Taurus™ TSUPREM-4

Taurus TSUPREM-4

User Guide

Version Z-2007.03, March 2007

SYNOPSYS®

Copyright Notice and Proprietary Information

Right to Copy Documentation

The license agreement with Synopsys permits licensee to make copies of the documentation for its internal use only. Each copy shall include all copyrights, trademarks, service marks, and proprietary rights notices, if any. Licensee must assign sequential numbers to all copies. These copies shall contain the following legend on the cover page:

> "This document is duplicated with the permission of Synopsys, Inc., for the exclusive use of ____ and its employees. This is copy number __________."

Destination Control Statement

All technical data contained in this publication is subject to the export control laws of the United States of America. Disclosure to nationals of other countries contrary to United States law is prohibited. It is the reader's responsibility to determine the applicable regulations and to comply with them.

Disclaimer

Registered Trademarks (®)

Synopsys, AMPS, Cadabra, CATS, CRITIC, CSim, Design Compiler, DesignPower, DesignWare, EPIC, Formality, HSIM, HSPICE, iN-Phase, in-Sync, Leda, MAST, ModelTools, NanoSim, OpenVera, PathMill, Photolynx, Physical Compiler, PrimeTime, SiVL, SNUG, SolvNet, System Compiler, TetraMAX, VCS, Vera, and YIELDirector are registered trademarks of Synopsys, Inc.

Trademarks (™)

AFGen, Apollo, Astro, Astro-Rail, Astro-Xtalk, Aurora, AvanWaves, Columbia, Columbia-CE, Cosmos, CosmosEnterprise, CosmosLE, CosmosScope, CosmosSE, DC Expert, DC Professional, DC Ultra, Design Analyzer, Design Vision, DesignerHDL, Direct Silicon Access, Discovery, Encore, Galaxy, HANEX, HDL Compiler, Hercules, Hierarchical Optimization Technology, HSIM^{plus}, HSPICE-Link, iN-Tandem, i-Virtual Stepper, Jupiter, Jupiter-DP, JupiterXT, JupiterXT-ASIC, Liberty, Libra-Passport, Library Compiler, Magellan, Mars, Mars-Xtalk, Milkyway, ModelSource, Module Compiler, Planet, Planet-PL, Polaris, Power Compiler, Raphael, Raphael-NES, Saturn, Scirocco, Scirocco-i, Star-RCXT, Star-SimXT, Taurus, TSUPREM-4, VCS Express, VCSi, VHDL Compiler, VirSim, and VMC are trademarks of Synopsys, Inc.

Service Marks (^{SM})

MAP-in, SVP Café, and TAP-in are service marks of Synopsys, Inc.

SystemC is a trademark of the Open SystemC Initiative and is used under license.
ARM and AMBA are registered trademarks of ARM Limited.
Saber is a registered trademark of SabreMark Limited Partnership and is used under license.
All other product or company names may be trademarks of their respective owners.

Printed in the U.S.A.

Taurus TSUPREM-4 User Guide, version Z-2007.03

About This Guide

Overview

This manual includes details for using TSUPREM-4 and covers all aspects of the TSUPREM-4 2D process simulation program.

This manual contains the following chapters:

About This Guide	Includes conventions, related publications, and customer support information.
Chapter 1	Gives an overview of the program.
Chapter 2	Discusses the execution of TSUPREM-4, the required input files, the output files generated, and other files required to run the program.
Chapter 3	Describes the physical models for the physical processes simulated by TSUPREM-4 and discusses some of the numerical methods used during the simulation.
Chapter 4	Contains detailed descriptions of the input statements recognized by TSUPREM-4. The description of each statement includes a summary of the statement syntax, descriptions of the statement parameters, and a discussion of the use of the statement, with examples.
Chapter 5	Presents simple examples illustrating the use of the program.
Chapter 6	Presents more complicated examples illustrating the use of the program for simulating complete processes.
Chapter 7	Describes the USEIT Option, Physical Model, and Equation Interface language, and details examples.
Appendix A	Lists the default simulation coefficient values and the literature references from which they were derived.

Appendix B	Describes the plot device definition file *s4pcap*. This file contains information that describes the available graphical output devices.
Appendix C	Describes the data format used by mask data files.
Appendix D	Describes the data formats files created with the **`SAVEFILE`** statement.
Appendix E	Contains a detailed description of the interface to the MINIMOS 5 device simulation program.
Appendix F	Describes new features, enhancements, and changes in this version.

Related Publications

For additional information about TSUPREM-4, see:

- *TCAD Products and Utilities Installation Manual.*
- The documentation installed with the TSUPREM-4 software are available through the TSUPREM-4 Help menu.
- The TSUPREM-4 release notes are available on SolvNet (see Accessing SolvNet on page 0-v).
- *Taurus Layout Tutorial* is available on Docs on the Web (DOW) via SolvNet (see Accessing SolvNet on page 0-v) and on the product CD.

Reference Materials

This manual uses many references from the changing body of industry literature. Where appropriate, you are directed to source material. References are included in Chapter 3 and Appendix A.

If a button label appears grayed out, the button is not currently active. Attempts to click the button are ignored.

Conventions

The following conventions are used in Synopsys documentation.

Convention	Description
`Courier`	Indicates command syntax.
`Courier italic`	Indicates a user-defined value in Synopsys syntax, such as *`object_name`*. (A user-defined value that is not Synopsys syntax is indicated by regular text font italic.)

Convention	Description
`Courier bold`	Indicates user input—text you type verbatim—in Synopsys syntax and examples. (User input that is not Synopsys syntax, such as a user name or password you enter in a GUI, is indicated by regular text font bold.)
[ ]	Denotes optional parameters, such as *`pin1`* **`[`***`pin2 ... pinN`***`]`**
|	Indicates a choice among alternatives, such as **`low	medium	high`** (This example indicates that you can enter one of three possible values for an option: low, medium, or high.)
_	Connects terms that are read as a single term by the system, such as **`set_annotated_delay`**
Control-c	Indicates a keyboard combination, such as holding down the Control key and pressing c.
\	Indicates a continuation of a command line.
/	Indicates levels of directory structure.
Edit > Copy	Indicates a path to a menu command, such as opening the Edit menu and choosing Copy.

Customer Support

Customer support is available through SolvNet online customer support and through contacting the Synopsys Technical Support Center.

Accessing SolvNet

SolvNet includes an electronic knowledge base of technical articles and answers to frequently asked questions about Synopsys tools. SolvNet also gives you access to a wide range of Synopsys online services, which include downloading software, viewing Documentation on the Web, and entering a call to the Support Center.

To access SolvNet:

1. Go to the SolvNet Web page at http://solvnet.synopsys.com.
2. If prompted, enter your user name and password. (If you do not have a Synopsys user name and password, follow the instructions to register with SolvNet.)
3. If you need help using SolvNet, click SolvNet Help in the Support Resources section.

Contacting the Synopsys Technical Support Center

If you have problems, questions, or suggestions, you can contact the Synopsys Technical Support Center in the following ways:

- Open a call to your local support center from the Web by going to http://solvnet.synopsys.com (Synopsys user name and password required), then clicking "Enter a Call to the Support Center."
- Send an e-mail message to your local support center.
  - E-mail support_center@synopsys.com from within North America.
  - Find other local support center e-mail addresses at http://www.synopsys.com/support/support_ctr.
- Telephone your local support center.
  - Call (800) 245-8005 from within the continental United States.
  - Call (650) 584-4200 from Canada.
  - Find other local support center telephone numbers at http://www.synopsys.com/support/support_ctr.

Table of Contents

About This Guide iii

Overview iii
Related Publications iv
  Reference Materials iv
Conventions iv
Customer Support v
  Accessing SolvNet v
  Contacting the Synopsys Technical Support Center vi

Introduction to TSUPREM-4 1-1

Product Name Change 1-1
Program Overview 1-1
Processing Steps 1-2
Simulation Structure 1-2
Additional Features 1-2

Using TSUPREM-4 2-1

Program Execution and Output 2-1
  Starting TSUPREM-4 2-1
  Program Output 2-2
    Printed Output 2-2
    Graphical Output 2-2
  Errors, Warnings, and Syntax 2-3
File Specification 2-3
  File Types 2-3
  Default File Names 2-3
  Environment Variables 2-4
Input Files 2-4

- Command Input Files . . . 2-4
- Mask Data Files . . . 2-5
- Profile Files . . . 2-5
- Other Input Files . . . 2-5

Output Files . . . 2-5
- Terminal Output . . . 2-6
- Output Listing Files . . . 2-6
  - Standard Output File—s4out . . . 2-6
  - Informational Output File—s4inf . . . 2-6
  - Diagnostic Output File—s4dia . . . 2-6
- Saved Structure Files . . . 2-7
  - TSUPREM-4 . . . 2-7
  - TIF . . . 2-7
  - Medici . . . 2-7
  - MINIMOS 5 . . . 2-7
  - Wave . . . 2-7
- Graphical Output . . . 2-7
- Extract Output Files . . . 2-8
- Electrical Data Output Files . . . 2-8

Library Files . . . 2-8
- Initialization Input File—s4init . . . 2-9
- Ion Implant Data File—s4imp0 . . . 2-9
- Ion Implant Damage Data File—s4idam . . . 2-9
- Plot Device Definition File—s4pcap . . . 2-10
- Key Files—s4fky0 and s4uky0 . . . 2-10
- Authorization File—s4auth . . . 2-10

TSUPREM-4 Models 3-1

Simulation Structure . . . 3-2
- Coordinates . . . 3-2
- Initial Structure . . . 3-2
- Regions and Materials . . . 3-2

Grid Structure . . . 3-3
- Mesh, Triangular Elements, and Nodes . . . 3-3
- Defining Grid Structure . . . 3-3
- Explicit Specification of Grid Structure . . . 3-3
  - The LINE Statement . . . 3-4
  - Generated Grid Lines . . . 3-4
  - Eliminating Grid Lines . . . 3-5
- Automatic Grid Generation . . . 3-5
  - Automatic Grid Generation in the X Direction . . . 3-6
  - Automatic Grid Generation in the Y Direction . . . 3-7
- Changes to the Mesh During Processing . . . 3-7
  - DEPOSITION and EPITAXY . . . 3-7
  - Structure Extension . . . 3-8

    - ETCH and DEVELOP 3-8
    - Oxidation and Silicidation 3-9
  - Adaptive Gridding 3-10
    - Refinement 3-10
    - Unrefinement 3-11
    - Adaptive Gridding for Damage 3-12
    - Usage 3-12
  - 1-D Simulation of Simple Structures 3-13
  - Initial Impurity Concentration 3-13
- Diffusion 3-14
  - DIFFUSION Statement 3-15
    - Temperature 3-15
    - Ambient Gas Pressure 3-15
    - Ambient Gas Characteristics 3-15
    - Ambients and Oxidation of Materials 3-16
    - Default Ambients 3-16
    - Chlorine 3-17
    - Chemical Predeposition 3-17
    - Solution of Diffusion Equations 3-17
  - Diffusion of Impurities 3-18
    - Impurity Fluxes 3-18
    - Mobile Impurities and Ion Pairing 3-19
    - Electric Field 3-20
    - Diffusivities 3-22
  - Point Defect Enhancement 3-24
    - PD.FERMI Model 3-24
    - PD.TRANS and PD.FULL Models 3-24
    - Paired Fractions of Dopant Atoms 3-25
    - 5-Stream Diffusion Model 3-26
    - Initialization of Pairs 3-27
    - Boundary Conditions for Pairs 3-27
    - Reaction Rate Constants 3-31
  - Activation of Impurities 3-34
    - Physical Mechanisms 3-34
    - Activation Models 3-35
    - Model Details 3-35
    - Solid Solubility Model 3-36
    - Solid Solubility Tables 3-36
    - Dopant Clustering Model 3-37
    - Transient Clustering Model 3-37
    - Dopant-Interstitial Clustering Model 3-38
    - Dopant-Vacancy Clustering Model 3-39
    - Full Dynamics of Dopant-Defect Clustering 3-40
    - Transient Precipitation Model 3-50
    - DDC and Precipitation Initialization 3-51
  - Segregation of Impurities 3-54
    - Segregation Flux 3-54

- Interface Trap Model 3-55
  - Using the Interface Trap Model 3-58
- Dose Loss Model 3-58
- Diffusion of Point Defects 3-59
  - Equilibrium Concentrations 3-59
  - Charge State Fractions 3-60
  - Point Defect Diffusion Equations. 3-61
  - Interstitial and Vacancy Diffusivities. 3-63
  - Reaction of Pairs with Point Defects 3-64
  - Recombination of Interstitials with Vacancies. 3-64
  - Absorption by Traps, Clusters, and Dislocation Loops 3-66
- Injection and Recombination of Point Defects at Interfaces 3-66
  - Surface Recombination Velocity Models. 3-67
  - Trapped Nitrogen Dependent Surface Recombination. 3-68
  - Injection Rate 3-68
  - Moving-Boundary Flux 3-70
- Interstitial Traps 3-70
  - Enabling, Disabling, and Initialization. 3-71
- Interstitial Clustering Models. 3-71
  - 1-Moment Clustering Model 3-72
  - 2-Moment Interstitial Clustering Model. 3-76
- Small Clusters of Point Defects 3-78
  - Concentration of Defects in Small Clusters 3-79
  - Recombination of Defects in Small Clusters 3-79
  - Transient Small Clustering. 3-80
- Dislocation Loop Model 3-82
  - Equations for Dislocation Loop Model 3-82
  - Loop Density Specified by L.DENS 3-83
  - Loop Density Specified by L.THRESH. 3-84
  - Evolution of Loops. 3-84
  - Effects of Dislocation Loops 3-84

Oxidation 3-84
- Theory of Oxidation. 3-85
- Analytical Oxidation Models. 3-87
  - Overview 3-87
  - Usage 3-89
  - The ERFC Model 3-90
  - ERF1, ERF2, and ERFG Models 3-90
- Numerical Oxidation Models. 3-93
  - Oxide Growth Rate. 3-93
  - VERTICAL Model. 3-95
  - COMPRESS Model 3-96
  - VISCOUS Model 3-97
  - VISCOELA Model. 3-99
  - Anisotropic Stress Dependent Reaction. 3-104
  - Strength-Limited Stress 3-104
- Polysilicon Oxidation. 3-104

- Surface Tension and Reflow 3-104
- N2O Oxidation 3-105
  - Nitrogen Trap and Generation 3-105
  - Surface Reaction Rate 3-106
  - Thin Oxidation Rate 3-106
- Boron Diffusion Enhancement in Oxides 3-106
  - Diffusion Enhancement in Thin Oxides 3-107
  - Diffusion Enhancement Due to Fluorine 3-109

Silicide Models 3-109
- TiSi2 Growth Kinetics 3-109
  - Reaction at TiSi2/Si Interface 3-110
  - Diffusion of Silicon 3-110
  - Reaction at TiSi2/Si Interface 3-110
  - Oxygen Dependence 3-110
  - Initialization 3-110
  - Material Flow 3-111
- Impurities and Point Defects 3-111
- Specifying Silicide Models and Parameters 3-111
  - Materials 3-111
  - Impurities 3-112
  - Reactions 3-112
  - Dopants 3-114
- Tungsten, Cobalt, and Nickel Silicide Models 3-114
- Other Silicides 3-114

Stress Models 3-115
- Stress History Model 3-115
- Thermal Stress Model Equations 3-115
  - Boundary Conditions 3-115
  - Initial Conditions 3-116
  - Intrinsic Stress in Deposited Layers 3-116
  - Effect of Etching and Depositionon Stress 3-117
  - Using the Stress History Model 3-117
  - Limitations 3-117
- Dopant-Induced Stress Models 3-117
  - Dopant-Induced Stress during Deposition 3-118
  - Dopant-Induced Stress during Diffusion 3-119

Ion Implantation 3-119
- Analytic Ion Implant Models 3-119
  - Implanted Impurity Distributions 3-120
  - Implant Moment Tables 3-121
  - Gaussian Distribution 3-123
  - Pearson Distribution 3-123
  - Dual Pearson Distribution 3-124
  - Dose-dependent Implant Profiles 3-124
  - Tilt and Rotation Tables 3-127
  - Multilayer Implants 3-127
  - Lateral Distribution 3-128

- Depth-Dependent Lateral Distribution . . . 3-128
- Dose Integration from Lateral Distribution . . . 3-128
- Wafer Tilt and Rotation . . . 3-129
- BF2 Implant . . . 3-129
- Analytic Damage Model. . . . 3-129
- Taurus Analytic Implant Model . . . 3-130
- MC Ion Implant Model . . . 3-131
  - Taurus MC Implant . . . 3-131
  - Old TSUPREM-4 MC Model. . . . 3-133
  - Binary Scattering Theory . . . 3-133
  - Amorphous Implant Calculation . . . 3-137
  - Crystalline Implant Model . . . 3-140
  - Damage Dechanneling . . . 3-143
- Implant Damage Model . . . 3-144
  - Damage Produced During Implant. . . . 3-144
  - Profiling Implant Damages. . . . 3-146
  - Cumulative Damage Model . . . 3-146
  - Conservation of Total Defect Concentrations . . . 3-148
  - Using the Implant Damage Model . . . 3-148
- Boundary Conditions for Ion Implantation . . . 3-149
  - Boundary Conditions for OLD.MC Implant . . . 3-149
  - Boundary Conditions for Taurus Analytic and Taurus MC Implant . . . 3-150
- MC Implant into Polysilicon . . . 3-151
- MC Implant into Hexagonal Silicon Carbide . . . 3-152
- Fields to Store Implant Information. . . . 3-153

Epitaxial Growth. . . . 3-153
- Layer Thickness . . . 3-154
- Incorporation of Impurities . . . 3-154
- Diffusion of Impurities. . . . 3-154
- Selective Epitaxy . . . 3-154

Deposition. . . . 3-155
- Layer Thickness . . . 3-155
- Anisotropy . . . 3-155
- Incorporation of Impurities . . . 3-156
- Photoresist Type. . . . 3-156
- Polycrystalline Materials . . . 3-156
- Deposition with Taurus Topography . . . 3-156

Masking, Exposure, and Development of Photoresist . . . 3-156

Etching . . . 3-157
- Defining the Etch Region. . . . 3-157
- Removal of Material . . . 3-158
- Trapezoidal Etch Model. . . . 3-158
  - Parameters . . . 3-158
  - Etch Steps . . . 3-158
  - Etch Examples . . . 3-159
- Etching with Taurus Topography. . . . 3-161

Modeling Polycrystalline Materials 3-161
  Diffusion 3-162
    Diffusion in Grain Interiors 3-162
    Grain Boundary Structure. 3-162
    Diffusion Along Grain Boundaries 3-163
  Segregation Between Grain Interior and Boundaries 3-164
  Grain Size Model 3-165
    Grain Growth 3-166
  Interface Oxide Break-up and Epitaxial Regrowth 3-167
    Oxide Break-Up 3-167
    Epitaxial Regrowth. 3-168
  Dependence of Polysilicon Oxidation Rate on Grain Size 3-168
  Using the Polycrystalline Model 3-169
Electrical Calculations 3-170
  Automatic Regrid. 3-171
  Poisson's Equation. 3-171
    Boltzmann and Fermi-Dirac Statistics 3-171
    Ionization of Impurities 3-172
    Solution Methods 3-173
  Carrier Mobility 3-173
    Tabular Form 3-174
    Arora Mobility Model 3-174
    Caughey Mobility Model 3-175
  Quantum Mechanical Model for MOSFET 3-175
  Capacitance Calculation. 3-176
    DC Method. 3-176
    MOS Capacitances 3-177
References. 3-178

Input Statement Descriptions 4-1

Input Statements 4-2
  Format 4-2
  Syntax. 4-2
  Specifying Materials and Impurities 4-2
  Parameters 4-3
    Character. 4-3
    Logical 4-3
    Numerical. 4-3
  Statement Description Format 4-4
    Parameter Definition Table 4-4
    Syntax of Parameter Lists. 4-4
Documentation and Control 4-5
  COMMENT 4-7
    Description 4-7
    Examples 4-7

  - Notes 4-7
- SOURCE 4-8
  - Description 4-8
  - Reusing Combinations of Statements. 4-8
  - Generating Templates. 4-8
  - Examples 4-8
- RETURN 4-9
  - Description 4-9
  - Returning from Batch Mode. 4-9
  - Exiting Interactive Input Mode 4-9
  - Example 4-10
- INTERACTIVE 4-11
  - Description 4-11
  - Interactive Input Mode 4-11
  - Example 4-11
- PAUSE. 4-12
  - Description 4-12
  - Example 4-12
- STOP 4-13
  - Description 4-13
  - Example 4-13
- FOREACH/END 4-14
  - Description 4-14
  - Examples 4-14
- LOOP/L.END 4-16
  - Description 4-17
  - Termination of Optimization Looping 4-17
  - Parameter Sensitivity 4-17
  - Example 4-18
  - Advantages 4-19
- L.MODIFY 4-20
  - Description 4-20
- IF/ELSEIF/ELSE/IF.END 4-21
  - Description 4-21
  - Conditional Operators 4-22
  - Expression for Condition 4-22
- ASSIGN 4-23
  - Description 4-26
  - Varying During Statement Looping 4-27
  - ASSIGN with Mathematical Expressions 4-27
  - ASSIGN and Optimization. 4-28
  - Expansion of ASSIGNed Variable. 4-28
  - Reading the External Data File. 4-29
  - Reading the Array from a String 4-29
- INTERMEDIATE 4-30
  - Advantages of INTERMEDIATES 4-31
  - Scope Definition. 4-31

    Value Type 4-33
    Array Values. 4-34
    Comparison to ASSIGN. 4-36
    Snapshot 4-37
    Load/Save Intermediates 4-38
  ECHO. 4-40
    Description 4-40
    Examples 4-40
  OPTION 4-41
    Selecting a Graphics Device. 4-42
    Redirecting Graphics Output 4-42
    Printed Output 4-42
    Informational and Diagnostic Output. 4-43
    Echoing and Execution of Input Statements 4-43
    Version Compatibility 4-43
    Strict Syntax Check 4-43
    Automatic Save 4-44
    Load/Save Files in Compressed Format. 4-44
    Examples 4-45
  DEFINE 4-46
    Description 4-46
    Format and Syntax 4-46
    Examples 4-46
  UNDEFINE 4-49
    Description 4-49
    Example 4-49
    Redefined Parameter Names 4-49
  CPULOG 4-50
    Description 4-50
    Examples 4-50
    Limitations 4-50
  HELP 4-51
    Description 4-51
    Example 4-51
  !(Exclamation Mark) 4-51
Device Structure Specification 4-52
  MESH. 4-53
    Description 4-54
    Grid Creation Methods. 4-54
    Horizontal Grid Generation 4-55
    Vertical Grid Generation 4-55
    Scaling the Grid Spacing 4-56
    1D Mode. 4-56
    Examples 4-56
  LINE. 4-57
    Description 4-57
    Placing Grid Lines 4-57

- Example 4-58
- Additional Notes. . . . 4-58

ELIMINATE 4-59
- Description 4-59
- Overlapping Regions 4-60
- Examples 4-60

BOUNDARY 4-62
- Description 4-62
- Limitations 4-63
- Example 4-63

REGION. 4-64
- Description 4-65
- Redefining Material of the Region 4-65
- Examples 4-65

INITIALIZE. 4-66
- Description 4-68
- Mesh Generation 4-69
- Previously Saved Structure Files 4-69
- Crystalline Orientation 4-69
- Specifying Initial Doping 4-69
- Redefining Material of the Region 4-70
- Examples 4-70

LOADFILE 4-71
- Description 4-71
- TSUPREM-4 Files 4-72
- Older Versions 4-72
- User-Defined Materials and Impurities 4-72
- Replacing Materials of Regions 4-72
- Examples 4-72

SAVEFILE. 4-74
- Description 4-77
- TSUPREM-4 Files 4-77
- Older Versions 4-77
- TIF Files 4-77
- Medici Files. 4-78
- MINIMOS 4-78
- Temperature 4-78
- Examples 4-78

STRUCTURE 4-79
- Description 4-81
- Order of Operations 4-82
- Usage 4-82
- TSUPREM-4 Version Compatibility 4-82
- Examples 4-82

MASK 4-84
- Description 4-84
- Examples 4-85

- PROFILE 4-86
  - Description 4-88
  - OFFSET Parameter 4-89
  - Interpolation 4-89
  - IMPURITY Parameter 4-89
  - Profile in 2D Rectangular Region 4-90
  - Importing Columwise 2D Profiles 4-90
  - Importing 2D Profiles in TIF or TS4 Format 4-90
  - Example 4-91
- ELECTRODE 4-96
  - Description 4-96
  - Examples 4-97

Process Steps. 4-98

- DEPOSITION 4-99
  - Description 4-102
  - Deposition with Taurus Topography 4-103
  - Examples 4-104
  - Additional DEPOSITION Notes 4-105
- EXPOSE. 4-106
  - Description 4-106
  - Example 4-107
- DEVELOP 4-108
  - Description 4-108
  - Example 4-108
- ETCH 4-109
  - Description 4-111
  - Removing Regions 4-111
  - Etching with Taurus Topography. 4-112
  - Examples 4-113
- IMPLANT 4-114
  - Description 4-123
  - Gaussian and Pearson Distributions 4-124
  - Table of Range Statistics 4-124
  - Point Defect Generation 4-125
  - Extended Defects 4-126
  - Boundary Conditions 4-126
  - Rotation Angle and Substrate Rotation 4-127
  - TSUPREM-4 Version Considerations 4-127
  - Examples 4-127
- DIFFUSION. 4-131
  - Description 4-134
  - Ambient Gas. 4-134
  - Oxidation Limitations. 4-135
  - Reflow 4-136
  - Examples 4-136
- EPITAXY. 4-137
  - Description 4-139

  - Example 4-141
- STRESS 4-142
  - Description 4-142
  - Printing and Plotting of Stresses and Displacements 4-144
  - Example 4-144

Output 4-145
- SELECT 4-146
  - Description 4-146
  - Solution Values 4-146
  - Mathematical Operations and Functions 4-148
  - Examples 4-149
- PRINT.1D 4-151
  - Description 4-153
  - Layers 4-153
  - Interface Values 4-153
  - Saving Profiles in a File 4-153
  - Examples 4-154
- PLOT.1D 4-155
  - Description 4-161
  - Line Type and Color. 4-161
  - Examples 4-162
- PLOT.2D 4-163
  - Description 4-166
  - Line Type and Color. 4-166
  - Examples 4-166
- CONTOUR 4-168
  - Description 4-169
  - Line Type and Color. 4-169
  - Example 4-169
- COLOR 4-170
  - Description 4-171
  - Plot Device Selection 4-171
  - Examples 4-171
- PLOT.3D 4-172
  - Description 4-173
  - Line Type and Color. 4-174
  - Examples 4-174
- LABEL. 4-175
  - Description 4-178
  - Label Placement 4-178
  - Line, Symbol, and Rectangle 4-178
  - Color. 4-179
  - Examples 4-179
- EXTRACT 4-180
  - Description 4-185
  - Solution Variables 4-186
  - Extraction Procedure 4-186

Targets for Optimization 4-188
File Formats 4-188
Error Calculation 4-188
Examples 4-189
Optimization Examples 4-191
ELECTRICAL 4-194
Description 4-199
Files and Plotting 4-199
Examples 4-199
Optimization Examples 4-201
Quantum Effect in CV Plot 4-201
VIEWPORT 4-204
Description 4-204
Scaling Plot Size 4-204
Examples 4-205
Models and Coefficients 4-206
METHOD 4-207
Description 4-221
Oxidation Models 4-221
Grid Spacing in Growing Oxide 4-221
Rigid vs. Viscous Substrate 4-221
Point Defect Modeling 4-222
PD.FERMI Model 4-222
PD.TRANS Model 4-222
PD.FULL Model 4-222
PD.5STR Model 4-222
Customizing the Point Defect Models 4-223
Enable/Disable User-Specified Models 4-223
Solving Poisson’s Equation 4-223
Adaptive Gridding 4-224
Fine Control 4-225
Initial Time Step 4-225
Internal Solution Methods 4-225
Time Integration 4-225
System Solutions 4-226
Minimum-Fill Reordering 4-226
Block Solution 4-227
Automatic Tune for Optimal Numerical Method 4-227
Solution Method 4-227
Matrix Structure 4-227
Matrix Refactoring 4-227
Discontinuous Temperature Change 4-227
Error Tolerances 4-228
More Accurate Dose Integration from Lateral Distribution 4-228
Sensitivity of Implanted Profiles to Non-planar Interfaces 4-228
Examples 4-228
EQUATION 4-230

  - Description 4-231
  - Initialization of Solution. 4-231
  - See Also 4-232
- AMBIENT 4-233
  - Description 4-245
  - Oxidation Models 4-245
  - Examples 4-248
  - Parameter Dependencies 4-249
  - Examples 4-249
  - Additional AMBIENT Notes 4-250
- MOMENT 4-251
  - Description 4-255
  - Using the MOMENT Statement. 4-255
  - Examples 4-256
- MATERIAL. 4-258
  - Description 4-270
  - Viscosity and Compressibility 4-270
  - Stress Dependence 4-270
  - Modeling the Energy Bandgap. 4-271
  - Examples 4-271
- IMPURITY 4-273
  - Description 4-296
  - Impurity Type. 4-297
  - Solution Options. 4-297
  - Other Parameters 4-297
  - Multiplication to Diffusivity. 4-297
  - Anisotropic Diffusivity. 4-297
  - User-defined Active Concentration 4-298
  - See Also 4-298
  - Examples 4-298
- REACTION 4-300
  - Description 4-302
  - Insertion of Native Layers 4-303
  - Reaction Equation 4-303
  - Parameters 4-303
  - Effects. 4-304
- MOBILITY 4-305
  - Description 4-308
  - Tables and Analytic Models. 4-309
  - Analytic Models 4-309
  - Tables or Model Selection 4-309
  - Example 4-309
- INTERSTITIAL. 4-311
  - Description 4-324
  - Bulk and Interface Parameters 4-325
  - Examples 4-325
- VACANCY 4-327

    Description 4-335
    Bulk and Interface Parameters 4-335
    Examples 4-335
  ANTIMONY 4-337
    Description 4-342
    See Also 4-342
    Examples 4-343
  ARSENIC. 4-344
    Description 4-349
    See Also 4-349
    Examples 4-350
  BORON 4-351
    Description 4-356
    See Also 4-356
    Examples 4-357
  PHOSPHORUS 4-358
    Description 4-363
    See Also 4-364
    Examples 4-364

Tutorial Examples 5-1

1-D Bipolar Example 5-2
  TSUPREM-4 Input File Sequence. 5-2
  Initial Active Region Simulation 5-3
  Mesh Generation 5-4
    Adaptive Gridding 5-4
  Model Selection 5-4
    Oxidation Model. 5-4
    Point Defect Model. 5-5
  Processing Steps. 5-5
    Buried Layer Masking Oxide 5-5
    Buried Layer. 5-5
    Epitaxial Layer 5-6
    Pad Oxide and Nitride Mask 5-6
  Saving the Structure 5-6
  Plotting the Results 5-6
    Specifying a Graphics Device 5-7
    The SELECT Statement 5-7
    The PLOT.1D Statement 5-7
    Labels 5-8
  Printing Layer Information 5-8
    The PRINT.1D Statement 5-8
    Using PRINT.1D LAYERS 5-9
  Completing the Active Region Simulation 5-9
    Reading a Saved Structure 5-11

    Field Oxidation 5-11
  Final Structure 5-12
Local Oxidation 5-13
  Calculation of Oxide Shape 5-13
    Mesh Generation 5-14
    Pad Oxide and Nitride Layers 5-15
    Plotting the Mesh 5-15
    Model Selection 5-15
    Plotting the Results 5-16
    Plotting Stresses 5-17
  2-D Diffusion with Point Defects 5-20
    Automatic Grid Generation 5-20
    Field Implant 5-20
    Oxidation 5-20
    Grid Plot 5-22
    Contour of Boron Concentration 5-22
    Using the FOREACH Statement 5-24
    Vertical Distribution of Point Defects 5-25
    Lateral Distribution of Point Defects 5-26
  Local Oxidation Summation 5-27
Point Defect Models 5-28
  Creating the Test Structure 5-31
    Automatic Grid Generation 5-31
    Outline of Example. 5-31
  Oxidation and Plotting of Impurity Profiles 5-31
    Simulation Procedure 5-31
    PD.FERMI and PD.TRANS Models 5-32
    PD.FULL Model. 5-32
    Printing Junction Depth 5-33
    Doping and Layer Information 5-33
  Point Defect Profiles 5-33
  Commentary 5-34
    Choosing a Point Defect Model 5-34

Advanced Examples 6-1

NMOS LDD Process 6-2
  Creating the Initial Structure 6-2
    Setting the Grid Density 6-3
    Adaptive Gridding 6-4
    Masking Information 6-4
  Field Isolation Simulation 6-4
    Displaying the Plot 6-5
  Active Region Simulation 6-7
    Modeling Polysilicon 6-7
    LDD Implant 6-8

Oxide Spacer and Source/Drain Implant 6-9
Source/Drain Contacts 6-9
Plots 6-11
Formation of the Complete NMOS Transistor 6-12
Electrical Extraction. 6-14
Threshold Voltage 6-14
MOS Capacitance. 6-15
Source/Drain Junction Capacitance 6-16
Plotting Results of Electrical Extraction 6-17
Trench Implant Simulation. 6-18
Structure Generation 6-19
Analytic Implant 6-20
Plotting the Results of the Analytic Method 6-21
MC Implant 6-23
Overview 6-23
Using the MC Model 6-23
Plotting the Results of the MC Method 6-24
Boron Contours 6-24
Vertical Profiles 6-25
Sidewall Profiles. 6-27
Summary 6-28
Poly-Buffered LOCOS 6-28
Structure Generation 6-28
Using the VISCOEL Model. 6-29
Plotting the Results 6-30
CMOS Process 6-32
Main Loop 6-34
Mesh Generation 6-34
CMOS Processing 6-36
Models 6-36
Channel Doping Plot 6-36
Lightly Doped Drain Structure. 6-36
Contacts 6-36
Saving the Structure 6-37
End of Main Loop 6-37
Plotting the Results 6-37
0.8 Micron Device 6-39
Final Mesh 6-39
Arsenic Profiles in Gate 6-39
1.2 Micron Device 6-40
DMOS Power Transistor 6-42
Mesh Generation 6-43
Processing the DMOS Power Transistor 6-43
Gate Processing 6-45
Source Processing. 6-46
Summary 6-48
SOI MOSFET 6-49

    Mesh Generation 6-49
        Depositing a Layer with Nonuniform Grid Spacing 6-50
    Process Simulation. 6-53
MOSFET with Self-Aligned Silicides 6-54
    Preparation for Silicidation 6-54
    Silicidation 6-55
Polysilicon Emitter Study 6-59
    Process Simulation. 6-59
        Models 6-59
        Processing. 6-59
    Plotting the Results 6-61
        After Implant 6-61
        Doping and Grain Size 6-63
        Doping vs. Stripe Width. 6-63
SiGe HBT Process 6-64
    Initial Structure and Collector Region Generation 6-65
    Using SiGe Related Models. 6-66
    Base and Emitter 6-73

User-Specified Equation Interface 7-1

Overview. 7-1
Using USEIT. 7-3
    EQUATION. 7-3
        New Solution Variable 7-3
        Solving Method 7-3
        Initialization 7-4
        Example for Initialization. 7-5
        Adding Expressions to Equations. 7-6
        Flux at Interfaces 7-6
        Diffusion Along Boundaries. 7-7
        Update Solution 7-7
    INTERMEDIATE 7-8
        Advantages of Intermediates 7-8
    Application of User-Specified Models 7-8
    Modification of Built-in Functions 7-9
        ACTIVE 7-9
        MOBILE. 7-9
    Built-in Keywords 7-10
    Operators 7-11
    Mathematical Functions. 7-11
    Physical Functions 7-13
    Interface Functions. 7-14
    Miscellaneous Functions 7-15
Examples. 7-15
    Interstitial Clustering Model 7-15

  Transient Clustering Model 7-16
  Dislocation-Loop Model 7-17
  Diffusion in PD.TRANS 7-19
  Interface Trap Model 7-20
  Boron-Interstitial Clustering Model with Small Clusters 7-21
  Diffusion Enhancement in Thinner Oxides 7-25
  Surface Recombination Rate Depending On Nitrogen 7-27
  5-Stream Diffusion Model 7-27
    Bulk Equations 7-28
    Boundary Conditions 7-29
    Implementation. 7-29
  Coupling with Small Clusters 7-35
  SiGe Model 7-38

Default Coefficients A-1

Default Coefficient Values A-1
  Impurity Parameters. A-2
  Impurity Diffusion Coefficients. A-2
  Segregation and Transport Coefficients. A-5
  Polysilicon Grain Segregation A-7
  Clustering and Solid Solubility A-7
  Point Defect Parameters. A-8
  Oxidation A-11
  Diffusion Enhancement in Oxides A-15
  Silicidation A-16
  Analytical Implant A-17
  MC Implant A-19
  Electrical Parameters A-21
  Material Coefficients A-23
  Numerical Methods A-26
  Automatic Grid Generation Parameters. A-27
  Adaptive Grid Parameters A-28
Default Coefficient References. A-29
Default Coefficient Notes A-34

Graphics Devices B-1

Determining the Device B-1
Output B-2
  Supported Devices B-2
    X (Window) B-2
    I/X. B-3
    X/BW B-3
    POSTSCRIPT (PS,PS-P) B-3

L/POSTSCRIPT (PS-L) B-3
PS-INSERT B-3
C/POSTSCRIPT (PS-C,PS-CP) B-4
CL/POSTSCRIPT (PS-CL) B-4
C/PS-INSERT. B-4
REPLOT B-4
HP2648 (2648) B-4
HP2623 (2623) B-4
TEK4100 (4100). B-4
TEKBW B-4
TEK4010 (4010). B-4
XTERM B-4
REGIS (VT240, VT241) B-4
HP7550 (7550) B-4
HP7550-P (7550-P) B-4
PRINTRONIX B-4
SELANAR B-5
SUN (SUNVIEW) B-5
I/SUN B-5
APOLLO B-5
I/APOLLO B-6
Unsupported Devices B-6
TEK4510 (4510). B-6
HPJET (THINKJET) B-6
HPDESK (HP2671G, HP2673) B-6
HPLP (LP2563) B-6
IMAGEN B-6
DITROFF B-6
TGPLOT. B-6
TGPLOT-P B-6
The Default Device. B-7
DEFAULT B-7
Modifying s4pcap B-7
PIXX and PIXY B-7
PUNX and PUNY. B-7
FILE B-7
LIKE. B-7
PEN and AREA B-8
BFSZ. B-8

Format of Mask Data Files **C-1**

Introduction. C-1
Format. C-1

Data Format of Saved Structure Files **D-1**

Introduction. D-1
TSUPREM-4 Structure File Format D-1
  Current Version D-2
    Program Identification D-2
    Coordinates. D-2
    Edges D-2
    Regions. D-2
    Triangles. D-3
    Model Parameters. D-3
    Solution Values D-5
    Nodes D-5
    Example D-5
  Older Versions of TSUPREM-4 . . . . D-7
    Regions. D-7
    Model Parameters. D-7
    Solution Values D-8
    Nodes D-8
Medici Structure File Format D-9
    Example D-10

Using the MINIMOS 5 Interface **E-1**

Overview of the TSUPREM-4 Interface to MINIMOS 5. . . . . E-1
Step 1: Directing TSUPREM-4 to Generate a Formatted File E-2
  Defining the MINIMOS 5 Simulation Region. E-2
  Notes on the Size of the MINIMOS 5 Simulation Region E-5
  Nonplanar Oxide Regions in MINIMOS 5 . . . . E-6
Step 2: Converting the Formatted File to FORTRAN Binary E-7
Step 3: Running MINIMOS 5. . . . . E-7
  Notes on Using MINIMOS 5 . . . . E-8
Interpreting Error Messages E-9

Enhancements **F-1**

Advanced Parameter Set Update F-2
Automatic Tune for Optimal Numerical Method F-2
Automatic Save at Abnormal Stop F-3

ACT.FULL for Dopant–Vacancy Clustering F-3
  Arsenic Clustering and Deactivation F-5
Enhancements in DDC.FULL. F-8
  DDC.FULL with PD.FULL F-9
  Use of DDC.EQ.0 and DDC.EQ.E F-10
  PAIR.EQU with DDC.FULL. F-13
  DDC.FULL with Vacancy F-15
    XS + XV. F-17
    V + XV F-18
    XV + XV F-19
    Xn-1Vm-1 + XV and Xn-1Im+1 + XV. F-20
    Xn-1Vm-1 + Vand Xn-1Im+1 + V. F-21
    In+1 + XV F-22
    XV2 + I. F-22
    XI2 + V. F-23
    XI + XV F-24
Initialization of Dopant-Defect Clusters. F-25
Pressure-Dependent Pair/Substitutional Formation Energy F-27
Adaptive Gridding at Amorphous and Crystalline Boundary F-29
Modifiable Hobler’s Damage Moments F-31
  Parameters Added to MOMENT Command F-31
  Parameters Added to IMPLANT Statement F-32
  Parameter Added to IMPURITY Statement F-32
  Implant Damage Moment File Format. F-32
Profiling Implant Damages. F-33
Dopant-Dependent Diffusion and Reaction of Oxidants F-37
  Parameters Added to the AMBIENT Command F-37
Gradient-Doped Epitaxial Growth F-39
  Parameters Added to the DEPOSITION and EPITAXY Statements F-39
  Parameter Changed on the DEPOSITION and EPITAXY Statements F-39
  New Behavior F-39
Strained Epitaxial Growth F-44
New Initialization Methods for Solutions. F-45
User-Defined Equilibrium Active Function F-47
INTERMED Snapshot F-48
INTERMED Scope Definition F-50
Fields to Store the Implant Dopant and Point-Defect Concentrations . F-52
Improved STRESS Command F-53
Taurus Monte Carlo Implant Updates F-54
TIF Format Change. F-54
Miscellaneous F-56

Glossary **Glossary-1**

Index **Index-1**

Introduction to TSUPREM-4

The chapter includes discussions of the following:

- Product Name Change
- Program Overview
- Processing Steps
- Simulation Structure
- Additional Features

Product Name Change

Synopsys process simulation tools, Taurus-Process and TSUPREM-4 have been consolidated into one product under the name Taurus-TSUPREM-4. The executables for both tools (process or tsuprem4) are included with Taurus-TSUPREM-4 distribution and you may choose either executable. If you own TSUPREM-4, you now have the right to run Taurus-Process in 2D. Taurus-Process 3D requires purchase of an upgrade to 3D.

Program Overview

TSUPREM-4 is part of the Taurus-TSUPREM-4 product which is comprised of the following tools:

- Taurus-Process
- TSUPREM-4

TSUPREM-4 is a computer program for simulating the processing steps used in the manufacture of silicon integrated circuits and discrete devices. TSUPREM-4 simulates the incorporation and redistribution of impurities in a two-dimensional

(2-D) device cross-section perpendicular to the surface of the silicon wafer. The output information provided by the program includes:

- Boundaries of the various layers of materials in the structure
- Distribution of impurities within each layer
- Stresses produced by oxidation, thermal cycling, or film deposition

Processing Steps

The types of processing steps modeled by the current version of the program include:

- Ion implantation
- Inert ambient drive-in
- Silicon and polysilicon oxidation and silicidation
- Epitaxial growth
- Low temperature deposition and etching of various materials

Simulation Structure

A TSUPREM-4 simulated structure consists of a number of regions, each of which is composed of one of a number of materials. Each material can be doped with multiple impurities. The materials available in TSUPREM-4 are single-crystal silicon, polycrystalline silicon, silicon dioxide, silicon nitride, silicon oxynitride, titanium, titanium silicide, tungsten, tungsten silicide, photoresist, aluminum, and user-defined materials. The available impurities are boron, phosphorus, arsenic, antimony, and user-defined impurities.

Additional Features

TSUPREM-4 also simulates the distribution of point defects (interstitials and vacancies) in silicon layers and their effects on the diffusion of impurities. The distribution of the oxidizing species in silicon dioxide layers is simulated to calculate oxidation rates.

Using TSUPREM-4

This chapter discusses starting TSUPREM-4, required input files, output files generated, and miscellaneous files required to execute the program. The chapter includes discussions of the following:

- Program Execution and Output
- File Specification
- Input Files
- Output Files
- Library Files

Program Execution and Output

This section describes execution of TSUPREM-4 and program output.

Starting TSUPREM-4

The execution of TSUPREM-4 is initiated with the command:

`tsuprem4` <*input filename*>

where the optional command-line argument, <*input filename*>, specifies the name of a TSUPREM-4 command input file.

If you specify<*input filename*>, TSUPREM-4 will execute the statements contained in the input file. If you leave the file specification blank, the program will respond by printing a header identifying the program version on the your terminal. You then will be prompted for the file specification of a command input file.

Note:

The file specification must conform to conventions in the operating system; it may not contain more than 80 characters.

If you do not specify a file in response to the prompt, the program enters interactive input mode, described in Documentation and Control, p. 4-5. In this case, you must enter the input statements from your terminal.

Program Output

Commands entered from the your terminal or from an input file are treated identically. In either case, the command will be read and displayed as soon as you have executed it. (See also, Errors, Warnings, and Syntax, p. 2-3.)

TSUPREM-4 generates both printed and graphical outputs that describe the simulation results. You can obtain all outputs generated before the termination of program execution. The locations of these outputs are described in Output Files, p. 2-5.

Printed Output

You can obtain the following printed output:

- Solution information (e.g., impurity concentrations) along vertical or horizontal lines through the structure or along material interfaces (**`PRINT.1D`** statement)
- Results produced by the **`EXTRACT`** statement
- Extracted electrical characteristics (e.g., sheet resistance) produced by the **`ELECTRICAL`** statement
- Summary of the mask information for each mask level (**`MASK`** statement)
- Summary of ion implantation parameters (**`IMPLANT`** statement)
- Informational and error messages to indicate the progress of the simulation

Graphical Output

You can obtain the following graphical output:

- Plots of solution values along a line through the structure or along a material interface (**`PLOT.1D`** statement)
- Two-dimensional plots of the structure, showing material boundaries, simulation grid, contours of impurity or point defect concentrations, or growth velocity and stress vectors (**`PLOT.2D`** statement)
- Three-dimensional (“bird’s-eye view” or “surface projection”) plots of solution values (**`PLOT.3D`** statement)
- Plots of electrical parameters, such as capacitance or channel conductance vs. bias voltage (**`PLOT.1D`** statement)
- Plots of user-specified data (e.g., for comparing measured and simulated profiles)

Solutions can also be saved for later analysis with graphical post-processing programs.

Errors, Warnings, and Syntax

If an error is detected in processing a command, a message is printed. If commands are being read from a file, no-execute mode is set, and the remainder of the statements in the input file are checked for syntax, but not executed.

The program also prints warning messages. Warnings are not fatal and serve only to indicate potential problems that you should be aware of. Warning messages normally indicate any corrective action taken automatically by the program.

File Specification

In this manual, file names are highlighted by printing them in an *italic* font. Lowercase names are used for input files, library files, and plot files, while uppercase names are used for saved structure files.

File Types

Files used by TSUPREM-4 are grouped into two categories:

- Files known to the program (library files)
- Files specified by the user

Files known to the program (e.g., *s4init*, and *s4pcap*) have names assigned by *Synopsys TCAD*. You can change the file names when you execute the program or have the system administrator who installs the program at your site change them.

You can specify the following files:

- Command input files
- Plot output files
- Saved solution files

You can use any names for these files provided that they conform to the file naming conventions of the operating system.

Default File Names

The default names for output listing files are derived from the name of the command input file, if one was specified on the **`tsuprem4`** command line or in response to the file name prompt. This allows multiple copies of TSUPREM-4 to be executed simultaneously (using different command input files) in a single directory, without encountering naming conflicts among the output files. The output file names are derived by removing the extension (the last "." and any following characters in the file name), if any, from the input file name and adding the extensions *.out, .inf,* and *.dia* for the output, informational, and diagnostic output files, respectively. If no input file name was specified (i.e., TSUPREM-4 is being run in interactive mode), you may use the default names *s4out, s4inf,* and *s4dia*.

Environment Variables

Environment variables can be used to override the default values for library file names, standard file identifiers, and graphics output device names (see Appendix B). A unique environment variable is associated with each file. These environment variables are described in the remainder of this chapter in the descriptions of the file identifiers they control.

The following environment variables are used in TSUPREM-4:

S4OUT	Standard output file identifier p. 2-6
S4INF	Informational output file identifier p. 2-6
S4DIA	Diagnostic output file identifier p. 2-6
S4FKY0	Formatted key file identifier p. 2-10
S4UKY0	Unformatted key file identifier p. 2-10
S4INIT	Initialization input file identifier p. 2-9
S4IMP0	Ion implant data file identifier p. 2-9
S4IDAM	Ion implant damage data file identifier p. 2-9
S4ADVANCED	Advanced parameter file identifier p. 2-10
S4PCAP	Plot device definition file identifier p. 2-10
S4AUTH	Authorization file identifier p. 2-10
DEFPDEV	Graphics output device name p. 2-7 and Appendix B

Input Files

TSUPREM-4 requires several types of input files. You will supply one or more command input files. In addition, the program can read solution files generated by previous TSUPREM-4, Taurus-Lithography simulations, mask data files, and library files. TSUPREM-4 also reads Technology Interchange Format (TIF) files for use with Taurus Workbench, Michelangelo, and Taurus Topography.

Command Input Files

Command input files contain statements that direct the TSUPREM-4 simulation. These are text files that can be created and modified using any text editor. In interactive applications, your terminal serves as the primary command input file, while secondary command input files are specified with the **`SOURCE`** statement.

Descriptive names can be used for specialized command input files—process description files, coefficient files, and simulation control files are examples of special-purpose command input files. A detailed description of the valid input statements and their proper format is provided in Chapter 4.

For convenience when using the STUDIO visualization program, end command input file names with the extension *.inp*.

Mask Data Files

Mask data files are created by Taurus-Layout. These files contain the mask information from a cross section of a mask layout. Mask data files are formatted text files; the format of these files is described in Appendix C. By convention, the extension *.tl1* is used for the mask layout files used by TSUPREM-4.

Profile Files

Profile files contain doping profile information to be read by the **`PROFILE`** statement. Profile data can also be plotted with the **`PLOT.1D`** statement or used as a target for optimization. The format of these files is described in Chapter 4, PROFILE, p. 4-86.

Other Input Files

The following files also are read by TSUPREM-4; they are described elsewhere in this chapter:

- TSUPREM-4 structure files contain saved solution information. (See Program Output, p. 2-2.)
- The *s4init* library file contains commands to initialize the model coefficients used by TSUPREM-4. (See Appendix B.)
- The *s4compat64* file contains commands to modify the model coefficients to make TSUPREM-4 version 6.5 give the same results as version 6.4.
- The *s4imp0* library file contains ion implant range statistics for use by the **`IMPLANT`** statement. (See Ion Implant Data File—s4imp0, p. 2-9.)
- The *s4pcap* library file defines the characteristics of the various graphical output devices available to TSUPREM-4. (See Plot Device Definition File—s4pcap, p. 2-10 and Appendix B.)
- The *s4auth* library file contains information on the computer systems for which TSUPREM-4 has been licensed. (See Authorization File—s4auth, p. 2-10.)

Output Files

TSUPREM-4 produces a variety of printed and graphical output and data files describing the simulation results. The various types of output are described in the remainder of this section.

Terminal Output

The standard and error output streams normally appear at your terminal, but in some computing environments they can be redirected to a file, or appear both at your terminal and in an output file. The standard output consists of output from the `PRINT.1D` command as well as informational, error, and warning messages generated by many commands. If `ECHO` is set with the `OPTION` command (it is set by default), the input statements are also printed on the standard output.

The error output usually will be interspersed with the standard output. The error output receives error messages generated by the program. In interactive mode, input prompts are also sent to the error output. (See also Errors, Warnings, and Syntax, p. 2-3.)

Output Listing Files

This section describes the various output listing files.

Standard Output File—*s4out*

A record of each TSUPREM-4 execution is sent to the output listing file. The default name for this file is derived from the name of the input file, but use of the environment variable *S4OUT* overrides this name during program execution (see File Specification, p. 2-3). This text file includes a listing of all input statements, error messages, and printed output produced by the program.

Informational Output File—*s4inf*

Additional information produced by TSUPREM-4 can be sent to the informational output file. The default name for this file is derived from the name of the input file, but the environment variable *S4INF* can be used to override this name during program execution. (See File Specification, p. 2-3.)

This text file can be useful in understanding the operation of the program, but it is not normally of interest to you. Output to the informational file can be enabled or disabled using the `INFORMAT` keyword on the `OPTION` statement; by default, it is disabled.

Diagnostic Output File—*s4dia*

Diagnostic information produced by TSUPREM-4 can be sent to the diagnostic output file. The default name for this file is derived from the name of the input file, but the environment variable *S4DIA* can be used to override this name during program execution. (See File Specification, p. 2-3.)

This text file receives diagnostic information on the internal operation of the program, and is not normally of interest to you. Output to the diagnostic file can be enabled or disabled using the `DIAGNOST` keyword on the `OPTION` statement; by default, it is disabled.

Saved Structure Files

The structure and impurity distributions can be saved in a number of formats. The name of a saved structure file is specified by the **`OUT.FILE`** parameter on the **`SAVEFILE`** statement. All files are written in text format, so they can be transferred easily between hardware platforms. Some structure files can both be read and written by TSUPREM-4. The structure file formats used by TSUPREM-4 are described in the following sections.

TSUPREM-4

This is the primary file format for saving structures and solution information for later use in TSUPREM-4. This is the default format produced by the **`SAVEFILE`** statement. TSUPREM-4 structure files contain complete structure and solution information plus some model specifications. They can be read with the **`INITIALIZE`** and **`LOADFILE`** statements and used as a basis for further simulations. (See Appendix B for the format of TSUPREM-4 structure files.)

TIF

TIF files are used to communicate with Taurus-WorkBench, Michelangelo, Taurus Topography, and Taurus Visual. TIF files are produced by specifying the **`TIF`** parameter on the **`SAVEFILE`** statement. TIF files contain complete structure and solution information plus some model specifications. They can be read by TSUPREM-4 with the **`INITIALIZE`** and **`LOADFILE`** statements and used as a basis for further simulations. TIF files also can be used to pass structures to Medici.

Medici

medici structure files are used to communicate with Synopsys TCAD's Medici device simulation program. They contain the full physical structure plus net and total doping concentrations. medici structure files can also be read by many versions of PISCES. TIF files also can be used to pass structures to Medici. (See Appendix D for the format of Medici structure files.)

MINIMOS 5

Structure files can also be created with a format that can be read into the MINIMOS 5 device simulation program from the Technical University of Vienna. These files contain the structure and doping information needed by MINIMOS 5. (See Appendix D for the use of MINIMOS 5 structure files.)

Wave

Solution data can be saved in *wave* format for later graphical display using Wavefront Technologies' Data Visualizer program. The format of these files is described in *The Data Visualizer Version 2.0 Programming Guide* from Wavefront Technologies.

Graphical Output

Graphical output is sent to the device determined by the **`DEVICE`** parameter on the **`OPTION`** statement or to a default output device (see OPTION, p. 4-41). This is typically the user's terminal, but some graphics devices use output files specified in the *s4pcap* file (see Plot Device Definition File—s4pcap, p. 2-10).

The device names that can be specified are defined in the *s4pcap* file. Drivers are available for a variety of devices, including graphics terminals, pen plotters, and laser printers. (See Appendix B for a list of available devices is given in Appendix B.)

TSUPREM-4 selects a graphics output device by the following process:

1. If a valid **DEVICE** parameter has been specified on an **OPTION** statement, its value is used as the device name.
2. If the *DEFPDEV* environment variable specifies a valid device name, that device is used.
3. If the *TERM* environment variable specifies a valid device name, that device is used.
4. If none of the above steps produces a valid device name, the *default* device in the *s4pcap* file is used. The *default* device can be linked to any device in *s4pcap;* in the *s4pcap* file shipped by *Synopsys TCAD*, the *default* device is equivalent to *ps*, which produces files in PostScript format.

This selection process occurs the first time that plotting is requested in a job.

Extract Output Files

The **EXTRACT** statement (see EXTRACT, p. 4-180) allows printing of arbitrary device structure information such as layer thicknesses and impurity concentrations. Extracted information is sent to the file specified by the **OUT.FILE** parameter on the **EXTRACT** statement. The formatting features of the **EXTRACT** statement allows the flexible combination of text and data in the output file.

Electrical Data Output Files

The **ELECTRICAL** statement saves extracted electrical characteristics in the file specified by the **OUT.FILE** parameter. (See ELECTRICAL, p. 4-194.)

Library Files

The following files are used for specific purposes by TSUPREM-4 and, in most cases, you will not need to reference or modify these files directly. The names of these files are predefined, but can be overridden by specifying appropriate environment variables. (See Environment Variables, p. 2-4.) These files are typically installed in a library common to all users of the program, but you can use your own customized version, if you wish.

CAUTION

TSUPREM-4 will not run correctly if these files are missing or inaccessible.

Initialization Input File—*s4init*

The initialization input file contains simulation input statements that specify the default coefficients for the materials, impurities, and numerical models in TSUPREM-4. This file is read automatically each time TSUPREM-4 is executed. The default name for this file is *s4init,* but the environment variable *S4INIT* can be used to override this name during program execution. Default values of coefficients and model parameters can be changed by modifying this file. It is a text file, and can be modified with any text editor or with the studio TSUPREM-4 Command Editor.

Note:

> ***Normally, all files read by TSUPREM-4 must be accessible by the user who runs the program. An exception is made for s4init: If the set-user-ID (or set-group-ID) mode is set for the TSUPREM-4 executable file (and set-user-ID execution is allowed by the file system containing the executable), then s4init may be owned by the user (or group) that owns the TSUPREM-4 executable. Thus, it is possible for TSUPREM-4 to use an s4init file that cannot otherwise be read by the user.***

Ion Implant Data File—*s4imp0*

The ion implant data file defines the range statistics for the ion implantation of impurities in various materials. The default name for this file is *s4imp0,* but the environment variable *S4IMP0* can be used to override this name during program execution. This file is in text format, and can be modified by any standard text editor. The data in the file is formatted in a manner that is defined in the file. Range statistics data for any number of materials can be represented. For each material, data for several impurity ions is present, with range statistics listed for a sequence of implantation energies for each ion.

Ion Implant Damage Data File—*s4idam*

The ion implant damage data file defines the range statistics for the ion implantation damages of impurities in silicon. The default name for this file is *s4idam,* but the environment variable *S4IDAM* can be used to override this name during program execution. This file is in text format and can be modified by any standard text editor. The data in the file is formatted in a manner that is defined in the file. Range statistics data for any number of tables can be represented. A file format is defined to contain tables of parameters for the damage model. The format of the implant damage file is the same as that used by Taurus, except:

- Multiple tables may be specified in a file. A line of the form:

  ```
  table <tablename>
  ```

  where <*tablename*> is the name specified by `DAM.TABL` which must appear before the header line in each table.
- The only implant condition recognized in the header line is “energy”.

- Blank lines may be inserted between tables.

The data values are the same as in the Taurus file; i.e., range, sigma, lateral spread, dl (depth dependence of lateral spread), asym (asymmetry of vertical profile; not used), kurt (kurtosis of vertical profile; not used), lambda, and (scale) factor. This format allows for expansion to include other implant conditions (e.g., tilt) at a later time. It also allows the existing data in Taurus to be copied without changes for use in TSUPREM-4.

Plot Device Definition File—*s4pcap*

The plot device definition file contains the information required to set up and use various graphics output devices. The default name for this file is *s4pcap,* but the environment variable *S4PCAP* can be used to override this name during program execution. The file is in standard text format, and can be modified by any text editor. The format is not intended to be self-explanatory. (See Appendix B for more information on this file.)

Key Files—*s4fky0* and *s4uky0*

The files *s4fky0* and *s4uky0* define the statement names, parameter names, and default values used by TSUPREM-4, and are used to check the syntax of the command input file. They contain identical information in different formats: The file *s4fky0* is in text format, and can be modified by any standard text editor. The file *s4uky0* is in binary format, and can be used more efficiently than *s4fky0* during the syntax check. *s4uky0* is generated from *s4fky0* during the initial installation of TSUPREM-4. Only the unformatted key file *s4uky0* needs to be available when the program is executed. *Note* that statement and parameter names must agree with names coded in the program source, and cannot be changed simply by modifying the key files. The environment variables *S4FKY0* and *S4UKY0* can be used to override the default names of these files during program execution.

Authorization File—*s4auth*

The authorization file *s4auth* contains authorization values that enable the execution of TSUPREM-4. If this file contains invalid authorization values, an error will be displayed, indicating that the program is not authorized for execution on the machine. *Synopsys TCAD* must be contacted for assistance in correcting this problem. The file *s4auth* is a text file. The environment variable *S4AUTH* can be used to override the default name of this file during program execution. The *s4auth* file is not required by versions of TSUPREM-4 that use a license management program.

TSUPREM-4 Models

This chapter describes the modeling capabilities of TSUPREM-4 for the analysis of fabrication processing steps. Discussions include the following:

- Simulation Structure
- Grid Structure
- Diffusion
- Oxidation
- Silicide Models
- Stress Models
- Ion Implantation
- Epitaxial Growth
- Deposition
- Masking, Exposure, and Development of Photoresist
- Etching
- Modeling Polycrystalline Materials
- Electrical Calculations
- Dislocation Loop Model
- References
- Equations used to model the physical processes are discussed in terms of the statement parameters documented in Chapter 4.
- For a more detailed discussion of the physical basis of many of the models described in this chapter, refer to the Stanford University SUPREM-III Technical Report Reference [1].

Simulation Structure

A TSUPREM-4 simulation represents a 2-D cross-section of a portion of a semiconductor wafer.

Coordinates

Usually, the x coordinate represents distance parallel to the surface of the wafer, and the y coordinate corresponds to depth into the wafer. In plots of the structure, x increases from left to right, and y increases from top to bottom. In specialized applications, the y coordinate also lies parallel to the surface of the wafer, giving a simulation space in the plane of the wafer surface.

The coordinate system is defined by the user and is fixed relative to the initial structure, meaning that the coordinate system is tied to the substrate, not to the wafer surface.

Initial Structure

The initial structure is defined as a rectangular region of arbitrary (user-defined) width and depth. By default, the top of the structure is exposed, and reflecting boundary conditions are applied to the sides and bottom. However, this can be changed using the `BOUNDARY` statement (see Chapter 4, BOUNDARY, p. 4-62). Deposition, etching, impurity predeposition, oxidation, silicidation, reflow, and out-diffusion occur at exposed surfaces, while photolithographic exposure and ion implantation always occur at the top surface.

Note:

> ***There are restrictions on these processes when surfaces other than the top surface are exposed; see the description of the*** `BOUNDARY` ***statement*** (Chapter 4, BOUNDARY, p. 4-62)***) for details.***

Regions and Materials

The structure is composed of from one to forty regions of arbitrary shape. Each region consists of a single material. By definition, adjacent regions (i.e., regions that meet along an edge) contain different materials. The same material can be present in multiple (non-adjacent) regions.

Grid Structure

The continuous physical processes modeled by TSUPREM-4 are approximated numerically using finite difference (for diffusion) and finite element (for oxide flow) solution techniques.

Mesh, Triangular Elements, and Nodes

Each region of the structure is divided into a mesh of nonoverlapping triangular elements. There can be up to 200,000 triangles in a TSUPREM-4 mesh. Solution values are calculated at the mesh nodes at the corners of the triangular elements. At points where two or more materials meet, there are multiple solution values (multiple nodes), one for each material at the meeting point. On an exposed boundary, there is also an extra node at each point, which represents concentrations in the ambient gas.

The total number of nodes in a structure is calculated by adding the number of mesh points in each material, plus the number of mesh points along exposed boundaries. There can be up to 100,000 nodes in a TSUPREM-4 mesh. Additional nodes may be required on a temporary basis when simulating process steps that modify the structure. The total of the user-defined nodes plus temporary nodes must not exceed the maximum of 100,000. In addition, the initial grid must not have more than 1000 grid lines in either the *x* or *y* direction.

Defining Grid Structure

A grid structure must be defined before process simulation can start. You define an initial grid structure by explicitly specifying the locations and spacing of grid lines or, automatically, by generating a grid given the width and (optionally) the locations of mask edges. Once an initial grid has been defined, it is adjusted automatically as various process steps are simulated.

The statements that generate and modify the grid have parameters to control the grid spacing. All grid spacing specifications are multiplied by the value of the **`GRID.FAC`** parameter on the **`MESH`** statement.

Explicit Specification of Grid Structure

You can specify a nonuniform rectangular grid that is modified by removing portions of some grid lines. The modified rectangular grid is then converted to a triangular grid by adding diagonals. The nonuniform rectangular grid is specified by means of the **`LINE`**, **`BOUNDARY`**, and **`REGION`** statements and modified with the **`ELIMINATE`** statement. The triangular grid is produced by the **`INITIALIZE`** statement (see Chapter 4, INITIALIZE, p. 4-66 for complete descriptions of these statements).

The LINE Statement

The **LINE** statement is used to specify a series of grid lines. The location of each line is given by the **LOCATION** parameter, and the spacing is specified with the optional **SPACING** parameter. Grid lines must be specified in order of increasing **LOCATION**. The result is a set of locations:

$$x_1,\ x_2\ \ldots,\ x_m \tag*{Equation 3-1}$$

and spacings:

$$h_1,\ h_2\ \ldots,\ h_m \tag*{Equation 3-2}$$

for the user-specified grid lines. If a **SPACING** is specified for line i on the **LINE** statement, then h_i is given by:

$$h_i = \mathtt{SPACING} \times \mathtt{GRID.FAC} \tag*{Equation 3-3}$$

otherwise, h_i is taken as:

$$h_i = \min(x_{i+1} - x_i,\ x_i - x_{i+1}) \times \mathtt{GRID.FAC} \tag*{Equation 3-4}$$

Generated Grid Lines

Grid lines are added between the user-specified lines based on the locations and spacings of the user-specified lines and on the value of the **RATIO** parameter on the **INITIALIZE** statement. The goal is to choose a ratio r and number of spaces n that satisfy the equations:

$$x_{i+1} - x_i = \frac{r^n - 1}{r - 1} h_i \tag*{Equation 3-5}$$

$$h_{i+1} = h_i r^{n-1} \tag*{Equation 3-6}$$

subject to the constraints:

$$n \text{ is an integer} \tag*{Equation 3-7}$$

$$\frac{1}{\mathtt{RATIO}} \le r \le \mathtt{RATIO} \tag*{Equation 3-8}$$

A total of $n - 1$ grid lines is added with spacings:

$$h_i,\ h_i r,\ h_i r^2,\ \ldots,\ h_i r^{n-2} \tag*{Equation 3-9}$$

To satisfy the constraints, it is usually necessary to modify the values of h_i and h_{i+1}. Equation 3-8 is satisfied by reducing the larger of h_i and h_{i+1}, as needed. Equation 3-7 is then satisfied by increasing or decreasing both h_i and h_{i+1} by the same factor. The factor is chosen so that the final value of n is the nearest integer to the value obtained by solving Equations 3-5 and 3-6 exactly, with the constraint Equation 3-8 on r. *Note* that the final spacings h_i and h_{i+1} may be slightly greater than their user-specified values, and the ratio r may be slightly larger than **RATIO** or smaller than $1/$ **RATIO** to satisfy the constraint that n be an integer.

Two special cases should be noted: If h_i and h_{i+1} are both greater than or equal to the distance $x_{i+1} - x_i$, then no grid lines are added between x_i and x_{i+1}; and if h_i and h_{i+1} are equal, then uniformly spaced lines are added.

The x and y grids are generated independently, using the same algorithms.

Eliminating Grid Lines

The **ELIMINATE** statement (see Chapter 4, ELIMINATE, p. 4-59) can be used to thin out the grid in user-specified portions of the structure. The user specifies a rectangular subregion of the structure and whether vertical grid lines (**COLUMNS**) or horizontal grid lines (**ROWS**) are to be eliminated. The program then removes every other grid line in the specified direction within the specified region. Each additional **ELIMINATE** statement that specifies the region removes half of the remaining grid lines.

In some instances, it is not possible to eliminate grid lines when the specified region overlaps the region specified on a previous **ELIMINATE** statement. In these cases, a warning is printed and the elimination is not performed. The **ELIMINATE** statement is guaranteed to work when the specified region is the same as, totally included in, or totally separate from the regions specified on previous **ELIMINATE** statements.

The **ELIMINATE** statement is used during initialization to create a structure. It cannot be used to remove grid lines after initialization. The only way to remove grid lines after structure initialization is with **STRUCTURE UNREFINE**.

Automatic Grid Generation

If no **LINE X** statements are specified, grid lines in the *x* direction are generated automatically. Similarly, if no **LINE Y** statements are specified, the vertical grid is generated automatically. Automatic grid generation in either direction can be combined with manual grid generation in the other direction.

Automatic grid generation is controlled by parameters on the **MESH** statement and by the **WIDTH** and **DX** parameters on the **INITIALIZE** statement. Automatic grid generation in the *x* direction also depends on the location of mask edges obtained from mask information read with the **MASK** statement.

Automatic grid generation is intended for applications in which fine manual control over the grid is not needed. It is especially useful when mask information from Taurus Layout—IC Layout Interface is available.

Note:

> ***Specifications for automatic grid generation can be put in the s4init file, allowing advanced users of TSUPREM-4 to set process-specific defaults for use by less experienced users.***

Automatic Grid Generation in the X Direction

A grid in the *x* direction (i.e., a set of vertical grid lines) is generated automatically if an **INITIALIZE** statement without an **IN.FILE** is processed and no **LINE X** statements have been specified since the last **INITIALIZE** statement.

Placement of grid lines in the *x* direction is controlled by the **DX.MIN**, **DX.MAX**, and **DX.RATIO** parameters on the **MESH** statement and the **WIDTH** and **DX** parameters on the **INITIALIZE** statement. Automatic elimination of vertical grid lines is controlled by the **LY.SURF** and **LY.ACTIV** parameters on the **MESH** statement.

X Grid from WIDTH Parameter

The **WIDTH** parameter on the **INITIALIZE** statement specifies the width of the device. The grid spacing in the *x* direction is specified by the **DX** parameter on the **INITIALIZE** statement or by **DX.MAX** on the **MASK** statement if **DX** is not specified.

X Grid from MASK Statement

If the **WIDTH** parameter is not specified, but a **MASK** statement has been used to read mask information, the locations of mask edges are used to guide generation of the grid. A line is placed at each mask edge, with spacing given by:

$$h_{min} = \mathbf{DX.MIN} \times \mathbf{GRID.FAC} \qquad \text{Equation 3-10}$$

To either side of the mask edge, the grid spacing increases by a factor of **DX.RATIO** until a spacing of:

$$h_{max} = \mathbf{DX.MAX} \times \mathbf{GRID.FAC} \qquad \text{Equation 3-11}$$

is reached, or until a point halfway between two mask edges is reached. A spacing of h_{max} is used far from mask edges.

If the **GRID** parameter is used on the **MASK** statement, only edges on the specified mask levels are used for grid generation. If the **G.EXTENT** parameter is used, the spacing h_{min} is used for the specified distance under the mask line before increasing towards h_{max}.

Column Elimination

If no **ELIMINATE COLUMNS** statements are specified, a default set of eliminate operations is performed on the vertical grid lines. One elimination is done between:

$$y = \frac{1}{2}(\mathbf{LY.SURF} + \mathbf{LY.ACTIV}) \qquad \text{Equation 3-12}$$

and the bottom of the structure. Eight eliminations are done between **LY.ACTIV** and the bottom of the structure. These eliminations are intended to remove all but two grid lines below **LY.ACTIV**.

If the **WIDTH** parameter is not specified and if no mask information has been read with a **MASK** statement, vertical grid lines are placed at *x*=0 and *x*=1 micron producing a grid for 1-D simulations.

Automatic Grid Generation in the Y Direction

A grid in the *y* direction (i.e., a set of horizontal grid lines) is generated automatically if an **INITIALIZE** statement without an **IN.FILE** is processed and if no **LINE Y** statements have been specified since the last **INITIALIZE** statement.

Automatic grid generation in the *y* direction is controlled by the **LY.SURF**, **DY.SURF**, **LY.ACTIV**, **DY.ACTIV**, **LY.BOT**, **DY.BOT**, and **DY.RATIO** parameters on the **MESH** statement.

For grid generation, the structure is divided into three regions.

- The surface region extends from *y*=0 down to **LY.SURF**, and has grid spacing **DY.SURF** × **GRID.FAC**. The surface region has the finest grid spacing. It normally contains shallow implants and the channels of MOS transistors.
- The active region extends from **LY.SURF** down to **LY.ACTIV** and has a grid spacing that varies smoothly from **DY.USRF** × **GRID.FAC** to **DY.ACTIV** × **GRID.FAC**. The active region extends to below the deepest junctions in the structure.
- The substrate region extends from **LY.ACTIV** to **LY.BOT** and has a grid spacing that starts at **DY.ACTIV** × **GRID.FAC**. It increases by factors of **DY.RATIO** until it reaches a spacing of **DY.BOT** × **GRID.FAC**. The substrate region is very deep, but has few vertical grid lines. It provides for accurate modeling of point defect recombination.

Note:

> ***The default values for the automatic grid generation parameters are typical of those required for simulating small-geometry MOS processes. Especially when using bipolar or power processes, you are encouraged to customize the default values for your needs.***

Changes to the Mesh During Processing

The initial grid applies to the structure, not to the space containing the structure. Thus, processing steps that change the device structure must necessarily change the mesh structure as well. The processing steps that change the grid are deposition, epitaxy, etching, photoresist development, oxidation, and silicidation. In addition, if adaptive gridding is enabled, the grid may be modified during ion implantation and diffusion. The structure can also be truncated, reflected, or extended using the **STRUCTURE** statement.

DEPOSITION and EPITAXY

The **DEPOSITION** and **EPITAXY** statements deposit a conformal layer on the exposed surface of the structure (see Chapter 4, DEPOSITION, p. 4-99 and Chapter 4, EPITAXY, p. 4-137). The grid distribution normal to the exposed surface is controlled by four parameters:

- Layer thickness (**THICKNES**)
- Nominal grid spacing (**DY**)

- Location of the nominal grid spacing relative to the top surface of the layer (**YDY**)
- Number of grid spaces in the layer (**SPACES**)

The thickness must always be specified. Effects produced by various combinations of parameters are as follows:

- If none of the other parameters is specified, a single grid space is placed in the layer.
- If **SPACES** or **DY** (but not both) is specified, a uniform grid spacing of **THICKNES**/**SPACES** or **DY** (multiplied by **GRID.FAC**) is used.
- If both **SPACES** and **DY** are specified, a nonuniform grid spacing is used, with a spacing of **DY** × **GRID.FAC** at a depth **YDY** below the surface and spacing that increases or decreases by a constant ratio for a total of **SPACES**/ **GRID.FAC** grid spaces. The grid spacing varies linearly from the value at **YDY** to the top and bottom of the layer, increasing or decreasing at the same rate on either side of **YDY** to produce the required number of grid spaces. Not all combinations of **SPACES**, **DY**, and **YDY** will produce a valid grid; in particular, the average grid spacing can never be less than half of **DY**.

Normally, the grid spacing parallel to the exposed surface is the same as the spacing along the original surface. However, when the grid spacing perpendicular to the surface is large compared to the parallel spacing, the parallel spacing is increased to avoid grid elements with low ratios of width to height. By default the width/height ratio is constrained to be at least 0.5, but this value can be changed with the **AR.MIN** parameter on the **DEPOSITION** or **EPITAXY** statements. Exposed corners on the original surface produce arcs at the surface of the deposited layer. These arcs are approximated by straight segments of length **ARC.SPAC** × **GRID.FAC** or smaller.

Structure Extension

The **STRUCTURE** statement with the **EXTEND** parameter works similarly to deposition, except that grid is added horizontally to one edge of the structure; the parameters **WIDTH**, **DX**, and **XDX** are used in place of **THICKNES**, **DY**, and **YDY**. *Note* that **XDX** in this case is the absolute *x* coordinate at which the grid spacing specified by **DX** applies.

ETCH and DEVELOP

The **ETCH** and **DEVELOP** statements remove a specified portion of the structure. The grid is modified in two steps:

1. Nodes are added along the etch boundary. Nodes are placed at points where the etch boundary intersects the boundary of the material to be removed. Within the material being removed, the spacing of the added nodes is set to the smaller of the grid spacing of the points defining the etch boundary and the grid spacing in the material being removed.
2. Mesh elements inside the etch boundary are removed.

Oxidation and Silicidation

The moving silicon/oxide interface and the material expansion produced during oxidation require continuous modifications to the simulation mesh. Three kinds of grid manipulation are required:

- Removal of nodes in consumed layers
- Addition of nodes in a growing layer (e.g., during oxidation)
- Removal or rearrangement of nodes in regions where oxide is deforming

The following descriptions are in terms of oxidation, but apply equally to silicidation and other reactions. The only difference is that **`DY.DEFAU`** is used for the growing material instead of **`DY.OXIDE`** and **`GRID.OXI`**. This difference has no effect for any processes other than oxidation.

Removal of Nodes in Consumed Silicon

As oxidation occurs, the silicon/oxide interface advances into the silicon, while the growing oxide expands away from the silicon. Nodes (both silicon and oxide) on the interface move with the interface; nodes in the silicon interior remain fixed, while nodes in the oxide move with the oxide flow. As a consequence, interface nodes are continuously moving towards silicon nodes, which must be removed to allow the interface to advance.

Addition of Nodes in a Growing Layer

On the oxide side of the interface, the triangular mesh elements are expanding. To maintain solution accuracy in the oxide (for example, calculating the diffusion of oxidant in the oxide), it is necessary to add nodes to the oxide. The addition of nodes to the oxide is controlled by the **`DY.OXIDE`**, **`DY.EXACT`**, **`DY.LOCAL`**, and **`GRID.OXI`** parameters on the **`METHOD`** statement. (See Chapter 4, METHOD, p. 4-207.)

Precise grid spacing is obtained by adding new nodes in a growing layer at the distance specified by **`DY.OXIDE`** × **`GRID.FAC`** from an existing node in the layer. Because only one node can be added at each point on an interface during a simulation time step, the size of the time steps may need to be reduced to achieve the desired spacing. This reduction in the time step can be disabled by specifying **`^DY.EXACT`** on the **`METHOD`** statement. By default, **`DY.EXACT`** is set to allow cutting of the time steps to control the grid spacing. The algorithm will not allow grid points to be added at spacings less than 1Å, and control of the spacing may not be precise for spacings less than 2Å.

The grid control algorithms and parameters apply to the entire structure. To avoid adding very fine grid in field regions when growing gate oxides, an option has been added to allow the grid spacing to vary with the oxide growth rate. When **`DY.LOCAL`** is specified on the **`METHOD`** statement, the grid spacing to be used at each point in the growing oxide is:

$$h = \frac{v_{max}}{v}\mathtt{DY.OXIDE} \times \mathtt{GRID.FAC} \qquad \text{Equation 3-13}$$

where v is the growth rate at a point in the structure and v_{max} is the maximum growth rate at all interfaces of the same type in the structure. (**`DY.DEFAU`** is used instead of **`DY.OXIDE`** for reactions other than oxidation.) **`DY.LOCAL`** is set true by default.

If **`DY.OXIDE`** is zero, an extra node is added when the spacing in the oxide is approximately equal to the spacing in the silicon multiplied by the value of **`GRID.OXI`**. For **`GRID.OXI`** $= 2.2$ (the silicon to silicon/dioxide volumetric expansion ratio) you might expect the number of nodes added to the oxide to equal the number of nodes removed from the silicon. Actually, more nodes are added to the oxide, because the apparent spacing of nodes in the silicon is reduced by the movement of the interface.The **`DY.OXIDE`** parameter is the preferred means of controlling grid spacing in the oxide. The **`GRID.OXI`** parameter is considered obsolete, and is retained only for compatibility with older versions of the program.

Nodes in Regions Where Oxide is Deforming

The flow of oxide may not be uniform when masking layers are present. At the edge of a nitride layer, for example, two corners of a triangular mesh element may be constrained by the presence of the nitride, while the third is being carried by the flow of oxide out from under the mask. In such situations, mesh triangles can become severely distorted, and could prevent further oxidation unless appropriate remedies are applied, such as *flipping* triangles and removing nodes or triangles in congested portions of the mesh. (In flipping triangles, the line common to two adjacent triangles is removed to form a quadrilateral, then a line is added to divide the quadrilateral into two triangles using the opposite two corners.)

In some cases, it may be necessary to eliminate a very thin triangle of one material that protrudes into another material. To avoid creating a hole in the structure, the material type of the triangle is changed to that of the surrounding structure.

Numerical Integrity

After any modification to the mesh, a check is made for triangles that might lead to numerical difficulties (i.e., loss of accuracy or poor convergence). Where possible, these triangles are eliminated by adjusting the triangularization or by adding nodes. Where such adjustment is not possible or would lead to large numerical errors, the discretization of the triangle is modified to avoid numerical instability.

Adaptive Gridding

To reduce the effort required to set up an initial grid and to improve simulation accuracy, TSUPREM-4 can perform adaptive mesh refinement during ion implantation, diffusion, and oxidation. It can perform adaptive mesh unrefinement during diffusion and oxidation.

Refinement

Adaptive mesh refinement in TSUPREM-4 consists of splitting an edge of a triangle by adding a node at its midpoint if the estimated error at the midpoint is greater than the user-specified limit:

$$\varepsilon > \mathbf{REL.ADAP} \times \mathbf{ERR.FAC} \times C + \mathbf{ABS.ADAP} \qquad \text{Equation 3-14}$$

where **REL.ADAP**, **ABS.ADAP**, and **ERR.FAC** are parameters on the **METHOD** statement and C is the value of a solution variable.

For adaptive gridding during ion implantation (with the analytical models):

- C is the concentration of the impurity being implanted.
- ε is calculated directly from the ion implantation model.

No mesh refinement takes place during implants with the Monte Carlo (MC) model.

For adaptive gridding during diffusion:

- The calculation is done for each mobile impurity species, using values of ε based on estimates of the curvature of the impurity profile.

For adaptive gridding during oxidation:

- The calculation is done for the oxidizing species, using values of ε based on estimates of the curvature of the profile of the oxidant.

Separate values of **REL.ADAP** and **ABS.ADAP** are specified for each solution value in each material. **ERR.FAC** is a single value that scales the relative errors specified by **REL.ADAP**. The minimum grid spacing produced during adaptive refinement is specified for each solution value and material by the **MIN.SPAC** parameter on the **METHOD** statement.

Unrefinement

Adaptive mesh unrefinement in TSUPREM-4 consists of removing mesh points that are not required to maintain the accuracy of the solution. The criterion for point removal is:

$$\varepsilon < \mathbf{UNREFINE} \times (\mathbf{REL.ADAP} \times \mathbf{ERR.FAC} \times C + \mathbf{ABS.ADAP})$$

Equation 3-15

where **UNREFINE** is a parameter on the **METHOD** statement. Unrefinement is done twice in each diffusion step, about one quarter and three quarters of the way through the step. The maximum grid spacing produced by unrefinement is specified for each material by the **MIN.SPAC** parameter on the **METHOD** statement.

You can also request mesh unrefinement with the **UNREFINE** parameter on the **STRUCTURE** statement. The **REPEAT** parameter specifies the maximum number of unrefinement passes. Normally unrefinement does not remove points in the undoped (or background doped) substrate, but if **ROWS** or **COLUMNS** is specified on the **STRUCTURE** statement, these points can be removed. This can be useful in preparing a structure for device simulation. The unrefinement can be limited to a portion of the structure with the **X.MIN**, **X.MAX**, **Y.MIN**, and **Y.MAX** parameters.

Adaptive Gridding for Damage

The interstitial distribution is necessarily abrupt near amorphous and crystalline boundary after regrowth because the boundary location is determined by comparing implant damage concentration to the **`MAX.DAM`** parameter value, and then the interstitial concentration is reset to the equilibrium value in the regrowth region. Thus, the interstitial dose Transient Enhanced Diffusion (TED) is sensitive to the mesh density near the boundary region. The adaptive gridding method based on the phase transition between amorphous and crystalline regions makes denser mesh near the boundary so that it increases simulation accuracy. Unlike the adaptive gridding of other solutions which uses the absolute and relative errors based on the solution values, the adaptive gridding for damage solution is performed on the transformed profile of damage:

$$s = 1 + \tanh[\mathbf{DAM.GRAD} \times \ln(D_0 / \mathbf{MAX.DAMA})] \qquad \text{Equation 3-16}$$

Usage

The **`IMP.ADAP`** parameter on the **`METHOD`** statement enables or disables adaptive gridding during ion implantation. The **`DIF.ADAP`** parameter enables or disables adaptive gridding (both refinement and unrefinement) during diffusion. The **`OX.ADAP`** parameter enables or disables adaptive gridding in oxide based on oxidant concentration. **`DAM.ADAP`** parameter enables or disables adaptive gridding in the region near amorphous and crystalline boundary. Refinement is disabled for impurity/material combinations for which **`REL.ADAP`** is zero. No unrefinement will occur in materials for which **`MIN.SPAC`** is zero, and no unrefinement will occur if **`UNREFINE`** is zero.

Note:

> ***IMP.ADAP and DAM.ADAP are applicable for analytic implants for regrid prior to a Monte Carlo implant.***

Adaptive gridding reduces the impact of the initial grid on the speed and accuracy of the simulation but does not eliminate it entirely. Meshes that rely heavily on adaptive gridding tend to be less regular than manually specified meshes of similar density. Results using these irregular meshes may appear less accurate because of the irregular variation of the numerical error from point to point. Regular, manually generated meshes may be preferable when precise comparison of similar solutions is required.

Even with adaptive gridding, you must verify that the following requirements are met to ensure accurate simulations:

- The initial grid spacing in layers that receive implants must be smaller than the standard deviation of the implanted profile.
- When an epitaxial layer is grown on a substrate with a high surface impurity concentration, the first grid space in the epi layer must be small compared to the expected depth of out-diffusion into the epi layer.

There may be cases where you want to modify or disable adaptive gridding:

- Adaptive refinement during diffusion following a MC implant may cause a large number of nodes to be added in response to statistical variations in the implanted profile. It is often a good idea to disable **DIF.ADAP** during the first few minutes of diffusion following a MC implant.

- When oxidizing through a poly layer you may want to disable **OX.ADAPT** between the time when the oxidation first breaks through the poly and the time when the poly layer has been consumed across its entire width. This avoids adding unnecessary extra nodes in the pad oxide under the poly.
- In situations where less accuracy is needed, you may want to set **ERR.FAC** to a value greater than 1. Similarly, where more accuracy is needed you may want to use a value less than 1.
- For large structures that undergo long high-temperature diffusion cycles, you may want to be more aggressive in removing unneeded mesh points. It is usually safe to use values of **UNREFINE** as high as 1.0.
- There may be cases where unrefinement is removing mesh points you will soon be needing. (For example, nodes near the surface may be removed prior to an implant that needs them.) In this case you can disable unrefinement by setting **UNREFINE** to 0 until the points are used.
- You may want to reduce the value of **MAX.SPAC** in silicon for small geometry devices and increase the value for power devices.

1-D Simulation of Simple Structures

In many simulations, the geometry and doping vary in the *y* direction only. Such structures are represented internally as 1-D structures, with considerable savings in simulation time and memory requirements.

1-D structures are automatically converted to 2-D structures whenever an etch or expose step destroys the uniformity of the structure in the *x* direction. 1-D structures are also converted to two dimensions for display (using the **PLOT.2D** statement) or saving to a file. Such conversions are temporary, and 1-D simulation resumes after plotting or saving.

All saved structures are stored as 2-D structures; structures that are uniform in the *x* direction are converted to 1-D structures when they are read from a file. Full 2-D simulation of 1-D structures can be forced by specifying **^FAST** on the **MESH** statement.

Initial Impurity Concentration

The initial impurity concentration in a structure can be specified directly or by specifying the resistivity of the material. In either case, one of two styles can be used:

- In the old-style specification, **ANTIMONY**, **ARSENIC**, **BORON**, and **PHOSPHOR** parameters are used to set the impurity concentration or resistivity of antimony, arsenic, boron, and phosphorus, respectively; **CONCENTR** or **RESISTIV** is used to determine whether the concentration or resistivity is specified.

- In the new-style specification, impurity is specified by name with the **IMPURITY** parameter and the concentration or resistivity is specified with the **I.CONC** or **I.RESIST** parameters.

The old-style and new-style specifications can be mixed when specifying impurity concentrations directly, i.e., the **IMPURITY** and **I.CONC** parameters can be used on the same statement with the **ANTIMONY**, **ARSENIC**, **BORON**, and **PHOSPHOR** parameters. Only one impurity can be specified when **RESISTIV** or **I.RESIST** is specified.

When the resistivity is specified, the concentration is given by:

$$C = \begin{cases} \dfrac{1}{q\mu_n\rho} & \text{donor impurity} \\ \dfrac{1}{q\mu_p\rho} & \text{acceptor impurity} \end{cases} \qquad \text{Equation 3-17}$$

where μ_n and μ_p are the electron and hole mobilities, respectively, and ρ is the resistivity. The carrier mobilities are contained in three internal tables—one for hole mobility in p-type silicon, one for electron mobility in arsenic- and antimony-doped silicon, and one for electron mobility in silicon doped with other donor impurities. The tables are from Masetti, et al. Reference [2].

Note:

The mobility tables used for determining the initial concentration are not the same as those used for calculating sheet resistance with the **ELECTRICAL** ***statement. Thus, the extracted sheet resistance is not identical to the specified resistivity of the starting material.***

The same calculation is used for all materials, even though it is only meaningful for silicon and polysilicon; a warning is printed if the resistivity is specified for other materials.

Specification of the initial impurity concentration is the same for the **INITIALIZE**, **DEPOSITION**, and **EPITAXY** statements.

Diffusion

The most fundamental process simulated by TSUPREM-4 is diffusion, which, when performed in an oxidizing ambient or in the presence of an appropriate metal, also produces oxidation or silicidation. In addition to simulating the diffusion of the dopant atoms, TSUPREM-4 models the diffusion of point defects (i.e., self-interstitials and vacancies) and, in some cases, an oxidizing species (assumed to be O_2 or H_2O). For silicidation, diffusion of metal and/or silicon atoms through silicide is modeled. The models used by the **DIFFUSION** statement are described in the following sections.

DIFFUSION Statement

The **DIFFUSION** statement (see Chapter 4, DIFFUSION, p. 4-131) is used to model the diffusion of impurities under oxidizing and nonoxidizing conditions. The duration of the diffusion step (in minutes) is specified with the **TIME** parameter.

Temperature

The initial temperature of the step (in $^\circ$C) is given by the **TEMPERAT** parameter. Linear variation of the temperature over the step can be specified with the **T.RATE** or **T.FINAL** parameters. If **T.RATE** is specified, the temperature varies as:

$$T_c = \mathtt{TEMPERAT} + \mathtt{T.RATE} \times t \tag{Equation 3-18}$$

where t is the time since the start of the step and T_c is the diffusion temperature (in $^\circ$C) at time t. If **T.FINAL** is specified, the temperature varies as:

$$T_c = \mathtt{TEMPERAT} + \frac{(\mathtt{T.FINAL} - \mathtt{TEMPERAT})}{\mathtt{TIME}} t \tag{Equation 3-19}$$

If neither **T.RATE** nor **T.FINAL** is specified, the temperature is constant. The physical coefficients that depend on temperature are presumed to be valid in the range 800 to 1250×° C, but temperatures outside this range are allowed.

Ambient Gas Pressure

The pressure of the ambient gas during the step can vary linearly with time, and is specified with the **PRESSURE** parameter and either the **P.RATE** or **P.FINAL** parameter. The **PRESSURE** parameter specifies the initial pressure. If neither **P.RATE** nor **P.FINAL** is specified, the pressure is constant. If **P.RATE** is specified, the pressure varies as:

$$P = \mathtt{PRESSURE} + \mathtt{P.RATE} \times t \tag{Equation 3-20}$$

where t is the time since the start of the step and P is the pressure at time t. If **P.FINAL** is specified, the pressure varies as:

$$P = \mathtt{PRESSURE} + \frac{(\mathtt{P.FINAL} - \mathtt{PRESSURE})}{\mathtt{TIME}} t \tag{Equation 3-21}$$

These values must be chosen to yield positive, nonzero pressures throughout the step.

Ambient Gas Characteristics

The characteristics of the ambient gas can be specified in one of two ways:

- Specify a previously defined ambient with one of the parameters **DRYO2**, **WETO2**, **STEAM**, **INERT**, **AMB.1**, **AMB.2**, **AMB.3**, **AMB.4**, or **AMB.5**.
- Define the ambient gas directly by specifying the flows of the oxidizing and nonoxidizing species with the parameters **F.O2**, **F.H2O**, **F.H2**, **F.N2**, and **F.HCL** or **HCL**.

Ambients and Oxidation of Materials

The characteristics of defined ambients and the physical coefficients describing the oxidation of materials are specified with the **`AMBIENT`** statement (see Chapter 4, AMBIENT, p. 4-233). The flows of the oxidizing and nonoxidizing species associated with the ambient are specified with the **`F.O2`**, **`F.H2O`**, **`F.H2`**, **`F.N2`**, and **`F.HCL`** parameters. The default gas pressure can be specified for the ambient with the **`PRESSURE`** parameter, and the default chlorine percentage can be specified with **`HCL`**.

If flows of both O_2 and H_2 are present in the ambient, these gases are assumed to undergo a complete pyrogenic reaction to form H_2O as given by:

$$H_2 + \frac{1}{2}O_2 \rightarrow H_2O \qquad \text{Equation 3-22}$$

The final flows of O_2, H_2, and H_2O after the pyrogenic reaction are given by:

$$F_{\texttt{O2}} = \texttt{F.O2} - \min\left(\texttt{F.O2}, \frac{\texttt{F.H2}}{2}\right) \qquad \text{Equation 3-23}$$

$$F_{\texttt{H2}} = \texttt{F.H2} - \min(2 \times \texttt{F.O2}, \texttt{F.H2}) \qquad \text{Equation 3-24}$$

$$F_{\texttt{H2O}} = \texttt{F.H2O} + \min(2 \times \texttt{F.O2}, \texttt{F.H2}) \qquad \text{Equation 3-25}$$

The partial pressures of the oxidizing species are given by:

$$P_{\texttt{O2}} = \frac{F_{\texttt{O2}}}{F_{\texttt{O2}} + F_{\texttt{H2O}} + F_{\texttt{H2}} + \texttt{F.N2} + \texttt{F.HCL}} P \qquad \text{Equation 3-26}$$

$$H_{20} = \frac{F_{\texttt{H2O}}}{F_{\texttt{O2}} + F_{\texttt{H2O}} + F_{\texttt{H2}} + \texttt{F.N2} + \texttt{F.HCL}} P \qquad \text{Equation 3-27}$$

If the resulting ambient contains both O_2 and H_2, the oxidation rate is based on the partial pressure of H_2O.

Default Ambients

The following default ambients are defined:

- **`DRYO2`**: The dry oxygen ambient contains 100% O_2.
- **`WETO2`**: The wet oxygen ambient contains 92% H_2O and 8% N_2. This reflects evidence that wet O_2 (oxygen bubbled through H_2O at 95° C) is equivalent to pyrogenic steam with O_2 and H_2 flow rates of 1.175 and 2.0 liters/minute, respectively Reference [3]. (The actual ambient contains 8% O_2, but because TSUPREM-4 cannot model simultaneous oxidation by H_2O and O_2, nitrogen is substituted for the oxygen in the simulation.)
- **`STEAM`**: The steam ambient contains 100% H_2O. This represents formation of H_2O by a complete pyrogenic reaction of O_2 and H_2, without excess O_2 or H_2.
- **`INERT`**: The inert ambient contains 100% N_2 (or other inert gasses).

The default **`PRESSURE`** for each of these ambients is 1 atmosphere.

Chlorine

The inclusion of chlorine in the ambient is specified with either the **HCL** or **F.HCL** parameter. These parameters are related by:

$$\mathtt{HCL} = 100\frac{\mathtt{F.HCL}}{F_{\mathtt{O2}} + F_{\mathtt{H2O}} + F_{\mathtt{H2}} + \mathtt{F.N2} + \mathtt{F.HCL}}$$

Equation 3-28

where $F_{\mathtt{O2}}$, $F_{\mathtt{H2}}$, and $F_{\mathtt{H2O}}$ are the final flows of O_2, H_2, and H_2O, respectively, after the pyrogenic reaction of O_2 and H_2 to form H_2O. The chlorine percentage is defined as 100 times the mole fraction of atomic chlorine relative to the total ambient gas.

Example

For example, in dry oxygen 1% chlorine represents the presence of one chlorine atom for every 99 O_2 molecules. Either **HCL** or **F.HCL** can be used when defining an ambient on the **AMBIENT** or **DIFFUSION** statement. If **HCL** is specified along with a predefined ambient on the **DIFFUSION** statement, the specified chlorine percentage is used instead of the percentage or flow rate defined for the ambient.

Coefficient Tables

The effects of chlorine in the ambient gas on the oxidation rate of silicon are represented by tables of coefficients that modify the linear oxidation rate. The tables are specified with the **AMBIENT** statement. (See Chapter 4, AMBIENT, p. 4-233.) The tables are 2D with the rows representing chlorine percentages and the columns representing diffusion temperatures. Linear interpolation is used to obtain values for temperatures or percentages between the values in the table. For temperatures or percentages outside the range of values present in the tables, the values in the first or last rows or columns, as appropriate, are used. (SeeAnalytical Oxidation Models, p. 3-87 for the use of the coefficients in these tables.)

Chemical Predeposition

A chemical predeposition step can be modeled with the **DIFFUSION** statement by specifying the concentrations of one or more impurities in the ambient gas. The impurity concentrations are specified explicitly with the parameters **ANTIMONY**, **ARSENIC**, **BORON**, and **PHOSPHOR** or with the **IMPURITY** and **I.CONC** parameters. Impurities can also be included in a thermal oxidation step, but the resulting oxide is always modeled as a high-quality thermal oxide; the effects of high impurity concentrations on the oxide characteristics are not simulated.

Solution of Diffusion Equations

The diffusion equations are nonlinear because of the dependence of the diffusion coefficients and electric field on the impurity, point defect, and carrier concentrations. These dependencies also couple the equations for multiple impurities and point defects. An accurate solution to this coupled nonlinear system is obtained by dividing the total diffusion time specified by the **TIME** parameter into a series of smaller time steps represented by Δt. The initial time step is set by the **INIT.TIM** parameter on the **METHOD** statement (see Chapter 4, METHOD, p. 4-207). For each time step thereafter, the value of Δt is made as large as possible while holding the estimated error in the solution to acceptably small values. The time step Δt is chosen to satisfy Reference [4].

$$\sqrt{\frac{1}{n}\sum_{i=1}^{n}\left(\sum_{j=1}^{m}\left|\frac{\Delta C_{ij}}{\mathbf{REL.ERR}_j \cdot C_{ij} + \mathbf{ABS.ERR}_j}\right|\right)^2} \leq 1 \qquad \text{Equation 3-29}$$

where:

- n is the number of nodes in the structure.
- m is the number of diffusing species (impurities and point defects) at each node.
- C_{ij} is the concentration of species j at node i.
- ΔC_{ij} is the estimated error in C_{ij}.
- `REL.ERR` and `ABS.ERR` are the relative and absolute error for each species, specified with the `METHOD` statement.

The time step may be also be reduced during oxidation to avoid numerical difficulties or to prevent mesh tangling.

Diffusion of Impurities

This section describes the equations that model the diffusion of dopant atoms in the device structure. Diffusion of dopants is simulated in all materials.

The diffusion equation solved for each impurity present in the structure is:

$$\frac{\partial C}{\partial t} = -\overrightarrow{\nabla} \cdot \left(\overrightarrow{J_m} + \overrightarrow{J_n}\right) \qquad \text{Equation 3-30}$$

where:

- C is the chemical impurity concentration.
- $\overrightarrow{\nabla} \cdot$ is the divergence operator.

Impurity Fluxes

The impurity fluxes in the interior of a material region are given by:

$$\overrightarrow{J_m} = -D_m\left[\overrightarrow{\nabla}\left(C_m\frac{M}{M'}\right) - z_s\left(C_m\frac{M}{M'}\right)\frac{q\overrightarrow{E}}{kT} - \left(C_m\frac{M}{M'}\right)\left(\frac{\mathbf{VALPHAM}}{kT}\overrightarrow{\nabla}p\right)\right]$$

Equation 3-31

$$\overrightarrow{J_n} = -D_n\left[\overrightarrow{\nabla}\left(C_m\frac{N}{N'}\right) - z_s\left(C_m\frac{N}{N'}\right)\frac{q\overrightarrow{E}}{kT} - \left(C_m\frac{N}{N'}\right)\left(\frac{\mathbf{VALPHAN}}{kT}\overrightarrow{\nabla}p\right)\right]$$

Equation 3-32

where:

- $\vec{J}_m$ and D_m are the flux and diffusivity of impurities diffusing with (or as) interstitials.
- $\vec{J}_n$ and D_n are the flux and diffusivity of impurities diffusing with vacancies.
- z_s is the charge of the ionized impurity (+1 for donors and −1 for acceptors).
- q is the electronic charge.
- k is Boltzmann's constant.
- T is the absolute temperature.
- C_m is the mobile impurity concentration.
- $\vec{\nabla}$ is the gradient operator.
- M/M' and N/N' model the enhancement (or retardation) of diffusion due to nonequilibrium point defect concentrations.
- p is the hydrostatic pressure. **VALPHAM** and **VALPHAN** are the parameters associated with the pressure-dependent dopant-interstitial and dopant-vacancy pair formation energies respectively. Only applicable when the boolean parameter **ST.ALPHA** in the **METHOD** statement is turned on. See Pressure Dependence of Formation Energy, p. 3-23

Mobile Impurities and Ion Pairing

TSUPREM-4 includes a model for the pairing of positively and negatively charged dopant ions, see References [5], [6], and [7]. This model reduces the concentration of mobile dopant atoms C_m according to:

$$C_m = F_{pd}\, C_a \quad \text{(donors)} \qquad \text{Equation 3-33}$$

$$C_m = F_{pa}\, C_a \quad \text{(acceptors)} \qquad \text{Equation 3-34}$$

where C_a is the electrically active dopant concentration (Activation of Impurities, p. 3-34) and F_{pd} and F_{pa} are the ion pairing factors for donors and acceptors, respectively. Ion pairing reduces the diffusivity of dopants where the concentration of dopants of the opposite type is large.

The ion pairing model assumes that positively charged donors can bind with negatively charged acceptors to form neutral pairs:

$$D^{+} + A^{-} \rightleftarrows P \qquad \text{Equation 3-35}$$

The forward reaction rate is proportional to the number of unpaired donor and acceptor ions, while the reverse reaction rate is proportional to the number of pairs. In equilibrium:

$$(N_d - N_p)(N_a - N_p) = \Omega N_p \qquad \text{Equation 3-36}$$

Where N_d and N_a are the total concentrations of electrically active donors and acceptors, respectively, N_p is the concentration of ion pairs, and Ω is a proportionality factor. Equation 3-36 can be solved for the number of ion pairs, giving:

$$N_p = \frac{1}{2}\left[(N_d + N_a + \Omega) - \sqrt{(N_d + N_a + \Omega)^2 - 4N_d N_a}\right] \qquad \text{Equation 3-37}$$

The pairing factors are then given by:

$$F_{pd} = \left(1 - \frac{N_p}{N_d}\right) \qquad \text{(donors)} \qquad \text{Equation 3-38}$$

$$F_{pa} = \left(1 - \frac{N_p}{N_a}\right) \qquad \text{(acceptors)} \qquad \text{Equation 3-39}$$

The parameter Ω is given by:

$$\Omega = \mathbf{IP.OMEGA}\; n_i \qquad \text{Equation 3-40}$$

where **IP.OMEGA** is a parameter on the **MATERIAL** statement; the default value for silicon and polysilicon is 6.0 Reference [6]. The ion pairing model is enabled or disabled for each material by the **ION.PAIR** parameter on the **MATERIAL** statement; by default, it is enabled for silicon and polysilicon, but disabled for all other materials, including new user-defined materials.

The ion pairing model is significant because it allows the dependence of the impurity diffusivity to be modeled in both n-type and p-type materials. In particular, it may reduce the effective diffusivity of boron in n-type materials without affecting its diffusivity at high p-type concentrations.

Electric Field

In insulator and conductor materials, the electric field vector $\vec{E}$ is zero. In semiconductor materials, the electric field is given by:

$$\begin{aligned}\vec{E} &= -\vec{\nabla}\psi \\ &= -\frac{kT}{q}\frac{\vec{\nabla}\eta}{\eta}\end{aligned} \qquad \text{Equation 3-41}$$

where η is the ratio of the electron concentration to the effective intrinsic carrier concentration, i.e. n/n_{ie}. The distribution of the charge carriers can be extracted either by assuming local charge neutrality or by numerically solving the Poisson's equation. By default, local charge neutrality is assumed so that the electron concentration is written as:

$$n = \frac{N_d - N_a}{2} + \sqrt{\left(\frac{N_d - N_a}{2}\right)^2 + n_{ie}^2}$$

Equation 3-42

where:

- N_d and N_a are the sums of the electrically active donor and acceptor impurity concentrations, respectively.
- n_{ie} is the effective intrinsic carrier concentration given by:

$$n_{ie} = n_i \exp\left(\frac{-\texttt{EG.ADD}}{2kT}\right)$$

Equation 3-43

$$n_i = \texttt{NI.0} \cdot \exp\left(\frac{-\texttt{NI.E}}{kT}\right) T^{\texttt{NI.F}}$$

Equation 3-44

where **`EG.ADD`**, **`NI.0`**, **`NI.E`**, and **`NI.F`** are specified in the **`MATERIAL`** statement. **`EG.ADD`** can be specified with a number or an expression.

Presence of the electric field term produces couplings among the diffusion equations for the different impurities.

Note:

Turning on the `POISSON` parameter in the `METHOD` statement specifies that the Poisson's equation is to be solved instead of assuming local charge neutrality. The Poisson's equation is solved only for silicon or polysilicon materials. However, in polycrystalline material, the electric field on grain boundary is always calculated by local charge neutrality regardless of the specification of solving the Poisson equation.

The electrically active and mobile impurity concentrations (C_a and C_m, respectively) are assumed to be the same. The model for calculating these values is described in Activation of Impurities, p. 3-34.

Diffusivities

It is assumed that impurities diffuse in semiconductor materials as dopant-defect pairs. The diffusion coefficients D_m and D_n are sums of the effective diffusivities of impurities due to pairing with defects in various charge states:

$$D_m = F_{stn} \cdot F_{stsm} \cdot \mathtt{FGB} \cdot \mathtt{DI.FAC} \cdot \begin{bmatrix} \mathtt{DI.F11} & 0 \\ 0 & \mathtt{DI.F22} \end{bmatrix} \cdot \sum_{k=-6}^{6} \left(D_{mk} \eta^{-k}\right)$$

$$D_n = F_{stn} \cdot F_{stsn} \cdot \mathtt{FGB} \cdot \mathtt{DV.FAC} \cdot \begin{bmatrix} \mathtt{DV.F11} & 0 \\ 0 & \mathtt{DV.F22} \end{bmatrix} \cdot \sum_{k=-6}^{6} \left(D_{nk} \eta^{-k}\right)$$

Equation 3-45

The components of diffusivity are given by:

$$D_{mk} = F_{\alpha m} \cdot D_{0,\ mk} \qquad D_{nk} = F_{\alpha n} \cdot D_{0,\ nk}$$

Equation 3-46

$$D_{0,\ m0} = \mathtt{DIX.0} \cdot \exp\left(\frac{-\mathtt{DIX.E}}{kT}\right) \qquad D_{0,\ n0} = \mathtt{DVX.0} \cdot \exp\left(\frac{-\mathtt{DVX.E}}{kT}\right)$$

$$D_{0,\ m1} = \mathtt{DIP.0} \cdot \exp\left(\frac{-\mathtt{DIP.E}}{kT}\right) \qquad D_{0,\ n1} = \mathtt{DVP.0} \cdot \exp\left(\frac{-\mathtt{DVP.E}}{kT}\right)$$

$$D_{0,\ m(-1)} = \mathtt{DIM.0} \cdot \exp\left(\frac{-\mathtt{DIM.E}}{kT}\right) \qquad D_{0,\ n(-1)} = \mathtt{DVM.0} \cdot \exp\left(\frac{-\mathtt{DVM.E}}{kT}\right)$$

$$D_{0,\ m(-2)} = \mathtt{DIMM.0} \cdot \exp\left(\frac{-\mathtt{DIMM.E}}{kT}\right) \qquad D_{0,\ n(-2)} = \mathtt{DVMM.0} \cdot \exp\left(\frac{-\mathtt{DVMM.E}}{kT}\right)$$

Equation 3-47

where the parameters **DIX.0**, **DIX.E**, **DIP.0**, **DIP.E**, **DIM.0**, **DIM.E**, **DIMM.0**, **DIMM.E**, **DVX.0**, **DVX.E**, **DVP.0**, **DVP.E**, **DVM.0**, **DVM.E**, **DVMM.0**, and **DVMM.E** are specified in the **ANTIMONY**, **ARSENIC**, **BORON**, and **PHOSPHORUS** statements (see Chapter 4, PHOSPHORUS, p. 4-358).

Diffusivities for arbitrary charge states are given by:

$$D_{0,\ mk} = \mathtt{DIC.0} \cdot \exp\left(\frac{-\mathtt{DIC.E}}{kT}\right) \qquad D_{0,\ nk} = \mathtt{DVC.0} \cdot \exp\left(\frac{-\mathtt{DVC.E}}{kT}\right)$$

Equation 3-48

where the parameters **DIC.0**, **DIC.E**, **DVC.0**, and **DVC.E** for the charge state $k = \mathtt{C.STATE}$ are specified on the **IMPURITY** statement (see Chapter 4, IMPURITY, p. 4-273).

Polysilicon Enhancement

The factor **FGB** is applied only for materials for which the polycrystalline model is disabled, and only if the value specified is nonzero. It compensates for the omission of the grain-boundary diffusion flux in polycrystalline materials when the polycrystalline diffusion models are not used.

Strain Enhancement

The Strain Enhancement model assumes that the activation energy depends linearly on a x-directional strain E_{xx}. The factor F_{stn} is applied only for diffusion in silicon materials for which the stress history is modeled, and the **`STN.DIFF`** model is enabled. The strain factor **`STN.FAC`** has dimension of energy per unit strain and is specified on the **`IMPURITY`** statement. The parameters for setting stress history and strain dependent diffusion are specified on the **`METHOD`** statement.

$$F_{stn} = \exp\left(\frac{-E_{xx} \cdot \ \mathtt{STN.FAC}}{kT}\right)$$

Equation 3-49

Pressure Dependence of Migration Energy

The pressure dependence of a dopant-defect pair or a substitutional atom is modeled by assuming that the migration energy depends linearly on the pressure p and a activation volume, **`VMIPAIR`** for a dopant-interstitial pair and **`VMVPAIR`** for a dopant-vacancy pair. The factors F_{stsm} or F_{stsn} are applied only for diffusion in silicon materials for which the stress history is modeled, and the **`ST.DIFF`** model is enabled. The activating volumes, **`VMIPAIR`** and **`VMVPAIR`** have a dimension of A^3 and are specified on the **`IMPURITY`** statement. The parameters for setting stress history and stress dependent diffusion are specified on the **`METHOD`** statement.

$$F_{stsm} = \exp\left(\frac{-p \cdot \ \mathtt{VMIPAIR}}{kT}\right), \qquad F_{stsn} = \exp\left(\frac{-p \cdot \ \mathtt{VMVPAIR}}{kT}\right)$$

Equation 3-50

Note:

> ***If you define VD, the value of VD is assigned to both VMIPAIR and VMVPAIR.***

User-Defined Isotropic Multiplication

The diffusivities can be multiplied by the parameters **`DI.FAC`** and **`DV.FAC`** in the **`IMPURITY`** statement. Parameters values can be numbers or formulas. This provides the method for users to define more complicated diffusivity models.

User-Defined Anisotropic Multiplication

Dopant diffusion in presence of anisotropic stress is not isotropic anymore. To provide the modeling capability for this type of anisotropic diffusion, the user-specified tensor expression can be taken. It describes the anisotropic diffusion path, which is given by the parameters **`DI.F11`**, **`DI.F22`**, **`DV.F11`**, and **`DV.F22`** in the **`IMPURITY`** statement. Each parameter can be given by any formula expression.

Pressure Dependence of Formation Energy

The pressure dependence of the formation energy of a dopant-defect pair and a substitutional atom is modeled by assuming that the formation energy depends linearly on the pressure p and a activation volume, **`VALPHAM`** for a dopant-interstitial pair and **`VALPHAN`** for a dopant-vacancy pair. The factors $F_{\alpha m}$ or F_{stsn} are applied only for diffusion in silicon materials for which the stress history is modeled, and the **`ST.DIFF`** model is enabled. The activating volumes, **`VALPHAM`** and **`VALPHAN`** have a dimension of A^3 and are specified on the **`IMPURITY`** statement. The parameters for setting stress history and stress dependent diffusion are specified on the **METHOD** statement.

$$F_{\alpha m} = \exp\left(\frac{-p \cdot \ \mathtt{VALPHAM}}{kT}\right), \qquad F_{\alpha n} = \exp\left(\frac{-p \cdot \ \mathtt{VALPHAN}}{kT}\right)$$

Equation 3-51

Point Defect Enhancement

The definition of the point defect enhancement factors $M/\ M'$ and $N/\ N'$ depends on the **NSTREAMS**, **PAIR.SAT**, and **PAIR.REC** parameters specified on the **METHOD** statement. For **NSTREAMS** =1 (e.g., if **PD.FERMI** is set on the **METHOD** statement) and in materials other than silicon, the point defect concentrations are assumed to be at their thermal equilibrium values so that:

PD.FERMI Model

$$\frac{M}{M'} = \frac{N}{N'} = 1$$

Equation 3-52

and Equations 3-30, 3-31, and 3-32 reduce to:

$$\frac{\partial C}{\partial t} = -\vec{\nabla} \cdot \left[-(D_m + D_n)\left(\vec{\nabla} C_m - z_s C_m \frac{q\vec{E}}{kT}\right) + (D_m \cdot \ \mathtt{VALPHAM} + D_n \cdot \ \mathtt{VALPHAN})\left(C_m \frac{\vec{\nabla} p}{kT}\right) \right]$$

Equation 3-53

PD.TRANS and PD.FULL Models

For **NSTREAMS** =3 (e.g., if **PD.TRANS** or **PD.FULL** is set on the **METHOD** statement), use the full equations for the enhancement factors:

$$\frac{M}{M'} = \frac{\dfrac{I}{I^*}\left(\dfrac{K_m + K_{mv}\dfrac{I^*}{I}}{K_m + K_{mv}\dfrac{V}{V^*}}\right)}{1 + \alpha_m \dfrac{I}{I^*}\left(\dfrac{K_m + K_{mv}\dfrac{I^*}{I}}{K_m + K_{mv}\dfrac{V}{V^*}}\right) + \alpha_n \dfrac{V}{V^*}\left(\dfrac{K_n + K_{ni}\dfrac{V^*}{V}}{K_n + K_{ni}\dfrac{I}{I^*}}\right)}$$

Equation 3-54

$$\frac{N}{N'} = \frac{\dfrac{V}{V^*}\left(\dfrac{K_n + K_{ni}\dfrac{V^*}{V}}{K_n + K_{ni}\dfrac{I}{I^*}}\right)}{1 + \alpha_m\dfrac{I}{I^*}\left(\dfrac{K_m + K_{mv}\dfrac{I^*}{I}}{K_m + K_{mv}\dfrac{V}{V^*}}\right) + \alpha_n\dfrac{V}{V^*}\left(\dfrac{K_n + K_{ni}\dfrac{V^*}{V}}{K_n + K_{ni}\dfrac{I}{I^*}}\right)}$$

Equation 3-55

α_m and α_n are the ratios of the dopant-defect pair concentrations to the substitutional dopant concentration under equilibrium conditions. They are used to define M' and N' :

$$M' \equiv \alpha_m C_m$$

Equation 3-56

$$N' \equiv \alpha_n C_m$$

Equation 3-57

The values of α_m and α_n are calculated from:

$$\alpha_m = \sum_k \frac{D_{mk}}{D_{Mk}}\eta^{-k} \quad \text{and} \quad \alpha_n = \sum_k \frac{D_{nk}}{D_{Nk}}\eta^{-k}$$

Equation 3-58

where:

$$D_{Mk} = \mathtt{DIPAIR.0} \cdot \exp\left(-\frac{\mathtt{DIPAIR.E}}{kT}\right)$$

Equation 3-59

$$D_{Nk} = \mathtt{DVPAIR.0} \cdot \exp\left(-\frac{\mathtt{DVPAIR.E}}{kT}\right)$$

Equation 3-60

and **DIPAIR.0**, **DIPAIR.E**, **DVPAIR.0**, and **DVPAIR.E** are parameters on the **IMPURITY**, **ANTIMONY**, **ARSENIC**, **BORON**, and **PHOSPHORUS** statements. Physically, D_{Mk} and D_{Nk} are the diffusivities of the $(z+k)$-charged dopant-defect pairs; they are expected to have values comparable to the diffusivities of interstitials and vacancies, respectively. The charge state $(z+k)$'s of the dopant-interstitial pairs and the dopant-vacancy pairs are specified with the parameters **C.IPAIR** and **C.VPAIR**, respectively. Instead, turing on the boolean parameters **C.IP.ALL** and **C.VP.ALL** assumes the diffusivities of dopant-interstitial pairs and dopant-vacancy pairs respectively to be independent of charge state.

Paired Fractions of Dopant Atoms

The values of α_m and α_n vary with the Fermi level, but do not otherwise depend on the dopant or defect concentrations. Physically, they represent the fractions of dopant atoms that are coupled with interstitials and vacancies. The α_m and α_n terms prevent the pair concentrations from exceeding the total dopant concentrations when the concentrations of point defects are very high Reference [8]. These terms are calculated only if **PAIR.SAT** is true (set on the **METHOD** statement, directly or by specifying **PD.FULL**); otherwise, they are set to 0.

5-Stream Diffusion Model

Setting the parameter **NSTREAMS** to 5 or turning on the parameter **PD.5STR** specifies to solve the transient equations of both dopant-interstitial and dopant-vacancy pairs:

$$\frac{\partial M}{\partial t} = -\vec{\nabla}\cdot\ J_m + (G_m - R_m) - (G_{mv} - R_{mv}) - R_{DDCM} \qquad \text{Equation 3-61}$$

$$\frac{\partial N}{\partial t} = -\vec{\nabla}\cdot\ J_n + (G_n - R_n) - (G_{ni} - R_{ni}) - R_{DDCN} \qquad \text{Equation 3-62}$$

where $G_m - R_m$ and $G_n - R_n$ are the reaction rates for generation of dopant-interstitial and dopant-vacancy pairs, respectively;

$$G_m - R_m = K_m\left(S\left(\frac{I}{I^*}\right) - \frac{M}{\alpha_m}\right) \qquad \text{Equation 3-63}$$

$$G_n - R_n = K_n\left(S\left(\frac{V}{V^*}\right) - \frac{N}{\alpha_n}\right) \qquad \text{Equation 3-64}$$

$G_{mv} - R_{mv}$ and $G_{ni} - R_{ni}$ are the dopant-assisted recombination rates for interstitials and vacancies;

$$G_{mv} - R_{mv} = K_{mv}\left(\left(\frac{M}{\alpha_m}\right)\left(\frac{V}{V^*}\right) - S\right) \qquad \text{Equation 3-65}$$

$$G_{ni} - R_{ni} = K_{ni}\left(\left(\frac{N}{\alpha_n}\right)\left(\frac{I}{I^*}\right) - S\right) \qquad \text{Equation 3-66}$$

and R_{DDCM} is the reaction rate of dopant-interstitial pairs for dopant-defect clusters (see the details in the **DDC.FULL** model description):

$$R_{DDCM} = R_{DDC,\ S+M} + R_{DDC,\ I+M} + R_{DDC,\ I(n)+M} + 2R_{DDC,\ M+M} + R_{DDC,\ C+M} \qquad \text{Equation 3-67}$$

The concentration S of substitutional atoms is calculated by;

$$S = C_m - M - N \qquad \text{Equation 3-68}$$

The flux equations of dopant-interstitial pairs and dopant-vacancy pairs are given by:

$$\vec{J}_m = -D_m \cdot \left(\vec{\nabla}\left(\frac{M}{\alpha_m}\right) - z_s\left(\frac{M}{\alpha_m}\right)\frac{q\vec{E}}{kT} - \left(\frac{M}{\alpha_m}\right)\left(\frac{\mathbf{VALPHAM}}{kT}\vec{\nabla} p\right)\right) \quad \text{Equation 3-69}$$

$$\vec{J}_n = -D_n \cdot \left(\vec{\nabla}\left(\frac{N}{\alpha_n}\right) - z_s\left(\frac{N}{\alpha_n}\right)\frac{q\vec{E}}{kT} - \left(\frac{N}{\alpha_n}\right)\left(\frac{\mathbf{VALPHAN}}{kT}\vec{\nabla} p\right)\right) \quad \text{Equation 3-70}$$

Initialization of Pairs

The parameters **`IP.FRACA`**, **`VP.FRACA`** and **`S.FRACA`** in the **`IMPURITY`** statement determines the redistribution of the dopant-defect pairs and the substitutional atoms in the epitaxial regrowth region as follows;

$$M = \left(\frac{\mathbf{IP.FRACA}}{\mathbf{IP.FRACA} + \mathbf{VP.FRACA} + \mathbf{S.FRACA}}\right)C_m \quad \text{Equation 3-71}$$

$$N = \left(\frac{\mathbf{VP.FRACA}}{\mathbf{IP.FRACA} + \mathbf{VP.FRACA} + \mathbf{S.FRACA}}\right)C_m \quad \text{Equation 3-72}$$

Boundary Conditions for Pairs

The flux of dopant-defect pairs at silicon boundary can be calculated by one of the following models;

- Surface recombination flux
- Segregation flux

Turning on **`PAIR.SEG`** in the **`METHOD`** statement applies the segregation flux of pairs instead of the surface recombination flux.

Surface Recombination of Dopant-Defect Pairs

The surface recombination rates of pairs in the 5-stream diffusion model can be defined for the interface between two materials in the **`IMPURITY`** statement. The parameters **`KSIP.0`** and **`KSIP.E`** define the prefactor and the activation energy for the surface recombination of dopant-interstitial pairs, respectively. Likewise, **`KSVP.0`** and **`KSVP.E`** for dopant-vacancy pairs.

$$F_m = k_{ms}(M^* - M) \quad \text{Equation 3-73}$$

$$F_n = k_{ns}(N^* - N) \quad \text{Equation 3-74}$$

where:

$$M^* = \frac{\alpha_m}{1 + \alpha_m + \alpha_n}C \quad \text{Equation 3-75}$$

$$N^* = \frac{\alpha_n}{1+\alpha_m+\alpha_n}C \qquad \text{Equation 3-76}$$

k_{ms} and k_{ns} are defined as follows:

$$k_{ms} = \texttt{KSIP.0}\cdot\ \exp\left(-\frac{\texttt{KSIP.E}}{kT}\right) \qquad \text{Equation 3-77}$$

$$k_{ns} = \texttt{KSVP.0}\cdot\ \exp\left(-\frac{\texttt{KSVP.E}}{kT}\right) \qquad \text{Equation 3-78}$$

Pair Segregation Model

The segregation model for dopant-interstitial pairs is derived with the following reaction:

$$M_{(z+k)} + q_{ij}n \leftrightarrow C_j + I_{(k)} + q_{ji}n \qquad \text{Equation 3-79}$$

where $M_{(z+k)}$ is the concentration of (z+k)-charged dopant-interstitial pairs, C_j is the concentration of dopants at the boundary of a neighbor material adjacent to silicon material, $I_{(k)}$ is the k-charged interstitial at silicon boundary which is generated by the dissociation of the dopant-interstitial pair. Likewise, for dopant-vacancy pairs,

$$N_{(z+k)} + q_{ij}n \leftrightarrow C_j + V_{(k)} + q_{ji}n \qquad \text{Equation 3-80}$$

For the 2-phase segregation model (Segregation of Impurities, p. 3-54):

$$F_m = \texttt{SEG.IFRA}\cdot\ h\left(\frac{I}{I_{i0}^*}\right)^{\texttt{SEG.IPOW}}\left(C_j\eta^{q_{ji}} - \left(\frac{M}{\alpha_m I/\ I^*}\right)\frac{\eta^{q_{ij}}}{m}\right)$$

Equation 3-81

$$F_n = (1-\texttt{SEG.IFRA})h\left(\frac{V}{V_{i0}^*}\right)^{\texttt{SEG.VPOW}}\left(C_j\eta^{q_{ji}} - \left(\frac{N}{\alpha_n V/\ V^*}\right)\frac{\eta^{q_{ij}}}{m}\right)$$

Equation 3-82

For the interface trap model (Interface Trap Model, p. 3-55):

$$F_m = \texttt{SEG.IFRA}\cdot\ h_i\left(\frac{I}{I_{i0}^*}\right)^{\texttt{SEG.IPOW}} \times \left(\kappa_i\sigma - \left(\frac{M}{\alpha_m I/\ I^*}\right)\left(f\eta^{q_{ij}} + r_i\frac{\sigma}{\sigma_{max}}\right)\right)$$

Equation 3-83

$$F_n = (1-\mathtt{SEG.IFRA})h_i\left(\frac{V}{V_{i0}^{*}}\right)^{\mathtt{SEG.VPOW}} \times \left(\kappa_i\sigma - \left(\frac{N}{\alpha_n V/\ V^{*}}\right)\left(f\eta^{q_{ij}} + r_i\frac{\sigma}{\sigma_{max}}\right)\right)$$

Equation 3-84

Unlike the pair surface recombination model, the interstitial and vacancy concentrations and the dopant concentration at silicon boundaries are directly affected by Equation 3-79 and Equation 3-80. The generation flux of interstitial and vacancy at silicon boundary due to pair segregation is $-F_m$ and $-F_n$, respectively. And the segregation flux of total dopant concentration is calculated by $F_m + F_n$.

Turning on the **PAIR.SEG** parameter in the **METHOD** statement specifies the pair segregation model, instead of surface recombination model (Equation 3-73 and Equation 3-74). The parameters **SEG.IFRA**, **SEG.IPOW**, **SEG.VPOW** can be defined in the **IMPURITY** statement.

Automatic Transition from 5- to 4-, and 3-Stream Diffusion

Turning on the boolean parameter **PAIR.EQU** in the **METHOD** statement enables the automatic transition from the 5-stream diffusion model to 4- or 3- stream diffusion model. This method is available only when the **DDC.FULL** model is disabled. It considerably speeds up because the concentrations of dopant-defect pairs are analytically calculated instead of numerically solving the equations. After each incremental time step in diffusion, the following conditions are tested at all nodes to determine if the local equilibrium assumption can be applied:

$$\left|M - M_{eq}\right| < \mathtt{T.EQPAIR}\cdot\ \mathtt{REL.ERR}$$

Equation 3-85

$$\left|N - N_{eq}\right| < \mathtt{T.EQPAIR}\cdot\ \mathtt{REL.ERR}$$

Equation 3-86

where M_{eq} and N_{eq} are the concentrations of dopant-interstitial pairs and dopant-vacancy pairs in local equilibrium. respectively.

When the condition in Equation 3-85 is satisfied, the concentration of dopant-interstitial pairs is analytically calculated instead of numerically solving Equation 3-61.

$$\frac{M}{\alpha_m} = \frac{(C_m - N)\dfrac{I}{I^*}\left(\dfrac{K_m + K_{mv}\dfrac{I^*}{I}}{K_m + K_{mv}\dfrac{V}{V^*}}\right)}{1 + \alpha_m\dfrac{I}{I^*}\left(\dfrac{K_m + K_{mv}\dfrac{I^*}{I}}{K_m + K_{mv}\dfrac{V}{V^*}}\right)} = S\frac{I}{I^*}\left(\frac{K_m + K_{mv}\dfrac{I^*}{I}}{K_m + K_{mv}\dfrac{V}{V^*}}\right)$$

Equation 3-87

and the substitutional concentration is computed from:

$$S = \frac{C_m - N}{1 + \alpha_m \frac{I}{I^*}\left(\frac{K_m + K_{mv}\frac{I^*}{I}}{K_m + K_{mv}\frac{V}{V^*}}\right)} \qquad \text{Equation 3-88}$$

The reaction rates $G_m - R_m$ and $G_{mv} - R_{mv}$ are reformulated as follows:

$$G_m - R_m = \left(\frac{\partial M}{\partial t} + \vec{\nabla} \cdot \vec{J}_m + R_{DDCM}\right) + (G_{mv} - R_{mv}) \qquad \text{Equation 3-89}$$

$$G_{mv} - R_{mv} = -\frac{K_{mv}\frac{V}{V^*}}{K_m + K_{mv}\frac{V}{V^*}}\left(\frac{\partial M}{\partial t} + \vec{\nabla} \cdot \vec{J}_m + R_{DDCM}\right)$$

$$+ \frac{K_m K_{mv}}{K_m + K_{mv}\frac{V}{V^*}} S\left(\frac{I}{I^*}\frac{V}{V^*} - 1\right)$$

Equation 3-90

Likewise, when the condition Equation 3-86 is satisfied, the concentration of dopant-vacancy pairs is analytically calculated instead of numerically solving Equation 3-62.

$$\frac{N}{\alpha_n} = \frac{(C_m - M)\frac{V}{V^*}\left(\frac{K_n + K_{ni}\frac{V^*}{V}}{K_n + K_{ni}\frac{I}{I^*}}\right)}{1 + \alpha_n \frac{V}{V^*}\left(\frac{K_n + K_{ni}\frac{V^*}{V}}{K_n + K_{ni}\frac{I}{I^*}}\right)} = S\frac{V}{V^*}\left(\frac{K_n + K_{ni}\frac{V^*}{V}}{K_n + K_{ni}\frac{I}{I^*}}\right) \qquad \text{Equation 3-91}$$

and the substitutional concentration is computed from:

$$S = \frac{C_m - M}{1 + \alpha_n \frac{V}{V^*}\left(\frac{K_n + K_{ni}\frac{V^*}{V}}{K_n + K_{ni}\frac{I}{I^*}}\right)} \qquad \text{Equation 3-92}$$

The reaction rates $G_n - R_n$ and $G_{ni} - R_{ni}$ are reformulated as follows:

$$G_n - R_n = \left(\frac{\partial N}{\partial t} + \vec{\nabla} \cdot \vec{J}_n + R_{DDCN}\right) + (G_{ni} - R_{ni}) \qquad \text{Equation 3-93}$$

$$G_{ni} - R_{ni} = -\frac{K_{ni}\frac{I}{I^*}}{K_n + K_{ni}\frac{I}{I^*}}\left(\frac{\partial N}{\partial t} + \vec{\nabla} \cdot \vec{J}_n + R_{DDCN}\right)$$

$$+ \frac{K_n K_{ni}}{K_n + K_{ni}\frac{I}{I^*}} S\left(\frac{I}{I^*}\frac{V}{V^*} - 1\right) \qquad \text{Equation 3-94}$$

When both conditions of Equation 3-85 and Equation 3-86 are satisfied, the 5-stream diffusion model becomes the exactly same as the **`PD.FULL`** model.

Reaction Rate Constants

The K_m and K_n terms arise from the competition between the kick-out and Frank-Turnbull mechanisms of pair formation Reference [9].

- K_m and K_n are the reaction rate constants for generation of dopant-interstitial and dopant-vacancy pairs.
- K_{mv} is the rate constant for the reaction of dopant-interstitial pairs with vacancies.
- K_{ni} is the rate constant for the reaction of dopant-vacancy pairs with interstitials.

Thus, K_{mv} and K_{ni} are the reaction rate constants for dopant-assisted recombination of interstitials and vacancies. The values of the reaction rate constants are computed from:

$$K_m = \sum_{k=-2}^{1} \left(g_{mk} \phi_{Ik} \eta^{-k}\right) \qquad \text{Equation 3-95}$$

$$K_n = \sum_{k=-2}^{1} \left(g_{nk} \phi_{Vk} \eta^{-k}\right) \qquad \text{Equation 3-96}$$

$$K_{mv} = \sum_{j=-2}^{1} \sum_{k=-2}^{1} \left(r_{mvjk} \frac{D_{mj}}{D_{Mj}} \phi_{Vk} \eta^{-(j+k)}\right) \qquad \text{Equation 3-97}$$

$$K_{ni} = \sum_{j=-2}^{1}\sum_{k=-2}^{1}\left(r_{nijk}\frac{D_{nj}}{D_{Nj}}\phi_{Ik}\eta^{-(j+k)}\right)$$

Equation 3-98

where g_{mk}, g_{nk}, r_{mvjk}, and r_{nijk} are given by:

$$g_{mk} = \mathbf{R.I.S}\cdot\ 4\pi\overline{D_I}\underline{I_i^*}\ \exp\left(-\frac{\mathbf{E.I.S}}{kT}\right)$$

Equation 3-99

$$g_{nk} = \mathbf{R.V.S}\cdot\ 4\pi\overline{D_V}\underline{V_i^*}\ \exp\left(-\frac{\mathbf{E.V.S}}{kT}\right)$$

Equation 3-100

$$r_{mvjk} = \mathbf{R.IP.V}\cdot\ 4\pi(\overline{D_{Mj}}+\overline{D_{Vk}})\underline{V_i^*}\ \exp\left(-\frac{\mathbf{E.IP.V}}{kT}\right)$$

Equation 3-101

$$r_{nijk} = \mathbf{R.VP.I}\cdot\ 4\pi(\overline{D_{Nj}}+\overline{D_{Ik}})\underline{I_i^*}\ \exp\left(-\frac{\mathbf{E.VP.I}}{kT}\right)$$

Equation 3-102

The capture radii **R.I.S**, **R.V.S**, **R.IP.V**, and **R.VP.I** and barrier energies **E.I.S**, **E.V.S**, **E.IP.V**, and **E.VP.I** are specified by parameters on the **IMPURITY**, **ANTIMONY**, **ARSENIC**, **BORON**, and **PHOSPHORUS** statements. The capture radii and barrier energies can be dependent on the charge state by setting the charge-state parameters **C.INTERS**, **C.VACANC**, **C.IPAIR** and **C.VAPIR**. Instead, turning on **C.I.ALL**, **C.V.ALL**, **C.IP.ALL** and **C.VP.ALL** specifies the charge-independence.

Turning on the parameter **KO.FULL** in the **METHOD** statement specifies to use the consistent point-defect diffusivity with the that of a point-defect diffusion for kick-out reaction rate.

$$\overline{D_{Ik}} = F_{DIsts}\cdot\ \mathbf{D.FACTOR}\cdot\ (\mathbf{D.F11}+\mathbf{D.F22})/\ 2\cdot\ D_{Ik}$$

Equation 3-103

$$\overline{D_{Vk}} = F_{DVsts}\cdot\ \mathbf{D.FACTOR}\cdot\ (\mathbf{D.F11}+\mathbf{D.F22})/\ 2\cdot\ D_{Vk}$$

Equation 3-104

$$\underline{I_i^*} = F_{Ists}\cdot\ \mathbf{CEQUIL.F}\cdot\ I_{i0}^*$$

Equation 3-105

$$\underline{V_i^*} = F_{Vsts}\cdot\ \mathbf{CEQUIL.F}\cdot\ V_{i0}^*$$

Equation 3-106

By default, the **KO.FULL** parameter is turned off so that:

$$\overline{D_{Ik}} = D_{Ik},\ \overline{D_{Vk}} = D_{Vk},\ \underline{I_i^*} = I_{i0}^* \text{ and } \underline{V_i^*} = V_{i0}^*$$

Equation 3-107

For the detail description of the stress-dependent multiplication factors F_{DIsts}, F_{DVsts}, F_{Ists}, F_{Vsts}, and the user-defined multiplication factor parameters,

D.FACTOR, **D.F11**, **D.F22**, and **CEQUIL.F**, see Diffusion of Point Defects, p. 3-59.

Turning on the parameter **FT.FULL** in the **METHOD** statement specifies to calculate Equation 3-103 to Equation 3-106 and the following:

$$\overline{D_{Mk}} = (F_{stn} \cdot F_{sts} \cdot \mathtt{FGB}) \cdot \mathtt{DI.FAC} \cdot (\mathtt{DI.F11} + \mathtt{DI.F22})/2 \cdot D_{Mk}$$

Equation 3-108

$$\overline{D_{Nk}} = (F_{stn} \cdot F_{sts} \cdot \mathtt{FGB}) \cdot \mathtt{DV.FAC} \cdot (\mathtt{DV.F11} + \mathtt{DV.F22})/2 \cdot D_{Nk}$$

Equation 3-109

By default, the **FT.FULL** parameter is turned off, which specifies to set $\overline{D_{Mk}} = D_{Mk}$ and $\overline{D_{Nk}} = D_{Nk}$ as well as Equation 3-107.

The interstitial and vacancy diffusivities (D_{Ik} and D_{Vk}) and charge fractions (ϕ_{Ik} and ϕ_{Vk}) are described in Diffusion of Point Defects, p. 3-59.

Three plausible assumptions are provided to define the dopant-assisted recombination rates K_{mv} and K_{ni} in the **IMPURITY** statement:

- **IP.V.LOW** assumes that only (z+k)-charged dopant-interstitial paris and (-k)-charged vacancies react in bulk material.(default).
- **IP.V.MED** assumes that z-charged dopant-interstitial pairs (or uncharged vacancies) can react with vacancies (or dopant-interstitial pairs) in any charged state, and that (z+k)-charged dopant-interstitial paris and (-k)-charged vacancies react in bulk material.
- **IP.V.HIGH** assumes that dopant-interstitial pairs in any charge state can react with vacancies in any charge state.

Likewise, **VP.I.LOW**, **VP.I.MED**, and **VP.I.HIGH** for the reaction of dopant-vacancy pairs and interstitials.

The dopant-assisted recombination factors are calculated only if **PAIR.REC** is true (set on the **METHOD** statement, directly or by specifying **PD.FULL**); otherwise, they are set to 1.

If neither **PAIR.SAT** nor **PAIR.REC** is set, Equations 3-54 and 3-55 reduce to:

$$\frac{M}{M'} = \frac{I}{I^*}$$

Equation 3-110

$$\frac{N}{N'} = \frac{V}{V^*}$$

Equation 3-111

This is the approximation used in many other process simulators.

If the spatial variation of M/M' and N/N' is small, you can approximate Equations 3-30, 3-31, and 3-32 by:

$$\frac{\partial C}{\partial t} = -\vec{\nabla} \cdot \left[-\left(D_m \frac{M}{M'} + D_n \frac{N}{N'} \right) \left(\vec{\nabla} C_m - z_s C_m \frac{q\vec{E}}{kT} \right) \right] \qquad \text{Equation 3-112}$$

This form of the equation (with the approximation of Equations 3-110 and 3-111) is used for the **TWO.DIM** model in TSUPREM-4 prior to version 6.0. It is used in versions 6.0 and later (without the approximation of Equations 3-110 and 3-111) if **PAIR.GRA** and **PD.PFLUX** are false (reset on the **METHOD** statement, directly or by specifying **PD.TRANS**). (Equations 3-30, 3-31, and 3-32 are always used if **PD.PFLUX** is true to avoid numerical difficulties.)

The parameter **PD.PAIR** in **IMPURITY** statement specifies that the impurity diffuses by pairing with point-defects. With **PD.TRANS** or **PD.FULL**, **PD.PAIR** is on by default. Turning off **PD.PAIR** specifies to solve Equations 3-53 for the diffusion of the impurity regardless of the specification of **PD.TRANS** or **PD.FULL**.

Activation of Impurities

The electrically active concentration of an impurity may be less than its chemical concentration.

Physical Mechanisms

Four physical mechanisms are considered in calculating the active concentration:

- Solid solubility limits: These are presumed to be the result of precipitation at high concentrations and are present at equilibrium. The formation and dissolution of precipitates occurs over time rather than instantaneously. Implanted dopants are considered to be initially active only to some minimum activation level.
- Dopant clustering: At high concentrations, some dopants form inactive clusters that are stable at equilibrium. These clusters are modeled by assuming that the active concentration is in equilibrium with the precipitated dopant.
- Dopant-defect clusters (e.g., boron-interstitial clusters or BICs): These clusters incorporate point defects as well as dopant atoms. Their formation and dissolution rates depend on the concentration of point defects, which in turn is affected by the formation and dissolution of the clusters. These clusters often form following ion implantation and can persist for minutes or hours at low temperatures but are not stable in equilibrium. They are modeled by following their transient behavior as they form and dissolve.
- Small dopant-defect clusters: These are small, short-lived precursors to the larger dopant-defect clusters described above. They are modeled by assuming that they are in equilibrium with the instantaneous concentrations of dopants and point defects.

Activation Models

These physical mechanisms are included in the following TSUPREM-4 activation models:

- **ACT.EQUI**: Solid solubility limits and dopant clustering are considered. Both are treated as equilibrium phenomena (i.e., the activation is a function of the instantaneous temperature and does not depend on the processing history). This is the simplest model; it is best suited for long anneals where transient effects are insignificant. If you include the final temperature ramp-down cycle when using this model you will probably obtain an unrealistically low activation level. This can be remedied by specifying the final anneal temperature with the **TEMPERAT** parameter on the **SELECT** or **SAVEFILE** statement.
- **ACT.TRAN**: Solid solubility limits and dopant clustering are considered, but the activation occurs over time rather than instantaneously. Implanted dopants are considered to be initially active only to some minimum activation level. For more on the **ACT.TRAN** model, see Transient Clustering Model, p. 3-37.
- **ACT.FULL**: All four of the physical mechanisms described above are included. A transient solution is used for the dopant-defect clusters, while the other mechanisms are treated as equilibrium phenomena.
- **DDC.FULL**: Most advanced activation model. All activation mechanisms described above are included. In addition, the full dynamics of the dopant-defect clustering are taken into account.

The activation model is specified in the **METHOD** statement.

Model Details

The active dopant concentration is found by adjusting the total dopant concentration C to account for grain boundaries (in polycrystalline materials), dopant-defect clusters, small dopant-defect clusters, solid solubility limits, and dopant clustering, in that order. The concentration after adjusting for grain boundaries and dopant-defect clusters is:

$$C_{a1} = C - w_{gb} - C_{cl0} - C_{cl1} - C_{cl2} - \sum \mathbf{DDC.N.D}\, C_{DDCLUST}$$

Equation 3-113

where w_{gb} is the dopant concentration in polycrystalline grain boundaries (see Modeling Polycrystalline Materials, p. 3-161), C_{cl0} is the precipitation concentration(Transient Precipitation Model, p. 3-50), C_{cl1} is the concentration in the dopant-interstitial clusters (Dopant-Interstitial Clustering Model, p. 3-38) and C_{cl2} is the concentration in the dopant-vacancy clusters (Dopant-Vacancy Clustering Model, p. 3-39). $C_{DDCLUST}$ is the concentration in the dopant-defect cluster which consists of **DDC.N.D** dopant atoms.(Full Dynamics of Dopant-Defect Clustering, p. 3-40) The subtraction of C_{cl0}, C_{cl1} and C_{cl2} is done when the **ACT.FULL** model or the **DDC.FULL** model are used. The subtractions of whole $C_{DDCLUST}$'s are only done when the **DDC.FULL** model is used. C_{a1} is adjusted to account for small dopant-defect clusters by:

$$C_a + \mathbf{DDCS.0} \cdot \exp\left(\frac{-\mathbf{DDCS.E}}{kT}\right)\left[C_a\left(\frac{I}{I_i^*}\right)^{\mathbf{IFRACS}}\right]^{\mathbf{DDCS.N}} = C_{a1}$$

Equation 3-114

where **DDCS.0**, **DDCS.E**, **IFRACS**, and **DDCS.N** are parameters on the **IMPURITY** statement. **DDCS.0** and **DDCS.E** specify the fraction of dopants that are clustered in equilibrium and **IFRACS** specifies the number of interstitials associated with each dopant atom. **DDCS.N** is the number of dopant atoms in the cluster, and must be either 1 or 2 (any other value is ignored). The number of interstitials stored in small dopant-defect clusters is given by

$$I_{dds} = \mathbf{IFRACS} \cdot (C_{a1} - C_a)$$

Equation 3-115

When **ACT.TRAN** or **ACT.EQUI** is used, $I_{dds} = 0$.

Solid Solubility Model

At high doping concentrations, the active concentration is limited by solid solubility to:

$$C_{as} = \begin{cases} C_{a2}\ , & C_{a2} \le 0.9C_{ss} \\ C_{ss} - \dfrac{(C_{a2} - 1.1C_{ss})^2}{0.4C_{ss}}, & 0.9C_{ss} \le C_{a2} \le 1.1C_{ss} \\ C_{ss}\ , & C_{a2} \ge 1.1C_{ss} \end{cases}$$

Equation 3-116

where C_{ss} is the solid solubility and C_{a2} is calculated by:

$$C_{a2} + \mathbf{DDCS.0} \cdot \exp\left(\frac{-\mathbf{DDCS.E}}{kT}\right)\left[C_{a2}\left(\frac{I}{I_i^*}\right)^{\mathbf{IFRACS}}\right]^{\mathbf{DDCS.N}} = C_{a1} + C_{cl0}$$

Equation 3-117

These calculations are only done when the **ACT.FULL** model is used; When **ACT.TRAN** or **ACT.EQUI** is used, $C_{a2} = C_{a1}$.

In version 6.4 and earlier, the active concentration is simply:

$$C_{as} = \min(C_{a2},\ C_{ss});$$

this form is still used if **V.COMPAT=6.4** is specified on the **OPTION** statement.

Solid Solubility Tables

The solid solubilities of impurities are represented by tables of values that are specified with the **IMPURITY** statement (see Chapter 4, IMPURITY, p. 4-273). Each table is one dimensional, with up to 100 rows corresponding to the diffusion temperatures. Pairs of temperatures and concentrations are specified using the **SS.TEMP** and **SS.CONC** parameters; a separate **IMPURITY** statement must be used for each table entry. The table can be cleared by specifying the **SS.CLEAR** parameter.

The material to which the table applies is specified with the **MATERIAL** parameter. The solid solubility data for boron, phosphorus, and antimony can also be specified on the **BORON**, **PHOSPHORUS**, and **ANTIMONY** statements, respectively. The solubility data for silicon and polysilicon is obtained from published polynomial approximations Reference [10].

Logarithmic interpolation is used to obtain values of solid solubility for temperatures between the values in a table. For temperatures outside the range of values present in the table, the value is extrapolated using the first two or last two values in the table. In the default tables, the first two values in each table are the same, so the extrapolation results in the first value being used at temperatures below the lowest temperature in the table. Similarly, the last two values of each table are the same, so that the last value is used for temperatures above the highest temperature in the table.

Dopant Clustering Model

In the clustering model, the electrically active impurity concentration $C_a{}^*$ is obtained by solving:

$$C_{as} = C_a{}^* + \left[\texttt{CTN.0} \cdot \exp\left(\frac{-\texttt{CTN.E}}{kT}\right) C_a{}^*\right]^{\texttt{CTN.F}} \quad \text{Equation 3-118}$$

where the parameters **CTN.0**, **CTN.E**, and **CTN.F** are specified in the **IMPURITY** statement (see Chapter 4, IMPURITY, p. 4-273). The clustering parameters for arsenic can also be specified on the **ARSENIC** statement.

Both the solid solubility and clustering models are used to calculate the active concentration $C_a{}^*$, although usually the parameters are chosen so that only one of them affects $C_a{}^*$. If no solid solubility is specified, then $C_{as} = C_{a2}$, while if **CTN.0** is zero, then $C_a{}^* = C_{as}$. When **ACT.EQUI** is used, $C_a = C_a{}^*$

Transient Clustering Model

The transient clustering model simulates the activation of dopant atoms following ion implantation. It does this by solving the transient equation for the active concentration C_a:

$$\frac{\partial(C - C_a)}{\partial t} = \frac{C_a - C_a^*}{\tau_a} \quad \text{Equation 3-119}$$

where C is the chemical concentration of the dopant and C_a^* is the equilibrium active concentration given by C_a in Activation of Impurities, p. 3-34. The time constant for the activation of dopants τ_a is given by:

$$\tau_a = \texttt{T.ACT.0} \times \exp\left(\frac{-\texttt{T.ACT.E}}{kT}\right) \quad \text{Equation 3-120}$$

where **T.ACT.0** and **T.ACT.E** are parameters on the **IMPURITY** statement.

After an implant, it is assumed that all implanted dopant atoms are inactive, i.e., ion implantation increases the value of C but does not change the value of C_a; in amorphized regions, the implanted dopants are assumed to be inactive unless **^CL.INI.A** has been specified for the dopant and material on the **IMPURITY** statement. This assumption is modified by the constraint:

$$C_a \geq \min(C_a^*,\ \texttt{ACT.MIN} \times n_i) \qquad \text{Equation 3-121}$$

where n_i is the intrinsic carrier concentration and **ACT.MIN** is a parameter on the **IMPURITY** statement. Thus activation to a level comparable to n_i occurs instantaneously, after which Equation 3-120 takes over. The result is that transient-enhanced diffusion can occur in the tail of an implanted profile without significant diffusion near the peak.

The transient clustering model is activated by specifying the **ACT.TRAN** parameter on the **METHOD** statement; it is disabled by specifying the **ACT.EQUI** parameter.

Dopant-Interstitial Clustering Model

In some cases, the activation of impurities is strongly dependent on the past sequence of processing conditions to which a wafer has been subjected. An important example is the transient clustering of boron following an ion implantation and annealing at about 900° C or less. This clustering can by modeled by:

$$\frac{\partial C_{cl1}}{\partial t} = \frac{1}{\tau_{dd}}\left(K_{ddF}\, n_i\left(\frac{C_a}{n_i}\right)^{\texttt{DDCF.D.N}} \eta^{\texttt{DDCF.N.N}}\left(\frac{I}{I^*}\right)^{\texttt{DDCF.I.N}} - C_{cl1}\,\eta^{\texttt{DDCR.N.N}}\left(\frac{I}{I^*}\right)^{\texttt{DDCR.I.N}}\right) \qquad \text{Equation 3-122}$$

$$\tau_{dd} = \texttt{DDC.T.0} \cdot \exp\left(-\frac{\texttt{DDC.T.E}}{kT}\right) \qquad \text{Equation 3-123}$$

$$K_{ddF} = \texttt{DDC.F.0} \cdot \exp\left(-\frac{\texttt{DDC.F.E}}{kT}\right) \qquad \text{Equation 3-124}$$

where C_{cl1} is the concentration of dopant atoms in transient dopant-defect clusters and **DDC.T.0**, **DDC.T.E**, **DDC.F.0**, **DDC.F.E**, **DDCF.D.N**, **DDCF.N.N**, **DDCF.I.N**, **DDCR.N.N**, and **DDCR.I.N** are parameters on the **IMPURITY** statement.

This equation models dopant-defect clusters with **DDCF.D.N** dopant atoms per cluster, **DDCF.I.N** interstitials required to form a cluster, **DDCR.I.N** interstitial atoms required to dissolve a cluster, and **DDCF.I.N** – **DDCR.I.N** interstitials in the cluster. **DDC.F.0** and **DDC.F.E** control the amount of clustering while **DDC.T.0** and **DDC.T.E** control the rate of clustering and declustering. Because C_{cl1} is subtracted from the total concentration before computing C_a, the dopant in transient clusters is inactive and immobile.

The net capture rate of interstitials by clusters is given by:

$$R_{dd} = \mathbf{IFRACM}\frac{\partial C_{cl1}}{\partial t} \qquad \text{Equation 3-125}$$

where **IFRACM** is a parameter on the **IMPURITY** statement (see Point Defect Diffusion Equations, p. 3-61). Ideally, **IFRACM** should be equal to $(\mathbf{DDCF.I.N} - \mathbf{DDCR.I.N}) / \mathbf{DDCF.D.N}$, but it is available as a separate parameter for flexibility in modeling.

The dopant-defect clustering model is enabled by specifying **ACT.FULL** on the **METHOD** statement.

Dopant-Vacancy Clustering Model

The dopant-vacancy clustering can by modeled by:

$$\frac{\partial C_{cl2}}{\partial t} = \frac{1}{\tau_{dv}}\left(K_{dvF}n_i\left(\frac{C_a}{n_i}\right)^{\mathbf{DVCF.D.N}}\eta^{\mathbf{DVCF.N.N}}\left(\frac{V}{V^*}\right)^{\mathbf{DVCF.V.N}} - C_{cl2}\eta^{\mathbf{DVCR.N.N}}\left(\frac{V}{V^*}\right)^{\mathbf{DVCR.V.N}}\right) \qquad \text{Equation 3-126}$$

$$\tau_{dv} = \mathbf{DVC.T.0}\cdot\ \exp\left(-\frac{\mathbf{DVC.T.E}}{kT}\right) \qquad \text{Equation 3-127}$$

$$K_{dvF} = \mathbf{DVC.F.0}\cdot\ \exp\left(-\frac{\mathbf{DVC.F.E}}{kT}\right) \qquad \text{Equation 3-128}$$

where C_{cl2} is the concentration of dopant atoms in transient dopant-vacancy clusters and **DVC.T.0**, **DVC.T.E**, **DVC.F.0**, **DVC.F.E**, **DVCF.D.N**, **DVCF.N.N**, **DVCF.V.N**, **DVCR.N.N**, and **DVCR.V.N** are parameters on the **IMPURITY** statement.

This equation models dopant-vacancy clusters with **DVCF.D.N** dopant atoms per cluster, **DVCF.V.N** vacancies required to form a cluster, **DVCR.V.N** vacancies required to dissolve a cluster, and $\mathbf{DVCF.V.N} - \mathbf{DVCR.V.N}$ vacancies in the cluster.

The net capture rate of vacancies by clusters is given by:

$$R_{dv} = \mathbf{VFRACM}\frac{\partial C_{cl2}}{\partial t} \qquad \text{Equation 3-129}$$

where **VFRACM** is a parameter on the **IMPURITY** statement (see Point Defect Diffusion Equations, p. 3-61). Ideally, **IFRACM** should be equal to $(\mathbf{DVCF.V.N} - \mathbf{DVCR.V.N}) / \mathbf{DVCF.D.N}$, but it is available as a separate parameter for flexibility in modeling.

The dopant-vacancy clustering model is enabled by specifying **`ACT.FULL`** on the **`METHOD`** statement.

Full Dynamics of Dopant-Defect Clustering

The **`DDC.FULL`** model is a superset of the **`ACT.FULL`** model. In other words, **`DDC.FULL`** enables to solve not only C0, C1 and C2 solutions but also the dopant-defect cluster solutions for full dynamics. Turning on **`DDC.FULL`** in the **`METHOD`** statement specifies that the full dynamics of dopant-defect clustering are to be considered. In the **`IMPURITY`** statement, the reaction rate constants for a given dopant-defect cluster **`DDCLUSTE`** can be defined by the parameters **`DDC.KF.0`** and **`DDC.KF.E`** for the prefactor and energy barrier respectively of the forward (**`DDCLUSTE`** formation) reaction, and the parameters **`DDC.KR.0`** and **`DDC.KR.E`** for the reverse (**`DDCLUSTE`** dissolution) reaction. When the equilibrium concentration of a dopant-defect cluster is assumed to be identical for all reactions, the parameters **`DDC.EQ.0`** and **`DDC.EQ.E`** can be set instead of **`DDC.KR.0`** and **`DDC.KR.E`**. The reverse reaction constant is calculated by one of the following equations:

$$\mathtt{DDC.KR.0}\cdot\ \exp\left(\frac{-\mathtt{DDC.KR.E}}{kT}\right)$$

Equation 3-130

or

$$\left(\mathtt{DDC.KF.0}\cdot\ \exp\left(\frac{-\mathtt{DDC.KF.E}}{kT}\right)\right)\left(\mathtt{DDC.EQ.0}\cdot\ \exp\left(\frac{-\mathtt{DDC.EQ.E}}{kT}\right)\right)$$

Equation 3-131

The charge state of a dopant-defect cluster can be specified with the one of the parameters, **`CHARGE`**, **`ACCEPTOR`**, **`NEUTRAL`** and **`DONOR`** in the **`IMPURITY`** statement.

The reaction rate constants depend on the charge states of the reactants which can be specified with some of the parameters **`C.INTERS`**, **`C.IPAIR`**, and **`DDC.C.IP`**. Turning on the boolean parameters **`C.I.ALL`** or **`C.IP.ALL`** assumes the reaction rate constants to be independent of the charge states.

S+M

The reaction rate of a z-charged substitutional dopant atom and a dopant-interstitial pair to form a $q_{DDCSLUST}$-charged dopant-defect cluster is given by:

$$R_{DDC,\ S+M} = k_{fSM}S\left(\frac{M}{\alpha_m}\right) - k_{rSM}C_{DDCLUST}\eta^{-2z+q_{DDCLUST}}$$

Equation 3-132

or

$$R_{DDC,\ S+M} = k_{fSM}\left(S\left(\frac{M}{\alpha_m}\right) - r_{EQ,\ SM}C_{DDCLUST}\eta^{-2z+q_{DDCLUST}}\right)$$

Equation 3-133

where:

$$k_{fSM} = \sum_{k} f_{SM,\ k}\left(\frac{D_{mk}}{D_{Mk}}\right)\eta^{-k} \qquad \text{Equation 3-134}$$

$$k_{rSM} = \sum_{k} r_{SM,\ k}\left(\frac{D_{mk}}{D_{Mk}}\right)\eta^{-k} \qquad \text{Equation 3-135}$$

where $f_{SM,k}$ and $r_{SM,k}$ are the forward and reverse reaction rate constants, respectively. By turning on **`C.IP.ALL`** so that the reaction rate constants are independent of the charge state of dopant-interstitial pairs, Equation 3-132 is simplified to;

$$R_{DDC,\ S+M} = f_{SM}S \cdot \ M - r_{SM}\alpha_m C_{DDCLUST}\eta^{-2z+q_{DDCLUST}}$$

Equation 3-136

S+N The reaction rate of a z-charged substitutional dopant atom and a dopant-vacancy pair to form a $q_{DDCSLUST}$-charged dopant-defect cluster is given by:

$$R_{DDC,\ S+N} = k_{fSN}S\left(\frac{N}{\alpha_n}\right) - k_{rSN}C_{DDCLUST}\eta^{-2z+q_{DDCLUST}} \qquad \text{Equation 3-137}$$

or

$$R_{DDC,\ S+N} = k_{fSN}\left(S\left(\frac{N}{\alpha_n}\right) - r_{EQ,\ SN}C_{DDCLUST}\eta^{-2z+q_{DDCLUST}}\right)$$

Equation 3-138

where:

$$k_{fSN} = \sum_{k} f_{SN,\ k}\left(\frac{D_{nk}}{D_{Nk}}\right)\eta^{-k} \qquad \text{Equation 3-139}$$

$$k_{rSN} = \sum_{k} r_{SN,\ k}\left(\frac{D_{nk}}{D_{Nk}}\right)\eta^{-k} \qquad \text{Equation 3-140}$$

where $f_{SN,k}$ and $r_{SN,k}$ are the forward and reverse reaction rate constants, respectively. By turning on **`C.VP.ALL`** so that the reaction rate constants are independent of the charge state of dopant-interstitial pairs, Equation 3-132 is simplified to;

$$R_{DDC,\ S+N} = f_{SN}S \cdot \ N - r_{SN}\alpha_n C_{DDCLUST}\eta^{-2z+q_{DDCLUST}} \qquad \text{Equation 3-141}$$

I+M

The reaction rate of a self-interstitial and a dopant-interstitial pair to form a $q_{DDCSLUST}$-charged dopant-defect cluster is given by;

$$R_{DDC,\ I+M} = k_{fIM}\left(\frac{I}{I^*}\right)\left(\frac{M}{\alpha_m}\right) - k_{rIM}C_{DDCLUST}\eta^{-z+q_{DDCLUST}}$$

Equation 3-142

or:

$$R_{DDC,\ I+M} = k_{fIM}\left(\left(\frac{I}{I^*}\right)\left(\frac{M}{\alpha_m}\right) - r_{EQ,\ IM}C_{DDCLUST}\eta^{-z+q_{DDCLUST}}\right)$$

Equation 3-143

where:

$$k_{fIM} = I_i^* \sum_{j,\ k} f_{IM,\ jk}\phi_{Ij}\left(\frac{D_{mk}}{D_{Mk}}\right)\eta^{-(j+k)}$$

Equation 3-144

$$k_{rIM} = I_i^* \sum_{j,\ k} r_{IM,\ jk}\phi_{Ij}\left(\frac{D_{mk}}{D_{Mk}}\right)\eta^{-(j+k)}$$

Equation 3-145

where $f_{IM,jk}$ and $r_{IM,jk}$ are the forward and reverse reaction rate constants, respectively. By turning on **`C.I.ALL`** and **`C.IP.ALL`** so that the reaction rate constants are independent of the charge states of interstitials and dopant-interstitial pairs, Equation 3-142 is simplified to:

$$R_{DDC,\ I+M} = f_{IM}I \cdot M - r_{IM}\alpha_m I^* C_{DDCLUST}\eta^{-z+q_{DDCLUST}}$$

Equation 3-146

V+N

The reaction rate of a vacancy and a dopant-vacancy pair to form a $q_{DDCSLUST}$-charged dopant-defect cluster is given by;

$$R_{DDC,\ V+N} = k_{fVN}\left(\frac{V}{V^*}\right)\left(\frac{N}{\alpha_n}\right) - k_{rVN}C_{DDCLUST}\eta^{-z+q_{DDCLUST}}$$

Equation 3-147

or:

$$R_{DDC,\ V+N} = k_{fVN}\left(\left(\frac{V}{V^*}\right)\left(\frac{N}{\alpha_n}\right) - r_{EQ,\ VN}C_{DDCLUST}\eta^{-z+q_{DDCLUST}}\right)$$

Equation 3-148

where:

$$k_{fVN} = V_i^* \sum_{j,\ k} f_{VN,\ jk}\phi_{Vj}\left(\frac{D_{nk}}{D_{Nk}}\right)\eta^{-(j+k)}$$

Equation 3-149

$$k_{rVN} = V_i^* \sum_{j,\ k} r_{VN,\ jk} \phi_{Vj} \left(\frac{D_{nk}}{D_{Nk}}\right) \eta^{-(j+k)}$$

Equation 3-150

where $f_{VN,jk}$ and $r_{VN,jk}$ are the forward and reverse reaction rate constants, respectively. By turning on **`C.V.ALL`** and **`C.VP.ALL`** so that the reaction rate constants are independent of the charge states of interstitials and dopant-interstitial pairs, Equation 3-142 is simplified to:

$$R_{DDC,\ V+N} = f_{VN} V \cdot \ N - r_{VN} \alpha_n V^* C_{DDCLUST} \eta^{-z + q_{DDCLUST}}$$

Equation 3-151

M+M The reaction rate of two dopant-interstitial pairs to form a $q_{DDCSLUST}$-charged dopant-defect cluster is given by:

$$R_{DDC,\ M+M} = k_{fMM} \left(\frac{M}{\alpha_m}\right)^2 - k_{rMM} C_{DDCLUST} \eta^{-2z + q_{DDCLUST}}$$

Equation 3-152

or:

$$R_{DDC,\ M+M} = k_{fMM} \left(\left(\frac{M}{\alpha_m}\right)^2 - r_{EQ,\ MM} C_{DDCLUST} \eta^{-2z + q_{DDCLUST}} \right)$$

Equation 3-153

where:

$$k_{fMM} = \sum_{j,\ k} f_{MM,\ jk} \left(\frac{D_{mj}}{D_{Mj}}\right) \left(\frac{D_{mk}}{D_{Mk}}\right) \eta^{-(j+k)}$$

Equation 3-154

$$k_{rMM} = \sum_{j,\ k} r_{MM,\ k} \left(\frac{D_{mj}}{D_{Mj}}\right) \left(\frac{D_{mk}}{D_{Mk}}\right) \eta^{-(j+k)}$$

Equation 3-155

where $f_{MM,jk}$ and $r_{MM,jk}$ are the forward and reverse reaction rate constants, respectively. By turning on **`C.IP.ALL`** so that the reaction rate constants are independent of the charge state of dopant-interstitial pairs, Equation 3-152 is simplified to:

$$R_{DDC,\ M+M} = f_{MM} M^2 - r_{MM} \alpha_m^2 C_{DDCLUST} \eta^{-2z + q_{DDCLUST}}$$

Equation 3-156

N+N The reaction rate of two dopant-vacancy pairs to form a $q_{DDCSLUST}$-charged dopant-defect cluster is given by:

$$R_{DDC,\ N+N} = k_{fNN} \left(\frac{N}{\alpha_n}\right)^2 - k_{rNN} C_{DDCLUST} \eta^{-2z + q_{DDCLUST}}$$

Equation 3-157

or:

$$R_{DDC,\ N+N} = k_{fNN}\left(\left(\frac{N}{\alpha_n}\right)^2 - r_{EQ,\ NN}C_{DDCLUST}\eta^{-2z + q_{DDCLUST}}\right)$$

Equation 3-158

where:

$$k_{fNN} = \sum_{j,\ k} f_{NN,\ jk}\left(\frac{D_{nj}}{D_{Nj}}\right)\left(\frac{D_{nk}}{D_{Nk}}\right)\eta^{-(j+k)}$$

Equation 3-159

$$k_{rNN} = \sum_{j,\ k} r_{NN,\ k}\left(\frac{D_{nj}}{D_{Nj}}\right)\left(\frac{D_{nk}}{D_{Nk}}\right)\eta^{-(j+k)}$$

Equation 3-160

where $f_{NN,jk}$ and $r_{NN,jk}$ are the forward and reverse reaction rate constants, respectively. By turning on **C.VP.ALL** so that the reaction rate constants are independent of the charge state of dopant-interstitial pairs, Equation 3-152 is simplified to:

$$R_{DDC,\ N+N} = f_{NN}N^2 - r_{NN}\alpha_n^2 C_{DDCLUST}\eta^{-2z + q_{DDCLUST}}$$

Equation 3-161

DDC+M The reaction rate of a q_{DDC}-charged dopant-defect cluster and a dopant-interstitial pair to form a $q_{DDCSLUST}$-charged dopant-defect cluster is given by:

$$R_{DDC,\ C+M} = k_{fCM}C_{DDC}\left(\frac{M}{\alpha_m}\right) - k_{rCM}C_{DDCLUST}\eta^{-z - q_{DDC} + q_{DDCLUST}}$$

Equation 3-162

or:

$$R_{DDC,\ C+M} = k_{fCM}\left(C_{DDC}\left(\frac{M}{\alpha_m}\right) - r_{EQ,CM}C_{DDCLUST}\eta^{-z - q_{DDC} + q_{DDCLUST}}\right)$$

Equation 3-163

where:

$$k_{fCM} = \sum_{k} f_{CM,\ k}\left(\frac{D_{mk}}{D_{Mk}}\right)\eta^{-k}$$

Equation 3-164

$$k_{rCM} = \sum_{k} r_{CM,\ k}\left(\frac{D_{mk}}{D_{Mk}}\right)\eta^{-k}$$

Equation 3-165

$$r_{EQ,CM} = \left.\frac{f_{CM,\ k}}{r_{CM,\ k}}\right|_{\text{all k}}$$

Equation 3-166

where $f_{CM,k}$ and $r_{CM,k}$ are the forward and reverse reaction rate constants, respectively. By turning on **`C.IP.ALL`** so that the reaction rate constants are independent of the charge state of dopant-interstitial pairs, Equation 3-162 is simplified to:

$$R_{DDC,\ C+M} = f_{CM}C_{DDC}M - r_{CM}\alpha_m C_{DDCLUST}\eta^{-z-q_{DDC}+q_{DDCLUST}}$$

Equation 3-167

DDC+N The reaction rate of a q_{DDC}-charged dopant-defect cluster and a dopant-vacancy pair to form a $q_{DDCSLUST}$-charged dopant-defect cluster is given by:

$$R_{DDC,\ C+N} = k_{fCN}C_{DDC}\left(\frac{N}{\alpha_n}\right) - k_{rCN}C_{DDCLUST}\eta^{-z-q_{DDC}+q_{DDCLUST}}$$

Equation 3-168

or:

$$R_{DDC,\ C+N} = k_{fCN}\left(C_{DDC}\left(\frac{N}{\alpha_n}\right) - r_{EQ,CM}C_{DDCLUST}\eta^{-z-q_{DDC}+q_{DDCLUST}}\right)$$

Equation 3-169

where:

$$k_{fCN} = \sum_k f_{CN,\ k}\left(\frac{D_{nk}}{D_{Nk}}\right)\eta^{-k}$$

Equation 3-170

$$k_{rCN} = \sum_k r_{CN,\ k}\left(\frac{D_{nk}}{D_{Nk}}\right)\eta^{-k}$$

Equation 3-171

$$r_{EQ,CN} = \left.\frac{f_{CN,\ k}}{r_{CN,\ k}}\right|_{\text{all k}}$$

Equation 3-172

where $f_{CN,k}$ and $r_{CN,k}$ are the forward and reverse reaction rate constants, respectively. By turning on **`C.VP.ALL`** so that the reaction rate constants are independent of the charge state of dopant-interstitial pairs, Equation 3-162 is simplified to:

$$R_{DDC,\ C+N} = f_{CN}C_{DDC}N - r_{CN}\alpha_n C_{DDCLUST}\eta^{-z-q_{DDC}+q_{DDCLUST}}$$

Equation 3-173

DDC+I The reaction rate of a q_{DDC}-charged dopant-defect cluster and an interstitial to form a $q_{DDCSLUST}$-charged dopant-defect cluster is given by:

$$R_{DDC,\ C+I} = k_{fCI}C_{DDC}\left(\frac{I}{I^*}\right) - k_{rCI}C_{DDCLUST}\eta^{-q_{DDC}+q_{DDCLUST}}$$

Equation 3-174

or:

$$R_{DDC,\ C+I} = k_{fCI}\left(C_{DDC}\left(\frac{I}{I^*}\right) - r_{EQ,\ CI}C_{DDCLUST}\eta^{-q_{DDC}+q_{DDCLUST}}\right)$$

Equation 3-175

where:

$$k_{fCI} = I_i^*\sum_k f_{CI,\ k}\phi_{Ik}\eta^{-k}$$

Equation 3-176

$$k_{rCI} = I_i^*\sum_k r_{CI,\ k}\phi_{Ik}\eta^{-k}$$

Equation 3-177

where $f_{CI,k}$ and $r_{CI,k}$ are the forward and reverse reaction rate constants, respectively. By turning on **`C.I.ALL`** so that the reaction rate constants are independent of the charge state of interstitials, Equation 3-174 is simplified to:

$$R_{DDC,\ C+I} = f_{CI}C_{DDC}I - r_{CI}C_{DDCLUST}I^*\eta^{-q_{DDC}+q_{DDCLUST}}$$

Equation 3-178

DDC+V The reaction rate of a q_{DDC}-charged dopant-defect cluster a vacancy to form a $q_{DDCSLUST}$-charged dopant-defect cluster is given by:

$$R_{DDC,\ C+V} = k_{fCV}C_{DDC}\left(\frac{V}{V^*}\right) - k_{rCV}C_{DDCLUST}\eta^{-q_{DDC}+q_{DDCLUST}}$$

Equation 3-179

or:

$$R_{DDC,\ C+V} = k_{fCV}\left(C_{DDC}\left(\frac{V}{V^*}\right) - r_{EQ,\ CV}C_{DDCLUST}\eta^{-q_{DDC}+q_{DDCLUST}}\right)$$

Equation 3-180

where:

$$k_{fCV} = V_i^*\sum_k f_{CV,\ k}\phi_{Vk}\eta^{-k}$$

Equation 3-181

$$k_{rCV} = V_i^{*} \sum_{k} r_{CV,\ k} \phi_{Vk} \eta^{-k}$$ Equation 3-182

where $f_{CV,k}$ and $r_{CV,k}$ are the forward and reverse reaction rate constants, respectively. By turning on `C.V.ALL` so that the reaction rate constants are independent of the charge state of interstitials, Equation 3-174 is simplified to:

$$R_{DDC,\ C+V} = f_{CV} C_{DDC} V - r_{CV} C_{DDCLUST} V^{*} \eta^{-q_{DDC} + q_{DDCLUST}}$$ Equation 3-183

M+N The reaction rate of a dopant-interstitial pair and a dopant-vacancy pair to form a $q_{DDCSLUST}$-charged dopant-defect cluster is given by:

$$R_{DDC,\ M+N} = k_{fMN} \left(\frac{M}{\alpha_m}\right) \left(\frac{N}{\alpha_n}\right) - k_{rMN} C_{DDCLUST} \eta^{-2z + q_{DDCLUST}}$$ Equation 3-184

where:

$$k_{fMN} = \sum_{j,\ k} f_{MN,\ jk} \left(\frac{D_{mj}}{D_{Mj}}\right) \left(\frac{D_{nk}}{D_{Nk}}\right) \eta^{-(j+k)}$$ Equation 3-185

$$k_{rMN} = \sum_{j,\ k} r_{MN,\ jk} \left(\frac{D_{mj}}{D_{Mj}}\right) \left(\frac{D_{nk}}{D_{Nk}}\right) \eta^{-(j+k)}$$ Equation 3-186

where $f_{MN,jk}$ and $r_{MN,jk}$ are the forward and reverse reaction rate constants, respectively. By turning on `C.IP.ALL` so that the reaction rate constants are independent of the charge state of dopant-interstitial pairs, Equation 3-187 is simplified to:

$$R_{DDC,\ I(n)+M} = f_{I(n)M} I_{(n)} M - r_{I(n)M} \alpha_m C_{DDCLUST} \eta^{-z + q_{DDCLUST}}$$

$I_{(n)}$+M The reaction rate of a *n*-size interstitial cluster and a dopant-interstitial pair to form a $q_{DDCSLUST}$-charged dopant-defect cluster is given by:

$$R_{DDC,\ I(n)+M} = k_{fI(n)M} I_{(n)} \left(\frac{M}{\alpha_m}\right) - k_{rI(n)M} C_{DDCLUST} \eta^{-z + q_{DDCLUST}}$$ Equation 3-187

where:

$$k_{fI(n)M} = \sum_{k} f_{I(n)M,\ k} \left(\frac{D_{mk}}{D_{Mk}}\right) \eta^{-k}$$ Equation 3-188

$$k_{rI(n)M} = \sum_k r_{I(n)M,\ k}\left(\frac{D_{mk}}{D_{Mk}}\right)\eta^{-k}$$ Equation 3-189

where $f_{I(n)M,k}$ and $r_{I(n)M,k}$ are the forward and reverse reaction rate constants, respectively. By turning on **`C.IP.ALL`** so that the reaction rate constants are independent of the charge state of dopant-interstitial pairs, Equation 3-187 is simplified to:

$$R_{DDC,\ I(n)+M} = f_{I(n)M}I_{(n)}M - r_{I(n)M}\alpha_m C_{DDCLUST}\eta^{-z+q_{DDCLUST}}$$ Equation 3-190

I_{n+1} + N

$$I_{n+1} + XV^{z+k} \leftrightarrow XI_n^l + (z+k-l)p$$ Equation 3-191

$$R_{DDC,\ I_{n+1}+N} = k_{fI_{n+1}N}\left(I_{n+1}\left(\frac{N}{\alpha_n}\right) - r_{EQ,\ I_{n+1}N}C_{\mathrm{DDC.R}}\eta^{-z+q_{\mathrm{DDC.R}}}\right)$$ Equation 3-192

where I_{n+1} is the small interstitial cluster with size $n+1$

$$k_{fI_{n+1}N} = \sum_k f_{I_{n+1}N,\ k}\left(\frac{D_{nk}}{D_{Nk}}\right)\eta^{-k}$$ Equation 3-193

$$f_{I_{n+1}N,\ k} = \mathtt{DDC.KF.0}\exp\left(-\frac{\mathtt{DDC.KF.E}}{kT}\right)$$ Equation 3-194

$$r_{EQ,\ I_{n+1}N} = \mathtt{DDC.EQ.0}\exp\left(-\frac{\mathtt{DDC.EQ.E}}{kT}\right)$$ Equation 3-195

The parameters for the reaction is defined in the **`IMPURITY`** statement. When **`C.VP.ALL`** is true (default),

$$R_{DDC,\ I_{n+1}+N} = f_{I_{n+1}N}(I_{n+1}N - r_{EQ,\ I_{n+1}N,\ k}\alpha_n C_{\mathrm{DDC.R}}\eta^{-z+q_{\mathrm{DDC.R}}})$$ Equation 3-196

The reaction is taken into account in the equations for dopant-vacancy pairing and interstitial clustering as follows;

$$\frac{\partial N}{\partial t} = \left.\frac{\partial N}{\partial t}\right|_0 - R_{DDC,\ I_{n+1}+N}$$ Equation 3-197

$$\frac{\partial I_{n+1}}{\partial t} = \left.\frac{\partial I_{n+1}}{\partial t}\right|_0 - R_{DDC,\ I_{n+1}+N}$$ Equation 3-198

$$\frac{\partial C_{\mathrm{DDC.R}}}{\partial t} = \left.\frac{\partial C_{\mathrm{DDC.R}}}{\partial t}\right|_0 + R_{DDC,\ I_{n+1}+N} \qquad \text{Equation 3-199}$$

XV$_2$ + I

$$XV_2^l + I^j \leftrightarrow XV^{z+k} + (-z+l+j-k)p \qquad \text{Equation 3-200}$$

$$R_{DDC,\ XV_2+I} = k_{fXV_2I}\left(C_{\mathrm{DDC.L}}\left(\frac{I}{I^*}\right) - r_{EQ,\ XV_2I}\left(\frac{N}{\alpha_n}\right)\eta^{z-q_{\mathrm{DDC.L}}}\right) \qquad \text{Equation 3-201}$$

where:

$$k_{fXV_2I} = I_i^* \sum_j f_{XV_2I,j}\phi_{Ij}\eta^{-j} \qquad \text{Equation 3-202}$$

$$f_{XV_2I,j} = \mathtt{DDC.KF.0}\exp\left(-\frac{\mathtt{DDC.KF.E}}{kT}\right) \qquad \text{Equation 3-203}$$

$$r_{EQ,\ XV_2I} = \mathtt{DDC.EQ.0}\exp\left(-\frac{\mathtt{DDC.EQ.E}}{kT}\right) \qquad \text{Equation 3-204}$$

When `C.I.ALL` is true (default), then Equation 3-201 will be;

$$R_{DDC,\ XV_2+I} = f_{XV_2I}\left(C_{\mathrm{DDC.L}}I - r_{EQ,\ XV_2I}I^*\left(\frac{N}{\alpha_n}\right)\eta^{z-q_{\mathrm{DDC.L}}}\right) \qquad \text{Equation 3-205}$$

The reaction is taken into account in the equations for dopant-vacancy pairs and interstitials as follows;

$$\frac{\partial N}{\partial t} = \left.\frac{\partial N}{\partial t}\right|_0 + R_{DDC,\ XV_2+I} \qquad \text{Equation 3-206}$$

$$\frac{\partial I}{\partial t} = \left.\frac{\partial I}{\partial t}\right|_0 - R_{DDC,\ XV_2+I} \qquad \text{Equation 3-207}$$

$$\frac{\partial C_{\mathrm{DDC.L}}}{\partial t} = \left.\frac{\partial C_{\mathrm{DDC.L}}}{\partial t}\right|_0 - R_{DDC,\ XV_2+I} \qquad \text{Equation 3-208}$$

XI$_2$ + V

$$XI_2^l + V^j \leftrightarrow XI^{z+k} + (-z+l+j-k)p \qquad \text{Equation 3-209}$$

$$R_{DDC,\ XI_2+V} = k_{fXI_2V}\left(C_{\mathrm{DDC.L}}\left(\frac{V}{V^*}\right) - r_{EQ,\ XI_2V}\left(\frac{M}{\alpha_m}\right)\eta^{z-q_{\mathrm{DDC.L}}}\right) \qquad \text{Equation 3-210}$$

where:

$$k_{fXI_2V} = V_i^* \sum_j f_{XI_2V,j} \phi_{Vj} \eta^{-j} \qquad \text{Equation 3-211}$$

$$f_{XI_2V,j} = \mathtt{DDC.KF.0} \exp\left(-\frac{\mathtt{DDC.KF.E}}{kT}\right) \qquad \text{Equation 3-212}$$

$$r_{EQ,\ XI_2V} = \mathtt{DDC.EQ.0} \exp\left(-\frac{\mathtt{DDC.EQ.E}}{kT}\right) \qquad \text{Equation 3-213}$$

When **`C.V.ALL`** is true (default), then Equation 3-210 will be;

$$R_{DDC,\ XI_2+V} = f_{XI_2V}\left(C_{\mathrm{DDC.L}} V - r_{EQ,\ XI_2V} V^* \left(\frac{M}{\alpha_m}\right) \eta^{z-q_{\mathrm{DDC.L}}}\right)$$

Equation 3-214

The reaction is taken into account in the equations for dopant-interstitial pairs and vacancies as follows;

$$\frac{\partial M}{\partial t} = \left.\frac{\partial M}{\partial t}\right|_0 + R_{DDC,\ XI_2+V} \qquad \text{Equation 3-215}$$

$$\frac{\partial V}{\partial t} = \left.\frac{\partial V}{\partial t}\right|_0 - R_{DDC,\ XI_2+V} \qquad \text{Equation 3-216}$$

$$\frac{\partial C_{\mathrm{DDC.L}}}{\partial t} = \left.\frac{\partial C_{\mathrm{DDC.L}}}{\partial t}\right|_0 - R_{DDC,\ XI_2+V} \qquad \text{Equation 3-217}$$

Transient Precipitation Model

When **`ACT.FULL`** or **`DDC.FULL`** models are turned on, the transient precipitation is simulated by solving the following equation:

$$\frac{\partial C_{cl0}}{\partial t} = -\frac{C_{cl0} - C_{cl0}^*}{\tau_{cl0}} \qquad \text{Equation 3-218}$$

where C_{cl0}^* is given by:

$$C_a^* + \mathtt{DDCS.0} \cdot \exp\left(\frac{-\mathtt{DDCS.E}}{kT}\right)\left[C_a^* \left(\frac{I}{I_i^*}\right)^{\mathtt{IFRACS}}\right]^{\mathtt{DDCS.N}}$$
$$= C - w_{gb} - C_{cl1} - C_{cl2} - \sum \mathtt{DDC.N.D} C_{DDCLUST} - C_{cl0}^*$$

Equation 3-219

C_a^* is the equilibrium active concentration (Dopant Clustering Model, p. 3-37).

The time constant τ_{cl0} is given by;

$$\tau_{cl0} = \mathtt{C0.TF.0} \cdot \exp\left(-\frac{\mathtt{C0.TF.E}}{kT}\right) \text{ for } C_{cl0} \leq C^{*}_{cl0} \qquad \text{Equation 3-220}$$

$$\tau_{cl0} = \mathtt{C0.TR.0} \cdot \exp\left(-\frac{\mathtt{C0.TR.E}}{kT}\right) \text{ for } C_{cl0} > C^{*}_{cl0} \qquad \text{Equation 3-221}$$

Equations 3-218 is solved by two methods, i.e. either the analytical method or the numerical method. The analytical method solves by assuming that C^{*}_{cl0} does not vary within a time step. Likewise, with the **ACT.TRAN** model, Equations 3-119 which is similar to Equations 3-218 is always solved by the analytical method. Calculating the solution by the analytical method is much faster than iteratively solving the equation by the numerical method. However, when the total concentration is much larger than the equilibrium active concentration, i.e. $C \sim C^{*}_{cl0}$ and the boundary moves so quickly that the moving flux reduces a lot of the total concentration near surface, the total concentration can be smaller than the clustering concentration, which causes the negative active concentration. Thus, the numerical method which takes account of the moving boundary flux as well as the transient equilibrium active concentration is necessary in such a case. **C0.NUMER**, **C0.ANALY** and **C0.HYBRI** in the **METHOD** statement are the parameters to control the methods. **C0.NUMER** specifies to solve always by the numerical method while **C0.ANALY** does always by the analytical method. **C0.HYBRI** specifies that the numerical method is to be used when a boundary moves or temperature ramps; otherwise, the analytical method to be used.

Note:

> ***In ACT.TRAN, the clustering concentration is intentionally reset so that the active concentration is set to* ACT.MIN $\cdot$ n_i *when it becomes negative.***

DDC and Precipitation Initialization

The parameter **DDCTOT.C** defines the fraction of total dopant-defect clusters to implanted dopants in a crystalline region. And the parameter **DDCTOT.A** defines the fraction of total dopant-defect clusters to the total clusters in the regrowth region. For correct initialization, these parameters must be defined when:

- **DDC.T.0** of both C_{cl1} and C_{cl2} are non-zero and **ACT.FULL** is turned on

or:

- **DDC.FULL** is turned on

Hereinafter, it will be described by assuming that at least one of the above conditions is satisfied.

The parameters **DDCFRACC** and **DDCFRACA** define the fractions of a dopant-defect cluster of full dynamics to the total dopant-defect clusters in a crystalline region after implants and a regrowth region i.e. an amorphous region after implants, respectively.

The cluster solutions are initialized after implantation as follows;

$$C_{cl0} = C'_{cl0} + \Delta C_{cl0} \qquad \text{Equation 3-222}$$

where $$\Delta C_{cl0} = (1 - \mathbf{DDCTOT.C}) \cdot \mathbf{C0.INI.F} \cdot C_{implant} \qquad \text{Equation 3-223}$$

The parameter **C0.INI.F** determines the ratio of the initial precipitation to the implant dopant concentration subtracted by dopant-defect cluster concentration. Initializing a dopant-interstitial cluster requires the adjustment of the interstitial concentration.(Cumulative Damage Model, p. 3-146)

$$C_{cl1} = C'_{cl1} + \Delta C_{cl1} \qquad \text{Equation 3-224}$$

where:

$$\Delta C_{cl1} = \frac{\mathbf{C1.INI.F} \cdot \mathbf{DDCTOT.C}}{\mathbf{C1.INI.F} + \mathbf{C2.INI.F} + \sum \mathbf{DDCFRACC}} \cdot C_{implant}$$

Equation 3-225

$$C_{cl2} = C'_{cl2} + \Delta C_{cl2} \qquad \text{Equation 3-226}$$

where:

$$\Delta C_{cl2} = \frac{\mathbf{C2.INI.F} \cdot \mathbf{DDCTOT.C}}{\mathbf{C1.INI.F} + \mathbf{C2.INI.F} + \sum \mathbf{DDCFRACC}} \cdot C_{implant}$$

Equation 3-227

$$C_{DDC} = C'_{DDC} + \Delta C_{DDC} \qquad \text{Equation 3-228}$$

where:

$$\Delta C_{DDC} = \frac{\mathbf{DDCFRACC} \cdot \mathbf{DDCTOT.C}}{\mathbf{C1.INI.F} + \mathbf{C2.INI.F} + \sum \mathbf{DDCFRACC}} \cdot C_{implant}$$

Equation 3-229

The net capture of point-defects by clusters is given by:

$$\Delta I = \mathbf{IFRACM} \cdot \Delta C_{cl1} + \sum \frac{\mathbf{DDC.I.N} \cdot \Delta C_{DDC}}{\mathbf{DDC.D.N}} \qquad \text{Equation 3-230}$$

$$\Delta V = \mathbf{VFRACM} \cdot \Delta C_{cl2} + \sum \frac{\mathbf{DDC.V.N} \cdot \Delta C_{DDC}}{\mathbf{DDC.D.N}} \qquad \text{Equation 3-231}$$

Since the formation of dopant-defect clusters is limited by the number of point-defects, Equations 3-225, Equations 3-227 and Equations 3-229 are modified by:

$$\Delta C_{cl1}^{\mathrm{mod}} = \Delta C_{cl1} \cdot min(1.0,\ I/\ \Delta I) \qquad \text{Equation 3-232}$$

$$\Delta C_{cl2}^{\mathrm{mod}} = \Delta C_{cl2} \cdot \; min(1.0,\; V/\;\Delta V)$$

Equation 3-233

$$\Delta C_{DDC}^{\mathrm{mod}} = \Delta C_{DDC} \cdot \; min\left(1.0,\; \frac{\mathtt{DDC.I.N} \cdot \; I/\;\Delta I + \mathtt{DDC.V.N} \cdot \; V/\;\Delta V}{\mathtt{DDC.I.N} + \mathtt{DDC.V.N}}\right)$$

Equation 3-234

Note:

You may not define both non-zero `DDC.I.N` and `DDC.V.N`. But it is allowed to define both zero `DDC.I.N` and `DDC.V.N`, i.e. dopant-only clusters. In such a case, $\Delta C_{DDC}^{\mathrm{mod}} = \Delta C_{DDC}$.

In recrystallization during the anneal process following implant, clusters are initialized in an amorphous region as follows;

$$C_{cl0} = (1 - \mathtt{DDCTOT.A}) \cdot \; C_{cl,\; total}$$

Equation 3-235

$$C_{cl1} = \frac{\mathtt{C1FRAC.A} \cdot \; \mathtt{DDCTOT.A}}{\mathtt{C1FRAC.A} + \mathtt{C2FRAC.A} + \sum \mathtt{DDCFRACA}} \cdot \; C_{cl,\; total}$$

Equation 3-236

$$C_{cl2} = \frac{\mathtt{C2FRAC.A} \cdot \; \mathtt{DDCTOT.A}}{\mathtt{C1FRAC.A} + \mathtt{C2FRAC.A} + \sum \mathtt{DDCFRACA}} \cdot \; C_{cl,\; total}$$

Equation 3-237

$$C_{DDC} = \frac{\mathtt{DDCFRACA} \cdot \; \mathtt{DDCTOT.A}}{\mathtt{C1FRAC.A} + \sum \mathtt{DDCFRACA}} \cdot \; C_{cl,\; total}$$

Equation 3-238

When the boolean parameter **`CL.INI.A`** has been turned on, the total dopant clustering concentration in an amorphous region is initialized as follows:

$$C_{cl,\; total} = max(C - min(\mathtt{ACT.AMOR},\; \mathtt{ACT.NI} \cdot \; n_i),\; 0.0)$$

Equation 3-239

or:

$$C_{cl,\; total} = max(C - \mathtt{ACT.AMOR},\; 0.0) \text{ when } \mathtt{ACT.NI} = 0.0$$

Equation 3-240

However, turning off **`CL.INI.A`** specifies to assume that all dopants are activated, i.e. $C_{cl,\; total} = 0.0$.

`ACT.AMOR` is the maximum active concentration in the recrystallized region of an amorphous region, and **`ACT.NI`** is the multiplication factor of the intrinsic carrier concentration, n_i. **`C1FRAC.A`** denotes the fraction of C_{cl1} to the total cluster concentration, $C_{cl,\; total}$. **`ACT.NI`** is corresponding to **`ACT.MIN`** in the **`ACT.TRAN`** model.

Having the **`DDC.FULL`** model disabled sets all **`DDCFRACC`** to zero, **`DDCTOT.C`** to **`C1.INI.F`**, and **`DDCTOT.A`** to **`C1FRAC.A`**.

In a crystalline region:

$$C_{cl0} = min(C_{cl0\_\mathrm{before\_recrystallization}},\ C_{cl0,\ min}) \qquad \text{Equation 3-241}$$

where $C_{cl0,\ min}$ is calculated by:

$$C_{amin} + \mathtt{DDCS.0} \cdot \ \exp\left(-\frac{\mathtt{DDCS.E}}{kT}\right)\left(C_{amin}\left(\frac{I}{I_i*}\right)^{\mathtt{IFRACS}}\right)^{\mathtt{DDCS.N}}$$

$$= C - w_{gb} - C_{cl0,\ min} - C_{cl1} - C_{cl2} - \sum \mathtt{DDC.N.D} C_{DDCLUST}$$

Equation 3-242

and $C_{amin} = min(C_a*,\ \mathtt{ACT.NI} \cdot \ n_i)$ Equation 3-243

For the deposition of a doped material, all the dopants are assumed to be initially clustered. The precipitation is instantly dissolved at the very beginning of a post-thermal process with Equation 3-241.

Segregation of Impurities

The segregation of impurities at material interfaces is treated as a nonequilibrium process by TSUPREM-4.

Segregation Flux

At an interface between materials i and j, the impurity flux J from material i to material j (normal to the interface) is given by:

$$J = h\left(\frac{C_i}{m}\eta^{q_{ij}} - C_j\eta^{q_{ji}}\right) \qquad \text{Equation 3-244}$$

where:

- C_i and C_j are the concentrations in materials i and j, respectively.
- q_{ij} and q_{ji} are the charges
- h is the interface transport coefficient.
- m is the intrinsic equilibrium interface segregation coefficient.

This expression represents the impurity flux in the diffusion equation at interfaces between different materials and between the ambient gas and the exposed surfaces of the simulation structure.

The segregation is based on the chemical impurity concentrations when the **ACT.EQUI** model is used and the active concentrations when other activation models are used. (When $\mathtt{V.COMPAT} \le 6.4$, the chemical concentrations are used in all cases.) For materials using the polycrystalline model, segregation at material interfaces is based on the concentration in the grain interior.

Transport Coefficient

The transport coefficient is given by:

$$h = \mathbf{TRANS.0} \cdot \exp\left(\frac{-\mathbf{TRANS.E}}{kT}\right) \tag*{Equation 3-245}$$

Segregation Coefficient

The segregation coefficient defines the ratio C_i / C_j in intrinsic equilibrium, when the interface impurity flux vanishes. The segregation coefficient is given by:

$$m = \mathbf{SEG.0} \cdot \exp\left(\frac{-\mathbf{SEG.E}}{kT}\right) \tag*{Equation 3-246}$$

Fermi-level Dependence32bfsw

When the charge state of a dopant in a material is different from that in the neighbor material, the segregation of the charged dopant in the material i (or j) must depend on the Fermi-level for charge conservation. The charges q_{ij} and q_{ji} are defined by the **`SEG.CPOW`** parameter.

The coefficients **`TRANS.0`**, **`TRANS.E`**, **`SEG.0`**, **`SEG.E`**, and **`SEG.CPOW`** for each impurity and pair of materials are defined on the coefficient statements for impurities (see Chapter 4, IMPURITY, p. 4-273, Chapter 4, ANTIMONY, p. 4-337, Chapter 4, ARSENIC, p. 4-344, Chapter 4, BORON, p. 4-351, and Chapter 4, PHOSPHORUS, p. 4-358). For an interface between materials i and j, material i is specified by the **`MATERIAL`** parameter and material j is specified with the **`/MATERIA`** parameter.

Moving-Boundary Flux

There is an additional flux at oxidizing interfaces due to the consumption of silicon containing impurities. The flux is from the silicon into the oxide and is of magnitude vC_{Si}, where C_{Si} is the impurity concentration on the silicon side of the interface and v is the velocity of the interface.

Interface Trap Model

Assuming that there are trap sites at the interface between two adjacent materials, the dopant diffusing through the interface can be trapped into the trap site Reference [11]. The model is activated by specifying the **`ITRAP`** parameter on the **`METHOD`** statement.

$$\frac{\partial\sigma}{\partial t} = -\frac{\partial}{\partial l}\left(D\frac{\partial\sigma}{\partial l}\right) + F_i + F_j \tag*{Equation 3-247}$$

where σ is the areal density of occupied trap sites and l is the length along the boundary and, F_i and F_j are the dopant flux to the interface trap sites from materials i and j, respectively.

$$F_i = h_i\left(C_i\left(f\eta^{q_{ij}} + r_i\frac{\sigma}{\sigma_{max}}\right) - \kappa_i\sigma\right) \tag*{Equation 3-248}$$

$$F_j = h_j\left(C_j\left(f\eta^{q_{ji}} + r_j\frac{\sigma}{\sigma_{max}}\right) - \kappa_j\sigma\right)$$ Equation 3-249

where C_i and C_j are the concentrations in materials i and j, respectively. The active concentration is used for the surface concentrations of silicon and polysilicon materials unless the **IT.ACT** parameter on the **METHOD** statement is turned *off*. If the transient clustering model is turned *on*, the active concentration is used regardless of the specification of the **IT.ACT** parameter. The interface trap exists on the boundary of either material i or material j. The material that contains the traps is specified by the **MATERIAL** parameter, while the adjacent material is specified by the **/MATERIA** parameter on the **IMPURITY** statement. f is the fraction of unfilled trap sites. If the **IT.CPL** parameter on the **METHOD** statement is specified:

$$f = 1 - \sum\frac{\sigma}{\sigma_{max}}$$ Equation 3-250

where the sum is taken over all the trapped dopant species present in the interface. Otherwise, as default, f is given by:

$$f = 1 - \frac{\sigma}{\sigma_{max}}$$ Equation 3-251

D is the diffusivity of trapped dopant moving along the interface, and σ_{max} denotes the maximum trap density dependent on the property of the interface for each dopant species.

$$\sigma_{max} = \mathbf{Q.MAX.0} \cdot \exp\left(\frac{-\mathbf{Q.MAX.E}}{kT}\right)$$ Equation 3-252

$$D = \mathbf{DIX.0} \cdot \exp\left(\frac{-\mathbf{DIX.E}}{kT}\right)$$ Equation 3-253

When the interface is formed by deposition, epitaxy or oxidation, the initial density, σ_{ini}, of trapped dopant is set as one of following:

$$\sigma_{ini} = 0$$ Equation 3-254

$$\sigma_{ini} = \mathbf{Q.INI.0} \cdot \exp\left(\frac{-\mathbf{Q.INI.E}}{kT}\right)$$ Equation 3-255

$$\sigma_{ini} = \frac{\sigma_{max}f(h_iC_i + h_jC_j)}{h_iC_i(1-r_i) + h_jC_j(1-r_j) + \sigma_{max}(h_i\kappa_i + h_j\kappa_j)}$$ Equation 3-256

The initial density of trapped dopant is set by specifying one of parameters, **IT.ZERO** (Equation 3-254), **IT.THERM** (Equation 3-255), or **IT.STEAD** (Equation 3-256) on the **METHOD** statement. Equation 3-256 satisfies the steady state, F_i+F_j=0.

Note:

If the impurity is first introduced by implantation, the initial value of the occupied trap density of the impurity is set to zero regardless of the above specification.

Assuming that the interface trap exists on the boundary of material *i*, hereinafter, the transport coefficients h_i and h_j are given by:

$$h_i = \mathtt{TRANS.0} \cdot \exp\left(\frac{-\mathtt{TRANS.E}}{kT}\right) \quad \text{Equation 3-257}$$

$$h_j = \mathtt{/TRANS.0} \cdot \exp\left(\frac{-\mathtt{/TRANS.E}}{kT}\right) \quad \text{Equation 3-258}$$

The r_i and r_j denote the ratio of detrapping rate to trapping rate at the interface with materials *i* and *j*, respectively, which are given by:

$$r_i = \mathtt{RATIO.0} \cdot \exp\left(\frac{-\mathtt{RATIO.E}}{kT}\right) \quad \text{Equation 3-259}$$

$$r_j = \mathtt{/RATIO.0} \cdot \exp\left(\frac{-\mathtt{/RATIO.E}}{kT}\right) \quad \text{Equation 3-260}$$

The detrapping of trapped dopant from an interface is determined by how many more dopants can be accepted into the material, as well as how many trapped dopants exist. For concentrations over critical concentration, no more detrapping occurs. **`SEG.SS`** and **`/SEG.SS`** specify that the critical concentrations are the same as the solid solubilities of dopant in materials i and j, respectively. **`SEG.SS`** and **`/SEG.SS`** may be applied only to the silicon or polysilicon material. The κ_i and κ_j are then given by:

$$\kappa_i = r_i\frac{C_{ss,i}}{\sigma_{max}}, \qquad \kappa_j = r_j\frac{C_{ss,j}}{\sigma_{max}} \quad \text{Equation 3-261}$$

Note:

For a dopant for which solid solubility is not known, solid solubility is calculated from the clustering model by setting the total concentration to the concentration of atoms in the material.

Also, the κ_i and κ_j can be explicitly given by:

$$\kappa_i = \mathtt{SEG.0} \cdot \exp\left(\frac{-\mathtt{SEG.E}}{kT}\right) \quad \text{Equation 3-262}$$

$$\kappa_j = \mathtt{/SEG.0} \cdot \exp\left(\frac{-\mathtt{/SEG.E}}{kT}\right) \quad \text{Equation 3-263}$$

The specification of either `SEG.EQ2` or `/SEG.EQ2` makes the segregation in equilibrium equal to the two-phase segregation in equilibrium. `SEG.EQ2` specifies that the κ_i is calculated with κ_j and the segregation in the two-phase system instead of with Equation 3-262. For `/SEG.EQ2`, κ_j is calculated in the same manner.

The interface trap model is applied only when the impurity and adjacent materials are well defined. Otherwise, the normal two-phase segregation model is used. When the interface trap model is applied, the two-phase segregation flux is excluded unless the `TWO.PHAS` parameter is specified.

The parameters, `TRANS.0`, `TRANS.E`, `TRANS.0`, `TRANS.E`, `RATIO.0`, `RATIO.E`, `/RATIO.0`, `/RATIO.E`, `SEG.0`, `SEG.E`, `/SEG.0`, `/SEG.E`, `SEG.SEG.SS`, `SEG.EQ2`, `/SEG.EQ2`, `Q.MAX.0`, `Q.MAX.E`, `Q.INI.0`, `Q.INI.E`, `DIX.0`, `DIX.E`, and `TWO.PHAS` are specified on the `IMPURITY` statement.

Using the Interface Trap Model

The trapped component is stored as a separate impurity. The name of the trapped impurity is created from the name of the base impurity by prepending `I_` to the name. Thus, you set the diffusivities of the trapped component of boron with

```
IMPURITY IMP=I_BORON MAT=OXIDE /MAT=SILICON DIX.0=...
```

The interface trap model for the specified trapped impurity works only at the interface between the materials specified with the `MATERIAL` and `/MATERIA` parameters. *Note* that the trapped impurity exists on the surface node of the material specified by `MATERIAL`. Thus, you get the occupied trap density with

```
SELECT Z=I_BORON
EXTRACT OXIDE /SILICON X=0
```

Dose Loss Model

There are several physical mechanisms that lead to the impurity dose loss. Typically, interface trapping and segregation at the interface cause impurity outflux from silicon to the adjacent layers. When the adjacent layers are removed later in the process flow, like a sacrificial oxide, the impurity trapped there is also removed. This means dose loss for the impurity that remains inside silicon wafer.

This section describes dose loss mechanism due to the impurity sweeping by the advancing epitaxial recrystallization front when an amorphized silicon layer is annealed. This effect has been observed for indium, where up to 80% of the dose can disappear during the epitaxial recrystallization of the amorphized layer. For arsenic, similarly fast dose loss has been reported, although not as dramatic as for indium.

A simple model can describe this effect. During epitaxial recrystallization of the amorphized layer, all indium above certain threshold concentration is swept by the advancing recrystallization front. This applies to all dopants that happen to be inside the amorphous layer, regardless of which ion was implanted.

Threshold concentration of 6e18 cm^{-3} was identified for indium for this purpose. The threshold concentration can be specified by parameter **MAX.AMOR** on the **IMPURITY** command. The values of **MAX.AMOR** parameter for the other dopants is set to a very high value of 5e22 cm^{-3}.

Diffusion of Point Defects

This section describes the equations that model the diffusion of interstitials and vacancies in silicon. The modeling of point defects depends on the **NSTREAMS**, **PD.PFLUX**, **PD.PTIME**, and **PD.PREC** parameters on the **METHOD** statement (see p. 3-18). For **NSTREAMS** =1 (e.g., if **PD.FERMI** is set on the **METHOD** statement), the interstitial and vacancy concentrations depend only on the Fermi level:

$$I = I^* \quad \text{and} \quad V = V^* \qquad \text{Equation 3-264}$$

The point defect enhancement factors M/M' and N/N' in the equation for diffusion of impurities are unity (see Diffusion of Impurities, p. 3-18), and there is no enhancement (or retardation) of impurity diffusion due to oxidation.

Equilibrium Concentrations

The equilibrium concentrations I^* and V^* are given by:

$$I^* = I_i^* \sum_{k=-2}^{2}\left(\phi_{Ik}\eta^{-k}\right) \qquad V^* = V_i^* \sum_{k=-2}^{2}\left(\phi_{Vk}\eta^{-k}\right) \qquad \text{Equation 3-265}$$

$$I_i^* = F_{Ists}\cdot \mathbf{CEQUIL.F}\cdot I_{i0}^* \qquad \text{Equation 3-266}$$

$$V_i^* = F_{Vsts}\cdot \mathbf{CEQUIL.F}\cdot V_{i0}^* \qquad \text{Equation 3-267}$$

$$I_{i0}^* \text{ and } V_{i0}^* = \mathbf{CEQUIL.0}\cdot \exp\left(\frac{-\mathbf{CEQUIL.E}}{kT}\right) \qquad \text{Equation 3-268}$$

where **CEQUIL.0** and **CEQUIL.E** are parameters on the **INTERSTITIAL** and **VACANCY** statements.

Pressure Dependence

The factor F_{Ists} and F_{Vsts} is applied only in silicon materials for which the stress history is modeled, and the **ST.PDEQ** model is enabled in the **METHOD** statement. The activating volume, **VCEQUIL** has a dimension of A^3 and is specified on the **INTERSTI** and **VACANCY** statements.

$$F_{Ists} \text{ and } F_{Vsts} = \exp\left(\frac{-p\cdot \mathbf{VCEQUIL}}{kT}\right) \qquad \text{Equation 3-269}$$

User-Defined Isotropic Multiplication

The point-defect concentration in intrinsic equilibrium can be multiplied by the parameter **`CEQUIL.F`** in the **`INTERSTI`** and **`VACANCY`** statements. Parameters values can be numbers or formulas. This allows you to define more complicated equilibrium models for point-defects.

Charge State Fractions

The charge state fractions ϕ_{Ik} and ϕ_{Vk} are given by:

$$\phi_{Ik} = \frac{\phi'_{Ik}}{\sum_{k=-6}^{6} \phi'_{Ik}} \quad \text{and} \quad \phi_{Vk} = \frac{\phi'_{Vk}}{\sum_{k=-6}^{6} \phi'_{Vk}} \qquad \text{Equation 3-270}$$

$$\phi'_{I(-2)} \text{ and } \phi'_{V(-2)} = \mathbf{DNEG.0} \cdot \exp\left(\frac{-\,\mathbf{DNEG.E}}{kT}\right) \qquad \text{Equation 3-271}$$

$$\phi'_{I(-1)} \text{ and } \phi'_{V(-1)} = \mathbf{NEG.0} \cdot \exp\left(\frac{-\,\mathbf{NEG.E}}{kT}\right) \qquad \text{Equation 3-272}$$

$$\phi'_{I0} \text{ and } \phi'_{V0} = \mathbf{NEU.0} \cdot \exp\left(\frac{-\,\mathbf{NEU.E}}{kT}\right) \qquad \text{Equation 3-273}$$

$$\phi'_{I1} \text{ and } \phi'_{V1} = \mathbf{POS.0} \cdot \exp\left(\frac{-\,\mathbf{POS.E}}{kT}\right) \qquad \text{Equation 3-274}$$

$$\phi'_{I2} \text{ and } \phi'_{V2} = \mathbf{DPOS.0} \cdot \exp\left(\frac{-\,\mathbf{DPOS.E}}{kT}\right) \qquad \text{Equation 3-275}$$

$$\phi'_{Ik} \text{ and } \phi'_{Vk} = \mathbf{FRAC.0} \cdot \exp\left(\frac{-\,\mathbf{FRAC.E}}{kT}\right) \qquad \text{Equation 3-276}$$

The last form gives the fractions for arbitrary charge states $-6 \le k \le 6$. The parameters for interstitials are specified on the **`INTERSTITIAL`** statement (see Chapter 4, INTERSTITIAL, p. 4-311), while the parameters for vacancies are specified on the **`VACANCY`** statement (see Chapter 4, VACANCY, p. 4-327); the charge state k is specified by the **`C.STATE`** parameter. *Note* that the defect concentrations are not actually calculated when **`NSTREAMS`**=1; the values I^* and V^* are used when the interstitial and vacancy concentrations are printed or plotted.

Point Defect Diffusion Equations

For **NSTREAMS** =3 (e.g., if **PD.TRANS** or **PD.FULL** is set on the **METHOD** statement), the equations for point defect diffusion are solved:

$$\frac{\partial(I+I_c+I_{dds})}{\partial t}+\sum\frac{\partial M}{\partial t}=-\vec{\nabla}\cdot\left[-D_I I^*\left(\vec{\nabla}\left(\frac{I}{I^*}\right)+\left(\frac{I}{I^*}\right)\vec{\nabla}\ln I_i^*\right)+\sum\vec{J}_m\right]$$
$$+G_p-R_b-R_t-R_{sIcI}-R_{lIcI}-R_l-R_{dd}$$
$$-\sum R_{DDCM}-\sum R_{DDC,\ C+I}-\sum R_{DDC,\ I+M}$$

Equation 3-277

$$\frac{\partial(V+V_c)}{\partial t}+\sum\frac{\partial N}{\partial t}=-\vec{\nabla}\cdot\left[-D_V V^*\left(\vec{\nabla}\left(\frac{V}{V^*}\right)+\left(\frac{V}{V^*}\right)\vec{\nabla}\ln V_i^*\right)+\sum\vec{J}_n\right]$$
$$+G_p-R_b-R_{sIcV}-R_{lIcV}-R_{dv}$$
$$-\sum R_{DDCN}-\sum R_{DDC,\ C+V}-\sum R_{DDC,\ V+N}$$

Equation 3-278

$$G_p=\sum\left[\frac{K_{mv}\frac{V}{V*}}{K_m+K_{mv}\frac{V}{V*}}\left(\frac{\partial M}{\partial t}+\vec{\nabla}\cdot\vec{J}_m+R_{DDCM}\right)\right]$$
$$+\sum\left[\frac{K_{ni}\frac{I}{I*}}{K_n+K_{ni}\frac{I}{I*}}\left(\frac{\partial N}{\partial t}+\vec{\nabla}\cdot\vec{J}_n+R_{DDCN}\right)\right]$$

Equation 3-279

where:

- M and N are the concentrations of dopant-interstitial and dopant-vacancy pairs defined in Diffusion of Impurities, p. 3-18.
- $\vec{J}_m$ and $\vec{J}_n$ are the corresponding fluxes.
- I_c and V_c are the concentrations of interstitials and vacancies in small clusters (see Small Clusters of Point Defects, p. 3-78).
- I_{dds} is the concentration of interstitials in small dopant-defect clusters (see Activation of Impurities, p. 3-34).
- R_{dd}, R_b, R_t, R_{sIcI}, R_{lIcI} and R_l model the loss of free interstitials to dopant-defect clusters (Dopant-Interstitial Clustering Model, p. 3-38), bulk recombination (Recombination of Interstitials with Vacancies, p. 3-64), interstitial traps (Interstitial Traps, p. 3-70), small interstitial clusters (Transient Small Clustering, p. 3-80), large interstitial clusters i.e., {311} defects (Interstitial Clustering Models, p. 3-71), dislocation loops

(Dislocation Loop Model, p. 3-82), respectively. The R_{sIcV} and R_{lIcV} term model the loss of free vacancies to the small interstitial clusters(Transient Small Clustering, p. 3-80) and {311} defects (Interstitial Clustering Models, p. 3-71).

- $$R_{DDCM} = R_{DDC,\ S+M} + R_{DDC,\ I+M} + R_{DDC,\ I_{n-1}+M} + 2R_{DDC,\ M+M} + R_{DDC,\ C+M} + R_{DDC,\ M+N} - R_{DDC,\ XI_2+V}$$
- $$R_{DDCN} = R_{DDC,\ S+N} + R_{DDC,\ V+N} + R_{DDC,\ I_{n+1}+N} + 2R_{DDC,\ N+N} + R_{DDC,\ C+N} + R_{DDC,\ M+N} - R_{DDC,\ XV_2+I}$$

The sums are taken over all dopant species present in the structure. The M and N terms are included only if **PD.PTIME** is true (set on the **METHOD** statement, directly or by specifying **PD.FULL**); otherwise, they are set to 0. Likewise, the $\vec{J}_m$ and $\vec{J}_n$ terms are included only if **PD.PFLUX** is true (set on the **METHOD** statement, directly or by specifying **PD.FULL**).

For **NSTREAMS** =5 or **PD.5STR**, the equations for point defect diffusion are solved:

$$\frac{\partial(I + I_c + I_{dds})}{\partial t} = -\vec{\nabla} \cdot \left(-D_I I^* \left(\vec{\nabla}\left(\frac{I}{I^*}\right) + \left(\frac{I}{I^*}\right)\vec{\nabla} \ln I_i^*\right)\right)$$

$$-(G_m - R_m) - (G_{ni} - R_{ni})$$

$$-R_b - R_t - R_{sIcI} - R_{lIcI} - R_l - R_{dd} - \sum R_{DDC,\ C+I} - \sum R_{DDC,\ I+M}$$

Equation 3-280

$$\frac{\partial(V + V_c)}{\partial t} = -\vec{\nabla} \cdot \left(-D_V V^* \left(\vec{\nabla}\left(\frac{V}{V^*}\right) + \left(\frac{V}{V^*}\right)\vec{\nabla} \ln V_i^*\right)\right)$$

Equation 3-281

$$-(G_n - R_n) - (G_{mv} - R_{mv})$$

$$-R_b - R_{sIcV} - R_{lIcV} - \sum R_{DDC,\ C+V} - \sum R_{DDC,\ V+N}$$

where $R_{DDC,\ I+M}$ and $R_{DDC,\ C+I}$ model the loss of free interstitials to dopant-defect clusters by reacting with a dopant-interstitial pair and another dopant-defect cluster respectively(Full Dynamics of Dopant-Defect Clustering, p. 3-40).

Interstitial and Vacancy Diffusivities

D_I and D_V are the diffusivities of interstitials and vacancies, respectively, given by:

$$D_I = F_{DIsts} \cdot \ \mathtt{D.FACTOR} \cdot \begin{bmatrix} \mathtt{D.F11} & 0 \\ 0 & \mathtt{D.F22} \end{bmatrix} \cdot \frac{\sum_k D_{Ik} \phi_{Ik} \eta^{-k}}{\sum_k \phi_{Ik} \eta^{-k}}$$

Equation 3-282

$$D_V = F_{DVsts} \cdot \ \mathtt{D.FACTOR} \cdot \begin{bmatrix} \mathtt{D.F11} & 0 \\ 0 & \mathtt{D.F22} \end{bmatrix} \cdot \frac{\sum_k D_{Vk} \phi_{Vk} \eta^{-k}}{\sum_k \phi_{Vk} \eta^{-k}}$$

Equation 3-283

where D_{Ik} and D_{Vk} are specified by:

$$D_{Ik} \text{ and } D_{Vk} = \mathtt{D.0} \cdot \ \exp\left(\frac{-\mathtt{D.E}}{kT}\right) \times \mathtt{DC.0} \cdot \ \exp\left(\frac{-\mathtt{DC.E}}{kT}\right)$$

Equation 3-284

where **`DC.0`** and **`DC.E`** are specified for **`C.STATE`** $= k$. **`D.0`**, **`D.E`**, **`DC.0`**, **`DC.E`**, and **`C.STATE`** for D_I and D_V are specified on the **`INTERSTITIAL`** (Chapter 4, INTERSTITIAL, p. 4-311) and **`VACANCY`** (Chapter 4, VACANCY, p. 4-327) statements, respectively. **`DC.0`** and **`DC.E`** can also be set for all charge states by specifying **`C.ALL`**.

Two approaches can be used to specify the diffusivity of point defects:

1. **`D.0`** (in cm^2/sec or microns^2/min) and **`D.E`** can be used to specify the basic diffusivity while **`DC.0`** (unitless) and **`DC.E`** specify relative diffusivities of the various charge states
2. **`DC.0`** (in cm^2/sec or microns^2/min) and **`DC.E`** can be used to specify the diffusivities of the various charge states while **`DC.0`** (unitless) and **`DC.E`** specify a scale factor for adjusting the total diffusivity.

Note:

While theoretical calculations suggest that the point defect diffusivities may depend on the Fermi level, this dependence has not been experimentally characterized. By default, all of the `DC.0` are set to 1.0 and all of the `DC.E` are set to zero, so the diffusivity (determined by the values of `D.0` and `D.E` using approach 1 above) is independent of the Fermi level.

Pressure Dependence

The factor F_{DIsts} and F_{DVsts} is applied only for diffusion in silicon materials for which the stress history is modeled, and the **`ST.DIFF`** model is enabled. The activating volume, **`VD`** has a dimension of A^3 and is specified on the **`INTERSTI`** and **VACANCY** statements. The parameters **ST.HIST** and **ST.DIFF** for setting stress history and stress dependent diffusion are specified on the **METHOD** statement.

$$F_{DIsts} \text{ and } F_{DVsts} = \exp\left(\frac{-p \cdot \ \mathtt{VD}}{kT}\right) \qquad \text{Equation 3-285}$$

User-Defined Isotropic Multiplication

The diffusivities can be multiplied by the parameter **D.FACTOR** in the **INTERSTI** and **VACANCY** statements. Parameters values can be numbers or formulas. This allows you to define more complicated point-defect diffusivity models.

User-Defined Anisotropic Multiplication

To provide the modeling capability for an anisotropic point-defect diffusion, the user-specified tensor expression can be taken. It describes the anisotropic diffusion path, which is given by the parameters **D.F11** and **D.F22** in the **INTERSTI** and **VACANCY** statement. Each parameter can be given by any formula expression.

Reaction of Pairs with Point Defects

In the **PD.FULL** model, the G_p terms are the result of dopant-defect pairs reacting with dopants. Normally, when dopant-interstitial pairs break up, they produce an interstitial, which is accounted for by the M and $\vec{J}_m$ terms in Equation 3-277. However, when dopant-assisted recombination is dominant, dopant-interstitial pairs are destroyed by absorbing a vacancy rather than producing an interstitial. Under these circumstances, the G_p term cancels the M and $\vec{J}_m$ terms in Equation 3-277 and subtracts them from Equation 3-278 instead. Similarly, dopant-vacancy pairs can either produce a vacancy or absorb an interstitial when they dissolve.

Recombination of Interstitials with Vacancies

The net recombination rate of interstitials with vacancies in the bulk silicon is given by:

$$R_b = K_b(F_{IV} + F_{cV} + F_{Ic})\left(\frac{I}{I^*}\frac{V}{V^*} - 1\right) \quad + R_{prec} \qquad \text{Equation 3-286}$$

$$R_{prec} = \sum \left[\left(\frac{K_m K_{mv}}{K_m + K_{mv}\frac{V}{V^*}} + \frac{K_n K_{ni}}{K_n + K_{ni}\frac{I}{I^*}}\right) S\right]\left(\frac{I}{I^*}\frac{V}{V^*} - 1\right)$$

Equation 3-287

for true **PD.PREC** in the **PD.FULL** model, otherwise $R_{prec} = 0$

where:

- $S = C - M - N$ is the substitutional dopant concentration.
- K_m, K_n, K_{mv} and K_{ni} are described in Diffusion of Impurities, p. 3-18.

- F_{Ic} and F_{cV} are described in Small Clusters of Point Defects, p. 3-78.

Again, the sums are taken over all dopant species. The K_{mv} and K_{ni} terms model dopant-assisted recombination; they are included only if **PD.PREC** is true (set on the **METHOD** statement, directly or by specifying **PD.FULL**); otherwise, they are set to zero.

The bulk recombination factor K_b is specified by:

$$K_b = \mathtt{KB.0} \cdot \exp\left(\frac{-\mathtt{KB.E}}{kT}\right) \tag{Equation 3-288}$$

where **KB.0** and **KB.E** are parameters on the **INTERSTITIAL** statement.

The factor F_{IV} describes the dependence of the recombination rate on the charge states of the point defects:

$$F_{IV} = I_i^* V_i^* \sum_{j,\ k}\left(K_{IV,\ jk}\phi_{Ij}\eta^{-j}\phi_{Vk}\eta^{-k}\right) / \ \alpha \tag{Equation 3-289}$$

$$K_{IV,\ jk} = \mathtt{KIV.0} \cdot \exp\left(\frac{-\mathtt{KIV.E}}{kT}\right) \tag{Equation 3-290}$$

where **KIV.0** and **KIV.E** are specified on the **INTERSTITIAL** statement with $\mathtt{C.I} = j$ and $\mathtt{C.V} = k$. The normalizing factor α depends on whether **KIV.NORM** (set on the **INTERSTITIAL** statement) is true or false:

$$\alpha = \begin{cases} \sum_{j,\ k}\left(K_{IV,\ jk}\phi_{Ij}\phi_{Vk}\right) & \mathtt{KIV.NORM}\ \text{true} \\ 1 & \mathtt{KIV.NORM}\ \text{false} \end{cases} \tag{Equation 3-291}$$

When **KIV.NORM** is true, $F_{IV} = I_i^* V_i^*$ under intrinsic doping conditions.

The values of $K_{IV,\ jk}$ are unknown, but TSUPREM-4 provides parameters on the **INTERSTITIAL** statement for setting them based on plausible assumptions:

- **KB.HIGH** sets $K_{IV,\ jk} = 1$ for all j and k; this assumes that the rate of recombination between interstitials and vacancies is independent of their charge states; $\alpha = 1$ regardless of the setting of **KIV.NORM**.
- **KB.MED** sets $K_{IV,\ jk} = 1$ for $j = -k$ or $jk = 0$ and $K_{IV,\ jk} = 0$ otherwise; this assumes that a point defect can recombine only with a neutral or oppositely charged defect.
- **KB.LOW** sets $K_{IV,\ jk} = 1$ only for $j = -k$ and $K_{IV,\ jk} = 0$ otherwise; this assumes that a point defect can recombine only with an oppositely charged defect.

The three bulk recombination models reflect differing assumptions about the reaction rates between interstitials and vacancies in various charge states. The **KB.LOW** model assumes that charged interstitials recombine only with oppositely charged vacancies. This model, with **KIV.NORM** set, is the default; it is equivalent to the model used in older versions of TSUPREM-4. The **KB.HIGH** model assumes that any interstitial is equally likely to recombine with any vacancy, regardless of their charge states. This is the model used in many other process simulators, giving:

$$R_b = K_b(IV - I^* V^*)$$ Equation 3-292

The **KB.MED** model assumes that reactions involving an uncharged point defect and reactions involving oppositely charged point defects are equally likely, while reactions between point defects of like charge do not occur (because of Coulomb repulsion).

It is expected that for a given value of K_b the **KB.LOW** model underestimates the recombination rate at high doping levels because it neglects the recombination of neutral defects with charged defects. Similarly, the **KB.HIGH** model overestimates the recombination because it includes recombination of similarly charged defects. The **KB.MED** should work the best, although it ignores reactions between defects having opposite charge of different magnitudes (e.g., doubly negative vacancies and singly positive interstitials). *Note* that the values of K_b and (perhaps) other parameters may need to be recalibrated if the model is changed.

Note:

> ***Versions of the program prior to 1999.2 did not use the* `KIV.0`*,* `KIV.E`*, or* `KIV.NORM` *parameters, but some of them did use* `KB.LOW`*,* `KB.MED`*, and* `KB.HIGH`*. If compatibility with those versions is required, use* `KIV.NORM` *when* `KB.LOW` *is used and* `^KIV.NORM` *when* `KB.MED` *is used, or use* `V.COMPAT=1998.4` *on the* `OPTION` *statement.***

Absorption by Traps, Clusters, and Dislocation Loops

R_t is the rate of absorption of interstitials at stationary interstitial trapping sites; R_t is given by Equation 3-313 in Interstitial Traps, p. 3-70. R_c is the rate of absorption by interstitial clusters (see Interstitial Clustering Models, p. 3-71). R_l is the rate of absorption of interstitials by dislocation loops (see Dislocation Loop Model, p. 3-82).

Injection and Recombination of Point Defects at Interfaces

Recombination of interstitials and vacancies at interfaces with other materials is modeled as Reference [12].

$$R_s = K_s(C - C^*)$$ Equation 3-293

where:

- C is the local concentration of interstitials or vacancies.
- C^* is the equilibrium concentration of interstitials or vacancies.
- K_s is the surface recombination rate.

Surface Recombination Velocity Models

There are three models for the surface recombination velocity, specified by the **V.MAXOX**, **V.INITOX**, and **V.NORM** parameters on the **INTERSTITIAL** and **VACANCY** statements. In each case, the surface recombination rate depends on the motion of the interface due to oxidation:

$$K_s = K_{surf} + (K_{svel} - K_{surf})\left(\frac{v}{v_{max}}\right)^{K_{pow}} \qquad (\mathtt{V.MAXOX}\text{ model})$$

Equation 3-294

$$K_s = K_{surf}\left[1 + K_{srat}\left(\frac{v}{v_{init}}\right)^{K_{pow}}\right] \qquad (\mathtt{V.INITOX}\text{ model})$$

Equation 3-295

$$K_s = K_{surf} + K_{svel}\left(\frac{v}{v_{norm}}\right)^{K_{pow}} \qquad (\mathtt{V.NORM}\text{ model})$$

Equation 3-296

where v is the local velocity of the interface and K_{surf}, K_{svel} and K_{srat} are specified by:

$$K_{surf} = \mathtt{KSURF.0} \cdot \exp\left(\frac{-\mathtt{KSURF.E}}{kT}\right)$$

Equation 3-297

$$K_{svel} = \mathtt{KSVEL.0} \cdot \exp\left(\frac{-\mathtt{KSVEL.E}}{kT}\right)$$

Equation 3-298

$$K_{srat} = \mathtt{KSRAT.0} \cdot \exp\left(\frac{-\mathtt{KSRAT.E}}{kT}\right)$$

Equation 3-299

$$K_{pow} = \mathtt{KPOW.0} \cdot \exp\left(\frac{-\mathtt{KPOW.E}}{kT}\right)$$

Equation 3-300

where **KSURF.0**, **KSURF.E**, **KSVEL.0**, **KSVEL.E**, **KSRAT.0**, **KSRAT.E**, **KPOW.0**, and **KPOW.E** are parameters on the **INTERSTITIAL** and **VACANCY** statements.

The three models differ primarily in the normalizing factor for the interface velocity. This section describes the differences between the models, and their advantages and disadvantages.

V.MAXOX Model

The **V.MAXOX** model (the model used in releases of TSUPREM-4 prior to version 6.0) uses v_{max}, the maximum *y* component of interface velocity in the structure. The disadvantage of this model is that the normalization factor varies with time and oxidation conditions so that the peak recombination velocity does not depend on the oxidation rate.

V.INITOX Model

The **V.INITOX** model (the model used in *SUPREM-IV* from Stanford University and the University of Florida) corrects the time dependence by using a normalization factor:

$$v_{init} = \frac{B}{A} + r_{thin} \qquad \text{Equation 3-301}$$

where $\frac{B}{A}$ and r_{thin} are the linear and thin regime oxidation rates defined in Oxidation, p. 3-84. Thus v_{init} is the initial oxidation rate of a bare silicon surface. This normalizing factor gives a good time dependence, but the initial surface recombination velocity is still independent of the oxidation conditions such as wet or dry ambient, pressure, or presence of HCl.

V.NORM Model

The **V.NORM** model provides both the time dependence and the dependence on the oxidation conditions by using a constant normalizing factor specified by you:

$$v_{norm} = \mathbf{VNORM.0} \cdot \exp\left(\frac{-\mathbf{VNORM.E}}{kT}\right) \qquad \text{Equation 3-302}$$

where **VNORM.0** and **VNORM.E** are parameters on the **INTERSTITIAL** and **VACANCY** statements.

Trapped Nitrogen Dependent Surface Recombination

The point defect recombination at interfaces depends on trapped nitrogen atoms. Since the recombination rate at interfaces is proportional to the number of kink sites which is supposed to be reduced by trapping nitrogen, the recombination of interstitials and vacancies at interfaces is given by:

$$R = K_s\left(1 - \frac{\sigma_N}{\mathbf{N.MAX.0}\exp\left(-\frac{\mathbf{N.MAX.E}}{kT}\right)}\right)(C - C^*) \qquad \text{Equation 3-303}$$

where **N.MAX.0** and **N.MAX.E** are parameters on the **INTERSTITIAL** and **VACANCY** statements. The model is taken into account by specifying the **PD.NREC** parameter on the **METHOD** statement.

