (parameter 4 in Figure 5.8); and the source resistance (parameter 5 in Figure 5.8). The fitting of the source resistance allows for a comparison of the predicted source resistance with that from the model for the source resistance.

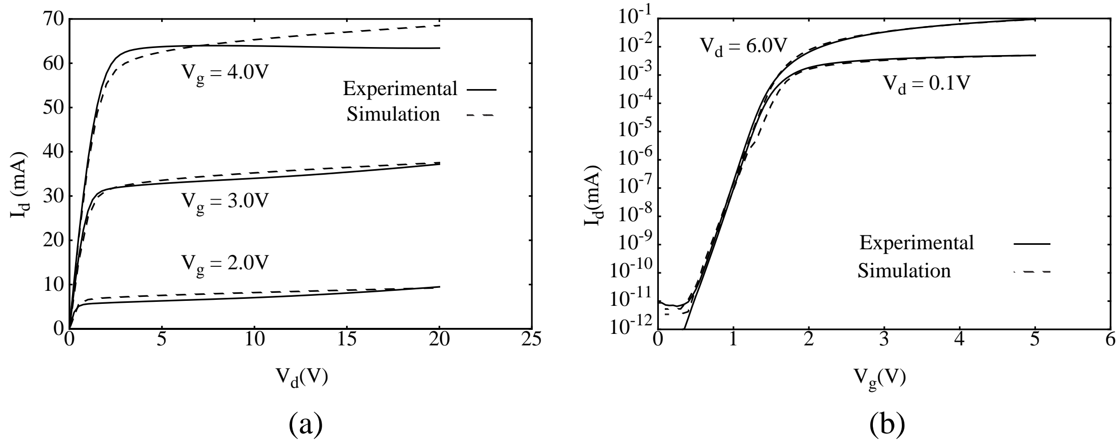


Figure 5.9: Comparison of simulated and measured I-V characteristics for the modeled LDMOS device.

Running a factorial experiment with the five variables limited by the processing characteristics, a statistical model is developed. Three independent statistical models relate the threshold voltage, transconductance, and sub-threshold slope (*i.e.* all three being responses) to the variables. The simulated parameters are determined (*i.e.* measured virtually) using the same methods as in the lab. Upon generating statistical models for each of the responses, they are used to determine the optimum set of parameters. The results of that optimization is shown in Figure 5.9 where simulated and measured I-V characteristics are presented.

In addition to calibrating the intrinsic LDMOS model, the statistical models may be used to do sensitivity analysis. Sensitivity analysis tells a design engineer the degree to which a parameter affects the device performance. For this LDMOS device the two variables most strongly affecting the responses are as follows (most significant variable is listed first):

- The measured G_m at high current is affected strongly by source resistance and source location. The internal V_{gs} is reduced by the voltage drop across resistances on the source side of the device; therefore, any changes that cause the resistance to change will affect the transconductance.

- The measured V_t is affected strongly by channel location and source location. Threshold is determined by the inversion of the channel nearest the source side of the device; thus, any changes on the source side of the device affects the threshold voltage.

- The measured sub-threshold slope is affected strongly by source location and channel location. The slope is determined by the channel length and punch through characteristics; thus, any change in doping concentration near the source will affect the sub-threshold slope.

In addition, the drain location has no effect on dc characteristics, but later analysis shows there is a significant effect on ac characteristics.

In this work, RF power devices are optimized to transmit large signals where signal spikes can generate heat. With increasing power densities, self-heating becomes important and thus the model will require that the lattice thermal diffusion equation be solved simultaneously with the semiconductor equations [90]. In addition, breakdown will limit the high power operation thus requiring the modeling of impact ionization for devices operated in that region [91] [92]. Both of these effects require complex computations [93] and require improvements in harmonic balance solution techniques [41].

Even though, the device can exhibit a negative differential resistance as is apparent in the measured drain characteristics shown in Figure 5.9a, the application for which this device is targeted does not require its operation in the region of high currents. Thus the exclusion of self-heating effects should not affect the results at low power levels with an error less than 10% at high power levels.

5.4.2 C-V Characteristics

Having calibrated and characterized the dc response, small signal analysis can provide detailed understanding of the device operation. The standard C-V characteristics are simulated under the same conditions as the measurement. A dc value is swept while an ac perturbation is applied to one input of the LDMOS model. The ac current is obtained and the capacitance is computed from the imaginary part of the simulated admittance (Y-parameters).

The capacitances can be divided into three categories: output capacitance (C_{ds}), the Miller capacitance (C_{dg}), and the input capacitance (C_{gs}). The output and Miller capacitance are essentially junction capacitances as shown in Figure 5.10. The plots show the small signal characteristics with and without parasitic components which comprise about 15% of the small signal response. The output capacitance is dominated by the depletion capacitance formed from the junction of the N^- LDD/ N^+ drain and P^- epi. Because the substrate is

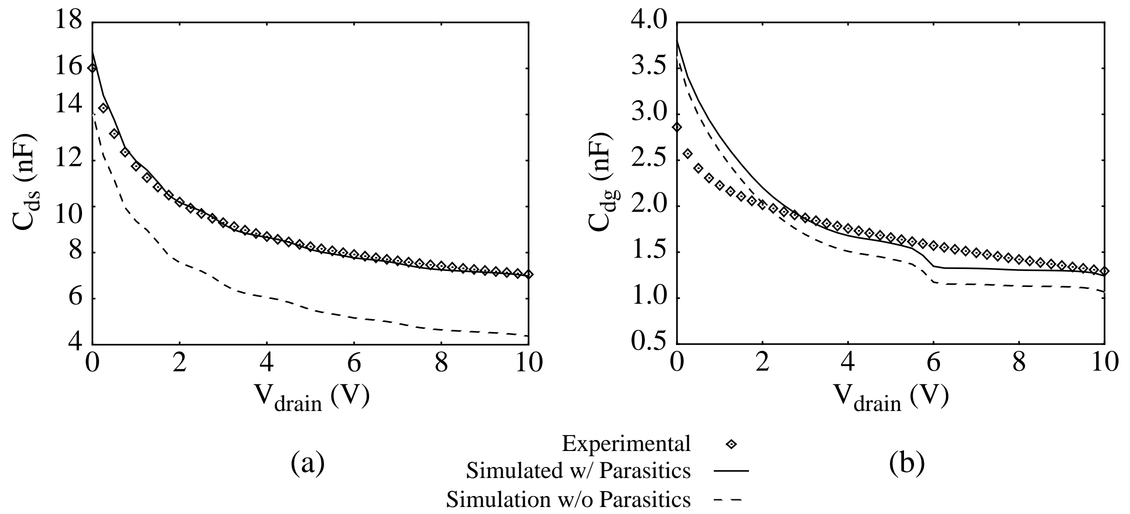


Figure 5.10: Measured and simulated capacitance for C_{dg} and C_{ds}.

heavily doped, the output capacitance tends to flatten out once the depletion layer reaches the P^+ substrate. The Miller capacitance is dominated by the overlap of the N^- LDD and gate. With increasing drain bias, the N^- LDD depletes until the N^+ drain is reached at which point the capacitance will tend to flatten out.

Since specific parasitic components affect each curve independently, their values are adjusted from their initially computed value. For example the impedance of the pads can be computed from a one dimensional PISCES simulation of the structure. Tuning tells a design engineer the variations that can be expected in the structures and thus provide valuable information for next generation technology development.

The measurement of the input capacitance can provide some interesting details about the channel region. Figure 5.11 shows three forms of measurements for the input capacitance with the drain contact in different states. The impact of such a measurement is that individual components of the channel capacitance can be isolated since the doping and consequently the flat band voltage change across the channel. As a result, the channel can be in different states depending upon the gate bias as shown in Figure 5.12.

Starting with a negative V_{gs} in Region 1, the channel is in accumulation. With increasing V_{gs}, the drain side of the device enters depletion; the source side is in accumulation; and the center of the channel is at flat-band for tens of millivolts (Region 2). As the gate

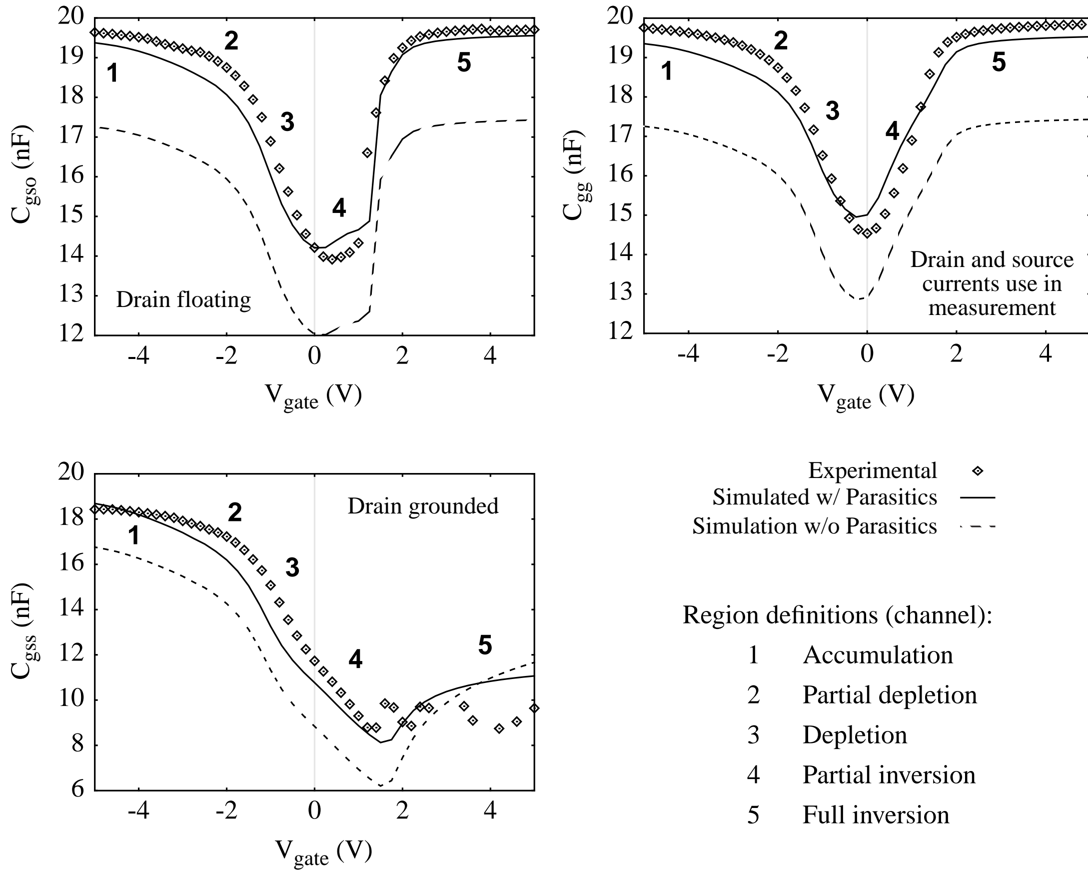


Figure 5.11: Measured and simulated values for the input capacitance (C_{gs}) with the drain floating, grounded, and included in the measurement. The numbers correspond to different states of the channel.

voltage continues to increase the depletion layer reduces the capacitance until around V_{gs}=0V (Region 3). At this point the drain side of the device enters inversion (Region 4).

When the drain side of the device enters inversion, the measurement configuration becomes important. When measuring C_{gso} (open circuit drain) there is no current path through the drain side inversion layer and the capacitances stays low. When measuring C_{gg} (both drain and source current) there is a drain inversion by-pass capacitance in parallel with the source side depletion capacitance. Since the inversion capacitance is larger and the inversion layer moves across the channel from drain side to source side; the net capacitance measurement increases with voltage as the inversion layer moves across the channel. When measuring C_{gss} (short drain to ground), the by-pass capacitance is connected to ground. As a result, the measured capacitance continues to decrease as the by-pass capacitance becomes larger in area as the inversion layer moves across the

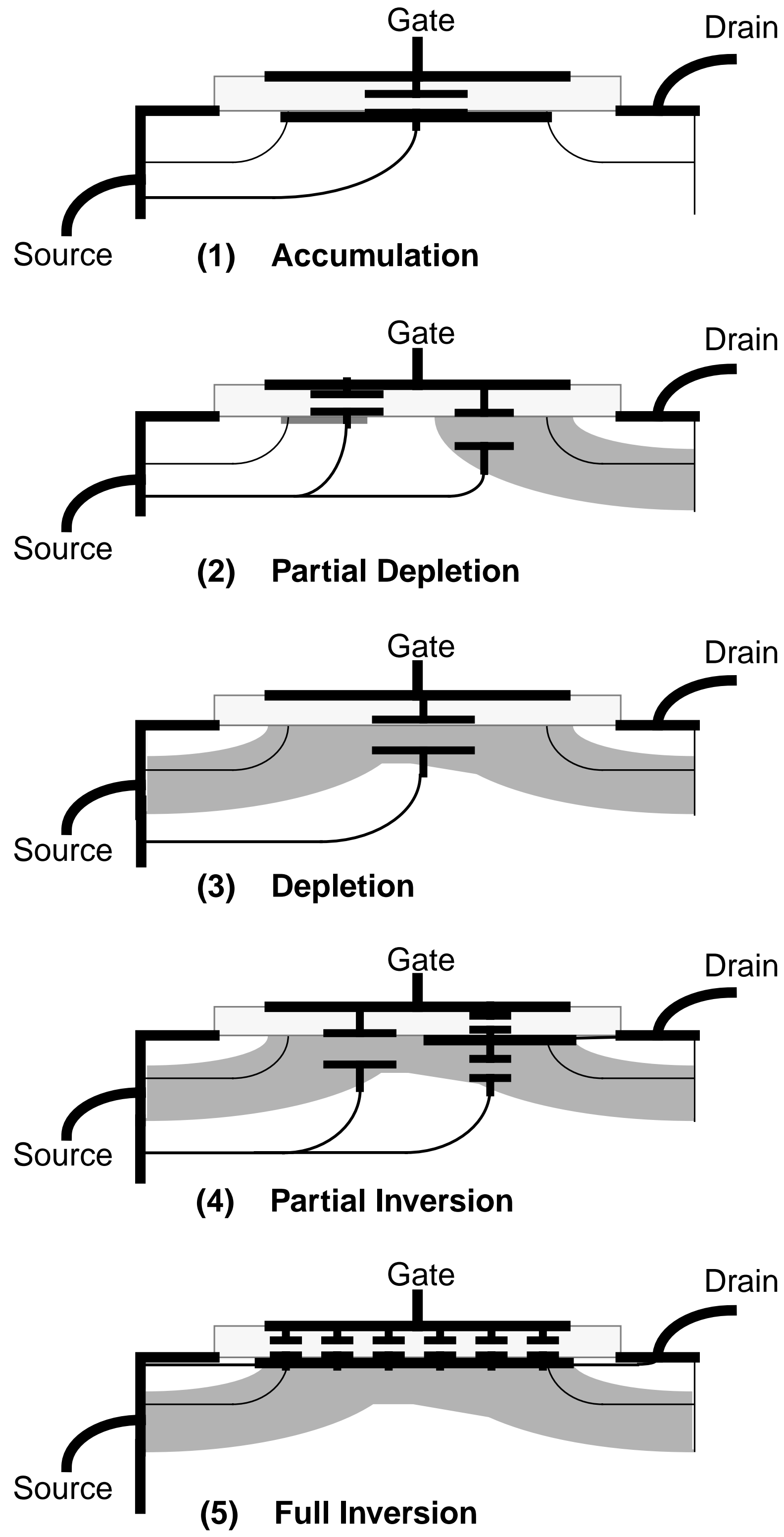


Figure 5.12: Plots of accumulation, depletion, and inversion as each tracks across the channel from the low doped drain side to a high doped source side of the channel region.

channel region. When the source side of the channel inverts, C_{gso} immediately switches to the inversion capacitance whereas the C_{gg} had slowly approach that value. The C_{gss} measurement becomes indeterminate since the inverted channel connects the source (measurement side) to ground.

Isolating the different contributions during the input capacitance measurements provides information on the channel doping. For example, Pieracci used this property of the graded channel device in order to estimate doping profile based upon the incremental threshold voltage [94].

Some insights into the Miller capacitance may be ascertained from the capacitance plots. The device is biased at six volts on the drain. As a result, the Miller capacitance (C_{dg}) is relatively low at 1.8nF. Unfortunately, as the power level increases, the drain voltage can swing low while the gate voltage is high resulting in a C_{dg} of about 2.4nF. Although a capacitance can not really be defined for high power levels, it is apparent that the displacement current coupling between the gate and drain can become more severe resulting in some additional gain degradation. Hence, the design of the LDD region becomes critical for device performance and novel process design is required [7].

5.4.3 S-Parameters

S-parameters provide an evaluation of the high frequency accuracy of the model. Because of the difficulty in measuring the full multi-cell device, a single cell of the structure is measured and compared to a single cell model. The device is biased under the typical conditions for its targeted application. The measured S-parameters match the simulated S-parameters up to about 6GHz which encompasses up to the 7th order harmonic for the 850MHz application. Figure 5.13 and Figure 5.14 show that the parasitic components contribute significantly to the impedances looking into the device and lead to a degradation in the small signal gain of the structure. Hence, layout optimization and reduction of parasitic components is a key design factor in developing next generation technology as demonstrated in Chapter 4 for a MESFET.

5.5 RF Characterization

The model development yields a deeper understanding of the device physics. The previous sections described how the intrinsic device simulation and parasitic analysis can be used to better understand performance trade-off. In this section, the model is used to evaluate the

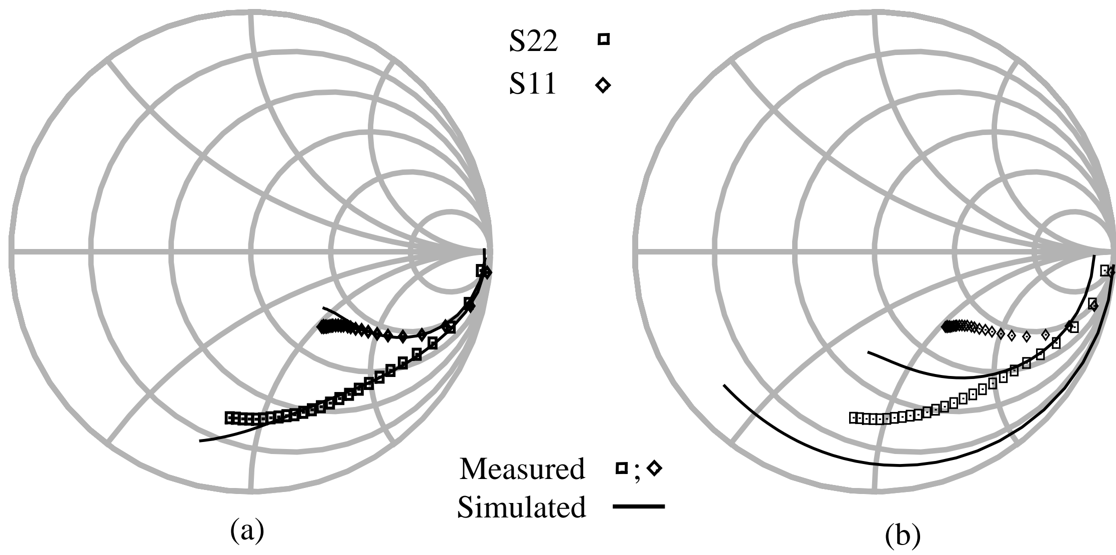


Figure 5.13: LDMOS impedances looking into the device (a) with and (b) without parasitic components.

