INFORMATION TO USERS

The most advanced technology has been used to photograph and reproduce this manuscript from the microfilm master. UMI films the text directly from the original or copy submitted. Thus, some thesis and dissertation copies are in typewriter face, while others may be from any type of computer printer.

The quality of this reproduction is dependent upon the quality of the copy submitted. Broken or indistinct print, colored or poor quality illustrations and photographs, print bleedthrough, substandard margins, and improper alignment can adversely affect reproduction.

In the unlikely event that the author did not send UMI a complete manuscript and there are missing pages, these will be noted. Also, if unauthorized copyright material had to be removed, a note will indicate the deletion.

Oversize materials (e.g., maps, drawings, charts) are reproduced by sectioning the original, beginning at the upper left-hand corner and continuing from left to right in equal sections with small overlaps. Each original is also photographed in one exposure and is included in reduced form at the back of the book. These are also available as one exposure on a standard 35mm slide or as a 17" x 23" black and white photographic print for an additional charge.

Photographs included in the original manuscript have been reproduced xerographically in this copy. Higher quality 6" x 9" black and white photographic prints are available for any photographs or illustrations appearing in this copy for an additional charge. Contact UMI directly to order.

U·M·I

University Microfilms International
A Bell & Howell Information Company
300 North Zeeb Road, Ann Arbor, MI 48106-1346 USA
313/761-4700 800/521-0600

Order Number 1337491

Two-dimensional device simulation of junction termination structures for determination of breakdown behavior

Tan, Leong Hin, M.S.

The University of Arizona, 1989

U·M·I
300 N. Zeeb Rd.
Ann Arbor, MI 48106

TWO-DIMENSIONAL DEVICE SIMULATION OF JUNCTION TERMINATION STRUCTURES FOR DETERMINATION OF BREAKDOWN BEHAVIOR

by

Leong Hin Tan

A Thesis Submitted to the Faculty of the

DEPARTMENT OF ELECTRICAL AND COMPUTER ENGINEERING

In Partial Fulfilment of the Requirements

For the Degree of

MASTER OF SCIENCE

WITH A MAJOR IN ELECTRICAL ENGINEERING

In the Graduate College

THE UNIVERSITY OF ARIZONA

1 9 8 9

STATEMENT BY AUTHOR

This thesis has been submitted in partial fulfilment of requirements for an advanced degree at The University of Arizona and is deposited in the University Library to be made available to borrowers under rules of the library.

Brief quotations from this thesis are allow without special permission, provided that accurate acknowledgment of source is made. Requests for permission for extended quotation from or reproduction of this manuscript in whole or in part may be granted by the head of the major department or the Dean of the Graduate College when in his or her judgment the proposed use of the material is in the interests of scholarship. In all other instances, however, permission must be obtained from the author.

SIGNED: _____

APPROVAL BY THESIS DIRECTOR

This thesis has been approved on the date shown below:

Dr. François E. Cellier
Associate Professor of ECE

July 3, 1989
Date

Dedicated to:

my beloved wife, Vijit Chinburapa for her loving support, encouragement and patience;

my late eldest brother, Tan Leong Piow, who was dearly missed by our whole family;

my beloved sister, Tan Swee Tee, who worked very hard during hard times to support our family.

ACKNOWLEDGMENTS

As the author of this thesis, I feel it is necessary to acknowledge those individuals who helped make this educational experience a reality. I am truly grateful to all who assisted me throughout my education. I am particularly grateful to Dr. François E. Cellier, who served as my thesis director and major advisor. His continual support and guidance made this endeavor extremely rewarding. I would also like to thank Drs. Kenneth F. Galloway and Ronald D. Schrimpf for reviewing the original manuscript. I would also like to thank Chi-min Yen, who initially developed the device simulator, SEPSIP 1.0, for his suggestions. In particular, I acknowledge my debt to Burr–Brown Corporation, Tucson, Arizona, who provided technical and financial support in this work.

I owe a special thanks to my family. First, to my Dad, Tan Kong Chiam, my Mum, Lee Geok Khim and my brother Tan Leong Beng, who gave full encouragement and financial support throughout my education. I owe a great thank to my wife, Vijit Chinburapa for her constant encouragement and patience while this work has been in progress. I owe special thanks to my best friends, Chiang Chye Huat and Sim See Wah, who gave encouragement and financial assistance during the initial stage of my overseas education.

CONTENTS

LIST OF FIGURES

ABSTRACT

Breakdown voltage is one of the major parameters to be considered in the design of power semiconductor devices. Many techniques have been developed to improve the breakdown behavior of power semiconductor devices. These include field plates and floating field rings. It is important to be able to predict the breakdown behavior of a device before the device is actually fabricated. Analytical techniques can only be used to predict the breakdown behavior of very simple devices. The breakdown behavior of more advanced devices that make use of complex termination structures cannot be predicted analytically. The breakdown behavior of such device can, however, be quite accurately predicted by two-dimensional simulation.

In this work, we have investigated numerical techniques to determine the breakdown behavior of complex semiconductor devices using two-dimensional simulation. In particular, we have augmented the device simulator SEPSIP with a capability for handling single and multiple floating field rings, and for handling devices with slanted edges. We have furthermore improved the grid width selection algorithm in SEPSIP. A capability for plotting equi-field contours was added to the code. Finally, all system dependencies were removed from the SEPSIP code, and a new version of SEPSIP (Version 2.0) was generated which can be executed on any PC/XT, PC/AT, or PC/386 compatible computer. This eliminates the need for transfering files back and forth between the PC, which had formerly been used as an I/O processor, and the VAX, which was used for numerically intensive computations. It also makes the code more accessible to scientists and engineers who are working in this important research area.

CHAPTER 1
INTRODUCTION

Breakdown voltage is one of the major parameters to be considered in the design of power semiconductor devices. Many techniques have been developed to improve the breakdown behavior of power semiconductor devices. These include field plates and floating field rings. All these techniques influence the surface behavior of the device. They are also applied to the vicinity of the edge of the device. For these reasons, they are globally referred to as *junction termination structures.*

Since device fabrication is an expensive and time consuming process, it is important to be able to predict the breakdown behavior of a device before the device is ever actually fabricated. Analytical techniques can only be used to predict the breakdown behavior of very simple devices. The breakdown behavior of more advanced devices that make use of complex termination structures cannot be predicted analytically. The breakdown behavior of such devices can, however, be quite accurately predicted by two-dimensional simulation.

Two-dimensional numerical simulation of semiconductor devices has become an extremely important field in semiconductor device physics. Two-dimensional simulation is often necessary for an accurate analysis of complex device structures. As device structures incorporating special junction terminations are complex, analytical results cannot be found or are coarse approximations, and therefore, accurate results can only be obtained using two-dimensional numerical simulation.

Many two-dimensional device simulators such as MINIMOS[1], BAMBI[2], and PISCES-IIB[3] have been developed. These software systems are capable of analyzing the avalanche breakdown behavior of a device. The numerical techniques employed in these software systems can be categorized into those using Poisson's equation simultaneously with the current continuity equations[4–9] and those using only Poisson's equation[10–16]. The former numerical solution technique requires much more computational time and is therefore much more expensive than the latter approach of using the solution of Poisson's equation alone.

In the analysis of breakdown behavior, the numerical solution of Poisson's equation alone can be used. This is because the semiconductor device operates under high reverse voltage conditions, and therefore, the currents are negligible. Hence, the physical model can be simplified. However, the previously developed software systems were not designed for analyzing breakdown behavior in particular. For the application discussed here, they all have severe limitations since they make rigid assumptions concerning the device structures to be simulated and/or the impurity distributions to be used.

A flexible and more specialized two-dimensional power device simulator, SEPSIP[17] - SEmiconductor Power device SImulation Program, has been developed to investigate and analyze the breakdown behavior of junction termination structures under reverse-bias conditions. SEPSIP is capable of analyzing the breakdown behavior in nearly all kinds of power devices such as metal oxide semiconductor field effect transistors (MOSFET's), and bipolar junction transistors (BJT's) because of its unique treatment of the device topology. In addition, its numerical solution, namely the Newton-SOR[18–19] iteration scheme of the semiconductor Poisson's equation is efficient and numerically stable.

To improve the versatility of SEPSIP, a capability for handling various types of junction termination structures that improve the breakdown behavior of power devices, such as single and multiple floating field rings, has been added to the code. Also, SEPSIP has been enhanced to allow the simulation of slanted-edge semiconductor device geometries. With these additions, SEPSIP has become a very powerful tool for the design and optimization of power electronic semiconductor devices.

In Chapter 2, a review of the literature regarding various breakdown improvement techniques is presented. Field plates and floating field rings are examples of the breakdown improvement techniques being reviewed. Also, the slanted edged semiconductor device geometry is discussed.

Implementation of additional features into SEPSIP are discussed in Chapter 3. More specifically, an adaptive grid generation scheme, non-rectangular device structures, floating structures, equipotential contours, and electric field contours are discussed in this chapter.

Chapter 4 illustrates the use of SEPSIP to investigate one of the breakdown improvement techniques, the floating ring. This technique is used to demonstrate the apparent improvement in breakdown behavior of a power semiconductor device. Additionally, a slanted edged power device is simulated to show the equifield and equipotential contours.

The necessary information on the usage of the SEPSIP software simulator is provided in Appendix A. The required input files for the simulator are the device structure, the input file, and the impurity concentration profiles. A detailed description of how to create these input files will be presented in this Appendix.

CHAPTER 2

REVIEW OF THE LITERATURE

This chapter presents a review of the literature regarding breakdown improvement techniques. The techniques reviewed are field plates and floating field rings. In addition, the major characteristics of selected previously developed simulators for two-dimensional device structures will be reviewed.

It is known that avalanche breakdown of a *p-n* junction occurs either near the surface of the semiconductor region or at the maximum junction curvature[20]. Gibbons and Kocsis[21] studied the breakdown voltages of cylindrical junctions with an abrupt impurity distribution. They found that the breakdown voltage decreased with decreasing radius of curvature. For a given background doping, the plane junction always has a larger breakdown voltage than the cylindrical junction.

Sze and Gibbons[20] examined the effect of the radius of curvature of the metallurgical junction (or junction curvature) on avalanche breakdown for spherical and cylindrical *p-n* junctions in Ge, Si, GaAs, and GaP. Three types of impurity distributions were considered: abrupt, linearly graded, and composite. The effect of the radius of curvature on the breakdown voltage was found to be greater for spherical junctions than for cylindrical junctions. In addition, this effect was found to differ depending upon the composition of the impurity distribution studied. For abrupt junctions, the breakdown voltage was found to decrease with decreasing radius of curvature, whereas, for linearly graded junctions, no relationship was found between the breakdown voltage and the radius of curvature. For composite[20] junctions, the breakdown voltage behavior at a small impurity gradient was similar

to that of a linearly graded junction. However, at large impurity gradients, the breakdown voltage is determined by the radius of curvature and the background doping.

2.1 Field Plate Improvement Technique[22]

The investigations of Gove, Leistiko, and Hooper[23] indicated that a field plate (see Fig. 2.1) can be used to control the curvature of the space-charge layer near the surface. The breakdown voltage can be varied over a very wide range by the application of an external field plate. When the gate voltage was varied to deplete the low doped side of the junction, the breakdown voltage tended to saturate at the maximum value of the breakdown range. However, if the highly doped side of the junction is depleted by varying the gate voltage, the breakdown voltage tended to saturate at the minimum value of the breakdown range. Both the maximum and the minimum saturation of the breakdown voltage occurred as a result of the formation of field-induced junctions which prevent further variation of the junction shape in the depletion region.

Conti and Conti[24] studied surface breakdown in silicon planar diodes equipped with field plates. The field distribution around the surface junction was found to be influenced by the field plate and by the oxide thickness. The thinner the oxide, the higher is the electric field in the oxide and around the edge of the field plate. This caused the avalanche multiplication to be localized near the edge of the field plate due to the geometrical enhancement of the electric field. Hence, if the oxide layer is not sufficiently thick, the avalanche breakdown voltage is considerably lower than the value allowed by the semiconductor material. Furthermore, the voltage drop into the oxide is an increasing function of the oxide thickness. As the oxide thickness was increased to avoid edge avalanche, the surface junction

breakdown increased. This caused the breakdown voltage at the surface junction to approach the breakdown voltage value of the cylindrical junction without a field plate.

Rusu and Bulucea[12] went on to show that the breakdown voltage is controlled by the oxide thickness and by the substrate impurity concentration. A universal, normalized criterion was derived for field uniformity as a ratio of oxide thickness to the maximum width of the silicon depletion region. This ratio should be larger than 0.3 to avoid field concentration around the edges of the metal field plate.

O'Neil and Alonas[25] studied other factors affecting the breakdown voltage when using field plates. The electric field in the silicon is influenced by the oxide thickness as well as by the junction depth of the *p-n* junction. However, oxide thickness was found to exert a greater effect on the electric field in silicon than did the junction depth. A unit increase in oxide thickness resulted in a greater reduction in the silicon electric field than did the same unit increase in the junction depth. A simple closed-form analytical expression for the breakdown voltage of uniformly doped *p-n* junctions using a field plate has been developed.

$$V_{B_{fp}} = 60\left(\frac{N_D}{10^{16}}\right)^{-0.75}\left(\sqrt{\gamma(2+\gamma)}-\gamma\right) \tag{2.1}$$

$$\gamma = \frac{\epsilon_{Si} t_{ox}}{\epsilon_{ox} W_d} \tag{2.2}$$

where N_D is the donor concentration in the n region and W_d is the maximum depletion depth at the planar breakdown voltage.

Fig. 2.1 shows the device topology for which these expressions were derived. The analytical expression is useful when designing planar devices of this type requiring high voltage capabilities.

Hwang and Navon[26] studied the design of optimized high voltage planar p–π–ν diode structures. This diode structure is shown in Fig. 2.2. A heavily doped p^+-region is embedded in a more lightly doped p^--region (here referred to as the π-region), which in turn is embedded in a lightly doped n^--region (referred to as the ν-region). In this way, the transition between the p^+-region and the n^--region is made less sharp which helps in reducing the highest fields in the vicinity of the p-n junction. To determine the individual effects of the device parameters on the breakdown voltage, only one parameter was varied at a time while all other parameters were held constant.