Injection Rate

At moving interfaces there can be injection of interstitials and/or vacancies into the silicon. The injection rate is given either as a function of the interface velocity or by an analytical function of time, depending on whether the **GROWTH** parameter has been specified on the **INTERSTITIAL** or **VACANCY** statement. The injection rate also depends on which of the injection/recombination models (**V.MAXOX**, **V.INITOX**, or **V.NORM**) is specified.

When **GROWTH** is true, the injection G_s is calculated based on the local interface velocity:

$$G_s = \mathtt{VMOLE} \cdot \theta v^{K_{pow}} \qquad (\mathtt{V.MAXOX}\ \text{model}) \qquad \text{Equation 3-304}$$

$$G_s = \mathtt{VMOLE} \cdot \theta v \left(\frac{v}{v_{init}}\right)^{G_{pow}} \qquad (\mathtt{V.INITOX}\ \text{model}) \qquad \text{Equation 3-305}$$

$$G_s = \mathtt{VMOLE} \cdot \theta v \left(\frac{v}{v_{norm}}\right)^{G_{pow}} \qquad (\mathtt{V.NORM}\ \text{model}) \qquad \text{Equation 3-306}$$

where:

- **VMOLE** is the number of silicon atoms per cubic centimeter.
- θ is the fraction of silicon atoms injected.
- v is the local interface velocity.
- K_{pow}, v_{init}, and v_{norm} are given by Equations 3-300, 3-301, and 3-302, respectively.

The values of θ and G_{pow} are specified as:

$$\theta = \mathtt{THETA.0} \cdot \exp\left(\frac{-\mathtt{THETA.E}}{kT}\right) \qquad \text{Equation 3-307}$$

$$G_{pow} = \mathtt{GPOW.0} \cdot \exp\left(\frac{-\mathtt{GPOW.E}}{kT}\right) \qquad \text{Equation 3-308}$$

where **THETA.0**, **THETA.E**, **GPOW.0**, and **GPOW.E** are parameters on the **INTERSTITIAL** and **VACANCY** statements.

If **GROWTH** is false, the analytical model is used:

$$G_s = \frac{A}{(T_0 + t)^{K_{pow}}} \qquad (\mathtt{V.MAXOX}\ \text{model}) \qquad \text{Equation 3-309}$$

$$(G_s = A(T_0 + t)^{G_{pow}}) \qquad (\mathtt{V.INITOX}\ \text{and}\ \mathtt{V.NORM}\ \text{models}) \qquad \text{Equation 3-310}$$

where t is the time into the oxidation step and A and T_0 are given by:

$$A = \mathtt{A.0} \cdot \exp\left(\frac{-\mathtt{A.E}}{kT}\right) \qquad \text{Equation 3-311}$$

$$T_0 = \mathbf{T0.0} \cdot \exp\left(\frac{-\mathbf{T0.E}}{kT}\right)$$ Equation 3-312

where **`A.0`**, **`A.E`**, **`T0.0`**, and **`T0.E`** are parameters on the **`INTERSTITIAL`** and **`VACANCY`** statements. *Note* that this model must be calibrated for a particular starting structure and growth conditions.

The **`GROWTH`** model is normally used for injection of interstitials at an oxide/silicon interface. The analytical model is used with **`A.0`** = 0 to disable injection of interstitials at other interfaces and injection of vacancies at all interfaces.

Moving-Boundary Flux

There is an additional flux at oxidizing interfaces due to the consumption of silicon containing point defects. The flux is directed out of the silicon and is of magnitude vC_{Si}, where C_{Si} is the interstitial or vacancy concentration in the silicon and v is the velocity of the interface. Point defects are not modeled in materials other than silicon, so point defects removed in this fashion have no further effect on the simulation.

Interstitial Traps

The rate of absorption of interstitials at stationary trapping sites is given by References [13] and [14]:

$$R_t = \frac{\partial T_F}{\partial t} = k_f T_E I - k_r T_F$$ Equation 3-313

where:

- T_F is the concentration of filled interstitial traps.
- T_E is the concentration of empty traps.
- k_f and k_r are the forward and reverse rates for the trap-filling reaction.

In equilibrium, the forward and reverse reactions proceed at equal rates giving:

$$k_r = \frac{k_f T_E^* I^*}{T_F^*}$$ Equation 3-314

where T_F^* and T_E^* are the equilibrium concentrations of filled and empty traps, respectively. Thus, Equation 3-313 becomes:

$$R_t = \frac{\partial T_F}{\partial t} = k_f\left[T_E I - \frac{T_E^* I^*}{T_F^*} T_F\right]$$ Equation 3-315

The total number of traps is given by:

$$T_T = T_F + T_E = T_F^* + T_E^* = \mathtt{TRAP.CON} \qquad \text{Equation 3-316}$$

where **`TRAP.CON`** is a parameter on the **`INTERSTITIAL`** statement. The reaction rate and equilibrium concentration of empty traps are given by:

$$k_f = \mathtt{K.TRAP.0} \cdot \exp\left(-\frac{\mathtt{K.TRAP.E}}{kT}\right) \qquad \text{Equation 3-317}$$

$$T_E^* = \mathtt{F.TRAP.0} \cdot \exp\left(-\frac{\mathtt{F.TRAP.E}}{kT}\right) \times \mathtt{TRAP.CON} \qquad \text{Equation 3-318}$$

where **`K.TRAP.0`**, **`K.TRAP.E`**, **`F.TRAP.0`**, and **`F.TRAP.E`** are parameters on the **`INTERSTITIAL`** statement.

Enabling, Disabling, and Initialization

The interstitial trap model is enabled by setting **`TRAP.CON`** to a nonzero value and disabled by setting **`TRAP.CON`** to zero. The empty trap concentration T_E is initialized to its equilibrium value T_E^* at the start of the first diffusion step (**`DIFFUSION`** or **`EPITAXY`** statement) after the trap model is first enabled. In some cases it may be desirable to initialize the empty trap concentration to a value smaller than its equilibrium value. This can be accomplished by first setting T_E^* to the desired initial concentration of empty traps, doing a short diffusion (with **`TRAP.CON`** set nonzero to enable the trap model), then setting T_E^* to the actual equilibrium concentration of empty traps.

Interstitial Clustering Models

TSUPREM-4 includes two models for the formation and dissolution of interstitial clusters ({311} or {113} defects). These clusters play an important part in transient-enhanced diffusion (TED) of impurities following ion implantation. The main effect of the models is to delay the onset of TED at low temperatures and to distribute the diffusion enhancement over a longer period of time. This eliminates the excessive diffusion at low temperatures that is predicted by older versions of TSUPREM-4.

The two interstitial clustering models are distinguished by the number of equations used to characterize the population of clusters. The 1-moment model uses a single equation to calculate the total number of interstitials bound in clusters. The 2-moment model uses an additional equation to calculate the number of clusters, and thus contains information on the size distribution of the clusters. All other things being equal, a 2-moment model should provide more flexible modeling and give more accurate results, but in practice the obtainable accuracy depends on the form of the model as well as on the number of moments. The 1-moment model in TSUPREM-4 is very general and can be adjusted to agree well with measured data. The 2-moment model is more constrained by theoretical concerns and is less flexible; while it can achieve reasonable agreement with experiment, it does not appear to have any advantages over the 1-moment model at present.

In addition to the model of {311} large interstitial clustering, TSUPREM-4 includes the two small interstitial clustering models, i.e., the equilibrium model and the transient model. The transient small interstitial clustering model provides the full dynamics of the interstitial clusters. The boolean parameter **CL.FULL** specifies to use the transient small interstitial clustering model.

1-Moment Clustering Model

The kinetics of {311} formation and dissolution are not well understood, and several models have been proposed (e.g., Reference [15]). TSUPREM-4, therefore, uses a generalized model that includes many of the proposed models as subsets.

$$\frac{\partial C_{Ic}}{\partial t} = R_{IcI} - R_{IcV} \qquad \text{Equation 3-319}$$

where C_{Ic} is the concentration of clustered interstitials, and R_{IcI} and R_{IcV} describe cluster interaction with interstitials and vacancies, respectively. The interstitial interaction includes three terms describing the nucleation, the clustering of interstitials and one describing the declustering:

$$R_{IcI} = R_{sIcI} + R_{lIcI} \text{ for } \hat{}\mathbf{CL.FULL} \qquad \text{Equation 3-320}$$

or:

$$R_{IcI} = (max + 1)R_{sIcI(max+1)} + R_{lIcI} \text{ for } \mathbf{CL.FULL} \qquad \text{Equation 3-321}$$

where I is the concentration of unclustered interstitials, and I^* is the equilibrium concentration of interstitials. The first term R_{sIcI} in Equation 3-320 models the nucleation of the interstitial clusters at high interstitial supersaturation, which is given by:

$$R_{sIcI} = K_{fi} \frac{I^{\mathbf{CL.IFI}}}{I^{*\,\mathbf{CL.ISFI}}} \qquad \text{Equation 3-322}$$

When the **CL.FULL** model is set, the smallest large cluster forms when the small cluster $I_{(max)}$ captures one free interstitial by the reaction rate $R_{sIc(max+1)}$ (Transient Small Clustering, p. 3-80). The second term R_{lIcI} models growth and dissolution of the large interstitial clusters by consuming and emitting free interstitials.

$$R_{lIcI} = K_{fc} \frac{I^{\mathbf{CL.IFC}}}{I^{*\,\mathbf{CL.ISFC}}} (C_{Ic} + \alpha I)^{\mathbf{CL.CF}} - K_r C_{Ic}^{\mathbf{CL.CR}} \qquad \text{Equation 3-323}$$

The vacancy interaction includes the one recombination and one generation term, as well as the reaction with the interstitial small clusters (only for the **`CL.FULL`** model):

$$R_{IcV} = R_{lIcV} \text{ for } \hat{}\,\mathbf{CL.FULL} \qquad \text{Equation 3-324}$$

or:

$$R_{IcV} = (max+1)R_{sIcV(max+1)} + R_{lIcV} \text{ for } \mathbf{CL.FULL} \qquad \text{Equation 3-325}$$

$R_{sIcV(max+1)}$ is the reaction rate of small interstitial clusters with vacancies (Transient Small Clustering, p. 3-80).

$$R_{lIcV} = K_{rv}\frac{V^{\mathbf{CL.VRV}}}{V^{*\,\mathbf{CL.VSRV}}}C_{Ic}^{\;\mathbf{CL.CRV}} - K_{fv}V^{*\,\mathbf{CL.VSFV}}C_{Ic}^{\;\mathbf{CL.CFV}}$$

Equation 3-326

where V is the concentration of free vacancies, and V^* is the equilibrium concentration of vacancies. The first term models dissolution of the interstitial clusters by consuming vacancies. The second term models emission of vacancies by the interstitial clusters. The reactions constants are given by:

$$K_{fi} = \mathbf{CL.KFI.0} \cdot \exp\left(-\frac{\mathbf{CL.KFI.E}}{kT}\right) \qquad \text{Equation 3-327}$$

$$K_{fc} = \mathbf{CL.KFC.0} \cdot \exp\left(-\frac{\mathbf{CL.KFC.E}}{kT}\right) \qquad \text{Equation 3-328}$$

$$K_{r} = \mathbf{CL.KR.0} \cdot \exp\left(-\frac{\mathbf{CL.KR.E}}{kT}\right) \qquad \text{Equation 3-329}$$

$$K_{rv} = \mathbf{CL.KRV.0} \cdot \exp\left(-\frac{\mathbf{CL.KRV.E}}{kT}\right) \qquad \text{Equation 3-330}$$

$$K_{fv} = \mathbf{CL.KFV.0} \cdot \exp\left(-\frac{\mathbf{CL.KFV.E}}{kT}\right) \qquad \text{Equation 3-331}$$

and α = **`CL.KFCI`**, **`CL.IFI`**, **`CL.ISFI`**, **`CL.IFC`**, **`CL.ISFC`**, **`CL.CF`**, **`CL.CR`**, **`CL.KFI.0`**, **`CL.KFI.E`**, **`CL.KFC.0`**, **`CL.KFC.E`**, **`CL.KR.0`**, and **`CL.KR.E`** are parameters specified on the **`INTERSTITIAL`** statement. Besides, **`CL.KRV.0`**, **`CL.KRV.E`**, **`CL.KFV.0`**, **`CL.KFV.E`**, **`CL.VRV`**, **`CL.VSRV`**, **`CL.CRV`**, **`CL.VSFV`** and **`CL.CFV`** are parameters specified on the **`VACANCY`** statement.

The changes in C_{Ic} are accompanied by corresponding inverse changes in I. Thus clustering reduces the number of free interstitials, while the dissolution of clusters releases interstitials.

The clustered interstitial concentration C_{Ic} is initialized to the value specified for **ABS.ERR** (10^9/cm^3 in silicon) when the point defect models are first activated. After that, the concentration is determined by Equation 3-319.

Choosing Model Parameters

The interstitial clustering model is designed to work automatically, but for correct operation suitable parameter values must be chosen. Parameters for the model are set on the **INTERSTITIAL** and **VACANCY** statements specifying **MATERIAL=SILICON** as the material. Parameters should be chosen so that:

- Clusters form rapidly when the interstitial concentration is very high (i.e., following an implant).
- Clusters decay at a suitable rate when the interstitial concentration is low.
- There is a small but nonzero concentration of clustered interstitials at equilibrium. This is required for numerical reasons, and may be needed to start the clustering process following a subsequent implant.
- The clustering model does not interfere with simulations of high-concentration diffusion, oxidation-enhanced diffusion, or other situations in which nonequilibrium interstitial concentrations may be present.
- The clustering model is numerically well behaved.

To obtain clustering following an implant, either the K_{fi} or K_{fc} term (or both) must be nonzero. For rapid clustering in response to an excess of interstitials, **CL.IFI**, **CL.IFC**, and/or **CL.CF** must be set appropriately; larger values increase the dependence of the clustering rate on the interstitial concentration. Use **CL.ISFI** and **CL.ISFC** to modify the clustering rate at high doping concentrations.

Dissolution of clusters and the release of interstitials requires a nonzero K_r term. The parameter **CL.CR** must be greater than zero to avoid numerical difficulties (a value of 1.0 is typical). The forward reaction terms are also active during dissolution, so the net dissolution rate depends on the K_{fi} and K_{fc} terms as well as on the K_r term.

The clustering model should also be examined to determine its impact on high-concentration diffusion (with **PD.FULL**) or OED. In either of these cases, excess interstitials could produce clustering, which can affect the diffusion of impurities.

The interstitial clustering model has been designed to be numerically well behaved when used with “reasonable” parameter values. In general this means that the clustering and dissolution rates must not be too large and the concentration of clustered interstitials must approach a reasonable value in equilibrium. In practice, it has been found that parameter values that give physically reasonable results are numerically well behaved.

The default parameter values use only the K_{fc} and K_r terms. They have been chosen to fit the clustering data of Poate, et al. Reference [16] when used with the default values of the point defect parameters.

Model Initialization

The initial concentration of interstitial clusters after implants can be set by using the parameters, either **`CL.INI.F`** or **`CL.EQUIL`** in the **`INTERSTI`** statement. Turning on **`CL.EQUIL`** precedes the specification of **`CL.INI.F`**. **`CL.INI.F`** and determines the fraction of interstitials that form interstitial clusters at the beginning of the post-implant anneal. The value of the **`CL.INI.F`** parameter must be in the range from 0 to 1. **`CL.INI.F`** applies to the interstitials that are remaining after the dopant-defect cluster formation. For example, if the value of **`CL.INI.F`** is 1, it will grab all interstitials that are remaining after the dopant-defect clusters are created. It is not recommended to set it to 1 because there will be no interstitials left to form the mobile equilibrium dopant-defect pairs. Turning on **`CL.EQUIL`** specifies C_{Ic} to be initialized to the equilibrium concentration. It is preferable for the 3 stream diffusion model to assume the interstitial cluster concentration in equilibrium at the beginning of an anneal following implantation Reference [17]. The equilibrium concentration of clustered interstitials depends on the balance between all its terms. It can be found by setting $\partial C_{Ic}/\partial t = 0$ in Equation 3-319 and solving it consistently with other equations for I, V and C_{Ic}. The concentrations of point defects, I and V, and the concentration of interstitial clusters, C_{Ic} ,are calculated by numerically solving the following equations:

$$(I - I_0) + \sum \frac{k_m}{k_m + k_{mv}V/V^*}(M - M_0) - \sum \frac{k_{ni}I/I^*}{k_n + k_{ni}I/I^*}(N - N_0) +$$

$$(I_c - I_{c,0}) + (I_{ddcs} - I_{ddcs,0}) + (C_{icl} - C_{icl,0}) = \Delta I$$

Equation 3-332

$$(V - V_0) + \sum \frac{k_n}{k_n + k_{ni}I/I^*}(N - N_0) - \sum \frac{k_{mv}V/V^*}{k_m + k_{mv}V/V^*}(M - M_0)$$

$$+ (V_c - V_{c,0}) = \Delta V$$

Equation 3-333

$$R_{IcI} - R_{IcV} = 0$$

Equation 3-334

where the subscript ‘0’ denotes the solutions before the point defects are added by ΔI and ΔV into the structure and/or the system temperature is abruptly changed.

Note:

The CL.EQUIL model is not applicable when the CL.FULL model is turned on. If no convergence happens with CL.EQUIL, the initial intersitial clustering concentration will be calculated with CL.INI.F.

The description of the **`CL.INI.F`** parameter and **`CL.EQUIL`** boolean parameter provided above, applies outside of the amorphized layer. Inside the amorphized layer, all interstitial clusters, even those created before this implant, are completely wiped out during the epitaxial recrystallization of the amorphized layer.

Using the Model

The model is enabled by using the **`CL.1M`** parameter on the **`INTERSTITIAL`** statement:

```
INTERST MAT=SILICON CL.1M
```

The model is enabled by default. To disable the model use:

```
INTERST MAT=SILICON CL.NONE
```

The concentration of clustered interstitials is obtained by specifying the name *cl_interst* in the **`SELECT`** statement:

```
SELECT Z=LOG10(CL_INTERST)
```

In Synopsys TCAD's graphical post-processing programs and in TIF files, the name *clInterst* is used.

2-Moment Interstitial Clustering Model

The 2-moment interstitial clustering model, based on the work of Gencer and Dunham Reference [18], includes the nucleation and growth of {311} clusters, Ostwald ripening and decay of clusters, and conversion of {311} clusters into dislocation loops. The model calculates the first two moments of the size distribution of interstitial clusters, i.e., the number of clusters and the number of interstitials contained in the clusters. It can be used with the existing model for dislocation loops, although it is designed to include modeling of at least some dislocation loops.

Model Equations

The 2-moment clustering model is an implementation of Gencer's analytical kinetic precipitation model (AKPM). The density of clusters D_{Ic} and concentration of interstitials C_{Ic} contained in clusters are calculated as:

$$\frac{\partial D_{Ic}}{\partial t} = D_I \lambda_0 (I^2 - D_{Ic} I_i^* C_s \gamma_0) \qquad \text{Equation 3-335}$$

$$\frac{\partial C_{Ic}}{\partial t} = R_{sIcI} + R_{lIcI} \qquad \text{Equation 3-336}$$

$$R_{sIcI} = 2\frac{\partial D_{Ic}}{\partial t} \qquad \text{Equation 3-337}$$

$$R_{lIcI} = D_I \lambda_1 D_{Ic} (I - I_i^* C_s \gamma_1) \qquad \text{Equation 3-338}$$

where I and D_I are the concentration and diffusivity of free interstitials.

When the **`CL.FULL`** model is set, Equation 3-335 and Equation 3-336 are modified as follows:

$$\frac{\partial D_{Ic}}{\partial t} = R_{sIcI(max+1)} - R_{sIcV(max+1)} - D_I\lambda_0 D_{Ic} I_i^* C_s\gamma_0 \qquad \text{Equation 3-339}$$

$$\frac{\partial C_{Ic}}{\partial t} = (max+1)(R_{sIcI(max+1)} - R_{sIcV(max+1)}) + R_{IIcI} \qquad \text{Equation 3-340}$$

$C_s\gamma_0$, which gives the dependence of the dissolution rate on the average cluster size, is given by:

$$C_s\gamma_0 = \begin{cases} C_{ss}\dfrac{K_1}{s-1} & \text{for } s < n_{crit} \\ C_{sl}K_3\left(\dfrac{1}{s-1}\right)^{\alpha} & \text{for } s > n_{crit} \end{cases} \qquad \text{Equation 3-341}$$

where K_1 is specified by **`CL.K1`**, which directly controls the dissolution of two-atom clusters; and $s = C/\ D$ is the average number of interstitials in a cluster.

$C_s\gamma_1$, which gives the dependence of the rate of interstitial release on the average cluster size, is given by:

$$C_s\gamma_1 = \begin{cases} C_{ss}\dfrac{s-2}{s+K_0}\left[1+\dfrac{(K_0+2)K_2}{s+K_0}\right] & \text{for } s < n_{crit} \\ C_{sl}\left[1+K_4\left(\dfrac{K_0+2}{s+K_0}\right)^{\alpha}\right] & \text{for } s > n_{crit} \end{cases} \qquad \text{Equation 3-342}$$

where K_3 and K_4 are chosen to make $C_s\gamma_0$ and $C_s\gamma_1$ continuous at $s = n_{crit}$; $\lambda_0 = $ **`CL.LAM0`**. $\lambda_1 = $ **`CL.LAM1`**. $K_0 = $ **`CL.K0`**. $K_2 = $ **`CL.K2`**. $\alpha = $ **`CL.ALPHA`**. $n_{crit} = $ **`CL.NCRIT`**; and

$$C_{ss} = \mathtt{CL.CSS.0} \cdot \exp\left(-\frac{\mathtt{CL.CSS.E}}{kT}\right) \qquad \text{Equation 3-343}$$

$$C_{sl} = \mathtt{CL.CSL.0} \cdot \exp\left(-\frac{\mathtt{CL.CSL.E}}{kT}\right) \qquad \text{Equation 3-344}$$

`CL.LAM0`, **`CL.LAM1`**, **`CL.K0`**, **`CL.K1`**, **`CL.K2`**, **`CL.CSS.0`**, **`CL.CSS.E`**, **`CL.CSL.0`**, **`CL.CSL.E`**, **`CL.NCRIT`**, and **`CL.ALPHA`** are parameters specified for **`MATERIAL=SILICON`** on the **`INTERSTITIAL`** statement.

Equation 3-335 models the nucleation and dissolution of two-atom clusters. λ_0 is the capture length for these processes.

Equation 3-336 models the absorption and release of interstitials by clusters. The three terms on the right side model the absorption of interstitials during nucleation, the absorption of interstitials by nucleated clusters, and the release of interstitials by nucleated clusters. λ_1 is the capture length for absorption and release of interstitials by nucleated clusters. $I_i{*}C_{ss}$ is the concentration of interstitials in equilibrium with a population of large {311} clusters.

For $s > n_{crit}$, some of the {311} defects unfault to form dislocation loops. These dislocation loops are included in the 2-moment model by modifying the cluster dissolution rates. α controls the dissolution rate when loops are present, and $I_i{*}C_{sl}$ is the concentration of interstitials in equilibrium with a population of large dislocation loops.

Using the Model

The 2-moment clustering model is an alternative to the previously available (1-moment) model of {311} clusters. It is selected by specifying the parameter **`CL.2M`** on the **`INTERSTITIAL`** statement. The old, 1-moment model is now selected by **`CL.1M`** (with **`CL.MODEL`** as a synonym) and both models are disabled by specifying **`CL.NONE`**. The clustering models can be used in conjunction with any other models in the program, including the existing models for dislocation loops and small interstitial clusters, but you may wish to disable the older models when the 2-moment model is used to avoid modeling the same physical phenomena twice.

At present, the 2-moment clustering model is an experimental model: It is intended for those doing research into alternative models and not for routine simulations. No defaults are provided for the model parameters, but sample parameter values are available on request.

The concentration of clustered interstitials is obtained by specifying the name *cl_interst* in the **`SELECT`** statement:

```
SELECT Z=LOG10(CL_INTERST)
```

The concentration of clusters (the number of clusters per unit volume) is obtained by specifying *cl_im0*:

```
SELECT Z=LOG10(CL_IM0)
```

Small Clusters of Point Defects

At high concentrations, point defects will tend to form small, loosely-bound clusters, reducing the number of free defects available to enhance diffusion of dopants.

Concentration of Defects in Small Clusters

These are modeled in TSUPREM-4 by assuming that the number of defects in small clusters in equilibrium with the number of free defects:

$$I_c = \mathbf{\mathtt{ECLUST.N}}\sum_i \left(K_{Ii}\eta^{-i}\right)\left(\frac{I}{I^*}\right)^{\mathbf{\mathtt{ECLUST.N}}}$$

$$V_c = \mathbf{\mathtt{ECLUST.N}}\sum_j \left(K_{Vj}\eta^{-j}\right)\left(\frac{V}{V^*}\right)^{\mathbf{\mathtt{ECLUST.N}}} \qquad \text{Equation 3-345}$$

$$K_{Ii} = \mathbf{\mathtt{ECLUST.0}} \cdot \exp\left(\frac{-\mathbf{\mathtt{ECLUST.E}}}{kT}\right)$$

$$K_{Vj} = \mathbf{\mathtt{ECLUST.0}} \cdot \exp\left(\frac{-\mathbf{\mathtt{ECLUST.E}}}{kT}\right) \qquad \text{Equation 3-346}$$

where **ECLUST.0**, **ECLUST.E**, and **ECLUST.N** for interstitials of charge state $\mathbf{\mathtt{C.STATE}} = i$ and vacancies of charge state $\mathbf{\mathtt{C.STATE}} = j$ are specified on the **INTERSTITIAL** and **VACANCY** statements, respectively. **ECLUST.N** denotes the number of defects in the cluster (typically 2–4) while **ECLUST.0** and **ECLUST.E** specify the concentration of defects in small clusters when the free point defect concentrations are in thermal equilibrium.

Recombination of Defects in Small Clusters

Vacancies can react with small clusters of interstitials to reduce the total number of interstitials and vacancies:

$$F_{cV} = \left(\frac{I}{I^*}\right)^{\mathbf{\mathtt{ECLUST.N}}-1} V_i^* \sum_{i,\ j}\left(K_{cV,\ ij}K_{Ii}\eta^{-i}\phi_{Vj}\eta^{-j}\right) \qquad \text{Equation 3-347}$$

$$K_{cV,\ ij} = \mathbf{\mathtt{KCV.0}} \cdot \exp\left(\frac{-\ \mathbf{\mathtt{KCV.E}}}{kT}\right) \qquad \text{Equation 3-348}$$

where **KCV.0** and **KCV.E** (parameters on the **INTERSTITIAL** statement) specify the relative reaction rate between small interstitial clusters in charge state $\mathbf{\mathtt{C.I}} = i$ and vacancies in charge state $\mathbf{\mathtt{C.V}} = j$.

Similarly, interstitials can react with small clusters of vacancies to reduce the total number of interstitials and vacancies:

$$F_{Ic} = I_i^*\left(\frac{V}{V^*}\right)^{\mathbf{\mathtt{ECLUST.N}}-1} \sum_{i,\ j}\left(K_{Ic,\ ij}\phi_{Ii}\eta^{-i}K_{Vj}\eta^{-j}\right) \qquad \text{Equation 3-349}$$

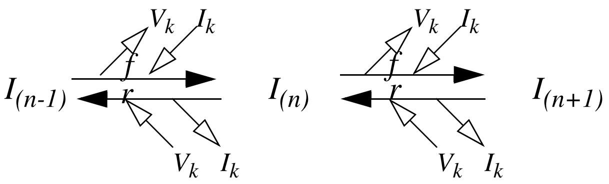
$$K_{Ic,\ ij} = \mathbf{\mathtt{KIC.0}} \cdot \exp\left(\frac{-\ \mathbf{\mathtt{KIC.E}}}{kT}\right) \qquad \text{Equation 3-350}$$

where **KIC.0** and **KIC.E** (parameters on the **VACANCY** statement) specify the relative reaction rate between interstitials in charge state **C.I** $= i$ and small vacancy clusters in charge state **C.V** $= j$. **KCV.0**, **KCV.E**, **KIC.0**, and **KIC.E** for all charge state combinations can be set using **C.ALL**.

The number of point defects in small clusters can be significant following an implant with damage, especially at low temperatures. These clusters can reduce the number of free point defects and the amount of diffusion enhancement of dopants.

Transient Small Clustering

The reactions associated with the size-n small interstitial cluster is as follows:



$I_{(n)}$, I_k and V_k denote the n-size interstitial small cluster, k-charged interstitial and k-charged vacancy, respectively. The transient equation for the n-size small interstitial cluster is:

$$\frac{\partial I_{(n)}}{\partial t} = R_{cI(n)} - R_{cV(n)} - \sum R_{DDC,\ I(n)+M} \qquad \text{Equation 3-351}$$

for $2 \le n \le max$, where $R_{DDC,\ I(n)+M}$ models the lost of the n-size small clusters by reaction with dopant-interstitial pairs to form the dopant-interstitial clusters(Full Dynamics of Dopant-Defect Clustering, p. 3-40), and $R_{cI(n)}$ and $R_{cV(n)}$ are described as follows:

$$R_{cI(n)} = R_{cI((n-1)\to(n))} - R_{cI((n)\to(n+1))} \qquad \text{Equation 3-352}$$

$$R_{cV(n)} = R_{cV((n)\to(n-1))} - R_{cV((n+1)\to(n))} \qquad \text{Equation 3-353}$$

$$R_{cI((n-1)\to(n))} = k_{fi}^{(n)} I_{(n-1)} \frac{I}{I^*} - k_{ri}^{(n)} I_{(n)} \qquad \text{Equation 3-354}$$

$$R_{cI((n)\to(n+1))} = k_{fi}^{(n+1)} I_{(n)} \frac{I}{I^*} - k_{ri}^{(n+1)} I_{(n+1)} \qquad \text{Equation 3-355}$$

$$R_{cV((n)\to(n-1))} = k_{rv}^{(n)} I_{(n)} \frac{V}{V^*} - k_{fv}^{(n)} I_{(n-1)} \qquad \text{Equation 3-356}$$

$$R_{cV((n+1)\to(n))} = k_{rv}^{(n+1)} I_{(n+1)} \frac{V}{V^*} - k_{fv}^{(n+1)} I_{(n)} \qquad \text{Equation 3-357}$$

where:

$$k_{fi}^{(n)} = I_i^* \sum_k f_{nk}^i \phi_{Ik} \eta^{-k} \text{ for n>2}$$

Equation 3-358

$$k_{fi}^{(2)} = \frac{I_i^{*2}}{I^*} \sum_{j,\ k} f_{2jk}^i \phi_{Ij} \phi_{Ik} \eta^{-(j+k)} \text{ for n=2}$$

Equation 3-359

$$k_{ri}^{(n)} = I_i^* \sum_k r_{nk}^i \phi_{Ik} \eta^{-k} \text{ for n>2}$$

Equation 3-360

$$k_{ri}^{(2)} = I_i^* \sum_{j,\ k} r_{2jk}^i \phi_{Ij} \phi_{Ik} \eta^{-(j+k)} \text{ for n=2}$$

Equation 3-361

$$k_{fv}^{(n)} = V_i^* \sum_k f_{nk}^v \phi_{Vk} \eta^{-k} \text{ for n>2}$$

Equation 3-362

$$k_{fv}^{(2)} = \frac{I_i^* V_i^*}{I^*} \sum_k f_{nk}^v \phi_{Vk} \eta^{-k} \text{ for n=2}$$

Equation 3-363

$$k_{rv}^{(n)} = V_i^* \sum_k r_{nk}^v \phi_{Vk} \eta^{-k}$$

Equation 3-364

The parameter **`SC.SIZE`** defines the size *n* and the charge state *k* of interstitials and vacancies can be specified with the parameters **`C.INTERS`** and **`C.VACANC`** respectively in the **`INTERSTI`** statement. The reaction rate constants f_{nk}^i, r_{nk}^i, f_{nk}^v, and r_{nk}^v are computed as follows:

$$f_{nk}^i = \texttt{SC.KFI.0} \cdot \exp\left(-\frac{\texttt{SC.KFI.E}}{kT}\right)$$

Equation 3-365

$$r_{nk}^i = \texttt{SC.KRI.0} \cdot \exp\left(-\frac{\texttt{SC.KRI.E}}{kT}\right)$$

Equation 3-366

$$f_{nk}^v = \texttt{SC.KFV.0} \cdot \exp\left(-\frac{\texttt{SC.KFV.E}}{kT}\right)$$

Equation 3-367

$$r_{nk}^v = \texttt{SC.KRV.0} \cdot \exp\left(-\frac{\texttt{SC.KRV.E}}{kT}\right)$$

Equation 3-368

For **SC.SIZE**=2, in addition to **C.STATE**, the parameter **SC.C.I** is provided to specify the charge state of the other interstitial. **SC.C.I** must be equal to or less than **C.STATE**.

The net capture rate of free interstitials by the small interstitial clusters is given by:

$$R_{sIcI} = R_{cI(1 \to 2)} + \sum_{n=1}^{max} R_{cI(n \to n+1)} \qquad \text{Equation 3-369}$$

The net capture rate of free vacancies by the small interstitial clusters is given by:

$$R_{sIcV} = \sum_{n=1}^{max} R_{cV(n+1 \to n)} \qquad \text{Equation 3-370}$$

Dislocation Loop Model

It has been observed that amorphizing implants give rise to dislocation loops at the edge of the amorphized region following annealing. The size of these loops has been observed to grow (presumably by absorption of interstitials) in oxidizing ambients and shrink (by emission of interstitials) in inert ambients. The observed reduction in the amount of oxidation-enhanced diffusion (OED) in underlying dopant distributions is consistent with the presumption of growth and shrinkage by absorption and emission of interstitials.

Dislocation loops can also form at the peak of the damage distribution following nonamorphizing implants.

Equations for Dislocation Loop Model

The Dislocation Loop Model implements the interaction between interstitials and dislocation loops Reference [103]. The rate of absorption of interstitials by dislocation loops R_l is given by:

$$R_l = 2\pi^2 K_L D_I \rho_v r \left(I - I^*_{loop}\right) \qquad \text{Equation 3-371}$$

$$K_L = \mathbf{KLOOP.0}\ \exp\left(-\frac{\mathbf{KLOOP.E}}{kT}\right) \qquad \text{Equation 3-372}$$

$$I^*_{loop} = I^* \exp\left(\frac{\gamma\Omega}{bkT}\right) \exp\left[\frac{\mu b \Omega}{4\pi r k T(1-\nu)} \ln\left(\frac{8r}{b}\right)\right] \qquad \text{Equation 3-373}$$

where ρ_v is the volume density (number/cm^3), r is the loop radius (in cm), and **KLOOP.0** and **KLOOP.E** are parameters on the **INTERSTITIAL** statement. The default value for **KLOOP.0** (29.8047) is the value fit by Huang and Dutton

Reference [103]; the value of the activation energy in the expression for **KLOOP.E** (0.4) and the expression for I^*_{loop} are from Hu Reference [104].

The other quantities in the equation above are:

- D_I, diffusivity of interstitials.
- I, interstitial concentration.
- k, Boltzmann's constant.
- T, temperature (Kelvins).
- γ, internal energy associated with a stacking fault in silicon (70×10^{-5} N/cm).
- Ω, volume per silicon atom (2×10^{-23} cm^3).
- b, magnitude of the Burger's vector for the dislocation loop (3.14×10^{-8} cm).
- μ, shear modulus of silicon (6.6406×10^6 N/cm^2).
- v, Poisson's ratio for silicon (0.28).

The evolution of the loop radius r is given by:

$$\frac{dr}{dt} = \left(\frac{\pi}{N_o}\right) K_L D_I (I - I^*_{loop})$$ Equation 3-374

where:

- N_o is the {111} planar density of silicon ($1.57 \times 10^{15}/cm^2$).
- I^*_{loop} is the equilibrium concentration of interstitials in the presence of dislocation loops.

I^*_{loop} decreases from about 10 times I^* for 100Å loops down to one or two times I^* for loops of 1000Å or more. Loops of radius **RLMIN** or less dissolve quickly during diffusion in an inert ambient. To save simulation time, such loops are discarded. **RLMIN** is specified on the **INTERSTITIAL** statement; it initially defaults to 100×10^{-8} cm (100Å).

The initial loop radius is specified by the **L.RADIUS** parameter on the **IMPLANT** statement. The density of loops can either be set to a fixed value (specified by the **L.DENS** parameter with the location derived from the **L.DMIN** and **L.DMAX** parameters) or the location and density can be specified indirectly with the **L.THRESH** and **L.FRAC** parameters.

Loop Density Specified by L.DENS

End-of-range dislocation loops are created following amorphizing ion implantation steps. The initial volume density ρ_v and radius r are specified by the **L.DENS** and **L.RADIUS** parameters, respectively, on the **IMPLANT** statement. Loops are produced in that portion of the structure where the interstitial concentration (due to implant damage, before recombination) is in the range:

$$\mathbf{L.DMIN} < I \leq \mathbf{L.DMAX}$$ Equation 3-375

where **`L.DMIN`** and **`L.DMAX`** are parameters on the **`IMPLANT`** statement (default values of 10^{20}/cm^3 and 1.15x10^{22}/cm^3). The concentration of interstitials corresponding to the edge of the amorphous region is from the work of Cerva and Hobler Reference [5]. The concentration of interstitials is not reduced by the formation of end-of-range loops. Pre-existing dislocation loops in the region where $I > \texttt{L.DMIN}$ are presumed to be destroyed by the implant. The loop density and radius are taken to be constant for a particular implant.

Loop Density Specified by L.THRESH

Dislocation loops can also be created near the peak of the damage profile produced by nonamorphizing implants. These loops are modeled by assuming that a specified fraction **`L.FRAC`** of the interstitials above some threshold **`L.THRESH`** are incorporated into dislocation loops of radius **`L.RADIUS`**, where **`L.FRAC`**, **`L.THRESH`**, and **`L.RADIUS`** are parameters on the **`IMPLANT`** statement. The loop density is given by:

$$\rho_v = \frac{\texttt{L.FRAC} \cdot \max(I - \texttt{L.THRESH},\ 0)}{\pi N_o r^2} \qquad \text{Equation 3-376}$$

The concentration of interstitials is reduced by the number of interstitials incorporated into dislocation loops, $\texttt{L.FRAC}\ \max(I - \texttt{L.THRESH},\ 0)$.

Evolution of Loops

Following the implant, the loop density remains constant, but the change in radius is a function of the interstitial concentration and thus depends on position. When the radius decreases to less than **`RLMIN`**, both the radius and density are set to zero. The density can vary with position as a result of multiple implants. The density and radius can be selected for printing or plotting by specifying *dloop* or *rloop*, respectively, as values of the **`Z`** parameter on the **`SELECT`** statement. The loop radius is reported in centimeters.

Effects of Dislocation Loops

Dislocation loops affect the diffusion of impurities by absorbing or emitting interstitials. They tend to reduce the impact of transient-enhanced diffusion effects immediately after an implant, but can produce a long-term diffusivity enhancement as they dissolve; this effect is particularly important for high-dose subamorphizing boron implants. Similarly, they tend to reduce the immediate impact of oxidation on impurity diffusion while producing a smaller long-term enhancement.

Oxidation

Oxidation occurs whenever a **`DIFFUSION`** statement specifies an oxidizing ambient and either exposed silicon or polysilicon or exposed oxide adjacent to silicon or polysilicon is present in the structure. An oxidizing ambient is one containing O_2, H_2O, or N_2O. The flows of O_2, H_2O, or N_2O can be specified directly on the **`DIFFUSION`** statement or a predefined ambient containing O_2, H_2O, or N_2O (i.e., **`DRYO2`**, **`WETO2`**, **`STEAM`**,or **`N2O`**) can be specified (see Diffusion, p. 3-14, and Chapter 4, DIFFUSION, p. 4-131, and Chapter 4, AMBIENT, p. 4-233). If an ambient contains O_2 or N_2O with H_2O, the oxidation rate is based on the partial pressure of H_2O.

Oxidation occurs at points in the structure where an oxide region is adjacent to a silicon or polysilicon region. If a structure contains an exposed silicon or polysilicon surface at the start of an oxidation step, a native oxide of thickness **`INITIAL`** (specified on the **`AMBIENT`** statement) is deposited on this surface before proceeding with the oxidation.

Five oxidation models are available. All are based on the 1-D theory of Deal and Grove Reference [19]. The differences are in the way they extend the Deal-Grove model to 2-D. Theory of Oxidation, p. 3-85 outlines the basic theory of oxidation, while the sections Analytical Oxidation Models, p. 3-87 through VISCOELA Model, p. 3-99 describe how the theory is applied by the models available in TSUPREM-4.

Theory of Oxidation

Oxidation in TSUPREM-4 is based on the theory of Deal and Grove Reference [19], which is briefly outlined. The flux of oxidant (assumed to be O_2 or H_2O) entering the oxide from the ambient gas is given by:

$$\vec{F} = h(C^* - C_o)\vec{n}_s \qquad \text{Equation 3-377}$$

where:

- h is the gas-phase mass-transfer coefficient.
- C_o is the concentration of oxidant in the oxide at the surface
- $C^* = HP_{ox}$ where:

  H is the Henry's law coefficient for the oxidant in oxide and
  P_{ox} is the partial pressure of oxidant in the ambient.
- $\vec{n}_s$ is the unit vector normal to the oxide surface, pointing towards the oxide.

The flux of oxidant in the oxide is:

$$\vec{F} = D\vec{\nabla}C \qquad \text{Equation 3-378}$$

where:

- D is the diffusivity of oxidant in the oxide.
- C is the local concentration of oxidant.
- $\vec{\nabla}$ is the gradient operator.

The rate of oxidant consumption at the oxidizing interface is:

$$\vec{F} = k_s C\vec{n}_i \qquad \text{Equation 3-379}$$

where:

- k_s is the surface reaction rate.
- C is the oxidant concentration at the interface.
- $\vec{n}_i$ is the unit vector normal to the interface pointing away from the oxide.

In steady state, the divergence of the fluxes is zero:

$$\vec{\nabla} \cdot \vec{F} = 0 \tag{Equation 3-380}$$

The oxide growth rate is given by:

$$\frac{d\vec{Y}}{dt} = \frac{\vec{F}}{N_1} + r_{thin} \tag{Equation 3-381}$$

where:

- $d\vec{Y}/dt$ represents the interface velocity relative to the oxide.
- N_1 is the number of oxidant molecules needed to form each cubic centimeter of oxide.

The r_{thin} term models the rapid growth that is seen during the initial stages of oxidation; the calculation of this term depends on whether an analytical or numerical model is used.

In one dimension, Equations 3-377 through 3-381 can be solved to give:

$$\frac{dy}{dt} = \frac{B}{A + 2y} \tag{Equation 3-382}$$

where y is the oxide thickness and A and B are given by:

$$A = 2D\left(\frac{1}{k_s} + \frac{1}{h}\right) \tag{Equation 3-383}$$

$$B = \frac{2DC^*}{N_1} \tag{Equation 3-384}$$

In TSUPREM-4, A and B are specified in terms of the linear and parabolic rate constants B/A and B, respectively.

The analytical models are based on Equation 3-382, while the numerical models use Equations 3-377 through 3-381. Values of D and k_s for the numerical models are derived from Equations 3-383 and 3-384, using the user-specified linear and parabolic rate constants.

Analytical Oxidation Models

There are two analytical models, **ERFC** and **ERFG** (**ERF1** and **ERF2** are subsets of **ERFG**). They differ in how the growth rate depends on the x coordinate near the mask edge.

Overview

The analytical oxidation models in TSUPREM-4 are designed for fast simulation of simple structures. They are limited to structures consisting of a planar silicon substrate covered by an optional initial oxide layer; masking layers over the oxide are ignored. Oxidation occurs to the right of an assumed mask location specified by **MASK.EDG** on the **AMBIENT** statement. Any actual mask layers on the structure move vertically with the surface of the growing oxide, but do not affect the oxide shape. Furthermore, the **ERF1**, **ERF2**, and **ERFG** models assume that the initial silicon surface is planar, at $y = 0$.

Oxide Growth Rate

The analytical oxidation models are based on Equation 3-382, with an added term to model thin oxide growth. Far to the right of **MASK.EDG** the oxidation rate is:

$$\frac{dy_\infty}{dt} = \frac{B}{A + 2y_\infty} + r_{thin} \qquad \text{Equation 3-385}$$

where:

- y_∞ is the unmasked (one-dimensional) oxide thickness.
- B/A, B, and r_{thin} are the linear, parabolic, and thin regime oxidation rate constants, respectively, described below.

Thin Regime

The thin regime oxidation rate constant is given by Reference [20].

$$r_{thin} = \mathtt{THINOX.0} \cdot \exp\left(\frac{-\mathtt{THINOX.E}}{kT}\right) \exp\left(\frac{-y_\infty}{\mathtt{THINOX.L}}\right) \qquad \text{Equation 3-386}$$

where the parameters **THINOX.0**, **THINOX.E**, and **THINOX.L** are specified in the **AMBIENT** statement (see Chapter 4, AMBIENT, p. 4-233) for each of the oxidizing species. Different values of **THINOX.0**, **THINOX.E**, and **THINOX.L** can be defined for each of the three available silicon orientations and for polysilicon by specifying the **<111>**, **<110>**, **<100>**, or **POLYSILI** parameters in the **AMBIENT** statement.

Linear Rate

The linear oxidation rate constant is given by References [3], [21], [22], and [23]:

$$B/A = l_0 l_P l_C l_{Cl} \qquad \text{Equation 3-387}$$

where l_0 represents the intrinsic linear oxidation rate and l_P, l_C, and l_{Cl} represent the dependence on partial pressure, carrier concentration, and chlorine concentration, respectively. The intrinsic linear oxidation rate is given by:

$$l_0 = \begin{cases} \mathtt{L.LIN.0} \cdot \exp\left(\dfrac{-\mathtt{L.LIN.E}}{kT}\right) & T_c < \mathtt{LIN.BREA} \\ \mathtt{H.LIN.0} \cdot \exp\left(\dfrac{-\mathtt{H.LIN.E}}{kT}\right) & T_c \geq \mathtt{LIN.BREA} \end{cases} \qquad \text{Equation 3-388}$$

where T_c is the diffusion temperature in °C, and the parameters **L.LIN.0**, **L.LIN.E**, **H.LIN.0**, **H.LIN.E**, and **LIN.BREA** are specified in the **AMBIENT** statement for each of the oxidizing species. Different values of **L.LIN.0**, **L.LIN.E**, **H.LIN.0**, and **H.LIN.E** can be defined for each of the three available silicon orientations and for polysilicon by specifying the **<111>**, **<110>**, **<100>**, or **POLYSILI** parameters in the **AMBIENT** statement.

The partial pressure dependence of the linear oxidation rate is given by:

$$l_P = \left(\frac{P}{1\ \text{atmosphere}}\right)^{\mathtt{LIN.PDEP}} \qquad \text{Equation 3-389}$$

where the parameter **LIN.PDEP** is specified in the **AMBIENT** statement for each of the oxidizing species.

The concentration dependence l_C is only used with the numeric models; it is described in Numerical Oxidation Models, p. 3-93.

The chlorine dependence of the linear oxidation rate is obtained by interpolating within a table of values depending on the chlorine percentage and the diffusion temperature (see DIFFUSION Statement, p. 3-15). The values in the table are specified with the **LIN.CLDE**, **COLUMN**, **TEMPERAT**, **LIN.PCT**, and **TABLE** parameters in the **AMBIENT** statement for each of the oxidizing species. Values of l_{Cl} are specified with the **LIN.CLDE** and **TABLE** parameters for the column of the table defined by the **COLUMN** parameter. The chlorine percentages associated with the rows of the table are defined with the **LIN.PCT** and **TABLE** parameters. The temperature associated with each column of the table is defined with the **TEMPERAT** parameter.

Parabolic Rate

The parabolic oxidation rate constant is given by References [3], [21], [22], and [23]:

$$B = p_0 p_P p_{Cl} \qquad \text{Equation 3-390}$$

where p_0 represents the intrinsic parabolic oxidation rate and p_P and p_{Cl} represent the dependence on partial pressure and chlorine concentration, respectively. The intrinsic parabolic oxidation rate is given by:

$$p_0 = \begin{cases} \mathtt{L.PAR.0} \cdot \exp\left(\dfrac{-\mathtt{L.PAR.E}}{kT}\right) & T_c < \mathtt{PAR.BREA} \\ \mathtt{H.PAR.0} \cdot \exp\left(\dfrac{-\mathtt{H.PAR.E}}{kT}\right) & T_c \geq \mathtt{PAR.BREA} \end{cases} \qquad \text{Equation 3-391}$$

where T_c is the diffusion temperature in ° C, and the parameters **L.PAR.0**, **L.PAR.E**, **H.PAR.0**, **H.PAR.E**, and **PAR.BREA** are specified in the **AMBIENT** statement for each of the oxidizing species.

The partial pressure dependence of the parabolic oxidation rate is given by:

$$p_P = \left(\frac{P}{1 \ \ \text{atmosphere}}\right)^{\mathtt{PAR.PDEP}} \qquad \text{Equation 3-392}$$

where the parameter **PAR.PDEP** is specified in the **AMBIENT** statement for each of the oxidizing species.

The chlorine dependence of the parabolic oxidation rate is obtained by interpolating within a table of values depending on the chlorine percentage and the diffusion temperature (see DIFFUSION Statement, p. 3-15).

The values in the table are specified with the **PAR.CLDE**, **COLUMN**, **TEMPERAT**, **PAR.PCT**, and **TABLE** parameters in the **AMBIENT** statement for each of the oxidizing species. Values of p_{Cl} are specified with the **PAR.CLDE** and **TABLE** parameters for the column of the table defined by the **COLUMN** parameter.

The chlorine percentages associated with the rows of the table are defined with the **PAR.PCT** and **TABLE** parameters. The temperature associated with each column of the table is defined with the **TEMPERAT** parameter.

Usage

Oxide growth is vertical, with the oxide interface moving in the +*y* direction and everything above it moving in the -*y* direction. (Actually, all silicon nodes, except for those at the oxide interface, remain stationary, while all nonsilicon nodes, except for oxide nodes at the interface, move with the oxide surface. This is the origin of the restrictions on the device structure for the analytical models.)

The analytical models are appropriate for oxidation of planar or near-planar surfaces; they are not appropriate for very nonplanar structures, or structures containing non-silicon layers below the top layer of oxide (e.g., silicon-on-insulator structures). Likewise, the analytical models cannot be used to simulate oxidation of polysilicon.

The unmasked thickness y_∞ is calculated by numerically integrating Equation 3-385, starting with the initial value specified by the **INITIAL** parameter on the **AMBIENT** statement. The analytical models are only correct when the value of the **INITIAL** parameter is equal to the actual initial oxide thickness. Because the **INITIAL** parameter also specifies the thickness of the native oxide deposited on exposed silicon surfaces prior to oxidation, the results are accurate when oxidizing bare silicon.

The ERFC Model

The **ERFC** model is the simplest oxidation model available in TSUPREM-4. It is selected by specifying the **ERFC** parameter on the **METHOD** statement (see Chapter 4, METHOD, p. 4-207). The **ERFC** model uses one parameter, **SPREAD**, in addition to the **INITIAL** and **MASK.EDG** parameters. All three parameters are specified on the **AMBIENT** statement (see Chapter 4, AMBIENT, p. 4-233).

The oxidation rate as a function of x is given by:

$$\frac{dy(x)}{dt} = \frac{1}{2}\,\mathrm{erfc}\left(\frac{\sqrt{2}}{\mathtt{SPREAD}} \cdot \frac{x_o - x}{y_\infty - \mathtt{INITIAL}}\right)\frac{dy_\infty}{dt} \qquad \text{Equation 3-393}$$

where:

- x_o = **MASK.EDG** is the position of the mask edge.
- y_∞ is the unmasked oxide thickness.

The **SPREAD** parameter controls the width of the “bird’s beak” relative to the unmasked oxide thickness. From Equation 3-393, you can see that the growth rate at x_o is half the unmasked growth rate; thus **MASK.EDG** actually represents the half-thickness point, not necessarily the true location of the mask edge.

Recommended Usage

The **ERFC** model is accurate for 1-D simulations, provided that the coefficients are accurate, **INITIAL** is correctly set or a bare silicon surface is being oxidized, and the dependence of the oxidation rate on doping can be neglected. The **ERFC** model can also be used for simulating local oxidation of planar or near-planar structures, if accurately calibrated. It is the fastest of the oxidation models, but this speed advantage is rarely significant. The **ERFC** model does not simulate the oxidation of polysilicon, nor does it take into account masking layers. Because of its limitations, the **ERFC** is not often used.

ERF1, ERF2, and ERFG Models

The **ERFG** model is a more complex analytical model for oxidation of silicon under a nitride mask. It is based on the work of Guillemot, et al. Reference [24] and contains two models, selected by the **ERF1** and **ERF2** parameters on the **METHOD** statement (see Chapter 4, METHOD, p. 4-207). If **ERFG** is specified, **ERF1** or **ERF2** is selected by the program based on the initial pad oxide and nitride mask thicknesses.

Parameters

The **ERF1** model uses three parameters, **NIT.THIC**, **ERF.H**, and **ERF.LBB** in addition to the **INITIAL** and **MASK.EDG** parameters. The **ERF2** model uses two additional parameters, **ERF.Q** and **ERF.DELT**. All seven parameters are specified on the **AMBIENT** statement (see Chapter 4, AMBIENT, p. 4-233).

Initial Structure

The **ERF1**, **ERF2**, and **ERFG** models assume that the initial structure consists of a planar silicon surface at $y = 0$, with a pad oxide of thickness **INITIAL** and a nitride mask of thickness **NIT.THIC** to the left of **MASK.EDG**. The nitride mask need not be present in the simulated structure.

ERF1 Model

The **ERF1** model is used when the thickness of the nitride mask is small compared to the pad oxide thickness. The oxidation rate as a function of x for the **ERF1** model is given by:

$$\frac{dy(x)}{dt} = \frac{1}{2}\,\mathrm{erfc}\left[\frac{x_o - x}{\mathbf{ERF.LBB}}\left(\sqrt{c^2 + \ln(10)} - c\right) + c\right]\frac{dy_\infty}{dt}$$

Equation 3-394

where:

$$c = \frac{\sqrt{\pi}}{2}\left(1 - \frac{2H'}{1 - 0.44}\right)$$

Equation 3-395

and x_o = **MASK.EDG** is the position of the mask edge, $H' = \mathbf{ERF.H} / (1 - 0.44)$, and **ERF.LBB** and **ERF.H** are user-supplied empirical expressions. The default values of **ERF.LBB** and **ERF.H** are:

$$\mathbf{ERF.LBB} = 8.25 \times 10^{-3}(1580.3 - T_{ox})\,y_\infty^{0.67}\,\mathbf{INITIAL}^{0.3} \times \exp\left(-\frac{(\mathbf{NIT.THIC} - 0.08)^2}{0.06}\right)$$

Equation 3-396

$$\mathbf{ERF.H} = 402 \times (0.445 - 1.75 \times \mathbf{NIT.THIC})e^{-T_{ox}/200}$$

Equation 3-397

where:

- T_{ox} is the oxidation temperature (in degrees Kelvin).
- y_∞ is the unmasked oxide thickness.
- **INITIAL** and **NIT.THIC** are the pad oxide and nitride mask thicknesses, respectively.

ERF2 Model

The **ERF2** model is used when the nitride is thick compared to the pad oxide thickness. The oxidation rate as a function of x for the **ERF2** model is given by:

$$\frac{dy(x)}{dt} = \begin{cases} H' \ \operatorname{erfc}\left(\dfrac{x_o - \delta - x}{\mathbf{ERF.LBB}}\sqrt{\ln 10}\right) & x < x_o - \delta \\ \dfrac{x - (x_o - \delta) + \dfrac{H' \ \mathbf{ERF.Q}}{1 - H'}}{x - (x_o - \delta) + \dfrac{\mathbf{ERF.Q}}{1 - H'}} & x \geq x_o - \delta \end{cases}$$

Equation 3-398

where:

- **ERF.LBB** is a parameter on the **AMBIENT** statement (with a default value of 0.05 microns).
- H', x_o, and **ERF.LBB** have the same meanings and values as for the **ERF1** model.

The value used for δ depends on the y coordinate of the point under consideration:

$$\delta = \begin{cases} 0 & y \leq -10^{-5} \text{ microns} \\ \mathbf{ERF.DELT} & y > -10^{-5} \text{ microns} \end{cases}$$

Equation 3-399

where **ERF.DELT** is a parameter on the **AMBIENT** statement (with a default value of 0.04 microns). Thus the breakpoint in Equation 3-398 occurs at **MASK.EDG** when calculating the position of the top surface of the oxide ($y \leq -10^{-5}$ microns), but is offset by **ERF.DELT** when calculating the position of the silicon/oxide interface ($y > -10^{-5}$ microns).

ERFG Model

The **ERFG** model selects either **ERF1** or **ERF2** depending on the values of **INITIAL**, and **NIT.THIC**. **ERF1** is used if $\mathbf{NIT.THIC} \leq 2.5 \times \mathbf{INITIAL} + 0.035\mu\text{m}$, and **ERF2** is used otherwise.

Recommended Usage

The **ERFG** models provide a fast, analytical simulation of local oxidation of planar surfaces. The accuracy of this model has not been determined. (*Note* that these models differ slightly from those proposed by Guillemot et al. Guillemot computes the final oxide shape as a function of the final field oxide thickness, while TSUPREM-4 uses the same equations to calculate the oxide growth rate as a function of the field oxide thickness at each time point in the simulation.)

The **ERFG** shares all the limitations of the **ERFC** model. In addition, it places more restrictions on the initial structure and has more parameters that need to be determined. The **ERFC** model is rarely used in practical simulations.

Numerical Oxidation Models

There are four numerical models of oxidation, **VERTICAL**, **COMPRESS**, **VISCOUS**, and **VISCOELA** in TSUPREM-4. These models are designed for accurate simulation of arbitrary structures, using the masking layers present in the simulated structure. They work by solving Equations 3-377 through 3-381 directly to obtain the growth rate at each point on the oxide/silicon interface. The models differ in the way in which the oxide flow caused by volume expansion is calculated. This section describes the calculation of the oxide growth rate. Sections VERTICAL Model, p. 3-95, COMPRESS Model, p. 3-96,VISCOUS Model, p. 3-97, and VISCOELA Model, p. 3-99 discuss how the four models simulate the oxide flow.

Oxide Growth Rate

The oxide growth rate is calculated using Equations 3-377 through 3-381. Equations 3-377 gives the flux of oxidant molecules entering the oxide from the ambient gas. The gas-phase mass transfer coefficient h is given by:

$$h = \mathbf{TRANS.0} \cdot \exp\left(\frac{-\mathbf{TRANS.E}}{kT}\right) \qquad \text{Equation 3-400}$$

where **TRANS.0** and **TRANS.E** are specified on the **AMBIENT** statement with the **OXIDE** and **/AMBIENT** parameters (see Chapter 4, AMBIENT, p. 4-233). Henry's law coefficient H is specified for each oxidizing species (O_2 or H_2O) with the **HENRY.CO** parameter on the **AMBIENT** statement.

Diffusion of oxidant through the oxide is modeled by Equation 3-378. The diffusivity is calculated from the parabolic oxidation rate B using Equations 3-383 and 3-384:

$$D = B\frac{N_1}{2C^*} \qquad \text{Equation 3-401}$$

where N_1 is given by:

$$N_1 = \begin{cases} \mathbf{THETA} & \text{for } O_2 \\ 2 \times \mathbf{THETA} & \text{for } H_2O \end{cases} \qquad \text{Equation 3-402}$$

and **THETA** is a parameter on the **AMBIENT** statement. The parabolic oxidation rate B is given by Equation 3-390 in Analytical Oxidation Models, p. 3-87.

The diffusivity in other materials is given by:

$$D = \mathbf{D.0} \cdot \exp\left(\frac{-\mathbf{D.E}}{kT}\right) \qquad \text{Equation 3-403}$$

where **D.0** and **D.E** are parameters on the **AMBIENT** statement.

The rate at which the oxidant molecules react with silicon at the oxide/silicon interface is given by Equation 3-379. The reaction rate k_s is derived from Equation 3-383:

$$k_s = \frac{B}{A}\frac{N_1}{C^*}$$

Equation 3-404

where it is assumed that $k \ll h$. The value of B/A is the linear growth rate given by Equation 3-387 in Analytical Oxidation Models, p. 3-87. For polycrystalline materials, this rate is enhanced by the presence of grain boundaries as described in Dependence of Polysilicon Oxidation Rate on Grain Size, p. 3-168.

Concentration Dependence

The electron concentration dependence l_C of the linear oxidation rate is given by References [25], [26], and [27]:

$$l_C = \begin{cases} 1 & \text{analytical oxidation models} \\ 1 + \gamma_V (C_V - 1) & \text{numerical oxidation models} \end{cases}$$

Equation 3-405

where:

$$\gamma_V = \mathbf{\mathtt{GAMMA.0}} \cdot \exp\left(\frac{-\mathbf{\mathtt{GAMMA.E}}}{kT}\right)$$

Equation 3-406

$$C_V = \frac{1 + C^+\left(\frac{n_i}{n}\right) + C^-\left(\frac{n}{n_i}\right) + C^=\left(\frac{n}{n_i}\right)^2}{1 + C^+ + C^- + C^=}$$

Equation 3-407

The parameters **`GAMMA.0`** and **`GAMMA.E`** are specified in the **`AMBIENT`** statement for each of the oxidizing species; n is the electron concentration in the silicon at the oxidizing interface. The terms C^+, C^-, and $C^=$ are the normalized intrinsic concentrations of positive, negative, and double negative vacancies, respectively, given by:

$$C^+ = \exp\left(\frac{E^+ - E_i}{kT}\right)$$

Equation 3-408

$$C^- = \exp\left(\frac{E_i - E^-}{kT}\right)$$

Equation 3-409

$$C^= = \exp\left(\frac{2E_i - E^- - E^=}{kT}\right)$$

Equation 3-410

$$E^+ = 0.35 \ eV$$

Equation 3-411

$$E^{-} = E_g - 0.57 \;\; eV \tag*{Equation 3-412}$$

$$E^{=} = E_g - 0.12 \;\; eV \tag*{Equation 3-413}$$

$$E_i = \frac{E_g}{2} + 0.75\ln(0.719)kT \;\; eV \tag*{Equation 3-414}$$

$$E_g = 1.17 - \frac{4.73 \times 10^{-4} T^2}{T + 636} \;\; eV \tag*{Equation 3-415}$$

The dependence on carrier concentration is a function of the location along the oxidizing interface, and is only used with the numerical oxidation models. The surface reaction rate depends on the local crystal orientation of the interface, which is derived from the user-specified substrate orientation and rotation.

Thin Regime

The thin regime oxidation rate constant in Equation 3-381 is given by Reference [20].

$$r_{thin} = \mathtt{THINOX.0} \cdot \exp\left(\frac{-\,\mathtt{THINOX.E}}{kT}\right) \exp\left(\frac{-y_{eff}}{\mathtt{THINOX.L}}\right) \tag*{Equation 3-416}$$

where the parameters **`THINOX.0`**, **`THINOX.E`**, and **`THINOX.L`** are specified in the **`AMBIENT`** statement (see Chapter 4, AMBIENT, p. 4-233) for each of the oxidizing species. Different values of **`THINOX.0`**, **`THINOX.E`**, and **`THINOX.L`** can be defined for each of the three available silicon orientations and for polysilicon by specifying the **`<111>`**, **`<110>`**, **`<100>`**, or **`POLYSILI`** parameters in the **`AMBIENT`** statement. The effective thickness y_{eff} at each point on the oxidizing interface is calculated as the oxide thickness required to produce the observed oxidant concentration in a 1-D solution to the Deal-Grove equations.

Usage

The numerical models have no restrictions regarding initial oxide thickness. The value of **`INITIAL`** need not correspond to the oxide thickness in the starting structure, but is still used for the thickness of the native oxide to be deposited on bare silicon or polysilicon surfaces prior to oxidation.

VERTICAL Model

The **`VERTICAL`** model is the simplest of the numerical oxidation models in TSUPREM-4. In this model, the oxide/silicon interface is constrained to move in the $+y$ direction while the expansion of the oxide occurs in the $-y$ direction. The implementation assumes that there is a single active oxide/silicon interface, with the oxide on top. All layers above the interface move with the oxide surface, and all layers below the interface remain fixed. Because the growth is only in the y direction, the growth rate depends only on the orientation of the silicon substrate (and not on the orientation of the interface).

Recommended Usage

The **VERTICAL** model is the fastest of the numerical oxidation models. It is useful for uniform oxidation with arbitrary initial oxide thickness (**INITIAL** need not be set), and for local oxidation when the initial structure is approximately planar. It is not appropriate for fully recessed oxides, trenches, or other nonplanar structures. The **VERTICAL** model can be used to model silicon-on-insulator structures, provided that only one silicon/oxide interface is being oxidized. There must be no path for oxidant to diffuse to underlying oxide layers. The **VERTICAL** model does not simulate oxidation of polysilicon.

COMPRESS Model

The **COMPRESS** model simulates viscous flow of the oxide during oxidation. The movement of the oxide/silicon interface is two dimensional (movement is perpendicular to the interface), and variation of crystal orientation along the interface is taken into account when calculating the oxidation rate. The viscous flow of the oxide is calculated in two dimensions using linear (3-node) finite elements.

The name **COMPRESS** comes from the fact that a small amount of compressibility must be allowed to ensure that the model remains numerically well-behaved. Because of the simplicity of the model and the small amount of compressibility, the model cannot be used to calculate accurate values of stress, and does not include the effects of stress on the oxidation process.

Compressible Viscous Flow

The equations governing oxide flow were derived by Chin, et al. Reference [28]. The oxide flow is described by a creeping-flow equation:

$$\mu\nabla^2\vec{V} = \vec{\nabla}P \qquad \text{Equation 3-417}$$

where:

- μ is the shear viscosity of oxide.
- $\vec{V}$ is the local velocity.
- P is the hydrostatic pressure.

The incompressibility condition can be written:

$$\vec{\nabla}\cdot\vec{V} = 0 \qquad \text{Equation 3-418}$$

To solve these equations with the 3-node element, the incompressibility condition is modified to relate the divergence of velocity to the pressure:

$$\vec{\nabla}\cdot\vec{V} = -\left(\frac{1-2v}{\mu}\right)P \qquad \text{Equation 3-419}$$

where ν, specified by **POISS.R** on the **MATERIAL** statement (Chapter 4, MATERIAL, p. 4-258), is a parameter analogous to Poisson's ratio. The viscosity is specified by analogy to an elastic model:

$$\mu = \frac{E}{2(1+\nu)} \qquad \text{Equation 3-420}$$

where E is specified by the **YOUNG.M** parameter on the **MATERIAL** statement. Equations 3-417, 3-419, and 3-420 are applied to all materials except single-crystal silicon, which remains fixed.

Boundary Conditions

The stress in the direction normal to a material interface is continuous across the interface, while the stress normal to a free surface is zero. There is no slippage between materials at an interface.

Model Parameters

Default values of **YOUNG.M** and **POISS.R** are provided for each material, except for aluminum and photoresist, which should not be present in the structure during oxidation; users must provide values for these parameters for user-defined materials that are present in the structure during oxidation with the **COMPRESS** model.

COMPRESS Model: Recommended Usage

The **COMPRESS** model is recommended for simulating the oxidation of general nonplanar structures and structures containing polysilicon when stress calculations are not required. Because it does not include the effects of stress on the oxidation process, the **COMPRESS** model should be used only when one or more of the following conditions is satisfied:

- Structure is planar.
- Amount of oxide grown is small.
- Exact details of the shape of the oxide are not critical.

When none of these conditions is satisfied, the **VISCOELA** or **VISCOUS** model should be used. The **COMPRESS** mode is slower than the **VERTICAL** model and has somewhat larger memory requirements.

VISCOUS Model

The **VISCOUS** model simulates viscous flow of the oxide during oxidation. The movement of the oxide/silicon interface is 2-D (movement is perpendicular to the interface), and variation of crystal orientation along the interface is taken into account when calculating the oxidation rate. The viscous flow of the oxide is calculated in two dimensions using 7-node finite elements, which allow accurate values of stress to be computed Reference [29].

Incompressible Viscous Flow

The equations and boundary conditions governing oxide flow are the same as for the **COMPRESS** model (Equations 3-417 through 3-420), except that surface tension is included (refer to Surface Tension and Reflow, p. 3-104) and a different set of parameters is used. The viscosity is specified for each material by:

$$\mu_o = \mathtt{VISC.0} \cdot \exp\left(\frac{-\mathtt{VISC.E}}{kT}\right) \qquad \text{Equation 3-421}$$

where **VISC.0** and **VISC.E** are parameters on the **MATERIAL** statement (see Chapter 4, MATERIAL, p. 4-258). The parameter ν, which determines the degree of compressibility, is specified as **VISC.X** on the **MATERIAL** statement. The default value of ν for all materials (except aluminum and photoresist) is 0.499, which produces a negligible amount of compressibility; the user must provide values for these parameters for user-defined materials that are present in the structure during oxidation with the **VISCOUS** model.

The stresses are calculated from the constitutive equations for each material:

$$\sigma_{xx} + \sigma_{yy} = \frac{\mu}{\frac{1}{2} - \nu}\left(\frac{\partial u_x}{\partial x} + \frac{\partial u_y}{\partial y}\right) \qquad \text{Equation 3-422}$$

$$\sigma_{xx} - \sigma_{yy} = 2\mu\left(\frac{\partial u_x}{\partial x} - \frac{\partial u_y}{\partial y}\right) \qquad \text{Equation 3-423}$$

$$\sigma_{xy} = \mu\left(\frac{\partial u_x}{\partial y} + \frac{\partial u_y}{\partial x}\right) \qquad \text{Equation 3-424}$$

where u_x and u_y are the x and y components of velocity, and σ_{xx}, σ_{yy}, and σ_{xy} are the components of the stress tensor. The stresses in a purely viscous model are proportional to the flow velocities, and thus fall to zero when oxidation stops. The stresses reported by TSUPREM-4 after an oxidation step are the values calculated at the end of the step, before the flow is stopped. The stresses calculated by the **VISCOUS** model replace any stresses that may have been previously calculated using the **ST.HISTO** model or the **STRESS** statement.

Stress Dependence

When **STRESS.D** is set true on the **AMBIENT** statement, the surface reaction rate k_s, diffusivity of oxidant in oxide D, and the oxide viscosity μ are modified to reflect their dependence on the stresses in the oxide:

$$k'_s = k_s \exp\left(-\frac{\sigma_n \mathtt{VR}}{kT}\right) \exp\left(-\frac{\sigma_t \mathtt{VT}}{kT}\right) \qquad \text{Equation 3-425}$$

$$D' = D \exp\left(-\frac{p\mathtt{VD}}{kT}\right) \qquad \text{Equation 3-426}$$

$$\mu = \mu_o \frac{\sigma_s \mathrm{VC} / 2kT}{\sinh(\sigma_s \mathrm{VC} / 2kT)} \qquad \text{Equation 3-427}$$

where k'_s , D' , and μ are the stress-dependent reaction rate, diffusivity, and viscosity, respectively, and **VR**, **VT**, **VD**, and **VC** are parameters on the **AMBIENT** statement. The exponential in Equation 3-426 is limited for positive arguments (i.e., negative values of p) to the value of **VDLIM** (a parameter on the **AMBIENT** statement) to prevent unrealistic enhancement of the diffusivity. The surface reaction rate depends both on the stress normal to the interface:

$$\sigma_n = -(\sigma_{xx} n_x^2 + \sigma_{yy} n_y^2 + 2\sigma_{xy} n_x n_y) \qquad \text{Equation 3-428}$$

and on the stress in the plane perpendicular to the interface:

$$\sigma_t = -(\sigma_{xx} n_y^2 + \sigma_{yy} n_x^2 - 2\sigma_{xy} n_x n_y) \qquad \text{Equation 3-429}$$

where n_x and n_y are the components of the unit vector normal to the interface. The oxidant diffusivity depends on the hydrostatic pressure p defined by:

$$p = -\frac{1}{2}(\sigma_{xx} + \sigma_{yy}) \qquad \text{Equation 3-430}$$

while the oxide viscosity depends on the total shear stress σ_s:

$$\sigma_s = \sqrt{\frac{1}{4}(\sigma_{xx} - \sigma_{yy})^2 + \sigma_{xy}^2} \qquad \text{Equation 3-431}$$

Newton-Raphson iteration is used to solve the nonlinear equations produced by the stress dependences. Many iterations are usually required for convergence, and in some cases full convergence is not be obtained. Thus, the time required to simulate stress-dependent oxidation may be 20-200 times that required without the stress dependences.

Recommended Usage

The **VISCOUS** model has been made largely obsolete by the **VISCOELA** model. It is more accurate than the **VISCOELA** model when the viscosity is much smaller than Young's modulus, but is much slower when stress dependence is used. It is occasionally useful in verifying the results of the **VISCOELA** model, but it is often difficult to determine whether the difference between the two models is due to the differences in the physical model or to differences in numerical approach.

VISCOELA Model

The **VISCOELA** model simulates viscoelastic flow of the oxide during oxidation. The movement of the oxide/silicon interface is 2-D (movement is perpendicular to the interface), and variation of crystal orientation along the interface is taken into account when calculating the oxidation rate. The viscoelastic flow of the oxide is calculated in two dimensions by using three-node finite elements, which use numerical techniques that allow approximate values of stress to be computed. The model is similar to that developed by Senez, et al., References [30] and in Appendix A.

Viscoelastic Flow

The **`VISCOELA`** model adds an elastic component to the equations used by the **`COMPRESS`** and **`VISCOUS`** models. In one dimension, the stress σ is related to the strain ε by:

$$\frac{\partial\sigma}{\partial t} = 2G\frac{\partial\varepsilon}{\partial t} - \frac{\sigma}{\tau} \qquad \text{Equation 3-432}$$

where:

- G is the modulus of elasticity in shear.
- $\tau = \mu/G$ is the stress relaxation time (μ is the viscosity).

This states that the change in stress is proportional to the change in strain, minus a relaxation term that is proportional to the stress. In three dimensions, the equations are written most simply in terms of volumetric (dilatory) and shear components:

$$\begin{aligned} \frac{\partial\sigma_v}{\partial t} &= 3K\frac{\partial\varepsilon_v}{\partial t} - \frac{\sigma_v}{\tau_v} \quad , \quad \tau_v = \frac{\mu_v}{K} \\ \frac{\partial\vec{\sigma}'}{\partial t} &= 2G\frac{\partial\vec{\varepsilon}'}{\partial t} - \frac{\vec{\sigma}'}{\tau'} \quad , \quad \tau' = \frac{\mu}{G} \end{aligned} \qquad \text{Equation 3-433}$$

where:

$$\sigma_v \equiv \frac{1}{3}(\sigma_{xx} + \sigma_{yy} + \sigma_{zz}) \quad \text{and} \quad \varepsilon_v \equiv \frac{1}{3}(\varepsilon_{xx} + \varepsilon_{yy} + \varepsilon_{zz}) \qquad \text{Equation 3-434}$$

are the volumetric components of the stress and strain, respectively, and:

$$\vec{\sigma}' \equiv \begin{bmatrix} \sigma_{xx} - \sigma_v \\ \sigma_{yy} - \sigma_v \\ \sigma_{zz} - \sigma_v \\ \sigma_{xy} \end{bmatrix} \quad \text{and} \quad \vec{\varepsilon}' \equiv \begin{bmatrix} \varepsilon_{xx} - \varepsilon_v \\ \varepsilon_{yy} - \varepsilon_v \\ \varepsilon_{zz} - \varepsilon_v \\ \varepsilon_{xy} \end{bmatrix} \qquad \text{Equation 3-435}$$

are the shear components.

The strains are defined by:

$$\begin{gathered} \varepsilon_{xx} \equiv \frac{\partial u_x}{\partial x} \qquad \varepsilon_{yy} \equiv \frac{\partial u_y}{\partial y} \qquad \varepsilon_{zz} \equiv \frac{\partial u_z}{\partial y} \\ \varepsilon_{xy} \equiv \frac{1}{2}\left(\frac{\partial u_x}{\partial y} + \frac{\partial u_y}{\partial x}\right) \end{gathered} \qquad \text{Equation 3-436}$$

where u_x, u_y, and u_z are the components of the flow displacements. *Note* that u_z and ε_{zz} are zero for the 2-D case analyzed in TSUPREM-4.

The stresses must satisfy the force balance equations:

$$\frac{\partial\sigma_{xx}}{\partial x}+\frac{\partial\sigma_{xy}}{\partial y}=0$$
$$\frac{\partial\sigma_{yy}}{\partial y}+\frac{\partial\sigma_{xy}}{\partial x}=0$$

Equation 3-437

These equations can be combined with Equation 3-433 and solved for the flow velocities. The stress in the direction normal to a material interface is continuous across the interface, while the stress normal to a free surface is zero (unless surface tension is included). (See Surface Tension and Reflow, p. 3-104.) There is no slippage between mat.erials at an interface. The equations are solved in silicon only if **^SKIP.SIL** has been specified on the **METHOD** statement.

Model Parameters

The **VISCOELA** model uses the same elasticity parameters as the **COMPRESS** model and the same viscosity and stress dependence parameters as the **VISCOUS** model. The bulk modulus K and shear modulus G are given by:

$$K=\frac{E}{3(1-2\nu)}\quad,\quad G=\frac{E}{2(1+\nu)}$$

Equation 3-438

where E = **YOUNG.M** and ν = **POISS.R** are specified on the **MATERIAL** statement.

The bulk viscosity μ_v and shear viscosity μ are given by:

$$\mu_v=\frac{2\mu(1+\nu_v)}{3(1-2\nu_v)}\quad,\quad \mu=\mathbf{VISC.0}\cdot\exp\left(\frac{-\mathbf{VISC.E}}{kT}\right)$$

Equation 3-439

where **VISC.0**, **VISC.E**, and ν_v = **VISC.X** are specified on the **MATERIAL** statement. *Note* that ν_v is normally very slightly less than 0.5, so μ_v is much larger than K and τ_v is much larger than typical oxidation times. Thus the relaxation in volumetric stress is negligible.

Stress Dependence

The oxidant diffusivity, interface reaction rate, and material viscosities depend on stress in the same way as in the **VISCOUS** model except that Von Mises shear stress is used when computing material viscosities and that the user-defined multiplication factors can be applied to diffusivity or surface reaction rate. Stress dependence is enabled by default; it can be disabled by specifying **^STRESS.D** on the **AMBIENT** statement.

$$k'_s=\mathbf{R.FACTOR}\cdot k_s\exp\left(-\frac{\sigma_n\mathbf{VR}}{kT}\right)\exp\left(-\frac{\sigma_t\mathbf{VT}}{kT}\right)$$

Equation 3-440

$$D' = \mathbf{D.FACTOR} \cdot D \exp\left(-\frac{p\mathtt{VD}}{kT}\right)$$ Equation 3-441

R.FACTOR and **D.FACTOR** are the expressions of the multiplication factors which can be defined for each oxidant in the **AMBIENT** statement.

The linear elements used in the **VISCOELA** model produce stresses that are constant across each element and discontinuous between elements. In order to calculate the stress dependences, these discontinuous stresses must be smoothed. The amount of smoothing is controlled by the **VE.SMOOT** parameter on the **METHOD** statement. **VE.SMOOT** can be varied between 0.0 (minimum smoothing) and 1.0 (maximum smoothing). With smaller amounts of smoothing, the stress contours become rougher; larger amounts give smoother contours, but may lose some detail in the solution.

Temperature-dependent Anisotropic Elastic Constants

The anisotropic elasticity model is applied by turning on **ANISOEL** parameter in the **MATERIAL** statement. For example:

```
MATERIAL SILICON ANISOEL
```

The anisotropic elasticity model uses the elasticity tensor C instead of Young’s modulus and Poisson’s ratio. For anisotropic elastic materials, the constitutive equations are expressed in:

$$\begin{bmatrix}\sigma_{xx}\\ \sigma_{yy}\\ \sigma_{zz}\\ \sigma_{yz}\\ \sigma_{xz}\\ \sigma_{xy}\end{bmatrix} = C' \begin{bmatrix}\varepsilon_{xx}\\ \varepsilon_{yy}\\ \varepsilon_{zz}\\ 2\varepsilon_{yz}\\ 2\varepsilon_{xz}\\ 2\varepsilon_{xy}\end{bmatrix}$$ Equation 3-442

The transformed elasticity tensor C’ is calculated as follows:

$$C' = RCR^T$$ Equation 3-443

where the 6-x-6 rotation matrix R which is composed of 4 3-x-3 matrix $R^{(i)}$, is computed as:

$$R = \begin{bmatrix} R^{(1)} & R^{(2)} \\ R^{(3)} & R^{(4)} \end{bmatrix}$$ Equation 3-444

$$R_{ij}^{(1)} = \chi_{ij}^2$$ Equation 3-445

$$R_{ij}^{(2)} = 2\chi_{i\ mod(j+1,\ 3)}\chi_{i\ mod(j+2,\ 3)}$$ Equation 3-446

$$R_{ij}^{(3)} = \chi_{mod(i+1,\ 3)\ \mathrm{j}}\chi_{mod(i+2,\ 3)\ \mathrm{j}} \qquad \text{Equation 3-447}$$

$$\begin{aligned} R_{ij}^{(4)} &= \chi_{mod(i+1,\ 3)\ mod(j+1,\ 3)}\chi_{mod(i+2,\ 3)\ mod(j+2,\ 3)} \\ &\quad + \chi_{mod(i+1,\ 3)\ mod(j+2,\ 3)}\chi_{mod(i+2,\ 3)\ mod(j+1,\ 3)} \end{aligned} \qquad \text{Equation 3-448}$$

for the transformation matrix, χ to crystallography and $mod(i,\ j)$ which is the remainder of *i* divided by *j*.

The elasticity tensor C is given by:

$$C = \begin{bmatrix} C_{11} & C_{12} & C_{12} & 0 & 0 & 0 \\ C_{12} & C_{11} & C_{12} & 0 & 0 & 0 \\ C_{12} & C_{12} & C_{11} & 0 & 0 & 0 \\ 0 & 0 & 0 & C_{44} & 0 & 0 \\ 0 & 0 & 0 & 0 & C_{44} & 0 \\ 0 & 0 & 0 & 0 & 0 & C_{44} \end{bmatrix} \qquad \text{Equation 3-449}$$

where:

$$C_{11} = \mathbf{C11.0} + \mathbf{C11.1} \cdot \ (T - 300) \qquad \text{Equation 3-450}$$

$$C_{12} = \mathbf{C12.0} + \mathbf{C12.1} \cdot \ (T - 300) \qquad \text{Equation 3-451}$$

$$C_{44} = \mathbf{C44.0} + \mathbf{C44.1} \cdot \ (T - 300) \qquad \text{Equation 3-452}$$

for Kelvin temperature T.

The parameters **C11.0**, **C11.1**, **C12.0**, **C12.1**, **C44.0**, and **C44.1** are defined in the **MATERIAL** statement.

Recommended Usage

The **VISCOELA** model is recommended for simulating 2D structures when details of the resulting oxide shape are important or when stress values are required. It is designed to be used with **STRESS.D** enabled. (It is slower than the **COMPRESS** model, and without stress dependence it is not significantly more accurate.) It is much faster than the **VISCOUS** model, especially when stress dependence is considered. The model is designed to give a good approximation to the shape of the oxide in a minimum simulation time. While stress and flow rate information are available, these quantities are provided only as rough estimates.

The **VISCOELA** model must be used when comprehensive stress history modeling (using **ST.HISTO**) is required; see Stress History Model, p. 3-115. When **ST.HISTO** is used with the **VISCOELA** model, intrinsic and thermal mismatch stresses are included in the oxidation model and the stress model is enabled during inert anneals to model relaxation of stresses in the structure.

Anisotropic Stress Dependent Reaction

Due to crystal anisotropy, reactions across silicon-oxide interface take place at different rates, and the dependence of the reaction rates on local stress is a function of the local crystal orientation. This anisotropic stress dependence of reaction rate often leads to the formation of distinctive <111> facets at the silicon-oxide interface near the STI bottom corners. When **ST.R.ANI**is set true on the **AMBIENT** statement, the surface reaction rate k'_s in Equation 3-425 is computed using crystal orientation dependent parameters **VR**, and **VT,** where **VR** is extracted from normal activation volumes in 100, 110 and 111 crystal directions, **VR100**, **VR110**, and **VR111** respectively; and **VT** is extracted from tangential activation volumes in 100, 110 and 111 crystal directions, **VT100**, **VT110**, and **VT111** respectively. The individual activation volumes in normal and tangential directions across the reaction interface are parameters on the **AMBIENT** statement. It is applicable only to the **VISCOUS** and **VISCOELA** models.

Strength-Limited Stress

The strength limit model can be used to prevent unphysical large stress levels during the stress simulation. Once the model is turned on, the absolute stress values for each stress component are bounded within the strength limit. The **ST.LIMIT** parameter in the **METHOD** command can be used to turn on the model. The default setting for the parameter is off and the default strength limit is taken as 1/20 of the Young's modulus. The **STRENGTH** parameter can be reset by using the **MATERIAL** command. It is applicable only to the **VISCOUS** and **VISCOELA** models.

Polysilicon Oxidation

The oxidation of polysilicon is treated using the models for the oxidation of silicon. Only the **COMPRESS**, **VISCOUS**, and **VISCOELA** models allow oxidation of polysilicon. Values of **THINOX.0**, **THINOX.E**, **THINOX.L**, **L.LIN.0**, **L.LIN.E**, **H.LIN.0**, and **H.LIN.E** may be defined for polysilicon by specifying the **POLYSILI** parameter on the **AMBIENT** statement. The ratio of polysilicon thickness consumed to oxide thickness grown can be defined with the **ALPHA** parameter on the **AMBIENT** statement.

Surface Tension and Reflow

The **VISCOUS** and **VISCOELA** models include the effects of surface tension. The surface tension is modeled as a tensile force along the surface of magnitude specified by the **SURF.TEN** parameter on the **MATERIAL** statement. On planar surfaces, the forces on either side of any point of the surface cancel, giving no net force. But on curved surfaces there is a net force in the direction of the curvature with magnitude inversely proportional to the radius of curvature. The resulting force tends to round off corners and smooth irregularities in the surface. With appropriate values of **SURF.TEN** and material viscosities, the surface tension model can be used to simulate reflow processes.

The surface tension model is active during oxidation with the **`VISCOUS`** and **`VISCOELA`** model and whenever stress history modeling is active (i.e., when both **`VISCOELA`** and **`ST.HISTO`** are used).

N_2O Oxidation

In N_2O oxidation or oxynitridation, nitrogens are trapped at Si/SiO_2 interfaces so that the number of oxidizing sites, for this reason the oxidation rate is reduced. N_2O oxidation is specified by the **`N2O`** parameter in the **`DIFFUSION`** statement.

Nitrogen Trap and Generation

It is assumed that N_2O oxidants reach to Si/SiO_2 interfaces then break up there so that nitrogens are generated. Some of nitrogen are trapped at the interfaces. The trapping flux of nitrogen is described with the interface trap model by ignoring the detrapping flux.

$$\vec{F}_{trap} = hC_i\left(1 - \frac{\sigma_N}{\sigma_{max}}\right)\vec{n}_i \qquad \text{Equation 3-453}$$

where h is the trapping rate, C_i is the nitrogen concentration at the interface. σ_N and σ_{max} are the trapped density and the maximum trap density, respectively. The coefficients h and σ_{max} are specified in the **`IMPURITY`** statement (See Segregation Flux, p. 3-54). The total nitrogen flux at interfaces is the sum of the trapping flux into interfaces and the 2-phase segregation flux into adjacent materials. The generation rate of nitrogen is dependent on the boundary velocity and is given by:

$$G = \mathbf{N.G.0}\exp\left(-\frac{\mathbf{N.G.E}}{kT}\right)v\left(\frac{v}{\mathbf{N.V.0}\exp\left(-\frac{\mathbf{N.V.E}}{kT}\right)}\right)^{gpow} \qquad \text{Equation 3-454}$$

$$gpow = \mathbf{N.GPOW.0}\exp\left(-\frac{\mathbf{N.GPOW.E}}{kT}\right) \qquad \text{Equation 3-455}$$

The **`N.G.0`**, **`N.G.E`**, **`N.V.0`**, **`N.V.E`**, **`N.GPOW.0`**, and **`N.GPOW.E`** are specified in the **`AMBIENT`** statement.

Note:

N_2O oxidation invokes the interface trap model for nitrogen regardless of the specification of the `ITRAP` parameter in the `METHOD` statement.

Surface Reaction Rate

Since the surface reaction rate is proportional to the number of available oxidizing sites, the rate of oxidant consumption at the oxidizing interface is given by:

$$\vec{F} = k_s C_i \left(1 - \frac{\sigma_N}{\mathbf{N.0} \exp\left(-\frac{\mathbf{N.E}}{kT} \right)} \right) \vec{n}_i$$

Equation 3-456

where:

- k_s is the surface reaction rate.
- C_i is the oxidant concentration at the interface.
- σ_N is the nitrogen density trapped at the interface.
- Different values of **`N.0`** and **`N.E`** can be defined for each of the three available silicon orientations and for polysilicon by specifying the **`<111>`**, **`<110>`**, **`<100>`**, or **`POLYSILI`** parameters in the **`AMBIENT`** statement.

Thin Oxidation Rate

For the thin oxidation regime, the thin oxidation rate is given by:

$$r_{thin} = \mathbf{THINOX.0} \exp\left(-\frac{\mathbf{THINOX.E}}{kT} \right) \exp\left(\frac{-y_{eff}}{\mathbf{THINOX.L}} \right)$$

$$\times \left(1 - \frac{\sigma_N}{\mathbf{N.THIN.0} \exp\left(-\frac{\mathbf{N.THIN.E}}{kT} \right)} \right)$$

Equation 3-457

Different values of **`N.THIN.0`** and **`N.THIN.E`** can be defined for each of the three available silicon orientations and for polysilicon by specifying the **`<111>`**, **`<110>`**, **`<100>`**, or **`POLYSILI`** parameters in the **`AMBIENT`** statement.

Boron Diffusion Enhancement in Oxides

Ultrathin gate oxides allow easier dopant penetration between polysilicon and bulk silicon. Boron penetrates more easily than most other dopants because it has a larger diffusivity than the others do. A nonstoichiometric layer rich in peroxy-linkage-defect forms when oxide is grown on top of the silicon Reference [31]. Dopant diffusion is enhanced in thinner oxides because of the greater ratio of nonstoichiometric layer thickness to the total oxide thickness. Also fluorine increases the number of peroxy-linkage-defects and thus enhances dopant diffusion.

Diffusion Enhancement in Thin Oxides

It is assumed that the formation of silicon-oxide (SiO) due to the reaction between oxygen and silicon atom increases the number of peroxy-linkage-defects. The interstitial segregation at silicon/oxide interfaces introduces silicon atoms into the oxide. The SiO is assumed to be immobile and to react with oxygen to produce SiO_2 to rarely dissolve into SiO and O. The volume expansion due to this SiO_2 formation in bulk is assumed to be negligible. The diffusivity of impurities depends on the distribution of SiO.

$$\frac{\partial C_{Si}}{\partial t} = -\vec{\nabla} \cdot \left(-D_{Si}\vec{\nabla} C_{Si}\right) - k_{Si}C_{Si}\sqrt{C_{O_2}} \qquad \text{Equation 3-458}$$

$$\frac{\partial C_{SiO}}{\partial t} = k_{Si}C_{Si}\sqrt{C_{O_2}} - k_{SiO}C_{SiO}\sqrt{C_{O_2}} \qquad \text{Equation 3-459}$$

$$F_{Si} = h_{Si}(C_{Si} - mI) \quad \text{at silicon/oxide interfaces} \qquad \text{Equation 3-460}$$

$$F_{Si} = k_i C_{Si} \quad \text{at oxide surfaces} \qquad \text{Equation 3-461}$$

C_{Si}, C_{SiO} and I are the concentrations of silicon atom, SiO in oxide, and interstitial in silicon, respectively. The diffusivity D_{Si} of silicon atom in oxide is set to $13.0\exp(-4.5\mathrm{eV}/\ kT)$ Reference [32]. m is the segregation value of interstitial at the interface between silicon and oxide. Its value is set to $2.9 \times 10^{-5}\exp(2.19\mathrm{eV}/\ kT)$ corresponding to the value mC_I^* of Agarwal's work Reference [32]. k_{Si} and k_{SiO} are the bulk reaction rates of silicon atom and SiO with oxygen, respectively. k_i is the reaction rate of silicon atom with oxygen at oxide surfaces.

The diffusion enhancement in oxide is described as follows:

$$D = D_o\left(1 + f_{SiO}\frac{C_{SiO}}{C_{SiO_2}}\right) \qquad \text{Equation 3-462}$$

D_0 is a diffusivity in pure thick oxides. f_{SiO} is the temperature-dependent factors for the diffusion-enhancement due to SiO. C_{SiO_2} is the constant for normalization (2.2×10^{22} /cm^3).

The activation energy of f_{SiO} is set to -0.28 eV Reference [31]. The experimental data for calibration were taken from the published paper Reference [33]. It is assumed that K_{SiO} has the same value as K_{Si} in order to avoid the redundancy due to the lack of measured data. The new impurity names for silicon atom and SiO are given by **IOX** and **SIO**, respectively.

```
IMPURITY MODEL=DEOX IMP=IOX NEW C.INIT=1E5
IMPURITY MODEL=DEOX IMP=SIO NEW C.INIT=1E5
METHOD   IMP=IOX PART REL.ERR=0.01 ABS.ERR=1E9
METHOD    IMP=SIO NONE REL.ERR=0.01 ABS.ERR=1E9
```

The diffusivities of silicon atom and boron in oxide is given by

```
IMPURITY IMP=IOX MAT=OXIDE +
         DIX.0=13.0 DIX.E=4.5 CM.SEC
IMPURITY D.MODEL=DEOX IMP=BORON MAT=OXIDE +
       DI.FAC=1+4.55e-22*exp(0.28/kt)*SIO
```

The bulk reactions are given by

```
REACTION MODEL=DEOX                      +
         NAME=KbIOXnO MAT.BULK=OXIDE     +
         IMP.L=IOX /IMP.L=O2  IMP.R=SIO  +
         EI.L=1.0  /EI.L=0.5  EI.R=1.0   +
         NI.L=1.0  /NI.L=0.0  NI.R=1.0   +
         RATE.0=7.673E6 RATE.E=4.0 EQUIL.0=0.0
REACTION MODEL=DEOX                      +
         NAME=KbSiOnO MAT.BULK=OXIDE     +
         IMP.L=SiO /IMP.L=O2             +
         EI.L=1.0  /EI.L=0.5             +
         NI.L=1.0  /NI.L=0.0             +
         RATE.0=7.673E6 RATE.E=4.0 EQUIL.0=0.0
```

The segregation flux of interstitial is given by:

```
REACTION MODEL=DEOX                  +
         NAME=ISEG                   +
         /MAT.L=OXIDE MAT.R=SILICON  +
         /IMP.L=IOX   IMP.R=INTERST  +
         RATE.0=0.01 RATE.E=0.0      +
         EQUIL.0=2.9E-5 EQUIL.E=-2.19
```

The surface reaction rate of silicon atom is given by:

```
REACTION MODEL=DEOX                  +
         NAME=KiIOX                  +
         /MAT.L=OXIDE MAT.R=AMBIENT  +
         /IMP.L=IOX   IMP.R=O2       +
         /EI.L=1.0    EI.R=1.0       +
         /NI.L=1.0    NI.R=0.0       +
         RATE.0=8.67E7 RATE.E=4.0 EQUIL.0=0.0
```

Since the model `DEOX` is initially disabled in *s4init* file, it needs to be activated to use the model by:

```
METHOD MODEL=DEOX ENABLE
```

The SiO distribution is believed to be mostly concentrated near the Si/SiO_2 interface and thus the dense grid structure is needed near the interface. Usually it is believed that the native oxide thickness must be thinner than the default value 20 angstroms.

```
METHOD DY.OXIDE=0.0005
AMBIENT INITIAL=0.0005
```

Note:

This model is only applicable to dry oxidation. And the accuracy of this model is not verified for the oxidation on low oxygen pressure.

Diffusion Enhancement Due to Fluorine

Fluorine enhances the boron diffusion in oxides. For the diffusion model of fluorine, fluorine is assumed to be electrically neutral. The diffusion enhancement due to fluorine is added to Equation 3-462.

$$D = D_o\left(1 + f_{SiO}\frac{C_{SiO}}{C_{SiO_2}} + f_F\frac{C_F}{C_{SiO_2}}\right) \qquad \text{Equation 3-463}$$

where f_F is an equilibrium constant and C_F is a fluorine concentration. f_F and the diffusivities and segregation values of fluorine were extracted by fitting the published data Reference [34].

```
IMPURITY D.MODEL=DEOX IMP=BORON MAT=OXIDE +
 DI.FAC=1+4.55e-22*exp(0.28/kt)*SIO+2.23e-20*Fluorine
```

In order to consider the fluorine effect, the model `FLUORINE` as well as the model `DEOX` must be activated as follows:

```
METHOD MODEL=DEOX ENALE
METHOD MODEL=FLUORINE ENABLE
```

If the model `FLUORINE` is enabled while the model `DEOX` is disabled, the diffusion enhancement of boron due to fluorine is ignored even though the diffusion equation for fluorine is solved.

Silicide Models

TSUPREM-4 allows you to define models for new materials and reactions. This ability has been used to define models for the growth of titanium and tungsten silicides. The following sections describe the kinetics of $TiSi_2$ growth, the specification of the model and parameters, and suggestions for how you can model other silicides.

$TiSi_2$ Growth Kinetics

Titanium silicide is assumed to form when silicon atoms react in the silicide with titanium at the $TiSi_2$/Ti interface. The consumption of silicon and titanium lead to deformation of the material layers in the structure. *Note* that while the discussion that follows describes the growth of $TiSi_2$ on silicon, it also applies to growth of $TiSi_2$ on polycrystalline silicon.

Reaction at $TiSi_2$/Si Interface

At the $TiSi_2$/Si interface you have the reaction:

$$\mathrm{Si(Si)} \rightleftarrows \mathrm{Si(TiSi_2)} + a\mathrm{V(Si)} \qquad \text{Equation 3-464}$$

Thus silicon (on the Si side of the interface) reacts to form silicon atoms (on the $TiSi_2$ side of the interface) plus some number a of vacancies (on the Si side of the interface). The reaction is reversible, allowing the reformation of silicon (if silicon is released by nitridation of $TiSi_2$, for example). The forward rate of this reaction depends only on temperature, while the reverse rate is also proportional to the concentration of diffusing silicon atoms in the $TiSi_2$. For each silicon atom removed from the silicon side of the interface, the volume of silicon is reduced by:

$$\Delta V = \frac{\mathtt{MOL.WT}}{6.022 \times 10^{23} \times \mathtt{DENSITY}} \qquad \text{Equation 3-465}$$

where **`MOL.WT`** and **`DENSITY`** are the molecular weight and density, respectively, of silicon, as specified on the **`MATERIAL`** statement.

Diffusion of Silicon

Silicon is transported across the $TiSi_2$ layer by simple diffusion:

$$\frac{\partial C}{\partial t} = -\vec{\nabla} \cdot \left(-D\vec{\nabla} C\right) \qquad \text{Equation 3-466}$$

where C is the concentration of diffusion silicon atoms and D is their diffusivity.

Reaction at $TiSi_2$/Si Interface

At the $TiSi_2$/Ti interface you have the reaction:

$$2\mathrm{Si(TiSi_2)} + \mathrm{Ti(Ti)} \rightleftarrows \mathrm{TiSi_2(TiSi_2)} \qquad \text{Equation 3-467}$$

This reaction is assumed to be irreversible. The forward reaction rate is proportional to the concentration of diffusing silicon at the $TiSi_2$ side of the interface. The volumes of Ti and $TiSi_2$ change according to Equation 3-465.

Oxygen Dependence

The silicidation process may be influenced by the presence of oxygen in the silicide[35]. This oxygen is assumed to enter the silicide at interfaces with silicon dioxide and diffuse in the silicide according to Fick's law. The oxygen retards the reaction of silicon atoms at the silicide/silicon and silicide/metal interfaces and the diffusion of silicon in the silicide. The retardation factor is assumed to be of the form $1 - \min(C_O / C_{Ocrit}, 1)$, where C_O is the oxygen concentration in the silicide and C_{Ocrit} is a critical oxygen concentration for the reduction.

Initialization

The $TiSi_2$ growth model is initialized by inserting a thin layer of titanium silicide between layers of titanium and silicon (or polysilicon) wherever they come into contact. This layer is added automatically as needed when titanium is deposited.

Material Flow

Consumption of silicon and titanium and growth of $TiSi_2$ cause distortion of the layers making up the structure. The flow of material caused by silicidation is calculated using the **VERTICAL**, **COMPRESS**, or **VISCOELA** growth models described in VERTICAL Model, p. 3-95, the COMPRESS Model, p. 3-96, and the VISCOELA Model, p. 3-99; silicide growth occurs only if one of these models has been specified.

Impurities and Point Defects

Impurities in silicides are modeled in the same way as in other nonsemiconductor materials. Transport within a silicide is governed by simple diffusion (i.e., no electric field effects). Segregation at material interfaces is as described in Segregation of Impurities, p. 3-54.

Point defects can participate in reactions at interfaces with silicon. The current titanium silicide model specifies the generation of vacancies by the consumption of silicon (Equation 3-464). The reaction rate has not yet been adequately calibrated to experimental data, however; it is suggested that the **PD.FERMI** model be used for silicide growth processes.

Specifying Silicide Models and Parameters

The specification of the titanium silicide model requires the following:

- The new materials (titanium and $TiSi_2$ in this case) must be defined.
- Any diffusing species that participates in the growth reactions must be defined.
- The growth reactions themselves (one at each interface) must be defined.
- The deposition of initial layers (e.g., the initial $TiSi_2$ layer between titanium and silicon) must be specified.
- The diffusion and segregations of impurities in the new materials and at interfaces must be specified.

All of these are accomplished with TSUPREM-4 input statements. Thus no new code is required to implement new silicide models.

Materials

The required materials are specified with the **MATERIAL** statement:

```
MATERIAL NEW MAT=TITANIUM TIF.NAME=TI +
     MD.INDEX=-5  DENSITY=4.5  AT.NUM=22.0 +
     AT.WT=47.90  MOL.WT=47.90

MATERIAL NEW MAT=TISI2  TIF.NAME=TISI2 +
     MD.INDEX=-5  DENSITY=4.043 AT.NUM=16.67 +
     AT.WT=34.68  MOL.WT=104.038   DY.DEFAU=0.025
```

For purposes of defining a silicide growth model, the important parameters are the names of the material (the **MATERIAL** parameter) and the density and molecular weight (**DENSITY** and **MOL.WT**, respectively). The atomic number and atomic weight (**AT.NUM** and **AT.WT**) are used when implanting into the material with the MC implant model. The default grid spacing in a growing material is given by **DY.DEFAU** × **GRID.FAC**. A full description of the **MATERIAL** statement is given in Chapter 4, MATERIAL, p. 4-258.

Impurities

For the titanium silicide model, you define silicon as a diffusing impurity:

```
IMPURITY NEW IMP=SILICON TIF.NAME=SI STEADY
IMPURITY IMP=SILICON MAT=TISI2 DIX.0=2.0 DIX.E=1.86 +
    CM.SEC
```

The important parameters here are the pre-exponential factor and activation energy (**DIX.0** and **DIX.E**, respectively) for the diffusivity of silicon in $TiSi_2$. The **STEADY** parameter specifies that the equation for silicon diffusion is solved under steady-state conditions; this is appropriate for fast-diffusing impurities and is recommended for impurities that produce material growth.

Reduction of the silicidation rate due to oxygen requires another impurity:

```
IMPURITY MODEL=ORSILICIDE NEW IMP=ORSOXYGEN +
    TIF.NAME=ORSOXYGEN ^STEADY
METHOD IMP=ORSOXYGEN LU PART ^SYMM TIME.STE +
    REL.ERR=0.01 ABS.ERR=1.0E9
IMPURITY IMP=ORSOXYGEN MAT=TISI2 CM.SEC +
    DIX.0=15 DIX.E=2.56
```

The impurity *ORSoxygen* is solved as a function of time (rather than in the steady state). The lateral extent of the retardation is determined by the diffusivity of *ORSoxygen* in silicide. For a complete description of the **IMPURITY** statement see Chapter 4, IMPURITY, p. 4-273.

Reactions

Two reactions are needed to model the growth of titanium silicide. The first describes the solution of silicon atoms in $TiSi_2$:

```
REACTION /MAT.L=TISI2 MAT.R=SILICON NM.R=1.0 +
      /IMP.L=SILICON IMP.L=VACANCY NI.L=1E-3 EI.L=0 +
      RATE.0=1E-3 EQUIL.0=1E20 +
      /R.FAC=MAX((1-ORSOXYGEN/1E22),0.)
```

This statement implements the reverse of the reaction of Equation 3-464:

$$\mathrm{Si(TiSi_2)} + 10^{-3}\mathrm{V(Si)} \rightleftarrows \mathrm{Si(Si)} \qquad \text{Equation 3-468}$$

Parameters ending in "**.L**" denote reactants (silicon atoms in $TiSi_2$ and vacancies in silicon) while parameters ending in "**.R**" denote products (bulk silicon).

Parameters can also be classified by which side of the material interface they are associated with. In this case, the vacancies are on the silicon side of the interface (no “`/`” in the parameter name), while silicon (as an impurity) is on the $TiSi_2$ side of the interface (with “`/`” in the parameter name).

By default, one molecule of each impurity and no molecules of each material participate in the reaction. These defaults have been overridden for silicon (**`NM.R`**=1.0 implies that one silicon atom participates) and for vacancies (**`NI.L`**=1e-3 implies that one vacancy is generated for each 1000 reactions). The forward reaction rate in this case is given by

$$R_f = \mathbf{/R.FAC} \cdot \mathbf{RATE.0}\, \exp\left(\frac{-\,\mathbf{RATE.E}}{kT}\right)[\mathrm{V}]^{\mathbf{EI.L}}[\mathrm{Si}]^{\mathbf{/EI.L}} \qquad \text{Equation 3-469}$$

where `[V]` and `[Si]` denote the vacancy and silicon concentrations, respectively. **`EI.L`**=0 has been specified, so there is no dependency on the vacancy concentration; **`/EI.L`** defaults to 1.0. **`/R.FAC`** models the reduction in the reaction rate caused by oxygen in the silicide. The equilibrium concentration of reactant impurities to product impurities is given by:

$$[\mathrm{V}]^{\mathbf{EI.L}}[\mathrm{Si}]^{\mathbf{/EI.L}} = \mathbf{EQUIL.0}\, \exp\left(\frac{-\,\mathbf{EQUIL.E}}{kT}\right) \qquad \text{Equation 3-470}$$

Note:

> ***There are no product impurities in this case. The equilibrium concentration of silicon diffusing in*** $TiSi_2$ ***has been specified as 1e20.***

For a complete description of the **`REACTION`** statement see Chapter 4, REACTION, p. 4-300.

The reaction at the interface between $TiSi_2$ and polysilicon is the same as at silicon interfaces except that no vacancies are involved:

```
REACTION /MAT.L=TISI2 MAT.R=POLY   NM.R=1.0 +
       /IMP.L=SILICON RATE.0=1E-3 EQUIL.0=1E20
       /R.FAC=MAX((1-ORSOXYGEN/1E22),0.)
```

The reaction at the Ti/$TiSi_2$ interface is specified by

```
REACTION /MAT.L=TITANIUM /NM.L=0.5 +
       MAT.R=TISI2 NM.R=0.5 IMP.L=SILICON +
       RATE.0=104  RATE.E=1.0 EQUIL.0=1.0
       R.FAC=MAX((1-ORSOXYGEN/1E22),0.)
```

This corresponds directly with Equation 3-467 except that you have divided the quantities of all reactants by two. The forward reaction rate is proportional to the silicon concentration at the Ti/$TiSi_2$ interface. The equilibrium ratio of reactant impurities to product impurities (= `[Si]`) has been set to 1.0, a very small value. The reduction in the rate due to oxygen is specified by **`R.FAC`** rather than

/R.FAC because in this reaction the silicide is specified as **MAT.R** (instead of **/MAT.L**).

The initial $TiSi_2$ layers to be deposited between titanium layers and exposed silicon and poly silicon layers are also specified with **REACTION** statements:

```
REACTION MAT=TITANIUM /MAT=SILICON +
      MAT.NEW=TISI2 THICK=0.002
REACTION MAT=TITANIUM /MAT=POLY +
      MAT.NEW=TISI2 THICK=0.002
```

These specify that 0.002 microns of $TiSi_2$ should be deposited on silicon or polysilicon before depositing titanium.

Dopants

Diffusion and segregation of impurities in silicides are specified in the usual way:

```
BORON MAT=TISI2 CM.SEC DIX.0=6.0E-7 DIX.E=2.0
BORON SILICON   /MAT=TISI2 SEG.0=0.3 +
     TRANS.0=1E-6 TRANS.E=2.0
BORON MAT=POLY   /MAT=TISI2 SEG.0=0.3 +
     TRANS.0=1E-6 TRANS.E=2.0
```

The **MATERIAL** and **/MATERIA** parameters must be used when specifying new material names; they are optional when specifying old built-in materials. These parameters could also be specified with the **IMPURITY** statement:

```
IMPURITY IMP=BORON MAT=TISI2 CM.SEC DIX.0=6.0E-7 +
     DIX.E=2.0
IMPURITY IMP=BORON MAT=SILICON /MAT=TISI2 +
     SEG.0=0.3 TRANS.0=1E-6 TRANS.E=2.0
```

Tungsten, Cobalt, and Nickel Silicide Models

The tungsten, cobalt, and nickel silicide models are identical in form to the titanium silicide model. The parameters of the models are different, however, reflecting the differences between the materials (see References [36] to [42] for the tungsten silicide model and References [43] to [47] for the cobalt silicide model). The names of the relevant materials are *tungsten* and *WSi2* for tungsten silicide, *cobalt* and *CoSi2* for cobalt silicide, and *nickel* and *NiSi2* for nickel silicide.

Other Silicides

Models for other silicides can be defined following the example given above for titanium silicide. The diffusing impurity in the silicide can be metal or silicon or both.

Stress Models

TSUPREM-4 has several models for calculating the stresses produced during the fabrication model. The most complete is the stress history (**`ST.HISTO`**) model. Older and more limited capabilities are provided by the **`STRESS`** statement and the **`VISCOUS`** oxidation model.

Stress History Model

A number of physical phenomena give rise to stress in a structure during fabrication. These include volume changes during oxidation, thermal mismatch between materials, intrinsic strain in deposited layers, and surface tension. These phenomena are simulated by the stress history (**`ST.HISTO`**) model in TSUPREM-4.

Thermal Stress Model Equations

The effect of thermal expansion during temperature ramping is modeled by adding an additional term to Equation 3-433 for the volumetric stress:

$$\frac{\partial\sigma_v}{\partial t} = 3K\left(\frac{\partial\varepsilon_v}{\partial t} - \mathbf{LCTE}\frac{dT}{dt}\right) - \frac{\sigma_v}{\tau_v} \qquad \text{Equation 3-471}$$

where **`LCTE`** is the linear coefficient of thermal expansion specified on the **`MATERIAL`** statement and T is the temperature.

Boundary Conditions

The boundary conditions assume that the thermal expansion is dominated by a thick silicon substrate with equivalent conditions on the front and back surfaces (so there is no curvature of the wafer). Thermal expansion of the substrate produces a constant strain $\varepsilon_{zz} = \mathbf{LCTE}\,\Delta T$ in the z direction. The displacement in the x direction at vertical reflecting boundaries and in silicon when **`SKIP.SIL`** is set is proportional to x, $u_x = x\mathbf{LCTE}\,\Delta T$, while the displacement in the y direction at horizontal reflecting boundaries and in silicon when **`SKIP.SIL`** is set is proportional to y, $u_y = y\mathbf{LCTE}\Delta T$. The value of **`LCTE`** for silicon is used for these boundary conditions.When solving thermal stress equations with above boundary conditions, no thermal stresses are generated in the thick silicon substrate. The expanded substrate is then used as a reference to calculate the relative thermal expansions for the layer structures above the substrate. (When saving strain components to results files, only the mechanical deformation strains are saved.}

Surface tension acts on the surface of the structure; see Surface Tension and Reflow, p. 3-104 for details.

Initial Conditions

The final stress from one high-temperature processing step is used as the starting stress for the next; if the temperature changes between the end of one step and the start of the next, a very short ramping step is inserted automatically by the program to compute the change in stress caused by the change in temperature. The stress is set to zero at the first temperature specified after structure initialization; this temperature may be specified as part of a high-temperature processing step (i.e., **`DIFFUSION`**, **`DEPOSITION`**, or **`EPITAXY`**) or on a **`SELECT`** or **`SAVEFILE`** statement. The stresses may be modified during deposition and etching, as described below. The **`STRESS`** statement and oxidation with the **`VISCOUS`** model compute new stresses, ignoring and replacing previously calculated values.

Intrinsic Stress in Deposited Layers

The intrinsic stress in deposited layers can be specified for each material with the **`INTRIN.S`** parameter on the **`MATERIAL`** statement. This intrinsic stress is included whenever stress history modeling is enabled. Each deposition step is followed by a stress relaxation calculation to determine the stresses in the structure following the deposition. The model includes the effects of surface tension; the stress is only calculated when **`ST.HISTO`** and **`VISCOELA`** models are active.

Note:

> ***The interpretation of the `INTRIN.S` parameter by the stress history model is different from that used by the `STRESS` statement (obsolete). Both work by placing an initial stress in the deposited layer then allowing the layer to relax to conform to the boundary conditions (e.g., zero stress normal to free surfaces). The `STRESS` statement uses `INTRIN.S` as the initial stress, while the stress history model uses an initial stress that gives `INTRIN.S` as the x (and z) component of stress in a uniform, planar layer after relaxation. The difference is summarized by Table 1. For the `ST.HISTO` model to generate the same results as the `STRESS` statement, you must multiply `INTRIN.S` by*** $(1-2\nu)/\ (1-\nu)$***, where*** ν ***is the value of `POISS.R`.***

Table 2-1. Model Comparisons After Relaxation

Model	Initial	After Relaxation
`ST.HISTO`	$\sigma_{xx} = \sigma_{yy} = \sigma_{zz} = \texttt{INTRIN.S}\dfrac{1-\nu}{1-2\nu}$	$\sigma_{yy} = 0,$ $\sigma_{xx} = \sigma_{zz} = \texttt{INTRIN.S}$
`STRESS (OBSLETE)`	$\sigma_{xx} = \sigma_{yy} = \sigma_{zz} = \texttt{INTRIN.S}$	$\sigma_{yy} = 0,$ $\sigma_{xx} = \sigma_{zz} = \texttt{INTRIN.S}\dfrac{1-2\nu}{1-\nu}$

Effect of Etching and Depositionon Stress

When stressed material is removed or deposited from a structure, the balance of forces acting on the remaining material changes, as do the stresses in the remaining structure. Thus, each etching or deposition step is followed by a stress relaxation calculation to determine the stresses in the structure after the etch. The model includes the effects of surface tension; the stress is only calculated when the **`ST.HISTO`** and **`VISCOELA`** models are active.

Using the Stress History Model

The stress history in a structure is simulated when the following conditions are satisfied:

- Stress history simulation has been enabled by the **`ST.HISTO`** parameter on the **`METHOD`** statement:

  ```
  METHOD ST.HISTO
  ```

- The **`VISCOELA`** oxidation model has been selected

Stresses in silicon are simulated only if **`^SKIP.SIL`** has been specified on the **`METHOD`** statement. Thermal mismatch stresses are simulated whenever temperature ramping is specified on the **`DIFFUSION`** statement. Intrinsic stresses are included during deposition whenever **`INTRIN.S`** for the deposited material is nonzero. Surface tension is included whenever **`SURF.TEN`** for an exposed material is nonzero.

Limitations

The thermal stress model has the following limitations:

- The model assumes that thermal expansion is dominated by a thick silicon substrate with equivalent conditions on the front and back sides of the wafer (i.e., no bending of the wafer). Although thermal expansions are assumed to take place isotropically, which introduce a thermal strain in z direction, the modeling of mechanical deformation is based on plane strain conditions.
- The model is active only when the **`VISCOELA`** oxidation model is active. Stresses in silicon are calculated only if **`^SKIP.SIL`** has been specified. The stress history is lost and replaced with newly calculated stresses by the **`STRESS`** statement and during oxidation with the **`VISCOUS`** model.
- While the thermal mismatch model works down to room temperature, the diffusion models do not. At temperatures below about 50° C they may produce arithmetic exceptions or cause the program to hang; below about 600° C they may produce inaccurate results.
- The stresses caused by phase changes during temperature cycling are not modeled. It may be possible to approximate these stresses by changing the thermal expansion coefficients as a function of time and temperature.

Dopant-Induced Stress Models

Each dopant atom induces some local lattice distortion in silicon lattice. These local microscopic distortions manifest themselves as macroscopic strain. When these dopants enter the silicon lattice as substitutional atoms, the corresponding macroscopic strain can be estimated from the atomic radius mismatch. This mismatch strain introduces a mismatch stress.

Using linear extrapolations between the different atomic radii and the appropriate relative concentrations (extension of Vegard's law), the native atomic radius, a_{native}, for a doped silicon with dopant radii of (a_1, a_2, ..., a_i, ...) and corresponding mole fractions of (x_1, x_2, ..., x_i, ...) can be estimated as the following:

$$a_{native} = \left(1 - \sum_i x_i\right) a_{si} + \sum_i x_i a_i \qquad \text{Equation 3-472}$$

The effective atomic radius, a_e, of the doped silicon layer depends also on its strain state. For epitaxial growth, the grown layer is strained so the effective atomic radius is that of the underlayer. If the new layer is not strained its effective atomic radius is equal to its native one. The macroscopic strain is computed based on the effective and the native atomic radii using the following expression:

$$\varepsilon = \frac{a_e - a_{native}}{a_{native}} \qquad \text{Equation 3-473}$$

This mismatch strain enters the stress equations as an initial condition and the corresponding lattice mismatch stress is obtained as a result of stress rebalance.

The contribution to the native atomic radius of each individual dopant can be turned on by setting **STN.MIS** in the **METHOD** statement as the following:

```
METHOD IMPURITY=”GERMANIUM CARBON” STN.MIS
```

The atomic radius of each dopant, **AT.RADIU**, can be set by using the **IMPURITY** statement. The dopant induced stress model is only effective when the **ST.HISTO** and **VISCOELA** models are active.

Dopant-Induced Stress during Deposition

The dopant induced stress can be introduced during deposition or epitaxy by setting the **STRAINED** parameter in the **DEPOSITION** statement or the **EPITAXY** statement. The following procedure is performed when the **STRAINED** parameter is enabled during deposition:

- Compute the native atomic radii of the new layer and the underlayer. The native atomic radius of a doped silicon layer is determined by applying Equations 3-472 over all contributing species. Whether a dopant contributes to the native atomic radius is determined by its **DEPO.MIS** setting.
- Set the effective atomic radius to that of the underlayer and Equations 3-473 is applied to determine the mismatch strain.
- Perform stress relaxation operation to establish stress equilibrium after deposition when the boolean parameter **RELAX** is turned on (default).
- Stress equation is solved during diffusion for **EPITAXY** when the Boolean parameter **DIFFUSE** is turned on (default).

When the **`STRAINED`** parameter in the **`DEPOSITION`** statement or **`EPITAXY`** statement is not set, the effective atomic radius of the new layer is that of its native one and no mismatch strain is introduced.

Dopant-Induced Stress during Diffusion

The substitutional concentration for each dopant changes during annealing, so the dopant induced strains and stresses are updated at each time step according to the following procedure:

- The atomic radius at each silicon location from the last time step is retrieved.
- The substitutional dopant concentrations for the current step are retrieved if the corresponding **`DIFF.MIS`** parameters are set.
- The new atomic radius at each location is computed using the new substitutional mole fractions according to Equations 3-472.
- The mismatch strain change at each location is computed using relative change of the atomic radius. The stress change is updated accordingly.

Note:

Setting STN.MIS is identical to setting both DEPO.MIS and DIFF.MIS.

Ion Implantation

The **`IMPLANT`** statement (see Chapter 4, DIFFUSION, p. 4-131) is used to model the implantation of ionized impurities into the simulation structure. Two distinct approaches are available for modeling ion implantation.

- The analytic approach models the impurity and point defect distributions using Gaussian or Pearson functions based on distribution moments contained in a data file.
- The MC approach calculates the trajectories of implanted ions through the 2-D target structure, based on physical models.

These two approaches are described in the following sections.

Analytic Ion Implant Models

The analytic ion implantation models and moment tables from Taurus-Process are now built into TSUPREM-4. These models and tables are referred to as the “Taurus Analytic Implant Model” or the “Taurus Implant Model”. For more information on the Taurus Analytic Implant Model, consult Taurus Analytic Implant Model, p. 3-130 and the *Taurus-Process Reference Manual*. The original TSUPREM-4 analytic implant models are still available and are used by default.

The impurity being implanted is selected with one of the parameters **`ANTIMONY`**, **`ARSENIC`**, **`BORON`**, **`PHOSPHOR`**, or **`BF2`**. The **`DOSE`** parameter specifies the total number of impurity ions per square centimeter provided by the ion implanter. The acceleration energy of the ions is specified with the **`ENERGY`** parameter. Only the portion of the distribution within the simulation region contributes to the dose in the structure. If the **BACKSCAT** parameter is specified, the portion of the distribution above the top of the simulation structure is assumed to be backscattered at the surface. Any portion of the distribution below the bottom of the simulation region is assumed to have passed through the structure.

Implanted Impurity Distributions

Implanted impurity distributions in a 2-D structure are derived from distributions calculated along vertical lines through the structure. The 1-D procedures described below are used to find the vertical implant distribution along each line.

Each 1-D profile is converted to a 2-D distribution by multiplying by a function of x. The final profile is determined by integrating the contributions of all the 2-D distributions to the doping at each node. If the **TILT** parameter is nonzero, the lines for the 1-D calculation are taken at the specified angle from the vertical. The variable u in the discussion that follows then represents the distance along the angled line, while the variable x corresponds to distance perpendicular to the slices.

The vertical distribution along each line is given by:

$$I(u) = \mathbf{DOSE} \times f(u) \qquad \text{Equation 3-474}$$

where $u = 0$ occurs at the surface of the top material layer along the line, and $f(u)$ is a normalized Gaussian or Pearson distribution, depending on whether **GAUSSIAN** or **PEARSON** is selected on the **IMPLANT** statement. The equations for $f(u)$ are described below.

The vertical distribution function $f(u)$ is calculated from its spatial distribution moments. The first four moments are defined as:

$$R_p = \int_{-\infty}^{\infty} u f(u)\, du \qquad \text{Equation 3-475}$$

$$\sigma = \sqrt{\int_{-\infty}^{\infty} (u - R_p)^2 f(u)\, du} \qquad \text{Equation 3-476}$$

$$\gamma = \frac{\int_{-\infty}^{\infty} (u - R_p)^3 f(u)\, du}{\sigma^3} \qquad \text{Equation 3-477}$$

$$\beta = \frac{\int_{-\infty}^{\infty} (u - R_p)^4 f(u)\, du}{\sigma^4} \qquad \text{Equation 3-478}$$

The values of R_p, σ, γ, and β are obtained from the implant data file *s4imp0* or from an alternate implant data file specified with the **IN.FILE** parameter in the **IMPLANT** statement. For each combination of impurity and material, these files contain the distribution moments for a series of acceleration energies in order of increasing energy.

Implant Moment Tables

The implant data file associates distribution moments with each ion and material by using ion and material names present in the file. The material names are the same as those used elsewhere in the program (i.e., **SILICON**, **OXIDE**, **NITRIDE**, **POLYSILI**, **PHOTORES**, and **ALUMINUM**). The data for silicon dioxide is also used for silicon oxynitride.

The data table to be used for an implantation step can be specified explicitly with the **IMPL.TAB** parameter in the **IMPLANT** statement. This allows the implant data file to contain several sets of distribution moments for the same impurity. The implant data file defines the following table names:

- **ANTIMONY**: Original antimony data with extended energy ranges fitted to results of amorphous MC calculations (energies: 5–1000 keV). This is the default for antimony implantation.
- **ARSENIC**: Original arsenic data with extended energy ranges fitted to results of amorphous MC calculations (energies: 5–11,000 keV).
- **DUAL.ARS**: Dual-Pearson data for arsenic with channeling in silicon (energies: 10-1000 keV) Reference [48]. This is the default for arsenic implantation.
- **TR.ARSENIC**: Dual-Pearson data for arsenic in <100> silicon with full energy, dose, tilt, and rotation dependence (energy: 0.5–180 keV; dose: 10^{13}–8×10^{15} atoms/cm^2; tilt: $0° – 10°$; rotation: $0° – 45°$) References [49] and [107].
- **BF2**: Data for boron from a BF_2 source (energies: 5–120 keV) Reference [51].
- **DUAL.BF2**: Dual-Pearson data for boron from a BF_2 source with channeling in silicon (energies: 10–200 keV) Reference [48]. This is the default for BF_2 implantation.
- **UT.BF2**: Dual-Pearson data for boron from a BF_2 source with channeling in silicon (energies: 15–120 keV) Reference [49].
- **TR.BF2**: Dual-Pearson data for BF_2 in <100> silicon with full energy, dose, tilt, and rotation dependence (energy: 0.5–65 keV; dose: 10^{13}–8×10^{15} atoms/cm^2; tilt: $0° – 10°$; rotation: $0° – 45°$) References [49] and [107]. The data for < 5 keV implants were generated by using the MC model in Taurus Process & Device, calibrated using Eaton data.

- **BORON**: Original boron data with extended energy ranges fitted to results of amorphous MC calculations (energies: 5–4000 keV).
- **LEBORON**: Data for low-energy boron with channeling in silicon (energies: 10–30 keV) Reference [51].
- **CHBORON**: Data for boron with channeling in silicon (energies: 5–2000 keV) Reference [48]. This is the default for boron implantation.
- **UT.BORON**: Dual-Pearson data for boron with channeling in silicon (energies: 15–100 keV) Reference [49].
- **TR.BORON**: Dual-Pearson data for boron in <100> silicon with full energy, dose, tilt, and rotation dependence (energy: 0.5–80 keV; dose: 10^{13}–8×10^{15} atoms/cm^2; tilt: $0°$–$10°$; rotation: $0°$–$45°$) References [52] and [107]. The data for < 5 keV implants were generated by using the MC model in Taurus Process & Device, calibrated using Eaton data.
- **SCR.BORON**: Dual-Pearson data for boron in <100> silicon with full energy, dose, tilt, rotation, and *screening oxide thickness* dependence. Data from The University of Texas at Austin (energy: 15 keV-80 keV; dose: 10^{13}–8×10^{15} atoms/cm^2; tilt: $0°$–$10°$; rotation: $0°$–$45°$; oxide thickness: 15-400 Angstroms).
- **PHOSPHORUS**: Original phosphorus data with extended energy ranges fitted to results of amorphous MC calculations (energies: 5–7000 keV).
- **DUAL.PHO**: Dual-Pearson data for phosphorus with channeling in silicon (energies: 10–200 keV) Reference [48]. This is the default for phosphorus implantation.
- **TR.PHOSPHORUS**: Dual-Pearson data for phosphorus into bare <100> silicon with full energy, dose, tilt, and rotation dependence (energy: 15—180 keV; dose: 10^{13}–8×10^{15} atoms/cm^2; tilt: $0°$–$10°$; rotation: $0°$–$45°$) Reference [50].
- **FLUORINE**: Dual-Pearson data for fluorine fitted to results of MC calculations (energies: 2–95 keV).
- **TR.INDIUM**: Dual-Pearson data for indium in <100> silicon with full energy, dose, tilt, and rotation dependence (energy: 1–100 keV; dose: 10^{13}–8×10^{15} atoms/cm^2; tilt: $0°$–$10°$; rotation: $0°$–$45°$). The 200 keV parameters are based on tilt=0 implants, while 300 keV are based on tilt=7 and rotation=30 implants. The range statistics were extracted from the data generated by MC simulations. This is the default for indium implantation into silicon. For other materials, the tables of antimony are used for indium implants.

The energy ranges shown are for implantation into silicon. If no data is present in the implant data file for the specified energy, linear interpolation is used with the available data to determine the distribution moments. No extrapolation is done for energies outside the range of the table. For implants into silicon, an error is printed and the program terminates; for other materials, a warning is printed and the nearest available energy is used. If the first three moments for the specified energy are present in the implant data file, but the value of β is missing or zero, then β is calculated using the expression Reference [53]:

$$\beta = 2.91 + 1.56\gamma^2 + 0.59\gamma^4 \quad \text{Equation 3-479}$$

which yields a reasonable value for β and requires knowledge of only the first three distribution moments.

Moments for up to 20 materials can be included in the implant moments file. Each material in the file has an unique name. Usually, TSUPREM-4 searches the moments file for a material with a name that matches the material name used in TSUPREM-4. For example, data for the material called *photoresist* in the moments file is used for implantation into photoresist. But you can request that a different set of data be used with the **IMPL.TAB** parameter on the **MATERIAL** statement. Thus the statement

```
MATERIAL MAT=PHOTORESIST IMPL.TAB=AZ-7500
```

requests that data for the material named *az-7500* in the implant moments file be used for implantation into photoresist.

Gaussian Distribution

A Gaussian distribution requires only the moments R_p and σ and is defined as:

$$f(u) = \frac{1}{\sqrt{2\pi}\,\sigma}\exp\left[\frac{-(u-R_p)^2}{2\sigma^2}\right] \quad \text{Equation 3-480}$$

where R_p and σ are defined above.

Pearson Distribution

A Pearson distribution requires the moments R_p, σ, γ, and β, and is defined by the differential equation Reference [54]:

$$\frac{df(v)}{dv} = \frac{(v-a)f(v)}{b_0 + av + b_2v^2} \quad \text{Equation 3-481}$$

where:

$$v = u - R_p \quad \text{Equation 3-482}$$

$$a = \frac{-\sigma\gamma(\beta+3)}{A} \quad \text{Equation 3-483}$$

$$b_0 = \frac{-\sigma^2(4\beta - 3\gamma^2)}{A} \quad \text{Equation 3-484}$$

$$b_2 = \frac{-2\beta + 3\gamma^2 + 6}{A} \quad \text{Equation 3-485}$$

$$A = 10\beta - 12\gamma^2 - 18$$ Equation 3-486

Not all combinations of β and γ produce useful profiles. The characteristics of the profile can be determined by examining the denominator of Equation 3-481:

$$p(\nu) = b_0 + a\nu + b_2\nu^2$$ Equation 3-487

For the profile to have a maximum at $\nu = a$, $p(a)$ must be negative; for the mean of the profile to be at R_p requires that $b_2 > -1/2$.

TSUPREM-4 checks these conditions and produces an error unless both are satisfied. The fourth moment is equal to β only if $b_2 > -1/5$, which can only occur when $A > 0$. However, the program produces profiles for $-1/2 < b_2 < -1/5$ without printing a warning (provided that $p(a) < 0$).

The nature of the profile depends on the value of b_2 and whether there are real solutions to the equation $p(\nu) = 0$. In the following, it is assumed that $p(a) < 0$ and the solutions to $p(\nu) = 0$ (if they exist) are ν_1 and ν_2, with $\nu_1 \le \nu_2$. For $b_2 > 0$, the profile is nonzero for $\nu_1 < \nu < \nu_2$; this corresponds to a Pearson type I or II distribution. If $b_2 \le 0$ and $p(\nu) = 0$ has no real solutions, then the profile is nonzero for all ν, but approaches zero as ν approaches $\pm\infty$; this is a Pearson type IV or VII distribution for $b_2 < 0$ and a Gaussian for $b_2 = 0$.

If $b_2 \le 0$ and $p(\nu) = 0$ has real solutions, then the profile either goes to zero at ν_1 and $-\infty$ (for $a < \nu_1$) or at ν_2 and ∞ (for $a > \nu_2$); this is Pearson type V or VI distribution for $b_2 < 0$ and a type III distribution for $b_2 = 0$.

Dual Pearson Distribution

In the case of implantation into crystalline silicon, for low to medium dose implants the shape of the resultant profile is relatively insensitive to implant dose. In this range, a single Pearson function can be used to model the range of implant doses at a given energy by simply scaling the magnitude of the profile to match the implant dose. This is the method used when the **PEARSON** distribution is selected and data for a single Pearson function is included in the implant moment data file for the specified impurity name.

Dose-dependent Implant Profiles

At higher doses, substrate damage causes the implant profile to vary significantly with dose. A dual-Pearson approach has been found to work well for modeling such dose-dependent implant profiles References [48], [49], and [52]. When the **PEARSON** distribution is selected and data for a dual-Pearson function is included in the implant moment data file for the specified impurity name, a dual-Pearson function is used.

Following this approach, one Pearson profile models the channeled portion of the implant profile, while the second Pearson profile models the nonchanneled portion resulting from implantation into partially amorphized silicon. The dose dependence is modeled by varying the relative magnitude of the channeled and nonchanneled Pearson functions. The moments of each of the Pearson functions are independent of dose. The composite dual-Pearson profile can be described as:

$$I_{composite}(u) = rI_{amorphous}(u,\ R_{Pa},\ \sigma_a,\ \gamma_a,\ \beta_a) + (1-r)I_{channeled}(u,\ R_{Pc},\ \sigma_c,\ \gamma_c,\ \beta_c)$$

where:

- $I_{amorphous}$ and $I_{channeled}$ are the normalized amorphous and channeled Pearson profiles, respectively.
- r is the ratio of the dose of the amorphous profile to the total dose.
- subscripts a and c refer to amorphous and channeled profiles, respectively.
- u is the depth coordinate along the line.

The implant data file for dual-Pearson data has two sets of four moments, corresponding to both Pearson profiles, and a table of ratio values as a function of dose. Figure 3-1 shows the variation of profile shape with implantation dose, in the case of an initially crystalline silicon substrate, and compares the dual-Pearson model with data taken from Reference [49]. In Figure 3-1 (a)-(d), the composite profile is plotted (solid line), along with the channeled and amorphous profile contributions (dashed lines). The ratio of the dose allocated to the amorphous profile relative to the channeled dose is indicated in each figure. A ratio of unity indicates a highly damaged substrate with little channeling, whereas a ratio of zero indicates a highly channeled profile with little or no implant damage.

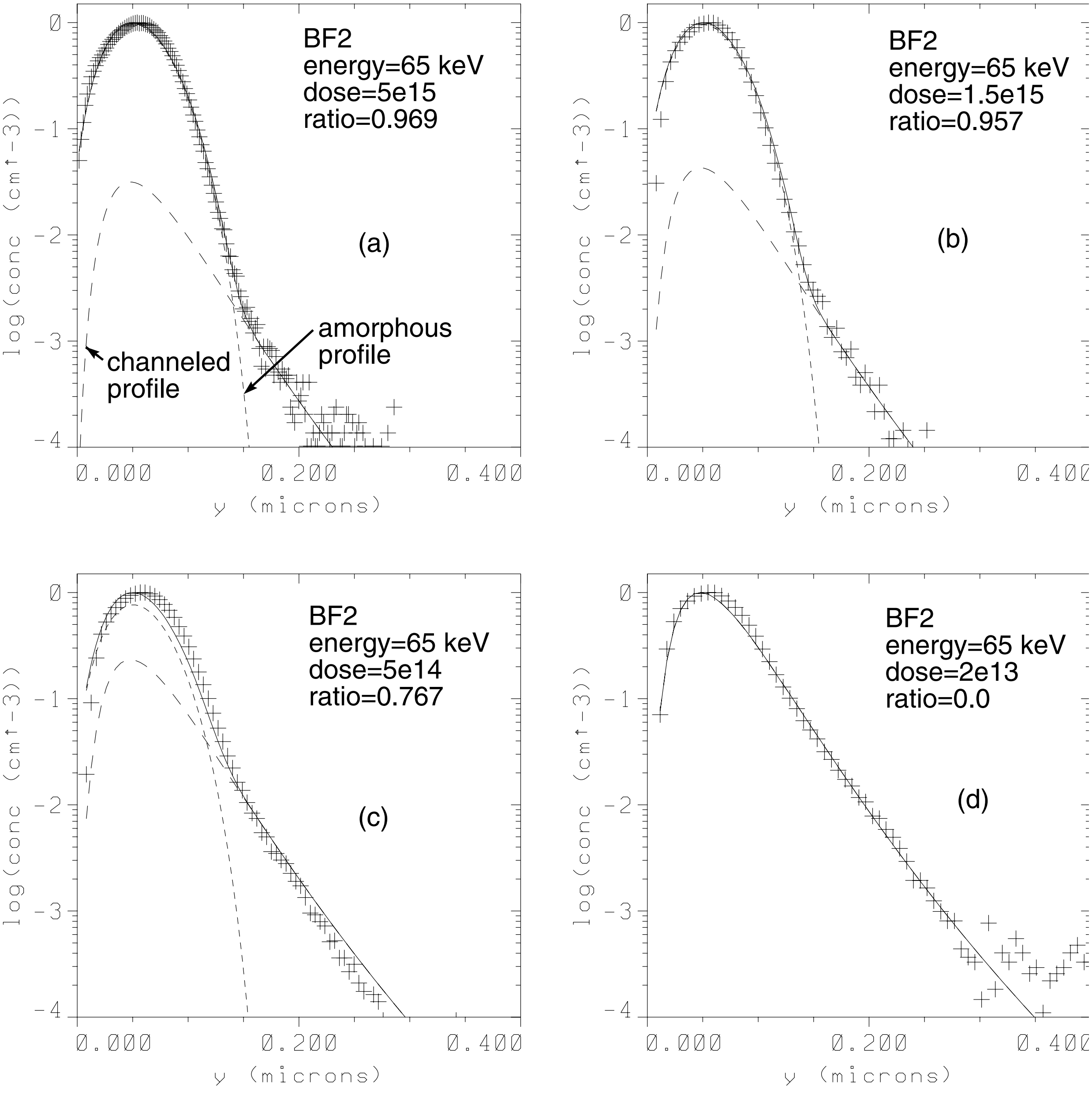


Figure 3-1 BF_2 implant profile

When the **GAUSSIAN** distribution is selected and dual-Pearson data is contained in the implant data file, the first set of moments is used as described above for the calculation of the distribution. If the dose is outside the range of values available in the implant data file, the fraction corresponding to the nearest available dose is used. The dual-Pearson functions are useful for modeling channeling effects, with the primary Pearson function modeling the bulk of the implant and the secondary Pearson function modeling the distribution of channeled ions.

Tilt and Rotation Tables

The *tr.arsenic, tr.bf2,* and *tr.boron* tables contain dual-Pearson distributions for a range of energies, doses, wafer tilts, and rotations for implantation into bare, <100> silicon. When one of these tables is specified with the **IMPL.TAB** parameter, the distribution for the specified energy, dose, tilt and rotation are obtained from the table. If the specified implant parameters do not appear in the table, linear interpolation (in four dimensions) between table entries is used. If the specified implant parameters lie outside of the range of the table, a warning message is printed and the nearest available values are used.

Note:

> ***The tilt and rotation tables give accurate results after implantation into bare, <100> silicon, but may not be appropriate under other implant conditions.***

Multilayer Implants

A multilayer implant is represented by treating each layer sequentially, starting with the top layer in the structure. The impurity distribution $I(u)$ is determined by first obtaining the moments from the implant data file for the impurity in the material comprising the layer. The distribution $I(u-u_l+u_s)$ is used for the impurity distribution within the layer, where:

$$u_l = \sum_i t_i \qquad \text{Equation 3-488}$$

The summation is performed over all previously treated layers of the structure, and t_i is the thickness of layer i. Either of two approaches can be used to determine u_s: effective range model or dose matching.

Effective Range Model

By default, u_s is given by Reference [55]:

$$u_s = \sum_i \frac{t_i R_p}{R_{p_i}} \qquad \text{Equation 3-489}$$

where R_{p_i} is the first moment of $f(u)$ in layer i, R_p is the first moment of $f(x)$ in the present layer, and the summation is performed over all previously treated layers of the structure. For layers below the first, the magnitude of the distribution is scaled so that the integral of $I(u)$ from $u = u_s$ to $u = \infty$ plus the total dose placed in all previously treated layers is equal to the specified implant dose. This method is referred to as the *effective range* or *effective thickness* approach.

Dose Matching

If the parameter **RP.EFF** is set as false on the **IMPLANT** statement, u_s is determined such that the integral of $I(u)$ from $u = 0$ to $u = u_s$ equals the total dose placed in all previously treated layers Reference [56]. For the top layer, $u_s = 0$. This method is referred to as the *dose matching* approach.

The effective range approach has proved to be more accurate than the dose matching approach; the dose matching approach is retained only for compatibility with older revisions (8926 and older) of the program.

Lateral Distribution

Each 1-D profile is expanded to two dimensions by multiplying by a Gaussian distribution in the direction perpendicular to the line Reference [57]:

$$I(u,\ v) = I(u) \times \frac{1}{\sqrt{2\pi}\ \sigma_x} \exp\left(-\frac{v^2}{2\sigma_x^2}\right) \qquad \text{Equation 3-490}$$

where v is the distance perpendicular to the line. The quantity σ_x is the lateral standard deviation of the implant profile in the given material and is found by interpolation in the implant data file. The complete implant profile is obtained by summing together the 2-D profiles produced by all of the lines.

Depth-Dependent Lateral Distribution

In order to better model the lateral distribution, TSUPREM-4 has two additional parameters, **`LSLOPE`** (for the first Pearson) and **`D.LSLOPE`** (for the second Pearson) to allow for the depth dependent lateral straggle:

```
σ1(z)  = SIGMA*( 1 + LSLOPE*(z/RANGE -1) ),
σ2(z)  = D.SIGMA *( 1 + D.LSLOPE*(z/D.RANGE-1) ),
```

where z is the depth. **`LSLOPE`** and **`D.LSLOPE`** can be specified on the **`MOMENT`** statement. TSUPREM-4 versions 1999.4 or later also provide the default parameters for **`LSLOPE`** and **`D.LSLOPE`** in all "**`TR`**" tables. By choosing the parameters **`LSLOPE`** and **`D.LSLOPE`** properly, the 2-D profiles can be modeled more accurately.

The depth-dependent lateral straggle model will be automatically used if the "**`TR`**" tables are specified. This model is not used for any other tables.

Please note that starting from TSUPREM-4 version 1999.4, the lateral straggle formula has been changed in such a way that

```
LSLOPE(new)  = LSLOPE(old)/SIGMA,
D.LSLOPE(new)  = D.LSLOPE(old)/D.SIGMA
```

Identical results will be obtained from previous input files if **`V.COMPAT`** is set to versions earlier than 1999.4. On the other hand, if no compatibility mode is set, **`LSLOPE`** and **`D.LSLOPE`** should be adjusted accordingly on the **`MOMENT`** statement.

Dose Integration from Lateral Distribution

When the dose from lateral distribution is added to the neighbor nodes, TSUPREM-4 reduces the calculation time by checking whether the dose to be added to a neighbor node is smaller than the certain tolerance so that it can be ignored to the current dose of the neighbor. The parameter **`II.TOL.F`** in **`METHOD`** statement defines the control factor for the tolerance. It may not be negative and not exceed 1.0. The smaller **`II.TOL.F`** means the smaller tolerance so that the calculation is more accurate, but takes more time. By default, it is set to 0.1 from version 2004.09 while it was 1.0 in version 2003.12 and the older versions.

Wafer Tilt and Rotation

Tilt and rotation of the wafer during implantation are specified by the **`TILT`** and **`ROTATION`** parameters, respectively. The various effects of tilting the wafer are simulated as follows:

- When a nonplanar structure is tilted, shadowing of portions of the structure can occur. The amount of shadowing for a given tilt depends on the rotation—from full shadowing at zero rotation to no shadowing at 90° rotation. Shadowing is simulated by tilting the simulated structure by an appropriate amount.
- The number of incident ions per square centimeter of the wafer is reduced when the wafer is tilted. This dose reduction is simulated naturally when the simulation structure is tilted, but an analytical adjustment to the dose is required if the tilt of the simulation structure is less than the specified wafer tilt because of rotation.
- Tilting the wafer causes the implanted profiles to be foreshortened. As with dose reduction, foreshortening of the profiles is simulated by a combination of tilting the simulation structure and by analytical adjustments to the implant parameters. If the implantation data is derived from measurements based on tilted wafer samples, the tilt of the data also affects foreshortening. For this reason, each table of data in the implantation data file has an associated tilt and rotation value that reflects the conditions under which the implantation data were measured. The tilt value for each table is used in the calculation of foreshortening effects.
- The tilt and rotation of the wafer affect the number of ions that are channeled along crystal planes in silicon. Channeling effects in bare <100> silicon as a function of tilt and rotation are included in the *tr.arsenic, tr.bf2,* and *tr.boron* implant tables.

BF_2 Implant

When the model `FLUORINE` is enabled in **`METHOD`** statement, i.e.,

```
METHOD MODEL=FLUORINE ENABLE
```

fluorine is automatically implanted during BF_2 implantation. the fluorine of BF_2 is implanted by a fluorine implant with an energy of 0.3893 times the user-specified implant energy. The fluorine dose is two times the BF_2 dose.

Analytic Damage Model

When the **`DAMAGE`** parameter on the **`IMPLANT`** statement is set true, an analytic model for the production of point defects during ion implantation is invoked. The interstitial and vacancy distributions created by the implantation are added to any interstitials and vacancies that may have existed in the structure prior to implantation. For more information on how damage is used to model transient-enhanced discussion (see Implant Damage Model, p. 3-144).

Damage Distribution Calculations

The damage distributions are calculated using the model of Hobler and Selberherr in its 1-D form Reference [58]. This model approximates the damage profiles by combinations of Gaussian and exponential functions. The parameters of these functions were chosen to fit damage profiles predicted by MC simulations over the range of implant energies between 1 and 300 keV. For a complete description of the equations and parameter values, see Reference Reference [58]. The implementation in TSUPREM-4 differs from that in Reference [58] in that both the Gaussian and exponential components are used when implanting arsenic at energies above 170 keV. The model is extended to two dimensions by multiplying by a Gaussian with a standard deviation equal to the value of Hobler and Selberherr's a_3 parameter (the standard deviation of the Gaussian component of the vertical distribution). For BF_2 implants, the model for boron is used and the damage is scaled by 49/11. The scaling is omitted if **`V.COMPAT`**=6.4 is specified on the **`OPTION`** statement.

Recommended Usage and Limitations

The damage model is designed only for antimony, arsenic, boron, and phosphorus. If **`DAMAGE`** is specified on a BF_2 implantation, the model for boron is used, with energy reduced by a factor of 0.2215. For other impurities, the damage model for one of the ions listed above (the one having atomic weight closest to that of the implanted ion) is used. There are no user-accessible coefficients associated with the damage model. If the damage calculation is followed by a **`DIFFUSION`** step, care should be used to ensure that a simulation region with sufficient depth is used to accommodate the rapid diffusion of point defects into the structure.

Taurus Analytic Implant Model

The Taurus Analytic Implant Model offers the following improvements compared to the original TSUPREM-4 models:

- Improved calibration for a wider range of implant conditions.
- No need to chose between alternative moment tables. The best known data has been consolidated into a single table for each ion/target combination.
- Moments available for effectively all energy values. Moments are automatically interpolated down to zero energy and go up to 10 MeV
- Improved lateral moments based on a matrix of crystalline MC simulations.
- Automatic channeling suppression and moment relaxation for implants through amorphous screen layers.
- Automatic channeling suppression and moment relaxation for implants into pre-damaged (PAI) wafers.
- Ion backscattering is considered for scattering angles larger than 90 degrees.
- Moment tables are easily extended. When moment data is added to the standard moment tables, it is automatically combined with all other data for interpolation to other implant conditions.

To use the Taurus Analytic Implant Model, specify **`IMPL.TAB=taurus`** in the **`IMPLANT`** command.

For a complete description of the Taurus Analytic Implant model, see Chapter 3 of the *Taurus-Process Reference Manual*. Use the **`IMPLANT`**, **`METHOD`**, **`MOMENT`**, and **`MATERIAL`** commands to set the relevant parameters in TSUPREM-4.

MC Ion Implant Model

This version of TSUPREM-4 offers two different models for MC implant:

- Old TSUPREM-4 MC model
- Taurus MC Implant model which is the default MC model

Taurus MC Implant

You can select Taurus MC Implant model by specifying **MONTECAR** on the **IMPLANT** statement. Taurus MC Implant model generally provides better accuracy and more robust solutions for most of implant conditions Reference [59]. It has been calibrated from sub-keV to above 10 MeV, and for different implant conditions including random implant direction, <100>, <111>, and <110> channeling directions, with the same set of parameters for boron, phosphorus, and arsenic implants Reference [60]. It is also accurate for other implant species such as BF2, F, Al, Ge, In, Sb, etc. Reference [61] For most of MC implant simulations, Taurus MC Implant model should be used. For a complete description of the physical models implemented in Taurus MC Implant, see Chapter 3 of the *Taurus Process Reference Manual* or Chapter 3 of the *Sentaurus Process Users Guide*.

The default model parameters in Taurus MC Implant can be changed by using TSUPREM-4 **MATERIAL** and **IMPURITY** statements. The Debye temperature (**DEBYE.T**), the critical concentration for amorphization (**MAX.DAM**), and the surface disorder (**SURF.DIS**) can be specified in the **MATERIAL** statement, while angular divergence angle (**DIV.ANG**), LSS correction factor (**LSS.PRE**), nonlocal prefactor (**NLOC.PRE**), nonlocal exponent (**NLOC.EXP**), electron stopping screening length parameter (**SCR.PAR**), survival rate (**SURV.RAT, CASC.SUR**), amorphousness correction factor (**AMOR.PAR, CASC.A**mo), damage saturation parameter (**SAT.PAR**, **CASC.SAT**), , and displacement threshold (**DISP.THR, CASC.DIS**) can be specified in the **IMPURITY** statement. Note that the parameters starts with **CASC** are for Cascade Damage model. Although the default parameter values for Cascade Damage model are the same as those for the Plus Damage model, using different set of parameters allows for more flexibility in calibration.

Model switches, depth split (**DEPTH.SP**), dose split (**DOSE.SPL)**, trajectory split (**TRAJ.SPL)**, trajectory replication (**TRAJ.REP)**, oxygen recoil (**REC.OXYG)**, seed randomization (**RANDOM)**, point response implant mode (**POINT.RE)**, and **SAVE1D** can be selected in the **IMPLANT** statement. In addition, damage model (**DAM.MOD**), subwindow length of the trajectory replication (**X.LENGTH**), and the maximal level of split for trajectory splitting (**MAX.SPLI**) also can be specified in the **IMPLANT** statement.

For reference, Table 3- 1, “Model Comparisons After Relaxation,” on page 116 lists the corresponding parameters in *Taurus Proces* and *Sentaurus Process.*

Table 3-1 Correspondence of Taurus MC parameters among TSUPREM-4, Taurus Process, and Sentaurus Process

TSUPREM-4	Taurus Process	Sentaurus Process
`debye.t`	`DebyeTemperature`	`DebyeTemperature`
`max.dam`	`AmorphizationThreshold`	`AmorpThreshold`
`surf.dis`	`SurfaceDisorder`	`SurfaceDisorder`
`div.ang`	`BeamDivergenceAngle`	`BeamDivergence`
`lss.pre`	`LSSCorrectionFactor`	`LSS.pre`
`nloc.pre`	`NonlocalPrefactor`	`nloc.pre`
`nloc.exp`	`NonlocalExponent`	`nloc.exp`
`surv.rat`	`SurvivalRate`	`surv.rat`
`SCR.PAR`	-	`scr.par`
`SAT.PAR`	-	`sat.par`
`CASC.SAT`	-	`casc.sat`
`casc.sur`	`SurvivalRate`	`casc.sur`
`amor.par`	`AmorphousParameter`	`amor.par`
`casc.amo`	`AmorphousParameter`	`casc.amo`
`disp.thr`	`DisplacementThreshold`	`disp.thr`
`casc.dis`	`DisplacementThreshold`	`casc.dis`
`dam.mod`	`DamageModel`	-
`depth.sp`	`DepthSplit`	-
`dose.spl`	`DoseSplit`	-
`max.spli`	`MaxSplit`	-
`traj.spl`	`TrajectorySplit`	`TrajectorySplitting`
`traj.rep`	`TrajectoryReplication`	`TrajectoryReplication`
`random`	`Randomize`	-
`rec.oxyg`	`OxygenRecoil`	-
`point.re`	`PointResponse`	-
`save1d`	`Save1D`	-
`x.length`	`xLength`	-

Old TSUPREM-4 MC Model

You can select the old MC model for ion implantation by specifying the **`OLD.MC`** parameter on the **`IMPLANT`** statement. The following sections in this manual describe the old TSUPREM-4 MC implant model only. For more information on the Taurus MC Implant model, see Chapter 3 of *Taurus-Process Reference Manual*, or Chapter 3 of *Sentaurus Process Users Guide*.

TSUPREM-4 contains a comprehensive MC model that incorporates models for crystalline silicon as well as amorphous models for silicon and other materials. The calculation models the crystal to amorphous transition that occurs during implantation. The model includes the effect of reflected ions and produces damage information in the form of vacancy and interstitial profiles. In addition, TSUPREM-4 contains models for calculating damage self-annealing of silicon substrates.

The MC calculation is useful for examining a number of dependencies for which the empirical models are imperfect or incompletely calibrated. Some examples of these are:

- Profile dependence on tilt and rotation angles
- Dose dependence
- Implant temperature dependence
- Low energy implants

The MC model is the only implant model in TSUPREM-4 that can simulate the effects of reflected ions.

The capabilities contained in TSUPREM-4 are a superset of the MC ion implant functionality of the 1-D process simulator, PEPPER Reference [62]. The calculation used in TSUPREM-4 assumes that ions lose energy through two processes.

- Nuclear scattering, where the nucleus of the ion elastically scatters off the nucleus of an atom in the target. This interaction is based on binary collision theory and is described in the following section.
- Interaction of the ion with the electrons of the target atoms. This mechanism is inelastic and does not alter the direction of the ion’s motion. This is described in Amorphous Implant Calculation, p. 3-137.

The calculation of damage and damage self-annealing of silicon substrates are described in Crystalline Implant Model, p. 3-140.

Binary Scattering Theory

TSUPREM-4 models the nuclear collision energy loss according to classical binary scattering theory. The basic assumption of the nuclear collision energy loss mechanism is that the ion interacts with only one target atom at a time. This assumption allows the use of binary scattering theory from classical mechanics Reference [63]. This section briefly outlines the pertinent results of this theory.

Consider a particle of mass M_1 and kinetic energy E_0 approaching a stationary particle with mass M_2. The impact parameter, b, is the distance of closest approach if the particle is not deflected and gives a convenient measure of how close the collision is. After collision, the first particle deviates from its original course by an angle θ.

Energy Loss

It can be shown that the first particle loses kinetic energy

$$\frac{\Delta E_n}{E_0} = \frac{4M_1M_2}{(M_1+M_2)^2}\cos^2(bI) \qquad \text{Equation 3-491}$$

where ΔE_n is the energy lost by particle 1, E_0 is its energy before collision, and I is the integral:

$$I = \int_0^{s_{max}} \frac{ds}{\sqrt{1-\frac{V(s)}{E_r}-b^2s^2}} \qquad \text{Equation 3-492}$$

where $s = 1/r$ is the inverse separation between the two particles, $V(s)$ is the potential between the two particles (assumed to be repulsive), and:

$$E_r = \frac{E_0}{1+M_1/M_2} \qquad \text{Equation 3-493}$$

is the reduced energy in the center of mass coordinates. The upper limit of the integral, s_{max}, is the inverse distance of closest approach of the two particles, and is given by the solution to the equation:

$$1-\frac{V(s_{max})}{E_r}-b^2s_{max}^2 = 0 \qquad \text{Equation 3-494}$$

Scattering Angle

The angle θ by which particle 1 is deflected is given by:

$$\cos\theta = \frac{1-0.5\left[1+\frac{M_2}{M_1}\right]\Delta E/E_0}{\sqrt{1-\Delta E/E_0}} \qquad \text{Equation 3-495}$$

Dimensionless Form

Equations 3-491 through 3-495 are the basic equations for classical two-body scattering. The scattering integral, Equation 3-492, can be cast into a dimensionless form by assuming the potential has the form:

$$V(s) = Z_1Z_2k_1sg(a_us) \qquad \text{Equation 3-496}$$

where Z_1 is the charge on particle 1, Z_2 is the charge on particle 2, k_1 is the constant:

$$k_1 = \frac{q^2}{4\pi\varepsilon_0} = 14.39495 \times 10^{-7}\,\mathrm{keV}\,\mu\mathrm{m} \qquad \text{Equation 3-497}$$

$g(a_u s)$ is an arbitrary function of $a_u s$, to be defined later, and a_u is a unit of length. TSUPREM-4 uses:

$$a_u = 0.8854 \times 10^{-4}\frac{0.529}{\left(Z_1^{0.23} + Z_2^{0.23}\right)}\,\mu\mathrm{m} \qquad \text{Equation 3-498}$$

which is the so-called Universal screening length Reference [64] and a dimensionless impact parameter:

$$b_n = b/\,a_u \qquad \text{Equation 3-499}$$

and a dimensionless energy:

$$\varepsilon = \frac{a_u E_r}{Z_1 Z_2 k_1} \qquad \text{Equation 3-500}$$

Using Equations 3-498, 3-499, and 3-500 in the scattering integral Equation 3-492, and making the substitution $s' = a_u s$ gives:

$$I = \frac{1}{a_u}\int_0^{s'_{max}} \frac{ds'}{\sqrt{1 - s'\,g(s')/\,\varepsilon - b_n^2 s'^2}} \qquad \text{Equation 3-501}$$

From Equation 3-491, the quantity of interest is $\cos^2(bI)$, which becomes:

$$\cos^2(bI) = \cos^2\left[b_n\int_0^{s'_{max}} \frac{ds'}{\sqrt{1 - s'\,g(s')/\,\varepsilon - b_n^2 s'^2}}\right] \qquad \text{Equation 3-502}$$

Thus using Equation 3-502, $\cos^2(bI)$ can be evaluated in terms of the dimensionless variables b_n and ε, without reference to a particular particle's charge or mass.

Coulomb Potential

As an example of the above procedure, consider the Coulomb potential between two particles:

$$V(r) = \frac{Z_1 Z_2 k_1}{r} \qquad \text{Equation 3-503}$$

or $V(s) = Z_1 Z_2 k_1 s$. In this case, $g(a_u s) = 1$. Then from Equation 3-502.

$$\cos^2(bI) = \cos^2\left[b_n \int_0^{s'_{max}} \frac{ds'}{\sqrt{1 - s'/\varepsilon - b_n^2 s'^2}}\right] \qquad \text{Equation 3-504}$$

with:

$$s'_{max} = \frac{\left(\sqrt{1 + 4b_n^2\varepsilon^2} - 1\right)}{2\varepsilon b_n^2} \qquad \text{Equation 3-505}$$

from a solution of Equation 3-494. Then the integral can be evaluated exactly, giving:

$$\cos^2(bI) = \frac{1}{1 + 4b_n^2\varepsilon^2} \qquad \text{Equation 3-506}$$

For a given impact parameter b and incident energy E_0, the dimensionless b_n and ε can be obtained from Equations 3-499 and 3-500, giving $\cos^2(bI)$ from Equation 3-506. Then the energy loss due to the collision is given by Equation 3-491, and the angle at which particle 1 leaves the collision is given by Equation 3-495.

Universal Potential

For the simple form of the Coulomb potential used in the example above, the scattering integral can be solved analytically. For more realistic inter-atomic potentials, however, the scattering integral cannot be evaluated analytically. For example, the Universal potential Reference [64] that is used in TSUPREM-4 is:

$$V(r) = \frac{Z_1 Z_2 k}{r}\left[0.18175e^{-3.1998r/a_u} + 0.50986e^{-0.94229r/a_u} + 0.28022e^{-0.4029r/a_u} + 0.028171e^{-0.20162r/a_u}\right] \qquad \text{Equation 3-507}$$

An analytic solution does not exist since the upper limit of the integral in Equation 3-492 is given by Equation 3-494, which becomes a transcendental equation with this potential.

All MC ion implantation codes use a formalism similar to Equations 3-491 through 3-495 to treat the nuclear scattering; the difference between codes is in the method of evaluating the scattering integral, Equation 3-492. The code MARLOWE Reference [65] numerically integrates the scattering integral, providing accurate solutions at great computational expense. The program TRIM Reference [66] fits an analytic function of five parameters to the values of the scattering integral obtained by numerical integration. This technique retained the accuracy of MARLOWE while improving efficiency by an order of magnitude. However, TRIM still requires the evaluation of s_{max}, the inverse distance of closest approach at each collision. This requires solving the nonlinear Equation 3-494. With an initial guess of $s_{max} = 1/b$, Newton's method converges to an answer in about 3 to 5 iterations. If the Universal potential Reference [64] is used, this requires 18 to 30 exponentials to be evaluated at each collision.

A different algorithm is used in TSUPREM-4 Reference [67]. The quantity $\cos^2(bI)$, in its dimensionless form, Equation 3-502, is numerically integrated for a wide range of its parameters b_n and ε. These results are stored in tables. Then at each collision, $\cos^2(bI)$ is obtained from these tables. This scheme eliminates the need to find s_{max} for each collision, minimizing the amount of arithmetic performed during the calculation of an ion's trajectory, while retaining accuracy.

Tables for the Universal potential over a wide range of energies and impact parameters are provided for immediate use in TSUPREM-4. They span the normalized energy range of $10^{-5} \le \varepsilon \le 100$ and the normalized impact parameter range $0 \le b_n \le 30$. For $\varepsilon > 100$, the Coulomb form Equation 3-507 is used. Values of $\varepsilon < 10^{-5}$ are not encountered for ion-atom combinations of interest at energies above the energy at which the ion is assumed to be stopped (10 eV). For values of $b_n > 30$, the ion is assumed to be undeflected.

Amorphous Implant Calculation

This section describes how the binary scattering theory of the previous section is used to calculate ion trajectories in an amorphous solid. TSUPREM-4 calculates a number of ion trajectories that can be specified using the **N.ION** parameter on the **IMPLANT** statement. The implant species can be any impurity (including user-defined impurities) for which the required information (atomic number and weight, electronic stopping powers, and so on) has been specified.

The calculation of ion trajectories proceeds as follows. Assume an ion with kinetic energy E_0 hits a target with an angle θ_0 with respect to the target normal.

The surface of the target is assumed to be at $y = 0$, with y increasing vertically into the target. The incident energy E_0 can be set on the **IMPLANT** statement using the **ENERGY** parameter. The incident angle θ_0 can be specified on the **IMPLANT** statement using the **TILT** parameter.

Given the atomic density N_{dens} for the target material, the mean atomic separation between atoms in the target is $1/(N_{dens})^{1/3}$. The ion is assumed to travel a distance:

$$L = 1/(N_{dens})^{1/3} \qquad \text{Equation 3-508}$$

between scattering events. As the ion enters the target material, it approaches the first target atom with impact parameter b, defined in the previous section. The probability of finding a target atom between b and $b + \delta b$ is given by:

$$w(b)\delta b = 2\pi N_{dens}^{2/3} b\ \delta b \qquad \text{Equation 3-509}$$

for $b < 1/\sqrt{\pi N_{dens}^{2/3}}$. If R_{rand} is a uniformly distributed random number between 0 and 1, then the probability distribution gives:

$$b = \sqrt{\frac{R_{rand}}{\pi N_{dens}^{2/3}}} \qquad \text{Equation 3-510}$$

as described in Reference [66].

Nuclear Stopping

Given the above definitions, the algorithm for calculating the energy loss through nuclear collisions experienced by the ion proceeds as follows:

1. A random number between 0 and 1 is chosen.
2. The normalized impact parameter for this collision is calculated from Equations 3-499 and 3-510:

$$b = \frac{1}{a_u}\sqrt{\frac{R_{rand}}{\pi N_{dens}^{2/3}}} \qquad \text{Equation 3-511}$$

3. The ion energy, E_0, is normalized to:

$$\varepsilon = \frac{a_u E_0}{(1 + M_1/M_2) Z_1 Z_2 k_1} \qquad \text{Equation 3-512}$$

   from Equations 3-493 and 3-500.

4. Now the value of $\cos^2(bI)$ can be obtained from the tables, and Equation 3-491 gives the energy loss due to nuclear scattering:

$$\Delta E_0 = E_0 \frac{4 M_1 M_2}{(M_1 + M_2)^2} \cos^2(bI) \qquad \text{Equation 3-513}$$

This procedure is repeated for each collision event.

Electronic Stopping

The ion also loses energy by inelastic electronic processes, which include both nonlocal and local stopping power. At low energies this is modeled by:

$$\Delta E_e = x^{nl} \cdot \Delta E_e^{nl} + x^{loc} \cdot \Delta E_e^{loc} \qquad \text{Equation 3-514}$$

$$\Delta E_e^{nl} = L \cdot N_{dens} \cdot S_e \qquad \text{Equation 3-515}$$

$$\Delta E_e^{loc} = \frac{S_e}{2\pi a^2} \cdot \exp(-p/a) \qquad \text{Equation 3-516}$$

$$x^{nl} = \min(\mathbf{NLOC.PRE} \cdot \varepsilon^{\mathbf{NLOC.EXP}}, \mathbf{NLOC.MAX}) \qquad \text{Equation 3-517}$$

$$x^{nl} + x^{loc} = 1 \qquad \text{Equation 3-518}$$

$$S_e = \mathbf{NLOC.K} \cdot \mathbf{ES.RAND} \cdot E_0^{\mathbf{ES.F.RAN}} \qquad \text{Equation 3-519}$$

$$a = \mathbf{LOC.FAC} \cdot \frac{a_U}{0.3} \qquad \text{Equation 3-520}$$

where L is the free flight path between collisions, p is the impact parameter, ε is the scaled dimensionless energy, E_0 is the ion energy, and a_U is the universal screening length. **`NLOC.PREN`**, **`NLOC.EXP`**, **`NLOC.MAX`**, **`NLOC. K`**, **`ES.RAND`**, **`ES.F.RAN`**, and **`LOC.FAC`** can be defined on the **`IMPURITY`** statement.

In the absence of a specification using an **`IMPURITY`** statement, or if the specified values are zero, values for these parameters are obtained from the coefficient file. If no value or a value of zero is specified in the coefficient file for parameters **`ES.RAND`** and **`ES.F.RAN`**, the default values are given by Reference [68].

$$\mathbf{ES.RAND} = \frac{1.212\, Z_1^{7/6}\, Z_2}{\left[Z_1^{2/3} + Z_2^{2/3}\right]^{3/2} M_1^{1/2}} \qquad \text{Equation 3-521}$$

$$\mathbf{ES.F.RAN} = 0.5 \qquad \text{Equation 3-522}$$

where Z_1 is the ion atomic number and Z_2 is the composite target atomic number. For boron and phosphorus in silicon, the default values of **`ES.RAND`** have been set to 2.079 and 2.5, respectively Reference [66]. For light ions ($Z_1 \leq 5$) and α-particles, the value for electronic stopping given by Equation 3-521 is very crude. Experimental values should be specified wherever possible References [64], [69], and [70], and [72], using **`ES.RAND`** and **`ES.F.RAN`** on the impurity statements.

Electronic Stopping at High Energies

At energies above **`ES.BREAK`**, Equation 3-522 is replaced by:

$$\Delta E_e = LN_{dens}\mathbf{ES.RAND} \cdot \mathbf{ES.BREAK}^{\mathbf{ES.F.RAND}} \left(\frac{E_0}{\mathbf{ES.BREAK}}\right)^{\mathbf{ES.F.H}}$$

Equation 3-523

where **`ES.BREAK`** and **`ES.F.H`** are parameters on the **`IMPURITY`** statement. *Note* that the high energy stopping model is used only when **`ES.BREAK`** is nonzero, i.e., the model can be disabled by setting **`ES.BREAK`**=0.

Total Energy Loss and Ion Deflection

The total change in energy of the ion after the i^{th} collision is the sum of nuclear energy loss, Equation 3-513, and electronic energy loss, Equation 3-514.

$$E_i = E_{i-1} - \Delta E_n - \Delta E_e \qquad \text{Equation 3-524}$$

The corresponding angle through which the ion is scattered is given by:

$$\theta = \cos^{-1}\left[\frac{1-\frac{1}{2}(1+M_2/\ M_1)\Delta E_n/\ E_0}{\sqrt{1-\Delta E_n/\ E_0}}\right] \qquad \text{Equation 3-525}$$

Note:

For $\Delta E_n/\ E_0 \ll 1$, θ ***approaches zero.***

Ion Beam Width

The incident angle used in the MC calculation given by **`TILT`** can be varied about its nominal value by specifying the **`BEAMWIDT`** parameter. For a nonzero value of **`BEAMWIDT`**, the angle used to calculate the incident velocity of each ion is varied about its nominal value, **`TILT`**, by the addition of (**`BEAMWIDT`**) R_{rand} where R_{rand} is selected from a uniform distribution of random numbers from -1/2 to 1/2.

Crystalline Implant Model

The calculation described in the previous section is for amorphous layers. When the **`CRYSTAL`** parameter is set during a MC ion implant calculation, silicon layers are treated with a more sophisticated calculation that incorporates the crystal structure of the silicon lattice. This calculation is intended to model channeling, which is the preferential penetration of implanted ions along crystal axes.

The calculation proceeds as in the amorphous case except that rather than selecting the collision of the implanted ion with target atoms based on the density of the target material and a random number, the simulation determines an impact parameter based on the implanted ion's position relative to sites on an idealized lattice. This is accomplished by discretizing the silicon lattice and calculating the impact parameter for each of eight lattice sites within each discretization cell. A single site is then selected for collision, and the energy loss is calculated as in the amorphous case.

Channeling

The effect of the **`TILT`** parameter is much more pronounced for implants into crystalline silicon than into amorphous silicon. Axial implants (obtained by setting **`TILT`** to zero) show an enhanced penetration due to channeling.

Channeling occurs naturally due to the inclusion of the structure of the silicon lattice. For a given incident ion energy, the critical angle is a measure of the angle of deviation from a crystal axis that is required to prevent an ion from being channeled along that axis. For channeling along the <100> axial direction, the critical angle is given by:

$$\psi_{100} = \psi_0 E^{-\mathtt{CRIT.F}} \qquad \text{Equation 3-526}$$

where:

$$\psi_0 = 2.314 \times \left[a_0 Z_1 Z_2 a_{conv}^{-3}\right]^{\mathbf{CRIT.F}} \qquad \text{Equation 3-527}$$

a_{conv} is the conventional cell dimension of 5.431Å for silicon and **CRIT.F** is a parameter on the **IMPLANT** statement. The critical angle for the <110> direction is given by Reference [73].

$$\psi_{110} = \mathbf{CRIT.110} \times \psi_{100} \qquad \text{Equation 3-528}$$

where **CRIT.110** is a parameter on the **IMPLANT** statement.

Due to the lower electron density in the center of an axial channel, the electronic stopping of ions that are channeled is generally less than that for ions travelling in random directions. TSUPREM-4 allows explicit specification of the electronic stopping along the <100> and <110> directions. The channeled electronic stopping is selected whenever the ion is traveling within an angle with respect to a particular crystal axis given by:

$$\psi_{100} = \mathbf{CRIT.PRE} \cdot \psi_{100} \qquad \text{Equation 3-529}$$

and:

$$\psi_{110} = \mathbf{CRIT.PRE} \cdot \psi_{100} \qquad \text{Equation 3-530}$$

for the <100> and <110> directions, respectively. The factor **CRIT.PRE** is a fraction of the critical angle that can be specified on the **IMPLANT** statement.

The electronic stopping along the <100> and <110> crystal axes can be specified with the parameters **ES.100**, **ES.F.100**, **ES.110**, and **ES.F.110**. When the implanted ion is determined to be travelling along a <100> crystal axis, as described above, the values of **ES.100** and **ES.F.100** are substituted for **ES.RAND** and **ES.F.RAN**, respectively, in Equations 3-521 and 3-523 to determine the energy loss of the ion through electronic stopping.

For nonzero values of **TILT**, the penetration of implanted ions through silicon depends on the crystalline orientation of the *x* axis of the simulation space and the value of **ROTATION** because of planar channeling. The orientation can be set on the **INITIALIZE** statement with the **X.ORIENT** or **ROT.SUB** parameters.

For the MC code, the multiple collision should be handled carefully, because the traditional multiple collision algorithm does not conserve both energy and momentum simultaneously Reference [65], yet increases the computational time significantly due to the calculation of multiple collision partners. In TSUPREM-4, a novel approach is used to simulate the channeling effect. In this new approach, a small scattering angle is identified to be that of channel ions. This angle is further reduced to reflect the nature of multiple collision. So, if the scattering angle is: $\theta <$ **CHAN.CRI**, it is replaced by an effective scattering angle, $\theta_{eff} = \theta /$**CHAN.FAC**.

The default parameters are **CHAN.CRI** = 11.54, and **CHAN.FAC** = 2.0 for all species. These parameters can also be changed on the **IMPURITY** statement. Increasing **CHAN.CRI** and/or **CHAN.FAC** causes more channeling.

Lattice Temperature

The temperature of the lattice can be specified using the **TEMPERAT** parameter. When the **VIBRATIO** parameter is set, the temperature specified by **TEMPERAT** is used in a Debye calculation to determine the rms vibration amplitude of silicon lattice atoms from their sites. This calculation can be superseded by specifying the rms vibration amplitude explicitly using the **X.RMS** parameter. At each collision site, the displacement of silicon lattice atoms from their idealized sites is selected as (**X.RMS**) R_{norm} where R_{norm} is a value selected from a normal distribution of random numbers with unity standard deviation.

Empirical observations of implant profiles produced by TSUPREM-4 have shown that better agreement with measured profiles is achieved by neglecting the dechanneling effect of lattice vibrations for each collision at which the ion’s energy is above some limit. This limit can be specified using the **E.LIMIT** parameter. If no value for **E.LIMIT** is specified, TSUPREM-4 neglects lattice vibrations if the ion’s energy is greater than 5 keV for boron or 40 species.

Lattice Damage

TSUPREM-4 contains a damage calculation that models the transition from crystalline to amorphous material which occurs in silicon as an implant proceeds. Other workers have modeled the crystal-to-amorphous transition by a critical energy model Reference [74]. When the **DAMAGE** parameter is specified, a calculation is performed to determine the trajectories of silicon lattice atoms that are knocked from their sites in the lattice by collisions with implanted ions. A silicon atom is assumed to be knocked from its site when it absorbs an energy greater than a damage threshold from a collision. The value of this damage threshold can be set using the **THRESHOL** parameter on the **IMPLANT** statement. The silicon atoms freed from the lattice can in turn knock other atoms from their sites so that cascades of damage result Reference [48]. TSUPREM-4 calculates the trajectories of these knock-ions with the same detail as the implanted ions. The program maintains efficiency by calculating only a weighted fraction of these secondaries. The fraction can be specified using the **REC.FRAC** parameter; setting this parameter equal to unity results in the calculation of trajectories for all secondaries as they are generated.

The output from the damage calculation produces information in the form of vacancy and interstitial profiles Reference [75]. A vacancy is assumed to be formed whenever a lattice atom is knocked from its site. An interstitial is assumed to be formed whenever a silicon lattice atom that has been knocked from its site comes to rest. The profiles of interstitials and vacancies that result are retained as an initial condition for subsequent diffusion steps if the **DAMAGE** parameter is specified.

Damage Dechanneling

The accumulated damage has significant effect on the destination of the subsequent ions, thus altering the shape of the impurity profiles. This effect is known as *damage dechanneling*. TSUPREM-4 handles this problem by switching from the crystal model to the amorphous model based on the damage that has accumulated in the substrate. The probability for the selection of the amorphous model is proportional to the local interstitial concentration and a random number call. Silicon is treated as amorphous when:

$$R_{rand} < \frac{C_I(x)}{\texttt{DISP.FAC} \cdot \texttt{MAX.DAMA}}$$

Equation 3-531

where $C_I(x)$ is the local interstitial concentration, and **MAX.DAMA** is the maximum damage allowed. **MAX.DAMA** can be specified on the **IMPLANT** statement, while **DISP.FAC** can be specified on the **IMPURITY** statement. Increasing **DISP.FAC** makes the profiles more like those implants into amorphous materials.

Note:

The amorphization process is not explicitly simulated by TSUPREM-4. However, for the MC model, by common practice, when a critical amount of damage is accumulated in a certain region, a crystal/amorphous phase transition is assumed to occur in this region. For silicon implants, this critical damage is approximately 10% of the lattice density. Thus, if the interstitial concentration reaches more than 5e21 /cm^{-3} ***for silicon, this region is considered to be amorphized. Using this criteria, TSUPREM-4 can reasonably predict the onset of amorphization and the thicknesses of the amorphous layers for high dose implants.***

Damage Annealing

TSUPREM-4 contains a model for the self-annealing of the damage produced during implantation. Experimental measurements of the dose required to amorphize silicon as a function of temperature show an increase with increasing temperature Reference [76]. This is modeled in TSUPREM-4 by assuming that a temperature-dependent fraction of the point defects is self-annealing. A temperature-dependent value for **THRESHOL** has been empirically determined. This compensates for recombination by eliminating calculation of trajectories of ions that eventually recombine Reference [77].

Besides the *in situ* annealing that occurs during ion implantation, the damage which accumulates in the silicon material is annealed during subsequent high temperature processing steps. Until such a step, the damage produced by an implant step serves as an initial condition for subsequent implantations.

Number of Ions

The results of the MC calculation are subject to statistical variation due to the finite number of particles that make up the solution. The resulting noise in the solution can be reduced by increasing the value of the **N.ION** parameter above its default value of 1000. The solution time is directly proportional to this value. The solution time can be reduced at the expense of a higher statistical variation in the results by reducing the value of **N.ION**.

BF_2 Implantation

The old TSUPREM-4 MC model does not model the dissociation of BF_2 ions. If `BF`$_2$ is specified with `OLD.MC` on the `IMPLANT` statement, the BF_2 implant is approximated by a boron implant with an energy of 0.2215 times the user-specified implant energy. The damage is scaled by 49/11 to account for the fluorine ions. (The scaling is omitted if `V.COMPAT`=6.4 is specified on the `OPTION` statement.)

Note:

In contrast to old TSUPREM-4 MC model, Taurus MC Implant model does simulate the dissociation of BF2 ions and calculate the trajectories of both boron and fluorine.

Implant Damage Model

The implant damage model in TSUPREM-4 accounts both for the silicon atoms knocked out of lattice sites and for interstitials produced when silicon atoms are displaced by implanted ions (the “plus one” model). In addition, the effects of amorphization are taken into account, and an analytical model of point defect recombination has been included to speed up subsequent diffusion steps. The implant damage model works with both the analytical and MC implantation models.

There are two slightly different versions of the implant damage model in TSUPREM-4. The newer version, the *cumulative damage model*, automatically accumulates the damage produced by successive implants and simulates defect recombination and amorphous regrowth at the start of the first high-temperature step following a series of implants. The *old damage model* simulates defect recombination and amorphous regrowth at every implant step by default.

Damage Produced During Implant

Both models start by calculating the damage produced during each implantation step. The concentrations of interstitials and vacancies produced by the impact of implanted ions (or recoiling silicon atoms) are denoted by I_F and V_F, respectively. When the analytical implant model is used, I_F and V_F are calculated according the model of Hobler and Selberher Reference [58] and are assumed to be equal at every point in the structure. When the MC model is used, I_F and V_F may be slightly different because most displaced lattice ions are knocked forward while a few are scattered out of the structure entirely. If the increase in concentration of the implanted species is ΔC, then the number of added interstitials and vacancies due to the implant is:

$$\begin{aligned} \Delta I &= \texttt{D.SCALE} \times I_F + f_{pl} \times \Delta C \\ \Delta V &= \texttt{D.SCALE} \times V_F \end{aligned} \qquad \text{Equation 3-532}$$

$$f_{pl} = \begin{cases} \texttt{D.PLUS}, & \text{if } \texttt{D.PLUS} \text{ specified} \\ f_{pl\infty} f_{pld}, & \text{otherwise} \end{cases} \qquad \text{Equation 3-533}$$

$$f_{pl\infty} = 1 + \mathbf{D.PHDF}\, m^{\mathbf{D.PME}} E^{\lambda_\infty}$$ Equation 3-534

$$f_{pld} = \mathbf{D.P1} + \mathbf{D.PCONST}\, m^{\mathbf{D.PDME}} E^{\lambda_d} \left(\frac{\mathbf{D.PNORM}}{d}\right)^{\mathbf{D.PDE}} R_p^{\mathbf{D.PDRE}}$$

Equation 3-535

where:

$$\lambda_\infty = \mathbf{D.PLF}\, m^{\mathbf{D.PLME}}$$ Equation 3-536

$$\lambda_d = \mathbf{D.PDLF}\, m^{\mathbf{D.PDLME}}$$ Equation 3-537

and:

- d and E are the dose (per cm^2) and energy (in keV) of the implant.
- R_p is the projected range (in *nm*).
- m is the mass of the implanted ion (in atomic mass units).
- **D.PNORM** and **D.PDE** are parameters on the **IMPURITY** and/or **MATERIAL** statements (see below).
- **D.PHDF**, **D.PME**, **D.PCONST**, **D.PDME**, **D.PDRE**, **D.PLF**, **D.PLME**, **D.PDLF**, and **D.PDLME** are parameters on the **MATERIAL** statement.
- **D.PLUS** and **D.P1** are parameters on the **IMPLANT** statement.

The parameter **D.SCALE** (on the **IMPLANT** statement) adjusts the Frenkel pair density to account for uncertainties in the damage creation models. You can use it to change the simulated amorphization depth of amorphizing implants.

+1 and Effective +n Damage Models

The ΔC term accounts for the fact that each implanted ion displaces a silicon atom, producing an interstitial. The recombination of vacancies at the surface leaves an additional excess of interstitials, which is accounted for by the factor f_{pl}. This excess can be specified directly by the **D.PLUS** parameter on the **IMPLANT** statement. If **D.PLUS** is not specified, an *effective +n* model Reference [78] is used to determine the number of excess interstitials as a function of energy, ion mass, and projected range. Higher energy or ion mass increases the separation between the generated interstitials and vacancies, increasing the chance that vacancies will recombine at the surface before recombining with an interstitial. Increasing the range of the implant has the opposite effect. The effectiveness of the I-V recombination decreases at low doses References [79] and [80].

The effective +n model is designed for use with the analytical implant model. For MC implants the **D.PHDF** and **D.PCONST** terms are ignored. (The MC damage model inherently includes the effects modeled by the new analytical model.)

The parameters **D.PNORM** and **D.PDE** can be specified on either the **IMPURITY** or **MATERIAL** statement. If either parameter is given a nonzero value on an **IMPURITY** statement, the values on the **IMPURITY** statement are used for that impurity. If neither value is specified on an **IMPURITY** statement (or if both values are set to zero), the values specified on the **MATERIAL** statement are used.

This model is a superset of several models that have been published in the literature; by setting the parameters appropriately, any of these published models can be used. By default, the parameters are set to give the model of Hobler and Moroz Reference [81]:

```
MATERIAL MAT=SILICON D.PHDF=0.0905 D.PME=0.85 +
  D.PLF=-2 D.PLME=-0.5 D.PCONST=3.25 D.PDME=-0.5 +
  D.PDLF=-2 D.PDLME=-0.5 D.PNORM=3e12 D.PDE=1 +
  D.PDRE=0
```

Another choice of parameters gives the model of Pelaz *et al.* Reference [78] with added dose dependence References [79] and [80], as implemented in TSUPREM-4 1999.4:

```
MATERIAL MAT=SILICON D.PHDF=0 D.PCONST=0.42 +
  D.PDME=0.5 D.PDLF=0.5 D.PDLME=0 D.PNORM=1e14 +
  D.PDE=0.5 D.PDRE=-0.75
```

The original model of Pelaz *et al.* is obtained with the parameters

```
MATERIAL MAT=SILICON D.PHDF=0 D.PCONST=0.42 +
  D.PDME=0.5 D.PDLF=0.5 D.PDLME=0 D.PNORM=0 +
  D.PDRE=-0.75
```

For BF_2 implants the mass and energy of the boron ion are used, and the dose is multiplied by three to account for the effect of the fluorine atoms.

You can use **D.P1** on the **IMPLANT** statement to adjust the amount of TED produced by an implant; it defaults to 1.0 if not specified.

Profiling Implant Damages

In analytic implants, +n model can be applied to not only interstitials but also vacancies with the optional shift, and I_F displacement to V_F can be emulated by specifying the distance. Equation 3-532 is modified to:

$$\Delta I = \mathbf{D.SCALE} \times V_F(x - \mathbf{DX.FP},\ y - \mathbf{DY.FP}) + f_{pl} \times \Delta C(x - \mathbf{DX.IPLUS},\ y - \mathbf{DY.IPLUS}) \quad \text{Equation 3-538}$$

$$\Delta V = \mathbf{D.SCALE} \times V_F(x,\ y) + \mathbf{V.PLUS} \times \Delta C(x - \mathbf{DX.VPLUS},\ y - \mathbf{DY.VPLUS}) \quad \text{Equation 3-539}$$

Cumulative Damage Model

The cumulative damage model calculates a quantity called *damage* that represents the total damage at any point in the structure from all implants since the last high-temperature processing step. An implant step increases the damage by:

$$\Delta D = \min(\Delta I,\ \Delta V) \quad \text{Equation 3-540}$$

The interstitial, vacancy, and damage concentrations following an implant are then given by:

$$\begin{aligned} I &= I_0 + \Delta I - I_{DDC} - \Delta D \\ V &= V_0 + \Delta V - V_{DDC} - \Delta D \end{aligned}$$

Equation 3-541

$$D = D_0 + \Delta D$$

Equation 3-542

where I_0, V_0, and D_0 are the concentrations of interstitials, vacancies, and damage, respectively, before the implant, and I_{DDC} is the concentration of captured interstitials in dopant-defect clusters.

$$\begin{aligned} I_{DDC} = {} & \left(\mathbf{IFRACM} \cdot \ \mathbf{C1.INI.F} + \sum \frac{\mathbf{DDC.I.N} \cdot \ \mathbf{DDCFRACC}}{\mathbf{DDC.D.N}}\right) \\ & \cdot \left(\frac{\mathbf{DDCTOT.C}}{\mathbf{C1.INI.F} + \mathbf{C2.INI.F} + \sum \mathbf{DDCFRACC}} \cdot \ \Delta C\right) \end{aligned}$$

Equation 3-543

$$\begin{aligned} V_{DDC} = {} & \left(\mathbf{VFRACM} \cdot \ \mathbf{C2.INI.F} + \sum \frac{\mathbf{DDC.V.N} \cdot \ \mathbf{DDCFRACC}}{\mathbf{DDC.D.N}}\right) \\ & \cdot \left(\frac{\mathbf{DDCTOT.C}}{\mathbf{C1.INI.F} + \mathbf{C2.INI.F} + \sum \mathbf{DDCFRACC}} \cdot \ \Delta C\right) \end{aligned}$$

Equation 3-544

Most of the Frenkel pair damage produced during implantation will recombine during the first few seconds of the first post-implant annealing step. This is modeled by setting $D = 0$ at the start of the first diffusion step following an implant. Incomplete recombination of Frenkel pair damage is modeled with the **`D.RECOMB`** parameter on the **`DIFFUSION`** statement:

$$\begin{aligned} I_1 &= I_0 + \mathbf{D.RECOMB} \times D_0 \\ V_1 &= V_0 + \mathbf{D.RECOMB} \times D_0 \end{aligned}$$

Equation 3-545

where I_0, V_0, and D_0 are the concentrations of interstitials, vacancies, and damage, respectively, before the start of the diffusion.

Crystalline regrowth of silicon amorphized by implant damage normally occurs during the ramp-up phase of the first post-implant annealing step. This is modeled by setting the interstitial and vacancy concentrations to their equilibrium values in the amorphized region. The transition between the crystalline and amorphized regions is smoothed by

$$s = 0.5 - 0.5 \times \tanh[\mathbf{DAM.GRAD} \times \ln(D_0 / \ \mathbf{MAX.DAMA})]$$

Equation 3-546

$$I = I^* + s(I_1 - I^*)$$
$$V = V^* + s(V_1 - V^*)$$

Equation 3-547

where **`MAX.DAMA`** and **`DAM.GRAD`** are parameters specified for silicon on the **`MATERIAL`** statement. When the **`ACT.TRAN`** model is used and **`^CL.INI.A`** has been specified for an implanted dopant in silicon, the same smoothing function is used to smooth the transition from no activation (in the crystalline region) to full activation (in the amorphized and regrown region) of the dopant.

The cumulative damage model is used whenever a nonzero value for **`MAX.DAMA`** has been specified for silicon on the **`MATERIAL`** statement. It is the preferred model and is enabled by default.

Conservation of Total Defect Concentrations

Some of the point defects introduced during ion implantation will combine with dopants or other defects to produce pairs or clusters. These point defects are removed from the population of free point defects to ensure that the total number of added defects (free and otherwise) is given by the models described above. Thus the reported number of (free) point defects following an implant may be less than the values calculated by the models above; the remaining defects are contained in dopant-defect pairs, small clusters of defects, or other defect-absorbing complexes that are modeled by the program.

The incorporation of defects into dopant-defect pairs only occurs when **`PD.PTIME`** is enabled (i.e., when **`PD.FULL`** is used). Thus the action of the damage model depends on the models in effect at the time of the implant. For correct simulation of TED with the **`PD.FULL`** model, **`PD.FULL`** should be specified *before* the implant. In general, parameters that affect the pairing or clustering of point defects should not be changed between an implant with damage and the subsequent annealing step.

Using the Implant Damage Model

The default parameter values are expected to give reasonable results in most cases. However, you might obtain improved results with the following modifications:

- You can simplify the calculation by setting **`D.SCALE`**=0.0. The result is the scaled plus one model without the effects of amorphization.
- You can modify the amorphization threshold by reducing **`MAX.DAMA`**. This causes amorphization effects to appear at lower doses. You can also change the amorphization depth for a particular implant by modifying **`D.SCALE`**.
- **`D.SCALE`** can be reduced to model the effects of self-annealing during implantation.
- **`D.PLUS`** can be adjusted to change the amount of transient-enhanced diffusion. This may be necessary for modeling situations in which interstitials are bound in clusters after an implantation or where an imbalance between interstitials and vacancies is produced by recombination at the surface.
- *Old model only*: To model the accumulation of damage in consecutive implants, you should specify **`^D.RECOMB`** in all implants except the last. This allows the full damage produced by each implant to accumulate, with

recombination occurring only after the last one. Damage accumulation is automatic with the cumulative model.

- The abruptness of the amorphous/single-crystal interface can be adjusted with the **DAM.GRAD** parameter on the **MATERIAL** statement. Smaller values give a more extended interface while larger values give a more abrupt interface; typical values are between 1 and 100. **DAM.GRAD** can be used to reduce the sensitivity of the amorphization model to the grid spacing.

The damage model can produce concentrations of point defects that are much greater than those produced by oxidation. Accurate simulation of diffusion in the presence of such high defect concentrations requires that the **PD.FULL** diffusion model be used. *Note* that very small time steps are required in the initial stages of diffusion following an implant with damage. It is suggested that an initial time step on the order of 10^{-6} minutes be used. If a larger initial time step is used, there is a delay during the first step while the start of the post-implant transient is simulated. Use of **D.RECOMB** reduces the peak point defect concentration and lessens the need for small time steps and the **PD.FULL** model. If the **PD.FERMI** model is in effect when an implant with **DAMAGE** is specified, the **PD.TRANS** model is automatically enabled.

Boundary Conditions for Ion Implantation

Boundary Conditions for OLD.MC Implant

Ion implantation does not obey the reflecting boundary conditions that are usually used at the left and right edges of the structure. Instead, the analytic implantation extends the structure at a reflecting boundary out to infinity, while the MC model uses the boundary condition specified by your choice of the parameters, **VACUUM**, **PERIODIC**, or **REFLECT**. Thus there a loss of accuracy in the implanted profile unless the lateral spread of the implant distribution is small compared to the distance between the edge of the structure and the nearest mask edge. In some cases, you may need to reflect the structure before implantation and truncate it afterwards to ensure the accuracy of the implanted profile.

For MC implants, three choices of boundary conditions are available.

- The default, **PERIODIC**, specifies that ions leaving one side of the structure re-enter on the other side, with the same velocity.

  This condition gives accurate answers for 1-D structures and for structures where the sequence and thickness of layers are the same at the left and right edges.

- **REFLECT** specifies that ions hitting the edge of the structure are reflected back into the structure.

  This condition is accurate for **TILT=0** implants and for pairs of implants with opposite tilts.

- **VACUUM** specifies that ions leaving the structure through the sides are lost.

  This boundary condition is a poor approximation for most structures.

There are many situations (particularly single-tilted implants into 2-D structures) where none of the available boundary conditions is perfectly accurate. As with the analytical implant model, you lose some accuracy unless the lateral spread of the implant distribution is small compared to the distance between the edge of the structure and the nearest mask edge. Again, you may need to reflect the structure before implantation and truncate it afterwards to ensure the accuracy of the implanted profile.

Boundary Conditions for Taurus Analytic and Taurus MC Implant

Taurus analytic and Taurus MC implant models support the same set of boundary conditions: **`EXTEND`**, **`PERIODIC`**, **`REFLECT`**, or **`VACUUM`**. The default is **`EXTEND`** for both analytic and MC models.

In contrast to old MC boundary conditions, TSUPREM-4 creates additional mesh elements to account for the effects of image structures implied by the **`EXTEND`**, **`PERIODIC`**, or **`REFLECT`** boundary conditions. These additional mesh elements are created automatically and deleted at the end of the **`IMPLANT`** command.

The Taurus Analytic Implant model constructs additional mesh edges for each material interface that touches the simulation domain boundary. This construction is performed for all boundary conditions, except vacuum; the new edges satisfy the boundary condition. The constructed surface elements extend the simulation domain by a distance that is adequate to account for lateral straggling and the specified tilt angle. This distance depends on the implant conditions and is determined automatically. If necessary, you can override the default distance calculation using the **`MARGIN`** parameter in the **`IMPLANT`** command.

When the Taurus MC Implant model is selected, the construction of additional mesh elements depends on the boundary conditions. For **`EXTEND`** boundary conditions, the construction is similar to that used by the analytic model except that the entire mesh (including volume elements) and all data on the mesh are extended. The extension length is dependent on the simulation domain size and tilt/rotation angles, and is determined automatically. Like the analytic implant, you can use the **`MARGIN`** parameter to override the default extension length.

For **`REFLECT`** boundary conditions, a complete copy of the mesh is created by reflecting the structure once in the x-direction. This produces a structure with periodic symmetry, and periodic conditions are then applied. For **`PERIODIC`** and **`VACUUM`** boundary conditions, no additional mesh elements are needed with the Taurus MC model.

Note:

For Monte Carlo implant, for boundary condition REFLECT, there will be a performance penalty due to the expansion of the simulation domains. There is also likely a performance penalty with the boundary condition EXTEND for non-zero tilt implants. If the accuracy near boundary is not critical, set the MARGIN=0.0 to eliminate the automatic extension.

MC Implant into Polysilicon

In older versions of TSUPREM-4, Taurus MC treats polysilicon as amorphous material only. In this version, Taurus MC has the options to treat polysilicon as amorphous, polycrystalline, or crystalline silicon. A polycrystalline material is characterized by its orientation and grain size. In this version, for polysilicon, both poly orientation (<100>, <110>, or <111>) and poly grain size can be specified in the **MATERIAL** statement.

To specify the state of the crystallinity, use the boolean parameters (**AMORPH**, **POLY.MC**, or **CRYSTAL**) in the **MATERIAL** statement. The default crystallinity is amorphous, which provides the backward compatibility. To change the grain size and the scaling factor, use parameters **POLY.GSZ** and **POLY.FAC** in the **MATERIAL** statement. To change the poly orientation, use integer parameter **POLY.ORI** (one of 100, 110, or 111) in the **MATERIAL** statement.

The polysilicon MC implant model works by frequently switching between the crystal model and the amorphous model. The probability of switching from the crystal model to the amorphous model is determined by the accumulative path length (*pathlength*) and polysilicon grain size (**POLY.GSZ**).

The model switches from the crystal to the amorphous, if:

pathlength > (`POLY.FAC*POLY.GSZ`)

or:

pathlength > *rand(x)* * (`POLY.FAC*POLY.GSZ`)

rand(x) is a uniform random number between 0 and 1. **POLY.FAC** and **POLY.GSZ** are specified in the **MATERIAL** statement.

After processing a collision for the amorphous model, the *pathlength* is reset to zero, and the crystal model is selected. Then it starts accumulating the *pathlength* again. The model used for the next collision is again determined by the same rules. This process is repeated until the ion exits the polysilicon region.

Note:

In OLD.MC, the polysilicon MC implant model is activated by specifying the grain size POLY.GSZ in the IMPLANT statement, which is not used in Taurus MC. Instead, for Taurus MC, POLY.GSZ is specified in the MATERIAL statement. Polysilicon model is selected by specifying its crystallinity POLY.MC in MATERIAL statement.

MC Implant into Hexagonal Silicon Carbide

Silicon carbide (SiC) can be specified either by the **INITIALIZE** or **REGION** statements. For example:

- **INITIALIZE MATER** = sic
- **REGION MATER** = sic **XLO** = left **XHI** = right **YLO** = top **YHI** = bottom
  INITIALIZE

Both of these commands create silicon carbide material in the structure.

In this new version of TSUPEM-4, Taurus MC supports crystalline SiC with the hexagonal lattice. In the hexagonal system, there are four crystallographic axes, i.e., three *a* axes (*a1*, *a2*, *a3*) forming a plane and ***c*** axis which is normal to the plane. The crystallographic planes and directions are normally described with four Miller indices (*hkil*). However, for the hexagonal system, since the sum of the first three indices is zero, the third index sometimes can be omitted. For example, <11-20> direction can be represented as <110>.

Two SiC wafer orientations (<0001> and <11-20>) are supported. These wafer orientations can be specified as <100> and <110> in the **INITIALIZE** command, respectively. For (0001) SiC wafer, the primary flat orientation is <10-10>. For (11-20) SiC wafer, the primary flat orientation is <0001>.

SiC exists in many different crystal structures, called polytypes. All polytypes have a hexagonal frame with a carbon atom situated above the center of a triangle of Si atoms and underneath a Si atom belonging to the next layer. The difference among the polytypes is the stacking sequence between the succeeding double layers of carbon and silicon atoms. For example, 2H-SiC, 4H-SiC, and 6H-SiC have the AB, ABCB, and ABCACB stacking sequences, respectively. 3C-SiC is the only form of SiC with a zinc-blend lattice structure. Four different SiC polytypes are supported in Taurus MC. These polytypes (**SIC.2H**, **SIC.3C**, **SIC.4H**, or **SIC.6H**) can be specified in the **MATERIAL** statement.

Examples:

```
$ Select 4H-SiC
MATERIAL MAT=SiC SIC.4H

$ Set (11-20) SiC wafer orientation
INITIAL MAT=SiC <110>

$ Do Taurus MC implant
IMPLANT IMP=aluminum DOSE=1E14 ENERGY=5.0
         TILT=0 ROT=0 N.ION=1000 MONTECARLO
```

Fields to Store Implant Information

The built-in fields **DIMPLANT**, **IIMPLANT** and **VIMPLANT** stores implanted dopant concentration, interstitial and vacancies by implant damage respectively.

$$C_{\mathtt{IIMPLANT}} = I_{FP} + I_{plus} \qquad \text{Equation 3-548}$$

where I_{FP} and I_{plus} are the shifted **DAMAGE** concentration by **DX.FP** and **DY.FP**, and the shifted interstitial profile from +n model by **DX.IPLUS** and **DY.IPLUS**, respectively.

$$C_{\mathtt{VIMPLANT}} = \mathrm{DAMAGE} + V_{plus} \qquad \text{Equation 3-549}$$

where V_{plus} is the shifted vacancy profile from +n model by **DX.VPLUS** and **DY.VPLUS**, respectively.

DIMPLANT, **IIMPLANT**, and **VIMPLANT** are lost after regrowth. To keep the fields after regrowth, you must copy the fields to the intermediates to be fixed. For example:

```
MESH DY.SURF=0.002 LY.SURF=0.002 DY.ACT=0.01 LY.ACT=0.5
INIT BORON=1E16
INTERMED NAME=BIMP1 EXPR=DIMPLANT
INTERMED NAME=BIMP2 EXPR=DIMPLANT
IMPLANT BORON DOSE=1E15 ENERGY=10
INTERMED NAME=BIMP1 FIXNOW
IMPLANT BORON DOSE=2E14 ENERGY=20
INTERMED NAME=BIMP2 FIXNOW
DIFFUSE TEMP=900 TIME=20

$ BIMP1 : 1ST B AS-IMPLANT PROFILE
$ BIMP2 : 2ND B AS-IMPLANT PROFILE
SELECT Z=LOG10(BIMP1)
PLOT.1D X.V=0.0 TOP=22 BOT=15 RIGHT=0.5 COLOR=2 LINE=2
SELECT Z=LOG10(BIMP2)
PLOT.1D X.V=0.0 !CL !AX COLOR=3 LINE=2
SELECT Z=LOG10(BIMP1+BIMP2)
PLOT.1D X.V=0.0 !CL !AX COLOR=4 LINE=2
SELECT Z=LOG10(BORON)
PLOT.1D X.V=0.0 !CL !AX
```

Epitaxial Growth

The **EPITAXY** statement (see Chapter 4, EPITAXY, p. 4-137) is used to model the epitaxial growth of silicon layers. The duration of the epitaxy step (in minutes) is specified with the **TIME** parameter. The initial temperature of the step (in °C) is given by the **TEMPERAT** parameter. Linear variation of the temperature over the step can be specified with the **T.RATE** or **T.FINAL** parameters. If **T.RATE** is specified, the temperature varies as

$$T_c = \mathbf{TEMPERAT} + \mathbf{T.RATE} \times t \qquad \text{Equation 3-550}$$

where:

- t is the time since the start of the step.
- T_c is the diffusion temperature (in $^\circ$C) at time t.

If **T.FINAL** is specified, the temperature varies as

$$T_c = \mathtt{TEMPERAT} + \frac{(\mathtt{T.FINAL} - \mathtt{TEMPERAT})}{\mathtt{TIME}} t \qquad \text{Equation 3-551}$$

If neither **T.RATE** nor **T.FINAL** is specified, the temperature is constant. The physical coefficients that depend on temperature are presumed to be valid in the range 800° to 1250° C, but temperatures outside this range may be specified.

Layer Thickness

The thickness of the epitaxial layer must be specified with the **THICKNES** parameter. The grid distribution within the epitaxial layer can be controlled with the **DY**, **YDY**, and **SPACES** parameters, as described in Changes to the Mesh During Processing, p. 3-7.

Incorporation of Impurities

One or more impurities may be incorporated into the growing layer by using the parameters **ANTIMONY**, **ARSENIC**, **BORON**, and **PHOSPHOR** to indicate the concentrations of impurities present in the ambient gas. The concentrations of multiple impurities can be specified with these parameters. Impurities can also be specified using the **IMPURITY** and **I.CONC** parameters. For a single impurity, the resistivity may be specified in place of the impurity concentration.

Diffusion of Impurities

The diffusion equations are solved for all the mobile species present in the structure during the epitaxial growth step (see Diffusion of Impurities, p. 3-18 and Dose Loss Model, p. 3-58).The epitaxial growth is divided into diffusion steps proportional in length to the thicknesses of the grid spacings in the deposited layer. An epitaxial step that specifies five grid spaces in the deposited layer is simulated by five diffusion steps. To skip solving the diffusion equations, turn off the boolean parameter **DIFFUSE** in **EPITAXY** command.

Selective Epitaxy

By default, (single-crystal) silicon is grown on (single-crystal) silicon surfaces and polysilicon is grown on all other surface materials. If **SELECTIV** is set true, growth occurs only on silicon and polysilicon surfaces and no growth occurs on other materials. If **BLANKET** is set true, then all grown material will be single-crystal silicon, regardless of the underlying material. Setting **BLANKET** true and leaving **SELECTIV** false causes a uniform layer of single-crystal silicon to be grown over the entire surface, as in older versions of the program.

Deposition

TSUPREM-4 models the deposition of material layers at temperatures low enough that impurity diffusion can be ignored during the deposition. The **`DEPOSITION`** statement specifies the deposition of a material on the exposed surfaces of the existing structure (see Chapter 4, DEPOSITION, p. 4-99). The material to be deposited is specified by one of the seven logical parameters for identifying a material:
`SILICON`, **`OXIDE`**, **`OXYNITRI`**, **`NITRIDE`**, **`POLYSILI`**, **`ALUMINUM`**, or **`PHOTORES`**; or by specifying the name of the material with the **`MATERIAL`** statement.

Note:

The implementation of the deposition capability assumes that the top surface is exposed, and allows the right side of the structure to either be exposed or on a reflecting boundary. Deposition should not be attempted when the left or bottom sides of the structure are exposed, or when the top surface is not exposed.

Layer Thickness

The thickness of the deposited layer must be specified with the **`THICKNES`** parameter. The deposition is conformal, i.e., all points within a distance of **`THICKNES`** of the exposed surface are included in the new layer. The generation of the mesh in the new layer and the use of the **`DY`**, **`YDY`**, **`SPACES`** and **`ARC.SPAC`** parameters are described in Changes to the Mesh During Processing, p. 3-7.

Anisotropy

Anisotropy of deposition is defined as a ratio of the layer thickness on top of the flat surface to the thickness of the layer deposited on a vertical wall. Anisotropy of 1 means isotropic deposition. Anisotropy of more than one means thinner layer on the sloped surface compared to the flat surface.

Anisotropy can be any positive non-zero number. If it approaches zero, then you get zero layer thickness on the vertical walls. If it approaches infinity, then you are getting planarizing deposition filling everything up to the topmost point on the structure plus the nominal layer thickness.

Anisotropy can be specified by parameter **`ANISOTRO`** on the **`DEPOSIT`** statement. Its default value is 1, resulting in isotropic deposition.

Incorporation of Impurities

The deposited layer can be doped with one or more impurities. The impurity concentrations are specified with the **`ANTIMONY`**, **`ARSENIC`**, **`BORON`**, and **`PHOSPHOR`** parameters. Impurities also can be specified using the **`IMPURITY`** and **`I.CONC`** parameters. For a single impurity, the resistivity may be specified in place of the impurity concentration. If single crystal silicon is deposited, its orientation the same as the substrate orientation specified by the **`INITIALIZE`** statement.

Photoresist Type

The type of deposited photoresist can be specified as either **`POSITIVE`** or **`NEGATIVE`**. All photoresist in the structure is assumed to be of this type. This type of resist is used by the **`DEVELOP`** statement to determine whether exposed or unexposed resist should be removed (see Masking, Exposure, and Development of Photoresist, p. 3-156).

Polycrystalline Materials

Deposition of polycrystalline materials uses the models for initial grain size and orientation described in Modeling Polycrystalline Materials, p. 3-161. These models depend on the **`TEMPERAT`** and **`GSZ.LIN`** parameters on the **`DEPOSITION`** statement.

Deposition with Taurus Topography

TSUPREM-4 can call Taurus Topography to simulate deposition steps using detailed physical models for processes such as physical vapor deposition (PVD), chemical vapor deposition (CVD), plasma-enhanced CVD, high-density plasma deposition, atmospheric pressure CVD, spin-on glass (SOG), and reflow. Deposition using Taurus Topography is specified with the **`TOPOGRAP`** parameter on the **`DEPOSITION`** statement (see Chapter 4, DEPOSITION, p. 4-99).

Masking, Exposure, and Development of Photoresist

Masking, exposure, and development of photoresist are used to transfer an image on a mask to a structure on a semiconductor wafer. Masking information is read from a mask file created by Taurus Layout—IC Layout Interface. For each mask level, the starting and ending coordinates of each opaque region are recorded. The **`EXPOSE`** statement uses these x coordinates to determine which portions of the photoresist in a structure should be marked as exposed (in the photographic sense). The **`DEVELOP`** statement removes all positive photoresist that has been marked as exposed, or all negative photoresist that has not been marked as exposed.

TSUPREM-4 uses idealized exposure and development models: photoresist lines always have vertical sidewalls, positioned directly beneath mask edges. If accurate physical models of photolithographic processes are needed, a simulator such as Taurus-Lithography should be used.

Etching

TSUPREM-4 allows the removal of material layers or portions of layers using the **ETCH** statement (see Chapter 4, ETCH, p. 4-109). The material to be removed is specified by one of the seven logical parameters for identifying a material: **SILICON**, **OXIDE**, **OXYNITRI**, **NITRIDE**, **POLYSILI**, **ALUMINUM**, or **PHOTORES**; or by specifying the name of the material with the **MATERIAL** parameter. If a material is specified, only regions of that material are subject to removal; if no material is specified, the entire structure is subject to removal.

TSUPREM-4 can also call Taurus Topography to simulate etch steps using detailed physical models.

Defining the Etch Region

The region to be removed (the *etch region*) can be defined in one of six ways:

- The **TRAPEZOI** parameter specifies that removal occurs where etchable material appears at the exposed surface of the structure. The shape of the region is specified with the **THICKNES**, **ANGLE**, and **UNDERCUT** parameters. In a planar substrate, the region to be removed is trapezoidal in shape. Trapezoidal Etch Model, p. 3-158 describes the **TRAPEZOI** model.
- The **ISOTROPI** parameter specifies that removal occurs in a region within the given **THICKNES** of the exposed surface.
- The **LEFT** or **RIGHT** parameter specifies that removal occurs to the left or right of a line defined by the coordinates **P1.X**, **P1.Y**, **P2.X**, and **P2.Y**.
- The **START**, **CONTINUE**, and **DONE** parameters are used to specify an arbitrary region to be removed. A series of **ETCH** statements are given, each one using the **X** and **Y** parameters to specify the location of one point on a polygon that defines the etch boundary. The first **ETCH** statement in the series uses the **START** parameter, the last uses the **DONE** parameter, and the statements in between use the **CONTINUE** parameter. The polygon defining the region to be removed is closed automatically by connecting the last point to the first. A material specification can appear on any of the **ETCH** statements; if more than one of the **ETCH** statements contains a material specification, all but the last are ignored.
- The **ALL** parameter specifies that the entire structure is subject to removal. It is only useful with a material specification.
- The **OLD.DRY** parameter specifies that the region to be removed consists of all points within a vertical distance **THICKNES** of the top surface. This model

(called the **DRY** model in TSUPREM-4 versions 5.0 and older) has been superseded by the **TRAPEZOI** model.

If no region is specified, **TRAPEZOI** is assumed.

Removal of Material

Etching proceeds from exposed surfaces through material of the specified type. Thus a point within the structure is removed if it meets all of the following conditions:

- The material at the point is of the specified type (or no material is specified).
- The point lies within the etch region.
- There is a continuous path, through the specified material and within the etch region, from the point to an exposed surface.

Note:

It is not possible to etch holes in a structure, but it is possible to cut a structure into two or more pieces with the ETCH statement. All pieces except the one with the largest area are discarded. A warning is printed for each detached piece of the structure that is removed.

Trapezoidal Etch Model

The **TRAPEZOI** etch model provides a simple but flexible approximation to a number of real etching processes. The location of the etch is determined by masking layers (i.e., layers of nonetchable material), and thus does not require mask-dependent coordinates to be specified. This means the **TRAPEZOI** model can be used with the photoresist masking, exposure, and development capabilities of TSUPREM-4 to create process descriptions that are independent of any particular mask layout.

Parameters

The **TRAPEZOI** model uses the three parameters **THICKNES**, **ANGLE**, and **UNDERCUT** to specify the shape of the region to be removed. **THICKNES** specifies the vertical depth (in microns), **ANGLE** specifies the angle (in degrees) of the resulting sidewalls, and **UNDERCUT** specifies the horizontal penetration (in microns) of the etch under the edges of the masking layer.

These parameters can be used to approximate a number of real etching processes, including combinations of vertical and isotropic etches, V-groove etches, and etches that produce retrograde sidewall profiles.

Etch Steps

An etch with the **TRAPEZOI** model is done in three steps:

1. A vertical etch to depth **THICKNES** is performed. This etch does not apply to portions of the surface that are masked by nonetchable materials or shadowed

by etchable or nonetchable materials, nor is it used on segments of the surface that form an angle greater than **ANGLE** to the horizontal.

2. A horizontal etch is performed. Surfaces that were exposed at the start of Step 1 are etched horizontally by the distance **UNDERCUT**. Surfaces that were exposed during Step 1 are etched by a distance proportional to the length of time between when they first became exposed and the end of Step 1. Thus, a sidewall exposed 3/4 of the way into Step 1 is etched horizontally by 1/4 of **UNDERCUT**. (An exception is made when an **ANGLE** greater than 90° is specified; this case is described below.)
3. Where overhangs of etchable material are present at the end of Step 2, a vertical *upwards* etch (i.e., in the $-y$ direction) is performed. On surfaces that were exposed at the start of Step 2, this etch is to a distance **UNDERCUT**; on surfaces that were first exposed during the course of Step 2, the distance of this etch is reduced in proportion to the time from the start of Step 2. This step approximates the undercutting of the mask due to the isotropic component of the etch.

Etch Examples

Figure 3-2 shows a number of examples of the **TRAPEZOI** etch model. In each case, an etchable layer (light and medium gray) is partially masked by a nonetchable layer (dark gray).

Simple Structure

When the **THICKNES**, **ANGLE**, and **UNDERCUT** parameters satisfy the relationship

$$\frac{\mathbf{THICKNES}}{\mathbf{UNDERCUT}} = \tan(\mathbf{ANGLE}) \qquad \text{Equation 3-552}$$

the etch approximates a vertical etch with an isotropic component. This is the case whenever two or fewer of the parameters **THICKNES**, **ANGLE**, and **UNDERCUT** are specified. Figure 3-2 (a) and (b) show the effect of this type of etch on various starting structures.

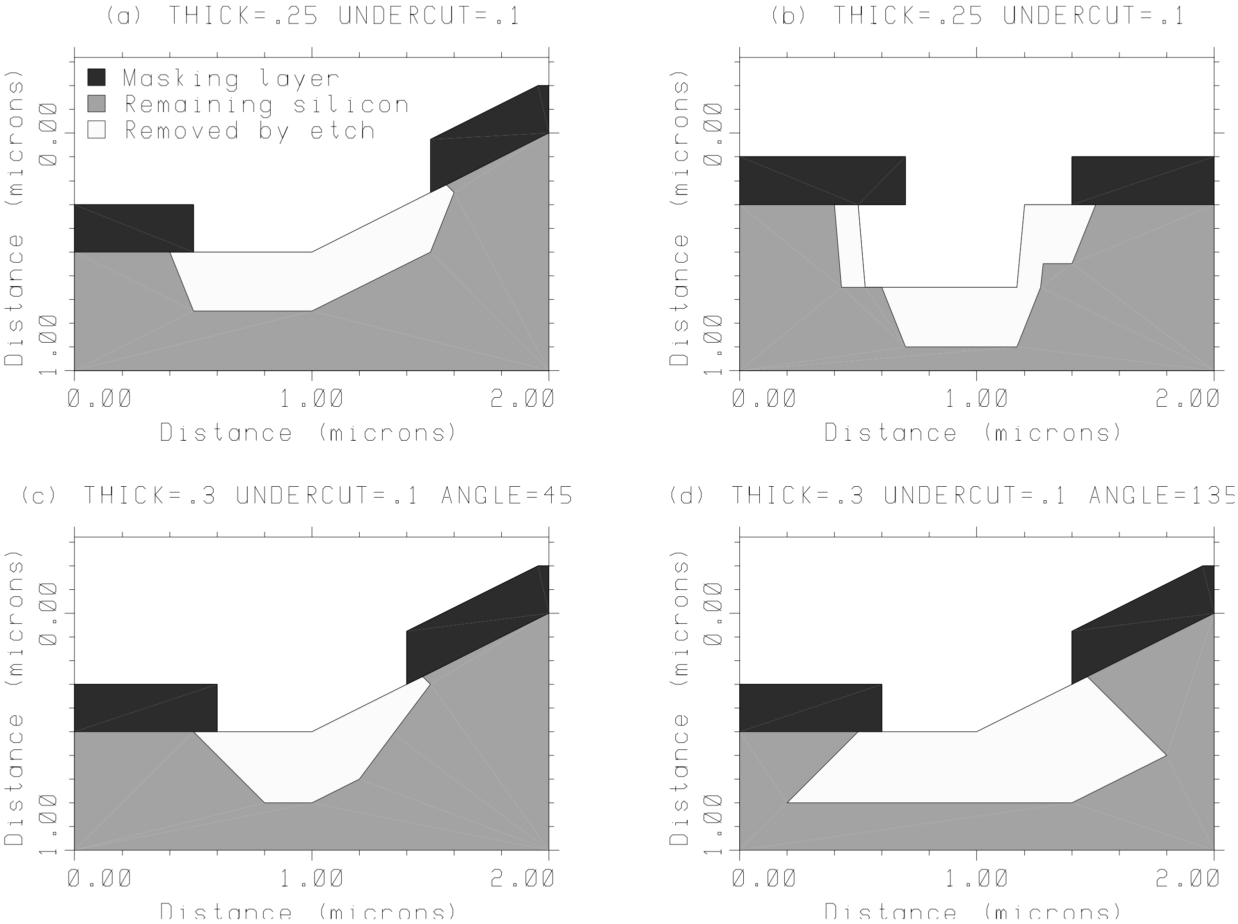


Figure 3-2 Examples of the trapezoidal etch model

The left half of Figure 3-2 (a) shows the result when etching a planar substrate: the etch region is a trapezoid of depth **THICKNES**, extending a distance **UNDERCUT** beneath the mask edge, and with a sidewall slope of **ANGLE** degrees. The right half of Figure 3-2 (a) shows the result when etching a nonplanar surface:

Step 1 of the etch sequence etches the exposed surface vertically to a depth of **THICKNES** microns. Step 2 etches the resulting sidewall in the horizontal direction, producing an undercutting of the mask and the sloped sidewall. In this case Step 3 also has an effect, etching upwards from the undercut region. Thus, the *hook* in the final silicon profile is the result of approximating the isotropic component of the etch. In every case, the intersection between the bottom of the etch region and the sidewall occurs directly under the edge of the mask.

Structure with Overhangs

Figure 3-2 (b) shows the result of etching a more complicated structure. The left side of the figure shows the effects of overhangs. The vertical etch (Step 1) is suppressed where the silicon is shadowed by the masking layer. The horizontal etch (Step 2) applies to the entire structure, including the near-vertical sidewalls sheltered under the overhang. The right half of the figure shows what happens to exposed near-vertical sidewalls: Again the horizontal surfaces are etched vertically in Step 1, then both the sidewalls created by the vertical etch and the original near-vertical sidewalls are etched horizontally in Step 2. The effect is that material is removed if it can be etched by *either* the vertical (anisotropic) component or the horizontal (isotropic) component of the etch.

Complex Structures

Figure 3-2 (c) and (d) show what happens when Equation 3-552 is not satisfied. In Figure 3-2 (c) you have **THICKNESS/UNDERCUT** < tan(**ANGLE**). In this case the sloped sidewall of the etch extends out under the opening in the mask. The intersection between the bottom of the etch region and the sidewall is no longer directly beneath the edge of the mask. If the mask opening is narrow enough, the bottom of the etch region disappears entirely, resulting in a V-groove etch. To produce this etch shape, Step 1 of the etch process is modified to reduce the depth of the vertical etch near the edges of the mask opening. *Note* that in this situation, even the tiniest speck of nonetchable material can produce a triangular mound of unetched material in the final structure.

Figure 3-2 (d) shows the case where **ANGLE** > 90. In this case, the bottom of the etched region is wider than the opening in the masking layer, producing overhanging sidewalls. This etch is accomplished by modifying Step 2 of the etch procedure to etch further horizontally at the bottom of the sidewalls formed by Step 1 than at the top. The apparent etch depth of 0.5 microns at the right side of the mask opening is the result of a 0.3 micron vertical etch of the original sloped surface (Step 1) followed by a 0.4 micron horizontal etch of the sloped “bottom wall” that results from Step 1.

Etching with Taurus Topography

TSUPREM-4 can call Taurus Topography to simulate etch steps using detailed physical models for processes such as wet etching, dry etching (RIE), ion milling, and chemical mechanical polishing (CMP). Etching using Taurus Topography is specified with the **TOPOGRAP** parameter on the **ETCH** statement (see Chapter 4, ETCH, p. 4-109).

Modeling Polycrystalline Materials

The behavior of dopants in polycrystalline materials is strongly influenced by the boundaries between crystalline grains. Dopant atoms tend to segregate from the interior of a grain to the boundaries, which provide paths for rapid diffusion. The rate of segregation depends on the rate of grain growth, while the number of diffusion paths along the boundaries depends on the grain size. In addition, the boundaries of the polycrystalline material act like grain boundaries, providing sites for electrically inactive dopant atoms and paths for diffusion.

The diffusion equations for the grain-interior and grain-boundary components of the doping profile are solved separately. The equations are coupled by terms describing the segregation between the grain interiors and grain boundaries. To determine the rate of segregation and the density of grain-boundary diffusion paths, you also solve for the growth in grain size during high-temperature processing. The boundaries of the polycrystalline region are included as explicit grain boundaries in the diffusion and segregation equations.

The poly model has been implemented in TSUPREM-4 in collaboration with GEC Plessey Semiconductors (GPS) in the UK. The physical model has been developed at GPS and GEC-Marconi Materials Technology, Caswell, UK within the collaborative European ESPRIT project STORM. The key contributors to this work were S. K. Jones, C. Hill, and A. G. O'Neill References [82] to [85]. Although the model was developed and optimized for polysilicon, it can be applied to other polycrystalline materials (e.g., silicides).

Diffusion

Redistribution of dopants in polycrystalline materials occurs by the parallel diffusion of dopants through the interiors of grains and along grain boundaries.

Diffusion in Grain Interiors

In the grain interiors diffusion of the active dopant is given by

$$\frac{\partial c_g}{\partial t} = -\vec{\nabla} \cdot \left[-D_g \left(\vec{\nabla} c_g - z_s c_g \frac{q\vec{E}_g}{kT} \right) \right] - G \qquad \text{Equation 3-553}$$

where c_g is the active concentration in the grain interior and the other symbols are described in Diffusion of Impurities, p. 3-18. The diffusivity D_g and electric field $\vec{E}_g$ in the grain interior are calculated from the electron concentration n_g, which is in turn calculated from the doping concentrations c_g. G accounts for the segregation of dopant to grain boundaries, as described in Segregation Between Grain Interior and Boundaries, p. 3-164. (The calculation of the active concentration is described in Activation of Impurities, p. 3-34, and the calculation of the electron concentration is described in Diffusion of Impurities, p. 3-18.)

Grain Boundary Structure

Diffusion along grain boundaries is described in terms of the dopant concentration *per unit area* of grain boundary c_{gb} and the average area of grain boundaries per unit volume:

$$\rho' = \rho + \delta_{if} \qquad \text{Equation 3-554}$$

where ρ is the average area of grain boundaries per unit volume in the bulk of the poly layer and δ_{if} accounts for the dopant at interfaces between poly and other materials (or ambient). ρ is inversely proportional to the average grain size L_g

$$\rho = \frac{\texttt{G.DENS}}{L_g} \qquad \text{Equation 3-555}$$

where **`G.DENS`** is a geometrical factor specified on the **`MATERIAL`** statement. δ_{if} is a function of position defined by the fact that its integral over any area is equal to the length L_{if} of the polysilicon interface passing through that area:

$$\int \delta_{if} dA = L_{if}$$ Equation 3-556

The concentration of dopants in the grain boundaries per unit volume of material is then given by:

$$w_{gb} = \rho' c_{gb}$$ Equation 3-557

Diffusion Along Grain Boundaries

The diffusion of dopant in the grain boundaries is given by:

$$\frac{\partial w_{gb}}{\partial t} = -\vec{\nabla} \cdot \left[-F D_{gb} \left(\vec{\nabla} c_{gb} - z_s c_{gb} \frac{q \vec{E}_{gb}}{kT} \right) \right] + G$$ Equation 3-558

Diffusivity D_{gb} and electric field $\vec{E}_{gb}$ along the grain boundaries are calculated from the electron concentration n_{gb}; n_{gb} is calculated as in Equation 3-42 except that the net donor and acceptor concentrations are calculated from c_{gb}/K, the equilibrium dopant concentrations in the grain interior near the grain boundary.

Anisotropic Diffusion

F is a tensor that describes the diffusion paths available to dopant in the grain boundaries. It is composed of two parts: $F = F_b + (1 - F_{bu}) F_{if}$. F_b describes the available paths within the bulk of the poly layer. For a horizontal poly layer, it is given by:

$$F_b = \text{diag}\left(\frac{\mathtt{F11}}{L_g}, \frac{\mathtt{F22}}{L_g} \right)$$ Equation 3-559

where **`F11`** and **`F22`** are parameters on the **`MATERIAL`** statement. Because of the columnar grain structure, **`F22`** is larger than **`F11`**, which implies that diffusion through the layer is faster than diffusion parallel to the layer. F_{if} describes the available paths for diffusion along material interfaces. In the vicinity of a horizontal interface it has the value:

$$F_{if} = \text{diag}(\delta_{if}, 0)$$ Equation 3-560

For the interface between polysilicon and silicon, the phenomenon of interfacial break-up accompanied by epitaxial realignment can occur, as described in Interface Oxide Break-up and Epitaxial Regrowth, p. 3-167. F_{bu} is the fraction of the polysilicon/silicon interface that has broken up. For layers or interfaces that are not horizontal, F_b and F_{if} are rotated by the angle of the layer or interface, respectively, with respect to the x axis.

Segregation Between Grain Interior and Boundaries

When dopant is initially introduced into a polycrystalline material, some of the dopant occupies sites in the interior of a grain and some occupies sites on a grain boundary. The initial segregation of dopant is given by:

$$c_{gb} = \frac{\texttt{Q.SITES}}{\texttt{CG.MAX}} \texttt{GSEG.INI}\ c_g \qquad \text{Equation 3-561}$$

`Q.SITES`, **`CG.MAX`**, and **`GSEG.INI`** are parameters on the **`IMPURITY`** statement; they represent the density of available sites on grain boundaries and in the grain interiors and the initial segregation entropy, respectively. In the case of ion implantation c_g and c_{gb} describe the additional dopant introduced by the implantation; dopant that is present before the implantation is not redistributed.

Dopant atoms are free to move between sites in the interior of a grain and sites on the grain boundary during high-temperature processing. The rate of segregation is given by:

$$G = (\rho q_b + (1 - F_{bu})\delta_{if} q_{if})\left(f_{gb} c_g - f_g \frac{c_{gb}}{K}\right) \qquad \text{Equation 3-562}$$

The segregation coefficient K is given by Reference [86]:

$$K = \frac{\texttt{Q.SITES}}{\texttt{CG.MAX}} \texttt{GSEG.0}\ \exp\left(\frac{\texttt{GSEG.E}}{kT}\right) \qquad \text{Equation 3-563}$$

where **`GSEG.0`** and **`GSEG.E`** are parameters on the **`IMPURITY`** statement. The segregation velocities associated with the bulk of the poly region and the material interfaces are given by:

$$q_b = \frac{1}{\texttt{ALPHA}} \frac{\partial L_g}{\partial t} + \texttt{FVELB.0} \cdot \exp\left(-\frac{\texttt{FVELB.E}}{kT}\right) \frac{D_g}{L_g} \qquad \text{Equation 3-564}$$

$$q_{if} = \texttt{VELIF.0}\ \exp\left(-\frac{\texttt{VELIF.E}}{kT}\right) \qquad \text{Equation 3-565}$$

where **`ALPHA`** is a parameter on the **`MATERIAL`** statement, **`FVELB.0`**, **`FVELB.E`**, **`VELIF.0`** and **`VELIF.E`** are parameters on the **`IMPURITY`** statement. f_g and f_{gb} are the fractions of unfilled interior and boundary sites:

$$f_g = 1 - \frac{c_g}{\texttt{CG.MAX}} \qquad \text{Equation 3-566}$$

$$f_{gb} = 1 - \sum \frac{c_{gb}}{\texttt{Q.SITES}} \qquad \text{Equation 3-567}$$

where the sum is taken over all the dopant species present in the structure. F_{bu} is the fraction of the polysilicon/silicon interface that has broken up, as described in Interface Oxide Break-up and Epitaxial Regrowth, p. 3-167.

When calculating the segregation between poly and another material, the total (active plus inactive) concentration in the grain interior is used for the concentration in poly in Equation 3-244.