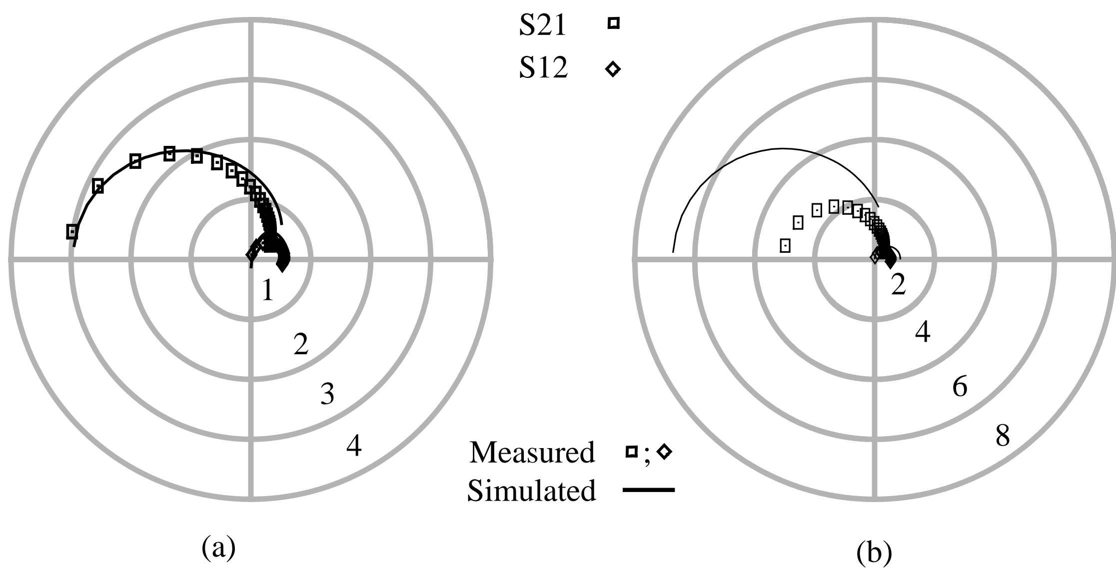


Figure 5.14: LDMOS gain (a) with and (b) without parasitic components.

RF performance for the power amplifier configuration shown in Figure 5.15. The RF performance is evaluated by the transducer gain (G_t), power added efficiency (PAE), and inter-modulation distortion (IMD).

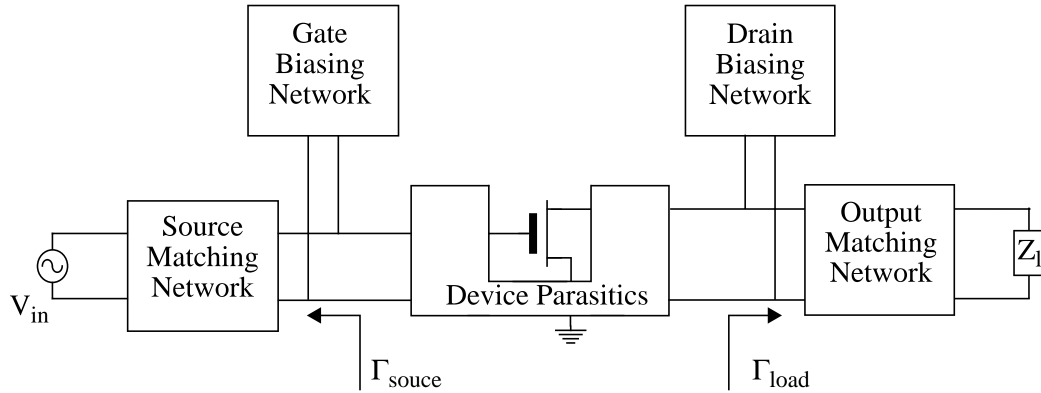


Figure 5.15: Configuration of a power amplifier. The biasing networks set the load lines and the matching networks provide for transfer of RF power.

5.5.1 Large Signal Distortion

For single tone harmonic distortion analysis, a large signal sinusoid is applied at the input to the power device (V_{in}) and the steady state large signal response is computed numerically. The transducer gain and power added efficiency characterize the performance of the device. The gain is given by

$$G = 10\log\left(\frac{P_{avs}}{P_{load}}\right) \tag{5.6}$$

which takes the ratio of the output power delivered to the load and the power available from the source (P_{avs}). The power added efficiency gives the percentage of power that goes into amplification of the signal and is given by

$$\mathrm{PAE} = \left(\frac{P_{load} - P_{avs}}{I_{avg} V_{dc}}\right) \times 100\% \quad . \tag{5.7}$$

Figure 5.16 compares the simulated gain and efficiency with experimental data for an input frequency of 850Mhz using matching networks optimized for the device application (refer to Section 5.5.4). The gain and efficiency are plotted versus the output power (P_{load}) since the power amplifier has to meet a minimum output power specification for the cell phone industry standard. The gain at low power is equivalent to the small signal gain. At an input power of -30dbm, the simulated response of 15.1db agrees well with the

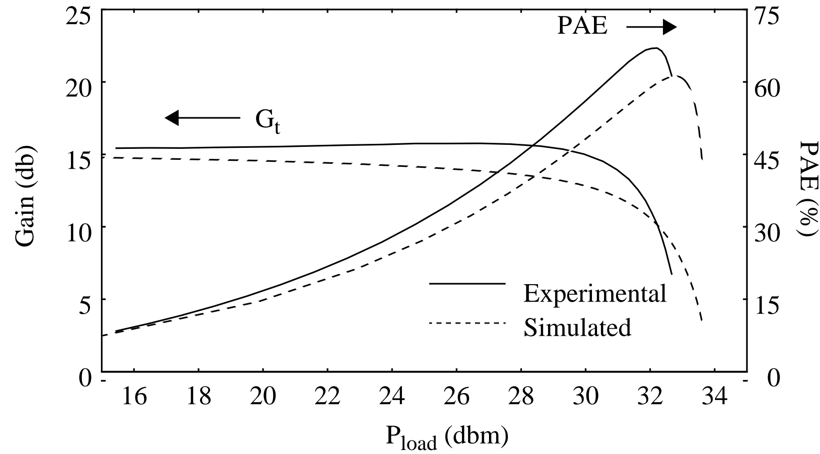


Figure 5.16: Simulated and measured RF responses for gain and efficiency of a power amplifier.

measured response of 15.4db. The gain rolls off at higher power levels because the device operates in g_m compression and the output power is limited by the saturation current. The efficiency is low for small P_{load} because the device drains more power due to Class A operation. Efficiency increases until just after the gain starts to roll off. At this point, P_{avs} approaches P_{load} resulting in very little power added to the input signal.

Although the curves differ depending on the specific region of operation, the two results show an over all good agreement consistent with several factors that can affect the results. The strongest influencing factor is that of the matching networks. The matching networks used in the simulation are those determined experimentally by a load-pull system. (Section 5.5.4 discusses simulated load-pull analysis.) In spite of a good match for the I-V and C-V characteristics this matching network is optimized for the manufactured device used in the measurement with all of its inherent process variations and not the specific device in the model. It should be noted that in actual power amplifier circuits, it is often necessary to provide a variable resistor or capacitor on the circuit board to provide for tuning of the matching networks and other resonant circuits in order to meet the manufacturing specifications.

In addition to the matching network, the gain roll-off is associated with the edge of the linear and saturation region as the device enters g_m compression (*i.e.* high drain current and high gate voltage). Under this condition, the device exhibits self-heating thus causing degradation in performance. As stated previously, the simulator neglects these self-heating effects and over predicts the performance by about 10%.

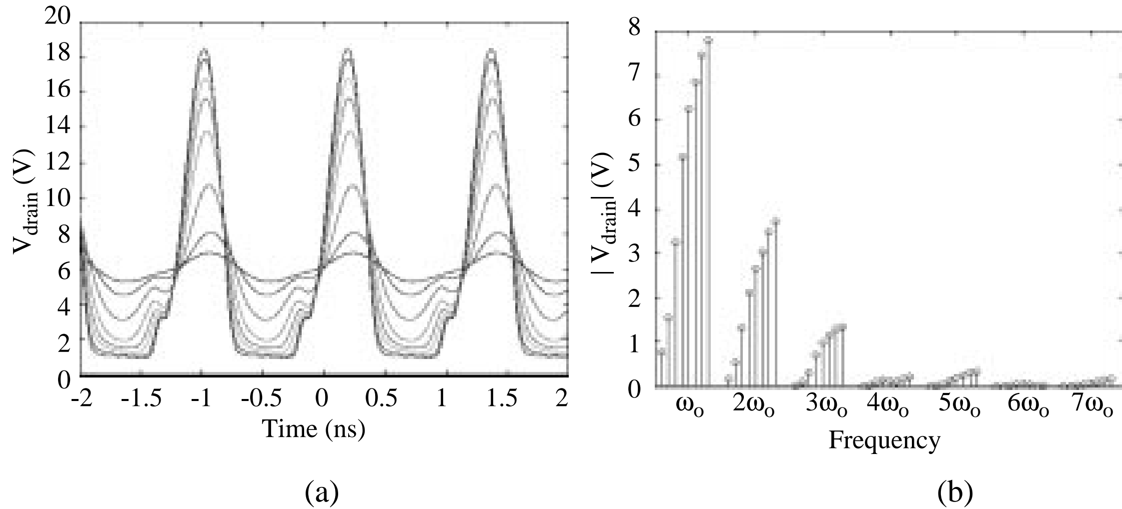


Figure 5.17: (a) Time domain signal and (b) frequency spectrum of the drain voltage in an harmonic distortion large signal sinusoidal simulation. The larger magnitude of the signal indicates increasing power along the gain curve.

Using the results of the harmonic balance device simulation, it is possible to examine the signals in the time domain as well as their associated spectra. Figure 5.17 shows the drain voltage in both the time domain and the frequency domain for increasing power. At low voltage levels the device operates in Class A. As the power level increases, the voltage cannot swing below threshold and thus enters Class AB operation. As the power level continues to increase, the drain voltage is limited by G_m compression and thus the upper swing becomes limited. The spectrum exhibits this effect as higher order frequency components become more apparent.

5.5.2 Impact of Parasitic Components

Having verified the RF performance of the physics based simulation model, it is now possible to use it to evaluate the performance for design variations. Since parasitic components are important in the device performance, this section examines those components through manufacturing variations and improvements.

Knowing which parasitics have the largest impact on the performance can lead to judicious selection of the parasitics for subsequent design optimization. A DOE is used to analyze the impact of three components at their high and low values as observed in manufacturing. R_g is selected because it is determined by the silicide processing step and

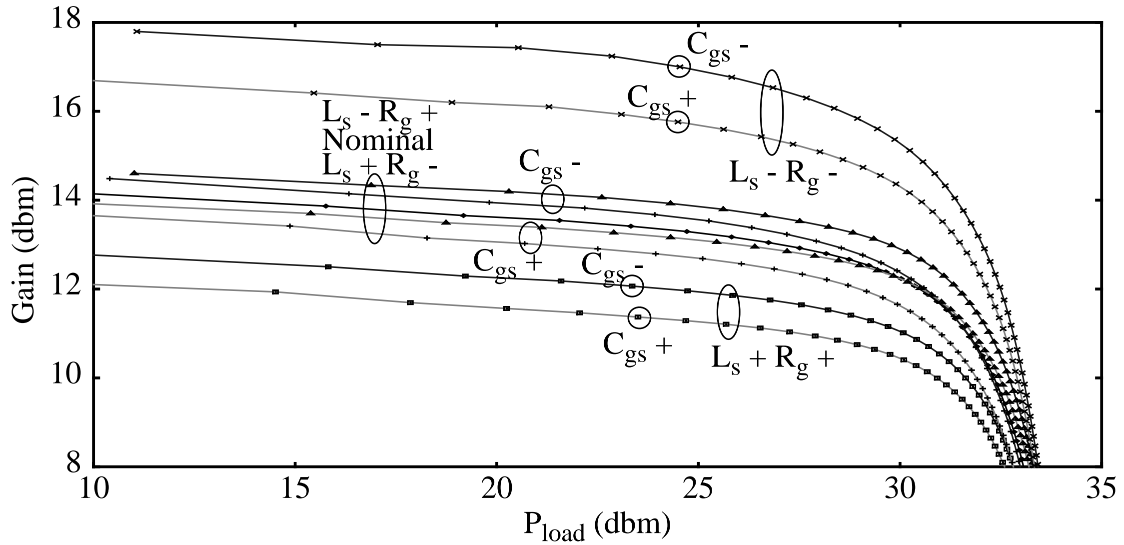


Figure 5.18: Simulated effects of parasitic components on the gain of the LDMOS device. The “+” and “-” indicate whether the variable is at it’s manufacturing high or low value.

hence, can be adjusted. C_{gs} is related to the Faraday shield of the device and the value of L_s (source inductance) is difficult to model physically due to back-side contact effects.

The effects of the parasitic components on gain and efficiency are shown graphically in Figures 5.18 and 5.19 The general effects of the parasitic components on the amplifier performance are apparent, but the important observations come from the degree to which these parasitics affect the performance. In the gain curves, both R_g and L_s have a similar impact within the limits chosen for these parameters. When both parameters are at the high level, the gain is the lowest and likewise when both are at their low levels, the gain reaches its highest value. When one increases and the other decreases, very little improvement is observed. Variations in C_{gs} has only a minor impact on the gain and “modulates” the curve about the solution established by R_g and L_s. The efficiency curves clearly show that the strongest effect can be attributed to the gate resistance. C_{gs} has the next strongest effect and L_s has the least impact on performance.

In addition to the parasitics components considered above, an RF device engineer can also modify the intrinsic device structure to address improvements as technology evolves. Sato-Iwanga has shown how using a similar modeling approach, the inter-modulation

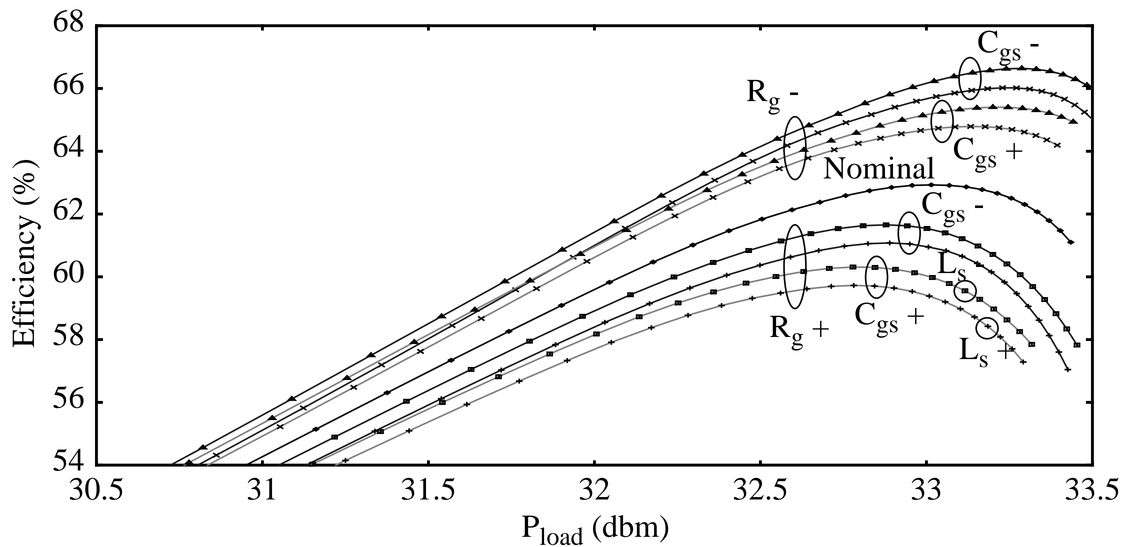


Figure 5.19: Simulated effects of parasitic components on the efficiency of the LDMOS device. The “+” and “-” indicate whether the variable is at it’s manufacturing high or low value.

components of a MESFET is studied [95]. Jang shows the importance of mixed circuit and device simulation for RF performance analysis of BJT’s [96].

5.5.3 Inter-modulation Distortion

Inter-modulation distortion (IMD) provides a way to characterize the linearity of the device by applying two closely spaced tones in the frequency domain. Harmonics are not only created at multiples of the frequencies, but also at integer sums and differences of the frequencies. This leads to the generation of an harmonic in the base band of interest and thus; it can not be filtered easily as shown graphically in Figure 5.20. The amount of IMD is characterized by computing the ratio of power in the generated harmonics to that of the fundamentals.

Using an 8th order simulation, the inter-modulation distortion for the 3rd order harmonic, 5th order harmonic, and 7th order harmonic can be computed for simulation frequencies of f_1=851MHz and f_2=849MHz. The simulated results are compared to experimental data in Figure 5.21. The 3rd order inter-modulation distortion shows very good agreement to experimental data as does the 5th order IMD. The 7th order IMD follows the correct trend, but IMD7 is very difficult to measure and the simulation was limited to one harmonic beyond it.

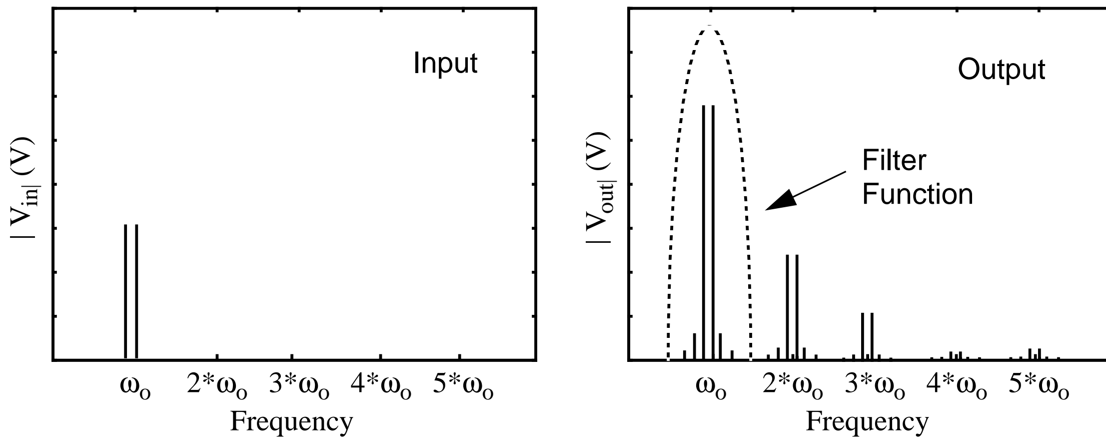


Figure 5.20: Frequency domain response of a non-linear system given an input of two tightly spaced tones.

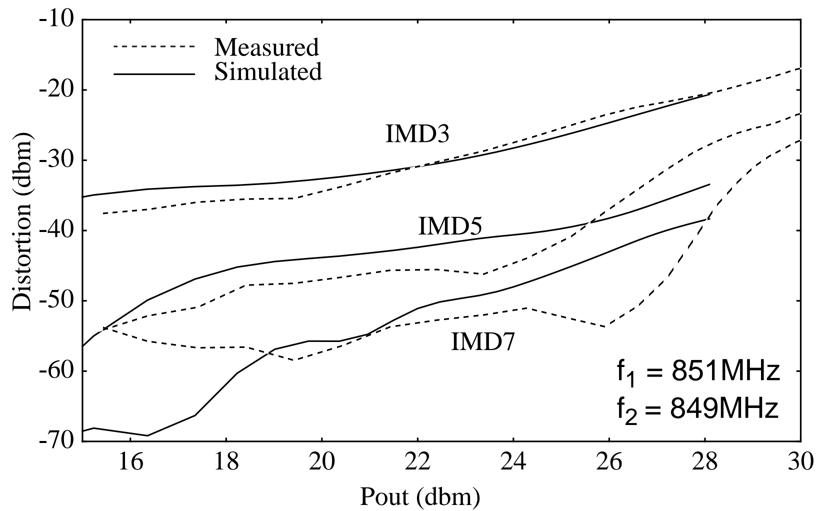


Figure 5.21: Simulated and measured inter-modulation distortion of the LDMOS power amplifier.

Linearity is important from a circuit design perspective. In typical systems, a design requires -100dbm of distortion. The intrinsic device itself is not capable of such high linearity and thus circuit techniques must be utilized to obtain the required distortion [97]. The goal of the device designer is to adjust the device such that the linearity is below a certain specification over a power range and then use a non-linear circuit to further reduce the IMD.

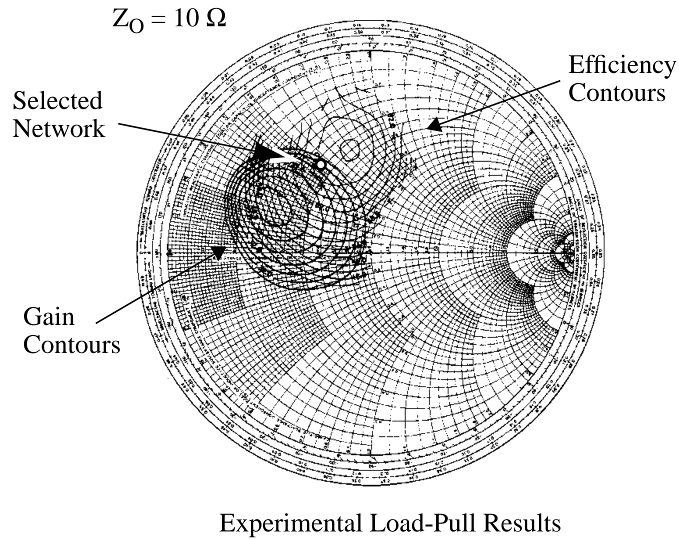
5.5.4 Matching Networks

In addition to the intrinsic device, gain and efficiency depend upon the matching networks that connect the signal source and the load to the amplifier. One important design choice is a matching network that allows for a smooth impedance transition from the input signal to the input impedance of the amplifier and from the output impedance to the load. The main design criterion is to have the networks transmit at the fundamental frequency without reflections or attenuation due to filtering while the networks filter out higher order harmonics without reflecting them back into the amplifier.