First, the width of the π-region was considered. A wide layer of the π-region near the region of large junction curvature and the surface decreased the electric field and reduced the field concentration due to the curvature effect. Hence, the peak field at the surface and at the junction curvature were found to be reduced when the width was increased. However, the increase of the breakdown voltage saturates when the width is made sufficiently large.

The impurity concentration in the π-region also affects the high field point. When the doping concentration is low, the highest field point occurs near the p-π junction. For high doping concentrations, the highest field point occurs near the $\pi - \nu$ junction.

Additionally, the length of the field plate affects the electric field at the curvature of the p-π junction. As the length increases toward the $\pi - \nu$ junction, the electric field at the p-π junction is reduced. However, the electric field at the

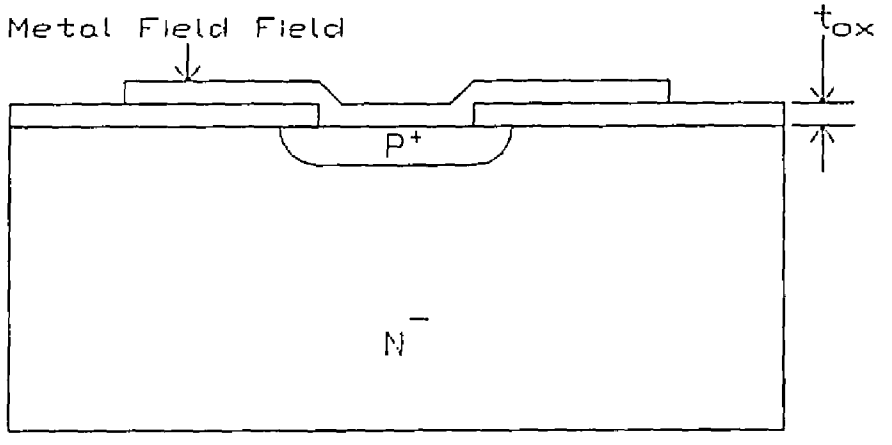

Figure 2.1 p^+-n diode with field plate.

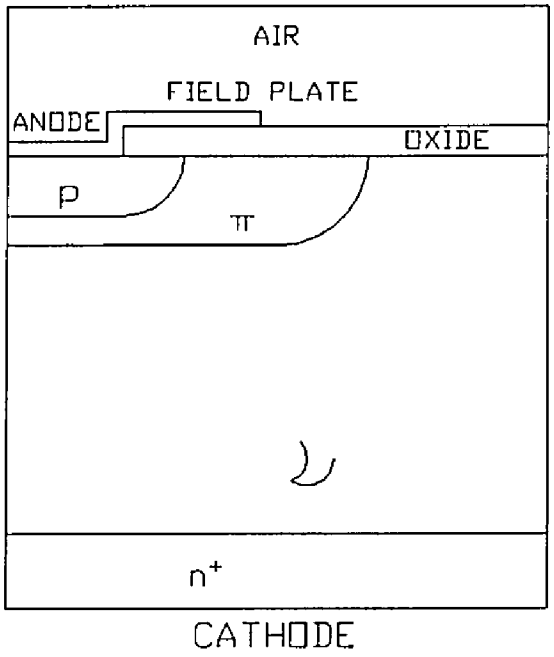

Figure 2.2 Geometry of the $p - \pi - \nu$ diode.

edge of the field plate is also increased. Therefore, there exists an optimal plate length that will yield the highest breakdown voltage.

There can be both mobile and immobile charges near the interface of the oxide and the silicon. In silicon, the surface charge is usually positive. The fixed positive surface charge Q_{ss} attracts electrons and repels holes near the surface. Thus, the surface charge affects the device characteristics. Hwang and Navon[26] showed that the decrease of the breakdown voltage is not significant when the surface charge is small, i.e. $Q_{ss} < 10^{11} cm^{-2}$, but the breakdown voltage decreases drastically when the surface charge is large, $Q_{ss} > 5 \times 10^{11} cm^{-2}$. These effects were studied in great detail by my colleague, Davies. The results of his findings were presented in his MS Thesis[27].

2.2 Floating Field Rings Improvement Technique[22]

For some time, it has been known that, by using one or more floating rings, the effects of the junction curvature can be reduced resulting in a significantly lower surface electric field [28–33]. Thus, the implementation of field limiting rings increases the avalanche breakdown voltage and improves the reliability of a high-voltage *p-n* junction of a power semiconductor device.

In the analysis of breakdown behavior for high-voltage bipolar devices, Wu and Cellier[34] demonstrated that when the length of the field plate of one electrode is decreased, the highest field value also decreases, and the highest field point moves from the surface of the silicon to the *p-n* junction curvature. In addition, the thickness of the oxide layer (SiO_2) affected the highest field location on the surface of the *p-n* junction high resistance area. As the oxide thickness is increased, the maximum electric field decreases.

A floating field limiting ring was then added to the high-voltage device. It was found that the field limiting ring caused the equi-potential lines to move closer to the high reverse-bias electrode contact region. This affected the location of the highest field which moved from the p-n junction curvature down toward the silicon bulk, hence reducing the field near the p-n junction.

A non-rectangular geometry device was also simulated to study the influence of dielectric isolation at the slanted edge of the device. A 100 V reverse-bias voltage was applied to the device. The simulation results indicated that the highest field point occurred on the interface between the dielectric isolation layer and the silicon[34].

Adler, *et al*[35] investigated the extent of the improvement in breakdown voltage obtained by using floating field rings. Exact avalanche multiplication factor calculations were made once the electric field and potential distributions had been determined. This multiplication factor was used to determine the breakdown voltage. The study indicated that a single floating field ring can be optimally placed near the p-n junction to improve the breakdown voltage. In addtion, the single field ring was found to be most effective on devices with small radii of curvature. The improvement ranged from a factor of 1.8 at low effective radii of curvature down to a factor of 1.1 at high effective radii of curvature.

2.3 Other Semiconductor Device Simulators

Two-dimensional simulation of power devices under high reverse-bias voltage is computationally costly. Wu and Cellier[36] investigated different numerical techniques for the simulation of bipolar high-voltage devices under the assumption of negligible current flow, and determined the most numerically effective set of simulation parameters for this type of simulation.

BAMBI[2] is a special purpose steady-state device simulator which requires only the device topology and the doping concentration to be specified as inputs. BAMBI works quite well for low-voltage devices. However, the computational time increases rapidly when high reverse-bias voltage devices are being simulated. When applied to devices that are operated under high reverse-bias conditions, BAMBI frequently does not converge at all. This is due to the fact that BAMBI solves the Poisson's equation and the current continuity equations simultaneously.

Another frequently used device simulator is MINIMOS[1]. MINIMOS is a very user-friendly software package for the two-dimensional simulation of planar MOS transistors. It uses robust programming techniques and optimized numerical algorithms to ensure flexibility and low computing cost. However, MINIMOS suffers from the same drawbacks as BAMBI, i.e., it works reasonably well for low-voltage devices, but often fails when used for high-voltage power devices.

PISCES-IIB[3] is a less specialized device simulator. It is somewhat less user-friendly than the previously mentioned programs, but it is applicable to a wider range of device structures. PISCES-IIB often offers alternate algorithms for the same purpose which is a useful feature since it enables the user to rerun the same simulation using a different algorithm, and helps him thereby to identify whether an observed effect is a true device feature or simply a numerical artifact. However, PISCES-IIB has severe drawbacks. The software is unfortunately not very robust. On several occassions, incorrect results rather than an error message were obtained when PISCES-IIB is unable to come up with the correct answer, i.e., the software did not properly detect its own limitations, and did not come up with an appropriate error condition. These problems occurred, in particular when PISCES-IIB was used in the CARRIERS = 0 mode, i.e., when PISCES-IIB was set to solve the Poisson's equation alone. Also, while PISCES-IIB can either solve the Poisson's equation

together with the current continuity equations or the Poisson's equation alone, this second alternative will not work at all for field ring structures. The only way the user can formulate in PISCES-IIB the existence of a floating field limiting ring is by declaring it as a current contact with zero current. However, under those conditions, PISCES-IIB will reject solving the Poisson's equation alone.

CHAPTER 3

NEW ADDITIONS FOR SEPSIP

SEPSIP was developed to investigate the breakdown behavior of power semiconductor devices such as MOSFETs and BJTs. The original version of the code, SEPSIP 1.0, was able to handle only the basic device structures. SEPSIP 1.0 was inadequate to simulate the breakdown behavior of today's commercial power devices that employ special junction termination structures such as floating field rings. For this purpose, the software needed to be modified to enhance its capabilities. The new additions to SEPSIP are described in this chapter.

3.1 Description of the Numerical Techniques Used in SEPSIP[17]

SEPSIP uses a five-point finite difference approximation to discretize Poisson's equation. Poisson's equation for a semiconductor device is

$$-\nabla \cdot (\epsilon \nabla \phi) = \rho \tag{3.1}$$

where ϕ is the potential to be solved, ϵ is the permittivity of the semiconductor and ρ is the total charge density given by

$$\rho = \begin{cases} q\,(p - n + N_D - N_A) & \text{for semiconductor} \\ 0 & \text{for dielectrics.} \end{cases} \tag{3.2}$$

In (3.2), N_D and N_A denote the densities of ionized donors and acceptors, respectively. In our case, ρ in the dielectric is assumed 0, but for some cases, it may be

equal to a volumetric charge density of ρ_v. Using Boltzmann's approximation, the carrier concentrations can be expressed as

$$\begin{aligned} n &= n_i e^{\frac{q}{kT}(\phi-\phi_n)} \\ p &= n_i e^{\frac{q}{kT}(\phi_p-\phi)} \end{aligned} \tag{3.3}$$

where n_i is the intrinsic carrier concentration ϕ_n is the quasi-Fermi potential for electrons, ϕ_p is the quasi-Fermi potential for holes and the quantity kT/q is often referred to as the thermal voltage.

Equation (3.1) is discretized into

$$a_1\phi_{i,j} - (a_2\phi_{i-1,j} + a_3\phi_{i+1,j} + a_4\phi_{i,j-1} + a_5\phi_{i,j+1}) = q_v + q_{ss} \tag{3.4}$$

Fig. 3.1 shows an individual cell with their coefficients a_1 through a_5, each of which can be identified as

$$\begin{aligned} a_1 &= a_2 + a_3 + a_4 + a_5, \\ a_2 &= \frac{\epsilon_{21}t_j + \epsilon_{31}t_{j-1}}{2s_{i-1}}, \\ a_3 &= \frac{\epsilon_{41}t_{j-1} + \epsilon_{11}t_j}{2s_i}, \\ a_4 &= \frac{\epsilon_{32}s_{i-1} + \epsilon_{42}s_i}{2t_{j-1}}, \\ a_5 &= \frac{\epsilon_{12}s_i + \epsilon_{22}s_{i-1}}{2t_j}. \end{aligned} \tag{3.5}$$

and

$$\begin{aligned} q_v &= \int_A \rho da, \\ q_{ss} &= \int_L \rho_{ss} dl. \end{aligned} \tag{3.6}$$

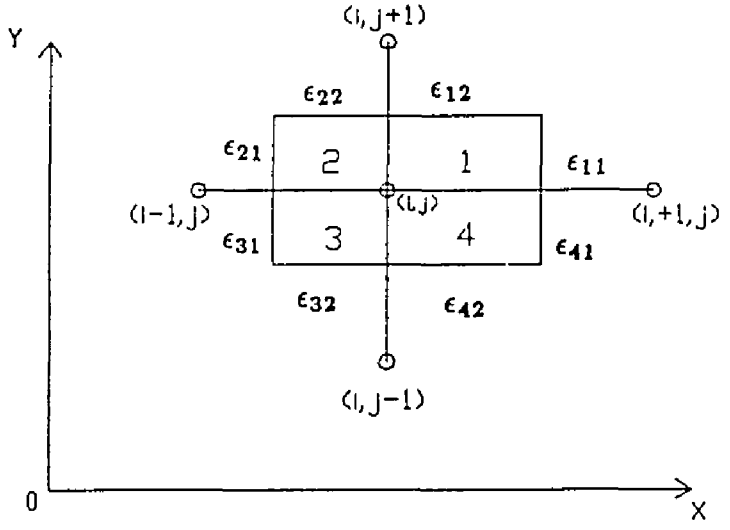

Figure 3.1 Five-point star computational cell.

To streamline the calculation, normalization is performed on the difference equations. Normalization turns the equations into their dimensionless form. After normalization, the equation (3.4) can be written in a matrix form to facilitate the iteration scheme,

$$\mathrm{Bu} = \mathrm{f} \tag{3.7}$$

where u is the solution vector of which element $u_{i,j}$ is the potential u evaluated at point (i, j), and f is the driving vector of which element $f_{i,j}$ is the value of f at (i, j) plus the contribution from the boundaries.

Since the right hand side of equation(3.7) is non-linear, Newton's Method[37] of linearization is used. Then the matrix equation is solved repetitively, until convergence occurs. Finally, Sucessive Over-Relaxation(SOR)[18] is the iteration scheme used to solve the linearized system of equation(3.7).