Grain Size Model

The grains in the polycrystalline material are assumed to be oriented as columns that extend through the wafer. The structure is characterized by L_g, the average grain size in the lateral direction (i.e., in the plane of the layer), and $\vec{\xi}$, a vector describing the orientation of the columnar grains. The grain size can be examined by specifying the *lgrain* solution variable on the **SELECT** statement:

```
SELECT Z=LGRAIN
```

This returns the average grain size in microns or a negative value at nodes where the material is amorphous. (*Note* that the grain size is reported in centimeters by Michelangelo and Taurus-Visual.)

The initial grain size is determined by the temperature of the poly deposition process and the value of the **GSZ.LIN** parameter on the **DEPOSITION** statement. If **GSZ.LIN** is false, the grain size in the layer is constant [87]:

$$L_g = \begin{cases} \max(\mathbf{FRAC.TA} \times t_a,\ \mathbf{MIN.GRAI}) & T_c \le \mathbf{TEMP.BRE} \\ \mathbf{GRASZ.0}\ \exp\left(\dfrac{-\mathbf{GRASZ.E}}{kT}\right) \mathbf{THICKNES} & T_c > \mathbf{TEMP.BRE} \end{cases}$$

Equation 3-568

where **FRAC.TA**, **MIN.GRAI**, **GRASZ.0**, **GRASZ.E**, and **TEMP.BRE** are parameters on the **MATERIAL** statement, T_c is the deposition temperature (specified on the **DEPOSITION** statement) in °C, and T is the deposition temperature in Kelvins. **THICKNES** is the thickness of the deposited layer, as specified on the **DEPOSITION** statement. t_a is the thickness of the amorphous silicon layer produced by low-temperature deposition. If **GSZ.LIN** is true, the grain size increases linearly from the bottom of the layer to the top:

$$L_g = \begin{cases} \max(\mathbf{FRAC.TA} \times t_a,\ \mathbf{MIN.GRAI}) & T_c \le \mathbf{TEMP.BRE} \\ \mathbf{MIN.GRAI} + 2\mathbf{GRASZ.0}\ \exp\left(\dfrac{-\mathbf{GRASZ.E}}{kT}\right) z & T_c > \mathbf{TEMP.BRE} \end{cases}$$

Equation 3-569

where z is the distance from the bottom of the layer.

For high-temperature depositions, grain size depends on the thickness specified on the **DEPOSITION** statement. Dividing a deposition into multiple smaller depositions produces different results for the grain size. For low-temperature depositions, the material is assumed to be amorphous (a negative grain size is reported in printing or plotting). The initial grain size (above) is calculated from the actual thickness t_a of the amorphous layer at the beginning of the next diffusion step. Thus, successive amorphous depositions are merged in computing the grain size.

Grain Growth

The growth of the grains during high-temperature processing is given by Reference [87]:

$$\frac{\partial L_g}{\partial t} = \frac{1}{L_g}\mathbf{GEOM} \times \mathbf{GAMMA.0}\ \exp\left(\frac{-\ \mathbf{GAMMA.E}}{kT}\right) \times D_{Si} \times \frac{E_{gb}}{kT} \times F_{seg} \times G_{EA}$$

Equation 3-570

where **GEOM**, **GAMMA.0**, and **GAMMA.E** are parameters on the **MATERIAL** statement, D_{Si} is the silicon self-diffusivity in the vicinity of a grain boundary, E_{gb} is the surface energy per atom associated with the grain boundary, F_{seg} models the segregation drag effect, and G_{EA} models epitaxial regrowth of the poly layer. (See Interface Oxide Break-up and Epitaxial Regrowth, p. 3-167.)

Concentration Dependence

The silicon self-diffusivity is given by:

$$D_{Si} = D^{x}_{Si} + D^{+}_{Si}\left(\frac{n_i}{n_g}\right) + D^{-}_{Si}\left(\frac{n_g}{n_i}\right) + D^{=}_{Si}\left(\frac{n_g}{n_i}\right)^2$$

Equation 3-571

where n_i is the intrinsic carrier concentration at the processing temperature and n_g is the electron concentration in the grain interior. The components of D_{Si} are given by:

$$D^{x}_{Si} = \mathbf{DSIX.0}\ \exp\left(-\frac{\mathbf{DSIX.E}}{kT}\right)$$

Equation 3-572

$$D^{+}_{Si} = \mathbf{DSIP.0}\ \exp\left(-\frac{\mathbf{DSIP.E}}{kT}\right)$$

Equation 3-573

$$D^{-}_{Si} = \mathbf{DSIM.0}\ \exp\left(-\frac{\mathbf{DSIM.E}}{kT}\right)$$

Equation 3-574

$$D^{=}_{Si} = \mathbf{DSIMM.0}\ \exp\left(-\frac{\mathbf{DSIMM.E}}{kT}\right)$$

Equation 3-575

where **DSIX.0**, **DSIX.E**, **DSIP.0**, **DSIP.E**, **DSIM.0**, **DSIM.E**, **DSIMM.0**, and **DSIMM.E** are parameters on the **MATERIAL** statement.

Grain Surface Energy

The grain boundary energy is given by References [87] to [89].

$$E_{gb} = \frac{\mathbf{GBE.0}}{1 + \mathbf{GBE.H} f_n} + \mathbf{GBE.1}\frac{L_g}{t_{poly}} \qquad \text{Equation 3-576}$$

$$f_n = \begin{cases} \dfrac{L_g}{2t_{poly} - L_g} & L_g < t_{poly} \\ \dfrac{L_g}{t_{poly}} & L_g \geq t_{poly} \end{cases} \qquad \text{Equation 3-577}$$

where **`GBE.0`**, **`GBE.H`**, and **`GBE.1`** are parameters on the **`MATERIAL`** statement, and t_{poly} is the thickness of the polycrystalline layer. *Note* that **`GBE.0`** and **`GBE.1`** are in units of electron volts per atom; to convert from values given in eV/µm^2 you must multiply by 6.25 × 10^{-8} µm^2/atom.

Segregation Drag

The segregation drag effect reduces the grain growth rate Reference [90]:

$$F_{seg} = \left(1 + \sum \frac{c_{gb}}{\mathbf{Q.SITES}}\right)^{-\mathbf{NSEG}} \qquad \text{Equation 3-578}$$

`Q.SITES` is a parameter on the **`IMPURITY`** statement, **`NSEG`** is a parameter on the **`MATERIAL`** statement, and the summation is taken over the impurities present in the structure.

Interface Oxide Break-up and Epitaxial Regrowth

A thin interfacial oxide layer is typically present between a deposited polysilicon layer and any underlying single-crystal silicon. This interfacial oxide presents a barrier to epitaxial realignment of the poly layer. With sufficient high-temperature processing, the oxide layer breaks up into a discrete set of small spheres, allowing epitaxial regrowth of the poly to proceed.

Oxide Break-Up

The oxide break-up is modeled by the formation of voids in the interfacial oxide layer References [91] to [94]. The radius of the voids R_{void} increases as:

$$\frac{dR_{void}}{dt} = \frac{\beta}{t_{ox}^3} \times \exp\left(-\frac{E_{bu}}{kT}\right) \qquad \text{Equation 3-579}$$

where β is a constant, t_{ox} is the initial oxide thickness, and E_{bu} is the activation energy of the break-up process. R_{void} is initialized to zero whenever poly is deposited on exposed silicon. The fraction of the interface that is broken up is given by:

$$F_{bu} = 1 - \exp(-\pi N_{EA} R_{void}^2) \qquad \text{Equation 3-580}$$

where N_{EA} is the areal density of the voids. The parameters for the model are specified in terms of a characteristic break-up time for the thinnest (5Å) interfacial oxide layers:

$$t_{bu} \equiv \frac{5\dot{A}^3}{\sqrt{\pi N_{EA}}\beta} \times \exp\frac{E_{bu}}{kT} = \mathtt{TBU.0} \cdot \exp\left(-\frac{\mathtt{TBU.E}}{kT}\right) \qquad \text{Equation 3-581}$$

where **TBU.0** and **TBU.E** are parameters on the **MATERIAL** statement. Equation 3-579 can be then written:

$$\frac{dR_{void}}{dt} = \frac{1}{t_{bu}}\left(\frac{5\dot{A}}{t_{ox}}\right)^3 \frac{1}{\sqrt{\pi N_{EA}}} \qquad \text{Equation 3-582}$$

In the present implementation, you assume that all poly/single-crystal interfaces share a common oxide thickness given by:

$$t_{ox} = \mathtt{TOXIDE} \qquad \text{Equation 3-583}$$

where **TOXIDE** is a parameter on the **MATERIAL** statement.

Epitaxial Regrowth

Epitaxial regrowth is modeled by increasing the poly grain size to a value much larger than the thickness of the poly layer. This grain growth is described by G_{EA} in Equation 3-560 for the grain size:

$$G_{EA} = D_{L_g}\nabla^2 L_g + F_{bu} v_{EA} \delta_{if} \qquad \text{Equation 3-584}$$

The first term models the propagation of the regrowth through the poly layer, while the second term (which is nonzero only at the silicon/polysilicon interface) serves as a driving force for epitaxial regrowth from the interface. Parameters for this model are given by:

$$D_{L_g} = \mathtt{DLGX.0} \cdot \exp\left(-\frac{\mathtt{DLGX.E}}{kT}\right) \qquad \text{Equation 3-585}$$

$$v_{EA} = \mathtt{EAVEL.0} \cdot \exp\left(-\frac{\mathtt{EAVEL.E}}{kT}\right) \qquad \text{Equation 3-586}$$

where **DLGX.0**, **DLGX.E**, **EAVEL.0**, and **EAVEL.E** are parameters on the **MATERIAL** statement.

Dependence of Polysilicon Oxidation Rate on Grain Size

It has been observed experimentally that the oxidation rate for fine-grained polysilicon is faster than for coarser-grained poly or single-crystal silicon, presumably because of enhanced oxidation at the grain boundaries.

This enhancement can be modelled by assuming a faster surface reaction rate where grain boundaries intersect the oxide/poly interface:

$$k_s = (1-f)k_g + fk_{gb} \qquad \text{Equation 3-587}$$

where:

$$k_g = \frac{B}{A}\frac{N_1}{C^*} \qquad \text{Equation 3-588}$$

is the surface reaction rate in the absence of grain boundaries, k_{gb} is the reaction rate at a grain boundary, and:

$$f = \min\left(\frac{\delta T}{L_g},\ 1\right) \qquad \text{Equation 3-589}$$

is the fraction of the surface within a distance $\delta T/\ 2$ of a grain boundary. The enhancement factor at grain boundaries is specified as:

$$\frac{k_{gb}}{k_g} = \mathbf{GB.FAC.0} \cdot \ \exp\left(\frac{-\mathbf{GB.FAC.E}}{kT}\right) \qquad \text{Equation 3-590}$$

where **GB.FAC.0** and **GB.FAC.E** are given on the **AMBIENT** statement for each oxidizing species and k_g is calculated from the linear rate constant for the bulk material. The effective thickness of the grain boundaries is given by $\delta T = \mathbf{GB.THICK}$, where **GB.THICK** is specified for each material on the **MATERIAL** statement. The default values are $\mathbf{GB.FAC.0} = 10$ and $\mathbf{GB.FAC.E} = 0$ for all ambients and $\mathbf{GB.THICK} = 0.0005$ (0.5 nm) for polysilicon. This gives an enhancement of less than 10% for a grain size of 50 nm.

Using the Polycrystalline Model

For each dopant in a polycrystalline material there are two solution values: one for the total doping concentration at each node in the structure and one for the concentration in grain boundaries. The total concentration is accessed by the name of the impurity, e.g.

```
SELECT Z=LOG10(BORON)
```

The grain-boundary component is accessed with the *gb()* function on the **SELECT** statement:

```
SELECT Z=LOG10(GB(BORON))
```

The grain-boundary component includes the dopant stored at the boundary of the polycrystalline material. Although this dopant is confined to the boundary, it is reported as if it were evenly distributed over the areas associated with the nodes on the boundary. Thus, the concentration at the boundary of the polycrystalline region is discontinuous and depends on the grid spacing at the boundary. While it is not possible to resolve the profile at the boundary, care has been taken that the total dose at and near the boundary is correct.

The grain-interior component can be computed as the difference between the total concentration and the grain-boundary concentration:

```
SELECT Z=LOG10(BORON-GB(BORON))
```

The grain-boundary component is stored as a separate impurity. The name of the grain-boundary impurity is created from the name of the base impurity by prepending **GB_** to the name. Thus, you set the diffusivities of the grain-boundary component of boron with:

```
IMPURITY IMP=GB_BORON MAT=POLY DIX.0= . . .
```

The grain size can be examined by specifying the *lgrain* solution variable on the **SELECT** statement:

```
SELECT Z=LGRAIN
```

This returns the average grain size in microns; a negative value is returned for nodes where the material is amorphous. (*Note* that the grain size is reported in centimeters by Michelangelo and Taurus Visual.)

The polycrystalline models address the need to model polycrystalline silicon, but the implementation allows them to be used for other polycrystalline materials. The polycrystalline model is enabled or disabled for a material by the **POLYCRYS** parameter on the **MATERIAL** statement. When the polycrystalline models are disabled with **^POLYCRYS**, the impurity concentrations in the grain boundaries are set to zero and the corresponding equations are not solved. Thus, the diffusivity reduces to that of the grain interiors. To compensate for the lack of grain boundary solutions, the grain interior diffusivity is multiplied by **FGB** in materials for which the polycrystalline models are disabled. **FGB** is a parameter on the **IMPURITY** statement.

Electrical Calculations

TSUPREM-4 calculates a limited set of electrical characteristics along vertical cut-lines through a simulation structure. This is accomplished by saving the 1-D Poisson's equation for specified bias conditions. The electrical information consists of the threshold voltage, the low-frequency, high-frequency, and deep-depletion MOS capacitances, spreading resistance profile and sheet resistances for all diffused regions in all semiconductor layers within the structure. This information can be saved in a file for later plotting by using the **OUT.FILE** parameter in the **ELECTRICAL** statement (see Chapter 4, ELECTRICAL, p. 4-194).

TSUPREM-4 extends the structure automatically if the bias voltage expands the depletion region beyond the bottom of a simulation structure. Also, the grid for simulation may not be adequate for solving Poisson's equation; the grid is automatically extended if necessary. The extended structure and the added grid are temporary and are used only for the electrical extraction.

Automatic Regrid

The **E.REGRID** in the **METHOD** statement specifies the automatic regrid for electrical extraction. The automatic regrid follows two steps:

1. The first step regrids the region in the estimated maximum depth of depletion region. The minimum and maximum grid spacing are determined by the incremental depletion depth calculated for given bias conditions.
2. The second step generates more dense grids near the surface. This regrid results in the accurate calculation specially for the quantum effect.

Placement of grid near the surface is controlled by the thickness of the regrid region (**E.TSURF**), the first grid spacing at the surface (**E.DSURF**) and the incremental ratio of grid spacing (**E.RSURF**) in the **METHOD** statement.

Poisson's Equation

The form of Poisson's equation solved in semiconductor and insulator regions is:

$$\frac{\partial}{\partial x}\left(\mathbf{EPSILON}\ \varepsilon_0 \frac{\partial \psi}{\partial x}\right) = \begin{cases} -q(p - n + N_d^+ - N_a^-) & \text{semiconductor} \\ 0 & \text{insulator} \end{cases}$$

Equation 3-591

where ε_0 is the dielectric constant in vacuum, ψ is the potential, p is the hole concentration, n is the electron concentration, and N_d^+ and N_a^- are the sums of the ionized electrically active donor and acceptor impurity concentrations, respectively. The **EPSILON** parameter is the relative dielectric constant specified in the **MATERIAL** statement (see Chapter 4, MATERIAL, p. 4-258). The potential in a semiconductor region is defined as the potential of the edge of the conduction band.

Boltzmann and Fermi-Dirac Statistics

Either Boltzmann or Fermi-Dirac statistics may be used to represent the electron and hole concentrations in semiconductor regions. The type of statistics used in a semiconductor material is specified with the **BOLTZMAN** parameter in the **MATERIAL** statement. If Boltzmann statistics are used, the electron and hole concentrations are given by:

$$n = \mathbf{N.CONDUC}\left(\frac{T}{300}\right)^{\mathbf{NC.F}} \exp\left[\frac{q(\psi - \phi_n)}{kT}\right]$$

Equation 3-592

$$p = \mathbf{N.VALENC}\left(\frac{T}{300}\right)^{\mathbf{NV.F}} \exp\left[\frac{q(\phi_p - \psi) - E_g}{kT}\right]$$

Equation 3-593

$$E_g = \mathbf{BANDGAP} + \mathbf{EGALPH} \cdot \left(\frac{300^2}{300 + \mathbf{EGBETA}} - \frac{T^2}{T + \mathbf{EGBETA}}\right)$$

Equation 3-594

where:

- ϕ_n is the electron quasi-Fermi potential.
- ϕ_p is the hole quasi-Fermi potential.

If Fermi-Dirac statistics are used, the electron and hole concentrations are given by:

$$n = \mathbf{N.CONDUC}\left(\frac{T}{300}\right)^{\mathbf{NC.F}} F_{1/2}\left[\frac{q(\psi - \phi_n)}{kT}\right]$$

Equation 3-595

$$p = \mathbf{N.VALENC}\left(\frac{T}{300}\right)^{\mathbf{NV.F}} F_{1/2}\left[\frac{q(\phi_p - \psi) - E_g}{kT}\right]$$

Equation 3-596

where:

- $F_{1/2}$ is the Fermi-Dirac integral of one-half order.
- The parameters **`N.CONDUC`**, **`N.VALENC`**, **`NC.F`**, **`NV.F`**, **`BANDGAP`**, **`EGALPH`**, and **`EGBETA`** are specified in the **`MATERIAL`** statement.
- The temperature *T* used for the device simulation is specified with the **`TEMPERAT`** parameter in the **`ELECTRICAL`** statement.

Ionization of Impurities

Either complete or incomplete ionization of impurities may be used to represent the ionized donor and acceptor impurity concentrations in semiconductor regions. The type of impurity ionization used in a semiconductor material is specified with the **`IONIZATI`** parameter in the **`MATERIAL`** statement. If complete ionization is used, the ionized donor and acceptor impurity concentrations are given by:

$$N_d^+ = N_d$$

Equation 3-597

$$N_a^- = N_a$$

Equation 3-598

If incomplete ionization (**`^IONIZATI`**) is used, the ionized donor and acceptor impurity concentrations are given by Reference [96]:

$$N_d^+ = \frac{N_d}{1 + \mathbf{G.DONOR}\ \exp\left[\frac{q(\psi - \phi_n) + \mathbf{E.DONOR}}{kT}\right]}$$

Equation 3-599

$$N_a^- = \frac{N_a}{1 + \texttt{G.ACCEP}^{-1}\exp\left[\dfrac{q(\phi - \psi) - \texttt{BANDGAP} + \texttt{E.ACCEP}}{kT}\right]}$$

Equation 3-600

where N_d and N_a are the sums of the electrically active donor and acceptor impurity concentrations, respectively. The parameters **`G.DONOR`**, **`E.DONOR`**, **`G.ACCEP`**, and **`E.ACCEP`** are specified in the **`MATERIAL`** statement.

The potential in a conductor region is constant with a value given by:

$$\psi = V_c - \texttt{WORKFUNC} + \chi$$

Equation 3-601

where V_c is the conductor bias and the **`WORKFUNC`** parameter is specified in the **`MATERIAL`** statement. If any semiconductor regions are present in the simulation structure, χ is the electron affinity for the bottommost semiconductor layer in the structure. The value of χ is specified with the **`AFFINITY`** parameter in the **`MATERIAL`** statement. If no semiconductor regions are present, χ is set to zero.

Solution Methods

Poisson's equation is solved numerically using a three-point finite difference approach. Newton's method is used to iteratively solve the nonlinear equation by linearizing the equation at each iteration and using Gaussian elimination to solve the resulting tridiagonal system. The iteration is normally terminated when the relative change in the potential between successive iterations is less than the allowed error at each node in the structure. The iteration may also be terminated when the maximum number of allowed Newton iterations has been performed. A minimum number of required iterations are performed before the iteration is terminated. The allowed error, maximum number of iterations, and minimum number of iterations may be specified with the **`E.RELERR`**, **`E.ITMAX`**, and **`E.ITMIN`** parameters, respectively, in the **`METHOD`** statement.

Carrier Mobility

The mobilities of electrons and holes depend on the impurity concentration, electric field, and temperature. The mobility is given by References [98] and [99]:

$$\mu_n = \texttt{GSURFN}\frac{\mu_{0n}}{\sqrt{1 + \left(\dfrac{|E|}{\texttt{ECN.MU}}\right)}}$$

Equation 3-602

$$\mu_p = \texttt{GSURFP}\frac{\mu_{0p}}{\sqrt{1 + \left(\dfrac{|E|}{\texttt{ECP.MU}}\right)}}$$

Equation 3-603

where μ_0 is the low-field mobility, E is the electric field, and **ECN.MU** and **ECP.MU** are the critical electric fields. The mobility used for calculating current along semiconductor-insulator interfaces may be reduced by specifying degradation factors **GSURFN** and **GSURFP** which multiply the low field mobility. It is important to note that the factors **GSURFN** and **GSURFP** are only applied at interfaces between semiconductor and insulator. Everywhere else, these factors should be considered to have values of unity. **ECN.MU**, **ECP.MU**, **GSURFN**, and **GSURFP** can be specified on the **MOBILITY** statement (see Chapter 4, MOBILITY, p. 4-305).

TSUPREM-4 provides three alternatives—tabular form, the Arora model, and the Caughey model—for specifying low field electron and hole mobilities dependent on impurity concentration and temperature.

Tabular Form

The low-field mobilities for the tabular form depend on the electrically active impurity concentration and device temperature. The mobilities are represented by the table of values specified with the **MOBILITY** statement. 2-D interpolation is used to obtain values of low-field mobility for impurity concentration and temperature between the values in a table. The concentration is assumed to vary exponentially and the temperature and mobility are assumed to vary linearly. TSUPREM-4 uses this method by default. The mobility using this table can be selected with the **MOB.TABL** on the **METHOD** statement. The table values may be modified using the **MOBILITY** statement.

Note:

> ***The default mobility tables are the same as those used in Medici. They differ slightly from the tables used for calculating initial impurity concentrations from resistivity.***

Arora Mobility Model

Also available is the analytic model based on work by Arora, et al., Reference [100] which takes into account total impurity concentration and temperature. The mobility expressions are:

$$\mu_{0n} = \mathbf{MUN1}\left(\frac{T}{300}\right)^{\mathbf{EXN1}} + \frac{\mathbf{MUN2}\left(\frac{T}{300}\right)^{\mathbf{EXN2}}}{1 + \left[\dfrac{N_{total}}{\mathbf{CN}\left(\frac{T}{300}\right)^{\mathbf{EXN3}}}\right]^{\alpha_n}} \qquad \text{Equation 3-604}$$

$$\mu_{0p} = \mathbf{MUP1}\left(\frac{T}{300}\right)^{\mathbf{EXP1}} + \frac{\mathbf{MUP2}\left(\frac{T}{300}\right)^{\mathbf{EXP2}}}{1 + \left[\frac{N_{total}}{\mathbf{CP}\left(\frac{T}{300}\right)^{\mathbf{EXP3}}}\right]^{\alpha_p}} \qquad \text{Equation 3-605}$$

where:

$$\alpha_n = \mathbf{AN}\left(\frac{T}{300}\right)^{\mathbf{EXN4}} \qquad \text{Equation 3-606}$$

$$\alpha_p = \mathbf{AP}\left(\frac{T}{300}\right)^{\mathbf{EXP4}} \qquad \text{Equation 3-607}$$

where N_{total} is the local total impurity concentration in atoms/cm^3, and T is the temperature in Kelvins. The Arora mobility model is selected with the **MOB.AROR** parameter on the **METHOD** statement. The default parameter values used by the Arora model may be modified with the **MOBILITY** statement.

Caughey Mobility Model

The Caughey mobility model Reference [101] is given by the expressions:

$$\mu_{0n} = \mathbf{MUN.MIN} + \frac{\mathbf{MUN.MAX}\left(\frac{T}{300}\right)^{\mathbf{NUN}} - \mathbf{MUN.MIN}}{1 + \left(\frac{T}{300}\right)^{\mathbf{XIN}}\left(\frac{N_{total}}{\mathbf{NREFN}}\right)^{\mathbf{ALPHAN}}} \qquad \text{Equation 3-608}$$

$$\mu_{0p} = \mathbf{MUP.MIN} + \frac{\mathbf{MUP.MAX}\left(\frac{T}{300}\right)^{\mathbf{NUP}} - \mathbf{MUP.MIN}}{1 + \left(\frac{T}{300}\right)^{\mathbf{XIP}}\left(\frac{N_{total}}{\mathbf{NREFP}}\right)^{\mathbf{ALPHAP}}} \qquad \text{Equation 3-609}$$

where N_{total} is the local total impurity concentration in atoms/cm^3, and T is the temperature in Kelvins. The Caughey mobility model is selected with the **MOB.CAUG** parameter on the **METHOD** statement. The default parameter values used by the Caughey model can be modified with the **MOBILITY** statement.

Quantum Mechanical Model for MOSFET

The high electric fields caused by thin gate oxides and high substrate doping quantize electron motion perpendicular to semiconductor/insulator interfaces in MOSFET inversion layers. This affects threshold voltages, inversion layer areal charge densities, and capacitance-voltage characteristics. A robust solution for the quantum effect can be achieved by solving the Schrö dinger equation coupled with

Poisson's equation. However, it takes considerable CPU time to solve the Schrö dinger equation. In TSUPREM-4, the quantum effect is taken into account in an approximate manner by using a method suggested by van Dort, et al Reference [102]. This approach provides a reasonable solution in spite of very short CPU time.

The approximate bandgap-widening effect in the inversion layer due to quantization is:

$$\Delta E_g = \frac{13}{9}\mathbf{QM.BETA}\left(\frac{\varepsilon_{Si}}{4kT}\right)^{1/3}|E_n(0)|^{2/3} \qquad \text{Equation 3-610}$$

where $E_n(0)$ is the electric field at the interface.

The bandgap-widening effect causes a decrease in the intrinsic carrier concentration in the inversion layer:

$$n_{i,\,QM} = n_{i,\,CL}\exp\left(\frac{-\Delta E_g}{2kT}\right) \qquad \text{Equation 3-611}$$

in which $n_{i,CL}$ is the classical model for the intrinsic carrier concentration. However, this model should be applied to the region where the electrons are confined to a layer near the interface. The smoothing function *s(a)* is used to describe the transition:

$$s(a) = 2\frac{\exp(-a^2)}{(1+\exp(-2a^2))} \qquad \text{Equation 3-612}$$

in which $a = y/$**QM.YCRIT**, where y is the distance from the surface of an inversion layer. Thus, the intrinsic carrier concentration becomes:

$$n_i = n_{i,\,CL}(1-s(a)) + s(a)n_{i,\,QM} \qquad \text{Equation 3-613}$$

QM.BETA and **QM.YCRIT** are parameters on the **MATERIAL** statement. The quantum mechanical model is used when **QM** is specified on the **ELECTRICAL** statement.

Capacitance Calculation

To achieve the accurate capacitance dependent on the frequency and waveform of input small signal, the transient equation must be solved by device simulators.

DC Method

The capacitance calculation in TSUPREM-4 is based on the DC method which extracts the charge quantities at different DC biases. The voltage difference is specified in the **METHOD** statement. If **E.USEAVC** is false (default), the relative ratio **E.RVCAP** to the DC bias step is used.

$$C = \frac{Q(V + \mathbf{E.RVCAP} \cdot \ VSTEP) - Q(V - \mathbf{E.RVCAP} \cdot \ VSTEP)}{2 \cdot \ \mathbf{E.RVCAP} \cdot \ VSTEP}$$

Equation 3-614

Otherwise, the absolute voltage difference **E.AVCAP** is used:

$$C = \frac{Q(V + \mathbf{E.AVCAP}) - Q(V - \mathbf{E.AVCAP})}{2 \cdot \ \mathbf{E.AVCAP}}$$

Equation 3-615

If only one bias is given so that **VSTEP** is zero, Equation 3-615 is used regardless of **E.USEAVC** status.

MOS Capacitances

There are three kinds of measurement for MOS capacitance.

- Slow DC sweep and low-frequency AC small signal
- Fast DC sweep and high-frequency AC small signal
- Slow DC sweep and high-frequency AC small signal

The parameters **LOW**, **DEEP**, and **HIGH** in the **ELECTRICAL** statement correspond to the above measurements in order. Since TSUPREM-4 does not permit AC analysis, the high-frequency effect is modeled by applying a bias not only to the majority carriers, but also to the minority carriers. This has no effect on the C-V characteristics up to the threshold level, since the charge storage mechanism in this region is due primarily to modulation of majority carriers at the silicon surface. However, above the threshold voltage, setting the majority Fermi-level to a value higher than the maximum applied gate bias, prevents the build-up of minority carriers at the surface of the silicon and results in the high-frequency effect. However, this approximation can cause inaccurate results when a lot of intrinsic carriers are generated at high temperature.

References

[1] C. P. Ho, S. E. Hansen, and P. M. Fahey. "SUPREM III—A Program for Integrated Circuit Process Modeling and Simulation," Technical Report No. SEL84-001, Stanford Electronics Laboratories, Stanford University, Stanford, California, July 1984.

[2] G. Masetti, M. Severi, and S. Solmi. "Modeling of Carrier Mobility Against Carrier Concentration in Arsenic-, Phosphorus-, and Boron-Doped Silicon," *IEEE Trans. Electron Devices*, Vol. ED-30, No. 7, pp. 764-769, July 1983.

[3] B. E. Deal. "Thermal Oxidation Kinetics of Silicon in Pyrogenic H_2O and 5% HCl/H_2O Mixtures," *J. Electrochem. Soc.*, Vol. 125, No. 4, pp. 576-579, April 1978.

[4] R. E. Bank, W. M. Coughran Jr., W. Fichtner, E. H. Grosse, D. J. Rose, and R. K. Smith. "Transient Simulation of Silicon Devices and Circuits," *IEEE Trans. Elec. Dev.*, Vol. ED-32, No. 10, pp. 1992-2007, Oct. 1985.

[5] R. B. Fair and P. N. Pappas. "Diffusion of Ion-Implanted B in High Concentration P- and As-Doped Silicon," *J. Electrochem. Soc.*, Vol. 122, No. 9, pp. 1241-4, Sept. 1975.

[6] N. E. B. Cowern and D. J. Godfrey. "A Model for Coupled Dopant Diffusion in Silicon," *Fundamental Research on the Numerical Modelling of Semiconductor Devices and Processes (Papers from NUMOS I, the First International Workshop on the Numerical Modelling of Semiconductors, 11th - 12th December 1986, Los Angeles, USA)*, edited by J. J. H. Miller, Dublin, Ireland: Boole Press, 1987.

[7] F. Wittel and S. T. Dunham. "Diffusion of Phosphorus in Arsenic and Boron Doped Silicon," *Appl. Phys. Lett.*, Vol. 66, p. 1415, 1995.

[8] M. E. Law, H. Park, and P. Novell. "Theory of Dopant Diffusion Assuming Nondilute Concentrations of Dopant-Defect Pairs," *Appl. Phys. Lett.,* Vol. 59, No. 26, pp. 3488-3489, Dec. 1991.

[9] N. E. B. Cowern. "General Model for Intrinsic Dopant Diffusion in Silicon Under Nonequilibrium Point-Defect Conditions," *J. Appl. Phys.,* Vol. 64, No. 9, pp. 4484-4490, Nov. 1988.

[10] F. A. Trumbore. "Solid Solubilities of Impurity Elements in Germanium and Silicon," *Bell Sys. Tech. J.*, pp. 205-233, Jan. 1960.

[11] Y. Oh and D. E. Ward, "A Calibrated Model for Trapping of Implanted Dopants at Material Interface During Thermal Annealing," 1998 IEDM *Tech. Dig.*, pp. 509-512, 1998.

[12] S. M. Hu. "On Interstitial and Vacancy Concentrations in Presence of Injection," *J. Appl. Phys.*, Vol. 57, p. 1069, 1985.

[13] P. B. Griffin and J. D. Plummer. “Process Physics Determining 2-D Impurity Profiles in VLSI Devices,” *IEDM Technical Digest,* pp. 522-525, 1986.

[14] P. B. Griffin. “Physics and Modeling of 2-D Diffusion in SUPREM-IV,” Doctoral dissertation, Stanford University, Palo Alto, California, 1989.

[15] C. S. Rafferty, G. H. Gilmer, J. Jaraiz, D. Eaglesham, and H.-J. Gossmann, “Simulation of Cluster Evaporation and Transient Enhanced Diffusion in Silicon,” *Appl. Phys. Lett.*, Vol. 68, No. 17, pp. 2395-7, 22 April 1996.

[16] J. M. Poate, D. J. Eaglesham, G. H. Gilmer, H.-J. Gossmann, M. Jaraiz, C. S. Rafferty, and P. A. Stolk, “Ion Implantation and Transient Enhanced Diffusion,” 1995 IEDM *Tech. Dig.*, pp. 77-80, 1995.

[17] I. Bork and W. Molzer, “Appropriate Initial Damage Conditions for Three-Stream Point Defect Diffusion Models”, *SISPAD2000 Proc.*, pp.175-178, 2000.

[18] A. H. Gencer, “Modeling and Simulation of Transient Enhanced Diffusion Based on Interactions of Point and Extended Defects,” Doctoral dissertation, Boston University College of Engineering, 1999.

[19] B. E. Deal and A. S. Grove. “General Relationship for the Thermal Oxidation of Silicon,” *J. Appl. Phys.*, Vol. 36, p. 3770, 1965.

[20] H. Z. Massoud. “Thermal Oxidation of Silicon in Dry Oxygen—Growth Kinetics and Charge Characterization in the Thin Regime,” Technical Report, Stanford Electronics Laboratories, Stanford University, Stanford, California, June 1983.

[21] D. W. Hess and B. E. Deal. “Kinetics of the Thermal Oxidation of Silicon in O_2/HCl Mixtures,” *J. Electrochem. Soc.*, Vol. 124, No. 5, pp. 735-739, May 1977.

[22] R. R. Razouk, L. N. Lie, and B. E. Deal. “Kinetics of High Pressure Oxidation of Silicon in Pyrogenic Steam,” *J. Electrochem. Soc.*, Vol. 128, No. 10, pp. 2214-2220, Oct. 1981.

[23] L. N. Lie, R. R. Razouk, and B. E. Deal. “High Pressure Oxidation of Silicon in Dry Oxygen,” *J. Electrochem. Soc.*, Vol. 129, No. 12, pp. 2828-2834, Dec. 1982.

[24] N. Guillemot, G. Pananakakis, and P. Chenevier. “A New Analytical Model of the “Bird’s Beak,” *IEEE Trans. Elec. Dev.*, Vol. ED-34, No. 5, pp. 1033-1038, May 1987.

[25] C. P. Ho and J. D. Plummer. “Si/SiO_2 Interface Oxidation Kinetics: A Physical Model for the Influence of High Substrate Doping Levels—I. Theory,” *J. Electrochem. Soc.*, Vol. 126, No. 9, pp. 1516-1522, Sept. 1979.

[26] C. P. Ho and J. D. Plummer. “Si/SiO_2 Interface Oxidation Kinetics: A Physical Model for the Influence of High Substrate Doping Levels—II. Comparison with Experiment and Discussion,” *J. Electrochem. Soc.*, Vol. 126, No. 9, pp. 1523-1530, Sept. 1979.

[27] C. P. Ho, J. D. Plummer, and J. D. Meindl. “Thermal Oxidation of Heavily Phosphorus-Doped Silicon,” *J. Electrochem. Soc.*, Vol. 125, No. 4, pp. 665-671, April 1978.

[28] D. Chin, S. Y. Oh, S. M. Hu, R. W. Dutton, and J. L. Moll. “2-D Oxidation,” *IEEE Trans. Elec. Dev.*, Vol. ED-30, No. 7, pp. 744-749, July 1983.

[29] C. S. Rafferty. “Stress Effects in Silicon Oxidation—Simulation and Experiments,” Doctoral dissertation, Stanford University, Stanford, California, 1989.

[30] V. Senez, P. Ferreira, and B. Baccus. “2-D Simulation of Local Oxidation of Silicon: Calibrated Viscoelastic Flow Analysis,” *IEEE Trans. Electron Dev.,* Vol. 43, No. 5, pp. 720–731, May 1996.

[31] R. B. Fair, “Physical Models of Boron Diffusion in Ultrathin Gate Oxides,” *J. Electrochem. Soc.,* Vol. 144, No. 2, pp. 708-717, Sept. 1997.

[32] A. M. Agarwal and S. T. Dunham, “Consistent Quantitative Model for the Spatial Extent of Point Defect Interactions in Silicon,” *J. Appl. Phys.,* Vol. 78, No. 9, pp. 5313-5319, Nov. 1995.

[33] S. Nedelec, D. Mathiot, E. Andre, A. Straboni, and M. Gauneau. “Influence of Nitrogen Content on B Disffusion in SiO_2”, *Silicon Nitride and Silicon Dioxide Thin Insulating Films - The Electrochemical Society Proceedings Series*, pp.295-304, Penington, NJ 1994.

[34] T. Aoyama, K. Suzuki, H. Tashiro, Y. Toda, T. Yamazaki, K. Takasaki, and T. Ito. “Effect of Fluorine on Boron Diffusion in Thin Silicon Dioxides and Oxynitride,” *J. Appl. Phys.,* Vol. 77, No. 1, pp. 417-419, Jan. 1995.

[35] V. Moroz and T. Okada, “Modeling of Self-Aligned Silicidation in 2D and 3D: Growth Suppression by Oxygen Diffusion,” *Mat. Res. Soc. Symp.*, Vol. 611, p. C9.3, 2000.

[36] S. L. Zhang, R. Buchta, and M. Ostling. “A Study of Silicide Formation form LPCVD-Tungsten Films: Film Texture and Growth Kinetics,” *J. Mater. Res.,* Vol. 6, pp. 1886-1891, Sept. 1991.

[37] Locker and C. D. Capio, “Reaction Kinetics of Tungsten Thin Films on Silicon (100) Surfaces,” *J. Appl. Phys.,* Vol. 44, No. 10, pp. 4366-4369, Oct. 1973.

[38] E. Ma, B. S. Lim, M. A. Nicolet, N. S. Alvi, and A. H. Hamdi. *J. Electron. Mater.,* Vol. 17, p. 207, 1988.

[39] C. L. Chu, K. C. Saraswat, and S. S. Wong. “Measurement of Lateral Dopant Diffusion in Thin Silicide Layers,” *IEEE Trans. Elec. Dev.,* Vol. 39, No. 10, pp. 2333-2340, Oct. 1992.

[40] V. Probst, H. Schaber, A. Mitwalsky, and H. Kabza. “WSi2 and CoSi2 as Diffusion Sources for Shallow-Junction Formation in Silicon,” *J. Appl. Phys.,* Vol. 70, No. 2, pp. 708-719, July 1991.

[41] M. Y. Tsai, F. M. d'Heurle, C. S. Petersson, and R. W. Johnson. "Properties of Tungsten Silicide Film on Polycrystalline Silicon," *J. Appl. Phys.,* Vol. 52, No. 8, pp. 5350-5355, Aug. 1981.

[42] G. Giroult, A. Nouailhat, and M. Gauneau. "Study of a WSi2/Polycrystalline Silicon/Monocrystalline Silicon Structure for a Complementary Metal-Oxide-Semiconductor for a Compatible Self-Aligned Bipolar Transistor Emitter," *J. Appl. Phys.,* Vol. 67, No. 1, pp. 515-523, Jan. 1990.

[43] C. M. Comrie and R. T. Newman, "Dominant diffusing species during cobalt silicide formation," *J. Appl. Phys.*, Vol. 79, No. 1, pp. 153 - 156, 1996.

[44] T. Barge, "Analysis of the diffusion controlled growth of cobalt silicides in bulk and thin film couples," *J. Mater. Res.*, Vol. 10, No. 5, pp. 1134 - 1145, 1995.

[45] S. B. Herner, K. S. Jones, H.-J. Gossmann *et al.*, "The influence of TiSi2 and CoSi2 growth on Si native point defects: The role of the diffusing species," *Appl. Phys. Lett.*, Vol. 68, No. 20, pp. 2870 - 2872, 1996.

[46] S. B. Herner, H.-J. Gossmann and R. T. Tung, "Native point defects in Si with a CoSi2 film: Single crystal versus polycrystalline," *Appl. Phys. Lett.*, Vol. 72, No. 18, pp. 2289 - 2291, 1998.

[47] R. Stadler, W. Wolf, R. Podloucky et al., "Ab initio calculations of the cohesive, elastic, and dynamical properties of CoSi2 by pseudopotential and all-electron techniques," *Phys. Rev. B*, Vol. 54, No. 3, pp. 1729 - 1734, 1996.

[48] T. L. Crandle, W. B. Grabowski, and M. R. Kump. "Empirically and Physically Based Approaches to Ion Implant Modeling," *Proceedings of the NASECODE VI Short Course on Software Tools for Process, Device and Circuit Modeling*, Dublin, Ireland, pp. 32-44, 1989.

[49] A. F. Tasch, H. Shin, C. Park, J. Alvis, and S. Novak. "An Improved Approach to Accurately Model Shallow B and BF2 Implants in Silicon," *J. Electrochem. Soc.,* Vol. 136, No. 3, March 1989.

[50] S. Morris, V. Ghante, L. M. Lam, S.-H. Yang, M. Morris, and A. F. Tasch, "An Accurate and Computationally Efficient Model for Phosphorus Implants into (100) Single-Crystal Silicon," *Proc. of the XIth International Conference on Ion Implantation Technology*, Austin, TX, June 16-21, 1996.

[51] M. Simard-Normandin and C. Slaby. "Empirical Modeling of Low Energy Boron Implants in Silicon," *J. Electrochem. Soc.,* Vol. 132, No. 9, pp. 2218-2223, Sept. 1985.

[52] C. Park, K. Klein, A. Tasch, R. Simonton, and S. Novak. "Comprehensive Modeling of Boron Ion Implantation for the ULSI Era," Extended Abstract Volume, TECHCON'90 (San Jose, CA), pp. 443-446, 1990.

[53] J. F. Gibbons. *Handbook on Semiconductors,* Vol. 3, Chapter 10, edited by T. S. Moss and S. P. Keller, Amsterdam: North-Holland, 1980.

[54] W. K. Hofker. “Implantation of Boron in Silicon,” *Philips Res. Reports*, Suppl. No. 8, pp. 1-121, 1975.

[55] R. Tielert. “2-D Numerical Simulation of Impurity Redistribution in VLSI Processes,” *IEEE Trans. Elec. Dev.*, Vol. ED-27, No. 8, pp. 1479-1483, Aug. 1980.

[56] J.Amaratunga, K. Sabine, and A. G. R. Evans. “The Modeling of Ion Implantation in a Three-Layer Structure Using the Method of Dose Matching,” *IEEE Trans. Elec. Dev.*, Vol. ED-32, No. 9, pp. 1889-1890, Sept. 1985.

[57] S. Furukawa, H. Matsumura, and H. Ishiwara. “Theoretical Considerations on Lateral Spread of Implanted Ions,” *Jap. J. Appl. Phys.*, Vol. 11, No. 2, pp. 134-142, Feb. 1972.

[58] G. Hobler and S. Selberherr. “2-D Modeling of Ion Implantation Induced Point Defects,” *IEEE Trans. Computer-Aided Design*, Vol. 7, No. 2, pp. 174-180, 1988.

[59] S. Tian, V. Moroz, and N. Strecker, “Accurate Monte Carlo Simulation of Ion Implantation into Arbitrary 1D/2D/3D Structures for Silicon Technology”, *Mat. Res. Soc. Symp. Proc.*, Vol. 810, pp. C6.5.1-C6.5.6, (2004).

[60] S. Tian, “Predictive Monte Carlo ion implantation simulator from sub-keV to above 10 MeV”, *J. Appl. Phys.*, Vol. 93, No. 10, pp. 5893-5904, (2003).

[61] S. Tian, “Accurate Monte Carlo simulation of fluorine and BF2 ion implantation into crystalline silicon”, *Nucl. Instr. Meth. Phys. Res. B*, Vol. 215, pp. 403-412, (2004).

[62] B. J. Mulvaney, W. B. Richardson, and T. L. Crandle. “PEPPER—A Process Simulator for VLSI,” *IEEE Trans. Computer-Aided Design*, Vol. 8, No. 4, pp. 336-349, 1989.

[63] H. Goldstein. *Classical Mechanics*, Reading, Massachusetts: Addison-Wesley, 1950.

[64] J. F. Ziegler, J. P. Biersack, and U. Littmark. *The Stopping and Ranges of Ions in Matter*, Vol. 1, New York: Pergamon Press, 1985.

[65] M. T. Robinson and I. M. Torrens. “Computer Simulation of Atomic Displacement Cascades in Solids in the Binary-Collision Approximation,” *Phys. Rev.*, Vol. B9, p. 5008, 1974.

[66] J. P. Biersack and L. G. Haggmark. “A Monte Carlo Computer Program for the Transport of Energetic Ions in Amorphous Targets,” *Nucl. Instr. and Meth.*, Vol. 174, p. 257, 1980.

[67] G. Hobler and S. Selberherr. “Efficient 2-D Simulation of Ion Implantation,” Proceedings of NASECODE V, Trinity College, Dublin, p. 225, 1987.

[68] J. Lindhard, M. Scharff, and H. E. Schiott. *K. Dan Viedensk. Selsk. Mat. Fys. Medd.*, Vol. 33, No. 1, 1963; J. Lindhard and M. Scharff, *Phys. Rev.*, Vol. 124, p. 128, 1961.

[69] D. V. Morgan. *Channeling*, New York: Wiley, 1973.

[70] R. G. Wilson. "Ion Channeling in GaAs: Be, Mg, Zn, and Cd, and Calculations of Electronic Stopping Powers," *J. Appl. Phys.*, Vol. 53, p. 5641, 1982.

[71] R. G. Wilson. "Random and Channeled Implantation Profiles and Range Parameters for P and Al in Crystalline and Amorphized Si," *J. Appl. Phys.*, Vol. 59, p. 2797-2805, Oct. 1986.

[72] R. G. Wilson and V. R. Deline. "Ion Channeling in GaAs: Si, S, Se, and Te," *Appl. Phys. Lett.*, Vol. 37, pp. 793-796, 1980.

[73] *Ion Beam Handbook for Material Analysis*, New York: Academic, 1977.

[74] T. Saito, H.Yamakawa, S. Komiya, H. J. Kang, and R. Shimuzu. "Dynamic Simulation of Ion Implantation with Damage Processes Included," *Nucl. Instr. and Meth.*, Vol. B21, p. 456, 1987.

[75] T. L. Crandle, W. B. Richardson, and B. J. Mulvaney. "A Kinetic Model for Anomalous Diffusion During Post-Implant Annealing," Intl. Elec. Dev. Mtng., *Tech. Digest*, p. 636, 1988.

[76] W. P. Maszara and G. A. Rozgonyi. "Kinetics of Damage Production in Silicon During Self-Implantation," *J. Appl. Phys.*, Vol. 60, p. 2310, 1986.

[77] T. L. Crandle and B. J. Mulvaney. "An Ion Implantation Model Incorporating Damage Calculations in Crystalline Targets," *IEEE Elec. Dev. Lett.*, Vol. 11, No. 1, 1990.

[78] L. Pelaz, G. H. Gilmer, M. Jaraiz, S. B. Herner, H.-J. Gossmann, D. J. Eaglesham, G. Hobler, C. S. Rafferty, and J. Barbolla, "Modeling of the Ion Mass Effect on Transient Enhanced Diffusion: Deviation from the '+1' Model,", *Appl. Phys. Lett.,* Vol. 73, No. 10, pp. 1421–1423, 7 Sep 1998.

[79] D. Chidambarrao and R. Young, private correspondence.

[80] S. Chakravarthi and S. T. Dunham, "Influence of Extended Defect Models on Prediction of Boron Transient Enhanced Diffusion," in *Silicon Front End Technology—Materials Processing and Modeling,* N. Cowern, P. Griffin, D. Jacobsen, P. Packan, and R. Webb, Eds. (Mat. Res. Soc. Proc. Vol. 532, Pittsburgh, PA, 1998).

[81] G. Hobler and V. Moroz, "Simple Formulae for the Effective Plus-Factor for Transient Enhanced Diffusion," ESSDERC 2000, *Proceedings of the 30th European Solid-State Device Research Conference*, Cork, Ireland, pp. 168-71.

[82] S. K. Jones and A. Gerodolle. "2D Process Simulation of Dopant Diffusion in Polysilicon," *NASECODE VII Transactions,* COMPEL, Vol. 10, No. 4, pp. 401-410, 1991.

[83] A. Gerodolle and S. K. Jones. "Integration in the 2D Multi-Layer Simulator TITAN of an Advanced Model for Dopant Diffusion in Polysilicon," in *Simulation of Semiconductor Devices and Process*, Vol. 4, pp. 381-388, edited by W. Fichtner and D. Aemmer, Hartning-Gorre Verlag (Konstanz), 1991.

[84] S. K. Jones, A. Gerodolle, C. Lombardi, and M. Schafer. "Complete Bipolar Simulation Using STORM," IEDM-92, *Tech. Digest*, p. 931, 1992.

[85] A. G. O'Neill, C. Hill, J. King, and C. Please. "A New Model for The Diffusion of Arsenic in Polycrystalline Silicon," *J. Appl. Phys.*, Vol. 64, No. 1, p. 167, 1988.

[86] M. M. Mandurah, K. C. Saraswat, R. W. Dutton, and T. I. Kamins. "Dopant Segregation in Polysilicon," *J. Appl. Phys.*, Vol. 51, pp. 5755, 1981.

[87] L. Mei and R. W. Dutton. "A Process Simulation Model for Multilayer Structures Involving Polycrystalline Silicon," *IEEE Trans. Elec. Dev.,* Vol. ED-29, No. 11, p. 1726, 1982.

[88] L. Mei, M. Rivier, Y. Kwark, and R. Dutton. "Grain Growth Mechanisms in Polysilicon," *J. Electrochem. Soc.,* Vol. 129, No. 8, p. 1791, 1982.

[89] C. V. Thompson. "Secondary Grain Growth in Thin Films of Semiconductors: Theoretical Aspects," *J. Appl. Phys.,* Vol. 58, No. 2, p. 763, 1985.

[90] D. Gupth, D. R. Campbell, and P. S. Ho. *Thin Films—Interdiffusion and Reactions,* New York: Wiley, p. 161, 1980.

[91] C. Hill and S. K. Jones. "Modelling Dopant Diffusion in and from Polysilicon," *Mat. Res. Symp. Proc.,* No. 182, p. 129, 1990.

[92] S. Ajuria and R. Reif. "Early Stage Evolution Kinetics of the Polysilicon/Single-Crystal Silicon Interfacial Oxide Upon Annealing," *J. Appl. Phys.*, Vol. 69, No. 2, p. 662, 1991.

[93] J. D. Williams. "Epitaxial Alignment of Polycrystalline Silicon and its Implications for Analogue Bipolar Circuits," Doctoral dissertation, University of Southampton, Faculty of Engineering and Applied Science, 1992.

[94] F. Benyaich, F. Priolo, E. Rimini, C. Spinella, and P. Ward. "Kinetic and Structural Study of the Epitaxial Realignment of Polycrystalline Si Films," *J. Appl. Phys.*, Vol. 71, No. 2, p. 638, 1992.

[95] E. A. Irene. "Residual Stress in Silicon Nitride Films," *J. Electronic Mat.*, Vol. 5, No. 3, p. 287, 1976.

[96] S. M. Hu. "Film-Edge-Induced Stress in Silicon Substrates," *Appl. Phys. Lett.*, Vol. 32, p. 5, 1978.

[97] S. M. Sze. *Physics of Semiconductor Devices*, New York: Wiley-Interscience, pp. 28-34, 1969.

[98] K. Yamaguchi. “Field-Dependent Mobility Model for 2-D Numerical Analysis of MOSFET’s,” *IEEE Trans. Electron Devices*, Vol. ED-206, pp. 1068-1074, July 1979.

[99] K. Yamaguchi. “A Mobility Model for Carriers in the MOS Inversion Layer,” *IEEE Trans. Electron Devices*, Vol. ED-30, pp. 658-663, June 1983.

[100] N. D. Arora, J. R. Hauser, and D. J. Roulston. “Electron and Hole Mobilities in Silicon as a Function of Concentration and Temperature,” *IEEE Trans. Electron Devices*, Vol. ED-29, pp. 292-295, Feb. 1982.

[101] D. M. Caughey and R. E. Thomas. “Carrier Mobilities in Silicon Empirically Related to Doping and Field,” *Proc. IEEE*, Vol. 55, pp. 2192-2193, 1967.

[102] M. J. van Dort, P. H. Woerlee, and A. J. Walker. “A Simple Model for Quantization Effects in Heavily-Doped Silicon MOSFETs at Inversion Conditions,” *Solid State Electronics*, Vol. 37, No. 3, pp. 411-414, 1994.

[103] R. Y. S. Huang and R. W. Dutton. “Experimental Investigation and Modeling of the Role of Extended Defects During Thermal Oxidation,” *J. Appl. Phys.*, Vol. 74, No. 9, pp. 5821-5827, Nov. 1993.

[104] S. M. Hu. in *Defects in Semiconductors,* edited by J. Narayan and T. Y. Tan, Amsterdam: North-Holland, pp. 333-354, 1981.

[105] H. Cerva and G. Hobler. “Comparison of Transmission Electron Microscope Cross Sections of Amorphous Regions in Ion Implanted Silicon with Point-Defect Density Calculations,” *J. Electrochm. Soc.,* Vol. 139, No. 12, pp. 3631-8, 1992.

[106] R. Rios, et al. “A Physical Compact MOSFET Model, Including Quantum Mechanical Effects for Statistical Circuit Design Applications,” IEDM ´95 *Tech. Digest*, pp. 937-940.

[107] K. B. Parab, S. -H. Yang, S. J. Morris, S. Tian, M. Morris, B. Obradovich, A. F. Tasch, D. Kamenitsa, R. Simonton, and C. Magee. “Detailed Analysis and Computationally Efficient Modeling of Ultra-Shallow Profiles Obtained by Low Energy B, BF2, and As Ion Implantation,” *MRS* Fall Meeting, Boston, 1995.

Input Statement Descriptions

The TSUPREM-4 user directs the program via input statements. These statements can appear in an input command file or can be entered directly from the user's terminal. They also appear in the *s4init* file.

This chapter describes the statements recognized by TSUPREM-4. The first section gives the general format of the input and defines the syntax used in the detailed documentation of the following sections. The following sections include:

- Documentation and Control, p. 4-5: discusses statements that control the execution of TSUPREM-4.
- Device Structure Specification, p. 4-52: discusses statements that specify the device structure used by TSUPREM-4.
- Process Steps, p. 4-98: discusses statements that simulate processing steps.
- Output, p. 4-145: discusses statements to print and plot results, and extract, structural, doping, and electrical characteristics.
- Models and Coefficients, p. 4-206: discusses statements that specify the models and coefficients used by TSUPREM-4.

Input Statements

The input to TSUPREM-4 consists of various statements. This section describes the format and syntax of these statements.

Note:
The input syntax recognized by TSUPREM-4 differs slightly from that used by other Synopsys TCAD products.

Format

Each statement consists of a statement name followed by a list of parameters. Valid statement and parameter names are described in the following sections of this manual. Statements have the following general format:

- Statement and parameter names are recognized in either upper or lower case.
- Usually, each statement is terminated by the end of an input line. If a statement requires more than one input line, each line except the last must end with a plus (+) as the last nonblank character. A statement is automatically continued if the end of line occurs inside a quoted character string.
- A statement (other than **`COMMENT`**, **`DEFINE`**, or **`UNDEFINE`**) can also be terminated by a semicolon (;) appearing outside of a quoted character string; this allows more than one statement to be placed on a line.
- Blank lines are ignored.

Syntax

Parameter and statement names can be abbreviated by dropping characters from the end of the name. Ambiguous abbreviations are not permitted. The abbreviation for a parameter name must be long enough to distinguish among the parameters associated with a statement. The abbreviation for a statement name must be long enough to distinguish it from other statements. Parameter and statement names can also be extended by adding characters to the end of the unabbreviated name.

Most of the statements recognized by TSUPREM-4 obey a single, simple set of syntax rules. A few statements are treated specially, however. These are referred to as *special* statements in the discussion that follows.

Specifying Materials and Impurities

Many statements require the specification of materials. In all cases, materials can be specified by name using the **`MATERIAL`** and **`/MATERIA`** parameters. The material names recognized by these parameters are *silicon, polysili, oxide, oxynitri, nitride, photores, aluminum, titanium, TiSi2, tungsten, WSi2,* and the names of any user-defined materials. *Note* that material names are not case sensitive, and can be abbreviated or extended like parameters or statements names. Some statements also have the names of materials as parameters; only the materials available in older versions of TSUPREM-4 can be specified in this way.

Many statements require the specification of impurities or solution values. In all cases, impurities and solution values can be specified by name using the **`IMPURITY`** parameter. Names of impurities and solution values are not case sensitive. *Note* that the list of meaningful names depends on the semantics of the statement: some statements only take the names of impurities, while others work with any solution value. Some statements also have the names of impurities or solution values as parameters; only the impurities and solution values available in older versions of TSUPREM-4 can be specified in this way.

Parameters

The parameters are separated from the statement name by one or more spaces. There are three types of parameters.

- Character
- Logical
- Numerical

Special statements may require other types of parameters, specified in a particular order. These are described in the detailed statement description sections.

Character

Character parameters require that a value be specified. The value is separated from the parameter name by an equals character (=). Character parameters can appear in one of two ways, depending on the statement involved:

- A named character parameter is assigned a string value using an equals character in the same manner as a numerical parameter. If the string contains spaces, it must be enclosed in quotes (`"`).
- An unnamed character parameter is used when a statement consists of a single input line containing the statement name followed by an arbitrary character string. In this case there is no parameter name or equals character (=), and quotes are not needed

Logical

Logical parameters are set true by specifying the parameter name only, and can be set false by preceding the name by one of the *not* characters, “^”, “!”, or “#”.

Numerical

Numerical parameters require that a value be specified. The value is separated from the parameter name by an equals character (=). Blanks on either side of the equals character are ignored. The value can be specified as an integer or fixed-point decimal number, or as a floating-point number using the character “E” or “e” to delimit the exponent (e.g., 101, 101.0, or 1.01E2).

The value of a numerical parameter can also be specified as a mathematical expression. The mathematical operators +, –, *, /, and ^ (for exponentiation) can be used, as well as a variety of mathematical functions. See the description of the **`SELECT`** statement on p. 4-146 for a list. If a mathematical expression includes spaces, the entire expression must be enclosed in parentheses. If an expression is given where an integer value is required, the value of the expression is rounded to the nearest integer.

Statement Description Format

The remainder of this chapter describes the input statements recognized by TSUPREM-4. The description of each statement consists of a formatted list of the parameters associated with the statement. This is followed by the parameter definition table.

Parameter Definition Table

The parameter definition table includes the following:

- Parameter name
- Parameter type
- A description of the parameter's function
- Synonyms (if any) which can be used instead of the standard parameter name
- The physical units (if any) for a numerical parameter
- The default value TSUPREM-4 uses in the absence of a user-specified value

Syntax of Parameter Lists

Several special characters are used in the formatted parameter list that appears at the beginning of each statement description:

- Angle brackets **< >**
- Square brackets **[ ]**
- Vertical bar **|**
- Braces **{ }**
- Parentheses **( )**

Note:

The special characters, < >, [ ], |, { }, and ( ), indicate parameter types, optional groups, alternate choices, and group hierarchy. They do not form part of the actual input to TSUPREM-4 (i.e., these special characters are not typed in). Only the information enclosed in the special characters is typed into command strings.

Parameter Types < >

A lower case letter in angle brackets indicates the type of a parameter. Thus **`<n>`** represents the value of a numerical parameter and **`<c>`** represents the value of a character parameter. Logical parameters are denoted by a parameter name without a value. For example,

```
WIDTH=<n>
```

indicates that the **`WIDTH`** parameter is assigned a numerical value. In a few special cases a descriptive word is used in angle brackets to denote a nonstandard parameter.

Parameter Groups

In the following, the term *group* refers to a parameter by itself or a set of parameters enclosed in a matched pair of square brackets, braces, or parentheses. For example,

```
{( PARM1 [PARM2 [PARM3]] PARM4 ) PARM5}
```

constitutes a valid group, composed of the subgroups **`( PARM1 [PARM2 [PARM3]] PARM4 )`** and **`PARM5`**. The first subgroup can further be subdivided into the subgroups **`PARM1`**, **`[PARM2 [PARM3]]`** and **`PARM4`**, etc.

Optional Parameters [ ]

Square brackets enclose groups that are optional. For example,

```
NEWCARD [PARM1] [ PARM2 PARM3 ] [ PARM4 [PARM5] ]
```

indicates that in the **`NEWCARD`** statement, the parameter **`PARM1`** is optional. The group **`[PARM2 PARM3]`** is optional, but if **`PARM2`** is specified, then **`PARM3`** must also be specified. The group **`[ PARM4 [PARM5] ]`** is optional; **`PARM5`** can be specified only if **`PARM4`** is specified.

Choices { }, |

When one of a list of groups must be selected, the groups are enclosed in braces and separated by vertical bars. For example,

```
NEWCARD {PARM1 | PARM2 | ( PARM3 PARM4 )}
```

indicates that the **`NEWCARD`** statement requires that one and only one of the three groups **`PARM1`**, **`PARM2`**, or **`( PARM3 PARM4 )`** be specified.

Group Hierarchy ( )

Parentheses enclose groups that are to be considered as single items in higher level groupings. For example, in the above **`NEWCARD`** statement, the group **`( PARM3 PARM4 )`** constitutes one of three possible choices, and is therefore enclosed in parentheses.

Documentation and Control

The following statements control execution of TSUPREM-4:

Statement Name	Description	Page
`COMMENT`	Documents the input file.	p. 4-7
`SOURCE`	Reads input statements from a file.	p. 4-8
`RETURN`	Exits from a sourced input file or from interactive mode.	p. 4-9
`INTERACTIVE`	Enters interactive input mode.	p. 4-11
`PAUSE`	Temporarily interrupts execution of the program.	p. 4-12

`STOP`	Stops execution of the program.	p. 4-13
`FOREACH/end`	Repeats a group of input statements.	p. 4-14
`LOOP/l.end`	Begins an input statement loop.	p. 4-16
`L.MODIFY`	Modifies processing of an input statement loop.	p. 4-20
`IF/ELSEIF/ELSE/IF.END`	Begins and terminates a sequence of one or more conditionally processed input statement blocks.	p. 4-21
`ASSIGN`	Assigns values to an assigned name.	p. 4-23
`INTERMEDIATE`	Species intermediate values	p. 4-30
`ECHO`	Sends a string to the output file.	p. 4-40
`OPTION`	Selects level of printed output and device for plotted output.	p. 4-41
`DEFINE`	Defines macros (abbreviations).	p. 4-46
`UNDEFINE`	Undefines macros.	p. 4-49
`CPULOG`	Enables or disables reporting of execution times.	p. 4-50
`HELP`	Prints a brief description of statements and parameters.	p. 4-51
`!(exclamation mark)`	Unix shell command call	p. 4-51

COMMENT

The `COMMENT` statement is used to specify character strings for documenting the user input and the program output.

```
COMMENT
   [<c>]
or
$
   [<c>]
```

Description

The `COMMENT` statement is used to document the input file. `COMMENT` statements can appear at any point in the input file.

Examples

- The following is a simple comment:

```
COMMENT   A SHORT COMMENT
```

- The “$” form is easier to type and stands out from other statements:

```
$ The "$" form is simpler
```

- Comments can be continued using the “+” character:

```
$ THIS IS A LONGER COMMENT WHICH HAS BEEN +
CONTINUED USING THE "+" CHARACTER.
```

- It is simpler and less confusing to just put a “$” before each line of a long comment:

```
$ THIS IS A LONG COMMENT WHICH SPANS TWO
$   LINES WITHOUT USING A CONTINUATION CHARACTER
```

Notes

The `COMMENT` statement has the following syntax considerations:

- The `$` form of the `COMMENT` statement may produce unexpected results if the “$” character is immediately followed by the name of a defined abbreviation. (See description of the `DEFINE` statement.) In that case, the “$” character and the name are replaced by the expansion of the abbreviation.
- A `COMMENT`, like any other statement, can be continued by placing a “+” character at the end of the line.
- A `COMMENT`, unlike most other statements, is not terminated by a semicolon (;).

SOURCE

The **SOURCE** statement causes TSUPREM-4 to read input statements from a file.

```
SOURCE

    <filename>
```

Description

The **SOURCE** statement causes TSUPREM-4 to read input statements from the specified file. **SOURCE** statements can be nested.

Reusing Combinations of Statements

The **SOURCE** statement is useful for reading in previously defined sequences of input statements. Frequently used combinations of statements, such as implant/anneal or mask/expose/etch sequences, can be placed in files and read with the **SOURCE** statement.

Generating Templates

You can use the **SOURCE** statement with the **DEFINE** or **ASSIGN** statements to generate template files with variable input values. A template file can be constructed by replacing portions of the input, such as character strings, parameter names, and parameter values, with defined names and numerical expressions including defined names. An input file that uses the template file must first include **DEFINE** or **ASSIGN** statements that set the values of defined names appearing in the template file and then include a **SOURCE** statement that references the template file. The values of the defined names in the template file are replaced by the values specified in the original input file.

Examples

The following defines the variable *PTITLE* and then reads input statements from file *DOPLOT:*

```
DEFINE  PTITLE  "After Source/Drain Implant"
SOURCE  DOPLOT
```

The file *DOPLOT* might contain statements to plot the structure:

```
SELECT  Z=LOG10(BORON) TITLE=@PTITLE
PLOT.2D SCALE Y.MAX=2
FOREACH X (15 TO 21)
   CONTOUR  VALUE=X
 END
```

The value of the variable *PTITLE* is used by the called file as the title of the plot.

RETURN

The **RETURN** statement exits the current input mode and returns to the previous level of input. **BATCH** is a synonym for the **RETURN** statement.

```
RETURN

    [<c>]
```

Description

The character string associated with the **RETURN** statement is ignored by the program, and serves only to document user input.

The **RETURN** statement is used in batch input mode to exit an input command file before the end-of-file. Input statements after the **RETURN** statement are ignored.

Returning from Batch Mode

Three possibilities exist when returning from batch mode:

- If TSUPREM-4 is started in batch input mode by specifying the input command file at the beginning of program execution, then execution of the program terminates after the **RETURN** statement.
- If batch input mode is entered by executing the **SOURCE** statement from interactive input mode, then interactive input mode resumes.
- If batch input mode is entered by executing a **SOURCE** statement from another input command file, then processing resumes with the statement following the **SOURCE** statement.

Exiting Interactive Input Mode

The **RETURN** (or **BATCH**) also statement is used to exit interactive input mode. There are two possibilities:

- If interactive input mode is entered by executing the **INTERACTIVE** statement in an input command file, then processing resumes with the statement following the **INTERACTIVE** statement.
- If TSUPREM-4 is started in interactive input mode at the beginning of program execution, then execution of the program terminates after the **RETURN** statement.

Typical end-of-file characters are *control-D* (*EOT*) and *control-Z* (*SUB*). An end-of-file is equivalent to the **RETURN** statement. Thus, interactive input mode can be terminated either with a **RETURN** statement or an end-of-file character. Furthermore, a **RETURN** statement is not necessary at the end of an input command file.

Example Suppose an input file contains the following lines:

```
$ LINE 1
$ LINE 2
RETURN
$ LINE 4
LINE 5 (BAD SYNTAX!)
```

When this file is read, lines 1 and 2 were executed, but lines 4 and 5 were ignored. *Note* that no syntax check is performed on lines that follow a `RETURN` statement.

INTERACTIVE

The **`INTERACTIVE`** statement starts the interactive input mode, allowing statements to be entered interactively from the terminal.

```
INTERACTIVE

[<c>]
```

Description

The character string associated with the **`INTERACTIVE`** statement is ignored by the program, and serves only to document user input.

Interactive input mode can be initiated using either of the following methods:

- An **`INTERACTIVE`** statement is executed from a batch input file. In this case, the program resumes processing statements from the batch input file when interactive input is terminated.
- A blank file specification is given for the command input file at the beginning of program execution. In this case, the program automatically enters interactive input mode. All statements are entered interactively, and the program terminates when interactive input is terminated.

Interactive Input Mode

When interactive input mode is started, the program indicates this by printing a message on the terminal, printing the interactive input prompt ("TS4"), and awaiting input of statements. A statement can be continued on a subsequent line by ending the current input line with a plus (**+**). Continuation can be used repeatedly to generate input statements consisting of any number of input lines. The program indicates that continuation lines are expected by changing the interactive input prompt to ">" until the statement is complete. A continued statement can be completed by not ending the last line with a plus (**+**) or by entering a blank line.

Terminate interactive input mode either by entering a **`RETURN`** statement or by entering an end-of-file during interactive input from the terminal. Typical end-of-file characters are *control-D* (*EOT*) and *control-Z* (*SUB*).

Example

Suppose the following input file is executed:

```
$ LINE 1
$ LINE 2
INTERACTIVE
$ LINE 4
$ LINE 5
```

Lines 1 and 2 are executed. When the **`INTERACTIVE`** statement is executed, the program prompts for input at your terminal. When you enter a **`RETURN`** statement or an end-of-file character, execution continues with lines 4 and 5.

PAUSE

The **`PAUSE`** statement causes the program to stop executing input statements until a response is received from your terminal.

```
PAUSE

    [<c>]
```

Description

The character string associated with the **`PAUSE`** statement is ignored by the program, and serves only to document user input. When a **`PAUSE`** statement is executed, the prompt

```
Type <RETURN> to continue, or a command to be executed:
```

is issued at your terminal. The program then prompts you to enter a line of input. If you enter a TSUPREM-4 input statement, the statement is executed and another prompt is issued. If the line is blank, the processing of input statements continues in the normal manner.

You can use the **`PAUSE`** statement to interrupt statement processing temporarily to view graphics output. The **`INTERACTIVE`** statement can be used for the same purpose.

Example

Consider the following input file:

```
PLOT.2D SCALE GRID
PAUSE
PLOT.2D SCALE FLOW VLENG=.1
```

The program pauses between the two plots. It continues when you enter a blank line.

STOP

The **`STOP`** statement terminates the execution of the program. **`EXIT`** and **`QUIT`** are synonyms for the **`STOP`** statement.

```
STOP

    [<c>]
```

Description

Input statements following a **`STOP`** statement are not checked for syntax or executed. The character string associated with the **`STOP`** statement is ignored by the program and serves only to document the input. A **`STOP`** statement is not necessary to terminate program execution—an end-of-file condition on the input file has the same effect.

Example

Consider the input file:

```
$ LINE 1
$ LINE 2
STOP
THIS IS LINE 4; IT WILL NOT BE READ
```

Lines 1 and 2 are read and executed. When the **`STOP`** statement is executed, the program terminates, so the last line is not read or executed.

FOREACH/END

The **FOREACH** statement causes a group of input statements to be processed repeatedly. The **END** statement marks the end of a **FOREACH** loop.

```
FOREACH

    <name> <list>
```

Description

The **FOREACH** statement is used to specify loops in the input. The <*name*> takes on the values in the <*list*> consecutively until no values remain. The commands between the **FOREACH** statement and the matching **END** statement is executed once for each value in the <*list*>.

The <*list*> is a set of strings enclosed in parentheses and separated by commas or spaces. It can also take the form

(<*start*> **TO** <*end*> **STEP** <*increment*> **)**

where <*start*> is a numerical start value, <*end*> is the last value, and <*increment*> is the size of step to take between them. <*end*> must be greater than <*start*>, and <*increment*> must be greater than zero. The **STEP** and <*increment*> parameters can be omitted, in which case the <*increment*> defaults to one.

The <*name*> is set to the value in the <*list*> in a manner analogous to the **DEFINE** statement; the value is substituted for the <*name*> in the body of the loop accordingly.

Examples

- In the following code fragment, the `ECHO` statement is executed four times.

```
FOREACH STRING (antimony, arsenic, boron,
phosphorus )
   ECHO STRING
END
```

- The name *STRING* is set to the values “antimony,” “arsenic,” “boron,” and “phosphorus” consecutively. This produces the output:

```
antimony
arsenic
boron
phosphorus
```

- The following increments *VAL* from 1.0 to 10.0 in steps of 0.5. The inner body of the loop is executed 19 times.

```
FOREACH VAL ( 1.0 TO 10.0 STEP 0.5 )
   ECHO VAL
END
```

Notes

- If the value of <name> is changed with a **`DEFINE`** or **`UNDEFINE`** statement, the results are undefined.
- Substitution of **`DEFINE`**d names within a **`FOREACH`** loop does not occur in exactly the way one might expect. For details of the interactions between the **`FOREACH`** and **`DEFINE`** statements refer to DEFINE, p. 4-46.

LOOP/L.END

The **`LOOP`** statement begins an input statement loop and specifies the number of times necessary to process the statements within the loop. Optimization may be performed by a loop. The values of numerical and array parameters and assigned names may be varied on statements within loops. The **`L.END`** statement marks the end of a **`LOOP`**.

```
LOOP

    [STEPS=<c>] [INDEX=<c>] [START=<n>]]
    [ OPTIMIZE [DSSQ=<n>] [DNORM=<n>] [PLOT] ]
```

Parameter	Type	Definition
`STEPS`	number	The maximum number of passes through the loop. The statements between the **`LOOP`** statement and its matching **`L.END`** statement are processed once during each pass through the loop. The loop terminates when the number of passes equals the value of the **`STEPS`** parameter. If **`OPTIMIZE`** is specified, the loop also terminates when the optimization is successful. The value of the **`STEPS`** parameter must be a positive integer. **Units:** none **Default:** 50 for **`OPTIMIZE`**
`INDEX`	character	The name of the variable to store the number of a looping count which starts with 1 by default. The value is substituted for the name whenever the name appears in the input file preceded by the "@" character. **Default:** none
`START`	number	The starting number of the index variable specified by the **`INDEX`** parameter. **Default:** 1
`OPTIMIZE`	logical	Specifies that this loop performs optimization of values that are defined by **`ASSIGN`** statements specifying the **`LOWER`** and **`UPPER`** parameters. **Default:** false
`DSSQ`	number	The relative change in the sum of squares for convergence of optimization. This parameter can be used to change the criteria of convergence. **Units:** none **Default:** 1e-5
`DNORM`	number	The change in norm of parameter vector for convergence of optimization. This parameter can be used to change the criteria of convergence. **Units:** none **Default:** 2e-3
`PLOT`	logical	Specifies that a plot of optimization and sensitivity analysis results is to be drawn after optimization. The sensitivity graph is plotted only when a profile is specified as a target with the **`T.FILE`** parameter on an **`EXTRACT`** or **`ELECTRICAL`** statement. **Default:** false

Description

The **`LOOP`** statement defines the beginning of a sequence of statements which are processed repeatedly. An **`L.END`** statement is used to indicate the end of the statement sequence. The statement sequence is processed the number of times specified by the **`STEPS`** parameter or, if **`OPTIMIZE`** is specified, until the optimization is completed.

Each **`LOOP`** statement must be paired with a matching **`L.END`** statement. The nesting of the loop levels is not restricted (as long as system memory is available). The repeated processing of a statement sequence behaves in the same manner as the case where the sequence of statements is explicitly repeated multiple times. However, if optimization is performed, the state of the simulation is saved internally before the optimization starts and restored at the start of each iteration.

Termination of Optimization Looping

The termination of looping for optimization is determined, in order, by the following:

1. When the RMS error for each target is less than the **`TOLERANC`**, specified in the **`ELECTRICAL`** or **`EXTRACT`** statement.

   $\text{RMSerror}(\text{Target}_i) < \texttt{TOLERANC}_i$ for each target i

2. When the SSQ (Sum of SQuares) value of the targets increases or the decremental ratio of SSQ value for targets is less than **`DSSQ`**, and the SSQ value of parameter variations is less than **`DNORM`**.

   For j-th loop,
   $SSQ_{j-1} < SSQ_j$ or $(SSQ_{j-1} - SSQ_j) / SSQ_j < \texttt{DSSQ}$ for targets,
   and
   $SSQ_j < \texttt{DNORM}$ for parameters

3. When the looping number exceeds the maximum number, **`STEPS`**.

Parameter Sensitivity

The sensitivity of each parameter to each target is printed out after optimization finishes. The sensitivities show how much effect each parameter has on each target. The sensitivity is calculated with the normalized value of each parameter. The sensitivity is defined as:

$$Sensitivity(\%) = 100 \times \frac{\Delta \text{Target}}{\Delta P_{norm}}$$

in which ΔT is the change of target with respect to the change of normalized parameter, ΔP_{norm} (see ASSIGN, p. 4-23).

Dependence and Variability

Dependence and variability show how each parameter is expected to depend on the other parameters, while sensitivity shows the dependence of each target on each parameter. Dependence is an estimate of how much the sensitivity to each parameter is decreased if the other parameters are adjusted for a minimum rather than being held fixed. A zero value of dependence implies that the parameter is independent of the other parameters so that the optimized value of the parameter is unique. In general, small values of dependence (<10) are desirable, while large values (>100) imply that an accurate parameter cannot be extracted from the given data. The variability is the possible change of parameter for the same increase in error if the other parameters are simultaneously adjusted for minimum error.

Example

The output below shows the target `tox` depends mainly on the parameter `temp`, while both of the parameters, `temp` and `dose` contribute to the target `xj`.

```
Sensitivities:  100*(change in target)/(change in nor-
malized parameter)

 Parameter  Target Name
 Name       tox         xj
 --------   ----------  ---------
 temp          115.205     16.325
 dose            4.688     13.444
```

If the target is given as multiple data points rather than a specified value, for example, as SIMS data, the sensitivity can be shown in the plot. The **PLOT** parameter in the **LOOP** statement specifies that the sensitivity graph is plotted if necessary (Figure 4-1). The output shows two kinds of sensitivity: RMS sensitivity and maximum sensitivity.

```
Sensitivities:  100*(change in target)/(change in nor-
malized parameter)

Parameter   Target Name
Name        boron_sims boron_sims
----------  ---------- ----------
theta0      *   28.939 ^   46.885
seg0        *    4.797 ^   16.798

(*) means a RMS sensitivity.
(^) means a maximum sensitivity.
```

In Figure 4-1, the x-axis named `variable` represents the variable of target data. In this example, the variable is a depth in microns. The negative sensitivity means that the simulation result decreases in comparison with the target as the parameter value increases. The figure implies that the diffusivity of boron increases as the parameter `theta0` increases, and that the parameter `seg0` is dominant at the surface.

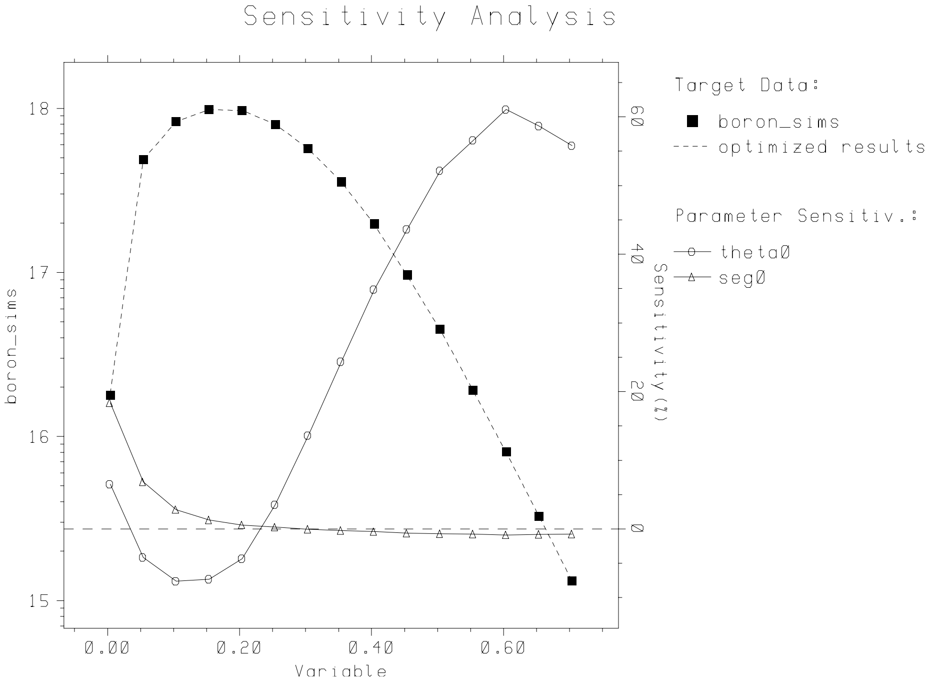


Figure 4-1 Example of sensitivity plot for target with multiple data

The **PLOT** parameter also specifies the plot of the optimization procedure, which shows how the parameters and the simulation results approach the optimized values and the desired values.

Advantages

The use of **LOOP** and **ASSIGN** statements instead of **FOREACH** provides some advantages as follows:

- Different assigned variables can be varied in unrelated ways for each pass through the loop.
- The number of passes through the loop is specified directly. (The number of passes through a **FOREACH** loop is sometimes less than expected due to round off error in computing the loop variable.)
- The **LOOP** statement allows optimization.
- The substitution of **ASSIGN**ed variables in a **LOOP** is more logical and intuitive than the substitution of **DEFINE**d names in a **FOREACH** loop.
- The **ASSIGN** statement evaluates numerical expressions before assigning them to variables.
- Assigned variables are only substituted when preceded by the “@” character. Thus, a statement or parameter name can never be inadvertently replaced by a variable specified by the **ASSIGN** statement.

L.MODIFY

The **`L.MODIFY`** statement modifies the processing of a currently active statement loop associated with a **`LOOP`** statement.

```
L.MODIFY

[STEPS=<n>] [ {NEXT | BREAK} ]
```

Parameter	Type	Definition
`STEPS`	number	The number of times the statements between the **`LOOP`** statement and its matching **`L.END`** statement are processed for the current loop level. The value of this parameter must be a positive integer. **Units:** none **Default:** the current value for the current loop level
`NEXT`	logical	Specifies that the next statement processed is the **`L.END`** statement for the current loop level. The statement between the **`L.MODIFY`** statement and **`L.END`** statement for the current loop level are not processed during this pass through the loop. **Default:** false **Synonym: `CONTINUE`**
`BREAK`	logical	Specifies that the next statement processed is the statement following the **`L.END`** statement for the current loop level. No subsequent passes through the loop is performed. The statements between the **`L.MODIFY`** statement and the **`L.END`** statement for the current loop level are not processed during this pass through the loop. **Default:** false

Description

The **`L.MODIFY`** statement can be used to modify the number of passes through a loop. Based on the results of previous statements, you may choose to modify the number of subsequent passes through the loop. An **`L.MODIFY`** statement specifying the **`STEPS`** statement can be used to increase or decrease the total number of passes through the loop.

IF/ELSEIF/ELSE/IF.END

The **IF** statement begins a sequence of one or more conditionally processed input statement blocks.

The **ELSEIF** statement terminates a conditionally processed input statement block associated with an **IF** or other previous **ELSEIF** statements and begins a new conditionally processed input statement block.

The **ELSE** statement specifies alternative action if the condition part of an **IF** and previous **ELSEIF** statements are false.

IF (*condition*)

[ELSEIF (*condition*)]

[ELSE]

IF.END

Description

The **IF** statement defines the beginning of a sequence of conditionally processed blocks of statements. An **IF.END** statement is used to indicate the end of the sequence of statement blocks. The first statement block in the sequence begins with the **IF** statement, while subsequent statement blocks begin with **ELSEIF** statements and finally with the **ELSE** statement.

At most, one statement block in a sequence of blocks is processed. The statement block processed is the first in the sequence with a true value for the condition on the **IF** or **ELSEIF** statement that begins the block. Only the statement block that begins with the **ELSE** statement in a sequence is processed if the **IF** statement and all **ELSEIF** statements in a sequence have a false value for the condition.

Each **IF** statement must be paired with a matching **IF.END** statement, with possibly intervening **ELSEIF** or **ELSE** statements. Pairs of **IF** and **IF.END** statements may be nested as deeply as system memory allows.

The **ELSEIF** statement defines the beginning of one statement block within a sequence of conditionally processed blocks of statements begun with an **IF** statement. The statement block is terminated by the following **ELSEIF** statement, by the **ELSE** statement, or by the **IF.END** statement that terminates the sequence of statement blocks. The statement block is processed if the value of condition is non-zero (i.e., true) and no previous statement blocks in the sequence have been processed.

The **ELSE** statement defines the beginning of the last statement block within a sequence of conditionally processed blocks of statements begun by an **IF** statement and **ELSEIF** statements. One and only one **ELSE** statement can be used within an **IF-IF.END** block.

Conditional Operators

The relational operators are:

> (*greater than*), **>=** (*greater than or equal to*), **<** (l*ess than*), **<=** (*less than or equal to*), **==** (*equal to*), **!=** (*not equal to*)

These operators all have the same precedence. Below them in precedence are the logical operators:

`&&` (*and*), | | (*or*)

Expressions connected by **`&&`** or | | are evaluated left to right.

Expression for Condition

The compared content can be a numerical expression or a character string. The following example shows the comparison between a number and a formula calculation,

```
ASSIGN NAME=X N.V=0.2 PROMPT="x? "
ASSIGN NAME=Y N.V=0.5 PROMPT="y? "
ASSIGN NAME=FUNC C.V=EXP PROMPT="Function? "
IF (@X*@FUNC(@Y)<10); ECHO x*@FUNC(y)<10; IF.END
```

The character string can be compared. In the expression of condition, the character string must be wrapped by the double quotation “ ”.

```
ASSIGN NAME=SHAPE C.V=none PROMPT=shape=
IF ("@SHAPE"=="triangle")

ECHO "3 sides"

ELSEIF ("@SHAPE”=="none")

ECHO "not specified"

ELSE

ECHO "invalid shape"

IF.END
```

Note:

There should be at least one blank between an `IF` (or `ELSEIF`) statement and the open parenthesis “(” of expression for condition.

ASSIGN

The **`ASSIGN`** statement assigns values to an assigned name.

```
ASSIGN

{ ( NAME=<c> [PRINT] [DELETE] [PROMPT=<c>]
    [{ ( {N.EXPRES=<n> | N.VALUE=<c>}
           [{ DELTA=<n> | RATIO=<n> | (LOWER=<n> UPPER=<n> [LOG]) }] )
     | C.VALUE=<c>
     | ( C.FILE=<c> [LINE=<n>] )
     | ( [C1=<c>] [C2=<c>] [C3=<c>] [C4=<c>] [C5=<c>]
           [C6=<c>] [C7=<c>] [C8=<c>] [C9=<c>] [C10=<c>] )
     }]
| ( ARRAY=<c>
    { ( IN.FILE=<c> DATA=<c> [{TIF | ROW | COLUMN}] )
      | IN.NVALU=<c> | IN.CVALU=<c> }
    [C.COUNT=<c>] )
}
```

Parameter	Type	Definition
`NAME`	character	The assigned name to which a value is being assigned or for which the current value is printed. **Default:** none
`PRINT`	logical	Specifies that the current value of the specified name is printed. **Default:** false
`DELETE`	logical	Specifies that the variable is deleted. **Default:** false
`PROMPT`	character	The character string used to prompt the user for interactive input of an alternative to the value specified by the **`N.VALUE`**, **`N.EXPRES`**, or **`C.VALUE`** parameter. If the character string read from the terminal input is blank, the value specified by the **`N.VALUE`**, **`N.EXPRES`**, or **`C.VALUE`** parameter is used. This parameter is only allowed with the **`N.VALUE`**, **`N.EXPRES`**, or **`C.VALUE`** parameters. **Default:** none
`N.EXPRES`	number	The numerical value assigned to the assigned name. If neither **`LOWER`** nor **`UPPER`** are specified, the **`DELTA`** or **`RATIO`** parameters can be specified to vary the value of the assigned name with each iteration of the enclosing **`LOOP`** statement. **Units:** none **Default:** none

Parameter	Type	Definition
`N.VALUE`	character	The numerical values assigned to the assigned name. If a single value is specified and neither **`LOWER`** nor **`UPPER`** are specified, the **`DELTA`** or **`RATIO`** parameters can be specified to vary the value of the assigned name with each iteration of the enclosing **`LOOP`** statement. If multiple values are specified, the value of the assigned name is varied by choosing successive values from the list of values specified with this parameter. After the last value in the list is taken, the sequence begins again with the first value in the list. Only a single value may be specified if **`LOWER`** and **`UPPER`** are specified. **Default:** none **Synonyms: `N.ARRAY`, `VALUE`**
`DELTA`	number	The constant difference by which the value of the assigned name is varied. This parameter is only allowed if a single value is specified with the **`N.VALUE`** parameter and neither **`LOWER`** nor **`UPPER`** is specified. This parameter works only within a **`LOOP/L.END`** block that does not specify optimization. **Units:** none **Default:** none
`RATIO`	number	The constant ratio by which the value of the assigned name is varied. The value of this parameter must be nonzero. This parameter is only allowed if a single value is specified with the **`N.VALUE`** parameter and neither **`LOWER`** nor **`UPPER`** is specified. This parameter works only within a **`LOOP/L.END`** block that does not specify optimization. **Units:** none **Default:** none
`LOWER`	number	The lower bound for the value of the assigned name during optimization. If the **`LOWER`** value is greater than the **`UPPER`** value, the **`LOWER`** value is automatically exchanged with the **`UPPER`** value. **Units:** none **Default:** none
`UPPER`	number	The upper bound for the value of the assigned name during optimization. If the **`UPPER`** value is less than the **`LOWER`** value, the **`UPPER`** value is automatically exchanged with the **`LOWER`** value. **Units:** none **Default:** none
`LOG`	logical	Specifies that the value assigned to the **`NAME`** is to be varied on a logarithmic scale during optimization. **Default:** false
`C.VALUE`	character	The character value assigned to the assigned name. **Default:** none
`C.FILE`	character	The name of the file including the character string to be read. **Default:** none
`LINE`	number	The line number of the character string to be read from the file **`C.FILE`**. **Units:** none **Default:** none

Parameter	Type	Definition
`C1`	character	The first in a list of character values assigned to the assigned name. The value of the assigned name is varied by choosing successive values from the list of values specified with the parameters `C1` through `C10`. If the looping step exceeds the maximum parameter, the assigned character value is circulated from `C1` again. This parameter works only within a **`LOOP/L.END`** block that does not specify optimization. **Default:** none
`C2`	character	The second in a list of character values assigned to the assigned name. The value of the assigned name is varied by choosing successive values from the list of values specified with the parameters `C1` through `C10`. This parameter works only within a **`LOOP/L.END`** block that does not specify optimization. **Default:** none
`C3`	character	The third in a list of character values assigned to the assigned name. The value of the assigned name is varied by choosing successive values from the list of values specified with the parameters `C1` through `C10`. This parameter works only within a **`LOOP/L.END`** block that does not specify optimization. **Default:** none
`C4`	character	The fourth in a list of character values assigned to the assigned name. The value of the assigned name is varied by choosing successive values from the list of values specified with the parameters `C1` through `C10`. This parameter works only within a **`LOOP/L.END`** block that does not specify optimization. **Default:** none
`C5`	character	The fifth in a list of character values assigned to the assigned name. The value of the assigned name is varied by choosing successive values from the list of values specified with the parameters `C1` through `C10`. This parameter works only within a **`LOOP/L.END`** block that does not specify optimization. **Default:** none
`C6`	character	The sixth in a list of character values assigned to the assigned name. The value of the assigned name is varied by choosing successive values from the list of values specified with the parameters `C1` through `C10`. This parameter works only within a **`LOOP/L.END`** block that does not specify optimization. **Default:** none
`C7`	character	The seventh in a list of character values assigned to the assigned name. The value of the assigned name is varied by choosing successive values from the list of values specified with the parameters `C1` through `C10`. This parameter works only within a **`LOOP/L.END`** block that does not specify optimization. **Default:** none
`C8`	character	The eighth in a list of character values assigned to the assigned name. The value of the assigned name is varied by choosing successive values from the list of values specified with the parameters `C1` through `C10`. This parameter works only within a **`LOOP/L.END`** block that does not specify optimization. **Default:** none

Parameter	Type	Definition
`C9`	character	The ninth in a list of character values assigned to the assigned name. The value of the assigned name is varied by choosing successive values from the list of values specified with the parameters **`C1`** through **`C10`**. This parameter works only within a **`LOOP/L.END`** block that does not specify optimization. **Default:** none
`C10`	character	The tenth in a list of character values assigned to the assigned name. The value of the assigned name is varied by choosing successive values from the list of values specified with the parameters **`C1`** through **`C10`**. This parameter works only within a **`LOOP/L.END`** block that does not specify optimization. **Default:** none
`ARRAY`	character	The prefix name for an array to store the sequential data from the external data file. The index of an array starts with 1. **Default:** none
`IN.FILE`	character	The identifier for the file containing the data to be stored to the **`ARRAY`** parameter. The file format must be either a TIF or a columnwise format. **Default:** none
`DATA`	character	The name of the data in a TIF file or the column number of the data in a columnwise file to be stored to the **`ARRAY`** parameter. **Default:** none
`TIF`	logical	Specifies that the **`IN.FILE`** is a TIF file. **Default**: false
`ROW`	logical	Specifies that the **`IN.FILE`** is a rowwise file. **Default**: false
`COLUMN`	logical	Specifies that the **`IN.FILE`** is a columnwise file. **Default**: true
`IN.NVALU`	character	The string composed of numbers, for example, “1.2 3 4.5 6.” **Default:** none
`IN.CVALU`	character	The string composed of character strings, for example, “a bc def g.” **Default:** none
`C.COUNT`	character	The name of the variable to which the number of the stored data is assigned. **Default:** none

Description

The **`ASSIGN`** statement associates a value with a name. The value is substituted for the name whenever the name appears in the input file preceded by the “@” character. The name must start with a letter and may contain only letters and numbers.

The **ASSIGN** statement performs a function similar to that of the **DEFINE** statement, but in a way that is more convenient for many applications. Some of its advantages are:

- An **ASSIGN**ed name can take on a different value for each iteration of a **LOOP** statement. Different names can be varied in unrelated ways during a single loop.
- The syntax for using **ASSIGN**ed names is simpler than for **DEFINE**d names, leaving less chance for confusion or error.
- The **ASSIGN** statement works in the same way in TSUPREM-4 as it does in Medici and Davinci.

Varying During Statement Looping

The **ASSIGN**ed variable may be varied during statement looping by one of the following:

- A constant increment, by specifying the **DELTA** parameters:

```
LOOP STEPS=3
  ASSIGN  NAME=X N.VAL=10 DELTA=10 PRINT
L.END
```

  The variable X takes the values 10, 20, and 30.

- A constant ratio, by specifying the **RATIO** parameter:

```
LOOP STEPS=3
  ASSIGN  NAME=X N.VAL=10 RATIO=10 PRINT
L.END
```

  The variable X takes the values 10, 100, and 1000.

- A list of values from which successive values are taken during each pass through a statement loop:

```
LOOP STEPS=3
  ASSIGN  NAME=X N.VAL="20, 10, 90" PRINT
L.END
```

  The variable X takes the values 20, 10, and 90.

ASSIGN with Mathematical Expressions

The assigned value may be specified with another variable and/or a mathematical expression as well as a constant. For example,

```
ASSIGN NAME=X  N.VAL=0.2
ASSIGN NAME=Y  N.VAL=ERFC(@X) PRINT
LOOP STEPS=2
  ASSIGN NAME=Z N.VAL="@X+@Y, LOG10(@X*@Y)"
L.END
```

Note:

The `N.VALUE` (`N.ARRAY`) is handled as a character string; the separator of an `N.VALUE` is a blank or a comma character. Thus, if there is a blank character in the specified formula to `N.VALUE`, for example, `N.VALUE=( @X + @Y )` may not be accepted. Instead, use the `N.EXPRES`. Since the `N.EXPRES` is not a character string but a number, the blank character can be accepted as long as the formula is wrapped by parentheses. If the formula is not wrapped by parentheses, even `N.EXPRES` disallows a blank character.

ASSIGN and Optimization

- The **ASSIGN**ed variable specifies a parameter to be optimized when both **LOWER** and **UPPER** parameters are specified. The initial value, with which the optimization starts, is specified with the **N.VALUE** parameter. The optimization loop determines an appropriate value for each assigned name for each pass through the loop after the first. The range of allowed values for the assigned name is specified with the **LOWER** and **UPPER** parameters.
- The sensitivity shown after optimization is based on the value of normalized parameters in order to eliminate the unit dependency. The normalization is performed as follows:

$$Param,_{norm} = \frac{Param\texttt{-LOWER}}{\texttt{UPPER-LOWER}}$$

Thus, the reported sensitivity value depends on the values of **LOWER** and **UPPER**.

Expansion of ASSIGNed Variable

The combination of assigned variables is expanded from left to right. For example,

```
ASSIGN NAME=A C.V=T
ASSIGN NAME=B C.V=M
ASSIGN NAME=C C.V=A
ASSIGN NAME=D C.V=@A@B@C PRINT
```

The variable, D, is assigned with the character string, TMA.

```
ASSIGN NAME=EF C.V=BEST
ASSIGN NAME=GH C.V=F
$ 1)
ASSIGN NAME=IJ C.V=@E@GH PRINT

ASSIGN NAME=E C.V=GOL
$ 2)
ASSIGN NAME=F C.V=@E@GH PRINT
```

At 1), E is an unknown variable and, thus, is not expanded. @GH is expanded to the character, F. And then, @EF is expanded to the string BEST. However, at 2), E is not an unknown variable any longer so that @E is expanded to the string, GOL. Finally, the variable, F, is assigned with the string, GOLF, to which GOL@GH has been expanded.

Reading the External Data File

The data in a TIF file or a rowwise or a columnwise file can be read through the **`ASSIGN`** statement. The data is sequentially read into the variable **`ARRAY`** to be appended with the index of which the starting number is 1. The parameter **`DATA`** specifies which data is to be read, the name of the data for a TIF file, the row number of data for a rowwise file, or the column number of data for a columnwise file. The variable defined by **`C.COUNT`** stores the number of data to be read.

```
ASSIGN ARRAY=VG IN.FILE=VGID.IVL TIF DATA=V(GATE)+
       C.COUNT=NDATA
ASSIGN ARRAY=ID IN.FILE=VGID.IVL TIF DATA=I(DRAIN)
LOOP STEPS=@NDATA

  ASSIGN NAME=i N.V=1 DELTA=1
  EXTRACT PREFIX="@VG@i @ID@i"

L.END
```

If the number of data is 12, the value of **`NDATA`** is 12, and each variable of VG*n* and ID*n*, in which *n* has the number from 1 to 12, stores the V(GATE) and I(DRAIN), respectively. The example shows how to read the *.ivl* (TIF format) file written by Medici. It prints out the data, V(GATE) vs. I(DRAIN), in a *vgid.ivl* file.

Note:

> ***If there is already a variable with the same name as the*** **`ARRAY`*, the previous value of the variable is overridden. In the above example, if the variable, for example,*** VG1***, already exists before reading the data file, the current value of*** VG1 ***is lost and replaced with the data read.***

Also the character string can be read from an external file.

```
ASSIGN NAME=str C.FILE=text.dat LINE=10
```

The variable *str* stores the string of 10-th line in the file *text.dat*.

Reading the Array from a String

The array can be read from a data string composed of either numbers or characters.

```
ASSIGN ARRAY=vg IN.NVALU=”0.05 1.0” C.COUNT=nvgs
ASSIGN ARRAY=type IN.CVALU=”linear saturat”
```

The numbers of the variables *vg1* and *vg2* are 0.05 and 1.0, respectively. The strings of the variables *type1* and *type2* are “linear’ and “saturat,” respectively. The variable *nvgs* stores the number 2 of the count of the array.

INTERMEDIATE

The **`INTERMEDIATE`** statement specifies the intermediate quantities associated with equations. Aliases for the **`INTERMEDIATE`** statement are **`MODEL`** (see Chapter 7, INTERMEDIATE, p. 7-8) or **`ARRAY`**..

```
INTERMEDIATE

   {  SCOPE=<c>
    | ENDSCOPE
    | { NAME=<c> [EXPRESS=<c>] [SIZE=<c> [MININDEX=<c>][DEFAULT=<c>]]
        [{ PRINT | DELETE }] [{ FIXNOW | UNFIXNOW }]
   }
```

Parameter	Type	Definition
`SCOPE`	character	The name of the scope which defines the start of the block that must end by the parameter **`ENDSCOPE`**. All intermediates within the block belong to the **`SCOPE`**. **Default:** none
`ENDSCOPE`	logical	Specifies to close the block defined by the **`SCOPE`**. **Default:** false
`NAME`	character	The intermediate name. **Default:** none
`EXPRESS`	character	The expression for the intermediate named by the **`NAME`** parameter. The **`EXPRESS`** parameter can specify a single expression or multiple expressions separated by commas (,). Multiple expressions are allowed only when the **`SIZE`** parameter is specified. **Default:** none **Synonym**: **`VALUE`**
`SIZE`	character	The array size. The **`SIZE`** specification implies that the intermediate is an array, the size of which is specified by the **`SIZE`** parameter. **Default:** none **Synonym**: **`LENGTH`**
`MININDEX`	character	The array minimum index number **Default:** 0
`DEFAULT`	character	The default initial expression for array elements. Only one expression is allowed. The **`DEFAULT`** parameter may not be specified together with the **`EXPRESS`** parameter. **Default:** none
`PRINT`	logical	Specifies that the values of intermediates are to be printed. **Default:** false
`DELETE`	logical	Specifies that the intermediate is to be removed. **Default:** false

Parameter	Type	Definition
`FIXNOW`	logical	Specifies to fix the current values of the intermediates. **Default:** false
`UNFIXNOW`	logical	Specifies to unfix the fixed intermediate. **Default:** false

Advantages of INTERMEDIATES

The advantages of using intermediates quantities are:

- Simplifying long and complicated equations
- Avoiding the duplicated calculation of quantities used in several equations

  Once the intermediates are evaluated, the calculated values are shared by all associated equations until the intermediates need to be reset
- Avoiding duplicated calculation of quantities depending *only* on temperature

  If the intermediates depend *only* on temperature, they are evaluated *only* when the temperature changes.
- Using intermediates in the same manner as the built-in solution values in the `SELECT` statement.

Note:

Intermediates are stored as symbolic expressions that are evaluated later (e.g., when solving the model equations).

Scope Definition

The parameters `SCOPE` and `ENDSCOPE` specifies the scope range within which the intermediates are applied. The scopes can be hierachically nested.

```
INTERMED NAME=A1 EXPR=10
INTERMED SCOPE=family
  INTERMED NAME=A1 EXPR=20
  INTERMED SCOPE=genus1
    INTERMED NAME=A1 EXPR=30
    INTERMED NAME=A1 PRINT  ;$(1)
  INTERMED ENDSCOPE
  INTERMED NAME=A1 PRINT    ;$(2)
  INTERMED SCOPE=genus2
    INTERMED NAME=A1 EXPR=35
  INTERMED ENDSCOPE
INTERMED ENDSCOPE
```

The (1) command prints 30 while the (2) prints 20.

The intermediates in the higher or same level scopes can be accessed from the lower level by specifying the intermediate name followed by the full path of scopes. ‘#’ is the character to separate the intermediate name and its scope path. The path string following ‘#’ always starts with ‘.’ which denote the top level.

Then the scope names from the top level to the level in which the intermediate is defined are concatenated by the character ‘.’.

In the above example,

```
INTERMED NAME=A1 EXPR=10
INTERMED SCOPE=family
  INTERMED NAME=A1 EXPR=20
  INTERMED SCOPE=genus1
    INTERMED NAME=A1 EXPR=30
    INTERMED NAME=A1
    INTERMED NAME=A1#.        PRINT ;$(1)
    INTERMED NAME=A1#.family PRINT ;$(2)
  INTERMED ENDSCOPE
  INTERMED NAME=A1 PRINT
  INTERMED SCOPE=genus2
    INTERMED NAME=A1 EXPR=35
    INTERMED NAME=A1#.family.genus1 PRINT ;$(3)
  INTERMED ENDSCOPE
INTERMED ENDSCOPE
INTERMED PRINT ;$(4)
```

(1) prints A1 value in the top level, i.e. 10. (2) prints 20 in the scope ‘*family*’, one level lower than the top. (3) prints 30 which is defined in the scope ‘*genus1*’, the neighbor scope ‘*genus2*’. (4) prints all intermediates. The full path of each intermediate is written in ‘Scope’ column. If the full path name is longer than 10 characters, it is truncated as shown in ‘*A1#.family.genus1*’ and ‘*A1#.family.genus2*’.

```
---------- ---------- --------- ------------------------------------- ----------
 Intermed     Scope      Type                 Expression                  Value
---------- ---------- --------- ------------------------------------- ----------
A1         .          CONSTANT  10                                            10
A1         .family    CONSTANT  20                                            20
A1         .family.ge CONSTANT  30                                            30
A1         .family.ge CONSTANT  35                                            35
---------- ---------- --------- ------------------------------------- ----------
```

The following example shows the case using the array type. (See the details for the array type in Array Values, p. 4-34)

```
INTERMED NAME=R1 SIZE=2 EXPR=”10,20”
INTERMED SCOPE=lev1
  INTERMED NAME=R1 SIZE=3 EXPR=”30,40,R1#.[1]”
  INTERMED SCOPE=lev2a
   INTERMED NAME=R1 SIZE=2 EXPR=”R1#.[0], R1#.lev1[0]”
  INTERMED ENDSCOPE
  INTERMED SCOPE=lev2b
    INTERMED NAME=R1 SIZE=2 +
             EXPR=”R1#.lev1.lev2a[1], 50”
  INTERMED ENDSCOPE
INTERMED ENDSCOPE
INTERMED PRINT  ;$ (1)
```

The array intermediates in higher or neighbor scopes can be accessed by appending the full path to the array name. (1) prints as follows:

```
Array Name : R1,   Scope: .
---------- --------- ------------------------------------------------ ----------
  Index      Type                      Expression                       Value
---------- --------- ------------------------------------------------ ----------
[0]        CONSTANT  10                                                       10
[1]        CONSTANT  20                                                       20
---------- --------- ------------------------------------------------ ----------

Array Name : R1,   Scope: .lev1
---------- --------- ------------------------------------------------ ----------
  Index      Type                      Expression                       Value
---------- --------- ------------------------------------------------ ----------
[0]        CONSTANT  30                                                       30
[1]        CONSTANT  40                                                       40
[2]        CONSTANT  R1#.[1]                                                  20
---------- --------- ------------------------------------------------ ----------

Array Name : R1,   Scope: .lev1.lev2a
---------- --------- ------------------------------------------------ ----------
  Index      Type                      Expression                       Value
---------- --------- ------------------------------------------------ ----------
[0]        CONSTANT  R1#.[0]                                                  10
[1]        CONSTANT  R1#.lev1[0]                                              30
---------- --------- ------------------------------------------------ ----------

Array Name : R1,   Scope: .lev1.lev2b
---------- --------- ------------------------------------------------ ----------
  Index      Type                      Expression                       Value
---------- --------- ------------------------------------------------ ----------
[0]        CONSTANT  R1#.lev1.lev2a[1]                                        30
[1]        CONSTANT  50                                                       50
---------- --------- ------------------------------------------------ ----------
```

Value Type

When the expression is described by the **EXPRESS** parameter, the value type is automatically determined to be one of the following:

- `CONSTANT`
  The value is a constant number. It is recalculated whenever any other constant intermediate is redefined.

```
INTERMED NAME=C1 EXPRESS=1.0       ;$1
INTERMED NAME=C2 EXPRESS=C1        ;$2
INTERMED NAME=C1 EXPRESS=2.0       ;$3
```

In the third line, the *C2* value is changed from 1.0 to 2.0.

- `PARAMETER`
  The value is dependent *only* on temperature. The value will be recalculated whenever the temperature changes.

```
INTERMED NAME=P1 EXPRESS=1E-5*exp(-2.0/kt) ;$1
INTERMED NAME=P2 EXPRESS=P1                ;$2
```

The values of the *P1* and *P2* intermediates are always the same. The value type of the intermediate *P2* is `PARAMETER`.

- `SOLUTION`
  The value is a node solution or an edge solution. The *node solution* type specifies that the quantity varies on nodes. The *edge solution* type represents the quantity evaluated on the edges between nodes.

```
INTERMED NAME=S1 EXPRESS=BORON+PHOS  ;$1
INTERMED NAME=S2 EXPRESS=MIN(S1,1E21);$2
```

The value type of the intermediate *S2* is `SOLUTION`.

In case the type of an intermediate is changed by reassignment, the value types of the others associated with the intermediate are also reviewed and changed, if needed.

```
INTERMED NAME=P1 EXPRESS=1E-5*exp(-2.0/kt) ;$1
INTERMED NAME=P2 EXPRESS=P1                ;$2
INTERMED NAME=P1 EXPRESS=2.0               ;$3
```

In the third line, the *P2* value type is changed to `CONSTANT`.

Array Values

The **SIZE** parameter specifies that the intermediate is an array as well as its array size. The values for the array elements are provided by the string. In the string, each element is separated by a comma (,) in the **EXPRESS** parameter. When the **DEFAULT** parameter is specified instead of the **EXPRESS** parameter, all elements are initialized to the **DEFAULT** parameter. The **DEFAULT** parameter allows only a single expression. The expression type can be `CONSTANT`, `PARAMETER`, and also `SOLUTION`. If the number of data items listed in the **EXPRESS** parameter is smaller than the **SIZE** parameter, the undefined array elements are assigned with `N/A` values. Conversely, if the number of data items listed are greater than the **SIZE** parameter, all the data items in the list beyond the **SIZE**th item are ignored. You can also define the starting value of array indices with the **MININDEX** parameter. The default value of the **MININDEX** parameter is 0. Each array element can have a different value type.

```
INTERMED NAME=a SIZE=3 MININDEX=-1 +
         EXPRESS="1,BORON+PHOS,exp(-1.0/KT)" PRINT
```

The array size 3 and the minimum index -1 means that the first array element is *a[-1]* and the last one is *a[1]*. The array elements are assigned with the different value types `CONSTANT`, `SOLUTION`, and `PARAMETER`. The output shows:

```
Array Name : a,   Scope: .
---------- --------- -------------------------------- ----------
  Index      Type               Expression               Value
---------- --------- -------------------------------- ----------
[-1]       CONSTANT  1                                         1
[0]        SOLUTION  BORON+PHOS
[1]        PARAMETER exp(-1.0/KT)                       2.01e-05
---------- --------- -------------------------------- ----------
```

Multi-dimensional arrays are defined in a similar way to single arrays. The array dimension is determined by the number of sizes listed in the **SIZE** parameter. Each number listed corresponds to the size of each dimension. Similarly, the minimum numbers of each index are specified with the string specified by the

MININDEX parameter.

```
INTERMED NAME=b SIZE="2,3" MININDEX="0,-1" +
        EXPRESS="1,2,3,SUM(i,1,10,i^2),KT,BORON" PRINT
```

There are two sizes listed in the **SIZE** parameter, 2 and 3. That means it is a 2D array. The size of the first dimension is 2 and its index starts from 0. The size of the second dimension is 3, and its minimum index is -1. Thus, the first array element is *b[0][-1]* and the last one is *b[1][1]*. The output shows:

```
Array Name : b,   Scope: .
---------- --------- ------------------------------- ----------
  Index      Type               Expression              Value
---------- --------- ------------------------------- ----------
[0][-1]    CONSTANT  1                                        1
[0][0]     CONSTANT  2                                        2
[0][1]     CONSTANT  3                                        3
[1][-1]    CONSTANT  SUM(i,1,10,i^2)                        385
[1][0]     PARAMETER KT                                  0.0925
[1][1]     SOLUTION  BORON
---------- --------- ------------------------------- ----------
```

Getting Array Values

Use the bracket symbol (**[]**) to get the value of an array element.

```
INTERMED NAME=a SIZE=2 EXPRESS="10,20"             ;$1
INTERMED NAME=b EXPRESS=a[1]                       ;$2
INTERMED NAME=c SIZE=3 EXPRESS="a[0],b,30"         ;$3
INTERMED NAME=c PRINT
```

The array element can be the other array element as well as an intermediate, as shown in the third line. The above example prints out as:

```
Array Name : c,   Scope: .
---------- --------- ------------------------------- ----------
  Index      Type               Expression              Value
---------- --------- ------------------------------- ----------
[0]        CONSTANT  a[0]                                    10
[1]        CONSTANT  b                                       20
[2]        CONSTANT  30                                      30
---------- --------- ------------------------------- -----------
```

Redefinition of Array Elements

Redefine array elements in the same manner as non-array intermediates.

```
INTERMED NAME=a SIZE="3,2" +
         EXPRESS="1,2,temp,4,arsenic,0"
INTERMED NAME=a[0][1] EXPRESS=boron
```

The *a[0][1]* is changed to the solution variable *boron* from the constant 2.

Effect Due to Redefinition

Changing the value type of an array element affects the types of the other intermediates.

```
INTERMED NAME=a SIZE=”3,2” +
         EXPR=”1,2,temp,4,arsenic,0”                   ;$1
INTERMED NAME=b EXPR=sum(i,0,0,sum(j,0,1,a[i][j])) ;$2
INTERMED NAME=c EXPR=sum(i,0,1,sum(j,0,1,a[i][j])) ;$3
INTERMED NAME=d EXPR=sum(i,1,2,sum(j,0,1,a[i][j])) ;$4
INTERMED PRINT                                         ;$5
INTERMED NAME=a[0][1] EXPR=boron                       ;$6
INTERMED PRINT                                         ;$7
```

In the fifth line, the output shows:

```
---------- ---------- --------- ------------------------------------ ----------
Intermed     Scope      Type                 Expression                Value
---------- ---------- --------- ------------------------------------ ----------
b          .          CONSTANT  sum(i,0,0,sum(j,0,1,a[i][j]))                 3
c          .          PARAMETER sum(i,0,1,sum(j,0,1,a[i][j]))          1.08e+03
d          .          SOLUTION  sum(i,1,2,sum(j,0,1,a[i][j]))
---------- ---------- --------- ------------------------------------ ----------
```

At line six, the value of the array element *a[0][1]* is changed to a solution variable, *boron* from a constant of 1. It implies that the value type of *a[0][1]* is changed to `SOLUTION` from `CONSTANT`. Thus, it affects the value types of the intermediates *b* and *c* of which the expressions include *a[0][1]*. In the seventh line, it prints out:

```
---------- ---------- --------- ------------------------------------ ----------
Intermed     Scope      Type                 Expression                Value
---------- ---------- --------- ------------------------------------ ----------
b          .          SOLUTION  sum(i,0,0,sum(j,0,1,a[i][j]))
c          .          SOLUTION  sum(i,0,1,sum(j,0,1,a[i][j]))
d          .          SOLUTION  sum(i,1,2,sum(j,0,1,a[i][j]))
---------- ---------- --------- ------------------------------------ ----------
```

Comparison to ASSIGN

The **ASSIGN** statement assigns numeric or string values. The values assigned by the **ASSIGN** statement are never changed unless they are respecified by **ASSIGN**. The value assigned by the **ASSIGN** statement is substituted for the name preceded by the ‘@’ character. The differences between the **INTERMED** and **ASSIGN** statements are as follows:

Differences	**INTERMED**	**ASSIGN**
Numeric value type	**INTERMED NAME=a EXPRESS=exp(2)**	**ASSIGN NAME=a N.VALUE=exp(2)**
String value type	Not available	**ASSIGN NAME=a C.VALUE=”TS4”**
Solution value type	**INTERMED NAME=a EXPRESS=boron**	Not available
Array definition	**INTERMED NAME=b +** **EXPRESS=”4,5” SIZE=2**	**ASSIGN ARRAY=b +** **IN.NVALU=”4,5”**
Array dimension	Multi-dimension allowed	one dimension only
Default minimum index of arrays	0	1

Differences	INTERMED	ASSIGN
Getting values	`a` `b[1]`	`@a` `@b1`

Evaluation of expressions	Expression are saved and evaluated only when necessary
	`INTERMED NAME=a EXPRESS=1`
	`INTERMED NAME=b EXPRESS=a`
	`INTERMED NAME=a EXPRESS=2`
	`INTERMED NAME=b PRINT ;$ b=2`
	
	`SELECT TEMP=800`
	`INTERMED NAME=c EXPRESS=temp`
	`SELECT TEMP=900`
	`INTERMED NAME=c PRINT;$ c=1173`
		Expressions are evaluated immediately
		`ASSIGN NAME=a N.VALUE=1`
		`ASSIGN NAME=b N.VALUE=@a`
		`ASSIGN NAME=a N.VALUE=2`
		`ASSIGN NAME=b PRINT ;$ b=1`
		
		`SELECT TEMP=800`
		`ASSIGN NAME=c N.VALUE=temp`
		`SELECT TEMP=900`
		`ASSIGN NAME=c PRINT;$ c=1073`

Snapshot

The **`INTERMED`** expression is evaluated by call-by-reference. For example,

```
INTERMED NAME=A1 EXPR=10
INTERMED NAME=A2 EXPR=A1 PRINT ;$(1)
INTERMED NAME=A1 EXPR=20
INTERMED NAME=A2 PRINT         ;$(2)
```

At (1), A2 stores the memory address of A1 instead of the value of A1, 10. Thus (2) shows that A2 value is 20 because the value of A1 is changed to 20.

The parameter **`FIXNOW`** snapshots and fixes the current values. On the other hand, the parameter **`UNFIXNOW`** unfixes the fixed **`INTERMED`**. For example,

```
INTERMED NAME=A1 EXPR=10

$ The following statement is same as:
$ INTERMED NAME=A2 EXPR=A1
$ INTERMED NAME=A2 FIXNOW
INTERMED NAME=A2 EXPR=A1 FIXNOW

INTERMED NAME=A1 EXPR=20
INTERMED NAME=A2 PRINT          ;$(1)
INTERMED NAME=A2 UNFIXNOW PRINT ;$(2)
```

The command (1) and (2) print 10 and 20 respectively. As shown in the following output of the commands (1) and (2), the name of a fixed intermediate is followed by the character ‘*’;

```
INTERMED NAME=A2 PRINT
---------- ---------- --------- ------------------------------------ ----------
 Intermed     Scope     Type                 Expression                 Value
---------- ---------- --------- ------------------------------------ ----------
A2*        .          CONSTANT  A1                                           10
---------- ---------- --------- ------------------------------------ ----------

INTERMED NAME=A2 UNFIXNOW PRINT
---------- ---------- --------- ------------------------------------ ----------
 Intermed     Scope     Type                 Expression                 Value
---------- ---------- --------- ------------------------------------ ----------
A2         .          CONSTANT  A1                                           20
---------- ---------- --------- ------------------------------------ ----------
```

The following example shows how to compare as-implant and post-anneal profiles. The field name `DIMPLANT` stores the as-implant dopant concentration. The *BIMPLANT* and *PIMPLANT* intermediates store the boron and phosphorus as-implant profiles respectively.

```
INTERMED NAME=BIMPLANT EXPR=DIMPLANT
INTERMED NAME=PIMPLANT EXPR=DIMPLANT

IMPLANT BORON DOSE=1E15 ENERGY=10
$ Snapshot boron as-implant profile
INTERMED NAME=BIMPLANT FIXNOW

IMPLANT PHOSPHOR DOSE=2E14 ENERGY=30
$ Snapshot phosphorus as-implant profile
INTERMED NAME=PIMPLANT FIXNOW
DIFFUSE TEMP=900 TIME=10

$ PLOT COMPARISON BEFORE/AFTER DIFFUSION
SELECT Z=LOG10(BIMPLANT)
PLOT.1D X.V=0 TOP=22 BOT=15 RIGHT=0.5 COLOR=2
SELECT Z=LOG10(BORON)
PLOT.1D X.V=0 !CL !AX COLOR=2 LINE=2
SELECT Z=LOG10(PIMPLANT)
PLOT.1D X.V=0 !CL !AX COLOR=4 LINE=2
SELECT Z=LOG10(PHOSPHOR)
PLOT.1D X.V=0 !CL !AX COLOR=4 LINE=2
```

Load/Save Intermediates

Using **INTERMED** is the convenient method to manipulate solution data. By saving the expressions and calculated values of intermediates to TIF or TS4 structure format files, you can use the intermediates continuously after loading them. You also can plot the intermediate values in visualizer tools such as Taurus Visual.

The intermediates are saved when the parameter **`INTERMED`** is turned on in the **`SAVEFILE`** command.

```
SAVEFILE OUT.FILE=test.tif TIF INTERMED ;$ TIF format
SAVEFILE OUT.FILE=test.sav     INTERMED ;$ TS4 format
```

The intermediates are not loaded when the parameter **`INTERMED`** is turned off in the **`INITIALIZE`** or **`LOADFILE`** commands.

```
INITIALIZE IN.FILE=test.tif TIF ^INTERMED
INITIALIZE IN.FILE=test.sav ^INTERMED
LOADFILE   IN.FILE=test.tif TIF ^INTERMED
LOADFILE   IN.FILE=test.sav ^INTERMED
```

Example To save the equilibrium concentration and supersaturation of interstitial:

```
INTERMED NAME=ISTAR EXPR=CI.STAR
INTERMED NAME=IRATIO EXPR=CI.RATIO
METHOD PD.FULL
MESH DY.SURF=0.001 LY.SURF=0.001 DY.ACT=0.05
LY.ACT=0.5
INIT
DEPOSIT OXIDE THICKNES=0.001
IMPLANT BORON DOSE=1E15 ENERGY=10
DIFFUSE TEMP=700 TIME=0.1/60
SAVEFILE OUT.FILE=test.tif TIF INTERMED
```

ECHO

The **ECHO** statement prints its parameters, evaluating arithmetic expressions.

```
ECHO
   <string>
```

Description

The **ECHO** statement prints the given <*string*> on the standard output. Defined abbreviations are substituted into the string. If the result is a valid arithmetic expression, it is evaluated. (See COMMENT, p. 4-7 and SELECT, p. 4-146.) Otherwise, the resulting string value is printed.

Examples

- The statements:

  ```
  DEFINE W 2.0
  ECHO The width is W - 0.5
  ECHO W - 0.5
  ```

  print the two lines:

  ```
  The width is 2.0 - 0.5
  1.5
  ```

  to your terminal and to the standard output file. The first **ECHO** statement prints a string because its argument is not a valid arithmetic expression. The second **ECHO** statement shows evaluation of a valid expression.

- The statement:

  ```
  ECHO ( 15.0 - 12.0 * exp( 4.0 - 2.0 / 6.0 ) )
  ```

  prints:

  ```
  -454.455
  ```

  which is the value of the arithmetic expression.

OPTION

The `OPTION` statement sets terminal and plotting options.

```
OPTION

   [DEVICE=<c>] [PLOT.OUT=<c>]
   [ {QUIET | NORMAL | VERBOSE | DEBUG} ]
   [INFORMAT] [DIAGNOST] [STRICT]
   [ECHO] [EXECUTE] [V.COMPAT]
   [ZIP.FILE] [ZIP.EXE=<c>] [UNZIP.EX=<c>]
   [ZIP.SUFF=<c>]
   [AUTOSAVE=<n>]
```

Parameter	Type	Definition
DEVICE	character	The name of the graphics device to be used for plotting. **Default:** none
PLOT.OUT	character	The file to which graphics output is sent. If the name is preceded by a "+", the output is appended to an existing file; otherwise, an existing file is overwritten or a new file is created. **Default:** none **Synonyms: FILE.SAV**, **PLOT.SAV**
QUIET	logical	Print a minimum of information in the standard output and the output listing file. **Default:** the current value; initially false
NORMAL	logical	Print the "normal" amount of information in the standard output and the output listing file. **Default:** the current value; initially true
VERBOSE	logical	Print more than the normal amount of information in the standard output and the output listing file. **Default:** the current value; initially false
DEBUG	logical	Include all available debugging information in the standard output and the output listing file. **Default:** the current value; initially false
INFORMAT	logical	Send additional output to the informational output file. **Default:** the current value; initially false
DIAGNOST	logical	Send additional output to the diagnostic output file. **Default:** the current value; initially false
STRICT	logical	Verify the syntax strictly **Default:** the current value; initially false
ECHO	logical	Echo each input statement to the standard output as it is processed. **Default:** the current value; initially true

Parameter	Type	Definition
EXECUTE	logical	Execute each input statement as it is read. If **EXECUTE** is false, input statements are checked for syntax but not executed. **Default:** the current value; initially true
V.COMPAT	number	Change certain models and algorithms to be compatible with the specified version of TSUPREM-4. **Default:** the current version of TSUPREM-4
ZIP.FILE	logical	Specifies that all files to be saved and loaded are in a compressed format. **Default:** false
ZIP.EXE	character	String to command to compress. **Default:** “gzip -c”
ZIP.FILE	logical	Specify that all files to be saved and loaded are in a compressed format. **Default:** false
AUTOSAVE	number	Specify the level of automatic save when a simulation stops abnormally by a syntax error or an internal fatal error or CTRL-C. **Units:** none **Default:** 0

Selecting a Graphics Device

Use the **DEVICE** parameter to specify the type of graphics device to be used for plotting. The device types known to the program are described in the *s4pcap* file; a listing of the available devices is given in Appendix B. If no **DEVICE** is specified, a default device is chosen as follows:

1. If the environment variable *DEFPDEV* is defined as the name of a valid plot device, its value is used as the plot device.
2. Otherwise, if the environment variable *TERM* is defined as the name of a valid plot device, its value is used as the plot device.
3. Otherwise, the **DEFAULT** device in the *s4pcap* file is used. *Note* that the *s4pcap* file can be modified to make the **DEFAULT** device refer to any available real plotting device (see Appendix B).

Redirecting Graphics Output

Graphics output normally goes to your terminal or to the file specified in the *s4pcap* file. In either case, the output can be redirected to the file specified by the **PLOT.OUT** parameter. This allows graphics output to be saved for later processing or display. When graphics output is redirected by **PLOT.OUT**, it does not appear on your terminal or in the file specified in *s4pcap*.

Printed Output

The **QUIET**, **NORMAL**, **VERBOSE**, and **DEBUG** parameters control the amount of printed output generated by the program. **NORMAL** mode is assumed when the program begins. Output from the **VERBOSE** and **DEBUG** modes is not needed during typical usage of TSUPREM-4 and may be difficult to interpret.

Informational and Diagnostic Output

The **INFORMAT** and **DIAGNOST** parameters specify whether informational and diagnostic output are written to the appropriate files. If output to one of these files is disabled before any information is written, the file is not created. The default initialization file *s4init* sets both these parameters false, disabling both files.

Echoing and Execution of Input Statements

The **ECHO** parameter controls whether input statements are listed on the standard output (usually your terminal) as they are processed. By default, this option is turned *on*. The **EXECUTE** parameter controls whether statements are executed as they are encountered. If **EXECUTE** is false, statements are checked for syntax but not processed. This is useful for doing a preliminary syntax check of an input file. The **DEFINE**, **UNDEFINE**, **FOREACH**, **SOURCE**, **RETURN**, **OPTION**, and **STOP** statements are always executed, regardless of the setting of the **EXECUTE** option.

Version Compatibility

The **V.COMPAT** parameter is used to improve compatibility with older versions of TSUPREM-4. When an older version number is specified, certain models and algorithms from the older version are used in place of the current models and algorithms. Only some changes between versions are reversed by using **V.COMPAT**. For a description of changes in the current version that are affected by **V.COMPAT**, see the TSUPREM-4 Release Notes for Version TS4 2007.03.

Strict Syntax Check

Stricter syntax verification of the input keywords has been implemented. Strict syntax verification can be switched on in the input file by the command **OPTION STRICT**. By turing on **STRICT**, a vocabulary keyword can not be a substring of the input keyword. This is valid for up to 8 letters of the input keywords; all other letters are not verified. For example, TSUPREM-4 recognizes **MOMEN** in the input file as **MOMENT** keyword, but does not recognize **MOMENTS**. The strict syntax check detects the mistake that users made when they confused a built-in keyword and an input string. For example, users define a new impurity BSIGE by:

```
IMPURITY IMP=BSIGE NEW IMPL.TAB=TR.BORON C.INIT=1E5
```

Since the new impurity BSIGE is different from the built-in impurity B, the implantation of BSIGE must be given by:

```
IMPLANT IMP=BSIGE DOSE=1E15 ENERGY=30
```

Note that BSIGE is given as an input string for the **IMP** built-in keyword. However, when BSIGE is given like a built-in keyword, TSUPREM-4 recognizes B for BSIGE because the ‘SIGE’ of BSIGE is ignored. For example,

```
IMPLANT BSIGE DOSE=1E15 ENERGY=30
```

the above statement performs not BSIGE implant, but B implant. Turning on **STRICT** catches this kind of mistake.

Automatic Save

When a simulation stops abnormally by a syntax error or an internal fatal error or CTRL-C, the **AUTOSAVE** mode stores the simulation result before such an accident so that users can continue the simulation after fixing the error without rerunning whole simulation again. The **AUTOSAVE** mode is specified in the **OPTION** statement. Three modes are provided as follows:

- **OPTION AUTOSAVE=0**

  No autosave mode.
- **OPTION AUTOSAVE=1**

  Recommended. Save the latest result among CPU intensive processes such as **DIFFUSE**, **EPITAXY** or **IMPLANT** before abnormal stop.
- **OPTION AUTOSAVE=2**

  Save the latest result among **DIFFUSE**, **EPITAXY**, **IMPLANT**, **STRUCTURE**, **ETCH**, **DEPOSIT**, **DEVELOP**, **EXPOSE**, **PROFILE** or **STRESS** before abnormal stop

The file format to save temporary solutions is the TSUPREM-4 structure file format and the file is named with

<prefix of input file name>

For example, if the input file name is *test.inp*, the saved file is

*_test_*

Note:

> ***AUTOSAVE does not work for the unexpected exceptions, such as segmentation faults or bus errors caused by memory conflicts.***

Load/Save Files in Compressed Format

The size of a saved file tends to increase as the simulation models used for the recent process technology becomes complicated. In addition, saving intermediates increases the file size. Saving the text files such as TIF or TS4 structure files in compressed format reduces the size by the factor of 7~8, which depends on the compression program.

The parameter **ZIP.EXE** and **UNZIP.EX** defines the command for compression and uncompression, respectively. The **ZIP.SUFF** defines the suffix of a compressed file name.

The parameter **ZIP.FILE** specifies to save/load the files in compressed format:

```
OPTION ZIP.FILE
```

The compression/uncompression also can be given by turning on/off the **ZIP.FILE** parameter on each statement of the **INITIALIZE**, **LOADFILE**, **SAVEFILE**, and **PROFILE** commands.

Examples

- The following statement causes the graphics output to be produced for a Tektronix 4100-series terminal and appended to the file *plotsave*:

```
OPTION DEVICE=4100 PLOT.OUT="+plotsave"
```

- Additional information is printed on the standard output with the statement:

```
OPTION VERBOSE
```

- Enhanced compatibility with TSUPREM-4 Version 6.4 can be obtained with:

```
OPTION V.COMPAT=6.4
```

- In *s4init*, gzip command is set for compression as follows:

```
OPTION ZIP.EXE="gzip -c" UNZIP.EX="gunzip -c" +
       ZIP.SUFF=gz
```

  For using the UNIX compress command:

```
OPTION ZIP.EXE="compress -c" +
       UNZIP.EX="uncompress -c" ZIP.SUFF=Z
```

- When **ZIP.FILE** is turned on in the **OPTION** statement, the file is saved in the compressed format even when **ZIP.FILE** is not specified in **SAVEFILE**:

```
OPTION    ZIP.FILE
INITIAL   BORON=1E15
SAVEFILE  OUT.FILE=test.sav ;$--> test.sav.gz
```

- If you do not want to compress the specific file, then turn off **ZIP.FILE** on the specific command:

```
OPTION  ZIP.FILE
INITIAL BORON=1E15
SAVEFIL OUT.FILE=test1.sav ;$--> test1.sav.gz
IMPLANT ARSENIC DOSE=1E14 ENERGY=30
SAVEFIL OUT.FILE=test2.tif TIF
                           ;$--> test2.tif.gz
DIFFUSE TEMP=900 TIME=10
SAVEFIL OUT.FILE=test3.tif TIF ^ZIP.FILE
                              ;$--> test3.tif
```

- When loading the file, the file is assumed to be compressed if the parameter **ZIP.FILE** was turned on in the **OPTION** statement.

```
OPTION  ZIP.FILE
INITIAL IN.FILE=test1.tif TIF ;$<-- test1.tif.gz
INITIAL IN.FILE=test2.tif TIF ^ZIP.FILE
                              ;$<-- test2.tif

OPTION  ^ZIP.FILE
INITIAL IN.FILE=test1.tif TIF ;$<-- test1.tif
INITIAL IN.FILE=test2.tif TIF ZIP.FILE
                              ;$<-- test2.tif.gz
```

DEFINE

The **DEFINE** statement defines strings for substitution in subsequent input statements.

```
DEFINE

    [<name> <body>]
```

Description

This statement defines the name <*name*> as an abbreviation for the string <*body*>. Any time <*name*> appears in an input line as a separate token (see below), <*body*> is substituted. This allows long, often-used sequences to be abbreviated. **DEFINE** without any parameters lists the current definitions. To undefine a <*name*>, use the **UNDEFINE** statement.

Format and Syntax

The **DEFINE** statement uses the following format and syntax. The defined <*name*> can only contain letters and digits. A name is recognized and expanded only if it is preceded and followed by one of the following separators: space, tab, newline, <, >, &, “;”, !, {, }, =, “,”, $, @, or (. To ensure that defined names are recognized in arithmetic expressions, the name should be preceded and followed by spaces, and the entire expression enclosed in parentheses. Expansion of abbreviations can be forced by preceding the defined name with the “@” character; in this case the name can also be enclosed in braces to separate it from surrounding characters. Older versions of the program used the “$” character to force expansion; this usage is now obsolete, although it is still recognized by the program. Within a character string, you can force recognition of a defined name by enclosing the name in braces (“{” and “}”) and preceding the left brace with the “@” (or “$”) character.

Substitution of abbreviations is inhibited by the “%” character. When “%” is encountered in an input line, expansion of abbreviations is inhibited for the rest of the line, except when forced by the “@” (or “$”) character.

Examples

- The following statements define and use the name *LIMITS*:

  ```
  DEFINE LIMITS X.MIN=0.0 X.MAX=5.0 +
         Y.MIN=0.0 Y.MAX=20.0
  PLOT.2D LIMITS
  ```

  The parameter specifications for **X.MIN**, **X.MAX**, **Y.MIN**, and **Y.MAX** are substituted whenever the name *LIMITS* is encountered. Thus, the **PLOT.2D** statement is equivalent to

  ```
  PLOT.2D X.MIN=0.0 X.MAX=5.0 Y.MIN=0.0 Y.MAX=20.0
  ```

- For the statements:

  ```
  DEFINE W 2.0
  ECHO 1-W
  ECHO 1-@W
  ```

the output is:

```
1-W
-1
```

In the first **ECHO** statement, "-" is not recognized as a separator. In the second **ECHO** statement, the "@" forces expansion of the abbreviation *W.*

- The statements:

```
DEFINE PROC C41
SAVEFILE OUT.FILE=MOS@{PROC}A.DAT
```

save the structure in file *MOSC41A.DAT.* The defined name *PROC* is inserted in the middle of a text string with the "@{}" construct to force expansion.

Usage Notes

- Abbreviations are expanded whenever they are encountered, including in **DEFINE** and **UNDEFINE** statements. The expanded version of an abbreviation is not checked for further abbreviations, however. For example,:

```
DEFINE a b
DEFINE c a
```

defines both *a* and *c* as "b," but:

```
DEFINE a b
%DEFINE c a
```

defines *a* as "b" and *c* as "a." References to *c* produce "a," which is *not* expanded to "b."

- Similarly, to redefine an name, you must use the "%" character. For example:

```
DEFINE a b
DEFINE a c
```

defines *a* as "b" and *b* as "c." To define *a* as "c," use:

```
DEFINE a b
%DEFINE a c
```

The substitution of "b" for *a* in the second line is prevented by the "%" character.

- When undefining an abbreviation, be sure to use the "%" character on the **UNDEFINE** statement:

```
DEFINE a b
UNDEFINE a
```

does not work, because the **UNDEFINE** statement is expanded to

```
UNDEFINE b
```

and *b* is not defined. To undefine *a*, use:

```
%UNDEFINE a
```

- A line beginning with "$" is not treated as a **COMMENT** statement if the "$" is immediately followed by the name of a defined abbreviation. The expansion of the abbreviation is substituted. Thus, in the statements

```
DEFINE  THIS  1.0
$THIS IS NOT A COMMENT
```

the value 1.0 is substituted for the name *THIS* in the second statement, giving:

```
1.0 IS NOT A COMMENT
```

which results in a syntax error.

- It is not possible to assign a null value to a name with the **DEFINE** statement.
- Use care when choosing names to be defined. You normally want to avoid using names that are also valid statement or parameter names. For example, given:

```
DEFINE TIME 20
DIFFUSE TIME=TIME
```

The second statement is expanded as:

```
DIFFUSE 20=20
```

resulting in a syntax error.

- For many applications it is easier to use the **ASSIGN** and **LOOP** statements instead of **DEFINE** and **FOREACH**. See ASSIGN, p. 4-23 and LOOP/L.END, p. 4-16.

UNDEFINE

The **UNDEFINE** statement undefines a previously defined abbreviation.

```
UNDEFINE

    <name>
```

Description

The **UNDEFINE** statement is used to turn *off* previously defined abbreviations. The character string <*name*> gives the name of the abbreviation to be deleted. The statement should normally be given as **%UNDEFINE**; the "%" is required to prevent substitution for abbreviations in the **UNDEFINE** statement itself.

The **UNDEFINE** statement is also useful for correcting mistakes made with the **DEFINE** statement. For more examples of the **UNDEFINE** statement, see DEFINE, p. 4-46.

Example

The statements:

```
DEFINE   W  2.0
ECHO     1 - W
%UNDEFINE W
ECHO     1 - W
```

produce the output:

```
-1
1 - W
```

Redefined Parameter Names

The **UNDEFINE** statement is most often needed when a parameter name has been redefined. For example, you may wish to do the following:

```
DEFINE   COMPRESS  VISCOUS
SOURCE   OXIDE1
%UNDEFINE COMPRESS
SOURCE   OXIDE2
```

This forces any **METHOD COMPRESS** statements in file *OXIDE1* to be interpreted as **METHOD VISCOUS** statements. By undefining the name *COMPRESS*, this substitution is not made when executing file *OXIDE2*.

CPULOG

The **`CPULOG`** statement controls the logging of CPU usage.

```
CPULOG

    [LOG] [OUT.FILE=<c>]
```

Parameter	Type	Definition
`LOG`	logical	Enables logging of CPU usage when true; disables CPU logging when false. **Default**: true
`OUT.FILE`	character	The file to which the CPU log is written. **Default**: standard output **Synonym**: **`CPUFILE`**

Description

The **`CPULOG`** statement instructs the program to record the amount of CPU time used in various internal operations. The information appears on the standard output and in the output listing file unless it is redirected with the **`OUT.FILE`** parameter. Most CPU-intensive operations report the time used.

Examples

- The following statement enables reporting of CPU statistics to the standard output and the output listing file:

  ```
  CPULOG LOG
  ```

- The following statement enables CPU statistics reporting, and stores the output in the file *timefile*.

  ```
  CPU LOG OUT.FILE=timefile
  ```

Limitations

The accuracy of the times depends on the computer being used. A resolution of 1/60 seconds is typical. Some systems cannot report CPU time at all. Reported times are zero for those systems. Timing for a complete TSUPREM-4 simulation is best obtained using operating system commands, such as the **`TIME`** command in the UNIX C-shell.

HELP

The **`HELP`** statement prints a summary of statement names and parameters.

```
HELP
    [<name>]
```

Description

The **`HELP`** statement prints a summary of statement names and parameters to the standard output. If no <*name*> is given, a summary of statement names is printed. If the name of a statement is given, a summary of parameters for that statement is printed. The parameter summary includes the name type, and default for each parameter.

Example

The statement:

```
HELP DIFFUSION
```

prints a summary of the parameters on the **`DIFFUSION`** statement.

Notes

- The default values printed by the **`HELP`** statement are not always helpful. This is especially true for statements that specify the values of model coefficients. These statements usually ignore parameters that are not specified rather than use the default values. The default value is likewise meaningless for parameters that must be specified on a statement.
- Help is not always available for statements that use nonstandard syntax (i.e., special statements)

!(Exclamation Mark)

The exclamation mark allows to call an unix shell command. For example:

```
SAVEFILE OUT.FILE=a.tif TIF
$ copy a.tif to b.tif by calling unix command ‘cp’
!cp a.tif b.tif
```

Device Structure Specification

The following statements specify the device structure used by TSUPREM-4:

Statement Name	Description	Page
`MESH`	Sets grid spacing scale factor and defaults for automatic grid generation.	p. 4-53
`LINE`	Specifies a grid line in a rectangular mesh.	p. 4-57
`ELIMINATE`	Specifies grid lines to be removed from parts of the mesh.	p. 4-59
`BOUNDARY`	Sets boundary conditions for a rectangular mesh.	p. 4-62
`REGION`	Sets material types for a rectangular mesh.	p. 4-64
`INITIALIZE`	Initializes a rectangular mesh or reads mesh and solution information from a file.	p. 4-66
`LOADFILE`	Reads mesh and solution information from a file.	p. 4-71
`SAVEFILE`	Writes mesh and solution information to a file.	p. 4-74
`STRUCTURE`	Reflects, truncates, extends, or unrefines a structure.	p. 4-79
`MASK`	Reads mask information from a file.	p. 4-84
`PROFILE`	Reads a 1D doping profile from a file.	p. 4-86
`ELECTRODE`	Specifies the name and position of electrodes.	p. 4-96

MESH

The **`MESH`** statement specifies a grid spacing scale factor and default values for controlling automatic grid generation.

```
MESH

    [GRID.FAC=<n>]
    [DX.MAX=<n>] [DX.MIN=<n>]  [DX.RATIO=<n>]
    [LY.SURF=<n>] [DY.SURF=<n>] [LY.ACTIV=<n>] [DY.ACTIV=<n>]
    [LY.BOT=<n>] [DY.BOT=<n>]  [DY.RATIO=<n>]
    [FAST]
```

Parameter	Type	Definition
`GRID.FAC`	number	A factor by which all grid spacing specifications are multiplied. The scaling occurs when the spacings are used (not when they are specified.) **Units:** none **Default:** the current value
`DX.MAX`	number	The maximum grid spacing in the *x* direction (i.e., between vertical grid lines) in the default horizontal grid. This is the spacing to be used far from mask edges. This spacing is multiplied by **`GRID.FAC`** when it is used. **Units:** microns **Default:** the current value
`DX.MIN`	number	The minimum grid spacing in the *x* direction (i.e., between vertical grid lines) in the default horizontal grid. This is the spacing to be used at mask edges. This spacing is multiplied by **`GRID.FAC`** when it is used. **Units:** microns **Default:** the current value
`DX.RATIO`	number	The ratio by which grid spaces are increased from **`DX.MIN`** to **`DX.MAX`** in the default horizontal grid. **Units:** microns **Default:** the current value
`LY.SURF`	number	The depth of the surface region in the default vertical grid. **Units:** microns **Default:** the current value
`DY.SURF`	number	The grid spacing in the *y* direction (i.e., between horizontal grid lines) in the surface region of the default vertical grid. This spacing is used between *y*=0 and *y*=**`LY.SURF`**. This spacing is multiplied by **`GRID.FAC`** when it is used. **Units:** microns **Default:** the current value
`LY.ACTIV`	number	The depth of the bottom of the active region in the default vertical grid. **Units:** microns **Default:** the current value

Parameter	Type	Definition
DY.ACTIV	number	The grid spacing in the *y* direction (i.e., between horizontal grid lines) at the bottom of the active region in the default vertical grid. The grid spacing varies geometrically between **DY.SURF** at **LY.SURF** and **DY.ACTIV** at **LY.ACTIV**. This spacing is multiplied by **GRID.FAC** when it is used. **Units:** microns **Default:** the current value
LY.BOT	number	The depth of the bottom of the structure in the default vertical grid. **Units:** microns **Default:** the current value
DY.BOT	number	The grid spacing in the *y* direction (i.e., between horizontal grid lines) at the bottom of the structure in the default vertical grid. This spacing is multiplied by **GRID.FAC** when it is used. **Units:** microns **Default:** the current value
DY.RATIO	number	The ratio by which grid spaces are increased from **DY.ACTIV** to **DY.BOT** in the default vertical grid. **Units:** microns **Default:** the current value
FAST	logical	Specifies that 1D simulation is used until an etch step creates a structure that varies in the *x* direction. The structure is automatically converted to 2D status when required. **Default:** true

Description

The **MESH** statement controls the automatic generation of simulation grids for TSUPREM-4. A grid is generated whenever an **INITIALIZE** statement without an **IN.FILE** parameter is processed. Automatic grid generation is used unless an appropriate set of **LINE** statements has been processed since the previous **INITIALIZE** statement was processed. Grids can be automatically generated in both the *x* and *y* directions, and automatic grid generation in one direction can be combined with manual grid specification on the other direction.

Grid Creation Methods

A simulation grid for TSUPREM-4 can be created in one of four ways:

- It can be read from a saved structure file, using the **INITIALIZE** or **LOADFILE** statements.
- It can be generated from user-specified **LINE**, **REGION**, **BOUNDARY**, and **ELIMINATE** statements.
- It can be generated automatically from parameters supplied on the **MESH** statement and from information read from mask data files.
- The last two methods can be combined, using one of the methods for generating the horizontal grid and the other method for generating the vertical grid.

If no **`LINE`** statements are supplied before an **`INITIALIZE`** statement, a grid is generated automatically. If **`LINE X`** statements are supplied, they are used (instead of automatic generation) for the horizontal grid; if **`LINE Y`** statements are supplied, they are used to generate the vertical grid.

Horizontal Grid Generation

The automatic generation of the horizontal grid depends on whether mask information has been read with the **`MASK`** statement.

- If no mask information has been read, the default horizontal grid consists of two (vertical) grid lines located at y=0.0 and y=1.0. This produces a quasi 1D simulation.
- If mask information is available, it is used with the **`DX.MIN`**, **`DX.MAX`**, and **`DX.RATIO`** parameters to automatically generate a horizontal grid.

The horizontal grid has a spacing between (vertical) grid lines of approximately **`DX.MIN`** at mask edges, expanding to a spacing of **`DX.MAX`** at distances far from any mask edge. The **`DX.RATIO`** parameter specifies the rate at which the spacing increases away from mask edges. The minimum and maximum grid locations are set to the minimum and maximum x coordinates for which mask information is supplied in the mask data file(s).

For a complete description of the grid generation process, see Chapter 3, Automatic Grid Generation, p. 3-5.

Vertical Grid Generation

The automatically generated vertical grid is controlled by the various **`LY`** and **`DY`** parameters.

- **`LY.SURF`**, **`LY.ACTIV`**, and **`LY.BOT`** specify the locations of the bottom of the surface region, active region, and structure, respectively.
- **`DY.SURF`**, **`DY.ACTIV`**, and **`DY.BOT`** specify the grid spacings at these locations.

Between the top of the structure and **`LY.SURF`**, the grid spacing **`DY.SURF`** is used. The spacing increases geometrically (i.e., with a constant ratio) from **`DY.SURF`** at **`LY.SURF`** to **`DY.ACTIV`** at **`LY.ACTIV`**. The spacing expands to **`DY.BOT`** below **`LY.ACTIV`**. The **`DY.RATIO`** parameter specifies the rate at which the spacing increases below **`DY.ACTIV`**.

In a automatically generated vertical grid, vertical grid lines deeper than **`LY.ACTIV`** are eliminated, to increase the speed of the simulation. It is assumed that only point defects diffuse below this depth and that the point defect profiles are approximately constant in the horizontal direction, so that the low horizontal resolution does not cause significant loss of accuracy in the simulation.

For a complete description of the grid generation process see Chapter 3, Automatic Grid Generation in the Y Direction, p. 3-7.

Scaling the Grid Spacing

The **GRID.FAC** parameter multiplies all grid spacing specifications, including those on the **MESH**, **LINE**, **DEPOSITION**, and **EPITAXY** statements. The default values of all grid spacings are set so that **GRID.FAC**=1.0 produces a moderately fine grid, useful for reasonably, accurate simulations. For faster simulations, **GRID.FAC** should be given a value greater than 1.0. For more accurate simulations, **GRID.FAC** should be reduced as needed.

The **GRID.FAC** parameter modifies the values of other parameters in the following ways:

- The **DX.MIN**, **DX.MAX**, **DY.SURF**, **DY.ACTIV**, and **DY.BOT** parameters on the **MESH** statement are multiplied by **GRID.FAC**.
- The **DY** and **ARC.SPAC** parameters on the **DEPOSITION** and **EPITAXY** statements are multiplied by **GRID.FAC**.
- The **SPACES** parameter on the **DEPOSITION** and **EPITAXY** statements is divided by **GRID.FAC**.
- The **DY.OXIDE** on the **METHOD** statement and the **DY.DEFAU** parameter on the **MATERIAL** statement are multiplied by **GRID.FAC** (the modification takes place when the parameters are used, not when they are specified).

Note:

Moderation should be used in adjusting* GRID.FAC*—reducing the value from 1.0 to 0.1 increases the grid density by a factor of 10 in each direction, increasing the node count by a factor of 100 and the simulation time by a factor of more than 100.

1D Mode

Normally, when a mesh is generated or read from a file, it is examined to determine whether there is any variation of the structure or solution values in the *x* direction. If there is no variation, then simulation proceeds in 1D mode. By setting the **FAST** parameter false, the use of 1D mode is disabled. If 1D mode is already in use, the structure is converted to 2D.

Examples

- The following **MESH** statements could be used in the *s4init* file to set the default parameters for automatic grid generation:

```
MESH DX.MIN=0.1 DX.MAX=0.4 DX.RATIO=1.5
MESH LY.SURF=0.1 DY.SURF=0.03 LY.ACTIV=4.0
+
     DY.ACTIV=0.3 LY.BOT=200 DY.BOT=100 +
     DY.RATIO=1.5
```

- The following statement cuts all grid spacings to one half their specified value, giving twice the specified grid density in the horizontal and vertical directions:

```
MESH GRID.FAC=0.5
```

LINE

The **LINE** statement specifies a horizontal or vertical mesh line in a non-uniform rectangular grid.

```
LINE

  {X | Y} LOCATION=<n> [SPACING=<n>] [TAG=<c>]
```

Parameter	Type	Definition
X	logical	If true, then **LOCATION** specifies the *x* coordinate of a vertical grid line. **Default:** false
Y	logical	If true, then **LOCATION** specifies the *y* coordinate of a horizontal grid line. **Default:** false
LOCATION	number	The *y* coordinate of a horizontal grid line or the *x* coordinate of a vertical grid line. **Units:** microns **Default:** none
SPACING	number	The local grid spacing. **Units:** microns **Default:** none
TAG	character	A name used to refer to this grid line on a **BOUNDARY** or **REGION** statement. **Default:** none

Description

This statement specifies the position and spacing of one line in a rectangular mesh. A complete mesh specification consists of a group of **LINE** statements followed by **ELIMINATE** statements (optional), **REGION** and **BOUNDARY** statements (optional), and an **INITIALIZE** statement.

TSUPREM-4 uses an inverted Cartesian coordinate system, with *x* increasing from left to right and *y* increasing from the top surface of the structure into the substrate.

Placing Grid Lines

Grid lines are placed at the locations specified on **LINE** statements. Additional lines are added automatically so that the final grid spacing varies geometrically between the **SPACING** values at the user-specified lines. The spacing ratio between adjacent pairs of grid lines is guided by the value of the **RATIO** parameter on the **INITIALIZE** statement.

The final grid spacing may be slightly larger than that specified on the **LINE** statement because of the need for an integral number of spaces; it may be much smaller than specified to satisfy constraints imposed by neighboring grid lines and the **RATIO** parameter. A complete description of the grid setup process is given in Chapter 3, Explicit Specification of Grid Structure, p. 3-3.

Example

The following statements include three user-specified vertical grid lines and two user-specified horizontal grid lines:

```
LINE  X  LOCATION=0  SPACING=1
LINE  X  LOCATION=1  SPACING=0.1
LINE  X  LOCATION=2  SPACING=1
LINE  Y  LOCATION=0  SPACING=0.02
LINE  Y  LOCATION=3  SPACING=0.5
```

After processing with **RATIO**=1.5, TSUPREM-4 produces a mesh with vertical grid lines at 0.0, 0.42, 0.69, 0.88, 1.0, 1.12, 1.31, 1.58, and 2.0. Around the center, the spacing is 0.12, approximately what was requested. At each edge, the spacing is 0.42, because that is as coarse as it could get without exceeding the allowed ratio between adjacent grid spaces. If the allowed ratio were 9, then you would get one space of 0.9 microns and one space of 0.1 micron on each side of the center line.

Additional Notes

Below are some additional notes on the **LINE** statement.

Structure Depth and Point Defect Models

When the **PD.TRANS** or **PD.FULL** or **NSTREAM=5** point defect model is to be used, the structure should be deep to accommodate the deep diffusion of point defects. A depth of 50 to 200 microns is suggested. **ELIMINATE** statements can be used to eliminate all but two vertical grid lines deep in the structure. When **ELIMINATE** statements are used with a coarse vertical grid spacing, the computational overhead of using a deep structure can be made negligible.

- Maximum Number of Nodes and Grid Lines

The program can handle up to 100,000 nodes; this includes one node per grid point, plus one node for each exposed point and another node for each point on a boundary between two materials. Thus, the grid must be smaller than 100,000 points. In addition, the initial grid must not have more than 1000 grid lines in either the *x* or *y* direction.

Default Regions and Boundaries

No tags are required if the default regions and boundary conditions are to be used, i.e., if the initial structure consists entirely of silicon with only the top surface exposed.

ELIMINATE

The `ELIMINATE` statement eliminates mesh nodes along lines in a grid structure over a specified rectangular region.

```
ELIMINATE

   {ROWS | COLUMNS}
  [X.MIN=<n>] [X.MAX=<n>] [Y.MIN=<n>] [Y.MAX=<n>]
```

Parameter	Type	Definition
`ROWS`	logical	Specifies that horizontal lines of nodes are eliminated. **Default:** false **Synonym:** `X.DIREC`
`COLUMNS`	logical	Specifies that vertical lines of nodes are eliminated. **Default:** false **Synonym:** `Y.DIREC`
`X.MIN`	number	The minimum horizontal location of the rectangular region over which nodes are eliminated. **Units:** microns **Default:** the minimum horizontal location of the device structure **Synonym:** `X.LOW`
`X.MAX`	number	The maximum horizontal location of the rectangular region over which nodes are eliminated. **Units:** microns **Default:** the maximum horizontal location of the device structure **Synonym:** `X.HIGH`
`Y.MIN`	number	The minimum vertical location of the rectangular region over which nodes are eliminated. **Units:** microns **Default:** the minimum vertical location of the device structure **Synonym:** `Y.LOW`
`Y.MAX`	number	The maximum vertical location of the rectangular region over which nodes are eliminated. **Units:** microns **Default:** the maximum vertical location of the device structure **Synonym:** `Y.HIGH`

Description

The `ELIMINATE` statement is used to eliminate nodes in regions of the device structure where the grid would otherwise be more dense than necessary. The `ROWS` parameter specifies that every second row of nodes within the specified region is to be removed. Similarly, the `COLUMNS` parameter specifies that every second column of nodes within the specified region is to be removed.

The **`ELIMINATE`** statement can only be used after the **`LINE`** statements and before the **`INITIALIZE`** statement.

Note:

Elimination of grid lines takes place during initial structure generation. The `ELIMINATE` statement cannot be used to eliminate grid lines after the `INITIALIZE` statement has been processed.

It is frequently a good idea to plot and examine the initial grid after the **`LINE X`**, **`LINE Y`**, **`ELIMINATE`**, and **`INITIALIZE`** statements to verify the desired grid structure before proceeding.

Reducing Grid Nodes

The **`ELIMINATE`** statement can be very useful for reducing the number of grid nodes (and hence execution time), especially for large structures. In particular, you might use **`LINE X`** and **`LINE Y`** statements to define a large structure that would exceed the limit of 100,000 nodes, then eliminate enough rows and columns to bring the final number of nodes under 100,000.

Overlapping Regions

If you get an error message stating that one of the **`ELIMINATE`** statements is causing a mesh generation error, it is due to overlapping or adjacent eliminate regions. For best results when performing multiple eliminations over a structure, either:

1. Do the multiple eliminations over identical regions of the structure, or
2. Avoid overlapping or adjacent eliminate regions (i.e., separate the regions by at least one grid line).

If you must have overlapping eliminate regions, best results are obtained if the **`ELIMINATE`** statements define successively smaller regions. That is, start with the largest region, then eliminate additional nodes in subsets of that region. If you need to eliminate over partially overlapping regions, best results are obtained if you minimize the amount of overlap.

Examples

- The following statement eliminates every other row of nodes over the entire structure:

  ```
  ELIMINATE ROWS
  ```

  The same result could be obtained by changing the locations and grid spacings on the **`LINE Y`** statements.

- The following statement eliminates every other column of nodes over the full width of the structure, but only for values of *y* greater than 4.5 microns:

  ```
  ELIMINATE COLUMNS Y.MIN=4.5
  ```

- The following statements perform multiple eliminations of rows and columns over the same region of the structure:

```
ELIMINATE ROWS    X.MAX=3  Y.MAX=2.2
ELIMINATE ROWS    X.MAX=3  Y.MAX=2.2
ELIMINATE COLUMNS X.MAX=3  Y.MAX=2.2
ELIMINATE COLUMNS X.MAX=3  Y.MAX=2.2
```

- The following statements eliminate columns in two non-overlapping, non-adjacent regions of the structure:

```
ELIMINATE COLUMNS X.MIN=1 X.MAX=3
ELIMINATE COLUMNS X.MIN=7 X.MAX=9
```

  Each region extends from the top to the bottom of the structure.

BOUNDARY

The **`BOUNDARY`** statement specifies a boundary condition along an edge of the structure.

```
BOUNDARY

    {REFLECTI | EXPOSED}
    XLO=<c> XHI=<c> YLO=<c> YHI=<c>
```

Parameter	Type	Definition
`REFLECTI`	logical	The specified boundary is reflecting. **Default:** false
`EXPOSED`	logical	The specified boundary is exposed. **Default:** false
`XLO`	character	The tag corresponding to the left edge of the boundary. **Default:** none
`XHI`	character	The tag corresponding to the right edge of the boundary. **Default:** none
`YLO`	character	The tag corresponding to the top edge of the boundary. **Default:** none
`YHI`	character	The tag corresponding to the bottom edge of the boundary. **Default:** none

Description

This statement specifies the boundary conditions that apply at the surfaces in a rectangular mesh. Two surface types are recognized: exposed and reflecting.

Exposed surfaces normally correspond to the top of the wafer. Deposition, oxidation, and out-diffusion occur at exposed surfaces. Impurity predeposition also happens at exposed surfaces, as do defect recombination and generation. The default boundary condition for the sides and the back of the structure is reflecting, while the default for the top surface is exposed. Thus the **`BOUNDARY`** statement is not required for most simulations. The tags specified with the **`XLO`**, **`XHI`**, **`YLO`**, and **`YHI`** parameters correspond to tags specified on the **`LINE`** statements used to define the mesh. **`XLO`**, **`XHI`**, **`YLO`**, and **`YHI`** must specify one entire edge of the structure. It is not permissible to specify part of an edge, a region of nonzero area, or a line in the interior of the structure.

Limitations

At present, the **`BOUNDARY`** statement is only useful in special situations, due to the following limitations in the various processing steps:

- **`DEPOSITION`** only works on the top and right side (when exposed) of the structure, and may fail if the bottom of the structure is exposed.
- **`ETCH`** with the **`TRAPEZOI`** and **`OLD.DRY`** options assumes that the exposed surface is etched from the top. Etching with the **`TRAPEZOI`** and **`OLD.DRY`** options may fail if the sides or bottom of the structure are exposed.
- Ion implantation always occurs on the top surface.
- **`DIFFUSION`** works correctly on all exposed surfaces, but the analytical and **`VERTICAL`** oxidation models assume that oxidation only occurs at the top surface.

The **`BOUNDARY`** statement is only used in setting up the initial structure. It cannot be used to change a boundary condition after the structure has been initialized.

Example

The following statement specifies that the top of the mesh is the exposed surface:

```
     BOUNDARY EXPOSED XLO=left XHI=right YLO=surf
YHI=surf
```

The tags **`left`**, **`right`**, and **`surf`** must have been previously specified on the **`LINE`** statements used to define the mesh. See the description of the **`LINE`** statement on p. 4-57.

REGION

The **`REGION`** statement specifies the material type of a mesh region.

```
REGION

   { MATERIAL=<c> | SILICON | OXIDE | OXYNITRI | NITRIDE | POLYSILI
     | PHOTORES | ALUMINUM
   }
   {  (XLO=<c> XHI=<c> YLO=<c> YHI=<c>)
     | (X=<n> Y=<n> NEW.MATE=<c> [KEEP])
   }
```

Parameter	Type	Definition
`MATERIAL`	character	The specified region is defined to be of the named material. **Default:** none
`SILICON`	logical	The specified region is defined to be silicon. **Default:** false
`OXIDE`	logical	The specified region is defined to be oxide. **Default:** false
`OXYNITRI`	logical	The specified region is defined to be oxynitride. **Default:** false
`NITRIDE`	logical	The specified region is defined to be nitride. **Default:** false
`POLYSILI`	logical	The specified region is defined to be polysilicon. **Default:** false
`PHOTORES`	logical	The specified region is defined to be photoresist. **Default:** false
`ALUMINUM`	logical	The specified region is defined to be aluminum. **Default:** false
`XLO`	character	The tag specifying the left edge of the region to be defined. **Default:** none
`XHI`	character	The tag specifying the right edge of the region to be defined. **Default:** none
`YLO`	character	The tag specifying the top edge of the region to be defined. **Default:** none
`YHI`	character	The tag specifying the bottom edge of the region to be defined. **Default:** none

Parameter	Type	Definition
`X`	number	x-coordinate of the point inside the region, the material of which should be changed. **Units:** microns **Default:** none
`Y`	number	y-coordinate of the point inside the region, the material of which should be changed **Units:** microns **Default:** none
`NEW.MATE`	character	New material of the region
`KEEP`	logical	This option, being defined, keeps impurity and defect concentrations for previous material, when it is replaced by the new one **Default:** false

Description

This statement specifies the material type of rectangular subregions when generating a rectangular mesh. `REGION` statements appear after `LINE` statements but before the `INITIALIZE` statement. The default material is silicon; if no `REGION` statements are specified, the entire initial structure is assumed to be silicon.

The tags referenced on `REGION` statements must be defined on the preceding `LINE` statements.

Redefining Material of the Region

Material of the specific region can be redefined. Region names are not defined during simulation, because regions can be created, removed or merged. So a region can be specified by three parameters: material of the region and X and Y coordinates of the point inside the region. New material is specified by the `NEW.MATE` statement. It should be one of predefined material names or a name of material, defined by the `MATERIAL` command before the `REGION` command. Option `KEEP`, being specified, keeps previous solution in the region. By default, all values are reset to the default for new material.

Examples

- The following statement defines the region bounded by the *x* coordinates `left` and `right` and the *y* coordinates `surf` and `back` to be silicon. See the description of the `LINE` statement on p. 4-57.

  ```
  REGION SILICON XLO=left XHI=right YLO=surf YHI=back
  ```

- The following statement redefines material of a region. Previous material was oxide, new material is oxynitride. Region is defined by it’s previous material and point inside the region.

  ```
  REGION OXIDE X=0.6 Y=1.2 NEW.MATE=OXYNITRIDE
  ```

Note:

> ***The* KEEP *option should be used carefully, because some impurities and defects from the previous material may not exist in the new material.***

INITIALIZE

The **INITIALIZE** statement sets up the initial structure for a simulation:

```
INITIALIZE

  {  ( IN.FILE=<c> [INTERMED] [ZIP.FILE]
                   {  ( [SCALE=<n>] [FLIP.Y] )
                    | ( TIF [ REGION=<c> NEW.MATE=<c> [KEEP] ] )
                   }
     )
   | ( [WIDTH=<n> [DX=<n>]] [ {<111> | <110> | <100> | ORIENTAT=<n>} ]
       [{ROT.SUB=<n> | X.ORIENT=<n>}] [RATIO=<n>] [LINE.DAT] [TEMPERAT=<n>]
     )
  }
  [ IMPURITY=<c> { I.CONC=<c> | I.RESIST=<n> } ]
  [MATERIAL=<c>] [ANTIMONY=<n>] [ARSENIC=<n>] [BORON=<n>] [PHOSPHOR=<n>]
  [ {CONCENTR | RESISTIV} ]
```

Parameter	Type	Definition
IN.FILE	character	Name of a saved structure file. If this parameter is omitted, a rectangular grid is generated, using previously specified **LINE**, **ELIMINATE**, **REGION**, and **BOUNDARY** statements. **Default**: none **Synonym**: **INFILE**
INTERMED	logical	Load intermediates **Default**: true
ZIP.FILE	logical	Specifies that the loaded file is in a compressed format **Default**: false
SCALE	number	The mesh read in from **IN.FILE** is scaled by this factor. **Units**: none **Default**: 1.0
FLIP.Y	logical	Specifies that the input structure is to be reflected about y=0. **Default**: false
TIF	logical	Specifies that the input file is a TIF (Technology Interchange Format) file. **Default**: false
REGION	character	Specifies a **REGION** in the structure, read from the TIF file, in which material should be redefined **Default**: false
NEW.MATE	character	Specifies new material in the region, which is defined by the **REGION** statement **Default**: false

Parameter	Type	Definition
KEEP	logical	This option, being defined, keeps impurity and defect concentrations for previous material, when it is replaced by the new one **Default**: false
WIDTH	number	The width of the initial structure. (Only used if no **LINE** X statements are specified.) **Units**: microns **Default**: the width of the **MASK** information, if any, or 1.0
DX	number	The grid spacing to use in the *x* direction. **Units**: microns **Default**: the current value of **DX.MAX** from the **MESH** statement
<111>	logical	Specifies that the crystalline orientation of the silicon substrate is <111>. **Default**: false
<110>	logical	Specifies that the crystalline orientation of the silicon substrate is <110>. **Default:** false
<100>	logical	Specifies that the crystalline orientation of the silicon substrate is <100>. **Default**: True, if no other orientation is specified.
ORIENTAT	number	The crystalline orientation of the silicon substrate. Only 100, 110, and 111 are recognized. **Units**: none **Default**: 100 **Synonym: Y.ORIENT**
ROT.SUB	number	The rotation of the substrate about the *y* axis. The reference orientation (**ROT.SUB=0**) is defined such that the *x* axis points in a <110> direction. Not allowed to be less than zero and bigger than 180 degrees. **Units**: degrees **Default**: 0.0
X.ORIENT	number	The crystalline orientation of the *x* axis. This must be a 3-digit integer value. **Units**: none **Default**: 110
RATIO	number	The maximum ratio of adjacent grid spacings to be used in generating a grid. **Units:** none **Default**: 1.5 **Synonym**: **INTERVAL**
LINE.DAT	logical	Specifies that the location of each *x* and *y* grid line be listed on the standard output and in the output listing file. **Default**: false
TEMPERAT	number	The initial temperature of the structure **Units**: degrees **Default**: 800

Parameter	Type	Definition
`IMPURITY`	character	The names of the impurities with which the initial structure is doped. If there are several impurities provided, their names should be enclosed in quotes and separated by the commas. **Default**: none
`I.CONC`	character	The concentrations of the specified impurities in the initial structure. If there is only one impurity provided, then its concentration will be interpreted as a floating number. For several impurities, a list of their concentrations should be provided in the same order as the list of impurity names in the **`IMPURITY`** parameter. The list should be enclosed in quotes and the concentrations should be separated by the commas. **Units**: atoms/cm^3 **Default**: none
`I.RESIST`	number	The resistivity of the initial structure. **Units**: ohm-cm **Default**: none
`MATERIAL`	character	Specifies the material of the initial structure. **Default**: **`SILICON`**
`ANTIMONY`	number	The uniform concentration or resistivity of antimony in the initial structure. **Units**: atoms/cm^3 or ohm-cm **Default**: 0.0 **Synonym**: **`SB`**
`ARSENIC`	number	The uniform concentration or resistivity of arsenic in the initial structure. **Units**: atoms/cm^3 or ohm-cm **Default**: 0.0 **Synonym**: **`AS`**
`BORON`	number	The uniform concentration or resistivity of boron in the initial structure. **Units**: atoms/cm^3 or ohm-cm **Default**: 0.0 **Synonym**: **`B`**
`PHOSPHOR`	number	The uniform concentration or resistivity of phosphorus in the initial structure. **Units**: atoms/cm^3 or ohm-cm **Default**: 0.0 **Synonym**: **`P`**
`CONCENTR`	logical	Specifies that the impurity concentration in the initial structure is given. **Default**: true
`RESISTIV`	logical	Specifies that the impurity resistivity in the initial structure is given. **Default**: false

Description

The **`INITIALIZE`** statement sets up the mesh from either a rectangular specification or from a previously saved structure file. This statement also initializes the background doping concentrations of the impurities specified.

Mesh Generation

If **IN.FILE** is not specified, a rectangular mesh is generated. If **LINE** statements have been specified for the *x* or *y* direction, they are used along with any **ELIMINATE**, **BOUNDARY**, and **REGION** statements and the value of the **RATIO** parameter to generate the mesh in that direction. If no **LINE** statements are specified for the *y* direction, a default *y* mesh is generated. If no **LINE** statements are specified for the *x* direction, and **WIDTH** is specified, a mesh of the requested width is generated, with spacing given by **DX**. If no **LINE X** statements are given and **WIDTH** is not specified, the width is taken from mask information read with a preceding **MASK** statement, if any. Otherwise, a one-micron wide structure with a grid spacing of one micron (i.e., one grid space wide) is generated. See Chapter 3, Grid Structure, p. 3-3 for a complete description of the mesh generation process. The locations of the generated grid lines are listed if **LINE.DAT** is true.

Previously Saved Structure Files

A mesh read from a file must be in either TSUPREM-4 format or Technology Interchange Format (TIF). Meshes read from TSUPREM-4 files can be scaled or flipped about *y*=0 by specifying the **SCALE** or **FLIP.Y** parameters, respectively.

When a structure is read from a file, the last processing temperature as well as the choice of silicon substrate orientation, oxidation model, and point defect model are automatically set through information stored in the file. You need not respecify these parameters after reading in a structure file at the start of a simulation.

Crystalline Orientation

The crystalline orientation of any silicon regions in a generated structure can be specified by <100>, <110>, or <111> parameters. The **ORIENTAT** parameter is also accepted, for compatibility with older versions of the program. The specified orientation is used for all single-crystal silicon regions in the simulation, whether present in the initial structure or deposited later. By default, the *x* axis points in a <110> direction. This can be changed by specifying a rotation about the *y* axis (**ROT.SUB**) or the crystal orientation of the *x* axis (**X.ORIENT**).

The orientation parameters do not apply when reading a structure from a file; the orientation of the saved structure is used instead.

Specifying Initial Doping

The **ANTIMONY**, **ARSENIC**, **BORON**, and **PHOSPHOR** parameters can be used to specify the initial resistivity or impurity concentrations in the structure. Any combination of these parameters can be specified if impurity concentrations are given (**CONCENTR** true), but only one impurity can be specified if the resistivity is given (**RESISTIV** true). Impurities can also be specified by name (in the case of a single impurity) or by a list of names (in the case of multiple impurities) with the **IMPURITY** parameter; **I.CONC** or **I.RESIST** are used to specify the concentrations of impurities or resistivity, respectively, associated with the named impurities.

If multiple impurities are specified, their concentrations should be specified as a list of numbers with **I.CONC** parameter. The order of concentrations in the list should follow the order of the impurity names, specified with the **IMPURITY** parameter

The **I.RESIST** parameter can be used only if a single impurity is specified by the **IMPURITY** parameter. The resistivity is calculated from tables of mobility as a function of doping concentration. These tables are described in Appendix A, Table A-55, p. A-21.

Although the source of the grid (read or generated) and the specification of doping are independent, the doping specification is normally used when a grid is generated but not when a grid is read from a file.

Redefining Material of the Region

If structure is loaded from the TIF file, you can redefine the material of the specific region. Specify the name of the region using the **REGION** statement. It should be a valid region name from the TIF file. Specify the name of material using the **NEW.MATE** statement. It should be one of predefined material names or a name of material, which is defined by the **MATERIAL** command before the **INITIALIZE** command. **REGION** and **NEW.MATE** statements can be used only along with **IN.FILE** and **TIF** statements. Option **KEEP**, being specified, keeps previous solution in the region. By default, all values are reset to the default for new material. See Note on p. 4-65.

Examples

- The following statement reads in a previously saved structure in file *oldstr*:

  ```
  INITIALIZE IN.FILE=oldstr
  ```

- The following statement generates a rectangular mesh and initializes the structure with a boron doping of 10^{15}/cm^3:

  ```
  INIT <111> X.ORIENT=211 BORON=1e15
  ```

  The orientation of single-crystal silicon regions are <111>, while the x axis points in a <211> direction.

- The following statement generates a mesh and initializes the structure to contain arsenic with resistivity of 20 ohm-cm:

  ```
  INIT IMPURITY=arsenic I.RESIST=20
  ```

- The following statement generates a mesh and initializes the structure to contain arsenic and boron with concentrations 10^{15}/cm^3 and 2×10^{15}/cm^3 respectively.

  ```
  INIT IMPUR="arsenic,boron" I.CONC="1e15,2e15"
  ```

- The following statement reads a previously saved structure from the TIF file *oldstr* and then redefines material of the region oxide_1. New material of this region is oxynitride.

  ```
  INIT IN.FILE=oldstr TIF +
  REGION=oxide_1 NEW.MATERIAL=oxynitride
  ```

Note:

> ***The conversion from a resistivity to a concentration is based on Masetti's mobility table, while the calculation of electrical characteristics in the* ELECTRICAL *statement uses the same mobility table as in Medici. Thus, the sheet resistances of the initial structure given by the* EXTRACT *statement do not correspond exactly to the resistivity specified on the* INITIALIZE *statement.***

LOADFILE

The **`LOADFILE`** statement reads mesh and solution information from a file.

```
LOADFILE

   IN.FILE=<c> [INTERMED] [ZIP.FILE]
   {  ( [SCALE=<n>] [FLIP.Y] )
    | ( TIF
          [ REGION=<c> NEW.MATE=<c> [KEEP] ] )
   }
```

Parameter	Type	Definition
`IN.FILE`	character	The identifier of a structure file to be read. **Default:** none **Synonyms**: **`INFILE`**
`INTERMED`	logical	Load intermediates. **Default**: true
`ZIP.FILE`	logical	Specifies that the loaded file is in a compressed format. **Default**: false
`SCALE`	number	A scaling factor to be applied to the mesh when reading TSUPREM-4 structure files. All coordinate values are multiplied by this factor as they are read. **Units:** none **Default:** 1.0
`FLIP.Y`	logical	Specifies that the structure should be reflected about *y*=0 when reading TSUPREM-4 structure files. All *y* coordinates are multiplied by –1 as they are read. **Default:** false
`TIF`	logical	Specifies that the input file is a Technology Interchange Format (TIF) file. **Default:** false
`REGION`	character	Specifies a region of the structure from the TIF file in which material should be redefined **Default:** false
`NEW.MATE`	character	Specifies new material in the region, which is defined by the **`REGION`** statement **Default:** false

Description

The **`LOADFILE`** statement reads a mesh and solution from a file. Either TSUPREM-4 or TIF structure files (created with the **`SAVEFILE`** statement) can be read. TSUPREM-4 structure files can be scaled or flipped about the *x* axis (*y*=0) during reading.

TSUPREM-4 Files

The silicon substrate orientation, last processing temperature, current oxidation model, and current point defect model are saved in TSUPREM-4 structure files, and are automatically restored when the file is read. There is no need to respecify these parameters after reading in a structure. (This does not apply to versions of TSUPREM-4 before Version 9035; versions older than 9035 only saved the substrate orientation.)

Older Versions

Files saved with older versions of TSUPREM-4 can be loaded by newer versions of the program. Files saved with Version 5.1 and newer of TSUPREM-4 cannot be read by older versions of the program.

In versions of TSUPREM-4 prior to Version 5.1, the **STRUCTURE** statement is used to load structure files. Starting with Version 5.1, the **LOADFILE** statement should be used instead. The **IN.FILE**, **SCALE**, and **FLIP.Y** parameters are still accepted on the **STRUCTURE** statement, however, for compatibility with older TSUPREM-4 input files.

User-Defined Materials and Impurities

Any user-defined materials and impurities referenced in a structure file should be defined before the file is loaded. If a file containing undefined materials or impurities is loaded, a warning is issued for each undefined material or impurity encountered. The material or impurity is defined, but its properties are not set. Further simulation using the material or impurity may fail unless the properties are set.

Replacing Materials of Regions

When the structure is loaded from a TIF file, the material of the region can be replaced with the other material. The parameter **REGION** specifies the name of the region in which the material is to be replaced with the other material specified by the **NEW.MATE** parameter. The optional parameter **KEEP** specifies that the solution values in the region will be kept even though the material is replaced with the other one. By default, all solution values are reset to the default values in the material. See Note on p. 4-65.

Examples

- The following statement reads in a previously saved structure from file *savestr*:

  ```
  LOADFILE IN.FILE=savestr
  ```
- The statements:

  ```
  LOADFILE IN.FILE=savestr
  ```

  and:

  ```
  INITIALIZE IN.FILE=savestr
  ```

  are equivalent, except that the program recognizes that an initial structure has been set up in the second case. An **INITIALIZE** statement must be given before any processing or output statement can be processed.

- The following statement reads a previously saved structure TIF file *oldstr* and then redefines material of the region oxide_1. New material of this region is oxynitride.

```
LOADFILE IN.FILE=oldstr TIF +
        REGION=oxide_1 NEW.MATE=oxynitride
```

SAVEFILE

The **`SAVEFILE`** statement writes mesh and solution information to a file.

```
SAVEFILE

    OUT.FILE=<c> [TEMPERAT=<n>] [INTERMED] [ZIP.FILE]
    {  ( [SCALE=<n>] [FLIP.Y] [ACTIVE] )
     | (TIF [TIF.VERS=<c>])
     | ( MEDICI  [POLY.ELE] [ELEC.BOT] ] )
     | ( MINIMOS5  X.MASK.S=<n>
         { HALF.DEV | ( FULL.DEV  X.MASK.D=<n> [X.CHANNE=<n>] ) }
          [X.MIN=<n>] [X.MAX=<n>] [Y.MIN=<n>] [Y.MAX=<n>]
          [DX.MIN=<n>] [DY.MIN=<n>]
       )
     | ( WAVE  [ACTIVE] [CHEMICAL] [DEFECT] [OXID] [MISC] )
    }
```

Parameter	Type	Definition
`OUT.FILE`	character	The identifier of the structure file to be written. **Default:** none **Synonym: `OUTFILE`**
`TEMPERAT`	number	The temperature used for evaluating active impurity concentrations. This feature is not recommended for applications involving advanced diffusion, activation and stress models. **Units:**° Celsius **Default:** the last processing temperature specified or 800° C
`INTERMED`	logical	Save intermediates. **Default**: false
`ZIP.FILE`	logical	Specifies to compress the saved file. **Default**: false
`SCALE`	number	A scaling factor to be applied to the mesh when writing TSUPREM-4 structure files. All coordinate values saved in the file are multiplied by this value. The **`SCALE`** parameter does not affect the structure used by subsequent simulation steps. **Units:** none **Default:** 1.0
`FLIP.Y`	logical	Specifies that the structure should be reflected about *y*=0 when writing TSUPREM-4 structure files. All *y* coordinates are multiplied by -1 as they are written to the file. **`FLIP.Y`** does not affect the structure used by subsequent simulation steps. **Default:** false

Parameter	Type	Definition
`TIF`	logical	Specifies that the output file be saved as a Technology Interchange Format (TIF) file. **Default:** false
`TIF.VERS`	character	The version of TIF to be used for saving the file. The default is to use the latest version of TIF; a value of 0 produces files compatible with Version 6.0 of TSUPREM-4. **Units:** none **Default:** 1.2.0
`MEDICI`	logical	Specifies that the saved output file is a formatted file that can be read by the Medici device simulator. The output file can also be read by older versions of TMA PISCES-2B and by other versions of PISCES. **Default:** false **Synonym**: **`PISCES`**
`POLY.ELE`	logical	Specifies that polysilicon regions should be converted to electrodes in the Medici output file. **Default:** false
`ELEC.BOT`	logical	Specifies that an electrode should be placed along the backside of the structure in the Medici output file. **Default:** false
`MINIMOS5`	logical	Specifies that the saved output file contains a 2D doping profile that can be read by MINIMOS 5. **Default:** false
`X.MASK.S`	number	The *x* coordinate of the mask edge in the source area of the MINIMOS 5 simulation region. MINIMOS 5 interprets this coordinate as the left edge of the gate electrode. **Units:** microns **Default:** none
`HALF.DEV`	logical	Specifies that the MINIMOS 5 simulation region includes only the source area of the device. Either **`FULL.DEV`** or **`HALF.DEV`** must be specified if **`MINIMOS5`** is specified. **Default:** false
`FULL.DEV`	logical	Specifies that the MINIMOS 5 simulation region includes both the source and drain areas of the device. Either **`FULL.DEV`** or **`HALF.DEV`** must be specified if **`MINIMOS5`** is specified. **Default:** false
`X.MASK.D`	number	The *x* coordinate of the mask edge in the drain area of the MINIMOS 5 simulation region. MINIMOS 5 interprets this coordinate as the right edge of the gate electrode. **`X.MASK.D`** must be specified if **`FULL.DEV`** is specified; it must not be specified if **`HALF.DEV`** is specified. **Units:** microns **Default:** none

Parameter	Type	Definition
X.CHANNE	number	The *x* coordinate of the center of the channel of the MINIMOS 5 simulation region. **Units:** microns **Default:** (**X.MIN** + **X.MAX**)/2 if **FULL.DEV** is specified; not applicable if **HALF.DEV** is specified
X.MIN	number	The *x* coordinate of the left edge of the MINIMOS 5 simulation region. **Units:** microns **Default:** left edge of the TSUPREM-4 simulation region
X.MAX	number	The *x* coordinate of the right edge of the MINIMOS 5 simulation region. **Units:** microns **Default:** right edge of the TSUPREM-4 simulation region
Y.MIN	number	The *y* coordinate of the top edge of the MINIMOS 5 simulation region. MINIMOS 5 interprets this coordinate as the gate oxide/silicon interface. **Units:** microns **Default:** 0.0
Y.MAX	number	The *y* coordinate of the bottom edge of the MINIMOS 5 simulation region. **Units:** microns **Default:** bottom edge of the TSUPREM-4 simulation region
DX.MIN	number	The minimum spacing in the *x* direction used to specify the doping profiles in the output file. **Units:** microns **Default:** min((**X.MAX**-**X.MIN**)/80, 0.01)
DY.MIN	number	The minimum spacing in the *y* direction used to specify the doping profiles in the output file. **Units:** microns **Default:** min((**Y.MAX**-**Y.MIN**)/80, 0.01)
WAVE	logical	Specifies that the output file is a formatted file in Wavefront Technologies' *wave* file format. These files can be read by Wavefront Technologies' Data Visualizer program. **Default:** false
ACTIVE	logical	Specifies that active impurity concentrations are to be saved in TSUPREM-4 and **WAVE** output files. **Default:** true
CHEMICAL	logical	Specifies that chemical impurity concentrations are to be included in the **WAVE** output file. **Default:** false
DEFECT	logical	Specifies that point defect concentrations are to be included in the **WAVE** output file. **Default:** false

Parameter	Type	Definition
OXID	logical	Specifies that oxidant concentrations, oxidation flow rates, and stresses (if available) are to be included in the **WAVE** output file. **Default:** false Synonym: **STRESS**
MISC	logical	Specifies that miscellaneous solution values are to be included in the **WAVE** output file. At present, this includes diffusivities of impurities and point defects. **Default:** false

Description

The **SAVEFILE** statement writes mesh and solution information into a file, in one of several formats. If no format is specified, a TSUPREM-4 structure file is created. TSUPREM-4 structure files can be read with the **LOADFILE** or **INITIALIZE** statements. The mesh can be scaled or flipped about the x-axis as the structure is written or when it is read. Scaling and flipping during writing affect only the saved structure, and do not affect the structure used by subsequent simulation steps.

TSUPREM-4 Files

The silicon substrate orientation, last processing temperature, current oxidation model, and current point defect model are saved in TSUPREM-4 structure files, and are restored automatically when the file is read. There is no need to respecify these parameters after reading in a structure. (This does not apply to versions of TSUPREM-4 before Version 9035; versions older than 9035 only saved the substrate orientation.)

Older Versions

Files saved with older versions of TSUPREM-4 can be loaded by newer versions of the program. Files saved with Version 5.1 and newer of TSUPREM-4 cannot be read by older versions of the program.

In versions of TSUPREM-4 before Version 5.1, the **STRUCTURE** statement is used to load structure files. Starting with Version 5.1, the **LOADFILE** statement should be used instead. The **IN.FILE**, **SCALE**, **FLIP.Y**, and **DEPICT** parameters are still accepted on the **STRUCTURE** statement, however, for compatibility with older TSUPREM-4 input files.

Effective in Version 5.2 of TSUPREM-4, active impurity concentrations are saved by default in TSUPREM-4 structure files. Structure files without the active impurity concentrations can be produced by specifying **^ACTIVE** on the **SAVEFILE** statement. This is necessary if the structure files are to be read by older (before 5.2) versions of TSUPREM-4 or by other programs that cannot accept the active concentration information.

TIF Files

The **TIF** parameter specifies that the file should be saved as a Technology Interchange Format (TIF) file. The version of TIF can be specified with the **TIF.VERS** parameter; newer products use Version 1.2.0, while older products (including Version 6.0 of TSUPREM-4) use Version 1.00 or Version 0 (which are equivalent, as far as TSUPREM-4 is concerned).

Correct writing of a user-defined material or impurity to a TIF file requires that a **TIF.NAME** be specified when the material or impurity is defined; **MD.INDEX** also must be specified for materials. Before other programs can read the saved TIF file, an entry corresponding to the TIF name must be added to the appropriate database (*mat.dbs* for materials, or *sol.dbs* for impurities).

Note:

> ***Versions 6.1 and later of*** TSUPREM-4 ***can read TIF files created by Version 6.0, but Version 6.0 cannot read TIF files created by Versions 6.1 and later unless*** **TIF.VERS =0** ***is specified when the file is written.***

Medici Files

The **MEDICI** parameter creates an output file that can be read by the Medici device simulator. **MEDICI** structures can also be read by older versions of TMA pisces-2b and by other versions of pisces. The **POLY.ELE** and **ELEC.BOT** parameters are not needed when creating files for *Synopsys* TCAD's device simulators, because these simulators allow the treatment of polysilicon and backside contacts to be specified an a **MESH** statement.

Correct writing of a user-defined material to a Medici file requires that **MD.INDEX** be specified when the material is defined (see MATERIAL, p. 4-258).

MINIMOS

The **MINIMOS5** parameter creates an output file that can be read by MINIMOS 5, a 2D program for the simulation of MOSFETs and MESFETs. The output file contains a 2D doping profile. Detailed instructions on interfacing TSUPREM-4 to MINIMOS 5 are given in Appendix E.

Temperature

The final temperature of the last high-temperature processing step is needed to calculate active impurity concentrations. Normally, the value at the end of the last diffusion or epitaxy step or the value read when a structure is loaded is used, but if **TEMPERAT** is specified, its value is used instead. The value of **TEMPERAT** (if specified) is retained for use in subsequent input statements. The last temperature value is saved in TSUPREM-4 structure files and is restored when the structure is loaded with a **LOADFILE** or **INITIALIZE** statement.

Examples

- The following statement saves a structure in the file *savestr*:

```
SAVEFILE OUT.FILE=savestr
```

- The following statement saves the structure in file *PIOUTSTR*, in a format that can be read by Medici:

```
SAVEFILE OUT.FILE=PIOUTSTR MEDICI
```

STRUCTURE

The **STRUCTURE** statement reflects, truncates, or extends the current structure.

```
STRUCTURE

  [ TRUNCATE {( {RIGHT | LEFT} X=<n> ) | ( {BOTTOM | TOP} Y=<n> )} ]
  [ REFLECT [ {RIGHT | LEFT} ] ]
  [ EXTEND  [ {RIGHT | LEFT} ]  WIDTH=<n>
     [SPACES=<n>]  [DX=<n>]  [XDX=<n>]  [Y.ELIM=<c>] ]
  [ UNREFINE [REPEAT=<n>] [ROWS] [COLUMNS]
     [X.MIN=<n>] [X.MAX=<n>] [Y.MIN=<n>] [Y.MAX=<n>] ]
  [TEMPERAT=<n>]
```

Parameter	Type	Definition
TRUNCATE	logical	Causes the current mesh to be truncated according to the values of the **RIGHT**, **LEFT**, **BOTTOM**, **TOP**, **X**, and **Y** parameters. The boundary condition is set to reflecting along the new surface created by the truncation. **Default:** false
RIGHT	logical	Specifies that truncation, reflection, or extension occurs at the right edge of the structure. **Default:** true unless **LEFT**, **BOTTOM**, or **TOP** is specified
LEFT	logical	Specifies that truncation, reflection, or extension occurs at the left edge of the structure. **Default:** false
X	number	The *x* coordinate where truncation occurs. If **LEFT** is specified, truncation occurs to the left of this location; if **RIGHT** is specified, truncation occurs to the right of this location. Only valid if **TRUNCATE** and **RIGHT** or **LEFT** are specified. **Units:** microns **Default:** none
BOTTOM	logical	Specifies that truncation occurs at the bottom edge of the structure. **Default:** false
TOP	logical	Specifies that truncation occurs at the top edge of the structure. **Default:** false
Y	number	The *y* coordinate where truncation occurs. If **BOTTOM** is specified, the portion of the structure below this location (higher values of *y*) is truncated; if **TOP** is specified, the portion of the structure above this location (lower values of *y*) is truncated. Only valid if **TRUNCATE** and **BOTTOM** or **TOP** are also specified. **Units:** microns **Default:** none

Parameter	Type	Definition
`REFLECT`	logical	Causes the current mesh to be reflected at its left or right edge, doubling the width of the structure. **Default:** false
`EXTEND`	logical	Causes the current mesh to be extended from its left or right edge by the specified width. **Default:** false
`WIDTH`	number	The width by which the structure is extended. **Units:** microns **Default:** none
`SPACES`	number	The number of grid spaces to be added in the *x* direction when extending the structure. **Units:** none **Default:** 1
`DX`	number	The nominal grid spacing in the *x* direction in the extension of the structure. **Units:** microns **Default:** the width of the extension/**`SPACES`**
`XDX`	number	The absolute *x* location at which the grid spacing in the extension is equal to **`DX`**. **Units:** microns **Default:** the location of the new edge of the extended structure
`Y.ELIM`	character	A list of 10 or fewer *y* locations, separated by spaces or commas, below which vertical grid lines are eliminated in the extended portion of the structure. If spaces appear in the list, the entire list must be enclosed in quotation marks. **Units:** microns **Default:** no elimination is done
`UNREFINE`	logical	Causes one or more mesh unrefinement steps to be performed. **Default:** false
`REPEAT`	number	Specifies the maximum number of unrefinement steps to be performed. **Units:** none **Default:** 1
`ROWS`	logical	Specifies that rows of points in the undoped (or background doped) substrate may be removed during unrefinement. **Default:** false
`COLUMNS`	logical	Specifies that columns of points in the undoped (or background doped) substrate may be removed during unrefinement. **Default:** false
`X.MIN`	number	The *x* coordinate of the left edge of the region in which unrefinement is allowed. **Units:** microns **Default:** the left edge of the structure

Parameter	Type	Definition
`X.MAX`	number	The *x* coordinate of the right edge of the region in which unrefinement is allowed. **Units:** microns **Default:** the right edge of the structure
`Y.MIN`	number	The *y* coordinate of the top of the region in which unrefinement is allowed. **Units:** microns **Default:** the top of the structure
`Y.MAX`	number	The *y* coordinate of the bottom of the region in which unrefinement is allowed. **Units:** microns **Default:** the bottom of the structure
`TEMPERAT`	number	Specifies the last processing temperature to be used for evaluating active impurity concentrations. **Units:** ° Celsius **Default:** the last processing temperature specified, or 800° C

Description

The **`STRUCTURE`** statement is used to reflect, truncate, extend, or unrefine the current structure. The structure can be truncated on the left or right or at the bottom or top; reflected about its left or right edge, doubling its width; or extended from its left or right edge to the specified width. Truncation occurs before reflection, and reflection occurs before extension. Unrefinement occurs after truncation and extension. The last processing temperature can also be set for use in calculating active impurity concentrations for the **`SELECT`** or **`SAVEFILE`** statements.

Reflecting a structure also reflects any mask information associated with the structure. The mask information is truncated or extended (if necessary) to the value specified by **`X`**, then reflected.

The **`SPACES`**, **`DX`**, and **`XDX`** parameters specify the grid when a structure is extended. These parameters are used in the same way as the **`THICKNES`**, **`DY`**, and **`YDY`** parameters are used in the **`DEPOSITION`** and **`EPITAXY`** statements. The use of these parameters is described in Chapter 3, Changes to the Mesh During Processing, p. 3-7.

Note:

> ***The `EXTEND` feature should be used with caution. Simulating a narrow structure and then extending it does not give the same results as simulation of the wider structure unless the topography and doping concentrations at the edge of the narrow structure are essentially one-dimensional.***

Order of Operations

The operations specified on the **`STRUCTURE`** statement are performed in the following order:

1. Set the last processing temperature, if specified with the **`TEMPERAT`** parameter.
2. Truncate the structure (if **`TRUNCATE`** is specified).
3. Reflect the structure (if **`REFLECT`** is specified).
4. Extend the structure (if **`EXTEND`** is specified).
5. Unrefine the structure (if **`UNREFINE`** is specified).

Usage

Reflecting the structure about an edge that is not absolutely vertical can result in very thin triangles that could cause numerical problems. A structure should only be reflected about a reflecting boundary, created during an initial structure specification or by the **`TRUNCATE`** parameter.

`TRUNCATE TOP` can be used to remove the entire exposed surface, giving a structure with reflecting boundary conditions on all four sides. The usefulness of such structures is limited because some process steps (e.g., **`DEPOSITION`** and **`ETCH`**) do not work on structures that have no exposed surface.

The **`UNREFINE`** parameter is used to remove unneeded mesh points from the structure. If **`REPEAT`** is more than 1, the unrefinement is repeated the specified number of times or until no further points can be removed. **`ROWS`** and **`COLUMNS`** allow removal of rows and columns of points in the undoped (or background doped) substrate. If both **`ROWS`** and **`COLUMNS`** are specified, rows and columns will be removed in alternating unrefinement steps. The **`X.MIN`**, **`X.MAX`**, **`Y.MIN`**, and **`Y.MAX`** parameters can be used to limit unrefinement to a portion of the structure.

TSUPREM-4 Version Compatibility

In versions of TSUPREM-4 before Version 5.1, the **`STRUCTURE`** statement is used to save and load structure files. Starting with Version 5.1, the **`SAVEFILE`** and **`LOADFILE`** statements should be used instead. Starting with Version 2000.2, the parameters for saving TSUPREM-4, Taurus Lithography, Medici, and minimos 5 files and for loading TSUPREM-4 and Taurus Lithography files are no longer accepted on the **`STRUCTURE`** statement.

Examples

1. The following statement truncates the structure to the right of *x*=1.2 microns, then mirrors the structure about the new right edge:

```
STRUCTURE TRUNCATE RIGHT X=1.2 REFLECT
```

2. The following statement extends the structure to the right by 0.5 microns. Two grid spaces are added, with elimination of every other vertical line below *y*=1 µm:

```
STRUCTURE EXTEND RIGHT WIDTH=0.5 +
      SPACES=2 Y.ELIM=1.0
```

3. The following statement causes up to four unrefinement steps to be performed:

   ```
   STRUCTURE UNREFINE REPEAT=4 ROWS COLUMNS
   ```

   The first and third steps allow rows of points to be removed in the substrate while the second and fourth steps allow columns to be removed.

MASK

The **`MASK`** statement reads masking information from a file.

```
MASK

   [IN.FILE=<c>[SCALE=<n>] [GRID=<c>] [G.EXTENT=<c>]]   [PRINT]
```

Parameter	Type	Definition
`IN.FILE`	character	Specifies a file containing mask information. **Default:** none
`SCALE`	number	Specifies a scale factor to be applied to all dimensions and locations in the mask file. **Units:** none **Default:** 1.0
`GRID`	character	One or more names of mask levels (separated by spaces or commas) for which fine horizontal grid is to be used during automatic grid generation. **Default:** all mask levels used
`G.EXTENT`	character	The distance that fine grid spacing extends under the lines on a mask. If a single value is specified, it is used for all mask levels specified with the **`GRID`** parameter (or all mask levels if **`GRID`** is not specified). If multiple values (separated by spaces or commas) are specified, then each value corresponds to a mask level specified with the **`GRID`** parameter. Negative values specify that fine grid extends into the spaces between the mask lines. **Units:** microns **Default:** 0.0
`PRINT`	logical	Prints mask information on the standard output and in the output listing file. The listing includes the names of all masks and the minimum and maximum *x* locations of each opaque segment of each mask. **Default:** false

Description

The **`MASK`** statement reads a file describing the mask levels used in a process. This mask information is used by the **`EXPOSE`** statement for patterning photoresist, which can be used as a masking layer for subsequent processing. By using mask files, the process description can be separated from the layout information, making it easy to simulate multiple layouts with a single process or multiple processes with a given layout.

The **`SCALE`** parameter specifies a scale factor to be applied to all dimensions and locations in the mask file.

The **`GRID`** and **`G.EXTENT`** parameters control how mask information is used to set the horizontal grid spacing during automatic grid generation. The **`GRID`** parameter specifies which mask levels are used. Fine grid spacing is used at the edges of lines on the specified levels. The **G.EXTENT** parameter determines how far the fine grid extends under the lines of the mask. The **GRID** and **G.EXTENT** parameters are character strings that specify lists of values; be sure to enclose them in quotation marks if they contain spaces.

TSUPREM-4 input mask files are created by Taurus Layout. Taurus Layout lets you view a mask layout file and interactively select regions to be simulated.

Examples

1. The following statement reads mask information from the file *HVNCH.TL1* then prints the information that is read:

```
MASK  IN.FILE=HVNCH.TL1 PRINT
```

2. The following statement specifies that only the *POLY* and *FIELD* masks should be considered when doing automatic grid generation. Fine grid should extend 0.5 microns under the edges of lines on the *POLY* mask and 0.3 microns under the edges of lines on the *FIELD* mask:

```
MASK IN.FILE=S4EX4M.TL1 GRID=”Poly,Field”+
     G.EXTENT="0.5,0.3"
```

3. TSUPREM-4 input files can be structured to separate the masking information from the processing information:

```
$ MESH DESCRIPTION (DEPENDS ON MASKS)
LINE  X  LOC=0.0  SPAC=0.01
  .
  .
  .
LINE  Y  LOC=200.0 SPAC=50.0
INITIALIZE  BORON=1E14
$ MASK INFORMATION
MASK  IN.FILE=MINNCH.TL1
$ PROCESS INFORMATION (DOES NOT DEPEND ON MASKS)
SOURCE  CMOS07
```

To simulate a different portion of a layout with the same process, you only need to change the mesh description and the name of the **MASK** file. To simulate the same device with a modified process, you only need to change the name of the **SOURCE** file.

PROFILE

The **`PROFILE`** statement allows a 1D impurity profile to be read from a file.

```
PROFILE

   { IMPURITY=<c> | ALL.IMP | ANTIMONY | ARSENIC | BORON |
     PHOSPHOR }
   [MATERIAL=<c>]
   [X.MIN=<n>] [X.MAX=<n>] [Y.MIN=<n>] [Y.MAX=<n>]
   [REPLACE] [EQ.RESET]
   {  IN.FILE=<c> [ZIP.FILE] [I.SCALE=<n>] [ { NM | UM | CM } ]
        { ( ONE.DIM [OFFSET=<n>] [SIGMA=<n>] )
         |( TWO.DIM [X.OFFSET=<n>] [Y.OFFSET=<n>] [MIN.WT=<n>]
                    [I.SOURCE=<c>] [{ I.COLUMN=<c> | TIF | TS4 }])
        }
    | EXPRESS=<c>
   }
```

Parameter	Type	Definition
`IMPURITY`	character	The name of the impurity for **`ONE.DIM`** or the name list of impurities for **`TWO.DIM`** to be read from the input file, or the impurity to be generated by the user-defined expression **`EXPRESS`**. **Default:** none
`ALL.IMP`	logical	Specifies to read all impurities from the **`IN.FILE`** file. Only applicable when the format of the **`IN.FILE`** file is in either TIF or TSUPREM-4 structure format. **Default:** false
`ANTIMONY`	logical	Specifies that the input file contains a profile of antimony concentration. **Default:** false **Synonym:** **`SB`**
`ARSENIC`	logical	Specifies that the input file contains a profile of arsenic concentration. **Default:** false **Synonym:** **`AS`**
`BORON`	logical	Specifies that the input file contains a profile of boron concentration. **Default:** false **Synonym:** **`B`**
`PHOSPHOR`	logical	Specifies that the input file contains a profile of phosphorus concentration. **Default:** false **Synonym:** **`P`**
`MATERIAL`	character	The name of the material into which data are to be imported. Only applicable for the import by **`TWO.DIM`** and **`EXPRESS`** **Default:** silicon

Parameter	Type	Definition
`REPLACE`	logical	Specifies that the profile read from the file should replace the existing profile (rather than adding to it). **Default:** false
`EQ.RESET`	logical	Specifies that the equilibrium state is automatically recalculated according to the change of solutions by importing the profiles. **Default:** true
`X.MIN`	number	Specifies the left boundary of the rectangular region into which the profile is imported. **Units:** microns or cm or nm, which is determined by **`UM`** or **`CM`** or **`NM`** **Default:** leftmost location of the device
`X.MAX`	number	Specifies the right boundary of the rectangular region into which the profile is imported. **Units:** microns or cm or nm, which is determined by **`UM`** or **`CM`** or **`NM`** **Default:** rightmost location of the device
`Y.MIN`	number	Specifies the top boundary of the rectangular region into which the profile is imported. **Units:** microns or cm or nm, which is determined by **`UM`** or **`CM`** or **`NM`** **Default:** top location of the device
`Y.MAX`	number	Specifies the bottom boundary of the rectangular region into which the profile is imported. **Units:** microns or cm or nm, which is determined by **`UM`** or **`CM`** or **`NM`** **Default:** bottom location of the device
`IN.FILE`	character	The name of the data file containing the 1D or 2D profile information. **Default:** none **Synonym: `INFILE`**
`ZIP.FILE`	logical	Specifies that the **`IN.FILE`** file is in compressed format. **Default:** false
`I.SCALE`	character	The list of the scaling number which corresponds to the list of impurities in the **`IMPURITY`** parameter. Scales the data when reading from the **`IN.FILE`** file. **Default:** 1 for one impurity, "1,1,1..." for multiple impurities
`NM`	logical	Specifies that the distance unit of data and the unit of **`X.MIN`**, **`X.MAX`**, **`Y.MIN`**, **`Y.MAX`**, **`OFFSET`**, **`X.OFFSET`** and **`Y.OFFSET`** are in nanometers. **Default:** false
`UM`	logical	Specifies that the distance unit of data and the unit of **`X.MIN`**, **`X.MAX`**, **`Y.MIN`**, **`Y.MAX`**, **`OFFSET`**, **`X.OFFSET`** and **`Y.OFFSET`** are in microns. **Default:** true
`CM`	logical	Specifies that the distance unit of data and the unit of **`X.MIN`**, **`X.MAX`**, **`Y.MIN`**, **`Y.MAX`**, **`OFFSET`**, **`X.OFFSET`** and **`Y.OFFSET`** are in centimeters. **Default:** false
`ONE.DIM`	logical	Specifies that the input file contains a 1D impurity profile **Default:** true

Parameter	Type	Definition
OFFSET	number	The amount by which the profile is shifted in the *y*-direction before being applied to the TSUPREM-4 structure. The concentration at *y*=0 in the data file is added at *y*=**OFFSET** in the TSUPREM-4 structure. **Units:** nanometers or microns or centimeters **Default:** 0
SIGMA	number	Parameter of the erfc function, which is used for extrapolation of the profile outside of the specified rectangular region **Units:** microns
TWO.DIM	logical	Specifies that the input file contains a 2D impurity profile. **Default:** false
X.OFFSET	number	The amount by which the profile is shifted in the *x*-direction before being applied to the TSUPREM-4 structure. Only applicable for the 2D profile. **Units:** nanometers or microns or centimeters **Default:** 0
Y.OFFSET	number	The amount by which the profile is shifted in the *y*-direction before being applied to the TSUPREM-4 structure. Only applicable for the 2D profile. **Units:** nanometers or microns or centimeters **Default:** 0
MIN.WT	number	Minimum weight criteria not to discard the data out of material regions. **Units:** none **Default:** 0.001
I.SOURCE	character	The list of solution names to be read. Each name must correspond to the one in the **IMPURITY** list. The name must match exactly one in the **IN.FILE** file. Putting the character '-' instead of a solution name automatically chooseS the proper name corresponding the **IMPURITY** name. Only applicable for TIF and TSUPREM-4 structure format files. **Default:** "-,-,-,..."
I.COLUMN	character	The column list which corresponds to the list of impurities in the **IMPURITY** parameter. **Default:** 3 for one impurity, "3,4,5..." for multiple impurities
TIF	logical	Specifies that the **IN.FILE** file format is TIF. **Default:** false
TS4	logical	Specifies that the **IN.FILE** file format is TSUPREM-4 structure file format. **Default:** false
EXPRESS	character	The expression to generate the impurity profile to be imported. **Default:** none

Description

This statement allows a 1D impurity profile (or other solution variable) to be read from a data file. The data file must be a text file containing two columns of numbers. The first column gives the *y* coordinate of a point, in microns, and the second contains the impurity concentration (or other solution value), in appropriate units. Lines in the data file that do not contain exactly two recognizable numeric values are ignored.

If **REPLACE** is specified, the profile read from the file replaces any existing profile. Replacement occurs only at nodes whose *y* coordinates are within the range of *y* coordinates given in the file. Adaptive gridding does not apply to the **PROFILE** statement. You must insure that the grid is dense enough to resolve the profile.

OFFSET Parameter

The profile read from the data file is expanded in the *x* direction to the width of the structure and added to any existing impurities in the structure; negative data values can be used to reduce the concentration. The profile can be shifted in the *y* direction by specifying the **OFFSET** parameter. The profile is shifted by **OFFSET**, so that the concentration at *y*=0 in the data file appears at location *y*=**OFFSET** in the TSUPREM-4 structure.

Interpolation

Values between the *y* coordinates specified in the input file are calculated using linear interpolation on the function

$$f(c) = \operatorname{sign}(c)\log\left(\frac{|c|}{2k} + \sqrt{\left(\frac{c}{2k}\right)^2 + 1}\right) \qquad \text{Equation 4-1}$$

where:

- c is the concentration in the file.
- k is a scaling factor.

The final value of c at each point is found by inverting $f(c)$ using the value of f obtained from the linear interpolation. This gives logarithmic interpolation for $|c|$ » k and linear interpolation for $|c|$ « k. The value of k is taken to be 10^5 for impurity and point defect concentrations and 1.0 for other solution variables (e.g., *dloop* and *rloop*). At points outside the range of values in the input file, the impurity concentration remains unchanged.

IMPURITY Parameter

The **IMPURITY** parameter can be used to read in values for a user-specified impurity. This can be used to read in a profile for comparison without altering the desired doping:

```
IMPLANT BORON ...
SELECT Z=LOG10(BORON)
PLOT.1D ...
IMPURITY NEW IMPURITY=PBORON
PROFILE IMPURITY=PBORON ...
SELECT Z=LOG10(PBORON)
PLOT.1D ^AX ^CL ...
```

If *boron* had been specified on the **PROFILE** statement, the profile would have been added to the implanted boron, making it impossible to compare the implanted profile with the contents of the data file.

The **IMPURITY** parameter can be used to read in values of interstitial and vacancy profiles. Remember that the profiles read from the file are added to the profiles already in the structure, unless **REPLACE** is specified.

Profile in 2D Rectangular Region

Profile can be specified in 2D rectangular region inside the device. In this case impurity concentration is extrapolated by the *erfc* function outside the region:

$$N(x,\ y)\ =\ N(x)\frac{1}{4}\left[erfc\left(\frac{x-x_{max}}{\sigma}\right)-erfc\left(\frac{x-x_{min}}{\sigma}\right)\right]\left[erfc\left(\frac{y-y_{max}}{\sigma}\right)-erfc\left(\frac{y-y_{min}}{\sigma}\right)\right],$$

where *N(x)* is initial 1D profile.

Importing Columwise 2D Profiles

2D impurity profiles which are written in the following format can be imported into the structure.

```
/x(um) y(um) C1(col=3) C2(col=4) C3(col=5) ... Cn
0.0    0.1   1.2E21    3.8E19    8.4E17    ... 5.4E20
0.1    0.2   1.1E21    2.7E19    1.7E18    ... 9.2E19
:
```

The line starting with '/' is ignored. The first and second columns are always the x-axis and y-axis coordinates in microns, respectively. And the data of the impurity species to be imported are defined by one of the following specifications:

- Either one of **BORON**, **PHOSPHOR**, **ARSENIC** and **ANTIMONY**, or **IMPURITY**=*impurity_name* with the optional specification of **I.COLUMN**(default:3) when the data for one impurity are imported.
- **IMPURITY="***list_of_impurities***"** with the optional specification of **I.COLUMN**(default:"3,4,5...") when the data of multiple impurities are imported.

The **MATERIAL** parameter(default:silicon) specifies the material region into which the data will be imported.

Note:

The data which are located in the outside of the structure and the rectangle, (X.MIN,Y.MIN) and (X.MAX,Y.MAX), are ignored.

Importing 2D Profiles in TIF or TS4 Format

- When **I.SOURCE** is specified, the names in the **I.SOURCE** list must be the exactly same names of the corresponding solutions in the file.

```
PROFILE TWO.DIM IN.FILE=test.tif TIF +
        IMPURITY="arsenic,boron" I.SOURCE="As,B"
```

- The character ‘-’ in the **I.SOURCE** list indicates to import the solution data which corresponds to the given impurity name.:

  ```
  PROFILE TWO.DIM IN.FILE=test.tif TIF +
          IMPURITY=”arsenic,boron” I.SOURCE=”-,B”
  ```

  In the above example, the character ‘-’ means the TIF solution name “As” corresponding to arsenic. And, the B solution, which is the boron concentration in TIF, is to be added to boron solution into the current structure.
- When the value of the data to be imported is smaller than its minimum, which is either the hardwired internal value for the built-in solution (e.g., 1E5 for boron) or the value defined with the parameter **C.INIT** for the user-defined impurity, it is set to the minimum value. However, when the solution has negative data, for example, `SXX`, the data must be imported as it is. Turning on the parameter **NEGATIVE** in the **IMPURITY** statement specifies that the impurity solution is allowed to have negative values. By default, it is turned on for all stress solutions, velocities, and the potential solution.
- The parameter **EXPRESS** lets you reset a solution whenever they want. For example,

  ```
  PROFILE IMP=inters MAT=Si EXPR=1E22*exp(-y^2/0.1)
  ```

- You can reset a solution within a specific box region by setting **X.MIN**, **X.MAX**, **Y.MIN** and **Y.MAX**. And you can reset multiple solutions at once:

  ```
  PROFILE IMP="Sxx, Syy, Szz" MAT=Si EXPR=1E10 +
          X.MIN=0.1 X.MAX=0.2 Y.MIN=0 Y.MAX=0.1
  ```

- For **TWO.DIM** and **EXPRESS** imports, the equilibrium models associated with point-defects, such as point-defect pairing in **PD.FULL**, are recalculated automatically. Turning off **EQ.RESET** stops the equilibrium state from recalculating; this is not recommended.

Note:

The* `PROFILE` *command does not import the model definition. Thus, you must set the proper model associated with the data to be imported before importing the data. For example,* `CL.FULL` *must be turned on before importing I2, I3, I4 solution data.

```
METHOD  PD.FULL CL.FULL
PROFILE TWO.DIM IN.FILE=test.tif TIF +
        IMPURITY=”I2,I3,I4”
```

Example

- The following statement reads boron concentration data from the file *bprof.dat*:

  ```
  PROFILE BORON IN.FILE=bprof.dat OFFSET=-0.1
  ```

If the file contained the lines:

```
0.0  1.0e17
0.1  1.4e17
0.2  1.8e17
0.4  2.0e17
0.6  1.8e17
0.8  1.4e17
1.0  1.0e17
2.0  1.0e16
5.0  1.0e14
```

then a boron concentration of 1.0e17 would be added at *y*=-0.1, a concentration of 1.4e17 would be added at *y*=0, a concentration of 1.8e17 at *y*=0.1, and so on.

- The following statement defines boron profile in rectangular region.

```
PROFILE BORON IN.FILE=bprof.dat +
        X.MIN=1 X.MAX=3 Y.MIN=0.5 Y.MAX=3.5 +
        SIGMA=0.01
```

- The following statement imports the 2D data of the 3rd column for boron into the region of silicon material.

```
PROFILE TWO.DIM BORON IN.FILE=2dprof.dat
```

- The following statement imports the 2D data of the 5th column for indium into the region of polysilicon material.

```
PROFILE TWO.DIM IMP=INDIUM I.COLUMN=5 +
        IN.FILE=2dprof.dat
```

- The following statement imports the 2D data of the 3rd and 5th columns for boron and arsenic respectively into the region of silicon material.

```
PROFILE TWO.DIM IMP=”BORON,ARSENIC” I.COL=”3,5” +
        IN.FILE=2dprof.dat
```

- The following statement imports the 2D data of the 3rd column for arsenic into the region of silicon material only when the data is located in the inside of the rectangle, (0.5,0.0) and (1.0,2.0).

```
PROFILE TWO.DIM AS IN.FILE=2dprof.dat +
        X.MIN=0.5 Y.MIN=0.0 X.MAX=1.0 Y.MAX=2.0
```

- The following example shows that only arsenic, interstitial, and damage data among the data in test.tif are loaded into the default (silicon) material region. And arsenic data also are imported into oxide material region.

```
PROFILE TWO.DIM IN.FILE=test.tif TIF +
        IMPURITY="arsenic,interst,damage"
PROFILE TWO.DIM IN.FILE=test.tif TIF MAT=Ox +
        IMPURITY=arsenic
```

- The following example shows that the arsenic data are loaded within (left(=0.15um), top) and (right, bottom(=0.05um)) from test.sav which is written in TS4 structure format.

```
PROFILE TWO.DIM IN.FILE=test.sav TS4 IMP=arsenic +
        X.MIN=0.15 Y.MAX=0.05
```

- The following example shows that the boron data is loaded after shifting to the right by 0.03um, downward by 0.02 um.

```
PROFILE TWO.DIM IN.FILE=test.tif TIF IMP=boron +
        X.OFFSET=0.03 Y.OFFSET=0.02
```

- The following example shows that the boron and arsenic data are scaled by 2 and 5, respectively, and loaded.

```
PROFILE TWO.DIM IN.FILE=test.tif TIF +
        IMPURITY="boron,arsenic" +
        I.SCALE ="  2.0,   5.0"
```

- The following statement imports the user-defined intermediate data *AsImp* and *Iplus* and adds them to arsenic and interstitial concentrations, respectively. The damage solution also is updated by adding the data from *test.tif*.

```
PROFILE TWO.DIM IN.FILE=test.tif TIF      +
        IMPURITY="arsenic,interst,damage" +
        I.SOURCE="  AsImp,  Iplus,      -"
```

- There are two identical structures in equilibrium, but with different dopant distribution. When both structures need to be implanted with the same implant condition, the identical implantation on each structure must be performed. When the implantations are simulated by the MC implant model, using the **PROFILE** feature can remove the duplicated MC implant simulations. The example shows how to generate the arsenic profile of the MC implantation in such a case. The new impurity *myAs* is introduced because the implanted arsenic atoms must be separated from the background arsenic atoms.
- To calculate the +n interstitial profile, the current point-defect profiles are reset to their equilibrium concentrations by setting **PD.FERMI** followed by dummy diffusion for time=0. Then, the point-defect model is set to the 5-stream diffusion model to prevent the free interstitial concentration from being reduced by local equilibrium pairing. Since the implant adaptive gridding is internally applied by performing the analytic Taurus implantation first before the Monte Carlo implantation, TSUPREM-4 searches for the moment tables associated with *myAs*. Instead of manually creating the *myAs* moment tables by copying arsenic tables, the TIF name of the impurity *myAs* is temporarily assigned with *As* which is taken by Taurus analytic implant library for finding the analytic moment table for regrid. The simulation result is saved in TSUPREM-4 format so that the *myAs* data is stored in the name of the solution name *myAs*. However, if the simulation result is saved in TIF format, the TIF name of *myAs* should be different from *As*. Otherwise, the

myAs data is saved in the name of its TIF name *As* which is also the arsenic's TIF name, so the duplicated TIF names cause conflict.

```
$ The file for generating the MC implant profile
IMPURIT NEW IMP=myAs DONOR AT.NUM=33.0 AT.WT=74.91
INTERM  NAME=Iplus EXPR=INTERST-CI.STAR

INITIAL IN.FILE=test1.tif TIF

METHOD  PD.FERMI
ASSIGN  NAME=tempC  N.V=TEMP-273.16
DIFFUSE TEMP=@tempC TIME=0

METHOD  PD.5STR
IMPURIT IMP=myAs TIF.NAME=As
IMPLANT IMP=myAs DOSE=1E15 ENERGY=30 TILT=30 +
        MONTE N.ION=100000
SAVEFIL OUT.FILE=mcimplant.str INTERMED ZIP
```

In the other input file, the arsenic profile is imported instead of simulating the MC implantation.

```
$ In other input files,
$ instead of doing MC implant, importing the data
$ IMPLANT ARSENIC DOSE=1E15 ENERGY=30 TILT=30 +
$          N.ION=100000
PROFILE TWO.DIM IN.FILE=mcimplant.str TS4 ZIP +
        IMPURITY="arsenic,inters,damage" +
        I.SOURCE="   myAs, Iplus,damage"
PROFILE TWO.DIM IN.FILE=mcimp.str TS4 ZIP MAT=Ox +
        IMPURITY=arsenic I.SOURCE=myAs
```

- When the transistors are formed in the epitaxy layer which was grown on the top of a silicon substrate, there is little flexibility in controlling the horizontal grids in the epitaxy layer because the horizontal grids in the epitaxy layer are determined by those of the underlying layer. The **`PROFILE`** statement provides the method to conveniently remesh the epitaxy region. Since the mesh is optimized to form the buried layer in the substrate region, the grid structure is not adequate for emitter, base, and collector which are formed near the top of the epitaxy layer. Thus, regridding is essential to efficiently and accurately simulate the bipolar process. After generating the new grid structure aimed to the formation of emitter, base, and collector, the solutions are imported to the new structure by the **`PROFILE`** statement. To import all solutions at once, the parameter **`ALL.IMP`** is turned on. After 0.2 um epitaxy growth, the top coordinate is -0.2 um. However, setting the top coordinate to zero might be more convenient for etching definition later. So **`Y.OFFSET`** is given to 0.2.

```
METHOD  PD.FERMI
PROFILE TWO.DIM IN.FILE=epi.tif TIF MAT=Si +
        ALL.IMP Y.OFFSET=0.2
```

ELECTRODE

The **ELECTRODE** statement is used to name the electrodes for device simulation.

```
ELECTRODE
   [NAME=<c>] [ { ( X=<n> [Y=<n>] ) | BOTTOM } ]
   [CLEAR [ALL]]
   [MERGE]
   [PRINT]
```

Parameter	Type	Definition
NAME	character	The name of the electrode to define or delete. This parameter is required unless both **CLEAR** and **ALL** are specified. **Default:** none
X	number	The *x* coordinate of an (*x,y*) position in the region to be defined as an electrode. **Units:** microns **Default:** none
Y	number	The *y* coordinate of an (*x,y*) position in the region to be defined as an electrode. **Units:** microns **Default:** the *y* coordinate inside the conductor at the *x* position
BOTTOM	logical	Specifies that the electrode to be defined is at the bottom of structure. **Default:** false
CLEAR	logical	Specifies that the named electrode is to be deleted. **Default:** false
MERGE	logical	Specifies the polysilicon adjacent to the electrode to be merged as the electrode. Only applicable to the saved file in TIF format. **Default:** true
ALL	logical	Specifies that all defined electrodes are to be deleted. **Default:** false
PRINT	logical	Specifies that the summary of defined electrodes is printed. **Default:** true

Description

The **ELECTRODE** statement allows you to specify names for electrodes. These electrode names are saved in TIF and **MEDICI** output files for use in Medici and other programs. The materials which can be named to an electrode are polysilicon and all materials which have been defined as conductors on a **MATERIAL** statement. Once an electrode is named, the information is kept unless it is undefined by **CLEAR** in an **ELECTRODE** statement.

Examples

1. The following statements show how to name the electrodes.

```
ELECTROD  X=0.1  NAME=Source
ELECTROD  X=1.2  NAME=Gate
ELECTROD  X=2.3  NAME=Drain
ELECTROD  BOTTOM NAME=Bulk
SAVEFILE  OUT.FILE=mos.mdc MEDICI POLY.ELE
ELEC.BOT
```

2. The following statements show how to undefine the electrode.

```
ELECTROD  X=0.1 NAME=Wrong
     ....
ELECTROD  NAME=Wrong CLEAR
```

Additional ELECTRODE Notes

1. The polysilicon region named in **ELECTROD** is stored as an electrode in the saved file in TIF format. For Medici format, **POLY.ELE** must be specified in **SAVEFILE**.
2. The polysilicon region adjacent to the conductor named in **ELECTROD** is merged as the same electrode if **MERGE** parameter for TIF format, or **POLY.ELE** parameter (in **SAVEFILE**) for Medici format is specified.
3. If there are more than two conductors and/or polysilicon regions at the **X** position, for example, like EPROM structure, the **Y** must be specified individually.

```
ELECTROD  X=1.2 Y=-0.02 NAME=Erase
ELECTROD  X=1.2 Y=-0.35 NAME=Program
```

Process Steps

The following statements simulate processing steps:

Statement Name	Description	Page
`DEPOSITION`	Deposits a material on the exposed surface.	p. 4-99
`EXPOSE`	Exposes photoresist using mask.	p. 4-106
`DEVELOP`	Removes exposed positive photoresist or unexposed negative photoresist.	p. 4-108
`ETCH`	Etches an exposed material.	p. 4-109
`IMPLANT`	Implants an impurity.	p. 4-114
`DIFFUSION`	Performs a diffusion step, possibly with oxidation.	p. 4-131
`EPITAXY`	Performs a silicon epitaxial deposition step.	p. 4-137
`STRESS`	Calculates the stresses in the structure.	p. 4-142

DEPOSITION

The **`DEPOSITION`** statement is used to deposit a specified material on the exposed surface of the current structure.

```
DEPOSITION

   {  MATERIAL=<c> | SILICON | OXIDE | OXYNITRI | NITRIDE
    | POLYSILI | ALUMINUM
    | ( PHOTORES [ { POSITIVE | NEGATIVE } ] )
   }
   [ IMPURITY=<c>
     {  IT.CONC=<c> [IB.CONC=<c>] [LOG.GRAD]
      | I.PROFILE=<c> | I.RESIST=<n> } ]
   [ANTIMONY=<n>] [ARSENIC=<n>] [BORON=<n>] [PHOSPHOR=<n>]
   [ {CONCENTR | RESISTIV} ]
   THICKNES=<n> [SPACES=<n>] [DY=<n>] [YDY=<n>] [ARC.SPAC=<n>]
   [AR.MIN=<n>][TEMPERAT=<n>] [GSZ.LIN] [STRAINED]
   [ANISOTRO=<n>] TOPOGRAP=<c>
   [RELAX]
```

Parameter	Type	Definition
MATERIAL	character	The name of the material to be deposited. **Default:** none
SILICON	logical	Deposit silicon. **Default:** false
OXIDE	logical	Deposit oxide. **Default:** false
OXYNITRI	logical	Deposit oxynitride. **Default:** false
NITRIDE	logical	Deposit nitride. **Default:** false
POLYSILI	logical	Deposit polysilicon. **Default:** false
ALUMINUM	logical	Deposit aluminum. **Default:** false
PHOTORES	logical	Deposit photoresist. **Default:** false
POSITIVE	logical	Specifies that the deposited photoresist (and all other photoresist in the structure) is positive, i.e., that the **DEVELOP** statement removes exposed photoresist while leaving unexposed photoresist. **Default:** true, unless **NEGATIVE** is specified

Parameter	Type	Definition
NEGATIVE	logical	Specifies that the deposited photoresist (and all other photoresist in the structure) is negative, i.e., that the **DEVELOP** statement removes unexposed photoresist while leaving exposed photoresist. **Default:** false
IMPURITY	character	The names of the impurities with which the deposited layer is doped. If there are several impurities provided, their names should be enclosed in quotes and separated by the commas. **Default**: none
IT.CONC	character	The concentrations of the specified impurities in the deposited layer. If there is only one impurity provided, then its concentration will be interpreted as a floating number. For several impurities, a list of their concentrations should be provided in the same order as the list of impurity names in the **IMPURITY** parameter. The list should be enclosed in quotes and the concentrations should be separated by the commas. **Units**: atoms/cm^3 **Default**: none **Synonyms**: **I.CONC**
IB.CONC	character	The impurity concentrations at the bottom of the deposited layer of the impurity species listed in the **IMPURITY** parameter. The impurity concentrations at the top of the layer must be specified by the **IT.CONC** parameter. **Units**: atoms/cm^3 **Default**: Uniform impurity concentrations specified by **IT.CONC**
LOG.GRAD	logical	Specifies that the impurity concentrations specified by **IMPURITY** will vary logarithmically between the values specified by **IB.CONC** and **IT.CONC**. **Default:** false
I.PROFIL	character	The expression string for an equation that specifies the concentration of the single impurity specified by **IMPURITY** as a function of `F`, the fraction of the total layer thickness (or epitaxy time). `F` must obey the rules for a **DEFINE**d variable; it is recommended that it always by enclosed in braces and preceded by the "`@`" character; i.e., it should always appear as "`@{F}`" in the equation. **Units:** atoms/cm^3 **Default**: none
I.RESIST	number	The resistivity of the deposited layer. **Units:** ohm-cm **Default**: none
ANTIMONY	number	The uniform concentration or resistivity of antimony in the deposited layer. **Units**: atoms/cm^3 or ohm-cm **Default**: 0.0 **Synonyms**: **SB**
ARSENIC	number	The uniform concentration or resistivity of arsenic in the deposited layer. **Units**: atoms/cm^3 or ohm-cm **Default**: 0.0 **Synonym**: **AS**

Parameter	Type	Definition
BORON	number	The uniform concentration or resistivity of boron in the deposited layer. **Units**: atoms/cm^3 or ohm-cm **Default**: 0.0 **Synonym**: **B**
PHOSPHOR	number	The uniform concentration or resistivity of phosphorus in the deposited layer. **Units**: atoms/cm^3 or ohm-cm **Default**: 0.0 **Synonym**: **P**
CONCENTR	logical	Specifies that the impurity concentration in the deposited layer is given. **Default**: true
RESISTIV	logical	Specifies that the resistivity in the deposited layer is given. **Default**: false
THICKNES	number	The thickness of the deposited layer. **Units**: microns **Default**: none
SPACES	number	The number of vertical grid spacings in the layer. This value is divided by the value of **GRID.FAC** (see MESH, p. 4-53). **Units**: none **Default**: **1/GRID.FAC** **Synonym**: **DIVISION**
DY	number	The nominal grid spacing to be used in the deposited material layer at the location specified by the **YDY** parameter. This value is multiplied by the value of **GRID.FAC** (see MESH, p. 4-53). **Units**: microns **Default**: **GRID.FAC*THICKNES/SPACES** **Synonym**: **DX**
YDY	number	The location of the nominal grid spacing specified by **DY** relative to the top of the deposited layer. Units: microns **Default**: 0.0 **Synonym**: **XDX**
ARC.SPAC	number	The maximum spacing allowed along an arc on the new surface. This value is multiplied by the value of **GRID.FAC** (see MESH, p. 4-53). Units: microns **Default**: 0.5***THICKNES*GRID.FAC**
AR.MIN	number	The minimum aspect (width/height) ratio for deposited grid spaces on planar surfaces. **Default**: 0.5
TEMPERAT	number	The deposition temperature; used to determine initial grain size when depositing polycrystalline materials. **Units**: ° Celsius **Default**: 0.0

Parameter	Type	Definition
GSZ.LIN	logical	Specifies that the grain size increases linearly with depth from the bottom of the deposited layer. If false, the grain size is constant through the layer. **Default**: true
STRAINED	logical	Specifies whether the new SiGe or silicon layer is strained or relaxed. **Default**: false
ANISOTRO	number	Anisotropy of deposition, defined as a ratio of the layer thickness on top of a flat surface to the thickness of the layer deposited on a vertical wall. Anisotropy of 1 means isotropic deposition. Anisotropy of more than one means thinner layer on the sloped surface compared to the flat surface. Anisotropy can be any positive non-zero number. If it approaches infinity, then you get zero layer thickness on the vertical walls. If it approaches zero, then you are getting planarizing deposition filling everything up to the topmost point on the structure plus the nominal layer thickness. **Default**: 1
TOPOGRAP	character	The name of a file containing Taurus Topography input commands that define the deposition to be performed. **Default**: none **Synonym**: **TERRAIN**
RELAX	logical	Specifies to perform the stress relaxation after deposition **Default**: true

Description

This statement provides a basic deposition capability. Material is deposited on the exposed surface of the structure, with the upper surface of the deposited layer becoming the new exposed surface. The **ANTIMONY**, **ARSENIC**, **BORON**, and **PHOSPHOR** parameters specify the initial impurity concentrations or resistivity in the deposited layer, depending on whether **CONCENTR** or **RESISTIV** is true. Doping can also be specified with the **IMPURITY** and **I.CONC** or **I.RESIST** parameters (see Specifying Initial Doping, p. 4-69).