The input and output matching networks have limited interactions with each other and are thus tuned independently. The input matching network is tuned to minimize reflections back to the signal source using a best guess for the output matching network. To determine the best output matching network, the reflection coefficient looking towards the load (Γ_{load} in Figure 5.15) is varied by sweeping the load across a range of values over which the best response is expected (*i.e.* a load-pull). For each matching network, the gain and efficiency are measured for a constant input power of 20dbm and plotted over the range of the network's reflection coefficients [98].

Figure 5.22 shows experimental contour plots for gain and efficiency on a Smith chart which is used to represent the different reflection coefficients. (Note that the impedances on the Smith chart are normalized to 10Ω rather than the standard 50Ω to obtain better visibility.) The white point on the graph indicates the matching network used to generate the earlier plots of gain and efficiency. The selection criteria for this point depends upon the application for the device. The alignment of the maxima and the shape of the contours demonstrate that the simulated results shown in Figure 5.23 agree with the experimental data.

There are some important criteria for load-pull simulations. Load-pull equipment is designed to optimize the impedance for a specific reflection coefficient given that a reflection coefficient can be realized by multiple combinations of impedances. In simulation, the impedance is set to obtain a specified reflection coefficient. If this impedance is chosen poorly, reflections back into the structure or filtering may affect the performance of the device in unexpected ways. Hence, care has been taken so that the choice of components maximizes the performance of the device for a given reflection coefficient of the output matching network.



Experimental Load-Pull Results

Efficiency contours spaced at 2% with a peak at 68%
Gain contours spaced at 2dbm with a peak at 32dbm

Figure 5.22: Experimental load-pull analysis. The white point represents the matching network used in the earlier RF response.

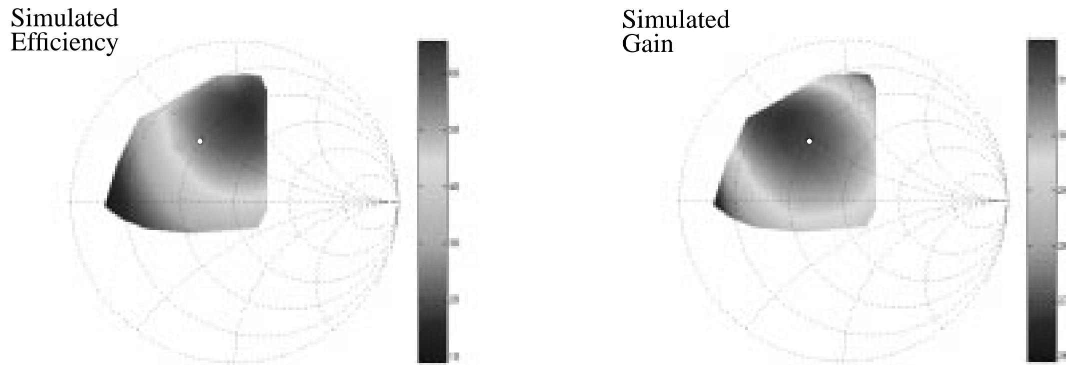


Figure 5.23: Simulated load-pull analysis. The white point represents the matching network used in the earlier RF response.

Load-pull contours allow the device engineer and circuit engineer to evaluate the maximum performance of the structure under a set bias. The simulations require significant computational time and thus, can only be used sparingly. Upon settling on a device structure though optimization of the dc parameters, ac performance, S-parameters, and gain/efficiency in a standard test fixture, load-pull may be used. Load-pull provides

information on how best to design a matching network and whether the device is fully capable of the performance required. This devices is targeted for cell phone applications and requires a minimum gain. If the efficiency at that minimum gain does not meet circuit design requirements, iterations on the device structure are required starting back at the linear ac measurements.

5.6 Conclusions

PISCES for RF simulation provides a powerful tool for the device engineer. As demonstrated in this work, it allows the engineer to investigate variations of a device and the parasitic components associated with it. Hence, it allows for rapid improvement in device performance and leads to a competitive edge in the personal communications market.

Chapter 6: Conclusions

6.1 Summary of Contributions

The goal of this work has been to develop mixed circuit and device simulation capabilities for the analysis, design, and optimization of opto-electronic, radio frequency, and high speed semiconductor devices. This dissertation provides the algorithms and methodologies for the implementation of such simulations and includes many industrial examples to demonstrate the capabilities of the different approaches.

The contributions of this work are as follows:

- Extension of mixed circuit and device simulation for use with larger and more complex problems.

- Demonstration of a parallel version of the mixed circuit and device simulation so that large problems can be solved on a network of workstations.

- Improvements to PISCES to include a generalized linear circuit through boundary condition equations.

- Integration of PISCES with an harmonic balance solver and development of the boundary condition equations for harmonic balance analysis with circuit components.

- Demonstration through examples that mixed circuit and device simulation is important and useful for the development of opto-electronic, radio frequency, and high speed devices.

- Modeling of RF LDMOS devices for use in device simulation to provide predictive analysis and optimization.

- Development of tools and methodologies to aid in RF device design by including the intrinsic physical device with external parasitics and networks.

6.2 Future Work

The future work in mixed circuit and device simulation involves a tighter integration of tools. Specifically, the following improvements will greatly enhance over all mixed signal analysis capabilities:

- **Thermal analysis**
  With higher packing densities for digital circuits and larger power densities for RF amplifiers, thermal effects become important for accurate simulation. Device simulators currently contain the ability to solve for self-heating and temperature dependencies [93]. This capability needs to be expanded to incorporate the circuit effects in two areas.

  The first area involves adjacent devices where one device has a high power density. This device acts as a heat source which can then impact the adjacent devices. In CMOS circuits, this temperature increase degrades the performance of all adjacent transistors even though they may not have high current densities.

  The self-heating temperature equations need to be included with harmonic balance simulation. The RF LDMOS device described in this work is relatively small and does not handle a significant amount of power. RF power transistors used in base station applications are designed to handle 60W in today's technology and that power level is increasing with each design cycle. For these devices, thermal effects become significant and must be included in the RF large signal analysis.

- **Optical photon simulation**
  In order to characterize the photon output of an opto-electronic device, this work used the radiative recombination profile as a reference. A better approach would be to actually solve for the optical profile. Currently, there are tools for solving the photon output of devices, but they are stand alone systems [99]. In order to fully characterize opto-electronic device performance, a complete mixed circuit and device simulation is necessary. As a result, time delays and responses of the optical output can be studied in the application for the device.

- **Multiple device harmonic balance simulation**
  The harmonic balance solver is currently configured to work with a single device and any arbitrary linear network surrounding it. This capability allows for the RF characterization of the device as it would be used in a circuit application. Future work requires the simulation of multiple devices in order to study the physical response of more complex RF circuits that consists of two or three transistors and may include non-linear matching networks.

- **RF Layout Dependencies**
  The signal paths for large RF transistors are quite complex. As devices become larger, the impact becomes more significant since it is more difficult to feed every cell and gate length exactly the same. In addition, process variation across these large devices yield slight variations in performance across the die. As a result, parts of the device may not turn on fully leading to non-uniformities in the device currents. One approach to study this effect is to break a device block into individual cells and model the layout that interconnect those cells. This approach coupled with thermal analysis can provide important layout optimization.

These non-trivial improvements in mixed circuit and device integration can lead to improved analysis and optimization of device performance and open new avenues for efficient semiconductor technology development.

Appendix A: Users Manual for Mixed-Mode Simulation

A.1 Introduction

The Stanford mixed-mode simulator provides a mechanism to incorporate numerical devices into a full circuit simulation with linear and non-linear components. The mixed-mode simulator uses the industry standard SPICE as the circuit simulator with an added model for the numerical device. This numerical model is configured modularly such that any device simulator may be added. For the purpose of this document, Stanford PISCES is used.

This chapter describes how to include a numerical model in a circuit simulation. All descriptions are based upon SPICE version 3f and hereafter referred to as SPICE only.

A.2 New SPICE Cards

Upon installing and configuring SPICE, the numerical model may be utilized in any circuit. The next two sections describe how to use the SPICE element card to specify the numerical device instance and the SPICE **.model** card for setting the parameters for the numerical device. The format for the cards is given in the SPICE manual notation .

COMMAND SPICE Element Card

As with all element cards in SPICE, this card describes a specific instance of the numerical device in the circuit. The name of this instance is used as the predecessor for all files accessed in solving the specific numerical device.

SYNTAX

Nxxxxxxxx n1 n2 . . . nM mname <area> <off>
+ <ic=v1M, v2M . . . v(M-1)M> <temp=T>

PARAMETERS

Nxxxxxxxx

The **N** identifies the element type as being a numerical device and the **x**'s are used to represent a unique character string, which can be up to eight characters long, to identify the device.

n1 n2 . . . nM

The node numbers in the circuit to which the **nM** nodes of the numerical device are connected. They correspond in sequence to the electrodes of the device as specified in PISCES. There is a maximum limit of ten nodes per device. The voltage on any node is specified with respect to the last node as given by the **nodes** parameter on the **.model** card.

mname

The model name links this instance of the numerical device to the appropriate **.model** card.

area

The area or length factor for the device. All currents, capacitances, and conductances are multiplied by this value. The default is 1.0.

off

If **off** is specified, the DC operating point is determined with the terminal voltages for that instance of the device set to zero.

ic

These initial conditions for the device are used with the **UIC** specification on the **.tran** card. Each voltage initial condition is specified with respect to the last node.

temp

Operating temperature of the device. If none is given, the circuit temperature is used as the default.

EXAMPLES

```
Nbjt1 8 7 5 npndev 5
```

This adds a numerical device with the name of Nbjt1 connected to circuit nodes 8, 7, and 5. These nodes correspond to electrode numbers 1, 2, and 3 in the numerical device's mesh. Electrode number 3 is taken as ground. Analogously with SPICE's compact model for a bipolar junction transistor, Electrode 1 (Node 8) is the collector, Electrode 2 (Node 7) is the base, and Electrode 3 (Node 5) is the emitter. An area factor of 5.0 scales the device appropriately.

```
Npn1 6 4 pndev area=6.5 ic=0.75
```

This is a two node numerical device with circuit Node 6 corresponding to Electrode 1 and circuit Node 4 corresponding to Electrode 2. An initial condition of V_{64}=0.75 V is placed on the device. The area is scaled by a factor of 6.5.

BUGS

The **ic** and **off** parameter have not been tested extensively.

Some software paths have not been tested and may result in problems.

COMMAND — SPICE Model Card

As with all .**mode**l cards in SPICE, this one is used to describe all instances of the same type. Common parameters are specified and used by all instances that reference the given model name.

SYNTAX

.model mname mtype nodes=ival [dtype=ival] [vto=rval]
+ [model=ival] [method=ival] [vmax=rval] [vmin=rval]
+ [hifreq=rval] [maxtrys=ival] [level=ival] [float=ival vbc=rval
+ bcdep=ival] [oedev=ival] [tnom=rval]

PARAMETERS

mname

Model name used for reference on the element cards. The mixed-mode interface expects to find certain files with this name as the predecessor to the file names.

mtype

The model type is given by a word mnemonic from Table A.1. The mnemonic indicates the polarity of the numerical device and the mechanism by which the numerical simulation is executed.

A plain **pis** device (independent of polarity) tells the mixed-mode interface to execute each of the numerical device runs sequentially on the same machine as SPICE. For the **prl** model, the mixed-mode interface uses UNIX sockets to execute each numerical simulation on a different machine. For the **pvm** model, the mixed-mode interface uses PVM message passing to send the biases to a queueing routine and ultimately a "slave" routine on a node of the parallel

virtual machine. See “Running a Mixed-Mode Simulation” on page 123.

Table A.1: Model Types for Mixed-Mode Simulations.

mtype	Description
pis	generic numerical device (default)
npis	n polarity numerical device (npn, nmos, pn junction)
ppis	p polarity numerical device (pnp, pmos, np junction)
prl	generic numerical device simulated in parallel
nprl	n polarity numerical device simulated in parallel
pprl	p polarity numerical device simulated in parallel
pvm	generic numerical device simulated using pvm
npvm	n polarity numerical device simulated using pvm
ppvm	p polarity numerical device simulated using pvm

nodes

The number of numerical device nodes. This parameter defaults to 10 if no value is given. SPICE does not have a problem if this number is larger than the actual number of electrodes on the numerical device. The unidentified nodes are connected to ground by default. Likewise, one must be certain that the numerical device simulator can handle a “solve” statement with voltages specified for electrodes that do not exist. To avoid this problem, one should always specify this parameter.

The **nodes** parameter may be less than the actual number of nodes in the numerical device. In this case, the extra nodes on the numerical device are considered floating and always ignored by the SPICE circuit simulator.

Other **.model** parameters related to floating nodes include **float**, **vbc**, and **bcdep**. These parameters allow the user to switch boundary conditions on the floating node during a DC circuit solution.

dtype

This integer value provides SPICE with some a-priori knowledge of the device behavior. This parameter is not necessary, but is useful in achieving circuit convergence. Specifying a device type allows SPICE to make good guesses and limit voltage changes across pn junctions. The supported devices and their integer representations are configured in the NPISCdefs.h file. The default configuration is given in Table A.2. The **mtype** parameter determines the polarity of the device. The electrode sequence for the electrodes is analogous to the SPICE compact model equivalents.

Table A.2: Device Configurations for Mixed-Mode Simulation.

dtype	Description	Electrode Sequence
0	generic (default)	does not matter
1	BJT	Collector Base Emitter Substrate
2	MOSFET	Drain Gate Source Bulk
3	diode	Anode Cathode
4	MESFET	Drain Gate Source Bulk
5	LED	Anode Cathode
6	Inverter	PSource Pgate Pdrain Ndrain Ngate NSource Pbulk Nbulk
7	Resistor	Contact1 Contact2 Substrate (Cathode of diode)
8	JFET	Drain Gate1 Gate2 Source

vt0 or **vto**

This parameter represents the absolute value of the turn on voltage for the device. For a bipolar junction transistor it is Vbe_{on}, for a diode it is Vd_{on}, and for a FET it is Vgs_{on}. The default value at room temperature is 0.7 V which is typical for a silicon pn junction. The polarity of the device model determines the sign.

vmax

This is the maximum voltage (in signed real number) placed on any numerical device electrode with respect to the reference electrode.

The default value is 50.0 volts which one needs to increase for power devices.

vmin

This is the minium voltage (in signed real number) placed on any numerical device electrode with respect to the reference electrode of that numerical device. The default value is -50.0 volts.

hifreq

This value separates low and high frequency simulation of the numerical device. When the circuit frequency is below **hifreq**, the device's small signal parameters are calculated from a zero frequency device simulation. When the circuit frequency is above **hifreq**, the device's small signal parameters are calculated from a small signal device simulation at the given frequency. The trade-off is accuracy versus simulation time. The default value for **hifreq** is 1kHz.

maxtrys

The maximum number of attempts the mixed-mode interface tries to get the device simulation to converge. Different bias stepping schemes are attempted with each try.

level

This integer represents a specific device simulator as defined in the NPISCdefs.h. Currently, the only one supported (and the default) is Stanford PISCES which is defined as level=0. This integer value determines how the **method** and **model** parameters are interpreted as well as the calling of the correct device simulator. More **levels** may be added so that other device simulators may be used in a mixed-mode simulation.

model

This is an integer value for the binary encoded representation of the model card in the device simulator input deck. The meaning of the

integer value is described in the NUMDEVdefs.h file and interpreted by a routine specifically written for that device simulator. If this parameter is not specified and the file **mname.model** exists, the model card is taken as the content of that file. If neither is given, this card is not included in the input deck for the device simulator. See "Running a Mixed-Mode Simulation" on page 123.

method

This is an integer value for the binary encoded representation of the method card in the device simulator deck. The meaning of the integer value is described in the NUMDEVdefs.h file and interpreted by a routine specifically written for that device simulator. If this parameter is not specified and the file **mname.method** exists, the method card is taken as the content of that file. If neither is given, this card is not included in the input deck for the device simulator. See "Running a Mixed-Mode Simulation" on page 123.

float, vbc, and **bcdep**

These three parameters are used to control the boundary conditions on a floating node that can switch between a current and voltage boundary condition during DC analysis. This capability aids in the convergence of numerical devices with a floating node. At low currents in the device one uses a voltage boundary condition on the floating node and for high currents, one uses a current boundary condition on the floating node. The value of the voltage and current source on the floating node is zero.

The parameter **float** contains the integer number for the floating node. It must be greater than the number specified by the **nodes** parameter. The parameter **vbc** determine the voltage at which the boundary condition is switched. The parameter **bcdep** specifies the positive and negative *electrode* on which the boundary condition depends. Refer to the examples for clarification.

oedev

An integer value of one for **oedev** tells SPICE that the numerical device will produce optical output. As a result, SPICE adds an equation to its circuit matrix to store the value of that output. In a future release, an integer value of two will tell SPICE that the numerical device has light incident upon it. The amount of incident light will be determined by referencing a circuit node. The default of zero means no photons are involved.

tnom

This parameter specifies the nominal temperature at which the parameters are given. Currently, the only parameter affected by **tnom** is **vt0**. The default value is 300K.

EXAMPLES

.model npndev npis vt0=0.7 dtype=1 model=93 method=336
+ nodes=3

This example defines the **npndev** model used in the previous example of an element card. This **.model** card tells SPICE that the **npndev** is of polarity **n** and that sequential simulations are executed for each instance of this model. The device type is a bipolar transistor as specified by the **dtype** parameter. The turn-on voltage for the device is 0.7 volts. The **model** and **method** for the device simulation is specified on the card and is explained in the next section. The number of **nodes** is given as three which corresponds to the three nodes on the previous element card.

.model hpled ppis vt0=1.4 maxtrys=2 nodes=2 dtype=5
+ float=3 vbc=1.0 bcdep=21 oedev=1

This is a very complex model used for the simulation of an GaAs/AlGaAs LED with a floating layer. The turn-on voltage for the LED is 1.4 volts as defined by **vt0**. The number of **nodes** is two. The maximum number of tries for a given bias is limited to two. The floating node is electrode number 3 as given by the **float** parameter.

The voltage boundary condition is used when V_{21} is below 1.0 volts and the current boundary is used when V_{21} is greater than or equal to 1.0 volts. The parameter **bcdep** specified V_{21} as the controlling voltage and the parameter **vbc** specifies the voltage of V_{21} when the boundary condition change is to takes place.

In addition, this device will produce optical output and hence, an optical response. The **model** and **method** lines for the numerical simulator input deck is given in the files **hpled.method** and **hpled.model.**

BUGS

The advanced parameters for floating layers have been generalized, but may not be useful for all cases.

When **vmax** and **vmin** are set stringently, they tend to cause SPICE to oscillate between the extremes. These parameters should be set such that they prevent the numerical device from entering breakdown.

Some software paths have not been tested and may result in problems.

A.3 Running a Mixed-Mode Simulation

This section will describe how to set up a mixed-mode simulation using **level**=0 (Stanford's version of PISCES). The first subsection describes the files the user must supply and those generated during the simulation. The next subsection describes the concept of the binary encoded method line and binary encoded model line. Finally, each of the different execution methods is addressed and the requirements for each is described.