3.2 Coefficient Derivation for Slanted Edges

A slanted edge may be oriented either as a left slant or a right slant as shown in Fig. 3.2(a) and (b), respectively. Points lying on the slanted edge involve two different permittivities, ϵ_1 and ϵ_2. Let us consider the left slant shown in Fig.3.2(a) where ϵ_1 is the permittivity below the slanted edge and ϵ_2 is the permittivity above the slanted edge. Assuming the grid spacings s_{i-1} is equal to s_i, and t_{j-1} is equal to t_i, then equation(3.5) becomes

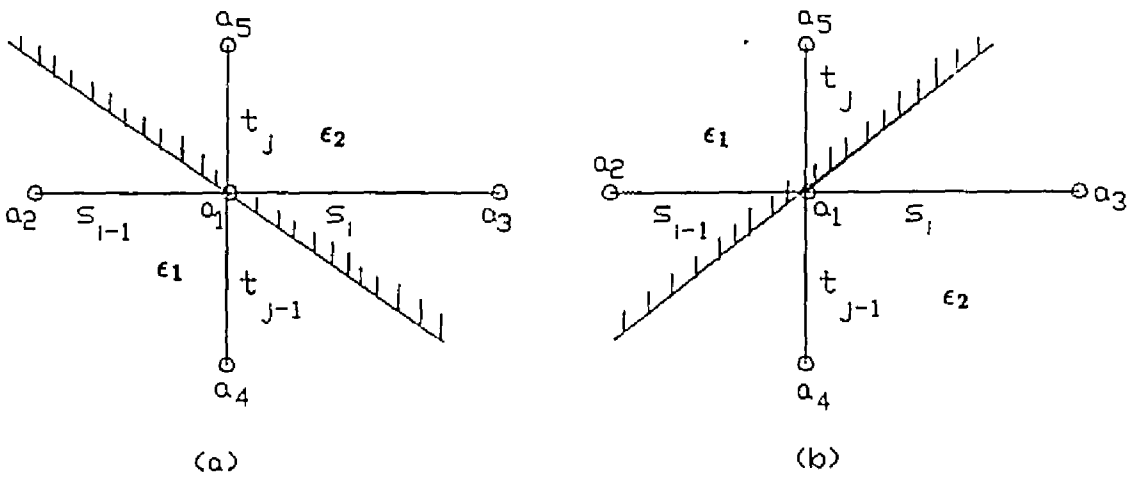

Figure 3.2 (a) left slanted interface (b) right slanted interface.

$$
\begin{aligned}
a_1 &= \frac{\bar{\epsilon}}{2}(s_{i-1}+s_i)(t_{j-1}+t_j)\left(\frac{1}{s_{i-1}s_i}+\frac{1}{t_{j-1}t_j}\right),\\
a_2 &= \frac{\epsilon_1}{2}\frac{t_{j-1}+t_j}{s_{i-1}},\\
a_3 &= \frac{\epsilon_2}{2}\frac{t_{j-1}+t_j}{s_i},\\
a_4 &= \frac{\epsilon_1}{2}\frac{s_{i-1}+s_i}{t_{j-1}},\\
a_5 &= \frac{\epsilon_2}{2}\frac{s_{i-1}+s_i}{t_j}
\end{aligned}
\tag{3.8}
$$

where $\bar{\epsilon} = \frac{1}{2}(\epsilon_1 + \epsilon_2)$.

3.3 Adaptive Grid Generation

In the version of SEPSIP developed by Yen[17], the *xstep* and *ystep* parameters are the number of grid spacings per unit length in the horizontal and vertical directions, respectively. The *xslope* and *yslope* parameters are the slopes of the respective grid spacing distribution. These parameters controlled the total number of nodes in the x and y directions. These parameters could be somewhat confusing to the user. It seemed more reasonable to request an input as to the maximum number of grid points to be used in each direction, and then allow the code to determine an optimal grid distribution. Consequently, in the new version of SEPSIP, *xmaxpt* denotes the maximum number of grid points to be used in the x direction, and *ymaxpt* denotes the maximum number of grid points in the y direction. Assuming that the total length in x-direction and y-direction are x_t and y_t, respectively, SEPSIP 2.0 will then automatically compute the values for *xstep* and *ystep* as follows:

$$
\begin{aligned}
xstep &= \frac{xmaxpt}{x_t}\\
ystep &= \frac{ymaxpt}{y_t}
\end{aligned}
\tag{3.9}
$$

The *xslope* parameter does not influence the number of nodes in the x-direction if $xstep \leq 15$. For $xstep > 15$, SEPSIP 2.0 will adjust *xslope* automatically as follows:

$$xslope = \begin{cases} 0.1 & \text{for } xstep \leq 15; \\ \left(\frac{xstep}{100.0}\right) - 0.05 & \text{for } xstep > 15. \end{cases} \tag{3.10}$$

The *yslope* parameter is adjusted accordingly.

3.4 Handling of Floating Structures in SEPSIP 2.0

When a conductor is not biased, it is considered to be a floating structure. Given a two-dimensional impurity profile, SEPSIP 2.0 will identify all junctions present in the structure. If any region within the semiconductor is not biased, it is considered to be a floating structure.

Initially, all floating structures are set to the lowest potential in the semiconductor device. After each iteration, the minimum potential around the boundary of each floating ring is calculated. If the minimum potential is greater than the potential of the floating ring, then the floating ring is initialized to the new minimum potential plus the built-in potential of the floating ring. This procedure is repeated for all floating structures after each iteration step. Hence, the potential of each floating structure is constantly updated. The iteration does not stop until consecutive values of all floating structures don't change significantly any longer.

This algorithm works correctly if no punch-through occurs between neighboring junctions. The consideration of punch-through is discussed in the MS Thesis of my colleague, Davis[27].

A simple floating ring geometry with its grid nodes is shown in Fig. 3.3. After each iteration, the potentials around the boundary of the floating ring are evaluated, and the minimum potential around this floating ring is determined. In

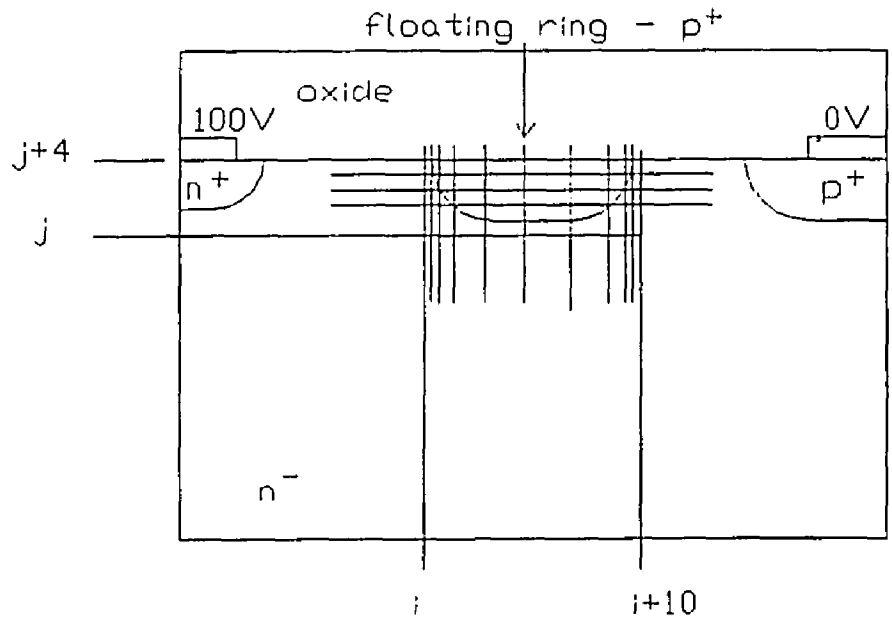

Figure 3.3 Comparing potentials around the floating ring.

this example, the potentials at the following nodes are taken into consideration to obtain the minimum potential.

(i, j+4), (i, j+3), (i+1, j+2), (i+2, j+1), (i+3, j) to (i+7, j), (i+8, j+1), (i+9, j+2), (i+10, j+3), and (i+10, j+4)

The minimum value among these potentials is then compared to the current potential of the floating field ring. If the minimum boundary potential is greater, then the entire floating field ring is re-initialized to this value plus the built-in potential of the floating field ring.

3.5 Other Features of SEPSIP 2.0

The user is now able to specify the values of the equipotential contours to be plotted. This gives the user greater control over the values of the plotted equipotential lines. For example, if the number of requested equipotential contours, *nlevel* is five, the values of the equipotentials, *vlevel*(i) can be specified, where i goes from 1 to 5. The maximum number of equipotential contours that can be plotted is nine.

Another new feature is the availability of equi-field contours. This enables the user to visualize where the concentration of the equi-field contours is on the semiconductor structure. When *plote* = *.true.*, the equi-field contours are plotted from the file *sep1.dxf* with the number of contours equal to *nlevel*.

CHAPTER 4

POWER SEMICONDUCTOR DEVICE DESIGN USING SEPSIP

4.1 Breakdown Behavior of High Voltage Devices

The most common cause of breakdown for power electronic devices operating under a high reverse-bias condition is avalanche breakdown[38–40]. For a power device operating under high-reverse bias, the high doping region pushes the equipotential contours into the depletion region. This crowding of equipotential lines gives rise to the localization of a high electric field around the curvature of the p-n junction.

The high electric field gives the free carriers enough energy to break the covalent bond between the atom of the crystal and its electrons[38–40]. This generates more free carriers. The free carriers, in turn, break more covalent bonds. Hence, it generates a domino effect that results in the avalanche breakdown.

4.2 Simulation of a Power Device Without a Floating Ring.

The geometry of a bipolar power device to be simulated under high reverse bias conditions is shown in Fig. 4.1. This structure can either represent a vertically diffused NPN transistor, or a laterally diffused PNP transistor. We want to assume this to be a NPN transistor. The emitter (which is diffused into the base) has been omitted from the simulation since the base-emitter junction is forward biased, and since we operate under the assumption of a negligible current flow. In our simulation, the substrate is biased at 45 *Volts*. This can be achieved by either placing a highly doped n^+ burried layer at the bottom of the structure, and operating with

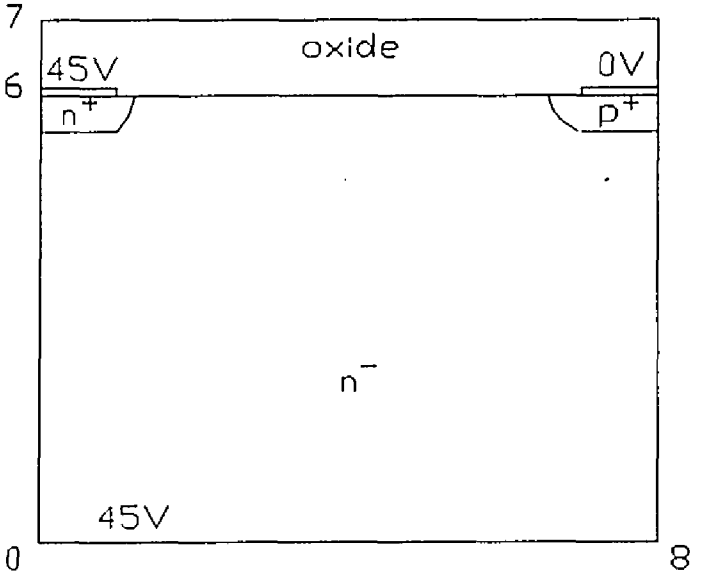

Figure 4.1 Bipolar power device without floating ring.

a "sinker", i.e., the n^+ collector is diffused down into the burried layer, or alternatively, we can achieve a similar effect with a dielectric isolation, by biasing the polysilicon at 45 $Volts$.

In our simulation, both the p^+ and the n^+ regions have a depth of 0.5 μm. The applied reverse bias voltage is 45 V. The simulation results are shown in Fig. 4.2 for the equipotential contours and high electric field points. As seen from the diagram, the equipotential lines are close to each other and curve around the curvature of the p-n junction. This causes the high electric field points to be located near the curvature of the p-n junction.

In Fig. 4.3, the concentration of high equi-field contours around the curvature of the p-n junction is plotted. The high electric field points are located in the center of the equi-field contours. The values of the high electric field points are found to be above 3×10^5 V/cm. These values exceed the critical field value of about 3×10^5 V/cm for the given impurity concentration of 1×10^{15} cm^{-3}, indicating that breakdown occurs.

It can be observed that, in this simulation, the depletion region of the p-n junction reaches through to the n^+ layer. Consequently, we do not gain the full benefit of the low background doping since the depletion region is stopped by the n^+ region which increases the crowding of the field around the p-n junction. It can also be observed that the substrate is a little too shallow. The depletion region reaches down to the bottom of the substrate, where, for the same reasons as explained before, it is prevented from growing any further, and consequently, leads to an increased crowding of the field around the junction.

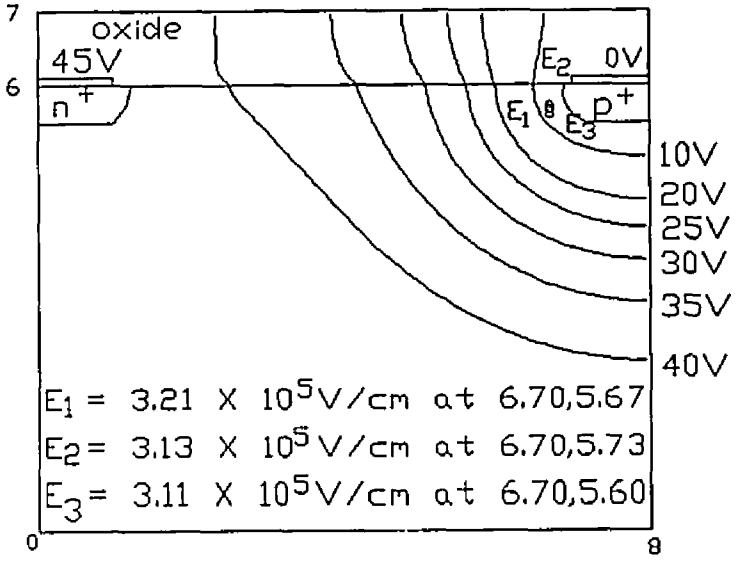

Figure 4.2 Equipotential contours for the bipolar power device without floating ring.

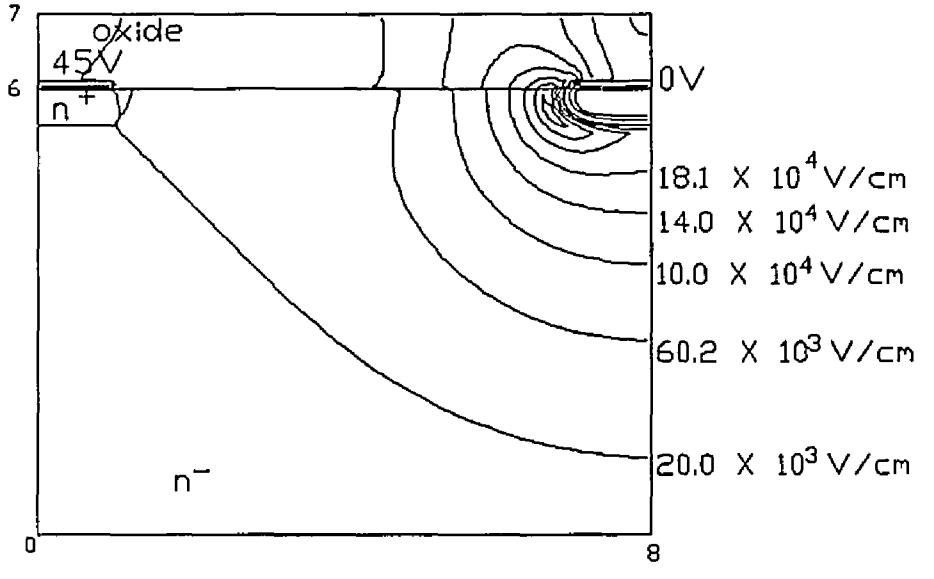

Figure 4.3 Equi-field contours for the bipolar power device without floating ring.