The deposited material conforms to the contours of the original surface. Outside corners on the original surface produce arcs on the new surface, which are approximated by straight line segments. The maximum segment length is set by the **ARC.SPAC** parameter. The **SPACES**, **DY**, and **YDY** parameters used to control the grid spacing in the deposited layer are scaled by the value of the **GRID.FAC** parameter on the **MESH** statement (see Chapter 3, Changes to the Mesh During Processing, p. 3-7). The horizontal spacing in a thin deposited layer is the same as in the substrate, but for thick layers it is increased to avoid mesh elements with high ratios of height to width. This increase in horizontal spacing can be controlled with the **AR.MIN** parameter.

Note:

It is not possible to deposit a layer on the bottom of a structure, even if it is exposed. Attempting to do so may cause the program to fail.

Polycrystalline Materials

The **TEMPERAT** and **GSZ.LIN** parameters are used only when depositing a polycrystalline material. If no temperature is specified, or if the temperature is less than the value of **TEMP.BRE** for the material, the deposited layer is amorphous; recrystallization occurs at the start of the next high-temperature step.

Photoresist

Photoresist can be positive or negative, but all photoresist in a structure must be of the same type. If photoresist of one type is deposited on a structure containing photoresist of the other type, a warning is issued and the type of the old photoresist is changed to that of the newly deposited photoresist.

Gradient-Doped Deposition

If **IB.CONC** is specified, the impurities named in the **IMPURITY** parameter will have the concentrations **IB.CONC** at the bottom of the deposited or grown layer and the concentrations **IT.CONC** at the top of the layer. The concentration will vary linearly through the layer unless **LOG.GRAD** is specified; in this case, the variation will be logarithmic. **IB.CONC** can only be used when **IMPURITY** and **IT.CONC** (alias **I.CONC**) are also specified.

If **I.PROFIL** is specified, it is interpreted as an equation specifying the concentration of a single impurity (specified by **IMPURITY**) as a function of `F`, the fraction of the layer thickness (`F`=0 at the bottom of the layer and `F`=1 at the top). The parameter `F` must obey the rules for a **DEFINE** variable; it is recommended that it be enclosed in braces and preceded by the "`@`" character, i.e., "`@{F}`".

Note:

The doping at interior nodes in the deposited layer is calculated by linear or logarithmic interpolation or by I.PROFIL. Thus, the total dose will be expected to differ from that calculated by analytical integration of the specified gradient (except in the special case of a linear gradient and a uniformly spaced grid).

Deposition with Taurus Topography

The **TOPOGRAP** parameter invokes Taurus Topography with the specified command input file. The command input file contains Taurus Topography commands describing one or more processing steps to be simulated by Taurus Topography. It should not contain the **INITIALIZE** or **STOP** statements.

The values of variables set with the **ASSIGN**, **DEFINE**, and **EXTRACT** statements are substituted in the Taurus Topography command input file. In addition to variables set explicitly by you, if the **THICKNES** parameter is set on the **DEPOSITION** statement, then its value is assigned to the variable **THICK** before substitution. (If the variable **THICK** is assigned in this way, it will be unset after the **DEPOSITION** statement, even if it was set by you previously.) This allows parameter values (such as deposition thickness) to be passed to Taurus Topography. The most recent mask file specified in the TSUPREM-4 input file is passed to Taurus Topography for use in masked etch steps.

By default, Taurus Topography is called by requesting that the command **TOPOGRAPHY** be executed by the operating system, but if the environment variable **`S4TERRAIN`** is set, its value is used instead. It may be necessary for you to define other environment variables (e.g., **`TERR_LIB`**) for Taurus Topography to run correctly. For additional details, refer to the *Taurus Topography Reference Manual*.

When Taurus Topography is invoked with the **`DEPOSITION`** statement, the full set of parameters is used for any deposited layer of the specified material. Thus you have full control over grid spacing, doping, and polycrystalline grain size. *Note* that the parameters specified on the **`DEPOSITION`** statement only apply to the specified material. They are ignored for other materials deposited by Taurus Topography.

Examples

1. The following statement deposits 200 angstroms of silicon dioxide:

```
DEPOSIT OXIDE THICK=0.02
```

2. The following statement deposits a one-micron thick layer of photoresist, using four grid spaces in the layer:

```
DEPOSIT PHOTO THICK=1.0 SPACES=4 ARC.SPAC=0.1
```

The maximum segment length used to approximate arcs is 0.1 micron. By default, the newly deposited photoresist (and any photoresist already present in the structure) is assumed to be positive.

3. The following statement deposits 0.1 micron of poly:

```
DEPOSIT MAT=POLY THICK=0.1 TEMPERAT=650 GSZ.LIN
```

The initial grain size is calculated at 650× ° C, and increases linearly over the thickness of the layer, with minimum value at the bottom of the layer.

4. The following statement calls Taurus Topography with the input command file *PolyDep.inp*:

```
DEPOSIT MAT=POLY THICK=0.1 TEMPERAT=650 GSZ.LIN +
             TOPOGRAPHY=PolyDep.inp
```

References to “**`@THICK`**” in the Taurus Topography input file are replaced with the value “0.1”. If the call to Taurus Topography causes the deposition of a polysilicon layer, then the specified temperature and grain size model are used for that layer.

5. The following statement deposits a strained SiGe layer:

```
DEPOSIT STRAINED SILICON THICKNES=0.1 SPAC=15 +
        IMPURITY=germanium I.CONC=2.5e21
```

6. Silicon deposition with Ge that varies from 10% (bottom) to 20% (top):

```
DEPOSIT SILICON THICKNES=0.1 SPACES=10 +
        IMP=GERMANIUM I.CONC=1E22 IB.CONC=0.5E22
```

7. Same as 6. But use **`I.PROFILE`**.

```
DEPOSIT SILICON THICKNES=0.1 SPACES=10 +
        IMP=GERMANIUM +
        I.PROFIL=( 0.5E22 + @{F}*(1E22-0.5E22) )
```

8. Deposition with linear gradient multiple species:

```
DEPOSIT POLY THICK=0.2 SPACES=10 TEMP=800 GSZ.LIN +
        IMP="BORON, PHOSPHORUS" +
        IB.CONC="5E13 1E15" +
        I.CONC="5E14 2E15"
```

Additional DEPOSITION Notes

1. The calculation doping concentration from resistivity uses mobility tables for silicon and polysilicon. If the resistivity is specified when depositing some other material, a warning is issued. The mobility tables used for calculating the doping concentration are not the same as the tables used by the **ELECTRICAL** statement. The extracted sheet resistance for the deposited layer does not correspond exactly to the resistivity specified during deposition.
2. Deposition of one material on top of another can cause a third material to be added between them. This happens when titanium is deposited on silicon, for example—a layer of $TiSi_2$ is inserted. Insertion of extra layers is specified with the **REACTION** statement (see p. 4-300).
3. Diffusion of impurities during a deposition step is not simulated, even though a temperature may be specified.

EXPOSE

The **`EXPOSE`** statement exposes photoresist using a mask.

```
EXPOSE

   MASK=<c> [SHRINK=<n>] [OFFSET=<n>]
```

Parameter	Type	Definition
`MASK`	character	Specifies the name of the mask to be used for the exposure step. The mask name must match the name of a mask previously read with a **`MASK`** statement. **Default:** none
`SHRINK`	number	The reduction in line width on each side of each line on the mask. The total reduction in line width is twice the specified value. **Units:** microns **Default:** 0.0
`OFFSET`	number	The amount by which lines on the mask are shifted (in the positive *x* direction). **Units:** microns **Default:** 0.0

Description

The **`EXPOSE`** statement exposes photoresist, using masking information previously read with a **`MASK`** statement. Regions of photoresist that are directly below transparent regions of the specified mask (and not obscured by any other material) are marked as exposed. If positive photoresist is present, exposed regions are removed by a subsequent **`DEVELOP`** statement; if negative photoresist is present, the unexposed portion of the photoresist is removed.

The **`SHRINK`** and **`OFFSET`** parameters can be used to model the effects of deliberate or accidental adjustments to the mask-making or photolithographic processes. **`SHRINK`** specifies a reduction in the width of lines on the mask, while **`OFFSET`** specifies a shifting of the mask (relative to other masks). The program does not allow shrinks or offsets that are large enough to make mask lines or spaces disappear.

Note:

> ***The `OFFSET` parameter breaks the symmetry that is assumed at reflecting boundaries at the edge of a structure. Shifting a line towards a reflecting boundary also makes it narrower, while shifting a line away from a reflecting boundary makes it wider.***

Example

The following sequence illustrates a typical use of the **`EXPOSE`** statement:

```
MASK  IN.FILE=CMOS3.TL1
  .
  .
  .
DEPOSIT  POLY  THICKNES=.2
DEPOSIT  POSITIVE PHOTORES  THICKNES=1
EXPOSE   MASK=POLY SHRINK=0.05
DEVELOP
ETCH     POLY  TRAP  THICK=0.4  ANGLE=80
ETCH     PHOTORES  ALL
```

This sequence produces lines of polycrystalline silicon under the opaque regions of the mask named *POLY*. The width of each *POLY* line is reduced by 0.05 microns on each side (for a total reduction of 0.1 micron).

DEVELOP

The **`DEVELOP`** statement removes exposed positive photoresist and unexposed negative photoresist.

```
DEVELOP

    [<c>]
```

Description

The **`DEVELOP`** statement is used to pattern photoresist by removing exposed positive resist and unexposed negative resist. The character string associated with the **`DEVELOP`** statement is ignored, and serves only to document the input.

Example

The following sequence illustrates a typical use of the **`DEVELOP`** statement:

```
MASK   IN.FILE=CMOS3.TL1
  .
  .
  .
DEPOSIT  POLY  THICKNES=.2
DEPOSIT  POSITIVE PHOTORES  THICKNES=1
EXPOSE   MASK=POLY
DEVELOP
ETCH     POLY  TRAP  THICK=0.4  ANGLE=80
ETCH     PHOTORES  ALL
```

This sequence produces lines of polycrystalline silicon under the opaque regions of the mask named *POLY.*

ETCH

The **ETCH** statement is used to remove portions of the current structure.

```
ETCH

   [ {  MATERIAL=<c> | SILICON | OXIDE | OXYNITRI | NITRIDE
      | POLYSILI | PHOTORES | ALUMINUM
     }
   ]
   {  ( TRAPEZOI [THICKNES=<n>] [ANGLE=<n>] [UNDERCUT=<n>] )
    | ( {LEFT | RIGHT} [P1.X=<n>] [P1.Y=<n>] [P2.X=<n>] [P2.Y=<n>] )
     | ( {START | CONTINUE | DONE} X=<n> Y=<n> )
     | ISOTROPI
     | ( OLD.DRY THICKNES=<n> )
     | ALL
     | TOPOGRAP=<c>
   }
   [RELAX]
```

Parameter	Type	Definition
MATERIAL	character	The name of the material to be etched. **Default:** none
SILICON	logical	Etch silicon only. **Default:** false
OXIDE	logical	Etch oxide only. **Default:** false
OXYNITRI	logical	Etch oxynitride only. **Default:** false
NITRIDE	logical	Etch nitride only. **Default:** false
POLYSILI	logical	Etch polysilicon only. **Default:** false
PHOTORES	logical	Etch photoresist only. **Default:** false
ALUMINUM	logical	Etch aluminum only. **Default:** false
TRAPEZOI	logical	Use an etch model that removes material from a trapezoidal region when applied to a planar surface. This is a generalization of the **DRY** model in older versions of TSUPREM-4. **Default:** true **Synonym**: **DRY**

Parameter	Type	Definition
THICKNES	number	The thickness of the layer to be removed when the **TRAPEZOI** or **OLD.DRY** parameter is specified. **Units:** microns **Default:** **UNDERCUT***tan(**ANGLE**) for **ANGLE**<90, infinite otherwise; no default if **OLD.DRY** is specified
ANGLE	number	The angle of the sidewalls produced when the **TRAPEZOI** parameter is specified. The angle is measured from the horizontal, so that vertical sidewalls have an angle of 90 deg. **Units:** degrees **Default:** arctan(**THICKNES/UNDERCUT**) if **THICKNES** and undercut are both specified, 90 otherwise
UNDERCUT	number	The distance that the etch extends under masking layers when the **TRAPEZOI** parameter is specified. **Units:** microns **Default:** **THICKNES**/tan(**ANGLE**) for **ANGLE**<90, 0 otherwise
LEFT	logical	Etch material to the left of the specified position. **Default:** false
RIGHT	logical	Etch material to the right of the specified position. **Default:** false
P1.X	number	The *x* coordinate of the first point used when **LEFT** or **RIGHT** is specified. **Units:** microns **Default:** 0.0
P1.Y	number	The *y* coordinate of the first point used when **LEFT** or **RIGHT** is specified. **Units:** microns **Default:** a point above the top of the structure
P2.X	number	The *x* coordinate of the second point used when **LEFT** or **RIGHT** is specified. **Units:** microns **Default:** the value of **P1.X**
P2.Y	number	The *y* coordinate of the second point used when **LEFT** or **RIGHT** is specified. **Units:** microns **Default:** a point below the bottom of the structure
START	logical	The point (**X**,**Y**) is the first point in a series defining the region to be etched. **Default:** false
CONTINUE	logical	The point (**X**,**Y**) is the next point in a series defining the region to be etched. **Default:** false
DONE	logical	The point (**X**,**Y**) is the last point in a series defining the region to be etched. **Default:** false **Synonym: END**
ISOTROPI	logical	Specifies removal of material that lies within **THICKNES** of an exposed surface. **Default:** false

Parameter	Type	Definition
X	number	The *x* coordinate used with the **START**, **CONTINUE**, or **DONE** parameter. **Units:** microns **Default:** none
Y	number	The *y* coordinate used with the **START**, **CONTINUE**, or **DONE** parameter. **Units:** microns **Default:** none
OLD.DRY	logical	The exposed surface is etched vertically by an amount given by the **THICKNES** parameter. **Default:** false
ALL	logical	The specified material is etched away entirely. **Default:** false
TOPOGRAP	character	The name of a file containing Taurus Topography input commands that define the etch to be performed. **Default:** none **Synonym: TERRAIN**
RELAX	logical	Specifies to perform the stress relaxation after etch **Default**: true

Description

This statement is used to remove a portion of the current structure. You can define the region to be removed and optionally specify a material to be removed; if no material is specified, all materials are considered to be etchable. Portions of the structure are removed, provided that they are of an etchable material, lie within the defined etch region, and are exposed to the ambient. See Chapter 3, Etching, p. 3-157 for a more complete description. The surface under the etched portions of the structure is marked as exposed. The warning:

```
*** Warning:  No material removed by ETCH statement.
```

is produced by an attempt to etch a material that is not exposed, or by specifying etch coordinates that do not include any etchable material.

Removing Regions

You can specify the region to be removed in one of the following ways:

1. If **TRAPEZOI** is specified, the etch region is found from a simple model of a primarily anisotropic (i.e., vertical, or directional) etch with a small isotropic component. This model can produce profiles with sloped sidewalls and undercutting of masking layers. See Chapter 3, Trapezoidal Etch Model, p. 3-158.
2. If **LEFT** or **RIGHT** is specified, the etch region includes all material to the left or right of the line between (**P1.X**,**P1.Y**) and (**P2.X**,**P2.Y**).

3. The **START**, **CONTINUE**, and **DONE** parameters are used with the **X** and **Y** parameters to define arbitrarily complex etch regions. The boundary of the region is determined by a series of **ETCH** statements, each specifying a point on the boundary. The first statement of the series should contain the **START** parameter, the last should contain the **DONE** parameter, and statements in between should use the **CONTINUE** parameter. The last point is connected to the first point to produce a closed region defining the portion of the structure to be removed.
4. If **ISOTROPI** is specified, the etch region includes all material within the distance **THICKNES** of the exposed surface. This produces a simple isotropic etch, without rounding of outside corners.
5. If the **OLD.DRY** parameter is specified, the etch region includes all material within a vertical distance **THICKNES** of the exposed surface. This model (the **DRY** model in previous versions of TSUPREM-4) has been replaced by the **TRAPEZOI** model.

   The **OLD.DRY** model in Version 5.1 of TSUPREM-4 is the same as the **DRY** model in older versions of the program. In Version 5.1, **DRY** is a synonym for **TRAPEZOI**. The **TRAPEZOI** model with default values of **ANGLE** and **UNDERCUT** is equivalent to the **OLD.DRY** model, except that surface layers of nonetchable material blocks etching of underlying material, even if the surface layer is thinner than **THICKNES**. Specifying **DRY** in Version 5.1 is equivalent to specifying **DRY** in previous versions in cases of practical interest.
6. If the **ALL** parameter is specified, the etch region includes the entire structure.

- If no region specification is given, **TRAPEZOI** is assumed.
- If a material is specified, only that material is etched; otherwise, the entire region specified is subject to removal.

It is possible to cut the structure into two or more pieces with an **ETCH** statement. In this case, all pieces except the one with the largest area are removed. A warning is issued for each piece removed.

Note:

The* ETCH *statement (except when used with the* TERRAIN *parameter) is not intended to simulate a physical etching process; its purpose is to provide a means to generate the required structures for simulation of diffusion and oxidation. Note in particular that the statement "*ETCH OXIDE TRAP*" does not implement a selective etch of oxide, but rather defines a region geometrically in which all exposed oxide is removed.

Etching with Taurus Topography

The **TOPOGRAP** parameter invokes Taurus Topography with the specified command input file. The command input file contains Taurus Topography commands describing one or more processing steps to be simulated by Taurus Topography. It should not contain the **INITIALIZE** or **STOP** statements.

The values of variables set with the **ASSIGN**, **DEFINE**, and **EXTRACT** statements are substituted in the Taurus Topography command input file. In addition to variables set explicitly by you, if the `THICKNES` parameter is set on the `ETCH` statement, then its value is assigned to the variable `THICK` prior to substitution. (If the variable `THICK` is assigned in this way, it will be unset after the `ETCH` statement, even if it was set by you previously.) This allows parameter values (such as etch thickness) to be passed to Taurus Topography. The most recent mask file specified in the TSUPREM-4 input file is passed to Taurus Topography for use in masked etch steps.

By default, Taurus Topography is called by requesting that the command `topography` be executed by the operating system, but if the environment variable `S4TERRAIN` is set, its value is used instead. It may be necessary for you to define other environment variables (e.g., `TERR_LIB`) for Taurus Topography to run correctly. For additional details, refer to the *Taurus Topography Reference Manual*.

Examples

1. The following statement etches the nitride to the left of 0.5 μ to a depth of 1 micron:

```
ETCH NITRIDE LEFT P1.X=0.5 P2.Y=-1.0
```

Note:

`P1.Y` defaults to a location above the top of the structure and `P2.X` defaults to the value of `P1.X` (i.e., 0.5).

2. This statement etches the oxide in the square defined by (0,0), (1,0), (1,1), (0,1):

```
ETCH OXIDE START X=0.0 Y=0.0
ETCH    CONTINUE X=1.0 Y=0.0
ETCH    CONTINUE X=1.0 Y=1.0
ETCH        DONE X=0.0 Y=1.0
```

Material is removed only if there is an exposed oxide surface somewhere within the boundaries of the etch.

3. The following statement calls Taurus Topography with the input command file *PolyDep.inp*:

```
ETCH THICK=0.1 TOPOGRAPHY=PolyDep.inp
```

References to “`@THICK`” in the Taurus Topography input file are replaced with the value “0.1”.

4. The following statement etches all materials except silicon:

```
ETCH ^SILICON ALL
```

IMPLANT

The **`IMPLANT`** statement is used to simulate the ion implantation of impurities into the structure.

```
IMPLANT

   {IMPURITY=<c> | ANTIMONY | ARSENIC | BORON | BF2 | PHOSPHOR}
   DOSE=<n> ENERGY=<n> [TILT=<n>] [ROTATION=<n>] [MULT.ROT=<n>]
   [{EXTEND | PERIODIC | REFLECT | VACUUM}] [MARGIN=<n>]
   [DAMAGE] [D.PLUS=<n> | D.P1=<n>] [D.SCALE=<n>] [D.RECOMB]
   [L.RADIUS=<n> { (L.DENS=<n> [L.DMIN=<n>] [L.DMAX=<n>])
                 | (L.THRESH=<n> [L.FRAC=<n>]) }]
   [REGRID] [PAI] [DOSE.SCA]
   {  ( [{GAUSSIAN | PEARSON}] [BACKSACT] [PAI.THRE=<n>]
        [IMPL.TAB=<c>] [DAM.TABL=<c>] [DAM.FILE=<c>] [MOMENTS]
        [ALSO.TAB] [PRINT] [DAM.ADAP]
        [V.PLUS=<n>] [DX.FP=<n>] [DY.FP=<n>] [DX.IPLUS=<n>] [DY.IPLUS=<n>]
        { ([RP.EFF] [IN.FILE=<c>] [OXTH.FAC=<n>])
        | ([DATA.SUF=<c>] [DAM.SUF=<c>] [DATA.PAT=<c>] [RANGE.SH=<n>])}
      )
    | ( [N.ION=<n>] [BEAMWIDT=<n>] [SEED=<n>] [CRYSTAL] [TEMPERAT=<n>]
        { MONTECAR
            [DAM.MOD=<c>] [DEPTH.SP] [DOSE.SPL] [TRAJ.SPL] [TRAJ.REP]
            [X.LENGTH=<n>] [REC.OXYG] [RANDOM] [POINT.RE] [MAX.SPLI]
            [SAVE1D [SAVE1D.F=<c>] [SAVE1D.S] [SAVE1D.H] [SAVE1D.U=<c>]]
        | OLD.MC
            [VIBRATIO [X.RMS=<n>] [E.LIMIT=<n>]]
            [THRESHOL=<n>] [POLY.GSZ=<n>] [INTERST=<c>] [MAX.DAM=<n>]
            [REC.FRAC=<n>]
        }
      )
   }
```

Parameter	Type	Definition
`IMPURITY`	character	The name of the impurity to be implanted. **Default:** none **Synonym:** **`IMP`**
`ANTIMONY`	logical	Specifies that antimony is to be implanted. **Default:** false **Synonym:** **`SB`**
`ARSENIC`	logical	Specifies that arsenic is to be implanted. **Default:** false **Synonym:** **`AS`**
`BORON`	logical	Specifies that boron is to be implanted. **Default:** false **Synonym:** **`B`**

Parameter	Type	Definition
`BF2`	logical	Specifies that boron in the form of BF2 is to be implanted. **Default:** false
`PHOSPHOR`	logical	Specifies that phosphorus is to be implanted. **Default:** false **Synonym:** **`P`**
`DOSE`	number	The dose of implanted ions. **Units:** atoms/cm^2 **Default:** none
`ENERGY`	number	The acceleration energy of the ion implant beam. **Units:** keV **Default:** none
`TILT`	number	Specifies the tilt angle of the wafer, measured in a clockwise direction from the horizontal in the plane of the simulation. Positive tilt angles correspond to implanted ions arriving from the left side of vertical, and negative tilt angles correspond to implanted ions arriving from the right side of vertical (assuming the wafer rotation angle is less than 90 degrees and greater than –90 degrees). Tilting of the wafer with respect to the ion beam can contribute to shadowing, dose reduction and foreshortening of the implant distribution; see Chapter 3, Analytic Ion Implant Models, p. 3-119. **Units:** degrees **Default:** 0
`ROTATION`	number	The angle by which the ion beam has been rotated from the [110] primary flat orientation, measured in a clockwise direction about an axis perpendicular to and facing into the wafer surface. Implanting with a value of 0.0 for this parameter corresponds to implantation in the plane formed by Y-axis and [110] flat direction. Wafer rotation can contribute to shadowing of portions of the simulation structure (see Chapter 3, Analytic Ion Implant Models, p. 3-119). **Units:** degrees **Default:** 0
`MULT.ROT`	number	The number of rotations by the rotational angle 360 deg/**`MULT.ROT`** degrees with the starting angle **`ROTATION`**. **Default:** 1
`EXTEND`	logical	Specifies that extend (1D-continuation) boundary conditions be imposed at the left and right edges of the structure during implantation. Applies to the Taurus analytic and Taurus MC models only. **Default:** true
`PERIODIC`	logical	Specifies that periodic boundary conditions be imposed at the left and right edges of the structure during implantation. Applies to the Taurus analytic and all MC models. **Default:** false

Parameter	Type	Definition
REFLECT	logical	Specifies that reflecting boundary conditions be imposed at the left and right edges of the structure during implantation. Applies to the Taurus analytic and all MC models. **Default:** false
VACUUM	logical	Specifies that boundary conditions corresponding to vacuum material be imposed at the left and right edges of the structure during MC implantation. Applies to the Taurus analytic and all MC models. **Default:** false
MARGIN	number	Specifies the size of the margins by which the simulation domain is expanded during implantation to account for the boundary conditions. Applies to the Taurus analytic and Taurus MC models only. **Default:** automatically determined.
DAMAGE	logical	Specifies that point defects generated by ion implantation are to be retained and used as an initial condition for subsequent process steps. **Default:** true when **PD.TRANS** or **PD.FULL** is selected
D.PLUS	number	The scale factor applied to the implanted profile to obtain the *plus one* component of the damage profile. **Units:** none **Default:** effective +n model used
D.P1	number	The +1 component of the effective +n model for implant damage. **Units:** none **Default:** 1.0
D.SCALE	number	The scale factor applied to the Frenkel pair component of the damage profile. **Units:** none **Default:** 1.0
D.RECOMB	logical	Compute the recombination of interstitials and vacancies analytically at the end of the **IMPLANT** step. **Default:** true
L.RADIUS	number	The radius of dislocation loops (if any) resulting from the implant. **Units:** cm **Default:** none
L.DENS	number	The density of dislocation loops (if any) resulting from the implant. **Units:** #/cm^3 **Default:** none
L.DMIN	number	The minimum damage level used to define the region where dislocation loops are produced. **Units:** #/cm^3 **Default:** 1e20

Parameter	Type	Definition
`L.DMAX`	number	The maximum damage level used to define the region where dislocation loops are produced. **Units:** #/cm3 **Default:** 1.15e22
`L.THRESH`	number	The interstitial concentration above which dislocation loops are produced. **Units:** #/cm3 **Default:** none
`L.FRAC`	number	The fraction of interstitials in excess of **`L.THRESH`** that are incorporated into dislocation loops. **Units:** none **Default:** 1.0
`REGRID`	logical	If true, the mesh is adoptively regridded using Taurus analytic implant for library-based Taurus MC implant. **Synonym**: **`TS4.REGR`** **Default:** true
`PAI`	logical	Taurus implant parameter (PAI). Determines if the pre-existing implant damage should be taken into account during the implantation. Applies to the Taurus analytic and Taurus MC models only. **Default:** true
`DOSE.SCA`	logical	Enables the nominal implant dose integration along the direction of wafer normal. When it is false, the nominal dose is the value integrated along the ion beam direction. Applies to the Taurus analytic and Taurus MC models only. **Default:** false
`GAUSSIAN`	logical	Specifies that a simple Gaussian distribution is to be used to represent the implanted impurity profile. **Default:** false
`PEARSON`	logical	Specifies that a Pearson or dual-Pearson distribution is to be used to represent the implanted impurity profile. **Default:** true, unless **`GAUSSIAN`** is specified
`BACKSCAT`	logical	Enables modeling of backscattering of ions from the surface. **Default:** true
`PAI.THRE`	number	Threshold implant damage concentration for the conversion of the implant damage into an equivalent amorphous layer thickness. Applies to the Taurus analytic model only **Units**: cm-3 **Default:** 2.5e22

Parameter	Type	Definition
IMPL.TAB	character	The name used to choose the range statistics from the implant moment data file. **Default:** *antimony*, *tr.indium*, *dual.ars*, *chboron*, *dual.bf2*, and *dual.pho* for **ANTIMONY**, **INDIUM**, **ARSENIC**, **BORON**, **BF2**, and **PHOSPHOR** implantations, respectively. Use **IMPL.TAB=taurus** to select the Taurus Analytic Implant Model. **Synonym: NAME**
DAM.TABL	character	The name of a table of damage parameters in an implant damage file. Values of the **RANGE**, **SIGMA**, **LAMBDA**, **LSIGMA**, **LSLOPE**, and **SCALE** parameters for implant damage are taken from the specified table in the implant damage file (unless specified on a preceding **MOMENT** statement). **Default:** If **DAM.TABL** is not specified on the **IMPLANT** statement, a table name specified on an **IMPURITY** statement is used or parameters are obtained from the built-in equations
DAM.FILE	character	The name of a file continuing tables of damage parameters. **Default:** The file specified by the environment variable **S4IDAM** or the file named "*s4idam*" in the current working directory.
MOMENTS	logical	Use range statistics from previously specified **MOMENT** statements, instead of from a moment data file. **Default:** false
ALSO.TAB	logical	This parameter is only valid when paired with **MOMENTS**. Enables TSUPREM-4 to get the missing moments from the implant tables if necessary. The implant table for silicon could be specified using **IMPL.TAB** keyword. If **IMPL.TAB** is not specified, the default table is used. **Default:** false
PRINT	logical	Prints a summary of the current implantation on the standard output and in the output listing file. **Default:** false
DAM.ADAP	logical	Apply adaptive gridding in the regions where the damage concentration is close to the **MAX.DAM** value. Applies to analytic implant models only. **Default:** false
VPLUS	number	The scale factor applied to the implanted profile to obtain additional vacancies to be added to the vacancy profile. Applies to analytic implant models only. **Units:** none **Default:** 0.0
DX.FP	number	Specifies the right-shift of interstitial locations to vacancy locations in Frenkel-pair generation. Applies to analytic implant models only. **Units:** microns **Default:** 0.0

Parameter	Type	Definition
`DY.FP`	number	Specifies the downward shift of interstitial locations to vacancy locations in Frenkel-pair generation. Applies to analytic implant models only. **Units:** microns **Default:** 0.0
`DX.IPLUS`	number	Specifies the right-shift of +n model interstitial locations to implanted dopant locations. Applies to analytic implant models only. **Units:** microns **Default:** 0.0
`DY.IPLUS`	number	Specifies the downward shift of +n model interstitial locations to implanted dopant locations. Applies to analytic implant models only. **Units:** microns **Default:** 0.0
`DX.VPLUS`	number	Specifies the right-shift of **`VPLUS`** vacancy locations to implanted dopant locations. Applies to analytic implant models only. **Units:** microns **Default:** 0.0
`DY.VPLUS`	number	Specifies the downward shift of **`VPLUS`** vacancy locations to implanted dopant locations. Applies to analytic implant models only. **Units:** microns **Default:** 0.0
`RP.EFF`	logical	Specifies that the effective range scaling method is to be used to calculate the implanted profile in multi-layer targets. If **`RP.EFF`** is specified to be false, the dose matching method is used to calculate the implanted profile in multi-layer targets. **Default:** true
`IN.FILE`	character	The identifier for the file containing implant range statistics. If specified, this file is used instead of the default implant moment data file *s4imp0*. Not applicable to Taurus analytic model. **Default:** none **Synonym**: **`FILE`**
`OXTH.FAC`	number	The thickness correction factor when other materials are converted into an equivalent oxide thickness. This parameter is effective only if scr.* implant tables are used. Not applicable to Taurus analytic model. **Units:** none **Default:** 1.0
`DATA.SUF`	character	Taurus analytic implant parameter (**`IMPLANTDATA`**). Specifies the file name suffix for the implant tables which contain the desired implant data in the format of ion_in_material_suffix. This parameter changes the suffix for all materials. **Default:** **`STANDARD`**

Parameter	Type	Definition
DAM.SUF	character	Taurus analytic implant parameter (**IMPLANTDAMAGEDATA**). Specifies the file name suffix for the implant tables which contain the desired implant damage data in the format of ion_damage_in_material_suffix. This parameter changes the suffix for all materials. **Default: STANDARD**
RANGE.SH	logical	Taurus analytic implant parameter (**PROPORTIONALRANGESHIFT**). Selects an approach for shifting the projected ranges in a multilayer target. If it is false, the projected ranges for channeling and damage profiles in the multi-layer target are shifted by the same amount as the main projected range. If it is true (default), the projected ranges for channeling and damage profiles are shifted by an amount, proportional to their ratio, to the main projected range. **Default:** true
DATA.PAT	character	Taurus analytic implant parameter. Specifies the path (directory) for the implant data and implant damage data where Taurus implant should look first. If implant data are not found in the specified directory, Taurus will look for the data in the directory specified by the environment variable *PHYSICS_DATA*. **Default**: current working directory
N.ION	number	The number of ion trajectories to calculate to generate the MC implant distribution. **Units:** none **Default:** 1000
BEAMWIDT	number	The divergence angle of the ion beam. This parameter is used to select a uniform distribution of angles about the normal values for **TILT** and **ROTATION**. The range of values by which **TILT** and **ROTATION** are varied is from –1/2 to +1/2 times the value specified for **BEAMWIDT**. **Units:** degrees **Default:** 0.0
SEED	number	A positive integer less than 2^{31} used as an initial value for the random number generator. Changing the seed gives a different numerical simulation for a given implant. This can be used to estimate the statistical uncertainty inherent in the MC simulation. **Units:** none **Default:** 101
CRYSTAL	logical	Specifies that the crystal structure of the silicon target material is to be included in the MC implant calculation. **Default:** true

Parameter	Type	Definition
TEMPERAT	number	The temperature of the target material during implantation. This parameter is used for calculating the amplitude of silicon lattice vibrations and the amount of damage self-annealing. This parameter is used only for the MC implant calculation. **Units:** degrees Celsius **Default:** 26.84
MONTECAR	logical	Specifies that Taurus MC Implant is used to perform the implantation. **Default:** false
DAM.MOD	character	Taurus MC damage model: one of none, plus, or cascade. **Default:** plus (Taurus MC default)
DEPTH.SP	logical	Enables trajectory splitting using depth as a criterion. This parameter is available only for Taurus MC implant. **Default:** false
DOSE.SPL	logical	Enables dose split algorithm to reduce the channeling tail noise. This parameter is available only for Taurus MC implant. **Default:** true for dose above amorphization, false below amorphization
TRAJ.SPL	logical	Enables a general trajectory splitting algorithm. This parameter is available only for Taurus MC implant. **Default:** false
TRAJ.REP	logical	Enables trajectory replication algorithm. This parameter is available only for Taurus MC implant. **Default:** false
X.LENGTH	number	Taurus MC implant parameter (xLength). Specifies the length of the replication subwindows in x direction (in microns). This is used only if **TRAJ.REPL** is turned on. **Units**: um **Default:** 0.01
REC.OXYG	logical	Taurus MC implant parameter (OxygenRecoil). Specifies whether to calculate the oxygen knock-on effect. **Default:** false
RANDOM	logical	Taurus MC implant parameter (Randomize). Randomizes the seed of the random number generator for MC ion implantation such that statistics analysis can be performed without modifying the input file. **Default:** false
POINT.RE	logical	Taurus MC implant parameter. Point response implant mode in which all ions are implanted into the central location of the exposed surface. **Default:** false
MAX.SPLI	logical	Taurus MC implant parameter. Maximum level of trajectory splits. **Default:** 5

Parameter	Type	Definition
SAVE1D	logical	Taurus MC implant parameter. Save1D* are a group of Taurus MC implant parameters:

```
SAVE1D(
    FILENAME=<STRING>
    ORIGINSHIFT=<BOOLEAN>
    HISTOGRAM=<BOOLEAN>
    UNITS=<STRING>
).
```

This group of parameters specifies whether to write and how to write 1D concentration-depth profiles in ASCII text files. This is valid for 1D or quasi-1D structures only. These files can be used for easy comparison with SIMS profiles. Specifies any of **SAVE1D**, **SAVE1D.FILE**, **SAVE1D.SHIFT**, s**AVE1D.HISTO**, and **SAVE1D.UNITS** will cause TSUPREM-4 to output the simulated 1D profiles.
Default: false

SAVE1D.F	character	Specifies the file name of the depth profile for the primary species. In addition, the profiles for all species will be written with the species' names as the extensions. See save1d for more information. **Default**: input file name with the extension replaced by the species' names. In addition, for primary species, input file name with the extension .imp will also be created.
SAVE1D.S	logical	Determines if the origin of the depth axis is the same as the simulation coordinate system (false) or at the top of the surface (true). **Default:** false
SAVE1D.H	logical	Determines if the concentration is stored in histogram format, i.e., at two locations (at the beginning and at the end of the respective depth range). **Default:** true
SAVE1D.U	character	One of A, nm, um, and cm. **Default:** um
OLD.MC	logical	Specifies that old TSUPREM-4 Monte Carlo (MC) implant model is used to perform the implantation. **Default:** false
VIBRATIO	logical	Specifies that lattice vibrations be included in the MC implant calculation. By default, a Debye calculation is performed to determine the amplitude of the displacement of silicon lattice atoms. The inclusion of this effect is an important determinant of dechanneling. Only applicable to the old TSUPREM-4 MC implant model. **Default:** true
X.RMS	number	The RMS amplitude of lattice vibrations of silicon atoms during MC implantation. The calculation of a default value for this parameter is described in Chapter 3, Crystalline Implant Model, p. 3-140. Only applicable to the old TSUPREM-4 MC implant model. **Units:** microns **Default:** a value calculated in the program

Parameter	Type	Definition
E.LIMIT	number	An empirical parameter specifying the ion energy above which the effect of lattice vibrations is ignored during a MC implantation. The calculation of a default value for this parameter is described in Chapter 3, Crystalline Implant Model, p. 3-140. Only applicable to the old TSUPREM-4 MC implant model. **Units:** keV **Default:** 5 keV for boron; 40keV for other ions
THRESHOL	number	The energy threshold that must be imparted to a silicon lattice site to generate a secondary that is not self-annealed. The calculation of a default value for this parameter is described in Chapter 3, Crystalline Implant Model, p. 3-140. Only applicable to the old TSUPREM-4 MC implant model. **Units:** eV **Default:** a value calculated in the program
POLY.GSZ	number	Specifies the polysilicon grain size in units of angstroms. Only applicable to the OLD.MC implant model. For Taurus MC, **POLY.GSZ** is specified in **MATERIAL** statement. **Units:** angstroms **Default**: none
INTERST	character	For implants into SiC only. Specifies which interstitial (either **silicon** or **carbon**) is to be plotted when selecting interstitial on the **SELECT** statement. Only applicable to the old TSUPREM-4 MC implant model. **Default**: silicon
MAX.DAMA	number	The maximum concentration of point defects in silicon. The final point defect concentrations is limited to this value. Only applicable to the old TSUPREM-4 MC implant model. **Units:** #/cm^3 **Default:** 5e22
REC.FRAC	number	The fraction of secondaries to be calculated by the MC implant calculation. Only applicable to the old TSUPREM-4 MC implant model. **Units:** none **Default:** a value calculated in the program

Description

The **IMPLANT** statement is used to simulate the implantation of impurities into the structure. The implanted impurity distribution can be calculated either numerically or analytically. For more detailed descriptions of the ion implantation models see, Chapter 3, Ion Implantation, p. 3-119.

Gaussian and Pearson Distributions

The analytic implant model uses either Gaussian or Pearson distributions. Both use an implant moment data file (*s4imp0*) or moments supplied on **`MOMENT`** statements, if the **`MOMENTS`** parameter is specified. The shapes of the functions used by the analytic calculations are defined by their first four moments:

- Range
- Standard Deviation
- Skewness (also called gamma)
- Kurtosis (also called beta)

The Gaussian implant uses the first two moments, while the Pearson distribution uses the first four. A dual-Pearson model uses two sets of four moments for the dual-Pearson function. In the lateral direction, a Gaussian distribution is used in all cases, with a characteristic length specified in the implant moment data file or on a **`MOMENT`** statement.

Table of Range Statistics

The **`IMPL.TAB`** parameter can be used to specify the table of range statistics from the implant moment data file. The following tables are currently available:

antimony	**default** data for antimony (energies: 5–1000 keV).
arsenic	data for arsenic (energies: 5–11000 keV).
dual.ars	**default** dual-Pearson data for arsenic (energies: 10–1000 keV).
tr.arsenic	dual-Pearson data for arsenic in <100> silicon with full energy, dose, tilt and rotation dependence, from University of Texas (energy: 0.5–180 keV; dose: 10^{13}–8×10^{15} atoms/cm^2; tilt: 0–10°; rotation: 0–45°).
bf2	data for boron from BF_2 source (energies: 5–120 keV).
dual.bf2	**default** dual-Pearson data for BF_2 (energies: 10–200 keV).
ut.bf2	dual-Pearson data for boron from a BF_2 source (energies: 15–120 keV).
tr.bf2	dual-Pearson data for boron from a BF_2 source in <100> silicon with full energy, dose, tilt, and rotation dependence, from University of Texas (energy: 0.5–65keV; dose: 10^{13}–8×10^{15} atoms/cm^2; tilt: 0–10°; rotation: 0–45°).
boron	original boron data with extended ranges fitted to results of amorphous MC calculations (energies: 5-4000 keV).
leboron	data for low-energy boron with channeling in silicon (energies: 10–30 keV).
chboron	**default** data for boron with channeling in silicon (energies: 5–2000 keV).
ut.boron	dual-Pearson data for boron (energies: 15–100 keV).

tr.boron	dual-Pearson data for boron in <100> silicon with full energy, dose, tilt and rotation dependence, from University of Texas (energy: 0.5–80 keV; dose: 10^{13}–8x10^{15}atoms/cm^2; tilt: 0–10° ; rotation: 0–45°).
scr.boron	dual-Pearson data for boron in <100> silicon with full energy, dose, tilt, rotation, and *screening oxide thickness* dependence. Data from The University of Texas at Austin (energy: 15 keV-80 keV; dose: 10^{13}–8×10^{15} atoms/cm^2; tilt: 0° –10° ; rotation: 0° – 45° ; oxide thickness: 15-400 Angstroms).
phosphorus	original phosphorus data with extended energy ranges fitted to results of amorphous MC calculations (energies: 5–7000 keV).
dual.pho	**default** dual-Pearson data for phosphorus with channeling in silicon (energies: 10–200 keV).
tr.phosphorus	dual-Pearson data for phosphorus into bare <100> silicon with full energy, dose, tilt, and rotation dependence (energy: 15–180 keV; dose: 10^{13}–8x10^{15} atoms/cm^2; **tilt:** 0°–10°; rotation: 0°–45°).
tr.*in*dium	**default** dual-Pearson data for indium in <100> silicon with full energy, dose, tilt, and rotation dependence (energy: 1–100 keV; dose: 10^{13}–8×10^{15} atoms/cm^2; tilt: 0° –10° ; rotation: 0° –45°). The 200 keV parameters are based on tilt=0 implants, while 300 keV are based on tilt=7 and rotation=30 implants.

The energy ranges shown are for implantation into silicon; the ranges may be different for other materials. The dual-Pearson model is used when one of the dual-Pearson distribution tables is specified with the **IMPL.TAB** parameter.

The default table for each impurity (except BF$_2$) can be changed with the **IMPL.TAB** parameter on the **IMPURITY** statement (see p. 4-273).

MC Implant Model

An alternative to the analytic implant calculation is provided by the MC based calculation. This model is physically based, and allows more general implant conditions and characteristics to be modeled. The MC calculation allows simulation of varying rotation angles and temperature dependence of damage self-annealing, and of reflection of ions from the surface.

When **BF2** is specified with **OLD.MC**, the BF$_2$ implant is approximated by a boron implant with an energy of 0.2215 times the specified energy.

Point Defect Generation

Generation of point defects during the implantation is specified by the **DAMAGE** parameter. **DAMAGE** defaults to true if the **PD.TRANS** or **PD.FULL** point defect model is in effect, and defaults to false if **PD.FERMI** is in effect. If **DAMAGE** is specified true and **PD.FERMI** is being used, the **PD.TRANS** point defect model is used instead. The damage model is controlled by the **D.PLUS**, **D.P1**, **D.SCALE**,

MAX.DAMA, and **D.RECOMB** parameters. Details of the damage model are given in Chapter 3, Implant Damage Model, p. 3-144.

Note:

> ***The DAMAGE parameter does not control the calculation of crystalline damage in the MC model. This parameter controls the retention or omission of point defect damage information for process steps subsequent to either analytic or MC implantations. Calculation of crystalline damage is controlled using the CRYSTAL parameter. To reduce the effects of damage in the MC model, use the MAX.DAMA parameter set to a small value, increase the temperature so that damage self-annealing becomes more dominant, or modify the final damage calculation using D.SCALE.***

Extended Defects

If the Extended Defects AAM is enabled, the creation of dislocation loops can be modeled by specifying **L.DENS** and **L.RADIUS** (see Chapter 3, Dislocation Loop Model, p. 3-82).

Channeling Effects

The default implant tables for arsenic, BF_2, and phosphorus include significant channeling effects. When implanting one of these impurities through a screening oxide, it may be preferable to use one of the tables that does not include channeling (i.e., *arsenic, bf2*, or *phosphorus).* Some users may wish to change the default implant table for these impurities (with the **IMPL.TAB** parameter on the **IMPURITY** statement).

Boundary Conditions

Ion implantation does not obey the reflecting boundary conditions that are usually used at the left and right edges of the structure. Instead, the analytic implantation extends the structure at a reflecting boundary out to infinity, while the MC model uses the boundary condition specified by your choice of the parameters **VACUUM**, **PERIODIC**, or **REFLECT**. Thus, there is a loss of accuracy in the implanted profile, unless the lateral spread of the implant distribution is small compared to the distance between the edge of the structure and the nearest mask edge. In some cases, you may need to reflect the structure before implantation and truncate it afterwards to ensure the accuracy of the implanted profile.

For MC implants, three boundary conditions are available.

- The default, **PERIODIC**, specifies that ions leaving one side of the structure re-enter on the other side, with the same velocity.

  This condition gives accurate answers for 1D structures and for structures where the sequence and thickness of layers are the same at the left and right edges.
- **REFLECT** specifies that ions hitting the edge of the structure are reflected back into the structure.

  This condition is accurate for **TILT=0** implants and Mfor pairs of implants with opposite tilts.
- **VACUUM** specifies that ions leaving the structure through the sides are lost.

This boundary condition is a poor approximation for most structures.
There are many situations (particularly single tilted implants into 2D structures) where none of the available boundary conditions is perfectly accurate. As with the analytical implant model, you lose some accuracy unless the lateral spread of the implant distribution is small compared to the distance between the edge of the structure and the nearest mask edge. Again, you may need to reflect the structure before implantation and truncate it afterwards to ensure the accuracy of the implanted profile.

Rotation Angle and Substrate Rotation

In addition to the rotation angle on implant statement, TSUPREM-4 allows you to specify substrate rotation on **`INITIALIZE`** statement with parameters **`ROT.SUB`** or **`X.ORIENT`**. **`ROT.SUB`** is defined as the angle between the X-axis of the simulation plane and the [110] primary flat on the wafer. **`ROTATION`** on **`IMPLANT`** statement is also referred to the [110] flat, measured in a clockwise direction about an axis perpendicular to and facing into the wafer surface. Implanting with a rotation angle of 0.0 corresponds to implantation in the plane formed by Y-axis and [110] flat direction. These two angles are added when calculating the geometric effect, i.e., shadowing effect. Therefore, both these rotation angles could affect the ion implantation shadowing, and implant rotation angle could also affect the channeling effect. For example, for the same tilt angle, an implant rotation angle of 75 deg with default substrate rotation is equivalent to the an implant rotation angle of 45 deg plus an substrate rotation of 30 deg, as far as the shadowing effect is concerned. However, the channeling behaviors could be different due to different implant rotation angles

TSUPREM-4 Version Considerations

The **`IMPLANT`** statement also accepts the parameters **`DX.MIN`** and **`DY.MAX`**. Before Version 5.2 of TSUPREM-4, these parameters were used to specify the spacings for a rectangular grid used internally in performing the implantation calculations. These parameters are still accepted for compatibility with older versions of the program, but their values are ignored.

When **`BACKSCAT`** is true, TSUPREM-4 uses the same model for backscattering of ions as in TMA SUPEREM-III. When **`BACKSCAT`** is false, the backscattering model is disabled, giving the same results as in versions of TSUPREM-4 before Version 5.2.

Note:
The effect of backscattering is usually small.

Examples

1. The following statement specifies a 100 keV implant of phosphorus with a dose of 10^{14} atoms/cm^2:

```
IMPLANT PHOSPH DOSE=1E14 ENERGY=100
IN.FILE=altmom
```

The Pearson model (the default) is used for the distribution function using moment data from the file *altmom*.

2. The following statement specifies a 50 keV implant of boron from a BF_2 source with a dose of 10^{13} atoms/cm^2:

```
IMPLANT BF2 DOSE=1E13 ENERGY=50 TILT=15
ROTATION=45
```

The default range statistics when the **BF2** parameter is specified are taken from the *dual.bf2* table in the standard moments file. The wafer is tilted at an angle of 15 degrees (clockwise from the horizontal) in the plane of the simulation and rotated by an angle of 45 deg (clockwise from the simulation plane, facing into the wafer) about the surface normal of the wafer. This means that the typical shadowing effect of the 15-deg tilt is moderated by the effect of the rotation out of the plane of the simulation. The tilting of the wafer with respect to the ion beam also leads to some degree of dose reduction and foreshortening of the resulting implant distribution in the simulation plane (see Chapter 3, Analytic Ion Implant Models, p. 3-119).

3. The following statement performs a boron implant of 2x10^{13} atoms/cm^2 at an energy of 500 keV with ion beam tilt and rotation of 7 and 30 deg, respectively, using the energy, dose, tilt, and rotation dependent data for boron in <100> silicon.

```
IMPLANT BORON DOSE=2E13 ENERGY=500 +
     IMPL.TAB=tr.boron TILT=7 ROTATION=30
PRINT
```

The **PRINT** parameter causes a summary of the implantation to be printed on the standard output and in the output listing file. The dual-Pearson model is used with the *tr.boron* moment tables.

4. The following statements show how the **MOMENT** statement can be used with the **IMPLANT** statement to specify the range statistics for an implant:

```
MOMENT SILICON RANGE=0.21 SIGMA=0.073 +
    GAMMA=-0.1 KURT=12

IMPLANT BORON DOSE=1E13 +
    ENERGY=50 TILT=15 MOMENTS
```

The **MOMENT** statement specifies the range statistics for implantation into silicon. Moments for implantation into other materials are read from the default moment data file.

5. In the following statement, the MC method (Taurus MC Implant) is used to implant arsenic with a dose of 10^{14} atoms/cm^2, an energy of 50 keV, and a tilt angle of 45° counterclockwise from vertical.

```
IMPLANT ARSENIC DOSE=1E14 ENERGY=50 MONTECAR +
             N.ION=10000 TILT=45 PERIOD
```

The number of ions is increased to 10,000 (from the default of 1000) to improve the accuracy in the tail of the implant profile. This implant uses the periodic boundary condition. Thus, ions that leave the simulation regime on one side of the structure re-enter at the other side.

6. The following statements perform symmetric plus and minus 7 degree tilted implants:

```
IMPLANT ARSENIC DOSE=5E13 ENERGY=50 MONTECAR +
        N.ION=10000 TILT=7 REFLECT
IMPLANT ARSENIC DOSE=5E13 ENERGY=50 MONTECAR +
        N.ION=10000 TILT=-7 REFLECT
```

   Specifying the reflecting boundary condition causes ions that exit from either side of the simulation regime to re-enter at the same point with the horizontal velocity reversed.

7. The following statement is similar to the previous example, except that ions that exit from either side re-enter on the opposite side with their velocity unchanged:

```
IMPLANT ARSENIC DOSE=1E14 ENERGY=50 MONTECAR +
             N.ION=10000 TILT=7 PERIODIC
```

   This corresponds to reproducing the simulation regime on each side, as in simulating one cell of a periodic structure.

8. The following statement simulates ion implantation using analytic models and generates damage for use as an initial condition for subsequent high-temperature processing steps:

```
IMPLANT ARSENIC DOSE=1E14 ENERGY=50 DAMAGE +
        D.PLUS=0.2
```

   The **PD.TRANS** model is enabled if it is not already in use. The *plus one* component of damage is reduced to 0.2 times the concentration of implanted arsenic.

9. The following statement simulates the 4 rotational implants starting in one implant command.

```
IMPLANT BORON DOSE=1E15 ENERGY=10 TILT=7 +
        ROTATION=10 MULT.ROT=4
```

   The above statement can be replaced with;

```
IMPLANT BORON DOSE=2.5E14 ENERGY=10 TILT=7 +
        ROTATION=10
IMPLANT BORON DOSE=2.5E14 ENERGY=10 TILT=7 +
        ROTATION=100
IMPLANT BORON DOSE=2.5E14 ENERGY=10 TILT=7 +
        ROTATION=190
IMPLANT BORON DOSE=2.5E14 ENERGY=10 TILT=7 +
        ROTATION=280
```

10. 5 angstroms shift of interstitial profile of Frenkel-pairs

```
IMPLANT BORON DOSE=1E14 ENERGY=10 DY.FP=0.0005
```

11. 0.01um Shift of +n interstitial profile

```
IMPLANT BORON DOSE=1E14 ENERGY=10 DY.IPLUS=0.01
```

12. -0.01um Shift of +1 Vacancy profile

```
IMPLANT BORON DOSE=1E14 ENERGY=10 +
        DY.VPLUS=-0.01 V.PLUS=1.0
```

13. 0.01um Shift of +n interstitials and -0.01um Shift of +1 vacancies

```
IMPLANT BORON DOSE=1E14 ENERGY=10 +
        DY.IPLUS=0.01 DY.VPLUS=-0.01 V.PLUS=1.0
```

DIFFUSION

The **`DIFFUSION`** statement is used to model high temperature diffusion in both oxidizing and nonoxidizing ambients. **`DIFFUSE`** is accepted as a synonym for the **`DIFFUSION`** statement.

```
DIFFUSION

    TIME=<n> [CONTINUE]
    [TEMPERAT=<n>] [ {T.RATE=<n> | T.FINAL=<n>} ]
    [ {  DRYO2 | WETO2 | STEAM | N2O | INERT
       | AMB.1 | AMB.2 | AMB.3 | AMB.4 | AMB.5
       | ( [F.O2=<n>] [F.H2O=<n>] [F.N2O=<n>]
           [F.H2=<n>] [F.N2=<n>] [F.HCL=<n>] )
      }
    ]
    [IMPURITY=<c> I.CONC=<c>]
    [ANTIMONY=<n>] [ARSENIC=<n>] [BORON=<n>] [PHOSPHOR=<n>]
    [PRESSURE=<n>] [ {P.RATE=<n> | P.FINAL=<n>} ] [HCL=<n>]
    [D.RECOMB=<n>] [MOVIE=<c>] [DUMP=<n>]
```

Parameter	Type	Definition
`TIME`	number	The duration of the diffusion step. **Units:** minutes **Default:** none
`CONTINUE`	logical	Indicates that this step is a continuation of a previous diffusion step. No native oxide deposition occurs, and the time step is not reset. No processing steps should be specified between the preceding **`DIFFUSION`** statement and the **`DIFFUSION CONTINUE`** statement. The starting temperature of the step should be the same as the final temperature of the preceding step, and the ambient must also be the same. **Default:** false
`TEMPERAT`	number	The ambient temperature at the beginning of the step. **Units:** ° Celsius **Default:** the last processing temperature specified
`T.RATE`	number	The time rate of change of the ambient temperature. **Units:** ° Celsius/minute **Default:** 0.0
`T.FINAL`	number	The ambient temperature at the end of the step. **Units:** ° Celsius **Default:** **`TEMPERAT+T.RATE×TIME`**
`DRYO2`	logical	Specifies that the ambient gas is dry oxygen. **Default:** false
`WETO2`	logical	Specifies that the ambient gas is wet oxygen. **Default:** false

Parameter	Type	Definition
STEAM	logical	Specifies that the ambient gas is steam. **Default:** false
N2O	logical	Specifies that the ambient gas is N_2O. **Default:** false
INERT	logical	Specifies that the ambient gas is inert. **Default:** true **Synonyms: NEUTRAL**, **NITROGEN**, **ARGON**
AMB.1	logical	Specifies that the ambient gas is ambient number one. Ambient number one is defined by the user with the **AMBIENT** statement. **Default:** false
AMB.2	logical	Specifies that the ambient gas is ambient number two. Ambient number two is defined by the user with the **AMBIENT** statement. **Default:** false
AMB.3	logical	Specifies that the ambient gas is ambient number three. Ambient number three is defined by the user with the **AMBIENT** statement. **Default:** false
AMB.4	logical	Specifies that the ambient gas is ambient number four. Ambient number four is defined by the user with the **AMBIENT** statement. **Default:** false
AMB.5	logical	Specifies that the ambient gas is ambient number five. Ambient number five is defined by the user with the **AMBIENT** statement. **Default:** false
F.O2	number	The flow of O_2 associated with the ambient gas. If H_2 is also present, the O_2 and H_2 is assumed to react completely to form H_2O. The flows of O_2 and H_2 are reduced, and the flow of H_2O is increased. **Units:** none **Default:** 0.0
F.H2O	number	The flow of H_2O associated with the ambient gas. If O_2 and H_2 are also present, the O_2 and H_2 are assumed to react completely to form H_2O. The flows of O_2 and H_2 are reduced, and the flow of H_2O is increased. **Units:** none **Default:** 0.0
F.N2O	number	The flow of N_2O associated with the ambient gas. **Units:** none **Default:** 0.0
F.H2	number	The flow of H_2 associated with the ambient gas. If O_2 is also present, the O_2 and H_2 are assumed to react completely to form H_2O. The flows of O_2 and H_2 are reduced, and the flow of H_2O is increased. **Units:** none **Default:** 0.0

Parameter	Type	Definition
`F.N2`	number	The flow of N2 (and other inert components) associated with the ambient gas. **Units:** none **Default:** 0.0
`F.HCL`	number	The flow of chlorine associated with the ambient gas. **Units:** none **Default:** 0.0
`IMPURITY`	character	The names of the impurities present in the ambient for in-diffusion into the structure through the exposed surface. If there are several impurities provided, their names should be enclosed in quotes and separated by the commas. **Default:** none
`I.CONC`	character	The concentrations of the specified impurities in the ambient. If there is only one impurity provided, then its concentration will be interpreted as a floating number. For several impurities, a list of their concentrations should be provided in the same order as the list of impurity names in the **`IMPURITY`** parameter. The list should be enclosed in quotes and the concentrations should be separated by the commas. **Units:** atoms/cm^3 **Default:** none
`ANTIMONY`	number	The concentration of antimony in the ambient gas at the surface of the structure. **Units:** atoms/cm^3 **Default:** 0.0 **Synonym:** **`SB`**
`ARSENIC`	number	The concentration of arsenic in the ambient gas at the surface of the structure. **Units:** atoms/cm^3 **Default:** 0.0 **Synonym:** **`AS`**
`BORON`	number	The concentration of boron in the ambient gas at the surface of the structure. **Units:** atoms/cm^3 **Default:** 0.0 **Synonym:** **`B`**
`PHOSPHOR`	number	The concentration of phosphorus in the ambient gas at the surface of the structure. **Units:** atoms/cm^3 **Default:** 0.0 **Synonym:** **`P`**
`PRESSURE`	number	The total pressure of the ambient gas at the start of the step. **Units:** atmospheres **Default:** the pressure specified in the corresponding **`AMBIENT`** statement, or 1.0 if flows are specified

Parameter	Type	Definition
P.RATE	number	The time rate of change of the ambient gas pressure. **Units:** atmospheres/minute **Default:** 0.0
P.FINAL	number	The ambient gas pressure at the end of the step. **Units:** atmospheres **Default: PRESSURE**
HCL	number	The percentage of chlorine present in the ambient gas. **Units:** percent **Default:** value calculated from **F.HCL** or specified on **AMBIENT** statement
D.RECOMB	number	The fraction of Frenkel pair implant damage remaining after initial recombination. **Units:** none **Default:** 0.0
MOVIE	character	A string of TSUPREM-4 commands to be executed at the beginning of each time step. Multiple input statements can be given, separated by semicolons (;). **Default:** no commands executed
DUMP	number	Write a solution file after every **DUMP** time step. The files are readable with the **LOADFILE** and **INITIALIZE** statements. The names are of the form s<*time*>, where <*time*> is the time in seconds from the start of the diffusion step. **Units:** none **Default:** no intermediate solutions saved

Description

This statement specifies a diffusion step, with or without oxidation. Any impurities present in the wafer are diffused. If the wafer is exposed to a gas, predeposition and/or oxidation can be performed. If an oxidizing ambient is specified and the **VISCOUS** oxidation model is in effect, reflow of surface layers occurs.

The duration of the step must be specified with the **TIME** parameter. The ambient temperature must be specified with the **TEMPERAT** keyword (unless **CONTINUE** is specified). For linear ramping of the temperature, specify either the ramp rate (with **T.RATE**) or the temperature at the end of the step (with **T.FINAL**).

Ambient Gas

The ambient gas used during the diffusion step can be specified in one of two ways:

- Specify one of the **DRYO2**, **WETO2**, **STEAM**, **INERT**, or **AMB.1** through **AMB.5** parameters. These select an ambient that has been predefined with the **AMBIENT** statement. The **DRYO2**, **WETO2**, **STEAM**, and **INERT** ambients are defined by the standard initialization file; the **AMB.1** through **AMB.5** ambients must be defined by the user before they are used. The predefined ambients include a default pressure and HCl percentage, which can be overridden with the **PRESSURE** and **HCL** parameters, respectively, on the **DIFFUSION** statement.

- Define the ambient by specifying the flows of oxidizing (O_2 and H_2O) and nonoxidizing (H_2, N_2, and HCl) species. The flows can be specified as flow rates, fractions, or percentages, but the units of all the flows in a single **`DIFFUSION`** statement are assumed to be the same.

Any O_2 and H_2 in the gas are assumed to react (two units of H_2 for each unit of O_2) to form H_2O. Thus, the effective flow contains zero units of O_2 or H_2 (or both). If the effective gas contains nonzero amounts of both O_2 and H_2O, the oxidation rate is based on the partial pressure of H_2O.

Ambient Gas Parameters

The amount of chlorine in the ambient can be specified either by the flow of HCl (**`F.HCL`** parameter) or by the percentage of HCl (**`HCL`** parameter), but not both. If the specified ambient contains O_2 or H_2O, oxidation takes place at interfaces between silicon dioxide and silicon or polysilicon. A native oxide (with thickness given by the **`INITIAL`** parameter on the **`AMBIENT`** statement) is deposited on any exposed silicon or polysilicon surfaces before the start of the diffusion step. **`ANTIMONY`**, **`ARSENIC`**, **`BORON`**, and **`PHOSPHOR`**, and the combination of **`IMPURITY`** and **`I.CONC`** specify the concentrations of impurities at the wafer surface for predeposition. If multiple impurities are specified, their concentrations should be specified as a list of numbers with **`I.CONC`** parameter. The order of concentrations in the list should follow the order of the impurity names, specified with the **`IMPURITY`** parameter. The total pressure for an oxidizing ambient is given by **`PRESSURE`**. To ramp the pressure, specify either the ramp rate (**`P.RATE`**) or the pressure at the end of the step (**`P.FINAL`**).

The parameters for oxidation are set by the **`AMBIENT`** statement. Diffusivities and segregation parameters are set on the various impurity statements (i.e., the **`IMPURITY`**, **`ANTIMONY`**, **`ARSENIC`**, **`BORON`**, and **`PHOSPHORUS`** statements). The oxidation and point defect models and the numerical methods to be used are specified on the **`METHOD`** statement. The default values for these parameters are normally set by the *s4init* file, which is read each time TSUPREM-4 is executed.

See Chapter 3 for complete descriptions of the models used for diffusion and oxidation and Appendix A for a list of default model coefficients.

Oxidation Limitations

Oxidation of polysilicon is simulated only when the **`COMPRESS`**, **`VISCOELA`**, or **`VISCOUS`** model has been specified (with the **`METHOD`** statement).

The oxidation algorithms provide limited support for the case where silicon (or polysilicon), oxide, and a third material meet at a point. The results are reasonably accurate when only one of the materials in contact with oxide is oxidizing; results are less accurate if both materials in contact with oxide are oxidizing at a significant rate.

Impurities present in the ambient during an oxidation step are incorporated into the growing oxide. *Note*, however, that the program does not currently contain models for the changes in physical properties of heavily doped glasses.

Reflow

Reflow of surface layers occur whenever oxidation with the **`VISCOUS`** model is specified. The amount of reflow is proportional to the ratio of the surface tension (specified by the **`SURF.TEN`** parameter on the **`MATERIAL`** statement) to the viscosity for each material. Reflow can occur in any material having a nonzero value of **`SURF.TEN`**. Only exposed layers flow due to surface tension, but underlying layers can deform due to stresses produced by reflow of the exposed layers.

Reflow in an inert ambient can be approximated by specifying an oxidizing ambient with a negligible partial pressure of oxidant (e.g., by setting **`PRESSURE`**=1e-6 or by a combination such as **`F.N2`**=1.0 and **`F.O2`**=1e-6). *Note* that a native oxide is added whenever the partial pressure of the oxidizing species is nonzero; it may be necessary to do a shallow etch to remove this oxide following a reflow step.

Examples

1. The following statement specifies a 1000-deg, 30-min boron predeposition:

   ```
   DIFFUSION TIME=30 TEMP=1000 BORON=1E20
   ```

2. The following statement calls for a 30-min diffusion in an inert ambient:

   ```
   DIFFUSION TIME=30 TEMP=800 T.FINAL=1000 INERT
   ```

   The temperature is ramped from 800° to 1000° C during the step.

3. The following statement calls for a 60-min dry oxidation at 900° C with an ambient containing 2% HCl

   ```
   DIFFUSION TIME=60 TEMP=900 DRYO2 HCL=2
   ```

4. The following statement performs a 30-min, 1000° C diffusion:

   ```
   DIFFUSION TIME=30 TEMP=1000 +
        MOVIE="SELECT Z=log10(Boron)
               PLOT.1D X.V=1.0"
   ```

   The boron concentration is plotted before each time step.

EPITAXY

The **EPITAXY** statement is used to grow silicon material on the exposed silicon surface of the current structure.

```
EPITAXY

   TIME=<n> TEMPERAT=<n> [ {T.RATE=<n> | T.FINAL=<n>} ]
   [IMPURITY=<c>
    {  IT.CONC=<c> [IB.CONC=<c>] [LOG.GRAD]
     | I.PROFIL=<c> | I.RESIST=<n> } ]
   [ANTIMONY=<n>] [ARSENIC=<n>] [BORON=<n>] [PHOSPHOR=<n>]
   [ {CONCENTR | RESISTIV} ]
  THICKNES=<n> [SPACES=<n>] [DY=<n>] [YDY=<n>] [ARC.SPAC=<n>]
   [AR.MIN=<n>] [SELECTIV] [MOVIE=<c>] [BLANKET]
   [DIFFUSED] [STRAINED]
```

Parameter	Type	Definition
TIME	number	The duration of the epitaxy step. **Units**: minutes **Default**: none
TEMPERAT	number	The ambient temperature at the beginning of the step. **Units**: ° Celsius **Default**: none
T.RATE	number	The time rate of change of the ambient temperature. **Units**: ° Celsius/minute **Default**: 0.0
T.FINAL	number	The ambient temperature at the end of the step. **Units**: ° Celsius **Default**: **TEMPERAT**
IMPURITY	character	The names of the impurities with which the epitaxial layer is doped. If there are several impurities provided, their names should be enclosed in quotes and separated by the commas. **Defaul**t: none
IT.CONC	character	The concentrations of the specified impurities in the epitaxial layer. If there is only one impurity provided, then its concentration will be interpreted as a floating number. For several impurities, a list of their concentrations should be provided in the same order as the list of impurity names in the **IMPURITY** parameter. The list should be enclosed in quotes and the concentrations should be separated by the commas. **Units**: atoms/cm^3 **Default**: none

Parameter	Type	Definition
IB.CONC	character	The impurity concentrations at the bottom of the grown layer of the impurity species listed in the **IMPURITY** parameter. The impurity concentrations at the top of the layer must be specified by the **IT.CONC** parameter. **Units**: atoms/cm^3 **Default**: uniform impurity concentrations specified by **IT.CONC**
LOG.GRAD	logical	Specifies that the impurity concentrations specified by **IMPURITY** will vary logarithmically between the values specified by **IB.CONC** and **IT.CONC**. **Default:** false
I.PROFIL	character	The expression string for an equation that specifies the concentration of the single impurity specified by **IMPURITY** as a function of `F`, the fraction of the total layer thickness (or epitaxy time). `F` must obey the rules for a **DEFINE** variable; it is recommended that it always by enclosed in braces and preceded by the "`@`" character; i.e., it should always appear as "`@{F}`" in the equation. **Units:** atoms/cm^3 **Default**: none
I.RESIST	number	The resistivity produced by the presence of `IMPURITY` in the ambient gas at the surface of the wafer. **Units**: ohm-cm **Default**: none
ANTIMONY	number	The uniform concentration or resistivity of antimony in the deposited layer. **Units**: atoms/cm^3 or ohm-cm **Default**: zero concentration **Synonym**: **SB**
ARSENIC	number	The uniform concentration or resistivity of arsenic in the deposited layer. **Units**: atoms/cm^3 or ohm-cm **Default**: zero concentration **Synonym**: **AS**
BORON	number	The uniform concentration or resistivity of boron in the deposited layer. **Units**: atoms/cm^3 **Default**: zero concentration **Synonym**: **B**
PHOSPHOR	number	The uniform concentration or resistivity of phosphorus in the deposited layer. **Units**: atoms/cm^3 or ohm-cm **Default**: zero concentration **Synonym**: **P**
CONCENTR	logical	Specifies that the impurity concentration in the deposited layer is given. **Default**: true
RESISTIV	logical	Specifies that the resistivity of the deposited layer is given. **Default**: false
THICKNES	number	The thickness of the deposited layer. **Units**: microns **Default**: none

Parameter	Type	Definition
SPACES	number	The number of vertical grid spacings in the layer. This value is divided by the value of **GRID.FAC** (see MESH, p. 4-53). **Units**: none **Default**: 1/**GRID.FAC** **Synonym**: **DIVISION**
DY	number	The nominal grid spacing to be used in the deposited material layer at the location specified by the **YDY** parameter. This value is multiplied by the value of **GRID.FAC** (see MESH, p. 4-53). **Units**: microns **Default**: **GRID.FAC*THICKNES/SPACES** **Synonym**: **DX**
YDY	number	The location of the nominal grid spacing specified by **DY**, relative to the top of the deposited layer. **Units**: microns) **Default**: 0.0 **Synonym**: **XDX**
ARC.SPAC	number	The maximum spacing allowed along an arc on the new surface. This value is multiplied by the value of **GRID.FAC** (see MESH, p. 4-53). **Units**: microns **Default**: 0.5***THICKNES*GRID.FAC**
AR.MIN	number	The minimum aspect (width/height) ratio for deposited grid spaces on planar surfaces. **Default**: 0.5
SELECTIV	logical	Specifies that material will only be grown on silicon or polysilicon surfaces. **Default**: false
MOVIE	character	A string of TSUPREM-4 commands to be executed at the beginning of each time step. Multiple input statements can be given, separated by semicolons (;). **Default**: no commands executed
BLANKET	logical	Specifies that grown material will be single-crystal silicon, regardless of the underlying material. **Default**: false
DIFFUSED	logical	Specifies to solve the diffusion equations. **Default**: true
STRAINED	logical	Specifies to add lattice mismatch strains to epitaxial layers. Only applicable with the **VISCOEL** and **ST.HIST** models. **Default**: false

Description

This statement provides a basic epitaxy capability. Silicon is deposited on the exposed surface of the structure, and its upper surface becomes the new exposed surface. A uniform concentration of each dopant can be specified, either directly or by the resistivity of the grown material. Impurities diffuse according to the models in Chapter 3, Diffusion, p. 3-14.

By default, (single-crystal) silicon is grown on (single-crystal) silicon surfaces, and polysilicon is grown on all other surface materials. If `SELECTIV` is set true, growth occurs only on silicon and polysilicon surfaces and no growth occurs on other materials. If `BLANKET` is set true, then all grown material will be single-crystal silicon, regardless of the underlying material. Setting `BLANKET` true and leaving `SELECTIV` false causes a uniform layer of single-crystal silicon to be grown over the entire surface, as in older versions of the program.

The deposited material conforms to the contours of the original surface. Outside corners on the original surface produce arcs on the new surface, which are approximated by straight line segments. The maximum segment length is set by the `ARC.SPAC` parameter.

The `SPACES`, `DY`, and `YDY` parameters used to control the grid spacing in the deposited layer are scaled by the value of the `GRID.FAC` parameter on the `MESH` statement (see Chapter 3, Changes to the Mesh During Processing, p. 3-7).

The `EPITAXY` statement works by alternately depositing layers and diffusing impurities in the resulting structure. The number of deposit/diffuse steps is equal to the number of grid spaces in the deposited layer (i.e., it is controlled by the `SPACES`, `DY`, and `YDY` parameters). The horizontal spacing in a thin deposited layer is the same as in the substrate, but for thick layers it is increased to avoid mesh elements with high ratios of height to width. This increase in horizontal spacing can be controlled with the `AR.MIN` parameter.

If multiple impurities are specified, their concentrations should be specified as a list of numbers with `I.CONC` parameter. The order of concentrations in the list should follow the order of the impurity names, specified with the `IMPURITY` parameter

Gradient-Doped Growth

If `IB.CONC` is specified, the impurities named in the `IMPURITY` parameter will have the concentrations `IB.CONC` at the bottom of the deposited or grown layer and the concentrations `IT.CONC` at the top of the layer. The concentration will vary linearly through the layer, unless `LOG.GRAD` is specified; in this case, the variation will be logarithmic. `IB.CONC` can only be used when `IMPURITY` and `IT.CONC` (alias `I.CONC`) are also specified.

If `I.PROFIL` is specified, it is interpreted as an equation specifying the concentration of a single impurity (specified by `IMPURITY`) as a function of `F`, the fraction of the layer thickness (`F`=0 at the bottom of the layer and `F`=1 at the top). The parameter `F` must obey the rules for a `DEFINE` variable; it is recommended that it be enclosed in braces and preceded by the "`@`" character, i.e., "`@{F}`".

Note:

> ***The doping at interior nodes in the deposited layer is calculated by linear or logarithmic interpolation or by `I.PROFIL`. Thus, the total dose will be expected to differ from that calculated by analytical integration of the specified gradient (except in the special case of a linear gradient and a uniformly spaced grid).***

Example

1. The following statement deposits 1 micron of silicon while simultaneously diffusing at 1100° C:

   ```
   EPITAXY THICK=1.0 TIME=180 TEMPERAT=1100 +
            ANTIMONY=1E19 SPACES=10
   ```

   The deposition and diffusion processes are subdivided into 10 steps corresponding to the value specified by the **`SPACES`** parameter.

2. Epitaxial growth with linear gradient boron profile:

   ```
   EPITAXY TIME=10 TEMP=600 THICK=0.1 IMP=BORON +
           IB.CONC=1E14 IT.CONC=1E15 SPACES=20
   ```

3. Same as 2. But use **`I.PROFILE`**:

   ```
   EPITAXY TIME=10 TEMP=600 THICK=0.1 IMP=BORON +
           I.PROFIL=( 5E12*(5E13/5E12)^@{F} ) SPACES=20
   ```

4. Epitaxial growth with logarithmic gradient boron profile:

   ```
   EPITAXY TIME=10 TEMP=600 THICK=0.1 IMP=BORON +
           IB.CONC=1E14 IT.CONC=1E15 SPACES=20 LOG.GRAD
   ```

5. Epitaxy with logarithmic gradient multiple species:

   ```
   EPITAXY THICK=0.2 SPACES=10 TIME=10 TEMP=600 +
           IMP="BORON, PHOSPHORUS" +
           IB.CONC="5E13 5E15" +
           I.CONC="5E14 2E15" LOG.GRAD
   ```

Note:

The mobility tables used to calculate the doping from the resistive are not the same as the tables used by the `ELECTRICAL` statement. The extracted sheet resistance for the epitaxial layer does not correspond exactly to the resistivity specified during epitaxy.

STRESS

The **`STRESS`** statement solves the stress equation only to calculate the stresses caused by thermal mismatch between materials, imbalanced forces, or due to external pressure for applications in chip packaging.

```
STRESS

{   ( TIME=<n> TEMPERAT=<n>
        [ {T.RATE=<n> | T.FINAL=<n>} ]
        [MOVIE=<c>])
  | ( [ P.TOP=<n>] [P.SIDE=<n>] )
  | RELAX
}
```

Parameter	Type	Definition
`TIME`	number	The duration of the stress step. **Units:** minutes **Default:** none
`TEMPERAT`	number	The ambient temperature at the beginning of the step. **Units:** ° Celsius **Default:** none
`T.RATE`	number	The time rate of change of the ambient temperature. **Units:** ° Celsius/minute **Default:** 0.0
`T.FINAL`	number	The ambient temperature at the end of the step. **Units:** ° Celsius **Default:** **`TEMPERAT+T.RATE×TIME`**
`MOVIE`	character	A string of TSUPREM-4 commands to be executed at the beginning of each time step. Multiple input statements can be given, separated by semicolons (;). **Default**: no commands executed
`P.TOP`	number	The applied pressure on the top exposed surface. **Units:** dynes/cm^2 **Default:** 0.0
`P.SIDE`	number	The applied pressure on the right side surface. **Units:** dynes/cm^2 **Default:** 0.0
`RELAX`	logical	Specifies to perform the stress relaxation **Default:** false

Description

This statement calculates stresses due to thermal mismatch between materials, imbalanced forces, or external pressure for chip packaging applications. The

STRESS command is applicable only when the Boolean parameters **ST.HIST** and **VISCOEL** are switched on in the **METHOD** statement.

Thermal Mismatch

For thermal mismatch stresses, only stress equations are solved for the specified temperature change. Parameter **TEMPERATURE** specifies the initial temperature. **TIME** specifies the duration time of the step, **T.RATE** specifies the temperature change rate, and **T.FINAL** specifies the final temperature of the ramp. If the duration time is zero, or the temperature change rate is zero, or the final temperature is equal to the initial temperature. **STRESS** command updates only the stress values by solving the stress equations while the values of the other solutions are kept. Thus, the temperature at which stress values are calculated, hereinafter '*stress temperature*' must be distinguished from the diffusion temperature. During all thermal processes but the **STRESS** command, the stress temperature is always identical to the diffusion temperature. Both the diffusion temperature and the stress temperature are saved to an output file.

```
DIFFUSION TEMP=@DTEMP1 TIME=1 ;$(1)
STRESS    TEMP=@STEMP1 T.FINAL=@STEMP2 TIME=1 ;$(2)
DIFFUSION TEMP=@DTEMP2 TIME=1 ;$(3)
```

- At (1), the diffusion equations and the stress equations are solved at **DTEMP1**. (stress temperature = diffusion temperature = **DTEMP1**).
- At (2), when **STEMP1** is different from **DTEMP1**, the stress equations are solved first for the temperature ramp from **DTEMP1** to **STEMP1** before processing from **STEMP1** to **STEMP2**. The final stress temperature is **STEMP2** while the diffusion temperature keeps **DTEMP1**, which means all equilibrium values associated with diffusion are calculated at **DTEMP** , while the stress are calculated at **STEMP2**. (stress temperature = **STEMP2**, diffusion temperature = **DTEMP1**).
- At (3), When **STEMP2** is different from **DTEMP1**, the stress equations are solved first for the temperature ramp from **STEMP2** to **DTEMP1** before processing the diffusion step. Then, in processing the diffusion step, when **DTEMP2** is different from **DTEMP1**, the diffusion equations and the stress equations are solved for the temperature ramp from **DTEMP1** to **DTEMP2** before processing 1-minute diffusion at **DTEMP2**. (stress temperature = diffusion temperature = **DTEMP2**).
- The time step for the internal temperature ramp is specified by the **TIME.DEL** parameter in the **METHOD** statement.
- If **TEMPERAT** is not specified, the latest stress temperature is used for initial temperature.

Relaxation

Simply specifying the **RELAX** parameter forces the instantaneous stress relaxation:

```
STRESS RELAX
```

External Pressure

For stresses due to pressure boundary condition, **`P.TOP`** specifies the value of the applied pressure on the top exposed surface, and **`P.SIDE`** specifies the value of the applied pressure on the right side surface. To apply the pressure boundary condition, the original reflective boundary condition is removed. This feature is intended for chip packaging application only and the drastic change of boundary condition can lead to multiple complications depending on the applications.

Printing and Plotting of Stresses and Displacements

The calculated stresses can be accessed for printing or plotting by referencing the variables *Sxx*, *Syy*, and *Sxy* on the **`SELECT`** statement; the calculated *x* and *y* displacements can be referenced as *x.vel* and *y.vel*. The stresses and displacements can also be displayed with the **`STRESS`** and **`FLOW`** parameters on the **`PLOT.2D`** statement.

Example

The following statement calculates the thermal mismatch stress between materials due to a temperature ramp from 50° to 1050° C with duration time of 1 minute:

```
STRESS TIME=1 TEMPERATURE=50 T.FINAL=1050
```

The Following statement calculates the stresses due to packaging pressure of 500e6 dynes/cm^2 applied on the top surface:

```
STRESS P.TOP=500e6
```

Output

The following statements print and plot results:

Statement Name	Description	Page
`SELECT`	Evaluates the quantity to be printed or plotted; specifies titles and axis labels.	p. 4-146
`PRINT.1D`	Prints values of a quantity along a line through the structure.	p. 4-151
`PLOT.1D`	Plots a quantity along a line through the structure.	p. 4-155
`PLOT.2D`	Plots axes and boundaries for 2D structure plots; plots grid and/or velocity and stress vectors.	p. 4-163
`CONTOUR`	Plots contours in two dimensions.	p. 4-168
`COLOR`	Fills areas of a 2D plot.	p. 4-170
`PLOT.3D`	Plots a 3D projection plot of the selected quantity.	p. 4-172
`LABEL`	Adds labels to a plot.	p. 4-175
`EXTRACT`	Extracts information about a structure, and prints it or writes it to a file.	p. 4-180
`ELECTRICAL`	Extracts electrical information, and prints it or writes it to a file.	p. 4-194
`VIEWPORT`	Specifies a subset of the plotting surface to plot on.	p. 4-204

SELECT

The **`SELECT`** statement evaluates a quantity to be printed or plotted. It is also used to specify plot titles and axis labels.

```
SELECT

   [Z=<c>] [TEMPERAT=<n>] [LABEL=<c>] [TITLE=<c>]
```

Parameter	Type	Definition
`Z`	character	A mathematical expression defining the quantity to be printed or plotted. If the expression contains spaces, it must be enclosed in parentheses. **Default:** “0”
`TEMPERAT`	number	The temperature at which the solution is to be evaluated. **Units:** degrees Celsius **Default:** last specified temperature or 800
`LABEL`	character	The label to be used on the *y* axis of a 1D plot, or the *z* axis of a three-dimensional plot. **Default:** the expression given by **`Z`**
`TITLE`	character	The title to be used on plots. **Default:** the name and version number of the program

Description

The **`SELECT`** statement evaluates the quantity to be displayed by the **`CONTOUR`**, **`PLOT.1D`**, **`PRINT.1D`**, **`PLOT.2D`**, and **`PLOT.3D`** statements or to be extracted by the **`EXTRACT`** statement. No solution data can be printed or plotted until this statement is specified. The values calculated by a **`SELECT`** statement are used until another **`SELECT`** statement is specified. If the solution changes, a new **`SELECT`** statement is given in order for the new values to be printed or plotted.

Solution Values

The **`Z`** parameter specifies a mathematical expression for the quantity to be plotted. The following solution values can be used in the expression, provided that the solution is available:

antimony	antimony concentration (atoms/cm^3)
arsenic	arsenic concentration (atoms/cm^3)
boron	boron concentration (atoms/cm^3)
phosphorus	phosphorus concentration (atoms/cm^3)
indium	indium concentration (atoms/cm^3)
i_antimony	antimony interface trap density (atoms/cm^2)
i_arsenic	arsenic interface trap density (atoms/cm^2)

i_boron	boron interface trap density (atoms/cm^2)
i_phosphorus	phosphorus interface trap density (atoms/cm^2)
c0_antimony	antimony precipitation concentration (atoms/cm^3)
c0_arsenic	arsenic precipitation concentration (atoms/cm^3)
c0_boron	boron precipitation concentration (atoms/cm^3)
c0_phosphor	phosphorus precipitation concentration (atoms/cm^3)
c1_antimony	antimony-interstitial cluster concentration (atoms/cm^3)
c1_arsenic	arsenic-interstitial cluster concentration (atoms/cm^3)
c1_boron	boron-interstitial cluster concentration (atoms/cm^3)
c1_phosphor	phosphorus-interstitial cluster concentration (atoms/cm^3)
c2_antimony	antimony-vacancy cluster concentration (atoms/cm^3)
c2_arsenic	arsenic-vacancy cluster concentration (atoms/cm^3)
c2_boron	boron-vacancy cluster concentration (atoms/cm^3)
c2_phosphor	phosphorus-vacancy cluster concentration (atoms/cm^3)
ipair_antimony	antimony-interstitial pair concentration (atoms/cm^3)
ipair_arsenic	arsenic-interstitial pair concentration (atoms/cm^3)
ipair_boron	boron-interstitial pair concentration (atoms/cm^3)
ipair_phosphor	phosphorus-interstitial concentration (atoms/cm^3)
vpair_antimony	antimony-vacancy pair concentration (atoms/cm^3)
vpair_arsenic	arsenic-vacancy pair concentration (atoms/cm^3)
vpair_boron	boron-vacancy pair concentration (atoms/cm^3)
vpair_phosphor	phosphorus-vacancy concentration (atoms/cm^3)
doping	net active concentration (atoms/cm^3)
oxygen	oxidant concentration (atoms/cm^3)
silicon	silicon concentration (in silicide) (atoms/cm^3)
interstitial	interstitial concentration (#/cm^3)
vacancy	vacancy concentration (#/cm^3)
damage	damage concentration (#/cm^3)
ci.star	equilibrium interstitial concentration (#/cm^3)
cv.star	equilibrium vacancy concentration (#/cm^3)
trap	concentration of filled interstitial traps (#/cm^3)
cl_interst	concentration of clustered interstitials (#/cm^3)
cl_im0	concentration of interstitial clusters (#/cm^3)
dloop	density of dislocation loops (#/cm^3)

rloop	radius of dislocation loops (cm)
lgrain	average polycrystalline grain size (um)
electron	concentration of electrons (#/cm^3)
x.v	*x* velocity (cm/sec)
y.v	*y* velocity (cm/sec)
Sxx, Sxy, Syy	components of the stress tensor (dynes/cm^2)
x	*x* coordinate (microns)
y	*y* coordinate (microns)

The names of user-specified impurities can also be used, giving the concentration of the impurity in atoms/cm^3.

- The chemical concentration is reported unless the active function is used, for example, *active(phosphorus)*.
- The net concentration is defined as the sum of the donor concentrations minus the sum of the acceptor concentrations.
- The electron concentration is calculated using the assumptions of local charge neutrality and complete ionization of impurities.
- The dopant-interstitial and dopant-vacancy pair concentration in local equilibrium are calculated by, for example, *eqipair(boron)* and *eqvpair(boron)* for boron.
- Any intermediates.

Mathematical Operations and Functions

The symbols “+”, “-”, “*”, “/”, and “^” are used for the mathematical operations of addition, subtraction, multiplication, division, and exponentiation, respectively. In addition, the following functions are available:

abs	absolute value
active	electrically active part of impurity concentration
asin, acos, atan	inverse trigonometric functions (results in radians)
diffusivity	diffusivity (in cm^2/sec) of an impurity or point defect species
eqactive	electrically active part of impurity concentration in equilibrium
eqipair	dopant-interstitial pair concentration of an impurity in local equilibrium
eqvpair	dopant-vacancy pair concentration of an impurity in local equilibrium
erf	error function

erfc	complementary error function
exp	exponential
gb	concentration of impurity in polycrystalline grain boundaries
log	natural logarithm of the absolute value
log10	base-10 logarithm of the absolute value
sinh, asinh, cosh, acosh, tanh, atanh	hyperbolic and inverse hyperbolic functions
slog10	base-10 logarithm of the absolute value times the sign of the value
sqrt	square root
sin, cos, tan	trigonometric functions (arguments in radians)

The *log, log10,* and *slog10* functions return the value 0.0 if their argument is zero; the *log*, *log10*, *slog10*, and *sqrt* functions take absolute value of their arguments. The following constant is available:

Kb	Boltzmann's constant (eV/° C)

Note:

The active and net concentrations depend on the temperature. If `TEMPERAT` is not specified, the last processing temperature is used. If the last process step ended with a ramp to a low temperature, you may need to specify a higher value of `TEMPERAT` in order to obtain realistic levels of dopant activation.

Examples

1. The logarithm (base 10) of the arsenic concentration is evaluated with:

```
SELECT Z=log10(Arsenic)
```

2. The phosphorus concentration minus a constant profile of $5x10^{14}$ is evaluated with:

```
SELECT Z=(Phosphorus - 5.0e14)
```

3. The difference between the phosphorus concentration and an analytic profile is evaluated with:

```
SELECT Z=(phos - 1.0e18 * exp ( y * y ) )
```

4. The excess vacancy-interstitial product is evaluated with:

```
SELECT Z=(inter * vacan - ci.star * cv.star)
```

5. The diffusivity (in cm^2/sec) of boron at each point in the structure is evaluated with:

```
SELECT Z=diffusivity(Boron)
```

Note:

When the `PD.TRANS` or `PD.FULL` model for point defects is used, the diffusivity can be different at each point in the structure and may vary with time.

6. The boron-interstitial pair concentration in local equilibrium is evaluated with:

```
SELECT Z=eqipair(Boron)
```

Note:

eqipair() and eqvpair() give the dopant-defect pair concentrations in the 3-stream diffusion model i.e. `PD.TRANS` or `PD.FULL` model. Thus, it is useful to see the difference of dopant-defect pair concentrations between the 5- and 3- stream diffusion models.

7. The following statements print junction depths

```
SELECT Z=doping
PRINT LAYERS X.V=0
```

The **`PRINT LAYERS`** statement assumes that a new layer begins whenever the selected value (net doping in this case) changes sign.

8. The following statements print the thicknesses of material layers:

```
SELECT Z=1.0
PRINT LAYERS X.V=0
```

The **`SELECT`** statement specifies a constant value of one, so the **`PRINT`** statement only uses material boundaries to define layers. Further, when the value “1.0” is integrated over each layer, the result is just the layer thickness.

9. The following statement specifies the title to be used on the next plot:

```
SELECT TITLE="Final N-Channel Structure"
```

Because no **`Z`** value is specified, **`Z=0`** is assumed, and any attempt to print or plot solution data uses the value zero.

10. The following statements print the pressure:

```
INTERMED NAME=pressure EXPR=-(SXX+SYY+SZZ)/3
SELECT   Z=pressure
PRINT.1D X.V=0
```

PRINT.1D

The **`PRINT.1D`** statement prints the value of the selected expression along a line through the structure. It can also print layer thickness and integrated doping information.

```
PRINT.1D

   {  X.VALUE=<n> | Y.VALUE=<n>
    | ( {  MATERIAL=<c> | SILICON | OXIDE | OXYNITRI | NITRIDE
         | POLYSILI | PHOTORES | ALUMINUM
        }
        {  /MATERIA=<c> | /SILICON | /OXIDE | /OXYNITR | /NITRIDE
         | /POLYSIL | /PHOTORE | /ALUMINU | /AMBIENT | /REFLECT
        }
      )
   }
   [OUT.FILE=<c> [APPEND] [{CM | UM | NM}]
   [SPOT=<n>] [LAYERS]
   [X.MIN=<n>] [X.MAX=<n>]
```

Parameter	Type	Definition
X.VALUE	number	The *x* coordinate of a vertical section along which values are to be printed. **Units:** microns **Default:** 0.0
Y.VALUE	number	The *y* coordinate of a horizontal section along which values are to be printed. **Units:** microns **Default:** none
MATERIAL	character	Print values in the named material, at the interface with the other specified material. **Default:** none
SILICON	logical	Print values in silicon, at the interface with the other specified material. **Default:** false
OXIDE	logical	Print values in oxide, at the interface with the other specified material. **Default:** false
OXYNITRI	logical	Print values in oxynitride, at the interface with the other specified material. **Default:** false
NITRIDE	logical	Print values in nitride, at the interface with the other specified material. **Default:** false
POLYSILI	logical	Print values in polysilicon, at the interface with the other specified material. **Default:** false
PHOTORES	logical	Print values in photoresist, at the interface with the other specified material. **Default:** false

Parameter	Type	Definition
`ALUMINUM`	logical	Print values in aluminum, at the interface with the other specified material. **Default:** false
`/MATERIA`	character	Print values in the other specified material, at the interface with the named material. **Default:** none
`/SILICON`	logical	Print values in the other specified material, at the interface with silicon. **Default:** false
`/OXIDE`	logical	Print values in the other specified material, at the interface with oxide. **Default:** false
`/OXYNITR`	logical	Print values in the other specified material, at the interface with oxynitride. **Default:** false
`/NITRIDE`	logical	Print values in the other specified material, at the interface with nitride. **Default:** false
`/POLYSIL`	logical	Print values in the other specified material, at the interface with polysilicon. **Default:** false
`/PHOTORE`	logical	Print values in the other specified material, at the interface with photoresist. **Default:** false
`/ALUMINU`	logical	Print values in the other specified material, at the interface with aluminum. **Default:** false
`/AMBIENT`	logical	Print values in the other specified material, at the interface with the exposed surface (if any). **Default:** false **Synonym:** **`/EXPOSED`**, **`/GAS`**
`/REFLECT`	logical	Print values in the other specified material, at the interface with the reflecting boundary (if any). **Default:** false
`OUT.FILE`	character	The name of the output file in which the printed profile is saved. **Default:** data not written to file
`APPEND`	logical	Specifies that saved profiles are to be appended to the file specified by **`OUT.FILE`**. If **`APPEND`** is false, any existing data in the specified file are discarded before writing the new data. **Default:** false
`CM`	logical	Specifies that the distances are to be printed in centimeters. **Default:** false
`UM`	logical	Specifies that the distances are to be printed in microns. **Default:** true
`NM`	logical	Specifies that the distances are to be printed in nanometers. **Default:** false

Parameter	Type	Definition
SPOT	number	Print the coordinate along the cross-section at which the selected quantity equals the specified value. **Units:** units of the selected quantity **Default:** none
LAYERS	logical	Report the integral of the selected quantity over each layer of the device structure. **Default:** false
X.MIN	number	The minimum position along the cross-section to be printed. **Units:** microns **Default:** none
X.MAX	number	The maximum position along the cross-section to be printed. **Units:** microns **Default:** none

Description

The **PRINT.1D** statement prints the values of the selected quantity along a cross-section through the device. Cross-sections are defined as vertical or horizontal (by the **X.VALUE** and **Y.VALUE** parameters, respectively), along the interface between two materials, or along a boundary of the device structure. The quantity to be printed must be specified on a **SELECT** statement preceding the **PRINT.1D** statement. The **SPOT** parameter finds all points along the specified path at which the selected quantity equals the specified value.

Layers

If **LAYERS** is specified, the integral of the selected quantity over each layer is printed. The integration is along the path defined by the **X.VALUE**, **Y.VALUE**, or interface specification. Layers are delimited by those points along the path where the material type changes or the sign of the selected quantity changes. If a cross-section passes out a structure into the ambient and then re-enters the structure, the ambient “layer” may be omitted from the **LAYERS** output.

Interface Values

The values along the interface between two materials depend on the order in which the materials are specified. Thus **SILICON /OXIDE** and **OXIDE /SILICON** print values at the same interface, but the first prints the values in the silicon, while the second prints the values in the oxide.

Values along an interface are sorted by their *x* coordinates. The values printed may not be in order if there are vertical or re-entrant interfaces, or if the structure contains more than one interface between the specified materials.

Saving Profiles in a File

If **OUT.FILE** is specified, profiles printed by **PRINT.1D** are saved in a file. The file contains the same information that is sent to the standard output except that a ‘/’ is substituted for the first blank space in the heading line. Thus the file consists of a heading line (with a ‘/’ in the first column) followed by a series of data lines, each containing three values: the *x* or *y* location, the solution value (specified by the **Z** parameter on a preceding **SELECT** statement), and the material type. Results produced by the **SPOT** and **LAYERS** parameters are not saved in the file. If **APPEND** is specified, the profile information is appended following any previously existing data in the output file; if **APPEND** is false (the default), existing data in the output file is discarded before the new information is saved. Files produced by the **OUT.FILE** parameter on the **PRINT.1D** statement can be read with the **PROFILE** statement or used as targets for optimizing profiles.

Examples

1. The following statements print the boron concentration at *x*=1.0 micron between the top of the mesh and *y*=3.0 microns:

   ```
   SELECT   Z=Boron
   PRINT.1D X.VAL=1.0 X.MAX=3.0 OUT.FILE=boron.dat
   ```

   The profile is saved in the file *boron.dat*.

2. The following statements print the *x* and *y* coordinates of the interface between silicon and oxide:

   ```
   SELECT   Z=y
   PRINT.1D SILICON /OXIDE
   ```

3. The following statements prints junction depths:

   ```
   SELECT Z=doping
   PRINT LAYERS X.V=0
   ```

   The **PRINT** statement assumes that a new layer begins whenever the selected value (net doping in this case) changes sign.

4. The following statements print the thicknesses of material layers:

   ```
   SELECT Z=1.0
   PRINT LAYERS X.V=0
   ```

   The **SELECT** statement specifies a constant value of one, so the **PRINT** statement only uses material boundaries to define layers. Further, when the value "1.0" is integrated over each layer, the result is just the layer thickness.

PLOT.1D

The **`PLOT.1D`** statement plots the value of the selected expression along a line through the structure.

```
PLOT.1D

    { {  [ { X.VALUE=<n> | Y.VALUE=<n> } ]
      |  ( {  MATERIAL=<c> | SILICON | OXIDE | OXYNITRI | NITRIDE
             | POLYSILI | PHOTORES | ALUMINUM
            }
            {  /MATERIA=<c> | /SILICON | /OXIDE | /OXYNITR | /NITRIDE
             | /POLYSIL | /PHOTORE | /ALUMINU | /AMBIENT | /REFLECT
            }
         )
      }
    | {  IN.FILE=<c>
                 { (TIF X.AXIS=<c> Y.AXIS=<c>)
                  |( { (COLUMN [X.COLUMN=<n>] [Y.COLUMN=<n>])
                       | (ROW    [X.ROW=<n>] [Y.ROW=<n>]) }
                     [X.LABEL=<c>]  [Y.LABEL=<c>] )
                 }
                      [X.SHIFT=<n>]  [Y.SHIFT=<n>]
                      [X.SCALE=<n>]  [Y.SCALE=<n>]
                      [Y.LOG] [X.LOG]
      }
    |  ELECTRIC
    }
    [BOUNDARY] [CLEAR] [AXES]
    [SYMBOL=<n>] [CURVE] [LINE.TYP=<n>] [COLOR=<n>]
    [LEFT=<n>] [RIGHT=<n>] [BOTTOM=<n>] [TOP=<n>]
    [X.OFFSET=<n>] [X.LENGTH=<n>] [X.SIZE=<n>]
    [Y.OFFSET=<n>] [Y.LENGTH=<n>] [Y.SIZE=<n>]
    [T.SIZE=<n>]
```

Parameter	Type	Definition
`X.VALUE`	number	A vertical cross-section is to be plotted at this value of *x*. **Units:** microns **Default:** 0.0
`Y.VALUE`	number	A horizontal cross-section is to be plotted at this value of *y*. **Units:** microns **Default:** none
`MATERIAL`	character	Plot a cross-section through the named material, at the interface with the other specified material. **Default:** none

Parameter	Type	Definition
SILICON	logical	Plot a cross-section through silicon, at the interface with the other specified material. **Default:** false
OXIDE	logical	Plot a cross-section through oxide, at the interface with the other specified material. **Default:** false
OXYNITRI	logical	Plot a cross-section through oxynitride, at the interface with the other specified material. **Default:** false
NITRIDE	logical	Plot a cross-section through nitride, at the interface with the other specified material. **Default:** false
POLYSILI	logical	Plot a cross-section through polysilicon, at the interface with the other specified material. **Default:** false
PHOTORES	logical	Plot a cross-section through photoresist, at the interface with the other specified material. **Default:** false
ALUMINUM	logical	Plot a cross-section through aluminum, at the interface with the other specified material. **Default:** false
/MATERIA	character	Plot a cross-section through the other specified material, at the interface with the named material. **Default:** none
/SILICON	logical	Plot a cross-section through the other specified material, at the interface with silicon. **Default:** false
/OXIDE	logical	Plot a cross-section through the other specified material, at the interface with oxide. **Default:** false
/OXYNITR	logical	Plot a cross-section through the other specified material, at the interface with oxynitride. **Default:** false
/NITRIDE	logical	Plot a cross-section through the other specified material, at the interface with nitride. **Default:** false
/POLYSIL	logical	Plot a cross-section through the other specified material, at the interface with polysilicon. **Default:** false

Parameter	Type	Definition
`/PHOTORE`	logical	Plot a cross-section through the other specified material, at the interface with photoresist. **Default:** false
`/ALUMINU`	logical	Plot a cross-section through the other specified material, at the interface with aluminum. **Default:** false
`/AMBIENT`	logical	Plot a cross-section through the other specified material, at the interface with the exposed surface. **Default:** false **Synonyms:** **`/GAS`**, **`/EXPOSED`**
`/REFLECT`	logical	Plot a cross-section through the other specified material, at the interface with the reflecting boundary (if any). **Default:** false
`IN.FILE`	character	The identifier for the file containing the data to plot. This file may contain experimental data or data produced by the **`EXTRACT`** or **`ELECTRICAL`** statements. **Default:** none
`TIF`	logical	Specifies that the format of **`IN.FILE`** is TIF (*.ivl* file from Medici). **Default:** false
`X.AXIS`	character	The quantity used for the horizontal axis when plotting data stored in a TIF file. The label is automatically assigned with the string composite of the **`X.AXIS`** and the unit associated with **`X.AXIS`** in a TIF file. **Default:** none
`Y.AXIS`	character	The quantity used for the vertical axis when plotting data stored in a TIF file. The label is automatically assigned with the string composite of the **`Y.AXIS`** and the unit associated with **`Y.AXIS`** in a TIF file. **Default:** none
`COLUMN`	logical	Specifies that the format of **`IN.FILE`** is column-wise. **Default:** true
`X.COLUMN`	number	The index of the column in the file specified by the **`IN.FILE`** parameter that contains the horizontal coordinates of the plot. **Units:** none **Default:** 1
`Y.COLUMN`	number	The index of the column in the file specified by the **`IN.FILE`** parameter that contains the vertical coordinates of the plot. **Units:** none **Default:** 2
`ROW`	logical	Specifies that the format of **`IN.FILE`** is row-wise. **Default:** false

Parameter	Type	Definition
X.ROW	number	The index of the row (line) in the file specified by the **IN.FILE** parameter that contains the horizontal coordinates of the plot. **Units:** none **Default:** 1
Y.ROW	number	The index of the row (line) in the file specified by the **IN.FILE** parameter that contains the vertical coordinates of the plot. **Units:** none **Default:** 2
X.LABEL	character	The label of the horizontal axis for **X.COLUMN** data in the **IN.FILE** file. **Default:** “Distance (microns)”
Y.LABEL	character	The label of the vertical axis for **Y.COLUMN** data in the **IN.FILE** file. **Default:** if the **SELECT** statement is defined, **LABEL** in **SELECT**, otherwise, “Concentration (#/cm^3)”
X.SHIFT	number	The offset by which **X.COLUMN** data are shifted when reading **IN.FILE** file. **Units:** the same as for the **X.COLUMN** data **Default:** 0.0
Y.SHIFT	number	The offset by which **Y.COLUMN** data are shifted when reading **IN.FILE** file. **Units:** the same as for the **Y.COLUMN** data **Default:** 0.0
X.SCALE	number	The scaling factor by which **X.COLUMN** data are multiplied when reading **IN.FILE** file. **Units:** none **Default:** 1.0
Y.SCALE	number	The scaling factor by which **Y.COLUMN** data are multiplied when reading **IN.FILE** file. **Units:** none **Default:** 1.0
Y.LOG	logical	Specifies that the vertical axis for **Y.COLUMN** data in the **IN.FILE** file is logarithmic. **Default:** the current value dependent on **Z** quantity in the **SELECT** statement **Synonym**: **LOG**
X.LOG	logical	Specifies that the horizontal axis for **X.COLUMN** data in the **IN.FILE** file is logarithmic. **Default:** the current value in the previous specification. Otherwise, false.
ELECTRIC	logical	Specifies plotting of results from a preceding **ELECTRICAL** statement. **Default:** false
BOUNDARY	logical	If true, material boundaries that are crossed are indicated by dashed vertical lines on the plot. **Default:** true

Parameter	Type	Definition
CLEAR	logical	If true, the graphics screen is cleared before the graph is drawn. **Default:** true
AXES	logical	Specifies that axes should be drawn, using scaling information from this statement and/or the current structure. If **AXES** is false, no axes are drawn and scaling information from the previous plotting statement is used (i.e., **LEFT**, **RIGHT**, **BOTTOM**, and **TOP** are ignored). If **AXES** is false and no previous plotting statement has been given, an error is reported. **Default:** true **Synonym: AXIS**
SYMBOL	number	The type of centered symbol to be drawn at each point where the cross-section intersects a mesh line. This value must be in the range 1 to 15. Values of this parameter are associated with the following symbols: Square Circle Triangle Plus Upper case X Diamond Up-arrow Roofed upper case X Upper case Z Upper case Y Curved square Asterisk Hourglass Bar Star **Units:** none **Default:** no symbols drawn
CURVE	logical	Specifies that a line is to be drawn through the data points. **Default:** true
LINE.TYP	number	The dashed line type used for the plotted data. (The axes are always drawn with line type 1.) **Units:** none **Default:** 1
COLOR	number	The color of line used for the plotted data. (The axes are always drawn with color 1.) **Units:** none **Default:** 1

Parameter	Type	Definition
LEFT	number	The minimum value to be plotted on the *x* axis. **Units:** microns **Default:** minimum *x* or *y* coordinate of the structure **Synonym: X.MIN**
RIGHT	number	The maximum value to be plotted on the *x* axis. **Units:** microns **Default:** maximum *x* or *y* coordinate of the structure **Synonym: X.MAX**
BOTTOM	number	The minimum value of the selected expression to be plotted, in units of the plot variable. **Units:** units of the selected expression **Default:** minimum value of the selected expression **Synonym: Y.MIN**
TOP	number	The maximum value of the selected expression to be plotted, in units of the plot variable. **Units:** units of the selected expression **Default:** maximum value of the selected expression **Synonym: Y.MAX**
X.OFFSET	number	The distance by which the left end of the horizontal axis is offset from the left edge of the graphics viewport. **Units:** cm **Default:** 2.0
X.LENGTH	number	The length of the horizontal axis. **Units:** cm **Default:** viewport width - **X.OFFSET** - 1.25
X.SIZE	number	The height of the characters used to label the horizontal axis. **Units:** cm **Default:** 0.25
Y.OFFSET	number	The distance by which the bottom end of the vertical axis is offset from the bottom edge of the graphics viewport. **Units:** cm **Default:** 2.0
Y.LENGTH	number	The length of the vertical axis. **Units:** cm **Default:** viewport height - **Y.OFFSET** - 1.25
Y.SIZE	number	The height of the characters used to label the vertical axis. **Units:** cm **Default:** 0.25
T.SIZE	number	The height of the characters in the character string used as the plot title. **Units:** cm **Default:** 0.4

Description

The **PLOT.1D** statement plots cross-sections vertically or horizontally through the device, or along an interface between two materials, or along a boundary of the device. The statement has options to provide for initialization of the graphics device and plotting of axes. The statement can optionally draw vertical lines whenever a material boundary is crossed. The vertical axis corresponds to the variable selected with the **SELECT** statement.

Limits can be specified so that only a portion of the entire device is shown, or more than one variable can be conveniently plotted. By default the limits of the *x* axis extend to the edges of the structure, and the *y* axis is scaled according to the minimum and maximum values of the selected value over the entire structure.

The quantity to be plotted must be defined by a preceding **SELECT** statement. The type of graphics device must be set, either with an **OPTION** statement or through use of a suitable default. (See OPTION, p. 4-41 and Appendix B.)

If two materials are specified (e.g., **OXIDE /SILICON**), a cross-section is plotted in the first material (e.g., oxide) at the interface with the second material (e.g., silicon). *Note* that **OXIDE /SILICON** produces different results from **SILICON /OXIDE**. For interface plots, the points along the interface are sorted by their *x* coordinates; specifying interfaces containing vertical segments or reentrant angles may not produce useful plots.

Line Type and Color

The **LINE.TYP** parameter specifies the dashed line type for plotting the data. Line type 1 is solid, while types 2 through 7 are dashed lines with increasing dash sizes. Types 8 through 10 produce more complicated patterns of dashes.

The **COLOR** parameter specifies the color for plotting the data. Color 1 is the default, and produces a line that contrasts with the background color (e.g., black on white or white on black). The **COLOR** parameter has no effect on monochrome displays.

The colors produced by the **COLOR** parameter depend on the type of display being used. Where possible, the colors 2 through 7 have been set up to produce the colors red, green, blue, cyan (light blue), magenta (light purple), and yellow, in that order. Colors 8 and above produce a repeating series of 12 colors in rainbow order, from red to violet.

IN.FILE Parameter

The format of the file specified by the **IN.FILE** parameter can be either columnwise or TIF. In the case of a columnwise format, the file may contain the following two types of lines:

1. Lines that are blank or contain a slash (/) as the first nonblank character are ignored and can be used to document the file.
2. Other lines define the data at one point in the distribution. These lines must contain the following values:
    a. Value number **X.COLUMN** is the horizontal coordinate of the point.
    b. Value number **Y.COLUMN** is the vertical coordinate of the point.

If this line contains fewer than *N* numerical values in free-field format, where *N* is the maximum of indices (**`X.COLUMN`** and **`Y.COLUMN`**) for the values listed above, the line is ignored.

Since the *.ivl* log file of Medici is a TIF format file, the results of device simulation can be plotted if **`TIF`**, **`X.AXIS`** and **`Y.AXIS`** are specified. This capability makes it possible to easily compare the electrical calculations of TSUPREM-4 and Medici.

The transformation of data read by the **`IN.FILE`** parameter is as follows:

Data of X axis = **`X.SCALE`** x *Data of* **`X.COLUMN`** *or* **`X.AXIS`** + **`X.SHIFT`**
Data of Y axis = **`Y.SCALE`** x *Data of* **`Y.COLUMN`** *or* **`Y.AXIS`** + **`Y.SHIFT`**

Examples

1. The following statement clears the screen, draws a set of axes, and plots a (vertical) cross-section at *x*=1.0 micron:

```
PLOT.1D X.V=1.0 SYMB=1 ^CURVE
```

   Symbol 1 (a small square) is drawn at each data point; the line through the data points is suppressed.

2. The following statement plots a cross-section at *x*=2.0 microns on the previous set of axes, without clearing the screen:

```
PLOT.1D X.V=2.0 ^AXES ^CLEAR LINE.TYP=2 COLOR=3
```

   A line consisting of short dashes is used, and appears in color 3 on color displays.

PLOT.2D

The **`PLOT.2D`** statement specifies a 2D plot of the device structure.

```
PLOT.2D

   [X.MIN=<n>] [X.MAX=<n>] [Y.MIN=<n>] [Y.MAX=<n>] [SCALE]
   [CLEAR] [AXES] [BOUNDARY] [L.BOUND=<n>] [C.BOUND=<n>]
   [GRID] [L.GRID=<n>] [C.GRID=<n>]
   [ [STRESS] [FLOW] VLENG=<n> [VMAX=<n>]
     [L.COMPRE=<n>] [C.COMPRE=<n>] [L.TENSIO=<n>] [C.TENSIO=<n>]
   ]
   [DIAMONDS]
   [X.OFFSET=<n>] [X.LENGTH=<n>] [X.SIZE=<n>]
   [Y.OFFSET=<n>] [Y.LENGTH=<n>] [Y.SIZE=<n>]
   [T.SIZE=<n>]
```

Parameter	Type	Definition
`X.MIN`	number	The *x* coordinate of the left edge of the region to be plotted. **Units:** microns **Default:** left edge of the device **Synonym:** **`LEFT`**
`X.MAX`	number	The *x* coordinate of the right edge of the region to be plotted. **Units:** microns **Default:** right edge of the device **Synonym:** **`RIGHT`**
`Y.MIN`	number	The *y* coordinate of the top edge of the region to be plotted. **Units:** microns **Default:** Ytop-(**`Y.MAX`**-Ytop)/10, where Ytop is the *y* coordinate of the top of the device **Synonym:** **`TOP`**
`Y.MAX`	number	The *y* coordinate of the bottom edge of the region to be plotted. **Units:** microns **Default:** bottom edge of the device **Synonym:** **`BOTTOM`**
`SCALE`	logical	If true, the *x* and *y* axes are scaled to preserve the true aspect ratio of the device. If false, the axes are scaled to fill the available plotting area. **Default:** false
`CLEAR`	logical	If true, the graphics screen is cleared before the graph is drawn. **Default:** true

Parameter	Type	Definition
AXES	logical	Specifies that axes should be drawn, using scaling information from this statement and/or the current structure. If **AXES** is false, no axes are drawn and scaling information from the previous plotting statement is used (i.e., **X.MIN**, **X.MAX**, **Y.MIN**, and **Y.MAX** are ignored). If **AXES** is false and no previous plotting statement has been given, an error is reported. **Default:** true **Synonym: AXIS**
BOUNDARY	logical	Plot the device boundary and material interfaces. **Default:** true
L.BOUND	number	The dashed line type used for plotting the device boundary and material interfaces. **Units:** none **Default:** 1 **Synonyms: LINE.BOU**, **LINE.TYP**
C.BOUND	number	The line color used for plotting the device boundary and material interfaces. **Units:** none **Default:** 1
GRID	logical	Plot the grid used for the numerical solution. **Default:** false
L.GRID	number	The dashed line type used for plotting the grid. **Units:** none **Default:** 1 **Synonym: LINE.GRI**
C.GRID	number	The line color used for plotting the grid. **Units:** none **Default:** 1
STRESS	logical	Plot the principal stresses in the structure. Vectors are drawn along the two principal axes of the stress tensor at each mesh point. **Default:** false
FLOW	logical	Plot vectors indicating the velocity (due to oxidation) or displacement (calculated by the **STRESS** statement) at each mesh point. **Default:** false
VLENG	number	The length of the vector drawn for the maximum value of stress or velocity. **Units:** microns **Default:** none
VMAX	number	The maximum value of stress or velocity to be plotted. **Units:** dynes/cm^2 for stress, cm/sec for flow **Default:** no limit on maximum value

Parameter	Type	Definition
`L.COMPRE`	number	The dashed line type used for plotting compressive stress vectors and flow vectors. **Units:** none **Default:** 1 **Synonyms:** **`L.FLOW`**, **`L.VELOCI`**, **`LINE.COM`**
`C.COMPRE`	number	The line color used for plotting compressive stress vectors and flow vectors. **Units:** none **Default:** 1 **Synonyms:** **`C.FLOW`**, **`C.VELOCI`**
`L.TENSIO`	number	The dashed line type used for plotting tensile stress vectors. **Units:** none **Default:** 1 **Synonym:** **`LINE.TEN`**
`C.TENSIO`	number	The line color used for plotting tensile stress vectors. **Units:** none **Default:** 1
`DIAMONDS`	logical	Plot a small symbol at each mesh point location. **Default:** false
`X.OFFSET`	number	The distance by which the left end of the horizontal axis is offset from the left edge of the graphics viewport. **Units:** cm **Default:** 2.0
`X.LENGTH`	number	The length of the horizontal axis. **Units:** cm **Default:** viewport width - **`X.OFFSET`** - 1.25
`X.SIZE`	number	The height of the characters used to label the horizontal axis. **Units:** cm **Default:** 0.25
`Y.OFFSET`	number	The distance by which the bottom end of the vertical axis is offset from the bottom edge of the graphics viewport. **Units:** cm **Default:** 2.0
`Y.LENGTH`	number	The length of the vertical axis. **Units:** cm **Default:** viewport height - **`Y.OFFSET`** - 1.25

Parameter	Type	Definition
`Y.SIZE`	number	The height of the characters used to label the vertical axis. **Units:** cm **Default:** 0.25
`T.SIZE`	number	The height of the characters in the character string used as the plot title. **Units:** cm **Default:** 0.4

Description

The **`PLOT.2D`** statement specifies a 2D plot. It is useful in itself for plotting structure outlines and interface locations, grid, stress, and velocities. It is also used to plot axes for the **`CONTOUR`** and **`COLOR`** statements.

The type of graphics device must be set, either with an **`OPTION`** statement or use of a suitable default. See OPTION, p. 4-41 and in Appendix B.

The x and y limits are in microns, and refer to the coordinates of the structure being simulated. The x coordinate increases from left to right on the plot, while y increases from top to bottom. If **`Y.MIN`** is greater than **`Y.MAX`**, the plot is flipped. Thus, the value at the top of the plot (**`Y.MIN`**) is greater than the value at the bottom of the plot (**`Y.MAX`**). Similarly, **`X.MIN`** can be greater than **`X.MAX`**.

For **`STRESS`** and **`FLOW`** plots, the maximum vector length must be given by **`VLENG`**. **`VLENG`** is in the same units as the device structure, namely microns. If **`VMAX`** is specified, values of stress or velocity larger than the specified value are not plotted.

Line Type and Color

The **`L.BOUND`**, **`L.GRID`**, **`L.COMPRE`**, and **`L.TENSIO`** parameters specify dashed line types to be used for plotting various quantities. Type 1 produces a solid line while types 2 through 10 produce various dashed line styles.

The **`C.BOUND`**, **`C.GRID`**, **`C.COMPRE`**, and **`C.TENSIO`** parameters specify the colors to be used in plotting the various quantities. Color 1 (the default) gives a line color that contrasts with the background (e.g., black on white or green on black). Colors 2 through 7 give red, green, blue, cyan (light blue), magenta (light purple), and yellow on most displays, while colors 8 and above produce a repeating sequence of 12 colors in rainbow order from red to violet. *Note* that not all display types give these colors; the color parameters have no effect on monochrome displays.

Examples

1. The following statement plots the axes, triangular grid, and structure boundaries (in that order):

```
PLOT.2D  SCALE  GRID  C.GRID=2
```

The grid is plotted with color 2 (usually red on color devices). The plot is scaled the same in the *x* and *y* directions in order to display the correct aspect ratio of the structure.

2. The following statement draws the material interfaces and axis between 2.0 and 5.0 microns, without clearing the screen first:

   ```
   PLOT.2D  X.MIN=2 X.MAX=5 SCALE ^CLE
   ```

   The plot is scaled to show the true aspect ratio of the structure.

3. The following statement draws the structure boundaries and material interfaces using line type 2, and shows the grid points as diamonds:

   ```
   PLOT.2D  LINE.TYP=2 DIAMONDS ^AXES ^CLEAR
   ```

   The plotting area is not cleared, and axes from the previous plot are used. The aspect of the structure can be distorted because **`SCALE`** is not specified.

CONTOUR

The **`CONTOUR`** statement plots a contour of the selected variable on a 2D plot.

```
CONTOUR

   VALUE=<n> [LINE.TYP=<n>] [COLOR=<n>] [SYMBOL=<n>]
```

Parameter	Type	Definition
`VALUE`	number	The value of the selected variable at which to plot a contour. **Units:** units of the selected variable **Default:** none
`LINE.TYP`	number	The dashed line type to be used for the contour. **Units:** none **Default:** 1
`COLOR`	number	The line color to be used for the contour. **Units:** none **Default:** 1
`SYMBOL`	number	The type of centered symbol to be drawn at each mesh line intersection on the contour. This value must be in the range 1 to 15. Values of this parameter are associated with the following symbols: Square Circle Triangle Plus Upper case X Diamond Up-arrow Roofed upper case X Upper case Z Upper case Y Curved square Asterisk Hourglass Bar Star **Units:** none **Default:** no symbols drawn

Description

The **CONTOUR** statement draws a contour of the selected variable at the value specified. The value must be specified in units of the selected variable.

This statement assumes that a **PLOT.2D** statement has been specified previously and the screen has been set up for plotting a 2D picture. The variable to be plotted must have been specified on a preceding **SELECT** statement.

Note:

> ***The results of this statement are undefined unless valid* PLOT.2D *and* SELECT *statements have been executed beforehand.***

Line Type and Color

Line type 1 gives a solid line, while types 2 through 10 give dashed lines of various sorts. Color 1 produces contours that contrast with the background (e.g., black on white or green on black). On most color devices, colors 2 through 7 give red, green, blue, cyan (light blue), magenta (light purple), and yellow, while colors 8 and above give a repeating series of 12 colors in rainbow order (from red to violet). On monochrome devices, the **COLOR** parameter has no effect.

Example

The following statements plot a series of contours with line type 2, where the boron concentration is equal to 10^{15}, 10^{16}, 10^{17}, 10^{18}, and 10^{19}:

```
SELECT  Z=log10(Boron)
FOREACH X ( 15 TO 19 STEP 1 )
  CONTOUR VALUE=X LINE.TYP=2
END
```

Note the use of the **CONTOUR** statement inside a **FOREACH** loop, to plot a series of contours.

Additional CONTOUR Notes

- Values of contours of *log10(concentration)* should be specified as exponents, e.g., 16 and not 1e16.
- Symbols are placed where the contour crosses a boundary between mesh elements. Thus, the density of symbols reflects the density of the mesh along a contour.

COLOR

The **`COLOR`** statement fills an area of a 2D plot.

```
COLOR

    [COLOR=<n>] [MIN.VALU=<n>] [MAX.VALU=<n>]
    [ {  MATERIAL=<c> | SILICON | OXIDE | OXYNITRI | NITRIDE | POLYSILI
       | ALUMINUM | PHOTORES
      }
    ]
```

Parameter	Type	Definition
`COLOR`	number	The “color” to be used for the area fill. This can correspond to an actual color, a shade of gray, or to a fill pattern, depending on the plot device. **Units:** none **Default:** 1
`MIN.VALU`	number	The minimum value of the selected variable to be filled. **Units:** units of the selected variable **Default:** the minimum value of the selected variable
`MAX.VALU`	number	The maximum value of the selected variable to be filled. **Units:** units of the selected variable **Default:** the maximum value of the selected variable
`MATERIAL`	character	Only regions of the named material are filled. **Default:** none
`SILICON`	logical	Only regions of the device composed of silicon are filled. **Default:** false
`OXIDE`	logical	Only regions of the device composed of oxide are filled. **Default:** false
`OXYNITRI`	logical	Only regions of the device composed of oxynitride are filled. **Default:** false
`NITRIDE`	logical	Only regions of the device composed of nitride are filled. **Default:** false
`POLYSILI`	logical	Only regions of the device composed of polysilicon are filled. **Default:** false
`ALUMINUM`	logical	Only regions of the device composed of aluminum are filled. **Default:** false
`PHOTORES`	logical	Only regions of the device composed of photoresist are filled. **Default:** false

Description

The **COLOR** statement performs area fills on isoconcentration bands in the selected variable. If either the **MIN.VALU** or **MAX.VALU** keyword is specified, regions of the device having values of the plot variable between the two values are filled. If a material type is specified, only regions of the structure composed of the specified material are filled. If neither **MIN.VALU** or **MAX.VALU** is specified, all regions of the specified material are filled. If no values or materials are specified, a warning is printed and the statement is ignored.

The quantity referred to by **MIN.VALU** and **MAX.VALU** must be specified on a **SELECT** statement preceding the **COLOR** statement. The axes for the plot must be set up by a preceding **PLOT.2D** statement.

Plot Device Selection

The effect of the **COLOR** statement depends on the type of plot device selected. On color devices, the specified areas are colored. In most cases, colors 2 through 7 are red, green, blue, cyan (light blue), magenta (light purple), and yellow, while colors 8 and above produce a repeating series of 12 colors in rainbow order (red to violet). On some monochrome devices, different values of **COLOR** produce varying gray-scale or halftone values. On devices without hardware area fill capabilities (defined in the *s4pcap* file), a cross-hatch pattern is used.

On some devices an area fill may overwrite previously plotted information, such as grid or material boundaries. In this case the grid or boundaries can be re-plotted after doing the area fill, to ensure that they are visible.

Examples

The following statement fills all oxide regions with color 4:

```
COLOR OXIDE COLOR=4
```

4. The following statements fill those portions of the structure having a boron concentration between 10^{15} and 10^{16} with color:

```
SELECT Z=log10(Boron)
COLOR MIN.V=15 MAX.V=16 COLOR=3
```

PLOT.3D

Create a 3D projection plot of the solution.

```
PLOT.3D

  [THETA=<n>] [PHI=<n>]
  [CLEAR]
  [X.MIN=<n>] [X.MAX=<n>] [Y.MIN=<n>] [Y.MAX=<n>]
  [Z.MIN=<n>] [Z.MAX=<n>] [LINE.TYP=<n>] [COLOR=<n>]
  [NUM.CNTR=<n>]
  [BOUNDARY] [L.BOUND=<n>] [C.BOUND=<n>]
```

Parameter	Type	Definition
THETA	number	The angle above the *x-y* plane from which the device is viewed. If an angle of 0 is specified, the viewpoint is in the *x-y* plane. Values between –90 and 90 are permitted. **Units:** degrees **Default:** 45 **Synonym:** **ELEVATIO**
PHI	number	The angle of counter-clockwise rotation of the device in the *x-y* plane. Only values within 30 degrees of 45, 135, 225, and 315 degrees should be used; values outside this range are not supported. Never use values of 0, 90, 180, or 270 degrees. **Units:** degrees **Default:** 45 **Synonym:** **AZIMUTH**
CLEAR	logical	Specifies that the graphics display area is to be cleared before beginning the plot. **Default:** true
X.MIN	number	The minimum value to be plotted along the *x* axis. **Units:** microns **Default:** the minimum *x* coordinate of the current structure
X.MAX	number	The maximum value to be plotted along the *x* axis. **Units:** microns **Default:** the maximum *x* coordinate of the current structure
Y.MIN	number	The minimum value to be plotted along the *y* axis. **Units:** microns **Default:** the minimum *y* coordinate of the current structure
Y.MAX	number	The maximum value to be plotted along the *y* axis. **Units:** microns **Default:** the maximum *y* coordinate of the current structure

Parameter	Type	Definition
`Z.MIN`	number	The minimum value on the *z* axis. **Units:** units of the plot variable **Default:** the minimum value of the plot variable
`Z.MAX`	number	The maximum value on the *z* axis. **Units:** units of the plot variable **Default:** the maximum value of the plot variable
`LINE.TYP`	number	The dashed line type to be used for plotting the data. **Units:** none **Default:** 1
`COLOR`	number	The line color to be used for plotting the data. **Units:** none **Default:** 1
`NUM.CNTR`	number	The number of grid lines drawn to represent the surface. One axis has this number of lines, the other is drawn with a number calculated from it dependent on the value of **`PHI`**. **Units:** none **Default:** 20
`BOUNDARY`	logical	Draw the device boundaries on the *z*=**`Z.MIN`** plane of the plot. **Default:** true
`L.BOUND`	number	The dashed line type used for plotting the boundary. **Units:** none **Default:** 1 **Synonym: `LINE.BOU`**
`C.BOUND`	number	The line color used for plotting the boundary. **Units:** none **Default:** 1

Description

The **`PLOT.3D`** statement plots a “bird’s eye view” of a three-dimensional wire diagram of the selected data. The routine interpolates a series of cross-section lines and plots them with the given viewpoint parameters. Axes can be drawn and labeled.

The variable to be plotted must be specified with a **`SELECT`** statement prior to the **`PLOT.3D`** statement. The graphics device to be used must be set with an **`OPTION`** statement, unless an appropriate default device is available.

Note:

The algorithms used work only for certain values of* `PHI`*; values near 0, 90, 180, or 270 deg should be avoided.

Line Type and Color

The **LINE.TYP** and **L.BOUND** parameters specify the style of dashed line to be used for plotting the data and boundaries, respectively. Type 1 produces a solid line, while types 2 through 10 produce various dashed styles.

The **COLOR** and **C.BOUND** parameters specify the color of lines to be used for data and boundaries. Color 1 produces lines that contrast with the background (e.g., black on white or green on black). On most color displays, colors 2 through 7 give red, green, blue, cyan (light blue), magenta (light purple), and yellow, while colors 8 and above give a repeating sequence of 12 colors in rainbow order (red through violet).

Examples

1. The following statement plots the device as viewed from straight above:

   ```
   PLOT.3D  NUM.C=30 THETA=90
   ```

   Thirty contour lines are selected.

2. The following statement plots the bird's eye view plot from 60 deg above the horizon and 30 deg off the *x* axis:

   ```
   PLOT.3D  PHI=30 THETA=60 COLOR=4
   ```

   Color 4 (usually blue) is used.

Additional PLOT.3D Notes

- Nonrectangular areas are padded with **Z.MIN** values to make the final surface rectangular.
- **Z.MIN** and **Z.MAX** specify the limits of the *z* axis, but values being plotted are not clipped at these values.

LABEL

The **`LABEL`** statement is used to add a label to a plot.

```
LABEL

   { ( X=<n> Y=<n> [CM] ) | ( [X.CLICK=<c>] [Y.CLICK=<c>] ) }
   [SIZE=<n>] [COLOR=<n>]
   [ LABEL=<c> [ {LEFT | CENTER | RIGHT} ] ]
   [LINE.TYP=<n>] [C.LINE=<n>] [LENGTH=<n>]
   [ {  ( [SYMBOL=<n>] [C.SYMBOL=<n>] )
      | ( [RECTANGL] [C.RECTAN=<n>] [W.RECTAN=<n>]
[H.RECTAN=<n>] )
     }
   ]
```

Parameter	Type	Definition
`X`	number	The horizontal location corresponding to the left end, center, or right end of the character string (depending on whether **`LEFT`**, **`CENTER`**, or **`RIGHT`** is specified). If the **`CM`** parameter is specified, then this parameter specifies a location in centimeters relative to the left edge of the graphics viewport. Otherwise, this parameter specifies the location in axis units along the horizontal axis. **Units:** cm or horizontal axis units **Default:** none
`Y`	number	The vertical location corresponding to the bottom of the character string. If the **`CM`** parameter is specified, then this parameter specifies a location in centimeters relative to the bottom edge of the graphics viewport. Otherwise, this parameter specifies the location in axis units along the vertical axis. **Units:** cm or vertical axis units **Default:** none
`CM`	logical	Specifies that the **`X`** and **`Y`** parameters are locations in centimeters relative to the lower left edge of the graphics viewport. **Default:** false
`X.CLICK`	character	The variable name to store the x-coordinate of the position at which a mouse is clicked. **Units:** horizontal axis units **Default:** none
`Y.CLICK`	character	The variable name to store the y-coordinate of the position at which a mouse is clicked. **Units:** cm or vertical axis units **Default:** none

Parameter	Type	Definition
SIZE	number	The height of the characters in the character string, and the default size to be used for rectangles and centered symbols. **Units:** cm **Default:** 0.25 **Synonym:** **C.SIZE**
COLOR	number	The color of the label text, and the default color for rectangles, centered symbols, and line segments. **Units:** none **Default:** 1
LABEL	character	The character string to be used to label the plot. **Default:** none
LEFT	logical	Specifies that the character string is to start at the position given by **X** and **Y**. **Default:** true if neither **CENTER** or **RIGHT** is true
CENTER	logical	Specifies that the character string is to be centered horizontally about the position given by **X** and **Y**. **Default:** false
RIGHT	logical	Specifies that the character string is to end at the position given by **X** and **Y**. **Default:** false
LINE.TYP	number	The dashed type of a line segment to be plotted before the label. If **LABEL** is not specified, the line segment is centered at the point given by **X** and **Y**. **Units:** none **Default:** 1
C.LINE	number	The color of the line segment to be plotted before the label. **Units:** none **Default:** **COLOR**
LENGTH	number	The length of the line segment to be plotted before the label. **Units:** cm **Default:** 4***SIZE**

Parameter	Type	Definition
`SYMBOL`	number	The type of centered symbol to be drawn before the label. This value must be in the range 1 to 15. Values of this parameter are associated with the following symbols: Square Circle Triangle Plus Upper case X Diamond Up-arrow Roofed upper case X Upper case Z Upper case Y Curved square Asterisk Hourglass Bar Star If **`LABEL`** is specified, the symbol is placed to the left of the label with one character space between the symbol and the label text. If **`LABEL`** is not specified, the rectangle is centered at the point given by **`X`** and **`Y`**. **Units:** none **Default:** 1
`C.SYMBOL`	number	The color of the symbol (if any). **Units:** none **Default: `COLOR`**
`RECTANGL`	logical	Specifies that a filled rectangle be plotted with the label. If **`LABEL`** is specified, the rectangle is placed to the left of the label with one character space between the rectangle and the label text. If **`LABEL`** is not specified, the rectangle is centered at the point given by **`X`** and **`Y`**. **Default:** false
`C.RECTAN`	number	The color of the filled rectangle. **Units:** none **Default: `COLOR`**
`W.RECTAN`	number	The width of the filled rectangle. **Units:** cm **Default: `SIZE`**
`H.RECTAN`	number	The height of the filled rectangle. **Units:** cm **Default: `SIZE`**

Description

The `LABEL` statement is used to add text, symbols, and/or filled rectangles to a plot. This statement is meaningless unless a `PLOT.1D`, `PLOT.2D`, or `PLOT.3D` statement has been previously specified.

Label Placement

The rules for placing these annotations are as follows:

1. Labels are always placed at the location specified by `X` and `Y`. The label is left justified, centered, or right justified at this location, depending on whether `LEFT`, `CENTER`, or `RIGHT` is specified.
2. The placement of line segments depends on whether a `LABEL` is specified:
   a. If a `LABEL` is specified, the line segment is placed one character width to the left of the label.
   b. If no `LABEL` is specified, the line segment is centered at the location specified by `X` and `Y`.
3. The placement of symbols and filled rectangles depends on whether a line segment or label is specified:
   a. If a line segment is specified, the symbol or filled rectangle is centered on the line segment.
   b. If no line segment is specified but a `LABEL` is specified, the symbol or filled rectangle is placed one character width to the left of the label.
   c. If neither a line segment nor a `LABEL` is specified, the symbol or filled rectangle is centered at the location specified by `X` and `Y`.
4. The coordinates `X` and `Y` should be in the units of the plot axes, e.g., microns or 1/cm^3, unless `CM` is specified.

Note:

Some graphics devices have a cursor whose position can be read by the program. On such devices, a `LABEL` ***statement without*** `X` ***and*** `Y` ***coordinates attempts to read the cursor position and plot the label at that point. On terminals from which the cursor position cannot be read, a*** `LABEL` ***statement without*** `X` ***and*** `Y` ***coordinates may produce unpredictable results.***

Line, Symbol, and Rectangle

If `LINE.TYP`, `C.LINE`, or `LENGTH` is specified, a line segment is drawn. `LINE.TYP` specifies the dashed line type of the line segment. Type 1 produces a solid line, while types 2 through 10 produce various styles of dashed lines. `C.LINE` specifies the color of the line and `LENGTH` gives the length. If either `SYMBOL` or `C.SYMBOL` is specified, a symbol is drawn. If `RECTANGL`, `C.RECTAN`, `W.RECTAN`, or `H.RECTAN` is specified, a filled rectangle is drawn.

The `SIZE` parameter specifies the character size to be used for the label and the default width and height for filled rectangles.

Color

The **COLOR** parameter specifies the color to be used for the label and the default color for any line segment, symbol, or filled rectangle. Color 1 contrasts with the background (e.g., black on white or white on black). On most color devices, colors 2 through 7 produce red, green, blue, cyan (light blue), magenta (light purple), and yellow, while colors 8 and above give a repeating sequence of 12 colors in rainbow order (red through violet). The **COLOR** parameter has no effect on monochrome devices.

Examples

1. The following statements put two labels on the plot starting at *x*=3 microns and *y*=1.4 and 1.6 microns, with a short line of the specified type before each one:

```
LABEL X=3.0 Y=1.4 LABEL="Arsenic" LINE=3
LABEL X=3.0 Y=1.6 LABEL="Phosphorus" LINE=4
```

2. The following statement plots a label preceded by a filled rectangle:

```
LABEL RIGHT X=12.5 Y=9.0 CM LABEL="Oxide" C.RECT=2
```

The label ends at a point 12.5 cm from the *x* axis and 9.0 cm from the *y* axis. It is preceded by a rectangle filled with color 2.

3. The following statement stores the coordinate of the position at which a mouse is clicked:

```
LABEL LABEL=x X.CLICK=px Y.CLICK=py
```

The variables, *px, py* store the coordinate of the position at which a mouse is clicked.

Note:

The unit of the stored value in `X.CLICK` is the same as the x-coordinate unit. However, in the case of `Y.CLICK`, the unit is cm for the distance. In `PLOT.2D` graph, for example, the variable of `Y.CLICK` stores the y-coordinate value in cm unit, while the variable of `X.CLICK` value stores the x-coordinate value in um unit.

EXTRACT

The **EXTRACT** statement extracts information about the structure, prints the results, and/or writes them to a file.

```
EXTRACT

    [ {MATERIAL=<c> | SILICON | OXIDE | OXYNITRI | NITRIDE | POLYSILI |
       PHOTORES | ALUMINUM} ]
    [P1.X=<n>] [P1.Y=<n>] [P2.X=<n>] [P2.Y=<n>]
    [ {/MATERIA=<c> | /SILICON | /OXIDE | /OXYNITR | /NITRIDE |
       /POLYSIL | /PHOTORE | /ALUMINU | /AMBIENT} [CLOCKWIS] ]
    [X=<n>] [Y=<n>] [ {DISTANCE=<n> | MINIMUM | MAXIMUM | VALUE=<n>} ]
    { ( [X.EXTRAC] [Y.EXTRAC] [D.EXTRAC] [VAL.EXTR] )
    | ( [INT.EXTR] [AREA.EXT] [AVG.EXTR] ) }
    [PREFIX=<c>] [SEPARAT=<c>] [SUFFIX=<c>]
    [WRITE] [PRINT]
    [ NAME=<c> [ASSIGN]
      [ { TARGET=<n>
        | ( T.FILE=<c> [V.COLUMN=<n>] [V.LOWER=<n>] [V.UPPER=<n>]
                       [T.COLUMN=<n>] [T.LOWER=<n>] [T.UPPER=<n>]
                       [V.TRANSF=<c>] [T.TRANSF=<c>]
                       [Z.VALUE=<c>]  [SENSITIV])
        }
        [TOLERANC=<n>] [WEIGHT=<n>]
        [MIN.REL=<n>] [MIN.ABS=<n>]
      ]
    ]
    [ OUT.FILE=<c> [APPEND] ] [CLOSE]
```

Parameter	Type	Definition
MATERIAL	character	Extract parameters in the named material. **Default:** none
SILICON	logical	Extract parameters in silicon. **Default:** false
OXIDE	logical	Extract parameters in oxide. **Default:** false
OXYNITRI	logical	Extract parameters in oxynitride. **Default:** false
NITRIDE	logical	Extract parameters in nitride. **Default:** false
POLYSILI	logical	Extract parameters in polysilicon. **Default:** false **Synonym:** **POLY**

Parameter	Type	Definition
`PHOTORES`	logical	Extract parameters in photoresist. **Default:** false
`ALUMINUM`	logical	Extract parameters in aluminum. **Default:** false
`P1.X`	number	The *x* coordinate of the starting point of a line through the structure. **Units:** microns **Default:** the left edge of the structure if an interface is specified, or 0.0, otherwise
`P1.Y`	number	The *y* coordinate of the starting point of a line through the structure. **Units:** microns **Default:** 0.0 if an interface is specified, or the top of the structure otherwise
`P2.X`	number	The *x* coordinate of the end point of a line through the structure. **Units:** microns **Default:** the right edge of the structure, or **`P1.X`** otherwise
`P2.Y`	number	The *y* coordinate of the end point of a line through the structure. **Units:** microns **Default:** 0.0 if an interface is specified, or the bottom of the structure, otherwise
`/MATERIA`	character	Extract parameters along the interface with the named material. **Default:** none
`/SILICON`	logical	Extract parameters along the interface with silicon. **Default:** false
`/OXIDE`	logical	Extract parameters along the interface with oxide. **Default:** false
`/OXYNITR`	logical	Extract parameters along the interface with oxynitride. **Default:** false
`/NITRIDE`	logical	Extract parameters along the interface with nitride. **Default:** false
`/POLYSIL`	logical	Extract parameters along the interface with polysilicon. **Default:** false
`/PHOTORE`	logical	Extract parameters along the interface with photoresist. **Default:** false
`/ALUMINU`	logical	Extract parameters along the interface with aluminum. **Default:** false
`/AMBIENT`	logical	Extract parameters along exposed surfaces. **Default:** false **Synonyms: `/EXPOSED`**, **`/GAS`**

Parameter	Type	Definition
CLOCKWIS	logical	If the specified material interface forms a closed loop, traverse it in the clockwise direction. **Default:** false
X	number	Define a line or a point on a line by the value of its *x* coordinate. **Units:** microns **Default:** none
Y	number	Define a line or a point on a line by the value of its *y* coordinate. **Units:** microns **Default:** 0.0
DISTANCE	number	Define a point on a line by its distance from the start of the line. **Units:** microns **Default:** 1.0
MINIMUM	logical	Extract at the first point on a line where the value of the selected variable is at its minimum. **Default:** false
MAXIMUM	logical	Extract at the first point on a line where the value of the selected variable is at its maximum. **Default:** false
VALUE	number	Extract at the first point on a line where the selected variable has the specified value. **Units:** the units of the selected variable **Default:** none
X.EXTRAC	logical	Extract the *x* coordinate of the specified point. **Default:** false
Y.EXTRAC	logical	Extract the *y* coordinate of the specified point. **Default:** false
D.EXTRAC	logical	Extract the distance of the specified point from the start of the specified line. **Default:** false
VAL.EXTR	logical	Extract the value of the selected variable at the specified point. **Default:** false
INT.EXTR	logical	Extract the integral of the selected variable along the specified line. **Default:** false **Synonym**: **INTEGRAL**
AREA.EXT	logical	Extract the length of the specified line. **Default:** false **Synonym**: **THICKNES**
AVG.EXTR	logical	Extract the average value of the selected variable along the specified line. **Default:** false

Parameter	Type	Definition
PREFIX	character	A character string to be printed before the extracted value(s). *Note* that if a space is desired before the first value, it must be explicitly specified as part of the prefix. **Default:** none
SEPARAT	character	A character string to be printed between extracted values if more than value is extracted. **Default:** **" "** (a single space)
SUFFIX	character	A character string to be printed after the extracted value(s). *Note* that if a space is desired after the last value, it must be explicitly specified as part of the suffix. **Default:** none
WRITE	logical	Specifies that the extracted result is to be written to the extract output file, if one is open. **Default:** true
PRINT	logical	Specifies that the extracted result is to be printed on the standard output (including the output listing file and the user's terminal). **Default:** true
NAME	character	Specifies that the extracted result is to be assigned to the specified name, as if it had been defined with a **DEFINE** statement or an **ASSIGN** statement when either **TARGET** or **T.FILE** is specified. *Note* that if the specified name has been defined with a **DEFINE** statement previously, you need to precede the **EXTRACT** statement with a "%" (percent character) to prevent substitution of its previous value. The **NAME** parameter allows extracted results to be used in subsequent extractions or simulations. **Default:** none
ASSIGN	logical	Specifies that the extracted quantity to be assigned to the variable specified by **NAME** is the same style as the **ASSIGN** statement. **Default:** false
TARGET	number	The desired value of the target being defined for an optimization loop. The optimization attempts to match the extracted value with the value of this parameter. **Units:** determined by the extracted values **Default:** none
T.FILE	character	The file name containing the desired values of the target being defined for an optimization loop. The desired values are in the column specified by the **T.COLUMN** parameter. The optimization attempts to match the extracted values with the desired values. **Default:** none
V.COLUMN	number	The index of the column in the file specified by the **T.FILE** parameter, which contains the variable at which the extraction is performed. **Units:** none **Default:** 1

Parameter	Type	Definition
V.LOWER	number	The lower limit of the variable to be read from **T.FILE**. The target data at variable values less than the value of **V.LOWER** parameter are excluded during an optimization loop. *Note* that the **V.LOWER** specifies the value prior to the transformation by **V.TRANSF**. **Units:** The same as the unit of variable specified by **V.COLUMN** **Default:** -1e+30
V.UPPER	number	The upper limit of the variable to be read from **T.FILE**. The target data at variable values greater than the value of **V.UPPER** parameter are excluded during an optimization loop. *Note* that the **V.UPPER** specifies the value prior to the transformation by **V.TRANSF**. **Units:** the same as the unit of variable specified by **V.COLUMN** **Default:** 1e+30
T.COLUMN	number	The index of the column in the file specified by the **T.FILE** parameter, which contains the desired values of the target being defined for an optimization loop. **Units:** none **Default:** 2
T.LOWER	number	The lower limit of the desired value of a target to be read from **T.FILE**. Target values less than the value of **T.LOWER** parameter are excluded during an optimization loop. *Note* that the **T.LOWER** specifies the value prior to the transformation by **T.TRANSF**. **Units:** the same as the unit of variable specified by **T.COLUMN** **Default:** -1e+30
T.UPPER	number	The upper limit of the desired value of a target to be read from **T.FILE**. Target values greater than the value of **T.UPPER** parameter are excluded during an optimization loop. *Note* that the **T.UPPER** specifies the value prior to the transformation by **T.TRANSF**. **Units:** the same as the unit of variable specified by **T.COLUMN** **Default:** 1e+30
V.TRANSF	character	The function for transformation of the variable data read from **T.FILE**. The specified character string represents the function of a variable **V**, with which the variable data specified by **V.COLUMN** parameter are transformed. **Default:** none
T.TRANSF	character	The function for transformation of the desired data of a target read from **T.FILE**. The specified character string represents the function of a variable **T**, with which the target data specified by **T.COLUMN** parameter are transformed. **Default:** none
Z.VALUE	character	The function which generates the data instead of simulation. The specified character string represents a function of the variable **V**, which corresponds to the variable specified by **V.COLUMN** parameter. **Units:** the same as the unit of target data specified by **T.COLUMN** **Default:** none

Parameter	Type	Definition
`TOLERANC`	number	The RMS (root-mean-square) error for convergence criterion. An optimization loop terminates when the RMS errors of all of specified targets are smaller than **`TOLERANC`**. **Units:** % **Default:** 0.0
`SENSITIV`	logical	Specifies the plot of sensitivity analysis of the target defined by this **`EXTRACT`** statement. This parameter works only if the **`PLOT`** parameter in the **`LOOP`** statement is true. **Default:** true
`WEIGHT`	number	The weighting factor applied to the target being defined for an optimization loop. The weights are used to control the importance of individual targets in calculations of the error during optimization. **Units:** none **Default:** 1.0
`MIN.REL`	number	The minimum target ratio for which relative error is used to calculated the error during optimization. This value is compared with the ratio of the absolute target value to the maximum absolute target value defined by this **`EXTRACT`** statement. **Units:** none **Default:** 1e-10
`MIN.ABS`	number	The minimum target value for which relative error is used to calculate the error during optimization. This value is compared with the absolute target value. **Units:** determined by the extracted values **Default:** 1e-10
`OUT.FILE`	character	The name of an extract output file to be opened. The file is opened before extracting any values, so extracted results appear in the file. **Default:** none
`APPEND`	logical	Specifies that extracted values are to be appended to the file specified by the **`OUT.FILE`** parameter. If **`APPEND`** is false, any existing data in the specified file is discarded when the file is opened. **Default:** false
`CLOSE`	logical	Specifies that the extract output file is to be closed. The file is closed after any values extracted by this **`EXTRACT`** statement are written. **Default:** false

Description

The **`EXTRACT`** statement is used to extract values along a line through a structure or at a point. It has the following properties and uses:

- When a line is specified, layer thicknesses and integrals and averages of solution values can be extracted.
- Lines through the structure can be vertical, horizontal, or oblique; material interfaces can also be treated as lines for purposes of extraction.

- When a point is specified, the solution value, the coordinates of the point, and the distance from the start of a line can be extracted.
- Points can be specified by their coordinates or by the distance from the start of a line. You can also request extraction at the point at which the solution value is a minimum or maximum or takes on a specified value.
- Extraction can be limited to regions of specified materials, or can encompass the entire structure.
- Extracted results can be printed on the standard output or written to a separate output file. Text surrounding and between extracted values can be specified by you.
- Parameters are available for opening and closing extract output files and for appending to existing files.
- The **EXTRACT** statement can be used to define targets for optimization.

Solution Variables

Any solution variable to be used in the extraction must have been previously specified by the **Z** parameter on the **SELECT** statement (see p. 4-146).

Note:

Some extractions (e.g., layer thicknesses) do not require a solution variable. In the description that follows, the quantity specified on the SELECT statement is referred to as the selected value or selected variable.

Extraction Procedure

Extraction proceeds as follows:

1. If **OUT.FILE** is specified, any previously opened output file is closed and the new file is opened. If the file already exists, its contents are discarded unless **APPEND** is specified.
2. The line along which the extraction is to occur is determined as follows:
   a. If a material interface is specified (i.e., exactly one of **MATERIAL**, **SILICON**, **OXIDE**, **OXYNITRI**, **NITRIDE**, **POLYSILI**, **PHOTORES**, and **ALUMINUM** is specified along with exactly one of **/MATERIA**, **/SILICON**, **/OXIDE**, **/OXYNITR**, **/NITRIDE**, **/POLYSIL**, **/PHOTORE**, **/ALUMINU**, and **/AMBIENT**), it is used as the line.

      The extraction can be limited to a portion of an interface by specifying a starting point with **P1.X** and **P1.Y** and an ending point with **P2.X** and **P2.Y**, in which case the start of the line is taken to be the point on the interface closest to the specified starting point, and the end of the line is taken to be the point on the interface closest to the specified ending point.

      By default, the starting and ending points are taken to be at the left and right edges of the structure, respectively, at $y = 0$. If more than one path exists along the interface between the starting and ending points, a choice is made based on the value of the **CLOCKWIS** parameter.

If two or more interfaces exist between the specified materials, only one is used; the interface to be used can be selected using **P1.X**, **P1.Y**, **P2.X**, and **P2.Y**.

b. If at least one of **P1.X**, **P1.Y**, **P2.X**, and **P2.Y** (but no material interface) is specified, the points (**P1.X**, **P1.Y**) and (**P2.X**, **P2.Y**) are used as the starting and ending points of a straight line.

c. If neither of the above is specified, the value of **Y** is used to define a horizontal line through the structure. If **Y** is not specified, the value of **X** is used to define a vertical line through the structure. If neither **Y** nor **X** is specified, then no line is defined and no values are extracted.

If one or more materials are specified, the starting point for extraction is taken to be the point where the line first enters a region of one of the specified materials, and the ending point is taken to be the point where the line first leaves one of the specified materials. If no materials are specified, the starting and ending points are taken to be the points where the line first enters and leaves the structure. Distance is always measured from the first point on the line that lies in one of the specified materials (or in any portion of the structure, if no materials were specified).

3. The point at which extraction is to occur is determined as follows:
   a. If both **X** and **Y** are specified, their values are used as the coordinates of the point.
   b. If **X** or **Y** is specified and a line is defined by its endpoints or as an interface, extraction occurs at the point on the line where the *x* or *y* coordinate takes the specified value.
   c. If **DISTANCE** is specified, extraction occurs at the specified distance from the start of the line.
   d. If **VALUE** is specified, extraction occurs at the first point along the line where the selected variable has the specified value.
   e. If **MINIMUM** or **MAXIMUM** is specified, extraction occurs at the first point along the line where the selected variable takes on its minimum or maximum value. Only values along the line are considered when computing the minimum and maximum values.

   If none of the above is specified, then no point is defined and extraction occurs along the line specified by step 2 above.

4. If a point is specified in step 3, the extracted result consists of the **PREFIX** followed (with no added spaces) by the values of the *x* location, *y* location, distance from the start of the line, and the selected value (if specified by **X.EXTRAC**, **Y.EXTRAC**, **D.EXTRAC**, and **VAL.EXTR**, respectively) separated by **SEPARAT**, followed (again with no added spaces) by the **SUFFIX**. If no point is specified in step 3, the extracted result consists of the **PREFIX** followed (with no added spaces) by the integral of the selected variable, the length of the line, and the average of the selected variable along the line (if specified by **INT.EXTR**, **AREA.EXT**, and **AVG.EXTR**, respectively) separated by **SEPARAT**, followed (again with no added spaces) by the **SUFFIX**.

5. If **PRINT** is true, the extracted result is printed to the user's terminal and to the output listing file; if **WRITE** is true and an extract output file is open, the extracted result is written to the extract output file.
6. If **NAME** is specified, the extracted result is assigned as its value. The assigned variable follows the macro expansion rule as default. If the **ASSIGN** parameter is specified, the variable is assigned in the same way as it is assigned by the **ASSIGN** statement.
7. If **CLOSE** is specified and an extract output file is open, the file is closed.
8. If **T.FILE** is specified, extraction along a line horizontal or vertical must be specified by step 2 above, unless **Z.VALUE** is defined. The value of variable axis corresponds to the data of the column specified by **V.COLUMN**.

Some of the capabilities of the **EXTRACT** statement are illustrated by the following examples.

Targets for Optimization

The **EXTRACT** statement can be used to define targets for optimization by specifying the **NAME** parameter and either the **TARGET** or **T.FILE** parameter. Within an optimization loop, the **TARGET** and **T.FILE** parameters define the desired target values which the optimization attempts to achieve by varying the values of assigned names. The optimization attempts to simultaneously achieve the desired values of all targets defined within an optimization loop.

File Formats

The file specified by the **T.FILE** parameter is written in the same format as used for the file specified by the **IN.FILE** parameter in **PLOT.1D** statement:

1. Lines that are blank or contain a slash (/) as the first nonblank character are ignored and can be used to document the file.
2. Other lines define the data at one point in the distribution. These lines must contain the following values:
   a. Value number **V.COLUMN** is the variable data of the point.
   b. Value number **T.COLUMN** is the desired data of the point.

   If this line contains less than N numerical values in free-field format, where N is the maximum of indices (**V.COLUMN** and **T.COLUMN**) for the values listed above, the line is ignored.

Error Calculation

Targets that are defined for an optimization loop require the calculation of the error between the extracted and desired target values. A single RMS error is obtained by combining these errors for all targets defined within the optimization loop. The RMS error is used to control the search for an optimal solution and to determine when to terminate the optimization process. The error for a target value is calculated either as relative error or as absolute error, depending on the desired target value V_{des} and the minimum significant target value V_{min}. V_{des} is either a single value specified by the **TARGET** parameter or a set of values obtained from the input data file specified by the **T.FILE** parameter. V_{min} is given by:

$$V_{min} = \max(\mathbf{MIN.REL}|V_{des}|,\ \mathbf{MIN.ABS}) \qquad \text{Equation 4-2}$$

A relative error calculation is used if V_{des} is greater than V_{min}. The relative error is given by:

$$error = V_{weight}\frac{V_{ext} - V_{des}}{V_{des}} \qquad \text{Equation 4-3}$$

where V_{ext} is the target value extracted by the **`EXTRACT`** or **`ELECTRICAL`** statements and V_{weight} is the product of the target weight defined by the **`WEIGHT`** parameter.

An absolute error calculation is used if V_{des} is less than V_{min}. The absolute error is given by:

$$error = V_{weight}\frac{V_{ext} - V_{des}}{V_{min}} \qquad \text{Equation 4-4}$$

Relative error calculations are performed for single target values. Both relative and absolute error calculations can be performed for a set of target values obtained from the input data file specified by the **`T.FILE`** parameter. In this case, absolute error calculations are used for target values that are smaller by a factor of **`MIN.REL`** than the maximum absolute target value in the set. The use of absolute error calculation for these target values prevents large relative errors associated with insignificant target values from controlling the optimization process.

Examples

1. Open a file named *data.ext* to receive extracted results:

```
EXTRACT OUT.FILE=data.ext
```

2. Extract the value of net doping at (x,y) = (1.5,0.25) and write it to the extract file:

```
SELECT Z=DOPING
EXTRACT X=1.5 Y=0.25 VAL.EXTR
```

3. Extract the depth of the first junction in silicon at x=1.5:

```
SELECT Z=DOPING
EXTRACT SILICON X=1.5 VALUE=0.0 D.EXTRAC
```

4. Extract the width and integrated doping of the base of a bipolar transistor, assuming that the center of the emitter is at x=1.5:

```
SELECT Z=DOPING
%EXTRACT SILICON X=1.5 VALUE=0.0 Y.EXTRAC NAME=J1
%EXTRACT SILICON P1.X=1.5 P1.Y=(@{J1}+0.001) +
   VALUE=0.0 Y.EXTRACT NAME=J2
EXTRACT P1.X=1.5 P1.Y=@{J1} P2.Y=@{J2} +
   INT.EXT AREA.EXT
```

The first **`EXTRACT`** statement extracts the y location of the emitter-base junction by finding the first point along the line at x=1.5 where the net doping is zero. The second **EXTRACT** statement extracts the *y* location of the base-collector junction using a similar technique, but starting at a point 0.001 microns below the first junction. The third **EXTRACT** statement calculates the length and the integral of the doping along the line between the two junctions. The "%" characters are used on the first two **EXTRACT** statements to prevent substitution of any previous definitions of the names *J1* and *J2*. *Note* that the results of the first two extractions are printed and/or written to the extract file; this can be prevented by specifying **^PRINT** or **^WRITE**.

5. Extract the thickness of the topmost oxide layer at *x*=0:

```
EXTRACT OXIDE X=0.0 THICKNES
```

Note:

`THICKNES` is a synonym for the `AREA.EXT` parameter.

6. Extract the thickness of the gate oxide of a polysilicon-gate transistor at *x*=0:

```
%EXTRACT POLY X=0.0 DISTANCE=0.001 Y.EXTRACT +
   NAME=YPOLY
EXTRACT OXIDE P1.X=0.0 P1.Y=@{YPOLY} THICKNES
```

The first **EXTRACT** statement finds the *y* location of a point 0.001 microns into the poly gate. The second **EXTRACT** statement uses this as a starting point to search for the gate oxide.

7. Find the gate length of a poly-gate MOSFET assuming that the gate is located near *y*=0:

```
EXTRACT POLY /OXIDE ^CLOCKWIS AREA.EXT
```

This statement finds the length of the polysilicon/oxide interface, starting at the point closest to the left edge of the structure at *y*=0 and ending at the point closest to the right edge of the structure at *y*=0. The interface is traversed in the counterclockwise direction.

8. Create a file containing doping as a function of *y* location in silicon at *x*=0:

```
SELECT Z=DOPING
EXTRACT OUT.FILE=profile +
  PREFIX="/Net doping vs. Y"
FOREACH DEPTH (0 TO 4.0 STEP 0.02)
  EXTRACT SILICON X=0.0 DISTANCE=@{DEPTH} +
   Y.EXT VAL.EXT
END
EXTRACT CLOSE
```

The first **EXTRACT** statement opens the output file and writes a line of text to it. The **FOREACH** loop steps through the structure from a depth of zero to four microns, in 0.02 micron increments. The second **EXTRACT** statement extracts the *y* location and doping values. The third **EXTRACT** statement closes the file.

9. Extract the arsenic surface concentration at *x*=0.25:

```
SELECT Z=ARSENIC
EXTRACT SILICON X=0.25 DISTANCE=0.0 VAL.EXTR
```

10. Extract the coordinate at the right edge point of an oxide/silicon interface:

```
SELECT Z=X
EXTRACT OXIDE /SILICON MAXIMUM X.EXTRAC
EXTRACT OXIDE /SILICON MAXIMUM Y.EXTRAC
```

Optimization Examples

1. Extract the process conditions for the given oxide thickness and junction depth:

```
INIT B=1E15
LOOP   OPTIMIZE

  $ Specify the parameters to be optimized
  ASSIGN NAME=TEMP N.V=1000 LOWER=900 +
    UPPER=1200
  ASSIGN NAME=DOSE N.V=1E14 LOWER=1E12 +
    UPPER=1E15 LOG

  $ Process
  IMPLANT PHOS DOSE=@DOSE ENERGY=80
  DIFFUSE TEMP=@TEMP TIME=20 DRYO2

  $ Specify the targets - TOX and XJ
  EXTRACT NAME=TOX X=0 THICKNESS +
    OXIDE TARGET=0.06
  SELECT Z=DOPING
  EXTRACT NAME=XJ X=0 VALUE=0 D.EXTRAC +
    TARGET=0.9

L.END
ASSIGN NAME=TOX PRINT
ASSIGN NAME=XJ  PRINT
```

The value of implant dose is varied logarithmically by the **LOG** parameter during optimization because the difference between its lower and upper bound values is too large. *Note* that the extracted result is to be assigned to the **NAME** parameter, as if it had been defined with a **ASSIGN** statement, when either **TARGET** or **T.FILE** is specified.

2. Extract the SPICE JCAP area capacitance parameters:

```
INIT    P=1E16
IMPLANT BORON DOSE=1E14 ENERGY=50
DIFFUSE TEMP=1000 TIME=30 STEAM
ELECTRI JCAP JUNCTION=1 V="0 5 0.5" +
        OUT.F=jcap.dat NAME=CJ0 V.SE-
LECT=0.0

LOOP OPTIMIZE

  $ Specify the SPICE JCAP parameter
    to be optimized
  ASSIGN NAME=CJA N.V=@CJ0 LOWER=@CJ0/2 +
         UPPER=@CJ0*2
  ASSIGN NAME=VJA N.V=0.7  LOWER=0.1    +
         UPPER=1.0
  ASSIGN NAME=MJA N.V=0.33 LOWER=0.1
         UPPER=1.0

  $ Define the JCAP model in SPICE
  EXTRACT NAME=JCAP T.FILE=jcap.dat +
          Z.VALUE=@CJA/(1+V/@VJA)^@MJA

L.END
```

The **Z.VALUE** describes the modeling of a junction capacitance in SPICE. *Note* that a character “**V**” is used as a reserved keyword in the definition of the **Z.VALUE** parameter. Since the CJA is the value of a unit areal junction capacitance at zero bias, its initial value is taken at zero value of the **V.SELECT** parameter in the **ELECTRICAL** statement.

3. Extract the model coefficients concerned with oxidation-enhanced-diffusion and segregation by fitting a SIMS profile:

```
$ Extract the coefficients, THETA.0 and SEG.0
INIT     P=1E15
DEPOSIT  OXIDE THICKNES=0.03
IMPLANT  BORON DOSE=5E13 ENERGY=40

LOOP     OPTIMIZE PLOT

  ASSIGN NAME=THETA0 N.V=0.01    LOWER=0.0 +
         UPPER=0.1
  ASSIGN NAME=SEG0   N.V=1.126E3 LOWER=1.0 +
         UPPER=1E5 LOG

  INTERS SILICON /OXIDE THETA.0=@THETA0
  BORON  SILICON /OXIDE SEG.0=@SEG0
  DIFFUS TEMP=950 TIME=30 DRYO2

  EXTRAC NAME=YSURF SILICON DISTANCE=0 +
         X=0 Y.EXTRAC ASSIGN
  SELECT Z=LOG10(BORON)
  EXTRAC NAME=boron_sims SILICON X=0 VAL.EXT +
         T.FILE=boron.sims T.LOWER=1E15 +
         V.TRANSF=V+@YSURF T.TRANSF=LOG10(T)
+
         TOLER=0.1 SENS

L.END
```

This example assumes that the file `boron.sims` includes the SIMS data on a linear scale measured from a silicon sample. The **T.LOWER** parameter is used to eliminate the noise in the SIMS measurement at low concentrations. **V.TRANSF** is used to match the vertical coordinate at the surface of silicon since the segregation coefficient has the most sensitivity for the data near interface. The fitting for impurity profile is based on the logarithmic scale data because the distribution of impurity profile has a large range, so **Z** is specified as the log of the simulation result and **T.TRANSF** is used to take the log of the SIMS data. *Note* that the character "**V**" is used as a reserved keyword in the definition of the **V.TRANSF** parameter and that the character "**T**" is used in the definition of the **T.TRANSF** parameter.

ELECTRICAL

The **`ELECTRICAL`** statement is used to extract electrical characteristics.

```
ELECTRICAL

    [X=<n>]
    [ {  ( SRP
           [ANGLE=<n>] [PITCH=<n>] [ {POINT=<n> | DEPTH=<n>} ]
           [Y.SURFAC=<n>]
         )
      |  ( {V=<c> | (VSTART=<n> VSTOP=<n> VSTEP=<n>)}
           { ( RESISTAN [EXT.REG=<n>] [BIAS.REG=<n>] )
           | ( JCAP [JUNCTION=<n>] )
           | ( {   ( MOSCAP [HIGH] [LOW] [DEEP] )
               |   ( THRESHOL [VB=<n>] )
               }
               {NMOS | PMOS} [QM]
               [QSS=<n>] [GATE.WF=<n>] [GATE.ELE]
               [BULK.REG=<n>] )

           }
           [BULK.LAY=<n>] [PRINT] [DISTRIB]
         )
      }
    ]
    [TEMPERAT=<n>]
    [OUT.FILE=<c>]
    [NAME=<c> [V.SELECT=<n>]
     { TARGET=<n> [SENSITIV]
     | T.FILE=<c> [V.COLUMN=<n>] [V.LOWER=<n>] [V.UPPER=<n>]
                  [T.COLUMN=<n>] [T.LOWER=<n>] [T.UPPER=<n>]
                  [V.TRANSF=<c>] [T.TRANSF=<c>] )
                  [Z.VALUE]
     }
     [TOLERANC=<n>] [WEIGHT=<n>] [MIN.REL=<n>] [MIN.ABS=<n>]
    ]
```

Parameter	Type	Definition
`X`	number	The *x* coordinate of a vertical section along which electrical characteristics are extracted. **Units:** microns **Default:** none
`SRP`	logical	Specifies that a simulated spreading resistance profile is to be extracted. **Default:** false
`ANGLE`	number	The beveling angle for SRP extraction. **Units:** degrees **Default:** 2.0

Parameter	Type	Definition
PITCH	number	The probing pitch for SRP extraction. **Units:** microns **Default:** 5.0 **Synonym**: **SPACE**
POINT	number	The number of probing points for SRP extraction. **Default:** 50
DEPTH	number	The beveling depth for SRP extraction. **Units:** microns **Default:** none
Y.SURFAC	number	The starting value of *y* for the SRP profile. **Units:** microns **Default:** 0.0
V	character	This parameter is interpreted as a series of numeric values, separated by spaces or commas. The series must be composed of three values which is ordered by starting voltage, final voltage and incremental voltage. **Units:** volts **Default:** none
VSTART	number	The starting voltage. **Units:** volts **Default:** none
VSTOP	number	The final voltage. **Units:** volts **Default:** none
VSTEP	number	The incremental voltage. **Units:** volts **Default:** none
RESISTAN	logical	Specifies that the sheet resistance is to be extracted. **Default:** true
EXT.REG	number	The bottom-up number of the doping region in which the sheet resistance is to be extracted. **Default:** none **Synonym**: **PLOT.REG**
BIAS.REG	number	The bottom-up number of the doping region to which the given bias is applied when extracting the sheet resistance. **Default:** none
JCAP	logical	Specifies that junction capacitance is to be extracted. **Default:** false
JUNCTION	number	The bottom-up number of the junction at which the capacitance is to be extracted. **Default:** 1

Parameter	Type	Definition
`MOSCAP`	logical	Specifies that MOS capacitance is to be extracted. **Default:** false
`HIGH`	logical	Select the slow DC and fast AC input signal to extract MOS capacitance. **Default:** true
`LOW`	logical	Select the slow DC and slow AC input signal to extract MOS capacitance. **Default:** false
`DEEP`	logical	Select the fast DC and fast AC input signal to extract MOS capacitance. **Default:** false
`THRESHOL`	logical	Specifies that the MOS threshold voltage is to be extracted. The extracted threshold voltage is the x-intercept value extrapolated from the gate bias at which the slope, G_m, is maximum. **Default:** false
`VB`	number	The back bias applied to the bulk of the MOS transistor. **Units:** volts **Default:** 0.0
`NMOS`	logical	Specifies N-channel MOSFET. **Default:** false
`PMOS`	logical	Specifies P-channel MOSFET. **Default:** false
`QM`	logical	Specifies that the quantum effect is to be considered. **Default:** false
`QSS`	number	The surface fixed-state density at the interface between silicon and gate oxide. **Units:** #/cm^2 **Default:** 1e10 **Synonym**: **`QF`**
`GATE.WF`	number	The work function of the gate material. **Units:** volts **Default:** The current value for this material **Synonym**: **`WORKFUNC`**
`GATE.ELE`	logical	Specifies that the region defined as polysilicon gate is treated as electrodes. **Default:** false **Synonym**: **`CONDUCTO`**
`BULK.REG`	number	The bottom-up index of the doping region of the bulk of MOS transistor within **`BULK.LAY`**. **Default:** automatically decided
`BULK.LAY`	number	The bottom-up index of the material layer in which the extraction is to be performed. **Default:** the number of the first semiconductor layer

Parameter	Type	Definition
PRINT	logical	Specifies that the electron and hole charges, conductances, and resistances be printed for each bias. **Default:** true
DISTRIB	logical	Specifies that the spatial distributions of potential, electron concentration, and hole concentration be printed for each bias. **Default:** false
TEMPERAT	number	The device temperature used during the solutions of Poisson's equation. **Units:** ° Celsius **Default:** 25.0
OUT.FILE	character	The identifier for the file in which the electrical information is saved. **Default:** none
NAME	character	Specifies that the extracted result is to be assigned the specified name, as if it had been defined with an **ASSIGN** statement. The **NAME** parameter allows extracted results to be used in subsequent extractions or simulations. **Default:** none
V.SELECT	number	The variable value at which the assigned value to the **NAME** parameter is to be extracted. The **V.SELECT** parameter can not be used together with **THRESHOL** parameter. **Units:** determined by the extracted values **Default:** none
TARGET	number	The desired value of a target to be used for optimization. The optimization attempts to match the extracted value with the value of this parameter. **Units:** determined by the extracted values **Default:** none
SENSITIV	logical	Specifies the plot of sensitivity analysis of the target defined by this **ELECTRICAL** statement. This parameter works only if the **PLOT** parameter in the **LOOP** statement is true. **Default:** true
T.FILE	character	The file name containing the desired values of the target being defined for an optimization loop. The desired values are in the column specified by the **T.COLUMN** parameter. The optimization attempts to match the extracted values with the desired values. **Default:** none
V.COLUMN	number	The index of the column in the file specified by the **T.FILE** parameter, which contains the variable at which the extraction is performed. **Units:** none **Default:** 1

Parameter	Type	Definition
V.LOWER	number	The lower limit of the variable to be read from **T.FILE**. The target data at variable values less than the value of **V.LOWER** parameter are excluded during an optimization loop. *Note* that the **V.LOWER** specifies the value prior to the transformation by **V.TRANSF**. **Units:** the same as the unit of variable specified by **V.COLUMN** **Default:** -1e+30
V.UPPER	number	The upper limit of the variable to be read from **T.FILE**. The target data at variable values greater than the value of **V.UPPER** parameter are excluded during an optimization loop. *Note* that the **V.UPPER** specifies the value prior to the transformation by **V.TRANSF**. **Units:** the same as the unit of variable specified by **V.COLUMN** **Default:** 1e+30
T.COLUMN	number	The index of the column in the file specified by the **T.FILE** parameter, which contains the desired values of the target being defined for an optimization loop. **Units:** none **Default:** 2
T.LOWER	number	The lower limit of the desired value of a target to be read from **T.FILE**. Target values less than the value of **T.LOWER** parameter are excluded during an optimization loop. *Note* that the **T.LOWER** specifies the value prior to the transformation by **T.TRANSF**. **Units:** the same as the unit of variable specified by **T.COLUMN** **Default:** -1e+30
T.UPPER	number	The upper limit of the desired value of a target to be read from **T.FILE**. Target values greater than the value of **T.UPPER** parameter are excluded during an optimization loop. *Note* that the **T.UPPER** specifies the value prior to the transformation by **T.TRANSF**. **Units:** the same as the unit of variable specified by **T.COLUMN** **Default:** 1e+30
V.TRANSF	character	The function for transformation of the variable data read from **T.FILE**. The specified character string represents the function of a variable **V**, with which the variable data specified by **V.COLUMN** parameter are transformed. **Default:** none
T.TRANSF	character	The function for transformation of the desired data of a target read from **T.FILE**. The specified character string represents the function of a variable **T**, with which the target data specified by **T.COLUMN** parameter are transformed. **Default:** none
Z.VALUE	character	The function which generates the data instead of simulation. The specified character string represents a function of the variable **V**, which corresponds to the variable specified by **V.COLUMN** parameter. **Units:** the same as the unit of target data specified by **T.COLUMN** **Default:** none

Parameter	Type	Definition
TOLERANC	number	The RMS (root-mean-square) error for convergence criterion. An optimization loop terminates when the RMS errors of all of the specified targets are smaller than **TOLERANC**. **Units:** % **Default:** 0.0
WEIGHT	number	The weighting factor applied to the target being defined for an optimization loop. The weights are used to control the importance of individual targets in calculations of the error during optimization. **Units:** none **Default:** 1.0
MIN.REL	number	The minimum target ratio for which relative error is used to calculated the error during optimization. This value is compared with the ratio of the absolute target value to the maximum absolute target value defined by this **ELECTRICAL** statement. The use of this parameter is described at the end of this section. **Units:** none **Default:** 1e-10
MIN.ABS	number	The minimum target value for which relative error is used to calculate the error during optimization. This value is compared with the absolute target value. The use of this parameter is described at the end of this section. **Units:** determined by the extracted values **Default:** 1e-10

Description

The **ELECTRICAL** statement solves the 1D Poisson's equation along a vertical section through the structure. The coordinate of the vertical section is specified by **X**. Locations of features within the structure are specified by material layer number and by doping region numbers within layers. Layers and regions are numbered from bottom up, starting with layer or region number 1.

Files and Plotting

The extracted values can be plotted by specifying the **ELECTRIC** parameter on the next **PLOT.1D** statement. They can also be saved in the file specified by **OUT.FILE**. The file is in a text format with two columns; the first represents the independent variable (voltage or depth) and the second contains the extracted information. Extracted values saved in a file can be plotted using the **IN.FILE** parameter on the **PLOT.1D** statement.

Examples

1. The following statement shows the summary including the information of structure and the sheet resistance for each diffusion region.

```
ELECTRIC  X=1.0
```

2. The following statements extract the threshold voltage and plot the gate voltage vs. the sheet conductance in channel.

```
ELECTRIC  X=0.0 THRESHOLD NMOS V="0 2 0.05"
PLOT.1D   ELECTRIC
```

3. The following statement extracts the threshold voltage of NMOS with a highly ($\geq 10^{20} / \text{cm}^3$) phosphorus-doped polysilicon gate.

```
ELECTRIC  X=0.0 THRESHOLD NMOS V="0 2 0.05" +
               GATE.WF=4.35 GATE.ELE
```

4. The following statement extracts the high-frequency MOS capacitance for each bias and saves it in file *vgvscap*, in a text format.

```
ELECTRIC  X=0.0 MOSCAP NMOS V="-5 5 0.5" +
               OUT.FILE=vgvscap
```

5. The following statement extracts the junction capacitance for each reverse bias.

```
ELECTRIC  X=1.0 JCAP JUNCTION=2 V="0 5 0.1"
```

6. The following statements extract the sheet resistance for each bias and plot the voltage vs. sheet resistance. This gives the information about the increase of resistance due to the expansion of depletion region.

```
ELECTRIC  X=2.5 RESIST V="0 5 0.5" PLOT.REG=2 +
          BIAS.REG=2
PLOT.1D   ELECTRICAL
```

7. The following statements show how to save the output file to be used in the Taurus Workbench environment.

```
ELECTRIC  NAME=VTH THRESHOLD NMOS V="0 3 0.1"
EXTRACT   OUT.FILE=TWB.out
SELECT    Z=@VTH
EXTRACT   NAME=VT X=0 Y=0 VAL.EXT +
          PREFIX="Vth_ext " SUFFIX=" Volts"
EXTRACT   CLOSE
```

Optimization Examples

1. Extraction of process conditions obtains the desired threshold voltage.

```
INITIAL   P=1E15
DEPOSIT   OXIDE THICKNES=0.03

LOOP      OPTIMIZE
  ASSIGN NAME=DOSE N.V=1E13 LOWER=1E11 +
          UPPER=1E14 LOG

  IMPLANT BORON DOSE=@DOSE ENERGY=40
  ETCH    OXIDE ALL
  DIFFUSE TEMP=1000 TIME=30 DRYO2
  DIFFUSE TEMP=950  TIME=20 INERT
  DEPOSIT POLYSILI THICKNES=0.2

  ELECTRI NAME=VTH THRESHOLD NMOS +
          V="0 3 0.1" GATE.WF=4.35 +
          GATE.ELE TARGET=0.7
L.END
```

2. Extraction of process conditions obtains the desired sheet resistance.

```
INITIAL   P=1E15
LOOP      OPTIMIZE
  ASSIGN  NAME=DOSE N.V=1E13 LOWER=1E11 +
          UPPER=1E15 LOG

  IMPLANT BORON DOSE=@DOSE ENERGY=30
  DIFFUSE TEMP=1000 TIME=50 DRYO2

  ELECTRI NAME=RS RESIST EXT.REG=2 TARGET=200
L.END
```

Quantum Effect in CV Plot

The quantum effect becomes important when the gate oxide is thinner and the substrate doping concentration is higher. The following example shows the quantum effect in the case of 68Å for the gate oxide thickness and $1.37\text{x}10^{17}/\text{cm}^3$ for the substrate doping concentration. The graph shown in Figure 4-2 compares three types of capacitance extraction:

- Nonquantum effect and a polysilicon gate as conductor
- Nonquantum effect and a polysilicon gate as semiconductor
- Quantum effect and a polysilicon gate as semiconductor

  The measured data are referenced from Rafael Rios, et al. Reference [106].

The depletion in a polysilicon gate occurs in the very thin layer near the interface. Although the polysilicon for a gate electrode is heavily doped, the thin layer near the interface can be depleted as the voltage bias to the polysilicon gate increases. The formation of a thin depletion layer usually occurs when the gate voltage is higher than its threshold voltage. Therefore, the depletion effect of polysilicon can be ignored for the subthreshold current and the threshold voltage characteristics. However, as the gate voltage increases enough to deplete the interface region in the polysilicon, the effect no longer can be neglected. If the parameter **GATE.ELE** is not specified, the polysilicon gate is considered to be a semiconductor. Accurate simulation requires regrid for making the grid dense near the interface because the depletion layer is very thin. TSUPREM-4 regrids automatically if the parameter **E.REGRID** is specified as true (default) in the **METHOD** statement.

```
# without poly-depletion effect
ELECTRIC X=0 MOSCAP NMOS V="-4 4 0.1" LOW GATE.ELE
```

```
# with poly-depletion effect
ELECTRIC X=0 MOSCAP NMOS V="-4 4 0.1" LOW
```

The introduction of the quantum effect produces a much better result. The parameter **QM** specifies that the quantum effect is to be considered. Similar to polysilicon depletion, quantization occurs near the interface so that automatic regridding is performed. The parameter **QM.YCRIT** (default: 20Å) in the **MATERIAL** statement determines how far the quantization goes.

```
# QM effect and gate material as semiconductor
ELECTRIC X=0 MOSCAP NMOS V="-4 4 0.1" LOW QM
```

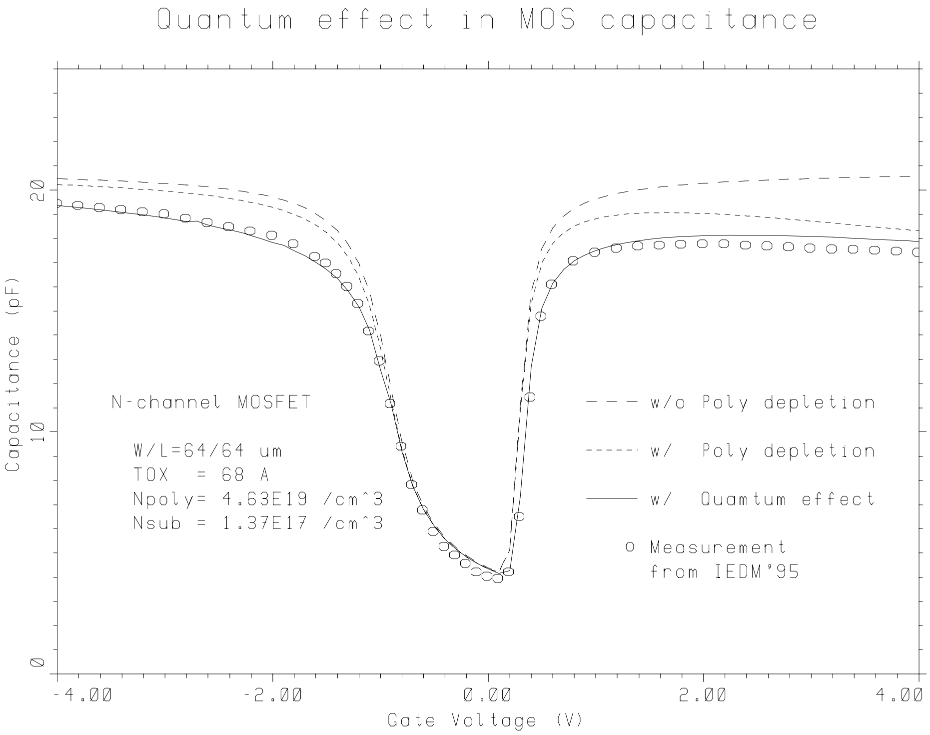


Figure 4-2 Quantum effect in MOS capacitance

Additional ELECTRICAL Notes

1. In the MOS capacitance of depletion MOSFET, only the **LOW** capacitance can be extracted.
2. The characteristics of gate material can be specified by the **MATERIAL** statement. For instance, the above example 3 can be replaced as follows:

```
MATERIAL  POLYSILI CONDUCTOR WORKFUNC=4.35
ELECTRIC  X=0.0 THRESHOLD NMOS V="0 2 0.05" +
           QSS=1e10
```

3. The default table used in the **ELECTRICAL** statement is the same used in Medici, while the table used for conversion from resistivity to concentration in the **INITIALIZE** statement comes from Masetti's work Reference [2]. The mobility table used in the **ELECTRICAL** statement can be redefined by using the **MOBILITY** statement.

VIEWPORT

The **`VIEWPORT`** statement limits plotting to a subset of the available drawing surface. It can be used to scale plots.

```
VIEWPORT

   [X.MIN=<n>] [X.MAX=<n>] [Y.MIN=<n>] [Y.MAX=<n>]
```

Parameter	Type	Definition
`X.MIN`	number	A value between 0 and 1 specifying the left edge of the plotting region to use, as a fraction of the total width. **Units:** none **Default:** 0
`X.MAX`	number	A value between 0 and 1 specifying the right edge of the plotting region to use, as a fraction of the total width. **Units:** none **Default:** 1
`Y.MIN`	number	A value between 0 and 1 specifying the bottom edge of the plotting region to use, as a fraction of the total height. **Units:** none **Default:** 0
`Y.MAX`	number	A value between 0 and 1 specifying the top edge of the plotting region to use, as a fraction of the total height. **Units:** none **Default:** 0

Description

This statement specifies a subset of the available plotting area on which to plot. It works with all plotting calls. The viewport remains in the current state until it is reset with a subsequent **`VIEWPORT`** statement. A **`VIEWPORT`** statement with no parameters resets the viewport to the full extent of the plotting area.

The **`VIEWPORT`** statement does not take effect until the next plotting statement that specifies that axes be drawn.

The **`CLEAR`** options of the various plotting statements clear the whole screen, not just the current **`VIEWPORT`** area.

Scaling Plot Size

The **`VIEWPORT`** statement can be used to scale plots because the default axis lengths for **`PLOT.1D`**, **`PLOT.2D`**, and **`PLOT.3D`** depend on the viewport size. Scaling the plot size with **`VIEWPORT`** does not scale the size of titles, axis labels, or other objects that have absolute sizes (i.e., sizes given in centimeters).

Examples

The following statement causes the next plot to occupy 80% of the available plotting width, centered within that width:

```
VIEWPORT X.MIN=.1 X.MAX=.9
```

Models and Coefficients

The following statements specify the models and coefficients used by TSUPREM-4:

Statement Name	Description	Page
METHOD	Specifies models and numerical methods to be used.	p. 4-207
EQUATION	Defines user-specified equations. Only available with the USEIT Option.	p. 4-230
AMBIENT	Specifies parameters for the oxidation models.	p. 4-233
MOMENT	Specifies moment parameters for ion implantation.	p. 4-251
MATERIAL	Defines materials and specifies their properties.	p. 4-258
IMPURITY	Defines impurities and specifies their properties.	p. 4-273
REACTION	Defines the reactions that occur between impurities and materials at material interfaces.	p. 4-300
MOBILITY	Defines or modifies the parameters describing carrier mobility.	p. 4-305
INTERSTITIAL	Specifies coefficients for interstitials.	p. 4-311
VACANCY	Specifies coefficients for vacancies.	p. 4-327
ANTIMONY	Specifies coefficients for antimony.	p. 4-337
ARSENIC	Specifies coefficients for arsenic.	p. 4-344
BORON	Specifies coefficients for boron.	p. 4-351
PHOSPHORUS	Specifies coefficients for phosphorus.	p. 4-358

METHOD

The **METHOD** statement selects models for oxidation and diffusion, and specifies numerical methods.

```
METHOD

    [ {ERFC | ERF1 | ERF2 | ERFG | VERTICAL
      | COMPRESS | VISCOELA | VISCOUS} ] [ST.HISTO]
    [{ST.DIREC | ST.CG1 | ST.CG2}] [ST.LEVEL=<n>] [ST.RSTEP=<n>] [ST.FSTEP=<n>]
    [ST.DIFF] [ST.ALPHA] [ST.PDEQ] [ST.LIMIT] [STN.DIFF] [STN.MIS]
    [DEPO.MIS] [DIFF.MIS]
    [DY.OXIDE=<n>] [DY.EXACT] [DY.LOCAL] [GRID.OXI=<n>] [SKIP.SIL]
    [ {PD.FERMI | PD.TRANS | PD.FULL | PD.5STR} ]
    [NSTREAMS=<n>] [PAIR.GRA] [PAIR.SAT] [PAIR.REC] [PAIR.EQU] [T.EQPAIR=<n>]
    [PAIR.SEG] [PD.PFLUX] [PD.PTIME] [PD.PREC] [PD.NREC] [KO.FULL] [FT.FULL]
    [IMP.ADAP] [DIF.ADAP] [OX.ADAPT] [DAM.ADAP] [ERR.FAC=<n>] [UNREFINE=<n>]
    [ {ACT.EQUI | ACT.TRAN | ACT.FULL }] [DDC.FULL]
    [INIT.TIM=<n>] [TIME.DEL=<n>]
    [M.AUTO] [ {CG | GCR | SPLU} ] [BACK=<n>] [BLK.ITLI=<n>] [PRECOND=<n>]
    [MIN.FILL] [MIN.FREQ=<n>] [MF.METH=<n>] [MF.DIST=<n>]
    [MF.P.LEV=<n>] [MF.F.LEV=<n>]
    ( [IMPURITY=<c> ] [VACANCY] [INTERSTI] [ANTIMONY] [ARSENIC] [BORON]
        [PHOSPHOR] [OXIDANT] [TRAP]
      [ {FULL | PART | NONE} ] [SYMMETRY]
      [ {TIME.STE | ERROR | NEWTON} ]
      [REL.ERR=<n>] [ABS.ERR=<n>] [TEMP.MIN=<n>]
      ( [MATERIAL=<c>] [SILICON] [POLYSILI] [OXIDE] [OXYNITRI]
          [NITRIDE] [ALUMINUM] [PHOTORES]
        [REL.ADAP=<n>] [ABS.ADAP=<n>] [MIN.SPAC=<n>] [MAX.SPAC=<n>]
      )
    )
    [OX.REL=<n>] [CONTIN.M=<n>] [VE.SMOOT=<n>]
    [E.ITMIN=<n>] [E.ITMAX=<n>] [E.RELERR=<n>] [E.RVCAP=<n>] [E.AVCAP=<n>]
       [E.USEAVC]
    [E.REGRID] [E.TSURF=<n>] [E.DSURF=<n>] [E.RSURF=<n>]
    [ {MOB.TABL | MOB.AROR | MOB.CAUG} ]
    [ ITRAP [IT.CPL] [IT.ACT] [{IT.ZERO | IT.THERM | IT.STEAD}] ]
    [MODEL=<c> [ENABLE] ] [POISSON] [ADVANCED]
    [{C0.ANALY | C0.NUMER | C0.HYBRI}]
    [II.TOL.F=<n>] [II.IF.DY=<n>]
    [TABLE=<c>] [DAM.TABL=<c>] [BACKSCAT] [RANGE.SH] [D.SCALE=<n>]
    [TRAJ.REP] [X.INTERV] [TRAJ.SPL] [{DEPTH.SP |  COLL.DEN | TRAJ.DEN}]
    [PRINT]
```

Parameter	Type	Definition
ERFC	logical	Uses the simplest analytical model for oxidation. **Default:** the current value; initially false
ERF1	logical	Uses Guillemot’s “shape 1” analytical model for oxidation. **Default:** the current value; initially false
ERF2	logical	Uses Guillemot’s “shape 2” analytical model for oxidation. **Default:** the current value; initially false

Parameter	Type	Definition
`ERFG`	logical	Uses Guillemot's "shape 1" or "shape 2" analytical model, as appropriate, for oxidation. **Default:** the current value; initially false
`VERTICAL`	logical	Uses the vertical growth numerical model for oxidation. **Default:** the current value; initially true
`COMPRESS`	logical	Uses the compressible viscous flow numerical model for oxidation. **Default:** the current value; initially false
`VISCOELA`	logical	Uses the viscoelastic numerical model for oxidation. **Default:** the current value; initially false
`VISCOUS`	logical	Uses the incompressible viscous flow numerical model for oxidation. **Default:** the current value; initially false
`ST.HISTO`	logical	Calculates the stresses in the structure caused by material growth, thermal mismatch, intrinsic strain, and surface tension during all high-temperature steps. Only effective when the **`VISCOELA`** model is also active. **Default:** the current value; initially false
`ST.DIREC`	logical	Uses the sparse direct method for solving the stress equations. **Default:** the current value; initially false
`ST.CG1`	logical	Uses the iterative method for solving viscoelastic model for oxidation. Only applies when the **`VISCOELA`** model is selected. **Default:** the current value; initially false **Synonym: `ST.CG`**
`ST.CG2`	logical	Uses the iterative method for solving viscoelastic model for oxidation. Only applies when the **`VISCOELA`** model is selected. It is based on block preconditioning, and typically shows the better performance than **`ST.CG1`**. **Default:** the current value; initially false
`ST.LEVEL`	number	Controls the fill level of the factorization of the stress matrix **Units:** none **Default:** the current value; initially 4
`ST.RSTEP`	number	Controls the incremental ratio of time step. Only applicable for **`STRESS`** command. **Units:** none **Default:** the current value; initially 1.9
`ST.FSTEP`	number	Controls the scaling factor of time step. Only applicable for **`STRESS`** command. **Units:** none **Default:** the current value; initially 1.2
`ST.DIFF`	logical	Uses the stress dependent dopant-defect pair migration energy model for diffusion. **Default:** the current value; initially false
`ST.ALPHA`	logical	Uses the stress dependent dopant-defect pair formation energy model for diffusion. **Default:** the current value; initially false

Parameter	Type	Definition
ST.PDEQ	logical	Uses the pressure dependent equilibrium concentration of a point-defect. Only effective when **ST.HIST** and **VISCOELA** are also active. **Default:** the current value; initially false
ST.LIMIT	logical	Prevents unphysical large stress levels during the stress simulation. The absolute stress values for each stress component are bounded within the strength limit. **Default:** the current value; initially false
STN.DIFF	logical	Uses the strain dependent diffusivity enhancement model for diffusion. **Default:** the current value; initially false
STN.MIS	logical	Uses the generalized dopant induced stress model for computing lattice mismatch strain. Setting **STN.MIS** to true is identical to turning on both **DEPO.MIS** and **DIFF.MIS**. Only effective when **ST.HIST** and **VISCOELA** are also active. **Default:** the current value; initially false
DEPO.MIS	logical	Uses the generalized dopant induced stress model for computing lattice mismatch strain for deposition. Only effective when **ST.HIST** and **VISCOELA** are also active. **Default:** the current value; initially false
DIFF.MIS	logical	Uses the generalized dopant induced stress model for computing lattice mismatch strain for diffusion. Only effective when **ST.HIST** and **VISCOELA** are also active. **Default:** the current value; initially false
DY.OXIDE	number	The grid spacing to be used in growing oxides. This value is scaled by the value of the **GRID.FAC** parameter on the **MESH** statement (see p. 4-53). **Units:** microns **Default:** the current value; initially 0.1
DY.EXACT	logical	Specifies that the simulation time step should be reduced if needed to achieve precise grid spacing in growing layers. **Default:** the current value; initially true
DY.LOCAL	logical	Specifies that the grid spacing in a growing layer should be inversely proportional to the growth rate. **Default:** the current value; initially true
GRID.OXI	number	The ratio of grid spacing in a growing oxide to the grid spacing in the consumed silicon. Used only if **DY.OXIDE** is zero. A value of zero produces no grid in the oxide. **Units:** none **Default:** 0 for **ERFC**, **ERF1**, and **ERF2**; 2.2 for others
SKIP.SIL	logical	If true, silicon regions are treated as fixed, rigid structures during oxidation, with no stress calculation in the silicon. If false, silicon is treated as a viscous material, with stress calculations. **Default:** the current value; initially true

Parameter	Type	Definition
PD.FERMI	logical	Selects a model in which the point defect concentrations depend only on the Fermi level. Equivalent to setting **NSTREAMS**=1, **^PAIR.GRA**, **^PAIR.SAT**, **^PAIR.REC**, **^PD.PFLUX**, **^PD.PTIME**, and **^PD.PREC**. Does not model oxidation-enhanced diffusion. Recommended only where speed is more important than accuracy. This is the default selection at the start of a simulation. Equivalent to the **FERMI** model in older versions of TSUPREM-4. **Default:** the current value; initially true **Synonym: FERMI**
PD.TRANS	logical	Selects the simplest model that includes a full 2D solution for the point defect concentrations. Equivalent to setting **NSTREAMS**=3, **^PAIR.GRA**, **^PAIR.SAT**, **^PAIR.REC**, **^PD.PFLUX**, **^PD.PTIME**, and **^PD.PREC**. Models oxidation-enhanced diffusion. Recommended for routine simulations. Equivalent to the **TWO.DIM** model in older versions of TSUPREM-4. **Default:** the current value; initially false **Synonym: TWO.DIM**
PD.FULL	logical	Selects the 3-stream diffusion model available. Equivalent to setting **NSTREAMS**=3, **PAIR.GRA**, **PAIR.SAT**, **PAIR.REC**, **PD.PFLUX**, **PD.PTIME**, and **PD.PREC**. Recommended when using the implant damage model. **Default:** the current value; initially false **Synonym: FULL.CPL**
PD.5STR	logical	Selects the most complete diffusion model available. Equivalent to setting **NSTREAMS**=5. Recommended for simulations where the maximum available accuracy is needed. **Default:** the current value; initially false
NSTREAMS	number	Specifies the level of point defect diffusion modeling. The level is specified as the number of diffusion equations required to simulate diffusion with a single dopant species. A value of 1 indicates that no equations are to be solved for the point defects (i.e., the **PD.FERMI** model, while a value of 3 indicates that two equations are to be used, 1 for interstitials and one for vacancies (the **PD.TRANS** or **PD.FULL** models). A value of 5 specifies that the 5-stream diffusion model is to be used. Values other than 1, 3 and 5 are not meaningful. **Units:** none **Default:** the current value; initially 1
PAIR.GRA	logical	Specifies that the pair concentration terms be included in the gradient term (i.e., use Equations 3-32 and 3-33 rather than Equations 3-111 in Chapter 3). **Default:** the current value; initially false
PAIR.SAT	logical	Specifies that pair saturation effects (the a_m and a_n terms in Equations 3-53 and 3-54) be included in the equations for dopant diffusion. **Default:** the current value; initially false
PAIR.REC	logical	Specifies that dopant-assisted recombination effects (the K_{mv} and K_{ni} terms in Equations 3-53 and 3-54)) be included in the equations for dopant diffusion. **Default:** the current value; initially false

Parameter	Type	Definition
PAIR.EQU	logical	Specifies that the pair equations are to be turned off when the differences of the results with/without solving the equations are within the tolerance error. **Default:** the current value; initially false
T.EQPAIR	number	The tolerance factor to the relative errors of the pair solutions to determine whether if the pair equations will be turned off. **Units:** none **Default:** 70.0
PAIR.SEG	logical	Specifies that the dopant-defect pairs are dissociated into a substitutional atom and a point-defect prior to the segregation transport from silicon to other materials. Turning on **PAIR.SEG** disables the surface recombination model for dopant-defect pairs. Only applicable for the 5-stream diffusion model. **Default:** the current value; initially false
PD.PFLUX	logical	Specifies that the dopant-defect pair fluxes (the J_m and J_n terms) be included in the equations for point defect diffusion (Equations 3-197, 3-301, and 3-302). Also causes the pair concentration terms to be included in the gradient term (i.e., use Equations 3-32 and 3-33 rather than Equations 3-111). **Default:** the current value; initially false
PD.PTIME	logical	Specifies that the time derivative of the dopant-defect pair concentrations (the dM/dt and dN/dt terms) be included in the equations for point defect diffusion (Equations 3-197, 3-301, and 3-302). **Default:** the current value; initially false
PD.PREC	logical	Specifies that dopant-assisted recombination effects (K_{mv} and K_{ni} terms) be included in the equation for interstitial-vacancy recombination (Equation 3-287). **Default:** the current value; initially false
PD.NREC	logical	Specifies that the recombination reduction due to trapping nitrogen be taken into account in the equation for interstitial-vacancy recombination (Equation 3-223). **Default:** the current value; initially false
KO.FULL	logical	Specifies to calculate the kick-out reaction rate by using the consistent diffusivity and intrinsic equilibrium concentration of point-defect with those of the point-defect diffusion flux equation. **Default:** the current value; initially false
FT.FULL	logical	Specifies to calculate the Frank-Turnbull reaction rate by using the consistent diffusivity and intrinsic equilibrium concentration of point-defect with those of the point-defect diffusion flux equation, as well as the consistent diffusivity of a dopant-defect pair with that of a dopant diffusion flux equation. **Default:** the current value; initially false
IMP.ADAP	logical	Enables adaptive grid modification during ion implantation. **Default:** the current value; initially true
DIF.ADAP	logical	Enables adaptive grid modification during diffusion. **Default:** the current value; initially true

Parameter	Type	Definition
`OX.ADAPT`	logical	Enables adaptive grid modification based on oxidant concentration. **Default:** the current value; initially false
`DAM.ADAP`	logical	Enables adaptive grid modification along amorphous/crystalline boundaries. **Default:** the current value; initially false
`ERR.FAC`	number	A factor that multiplies the relative errors **`REL.ADAP`** for adaptive gridding (both refinement and unrefinement). Larger values allow larger errors and produce coarser grids; smaller values reduce the error and produce finer grids. **Units:** none **Default:** the current value; initially 1.0
`UNREFINE`	number	A factor that multiplies the relative errors **`REL.ADAP`** for mesh unrefinement. Larger values allow more unrefinement, while smaller values allow less. A value of 0.0 disables mesh unrefinement. Values greater than 1.0 may cause excessive addition and removal of grid points, and should be used with caution. **Units:** none **Default:** the current value; initially 0.1
`ACT.EQUI`	logical	Use the equilibrium model for activation of dopants. **Default:** the current value; initially true
`ACT.TRAN`	logical	Use the transient model for activation of dopants (available only with the Extended Defects AAM). **Default:** the current value; initially false
`ACT.FULL`	logical	Use the dopant-defect clustering model for activation of dopants. **Default:** the current value; initially false
`DDC.FULL`	logical	Use the full dopant-defect clustering model for activation of dopants. **Default:** the current value; initially false
`INIT.TIM`	number	The size of the initial time step. **Units:** minutes **Default:** the current value; initially 0.002
`TIME.DEL`	number	The virtual delta time duration for abrupt temperature ramp. **Units:** minutes **Default:** the current value; initially 1.67×10^{-26}
`M.AUTO`	logical	Automatically tune the numerical method on the fly. **Default:** the current value; initially true
`CG`	logical	Use a conjugate residual method to solve the system of blocks. **Default:** the current value; initially true
`GCR`	logical	Use a modified conjugate residual method to solve the system of blocks. **Default:** the current value; initially false
`SPLU`	logical	Use a sparse direct method with pivoting to solve the system of blocks. **Default:** the current value; initially false

Parameter	Type	Definition
BACK	number	The number of back vectors to be used in the CG outer iteration. The maximum value is 48. More back vectors should give faster convergence but require more memory. **Units:** none **Default:** the current value; initially 18
BLK.ITLI	number	The maximum number of block iterations allowed. The block iteration terminates after this number of iterations, whether convergence is obtained or not. **Units:** none **Default:** the current value; initially 20
PRECOND	number	Selects the preconditioner to be used for CG and GCR methods. **PRECOND=1** selects the preconditioner used in previous releases while PRECOND=2 selects a new preconditioner. **PRECOND=2** is faster for difficult problems, including most cases using **PD.5STR** (i.e. **NSTREAM=5**); **PRECOND=1** is faster for most other problems. **PRECOND=2** may be faster for 1D (or quasi 1D) problems. **PRECOND=2** (with appropriate values of **MF.F.LEV** and **MF.P.LEV**) is faster than **SPLU** in most cases. **Units:** none **Default:** the current value; initially 1
MIN.FILL	logical	Use a minimum fill reordering for the matrices, to increase solution speed. **Default:** the current value; initially true
MIN.FREQ	number	Controls how much the solution matrix may increase before a new minimum fill reordering is done. A value of zero forces a minimum fill reordering whenever the matrix structure changes. **Units:** none **Default:** the current value; initially 1.1
MF.METH	number	Selects the algorithm to use for minimum fill reordering. A value of 0 selects the algorithm used prior to Version 6.6, while a value of 1 selects the algorithm prior to Version 2006.06. A value of 2 selects a minimum fill reordering method that reduces the number of low-order fill terms, which usually produces faster solutions than 0 or 1 methods. Units: none Default: the current value; initially 1
MF.DIST	number	The maximum distance between nodes, for which fill terms are included when **FULL** fill is specified. Smaller values decrease the time and memory required for matrix decomposition, but may increase the number of iterations required for solution; very large values (larger than the size of the structure) retain all fill terms, as in versions prior to 6.6. **Units:** microns **Default:** the current value; initially 0.5

Parameter	Type	Definition
MF.P.LEV	number	Limits the level of fill terms used for **METHOD PART** solutions (**PRECOND=1**) or for coupling between different solutions (**PRECOND=2**). Values are the same as for **MF.F.LEV**. The default value of 2 produces the same result for **PRECOND=1** as in previous versions of the program. On difficult problems with **PRECOND=2**, the solution may be considerably faster with a value of 1 or 3 for **MP.P.LEV**. A value of zero specifies that the level of fill terms is not limited. **MF.P.LEV** is only used when **MF.DIST** is nonzero. If **MF.P.LEV >** **MF.F.LEV**, the value of **MF.F.LEV** is used instead. **Units:** none **Default:** the current value; initially 2
MF.F.LEV	number	Limits the level of fill terms used for **METHOD FULL** solutions (**PRECOND=1**) or for coupling between nodes for a given solution (**PRECOND=2**). A value of 1 uses no fill terms; a value of 2 uses fill terms produced by original matrix entries; a value of 3 uses fill terms arising from level 2 fill terms; and so on. Larger values require more solution time per CG or GCR iteration but may reduce the number of iterations required. A value of zero specifies that the level of fill terms is not limited. **MF.F.LEV** is only used when **MF.DIST** is nonzero. **Units:** none **Default:** the current value; initially 5
IMPURITY	character	A list of one or more impurities (separated by spaces or commas) to which the remaining parameters apply. (The list must be enclosed in quotes if it contains spaces.) **Default:** none **Synonym: IMP**
VACANCY	logical	The remaining parameters apply to the solution for vacancies. **Default:** false **Synonym: VACANCIE**
INTERSTI	logical	The remaining parameters apply to the solution for interstitials. **Default:** false
ANTIMONY	logical	The remaining parameters apply to the solution for antimony. **Default:** false **Synonym: SB**
ARSENIC	logical	The remaining parameters apply to the solution for arsenic. **Default:** false **Synonym: AS**
BORON	logical	The remaining parameters apply to the solution for boron. **Default:** false **Synonym: B**
PHOSPHOR	logical	The remaining parameters apply to the solution for phosphorus. **Default:** false **Synonym: P**

Parameter	Type	Definition
`OXIDANT`	logical	The remaining parameters apply to the solution for oxidant. **Default:** false **Synonym: `OXYGEN`**
`TRAP`	logical	The remaining parameters apply to the solution for interstitial traps. **Default:** false
`FULL`	logical	Use all terms produced in factorization of the block matrices for the selected solution variables. **Default:** the current value; initially true for **`VACANCY`** and **`INTERSTI`** blocks, false for others
`PART`	logical	Use only nearest neighbor fill terms produced in factorization of the block matrices for the selected solution variables. **Default:** the current value; initially true for mobile species except **`VACANCY`** and **`INTERSTI`**, and false for all others
`NONE`	logical	Do not use fill terms produced during factorization of the block matrices for the selected solution variables. **Default:** the current value; initially true for immobile species, false for others
`SYMMETRY`	logical	Treat the block matrix equations for the selected solution variables as if they were symmetric. **Default:** the current value; initially false for all blocks **Synonym: `SYMMETRI`**
`TIME.STE`	logical	The matrices for the selected solution variables are refactored at each time step. **Default:** the current value; initially true for all blocks
`ERROR`	logical	The matrices for the selected solution variables are refactored whenever the error in the block is decreasing. **Default:** the current value; initially false for all blocks
`NEWTON`	logical	The matrices for the selected solution variables are refactored at each Newton step. **Default:** the current value; initially false for all blocks
`REL.ERR`	number	The relative error tolerance for solution of the impurity blocks for the selected solution variables. **Units:** none **Default:** the current value; initially 0.01 for all blocks
`ABS.ERR`	number	The absolute error tolerance for solution of the impurity blocks for the selected solution variables. **Units:** The units of the solution variable. **Default:** the current value; initially 1e5 for **`VACANCY`** and **`INTERSTI`** blocks, 1e9 for others
`TEMP.MIN`	number	The minimum temperature at which the equation is to be solved **Units:** ^oC **Default:** the current value; initially -273.16 ^oC for **`PSI`** and stress solutions, 450^oC for others

Parameter	Type	Definition
MATERIAL	character	A list of one or more materials (separated by spaces or commas), in which the specified values of **REL.ADAP**, **ABS.ADAP**, and **MIN.SPAC** apply. (The list must be enclosed in quotes if it contains spaces.) **Default:** none
SILICON	logical	The specified values of **REL.ADAP**, **ABS.ADAP**, and **MIN.SPAC** apply in single-crystal silicon. **Default:** false
POLYSILI	logical	The specified values of **REL.ADAP**, **ABS.ADAP**, and **MIN.SPAC** apply in polysilicon. **Default:** false
OXIDE	logical	The specified values of **REL.ADAP**, **ABS.ADAP**, and **MIN.SPAC** apply in oxide. **Default:** false
OXYNITRI	logical	The specified values of **REL.ADAP**, **ABS.ADAP**, and **MIN.SPAC** apply in oxynitride. **Default:** false
NITRIDE	logical	The specified values of **REL.ADAP**, **ABS.ADAP**, and **MIN.SPAC** apply in nitride. **Default:** false
ALUMINUM	logical	The specified values of **REL.ADAP**, **ABS.ADAP**, and **MIN.SPAC** apply in aluminum. **Default:** false
PHOTORES	logical	The specified values of **REL.ADAP**, **ABS.ADAP**, and **MIN.SPAC** apply in photoresist. **Default:** false
REL.ADAP	number	The relative error targets for adaptive gridding for the specified solutions and materials. A value of zero disables adaptive gridding for the specified solutions and materials. **Units:** none **Default:** the current value; see Appendix A for initial values
ABS.ADAP	number	The absolute error targets for adaptive gridding for the specified solutions and materials. **Units:** The units of the solution variable. **Default:** the current value; see Appendix A for initial values
MIN.SPAC	number	The minimum grid spacing produced by adaptive gridding for the specified solutions and materials. **Units:** microns **Default:** the current value; see Appendix A for initial values

Parameter	Type	Definition
`MAX.SPAC`	number	The maximum grid spacing produced by mesh unrefinement for the specified materials. *Note* that **`MAX.SPAC`** does not depend on the specification of impurities. **Units:** microns **Default:** the current value; initially 0.4 for silicon and 0.0 for other materials
`OX.REL`	number	The maximum relative error allowed when solving the stress-dependent oxidation equations. Values between 1e-4 and 1e-6 are recommended. **Units:** none **Default:** the current value; initially 1e-6
`CONTIN.M`	number	Specifies which continuation method should be used for solving the stress-dependent oxidation equations. Values from 2 to 7 are recognized, but only 2 or 7 should be used. **Units:** none **Default:** the current value; initially 2
`VE.SMOOT`	number	The amount of smoothing to be applied to stress values when the **`VISCOELA`** model is used. A value of 0.0 specifies a minimum of smoothing, while 1.0 specifies maximum smoothing. **Units:** none **Default:** the current value; initially 0.04
`E.ITMIN`	number	The minimum number of iterations required for each solution of Poisson’s equation in order to extract the electrical information by the **`ELECTRICAL`** statement. **Units:** none **Default:** 0
`E.ITMAX`	number	The maximum number of iterations allowed for each solution of Poisson’s equation in order to extract the electrical information by the **`ELECTRICAL`** statement. **Units:** none **Default:** 50
`E.RELERR`	number	The allowed relative error used to test for convergence during the iterative numerical solution of Poisson’s equation. This value determines the maximum relative change between successive approximations to the solution during iteration. This value is used to extract the electrical information by the **`ELECTRICAL`** statement. **Units:** none **Default:** 0.0001
`E.RVCAP`	number	The ratio of AC disturbance to DC incremental voltage to extract the capacitance by the **`ELECTRICAL`** statement. **Units:** none **Default:** 0.2
`E.AVCAP`	number	Disturbance voltage to extract the capacitance by the **`ELECTRICAL`** statement. **Units:** voltages **Default:** 0.025

Parameter	Type	Definition
`E.USEAVC`	number	Specifies that **`E.AVCAP`** is to be used for capacitance extraction in the **`ELECTRICAL`** statement. If false, **`E.RVCAP`** is used. **Default:** false
`E.REGRID`	logical	Specifies that the automatic regridding is to be performed before solving Poisson's equation. **Default:** true
`E.TSURF`	number	The thickness of surface region to be regridded by specifying **`E.REGRID`**. **Units:** um **Default:** 0.01
`E.DSURF`	number	The first grid space after regridding by specifying **`E.REGRID`**. **Units:** um **Default:** 0.0002
`E.RSURF`	number	The ratio between sequent grid spaces after regridding by specifying **`E.REGRID`**. **Units:** none **Default:** 1.2
`MOB.TABL`	logical	Specifies that the mobility tables are used to determine electron and hole mobilities. **Default:** true
`MOB.AROR`	logical	Specifies that the analytic mobility model based on the work of Arora, et al., is used to determine electron and hole mobilities. **Default:** false
`MOB.CAUG`	logical	Specifies that the analytic mobility model based on the work of Caughey, et al., is used to determine electron and hole mobilities. **Default:** false
`ITRAP`	logical	Use the interface trap model for segregation flux. **Default:** false
`IT.CPL`	logical	Specifies that each impurity occupies trap sites exclusive from each other. **Default:** false
`IT.ACT`	logical	Specifies that the active concentration is used when the material adjacent to the interface is either silicon or polysilicon. **Default:** true
`IT.ZERO`	logical	Specifies that the initial value of occupied trap density is zero. **Default:** true
`IT.THERM`	logical	Specifies that the initial value of occupied trap density is calculated with the coefficients on the **`IMPURITY`** statement, i.e., **`Q.INI.0`** exp(-**`Q.INI.E`**/kT). **Default:** false
`IT.STEAD`	logical	Specifies that the initial value of occupied trap density is calculated to satisfy the steady state in which neither accumulation nor depletion of trapped dopants at the interface occurs. **Default:** false

Parameter	Type	Definition
MODEL	character	The name of the user-specified model, which is defined in the **IMPURITY** or **REACTION** statements. **Default:** none
ENABLE	logical	Specifies that the user-specified model named by the **MODEL** parameter is turned on. **Default:** true
POISSON	logical	Specifies that Poisson's equation is to be solved instead of applying the charge neutrality. **Default:** false
ADVANCED	logical	Specifies that the coefficients and the models in *s4advanced.[version]* are to be used. **Default:** false
C0.ANALY	logical	Specifies that the equation of the precipitation model in the **ACT.FULL** model is to be solved by the analytical method. **Default:** false
C0.NUMER	logical	Specifies that the equation of the precipitation model in the **ACT.FULL** model is to be solved by the numerical method. **Default:** false
C0.HYBRI	logical	Specifies that the equation of the precipitation model in the **ACT.FULL** model is to be solved by the numerical method when a boundary moves or the temperature ramps; otherwise, it is solved by the analytical method. **Default:** true
II.TOL.F	number	The factor to control the tolerance for taking account of the contribution from the lateral distribution of neighbor as-implant profile. It may not be negative and not exceed 1.0. **Units:** none **Default:** 0.1
II.IF.DY	number	The thickness of the transition region between profiles at material interfaces for calculation of lateral scattering. **Units:** microns **Default:** the current value; initially 1e-4
TABLE	character	Sets the default value of the file name suffix for the implant tables which contain the desired implant data in the format of ion_in_material_suffix. This parameter changes the default suffix for all materials. The **DATA.SUF** parameter of the **IMPLANT** command can be used to change the suffix for a single process step. Applies only to the Taurus Analytic Implant Model. **Default**: **STANDARD**

Parameter	Type	Definition
DAM.TABL	character	Sets the default value of the file name suffix for the implant tables which contain the desired implant damage data in the format of ion_damage_in_material_suffix. This parameter changes the default suffix for all materials. The **DAM.SUF** parameter of the **IMPLANT** command can be used to change the suffix for a single process step. Applies only to the Taurus Analytic Implant Model. **Default**: **STANDARD**
BACKSCAT	logical	Enables or disables backscattering by default for Taurus implants. The value of this parameter is overridden by the value of the **BACKSCAT** parameter of the **IMPLANT** command (if specified). Applies only to the Taurus Analytic Implant Model. **Default:** true
RANGE.SH	logical	Selects the default approach for shifting the projected ranges in multilayer targets. The value of this parameter is overridden by the value of the **RANGE.SH** parameter of the **IMPLANT** command (if specified). Applies only to the Taurus Analytic Implant Model. **Default:** true
D.SCALE	number	The default scale factor applied to the Frenkel pair component of the damage profile. Applies only to the Taurus Analytic Implant Model. **Units:** none **Default:** 1.0
TRAJ.REP	logical	Global switch for trajectory replication. **Default:** false
X.INTERV	number	The length of each segment of the replication subwindow. **Unit:** microns **Default:** 0.02
TRAJ.SPL	logical	Global switch for trajectory splitting. **Default:** false
DEPTH.SP	logical	Depth distribution: Specify the trajectory splitting model which uses the depth distribution of the dopant as splitting criterion. **Default:** true
COLL.DEN	logical	Specify the trajectory splitting model which uses the collision density as splitting criterion. **Default:** false
TRAJ.DEN	logical	Specify the trajectory splitting model which uses the trajectory densityas splitting criterion. **Default:** false
PRINT	logical	Print the current model setting. **Default:** false

Description

The **METHOD** statement selects the models to be used for local oxidation and point defect kinetics, and specifies numerical algorithms to be used for solving the simulation equations. Most users need be concerned only with the parameters for selecting the local oxidation model and the point defect models. Appropriate defaults for all values are given in the *s4init* file.

Oxidation Models

The oxidation model is selected by specifying one of the **ERFC**, **ERF1**, **ERF2**, **ERFG**, **VERTICAL**, **COMPRESS**, **VISCOELA**, or **VISCOUS** parameters. Parameters for the oxidation models are given on the **AMBIENT** statement. See AMBIENT, p. 4-233 for an overview of the models and their parameters, and see Chapter 3, Oxidation, p. 3-84, for a complete description.

Grid Spacing in Growing Oxide

The **DY.OXIDE**, **DY.EXACT**, **DY.LOCAL**, and **GRID.OXI** parameters control the addition of grid to growing oxide layers, as described in Chapter 3, Addition of Nodes in a Growing Layer, p. 3-9. **DY.OXIDE** is the preferred parameter for controlling the grid spacing in growing oxides; the **GRID.OXI** parameter is considered obsolete, and is provided only for compatibility with old input files.

Note:

> ***The default value for GRID.OXI is set every time a METHOD statement is processed; for GRID.OXI to have an effect, it must be specified on the last METHOD statement before an oxidation step.***

Rigid vs. Viscous Substrate

The **SKIP.SIL** parameter determines whether silicon regions are treated as a rigid substrate (**SKIP.SIL** true) or a viscous or viscoelastic material (**SKIP.SIL** false) during oxidation with the **VISCOUS** or **VISCOELA** model. Simulation of oxidation is much faster when **SKIP.SIL** is true, but stresses in the silicon are only calculated when **SKIP.SIL** is false.

To obtain the stresses in the silicon at the end of an oxidation step using the **VISCOUS** model, **SKIP.SIL** can be set to false for a very short oxidation step, and then reset to true:

```
$ Assume SKIP.SIL is true to start with
DIFFUSE TIME=60 TEMPERAT=1000 WETO2

$ Very short step with skip.sil false
$ for stress calculation
METHOD  ^SKIP.SIL
DIFFUSE TIME=1E-6 TEMPERAT=1000 WETO2

$ Reset skip.sil to true for next time
METHOD  SKIP.SIL
```

When the **VISCOELA** model is used, **SKIP.SIL** must be set to false for the entire simulation if stresses in the substrate are needed.

Another case in which **SKIP.SIL** should be set false is when simulating mesa structures, where the lifting of the edge of a silicon mesa may occur. *Note* that the **SKIP.SIL** parameter only affects oxidation with the **VISCOUS** and **VISCOELA** models; silicon is always treated as a rigid substrate when the other oxidation models are used.

Point Defect Modeling

The level of point defect modeling is set by the **NSTREAMS** parameter, and the details of the model are selected by the **PAIR.GRA**, **PAIR.SAT**, **PAIR.REC**, **PD.PFLUX**, **PD.PTIME**, and **PD.PREC** parameters. The **PD.FERMI**, **PD.TRANS**, and **PD.FULL** parameters provide convenient ways of setting the most useful combinations of **NSTREAMS**, **PAIR.GRA**, **PAIR.SAT**, **PAIR.REC**, **PD.PFLUX**, **PD.PTIME**, and **PD.PREC**.

PD.FERMI Model

The simplest (and fastest) model is **PD.FERMI**, in which the point defect concentrations depend only on the Fermi level in the silicon. (The actual point defect concentrations are not calculated, the effects of the Fermi level being included implicitly in the models of impurity diffusion.) The **PD.FERMI** model does not model oxidation-enhanced diffusion, high concentration, or implant damage effects. This model should be used when speed of the simulation is more important than accuracy, or when it is known that the features of the more complicated models are not needed.

PD.TRANS Model

The **PD.TRANS** parameter causes a full, transient simulation of the 2D point defect distributions to be performed. The model includes the generation of point defects at interfaces, the diffusion of point defects into the substrate, and recombination at interfaces and in bulk silicon. The **PD.TRANS** model simulates oxidation-enhanced diffusion, but does not model high concentration effects (e.g., phosphorus kink and tail). It is less accurate than the **PD.FULL** model when simulating implant damage effects. The parameters for the point defect models are specified on the **INTERSTITIAL** and **VACANCY** statements. This model is recommended for routine simulations.

PD.FULL Model

The **PD.FULL** model includes all the effects of the **PD.TRANS** model plus the effects of dopant diffusion on the point defect concentration. It also includes pair saturation and dopant-assisted recombination effects. The **PD.FULL** model simulates oxidation-enhanced diffusion and high concentration (e.g., phosphorus kink and tail), and implant damage effects (when used with the **DAMAGE** parameter on the **IMPLANT** statement). This model is recommended when it is known that high-concentration or implant damage effects are important.

PD.5STR Model

Turning on **PD.5STR** or setting **NSTREAM** to 5 causes the transient simulation of dopant-defect pairing as well as all the effects of the **PD.FULL** model. This model is the most accurate diffusion model available, but requires the most computer time. This model is recommended only when the maximum available accuracy is needed, or when it is known that very low temperature or very short time process following implants is important. If the solution values of a dopant-defect pair are all zeros, the pair solution is removed. For example, by default, phosphorus atoms diffuse by pairing only with interstitials so that the values of the solution vpair_phosphor are all zeros. Then TSUPREM-4 removes the equation for vpair_phosphorus

```
METHOD PD.5STR
```

Customizing the Point Defect Models

The **PD.FERMI**, **PD.TRANS**, and **PD.FULL** parameters are processed before the other diffusion model specifications, so you can modify these models by simultaneous specification of the **PAIR.GRA**, **PAIR.SAT**, **PAIR.REC**, **PD.PFLUX**, **PD.PTIME**, and **PD.PREC** parameters. For example:

```
METHOD PD.TRANS PD.PFLUX
```

selects the basic 3-stream diffusion model, but includes the dopant-defect pair flux terms in the equations for interstitials and vacancies. This combination is useful because it provides a reasonable approximation to high-concentration effects without the computational overhead of the complete **PD.FULL** model. Similarly, the statement:

```
METHOD PD.FULL ^PAIR.REC
```

removes the complicated (but generally insignificant) dopant-assisted recombination factors from the equations for dopant diffusion.

Enable/Disable User-Specified Models

The **MODEL** and **ENABLE** parameters provides the way to enable or disable the equations specified by users in the **IMPURITY** or **REACTION** statements. For example, the below statement designates that all equations named to `myModel` in the **IMPURITY** or **REACTION** statements must be solved:

```
METHOD MODEL=myModel ENABLE
```

On the other hand, the below statement disables the equations:

```
METHOD MODEL=myModel ^ENABLE
```

Solving Poisson's Equation

By default, the electron concentration is analytically calculated by assuming local charge neutrality, which is reasonably good assumption for the semiconductor thermal process. However, as the temperature of the thermal process decreases, and as the abruptness of the doping profile near a PN junction increases, the charge neutrality assumption results in the incorrect electric field near the junction. To calculate more accurate electric field, Poisson's equation needs to be numerically solved. Turning on **POISSON** parameter in the **METHOD** statement specifies Poisson's equation to be solved.

```
METHOD POISSON
```

The solution name of the potential is **PSI**.

Note:

Note that Poisson's equation is solved only for silicon and polysilicon materials.

Adaptive Gridding

The **IMP.ADAP**, **DIF.ADAP**, **OX.ADAPT**, **ERR.FAC**, and **UNREFINE** parameters provide high-level control over adaptive gridding. **IMP.ADAP** enables or disables adaptive mesh refinement during ion implantation, which ensures that the grid is fine enough to resolve the implanted profile.

Note:

Adaptive grid only works with the analytical implant models; it is not available when the MC implant model is specified.

DIF.ADAP enables or disables adaptive gridding during diffusion. Adaptive mesh refinement during diffusion ensures that accuracy is not degraded when profiles diffuse into regions where the grid is too coarse. If the value of **UNREFINE** is greater than zero, mesh unrefinement will be performed to remove mesh points that are no longer needed to ensure an accurate solution.

Note:

You may want to disable adaptive gridding during diffusion immediately following a MC implant. Adaptive gridding can be turned back on after any statistical noise from the MC implant has been smoothed by the diffusion process.

OX.ADAPT controls adaptive gridding in oxide based on oxidant concentration. Its use can significantly improve the accuracy of oxide shapes, especially when specifying a coarse starting grid or large value of **DY.OXIDE**. On the other hand, there are some situations in which **OX.ADAPT** can produce an unnecessarily large number of grid points. For example, when oxidizing through a poly layer, it may be advisable to disable **OX.ADAPT** between the time when the oxidation first breaks through the poly and the time when the poly layer has been consumed across its entire width. This avoids adding unnecessary extra nodes in the pad oxide under the poly.

The interstitial distribution is necessarily abrupt near amorphous and crystalline boundary after regrowth because the boundary location is determined by comparing implant damage concentration to the **MAX.DAM** parameter value.Then the interstitial concentration is reset to the equilibrium value in the regrowth region. Thus, the interstitial dose so Transient Enhanced Diffusion (TED) is sensitive to the mesh density near the boundary region. By switching on **DAM.ADAP**, the adaptive gridding method based on the phase transition between amorphous and crystalline regions makes denser mesh near the boundary so that it increases simulation accuracy. Unlike the adaptive gridding of other solutions that uses the absolute and relative errors based on the solution values, the adaptive gridding for damage solution is performed on the transformed profile of damage:

The **ABS.ADAP** and **REL.ADAP** parameter values for damage is associated with the value *s* of Equations 4-5, not with the damage concentration itself:

$$s = 1 + \tanh[\mathbf{DAM.GRAD} \times \ln(D_0 / \ \mathbf{MAX.DAMA})] \qquad \text{Equation 4-5}$$

ERR.FAC controls the level of adaptive gridding by scaling the relative error targets specified by **REL.ADAP**. Larger values of **ERR.FAC** allow larger relative errors and produce coarser grids; smaller values of **ERR.FAC** reduce the relative errors, producing finer grids. **UNREFINE** controls the level of mesh unrefinement by scaling the relative error used for refinement. A value of 1.0 gives the same error level for refinement and unrefinement, while a value of 0.0 disables unrefinement.