A.3.1 Files Required and Generated

mynetlist.spi
This file contains the circuit netlist of all the element cards and model cards for the circuit. Any valid SPICE construct is permitted. The name of the file can be arbitrary, but it is recommended that the extension **.spi** is used so that it may easily be recognized.

mynetlist.raw
The easiest way to run SPICE is to create a net list such that the circuit simulation is performed in batch mode. As a result, the user needs to save the solution SPICE computes. The name of the file can be arbitrary, but it is recommended that the extension **.raw** is used. To run SPICE in batch mode one types the following on the command line:

```
spice3 -b mynetlist.spi -r mynetlist.raw
```

Refer to the SPICE manual for more information .

mname.msh or **mname.mesh.pis**
One of these files *must be supplied* to load or create the mesh during the PISCES simulation. The first is a standard PISCES mesh file. The second contains valid PISCES commands for creating the mesh from scratch. The **mname** corresponds to the model name given on the SPICE **.model** card. The extension **.msh** or **.mesh.pis** must be exactly as shown. The mixed-mode code first looks for **mname.msh**, if that file does not exist, it looks for **mname.mesh.pis**. If neither exists, SPICE aborts.

Although these files are based on PISCES, they should be completely compatible and/or analogous for other device simulators. The actual format is determined by the programmer who adds the new device simulator.

Using **mname.msh** provides for faster simulation because PISCES just loads the mesh and doesn't have to create it each time. However, for some simulations, one may choose to

create the mesh each time and therefore, the second option is available. When using **mname.mesh.pis**, the user *should not* include an end card.

mname.model and/or **mname.method**

These files contain the model and method cards for the PISCES input file, respectively. In writing an input deck, the mixed-mode interface should be given these cards. The binary encoded **model** and **method** parameters on the **.model** card in SPICE are the most efficient way to provide these cards without additional files or overhead. However, the encoding only accounts for logical variables; hence using the files provides a capability for including numerical variables on these PISCES cards.

mname.soln.init

This is a PISCES solution file containing the initial zero bias solution for the model **mname**. This only has to be found once for the model since all instances use the same starting point.

Nxxxxxxxx.pis

This file contains the PISCES input deck that the mixed-mode interface writes for the instance named **Nxxxxxxxx**.

Nxxxxxxxx.out

This file contains the output of the last PISCES simulation run for the specific instance.

Nxxxxxxxx.soln.prev0, Nxxxxxxxx.soln.prev1, and **Nxxxxxxxx.soln.prev2**

These files are generated by PISCES and contain the solutions for the specific instance at some previous bias. For DC analysis, the **.prev0** file contains the solution for the current circuit iteration, the **.prev1** file contains the solution for the previous circuit iteration, and the **.prev2** file contains the solution for iteration before the previous. For transient analysis, .**prev0** file contains the solution for the current time step, the **.prev1** file contains the solution for the previous time step, and the **.prev2** file contains the solution for the time step before the previous time step.

Nxxxxxxxx.log.ac and **Nxxxxxxxxx.log.iv**

These files are generated by PISCES and contain the conductance/small signal parameters and the current vector respectively.

hostname.pvm and hostname.prl

These files are supplied by the user and contain a listing of hostnames if the mixed-mode simulator is used in a parallel configuration. See "Types of Execution" on page 126.

A.3.2 Method and Model Parameters

The user may specify the binary encoded representation of the PISCES model card or PISCES method card on the SPICE **.model** line. This is accomplished by summing the numeral next to the appropriate action as given in the following tabulations.

The binary encoded model parameter is computed as follows:

+1	turns on srh
+2	turns on consrh
+4	turns on auger
+8	turns on bgn
+16	turns on conmob
+32	turns on analytic
+64	turns on fldmob
+128	turns on surfmob
+256	turns on impact
+512	turns on ccsmob
+1024	turns on fermi
+2048	turns on incomplete
+4096	turns on photogen

The binary encoded method parameter is computed as follows:

+1	turns off xnorm
+2	turns on rhsnorm
+4	turns on limit
+8	turns on fixqf
+16	turns on trap
+32	turns on autonr
+64	turns off 2nd
+128	turns on tauto
+256	turns off tauto
+512	turns off l2norm
+1024	turns of extrapolate

For example, to generate the following model and method lines in the PISCES input deck,

model consrh conmob bgn auger

method trap ^2nd ^tauto[1]

the SPICE input file should contain a line similar to:

.model fmrdev npis nodes=3 vt0=0.7 model=30 method=336

A.3.3 Types of Execution

There are three types of execution modes available to the user, but the compiled version of the mixed-mode code may not necessarily have all these capabilities. The type of execution is determined by the type of model given for the numerical device.

pis, npis, and **ppis**
This is the simplest of the possible models. A **pis** model sequentially executes the instances on the machine SPICE is executed. The minimum requirement is a SPICE netlist and a file containing the mesh.

prl, nprl, and **pprl**
This type of model takes advantage of a network of computers. It uses standard UNIX sockets to execute the device simulations on remote hosts. The file system must be mounted on all these hosts with the same absolute path name. After a failure of **maxtrys** attempts, the parallel mode switches to the sequential mode for one attempt only. As a result, if a remote machine crashes or becomes unavailable for some reason, the mixed-mode simulation may not necessarily halt.

The **hostname.prl** file contains a listing of the hosts on which the device simulations are executed. Each numerical device in the circuit is assigned a machine from the list and it is always executed on that machine. If there are more devices than machines then some machines can have more than one device executed on it. It is recommended that the machines be listed in decreasing computational power.

The login and password for the remote machine is required. The mixed-mode code looks for the host name in the **.netrc** file. However, since the login and password for many hosts

1. 2nd and tauto must be turned off for mixed-mode simulations using PISCES.

are the same, the user can use a machine name of “spice” in the **.netrc** file and the mixed-mode code uses that login and password for all hosts not found in the **.netrc**.

pvm, npvm, and ppvm

This type of model uses PVM to execute the numerical device simulations in parallel. PVM stands for Parallel Virtual Machine and may be obtained from netlib@ornl.gov. The user’s account must be configured for use with PVM as described in the manual and the user should have a working knowledge of PVM.

Three files are necessary for using PVM to execute a mixed-mode simulation. The user should place the executable files **npiscctrl** and **npiscslave** in the PVM bin directories of the appropriate machine architectures. In addition, **hostname.pvm** needs to contain a listing of all the nodes used for the mixed-mode simulation; however, they do not need to be added to the parallel virtual machine. If they are not in the parallel virtual machine, the mixed-mode simulator attempts to add them.

Finally, the user should start and/or make sure that the PVM daemon is running on the local machine and then start the SPICE simulation. If the PVM daemon is not running, mixed-mode defaults to sequential execution.

Appendix B: Examples from Mixed-Mode Manual

B.1 Introduction

This chapter describes examples using three types of analyses. The first is a CMOS inverter analyzed with a DC sweep on the input. The second is a GaAs MESFET analyzed for its high frequency response. The third is a six transistor SRAM cell analyzed through a read/write cycle. Another transient analysis is used to analyze a fiber optic transmitting circuit which includes a heterostructure LED with a floating layer.

B.2 Single Stage CMOS Inverter (DC Analysis)

This simple example demonstrates the basic capabilities of mixed-mode. The V_{out} versus V_{in} characteristics of a single CMOS inverter is simulated. Figure B.1: shows the circuit schematic and the SPICE net list for the circuit. The two MOS transistors are numerical devices with electrode number 1 being the drain, 2 the gate, 3 the source and 4 the bulk.

For this simulation the mesh files for the two devices are created separately and are called **fmrpmos.msh** and **fmrnmos.msh**. These mesh files contain the grid and doping for the numerical device. The PISCES method and model options are chosen by the binary encoded SPICE parameters **method** and **model** as follows.

method ^2nd ^tauto trap

model consrh auger bgn conmob fldmob

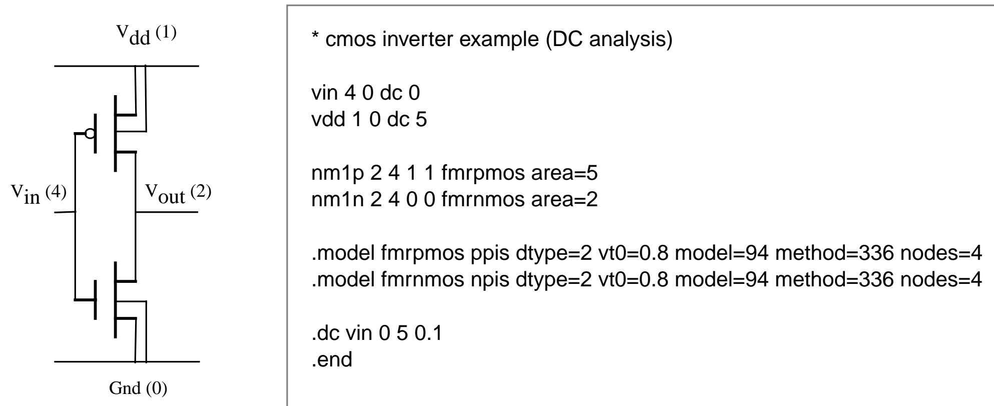


Figure B.1: CMOS inverter circuit schematic and netlist.

For the DC analysis, the input voltage is swept from zero volts to five volts and the output voltage and supply current are monitored. The output waveform is as what would be expected from a CMOS inverter and is not shown.

B.3 High Frequency GaAs MESFET (AC Analysis)

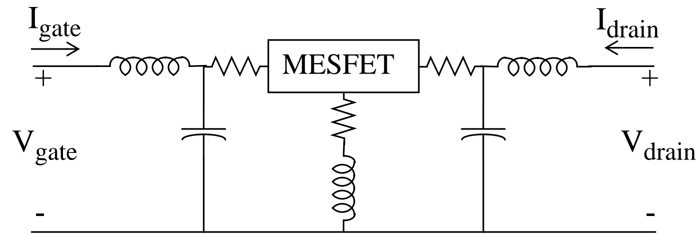
The analysis of the GaAs MESFET mixed-mode simulation consists of finding the S-parameters of the two port network shown in Figure B.2:. SPICE can not directly find the S-parameters; however, it can easily find the Y-parameters with an AC perturbation applied to one input while the other is shorted. Two SPICE net lists are created, **mes1.spi** and **mes2.spi**. One perturbs port 1 of the network and the other perturbs port 2; hence each provides Y*1 and Y*2 respectively. One of the netlists is shown in Figure B.2:.

For this simulation, the file **mes.mesh.pis** contains the PISCES cards for creating the MESFET's mesh during each device simulation. Note that nothing specific has to be written in the SPICE netlist to differentiate using a previously created mesh file. In addition, the files **mes.model** and **mes.method** are also supplied and each contains one of the following lines respectively.

model conmob fldmob srh hypert impact

method trap ^2nd ^tauto itlim=30

An AC analysis is performed at each input of the two port network with other shorted. The Y-parameters are calculated by taking the currents I_{gate} and I_{drain} and dividing by the AC




```
* MESFET circuit

vgg 9 0 dc 0.0 ac 1
vdd 10 0 dc 2.0

vgate 9 8 0
vdrain 10 1 0

ld 1 2 0.08n
rd 2 3 1.52
cd 2 0 212f

lg 8 7 0.105n
rg 7 4 0.92
cg 7 0 208f

ls 0 6 0.005n
rs 6 5 1.38

nm1 3 4 5 0 mes area=100

.model mes npis vto=1 maxtrys=10 nodes=4 dtype=4
.ac lin 40 1G 40G
.end
```

Figure B.2: Two port network and netlist for AC analysis to find the Y-parameters

excitation magnitude. A simple transformation yields the S-parameters for a 50 ohm cable. The results are displayed in Figure B.3:.

B.4 SRAM Cell (Transient Analysis)

This example shows a circuit simulated with transient analysis and utilizing a network of computers. Figure B.4: shows the circuit diagram for the SRAM circuit. The shaded region is the six transistor SRAM cell and is simulated using numerical MOS devices. The unshaded region is the control circuitry and is simulated with MOS level 2 compact models.

The transient analysis of the circuit consists of a write and read cycle. The output of the flip-flop and the data lines are observed. Figure B.5: contains the net list for the circuit. There is an extra compact NMOS transistor for leveling the charges of the data line

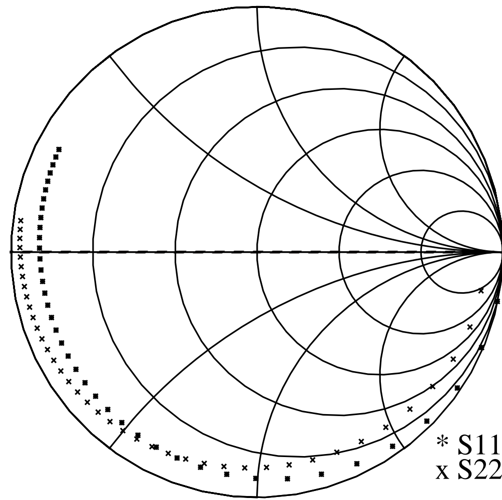
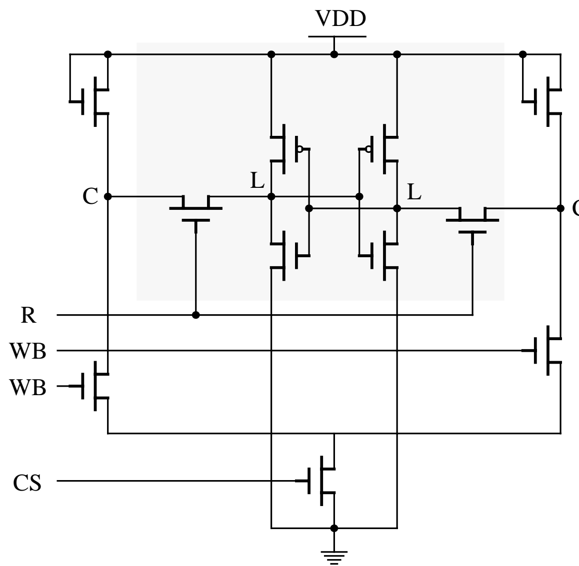


Figure B.3: S-parameters for GaAs MESFET in the frequency range of 1GHz to 40GHz.



CS:	Selects appropriate column
R:	Selects appropriate row
Read:	CS high, R high, WB low, $\overline{WB}$ low
Write 1:	CS high, R high, WB high, $\overline{WB}$ low
Write 0:	CS high, R high, WB low, $\overline{WB}$ high
Output:	Measured across $C\overline{C}$

Figure B.4: Simplified circuit diagram of SRAM cell including control circuitry.

```
* supply line
Vdd 1 0 5

* the cell itself
NM1 4 5 1 1 fmrpmos area=2
NM2 5 4 1 1 fmrpmos area=2
NM3 4 5 0 0 fmrnmos area=4
NM4 5 4 0 0 fmrnmos area=4

* estimated parasitic of the cell
Ci1 5 0 0.01p
Ci2 4 0 0.01p

* pass gates
NM5 4 7 2 0 fmrnmos area=2
NM6 5 7 3 0 fmrnmos area=2

* control lines
M7 1 1 2 0 ndev l=2u w=25u
M8 1 1 3 0 ndev l=2u w=25u
M9 2 9 6 0 ndev l=2u w=25u
M10 3 8 6 0 ndev l=2u w=25u

* access line
M11 6 10 0 0 ndev l=2u w=50u

* level line
M12 2 12 3 0 ndev l=2u w=25u

* estimated load on lines
Crow 7 0 0.5pf
Ccol 2 0 2pf
Ccolb 3 0 2pf

* access control sources
VCS 10 0 pulse(0 5 0.2n 0.2n 0.2n 8n 20n)
VR 7 0 pulse(0 5 0.2n 0.2n 0.2n 8n 20n)

* data control sources */
VWBb 8 0 pulse(0 5 40.2n 0.2n 0.2n 8n 80n)
VWB 9 0 pulse(0 5 0.2n 0.2n 0.2n 8n 80n)
```

Figure B.5: SPICE netlist for SRAM simulation.

capacitances. After a read or write cycle, this transistor turns on to allow the charge between the data lines to distribute evenly. The data lines must be equal before anymore actions can take place.

The models for the numerical devices are the same as those used for the CMOS inverter. The compact model is SPICE Level 2 model which closely represents the numerical model. The devices in the circuit are not the state-of-the-art, but they do provide a good example. The SPICE **.nodeset** line is used to set the flip-flop to an initial state rather than having it start indeterminately. The **.tran** card specifies the limits of the transient analysis.

```
* data line level sources
VL 12 0 pulse(0 5 9n 0.2n 0.2n 8n 20n)

.nodeset v(5)=0 v(4)=5 v(2)=3.5 v(3)=3.5
.tran 0.1n 40n

* compact models
.model pdev pmos
+ level = 2.000 vto = -0.5677 gamma = 0.51
+ phi = 0.73 tox = 2.5000E-08 nsub = 4.0251E+14
+ nfs = 6.8412E+11 xj = 2.5976E-8 ld = 1.8553E-07
+ uo = 1.705E+02 ucrit = 2.2295E+05 uexp = 0.2918
+ vmax = 6.8950E+04 neff = 9.285E+1 delta = 0.5587
+ cj = 2.7923E-04 cjsw = 2.1805E-10 pb = 0.7316
+ mj = 0.4729 mjsw = 0.2195 fc = 0.5
+ cgdo = 4.71e-10 cgso = 4.71e-10

.model ndev nmos
+ level = 2.000 vto = 0.6000 gamma = 0.4500
+ phi = 0.6000 tox = 2.5000E-08 nsub = 1.6300E+15
+ nfs = 3.0000E+11 tpg = 1.000 xj = 3.5000E-07
+ ld = 1.0000E-07 uo = 612.0 ucrit = 2.0000E+05
+ uexp = 0.2500 vmax = 5.8000E+04 neff = 45.00
+ delta = 3.900 cj = 1.4620E-04 cjsw = 3.1626E-10
+ pb = 0.6495 mj = 0.3640 mjsw = 0.2558
+ fc = 0.5000 cgdo = 4.39e-10 cgso = 4.39e-10

* numerical devices
.model fmrpmos npvm dtype=2 vt0=0.8 model=94 method=336 nodes=4
.model fmrnmos npvm dtype=2 vt0=0.8 model=94 method=336 nodes=4
```

Figure B.6: Continuation of SPICE netlist for SRAM simulation.

The numerical device models are specified as **pvm**. As a result, each numerical device simulation is executed on a different node of a parallel virtual computer. The file **hostname.pvm** is provided with a list of hostnames for the nodes. The network uses a common file server whose disks are mounted on each of the remote hosts.

Figure B.7: shows the output waveforms of the SRAM cell. It is important to note how fast the difference between the data lines becomes small. This time determines the minimum read/ write cycle time.

B.5 GaAs/AlGaAs LED (Transient Analysis)

This final example involves a device with a floating node. Figure B.8: shows the cross section of the cylindrical LED. The cathode contact is electrode number 1, the anode contact is electrode number 2, and the floating layer is electrode number 3. In order to improve convergence in the device simulation, the floating layer must switch from a

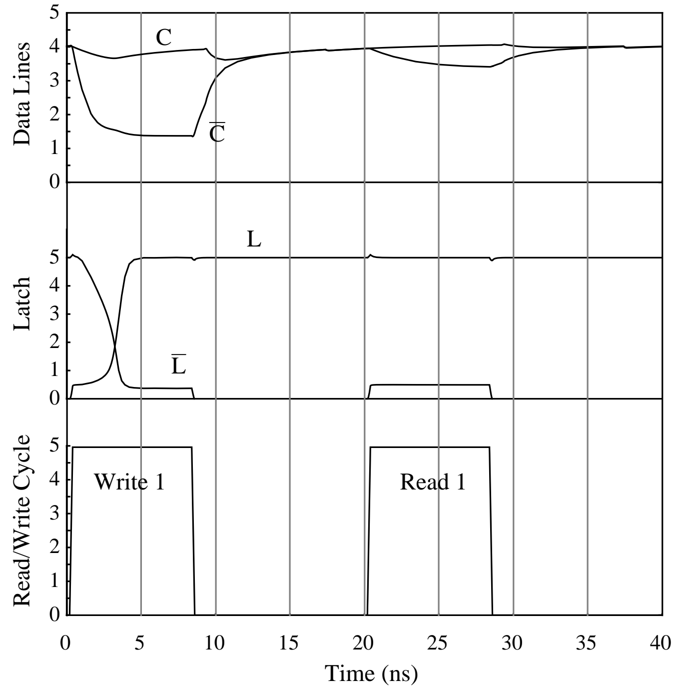


Figure B.7: Read/Write cycle for SRAM cell.