4.3 Simulation of a Power Device With a Floating Ring.

In order to improve the breakdown behavior of the above-mentioned device, a floating ring is diffused into the geometry as shown in Fig. 4.4. A two-dimensional simulation of a similar structure has been previously discussed by Boisson, *et al*[41]. The size and position of the floating ring can be optimized to produce lower values for the highest electric field points. In Fig. 4.5, the optimized position of the floating ring is shown. This single floating ring structure has pulled the equipotential contours away from the curvature of the *p-n* junction. Hence, the high field points no longer occur near the curvature of the *p-n* junction.

The equipotential contours are now crowding around the curvature of the floating region. Therefore, it creates a high concentration of equi-field contours around the curvature of the floating ring as seen on Fig. 4.6. The values of the high field points are found to be approximately $2.67 \times 10^5\ V/cm$, and therefore, they are lower than the device without floating ring. These values are well below the critical electric field of $3 \times 10^5\ V/cm$, hence, the breakdown behavior of this device is improved.

Fig. 4.6 shows the equi-field contours for this device. In an optimized design, the regions of high electric field should be equally distributed over the junction and the floating ring structures. This can be seen in Fig. 4.6. While the highest field points occurred all on the floating ring, the field points on the junction are of the same order of magnitude.

Notice that punch through occurs between the metallurgical junction and the floating field ring. Punch through creates a forward biased junction in which there would occur a current flow if there were any donors available. This punch through effect limits the voltage of the floating field ring to the voltage of the base

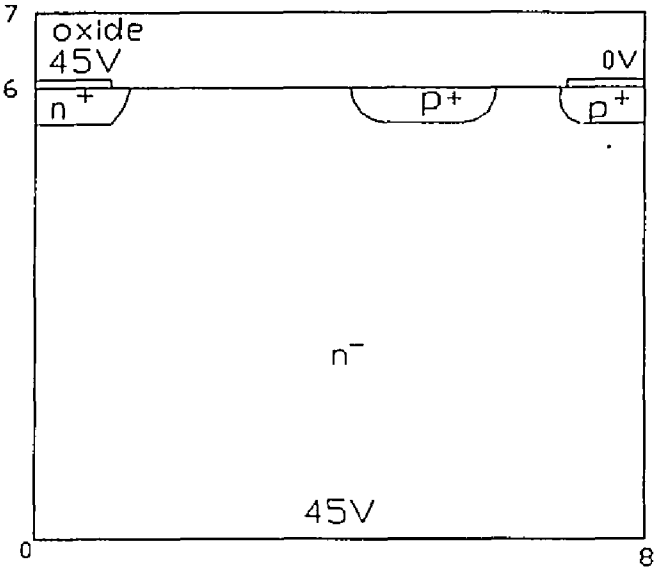

Figure 4.4 Bipolar power device with floating ring.

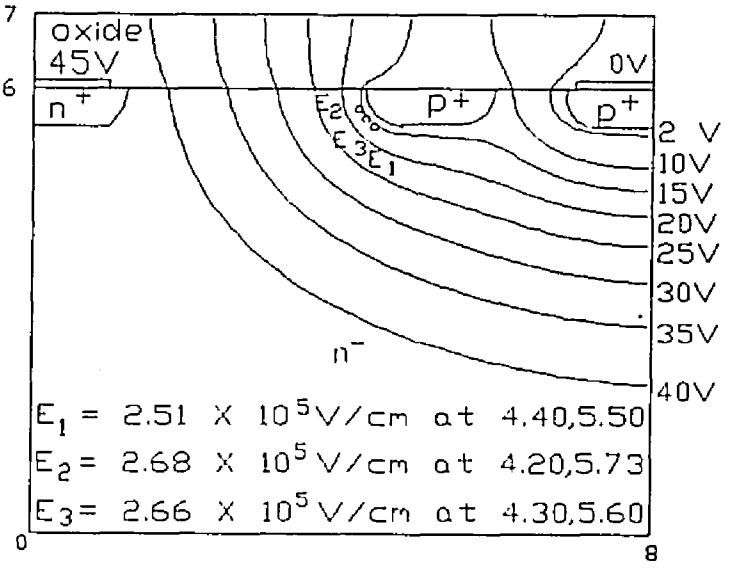

Figure 4.5 Equipotential contours for a bipolar device with floating ring.

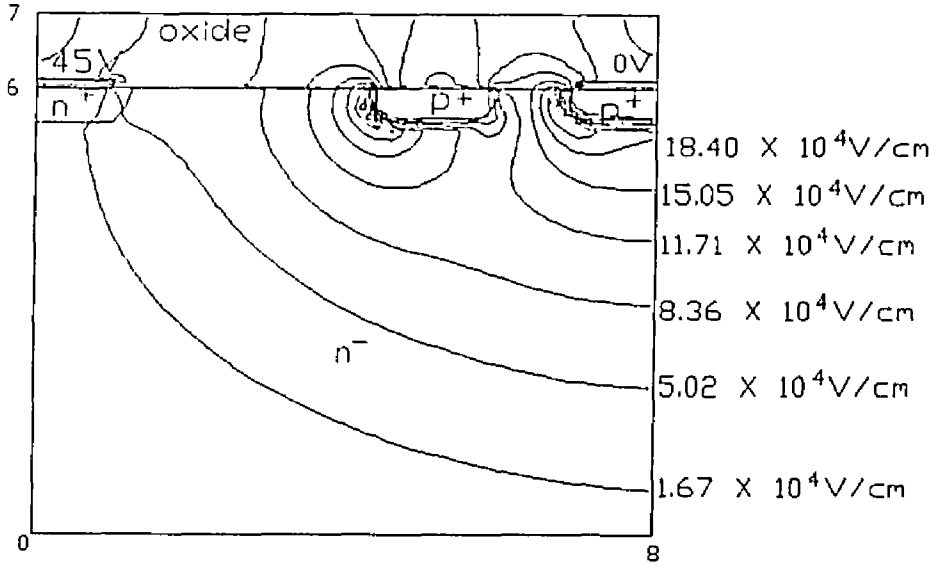

Figure 4.6 Equi-field contours for a bipolar device with floating ring.

contact plus the punch through voltage between the ring and the metallurgical junction. If the voltage of the ring grows higher, current will temporarily flow until the voltage has been reduced back to the punch through voltage. Since our device simulator, SEPSIP 2.0, does not take current flow into consideration, this phenomenon is not correctly modeled, and our simulation does indeed predict a voltage on the floating field ring which is higher than the punch through voltage. The thesis of my colleague, Davis[27], sheds more light on this problem.

4.4 Simulation of a Non-rectangular Device.

The slanted geometry of Fig. 4.7 is simulated to study the effect of the slanted edge on the breakdown behavior of the power device. Fig. 4.8 shows the equipotential contours if the substrate is biased at the high voltage (70 V). Fig. 4.9 shows the corresponding equi-field contours. Fig. 4.10 shows the equipotential contours if the substrate is biased at the low voltage (0 V). Fig. 4.11 shows the corresponding equi-field contours. The aim of this simulation was to study the effects of biasing the substrate in a dielectrically isolated structure.

It turns out that biasing the substrate at the high voltage works slightly better than grounding it. The highest values of the electric field are slightly lower in that case.

Grounding the substrate has a somewhat negative effect. Since the grounded substrate, in this technology, attracts a depletion region (as shown clearly on Fig. 4.10), there can be observed a stronger cross-coupling between devices in neighboring wells, that is: biasing the substrate on the high rather than on the low voltage helps with isolating neighboring devices from each other.

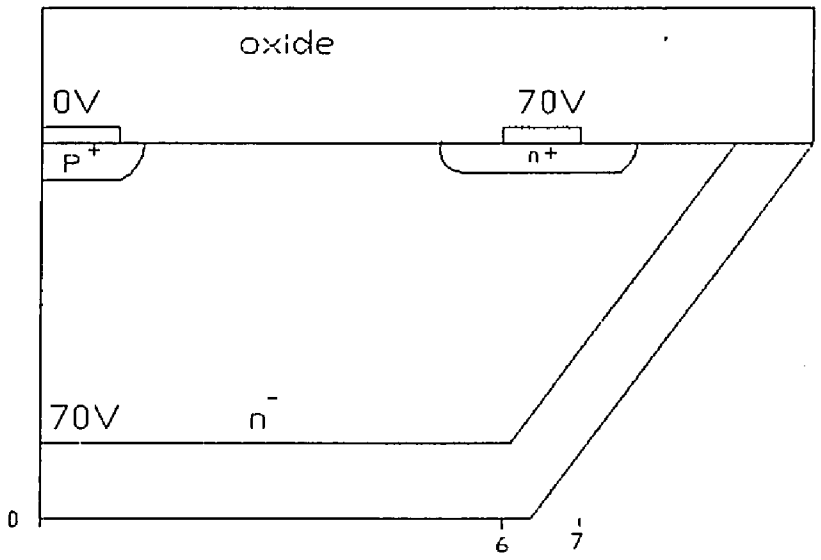

Figure 4.7 A non-rectangular power device.

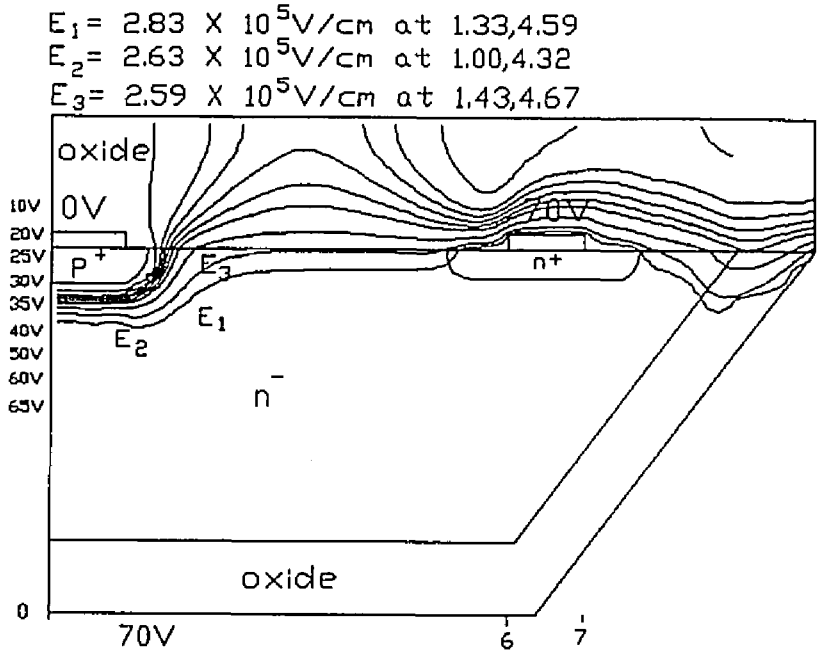

Figure 4.8 Equipotential contours for the non-rectangular power device (substrate = 70V).

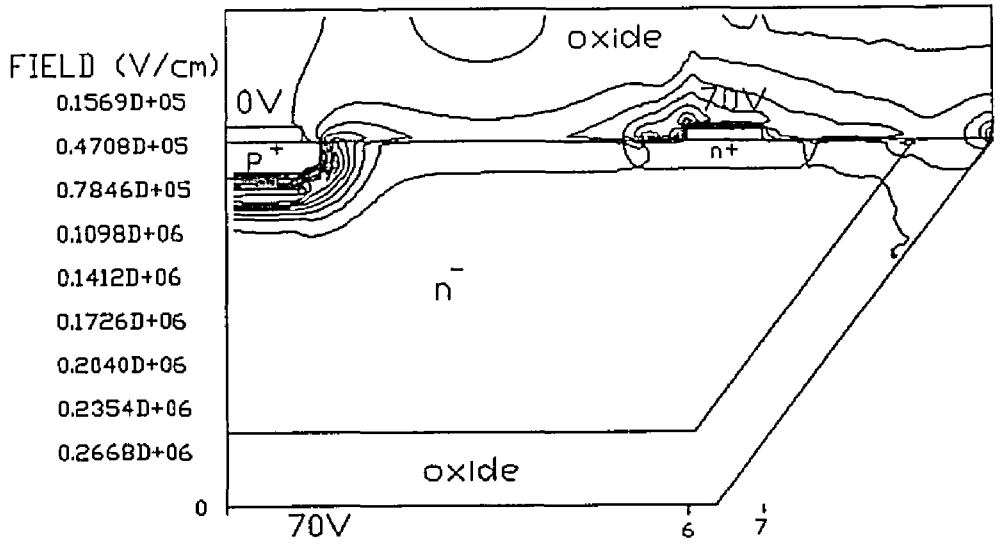

Figure 4.9 Equi-field contours for the non-rectangular power device (substrate = 70V).

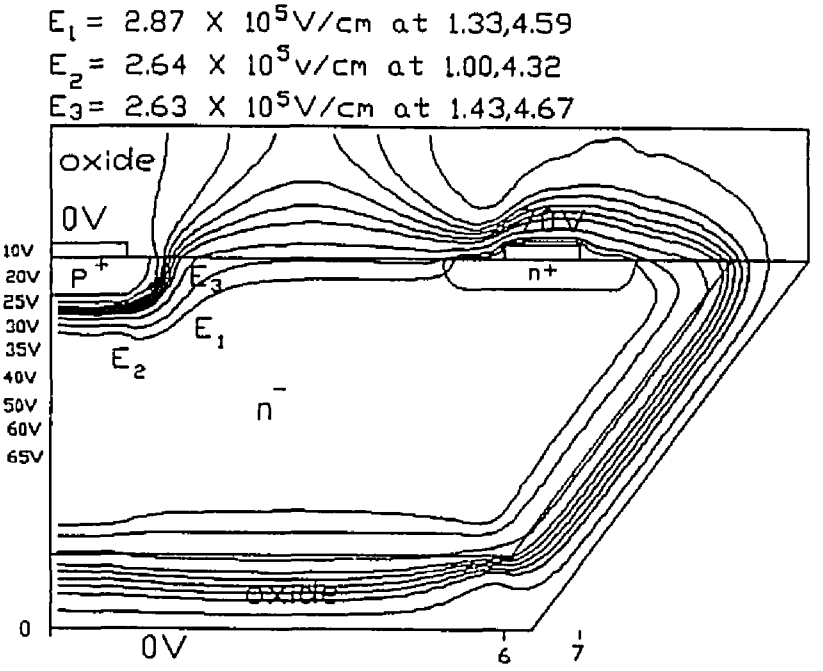

Figure 4.10 Equipotential contours for the non-rectangular power device (substrate = 0V).