Fine Control

Fine control over adaptive gridding is provided with the **REL.ADAP**, **ABS.ADAP**, **MIN.SPAC**, and **MAX.SPAC** parameters. **REL.ADAP** and **ABS.ADAP** specify the relative and absolute error targets for the specified solution variables and materials. **MIN.SPAC** specifies the minimum grid spacing produced by adaptive refinement, and **MAX.SPAC** specifies the maximum grid spacing produced by unrefinement. The value of **MIN.SPAC** should be small enough to resolve implanted and diffused impurity profiles, but if it is made too small, simulation times may become very large, especially during oxidation steps. **MAX.SPAC** can be set to zero to disable mesh unrefinement in selected materials.

For further information on the use of adaptive gridding see Chapter 4, Adaptive Gridding, p. 4-224.

Initial Time Step

The **INIT.TIM** parameter specifies the initial time step to be used for diffusion steps. The default value is usually adequate, but some speedup can be obtained by specifying a larger value when appropriate. Smaller values may be advisable following an ion implantation with damage, when growing an initial oxide, or when the structure contains sharply peaked impurity distributions.

Internal Solution Methods

The remaining parameters specify the internal solution methods to be used.

CAUTION

These values should not be changed except by the experienced user. Some of the remaining parameters may be changed or eliminated in future releases of the program, in which case input files that use these parameters need to be modified.

Time Integration

The **TRBDF** method is used for time integration. The **MILNE**, **HYBRID**, and **FORMULA** parameters are provided for compatibility with old input files, but only the **TRBDF** method is supported.

A **TRBDF** integration step consists of a trapezoidal step followed by a backward difference step. A second trapezoidal solution is used to estimate the local truncation error and to determine the size of the next time step.

Note:

> ***The parameters governing the matrix solution algorithms should be changed only if there is a demonstrated need for doing so. The use of inappropriate combinations of values for these parameters may prevent the solution algorithms from converging.***

System Solutions

The equations for each impurity (and for oxidant) form a block. Each block is solved independently, then the system of blocks is solved by an outer iteration method. The **CG** parameter selects a conjugate residual method for the outer iteration method. The **BACK** parameter specifies the maximum number of back vectors to be used in the **CG** method. The number of outer iterations is limited to **BLK.ITLI**; the iteration is terminated at this point, whether convergence has been reached or not. The **GAUSS** method is no longer supported, and may not work in this or future versions of the program.

The advanced direct solver **SPLU** implements sparse LU decomposition with partial pivoting, optimized toward the process simulation applications. The iterative **CG** method is still recommended to use by default, unless you experience convergence problems. In which case we suggest to use the direct **SPLU** solver instead.

Minimum-Fill Reordering

A minimum-fill reordering is performed to reduce the solution time for the impurity blocks if the **MIN.FILL** parameter is true. The **MIN.FREQ** parameter sets a limit on how much the solution matrix may expand before the minimum-fill reordering is repeated. A value of zero forces a reordering whenever the grid structure changes; a value greater than one reduces the time spent reordering equations but increases the time required to solve them. Using **MIN.FILL** is strongly advised.

The algorithm for minimum-fill reordering is selected by the **MF.METH** parameter. **MF.METH=0** specifies a complex algorithm designed to give optimum results when **FULL** factorization with a very large value of **MF.DIST** is used. **MF.METH=1** specifies a simpler algorithm that is much faster, and appears to work at least as well as the older algorithm for most cases encountered in TSUPREM-4. **MF.DIST** specifies the distance between nodes in the structure beyond which fill terms can be ignored when **FULL** fill is used.

Block Solution

The remaining parameters apply to solving the equations for the specified impurity.

Automatic Tune for Optimal Numerical Method

The more complicated the simulation problem, the more difficult it is to set the optimal values of the numerical parameters. Furthermore, you must have proficient knowledge of the numerical algorithm, which is beyond normal process engineering. This feature automatically tunes the parameter values on the fly so users do not need to know the numerical algorithm. This feature also improves the simulation speed. When the Boolean parameter **M.AUTO** is switched on, it automatically tunes the linear solver method to generate the optimal preconditioner, minimum fill reordering, and to control the block iteration limit, i.e. **CG/GCR/SPLU**, **PRECOND**, **MF.METH**, **MF.DIST**, **MF.P.LEV**, **MF.F.LEV**, and **BLK.ITLI**.

Solution Method

The **LU** solution method has been found to be most effective for all impurities, and is much faster than the others for point defects. The **SOR**, **SIP**, and **ICCG** methods are no longer supported, and may not work in this or future versions of the program.

Matrix Structure

The **FULL**, **PART**, and **NONE** parameters specify the degree to which fill-in terms are to be included in the matrix solution. **PART** and **NONE** produce smaller matrices that are faster to decompose, but which may require more iterations. **PART** is useful for impurities. Using **FULL** for point defects is generally fastest, but may generate a large number of floating-point underflow conditions. This can cause a large reduction in speed on computers that use software trapping to process subnormal floating point numbers. On such machines, it is better to use **PART** for point defects. An appropriate value for **MF.DIST** can also alleviate this problem. **NONE** can be used for interstitial traps, and other species that do not diffuse.

SYMMETRY forces the matrix to be treated as symmetric, whether it is or not. The program may fail if this parameter is set true when it should be false.

Matrix Refactoring

The **NEWTON**, **TIME.STE**, and **ERROR** parameters determine how often the matrix is refactored. Using **NEWTON** gives the fastest convergence, but spends the most time refactoring the matrix; using **TIME.STE** causes the matrix to be refactored only at the start of each time step. With **ERROR**, the matrix is refactored as needed. The **TIME.STE** choice has been found to give the fastest solution for most problems.

Discontinuous Temperature Change

To keep track of stress history by turning on the parameter **ST.HIST**, the step-like discontinuous change of temperature is not allowed. Instead, TSUPREM-4 internally solves the equations by assuming the delta time step **TIME.DEL**. Too small **TIME.DEL** value might cause the convergence problem. 1μs ~ 1ns are recommended.

Error Tolerances

The relative and absolute error tolerances for solution of the impurity blocks are given by **`REL.ERR`** and **`ABS.ERR`**, respectively. The relative error tolerance for stress-dependent oxidation solutions with the **`VISCOUS`** model is given by **`OX.REL`**, which should be between 1e-4 and 1e-6. **`CONTIN.M`** selects a continuation method for solving the stress-dependent oxidation equations (**`VISCOUS`** model only). Values from 2 to 7 are recognized, but only Methods 2 and 7 should be used. Method 2 is the method used in versions of TSUPREM-4 before Version 5.1; Method 7 is a new method. Method 7 is more likely to converge than Method 2, but when Method 7 does not converge, its results may be worse.

More Accurate Dose Integration from Lateral Distribution

When dopants coming from lateral distribution of the neighbors are integrated, the calculation time can be saved by checking whether if the dose to be added is smaller than the certain tolerance so that it can be ignored. The parameter **`II.TOL.F`** in the **`METHOD`** statement defines the control factor for the tolerance. It may not be negative and not exceed 1.0. The smaller value of **`II.TOL.F`** means the smaller tolerance so that the calculation is more accurate, but takes more time. By default, it is set to 0.1 from v.2004.09. Setting **`II.TOL.F`** to 1.0 provides the backward compatibility. However, note that the value 1.0 of **`II.TOL.F`** can cause the mesh-sensitive unsymmetrical as-implant profile even in the case of symmetrical tilt/rotation implants.

Sensitivity of Implanted Profiles to Non-planar Interfaces

The implant distributions in the layers at an interface overlap, with a linear transition between the distributions as the interface is crossed. Thus when the horizontal line of integration through a node passes through a material interface, the transition from one distribution to another is made over a distance of **`II.IF.DY`**/tan(θ), where θ is the angle of the interface. Since tan(θ)=dY/dX (dY and dX being the Y and X distances between two points on the interface), the transition occurs over a horizontal distance of **`II.IF.DY`**·dX/dY. For a nonplanarity of magnitude dY, the effect in the horizontal direction is spread out by the factor **`II.IF.DY`**/dY. The transition region between profiles in different materials is either placed entirely in one of the materials or centered on the interface. If the material of the node matches that of one of the materials at the interface, then the transition region is placed in the other material. This means that for an exactly planar interface, there is no difference in the implanted profile; you still get an ideal discontinuity at the interface. If the interface is at the top or bottom of the structure, the transition region is placed outside the structure. If the material at the node and the materials at the interface are all different, the transition region is centered on the interface.

Examples

1. The following statement specifies that the initial time step should be 0.1 minutes:

   ```
   METHOD INIT.TIM=0.1  PD.FERMI  VERTICAL
   ```

   The **`PD.FERMI`** model is used for the point defects, and the **`VERTICAL`** model for the oxide growth.

2. The following statement indicates that the arsenic blocks should be solved to a relative error of 0.001 (0.1%) or an absolute error of $10^{12}/\text{cm}^3$, whichever is greater:

   ```
   METHOD ARSENIC REL.ERR=0.001 ABS.ERR=1.0e12
   ```

EQUATION

The **`EQUATION`** statement is used to specify user-specified equations.

```
EQUATION

    VARIABLE=<c> MATERIAL=<c> [MODEL=<c>] [TIF.NAME=<c>]
    [ADDTOEXP=<c>] [INITIAL=<c>] [IMPL.INI=<c>]
    [UPDATE=<c>]
    [/MATERIA=<c> INTERFAC=<c> [BOUNDARY] [TRANSPOR]]
```

Parameter	Type	Definition
`VARIABLE`	character	The solution variable name to be solved. **Default:** none **Synonyms: `IMPURITY`**
`MATERIAL`	character	The name of the material in which the equations are to be solved. **Default:** none **Synonyms: `MAT`**, **`MAT1`**
`MODEL`	character	The identifier of the specified equations. The **`MODEL`** name can be used in the **`METHOD`** statement in order to enable or disable the application of the equations. **Default:** none **Synonym**: **`NAME`**
`TIF.NAME`	character	The name by which the solution variable **`VARIABLE`** is known in the TIF materials database. ***Note*****:** The value of this parameter is case-sensitive. **Default:** the name of the solution variable
`ADDTOEXP`	character	The expression to be added to the right-hand-side (RHS) of the equation for the solution variable **`VARIABLE`**. **Default:** none **Synonym**: **`ADD.RHS`**
`INITIAL`	character	The expression for the initial values of the solution variable **`VARIABLE`**. The expression is evaluated at the beginning of the diffusion step. **Default:** none **Synonym**: **`DIFF.INI`**
`IMPL.INI`	character	The expression for the initial values of the solution variable **`VARIABLE`**. The expression is evaluated at an implantation step. **Default:** none
`UPDATE`	character	The expression which describes the update of the solution variable during the solving of the equation. **Default:** none
`/MATERIA`	character	The name of the material adjacent to **`MATERIAL`** for interface specification. **Default:** none **Synonym**: **`/MAT2`**

Parameter	Type	Definition
INTERFAC	character	The expression for the flux from **/MATERIA** to **MATERIAL**. **Default:** none
BOUNDARY	logical	Specifies that the **ADDTOEXP** expression describes the interface equation. **Default:** false
TRANSPOR	logical	Specifies that the **INTERFAC** flux is taken into account on the nodes in **/MATERIA** as well. **Default:** true

Description

The **EQUATION** statement specifies model equations that have the general form:

$$\frac{\partial X}{\partial t} = \mathit{Previous\_RHS} + \texttt{ADDTOEXP} + \texttt{INTERFAC} \text{ at specified boundaries}$$

where *X* is the quantity to be solved.

TSUPREM-4 assembles the matrices to solve equations by assuming that the time derivative term of *X* is placed at the left-hand-side (LHS) of the equation and all the other terms at the right-hand-side (RHS). To describe this equation using USEIT, any new terms that appear on the RHS of the equation are specified by the **ADDTOEXP** (alias **ADD.RHS**) parameter and the **INTERFAC** parameter on the **EQUATION** statement. The **INTERFAC** parameter is used to define the flux at interfaces. The term *Previous_RHS* in the above expression represents terms that are already present on the RHS of the equation being modified. The initial condition of the variable *X* is specified with the **INITIAL** parameter in the **EQUATION** statement. The **INTERMED** statement specifies the intermediate quantities associated with equations.

Since USEIT is designed as a part of TSUPREM-4, you can easily use the physical quantities such as **CI.STAR**(=I^*) and **ETA**(n/n_i) already calculated in the built-in routines. Similarly, you have the choice of using the built-in functions for calculating active and mobile concentrations or defining your own functions with the **ACTIVE** and **MOBILE** parameters in the **IMPURITY** statement.

When a new solution variable is first specified in the **EQUATION** statement, it is defined automatically. A message is displayed warning that the **VARIABLE** does not currently exist and is being defined. Alternatively, the solution variable can be defined in the **IMPURITY** statement with the **NEW** parameter. The **METHOD** statement specifies the numerical methods used to solve the equations. The factorization method of block matrices, the absolute, and the relative error tolerances for solutions should be specified.

Initialization of Solution

You can define the expression for solution initialization by the **INITIAL** (alias: **DIFF.INI**) parameter. The **INITIAL** expression is performed to initialize a solution at the end of the amorphous regrowth in the first diffusion following an implant or at the beginning of an epitaxy step.

The parameter **`IMPL.INI`** defines the expression for the solution initialization at implant. The **`IMPL.INI`** expression is evaluated and applied at every implant step. .

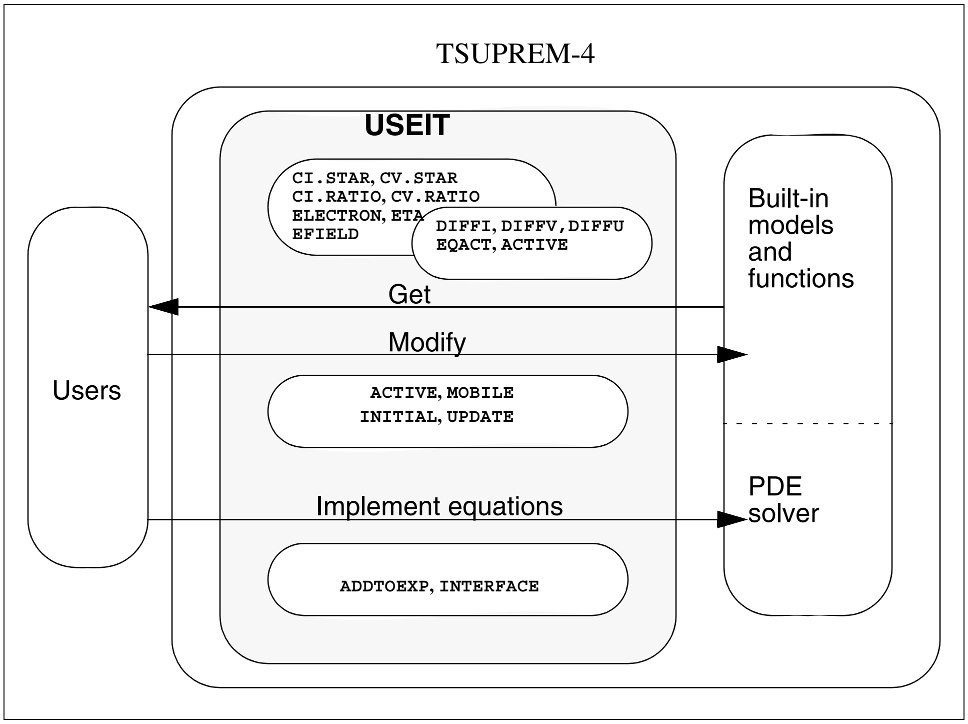


Figure 4-3 USEIT working diagram

See Also

For additional information see the following sections:

- INTERMEDIATE, p. 4-30 for definitions of **`INTERMEDIATES`** used in the expressions of equation.
- METHOD, p. 4-207 for description of numerical methods to solve equations.
- IMPURITY, p. 4-273 for descriptions of calculation methods for active and mobile concentrations.
- Chapter 7, User-Specified Equation Interface for additional details.

AMBIENT

The **`AMBIENT`** statement is used to specify oxidation coefficients. **`OXIDE`** is a valid synonym for the **`AMBIENT`** statement.

```
AMBIENT
   [ {  DRYO2 | WETO2 | STEAM | N2O | INERT  | AMB.1 | AMB.2 | AMB.3 | AMB.4
      | AMB.5}
     [F.O2=<n>] [F.H2O=<n>] [F.N2O=<n>] [F.H2=<n>] [F.N2=<n>]
     [F.HCL=<n>][PRESSURE=<n>] [HCL=<n>]
   ]
   [ {O2 | H2O | N2O }
     [ {<111> | <110> | <100> | ORIENTAT=<n> | POLYSILI}
       [THINOX.0=<n>] [THINOX.E=<n>] [THINOX.L=<n>]
       [L.LIN.0=<n>] [L.LIN.E=<n>] [H.LIN.0=<n>] [H.LIN.E=<n>]
       [N.0=<n>] [N.E=<n>] [N.THIN.0=<n>] [N.THIN.E=<n>]
     ]
     [L.PAR.0=<n>] [L.PAR.E=<n>] [H.PAR.0=<n>] [H.PAR.E=<n>]
     [LIN.BREA=<n>] [PAR.BREA=<n>] [LIN.PDEP=<n>] [PAR.PDEP=<n>]
     [GAMMA.0=<n>] [GAMMA.E=<n>] [GB.FAC.0=<n>] [GB.FAC.E=<n>]
     [ {  LIN.PCT | PAR.PCT
        | ( {LIN.CLDE | PAR.CLDE} COLUMN=<n> )
       }
       TABLE=<c>
     ]
     [ { MATERIAL=<c> | SILICON | OXIDE | OXYNITRI | NITRIDE | POLYSILI
        | AMBIENT}
       [D.0=<n>] [D.E=<n>] [VC=<c>]
       [MODEL=<c>] [D.FACTOR=<c>] [R.FACTOR=<c>]
       [HENRY.CO=<n>] [THETA=<n>]
       [ { /MATERIA=<c> | /SILICON | /OXIDE | /OXYNITR
          | /NITRIDE | /POLYSIL | /AMBIENT
         }
         [SEG.0=<n>] [SEG.E=<n>] [TRANS.0=<n>] [TRANS.E=<n>]
         [ALPHA=<n>]
         [N.G.0=<n>] [N.G.E=<n>] [N.GPOW.0=<n>] [N.GPOW.E=<n>]
         [N.V.0=<n>] [N.V.E=<n>]
       ]
     ]
   ]
   [STRESS.D] [ST.R.ANI] [VR=<c>] [VT=<c>] [VD=<c>] [VDLIM=<n>]
   [VR100=<c>] [VR110=<c>] [VR111=<c>]
   [VT100=<c>] [VT110=<c>] [VT111=<c>]
   [INITIAL=<n>] [SPREAD=<n>] [MASK.EDG=<n>]
   [ERF.Q=<n>] [ERF.DELT=<n>] [ERF.LBB=<c>] [ERF.H=<c>]
   [NIT.THIC=<n>]
   [CLEAR]
   [TEMPERAT=<c>]
   [CM.SEC]
```

Parameter	Type	Definition
DRYO2	logical	Specifies that the pressure, chlorine percentage, and flows of oxidizing and non-oxidizing species are associated with the dry oxygen ambient. **Default:** false
WETO2	logical	Specifies that the pressure, chlorine percentage, and flows of oxidizing and non-oxidizing species are associated with the wet oxygen ambient. **Default:** false
STEAM	logical	Specifies that the pressure, chlorine percentage, and flows of oxidizing and non-oxidizing species are associated with the steam ambient. **Default:** false
N2O	logical	Specifies that the pressure, chlorine percentage, and flows of oxidizing and non-oxidizing species are associated with the N_2O oxygen ambient. **Default:** false
INERT	logical	Specifies that the pressure, chlorine percentage, and flows of oxidizing and non-oxidizing species are associated with the inert ambient. **Default:** false **Synonyms: NEUTRAL**, **NITROGEN**
AMB.1	logical	Specifies that the pressure, chlorine percentage, and flows of oxidizing and non-oxidizing species are associated with ambient number 1. Ambient number 1 is defined by the user. **Default:** false
AMB.2	logical	Specifies that the pressure, chlorine percentage, and flows of oxidizing and non-oxidizing species are associated with ambient number 2. Ambient number 2 is defined by the user. **Default:** false
AMB.3	logical	Specifies that the pressure, chlorine percentage, and flows of oxidizing and non-oxidizing species are associated with ambient number 3. Ambient number 3 is defined by the user. **Default:** false
AMB.4	logical	Specifies that the pressure, chlorine percentage, and flows of oxidizing and non-oxidizing species are associated with ambient number 4. Ambient number 4 is defined by the user. **Default:** false
AMB.5	logical	Specifies that the pressure, chlorine percentage, and flows of oxidizing and non-oxidizing species are associated with ambient number 5. Ambient number 5 is defined by the user. **Default:** false
F.O2	number	The flow of O_2 associated with the specified ambient. If H_2 is also present, the O_2 and H_2 are assumed to react completely to form H_2O. The flows of O_2 and H_2 are reduced, and the flow of H_2O is increased. **Units:** none **Default:** 0.0

Parameter	Type	Definition
`F.H2O`	number	The flow of H_2O associated with the specified ambient. If O_2 and H_2 are also present, the O_2 and H_2 are assumed to react completely to form H_2O. The flows of O_2 and H_2 are reduced, and the flow of H_2O is increased. **Units:** none **Default:** 0.0
`F.N2O`	number	The flow of N_2O associated with the specified ambient. If H_2O or O_2 are also present, the flow of N_2O gas is assumed to be N_2. **Units:** none **Default:** 0.0
`F.H2`	number	The flow of H_2 associated with the specified ambient. If O_2 is also present, the O_2 and H_2 are assumed to react completely to form H_2O. The flows of O_2 and H_2 are reduced, and the flow of H_2O is increased. **Units:** none **Default:** 0.0
`F.N2`	number	The flow of N2 (or other inert gasses) associated with the specified ambient. **Units:** none **Default:** 0.0
`F.HCL`	number	The flow of chlorine associated with the specified ambient. **Units:** none **Default:** 0.0
`PRESSURE`	number	The default value of total gas pressure for the specified ambient. **Units:** atmospheres **Default:** the current value for this ambient; initially 1.0
`HCL`	number	The default percentage of chlorine present for the specified ambient. **Units:** percent **Default:** calculated from **`F.HCL`**
`O2`	logical	Specifies that the oxidation coefficients are associated with the O_2 oxidizing species. **Default:** false
`H2O`	logical	Specifies that the oxidation coefficients are associated with the H_2O oxidizing species. **Default:** false
`<111>`	logical	Specifies that linear and thin oxide growth rate coefficients apply to <111> orientation silicon. **Default:** false
`<110>`	logical	Specifies that linear and thin oxide growth rate coefficients apply to <110> orientation silicon. **Default:** false
`<100>`	logical	Specifies that linear and thin oxide growth rate coefficients apply to <100> orientation silicon. **Default:** false

Parameter	Type	Definition
`ORIENTAT`	number	Specifies that linear and thin oxide growth rate coefficients apply to silicon of the specified orientation. Allowed values are 100, 110, and 111. **Units:** none **Default:** 100
`POLYSILI`	logical	The specified coefficients apply to polysilicon, or the interface between polysilicon and some other material. **Default:** false
`THINOX.0`	number	The pre-exponential constant in the expression for the thin oxide growth rate parameter. **Units:** microns/min or cm/sec **Default:** current value for this orientation and oxidant
`THINOX.E`	number	The activation energy in the expression for the thin oxide growth rate parameter. **Units:** electron volts **Default:** current value for this orientation and oxidant
`THINOX.L`	number	The characteristic length in the expression for the thin oxide growth rate parameter. **Units:** microns **Default:** current value for this orientation and oxidant
`L.LIN.0`	number	The pre-exponential constant in the expression for the linear oxidation rate for temperatures below the temperature breakpoint set by **`LIN.BREA`**. **Units:** microns/min or cm/sec **Default:** current value for this orientation and oxidant **Synonym: `LIN.L.0`**
`L.LIN.E`	number	The activation energy in the expression for the linear oxidation rate for temperatures below the temperature breakpoint set by **`LIN.BREA`**. **Units:** electron volts **Default:** current value for this orientation and oxidant **Synonym: `LIN.L.E`**
`H.LIN.0`	number	The pre-exponential constant in the expression for the linear oxidation rate for temperatures above the temperature breakpoint set by **`LIN.BREA`**. **Units:** microns/min or cm/sec **Default:** current value for this orientation and oxidant **Synonym: `LIN.H.0`**
`H.LIN.E`	number	The activation energy in the expression for the linear oxidation rate for temperatures above the temperature breakpoint set by **`LIN.BREA`**. **Units:** electron volts **Default:** current value for this orientation and oxidant **Synonym: `LIN.H.E`**
`N.0`	number	The pre-exponential constant in the expression for the maximum trap density in the N_2O linear oxidation rate. **Units:** cm^{-2} **Default:** current value for this orientation

Parameter	Type	Definition
`N.E`	number	The activation energy in the expression for the maximum trap density in the N_2O linear oxidation rate **Units:** electron volts **Default:** current value for this orientation
`N.THIN.0`	number	The pre-exponential constant in the expression for the maximum trap density in the N_2O thin oxidation rate. **Units:** cm^{-2} **Default:** current value for this orientation
`N.THIN.E`	number	The activation energy in the expression for the maximum trap density in the N_2O thin oxidation rate **Units:** electron volts **Default:** current value for this orientation
`L.PAR.0`	number	The pre-exponential constant in the expression for the parabolic oxidation rate for temperatures below the temperature breakpoint set by **`PAR.BREA`**. **Units:** $microns^2$/min or cm^2/sec **Default:** current value for this oxidant **Synonym:** **`PAR.L.0`**
`L.PAR.E`	number	The activation energy in the expression for the parabolic oxidation rate for temperatures below the temperature breakpoint set by **`PAR.BREA`**. **Units:** electron volts **Default:** current value for this oxidant **Synonym:** **`PAR.L.E`**
`H.PAR.0`	number	The pre-exponential constant in the expression for the parabolic oxidation rate for temperatures above the temperature breakpoint set by **`PAR.BREA`**. **Units:** $microns^2$/min or cm^2/sec **Default:** current value for this oxidant **Synonym:** **`PAR.H.0`**
`H.PAR.E`	number	The activation energy in the expression for the parabolic oxidation rate for temperatures above the temperature breakpoint set by **`PAR.BREA`**. **Units:** electron volts **Default:** current value for this oxidant **Synonym:** **`PAR.H.E`**
`LIN.BREA`	number	The temperature breakpoint at which the temperature dependence of the linear oxidation rate changes. **Units:** degrees Celsius **Default:** current value for this oxidant **Synonym:** **`L.BREAK`**
`PAR.BREA`	number	The temperature breakpoint at which the temperature dependence of the parabolic oxidation rate changes. **Units:** degrees Celsius **Default:** current value for this oxidant **Synonym:** **`P.BREAK`**

Parameter	Type	Definition
`LIN.PDEP`	number	The exponent of the pressure in the expression for the linear oxidation rate. **Units:** none **Default:** current value for this oxidant **Synonym:** **`L.PDEP`**
`PAR.PDEP`	number	The exponent of the pressure in the expression for the parabolic oxidation rate. **Units:** none **Default:** current value for this oxidant **Synonym:** **`P.PDEP`**
`GAMMA.0`	number	The pre-exponential constant in the expression for the impurity concentration dependence of the linear oxidation rate. **Units:** none **Default:** current value for this oxidant
`GAMMA.E`	number	The activation energy in the expression for the impurity concentration dependence of the linear oxidation rate. **Units:** electron volts **Default:** current value for this oxidant
`GB.FAC.0`	number	The pre-exponential constant in the expression for the enhancement of the surface reaction rate at grain boundaries. **Units:** none **Default:** current value for this oxidant; initially 10 for all ambients
`GB.FAC.E`	number	The activation energy in the expression for the enhancement of the surface reaction rate at grain boundaries. **Units:** electron volts **Default:** current value for this oxidant; initially 0 for all ambients
`LIN.PCT`	logical	Specifies that the **`TABLE`** parameter defines chlorine percentages associated with the rows in the table of coefficients modifying the linear oxidation rate in the presence of chlorine. **Default:** false
`PAR.PCT`	logical	Specifies that the **`TABLE`** parameter defines chlorine percentages associated with the rows in the table of coefficients modifying the parabolic oxidation rate in the presence of chlorine. **Default:** false
`LIN.CLDE`	logical	Specifies that the **`TABLE`** parameter defines entries in a column of the table of coefficients modifying the linear oxidation rate in the presence of chlorine. The column number is specified with the **`COLUMN`** parameter, and is associated with the temperature specified by the **`TEMPERAT`** parameter. **Default:** false
`PAR.CLDE`	logical	Specifies that the **`TABLE`** parameter defines entries in a column of the table of coefficients modifying the parabolic oxidation rate in the presence of chlorine. The column number is specified with the **`COLUMN`** parameter, and is associated with the temperature specified by the **`TEMPERAT`** parameter. **Default:** false

Parameter	Type	Definition
`COLUMN`	number	The column number in the tables of coefficients modifying the linear or parabolic oxidation rates. The coefficients defined by the **`TABLE`** parameter are stored in this column of a table. The column number is associated with the temperature specified by the **`TEMPERAT`** parameter. There can be, at most, 8 columns in a table, and each column corresponds to one temperature. **Units:** none **Default:** none
`TABLE`	character	This parameter is interpreted as a series of numeric values, separated by spaces or commas. If the **`LIN.PCT`** or **`PAR.PCT`** parameter is specified, the **`TABLE`** parameter defines the chlorine percentages associated with the rows in the table of coefficients modifying the linear or parabolic oxidation rates, respectively. If the **`LIN.CLDE`** or **`PAR.CLDE`** parameter is specified, the **`TABLE`** parameter defines the entries in a column of the table of coefficients modifying the linear or parabolic oxidation rates, respectively. There can be, at most, eight rows in a table, and each row corresponds to one chlorine percentage. At most, 8 values can be defined with this parameter. **Units:** percent or none **Default:** none
`MATERIAL`	character	The specified coefficients apply to the named material or to the interface between the named material and some other material. **Default:** none
`SILICON`	logical	The specified coefficients apply to silicon, or the interface between silicon and some other material. **Default:** false
`OXIDE`	logical	The specified coefficients to apply to oxide, or the interface between oxide and some other material. **Default:** true if no other first material is specified
`OXYNITRI`	logical	The specified coefficients to apply to oxynitride, or the interface between oxynitride and some other material. **Default:** false
`NITRIDE`	logical	The specified coefficients to apply to nitride, or the interface between nitride and some other material. **Default:** false
`POLYSILI`	logical	The specified coefficients to apply to polysilicon, or the interface between polysilicon and some other material. **Default:** false
`AMBIENT`	logical	The specified coefficients apply to gas, or the interface between gas and some other material. **Default:** false **Synonym:** **`GAS`**

Parameter	Type	Definition
`D.0`	number	The pre-exponential constant in the expression for the diffusivity of oxidant in the specified material. **Units:** microns^2/min or cm^2/sec **Default:** current value for this oxidant and material **Synonym:** **`DIFF.0`**
`D.E`	number	The activation energy in the expression for the diffusion of oxidant in the specified material. **Units:** electron volts **Default:** current value for this oxidant and material **Synonym:** **`DIFF.E`**
`VC`	character	A table of activation volumes as a function of temperature for the dependence of material viscosity on shear stress for the specified material and oxidizing species (**`O2`** or **`H2O`**). Entries in the table correspond to temperatures given by the **`TEMPERAT`** parameter (see text). **Units:** Å^3 **Default:** the current value; initially 300
`MODEL`	character	The name of the model for user-defined multiplication factors **`D.FACTOR`** and **`R.FACTOR`**. Only applicable to **`D.FACTOR`** and **`R.FACTOR`**. **Default:** none
`D.FACTOR`	character	The expression of the multiplication factor to oxidant diffusivity. Only applicable to the **`VISCOEL`** model. **Units:** none **Default:** "1.0"
`R.FACTOR`	character	The expression of the multiplication factor to oxidant surface reaction rate. Only applicable to the **`VISCOEL`** model. **Units:** none **Default:** "1.0"
`HENRY.CO`	number	The solubility of oxidant in the specified material at one atmosphere. **Units:** atoms/cm^3/atm **Default:** current value for this oxidant and material
`THETA`	number	The number of oxide molecules per cubic centimeter of oxide. **Units:** atoms/cm^3 **Default:** current value
`/MATERIA`	character	The specified coefficients to apply to the interface between the other specified material and this named material. **Default:** none
`/SILICON`	logical	The specified coefficients to apply to the interface between the specified material and silicon. **Default:** True if no other second material is specified.
`/OXIDE`	logical	The specified coefficients to apply to the interface between the specified material and oxide. **Default:** false

Parameter	Type	Definition
`/OXYNITR`	logical	The specified coefficients to apply to the interface between the specified material and oxynitride. **Default:** false
`/NITRIDE`	logical	The specified coefficients to apply to the interface between the specified material and nitride. **Default:** false
`/POLYSIL`	logical	The specified coefficients to apply to the interface between the specified material and polysilicon. **Default:** false
`/AMBIENT`	logical	The specified coefficients to apply to the interface between the specified material and gas. **Default:** false **Synonym: `/GAS`**
`SEG.0`	number	The pre-exponential constant in the expression for segregation of oxidant between the two specified materials. **Units:** none **Default:** current value for this oxidant and these materials
`SEG.E`	number	The activation energy in the expression for segregation of oxidant between the two specified materials. **Units:** electron volts **Default:** current value for this oxidant and these materials
`TRANS.0`	number	The pre-exponential constant in the expression for transport of oxidant between the two specified materials. **Units:** microns/min or cm/sec **Default:** the current value for this oxidant and these materials **Synonym: `TRN.0`**
`TRANS.E`	number	The activation energy in the expression for transport of oxidant between the two specified materials. **Units:** electron volts **Default:** the current value for this oxidant and these materials **Synonym: `TRN.E`**
`ALPHA`	number	The volume expansion ratio between the two specified materials. The defaults are 0.44 for silicon/oxide and polysilicon/oxide, and 1.0 for all other combinations. **Units:** none **Default:** the current value for these materials
`N.G.0`	number	The pre-exponential constant in the expression for nitrogen generation rate due to N_2O oxidation. **Units:** cm^{-3} **Default:** the current value for these materials

Parameter	Type	Definition
N.G.E	number	The activation energy in the expression for nitrogen generation rate due to N_2O oxidation. **Units:** electron volts **Default:** the current value for these materials
N.GPOW.0	number	The pre-exponential constant in the expression for the exponent in the normalized boundary velocity. **Units:** none **Default:** the current value for these materials
N.GPOW.E	number	The activation energy in the expression for the exponent in the normalized boundary velocity of N_2O oxidation. **Units:** electron volts **Default:** the current value for these materials
N.V.0	number	The pre-exponential constant in the expression for the normalization velocity of N_2O oxidation. **Units:** cm/sec **Default:** the current value for these materials
N.V.E	number	The activation energy in the expression for the normalization velocity of N_2O oxidation. **Units:** none **Default:** the current value for these materials
STRESS.D	logical	Specifies that the stress-dependent models for oxide viscosity, oxidant diffusivity, and surface reaction rate are to be used. **Default:** the current value, initially true
ST.R.ANI	logical	Specifies that the anisotropic stress dependent reaction rate model for oxidation is to be used. **Default:** the current value, initially false
VR	character	A table of activation volumes as a function of temperature for the dependence of oxidation rate at the Si/SiO_2 interface on normal stress for the specified oxidizing species (**O2** or **H2O**). Entries in the table correspond to temperatures given by the **TEMPERAT** parameter (see text). **Units:** $Å^3$ **Default:** the current value; initially 15
VR100	character	A table of activation volumes for crystal direction 100 as a function of temperature for the dependence of oxidation rate at the Si/SiO_2 interface on normal stress for the specified oxidizing species (**O2** or **H2O**). Entries in the table correspond to temperatures given by the **TEMPERAT** parameter (see text). **Units:** $Å^3$ **Default:** the current value; initially 15

Parameter	Type	Definition
`VR110`	character	A table of activation volumes for crystal direction 110 as a function of temperature for the dependence of oxidation rate at the Si/SiO$_2$ interface on normal stress for the specified oxidizing species (**`O2`** or **`H2O`**). Entries in the table correspond to temperatures given by the **`TEMPERAT`** parameter (see text). **Units:** Å^3 **Default:** the current value; initially 15
`VR111`	character	A table of activation volumes for crystal direction 111 as a function of temperature for the dependence of oxidation rate at the Si/SiO$_2$ interface on normal stress for the specified oxidizing species (**`O2`** or **`H2O`**). Entries in the table correspond to temperatures given by the **`TEMPERAT`** parameter (see text). **Units:** Å^3 **Default:** the current value; initially 150
`VT`	character	A table of activation volumes as a function of temperature for the dependence of oxidation rate at the Si/SiO$_2$ interface on tangential stress for the specified oxidizing species (**`O2`** or **`H2O`**). Entries in the table correspond to temperatures given by the **`TEMPERAT`** parameter (see text). **Units:** Å^3 **Default:** the current value; initially 0.0
`VT100`	character	A table of activation volumes for crystal direction 100 as a function of temperature for the dependence of oxidation rate at the Si/SiO2 interface on tangential stress for the specified oxidizing species (O2 or H2O). Entries in the table correspond to temperatures given by the TEMPERAT parameter (see text). **Units**: Å3 **Default**: the current value; initially 0.0
`VT110`	character	A table of activation volumes for crystal direction 110 as a function of temperature for the dependence of oxidation rate at the Si/SiO2 interface on tangential stress for the specified oxidizing species (O2 or H2O). Entries in the table correspond to temperatures given by the TEMPERAT parameter (see text). **Units**: Å3 **Default**: the current value; initially 0.0
`VT111`	character	A table of activation volumes for crystal direction 111 as a function of temperature for the dependence of oxidation rate at the Si/SiO2 interface on tangential stress for the specified oxidizing species (O2 or H2O). Entries in the table correspond to temperatures given by the TEMPERAT parameter (see text). **Units**: Å3 **Default**: the current value; initially 0.0
`VD`	character	A table of activation volumes as a function of temperature for the dependence of oxidant diffusivity in oxide on pressure for the specified oxidizing species (**`O2`** or **`H2O`**). Entries in the table correspond to temperatures given by the **`TEMPERAT`** parameter (see text). **Units:** Å^3 **Default:** the current value; initially 75

Parameter	Type	Definition
`VDLIM`	number	The maximum increase in oxidant diffusivity produced by the **`VD`** parameter. **Units:** none **Default:** the current value; initially 1.2
`INITIAL`	number	The thickness of the existing oxide at the start of oxidation. Exposed silicon surfaces are covered with this thickness of native oxide before oxidation begins. **Units:** microns **Default:** the current value; initially 0.002
`SPREAD`	number	The ratio of width to height for the bird's beak, used in the **`ERFC`** model of local oxide shape. **Units:** none **Default:** the current value; initially 1.0
`MASK.EDG`	number	The assumed position of the mask, used by the analytical models for local oxidation. Oxide grows to the right of the mask edge. **Units:** microns **Default:** the current value; initially –200
`ERF.Q`	number	The *q* parameter for the **`ERFG`** (Guillemot) model. **Units:** microns **Default:** the current value; initially 0.05
`ERF.DELT`	number	The *delta* parameter for the **`ERFG`** (Guillemot) model. **Units:** microns **Default:** the current value; initially 0.04
`ERF.LBB`	character	The length of the bird's beak for the **`ERFG`** (Guillemot) model. This is an arithmetic expression involving the variables *Fox* (the field oxide thickness, in microns), *eox* (the pad oxide thickness, in microns), *Tox* (the oxidation temperature, in degrees Kelvin), and *en* (the nitride thickness, in microns). **Units:** microns **Default:** the current value; initially $(8.25e\text{-}3*(1580.3-Tox)*(Fox^{0.67})*(eox^{0.3})*\exp(-((en-0.08)^{2})/0.06))$.
`ERF.H`	character	The ratio of the nitride lifting to the field oxide thickness for the **`ERFG`** (Guillemot) model. This is an arithmetic expression involving the variables *Fox* (the field oxide thickness, in microns), *eox* (the pad oxide thickness, in microns), *Tox* (the oxidation temperature, in degrees Kelvin), and *en* (the nitride thickness, in microns). **Units:** none **Default:** the current value; initially (402*(0.445-1.75**en*)*exp(-*Tox*/200))
`NIT.THIC`	number	The nitride thickness, *en* used in the equations for **`ERF.LBB`** and **`ERF.H`**. **Units:** microns **Default:** none
`TEMPERAT`	character	The temperature associated with the column in the chlorine tables, given by the **`COLUMN`** parameter, or a list of temperatures corresponding to the values of the **`VC`**, **`VD`**, **`VR`**, and/or **`VT`** parameters. **Units:** degrees Celsius **Default:** none

Parameter	Type	Definition
`CLEAR`	logical	Clear table(s) specified by the `VC`, `VD`, `VR`, and/or `VT` parameters before adding new values (see text). **Default:** none
`CM.SEC`	logical	If true, parameters involving time are specified in centimeters and seconds; if false, parameters involving time are in microns and minutes. **Default:** false

Description

All parameters relating to oxidation are specified on this statement. The necessary parameters are set by `AMBIENT` statements in the *s4init* file, but can be changed by you.

Oxidation Models

The following models are available:

- An error-function fit to bird's beak shapes (the `ERFC` model)
- A parameterized error-function model from the literature (the Guillemot or `ERFG` model)
- A model in which oxidant diffuses and the oxide grows vertically at a rate determined by the local oxidant concentration (the `VERTICAL` model)
- A compressible viscous flow model (the `COMPRESS` model)
- A viscoelastic flow model (the `VISCOELA` model)
- An incompressible viscous flow model (the `VISCOUS` model)

A summary of the features and characteristics of these models follows; full descriptions are given in Chapter 3, Oxidation, p. 3-84.

Note:

> ***Oxidation of polycrystalline silicon is modeled by the `COMPRESS`, `VISCOELA`, and `VISCOUS` models only.***

ERFC Model

The `ERFC` model is the fastest of the oxidation models. It can be used for uniform oxidation of bare silicon, provided that modeling of the concentration dependence of the oxidation rate is not needed. It can be used for nonuniform oxidation of planar surfaces, provided that fitting data for the lateral spread of the bird's beak is available.

The `ERFC` model is controlled by the `SPREAD`, `MASK.EDG`, and `INITIAL` parameters. The growth rate vs. time is computed, assuming an initial oxide thickness `INITIAL` at the start of each diffusion step. This model should not be used with a structure having an unmasked initial oxide thickness other than `INITIAL`.

ERFG Model

The `ERFG` model is by Guillemot, et al., *IEEE Transactions on Electron Devices*, ED-34, May 1987. The bird's beak shape and nitride lifting are functions of process conditions. The `ERFG` model is controlled by the `ERF.Q`, `ERF.DELT`,

ERF.LBB, **ERF.H**, **NIT.THIC**, and **INITIAL** parameters. The above comments regarding **INITIAL** apply the **ERFG** model as well. The **ERF1** and **ERF2** models use the two shapes derived by Guillemot, et al; the **ERFG** model chooses between them based on process conditions.

VERTICAL Model

The **VERTICAL** model has no fitting parameters, but is only accurate when the growth is approximately vertical (within about 30° of vertical). The **VERTICAL** model does not simulate oxidation of polysilicon. The **VERTICAL** model can be used for oxidation of uniform substrates with arbitrary initial oxide thicknesses and for approximating nonrecessed LOCOS processes. Concentration dependence of the oxidation rate is included in the **VERTICAL** model.

COMPRESS Model

The **COMPRESS** model simulates the viscous flow of the oxide in two dimensions. It uses simple (three nodes per triangle) elements for speed, but must allow some compressibility as a consequence. It is more accurate than the **VERTICAL** model, but requires more computer time. It uses Young's modulus (**YOUNG.M**) and Poisson's ratio (**POISS.R**), specified for each material with the **MATERIAL** statement. The **COMPRESS** model is recommended for general use on arbitrary structures. It includes the concentration dependence of oxidation rate and models the oxidation of polysilicon.

VISCOELA Model

The **VISCOELA** model simulates viscoelastic flow in two dimensions. It uses simple (three nodes per triangle) elements for speed, but simulates elastic deformation as well as viscous flow. When used with stress dependent parameters (i.e., **STRESS.D** true), it can produce very accurate results with reasonable simulation times. It is slower than the **COMPRESS** model, but 10-100 times faster than the **VISCOUS** model with stress dependence. It uses the **YOUNG.M**, **POISS.R**, **VISC.0**, **VISC.E**, and **VISC.X** parameters for mechanical properties of materials plus the **VC**, **VR**, **VD**, and **VDLIM** parameters for describing stress dependence.

VISCOUS Model

The **VISCOUS** model simulates incompressible viscous flow of the oxide, using more complicated (seven nodes per triangle) elements. It calculates stresses, and is the only model that models reflow. The **VISCOUS** model is slower than the **COMPRESS**, and **VISCOELA** models, and may require large amounts of memory; it may be impossible to simulate large structures with this model on some computers, due to memory limitations. It uses the viscosity parameters (**VISC.0**, **VISC.E**, and **VISC.X**) specified for each material with the **MATERIAL** statement.

The **VISCOUS** model is needed only when stress calculations are required, when the stress-dependent oxidation parameters are used, or when **SKIP.SIL** must be set false to simulate structures with floating silicon mesas.

Stress Dependence

The parameter **STRESS.D** determines whether the stress dependence of oxidant diffusivity, surface reaction rate, and oxide viscosity are included when oxidizing with the **VISCOELA** or **VISCOUS** models. When **STRESS.D** is true, these stress dependencies are included; when **STRESS.D** is false, they are not.

The magnitude of the various stress effects are specified by the **VC**, **VR**, **VT**, **VD**, and **VDLIM** parameters.

- **VC** is the activation volume for the dependence of oxide viscosity on shear stress. **VC** can be specified for arbitrary materials using the **MATERIAL** statement.
- **VR** and **VT** are the activation volumes for the dependence of the surface reaction rate on normal and tangential stresses, respectively.
- **VD** is the activation volume for the dependence on pressure of the diffusivity of the oxidizing species in the oxide.
- **VDLIM** is the maximum increase in oxidant diffusivity produced by **VD**.

The parameters **VC**, **VD**, **VR**, **VT**, and **TEMPERAT** are used to specify the activation volumes as functions of temperature. A separate table is maintained for each oxidizing species, and for each material in the case of **VC**. Table entries are added or changed by specifying lists of values (with **VC**, **VD**, **VR**, or **VT**) and temperatures (with **TEMPERAT**). The portion of the table spanned by the specified temperatures is replaced by the specified values; the number of values must be the same as the number of temperatures, and the temperatures must be given in order, lowest to highest. The **CLEAR** parameter is used to clear a table before setting any values.

For example, the statement

```
AMBIENT O2 CLEAR VD="40 50 60" TEMP="800 900 1050"
```

removes any old values from the table of **VD** vs. temperature for O2 and adds three new values. The statement

```
AMBIENT O2 VD="55 75" TEMP="900 1100"
```

would then replace the values at 900° C and 1050° C with new values at 900° C and 1100° C.

If no oxidizing species is specified, the values apply to ambients containing either O_2 or H_2O. The material should be specified when setting **VC**; if no material is specified, **OXIDE** is assumed. If **V.COMPAT** (on the **OPTION** statement) is less than 6.6, the specified values of the activation volumes apply to oxide in all ambients, including inert ambients.

An Arrhenius interpolation is used between values in the table. For temperatures outside the range of the table, the nearest value is used.

Anisotropic Stress Dependent Reaction

The parameter **ST.R.ANI**determines whether to use the anisotropic stress dependent reaction model. When the parameter is set true, the surface reaction rate is computed using crystal orientation dependent parameters **VR**,and **VT**, where **VR** is extracted from normal activation volumes in 100, 110 and 111 crystal directions, **VR100**, **VR110**, and **VR111** respectively; and **VT** is extracted from tangential activation volumes in 100, 110 and 111 crystal directions, **VT100**, **VT110**, and **VT111**, respectively.

Coefficients The diffusion and segregation coefficients can be used to model oxidant diffusion in arbitrary layers, but the diffusion coefficient in oxide is derived from the parabolic rate constant. The transport coefficient between the ambient and oxide is interpreted as the gas-phase mass-transport coefficient for the specified oxidizing species.

Chlorine The effects of chlorine in the ambient gas on the oxidation rate of silicon are specified by tables of coefficients that modify the linear and parabolic oxidation rates. There are two tables for each oxidizing species, one each for the linear and parabolic oxidation rates. The tables are 2D with, at most, 8 rows corresponding to chlorine percentages and, at most, 8 columns corresponding to ambient temperatures. Linear interpolation is used to obtain values for temperatures or percentages between the values in the table. For temperatures or percentages outside of the range of values present in the table, the values in the first or last rows or columns, as appropriate, are used.

Examples

For example, consider the following table of chlorine coefficients with six rows of chlorine percentages and five columns of temperatures:

		1	2	3	4	5	column
row	%	800	900	1000	1100	1200	temperature
1	0	1.0	1.0	1.0	1.0	1.0	
2	1	1.1	1.2	1.3	1.4	1.5	
3	3	1.6	1.7	1.8	1.9	2.0	
4	5	2.1	2.2	2.3	2.4	2.5	
5	7	2.6	2.7	2.8	2.9	3.0	
6	10	3.1	3.2	3.3	3.4	3.5	

If this table represented the modification coefficients for the linear oxidation rates for the O2 oxidizing species, it could have been defined with the following series of input statements:

```
AMBIENT  O2  LIN.PCT  TABLE="0, 1, 3, 5, 7, 10"
AMBIENT  O2  LIN.CLDE  COLUMN=1  TEMPERAT=800  +
              TABLE="1.0, 1.1, 1.6, 2.1, 2.6, 3.1"
AMBIENT  O2   LIN.CLDE  COLUMN=2  TEMPERAT=900

                TABLE="1.0, 1.2, 1.7, 2.2, 2.7,
3.2"
AMBIENT  O2  LIN.CLDE  COLUMN=3  TEMPERAT=1000  +
              TABLE="1.0, 1.3, 1.8, 2.3, 2.8, 3.3"
AMBIENT  O2  LIN.CLDE  COLUMN=4  TEMPERAT=1100  +
              TABLE="1.0, 1.4, 1.9, 2.4, 2.9, 3.4"
AMBIENT  O2  LIN.CLDE  COLUMN=5  TEMPERAT=1200  +
              TABLE="1.0, 1.5, 2.0, 2.5, 3.0, 3.5"
```

The following values are obtained from this table for the indicated percentages and temperatures:

percent	temperature	table value	row	column
1.0	1000	1.3	2	3
1.0	1050	1.35	2	3,4
2.0	1000	1.55	2,3	3
12.0	1000	3.3	6	3
1.0	700	1.1	2	1
1.0	1250	1.5	2	5
2.0	1250	1.75	2,3	5

Parameter Dependencies

Parameters which have special dependencies are listed below. If insufficient information is given with a parameter (e.g., **L.LIN.0** without an orientation and an oxidant), the parameter is ignored (without warning).

Orientation

The following parameters are dependent on the specified orientation: **L.LIN.0**, **L.LIN.E**, **H.LIN.0**, **H.LIN.E**, **THINOX.0**, **THINOX.E**, and **THINOX.L**. *Note* that **POLYSILI** can be used in place of the orientation to specify coefficients for oxidation of polysilicon.

Oxidizing Species

The following parameters are dependent on the oxidizing species (O_2 or H_2O): **L.LIN.0**, **L.LIN.E**, **H.LIN.0**, **H.LIN.E**, **LIN.BREA**, **LIN.PDEP**, **L.PAR.0**, **L.PAR.E**, **H.PAR.0**, **H.PAR.E**, **PAR.BREA**, **PAR.PDEP**, **GAMMA.0**, **GAMMA.E**, **LIN.PCT**, **PAR.PCT**, **LIN.CLDE**, **PAR.CLDE**, **COLUMN**, **TEMPERAT**, **TABLE**, **THINOX.0**, **THINOX.E**, **THINOX.L**, **D.0**, **D.E**, **VC**, **HENRY.CO**, **SEG.0**, **SEG.E**, **TRANS.0**, and **TRANS.E**.

VD, **VR**, and **VT** also depend on the oxidizing species, but apply to both O_2 and H_2O, if neither is specified.

Specified Material

The following parameters are dependent only on the first material specified: **D.0**, **D.E**, **VC**, **HENRY.CO**, and **THETA**.

The following parameters are dependent on both materials specified: **SEG.0**, **SEG.E**, **TRANS.0**, **TRANS.E**, and **ALPHA**.

Specified Units

Parameters whose units include time are specified in units of microns and minutes, unless **CM.SEC** is true, in which case units of centimeters and seconds are assumed.

Examples

1. The statement:

```
AMBIENT AMB.1 F.O2=0.90 F.N2=0.08 F.HCL=0.02 +
        PRESSURE=2.0
```

defines ambient **AMB.1** to consist of 90% oxygen and 2% chlorine at a pressure of two atmospheres.

2. The statement:

```
AMBIENT H2O MAT=NITRIDE +
        VC="130 170" TEMP="900 1000"
```

replaces any values for temperatures between 900° C and 1000° C in the table for **VC** of nitride in ambients containing H_2O.

3. The initialization file *s4init* contains the definitive set of examples of use for the **AMBIENT** statement.

Additional AMBIENT Notes

- Oxidant in materials other than oxide is allowed to diffuse and segregate, but its concentration is then ignored (no oxynitridation, for instance). The diffusion coefficients in oxide and transport coefficients between oxide and silicon are derived from the Deal-Grove coefficients, so these parameters are ignored if read from input statements.
- The analytic models use the thickness of the oxide to compute the growth rate, and the **ERFG** model also uses the nitride thickness. These values are *not* inferred from the structure. Instead, the value of **NIT.THIC** is used for the nitride thickness, and the oxide thickness is calculated by adding the oxide grown in a given high-temperature step to the specified **INITIAL** oxide thickness. Thus, if the structure has other than **INITIAL** microns of oxide on it at the start of a diffusion step, the thickness must be specified with the **INITIAL** parameter. If there is no oxide on an exposed silicon surface, a layer of oxide of thickness **INITIAL** is deposited. If the **INITIAL** parameter does not correspond to the actual oxide thickness, the growth rate is incorrect. The **INITIAL** parameter need not be set when an oxidation is continued with the **CONTINUE** parameter on the **DIFFUSION** statement.
- The analytic models do not recognize masking layers in the structure. The location of the presumed mask edge must be specified by the **MASK.EDG** parameter.
- The material viscosities have been calibrated for the **VISCOELA** model, with stress dependence enabled. For use without stress dependence (with either the **VISCOELA** or **VISCOUS** model), it may be necessary to modify the viscosity values.

MOMENT

The **MOMENT** statement sets distribution moments for use in ion implantation.

```
MOMENT

      [ { MATERIAL=<c> | SILICON | OXIDE | OXYNITRI | NITRIDE | POLYSILI
        | PHOTORES | ALL.MATE } ]
      { ( [CLEAR]
          [ { RANGE=<n> SIGMA=<n> [GAMMA=<n>] [KURTOSIS=<n>] [LSIGMA=<n>]
          [LSLOPE=<n>] [ { DAMAGE [LAMBDA=<n>] [SCALE=<n>] } ]
          [ { D.FRAC=<n> D.RANGE=<n> D.SIGMA=<n> [D.GAMMA=<n>]
                [D.KURTOS=<n>] [D.LSIGMA=<n>] [D.LSLOPE=<n>] } ]
           [TILT=<n>] } ]
        )
       |( { IMPURITY=<c> | ANTIMONY | ARSENIC | BORON | BF2 | PHOSPHOR }
          [D.SUP.R=<n>] [D.SUP.E=<n>] [RANGE.F=<n>] [SIGMA.F=<n>]
          [D.RAN.F=<n>] [D.SIG.F=<n>] [TABLE=<c>] [DAM.TABL=<c>]
        )
      }
```

Parameter	Type	Definition
MATERIAL	character	The name of the material to which the given range statistics apply. **Default:** none
SILICON	logical	Specifies that the given range statistics apply to implantation into silicon. **Default:** true, unless another material is specified
OXIDE	logical	Specifies that the given range statistics apply to implantation into oxide. **Default:** false
OXYNITRI	logical	Specifies that the given range statistics apply to implantation into oxynitride. **Default:** false
NITRIDE	logical	Specifies that the given range statistics apply to implantation into nitride. **Default:** false
POLYSILI	logical	Specifies that the given range statistics apply to implantation into polysilicon. **Default:** false
PHOTORES	logical	Specifies that the given range statistics apply to implantation into photoresist. **Default:** false
ALL.MATE	logical	Specifies that the parameter values will be applied to all materials. Applies only to the Taurus Analytic Implant Model. **Default:** false
CLEAR	logical	Clears all moments specified on previous **MOMENT** statements. **Default:** false
RANGE	number	The first moment (projected range) of the first Pearson or Gaussian distribution. **units:** microns **Default:** none

Parameter	Type	Definition
`SIGMA`	number	The second moment (standard deviation) of the first Pearson or Gaussian distribution. **Units:** microns **Default:** none
`GAMMA`	number	The third moment ratio (skewness) of the first Pearson distribution. **Units:** none **Default:** 0.0
`KURTOSIS`	number	The fourth moment ratio (kurtosis) of the first Pearson distribution. **Units:** none **Default:** calculated from the first three moments
`LSIGMA`	number	The lateral (perpendicular to the ion beam) standard deviation for the first Pearson or Gaussian distribution. **Units:** microns **Default: `SIGMA`**
`LSLOPE`	number	The slope of the lateral standard deviation for the first Pearson distribution of the dual-Pearson model. **Units:** microns **Default**: 0.0
`DAMAGE`	logical	Specifies that the remaining parameters on the **`MOMENT`** statement are used for damage calculation. In particular, **`RANGE`** and **`SIGMA`** specify Hobler's **`a2`** and **`a3`** parameters. **`LSIGMA`** and **`LSLOPE`** specify the lateral spread and depth dependence of the lateral spread for the damage model. **Default**: false
`LAMBDA`	number	The slope of the exponential part of the damage profile (Hobler's **`a1`** parameter). **Units:** microns^{-1} **Default**: value taken from implant damage moments file or built-in model.
`SCALE`	number	The damage scaling factor (Hobler's nv[] parameter; related to the number of Frenkel pairs created for each implanted dopant ion). **Units:** none **Default**: value taken from implant damage moments file or built-in model.
`D.FRAC`	number	The fraction of the implant dose that uses the first set of Pearson distribution moments of the dual-Pearson model. **Units:** none **Default:** 1.0
`D.RANGE`	number	The first moment (projected range) of the second Pearson distribution of the dual-Pearson model. **Units:** microns **Default:** none
`D.SIGMA`	number	The second moment (standard deviation) of the second Pearson distribution of the dual-Pearson model. **Units:** microns **Default:** none

Parameter	Type	Definition
`D.GAMMA`	number	The third moment ratio (skewness) of the second Pearson distribution of the dual-Pearson model. **Units:** none **Default:** 0.0
`D.KURTOS`	number	The fourth moment ratio (kurtosis) of the second Pearson distribution of the dual-Pearson model. **Units:** none **Default:** calculated from the first three moments
`D.LSIGMA`	number	The lateral (perpendicular to the ion beam) standard deviation for the second Pearson distribution of the dual-Pearson model. **Units:** microns **Default: `D.SIGMA`**
`D.LSLOPE`	number	The slope of the lateral standard deviation for the second Pearson distribution of the dual-Pearson model. **Units:** microns **Default:** 0.0
`TILT`	number	The angle between the direction of the ion beam and direction along which the specified moments are measured. **Units:** degrees **Default:** 0.0
`IMPURITY`	character	The name of the impurity for which the Taurus Analytic Implant Model parameter values are set. Applies only to the Taurus Analytic Implant Model. **Default:** none **Synonym: `IMP`**
`ANTIMONY`	logical	Specifies that the Taurus Analytic Implant Model parameter values are set for antimony. Applies only to the Taurus Analytic Implant Model. **Default:** false **Synonym: `SB`**
`ARSENIC`	logical	Specifies that the Taurus Analytic Implant Model parameter values are set for arsenic. Applies only to the Taurus Analytic Implant Model. **Default:** false **Synonym: `AS`**
`BORON`	logical	Specifies that the Taurus Analytic Implant Model parameter values are set for boron. Applies only to the Taurus Analytic Implant Model. **Default:** false **Synonym: `B`**
`BF2`	logical	Specifies that the Taurus Analytic Implant Model parameter values are set for boron in the form of BF2. Applies only to the Taurus Analytic Implant Model. **Default:** false
`PHOSPHOR`	logical	Specifies that the Taurus Analytic Implant Model parameter values are set for phosphorus. Applies only to the Taurus Analytic Implant Model. **Default:** false **Synonym: `P`**

Parameter	Type	Definition
D.SUP.R	number	Sets the value of the **MINRATIO** parameter in the Effective Channeling Suppression Model for the specified material and impurity. Applies only to the Taurus Analytic Implant Model. **Units:** none **Default:** the current value for the specified impurities and materials
D.SUP.E	number	Sets the value of the **EXPONENT** parameter in the Effective Channeling Suppression Model for the specified material and impurity. Applies only to the Taurus Analytic Implant Model. **Units:** none **Default:** the current value for the specified impurities and materials
RANGE.F	number	Sets the value of the **RANGEFACTOR** parameter in the Profile Reshaping Model for the specified material and impurity. Applies only to the Taurus Analytic Implant Model. **Units:** none **Default:** the current value for the specified impurities and materials
SIGMA.F	number	Sets the value of the **SIGMAFACTOR** parameter in the Profile Reshaping Model for the specified material and impurity. Applies only to the Taurus Analytic Implant Model. **Units:** none **Default:** the current value for the specified impurities and materials
D.RAN.F	number	Sets the value of the **CHANNELINGRANGEFACTOR** parameter in the Profile Reshaping Model for the specified material and impurity. Applies only to the Taurus Analytic Implant Model. **Units:** none **Default:** the current value for the specified impurities and materials
D.SIG.F	number	Sets the value of the **CHANNELINGSIGMAFACTOR** parameter in the Profile Reshaping Model for the specified material and impurity. Applies only to the Taurus Analytic Implant Model. **Units:** none **Default:** the current value for the specified impurities and materials
TABLE	character	Sets the default value of the file name suffix for the implant tables which contain the desired implant data in the format of **ION_IN_MATERIAL_SUFFIX**. This parameter changes the default suffix for all materials. Use the **DATA.SUF** parameter of the **IMPLANT** command to change the suffix for a single process step. Applies only to the Taurus Analytic Implant Model. **Default**: **STANDARD**
DAM.TABL	character	Sets the default value of the file name suffix for the implant tables which contain the desired implant damage data in the format of **ION_DAMAGE_IN_MATERIAL_SUFFIX**. This parameter changes the default suffix for all materials. Use the **DAM.SUF** parameter of the **IMPLANT** command to change the suffix for a single process step. Applies only to the Taurus Analytic Implant Model. **Default**: **STANDARD**

Description

The **MOMENT** statement specifies moments to be used by the ion implantation model. The values specified by **MOMENT** statements are used instead of values in the implant data file *s4imp0* on **IMPLANT** statements that specify the **MOMENTS** parameter.

One full set of moments for the Gaussian, single Pearson, or dual Pearson implant model can be specified for each material. A **MOMENT** statement for a particular material replaces any previously specified moments for that material. The **CLEAR** parameter clears any user specified moment data for all materials.

When the **MOMENTS** parameter is specified on an **IMPLANT** statement, values specified on **MOMENT** statements are used for materials for which **MOMENT** statements were given. Moments for other materials are obtained from the moments data file, as usual.

Optional and Required Model Parameters

The models or the **MOMENT** parameter have mandatory and optional parameters. They are listed below.

- For the Gaussian implant model, **RANGE** and **SIGMA** must be specified, and **LSIGMA** is optional.
- For the (single) Pearson model, **RANGE**, and **SIGMA** must be specified, and **GAMMA**, **LSIGMA**, and **KURTOSIS** are optional.
- For the dual Pearson model, the **D.FRAC**, **D.RANGE**, and **D.SIGMA** parameters must be specified in addition to the parameters of the single Pearson model; **D.GAMMA**, **D.LSIGMA**, and **D.KURTOS** are optional.
- The **TILT** parameter is optional.

Using the MOMENT Statement

The **MOMENT** statement makes it easy to experiment with different implant distribution parameters or to specify alternate parameters for critical implant steps. However, if a particular set of moments is needed frequently, it is easier to create an alternate moment data file.

When the **MOMENTS** parameter is specified on an **IMPLANT** statement, moments from preceding **MOMENT** statements are used without regard to the implant energy or species being implanted. It is your responsibility to ensure that the moments are appropriate for the implant being simulated.

The **TILT** parameter specifies the angle between the direction of the ion beam during implant and the direction along which the specified moments were (or would be) measured. In a typical experiment, ions are implanted into a tilted wafer and the resulting dopant distribution is measured perpendicular to the surface. If the moments of the measured distribution are specified on the **MOMENT** statement, then the tilt of the wafer during the implant should be specified by the **TILT** parameter.

Modifiable Hobler Damage Moments

You now can specify the parameters of the implant damage model on the **MOMENT** statement or in a table contained in a file. The implant damage model has been extended to allow the lateral spread of the damage profile to be different from the vertical spread and to vary (linearly) with depth. Parameters for the implant damage model come from one of the following sources:

- The **MOMENT DAMAGE** statement can be used to specify the **RANGE, SIGMA**, **LAMBDA**, **LSIGMA**, **LSLOPE**, and **SCALE** parameters used by the damage implant model. **LAMBDA** specifies the location and slope of the exponential tail of the damage profile, and **SCALE** specifies the magnitude of the profile. The remaining parameters have the same meaning as the for the Gaussian model for implanted doping profiles. Parameters specified with **MOMENT DAMAGE** are used on subsequent **IMPLANT** statements that specify the **MOMENTS** parameter. Parameters not specified on a **MOMENT DAMAGE** statement are obtained from one of the following sources. A **MOMENT CLEAR** statement or a **MOMENT DAMAGE** statement with no model parameters cancels the effects of any preceding **MOMENT DAMAGE** statement.
- If **DAM.TABL** is specified on the **IMPLANT** statement or on an **IMPURITY** statement for the implanted dopant, damage parameters are taken from the specified table in an implant damage moments file. The file must be specified on the **DAM.FILE** parameter on the **IMPLANT** statement, by the *S4IDAM* environment variable; otherwise the file named “s4idam” in the current working directory is used. The format of the implant damage moments file is described below. A **DAM.TABL** specified on an I**MPLANT** statement takes precedence over **DAM.TABL** specified on an **IMPURITY** statement.

Damage parameters not specified on a `DAMAGE MOMENT` statement or in a damage moments file are calculated from the built-in model of Hobler, as in previous releases of TSUPREM-4. `LSIGMA` is assumed to be the same as **SIGMA** and **LSLOPE** is zero. The Hobler’s model ignores the exponential tail of the damage profile at low energies for light ions. This is only done if **LAMBDA** is obtained from the built-in model; if you specify **LAMBDA** on a **MOMENT DAMAGE** statement or in an implant damage moments file, the specified value is used to compute the exponential tail regardless of the implant energy or ion type.

Examples

1. In the following series of statements:

```
MOMENT   SILICON RANGE=0.195 SIGMA=0.072 +
         GAMMA=-0.65 KURTOSIS=22.1
IMPLANT BORON DOSE=1e15 ENERGY=70 MOMENTS
```

The implantation of boron into silicon uses the (single) Pearson distribution specified on the **MOMENT** statement. In materials other than silicon, distribution parameters from the implant data file are used.

2. In the statement:

```
IMPLANT ARSENIC DOSE=2e15 ENERGY=40
```

parameters for the arsenic distribution come from the implant data file. Parameters specified on **MOMENT** statements are not used because **MOMENTS** is not specified on the **IMPLANT** statement.

Additional Note

The implanted dose in the silicon is reduced by the dose that is stopped in screening layers or backscattered from the surface, and thus depends weakly on the implant moments in the screening layer. Because the calculated damage is proportional to the dose in the silicon, the amount of damage is also weakly dependent on the moments in the screening layer.

MATERIAL

The **`MATERIAL`** statement sets the physical properties of materials.

```
MATERIAL

   { ( MATERIAL=<c> [NEW])| SILICON | OXIDE | OXYNITRI | NITRIDE
    | POLYSILI | PHOTORES | ALUMINUM | AMBIENT
   }
   [TIF.NAME=<c>] [MD.INDEX=<n>] [IMPL.TAB=<c>]
   [DY.DEFAU=<n>] [E.FIELD] [ION.PAIR] [IP.OMEGA=<n>]
   [NI.0=<n>] [NI.E=<n>] [NI.F=<n>]
   [EG.ADD=<c> [MODEL=<c>]] [EPSILON=<n>] [DENSITY=<n>] [AT.RADIU=<n>]
   [VISC.0=<n>] [VISC.E=<n>] [VISC.X=<n>] [VC=<c>] [TEMPERAT=<c>]
   [YOUNG.M=<n>] [POISS.R=<n>] [LCTE=<c>]
   [INTRIN.S=<n>] [SURF.TEN=<n>] [STRENGTH=<n>]
   [C11.0=<n>] [C12.0=<n>] [C44.0=<n>]
   [C11.1=<n>] [C12.1=<n>] [C44.1=<n>] [ANISOEL]
   [ { (SEMICOND [AFFINITY=<n>] [BANDGAP=<n>] [EGALPH=<n>] [EGBETA=<n>]
                  [N.CONDUC=<n>] [N.VALENC=<n>] [NC.F=<n>] [NV.F=<n>]
                  [G.DONOR=<n>] [E.DONOR=<n>]
                  [G.ACCEP=<n>] [E.ACCEP=<n>]
                  [BOLTZMAN] [IONIZATI]
                  [QM.BETA=<n>] [QM.YCRIT=<n>])
     | ( CONDUCTO [WORKFUNC=<n>] ) }
   ]
   [MAX.DAMA=<n>] [DAM.GRAD=<n>] [D.PCONST=<n>] [D.PME=<n>] [D.PDLF=<n>]
   [D.PDLME=<n>] [D.PDRE=<n>] [D.PNORM=<n>] [D.PDE=<n>] [D.PHDF=<n>]
   [D.PME=<n>] [D.PLF=<n>] [D.PLME=<n>]
   [POLYCRYS] [GRASZ.0=<n>] [GRASZ.E=<n>] [TEMP.BRE=<n>] [MIN.GRAI=<n>]
   [FRAC.TA=<n>]  [G.DENS=<n>] [F11=<n>] [F22=<n>]  [ALPHA=<n>]
   [GEOM=<n>] [GAMMA.0=<n>] [GAMMA.E=<n>] [DSIX.0=<n>] [DSIX.E=<n>]
   [DSIM.0=<n>] [DSIM.E=<n>] [DSIMM.0=<n>] [DSIMM.E=<n>] [DSIP.0=<n>]
   [DSIP.E=<n>] [GBE.0=<n>] [GBE.H=<n>] [GBE.E=<n>] [NSEG=<n>]
   [TBU.0=<n>] [TBU.E=<n>] [TOXIDE=<n>] [EAVEL.0=<n>] [EAVEL.E=<n>]
   [DLGX.0=<n>] [DLGX.E=<n>] [GB.THICK=<n>]
   [N.SPECIE] [AT.NUM=<n>] [AT.WT=<n>] [MOL.WT=<n>]
   [AT.WT.1] [AT.WT.2] [AT.WT.3]
   [AT.NUM.1] [AT.NUM.2] [AT.NUM.3]
   [ABUND.1] [ABUND.2] [ABUND.3]
   [DEBYE.T=<n>] [AMOR.THR=<n>] [SURF.DIS] [D.SIM=<n>] [V.RAN.D=<n>]
   [AMORPH | POLY.MC | CRYSTAL]
   [POLY.ORI=<n>] [POLY.GSZ=<n>] [POLY.FAC=<n>]
   [SIC.2H | SIC.3C | SIC.4H | SIC.6H]
```

Parameter	Type	Definition
`MATERIAL`	character	The following parameters apply to the named material. **Default:** none **Synonym:** **`NAME`**
`NEW`	logical	Used to define a new material. The name specified by the **`MATERIAL`** parameter must not have been previously used as a material name. **Default:** false
`SILICON`	logical	The following parameters apply to silicon. **Default:** false
`OXIDE`	logical	The following parameters apply to oxide. **Default:** false
`OXYNITRI`	logical	The following parameters apply to oxynitride. **Default:** false
`NITRIDE`	logical	The following parameters apply to nitride. **Default:** false
`POLYSILI`	logical	The following parameters apply to polysilicon. **Default:** false **Synonym:** **`POLY`**
`PHOTORES`	logical	The following parameters apply to photoresist. **Default:** false
`ALUMINUM`	logical	The following parameters apply to aluminum. **Default:** false
`AMBIENT`	logical	The following parameters apply to ambient. **Default:** false **Synonym**: **`GAS`**
`TIF.NAME`	character	The name by which this material is known in the TIF materials database. ***Note***: This value of this parameter is case-sensitive. **Default:** the name of the material
`MD.INDEX`	number	The index to be used for this material when saving the structure in Medici file format. These correspond to the <*matx*> indices listed in Appendix D. **Units:** none **Default:** 0 (material ignored by Medici)
`IMPL.TAB`	character	The name of the material in the implant moment file to be associated with this material. **Default:** the current value for this material; initially the name of the material
`DY.DEFAU`	number	The grid spacing to be used in growing layers of the material. This value is scaled by the value of the **`GRID.FAC`** parameter on the **`METHOD`** statement. A value of zero specifies that no grid is to be added in growing layers. **Units:** microns **Default:** the current value; initially 0.0 **Synonym:** **`DX.DEFAU`**

Parameter	Type	Definition
`E.FIELD`	logical	Specifies that the electric field terms are to be included in the impurity diffusion equations for this material. (Meaningful only for silicon and polysilicon.) **Default:** the current value for this material; initially true
`ION.PAIR`	logical	Enables the model for pairing between donor and acceptor ions in this material. **Default:** the current value for this material; initially true for silicon and polysilicon
`IP.OMEGA`	number	The multiple of the intrinsic carrier concentration at which ion pairing becomes significant in this material. **Units:** none **Default:** the current value for this material; initially 6.0 for silicon and polysilicon
`NI.0`	number	The prefactor in the expression for the intrinsic carrier concentration. **Units:** carriers/cm^3*(degrees Kelvin)$^{\mathtt{NI.F}}$ **Default:** the current value for this material; initially 3.87e16
`NI.E`	number	The activation energy in the expression for the intrinsic carrier concentration. **Units:** electron volts **Default:** the current value for this material; initially 0.605
`NI.F`	number	The temperature exponent in the expression for the intrinsic carrier concentration. **Units:** none **Default:** the current value for this material; initially 1.5 **Synonym:** `NI.POW`
`EG.ADD`	character	The expression of the energy bandgap change **Units:** electron volts **Default:** 0.0
`MODEL`	character	The model name for the expression `EG.ADD` of the energy bandgap change **Default:** None
`EPSILON`	number	The relative permittivity of the material. **Units:** none **Default:** the current value for this material; see Appendix A for initial values
`DENSITY`	number	The density of the material. **Units:** gm/cm^3 **Default:** the current value; see Appendix A for initial values.
`AT.RADIU`	number	Atom covalent radius. Used for calculating the misfit strain when one of the `DEPO.MIS` or `DIFF.MIS` or `STN.MIS` parameters is turned on. **Units:** angstroms **Default:** the current value; see Appendix A for initial values.
`VISC.0`	number	The exponential prefactor for the viscosity. **Units:** gm/(cm*sec) **Default:** the current value for this material

Parameter	Type	Definition
`VISC.E`	number	The activation energy for the viscosity. **Units:** electron volts **Default:** the current value for this material
`VISC.X`	number	The incompressibility factor. A value of 0.5 corresponds to an infinitely incompressible material. The value must be strictly less than 0.5. **Units:** none **Default:** the current value for this material
`VC`	character	A table of activation volumes as a function of temperature for the dependence of material viscosity on shear stress for the specified material in inert-ambient diffusions. Entries in the table correspond to temperatures given by the `TEMPERAT` parameter (see text). **Units:** $Å^3$ **Default:** the current value for this material
`TEMPERAT`	character	A list of temperatures corresponding to the values of the `VC` parameter. **Units:** degrees Celsius **Default:** none
`YOUNG.M`	number	Young's modulus for the material. **Units:** $dynes/cm^2$ **Default:** the current value for this material
`POISS.R`	number	Poisson's ratio for the material. **Units:** none **Default:** the current value for this material
`LCTE`	character	An expression giving the linear coefficient of thermal expansion as a function of absolute temperature, called T in the expression. It is given as a fraction, not as a percentage. **Units:** none **Default:** the current value for this material
`INTRIN.S`	number	The initial uniform stress state of a material, such as a thin film of nitride deposited on the substrate. **Units:** $dynes/cm^2$ **Default:** the current value for this material
`SURF.TEN`	number	The surface tension for this material. Affects reflow during oxidation with the `VISCOUS` model. **Units:** dynes/cm **Default:** the current value for this material
`STRENGTH`	number	The maximum absolute stress value for this material. **Units:** $dynes/cm^2$ **Default:** the current value for this material; initially `YOUNG.M`/20.0
`ANISOEL`	logical	Specifies that the material is anisotropically elastic. **Default:** false

Parameter	Type	Definition
`C11.0`	number	The elastic constant Sxx/Exx in crystallographic axes. **Units:** dynes/cm^2 **Default:** the current value for this material
`C11.1`	number	The first-order temperature dependence of the elastic constant Sxx/Exx in crystallographic axes **Units:** dynes/cm^2/^{o}K **Default:** the current value for this material
`C12.0`	number	The elastic constant Sxx/Eyy in crystallographic axes **Units:** dynes/cm^2 **Default:** the current value for this material
`C12.1`	number	The first-order temperature dependence of the elastic constant Sxx/Eyy in crystallographic axes **Units:** dynes/cm^2/^{o}K **Default:** the current value for this material
`C44.0`	number	The elastic constant Sxy/Exy in crystallographic axes **Units:** dynes/cm^2 **Default:** the current value for this material
`C44.1`	number	The first-order temperature dependence of the elastic constant Sxy/Exy in crystallographic axes **Units:** dynes/cm^2/^{o}K **Default:** the current value for this material
`SEMICOND`	logical	Specifies that the material is a semiconductor. **Default:** false
`AFFINITY`	number	The electron affinity of the material. **Units:** electron volts **Default:** the current value for this material
`BANDGAP`	number	The energy band gap of the material. **Units:** electron volts **Default:** the current value for this material
`EGALPH`	number	The value of alpha used in calculating the energy bandgap as a function of temperature. **Units:** eV/Kelvin **Default:** the current value for this material
`EGBETA`	number	The value of beta used in calculating the energy bandgap as a function of temperature. **Units:** Kelvins **Default:** the current value for this material
`N.CONDUC`	number	The effective density of electron states in the conduction band of the material at 300° K. **Units:** #/cm^3 **Default:** the current value for this material

Parameter	Type	Definition
`N.VALENC`	number	The effective density of hole states in the valence band of the material at 300° K. **Units:** #/cm^3 **Default:** none
`NC.F`	number	The exponent of the temperature dependence for the effective density of electron states in the conduction band of the material. **Units:** none **Default:** the current value for this material
`NV.F`	number	The exponent of the temperature dependence for the effective density of hole states in the valence band of the material. **Units:** none **Default:** the current value for this material
`G.DONOR`	number	The donor impurity degeneracy factor of the material. **Units:** none **Default:** the current value for this material
`E.DONOR`	number	The donor impurity ionization energy of the material. **Units:** electron volts **Default:** the current value for this material
`G.ACCEP`	number	The acceptor impurity degeneracy factor of the material. **Units:** none **Default:** the current value for this material
`E.ACCEP`	number	The acceptor impurity ionization energy of the material. **Units:** electron volts **Default:** the current value for this material
`BOLTZMAN`	logical	Specifies that Boltzmann carrier statistics are used for this material during electrical calculations. If this parameter is false, Fermi-Dirac statistics are used. **Default:** the current value for this material
`IONIZATI`	logical	Specifies that complete impurity ionization is used for this material during electrical calculations. If this parameter is false, the impurities are considered to be incompletely ionized. **Default:** the current value for this material
`QM.BETA`	number	The proportional factor for van Dort's QM modeling. **Units:** eVcm **Default:** the current value for this material
`QM.YCRIT`	number	The critical depth of quantization for van Dort's QM modeling. **Default:** the current value for this material
`CONDUCTO`	logical	Specifies that the material is a conductor. **Default:** false
`WORKFUNC`	number	The work function of the material. **Units:** electron volts **Default:** the current value for this material

Parameter	Type	Definition
`MAX.DAMA`	number	The damage threshold for amorphization of this material. **Synonym**: **`AMOR.THR`** **Units:** #/cm^3 **Default:** the current value for this material; initially 1.15×10^{22} for silicon
`DAM.GRAD`	number	The sharpness of the amorphous/single-crystal interface in this material following solid-phase epitaxial regrowth **Units:** none **Default:** the current value for this material; initially 10 for silicon
`D.PCONST`	number	An empirical constant for the effective +n implant damage model. **Units:** none **Default:** the current value for this material; initially 3.25 for silicon
`D.PDME`	number	The mass exponent in the low dose factor for the effective +n implant damage model. **Units:** none **Default:** the current value for this material; initially –0.5 for silicon
`D.PDLF`	number	The prefactor for lambda in the low dose factor for the effective +n implant damage model. **Units:** none **Default:** the current value for this material; initially –2 for silicon
`D.PDLME`	number	The mass exponent in lambda in the low dose factor for the effective +n implant damage model. **Units:** none **Default:** the current value for this material; initially –0.5 for silicon
`D.PDRE`	number	The projected range exponent in the low dose factor for the effective +n implant damage model. **Units:** none **Default:** the current value for this material; initially 0 for silicon
`D.PNORM`	number	The normalization for dose for the effective +n implant damage model. **Units:** #/cm^3 **Default:** the current value for this material; initially 3e12 for silicon
`D.PDE`	number	The dose exponent for the effective +n implant damage model. **Units:** none **Default:** the current value for this material; initially 1.0 for silicon
`D.PHDF`	number	The constant in the high dose factor for the effective +n implant damage model. **Units:** none **Default:** the current value for this material; initially 0.0905 for silicon
`D.PME`	number	The mass exponent in the high dose factor for the effective +n implant damage model. **Units:** none **Default:** the current value for this material; initially 0.85 for silicon

Parameter	Type	Definition
`D.PLF`	number	The prefactor for lambda in the high dose factor for the effective +n implant damage model. **Units:** none **Default:** the current value for this material; initially –2 for silicon
`D.PLME`	number	The mass exponent in lambda in the high dose factor for the effective +n implant damage model. **Units:** none **Default:** the current value for this material; initially –0.5 for silicon
`POLYCRYS`	logical	Specifies that the polycrystalline diffusion and grain-growth models are to be used for this material. **Default:** the current value for this material; initially true for polysilicon
`GRASZ.0`	number	The pre-exponential factor for the as-deposited grain size during polycrystalline (high-temperature) deposition. **Units:** none **Default:** the current value for this material
`GRASZ.E`	number	The activation energy for the as-deposited grain size during polycrystalline (high-temperature) deposition. **Units:** electron volts **Default:** the current value for this material
`TEMP.BRE`	number	The threshold temperature for amorphous deposition; deposition is amorphous at lower temperatures and polycrystalline at higher temperatures. **Units:** ° Celsius **Default:** the current value for this material
`MIN.GRAI`	number	The minimum as-deposited grain size. **Units:** microns **Default:** the current value for this material
`FRAC.TA`	number	The geometrical factor for recrystallized grain size in amorphous regions. **Units:** none **Default:** the current value for this material
`G.DENS`	number	The geometrical factor for the density of grain boundaries. **Units:** none **Default:** the current value for this material; initially 2.0 for polysilicon
`F11`	number	The geometrical factor for grain-boundary diffusion perpendicular to the columnar direction. **Units:** none **Default:** the current value for this material; initially 1.0 for polysilicon
`F22`	number	The geometrical factor for grain-boundary diffusion parallel to the columnar direction. **Units:** none **Default:** the current value for this material; initially 2.0 for polysilicon

Parameter	Type	Definition
ALPHA	number	The geometrical factor relating the average grain boundary velocity to the grain growth rate. **Units:** none **Default:** the current value for this material; initially 1.33 for polysilicon
GEOM	number	The geometrical factor for the grain growth rate. **Units:** none **Default:** the current value for this material; initially 6.0 for polysilicon
GAMMA.0	number	Pre-exponential factor for the enhancement of silicon self-diffusivity at grain boundaries. **Units:** none **Default:** the current value for this material
GAMMA.E	number	Activation energy for the enhancement of silicon self-diffusivity at grain boundaries. **Units:** electron volts **Default:** the current value for this material
DSIX.0	number	Pre-exponential factor for the neutral component of silicon self diffusivity. **Units:** cm^2/sec **Default:** the current value for this material
DSIX.E	number	Activation energy for the neutral component of silicon self diffusivity. **Units:** electron volts **Default:** the current value for this material
DSIM.0	number	Pre-exponential factor for the singly negative component of silicon self-diffusivity. **Units:** cm^2/sec **Default:** the current value for this material
DSIM.E	number	Activation energy for the singly negative component of silicon self-diffusivity. **Units:** electron volts **Default:** the current value for this material
DSIMM.0	number	Pre-exponential factor for the doubly-negative component of silicon self-diffusivity. **Units:** cm^2/sec **Default:** the current value for this material
DSIMM.E	number	Activation energy for the doubly-negative component of silicon self-diffusivity. **Units:** electron volts **Default:** the current value for this material
DSIP.0	number	Pre-exponential factor for the singly-positive component of silicon self-diffusivity. **Units:** cm^2/sec **Default:** the current value for this material
DSIP.E	number	Activation energy for the singly-positive component of silicon self-diffusivity. **Units:** electron volts **Default:** the current value for this material

Parameter	Type	Definition
GBE.0	number	The grain boundary energy for normal grain growth. **Units:** electron volts **Default:** the current value for this material
GBE.H	number	The geometrical factor for the reduction of grain growth rate at large grain sizes. **Units:** none **Default:** the current value for this material
GBE.1	number	The grain boundary energy for secondary grain growth. **Units:** electron volts **Default:** the current value for this material
NSEG	number	The exponent for solute drag on grain growth. **Units:** none **Default:** the current value for this material
TBU.0	number	The prefactor in the expression for the time constant for polysilicon/silicon interfacial oxide break-up. **Units:** seconds **Default:** the current value for this material
TBU.E	number	The activation energy for the time constant for polysilicon/silicon interfacial oxide break-up. **Units:** electron volts **Default:** the current value for this material
TOXIDE	number	The thickness of the interfacial oxide between deposited polysilicon layers and single-crystal silicon. **Units:** microns **Default:** the current value for this material
EAVEL.0	number	The prefactor in the expression for the velocity driving epitaxial regrowth of a polycrystalline layer. **Units:** cm/sec **Default:** the current value for this material
EAVEL.E	number	The activation energy for the velocity driving epitaxial regrowth of a polycrystalline layer. **Units:** electron volts **Default:** the current value for this material
DLGX.0	number	The prefactor in the expression for the grain size diffusivity. **Units:** cm^2/sec **Default:** the current value for this material
DLGX.E	number	The activation energy for the grain size diffusivity. **Units:** electron volts **Default:** the current value for this material
GB.THIC	number	The effective grain boundary thickness for enhanced oxidation. **Units**: microns **Default**: the current value for this material; in it ally 0.0005 for polysilicon and 0 for other materials

Parameter	Type	Definition
N.SPECIE	number	The number of species in compound materials. Please note that this parameter is used in MC model only. **Units**: none **Default**: none
AT.NUM	number	The average atomic number of the material. This is the sum of the atomic numbers of the atoms in a molecule divided by the number of atoms. **Units:** none **Default:** the current value; see Appendix A for initial values.
AT.WT	number	The average atomic weight of the atoms of the material. This is the molecular weight of the material divided by the number of atoms per molecule. **Units:** atomic mass units **Default:** the current value; see Appendix A for initial values.
MOL.WT	number	The molecular weight of the material. **Units:** atomic mass units **Default:** the current value; see Appendix A for initial values.
AT.WT.1	number	The atomic weight of the first species in the compound material. **Units**: atomic mass units **Default**: none
AT.WT.2	number	The atomic weight of the second species in the compound material. **Units**: atomic mass units **Default**: none
AT.WT.3	number	The atomic weight of the third species in the compound material. **Units**: atomic mass units **Default**: none
AT.NUM.1	number	The atomic number of the first species in the compound material. **Units**: none **Default**: none
AT.NUM.2	number	The atomic number of the second species in the compound material. **Units**: none **Default**: none
AT.NUM.3	number	The atomic number of the third species in the compound material. **Units**: none **Default**: none
ABUND.1	number	The fractional abundance of the first species in the compound material. **Units**: none **Default**: none
ABUND.2	number	The fractional abundance of the second species in the compound material. **Units**: none **Default**: none
ABUND.3	number	The fractional abundance of the third species in the compound material. **Units**: none **Default**: none

Parameter	Type	Definition
DEBYE.T	number	Debye temperature for the material. Applied to Taurus MC only. **Units:** ^{0}C **Default:** 245.85^{0}C for silicon.
AMOR.THR	number	Critical point defect density for crystal amorphization. **Synonym**: **MAX.DAMA** **Units:** #/cm^3 **Default**: 1.15e22 cm-3 for silicon.
SURF.DIS	number	Surface disorder of the material. Applied to Taurus MC only. **Units**: um **Default:** 5e-4
D.SIM	number	This parameter is the maximum distance among the simultaneous collision partners, which can be used to fine tune the channeling tails. This parameter is used for Taurus MC only. **Units:** lattice constant **Default**: 0.5
V.RAN.D	number	This parameter allows the vacancy positions to be shifted by random displacements (**RAND*V.RAN.D**) from lattice sites (where rand is a random number from 0 to 1). This parameter is used for Taurus MC cascade damage model only. **Units**: microns **Default**: 0
AMORPH	logical	Specifies that this material is treated as amorphous. Used by Taurus MC only. **Default:** the current value for this material.
POLY.MC	logical	Specifies that this material is treated as polycrystalline. Used by Taurus MC only. **Default:** the current value for this material.
CRYSTAL	logical	Specifies that this material is treated as single-crystalline. Used by Taurus MC only. **Default:** the current value for this material; true for silicon, GaAs, and SiC.
POLY.ORI	number	The orientation of polycrystalline material (one of 100, 110, and 111). Valid for polysilicon and Taurus MC only. **Units**: none **Default**: 100
POLY.GSZ	number	The grain size for polycrystalline material. Valid for polysilicon and Taurus MC only. **Units**: um **Default**: 0
POLY.FAC	number	The adjustable parameter for polysilicon MC implant models (both OLD.MC and Taurus MC). Valid for polysilicon material only. **Units**: none **Default**: 1
SIC.2H	logical	Specifies that SiC material is 2H polytype. Used by Taurus MC only. **Default:** false

Parameter	Type	Definition
`SIC.3C`	logical	Specifies that SiC material is 3C polytype. Used by Taurus MC only. **Default:** false
`SIC.4H`	logical	Specifies that SiC material is 4H polytype. Used by Taurus MC only. **Default:** true for SiC
`SIC.6H`	logical	Specifies that SiC material is 6H polytype. Used by Taurus MC only. **Default:** false

Description

This statement is used to define materials and specify their properties. A material can be specified either by name, using the `MATERIAL` parameter, or with one of the `SILICON`, `OXIDE`, `OXYNITRI`, `NITRIDE`, `POLYSILI`, `PHOTORES`, `ALUMINUM`, or `AMBIENT` parameters. A new material can be defined by specifying the `NEW` parameter, the material name, and the properties of the material.

Note:

> ***You must ensure that all material properties required for a particular simulation step (e.g., viscosity and compressibility for oxidation or density and atomic number and weight for MC ion implantation) have been specified.***

The density, average atomic number, and average atomic weight of the material are used by the MC ion implantation model. The density and molecular weight are used to calculate the volume changes that occur when materials are produced or consumed during reactions with other materials.

Viscosity and Compressibility

The `VISC.0`, `VISC.E`, and `VISC.X` parameters specify the viscosity as a function of temperature of the material for use with the `VISCOELA` and `VISCOUS` models for material flow. `VC` specifies the activation volume for the dependence of the material viscosity on shear stress when the stress-dependent `VISCOELA` or `VISCOUS` model is used. `YOUNG.M` and `POISS.R` give the viscosity and compressibility for use with the `COMPRESS` model and elasticity for the `VISCOELA` model. See Reference [56] in Chapter 3 and Reference [55] in Appendix A.

The only material properties that apply to `AMBIENT` are the viscosity parameters `VISC.0`, `VISC.E`, and `VISC.X`. These parameters apply to included voids, which may be formed during oxidation.

Stress Dependence

The parameters `VC` and `TEMPERAT` are used to specify the activation volume for dependence of viscosity on shear stress as functions of temperature during inert-ambient diffusions. A separate table is maintained for each material. `VC` specifies a list of values corresponding to the temperatures given by `TEMPERAT`. The portion of the table spanned by the specified temperatures is replaced by the specified values; the number of values must be the same as the number of temperatures, and the temperatures must be given in order, lowest to highest. The **CLEAR** parameter is used to clear the table before setting any values.

If **V.COMPAT** (on the **OPTION** statement) is less than 6.6, the specified values of the activation volumes apply to the specified material in all ambients, including oxidizing ambients.

An Arrhenius interpolation is used between values in the table. For temperatures outside the range of the table, the nearest value is used.

Modeling the Energy Bandgap

The following example shows the effect of energy bandgap narrowing due to germanium contents because the energy bandgap of Ge material is smaller by 0.4 eV than that of Si material. By assuming the linear decrease of energy bandgap to the Ge fraction in Si, it might be modeled by;

```
MATERIAL MODEL=EGSIGE MAT=Si EG.ADD=-0.4*Ge/5E22
```

The **MODEL** specification is optional. It is useful to turn on/off the model. For example, the model can be turned off by;

```
METHOD MODEL=EGSIGE ^ENABLE
```

Examples

1. The following statement specifies the density of silicon:

   ```
   MATERIAL MATERIAL=SILICON DENSITY=2.33
   ```

2. The following statement gives the thermal expansion coefficient of nitride as a function of absolute temperature *T*:

   ```
   MATERIAL NITRIDE LCTE=(3e-6 + 2e-10 * T) +
            INTRIN.S=1.4e10
   ```

   Thus, at 0K the coefficient is .0003%/K. The initial stress in the nitride film is 1.4e10 dynes/cm^2.

3. The following statement specifies an oxide viscosity of 10^9 poise and a surface tension of 20 dynes/cm:

   ```
   MATERIAL MATERIAL=OXIDE VISC.0=1e9 VISC.e=0 +
            SURF.TEN=20
   ```

   These are values that might be used to simulate reflow of phosphosilicate or borophosphosilicate glass.

4. The following statements define a new material, tungsten disilicide, and specify some of its properties:

   ```
   MATERIAL NEW MAT=WSi2 DENSITY=9.857 AT.NUM=34 +
            AT.WT=80.01
   MATERIAL MAT=WSi2 DY.DEFAU=0.1
   ```

A grid spacing of 0.1 microns (scaled by the value of **GRID.FAC**) is used when growing tungsten silicide.

5. The following statement defines a new material, silicon carbide, and specifies some of its properties:

```
MATERIAL NEW MAT=SIC DENSITY=3.21 AT.NUM=10.0 +
  AT.WT=20.04 MOL.WT=40.08 N.SPECIE=2 +
  AT.NUM.1=14.0 AT.WT.1=28.069 AT.NUM.2=6 +
  AT.WT.2=12.011
```

IMPURITY

The **`IMPURITY`** statement is used to define impurities or modify their characteristics.

```
IMPURITY

   IMPURITY=<c> [( NEW [C.INIT=<n>] )] [TIF.NAME=<c>]
   [IMP.GB=<c>] [IMP.IT=<c>] [IMP.IP=<c>] [IMP.VP=<c>]
   [IMP.CL0=<c>] [IMP.CL1=<c>] [IMP.CL2=<c>]
   [IMP.DDC=<c> DDC.D.N=<n> [{ DDC.I.N=<n>] | DDC.V.N=<n> }] ]
   [ { DONOR | ACCEPTOR | NEUTRAL | CHARGE=<n> } ]
   [AT.NUM=<n>] [AT.WT=<n>] [AT.RADIU=<n>]
   [PD.PAIR] [NEGATIVE] [SOLVE] [STEADY]
   [IMPL.TAB=<c>] [DAM.TABL=<c>]
   [T.ACT.0=<n>] [T.ACT.E=<n>] [ACT.MIN=<n>]
   [D.PNORM=<n>] [D.PDE=<n>] [CM.SEC]

   [MATERIAL=<c>]
     [DIFFUSE]
     [ { VD=<n> | ( VMIPAIR=<n> VMVPAIR=<n> ) | STS.VOL=<n> } ]
     [VALPHAM=<c>] [VALPHAN=<n>] [STN.FAC=<n>]
     [DIP.0=<n>] [DIP.E=<n>] [DIX.0=<n>] [DIX.E=<n>]
     [DIM.0=<n>] [DIM.E=<n>] [DIMM.0=<n>] [DIMM.E=<n>]
     [DVP.0=<n>] [DVP.E=<n>] [DVX.0=<n>] [DVX.E=<n>]
     [DVM.0=<n>] [DVM.E=<n>] [DVMM.0=<n>] [DVMM.E=<n>]
     [DIPAIR.0=<n>] [DIPAIR.E=<n>] [DVPAIR.0=<n>] [DVPAIR.E=<n>]
     [C.STATE=<n> [DIC.0=<n>] [DIC.E=<n>] [DVC.0=<n>] [DVC.E=<n>]]
     [R.I.S=<n>]  [E.I.S=<n>]  [R.V.S=<n>]  [E.V.S=<n>]
     [R.IP.V=<n>] [E.IP.V=<n>] [R.VP.I=<n>] [E.VP.I=<n>]
     [ { IP.V.LOW | IP.V.MED | IP.V.HIG } ]
     [ { VP.I.LOW | VP.I.MED | VP.I.HIG } ]
     [EQ.FRACA] [IP.FRACA=<n>] [VP.FRACA=<n>] [S.FRACA=<n>]
     [ { C.I.ALL | C.INTERS=<n> } ] [ { C.V.ALL | C.VACANC=<n> } ]
     [ { C.IP.ALL | C.IPAIR=<n> } ] [ { C.VP.ALL | C.VPAIR=<n> } ]
     [SS.CLEAR] [SS.TEMP=<n> SS.CONC=<n>]
     [CTN.0=<n>] [CTN.E=<n>] [CTN.F=<n>] [CL.INI.A]
     [DDC.F.0=<n>] [DDC.F.E=<n>] [DDC.T.0=<n>] [DDC.T.E=<n>]
     [DDCF.D.N=<n>] [DDCF.N.N=<n>] [DDCF.I.N=<n>]
     [DDCR.N.N=<n>] [DDCR.I.N=<n>] [IFRACM=<n>]
     [DDCS.0=<n>] [DDCS.E=<n>] [DDCS.N=<n>] [IFRACS=<n>]
     [DVC.F.0=<n>] [DVC.F.E=<n>] [DVC.T.0=<n>] [DVC.T.E=<n>]
     [DVCF.D.N=<n>] [DVCF.N.N=<n>] [DVCF.V.N=<n>]
     [DVCR.N.N=<n>] [DVCR.V.N=<n>] [VFRACM=<n>]
     [C0.TF.0=<n>] [C0.TF.E=<n>] [C0.TR.0=<n>] [C0.TR.E=<n>]
     [C1FRAC.A=<n>] [C2FRAC.A=<n>]
     [C0.INI.F=<n>] [C1.INI.F=<n>] [C2.INI.F=<n>]
     [DDCTOT.C=<n>] [DDCTOT.A=<n>]
     [ACT.NI=<n>] [ACT.AMOR=<n>] [MAX.AMOR=<n>]
```

```
[ { DDCL=<c> | DDC.R=<c> }
  { I.DDC | IP.DDC | V.DDC | VP.DDC }
  [DDC.KF.0=<n>] [DDC.KF.E=<n>]
  {  ( [DDC.KR.0=<n>] [DDC.KR.E=<n>] )
   | ( [DDC.EQ.0=<n>] [DDC.EQ.E=<n>] ) }
  [DDCFRACC=<n>] [DDCFRACA=<n>] [DDC.C.IP=<n>] [DDC.C.VP=<n>]
] [DDCPRINT]

[FGB=<n>] [Q.SITES=<n>] [CG.MAX=<n>]
[GSEG.0=<n>] [GSEG.E=<n>] [GSEG.INI=<n>]
[FVELB.0=<n>] [FVELB.E=<n>] [VELIF.0=<n>] [VELIF.E=<n>]

[MODEL=<c>]
  [EQACTIVE=<c>] [ACTIVE=<c>] [MOBILE=<c>]
  [DI.FAC=<c>] [DV.FAC=<c>]
  [DI.F11=<c>] [DI.F22=<c>] [DV.F11=<c>] [DV.F22=<c>]
[ /MATERIA=<c>
  [SEG.0=<n>] [SEG.E=<n>] [TRANS.0=<n>] [TRANS.E=<n>]
  [RATIO.0=<n>] [RATIO.E=<n>] [SEG.SS]
  [/SEG.0=<n>] [/SEG.E=<n>] [/TRANS.0=<n>] [/TRANS.E=<n>]
  [/RATIO.0=<n>] [/RATIO.E=<n>] [/SEG.SS]
  {SEG.EQ3 | SEG.EQ2 | /SEG.EQ2}
  [Q.INI.0=<n>] [Q.INI.E=<n>] [Q.MAX.0=<n>] [Q.MAX.E=<n>]
  [TWO.PHAS]
  [KSIP.0=<n>] [KSIP.E=<n>] [KSVP.0=<n>] [KSVP.E=<n>]
  [SEG.IFRA=<n>] [SEG.IPOW=<n>] [SEG.VPOW=<n>] [SEG.CPOW=<n>]
]

[ES.RAND=<n>] [ES.F.RAN=<n>] [ES.BREAK=<n>] [ES.F.H=<n>]
[ES.100=<n>] [ES.F.100=<n>] [ES.110=<n>] [ES.F.110=<n>]
[NLOC.PRO=<n>] [NLOC.EXO=<n>] [NLOC.MAX=<n>] [NLOC.K=<n>]
[LOC.FAC=<n>] [CHAN.CRI=<n>] [CHAN.FAC=<n>] [DISP.FAC=<n>]
[DIV.ANG=<n>] [LSS.PRE=<n>] [NLOC.PRE=<n>]
[NLOC.EXP=<n>] [SCR.PAR]
[SURV.RAT=<n>] [AMOR.PAR=<n>] [SCR.PAR] [DISP.THR=<n>]
[CASC.SUR=<n>]
[CASC.AMO=<n>] [CASC.SAT] [CASC.DIS=<n>] [REC.FRAC=<n>]
```

Parameter	Type	Definition
`IMPURITY`	character	The name of the impurity to be defined or modified. **Default:** none **Synonym:** **`IMP`**, **`NAME`**
`NEW`	logical	Specifies that a new impurity is being defined. **Default:** false
`C.INIT`	number	The initial background concentration of the impurity. **Units:** atoms/cm^3 **Default:** the current value for this impurity; initially 0

Parameter	Type	Definition
`TIF.NAME`	character	The name by which this impurity is known in the TIF materials database. ***Note*:** This value of this parameter is case-sensitive. **Default:** the name of the impurity
`IMP.GB`	character	The name of the solution value representing the grain boundary concentration of the impurity named in the **`IMPURITY`** parameter. Indicates that the polycrystalline diffusion model applies to this impurity. **Default:** none
`IMP.IT`	character	The name of the solution value representing the density of the trapped impurity at interface named in the **`IMPURITY`** parameter. Required only when the **`ITRAP`** model is to be applied to the impurity. **Default:** none
`IMP.IP`	character	The name of the solution value representing the concentration of dopant-interstitial pairs named in the **`IMPURITY`** parameter. Required only when the 5-stream diffusion model is to be applied to the impurity. **Default:** none
`IMP.VP`	character	The name of the solution value representing the concentration of dopant-vacancy pairs named in the **`IMPURITY`** parameter. Required only when the 5-stream diffusion model is to be applied to the impurity. **Default:** none
`IMP.CL0`	character	The name of the solution value representing the concentration of precipitates named in the **`IMPURITY`** parameter. Required only when the **`ACT.FULL`** model or the **`DDC.FULL`** model is to be applied to the impurity. **Default:** none
`IMP.CL1`	character	The name of the solution value representing the concentration of precipitates (for **`ACT.TRAN`**) or dopant-interstitial clusters (for **`ACT.FULL`** or **`DDC.FULL`**) named in the **`IMPURITY`** parameter. **Default:** none
`IMP.CL2`	character	The name of the solution value representing the concentration of dopant-vacancy clusters (for **`ACT.FULL`** or **`DDC.FULL`**) named in the **`IMPURITY`** parameter. **Default:** none
`IMP.DDC`	character	The name of the solution value representing the concentration of dopant-defect clusters (for **`DDC.FULL`**) named in the **`IMPURITY`** parameter. **Default:** none
`DDC.D.N`	number	The number of dopant atoms to be clustered into the dopant-defect cluster. **Units:** none **Default:** 0 **Synonym**: **`DDC.N.D`**
`DDC.I.N`	number	The number of silicon interstitials to be clustered into the dopant-defect cluster. **Units:** none **Default:** 0 **Synonym**: **`DDC.N.I`**

Parameter	Type	Definition
`DDC.V.N`	number	The number of vacancies to be clustered into the dopant-defect cluster. **Units:** none **Default:** 0 **Synonym**: **`DDC.N.V`**
`DONOR`	logical	Specifies that the impurity is a donor. **Default:** the current value for this impurity
`ACCEPTOR`	logical	Specifies that the impurity is an acceptor. **Default:** the current value for this impurity
`NEUTRAL`	logical	Specifies that the charged state of the impurity is neutral. **Default:** the current value for this impurity
`CHARGE`	number	Specifies the charged state of the impurity. **Units:** electron charge units **Default:** the current value for this impurity
`AT.NUM`	number	The atomic number of the impurity. **Units:** none **Default:** the current value for this impurity
`AT.WT`	number	The atomic weight of the impurity. **Units:** atomic mass units **Default:** the current value for this impurity
`AT.RADIU`	number	The atomic radius of the impurity. **Units:** Angstrom. **Default:** the current value for this impurity
`PD.PAIR`	logical	Specifies that this impurity diffuses by pairing with point defects. **Default:** true if **`PD.TRANS`** or **`PD.FULL`** is specified;
`NEGATIVE`	logical	Specifies that this impurity solution can have a negative value. **Default:** true for stress solutions, potential, and velocities; otherwise false
`SOLVE`	logical	Specifies that the diffusion equations should be solved for this impurity. **Default:** true if **`NEW`** is specified; otherwise, is the current value for this impurity
`STEADY`	logical	Specifies that the steady-state conditions should be assumed when solving the diffusion equations for this impurity. **Default:** false if **`NEW`** is specified; otherwise, is the current value for this impurity
`IMPL.TAB`	character	The name of the implant moment table to use for this impurity. **Default:** the current implant moment table for this impurity

Parameter	Type	Definition
DAM.TABL	character	The name of a table of damage parameters in an implant damage file. Values of the **RANGE**, **SIGMA**, **LAMBDA**, **LSIGMA**, **LSLOPE**, and **SCALE** parameters for implant damage are taken from the specified table in the implant damage file unless a different table is specified on the **IMPLANT** statement or parameters from a **MOMENT** statement are used. Note: It is not possible to specify separate tables for boron and BF2 on the **IMPURITY** statement. **Default:** no table (i.e., built-in table) or he current implant moment table for this impurity
T.ACT.0	number	The prefactor in the expression for the activation time constant for the **ACT.TRAN** model for this impurity. **Units:** min or sec **Default:** the current value for this impurity.
T.ACT.E	number	The activation energy in the expression for the activation time constant for the **ACT.TRAN** model for this impurity. **Units:** electron volts **Default:** the current value for this impurity.
ACT.MIN	number	The factor by which the intrinsic carrier concentration is multiplied to obtain the minimum activation level used by the **ACT.TRAN** model for this impurity. **Units:** none **Default:** the current value for this impurity.
D.PNORM	number	The normalizing dose for the effective +n implant damage model. **Units:** #/cm^2 **Default:** the current value for this impurity.
D.PDE	number	The power of the dose factor in the effective +n implant damage model. **Units:** none **Default:** the current value for this impurity. **Synonym**: **D.PF**
CM.SEC	logical	If true, parameters involving time are specified in centimeters and seconds; if false, parameters involving time are in microns and minutes. **Default:** false
MATERIAL	character	The name of the material in which the diffusion, activation, and implant parameters apply and material 1 for the segregation and transport parameters. **Default:** Silicon
DIFFUSE	logical	Specifies that the diffusion equation is to be solved. **Default:** true
VD	number	The activation volume of the pressure-dependent migration energy to a dopant-defect pair diffusivity. **Units:** Å^3 **Default:** the current value for this material
VMIPAIR	number	The activation volume of the pressure-dependent migration energy to a dopant-interstitial pair diffusivity **Units:** Å^3 **Default:** the current value for this material

Parameter	Type	Definition
VMVPAIR	number	The activation volume of the pressure-dependent migration energy to a dopant-vacancy pair diffusivity **Units:** Å^3 **Default:** the current value for this material
STS.VOL	number	OBSOLETE. The stress conjugate activation volume used in the stress dependent diffusivity enhancement model **Units:** Å^3 **Default:** the current value for this material
VALPHAM	number	The activation volume of the pressure-dependent formation energy for a dopant-interstitial pair. **Units:** Å^3 **Default:** the current value for this material
VALPHAN	number	The activation volume of the pressure-dependent formation energy for a dopant-vacancy pair. **Units:** Å^3 **Default:** the current value for this material
STN.FAC	number	The strain factor that is used in the strain dependent diffusivity enhancement model **Units:** electron volts/strain **Default:** the current value for this material
DIP.0	number	The pre-exponential constant for diffusion with positively charged interstitials. **Units:** microns^2/min or cm^2/sec **Default:** the current value for this material
DIP.E	number	The activation energy for diffusion with positively charged interstitials. **Units:** electron volts **Default:** the current value for this material
DIX.0	number	The pre-exponential constant for diffusion with neutral interstitials. **Units:** microns^2/min or cm^2/sec **Default:** the current value for this material
DIX.E	number	The activation energy for diffusion with neutral interstitials. **Units:** electron volts **Default:** the current value for this material
DIM.0	number	The pre-exponential constant for diffusion with singly negative interstitials. **Units:** microns^2/min or cm^2/sec **Default:** the current value for this material
DIM.E	number	The activation energy for diffusion with singly negative interstitials. **Units:** electron volts **Default:** the current value for this material
DIMM.0	number	The pre-exponential constant for diffusion with doubly-negative interstitials. **Units:** microns^2/min or cm^2/sec **Default:** the current value for this material

Parameter	Type	Definition
DIMM.E	number	The activation energy for diffusion with doubly-negative interstitials. **Units:** electron volts **Default:** the current value for this material
DVP.0	number	The pre-exponential constant for diffusion with positively charged vacancies. **Units:** microns^2/min or cm^2/sec **Default:** the current value for this material
DVP.E	number	The activation energy for diffusion with positively charged vacancies. **Units:** electron volts **Default:** the current value for this material
DVX.0	number	The pre-exponential constant for diffusion with neutral vacancies. **Units:** microns^2/min or cm^2/sec **Default:** the current value for this material
DVX.E	number	The activation energy for diffusion with neutral vacancies. **Units:** electron volts **Default:** the current value for this material
DVM.0	number	The pre-exponential constant for diffusion with singly negative vacancies. **Units:** microns^2/min or cm^2/sec **Default:** the current value for this material
DVM.E	number	The activation energy for diffusion with singly negative vacancies. **Units:** electron volts **Default:** the current value for this material
DVMM.0	number	The pre-exponential constant for diffusion with doubly-negative vacancies. **Units:** microns^2/min or cm^2/sec **Default:** the current value for this material
DVMM.E	number	The activation energy for diffusion with doubly-negative vacancies. **Units:** electron volts **Default:** the current value for this material
DIPAIR.0	number	The pre-exponential constant for the diffusivity of dopant-interstitial pairs. **Units:** microns^2/min or cm^2/sec **Default:** the current value for this material
DIPAIR.E	number	The activation energy for the diffusivity of dopant-interstitial pairs. **Units:** electron volts **Default:** the current value for this material
DVPAIR.0	number	The pre-exponential constant for the diffusivity of dopant-vacancy pairs. **Units:** microns^2/min or cm^2/sec **Default:** the current value for this material
DVPAIR.E	number	The activation energy for the diffusivity of dopant-vacancy pairs. **Units:** electron volts **Default:** the current value for this material

Parameter	Type	Definition
`C.STATE`	number	The point defect charge state to which the `DIC.0`, `DIC.E`, `DVC.0`, and `DVC.E` parameters apply. The value must be in the range of -6 to +6. **Units:** none **Default:** none
`DIC.0`	number	The pre-exponential constant for diffusion with interstitials in the charge state `C.STATE`. **Units:** none **Default:** the current value for this material
`DIC.E`	number	The activation energy for diffusion with interstitials in the charge state `C.STATE`. **Units:** electron volts **Default:** the current value for this material
`DVC.0`	number	The pre-exponential constant for diffusion with vacancies in the charge state `C.STATE`. **Units:** none **Default:** the current value for this material
`DVC.E`	number	The activation energy for diffusion with vacancies in the charge state `C.STATE`. **Units:** electron volts **Default:** the current value for this material
`R.I.S`	number	The capture radius for the reaction between interstitials and substitutional dopant atoms. **Units:** Å **Default:** the current value for this material
`E.I.S`	number	The barrier energy for the reaction between interstitials and substitutional dopant atoms. **Units:** electron volts **Default:** the current value for this material
`R.V.S`	number	The capture radius for the reaction between vacancies and substitutional dopant atoms. **Units:** Å **Default:** the current value for this material
`E.V.S`	number	The barrier energy for the reaction between vacancies and substitutional dopant atoms. **Units:** electron volts **Default:** the current value for this material
`R.IP.V`	number	The capture radius for the reaction between dopant-interstitial pairs and vacancies. **Units:** Å **Default:** the current value for this material

Parameter	Type	Definition
`E.IP.V`	number	The barrier energy for the reaction between dopant-interstitial pairs and vacancies. **Units:** electron volts **Default:** the current value for this material
`R.VP.I`	number	The capture radius for the reaction between dopant-vacancy pairs and interstitials. **Units:** Å **Default:** the current value for this material
`E.VP.I`	number	The barrier energy for the reaction between dopant-vacancy pairs and interstitials. **Units:** electron volts **Default:** the current value for this material
`IP.V.LOW`	logical	Assume that only dopant-interstitial pairs and vacancies having opposite charges of equal magnitude react in bulk material. **Default:** true
`IP.V.MED`	logical	Assumes that uncharged dopant-interstitial pairs (or vacancies) can react with vacancies (or dopant-interstitial pairs) in any charged state, and that oppositely charged dopant-interstitial pairs and vacancies can react. **Default:** false
`IP.V.HIG`	logical	Assumes that dopant-vacancy pairs in any charged state can react with interstitials in any charged state. **Default:** false
`VP.I.LOW`	logical	Assumes that only dopant-vacancy pairs and interstitials having opposite charges of equal magnitude react in bulk material. **Default:** true
`VP.I.MED`	logical	Assumes that uncharged dopant-vacancy pairs (or interstitials) can react with interstitials (or dopant-vacancy pairs) in any charged state, and that oppositely charged dopant-vacancy pairs and interstitials can react. **Default:** false
`VP.I.HIG`	logical	Assumes that dopant-vacancy pairs in any charged state can react with interstitials in any charged state. **Default:** false
`EQ.FRACA`	logical	Specifies to set the dopant-interstitial pair and dopant-vacancy pair concentrations to local equilibrium in crystalline regrowth region of silicon amorphized by implant damage. **Default:** false
`IP.FRACA`	number	The fraction of the dopant-interstitial pair concentration to the active concentration in crystalline regrowth region of silicon amorphized by implant damage. **Units:** none **Default:** 0

Parameter	Type	Definition
VP.FRACA	number	The fraction of the dopant-vacancy pair concentration to the active concentration in crystalline regrowth region of silicon amorphized by implant damage. **Units:** none **Default:** 0
S.FRACA	number	The fraction of the substitutional atom concentration to the active concentration in crystalline regrowth region of silicon amorphized by implant damage. **Units:** none **Default:** 1
C.I.ALL	logical	Specifies that the parameter values for an interstitial are to be applied for all charge states of the interstitial. **Default:** true
C.INTERS	number	The charge state of an interstitial **Units:** none **Default:** 0
C.V.ALL	logical	Specifies that the parameter values for a vacancy are to be applied for all charge states of the vacancy. **Default:** true
C.VACANC	number	The charge state of a vacancy. **Units:** none **Default:** 0
C.IP.ALL	logical	Specifies that the parameter values for a dopant-interstitial pair are to be applied to all charge states of the dopant-interstitial pair. **Default:** true
C.IPAIR	number	The charge state of a dopant-interstitial pair. **Units:** none **Default:** 0
C.VP.ALL	logical	Specifies that the parameter values for a dopant-vacancy pair are to be applied to all charge states of the dopant-vacancy pair. **Default:** true
C.VPAIR	number	The charge state of a dopant-vacancy pair. **Units:** none **Default:** 0
SS.CLEAR	logical	Clears the solid solubility vs. temperature table. **Default:** false
SS.TEMP	number	The temperature at which the solid solubility in the material is **SS.CONC**. **Units:** ° Celsius **Default:** none
SS.CONC	number	The solid solubility in the material at temperature **SS.TEMP**. **Units:** atoms/cm^3 **Default:** none

Parameter	Type	Definition
CTN.0	number	The pre-exponential constant for impurity clustering. **Units:** (atoms/cm^3)$^{(1/\mathbf{CTN.F}-1)}$ **Default:** the current value for this material
CTN.E	number	The activation energy for impurity clustering. **Units:** electron volts **Default:** the current value for this material
CTN.F	number	The exponent of concentration for impurity clustering. **Units:** none **Default:** the current value for this material
CL.INI.A	logical	Specifies that impurities in an amorphized region are initially clustered when **ACT.TRAN** or **ACT.FULL** are used. **Default:** the current value for this impurity and material; initially true for built-in impurities in silicon
DDC.F.0	number	The pre-exponential constant for dopant-interstitial clustering. **Units:** none **Default:** the current value for this material
DDC.F.E	number	The activation energy for dopant-interstitial clustering. **Units:** electron volts **Default:** the current value for this material
DDC.T.0	number	The pre-exponential constant for the dopant-interstitial cluster dissolution time. **Units:** seconds **Default:** the current value for this material
DDC.T.E	number	The activation energy for the dopant-interstitial cluster dissolution time. **Units:** electron volts **Default:** the current value for this material
DDCF.D.N	number	The power of the dopant concentration in the dopant-interstitial cluster formation reaction. **Units:** none **Default:** the current value for this material
DDCF.N.N	number	The power of the electron concentration in the dopant-interstitial cluster formation reaction. **Units:** none **Default:** the current value for this material
DDCF.I.N	number	The power of the interstitial concentration in the dopant-interstitial cluster formation reaction. **Units:** none **Default:** the current value for this material
DDCR.N.N	number	The power of the electron concentration in the dopant-interstitial cluster dissolution reaction. **Units:** none **Default:** the current value for this material

Parameter	Type	Definition
`DDCR.I.N`	number	The power of the interstitial concentration in the dopant-interstitial cluster dissolution reaction. **Units:** none **Default:** the current value for this material
`IFRACM`	number	The number of interstitials per a dopant atom trapped in dopant-interstitial clusters. **Units:** none **Default:** the current value for this material
`DDCS.0`	number	The pre-exponential constant for the dopant contained in small dopant-interstitial clusters at equilibrium. **Units:** none **Default:** the current value for this material
`DDCS.E`	number	The activation energy for the dopant contained in small dopant-interstitial clusters at equilibrium. **Units:** electron volts **Default:** the current value for this material
`DDCS.N`	number	The number of dopant atoms contained in each small dopant-interstitial cluster. Only values of 1 and 2 are meaningful, and only the value in silicon is used. **Units:** none **Default:** the current value for this material
`IFRACS`	number	The number of interstitials per dopant atom trapped in small dopant-interstitial clusters. **Units:** none **Default:** the current value for this material
`DVC.F.0`	number	The pre-exponential constant for dopant-vacancy clustering. **Units:** none **Default:** the current value for this material
`DVC.F.E`	number	The activation energy for dopant-vacancy clustering. **Units:** electron volts **Default:** the current value for this material
`DVC.T.0`	number	The pre-exponential constant for the dopant-vacancy cluster dissolution time. **Units:** seconds **Default:** the current value for this material
`DVC.T.E`	number	The activation energy for the dopant-vacancy cluster dissolution time. **Units:** electron volts **Default:** the current value for this material
`DVCF.D.N`	number	The power of the dopant concentration in the dopant-vacancy cluster formation reaction. **Units:** none **Default:** the current value for this material

Parameter	Type	Definition
`DVCF.N.N`	number	The power of the electron concentration in the dopant-vacancy cluster formation reaction. **Units:** none **Default:** the current value for this material
`DVCF.V.N`	number	The power of the interstitial concentration in the dopant-vacancy cluster formation reaction. **Units:** none **Default:** the current value for this material
`DVCR.N.N`	number	The power of the electron concentration in the dopant-vacancy cluster dissolution reaction. **Units:** none **Default:** the current value for this material
`DVCR.V.N`	number	The power of the interstitial concentration in the dopant-vacancy cluster dissolution reaction. **Units:** none **Default:** the current value for this material
`VFRACM`	number	The number of vacancies per a dopant atom trapped in dopant-vacancy clusters. **Units:** none **Default:** the current value for this material
`C0.TF.0`	number	The prefactor in the expression for the activation time constant for the precipitation of this impurity in the **`ACT.FULL`** model. **Units:** seconds **Default:** the current value for this impurity.
`C0.TF.E`	number	The activation energy in the expression for the activation time constant for the precipitation of this impurity in the **`ACT.FULL`** or **`DDC.FULL`** model. **Units:** electron volts **Default:** the current value for this impurity.
`C0.TR.0`	number	The prefactor in the expression for the activation time constant for the dissolution of this impurity precipitation in the **`ACT.FULL`** or **`DDC.FULL`** model. **Units:** seconds **Default:** the current value for this impurity.
`C0.TR.E`	number	The activation energy in the expression for the activation time constant for the dissolution of this impurity precipitation in the **`ACT.FULL`** or **`DDC.FULL`** model. **Units:** electron volts **Default:** the current value for this impurity.
`DDC.L`	character	The solution name of the dopant-defect cluster in the left-hand side (reactant) of the reaction in the **`DDC.FULL`** model. **Default:** none
`DDC.R`	character	The solution name of the dopant-defect cluster in the right-hand side (product) of the reaction in the **`DDC.FULL`** model. **Default:** none **Synonym:** **`DDCLUSTE`**

Parameter	Type	Definition
I.DDC	logical	Specifies that the dopant-defect cluster **DDC.R** is generated by the reaction with an silicon interstitial in the **DDC.FULL** model. **Default:** false
IP.DDC	logical	Specifies that the dopant-defect cluster **DDC.R** is generated by the reaction with a dopant-interstitial pair in the **DDC.FULL** model. **Default:** false
V.DDC	logical	Specifies that the dopant-defect cluster **DDC.R** is generated by the reaction with a vacancy in the **DDC.FULL** model. **Default:** false
VP.DDC	logical	Specifies that the dopant-defect cluster **DDC.R** is generated by the reaction with a dopant-vacancy pair in the **DDC.FULL** model. **Default:** false
DDC.KF.0	number	The prefactor of the dopant-defect clustering forward reaction rate in the **DDC.FULL** model. **Units:** cm^3/sec **Default:** the current value for this impurity
DDC.KF.E	number	The activation energy of the dopant-defect clustering forward reaction rate in the **DDC.FULL** model. **Units:** electron volts **Default:** the current value for this impurity
DDC.KR.0	number	The prefactor of the dopant-defect clustering reverse reaction rate in the **DDC.FULL** model. **Units:** 1/sec **Default:** the current value for this impurity
DDC.KR.E	number	The activation energy of the dopant-defect clustering reverse reaction rate in the **DDC.FULL** model. **Units:** electron volts **Default:** the current value for this impurity
DDC.EQ.0	number	The prefactor of the equilibrium ratio of dopant-defect clustering reaction rate in the **DDC.FULL** model. **Units:** $atoms/cm^3$ **Default:** the current value for this impurity
DDC.EQ.E	number	The activation energy of the equilibrium ratio of dopant-defect clustering reaction rate in the **DDC.FULL** model. **Units:** electron volts **Default:** the current value for this impurity
DDCFRACC	number	The fraction of **DDC.R** or **DDC.L** to the tot al dopant-defect clusters in a crystalline region after implants in the **DDC.FULL** model. **Default:** the current value for the dopant-defect cluster
DDCFRACA	number	The fraction of **DDC.R** or **DDC.L** to the total dopant-defect clusters in a recrystallized region after implants in the **DDC.FULL** model. **Default:** the current value for the dopant-defect cluster

Parameter	Type	Definition
DDC.C.IP	number	The charge state of the second dopant-interstitial pair for the reaction, $XI + XI \leftrightarrow X_2I_2$. in the **DDC.FULL** model **Units:** none **Default:** 0
DDC.C.VP	number	The charge state of the second dopant-vacancy pair for the reaction, $XV + XV \leftrightarrow X_2V_2$. in the **DDC.FULL** model **Units:** none **Default:** 0
DDCPRINT	logical	Specifies to print all dopant-defect clustering reactions and their reaction parameter values for this impurity in the **DDC.FULL** model. **Default:** false
C1FRAC.A	number	The fraction of dopant-interstitial cluster to the whole clusters, i.e., dopant-interstitial clusters and precipitates upon implantation. Applies only inside of the amorphized regions whenever either **ACT.FULL** model or the **DDC.FULL** model is used. The value must be set in the range from 0.0 to 1.0. **Units:** none **Default:** 0
C2FRAC.A	number	The fraction of dopant-vacancy cluster to the whole clusters, i.e., dopant-interstitial clusters and precipitates upon implantation. Applies only inside of the amorphized regions whenever either **ACT.FULL** model or the **DDC.FULL** model is used. The value must be set in the range from 0.0 to 1.0. **Units:** none **Default:** 0
C0.INI.F	number	The fraction of dopant in precipitation upon implantation. Applies only outside of the amorphized regions whenever either **ACT.FULL** model or **DDC.FULL** model is used. Inside the amorphized regions, **ACT.AMOR** value determines the activation level. The value must be set in the range from 0.0 to 1.0. **Units:** none **Default:** 0
C1.INI.F	number	The fraction of dopant that is clustered upon implantation by the dopant-interstitial clustering model. Applies only outside of the amorphized regions whenever either **ACT.FULL** model or **DDC.FULL** model is used. Inside the amorphized regions, **ACT.AMOR** value determines the activation level. The value must be set in the range from 0.0 to 1.0. **Units:** none **Default:** 0 **Synonym: INI.DDC**

Parameter	Type	Definition
C2.INI.F	number	The fraction of dopant that is clustered upon implantation by the dopant-vacancy clustering model. Applies only outside of the amorphized regions whenever either **ACT.FULL** model or **DDC.FULL** model is used. Inside the amorphized regions, **ACT.AMOR** value determines the activation level. The value must be set in the range from 0.0 to 1.0. **Units:** none **Default:** 0 **Synonym: INI.DDC**
DDCTOT.C	number	The fraction of the total dopant-defect clusters including C1,C2 and all DDCs to the total dopants in a crystalline region after implants. Required when the **DDC.FULL** model is used or both C1 and C2 exist in the **ACT.FULL** model. **Default:** the current value for the dopant-defect cluster and the material
DDCTOT.A	number	The fraction of the total dopant-defect clusters to the total clusters in a recrystallized region after implants. Required when the **DDC.FULL** model is used or both C1 and C2 exist in the **ACT.FULL** model. **Default:** the current value for the dopant-defect cluster and the material
ACT.NI	number	The factor by which the intrinsic carrier concentration is multiplied to obtain the minimum activation level used by the **ACT.FULL** model for this impurity. **Units:** none **Default:** the current value for this impurity.
ACT.AMOR	number	The threshold concentration above which the dopant is inactive for the **ACT.FULL** dopant activation model at the beginning of the post-implant anneal. Applies only inside the amorphized regions. Outside of the amorphized regions - see **C0.INI.F**, **C1.INI.F** parameters. **Units:** cm^{-3} **Default:** the current value for this impurity
MAX.AMOR	number	Maximum impurity concentration that can survive epitaxial recrystallization of the amorphized layer. Everything above this concentration is swept by the advancing recrystallization front. **Units:** cm^{-3} **Default:** 6e18 for indium and 5e22 for the other impurities **Synonym: MAX.SPE**
FGB	number	The factor by which the diffusivity is increased in polycrystalline materials when the polycrystalline model is not enabled. **Units:** none **Default:** the current value for this material
Q.SITES	number	The density of dopant sites in grain boundaries (the maximum allowable value for the grain-boundary concentration) in a polycrystalline material. **Units:** $\#/cm^2$ **Default:** the current value for this material
CG.MAX	number	The density of dopant sites in the grain interior of a polycrystalline material. **Units:** $\#/cm^3$ **Default:** the current value for this material; initially 5e22 for polysilicon

Parameter	Type	Definition
`GSEG.0`	number	The entropy for segregation between grain interior and boundaries in a polycrystalline material. **Units:** none **Default:** the current value for this material **Synonym: `A.SEG`**
`GSEG.E`	number	The activation energy of the segregation entropy between grain interior and boundaries in a polycrystalline material. **Units:** electron volts **Default:** the current value for this material **Synonym: `E.SEG`**
`GSEG.INI`	number	The entropy for the initial segregation between grain interior and boundaries in a polycrystalline material. **Units:** none **Default:** the current value for this material **Synonym: `A0.SEG`**
`FVELB.0`	number	The pre-exponential factor of the multiplication factor to the segregation velocity in the grain interiors of a polycrystalline material. **Units:** none **Default:** the current value for this material
`FVELB.E`	number	The activation energy of the multiplication factor to the segregation velocity in the grain interiors of a polycrystalline material. **Units:** electron volts **Default:** the current value for this material
`VELIF.0`	number	The pre-exponential factor for the segregation velocity at the boundaries of a polycrystalline material. **Units:** cm/sec **Default:** the current value for this material
`VELIF.E`	number	The activation energy for the segregation velocity at the boundaries of a polycrystalline material. **Units:** electron volts **Default:** the current value for this material
`MODEL`	character	The name of the model to which the diffusion equations for the new impurity are registered. The **`ENABLE`** parameter, with the **`MODEL`** parameter given by this name in the **`METHOD`** statement, specifies the diffusion equation for this impurity to be solved. **Default:** none **Synonym**: **`D.MODEL`**
`EQACTIVE`	character	The expression of calculation for the equilibrium active concentration of the dopant. The equilibrium active concentration of the dopant in the given material is to be calculated by the **`EQACTIVE`** expression instead of calling the built-in function. **Default:** none

Parameter	Type	Definition
`ACTIVE`	character	The expression of calculation for the active concentration of the dopant. The active concentration of the dopant in the given material is to be calculated by the **`ACTIVE`** expression instead of calling the built-in function. **Default:** none
`MOBILE`	character	The expression of calculation for the mobile concentration of the dopant. The mobile concentration of the dopant in the given material is to be calculated by the **`MOBILE`** expression instead of calling the built-in function. **Default:** none
`DI.FAC`	character	The expression of multiplication factor to the diffusion with interstitials. **Default: "**1.0"
`DV.FAC`	character	The expression of multiplication factor to the diffusion with vacancies. **Default:** "1.0"
`DI.F11`	character	The expression of multiplication factor to the horizontal diffusion with interstitials. **Default:** "1.0"
`DI.F22`	character	The expression of multiplication factor to the vertical diffusion with interstitials. **Default:** "1.0"
`DV.F11`	character	The expression of multiplication factor to the horizontal diffusion with vacancies. **Default:** "1.0"
`DV.F22`	character	The expression of multiplication factor to the vertical diffusion with vacancies. **Default:** "1.0"
`/MATERIA`	character	The name of material 2 for the segregation and transport parameters. **Default:** none
`SEG.0`	number	The pre-exponential factor for segregation from material 1 to material 2. For the trapped impurity at interface, the pre-exponential factor for segregation from material 1 to interface adjacent to material 2. **Units:** none for the segregation from material 1 to material 2, cm^{-1} for the segregation from material 1 to interface **Default:** the current value for these materials
`SEG.E`	number	The activation energy for segregation from material 1 to material 2. For the trapped impurity at interface, the activation energy for segregation from material 1 to interface adjacent to material 2. **Units:** electron volts **Default:** the current value for these materials
`TRANS.0`	number	The pre-exponential factor for transport from material 1 to material 2. For the trapped impurity at interface, the pre-exponential factor for transport from material 1 to interface adjacent to material 2. **Units:** microns/min or cm/sec **Default:** the current value for these materials **Synonym: `TRN.0`**

Parameter	Type	Definition
`TRANS.E`	number	The activation energy for transport from material 1 to material 2. For the trapped impurity at interface, the activation energy for transport from material 1 to interface adjacent to material 2. **Units:** electron volts **Default:** the current value for these materials **Synonym:** **`TRN.E`**
`RATIO.0`	number	The pre-exponential factor for ratio of detrapping rate to trapping rate for material 1 at interface adjacent to material 2. Used only with the **`ITRAP`** model. **Units:** none **Default:** the current value for these materials
`RATIO.E`	number	The activation energy for ratio of detrapping rate to trapping rate for material 1 at interface adjacent to material 2. Used only with the **`ITRAP`** model. **Units:** none **Default:** the current value for these materials
`SEG.SS`	logical	Specifies that the segregation from material 1 to interface adjacent to material 2 is calculated from solid solubility instead of **`SEG.0`** and **`SEG.E`**. Used only with the **`ITRAP`** model. **Default:** false
`/SEG.0`	number	The pre-exponential factor for segregation from material 2 to interface adjacent to material 1 for the trapped impurity at interface. Used only with the **`ITRAP`** model. **Units:** cm^{-1} **Default:** the current value for these materials
`/SEG.E`	number	The activation energy for segregation from material 2 to interface adjacent to material 1 for the trapped impurity at interface. Used only with the **`ITRAP`** model. **Units:** electron volts **Default:** the current value for these materials
`/TRANS.0`	number	The pre-exponential factor for transport from material 2 to interface adjacent to material 1 for the trapped impurity at interface. Used only with the **`ITRAP`** model. **Units:** microns/min or cm/sec **Default:** the current value for these materials **Synonym:** **`/TRN.0`**
`/TRANS.E`	number	The activation energy for transport from material 2 to interface adjacent to material 1 for the trapped impurity at interface. Used only with the **`ITRAP`** model. **Units:** electron volts **Default:** the current value for these materials **Synonym:** **`/TRN.E`**
`/RATIO.0`	number	The pre-exponential factor for ratio of detrapping rate to trapping rate for material 2 at interface adjacent to material 1. Used only with the **`ITRAP`** model. **Units:** none **Default:** the current value for these materials

Parameter	Type	Definition
`/RATIO.E`	number	The activation energy for ratio of detrapping rate to trapping rate for material 2 at interface adjacent to material 1. Used only with the **`ITRAP`** model. **Units:** none **Default:** the current value for these materials
`/SEG.SS`	logical	Specifies that the segregation from material 2 to interface adjacent to material 1 is calculated from solid solubility instead of **`/SEG.0`** and **`/SEG.E`**. Used only with the **`ITRAP`** model. **Default:** false
`SEG.EQ3`	logical	Specifies that both of segregations from material 1 to interface and from material 2 to interface are determined from the values of the parameters, **`SEG.0`**, **`SEG.E`**, **`/SEG.0`** and **`/SEG.E`**. Used only with the **`ITRAP`** model. **Default:** true
`SEG.EQ2`	logical	Specifies that the segregation from material 1 to interface is determined to make the concentration ratio in materials 1 and 2 in 3-phase equilibrium equal to the one in 2-phase equilibrium. Used only with the **`ITRAP`** model. **Default:** false
`/SEG.EQ2`	logical	Specifies that the segregation from material 2 to interface is determined to make the concentration ratio in materials 1 and 2 in 3-phase equilibrium equal to the one in 2-phase equilibrium. Used only with the **`ITRAP`** model. **Default:** false
`Q.INI.0`	number	The pre-exponential factor for the initial occupied trap density at interface. Used only with the **`ITRAP`** model. **Units:** atoms/cm^2 **Default:** the current value for these materials
`Q.INI.E`	number	The activation energy for the initial occupied trap density at interface. Used only with the **`ITRAP`** model. **Units:** electron volts **Default:** the current value for these materials
`Q.MAX.0`	number	The pre-exponential factor for the maximum trap density at interface. Used only with the **`ITRAP`** model. **Units:** atoms/cm^2 **Default:** the current value for these materials
`Q.MAX.E`	number	The activation energy for the maximum trap density at interface. Used only with the **`ITRAP`** model. **Units:** electron volts **Default:** the current value for these materials
`TWO.PHAS`	logical	Specifies that the flux of 2-phase segregation is added to the one of 3-phase segregation flux. Used only with the **`ITRAP`** model. **Default:** false
`KSIP.0`	number	The pre-exponential constant for the surface recombination velocity of the dopant-interstitial pair at the interface between the specified materials. **Units:** microns/min or cm/sec **Default:** the current value for these materials

Parameter	Type	Definition
`KSIP.E`	number	The activation energy for the surface recombination velocity of the dopant-interstitial pair at the interface between the specified materials **Units:** electron volts **Default:** the current value for these materials
`KSVP.0`	number	The pre-exponential constant for the surface recombination velocity of the dopant-vacancy pair at the interface between the specified materials. **Units:** microns/min or cm/sec **Default:** the current value for these materials
`KSVP.E`	number	The activation energy for the surface recombination velocity of the dopant-vacancy pair at the interface between the specified materials **Units:** electron volts **Default:** the current value for these materials
`SEG.IFRA`	number	The fraction of a dopant-interstitial pairing to the dopant-defect pairing for pair segregation model. The value must be >= 0 and <= 1.0. Only applicable when the **`PAIR.SEG`** parameter is turned on in the **`METHOD`** statement. **Units:** none **Default:** the current value for these materials
`SEG.IPOW`	number	The power of interstitial concentration to which the segregation transport rate is proportional. The parameter value physically means the number of interstitials to be generated or recombined during segregation. Only applicable when the **`PAIR.SEG`** parameter is turned on in the **`METHOD`** statement. **Units:** none **Default:** the current value for these materials
`SEG.VPOW`	number	The power of vacancy concentration to which the segregation transport rate is proportional. The parameter value physically means the number of vacancies to be generated or recombined during segregation. Only applicable when the **`PAIR.SEG`** parameter is turned on in the **`METHOD`** statement. **Units:** none **Default:** the current value for these materials
`SEG.CPOW`	number	The power of n/n_i in the Fermi-level dependent segregation model. The parameter value physically means the number of charge carriers to be generated or annihilated during segregation. **Units:** none **Default:** the current value for these materials
`ES.RAND`	number	The electronic stopping power coefficient of implanted atoms in the specified material for materials other than silicon and for a nonchanneled direction in silicon. This value is used for the MC ion implant calculation only. **Units:** angstrom^2*eV$^{(1-\mathtt{ES.F.RAN})}$ **Default:** the current value for the specified materials
`ES.F.RAN`	number	The exponent of the electronic stopping power of implanted atoms in the specified material for materials other than silicon and for a nonchanneled direction in silicon. This value is used for the MC ion implant calculation only. **Units:** none **Default:** the current value for the specified materials

Parameter	Type	Definition
ES.BREAK	number	The energy above which **ES.F.H** is used, instead of **ES.F.RAN**, **ES.F.100**, or **ES.F.110**, in calculating the electronic stopping power coefficient of implanted atoms. This value is used for the MC ion implant calculation only. **Units:** keV **Default:** the current value for the specified materials
ES.F.H	number	The exponent of the electronic stopping power of implanted atoms at energies above **ES.BREAK**. This value is used for the MC ion implant calculation only. **Units:** none **Default:** the current value for the specified materials
ES.100	number	The electronic stopping power for implanted atoms in silicon along the <100> channeling axes. This value is used for the MC ion implant calculation only. **Units:** angstrom^2*eV$^{(1-\mathbf{ES.F.100})}$ **Default:** the current value for the specified materials
ES.F.100	number	The exponent of the electronic stopping power for implanted atoms in silicon along the <100> crystal axes. This value is used for the MC ion implant calculation only. **Units:** none **Default:** the current value for the specified material
ES.110	number	The electronic stopping power for implanted atoms in silicon along the <110> channeling axes. This value is used for the MC ion implant calculation only. **Units:** angstrom^2*eV$^{(1-\mathbf{ES.F.100})}$ **Default:** the current value for the specified material
ES.F.110	number	The exponent of the electronic stopping power for implanted atoms in silicon along the <110> crystal axes. This value is used for the MC ion implant calculation only. **Units:** none **Default:** the current value for the specified material
NLOC.PRO	number	The prefactor for the nonlocal electronic stopping power formula. This value is used for the OLD.MC ion implant calculation only. **Units:** none **Default:** 0.3
NLOC.EXO	number	The exponent for the nonlocal electronic stopping power formula. This value is used for the OLD.MC ion implant calculation only. **Units:** none **Default:** 0.2
NLOC.MAX	number	The maximum value allowed for the nonlocal part of electronic stopping power. This value is used for the OLD.MC ion implant calculation only. **Units:** none **Default:** 0.6
NLOC.K	number	A correction factor for the LSS electron stopping power. This value is used for the OLD.MC ion implant calculation only. **Units:** none **Default:** 1.2

Parameter	Type	Definition
LOC.FAC	number	A correction factor for the local part of electronic stopping power. This value is sued for the OLD.MC ion implant calculation only. **Units:** none **Default:** 0.8
CHAN.CRI	number	Specifies the critical angle below which the scattering angle is automatically reduced. This value is used for the OLD.MC ion implant calculation only. **Units:** none **Default:** 11.54
CHAN.FAC	number	Specifies the factor by which the scattering angle is reduced. This value is used for the OLD.MC ion implant calculation only. **Units:** none **Default:** 2.0
DISP.FAC	number	This is a correction factor for the probability of selecting the amorphous model. This value is used for the OLD.MC ion implant calculation only. **Units:** none **Default:** 0.5
DIV.ANG	number	Angular divergence angle of ion beam. This parameter is used for Taurus MC only. **Units:** degree **Default:** 0.5
LSS.PRE	number	A correction factor for the LSS electron stopping power. This parameter is used for Taurus MC only. **Units:** none **Default:** 1.0
NLOC.PRE	number	The prefactor for the nonlocal electronic stopping power formula. This parameter is used for Taurus MC only. **Units:** none **Default:** 0.5
NLOC.EXP	number	The exponent for the nonlocal electronic stopping power formula. This parameter is used for Taurus MC only. **Units:** none **Default:** 0.5
SCR.PAR	number	Adjustable parameter for screening length of local electronic stopping power. **Units:** none **Default:** 1.0
SURV.RAT	number	Survival rate of Frenkel pairs generated by the ion beam for the default **PLUS** damage model. This parameter is used for Taurus MC only. **Units:** none **Default:** 1.0

Parameter	Type	Definition
AMOR.PAR	number	Proportional factor for the switching probability from crystalline to amorphous model for the default **PLUS** damage model. This parameter is used for Taurus MC only. **Units:** none **Default:** 1.0
SAT.PAR	number	Damage saturation parameter for MC implant. The saturation value is equal to (lattice density)**/SAT.PAR**. This parameter is used for Taurus MC only. **Units:** none **Default:** 1.0
DISP.THR	number	Displacement threshold for the default **PLUS** damage model. This parameter is used for Taurus MC only. **Units:** eV **Default:** 15
CASC.SUR	number	Survival rate of Frenkel pairs generated by the ion beam for **CASCADE** damage model. This parameter is used for Taurus MC only. **Units:** none **Default:** 1.0
CASC.AMO	number	Proportional factor for the switching probability from crystalline to amorphous model for the **CASCADE** damage model. This parameter is used for Taurus MC only. **Units:** none **Default:** 1.0
CASC.SAT	number	Damage saturation parameter for cascade damage model.The saturation value is equal to (lattice density)**/CASC.SAT**. This parameter is used for Taurus MC only. **Units**: none **Default**: 1.0
CASC.DIS	number	Displacement threshold for **CASCADE** damage model. This parameter is used for Taurus MC only. **Units:** eV **Default:** 15
REC.FRAC	number	Fraction of secondary recoils to be traced for **CASCADE** damage model. For better accuracy, 1.0 should be used. **Units:** none **Default:** 1.0

Description

This statement is used to define new impurities and specify their properties. If **NEW** is specified, a new impurity is defined; otherwise the properties of an existing impurity are set.

The coefficients for standard impurities are given in the *s4init* file (which is read at the start of each TSUPREM-4 execution), but you can change these at any time. Coefficients that are *not* given in the *s4init* file or set by the user, default to 0.0, except for **SEG.0**, which defaults to 1.0.

Parameters whose units include time are specified in units of microns and minutes, unless **CM.SEC** is true, in which case units of centimeters and seconds are assumed.

The **IMPURITY** statement replaces the **ANTIMONY**, **ARSENIC**, **BORON**, and **PHOSPHORUS** statements; the old statements are still available, however, for compatibility with existing input files.

Impurity Type

DONOR and **ACCEPTOR** specify whether the impurity is a donor or an acceptor. By default, newly defined impurities are electrically inactive. Once they have been specified to be active (with the **DONOR** or **ACCEPTOR** parameters), they cannot be returned to their electrically inactive state.

Solution Options

SOLVE specifies whether diffusion equations are solved for this impurity; if **SOLVE** is false, the impurity is assumed to be immobile. If the **MODEL** name is given, turning off the given model in **METHOD** statement is the same as turning off **SOLVE** in **IMPURITY** statement. Turning off **DIFFUSE** parameter specifies that the diffusion equation is not solved for this impurity in the specified material. **STEADY** specifies whether a steady-state or full transient solution to the diffusion equations should be computed. The steady-state solution is preferred for impurities that diffuse rapidly or react at material interfaces to produce material growth or consumption. The full transient solution is preferred for impurities that diffuse slowly or interact with other impurities.

Other Parameters

AT.NUM and **AT.WT** are used by the MC implantation model. **IMPL.TAB** specifies the name of the implant moment table to be used for the impurity.

The initial background concentration of a user-defined impurity when it is first added to a structure is given by **C.INIT**. For normal impurities, **C.INIT** should be greater than zero; a value of 1e5 (the value used for built-in impurities) is suggested.

Multiplication to Diffusivity

DI.FAC and **DV.FAC** describe the formulas for multiplication factors to diffusivities. **D.MODEL** names the **DI.FAC** and/or **DV.FAC** so that its application can be turned *on* or *off*. For example

```
IMPURITY IMP=BORON MAT=OXIDE MODEL=myDiff +
         DI.FAC=1+1e-21*fluorine
METHOD    MODEL=myDiff !ENABLE
```

Anisotropic Diffusivity

DI.F11, **DI.F22**, **DV.F11** and **DV.FAC** describe the formulas for multiplication factors to anisotropic diffusivities. For example,:

```
INTERMED NAME=qkT EXPR=1E31*KB*TEMP
INTERMED NAME=SD11 EXPR=exp(Sxx*15.73/qkT)
INTERMED NAME=SD22 EXPR=exp(Syy*15.73/qkT)
IMPURITY MODEL=AnisoSD IMP=BORON MAT=Si +
  DI.F11=SD11 DV.F11=SD11 DI.F22=SD22 DV.F22=SD22
```

SD11 and **SD22** assigned to **DI.F11**, **DI.F22**, **DV.F11** and **DV.F22** can be replaced with their expressions. However, using the intermediates reduces the calculation time when the same expressions are used in several places. Specifying the **MODEL** parameter is optional. It is useful to turn on/off the expressions by the **METHOD** statement.

For example, the above model can be turned off by;

```
METHOD MODEL=AnisoSD ^ENABLE
```

User-defined Active Concentration

By setting an expression to the **ACTIVE** parameter in the **IMPURITY** command, you can change the function to calculate the dopant-active concentration in the transient state. The parameter **EQACTIVE** in the **IMPURITY** command defines the expression to calculate the equilibrium active concentration. Calling the **EQACTIVE()** function returns the equilibrium active concentration. There are two types of **EQACTIVE()** function call:

- **EQACTIVE**(*imp,conc*)

  For the given ‘conc’ concentration, it calls the built-in function to calculate the equilibrium active concentration with the ’*imp*’ impurity parameters, and returns it.

- **EQACTIVE**(*imp*)

  It calls the function defined by the **EQACTIVE** parameter in the **IMPURITY** command, if available. Otherwise, the built-in function is used.

The following example shows how to implement the transient clustering model.

```
INTERMED NAME=C0EQ  EXPR=BORON-EQACTIVE(BORON)
INTERMED NAME=TDECL EXPR=8E-16*EXP(4.2/KT)
IMPURITY MODEL=BPREMODEL IMP=BORON MAT=SILICON +
         ACTIVE=BORON-C0_BORON +
         EQACTIVE=EQACTIVE(BORON,BORON)
EQUATION MODEL=BPREMODEL IMP=C0_BORON MAT=SILICON +
         ADDTOEXP=-(C0_BORON-C0EQ)/TDECL
```

See Also

For additional information, see the following sections:

- Use of the diffusion parameters, Chapter 3, Diffusion of Impurities, p. 3-18.
- Solid solubility and clustering parameters, Chapter 3, Activation of Impurities, p. 3-34.
- Segregation parameters, Chapter 3, Activation of Impurities, p. 3-34.
- Electronic stopping power parameters, Chapter 3, MC Ion Implant Model, p. 3-131.

Examples

1. The following statement specifies the diffusivity of antimony diffusing with neutral vacancies in silicon:

```
IMPURITY IMP=ANTIMONY MAT=SILICON DVX.0=1.22e9 +
         DVX.E=3.65
```

2. The following statement specifies the diffusivity of phosphorus with triply-negative interstitials:

```
IMPURITY IMP=P MAT=SILI C.STATE=-3 DIC.0=2e11 +
         DIC.E=4.37
```

3. The following statement specifies the segregation parameters at the Si/SiO$_2$ interface:

```
IMPURITY IMP=ANTIMONY MAT=SILICON /MAT=OXIDE +
         SEG.0=30.0 TRANS.0=0.1
```

   The concentration in silicon is 30.0 times the concentration in oxide, at equilibrium.

4. The following statements define a new impurity, and set some of its properties:

```
IMPURITY NEW IMPURITY=CESIUM AT.NUM=55 +
         AT.WT=132.9 C.INIT=1e5
IMPURITY IMP=CESIUM MAT=OXIDE DIX.0=0.5 +
         DIX.E=2.9
```

REACTION

The **`REACTION`** statement defines the reactions that occur at material interfaces.

```
REACTION

   { MAT.BULK=<c> | ( MAT.R=<c> /MAT.L=<c> ) }
   [MODEL=<c>] [NAME=<c>]
   [ { DELETE | REPLACE } ]
   { ( [IMP.L=<c>]   [NI.L=<n>]   [EI.L=<n>  ]
       [/IMP.L=<c>] [/NI.L=<n>] [/EI.L=<n> ]
       [IMP.R=<c>]   [NI.R=<n>]   [EI.R=<n>  ]
       [/IMP.R=<c>] [/NI.R=<n>] [/EI.R=<n> ]
       [NM.R=<n>]    [/NM.L=<n>]
       [RATE.0=<n>] [RATE.E=<n>] [R.FAC=<c>] [/R.FAC=<c>]
       [EQUIL.0=<n>] [EQUIL.E=<n>] [E.FAC=<c>] [/E.FAC=<c>]
     )
   | ( MAT.NEW=<c> THICKNES=<n> )
   }
```

Parameter	Type	Definition
`MAT.BULK`	character	A name of bulk material, in which the reaction equation is applied. **Default:** none
`MAT.R`	character	Material 1, which appears on the right side of the reaction equation. **Default:** none **Synonyms**: **`MAT, MAT1`**
`/MAT.L`	character	Material 2, which appears on the left side of the reaction equation. **Default:** none **Synonym**: **`MAT2`**
`MODEL`	character	A name used to refer to this reaction. By turning on/off the **`ENABLE`** parameter with the **`MODEL`** parameter in the **`METHOD`** statement, the reaction can be enabled or disabled. **Default:** none
`NAME`	character	A name used to refer to this reaction in subsequent **`REACTION`** statements. **Default:** “<none>”
`DELETE`	logical	Deletes the specified reaction. **Default:** false
`REPLACE`	logical	Replaces the specified reaction. **Default:** false
`IMP.L`	character	The impurity in material 1 that appears on the left side of the reaction equation. **Default:** none
`NI.L`	number	The number of molecules of **`IMP.L`** that participate in the reaction. **Units:** none **Default:** 1.0

Parameter	Type	Definition
`EI.L`	number	The exponent describing the dependence of the forward reaction rate on the concentration of `IMP.L`. **Units:** none **Default**: `NI.L`
`/IMP.L`	character	The impurity in material 2 that appears on the left side of the reaction equation. **Default:** none
`/NI.L`	number	The number of molecules of `/IMP.L` that participate in the reaction. **Units:** none **Default:** 1.0
`/EI.L`	number	The exponent describing the dependence of the forward reaction rate on the concentration of `/IMP.L`. **Units:** none **Default**: `/NI.L`
`IMP.R`	character	The impurity in material 1 that appears on the right side of the reaction equation. **Default:** none
`NI.R`	number	The number of molecules of `IMP.R` that participate in the reaction. **Units:** none **Default:** 1.0
`EI.R`	number	The exponent describing the dependence of the reverse reaction rate on the concentration of `IMP.R`. **Units:** none **Default**: `NI.R`
`/IMP.R`	character	The impurity in material 2 that appears on the right side of the reaction equation. **Default:** none
`/NI.R`	number	The number of molecules of `/IMP.R` that participate in the reaction. **Units:** none **Default:** 1.0
`/EI.R`	number	The exponent describing the dependence of the reverse reaction rate on the concentration of `/IMP.R`. **Units:** none **Default**: `/NI.R`
`NM.R`	number	The number of molecules of `MAT.R` that participate in the reaction. **Units:** none **Default:** 0.0
`/NM.L`	number	The number of molecules of `/MAT.L` that participate in the reaction. **Units:** none **Default:** 0.0
`RATE.0`	number	The prefactor in the expression for the forward reaction rate. **Units:** varies **Default:** 0.0

Parameter	Type	Definition
RATE.E	number	The activation energy for the forward reaction rate. **Units:** none **Default:** 0.0
R.FAC	character	A formula using solution variables defined in **MAT.R**, that multiplies the forward reaction rate. **Units:** none **Default:** 1.0
/R.FAC	character	A formula using solution variables defined in **/MAT.L**, that multiplies the forward reaction rate. **Units:** none **Default:** 1.0
EQUIL.0	number	The prefactor in the expression for the ratio of the reverse reaction rate to the forward reaction rate. **Units:** varies **Default:** 0.0
EQUIL.E	number	The activation energy for the ratio of the reverse reaction rate to the forward reaction rate. **Units:** none **Default:** 0.0
E.FAC	character	A formula using solution variables defined in **MAT.R**, that multiplies the ratio of the reverse reaction rate to the forward reaction rate. **Units:** none **Default:** 1.0
/E.FAC	character	A formula using solution variables defined in **/MAT.L**, that multiplies the ratio of the reverse reaction rate to the forward reaction rate. **Units:** none **Default:** 1.0
MAT.NEW	character	The name of the material to be added between layers of **MAT.R** and **/MAT.L**, when one is deposited on the other. **Default:** none
THICKNES	number	The thickness of **MAT.NEW** to be added between layers of **MAT.R** and **/MAT.L**, when one is deposited on the other. **Units:** microns **Default:** none

Description

The **REACTION** statement can specify either a reaction that takes place at the interface between two materials or the addition of a *native* layer when one material is deposited on another. Each reaction is identified by its name and the materials on either side of the interface. The name is optional: if no name is specified, the name “<none>” is assumed.

Reactions defined with the **REACTION** statement take place during **DIFFUSION** steps with either oxidizing or inert ambients. The reactions occur only when the **VERTICAL**, **COMPRESS**, or **VISCOELA** oxidation model has been selected.

Defining and Deleting

A new reaction is defined by specifying a name (optional), the materials on either side of the interface, and the impurity and reaction rate parameters. The parameters for a previously defined reaction can be changed by specifying the name and materials for the reaction along with any parameters that need to be changed. A previously defined reaction can be deleted with the **DELETE** parameter. The **REPLACE** parameter deletes a previously defined reaction then replaces it with a new reaction defined by the parameters on the **REACTION** statement.

Insertion of Native Layers

The **MAT.NEW** and **THICKNES** parameters specify a new material to be inserted between **MAT.R** and **/MAT.L** when one is deposited on the other. A layer of **MAT.NEW** with thickness **THICKNES** separates **MAT.R** and **/MAT.L**. The process is analogous to the insertion of a native oxide layer between exposed silicon and an oxidizing ambient.

Reaction Equation

The general form of the reaction is:

$$n_{il1}I_{l1} + n_{il2}I_{l2} + n_{ml2}M_{l2} \rightleftarrows n_{mr1}M_{r1} + n_{ir1}I_{r1} + n_{ir2}I_{r2}$$

Equation 4-5

where the subscripts l and r denote terms on the left and right sides of the reaction, and subscripts 1 and 2 refer to materials 1 and 2. The forward and reverse reaction rates are given by:

$$R_f = k_f[I_{l1}]^{e_{l1}}[I_{l2}]^{e_{l2}}$$

Equation 4-6

$$R_r = k_r[I_{r1}]^{e_{r1}}[I_{r2}]^{e_{r2}}$$

Equation 4-7

where $[I_x]$ denotes the concentration of impurity I_x.

Parameters

Specify the parameters of this reaction as follows:

I_{l1} = **IMP.L**, I_{l2} =/ **IMP.L**, I_{r1} = **IMP.R**, I_{r2} =/ **IMP.R**

Equation 4-8

n_{il1} = **NI.L**, n_{il2} =/ **NI.L**, n_{ir1} = **NI.R**, I_{ir2} =/ **NI.R**

Equation 4-9

n_{mr1} = **NM.R**, n_{ml2} =/ **NM.L**

Equation 4-10

e_{il1} = **EI.L**, e_{il2} =/ **EI.L**, e_{ir1} = **EI.R**, e_{ir2} =/ **EI.R**

Equation 4-11

$$k_f = \mathtt{RATE.0} \cdot \exp\left(\frac{-\mathtt{RATE.E}}{kT}\right) \cdot \mathtt{R.FAC} \cdot \mathtt{/R.FAC}$$

Equation 4-12

$$\frac{k_r}{k_f} = \frac{[I_{l1}]^{e_{l1}}[I_{l2}]^{e_{l2}}}{[I_{r1}]^{e_{r1}}[I_{r2}]^{e_{r2}}} = \mathtt{EQUIL.0} \cdot \exp\left(\frac{-\mathtt{EQUIL.E}}{kT}\right) \cdot \mathtt{E.FAC} \cdot \mathtt{/E.FAC}$$

Equation 4-13

- **NI.L**, **/NI.L**, **NI.R**, **/NI.R**, **NM.R**, and **/NM.L** determine the number of molecules of each reactant that participate in the reaction.
- **EI.L**, **/EI.L**, **EI.R**, **/EI.R**, **RATE.0**, **RATE.E**, **R.FAC**, **/R.FAC**, **EQUIL.0**, **EQUIL.E**, **E.FAC**, and **/E.FAC** determine the rate of the reaction.

In theory, the n_i and e_i for each reaction would be equal, but this is not required by TSUPREM-4. This allows the rate to depend on the concentration of an impurity without the concentration being affected by the reaction ($e_i \neq 0,\ n_i = 0$), or for the concentration to change without affecting the rate ($e_i = 0,\ n_i \neq 0$).

The units of R_f and R_r are #/cm^2/sec; the units of k_f and k_r depend on the values of the n_i and e_i.

Effects

This reaction has two effects on the simulation:

1. As a boundary condition for diffusing species, it can result in generation, consumption, or transport of impurities across a material interface. The impurity generation fluxes (F, in #/cm^2/sec) at the interface are given by

$$-\frac{1}{n_{il1}}F_{l1} = -\frac{1}{n_{il12}}F_{l2} = \frac{1}{n_{ir1}}F_{r1} = \frac{1}{n_{ir2}}F_{r2} = R_f - R_r$$

Equation 4-14

(*Note* that when n_i is zero, the corresponding flux is also zero.)

2. It can produce growth or consumption of material regions. This occurs for materials for which $n_m \neq 0$. The growth rate of M_{rl}, in cm/sec, is given by:

$$V = \frac{n_{mr1} \times \mathtt{MOL.WT}}{A \times \mathtt{DENSITY}}(R_f - R_r)$$

Equation 4-15

where V is the growth velocity in cm/sec, $A = 6.022 \times 10^{23}$ is Avogadro's number, and **MOL.WT** and **DENSITY** are material parameters specified on the **MATERIAL** statement. The material is consumed ($V < 0$) if $R_f < R_r$.

MOBILITY

The **`MOBILITY`** statement is used to define or modify the dependence of carrier mobility on impurity concentration and temperature within a semiconductor.

```
MOBILITY

   [ {TAB.TEMP=<n> [KELVIN]
     TAB.CONC=<c> TAB.E.MU=<c> TAB.H.MU=<c> [TAB.CLEA]} ]
   [ECN.MU=<n>] [ECP.MU=<n>]
   [GSURFN=<n>] [GSURFP=<n>]
   [MUN1=<n>] [MUN2=<n>] [AN=<n>] [CN=<n>]
   [EXN1=<n>] [EXN2=<n>] [EXN3=<n>] [EXN4=<n>]
   [MUP1=<n>] [MUP2=<n>] [AP=<n>] [CP=<n>]
   [EXP1=<n>] [EXP2=<n>] [EXP3=<n>] [EXP4=<n>]
   [MUN.MIN=<n>] [MUN.MAX=<n>] [NREFN=<n>]
   [NUN=<n>] [XIN=<n>] [ALPHAN=<n>]
   [MUP.MIN=<n>] [MUP.MAX=<n>] [NREFP=<n>]
   [NUP=<n>] [XIP=<n>] [ALPHAP=<n>]
```

Parameter	Type	Definition
TAB.TEMP	number	The temperature at which the mobility table values were measured. **Units:** Kelvins if the **KELVIN** parameter is specified, otherwise, degrees Celsius **Default**: none
KELVIN	logical	Specifies that the units of **TAB.TEMP** is Kelvins. **Default:** false
TAB.CONC	character	This parameter is interpreted as a series of numeric values, separated by spaces or commas. The parameter defines the concentrations associated with the mobilities defined in **TAB.E.MU** or **TAB.H.MU**. **Units:** atoms/cm^3 **Default:** none
TAB.E.MU	character	This parameter is interpreted as a series of numeric values, separated by spaces or commas. The parameter defines the electron mobilities associated with the concentrations defined in **TAB.CONC**. **Units:** cm^2/V/sec **Default:** none
TAB.H.MU	character	This parameter is interpreted as a series of numeric values, separated by spaces or commas. The parameter defines the hole mobilities associated with the concentrations defined in **TAB.CONC**. **Units:** cm^2/V/sec **Default:** none
TAB.CLEA	logical	Specifies that the current table is to be cleared. **Default:** false

Parameter	Type	Definition
ECN.MU	number	The critical electric field used to calculate the electron mobility degradation by perpendicular electric field. **Units:** V/cm **Default:** 6.49e4
ECP.MU	number	The critical electric field used to calculate the hole mobility degradation by perpendicular electric field. **Units:** V/cm **Default:** 1.87e4
GSURFN	number	The low-field surface reduction factor for electron mobility. **Units:** none **Default:** 1.0
GSURFP	number	The low-field surface reduction factor for hole mobility. **Units:** none **Default:** 1.0
MUN1	number	The minimum electron mobility used in the Arora mobility model. **Units:** cm^2/V/sec **Default:** 88.0
MUN2	number	The maximum electron mobility used in the Arora mobility model. **Units:** cm^2/V/sec **Default:** 1252.0
AN	number	Parameter used in the exponent of normalized impurity concentration in the Arora mobility model for electrons. **Units:** none **Default:** 0.88
CN	number	The reference impurity concentration used in the Arora mobility model for electrons. **Units:** atoms/cm^3 **Default:** 1.26e17
EXN1	number	Exponent of normalized temperature used in the Arora mobility model for electrons. **Units:** none **Default:** -0.57
EXN2	number	Exponent of normalized temperature used in the Arora mobility model for electrons. **Units:** none **Default:** -2.33
EXN3	number	Exponent of normalized temperature used in the Arora mobility model for electrons. **Units:** none **Default:** 2.4

Parameter	Type	Definition
`EXN4`	number	Exponent of normalized temperature used in the Arora mobility model for electrons. **Units:** none **Default:** -0.146
`MUP1`	number	The minimum hole mobility used in the Arora mobility model. **Units:** cm^2/V/sec **Default:** 54.3
`MUP2`	number	The maximum hole mobility used in the Arora mobility model. **Units:** cm^2/V/sec **Default:** 407.0
`AP`	number	Parameter used in the exponent of normalized impurity concentration in the Arora mobility model for holes. **Units:** none **Default:** 0.88
`CP`	number	The reference impurity concentration used in the Arora mobility model for holes. **Units:** atoms/cm^3 **Default:** 2.35e17
`EXP1`	number	Exponent of normalized temperature used in the Arora mobility model for holes. **Units:** none **Default:** -0.57
`EXP2`	number	Exponent of normalized temperature used in the Arora mobility model for holes. **Units:** none **Default:** -2.23
`EXP3`	number	Exponent of normalized temperature used in the Arora mobility model for holes. **Units:** none **Default:** 2.4
`EXP4`	number	Exponent of normalized temperature used in the Arora mobility model for holes. **Units:** none **Default:** -0.146
`MUN.MIN`	number	The minimum electron mobility in the Caughey mobility model. **Units:** cm^2/V/sec **Default:** 55.24
`MUN.MAX`	number	The maximum electron mobility in the Caughey mobility model. **Units:** cm^2/V/sec **Default:** 1429.23
`NREFN`	number	The reference impurity concentration used in the Caughey mobility model for electrons. **Units:** atoms/cm^3 **Default:** 1.072e17

Parameter	Type	Definition
NUN	number	Exponent of normalized temperature used in the numerator of the Caughey mobility model for electrons. **Units:** none **Default:** -2.3
XIN	number	Exponent of normalized temperature used in the denominator of the Caughey mobility model for electrons. **Units:** none **Default:** -3.8
ALPHAN	number	Exponent of the ratio of the total impurity concentration to **NREFN** used in the Caughey mobility model for electrons. **Units:** none **Default:** 0.733
MUP.MIN	number	The minimum hole mobility in the Caughey mobility model. **Units:** $cm^2/V/sec$ **Default:** 49.705
MUP.MAX	number	The maximum hole mobility in the Caughey mobility model. **Units:** $cm^2/V/sec$ **Default:** 479.37
NREFP	number	The reference impurity concentration used in the Caughey mobility model for holes. **Units:** $atoms/cm^3$ **Default:** 1.606e17
NUP	number	Exponent of normalized temperature used in the numerator of the Caughey mobility model for holes. **Units:** none **Default:** -2.2
XIP	number	Exponent of normalized temperature used in the denominator of the Caughey mobility model for holes. **Units:** none **Default:** -3.7
ALPHAP	number	Exponent of the ratio of the total impurity concentration to **NREFN** used in the Caughey mobility model for holes. **Units:** none **Default:** 0.70

Description

This statement is used to modify parameters associated with the various carrier mobility models which are available in the program.

Tables and Analytic Models

The dependencies of electron and hole mobilities on impurity concentration and temperature are represented by tables or analytic expressions.

Analytic Models

There are two analytic models:

- Arora's model
- Caughey's model

The analytic expressions are defined in Chapter 3, Carrier Mobility, p. 3-173. Since the default parameters of analytic models are based on the data measured on concentrations less than about $10^{20}/\text{cm}^3$, the analytic mobilities on heavily doped concentration might be not accurate.

Tables or Model Selection

The table is 2D with, at most, 100 rows representing impurity concentrations and with columns representing temperatures. Interpolation is used to obtain values for impurity concentration and temperature between the values in the table. During interpolation, the impurity concentration is assumed to vary exponentially and the temperature and the mobility are assumed to vary linearly. If the number of table data is too few to interpolate, a warning message is displayed and the Arora analytic model is used. For example, by default, TSUPREM-4 only has tabulated data at 300K. Therefore, if another temperature is input in the **ELECTRICAL** statement, the Arora model is selected to calculate the mobility.

The new table values can be added to the current table, and the current values can be modified. If the **TAB.CLEA** parameter is specified, the current table values are totally replaced with the new values. Otherwise, the current table values for concentrations other than those defined **TAB.CONC** are retained.

Example

Consider the following table of hole mobilities with four rows of impurity concentrations at 300K, and suppose that these values need to be replaced with the current table values.

Concentration ($\#/\text{cm}^3$)	**Hole Mobility (cm^2/Vsec)**
1e20	49.9
2e20	45.6
5e20	35.1
1e21	24.9

Compare with the default table.

Concentration ($\#/\text{cm}^3$)	**Hole Mobility (cm^2/Vsec)**	
	Default	New
1e20	52.0	49.9
2e20	50.8	45.6
4e20	49.6	none

5e20	none	35.1
6e20	48.9	none
8e20	48.4	none
1e21	48.0	24.9

The following statement makes the modified table, on which some values are replaced with new ones.

```
MOBILITY  TAB.TEMP=300 KELVIN +
          TAB.CONC=”1e20, 2e20, 5e20, 1e21” +
          TAB.H.MU=”49.9, 45.6, 35.1, 24.9”
```

Concentration (#/cm^3)	**Hole Mobility (cm^2/Vsec)**
1e20	49.9
2e20	45.6
4e20	49.6
5e20	35.1
6e20	48.9
8e20	48.4
1e21	24.9

However, since the mobilities associated with concentrations not represented in the `TAB.CONC` are left, the interpolation might result in undesirable mobilities. This can be avoided by the following statement:

```
MOBILITY  TAB.TEMP=300 KELVIN +
   TAB.CONC=”1e20,2e20,4e20,5e20,6e20,8e20,1e21” +
   TAB.H.MU=”49.9,45.6,   0,35.1,   0,  0,24.9”
```

The mobilities set to zero are interpolated automatically, and then the interpolated values replace the default ones.

INTERSTITIAL

The **INTERSTITIAL** statement sets the coefficients for interstitial kinetics.

```
INTERSTITIAL

   {MATERIAL=<c> | SILICON | OXIDE | OXYNITRI | NITRIDE | POLYSILI
     | AMBIENT}
     [D.0=<n>] [D.E=<n>] [VD=<n>] [D.FACTOR=<c>] [D.F11=<c>] [D.F22=<c>]
     [KB.0=<n>] [KB.E=<n>] [{KB.LOW | KB.MED | KB.HIGH}] [KIV.NORM]
     [CEQUIL.0=<n>] [CEQUIL.E=<n>] [CEQUIL.F=<c>] [VCEQUIL=<n>]
     [{CL.NONE|CL.1M|CL.2M}] [CL.FULL]
     [VMOLE=<n>]
     [NEU.0=<n>] [NEU.E=<n>] [NEG.0=<n>] [NEG.E=<n>] [DNEG.0=<n>]
     [DNEG.E=<n>] [POS.0=<n>] [POS.E=<n>] [DPOS.0=<n>] [DPOS.E=<n>]
     [ ( C.STATE=<n> [C.VACANC=<n>] )| C.ALL ]
     [ [DC.0=<n>] [DC.E=<n>] [FRAC.0=<n>] [FRAC.E=<n>]
       [KIV.0=<n>] [KIV.E=<n>] [KCV.0=<n>] [KCV.E=<n>]
       [ECLUST.0=<n>] [ECLUST.E=<n>] ]
     [TRAP.CON=<n>] [K.TRAP.0=<n>] [K.TRAP.E=<n>]
     [F.TRAP.0=<n>] [F.TRAP.E=<n>]
     [CL.KFI.0=<n>] [CL.KFI.E=<n>] [CL.IFI=<n>] [CL.ISFI=<n>]
     [CL.KFC.0=<n>] [CL.KFC.E=<n>] [CL.IFC=<n>] [CL.ISFC=<n>]
     [CL.KFCI=<n>] [CL.CF=<n>] [CL.KR.0=<n>] [CL.KR.E=<n>] [CL.CR=<n>]
     [CL.K0=<n>] [CL.K1=<n>] [CL.K2=<n>] [CL.CSS.0=<n>] [CL.CSS.E=<n>]
     [CL.LAM0=<n>] [CL.LAM1=<n>] [CL.CSL.0=<n>] [CL.CSL.E=<n>]
     [CL.ALPHA=<n>] [CL.NCRIT=<n>]
     [CL.SIZE=<n>]
     [ SC.SIZE=<n> [SC.C.INT=<n>]
       [SC.KFI.0=<n>] [SC.KFI.E=<n>] [SC.KRI.0=<n>] [SC.KRI.E=<n>]
       [SC.KFV.0=<n>] [SC.KFV.E=<n>] [SC.KRV.0=<n>] [SC.KRV.E=<n>]
     ]
     [CL.INI.F=<n>] [CL.EQUIL]
     [ECLUST.N=<n>] [KLOOP.0=<n>] [KLOOP.E=<n>][RLMIN=<n>]
   [{/MATERIA=<c> | /SILICON | /OXIDE | /OXYNITR | /NITRIDE | /POLYSIL
     | /AMBIENT}]
     [{V.MAXOX | V.INITOX | V.NORM}]
     [KSURF.0=<n>] [KSURF.E=<n>] [KSVEL.0=<n>] [KSVEL.E=<n>]
     [KSRAT.0=<n>] [KSRAT.E=<n>] [VNORM.0=<n>] [VNORM.E=<n>]
     [GROWTH] [THETA.0=<n>] [THETA.E=<n>]
     [A.0=<n>] [A.E=<n>] [T0.0=<n>] [T0.E=<n>]
     [KPOW.0=<n>] [KPOW.E=<n>] [GPOW.0=<n>] [GPOW.E=<n>]
     [N.MAX.0=<n>][N.MAX.E=<n>]
   [CM.SEC]
```

Parameter	Type	Definition
`MATERIAL`	character	The name of the material to which the other parameters apply. **Default:** none
`SILICON`	logical	The other parameters apply to silicon. **Default:** True if no other material specified.
`OXIDE`	logical	The other parameters apply to oxide. **Default:** false
`OXYNITRI`	logical	The other parameters apply to oxynitride. **Default:** false
`NITRIDE`	logical	The other parameters apply to nitride. **Default:** false
`POLYSILI`	logical	The other parameters apply to polysilicon. **Default:** false
`AMBIENT`	logical	The other parameters apply to the ambient gas. **Default:** false **Synonym: `GAS`**
`D.0`	number	The pre-exponential constant for the diffusivity of interstitials in the specified material. Only used with **`SILICON`**. **Units:** microns^2/min or cm^2/sec or none **Default:** the current value for this material **Synonym: `DI`**
`D.E`	number	The activation energy for the diffusivity of interstitials in the specified material. Only used with **`SILICON`**. **Units:** electron volts **Default:** the current value for this material
`D.FACTOR`	character	The user-defined expression of the isotropic multiplication factor of diffusivity. Only used with **`SILICON`**. The expression can be registered and applied under the **`MODEL`** parameter which is turned on/off in the **`METHOD`** statement. **Units:** none **Default:** the current value for this material
`D.F11`	character	The user-defined expression of the anisotropic x-directional tensor multiplication factor of diffusivity. Only used with **`SILICON`**. The expression can be registered and applied under the **`MODEL`** parameter which is turned on/off in the **`METHOD`** statement. **Units:** none **Default:** the current value for this material
`D.F22`	character	The user-defined expression of the anisotropic y-directional tensor multiplication factor of diffusivity. Only used with **`SILICON`**. The expression can be registered and applied under the **`MODEL`** parameter which is turned on/off in the **`METHOD`** statement. **Units:** none **Default:** the current value for this material

Parameter	Type	Definition
VD	number	The activation volume for the pressure-dependent diffusivity **Units:** Å^3 **Default:** the current value for this material
KB.0	number	The pre-exponential constant for bulk recombination rate in the specified material. Only used with **SILICON**. **Units:** microns^3/min or cm^3/sec **Default:** the current value for this material **Synonym: KR.0**
KB.E	number	The activation energy for bulk recombination rate in the specified material. Only used with **SILICON**. **Units:** electron volts **Default:** the current value for this material
KB.LOW	logical	Assume that only interstitials and vacancies having opposite charges of equal magnitude recombine in bulk material. **Default:** the current value for this material; initially true
KB.MED	logical	Assume that uncharged point defects can recombine with defects in any charge state, and that oppositely charged point defects can recombine. **Default:** the current value for this material; initially false
KB.HIGH	logical	Assume the charge state of a point defect does not affect its rate of recombination. **Default:** the current value for this material; initially false
KIV.NORM	logical	Normalize the charge state dependence in the point defect recombination model. **Default:** the current value for this material; initially false
CEQUIL.0	number	The pre-exponential constant for equilibrium concentration of interstitials in the specified material. Only used with **SILICON**. **Units:** interstitials/cm^3 **Default:** the current value for this material **Synonym: CI**
CEQUIL.E	number	The activation energy for equilibrium concentration of interstitials in the specified material. Only used with **SILICON**. **Units:** electron volts **Default:** the current value for this material
CEQUIL.F	character	The user-defined expression of the multiplication factor of an intrinsic equilibrium interstitial concentration. Only used with **SILICON**. The expression can be registered and applied under the **MODEL** parameter which is turned on/off in the **METHOD** statement. **Units:** none **Default:** the current value for this material
VCEQUIL	number	The activation volume for the pressure-dependent interstitial concentration in intrinsic equilibrium. **Units:** Å^3 **Default:** the current value for this material

Parameter	Type	Definition
`CL.NONE`	logical	Disables the interstitial clustering model for the specified material. **Default:** the current value; initially false for silicon and true for other materials.
`CL.1M`	logical	Enables the 1-moment interstitial clustering model for the specified material. **Default:** the current value; initially true for silicon and false for other materials. **Synonym**: `CL.MODEL`
`CL.2M`	logical	Enables the 2-moment interstitial clustering model for the specified material. **Default:** the current value; initially false
`CL.FULL`	logical	Enables the full transient small interstitial clustering model for the specified material. **Default:** the current value; initially false
`VMOLE`	number	The concentration of silicon atoms, used in the `GROWTH` model. Only used with `SILICON`. **Units:** atoms/cm^3 **Default:** the current value for this material
`NEU.0`	number	The pre-exponential constant for the relative concentration of neutral interstitials. **Units:** none **Default:** the current value for this material
`NEU.E`	number	The activation energy for the relative concentration of neutral interstitials. **Units:** electron volts **Default:** the current value for this material
`NEG.0`	number	The pre-exponential constant for the relative concentration of negatively charged interstitials. **Units:** none **Default:** the current value for this material
`NEG.E`	number	The activation energy for the relative concentration of negatively charged interstitials. **Units:** electron volts **Default:** the current value for this material
`DNEG.0`	number	The pre-exponential constant for the relative concentration of doubly-negative charged interstitials. **Units:** none **Default:** the current value for this material
`DNEG.E`	number	The activation energy for the relative concentration of doubly-negative charged interstitials. **Units:** electron volts **Default:** the current value for this material
`POS.0`	number	The pre-exponential constant for the relative concentration of positively charged interstitials. **Units:** none **Default:** the current value for this material

Parameter	Type	Definition
`POS.E`	number	The activation energy for the relative concentration of positively charged interstitials. **Units:** electron volts **Default:** the current value for this material
`DPOS.0`	number	The pre-exponential constant for the relative concentration of doubly-positive charged interstitials. **Units:** none **Default:** the current value for this material
`DPOS.E`	number	The activation energy for the relative concentration of doubly-positive charged interstitials. **Units:** electron volts **Default:** the current value for this material
`C.STATE`	number	The interstitial charge state to which the **`DC.0`**, **`DC.E`**, **`FRAC.0`**, **`FRAC.E`**, **`KIV.0`**, **`KIV.E`**, **`KCV.0`**, **`KCV.E`**, **`ECLUST.0`**, **`ECLUST.E`**, **`SC.KFI.0`**, **`SC.KFI.E`**, **`SC.KRI.0`** and **`SC.KRI.E`** parameters apply. The value must be in the range of –6 to +6. **Units:** none **Default:** none **Synonym:** **`C.INTERS`**
`C.VACANC`	number	The vacancy charge state to which the **`KIV.0`**, **`KIV.E`**, **`KCV.0`**, **`KCV.E`**, **`SC.KFV.0`**, **`SC.KFV.E`**, **`SC.KRV.0`** and **`SC.KRV.E`** parameters apply. The value must be in the range of –6 to +6. **Units:** none **Default:** none
`C.ALL`	number	Specifies that the **`DC.0`**, **`DC.E`**, **`FRAC.0`**, **`FRAC.E`**, **`KIV.0`**, **`KIV.E`**, **`KCV.0`**, **`KCV.E`**, **`ECLUST.0`**, **`ECLUST.E`**, **`SC.KFI.0`**, **`SC.KFI.E`**, **`SC.KRI.0`**, **`SC.KRI.E`**, **`SC.KFV.0`**, **`SC.KFV.E`**, **`SC.KRV.0`** and **`SC.KRV.E`** parameters should be set for all charge states. **Default:** false
`DC.0`	number	The pre-exponential constant for diffusion of interstitials in the charge state(s) specified by **`C.STATE`** (or **`C.ALL`**) in the specified material. Only used with **`SILICON`**. **Units:** none or microns^2/min or cm^2/sec **Default:** the current value for this material; initially 1.0 for all charge states
`DC.E`	number	The activation energy for diffusion of interstitials in the charge state(s) specified by **`C.STATE`** (or **`C.ALL`**) in the specified material. Only used with **`SILICON`**. **Units:** electron volts **Default:** the current value for this material; initially 0.0 for all charge states
`FRAC.0`	number	The pre-exponential constant for the relative concentration of interstitials in the charge state **`C.STATE`** (or **`C.ALL`**). **Units:** none **Default:** the current value for this material

Parameter	Type	Definition
`FRAC.E`	number	The activation energy for the relative concentration of interstitials in the charge state **`C.STATE`** (or **`C.ALL`**). **Units:** electron volts **Default:** the current value for this material
`KIV.0`	number	The pre-exponential constant for the bulk recombination factor for interstitials in charge state **`C.I`** (**`C.STATE`**) and vacancies in charge state **`C.V`** (or for all combinations of charge states if **`C.ALL`** is set) in the specified material. Only used with **`SILICON`**. **Units:** none **Default:** the current value for this material
`KIV.E`	number	The activation energy for the bulk recombination factor for interstitials in charge state **`C.I`** (**`C.STATE`**) and vacancies in charge state **`C.V`** (or for all combinations of charge states if **`C.ALL`** is set) in the specified material. Only used with **`SILICON`**. **Units:** electron volts **Default:** the current value for this material
`KCV.0`	number	The pre-exponential constant for the bulk recombination factor for small interstitial clusters in charge state **`C.I`** (**`C.STATE`**) and vacancies in charge state **`C.V`** (or for all combinations of charge states if **`C.ALL`** is set) in the specified material. Only used with **`SILICON`**. **Units:** none **Default:** the current value for this material
`KCV.E`	number	The activation energy for the bulk recombination factor for small interstitial clusters in charge state **`C.I`** (**`C.STATE`**) and vacancies in charge state **`C.V`** (or for all combinations of charge states if **`C.ALL`** is set) in the specified material. Only used with **`SILICON`**. **Units:** electron volts **Default:** the current value for this material
`ECLUST.0`	number	The pre-exponential constant for the concentration of interstitials in small clusters in charge state **`C.STATE`** at thermal equilibrium in the specified material. Only used with **`SILICON`**. **Units:** interstitials/cm^3 **Default:** the current value for this material
`ECLUST.E`	number	The activation energy for the concentration of interstitials in small clusters in charge state **`C.STATE`** at thermal equilibrium in the specified material. Only used with **`SILICON`**. **Units:** electron volts **Default:** the current value for this material
`TRAP.CON`	number	The concentration of interstitial traps. **Units:** traps/cm^3 **Default:** the current value; initially 0.0

Parameter	Type	Definition
`K.TRAP.0`	number	The pre-exponential constant in the expression for the forward trap time constant. **Units:** cm^3/sec **Default:** the current value; initially 0.0
`K.TRAP.E`	number	The activation energy in the expression for the forward trap time constant. **Units:** electron volts **Default:** the current value; initially 0.0
`F.TRAP.0`	number	The pre-exponential constant in the expression for the fraction of empty interstitial traps at equilibrium. **Units:** none **Default:** the current value; initially 0.0
`F.TRAP.E`	number	The activation energy in the expression for the fraction of empty interstitial traps at equilibrium. **Units:** electron volts **Default:** the current value; initially 0.0
`CL.KFI.0`	number	The prefactor for the K_{fi} ($I+I \rightarrow C$) term in the interstitial clustering model. **Units:** $cm^{-(3*(1+\mathtt{CL.ISFI}-\mathtt{CL.IFI}))}$/sec **Default:** the current value for this material
`CL.KFI.E`	number	The activation energy for the K_{fi} ($I+I \rightarrow C$) term in the interstitial clustering model. **Units:** electron volts **Default:** the current value for this material
`CL.IFI`	number	The power of the interstitial concentration in the K_{fi} ($I+I \rightarrow C$) term in the interstitial clustering model. **Units:** none **Default:** the current value for this material
`CL.ISFI`	number	The power of the equilibrium interstitial concentration in the K_{fi} ($I+I \rightarrow C$) term in the interstitial clustering model. **Units:** none **Default:** the current value for this material
`CL.KFC.0`	number	The prefactor for the K_{fc} ($I+C \rightarrow C$) term in the interstitial clustering model. **Units:** $cm^{-(3*(1-\mathtt{CL.CF}+\mathtt{CL.ISFC}-\mathtt{CL.IFC}))}$/sec **Default:** the current value for this material
`CL.KFC.E`	number	The activation energy for the K_{fc} ($I+C \rightarrow C$) term in the interstitial clustering model. **Units:** electron volts **Default:** the current value for this material

Parameter	Type	Definition
`CL.IFC`	number	The power of the interstitial concentration in the K_{fc} ($I+C \rightarrow C$) term in the interstitial clustering model. **Units:** none **Default:** the current value for this material
`CL.ISFC`	number	The power of the equilibrium interstitial concentration in the K_{fc} ($I+C \rightarrow C$) term in the interstitial clustering model. **Units:** none **Default:** the current value for this material
`CL.KFCI`	number	The fraction of the interstitial concentration to be included in the K_{fc} ($I+C \rightarrow C$) term in the interstitial clustering model. **Units:** none **Default:** the current value for this material; initially 1.0
`CL.CF`	number	The power of the clustered interstitial concentration in the K_{fc} ($I+C \rightarrow C$) term in the interstitial clustering model. **Units:** none **Default:** the current value for this material
`CL.KR.0`	number	The prefactor for the K_r (cluster dissolution) term in the interstitial clustering model. **Units:** $cm^{-(3*(1-\mathtt{CL.CR}))}$/sec **Default:** the current value for this material
`CL.KR.E`	number	The activation energy for the K_r (cluster dissolution) term in the interstitial clustering model. **Units:** electron volts **Default:** the current value for this material
`CL.CR`	number	The power of the clustered interstitial concentration in the K_r (cluster dissolution) term in the interstitial clustering model. **Units:** none **Default:** the current value for this material
`CL.K0`	number	Gencer's K_0 parameter for dissociation of interstitial clusters. **Units:** none **Default:** the current value for this material
`CL.K1`	number	Gencer's K_1 parameter for dissociation of 2-atom interstitial clusters. **Units:** none **Default:** the current value for this material
`CL.K2`	number	Gencer's K_2 parameter for dissociation of interstitial clusters. **Units:** none **Default:** the current value for this material

Parameter	Type	Definition
`CL.CSS.0`	number	The prefactor for the solubility of interstitials in equilibrium with large {311} clusters. **Units:** none **Default:** the current value for this material
`CL.CSS.E`	number	The activation energy for the solubility of interstitials in equilibrium with large {311} clusters. **Units:** electron volts **Default:** the current value for this material
`CL.LAM0`	number	The capture length for nucleation of interstitial clusters. **Units:** cm **Default:** the current value for this material
`CL.LAM1`	number	The capture length for the growth of interstitial clusters. **Units:** cm **Default:** the current value for this material
`CL.CSL.0`	number	The prefactor for the solubility of interstitials in equilibrium with large dislocation loops. **Units:** none **Default:** the current value for this material
`CL.CSL.E`	number	The activation energy for the solubility of interstitials in equilibrium with large dislocation loops. **Units:** electron volts **Default:** the current value for this material
`CL.ALPHA`	number	The exponent for reducing the dissociation rate of dislocation loops. **Units:** none **Default:** the current value for this material
`CL.NCRIT`	number	The average cluster size above which interstitial clusters begin to behave like dislocation loops. **Units:** none **Default:** the current value for this material
`CL.SIZE`	number	The minimum size of 311 cluster, which is the same as (the maximum size of the small cluster + 1). The specifications of the parameter values for the clusters with `SC.SIZE` larger than `CL.SIZE` are ignored. When `SC.SIZE` is equal to `CL.SIZE`, only the reverse reaction parameters are taken into account. It may not be smaller than 2 or bigger than 20. **Units:** none **Default:** 0
`SC.SIZE`	number	The size of the small interstitial cluster which will be defined. It may not be smaller than 2 or bigger than 20. **Units:** none **Default:** 0

Parameter	Type	Definition
`SC.C.INT`	number	The charge state of interstitial to form a I_2 cluster. The charge state of the other interstitial is specified with `C.STATE`. **Units:** none **Default:** 0
`SC.KFI.0`	number	The prefactor of the forward reaction rate of $I_{(\mathtt{SC.SIZE}-1)} + I <-> I_{(\mathtt{SC.SIZE})}$ for the given `SC.SIZE` and the `C.STATE` charged interstitial. **Units**: microns^3/min or cm^3/sec **Default:** the current value for this material
`SC.KFI.E`	number	The activation energy of the forward reaction rate of $I_{(\mathtt{SC.SIZE}-1)} + I <-> I_{(\mathtt{SC.SIZE})}$ for the given `SC.SIZE` and the `C.STATE` charged interstitial. **Units:** electron volts **Default:** the current value for this material
`SC.KRI.0`	number	The prefactor of the reverse reaction rate of $I_{(\mathtt{SC.SIZE}-1)} + I <-> I_{(\mathtt{SC.SIZE})}$ for the given `SC.SIZE` and the `C.STATE` charged interstitial. **Units:** min^{-1} or sec^{-1} **Default:** the current value for this material
`SC.KRI.E`	number	The activation energy of the reverse reaction rate of $I_{(\mathtt{SC.SIZE}-1)} + I <-> I_{(\mathtt{SC.SIZE})}$ for the given `SC.SIZE` and the `C.STATE` charged interstitial. **Units:** electron volts **Default:** the current value for this material
`SC.KFV.0`	number	The prefactor of the forward reaction rate of $I_{(\mathtt{SC.SIZE}-1)} <-> I_{(\mathtt{SC.SIZE})} + V$ for the given `SC.SIZE` and the `C.VACANC` charged vacancy. **Units:** min^{-1} or sec^{-1} **Default:** the current value for this material
`SC.KFV.E`	number	The activation energy of the forward reaction rate of $I_{(\mathtt{SC.SIZE}-1)} <-> I_{(\mathtt{SC.SIZE})} + V$ for the given `SC.SIZE` and the `C.VACANC` charged vacancy. **Units:** electron volts **Default:** the current value for this material
`SC.KRV.0`	number	The prefactor of the reverse reaction rate of $I_{(\mathtt{SC.SIZE}-1)} <-> I_{(\mathtt{SC.SIZE})} + V$ for the given `SC.SIZE` and the `C.VACANC` charged vacancy. **Units:** microns^3/min or cm^3/sec **Default:** the current value for this material
`SC.KRV.E`	number	The activation of the reverse reaction rate of $I_{(\mathtt{SC.SIZE}-1)} <-> I_{(\mathtt{SC.SIZE})} + V$ for the given `SC.SIZE` and the `C.VACANC` charged vacancy. **Units:** electron volts **Default:** the current value for this material

Parameter	Type	Definition
`CL.INI.F`	number	The fraction of interstitials that are clustered upon implantation. Redistributes interstitials that are remaining after the dopant-defect clusters are formed according to **`Cl.INI.F`**. Applies only outside of the amorphized regions. Inside the amorphized regions, all interstitial clusters are destroyed. The value must be set in the range from 0.0 to 1.0. **Units:** none **Default:** 0 **Synonym**: **`INIT.311`**
`CL.EQUIL`	logical	Specifies that the initial interstitial clusters will be initialized to the equilibrium concentration. Turning on **`CL.EQUIL`** precedes the specification of **`CL.INI.F`**. **Default:** the current value; initially false
`ECLUST.N`	number	The number of interstitials in each small cluster. **Units:** none **Default:** the current value for this material
`KLOOP.0`	number	The prefactor for K_L in the dislocation loop model. **Units:** none **Default:** the current value for this material
`Kloop.E`	number	The activation energy for K_L in the dislocation loop model. **Units:** electron volts **Default:** the current value for this material
`RLMIN`	number	The minimum radius for a dislocation loop; loops smaller than this size are removed from the simulation. **Units:** cm **Default:** the current value for this material; initially 100e-8 cm
`/MATERIA`	character	The name of the second material for specifying interface injection and recombination parameters. **Default:** none
`/SILICON`	logical	The interface injection and recombination parameters apply to the interface between the specified material and silicon. **Default:** false
`/OXIDE`	logical	The interface injection and recombination parameters apply to the interface between the specified material and oxide. **Default:** false
`/OXYNITR`	logical	The interface injection and recombination parameters apply to the interface between the specified material and oxynitride. **Default:** false
`/NITRIDE`	logical	The interface injection and recombination parameters apply to the interface between the specified material and nitride. **Default:** false
`/POLYSIL`	logical	The interface injection and recombination parameters apply to the interface between the specified material and polysilicon. **Default:** false

Parameter	Type	Definition
`/AMBIENT`	logical	The interface injection and recombination parameters apply to the interface between the specified material and the ambient gas. **Default:** false **Synonym:** **`/GAS`**
`V.MAXOX`	logical	Use the injection/recombination model, in which the interface velocity is normalized by the maximum interface velocity in the structure (the model used in older versions of TSUPREM-4). **Default:** the current value for these materials
`V.INITOX`	logical	Use the injection/recombination model, in which the interface velocity is normalized by the initial growth velocity for a bare silicon surface. **Default:** the current value for these materials
`V.NORM`	logical	Use the injection/recombination model, in which the interface velocity is normalized by the value specified by **`VNORM.0`** and **`VNORM.E`** parameters. **Default:** the current value for these materials
`KSURF.0`	number	The pre-exponential constant for the surface recombination velocity at the interface between the specified materials under inert conditions. **Units:** microns/min or cm/sec **Default:** the current value for these materials **Synonyms:** **`KSMIN.0`**, **`KI.MIN`**
`KSURF.E`	number	The activation energy for the surface recombination velocity at the interface between the specified materials under inert conditions. **Units:** electron volts **Default:** the current value for these materials **Synonym:** **`KSMIN.E`**
`KSVEL.0`	number	The pre-exponential constant for the growth-rate-dependent component of the surface recombination velocity at the interface between the specified materials. Used only with the **`V.MAXOX`** and **`V.NORM`** models. **Units:** microns/min or cm/sec **Default:** the current value for these materials **Synonyms:** **`KSMAX.0`**, **`KI.MAX`**
`KSVEL.E`	number	The activation energy for the growth-rate-dependent component of the surface recombination velocity at the interface between the specified materials. Used only with the **`V.MAXOX`** and **`V.NORM`** models. **Units:** electron volts **Default:** the current value for these materials **Synonym:** **`KSMAX.E`**
`KSRAT.0`	number	The pre-exponential constant for the ratio of the growth-rate-dependent component of the surface recombination velocity to the inert component at the interface between the specified materials. Used only with the **`V.INITOX`** model. **Units:** none **Default:** the current value for these materials

Parameter	Type	Definition
`KSRAT.E`	number	The activation energy for the ratio of the growth-rate-dependent component of the surface recombination velocity to the inert component at the interface between the specified materials. Used only with the **`V.INITOX`** model. **Units:** electron volts **Default:** the current value for these materials
`VNORM.0`	number	The pre-exponential constant for the normalization velocity in the **`V.NORM`** model. **Units:** microns/min or cm/sec **Default:** the current value for these materials
`VNORM.E`	number	The activation energy for the normalization velocity in the **`V.NORM`** model. **Units:** electron volts **Default:** the current value for these materials
`GROWTH`	logical	Specifies that interstitial injection should be calculated using the computed velocities of the interface rather than the analytic model. **Default:** the current value for these materials
`THETA.0`	number	The pre-exponential constant for the fraction of consumed silicon atoms that are injected into the bulk as interstitials during oxidation. **Units:** none, for **`V.INITOX`** and **`V.NORM`** models; $(\text{microns/min})^{(1-\mathbf{KPOW})}$ or $(\text{cm/sec})^{(1-\mathbf{KPOW})}$ for **`V.MAXOX`** model **Default:** the current value for these materials **Synonym: `THETA`**
`THETA.E`	number	The activation energy for the fraction of consumed silicon atoms that are injected into the bulk as interstitials during oxidation. **Units:** electron volts **Default:** the current value for these materials
`A.0`	number	The pre-exponential constant for the injection rate of interstitials at the interface between the specified materials. **Units:** $\#/\text{micron}^2/\text{min}^{(1+\mathbf{GPOW})}$ or $\#/\text{cm}^2/\text{sec}^{(1+\mathbf{GPOW})}$ for **`V.INITOX`** and **`V.NORM`** models; $\#/\text{micron}^2/\text{min}^{(1-\mathbf{KPOW})}$ or $\#/\text{cm}^2/\text{sec}^{(1-\mathbf{KPOW})}$ for **`V.MAXOX`** model **Default:** the current value for these materials **Synonym: `A`**
`A.E`	number	The activation energy for the injection rate of interstitials at the interface between the specified materials. **Units:** electron volts **Default:** the current value for these materials
`T0.0`	number	The pre-exponential constant for the time constant for injection at the interface between the specified materials. **Units:** minutes or seconds **Default:** the current value for these materials **Synonyms: `T0`**

Parameter	Type	Definition
`T0.E`	number	The activation energy for the time constant for injection at the interface between the specified materials. **Units:** electron volts **Default:** the current value for these materials
`KPOW.0`	number	The pre-exponential constant in the expression for the exponent in the surface recombination models. (Also used for injection with the **`V.MAXOX`** model.) **Units:** none **Default:** the current value for these materials **Synonyms:** **`TPOW.0`**, **`POWER`**
`KPOW.E`	number	The activation energy in the expression for the exponent in the surface recombination models. (Also used for injection with the **`V.MAXOX`** model.) **Units:** electron volts **Default:** the current value for these materials **Synonym:** **`TPOW.E`**
`GPOW.0`	number	The pre-exponential constant in the expression for the exponent in the **`V.INITOX`** and **`V.NORM`** models for interstitial injection. **Units:** none **Default:** the current value for these materials
`GPOW.E`	number	The activation energy in the expression for the exponent in the **`V.INITOX`** and **`V.NORM`** models for interstitial injection. **Units:** electron volts **Default:** the current value for these materials
`N.MAX.0`	number	The pre-exponential constant in the expression for the exponent in the maximum trapping site of nitrogen for interstitial recombination. **Units:** /cm^2 **Default:** the current value for these materials
`N.MAX.E`	number	The activation energy in the expression for the exponent in the maximum trapping site of nitrogen for interstitial recombination. **Units:** electron volts **Default:** the current value for these materials
`CM.SEC`	logical	If true, parameters involving time are specified in centimeters and seconds; if false, parameters involving time are in microns and minutes. **Default:** false

Description

This statement specifies values for coefficients of interstitial diffusion, recombination, injection, equilibrium concentration, and interstitial trap parameters. These parameters are normally specified in the *s4init* file (which is read when the program starts up), but can be changed by you at any time. Values that have *not* been set in *s4init* or by the user, default to 0.0.

Parameters whose units include time are specified in microns and minutes, unless **`CM.SEC`** is true, in which case units of centimeters and seconds are assumed.

The interpretation of the various parameters on the **INTERSTITIAL** statement is described in Chapter 3, Dose Loss Model, p. 3-58.

Bulk and Interface Parameters

If bulk parameters (e.g., **D.0** or **CEQUIL.E**) are specified but no material is given, the parameters are assumed to apply to silicon. If interface parameters (e.g., **KSVEL.0** or **A.E**) are specified and only a “first” material (e.g., **OXIDE**) is given, the parameters are assumed to apply to the interface between silicon and the specified material. This usage is not recommended, and is intended only for compatibility with older releases of TSUPREM-4.

Examples

1. The following statement specifies the silicon diffusivity and equilibrium values for interstitials:

```
INTERSTITIAL SILICON D.0=3000 CEQUIL.0=1.0e13
```

2. The following statement causes the Si/SiO$_2$ interface injection is to be computed using the oxide growth velocity, with 1% of consumed silicon injected as interstitials:

```
INTERSTITIAL SILICON /OXIDE GROWTH THETA.0=0.01
```

3. The following statement specifies that the surface recombination velocity in the silicon at the interface with nitride is 3.5ex10^{-3} cm/s

```
INTERSTITIAL SILICON /NITRIDE CM.SEC +
   KSURF.0=3.5e-3 KSURF.E=0.0
```

4. The following statement specifies that the ratio of doubly-positive interstitials to neutral interstitials is 0.1 under intrinsic conditions:

```
INTERSTITIAL SILICON C.STATE=2 FRAC.0=0.1 +
   FRAC.E=0.0
```

The **C.STATE**, **FRAC.0**, and **FRAC.E** parameters can be used to specify the fraction of interstitials in any charge state from -6 to +6.

5. The following example shows how to define the small interstitial clusters (`I2`, `I3` and `I4`). The minimum size of the `cl_interst` cluster is 5.

```
$ define the two-moment large clustering model
INTERST SILICON CL.2M
$ define the transient small clustering model
INTERST SILICON CL.FULL CL.SIZE=5

$ small clusters
INTERST SC.SIZE=2 C.ALL  SC.KFI.0=1.26E-4
SC.KFI.E=1.77
INTERST SC.SIZE=2 C.ALL  SC.KRI.0=5.20E-8
SC.KRI.E=0.06
INTERST SC.SIZE=3 C.ALL  SC.KFI.0=1.34E-5
SC.KFI.E=1.77
INTERST SC.SIZE=3 C.ALL  SC.KRI.0=6.90E-11
SC.KRI.E=-0.23
INTERST SC.SIZE=4 C.ALL  SC.KFI.0=2.86E-5
SC.KFI.E=1.77
INTERST SC.SIZE=4 C.ALL  SC.KRI.0=9.00E-8
SC.KRI.E=-0.28
INTERST SC.SIZE=5 C.ALL  SC.KFI.0=3.78E-5
SC.KFI.E=1.77

$ two-moment large cluster model
$ Note that, in this example, the transformation
of {311}
$ clusters into dislocation loops is ignored by
setting
$ CL.NCRIT=5E22
INTERST CL.LAM0=7.4e-07 CL.LAM1=7.4e-07
INTERST CL.CSS.0=0.00840597 CL.CSS.e=-1.25
CL.NCRIT=5e22
INTERST CL.K0=-2 CL.K1=1.0 CL.K2=0.0
```

Additional INTERSTITIAL Notes

- The model for interstitial traps has not been calibrated. Typical values of **TRAP.CON** are in the range 10^{15} to 10^{18}. The combination of **F.TRAP.0** and **F.TRAP.E** should give a value between 0.0 and 1.0.
- Coefficients can be specified for each of the materials and interfaces, but at present only the coefficients for silicon and interfaces between silicon and other materials are used.
- **D.0** and **DC.0** are used together to specify the diffusivity of interstitials in each charge state. One of these should have the units of cm^2/sec or $microns^2$/min while the other should be unitless. See Chapter 3, Interstitial and Vacancy Diffusivities, p. 3-63.

VACANCY

The **VACANCY** statement sets the coefficients for vacancy kinetics.

```
VACANCY

   {MATERIAL=<c> | SILICON | OXIDE | OXYNITRI | NITRIDE | POLYSILI
     | AMBIENT}
     [D.0=<n>] [D.E=<n>] [D.FACTOR=<c>] [D.F11=<c>] [D.F22=<c>]
     [VD=<n>] [CEQUIL.0=<n>] [CEQUIL.E=<n>] [CEQUIL.F=<c>]
     [VCEQUIL=<n>] [VMOLE=<n>] [MODEL=<c>]
     [NEU.0=<n>] [NEU.E=<n>] [NEG.0=<n>] [NEG.E=<n>] [DNEG.0=<n>]
     [DNEG.E=<n>] [POS.0=<n>] [POS.E=<n>] [DPOS.0=<n>] [DPOS.E=<n>]
     [ { ( [C.I=<n>] C.STATE=<n> ) | C.ALL }
       [DC.0=<n>] [DC.E=<n>] [FRAC.0=<n>] [FRAC.E=<n>]
       [KIC.0=<n>] [KIC.E=<n>] [ECLUST.0=<n>] [ECLUST.E=<n>] ]
       [ECLUST.N=<n> ]
   [{/MATERIA=<c> | /SILICON | /OXIDE | /OXYNITR | /NITRIDE | /POLYSIL
     | /AMBIENT}]
     [{V.MAXOX | V.INITOX | V.NORM}]
     [KSURF.0=<n>] [KSURF.E=<n>] [KSVEL.0=<n>] [KSVEL.E=<n>]
     [KSRAT.0=<n>] [KSRAT.E=<n>] [VNORM.0=<n>] [VNORM.E=<n>]
     [GROWTH] [THETA.0=<n>] [THETA.E=<n>]
     [A.0=<n>] [A.E=<n>] [T0.0=<n>] [T0.E=<n>]
     [KPOW.0=<n>] [KPOW.E=<n>] [GPOW.0=<n>] [GPOW.E=<n>]
     [CL.KRV.0=<n>] [CL.KRV.E=<n>] [CL.VRV=<n>] [CL.VSRV=<n>]
     [CL.CRV=<n>] [CL.KFV.0=<n>] [CL.KFV.E=<n>] [CL.VSFV=<n>]
     [CL.CFV=<n>]
     [CM.SEC]
```

Parameter	Type	Definition
MATERIAL	character	The name of the material to which the other parameters apply. **Default:** none
SILICON	logical	The other parameters apply to silicon. **Default:** true if no other material specified
OXIDE	logical	The other parameters apply to oxide. **Default:** false
OXYNITRI	logical	The other parameters apply to oxynitride. **Default:** false
NITRIDE	logical	The other parameters apply to nitride. **Default:** false
POLYSILI	logical	The other parameters apply to polysilicon. **Default:** false
AMBIENT	logical	The other parameters apply to the ambient gas. **Default:** false **Synonym: GAS**

Parameter	Type	Definition
D.0	number	The pre-exponential constant for the diffusivity of vacancies in the specified material. Only used with **SILICON**. **Units:** microns^2/min or cm^2/sec or none **Default:** the current value for this material **Synonym: DI**
D.E	number	The activation energy for the diffusivity of vacancies in the specified material. Only used with **SILICON**. **Units:** electron volts **Default:** the current value for this material
D.FACTOR	character	The user-defined expression of the isotropic multiplication factor of diffusivity. Only used with **SILICON**. The expression can be registered and applied under the **MODEL** parameter which is turned on/off in the **METHOD** statement. **Units:** none **Default:** the current value for this material
D.F11	character	The user-defined expression of the anisotropic x-directional tensor multiplication factor of diffusivity. Only used with **SILICON**. The expression can be registered and applied under the **MODEL** parameter which is turned on/off in the **METHOD** statement. **Units:** none **Default:** the current value for this material
D.F22	character	The user-defined expression of the anisotropic y-directional tensor multiplication factor of diffusivity. Only used with **SILICON**. The expression can be registered and applied under the **MODEL** parameter which is turned on/off in the **METHOD** statement. **Units:** none **Default:** the current value for this material
VD	number	The activation volume for the pressure-dependent diffusivity **Units:** Å^3 **Default:** the current value for this material
CEQUIL.0	number	The pre-exponential constant for equilibrium concentration of vacancies in the specified material. Only used with **SILICON**. **Units:** vacancies/cm^3 **Default:** the current value for this material **Synonym: CV**
CEQUIL.E	number	The activation energy for equilibrium concentration of vacancies in the specified material. Only used with **SILICON**. **Units:** electron volts **Default:** the current value for this material
CEQUIL.F	character	The user-defined expression of the multiplication factor of an intrinsic equilibrium vacancy concentration. Only used with **SILICON**. The expression can be registered and applied under the **MODEL** parameter which is turned on/off in the **METHOD** statement. **Units:** none **Default:** the current value for this material

Parameter	Type	Definition
VCEQUIL	number	The activation volume for the pressure-dependent vacancy concentration in intrinsic equilibrium. **Units:** Å^3 **Default:** the current value for this material
VMOLE	number	The concentration of silicon atoms, used in the **GROWTH** model. Only used with **SILICON**. **Units:** atoms/cm^3 **Default:** the current value for this material
NEU.0	number	The pre-exponential constant for the relative concentration of neutral vacancies. **Units:** none **Default:** the current value for this material
NEU.E	number	The activation energy for the relative concentration of neutral vacancies. **Units:** electron volts **Default:** the current value for this material
NEG.0	number	The pre-exponential constant for the relative concentration of negatively charged vacancies. **Units:** none **Default:** the current value for this material
NEG.E	number	The activation energy for the relative concentration of negatively charged vacancies. **Units:** electron volts **Default:** the current value for this material
DNEG.0	number	The pre-exponential constant for the relative concentration of doubly-negative charged vacancies. **Units:** none **Default:** the current value for this material
DNEG.E	number	The activation energy for the relative concentration of doubly-negative charged vacancies. **Units:** electron volts **Default:** the current value for this material
POS.0	number	The pre-exponential constant for the relative concentration of positively charged vacancies. **Units:** none **Default:** the current value for this material
POS.E	number	The activation energy for the relative concentration of positively charged vacancies. **Units:** electron volts **Default:** the current value for this material
DPOS.0	number	The pre-exponential constant for the relative concentration of doubly-positive charged vacancies. **Units:** none **Default:** the current value for this material

Parameter	Type	Definition
DPOS.E	number	The activation energy for the relative concentration of doubly-positive charged vacancies. **Units:** electron volts **Default:** the current value for this material
C.I	number	The interstitial charge state to which the **KIC.0** and **KIC.E** parameters apply. The value must be in the range of –6 to +6. **Units:** none **Default:** none
C.STATE	number	The vacancy charge state to which the **DC.0**, **DC.E**, **FRAC.0**, **FRAC.E**, **KIC.0**, **KIC.E**, **ECLUST.0**, and **ECLUST.E** parameters apply. The value must be in the range of –6 to +6. **Units:** none **Default:** none **Synonym:** **C.V**
C.ALL	number	Specifies that the **DC.0**, **DC.E**, **FRAC.0**, **FRAC.E**, **KIC.0**, **KIC.E**, **ECLUST.0**, and **ECLUST.E** parameters should be used for all charge states. **Default:** false
DC.0	number	The pre-exponential constant for diffusion of vacancies in the charge state(s) specified by **C.STATE** (or **C.ALL**) in the specified material. Only used with **SILICON**. **Units:** none or microns^2/min or cm^2/sec **Default:** the current value for this material; initially 1.0 for all charge states
DC.E	number	The activation energy for diffusion of vacancies in the charge state(s) specified by **C.STATE** (or **C.ALL**) in the specified material. Only used with **SILICON**. **Units:** electron volts **Default:** the current value for this material; initially 0.0 for all charge states
FRAC.0	number	The pre-exponential constant for the relative concentration of vacancies in the charge state **C.STATE** (or **C.ALL**). **Units:** none **Default:** the current value for this material
FRAC.E	number	The activation energy for the relative concentration of vacancies in the charge state **C.STATE** (or **C.ALL**). **Units:** electron volts **Default:** the current value for this material
KIC.0	number	The pre-exponential constant for the bulk recombination factor for interstitials in charge state **C.I** and small vacancy clusters in charge state **C.V** (**C.STATE**) (or for all combinations of charge states if **C.ALL** is set) in the specified material. Only used with **SILICON**. **Units:** none **Default:** the current value for this material

Parameter	Type	Definition
`KIC.E`	number	The activation energy for the bulk recombination factor for interstitials in charge state **`C.I`** and small vacancy clusters in charge state **`C.V`** (**`C.STATE`**) (or for all combinations of charge states if **`C.ALL`** is set) in the specified material. Only used with **`SILICON`**. **Units:** electron volts **Default:** the current value for this material
`ECLUST.0`	number	The pre-exponential constant for the concentration of vacancies in small clusters in charge state **`C.STATE`** at thermal equilibrium in the specified material. Only used with **`SILICON`**. **Units:** vacancies/cm^3 **Default:** the current value for this material
`ECLUST.E`	number	The activation energy for the concentration of vacancies in small clusters in charge state **`C.STATE`** at thermal equilibrium in the specified material. Only used with **`SILICON`**. **Units:** electron volts **Default:** the current value for this material
`ECLUST.N`	number	The number of vacancies in each small cluster. **Units:** none **Default:** the current value for this material
`/MATERIA`	character	The name of the second material for specifying interface injection and recombination parameters. **Default:** none
`/SILICON`	logical	The interface injection and recombination parameters apply to the interface between the specified material and silicon. **Default:** false
`/OXIDE`	logical	The interface injection and recombination parameters apply to the interface between the specified material and oxide. **Default:** false
`/OXYNITR`	logical	The interface injection and recombination parameters apply to the interface between the specified material and oxynitride. **Default:** false
`/NITRIDE`	logical	The interface injection and recombination parameters apply to the interface between the specified material and nitride. **Default:** false
`/POLYSIL`	logical	The interface injection and recombination parameters apply to the interface between the specified material and polysilicon. **Default:** false
`/AMBIENT`	logical	The interface injection and recombination parameters apply to the interface between the specified material and the ambient gas. **Default:** false **Synonym: `/GAS`**

Parameter	Type	Definition
`V.MAXOX`	logical	Use the injection/recombination model, in which the interface velocity is normalized by the maximum interface velocity in the structure (the model used in older versions of TSUPREM-4). **Default:** the current value for these materials
`V.INITOX`	logical	Use the injection/recombination model, in which the interface velocity is normalized by the initial growth velocity for a bare silicon surface. **Default:** the current value for these materials
`V.NORM`	logical	Use the injection/recombination model, in which the interface velocity is normalized by the value specified by **`VNORM.0`** and **`VNORM.E`** parameters. **Default:** the current value for these materials
`KSURF.0`	number	The pre-exponential constant for the surface recombination velocity at the interface between the specified materials under inert conditions. **Units:** microns/min or cm/sec **Default:** the current value for these materials **Synonyms: `KSMIN.0`, `KV.MIN`**
`KSURF.E`	number	The activation energy for the surface recombination velocity at the interface between the specified materials under inert conditions. **Units:** electron volts **Default:** the current value for these materials **Synonym: `KSMIN.E`**
`KSVEL.0`	number	The pre-exponential constant for the growth-rate-dependent component of the surface recombination velocity at the interface between the specified materials. Used only with the **`V.MAXOX`** and **`V.NORM`** models. **Units:** microns/min or cm/sec **Default:** the current value for these materials **Synonyms: `KSMAX.0`, `KV.MAX`**
`KSVEL.E`	number	The activation energy for the growth-rate-dependent component of the surface recombination velocity at the interface between the specified materials. Used only with the **`V.MAXOX`** and **`V.NORM`** models. **Units:** electron volts **Default:** the current value for these materials **Synonym: `KSMAX.E`**
`KSRAT.0`	number	The pre-exponential constant for the ratio of the growth-rate-dependent component of the surface recombination velocity to the inert component at the interface between the specified materials. Used only with the **`V.INITOX`** model. **Units:** none **Default:** the current value for these materials
`KSRAT.E`	number	The activation energy for the ratio of the growth-rate-dependent component of the surface recombination velocity to the inert component at the interface between the specified materials. Used only with the **`V.INITOX`** model. **Units:** electron volts **Default:** the current value for these materials

Parameter	Type	Definition
VNORM.0	number	The pre-exponential constant for the normalization velocity in the **V.NORM** model. **Units:** microns/min or cm/sec **Default:** the current value for these materials
VNORM.E	number	The activation energy for the normalization velocity in the **V.NORM** model. **Units:** electron volts **Default:** the current value for these materials
GROWTH	logical	Specifies that vacancy injection should be calculated using the computed velocities of the interface rather than the analytic model. **Default:** the current value for these materials
THETA.0	number	The pre-exponential constant for the number of vacancies injected into the bulk per consumed silicon atom during oxidation. **Units:** none, for **V.INITOX** and **V.NORM** models; (microns/min)$^{(1-\mathbf{KPOW})}$ or (cm/sec)$^{(1-\mathbf{KPOW})}$ for **V.MAXOX** model **Default:** the current value for these materials **Synonyms: THETA**
THETA.E	number	The activation energy for the number of vacancies injected into the bulk per consumed silicon atom during oxidation. **Units:** electron volts **Default:** the current value for these materials
A.0	number	The pre-exponential constant for the injection rate of vacancies at the interface between the specified materials. **Units:** #/micron^2/min$^{(1+\mathbf{GPOW})}$ or #/cm^2/sec$^{(1+\mathbf{GPOW})}$ for **V.INITOX** and **V.NORM** models; #/micron^2/min$^{(1-\mathbf{KPOW})}$ or #/cm^2/sec$^{(1-\mathbf{KPOW})}$ for **V.MAXOX** model **Default:** the current value for these materials **Synonym: A**
A.E	number	The activation energy for the injection rate of vacancies at the interface between the specified materials. **Units:** electron volts **Default:** the current value for these materials
T0.0	number	The pre-exponential constant for the time constant for injection at the interface between the specified materials. **Units:** minutes or seconds **Default:** the current value for these materials **Synonym: T0**
T0.E	number	The activation energy for the time constant for injection at the interface between the specified materials. **Units:** electron volts **Default:** the current value for these materials

Parameter	Type	Definition
`KPOW.0`	number	The pre-exponential constant in the expression for the exponent in the surface recombination models. (Also used for injection with the **`V.MAXOX`** model.) **Units:** none **Default:** the current value for these materials **Synonyms:** **`TPOW.0`**, **`POWER`**
`KPOW.E`	number	The activation energy in the expression for the exponent in the surface recombination models. (Also used for injection with the **`V.MAXOX`** model.) **Units:** electron volts **Default:** the current value for these materials **Synonym:** **`TPOW.E`**
`GPOW.0`	number	The pre-exponential constant in the expression for the exponent in the **`V.INITOX`** and **`V.NORM`** models for vacancy injection. **Units:** none **Default:** the current value for these materials
`GPOW.E`	number	The activation energy in the expression for the exponent in the **`V.INITOX`** and **`V.NORM`** models for vacancy injection. **Units:** electron volts **Default:** the current value for these materials
`CL.KRV.0`	number	Prefactor for the reaction of vacancy recombination with the one-moment interstitial clusters. **Default:** 0
`CL.KRV.E`	number	Activation energy for the reaction of vacancy recombination with the one-moment interstitial clusters. **Default:** 0
`CL.VRV`	number	Exponent for the total vacancy concentration in the vacancy recombination term with the one-moment interstitial clusters. **Default:** 0
`CL.VSRV`	number	Exponent for the equilibrium vacancy concentration in the vacancy recombination term with the one-moment interstitial clusters. **Default:** 0
`CL.CRV`	number	Exponent for the interstitial cluster concentration in the vacancy recombination term with the one-moment interstitial clusters. **Default:** 0 **Synonym**: **`CL.CV`**
`CL.KFV.0`	number	Prefactor for the reaction of vacancy generation by one-moment interstitial clusters. **Default:** 0 **Synonym**: **`CL.KGV.0`**
`CL.KFV.E`	number	Activation energy for the reaction of vacancy generation by one-moment interstitial clusters. **Default:** 0 **Synonym**: **`CL.KGV.E`**

Parameter	Type	Definition
`CL.VSFV`	number	Exponent for the equilibrium vacancy concentration in the vacancy generation term with the one-moment interstitial clusters. **Default:** 0
`CL.CFV`	number	Exponent for the interstitial cluster concentration in the vacancy generation term by one-moment interstitial clusters. **Default:** 0 **Synonym**: **`CL.CG`**
`CM.SEC`	logical	If true, parameters involving time are specified in centimeters and seconds; if false, parameters involving time are in microns and minutes. **Default:** false

Description

This statement specifies values for coefficients of vacancy diffusion, recombination, injection, and equilibrium concentration. These parameters are normally specified in the *s4init* file (which is read when the program starts up) but can be changed by you at any time. Values that have *not* been set in *s4init* or by the user, default to 0.0.

Parameters whose units include time are specified in units of microns and minutes, unless **`CM.SEC`** is true, in which case units of centimeters and seconds are assumed.

The interpretation of the various parameters on the **`VACANCY`** statement is described in Chapter 3, Dose Loss Model, p. 3-58.

Bulk and Interface Parameters

If bulk parameters (e.g., **`D.0`** or **`CEQUIL.E`**) are specified but no material is given, the parameters are assumed to apply to silicon. If interface parameters (e.g., **`KSVEL.0`** or **`A.E`**) are specified and only a “first” material (e.g., **`OXIDE`**) is given, the parameters are assumed to apply to the interface between silicon and the specified material. This usage is not recommended, and is intended only for compatibility with older releases of TSUPREM-4.

Examples

1. The following statement specifies the silicon diffusivity and equilibrium values for vacancies:

```
VACANCY SILICON D.0=3000 CEQUIL.0=1.0e13
```

2. The following statement causes the Si/SiO_2 interface injection to be computed using the oxide growth velocity, with 1% of consumed silicon injected as vacancies

```
VACANCY SILICON /OXIDE GROWTH THETA.0=0.01
```

3. The following statement specifies that the surface recombination velocity in the silicon at the interface with nitride is 3.5×10^{-3} cm/s:

```
VACANCY SILICON /NITRIDE CM.SEC +
   KSURF.0=3.5e-3 KSURF.E=0.0
```

4. The following statement specifies that the ratio of doubly-positive vacancies to neutral vacancies is 0.1 under intrinsic conditions:

   ```
   VACANCY SILICON C.STATE=2 FRAC.0=0.1 FRAC.E=0.0
   ```

   The **C.STATE**, **FRAC.0**, and **FRAC.E** parameters can be used to specify the fraction of interstitials in any charge state from -6 to +6.

Additional VACANCY Notes

- Coefficients can be specified for each of the materials, but some coefficients are only used for silicon and others apply to interfaces between silicon and other materials.
- **D.0** and **DC.0** are used together to specify the diffusivity of vacancies in each charge state. One of these should have the units of cm^2/sec or $microns^2$/min, while the other should be unitless. See Chapter 3, Reaction of Pairs with Point Defects, p. 3-64.