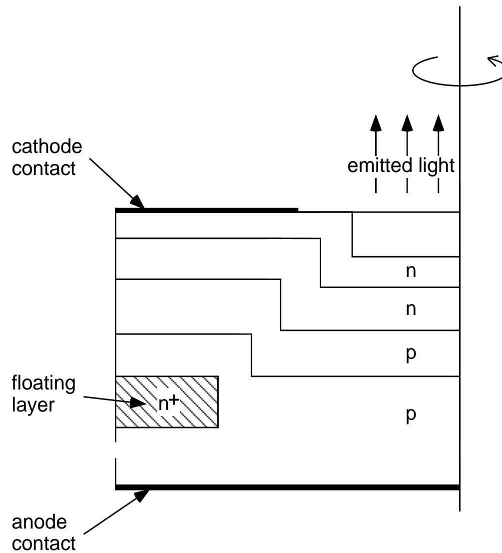
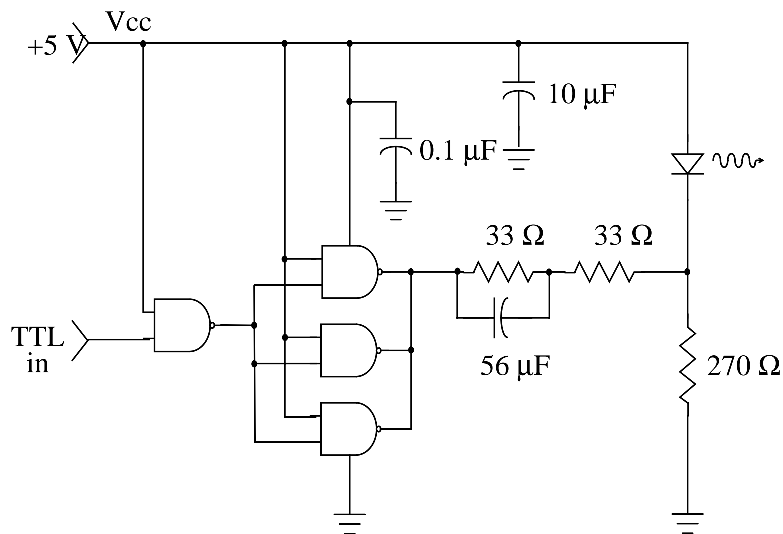


Figure B.8: Cross section of LED structure that is used in a fiber optic transmitter circuit.

voltage boundary condition to a current boundary condition when high currents are flowing. Refer to the paper by Dutton for a detailed discussion.

At low currents and biases, the voltage boundary condition introduces some error, but it is tolerable since that is not the range of interest for the operation of the device. For higher




```
* driver circuit for HP transmitter

* supply
Vcc 6 0 DC 5

Vin 1 0 pulse(5 0 1n 1.2n 1.2n 20n 40n)
Rin 1 2 1

* driver circuit
R8 2 3 33
C4 2 3 75p
R9 3 4 33
r10 4 0 270

Vhp1 6 5 DC 0
nhp1 4 5 hpled

.options abstol=1e-9
.model hpled ppis vt0=1.4 maxtrys=2 nodes=2 dtype=3 float=3
+ vbc=1.0 bcdep=21

.tran 100p 40n 0 200p
.end
```

Figure B.9: Circuit schematic and netlist for LED transmitter simulation.

currents, the node is kept floating by placing a current boundary condition with a current source value of zero. In order to handle this switching, there are three special parameters.

This device is placed into the transmitter circuit shown in Figure B.9:. The purpose of this circuit is to take an ECL signal, convert to TTL, and either turn on or off the LED. The SPICE netlist for the circuit is also given.

First, one notes that this device is called using **ppis** model because a **ppis** device is defined such that the cathode is electrode 1 and the anode is electrode 2. An **npis** device has the reverse. In general, the n-polarity devices are such that the device is in its SPICE standard forward operating state.

The rest of the parameters on the **.model** line are set to account for the heterostructure device. **vt0** is set to turn on the device. The **maxtrys** parameter limits the maximum number of tries to two. This parameter is set low because the cutting of the bias step does not take into account a changing boundary condition. The number of nodes is set to the actual number of contacts in the SPICE circuit. The floating node is not connected to anything and hence, SPICE does not need to know about its existence.

The three parameters **float**, **vbc**, and **bcdep** specify the node and the switching point for the boundary condition.The floating node is electrode number 3 as given by the **float** parameter. The voltage boundary condition is used when V_{21} is below 1.0 volts and the current boundary is used when V_{21} is greater than or equal to 1.0 volts. The parameter **bcdep** specifies V_{21} as the controlling voltage and the parameter **vbc** specifies the voltage of V_{21} when the boundary condition change is to take place.

The device uses cylindrical coordinates and hence, the area factor is allowed to default to 1.0. The mesh is created each time the device is solved because the version of PISCES being used can not save the heterostructure parameters. Hence, the file **hpled.mesh.pis** must be supplied. In addition, the model and method cards are given in the files **hpled.model** and **hpled.method** and are as follows.

```
models fldmob auger conmob rad consrh
method itlim=50 p.tol=5.e-5 c.tol=5e-5 ^2nd ^tauto
```

A communications circuit engineer is most interested in the turn-on turn-off characteristics of the diode. Therefore, the transient simulation consists of a pulse input to switch the diode on and back off. The TTL nand gates used for supplying the current are substituted with a voltage source and a resistor. The voltage source takes into account the finite rise and fall time of the nand gates while the resistor simulates the output resistance of the nand gate.

Figure B.10: shows the response of the diode. This current is directly related to the amount of photons out for the device.

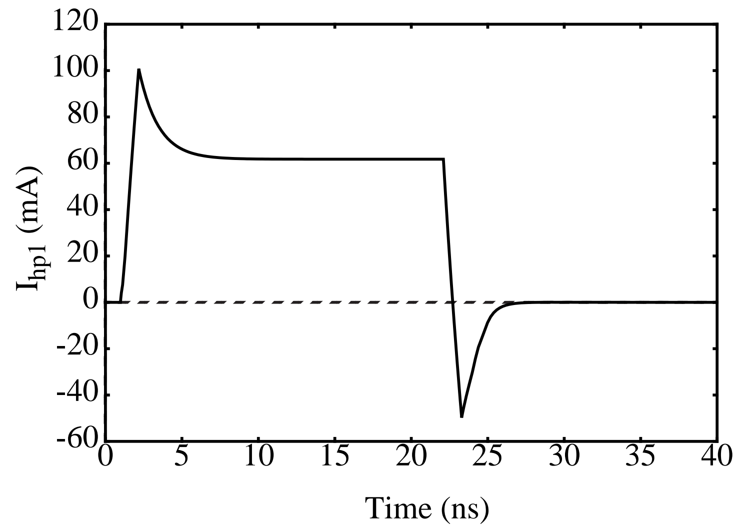


Figure B.10: Time transient response of the current flowing through LED in a transmitter circuit.

Appendix C: Users Manual for PISCES2-HB[1]

C.1 Introduction

New parameters and new cards were added to the PISCES in order to access the new capabilities. In addition, for circuit boundary conditions, a SPICE like input deck is required to describe the circuit configuration. This chapter describes the new PISCES cards and parameters as well as the circuit description input netlist.

C.2 PISCES Card Additions

The improvements from PISCES-2H to PISCES-2HB involves additions for harmonic balance simulation, circuit boundary conditions, new models, and improved numerics. Each change results in new parameters in order to invoke the capability. In addition, harmonic balance simulation necessitates the additional card. **hbmeth** is used to define the parameters for the numerics involved in harmonic balance simulation and is similar to the **method** card. In addition, a separate file containing a SPICE-like netlist is required to describe the circuit surrounding the PISCES device. The format for this file is described after the new PISCES parameters and cards are catalogued.

---

1. This version of PISCES, containing links to proprietary EEsof (Agilent) software, is not generally available for public distribution. In the cases of collaborative research cited in this thesis, EEsof has provided the necessary routines on a case-by-case basis.

COMMAND CONTACT

The new parameter on this card allows for the specification of the input file containing the circuit boundary conditions.

SYNTAX

Contact cktfile = <filename>

NEW PARAMETERS

cktfile

This parameters specifies the input file that contains the netlist for the linear circuit surrounding the device.

EXAMPLE

```
contact cktfile=myckt.ckt
```

COMMAND HBMETH

This new card specifies numerical limits and methods for the harmonic balance module. It is similar in nature to the **method** card.

SYNTAX

hbmeth [maxiter = <integer>] [useredpc] [krtol = <real>]
+ [kriter = <integer>] [gmrestart = <integer>]
+ [pkthresh = <real>] [gmrthresh = <real>]
+ [krloosetol = <real> krlooseiter = <integer>]
+ [krconvratio = <real>]

PARAMETERS

maxiter

This parameter specifies the maximum number of Newton iterations before the HB simulator aborts and steps the sources down to try again. A reasonable value is 30 although a good number is not known in practice (Default: 30).

useredpc

This logical variable is used to specify the 'reduced' pre-conditioner. This parameter is needed only for two-tone problems with very tightly or very widely spaced tones and reduces memory dramatically (Default: false).

krtol

This variable specifies the tolerance for the iterative linear (Krylov) solves during the Newton process. Reasonable values are 1×10^{-3} and 1×10^{-4} (Default: 1×10^{-4}).

kriter

This parameter specifies the maximum number of iterations during the iterative linear solve before it gives up and just takes the Newton step anyway. A value of 30-50 is reasonable (Default: 40).

gmrestart

This integer value is the number of iterations that GMRES will take before "restarting." In theory, the higher this number, the faster and more robust the linear solve will be. However, an additional vector has to be stored for each GMRES iteration, so its value cannot be too large. 10 tends to work well (Default: 10).

pkthresh

This parameter is used to reduce Jacobian memory storage for large problems. Spectral components less than (**pkthresh** x DC_value) are dropped (Default: 0.0).

gmrthresh

This parameters is used to reduce GMRES memory storage for large problems. GMRES vector entries less than **gmrthresh** are dropped (Default: 0.0).

krloosetol

This parameter is used jointly with **krlooseiter**. If **krlooseiter** GMRES iterations are completed and the GMRES error is below **krloosetol**, then the GMRES iterations are considered adequate for proceeding with a Newton step (Default: 0.1).

krlooseiter

This parameter is used jointly with **krloosetol**. If **krlooseiter** GMRES iterations are completed and the GMRES error is below **krloosetol**, then the GMRES iterations are considered adequate for proceeding with a Newton step (Default: 30).

krconvratio

When the ratio between two successive GMRES reaches the value of this parameter, GMRES assumes progress toward a solution is stalled and therefore, aborts (Default: 1.0).

EXAMPLE

```
hbmeth krtol=5e-4 kiter=35 ^useredpc
```

COMMAND LOG

The new parameters on this card allows for the storage of the circuit solution independent of the device solution.

SYNTAX

log cktfile=<filename> [column.out | spice.out | key.out]

NEW PARAMETERS

cktfile

This parameter specifies the output file name for the circuit solution. It contains the voltage at each circuit node, the current through all voltage sources and inductors, and the current flowing into each node of the PISCES device including any scaling factors (Default: No data saved).

column.out

This flag forces the output data into columns with a heading describing the data in columns. It can be post processed by many generic tools and spreadsheets (Default: True).

spice.out

This flag forces the data to be saved in a Berkeley SPICE-like raw file format. For those with access to Berkeley's plotting tools, this format provides an easily accessible tool to analyze the circuit solution. **spice.out** is only available for **opckt**, **acckt**, **dcckt**, or **trckt** (Default: False).

key.out

This flag forces the output data into two files. One file has an extension of **.dat** add to the **cktfile** name and the second has an

extension of **.key**. The **.dat** file contains the simulation results in columns. The **.key** contains a mapping of the solution variables to the column number in the **.dat**. This output format allows the user to plot the data in many generic tools by excluding the heading information in the data file which can cause problems with some plotting tools (Default: False).

EXAMPLE

log cktfile=mydata.dat column.out

log cktfile=mydata.raw spice.out

log cktfile=mydata key.out

COMMAND SOLVE

Many parameters are added to the **solve** card in order to take advantage of the new capabilities in PISCES. A new set of parameters are provided to invoke the circuit boundary condition and the harmonic balance solution. In addition, solution input/output parameters are provided for the harmonic balance results.

SYNTAX

solve [opckt | acckt | dcckt | trckt | acdcckt | hbckt]
+ [hboutfile=<filename> [savestep=<integer>]]
+ [hbinfile=<filename>]

PARAMETERS

opckt

Solves for the operating point on the circuit (Default: False).

acckt

Solves the small signal AC circuit. This parameter requires an **.ac** card in the circuit file (Default: False).

dcckt

Solves the DC circuit for the values of the swept source(s) specified on the **.dc** card in the circuit file (Default: False).

trckt

Solves for the transient response of the circuit. This card requires a **.tran** card in the circuit file. In addition the **tranckt** may be used in lieu of **trckt** (Default: False).

acdcckt

Solve the small signal AC circuit while sweeping DC source(s). This card requires **.dc** and **.ac** cards in the circuit file. In addition **dcacckt** may be used in lieu of **acdcckt** (Default: False).

hbckt

Solves the large signal AC circuit using the harmonic balance simulator. This card requires the HB simulator module and the **.hb** analysis card in the circuit file. A sweeping of the DC and/or AC source can be specified with the **.hbac** and/or the **.hbdc** card in the circuit file (Default: False).

hboutfile

The base part for the name of the file to store the solution at a given simulation point. To this name an extension of "**.sol**" is added for the data for Ψ, n, and p. An extension of "**.trm**" is added to the name for the data file containing the terminal characteristics. A "**.nds**" extension is added for the data file containing the circuit node solutions.

savestep

During a sweep of a source in a harmonic balance simulation, **savestep** is used to specify how often the harmonic balance solution is saved given a file base part specified by **hboutfile**. At each integer multiple of the given value, the last letter of the output file name is incremented and the solution is saved to that file. A value of 0 means only the last completed solution is saved (Default: 0).

hbinfile

Use the given file name as the starting point for the next harmonic balance simulation. This file contains the "**.sol**" extension and it must be specified when the file name is given.

```
solve acckt

solve hbckt hboutfile=hbsolnA hbsavestep=15
```

C.3 The Circuit File

A separate circuit file is needed to specify the linear circuit surrounding the PISCES device. The circuit file contains a SPICE-like netlist to describe the connections of the external circuitry. The standard SPICE linear elements are provided along with a special element for specifying the electrode connections for the linear devices. The standard analysis capabilities are provided with the **.op**, **.dc**, **.ac**, and **.tran** cards. In addition there are specialty dot (“**.**”) cards for the harmonic balance analysis.

The **contact** card in PISCES contains a parameter which is used to specify the file name for the circuit file. Even if a circuit file is specified for boundary conditions, the boundary conditions are not invoked unless one of the circuit solves is specified on the **solve** card. Therefore, the recommended sequence of cards in PISCES is as follows:

```
solve init outfile=soln.ini
solve v1=1.0 vstep=1.0 nstep=4 num=1
solve v2=0.25 vstep=0.25 nstep=5 num=2
solve v1=5.0 v2=1.5 outfile=soln.dcini
solve opckt outfile=soln.op
solve trckt
```

In this sequence, an initial solution is computed and saved in the file `soln.ini`. The next card ramps up the voltage on electrode number one, which could be a drain or collector. The third `solve` card ramps up the voltage on electrode two, which could be a gate or base. The fourth card solves for the approximate operating point and stores that solution in file `soln.dcini`. The fifth **solve** card accesses the circuit boundary conditions and solves for the operating point solution with the inclusion of circuit boundary conditions. The final **solve** card specifies a transient simulation with circuit boundary conditions. It could just as easily specify an ac, dc, or harmonic balance circuit analysis. When the simulation is restarted or fails, one can now use the **load** card to restart from anywhere solution has been saved for.

The next few pages provides description of the elements and analyses cards that can be specified in the circuit file. At the end of the chapter some limitation on node numbers and information on value specifications are outlined.

COMMAND COMMENT LINES

Any line with an ***** (asterisk) in the first column is considered a comment and is ignored during parsing.

```
* This is a comment
```

COMMAND NUMERICAL DEVICE

Nxxxxxxxx N1 . . . Ni

The PISCES device is specified by the numerical device element card. **xxxxxxxx** uniquely identifies the device and **N1** through **Ni** are the numerical nodes to which the device is connected in the circuit. The number of node connections must equal the number of electrodes defined in the PISCES deck, otherwise the simulation will abort.

Nldmos 14 3 0

STANDARD ELEMENTS

Rxxxxxxxx N1 N2 val
Lxxxxxxxx N1 N2 val
Cxxxxxxxx N1 N2 val

The **R**, **L**, and **C** identify the element as either a resistor, an inductor, or a capacitor, respectively. **N1** and **N2** are the numeric node numbers to which the element is connected. The **val** variable represents the value of that element. The standard MKS units can be used to specify values and are described later in this document.

```
Rfeeback 23 81 1e3
Lpackage 87 63 93n
Cintercon 9 38 2.33p
```

COMMAND TRANSMISSION LINE

Txxxxxxxx N1 N2 GND R L C G Len Nsections

This element specifies a two port (common ground) distributed T-Network transmission line as shown in the figure. **N1** and **N2** are the numeric node numbers at each end of the line and **GND** is the common ground contact. The distributed nature of the line is created by T-sections where **R**, **L**, **C**, and **G** specify the resistance per unit length, the inductance per unit length, the capacitance per unit length, and the admittance per unit length. **Len** is the total length of the line in unit lengths and **Nsections** is the number of T-sections used to simulate the line. Hence, the inductance of each T-section is given by **l** = **L*****Len**/(**2*****Nsections**), the resistance is given by **r** = **R*****Len**/(2***Nsections**), the capacitance is given by **c** = **C*****Len**/**Nsections**, and the admittance is given by **g** = **G*****Len**/**Nsections**. Note that the circuit portion of any simulation is very small compared to the device simulation. Hence, a relatively large number of sections not only improves the transmission line model, but also does not realistically affect the device simulation.

```
Tmatch 8 10 0 0 28n 100p 0 1 200
Tmatch 7 9 0 0 14n 50p 0 2 200
Tinput 2 3 0 1u 82n 54p 3n 5 100
```

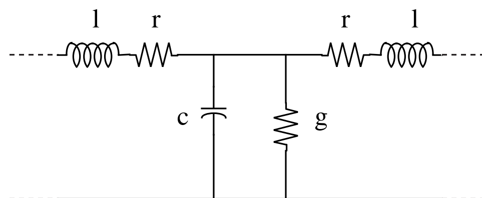


Figure C.1: One T-section of a transmission line defined with N sections.

DEPENDENT SOURCES

Gxxxxxxxx N+ N- n+ n- val
Exxxxxxxx N+ N- n+ n- val

G and **E** specify a linear voltage controlled current source and a linear voltage controlled voltage source. **N+** and **N-** specify the positive and negative nodes of the source. Current flows from the positive node to the negative node. **n+** and **n-** specify the voltage nodes which control the sources. **val** is the transconductance (in mhos) or the voltage gain respectively.

Fxxxxxxxx N+ N- Vname val
Hxxxxxxxx N+ N- Vname val

F and **H** specify a linear current controlled current source and a linear current controlled voltage source. **N+** and **N-** specify the positive and negative nodes of the device. Current flows from the positive node to the negative node. **Vname** specifies the voltage source whose current controls the devices. **val** is the current gain or transresistance (in ohms) respectively.

```
Gcond 0 12 20 0 1m
Egain 10 0 20 0 50
Fgain 0 40 VinCur 3
Ftran 23 0 VoutCur 1k
```

COMMAND INDEPENDENT SOURCES

Vxxxxxxxx N+ N- <source description>
Ixxxxxxxx N+ N- <source description>

V and **I** specify an independent voltage source and current source, respectively. **N+** and **N-** are the numerical nodes to which the source is connected. Current flows from the positive node to the negative node through the source.

The source descriptions are as follows. A voltage source is used for reference, but all descriptors can be used with a current source.

DC Sources

Vname 1 0 [dc] [value]

The **dc** specifies the source as a DC source and the value is the given source value. Neither **dc** nor **value** needs to be specified. If only **value** is specified then the source is assumed to be a DC source of that value. If neither **dc** nor **value** or if only **dc** is specified then the source is assumed to be of DC value zero.

Vgate 5 0 dc 2.3

Vcc 1 0 5.0

AC Sources

Vname 1 0 [dc] [DCvalue] ac [ACvalue]

The **ac** specifies this source as being AC and is required on all AC sources. **ACvalue** specifies the amplitude of the AC small signal generated by the source. If no value is given, **ACvalue** is assumed to be 1.0. An optional **dc** specification may be placed on the source to indicate that it also supplies a DC bias.