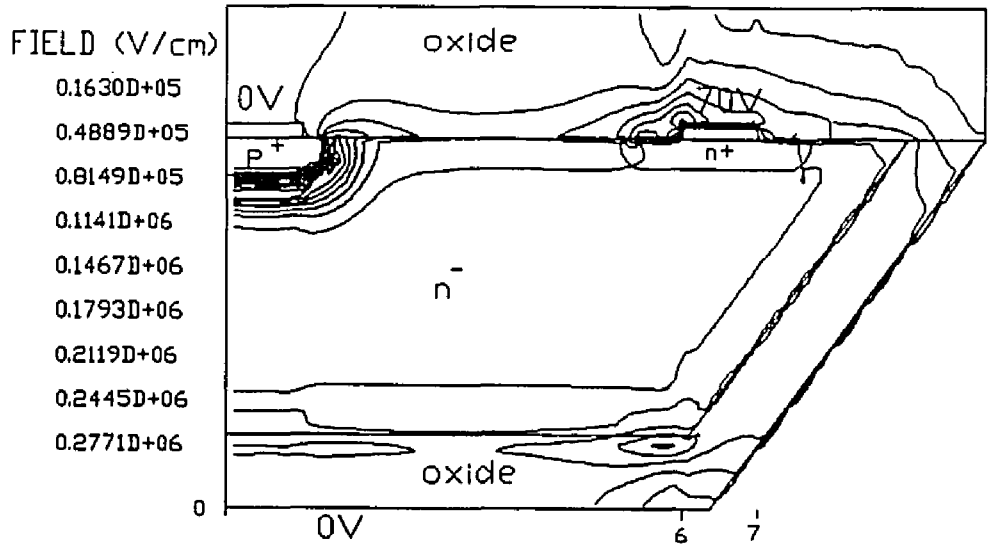

Figure 4.11 Equi-field contours for the non-rectangular power device (substrate = 0V).

CHAPTER 5
CONCLUSIONS

The study of the breakdown behavior of power semiconductor devices has become increasingly complex due to the enhanced complexity of today's commercial device structures. In order to maximize the breakdown voltage (i.e., in order to approach the theoretical limits imposed by a planar junction), more and more special junction termination structures have been added to commercial power devices, structures such as field plates and floating field rings.

The breakdown behavior of a power device depends not only on the physical geometry (the layout) of the semiconductor device structure, but also on other factors such as the background impurity concentration, the surface properties such as the oxide thickness and the influence exerted by a field plate, the influences of floating field rings on the junction field curvature, and finally, the influences of extraneous factors such as trapped interface charge[27]. It is difficult to determine the cumulative effects of all these factors on the breakdown behavior without the use of simulation. The two-dimensional numerical simulation software, SEPSIP 2.0, was developed to analyze the breakdown behavior of power devices taking all these factors into consideration.

The incorporation of floating structures and slanted edged geometries into SEPSIP has enhanced the capabilities of SEPSIP to simulate more general semiconductor power devices with arbitrary geometry. The ability of SEPSIP to display graphically the equipotential contours and the equi-field contours has greatly simplified the physical interpretation of breakdown behavior. This has been demonstrated

through simulation of a bipolar power device. The improvement in the breakdown behavior under the influence of a floating field limiting ring diffused into the bipolar device has been observed. In addition, the simulation of a non-rectangular device allowed the assessment of the effect of dielectric isolation, and enabled us to compare the effect of grounding/biasing the dielectric.

Additional features can still be added to the code. SEPSIP 2.0 does not provide a capability to analyze correctly the punch-through behavior between neighboring floating ring structures, and larger structures, such as power integrated circuits with cross-coupling between neighboring devices need to be simulated. It would also be very useful to analyze the interaction between a device simulator, such as SEPSIP, and a circuit analysis program, such as SPICE.

APPENDIX A

SEPSIP 2.0

User's Manual

A.1 General Overview of SEPSIP

The software simulator, SEPSIP, simulates two-dimensional power semiconductor devices under the high reverse-bias voltage condition. The main objective of SEPSIP is to enable users to simulate different device structure topologies in order to investigate their breakdown behavior. Furthermore, SEPSIP can be utilized to study various breakdown improvement techniques to optimize the design of semiconductor devices.

For a fast determination of breakdown behavior, SEPSIP computes the electric fields and records a number of high field points that can be compared against the critical field values. Values of high fields that are lower than the critical field indicate that the device operates below the breakdown condition. Interested readers are referred to previous research concerning the critical field for breakdown of semiconductor materials under various conditions[42].

SEPSIP is applicable to most types of device structures. However, in using the software, the user needs to specify the device topology such that is in agreement with the requirements of the boundary conditions that are predefined in SEPSIP. More specifically, the program requires that the entire bottom of the structure is biased at the same potential, and that there are no electric fields normal to the side boundaries[17]. These conditions pose a certain problem with respect to the simulation of devices with low background doping since such devices have large

depletion layers. SEPSIP will produce incorrect results if the simulated device topology is cut inside a depletion region.

A general program flowchart is displayed in Fig. A.1. As shown in the diagram, the command procedure *sepsip* oversees three separate major programs, namely *input*, *setup*, and *solve*. The initial stage, *input*, obtains data from an input file <*filename*>.*inp* (see section A.2), and from a number of data files that specify the device topology <*filename*>.*txt* (see section A.3.4) and the impurity concentrations <*filenames*>.*imp*. At the end of the *input* stage, all the given data are integrated into a single binary file, *sep.set*. This *sep.set* file becomes the input for the second stage, *setup*. In this stage, a system of difference equations across the domain of interest is being set up. Thereafter, the difference equations will be available and accessible for numerical iteration. The parameters for the difference equations are put in another binary file, *sep.sov*, which will be used by the final stage, *solve*.

In the final stage, *solve*, the iterative computation is solved by using a combined iteration scheme of Newton's Method of linearization and the method of Successive Over-Relaxation at each iteration step. The simulation outputs are presented in two forms: 1) tabular – in <*filename*>.*out* ; and 2) graphical – in <*filename*>.*dxf* for equipotential contours and *sep1.dxf* for equi-field contours. The graphical output files can be transferred to the personal computer in which simulation results can be viewed using the AutoCAD[43] graphic package.

The SEPSIP simulator is written in FORTRAN 77 and was initially developed in the VAX/VMS operating system environment. Unfortunately, SEPSIP was written in a way that made use of many special features of the VAX/VMS FORTRAN compiler. It was one of our tasks to remove all system dependencies, and to

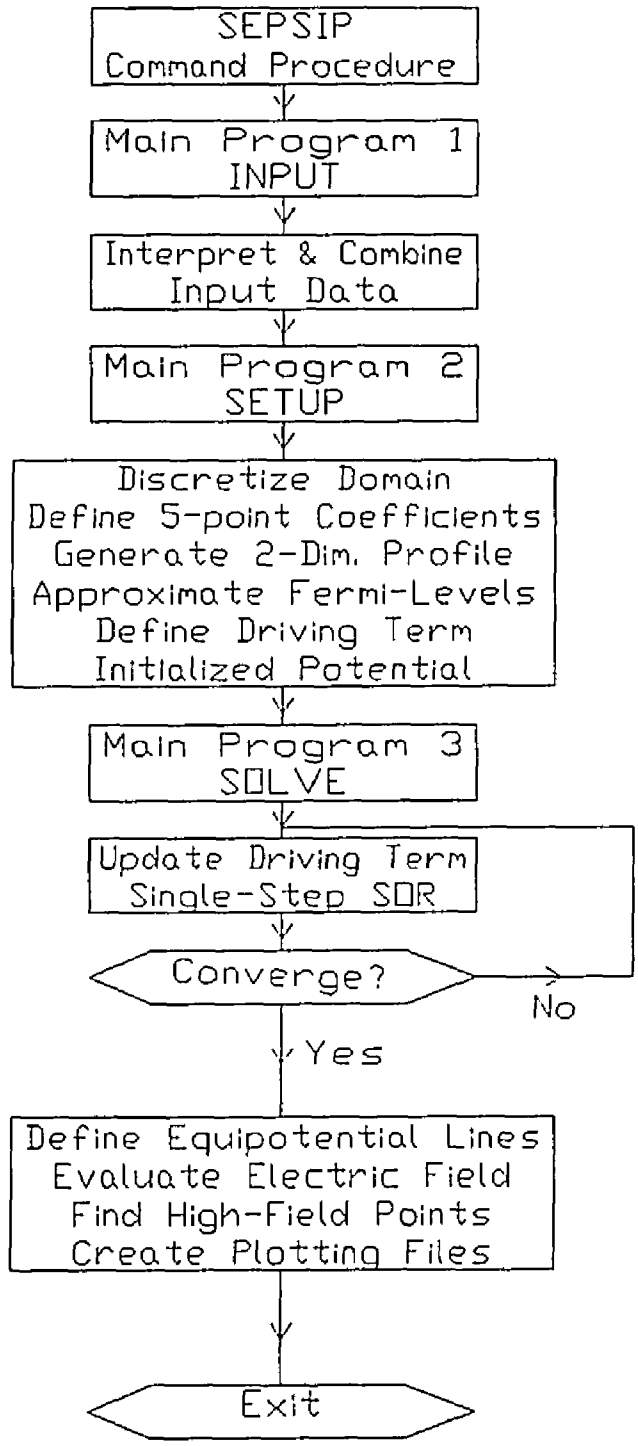

Figure A.1 Program flowchart for SEPSIP.

come up with a version of SEPSIP that could be easily ported to other hardware configurations. We ported SEPSIP successfully to a MS-DOS/MicroSoft Fortran environment. However, notice that, although the system dependencies were minimized, a successful installation and execution of SEPSIP on another system may still require certain minor changes in the program. In the following sections, information pertaining to the creation of the input files and installation considerations of the software will be discussed in detail.

A.2 Creation Of An Input File (extension: *inp*)

One of the user-specified data to the SEPSIP software is the input file called *<filename>.inp*. This file is made up of a set of statements, written in the following general form

$$< statement > \qquad < parameter >=< value > \cdots$$

where '$\cdots$' means that more parameters may be added. A statement in the input file can be used to provide the program with information concerning the device simulation or be used to request the program to perform certain actions.

The statement name is specified in a free format with one or more spaces apart from its first parameter. Spaces are also used to separate the parameter assignments if more than one parameter is stated on the same line. Other delimiters are not allowed throughout the input file. There is no restriction on where the spaces should appear in a statement line, however, spaces are not allowed within a name string.

This simulator requires that all the statement names and their parameter names be spelled using lower case characters precisely as specified in this manual.

Abbreviations are not permitted. Any unrecognized spelling causes either the program to terminate, or the statement to be ignored depending on the severity of the error condition.

No statement line can exceed 72 characters. Abortion of the program will occur if any line is longer than this limit. If a statement requires more than 72 characters, it must be continued on a separate line using a continuation delimiter. This continuation delimiter is a plus sign which must be specified on the first column of the continuation line. Alternatively, the statement name can be repeated followed by the remaining parameters. Successive lines with the same statement name are equivalent to a single statement possibly continued over several lines.

All values assigned to the parameters must be of valid FORTRAN data types. The required data type for each parameter is indicated in this document by an italic string placed between angular brackets. If more than one value is assigned to a parameter, the most recently assigned value will be used. Default values will be assigned to those parameters that are not specified by the user. The default values are stated in this manual (cf. also source code *inblk.for*). A sample of the input file for a non-rectangular geometry is presented in Fig. A.2.

The input file consists of eight input statements. They are *device, geometry, profile, grid, material, bias, print,* and *plot.* A detailed description of these input statements and their parameters follows.

DEVICE statement

Format:

device t = <*real*> qss = <*real*> fims = <*real*>

Parameters:

t device operating temperature in Kelvin. default is $300K$.

```
device      qss=1.0E+10     fims=0.0   t=300.0

geometry    dwgfil = 'fig44.txt'       scale = 4.0
+           leftang = 54.0

profile
+           supfil(1)='fig441.imp'
+           supfil(2)='fig442.imp'
+           supfil(3)='fig443.imp'

grid
+           xmaxpt = 200     ymaxpt = 200

material    mattyp(1)='Si'          mattyp(2)='SiO2'

bias
+           volt(1)=100.0    volt(2)=0.0
+           vsub=100.0

print       detail=.false.   printg=.true.   printc=.false.
+           printv=.false.   printe=.false.   printm=.true.
+           breakv=.false.   intface=.true.

plot        plotv=.true.   plote=.true.
+           nlevel=8         nspot=3
+           vlevel(1)=10.0   vlevel(2)=20.0 vlevel(3)=30.0
+           vlevel(4)=40.0 vlevel(5)=50.0 vlevel(6)=60.0
+           vlevel(7)=80.0 vlevel(8)=90.0

stop
```

Figure A.2 Example of an input file for a non-rectangular geometry.

qss semiconductor surface charge density in cm^{-2}. default is 0.

fims work function difference between metal and semiconductor in $Volts$. default is 0.

The device statement specifies the physical parameters associated with the device.

GEOMETRY statement

Format:

geometry dwgfil = <*char*>.*txt* scale = <*real*>

\+ leftang = <*real*> riteang = <*real*>

Parameters:

dwgfil drawing file name which represents the device geometry.

scale number of micrometers in each drawing unit. default is $1\mu m$.

leftang left angle value if left side is slanted. default is 90^o.

riteang right angle value if right side is slanted. default is 90^o.