ANTIMONY

The **ANTIMONY** statement sets some of the properties of antimony.

```
ANTIMONY

    {MATERIAL=<c> | SILICON | OXIDE | OXYNITRI | NITRIDE | POLYSILI
    | AMBIENT}
    [DIX.0=<n>] [DIX.E=<n>] [DIM.0=<n>] [DIM.E=<n>]
    [DVX.0=<n>] [DVX.E=<n>] [DVM.0=<n>] [DVM.E=<n>]
    [DIPAIR.0=<n>] [DIPAIR.E=<n>] [DVPAIR.0=<n>] [DVPAIR.E=<n>]
    [R.I.S=<n>] [E.I.S=<n>] [R.V.S=<n>] [E.V.S=<n>]
    [R.IP.V=<n>] [E.IP.V=<n>] [R.VP.I=<n>] [E.VP.I=<n>]
    [ { IP.V.LOW | IP.V.MED | IP.V.HIG } ]
    [ { VP.I.LOW | VP.I.MED | VP.I.HIG } ]
    [SS.CLEAR] [SS.TEMP=<n> SS.CONC=<n>]
    [ {/MATERIA=<c> | /SILICON | /OXIDE | /OXYNITR | /NITRIDE | /POLYSIL
      | /AMBIENT}
      [SEG.0=<n>] [SEG.E=<n>] [TRANS.0=<n>] [TRANS.E=<n>]
      [KSIP.0=<n>] [KSIP.E=<n>] [KSVP.0=<n>] [KSVP.E=<n>]
    ]
    [ES.RAND=<n> [ES.F.RAN=<n>]] [ES.100=<n> [ES.F.100=<n>]]
    [ES.110=<n> [ES.F.110=<n>]]
    [CM.SEC]
```

Parameter	Type	Definition
MATERIAL	character	The name of the material to which the other parameters apply (material 1 for the segregation terms). **Default:** none
SILICON	logical	Specifies that other parameters in this statement apply to antimony in silicon, and that silicon is material 1 for the segregation terms. **Default:** true if no other first material is specified.
OXIDE	logical	Specifies that other parameters in this statement apply to antimony in oxide, and that oxide is material 1 for the segregation terms. **Default:** false
OXYNITRI	logical	Specifies that other parameters in this statement apply to antimony in oxynitride, and that oxynitride is material 1 for the segregation terms. **Default:** false
NITRIDE	logical	Specifies that other parameters in this statement apply to antimony in nitride, and that nitride is material 1 for the segregation terms. **Default:** false
POLYSILI	logical	Specifies that other parameters in this statement apply to antimony in polysilicon, and that polysilicon is material 1 for the segregation terms. **Default:** false

Parameter	Type	Definition
AMBIENT	logical	Specifies that the ambient gas is material 1 for the segregation terms. **Default:** false **Synonym: GAS**
DIX.0	number	The pre-exponential constant for diffusion of antimony with neutral interstitials. **Units:** microns^2/min or cm^2/sec **Default:** the current value for this material
DIX.E	number	The activation energy for diffusion of antimony with neutral interstitials. **Units:** electron volts **Default:** the current value for this material
DIM.0	number	The pre-exponential constant for diffusion of antimony with singly-negative interstitials. **Units:** microns^2/min or cm^2/sec **Default:** the current value for this material
DIM.E	number	The activation energy for diffusion of antimony with singly-negative interstitials. **Units:** electron volts **Default:** the current value for this material
DVX.0	number	The pre-exponential constant for diffusion of antimony with neutral vacancies. **Units:** microns^2/min or cm^2/sec **Default:** the current value for this material
DVX.E	number	The activation energy for diffusion of antimony with neutral vacancies. **Units:** electron volts **Default:** the current value for this material
DVM.0	number	The pre-exponential constant for diffusion of antimony with singly-negative vacancies. **Units:** microns^2/min or cm^2/sec **Default:** the current value for this material
DVM.E	number	The activation energy for diffusion of antimony with singly-negative vacancies. **Units:** electron volts **Default:** the current value for this material
DIPAIR.0	number	The pre-exponential constant for the diffusivity of antimony-interstitial pairs. **Units:** microns^2/min or cm^2/sec **Default:** the current value for this material
DIPAIR.E	number	The activation energy for the diffusivity of antimony-interstitial pairs. **Units:** electron volts **Default:** the current value for this material
DVPAIR.0	number	The pre-exponential constant for the diffusivity of antimony-vacancy pairs. **Units:** microns^2/min or cm^2/sec **Default:** the current value for this material
DVPAIR.E	number	The activation energy for the diffusivity of antimony-vacancy pairs. **Units:** electron volts **Default:** the current value for this material

Parameter	Type	Definition
`R.I.S`	number	The capture radius for the reaction between interstitials and substitutional antimony atoms. **Units:** Å **Default:** the current value for this material
`E.I.S`	number	The barrier energy for the reaction between interstitials and substitutional antimony atoms. **Units:** electron volts **Default:** the current value for this material
`R.V.S`	number	The capture radius for the reaction between vacancies and substitutional antimony atoms. **Units:** Å **Default:** the current value for this material
`E.V.S`	number	The barrier energy for the reaction between vacancies and substitutional antimony atoms. **Units:** electron volts **Default:** the current value for this material
`R.IP.V`	number	The capture radius for the reaction between antimony-interstitial pairs and vacancies. **Units:** Å **Default:** the current value for this material
`E.IP.V`	number	The barrier energy for the reaction between antimony-interstitial pairs and vacancies. **Units:** electron volts **Default:** the current value for this material
`R.VP.I`	number	The capture radius for the reaction between antimony-vacancy pairs and interstitials. **Units:** Å **Default:** the current value for this material
`E.VP.I`	number	The barrier energy for the reaction between antimony-vacancy pairs and interstitials. **Units:** electron volts **Default:** the current value for this material
`IP.V.LOW`	logical	Assumes that only dopant-interstitial pairs and vacancies having opposite charges of equal magnitude react in bulk material. **Default:** true
`IP.V.MED`	logical	Assumes that uncharged dopant-interstitial pairs (or vacancies) can react with vacancies (or dopant-interstitial pairs) in any charged state, and that oppositely charged dopant-interstitial pairs and vacancies can react. **Default:** false
`IP.V.HIG`	logical	Assumes that dopant-vacancy pairs in any charged state can react with interstitials in any charged state. **Default:** false

Parameter	Type	Definition
`VP.I.LOW`	logical	Assumes that only dopant-vacancy pairs and interstitials having opposite charges of equal magnitude react in bulk material. **Default:** true
`VP.I.MED`	logical	Assumes that uncharged dopant-vacancy pairs (or interstitials) can react with interstitials (or dopant-vacancy pairs) in any charged state, and that oppositely charged dopant-vacancy pairs and interstitials can react. **Default:** false
`VP.I.HIG`	logical	Assumes that dopant-vacancy pairs in any charged state can react with interstitials in any charged state. **Default:** false
`SS.CLEAR`	logical	Clears the solid solubility vs. temperature table. **Default:** false
`SS.TEMP`	number	The temperature at which the solid solubility in material 1 is **`SS.CONC`**. **Units:** degrees Celsius **Default:** none
`SS.CONC`	number	The solid solubility in material 1 at temperature **`SS.TEMP`**. **Units:** atoms/cm^3 **Default:** none
`/MATERIA`	character	The name of material 2 for the segregation parameters. **Default:** none
`/SILICON`	logical	Specifies that segregation parameters given on this statement apply to silicon as material 2. **Default:** false
`/OXIDE`	logical	Specifies that segregation parameters given on this statement apply to oxide as material 2. **Default:** True if no other second material is specified.
`/OXYNITR`	logical	Specifies that segregation parameters given on this statement apply to oxynitride as material 2. **Default:** true
`/NITRIDE`	logical	Specifies that segregation parameters given on this statement apply to nitride as material 2. **Default:** false
`/POLYSIL`	logical	Specifies that segregation parameters given on this statement apply to polysilicon as material 2. **Default:** false
`/AMBIENT`	logical	Specifies that segregation parameters given on this statement apply to the ambient gas as material 2. **Default:** false **Synonym:** **`/GAS`**

Parameter	Type	Definition
`SEG.0`	number	The pre-exponential factor for segregation from material 1 to material 2. **Units:** none **Default:** the current value for these materials
`SEG.E`	number	The activation energy for segregation from material 1 to material 2. **Units:** electron volts **Default:** the current value for these materials
`TRANS.0`	number	The pre-exponential factor for transport from material 1 to material 2. **Units:** microns/min or cm/sec **Default:** the current value for these materials **Synonym: `TRN.0`**
`TRANS.E`	number	The activation energy for transport from material 1 to material 2. **Units:** electron volts **Default:** the current value for these materials **Synonym: `TRN.E`**
`KSIP.0`	number	The pre-exponential constant for the surface recombination velocity of the dopant-interstitial pair at the interface between the specified materials. **Units:** microns/min or cm/sec **Default:** the current value for these materials
`KSIP.E`	number	The activation energy for the surface recombination velocity of the dopant-interstitial pair at the interface between the specified materials **Units:** electron volts **Default:** the current value for these materials
`KSVP.0`	number	The pre-exponential constant for the surface recombination velocity of the dopant-vacancy pair at the interface between the specified materials. **Units:** microns/min or cm/sec **Default:** the current value for these materials
`KSVP.E`	number	The activation energy for the surface recombination velocity of the dopant-vacancy pair at the interface between the specified materials **Units:** electron volts **Default:** the current value for these materials
`ES.RAND`	number	The electronic stopping power coefficient of implanted antimony in the specified material for materials other than silicon and for a nonchanneled direction in silicon. This value is used for the MC ion implant calculation only. **Units:** angstrom^2*eV$^{(1-\mathtt{ES.F.RAN})}$ **Default:** the current value for antimony and the specified material
`ES.F.RAN`	number	The exponent of the electronic stopping power of implanted antimony in the specified material for materials other than silicon and for a nonchanneled direction in silicon. This value is used for the MC ion implant calculation only. **Units:** none **Default:** the current value for antimony and the specified material

Parameter	Type	Definition
ES.100	number	The electronic stopping power of antimony in silicon along the <100> channeling axes. This value is used for the MC ion implant calculation only. **Units:** angstrom^2*eV$^{(1\text{-}\mathbf{ES.F.100})}$ **Default:** the current value for antimony and the specified material
ES.F.100	number	The exponent of the electronic stopping power of antimony in silicon along the <100> crystal axes. This value is used for the MC ion implant calculation only. **Units:** none **Default:** the current value for antimony and the specified material
ES.110	number	The electronic stopping power of antimony in silicon along the <110> channeling axes. This value is used for the MC ion implant calculation only. **Units:** angstrom^2*eV$^{(1\text{-}\mathbf{ES.F.100})}$ **Default:** the current value for antimony and the specified material
ES.F.110	number	The exponent of the electronic stopping power of antimony in silicon along the <110> crystal axes. This value is used for the MC ion implant calculation only. **Units:** none **Default:** the current value for antimony and the specified material
CM.SEC	logical	If true, parameters involving time are specified in centimeters and seconds; if false, parameters involving time are in microns and minutes. **Default:** false

Description

This statement specifies properties and model coefficients for antimony. The values of the diffusivity, reaction constant, solid solubility, and electronic stopping parameters apply in material 1 (specified without the “/”), while **SEG.0**, **SEG.E**, **TRANS.0**, and **TRANS.E** apply at the interface between material 1 and material 2 (specified with the “/”). These coefficients are normally given in the *s4init* file (which is read at the start of each TSUPREM-4 execution) but can be changed by you at any time. Coefficients that are *not* given in the *s4init* file or set by the user, default to 0.0, except for **SEG.0** that defaults to 1.0.

The newer **IMPURITY** statement can be used to set all of the properties of antimony, including some that cannot be set with the **ANTIMONY** statement.

Parameters whose units include time are specified in units of microns and minutes, unless **CM.SEC** is true, in which case units of centimeters and seconds are assumed.

See Also

For additional information see the following sections:

- Chapter 3, Diffusion, p. 3-14 for diffusion and segregation parameters.
- Chapter 3, MC Ion Implant Model, p. 3-131 for electronic stopping power parameters.

Examples

1. The following statement specifies the diffusivity of antimony diffusing with neutral vacancies in silicon:

   ```
   ANTIMONY SILICON DVX.0=1.22e9 DVX.E=3.65
   ```

2. The following statement specifies the segregation parameters at the Si/SiO$_2$ interface:

   ```
   ANTIMONY SILICON /OXIDE SEG.0=30.0 TRANS.0=0.1
   ```

   The concentration in silicon is 30.0 times the concentration in oxide, at equilibrium.

Additional ANTIMONY Notes

- The **ANTIMONY** statement has been made obsolete by the **IMPURITY** statement, but remains available for compatibility with existing input files. *Note* that some properties of antimony can only be set on the **IMPURITY** statement.
- The fractional interstitial parameter **FI** that was used in TSUPREM-4 before Version 6.0 is no longer supported. Instead, it is now necessary to specify the diffusivities with interstitials and vacancies separately.

ARSENIC

The **`ARSENIC`** statement sets some of the properties of arsenic.

```
ARSENIC

   {MATERIAL=<c> | SILICON | OXIDE | OXYNITRI | NITRIDE | POLYSILI
   | AMBIENT}
   [DIX.0=<n>] [DIX.E=<n>] [DIM.0=<n>] [DIM.E=<n>]
   [DVX.0=<n>] [DVX.E=<n>] [DVM.0=<n>] [DVM.E=<n>]
   [DIPAIR.0=<n>] [DIPAIR.E=<n>] [DVPAIR.0=<n>] [DVPAIR.E=<n>]
   [R.I.S=<n>] [E.I.S=<n>] [R.V.S=<n>] [E.V.S=<n>]
   [R.IP.V=<n>] [E.IP.V=<n>] [R.VP.I=<n>] [E.VP.I=<n>]
   [ { IP.V.LOW | IP.V.MED | IP.V.HIG } ]
   [ { VP.I.LOW | VP.I.MED | VP.I.HIG } ]
   [CTN.0=<n>] [CTN.E=<n>] [CTN.F=<n>]
   [ {/MATERIA=<c> | /SILICON | /OXIDE | /OXYNITR | /NITRIDE | /POLYSIL
     | /AMBIENT}
     [SEG.0=<n>] [SEG.E=<n>] [TRANS.0=<n>] [TRANS.E=<n>]
     [KSIP.0=<n>] [KSIP.E=<n>] [KSVP.0=<n>] [KSVP.E=<n>]
   ]
   [ES.RAND=<n> [ES.F.RAN=<n>]] [ES.100=<n> [ES.F.100=<n>]]
   [ES.110=<n> [ES.F.110=<n>]]
   [CM.SEC]
```

Parameter	Type	Definition
`MATERIAL`	character	The name of the material to which the other parameters apply (material 1 for the segregation terms). **Default:** none
`SILICON`	logical	Specifies that other parameters in this statement apply to arsenic in silicon, and that silicon is material 1 for the segregation terms. **Default:** true if no other first material is specified.
`OXIDE`	logical	Specifies that other parameters in this statement apply to arsenic in oxide, and that oxide is material 1 for the segregation terms. **Default:** false
`OXYNITRI`	logical	Specifies that other parameters in this statement apply to arsenic in oxynitride, and that oxynitride is material 1 for the segregation terms. **Default:** false
`NITRIDE`	logical	Specifies that other parameters in this statement apply to arsenic in nitride, and that nitride is material 1 for the segregation terms. **Default:** false
`POLYSILI`	logical	Specifies that other parameters in this statement apply to arsenic in polysilicon, and that polysilicon is material 1 for the segregation terms. **Default:** false

Parameter	Type	Definition
`AMBIENT`	logical	Specifies that the ambient gas is material 1 for the segregation terms. **Default:** false **Synonym**: **`GAS`**
`DIX.0`	number	The pre-exponential constant for diffusion of arsenic with neutral interstitials. **Units:** microns^2/min or cm^2/sec **Default:** the current value for this material
`DIX.E`	number	The activation energy for diffusion of arsenic with neutral interstitials. **Units:** electron volts **Default:** the current value for this material
`DIM.0`	number	The pre-exponential constant for diffusion of arsenic with singly-negative interstitials. **Units:** microns^2/min or cm^2/sec **Default:** the current value for this material
`DIM.E`	number	The activation energy for diffusion of arsenic with singly-negative interstitials. **Units:** electron volts **Default:** the current value for this material
`DVX.0`	number	The pre-exponential constant for diffusion of arsenic with neutral vacancies. **Units:** microns^2/min or cm^2/sec **Default:** the current value for this material
`DVX.E`	number	The activation energy for diffusion of arsenic with neutral vacancies. **Units:** electron volts **Default:** the current value for this material
`DVM.0`	number	The pre-exponential constant for diffusion of arsenic with singly-negative vacancies. **Units:** microns^2/min or cm^2/sec **Default:** the current value for this material
`DVM.E`	number	The activation energy for diffusion of arsenic with singly-negative vacancies. **Units:** electron volts **Default:** the current value for this material
`DIPAIR.0`	number	The pre-exponential constant for the diffusivity of arsenic-interstitial pairs. **Units:** microns^2/min or cm^2/sec **Default:** the current value for this material
`DIPAIR.E`	number	The activation energy for the diffusivity of arsenic-interstitial pairs. **Units:** electron volts **Default:** the current value for this material
`DVPAIR.0`	number	The pre-exponential constant for the diffusivity of arsenic-vacancy pairs. **Units:** microns^2/min or cm^2/sec **Default:** the current value for this material
`DVPAIR.E`	number	The activation energy for the diffusivity of arsenic-vacancy pairs. **Units:** electron volts **Default:** the current value for this material

Parameter	Type	Definition
`R.I.S`	number	The capture radius for the reaction between interstitials and substitutional arsenic atoms. **Units:** Å **Default:** the current value for this material
`E.I.S`	number	The barrier energy for the reaction between interstitials and substitutional arsenic atoms. **Units:** electron volts **Default:** the current value for this material
`R.V.S`	number	The capture radius for the reaction between vacancies and substitutional arsenic atoms. **Units:** Å **Default:** the current value for this material
`E.V.S`	number	The barrier energy for the reaction between vacancies and substitutional arsenic atoms. **Units:** electron volts **Default:** the current value for this material
`R.IP.V`	number	The capture radius for the reaction between arsenic-interstitial pairs and vacancies. **Units:** Å **Default:** the current value for this material
`E.IP.V`	number	The barrier energy for the reaction between arsenic-interstitial pairs and vacancies. **Units:** electron volts **Default:** the current value for this material
`R.VP.I`	number	The capture radius for the reaction between arsenic-vacancy pairs and interstitials. **Units:** Å **Default:** the current value for this material
`E.VP.I`	number	The barrier energy for the reaction between arsenic-vacancy pairs and interstitials. **Units:** electron volts **Default:** the current value for this material
`IP.V.LOW`	logical	Assumes that only dopant-interstitial pairs and vacancies having opposite charges of equal magnitude react in bulk material. **Default:** true
`IP.V.MED`	logical	Assumes that uncharged dopant-interstitial pairs (or vacancies) can react with vacancies (or dopant-interstitial pairs) in any charged state, and that oppositely charged dopant-interstitial pairs and vacancies can react. **Default:** false
`IP.V.HIG`	logical	Assumes that dopant-vacancy pairs in any charged state can react with interstitials in any charged state. **Default:** false

Parameter	Type	Definition
`VP.I.LOW`	logical	Assumes that only dopant-vacancy pairs and interstitials having opposite charges of equal magnitude react in bulk material. **Default:** true
`VP.I.MED`	logical	Assumes that uncharged dopant-vacancy pairs (or interstitials) can react with interstitials (or dopant-vacancy pairs) in any charged state, and that oppositely charged dopant-vacancy pairs and interstitials can react. **Default:** false
`VP.I.HIG`	logical	Assumes that dopant-vacancy pairs in any charged state can react with interstitials in any charged state. **Default:** false
`CTN.0`	number	The pre-exponential constant for clustering of arsenic. **Units:** $(\text{atoms/cm}^3)^{(1/\texttt{CTN.F}-1)}$ **Default:** the current value for this material
`CTN.E`	number	The activation energy for clustering of arsenic. **Units:** electron volts **Default:** the current value for this material
`CTN.F`	number	The exponent of concentration for clustering of arsenic. **Units:** none **Default:** the current value for this material
`/MATERIA`	character	The name of material 2 for the segregation parameters. **Default:** none
`/SILICON`	logical	Specifies that segregation parameters given on this statement apply to silicon as material 2. **Default:** false
`/OXIDE`	logical	Specifies that segregation parameters given on this statement apply to oxide as material 2. **Default:** True if no other second material is specified.
`/OXYNITR`	logical	Specifies that segregation parameters given on this statement apply to oxynitride as material 2. **Default:** false
`/NITRIDE`	logical	Specifies that segregation parameters given on this statement apply to nitride as material 2. **Default:** false
`/POLYSIL`	logical	Specifies that segregation parameters given on this statement apply to polysilicon as material 2. **Default:** false
`/AMBIENT`	logical	Specifies that segregation parameters given on this statement apply to the ambient gas as material 2. **Default:** false **Synonyms**: `/GAS`

Parameter	Type	Definition
`SEG.0`	number	The pre-exponential factor for segregation from material 1 to material 2. **Units:** none **Default:** the current value for these materials
`SEG.E`	number	The activation energy for segregation from material 1 to material 2. **Units:** electron volts **Default:** the current value for these materials
`TRANS.0`	number	The pre-exponential factor for transport from material 1 to material 2. **Units:** microns/min or cm/sec **Default:** the current value for these materials **Synonym**: **`TRN.0`**
`TRANS.E`	number	The activation energy for transport from material 1 to material 2. **Units:** electron volts **Default:** the current value for these materials **Synonym**: **`TRN.E`**
`KSIP.0`	number	The pre-exponential constant for the surface recombination velocity of the dopant-interstitial pair at the interface between the specified materials. **Units:** microns/min or cm/sec **Default:** the current value for these materials
`KSIP.E`	number	The activation energy for the surface recombination velocity of the dopant-interstitial pair at the interface between the specified materials. **Units:** electron volts **Default:** the current value for these materials
`KSVP.0`	number	The pre-exponential constant for the surface recombination velocity of the dopant-vacancy pair at the interface between the specified materials. **Units:** microns/min or cm/sec **Default:** the current value for these materials
`KSVP.E`	number	The activation energy for the surface recombination velocity of the dopant-vacancy pair at the interface between the specified materials. **Units:** electron volts **Default:** the current value for these materials
`ES.RAND`	number	The electronic stopping power coefficient of implanted arsenic in the specified material for materials other than silicon and for a nonchanneled direction in silicon. This value is used for the MC ion implant calculation only. **Units:** angstrom^2*eV$^{(1-\mathtt{ES.F.RAN})}$ **Default:** the current value for arsenic and the specified material
`ES.F.RAN`	number	The exponent of the electronic stopping power of implanted arsenic in the specified material for materials other than silicon and for a nonchanneled direction in silicon. This value is used for the MC ion implant calculation only. **Units:** none **Default:** the current value for arsenic and the specified material

Parameter	Type	Definition
`ES.100`	number	The electronic stopping power of arsenic in silicon along the <100> channeling axes. This value is used for the MC ion implant calculation only. **Units:** angstrom^2*eV$^{(1-\mathtt{ES.F.100})}$ **Default:** the current value for arsenic and the specified material
`ES.F.100`	number	The exponent of the electronic stopping power of arsenic in silicon along the <100> crystal axes. This value is used for the MC ion implant calculation only. **Units:** none **Default:** the current value for arsenic and the specified material
`ES.110`	number	The electronic stopping power of arsenic in silicon along the <110> channeling axes. This value is used for the MC ion implant calculation only. **Units:** angstrom^2*eV$^{(1-\mathtt{ES.F.100})}$ **Default:** the current value for arsenic and the specified material
`ES.F.110`	number	The exponent of the electronic stopping power of arsenic in silicon along the <110> crystal axes. This value is used for the MC ion implant calculation only. **Units:** none **Default:** the current value for arsenic and the specified material
`CM.SEC`	logical	If true, parameters involving time are specified in centimeters and seconds; if false, parameters involving time are in microns and minutes. **Default:** false

Description

This statement specifies properties and model coefficients for arsenic. The values of the diffusivity, reaction constant, clustering, and electronic stopping parameters apply in material 1 (specified without the “/”), while **`SEG.0`**, **`SEG.E`**, **`TRANS.0`** and **`TRANS.E`** apply at the interface between material 1 and material 2 (specified with the “/”). These coefficients are normally given in the *s4init* file (which is read at the start of each TSUPREM-4 execution), but can be changed by you at any time. Coefficients that are *not* given in the *s4init* file or set by the user, default to 0.0, except for **`SEG.0`** that defaults to 1.0.

The newer **`IMPURITY`** statement can be used to set all of the properties of arsenic, including some that cannot be set with the **`ARSENIC`** statement.

Parameters whose units include time are specified in units of microns and minutes, unless **`CM.SEC`** is true, in which case units of centimeters and seconds are assumed.

See Also

For additional information see the following sections:

- Chapter 3, Diffusion, p. 3-14 for diffusion and segregation parameters.
- Chapter 3, MC Ion Implant Model, p. 3-131 for electronic stopping power parameters.

Examples

1. The following statement specifies the diffusivity of arsenic diffusing with negative vacancies in silicon:

2. **`ARSENIC SILICON DVM.0=1.49e11 DVM.E=4.15`**

3. The following statement specifies the segregation parameters at the Si/SiO$_2$ interface:

   `ARSENIC SILICON /OXIDE SEG.0=30.0 TRANS.0=0.1`

   The concentration in silicon is 30.0 times the concentration in oxide, at equilibrium.

Additional ARSENIC Notes

1. The **`ARSENIC`** statement has been obsoleted by the **`IMPURITY`** statement, but remains available for compatibility with existing input files. *Note* that some properties of arsenic can only be set on the **`IMPURITY`** statement.

2. The fractional interstitial parameter **`FI`** that was used in TSUPREM-4 before Version 6.0 is no longer supported. Instead, it is now necessary to specify the diffusivities with interstitials and vacancies separately.

BORON

The **BORON** statement sets some of the properties of boron.

```
BORON

   {MATERIAL=<c> | SILICON | OXIDE | OXYNITRI | NITRIDE | POLYSILI
   | AMBIENT}
   [DIX.0=<n>] [DIX.E=<n>] [DIP.0=<n>] [DIP.E=<n>]
   [DVX.0=<n>] [DVX.E=<n>] [DVP.0=<n>] [DVP.E=<n>]
   [DIPAIR.0=<n>] [DIPAIR.E=<n>] [DVPAIR.0=<n>] [DVPAIR.E=<n>]
   [R.I.S=<n>] [E.I.S=<n>] [R.V.S=<n>] [E.V.S=<n>]
   [R.IP.V=<n>] [E.IP.V=<n>] [R.VP.I=<n>] [E.VP.I=<n>]
   [ { IP.V.LOW | IP.V.MED | IP.V.HIG } ]
   [ { VP.I.LOW | VP.I.MED | VP.I.HIG } ]
   [SS.CLEAR] [SS.TEMP=<n> SS.CONC=<n>]
   [ {/MATERIA=<c> | /SILICON | /OXIDE | /OXYNITR | /NITRIDE | /POLYSIL
     | /AMBIENT}
     [SEG.0=<n>] [SEG.E=<n>] [TRANS.0=<n>] [TRANS.E=<n>]
     [KSIP.0=<n>] [KSIP.E=<n>] [KSVP.0=<n>] [KSVP.E=<n>]
   ]
   [ES.RAND=<n> [ES.F.RAN=<n>]] [ES.100=<n> [ES.F.100=<n>]]
   [ES.110=<n> [ES.F.110=<n>]]
   [CM.SEC]
```

Parameter	Type	Definition
MATERIAL	character	The name of the material to which the other parameters apply (material 1 for the segregation terms). **Default:** none
SILICON	logical	Specifies that other parameters in this statement apply to boron in silicon, and that silicon is material 1 for the segregation terms. **Default:** True if no other first material is specified.
OXIDE	logical	Specifies that other parameters in this statement apply to boron in oxide, and that oxide is material 1 for the segregation terms. **Default:** false
OXYNITRI	logical	Specifies that other parameters in this statement apply to boron in oxynitride, and that oxynitride is material 1 for the segregation terms. **Default:** false
NITRIDE	logical	Specifies that other parameters in this statement apply to boron in nitride, and that nitride is material 1 for the segregation terms. **Default:** false
POLYSILI	logical	Specifies that other parameters in this statement apply to boron in polysilicon, and that polysilicon is material 1 for the segregation terms. **Default:** false

Parameter	Type	Definition
AMBIENT	logical	Specifies that the ambient gas is material 1 for the segregation terms. **Default:** false **Synonym:** **GAS**
DIX.0	number	The pre-exponential constant for diffusion of boron with neutral interstitials. **Units:** microns^2/min or cm^2/sec **Default:** the current value for this material
DIX.E	number	The activation energy for diffusion of boron with neutral interstitials. **Units:** electron volts **Default:** the current value for this material
DIP.0	number	The pre-exponential constant for diffusion of boron with singly-positive interstitials. **Units:** microns^2/min or cm^2/sec **Default:** the current value for this material
DIP.E	number	The activation energy for diffusion of boron with singly-positive interstitials. **Units:** electron volts **Default:** the current value for this material
DVX.0	number	The pre-exponential constant for diffusion of boron with neutral vacancies. **Units:** microns^2/min or cm^2/sec **Default:** the current value for this material
DVX.E	number	The activation energy for diffusion of boron with neutral vacancies. **Units:** electron volts **Default:** the current value for this material
DVP.0	number	The pre-exponential constant for diffusion of boron with singly-positive vacancies. **Units:** microns^2/min or cm^2/sec **Default:** the current value for this material
DVP.E	number	The activation energy for diffusion of boron with singly-positive vacancies. **Units:** electron volts **Default:** the current value for this material
DIPAIR.0	number	The pre-exponential constant for the diffusivity of boron-interstitial pairs. **Units:** microns^2/min or cm^2/sec **Default:** the current value for this material
DIPAIR.E	number	The activation energy for the diffusivity of boron-interstitial pairs. **Units:** electron volts **Default:** the current value for this material
DVPAIR.0	number	The pre-exponential constant for the diffusivity of boron-vacancy pairs. **Units:** microns^2/min or cm^2/sec **Default:** the current value for this material
DVPAIR.E	number	The activation energy for the diffusivity of boron-vacancy pairs. **Units:** electron volts **Default:** the current value for this material

Parameter	Type	Definition
`R.I.S`	number	The capture radius for the reaction between interstitials and substitutional boron atoms. **Units:** Å **Default:** the current value for this material
`E.I.S`	number	The barrier energy for the reaction between interstitials and substitutional boron atoms. **Units:** electron volts **Default:** the current value for this material
`R.V.S`	number	The capture radius for the reaction between vacancies and substitutional boron atoms. **Units:** Å **Default:** the current value for this material
`E.V.S`	number	The barrier energy for the reaction between vacancies and substitutional boron atoms. **Units:** electron volts **Default:** the current value for this material
`R.IP.V`	number	The capture radius for the reaction between boron-interstitial pairs and vacancies. **Units:** Å **Default:** the current value for this material
`E.IP.V`	number	The barrier energy for the reaction between boron-interstitial pairs and vacancies. **Units:** electron volts **Default:** the current value for this material
`R.VP.I`	number	The capture radius for the reaction between boron-vacancy pairs and interstitials. **Units:** Å **Default:** the current value for this material
`E.VP.I`	number	The barrier energy for the reaction between boron-vacancy pairs and interstitials. **Units:** electron volts **Default:** the current value for this material
`IP.V.LOW`	logical	Assumes that only dopant-interstitial pairs and vacancies having opposite charges of equal magnitude react in bulk material. **Default:** true
`IP.V.MED`	logical	Assumes that uncharged dopant-interstitial pairs (or vacancies) can react with vacancies (or dopant-interstitial pairs) in any charged state, and that oppositely charged dopant-interstitial pairs and vacancies can react. **Default:** false
`IP.V.HIG`	logical	Assumes that dopant-vacancy pairs in any charged state can react with interstitials in any charged state. **Default:** false

Parameter	Type	Definition
`VP.I.LOW`	logical	Assumes that only dopant-vacancy pairs and interstitials having opposite charges of equal magnitude react in bulk material. **Default:** true
`VP.I.MED`	logical	Assumes that uncharged dopant-vacancy pairs (or interstitials) can react with interstitials (or dopant-vacancy pairs) in any charged state, and that oppositely charged dopant-vacancy pairs and interstitials can react. **Default:** false
`VP.I.HIG`	logical	Assumes that dopant-vacancy pairs in any charged state can react with interstitials in any charged state. **Default:** false
`SS.CLEAR`	logical	Clears the solid solubility vs. temperature table. **Default:** false
`SS.TEMP`	number	The temperature at which the solid solubility in material 1 is **`SS.CONC`**. **Units:** degrees Celsius **Default:** none
`SS.CONC`	number	The solid solubility in material 1 at temperature **`SS.TEMP`**. **Units:** atoms/cm^3 **Default:** none
`/MATERIA`	character	The name of material 2 for the segregation parameters. **Default:** none
`/SILICON`	logical	Specifies that segregation parameters given on this statement apply to silicon as material 2. **Default:** false
`/OXIDE`	logical	Specifies that segregation parameters given on this statement apply to oxide as material 2. **Default:** true if no other second material is specified
`/OXYNITR`	logical	Specifies that segregation parameters given on this statement apply to oxynitride as material 2. **Default:** false
`/NITRIDE`	logical	Specifies that segregation parameters given on this statement apply to nitride as material 2. **Default:** false
`/POLYSIL`	logical	Specifies that segregation parameters given on this statement apply to polysilicon as material 2. **Default:** false
`/AMBIENT`	logical	Specifies that segregation parameters given on this statement apply to the ambient gas as material 2. **Default:** false **Synonym:** **`/GAS`**

Parameter	Type	Definition
`SEG.0`	number	The pre-exponential factor for segregation from material 1 to material 2. **Units:** none **Default:** the current value for these materials
`SEG.E`	number	The activation energy for segregation from material 1 to material 2. **Units:** electron volts **Default:** the current value for these materials
`TRANS.0`	number	The pre-exponential factor for transport from material 1 to material 2. **Units:** microns/min or cm/sec **Default:** the current value for these materials **Synonym: `TRN.0`**
`TRANS.E`	number	The activation energy for transport from material 1 to material 2. **Units:** electron volts **Default:** the current value for these materials **Synonym: `TRN.E`**
`KSIP.0`	number	The pre-exponential constant for the surface recombination velocity of the dopant-interstitial pair at the interface between the specified materials. **Units:** microns/min or cm/sec **Default:** the current value for these materials
`KSIP.E`	number	The activation energy for the surface recombination velocity of the dopant-interstitial pair at the interface between the specified materials **Units:** electron volts **Default:** the current value for these materials
`KSVP.0`	number	The pre-exponential constant for the surface recombination velocity of the dopant-vacancy pair at the interface between the specified materials. **Units:** microns/min or cm/sec **Default:** the current value for these materials
`KSVP.E`	number	The activation energy for the surface recombination velocity of the dopant-vacancy pair at the interface between the specified materials **Units:** electron volts **Default:** the current value for these materials
`ES.RAND`	number	The electronic stopping power coefficient of implanted boron in the specified material for materials other than silicon and for a nonchanneled direction in silicon. This value is used for the MC ion implant calculation only. **Units:** angstrom^2*eV$^{(1-\mathtt{ES.F.RAN})}$ **Default:** the current value for boron and the specified material
`ES.F.RAN`	number	The exponent of the electronic stopping power of implanted boron in the specified material, for materials other than silicon and for a nonchanneled direction in silicon. This value is used for the MC ion implant calculation only. **Units:** none **Default:** the current value for boron and the specified material

Parameter	Type	Definition
`ES.100`	number	The electronic stopping power of boron in silicon along the <100> channeling axes. This value is used for the MC ion implant calculation only. **Units:** angstrom^2*eV$^{(1-\mathtt{ES.F.100})}$ **Default:** the current value for boron and the specified material
`ES.F.100`	number	The exponent of the electronic stopping power of boron in silicon along the <100> crystal axes. This value is used for the MC ion implant calculation only. **Units:** none **Default:** the current value for boron and the specified material
`ES.110`	number	The electronic stopping power of boron in silicon along the <110> channeling axes. This value is used for the MC ion implant calculation only. **Units:** angstrom^2*eV$^{(1-\mathtt{ES.F.100})}$ **Default:** the current value for boron and the specified material
`ES.F.110`	number	The exponent of the electronic stopping power of boron in silicon along the <110> crystal axes. This value is used for the MC ion implant calculation only. **Units:** none **Default:** the current value for boron and the specified material
`CM.SEC`	logical	If true, parameters involving time are specified in centimeters and seconds; if false, parameters involving time are in microns and minutes. **Default:** false

Description

This statement specifies properties and model coefficients for boron. The values of the diffusivity, reaction constant, solid solubility, and electronic stopping parameters apply in material 1 (specified without the “/”), while **`SEG.0`**, **`SEG.E`**, **`TRANS.0`**, and **`TRANS.E`** apply at the interface between material 1 and material 2 (specified with the “/”). These coefficients are normally given in the *s4init* file (which is read at the start of each TSUPREM-4 execution) but can be changed by you at any time. Coefficients that are *not* given in the *s4init* file or set by the user, default to 0.0, except for **`SEG.0`** that defaults to 1.0.

The newer **`IMPURITY`** statement can be used to set all of the properties of boron, including some that cannot be set with the **`BORON`** statement.

Parameters whose units include time are specified in units of microns and minutes, unless **`CM.SEC`** is true, in which case units of centimeters and seconds are assumed.

See Also

For additional information, see the following sections:

- Diffusion and segregation parameters, Chapter 3, Diffusion, p. 3-14.
- Electronic stopping power parameters, Chapter 3, MC Ion Implant Model, p. 3-131.

Examples

1. The following statement specifies the diffusivity of boron diffusing with neutral interstitials in silicon:

   ```
   BORON SILICON DIX.0=2.09e8 DIX.E=3.46
   ```

2. The following statement specifies the segregation parameters at the Si/SiO$_2$ interface:

   ```
   BORON SILICON /OXIDE SEG.0=0.91 TRANS.0=0.1
   ```

   The concentration in silicon is 0.91 times the concentration in oxide, at equilibrium.

Additional BORON Notes

1. The **BORON** statement has been made obsolete by the **IMPURITY** statement, but remains available for compatibility with existing input files. *Note* that some properties of boron can only be set on the **IMPURITY** statement.
2. The fractional interstitial parameter **FI** that was used in TSUPREM-4 before Version 6.0 is no longer supported. Instead, it is now necessary to specify the diffusivities with interstitials and vacancies separately.

PHOSPHORUS

The **`PHOSPHORUS`** statement sets some of the properties of phosphorus.

```
PHOSPHORUS

    {MATERIAL=<c> | SILICON | OXIDE | OXYNITRI | NITRIDE | POLYSILI
    | AMBIENT}
    [DIX.0=<n>] [DIX.E=<n>] [DIM.0=<n>] [DIM.E=<n>]
    [DIMM.0=<n>] [DIMM.E=<n>] [DVX.0=<n>] [DVX.E=<n>]
    [DVM.0=<n>] [DVM.E=<n>] [DVMM.0=<n>] [DVMM.E=<n>]
    [DIPAIR.0=<n>] [DIPAIR.E=<n>] [DVPAIR.0=<n>] [DVPAIR.E=<n>]
    [R.I.S=<n>] [E.I.S=<n>] [R.V.S=<n>] [E.V.S=<n>]
    [R.IP.V=<n>] [E.IP.V=<n>] [R.VP.I=<n>] [E.VP.I=<n>]
    [ { IP.V.LOW | IP.V.MED | IP.V.HIG } ]
    [ { VP.I.LOW | VP.I.MED | VP.I.HIG } ]
    [SS.CLEAR] [SS.TEMP=<n> SS.CONC=<n>]
    [ {/MATERIA=<c> | /SILICON | /OXIDE | /OXYNITR | /NITRIDE | /POLYSIL
      | /AMBIENT}
      [SEG.0=<n>] [SEG.E=<n>] [TRANS.0=<n>] [TRANS.E=<n>]
      [KSIP.0=<n>] [KSIP.E=<n>] [KSVP.0=<n>] [KSVP.E=<n>]
    ]
    [ES.RAND=<n> [ES.F.RAN=<n>]] [ES.100=<n> [ES.F.100=<n>]]
    [ES.110=<n> [ES.F.110=<n>]]
    [CM.SEC]
```

Parameter	Type	Definition
`MATERIAL`	character	The name of the material to which the other parameters apply (material 1 for the segregation terms). **Default:** none
`SILICON`	logical	Specifies that other parameters in this statement apply to phosphorus in silicon, and that silicon is material 1 for the segregation terms. **Default:** True if no other first material is specified
`OXIDE`	logical	Specifies that other parameters in this statement apply to phosphorus in oxide, and that oxide is material 1 for the segregation terms. **Default:** false
`OXYNITRI`	logical	Specifies that other parameters in this statement apply to phosphorus in oxynitride, and that oxynitride is material 1 for the segregation terms. **Default:** false
`NITRIDE`	logical	Specifies that other parameters in this statement apply to phosphorus in nitride, and that nitride is material 1 for the segregation terms. **Default:** false
`POLYSILI`	logical	Specifies that other parameters in this statement apply to phosphorus in polysilicon, and that polysilicon is material 1 for the segregation terms. **Default:** false

Parameter	Type	Definition
`AMBIENT`	logical	Specifies that the ambient gas is material 1 for the segregation terms. **Default:** false **Synonym: `GAS`**
`DIX.0`	number	The pre-exponential constant for diffusion of phosphorus with neutral interstitials. **Units:** microns^2/min or cm^2/sec **Default:** the current value for this material
`DIX.E`	number	The activation energy for diffusion of phosphorus with neutral interstitials. **Units:** electron volts **Default:** the current value for this material
`DIM.0`	number	The pre-exponential constant for diffusion of phosphorus with singly-negative interstitials. **Units:** microns^2/min or cm^2/sec **Default:** the current value for this material
`DIM.E`	number	The activation energy for diffusion of phosphorus with singly-negative interstitials. **Units:** electron volts **Default:** the current value for this material
`DIMM.0`	number	The pre-exponential constant for diffusion of phosphorus with doubly-negative interstitials. **Units:** microns^2/min or cm^2/sec **Default:** the current value for this material
`DIMM.E`	number	The activation energy for diffusion of phosphorus with doubly-negative interstitials. **Units:** electron volts **Default:** the current value for this material
`DVX.0`	number	The pre-exponential constant for diffusion of phosphorus with neutral vacancies. **Units:** microns^2/min or cm^2/sec **Default:** the current value for this material
`DVX.E`	number	The activation energy for diffusion of phosphorus with neutral vacancies. **Units:** electron volts **Default:** the current value for this material
`DVM.0`	number	The pre-exponential constant for diffusion of phosphorus with singly-negative vacancies. **Units:** microns^2/min or cm^2/sec **Default:** the current value for this material
`DVM.E`	number	The activation energy for diffusion of phosphorus with singly-negative vacancies. **Units:** electron volts **Default:** the current value for this material

Parameter	Type	Definition
DVMM.0	number	The pre-exponential constant for diffusion of phosphorus with doubly-negative vacancies. **Units:** microns^2/min or cm^2/sec **Default:** the current value for this material
DVMM.E	number	The activation energy for diffusion of phosphorus with doubly-negative vacancies. **Units:** electron volts **Default:** the current value for this material
DIPAIR.0	number	The pre-exponential constant for the diffusivity of phosphorus-interstitial pairs. **Units:** microns^2/min or cm^2/sec **Default:** the current value for this material
DIPAIR.E	number	The activation energy for the diffusivity of phosphorus-interstitial pairs. **Units:** electron volts **Default:** the current value for this material
DVPAIR.0	number	The pre-exponential constant for the diffusivity of phosphorus-vacancy pairs. **Units:** microns^2/min or cm^2/sec **Default:** the current value for this material
DVPAIR.E	number	The activation energy for the diffusivity of phosphorus-vacancy pairs. **Units:** electron volts **Default:** the current value for this material
R.I.S	number	The capture radius for the reaction between interstitials and substitutional phosphorus atoms. **Units:** Å **Default:** the current value for this material
E.I.S	number	The barrier energy for the reaction between interstitials and substitutional phosphorus atoms. **Units:** electron volts **Default:** the current value for this material
R.V.S	number	The capture radius for the reaction between vacancies and substitutional phosphorus atoms. **Units:** Å **Default:** the current value for this material
E.V.S	number	The barrier energy for the reaction between vacancies and substitutional phosphorus atoms. **Units:** electron volts **Default:** the current value for this material
R.IP.V	number	The capture radius for the reaction between phosphorus-interstitial pairs and vacancies. **Units:** Å **Default:** the current value for this material

Parameter	Type	Definition
E.IP.V	number	The barrier energy for the reaction between phosphorus-interstitial pairs and vacancies. **Units:** electron volts **Default:** the current value for this material
R.VP.I	number	The capture radius for the reaction between phosphorus-vacancy pairs and interstitials. **Units:** Å **Default:** the current value for this material
E.VP.I	number	The barrier energy for the reaction between phosphorus-vacancy pairs and interstitials. **Units:** electron volts **Default:** the current value for this material
IP.V.LOW	logical	Assumes that only dopant-interstitial pairs and vacancies having opposite charges of equal magnitude react in bulk material. **Default:** true
IP.V.MED	logical	Assume that uncharged dopant-interstitial pairs (or vacancies) can react with vacancies (or dopant-interstitial pairs) in any charged state, and that oppositely charged dopant-interstitial pairs and vacancies can react. **Default:** false
IP.V.HIG	logical	Assumes that dopant-vacancy pairs in any charged state can react with interstitials in any charged state. **Default:** false
VP.I.LOW	logical	Assumes that only dopant-vacancy pairs and interstitials having opposite charges of equal magnitude react in bulk material. **Default:** true
VP.I.MED	logical	Assumes that uncharged dopant-vacancy pairs (or interstitials) can react with interstitials (or dopant-vacancy pairs) in any charged state, and that oppositely charged dopant-vacancy pairs and interstitials can react. **Default:** false
VP.I.HIG	logical	Assumes that dopant-vacancy pairs in any charged state can react with interstitials in any charged state. **Default:** false
SS.CLEAR	logical	Clears the solid solubility vs. temperature table. **Default:** false
SS.TEMP	number	The temperature at which the solid solubility in material 1 is **SS.CONC**. **Units:** degrees Celsius **Default:** none
SS.CONC	number	The solid solubility in material 1 at temperature **SS.TEMP**. **Units:** atoms/cm^3 **Default:** none
/MATERIA	character	The name of material 2 for the segregation parameters. **Default:** none

Parameter	Type	Definition
`/SILICON`	logical	Specifies that segregation parameters given on this statement apply to silicon as material 2. **Default:** false
`/OXIDE`	logical	Specifies that segregation parameters given on this statement apply to oxide as material 2. **Default:** false
`/OXYNITR`	logical	Specifies that segregation parameters given on this statement apply to oxynitride as material 2. **Default:** false
`/NITRIDE`	logical	Specifies that segregation parameters given on this statement apply to nitride as material 2. **Default:** false
`/POLYSIL`	logical	Specifies that segregation parameters given on this statement apply to polysilicon as material 2. **Default:** false
`/AMBIENT`	logical	Specifies that segregation parameters given on this statement apply to the ambient gas as material 2. **Default:** false **Synonym:** **`/GAS`**
`SEG.0`	number	The pre-exponential factor for segregation from material 1 to material 2. **Units:** none **Default:** the current value for these materials
`SEG.E`	number	The activation energy for segregation from material 1 to material 2. **Units:** electron volts **Default:** the current value for these materials
`TRANS.0`	number	The pre-exponential factor for transport from material 1 to material 2. **Units:** microns/min or cm/sec **Default:** the current value for these materials **Synonym:** **`TRN.0`**
`TRANS.E`	number	The activation energy for transport from material 1 to material 2. **Units:** electron volts **Default:** the current value for these materials **Synonym:** **`TRN.E`**
`KSIP.0`	number	The pre-exponential constant for the surface recombination velocity of the dopant-interstitial pair at the interface between the specified materials. **Units:** microns/min or cm/sec **Default:** the current value for these materials
`KSIP.E`	number	The activation energy for the surface recombination velocity of the dopant-interstitial pair at the interface between the specified materials **Units:** electron volts **Default:** the current value for these materials

Parameter	Type	Definition
KSVP.0	number	The pre-exponential constant for the surface recombination velocity of the dopant-vacancy pair at the interface between the specified materials. **Units:** microns/min or cm/sec **Default:** the current value for these materials
KSVP.E	number	The activation energy for the surface recombination velocity of the dopant-vacancy pair at the interface between the specified materials **Units:** electron volts **Default:** the current value for these materials
ES.RAND	number	The electronic stopping power coefficient of implanted phosphorus in the specified material for materials other than silicon and for a nonchanneled direction in silicon. This value is used for the MC ion implant calculation only. **Units:** angstrom^2*eV$^{(1-\mathtt{ES.F.RAN})}$ **Default:** the current value for phosphorus and the specified material
ES.F.RAN	number	The exponent of the electronic stopping power of implanted phosphorus in the specified material for materials other than silicon and for a nonchanneled direction in silicon. This value is used for the MC ion implant calculation only. **Units:** none **Default:** the current value for phosphorus and the specified material
ES.100	number	The electronic stopping power of phosphorus in silicon along the <100> channeling axes. This value is used for the MC ion implant calculation only. **Units:** angstrom^2*eV$^{(1-\mathtt{ES.F.100})}$ **Default:** the current value for phosphorus and the specified material
ES.F.100	number	The exponent of the electronic stopping power of phosphorus in silicon along the <100> crystal axes. This value is used for the MC ion implant calculation only. **Units:** none **Default:** the current value for phosphorus and the specified material
ES.110	number	The electronic stopping power of phosphorus in silicon along the <110> channeling axes. This value is used for the MC ion implant calculation only. **Units:** angstrom^2*eV$^{(1-\mathtt{ES.F.100})}$ **Default:** the current value for phosphorus and the specified material
ES.F.110	number	The exponent of the electronic stopping power of phosphorus in silicon along the <110> crystal axes. This value is used for the MC ion implant calculation only. **Units:** none **Default:** the current value for phosphorus and the specified material
CM.SEC	logical	If true, parameters involving time are specified in centimeters and seconds; if false, parameters involving time are in microns and minutes. **Default:** false

Description

This statement specifies properties and model coefficients for phosphorus. The values of the diffusivity, reaction constant, solid solubility, and electronic stopping parameters apply in material 1 (specified without the “/”), while **SEG.0**, **SEG.E**, **TRANS.0**, and **TRANS.E** apply at the interface between material 1 and material 2 (specified with the “/”). These coefficients are normally given in the *s4init* file (which is read at the start of each TSUPREM-4 execution) but can be changed by you at any time. Coefficients that are *not* given in the *s4init* file or set by the user, default to 0.0, except for **`SEG.0`** that defaults to 1.0.

The newer **`IMPURITY`** statement can be used to set all of the properties of phosphorus, including some that cannot be set with the **`PHOSPHOR`** statement.

Parameters whose units include time are specified in units of microns and minutes, unless **`CM.SEC`** is true, in which case units of centimeters and seconds are assumed.

See Also

For additional information see the following sections:

- Chapter 3, Diffusion, p. 3-14 for diffusion and segregation parameters.
- Chapter 3, MC Ion Implant Model, p. 3-131 for electronic stopping power parameters.

Examples

1. The following statement specifies the diffusion of phosphorus diffusing with doubly-negative interstitials in silicon:

```
PHOSPHORUS SILICON DIMM.0=2.652e11 DIMM.E=4.37
```

2. The following statement specifies the segregation parameters at the Si/SiO$_2$ interface:

```
PHOS SILICON /OXIDE SEG.0=30.0 TRANS.0=0.1
```

The concentration in silicon is 30.0 times the concentration in oxide, at equilibrium.

Additional PHOSPHORUS Notes

1. The **`PHOSPHORUS`** statement has been made obsolete by the **`IMPURITY`** statement, but remains available for compatibility with existing input files. *Note* that some properties of phosphorus can only be set on the **`IMPURITY`** statement.
2. The fractional interstitial parameter **`FI`** that was used in TSUPREM-4 before Version 6.0 is no longer supported. Instead, it is now necessary to specify the diffusivities with interstitials and vacancies separately.

Tutorial Examples

This chapter presents three short examples showing how to use TSUPREM-4 to do some simple simulations. New users may wish to study these examples in sequence. Each example introduces new TSUPREM-4 commands and concepts. The examples in this chapter are fairly short and execute relatively quickly. The simulations presented in Chapter 6, Advanced Examples are more typical of real-life applications, and take more execution time. The examples include:

- 1-D Bipolar Example: a 1-D simulation of a bipolar transistor structure. This example illustrates the basic simulation steps using a simple 1-D grid.
- Local Oxidation: an example that shows how to set up a 2-D grid and how to activate the various oxidation and point defect models. (See)
- Point Defect Models: an example illustrating the effect of the point defect models on impurity diffusion.

This chapter uses lowercase file names with the *.inp* extension for the example input files. Note the following regarding the format of input files:

- Each statement occupies one line, unless the statement is explicitly continued with a “+” character at the end of a line.
- Statement and parameter names can be abbreviated.
- On most statements, parameters can appear in any order.
- Blank lines between statements and extra spaces between parameters are ignored.
- Comment statements begin with the “$” character and are used to document the input file.
- The input can contain a mixture of upper and lower case; the case is ignored, except in some character strings.

1-D Bipolar Example

This section presents a 1-D simulation of a bipolar transistor structure. The example simulates the active region of an oxide-isolated bipolar structure, from the buried collector region up through the emitter. The statements that direct the simulation are contained in the simulation input files *s4ex1a.inp* and *s4ex1b.inp*.

This example illustrates the mechanics of using TSUPREM-4 and the use of particular statements for mesh setup, model specification, simulation of processing steps, and printing and plotting of results. A 1-D example is presented primarily to simplify the discussion, but it also serves to illustrate a useful technique for performing fast simulations of simple structures.

TSUPREM-4 Input File Sequence

This example illustrates the organization of a typical TSUPREM-4 input file. In general, the sequence is as follows:

1. Identify the problem with comments, and set any necessary execution options (none are needed in most cases).
2. Generate an initial mesh, or read in a previously saved structure.
3. Simulate the desired process sequence, and print and/or plot the results.

Note that there is considerable flexibility in this sequence; for example, alternating between simulation and plotting. The only strict requirement is that a mesh must be defined before any processing or output can be performed.

Initial Active Region Simulation

The input statements in the file *s4ex1a.inp* simulate the initial steps in the formation of the active region of a bipolar structure, including the formation of the buried collector and deposition of the epitaxial layer. These input statements are shown in Figure 5-1.

```
$ TSUPREM-4  --  Example 1, Part A
$ Bipolar active device region:  Buried layer and epitaxial deposition

$ Use automatic grid generation and adaptive grid
INITIALIZE BORON=1E15

$ Grow buried layer masking oxide
DIFFUSION  TEMP=1150  TIME=120  STEAM

$ Etch the buried layer masking oxide
ETCH       OXIDE  ALL

$ Implant and drive in the antimony buried layer
IMPLANT    ANTIMONY   DOSE=1E15  ENERGY=75
DIFFUSION  TEMP=1150  TIME=30    DRYO2
DIFFUSION  TEMP=1150  TIME=360

$ Etch the oxide.
ETCH       OXIDE  ALL

$ Grow 1.8 micron of arsenic-doped epitaxy
EPITAXY    THICKNES=1.8  SPACES=9  TEMP=1050  TIME=6  ARSENIC=5E15

$ Grow pad oxide and deposit nitride
DIFFUSION  TEMP=1050  TIME=30  DRYO2
DEPOSITION NITRIDE  THICKNES=0.12

$ Save initial active region results
SAVEFILE   OUT.FILE=S4EX1AS

$ Plot results
SELECT     Z=LOG10(BORON) TITLE=”Active, Epitaxy” LABEL=LOG(CONCENTRATION)
PLOT.1D    BOTTOM=13  TOP=21  RIGHT=5  LINE.TYP=5  COLOR=2
SELECT     Z=LOG10(ARSENIC)
PLOT.1D    ^AXES  ^CLEAR  LINE.TYP=2  COLOR=3
SELECT     Z=LOG10(ANTIMONY)
PLOT.1D    ^AX    ^CL     LINE.TYP=3  COLOR=3

$ Label plot
LABEL      X=4.2  Y=15.1  LABEL=Boron
LABEL      X=-.8  Y=15.8  LABEL=Arsenic
LABEL      X=2.1  Y=18.2  LABEL=Antimony

$ Print layer information
```

Figure 5-1 Input file *s4ex1a.inp*, for simulating the buried layer and epitaxial deposition for a bipolar transistor structure

Mesh Generation

Traditionally, generation of the simulation mesh has been one of the most difficult and time-consuming tasks required for process simulation. If the mesh is too coarse, accuracy of the simulation is poor, while making the mesh too fine wastes time and computational resources. TSUPREM-4 simplifies the problem of creating an appropriate mesh by providing automatic mesh generation and adaptive gridding.Automatic Mesh Generation

A mesh is generated automatically when the **`INITIALIZE`** statement is processed. A boron concentration of $10^{15}/cm^3$ for the initial structure has been specified. By default, the mesh that has two vertical lines, one at x=0 and one at x=1 micron. (For a 2-D simulation it would also be necessary to specify the width of the initial structure.) The location of the horizontal grid lines is determined by defaults set in the *s4init* file. The mesh can be made finer or coarser by including a **`MESH`** statement with the **`GRID.FAC`** parameter; the default spacing has been chosen for this example.

Adaptive Gridding

Adaptive gridding is done whenever an ion implantation or diffusion step is simulated to ensure that the grid is fine enough to maintain the required accuracy. The accuracy criteria can be adjusted with the **`ERR.FAC`** parameter on the **`METHOD`** statement; in this example the default values are used. By using adaptive gridding, the need to predict the grid requirements of a simulation in advance is eliminated.

Model Selection

The choice of simulation models should be considered before performing any processing steps. The speed and accuracy of the simulation depends strongly on the choice of models. The default models have been carefully chosen to give the best results in many cases, but some choices depend on the structures being simulated and on individual requirements of the user.

Two model choices must be made in most simulations:

- Oxidation model
- Point defect model

These selections are made using the **`METHOD`** statement. In this example the default models are used, i.e., the **`VERTICAL`** oxidation model and the **`PD.FERMI`** point defect model.

The choice of models may be changed during the course of the simulation.

Oxidation Model

The **`VERTICAL`** oxidation model can be used in this case because the simulation is one-dimensional, so only planar surfaces are oxidized. The **`ERFC`** oxidation model could also be used in this example, but it does not model the dependence of the oxidation rate on the concentration of impurities in the silicon. Also, the **`ERFC`** model requires an additional statement (an **AMBIENT** statement with the **INITIAL** parameter) to specify the initial oxide thickness whenever a structure with an initial oxide layer is oxidized, and does not automatically recognize the presence of nitride masking layers.

Point Defect Model

For this example the default point defect model (**PD.FERMI**) is used. The **PD.TRANS** point defect model increases the simulation time significantly, and should be used only when required. In this example, the effects of nonequilibrium point defect concentrations are relatively small. The magnitude of the error can be checked by repeating the simulation using the **PD.TRANS** or **PD.FULL** model.

Processing Steps

The following processing steps are used:

1. Buried layer masking oxide growth and patterning
2. Buried layer implant and drive-in
3. Epitaxial layer growth
4. Pad oxide growth and nitride mask deposition

Buried Layer Masking Oxide

The first step in the process is to grow an oxide to mask the buried layer implant. Oxidation is accomplished by specifying an oxidizing ambient (**STEAM**, in this case) on the **DIFFUSION** statement. The time (in minutes) and temperature (in ° Celsius) must be specified. *Note* that specifying an oxidizing ambient turns on oxidation, but does not disable any other aspects of a diffusion step. Impurity diffusion and segregation at interfaces still occur.

The oxide must be removed from the active region of the device before implanting the buried layer. This is accomplished with the **ETCH** statement. The simplest form of the **ETCH** statement, used here, removes all of a specified material. The **ETCH** statement can also be used to remove portions of a layer.

Buried Layer

The next step is to implant the buried layer and drive it in. This is done with the **IMPLANT** statement and the following **DIFFUSION** statements. The **IMPLANT** statement specifies the type of impurity, and the dose (per cm^2) and energy (in keV). Use of adaptive gridding is generally sufficient to ensure that the mesh at the surface is fine enough to contain several grid points within the peak of the implanted distribution, but it is wise to plot the as-implanted distribution if there is any doubt.

The drive-in of the buried layer is done in two steps. In the first, a **DRYO2** ambient is used to grow a thin layer of oxide to prevent outdiffusion of the implanted antimony. No ambient is given for the second; the absence of an ambient specification implies that an inert ambient is used. Once the drive-in is finished, the oxide is removed with an **ETCH** statement.

Epitaxial Layer

Next, an epitaxial layer is grown. Epitaxial growth is simulated with the **`EPITAXY`** statement, which combines the effects of the **`DEPOSITION`** and **`DIFFUSION`** statements. The thickness of the deposited layer is specified to be 1.8 microns using the **`THICKNES`** parameter.

The grid density in the layer is determined by the **`SPACES`** parameter. This example specifies that 9 grid spaces be placed in the epitaxial layer. This produces a reasonable starting grid for the base and emitter processing that follows. *Note* that the specification of the number of spaces determines not only the grid in the resulting structure, but also the time discretization of the epitaxy step. The **`EPITAXY`** statement causes the deposition of one grid layer followed by a diffusion for some fraction of the total time. This process is repeated for each grid layer. If **`SPACES`** is not specified, its value defaults to one and the **`EPITAXY`** is equivalent to a **`DEPOSITION`** followed by a **`DIFFUSION`** statement.

The time and temperature for the epitaxial growth step are given by the **`TIME`** and **`TEMPERAT`** parameters, respectively. These determine the amount of diffusion that occurs during the growth process. The doping of the epitaxial layer can be specified with the **`ARSNIC`** parameter on the **`EPITAXY`** statement.

Pad Oxide and Nitride Mask

Next, a pad oxide is grown (using the **`DIFFUSION`** statement with the **`DRYO2`** parameter) and a nitride mask is deposited to be used for the field-isolation oxidation. The deposition is specified by the **`DEPOSITION`** statement, which specifies the material to deposit and the thickness of the layer. Other optional parameters can be used to specify the number and spacing of grid spaces in the deposited layer; by default, a single grid space is used in the deposited layer.

Saving the Structure

The resulting structure is saved using the **`SAVEFILE`** statement. In this case, it is saved in the output file named *S4EX1AS*. The saved structure is used as the starting point for the remaining process steps, which is simulated with a separate TSUPREM-4 input file.

It is recommended that the structure be saved after any long (in terms of computer time) sequence of operations. This allows the simulation to be resumed at that point should the need arise. It is also a good idea to save the structure at the end of any simulation, so that the results can be examined further at a later time.

Plotting the Results

The results of the simulation to this point are now ready to be displayed. This example demonstrates the use of the **`SELECT`**, **`PLOT.1D`**, and **`LABEL`** statements; 2-D plots are demonstrated in the next section.

Specifying a Graphics Device

Before displaying a plot, the program must be told what sort of graphics device is being used. This can be done either with the **`DEVICE`** parameter on the **`OPTION`** statement or by using a default plot device. In this example, the default plot device is used. See Input Statement Descriptions, Chapter 4, on page 4-1 and Appendix AB.

The SELECT Statement

The value to be plotted is given by the **`Z`** expression on the **`SELECT`** statement. **`Z`** defines a mathematical expression that may contain a number of variables, functions, and mathematical operators. (See the description of the **`SELECT`** statement in Chapter 4, SELECT, p. 4-146). This example plots the base-10 logarithm of the various impurity concentrations.

Note:

> ***The `Z` expression is evaluated when the `SELECT` statement is processed. Thus the `SELECT` statement should be specified after any process steps that affect the device structure or solution quantities (e.g., impurity concentrations) used in the expression.***

The **`SELECT`** statement can also be used to specify a title for the plot and/or the label to be used on the vertical axis. If no label is given, the **`Z`** expression is used.

The PLOT.1D Statement

The **`PLOT.1D`** statement plots the values of a quantity along a (one-dimensional) section through the device. The section can be a vertical line at the location specified by **`X.VALUE`**, a horizontal line at **`Y.VALUE`**, or along an interface between two materials. This default is to plot along the vertical line at x=0.

The first **`PLOT.1D`** statement in this example plots the axes and title as well as the logarithm of the antimony concentration. In addition to the x coordinate, the minimum and maximum values for the y axis and the maximum value for the x axis have also been specified.

Note:

> ***The limits of the y axis are in the units of the `Z` expression on the `SELECT` statement; thus, 13 and 21 are used, (not 1e13 and 1e21).***

If no axis limits are given, default values based on the dimensions of the device and the values of the **`Z`** expression are used.

By default, a dashed vertical line is drawn at the interfaces between materials. **`LINE.TYP=5`** specifies that dashed line type 5 is to be used for plotting the boron profile; **`COLOR=2`** specifies that color 2 (usually red) is to be used on color graphics devices.

The next two **`PLOT.1D`** statements add to the first plot. They do this by including the **`^CLEAR`** and **`^AXES`** specifications to prevent clearing of the screen and drawing of new axes, respectively. For each plot a new quantity to be plotted is specified (using a **`SELECT`** statement) along with a different line type and color.

Labels

Labels are added to the plot with **`LABEL`** statements. Each **`LABEL`** statement specifies a text string to be plotted and a pair of *x* and *y* coordinates. The text string is plotted starting at the specified coordinates. **`X`** and **`Y`** are in the units of the plot axes, in this case microns and log10 (concentration), respectively. **`X`** and **`Y`** could also be given in centimeters, using the **`CM`** parameter. The final plot is shown in Figure 5-2.

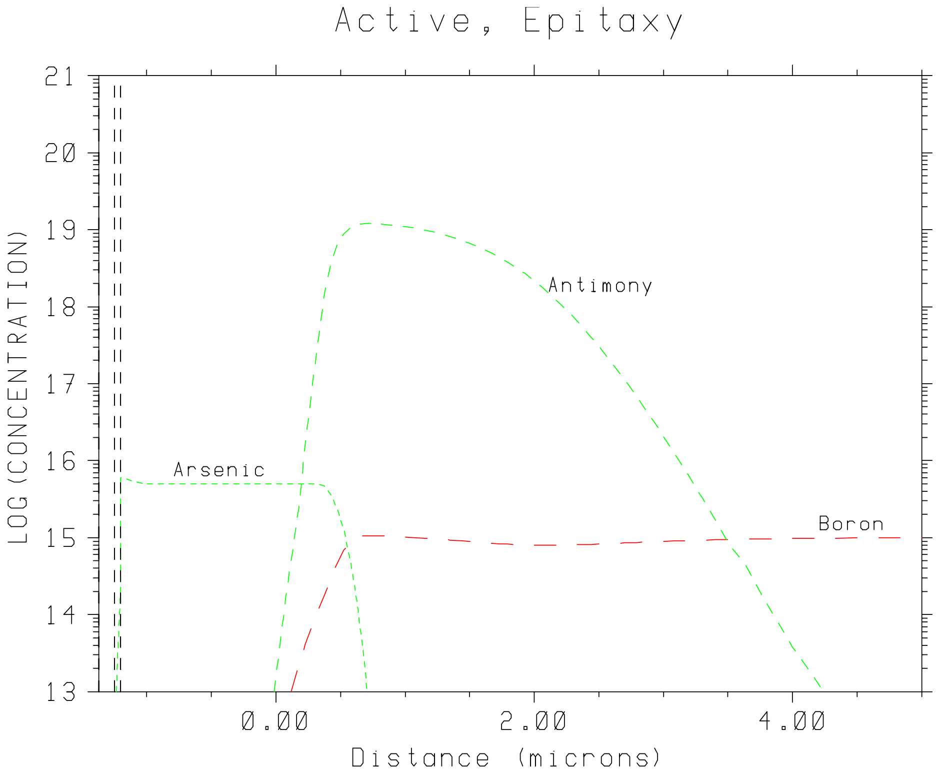


Figure 5-2 Impurity distributions in bipolar structure at end of input file *s4ex1a.inp*

Printing Layer Information

Although plots are the primary means of examining the results of a TSUPREM-4 simulation, useful information is also available from the **`PRINT.1D`** statement.

The PRINT.1D Statement

The **`PRINT.1D`** statement works with the **`Z`** expression given by a **`SELECT`** statement. Information is presented along a 1-D section specified in the same manner as in the **`PLOT.1D`** statement.

Three kinds of information can be printed:

1. A complete list of the values of the **`Z`** expression along the section.
2. The integral of the **`Z`** expression through each layer along the section.
3. The locations along the section where the **`Z`** expression has a specified value.

Using PRINT.1D LAYERS

The **PRINT.1D** statement at the end of file *s4ex1a.inp* uses the **LAYERS** parameter to request the integral of the **Z** expression (net doping) along a vertical section at *x*=0. The resulting output is shown in Figure 5-3.

```
$ Print layer information
SELECT     Z=DOPING
PRINT.1D   LAYERS
** Printing along X.VALUE=0:

 Num      Material       Top     Bottom  Thickness        Integral
   1       nitride   -1.3820    -1.2525     0.1200      1.2000e+00
   2         oxide   -1.2620    -1.2031     0.0491      5.5423e+08
   3       silicon   -1.2129     3.4935     4.7108      1.4215e+15
   4       silicon    3.4979   200.0000   196.5021     -1.9630e+13
```

Figure 5-3 Output listing from **PRINT.1D** command in file *s4ex1a.inp*

For purposes of **PRINT.1D LAYERS**, a layer is defined as a portion of the section in which the material does not change and the sign of the **Z** expression is constant. In the case of net doping, each material is separated into layers of net n-type or net p-type doping.

In this example, the **PRINT.1D LAYERS** statement provides a number of useful results. For each layer, the material type, the top and bottom coordinates of the layer, the thickness of the layer, and the integral of the selected quantity are printed. A number of useful values are evident, such as the oxide thickness (491 Å) and the depth of the buried layer/substrate junction (4.71 µm).

Note:

> **SELECT** ***statement is required before a*** **PRINT.1D LAYERS** ***statement is processed, but you can use*** **SELECT Z=0** ***if all you want to know is the layer thicknesses or the coordinates of the material interfaces.***

Completing the Active Region Simulation

The simulation is now completed with a separate execution of TSUPREM-4, using the input file *s4ex1b.inp* shown in Figure 5-4. This file follows the same outline as the first (initialize, select models, process, output), and most of the statements are of the types discussed previously. However, there are some differences as discussed in the following subsections.

.

```
$ TSUPREM-4  --  Example 1, Part B
$ Bipolar active device region:  Field oxide, base, and emitter

$ Read structure
INITIALIZE IN.FILE=S4EX1AS

$ Grow the field oxide
DIFFUSION  TEMP=800   TIME=20         T.FINAL=1000
DIFFUSION  TEMP=1000  TIME=10  DRYO2  T.FINAL=1100  P.FINAL=5
DIFFUSION  TEMP=1100  TIME=50  STEAM  PRESSURE=5
DIFFUSION  TEMP=1100  TIME=10  DRYO2  PRESSURE=5    P.FINAL=1
DIFFUSION  TEMP=1100  TIME=60         T.FINAL=800

$ Remove nitride and pad oxide
ETCH       NITRIDE  ALL
ETCH       OXIDE    ALL

$ Implant the boron base
IMPLANT    BORON  DOSE=2E13  ENERGY=100

$ Implant the phosphorus emitter
IMPLANT    PHOSPHORUS  DOSE=1E15  ENERGY=50

$ Anneal to activate base and emitter regions
DIFFUSION  TEMP=1000  TIME=12  DRYO2

$ Plot results
SELECT     Z=LOG10(BORON)  TITLE="Active Region"  LABEL=LOG(CONCENTRATION)
PLOT.1D    BOTTOM=13  TOP=21  RIGHT=5  LINE.TYP=5  COLOR=2
SELECT     Z=LOG10(PHOSPHORUS)
PLOT.1D    ^AXES  ^CLEAR  LINE.TYP=4  COLOR=4
SELECT     Z=LOG10(ARSENIC)
PLOT.1D    ^AXES  ^CLEAR  LINE.TYP=2  COLOR=3
SELECT     Z=LOG10(ANTIMONY)
PLOT.1D    ^AXES  ^CLEAR  LINE.TYP=3  COLOR=3

$ Label the impurities
LABEL      X= 2.0  Y=15.1  LABEL=Boron
LABEL      X=-1.0  Y=19.5  LABEL=Phosphorus
LABEL      X= 0.3  Y=15.8  LABEL=Arsenic
LABEL      X= 2.0  Y=18.4  LABEL=Antimony

$ Print the layer information
SELECT     Z=DOPING
PRINT.1D   X.V=0  LAYERS
```

Figure 5-4 Listing of input file *s4ex1b.inp*, showing statements for simulating the field oxide, base, and emitter region processing for a bipolar transistor

Reading a Saved Structure

The most important difference is that instead of generating a mesh, the saved structure saved by the previous simulation is read from the file *S4EX1AS*. This is accomplished with the **`IN.FILE`** parameter on the **`INITIALIZE`** statement. The saved file includes complete mesh and solution information.

The next step specifies any needed model specifications. Because the choices of oxidation and point defect models were saved in the structure file, they do not need to be specified again. Any different models could be specified at this step.

Field Oxidation

The next step in the process is to grow the field isolation oxide. Although the active region of the device is being simulated, and no oxide is grown because of the nitride mask, the field oxidation step is included in order to simulate the dopant redistribution that occurs during the step. The field oxidation step illustrates how temperature and pressure ramping are specified.

The temperature is first ramped from 800° C to 1000° C, over a time of 20 minutes. The **`TEMP`** parameter specifies the starting temperature, while the **`T.FINAL`** parameter gives the temperature at the end of the step. The next step starts at 1000° C in a dry oxygen ambient at one atmosphere, and ramps the temperature to 1100° C and the pressure to 5 atmospheres over a period of 10 minutes. The starting pressure is given by the **`PRESSURE`** parameter (not needed here because the default value of one atmosphere is used), and the final pressure is given by **`P.FINAL`**. *Note* that ramp rates could also have been specified in degrees/minute or atmospheres/minute, using the **`T.RATE`** and **`P.RATE`** parameters, respectively. The third diffusion step does a 50-minute oxidation in steam, at a pressure of 5 atmospheres. Steps 4 and 5 ramp down the pressure, then the temperature.The remaining process steps are similar to those performed in the first part. The nitride and oxide are removed, and the base and emitter are implanted.

Final Structure

The plotting and labeling of the profiles use the statements described previously. The final plot is shown in Figure 5-5.

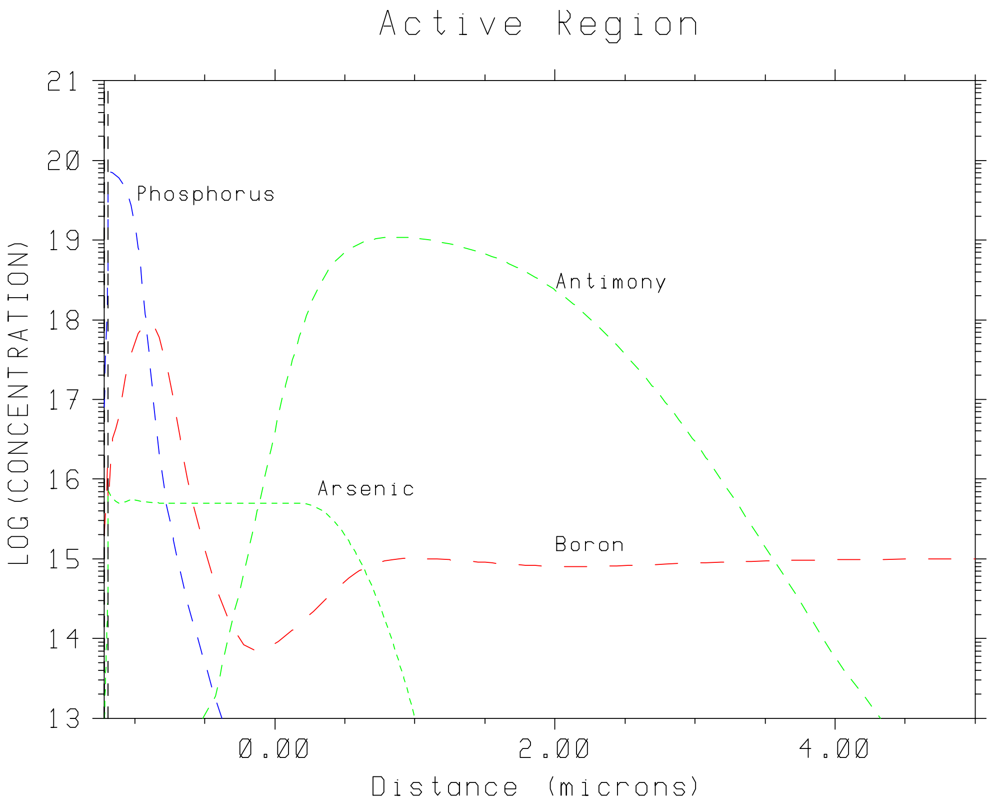


Figure 5-5 Final profiles produced by input file *s4ex1b.inp*

A **PRINT.1D** statement is used to print layer information for the final structure. The **SELECT Z=DOPING** statement specifies that integrals of net doping are to be calculated and that layers are defined by the sign of the doping.

From the printed output (Figure 5-6) a number of useful values can be extracted, such as the oxide thickness (255 Å), the thickness of the emitter region (0.28 µm), and the integrated doping in the base region (8.12x10^{12}/cm^3).

```
$ Print the layer information
SELECT     Z=DOPING
PRINT.1D   LAYERS
** Printing along X.VALUE=0:

 Num       Material       Top     Bottom  Thickness        Integral
   1          oxide   -1.2281    -1.2025     0.0255      3.7599e+12
   2        silicon   -1.2025    -0.9202     0.2823      1.0427e+15
   3        silicon   -0.9202    -0.5746     0.3456     -8.1165e+12
   4        silicon   -0.5746     3.5837     4.1584      1.4212e+15
   5        silicon    3.5837   200.0000   196.4136     -1.9623e+13
```

Figure 5-6 Output listing from **PRINT.1D** command in file *s4ex1b.inp*

Local Oxidation

This section presents an example of a 2-D simulation of a local oxidation process. The purpose of this example is to illustrate the 2-D simulation capabilities of TSUPREM-4, and to provide some practical hints on the effective use of these capabilities.

This example simulates a narrow, locally-oxidized isolation region. The two features of interest are the oxide thickness and shape and the impurity distribution in the underlying silicon. These are examined in separate simulations. The commands for performing the simulations are contained in the simulation input files *s4ex2a.inp* and *s4ex2b.inp*.

Calculation of Oxide Shape

The first goal of this simulation is to determine the effect of using a narrow mask opening on the oxide shape. The input statements for this simulation are shown in Figures 5-7 and 5-9. The steps are similar to those used in the previous example, but are complicated slightly by the 2-D nature of the simulation.

```
$ TSUPREM-4 narrow window example
$   Part 1:  Oxide shape

$ Set up the grid
LINE  X    LOC=0.0   SPAC=0.15
LINE  X    LOC=1.25  SPAC=0.05
LINE  X    LOC=1.5   SPAC=0.1

LINE  Y    LOC=0     SPAC=0.03
LINE  Y    LOC=0.5   SPAC=0.1

$ No impurities, for faster oxidation simulation
INITIALIZE

$ Deposit pad oxide and define nitride mask
DEPOSITION OXIDE    THICKNES=0.03  SPACES=2
DEPOSITION NITRIDE  THICKNES=0.10  SPACES=2
ETCH       NITRIDE  RIGHT  P1.X=1.25

$ Plot the grid
SELECT     TITLE="Grid for Oxidation"
PLOT.2D    GRID  SCALE  C.GRID=2

$ Do the oxidation
METHOD     VISCOEL DY.OXIDE=0.05  INIT=0.15
AMBIENT   MAT=OXIDE    VC=170
AMBIENT   MAT=NITRIDE  VC=100
DIFFUSION  TEMP=1000  TIME=100  WETO2

$ Save the structure
SAVEFILE   OUT.FILE=S4EX2AS
```

Figure 5-7 First part of input file *s4ex2a.inp*, for determining LOCOS shape

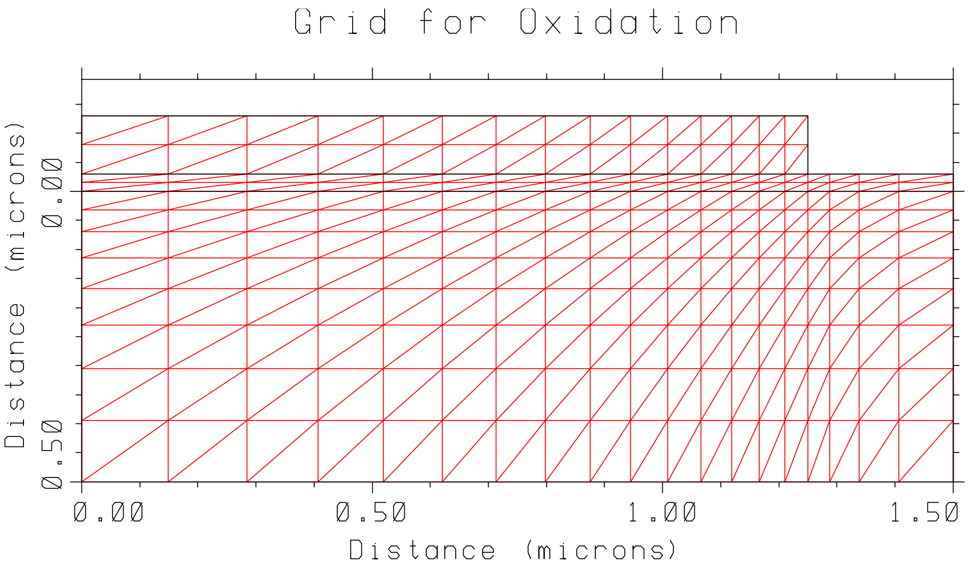


Figure 5-8 Mesh used for oxidation simulation. Produced by **PLOT.2D GRID** statement in input file *s4ex2a.inp*

Mesh Generation

The simulation starts by defining a mesh. The object in this case is to define as coarse a mesh as possible without losing accuracy in the solution. (The accuracy can be checked later by repeating the simulation with a finer mesh.) Only half of the structure needs to be simulated, because of the symmetry of the structure. The **LINE X** statements specify the horizontal locations of vertical grid lines. The first and last **LINE X** statements specify the left and right edges of the structure; the **SPACING** parameters specify the horizontal grid spacings at these locations. Other **LINE X** statements are used to add grid lines and specify spacings at other locations in the structure. In the vertical direction, **LINE Y** statements specify the top and bottom locations (and corresponding grid spacings) for the mesh.

The grid for this example does not need to be very fine. In the horizontal direction the spacing is set to 0.05 microns at the edge of the nitride mask, where the 2-D effects are greatest (x=1.25); the spacing expands to 0.1 micron at the right edge and 0.15 micron at the left edge. In the vertical direction, the spacing is set to 0.03 at the top, expanding to 0.1 at the bottom.

The effect of the grid spacing on accuracy can be checked by running the simulation again with a finer mesh; this can be accomplished by adding a **MESH** statement with the **GRID.FAC** parameter anywhere before the **INITIALIZE** statement. There are no impurities specified in this simulation, so no adaptive gridding occurs based on implantation or diffusion of impurities. However, the grid is refined as needed based on oxidant concentration.

The **INITIALIZE** statement performs the actual mesh generation. *Note* that no substrate doping has been specified. This speeds up the simulation slightly by eliminating the need to solve for impurity diffusion while oxidizing.

Pad Oxide and Nitride Layers

Next the pad oxide is deposited. To improve the accuracy when calculating oxidant diffusion in the oxide, two grid spaces are placed in the deposited layer. (When checking the accuracy of the simulation, more divisions would be useful.) More grid spaces are added automatically as the oxide grows.

To mask the oxidation, a nitride layer is deposited and then patterned with an **`ETCH`** statement. The **`RIGHT`** keyword on the **`ETCH`** statement specifies that material is to be etched to the right of the line defined by the **`P1.X`**, **`P1.Y`**, **`P2.X`**, and **`P2.Y`** parameters. **`P1.X=1.25`** is specified; **`P1.Y`** defaults to a value above the highest point of the structure; **`P2.X`** defaults to the value of **`P1.X`**; and **`P2.Y`** defaults to a value below the bottom of the structure. Thus the etch occurs to the right of a vertical line through the structure at x=1.25 microns.

Plotting the Mesh

At this point it is wise to inspect the mesh. This is done with the **`PLOT.2D`** statement. The **`PLOT.2D`** statement is used to set up any 2-D plot. By itself, it can plot the material boundaries of a structure, the grid, and oxide stress and velocity vectors. It is also used to set the scale and draw axes for contour plots. In this case, the **`GRID`** keyword requests that the grid be plotted, and the **`C.GRID`** keyword specifies the color to be used.

The **`SCALE`** parameter causes the x or y axis to be scaled to reflect the true aspect ratio of the structure. If **`SCALE`** had not been specified, the plot would be stretched in the y direction to fill the plotting area. The **`SELECT`** statement is used only to specify a title for the plot. The resulting plot is shown in Figure 5-8.

Model Selection

Next a **`METHOD`** statement is used with the **`VISCOELA`** keyword to select the oxidation model. The **`VISCOELA`** model is used because it allows fast simulation of two-dimensional, stress-dependent oxidation. The **`DY.OXIDE`** parameter sets the grid spacing to be used in the growing oxide. (The same grid is used both for the diffusion of the oxidizing species and for the calculation of oxide movement.) The **`INIT.TIM`** parameter specifies an initial time step of 0.15 minutes; this saves a small amount of calculation time. The default value is 0.002 minutes, which is appropriate for many diffusion steps but smaller than necessary for oxidation-only steps.

The effects of oxidant diffusion, interface reaction rate, and material viscosities on the stresses in the structure are included by default. Because the models for the stress effects do not include the dependence of parameter values on temperature, it is necessary to use **`AMBIENT`** statements to specify appropriate values for **`VC`** (the dependence of stress on material viscosity) at the oxidation temperature. (Suitable values are listed in the notes to Appendix AA.)

The oxidation is accomplished by the **`DIFFUSION`** statement, which specifies a wet oxygen ambient at 1000° C for 100 minutes. The **`SAVEFILE`** statement saves the final structure for later analysis.

```
$ Plot the final structure, showing flow lines
SELECT     TITLE="Flow at End of Oxidation Step"
PLOT.2D    SCALE  FLOW  VLENG=0.065  X.MIN=0.5  C.FLOW=4

$ Plot the final structure, showing stress vectors
SELECT     TITLE="Stresses After Oxidation"
PLOT.2D    SCALE  STRESS  VLENG=0.2  X.MIN=0.5 +
           C.COMPRE=4  C.TENSIO=2  L.TENSIO=2

$ Plot filled contours of hydrostatic pressure
SELECT    Z=( -0.5 * ( SXX + SYY ) )  TITLE="Contours of Hydrostatic Pressure"
PLOT.2D    SCALE  X.MIN=0.5  X.MAX=1.8
FOREACH I ( 1 TO 5 )
  COLOR MIN.V=(( I - 0.5)*2E9) MAX.V=(( I + 0.5)*2E9) COLOR=(13 + I )
  COLOR MIN.V=((- I - 0.5)*2E9) MAX.V=((- I + 0.5)*2E9) COLOR=(25 - I )
END

$ Create a legend
LABEL X=1.52 Y=-0.2  LABEL="Compression"       SIZE=0.3
LABEL X=1.60 Y=-0.15 LABEL="1-3E9" C.RECT=14 SIZE=0.3 W.RECT=0.35 H.R=0.35
LABEL X=1.60 Y=-0.1  LABEL="3-5E9" C.RECT=15 SIZE=0.3 W.RECT=0.35 H.R=0.35
LABEL X=1.60 Y=-0.05 LABEL="5-7E9" C.RECT=16 SIZE=0.3 W.RECT=0.35 H.R=0.35
LABEL X=1.60 Y= 0.0  LABEL="7-9E9" C.RECT=17 SIZE=0.3 W.RECT=0.35 H.R=0.35
LABEL X=1.52 Y= 0.05 LABEL="Tension"           SIZE=0.3
LABEL X=1.60 Y= 0.1  LABEL="1-3E9" C.RECT=24 SIZE=0.3 W.RECT=0.35 H.R=0.35
LABEL X=1.60 Y= 0.15 LABEL="3-5E9" C.RECT=23 SIZE=0.3 W.RECT=0.35 H.R=0.35
LABEL X=1.60 Y= 0.20 LABEL="5-7E9" C.RECT=22 SIZE=0.3 W.RECT=0.35 H.R=0.35
LABEL X=1.60 Y= 0.25 LABEL="7-9E9" C.RECT=21 SIZE=0.3 W.RECT=0.35 H.R=0.35

$ Redraw boundaries
PLOT.2D    ^AX  ^CL

$ Print location of interface
SELECT     Z=Y
PRINT.1D   SILICON  /OXIDE
```

Figure 5-9 Second part of statement input file *s4ex2a.inp*, showing statements for plotting results of LOCOS process

Plotting the Results

The results are plotted using the statements shown in Figure 5-9. The **PLOT.2D** statement is used to plot the structure boundaries and material interfaces. The following parameters specify the plot format and content:

- **FLOW** adds vectors showing the speed and direction of material flow in the structure due to oxidation.
- **VLENG** specifies the maximum length of a flow vector (in microns) to be plotted.
- The **X.MIN** parameter specifies the minimum value of the *x* axis: (**X.MAX**, **Y.MIN**, and **Y.MAX** can be used to specify the other axis limits, but the default values are used in this case).
- The **SCALE** parameter is used to avoid distorting the structure as it is scaled for plotting.

The result of the **PLOT.2D FLOW** statement is shown in Figure 5-10.

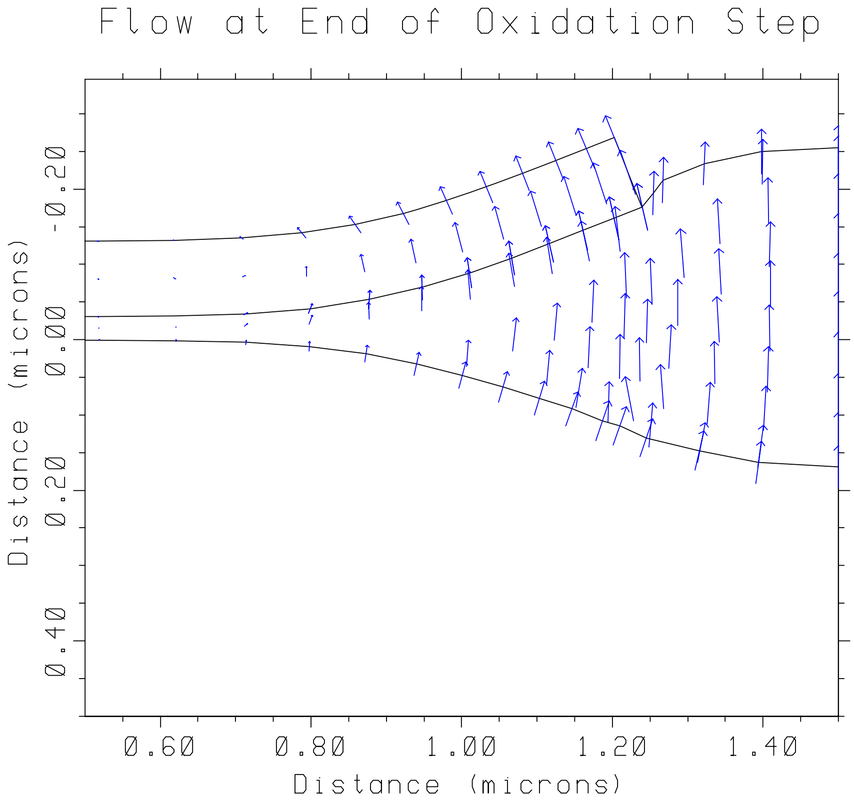


Figure 5-10 Plot produced by the **PLOT.2D FLOW** statement in input file *s4ex2a.inp*

Plotting Stresses

Stresses in the structure can be plotted in several ways. One way is to use the **PLOT.2D** statement with the **STRESS** parameter. The **STRESS** parameter requests that the principal components of stress be shown. The stresses are indicated by short lines whose lengths are proportional to the stress components. The line type indicates whether the material is in tension or compression. By default, both use line type one, but in this example **LINE.TEN=2** is specified so that a dashed line is used to indicate tension (the default is one). Tension and compression are also distinguished by color, using the **C.TENSIO** and **C.COMPRE** parameters. The **VLENG** parameter specifies the length of the line (in the units of the plot axes, i.e., microns) used for the maximum value of stress. Smaller stresses produce proportionally smaller lines.

The stress plot is shown in Figure 5-11. The long, solid lines indicate large compressive stresses in the direction of the lines; dashed lines indicate tension. As expected, the stresses are concentrated near the portion of the structure where the oxide growth is nonuniform and where the nitride is forced to bend.

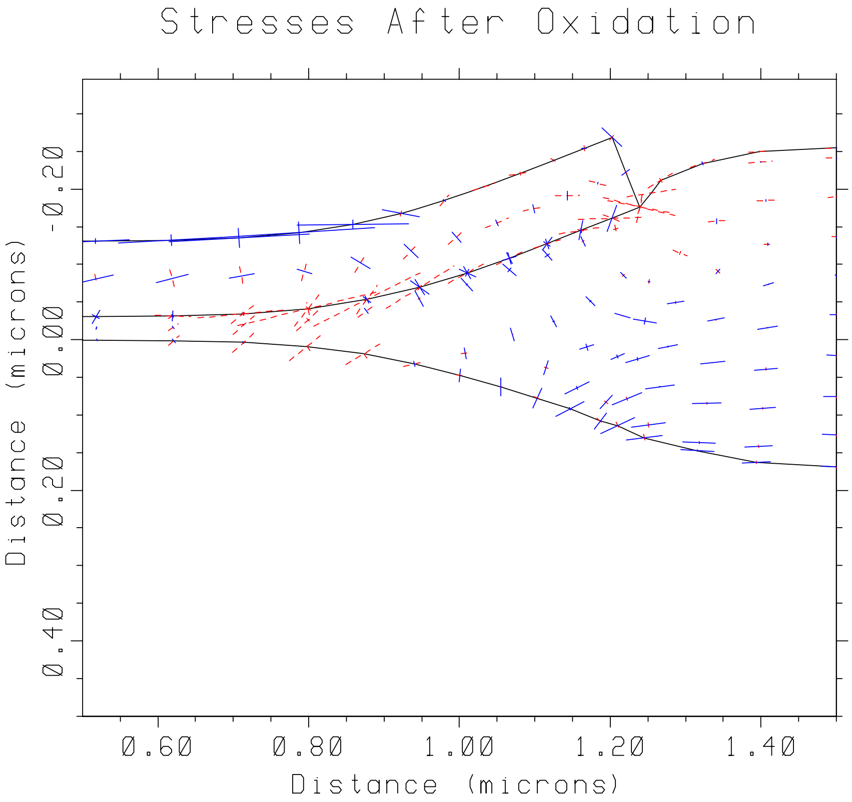


Figure 5-11 Plot produced by the **PLOT.2D STRESS** statement in input file *s4ex2a.inp*

Contour Plots

Another way to show the stresses in the structure is with contour plots. It is difficult to show all three components of stress in a contour plot, and it is not very useful to show individual components because of their dependence on the choice of coordinate system. However, meaningful combinations of the stresses can be calculated with the **SELECT** statement and plotted with the **PLOT.2D** and **CONTOUR** or **COLOR** statements.

This example shows how the hydrostatic pressure, defined as the negative average of the S_{xx} and S_{yy} stress components, can be plotted.

- The **SELECT** statement simply gives the mathematical expression for the function to be plotted.
- The **PLOT.2D** statement plots the axes, structure boundaries, and material interfaces, **X.MAX** is specified greater than the right edge of the structure in order to leave room for a legend.
- A pair of **COLOR** statements inside a **FOREACH** loop are used to shade regions of differing pressures.

  The **FOREACH** statement specifies a variable (*I*) and a range of values (1 to 5, in this case). The statements between the **FOREACH** statement and the matching **END** statement are executed once for each value of the variable.

The **COLOR** statements specify a color and the range of pressures to be represented by that color. Both the color and the minimum and maximum values are given by arithmetic expressions that depend on the variable *I*.

- A series of **LABEL** statements are used to produce a legend, showing the amount of tension or compression corresponding to each color.

  Each **LABEL** statement specifies the location of the label (in microns, in this case), the text of the label, and a filled rectangle that is placed before the label. The size of the characters (in centimeters) is specified by the **SIZE** parameter. Width and height of the rectangle are given by **W.RECTAN** and **H.RECTAN** (abbreviated to **W.R** and **H.R**, respectively).
- Finally, a last **PLOT.2D** with the **^AXES** and **^CLEAR** parameters is done in order to redraw the structure boundaries, which may have been obscured when plotting the shaded contours. The result is shown in Figure 5-12.

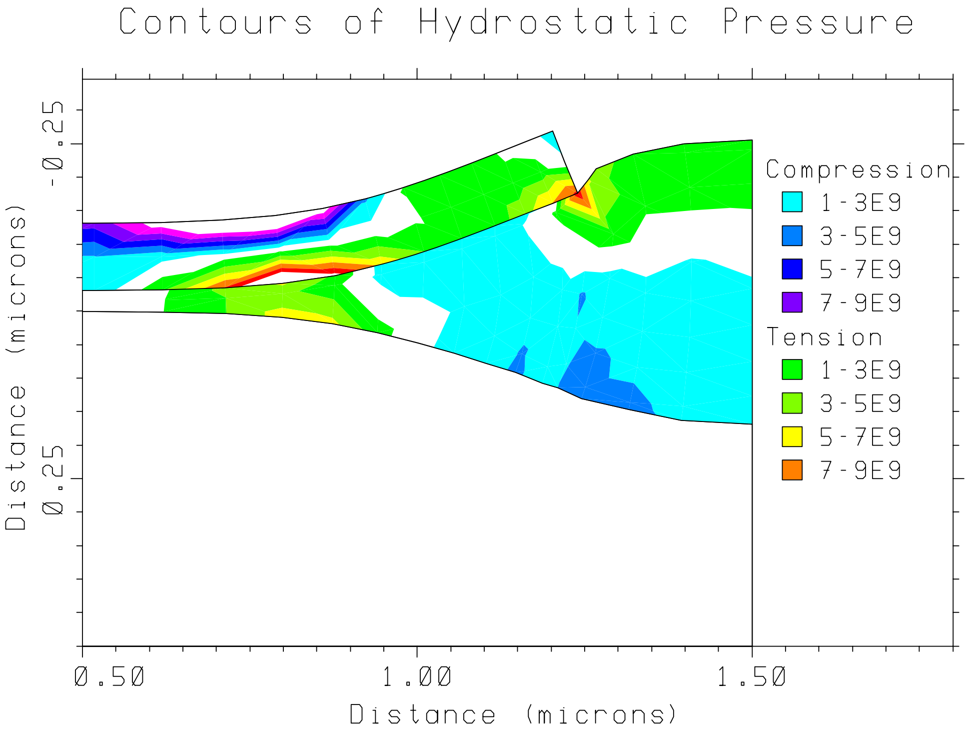


Figure 5-12 Contours of hydrostatic pressure plotted by statements in input file *s4ex2a.inp*

The last two lines of *s4ex2a.inp* show how the y coordinate of the silicon/oxide interface can be printed as a function of x, using the **PRINT.1D** statements to print solution values along the interface between two materials.

2-D Diffusion with Point Defects

This example analyzes the diffusion of impurities during the local oxidation step. The same process is simulated as in the previous sections, except with boron in the substrate and a boron implant. The effects of point defect generation during oxidation are also included, but with a simpler oxidation model to reduce the computer time requirements. The listing of the input (from file *s4ex2b.inp*) is shown in Figures 5-13 and 5-15.

Automatic Grid Generation

For convenience, automatic grid generation is used for this analysis. By specifying **`WIDTH`**=1.5 on the **`INITIALIZE`** statement, the grid extends from *x*=0 to *x*=1.5, just as in the previous simulation. The background concentration of the substrate is specified with the **`BORON`** parameter. The grid is automatically refined as needed to maintain the accuracy of the impurity profiles.

Field Implant

The deposition of the pad oxide and nitride are the same as before. Now, however, a photoresist layer is deposited and patterned to be used as a boron implant mask. The boron is implanted through the pad oxide, then the photoresist is removed.

Oxidation

The **`COMPRESS`** oxidation model is chosen in order to obtain a fast 2-D simulation. The 2-D point defect model is specified by the **`PD.TRANS`** keyword on the **`METHOD`** statement. The **`DIFFUSION`** statement is the same as in the previous section.

The structure is saved at the end of the simulation so the results can be examined further without rerunning the simulation. In this example, the solution is saved in the file *S4EX2BS*.

```
$ TSUPREM-4 narrow window example
$   Part 2:  Impurity distribution

$ Use default grid
INITIALIZE BORON=1E15 WIDTH=1.5

$ Deposit pad oxide, LOCOS mask, and implant mask
DEPOSITION OXIDE    THICKNES=0.03  SPACES=2
DEPOSITION NITRIDE  THICKNES=0.10
DEPOSITION PHOTO    THICKNES=2

ETCH       PHOTO    RIGHT  P1.X=1.25
ETCH       NITRIDE  TRAP

$ Implant boron
IMPLANT    BORON  DOSE=2E13  ENERGY=100
ETCH       PHOTORESIST  ALL

$ Do the drive-in, with point defects
METHOD     COMPRESS  PD.TRANS
DIFFUSION  TEMP=1000 TIME=100  WETO2

$ Save the structure
SAVEFILE   OUT.FILE=S4EX2BS

$ Plot the grid after diffusion
SELECT     TITLE=”Grid After Impurity Diffusion”
PLOT.2D    SCALE  GRID  Y.MAX=1.2  C.GRID=2
```

Figure 5-13 First part of input file *s4ex2b.inp*, showing processing steps

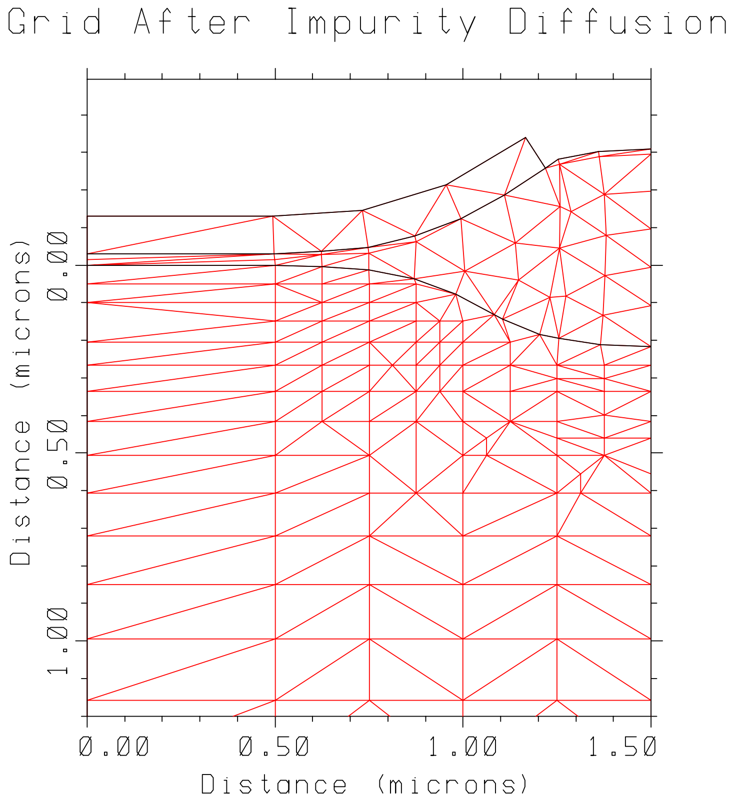


Figure 5-14 Grid plot produced by first **`PLOT.2D`** statement in input file *s4ex2b.inp*

Grid Plot

The grid at this point is plotted by the **`PLOT.2D GRID`** statement. The plot is shown in Figure 5-14. The grid has been refined near the surface and around the field isolation oxide to improve the resolution of the boron profile.

The **`Y.MAX`** parameter is used to restrict the plot to the upper part of the structure. If **`Y.MAX`** had been omitted, the entire grid would have been plotted down to the bottom of the structure (at *y*=200 microns) in its 1.5:200 aspect ratio, producing a nearly useless result. If the **`SCALE`** parameters had been omitted, the plot would be compressed in the *y* direction to make it fit the available plotting area, but the shape of the structure would be distorted.

The **`SELECT`** statement has been used to specify a title for the plot.

Contour of Boron Concentration

Figure 5-15 shows the input statements used to plot the results. The first plot shows contours of equal boron concentration. A **`PLOT.2D`** statement sets up the axes for the plot and draws the boundaries of the structure and the material interfaces. **`Y.MAX`**=1.2 (microns) is specified so that only the portion of the device near the surface is plotted. The **`SELECT`** statement specifies that the (base 10) logarithm of the boron concentration is to be plotted.

```
$ Plot contours of boron
SELECT     Z=LOG10(BORON)  TITLE=”Contours of Boron Concentration”
PLOT.2D    SCALE  Y.MAX=1.2
FOREACH    X (15.5 16.5 17)
  CONTOUR    VAL=X  COLOR=2
END
CONTOUR    VAL=16  LINE.TYP=2  COLOR=2

$ Plot interstitials and vacancies vs. depth
SELECT     Z=INTER/CI.STAR  TITLE=”Point Defects vs. Y  (X=1.5)” +
                            LABEL=”Normalized Defect Concentration”
PLOT.1D    X.VALUE=1.5  BOTTOM=-1  RIGHT=80.0  COLOR=4
SELECT     Z = VACAN/CV.STAR
PLOT.1D    X.VALUE=1.5  ^AXES  ^CLEAR  LINE.TYP=2  COLOR=6

$ Label the plot
LABEL      X=5  Y=4.0  LABEL=Interstitials
LABEL      X=4  Y=0.7  LABEL=Vacancies

$ Plot interstitials and vacancies vs. width
SELECT     Z=INTER/CI.STAR  TITLE=”Point Defects vs. X  (Y=2um)” +
                            LABEL=”Normalized Defect Concentration”
PLOT.1D    Y.VAL=2  BOTTOM=-1  COLOR=4
SELECT     Z=VACAN/CV.STAR
PLOT.1D    Y.VAL=2  ^AX  ^CL  LINE.TYP=2  COLOR=6
LABEL      X=0.2  Y=4.5  LABEL=Interstitials
LABEL      X=0.2  Y=0.5  LABEL=Vacancies

$ Prepare to plot contours of point defect concentrations
SELECT     Z=INTER/CI.STAR  TITLE=”Interstitial Contours”
PLOT.2D    SCALE  Y.MAX=1.5  X.MAX=2.1

$ Plot contours of interstitials
FOREACH    I ( 0 to 7 )
  COLOR MIN.V=(( I -0.5)*1.0+1.5) MAX.V=(( I +0.5)*1.0+1.5) COLOR=(9 + I )
END

$ Create a legend
LABEL X=1.68 Y=0.15 LABEL=”I/I*:” SIZE=.3
LABEL X=1.8 Y=0.30 LABEL=”1.5” SIZE=.3 W.RECT=.35 H.R=.35 C.R=9
LABEL X=1.8 Y=0.45 LABEL=”2.5” SIZE=.3 W.RECT=.35 H.R=.35 C.R=10
LABEL X=1.8 Y=0.60 LABEL=”3.5” SIZE=.3 W.RECT=.35 H.R=.35 C.R=11
LABEL X=1.8 Y=0.75 LABEL=”4.5” SIZE=.3 W.RECT=.35 H.R=.35 C.R=12
LABEL X=1.8 Y=0.90 LABEL=”5.5” SIZE=.3 W.RECT=.35 H.R=.35 C.R=13
LABEL X=1.8 Y=1.05 LABEL=”6.5” SIZE=.3 W.RECT=.35 H.R=.35 C.R=14
LABEL X=1.8 Y=1.20 LABEL=”7.5” SIZE=.3 W.RECT=.35 H.R=.35 C.R=15
LABEL X=1.8 Y=1.35 LABEL=”8.5” SIZE=.3 W.RECT=.35 H.R=.35 C.R=16

$ Redraw the boundaries
PLOT.2D    ^AX  ^CL

$ Label the plot
LABEL      X=1.4  Y= 0.0  LABEL=Oxide  RIGHT

$ Print boron vs. depth in field region
SELECT     Z=BORON
PRINT.1D   X.VALUE=0  X.MAX=2
```

Figure 5-15 Second part of input file *s4ex2b.inp,* showing statements for plotting the results of the diffusion simulation

The **CONTOUR** statement plots a contour at a single value of the selected quantity, so a series of **CONTOUR** statements is needed for a typical contour plot. The specification of these **CONTOUR** statements is simplified by using the **FOREACH** and **END** statements to define a loop.

Using the FOREACH Statement

The **FOREACH** statement is special in that it does not use the usual "parameter=value" type of syntax. Instead it requires a variable name followed by a list of values in parentheses. The variable name can have up to 8 characters and must start with a letter. The values in the list can be numbers or other syntactic items (e.g., character strings) and must be separated by spaces. When the **FOREACH** loop is executed, each of the following statements up to the matching **END** statement is executed with the variable replaced by successive values from the **FOREACH** list wherever it occurs.

In this example the variable *X* takes on the values 15.5, 16.5, and 17. The variable *X* is used as the value of the **VALUE** parameter in the **CONTOUR** statement to generate a contour at each of the values listed in the **FOREACH** statement. The value 16 is purposely omitted from the list so that the contour could be drawn with a separate **CONTOUR** statement using **LINE.TYP**=2. Thus the contour for a boron concentration of 10^{16} is drawn with a dashed line to distinguish it from the other contours. The resulting plot is shown in Figure 5-16.

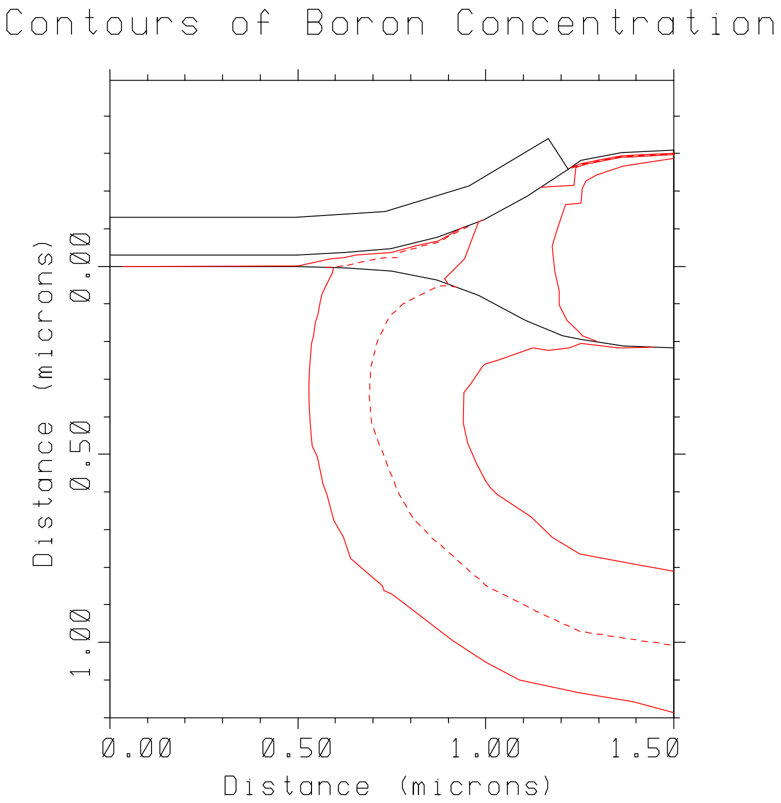


Figure 5-16 Contours of boron concentration produced by input file *s4ex2b.inp*

Vertical Distribution of Point Defects

The distribution of interstitials and vacancies is examined next. By understanding the behavior of these point defects, more accurate and efficient simulations may be set up. It is usually most convenient to normalize the interstitial and vacancy concentration to their equilibrium values, since it is the normalized quantities that are used in the equations for impurity diffusion. Thus, in the **`SELECT`** statements, divide the interstitial and vacancy concentrations by *CI.STAR* and *CV.STAR*, the equilibrium concentrations for interstitials and vacancies, respectively.

The first **`PLOT.1D`** statement plots the distribution of interstitials along a vertical section at **`X.VALUE`**=0. The maximum value on the *x* axis (which corresponds to depth, in this case) is given by **`RIGHT`**. The minimum value on the *y* axis is set to -1 with the **`BOTTOM`** parameter. The normalized vacancy concentration is plotted on the same graph, using line type 2 and color 6. The **`LABEL`** statements help us to remember which curve is which on the plot.

The resulting plot, shown in Figure 5-17, reveals several interesting things about the point defect concentrations during oxidation:

- The point defect profiles extend to a much greater depth than the dopant profiles. Thus a deeper simulation structure is needed, although in this case a structure only 50 to 100 microns deep would have sufficed.
- The oxidation has produced a greatly enhanced interstitial concentration near the surface, and a greatly reduced vacancy concentration. The diffusivity of impurities that diffuse with interstitials (e.g. boron and phosphorus) is enhanced, while the diffusion of impurities that diffuse with vacancies is retarded (e.g., antimony).

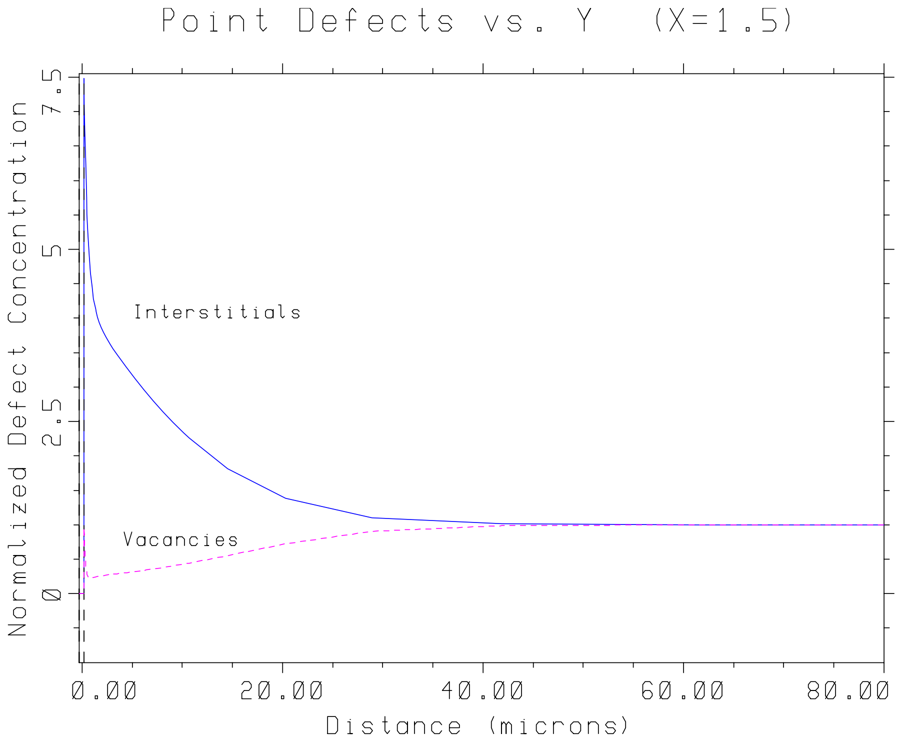


Figure 5-17 Concentration of point defects vs. depth, as plotted by input file *s4ex2b.inp*

Lateral Distribution of Point Defects

The distribution of point defects across the width of the device is examined to determine how dense the horizontal grid must be deep in the structure. The next set of **SELECT**, **PLOT.1D**, and **LABEL** statements plot the point defect distributions across the width of the device at y=2 microns. The sequence of statements is the same as for the previous plot, except that **Y.VALUE**=2 has been specified on the **PLOT.1D** statements instead of **X.VALUE**=0. The results are shown in Figure 5-18.

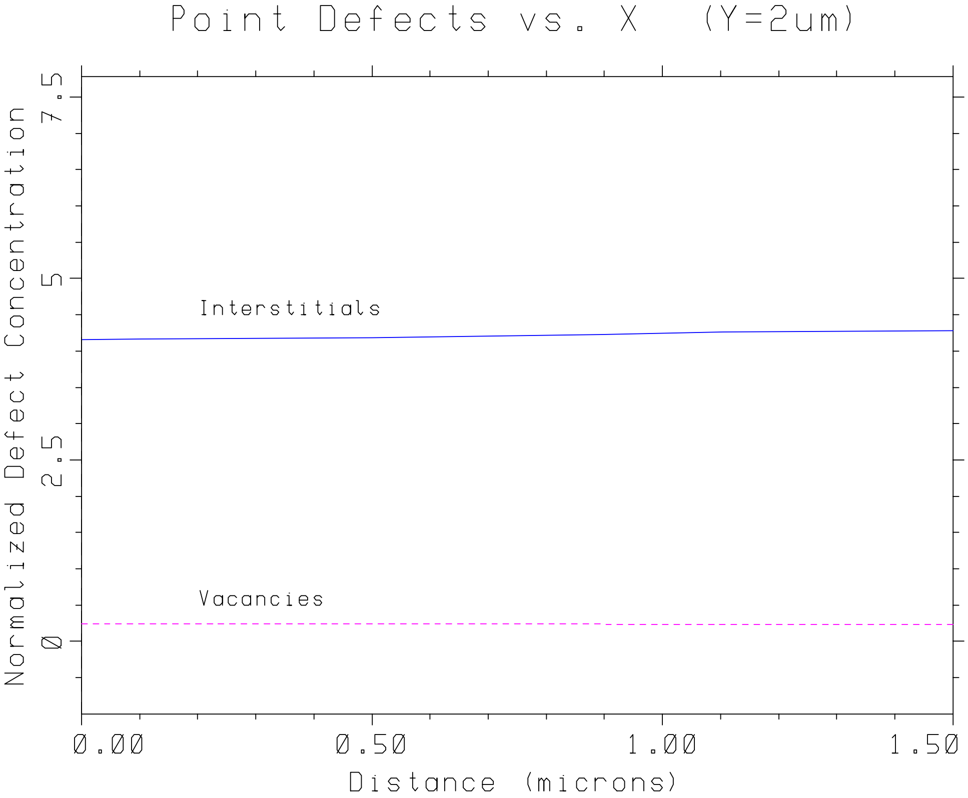


Figure 5-18 Concentration of point defects vs. width, as plotted by input file *s4ex2b.inp*

Although visually uninteresting, this plot reveals a very important property of the point defect distributions—because point defects diffuse so rapidly, the point defect profiles deep in the structure are essentially one-dimensional. Thus it is possible to eliminate the vertical grid lines deep in the structure in order to reduce the simulation time.Shaded Contours of Interstitial Concentration

Finally, shaded contours of interstitial concentrations are plotted. The procedure is the same as for plotting the boron contours. The axes and material boundaries are plotted with a **PLOT.2D** statement. In this case, **Y.MAX**=2 microns is used to plot a deeper section of the device. The **SELECT** statement is used to specify the normalized interstitial concentration as the plot quantity and to give a title for the plot. Again, the general expression capability of the **SELECT** statement is used to scale the interstitial concentration by the equilibrium value (*CI.STAR*). A series of shaded contours are plotted, using the **FOREACH**, **COLOR**, and **END** statements. **LABEL** statements with shaded rectangles are used to create a legend for the plot.

A final **`PLOT.2D`** with **`^AXES`** and **`^CLEAR`** is used to replot the boundaries of the structure. The final plot is shown in Figure 5-19.

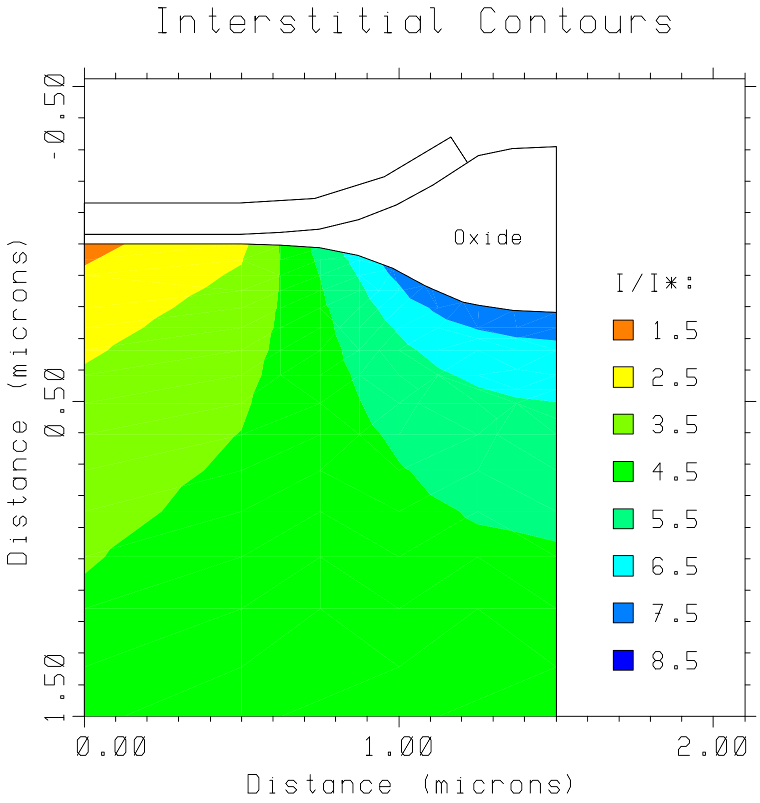


Figure 5-19 Contours of interstitial concentration, as plotted by input file *s4ex2b.inp*

Local Oxidation Summation

This section has shown how the shape of an isolation structure and the corresponding impurity profiles can be determined by separate simulations. In many cases this is the most efficient way to obtain an accurate simulation, because it allows the grids and solution methods to be optimized for each aspect of the problem. Adjustments to the process can be analyzed more quickly; often it is only necessary to repeat one of the simulations. As a final check, the two simulations could be combined, using a somewhat denser mesh. This solution would be more time-consuming, but a single simulation should suffice to verify the results obtained previously.

Point Defect Models

This example shows the differences in results using the three point defect models:

- The **`PD.FERMI`** model is the simplest point defect model; it is the fastest of the three, because it assumes that the point defect concentrations are at equilibrium and thus do not need to be calculated explicitly.

  The **`PD.FERMI`** model does not simulate the effects of nonequilibrium point defect concentrations on impurities, nor the effects of impurities on point defect concentrations (except for the dependence on Fermi level).
- The **`PD.TRANS`** model simulates the generation, diffusion, and recombination of point defects in two dimensions; it is slower, but is necessary for simulation of oxidation-enhanced and transient-enhanced diffusion.

  The **`PD.TRANS`** model simulates the effects of nonequilibrium point defect concentrations on impurity diffusion, but does not simulate the effects of impurity diffusion on the point defect profiles.
- The **`PD.FULL`** model simulates both the effects of nonequilibrium point defect profiles on impurity diffusion and the impact of impurity diffusion on the point defect distributions.

  The **`PD.FULL`** model is required for accurate simulation of high-concentration effects (e.g., phosphorus kink and tail and emitter push) and is often the best model for simulation of transient diffusion enhancement caused by implantation damage

The input file *s4ex3.inp*, shown in Figures 5-20 and 5-21, demonstrates some of the differences among the point defect models. It also illustrates a procedure for simulating alternative processing sequences. In this case, alternative models are being examined, but the procedure can also be used to simulate run splits.

```
$ TSUPREM-4 -- Example 3
$
$ Simulate the diffusion of impurities and point defects using the various
$ defect models.

MESH       GRID.FAC=0.5

INITIALIZE <100>  BORON=1E10

$ Implant phosphorus and boron
IMPLANT    PHOSPHORUS DOSE=2.0E15  ENERGY=50
IMPLANT    BORON      DOSE=1.0E13  ENERGY=120

$ Specify the point defect model (no point defects)
METHOD     PD.FERMI

$ Save the structure
SAVEFILE   OUT.FILE=S4EX3S

$ Perform the diffusion using the Fermi point defect model
DIFFUSION  TEMP=900  TIME=20  DRYO2

$ Plot 1-D profiles of concentration of boron and antimony (Fermi)
SELECT     Z=LOG10(PHOSPHORUS) TITLE="Comparison of Point Defect Models" +
                                LABEL=log10(concentration)
PLOT.1D    RIGHT=1.2  TOP=21  BOTTOM=14  LINE.TYP=3  COLOR=3  SYMBOL=5
SELECT     Z=LOG10(BORON)
PLOT.1D    ^AXES  ^CLEAR  LINE.TYP=3  COLOR=2  SYMBOL=4
```

Figure 5-20 First part of input file *s4ex3.inp*, showing processing and plotting using the **PD.FERMI** point defect model

```
$ Read in the structure
LOADFILE   IN.FILE=S4EX3S

$ Perform the diffusion again using the 2-D point defect model
METHOD     PD.TRANS
DIFFUSION  TEMP=900  TIME=20  DRYO2

$ Add profiles of phosphorus and boron (from the PD.TRANS model) to first
plot
SELECT     Z=LOG10(PHOSPHORUS)
PLOT.1D    ^AXES  ^CLEAR  LINE.TYP=2  COLOR=3  SYMBOL=3
SELECT     Z=LOG10(BORON)
PLOT.1D    ^AXES  ^CLEAR  LINE.TYP=2  COLOR=2  SYMBOL=6

$ Read in the structure
LOADFILE   IN.FILE=S4EX3S

$ Perform the diffusion using the full point defect model
METHOD     PD.FULL
DIFFUSION  TEMP=900  TIME=20  DRYO2

$ Add results to previous plot
SELECT     Z=LOG10(PHOSPHORUS)  TITLE=”Comparison of Point Defect Models” +
                                LABEL=log10(concentration)
PLOT.1D    ^AXES  ^CLEAR  LINE.TYP=1  COLOR=3  SYMBOL=1
SELECT     Z=LOG10(BORON)
PLOT.1D    ^AXES  ^CLEAR  LINE.TYP=1  COLOR=2  SYMBOL=2

$ Label the line types
LABEL      X=0.8  Y=20.5  LABEL=”Phosphorus (PD.FERMI)” +
                        LINE.TYP=3  C.LINE=3  SYMBOL=5  C.SYMB=3
LABEL      X=0.8  Y=20.1  LABEL=”Phosphorus (PD.TRANS)” +
                        LINE.TYP=2  C.LINE=3  SYMBOL=3  C.SYMB=3
LABEL      X=0.8  Y=19.7  LABEL=”Phosphorus (PD.FULL)” +
                        LINE.TYP=1  C.LINE=3  SYMBOL=1  C.SYMB=3
LABEL      X=0.8  Y=19.3  LABEL=”Boron      (PD.FERMI)” +
                        LINE.TYP=3  C.LINE=2  SYMBOL=4  C.SYMB=2
LABEL      X=0.8  Y=18.9  LABEL=”Boron      (PD.TRANS)” +
                        LINE.TYP=2  C.LINE=2  SYMBOL=6  C.SYMB=2
LABEL      X=0.8  Y=18.5  LABEL=”Boron      (PD.FULL)” +
                        LINE.TYP=1  C.LINE=2  SYMBOL=2  C.SYMB=2

$ Print junction locations
SELECT     Z=DOPING
PRINT.1D   SPOT=0  LAYERS

$ Plot 1-D profile of interstitials
SELECT     Z=INTER/CI.STAR  TITLE=”Point Defect Profiles” +
                            LABEL=”Normalized Concentration”
PLOT.1D    RIGHT=20.0  COLOR=4

$ Add 1-D profile of vacancies to second graph
SELECT     Z=VACAN/CV.STAR
PLOT.1D    ^AXES  ^CLEAR  LINE.TYP=2  COLOR=6

$ Label the plot
LABEL      X=5.5 Y=10   LABEL=”Interstitials”
LABEL      X=1   Y=1.0  LABEL=”Vacancies”
```

Figure 5-21 Second part of input file *s4ex3.inp*, using the full 2-D point defect model

Creating the Test Structure

To create the test structure, use the following directions.

Automatic Grid Generation

Because no **LINE** statements are specified in this example, a grid is generated automatically. In the *x* direction, two vertical lines are generated, at *x*=0 and *x*=1 micron. In the *y* direction, a more complicated grid is generated. The details of this grid are specified by a **MESH** statement in the *s4init* file (see Chapter 3, Automatic Grid Generation, p. 3-5, “Explicit Specification of Grid Structure,” p. 3-3, and Appendix AA). The default automatic grid extends to a depth of 200 microns to accommodate the deep diffusion of interstitials and vacancies.

The **GRID.FAC** parameter on the **MESH** statement provides an easy way to increase or decrease the grid spacings throughout a simulation. In this example, all grid spacings are multiplied by a factor of 0.5, doubling the grid density. The **GRID.FAC** parameter makes it easy to determine how the accuracy of the simulation depends on the grid spacing: simply decrease the value of **GRID.FAC** until the changes in the simulated result become insignificant.

Outline of Example

This example compares the three point defect models for two impurities, phosphorus and boron. A large implanted dose of phosphorus is used to demonstrate high-concentration effects. A smaller and deeper boron implant is used to monitor coupling between impurities caused by point defects. A dry oxidizing ambient is used to drive in the impurities. The oxidation produces point defects that enhance the impurity diffusion. The oxide layer also prevents impurities from escaping from the surface of the structure. After performing the implants, the structure is saved. By reloading the saved structure the simulation can be resumed from this point as many times as desired.

Oxidation and Plotting of Impurity Profiles

- Simulation procedure
- **PD.FERMI** and **PD.TRANS** models
- **PD.FULL** model
- Printing junction depth
- Doping and layer information

Simulation Procedure

The oxidation (20 minutes at 900° at a dry oxygen ambient) using the **PD.FERMI** model is first simulated. Because the **PD.FERMI** model is used, the effects of oxidation on the impurity diffusion are not simulated. The results of diffusion with the **PD.FERMI** model are plotted with a pair of **SELECT**/**PLOT.1D** sequences. A single plot is used for the phosphorus and boron concentrations produced by both models, using various different line types, symbols, and colors to distinguish the different profiles. Next, the saved structure is restored, using the **LOADFILE** statement, **PD.TRANS** model is selected. The oxidation step is repeated and the resulting profiles obtained using the **PD.TRANS** model are added to the plot. The procedure is repeated, loading the saved structure and simulating the diffusion, this time with the **PD.FULL** model. Finally, **LABEL** statements are added to create a legend, showing the significance of the different line types and symbols.

PD.FERMI and PD.TRANS Models

The final plot is shown in Figure 5-22. The lines with the longer and shorter dashes show the profiles obtained with the **PD.FERMI** and **PD.TRANS** models, respectively. The phosphorus profiles have flat tops and steep tails (the gentler tail beyond *y*=0.4 microns is due to channelling during the implant), and the boron profiles are symmetrical. The effect of oxidation-enhanced diffusion is significant, increasing the junction location from about *y*=0.26 microns when the **PD.FERMI** model is used to about *y*=0.36 microns with the **PD.TRANS** model.

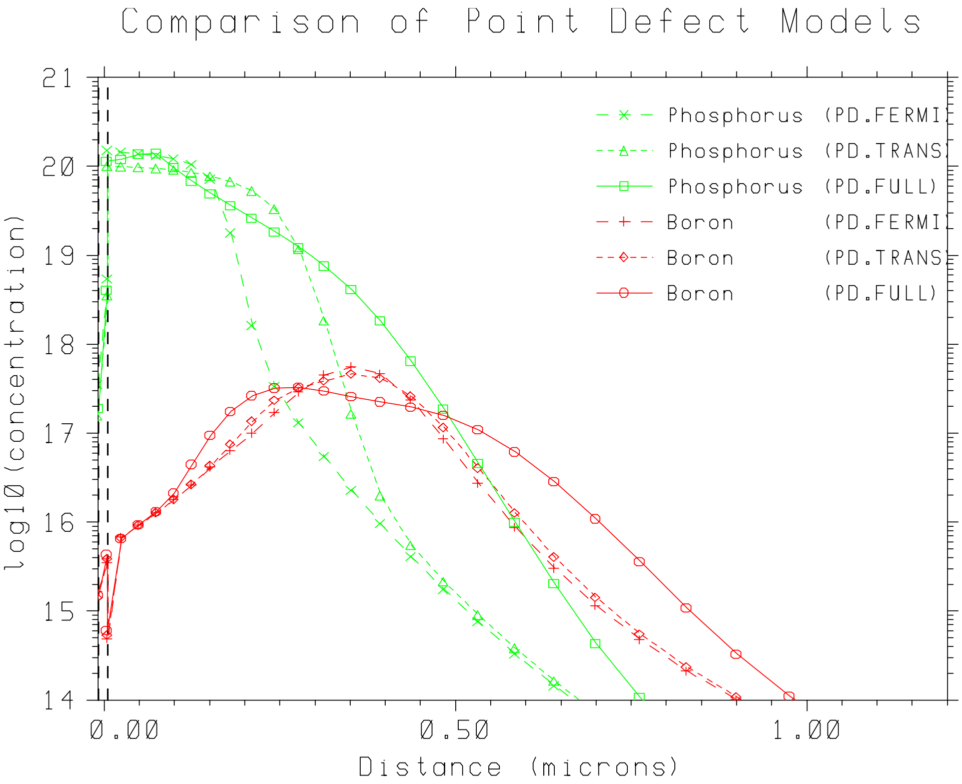


Figure 5-22 Profiles with **PD.FERMI** and **PD.FULL** models, from *s4ex3.inp*

PD.FULL Model

The solid lines show the results obtained with the **PD.FULL** model. With this model, the phosphorus profile is no longer flat on top and has developed an extended tail, while the boron profile has lost its symmetry. Both impurities have diffused significantly farther than they did with the **PD.TRANS** model, with the junction location increasing to about *y*=0.51 microns. All of these effects are caused by the two-way interactions between impurities and point defects: Near the peak of the phosphorus profile, interstitials interact with substitutional phosphorus atoms to form interstitial-phosphorus pairs. (Actually, they may result in phosphorus atoms in interstitial sites, but the result is the same.) The pairs diffuse into the substrate where they break up, leaving a substitutional phosphorus atom and an interstitial. The excess interstitials introduced into the silicon substrate by this process then contributes to the enhanced diffusion observed in both the boron profile and the phosphorus tail. The removal of interstitials from the region of the phosphorus peak reduces the effective diffusivity in this area, creating a kink between the region of retarded diffusivity near the peak and the region of enhanced diffusion around the tail of the profile.

Printing Junction Depth

Plots such as the one in Figure 5-22 are often used to determine the approximate junction locations in simulated structures. In many cases, however, a more accurate value is needed than that which can be obtained from the plot. A convenient way get a more accurate value is with the **PRINT.1D** statement. To obtain the junction depth, specify *DOPING* on the **SELECT** statement and then use **PRINT.1D** with the **SPOT** or **LAYERS** parameters. In this case, both are used, the results being shown in Figure 5-23. The **SPOT** parameter requests that the locations at which the selected expression has the specified value be printed. This case prints the points at which the net doping is zero, i.e., the metallurgical junctions. From Figure 5-23, observe that the junction is at *y*=0.50 in the structure.

```
$ Print junction locations
SELECT     Z=DOPING
PRINT.1D   SPOT=0  LAYERS
** Printing along X.VALUE=0:

   Value is 0 at 0.498930 microns.

 Num      Material       Top     Bottom  Thickness         Integral
   1         oxide   -0.0078     0.0045     0.0123       2.5782e+12
   2       silicon    0.0045     0.4989     0.4944       1.9716e+15
   3       silicon    0.4989   200.0000   199.5011      -8.1086e+11
```

Figure 5-23 Output produced by **PRINT.1D** statement in input file *s4ex3.inp*

Doping and Layer Information

The **LAYERS** parameter prints more information about the doping in the structure. The **LAYERS** output (included in Figure 5-23) gives the top and bottom coordinates of the oxide, the n-type surface region, and the p-type substrate. In addition, it calculates the thickness of each layer and the integral of the selected expression (doping, in this case) over each layer.

Note:

> ***The definition of a layer depends on the selected expression—the boundaries between layers are taken to be material interfaces or the points where the selected quantity is zero.***

To use **PRINT.1D LAYERS** to get junction depths, you must specify **SELECT Z=DOPING**.

Point Defect Profiles

The last step is to examine the interstitial and vacancy concentrations in the structure. Although the point defect profiles do not have a direct effect on device performance, they do aid understanding how the final doping profiles were produced. The point defect profiles, as plotted by the last two **SELECT/PLOT.1D** pairs, are shown in Figure 5-24. *Note* that the interstitial and vacancy concentrations have been normalized by *CI.STAR* and *CV.STAR*, the equilibrium interstitial and vacancy concentrations, respectively. Figure 5-24 shows that the interstitial profile has a large peak about 0.4 microns below the surface. This peak is produced by the interaction with the phosphorus profile. The smaller peak at the surface is caused by injection of interstitials by the oxidation process.

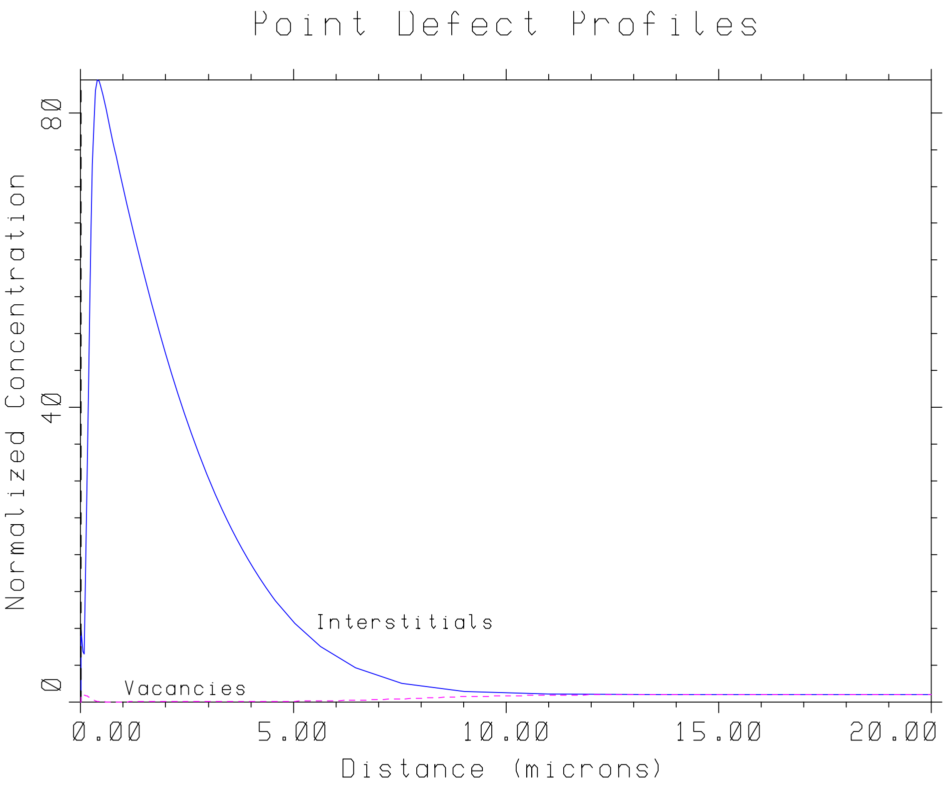


Figure 5-24 Point defect profiles plotted by *s4ex3.inp*

Commentary

Because of the extra accuracy afforded by the **PD.FULL** model, one may be tempted to use it at all times. This would be a good idea, except that the **PD.FULL** model is slower than the **PD.TRANS** model and there are many cases in which the extra accuracy obtained with the **PD.FULL** model is negligible. Similar arguments can be made in comparing the **PD.TRANS** and **PD.FERMI** models.

Choosing a Point Defect Model

The simplest way to determine which point defect model is needed in a particular simulation is by trial and error. Although it may appear easier to use **PD.FULL**, it is usually faster to set up a simulation using the **PD.FERMI** model and then switch to **PD.TRANS** or **PD.FULL** only after correct simulation of the structure and approximately correct simulation of the impurity profiles have been verified.

There are several cases in where it can be determined in advance which point defect model is most appropriate. Because point defects are generated only during oxidation or by implantation (with the **DAMAGE** parameter), there is no need to use **PD.TRANS** or **PD.FULL** before the first oxidation or implantation (with **DAMAGE**) step. During inert diffusions, any point defects present in the structure recombine and approach their equilibrium values. After a sufficiently long inert diffusion, the **PD.FERMI** model can be used without loss of accuracy.

Finally, it should be noted that the point defects affect the diffusivity of impurities. If the impurity concentrations are small, or if the gradients of the concentrations are small, there is little diffusion, and no need to use one of the more expensive point defect models. The **PD.FULL** should be considered whenever concentrations (particularly of phosphorus) are very high, or whenever it is necessary to model implant damage effects.

Advanced Examples

This chapter presents several examples that illustrate the application of TSUPREM-4 to real-world problems. Most of these examples show full, 2-D simulations of complete processes. Examples include:

- Simulations of several specialized processes such as poly-buffered LOCOS
- A power DMOS process
- A silicon-on-insulator process
- A SiGe HBT process
- Demonstrations of a number of techniques that are useful for a wide range of applications.

Sections in this chapter include:

- NMOS LDD Process
- Trench Implant Simulation
- Poly-Buffered LOCOS
- CMOS Process
- DMOS Power Transistor
- SOI MOSFET
- MOSFET with Self-Aligned Silicides
- Polysilicon Emitter Study
- SiGe HBT Process

Note:

> ***Because some of the examples in this chapter include simulations of complete processes, they require more computer time than the examples of Chapter 5. The initial grids in these examples have been carefully chosen to give reasonable accuracy while minimizing computer time requirements. These are the grids that would be used for the bulk of a process design or analysis project. Simpler grids would be used for initial checkout of the simulation input file, while finer grids would be used for a final check. Similarly, the oxidation and point defect models have been chosen for reasonable accuracy. In some cases, slower but more accurate models could be used for a final check of the simulation.***

NMOS LDD Process

This application presents a complete simulation of an n-channel MOS transistor with a lightly-doped drain. It illustrates how mask information extracted from a layout by Taurus Layout—IC Layout Interface can be used by TSUPREM-4.

Because this application is rather lengthy, it has been broken into four parts:

- Input file *s4ex4a.inp* simulates the growth of the field oxide and the consequent boron diffusion; the results are stored in the structure file *S4EX4AS*.
- The source/drain processing is simulated (input file *s4ex4b.inp*), and the results are stored in structure file *S4EX4BS*.
- The input file *s4ex4c.inp* reflects the half-structure about the left edge to form the complete NMOS device, saving the structure in file *S4EX4CS* and plotting contours of boron and arsenic.
- The input file *s4ex4d.inp* is used to extract electrical characteristics of the final structure.

The input statements in file *s4ex4a.inp*, shown in Figures 6-1 and 6-3, simulate the initial portion of the lightly-doped drain NMOS process up through the gate region enhancement implant.

Creating the Initial Structure

This example uses an automatically generated, 2-D grid with adaptive gridding. The grid in the *x* direction is derived from mask information extracted from a layout file by Taurus Layout. The width of the grid is set equal to the width of the cut line specified in Taurus Layout. The grid is made finer near mask edges and coarser far from mask edges. Vertical grid lines are eliminated deep in the structure, to save simulation time. The grid in the *y* direction is generated automatically using the default parameters; these default parameters are set on **MESH** statements in the *s4init* file, but can be changed as needed. See Chapter 3, Automatic Grid Generation, p. 3-5, Explicit Specification of Grid Structure, p. 3-3, and Appendix A.

To automatically generate a 2-D grid, a **MASK** statement is needed to read the mask information produced by Taurus Layout and an **INITIALIZE** statement is used to do the grid generation. No **LINE** statements are required, and **ELIMINATE** statements are optional.

```
$ TSUPREM4 NMOS transistor simulation
$   Part a:  Through field oxidation

$ Define the grid
MESH       GRID.FAC=1.5
$METHOD     ERR.FAC=2.0

$ Read the mask definition file
MASK       IN.FILE=s4ex4m.tl1  PRINT  GRID="Field,Poly"

$ Initialize the structure
INITIALIZE <100>  BORON=5E15

$ Initial oxidation
DIFFUSION  TIME=30  TEMP=1000  DRY  HCL=5

$ Nitride deposition and field region mask
DEPOSIT    NITRIDE      THICKNESS=0.07  SPACES=4
DEPOSIT    PHOTORESIST  POSITIVE  THICKNESS=1
EXPOSE     MASK=Field
DEVELOP
ETCH       NITRIDE      TRAP
ETCH       OXIDE        TRAP  UNDERCUT=0.1
ETCH       SILICON      TRAP  THICKNES=0.25  UNDERCUT=0.1

$ Boron field implant
IMPLANT    BORON  DOSE=5E12  ENERGY=50  TILT=7  ROTATION=30
ETCH       PHOTORESIST  ALL

$ Field oxidation
METHOD     PD.TRANS  COMPRESS
DIFFUSION  TIME=20   TEMP=800   T.FINAL=1000
DIFFUSION  TIME=180  TEMP=1000  WETO2
DIFFUSION  TIME=20   TEMP=1000  T.FINAL=800
ETCH       NITRIDE   ALL

$ Unmasked enhancement implant
IMPLANT    BORON  DOSE=1E12  ENERGY=40  TILT=7  ROTATION=30

$ Save structure
SAVEFILE   OUT.FILE=S4EX4AS
```

Figure 6-1 First part of input file *s4ex4a.inp*: Setting up the grid for simulating an NMOS process

Setting the Grid Density

To save execution time, the recommended procedure when developing a new TSUPREM-4 simulation is to start with a very coarse grid. After an input file has been entered and is running correctly (no syntax errors, all etches in the correct places, etc.), a finer grid is used for the bulk of the simulation work. Lastly, an even finer grid is used to get final answers or to verify that additional grid points (with the associated increased execution time) do not significantly improve the accuracy of the answers.

The **`GRID.FAC`** parameter on the **`MESH`** statement makes it easy to adjust the grid density during the various stages of a simulation project. In this example, **`GRID.FAC`** has been set to 1.5, so that all grid spacings are 1.5 times their specified value. This gives a rather coarse grid for efficient simulations.

When the results are satisfactory with this grid, **GRID.FAC** can be decreased to 1.0 or 0.5 to check the results.

Adaptive Gridding

The **ERR.FAC** parameter on the **METHOD** statement is used in a similar way to control the accuracy criteria used for adaptive gridding. A value of 2.0 could be used to reduce the accuracy for a faster simulation, but for this example the default value of 1.0 is satisfactory. When the simulation is running correctly, this can be reduced to 0.7 or 0.5 to check the results.

Masking Information

The **MASK** statement reads mask descriptions from the file *s4ex4m.tl1*. This file, produced by Taurus Layout, describes the mask levels present over a cross-section of the device layout. The **PRINT** parameter specifies that the mask information should be printed after it is read. The result is shown in Figure 6-2. Four mask levels, named *Field*, *Poly*, *Contact*, and *Metal* are defined. These names are used to refer to the masks as they are needed later in the processes. The **GRID** parameter specifies that only the *Field* and *Poly* layers in the mask file are considered when calculating the horizontal grid spacing; the *Contact* and *Metal* are ignored.

Note:

> ***There are no references to mask coordinates in the TSUPREM-4 input file. The mask file contains all the layout information required by the simulation. The same input file can be used to simulate other structures simply by specifying a different mask file.***

```
$ Read the mask definition file
MASK       IN.FILE=s4ex4m.tl1  PRINT

Comments from mask data file "s4ex4m.tl1":
/ Mask definition file s4ex4m.tl1, for use with s4ex4[abc].
(End of comments from mask data file)

The following masks are currently defined (locations in microns):
   Name:  Field        min X:    0.0000   max X:    5.0000
       opaque between    0.0000  and   3.9000
   Name:  Poly         min X:    0.0000   max X:    5.0000
       opaque between    0.0000  and   0.6500
   Name:  Contact      min X:    0.0000   max X:    5.0000
       opaque between    0.0000  and   1.9500
       opaque between    3.2500  and   5.0000
   Name:  Metal        min X:    0.0000   max X:    5.0000
       opaque between    1.3000  and   5.0000
```

Figure 6-2 Listing of mask information read from file *s4ex4m.tl1*

Field Isolation Simulation

1. The first step is to grow a pad oxide. The default **VERTICAL** model is sufficient for this planar oxidation step; the **PD.FERMI** model for defects (the default) is acceptable because there is no significant doping in the structure yet.

2. The next step is to define the active and field isolation regions of the structure. A typical photolithography sequence is used:
   a. Deposit a layer to be patterned (nitride, in this case).
   b. Deposit a a layer of photoresist (positive resist, in this example).
   c. Expose the resist, using the appropriate mask (the *Field* mask).
   d. Use the **`DEVELOP`** statement to remove the exposed photoresist.
   e. Etch the underlying layers, using the **`TRAP`** etch model. The remaining photoresist serves as a mask for the etch. In this case, the nitride is etched first, then the oxide, then the silicon. The oxide and silicon etches specify undercutting of the mask layer by 0.1 and 0.25 microns, respectively.
   f. In this example, the photoresist is also used to mask the boron field implant.
   g. Remove the remaining photoresist.

Note:

Don't forget to remove the remaining photoresist at the end of the photolithography sequence.

3. The next step is the field oxidation. The **`PD.TRANS`** model is used, resulting in the modeling of interstitials and vacancies, and hence oxidation-enhanced diffusion; the **`COMPRESS`** oxidation model is also selected at this point, because all further oxidations will be nonplanar. (If the details of the bird's beak shape were important, the **`VISCOEL`** model with stress dependence could be selected at this point). The field oxidation is done in three steps:
   a. For the first 20 minutes, the temperature is ramped up from 800° C to 1000° C at the rate of 10° C/minute in an inert ambient.
   b. The second step is a wet oxidation for 180 minutes at a constant temperature of 1000° C.
   c. Finally, the temperature is ramped down from 1000° C to 800° C at the rate of 10° C/minute in an inert ambient.
4. The final process steps specified in the first input file are the removal of the nitride oxidation mask and the implantation of boron for adjusting the n-channel threshold voltage.

The structure to this point is saved in the structure file *S4EX4AS* for use in continued simulation of the source and drain regions.

Displaying the Plot

Figure 6-3 shows the input statements for displaying the results thus far. The first **`PLOT.2D`** statement plots the grid at this point (Figure 6-4). As planned, the grid is fairly coarse; some refinement by adaptive gridding is evident near the surface and around the field oxide. The structure at this point is shown in Figure 6-5. Contour lines of boron concentration are plotted starting at a concentration of 10^{15} and extending up to a concentration of 10^{20} atoms/cm^3 in half-decade steps.

```
$ Plot the initial NMOS structure
SELECT    Z=LOG10(BORON)  TITLE="LDD Process - NMOS Isolation Region"
PLOT.2D     SCALE GRID C.GRID=2 Y.MAX=2.0
PLOT.2D     SCALE  Y.MAX=2.0

$ Color fill the regions
COLOR       SILICON  COLOR=7
COLOR       OXIDE    COLOR=5

$ Plot contours of boron
FOREACH     X  (15 TO 20 STEP 0.5)
  CONTOUR    VALUE=X  LINE=5  COLOR=2
END

$ Replot boundaries
PLOT.2D     ^AX  ^CL

$ Print doping information under field oxide
SELECT      Z=DOPING
PRINT.1D    X.VALUE=4.5  X.MAX=3
```

Figure 6-3 Second part of input file *s4ex4a.inp*, for simulating an NMOS process

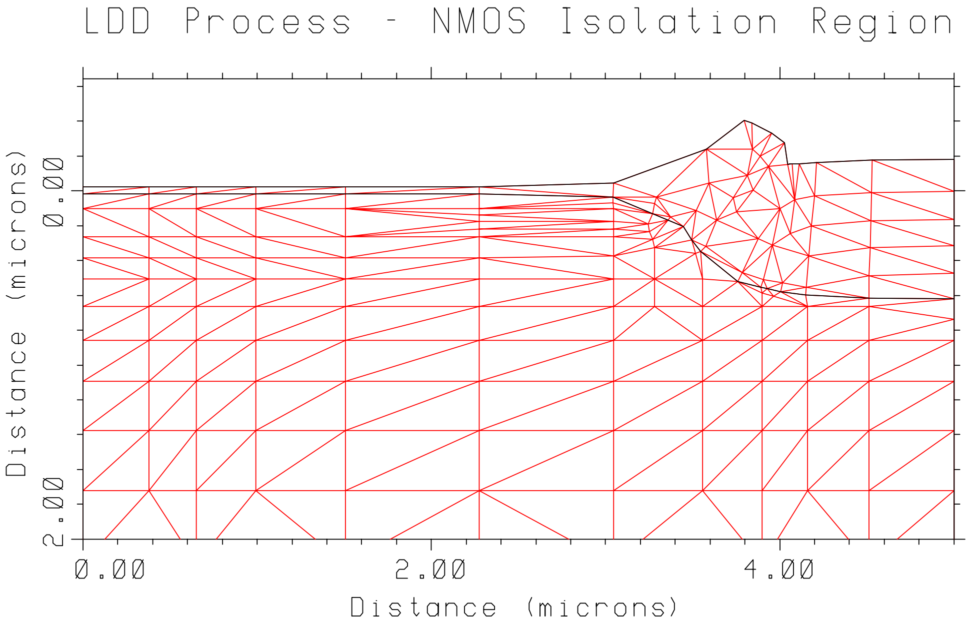


Figure 6-4 Grid after formation of isolation region, plotted by *s4ex4a.inp*

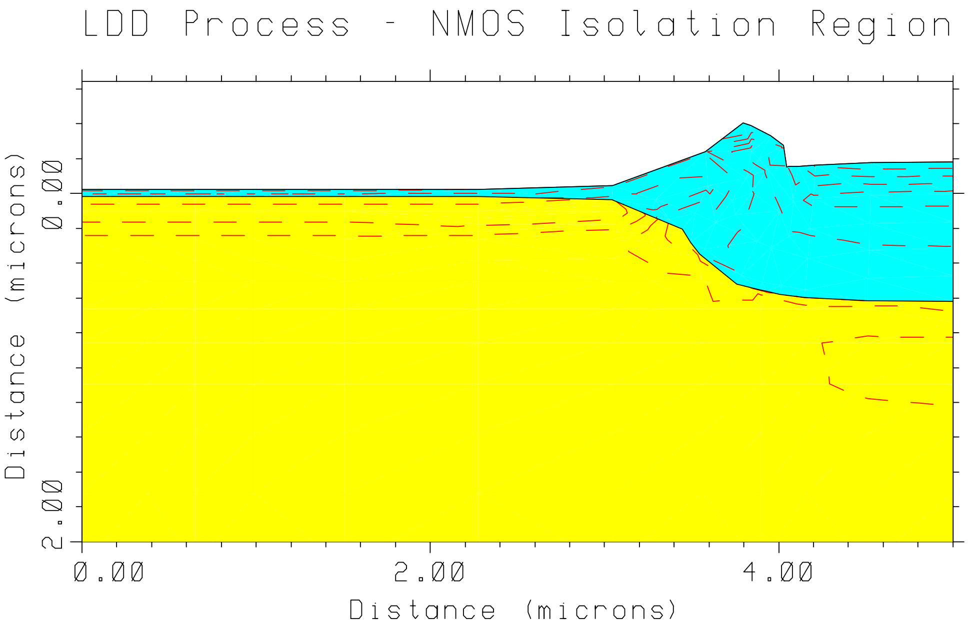


Figure 6-5 Structure with contours of boron concentration, after formation of isolation region, as plotted by file *s4ex4a.inp*

Active Region Simulation

The input statements in the file *s4ex4b.inp* complete the processing of the NMOS transistor. The listing of this input file is presented in Figures 6-6 and 6-8. The starting structure for this stage of the simulation is read from the file *S4EX4AS* generated by the simulation input file *s4ex4a.inp*. The grid spacing and error control factors are not saved in the structure file, so they must be reset to their desired values with the **MESH** and **METHOD** statements, respectively. Similarly, the mask information from Taurus Layout must be reread with the **MASK** statement.

The first process steps in this input file define and oxidize a polysilicon gate. The *Poly* mask is used in a typical photolithography sequence (i.e., **DEPOSITION**, **EXPOSE**, **DEVELOP**, and **ETCH**). The **COMPRESS** model is used to simulate the oxidation of the polysilicon gate and the **PD.TRANS** model is used for diffusion. These models are not explicitly specified in input file *s4ex4b.inp*; rather, they are set in *s4ex4a.inp*, saved (automatically) in the structure file *S4EX4AS*, and then read in by the **INITIALIZE** statement in *s4ex4b.inp*.

Modeling Polysilicon

The statement **MATERIAL MAT=POLY ^POLYCRYS** disables the advanced models for grain growth and impurity diffusion in polysilicon (to reduce the computation time). For studying the details of impurity diffusion and activation in the gate, this statement would be omitted. Also, a finer grid would be used in the poly, either by specifying more grid spaces in the **DEPOSITION** statement or by reducing the relative error for adaptive gridding in polysilicon (i.e., by specifying a smaller value for **REL.ADAP** for impurities in poly).

```
$ TSUPREM4 NMOS transistor simulation
$   Part b:  Through source/drain metallization

$ Set grid spacing and accuracy parameters
MESH      GRID.FAC=1.5
$METHOD    ERR.FAC=2.0

$ Read structure from initial simulation
INITIAL   IN.FILE=S4EX4AS

$ Read the mask definition file
MASK      IN.FILE=s4ex4m.tl1

$ Define polysilicon gate
MATERIAL  MAT=POLY ^POLYCRYS
DEPOSIT   POLYSILICON  THICK=0.4  SPACES=2
DEPOSIT   PHOTORESIST  THICK=1.0
EXPOSE    MASK=Poly
DEVELOP
ETCH      POLYSILICON  TRAP  THICK=0.7  ANGLE=79
ETCH  PHOTORESIST  ALL

$ Oxidize the polysilicon gate
DIFFUSION TIME=30  TEMP=1000  DRYO2

$ LDD implant at a 7-degree tilt
IMPLANT  ARSENIC DOSE=5E13 ENERGY=50  TILT=7.0 ROTATION=30  IMPL.TAB=ARSENIC

$ Plot structure
SELECT    Z=LOG10(BORON)  TITLE="LDD Process - After LDD Implant"
PLOT.2D   SCALE  Y.MAX=2.0

$ Add color fill
COLOR     SILICON  COLOR=7
COLOR     OXIDE    COLOR=5
COLOR     POLY     COLOR=3

$ Plot contours
FOREACH   X  (15 TO 18 STEP 0.5)
  CONTOUR   VALUE=X  LINE=5  COLOR=2
END
SELECT    Z=LOG10(ARSENIC)
FOREACH   X  (16 TO 20)
  CONTOUR   VALUE=X  LINE=2  COLOR=4
END

$ Replot boundaries
PLOT.2D   ^AX  ^CL
```

Figure 6-6 First part of input file *s4ex4b.inp*, showing polysilicon gate formation

LDD Implant

Next, the lightly doped source/drain region extension is implanted. Because the implantation is through an oxide, **`IMPL.TAB`**=*arsenic* is used instead of the default dual-Pearson implant moments (the *dual.ars* tables) with channelling tails. The mesh nodes needed to resolve the implanted profile are added automatically through adaptive gridding.

A 2-D plot of the structure, showing contours of boron and arsenic concentration after the LDD implant, is shown in Figure 6-7.

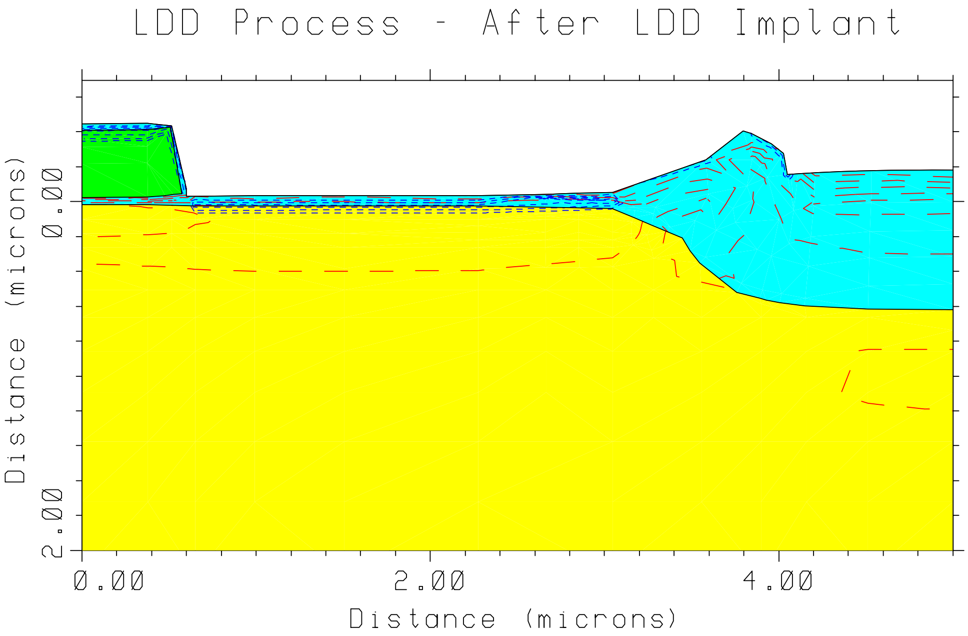


Figure 6-7 NMOS structure after LDD implant, as plotted by file *s4ex4b.inp*

Oxide Spacer and Source/Drain Implant

The remainder of the simulation input appears in Figure 6-8. The next two steps define an oxide sidewall spacer. The **`DEPOSITION`** statement adds a conformal layer of oxide 0.4 microns thick, while the **`ETCH TRAP`** statement removes all oxide within a vertical distance of 0.45 microns from the surface. As a result, the oxide is removed where the surface is planar, but a spacer oxide remains on the sidewalls of the poly gate. This is followed by the implantation of arsenic to form the heavily doped source/drain regions. For this implant into bare silicon, the default implant tables, which include channelling are used. A 15-minute anneal at 950° C is used to activate the arsenic implants.

Source/Drain Contacts

The last two masks are used to locate the source/drain contacts and to pattern the aluminum. A layer of BPSG is used as an insulator between the aluminum and the structure beneath it.

```
$ Define the oxide sidewall spacer
DEPOSIT   OXIDE  THICK=0.4
ETCH      OXIDE  THICK=0.45  TRAP

$ Heavy S/D implant at a 7-degree tilt
IMPLANT   DOSE=1E15  ENERGY=200  ARSENIC  TILT=7.0  ROTATION=30

$ Anneal to activate the arsenic
DIFFUSION TIME=15  TEMP=950

$ Deposit BPSG and cut source/drain contact holes
DEPOSIT   OXIDE  THICKNES=0.7
DEPOSIT   PHOTORESIST  POSITIVE  THICKNESS=1.0
EXPOSE    MASK=Contact
DEVELOP
ETCH      OXIDE  THICKNESS=1.0  TRAP  ANGLE=75
ETCH      PHOTORESIST  ALL

$ Define the metallization
DEPOSIT   ALUMINUM  THICKNESS=1.0
DEPOSIT   PHOTORESIST  POSITIVE  THICKNESS=1.0
EXPOSE    MASK=Metal
DEVELOP
ETCH      ALUMINUM  TRAP  THICKNESS=1.5 ANGLE=75
ETCH      PHOTORESIST  ALL

$ Save the final structure
SAVEFILE  OUT.FILE=S4EX4BS

$ Plot the half NMOS structure
SELECT    Z=LOG10(BORON)  TITLE="LDD Process - Half of NMOS Structure"
PLOT.2D   SCALE  Y.MAX=2.0  GRID C.GRID=2
PLOT.2D   SCALE  Y.MAX=2.0

$ Color fill
COLOR     SILICON  COLOR=7
COLOR     OXIDE    COLOR=5
COLOR     POLY     COLOR=3
COLOR     ALUM     COLOR=2

$ Plot contours
FOREACH   X  (15 TO 18 STEP 0.5)
  CONTOUR   VALUE=X  LINE=5  COLOR=2
END
SELECT    Z=LOG10(ARSENIC)
FOREACH   X  (15 TO 20)
  CONTOUR   VALUE=X  LINE=2  COLOR=4
END

$ Replot boundaries
PLOT.2D   ^AX  ^CL

$ Print doping through drain
SELECT    Z=DOPING
PRINT.1D  LAYERS  X.VALUE=2
```

Figure 6-8 Second part of input file *s4ex4b.inp*, showing source/drain processing and metallization

Plots

The final grid is plotted in Figure 6-9. Adaptive gridding has produced a fine grid where it is needed: in the source/drain region and particularly in the lightly doped extension. The 2-D contour plot of the boron and arsenic concentrations (Figure 6-10) shows that the shallow extension of the source/drain region is well defined. The results of the drain region simulation are saved with the **SAVEFILE** statement in the structure file *S4EX4BS*. This file is used as the basis for forming the complete NMOS device.

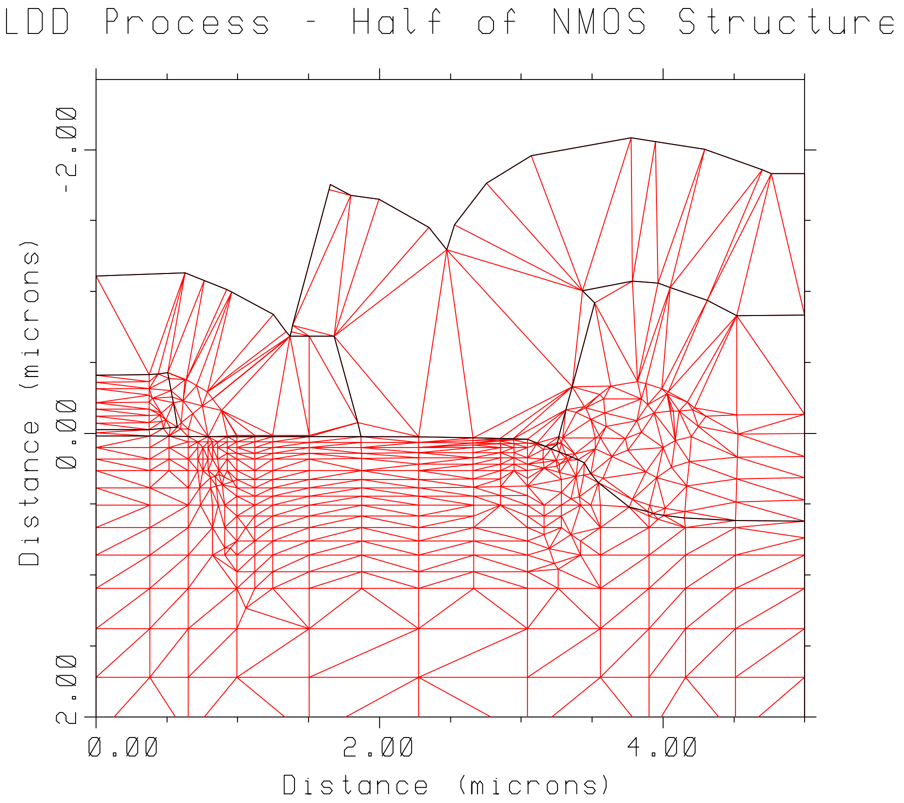


Figure 6-9 Final grid for LDD NMOS example, produced by input files *s4ex4a.inp* and *s4ex4b.inp*

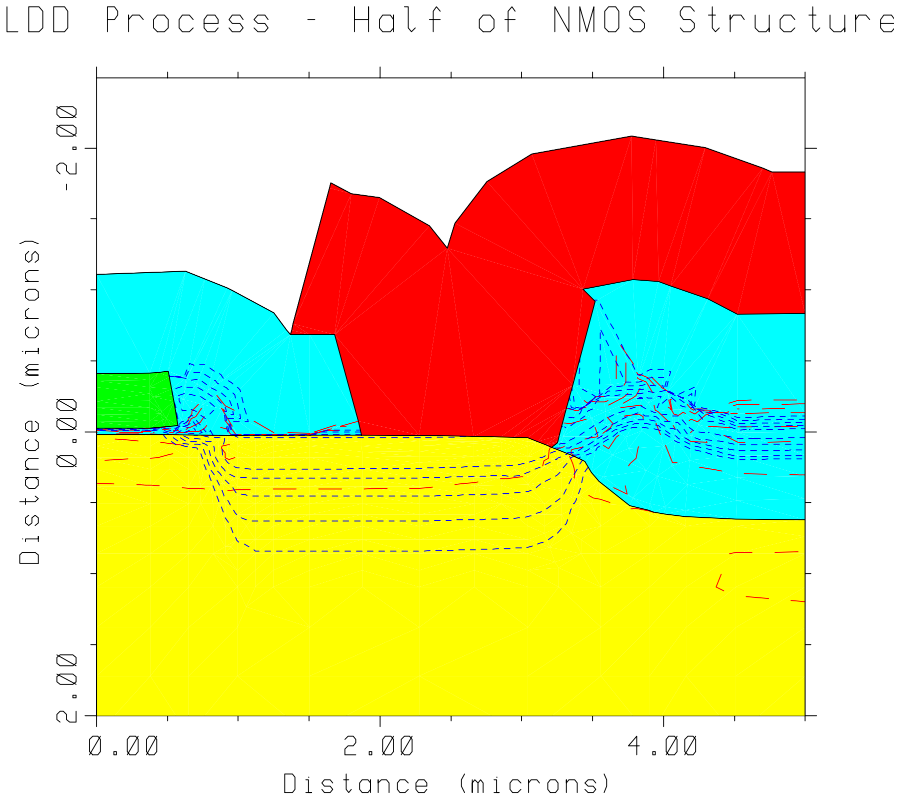


Figure 6-10 Final NMOS structure, as plotted by file *s4ex4b.inp*

Formation of the Complete NMOS Transistor

The complete NMOS transistor is formed by reading in the right half of the structure from the previous simulation and reflecting it about its left edge. The final structure is then plotted, and the various material regions are shaded and labeled. Contours of boron (long dashes) and arsenic (medium dashes) are plotted. The completed structure is saved in file *S4EX4CS*. The input statements for doing this are shown in Figure 6-11; the resulting plot is shown in Figure 6-12.

```
$ TSUPREM4 NMOS transistor simulation
$   Part c:  Formation of complete structure

$ Read right half of structure
INITIAL   IN.FILE=S4EX4BS

$ Reflect about the left edge to form the complete structure
STRUCTURE REFLECT  LEFT

$ Plot the complete NMOS structure
SELECT    Z=LOG10(BORON)  TITLE="Example 4 - Complete NMOS Structure"
PLOT.2D   SCALE  Y.MAX=2.0  Y.MIN=-3.0

$ Color fill
COLOR     SILICON   COLOR=7
LABEL  X=-4.1 Y=-2.5 LABEL="Silicon"     SIZE=.3 C.RECT=7 W.RECT=.4 H.R=.4
COLOR     POLYSILI  COLOR=3
LABEL  X=-1.8 Y=-2.5 LABEL="Polysilicon" SIZE=.3 C.RECT=3 W.RECT=.4 H.R=.4
COLOR     OXIDE     COLOR=5
LABEL  X=1.2 Y=-2.5 LABEL="Oxide"        SIZE=.3 C.RECT=5 W.RECT=.4 H.R=.4
COLOR     ALUMINUM  COLOR=2
LABEL  X=3.2 Y=-2.5 LABEL="Aluminum"     SIZE=.3 C.RECT=2 W.RECT=.4 H.R=.4

$ Plot contours
FOREACH   X  (15 16 17 18)
  CONTOUR   VAL=X  LINE=5  COLOR=2
END
SELECT    Z=LOG10(ARSENIC)
FOREACH   X  (15 16 17 18 19 20)
  CONTOUR   VAL=X  LINE=3  COLOR=4
END

$ Replot boundaries
PLOT.2D   ^AX  ^CL

SAVEFILE  OUT.FILE=S4EX4CS
```

Figure 6-11   Input file *s4ex4c.inp*, for plotting the final LDD NMOS structure

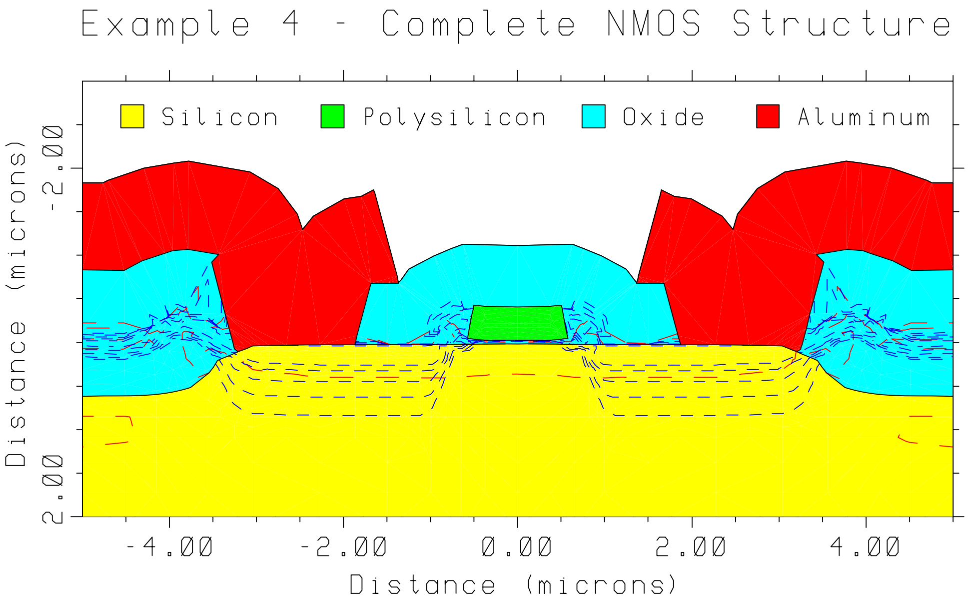


Figure 6-12 Complete NMOS structure, plotted by input file *s4ex4c.inp*

Electrical Extraction

This example shows the extraction of the following electrical information:

- Threshold voltage
- Gate capacitance
- Source/Drain junction capacitance

The statements for performing the electrical extraction are in the input file *s4ex4d.inp*.

Electrical extraction is specified by the **ELECTRICAL** statement. Results are calculated along a vertical slice through the device specified by the value of the **X** parameter. The type of extraction is specified by the **THRESHOLD**, **MOSCAP**, **JCAP**, and **RESISTAN** parameters.

Threshold Voltage

The **THRESHOLD** parameter is used to extract the threshold voltage. The **NMOS** parameter specifies that the type of MOS transistor is NMOS. The position **X** is located at the center of the gate. The gate voltage **V** is stepped from 0 to 2 volts in 0.1-volt increments. The other regions (i.e., the source, drain, and bulk) are grounded. The surface state density **QSS** defaults to $1\text{x}10^{10}/\text{cm}^2$.

Because TSUPREM-4 solves the 1-D Poisson's equation, the extracted quantity is not drain current, but sheet conductance of the channel. However, the drain current can be approximately calculated from the sheet conductance, assuming that the channel is long and wide so that small-geometry effects can be neglected. This example supposes that NMOS with channel length of 1.2 microns and width of 25 microns is measured and that the drain is biased to 0.1 volt to minimize lowering of the energy barrier in the channel region. The scaling factor is the drain voltage multiplied by W/L. **ASSIGN** statements are used to set the values of length, width, and drain voltage and to calculate the scale factor.

The shift in the threshold voltage due to the body effect is examined by changing the back bias. The value of the body-effect parameter is approximately equal to the difference between the threshold voltages at 0and 2.5-volts back bias. This example shows approximately 0.9-volt threshold voltage shift and approximately 1.0-$V^{1/2}$ body effect.

The input statements for extracting and plotting the threshold characteristics are shown in Figure 6-13; the results are shown in Figure 6-16.

```
$ TSUPREM-4 - Electrical Extraction

$ Read structure from Example 4
INITIAL    IN.FILE=S4EX4CS

$ Part A: Threshold voltage
$ Extract the gate bias vs. the sheet conductance in channel region
$ -- VBS=0V
ELECTRIC   X=0.0  THRESHOLD  NMOS  V="0 2 0.1"  OUT.FILE=S4EX4DS1

$ -- VBS=-2.5V
ELECTRIC   X=0.0  THRESHOLD  NMOS  V="0 3 0.05"  VB=-2.5 +
           OUT.FILE=S4EX4DS2

$ Plot the Vgs vs Ids
$ -- Define the scale to convert the sheet conductance to the current
ASSIGN     NAME=Lch   N.VAL=1.2
ASSIGN     NAME=Wch   N.VAL=25.0
ASSIGN     NAME=Vds   N.VAL=0.1
ASSIGN     NAME=Scale N.VAL=(@Vds*@Wch/@Lch)
$ -- Plot
SELECT     TITLE="Vgs vs. Ids"
VIEWPORT   X.MAX=0.5
PLOT.1D    IN.FILE=S4EX4DS1  Y.SCALE=@Scale +
           Y.LABEL="I(Drain) (Amps)"  X.LABEL="V(Gate) (Volts)" +
           TOP=1E-4  BOT=0  RIGHT=3.5  COLOR=2
PLOT.1D   IN.FILE=S4EX4DS2  Y.SCALE=@Scale  ^CL  ^AX  COLOR=3  LINE=2
LABEL      LABEL="Vbs=0"  X=1.9 Y=9E-5  RIGHT
LABEL      LABEL="Vbs=-2.5"  X=3.35 Y=7.6E-5  RIGHT
```

Figure 6-13 First part of input file *s4ex4d.inp*, showing the threshold voltage extraction

MOS Capacitance

Many process monitoring parameters—the grown oxide thickness, the surface state density, the flat band voltage, and the mobile charge in oxide, for example—are extracted from MOS C-V measurements during manufacturing. While transient device simulations are required for rigorous analysis of gate capacitance as a function of frequency, TSUPREM-4 can give reasonably accurate simulations of gate capacitance if a sufficiently high frequency (above 100 kHz) is assumed.

The parameter **MOSCAP** specifies that the MOS capacitance is extracted. Depending on the input frequency compared with the lifetime of carriers in the channel, there are three types of C-V plot.

1. In most cases, the input signal is composed of a slow bias signal and fast AC signal. This is the normal C-V plot, which is specified by **HIGH** (the default) in TSUPREM-4.
2. The second assumes a slow bias signal and a low-frequency AC input signal; that is specified by the **LOW** parameter.
3. Finally, if a fast bias sweep is performed with a high-frequency AC signal so that the inversion charge does not have time to accumulate, the depletion region expands deeper into the substrate. The **DEEP** parameter specifies this.

In this example, gate bias **V** is increased from -5 to 5 volts in 0.2-volt steps. The capacitance is calculated from *dQ/dV*. The perturbed AC bias, *dV* is calculated from the DC increment multiplied by a constant **E.RVCAP**, which is defined as the ratio of AC amplitude to DC increment in the **METHOD** statement (default value of 0.2 or 20%). The perturbed AC bias in this example is 0.04 V(= 0.2 V x 0.2). For example, the capacitance at 1 volt is extracted from the charge variation between 0.96 volts and 1.04 volts. The input statements for extracting the MOS capacitance are shown in Figure 6-14; the results are shown in Figure 6-16.

```
$ Part B: C-V plot for MOS capacitance
$ Extract the capacitance

$ -- High Frequency
ELECTRIC   X=0.0  MOSCAP  NMOS  V=”-5 5 0.2”  OUT.F=S4EX4DS3
$ -- Low  Frequency
ELECTRIC   X=0.0  MOSCAP  NMOS  V=”-5 5 0.2”  LOW  OUT.F=S4EX4DS4
$ -- Deep depletion
ELECTRIC   X=0.0  MOSCAP  NMOS  V=”-5 5 0.2”  DEEP

$ Plot the C-V curve
SELECT     TITLE=”MOS C-V”
VIEWPORT   X.MIN=0.5 Y.MIN=0.51
PLOT.1D    ELECTRIC  COLOR=2  TOP=1E-7  BOT=0  LEFT=-6  RIGHT=6 +
           X.OFF=1.5 ^CL
PLOT.1D    IN.FILE=S4EX4DS3  ^CL  ^AX  COLOR=3  LINE=2
PLOT.1D    IN.FILE=S4EX4DS4  ^CL  ^AX  COLOR=4  LINE=3
LABEL      LABEL=”Low”   X=3  Y=8.3E-8
LABEL      LABEL=”High”  X=3  Y=3.7E-8
LABEL      LABEL=”Deep”  X=3  Y=0.7E-8
```

Figure 6-14 Second part of input file *s4ex4d.inp*, showing the MOS capacitance extraction

Source/Drain Junction Capacitance

The junction capacitance between source (or drain) and bulk is one of important parameters to determine the delay characteristics of the MOS transistor. This capacitance is composed of two kinds of capacitance: areal and peripheral capacitance. In device characterization, these two components are separated and characterized by universal parameters independent of the shape of device. Usually, the areal capacitance is dominant. TSUPREM-4 extracts the areal junction capacitance (in Farads/cm^2) when the **JCAP** parameter is specified.

The **JUNCTION** parameter selects the junction to be analyzed. Junctions are numbered from the bottom of the structure to the top, with the deepest junction being junction number one. For example, if **X** specifies the emitter region of a bipolar transistor, there might be three junctions: E-B (emitter-base, junction number 3), B-C (base-collector, junction number 2), and C-S (collector-substrate, junction number 1). In this example, there is only one junction, identified as number 1.

The input bias **V** must be all positive or all negative including zero bias. TSUPREM-4 chooses the biased region by considering the polarity of the input voltage so that the junction is reverse biased. All regions except for the biased region are grounded. The perturbed AC bias is applied using the same method as in the MOS capacitance.

Input statements for extracting the junction capacitance are shown in Figure 6-15.

```
$ Part C: Junction Capacitance
$ Extract the S/D (area) junction capacitance
ELECTRIC   X=2.0  JCAP  JUNCTION=1  V="0 5 0.5"
SELECT     TITLE="S/D Junction C"
VIEWPORT   X.MIN=0.5 Y.MAX=0.49
PLOT.1D    ELECTRIC  COLOR=2  +
           TOP=2.5E-8  BOT=0  LEFT=-1  RIGHT=6  X.OFF=1.5 ^CL
```

Figure 6-15 Third part of input file *s4ex4d.inp*, showing the junction capacitance extraction

Plotting Results of Electrical Extraction

The results of the electrical extractions are shown in Figure 6-16. Note how the results of electrical extraction can be saved in an output file for later plotting.

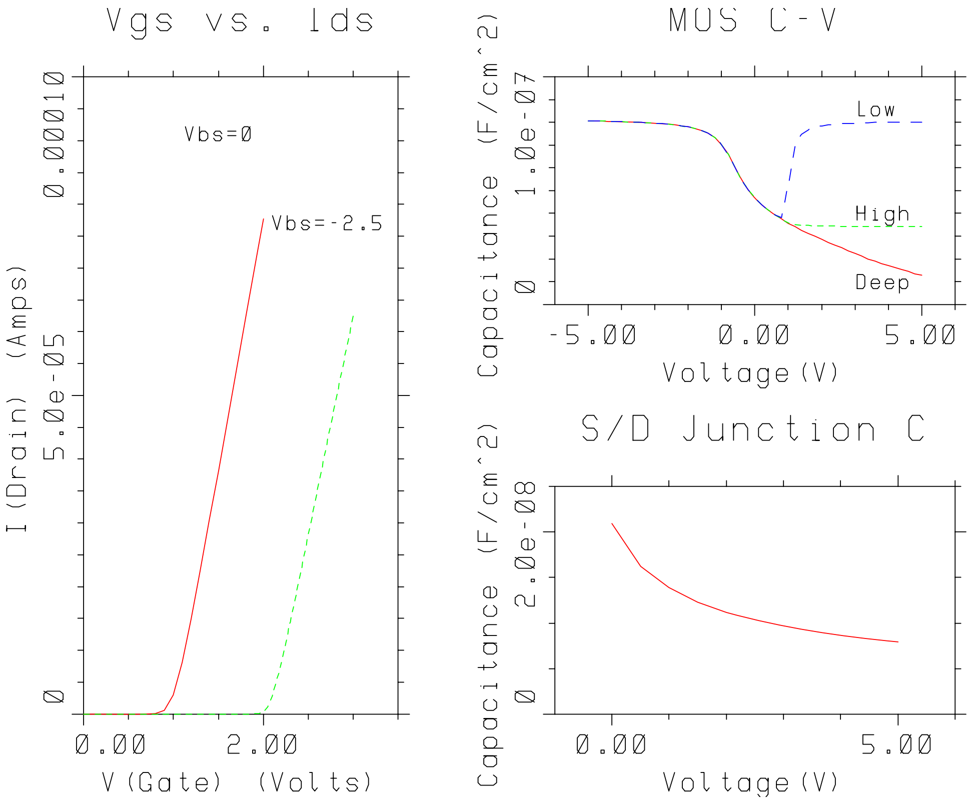


Figure 6-16 Electrical characteristics, plotted by input file *s4ex4d.inp*

Trench Implant Simulation

This example shows two methods for performing a tilted ion implantation into a trench. First, TSUPREM-4’s analytical model is used to implant ions at a 15-deg tilt. Next, TSUPREM-4’s Monte Carlo (MC) method is used to perform the same implantation. The results of the two methods are compared and contrasted.

```
$ TSUPREM-4 Example 5 - Implant Trench Application
$
$ Simulate ion implantation into a trench using the analytical approach
$ followed by the Monte Carlo approach.

$ Place the finest grid around the trench
LINE  X    LOCATION=0.0   SPACING=1.0
LINE  X    LOCATION=0.25  SPACING=0.01
LINE  X    LOCATION=0.32  SPACING=0.01
LINE  X    LOCATION=0.5   SPACING=0.2

LINE  Y    LOCATION=0.0   SPACING=0.01
LINE  Y    LOCATION=0.05  SPACING=0.01
LINE  Y    LOCATION=0.92  SPACING=0.04
LINE  Y    LOCATION=1.0   SPACING=0.01
LINE  Y    LOCATION=1.05  SPACING=0.01
LINE  Y    LOCATION=1.2   SPACING=1.0

ELIMINATE  ROWS     X.MAX=0.2   Y.MIN=0.15  Y.MAX=1.0
ELIMINATE  ROWS     X.MAX=0.25  Y.MIN=1.0   Y.MAX=1.1
ELIMINATE  COLUMNS  X.MIN=0.23  X.MAX=0.35  Y.MIN=1.1
ELIMINATE  COLUMNS  X.MIN=0.25  X.MAX=0.34  Y.MIN=1.1

$ Initialize the structure
INITIALIZE BORON=2E15

$ Etch the left half of the trench
ETCH       START     X=0.3   Y=0.0
ETCH       CONTINUE  X=0.3   Y=0.5
ETCH       CONTINUE  X=0.32  Y=0.9
ETCH       CONTINUE  X=0.34  Y=0.95
ETCH       CONTINUE  X=0.4   Y=1.0
ETCH       CONTINUE  X=2.0   Y=1.0
ETCH       DONE      X=2.0   Y=0.0

$ Form the complete trench structure
STRUCTURE  REFLECT

$ Save the structure
SAVEFILE   OUT.FILE=S4EX5S

$ Plot the grid
SELECT     TITLE=”Grid for Trench Application”
PLOT.2D    SCALE  GRID  C.GRID=2
```

Figure 6-17 First part of input file *s4ex5.inp*, showing grid setup

Structure Generation

The input statements for this simulation are shown in Figures 6-17, 6-19, and 6-23. The initial mesh setup uses the symmetry of the trench structure, as suggested by SEM photographs of trenches generated by reactive ion etching. The initial grid, established by the **LINE** and **ELIMINATE** statements, is for the left half of the trench only. Particular attention has been given to producing a fine grid where the side and bottom walls of the trench are formed in order to resolve the implanted profiles. The **ELIMINATE** statements are used to reduce the number of grid points in areas where the implant is not expected to penetrate. The sequence of **ETCH** statements specifies geometrically the shape of the left half of the trench structure. The resulting half-structure is reflected to generate the full structure, using the **REFLECT** parameter on the **STRUCTURE** statement. The full structure is saved with the **SAVEFILE** statement.

The grid for the full structure is then plotted (Figure 6-18). Note that a fine mesh has been placed around the trench, without wasting nodes in the lower corners.

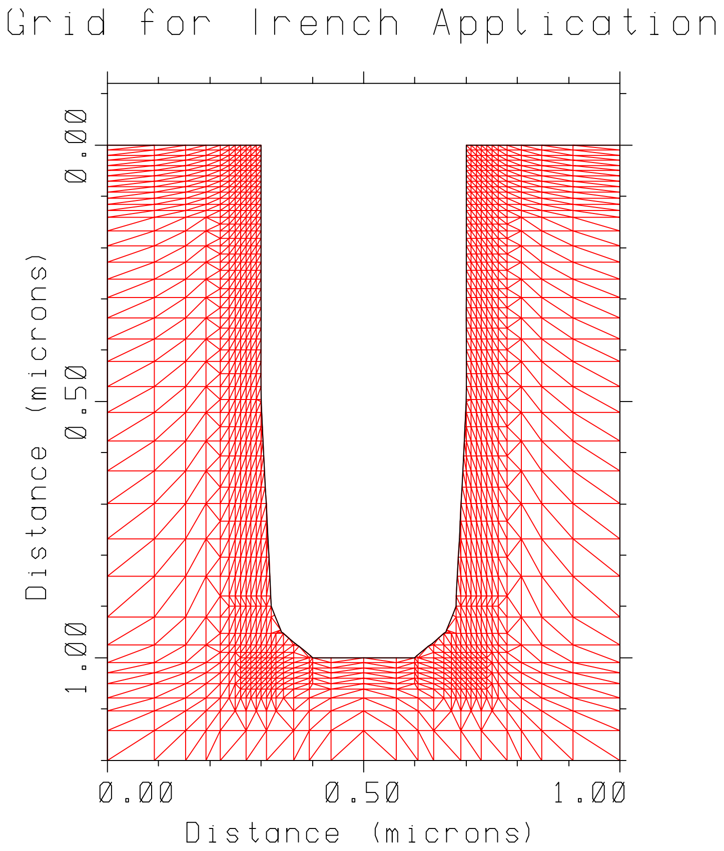


Figure 6-18 Grid for trench implant example

Analytic Implant

The boron implant is now performed at an energy of 5 keV. In the absence of the **MONTECAR** parameter, TSUPREM-4 uses the analytic method. The **TILT** parameter is used to specify that the angle of the incident ion beam is 15 deg counter-clockwise from the vertical (i.e., the beam enters from the left). The implant is followed by a short diffusion to activate the boron and anneal implant damage.

```
$ Analytic implant at a 15-degree tilt to dope the trench sidewalls
IMPLANT    BORON  ENERGY=5  DOSE=1E14  TILT=15

$ A short anneal
DIFFUSION  TIME=5  TEMP=900

$ 2D contour plot of boron contours
SELECT     Z=LOG10(BORON)  TITLE="Contours of Boron (Analytic)"
PLOT.2D    SCALE  X.MAX=1.5
FOREACH    X  (16 TO 21 STEP 1)
  COLOR      MIN.V=X  MAX.V=( X + 1)  COLOR=( X - 2)
END
PLOT.2D    ^AXES  ^CLEAR
LABEL      X=1.05  Y=0.25  LABEL="Log10(Boron)"  SIZE=0.3
LABEL      X=1.15  Y=0.33  LABEL="16-17" SIZE=0.3 C.RECT=14 W.R=0.4 H.R=0.4
LABEL      X=1.15  Y=0.41  LABEL="17-18" SIZE=0.3 C.RECT=15 W.R=0.4 H.R=0.4
LABEL      X=1.15  Y=0.49  LABEL="18-19" SIZE=0.3 C.RECT=16 W.R=0.4 H.R=0.4
LABEL      X=1.15  Y=0.57  LABEL="19-20" SIZE=0.3 C.RECT=17 W.R=0.4 H.R=0.4
LABEL      LABEL="Tilt angle=15 degrees"  X=0.02  Y=1.18

$ 1D plots of boron

$ Vertical profiles
SELECT     Z=LOG10(BORON)  TITLE="Vertical Profiles (Analytic)"
PLOT.1D    X.VALUE=0.1  TOP=21  BOTTOM=15  RIGHT=1.2  COLOR=2
LABEL      X=0.06  Y=19    LABEL="Surface"
LABEL      X=0.06  Y=18.8  LABEL="(x=0.1)"

PLOT.1D    X.VALUE=0.5  ^AXES  ^CLEAR  COLOR=2
LABEL      X=1.05  Y=18.5  LABEL="Trench"
LABEL      X=1.05  Y=18.3  LABEL="(x=0.5)"

$ Horizontal profiles
DEPOSIT    OXIDE  THICK=0.002
SELECT     Z=LOG10(BORON)  TITLE="Sidewall Profiles at y=0.2 (Analytic)"
PLOT.1D    Y.VALUE=0.2  LEFT=0.1  RIGHT=0.9  BOTTOM=15  TOP=21  COLOR=2

$ Print profile through bottom of trench
PRINT.1D   X.VALUE=0.5
```

Figure 6-19 Second part of input file *s4ex5.inp*, showing tilted implantation using analytic implant model

Plotting the Results of the Analytic Method

Figure 6-20 shows shaded contours of boron concentration produced by the **COLOR** statement and associated **SELECT**, **PLOT.2D**, **FOREACH**, **END**, and **LABEL** statements. Observe that dopant is present only on the right side of the trench where the ions are directly incident. This is because the analytic method does not account for reflected ions. The profile at the bottom right corner of the trench (where boron is directly incident) is approximately the same as that on the surface of the silicon.

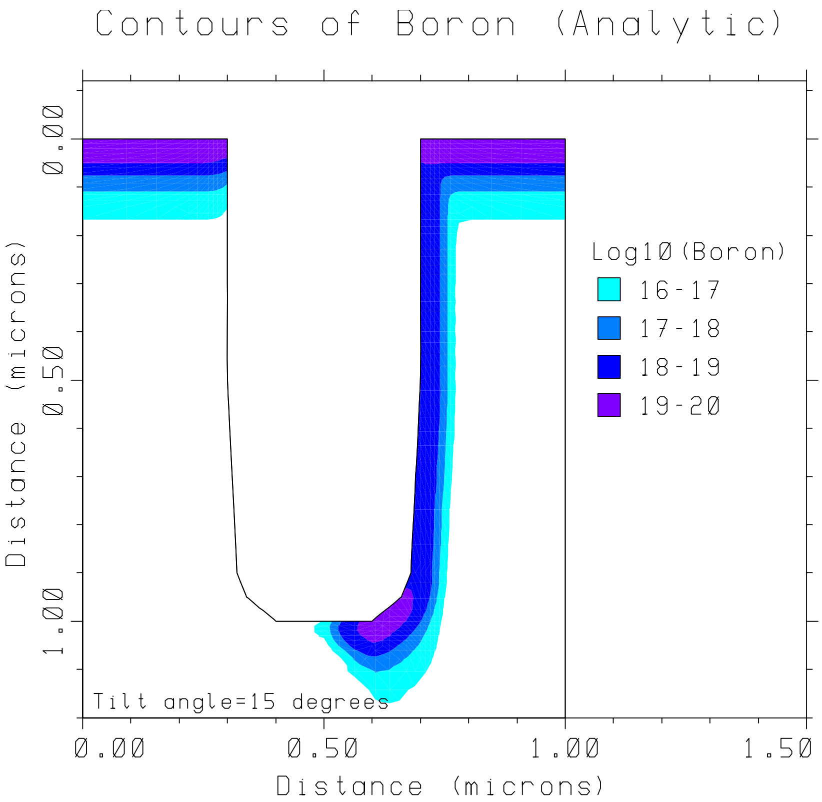


Figure 6-20 Contours of boron after analytic implant

Figure 6-21 shows 1-D boron profiles vertically through the middle of the trench bottom and at the surface of the silicon. The peak boron concentration is much greater at the silicon surface than at the bottom of the trench due to shadowing of the bottom by the sidewalls.

Figure 6-22 shows the boron concentration on the left and right sides of the trench at a depth of *y*=0.2 microns. The flat profile on the left hand side, at a value nearly that of the background concentration, confirms the lack of reflected ions when the analytic method is used. The corresponding plot for the MC implantation model is very different.

(A thin layer of oxide is deposited to ensure that the structure boundaries are correctly plotted by the **PLOT.1D** statement. Without the added oxide, the section at *y*=0.2 would pass through silicon, ambient, and silicon as *x* varies from 0 to 1.0 microns.

Due to a limitation in the current version of the program, the silicon/ambient and ambient/silicon interfaces are only recognized at the edges of the structure, but not in its interior. By adding the thin layer of oxide, silicon/oxide and oxide/silicon boundaries are added. These boundaries are plotted correctly in Figures 6-20 and 6-26).

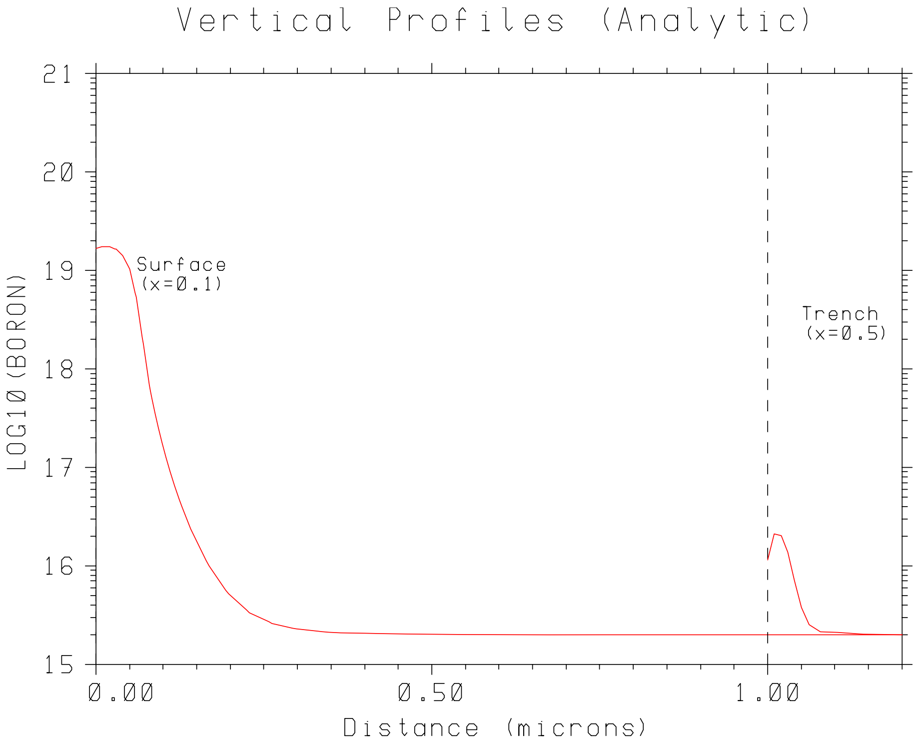


Figure 6-21 Vertical profiles produced by analytic implant

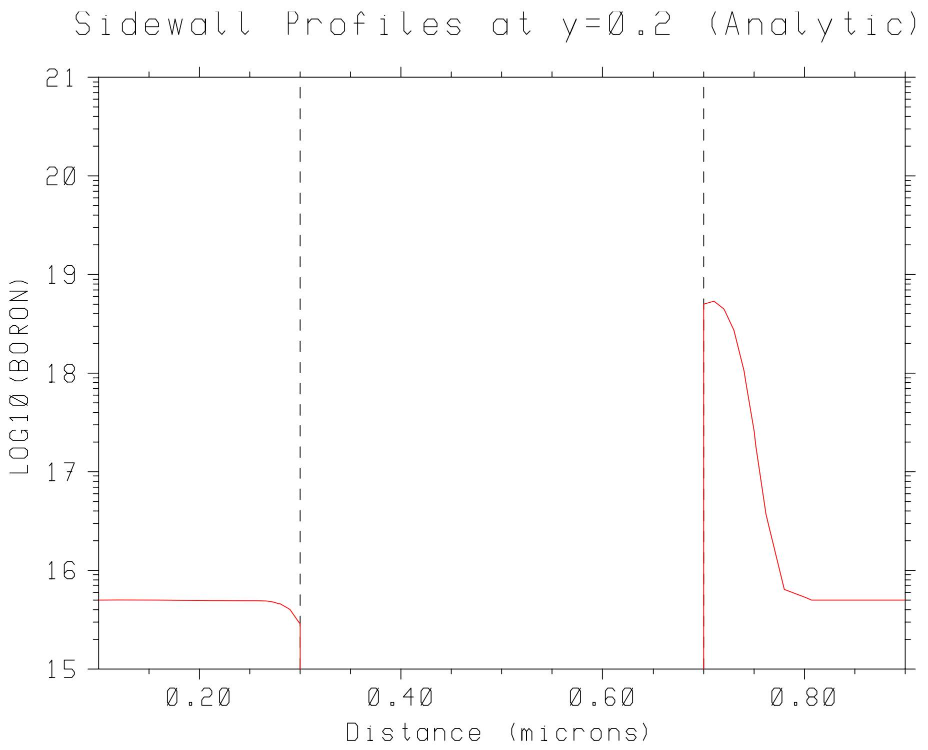


Figure 6-22 Sidewall profiles produced by analytic implant

MC Implant

The second half of the simulation demonstrates the use of the MC implantation method.

Overview

Unlike the analytic approach, the MC algorithm accurately models the effect of reflected ions in a trench structure. The MC calculation follows the trajectories of individual ions through each material present in the structure; ion trajectories through a vacuum, including those reflected off trench sidewalls, are calculated in exactly the same way as through other materials. The velocity of a reflected ion remains constant until it enters a material other than vacuum or leaves the top or bottom of the simulated structure. For ions that leave either side of the simulation space, the appropriate boundary condition (periodic, reflecting, or vacuum) is invoked.

Using the MC Model

The first step in the MC portion of the simulation is to read in the trench structure, just prior to the analytic implant. The boron implant is performed once again, but this time the **MONTECAR** parameter on the **IMPLANT** statement directs TSUPREM-4 to use the MC implant model. The **ENERGY**, **DOSE**, and **TILT** parameters are the same as those used during the analytic implant.

In addition to the **`MONTECAR`** parameter, two new parameters are now specified:

- **`N.ION`**—The **`N.ION`** parameter specifies the number of ion trajectories to calculate. The simulation time is directly proportional to the number of ion trajectories calculated. Decreasing this number, however, increases the statistical noise in the output of the MC calculation.
- **`^CRYSTAL`**—The specification of **`^CRYSTAL`** on the **`IMPLANT`** command disables the crystal model for ion penetration in silicon. This speeds the calculation significantly by eliminating the calculation of channeling and damage effects.

A short post-implant anneal is performed as before to activate the implant. The diffusion also acts to smooth the results of the MC calculation.

Plotting the Results of the MC Method

The three figures output after the MC implant (Figures 6-24 through 6-26) are the counterparts of the three analytic plots (Figures 6-20 through 6-22).

Boron Contours

Figure 6-24 shows the contours of boron concentration produced by the MC method. *Note* that dopant is present in substantial quantities on the left side of the trench (unlike the analytic results). The plot illustrates that dopant is present in greater quantity on the right side of the trench where the ions are directly incident. The boron in the left side of the trench is due entirely to reflection of ions. As before, the profile at the bottom right corner of the trench is approximately the same as that on the surface of the silicon.

```
$ Repeat the implantation using the Monte Carlo method

$ Read in the structure
LOADFILE   IN.FILE=S4EX5S

$ Monte Carlo implant at a 15-degree tilt to dope the trench sidewalls
IMPLANT    BORON  ENERGY=5  DOSE=1E14  MONTECAR  TILT=15  N.ION=10000
^CRYSTAL

$ A short anneal
DIFFUSION  TIME=5  TEMP=900

$ 2D contour plot of boron contours
SELECT     Z=LOG10(BORON)  TITLE="Contours of Boron (Monte Carlo)"
PLOT.2D    SCALE  X.MAX=1.5
FOREACH    X  (16 TO 21 STEP 1)
  COLOR      MIN.V=X  MAX.V=( X + 1)  COLOR=( X - 2)
END
PLOT.2D    ^AXES  ^CLEAR
LABEL      X=1.05  Y=0.25  LABEL="Log10(Boron)"  SIZE=0.3
LABEL      X=1.15  Y=0.33  LABEL="16-17" SIZE=0.3 C.RECT=14 W.R=0.4 H.R=0.4
LABEL      X=1.15  Y=0.41  LABEL="17-18" SIZE=0.3 C.RECT=15 W.R=0.4 H.R=0.4
LABEL      X=1.15  Y=0.49  LABEL="18-19" SIZE=0.3 C.RECT=16 W.R=0.4 H.R=0.4
LABEL      X=1.15  Y=0.57  LABEL="19-20" SIZE=0.3 C.RECT=17 W.R=0.4 H.R=0.4
LABEL      LABEL="Tilt angle=15 degrees"  X=0.02  Y=1.18

$ 1D plots of boron

$ Vertical profiles
SELECT     Z=LOG10(BORON)  TITLE="Vertical Profiles (Monte Carlo)"
PLOT.1D    X.VALUE=0.1  TOP=21  BOTTOM=15  RIGHT=1.2  COLOR=2
LABEL      X=0.06  Y=19    LABEL="Surface"
LABEL      X=0.06  Y=18.8  LABEL="(x=0.1)"

PLOT.1D    X.VALUE=0.5  ^AXES  ^CLEAR  COLOR=2
LABEL      X=1.05  Y=18.5  LABEL="Trench"
LABEL      X=1.05  Y=18.3  LABEL="(x=0.5)"

$ Horizontal profiles
DEPOSIT    OXIDE  THICK=0.002
SELECT     Z=LOG10(BORON)  TITLE="Sidewall Profiles at y=0.2 (Monte Carlo)"
PLOT.1D    Y.VALUE=0.2  LEFT=0.1  RIGHT=0.9  BOTTOM=15  TOP=21  COLOR=2

$ Print profile through bottom of trench
PRINT.1D   X.VALUE=0.5
```

Figure 6-23 Third part of file *s4ex5.inp,* using the MC implantation model

Vertical Profiles

Figure 6-25, shows 1-D boron profiles vertically through the middle of the trench bottom and at the surface of the silicon for the MC implant. The peak boron concentration is approximately a factor of five greater at the silicon surface than at the bottom of the trench due to shadowing of the bottom by the sidewalls. Observe that the difference between the peaks is much less for the MC approach than the analytic method. This is due to the reflected ion calculations included in the MC algorithm. The depth of the implant is less at the bottom of the trench than at the silicon surface, due to the loss of energy when an ion is reflected.

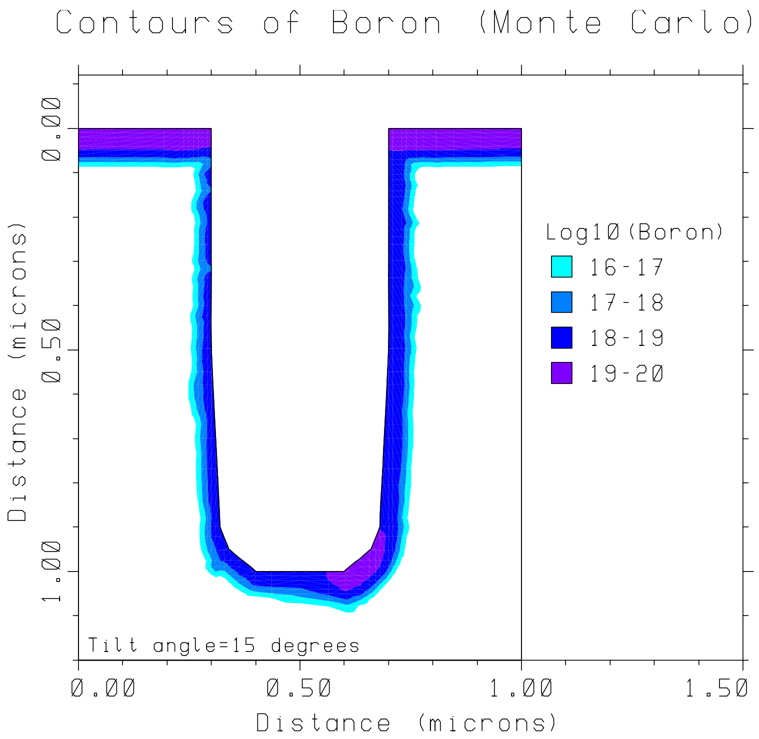


Figure 6-24 Contours of boron after Monte Carlo implant

Comparing the vertical profiles produced by the analytic and MC models (Figures 6-21 and 6-25, respectively), shows that the boron penetrates more deeply with the analytic model than with the MC approach. This is due to the specification of **^CRYSTAL** during the MC implantation. This causes the silicon to be treated as amorphous, which speeds the calculation significantly but prevents calculation of channeling in the crystal structure.

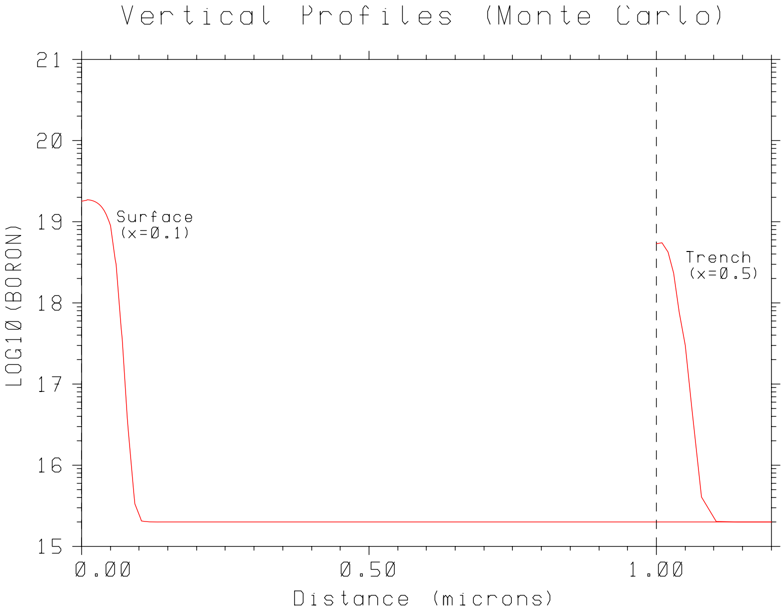


Figure 6-25 Vertical profiles after MC implant

In contrast, the analytic method automatically includes the channeling calculations, thus resulting in deeper penetration. If the **^CRYSTAL** were omitted, channeling calculations would be included in the MC implant, giving a depth of penetration very similar to the analytic results.

Sidewall Profiles

The last MC plot (Figure 6-26) shows the boron profiles on the left and right sidewalls of the trench. The profile on the left side of the trench is due entirely to reflected ions and is not as deep as the profile on the right, due to the reduced energy of reflected ions. The peak concentration of the profile on the left side is also reduced compared to that on the right. This result incorporates effects of ions reflecting from various points in the trench as well as multiple reflections.

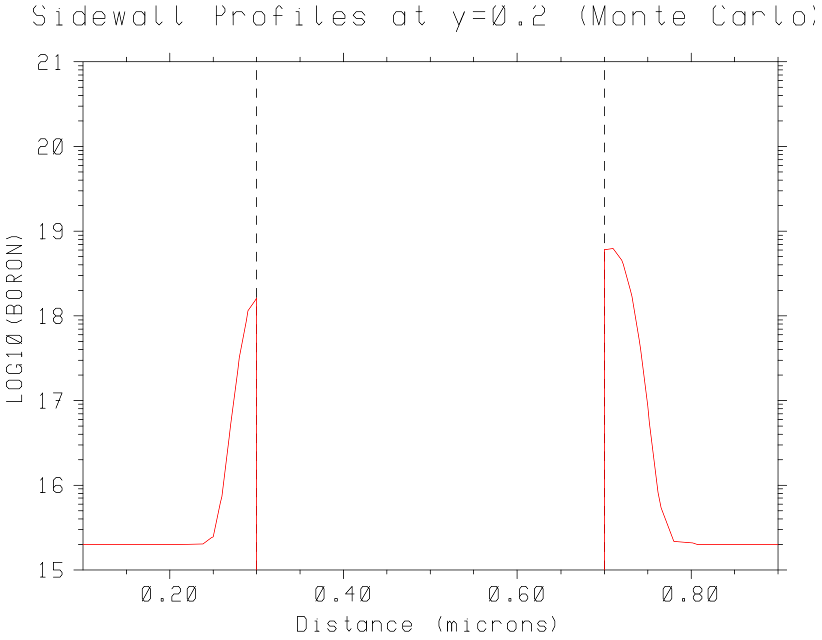


Figure 6-26 Sidewall profiles after MC implant

Summary

The MC model can be very useful in simulating implantation into complex structures, due to its ability to model the effects of reflected ions. It is also useful in cases where analytical range statistics are not available, and in some multilayer implant simulations. The analytic model is more suitable, however, for simulating the implant steps that are encountered in most practical situations.

Poly-Buffered LOCOS

This example illustrates the use of TSUPREM-4 to simulate a poly-buffered LOCOS process. This process reduces the typical “bird’s beak” formation and stresses on the silicon substrate through use of a layer of polycrystalline silicon between the pad oxide and the nitride mask.

Structure Generation

The input statements for generating the structure and performing the field isolation processing are shown in Figure 6-27. A relatively coarse initial grid is used. In the horizontal direction, it is finest at the mask edge at *x*=1 micron. The pad oxide and poly buffer layers are deposited with two grid spaces in each. Between the poly layer and the nitride mask, a thin (20Å) layer of oxide is deposited. A thick nitride mask layer is used. The nitride mask is etched to the right of *x*=1.0 to expose the region to be oxidized. This example oxidizes through the polysilicon layer. The grid prior to oxidation is shown in Figure 6-28.

```
$ TSUPREM-4:  Poly-Buffered LOCOS Application

$ Set up the grid
LINE  X     LOCATION=0      SPACING=0.1
LINE  X     LOCATION=1.0    SPACING=0.06
LINE  X     LOCATION=1.5    SPACING=0.1

LINE  Y     LOCATION=0      SPACING=0.03
LINE  Y     LOCATION=0.5    SPACING=0.1

INITIALIZE

$ Deposit pad oxide and define nitride mask
DEPOSITION  OXIDE    THICK=0.02   SPACES=2
DEPOSITION  POLY     THICK=0.04   SPACES=2
DEPOSITION  OXIDE    THICK=0.002
DEPOSITION  NITRIDE  THICK=0.2    SPACES=2

ETCH        NITRIDE  RIGHT  P1.X=1.0

$ Plot initial grid
SELECT      TITLE="Initial Grid"
PLOT.2D     SCALE  GRID  C.GRID=2

$ Use viscoelastic model for oxidation
METHOD      VISCOEL
AMBIENT MAT=OXIDE   VC=170
AMBIENT MAT=NITRIDE VC=100
DIFFUSION   TEMP=1050  WETO2  TIME=85

$ Save structure
SAVEFILE    OUT.FILE=S4EX6S
```

Figure 6-27 First part of input file *s4ex6.inp*: Poly-buffered LOCOS process

Using the VISCOEL Model

The **METHOD** statement selects the viscoelastic oxidation model. The stress dependence of the physical parameters of the model are included in the calculation by default. This combination gives results that are much more accurate than those obtained with the **COMPRESS** model, but takes much less simulation time than the **VISCOUS** model with stress dependence.

The stress dependence of the oxide and nitride viscosities is given by the **VC** parameter on the **AMBIENT** statement. Because these values depend on temperature, it is usually necessary to specify appropriate values for each oxidation step. The two **AMBIENT** statements in this example set **VC** for nitride and oxide to values given in the literature for a 1050°C oxidation. (Appendix A lists the default values for **VC** and suggests values for use at various temperatures.)

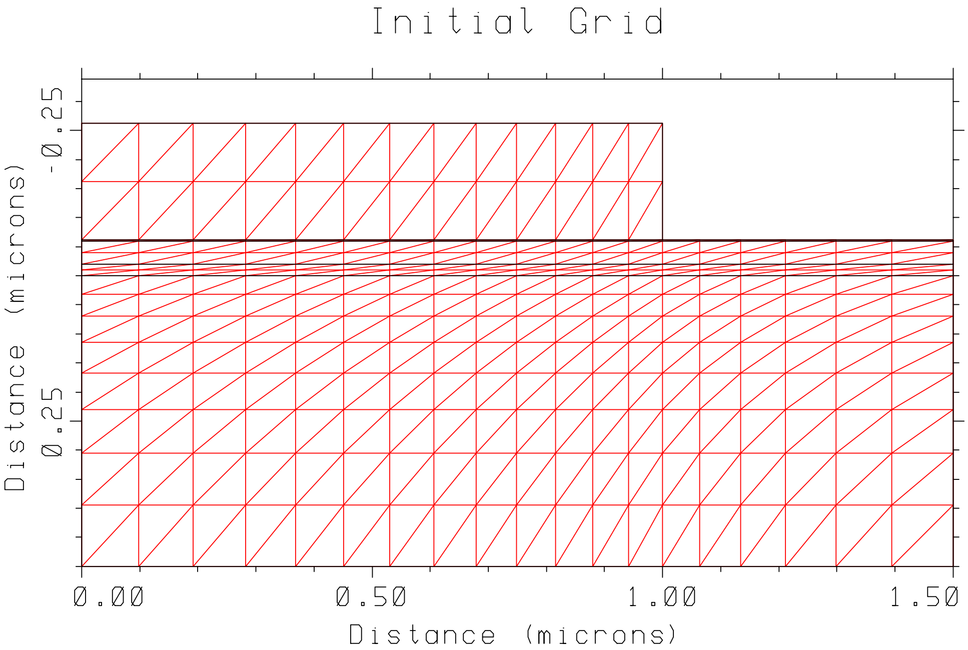


Figure 6-28 Grid for poly-buffered LOCOS application

Plotting the Results

Figure 6-29 shows the input statements used to plot the results. Of particular interest are the stresses in the structure, as indicated by the hydrostatic pressure. The statement **`SELECT Z=(-0.5*(SXX+SYY))`** calculates the hydrostatic pressure from the stress components. The minus sign is used so that positive values of **`Z`** correspond to compression. A **`FOREACH`** loop is used to plot contours of compressive and tensile stress at values from $3x10^9$ to $3x10^{10}$, in steps of $3x10^9$.

The final plot is shown in Figure 6-30. The very high pressure gradients at the end of the poly layer (near *x*=0, *y*=-0.05) result from the volume expansion produced by the oxidation of the poly layer. The high pressure in this region confirms that stress effects are important in this simulation, and that the stress-dependent oxidation models must be used to obtain an accurate simulation of this structure.

```
$ Plot the structure with oxide stress
SELECT      TITLE="Contours of Pressure"
PLOT.2D     SCALE
COLOR       SILICON  COLOR=7
COLOR       OXIDE    COLOR=5
COLOR       NITRIDE  COLOR=4
COLOR       POLY     COLOR=3
SELECT      Z=(-0.5*(SXX+SYY))
FOREACH     I (1 TO 10)
  CONTOUR     VALUE=(  I *3E9)  LINE=2  COLOR=6
  CONTOUR     VALUE=(- I *3E9)  LINE=3  COLOR=6
END
PLOT.2D     ^AX  ^CL

LABEL    LABEL="Poly"       X=0.1  Y=-0.03
LABEL      LABEL="Oxide"      X=1.2  Y=-0.05
LABEL    LABEL="Nitride"    X=0.1  Y=-0.15
LABEL    LABEL="Substrate"  X=0.1  Y=0.20

$ Print location of Si/SiO2 interface
SELECT      Z=Y
PRINT.1D    SILICON  /OXIDE
```

Figure 6-29 Second part of *s4ex6.inp:* Plotting final poly-buffered LOCOS structure

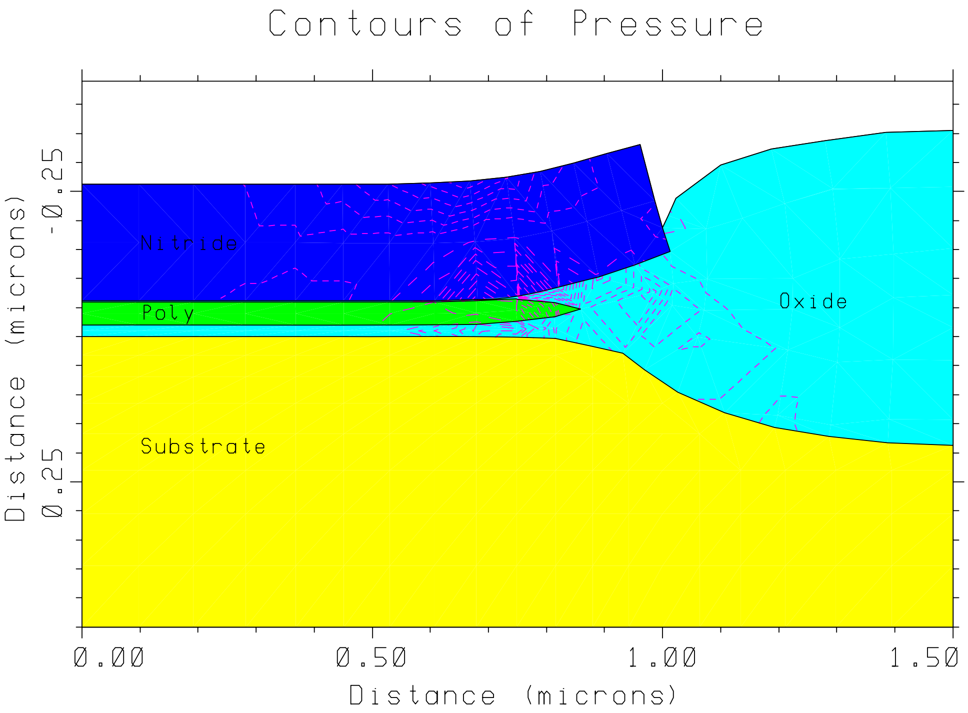


Figure 6-30 Contours of hydrostatic pressure in final poly-buffered LOCOS structure

CMOS Process

This example simulates the fabrication of NMOS transistors of two different channel lengths as part of a CMOS process. The goal is to produce an output file containing complete structure, mesh, and doping information that can be read into the Medici device simulator. The considerations needed to produce such a file are illustrated.

The two channel lengths are simulated with a single input file, using a **FOREACH** loop around the entire simulation, from mesh generation to saving of the final structure. The use of the **FOREACH** loop simplifies the input and ensures that any changes in the process sequence are applied to both devices equally. This use of the **FOREACH** loop illustrates several useful techniques for parameterizing TSUPREM-4 input files. The listing of the first input file for this example (*s4ex7a.inp*) is shown in Figures 6-31 through 6-33. Comment statements have been used to number the steps in the simulation.

```
$ TSUPREM-4 N-channel MOS application
$
$ 1. Identify the graphics driver
$ Default from DEFPDEV, TERM, or S4PCAP ”default” entry used

$ 2. Beginning of the main loop
FOREACH    LD  ( 3 5 )

$ 3. Specify the mesh
MESH       GRID.FAC=1.5
MESH       DY.SURF=0.01  LY.SURF=0.04  LY.ACTIV=2.0

$ 4. Initialize
INITIALIZE <100>  BORON=1E15  WIDTH=( 0.7 + ( LD / 10.0 ) ) DX=0.1

$ 5. Plot the initial mesh
SELECT     TITLE=”Mesh for Delta=0.@{LD}”
PLOT.2D    SCALE  GRID  Y.MAX=3.0  C.GRID=2

$ 6. Initial oxide pad
DEPOSIT    OXIDE  THICKNESS=0.03

$ 7. P-well implant
IMPLANT    BORON  DOSE=1E12  ENERGY=35

$ 8. Use a point defect models that simulates OED
METHOD     PD.TRANS

$ 9. P-well drive
DIFFUSE    TEMP=1100  TIME=120  DRYO2  PRESS=0.02

$ 10. P-well doping profile
SELECT     Z=LOG10(BORON)  TITLE=”Channel Doping (Delta=0.@{LD})”
PLOT.1D    X.VALUE=0  RIGHT=3.0  BOTTOM=15  TOP=19    LINE.TYP=2  COLOR=2
LABEL      X=1.8  Y=18.5  LABEL=”After p-well drive”  LINE.TYP=2  C.LINE=2
```

Figure 6-31 First part of input file *s4ex7a.inp*, to set up grid for simulating a CMOS process

```
$ 11. Pad nitride
DEPOSIT    NITRIDE  THICKNESS=0.1

$ 12. Field implant and oxidation
IMPLANT    BORON  DOSE=5E13  ENERGY=80
DIFFUSE    TEMP=1000  TIME=360  WETO2

$ 13. Etch to remove the pad
ETCH       NITRIDE  ALL

$ 14. Vt adjust implant
IMPLANT    BORON  ENERGY=100  DOSE=1E12

$ 15. P-well doping profile
SELECT     Z=LOG10(BORON)
PLOT.1D    X.VALUE=0  ^AXES  ^CLEAR  COLOR=2
LABEL      X=1.8  Y=18.2  LABEL=”After Vt implant”  LINE.TYP=1 C.LINE=2

$ 16. Print oxide and silicon thicknesses
SELECT     Z=1
PRINT.1D   X.VALUE=0.0  LAYERS

$ 17. Etch oxide
ETCH       OXIDE  TRAP  THICK=0.05

$ 18. Gate oxidation
DIFFUSE    TEMP=950  TIME=30  DRYO2

$ 19. Poly deposition
DEPOSIT    POLYSILICON  THICKNESS=0.3  DIVISIONS=4

$ 20. Poly and oxide etch between x = 0.0 and 0.5 microns
ETCH       POLY   LEFT  P1.X=0.5
ETCH       OXIDE  TRAP  THICK=0.04

$ 21. Deposit a thin layer of oxide
DEPOSIT    OXIDE  THICKNESS=0.02

$ 22. LDD implant
IMPLANT    PHOS  ENERGY=50  DOSE=5E13  IMPL.TAB=PHOSPHORUS

$ 23. LTO
DEPOSIT    OXIDE  THICK=0.2

$ 24. Establish a sidewall spacer
ETCH       OXIDE  TRAP  THICK=0.22

$ 25. Source/drain implant
IMPLANT    ARSENIC  ENERGY=100  DOSE=2E15

$ 26. Oxide etch
ETCH       OXIDE  LEFT  P1.X=0.5

$ 27. Use an oxidation model that understands polysilicon
METHOD     COMPRESS

$ 28. Source/drain reoxidation (including the polysilicon gate)
DIFFUSE    TEMP=900  TIME=30  DRYO2
```

Figure 6-32 Second part of input file *s4ex7a.inp*, showing statements for simulating a CMOS process

```
$ 29. BPSG -- etch to open windows for aluminum contact
DEPOSIT    OXIDE  THICK=0.3
ETCH       OXIDE  LEFT  P1.X=0.3

$ 30. Metallization -- etch to create a source contact
DEPOSIT    ALUMINUM  THICK=0.5  SPACES=3
DEPOSIT    PHOTORESIST THICK=1.0
ETCH       PHOTORESIST RIGHT  P1.X=0.6
ETCH       ALUMINUM  TRAP  ANGLE=85  THICK=0.8
ETCH       PHOTORESIST ALL

$ 31. Reflect to form the complete structure; then save it
SAVEFILE   OUT.FILE=S4EX7AS@LD
STRUCTURE  REFLECT  RIGHT
SAVEFILE   OUT.FILE=S4EX7AP@LD  MEDICI

$ 32. End of loop
END
```

Figure 6-33 Third part of input file *s4ex7a.inp*, for simulating a CMOS process

Main Loop

The main loop begins at step 2 and ends at step 32. The name *LD* is used as a looping parameter, taking on the values 3 and 5. These values represent an increment in channel length to be added to each half of the symmetrical structure. *LD* is referenced in step 4, where it results in the width of the simulated structure being set to 0.7 + 3/10 = 1.0 microns for *LD* = 3 and 0.7 + 5/10 = 1.2 microns for *LD* = 5. Because the device is symmetrical only the left half of the structure needs to be included in the simulation; the structure is reflected about its right edge to form the full structure in step 31.

The entire structure has a width of 1.0 × 2 = 2 microns for the first pass through the loop (*LD* = 3) and a width of 1.2 × 2 = 2.4 microns for the second pass (*LD* = 5). The final channel lengths are approximately 0.8 and 1.2 microns.

Mesh Generation

In generating the mesh for this example, keep in mind that the same mesh is used for the Medici simulation as for the TSUPREM-4 simulation. In particular, it must taken into account that in the vertical direction, the field-dependent mobility models are calibrated for a grid spacing of about 100 Angstroms.

Automatic grid generation with adaptive grid is used. A **MESH** statement is used to modify the default parameters: The spacing at the surface (**DY.SURF**) is set to 0.01 for Medici, and the locations of the bottoms of the surface and active regions (**LY.SURF** and **LY.ACTIV**, respectively) are reduced to 0.04 and 2 microns (from the defaults of 0.1 and 4 microns) to reduce the simulation time. Grid spacings are increased by 50% for this simulation, using **GRID.FAC**=1.5, but this should be changed to 1.0 to get a 100 Angstrom spacing at the surface before doing the final Medici simulations. The default mesh extends to a depth of 200 microns. The structure can be truncated (with **STRUCTURE TRUNCATE**) before saving it for Medici, or Medici can truncate the saved structure as it is read in.

Note:

> ***A separate mesh in** Medici **could be generated and the doping profiles read without the mesh from the TSUPREM-4 output file. In that case, the generation of the mesh would be guided more by process simulation considerations and less by the needs of the device simulation.***

It is often a good idea to check the initial mesh structure (as in step 5) before starting a TSUPREM-4 simulation. The **PLOT.2D** statement in step 5 specifies **Y.MAX** = 3 microns, since the area of interest does not extend much below the deepest junction depth. Given the times and temperatures of the processing steps to be used, do not expect the junction depth to exceed 1 micron.

The plot of the mesh produced the first time through the loop (*LD* = 3) is shown in Figure 6-34. The grid appears coarse, but is refined as needed by adaptive gridding.

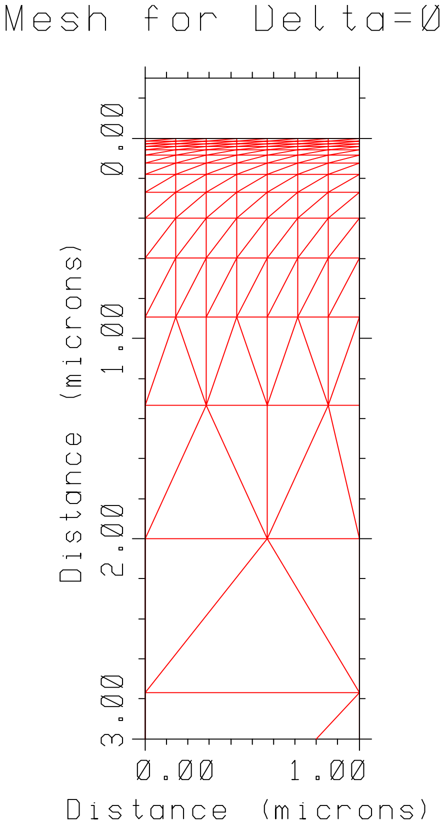


Figure 6-34 Initial grid for the 0.8 micron NMOS transistor, produced by *s4ex7a.inp*

The **FOREACH** parameter *LD* is used in step 5 in the title for the mesh plot. The name *LD* is enclosed in braces to separate it from the rest of the character string and preceded by a “@” character to force substitution of the value defined on the

FOREACH statement. The resulting titles are “Mesh for Delta=0.3” and “Mesh for Delta=0.5” for the two passes through the main loop.

CMOS Processing

The simulation of the CMOS process is straightforward. A variety of models are used to simulate the various processes.

Models

Because the process is planar up to the patterning of the polysilicon gate, the **VERTICAL** oxidation model can be used without any loss of accuracy. Once the poly gate is patterned, a switch is made to the **COMPRESS** model, both because of the nonplanarity of the structure and the desire to simulate the oxidation of the polysilicon gate. In order to accurately simulate oxidation-enhanced diffusion effects, the **PD.TRANS** model for point defects is chosen.

Note:

> ***There might be significant OED effects produced by the field oxidation; to accurately simulate these, the structure would need to be expanded to include the field region.***

The advanced models for grain growth and dopant diffusion in polysilicon are used by default. For a faster simulation but with less accuracy in the gate region, advanced poly models could be disabled before depositing the polysilicon layer.

Channel Doping Plot

The channel doping profile is plotted at two points in the process—after the p-well drive and after the threshold adjust implant. The result is shown in Figure 6-35. Again, *LD* has been used in the title of the plot.

Lightly Doped Drain Structure

The remaining steps in the loop are fairly standard, but the method for producing the lightly-doped drain structure should be pointed out. The light source/drain extension implant is performed in step 22; because the implantation is through an oxide, **IMPL.TAB**=*phosphorus* is used to access implant data tables that do not include channelling effects. This is followed with a conformal deposition of 0.2 microns of oxide. Step 24 uses the **TRAP** parameter on the **ETCH** statement to remove all oxide within a vertical distance of 0.22 microns from the surface. This produces a spacer on the sidewall of the poly gate that serves as an implant mask for the heavy source/drain implant of Step 25.

Contacts

An important final step for structures to be used in Medici is the deposition and patterning of aluminum contacts. Interfaces between aluminum and silicon are converted to contacts when the structure is passed to Medici. In addition, the Medici user can specify whether polysilicon regions are to be treated as semiconductor regions or contacts, and whether a contact should be added along the back side of the structure.