Vperturb 5 0 dc 2.3 ac 0.1

Vinput 9 0 ac 0.6

TRANSIENT SOURCES

Vname 1 0 <transient source type>

Transient sources can take on a number of different function as described by the following functions.

Piece-wise Linear (pwl)

Vname 1 0 pwl t0 v0 t1 v1 t2 v2 t3 v3 . . .

Piece-wise linear sources are described by a discontinuous function where each (time, voltage) point is connected by a linear change in the value of the source. The points in the function are given as:

Time	Value
t0	v0
t1	v1
t2	v2
...	...

if t0 != 0 then the voltage at t=0 is set equal to the voltage at t = t0.

Vinput 10 0 pwl 0 0.0 1n 0.0 2n 0.5 10n 2.0 20n 2.0 21n 0.0

Periodic Square Pulse (pulse)

Vname 1 0 pulse v1 v2 td tr tf pw per

A **pulse** is simply a piece-wise linear function specified in an easier manner to define a periodic function. The variables are defined as:

Variable	Description
v1	initial value
v2	pulsed value
td	time delay
tr	rise time
tf	fall time
pw	pulse width
per	period

and the function takes on the following values as a piece-wise linear function:

Time	Value
0.0	v1
td	v1
td+tr	v2
td+tr+pw	v2
td+tr+pw+tf	v1
per+td	v1
per+td+tr	v2
...	...

```
Vname 1 0 pulse -0.5 0.5 10n 1n 1n 50n 100n
```

Sinusoidal Function (sin)

Vname 1 0 sin Vo Va freq td theta

This source takes on the following sinusoidal functional values within the time ranges specified. Note that this does not define a source for harmonic balance, but rather defines a source for transient analysis.

$0 < t < \text{td}$	Vo
$\text{td} < t$	$\text{Vo} + \text{Va}\ e^{(-(t - \text{td})\text{theta})} \sin(2\pi\ \text{freq}\ (t + \text{td}))$

where:

Variable	Description
Vo	DC offset value
Va	sinusoid amplitude
freq	frequency in Hz
td	delay time
theta	damping factor

Vname 1 0 sin 3.3 0.5 50Meg 0.0 0.0

Exponential Pulse (exp)

Vname 1 0 exp v1 v2 td1 tau1 td2 tau2

The exponential source takes on the following exponential functional values within the time ranges specified.

$0 < t < \text{td1}$	V1
$\text{td1} < t < \text{td2}$	$\text{V1} + (\text{V2} - \text{V1})\left(1 - e^{\left(-\frac{-(t - \text{td1})}{\text{tau1}}\right)}\right)$
$\text{td2} < t$	$\text{V1} + (\text{V2} - \text{V1})\left(1 - e^{\left(-\frac{(t - \text{td1})}{\text{tau1}}\right)}\right) + (\text{V1} - \text{V2})\left(1 - e^{\left(-\frac{(t - \text{td2})}{\text{tau2}}\right)}\right)$

where:

Variable	Description
v1	initial value
v2	pulsed value
td1	rise delay time
tau1	rise time constant
td2	fall delay time
tau2	fall time constant

Vname 1 0 exp -1.5 0.75 3n 25n 50n 10

Single Frequency FM (sffm)

Vname 1 0 sffm vo va fc mdi fs

The sffm source takes on the following single frequency FM functional value over all time:

$$\mathrm{Vo} + \mathrm{Va}\sin(2\pi\ \mathrm{fc}\ t + \mathrm{mdi}\sin(2\pi\ \mathrm{fs}\ t))$$

where:

Variable	Description
Vo	DC offset voltage
Va	amplitude
fc	carrier frequency
mdi	modulation index
fs	signal frequency

Vname 1 0 sffm 0.0 1m 108Meg 5 40k

Vname 1 0 hb Vo Vmag freq theta

Harmonic balance source is used to define the large signal AC source applied to the circuit during harmonic balance analysis. Note that a sinusoidal transient source is ignored during harmonic balance analysis and likewise, a harmonic balance source is ignored during transient analysis. All AC sources in the HB simulation must be specified with this source type. The parameters are defined as follows:

Variable	Description
Vo	DC offset voltage
Vmag	amplitude of sinusoid
freq	frequency of sinusoid
theta	phase of sinusoid (deg.)

Hence the source takes on the following values for the given frequencies:

Frequency	Value
DC	Vo
freq	$\mathrm{Vm}\cos(\mathrm{theta}) + j\mathrm{Vm}\ \sin(\mathrm{theta})$
$\mathrm{n} \times \mathrm{freq}$	0.0

Vname 1 0 hb 0.0 0.5 1.2G -90

COMMAND .OPTIONS CARD

The **.options** card is used to set numerical parameters related to the circuit simulation.

SYNTAX

.options [gmin=value] [rmin=value] [area=value] [ltertol=value] [ltevtol=value] [lteitol=value] [ltelim=value]

PARAMETER

gmin

The **gmin** parameter specifies the smallest value for a conductance used in the circuit simulation. A **gmin** conductor is placed from every node to ground. This additional large resistance adds a leakage current, but guarantees that the matrix is not singular. Note that too small a value of **gmin** may cause ill-conditioning. For most applications, **gmin** could be set to zero and hence, have no affect. However, if there are very few or no passive components in the circuit, **gmin** has to be larger than zero. (Default: 1.0E-12 mhos).

rmin

The **rmin** parameter is used to specify a small resistance for places in the circuit where a small short-circuit is required. This resistance is typically used for an inductor connecting a voltage source solely to an electrode of the PISCES device. Inductors are replaced with a zero volts source during DC analysis and hence, a zero resistance in this situation can causes the matrices to go singular. Like **gmin**, the effect of **rmin** is minimal. Only a small voltage is typically lost across the resistor and it guarantees that the matrix is not singular. The **rmin** parameter can be set to zero for most cases, but should a matrix inversion error occur, this parameter should be set to some small inconsequential value (Default: 1.0E-12 ohms).

area

PISCES solves a device in only two dimensions. The area parameter is a scaling factor for all currents in order to provide a quasi-3D solution. For most applications, this parameter refers to the width of the device although symmetry (like in BJT's) could mean **area** is equal to twice the device length or more.

ltertol, ltevtol, lteitol, ltelim

These four parameters are used in the calculations of the local truncation error during transient analysis. For most applications, the default values for these parameters are sufficient, but a brief description is provided for a user who may want to adjust the time step controls of the circuit portion of the simulation. For a more detailed description, please refer to the paper by Bank, *et al*.

ltertol: The relative tolerance for calculating the errors associated with each time step (Default: 0.001).

ltevtol: The absolute error tolerance for voltages (Default: 1mV).

lteitol: The absolute error tolerance for currents (Default: 1pA).

ltelim: The limit on the value for the norm of the error tolerance. If the norm is greater than this value, the time step is reduced and repeated. If the norm is less than this value, the time step is accepted and the next dt is calculated (Default: 1.0).

In order to increase the accuracy in the overall solution, the relative error tolerance (**ltertol**) can be decreased. Likewise, in order to relax the accuracy, **ltertol** can be increased. In order to affect only the voltage or the current, **ltevtol** or **lteitol** should be adjusted in the same manner as **ltertol**. Finally, in order to tighten the time steps, **ltelim** can be reduced or likewise, to relax the time steps, **ltelim** should be increased.

All these parameters only affect the circuit portion of the transient analysis. In most cases, the device simulation tends to limit the time steps, and hence, these parameters should rarely be changed.

EXAMPLES

```
.options gmin=0.0 rmin=0.0 area=100
```

COMMAND ANALYSIS CARDS

A set of analysis cards provides the user a way to specify the limits and conditions on the different types of analysis. Note that PISCES is used to select the desired analysis via the solve card; hence, multiple analysis cards can exist in one file.

Operating Point Simulation

.op

Compute the operating point solution.

DC Sweep Simulation

.dc Sname1 start1 end1 step1 [Sname2 start2 end2 step2]

The **.dc** card allows for the sweeping of DC sources. Up to two sources may be swept simultaneously where the second source is swept inside the first source. The **.dc** requires at least one source and set of sweep parameters to be specified. The **start** parameters refers to the start value, the **end** parameter refers to the end value, and the **step** refers to the stepping value. The source is stepped until its value is greater than or equal to its end value.

.dc Vin 0.0 0.5 0.05

AC Simulation

.ac [dec | lin] numsteps startf endf

The **.ac** card specifies a small signal sweep in frequency. All ac sources are swept over the specified frequencies as determined by this card. The sweeping can be linear or logarithmic as specified by the **lin** or **dec** parameter. The **lin** parameter means the **numsteps** are taken linearly from **startf** to **endf**. The **dec** parameter means that

numsteps are taken logarithmically and there are **numsteps** per decade.

.ac lin 10 1k 10k

.ac dec 5 100k 1G

Transient Simulation

.tran tstep tstop [tstart tmax]

The **.tran** card specifies the time range and time steps for a transient analysis. The **tstep** parameter specifies the initial time step. If the backward Euler method is selected on the PISCES method card, this time step is used throughout the entire PISCES/circuit simulation. If the BDF method is selected on the PISCES method card, then **tstep** is the initial time step and time step estimation is used to select all future values. The **tstop** parameter specifies the time at which the simulation stops. The **tstart** parameter specifies the point in time from which the solution is to be saved. If this parameter is not given, the solution is saved from time equal to zero. The **tmax** parameter is the maximum time step to be taken. If it is not given, the maximum is calculated based upon the value of the stop time. This ratio is set at compilation and has a default value of 0.1.

.tran 0.1n 25n

.tran 0.1n 25n 0.0 5n

Harmonic Balance Simulation

.hb f1 order [f2] [f3] [f4] . . .
.hbdc Sname start end step [Sname start end step] . . .
.hbac Sname start end step [Sname start end step] . . .
.hbfr Sname start end step fnum [Sname start end step] . . .
.hbss nstep {a | d | f} Sname start end {fnum} [{a | d | f}
Sname . . .]

These analysis cards are used to specify various types of harmonic balance analysis. The **.hb** card is required and describes the Fourier expansion. The **f1** parameter specifies the first fundamental frequency and is required. The subsequent **f#** parameters specify the higher fundamental frequencies if inter-modulation distortion analysis is to be performed. The **order** parameter is the number of harmonics used in the Fourier expansion.

The **.hbdc**, **.hbac**, **.hbfr**, and **.hbss** card specifies any sweeping of sources. The **.hbdc** causes the HB source **Sname** to have its DC bias swept from the **start** value to the **end** value in **step** steps. The sources listed later on the card are swept inside the sources listed first.

The **.hbac** causes the HB source **Sname** to have its large signal AC value swept in magnitude from **start** value to **end** value in **step** steps. The sources listed later on the card are swept inside the sources listed first.

The **.hbfr** card causes the HB source **Sname** to have it frequency value swept from **start** value to **end** value in **step** steps. In addition, the fundament frequency number **fnum** is adjusted to this value as well.

The **.hbss** allows for simultaneous sweeps of multiple sources such that the specified value is adjusted for each and every source. Therefore, only one value is specified for the number of steps, **nstep**. Each source **Sname** has either its AC magnitude, DC value, or frequency value (**a** | **d** | **f**) swept from **start** value to **end** value with the same **nstep** steps. If frequency is swept, then the fundamental value is adjusted as given by **fnum**.

Multiple cards may be contained in the same simulation. For such situations, the sources on **.hbss** are swept inside the source on **.hbfr** which are swept inside the sources on the **.hbac** card, which are swept inside the sources on the **.hbdc** card.

```
.hb 1.2G 8
.hbac Vin 0.25 4.0 0.25
```

Sweep source Vin from 0.25 to 4.0 by 0.25. Do a HB simulation at 1.2 GHz with 8 harmonics.

```
.hb 849.5Meg 5 850.5Meg
.hbss 16 Vin1 a 0.25 4.0 Vin2 a 0.25 4.0
```

Sweep the magnitude of the AC voltage of Vin1 and Vin2 from 0.25 to 4.0 volts in 16 steps. Hence, during each step Vin1 and Vin2 take on the same value. This sweep description is designed to do intermodulation distortion analysis by setting the frequency of Vin1 to 849.5MHz and that of Vin2 to 850.5MHz. The **.hb** card specifies a 5th order expansion at the two given frequencies.

C.4 Node Numbers

The node numbers in the netlist must be unique positive integers. Negative numbers and alpha-numeric characters will cause an error.

C.5 Scaling Factors

The values in the ne list may be specified in decimal notation (e.g., 1.2), exponential notation (4.3e-5, 10e4) or using a scaling factor (1k). The scaling factors are defined as follows:

Unit	Symbol	Scaling
tera	t	10^{12}
giga	g	10^{9}
mega	meg	10^{6}
kilo	k	10^{3}
centi	c	10^{-2}
milli	m	10^{-3}
micro	u	10^{-6}
nano	n	10^{-9}
pico	p	10^{-12}
femto	f	10^{-15}
atto	a	10^{-18}

Appendix D: Examples from PISCES2-HB Manual

D.1 Description

This chapter provides many examples of using the new features provided in PISCES-2HB. The first section focuses on simulations with external boundary conditions by showing a variety of different analysis methods. The second section gives a number of examples that take advantage of the harmonic balance module to find large signal responses of some common RF circuits including a demodulator, mixer, and power amplifier.

D.2 Circuit Boundary Conditions

D.2.1 Diode with External Circuit Components and Distributed Contact Resistance

PISCES boundary conditions include distributed contact resistances, limited surface recombination, external resistances/capacitances, and linear circuit boundary conditions. This example uses two of these boundary conditions - contact resistance and circuit boundary condition - simultaneously and to plot the I-V characteristics of a diode with external resistances.

A diode has external resistors on each electrode and a feedback resistor between the two electrodes. In addition, electrode two has a distributed contact resistance as shown in Figure D.1 Next to the circuit diagram, the IV characteristics are given for the configuration. Note the large leakage current flowing through the shunt resistance.

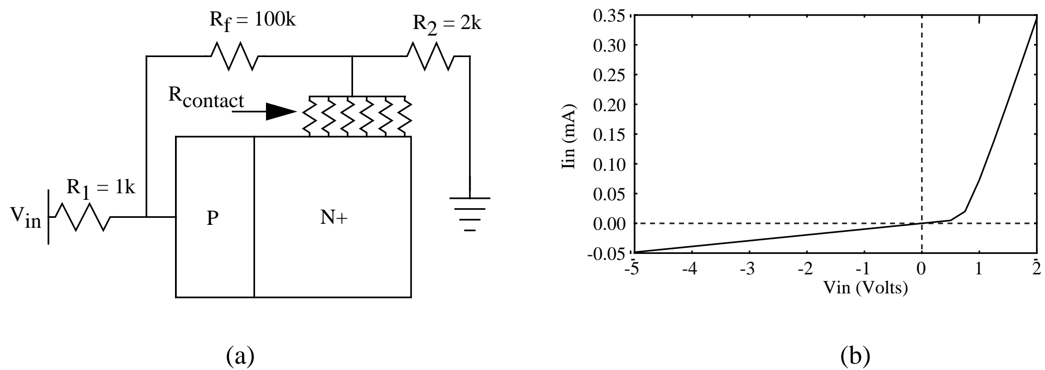


Figure D.1: (a) Circuit diagram for diode with extrinsic bulk resistance and distributed contact resistance. (b) The IV response of the structure shows a large leakage current generated by the shunt resistance.

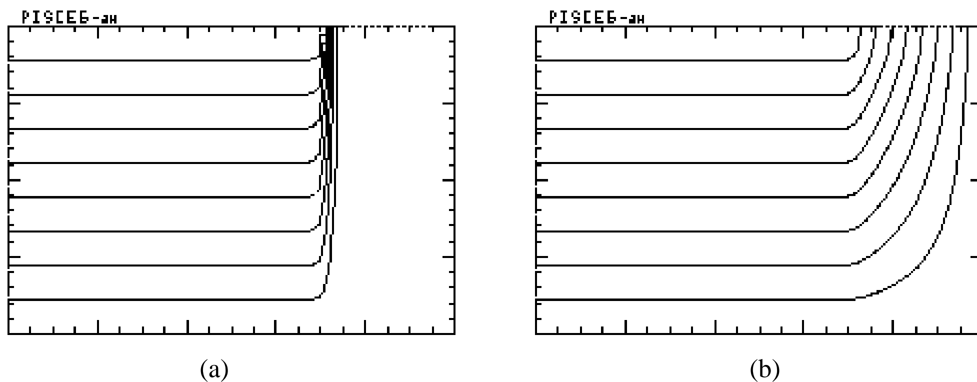


Figure D.2: Current flow lines for a diode with (a) a low contact resistance and (b) with a large contact resistance.

The input deck for PISCES and circuit description is provided in Figure D.3a and Figure D.3b, respectively. The distributed contact resistance is specified on the first **contact** card and the circuit boundary conditions are invoked by specifying a **ckfile** on the second **contact** card. The circuit description is given in the circuit file named diode.ckt. The solution for the circuit simulation is stored in diode.raw and is in a format that can be read by Berkely's version of SPICE3.

Figure D.2 contains plots of the current flow lines when the distributed contact resistance is low and high relative to the N- region, respectively. Notice that the current flow changes based upon the magnitude of the distributed resistance.

D.2.2 Transient Response of a BJT Inverter

Transient analysis allows for automatic time stepping to minimize the number of time steps that are required. To demonstrate this capability, a simple BJT inverter is switched from its on state to its off state and back to its on state. A circuit diagram and the transient response is provided in Figure D.4.

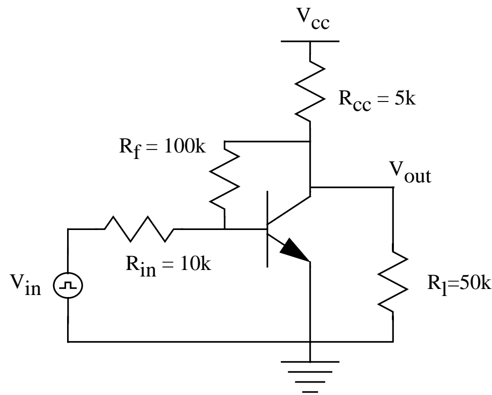
```
title diode with contact resistance and circuit BC's

$ create mesh
mesh rect nx=70 ny=11

x.mesh n=1 l=0
x.mesh n=70 l=10
y.mesh n=1 l=0
y.mesh n=11 l=0.2

region num=1 ix.lo=1 ix.hi=70 iy.lo=1 iy.hi=11 silicon

$ electrodes
electr num=1 ix.lo=1 ix.hi=1 iy.lo=1 iy.hi=11
electr num=2 ix.lo=50 ix.hi=70 iy.lo=1 iy.hi=1

$ uniform doping for simplicity
doping p.type conc=1e15 uniform
doping n.type conc=2e19 uniform x.left=4

$ place distribute resistance on contact #2
contact num=2 con.res=1e-3'

$ specify circuit file
contact cktfile=diode.ckt

$ set models
model consrh conmob fldmob bgn

$ do initial solution
symb carr=2 newton
method trap
solve init

$ ramp up voltage on diode
solve v1=0 vstep=-1.0 nstep=5 elect=1

$ set circuit logfile in Berkeley spice3f4 format
log cktfile=diode.raw spice.out

$ do dc sweep
solve dcckt
```

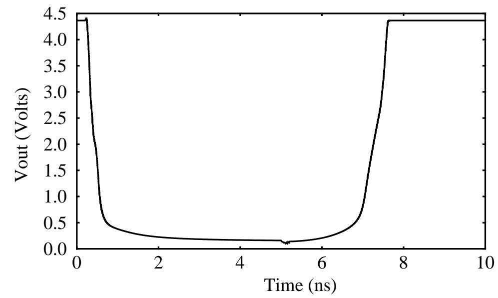
(a)

```
* test for multiple boundary conditions

* initial start value for source
vin 1 0 -5

* completely surround device with resistances
r1 1 3 1k
rf 3 4 100k
r2 4 0 2k

* device specification
nd 3 4

* analysis specification
.dc vin -5 2 0.25

* circuit options
.options area=100 rmin=0 gmin=0
```

(b)

Figure D.3: (a) PISCES input deck for DC sweep of diode surrounded by resistive network with a distributed contact resistance. (b) Netlist describing circuit diagram.