The device simulator, SEPSIP, extends its simulation beyond the semiconductor surface. Hence, the actual device geometry can be quite complex. However, SEPSIP does not require the user to specify the device geometry in many different parameters, instead, it uses a simpler approach to request a drawing file <*filename*>.*txt* of fixed format. With this simplification, only three types of blocks are defined in the software: (1) semiconductor bulk (abbreviated *bulk*), (2) dielectric insulator (abbreviated *insu*), and (3) electric conductor (abbreviated *cond*). The building blocks used by SEPSIP are rectangular in shape with differing sizes and aspect ratio. Each building block has an integer pointer which points to its respective array specified in the input file, <*filename*>.*inp*. For example, if the

pointer *i* of a *bulk* (second digit for non-rectangular geometry) or an *insu* block corresponds to the *i*th element of the material array *mattyp* (see *material* statement), this indicates that the block is made of that material. With regard to electric potential, conductors are assumed to be perfect. Therefore, the bias value is used to characterize a conductor. The index *i* of a *cond* block corresponds to the *i*th entry of the bias value array *volt*, indicating that the given conductor is biased at the value specified by the *i*th element of *volt*.

Another separate drawing unit existing in the software is called an impurity zone (abbreviated *impu*). This drawing entity is also related to the device topology. Each impurity zone is a rectangular region in which its impurity distribution is described by a one-dimensional profile in the vertical direction. In using the program, the two-dimensional impurity distribution of each impurity zone is required to be broken down into several one-dimensional impurity zones indexed by an integer *i*. Overlapping of these impurity zones in order of *i* assimilates the fabrication and diffusion process, and gives the required two-dimensional impurity profile of the semiconductor device. The block pointer *i* of the impurity zone is also directly related to the *i*th element of the doping file name array *supfil* in the *profile* statement.

With regard to the *geometry* statement, the size of *char* for parameter *dwgfil* must be less than or equal to five characters if SEPSIP is executed in the PC, whereas for the VAX/VMS version, this restriction does not apply. The *scale* parameter allows a larger geometry to be scaled down to a size that can be fitted into the drawing. For example, the actual geometry with a box of size $100 \times 50 \mu m$ in x and y direction, respectively can be represented by a drawing box of 10 by 5 with a *scale* equal to 10. If the scale is greater than 1, then the *depth* in the impurity profiles, *<filenames>.imp* (see *profile* statement) must be the actual

physical depth of the device. The software will take the *depth*, and divides it by the *scale* factor during the conversion from a series of one-dimensional impurity profiles to a two-dimensional doping profile.

An example of a drawing file is shown in Fig. A.3. This file is created during SEPSIP's drawing session to be elaborated in the next section. Each data record corresponds to a rectangular building block of the drawing. The total number of records is equal to the total number of rectangular boxes drawn by the user. Each record has six fields:

Field	column	type
name	1- 4	character
x1	5-14	real
y1	15-24	real
dx	25-34	real
dy	35-44	real
ptr	45-48	integer

where name = name of the block,

x1 = x-coordinate of lower left corner of the block,

y1 = y-coordinate of lower left corner of the block,

dx = x-dimension of the block,

dy = y-dimension of the block,

ptr = pointer of the block.

The x-coordinate of the upper-right corner can then be computed to be $x1 + dx$, and likewise $y1 + dy$ specifies the y-coordinate of that corner.

```
INSU   0.000000   5.000000 10.000000   1.800000    2
INSU   0.000000   0.000000 10.000000   5.000000    2
BULK   0.000000   1.000000  9.000000   4.000000   11
IMPU   0.000000   1.000000  9.000000   4.000000    1
COND   0.000000   5.000000  1.000000   0.200000    1
COND   6.000000   5.000000  1.000000   0.200000    2
IMPU   0.000000   4.500000  1.000000   0.500000    1
IMPU   5.500000   4.500000  2.000000   0.500000    1
```

Figure A.3 An example of a drawing file for non-rectangular geometry.

```
Net Active

depth     concentration
0.0000 -0.100000E+18
0.1000 -0.100000E+18
0.2000 -0.100000E+18
0.3000 -0.100000E+18
0.4000 -0.100000E+18
0.5000 -0.100000E+18
0.5000  0.100000E+16
0.6000  0.100000E+16
0.7000  0.100000E+16
0.8000  0.100000E+16
0.9000  0.100000E+16
1.0000  0.100000E+16
1.0000  0.100000E+16
1.1000  0.100000E+16
1.2000  0.100000E+16
1.3000  0.100000E+16
1.4000  0.100000E+16
1.5000  0.100000E+16
1.6000  0.100000E+16
1.7000  0.100000E+16
1.8000  0.100000E+16
1.9000  0.100000E+16
2.0000  0.100000E+16
2.1000  0.100000E+16
2.2000  0.100000E+16
  ...
```

Figure A.4 A typical impurity concentration file.

PROFILE statement

Format:

profile supfil(i) = <*char*>.*imp*

Parameters:

supfil array of doping file names. index i ranges from 1 to 10.

The number of characters for *char* of the *supfil* array must be less than or equal to five if SEPSIP is simulated in the PC. This restriction does not apply to SEPSIP running on the VAX/VMS. A typical impurity concentration file is shown in Fig. A.4. These files can be generated by the device process simulation software SUPREM[44], however, the example file shown in Fig. A.4 is generic and has been entered manually. The data file shown is from the third impurity zone of the example in Fig. 4.1 of Chapter 4. Each data line is made of two fields: the depth specified in μm, and the concentration specified in cm^{-3}.

SUPREM assumes its vertical coordinate to start at the surface, and increase downward to the bottom of the device. This is opposite to the coordinate convention that is used internal to SEPSIP. In order to be able to directly use SUPREM output files, SEPSIP's one-dimensional doping input files make the same assumption as SUPREM, i.e. $y = 0\mu m$ denotes here the surface of the semiconductor. SEPSIP will then automatically reverse the vertical coordinate information during the input data processing in order to make it consistent with the internally used convention according to which $y = 0\mu m$ denotes the lower boundary of the simulated region. i.e. the 'bottom' of the device.

Multi-layered impurity profiles may exist in a doping concentration file. However, the simulator only extrapolates the lateral diffusion for the first layer, using an aspect ratio of 0.8. Thus, the doping distribution must be disintegrated in

such a way that each concentration file consists of only one new impurity layer with respect to the background. Junctions that are deeper, i.e., that have been diffused earlier, must be specified first (must carry a lower index).

The bottom of an impurity zone is more flexible than the other three sides. SEPSIP maps the profile to the semiconductor bulk all the way down until either hitting the bottom of the bulk or reaching the end of the concentration file.

GRID statement

Format:

grid	xmaxpt = <*int*>
+	ymaxpt = <*int*>

Parameters:

xmaxpt	maximum number of grid nodes in the horizontal direction.
ymaxpt	maximum number of grid nodes in the vertical direction

The default number of grid nodes is 200 for either direction. This is also the largest number of grid points currently supported by SEPSIP.

MATERIAL statement

Format:

material	mattyp(i) = <*char*>

Parameters:

mattyp	array of material names. index i ranges from 1 to 10.

This *material* statement allows the user to specify an array of different semiconductors and insulators. The default values are 'Si' and '$SiO2$' for mattyp(1) and mattyp(2), respectively. The following are the possible semiconductor and

insulator materials that can be simulated by SEPSIP:

Si	SiO_2	Si_3N_4	Ge
AlSb	GaSb	GaAs	InP
CdS	CdSe	CdTe	PbS

When a drawing entity of a device configuration is *bulk* or *insu*, the material used to make up the entity must be named in this *material* statement. The index *i* of such an entity corresponds to the *i*th material in the array *mattyp*.

BIAS statement

Format:

bias volt(*i*) = <*real*> vsub = <*real*>

Parameters:

volt array of explicit biases associated with conductors.
index i ranges from 1 to 10.

vsub implicit bias along the bottom of the structure.

In this manual, the known bias of a conductor which is explicitly drawn is referred to as an explicit bias, while the bias not in association with any conductor is considered as the implicit bias. All of the biases in *volt* are considered as explicit biases. In drafting the conductor blocks, the bias values are not directly specified. However, a pointer *i* assigned to each conductor block indicates that the *i*th value of the bias array *volt* is the bias on that conductor.

The notion of implicit bias is unique to SEPSIP. All biases in a practical device should be 'explicit'. However, there is a good reason why we allow some biases to be implicit. A simulation running over the whole device structure is often not economical because of computational cost and time. Also, it may not be

accurate. Therefore, experienced device engineers select only an area of interest to be the simulation domain. Very often, important bias conditions may not be within the effective specified domain. Consequently, SEPSIP supplies a parameter *vsub* as the bias value along the bottom of the simulation domain to define the implicit bias condition.

PRINT statement

Format:

print	printg = <*log*>	printm = <*log*>
+	printv = <*log*>	printe = <*log*>
+	printc = <*log*>	breakv = <*log*>
+	intface = <*log*>	

Parameters:

printg	flag to print grid structure.
printm	flag to print device topology.
printv	flag to print potential distribution.
printe	flag to print electric field strength.
printc	flag to print impurity concentration.
breakv	flag to compute the ionization integral and output it to <*filename*>.*pth*.
intface	flag to include the interface in the high field computation.

All of the above parameters are of the logical data type with *false* default value. When *printg* is set, the software prints out two sets of grid points in the output file named <*filename*>.*out*. One set is printed in the x-direction and the other in the y-direction, accompanied with their respective grid spacings. The grid

points are numbered from the left to the right, and from the bottom to the surface in increasing order. The grid is automatically generated and is non-uniform in general. Since a smoothly varying grid structure is essential to the simulation, it is essential that the generated grid be checked carefully for smoothness.

Another useful flag is *printm*. This flag is responsible for writing the material distribution to the output file. The drawing entities are filled with their respective index values at the occupied grid points. The material distribution output consists of 2 two-dimensional arrays of integers. In the first array, conductors are represented by negative numbers to mark all conductor regions clearly, and to make them distinguishable from semiconductor or oxide regions with the same indices. Material interfaces are marked by 100 or 101 (horizontal and vertical), the intersections of two perpendicular interfaces (corners) are marked by 200, and the interfaces at slanted edges are marked by 105. This 'drawing' does not contain the junction information at all. In the second 'drawing', all regions but the semiconductor region are specified with negative numbers. In the semiconductor region, the bulk is now shown with its corresponding impurity index (the impurity region with the lowest index always coincides in shape with the semiconductor bulk). Overlaid are then the various junctions shown with their lateral diffusions. Regions with higher indices override regions with lower indices, another reason why the deeper diffused regions must be specified with a lower index.

When *printv* is set, the software prints out the potential values of all the grid points in the same format as *printm*. The same goes to *printe*, but this time the electric field values are printed. Finally, when *printc* is set, the impurity concentration is printed in the same manner as *printm*.

PLOT statement

Format:

plot	plotv = <*log*> plote = <*log*>
+	nlevel = <*int*> nspot =<*int*>
+	vlevel(i) = <*real*>

Parameters:

plotv	flag to plot equipotential contours. default is *.false.*
plote	flag to plot equi-field contours. default is *.false.*
nlevel	number of contours to be plotted. default is 5.
nspot	number of high field spots to be plotted. default is 1.
vlevel	specified potential values for equipotential contours.

The maximum number of contour lines in the plot file is *nlevel* = 9. The index *i* of *vlevel* goes from 1 to *nlevel*. The default values for *vlevel*(*i*) are computed by SEPSIP using equidistant spacing between the highest bias value and the lowest bias value. Up to four plot files can result from any SEPSIP simulation. If *plotv* = *.true.*, the equipotential contours are stored on file <*filename*>*.dxf* using the AutoCAD interchange file[43]. If *plote* = *.true.*, file *sep1.dxf* contains the equi-field contours specified in the same format. The method of transferring these files to the AutoCAD environment, will be elaborated in the next section. The other two files are generic hardware independent plot files, *sep.tmp* for the equipotential contours and *sep1.tmp* for the equi-field contours. These two files allow the results to be plotted using graphics software packages other than AutoCAD. A subroutine is needed to convert the results in these files into the format outlined by that graphics software. The information concerning the arrangement of these two files can be found in the subroutine *wrdxf*.

A.3 Specifying the Device Topology (extension: txt)

The graphic software package AutoCAD is used by the SEPSIP simulator in two ways: (1) to generate a compact data file that represents the device configuration, and (2) to display the graphical contours onto the device topology after the simulation has been completed. This section is intended to provide information necessary to create graphically the input file, *<filename>.txt*. For clarification, the user's responses to AutoCAD's prompts will be underlined. The return key is depressed after every response. An underlined italic string quoted in angular brackets indicates the response.

A.3.1 Drawing Session Using AutoCAD

To start the drawing session, log into the directory where the graphic software package, AutoCAD is installed, and enter:

acad

in response to the operating system's prompt. The AutoCAD's directory can also be placed in the operating system's search path. Then it is possible to execute AutoCAD from any directory. The following files: *format.txt*, *sepsip.mnx*, *sepsip.mnu*, and the prototype drawing, *sepsip.dwg* must also be present in the same directory where the drawing is in session.

After the software is loaded, AutoCAD's Main Menu appears on the text display screen. The menu display looks like

0. Exit AutoCAD
1. Begin a New Drawing
2. Edit an Existing Drawing
3. Plot a Drawing

...

To begin a new drawing, select Main Menu task 1. AutoCAD will then ask for the name of the drawing to be created. The dialogue proceeds as follows,

Enter Selection: 1

Enter Name of Drawing: < *newname* >= sepsip

This originates a new drawing the name of which is specified in place of < *newname* >, with its initial environment copied over from the prototype drawing file, *sepsip.dwg*. Note that < *newname* > cannot have more than five characters (numeric or alphabetic) if the simulation is executed in the PC.

Alternatively, an existing drawing can be retrieved for editing. By selecting Main Menu task 2, the existing drawing can be revised or modified. AutoCAD will again ask for the name of the drawing. Since the drawing has been previously named, the user does not need to specify the prototype drawing. The dialogue is similar to the previous case:

Enter Selection: 2

Enter Name of Drawing: < *oldname* >

Following this dialogue, AutoCAD's Drawing Editor is automatically loaded and the customized screen menu, *sepsip.mnu* for SEPSIP's application is shown on the monitor. Using a digitized Tableau with a lighted pen or a Mouse, this menu permits the user to enter commands by pointing to a menu item on the screen.

The customized screen menu of SEPSIP has a main menu and three sub-menus. The main menu displays the submenu names. When a particular submenu is selected, the screen displays all the menu items in that group.

A.3.2 SETUP Submenu

Upon selecting the SETUP submenu, the menu items of this submenu will appear between two lines of '********' in the screen menu area. The following describes the function of each menu item.