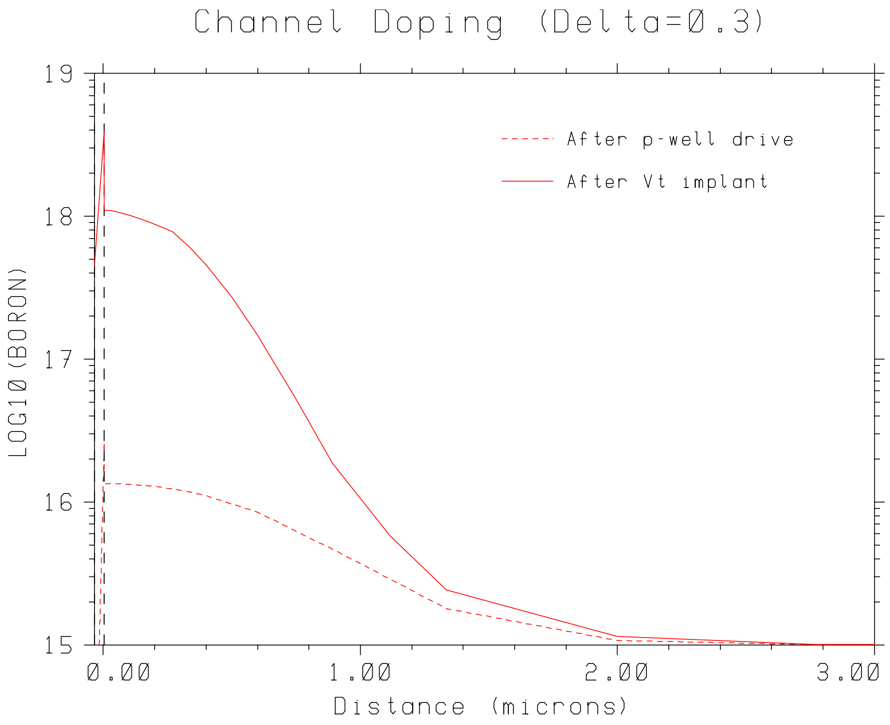


Figure 6-35 Channel doping profile for NMOS transistor

Saving the Structure

Once processing is complete, the finished structure is saved. Each structure is saved twice, once in TSUPREM-4's file format and once in the Medici format. For the Medici file, the structure must first be reflected to produce a full transistor. The structure is saved in TSUPREM-4 format before reflection. *Note* that the **FOREACH** parameter *LD* has been used once again to produce the output file names. This time the "@" character has been used to force substitution of the parameter, but the braces are not needed.

End of Main Loop

The **END** statement signals the end of the main **FOREACH** loop. After the entire sequence of input statements has been processed for *LD*=3, it is repeated for *LD*=5.

Plotting the Results

Once the complete structure has been saved, it can be read back in using the **INITIALIZE** statement and the results plotted without rerunning the simulation. Figure 6-36 shows an input file (*s4ex7b.inp*) that reads file *S4EX7AS3*, produced by the first **SAVEFILE** statement of step 31 on the first pass through the main loop.

```
$ TSUPREM-4 N-channel MOS application -- Part B
$ Plot the results for delta = 0.3

$ Read the structure
INITIALIZE IN.FILE=S4EX7AS3
STRUCTURE  REFLECT RIGHT

$ Prepare to plot contours of boron, phosphorus, and arsenic
SELECT     TITLE=”N-Channel (Delta=0.3)”
PLOT.2D    SCALE  Y.MAX=1.5
COLOR      SILICON  COLOR=7
COLOR      OXIDE    COLOR=5
COLOR      POLY     COLOR=3
COLOR      ALUMI    COLOR=2

SELECT     Z=LOG10(Boron)
FOREACH    VAL (14 TO 21 STEP 1)
  CONTOUR    VALUE=VAL  LINE=5  COLOR=2
END

SELECT     Z=LOG10(Phosphorus)
FOREACH    VAL (16 TO 21 STEP 1)
  CONTOUR    VALUE=VAL  LINE=4  COLOR=3
END

SELECT     Z=LOG10(Arsenic)
FOREACH    VAL (16 TO 21 STEP 1)
  CONTOUR    VALUE=VAL  LINE=2  COLOR=4
END

PLOT.2D    ^AX  ^CL

$ Add labels
LABEL      X=0.01  Y=-0.8    LABEL=”Aluminum”
LABEL      X=1.99  Y=-0.8    LABEL=”Aluminum”  RIGHT
LABEL      X=1.0   Y=-0.45   LABEL=”BPSG”      CENTER
LABEL      X=1.0   Y=-0.2    LABEL=”Poly”      CENTER
LABEL      X=0.05  Y=0.35    LABEL=”Source”
LABEL      X=1.95  Y=0.35    LABEL=”Drain”     RIGHT

$ Plot the grid
SELECT     TITLE=”Final Mesh (Delta=0.3)”
PLOT.2D    GRID  SCALE  Y.MAX=3  C.GRID=2

$ Plot arsenic profile in poly
SELECT    Z=Log10(Active(Arsenic))  TITLE=”Arsenic Concentration in Gate”
PLOT.1D   X.V=1  LEFT=-0.4 RIGHT=0 BOT=16 TOP=21  COLOR=4
SELECT    Z=Log10(Arsenic)
PLOT.1D   X.V=1  ^AX ^CL  COLOR=4 LINE=2
LABEL     X=-0.16 RIGHT Y=19   SIZE=0.28 COLOR=4 LABEL=”Active”
LABEL     X=-0.15 LEFT  Y=19.7 SIZE=0.28 COLOR=4 LABEL=”Total”
```

Figure 6-36 Input file *s4ex7b.inp*, for plotting results

0.8 Micron Device

In order to plot results for the full device, the structure is first reflected about its right edge. The structure is then plotted, with contours of boron, arsenic, and phosphorus concentration. The result is shown in Figure 6-37. The phosphorus LDD regions can be clearly seen, extending from the heavily-doped arsenic source/drain regions to the edge of the poly gate.

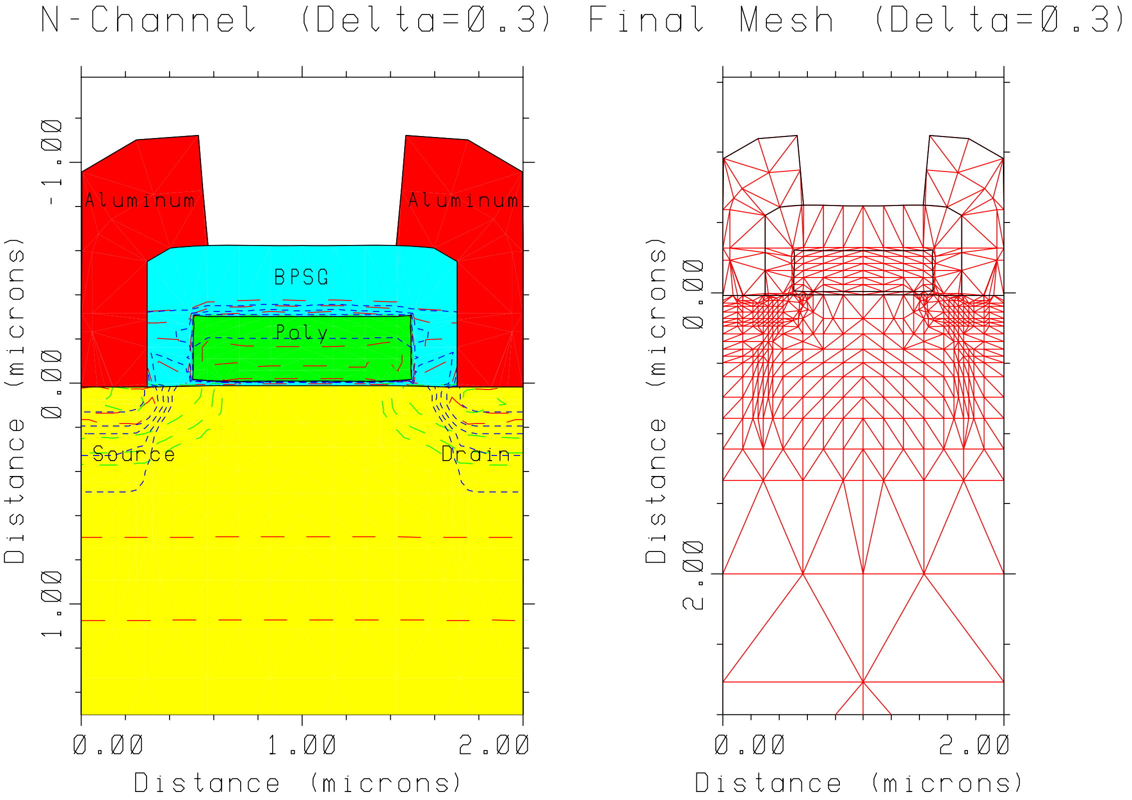


Figure 6-37 Final 0.8 micron structure, NMOS structure, plotted by *s4ex7b.inp* (left*) and* final mesh for 0.8 micron NMOS structure (right)

Labels have been added to facilitate interpretation of the plot. The labels near the center of the plot have been automatically centered (with the **CENTER** parameter on the **LABEL** statement), and the labels on the right have been right justified (with the **RIGHT** parameter).

Final Mesh

The final mesh is shown in Figure 6-37 (right). Observe the fine grid in the channel, as desired, and a fine grid around the source and drain junctions. Medici can refine the grid or use its own grid, if needed, provided that the doping profiles are accurately represented by the TSUPREM-4 grid.

Arsenic Profiles in Gate

Figure 6-38 shows the arsenic profiles through the poly gate. The active concentration includes only the electrically active arsenic in the interiors of the polycrystalline grains. The total concentration also includes any clustered arsenic in the grain interiors and the arsenic atoms that occupy sites on the grain boundaries.

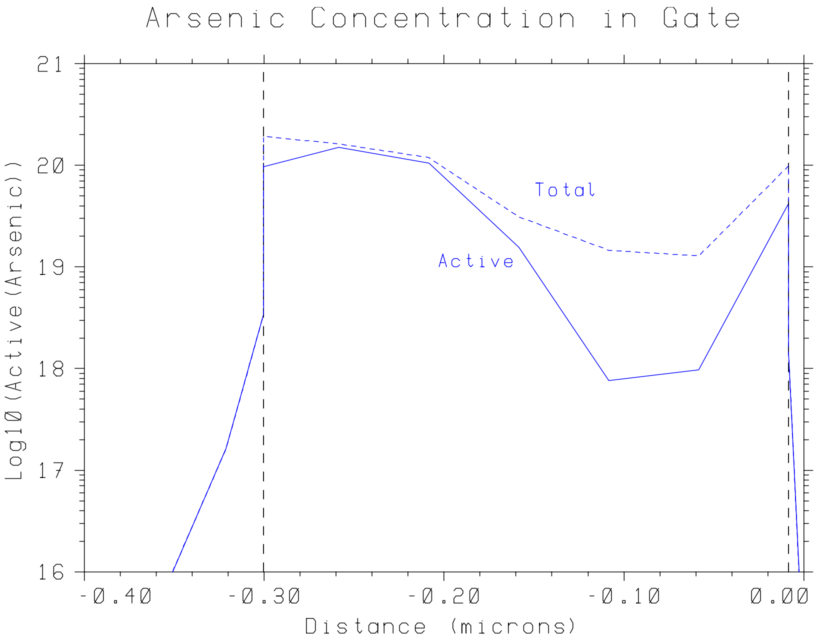


Figure 6-38 Profiles of active and total arsenic concentration through the poly gate

1.2 Micron Device

A third input file (*s4ex7c.inp*, not shown) can be used to plot the results saved by the second pass through the main loop, in the same way that *s4ex7b.inp* plotted the results of the first pass. Figure 6-39 shows the structure plot of the 1.2 micron NMOS transistor, as plotted by input file *s4ex7c.inp*.

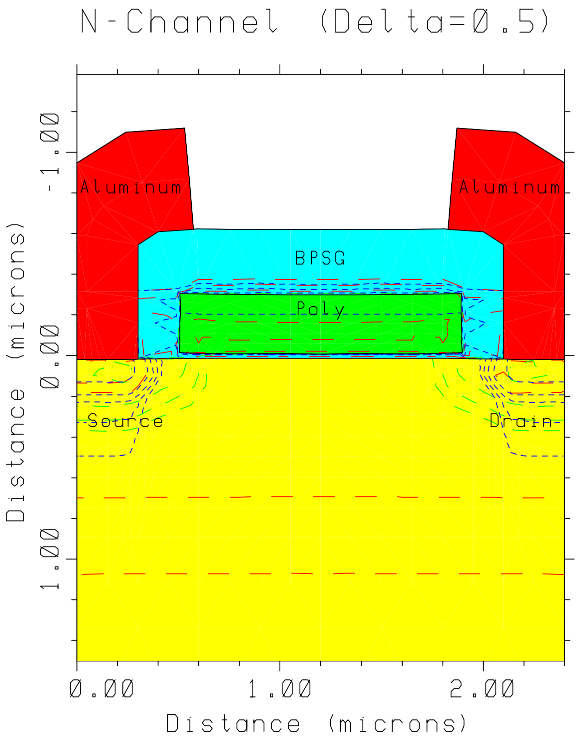


Figure 6-39 Final 1.2 micron NMOS structure, plotted by input file *s4ex7c.inp*

DMOS Power Transistor

This section illustrates the fabrication of a DMOS power transistor structure. The DMOS transistor is formed by diffusing boron into an n-type substrate, under a polysilicon gate. A self-aligned source region is formed by implanting phosphorus, using the poly gate as a mask. Contact to the drain is made at the back side of the wafer. The source metallization also provides contact to the p-type substrate of the n-channel MOS transistor. The TSUPREM-4 statements for simulating this DMOS power transistor are shown in Figures 6-40, 6-42, and 6-45. This particular input file features frequent saving of the structure. This is useful for analyzing the results at intermediate points in the processing, and for recovering from errors that may occur during simulation.

```
$ TSUPREM-4 DMOS Application

$ 1. Set grid spacing and error tolerance
MESH       LY.SURF=0.2  DY.SURF=0.1  LY.ACTIV=5.0 DX.MAX=0.75
METHOD     ERR.FAC=1.5

$ 2. Select models
METHOD     COMPRESS

$ 3. Initialize structure
INITIALIZE WIDTH=10.0  PHOSPHORUS=1E15

$ 4. Plot initial mesh
SELECT     TITLE="Initial Mesh"
PLOT.2D    SCALE  GRID C.GRID=2  Y.MAX=5

$ 5. P-well formation
DEPOSIT    OXIDE  THICK=0.70  SPACES=2
ETCH       OXIDE  RIGHT  P1.X=8.0

DIFFUSE    TEMP=1000  TIME=50  DRYO2

IMPLANT    BORON  ENERGY=50  DOSE=5E15
DIFFUSE    TEMP=950   TIME=120  STEAM
DIFFUSE    TEMP=1100  TIME=50

$ 6. Save the structure
SAVEFILE   OUT.FILE=S4EX8S4

$ 7. Boron contour plots
SELECT     Z=LOG10(BORON)  TITLE="After P-Well Diffusion"
PLOT.2D    SCALE  Y.MAX=5
COLOR      SILICON  COLOR=7
COLOR      OXIDE    COLOR=5
FOREACH    VAL (16 TO 21 STEP 1)
  CONTOUR    VALUE=VAL  LINE=5  COLOR=2
END
PLOT.2D    ^AX  ^CL

$ 8. Gate oxidation
ETCH       OXIDE  LEFT   P1.X=8.1
DIFFUSE    TIME=180  TEMP=1000  DRYO2
```

Figure 6-40 Mesh generation for DMOS power transistor, from input file *s4ex8.inp*

Mesh Generation

This simulation uses automatic grid generation and adaptive gridding. The automatic grid generation parameters have been adjusted to accommodate this large-geometry power device. The depth and spacing of the surface region (**LY.SURF** and **DY.SURF**, respectively) have been doubled and the depth of the active region (**DY.ACTIV**) has been increased from 4 to 5 microns.

A maximum horizontal grid spacing of **DX.MAX**=0.75 is used for this large-geometry device. The error factor for adaptive gridding has been set to 1.5 to reduce the number of nodes and speed up the simulation. A plot of the initial mesh is shown in Figure 6-41.

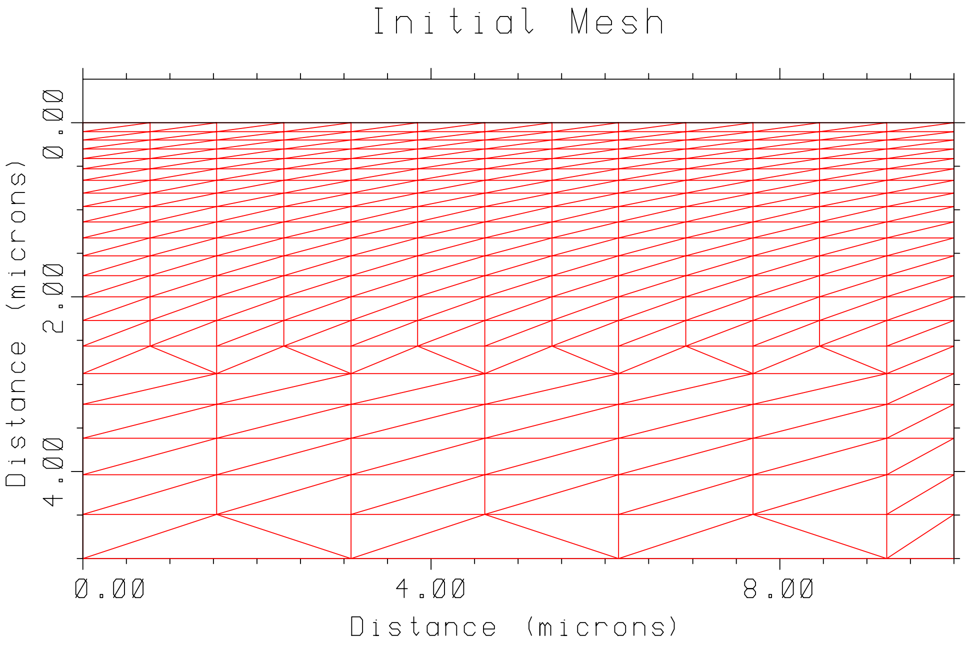


Figure 6-41 Initial grid for simulating DMOS power transistor

Processing the DMOS Power Transistor

Processing of the DMOS structure starts with the deposition of 7000Å of oxide. An opening in this oxide layer is then made and a thin oxide is grown on the exposed silicon. Boron is then implanted through the opening to form the p^+ contact for the body of the transistor. After implantation, the structure is oxidized, and the boron is driven further with an inert diffusion step. The **COMPRESS** oxidation model is used throughout the simulation to simulate oxidation of nonplanar structures.

```
$ 9. Save the structure
SAVEFILE   OUT.FILE=S4EX8S5

$ 10. Print oxide thickness
SELECT     Z=1
PRINT.1D   LAYERS  X.VALUE=0

$ 11. Poly gate formation
DEPOSIT    POLY  THICK=0.5  PHOS=1E20  SPACES=2
ETCH       POLY  RIGHT  P1.X=4

$ 12. Body formation
ETCH       OXIDE  START  X=4.0   Y=-10
ETCH       OXIDE  CONT   X=4.0   Y=10
ETCH       OXIDE  CONT   X=7.9   Y=10
ETCH       OXIDE  END    X=7.9   Y=-10

DIFFUSE    TIME=40  TEMP=1000  DRYO2

IMPLANT    BORON  DOSE=5E14  ENERGY=70
DIFFUSE    TIME=60   TEMP=950  DRYO2
DIFFUSE    TIME=200  TEMP=1100

$ 13. Save the structure
SAVEFILE   OUT.FILE=S4EX8S6
```

Figure 6-42 Second part of file *s4ex8.inp:* Processing of DMOS power transistor, through body diffusion

The structure after the oxidation and p-well drive is shown in Figure 6-43. *Note* that some oxide growth has occurred underneath the thick oxide, but not as much as occurred where the original oxide was removed.

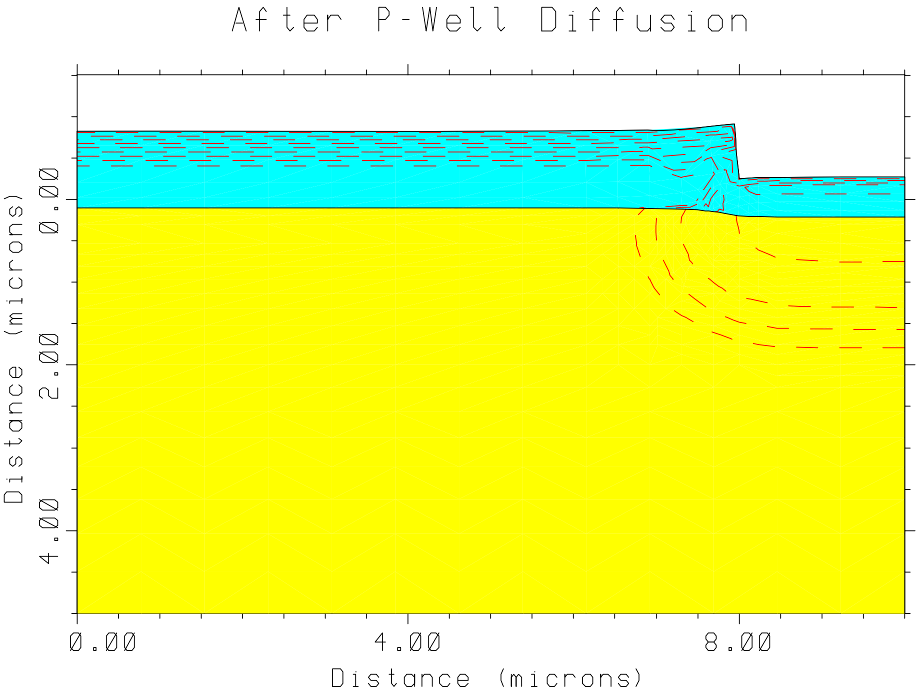


Figure 6-43 Structure with contours of boron concentration, after first p-well diffusion

Gate Processing

Next, the thick oxide is etched from the left portion of the structure, and a gate oxide is grown in this region. Again, one of the numerical oxidation models is required to simulate oxidation with nonuniform initial oxide thickness (**`COMPRESS`** is still being used). The thickness of the grown gate oxide is printed with the **`PRINT.1D LAYERS`** statement of step 10.

Next, the polysilicon gate is deposited and patterned, and the oxide is cleared from the source region. Boron is then implanted through the source opening and driven under the poly gate. The structure at this point is plotted in Figure 6-44. Observe that the boron has indeed diffused under the gate, where it forms the substrate of the n-channel transistor.

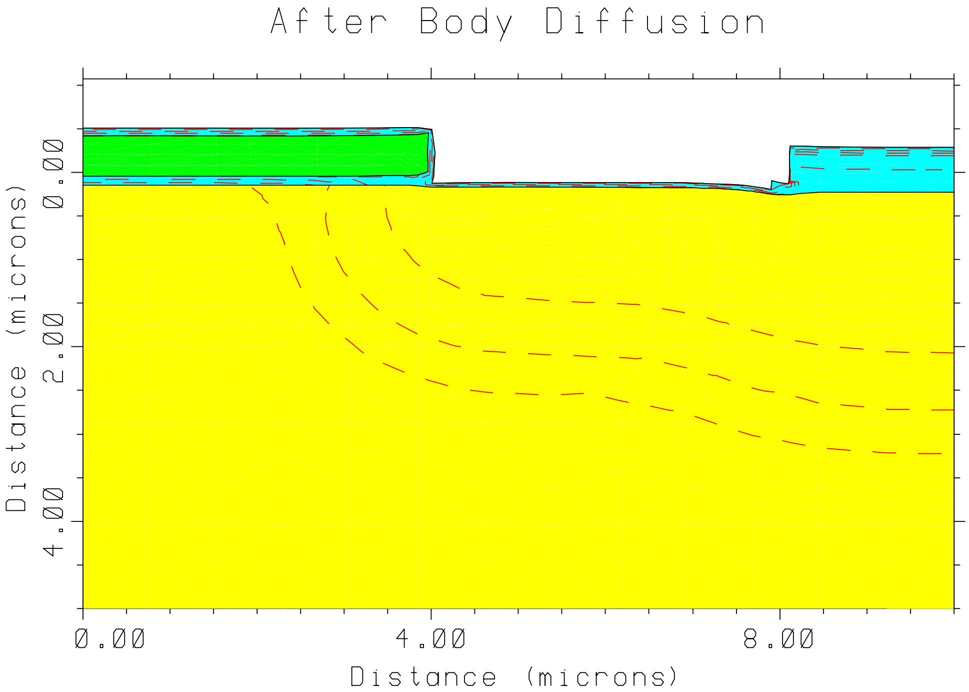


Figure 6-44 DMOS power transistor after p-type body diffusion

Source Processing

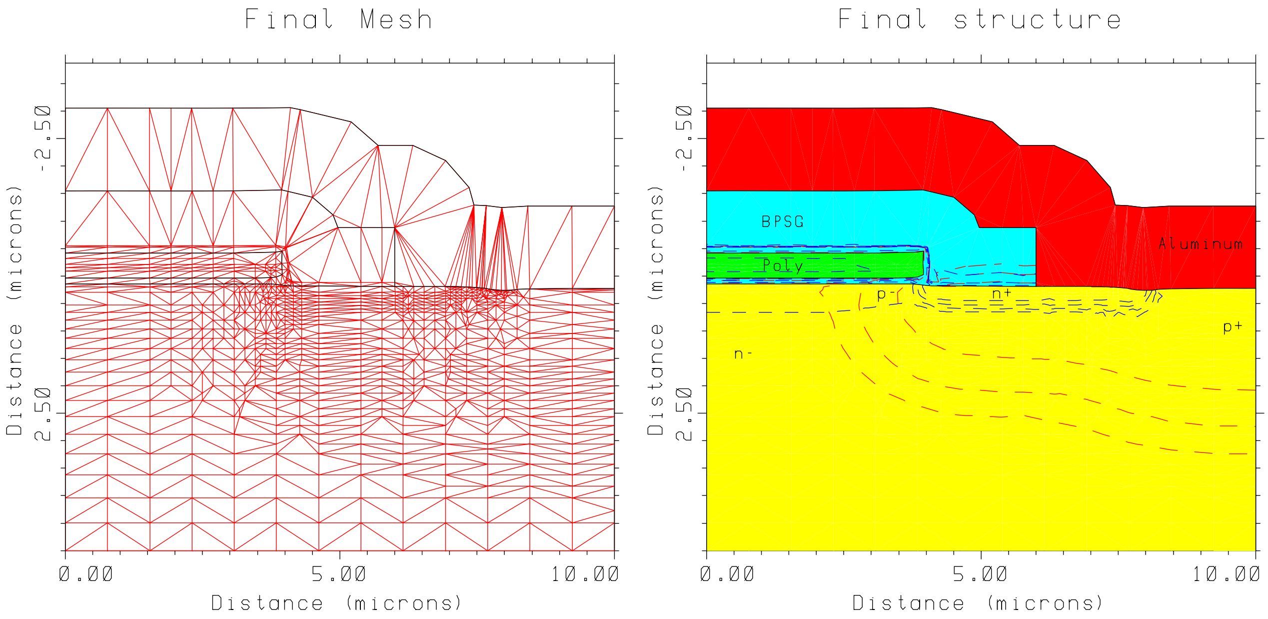
The opening for the source implant is now cleared, and a thin oxide is deposited. Phosphorus is then implanted and annealed to form the source of the transistor. Because the implantation is through an oxide, the *phosphorus* implant table, which does not include channelling, is used. BPSG is deposited to form an insulating layer over the gate. Finally, a contact hole is opened for the source and p-type body contact, and aluminum is deposited. The final structure is shown in Figure 6-46 (right). The silicon region has been labeled according to the doping type, and the other regions according to the type of material.

```
$ 14. Boron contour plots
SELECT     Z=LOG10(BORON)  TITLE="After Body Diffusion"
PLOT.2D    SCALE  Y.MAX=5
COLOR      SILICON  COLOR=7
COLOR      OXIDE    COLOR=5
COLOR      POLY     COLOR=3
FOREACH    VAL (16 TO 21 STEP 1)
  CONTOUR    VALUE=VAL  LINE.TYP=5  COLOR=2
END
PLOT.2D    ^AX  ^CL

$ 15. Source/drain formation
ETCH       OXIDE  START  X=4.1   Y=-10
ETCH       OXIDE  CONT   X=4.1   Y=10
ETCH       OXIDE  CONT   X=7.9   Y=10
ETCH       OXIDE  END    X=7.9   Y=-10

DEPOSIT    OXIDE  THICK=0.03

IMPLANT    PHOS  DOSE=1E15  ENERGY=50  IMPL.TAB=PHOSPHORUS
DIFFUSE    TEMP=950  TIME=60  DRYO2

$ 16. BPSG and metallization
DEPOSIT    OXIDE  THICK=1.00
DIFFUSE    TEMP=950  TIME=60
ETCH       OXIDE  RIGHT  P1.X=6

DEPOSIT    ALUMINUM  THICK=1.5
$ 17. Save the structure
SAVEFILE   OUT.FILE=S4EX8S7

$ 18. Plot final mesh
SELECT     TITLE="Final Mesh"
PLOT.2D    SCALE  GRID C.GRID=2  Y.MAX=5

$ 19. Boron and phosphorus contour plots
SELECT     Z=LOG10(BORON)  TITLE="Final structure"
PLOT.2D    SCALE  Y.MAX=5
COLOR      SILICON   COLOR=7
COLOR      OXIDE     COLOR=5
COLOR      POLY      COLOR=3
COLOR      ALUMINUM  COLOR=2
FOREACH    VAL (16 TO 21 STEP 1)
  CONTOUR    VALUE=VAL  LINE.TYP=5  COLOR=2
END

SELECT     Z=LOG10(PHOS)
FOREACH    VAL (16 TO 21 STEP 1)
  CONTOUR    VALUE=VAL  LINE.TYP=4  COLOR=4
END

PLOT.2D    ^AX  ^CL

LABEL      X=9.8   Y=-0.5   LABEL="Aluminum"  RIGHT
LABEL      X=1     Y=-0.9   LABEL="BPSG"
LABEL      X=1     Y=-0.1   LABEL="Poly"
LABEL      X=0.5   Y=1.5    LABEL="n-"
LABEL      X=9.8   Y=1.0    LABEL="p+"        RIGHT
LABEL      X=3.1   Y=0.4    LABEL="p-"
LABEL      X=5.2   Y=0.4    LABEL="n+"
```

Figure 6-45 Third part of *s4ex8.inp:* Final processing and plotting

Figure 6-46 (left) shows the final grid for the structure. Observe that the mesh has been refined (automatically, by adaptive gridding) where needed to resolve the impurity profiles. *Note* that the grid refinement is primarily vertical in the center of the source region where the profiles are essentially one-dimensional, but occurs in both directions at the corners of the source diffusion, where 2-D effects are important.

The final structure is shown in Figure 6-46 (right).



Final Mesh

Final structure

Figure 6-46 Final mesh for DMOS simulation (left), showing the result of adaptive gridding, and Final DMOS power transistor structure (right), produced by input file *s4ex8.inp*

Summary

The process shown here does not necessarily produce an ideal DMOS structure, nor is the simulation as accurate as it might be. Of particular importance are the conditions of the final drive of the p-type body diffusion in step 22 and the possibility that impurities may diffuse from the polysilicon gate through the gate oxide and into the channel. Observe in the final plot that there has apparently been some diffusion of both boron and phosphorus from the poly into the channel. Another concern is that oxidation-enhanced diffusion effects may be important in this example. This can be checked by repeating the simulation using the **`PD.TRANS`** point defect model.

SOI MOSFET

This example shows how TSUPREM-4 can be used to simulate a silicon-on-insulator (SOI) process. It illustrates several features of the program that have not been used in the other examples, including the use of the **REGION** statement and the technique for depositing layers with nonuniform grid spacings. The input statements from file *s4ex9.inp* are shown in Figures 6-47 and 6-49.

```
$ TSUPREM-4 SOI Structure (0.2um epi) Application

$ Specify x mesh
LINE  X    LOCATION=0     SPACING=0.25   TAG=LEFT
LINE  X    LOCATION=0.5   SPACING=0.025
LINE  X    LOCATION=0.6   SPACING=0.025
LINE  X    LOCATION=1.05  SPACING=0.1    TAG=RIGHT

$ Specify y mesh
LINE  Y    LOCATION=0     SPACING=0.02   TAG=OXTOP
LINE  Y    LOCATION=0.5   SPACING=0.15   TAG=OXBOTTOM
LINE  Y    LOCATION=2.0   SPACING=1.0    TAG=SIBOTTOM

ELIMINATE  COLUMNS  X.MIN=0.45  X.MAX=0.7  Y.MIN=0.05
ELIMINATE  COLUMNS  Y.MIN=0.5

$ Define isolation oxide and silicon substrate
REGION     OXIDE    XLO=LEFT  XHI=RIGHT  YLO=OXTOP     YHI=OXBOTTOM
REGION     SILICON  XLO=LEFT  XHI=RIGHT  YLO=OXBOTTOM  YHI=SIBOTTOM
INITIALIZE <100> BORON=1E15

$ Deposit epi with nonuniform vertical grid spacing
DEPOSIT    SILICON  BORON=1E17  THICKNESS=0.1  SPACES=5  DY=0.01  YDY=0.1
DEPOSIT    SILICON  BORON=1E17  THICKNESS=0.1  SPACES=5  DY=0.01

$ Plot initial mesh
SELECT     Z=1  TITLE=”Initial Mesh”
PLOT.2D    GRID  Y.MAX=1  SCALE  C.GRID=2
PRINT.1D    X.VALUE=0.0  LAYERS
```

Figure 6-47 Mesh generation for SOI MOSFET, from input file *s4ex9.inp*

Mesh Generation

The structure to be simulated consists of a silicon wafer with a half micron of oxide and 0.2 microns of silicon on top. The active device is fabricated in the top silicon layer. The actual structure is formed by implantation of oxygen and recrystallization of the silicon layer on top, a process that cannot be simulated by TSUPREM-4. Instead, an initial structure is defined consisting of the silicon substrate and the oxide layer, on which the top silicon layer is deposited. *Note* that due to the symmetry of the final structure, only the left half needs to be simulated.

The silicon substrate and oxide layers are defined with **LINE** statements, as in previous examples. In this case, the **TAG** parameter must be added to grid lines on structure boundaries and material interfaces. Thus, in the *y* direction the tag *OXTOP* is used to mark the top of the oxide, *OXBOTTOM* to mark the oxide/silicon interface, and *SIBOTTOM* to mark the bottom of the structure. **REGION** statements are added to identify the silicon and oxide regions. The **REGION** statements use the tags to specify the extent of the regions. Thus, for example, it is specified that the oxide region extends from the grid line tagged *OXTOP* to the line tagged *OXBOTTOM*, and from *LEFT* to *RIGHT* in the horizontal direction.

Depositing a Layer with Nonuniform Grid Spacing

The top silicon layer is added using **DEPOSITION** statements. To accurately simulate the final structure with the Medici device simulator, a fine grid spacing (about 100Å) is needed near each silicon/oxide interface. To reduce simulation time, a coarser grid is specified in the middle of the layer. Thus, the ideal mesh would have a spacing that varies smoothly from fine to coarse to fine.

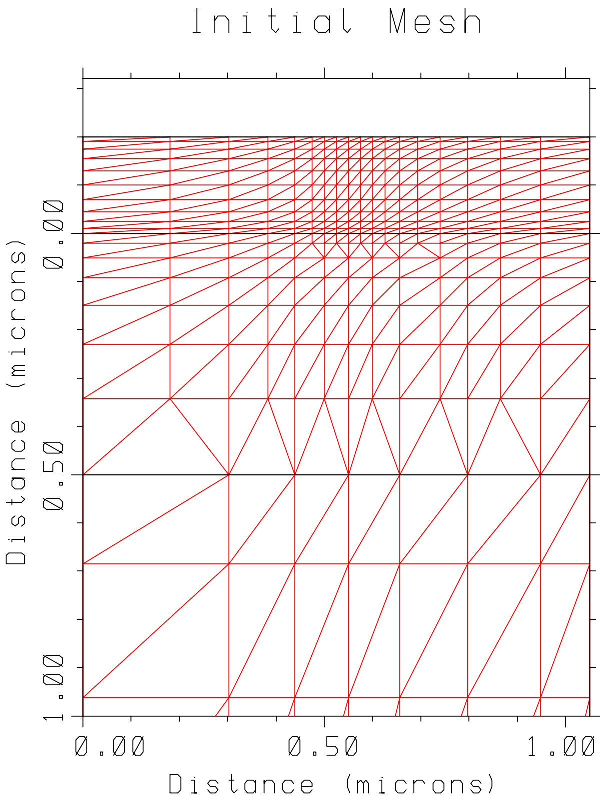


Figure 6-48 Initial grid for simulating SOI MOSFET

A mesh of this type (shown in Figure 6-48) can be generated using the **DY** and **YDY** parameters on the **DEPOSITION** statement. The **DY** parameter specifies the desired grid spacing at some depth in the deposited layer specified by the **YDY** parameter. The grid spacing above and below this depth is increased or decreased to produce the specified number of spaces. To specify the grid spacing at two points in the layer (i.e., at each interface), two **DEPOSITION** statements must be used. Each specifies a 1000Å layer of silicon divided into five grid spaces, as specified by the **SPACES** parameter.

In each case the **DY** parameter is used to specify the desired grid spacing of 0.01 microns. In the first case the 0.01-micron grid spacing occurs 0.1 microns below the top of the layer (i.e., at the bottom of the layer), while in the second case the default value of **YDY**=0 is used, so the fine grid spacing occurs at the top of the layer. *Note* that the same mesh could have been generated by adding **LINE** and **REGION** statements to define the top silicon layer. The difficulty is that only a single doping concentration can be specified for the entire structure on the **INITIALIZE** statement, and it was necessary to specify a doping of 10^{17}/cm^3 in the top layer, but only 10^{15}/cm^3 in the lower layers.

```
$ Use vertical oxidation model to grow gate oxide
DIFFUSE    TIME=55 TEMP=950  DRYO2

SELECT     Z=1
PRINT.1D   X.VALUE=0.0  LAYERS

$ Deposit, dope, and pattern poly
DEPOSIT    POLY  THICK=0.3  SPACES=2
METHOD     COMPRESS
DIFFUSE    TIME=30  TEMP=800  PHOS=1E20
ETCH       POLY  LEFT  P1.X=0.5

$ Source/drain implant and anneal
IMPLANT    ARSENIC  DOSE=4E15  ENERGY=45
DIFFUSE    TIME=70  TEMP=950

$ BPSG and contact holes
DEPOSIT    OXIDE  THICK=0.3
DEPOSIT    PHOTORESIST  THICK=1.0
ETCH       PHOTORESIST  LEFT  P1.X=0.3
ETCH       OXIDE  TRAP  THICK=0.7  ANGLE=85
ETCH       PHOTORESIST  ALL

$ Metallization
DEPOSIT    ALUMINUM  THICK=0.4
DEPOSIT    PHOTORESIST  THICK=1.0
ETCH       PHOTORESIST  RIGHT  P1.X=0.6
ETCH       ALUMINUM  TRAP  THICK=0.7  ANGLE=85
ETCH       PHOTORESIST  ALL

$ Reflect structure, then save complete MOSFET
SAVEFILE   OUT.FILE=S4EX9S1
STRUCTURE  REFLECT  RIGHT
SAVEFILE   OUT.FILE=S4EX9P2  MEDICI

$ Plot grid & profiles for complete structure
SELECT     Z=DOPING  TITLE=”SOI MOSFET”
PLOT.2D    SCALE  Y.MAX=1
COLOR      SILICON   COLOR=7
COLOR      OXIDE     COLOR=5
COLOR      POLY      COLOR=3
COLOR      ALUMINUM  COLOR=2
FOREACH    X ( 16 TO 21 STEP 1 )
  CONTOUR    VALUE=( 10^ X )    COLOR=4  LINE.TYP=2
  CONTOUR    VALUE=(-(10^ X ))  COLOR=2  LINE.TYP=5
END
PLOT.2D    ^AX  ^CL

PLOT.2D    GRID  SCALE  Y.MAX=1  C.GRID=2

$ Plot 1D concentration profiles
SELECT     Z=LOG10(ACTIVE(BORON))  TITLE=”Doping Profiles” +
           LABEL=”log(Active Concentration)”
PLOT.1D    X.VALUE=1  BOTTOM=15  TOP=21  LEFT=-0.3  RIGHT=0  LINE.TYP=5  +
           COLOR=2
LABEL      LABEL=”Boron (x=1)”  X=-0.1  Y=17

SELECT     Z=LOG10(ACTIVE(ARSENIC))
PLOT.1D    X.VALUE=0  ^AXES  ^CLEAR  LINE.TYP=2  COLOR=3
LABEL      LABEL=”Arsenic (x=0)”  X=-0.1  Y=20.3
```

Figure 6-49 Processing of SOI MOSFET, from input file *s4ex9.inp*

Process Simulation

The processing of the SOI MOSFET is straightforward. *Note* that the analytical oxidation models cannot be used in this example, even though the surface is planar and unoxidized, because of the existence of the lower oxide/silicon interfaces.

The **VERTICAL** model can be used, but only because there is no path for oxygen to diffuse to the lower oxide layer, and thus no growth at the lower interfaces. If there were a path for oxygen to diffuse to the lower oxide, the **COMPRESS** or **VISCOEL** model would be needed, and if it were necessary to model lifting of the upper silicon layer, the **VISCOUS** model with **SKIP.SIL** set false would be needed.

Source/drain contacts have been opened and aluminum has been deposited and patterned in order to form contacts for Medici. The structure is reflected along the right edge to form a full device, and saved in Medici format in file *S4EX9P2*. The full, final structure is shown in Figure 6-50 on the left; the final grid is plotted in on the right.

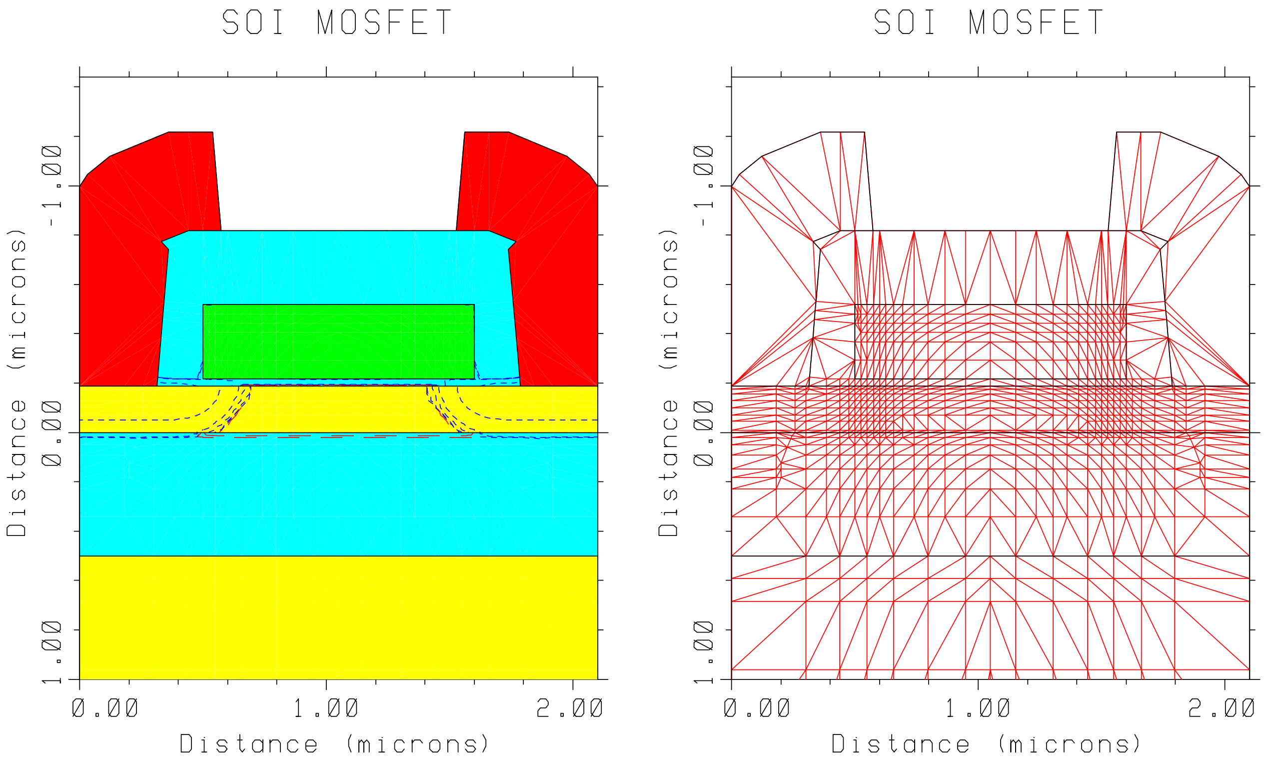


Figure 6-50 Final structure, showing contours of net doping for SOI MOSFET (left) and final grid for SOI MOSFET (right)

Figure 6-51 shows the doping profiles in the final structure. The arsenic profiles in the source/drain (*x*=0) and the boron profiles in the channel (*x*=1) are shown.

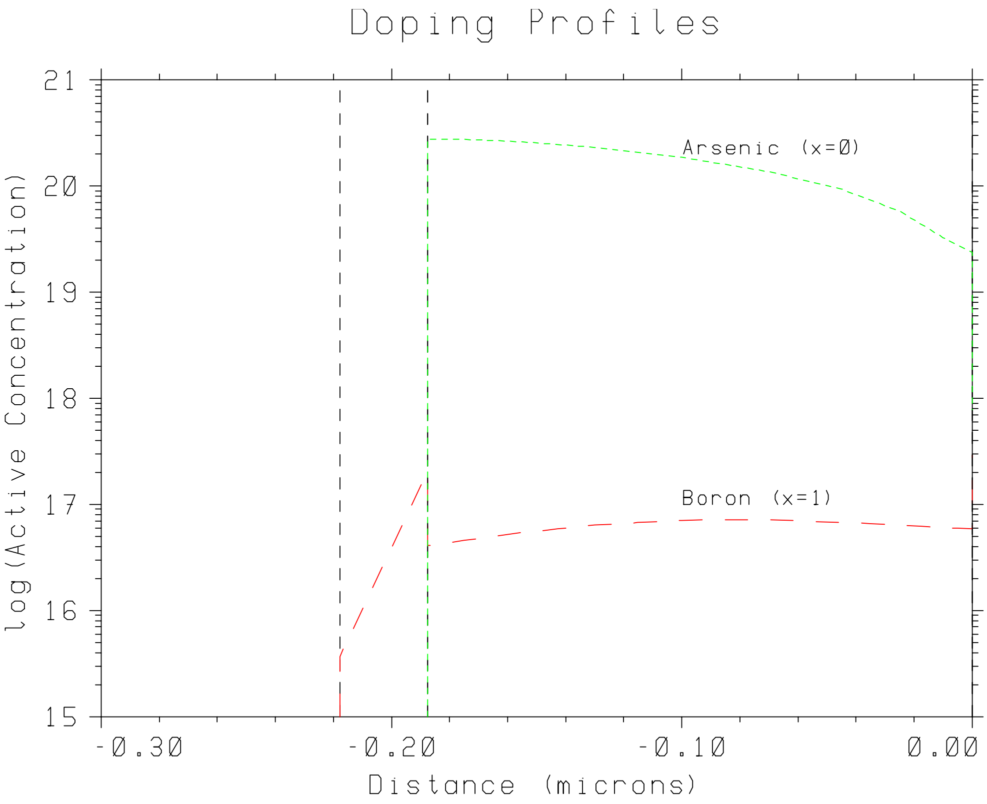


Figure 6-51 Channel and source/drain doping profiles for SOI MOSFET

MOSFET with Self-Aligned Silicides

This example illustrates the simulation of silicide growth using TSUPREM-4. The input files for this example are shown in Figures 6-52, 6-53, and 6-54. The statements are separated into two files—*s4ex10.inp* contains the processing statements for performing the simulation, while *s4ex10p.inp* contains the statements for plotting the structure.

Preparation for Silicidation

Figure 6-52 shows the statements used to initialize the structure, grow the gate oxide, deposit and pattern the poly gate, form the lightly-doped source/drain extension and the heavily-doped source/drain, and clear the oxide from the poly gate and the source/drain in preparation for silicidation. Automatic grid generation is used in both the vertical and horizontal directions, but because the structure is very small it is appropriate to reduce the depth of the active region and the grid space is provided automatically by adaptive gridding.

```
$ TSUPREM-4 Silicidation Example

MESH     GRID.FAC=0.9

MESH     DY.SURF=0.03  LY.ACTIV=0.5 DY.ACTIV=0.1
INITIALIZE BORON=1.0E15  WIDTH=0.6 DX=0.1

$ Grow gate oxide
DIFFUSION TIME=25 TEMP=850 HCL=4.5 DRYO2

$ Deposit and pattern gate
DEPOSIT  POLYSILI  THICKNES=0.2 SPACES=5 DY=0.02
ETCH     POLYSILI  RIGHT P1.X=0.125
ETCH     OXIDE     RIGHT P1.x=0.125

$ Oxidize
METHOD   COMPRESS
DIFFUSION TIME=25 TEMP=850 HCL=4.5 DRYO2

$ LDD implant
IMPLANT  ARSENIC  DOSE=5E13 ENERGY=30  IMPL.TAB=ARSENIC

$ Form sidewall spacer
DEPOSIT  OXIDE  THICKNESS=0.126 SPACES=2
ETCH     OXIDE  TRAPEZOI THICKNESS=0.175

$ Heavy source/drain implant and anneal
IMPLANT  ARSENIC  DOSE=1E15 ENERGY=60  IMPL.TAB=ARSENIC
METHOD   PD.TRANS
DIFFUSION TIME=10 TEMP=900 DRYO2

$ Isotropic etch to clear gate and source/drain
ETCH     OXIDE TRAP THICK=0.015 ANGLE=45
```

Figure 6-52 First part of input file *s4ex10.inp*: NMOS transistor processing

Silicidation

Figure 6-53 shows the statements for growing the self-aligned titanium silicide layer. First, titanium is deposited over the entire structure. Where titanium comes into contact with silicon or polysilicon, a thin (2 nm, by default) layer of $TiSi_2$ is inserted automatically by the program. The structure after titanium deposition is shown in Figure 6-55. The next step is to anneal for a little over a minute at 650 degrees. Because it is expected that impurity diffusion is negligible, the **PD.FERMI** model is selected to reduce the simulation time.

During silicide growth, silicon atoms enter the silicide layer and diffuse to the $TiSi_2$/Ti interface where they react to form more $TiSi_2$. Silicon and titanium are consumed at their respective interfaces with the silicide, and silicide is formed at the $TiSi_2$/Ti interface. The deformation of the various layers due to the consumption of silicon and titanium and the production of titanium silicide is calculated using the **COMPRESS** model for viscous flow

```
$ Deposit titanium
DEPOSIT  MAT=TITANIUM  THICK=0.025 SPACES=2

$ Plot structure before silicidation
SELECT   TITLE=”Before Silicidation”
SOURCE   s4ex10p.inp

$ Grow the silicide
METHOD   PD.FERMI
DIFFUSION TIME=1.287 TEMP=650

$ Plot structure after silicidation
SELECT   TITLE=”After Silicidation”
SOURCE   s4ex10p.inp

$ Final structure
ETCH     MAT=TITANIUM ALL
STRUCTURE REFLECT LEFT

$ Plot structure after silicidation
SELECT   TITLE=”After Titanium Etch”
SOURCE   s4ex10p.inp

$ Print layer information
SELECT Z=DOPING
PRINT.1D X.V=0.6 LAYERS
PRINT.1D X.V=0.0 LAYERS
```

Figure 6-53 Second part of input file *s4ex10.inp*: Silicide growth

```
$ Plot sequence for silicidation example
$
$ To use:
$   select title=”Title”
$   source s4ex10p.inp
$
PLOT.2D  Scale Y.MAX=0.3
COLOR    MAT.POLY    COLOR=3
SELECT   Z=LOG10(ARSENIC)
FOREACH  X ( 15 TO 20 )
  CONTOUR  VAL=@{X} COLOR=3 LINE=3
END
COLOR    MAT=TITANIUM COLOR=2
COLOR    MAT=TISI2    COLOR=6
COLOR    MAT=OXIDE    COLOR=5
LABEL    X=0.36 Y=-0.23 SIZE=0.3 RECT C.RECT=2 LABEL=”Titanium”
LABEL    X=0.36 Y=-0.18 Size=0.3 Rect C.RECT=6 LABEL=”TiSi2”
LABEL    X=0.36 Y=-0.13 SIZE=0.3 RECT C.RECT=3 LABEL=”Polysilicon”
LABEL    X=0.36 Y=-0.08 Size=0.3 Rect C.RECT=5 LABEL=”Oxide”
LABEL    X=0.55 Y=0.25  Size=0.3 Line=3 C.LINE=3 LABEL=”Arsenic” RIGHT
PLOT.2D  ^AX ^CL
```

Figure 6-54 Input file *se4ex10p.inp*: Plotting results

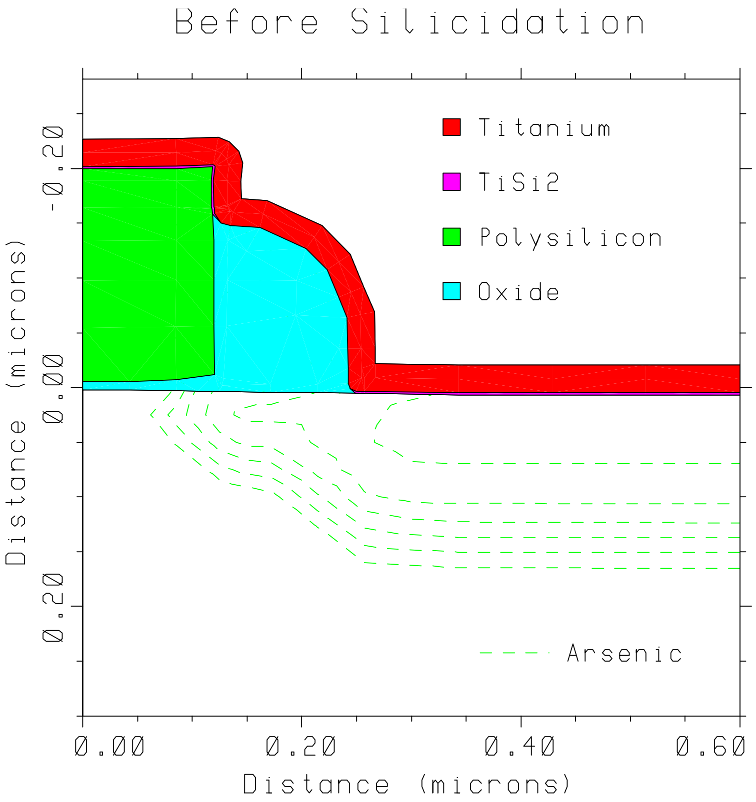


Figure 6-55 Structure immediately before silicide growth step

The structure after the silicide growth step is shown in Figure 6-56. Observe that most of the titanium has been consumed. After removing the remaining titanium the full structure is shown in Figure 6-57.

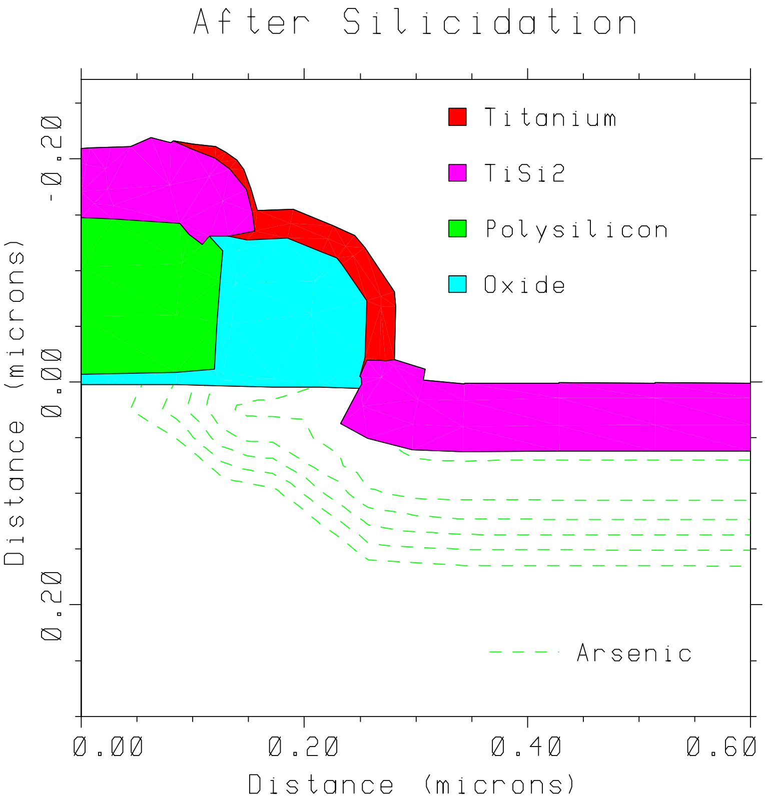


Figure 6-56 Structure after silicide growth step

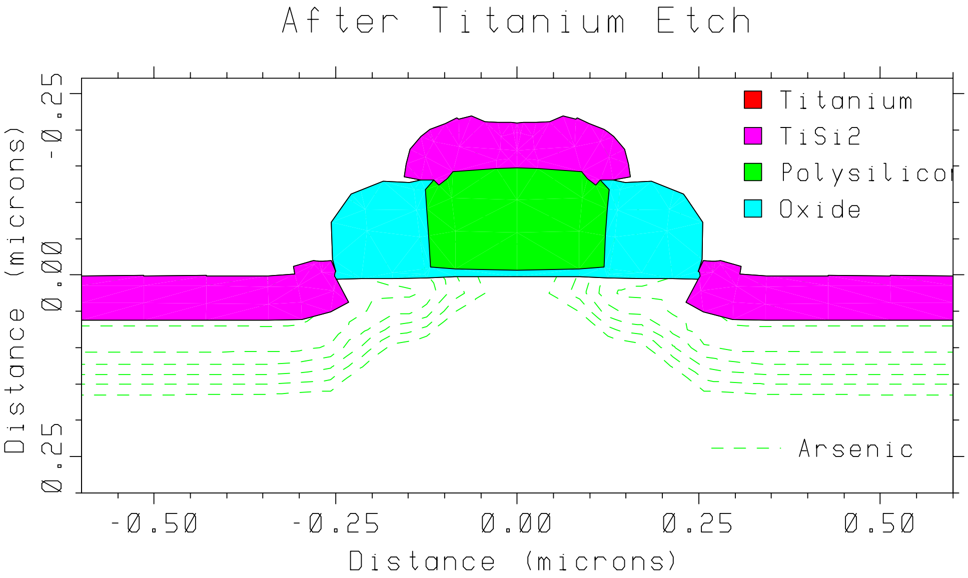


Figure 6-57 Final structure, after removal of remaining titanium

Polysilicon Emitter Study

This example shows a simulation of the emitter region of a double-polysilicon bipolar transistor. The effect of the arsenic out-diffusion from the emitter polysilicon on the emitter and base widths is shown. The simulations can be used to examine the dependence of the device performance on the width of the emitter stripe. The advanced polysilicon grain-growth and diffusion models are used to simulate the distribution of arsenic in the emitter.

Process Simulation

Figure 6-58 shows the input file *s4ex11a.inp* for simulating the emitter region of the bipolar transistor. The emitter width is defined by the name **`EMITWID`**. The input file *s4ex11b.inp* for simulating the 2-micron emitter is identical except for the values of **`IDENT`** (which is set to "**`B`**") and the value of **`EMITWID`** which is set to 2.0.

The critical dimensions of the structure are specified by defining the parameters **`EMITSTA`**, **`EMITWID`**, and **`SPACERWID`**. The grid spacings are set to give a reasonable compromise between speed and accuracy. Adaptive gridding is used to increase the grid density where the arsenic diffuses to form the emitter; because of the shallowness of the emitter, the minimum grid spacing for adaptive gridding on arsenic is decreased to 0.01 micron. The accuracy of the results can be verified by reducing the value of **`GRID.FAC`** for later simulations.

Models

Because oxidation-enhanced of transient-enhanced diffusion are not expected to be significant in this structure, the **`PD.FERMI`** diffusion model can be used with a shallow simulation structure. Because of the high arsenic concentration in the emitter, however, it would be wise to check the results using the **`PD.FULL`** model and a deep simulation structure.

Processing

The emitter structure is formed by a series of deposition and etch steps, culminating in the deposition of the poly layer in the trench formed by the emitter opening. The temperature of the poly deposition is 620° C; this temperature determines the initial grain size. The emitter is then implanted with arsenic at a 7-degree tilt. Because of the tilt, the right sidewall of the emitter receives more of the implant dose than the left sidewall, but much less than the bottom of the emitter opening. The arsenic is then diffused throughout the poly and into the single-crystal silicon with a high-temperature RTA step.

```
$ Parameters
%DEFINE IDENT    A
%DEFINE EMITSTA 0.5
%DEFINE EMITWID 1.0
%DEFINE SPACERWID 0.2

$ Half the structure
MESH   GRID.FAC=1.5

LINE   X LOC=0.0 SPACING=0.2
LINE   X LOC=@{EMITSTA}+@{SPACERWID}/2 SPACING=0.05
LINE   X LOC=@{EMITSTA}+@{EMITWID}/2 SPACING=0.15
LINE   Y LOC=0.0 SPACING=0.025
LINE   Y LOC=0.1 SPACING=0.025
LINE   Y LOC=1.0 SPACING=0.1
ELIMINATE COLUMNS Y.MIN=0.2
ELIMINATE COLUMNS Y.MIN=0.2
INITIAL <100> BORON=1E14

METHOD  MAT=SILICON IMP=ARSENIC MIN.SPAC=0.01

$ Emitter Cut
DEPOSIT OXIDE THICK=0.6 DY=0.05
ETCH    OXIDE RIGHT P1.X=@{EMITSTA}

$ Dielectric Screening Layers
DEPOSIT  OXIDE THICK=0.04 SPACES=2
DEPOSIT  NITRIDE THICK=0.05 SPACES=2

$ Spacer
ETCH NITRIDE RIGHT P1.X=@{EMITSTA}+@{SPACERWID}

$ Oxide dip
ETCH OXIDE   TRAPEZ   THICK=0.06 UNDERCUT=0.06

$ Build Full Structure
STRUCTUR REFLECT RIGHT

$ N-poly
DEPOSIT  POLYSILI THICK=0.22 ARC.SPAC=0.05 DY=0.02 TEMPERAT=620

$ N+ implant
IMPLANT  DOSE=1.0E+16  ENERGY=50 ARSENIC TILT=7.0

SAVE OUT.FILE=S4EX11@{IDENT}S1

$ Emitter drive-in
DIFFUSIO TEMPERAT=1125. TIME=0.5 NITROGEN

SAVE OUT.FILE=S4EX11@{IDENT}S2
```

Figure 6-58 Listing of input file *s4ex11a.inp* for simulating the bipolar emitter structure

Plotting the Results

The results are now plotted for the following structures:

- After Implant
- Doping and Grain Size
- Doping vs. Stripe Width

After Implant

Figure 6-59 shows the layers of the structure and the arsenic concentration in the poly immediately after the implant. This figure and the following figures were plotted using the input file *s4ex11c.inp*, which illustrates techniques for producing multiple plots on a single screen and for annotating shaded contour plots. The first part of *s4ex11c.inp* is shown in Figure 6-60.

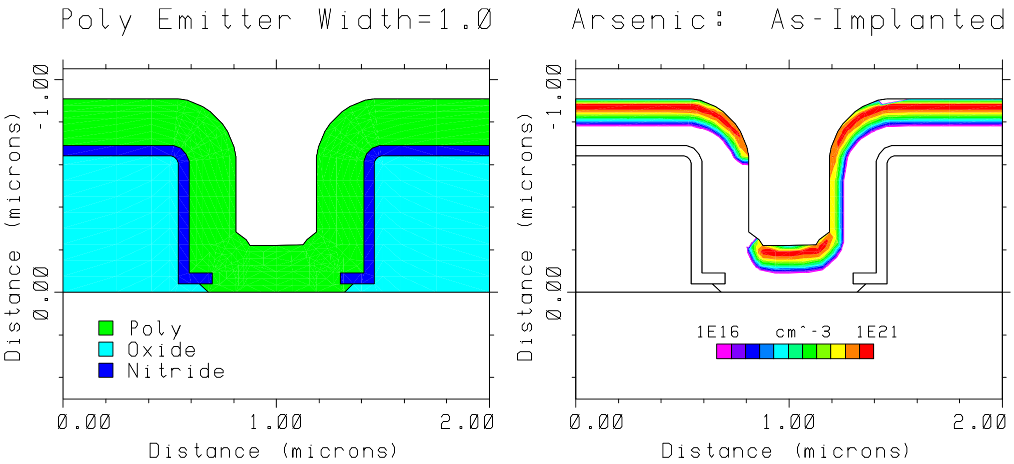


Figure 6-59 Bipolar emitter structure and as-implanted arsenic profiles, as plotted using *s4ex11c.inp*

```
$ Parameters for plot size and placement
%define YMAX 0.5
%define YLEN 5.8
%define XOFF1 1.5
%define XOFF2 10.8
%define LSIZ 0.3
%define TSIZ 0.35

$ As-implanted structure
initial in.file=S4EX11AS1

$ Plot structure
select title=”Poly Emitter Width=1.0”
plot.2d scale y.max=@{YMAX} x.off=@XOFF1 y.len=@YLEN +
  t.siz=@TSIZ x.siz=@LSIZ y.siz=@LSIZ
color   poly color=3
color   oxide color=5
color   nitride  color=4
label   label=”Poly”    c.rect=3 size=0.25 x=0.3 y=0.2
label   label=”Oxide”   c.rect=5 size=0.25 x=0.3 y=0.3
label   label=”Nitride” c.rect=4 size=0.25 x=0.3 y=0.4
plot.2d ^ax ^cl

$ Plot as-imlanted arsenic
select z=log10(arsenic) title=”Arsenic:  As-Implanted”
plot.2d scale y.max=@{YMAX} ^cl x.off=@XOFF2 y.len=@YLEN +
  t.siz=@TSIZ x.siz=@LSIZ y.siz=@LSIZ
%define TOP 1E21
%define BOTT 1E16
%define INC  (log10(@{TOP})-log10(@{BOTT}))/10
foreach I (0 to 10)
   color min.value=(log10(@{BOTT}) + @{INC}*( I )) +
         max.value=(log10(@{BOTT}) + @{INC}*( I +1)) +
         color=(18- I )
   label rectangle c.recta=(18- I ) c.siz=0.25 +
         x=(@{XOFF2}+2.6+0.25*( I )) y=2.5 cm
end
label label=”@{BOTT}” c.siz=0.2 x=@{XOFF2}+2.1 y=2.75 cm
label label=”cm^-3  @{TOP}” c.siz=0.2 x=@{XOFF2}+3.5 y=2.75 cm
plot.2d ^ax ^cl
```

Figure 6-60 First part of *s4ex11c.inp,* for plotting the structure and contours of as-implanted arsenic concentration

Doping and Grain Size

Figure 6-61 shows the total arsenic concentration and the poly grain size after the RTA step. The arsenic concentration is largest at the left and right edges of the structure, where the initial doping is high and the arsenic is confined to the poly layer by the underlying nitride layer. The concentration is lowest in the sidewalls of the emitter opening, where the as-implanted doping is low. The concentration at the center of the emitter opening is lower than that at the left and right edges of the structure because of diffusion into the sidewalls of the opening and into the single-crystal silicon. The rate of grain growth during high-temperature processing depends strongly on the doping concentration; thus the final contours of grain size are similar to the final contours of total arsenic concentration.

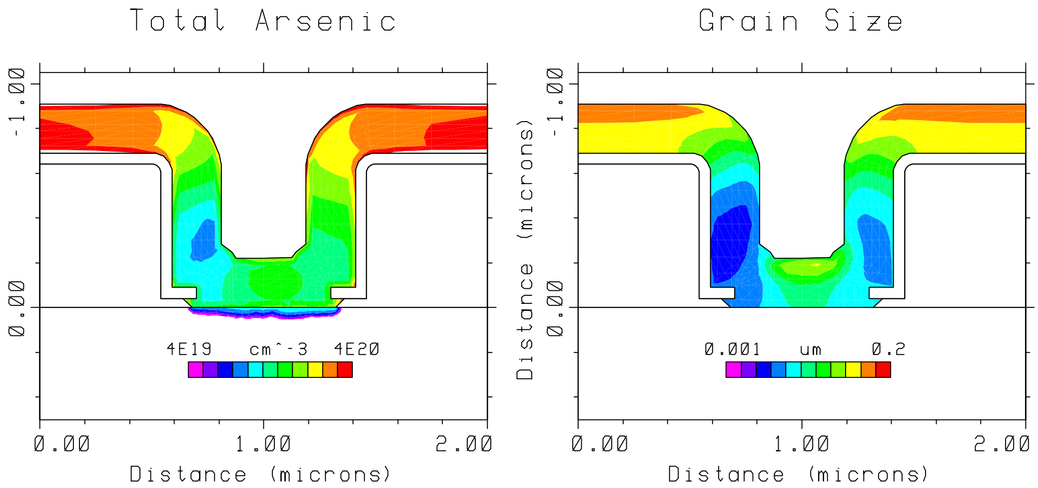


Figure 6-61 Contours of total arsenic concentration and poly grain size after RTA

Doping vs. Stripe Width

The depletion of the arsenic in the center of the emitter opening is a strong function of the width of the emitter stripe. Figure 6-62 compares the net doping for a 2-micron emitter stripe with that for a 1-micron stripe. For the 2-micron stripe, the doping in the poly across the width of the emitter opening is much higher. It can also be observed that more arsenic has diffused from the poly into the single-crystal silicon; the 2-micron emitter stripe produces a wider emitter and a narrower base than the 1-micron emitter stripe.

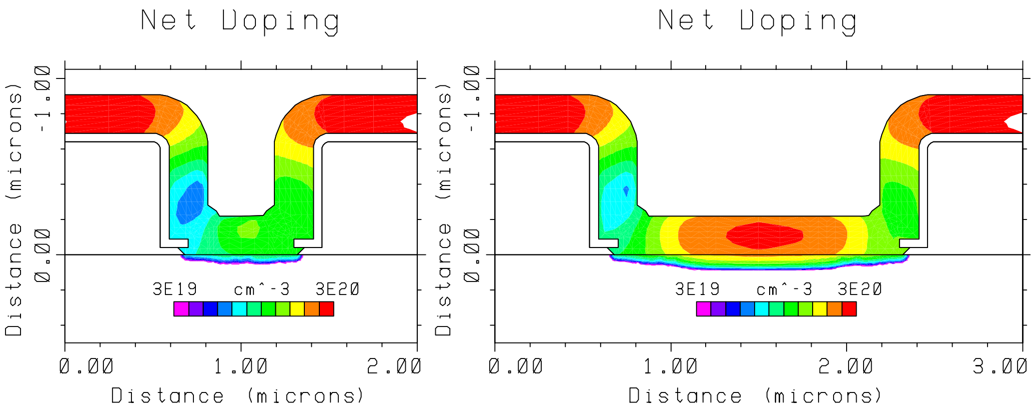


Figure 6-62 Contours of net doping for 1-micron and 2-micron emitter stripes

SiGe HBT Process

This application presents a complete process flow of making a SiGe HBT device. It illustrates how to use TSUPREM-4 USEIT to introduce SiGe related advanced models and how to compute lattice mismatch strain and stress. Within this example, strain and Germanium trap effects on Boron diffusion are considered. The Carbon interstitial trap effect is also taken into consideration.

Initial Structure and Collector Region Generation

The input statements for generating the initial structure are listed in Figure 6-63.

```
$ This example shows how to use USEIT to define Boron diffusion
$ model in SiGe considering the effects of Germanium induced
$ strain/stress, or Germanium traps, and also the model of
$ carbon-interstitial traps.

$ TSUPREM-4: SiGe HBT Application

$ Set device structure control parameters
assign name=width n.val=0.7
assign name=bwidth n.val=0.4
assign name=ewidth n.val=0.2

$ Mesh and initialize
mesh grid.fac=1 dx.max=0.5 dx.min=0.4 dx.ratio=1.5 +
     ly.surf=0.2 dy.surf=0.1 ly.active=8 dy.active=1 +
     ly.bot=8 dy.bot=1.5 dy.ratio=2 fast
init <100> impurity=boron I.resist=30 resistiv width=@width

$ Models
method pd.trans viscoela imp.adap dif.adap ^ox.adap

$ Buried Arsenic implant
deposit oxide thick=0.03
implant arsenic dose=5e+15 energy=200

$ Nburied_drive
diff temp=600 time=50 t.fin=1150 f.N2=10 f.O2=0.2
diff temp=1150 time=120 f.N2=10 f.O2=0.2
diff temp=1150 time=50 t.fin=800 f.N2=10 f.O2=0.2
diff temp=800 time=30

$ Epi si and pad oxide
etch oxide all
deposit silicon thick=0.5 space=10 boron=1e15
deposit oxide thick=0.03

$ Ncollector implant
implant phos dose=5e+13 energy=200

$ N collector drive
etch oxide all
diff time=15 temp=800 t.fin=950 dryO2
diff time=10 temp=950 dryO2
diff time=30 temp=950 t.fin=800

$ Etch si
etch oxide all
extract name=top silicon x=0 dist=0 y.ext ^write assign
assign name=ySi n.val=@top+0.25
etch silicon right p1.x=@bwidth p2.y=@ySi

$ Fill oxide
deposit oxide thick=0.4 space=3

$ Etch
extract name=top silicon x=0 dist=0 y.ext ^write assign
assign name=ytop1 n.val=@top-0
etch oxide start x=0 y=-10
etch cont x=0 y=@ytop1
```

```
etch cont x=10 y=@ytop1
etch cont x=10 y=-10
etch done x=0 y=-10
```

Figure 6-63 First part of sige_hbt.inp, for generating initial structure and the collector region.

A coarse mesh is first generated for the silicon wafer with Boron impurity. Several build-in models are then selected for impurity diffusion, deformation evolution, and adaptive meshing. The next steps generate the buried collector region.

Using SiGe Related Models

The next three SOURCE statements include USEIT files that define the SiGe data, the model for computing lattice mismatch strain and stress, the model for boron diffusion with strain effect and Germanium/Carbon trap effect. They are: Ge-in-Si.data, sige-stress-trap.usit and carbon.usit, as shown in Figure 6-64, Figure 6-65, and Figure 6-66.

```
$------ Germanium coefficients -----------
$ USEIT SiGe diffusion model implementation - data for germanium in
silicon
$ most of the data is similar to arsenic

impurity new imp=germanium at.num=32 at.wt=72.61 +
 tif.name=Ge c.init=1E5 impl.tab=arsenic
impurity imp=germanium t.act.0=8e-16 t.act.e=-4.2 act.min=1 cm.sec
impurity imp=germanium mater=silicon dix.0=0.037 dix.e=5.46 cm.sec
impurity imp=germanium mater=silicon /mater=oxide seg.0=30 +
 seg.e=0.6 trans.0=0.1 trans.e=0
impurity imp=germanium mat=TiSi2 cm.sec dix.0=4.0e-7 dix.e=1.8
impurity imp=germanium /mat=TiSi2 seg.0=0.8 trans.0=1e6 trans.e=2.0
impurity imp=germanium mat=poly /mat=TiSi2 seg.0=0.8 +
 trans.0=1e6 trans.e=2.0
impurity imp=germanium mat=WSi2  dix.0=2.6e-2  dix.e=2.11  cm.sec
impurity imp=germanium mat=silicon /mat=WSi2 seg.0=10 +
 seg.e=0.0 trans.0=3.0e-3  trans.e=0.0
impurity imp=germanium mat=poly /mat=WSi2 seg.0=10 +
 seg.e=0.0 trans.0=3.0e-3  trans.e=0.0
impurity imp=germanium mat=oxide dix.0 =1.05E10 dix.e =4.89
impurity imp=germanium mat=nitride dix.0 =0.0 dix.e =0.0
impurity imp=germanium mat=oxynitri dix.0 =0.0 dix.e =0.0
impurity imp=germanium mat=ambient dix.0 =5.00e13 dix.e =3.95
impurity imp=germanium mat=si   /mat=poly seg.0=1.0 +
 seg.e=0. trans.0=0.1   trans.e=0.
impurity imp=germanium mat=si   /mat=oxid seg.0=30. +
 seg.e=0. trans.0=0.1   trans.e=0.
impurity imp=germanium mat=si   /mat=nit  seg.0=1.0 +
 seg.e=0. trans.0=0.0   trans.e=0.
impurity imp=germanium mat=si   /mat=oxyn seg.0=1.0 +
 seg.e=0. trans.0=0.0   trans.e=0.
impurity imp=germanium mat=si /mat=amb  seg.0=1.0 seg.e=0. +
 trans.0=9.0E5 trans.e=1.99
impurity imp=germanium mat=poly /mat=oxid seg.0=30. seg.e=0. +
 trans.0=0.1   trans.e=0.
impurity imp=germanium mat=poly /mat=nit  seg.0=1.0 seg.e=0. +
 trans.0=0.0   trans.e=0.
impurity imp=germanium mat=poly /mat=oxyn seg.0=1.0 seg.e=0. +
 trans.0=0.0   trans.e=0.
impurity imp=germanium mat=poly /mat=amb  seg.0=1.0 seg.e=0. +
 trans.0=9.0E5 trans.e=1.99
impurity imp=germanium mat=oxid /mat=nit  seg.0=1.0 seg.e=0. +
 trans.0=0.0   trans.e=0.
impurity imp=germanium mat=oxid /mat=oxyn seg.0=1.0 seg.e=0. +
 trans.0=0.0   trans.e=0.
impurity imp=germanium mat=oxid /mat=amb  seg.0=1.0 seg.e=0. +
 trans.0=9.0E5 trans.e=1.99
impurity imp=germanium mat=nit  /mat=oxyn seg.0=1.0 seg.e=0. +
 trans.0=0.0   trans.e=0.
impurity imp=germanium mat=nit  /mat=amb  seg.0=1.0 seg.e=0. +
 trans.0=0.0   trans.e=0.
impurity imp=germanium mat=oxyn /mat=amb  seg.0=1.0 seg.e=0. +
 trans.0=0.0   trans.e=0.
impurity imp=germanium mat=polysili cm.sec dix.0=1.14e-2 +
 dix.e=3.44
method imp=germanium silicon abs.adap=1e14 rel.adap=0.2 +
```

```
 min.spac=0.025
method imp=germanium poly abs.adap=1e15 rel.adap=0.3 +
 min.spac=0.03
method imp=germanium oxide abs.adap=1e16 rel.adap=2.0 +
 min.spac=0.1
```

Figure 6-64 USEIT file, Ge-in-Si.data, defining Ge parameters.

```
$ USEIT file for B diffusion models in SiGe: Ge trapped boron, and/or
$ self-consistant SiGe strain/stress, or fixed strain. These models
$ are based on Lever et.al.'s work ("Boron diffusion across Si-SiGe
$ boundaries", JAP 83, 1988(1998)), and the work by Cowern et.al
$ ( "Diffusion in Strained SiGe", PRL 72, 2585(1994)).

$------ Germanium coefficients -----------
$ USEIT SiGe diffusion model implementation - data for germanium in
$ silicon most of the data is similar to arsenic
$ source ./Ge-in-Si.data

$ Chose model of B diffusion in SiGe
$ mod=mod1    consider Ge trap boron only
$ mod=mod2    consider Ge induced fixed strain only
$ mod=mod3    consider self-consistant SiGe strain/stress
$ mod=mod4    consider both Ge trap and SiGe strain/stress

$ assign name=mod c.v="mod1"; default mod=1

$ B diffusion model

intermed name=PoissR     express=0.28
intermed name=YoungM     express=1.87e12
intermed name=c1         express=0.5*(1-PoissR)/YoungM
intermed name=c2 express=PoissR/YoungM

intermed name=GeMax      express=5e22
intermed name=StrainMax  express=-0.042
intermed name=GeFrac     express=Germanium/GeMax

if ("@mod" == "mod2")
intermed name=StrainSiGe express=GeFrac*StrainMax
elseif ( ("@mod" == "mod3") || ("@mod" == "mod4") )
intermed name=StrainSiGe express=c1*(Sxx+Szz)-c2*Syy
else
intermed name=StrainSiGe express=0
if.end

intermed name=EaStrain   express=-17 ;$Boron dEa per unit strain
intermed name=dEa        express=EaStrain*StrainSiGe
intermed name=DX0        expr=0.0368
intermed name=DBE        expr=3.46+dEa
intermed name=DP0        expr=0.716
intermed name=Db         size=2 minindex=0 +
                         expr="DX0*exp(-DBE/kt),DP0*exp(-DBE/kt)"

intermed name=dEg      express=0.7
intermed name=Ni       expr=3.87e16*exp(-0.605/kt+1.5*log(temp))
intermed name=NiSiGe   express=Ni*exp(GeFrac*dEg/(2*kt))
intermed name=C0       express=3.2e21
intermed name=fTrap    express=1/(1.0+Germanium/C0)

intermed name=Na       express=active(boron)
```

```
intermed name=Nd       express=active(arsenic)+active(phosphorus)
intermed name=Nt       express=(Na-Nd)
intermed name=Q        express=sqrt(Nt^2+4*NiSiGe^2)
intermed name=nelectron express=0.5*(Q-Nt)
intermed name=nhole     express=0.5*(Q+Nt)

$pd.ferm for boron
$intermed name=DbSiGe   express=Db[0]+Db[1]*nhole/Ni

$pd.trans for boron
intermed name=fI      express=0.95
intermed name=DbSiGe  +
 express=(Db[0]+Db[1]*nhole/Ni)*(fI*CI.RATIO+(1-fI)*CV.RATIO)

if ( (“@mod” == “mod2”) || (“@mod” == “mod3”) )
intermed name=Bmobile  express=active(boron)
else
intermed name=Bmobile  express=fTrap*active(boron)
if.end

equation model=sige variable=boron mat=si +
         addtoexp=”div(DbSiGe*(grad(Bmobile)+ +
          Bmobile/Q*(grad(Nt)+2*nelectron*grad(log(NiSiGe)))))”

$ Turn off boron built-in model, turn on USEIT model
impurity imp=boron mat=si ^diffuse
method   model=sige enable init.time=1e-12

if ( (“@mod” == “mod3”) || (“@mod” == “mod4”) )
method st.histo viscoela ^skip.sil
if.end

$ Set silicon intrin.s before SiGe deposition command
```

Figure 6-65 USEIT file, sige-stress-trap.usit, defining the model for boron diffusion with strain effect and Germanium trap effect.

```
$ Carbon interstitial traps USEIT file

impurity imp=cs new
method imp=cs none
impurity imp=ci new diffuse

$ Parameters from Gosele
$ Diffusivity of carbon in silicon
intermed name=Dc  express=1.9*exp(-3.1/kT)

$ Diffusivity of carbon interstitials in silicon
$ Decreased by 1e-4 to improve convergence and speed
intermed name=Dci express=4.4e-4*exp(-0.88/kT)

$ 4*pi*a, where a=2.71A, silicon lattice spacing
assign name=pi4a n.val=3.40e-7

$ CS + I <-> CI
intermed name=kR1 express=@pi4a*diffu(interst)
intermed name=c1star express=ci.star*Dci/Dc
intermed name=R1 express=kR1*(cs*interst-ci*c1star)

$ CI + V <-> CS
intermed name=kR2 express=@pi4a*(diffu(vacancy)+Dci)
intermed name=c2star express=cv.star*Dc/Dci
intermed name=R2 express=kR2*(ci*vacancy-cs*c2star)

equation model=carbon variable=ci mat=si addtoexp=div(Dci*grad(ci))
equation model=carbon variable=cs mat=si init=exists(cs)?cs:1e15 +
addtoexp=-R1+R2
equation model=carbon variable=ci mat=si init=ex-
ists(ci)?ci:1e15*Dc/Dci +
addtoexp=R1-R2
equation model=carbon variable=interst mat=si addtoexp=-R1
equation model=carbon variable=vacancy mat=si addtoexp=-R2

$ Use diffusion limited IV recombination
interst kb.high cm.sec kb.0=2*@pi4a*3.65e-4 kb.e=1.58

method model=carbon ^enable
```

Figure 6-66 USEIT file, carbon.usit, defining the model for Carbon trap effect.

After seed Silicon layer deposition, thin SiGe layers are deposited with varying Germanium compositions. The epitaxial layers are in-situ doped with varying Boron, Phosphorus and Carbon. The Carbon profile is imported through USEIT file, import-carbon.usit, as shown in Figure 6-67. The SiGe region is capped with thin silicon and oxide layers.

```
$ Add 1e20 carbon to sige layer

intermed name=Cprofile   expr=” y<@YtopC? 0.0:
                               (y<@YbotC? 1e20:  0.0)”
select  z=Cprofile
print.1 x.v=0 out.file=mycarbon.profile
profile impurity=cs in.file=mycarbon.profile
```

Figure 6-67 USEIT file, import-carbon.usit, defining Carbon profile.

When a SiGe layer is deposited on a relaxed Si seed layer, the lattice constant mismatch introduces an in-plane mismatch strain. This strain in turn introduces an intrinsic stress in the SiGe layer. In this example, the mismatch strain is computed according to Vegard’s law. TSUPREM-4 first checks the local Germanium concentration and converts it to the mol fraction. According to Vegard’s law, in the absence of relaxation via dislocations, the in-plane strain stn is related to the Ge fraction x by stn = -0.042x. For a thin layer with top surface free, the intrinsic stress sts is related to the stn by sts = stn(1-pr)/ym where pr is the Poisson’s ratio and ym is the Young’s modulus. TSUPREM-4 then solves the stress equations to reach equilibrium.

The input statements for above steps are listed in Figure 6-68.

```
$ Chose model of B diffusion in SiGe
$ mod=mod1    consider Ge trap boron only
$ mod=mod2    consider Ge induced fixed strain only
$ mod=mod3    consider self-consistant SiGe strain/stress
$ mod=mod4    consider both Ge trap and SiGe strain/stress
assign name=mod c.v=”mod3”

$ Source USEIT files for boron diffusion in SiGe and carbon traps
source Ge-in-Si.data
source sige-stress-trap.usit
source carbon.usit

$ Predeposit Si
deposit si thick=0.1 phosph=1e18 spaces=8
diff temp=750 time=10

$ SiGe 50A, Ge=5e21
$ Set SiGe intrin.s coefficient
$ Set SiGe intrinsic stress from SiGe strain with the assumption
$ that SiGe layer is stress-free in y direction during deposition
assign name=intrinScoeffSiGe n.v=-(0.042*1.87e12/(1-0.28)/5e22)
assign name=Cge n.val=5e21
assign name=intrinSinit n.val=@Cge*@intrinScoeffSiGe
material silicon intrin.s=@intrinSinit
deposit mat=silicon thick=0.005 boron=1e17 phosph=2e17 spaces=2 +
        impurity=germanium i.con=@Cge
diff temp=700 time=0.25

$ Turn on carbon model, deposit 500A SiGe with carbon
method model=carbon enable
assign name=Cge n.val=1e22
assign name=intrinSinit n.val=@Cge*@intrinScoeffSiGe
material silicon intrin.s=@intrinSinit
$ Extract bottom of carbon doped layer
extract name=YbotC silicon x=0 dist=0 y.ext ^write assign
deposit mat=silicon thick=0.01 boron=5e17 phosph=2e17 spaces=2 +
        impurity=germanium i.con=@Cge
deposit mat=silicon thick=0.015 boron=1e18 phosph=2e17 spaces=3 +
        impurity=germanium i.con=@Cge
deposit mat=silicon thick=0.025 boron=1e19 phosph=2e17 spaces=5 +
        impurity=germanium i.con=@Cge
$ Extract top of carbon doped layer
extract name=YtopC silicon x=0 dist=0 y.ext ^write assign
$ Import carbon profile
source import-carbon.usit
diff temp=700 time=1.5

$ SiGe 50A
assign name=Cge n.val=7.5e21
assign name=intrinSinit n.val=@Cge*@intrinScoeffSiGe
material silicon intrin.s=@intrinSinit
deposit mat=silicon thick=0.005 boron=1e19 phosph=2e17 spaces=2 +
        impurity=germanium i.con=@Cge
diff temp=700 time=0.25

$ SiGe 50A
assign name=Cge n.val=5e21
assign name=intrinSinit n.val=@Cge*@intrinScoeffSiGe
material silicon intrin.s=@intrinSinit
deposit mat=silicon thick=0.005 boron=1e19 phosph=2e17 spaces=2 +
       impurity=germanium i.con=@Cge
diff temp=700 time=0.25
```

```
$ SiGe 50A
assign name=Cge n.val=2.5e21
assign name=intrinSinit n.val=@Cge*@intrinScoeffSiGe
material silicon intrin.s=@intrinSinit
deposit mat=silicon thick=0.005 boron=1e19 phosph=2e17 spaces=2 +
        impurity=germanium i.con=@Cge
diff temp=700 time=0.25
```

Figure 6-68 Second part of sige_hbt.inp, for using SiGe related models.

The in-plane stress Sxx distribution is shown in Figure 6-69.

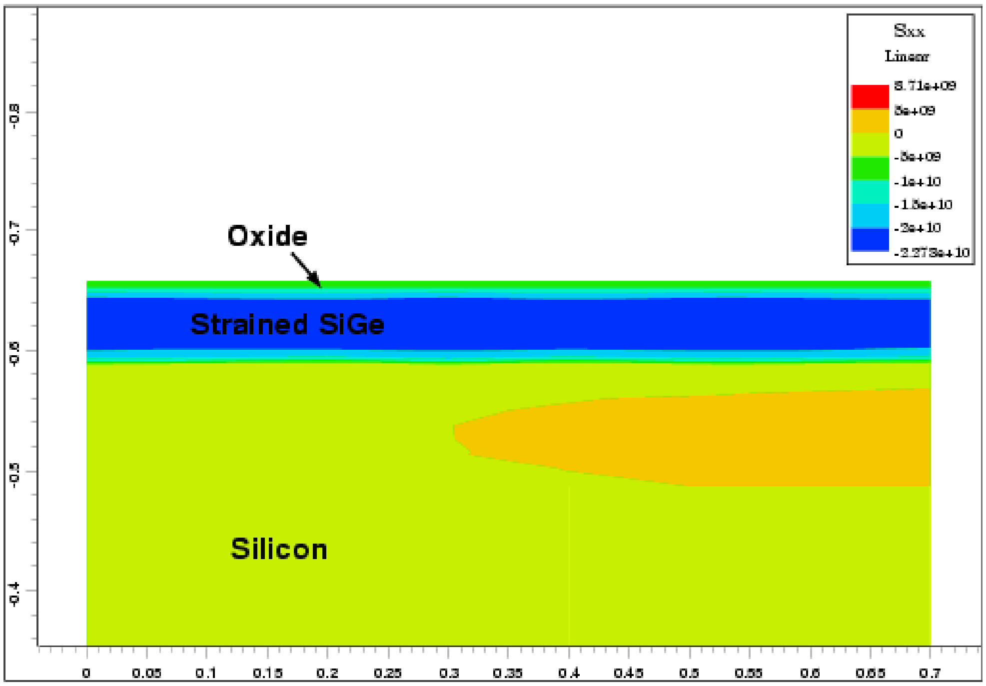


Figure 6-69 The in-plane stress Sxx distribution around SiGe region.

Base and Emitter

On top of the strained SiGe layers, several deposition, etching, implantation and diffusion steps are performed to make the extrinsic base and Emitter. The last steps in the flow fabricate the base and emitter contacts. The input statements for above steps are listed in Figure 6-70 and the top portion of the final structure is shown in Figure 6-71.

```
$ Reset intrin.s
material silicon intrin.s=0

$ Deposit Si cap
deposit mat=silicon thick=0.02 boron=1e12 spaces=10
diff temp=750 time=1

$ Deposit oxide
deposit oxide thick=0.01 spaces=3

$ Emitter_mask
deposit photoresist thickness=1.5 spaces=2

$ Emitter etch
extract name=topsi silicon x=0 dist=0 y.ext ^write assign
assign name=ytop1 n.val=@topsi+0.01
assign name=xbase n.val=@bwidth+0.1
etch photoresist start x=@xbase y=-10
etch cont x=@xbase y=@ytop1
etch cont x=10 y=@ytop1
etch cont x=10 y=-10
etch done x=@xbase y=-10

$ Extrinsic base implant
assign name=doseExtr n.val=5e+15
assign name=eExtr n.val=15
implant boron dose=@doseExtr energy=@eExtr tilt=7 rot=0
implant boron dose=@doseExtr energy=@eExtr tilt=-7 rot=0
etch photoresist all

$ Oxide etch
etch oxide left p1.x=@ewidth

$ Deposit Poly
deposit poly thick=0.15 spaces=5

$ Emitter poly implant
assign name=doseEmi n.val=2e16
assign name=eEmi n.val=20
implant arsenic dose=@doseEmi/2 energy=@eEmi tilt=-7 rotation=0
implant arsenic dose=@doseEmi/2 energy=@eEmi tilt=7 rotation=0

$ Emitter etch
assign name=ew2 n.val=(@ewidth+0.01)
etch poly right p1.x=@ew2

$savefile out.file=sige_hbt0.tif tif

$ Emitter drive
assign name=TempEm n.val=1000
assign name=TimeEm n.val=0.5
diff time=0.05 temp=800 t.fin=@TempEm dry
diff time=@TimeEm temp=@TempEm dry
diff time=0.05 temp=@TempEm t.fin=800

$ Base contact
assign name=xele n.val=(@bwidth+0.05)
etch oxide right p1.x=@bwidth
etch silicon thick=0.02 dry
deposit mat=TiSi2 thick=0.02 spaces=2
etch mat=TiSi2 left p1.x=@xele

$ Emitter contact
```

```
extract name=toppoly poly x=0 dist=0 y.ext ^write assign
assign name=ytop1 n.val=@toppoly+0.05
etch oxide start x=0 y=-10
etch cont x=0 y=@ytop1
etch cont x=10 y=@ytop1
etch cont x=10 y=-10
etch done x=0.0 y=-10
etch poly thick=0.05 dry
deposition aluminum thickness=0.02
etch aluminum right p1.x=@ewidth

$ Truncate bottom
structure truncate bot y=0.5

$ Rename contact
assign name=xbase n.val=(@xele+0.02)
extract name=topbase mat=TiSi2 X=@xbase dist=0 y.ext ^write assign
assign name=ybase n.val=@topbase+0.01
electrode X=@xbase Y=@ybase name=base
extract name=topEm aluminum x=0 dist=0 y.ext ^write assign
assign name=yEm n.val=@topEm+0.01
electrode X=0.05 Y=@yEm name=emitter

savefile out.file=sige_hbt.tif tif
```

Figure 6-70 Third part of sige_hbt.inp, for generating the base and emitter.

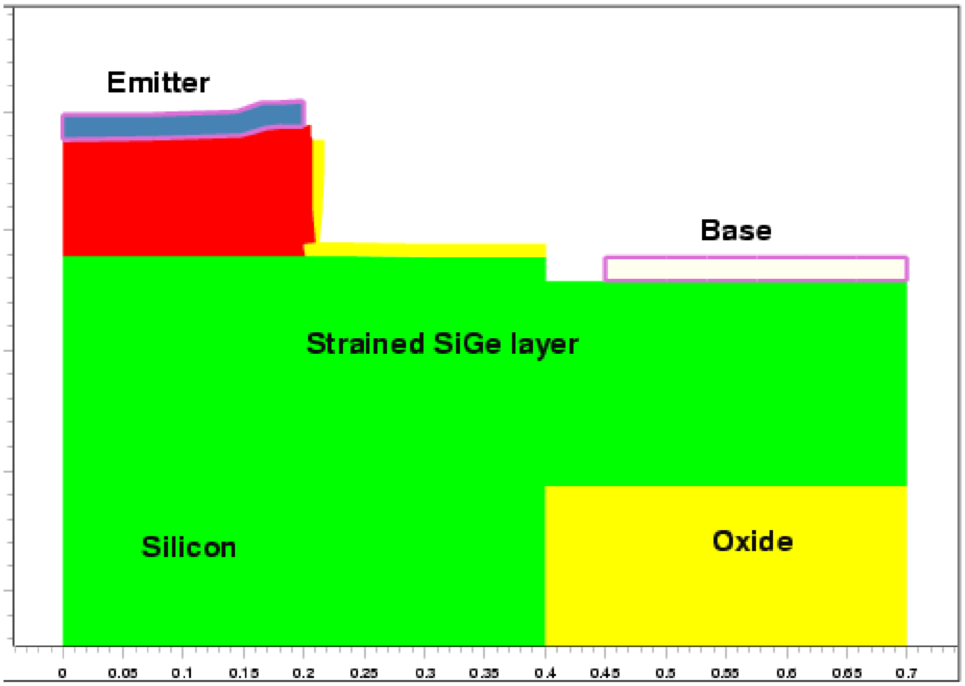


Figure 6-71 The top portion of the final SiGe HBT structure.

User-Specified Equation Interface

The User-Specified Equation Interface (USEIT) is an optionally available capability for TSUPREM-4 that helps you quickly develop and evaluate your own process models, without requiring the program to be recompiled and linked. USEIT supports Synopsys's Physical Model and Equation Interface (PMEI) standard.

The chapter includes the following sections:

- Overview
- Using USEIT
- Examples

Overview

You can use USEIT to specify model equations that have the general form:

$$\frac{\partial X}{\partial t} = \mathit{Previous\_RHS} + \mathbf{ADDTOEXP} + \mathbf{INTERFAC}\ \text{at specified boundaries}$$

where *X* is the quantity to be solved.

TSUPREM-4 assembles the matrices to solve equations by assuming that the time derivative term of *X* is placed at the left-hand-side (LHS) of the equation and all the other terms at the right-hand-side (RHS). To describe this equation using USEIT, any new terms that appear on the RHS of the equation are specified by the **ADDTOEXP** (alias **ADD.RHS**) parameter and the **INTERFAC** parameter on the **EQUATION** statement. The **INTERFAC** parameter is used to define the flux at interfaces.

The term *Previous_RHS* in the above expression represents terms that are already present on the RHS of the equation being modified. The initial condition of the variable *X* is specified with the **INITIAL** parameter or the **IMPL.INI** parameter in the **EQUATION** statement. The **INTERMED** statement specifies the intermediate quantities associated with equations.

Since USEIT (Figures 7-1) is designed as a part of TSUPREM-4, you can easily use the physical quantities such as **CI.STAR**(=I^*) and **ETA**(n/n_i) already calculated in the built-in routines. Similarly, you have the choice of using the built-in functions for calculating active and mobile concentrations or defining your own functions with the **ACTIVE** and **MOBILE** parameters in the **IMPURITY** statement.

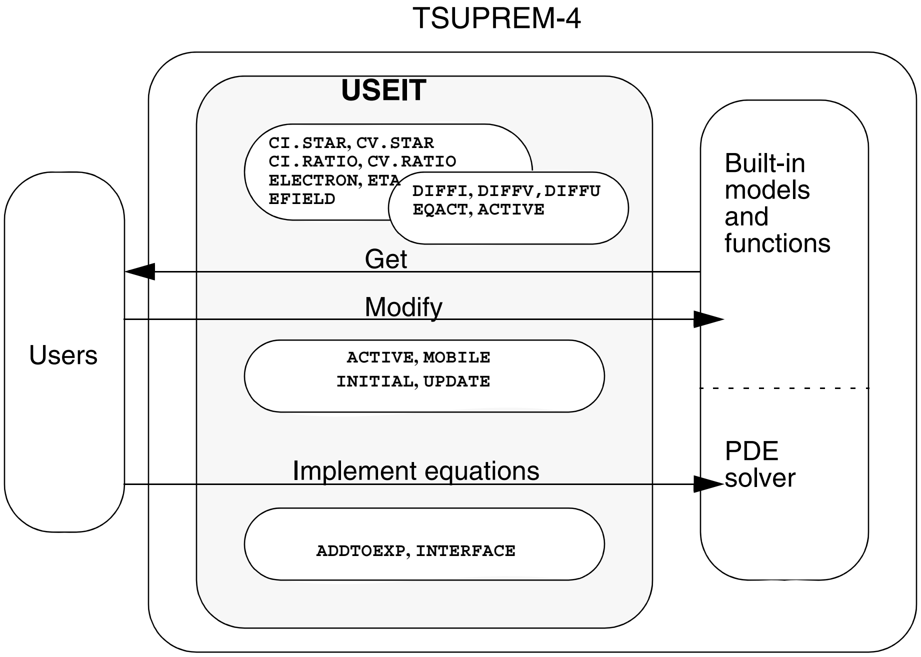


Figure 7-1 USEIT working diagram

Using USEIT

This section provides the following details on using USEIT.

- Additional details of the **EQUATION** statement and its advantages.
- Additional details of the **INTERMEDIATE** statement and its advantages.
- Details of some applications of user-specified models.
- Hints on defining active or mobile concentrations in USEIT.

This section also contains useful tables describing:

- Built-in keywords
- Operators
- Mathematical functions
- Physical functions
- Interface functions
- Miscellaneous functions

EQUATION

The **EQUATION** statement is used to specify user-specified equations, and is available only with the USEIT Option. For additional details, see EQUATION, p. 4-230.

New Solution Variable

When a new solution variable is first specified in the **EQUATION** statement, it is automatically defined. A warning message is displayed explaining that the **VARIABLE** does not currently exist and is being defined. Alternatively, the solution variable can be defined in the **IMPURITY** statement with the **NEW** parameter. The following statement defines the new impurity *BCL*.

```
IMPURITY IMP=BCL NEW
```

Solving Method

The **METHOD** statement specifies the numerical methods used to solve the equations. The factorization method of block matrices, the absolute, and the relative error tolerances for solutions should be specified. For example:

```
METHOD VARIABLE=RDL NONE ABS.ERR=0.1
```

where:

- *RDL* is a solution variable for dislocation-loop radius.
- The **VARIABLE** parameter is an alias for the **IMPURITY** parameter.

Since the *RDL* is supposed to be immobile, the factorization method is defined to **NONE**. The absolute error is set to 0.1. The unit of the absolute error is the same as the unit of the solution variable. The absolute error value is set to 1 x 10^9 by default, unless it is specified. Since the **REL.ERR** parameter is not specified, the relative error is set to 0.01, by default.

Initialization

For solutions (for example, cluster) not introduced by external sources (such as implantation, predeposition, or epitaxy), the solution variable must be initialized to solve the partial differential equations. The **INITIAL** (alias: **DIFF.INI**) parameter or the **IMPL.INI** parameter in the **EQUATION** statement specifies the initial values of a solution variable in the specified.

The initialization by the **INITIAL** parameter is performed when the following conditions are satisfied:

- The solution variable has not yet been initialized before thermal processes (**DIFFUSION** or **EPITAXY**), or the solution variable was initialized and implantation was performed prior to thermal process.
- All other solution variables associated with the **INITIAL** expression are available, so that the **INITIAL** expression can be evaluated; or the solution variable is produced by reactions in the **REACTION** statement or the **INTERFACE** parameter in the **EQUATION** statement.

The parameter **IMPL.INI** defines the expression for the solution initialization at implant. The **IMPL.INI** expression is evaluated and applied at every implant step.

```
EQUATION VARIABLE=RDL MAT=Si +
 INIT=”(1e20<inters+damage&&inters+damage<=1.15e22)?+
       300:(EXIST(RDL)?RDL:0.0)”
```

The example shows the initialization of the dislocation-loop radius. For additional details, see the Dislocation-Loop Model, p. 7-17.

```
IMPLANT ARSENIC DOSE=1E15 ENERGY=30 ;$1
SELECT Z=RDL; PRINT.1D X.V=0        ;$2
DIFFUSION TEMP=800 TIME=0           ;$3
SELECT Z=RDL; PRINT.1D X.V=0        ;$4
```

In the second line, the `RDL` solution has not yet been initialized, so a warning message is displayed, explaining that `RDL` does not currently exist. The `RDL` is initialized at the third line, and then the spatial distribution of `RDL` is printed out at the fourth line.

Note:

The intermediates in the **`INITIAL`** ***expression store the values at the previous thermal step. Thus, if the initialization includes the intermediates associated with the built-in point-defects, the initialization may be incorrect because the changes of the point-defect concentrations due to implants have not been reflected in the intermediates. For example:***

```
INTERMED NAME=Itotal VALUE=damage+interst
EQUATION VARIABLE=RDL MAT=Si +
         INIT="(1e20<Itotal&&Itotal<=1.15e22)? +
         LRAD:(EXIST(RDL)?RDL:0.0)"
```

> ***The damage is zero at the previous thermal step, so that the*** `Itotal` ***stores the only interstitial concentration at the previous thermal step. Thus, the above initialization is incorrect. You can obtain the correct initialization by avoiding the use of*** **`INTERMED`** ***values in the*** **`INITIAL`** ***expression.***

Example for Initialization

When **`DIFF.INI`** for interstitial is defined, the “total” interstitial concentration is reset to the values of the expression evaluated after regrowth. Then the “free” interstitial is calculated and stored to the interstitial solution. The example shows the method to initialize the interstitial concentration after regrowth. The example increases the interstitial concentration two times more than that of the implanted arsenic concentration.

First, the expression for the total interstitial concentration is described to the intermediate *ITOT*. The *MYDAM* and *ASIMP* intermediates store **`DAMAGE`** and **`DIMPLANT`**, respectively. **`DIMPLANT`** stores the newly added dopant concentration by implantation.

```
$ DEFINE POINT-DEFECT MODEL AND MESH
METHOD PD.FULL
MESH DY.SURF=0.001 DY.ACT=0.01 LY.ACT=0.5

$ DEFINE PARAMETERS FOR AMORPHOUS/CRYSTALLINE REGIONS
ASSIGN   NAME=DAMGRAD  N.V=10.0
ASSIGN   NAME=MAXDAM   N.V=1.15E22
ASSIGN   NAME=DPLUS    N.V=2.0

$ DEFINE SMOOTHING FACTOR AT A/C BOUNDARY
INTERMED NAME=MYDAM    EXPR=DAMAGE
INTERMED NAME=ASIMP    EXPR=DIMPLANT
INTERMED NAME=EXPA     +
         EXPR=EXP(@DAMGRAD*LOG(MYDAM/@MAXDAM))
INTERMED NAME=EXPAM EXPR=1/EXPA
INTERMED NAME=TANHA EXPR=(EXPA-EXPAM)/(EXPA+EXPAM)
INTERMED NAME=FS    EXPR=0.5-0.5*TANHA
INTERMED NAME=ITOT  EXPR=CI.STAR+FS*(@DPLUS*ASIMP)
```

The intermediate *ITOT* is assigned to **DIFF.INI** for interstitial in the **EQUATION** statement. Then arsenic implantation and 800^{o}C zero time diffusion are given to show the interstitial profile after regrowth.

```
EQUATION MODEL=IEQINI  IMP=INTERS MAT=SI DIFF.INI=ITOT

$ USE DIFF.INI FOR I INITIALIZATION
METHOD  MODEL=IEQINI ENABLE
INIT
IMPLANT ARSENIC DOSE=1E15 ENERGY=20
DIFFUSE TEMP=800 TIME=0
SELECT Z=LOG10(INTERS)
PLOT.1 X.V=0 TOP=21 BOT=15 RIGHT=0.5 COLOR=2
```

Adding Expressions to Equations

The terms specified by the **ADDTOEXP** parameter in the **EQUATION** statement are added to the current right-hand-side (RHS) term of equations.

```
EQUATION VARIABLE=X MAT=Si ADDTOEXP=DIV(dif*GRAD(X))
```

The above statement represents the following equation for the unknown X:

$$\frac{\partial X}{\partial t} = Previous\_RHS + \nabla \cdot \ (\texttt{dif} \nabla X)$$

where *Previous_RHS* means the previously specified expressions or the built-in terms in the right-hand-side (RHS).

An additional specification of **ADDTOEXP** for the same solution variable in the same material adds the new expression to the previously specified terms; it does *not* replace the present expression with the new expression. For example:

```
EQUATION VARIABLE=X MAT=Si ADDTOEXP=expr1
EQUATION VARIABLE=X MAT=Si ADDTOEXP=expr2
```

The final expression is the sum of expressions *expr1* and *expr2*. You can replace the above statements with:

```
EQUATION VARIABLE=X MAT=Si ADDTOEXP=expr1+expr2
```

Flux at Interfaces

The parameter **INTERFAC** in the **EQUATION** statement specifies the transport flux from material **/MAT2** to material **MAT1** at interfaces between the materials. The specified flux is added at the boundary nodes on the material **MAT1**. The parameter **TRANSPOR** specifies that the flux should also be subtracted at the boundary nodes on material **/MAT2**, so that the dose of the solution is conserved. The **TRANSPOR** parameter is turned *on* by default (see Figures 7-2).

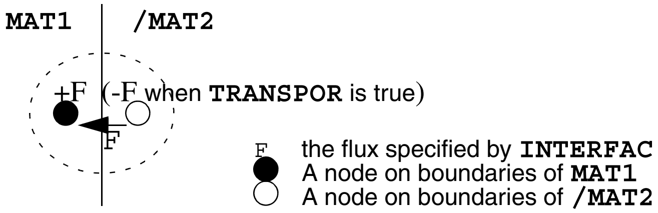


Figure 7-2 Flux at interfaces

```
$ B segregation at Si/Ox
INTERMED NAME=m VALUE=1126*exp(-0.91/kt)
EQUATION VARIABLE=BORON MAT1=Si /MAT2=Ox +
         INTERFAC=0.1*(ONMAT2(BORON)-ONMAT1(BORON)/m)
```

The traditional segregation flux of boron between silicon and oxide is described. No specification of **TRANSPOR** implies that the flux must be subtracted on the boundary nodes in oxides.

The additional specification for the same solution variable on the same interface adds the new flux to the previously specified terms; it does *not* replace the present flux with the new flux. For example:

```
EQUATION VARIABLE=X MAT1=Si /MAT2=Ox INTERFAC=expr1
EQUATION VARIABLE=X MAT1=Si /MAT2=Ox INTERFAC=expr2
```

The final flux is the sum of expressions *expr1* and *expr2*. You can replace the above statements with:

```
EQUATION VAR=X MAT1=Si /MAT2=Ox INTERFAC=expr1+expr2
```

Diffusion Along Boundaries

If the **ADDTOEXP** expression is specified with the **BOUNDARY** parameter in the **EQUATION** statement, the **ADDTOEXP** expression is applied only on the boundaries of **MAT1**, adjacent to **/MAT2**.

```
$ Diffusion on the silicon surface along the Si/Ox.
EQUATION VARIABLE=IBORON MAT1=Ox /MAT2=Si BOUNDARY +
         ADDTOEXP=DIV(dif*GRAD(IBORON))
```

The expression in the **ADDTOEXP** parameter shows the diffusion along the boundaries of oxide adjacent to silicon.

Update Solution

Use the parameter **UPDATE** in the **EQUATION** statement for the special case of updating the solution during solving equations. For example:

```
$ RDL : Dislocation loop radius (angstrom unit)
EQUATION VARIABLE=RDL MAT=Si UPDATE=RDL>100?RDL:0.0
```

This example shows how to discard dislocation-loops with a radius of less than

100 angstroms. When solving the equation, the values of *RDL* less than 100 angstroms are set to zero.

INTERMEDIATE

The **INTERMEDIATE** statement (alias **MODEL** or **ARRAY**) specifies the intermediate quantities associated with equations. For additional details, see INTERMEDIATE, p. 4-30.

Advantages of Intermediates

The advantages of using intermediates quantities are:

- Simplifying long and complicated equations
- Avoiding the duplicated calculation of quantities used in several equations
- Once the intermediates are evaluated, the calculated values are shared by all associated equations until the intermediates need to be reset.
- Avoiding duplicated calculation of quantities depending *only* on temperature
- If the intermediates depend *only* on temperature, they are evaluated *only* when the temperature changes.
- Using intermediates in the same manner as the built-in solution values in the **SELECT** statement.

Note:

Intermediates are stored as symbolic expressions that are evaluated later (for example, when solving the model equations).

Application of User-Specified Models

The **MODEL** parameter controls the application of specified equations. By turning *on/off* the **ENABLE** parameter with the **MODEL** parameter in the **METHOD** statement, the equations grouped by the **MODEL** parameter in the **IMPURITY** and **REACTION** statements can be enabled or disabled. The models not specified by the **MODEL** parameter are always turned *on*.

```
EQUATION MODEL=BIC VARIABLE=BORON MAT=Si ADDTOEXP=rhs1
EQUATION MODEL=BIC VARIABLE=INTER MAT=Si ADDTOEXP=rhs2
$ process1
METHOD   MODEL=BIC ^ENABLE
$ process2
METHOD   MODEL=BIC ENABLE
$ process3
```

At *process1*, the equations in the model *BIC* are applied because it is turned *on* by default. Turning *off* the **ENABLE** parameter prior to *process2* disables the *BIC* model. Simulate *process3* with the *BIC* model by turning *on* the **ENABLE** parameter again.

The additional specification for the same solution variable in the same material or on the same interface with the same model name adds the new expression to the previously specified terms; it does *not* replace the present expression with the new expression. For example:

```
EQUATION MODEL=mdl VARIABLE=X MATERIAL=m EXPR=expr1
EQUATION MODEL=mdl VARIABLE=X MATERIAL=m EXPR=expr2
```

The final expression is the sum of expressions *expr1* and *expr2*. You can replace the above statements with:

```
EQUATION MODEL=mdl VAR=X MAT=m EXPR=expr1+expr2
```

Specify the **INITIAL**, **ACTIVE**, **MOBILE**, and **UPDATE** parameters only once for the specified impurity and material in one model. The duplicated specification of these parameters for the same model replaces the previous definitions with the new ones. Even though it is possible for each different model to have its own **INITIAL**, **ACTIVE**, **MOBILE**, and **UPDATE** expressions, only one model among them should be enabled. Otherwise, incorrect results occur.

Modification of Built-in Functions

Define the modification of the active or mobile concentrations in USEIT using the following hints.

ACTIVE Usually, active concentrations are defined with the solid solubilities and the equilibrium clustering coefficients specified in the **IMPURITY** statement. When active concentrations need to be calculated differently, the **ACTIVE** parameter in the **IMPURITY** statement provides the new method to calculate active concentrations. The physical function **ACTIVE()** returns the values of active concentration calculated by the method that is specified in the **ACTIVE** parameter, while the **ACTIVE** parameter defines the calculation method itself for active concentration.

```
IMPURITY IMP=BCL NEW
INTERMED NAME=Ceq EXPRESS=EQACTIVE(boron,boron)
INTERMED NAME=ni  +
         EXPRESS=3.87e16*exp(-0.605/kt+1.5*log(temp))
IMPURITY IMP=boron MAT=Si +
         ACTIVE=MAX(boron-BCL,MIN(Ceq,ni))
```

The *Ceq* and *ni* are defined as the equilibrium active boron concentration and intrinsic carrier concentration, respectively.

MOBILE In most cases, mobile concentrations are the same as active concentrations. However, when the mobile concentration is different from the active concentration, use the **MOBILE** parameter in the **IMPURITY** statement. For example, the diffusion retardation due to ion-pairing is described with the reduction of the mobile concentration.

$$C_m = \left(1 - \frac{N_p}{N_a}\right)C_a \text{ for acceptors, } C_m = \left(1 - \frac{N_p}{N_d}\right)C_a \text{ for donors}$$

$$\text{where } N_p = \frac{1}{2}((N_d + N_a + \Omega) - \sqrt{(N_d + N_a + \Omega)^2 - 4N_dN_a})$$

The above equations are described using USEIT as follows:

```
INTERMED NAME=ni   +
         EXPR=3.87e16*exp(-0.605/kt+1.5*log(temp))
INTERMED NAME=Nd  EXPR=ACTIVE(arsenic)+ACTIVE(phospho)
INTERMED NAME=Na   EXPR=ACTIVE(boron)
INTERMED NAME=Nt   EXPR=Nd+Na+6.0*ni
INTERMED NAME=Np   EXPR=0.5*(Nt-sqrt(Nt*Nt-4*Nd*Na))
INTERMED NAME=Fpa EXPR=1-Np/Na
INTERMED NAME=Fpd EXPR=1-Np/Nd
IMPURITY MODEL=IONPAIR IMP=boro MAT=Si +
         MOBILE=Fpa*ACTIVE(boron)
IMPURITY MODEL=IONPAIR IMP=phos MAT=Si +
         MOBILE=Fpd*ACTIVE(phos)
IMPURITY MODEL=IONPAIR IMP=arse MAT=Si +
         MOBILE=Fpd*ACTIVE(arse)
```

Built-in Keywords

The following keywords are available.

Name	Type	Value	Unit
KB	constant	Boltzmann constant, 1.38066×10^{-23}	J/^{o}K
QE	constant	Electron charge, 1.602×10^{-19}	Coulomb
TEMP	parameter	Kelvin temperature	^{o}K
KT	parameter	Thermal voltage, **KB** × **TEMP**	Volts
CI.STAR	node solution	Equilibrium interstitial concentration	cm^{-3}
CV.STAR	node solution	Equilibrium vacancy concentration	cm^{-3}
CI.RATIO	node solution	Ratio of interstitial concentrations to equilibrium	None
CV.RATIO	node solution	Ratio of vacancy concentrations to equilibrium	None
ELECTRON	node solution	Electron concentration ***Note*:** Calculated by assuming local charge neutrality	cm^{-3}
ETA	node solution	Electron concentration ratio to intrinsic carrier concentration	None
EFIELD	edge solution	Electric field divided by thermal voltage	cm^{-1}

Operators

The operators in higher tables have higher precedence than the operators in lower tables. *expr* means an expression.

Name	Type	Value
^	*expr* ^ *expr*	Power **Example:** `CI.RATIO^2`
+	+*expr*	Unary positive
-	-*expr*	Unary negative
*****	*expr* * *expr*	Multiply
/	*expr* / *expr*	Divide
+	*expr* + *expr*	Add
-	*expr* - *expr*	Subtract
>	*expr* > *expr*	Greater than
>=	*expr* >= *expr*	Greater than or equal to
<	*expr* < *expr*	Less than
<=	*expr* <= *expr*	Less than or equal to
==	*expr* == *expr*	Equal
!=	*expr* != *expr*	Not equal
&&	*expr* && *expr*	And
||	*expr* || *expr*	Or
?:	*expr* ? *expr* : *expr*	*condition_expression* **?** *true_expression* **:** *false_expression* **Example:** `BORON>PHOS?BORON:PHOS` ***Note*:** Same as the `CONDITION` function. See Miscellaneous Functions, p. 7-15.

Mathematical Functions

The argument types, *number*, *node value*, and *edge value* in the argument list mean that the evaluation of the argument expressions results in number, node solution, and edge solution, respectively.

Function	Argument List		Return Value
LOG	*arg1*	number or node value	Natural logarithmic value Example: `LOG(10)` Example: `LOG(INTERST+DAMAGE)`
LOG10	*arg1*	number or node value	10-based logarithmic value
EXP	*arg1*	number or node value	Exponential value
ERF	*arg1*	number or node value	Error function value

Function	Argument List		Return Value
ERFC	*arg1*	number or node value	Complementary error function value
ABS	*arg1*	number or node value	Absolute value
SQRT	*arg1*	number or node value	Square root value
MIN	*arg1* : *argN* ***Note***: *N* must be larger than 1	number or node value number or node value	Smallest value **Example:** `MIN(1,2,3)` **Example:** `MIN(BORON+PHOS,1E21)`
MAX	*arg1* : *argN* ***Note***: *N* must be larger than 1	number or node value number or node value	Largest value **Example:** `MAX(1,2,3)` **Example:** `MAX(BORON,PHOS+ARSENIC)`

SUM	*arg1*
	arg2
	arg3
	arg4
		index
		number
		number
		number or node value
			Summed value
			Example: $\sum_{i=1}^{10} i^2$ is described as
			`SUM(i,1,10,i^2)`
			Example: $\sum_{i=-2}^{2}\sum_{j=0}^{5} k_{ij}$ is described as
			`SUM(i,-2,2,SUM(j,0,5,k[i][j]))`

SOLVE	*arg1*
	arg2
	arg3
		variable name
		expression
		initial value
			Extract the arg1's value, which makes the expression equal to zero by Newton-Raphson iteration
			Example: Extract *x* near 1.0 from the equation
			$x^2 - 3x + 1 = 0$
			`SOLVE(x,x^2-3*x+1,1.0)`
			Example: Extract C_a satisfying
			$C_{as} = C_a + \left(1\times10^{-17}\exp\left(\frac{0.4}{kT}\right)C_a\right)^{4.0}$
			`INTERM NAME=CT EXPR=1E-17*exp(0.4/kT)`
			`SOLVE(Ca,Ca+(CT*Ca)^4.0-Cas, Cas)`

GRADIENT	*arg1*	number or node value	Gradient vector **Example:** $\vec{\nabla}C$ is described as `GRAD(C)` ***Note***: Only allowed for the parameters **ADDTOEXP** and **INTERFAC** in the **EQUATION** statement.
DIVERGENCE	*arg1*	edge value	Divergence **Example:** $\vec{\nabla}\cdot\left(\vec{\nabla}C + C\frac{q\vec{E}}{kT}\right)$ is described as `DIV(GRAD(C)+C*EFIELD)` ***Note***: Only allowed for the parameters **ADDTOEXP** and **INTERFAC** in the **EQUATION** statement.

Function		Argument List	Return Value
DDT	*arg1*	number or node value	Time derivative value **Example:** $\frac{\partial}{\partial t}(C+I)$ is described as `DDT(C+I)` ***Note*:** Only allowed for the parameters **`ADDTOEXP`** and **`INTERFAC`** in the **`EQUATION`** statement.

Physical Functions

The following physical functions are examples of functions available in TSUPREM-4.

Function		Argument List	Return Value
EQACTIVE	*arg1* *arg2*	impurity name number or node value	Equilibrium active concentration of *arg2* expression, calculated with the solid solubility and equilibrium clustering coefficients of the impurity *arg1*. **Example:** `EQACT(BORON,BORON-BCL)`
ACTIVE	*arg1*	impurity name	Active concentration of the impurity *arg1* **Example:** `ACTIVE(BORON)` ***Note*:** May not include the functions **`DIFFI`**, **`DIFFV`** and **`DIFFUSIVITY`**.
DIFFI	*arg1*	impurity name	Diffusivity of the impurity *arg1* with interstitials, *i.e.* $\exp\left(-\frac{p\mathbf{VD}}{kT}\right)\cdot\ \mathtt{DI.FAC}\cdot\ (\mathtt{DI.F11}+\mathtt{DI.F22})/\ 2\sum_{i}D_m^{(i)}\left(\frac{n}{n_i}\right)^{-i}$, where $D_m^{(i)}$ values are defined by: **`DIX.0`**, **`DIX.E`**, **`DIP.0`**, **`DIP.E`**, **`DIM.0`**, **`DIM.E`**, **`DIMM.0`**, **`DIMM.E`**, **`DIC.0`**, and **`DIC.E`** parameters in the **`IMPURITY`** statement. **Example:** `DIFFI(BORON)` ***Note*:** Not allowed for the built-in point defects *interst* and *vacancy*
DIFFV	*arg1*	impurity name	Diffusivity of the impurity *arg1* with vacancies, *i.e.:* $\exp\left(-\frac{p\mathbf{VD}}{kT}\right)\cdot\ \mathtt{DV.FAC}\cdot\ (\mathtt{DV.F11}+\mathtt{DV.F22})/\ 2\sum_{i}D_n^{(i)}\left(\frac{n}{n_i}\right)^{-i}$, where $D_n^{(i)}$ values are defined by: **`DVX.0`**, **`DVX.E`**, **`DVP.0`**, **`DVP.E`**, **`DVM.0`**, **`DVM.E`**, **`DVMM.0`**, **`DVMM.E`**, **`DVC.0`**, and **`DVC.E`** parameters in the **`IMPURITY`** statement **Example:** `DIFFV(BORON)` ***Note*:** Not allowed for the built-in point defects *interst* and *vacancy.*

Function	Argument List		Return Value
DIFFUSIVIT	*arg1*	impurity name	Diffusivity of the impurity *arg1*. If *arg1* is neither *interst* and nor *vacancy*, $\frac{I}{I^*}\mathtt{DIFFI(arg1)} + \frac{V}{V^*}\mathtt{DIFFV(arg1)}$, otherwise, $\exp\left(-\frac{p\mathbf{VD}}{kT}\right) \cdot \ \mathbf{D.FACTOR} \cdot \ (\mathbf{D.F11} + \mathbf{D.F22})/\ 2\frac{\sum_i D_I^{(i)} \phi_I^{(i)} \left(\frac{n}{n_i}\right)^{-i}}{\sum_i \phi_I^{(i)} \left(\frac{n}{n_i}\right)^{-i}}$ and $\exp\left(-\frac{p\mathbf{VD}}{kT}\right) \cdot \ \mathbf{D.FACTOR} \cdot \ (\mathbf{D.F11} + \mathbf{D.F22})/\ 2\frac{\sum_i D_V^{(i)} \phi_V^{(i)} \left(\frac{n}{n_i}\right)^{-i}}{\sum_i \phi_V^{(i)} \left(\frac{n}{n_i}\right)^{-i}}$ for *interst* and *vacancy*, respectively. **D.0**, **D.E**, **DC.0**, **DC.E**, and **C.STATE** for D_I and D_V are specified on the **INTERSTITIAL** and **VACANCY** statements, respectively. **Example:** **DIFFU(BORON)**

Interface Functions

The following interface functions are examples of functions available in TSUPREM-4.

Function	Argument List		Return Value
ONMAT1	*arg1*	number or node value	Values for the given expression *arg1* on the boundary of the **MATERIAL** (alias **MAT1**) material adjacent to the **/MATERIA** (alias **/MAT2**) material in the **EQUATION** statement. **Example:** `ONMAT1(PHOS/m)` ***Note*:** Only allowed in the **INTERFAC** parameter in the **EQUATION** statement.
ONMAT2	*arg1*	number or node value	Values for the given expression *arg1* on the boundary of the **/MATERIA** (alias **/MAT2**) material adjacent to the **MATERIAL** (alias **MAT1**) material in the **EQUATION** statement. **Example:** `ONMAT2(PHOS/m)` ***Note*:** Only allowed in the **INTERFAC** parameter in the **EQUATION** statement.

Miscellaneous Functions

The following functions are examples of functions available in TSUPREM-4.

Function		Argument List	Return Value
`CONDITION`	*arg1* *arg2* *arg3*	logic expression number or node value number or node value	*arg2* if *arg1* is true, otherwise *arg3* **Example:** `CONDITION(PHOS>1E21,1E21,PHOS)` ***Note*:** Same as *arg1*?*arg2*:*arg3*. `(PHOS>1E21?1E21:PHOS)`
`EXIST`	*arg1*	solution variable name	true(1) if *arg1* exists in the structure, otherwise false (0). **Example:** `EXIST(BORON)`

Examples

The following USEIT models are detailed in this examples section:

- Interstitial clustering model
- Transient clustering model
- Dislocation-loop model
- Diffusion in **`PD.TRANS`**
- Interface trap model
- Boron-Interstitial clustering model with small clusters
- Diffusion enhancement in thinner oxides
- Trapped nitrogen dependent surface recombination rate
- 5-stream diffusion model
- SiGe model

Interstitial Clustering Model

In the interstitial clustering model:

- Name the new impurity, `ICL`.
- Specify the **`NONE`** parameter in the **`METHOD`** statement since the new impurity `ICL` is supposed to be immobile.
- Set the reaction rate k_{fi} of the initial clustering term to zero.
- Set k_{fc},k_r as intermediates because they depend *only* on temperature.
- Set G as an intermediate because its quantity is used in both `ICL` and interstitial equations.
- Set the **`CL.IFC`**, **`CL.ISFC`**, and **`CL.CR`** values to 1.0.
- Set the **`CL.CF`** value to 0.9398.
- Set the initial background concentration to 10^9/cm^3.

The function call, **`EXIST(ICL)`**, checks to see if the variable `ICL` already exists.

The **`INITIAL`** condition implies that the initialization takes place only when the *`ICL`* solution does not exist.

The change of interstitial concentration due to recombination with clusters must be taken into account by subtracting the *G* term from the interstitial equation.

$$\frac{\partial C}{\partial t} = k_{fi} \cdot \frac{I^{\mathbf{CL.IFI}}}{I^{*\mathbf{CL.ISFI}}} + k_{fc} \cdot \frac{I^{\mathbf{CL.IFC}}}{I^{*\mathbf{CL.ISFC}}} (C+I)^{\mathbf{CL.CF}} - k_r \cdot C^{\mathbf{CL.CR}}$$

```
$------------- Start of implementation --------------
IMPURITY IMP=ICL NEW
METHOD   IMP=ICL NONE
INTERMED NAME=kfc EXPRESS=5.207E14*exp(-3.774/kt)
INTERMED NAME=kr  EXPRESS=9.431E13*exp(-3.017/kt)
INTERMED NAME=G  +
         EXPRESS=kfc*CI.RATIO*(ICL+INTER)^0.9398-kr*ICL
EQUATION MODEL=ICL VARIABLE=ICL    MAT=Si +
         INIT=EXIST(ICL)?ICL:1E9          ADDTOEXP=G
EQUATION MODEL=ICL VARIABLE=INTERS MAT=Si ADDTOEXP=-G
$------------- End of implementation ----------------

METHOD    PD.FULL
IF (1) ;$ use USEIT model
  METHOD  MODEL=ICL ENABLE ;$ turn-on USEIT model
  INTERST MAT=Si ^CL.MODEL ;$ turn-off built-in model
ELSE   ;$ use built-in model
  METHOD  MODEL=ICL ^ENABLE;$ turn-off USEIT model
  INTERST MAT=Si CL.MODEL  ;$ turn-on built-in model
IF.END
```

Transient Clustering Model

Even though *Ca* appears as a solution variable in the model equation, solving the equation for the clustered boron concentration *`BCL`* (=*C*-*Ca*) is easier because the clustered borons are immobile.

- In the transient clustering model, define the intrinsic carrier concentration *ni* and the time constant *ta* for activation.

  The **`EQACTIVE(boron,boron)`** returns the equilibrium active concentration of total boron profile.

- Since the initial active concentration is assumed to be a level comparable to *ni*, define the initial clustered concentration to the value **`MAX(boron-ni,0.0)`** in the **`INITIAL`** parameter.

  If *`BCL`* already exists, it is initialized to the difference between the total boron concentration and the active concentration *Ca*.

The expression of the intermediate *Ca* ensures that the active concentration stays above *ni*.

$$\frac{\partial(C - C_a)}{\partial t} = \frac{C_a - C_a^*}{\tau_a} ,$$

$$C_a \geq \min(C_a^*,\ n_i) \quad \text{for initialized active concentration}$$

```
$------------ Start of implementation -------------
IMPURITY IMP=BCL NEW
METHOD   IMP=BCL NONE
INTERMED NAME=ni +
         EXPRESS=3.87E16*exp(-0.605/kt+1.5*log(temp))
INTERMED NAME=ta  EXPRESS=8.0E-16*exp(4.2/kt)
INTERMED NAME=Ceq EXPRESS=EQACT(boron,boron)
INTERMED NAME=Ca  EXPRESS=MAX(boron-BCL,MIN(Ceq,ni))
IMPURITY MODEL=TCL IMP=boron MAT=Si ACTIVE=Ca
EQUATION MODEL=TCL VARIABLE=BCL   MAT=Si +
         INIT=EXIST(BCL)?boron-Ca:MAX(boron-ni,0.0) +
         ADDTOEXP=(boron-BCL-Ceq)/ta
$------------ End of implementation ---------------

METHOD   PD.FULL
IF (1) ;$turn on USEIT model
  METHOD   MODEL=TCL ENABLE ;$ turn-on USEIT model
  METHOD   ACT.EQUI         ;$ turn-off built-in model
ELSE   ;$turn on built-in model
  METHOD   MODEL=TCL ^ENABLE;$ turn-off USEIT model
  METHOD   ACT.TRAN         ;$ turn-on built-in model
IF.END
```

Dislocation-Loop Model

In the dislocation-loop model:

- Specify the initial volume density and radius of dislocation-loops by the *LDENS* and *LRAD* intermediates, respectively.

  Even though the default values are assumed to be 1 x 10^{15}/cm^3 and 300Å for density and radius, respectively, their values can be changed for different implant conditions.
- Calculate the radius *RDL* in angstroms.
- Define the new variable *RDL* first in the **EQUATION** statement, so that the following warning message will be shown:

```
*** Warning:  rdl does not exist. It will be newly defined.
```

The expression in the **INITIAL** parameter represents that the 300Å dislocation-loops form initially in the place where the total interstitial concentration is in the range between 1 x 10^{20}/cm^3 and 1.15 x 10^{22}/cm^3 while the current dislocation-loops elsewhere are kept.

Since it is assumed that the Frenkel-pair point-defect *damage* and the previous interstitials *interst* are distinguished, the total interstitial concentration is the sum of both. *`LP1`* and *`LP2`* are $\exp(\gamma\Omega/(bkT))$, $\mu b\Omega/(4\pi kT(1-\nu))$, respectively.

`ileq` represents the equilibrium interstitial concentration around the dislocation-loop.

Loops of radius 100Å or less dissolve quickly during diffusion so that such loops are discarded, which is specified by the **`UPDATE`** parameter.

Thus, the time derivative of the loop radius *`RDL`* is taken into account only in the region *`RDL`*>100Å.

In the **`METHOD`** statement, the **`VARIABLE`** parameter is an alias for the **`IMPURITY`** parameter.

Since the *`RDL`* is supposed to be immobile, the factorization method is defined to **`NONE`**.

- Set the absolute error to 0.1. The unit of the absolute error is the same as one of the solution variable. The absolute error value is set to 1 x 10^9 by default, unless it is specified.

  Since the **`REL.ERR`** parameter is not specified, the relative error is set to 0.01 by default.

$$\frac{dr}{dt} = \frac{\pi}{N_0} K_L D_I (I - I^*_{loop})$$

$$R_l = 2\pi^2 K_L D_I \rho_v r (I - I^*_{loop})$$ for the rate of absorption of interstitials by dislocation loops

where $$I^*_{loop} = I^* \exp\left(\frac{\gamma\Omega}{bkT}\right)\exp\left(\frac{\mu b\Omega}{4\pi rkT(1-\nu)}\ln\left(\frac{8r}{b}\right)\right)$$

```
$------------ Start of implementation -------------
INTERMED NAME=LDEN  EXPR=1E15
INTERMED NAME=LRAD  EXPR=300
EQUATION MODEL=DLL VAR=RDL MAT=Si +
         INIT=”(1e20<inter+damage&&inter+damage<=1.15e22)? +
         LRAD:(EXIST(RDL)?RDL:0.0)”
INTERMED NAME=LP1   EXPR=exp(2.783E-2/kt)
INTERMED NAME=LP2   EXPR=2.877/kt
INTERMED NAME=KL    EXPR=29.8047E8*exp(-0.4/kt)
INTERMED NAME=ileq  +
       EXPR=CI.STAR*LP1*exp(LP2/RDL*(log(2.55*RDL)))
INTERMED NAME=KD EXPR=KL*DIFFU(inters)*(inters-ileq)
EQUATION MODEL=DLL VARIABLE=inters  MAT=silicon +
         ADDTOEXP=-1.97e-15*LDEN*RDL*KD
EQUATION MODEL=DLL VARIABLE=RDL MAT=silicon +
         ADDTOEXP=RDL>100?2.00E-15*KD:0.0 +
         UPDATE=RDL>100?RDL:0.0
METHOD   VARIABLE=RDL NONE ABS.ERR=0.1
$------------ End of implementation ---------------

METHOD     PD.FULL
IF (1) ;$ use USEIT model
  METHOD   MODEL=DLL ENABLE
  INTERMED NAME=LRAD EXPR=350
  INTERMED NAME=LDEN EXPR=2E15
  IMPLANT  ARSENIC DOSE=1E15 ENERGY=30
ELSE   ;$ use built-in model
  METHOD  MODEL=DLL ^ENABLE
  IMPLANT ARSENIC DOSE=1E15 ENERGY=30 +
          L.DEN=2E15 L.RAD=350E-8
IF.END
```

Diffusion in PD.TRANS

In the diffusion in **`PD.TRANS`**:

- Use the **`DIVERGENCE()`** and **`GRADIENT()`** functions for divergence and gradient calculations, respectively.

  The **`DIFFUSIVITY(boron)`** is the same as **`CI.RATIO*DIFFI(boron)+CV.RATIO*DIFFV(boron)`**.

  Turning *off* the **`DIFFUSE`** parameter in the **`IMPURITY`** statement disables the built-in diffusion equations for the specified impurity in the specified material.

$$\frac{\partial C}{\partial t} = \vec{\nabla} \cdot \left(\left(\frac{I}{I^*}\sum_i D_m^{(i)}\left(\frac{n}{n_i}\right)^i + \frac{V}{V^*}\sum_i D_n^{(i)}\left(\frac{n}{n_i}\right)^i\right)\left(\vec{\nabla} C_m + C_m \frac{q\vec{E}}{kT}\right)\right)$$

```
$------------- Start of implementation --------------
INTERMED NAME=Db EXPRESS=DIFFUS(boron)
INTERMED NAME=Cm EXPRESS=ACTIVE(boron)
EQUATION MODEL=PDTRANS VARIABLE=boron MAT=silicon +
         ADDTOEXP=DIV(Db*(GRAD(Cm)+Cm*EFIELD))
$------------- End of implementation ----------------

METHOD   PD.TRANS
IF (1) ;$ use USEIT model
  $ turn-off the built-in diffusion equation
  IMPURITY IMP=boron MAT=Si ^DIFFUSE
  METHOD   MODEL=PDTRANS ENABLE
ELSE
  METHOD   MODEL=PDTRANS ^ENABLE
IF.END
```

Interface Trap Model

In the interface trap model:

The **BOUNDARY** parameter in the **IMPURITY** statement specifies that the new impurity exists only on boundaries.

- For phosphorus, set the coefficient values for h_{Si}, h_{ox}, κ_{Si}, κ_{ox} and σ_{max} to $0.715\exp(-1.75\mathrm{eV}/kT)$, $10h_{Si}$, $5882\exp(0.37\mathrm{eV}/kT)$, $\kappa_{Si}/30.0$, and 6.8×10^{14}. respectively.
- Set the diffusivity D_o of trapped phosphorus along interfaces to $1 \times 10^5\exp(-4.44\mathrm{eV}/kT)$.
- Since the interfaces mediate phosphorus between oxide and silicon, you must turn *off* the parameter **TRANSPOR**.

  The function **ONMAT1()** returns the values on the boundaries of **MAT1** adjacent to **/MAT2**.

  The function **ONMAT2()** returns the values on the boundaries of **/MAT2** adjacent to **MAT1**.

  The **BOUNDARY** parameter in the **EQUATION** statement specifies that the diffusion term in the description of the **ADDTOEXP** parameter considers the diffusion of trapped phosphorus along the boundaries of **MAT1** adjacent to **/MAT2**.

$$\frac{\partial\sigma}{\partial t} = -\frac{\partial}{\partial l}\left(D_\sigma\frac{\partial\sigma}{\partial l}\right) + F_{ox} + F_{Si} \quad \text{where} \quad F_i = h_i\left(C_i\left(1 - \frac{\sigma}{\sigma_{max}}\right) - \kappa_i\sigma\right)$$

It is assumed that the detrapping flux from interface to silicon is independent of the surface concentration in silicon and that the competitive occupation for the trap sites is negligible.

```
$------------ Start of implementation -------------
IMPURITY IMP=IPHO NEW BOUNDARY
METHOD   IMP=IPHO NONE ABS.ERR=1.0e5
INTERMED NAME=dip   EXPR=1e5*exp(-4.44/kt)
INTERMED NAME=hsi   EXPR=0.715*exp(-1.75/kt)
INTERMED NAME=hox   EXPR=10*hsi
INTERMED NAME=ksi   EXPR=5882*exp(0.37/kt)
INTERMED NAME=kox   EXPR=ksi/30
INTERMED NAME=qmax  EXPR=6.8E14
EQUATION MODEL=DLOSS VARIABLE=PHOS MAT1=Si /MAT2=Ox  +
         INTERFAC=″-hsi*(ONMAT1(PHOS)*(1-ONMAT2(IPHO)/qmax)+
                 -ksi*ONMAT2(IPHO))″ ^TRANSPOR
EQUATION MODEL=DLOSS VARIABLE=PHOS MAT1=Ox /MAT2=Si  +
         INTERFAC=″-hox*(ONMAT1(PHOS)*(1-ONMAT1(IPHO)/qmax)+
                -kox*ONMAT1(IPHO))″ ^TRANSPOR
EQUATION MODEL=DLOSS VARIABLE=IPHO MAT1=Ox /MAT2=Si  +
         BOUNDARY ADDTOEXP=DIV(dip*GRAD(IPHO)) +
         INTERFAC=″hsi*(ONMAT2(PHOS)*(1-ONMAT1(IPHO)/qmax)+
                -ksi*ONMAT1(IPHO)) +
           +hox*(ONMAT1(PHOS)*(1-ONMAT1(IPHO)/qmax)+
                -kox*ONMAT1(IPHO))″ ^TRANSPOR
$------------ End of implementation ---------------

IF (1) ;$turn on USEIT model
  METHOD   MODEL=DLOSS ENABLE
  METHOD   ^ITRAP
ELSE   ;$turn on built-in model
  METHOD   MODEL=DLOSS ^ENABLE
  METHOD   ITRAP
  IMPURITY IMP=I_PHO MAT=Ox /MAT=Si CM.SEC +
           ^/SEG.SS SEG.EQ3                +
           DIX.0=1E5 DIX.E=4.44            +
           TRANS.0=7.15     TRANS.E=1.75   +
           /TRANS.0=0.715   /TRANS.E=1.75  +
           SEG.0=196        SEG.E=-0.37    +
           /SEG.0=5882      /SEG.E=-0.37
IF.END
```

Boron-Interstitial Clustering Model with Small Clusters

In the boron-interstitial clustering model with small clusters:

- The *cieq* and *cveq* intermediates are the intrinsic equilibrium concentrations of interstitials and vacancies, respectively.
- The intermediate *phiv* specifies the charge state fraction of point-defects so that it depends on the charge states.
- After initializing *phiv* array elements with zero values, the values are assigned by the **LOOP** statement.
- The index string *i* in the **LOOP** statement is expanded by ‘@’ character as the quantity defined by the **ASSIGN** statement.
- The small clusters of vacancies are ignored so that the K_{Vi},V_c, and F_{Ic} are set to zero.

- The relative reaction rate K_{cV} between small interstitial clusters and vacancies depends on each charge state so it can be represented by a 2-D array intermediate.
- It is assumed that all K_I values for all charge states but the neutral state are zeros. All K_{cV} values are initialized to a constant value of 1. This implies that the Fermi-level dependences of K_I and K_{cV} are ignored. Thus, the K_I and K_{cV} values do not need to be specified with arrays.

  K_I and K_{cV} values, however, are specified with arrays in order to show how to use arrays.
- The active concentration should be adjusted to account for dopant-defect clusters.
- It is hypothesized that B_3I is the dominant boron-interstitial cluster at longer diffusion times. Thus, the parameters **`DDCF.D.N`**, **`DDCF.N.N`**, **`DDCF.I.N`**, **`DDCR.N.N`**, and **`DDCR.I.N`** have been specified to 3.0, -2.0, 2.0, 0.0, and 1.0 respectively.
- Ideally, **`IFRACM`** should be equal to (**`DDCF.I.N-DDCR.I.N`**)/**`DDCF.D.N`**, but it is given the value 1.356 which is calibrated together with K_{dd} and τ_{dd}.
- Since the matrix assembly in TSUPREM-4 assumes the time derivative term of a solution variable to the left-hand-side (LHS) of equation, the term $\frac{\partial}{\partial t}(I_c + I_{dds})$ is subtracted from the right-hand-side (RHS) term of interstitial equation.

$$\frac{\partial I}{\partial t}+\frac{\partial(I_c+I_{dds})}{\partial t} = RHS-\texttt{IFRACM}\cdot\ \frac{\partial C_{dd}}{\partial t}-K_b(F_{cV}+F_{Ic})\left(\frac{I}{I^*}\frac{V}{V^*}-1\right)$$

$$\frac{\partial V}{\partial t}+\frac{\partial V_c}{\partial t} = RHS-K_b(F_{cV}+F_{Ic})\left(\frac{I}{I^*}\frac{V}{V^*}-1\right)$$

$$\frac{\partial C_{dd}}{\partial t} = \frac{1}{\tau_{dd}}\left(K_{dd}n_i\left(\frac{C_a}{n_i}\right)^{\texttt{DDCF.D.N}}\eta^{\texttt{DDCF.N.N}}\left(\frac{I}{I^*}\right)^{\texttt{DDCF.I.N}}-C_{dd}\eta^{\texttt{DDCR.N.N}}\left(\frac{I}{I^*}\right)^{\texttt{DDCR.I.N}}\right)$$

where $I_c = \texttt{ECLUST.N}\sum_i(K_{Ii}\eta^{-i})\left(\frac{I}{I^*}\right)^{\texttt{ECLUST.N}}$,

$$V_c = \texttt{ECLUST.N}\sum_i(K_{Vi}\eta^{-i})\left(\frac{V}{V^*}\right)^{\texttt{ECLUST.N}},$$

$$I_{dds} = \texttt{IFRACS}(C-C_{dd}-C_{a2}),$$

$$C_{a2} = \frac{C-C_{dd}}{1+\texttt{DDCS.0}\cdot\ \exp\left(-\frac{\texttt{DDCS.E}}{kT}\right)\left(\frac{I}{I_i^*}\right)^{\texttt{IFRACS}}},$$

$$F_{cV} = \left(\frac{I}{I^*}\right)^{\texttt{ECLUST.N-1}}V_i^*\sum_{i,\ j}(K_{cV,\ ij}K_{Ii}\phi_{Vj}\eta^{-i-j}),$$

$$F_{Ic} = \left(\frac{V}{V^*}\right)^{\texttt{ECLUST.N-1}}I_i^*\sum_{i,\ j}(K_{Ic,\ ij}K_{Vj}\phi_{Ii}\eta^{-i-j})$$

```
$------------- Start of implementation --------------
IMPURITY IMP=BIC NEW
METHOD   IMP=BIC NONE ABS.ERR=1.0e5
INTERMED NAME=ni +
         EXPR=3.87e16*exp(-0.605/kt+1.5*log(temp))
INTERMED NAME=cieq EXPR=1.25E29*exp(-3.26/kt)
INTERMED NAME=cveq EXPR=1.25E29*exp(-3.26/kt)
INTERMED NAME=phipv MININDEX=-2 SIZE=5 +
         EXPRESS=”32.47*exp(-0.62/kt), 5.68*exp(-0.145/kt), +
           1.0,                                       +
           5.58*exp(-0.455/kt), 0.0”
INTERMED NAME=tphiv EXPR=”SUM(i,-2,2,phipv[i])”
INTERMED NAME=phiv  MININDEX=-2 SIZE=5 DEFAULT=0.0
LOOP     STEPS=5 INDEX=i START=-2
         INTERMED NAME=phiv[@i] EXPR=phipv[@i]/tphiv
L.END
INTERMED NAME=kI MININDEX=-2 SIZE=5 DEFAULT=0.0
INTERMED NAME=kI[0] EXPRESS=6.69E-2*exp(1.096/kt)
INTERMED NAME=kcV +
         MININDEX=”-2,-2” SIZE=”5,5” DEFAULT=1
INTERMED NAME=FcV +
         EXPR=”CI.RATIO*cveq*SUM(i,-2,2,SUM(j,-2,2, +
         kcV[i][j]*kI[i]*phiv[j]*ETA^(-i-j)))”
INTERMED NAME=ddcs EXPR=8.73796*exp(-2.769/kt)
INTERMED NAME=kddf EXPR=4.3356E-5*exp(0.224/kt)
INTERMED NAME=tdd  EXPR=EXP(LOG(4.18529E-31)+7.765/kt)
INTERMED NAME=Ca1  EXPR=BORON-BIC
INTERMED NAME=Ca2  EXPR=Ca1/(1+ddcs*(inter/cieq)^2)
INTERMED NAME=Ca   EXPR=EQACTIVE(BORON,Ca2)
INTERMED NAME=Idds EXPR=2*(Ca1-Ca2)
INTERMED NAME=Ic +
         EXPR=2*SUM(i,-2,2,kI[i]/eta^i)*CI.RATIO*CI.RATIO
INTERMED NAME=Rb +
         EXPR=1E-21*exp(1/kt)*Fcv*(CI.RATIO*CV.RATIO-1)
INTERMED NAME=dBIC  +
         EXPR=”(kddf*ni*(Ca/ni)^3*(CI.RATIO/ETA)^2 +
               -BIC*CI.RATIO)/tdd”
IMPURITY MODEL=BIC IMP=BORON MAT=Si ACTIVE=Ca
EQUATION MODEL=BIC VARIABLE=INTER MAT=Si +
         ADDTOEXP=-DDT(Ic+Idds)-1.356*dBIC-Rb
EQUATION MODEL=BIC VARIABLE=VACAN MAT=Si ADDTOEXP=-Rb
EQUATION MODEL=BIC VARIABLE=BIC  MAT=Si ADDTOEXP=dBIC +
         INIT=EXIST(BIC)?BIC:1E5
$------------- End of implementation ----------------
```

```
IF (1) ;$turn on USEIT model
  METHOD   MODEL=BIC ENABLE ;$ turn-on USEIT model
  METHOD   ACT.EQUI         ;$ turn-off built-in model
ELSE   ;$turn on built-in model
  METHOD   MODEL=BIC !ENABLE;$ turn-off USEIT model
  METHOD   ACT.FULL         ;$ turn-on built-in model
  INTERST  MAT=SI C.STATE=0 +
           ECLUST.0=0.0669 ECLUST.E=-1.096 ECLUST.N=2
  INTERST  MAT=SI C.ALL KCV.0=1 KCV.E=0
  VACANCY  MAT=SI C.STATE=0 +
           ECLUST.0=0 ECLUST.E=3.26 ECLUST.N=2
  VACANCY  MAT=SI C.ALL KIC.0=1 KIC.E=0
  IMPURITY IMP=BORON MAT=SI                       +
           IFRACS=2 IFRACM=1.356                  +
           DDC.F.0=4.3356E-5 DDC.T.0=4.18529E-31   +
           DDC.F.E=-0.2240 DDC.T.E=-7.765         +
           DDCF.D.N=3.0 DDCF.N.N=-2.0 DDCF.I.N=2.0 +
           DDCR.N.N=0 DDCR.I.N=1.0                +
           DDCS.0=8.73796 DDCS.E=2.769
IF.END
```

Diffusion Enhancement in Thinner Oxides

In the diffusion enhancement in thinner oxides:

- Since TSUPREM-4 assumes that interstitials exist only in silicon material, the silicon atom transported by segregation from silicon into oxide is specified as the new impurity *SOX*.
- The *SOOX* is a SiO molecule produced by the reaction between *SOX* and oxygen O_2 in oxides.
- Solving the equation of mobile, *SOX* uses the nearest neighbor to fill terms produced in factorization of the block matrices, so that the **PART** method is specified.
- The *SOOX* is assumed to be immobile, so that the **NONE** method is used for factorization.
- Since the interstitials in silicon are transported to oxides such as *SOX* impurities, the **TRANSPOR** parameters for interstitials and *SOX* must be turned *off*.
- The surfaces of structures (i.e.,ambient/material) have the exposed boundary condition, so the **TRANSPOR** must be turned *off* for the surface recombination of *SOX*.

- The **DI.FAC** parameter is specified to the enhancement factor of boron diffusivity in oxides.

$$\frac{\partial C_{Si}}{\partial t} = -\vec{\nabla} \cdot \left(-D_{Si}\vec{\nabla} C_{Si}\right) - k_{Si}C_{Si}\sqrt{C_{O_2}}$$

$$\frac{\partial C_{SiO}}{\partial t} = k_{Si}C_{Si}\sqrt{C_{O_2}} - k_{SiO}C_{SiO}\sqrt{C_{O_2}}$$

where $F_{Si} = h_{Si}(C_{Si} - mI)$ at silicon/oxide interfaces,

$F_{Si} = k_i C_{Si}$ at oxide surfaces

Diffusion enhancement is $D = D_o\left(1 + f_{SiO}\frac{C_{SiO}}{C_{SiO_2}}\right)$

```
$------------ Start of implementation -------------
IMPURITY IMP=SOX    NEW
IMPURITY IMP=SOOX   NEW
METHOD   IMP=SOX    PART ABS.ERR=1E9
METHOD   IMP=SOOX   NONE ABS.ERR=1E9
INTERMED NAME=Dsi   EXPR=13.0*exp(-4.5/kt)
INTERMED NAME=ksi   EXPR=7.676E6*exp(-4.0/kt)
INTERMED NAME=ksio  EXPR=ksi
INTERMED NAME=hsi   EXPR=0.01
INTERMED NAME=m     EXPR=2.9E-5*exp(2.19/kt)
INTERMED NAME=ki    EXPR=8.67E7*exp(-4.0/kt)
INTERMED NAME=CO    EXPR=SQRT(O2)
INTERMED NAME=R     EXPR=ksi*SOX*CO
EQUATION MODEL=DENH VARIABLE=SOX MAT=Ox +
         ADDTOEXP=DIV(Dsi*GRAD(SOX))-R +
         INIT=EXIST(SOX)?SOX:1E5
EQUATION MODEL=DENH VARIABLE=SOOX MAT=Ox +
         ADDTOEXP=R-ksio*SOOX*CO       +
         INIT=EXIST(SOOX)?SOOX:1E5
EQUATION MODEL=DENH VARIABLE=INTERS MAT1=Si /MAT2=Ox +
         INTERF=hsi*(ONMAT2(SOX)-m*ONMAT1(INTERS)) ^TRANSPOR
EQUATION MODEL=DENH VARIABLE=SOX   MAT1=Ox /MAT2=Si  +
         INTERF=hsi*(m*ONMAT2(INTERS)-ONMAT1(SOX)) ^TRANSPOR
EQUATION MODEL=DENH VARIABLE=SOX MAT1=Ox /MAT2=AMB    +
         INTERFAC=-ki*ONMAT1(SOX) ^TRANSPOR
IMPURITY MODEL=DENH IMP=BORON MAT=Ox +
         DI.FAC=1+4.55E-22*exp(0.28/kt)*SOOX
$------------ End of implementation ---------------

IF (1) ;$turn on USEIT model
  METHOD   MODEL=DENH ENABLE
  METHOD   MODEL=DEOX ^ENABLE
ELSE   ;$turn on built-in model
  METHOD   MODEL=DENH ^ENABLE
  METHOD   MODEL=DEOX ENABLE
IF.END
```

Surface Recombination Rate Depending On Nitrogen

- In the surface recombination model of interstitials for nitridation effect:
- Specify the interface at which the recombination happens with the **MAT** and **/MAT** parameters.
- The surface recombination is reduced by $K_s(\sigma_N / N_{max})(I - I^*)$.
- Use the **INTERFACE** parameter in order to describe the reduction of the surface recombination rate.
- Since nitrogen atoms are trapped on the boundary nodes in oxides, call the function **ONMAT2()** to the trapped nitrogen solution, *i_nitrogen*.
- Turn *off* the **TRANSPOR** parameter in order to avoid transporting interstitials into oxides.

$$R = K_s\left(1 - \frac{\sigma_N}{\mathbf{N.MAX.0}\exp\left(-\frac{\mathbf{N.MAX.E}}{kT}\right)}\right)(I - I^*)$$

```
$------------ Start of implementation -------------
INTERMED NAME=Ks   EXPRESS=1.4E-6*EXP(1.75/kt)
INTERMED NAME=Nmax EXPRESS=15/14*2E14
EQUATION MODEL=INREC VARIABLE=interst MAT=silicon /MAT=oxide +
         INTERFAC=Ks*ONMAT2(i_nitrogen)/Nmax*(interst-CI.STAR) +
         ^TRANSPOR
$------------ End of implementation ----------------

IF (1) ;$ use USEIT model
  METHOD   MODEL=INREC ENABLE
ELSE   ;$turn on built-in model
  METHOD   MODEL=INREC ^ENABLE
  INTERST  SILICON /OXIDE N.MAX.0=15/14*2E14 N.MAX.E=0.0
  METHOD   PD.NREC
IF.END
```

5-Stream Diffusion Model

The 3-stream diffusion model (**PD.FULL**) in TSUPREM-4 has been proven to be accurate enough to simulate most silicon process technologies. In the **PD.FULL** model, the quasi-static assumption under local equilibrium eliminates the equations for dopant-defect pairs. However, at very low thermal temperature, this assumption may not be applied any more, and the whole 5-stream diffusion system must be introduced. This example shows how to implement the 5-stream diffusion model with USEIT.

Bulk Equations

$$\frac{\partial C}{\partial t} = -\left(\vec{\nabla}\cdot\ \vec{J_m} + \vec{\nabla}\cdot\ \vec{J_n}\right)$$

$$\frac{\partial M}{\partial t} = -\vec{\nabla}\cdot\ \vec{J_m} + K_m\left(S\frac{I}{I^*} - \frac{M}{\alpha_m}\right) + K_{mv}\left(S - \frac{M}{\alpha_m}\frac{V}{V^*}\right)$$

$$\frac{\partial N}{\partial t} = -\vec{\nabla}\cdot\ \vec{J_n} + K_n\left(S\frac{V}{V^*} - \frac{N}{\alpha_n}\right) + K_{ni}\left(S - \frac{N}{\alpha_n}\frac{I}{I^*}\right)$$

$$\frac{\partial I}{\partial t} = -\vec{\nabla}\cdot\ \vec{J_I} - R_b - K_m\left(S\frac{I}{I^*} - \frac{M}{\alpha_m}\right) + K_{ni}\left(S - \frac{N}{\alpha_n}\frac{I}{I^*}\right)$$

$$\frac{\partial V}{\partial t} = -\vec{\nabla}\cdot\ \vec{J_V} - R_b - K_n\left(S\frac{V}{V^*} - \frac{N}{\alpha_n}\right) + K_{mv}\left(S - \frac{M}{\alpha_m}\frac{V}{V^*}\right)$$

where I and V denote an interstitial and a vacancy respectively, and S, M and N denote a substitutional dopant, a dopant-interstitial pair, and a dopant-vacancy pair, respectively.

$$\vec{J_m} = -D_m\left(\vec{\nabla}\left(\frac{M}{\alpha_m}\right) + z_s\left(\frac{M}{\alpha_m}\right)\left(\frac{\vec{\nabla}\eta}{\eta}\right)\right)$$

$$\vec{J_n} = -D_n\left(\vec{\nabla}\left(\frac{N}{\alpha_n}\right) + z_s\left(\frac{N}{\alpha_n}\right)\left(\frac{\vec{\nabla}\eta}{\eta}\right)\right)$$

$$\vec{J_I} = -D_I I^*\vec{\nabla}\left(\frac{I}{I^*}\right)$$

$$\vec{J_V} = -D_V V^*\vec{\nabla}\left(\frac{V}{V^*}\right)$$

where $D_m \equiv D_M\sum_k a_m\phi_{mk}\eta^{-k}$, $D_n \equiv D_N\sum_k a_n\phi_{nk}\eta^{-k}$, $\alpha_m \equiv a_m F_m = \frac{D_m}{D_M}$

and $\alpha_n \equiv a_n F_n = \frac{D_n}{D_N}$.

The reaction rates K_m, K_n, K_{mv}, and K_{ni} are given by:

$$K_m \equiv \sum_k (g_{mk}\phi_{Ik}\eta^{-k})$$

$$K_n \equiv \sum_k (g_{nk}\phi_{Vk}\eta^{-k})$$

$$K_{mv} \equiv \sum_k (r_{mvk}a_m\phi_{mk}\phi_{V(-k)})$$

$$K_{ni} \equiv \sum_{k} (r_{nik} a_n \phi_{nk} \phi_{I(-k)})$$

The point-defect recombination R_b is given by:

$$R_b = -\sum_{j} \sum_{k} (r_{IV,\ jk} \phi_{Ij} \phi_{Vk} \eta^{-(j+k)}) \left(1 - \frac{I}{I^*}\frac{V}{V^*}\right)$$

Boundary Conditions

Since the dopant-defect pairs are mobile, their boundary conditions (recombination or transporting flux) must be specified. In materials other than silicon, the point-defects, and also dopant-defect pairs are ignored, so that the annihilation of dopant-defect pairs at interfaces must be specified. The recombinations are given by:

$$F_m = h\left(\frac{M}{m} - M^*\right)$$

$$F_n = h\left(\frac{N}{m} - N^*\right)$$

where: $M^* = \dfrac{\alpha_m C_a}{1 + \alpha_m + \alpha_n}$ and $N^* = \dfrac{\alpha_n C_a}{1 + \alpha_m + \alpha_n}$

The *h* and *m* are given by the transport and segregation coefficients of an impurity, respectively.

Implementation

Since USEIT allows the additional terms to be added to built-in equations, one might assume that the kick-out and Frank-Turnbull reaction terms are simply added to the built-in point-defect equations instead of using the built-in reaction terms in **`PD.FULL`**. However, this is not possible because the built-in initialization procedure of free point-defects at the first post-implant annealing process is performed by assuming that the dopant-defect pairs are already formed in equilibrium. Thus, the whole equation system for point-defects must be implemented.

Point Defect Parameter

The parameters for the diffusivities, equilibrium concentrations, charge fractions, surface recombination rates, and bulk recombination rate, as well as interstitial generation for oxidation, are specified as intermediates. *Note* that the charge fraction intermediates are the arrays starting with -2 (double negative), the size of which is 5, counting up to double positive. When you want to see the effect due to a different parameter set, the specified values can be replaced with the new ones.

```
intermed name=DI0    expr=3.65e-4
intermed name=DIE    expr=1.58
intermed name=DV0    expr=3.65e-4
intermed name=DVE    expr=1.58
intermed name=IiS0   expr=1.25e29
intermed name=IiSE   expr=3.26
intermed name=ViS0   expr=1.25e29
intermed name=ViSE   expr=3.26
intermed name=PhiI0  size=5 minindex=-2 +
                     expr="  0.0, 5.68, 1.0, 5.68,  0.0"
intermed name=PhiIE  size=5 minindex=-2 +
                     expr="  0.0, 0.50, 0.0, 0.26,  0.0"
intermed name=PhiV0  size=5 minindex=-2 +
                     expr="32.47, 5.68, 1.0, 5.68,  0.0"
intermed name=PhiVE  size=5 minindex=-2 +
                     expr=" 0.62, 0.145,0.0, 0.455, 0.0"
intermed name=KSI0   expr=1.4E-6
intermed name=KSIE   expr=-1.75
intermed name=KSV0   expr=4.0E-11
intermed name=KSVE   expr=-1.75
intermed name=KB0    expr=1E-21
intermed name=KBE    expr=-1.0
$ I small clusters
intermed name=kI0    size=5 minindex=-2 default=0.0
intermed name=kIE    size=5 minindex=-2 default=0.0
intermed name=kI0[0] expr=0.0668848
intermed name=kIE[0] expr=-1.096
intermed name=kcV   minindex="-2,-2" length="5,5" default=1.0
$ OED
intermed name=VNORM0 expr=1.0
intermed name=VNORME expr=0.0
intermed name=THETA0 expr=0.01
intermed name=THETAE expr=0.0
intermed name=GPOW0  expr=-0.7
intermed name=GPOWE  expr=0.0
```

Dopant Parameter

The diffusivities for dopant-defect pairs, the reaction rates for pairing with point-defects, the diffusivities in silicon, and the segregation flux are defined. The specified values here are for boron.

```
intermed name=DIPAIR0 expr=3.65e-4
intermed name=DIPAIRE expr=1.58
intermed name=DVPAIR0 expr=3.65e-4
intermed name=DVPAIRE expr=1.58
intermed name=RIS    expr=10
intermed name=RVS    expr=10
intermed name=RIPV   expr=10
intermed name=RVPI   expr=10
intermed name=EIS    expr=0.0
intermed name=EVS    expr=0.0
intermed name=EIPV   expr=0.0
intermed name=EVPI   expr=0.0
intermed name=FIB    expr=0.95
intermed name=DX0    expr=0.0368
intermed name=DXE    expr=3.46
intermed name=DP0    expr=0.716
intermed name=DPE    expr=3.46
intermed name=TRANS0 expr=1.67E-7
intermed name=TRANSE expr=0.0
intermed name=SEG0   expr=1.126E3
intermed name=SEGE   expr=0.91
```

Thermal Parameter for Point Defects

```
$ point-defect diffusivity
intermed name=DI     expr=DI0*exp(-DIE/kt)
intermed name=DV     expr=DV0*exp(-DVE/kt)
$ intrinsic point-defect concentration in equilibrium
intermed name=IiStar expr=IiS0*exp(-IiSE/kt)
intermed name=ViStar expr=ViS0*exp(-ViSE/kt)
$ point-defect charge fraction
intermed name=sphi  expr=sum(i,-2,2,PhiI0[i]*exp(-PhiIE[i]/kt))
intermed name=sphv  expr=sum(i,-2,2,phiV0[i]*exp(-PhiVE[i]/kt))
intermed name=phiI  size=5 minindex=-2 default=0
intermed name=phiV  size=5 minindex=-2 default=0
loop steps=5 index=i start=-2
  intermed name=phiI[@i] expr=phiI0[@i]*exp(-PhiIE[@i]/kt)/sphi
  intermed name=phiV[@i] expr=phiV0[@i]*exp(-PhiVE[@i]/kt)/sphv
l.end
$ equilibrium I and V
intermed name=FI    expr=sum(i,-2,2,phiI[i]*ETA^(-i))
intermed name=FV    expr=sum(i,-2,2,phiV[i]*ETA^(-i))
intermed name=IStar expr=IiStar*FI
intermed name=VStar expr=ViStar*FV
$ bulk recombination
intermed name=kbsi  expr=KB0*exp(-KBE/kt)
intermed name=FIV   expr=IiStar*ViStar
$ surface recombination
intermed name=ksurfi expr=KSI0*exp(-KSIE/kt)
intermed name=ksurfv expr=KSV0*exp(-KSVE/kt)
```

OED - Interstitial Generation due to Oxidation

Note that *x.vel* and *y.vel* are the reserved keywords for x- and y-directional boundary velocities, respectively.

```
$ surface interstitial generation during oxidation
intermed name=vnorm  expr=VNORM0*exp(-VNORME/kt)
intermed name=gpow   expr=GPOW0*exp(-GPOWE/kt)
intermed name=theta  expr=THETA0*exp(-THETAE/kt)
intermed name=vox    expr=sqrt(x.vel^2+y.vel^2)
intermed name=GSurf  expr=5E22*theta*vox*(vox/vnorm)^gpow
```

Thermal Parameter for Dopant

```
$ dopant-defect pair diffusivity
intermed name=DiPairb expr=DIPAIR0*exp(-DIPAIRE/kt)
intermed name=DvPairb expr=DVPAIR0*exp(-DVPAIRE/kt)

$ measured diffusivity
intermed name=Db     size=2 minindex=0 +
                     expr="DX0*exp(-DXE/kt),DP0*exp(-DPE/kt)"
intermed name=Dib    size=5 minindex=-2 +
                     expr="0,0,FIB*Db[0],FIB*Db[1],0"
intermed name=Dvb    size=5 minindex=-2 +
                     expr="0,0,(1-FIB)*Db[0],(1-FIB)*Db[1],0"
intermed name=Dmb    expr=Dib[0]+Dib[1]/ETA
intermed name=Dnb    expr=Dvb[0]+Dvb[1]/ETA
intermed name=Amb    expr=Dmb/DiPairb
intermed name=Anb    expr=Dnb/DvPairb

$ Kick-out and Frank-Turnbull reaction
intermed name=gmb    expr=12.57*RIS*1e-8*exp(-EIS/kt)*DI*IiStar
intermed name=gnb    expr=12.57*RVS*1e-8*exp(-EVS/kt)*DV*ViStar
intermed name=rmvb   +
         expr=12.57*RIPV*1e-8*exp(-EIPV/kt)*(DiPairb+DV)*ViStar
intermed name=rnib   +
         expr=12.57*RVPI*1e-8*exp(-EVPI/kt)*(DvPairb+DI)*IiStar
intermed name=Kmb    expr=gmb*FI
intermed name=Knb    expr=gnb*FV
intermed name=Kmvb   +
         expr=rmvb*sum(i,-2,2,Dib[i]/DiPairb*phiV[-i])
intermed name=Knib   +
         expr=rnib*sum(i,-2,2,Dvb[i]/DvPairb*phiI[-i])

$ transport and segregation coefficients
intermed name=Trans expr=TRANS0*exp(-TRANSE/kt)
intermed name=Seg   expr=SEG0*exp(-SEGE/kt)
```

Point Defect Solutions

```
impurity imp=Isi new
impurity imp=Vsi new
method   imp=Isi lu full ^symm time.ste rel.err=0.005 abs.err=1e5
method   imp=Vsi lu full ^symm time.ste rel.err=0.005 abs.err=1e5
```

The `Isi` and `Vsi` are the free interstitial and vacancy, respectively.

Pair Solutions

```
impurity imp=Mb new
impurity imp=Nb new
method   imp=Mb lu full ^symm time.ste rel.err=0.01 abs.err=1.0e5
method   imp=Nb lu full ^symm time.ste rel.err=0.01 abs.err=1.0e5
```

Substitutional Boron

```
intermed name=Subb   expr=active(boron)-Mb-Nb
```

Intermediates depending on Point Defects

```
$ I/I* and V/V*
intermed name=ciratio expr=Isi/Istar
intermed name=cvratio expr=Vsi/Vstar

$ Rb
intermed name=RIVB  expr=kbsi*FIV*(ciratio*cvratio-1)

intermed name=IPb   expr=(Kmb*ciratio+Kmvb)/(Kmb+Kmvb*cvratio)
intermed name=VPb   expr=(Knb*cvratio+Knib)/(Knb+Knib*ciratio)
intermed name=Asumb expr=1+Amb*Ipb+Anb*Vpb
intermed name=AmStar expr=Amb/(1+Amb+Anb)
intermed name=AnStar expr=Anb/(1+Amb+Anb)
intermed name=CMovrM expr=Amb*Ipb/Asumb
intermed name=CNovrN expr=Anb*Vpb/Asumb
```

Kick-out and Frank-Turnbull Reacions

```
$ generation/recombination
intermed name=MovrAb expr=Mb/Amb
intermed name=NovrAb expr=Nb/Anb
intermed name=GRmb    expr=Kmb*(Subb*ciratio-MovrAb)
intermed name=GRmvb   expr=Kmvb*(Subb-MovrAb*cvratio)
intermed name=GRnb    expr=Knb*(Subb*cvratio-NovrAb)
intermed name=GRnib   expr=Knib*(Subb-NovrAb*ciratio)
```

Boron Diffusion Equation

The **EQUATION** statements for implementation of the new boron equation replace the built-in boron diffusion equation by turning off the **DIFFUSE** parameter in the **IMPURITY** statement.

```
impurity imp=boron mat=silicon ^diffuse
equation model=fivestream variable=boron mat=silicon +
         addtoexp=div(Dmb*(grad(MovrAb)+MovrAb*efield))
equation model=fivestream variable=boron mat=silicon +
         addtoexp=div(Dnb*(grad(NovrAb)+NovrAb*efield))
```

Pair Diffusion Equations

```
equation model=fivestream variable=Mb mat=silicon +
         initial="exist(Mb)?Mb:1E5" +
         addtoexp=div(Dmb*(grad(MovrAb)+MovrAb*efield))+GRmb+GRmvb
equation model=fivestream variable=Nb mat=silicon +
         initial="exist(Nb)?Nb:1E5" +
         addtoexp=div(Dnb*(grad(NovrAb)+NovrAb*efield))+GRnb+GRnib
```

For the boundary condition, the dopant-defect pairs are annihilated at interfaces by turning off **`TRANSPORT,`** since the dopant-defect pairs exist only in silicon.

```
$ equilibrium M* and N*
intermed name=MbStar    expr=AmStar*eqact(boron,boron)
intermed name=NbStar    expr=AnStar*eqact(boron,boron)
$ boundary equations
equation model=fivestream variable=Mb mat1=silicon /mat2=oxide +
         ^transpor interfac=-Trans*(onmat1(Mb/Seg-MbStar))
equation model=fivestream variable=Nb mat1=silicon /mat2=oxide +
         ^transpor interfac=-Trans*(onmat1(Nb/Seg-NbStar))
```

Point Defect Equations

The *interstitial* and *vacancy* variables are the free interstitial and vacancy concentrations, respectively. In the 3-stream diffusion models, **`PD.TRANS`** or **`PD.FULL`**, the dopant-defect pairs are instantly determined before the first post-implant anneal, so that the total point-defect concentrations are conserved by subtracting the dopant-defect pairs and the interstitial small clusters from the total concentrations. When the *interstitial* or *vacancy* solution variables are used, they are the free not total concentrations. However, the *interstitial* or *vacancy* variables used in the **`INITIAL`** parameter are the total concentrations.

```
equation model=fivestream variable=Isi mat=silicon +
         initial=”damage>1.15e22? IStar: +
                   (exist(Isi)?interstitial+Isi:interstitial)” +
         addtoexp=div(DI*IStar*grad(ciratio))-RIVB-GRmb+GRnib
equation model=fivestream variable=Vsi mat=silicon +
         initial=”damage>1.15e22?VStar: +
                   (exist(Vsi)?vacancy+Vsi:vacancy)” +
         addtoexp=div(DV*VStar*grad(cvratio))-RIVB-GRnb+GRmvb
```

The surface recombination is described as follows:

```
equation model=fivestream variable=Isi mat=silicon /mat=oxide +
         interfac=-ksurfi*onmat1(Isi-IStar) ^transpor
equation model=fivestream variable=Vsi mat=silicon /mat=oxide +
         interfac=-ksurfv*onmat1(Vsi-VStar) ^transpor
```

Since the initialization uses the built-in calculation of the point defects for implantation, `Isi` and `Vsi` in the structure must be added to the generated built-in pointdefects due to implantation. After initializing and adding `Isi` and `Vsi`, the built-in point defect solutions are eliminated by setting **`PD.FERMI`**.

For example:

```
method model=fivestream enable
$ turn on PD.TRANS in order to use the built-in initialization
method pd.trans
implant boron dose=1e14 energy=20

$ initialize Isi and Vsi
diffuse temp=900 time=0

$ remove the built-in point defect solutions
method pd.fermi

diffuse temp=900 time=10/60

$ reinitialize Isi and Vsi for the second implant
method pd.trans
implant boron dose=1e15 energy=10
diffuse temp=1000 time=0
method pd.fermi

diffuse temp=1000 time=5/60
```

Coupling with Small Clusters

To see the certain effect of the 5-stream diffusion model, the non-equilibrium status for the formation of dopant-defect pairs must be dominant in the diffusion process. The diffusion process is usually performed at very low temperature and/ or for very short time. The small clusters must be taken into account for such a process condition. In TSUPREM-4, the small clusters, I_2 and BI_2, are assumed to form competitively with the boron-interstitial pair formation by assuming that I_2, BI_2 and M_b(boron-interstitial pair) are in equilibrium. Thus, the initialization of the small clusters with the zero dopant-defect pair concentrations is not physical. Unless the transient equations are implemented for I_2 and BI_2 formation in non-equilibrium, it is not possible to account for the small clusters in the 5-stream diffusion model. It will be shown how the initialized profiles of I_2, BI_2, I(free interstitial), and M_b in the built-in codes are different from those in the 5-stream model using USEIT.

Insterstitial Small Clusters - I_2

```
$ reaction for I small clusters
loop     steps=5 index=i start=-2
  intermed name=kI[@i] expr=kI0[@i]*exp(-(kIE[@i])/kt)
l.end
$ I small clusters
intermed name=Fcv +
         expr=ciratio*ViStar*sum(i,-2,2, +
                  sum(j,-2,2,kcV[i][j]*kI[i]*phiV[j]*ETA^(-i-j)))
intermed name=Ic    expr=2*ciratio^2*sum(i,-2,2,kI[i]/ETA^i)
intermed name=rc    expr=RIVB/IiStar/ViStar*Fcv
equation model=fivestream var=Isi mat=si addtoexp=-ddt(Ic)-rc
equation model=fivestream var=Vsi mat=si addtoexp=-rc
```

BI_2 and B_3I

```
$ BIC
impurity imp=BIC new
method   imp=BIC none abs.err=1e5
intermed name=Ca1  expr=boron-BIC
intermed name=Ca2  expr=Ca1/(1+ddcs*(Isi/IiStar)^2)
intermed name=Ca   expr=eqactive(boron,Ca1)
intermed name=dBIC +
         expr=(kddf*ni*(Ca/ni)^3*(ciratio/ETA)^2-BIC*ciratio)/tdd
intermed name=Idds expr=2*(Ca1-Ca2)
intermed name=Rdd  expr=1.356*dBIC
impurity model=BIC imp=boron  mat=si active=eqact(boron,Ca2)
equation model=BIC imp=BIC    mat=si addtoexp=dBIC +
                   initial=exist(BIC)?BIC:1e5
equation model=BIC imp=Isi    mat=si addtoexp=-ddt(Idds)-Rdd
```

Modified Initialization of Interstitial

The initialization for the 5-stream diffusion model can be achieved by calling the function **SOLVE** for Newton-Raphson iteration.

```
equation model=fivestream variable=Isi mat=silicon +
initial=damage>1.15e22?IStar:solve(Isi,Isi+Idds+Ic-inters,inters)
```

Comparison of Initialization

Figures 7-3 and 7-4 show the initialization results with the built-in **PD.FULL** codes and the USEIT 5-stream diffusion model, respectively. For the 3-stream diffusion model, the boron-interstitial pair concentration is calculated in equilibrium with the interstitial small clusters and the boron-interstitial small clusters. Due to the boron-interstitial pair formation by the local equilibrium assumption, the number of free interstitials is smaller than the 5-stream diffusion model, in which no initial dopant-defect pair exists. Solving the transient equations of a dopant-defect pair formation with the equilibrium assumption in the small cluster formation is not correct unless the small clusters form much faster than the dopant-defect pairing, which has not yet been proven. It might be a more reasonable approach to assume the formation of dopant-defect pairs in equilibrium, as well as I_2 and BI_2; in other words, to use the **PD.FULL** model rather than the 5-stream diffusion model, unless the transient formation of the small clusters is implemented.

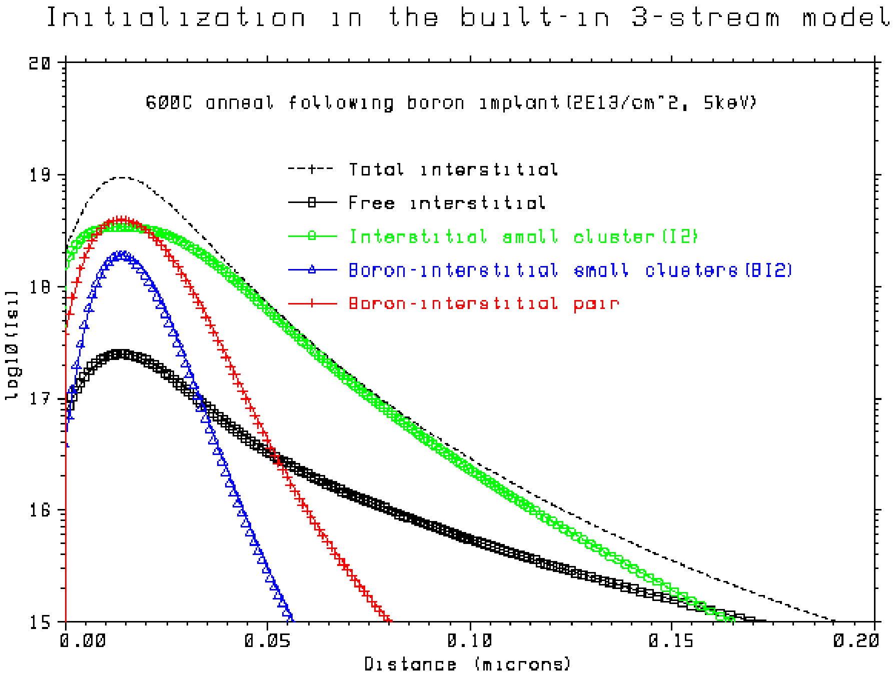


Figure 7-3 Initialization in the built-in 3-stream diffusion model

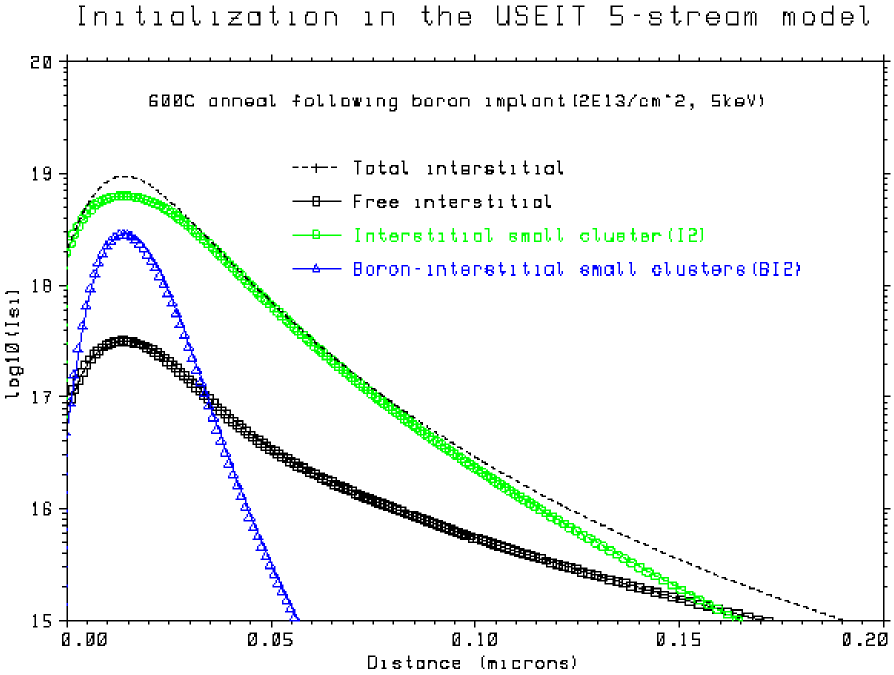


Figure 7-4 Initialization in the USEIT 5-stream diffusion model

SiGe Model

The model is based on Lever, Bonar, and Willoughby's work ("Boron diffusion across silicon-silicon germanium boundaries", *J. Appl. Phys.* **83**, 1988 (1998)). In SiGe, the isovalent atoms like Ge can be as traps for boron atoms, so that the number of mobile boron atoms is reduced. Trapping a boron atom forms an immobile complex Ge-B.

$$\frac{C_{GeB}}{C_a} = \frac{C_{as}-C_a}{C_a} = \frac{C_{Ge}}{C_o}$$

where C_0 is the characteristic concentration, $3.2\times 10^{21}/\mathrm{cm}^3$ and C_{Ge} is the germanium concentration. C_{as} is the active concentration when ignoring the boron trapping. The intrinsic carrier concentration increases with the Ge content because of the energy band-gap reduction.

$$n_i = n_{i,\ Si}\exp\left(\frac{\chi\delta E_g}{2kT}\right)$$

where χ is the fraction of Ge content, and δE_g is the experimental constant energy, 0.7eV. Assuming the equilibrium point defect concentrations, from the above equations, the boron diffusion flux can be calculated by:

$$\vec{J} = -D_B\left(\vec{\nabla}C_a + C_a\frac{\vec{\nabla}N_T}{Q} + C_a\frac{2n}{Q}\cdot\ \vec{\nabla}\ln(n_i)\right)$$

where N_T and Q mean the net doping concentration, N_a-N_d, and the total carrier concentration, $n+p$, respectively. The boron diffusivity D_B is given by:

$$D_B = \left(D_x + D_P\cdot\ \frac{p}{n_{i,\ Si}}\right)\cdot\ \exp\left(-\frac{\varepsilon s\chi}{kT}\right)$$

where ε and s mean the energy per unit strain and the characteristic strain in the plane of the SiGe layer, respectively. Cowern *et.al.*'s work shows that the diffusion of B in compressively strained SiGe decreases due to an increase in the activation energy of B diffusion ("Diffusion in Strained SiGe", *Phys. Rev. Lett.* **72**, 2585 (1994)). They extracted 40eV/strain for ε and 0.042 for s. However, in this example, ε is set to zero, so that the strain-dependence of the activation energy is ignored.

Note:

The intrinsic carrier concentration in the diffusivity model is acting just as the convenient normalization factor, so that $n_{i,Si}$ ***must be used instead of*** n_i***.***

```
impurity imp=Germanium new

intermed name=GeMax      express=5e22
intermed name=StrainMax  express=0.042
intermed name=GeFrac     express=Germanium/GeMax
intermed name=StrainSiGe express=GeFrac*StrainMax

intermed name=EaStrain   express=0 ;$ instead of 40
intermed name=dEa        express=EaStrain*StrainSiGe
intermed name=DX0        expr=0.0368
intermed name=DBE        expr=3.46+dEa
intermed name=DP0        expr=0.716
intermed name=Db         size=2 minindex=0 +
                         expr="DX0*exp(-DBE/kt),DP0*exp(-DBE/kt)"

intermed name=dEg      express=0.7
intermed name=Ni       expr=3.87e16*exp(-0.605/kt+1.5*log(temp))
intermed name=NiSiGe   express=Ni*exp(GeFrac*dEg/(2*kt))
intermed name=C0       express=3.2e21
intermed name=fTrap    express=1/(1.0+Germanium/C0)

intermed name=Na       express=active(boron)
intermed name=Nd       express=active(arsenic)+active(phosphorus)
intermed name=Nt       express=fTrap*(Na-Nd)
intermed name=Q        express=sqrt(Nt^2+4*NiSiGe^2)
intermed name=nelectron express=0.5*(Q-Nt)
intermed name=nhole     express=0.5*(Q+Nt)
intermed name=DbSiGe   express=Db[0]+Db[1]*nhole/Ni
intermed name=Bmobile  express=fTrap*active(boron)

equation model=sige variable=boron mat=si +
         addtoexp=”div(DbSiGe*(grad(Bmobile)+ +
         Bmobile/Q*(grad(Nt)+2*nelectron*grad(log(NiSiGe)))))”
impurity imp=boron mat=si ^diffuse
```

Default Coefficients

Default Coefficient Values

This appendix contains the values of the various simulation coefficients used in TSUPREM-4, along with the references from which the values were obtained. In some cases, there are also notes regarding the approximations that were made when the necessary values had not been experimentally determined or when there was wide scatter in the available data. These notes regarding the approximations involved in the coefficient determination should help you know how much latitude is available when modifying these coefficients to match fabrication results. The units for the coefficients are given in Chapter 4; where a choice of units is available, the default units (i.e., microns and minutes) are used, except as noted.

Tables	Page
Impurity Parameters	A-2
Impurity Diffusion Coefficients	A-2
Segregation and Transport Coefficients	A-5
Polysilicon Grain Segregation	A-7
Clustering and Solid Solubility	A-7
Point Defect Parameters	A-8
Oxidation	A-11
Diffusion Enhancement Oxides	A-15
Silicidation	A-16
Analytical Implant	A-17
Monte Carlo (MC) Implant	A-19
Electrical Parameters	A-21
Material Coefficients	A-23
Numerical Methods	A-26
Automatic Grid Generation Parameters	A-27
Adaptive Grid Parameters	A-28

The notes referenced in this section are described in Default Coefficient Notes, p. A-34. They appear in the form “**(1)**,” in which the digit is the number of the note in the list of notes.

The references listed in this section are described in Default Coefficient References, p. A-29. They appear in the form “[1],” in which the digit is the number of the reference in the list of references.

Impurity Parameters

Table A-1 Impurity Parameters (Silicon Dopants)

	Boron	Phosphorus	Arsenic	Indium	Antimony
TIF.NAME	B	P	As	In	Sb
Type	**ACCEPTOR**	**DONOR**	**DONOR**	**ACCEPTOR**	**DONOR**
Implant table	chboron	dual.pho	dual.ars	tr.indium	antimony
AT.NUM	5.0	15.0	33.0	49.0	51.0
AT.WT	10.8	30.97	74.91	115	121.76

Table A-2 Impurity Parameters (Other)

	Aluminum	Fluorine	Nitrogen	Silicon
TIF.NAME	Al	F	N	Si
AT.NUM	13.0	9.0	7.0	
AT.WT	26.98	18.9984	14.01	

Impurity Diffusion Coefficients

Diffusivities are in μm^2/min except as noted.

Table A-3 Impurity Diffusion Coefficients

	Boron	Phosphorus	Arsenic	Indium	Antimony	Fluorine	Germanium	Nitrogen
Silicon (1)								
DIP.0	4.10e9 [1]	0 [1]	0 [2]	0 **(2)**	0 [1]	0	0	0
DIP.E	3.46	0	0	0	0	0	0	0
DIX.0	2.11e8	2.31e10	1.37e7	1.8 cm^2/sec	6.420e7	4.23e3 cm^2/sec	80.0 cm^2/sec	0.05 cm^2/sec
DIX.E	3.46	3.66	3.44	3.6	3.65	3.5	4.7	3.65
DIM.0	0	2.664e10	3.72e10	0	4.50e9	0	0	0
DIM.E	0	4.0	4.15	0	4.08	0	0	0
DIMM.0	0	2.652e11	0	0	0	0	0	0
DIMM.E	0	4.37	0	0	0	0	0	0
DVP.0	2.16e8	0	0	0	0	0	0	0
DVP.E	3.46	0	0	0	0	0	0	0
DVX.0	1.11e7	0	5.47e7	0	1.220e9	0	320.0 cm^2/sec	0
DVX.E	3.46	3.66	3.44	0	3.65	0	0	0
DVM.0	0	0	1.49e11	0	8.55e10	0	0	0
DVM.E	0	4.0	4.15	0	4.08	0	0	0
DVMM.0	0	0	0	0	0	0	0	0
DVMM.E	0	4.37	0	0	0	0	0	0
VD	4.4	10.0	-13.2	0	-13.0	0	-13.0	0

Table A-3 Impurity Diffusion Coefficients (Continued)

	Boron	Phosphorus	Arsenic	Indium	Antimony	Fluorine	Germanium	Nitrogen
Oxide [8] **(3)**								
DIX.0	1.896e6	4.338e10	1.05e10	18.	7.86e25	2.858	1.05e10	0.05
DIX.E	3.53	4.44	4.89	3.6	8.75	3.5	4.89	3.65
Polysilicon (grain interior) (diffusivities in cm^2/sec) (4)								
DIP.0	0.72	0	0	0	0	0	0	0
DIP.E	3.46	0	0	0	0	0	0	0
DIX.0	0.037	3.85	1.14e-2	1.8	0.214	1.70e5	1.14e-2	0.05
DIX.E	3.46	3.66	3.44	3.6	3.65	3.5	3.44	3.65
DIM.0	0	4.44	31.0	0	15.0	0	0	0
DIM.E	0	4.0	4.15	0	4.08	0	0	0
DIMM.0	0	44.2	7.17e3	0	0	0	0	0
DIMM.E	0	4.37	5.21	0	0	0	0	0
Polysilicon (grain boundaries) (diffusivities in cm^2/sec) (5)								
DIX.0	420.	6.0e3	110.0	0	2.14e3	0	0	0
DIX.E	3.46	3.66	3.53	0	3.65	0	0	0
DIM.0	0	0	10.0	0	0	0	0	0
DIM.E	0	0	3.53	0	0	0	0	0
$TiSi_2$ (diffusivities in cm^2/sec)								
DIX.0	6.0e-7	4.6e-6	4.0e-7	0	7.0e-8	0	4.0e-7	0
DIX.E	2.0	2.0	1.8	0	1.8	0	1.8	0
$CoSi_2$ (diffusivities in cm^2/sec)								
DIX.0	1.64e-2	8.0e-5	7.37	1.64e-2	7.37	0	4.0e-7	0
DIX.E	2.05	2.05	3.3	2.05	3.3	0	1.8	0
WSi_2 (diffusivities in cm^2/sec) (6) [9]-[12]								
DIX.0	1.0e-5	4.20e-2	2.60e-2	0	2.60e-2	0	2.60e-2	0
DIX.E	1.17	2.14	2.11	0	2.11	0	2.11	0
Ambient (7)								
DIX.0	1e13	1e13	5e13	180.	5e13	1e13	5e13	5.0
DIX.E	3.5	3.5	3.95	3.6	3.95	3.5	3.95	3.65

Table A-4 Other Impurities and Materials

DIP.0	0
DIP.E	0
DIX.0	0
DIX.E	0
DIM.0	0
DIM.E	0
DIMM.0	0
DIMM.E	0

Table A-5 User-Defined Multiplication Factors for all materials

DI.FAC	1

Table A-5 User-Defined Multiplication Factors for all materials

`DI.F11`	1
`DI.F22`	1
`DV.FAC`	1
`DV.F11`	1
`DV.F22`	1

Table A-6 Pair Kinetics Parameters (all impurities; `CM.SEC` units) (8)

`DIPAIR.0`	3.65e-4
`DIPAIR.E`	1.58
`DVPAIR.0`	3.65e-4
`DVPAIR.E`	1.58
`R.I.S`	10
`E.I.S`	0
`R.V.S`	10
`E.V.S`	0
`R.IP.V`	10
`E.IP.V`	0
`R.VP.I`	10
`E.VP.I`	0

Table A-7 Pair Surface Recombination Coefficients (CM.SEC units)

	Boron	Phosphorus	Arsenic	Antimony
Silicon/oxide				
`KSIP.0`	1.4e-6	1.4e-6	1.4e-6	1.4e-6
`KSIP.E`	-1.75	-1.75	-1.75	-1.75
`KSVP.0`	4e-11	4e-11	4e-11	4e-11
`KSVP.E`	-1.75	-1.75	-1.75	-1.75
Silicon/ambient				
`KSIP.0`	150	150	150	150
`KSIP.E`	0	0	0	0
`KSVP.0`	150	150	150	150
`KSVP.E`	0	0	0	0
Other Impurities and Pairs of Materials				
`KSIP.0`	0	0	0	0
`KSIP.E`	0	0	0	0
`KSVP.0`	0	0	0	0
`KSVP.E`	0	0	0	0

Table A-8 Pair Segregation Coefficients

All Impurities and Pairs of Materials	
`SEG.IFRA`	1

Table A-8 Pair Segregation Coefficients

`SEG.IPOW`	0
`SEG.VPOW`	0
`SEG.CPOW`	0

Segregation and Transport Coefficients

Table A-9 Segregation Coefficients

	Boron	Phosphorus	Arsenic	Indium	Antimony	Fluorine	Nitrogen
Silicon/oxide							
`SEG.0`	1.126e3 [13]	30.0 [14] (9)	30.0 [14]	1.3e-13[15]	30.0 [14]	5.62E-8	0.14
`SEG.E`	0.91	0.00	0.0	-2.55	0.0	0.0	0.0
Polysilicon/oxide (10)							
`SEG.0`	1.126e3	30.00	30.0	1.3e-13	30.0	5.62E-8	0.14
`SEG.E`	0.91	0.0	0.0	-2.55	0.0	0.0	0.0
Silicon/$TiSi_2$							
`SEG.0`	0.3	1.0	0.8	1.0	0.8	1.0	1.0
`SEG.E`	0	0	0	0	0	0	0
Polysilicon/$TiSi_2$							
`SEG.0`	0.3	1.0	0.8	1.0	0.8	1.0	1.0
`SEG.E`	0	0.0	0	0	0	0	0
Silicon/WSi_2 [9]-[12]							
`SEG.0`	1.0	5.0	10.0	1.0	10.0	1.0	1.0
`SEG.E`	0	0	0	0	0	0	0
Polysilicon/WSi_2							
`SEG.0`	1.0	5.0	10.0	1.0	10.0	1.0	1.0
`SEG.E`	0	0	0	0	0	0	0
Other Impurities and Pairs of Materials							
`SEG.0`	1.0	1.0	1.0	1.0	1.0	1.0	1.0
`SEG.E`	0	0	0	0	0	0	0

Table A-10 Interface Transport Coefficients

	Boron	Phosphorus	Arsenic	Indium (11)	Antimony
Silicon/ambient [16] **(12)**					
`TRANS.0`	1.674e7	9.0e5	9.0e5	1.8e8cm/sec	1.5e3
`TRANS.E`	2.481	1.99	1.99	3.6	1.04
Oxide/ambient (13)					
`TRANS.0`	1.647e7	9.0e5	9.0e5	18e8cm/sec	1.5e3
`TRANS.E`	2.481	1.99	1.99	3.6	1.04
Polysilicon/ambient (13)					
`TRANS.0`	1.674e7	9.0e5	9.0e5	1.8e8cm/sec	1.5e3
`TRANS.E`	2.481	1.99	1.99	3.6	1.04

Table A-10 Interface Transport Coefficients

	Boron	Phosphorus	Arsenic	Indium (11)	Antimony
Polysilicon/silicon (14)					
TRANS.0	100.	100.	100.	1.8e8cm/sec	100.
TRANS.E	0	0	0	3.6	0
Silicon/oxide and Polysilicon/oxide					
TRANS.0	0.1	0.1	0.1	1.8e8cm/sec	0.1
TRANS.E	0	0	0	3.6	0
All materials with Nitride and Oxynitride					
TRANS.0	0	0	0	0	0
TRANS.E	0	0	0	3.6	0
Silicon/$TiSi_2$					
TRANS.0	1.0e6	1.0e6	1.0e6	0	1.0e6
TRANS.E	2.0	2.0	2.0	0	2.0
Polysilicon/$TiSi_2$					
TRANS.0	1.0e6	1.0e6	1.0e6	0	1.0e6
TRANS.E	2.0	2.0	2.0	0	2.0
Silicon/WSi_2 [9]-[12]					
TRANS.0	2.0e-2	3.0e-3	3.0e-3	0	3.0e-3
TRANS.E	0	0	0	0	0
Polysilicon/WSi_2					
TRANS.0	2.0e-2	3.0e-3	3.0e-3	0	3.0e-3
TRANS.E	0	0	0	0	0

Table A-11 Interface Trap Impurities (CM.SEC units)

	Boron(15)	Phosphorus(16)	Arsenic (17)	Nitrogen
Oxide/Silicon, and Oxide/Polysilicon				
TRANS.0	0.166	7.15	0.231	4.65E-10
TRANS.E	0.486	1.75	0.766	0.0
SEG.0	**/SEG.0** ÷ **SEG.0** (Table A-5)			0.0
SEG.E	**/SEG.E** - **SEG.E** (Table A-5)			0.0
RATIO.0	0	0	0	0.0
RATIO.E	0	0	0	0.0
/TRANS.0	0.0166	0.715	0.0231[18]	0.0
/TRANS.E	0.486	1.75	0.766[18]	0.0
/SEG.0	**/RATIO.0** ÷ **Q.MAX.0** × C_{ss}		5.96E3	0.0
/SEG.E	**/RATIO.E** - **Q.MAX.E**		-0.285	0.0
/RATIO.0	0.178	4.00E-3	6.29E-5	0.0
/RATIO.E	-0.086	-0.37	-0.738	0.0
Q.INI.0	0	0	0	0.0
Q.INI.E	0	0	0	0.0
Q.MAX.0	2.0E14[18]	6.8E14[17]	2.0E14[18]	2.0E14
Q.MAX.E	0	0	0	0.0
TWO.PHAS	false	false	false	true

Polysilicon Grain Segregation

Table A-12 Parameters for Polysilicon Grain Interior/Boundary Segregation (18)

	Boron	Phosphorus	Arsenic	Indium	Antimony
Q.SITES	2.5e15	2.5e15	2.5e15	0	2.5e15
CG.MAX	5.0e22	5.0e22	5.0e22	0	5.0e22
GSEG.0	12.0	1.25	0.6	0	1.0
GSEG.E	0	0.443	0.414	0	0
GSEG.INI	1.0	1.0	1.0	0	1.0
FVELB.0	4.0	4.0	4.0	0	4.0
FVELB.E	0.0	0.0	0.0	0	0.0
VELIF.0	1.0e7	1.0e7	1.0e7	0	1.0e7
VELIF.E	3.0	3.0	3.0	0	3.0
Polysilicon diffusivity enhancement factors (19)					
FGB	2.64	95.1	110.	100.	41.8

Clustering and Solid Solubility

Table A-13 Coefficients for Clustering in Silicon (20)

	Arsenic
CTN.0	1.03103e-17 [1]
CTN.E	-0.4 (21)
CTN.F	4.0

Table A-14 Solid Solubility in Silicon

Temperature (°C)	Boron	Phosphorus	Antimony
650	1.70e19 [19]	1.20e20 [20]	1.70e19 [20]
700	1.70e19	1.20e20	1.70e19
800	4.40e19	2.90e20	2.30e19
900	9.50e19	6.00e20	3.10e19
1000	1.70e20	1.00e21	4.00e19
1100	2.20e20	1.20e21	4.90e19
1200	2.20e20	1.25e21	5.90e19
1300	1.40e20	1.10e21	6.80e19
1350	1.40e20	1.10e21	6.80e19

Table A-15 Transient activation parameters (Extended Defects AAM)

	Boron	Phosphorus	Arsenic	Antimony
T.ACT.0	8.0e-16	8.0e-16	8.0e-16	8.0e-16
T.ACT.E	-4.2	-4.2	-4.2	-4.2
ACT.MIN	1.0	2.0	1.0	1.0
CL.INI.A	true	true	true	true

Table A-16 Advanced Activation Model (`ACT.FULL`)

	All Impurities
DDC.T.0	0
DDC.T.E	0
DDC.F.0	0
DDC.F.E	0
DDCF.N.N	0
DDCF.I.N	0
DDCF.D.N	0
DDCR.N.N	0
DDCR.I.N	0
IFRACM	0
DDCS.0	0
DDCS.E	0
DDCS.N	0 **(22)**
IFRACS	0

	Boron	Phosphorus	Arsenic	Antimony
C0.TF.0	8.0e-16	8.0e-16	8.0e-16	8.0e-16
C0.TF.E	-4.2	-4.2	-4.2	-4.2
C0.TR.0	8.0e-16	8.0e-16	8.0e-16	8.0e-16
C0.TR.E	-4.2	-4.2	-4.2	-4.2
ACT.NI	1.0	2.0	1.0	1.0
ACT.AMOR	3.0e+20	5.0e+22	5.0e+22	5.0e+22
MAX.AMOR	5.0e+22	5.0e+22	5.0e+22	5.0e+22
C0.INI.F	0	1.0	1.0	1.0
C1.INI.F	0	0	0	0
C1FRAC.A	0	0	0	0

Point Defect Parameters

Table A-17 Point Defect Parameters in Silicon (`CM.SEC` units) [21]

	Vacancy	Interstitial
D.0	3.65e-4	3.65e-4
D.E	1.58	1.58
DC.0	1.0	1.0
DC.E	0	0
VD	0	0
D.FACTOR	1	1
D.11	1	1
D.22	1	1
KB.0	**(23)**	1.0e-21

Table A-17 Point Defect Parameters in Silicon (CM.SEC units) [21] (Continued)

	Vacancy	Interstitial
KB.E		-1.0 **(24)**
KIV.0		**(25)**
KIV.E		0
KIV.NORM		true
CEQUIL.0	1.25e29	1.25e29
CEQUIL.E	3.26	3.26
CEQUIL.F	1	1
VCEQUIL	-5.773	8.593
VMOLE	0.0	5.0e22
DNEG.0	32.47	0
DNEG.E	0.62	0
NEG.0	5.68	5.68
NEG.E	0.145	0.50
NEU.0	1.0	1.0
NEU.E	0	0
POS.0	5.68	5.68
POS.E	0.455	0.26
DPOS.0	0	0
DPOS.E	0	0

Table A-18 Vacancy Parameters at Interfaces With Silicon (CM.SEC units)

	Oxide [21]	Nitride	Oxynitride	Ambient	$TiSi_2/WSi_2$
KSURF.0	4.0e-11	3.5e-3	0.1	2.5e-4	4.0e-10
KSURF.E	-1.75	0.0	0.0	0.0	-1.75
KSVEL.0	4.0e-11	3.5e-3	0.1	2.5e-4	4.0e-10
KSVEL.E	-1.75	0.0	0.0	0.0	-1.75
GROWTH	false	false	false	false	false
A.0	0.0	0.0	0.0	0.0	0.0
A.E	0.0	0.0	0.0	0.0	0.0
T0.0	100.02	100.02	100.02	100.02	100.02
T0.E	0.0	0.0	0.0	0.0	0.0
GPOW.0	0.0	0.0	0.0	0.0	0.0
GPOW.E	0.0	0.0	0.0	0.0	0.0
KPOW.0	0.5	0.5	0.5	0.5	0.5
KPOW.E	0.0	0.0	0.0	0.0	0.0
THETA.0	0.0	0.0	0.0	0.0	0.0
THETA.E	0.0	0.0	0.0	0.0	0.0

Table A-19 Interstitial Parameters at Interfaces With Silicon (CM.SEC units)

	Oxide [22] **(26)**	Nitride	Oxynitride	Ambient	$TiSi_2/WSi_2$
KSURF.0	1.4e-6	3.5e-3	0.1	2.5e-4	4.0e-10

Table A-19 Interstitial Parameters at Interfaces With Silicon (CM.SEC units) (Continued)

	Oxide [22] (26)	Nitride	Oxynitride	Ambient	$TiSi_2$/WSi_2
KSURF.E	-1.75	0.0	0.0	0.0	-1.75
KSVEL.0	0.0	3.5e-3	0.1	2.5e-4	0.0
KSVEL.E	-1.75	0.0	0.0	0.0	-1.75
GROWTH	true	false	false	false	false
A.0	0.0	0.0	0.0	0.0	0.0
A.E	0.0	0.0	0.0	0.0	0.0
T0.0	0.0	100.02	100.02	100.02	100.02
T0.E	0.0	0.0	0.0	0.0	0.0
GPOW.0	-0.7	0.0	0.0	0.0	0.0
GPOW.E	0.0	0.0	0.0	0.0	0.0
KPOW.0	0.0	0.5	0.5	0.5	0.5
KPOW.E	0.0	0.0	0.0	0.0	0.0
THETA.0	0.01	0.0	0.0	0.0	0.0
THETA.E	0.0	0.0	0.0	0.0	0.0

Table A-20 Point Defect Parameters at Silicon/Polysilicon Interfaces (CM.SEC units) (27)

	Interstitial	Vacancy
KSURF.0	2.73	6.4e-5
KSURF.E	1.37	0.92

Table A-21 Interstitial Traps Parameters

TRAP.CON	0.0
F.TRAP.0	0.0
F.TRAP.E	0.0
K.TRAP.0	0.0
K.TRAP.E	0.0

Table A-22 Interface Injection/Recombination Model

	Model	V.NORM.0	V.NORM.E
Vacancies	V.MAXOX		
Interstitials	V.NORM	1.0	0.0

Table A-23 Dislocation Loop Coefficients

KLOOP.0	29.8047 [24]
KLOOP.E	0.4 [25]
RLMIN	100e-8

Table A-24 Interstitial Clustering ({311} Model) Coefficients (28)

CL.KFC.0	5.207e14
CL.KFC.E	3.774
CL.CF	0.9398
CL.IFC	1
CL.ISFC	1
CL.KFI.0	0
CL.KFI.E	3.774
CL.KFCI	1.0
CL.IFI	2
CL.ISFI	2
CL.KR.0	9.431e13
CL.KR.E	3.017
CL.CR	1

Table A-25 Small Point Defect Clusters

	Interstitial	Vacancy
ECLUST.0	0	0
ECLUST.E	0	0
ECLUST.N	0	0
KCV.0	0	
KCV.E	0	
KIC.0		0
KIC.E		0

Oxidation

Table A-26 Ambient Definitions

	DRYO2	WETO2	N2O	STREAM	INERT
F.O2	1.0	0.0	0.0	0.0	0.0
F.H2O	0.0	0.92 [26] **(30)**	0.0	1.0	0.0
F.N2O	0.0	0.0	1.0	0.0	0.0
F.H2	0.0	0.0	0.0	0.0	0.0
F.N2	0.0	0.08	0.0	0.0	1.0
F.HCL	0.0	0.0	0.0	0.0	0.0
PRESSURE	1.0	1.0	1.0	1.0	1.0
HCL (%)	0.0	0.0	0.0	0.0	0.0

Table A-27 Linear Oxidation Rate Coefficients

	O2	H2O	N20
L.LIN.0			
<111>	1.038e5 [26]	3.450e4 [27]	8.06
<110>	8.650e4 (30)	2.870e4 (30)	8.22
<100>	6.176e4	2.058e4	5.87
POLYSILI	8.650e4 (31)	2.870e4 (31)	8.22
L.LIN.E			
<111>	2.0	1.60	1.09
<110>	2.0	1.60	1.09
<100>	2.0	1.60	1.09
POLYSILI	2.0	1.60	1.09
H.LIN.0			
<111>	1.038e5	2.950e6	8.06
<110>	8.650e4 (30)	2.457e6 (30)	8.22
<100>	6.176e4	1.755e6	5.87
POLYSILI	8.650e4 (31)	2.457e6 (31)	8.22
H.LIN.E			
<111>	2.0	2.05	1.09
<110>	2.0	2.05	1.09
<100>	2.0	2.05	1.09
POLYSILI	2.0	2.05	1.09
LIN.BREA	0.0	900.0	0.0
LIN.PDEP	0.75	1.0	2.48
GAMMA.0 [28]	2.63e3	2.63e3	1.59e3
GAMMA.E	1.1	1.1	1.3
GB.FAC.0	10	10	10
GB.FAC.E	0	0	0

Table A-28 Parabolic Oxidation Rate Coefficients

	O2	H2O	N20
L.PAR.0	12.87 [26] (30)	2.83e2 [27] (30)	4.73
L.PAR.E	1.23	1.17	1.42
H.PAR.0	12.87 (30)	7.00 (30)	4.73
H.PAR.E	1.23	0.78	1.42
PAR.BREA	0.0	950.0	0.0
PAR.PDEP	1.0	1.0	1.0

Table A-29 Thin Oxide Growth Coefficients [29]

	O2	H2O	N20
THINOX.0			
<111>	6.58e6	0.0	2.39e2
<110>	5.32e4	0.0	5.37e4

Table A-29 Thin Oxide Growth Coefficients [29] (Continued)

	O2	H2O	N2O
<100>	7.48e6	0.0	5.37e4
POLYSILI	5.32e4 **(31)**	0.0	5.37e4
THINOX.E			
<111>	2.33	0.0	1.09
<110>	1.80	0.0	1.85
<100>	2.38	0.0	1.85
POLYSILI	1.80	0.0	1.85
THINOX.L			
<111>	7.8e-3	1.0	1.2e-3
<110>	6.0e-3	1.0	6.5e-3
<100>	6.9e-3	1.0	6.5e-3
POLYSILI	6.0e-3	1.0	6.5e-3

Table A-30 Oxidation Rate for Chlorine Dependence

For O2 [30] and NO2												
Linear								**Parabolic**				
Temperature:		800	900	1000	1100	1200		800	900	1000	1100	1200
	%HCl											
	0.0	1.00	1.00	1.00	1.00	1.00		1.00	1.00	1.00	1.00	1.00
	1.0	1.76	1.76	1.26	1.63	1.63		1.09	1.09	1.67	1.37	1.37
	3.0	1.79	1.79	1.52	2.26	2.26		1.29	1.29	1.90	1.54	1.54
	5.0	1.82	1.82	1.54	2.29	2.29		1.52	1.52	2.18	1.73	1.73
	7.0	1.85	1.85	1.57	2.33	2.33		1.76	1.76	2.51	1.95	1.95
	10.0	1.89	1.89	1.61	2.39	2.39		2.25	2.25	3.07	2.34	2.34
For H2O [26]												
Linear								**Parabolic**				
Temperature:		800	900	1000	1100	1200		800	900	1000	1100	1200
	%HCl											
	0.0	1.00	1.00	1.00	1.00	1.00		1.00	1.00	1.00	1.00	1.00
	1.0	1.00	1.00	1.00	1.00	1.00		1.00	1.00	1.00	1.00	1.00
	3.0	1.00	1.00	1.00	1.00	1.00		1.00	1.00	1.00	1.00	1.00
	5.0	1.00	1.00	1.00	1.00	1.00		1.00	1.00	1.00	1.00	1.00
	7.0	1.00	1.00	1.00	1.00	1.00		1.00	1.00	1.00	1.00	1.00
	10.0	1.00	1.00	1.00	1.00	1.00		1.00	1.00	1.00	1.00	1.00

Table A-31 Coefficients for Analytical Oxidation Models

INITIAL	0.002
MASK.EDG	-200.0
SPREAD	1.0
NIT.THIC	(no default)
ERF.Q [31]	0.05
ERF.DELT [31]	0.04

Table A-31 Coefficients for Analytical Oxidation Models

ERF.LBB [31]	`8.25e-3*( 1580.3 - TOX )*( FOX^0.67 )*( EOX^0.3 ) *exp( -( ( EN - 0.08 )^2 )/0.06 )`
ERF.H [31]	`402*( 0.445 - 1.75*EN )*exp( -TOX/200 )`

Table A-32 Coefficients for Numerical Oxidation Models

	O2	**H2O**	**N2O**
HENRY.CO	5.0e16	3.0e19	5.0e16
TRANS.0 (32)	1.0e3	1.0e6	1.0e3
THETA (33)	2.2e22	2.2e22	2.2e22

Table A-33 Diffusivity of Oxidizing Species in Ambient (7)

	O2	**H2O**	**N2O**
D.0	3.5e9	2.5e7	3.5e9
D.E	1.25	0.8	1.25

Table A-34 Nitrogen-dependent Oxide Growth Coefficients (cm.sec)

	O2	**H2O**	**N2O**
Oxide/Silicon, Oxide/Polysilicon			
N.G.0	0.0	0.0	5.2e22
N.G.E	0.0	0.0	0.0
N.V.0	0.0	0.0	1.0
N.V.E	0.0	0.0	0.0
N.GPOW.0	0.0	0.0	0.0
N.GPOW.E	0.0	0.0	0.0
N.THIN.0			
<111>	0.0	0.0	1.55e14
<110>	0.0	0.0	1.55e14
<100>	0.0	0.0	1.55e14
N.THIN.E			
<111>	0.0	0.0	0.0
<110>	0.0	0.0	0.0
<100>	0.0	0.0	0.0

Table A-35 Parameters for Stress-Dependent Oxidation

STRESS.D	true
VC	300.0 (oxide), 130.0 (nitride) (34)
VR	15 [42]
VD	75.0 [42]
VT	0.0
VDLIM	1.2

Table A-36 Material Conversion Coefficient

	ALPHA
Silicon/oxide	0.44 [49]
Polysilicon/oxide	0.44 **(38)**
Other pairs of materials	1.0

Diffusion Enhancement in Oxides

Table A-37 Diffusion Enhancement Factor of Boron in Oxides

1+4.55e-22*exp(0.28/kt)*sio+2.23e-20*fluorine

Table A-38 Diffusivity of Silicon in Oxides (`CM.SEC` units)

D.0	13.0
D.E	4.5

Table A-39 Si and SiO Reactions in Oxides

MAT.BULK	oxide	oxide		
/MAT.L			oxide	oxide
/NM.L			0	0
MAT.R			silicon	ambient
NM.R			0	0
IMP.L	iox	sio		
NI.L	1.0	1.0		
EI.L	1.0	1.0		
/IMP.L	O2	O2	iox	iox
/NI.L	0.0	0.0	1.0	1.0
/EI.L	0.5	0.5	1.0	1.0
IMP.R	sio		interst	O2
NI.R	1.0		1.0	0
EI.R	1.0		1.0	1.0
RATE.0	7.673e6	7.673e6	0.01	8.67e7
RATE.E	4.0	4.0	0.0	4.0
EQUIL.0	0	0	2.95e-5	0
EQUIL.E	0	0	-2.19	0

Silicidation

Table A-40 Diffusivity of Silicon in Silicides (`CM.SEC` units)

	$TiSi_2$ (35) [32]	WSi_2 (36) [33]-[35]	$CoSi_2$	$NiSi_2$
D.0	2.0	6.86e3	5.0	5.0
D.E	1.86	2.72	2.30	2.30

Table A-41 Diffusivity of ORSoxygen in Silicides (`CM.SEC` units)

	$TiSi_2$	WSi_2	$CoSi_2$	$NiSi_2$
D.0	15.	15.	15.	15.
D.E	2.56	2.56	2.56	2.56

Table A-42 Titanium Silicidation Reactions

/MAT.L	TiSi2	TiSi2	titanium	TiSi2
/NM.L	0	0	0.5	0
MAT.R	silicon	polysilicon	TiSi2	oxide
NM.R	1.0	1.0	0.5	0
IMP.L	vacancy		silicon	
NI.L	1.0e-3		1.0	
EI.L	0		1.0	
/IMP.L	silicon	silicon		ORSoxygen
/NI.L	1.0	1.0		1.0
/EI.L	1.0	1.0		1.0
RATE.0	1.0e-3	1.0e-3	104.0	1.667e-4
RATE.E	0	0	1.0	0
EQUIL.0	1.0e20	1.0e20	1.0	1.0e22
EQUIL.E	0	0	0	0
R.FAC	max((1-ORSoxygen/1e22),0.)			1

Table A-43 Tungsten Silicidation Reactions [33]-[35]

/MAT.L	WSi2	WSi2	tungsten	WSi2
/NM.L	0	0	0.5	0
MAT.R	silicon	polysilicon	WSi2	oxide
NM.R	1.0	1.0	0.5	0
IMP.L	vacancy		silicon	
NI.L	1.0e-3		1.0	
EI.L	0		1.0	
/IMP.L	silicon	silicon		ORSoxygen
/NI.L	1.0	1.0		1.0
/EI.L	1.0	1.0		1.0
RATE.0	1.0e-1	1.0e-1	4.60e3 (37)	1.667e-4
RATE.E	0	0	1.64 (37)	0

Table A-43 Tungsten Silicidation Reactions [33]-[35] (Continued)

EQUIL.0	1.0e20	1.0e20	1.0	1.0e22
EQUIL.E	0	0	0	0
R.FAC	max((1-ORSoxygen/1e22),0.)			1

Table A-44 Cobalt Silicidation Reactions

/MAT.L	CoSi2	CoSi2	Cobalt	CoSi2
/NM.L	0	0	0.5	0
MAT.R	silicon	polysilicon	CoSi2	oxide
NM.R	1.0	1.0	0.5	0
IMP.L			silicon	
NI.L			1.0	
EI.L			1.0	
/IMP.L	silicon	silicon		ORSoxygen
/NI.L	1.0	1.0		1.0
/EI.L	1.0	1.0		1.0
RATE.0	1.667e-4	1.667e-4	1.0e4	1.667e-4
RATE.E	0	0	1.84	0
EQUIL.0	2.34e22	2.34e22	1.0	3.0e21
EQUIL.E	0	0	0	0
R.FAC	max((1-ORSoxygen/1e22),0.)			1

Table A-45 Nickel Silicidation Reactions

/MAT.L	NiSi2	NiSi2	Nickel	NiSi2
/NM.L	0	0	0.5	0
MAT.R	silicon	polysilicon	NiSi2	oxide
NM.R	1.0	1.0	0.5	0
IMP.L			silicon	
NI.L			1.0	
EI.L			1.0	
/IMP.L	silicon	silicon		ORSoxygen
/NI.L	1.0	1.0		1.0
/EI.L	1.0	1.0		1.0
RATE.0	1.667e-4	1.667e-4	1.0e4	1.667e-4
RATE.E	0	0	1.84	0
EQUIL.0	2.34e22	2.34e22	1.0	1.0e22
EQUIL.E	0	0	0	0
R.FAC	max((1-ORSoxygen/1e22),0.)			1

Analytical Implant

Table A-46 Effective +n Model (in Silicon)

D.PHDF	0.0905
D.PME	0.85

Table A-46 Effective +n Model (in Silicon) (Continued)

D.PLF	-2
D.PLME	-0.5
D.PCONST	3.25
D.PDME	-0.5
D.PDLF	-2
D.PDLME	-0.5
D.PNORM	3e12
D.PDE	1
D.PRE	0

Table A-47 Taurus Analytic Implant Model Parameters

	Boron	Phosphorus	Indium	Antimony	Fluorine	Nitrogen	Silicon	Aluminum	Others
All materials									
RANGE.F	6.0	5.0	3.0	3.0	6.0	6.0	5.0	5.5	4.0
SIGMA.F	6.0	5.0	3.0	3.0	6.0	6.0	5.0	5.5	4.0
D.RAN.F	0.6	5.0	3.0	3.0	3.0	1.0	5.0	4.0	4.0
D.SIG.F	6.0	5.0	3.0	3.0	6.0	6.0	5.0	5.5	4.0
D.SUP.R	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01
D.SUP.E	0.4	0.4	0.2	0.2	0.4	0.4	0.4	0.4	0.4

MC Implant

Table A-48 Electronic Stopping Power Pre-factor

	Boron	Phosphorus	Arsenic	Antimony	Aluminum	Other
Silicon	2.0799 [45]	2.5	1.7941	1.7456	0	0
Polysilicon	2.0799	1.7516	1.7941	1.7456	0	0
Oxide	1.8424	1.4609	1.4347	1.3674	0	0
Oxynitride	1.8424	1.4609	1.4347	1.3674	0	0
Nitride	1.8424	1.4609	1.4347	1.3674	0	0
Aluminum	2.0281	1.6857	1.7104	1.6564	0	0
Photoresist	1.0032	0.64016	0.5517	0.4957	0	0
SiC	1.1588	0	0	0	1.4341	0
Other (44)		0	0	0		0
ES.100 [46]-[48]						
Silicon	1.500	1.7516	1.7941	1.7456	0	0
Other	0	0	0	0	0	0
ES.110						
Silicon	0.95	1.7516	1.7941	1.7456	0	
Other	0	0	0	0	0	0

Table A-49 Electronic Stopping Power Pre-factor

	Boron	Phosphorus	Arsenic	Antimony	Aluminum	Other
ES.RAND (43)						
Silicon	2.0799 [45]	2.5	1.7941	1.7456	0	0
Polysilicon	2.0799	1.7516	1.7941	1.7456	0	0
Oxide	1.8424	1.4609	1.4347	1.3674	0	0
Oxynitride	1.8424	1.4609	1.4347	1.3674	0	0
Nitride	1.8424	1.4609	1.4347	1.3674	0	0
Aluminum	2.0281	1.6857	1.7104	1.6564	0	0
Photoresist	1.0032	0.64016	0.5517	0.4957	0	0
SiC	1.1588	0	0	0	1.4341	0
Other (44)		0	0	0		0
ES.100 [46]-[48]						
Silicon	1.500	1.7516	1.7941	1.7456	0	0
Other	0	0	0	0	0	0
ES.110						
Silicon	0.95	1.7516	1.7941	1.7456	0	
Other	0	0	0	0	0	0

Table A-50 Electronic Stopping Power Exponent

	Boron	Phosphorus	Arsenic	Antimony	Aluminum	Other
ES.F.RAN						
Silicon	0.5	0.5	0.5	0.5	0.5	0.5
Other	0.5	0.5	0.5	0.5	0.5	0.5

Table A-50 Electronic Stopping Power Exponent (Continued)

ES.BREAK (45)						
Silicon	400	150	5000	0	0	0
Polysilicon	400	150	5000	0	0	0
Oxide	400	150	5000	0	0	0
Oxynitride	400	150	5000	0	0	0
Nitride	400	150	5000	0	0	0
Aluminum	400	150	5000	0	0	0
Photoresist	400	150	5000	0	0	0
SiC	600	0	0	0	1000	0
Other	0	0	0	0	0	0
ES.F.H						
Silicon	0.1	0.4	1.0	0.5	0.5	0.5
Polysilicon	0.1	0.4	1.0	0.5	0.5	0.5
Oxide	0.1	0.4	1.0	0.5	0.5	0.5
Oxynitride	0.1	0.4	1.0	0.5	0.5	0.5
Nitride	0.1	0.4	1.0	0.5	0.5	0.5
Aluminum	0.1	0.4	1.0	0.5	0.5	0.5
Photoresist	0.1	0.4	1.0	0.5	0.5	0.5
SiC	0.1	0.5	0.5	0.5	0.3	0.5
Other	0.5	0.5	0.5	0.5	0.5	0.5
	Boron	**Phosphorus**	**Arsenic**	**Antimony**	**Aluminum**	**Other**
ES.F.100						
Silicon	0.5	0.5	0.5	0.5		0.5
Other	0.5	0.5	0.5	0.5		0.5
ES.F.110						
Silicon	0.5	0.5	0.5	0.5		0.5
Other	0.5	0.5	0.5	0.5		0.5

Table A-51 Improved Electronic Stopping Power Model (Silicon)

	Boron	Phosphorus	Arsenic	Antimony
NLOC.PRE	0.3	0.3	0.3	0.3
NLOC.EXP	0.2	0.2	0.2	0.2
NLOC.MAX	0.6	0.6	0.6	0.6
NLOC.K	1.2	1.2	1.2	1.2
LOC.FAC	0.8	0.8	0.8	0.8
CHAN.CRI	11.54	11.54	11.54	11.54
CHAN.FAC	2.0	2.0	2.0	2.0
DISP.FAC	0.5	0.5	0.5	0.5

Table A-52 Improved Electronic Stopping Power Model (SiC)

	Boron	Phosphorus	Aluminum	Nitrogen
NLOC.PRE	1.0	0.8	0.825	0.9
NLOC.EXP	0	0	0	0
NLOC.MAX	1.0	0.6	1.0	0.6

Table A-52 Improved Electronic Stopping Power Model (SiC)

	Boron	Phosphorus	Aluminum	Nitrogen
NLOC.K	1.25	1.2	1.05	1.2
LOC.FAC	0.8	0.8	0.8	0.8
CHAN.CRI	11.54	11.54	11.54	11.54
CHAN.FAC	2.0	2.0	2.0	2.0
DISP.FAC	0.5	0.5	0.5	0.5

Electrical Parameters

Table A-53 Critical Electric Field (volts/cm) [36] [37]

ECN.MU	6.49e4
ECP.MU	1.87e4

Table A-54 Surface Mobility Degradation Factor

GSURFN	1.0
GSURFP	1.0

Table A-55 Arora Mobility Model [38]

MUN1	88.0
MUN2	1252.0
AN	0.88
CN	1.26e17
EXN1	-0.57
EXN2	-2.33
EXN3	2.4
EXN4	-0.146
MUP1	54.3
MUP2	407.0
AP	0.88
CP	2.35e17
EXP1	-0.57
EXP2	-2.23
EXP3	2.4
EXP4	-0.146

Table A-56 Caughey Mobility Model [39]

MUN.MIN	55.24
MUN.MAX	1429.23
NREFN	1.072e17
NUN	-2.3
XIN	-3.8

Table A-56 Caughey Mobility Model [39]

ALPHAN	0.733
MUP.MIN	49.705
MUP.MAX	479.37
NREFP	1.606e17
NUP	-2.2
XIP	-3.7
ALPHAP	0.70

Table A-57 Mobility Table

Concentration	Mobility at 300K	
(#/cm^3)	Electron	Hole
1e14	1350	495
2e14	1345	495
4e14	1335	495
6e14	1320	495
8e14	1310	495
1e15	1300	491.1
2e15	1248	487.3
4e15	1200	480.1
6e15	1156	473.3
8e15	1115	466.9
1e16	1076	460.9
2e16	960	434.8
4e16	845	396.5
6e16	760	369.2
8e16	720	348.3
1e17	675	331.5
2e17	524	279.0
4e17	385	229.8
6e17	321	203.8
8e17	279	186.9
1e18	252	178.0
2e18	182.5	130.0
4e18	140.6	90.0
6e18	113.6	74.5
8e18	99.5	66.6
1e19	90.5	61.0
2e19	86.9	55.0
4e19	83.4	53.7
6e19	78.8	52.9
8e19	71.6	52.4
1e20	67.8	52.0
2e20	52.0	50.8
4e20	35.5	49.6
6e20	23.6	48.9
8e20	19.0	48.4

Table A-57 Mobility Table (Continued)

Concentration	Mobility at 300K	
1e21	17.8	48.0

Material Coefficients

Table A-58 Semiconductor Material Coefficients

	Silicon	Polysilicon (38)	SiC (39)
DENSITY	2.33	2.33	3.21
AT.NUM	14.0	14.0	10.0
AT.WT	28.069	28.069	20.04
AT.RAD	1.17 (angstroms)	0	
MOL.WT	28.069	28.069	40.08
POLYCRYS	false	true	
NI.0	3.87e16 [40]	3.87e16	
NI.E	0.605	0.605	
NI.F	1.5	1.5	
YOUNG.M	1.87e12 [42]	1.87e12 [42]	
POISS.R	0.28	0.28	
C11.0	16.57e11	0	
C11.1	-1.347e8	0	
C12.0	6.331e11	0	
C12.1	-0.538e8	0	
C44.0	8.022e11	0	
C44.1	-0.607e8	0	
VISC.0	1.0e30	1.0e16	
VISC.E	0.0	0.0	
VISC.X	0.499	0.499	
EPSILON	11.7 [42]	11.7	
E.FIELD	true	true	
AFFINITY	4.17	4.17	
BANDGAP	1.08	1.08	
EGALPH	4.73e-4	4.73e-4	
EGBETA	636	636	
N.CONDUC	2.82e19	2.82e19	
N.VALENC	1.04e19	1.04e19	
NC.F	1.5	1.5	
NV.F	1.5	1.5	
G.DONOR	2.0	2.0	
E.DONOR	0.049	0.049	
G.ACCEP	4.0	4.0	
E.ACCEP	0.045	0.045	
BOLTZMAN	false	false	
IONIZATI	true	true	

Table A-58 Semiconductor Material Coefficients (Continued)

	Silicon	Polysilicon (38)	SiC (39)
QM.BETA	4.1e-8	4.1e-8	
QM.YCRIT	25e-8	25e-8	
TIF.NAME	Si	Poly	SiC
MD.INDEX	1	3	8
ION.PAIR	true	true	
IP.OMEGA	6.0	6.0	
CL.MODEL	true		
MAX.DAMA	1.15e22		
DAM.GRAD	10		
D.PCONST	0.42		

Table A-59 Polycrystalline Material Coefficients (40)

	Polysilicon
F11	1.0
F22	2.0
G.DENS	2.0
GRASZ.0	0.1
GRASZ.E	0
TEMP.BRE	600.
MIN.GRAI	0.005
FRAC.TA	0.5
GEOM	6.0
GAMMA.0	5.6e-6
GAMMA.E	-1.73
GBE.0	0.39125
GBE.H	3.0
GBE.1	0
ALPHA	1.33
DSIP.0	30.0
DSIP.E	5.09
DSIX.0	0.015
DSIX.E	3.89
DSIM.0	16.0
DSIM.E	4.54
DSIMM.0	166.7
DSIMM.E	5.0
NSEG	2.0
TBU.0	1.0
TBU.E	-5.0
TOXIDE	0.0005
DLGX.0	0
DLGX.E	4.4
EAVEL.0	0
EAVEL.E	3.0
GB.THICK	0.0005

Table A-60 Insulator Material Coefficients

	Oxide	HfO2	Nitride	Oxynitride	Ambient
DENSITY	2.235	9.68	3.44	2.235	
AT.NUM (avg)	10.0	29.333	10.0	10.0	
AT.WT (avg)	20.023	70.163	20.030	20.023	
MOL.WT	60.069	210.488	140.21	60.069	
POLYCRYS	false	false	false	false	
YOUNG.M	6.6e11 [42]	5.654e11	3.89e12	3.89e12	
POISS.R	0.17 [42]	0.2	0.3	0.3	
VISC.0	5.25e4 (41)	5.25e4	1.3e6 (41)	1.3e6 (41)	1.0e9 (42)
VISC.E	-2.42 (41)	-2.42	-2.42 (41)	-2.42 (41)	0.0 (42)
VISC.X	0.499	0.499	0.499	0.499	0.3 (42)
EPSILON	3.9 [43]	21	7.5 [44]	7.5	1.0
TIF.NAME	Ox	HfO2	Nit	OxyNit	Ambient
MD.INDEX	-1	-4	-2	-6	-4
IMPL.TAB	oxide	oxide	nitride	oxide	

Table A-61 Other Material Coefficients

	Aluminum	Photoresist	Titanium	TiSi$_2$	Tungsten	WSi$_2$
DY.DEFAU				0.025		0.02
DENSITY	2.702	1.06	4.5	4.0443	19.3	9.857
AT.NUM	13.0	2.875	22.0	16.67	74.0	34.0
AT.WT	26.98	5.125	47.90	34.68	183.85	80.0
MOL.WT	26.98	41.0	47.90	104.038	183.85	240.0
POLYCRYS	false	false	false	false		
YOUNG.M	8.6e11	3.24e10	1.0e13	1.0e13	3.66e12	1.0e11
POISS.R	0.295	0.35	0.3	0.3	0.29592	0.29592
EPSILON	1.0	1.0				
WORKFUNC	4.1		4.80	4.8	4.80	4.80
TIF.NAME	Al	Photo	Ti	TiSi2	W	WSi2
MD.INDEX	-5	-4	-5	-5	-5	-5
IMPL.TAB	aluminum	az-7500	titanium	tisi2	tungsten	wsi2

Table A-62 Other Material Coefficients

	Cobalt	CoSi$_2$	Nickel	NiSi$_2$
DY.DEFAU		0.025		0.025
DENSITY	8.9	4.9	8.91	4.9
AT.NUM	27.0	18.33	28.0	18.67
AT.WT	58.93	38.36	58.69	38.29
MOL.WT	58.93	115.09	58.69	114.69
POLYCRYS	false	false	false	false
YOUNG.M	1.914e12	1.6e12	1.86e12	1.146e12

Table A-62 Other Material Coefficients

	Cobalt	CoSi$_2$	Nickel	NiSi$_2$
POISS.R	0.322	0.25	0.342	0.33
EPSILON				
WORKFUNC	5.0	5.0	5.0	5.0
TIF.NAME	Co	CoSi2	Ni	NiSi2
MD.INDEX	-5	-5	-5	-5
IMPL.TAB	Cobalt	cosi2	nickel	nisi2

Table A-63 Linear Coefficients of Thermal Expansion

Silicon	3.052e-6 + 2*6.206e-10*(T -293)
Oxide	1.206e-7 + 2*2.543e-10*(T -293)
HfO2	1.31e-6
Nitride	3.0e-6
Aluminum	2.438e-5 + 2*6.660e-9*(T -293)
Cobalt	13.8e-06 + 1.332e-8*(T -293)
CoSi2	10.8e-6 + 1.332e-8*(T -293)
Nickel	1.25e-5
Tunsten	4.6e-6
WSi2	8.21e-6
SiC	4.8e-6
Oxynitride	3.0e-6
Photores	5.5e-5
Other materials	0.0

Numerical Methods

Table A-64 Block Solution Parameters

	Impurities	Point Defects	Oxidant	Traps
REL.ERR	0.01	0.01	0.01	0.01
ABS.ERR	1.0e9	1.0e5	1.0e9	1.0e9
Factoring	**TIME.STE**	**TIME.STE**	**TIME.STE**	**TIME.STE**
SYMMETRY	false	false	false	false
Fill	**PART**	**FULL**	**PART**	**NONE**
Block solve	**LU**	**LU**	**LU**	**LU**

Table A-65 System Solution Parameters

MIN.FILL	true
MIN.FREQ	1.1
MF.METH	1

Table A-65 System Solution Parameters

MF.DIST	0.5
System solve	**CG**
BACK	18
BLK.ITLI	20
Integration	**TRBDF**
INIT.TIM	0.002

Table A-66 Models

Oxidation	**VERTICAL**
Point defect	**PD.FERMI**
Bulk recombination	**KB.LOW**
SKIP.SIL	true
OX.REL	1.0e-6
CONTIN.M	2
VE.SMOOT	0.04
V.COMPAT	current version

Table A-67 Program Options

INFORMAT	false
DIAGNOST	false
ECHO	true
EXECUTE	true
Output	**NORMAL**

Automatic Grid Generation Parameters

Table A-68 Automatic Grid Generation Parameters

DX.MIN	0.2		
DX.MAX	0.5		
DX.RATIO	1.5		
DY.SURF	0.05	**LY.SURF**	0.1
DY.ACTIV	0.5	**LY.ACTIV**	4.0
DY.BOT	100.0	**LY.BOT**	200.0
DY.RATIO	1.5		

Table A-69 Global Grid Spacing Factors

GRID.FAC	1.0
DY.OXIDE	0.1
DY.EXACT	true
DY.LOCAL	true

Adaptive Grid Parameters

Table A-70 Dopants

For Antimony, Arsenic, Boron, Indium, and Phosphorus				
	Silicon	**Polysilicon**	**Oxide**	**Other (46)**
REL.ADAP	0.2	0.3	2.0	0.0
ABS.ADAP	1.0e14	1.0e15	1.0e16	0.0
MIN.SPAC	0.025	0.03	0.1	0.0

Table A-71 Silicon in $TiSi_2$, WSi_2, $CoSi_2$, and $NiSi_2$

REL.ADAP	0.1
ABS.ADAP	1.0e15
MIN.SPAC	0.01

Table A-72 O_2 and H_2O in Oxide

	O_2	H_2O
REL.ADAP	0.2	0.2
ABS.ADAP	1.0e14	1.0e17
MIN.SPAC	0.01	0.01

Table A-73 Unrefinement Parameters

	Silicon	**Other (46)**
MAX.SPAC	0.25	0.0

Table A-74 Control Parameters

IMP.ADAP	true
DIF.ADAP	true
OX.ADAPT	true
ERR.FAC	1.0
UNREFINE	0.25

Table A-75 Grid Parameters for Electrical Extraction

E.TSURF	0.005
E.RSURF	1.2
E.DSURF	0.0002
E.REGRID	true

Default Coefficient References

[1] Fair, R. B., *Impurity Doping Processes in Silicon*, ed. F. F. Y. Wang, Amsterdam: North Holland, 1981.