(a) (b)

Figure D.4: Circuit diagram of a simplified BJT inverter and the response of that inverter given a pulse input.

```
$ transient analysis of bjt inverter

$ load mesh file
mesh infile=bjt.msh

$ set-up circuit information
contact cktfile=inv.ckt

$ set models for BJT inverter
model srh auger bgn conmob fldmob

$ do symbolic factorization and
$ set numerical methods
symb newton carrier=2
method itlimit=20 biaspart 2nd tauto

$ initial device solution
solve init

$ ramp up voltage on drain
solve v1=0 vstep=1 nstep=5 elect=1

$ solution logfile
log cktfile=inv.raw spice.out

$ do transient analysis of circuit
solve tranckt
```

(a)

```
* Circuit description for a BJT inverter

* apply transient piece-wise linear function at
*   input of the inverter
vin 4 0 pwl 0 0 0.2n 0 0.4n 5 5n 5 5.2n 0 1 0
vcc 1 0 dc 5.0

* surrounding circuitry
rcc 1 2 5k
rbb 4 3 10k
r31 2 3 100k

* load of next stage
rl 2 0 50k

* BJT device
nq1 2 3 0

* set device size
.options area=1

* transient analysis initial step and stop time
.tran 0.1n 10n
```

(b)

Figure D.5: (a) PISCES input deck for transient simulation of a BJT inverter. (b) Netlist describing circuit diagram.

The PISCES input deck and the circuit file are shown in Figure D.5. The time stepping algorithm is selected in the PISCES deck on the **method** card with the **2nd** and **tauto**

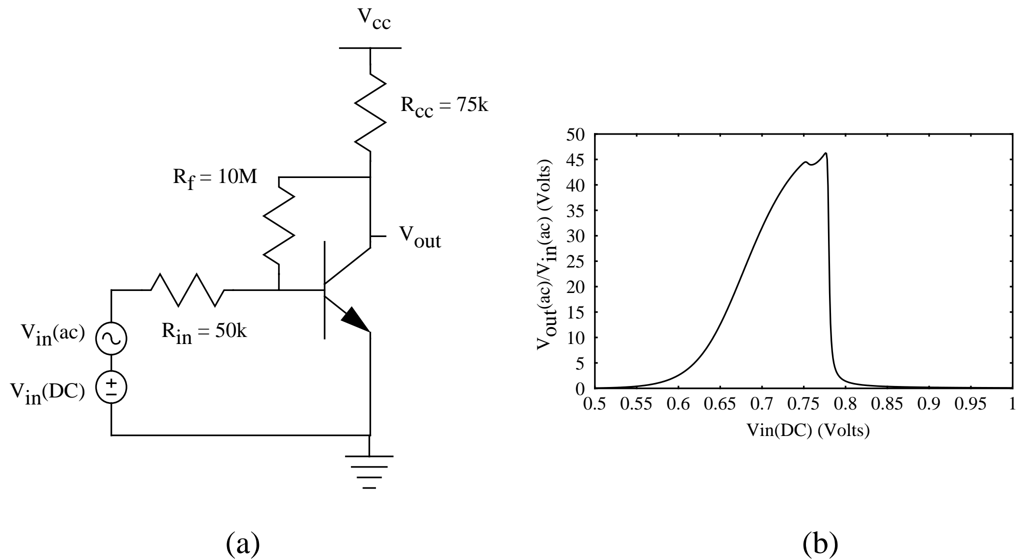


Figure D.6: (a) Circuit diagram for a BJT amplifier with (b) the small signal response for different bias conditions on the gate.

parameter set to true to invoke the TR-BDF method described by Bank. The circuit portion of the code calculates its own time step independent of PISCES. PISCES then decides which time should be taken: its own or that of the circuit where typically the smaller of the two is selected. In addition, PISCES handles break points in the input response by forcing a time point at that specific location.

D.2.3 Gain vs. Bias Condition for a BJT Amplifier

In this example, a BJT amplifier is analyzed for its small signal gain at various bias conditions. Hence, the simulation requires both a DC sweep and an AC small signal perturbation. This dual simulation capability allows a user to analyze how the small signal performance of a device changes with variations in the bias condition.

Figure D.6 shows the circuit diagram for the BJT amplifier and the small signal gain versus the base bias. V_{in}(DC) is swept from 0.5V to 1.0V in tightly space voltage steps while V_{in}(ac) is applied at a relatively low frequency in order to negate high frequency effects.

The PISCES input deck and the circuit file are given in Figure D.7. In the PISCES deck, the parameter to specify a joint ac/DC analysis is **acdcckt** on the **solve** card. In the circuit file, the circuit configuration is described in a SPICE-like format and an **.ac** card and a **.dc** card specifies the limits on the two analyses. The **.ac** card specifies the same start and end

```
$ find bias for maximum gain in amplifier

$ load mesh generated by bjt.pis
mesh infile=bjt.msh

$ load circuit BC's
contact cktfil=amp.ckt

$ set models
model srh auger bgn conmob fldmob

$ do init solution
symb newton carrier=2
method itlimit=20 trap
solve init
solve v1=0.0 vstep=1.0 nstep=5 elect=1

$ solve for small signal at multiple DC biases
log cktfile=amp.log
solve acdcckt
```

(a)

```
* test for analysis of inverter

* initial biases
vin 12 0 dc 0.5 ac 1.0
vcc 11 0 dc 5.0

* bias circuitry
rin 12 2 50k
rcc 11 1 75k
rf 1 2 10Meg

* numerical device
nq1 1 2 0

* set circuit options
.options area=200 rmin=1e-12 gmin=1e-12

* DC and ac sweep
.dc vin 0.5 1.0 0.005
.ac lin 1 1 1
```

(b)

Figure D.7: (a) PISCES input deck for determining small signal gain versus bias condition of BJT amplifier. (b) Netlist for circuit and analysis limits.

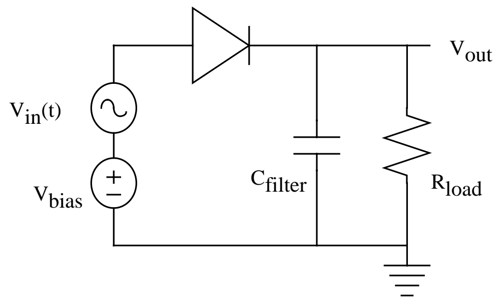
frequency with only a single step. When this is encountered on the **.ac** card, the small signal simulation is performed only at that one frequency. Single frequency AC analysis is used because it is the only one desired for this example. The user does have the option to do multiple frequency simulations in an ac/DC simulation.

D.3 Harmonic Balance Analysis

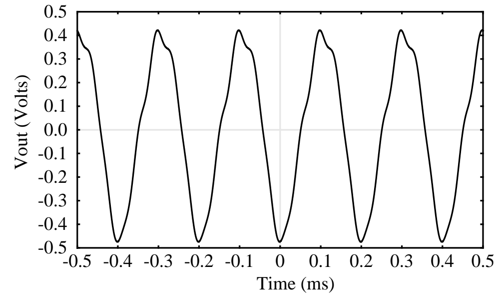
Harmonic balance analysis provides a technique to do large signal sinusoidal analysis in the frequency domain rather that the time domain. The next three examples demonstrate the power of this capability by solving for the large signal response of an AM demodulator, a BJT mixer, and MOS power amplifier. All three involve some simple circuitry in order to obtain the desired response.

D.3.1 Large Signal Analysis of an AM Demodulator

A simple AM demodulator, consisting of a diode, resistor, and capacitor, is fed a 5KHz signal on a 15MHz carrier as shown in the circuit diagram in Figure D.8 (a). The diode rectifies the amplitude modulated input voltage F(t). The output capacitor and resistor, acting as a low pass filter, smooths out the resultant signal. The output, less the DC offset, is shown in Figure D.8 (b). Note that the output signal is not a pure sine wave and some improvements are needed in the circuit and device in order to optimize the results.



(a)



(b)

$$V_{in}(t) = \left(1 + \frac{1}{2}\sin(2\pi(5x10^{3})t)\right)\sin(2\pi(5x10^{6})t)$$

Figure D.8: Circuit diagram of an AM demodulator. A 5KHz signal on a 5MHz carrier is fed into the demodulator.

```
$ PISCES simulation deck for am demodulator

$ load mesh generated by diode.pis
mesh in=diode.msh

$ tell PISCES about circuit description
contact cktfile=am.ckt

$ do symbolic factorization
symb newton carr=2

$ set numerical methods for PISCES and HB
method biaspart
hbmeth gmrestart=10 maxiter=40

$ set simulation models
models temp=300 srh auger conmob fldmob

$ do initial solution and operating point solution
solve init
solve v1=0.0 vstep=0.2 nstep=3 elect=1
solve opckt

$ set circuit solution file and do HB
$   analysis to find demodulated signal
log column.out cktfile=am.log
solve hbckt
```

(a)

```
* a simple AM demodulator

* 15MHz carrier plus 5kHz signal
Vcar 1 0 hb 0.65 0.0 15Meg 0
Vsigm 2 1 hb 0.0 0.25 14.995Meg 90
Vsigp 3 2 hb 0.0 -0.25 15.005Meg 90

* rectifier
Ndiode 3 4

* load resistance and filtering capacitance
Cfilter 4 0 0.013u
rload 4 0 5k

* frequencies of interest
.hb 15Meg 5 5k

* ramp up ac carrier signal magnitude for
* improved convergence
.hbac Vcar 0.0 1.0 0.1

* circuit options
.options gmin=0 rmin=0 area=100
```

(b)

Figure D.9: (a) PISCES input deck for analyzing an AM demodulator. (b) Netlist describing circuit diagram and analysis limits.

The PISCES input deck and the circuit file are shown in Figure D.9. The interesting aspect of this simulation is how the input signal is represented. In describing harmonic balance sources only a single frequency is specified; hence, the expression for $F(t)$ must be reduced

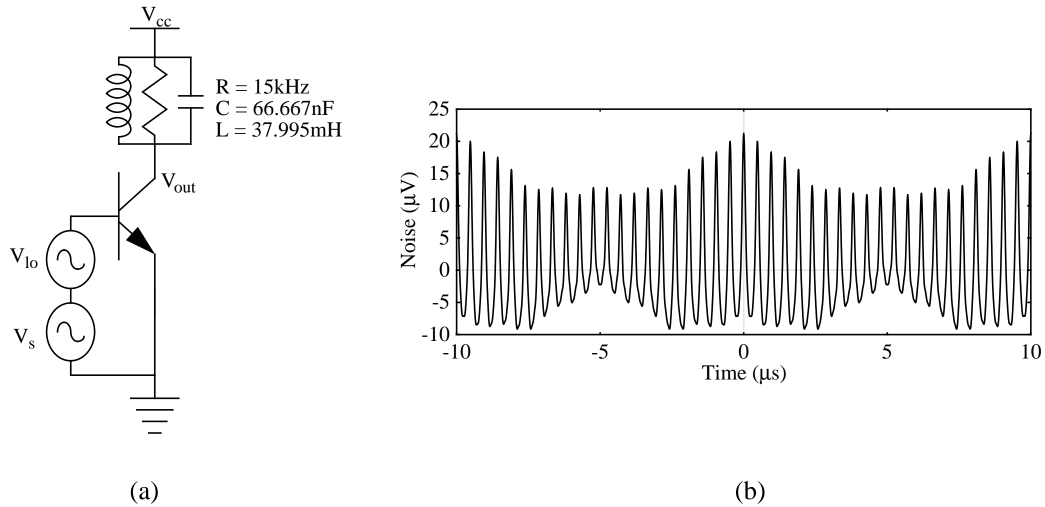


Figure D.10: (a) A BJT Mixer that down converts a 200MHz to a 100KHz signal.
(b) The noise in the 100kHz output signal.

to single sinusoids. This reduction is easily accomplished using trigonometric identities in order to generate Equation D.1.

$$F(t) = \sin(\omega_c t) + 0.25(\cos((\omega_c - \omega_s)t) - \cos((\omega_c + \omega_s)t)) \quad \text{(D.1)}$$

where ω_c is the carrier frequency and ω_s is the signal frequency.

Another issue of concern is the convergence. Harmonic balance convergence is based upon the frequency of the simulation and the amount of power or amplitude in the input signals. When either becomes large, convergence may become difficult. In order to facilitate improved convergence, the carrier frequency is ramped up from 0.0V to the 1.0V desired in the simulation by including the **.hbac** card in the circuit netlist. The solutions for the intermediate steps in ramping the carrier frequency are included in the output file, but those results are easily dismissed during analysis.

D.3.2 BJT Mixer to Down Covert a 2MHz Signal to 100KHz

This example consists of a simple, single-stage BJT mixer using an HP25 transistor from Hewlett Packard as shown in Figure D.10. The input to the circuit is driven by a 2.1MHz LO and a 2.0MHz RF signal, and mixes the signal down to 100KHz. A tuned resonant *RLC* circuit with a high Q in the collector filters out mixing products other than 100KHz. The noise in the output, also shown in Figure D.10, is in the micro-voltage range thus providing

```
title HP25 w/ 3-emitter, 2-base, 1-collector

$ load mesh for structure

$ set finite recombination on emitter
contact num=3 surf.rec vsurfn=1e5 vsurfp=1e5
contact cktfil=hp25mix.ckt

$ do symbolic factorization, set numerical meth-
ods and models
symbol carrier=2 newton
method itlim=50 p.tol=1e-7 c.tol=1e-7
models temp=300 bgn srh auger conmob fldmob

$ initial solve
solve init

$ sweep v1 (collector) up to 10 volts
solve v1=1 nstep=9 elect=1 vstep=1

$ sweep V2 (base) up to 0.7 volts
solve v2=0.1 nstep=5 elect=2 vstep=0.1

$ specify output file and do HB analysis
log cktfile=hp25mix.log column.out
solve hbckt
```

(a)

```
* Single-device BJT mixer based on the HP-25
* Q = 100, LO v=0.15

* bias voltages
Vcc 4 0 10.0
Vlo 1 0 hb 0.7 0.15 2.1Meg 0

* input signal
Vs 2 1 hb 0.0 0.01 2.0Meg 0

* resonant RLC circuit
L1 4 3 37.995e-6
C1 4 3 66.667e-9
R1 4 3 15k

* HP25 bjt
Mbyte 3 2 0

* harmonic balance limits
.hb 2.1Meg   6   0.1Meg

* circuit issues
.options gmin=1e-12 area=5 rmin=1e-12
```

(b)

Figure D.11: (a) PISCES input deck for analyzing the BJT Mixer. (b) Netlist describing circuit diagram and analysis limits.

a very clear output signal. This distortion can be reduced by increasing the Q of the resonator and reducing the RF power level.

The PISCES input file and circuit file are shown in Figure D.11. In this example, it is very important to ramp up the DC bias on the mixer. This biasing is done in the PISCES file by first increasing the voltage on the collector and then the voltage on the base. Upon obtaining the DC bias the harmonic balance simulation can then be executed with better convergence behavior.

D.3.3 Analyzing Gain and Efficiency in an MOS Power Amplifier

The modeling of an RF device coupled with Parasitics, matching network, and bias network can provide insight into the performance of a discrete power amplifier. Figure D.12: shows the intrinsic LDMOS device with Parasitics at the chip level due to the pad and interconnect. In addition to the parasitics shown, there are additional parasitics for the packaging. The input and output of the amplifier is connected through matching networks that are tuned for the individual device and isolated from the DC bias by a capacitor. A

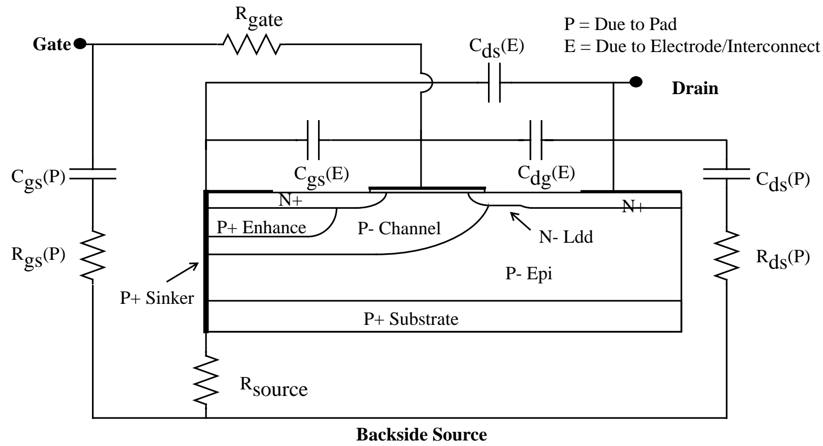


Figure D.12: Cross section of an LDMOS transistor with parasitics represented by lumped elements.

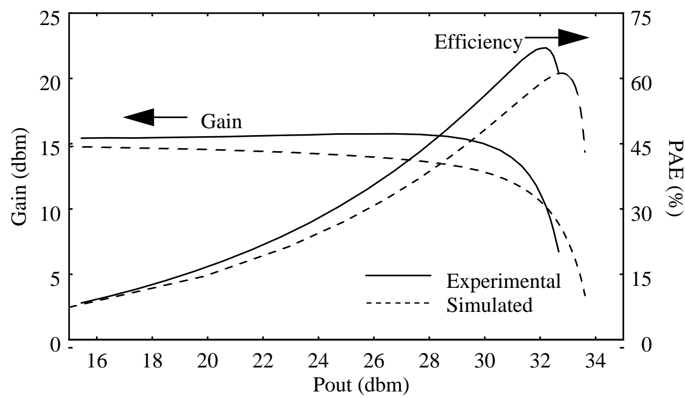


Figure D.13: Gain and power added efficiency (PAE) for LDMOS power amplifier.

biasing network sets the operating point and is isolated from the input by an inductance. A PISCES deck and netlist for the device and circuit is created in a similar manner to the previous examples and thus not provided here.

The physics of the device plays an important role in its performance. A laterally diffused graded channel enhances RF performance, prevents punch-through, and increases the device transconductance. A p+ sinker (represented by a side contact electrode) connects the source and substrate together to eliminate extra bonding wires and provide for a back side contact. An n- LDD decreases the electric field at the drain side of the channel and optimizes $R_{ds}(on)$, BV_{dss}, and C_{dg}. A metal field plate reduces the electric fields at the edge of the gate, thereby increasing the breakdown voltage and reducing C_{dg}.

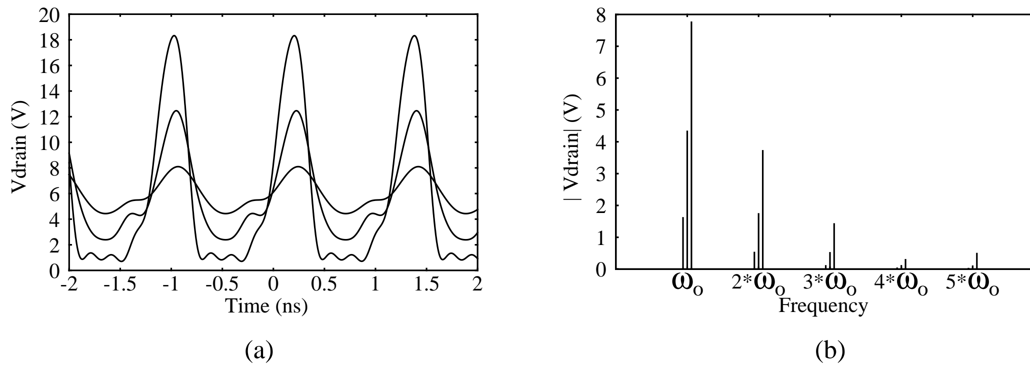


Figure D.14: (a) Time domain voltage and (b) spectrum of the voltage on the drain of the LDMOS transistor in a power amplifier as the device enters g_m compression.

The response of interest is the gain and efficiency as shown in Figure D.13:. The gain rolls off at higher power levels ($P_{in} > 30$ dBm) because the device operates in g_m compression region and the output power is limited by the saturation current. The power added efficiency (PAE) is low for a small P_{out} because the device drains more power in Class A operation. Efficiency increases until just after the gain starts to decrease. At this point, P_{in} approaches P_{out} resulting in very little power added to the input signal.

In addition to examining a calculated result of the simulation, one can analyze the state of the device at various levels in the simulation. For example, the device performance begins to degrade when it enters g_m compression. Figure D.14: shows the voltage on the drain in the time domain and frequency domain as the device enters this region. The three power levels correspond to an output of 19.9, 28.5, and 33.5 dBm. At the low power level the voltage on the drain is able to swing to its full limits. As the power increases, the voltage becomes limited on how low it can swing. The time domain plot and the frequency plot reflect that change by two different mechanisms. The time domain signal flattens out when it swings low. The spectrum reflects this effect with the addition of harmonics at the higher frequencies.