COORD, ^COORD

This pair of items turns the coordinate system on and off. By default, the coordinate system is on, and the x and y coordinates are shown near the top of the screen. One unit of the coordinate system represents $1\mu m$ in length. In addition to using the menu item, the 'F_6' key can be used to toggle the coordinate system.

GRID, ^GRID

AutoCAD has the ability to display a background reference grid with a spacing of 0.2 unit length when the item GRID is activated. Hence, ^GRID (default) inhibits the reference grid to be displayed as background on the screen. Also, the 'F_7' key can used to turn the background reference grid on and off. It should be noted that there is no relationship between the GRID item in this drawing session and the *grid* statement in SEPSIP's input file.

SNAP, ^SNAP

Movements on the drawing area can be synchronized into alignment with a rectangular grid cursor by a *snap* mechanism. Activation of SNAP (default), will cause the screen crosshairs and the drawing coordinates to move in synchronization, together with the nearest grid point. When ^SNAP is selected, the effect of SNAP is off. The 'F_9' key can also be used to toggle between these two features.

The snap resolution can be varied. The default snap resolution used in this drawing session has been set to 0.1 unit length. To change the snap resolution, type *snap* and press the return key at the command prompt. The dialogue is as follows:

Command: *snap*

snap resolution <0.1000>: *< real >*

The user can set the numerical value to *real* to get the required resolution.

FILL, ^FILL, REGEN

The device geometry drawn with the menu items of BULK, INSU, and COND can be filled with solid colors. This feature is useful in visualizing which rectangular boxes represent which device topologies. In the default state, the FILL mode is *on*. To cancel the FILL mode, select ^FILL, followed by REGEN. To return to the default state, select FILL, followed by REGEN. AutoCAD will regenerate the entire screen and redraw the whole device configuration when the REGEN item is chosen.

ZOOM

This item is an AutoCAD predefined command. As the name implies, it works with the same principle as the zoom lens attached to a camera. Hence, it enables the user to enlarge or reduce the apparent size of the pictorial entities on the screen without changing the actual representation of the device structures. When the ZOOM command is activated, there are many options that can be selected. The most commonly used option is the Window (W). The window option permits the user to view a selected portion of the drawing entities by pin pointing two diagonal corner points of a rectangular window. The center of the new display area is the center of the selected rectangular window. The drawing inside the window is

enlarged or reduced to fill the display. To return to the previous drawing display, activate the ZOOM command and select the P option.

PAN

This is another AutoCAD internal command. The PAN menu item allows the user to view another portion of the drawing display, without enlarging or reducing the drawing entities. This command enables the user to view a display area that was 'off screen' prior to selecting the PAN command. Upon issuing the PAN command, the user has to enter the numerical x and y coordinates separated by comma, to specify the relative displacement of the current drawing.

A.3.3 DRAW Submenu

SEPSIP assumes that the interfaces and boundaries of the device topology are either parallel or perpendicular to each other. With this assumption, the semiconductor device structure is made up from a combination of contiguous rectangular boxes. Combining this simplification with the flexibility of AutoCAD led to the following customized menu items for drawing the device topology.

BULK, INSU, COND, IMPU

These four menu items are required to draw the device structure. As stated earlier, each of the four items is responsible for drawing a particular type of blocks as itemized below

BULK	semiconductor bulk
INSU	dielectric insulator
COND	electric conductor
IMPU	impurity zone

When one of these menu items is selected, it activates AutoCAD's INSERT command that places a predefined block at a position specified by the user. The command dialogue proceeds as:

Insertion point: *< lower left corner >*

X-scale factor: *< upper right corner >*

Block pointer: *< integer pointer >*

When the user specifies the two diagonal corner points of the rectangular box, the box is drawn on the screen. At the same time, an integer pointer is requested. Every block must have one pointer associated with it. For *bulk* and *insu* the pointer *i* corresponds to the *i*th element of the material array, *mattyp* stated in the *material* statement of the input file, *<filename>.inp*. The same material (index) can be assigned to more than one region. To draw a non-rectangular geometry, activate the item *bulk*. Then, when specifying the block pointer, an additional prefix to the integer pointer is required. A prefix of '1' is added to the block pointer if the device is slanted inward from the top to the bottom. A prefix of '2' is added to the block pointer if the device is slanted outward from the top to the bottom. The angle of the slanted side is specified in the *geometry* statement of the input file. If both sides are slanted, both angles must be equal. The allowed range of the angles is between 30 degrees and 60 degrees inclusive for a slanted edge, and 90 degrees for a non-slanted edge.

For the *cond* block, the pointer *i* corresponds to the *i*th element of the bias array, *volt*, stated in the *bias* statement of the input file. The same *i*th element of bias array can also be used by more than one *cond* block.

Specifying the *impu* block is somewhat different from the former three cases in three ways. First, all impurity zones need to have distinguishable indices, although the same concentration file may be used by several of the impurity zones. For the *impu* block, the index i corresponds to the ith element of the *supfil* stated in the *profile* statement of the input file. An impurity profile can be used more than once, however, the file name must then be re-specified in the array, *supfil*. Second, the numerical pointers are ordered in accordance with the process of fabricating the device. If two *impu* blocks overlap, then the impurity zone with the higher index value will override the zone with the lower index value. Third, the depth (y-coordinate) of the *impu* block is irrelevant. This information is being overridden by the information stored in the impurity profile.

ERASE, REDRAW

To change the device structure of an existing drawing or to delete any entity from the drawing, the menu item ERASE is used. It permits the user to select the entities to be removed from the drawing. After selecting the REDRAW item, the current display is redrawn, and all unnecessary marker blips are removed.

A.3.4 DATA Submenu

The DATA submenu is the third group of menu items. These menu items can be used to transfer, display, and edit the data.

EXPORT

The EXPORT item makes use of AutoCAD's ATTEXT command. This item must have a template file named *format.txt* in the current directory. This file predefines the arrangement of the resulting data structure after exporting the drawing. The effect is similar to that of a format statement in FORTRAN.

When EXPORT is selected, the information regarding the semiconductor device configuration during the action of BULK, INSU, COND, and IMPU is extracted from AutoCAD's database and is transferred to a disk file. The file name is chosen by the user, however, the program assumes an extension of *txt* for the file name. Note that the number of characters of this file name must be less than or equal to five if SEPSIP is to be executed in the PC.

IMPORT

After the execution of SEPSIP is completed, the simulation outputs are recorded onto a graphic interchange file (extension: *dxf*). The equipotential contours are being stored in the plot file, *<filename>.dxf*. The equi-field contours are being stored in the plot file *sep1.dxf*. To display one of these files, the IMPORT item is activated. This item is supported by AutoCAD's DXFIN command. After the item IMPORT is selected, AutoCAD will request the name of the plot file to be supplied. The default name for the plot file is the name of the current drawing. The user can provide another name, but the file extension of *dxf* is mandatory.

PLOT

AutoCAD's PLOT command is not related to SEPSIP's *plot* statement. The action taken by PLOT is to make a hard copy of the current drawing. The PLOT command provides several options for plotting a drawing. The easiest way to plot a drawing is to select the W (window) or the D (display) option in response to AutoCAD's prompt.

DISP, ^DISP

Normally, the pointers representing the device structures and the impurity zones are not displayed on the screen. This default setting can be overridden by DISP. When this item is selected, AutoCAD regenerates the current drawing. As a consequence, the values of the block pointers representing the device are shown on the screen. These values can be modified by selecting the EDIT item. To return to the default setting, activate the ^DISP item.

EDIT

The EDIT item is used for index editing. This item simplifies AutoCAD's ATTEDT command by responding to most prompts internally. The item is to be used in combination with the DISP item. When the EDIT item is activated, the value of the pointer can be modified. For this purpose, the current value of the pointer, and the new value of that pointer will be requested. The prompts appear on the screen in the following sequence:

Select Attributes: < *pointer* >

String to change: < *current value* >

New string: < *new value* >

A.3.5 Exit From AutoCAD

The user needs to exit from the Drawing Editor before exiting from AutoCAD. To exit the Drawing Editor, select the END item at the bottom of the screen menu area. Upon exiting the session, an AutoCAD drawing file, *<filename>.dwg* is created, and AutoCAD's Main Menu appears on the monitor. Enter 0 in response to the selection prompt to exit from AutoCAD. After successfully exiting from AutoCAD, the operating system's prompt reappears on the screen.

To save the current drawing under a different drawing name, the command SAVE is typed at the command line. AutoCAD will then request a new drawing name. To exit the Drawing Editor without creating a drawing file, the command QUIT is used instead of END.

A.4 Examples of Drawing File Preparations

In the last two sections, the parameters of the input statements of SEPSIP and the drawing commands of AutoCAD customized for SEPSIP were introduced. Now, an example is presented to demonstrate the use of these statements and commands. The Bipolar planar structure without floating field rings is chosen as the example (see Fig. 4.1).

To begin a drawing of the device structure, follow the procedure as described in Sec. A.3.1. Then, draw the device configuration using the following blocks by invoking their respective menu items:

bulk	pt1 = (0.0, 0.0)	pt2 = (8.0, 6.0)	ptr = 1
insu	pt1 = (0.0, 6.0)	pt2 = (8.0, 7.0)	ptr = 2
cond	pt1 = (0.0, 6.0)	pt2 = (1.0, 6.1)	ptr = 1
cond	pt1 = (7.0, 6.0)	pt2 = (8.0, 6.1)	ptr = 2
impu	pt1 = (0.0, 0.0)	pt2 = (8.0, 6.0)	ptr = 1
impu	pt1 = (0.0, 0.0)	pt2 = (1.0, 6.0)	ptr = 2
impu	pt1 = (7.0, 0.0)	pt2 = (8.0, 6.0)	ptr = 3

where *pt*1 stands for the lower left corner, *pt*2 stands for the upper right corner, and *ptr* denotes the index of the block. Note that for item *bulk*, the *ptr* must be specified as 11 for an inward slanted edge and 21 for an outward slanted edge (for a non-rectangular device). To make sure that the indices are correct, the menu items DISP and EDIT of the DATA submenu are used. The next step then is to select

EXPORT to create the drawing file shown in Fig. A.5. The name of the file used in this example is *nring* with the extension *txt*.

The impurity distribution of this device has been divided into three one-dimensional impurity zones. Hence, three separate doping files are required. These files are created manually and are named *nring*1 through *nring*3 (extension: *imp*) repectively. Fig. A.4 presents the beginning part of the *nring*3 file representing the third impurity zone.

The input statement file for this example is displayed in Fig. A.6. As indicated by the pointer values of *bulk* and *insu*, the material name array consists of two entries: $mattyp(1) = 'Si'$, and $mattyp(2) = 'SiO2'$. Two conductors have been drawn corresponding to the base and the collector of the device structure. The base voltage is at ground potential, and the collector is biased with an applied voltage of 45 $Volts$.

Assume the input file is named *nring* with the extention *inp*. The input file, *nring.inp*, the drawing file, *nring.txt*, and the three impurity files, *nring1.imp* through *nring3.imp* must be in the same directory where the simulation will be executed. To begin the simulation, enter

sepsip nring

at the system's prompt. After the simulation is completed, the output file, *nring* (extension: *out*), is created. The contents of this output file depends on the values of the logical flags set by the user in the *print* statement.

A.5 Installation and Execution of SEPSIP

The device simulator, SEPSIP, is a medium size program. In order to facilitate the debugging and to minimize the memory requirements, the software has been divided into three separate major programs. The source files of each

```
BULK   0.000000   0.000000   8.000000   6.000000    1
INSU   0.000000   6.000000   8.000000   1.000000    2
COND   0.000000   6.000000   1.000000   0.100000    1
COND   7.000000   6.000000   1.000000   0.100000    2
IMPU   0.000000   0.000000   8.000000   6.000000    1
IMPU   0.000000   0.000000   1.000000   6.000000    2
IMPU   7.000000   0.000000   1.000000   6.000000    3
```

Figure A.5 An example of a drawing file for rectangular geometry.

```
device      qss=1.0E+10    fims=0.0

geometry    dwgfil = 'nring.txt'

profile
+           supfil(1)='nring1.imp'
+           supfil(2)='nring2.imp'
+           supfil(3)='nring3.imp'

grid
+           xmaxpt=70 ymaxpt=80

material    mattyp(1)='Si'         mattyp(2)='SiO2'

bias
+           volt(1)=45.0   volt(2)=0.0
+           vsub=45.0

print       detail=.false.      printg=.true.  printc=.false.
+           printv=.false.       printe=.false.  printm=.true.
+

plot        plotv=.true.  plote=.true.
+           nlevel=8       nspot=3
+           vlevel(1)=2.0  vlevel(2)=10.0 vlevel(3)=15.0
+           vlevel(4)=20.0 vlevel(5)=25.0 vlevel(6)=30.0
+           vlevel(7)=35.0 vlevel(8)=40.0
stop
```

Figure A.6 An example of input file for rectangular geometry.

major program are stored in three separate sub-directories, while the image files of the three programs are stored in a fourth sub-directory. The directory structure is shown in Fig. A.7.

Each source file consists of a single program unit, a main program, a subroutine, or an include file. The name of source file is identical with the name of the program unit. A listing of all the file names by directory is displayed in Fig. A.8.

A VAX/VMS command file, *make*, stored in the root directory, facilitates the compilation and linking of the three major programs of SEPSIP. To use this utility, enter the command

make :== @ <*complete-directory-name*>make

to the system's prompt. This line can also be placed permanently in the login file. In this way, the *make* procedure can be activated from any directory. To activate the *make* procedure, get into the subdirectory where the source files of one of the three main programs reside and type

make install

at the system's prompt.

An object library is then created in the directory during the execution of the *make* procedure. The source files are compiled one at a time. At the end of the compilation, a message is issued to the screen showing that *make* is now linking an image file. At the completion of the *make* procedure, the image file is transferred to the sub-directory [.EXECU], and all the object files in the current directory are removed. After the installation is completed, the sub-directory for execution consists of three image files and one indexed data file, *hierr*, which handles error messages for SEPSIP.

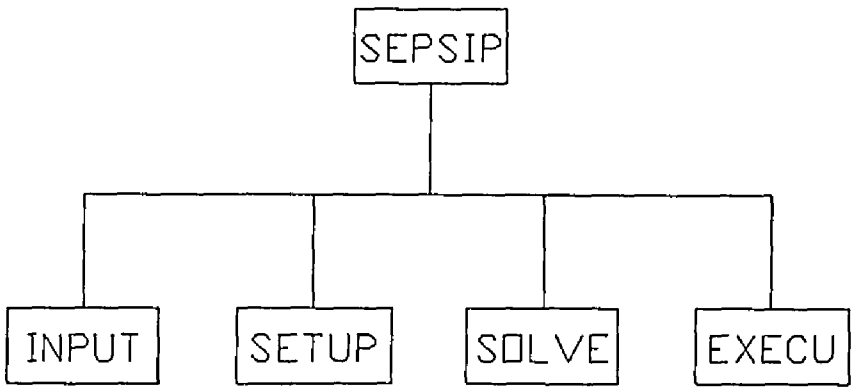

Figure A.7 SEPSIP directory structure.

DIRECTORY #### : [SEPSIP]

EXECU.DIR;1	INPUT.DIR;1	MAKE.COM;1	SEPSIP.COM;1
SETUP.DIR;1	SOLVE.DIR;1		

DIRECTORY #### : [SEPSIP.INPUT]

EPSLN.FOR;1	GTARG.FOR;1	INBLK.FOR;1	INDWG.CMN;1
INPUT.CMN;1	INPUT.FOR;1	LISTS.FOR;1	OKDWG.FOR;1
OKNML.FOR;1	OPNFL.FOR;1	REDWG.FOR;1	RENML.FOR;1
RWSUP.FOR;1	WRERR.FOR;1	WRSET.FOR;1	

DIRECTORY #### : [SEPSIP.SETUP]

COEFF.FOR;1	DISCR.FOR;1	DMAIN.FOR;1	DOPES.FOR;1
EDGES.FOR;1	ESTIM.FOR;1	FERMI.FOR;1	GRIDS.FOR;1
GTARG.FOR;1	LOCKB.FOR;1	MAPID.FOR;1	NODES.FOR;1
NORMS.FOR;1	OPNFL.FOR;1	PASTE.FOR;1	RESET.FOR;1
SEBLK.FOR;1	SESOV.CMN;1	SETUP.CMN;1	SETUP.FOR;1
SLANT.FOR;1	SORTA.FOR;1	TUSER.FOR;1	WRERR.FOR;1
WROUT.FOR;1	WRSOV.FOR;1	XLEFT.FOR;1	XRITE.FOR;1

DIRECTORY #### : [SEPSIP.SOLVE]

DIREC.FOR;1	DPATH.FOR;1	FIELD.FOR;1	GETPT.FOR;1
GTARG.FOR;1	INITL.FOR;1	IONIZ.FOR;1	LINKP.FOR;1
OPNFL.FOR;1	PARAM.FOR;1	PLFLD.FOR;1	PLINE.FOR;1
PLVOL.FOR;1	RELAX.FOR;1	RENEW.FOR;1	RESOV.FOR;1
SOBLK.FOR;1	SOEQS.CMN;1	SOLVE.CMN;1	SOLVE.FOR;1
WRDXF.FOR;1	WRERR.FOR;1	WRSIM.FOR;1	

DIRECTORY #### : [SEPSIP.EXECU]

HIERR.;1	INPUT.EXE;1	SETUP.EXE;1	SOLVE.EXE;1

Figure A.8 The listing of source file names.

If any error occurs during compilation, the *make* procedure reports linking failure, and terminates the procedure. Then the source files responsible for the detected errors must be corrected, and the *make* command is reactivated. If the files (extension: *cmn*) containing the common blocks of global variables are changed, all source files that include these common blocks must be recompiled and linked using the *make* procedure.

New error messages can be added to *hierr*, by entering the [.PROGM] directory to edit the file, *hierr.fky*. Then execute the program, *doerr*. After the execution of *doerr*, a new version of the file *hierr* with the new error messages integrated into the file, is created. Finally, transfer this *hierr* file to the subdirectory [.EXECU] to support SEPSIP in its error handling.

A command procedure file named *sepsip* in the root directory is available to automate the execution of the device simulator, SEPSIP. Before simulation begins, the input file, <*filename*>*.inp*, drawing file, <*filename*>*.txt*, and doping concentration files, <*filename*>*.imp* must be present in the user's directory. To execute SEPSIP from a user's directory, define

sepsip :== @ <*complete-directory-name*>sepsip

which, of course, can also be permanently stored in the *login.com* file, and then type

sepsip <*filename*>

in response to the system's prompt. The *filename* is the name of the input file without the extension *inp*. This input file must be prepared in accordance with the rules explained in the section A.2 and A.3. The simulation outputs are tabulated in the output file (extension: *out*). If the plot file, <*filename*>*.dxf* is transferred to the PC, a graphic display of equipotential contours and high field spots of the

results is possible, using the graphics program AutoCAD. If equi-field contours are requested during the simulation, then the graphic plot file *sep1.dxf* is created.

TWO-DIMENSIONAL DEVICE SIMULATION OF JUNCTION TERMINATION STRUCTURES FOR DETERMINATION OF BREAKDOWN BEHAVIOR

Leong Hin Tan, M.S.

The University of Arizona, 1989

Thesis Director: Dr. François E. Cellier

In this work, we have investigated numerical techniques to determine the breakdown behavior of complex semiconductor devices using two-dimensional simulation. In particular, we have augmented the device simulator SEPSIP with a capability for handling single and multiple floating field rings, and for handling devices with slanted edges. We have furthermore improved the grid width selection algorithm in SEPSIP. A capability for plotting equi-field contours was added to the code. Finally, all system dependencies were removed from the SEPSIP code, and a new version of SEPSIP (Version 2.0) was generated which can be executed on any PC/XT, PC/AT, or PC/386 compatible computer. This eliminates the need for transfering files back and forth between the PC, which had formerly been used as an I/O processor, and the VAX, which was used for numerically intensive computations. It also makes the code more accessible to scientists and engineers who are working in this important research area.

REFERENCES

1 S. Selberherr, A. Schütz, and H. W. Pötzl, "MINIMOS–A two–dimensional MOS transistor analyzer," *IEEE Trans. Electron Devices*, vol. ED–27, pp. 1540–1550, Aug. 1980.

2 A. F. Franz and G. A. Franz, "BAMBI 1.0 user's guide," Internal report, Institut für Allgemeine Elektrotechnik und Elektronik, Abt. Physikalische Elektronik, Techn. Universität Wien, 1985.

3 C. S. Rafferty, M. R. Pinto, and R.W. Dutton, "Iterative methods in semiconductor device simulation," *IEEE Trans. Computer–Aided Design*, vol. CAD–4, pp. 461–471, Oct. 1985.

4 T. D. Last, D. W. Lucas, and G. W. Sumerling, "A numerical analysis of the d.c. performance of small geometry lateral transistors," *Solid–State Electron.*, vol. 17, pp. 1111–1118, Nov. 1974.

5 O. Manck and W. L. Engl, "Two–dimensional computer simulation for switching a bipolar transistor out of saturation," *IEEE Trans. Electron Devices*, vol. ED–22, pp. 339–347, June 1975.

6 T. Toyabe and S. Asai, "Analytical models of threshold voltage and breakdown voltage of short–channel MOSFETs derived from two–dimensional analysis," *IEEE Trans. Electron Devices*, vol. ED–26, pp. 453–461, Apr. 1979.

7 T. Adachi, A. Yoshii, and T. Sado, "Two–dimensional semiconductor analysis using finite–element method," *IEEE Trans. Electron Devices*, vol. ED–26, pp. 1026–1031, June 1979.

8 L. M. Dang and M. Konaka, "A two–dimensional computer analysis of triode–like characteristics of short–channel MOSFETs," *IEEE Trans. Electron Devices*, vol. ED–27, pp. 1533–1539, Aug. 1980.

9 A. Schütz, S. Selberherr and H. W. Pötzl, "A two-dimensional model for the Avalanche effect in MOS transistors," *Solid–State Electron.*, vol. 25, pp. 177–183, Nov. 1982.

10 C. Bulucea, A. Rusu, and C. Postolache, "Surface breakdown in silicon planar junctions–A computer–aided experimental determination of the critical field," *Solid–State Electron.*, vol. 17, pp. 881–888, Sept. 1974.

11 K. E. Kroell and G. K. Ackermann, "Threshold voltage of narrow channel field effect transistors," *Solid–State Electron.*, vol. 19, pp. 77–81, Jan. 1976.

12 A. Rusu and C. Bulucea, "Deep–depletion breakdown voltage of silicon–dioxide/silicon MOS capacitors," *IEEE Trans. Electron Devices*, vol. ED–26, pp. 201–205, Mar. 1979.

13 P. K. Chatterjee, G. W. Taylor, and A. F. Tasch, Jr., "Charge–coupled device structures for VLSI memories," *IEEE Trans. Electron Devices*, vol. ED–26, pp. 871–881, June 1979.

14 J. A. Greenfield and R. W. Dutton, "Nonplanar VLSI device analysis using the solution of Poisson's equation," *IEEE Trans. Electron Devices*, vol. ED–27, pp. 1520–1532, Aug. 1980.

15 S. Liu, B. Hoefflinger and D. O. Pederson, "Iterative two–dimensional design of barrier–controlled MOS transistors," *IEEE Trans. Electron Devices*, vol. ED–27, pp. 1550–1558, Aug. 1980.

16 S. Yasuda and M. Kurata, "Two-dimensional field distribution analysis of reverse biased *p-n* junction devices," *Solid–State Electron.*, vol. 23, pp. 1077–1084, Oct. 1980.

17 C. Yen, "Two–Dimensional Simulation of Power MOSFET near Breakdown ," *Master thesis, University of Arizona*, 1988.

18 E. L. Wachspress, *Iterative Solution of Elliptic Systems.* Prentice–Hall, Englewood Cliffs, NJ, 1966.

19 R. S. Varga, *Matrix Iterative Analysis.* Prentice–Hall, Englewood Cliffs, NJ, 1962.

20 S. M. Sze and G. Gibbons, "Effect of junction curvature on breakdown voltage in semiconductors," *Solid–State Electron* , vol. 9, pp. 831–845, Sep. 1966.

21 G. Gibbons and J. Kocsis, "Breakdown voltages of germanium plane cylinderical junctions," *IEEE Trans. Electron Devices*, vol. ED–12, pp. 193–198, April 1965.

22 B. J. Baliga, *Modern Power Devices.* John Wiley, New York, 1987.

23 A. S. Grove, O. Leistiko, Jr. and W. W Hooper, "Effect of surface fields on the breakdown voltage of planar silicon *p-n* junctions," *IEEE Trans4. Electron Devices*, vol. ED–14, pp. 157–162, March 1967.

24 F. Conti and M. Conti, "Surface breakdown in silicon planar diodes equipped with field plate, " *Solid–State Electron* , vol. 15, pp. 93–105, Jan. 1972.

25 V. P. O'Neil and P. G. Alonas, "Relation between oxide thickness and the breakdown voltage of a planar junction with field relief electrode," *IEEE Trans. Electron Devices*, vol. ED–26, pp. 1098–1100, March 1979.

26 K. Hwang and D. H. Navon, "Breakdown voltage optimization of silicon *p-π-ν* planar junction diodes," *IEEE Trans. Electron Devices*, vol. ED–31, pp. 1126–1135, Sept. 1984.

27 K. R. Davis, "Two–Dimensional Simulation of the effects of Total Dose Ionizing Radiation on Power-MOSFET Breakdown," *Master thesis, University of Arizona*, 1989.

28 Y. C. Kao and E. D. Wolley, "High–voltage p–n junction," *Proc. IEEE* , vol. 55, pp. 1409–1414, 1967.

29 M. S. Adler and A. P. Ferro, "Field calculations of planar diffused structures with field limiting rings," *Technical Digest 1979 IEEE IEDM* , pp. 105–109, Dec. 1973.

30 M. S. Adler, "A new non-capacitive method of determining surface charge densities using floating field ring devices," *Technical Digest 1974 IEEE IEDM* pp. 255–259, Dec. 1974.

31 S. Yasuda and T. Yonezawa, "High–voltage planar junction with a field limiting ring," *Solid-State Electron* , vol. 25, pp. 423-427, May 1982.

32 Brieger ,"Blocking capability of planar devices with field limiting rings," *Solid-State Electron* , vol. 26, pp. 739-745, Aug. 1983.

33 K.P. Brieger, E. Falck and W. Gerlach, "The contour of an optimal field plate – An analytical approach," *IEEE Trans. Electron Devices*, vol. ED–31, pp. 1126–1135, Sept. 1984.

34 Q.-M. Wu, C.-M. Yen and F. Cellier, "Analysis of Breakdown Phenomena in high–voltage bipolar devices," *Transactions of SCS*, to appear.

35 M. S. Adler, V. A. K. Temple and et al., "Theory and breakdown voltage for planar devices with a single field limiting ring," *IEEE Trans. Electron Devices*, vol. ED–24, pp. 107–113, Feb. 1977.

36 Q. Wu and F. Cellier, "Simulation of bipolar high–voltage devices in the neighborhood of breakdown," *Mathematics and Computers in Simulation*, vol. 28, pp. 271–284, 1986.

37 R. Bellmann, M. Juncosa, and R. Kalaba, "Some numerical experiments using Newton's method for nonlinear parabolic and elliptic boundary value problems,"Rept. No. p–2200, Rand Corp., Santa Monica, California, 1961.

38 A. Blicher, *Field–Effect and Bipolar Power Transistor Physics.* Academic Press, New York, 1981.

39 S. M. Sze, *Physics of Semiconductor Devices.* John Wiley, New York, 1981.

40 S. K. Ghandhi, *Semiconductor Power Devices.* John Wiley, New York, 1977.

41 V. Boisson, M. L. Helley and J. Chante, "Computer study of a high-voltage p–π–n^-–n^+ diode and comparison with a field-limiting ring structure," *IEEE Trans. Electron Devices*, vol. ED–33, pp. 80–84, Jan. 1986.

42 S. M. Sze and G. Gibbons, "Avalanche breakdown voltages of abrupt and linearly graded p-n junctions in Ge, Si, GaAs, and GaP," *Appl. Phys. Lett.*, vol. 8, pp. 111–135, 1966.

43 Autodesk, Inc., *The AutoCAD Reference Manual*, Suasalito, California, 1987.

44 S. E. Hansen, *SUPREM-III User's Manual*, The Board of Trustees of Stanford University, 1982.