This example clearly shows the power of the harmonic balance device simulation. The physical characteristics of an advanced technology impact simulation significantly. Hence, this tool allows the design engineer to analyze and optimize a robust devices in order to achieve the maximum performance.

References

[1] R. Dutton and Z. Yu. *Technology CAD: Computer Simulation for IC Processes and Devices*. Boston: Kluwer Academic Publishers, 1993.

[2] R. Langevelde and F. Klaassen. “Effect of Gate-Field Dependent Mobility Degradation on Distortion Analysis in MOSFET’s.” *IEEE Transactions on Electron Devices*.Vol. 44, No 11, Nov. 1997. pp 2044-2052.

[3] Y. Tsividis and K. Suyama. “MOSFET Modeling for Analog Circuit CAD: Problems and Prospects.” *Journal of Solid-State Circuits*. Vol. 29, No. 3, March 1994. pp 210-216.

[4] D. Root. “Measurement-Based Mathematical Active Device Modeling for High Frequency Circuit Simulation.” *IEICE Transactions on Electronics*. Vol. E82-C, No. 6, June 1999. pp 924-936.

[5] D. Root, S. Fan, and J. Meyer. “Technology Independent Large Signal FET Non Quasi-Static FET Models by Direct Construction from Automatically Characterized Device Data.” *Proceedings of European Microwave Conference*. Sept. 1991. pp 927-932.

[6] S. Wolf. *Silicon Processing for the VLSI Era. Volume 3: The Sub-micron MOSFET*. Sunset Beach, CA: Lattice Press, 1995.

[7] S. Xu, P. Foo, J. Wen, Y. Liu, F. Lin, and C. Ren. “LDMOS with Extreme Low Parasitic Feedback Capacitance and High Hot-Carrier Immunity.” *Proceedings of the IEDM*. Dec. 1999. pp 201-204.

[8] *BSIM3 Version 3.0 Manual*. The Regents of the University of California. Berkeley, CA, 1995.

[9] W. Engl, R. Laur, H. Dirks. "MEDUSA - A Simulator for Modular Circuits." *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*. Vol. CAD-1, No. 2., April 1982. pp 85-93.

[10] Rollins, Gregory J. and John Choma. "Mixed-Mode PISCES-SPICE Coupled Circuit and Device Solver." *IEEE Transactions on Computer Aided Design*. Vol. 7, No. 8, Aug. 1988. pp 862-867.

[11] *Medici User's Manual*. Avant! Corp. Milpitas, CA.

[12] *Atlas Users Manual*. Silvaco International. Santa Clara, CA.

[13] G. Rollins. S*ingle Event Upset in Silicon Integrated Circuits*. Ph.D. Dissertation: University of Souther California. Aug. 1989.

[14] K. Mayaram and D. Pederson. "Coupling Algorithms for Mixed-Level Circuit and Device Simulation." *IEEE Transactions on Computer Aided Design*. Vol. 11, No 8. Aug. 1992. pp 1003-1012.

[15] D. Gates. *Design-Oriented Mixed-Level Circuit and Device Simulation*. Ph.D. Dissertation: University of California at Berkeley. June 1993.

[16] A. Lee and R. Dutton. "Parasitic Extraction Based on SWR Framework Tools." *Proceedings of the VLSI Multilevel Interconnection Conference*. June 1994. pp 515-517.

[17] B. Troyanovsky, Z. Yu, and R. Dutton. "Large Signal Frequency Domain Device Analysis Via the Harmonic Balance Technique." *Proceedings of SISDEP*. Sept. 1995. pp. 114-117.

[18] M. Pinto, C. Rafferty, H. Yeager, and R. Dutton. *PISCES-II - Poisson and Continuity Equation Solver*. Stanford Electronics Laboratory Report. Stanford University. Sept. 1984.

[19] F. Rotella, Z. Yu, R. Dutton. "Mixed-Mode Device/Circuit Simulator." *Part III: Pisces 2ET with Applications Documentation*. Board of Trustees of Leland Stanford Junior University. Stanford, CA. Dec. 1994.

[20] F. Rotella, B. Troyanovsky, Z. Yu, and R. Dutton. *Supplemental Manual for PISCES-2HB*. Board of Trustees of Leland Stanford Junior University. Stanford, CA. February 1999.

[21] Z. Yu, R. Dutton, and H. Wang. "A Modularized, Mixed IC Device/Circuit Simulation System." *Proceedings of the Synthesis and Simulation Meeting and International Exchange*. April 1992. pp 444-448.

[22] W. McCalla. *Fundamentals of Computer-Aided Circuit Simulation.* Boston: Kluwer Academic Publishers, 1993.

[23] L. Nagel. *SPICE2: A Computer Program to Simulate Semiconductor Circuits*. Ph.D. Dissertation: University of California at Berkeley. May 1975.

[24] S. Selberherr. *Analysis and Simulation of Semiconductor Devices*. New York: Springer-Verlag, 1984.

[25] M. Pinto. *Comprehensive Semiconductor Device Simulation for Silicon ULSI*. Ph.D. Dissertation: Stanford University. August 1990.

[26] G. Baccarani, M. Rudan, R. Guerrieri, and P. Ciampolini. "Physical Models for Numerical Device Simulation." *Process and Device Modeling*. Editor: W. Engl. Amsterdam: Elsevier Science Publishers, 1986. pp 107-158.

[27] A. Schenk. *Advanced Physical Models for Silicon Device Simulation*. New York: Springer-Verlag, 1998.

[28] S. Laux. "Techniques for Small Signal Analysis of Semiconductor Devices." *IEEE Transactions on Electronic Devices*. Vol. ED-32, No. 10, pp. 2028-2037, Oct. 1986.

[29] S. Furkay, J. Johnson, G. Fiorenza, and R. Logan. *Fielday II User's Manual.* Ver. 7.5.0. IBM Microelectronics, 1995.

[30] J. Johnson, S. Furkay, and D. Cole. *Fielday II Theoretical Manual.* Ver. 7.1.0. IBM Microelectronics, 1995.

[31] B. Johnson, T. Quarles, A, Newton, D. Pederson, and A. Sangiovanni-Vincentelli. *SPICE3 Version 3f User's Manual*. The Regents of the University of California. Berkeley, 1992.

[32] T. Quarles. *Adding Devices to SPICE3*. Memorandum No. UCB/ERL M89/45. Berkeley: University of California, 1989.

[33] T. Quarles. *SPICE3 Implementation Guide*. Memorandum No. UCB/ERL M89/44. Berkeley: University of California, 1989.

[34] B. Kernighan and R. Pike. *The Unix Programming Environment*. Englewood Cliffs: Prentice Hall, 1984

[35] K. Haviland and B. Salama. *UNIX System Programming*. Reading: Addison-Wesley, 1987.

[36] A. Geist, A. Beguelin, J. Dongarra, W. Jiang, R. Manchek, and V. Sunderam. *PVM 3 User's Guide and Reference Manual*. Oak Ridge: Oak Ridge National Laboratory, 1993.

[37] W. Gropp, E. Lusk, and A. Skjellum. *Using MPI. Portable Parallel Programming with the Message-Passing Interface*. Cambridge: The MIT Press, 1995.

[38] D. Gates, P. Ko, and D. Pederson. "Mixed-Level Circuit and Device Simulation on a Distributed-Memory Multi-computer." *Proceeding of Custom Integrated Circuits Conference*. May 1993. pp 8.5.1-8.5.4.

[39] G. Massobrio and P. Antognetti. *Semiconductor Device Modeling with SPICE*. New York: McGraw-Hill, 1988.

[40] F. Rotella, B. Troyanovsky, Z. Yu, R. Dutton, and G. Ma. "Harmonic Balance Device Analysis of an LDMOS RF Power Amplifier with Parasitics and Matching Network." *Proceedings of SISPAD*. Boston: Sept. 97.

[41] B. Troyanovsky. *Frequency Domain Algorithms for Simulating Large Signal Distortion in Semiconductor Devices*. Ph.D. Dissertation: Stanford University. November 1997.

[42] K. Kundert, J. White, and A. Sangiovanni-Vincentelli. *Steady State Methods for Simulating Analog and Microwave Circuits*. Boston: Kluwer Academic Publishers, 1990.

[43] S. Sze. *Physics of Semiconductor Devices*. New York: John Wiley and Sons. 1981.

[44] J. Andrews and M. Lepselter. "Reverse Current-Voltage Characteristics of Metal-Silicide Schottky Diodes." *Solid State Electronics*. Vol. 13 No. 7, July 1970. pp. 1011-1023.

[45] C. Crowell and S. Sze. "Current Transport in Metal-Semiconductor Barriers." *Solid State Electronics*. Vol. 9, Nos. 11/12, Nov./Dec. 1966. pp 1035-1048.

[46] C. Ho, A. Ruehli, and P. Brennan. "The Modified nodal approach to network analysis and design." *IEEE Transactions on Circuits and Systems*. Vol. CAS-22, No. 6, June 1975. pp 504-509.

[47] T. Quarles. *Analysis of Performance and Convergence Issues for Circuit Simulation*. Memorandum No. UCB/ERL M89/42. University of California, Berkeley. 1989.

[48] R. Bank, W. Coughran, W. Fichtner, E. Grosse, D. Rose, and R. Smith, "Transient simulation of silicon devices and circuits." *IEEE Transactions on Electron Devices*. Vol. ED-32, No. 10, Oct. 1986. pp. 1992-2007.

[49] B. Troyanovsky, F. Rotella, Z. Yu, R. Dutton, and J. Sato-Iwanga. "Large Signal Analysis of RF/Microwave Devices with Parasitics Using Harmonic Balance Device Simulation." *Proceedings of Synthesis and System Integration of Mixed Technologies*. Nov. 96.

[50] B. Troyanovsky, Z. Yu, L. So, and R. Dutton. "Relaxation-Based Harmonic Balance Technique for Semiconductor Device Simulation." *Proceedings of International Conference on Computer Aided Design*. Nov. 1995. pp 700-703.

[51] S. Pearton, M. Iannuzzi, C. Reynolds, and L. Peticolas. "Formation of Thermally Stable High-Resistivity AlGaAs by Oxygen Implantation." *Applied Physics Letters*. Vol 52, No. 5, Feb. 1988. pp 38-39.

[52] W. Shockley and W. Read. "Statistics of the Recombinations of Holes and Electrons." *Physical Review.* Vol. 87, No 5, Sept. 1952. pp 835-842.

[53] R. Hall. "Electron-Hole Recombination in Germanium." *Physical Review.* Vol. 87, No. 2, July 1952. pp 387.

[54] J. Fossum and D. Lee. "A Physical Model for the Dependence of Carrier Lifetime on Doping Density in Non-degenerate Silicon." *Solid-State Electronics*. Vol 25, No. 8, Aug. 1982. pp. 741-747.

[55] D. McCumber. "Einstein Relations Connecting Broadband Emission and Absorption Spectra." *Physical Review.* Vol. 136, No. 4A, Nov. 1964. pp 954-957,

[56] M. Shur. *GaAs Devices and Circuits*. New York: Plenum Press, 1987.

[57] R. Williams. *Modern GaAs Processing Methods*. Boston: Artech House, 1990.

[58] *Low-Cost Fiber-Optic Links for Digital Applications up to 150Mbd*. Application Bulletin 78. Hewlett Packard Corporation.

[59] *Low-Cost Fiber-Optic Transmitter and Receiver Interface Circuits*. Application Bulletin 73. Hewlett Packard Corporation.

[60] R. Dutton, F. Rotella, Z. Sahul, L. So, and Z. Yu. "Integrated TCAD for OEIC Applications." *Proceeding of the International Symposium on Optoelectronics for Information and Microwave Systems*. Jan. 1994. pp 80-89.

[61] K. Wu, Z. Yu, L. So, R. Dutton, and J. Sato-Iwanga. "Robust and Efficient ac Analysis of High-speed Devices." *Proceedings of the International Electron Devices Meeting*. Dec. 1992. pp 935-938.

[62] K. Wang, H. Park, Z. Yu, E. Kan, and R. Dutton. "3D Solid Modeling of IC Structures Using Simulated Surface Topography." *Proceedings of SISPAD*. Sept. 96. pp 131-132.

[63] K. Nabors and J. White, "FASTCAP: A Multi-pole Accelerated 3-D Capacitance Extraction Program." *IEEE Transactions on Computer Aided Design of Integrated Circuits and Systems*. Vol. 10, No. 11, Nov. 1991. pp 1447-1459.

[64] M. Kamon, M. Tsuk and J. White. "FASTHENRY: A Multi-pole Accelerated 3D Inductance Extraction Program." *IEEE Transactions on Microwave Theory and Techniques*.Vol 42, No. 9, Sept. 1994. pp 1750-1758.

[65] Stanley Wolf. *Silicon Processing for the VLSI Era. Volume 2: Process Integration*. Sunset Beach: Lattice Press, 1986.

[66] C-Y Lu, J. Sung, H. Kirsch, S. Hillenius, T. Smith, and L. Manchanda. "Anomalous C-V Characteristics of Implanted Poly MOS Structure in n+/p+ Dual Gate CMOS Technology." *IEEE Electron Device Letters*. Vol. 10, No. 5, May 1989. pp 192-194.

[67] R. Habas and S. Selberherr. "On the Effect of Non-Degenerate Doping of Polysilicon Gate in Thin Oxide MOS-Devices - Analytic Modeling." *Solid State Electronics*.Vol. 33, No. 12, 1990. pp 1539-1544.

[68] N. Arora, R. Rios, and C-L Huang. "Modeling the Polysilicon Depletion Effect and its Impact on Sub-micrometer CMOS Circuit Performance." *IEEE Transactions on Electron Devices*. Vol 42, No. 5, May 1995. pp 935 - 943.

[69] P. Habas and J. V. Faricelli. "Investigation of the Physical Modeling of the Gate Depletion Effect." *IEEE Transactions on Electron Devices*. Vol. 39, No. 6, June 1992. pp 1496 - 1500.

[70] S-W Lee, C. Liang, C-S Pan, W. Lin, and J. B. Mark. "A Study of the Physical Mechanism in the Recovery of Gate Capacitance to Cox in Implanted Polysilicon MOS Structures." *IEEE Transactions on Electron Devices*. Vol. 13, No. 1, Jan. 1992. pp 2 - 4.

[71] R. Rios, N. Arora, and C-L Huang. "An Analytic Polysilicon Depletion Effect Model for MOSFET's." *IEEE Electron Device Letters*. Vol. 15, No. 4, April 1994. pp 129 - 131.

[72] C-L Huang and N. Arora. “Measurement and Modeling of MOSFET I-V Characteristics with Polysilicon Depletion Effect.” *IEEE Transactions on Electron Devices*. Vol. 40, No. 12, December 1993.

[73] N. Arora, R. Rios, C.-L. Huang, and K. Raol. “PCIM: A Continuous Physically Based Short-Channel MOSFET Model for Circuit Simulation.” *IEEE Transactions on Electron Devices*. Vol. 41, No. 6, June 1994. pp 988-997.

[74] L. Mei and R. Dutton. “A Process Simulation Model for Multilayer Structures Involving Polycrystalline Silicon.” *IEEE Transactions on Electron Devices*. Vol. ED-29, No. 11, Nov. 1982. pp 1726 - 1734.

[75] *Tsuprem4 User’s Manual*. Avant! Corp. Milpitas, CA.

[76] L. Mei, M. Rivier, Y. Kwark, and R. Dutton. “Grain Growth Mechanisms in Polysilicon.” *Journal of Electrochemical Society*. Vol 129, No. 2, Aug. 1982. pp 1791-1795.

[77] C. Thompson. “Secondary Grain Growth in Thin Films of Semiconductors: Theoretical Aspects.” *Journal of Applied Physics*. Vol. 58. No 2, July 1985. pp. 763-772.

[78] A. O’Neill, C. Hill, J. King, and C Pease. “A New Model for the Diffusion of Arsenic in Polycrystalline Silicon.” *Journal of Applied Physics*. Vol. 64, No. 1, July 1988. pp 167-174.

[79] M. Mandurah, K Saraswat, R. Dutton, and T. Kamins. “Dopant Segregation in PolySilicon.” *Journal of Applied Physics*. Vol. 51, No. 11, Nov. 1980. pp 5755-5763.

[80] H. Sigg, G. Vendelin, T. Cauge, and J. Kocsis. “D-MOS Transistor for Microwave Applications.” *IEEE Transaction on Electron Devices*. Vol. ED-19, No. 1, Jan. 1972. pp 45-54.

[81] G. Ma, W. Burger, C. Dragon, and T. Gillenwater. “High Efficiency LDMOS Power FET for Low Voltage Wireless Communications.” *Proceedings of IEDM*. Dec. 1996. pp 91-94.

[82] A. Wood, C. Dragon, and W. Burger. “High Performance Silicon LDMOS Technology for 2GHz RF Power Amplifier Applications.” *Proceeding of IEDM*. Dec. 1996. pp 87-90.

[83] R. Mertens, R. Van Overstraeten, and H. De Man. “Heavy Doping Effects in Silicon” *Advanced Electronics and Electron Physics*. Editors: L. Marton and C. Matron. New York: Academic Press, 1981.

[84] C. Lombardi, S. Manzini, A. Saporito, and M. Vanzi. "A Physically Based Mobility Model for Numerical Simulation on Non-planar Devices." *IEEE Transactions Computer Aided Design*. Vol. 7, No 11, Nov. 1988. pp 1042-1048.

[85] K. Thornber. "Relation of Drift Velocity to Low-Field Mobility and High-Field Saturation Velocity." *Journal of Applied Physics*. Vol. 51, No. 4, April 1980. pp 2127-36.

[86] S. Wolf and R. Tauber. *Silicon Processing for the VLSI Era. Volume 1: Process Technology*. Sunset Beach: Lattice Press, 1986.

[87] *Terrain User's Manual*. Avant! Corp. Milpitas, CA.

[88] *Raphael User's Manual*. Avant! Corp. Milpitas, CA.

[89] D. Montgomery. *Design and Analysis of Experiments*. New York: John Wiley and Sons, 1997.

[90] R. Stratton, "Semiconductor Current-Flow Equations (Diffusion and Degeneracy)," *IEEE Transaction on Electron Devices.* Vol. ED-19, Dec. 1972. pp 1288-1292.

[91] W. Hansch and M. Miura-Mattausch, "The Hot-Electron Problem in Small Semiconductor Devices," *Journal of Applied Physics.* Vol. 60, No. 2, July 1986. pp 650-656.

[92] J. W. Slotboom, G. Streutker, M. van Dort, P. Woerlee, A. Pruijmboom, and D. Gravestejn, "Non-local impact ionization in silicon devices," *Proceedings of IEDM*. Dec. 1991. pp 127-130.

[93] Z. Yu, D. Chen, L. So, S. Beebe, R. Goossens, F. Rotella, R. Dutton. *Pisces 2ET with Applications Documentation*. Board of Trustees of Leland Stanford Junior University. Stanford, CA. Dec. 1994.

[94] A. Pieracci, M. Lanzoni, P. Galbiati, S. Manzini, C. Contiero, B. Ricco. "Extraction of channel doping profile in DMOS transistors." *Proceeding of IEDM*. Dec. 1996. pp 485-488.

[95] J. Sato-Iwanga, K. Fujimoto, H. Masato, Y. Ota, K. Inoue, B. Troyanovsky, Z. Yu, and R. W. Dutton. "Distortion Analysis of GaAs MESFET's Based on Physical Model using PISCES-HB." *Proceedings of IEDM*. Dec. 1996. pp. 163-166.

[96] J. Jang, E. Kan, L. So, R. Dutton. "Parasitic characterization of radio-frequency (RF) circuits using mixed-mode simulation." *Proceedings of Custom Integrated Circuits Conference*. May 1996. pp 445 -448.

[97] G. Gonzalez. *Microwave Transistor Amplifiers: Analysis and Design*. New Jersey: Prentice Hall, 1997.

[98] J. Cusack, S. Perlow, and B. Perlman. "Automatic Load Contour Mapping for Microwave Power Transistors." *Transactions on Microwave Theory and Techniques*. Vol. MTT-22, No. 12, Dec. 1974. pp 1146-1152.

[99] *Luminous: Optoelectric Device Simulator*. Silvaco International, Santa Clara, CA