[2] *TMA Newsletter,* June 7, 1984.

[3] C. S. Fuller and J. A. Ditzenberger, “Diffusion of donor and acceptor elements in silicon,” *J. Appl. Phys.*, Vol. 27, pp. 544–53, 1956.

[4] R. N. Ghoshtagore, “Dopant diffusion in silicon. III. Acceptors,” *Phys Rev. B*, Vol. 3, pp. 2507–14, 1971.

[5] T. Skotnicki, L. Guerin, D. Mathiot, M. Gauneau, A. Grouillet, C. D’Anterroches, E. Andre, P. Bouillon, and M. Haond, “Channel engineering by heavy ion implants,” *ESSDERC Tech. Dig.*, pp. 671–4, 1994.

[6] K. Suzuki, H. Tashiro, T. Aoyama, “Diffusion coefficient of indium in Si substrates and analytical redistribution profile model,” *Solid-State Electron.*, Vol. 43, pp. 27–31, 1999.

[7] R. D. Chang, P. S. Choi, D. L. Kwong, D. Y. C. Lie, and G. Wilk, “Diffusion modeling of implanted indium durign rapid thermal annealing,” 1999 Semiconductor Technology CAD Workshop & Exhibition, May 4–6, 1999, Hsinchu, Taiwan, p. 16-1.

[8] M. Ghezzo and D. M. Brown. “Diffusivity Summary of B, Ga, P, As, and Sb in SiO2,” *J. Electrochem. Soc.*, Vol. 120, No. 1, pp. 146-148, Jan. 1973.

[9] C. L. Chu, K. C. Saraswat, S. S. Wong. “Measurement of Lateral Dopant Diffusion in Thin Silicide Layers,” *IEEE Trans. Elec. Dev.,* Vol. 39, No. 10, pp. 2333-2340, Oct. 1992.

[10] V. Probst, H. Schaber, A. Mitwalsky, and H. Kabza. “WSi2 and CoSi2 as Diffusion Sources for Shallow-Junction Formation in Silicon,” *J. Appl. Phys.,* Vol. 70, No. 2, pp. 708-719, July 1991.

[11] M. Y. Tsai, F. M. d’Heurle, C. S. Petersson, and R. W. Johnson. “Properties of Tungsten Silicide Film on Polycrystalline Silicon,” *J. Appl. Phys.,* Vol. 52, No. 8, pp. 5350-5355, Aug. 1981.

[12] G. Giroult, A. Nouailhat, and M. Gauneau.“Study of a WSi2/Polycrystalline Silicon/Monocrystalline Silicon Structure for a Complementary Metal-Oxide-Semiconductor for a Compatible Self-Aligned Bipolar Transistor Emitter,” *J. Appl. Phys.,* Vol. 67, No. 1, pp. 515-523, Jan. 1990.

[13] D. A. Antoniadis, A. G. Gonzalez, and R. W. Dutton. “Boron in Near-Intrinsic <100> and <111> Silicon under Inert and Oxidizing Ambients—Diffusion and Segregation,” *J. Electrochem. Soc.*, Vol. 125, No. 5, pp. 813-819, May 1978.

[14] H. F. Wolf. *Semiconductors*, Interscience, p. 361, 1971.

[15] I. C. Kizilyalli, T. L. Rich, F. A. Stevie, and C. S. Rafferty, “Diffusion parameters of indium for silicon process modeling,” *J. Appl. Phys.*, Vol. 80, no. 9, pp. 4944–7, 1 Nov 1996.

[16] P. H. Langer and J. I. Goldstein.“Boron Autodoping During Silane Epitaxy,” *J. Electrochem. Soc.*, Vol. 124, No. 4, pp. 591-598, April 1977.

[17] F. Lau, L. Mader, C. Mazure, C. Werner, and M. Orlowski. “A Model for Phosphorus Segregation at the Silicon-Silicon Dioxide Interface,” *Appl. Phys. A.*, 49, pp. 671-675, 1989.

[18] H. Vuong, C. Rafferty, S. Eshraghi, J. Lentz, P. Zeitzoff, M. Pinto, and S. Hillenius, “Effects of Oxide Interface Traps and Transient Enhanced Diffusion on the Process Modeling of PMOS Devices,” *IEEE Trans. Electron Devices,* Vol. ED-43, No. 7, pp. 1144-1152, July 1996.

[19] G. L. Vick and K. M. Whittle. “Solid Solubility and Diffusion Coefficients of Boron in Silicon,” *J. Electrochem. Soc.*, Vol. 116, No. 8, pp. 1142-1144, Aug. 1969.

[20] F. A. Trumbore. “Solid Solubilities of Impurity Elements in Germanium and Silicon,” *Bell System Tech. J.*, Vol. 39, pp. 205-233, Jan. 1960.

[21] M. R. Kump. “2-D Computer Simulation of Diffusion in Silicon,” Doctoral Dissertation, Stanford University, Stanford, California, March 1988.

[22] S. T. Dunham. *J. Appl. Phys.*, Vol. 71, No. 1, 1992.

[23] J. M. Poate, D. J. Eaglesham, G. H. Gilmer, H. J. Gossmann, M. Jaraiz, C. S. Rafferty, and P. A. Stolk. “Ion Implantation and Transient Enhanced Diffusion,” *1995 IEDM Tech. Dig.*, pp. 77-80, 1995.

[24] R. Y. S Huang and R. W. Dutton. “Experimental Investigation and Modeling of the Role of Extended Defects During Thermal Oxidation,” *J. Appl. Phys.*, Vol. 74, No. 9, pp. 5821-5827, Nov. 1993.

[25] S. M. Hu. *Defects in Semiconductors,* ed. J. Narayan and T. Y. Tan, Amsterdam: North-Holland, pp. 333-354, 1981.

[26] Deal, B. E., “Thermal Oxidation Kinetics of Silicon in Pyrogenic H2O and 5% HC1/H2O Mixtures,” *J. Electrochem. Soc.*, Vol. 125, No. 4, pp. 576-579, Apr 1978.

[27] R. R. Razouk, L. N. Lie, and B. E. Deal. “Kinetics of High Pressure Oxidation of Silicon in Pyrogenic Steam,” *J. Electrochem. Soc.*, Vol. 128, No. 10, pp. 2214-2220, Oct. 1981.

[28] C. P. Ho and J. D. Plummer. “Si/SiO2 Interface Oxidation Kinetics: A Physical Model for the Influence of High Substrate Doping Levels,” *J. Electrochem. Soc.*, Vol. 126, No. 9, pp. 1516-1522, Sept. 1979.

[29] H. Z. Massoud. “Thermal Oxidation of Silicon in Dry Oxygen—Growth Kinetics and Charge Characterization in the Thin Regime,” Technical Report, Stanford Electronics Laboratories, Stanford University, Stanford, California, June 1983.

[30] D. W. Hess and B. E. Deal. “Kinetics of the Thermal Oxidation of Silicon in O2/HCl Mixtures,” *J. Electrochem. Soc.*, Vol. 124, No. 5, pp. 735-739.

[31] N. Guillemot, G. Pananakakis, and P. Chenevier. “A New Analytical Model of the ‘Bird’s Beak’,” *IEEE Trans. Electron Devices,* Vol. ED-34, No. 7, pp. 744-749, July 1983.

[32] L. S. Hung, *et al. J. Appl. Phys.,* Vol. 54 p. 5076, 1983.

[33] S. L. Zhang, R. Buchta, and M. Ostling. “A Study of Silicide Formation form LPCVD-Tungsten Films: Film Texture and Growth Kinetics,” *J. Mater. Res.,* Vol. 6, pp. 1886-1891, Sept. 1991.

[34] L. D. Locker and C. D. Capio. “Reaction Kinetics of Tungsten Thin Films on Silicon (100) Surfaces,” *J. Appl. Phys.,* Vol. 44, No. 10, pp. 4366-4369, Oct. 1973.

[35] E. Ma, B. S. Lim, M. A. Nicolet, N. S Alvi, and A. H. Hamdi. *J. Electron. Mater.,* Vol. 17, p. 207, 1988.

[36] K. Yamaguchi.“Field-dependent Mobility Model for 2-D Numerical Analysis of MOSFETs,” *IEEE Trans. Electron Devices*, Vol. ED-26, pp. 1068-1074, July 1979.

[37] K. Yamaguchi. “A Mobility Model for Carriers in the MOS Inversion Layer,” *IEEE Trans. Electron Devices*, Vol. ED-30, pp. 658-663, June 1983.

[38] N. D. Arora, J. R. Hauser, and D. J. Roulston. “Electron and Hole Mobilities in Silicon as a Function of Concentration and Temperature,” *IEEE Trans. Electron Devices*, Vol. ED-29, pp. 292-295, Feb. 1982.

[39] D. M. Caughey and R. E. Thomas. “Carrier Mobilities in Silicon Empirically Related to Doping and Field,” *Proc. IEEE*, Vol. 55, pp. 2192-2193, 1967.

[40] F. J. Morin and J. P. Maita. “Electrical Properties of Silicon Containing Arsenic and Boron,” *Phys. Rev.*, Vol. 96, No. 1, pp. 28-35, Oct. 1954.

[41] A. S. Grove. *Physics and Technology of Semiconductor Devices*, New York: John Wiley and Sons, pp. 102-103, 1967.

[42] V. Senez, P. Ferreira, and B. Baccus. “2-D Simulation of Local Oxidation of Silicon: Calibrated Viscoelastic Flow Analysis,” *IEEE Trans. Electron Dev.,* Vol. 43, No. 5, pp. 720–731, May 1996.

[43] *Handbook of Chemistry and Physics*, 53rd Edition, Cleveland: The Chemical Rubber Company, 1972.

[44] R. S. Muller, and T. I. Kamins. *Device Electronics for Integrated Circuits*, New York: John Wiley and Sons, 1977.

[45] J. P. Biersack and L. G. Haggmark. “A Monte Carlo Computer Program for the Transport of Energetic Ions in Amorphous Targets,” *Nucl. Instr. and Meth.*, Vol. 174, p. 257, 1980.

[46] R. G. Wilson. “Random and Channeled Implantation Profiles and Range Parameters for P and Al in Crystalline and Amorphized Si,” *J. Appl. Phys.*, Vol. 59, p. 2797-2805, Oct. 1986.

[47] J. Comas and R. G. Wilson. “Channeled and Random Equivalent Depth Distributions of 150keV Li, Be, and B Implanted in Si,” *J. Appl. Phys.*, Vol. 51, No. 7, pp. 3697-3701, July 1980.

[48] R. G. Wilson. “Channeling of 20-800 keV Arsenic Ions in the <110> and the <100> Directions of Silicon, and the Roles of Electronic and Nuclear Stopping,” *J. Appl. Phys.*, Vol. 52, No. 6, pp. 3985-3988, June 1981.

[49] E. Bassous, H. N. Yu, and V. Maniscalco. “Topology of Silicon Structure with Recessed SiO2,” *J. Electrochem. Soc.*, Vol. 123, No. 11, pp. 1729-1737, Nov. 1976.

[50] T. I. Kamins and E. L. MacKenna. “Thermal Oxidation of Polycrystalline Silicon Films,” *Metallurgical Transactions*, Vol. 2, pp. 2292-2294, Aug. 1971.

[51] P. Fahey, G. Barbuscia, M. Moslehi, and R. W. Dutton, “Kinetics of Thermal Nitridation Processes in the Study of Dopant Diffusion Mechanisms in Silicon,” *Appl Phys. Lett.,* Vol. 46, No. 8, p. 784, April 1985.

[52] T. E. Seidel, et al. “Rapid Thermal Annealing is Si,” *Mat. Res. Symp. Proc*., Vol. 35, p. 329, 1985.

[53] A. D. Buonaquisti, W. Carter, and P. H. Holloway. *Thin Solid Films*, Switzerland, Vol. 100, p. 235, 1983.

[54] T. Kamins, et al. *J. Appl. Phys.,* Vol. 43, p. 83, 1972.

[55] K. Sakamoto, K. Nishi, T. Yamaji, T. Miyoshi, and S. Ushio. *J. Electrochem. Soc.,* Vol. 132, p. 2457, 1985.

[56] W. Rausch, R. Lever, and R. Kastl. *J. Appl. Phys.,* Vol. 54, No. 8, p. 4405, 1983.

[57] H. Schaber, R. Criegern, and I. Weitzel. *J. Appl. Phys,* Vol. 58, No. 11, p. 4036, 1985.

[58] M. M. Mandurah, K. C. Saraswat, R. W. Dutton, and T. I. Kamins. “Dopant Segregation in Polysilicon,” *J. Appl. Phys.*, Vol. 51, p. 5755, 1981.

[59] M. J. van Dort, et al. *IEDM Tech. Digest,* p. 865, 1994.

[60] P. B. Griffin. “Physics and Modeling of 2-D Diffusion in SUPREM-IV,” Doctoral Dissertation, Stanford University, Palo Alto, California, Dec. 1989.

[61] A. G. O’Neill, C. Hill, J. King, and C. Please. “A New Model for the Diffusion of Arsenic in Polycrystalline Silicon,” *J. Appl. Phys.*, Vol. 64, No. 1, p. 167, 1988.

[62] C. Hill and S. Jones. “Growth and Microcrystalline Structure of Poly-Si,” *Properties of Silicon: EMIS Data Review*, No. 4, Inspec., p. 933, 1988.

[63] C. P. Ho, J. D. Plummer, S. E. Hansen, and R. W. Dutton. "VLSI Process Modeling—SUPREM III," *IEEE Trans. Elec. Dev.,* Vol. ED-30, No. 11, p. 1438, 1983.

[64] L. Mei and R. W. Dutton. "A Process Simulation Model for Multilayer Structures Involving Polycrystalline Silicon," *IEEE Trans. Elec. Dev.,* Vol. ED-29, No. 11, p. 1726, 1982.

[65] D. Gupth, D. R. Campbell, and P. S.Ho. *Thin Films—Interdiffusion and Reaction,* New York: John Wiley and Sons, p. 161, 1980.

[66] R. B. Fair. "Concentration Profiles of Diffused Dopants in Silicon," *Impurity Doping Processes in Silicon,* ed. F. F. Y. Wang, Amsterdam: North Holland, 1981.

Default Coefficient Notes

The notes listed in this section appear in Default Coefficient Values, pp. A-1 through A-28.

(1) The ratio of the interstitial component to the vacancy component of diffusion in silicon has been set according to the fractional interstitialcy factors given in [51]. In materials other than silicon, the vacancy component is zero.

(2) The diffusivity of indium in silicon is taken to be an average of several reported values [3]–[6]; larger values (1.6exp(-3.5/kT)) have been reported for RTA[7].

(3) Due to the wide spread in reported diffusivities in oxide, the values for phosphorus and arsenic diffusivity in oxide were taken to be an average of those reported in reference [8]. Diffusivity in oxide is reported to be very sensitive to diffusion ambient and impurity concentration, but this dependence is not well characterized and is not modeled in TSUPREM-4. In particular, diffusion from a gaseous source into oxide can result in a "melt-through" process with a much larger effective diffusivity than the value used in TSUPREM-4. The diffusivity of indium in oxide is known to be very fast; it has been approximated here as 10× the diffusivity in silicon.

(4) These values are for diffusion in the interior of a polysilicon grain. The diffusivities for boron, phosphorus, and antimony are identical to the values for single-crystal silicon. The values for arsenic are the same as for silicon but with the addition of the **`DIMM.0`** and **`DIMM.E`** terms. These values were taken from the data of Siedel et al. [52] for bulk single-crystal silicon at high concentration.

(5) The diffusivity of boron is taken from the analysis of Buonaquisti et al. [53] of data measured by Kamins et al. [54], but the activation energy has been increased to that of single-crystal to reproduce measured values at low temperature. For phosphorus, the activation energy for single-crystal silicon is used, while the pre-exponential factor is taken to reproduce measured data at low temperature. For arsenic, the activation energy has been taken from Sakamoto et al. [55] and the pre-exponential factors have been fitted to diffusion profiles at 900° C for various implant doses. For antimony, a diffusivity of 10^4 times the value in single-crystal silicon has been assumed. Due to lack of data, the polycrystalline diffusion model has not been enabled for indium in polysilicon.

(6) Tungsten silicide is assumed to be present in polycrystalline form, with diffusion dominated by grain boundary effects. The diffusivity within the crystalline grains is used here, however. The diffusivity can be increased to approximate the effect of diffusion along grain boundaries, but numerical difficulties may arise if very large values are used.

(7) The diffusivity of impurities in ambient is used for calculating impurity diffusion through included voids in a structure. These values were chosen to be an approximate order of magnitude larger than the diffusivity in solid materials at normal processing temperatures.

(8) The diffusivities of pairs are estimated to be the same as the diffusivities of the corresponding point defects. Arbitrary (but physically reasonable) values have been chosen for the capture radii and barrier energies.

(9) Reported values for segregation constants of n-type impurities between silicon and silicon dioxide range between 10 and 1000. The value used here is a rough average of the reported values.

(10) Due to the absence of data for the segregation constants between polysilicon and silicon dioxide, these segregation constants were set equal to the segregation constants between silicon and silicon dioxide.

(11) The transport of indium is known to be very fast; the value used here is computed by dividing the diffusivity (in the faster-diffusing material) by a distance of 1 angstrom.

(12) The values for transport constants between silicon and ambient are taken from the CASPER process modeling program referred to in reference [16]. The values are meant to characterize the impurity transport across a microscopically-clean silicon surface, as present during epitaxial deposition. These values may be too large if a native oxide is present on the surface of the silicon.

(13) Due to the absence of data for the transport constants between silicon dioxide and ambient and between polysilicon and ambient, these transport constants were set equal to the transport constants between silicon and ambient.

(14) The transport rate between polysilicon and silicon has been set sufficiently high, so that it is not the rate-limiting step in diffusion between the two materials. In the case where the behavior of a thin oxide diffusion barrier between the polysilicon and silicon must be modeled, the oxide layer should be explicitly included in the structure.

(15) The values for constants between silicon dioxide and silicon are extracted to fit the SIMS data for 30KeV~70KeV, 1E12/cm^2~1E14/cm^2 ion implantations followed by thermal processes for several different time conditions at 900^{o}C~1200^{o}C. Due to the absence of data for the constants between silicon dioxide and polysilicon, these constants were set equal to the constants between silicon dioxide and silicon.

(16) The values for constants between silicon dioxide and silicon are extracted to fit the Lau et.al. data [17] and the SIMS data for 30KeV and 40 KeV 1E14/cm^2 ion implantation followed by the RTA for several different time conditions at 800^{o}C~1000^{o}C. The value of `D.PLUS` parameter on the implantation is set to 1.5 for the simulation. Due to the absence of data for the constants between silicon dioxide and polysilicon, these constants were set at the same values as the constants between silicon dioxide and silicon.

(17) The values for constants between silicon dioxide and silicon are extracted to fit the SIMS data for 20KeV 5E14/cm^2 and 32KeV 1E15/cm^2 ion implantations followed by the RTA for several different time conditions at 1000^{o}C~1050^{o}C. The value of **`MAX.DAM`** parameter on the implantation is set to 1.15E22/cm^3 and the values for the clustering coefficients, **`CTN.0`** and **`CTN.E`** are set to 5.35E-20 and -0.874, respectively. Due to the absence of data for the constants between silicon dioxide and polysilicon, these constants were set at the same values as the constants between silicon dioxide and silicon.

(18) The values for boron are appropriate for concentrations below solid solubility and are derived from the results of Rausch et al. [56] which are confirmed by Schaber et al. [57] to match measured polysilicon/silicon interfacial segregation. The values for phosphorus and arsenic are from Mandurah et al. [58], except that **`GSEG.0`** has been reduced to match measured polysilicon/silicon interfacial segregation. The initial segregation parameters **`GSEG.INI`** are set to unity so that there is no initial segregation of dopant into the grain boundary after implant. The interfacial segregation velocities have been set to large values.

(19) The default values of **`FGB`** were chosen to give approximately the same low-concentration diffusivity in polysilicon at 1000° C as the default values used in versions of TSUPREM-4 prior to Version 6.3.

(20) The relationship between electrically active and chemical concentrations is assumed to be an equilibrium relationship that is attained immediately. For high concentrations of arsenic in silicon it has been found that the equilibrium clustering relationship may actually require many hours to be reached at low diffusion temperatures.

(21) The activation energy for the arsenic clustering coefficient has not been well characterized. The value for the activation energy is such that this coefficient agrees with the value in reference [1] at a temperature of 1050° C.

(22) `DDCS.N` for boron has been set to 1.0 so that parameter sets for modeling BICs in Version 1999.2 will give the same results in Version 1999.4.

(23) In Versions 6.0 and above the rate of bulk recombination between interstitials and vacancies is determined by the values of **`KB.0`** and **`KB.E`** specified on the **`INTERSTITIAL`** statement. The values of **`KB.0`** and **`KB.E`** specified on the **`VACANCY`** statement are ignored.

(24) The bulk recombination rate between interstitials and vacancies was chosen to give good results at lower temperatures. The value has been reduced at high temperatures to speed up convergence of the solution algorithms.

(25) The **`KIV.0`** are set indirectly by specifying **`KB.LOW`**. See Recombination of Interstitials with Vacancies, p. 3-64.

(26) A surface recombination velocity that is independent of oxidation rate was chosen to fit the results reported by Professor Scott Dunham of Boston University.

(27) These values have been chosen to reproduce the values of K_s/D reported by van Dort [59] for the silicon/oxide interface at 900° C. The activation energies are chosen to match the values of K_s/D reported by Griffin [60] for the silicon/oxide interface.

(28) These values were extracted to fit the data of Poate, et al. [23] using the default parameters for point defect kinetics. Only data for a single silicon implant at a single dose and energy were used; these values may need to be adjusted to provide satisfactory results for other implant conditions.

(29) Wet oxygen produced from a 95° C bubbler at one atmosphere has in the past been thought to produce an effective oxidant pressure equal to the vapor pressure of water at 95° C: 640 torr, which is equal to 0.84 atmospheres. More recent work [27] has demonstrated oxidation rates closely approximating that of a bubbler by using a pyrogenic steam process with O_2 and H_2 flow rates of 1.175 and 2 liters/minute, respectively.Complete reaction for these flow rates would produce an ambient containing 92% H_2O and 8% O_2. Because TSUPREM-4 cannot model ambients containing both H_2O and O_2, the wet oxygen ambient is defined as 92% H_2O and 8% N_2. This effective mixture of H_2O and N_2 will depend upon the gas flow rates, the diffusion furnace configuration, and even the altitude of the fabrication facility. As a result, these values frequently must be adjusted in order to achieve agreement with the oxide thicknesses grown at a particular facility.

(30) Oxidation rates for <110> silicon were not determined in references [23] and [27]. From other studies it has been found that the parabolic oxidation rates are not dependent upon the silicon orientation, but that the linear oxidation rates for <110> silicon are approximately a factor of 1.4 greater than those for <100> silicon. This factor has been used to calculate linear oxidation rates for <110> silicon from the rates for <100> silicon.

(31) The oxidation rates for polysilicon have been set equal to those for <110> silicon [50].

(32) These values are given in centimeters per second.

(33) Only one value of **`THETA`** may be given; the value applies to both O_2 and H_2O.

(34) The default value for **`VC`** is the recommended value at 900° C. Recommended values at other temperatures are 225 at 800° C, 425 at 1000° C, and 1100 at 1100° C and higher; for nitride, use 120 at 800° C, 170 at 1000° C, and 350 at 1100° C and higher.

(35) These values are valid for reaction in argon or vacuum. Value will be smaller at high n-type concentrations.

(36) These values apply when the amount of interfacial oxygen is minimal after tungsten deposition, e.g., after chemical vapor deposition. If there is significant interfacial oxygen between the tungsten and the silicon (e.g., after deposition by sputtering or evaporation), the values **`DIX.0`**=9.89e4 and **`DIX.E`**=3.13 are more recommended.

(37) These values apply when the amount of interfacial oxygen is minimal after tungsten deposition, e.g., after chemical vapor deposition. If there is significant interfacial oxygen between the tungsten and the silicon (e.g., after deposition by sputtering or evaporation), the values **`RATE.0`**=5.22e6 and **`RATE.E`**=2.46 are recommended.

(38) Due to the absence of data for the material properties of polysilicon, the values of many material coefficients were set equal to the coefficients for silicon.

(39) The only models available for silicon carbide in TSUPREM-4 are for ion implantation using the MC implant model.

(40) The polycrystalline microstructure parameters **`F11`**, **`F22`**, and **`G.DENS`** have been set to values appropriate for columnar grains with square cross-section, as in O'Neill et al. [61]. The deposition parameters have been set for the case of LPCVD polysilicon as reported in the review by Hill and Jones [62]. The parameter **`FRAC.TA`** for the recrystallized grain size after amorphous deposition or implant amorphization has been set to approximately reproduce measured values. The grain growth parameters **`GEOM`**, **`GBE.0`**, and **`GBE.H`** are taken directly from the model of Mei and Dutton [63] and [64] (with appropriate adjustments to the units). The silicon self-diffusivity values are as stated in Fair [66] with the exception of **`DSIMM.0`** and **`DSIP.0`**, which have been fitted to the measured doping dependence of the normal grain growth rate. **`GAMMA.0`** and **`GAMMA.E`** have been fit to known undoped normal grain growth rates. The solute drag coefficient **`NSEG`** is set to the value quoted by Gupth et al. [65] and used in the model of Mei and Dutton [64]. The geometric factor relating grain boundary velocity to average grain growth **`ALPHA`** has been set to the value recommended by O'Neill et al. [61]. The interface break-up and epitaxial regrowth parameters have been set to have been set effectively disable the models pending further calibration work.

(41) The viscosities for oxide and nitride were fit to the values reported by Senez et al. [42]. The fits are very good, except for the value for oxide at 900° C (Senez's value is higher) and the value for nitride at 800° C (Senez's value is lower).

(42) The mechanical properties of ambient are used for calculating the deformation of included voids during oxidation. They describe a material that is very soft and compressible compared to solid materials.

(43) The values for the electronic stopping prefactor for amorphous materials and random directions in crystalline silicon, **`ES.RAND`**, are calculated as documented in Chapter 3, Crystalline Implant Model, p. 3-140. The value for **`ES.RAND`** for boron implants has been multiplied by a factor of 1.59 as suggested by experimental data. The value for phosphorus in silicon has been adjusted to 2.5 to provide better agreement with experimentally measured implant profiles. Values for **`ES.100`**, **`ES.110`**, **`ES.F.100`**, and **`ES.F.110`** for boron have been adjusted to correspond with experimental data for intentionally channeled implants contained in references [45] through [47]. The values for these parameters were selected by comparing the results of MC implant calculations with the experimental implant profiles.

(44) When a value of 0.0 is specified for **`ES.RAND`**, **`ES.100`**, or **`ES.110`**, the value is calculated as documented in Chapter 3, Crystalline Implant Model, p. 3-140.

(45) Values of **`ES.BREAK`** and **`ES.F.H`** for silicon were obtained by fitting to measured implant profiles. The same values are assumed for other materials.

(46) The adaptive grid parameters for materials other than silicon, polysilicon, and oxide and for solution variables other than antimony, arsenic, boron, and phosphorus are all set to zero. This disables adaptive grid for these materials and solution variables.

Graphics Devices

Determining the Device

TSUPREM-4 supports a number of graphics terminals and hardcopy plotting devices. The file *s4pcap* contains a description of each device known to the program.

The graphics device to be used is determined as follows:

1. If a valid device name has been specified with the **DEVICE** parameter on the **OPTION** statement (see Chapter 4, OPTION, p. 4-41), that device is used.
2. Otherwise, if the environment variable *DEFPDEV* is defined as the name of a valid plot device, its value is used as the plot device.
3. Otherwise, if the environment variable *TERM* is defined as the name of a valid plot device, its value is used as the plot device.
4. Otherwise, the **DEFAULT** device in the *s4pcap* file is used. *Note* that the *s4pcap* file may be modified to make the “default” device refer to any available real plotting device.

If any step in the sequence above produces an invalid device name, the search continues with the next step.

Output

For interactive display devices (and terminal emulation windows in a windowing system), graphics output is sent to the standard output device (normally the user's terminal); for hardcopy devices, output is sent to a file, as noted in the descriptions below; for windowed displays, output is sent to a window created by the program. In any case, output may be redirected to a user-specified file (for later display) with the **`PLOT.OUT`** parameter on the **`OPTION`** statement. The current output device is closed and a new output device is opened whenever an **`OPTION`** statement with the **`DEVICE`** parameter is executed. This feature can be used to close the current output file or create a new graphics window without exiting TSUPREM-4. *Note*, however, that if the new graphics device specifies an output file that already exists, the old contents of the output file will be lost, unless the file name is preceded by the "+" character in the **`PLOT.OUT`** parameter on the **`OPTION`** statement.

The following sections list the devices defined in *s4pcap* by *Synopsys* TCAD. Names in parentheses are alternate names for the device.

Supported Devices

The following devices are fully supported by Synopsys TCAD:

X (Window)

X-Windows graphics window. Character sequences are generated and piped to the *tmaplot* program, which is executed by the program. Support is provided for 14-color output and filled polygons with a white background. The *tmaplot* program generates a X-Windows graphics window in which graphical output is produced. Multiple graphics images are retained.

If *tmaplot* is executed on a Sun Sparc system *and* the *TMAPLOT_XLIB* environment variable is not set, the display window is controlled through a control panel. Complete help information is provided as part of the user interface. The help information is only available if the directory where the file *studio_view.info* is located is included in the directory list set in the *HELPPATH* environment variable. The display window can also be controlled with the following keys:

- *f*: display next page
- *b*: display previous page
- *w*: print the cursor location in image coordinates
- *d*: use *replot* to convert the current image to a formatted plot file using the device specified by the *TMAPLOT_REPLOT* environment variable (the default is "FORMAT")

If *tmaplot* is executed on any system other than a Sun Sparc *or* the *TMAPLOT_XLIB* environment variable is set, no control panel is available. The display window can be controlled with the following keys and mouse buttons:

- *left mouse button*: display next page
- *right mouse button*: display previous page
- *z*: Zoom—magnify the image by a factor of 2 and center it at the cursor location
- *Z*: Unzoom—demagnify the image by a factor of 2 and center it at the cursor location
- *p*: Pan—center the image at the cursor location
- *r*: Reset—restore the image to its initial state
- *u*: Undo—undo the previous Zoom, Unzoom, Pan, or Reset operation
- *w*: print the cursor location in image coordinates
- *d*: use *replot* to convert the current image to a formatted plot file using the device specified by the *TMAPLOT_REPLOT* environment variable (the default is “FORMAT”)
- *control-C*: exit

This version of the *tmaplot* program is only provided with versions of TSUPREM-4 for the UNIX operating system.

If STUDIO has been installed, specifying the X Window device activates STUDIO View. Instructions for using STUDIO View are provided in the STUDIO reference manual.

I/X Like **X**, but with white lines on a black background.

X/BW Like **X**, but uses software crosshatch patterns instead of color area fill.

POSTSCRIPT (PS,PS-P) Apple LaserWriter and other PostScript printers, with grayscale for area fills. Output is sent to file *plotfile.ps*. This driver plots in “portrait” mode (i.e., with the short side of the paper along the *x* axis and the long side along the *y* axis).

L/POSTSCRIPT (PS-L) Same as **POSTSCRIPT**, except that it plots in “landscape” mode (i.e., with the long side of the paper along the *x* axis and the short side along the *y* axis.

PS-INSERT Same as **POSTSCRIPT**, except that no output sequences are generated for displaying the plot. The output file produced by this driver can be included in other PostScript documents.

C/POSTSCRIPT (PS-C,PS-CP)	Color version of the `POSTSCRIPT` driver (portrait mode).
CL/POSTSCRIPT (PS-CL)	Color version of the `L/POSTSCRIPT` driver (landscape mode).
C/PS-INSERT	Color version of the `PS-INSERT` driver.
REPLOT	Produces a text file *s4ofil*,which can be read by *Synopsys* TCAD's REPLOT utility program. The REPLOT program can redisplay the plot on a number of graphics devices and can interface to graphics libraries such as DISSPLAY, DI-3000, and GDDM.
HP2648 (2648)	Hewlett-Packard 2648 (or similar) graphics terminals.
HP2623 (2623)	Hewlett-Packard 2623 (or similar) graphics terminals.
TEK4100 (4100)	Tektronix 4100-series color graphics terminals.
TEKBW	Same as `TEK4100`, except uses software-defined crosshatch patterns instead of different colors for area fills (i.e., the `COLOR` statement). (This entry shows how selected hardware capabilities may be disabled for a device.)
TEK4010 (4010)	Tektronix 4010-series graphics terminals.
XTERM	Puts graphics in the Tektronix-compatible window of the *xterm* terminal emulator available with some versions of X Window system.
REGIS (VT240, VT241)	DEC and other REGIS-compatible graphics terminals.
HP7550 (7550)	Hewlett-Packard 7550 (and other HPGL) pen plotters. Output sent to file *plotfile.hpp*. This driver plots in "landscape" mode (i.e., with the long side of the paper along the x axis and the short side along the y axis).
HP7550-P (7550-P)	Same as `HP7550`, except that it plots in "portrait" mod (i.e., with the short side of the paper along the x axis and the long side along the y axis).
PRINTRONIX	Printronix line printer in plot mode. Output is sent to file *plotfile.lp*.

SELANAR

Selanar HiREZ 100XL and some Plessey graphics terminals. (This driver can be easily modified to support other terminals that switch between text and Tektronix-compatible graphics modes.)

SUN (SUNVIEW)

SunView graphics window. Character sequences are generated and piped to the *tmaplot* program, which is executed by the program. Support is provided for 14-color output and filled polygons with a white background. The *tmaplot* program generates a SunView graphics window in which graphical output is produced. Multiple graphics images are retained. The display window can be controlled with the following keys and mouse buttons:

- *left mouse button*: display next page
- *right mouse button*: display previous page
- *z*: Zoom—magnify the image by a factor of 2 and center it at the cursor location
- *Z*: Unzoom—demagnify the image by a factor of 2 and center it at the cursor location
- *p*: Pan—center the image at the cursor location
- *r*: Reset—restore the image to its initial state
- *u*: Undo—undo the previous Zoom, Unzoom, Pan, or Reset operation
- *w*: print the cursor location in image coordinates
- *d*: use *replot* to convert the current image to a formatted plot file using the device specified by the *TMAPLOT_REPLOT* environment variable (the default is “FORMAT”)
- *control-C*: exit

This version of the *tmaplot* program is only provided with versions of TSUPREM-4 for the UNIX operating system. A Sun version of the *tmaplot* program which supports SunView graphics must be executed on the local Sun computer, which controls the monitor display.

I/SUN

Like `SUN`, but with white lines on a black background.

APOLLO

Apollo GPR frame mode window. Character sequences are generated and piped to the *tmaplot* program, which is executed by the program. Support is provided for 8-color output and filled polygons with a white background. The *tmaplot* program generates a GPR graphics window, in which graphical output is produced. Multiple graphics images are retained. The display window can be controlled with the following keys:

- *downward vertical scroll*: display next page
- *upward vertical scroll*: display previous page
- *exit* or *abort*: exit

This version of the *tmaplot* program is only provided with versions of TSUPREM-4 for the UNIX operating system. An Apollo version of the *tmaplot* program must be executed on the local Apollo computer, which controls the monitor display.

I/APOLLO Like `APOLLO`, but with white lines on a black background.

Unsupported Devices

The *s4pcap* file provided by *Synopsys* TCAD defines a number of devices not described above. These devices are not supported by *Synopsys* TCAD, but are provided for the benefit of those who may find use for them. Many of these devices have not been tested. If you try one of these devices and find that it works satisfactorily, let us know at *Synopsys* TCAD. If you need one of these devices and are willing to help us make it work, let us know; we may be able to add it to the list of supported devices.

The following devices are not supported, but may be useful:

TEK4510 (4510) Tektronix 4510 rasterizer. Output is sent to file *plotfile.tek*.

HPJET (THINKJET) Hewlett-Packard Thinkjet printer. Output is sent to file *plotfile.hpj*.

HPDESK (HP2671G, HP2673) Hewlett-Packard 2671G and 2673A desktop printers.

HPLP (LP2563) Hewlett-Packard 256X series line printers. Output is sent to file *lp.out*.

IMAGEN Imagen laser printer, using ImPress. Output is sent to file *plotfile.ip*.

DITROFF Generates ditroff (device-independent troff) codes.

TGPLOT Generates a binary file, containing plotting information. On some machines, these files are compatible with *Synopsys* TCAD's REPLOT program. Output is sent to file *s4pfil*. This driver plots in “landscape” mode (i.e., with the long side of the paper along the *x* axis and the short side along the *y* axis).

TGPLOT-P Same as `TGPLOT`, except that it plots in “portrait” mode (i.e., with the short side of the paper along the *x* axis and the long side along the *y* axis).

The Default Device

DEFAULT A special device, used to specify a default graphics device from within the *s4pcap* file. The *s4pcap* entry for **`DEFAULT`** should consist of a link to one of the devices listed above (using the **`LIKE`** keyword).

Modifying *s4pcap*

CAUTION

Modify *s4pcap* at your own risk. Mistakes in the format of the file can produce some rather mysterious and unhelpful messages. Always save a good copy of the file before making any changes. And be sure that every device entry and every parameter in each entry ends with a ":" character.

You can modify the device descriptions in *s4pcap*. At present, complete documentation on the format of this file is not available, but some helpful hints are given in the following paragraphs, and much can be inferred by examining the existing entries. Some of the most useful parameters are the following:

PIXX and PIXY **`PIXX`** and **`PIXY`** specify the total number of addressable points in the *x* and *y* directions. Changing these values changes the size of the plotting area available to the program. (*Note* that some devices may require corresponding changes to the **`INIT`** entry.)

PUNX and PUNY **`PUNX`** and **`PUNY`** specify the number of addressable points per inch in the *x* and *y* directions. These values are used by the program to infer the spacing between points, so that absolute dimensions (i.e., dimensions in centimeters) may be specified in the plotting statements. You may wish to change these values if drawn dimensions are different from the dimensions specified in the plotting statements.

FILE **`FILE`** specifies the default output file for graphics output. If no **`FILE`** parameter is given, output is sent to the standard output (usually the user's terminal). If the file name is preceded by a "+", output is appended to an existing file (rather than overwriting the file). If the file name is preceded by "|", output is directed to the named program; (this only works on UNIX or UNIX-like systems).

LIKE The **`LIKE`** parameter is used to define one device to be similar to another. The current device uses the same specifications as the device named on the **`LIKE`** parameter, except for the items listed in the current entry. There are many examples of the **`LIKE`** parameter in the *s4pcap* file.

PEN and AREA

`PEN` and **`AREA`** are examples of entries to control the device. **`PEN`** specifies the sequence of characters used to change the hardware line type (pen or color), and **`AREA`** specifies the sequence of characters to use to produce area fill (e.g., for the **`COLOR`** statement). *Note*: These parameters can be used to turn off hardware line types and area fills (software area fills are used instead). To do this, specify the **`PEN`** or **`AREA`** parameter preceded by the "!" character without a value (or "="). An example of this may be found in the *tekbw* entry.

BFSZ

`BFSZ` sets the size of the internal buffer used to generate the plot output. It may be used to limit the maximum record length written to the plot output file.

Format of Mask Data Files

Introduction

TSUPREM-4 can read mask layout data from specially formatted files. Taurus Layout—IC Layout Interface can be used to create these mask data files interactively, extracting the appropriate information from a GDS description of a layout. This appendix describes the format of the mask data files used by TSUPREM-4. It is intended for use by experienced programmers who wish to generate mask data files for use by TSUPREM-4. It can also be used for generating simple mask files by hand.

Format

The figure below shows an example of a mask data file. The first line identifies the format of the file. It contains the characters “TL1” followed by a space and a 4-digit number. The number represents the version of Taurus-Layout that created the file. Current versions of Taurus-Layout specify values from 0000 to 0100.

```
TL1 0100
/ Mask definition file s4ex4m.tl1, for use with s4ex4[abc].
/
%%
%%
1e3
   0 5000
4
Field 1
   0 3900
Poly 1
   0 0650
Contact 2
   0 1950
3250 5000
Metal 1
1300 5000
```

Figure C-1 Example of a mask data file

The identification line is followed by any number of comment lines, identified by a "/" character in the first column. The comments are printed when the file is read with the **MASK** statement in TSUPREM-4. The comments are followed by any number of lines beginning with a "%" character. These lines are for use by Taurus-Layout and are ignored by TSUPREM-4. The first line following the "%" lines contains a single floating-point number that represents the scale factor (units per micron) used for coordinates in the mask file. In the Figure C-1 example, there are 1000 units per micron, so the quantity 3250 is used in the mask file to represent a value of 3.25 microns. Following the scale factor is a line containing the minimum and maximum coordinates in the mask specification. In this example, the mask specification extends from 0 microns to 5.0 microns.

The next line gives the number of masks in the file. It is followed by the data for each mask. The first line of data for each mask gives the name of the mask and the number of opaque segments. Following lines give the minimum and maximum coordinates of each opaque segment, in scaled units. TSUPREM-4 does not distinguish between upper and lower case in mask names.

Data Format of Saved Structure Files

Introduction

TSUPREM-4 can save simulated structures using the TSUPREM-4 format and the medici format. It reads saved structures in TSUPREM-4 format. Although these files contain formatted text, they are not intended to be understood or modified by users. The documentation that follows is provided for experienced programers who wish to interface TSUPREM-4 with other programs.

TSUPREM-4 can also read and write Technology Interchange Format (TIF) files. For more information on the format of TIF files, please contact Synopsys TCAD.

TSUPREM-4 uses other file formats for communicating with the Taurus-Lithography and minimos 5 programs. These formats contain subsets of the solution information provided by the TSUPREM-4 and medici formats, and are not described here.

TSUPREM-4 Structure File Format

The TSUPREM-4 structure file format has evolved over time. The following section describes the format used by the latest version of the program, while the section Older Versions of TSUPREM-4, p. D-7 describes differences between the current format and that used by older versions of TSUPREM-4.

Current Version

This section describes the data format of the TSUPREM-4 structure file. Indented lines denote lines that appear in the saved structure file. They are not indented in the file—the first code letter must be the first character of the line. Names in angle brackets (<>) denote quantities (usually numbers) that are described in the text. Floating-point values may use exponential notation, with the exponent denoted by "e" or "E" (e.g., 1.23e21).

Program Identification

v TMA TSUPREM-4 (1999.2S)

The version number (1999.2) and platform code (S) depend on the particular release of the program that produced the file. This line should appear exactly as shown, except for the version number and platform code.

Coordinates

`c` *<index>* *<x-coordinate>* *<y-coordinate>* **0**

There is one "c" line for each mesh point in the grid. <*index*> is the integer index of the mesh point, starting at 1 and increasing to the total number of points in the grid. <*x-coordinate*> and <*y-coordinate*> are the (floating-point) coordinates of the mesh point, in microns.

Edges

`e` *<index>* *<point-1>* *<point-2>* *<bcode>*

There is one "e" line for each edge segment in the structure, where an edge is defined as a mesh line that lies on the interface between two regions or on the boundary of the structure. <*index*> is the (integer) index of the edge segment. <*index*> starts at one and increases to the total number of edges in the mesh. <*point-1*> and <*point-2*> are the (integer) indices of the coordinates that define the starting and ending points of the edge. <*bcode*> has the (integer) value 2 for edges on exposed boundaries, and zero otherwise. The "e" lines are produced by TSUPREM-4 when writing a structure file, but are ignored when the structure file is read; they need not be present in the file.

Regions

`r` *<index>* *<material>*
`b` *<edge>*
. .
. .
. .

Each region is defined by an “r” line and a series of “b” lines. The regions are specified in the order of their region <*index*>, starting at one. The <*material*> of each region is specified by name. The name *e_photoresis* is used for exposed photoresist.

The “b” lines specify the list of edges, in order, that define the region. Each (integer) value <*edge*> refers to the <*index*> value on an “e” line. The “b” lines define the boundary of the region in a clockwise direction when using a left-handed coordinate system (*x* increasing towards the right, and *y* increasing from top to bottom). The “b” lines are produced by TSUPREM-4 when writing a structure file but are ignored when the structure file is read; they need not be present in the file.

Triangles

`t` <*index*> <*region*> <*c1*> <*c2*> <*c3*> <*t1*> <*t2*> <*t3*>

The triangles are specified in order of their (integer) <*index*>, starting at one. The (integer) <*region*> specifies the region to which the triangle belongs. <*c1*>, <*c2*>, and <*c3*> are the indices of the “c” lines that define the corners of the triangle. They are ordered to produce a clockwise triangle in a left-handed coordinate system. <*t1*>, <*t2*>, and <*t3*> are the (integer) indices of the neighboring triangles. <*t1*> is the neighbor on the side opposite corner <*c1*>; <*t2*> is the neighbor opposite <*c2*>; and <*t3*> is the neighbor opposite <*c3*>. A code of -1024 is used for edges on a reflecting boundary, and -1022 is used for edges on an exposed boundary.

Model Parameters

`M` <*subornt*> <*nadd*> <*temperature*> <*oxmodel*> <*pdmodel*>
<*subrotc*> <*phototype*> <*activmod*> <*orVoid*> <*customMod*>
<*clModel*> <*kbModel*> <*shModel*> <*itModel*>

All parameters appear on a single line in the structure file.

The “M” line specifies miscellaneous information about the structure and models used. The (integer) value <*subornt*> specifies the substrate orientation:

Table D-1

<*subornt*>	**Substrate Orientation**
0	<100>
1	<110>
2	<111>

The integer <*nadd*> specifies how many additional parameters appear on the “M” line; in version 6.6 <*nadd*> is 12, and all of the parameters listed above are present. <*temperature*> is the last processing temperature, in degrees Kelvin.

<*oxmodel*> indicates which oxidation model was selected by the user. The values of <*oxmodel*> are as follows:

Table D-2

Oxmodel	Oxidation Model
0	**ERFC**
1	**VERTICAL**
2	**COMPRESS**
3	**VISCOUS**
4	**VISCOELA**
5	**ERF1**
6	**ERF2**
7	**ERFG**

<*pdmodel*> indicates what point defects models are to be used. Its value is calculated by adding 8 to the value of **NSTREAMS** (1 or 3), then adding an additional value for each of the following point defect parameters that is set true:

Table D-3

Parameter	Value Added to <*pdmodel*> if True
PAIR.GRA	16
PAIR.SAT	32
PAIR.REC	64
PD.PFLUX	256
PD.PTIME	512
PD.PREC	1024

<*subrotc*> gives the orientation of the *x* axis of the simulation—it is the cosine of the angle between the *x* axis and a <110> direction. <*phototype*> specifies the polarity of any photoresist in the structure—it is 1 for positive photoresist and 0 for negative photoresist. <*activmod*> indicates whether the transient activation model is in effect; it has the value 1 when the **ACT.TRAN** model is enabled and 0 when the **ACT.EQUI** model is in effect. <*orVoid*> is the current void size at silicon/polysilicon interfaces. <*customMod*> indicates any custom or experimental models that are in effect; it is reserved for future use. <*clModel*> is 1 if the interstitial clustering model is enabled and 0 otherwise. <*kbModel*> indicates which point defect bulk recombination is selected:

Table D-4

<*kbModel*>	Recombination Model
-1	none
0	**KB.LOW**
1	**KB.MED**
2	**KB.HIGH**

where "none" indicates that the user has explicitly specified values of **KIV.0** and/or **KIV.E**. *Note* that <*kbModel*> is ignored when reading the file unless compatibility with version 1998.4 or earlier has been specified with **V.COMPAT**. <*shModel*> is 1 if the stress history model is enabled and 0 otherwise; <*itModel*> is 1 if the interface trap model is enabled and 0 otherwise.

Solution Values

s <*nsol*> <*sol1*> <*sol2*> . . . <*soln*>

The "s" line specifies which solution values are associated with each node of the structure. The (integer) value <*nsol*> specifies how many solution values are given, while <*sol1*> through <*soln*> give the names of the solutions that are present. Valid names in version 6.4 are: the names of the impurities (including user-defined impurities); the names of the impurities preceded by "*a_*", indicating the active portion of the impurity concentration; the names of the impurities preceded by "*gb_*", indicating the concentration of the impurity associated with polycrystalline grain boundaries; the names *vacancy, interstitial, x.vel, y.vel, o2, h2o, trap, sxx, syy, sxy, rloop, dloop, lgrain, gorient,* and *tpoly*. *Note* that this list of solutions is subject to change in future revisions of the program.

Nodes

n <*cindex-1*> <*region*> <*val1*> . . . <*valn*>

The "n" lines specify the solution values at each node. For each point in the mesh, there is an "n" line for each material at the point. The (integer) value <*cindex-1*> gives the index of the coordinate, minus one, and the (integer) <*region*> specifies the region. Points on exposed surfaces must also have an *ambient* node, denoted by a region index of zero. For each node, the solution values are given by <*val1*> through <*valn*> (floating-point).

Example

The following input produces the structure file shown in Figure D-1.

```
line x loc=0
line x loc=1
line y loc=0
line y loc=1
init boron=1e13
deposit oxide thick=.2
savefile out.file=test.str
stop
```

This structure contains 6 grid points, 2 regions (silicon and oxide), 4 triangles, and 10 nodes (4 in silicon, 4 in oxide, and 2 ambient nodes). Solution values for boron concentration are given at each node. No last processing temperature was specified, and the **VERTICAL** oxidation model and the **PD.FERMI** point defect model were used.

```
v TMA TSUPREM-4 (1999.2S)
c 1    0 0   0
c 2    0 0.99999997   0
c 3    0.99999997 0   0
c 4    0.99999997 0.99999997   0
c 5    0 -0.19999999   0
c 6    0.99999997 -0.19999999   0
e 1    2 1  0
e 2    4 2  0
e 3    3 4  0
e 4    5 6  2
e 5    1 5  0
e 6    6 3  0
e 7    3 1  0
r 1   silicon
b 1
b 7
b 3
b 2
r 2   oxide
b 4
b 6
b 7
b 5
t 1    1   1 3 2   2 -1024 4
t 2    1   3 4 2   -1024 1 -1024
t 3    2   1 5 6   -1022 4 -1024
t 4    2   1 6 3   -1024 1 3
M 0 12 1073.16 1 9 1.000000 0 0 0 0 1 0 0 1
s 2    boron a_boron
n 0    1   1.000000e+13 1.0000e+13
n 0    2   1.000000e+05 1.0000e+05
n 1    1   1.000000e+13 1.0000e+13
n 2    1   1.000000e+13 1.0000e+13
n 2    2   1.000000e+05 1.0000e+05
n 3    1   1.000000e+13 1.0000e+13
n 4    2   1.000000e+05 1.0000e+05
n 4    0   1.000000e+05 1.0000e+05
n 5    2   1.000000e+05 1.0000e+05
n 5    0   1.000000e+05 1.0000e+05
```

Figure D-1 TSUPREM-4 structure file

Older Versions of TSUPREM-4

This section describes differences between the current TSUPREM-4 structure file format and that used in older versions of the program.

Regions

In versions 6.0 and earlier, regions were identified by a <*material*> code:

Table D-5

<*material*>	**Material**
`1`	`oxide`
`2`	`nitride`
`3`	`silicon`
`4`	`polysilicon`
`5`	`oxynitride`
`6`	`aluminum`
`7`	`photoresist`
`8`	`ambient (included void)`

Model Parameters

In versions 8926 and before <*nadd*> and the parameters that follow are omitted. In versions 9035 and 5.0 <*nadd*> is 3, and the <*temperature*>*,* <*oxmodel*>*, and* <*pdmodel*> parameters are present. Starting with version 5.2, <*nadd*> is 5, and the “M” line includes the <*subrotc*> and <*phototype*> parameters. Starting with version 6.2, <*nadd*> is 6 and the “M” line includes the <*activmod*> parameter. Starting with version 6.3, <*nadd*> is 7 and the “M” line includes the <*orVoid*> parameter. Starting with version 6.4, <*nadd*> is 8 and the “M” line includes the <*customMod*> parameter. Starting with version 6.5, <*nadd*> is 10 and the “M” line includes the <*clModel*> and <*kbModel*> parameters. Starting with version 6.6, <*nadd*> is 12 and the “M” line includes the <*shModel*> and <*itModel*> parameters. Starting with version 1999.2, -1 is a valid value for <*kbModel*>.

In versions of TSUPREM-4 prior to version 6.0 the value of <*pdmodel*> is interpreted as follows:

Table D-6

<*pdmodel*>	**Point Defect Model**
`0`	**`TWO.DIM (PD.TRANS)`**
`2`	**`FERMI (PD.FERMI)`**
`4`	**`STEADY (Obsolete;PD.TRANS used instead)`**

Solution Values

In versions 6.0 and earlier, <*sol1*> through <*soln*> are integer values indicating which solution values are present:

Table D-7

<*solx*>	**Solution**
0	`vacancy concentration`
1	`interstitial concentration`
2	`arsenic concentration`
3	`phosphorus concentration`
4	`antimony concentration`
5	`boron concentration`
6	`steady-state vacancy concentration`
7	`steady-state interstitial concentration`
8	`x component of flow velocity`
9	`y component of flow velocity`
10	`oxidant concentration (dry O2)`
11	`oxidant concentration (wet O2)`
12	`concentration of filled interstitial traps`
15	`Sxx component of stress`
16	`Syy component of stress`
17	`Sxy component of stress`
20	`active component of arsenic concentration`
21	`active component of phosphorus concentration`
22	`active component of antimony concentration`
23	`active component of boron concentration`

Nodes

In version 6.0 and earlier, <*region*> specifies a material index rather than a region index. The same values are used to identify materials as are used in the “r” lines.

Medici Structure File Format

This section describes the output file format produced by the "**`SAVEFILE MEDICI`**" statement in TSUPREM-4. The TSUPREM-4 Medici file contains numerical values only. Floating point values use an exponent, denoted by "e". Integer values do not have an exponent or a decimal point. Values are separated by spaces. The format of the file is as follows.

Line 1:

<npt> *<ntri>* *<nelpt>*

This line specifies the number of points in the grid <*npt*>, the number of triangular elements <*ntri*>, and the number of electrode points <*nelpt*>.

Line 2:

<nelec> -*<nmat>*

This line specifies the number of electrodes <*nelec*> and the number of materials in the structure <*nmat*>. *Note* that the number of materials is preceded by a minus sign. TSUPREM-4 converts all points on an interface with aluminum to electrodes.

Lines 3 through <*npt*>+2:

<xcoord> *<ycoord>* *<cnet>* *<ctotal>*

These lines specify the x and y coordinates <*xcoord*> and <*ycoord*> of each mesh point (in centimeters), the net n-type doping <*cnet*> (in atoms/cm^3), and the total doping <*ctotal*> (in atoms/cm^3). *Note* that a negative value of <*cnet*> indicates a p-type region. <*xcoord*>, <*ycoord*>, <*cnet*>, and <*ctotal*> are all floating-point values.

Lines <*npt*>+3 through <*npt*>+<*ntri*>+2:

<reg> *<p1>* *<p2>* *<p3>*

These lines specify the triangular elements that make up the mesh. <*reg*> is the region number for the element. (Regions are numbered consecutively, starting at one.) <*p1*>, <*p2*>, and <*p3*> are the indices of the points at the vertices of the triangular element. The points are specified in a clockwise order, when viewed in a left-handed coordinate system (i.e., x increasing from left to right and y increasing from top to bottom).

Lines <*npt*>+<*ntri*>+3 through <*npt*>+<*ntri*>+<*nelpt*>+2:

<ept> *<elec>*

These lines specify the electrode <*elec*> associated with each electrode point <*ept*>. <*elec*> is either a name (assigned with the **ELECTRODE** statement) or a number. (Electrode numbers start at one and increasing to <*nelec*>, as needed.)

Line <*npt*>+<*ntri*>+<*nelpt*>+3 (last line in file):

<*mat1*> <*mat2*> . . . <*matn*>

This line specifies the material associated with each region in the structure:

Table D-8

<*matx*>	**Material**
`-6`	`oxynitride`
`-4`	`unspecified insulator (e.g., photoresist)`
`-2`	`nitride`
`-1`	`oxide`
`1`	`silicon`
`3`	`polysilicon`
`4`	`unspecified semiconductor`

Example

The following TSUPREM-4 input produces the structure file shown in Figure D-2:

```
line x loc=0
line x loc=1
line y loc=0
line y loc=1
init boron=1e13
deposit oxide thick=.2
deposit aluminum thick=.1
select z=0 temp=800
savefile medici out.file=testpi.str
stop
```

```
6 4 2
1 -2
    0.000000e+00     0.000000e+00    -1.000000e+13     1.000000e+13
    0.000000e+00     1.000000e-04    -1.000000e+13     1.000000e+13
    1.000000e-04     0.000000e+00    -1.000000e+13     1.000000e+13
    1.000000e-04     1.000000e-04    -1.000000e+13     1.000000e+13
    0.000000e+00    -2.000000e-05     0.000000e+00     0.000000e+00
    1.000000e-04    -2.000000e-05     0.000000e+00     0.000000e+00
1 1 3 2
1 3 4 2
2 1 5 6
2 1 6 3
5 1
6 1
1 -1
```

Figure D-2 Medici structure file

Using the MINIMOS 5 Interface

This appendix explains how to send TSUPREM-4 results to the MINIMOS 5 device simulator. It should be used in conjunction with the `SAVEFILE` input statement described Chapter 4, SAVEFILE, p. 4-74.

Overview of the TSUPREM-4 Interface to MINIMOS 5

The interface is file-oriented and unidirectional (from TSUPREM-4 to MINIMOS 5), and consists of three separate steps.

1. TSUPREM-4 writes a formatted file (i.e., ASCII or EBCDIC). This file contains a 2-D doping profile of the source area (and optionally the drain area) of the device being simulated.
(See Step 1: Directing TSUPREM-4 to Generate a Formatted File, p. E-2.)
2. Use a separate stand-alone FORTRAN program (named *mmatob*) to convert the formatted file to a FORTRAN binary file. (See Step 2: Converting the Formatted File to FORTRAN Binary, p. E-7)
3. Execute MINIMOS 5, which reads in the binary doping file. (See Step 3: Running MINIMOS 5, p. E-7)

Throughout this discussion, use the convention that the source is on the left of the channel and the drain is on the right of the channel. Also, remember that TSUPREM-4 expects user-specified values in microns, while MINIMOS 5 expects centimeters.

All references to MINIMOS 5 refer to the program supplied by the Technical University of Vienna, Austria, Institute for Microelectronics, and the accompanying MINIMOS 5 User's Guide, dated March 2, 1990.

Step 1: Directing TSUPREM-4 to Generate a Formatted File

Central to the preparation of the MINIMOS 5 output file is the concept of the MINIMOS 5 simulation region. This region is a subset of a simulated TSUPREM-4 device. While a TSUPREM-4 MOS device simulation may encompass an entire device, including the gate oxide, field oxide, and metallization layers, MINIMOS 5 expects to read a 2-D doping file encompassing only a portion of the full device. The **`X.MIN`**, **`X.MAX`**, **`Y.MIN`**, and **`Y.MAX`** parameters on the **`SAVEFILE`** statement are used to define a rectangular subset of the complete device. This subset is referred to as the MINIMOS 5 simulation region.

Defining the MINIMOS 5 Simulation Region

The MINIMOS 5 simulation region must be defined to include the source, drain, and channel of the MOS transistor; gate and field oxide regions are normally excluded. The MINIMOS 5 simulation region must be deep enough to include the source and drain junction depths, because MINIMOS 5 expects to see a change in the sign of the net doping concentration along both the left and right edges of its simulation space (except when simulating SOI devices).

The following TSUPREM-4 statement defines the MINIMOS 5 simulation region, and generates a formatted MINIMOS 5 file for the device shown in Figure E-1 (from Chapter 6, NMOS LDD Process, p. 6-2):

```
SAVEFILE  MINIMOS5 OUT.FILE=MMDOPF FULL.DEV+
   X.MIN=-2 X.MAX=2  Y.MIN=0.0174 Y.MAX=1.5+
   X.MASK.S=-0.57 X.MASK.D=0.57
```

The parameter **`MINIMOS5`** specifies that the file to be produced is a formatted 2-D doping file, to be read by MINIMOS 5. **`OUT.FILE=MMDOPF`** specifies that the formatted doping file is named *MMDOPF*. Because this is the filename expected by the file conversion program *mmatob* (discussed later in this section), it is the recommended name to use when directing TSUPREM-4 to prepare a MINIMOS 5 doping file.

In Figure E-1, you see a complete device with source, channel, and drain regions, so the **`FULL.DEV`** parameter should be specified. If the TSUPREM-4 simulation encompasses only the source half of a device, use **`HALF.DEV`**. If the TSUPREM-4 simulation is for the drain half of a device, reflect the TSUPREM-4 structure to create a complete device and specify **`FULL.DEV`**. Because the drain region is assumed to be to the right of the channel, use the statement

STRUCTURE REFLECT LEFT

(preceding the **`STRUCTURE MINIMOS5`** statement) to reflect the device at its left edge (the center of the channel), thus creating a complete device.

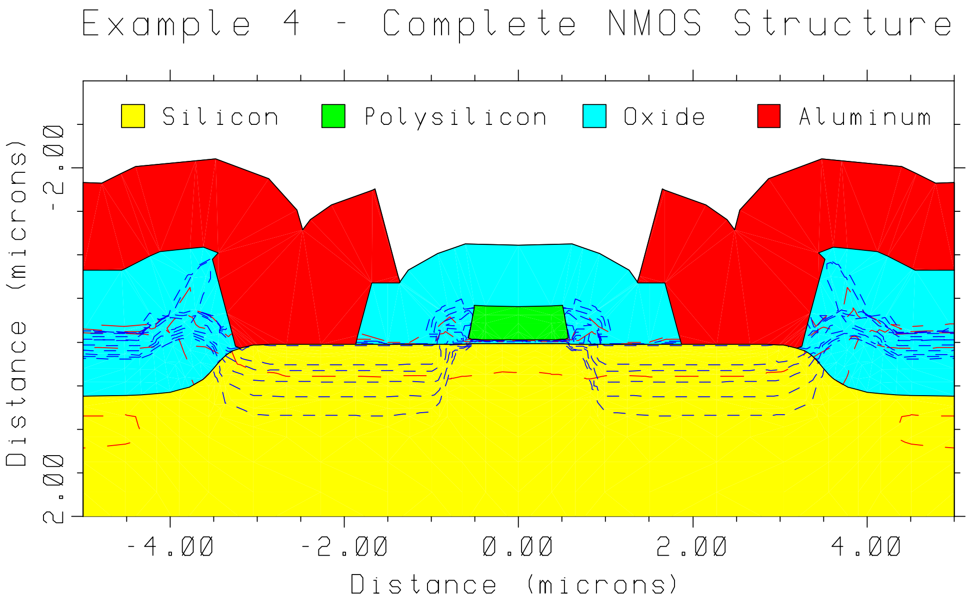


Figure E-1 NMOS structure to be transferred to MINIMOS 5

The **`X.MIN`**, **`X.MAX`**, **`Y.MIN`**, and **`Y.MAX`** parameters define the left, right, top, and bottom edges of the MINIMOS 5 simulation region, respectively. The assigned values for **`X.MIN`** and **`X.MAX`** are chosen by visual inspection of Figure E-1 and need not be exact. **`X.MIN`** is roughly centered between the left-most field oxide and the gate, in an area where the impurity contours are essentially flat. If **`X.MIN`** is too far to the left, the field oxide region is included. **`X.MAX`** is assigned the corresponding value on the right side of the structure.

Similar to **`X.MIN`** and **`X.MAX`**, **`Y.MAX`** is determined based on visual inspection of Figure E-1 and need not be exact. However, **`Y.MAX`** must be deep enough in the structure to include the source and drain junction depths, or MINIMOS 5 issues the error message

```
DOPING PROFILE INCONSISTENT
```

`Y.MAX` should also be deep enough in the structure to include the depletion region for the maximum bias to be applied, particularly if the dopant profile is still changing at that depth.

The values used for **X.MIN**, **X.MAX**, and **Y.MAX** are not critical, because if MINIMOS 5 needs a simulation region that is larger than the one defined by the doping file, MINIMOS 5 extends the profiles in the required direction. However, the value assigned to **Y.MIN** *is* critical, because MINIMOS 5 interprets **Y.MIN** as the gate oxide/silicon interface. Thus, **Y.MIN** has a physical significance and should be carefully determined. For TSUPREM-4 MOSFET simulations, the gate oxide/silicon interface is typically found just below *y*=0.0. Therefore, as a first approximation, the default value (**Y.MIN**=0.0) could be used. Another approximation could be obtained from visual inspection of a 2-D plot. However, the most accurate value is obtained by using the following pair of TSUPREM-4 statements:

```
SELECT     Z=1
PRINT.1D   LAYERS   X.VALUE=<xchan>
```

where <*xchan*> is the *x* coordinate of the vertical slice along which values are to be printed. In a typical TSUPREM-4 MOSFET simulation (such as Figure E-1), the gate oxide/silicon interface is bowed slightly upwards and is closest to *y*=0.0 in the middle of the channel. For such cases, <*xchan*> in the preceding code fragment should be the *x* coordinate of the middle of the channel. The center of the channel for the device shown in Figure E-1 is at *x*=0.0. Thus, the TSUPREM-4 statements

```
SELECT    Z=1
PRINT.1D  LAYERS  X.VALUE=0.0
```

are used to generate the following TSUPREM-4 output (results may vary slightly on your system):

```
Num      Material       Top     Bottom  Thickness        Integral
   1         oxide   -1.1111    -0.4107     0.7004      7.0040e-05
   2   polysilicon   -0.4107    -0.0245     0.3862      3.8615e-05
   3         oxide   -0.0245     0.0174     0.0419      4.1938e-06
   4       silicon    0.0174   200.0000   199.9826      1.9998e-02
```

From this output, note that the bottom of the oxide layer occurs at *y*=0.0174 microns, so **Y.MIN**=0.0174 is used. Also, note that the thickness of the insulating oxide is 0.0419 microns (419 Å). This value is used on the MINIMOS 5 **DEVICE** directive when specifying a value for the **TINS** parameter.

There is no need to specify **X.CHANNE** in this example, because the default value (midway between **X.MIN** and **X.MAX**) is appropriate. **X.CHANNE** is only needed if an asymmetric device is simulated, and serves to divide the MINIMOS 5 simulation region into a source half and a drain half.

The two remaining parameters, **X.MASK.S** and **X.MASK.D**, are similar to **Y.MIN** in that they have a physical significance. **X.MASK.S** is the *x* coordinate of the mask edge in the source area of the MINIMOS 5 simulation region. MINIMOS 5 interprets this coordinate as the left edge of the gate electrode and treats **X.MASK.D** as the right edge of the gate electrode. Values for these parameters can be obtained by visual inspection of Figure E-1. For more accuracy, use the following pair of TSUPREM-4 statements

```
SELECT    Z=Y
PRINT.1D  POLY  /OXIDE
```

to print the *x* and *y* coordinates of the interface between the polysilicon gate contact and the gate oxide. From this output (not shown), you will find that **X.MASK.S**=-0.57 and **X.MASK.D**=0.57.

The **SGAP** and **DGAP** parameters on the MINIMOS 5 **DEVICE** directive are used to specify the gap between the gate and source contact and the gap between the gate and drain contacts, respectively. The TSUPREM-4 statements

```
SELECT    Z=Y
PRINT.1D  SILICON  /OXIDE
```

can be used to print the *x* and *y* coordinates of the silicon/oxide interface, as an aid to determining **SGAP** and **DGAP** for the subsequent MINIMOS 5 run. Alternatively, the values could be approximated by visual inspection of Figure E-1.

Notes on the Size of the MINIMOS 5 Simulation Region

The **X.MIN**, **X.MAX**, **Y.MIN**, and **Y.MAX** parameters determine the size of the MINIMOS 5 simulation region. If this region is large, containing many TSUPREM-4 grid points, the preparation of the MINIMOS 5 doping file could take several minutes. Additionally, the subsequent MINIMOS 5 run can take more execution time if the doping file defines a large simulation region. For these reasons, you should attempt to keep the MINIMOS 5 simulation region reasonably small when choosing **X.MIN**, **X.MAX**, **Y.MIN**, and **Y.MAX**. This also reduces the size of the doping file.

The standard version of MINIMOS 5 expects the doping file to contain from 10 to 80 horizontal grid lines and from 10 to 80 vertical grid lines. Normally, this requirement is satisfied automatically by TSUPREM-4 during the preparation of the doping file. However, certain choices for **`X.MIN`**, **`X.MAX`**, **`Y.MIN`**, and **`Y.MAX`** can result in a TSUPREM-4 warning message stating that there are too many (or too few) grid lines. Unless the version of MINIMOS 5 you run has been modified to accept expanded limits, you must modify the **`X.MIN`**, **`X.MAX`**, **`Y.MIN`**, or **`Y.MAX`** values as advised in the TSUPREM-4 warning message and regenerate the doping file. Otherwise, when MINIMOS 5 attempts to read the doping file, it issues the error message

```
DOPING FILE FORMAT ERROR
```

and terminates execution. You can also modify the number of grid lines by specifying the **`DX.MIN`** and/or **`DY.MIN`** parameters. These change the minimum grid spacings in the *x* and *y* directions from their default values of min((**`X.MAX`** - **`X.MIN`**)/80,0.01) and min((**`Y.MAX`** - **`Y.MIN`**)/80,0.01), respectively.

Nonplanar Oxide Regions in MINIMOS 5

As stated in the *MINIMOS 5 User's Guide,* MINIMOS 5 can model nonplanar oxide regions. This is accomplished by including the **`GEOMETRY`** directive (and associated parameters) in the MINIMOS 5 input command file. However, the doping file (as dictated by MINIMOS 5) makes no provision for the inclusion of geometry information. As discussed, the doping file uses a single coordinate (**`Y.MIN`**) to specify the gate oxide/silicon interface, implying a planar device. Therefore, TSUPREM-4 cannot automatically export nonplanar geometry information to MINIMOS 5 by including it in the doping file—the MINIMOS 5 user must approximate the nonplanarities of the device simulated in TSUPREM-4 by hand, attempting to duplicate the TSUPREM-4 geometry in MINIMOS 5 by using the **`GEOMETRY`** directive. Fortunately, many devices (such as the MOSFET in Figure E-1) are approximately planar, and the MINIMOS 5 **`GEOMETRY`** directive is not required to simulate the intrinsic device operation. In such cases, assign **`Y.MIN`** to the top-most point of the silicon, as discussed earlier. Because the gate oxide/silicon interface is not absolutely flat, the left and right portions of the top of the MINIMOS 5 simulation region contain very thin layers of oxide. To correct this, TSUPREM-4 automatically ignores the impurity concentration in the oxide and extends the doping profile of the underlying silicon upward (using the concentration at the oxide/silicon interface). In general, TSUPREM-4 ignores the dopants in any nonsilicon material within the MINIMOS 5 simulation region, using instead the doping of the underlying silicon.

Step 2: Converting the Formatted File to FORTRAN Binary

TSUPREM-4 writes a formatted (i.e., ASCII or EBCDIC) file for MINIMOS 5. However, MINIMOS 5 requires a FORTRAN binary doping file. The formatted file written by TSUPREM-4 is converted to FORTRAN binary by executing the stand-alone FORTRAN program named *mmatob*. This step is needed because of incompatibilities between binary files written by C programs (such as TSUPREM-4) and FORTRAN programs (such as MINIMOS 5) on some computer systems.

The current version of *mmatob* is executed by simply typing

```
mmatob
```

followed by a carriage return. *Note* that *mmatob* expects to read a formatted file named *MMDOPF* and write a binary file named *MMDOPB*. The easiest way to assure that the formatted file is named *MMDOPF* is to specify **`OUT.FILE=MMDOPF`** on the TSUPREM-4 **`SAVEFILE`** statement used to generate the MINIMOS 5 file. Alternatively, rename or copy the formatted file to *MMDOPF* before executing *mmatob*. Any existing file named *MMDOPB* is overwritten when *mmatob* is executed.

Step 3: Running MINIMOS 5

MINIMOS 5 reads the binary doping file from FORTRAN logical unit 20. Similarly, the input command file must be available on FORTRAN unit 15. Refer to your MINIMOS 5 documentation for instructions on how to associate these input files with the correct FORTRAN logical unit numbers.

Figure E-2 shows a listing of a sample MINIMOS 5 input command file (*EX2D.INP,* included with standard versions of MINIMOS 5).

```
EXAMPLE FOR CLASSICAL 2-D MODEL (EX2D.INP)
DEVICE     CHANNEL=N GATE=NPOLY TINS=150.E-8 W=1.E-4 L=0.85E-4
BIAS       UD=4. UG=1.5
PROFILE    NB=5.2E16 ELEM=AS DOSE=2.E15 TOX=500.E-8 AKEV=160.
+          TEMP=1050. TIME=2700
IMPLANT    ELEM=B DOSE=1.E12 AKEV=12 TEMP=940 TIME=1000
OPTION     MODEL=2-D
OUTPUT     ALL=YES
END
```

Figure E-2 Listing of MINIMOS 5 command file *EX2D.INP*

This file uses the **PROFILE** and **IMPLANT** directives to generate a doping profile. To convert *EX2D.INP* so that it reads an external 2-D doping file, the **PROFILE** statement is changed to

PROFILE FILE=2-D

The **IMPLANT** directive, used in *EX2D.INP* to specify a channel implantation, is removed. In addition, the parameters and values specified on the **DEVICE** statement are modified as needed to match the geometry of the device simulated in TSUPREM-4. The resulting modified MINIMOS 5 input command file is shown in Figure E-3.

```
EXAMPLE FOR CLASSICAL 2-D MODEL (READS 2-D DOPING FILE)
DEVICE     CHANNEL=N GATE=NPOLY TINS=419E-8 W=1.E-4 L=1.14E-4
+          DGAP=1.87E-4 SGAP=1.87E-4
BIAS       UD=4. UG=1.5
PROFILE    FILE=2-D
OPTION     MODEL=2-D
OUTPUT     ALL=YES
END
```

Figure E-3 Listing of MINIMOS 5 command file *EX2D.INP*, modified to read doping profiles produced by TSUPREM-4

Recall that MINIMOS 5 expects dimensions in centimeters, not microns. To use MINIMOS 5 to simulate a nonplanar device, specify the **GEOMETRY** directive, as discussed earlier, in Nonplanar Oxide Regions in MINIMOS 5, p. E-6.

Notes on Using MINIMOS 5

Particular care must be taken when specifying the **L** parameter (gate electrode length) on the MINIMOS 5 **DEVICE** directive. The value assigned to **L** should correspond to the distance between the values assigned to TSUPREM-4 parameters **X.MASK.S** and **X.MASK.D**. For example, if **X.MASK.S**=-0.57 (microns) and **X.MASK.D**=0.57 (microns), the distance between them is 1.14 microns, so **L**=1.14e-4 (centimeters). In this case, specifying a value of **L** greater than
1.14e-4 centimeters has the effect of stretching the channel, and should not cause problems in MINIMOS 5. However, a value of **L** substantially smaller than 1.14e-4 centimeters could cause the source and drain profiles to overlap, effectively eliminating the channel. MINIMOS 5 responds with the error message

```
DOPING PROFILE INCONSISTENT
```

and terminates execution.

Interpreting Error Messages

The warning and error messages issued by TSUPREM-4 during the preparation of the doping file are clear and fairly detailed, recommending corrective action when necessary.

The following error messages from MINIMOS 5 may be encountered:

```
DOPING PROFILE INCONSISTENT
```

MINIMOS 5 issues this message for a variety of error conditions. This can occur when the **L** parameter on the MINIMOS 5 **DEVICE** directive is too small, causing the source and drain profiles to overlap, effectively eliminating the channel. Correct the error by using a value of **L** that is at least as large as the difference between the **X.MASK.S** and **X.MASK.D** parameters, as described above. The doping file does not need to be regenerated.

The above message can also occur when the MINIMOS 5 simulation region, as defined in TSUPREM-4, is not deep enough to include the junction depths in both the source and drain regions. Correct the error by using a larger value for **Y.MAX** on the TSUPREM-4 **SAVEFILE** statement. In this case, the doping file must be regenerated.

To avoid the above message when using TSUPREM-4 and MINIMOS 5 to simulate SOI devices, the MINIMOS 5 **DEVICE** directive must include the **BULK**, **FILM**, and **KBULK** parameters.

When too many (or too few) horizontal or vertical grid lines are present in the doping file, the following message can appear:

```
DOPING FILE FORMAT ERROR
```

TSUPREM-4 in this case should have issued a warning message, describing the problem and recommending corrective action. However, TSUPREM-4 went ahead and prepared a doping file after issuing the warning, in the event that your version of MINIMOS 5 has been modified to accept expanded limits. Correct the error by using different values for **X.MIN**, **X.MAX**, **Y.MIN**, or **Y.MAX** in TSUPREM-4, and regenerate the doping file.

Enhancements

This appendix details new features, enhancements, and changes as described in the following sections:

- Advanced Parameter Set Update
- Automatic Tune for Optimal Numerical Method
- Automatic Save at Abnormal Stop
- ACT.FULL for Dopant–Vacancy Clustering
- Enhancements in DDC.FULL
- Initialization of Dopant-Defect Clusters
- Pressure-Dependent Pair/Substitutional Formation Energy
- Adaptive Gridding at Amorphous and Crystalline Boundary
- Modifiable Hobler’s Damage Moments
- Profiling Implant Damages
- Dopant-Dependent Diffusion and Reaction of Oxidants
- Gradient-Doped Epitaxial Growth
- Strained Epitaxial Growth
- New Initialization Methods for Solutions
- User-Defined Equilibrium Active Function
- INTERMED Snapshot
- INTERMED Scope Definition
- Fields to Store the Implant Dopant and Point-Defect Concentrations
- Improved STRESS Command
- Taurus Monte Carlo Implant Updates
- TIF Format Change
- Miscellaneous

Advanced Parameter Set Update

Synopsys has performed a new calibration of the diffusion and electrical activation of dopants. The new Advanced Calibration model switches and parameters, which are included in the file s4advanced.2007.03, can be invoked by the command:

```
METHOD ADVANCED
```

In s4advanced.2007.03, many model switches and parameters have been changed with respect to the previous calibration file s4advanced.2006.06. The most important changes are:

- For each dopant, a single first-order reaction equation has been calibrated for the formation and dissolution of dopant clusters.
- Dopant dose loss is described by the the ITRAP model for all dopants.
- A model for carbon–interstitial clusters is included.

To continue using the old method of advanced calibration in TSUPREM-4 2006.06, load the s4advanced.2006.06 file by specifying the parameter **`VERSION`**:

```
METHOD ADVANCED VERSION=2006.06
```

For a detailed description of the different models in the s4advanced.2007.03 file, see the A*dvanced Calibration User Guide*.

In addition to s4advanced.2007.03, the merger file AdvCal_1d_tables_2007.03.smr is supplied. This file contains calibrated initial conditions for diffusion simulation (factor for the +N model, initial dopant activation level in the amorphized and crystalline silicon regions).

When inserted at implant lines, these calibrated initial conditions will improve simulation results, in particular, in the case of low energy implants. For details on the contents of AdvCal_1d_tables_2007.03.smr, see the *Advanced Calibration User Guide*.

Automatic Tune for Optimal Numerical Method

The more complicated the simulation problem, the more difficult it is to set the optimal values of the numerical parameters. Furthermore, it requires use of the proficient knowledge of the numerical algorithm which is beyond present process engineering. The automatic tune feature automatically tunes the parameter values on the fly so user do not have to know the numerical algorithm. More importantly, this new feature improves the simulation speed. The automatic tune feature is turned on by default.

```
METHOD M.AUTO
```

Automatic Save at Abnormal Stop

When a simulation stops abnormally from a syntax error, an internal fatal error, or CTRL-C, the **`AUTOSAVE`** mode stores simulation results before the stop, so that users can continue the simulation after fixing the error without rerunning the entire simulation. The **`AUTOSAVE`** mode is specified in the **`OPTION`** statement and it is turned off by default:

`OPTION AUTOSAVE=0`	Default. No autosave mode.
`OPTION AUTOSAVE=1`	Recommended. Save the latest result among CPU intensive processes such as **`DIFFUSE`**, **`EPITAXY`**, or **`IMPLANT`** before abnormal stop.
`OPTION AUTOSAVE=2`	Save the latest result among **`DIFFUSE`**, **`EPITAXY`**, **`IMPLANT`**, **`STRUCTURE`**, **`ETCH`**, **`DEPOSIT`**, **`DEVELOP`**, **`EXPOSE`**, **`PROFILE`**, or **`STRESS`** before abnormal stop.

The file format to save temporary solutions is the TSUPREM-4 structure file format, and the file is named with:

```
_<prefix of input file name>_
```

For example, if the input file name is *test.inp*, the saved file is:

```
_test_
```

Note:

`AUTOSAVE` does not work for unexpected exceptions, such as segmentation faults or bus errors caused by memory conflicts.

ACT.FULL for Dopant–Vacancy Clustering

To use **`ACT.FULL`** for dopant–vacancy clustering:

$$\frac{\partial C_{cl2}}{\partial t} = \frac{1}{\tau_{dv}}\left(K_{dvF} n_i \left(\frac{C_a}{n_i}\right)^{\mathrm{DVCF.D.N}} \eta^{\mathrm{DVCF.N.N}} \left(\frac{V}{V^*}\right)^{\mathrm{DVCF.V.N}} - C_{cl2}\,\eta^{\mathrm{DVCR.N.N}} \left(\frac{V}{V^*}\right)^{\mathrm{DVCR.V.N}}\right)$$

Equation F-1

$$\tau_{dv} = \mathrm{DVC.T.0} \cdot \exp\left(-\frac{\mathrm{DVC.T.E}}{kT}\right)$$

Equation F-2

$$K_{dvF} = \mathrm{DVC.F.0} \cdot \exp\left(-\frac{\mathrm{DVC.F.E}}{kT}\right)$$

Equation F-3

where C_{cl2} is the concentration of dopant atoms in transient dopant-vacancy clusters and **`DVC.T.0`**, **`DVC.T.E`**, **`DVC.F.0`**, **`DVC.F.E`**, **`DVCF.D.N`**,

DVCF.N.N, **DVCF.V.N**, **DVCR.N.N**, and **DVCR.V.N** are parameters on the **IMPURITY** statement.

This equation models dopant–vacancy clusters with **DVCF.D.N** dopant atoms per cluster, **DVC.V.N** vacancy required to form a cluster, **DVCR.V.N** vacancy atoms required to dissolve a cluster, and **DVCF.V.N**-**DVCR.V.N** vacancies in the cluster. **DVC.F.0** and **DVC.F.E** control the amount of clustering; while **DVC.T.0** and **DVC.T.E** control the rate of clustering and declustering. Because C_{cl2} is subtracted from the total concentration before computing C_a, the dopant in transient clusters is inactive and immobile.

The net capture of vacancies by clusters is given by:

$$R_{dv} = \mathbf{VFRACM} \cdot \frac{\partial C_{cl2}}{\partial t} \qquad \text{Equation F-4}$$

where **VFRACM** is a parameter on the **IMPURITY** statement. Ideally, **VFRACM** should be equal to (**DVCF.V.N**-**DVCR.V.N**)/**DVCF.D.N**, but it is available as a separate parameter for flexibility in modeling.

The dopant–vacancy clustering model is enabled by specifying **ACT.FULL** on the **METHOD** statement.

The following solutions are built in for dopant–vacancy clusters.

c2_boron, c2_phosphor, c2_arsenic, and c2_antimony.

To define the new solution for dopant–vacancy cluster; for example, carbon-vacancy cluster:

```
IMPURITY IMP=c2_carbon NEW TIF.NAME=Cc2
IMPURITY IMP=carbon IMP.CL2=c2_carbon
```

For C_{cl2} initialization, use **C2FRAC.A** and **C2.INI.F** parameters in the **IMPURITY** statement. **C2FRAC.A** sets the C_{cl2} concentration in a regrowth region, while **C2.INI.F** specifies the additional C_{cl2} concentration to be added to the current C_{cl2} concentration in a crystalline region. The C_{cl2} initialization with C_{cl1} or full dopant-defect cluster solutions (**DDC.FULL**) will be described later in Initialization of Dopant-Defect Clusters, p. F-25.

With neither C_{cl1} nor turning on **DDC.FULL**, C_{cl2} is initialized in a regrowth region by:

$$C_{cl0} = (1 - \mathbf{C2FRAC.A})C_{cl,total} \qquad \text{Equation F-5}$$

$$C_{cl2} = \mathbf{C2FRAC.A} \cdot C_{cl,total} \qquad \text{Equation F-6}$$

When the Boolean parameter **CL.INI.A** is turned on, the total dopant clustering concentration in a regrowth region is initialized as follows:

$$C_{cl,total} = max(C - min(\mathbf{ACT.AMOR}, \mathbf{ACT.NI} \cdot n_i), 0.0) \qquad \text{Equation F-7}$$

or:

$$C_{cl,total} = max(C - \mathbf{ACT.AMOR},\ 0.0) \text{ when } \mathbf{ACT.NI} = 0 \quad \text{Equation F-8}$$

However, turning off **`CL.INI.A`** specifies to assume that all dopants are activated, i.e.: $C_{cl,total} = 0.0$.

In a crystalline region, since the vacancy concentration is reduced by C_{cl2} formation of **`C2.INI.F`**$\cdot\ \Delta C$, i.e.:

$$\Delta V = -\mathbf{VFRACM} \cdot \mathbf{C2.INI.F} \cdot \Delta C \quad \text{Equation F-9}$$

where ΔC is the impurity concentration newly introduced by implantation, C_{cl2} is initialized by:

$$C_{cl2} = C_{cl2,0} + min(\mathbf{C2.INI.F} \cdot \Delta C, V/ \mathbf{VFRACM}) \quad \text{Equation F-10}$$

where $C_{cl2,0}$ is the C_{cl2} concentration before implantation.

Arsenic Clustering and Deactivation

The arsenic–vacancy cluster forms with four arsenic atoms and one vacancy in vacancy-rich region near surface after implantation.(Equation F-11). It also can be formed by deactivation which generates free interstitials when the temperature decreases (Equation F-12):

$$4As^{+} + V \leftrightarrow As_4V \quad \text{Equation F-11}$$

$$4As^{+} \leftrightarrow As_4V + I \quad \text{Equation F-12}$$

To set up both Equation F-11 and Equation F-12, use one of the following:

- *c1_arsenic* solution
- *c2_arsenic* solution

In this example, *c2_arsenic* is solved by the built-in code to solve the reaction involving a vacancy, i.e. Equation F-11 and the USEIT implementation for Equation F-12.

It is assumed that *c2_arsenic* is initially set to zero. For simplicity, *c0_arsenic* is turned off by setting **`C0.TF.0= c0.tr.0=0.0`**. The reaction parameters **`DVC.T.0`**, **`DVC.T.E`**, **`DVC.F.0`**, and **`DVC.F.E`** are set to the values which are stored in the **`ASSIGN`** variables, *dvct0*, *dvcte*, *dvcf0*, and *dvcfe*, respectively. It is assumed that the assigned variables are already defined.

Example 1 Arsenic Clustering and Deactivation

```
$ Set initialization of c2_arsenic (starting zeros)
IMPURITY IMP=ARSENIC MAT=SILICON C2.INI.F=0.0 C2FRAC.A=0
$ Turn off c0_arsenic
IMPURITY IMP=ARSENIC MAT=SILICON C0.TF.0=0 C0.TR.0=0

$ Setup Equation F-11
IMPURITY IMP=ARSENIC MAT=SILICON VFRACM=0.25
IMPURITY IMP=ARSENIC MAT=SILICON +
         DVC.T.0=@DVCT0 DVC.T.E=@DVCTE
IMPURITY IMP=ARSENIC MAT=SILICON +
         DVC.F.0=@DVCF0 DVC.F.E=@DVCFE
IMPURITY IMP=ARSENIC MAT=SILICON +
         DVCF.D.N=4 DVCF.N.N=4 DVCF.V.N=1
IMPURITY IMP=ARSENIC MAT=SILICON +
         DVCR.N.N=0 DVCR.V.N=0
```

Next, Equation F-12 is implemented by **`USEIT`**. First, the global intermediates *Ni* for the intrinsic carrier concentration and *Cas* for the arsenic active concentration, which are expected to be frequently used, are defined:

```
$ Global intermediates
INTERMED NAME=NI  EXPR=3.87E16*EXP(-0.605/KT+1.5*LOG(TEMP))
INTERMED NAME=CAS EXPR=ACTIVE(ARSENIC)
```

Then Equation F-12 is implemented inside the '*AsDeclust*' scope. Instead of '*AsDeclust*', you can name the scope with any name you want as long as it is an alphanumeric string. The purpose of the intermediate scope is to make the script more readable and to avoid conflicts caused by possible duplication of intermediates when you define many intermediates here and there. To access the global intermediates *Ni* and *Cas* from inside scope, '#.' must be appended to a global intermediate name. (For details, see INTERMED Scope Definition, p. F-50.)

```
$ Setup Equation F-12
INTERMED SCOPE=AsDeclust
  INTERMED NAME=DDCT EXPR=@DDCT0*EXP(-(@DDCTE)/KT)
  INTERMED NAME=DDCF EXPR=@DDCF0*EXP(-(@DDCFE)/KT)
  INTERMED NAME=FOR  +
           EXPR=DDCF/DDCT*NI#.*(CAS#./NI#.)^4*ETA^4
  INTERMED NAME=REV  EXPR=1/DDCT*C2_ARSENIC*CI.RATIO
  INTERMED NAME=G    EXPR=FOR-REV
  EQUATION IMP=C2_ARSENIC MAT=SI ADD=G
  EQUATION IMP=INTERST    MAT=SI ADD=0.25*G
INTERMED ENDSCOPE
```

Now all equations have been set up. Before starting simulation, specify the initial mesh and the proper models:

```
MESH   DY.SURF=0.001 LY.SURF=0.001 DY.ACT=0.005 LY.ACT=0.5
METHOD PD.FULL ACT.FULL
METHOD INIT.TIM=1E-12
```

To make a vacancy-rich region near the surface after I-V recombination, Frenkel-pair interstitial location is shifted down by, for example, 5 Angstroms.

The **DY.FP** parameter in an **IMPLANT** command is used (see Profiling Implant Damages, p. F-33). After implant, the **Diffusion** command performs the regrowth of an amorphous region and the instantaneous I-V recombination and then solves diffusion equation if the diffusion time is larger than zero. In this example, zero time diffusion is performed only to see the net point–defect profiles just after regrowth.

```
INIT
IMPLANT ARSENIC DOSE=1E13 ENERGY=10 DY.FP=0.0005
DIFFUSE TEMP=600 TIME=0
SELECT  Z=LOG10(ARSENIC) +
    TITLE="AFTER REGROWTH W/ 5A DOWNSHIFT OF INTERSTITIAL"
PLOT.1  X.V=0 TOP=21 BOT=15 RIGHT=0.1 COLOR=4
SELECT  Z=LOG10(INTERST)
PLOT.1  X.V=0 !CL !AX COLOR=1
SELECT  Z=LOG10(VACANCY)
PLOT.1  X.V=0 !CL !AX COLOR=1 LINE=2
LABEL   LABEL=V X=0.001 Y=20.15
LABEL   LABEL=I X=0.01 Y=20.15
LABEL   LABEL=AS X=0.02 Y=18.1 COLOR=4

DIFFUSE TEMP=600 TIME=10 T.FINAL=800
```

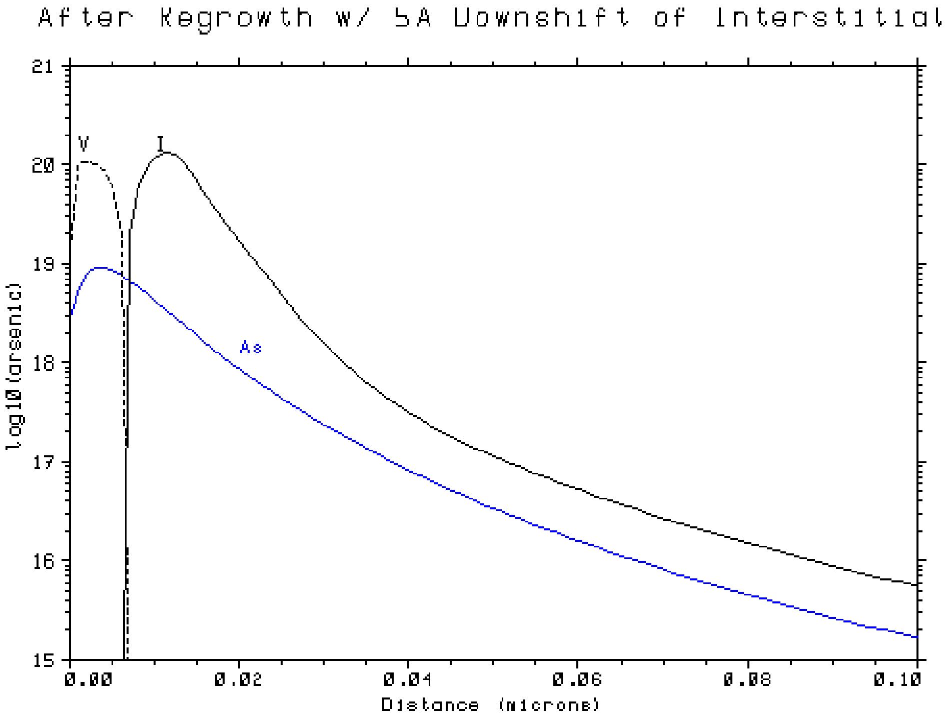


Figure F-1 After Regrowth with 5-Angstrom downshift of Interstitial

Enhancements in DDC.FULL

The **DDC.FULL** model is a superset of the **ACT.FULL** model. In other words, **DDC.FULL** solves not only *C0*, *C1*, and *C2* solutions, but also the dopant–defect cluster solutions for full dynamics. Thus, invoking the **DDC.FULL** model no longer requires turning on **ACT.FULL**. The following statement turns on the **DDC.FULL** model:

```
METHOD DDC.FULL
```

The following parameter names have been updated:

DDC.D.N (alias: **DDC.N.D**)	Dopant atom amount in **IMP.DDC**
DDC.I.N (alias: **DDC.N.I**)	Silicon atom amount in **IMP.DDC**
DDC.V.N (alias: **DDC.N.V**)	Vacancy amount in **IMP.DDC**
DDC.R (alias: **DDCLUSTE**)	Cluster solution in right-hand side of reaction

The following parameters have been added:

DDC.L	Cluster solution in left-hand side of reaction
DDC.EQ.0 (units: cm^{-3})	Equilibrium prefactor
DDC.EQ.E (units: eV)	Equilibrium activation energy
DDCPRINT	Print the **DDC** parameter values of **IMPURITY**

The reaction is defined by specifying either **DDC.L** or **DDC.R** with one among **IP.DDC**, **I.DDC**, **VP.DDC**, or **V.DDC**:

$$\mathrm{DDC.L} + \begin{bmatrix} \mathrm{IP.DDC} \\ \mathrm{I.DDC} \\ \mathrm{VP.DDC} \\ \mathrm{V.DDC} \end{bmatrix} \leftrightarrow \mathrm{DDC.R} \quad \text{Equation F-13}$$

You may not define both **DDC.L** and **DDC.R**. When you specify one of them, the other is automatically determined. **DDC.R** or **DDC.L** must be specified as a dopant–defect cluster solution. The **DDC.L** parameter is necessary when a dopant–defect cluster captures a point-defect and a dopant–defect pair is formed. For example:

$$BI_2 + V \leftrightarrow BI \quad \text{Equation F-14}$$

DDC.FULL with PD.FULL

Since the reactions of full dynamic dopant-defect clustering necessarily includes the reaction with dopant-defect pairs, the transient behavior of dopant-defect pairing must be coupled with the clustering reaction. It is the reason why the earlier version requires **`PD.5STR`** to simulate the **`DDC.FULL`** model. However, when the pairing reaction is faster than the clustering reaction, the local equilibrium for dopant-defect pairing can be assumed so as to simulate the **`DDC.FULL`** model with **`PD.FULL`**. In such a case, the point-defect equations are modified as follows:

$$\frac{\partial(I+I_c+I_{dds})}{\partial t}+\sum\frac{\partial M}{\partial t}=-\vec{\nabla}\cdot\left[-D_I I^{*}\left(\vec{\nabla}\left(\frac{I}{I^{*}}\right)+\left(\frac{I}{I^{*}}\right)\vec{\nabla}\ln I_i^{*}\right)+\sum\vec{J}_m\right]$$
$$+G_p-R_b-R_t-R_{sIcI}-R_{lIcI}-R_l-R_{dd}$$
$$-\sum R_{DDCM}-\sum R_{DDC,\ C+I}-\sum R_{DDC,\ I+M}$$

Equation F-15

$$\frac{\partial(V+V_c)}{\partial t}+\sum\frac{\partial N}{\partial t}=-\vec{\nabla}\cdot\left[-D_V V^{*}\left(\vec{\nabla}\left(\frac{V}{V^{*}}\right)+\left(\frac{V}{V^{*}}\right)\vec{\nabla}\ln V_i^{*}\right)+\sum\vec{J}_n\right]$$
$$+G_p-R_b-R_{sIcV}-R_{lIcV}-R_{dv}$$
$$-\sum R_{DDCN}-\sum R_{DDC,\ C+V}-\sum R_{DDC,\ V+N}$$

Equation F-16

where: $$G_p=\sum\left[\frac{K_{mv}\frac{V}{V*}}{K_m+K_{mv}\frac{V}{V*}}\left(\frac{\partial M}{\partial t}+\vec{\nabla}\cdot\vec{J}_m+R_{DDCM}\right)\right]$$
$$+\sum\left[\frac{K_{ni}\frac{I}{I*}}{K_n+K_{ni}\frac{I}{I*}}\left(\frac{\partial N}{\partial t}+\vec{\nabla}\cdot\vec{J}_n+R_{DDCN}\right)\right]$$

Equation F-17

$$R_{DDCM}=R_{DDC,\ S+M}+R_{DDC,\ I+M}+R_{DDC,\ I_{n-1}+M}+2R_{DDC,\ M+M}$$
$$+R_{DDC,\ C+M}+R_{DDC,\ M+N}-R_{DDC,\ XI_2+V}$$

Equation F-18

$$R_{DDCN} = R_{DDC,\ S+N} + R_{DDC,\ V+N} + R_{DDC,\ I_{n+1}+N} + 2R_{DDC,\ N+N} + R_{DDC,\ C+N} + R_{DDC,\ M+N} - R_{DDC,\ XV_2+I}$$

Equation F-19

Note:

When you set the parameter values so that the dopant-defect clustering reaction rate is comparable or faster than pairing reaction, turning on `PD.FULL` with `DDC.FULL` may cause a convergence problem.

Use of DDC.EQ.0 and DDC.EQ.E

The reverse reaction rate can be defined with either the **`DDC.KR.0`** and **`DDC.KR.E`** parameters or the **`DDC.EQ.0`** and **`DDC.EQ.E`** parameters. Unlike the **`DDC.KR.0`** and **`DDC.KR.E`** parameters, the **`DDC.EQ.0`** and **`DDC.EQ.E`** parameters do not depend on charge state of the reaction. The reverse reaction constant is calculated as follows:

$$r_{ddc} = \mathtt{DDC.KR.0} \cdot \exp\left(\frac{-\mathtt{DDC.KR.E}}{kT}\right)$$

Equation F-20

or:

$$r_{ddc} = \left(\mathtt{DDC.KF.0} \cdot \exp\left(\frac{-\mathtt{DDC.KF.E}}{kT}\right)\right)\left(\mathtt{DDC.EQ.0} \cdot \exp\left(\frac{-\mathtt{DDC.EQ.E}}{kT}\right)\right)$$

Equation F-21

Setting **`DDC.EQ.0`** or **`DDC.EQ.E`** overwrites all previous reverse reaction parameter values.

Example 2 Use of DDC.EQ.0 and DDC.EQ.E

```
$ DEFINE BORON-INTERSTITIAL CLUSTERING REACTION
IMPURITY IMP=B2I1 NEW TIF.NAME=B2I
IMPURITY IMP=BORON IMP.DDC=B2I1 DDC.D.N=2 DDC.N.I=1
IMPURITY IMP=BORON DDC.R=B2I1 IP.DDC +
         DDC.KF.0=2E-9 DDC.KF.E=1.58 +
         DDC.KR.0=2E11 DDC.KR.E=1.0
IMPURITY IMP=BORON DDCPRINT          ;$ (1)
IMPURITY IMP=BORON DDC.R=B2I1 IP.DDC +
         C.IPAIR=2 DDC.KR.0=5E12 DDC.KR.E=2.5
IMPURITY IMP=BORON DDCPRINT          ;$ (2)
IMPURITY IMP=BORON DDC.R=B2I1 IP.DDC +
         DDC.EQ.0=5E20 DDC.EQ.E=-0.38
IMPURITY IMP=BORON DDCPRINT          ;$ (3)
IMPURITY IMP=BORON DDC.R=B2I1 IP.DDC +
         DDC.KF.0=1e-8 DDC.KF.E=1.4
IMPURITY IMP=BORON DDCPRINT          ;$ (4)
IMPURITY IMP=BORON DDC.R=B2I1 IP.DDC C.IPAIR=1 +
         DDC.KF.0=5e-8 DDC.KF.E=2
IMPURITY IMP=BORON DDCPRINT          ;$ (5)
```

At (1), it prints all dynamic boron-defect clustering parameters as follows:

```
**************     B Dopant-Defect Clustering Parameters *********
B2I : 2B + 1I
------------------------------------------------------------------
(1) B + BIpair <-> B2I
CHG   DDC.KF.0     DDC.KF.E     DDC.KR.0     DDC.KR.E
all          T            T            T            T
 -6      2e-09         1.58        2e+11            1
 -5      2e-09         1.58        2e+11            1
 -4      2e-09         1.58        2e+11            1
 -3      2e-09         1.58        2e+11            1
 -2      2e-09         1.58        2e+11            1
 -1      2e-09         1.58        2e+11            1
  0      2e-09         1.58        2e+11            1
  1      2e-09         1.58        2e+11            1
  2      2e-09         1.58        2e+11            1
  3      2e-09         1.58        2e+11            1
  4      2e-09         1.58        2e+11            1
  5      2e-09         1.58        2e+11            1
  6      2e-09         1.58        2e+11            1
==================================================================
```

Since **C.IP.ALL** is turned on by default, the reaction parameter values for all k-charge state of (z+k) charged BI are same. The character 'T' denotes that **C.IP.ALL** is set to 'T' (true) value.

Then, by setting **C.IPAIR** to 2 and setting **DDC.KR.0** and **DDC.KR.E** to the other values, you can define different rates specific to the reverse reaction dissolving into a single positive charged BI pair and a single negative substitutional boron atom. At (2), it prints the following information. Note that **C.IP.ALL** is set to 'F' (false) value for the columns of **DDC.KR.0** and **DDC.KR.E**:

```
**************     B Dopant-Defect Clustering Parameters
B2I : 2B + 1I
------------------------------------------------------------------
(1) B + BIpair <-> B2I
CHG   DDC.KF.0     DDC.KF.E     DDC.KR.0     DDC.KR.E
all          T            T            F            F
 -6      2e-09         1.58        2e+11            1
 -5      2e-09         1.58        2e+11            1
 -4      2e-09         1.58        2e+11            1
 -3      2e-09         1.58        2e+11            1
 -2      2e-09         1.58        2e+11            1
 -1      2e-09         1.58        2e+11            1
  0      2e-09         1.58        2e+11            1
  1      2e-09         1.58        2e+11            1
  2      2e-09         1.58        5e+12          2.5
  3      2e-09         1.58        2e+11            1
  4      2e-09         1.58        2e+11            1
  5      2e-09         1.58        2e+11            1
  6      2e-09         1.58        2e+11            1
==================================================================
```

Then, the next command resets all reverse reaction rate values by setting **`DDC.EQ.0`** and **`DDC.EQ.E`**. At (3), it gives the following information:

```
**************     B Dopant-Defect Clustering Parameters
B2I : 2B + 1I
-------------------------------------------------------------------
(1) B + BIpair <-> B2I
CHG   DDC.KF.0     DDC.KF.E     DDC.KR.0     DDC.KR.E
all          T            T            T            T
 -6      2e-09         1.58        1e+12          1.2
 -5      2e-09         1.58        1e+12          1.2
 -4      2e-09         1.58        1e+12          1.2
 -3      2e-09         1.58        1e+12          1.2
 -2      2e-09         1.58        1e+12          1.2
 -1      2e-09         1.58        1e+12          1.2
  0      2e-09         1.58        1e+12          1.2
  1      2e-09         1.58        1e+12          1.2
  2      2e-09         1.58        1e+12          1.2
  3      2e-09         1.58        1e+12          1.2
  4      2e-09         1.58        1e+12          1.2
  5      2e-09         1.58        1e+12          1.2
  6      2e-09         1.58        1e+12          1.2

====================================================================
```

========Once **`DDC.EQ.0`** and **`DDC.EQ.E`** were set, changing forward reaction rates automatically updates reverse ones according to Equation F-21. At (4) and (5), the following tables are printed:

```
**************     B Dopant-Defect Clustering Parameters *********
B2I : 2B + 1I
-------------------------------------------------------------------
(1) B + BIpair <-> B2I
CHG   DDC.KF.0     DDC.KF.E     DDC.KR.0     DDC.KR.E
all          T            T            T            T
 -6      1e-08          1.4        5e+12         1.02
 -5      1e-08          1.4        5e+12         1.02
 -4      1e-08          1.4        5e+12         1.02
 -3      1e-08          1.4        5e+12         1.02
 -2      1e-08          1.4        5e+12         1.02
 -1      1e-08          1.4        5e+12         1.02
  0      1e-08          1.4        5e+12         1.02
  1      1e-08          1.4        5e+12         1.02
  2      1e-08          1.4        5e+12         1.02
  3      1e-08          1.4        5e+12         1.02
  4      1e-08          1.4        5e+12         1.02
  5      1e-08          1.4        5e+12         1.02
  6      1e-08          1.4        5e+12         1.02
===================================================================
```

```
**************     B Dopant-Defect Clustering Parameters *********
B2I : 2B + 1I
-------------------------------------------------------------------
(1) B + BIpair <-> B2I
CHG   DDC.KF.0     DDC.KF.E     DDC.KR.0     DDC.KR.E
all          F            F            F            F
 -6      1e-08          1.4        5e+12         1.02
 -5      1e-08          1.4        5e+12         1.02
 -4      1e-08          1.4        5e+12         1.02
 -3      1e-08          1.4        5e+12         1.02
 -2      1e-08          1.4        5e+12         1.02
 -1      1e-08          1.4        5e+12         1.02
  0      1e-08          1.4        5e+12         1.02
  1      5e-08            2      2.5e+13         1.62
  2      1e-08          1.4        5e+12         1.02
  3      1e-08          1.4        5e+12         1.02
  4      1e-08          1.4        5e+12         1.02
  5      1e-08          1.4        5e+12         1.02
  6      1e-08          1.4        5e+12         1.02
===================================================================
```

PAIR.EQU with DDC.FULL

In earlier versions, the **PAIR.EQU** method for automatic transition from 5-stream to 3-stream diffusion model (i.e., **PD.FULL**) was not allowed with the **DDC.FULL** model because the **DDC.FULL** model was required to solve the dopant–defect pairing equations. However, this restriction has been removed; now the **PAIR.EQU** method can be used with the **DDC.FULL** model.

This example simulates the boron uphill diffusion with the boron-interstitial clustering reaction by:

$$B_s + BI \leftrightarrow B_2I \qquad \text{Equation F-22}$$

Example 3 PAIR.EQU with DDC.FULL

```
$ DEFINE BORON-INTERSTITIAL CLUSTERING REACTION
$ NOTE: THE PARAMETER VALUES HERE ARE JUST FOR
$       DEMONSTRATION PURPOSE
IMPURITY IMP=B2I1 NEW TIF.NAME=B2I
IMPURITY IMP=BORON IMP.DDC=B2I1 DDC.D.N=2 DDC.N.I=1
IMPURITY IMP=BORON DDC.R=B2I1 IP.DDC +
         DDC.KF.0=2E-9 DDC.KF.E=1.58 +
         DDC.EQ.0=5E20 DDC.EQ.E=-0.38

$ 5-STREAM DIFFUION WITH AUTOMATIC TRANSITION TO PD.FULL
METHOD PD.5STR PAIR.EQ
METHOD DDC.FULL
METHOD INIT.TIM=1E-12

MESH DY.SURF=0.001 LY.SURF=0.001 DY.ACT=0.004 LY.ACT=0.3
INIT
DEPOSIT OXIDE THICKNES=0.001 SPACES=5
IMPLANT BORON DOSE=1E15 ENERGY=10      ;$ (1)
SELECT Z=LOG10(BORON) +
       TITLE="BORON UPHILL DIFFUSION WITH B2I"
PLOT.1 X.V=0 TOP=21 BOT=18 RIGHT=0.3
DIFFUSE TEMP=600 TIME=10 T.FINAL=800   ;$ (2)
SELECT Z=LOG10(BORON)
PLOT.1 X.V=0 !CL !AX COLOR=2 LINE=2
DIFFUSE TEMP=800 TIME=20               ;$ (3)
SELECT Z=LOG10(BORON)
PLOT.1 X.V=0 !CL !AX COLOR=2 LINE=3
```

The simulation starts with the 5-stream diffusion model. However, when boron-vacancy pairing reaction goes in local equilibrium, the transient equation for boron-vacancy pairing is automatically turned off. In other words, the 5-stream diffusion model becomes a 4-stream diffusion model, and later, the boron-interstitial pairing becomes a 3-stream diffusion model which is exactly same as **`PD.FULL`**. To know when the pairing equation is turned off, output the detail information with the **`OPTION VERBOSE`** and search the string "removed".

With **`OPTION VERBOSE`**, you will see the message saying that boron-vacancy pair solution, *vpair_boron*, is removed during the first diffusion at (2) as follows:

```
Time: current = 9.258    delta = 0.4288    new = 9.686     Nodes: 214
:
Solution is removed:  vpair_boron
*** vpair_boron solution has been removed.
Time: current = 9.686    delta = 0.3135    new = 10        Nodes: 214
```

And during the second diffusion at (3), the boron-interstitial pair solution, *ipair_boron*, also is removed:

```
Time: current = 5.3      delta = 2.232     new = 7.531     Nodes: 180
:
Solution is removed:  ipair_boron
*** ipair_boron solution has been removed.
Time: current = 7.531    delta = 2.767     new = 10.3      Nodes: 180
```

The simulation result of the example is shown in Figure F-2.

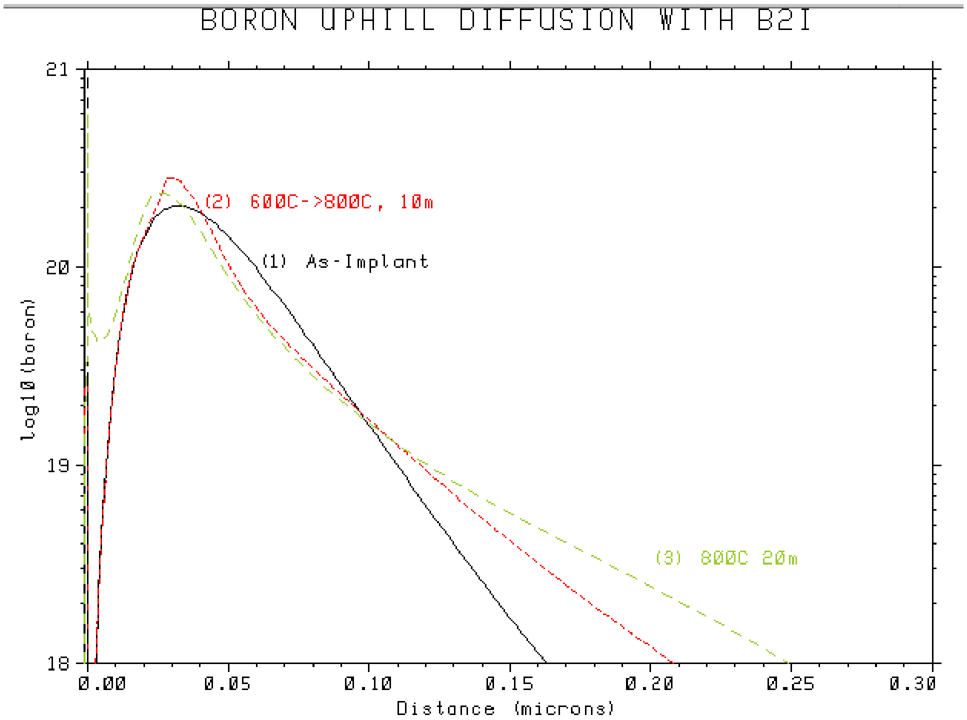


Figure F-2 Boron Uphill Diffusion

DDC.FULL with Vacancy

In earlier versions, the **`DDC.FULL`** model describes the full dynamics of dopant-defect clustering associated with interstitials or dopant-interstitial pairs. The **`DDC.FULL`** model has been extended to simulate the reactions with vacancy and dopant-vacancy pair as well. The following reactions have been added to the **`DDC.FULL`** model:

$$X_S^z + XV^{z+k} \leftrightarrow X_2V^l + (2z+k-l)p \tag{Equation F-23}$$

$$V^j + XV^{z+k} \leftrightarrow XV_2^l + (z+j+k-l)p \tag{Equation F-24}$$

$$XV^{z+j} + XV^{z+k} \leftrightarrow X_2V_2^l + (2z+j+k-l)p \tag{Equation F-25}$$

$$X_{n-1}V_{m-1}^j + XV^{z+k} \leftrightarrow X_nV_m^l + (z+j+k-l)p \tag{Equation F-26}$$

$$X_{n-1}I_{m+1}^j + XV^{z+k} \leftrightarrow X_nI_m^l + (z+j+k-l)p \tag{Equation F-27}$$

$$X_nV_{m-1}^j + V^k \leftrightarrow X_nV_m^l + (j+k-l)p \tag{Equation F-28}$$

$$X_n I_{m+1}^j + V^k \leftrightarrow X_n I_m^l + (j + k - l)p \qquad \text{Equation F-29}$$

$$I_{n+1} + XV^{z+k} \leftrightarrow XI_n^l + (z + k - l)p \qquad \text{Equation F-30}$$

$$XV_2^l + I^j \leftrightarrow XV^{z+k} + (-z + l + j - k)p \qquad \text{Equation F-31}$$

$$XI_2^l + V^j \leftrightarrow XI^{z+k} + (-z + l + j - k)p \qquad \text{Equation F-32}$$

$$XI^{z+j} + XV^{z+k} \leftrightarrow X_2^l + (2z + j + k - l)p \qquad \text{Equation F-33}$$

$$X_n V_{m+1}^j + I^k \leftrightarrow X_n V_m^l + (j + k - l)p \qquad \text{Equation F-34}$$

$$X_{n-1} V_{m+1}^j + XI^{z+k} \leftrightarrow X_n V_m^l + (z + j + k - l)p \qquad \text{Equation F-35}$$

The solution for the new dopant-defect cluster is defined in the **IMPURITY** statement. For example,

```
IMPURITY IMP=As2V1 NEW CHARGE=0 TIF.NAME=As2V
```

The method to solve the equation is defined in the **METHOD** statement. For example:

```
METHOD IMP=As2V1 LU NONE ^SYMM TIME.STE + REL.ERR=0.01 ABS.ERR=1.0e7
```

And the cluster must be defined as the dopant-defect cluster of a dopant species. For example, *As2V1* is the dopant-defect cluster of arsenic.

```
IMPURITY IMP=ARSENIC IMP.DDC=As2V1 DDC.D.N=2 DDC.V.N=1
```

Assigning *As2V1* to the parameter **IMP.DDC** links the solution *As2V1* to the impurity arsenic as a dopant-defect cluster of arsenic atoms which consists of two arsenic atoms and one vacancy. The parameters **DDC.D.N** and **DDC.V.N** specifies the number of dopant atoms and vacancies in the dopant-defect cluster, respectively:

The following syntax in the **IMPURITY** statement is the summary of the parameters associated with the definition of the new dopant-defect cluster for the **DDC.FULL** model:

```
IMPURITY IMPURITY=<c>
          IMP.DDC=<c> DDC.D.N=<n> { DDC.I.N=<n>| DDC.V.N=<n> }
```

The formation reaction of the dopant-defect cluster is defined in the **IMPURITY** statement. For example, the reaction:

```
Ass + AsV   <-->   As2V1
```

is defined with:

```
IMPURITY IMP=ARSENIC MAT=Silicon DDC.R=As2V1 +
         VP.DDC +
         DDC.KF.0=2E-6 DDC.KF.E=1.58 +
         DDC.EQ.0=5E20 DDC.EQ.E=-0.38
```

The Boolean parameter **`VP.DDC`** specifies that the dopant-defect cluster **`DDC.R`** is formed by reaction with the dopant-vacancy pair. Since *As2V1* consists of two arsenic atoms and one vacancy, the other reactant which reacts with one arsenic-vacancy pair should be one substitutional arsenic atom.

The parameters **`DDC.KF.0`** and **`DDC.KF.E`** specify the prefactor and the activation energy barrier of the forward reaction rate, respectively. The equilibrium constant is given by **`DDC.EQ.0`** and **`DDC.EQ.E`** parameters. The equilibrium constant does not depend on the charge states of reactants.

Note:

> ***The values of the parameters `DDC.KF.0` and `DDC.EQ.0` always must be given in $cm^3 sec^{-1}$ and cm^{-3} units, respectively, regardless of turning on/off the Boolean parameter `CM.SEC`.***

The following syntax in the **`IMPURITY`** statement is the summary of the parameters associated with reaction for the **`DDC.FULL`** model:

```
IMPURITY IMPURITY=<c>
         [ { DDC.L=<c> | DDC.R=<c> }
           { I.DDC | IP.DDC | V.DDC | VP.DDC }
           [ { C.INTERS=<c> | C.I.ALL } ]
           [ { C.VACANC=<c> | C.V.ALL } ]
           [ { C.IPAIR=<c>  | C.IP.ALL } ]
           [ { C.VPAIR=<c>  | C.VP.ALL } ]
           [DDC.KF.0=<n>] [DDC.KF.E=<n>]
           [  { [DDC.KR.0=<n>] [DDC.KR.E=<n>]  }
             |{  [DDC.EQ.0=<n>] [DDC.EQ.E=<n>]  } ]
         ]
```

`V.DDC` specifies that the dopant-defect cluster **`DDCLUST`** forms by the reaction with a vacancy while **`VP.DDC`** specifies the dopant-vacancy pair to be involved in the reaction.

When any solution required for a reaction is not available, the reaction is ignored. For example, the following command specifies the generation or dissolution of As_4V by the reaction between a arsenic-vacancy pair and the arsenic cluster AS_3. If there is no AS_3 solution, the reaction is ignored:

```
IMPURITY IMP=B MAT=Si DDC.R=As4V1 VP.DDC C.VP.ALL +
           DDC.KF.0=1E-5 DDC.KF.E=2.5 DDC.EQ.0=1E20 DDC.EQ.E=-1
```

X_S + XV

$$X_S^z + XV^{z+k} \leftrightarrow X_2V^l + (2z+k-l)p$$

$$R_{DDC,\ S+N} = k_{fSN}\left(S\left(\frac{N}{\alpha_n}\right) - r_{EQ,\ SN}C_{\mathrm{DDC.R}}\eta^{-2z+q_{\mathrm{DDC.R}}}\right) \qquad \text{Equation F-36}$$

where:

$$k_{fSN} = \sum_{k} f_{SN,\ k}\left(\frac{D_{nk}}{D_{Nk}}\right)\eta^{-k} \qquad \text{Equation F-37}$$

S and *N* denote the substitutional concentration, [*X*s] and the dopant-vacancy pair concentration, [*XV*]. For the reaction between the substitutional atom and the (*z*+*k*) charged dopant-vacancy pair:

$$f_{SN,\ k} = \mathbf{DDC.KF.0}\exp\left(-\frac{\mathbf{DDC.KF.E}}{kT}\right) \qquad \text{Equation F-38}$$

k is defined by the **`C.VPAIR`** parameter. And the equilibrium constant $r_{EQ,\ SN}$ is given by:

$$r_{EQ,\ SN} = \mathbf{DDC.EQ.0}\exp\left(-\frac{\mathbf{DDC.EQ.E}}{kT}\right) \qquad \text{Equation F-39}$$

When **`C.VP.ALL`** is true (default), the Equation F-36 is simplified as follows:

$$R_{DDC,\ S+N} = f_{SN}(S\cdot\ N - r_{EQ,\ SN}\alpha_n C_{\mathrm{DDC.R}}\eta^{-2z+q_{\mathrm{DDC.R}}}) \qquad \text{Equation F-40}$$

The equations of the dopant-vacancy pair and the dopant-defect cluster are described as follows:

$$\frac{\partial N}{\partial t} = \left.\frac{\partial N}{\partial t}\right|_0 - R_{DDC,\ S+N} \qquad \text{Equation F-41}$$

$$\frac{\partial C_{\mathtt{DDC.R}}}{\partial t} = \left.\frac{\partial C_{\mathtt{DDC.R}}}{\partial t}\right|_0 + R_{DDC,\ S+N} \qquad \text{Equation F-42}$$

V + XV

$$V^j + XV^{z+k} \leftrightarrow XV_2^l + (z+j+k-l)p$$

$$R_{DDC,\ V+N} = k_{fVN}\left(\left(\frac{V}{V^*}\right)\left(\frac{N}{\alpha_n}\right) - r_{EQ,\ VN}C_{\mathrm{DDC.R}}\eta^{-z+q_{\mathrm{DDC.R}}}\right) \qquad \text{Equation F-43}$$

where:

$$k_{fVN} = V_i^* \sum_{j,\ k} f_{VN,\ jk}\phi_{Vj}\left(\frac{D_{nk}}{D_{Nk}}\right)\eta^{-(j+k)} \qquad \text{Equation F-44}$$

$$f_{VN,\ jk} = \mathbf{DDC.KF.0}\exp\left(-\frac{\mathbf{DDC.KF.E}}{kT}\right) \qquad \text{Equation F-45}$$

$$r_{EQ,\ VN} = \mathbf{\mathtt{DDC.EQ.0}}\exp\left(-\frac{\mathbf{\mathtt{DDC.EQ.E}}}{kT}\right) \qquad \text{Equation F-46}$$

True **`C.VP.ALL`** and **`C.V.ALL`** (default) simplify Equation F-43 to:

$$R_{DDC,\ V+N} = f_{VN}(V\cdot\ N - r_{EQ,\ VN}\alpha_n V^* C_{\mathrm{DDC.R}}\eta^{-z+q_{\mathrm{DDC.R}}}) \qquad \text{Equation F-47}$$

The equations of the dopant-vacancy pair, the vacancy and the dopant-defect cluster take the reaction into account as follows:

$$\frac{\partial N}{\partial t} = \left.\frac{\partial N}{\partial t}\right|_0 - R_{DDC,\ V+N} \qquad \text{Equation F-48}$$

$$\frac{\partial V}{\partial t} = \left.\frac{\partial V}{\partial t}\right|_0 - R_{DDC,\ V+N} \qquad \text{Equation F-49}$$

$$\frac{\partial C_{\mathrm{DDC.R}}}{\partial t} = \left.\frac{\partial C_{\mathrm{DDC.R}}}{\partial t}\right|_0 + R_{DDC,\ V+N} \qquad \text{Equation F-50}$$

XV + XV

$$XV^{z+j} + XV^{z+k} \leftrightarrow X_2V_2^l + (2z+j+k-l)p$$

$$R_{DDC,\ N+N} = k_{fNN}\left(\left(\frac{N}{\alpha_n}\right)^2 - r_{EQ,\ NN}C_{\mathrm{DDC.R}}\eta^{-2z+q_{\mathrm{DDC.R}}}\right) \qquad \text{Equation F-51}$$

where:

$$k_{fNN} = \sum_{j,\ k} f_{NN,\ jk}\left(\frac{D_{nj}}{D_{Nj}}\right)\left(\frac{D_{nk}}{D_{Nk}}\right)\eta^{-(j+k)} \qquad \text{Equation F-52}$$

$$f_{NN,\ jk} = \mathbf{\mathtt{DDC.KF.0}}\exp\left(-\frac{\mathbf{\mathtt{DDC.KF.E}}}{kT}\right) \qquad \text{Equation F-53}$$

$$r_{EQ,\ NN} = \mathbf{\mathtt{DDC.EQ.0}}\exp\left(-\frac{\mathbf{\mathtt{DDC.EQ.E}}}{kT}\right) \qquad \text{Equation F-54}$$

When **`C.VP.ALL`** is true (default), the Equation F-51 is simplified as follows:

$$R_{DDC,\ N+N} = f_{NN}(N^2 - r_{EQ,\ NN}\alpha_n^2 C_{\mathrm{DDC.R}}\eta^{-2z+q_{\mathrm{DDC.R}}}) \qquad \text{Equation F-55}$$

Since two dopant-vacancy pairs are consumed for formation of one cluster, the dopant-vacancy pair equation and the cluster equation must be adjusted as follows:

$$\frac{\partial N}{\partial t} = \left.\frac{\partial N}{\partial t}\right|_0 - 2R_{DDC,\ N+N}$$

Equation F-56

$$\frac{\partial C_{\mathrm{DDC.R}}}{\partial t} = \left.\frac{\partial C_{\mathrm{DDC.R}}}{\partial t}\right|_0 + R_{DDC,\ N+N}$$

Equation F-57

$X_{n-1}V_{m-1}$ + XV and $X_{n-1}I_{m+1}$ + XV

$$X_{n-1}V^j_{m-1} + XV^{z+k} \leftrightarrow X_nV^l_m + (z+j+k-l)p$$

$$X_{n-1}I^j_{m+1} + XV^{z+k} \leftrightarrow X_nI^l_m + (z+j+k-l)p$$

Equation F-58

$$R_{DDC,\ \mathrm{C}+N} = k_{fCN}\left(C_{\mathrm{DDC.L}}\left(\frac{N}{\alpha_n}\right) - r_{EQ,\ CN}C_{\mathrm{DDC.R}}\eta^{-z-q_{\mathrm{DDC.L}}+q_{\mathrm{DDC.R}}}\right)$$

Equation F-59

where:

$$k_{fCN} = \sum_k f_{CN,\ k}\left(\frac{D_{nk}}{D_{Nk}}\right)\eta^{-k}$$

Equation F-60

$$f_{CN,\ k} = \mathtt{DDC.KF.0}\exp\left(-\frac{\mathtt{DDC.KF.E}}{kT}\right)$$

Equation F-61

$$r_{EQ,\ CN} = \mathtt{DDC.EQ.0}\exp\left(-\frac{\mathtt{DDC.EQ.E}}{kT}\right)$$

Equation F-62

When **`C.VP.ALL`** is true (default), Equation F-59 will be:

$$R_{DDC,\ C+N} = f_{CN}(C_{\mathrm{DDC.L}}N - r_{EQ,\ CN}\alpha_n C_{\mathrm{DDC.R}}\eta^{-z-q_{\mathrm{DDC.L}}+q_{\mathrm{DDC.R}}})$$

Equation F-63

$$\frac{\partial N}{\partial t} = \left.\frac{\partial N}{\partial t}\right|_0 - R_{DDC,\ C+N}$$

Equation F-64

$$\frac{\partial C_{\mathrm{DDC.L}}}{\partial t} = \left.\frac{\partial C_{\mathrm{DDC.L}}}{\partial t}\right|_0 - R_{DDC,\ C+N}$$

Equation F-65

$$\frac{\partial C_{\mathrm{DDC.R}}}{\partial t} = \left.\frac{\partial C_{\mathrm{DDC.R}}}{\partial t}\right|_0 + R_{DDC,\ C+N}$$

Equation F-66

Note:

You may define either `DDC.L` or `DDC.R`, but not both. When one (e.g. `DDC.R`) is specified, the other one (e.g. `DDC.L`) will be recognized automatically by counting the dopant atoms and point-defects of the remainder.

$X_{n-1}V_{m-1}$ + V and $X_{n-1}I_{m+1}$ + V

$$X_n V^j_{m-1} + V^k \leftrightarrow X_n V^l_m + (j+k-l)p$$

$$X_n I^j_{m+1} + V^k \leftrightarrow X_n I^l_m + (j+k-l)p$$

$$R_{DDC,\ C+V} = k_{fCV}\left(C_{\mathrm{DDC.L}}\left(\frac{V}{V^*}\right) - r_{EQ,\ CV} C_{\mathrm{DDC.R}} \eta^{-q_{\mathrm{DDC.L}} + q_{\mathrm{DDC.R}}}\right)$$

Equation F-67

where:

$$k_{fCV} = V_i^* \sum_k f_{CV,\ k} \phi_{Vk} \eta^{-k}$$

Equation F-68

$$f_{CV,\ k} = \mathtt{DDC.KF.0} \exp\left(-\frac{\mathtt{DDC.KF.E}}{kT}\right)$$

Equation F-69

$$r_{EQ,\ CV} = \mathtt{DDC.EQ.0} \exp\left(-\frac{\mathtt{DDC.EQ.E}}{kT}\right)$$

Equation F-70

When **`C.V.ALL`** is true (default), then Equation F-67 will be:

$$R_{DDC,\ C+V} = f_{CV}(C_{\mathrm{DDC.L}} V - r_{EQ,\ CV} C_{\mathrm{DDC.R}} V^* \eta^{-q_{\mathrm{DDC.L}} + q_{\mathrm{DDC.R}}})$$

Equation F-71

To take this reaction into account, the equations of a vacancy and two dopant-defect clusters are modified as follows:

$$\frac{\partial V}{\partial t} = \left.\frac{\partial V}{\partial t}\right|_0 - R_{DDC,\ C+V}$$

Equation F-72

$$\frac{\partial C_{\mathrm{DDC.L}}}{\partial t} = \left.\frac{\partial C_{\mathrm{DDC.L}}}{\partial t}\right|_0 - R_{DDC,\ C+V}$$

Equation F-73

$$\frac{\partial C_{\mathrm{DDC.R}}}{\partial t} = \left.\frac{\partial C_{\mathrm{DDC.R}}}{\partial t}\right|_0 + R_{DDC,\ C+V}$$

Equation F-74

$\mathbf{I_{n+1}}$ + XV $I_{n+1} + XV^{z+k} \leftrightarrow XI_n^l + (z+k-l)p$

$$R_{DDC,\ I_{n+1}+N} = k_{fI_{n+1}N}\left(I_{n+1}\left(\frac{N}{\alpha_n}\right) - r_{EQ,\ I_{n+1}N}C_{\mathrm{DDC.R}}\eta^{-z+q_{\mathrm{DDC.R}}}\right)$$

Equation F-75

where I_{n+1} is the small interstitial cluster with size *n+1:*

$$k_{fI_{n+1}N} = \sum_k f_{I_{n+1}N,\ k}\left(\frac{D_{nk}}{D_{Nk}}\right)\eta^{-k}$$

Equation F-76

$$f_{I_{n+1}N,\ k} = \texttt{DDC.KF.0}\exp\left(-\frac{\texttt{DDC.KF.E}}{kT}\right)$$

Equation F-77

$$r_{EQ,\ I_{n+1}N} = \texttt{DDC.EQ.0}\exp\left(-\frac{\texttt{DDC.EQ.E}}{kT}\right)$$

Equation F-78

The parameters for the reaction are defined in the **`IMPURITY`** statement. When **`C.VP.ALL`** is true (default):

$$R_{DDC,\ I_{n+1}+N} = f_{I_{n+1}N}(I_{n+1}N - r_{EQ,\ I_{n+1}N,\ k}\alpha_n C_{\mathrm{DDC.R}}\eta^{-z+q_{\mathrm{DDC.R}}})$$

Equation F-79

The reaction is taken into account in the equations for dopant-vacancy pairing and interstitial clustering as follows:

$$\frac{\partial N}{\partial t} = \left.\frac{\partial N}{\partial t}\right|_0 - R_{DDC,\ I_{n+1}+N}$$

Equation F-80

$$\frac{\partial I_{n+1}}{\partial t} = \left.\frac{\partial I_{n+1}}{\partial t}\right|_0 - R_{DDC,\ I_{n+1}+N}$$

Equation F-81

$$\frac{\partial C_{\mathrm{DDC.R}}}{\partial t} = \left.\frac{\partial C_{\mathrm{DDC.R}}}{\partial t}\right|_0 + R_{DDC,\ I_{n+1}+N}$$

Equation F-82

$\mathbf{XV_2}$ + I $XV_2^l + I^j \leftrightarrow XV^{z+k} + (-z+l+j-k)p$

$$R_{DDC,\ XV_2+I} = k_{fXV_2I}\left(C_{\mathrm{DDC.L}}\left(\frac{I}{I^*}\right) - r_{EQ,\ XV_2I}\left(\frac{N}{\alpha_n}\right)\eta^{z-q_{\mathrm{DDC.L}}}\right)$$

Equation F-83

where:

$$k_{fXV_2I} = I_i^* \sum_j f_{XV_2I,j} \phi_{Ij} \eta^{-j}$$

Equation F-84

$$f_{XV_2I,j} = \texttt{DDC.KF.0} \exp\left(-\frac{\texttt{DDC.KF.E}}{kT}\right)$$

Equation F-85

$$r_{EQ,\ XV_2I} = \texttt{DDC.EQ.0} \exp\left(-\frac{\texttt{DDC.EQ.E}}{kT}\right)$$

Equation F-86

When `C.I.ALL` is true (default), then Equation F-83 will be:

$$R_{DDC,\ XV_2+I} = f_{XV_2I}\left(C_{\mathrm{DDC.L}} I - r_{EQ,\ XV_2I} I^* \left(\frac{N}{\alpha_n}\right)\eta^{z-q_{\mathrm{DDC.L}}}\right)$$

Equation F-87

The reaction is taken into account in the equations for dopant-vacancy pairs and interstitials as follows:

$$\frac{\partial N}{\partial t} = \left.\frac{\partial N}{\partial t}\right|_0 + R_{DDC,\ XV_2+I}$$

Equation F-88

$$\frac{\partial I}{\partial t} = \left.\frac{\partial I}{\partial t}\right|_0 - R_{DDC,\ XV_2+I}$$

Equation F-89

$$\frac{\partial C_{\mathrm{DDC.L}}}{\partial t} = \left.\frac{\partial C_{\mathrm{DDC.L}}}{\partial t}\right|_0 - R_{DDC,\ XV_2+I}$$

Equation F-90

XI_2 + V

$$XI_2^l + V^j \leftrightarrow XI^{z+k} + (-z+l+j-k)p$$

Equation F-91

$$R_{DDC,\ XI_2+V} = k_{fXI_2V}\left(C_{\mathrm{DDC.L}}\left(\frac{V}{V^*}\right) - r_{EQ,\ XI_2V}\left(\frac{M}{\alpha_m}\right)\eta^{z-q_{\mathrm{DDC.L}}}\right)$$

Equation F-92

where:

$$k_{fXI_2V} = V_i^* \sum_j f_{XI_2V,j} \phi_{Vj} \eta^{-j}$$

Equation F-93

$$f_{XI_2V,j} = \texttt{DDC.KF.0} \exp\left(-\frac{\texttt{DDC.KF.E}}{kT}\right)$$

Equation F-94

$$r_{EQ,\ XI_2V} = \texttt{DDC.EQ.0}\exp\left(-\frac{\texttt{DDC.EQ.E}}{kT}\right)$$ Equation F-95

When **`C.V.ALL`** is true (default), then Equation F-92 will be:

$$R_{DDC,\ XI_2+V} = f_{XI_2V}\left(C_{\mathrm{DDC.L}}V - r_{EQ,\ XI_2V}V^*\left(\frac{M}{\alpha_m}\right)\eta^{z-q_{\mathrm{DDC.L}}}\right)$$

Equation F-96

The reaction is taken into account in the equations for dopant-interstitial pairs and vacancies as follows:

$$\frac{\partial M}{\partial t} = \left.\frac{\partial M}{\partial t}\right|_0 + R_{DDC,\ XI_2+V}$$ Equation F-97

$$\frac{\partial V}{\partial t} = \left.\frac{\partial V}{\partial t}\right|_0 - R_{DDC,\ XI_2+V}$$ Equation F-98

$$\frac{\partial C_{\mathrm{DDC.L}}}{\partial t} = \left.\frac{\partial C_{\mathrm{DDC.L}}}{\partial t}\right|_0 - R_{DDC,\ XI_2+V}$$ Equation F-99

XI + XV

$$XI^{z+j} + XV^{z+k} \leftrightarrow X_2^l + (2z+j+k-l)p$$ Equation F-100

$$R_{DDC,\ M+N} = k_{fMN}\left(\left(\frac{M}{\alpha_m}\right)\left(\frac{N}{\alpha_n}\right) - r_{EQ,\ MN}C_{\mathrm{DDC.R}}\eta^{-2z+q_{\mathrm{DDC.R}}}\right)$$

Equation F-101

where:

$$k_{fMN} = \sum_{j,\ k} f_{MN,jk}\left(\frac{D_{mj}}{D_{Mj}}\right)\left(\frac{D_{nk}}{D_{Nk}}\right)\eta^{-j-k}$$ Equation F-102

$$f_{MN,jk} = \texttt{DDC.KF.0}\exp\left(-\frac{\texttt{DDC.KF.E}}{kT}\right)$$ Equation F-103

$$r_{EQ,\ MN} = \texttt{DDC.EQ.0}\exp\left(-\frac{\texttt{DDC.EQ.E}}{kT}\right)$$ Equation F-104

When **`C.IP.ALL`** and **`C.VP.ALL`** are true (default), then Equation F-101 will be:

$$R_{DDC,\ M+N} = f_{MN}(MN - r_{EQ,\ MN}\alpha_m\alpha_n C_{\mathrm{DDC.R}}\eta^{-2z+q_{\mathrm{DDC.R}}})$$

Equation F-105

The reaction is taken into account in the equations for dopant-interstitial pairs and dopant-vacancy pairs as follows:

$$\frac{\partial M}{\partial t} = \left.\frac{\partial M}{\partial t}\right|_0 - R_{DDC,\ M+N} \quad \text{Equation F-106}$$

$$\frac{\partial N}{\partial t} = \left.\frac{\partial N}{\partial t}\right|_0 - R_{DDC,\ M+N} \quad \text{Equation F-107}$$

$$\frac{\partial C_{\mathrm{DDC.R}}}{\partial t} = \left.\frac{\partial C_{\mathrm{DDC.R}}}{\partial t}\right|_0 - R_{DDC,\ M+N} \quad \text{Equation F-108}$$

Initialization of Dopant-Defect Clusters

The parameters **`DDCTOT.C`** and **`DDCTOT.A`** define the fraction of total dopant-defect clusters to total dopants in a crystalline region and an amorphous region, respectively. For correct initialization, these parameters must be defined when:

- **`DDC.T.0`** of both C_{cl1} and C_{cl2} are non-zero and **`ACT.FULL`** is turned on

or:

- **`DDC.FULL`** is turned on.

Note:

Hereinafter, it will be described by assuming that at least one of the above conditions is satisfied.

The parameters **`DDCFRACC`** and **`DDCFRACA`** define the fractions of a dopant-defect cluster of full dynamics to the total dopant-defect clusters in a crystalline region after implants and a regrowth region, i.e. an amorphous region after implants, respectively.

The cluster solutions are initialized after implantation as follows:

$$C_{cl0} = C_{cl0}^{\prime} + \Delta C_{cl0} \quad \text{Equation F-109}$$

where:

$$\Delta C_{cl0} = (1 - \mathtt{DDCTOT.C}) \cdot \ \mathtt{C0.INI.F} \cdot \ C_{implant} \quad \text{Equation F-110}$$

$$C_{cl1} = C_{cl1}^{\prime} + \Delta C_{cl1} \quad \text{Equation F-111}$$

where:

$$\Delta C_{cl1} = \frac{\mathtt{C1.INI.F} \cdot \mathtt{DDCTOT.C}}{\mathtt{C1.INI.F} + \mathtt{C2.INI.F} + \sum \mathtt{DDCFRACC}} \cdot C_{implant}$$

Equation F-112

$$C_{cl2} = C'_{cl2} + \Delta C_{cl2}$$

Equation F-113

where:

$$\Delta C_{cl2} = \frac{\mathtt{C2.INI.F} \cdot \mathtt{DDCTOT.C}}{\mathtt{C1.INI.F} + \mathtt{C2.INI.F} + \sum \mathtt{DDCFRACC}} \cdot C_{implant}$$

Equation F-114

$$C_{DDC} = C'_{DDC} + \Delta C_{DDC}$$

Equation F-115

where:

$$\Delta C_{DDC} = \frac{\mathtt{DDCFRACC} \cdot \mathtt{DDCTOT.C}}{\mathtt{C1.INI.F} + \mathtt{C2.INI.F} + \sum \mathtt{DDCFRACC}} \cdot C_{implant}$$

Equation F-116

The net capture of point-defects by clusters is given by:

$$\Delta I = \mathtt{IFRACM} \cdot \Delta C_{cl1} + \sum \frac{\mathtt{DDC.I.N} \cdot \Delta C_{DDC}}{\mathtt{DDC.D.N}}$$

Equation F-117

$$\Delta V = \mathtt{VFRACM} \cdot \Delta C_{cl2} + \sum \frac{\mathtt{DDC.V.N} \cdot \Delta C_{DDC}}{\mathtt{DDC.D.N}}$$

Equation F-118

Since the formation of dopant-defect clusters is limited by the number of point-defects, Equation F-112, Equation F-114, and Equation F-116 are modified by:

$$\Delta C_{cl1}^{\mathrm{mod}} = \Delta C_{cl1} \cdot min(1.0, I / \Delta I)$$

Equation F-119

$$\Delta C_{cl2}^{\mathrm{mod}} = \Delta C_{cl2} \cdot min(1.0, V / \Delta V)$$

Equation F-120

$$\Delta C_{DDC}^{\mathrm{mod}} = \Delta C_{DDC} \cdot min\left(1.0, \frac{\mathtt{DDC.I.N} \cdot I / \Delta I + \mathtt{DDC.V.N} \cdot V / \Delta V}{\mathtt{DDC.I.N} + \mathtt{DDC.V.N}}\right)$$

Equation F-121

Note:

You may not define both non-zero `DDC.I.N` and `DDC.V.N`. But it is allowed to define both zero `DDC.I.N` and `DDC.V.N`, i.e. dopant-only clusters. In such a case, $\Delta C_{DDC}^{\mathrm{mod}} = \Delta C_{DDC}$.

In recrystallization during the anneal process following implant, clusters are initialized in an amorphous region as follows:

$$C_{cl0} = (1 - \mathbf{DDCTOT.A}) \cdot C_{cl,\ total}$$

Equation F-122

$$C_{cl1} = \frac{\mathbf{C1FRAC.A} \cdot \mathbf{DDCTOT.A}}{\mathbf{C1FRAC.A} + \mathbf{C2FRAC.A} + \sum \mathbf{DDCFRACA}} \cdot C_{cl,\ total}$$

Equation F-123

$$C_{cl2} = \frac{\mathbf{C2FRAC.A} \cdot \mathbf{DDCTOT.A}}{\mathbf{C1FRAC.A} + \mathbf{C2FRAC.A} + \sum \mathbf{DDCFRACA}} \cdot C_{cl,\ total}$$

Equation F-124

$$C_{DDC} = \frac{\mathbf{DDCFRACA} \cdot \mathbf{DDCTOT.A}}{\mathbf{C1FRAC.A} + \sum \mathbf{DDCFRACA}} \cdot C_{cl,\ total}$$

Equation F-125

Pressure-Dependent Pair/Substitutional Formation Energy

The macroscopic diffusivity is described by:

$$D_m = \sum_k D_{mk}\eta^{-k} = \left(\frac{M}{S}\right)_i^* \sum_k D_{Mk}\phi_{mk}\eta^{-k}$$

Equation F-126

where $\left(\frac{M}{S}\right)_i^*$ is the ratio of the dopant-interstitial pair concentration to the substitutional concentration in intrinsic equilibrium, D_{Mk} and ϕ_{mk} are the microscopic diffusivity and the charge state fraction of (z+k) charged dopant-interstitial pairs, respectively. The ratio α_m of the dopant-interstitial concentration to the substitutional concentration in equilibrium is described by:

$$\alpha_m = \left(\frac{M}{S}\right)^* = \left(\frac{M}{S}\right)_i^* \sum_k \phi_{mk}\eta^{-k} = \sum_k \left(\frac{D_{mk}}{D_{Mk}}\right)\eta^{-k}$$

Equation F-127

The pressure dependence of the dopant-interstitial pair migration energy can be taken into account in Equation F-126:

$$D_m' = \exp\left(-\frac{p\,\mathtt{VMIPAIR}}{kT}\right) \cdot \sum_k D_{mk}\eta^{-k}$$

Equation F-128

where p is the hydrostatic pressure, and **`VMIPAIR`** is the activation volume in A^3.for the migration energy of a dopant-interstitial pair. This model is invoked by turning on the **`ST.DIFF`** parameter in the **`METHOD`** statement.

The dependence of the formation energy on pressure is modeled by:

$$D_{mk} = \left(\frac{M}{S}\right)_i^* D_{Mk}\phi_{mk} = \exp\left(-\frac{p\mathtt{VALPHAM}}{kT}\right) \cdot \left(\frac{M}{S}\right)_{i,\ p=0}^* D_{Mk}\phi_{mk}$$

Equation F-129

where **`VALPHAM`** is the activation volume difference (units: A^3) between the formation energies of a dopant-interstitial pair and a substitutional atom. This model is invoked by turning on the **`ST.ALPHA`** parameter in the **`METHOD`** statement.

Likewise, for the dopant-vacancy pair:

$$D_n' = \exp\left(-\frac{p\mathtt{VMVPAIR}}{kT}\right) \cdot \sum_k D_{nk}\eta^{-k} \text{ with } \mathtt{ST.DIFF}$$

Equation F-130

$$D_{nk} = \left(\frac{N}{S}\right)_i^* D_{Nk}\phi_{nk} = \exp\left(-\frac{p\mathtt{VALPHAN}}{kT}\right) \cdot \left(\frac{N}{S}\right)_{i,\ p=0}^* D_{Nk}\phi_{nk} \text{ with } \mathtt{ST.ALPHA}$$

Equation F-131

When the **`ST.ALPHA`** parameter is turned on, the flux equation of the pair diffusion includes the flux due to spatial variation of pressure.

$$\vec{J}_m = \exp\left(-\frac{p\mathtt{VALPHAM}}{kT}\right) D_{mk,\ p=0}\left(\vec{\nabla}\left(\frac{M}{\alpha_m}\right) - \frac{M}{\alpha_m}\frac{\mathtt{VALPHAM}}{kT}\vec{\nabla}p\right)$$

Equation F-132

$$\vec{J}_n = \exp\left(-\frac{p\mathtt{VALPHAN}}{kT}\right) D_{nk,\ p=0}\left(\vec{\nabla}\left(\frac{N}{\alpha_n}\right) - \frac{N}{\alpha_n}\frac{\mathtt{VALPHAN}}{kT}\vec{\nabla}p\right)$$

Equation F-133

Note:

The default values of `VALPHAM` and `VALPHAN` are zeros. Setting them to negative values will result in the diffusion enhancement under compressive stress.

Note:

In the `IMPURITY` statement, setting a value to the `VD` parameter is the same as setting the same value to both `VMIPAIR` and `VMVPAIR` parameters.

Adaptive Gridding at Amorphous and Crystalline Boundary

The interstitial distribution is necessarily abrupt near the amorphous and crystalline boundary after regrowth because the boundary location is determined by comparing implant damage concentration to the **MAX.DAM** parameter value and then the interstitial concentration is reset to the equilibrium value in the regrowth region. Thus the interstitial dose so Transient Enhanced Diffusion (TED) is sensitive to the mesh density near the boundary region. The adaptive gridding method based on the phase transition between amorphous and crystalline regions makes denser mesh near the boundary so that it increases simulation accuracy. The following definition is defined in *s4init*:

```
METHOD IMP=DAMAGE MAT=SILICON +
       ABS.ADAP=0.05 REL.ADAP=0.05 MIN.SPAC=0.001
```

Unlike the adaptive gridding of other solutions which uses the absolute and relative errors based on the solution values, the adaptive gridding for damage solution is performed on the transformed profile of damage:

$$s = 1 + \tanh[\mathbf{DAM.GRAD} \times \ln(D_0 / \mathbf{MAX.DAMA})] \qquad \text{Equation F-134}$$

The **ABS.ADAP** and **REL.ADAP** parameter values for damage are associated with the value *s* of Equation F-134, not with the damage concentration itself.

Turning on the Boolean parameter **DAM.ADAP** (default: false) in the **METHOD** statement enables the automatic adaptive gridding on the region near the amorphous and crystalline boundary region. It is turned off by default.

```
METHOD DAM.ADAP
```

Figure F-3 shows that the **DAM.ADAP** method generates dense grids along the amorphous and crystalline boundary.

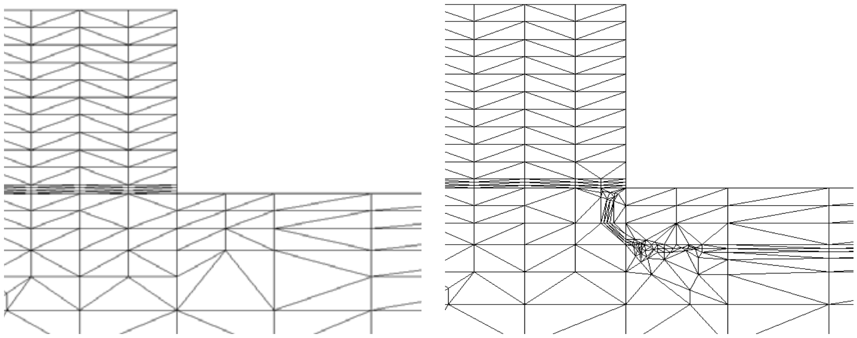


Figure F-3 The Mesh Comparison Before and After Recrystallization of the Amorphous Region

Example 4 Initial Grid Space vs. Interstitial Dose after Regrowth

Figure F-4 shows the interstitial dose change after regrowth with respect to the initial grid space. The dose oscillates with a large fluctuation without the **DAM.ADAP** method. The data were extracted from the following example:

```
METHOD DAM.ADAP  ;$ and !DAM.ADAP
EXTRACT OUT.F=test.dat
LOOP STEPS=10 INDEX=JJ
   ASSIGN NAME=DY N.V=0.002 RATIO=1.6
   MESH DY.SURF=@DY LY.SURF=@DY DY.ACT=@DY LY.ACT=0.5
   INIT
   IMPLANT IMP=GERMANIUM DOSE=1E15 ENERGY=20 NAME=TAURUS
   IMPLANT BORON DOSE=1E14 ENERGY=5 NAME=TAURUS
   DIFFUSE TEMP=800 TIME=0
   SELECT Z=INTERST
   EXTRACT NAME=IDOSE P1.X=0 P2.X=0 P1.Y=0 P2.Y=1.0 +
           INT.EXT ASSIGNED PREFIX="@DY "
L.END
EXTRACT CLOSE
```

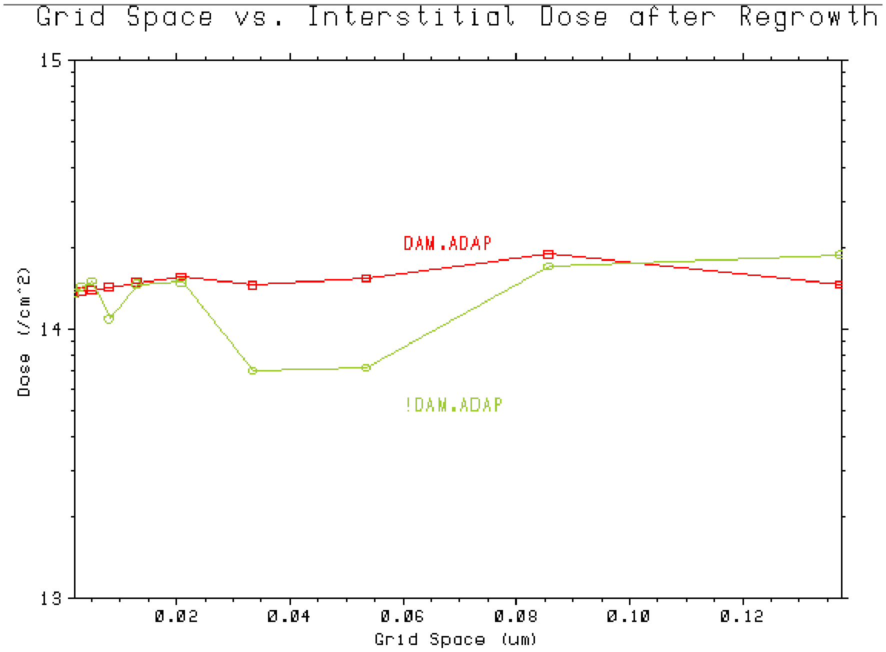


Figure F-4 Initial Grid Space vs. Interstitial Dose after Regrowth

Note:

For Monte Carlo implantations, DAM.ADAP is not applied to the damage profile from the Monte Carlo implant. Instead, it works on the damage profile from the analytic implant which is performed for the regrid purpose of the Monte Carlo implant.

Modifiable Hobler's Damage Moments

You now can specify the parameters of the implant damage model on the **MOMENT** statement or in a table contained in a file. The implant damage model has been extended to allow the lateral spread of the damage profile to be different from the vertical spread and to vary (linearly) with depth. Parameters for the implant damage model come from one of the following sources:

- The **MOMENT DAMAGE** statement can be used to specify the **RANGE, SIGMA**, **LAMBDA**, **LSIGMA**, **LSLOPE**, and **SCALE** parameters used by the damage implant model. **LAMBDA** specifies the location and slope of the exponential tail of the damage profile, and **SCALE** specifies the magnitude of the profile. The remaining parameters have the same meaning as the for the Gaussian model for implanted doping profiles. Parameters specified with **MOMENT DAMAGE** are used on subsequent **IMPLANT** statements that specify the **MOMENTS** parameter. Parameters not specified on a **MOMENT DAMAGE** statement are obtained from one of the following sources. A **MOMENT CLEAR** statement or a **MOMENT DAMAGE** statement with no model parameters cancels the effects of any preceding **MOMENT DAMAGE** statement.
- If **DAM.TABL** is specified on the **IMPLANT** statement or on an **IMPURITY** statement for the implanted dopant, damage parameters are taken from the specified table in an implant damage moments file. The file must be specified on the **DAM.FILE** parameter on the **IMPLANT** statement, by the *S4IDAM* environment variable; otherwise the file named "s4idam" in the current working directory is used. The format of the implant damage moments file is described below. A **DAM.TABL** specified on an **IMPLANT** statement takes precedence over **DAM.TABL** specified on an **IMPURITY** statement.
- Damage parameters not specified on a **DAMAGE MOMENT** statement or in a damage moments file are calculated from the built-in model of Hobler, as in previous releases of TSUPREM-4. **LSIGMA** is assumed to be the same as **SIGMA** and **LSLOPE** is zero. The Hobler's model ignores the exponential tail of the damage profile at low energies for light ions. This is only done if **LAMBDA** is obtained from the built-in model; if you specify **LAMBDA** on a **MOMENT DAMAGE** statement or in an implant damage moments file, the specified value is used to compute the exponential tail regardless of the implant energy or ion type.

Parameters Added to MOMENT Command

- Boolean **DAMAGE** –specifies that the remaining parameters on the **MOMENT** statement are used for the damage calculation. In particular, **RANGE** and **SIGMA** specify Hobler's a**2** and **a3** parameters. **LSIGMA** and **LSLOPE** specify the lateral spread and depth dependence of the lateral spread for the damage model. (Default: false, i.e., parameters specify the dopant profile as in previous versions.)

- float **LAMBDA**–is the slope of the exponential part of the damage profile (Hobler's **a1** parameter). (Default: Value taken from implant damage moments file or built-in model.)
- float **SCALE**–is the damage scaling factor (Hobler's **nv[?]** parameter; related to the number of Frenkel pairs created for each implanted dopant ion). (Default: Value taken from implant damage moments file or built-in model.)

The **RANGE**, **SIGMA**, **LSIGMA**, and **LSLOPE** on the **MOMENT** command parameters apply to implant damage model when **DAMAGE** is specified.

Parameters Added to IMPLANT Statement

- char **DAM.TABL**–is the name of a table of damage parameters in an implant damage file. Values of the **RANGE**, **SIGMA**, **LAMBDA**, **LSIGMA**, **LSLOPE**, and **SCALE** parameters for implant damage are taken from the specified table in the implant damage file (unless specified on a preceding **MOMENT** statement). (Default: If **DAM.TABL** is not specified on the **IMPLANT** statement, a table name specified on an **IMPURITY** statement is used or parameters are obtained from the built-in equations (as in previous versions).
- char **DAM.FILE**–is the name of a file continuing tables of damage parameters. (Default: The file specified by the environment variable *S4IDAM* or the file named "s4idam" in the current working directory.

Parameter Added to IMPURITY Statement

char **DAM.TABL**–is the name of a table of damage parameters in an implant damage file. Values of the **RANGE**, **SIGMA**, **LAMBDA**, **LSIGMA**, **LSLOPE**, and **SCALE** parameters for implant damage are taken from the specified table in the implant damage file unless a different table is specified on the **IMPLANT** statement or parameters from a **MOMENT** statement are used.

Note:

It is not possible to specify separate tables for boron and BF2 on the **IMPURITY** ***statement.***

Implant Damage Moment File Format

A new file format is defined to contain tables of parameters for the damage model. A new environment variable *S4IDAM* is defined as the default path to the implant damage file. The format of the implant damage file is the same as that used by Taurus, except:

- Multiple tables may be specified in a file. A line in the form table `<tablename>` where `<tablename>` is the name specified by **DAM.TABL** must appear before the header line in each table.
- The only implant condition recognized in the header line is "energy".

- Blank lines may be inserted between tables.

The data values are the same as in the Taurus file, i.e., range, sigma, lateral spread, dl (depth dependence of lateral spread), asym (asymmetry of vertical profile; not used), kurt (kurtosis of vertical profile; not used), lambda, and (scale) factor. This format allows for expansion to include other implant conditions (e.g., tilt) at a later time. It also allows the existing data in Taurus to be copied without change for use in TSUPREM-4.

Profiling Implant Damages

To provide the flexibility to control implant damage profiles, the new parameters have been added to the **`IMPLANT`** command:

- float **`V.PLUS`** –specifies the +n model of vacancy. (Default: 0.0.)

$$V = V_0 + \mathrm{V.PLUS} \cdot C_{implant} \quad \text{Equation F-135}$$

- float **`DX.FP`**–specifies the right-shift of interstitial locations to vacancy locations in Frenkel-pair generation. (Default: 0.0 um.)
- float **`DY.FP`**–specifies the downward shift of interstitial locations to vacancy locations in Frenkel-pair generation. (Default: 0.0 um.)
- float **`DX.IPLUS`**–specifies the right-shift of +n model interstitial locations to implanted dopant locations. (Default: 0.0 um.)
- float **`DY.IPLUS`**–specifies the downward shift of +n model interstitial locations to implanted dopant locations. (Default: 0.0 um.)
- float **`DX.VPLUS`**–specifies the right-shift of +n model vacancy locations to implanted dopant locations. (Default: 0.0 um.)
- float **`DY.IPLUS`**–specifies the downward shift of +n model vacancy locations to implanted dopant locations. (Default: 0.0 um.)

Example 5 5 A Shift of Interstitial profile of Frenkel-Pairs

```
MESH DY.SURF=0.001 LY.SURF=0.001 DY.ACT=0.01 LY.ACT=0.5
METHOD PD.FULL
INIT
IMPLANT BORON DOSE=1E14 ENERGY=10 DY.FP=0.0005 D.PLUS=0.0
DIFFUSE TEMP=800 TIME=0
SELECT Z=LOG10(BORON) TITLE="5A SHIFTED FP I PROFILE"
PLOT.1 X.V=0 TOP=21 BOT=15 RIGHT=0.3 COLOR=2
SELECT Z=LOG10(INTERST)
PLOT.1 X.V=0 !CL !AX COLOR=3 LINE=2
SELECT Z=LOG10(VACANCY)
PLOT.1 X.V=0 !CL !AX COLOR=4 LINE=2
```

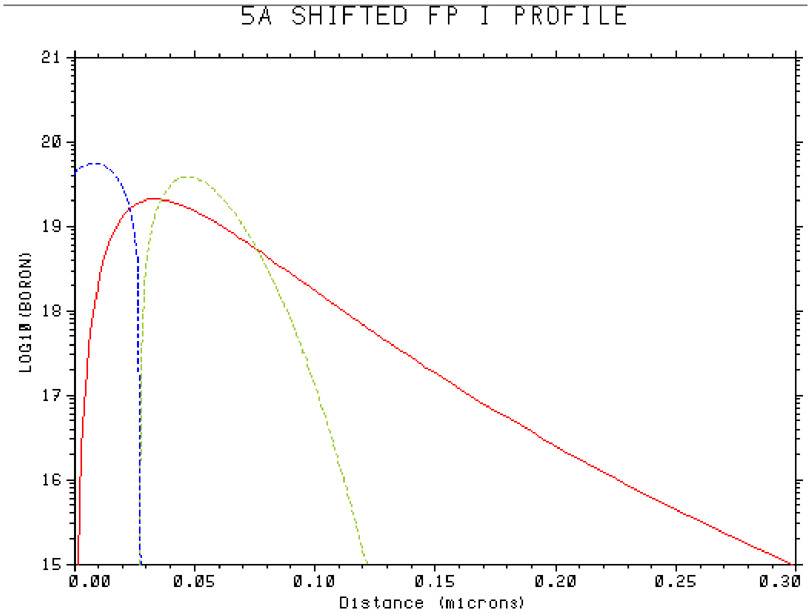


Figure F-5 Shift of Interstitial Profile of Frenkel-Pairs

Example 6 0.01 um Shift of +n Interstitial Profile

```
MESH DY.SURF=0.001 LY.SURF=0.001 DY.ACT=0.01 LY.ACT=0.5
METHOD PD.FULL
INIT
IMPLANT BORON DOSE=1E14 ENERGY=10 DY.IPLUS=0.01
DIFFUSE TEMP=800 TIME=0
SELECT Z=LOG10(BORON) TITLE=”100A SHIFTED +N I PROFILE”
PLOT.1 X.V=0 TOP=21 BOT=15 RIGHT=0.3 COLOR=2
SELECT Z=LOG10(INTERST)
PLOT.1 X.V=0 !CL !AX COLOR=3 LINE=2
SELECT Z=LOG10(VACANCY)
PLOT.1 X.V=0 !CL !AX COLOR=4 LINE=2
```

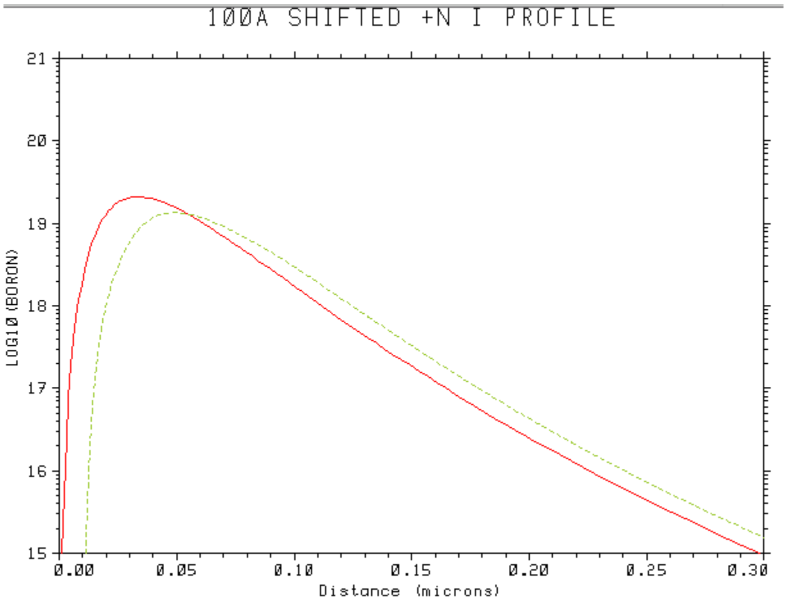


Figure F-6 0.01um Shift of +n Interstitial Profile

Example 7 –0.01 um Shift of +1 Vacancy profile

```
MESH DY.SURF=0.001 LY.SURF=0.001 DY.ACT=0.01 LY.ACT=0.5
METHOD PD.FULL
INIT
IMPLANT BORON DOSE=1E14 ENERGY=10 DY.VPLUS=-0.01 V.PLUS=1.0
DIFFUSE TEMP=800 TIME=0
SELECT Z=LOG10(BORON) TITLE=”-100A SHIFTED +1 V PROFILE”
PLOT.1 X.V=0 TOP=21 BOT=15 RIGHT=0.3 COLOR=2
SELECT Z=LOG10(INTERST)
PLOT.1 X.V=0 !CL !AX COLOR=3 LINE=2
SELECT Z=LOG10(VACANCY)
PLOT.1 X.V=0 !CL !AX COLOR=4 LINE=2
```

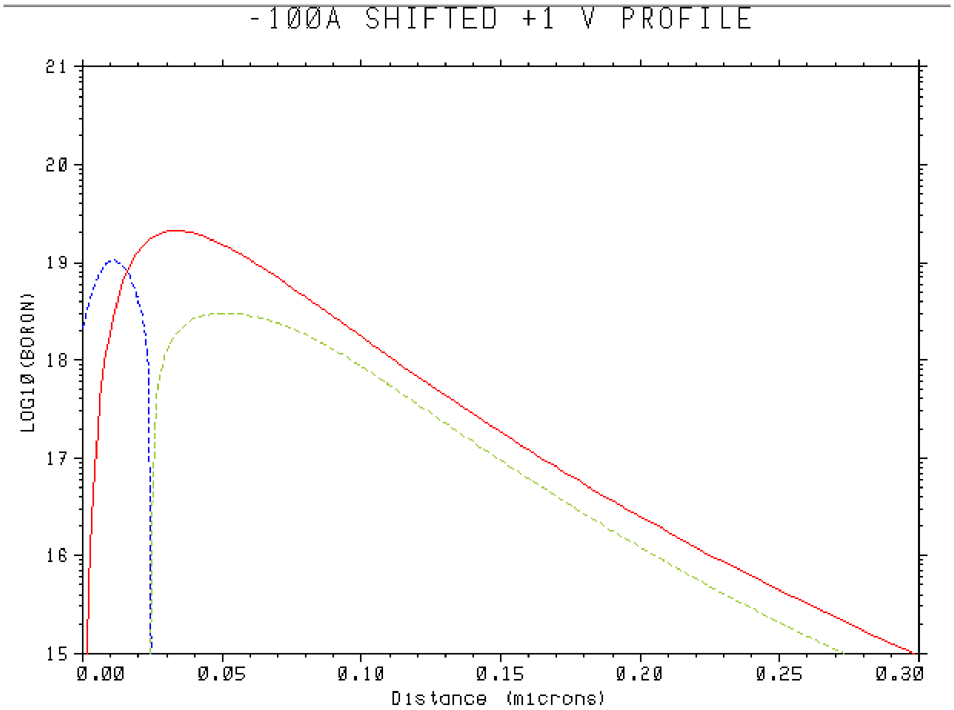


Figure F-7

Example 8 0.01 um Shift of +n Interstitials and -0.01um Shift of +1 Vacancies

```
MESH DY.SURF=0.001 LY.SURF=0.001 DY.ACT=0.01 LY.ACT=0.5
METHOD PD.FULL
INIT
IMPLANT BORON DOSE=1E14 ENERGY=10 +
        DY.IPLUS=0.01 DY.VPLUS=-0.01 V.PLUS=1.0
DIFFUSE TEMP=800 TIME=0
SELECT Z=LOG10(BORON) +
       TITLE="100A SHIFTED +N I AND -100A SHIFTED +1 V"
PLOT.1 X.V=0 TOP=21 BOT=15 RIGHT=0.3 COLOR=2
SELECT Z=LOG10(INTERST)
PLOT.1 X.V=0 !CL !AX COLOR=3 LINE=2
SELECT Z=LOG10(VACANCY)
PLOT.1 X.V=0 !CL !AX COLOR=4 LINE=2
```

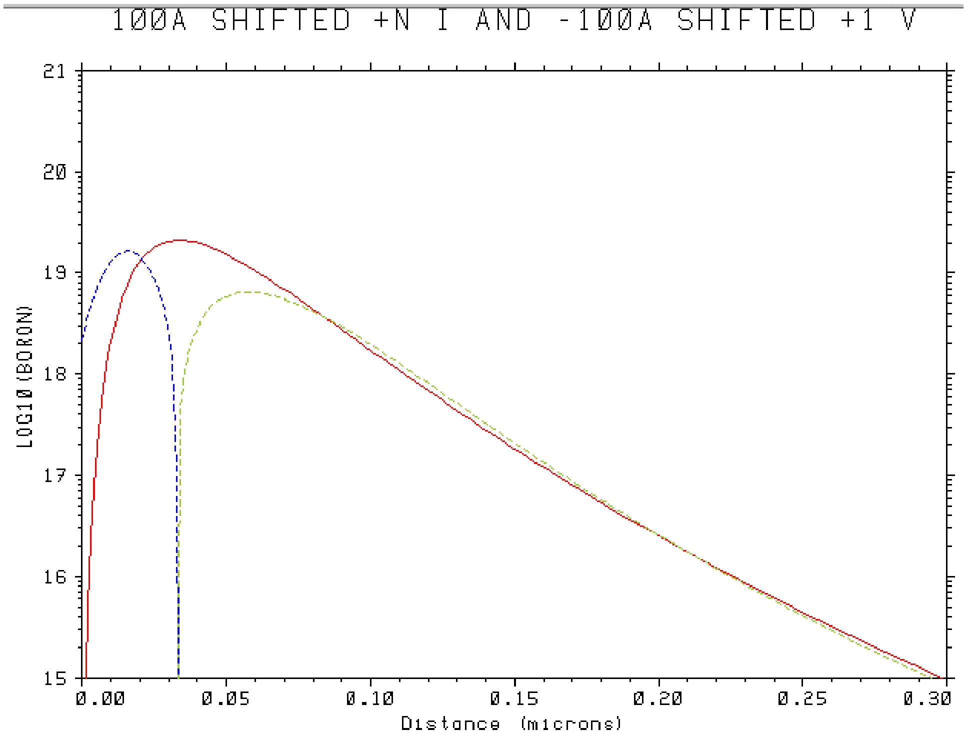


Figure F-8 0.01um Shift of +n Interstitials and -0.01um Shift of +1 Vacancies

Dopant-Dependent Diffusion and Reaction of Oxidants

The diffusivity and surface reaction rate of oxidants for oxidation can depend on the other impurity concentration. This feature allows users to define the expressions of multiplication factors to the diffusivity and reaction rate.

Parameters Added to the AMBIENT Command

- char **MODEL**–is the model name with that users can control turn in/off the model by the METHOD command.
- char **D.FACTOR**–is the expression of diffusivity multiplication factor.
- char **R.FACTOR**–is the expression of reaction multiplication factor.

The surface reaction rate k_s, and the diffusivity of oxidant in oxide D are modified to reflect the user-defined multiplication factor in the oxide:

$$D' = \mathbf{D.FACTOR} \cdot D \exp\left(-\frac{p\mathrm{VD}}{kT}\right) \qquad \text{Equation F-136}$$

$$k'_s = \mathbf{R.FACTOR} \cdot k_s \exp\left(-\frac{\sigma_n \mathbf{VR}}{kT}\right) \exp\left(-\frac{\sigma_t \mathbf{VT}}{kT}\right) \qquad \text{Equation F-137}$$

Note:

This feature is only applicable to the `VISCOEL` model.

Example 9 D.FACTOR and R.FACTOR in AMBIENT Command

```
METHOD MODEL=FLUORINE ENABLE
IMPURITY IMP=FLUORINE !PD.PAIR
IMPURITY IMP=FLUORINE MAT=OXIDE /MAT=AMBIENT +
         SEG.0=1.0 TRANS.0=0.1
AMBIENT INIT=0.0005
METHOD ST.HIST VISCOEL

$ FLUORINE DEPENDENT O2 DIFFUSION AND REACTION
AMBIENT O2 MODEL=OXRFACTOR R.FACTOR=1+FLUORINE/O2
AMBIENT O2 MODEL=OXDFACTOR D.FACTOR=1+FLUORINE/O2

$ DEFAULT
METHOD MODEL=OXRFACTOR !ENABLE
METHOD MODEL=OXDFACTOR !ENABLE
INIT TEMP=800
DIFFUSE TEMP=800 TIME=100 DRYO2 IMP=FLUORINE I.CONC=2.5E18
PRINT.1D X.V=0 LAYERS           ;$ TOX=62A

$ ONLY WITH FLUORINE-DEPENDENT REACTION
METHOD MODEL=OXRFACTOR ENABLE
INIT TEMP=800
DIFFUSE TEMP=800 TIME=100 DRYO2 IMP=FLUORINE I.CONC=2.5E18
PRINT.1D X.V=0 LAYERS           ;$ TOX=297A

$ ONLY WITH FLUORINE-DEPENDENT DIFFUSION
METHOD MODEL=OXRFACTOR !ENABLE
METHOD MODEL=OXDFACTOR ENABLE
INIT TEMP=800
DIFFUSE TEMP=800 TIME=100 DRYO2 IMP=FLUORINE I.CONC=2.5E18
PRINT.1D X.V=0 LAYERS           ;$ TOX=79A

$ WITH BOTH FLUORINE-DEPENDENT DIFFUSION AND REACTION
METHOD MODEL=OXRFACTOR ENABLE
INIT TEMP=800
DIFFUSE TEMP=800 TIME=100 DRYO2 IMP=FLUORINE I.CONC=2.5E18
PRINT.1D X.V=0 LAYERS           ;$ TOX=422A
```

Gradient-Doped Epitaxial Growth

A gradient of impurity concentration during deposition and epitaxy can be defined by linear, logarithmic, or arbitrary doping profiles.

Parameters Added to the DEPOSITION and EPITAXY Statements

- real_array **`IB.CONC`**–specifies the impurity concentrations at the bottom of the deposited (or grown) layer of the impurity species listed in the **`IMPURITY`** parameter. The impurity concentrations at the top of the layer must be specified by the **`IT.CONC`** parameter. Default: Uniform impurity concentrations (specified by **`IT.CONC`**) are used through the layer.
- Boolean **`LOG.GRAD`**–specifies that the impurity concentrations specified by **`IMPURITY`** will vary logarithmically between the values specified by **`IB.CONC`** and **`IT.CONC`**.
- string **`I.PROFIL`**–is an equation that specifies the concentration of the single impurity specified by **`IMPURITY`** as a function of **`F`**, the fraction of the total layer thickness (or epitaxy time). **`F`** must obey the rules for a **`DEFINE`** variable; it is recommended that it always by enclosed in braces and preceded by the @ character, i.e., it should always appear as `@{F}` in the equation.

Parameter Changed on the DEPOSITION and EPITAXY Statements

The **`I.CONC`** parameter on the **`DEPOSITION`** and **`EPITAXY`** statements is renamed to **`IT.CONC`** (the concentration at the top of the layer). **`I.CONC`** is added as an alias for **`IT.CONC`** to preserve compatibility with existing input files.

New Behavior

The program will work as before unless **`IB.CONC`** or **`I.PROFIL`** is specified. If **`IB.CONC`** is specified, the impurities named in the **`IMPURITY`** parameter will have the concentrations **`IB.CONC`** at the bottom of the deposited or grown layer and the concentrations **`IT.CONC`** at the top of the layer.

The concentration will vary linearly through the layer unless **`LOG.GRAD`** is specified, in which case the variation will be logarithmic. **`IB.CONC`** can only be used when **`IMPURITY`** and **`IT.CONC`** (alias **`I.CONC`**) also are specified.

If **`I.PROFIL`** is specified, it is interpreted as an equation specifying the concentration of a single impurity (specified by **`IMPURITY`**) as a function of `F`, the fraction of the layer thickness (`F=0` at the bottom of the layer and `F=1` at the top). The parameter `F` must obey the rules for a **`DEFINE`** variable; it is recommended that it be enclosed in braces and preceded by the @ character, i.e., `@{F}`.

Note:

The doping at interior nodes in the deposited layer is calculated by linear or logarithmic interpolation or by `I.PROFIL`. Thus, the total dose will be expected to differ from that calculated by analytical integration of the specified gradient (except in the special case of a linear gradient and an uniformly spaced grid.).

Example 10 Silicon Deposition with Ge that varies from 10% (Bottom) to 20% (Top)

```
LINE X LOC=0 SPAC=0.1
LINE X LOC=2 SPAC=0.1
LINE Y LOC=0 SPAC=0.1
LINE Y LOC=2 SPAC=0.1

METHOD PD.FERMI !IMP.ADAP ST.HIST VISCOEL !SKIP.SI

INIT
ETCH SILICON START X=1.0 Y=-1.0
ETCH CONTINUE X=1.0 Y=1.0
ETCH CONTINUE X=3.0 Y=1.0
ETCH DONE X=3.0 Y=-1.0

$ MISFIT STRAIN ONLY
DEPOSIT SILICON THICKNES=0.1 SPACES=10 STRAINED +
        IMP=GERMANIUM I.CONC=1E22 IB.CONC=0.5E22
DEPOSIT SILICON THICKNES=0.1 SPACES=10 +
        IMP=GERMANIUM I.CONC=1E22
DEPOSIT SILICON THICKNES=0.1 SPACES=10 STRAINED
```

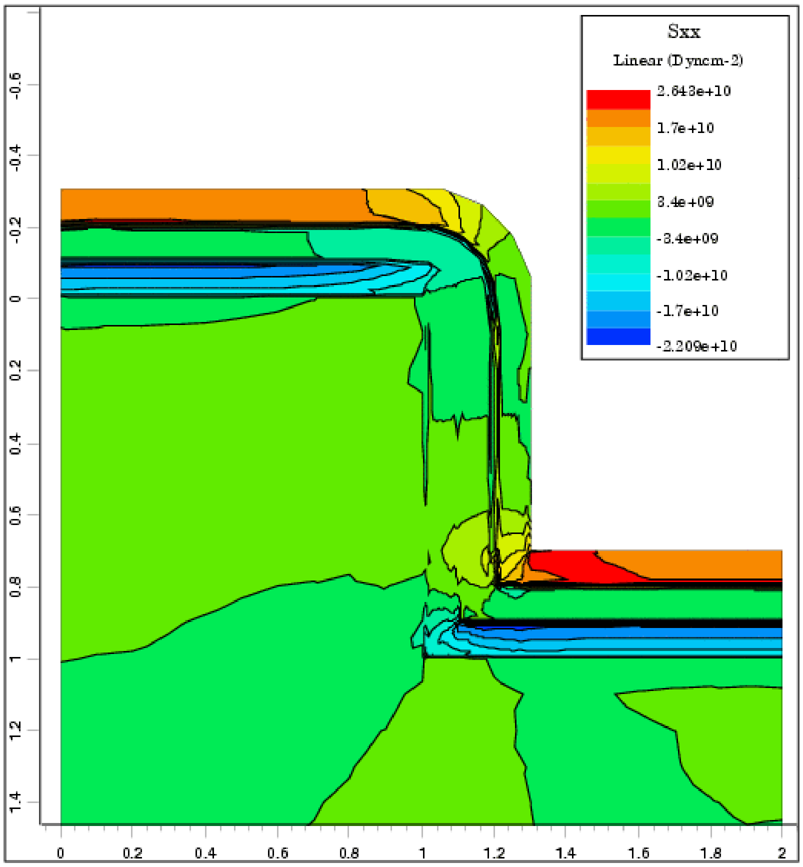


Figure F-9 ***Silicon Deposition with Ge that varies from 10% (Bottom) to 20% (Top)***

Example 11 Same as Example 10, but Use I.PROFIL:

```
LINE X LOC=0 SPAC=0.1
LINE X LOC=2 SPAC=0.1
LINE Y LOC=0 SPAC=0.1
LINE Y LOC=2 SPAC=0.1

METHOD PD.FERMI !IMP.ADAP ST.HIST VISCOEL !SKIP.SI

INIT
ETCH SILICON START X=1.0 Y=-1.0
ETCH CONTINUE X=1.0 Y=1.0
ETCH CONTINUE X=3.0 Y=1.0
ETCH DONE X=3.0 Y=-1.0

$ MISFIT STRAIN ONLY
DEPOSIT SILICON THICKNES=0.1 SPACES=10 STRAINED +
        IMP=GERMANIUM I.PROFIL=( 0.5E22 + @{F}*(1E22-0.5E22) )
DEPOSIT SILICON THICKNES=0.1 SPACES=10 +
        IMP=GERMANIUM I.CONC=1E22
DEPOSIT SILICON THICKNES=0.1 SPACES=10 STRAINED
```

Example 12 Epitaxially Grown Polysilicon and Silicon (Linear gradient boron profile)

```
LINE X LOC=0   SPAC=0.05
LINE X LOC=0.5 SPAC=0.02
LINE X LOC=1   SPAC=0.05
LINE Y LOC=0   SPAC=0.02
LINE Y LOC=0.3 SPAC=0.05

INIT
MATERIAL MAT=POLY ^POLYCR

DEPOSIT OXIDE THICK=0.05
ETCH MAT=OXIDE RIGHT P1.X=0.5
ETCH MAT=SILICON RIGHT P1.X=0.4 P2.Y=0.05

EPITAXY TIME=10 TEMP=600 THICK=0.1 IMP=BORON +
        IB.CONC=1E14 IT.CONC=1E15 SPACES=20
```

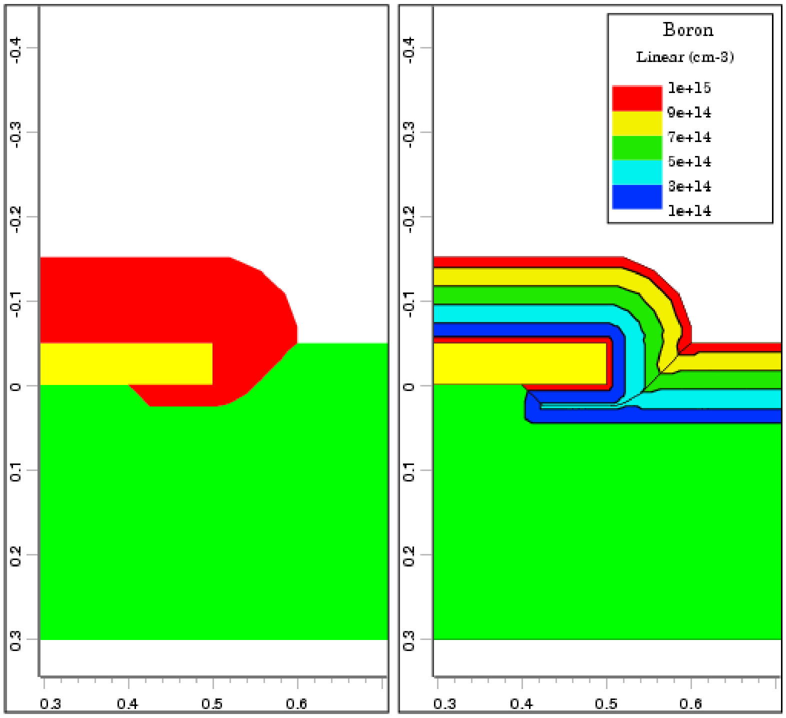


Figure F-10 ***Epitaxially Grown Polysilicon and Silicon (Linear gradient boron profile)***

Example 13 Same as Example 12 but with logarithmic gradient boron profile

```
LINE X LOC=0   SPAC=0.05
LINE X LOC=0.5 SPAC=0.02
LINE X LOC=1   SPAC=0.05
LINE Y LOC=0   SPAC=0.02
LINE Y LOC=0.3 SPAC=0.05

INIT
MATERIAL MAT=POLY ^POLYCR

DEPOSIT OXIDE THICK=0.05
ETCH MAT=OXIDE RIGHT P1.X=0.5
ETCH MAT=SILICON RIGHT P1.X=0.4 P2.Y=0.05

EPITAXY TIME=10 TEMP=600 THICK=0.1 IMP=BORON +
        IB.CONC=1E14 IT.CONC=1E15 SPACES=20 LOG.GRAD
```

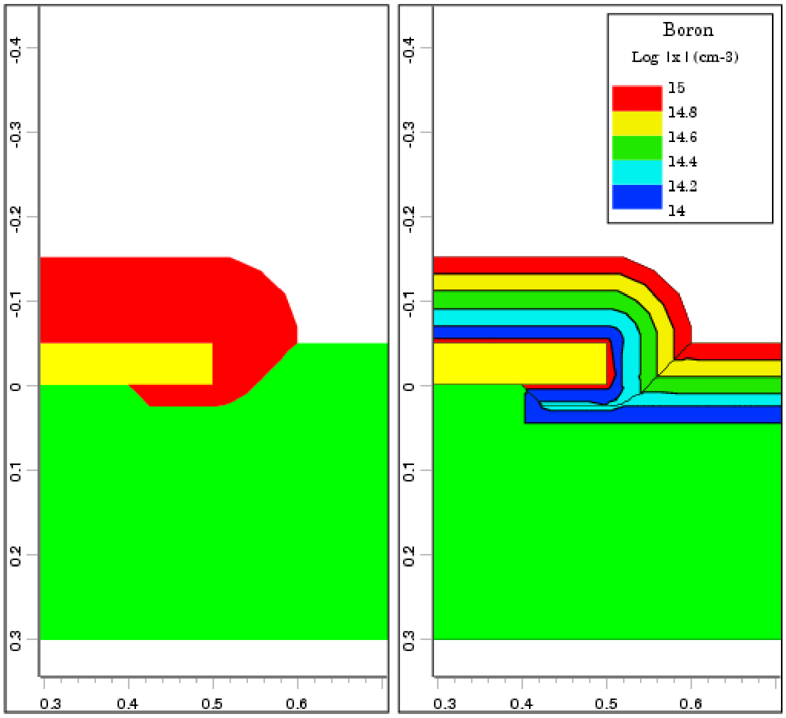


Figure F-11 *Epitaxially Grown Polysilicon and Silicon (Logarithmic gradient boron profile)*

Example 14 Same as Example 13 but use I.PROFIL

```
LINE X LOC=0   SPAC=0.05
LINE X LOC=0.5 SPAC=0.02
LINE X LOC=1   SPAC=0.05
LINE Y LOC=0   SPAC=0.02
LINE Y LOC=0.3 SPAC=0.05

INIT
MATERIAL MAT=POLY ^POLYCR

DEPOSIT OXIDE THICK=0.05
ETCH MAT=OXIDE RIGHT P1.X=0.5
ETCH MAT=SILICON RIGHT P1.X=0.4 P2.Y=0.05

EPITAXY TIME=10 TEMP=600 THICK=0.1 IMP=BORON +
        I.PROFIL=( 5E12*(5E13/5E12)^@{F} ) SPACES=20
```

Example 15 Deposition with Linear Gradient Multiple Species

```
LINE X LOC=0 SPAC=0.1
LINE X LOC=1 SPAC=0.1
LINE Y LOC=0 SPAC=0.1
LINE Y LOC=0.3 SPAC=0.1

INIT BORON=1E12

DEPOSIT POLY THICK=0.2 SPACES=10 TEMP=800 GSZ.LIN +
        IMP="BORON, PHOSPHORUS" +
        IB.CONC="5E13 1E15" +
        I.CONC="5E14 2E15"
```

Example 16 Epitaxy with Logarithmic Gradient Multiple Species

```
LINE X LOC=0 SPAC=0.1   TAG=L
LINE X LOC=0.5 SPAC=0.1 TAG=M
LINE X LOC=1 SPAC=0.1   TAG=R
LINE Y LOC=0 SPAC=0.1   TAG=T
LINE Y LOC=0.3 SPAC=0.1 TAG=B

REGION MAT=SILICON XLO=L XHI=M YLO=T YHI=B
REGION MAT=OXIDE   XLO=M XHI=R YLO=T YHI=B

INIT BORON=1E12

METHOD UNREFINE=0

EPITAXY THICK=0.2 SPACES=10 TIME=10 TEMP=600 +
        IMP="BORON, PHOSPHORUS" +
        IB.CONC="5E13 5E15" +
        I.CONC="5E14 2E15" LOG.GRAD
```

Strained Epitaxial Growth

Unlike in the **DEPOSIT** command, in the **EPITAXY** command turning on the new Boolean parameter **STRAINED** assigns the dopant-induced mismatch strain to the epitaxy layer. The mismatch strain calculation requires that **VISCOEL** and **ST.HIST** models are turned on in the **METHOD** command. Unlike the **DEPOSIT** command, which relaxes stress at the end of the process, the diffusion and stress equations are solved during epitaxial growth on the **EPITAXY** command. The final stress distribution from the **EPITAXY** command is different from that of the **DEPOSIT** command.

The new Boolean parameter **DIFFUSE** in the **EPITAXY** command specifies to skip solving diffusion and stress equations during epitaxial growth. The following example mimics the strained **DEPOSIT** command by using the **EPITAXY** command:

```
EPITAXY TEMPERAT=800 TIME=5 THICKNES=0.2 SPACES=5 STRAINED !DIFFUSE
STRESS RELAX
```

New Initialization Methods for Solutions

- You can define the expression for solution initialization using the **INITIAL** (alias: **DIFF.INI**) parameter in the **EQUATION** command. The **INITIAL** expression is always performed to initialize a solution at the end of the amorphous regrowth in the first diffusion following implant or at the beginning of an epitaxy step.
- Use the new parameter **IMPL.INI** in the **EQUATION** command to define the expression for the solution initialization at implant. The **IMPL.INI** expression is evaluated and applied at every implant step.
- In earlier versions, you could not initialize the point-defect solutions; this restriction has been removed.
- The new parameter **EXPRESS** has been added to the **PROFILE** command allows you to reset a solution whenever you want. For example:

```
PROFILE IMP=inters MAT=Si EXPR=1E22*exp(-y^2/0.1)
```

You can reset a solution within a specific box region by setting **X.MIN**, **X.MAX**, **Y.MIN** and **Y.MAX**. Furthermore, multiple solutions can be reset at once:

```
PROFILE IMP="Sxx, Syy, Szz" MAT=Si EXPR=1E10 +
        X.MIN=0.1 X.MAX=0.2 Y.MIN=0 Y.MAX=0.1
```

Note:

By default, the local equilibrium pairing (i.e. PD.FULL) and other equilibrium clustering models are internally reevaluated whenever a solution is initialized or reset. Turning off EQ.RESET in the PROFILE command skips such a re-evaluation.

Note:

When you initialize the free point-defect solutions, interstitial or vacancy, the initialized profile can differ from what you define because the equilibrium models (e.g., local equilibrium pairing in PD.FULL) must be re-evaluated, and so the free point-defect concentration must be recalculated.

Example 17 Interstitial Initialization After Regrowth

When **DIFF.INI** for interstitial is defined in the **EQUATION** statement, the "total" interstitial concentration is reset to the values of the expression evaluated after regrowth. And then the "free" interstitial is calculated and stored to the interstitial solution. The example shows two methods for initializing the interstitial concentration after regrowth using **EQUATION DIFF.INI** or using **PROFILE EXPRESS**.

The example adds the interstitial concentration by two times of implanted arsenic concentration.

First. the expression to describe the total interstitial concentration is described to the intermediate *ITOT*.

The *MYDAM* and *ASIMP* intermediates store **DAMAGE** and **DIMPLANT**, respectively. **DIMPLANT** stores the newly added dopant concentration by implantation. (For details, see in Fields to Store the Implant Dopant and Point-Defect Concentrations, p. F-52.) Reasons for using these types of intermediates instead of directly using **DAMAGE** and **DIMPLANT** will be explained in later sections.

```
$ DEFINE POINT-DEFECT MODEL AND MESH
METHOD PD.FULL
MESH DY.SURF=0.001 DY.ACT=0.01 LY.ACT=0.5

$ DEFINE PARAMETERS FOR AMORPHOUS/CRYSTALLINE REGIONS
ASSIGN   NAME=DAMGRAD  N.V=10.0
ASSIGN   NAME=MAXDAM   N.V=1.15E22
ASSIGN   NAME=DPLUS    N.V=2.0

$ DEFINE SMOOTHING FACTOR AT A/C BOUNDARY
INTERMED NAME=MYDAM    EXPR=DAMAGE
INTERMED NAME=ASIMP    EXPR=DIMPLANT
INTERMED NAME=EXPA     +
         EXPR=EXP(@DAMGRAD*LOG(MYDAM/@MAXDAM))
INTERMED NAME=EXPAM    EXPR=1/EXPA
INTERMED NAME=TANHA    EXPR=(EXPA-EXPAM)/(EXPA+EXPAM)
INTERMED NAME=FS       EXPR=0.5-0.5*TANHA
INTERMED NAME=ITOT     EXPR=CI.STAR+FS*(@DPLUS*ASIMP)
```

The intermediate *ITOT* is assigned to **DIFF.INI** for interstitial in the **EQUATION** statement. Then it is assigned to arsenic implantation and 800^{o}C zero time diffusion to see the interstitial profile after regrowth.

```
$ METHOD 1 : USE EQUATION DIFF.INI
EQUATION MODEL=IEQINI  IMP=INTERS MAT=SILICON DIFF.INI=ITOT

$ USE DIFF.INI FOR I INITIALIZATION
METHOD  MODEL=IEQINI ENABLE
INIT
IMPLANT ARSENIC DOSE=1E15 ENERGY=20
DIFFUSE TEMP=800 TIME=0

SELECT Z=LOG10(INTERS)
PLOT.1 X.V=0 TOP=21 BOT=15 RIGHT=0.5 COLOR=2  ;$ (1)
```

The other method to initialize interstitial concentration is to specify the **EXPRESS** parameter in the **PROFILE** command. The initialization is performed at that time the **PROFILE** command is given. Meanwhile the initialization by **DIFF.INI** in **EQUATION** is performed following any implantation at all the first diffusion steps.

When the **PROFILE** command following **DIFFUSE** is executed, the **DAMAGE** and **DIMPLANT** solutions no longer exist because they were removed at the first diffusion step following implantation. This causes the invalid solution error when the intermediate *ITOT*, which is the **EXPRESS** expression of the **PROFILE**, is evaluated. To avoid such an error, copy them to the storages, *IDAM* and *ASIMPL* by turning on **FIXNOW** before they are removed in the next diffusion step. (For details, see **FIXNOW** in INTERMED Snapshot, p. F-48.)

```
$ METHOD 2 : USE PROFILE EXPRESS
METHOD  MODEL=IEQINI !ENALE
INIT
IMPLANT ARSENIC DOSE=1E15 ENERGY=20
INTERMED NAME=IDAM FIXNOW
INTERMED NAME=ASIMPL FIXNOW
DIFFUSE TEMP=800 TIME=0
PROFILE IMP=INTERS MAT=SILICON EXPRESS=ITOT

SELECT Z=LOG10(INTERS)
PLOT.1 X.V=0 !CL !AX COLOR=3 LINE=2 ;$ (2)
```

To confirm that the results of the methods (1) and (2) are identical to the built-in calculation, the default simulation with **D.PLUS=2** is performed and compared to (1) and (2).

```
$ CONFIRM METHODS (1) and (2) IDENTICAL TO BUILT-IN
METHOD  MODEL=IEQINI !ENABLE
INIT
IMPLANT ARSENIC DOSE=1E15 ENERGY=20 D.PLUS=2.0
DIFFUSE TEMP=800 TIME=0

SELECT Z=LOG10(INTERS)
PLOT.1 X.V=0 !CL !AX COLOR=4 LINE=3 ;$ (3)
```

User-Defined Equilibrium Active Function

By setting an expression to the **ACTIVE** parameter in the **IMPURITY** command, you can change the function to calculate the dopant active concentration in transient state. In earlier versions, it also replaces the built-in function to calculate the equilibrium active concentration, which limits the USEIT application for describing the model such as A**CT.TRAN** or **C0** in **ACT.FULL**. Now, the expression of the **ACTIVE** parameter is used only for calculating the active concentration in transient state . The new parameter **EQACTIVE** in the **IMPURITY** command defines the expression to calculate the equilibrium active concentration. Calling the **EQACTIVE()** function returns the equilibrium active concentration. There are two types of **EQACTIVE()** function call:

- **EQACTIVE** (*imp,conc*)

  For the given 'conc' concentration, it calls the built-in function to calculate the equilibrium active concentration with the '*imp*' impurity parameters, and returns it.

- **EQACTIVE** (*imp*)

  It calls the function defined by the **EQACTIVE** parameter in the **IMPURITY** command if available. Otherwise, the built-in function is used.

Example 18 Transient clustering model of dopants

```
INTERMED NAME=C0EQ  EXPR=BORON-EQACTIVE(BORON)
INTERMED NAME=TDECL EXPR=8E-16*EXP(4.2/KT)
IMPURITY MODEL=BPREMODEL IMP=BORON MAT=SILICON +
         ACTIVE=BORON-C0_BORON +
         EQACTIVE=EQACTIVE(BORON,BORON)
EQUATION MODEL=BPREMODEL IMP=C0_BORON MAT=SILICON +
         ADDTOEXP=-(C0_BORON-C0EQ)/TDECL
```

INTERMED Snapshot

The expression of **INTERMED** is evaluated by call-by-reference. For example:

```
INTERMED NAME=A1 EXPR=10
INTERMED NAME=A2 EXPR=A1 PRINT ;$(1)
INTERMED NAME=A1 EXPR=20
INTERMED NAME=A2 PRINT         ;$(2)
```

At (1), A2 stores the memory address of A1 instead of the value of A1, 10. Thus, (2) shows that A2 value is 20 because the value of A1 is changed to 20.

The new parameter **FIXNOW** in the **INTERMED** command snapshots and fixes the current values. On the other hand, the new parameter **UNFIXNOW** unfixes the fixed **INTERMED**. For example:

```
INTERMED NAME=A1 EXPR=10

$ The following statement is same as:
$ INTERMED NAME=A2 EXPR=A1
$ INTERMED NAME=A2 FIXNOW
INTERMED NAME=A2 EXPR=A1 FIXNOW

INTERMED NAME=A1 EXPR=20
INTERMED NAME=A2 PRINT          ;$(1)
INTERMED NAME=A2 UNFIXNOW PRINT ;$(2)
```

The command (1) and (2) print 10 and 20, respectively. As shown in the following output of the commands (1) and (2), the name of a fixed intermediate is followed by the character '*':

```
INTERMED NAME=A2 PRINT
---------- ---------- --------- ------------------------------------- ----------
 Intermed     Scope     Type                   Expression                Value
---------- ---------- --------- ------------------------------------- ----------
A2*         .          CONSTANT  A1                                           10
---------- ---------- --------- ------------------------------------- ----------

INTERMED NAME=A2 UNFIXNOW PRINT
---------- ---------- --------- ------------------------------------- ----------
 Intermed     Scope     Type                   Expression                Value
---------- ---------- --------- ------------------------------------- ----------
A2          .          CONSTANT  A1                                           20
---------- ---------- --------- ------------------------------------- ----------
```

The following example shows how to compare as-implant and post-anneal profiles. The field name *DIMPLANT* stores the as-implant dopant concentration (For details, see Fields to Store the Implant Dopant and Point-Defect Concentrations, p. F-52). The *BIMPLANT* and *PIMPLANT* intermediates store the boron and phosphorus as-implant profiles, respectively:

```
INTERMED NAME=BIMPLANT EXPR=DIMPLANT
INTERMED NAME=PIMPLANT EXPR=DIMPLANT

IMPLANT BORON DOSE=1E15 ENERGY=10
$ Snapshot boron as-implant profile
INTERMED NAME=BIMPLANT FIXNOW

IMPLANT PHOSPHOR DOSE=2E14 ENERGY=30
$ Snapshot phosphorus as-implant profile
INTERMED NAME=PIMPLANT FIXNOW
DIFFUSE TEMP=900 TIME=10

$ PLOT COMPARISON BEFORE/AFTER DIFFUSION
SELECT Z=LOG10(BIMPLANT)
PLOT.1D X.V=0 TOP=22 BOT=15 RIGHT=0.5 COLOR=2
SELECT Z=LOG10(BORON)
PLOT.1D X.V=0 !CL !AX COLOR=2 LINE=2
SELECT Z=LOG10(PIMPLANT)
PLOT.1D X.V=0 !CL !AX COLOR=4 LINE=2
SELECT Z=LOG10(PHOSPHOR)
PLOT.1D X.V=0 !CL !AX COLOR=4 LINE=2
```

INTERMED Scope Definition

The new parameters **`SCOPE`** and **`ENDSCOPE`** in the **`INTERMED`** command specify the scope range within which the intermediates are applied. The scopes can be hierachically nested:

```
INTERMED NAME=A1 EXPR=10
INTERMED SCOPE=family
  INTERMED NAME=A1 EXPR=20
  INTERMED SCOPE=genus1
    INTERMED NAME=A1 EXPR=30
    INTERMED NAME=A1 PRINT  ;$(1)
  INTERMED ENDSCOPE
  INTERMED NAME=A1 PRINT    ;$(2)
  INTERMED SCOPE=genus2
    INTERMED NAME=A1 EXPR=35
  INTERMED ENDSCOPE
INTERMED ENDSCOPE
```

The (1) command prints 30 while the (2) prints 20.

The intermediates in the higher or same level scopes can be accessed from the lower level by specifying the intermediate name followed by the full path of scopes. ‘#’ is the character to separate the intermediate name and its scope path. The path string following ‘#’ always starts with ‘.’ which denote the top level. Then the scope names from the top level to the level in which the intermediate is defined are concatenated by the character ‘.’.

In the above example:

```
INTERMED NAME=A1 EXPR=10
INTERMED SCOPE=family
  INTERMED NAME=A1 EXPR=20
  INTERMED SCOPE=genus1
    INTERMED NAME=A1 EXPR=30
    INTERMED NAME=A1
    INTERMED NAME=A1#.       PRINT ;$(1)
    INTERMED NAME=A1#.family PRINT ;$(2)
  INTERMED ENDSCOPE
  INTERMED NAME=A1 PRINT
  INTERMED SCOPE=genus2
    INTERMED NAME=A1 EXPR=35
    INTERMED NAME=A1#.family.genus1 PRINT ;$(3)
  INTERMED ENDSCOPE
INTERMED ENDSCOPE
INTERMED PRINT ;$(4)
```

(1) prints A1 value in the top level, i.e. 10. (2) prints 20 in the scope ‘*family*’, one level lower than the top. (3) prints 30 which is defined in the scope ‘*genus1*’, the neighbor scope ‘*genus2*’. (4) prints all intermediates. The full path of each intermediate is written in the ‘Scope’ column. If the full path name is longer than

10 characters, it is truncated as shown in ‘*A1#.family.genus1*’ and ‘*A1#.family.genus2*’.

```
---------- ---------- --------- ------------------------------------- ----------
 Intermed     Scope     Type                 Expression                  Value
---------- ---------- --------- ------------------------------------- ----------
A1         .          CONSTANT  10                                            10
A1         .family    CONSTANT  20                                            20
A1         .family.ge CONSTANT  30                                            30
A1         .family.ge CONSTANT  35                                            35
---------- ---------- --------- ------------------------------------- ----------
```

The following example shows the case using the array type.

```
INTERMED NAME=R1 SIZE=2 EXPR=”10,20”
INTERMED SCOPE=lev1
  INTERMED NAME=R1 SIZE=3 EXPR=”30,40,R1#.[1]”
  INTERMED SCOPE=lev2a
    INTERMED NAME=R1 SIZE=2 EXPR=”R1#.[0], R1#.lev1[0]”
  INTERMED ENDSCOPE
  INTERMED SCOPE=lev2b
    INTERMED NAME=R1 SIZE=2 +
             EXPR=”R1#.lev1.lev2a[1], 50”
  INTERMED ENDSCOPE
INTERMED ENDSCOPE
INTERMED PRINT  ;$ (1)
```

The array intermediates in higher or neighbor scopes can be accessed by appending the full path to the array name. (1) prints as follows:

```
Array Name : R1,   Scope: .
---------- --------- ------------------------------------------------ ----------
  Index      Type                       Expression                       Value
---------- --------- ------------------------------------------------ ----------
[0]        CONSTANT  10                                                       10
[1]        CONSTANT  20                                                       20
---------- --------- ------------------------------------------------ ----------

Array Name : R1,   Scope: .lev1
---------- --------- ------------------------------------------------ ----------
  Index      Type                       Expression                       Value
---------- --------- ------------------------------------------------ ----------
[0]        CONSTANT  30                                                       30
[1]        CONSTANT  40                                                       40
[2]        CONSTANT  R1#.[1]                                                  20
---------- --------- ------------------------------------------------ ----------

Array Name : R1,   Scope: .lev1.lev2a
---------- --------- ------------------------------------------------ ----------
  Index      Type                       Expression                       Value
---------- --------- ------------------------------------------------ ----------
[0]        CONSTANT  R1#.[0]                                                  10
[1]        CONSTANT  R1#.lev1[0]                                              30
---------- --------- ------------------------------------------------ ----------

Array Name : R1,   Scope: .lev1.lev2b
---------- --------- ------------------------------------------------ ----------
  Index      Type                       Expression                       Value
---------- --------- ------------------------------------------------ ----------
[0]        CONSTANT  R1#.lev1.lev2a[1]                                        30
[1]        CONSTANT  50                                                       50
---------- --------- ------------------------------------------------ ----------
```

Fields to Store the Implant Dopant and Point-Defect Concentrations

To increase the flexibility for user-defined initialization or to remove the complexity of using **`PROFILE`** to avoid duplicated Monte Carlo implant, the implantation information–implanted dopant concentration, interstitial, and vacancies caused by implant damage–must be exposed to users.

Three new built-in fields (**`DIMPLANT`**, **`IIMPLANT`**, and **`VIMPLANT`**) have been added. **`DIMPLANT`**, **`IIMPLANT`**, and **`VIMPLANT`** store the dopant, interstitial and vacancy concentrations newly added by implant, respectively.

$$C_{\mathtt{IIMPLANT}} = I_{FP} + I_{plus} \qquad \text{Equation F-138}$$

where I_{FP} and I_{plus} are the shifted **`DAMAGE`** concentration by D**`X.FP`** and **`DY.FP`**, and the shifted interstitial profile from +n model by **`DX.IPLUS`** and **`DY.IPLUS`**, respectively.

$$C_{V\mathtt{IMPLANT}} = \text{DAMAGE} + V_{plus} \qquad \text{Equation F-139}$$

where V_{plus} is the shifted vacancy profile from +n model by **`DX.VPLUS`** and **`DY.VPLUS`**, respectively.

`DIMPLANT`, **`IIMPLANT`**, and **`VIMPLANT`** are lost after regrowth. To keep the fields after regrowth, copy the fields to the intermediates to be fixed.

Example 19 ***`INTERMED FIXNOW`***

```
MESH DY.SURF=0.002 LY.SURF=0.002 DY.ACT=0.01 LY.ACT=0.5
INIT BORON=1E16
INTERMED NAME=BIMP1 EXPR=DIMPLANT
INTERMED NAME=BIMP2 EXPR=DIMPLANT
IMPLANT BORON DOSE=1E15 ENERGY=10
INTERMED NAME=BIMP1 FIXNOW
IMPLANT BORON DOSE=2E14 ENERGY=20
INTERMED NAME=BIMP2 FIXNOW
DIFFUSE TEMP=900 TIME=20

$ BIMP1 : 1ST B AS-IMPLANT PROFILE
$ BIMP2 : 2ND B AS-IMPLANT PROFILE
SELECT Z=LOG10(BIMP1)
PLOT.1D X.V=0.0 TOP=22 BOT=15 RIGHT=0.5 COLOR=2 LINE=2
SELECT Z=LOG10(BIMP2)
PLOT.1D X.V=0.0 !CL !AX COLOR=3 LINE=2
SELECT Z=LOG10(BIMP1+BIMP2)
PLOT.1D X.V=0.0 !CL !AX COLOR=4 LINE=2
SELECT Z=LOG10(BORON)
PLOT.1D X.V=0.0 !CL !AX
```

Improved STRESS Command

`STRESS` command updates only the stress values by solving the stress equations while the values of the other solutions are kept. Thus, the temperature at which stress values are calculated; hereinafter, “stress temperature” must be distinguished from the diffusion temperature. However, during all thermal processes except the **`STRESS`** command, the stress temperature is always identical to the diffusion temperature.

```
DIFFUSE  TEMP=800 TIME=10            ;$(1)
STRESS   TEMP=800 T.FINAL=25 TIME=10 ;$(2)
SAVEFILE OUT.FILE=test.str           ;$(3)
DIFFUSE  TEMP=900 TIME=20            ;$(4)
```

After (1), the stress and diffusion temperatures are identical i.e. 800^{o}C. However after (2), the stress temperature is 25^{o}C, while the diffusion one is still 800^{o}C$^{\cdot}$ (3) command stores the stress values at 25^{o}C and the other solutions at 800^{o}C. The TIF format has been updated to store the stress temperature. (See diffusion temperature, 800^{o}C as well as the stress temperature 25^{o}C.) In this case, first, the stress temperature ramps up from the previous temperature 25^{o}C to the previous diffusion temperature 800^{o}C by solving only the stress equation. Then, all equations including stress equations are solved from the previous temperature 800^{o}C to the current starting temperature 900^{o}C.

The time step for such internal temperature ramp-up or ramp-down is given by the **`TIME.DEL`** parameter (default: 1E-25/60, units: minutes) in the **`METHOD`** statement. For example:

```
METHOD TIME.DEL=1E-9
```

To force the stress relaxation, use:

```
STRESS RELAX
```

The syntax and parameter description of the **`STRESS`** command is as follows:

```
STRESS  (  { [P.TOP=<n>] [P.SIDE=<n>] }
         | { [TEMPERAT=<n>] [TIME=<n>]
             [ {T.RATE=<n> | T.FINAL=<n>} ] [MOVIE=<c>] }
         | RELAX )
```

Taurus Monte Carlo Implant Updates

The following updates have been made to Taurus Monte Carlo:

- Added a parameter **SAT.PAR** (default: 1, unitless) in **IMPURITY** statement, which limits the maximum damage to the saturation value, i.e., (lattice density)**/SAT.PAR**. If saturation < **MAX.DAM** (amorphization threshold), the crystal cannot be amorphized regardless of the dose. This could happen in light species implants. The corresponding parameter for cascade damage model is **CASC.SAT**.
- Added a parameter **D.SIM** (default: 0.5, unitless) in the **MATERIAL** statement, which is the simultaneous collision distance in units of lattice constant. This parameter lets you fine tune the channeling tails.
- Added a parameter **V.RAN.D** (default: 0, units: um) in the **MATERIAL** statement. This parameter is used only in the cascade damage model. Since vacancies are created exactly at the lattice sites, net vacancy profiles usually are noisy. This parameter allows vacancies to be shifted by a random displacement (**RAND*V.RAN.D**) from lattice sites (where rand is a random number from 0 to 1), thus resulting in smoother net vacancy profiles.
- Added global switches **TRAJ.SP**L and **TRAJ.REP** in **METHOD** statement for trajectory splitting and trajectory replication, respectively. The default value for these switches is false. Added the interval of the replication segment **X.INTERV** in **METHOD** statement. These parameters can be overwritten by the corresponding switches in **IMPLANT** statement.
- Added switches for three different styles of trajectory splitting **DEPTH.SP**, **COLL.DEN**, and **TRAJ.DEN** in **METHOD** statement. These parameters control the splitting criterion which is based on the depth distribution of the dopant (**DEPTH.SP**), collision density (**COLL.DEN**), or trajectory density (**TRAJ.DEN**). The default value was **COLL.DEN** in earlier versions of TSUPREM-4, and is **DEPTH.SP** in this version of TSUPREM-4. The new default splitting model (**DEPTH.SP**) will produce smoother profiles at the expense of slightly increased CPU time than old default model (**COLL.DEN**).
- Improved the accuracy for aluminum implants into SiC.

TIF Format Change

The TIF format has been updated to store the stress temperature (see Improved STRESS Command, p. F-53) and the **FIXNOW** (INTERMED Snapshot, p. F-48) and **SCOPE** information(INTERMED Scope Definition, p. F-50) of intermediates.

- model

```
model  <orient> <13+n> <dtemp> <oxmod> <defmod> <rot>  <PRtype>
<actmod> <polymod#1> ... <polymod#n> <customod>  <clImod> <kbmod>
<stressmod> <itrapmod> <1> <stemp>
```

-

orient	Substrate orientation index
13+n	Total model index number
dtemp	Diffusion temperature
oxmod	Oxidation model index
defmod	Point-defect model index
rot	Substrate rotation angle
PRtype	Photoresist type index
actmod	Activation model index
polymod#1~n	Polycrystalline model parameter list
customod	Customer model index
clImod	Interstitial clustering model index
kbmod	Point-defect bulk recombination model index
stressmod	Stress model index
itrapmod	Interface trap model index
stemp	Stress temperature

- mo ts4 0

```
     mo ts4 0 <name> <order> <type+FIXMASK> <expression> <scope>
<fixed value>
```

-

name	Intermediate name.
order	Parsing order.
type	Constant(=2) or parameter(=3) or solution(=4). Solution type means the values are saved like a solution.
FIXMASK	4096 if the intermediate is fixed, otherwise, 0.
expression	Expression string (no space allowed in expression).
scope	The full path of the scope which the intermediate is defined.
fixed value	The value of the fixed constant or parameter intermediate. 0.0 for a solution type.

- mo ts4 1

```
   mo ts4 1 <name> <depth> <index_1> <len_1> ... <index_depth>
<len_depth> <scope>
```

-

name	Intermediate name.
depth	Array dimension.
index	Minimum (starting) index of each dimension.
length	Length (size) of each dimension.
scope	4096 if the intermediate is fixed, otherwise, 0.
scope	The full path of the scope which the intermediate is defined.

- mo ts4 2

```
    mo ts4 2 <name> <position> <order> <type+FIXMASK>
<expression> <scope> <fixed value>
```

name	Intermediate name.
position	The memory index location of this data.
order	Parsing order.
type	Constant(=2) or parameter(=3) or solution(=4). Solution type means the values are saved like a solution.
FIXMASK	4096 if the intermediate is fixed, otherwise, 0.
expression	Expression string (no space allowed in expression).
scope	The full path of the scope which the intermediate is defined.
fixed value	The value of the fixed constant or parameter intermediate. 0.0 for a solution type.

Miscellaneous

The following miscellaneous enhancements also were made:

- Turning on the parameter **`DDCPRINT`** in the **`IMPURITY`** command prints the **`DDC.FULL`** parameter values for the given impurity by the **`IMPURITY`** parameter.
- New Boolean parameter **`RELAX`** (default: true) has been added to **`DEPOSIT`** and **`ETCH`** commands. By turning it off, the stress relaxation is not performed.
- **`TOP`** and **`SIDE`** parameters in **`STRESS`** command are obsolete.
- Anisotropic elasticity information can be printed by the **`INITIALIZE`** command when either **`VERBOSE`** or **`DEBUG`** are turned on in the **`OPTION`** statement.

Example 20 Print out the Anisotropic Elastic Constant for Silicon

```
MATERIAL SILICON ANISOEL
OPTION   VERBOSE
INITIAL  <100> X.ORIENT=110 TEMP=800
OPTION   NORMAL
```

This example prints out the anisotropic elasticity tensor as follows:

```
*** silicon anisotropic elastic tensor at 300K (Crystallography) ***
   1.66e+12    6.33e+11    6.33e+11          0          0          0
   6.33e+11    1.66e+12    6.33e+11          0          0          0
   6.33e+11    6.33e+11    1.66e+12          0          0          0
          0           0           0   8.02e+11          0          0
          0           0           0          0   8.02e+11          0
          0           0           0          0          0   8.02e+11
*** silicon anisotropic elastic tensor at 300K (Simulation) ***
   1.95e+12    6.33e+11    3.43e+11          0  -2.11e-05          0
   6.33e+11    1.66e+12    6.33e+11          0  -8.05e-06          0
   3.43e+11    6.33e+11    1.95e+12          0  -2.11e-05          0
          0           0           0   8.02e+11          0  -1.17e-05
  -2.11e-05   -8.05e-06   -2.11e-05          0   5.12e+11          0
          0           0           0  -1.17e-05          0   8.02e+11
*** silicon anisotropic elastic tensor at 1073K (Simulation) ***
   1.83e+12    5.92e+11    3.17e+11          0   2.89e-05          0
   5.92e+11    1.55e+12    5.92e+11          0  -3.99e-06          0
   3.17e+11    5.92e+11    1.83e+12          0   2.89e-05          0
          0           0           0   7.55e+11          0  -2.98e-08
   2.89e-05   -3.99e-06    2.89e-05          0   4.81e+11          0
          0           0           0  -2.98e-08          0   7.55e+11
```

Glossary

This glossary contains terms frequently used in the *TSUPREM-4 User Guide.* A list of acronyms is included as the last section in the Glossary. For references to more information about a term, see the Index.

A

activation of dopants	Movement of dopant atoms onto substitutional sites on the silicon lattice where they are electrically active.
adaptive gridding	Automatic adaptation of the simulation mesh by the program to improve solution accuracy.
amorphization	The complete disruption of the silicon lattice structure caused by ion implantation. The amorphized region will recrystallize (through epitaxial regrowth) during the next high-temperature step, but residual damage may remain, especially at the edges of the amorphized layer.
analytical oxidation model	A model that uses empirical descriptions of the 2-D oxide shape to predict the results of local oxidation steps.
automatic mesh generation	Generation of the simulation mesh automatically by the program based on general guidelines supplied by the user.

C

channeling	Tendency of ions to travel along open directions in the silicon lattice during implantation. Channeling may cause implanted profiles to be deeper than expected, depending on the tilt and rotation of the wafer related to the ion beam. The effect is included in the dual Pearson implant tables and in the Monte Carlo implant model.
clustering	The formation of groups of atoms whose properties are different from the properties of the individual atoms within the group. Clustering usually refers to the clustering of dopant atoms to form immobile and electrically inactive complexes. (See interstitial clustering.)

D

deposition	One of the basic fabrication steps used for IC processing. TSUPREM-4 contains a simple model for deposition of conformal layers. If more advanced models are needed, TSUPREM-4 can be used with Taurus Topography.
diffusion	Spreading of impurities due to random motion at high temperatures. The term also refers to any process step in which diffusion occurs. TSUPREM-4 includes detailed physical models for diffusion and other processes, such as oxidation and silicidation, that occur during high temperature fabrication steps.
damage annealing	Removal of implantation damage with high temperatures.
dislocation loops	A class of extended defect in the silicon lattice structure. The ED-AAM models end-of-range dislocation loops produced by amorphizing ion implants.
dopant	An impurity added to a structure to control its electrical properties.

E

epitaxial growth	Growth of single crystal silicon on top of an existing silicon structure. TSUPREM-4 includes a model for epitaxial growth on planar substrates.
epitaxial regrowth	Recrystallization of a silicon region amorphized during ion implantation.

G

grid	(See mesh.)

I

implant damage	Damage to the silicon lattice structure caused by ion implantation.
impurity	Any atom other than silicon in a silicon structure. Often used to refer to a dopant.
interstitial	A type of point defect consisting of an extra silicon atom not on the lattice site. Interstitials promote diffusion by pairing with dopants.
interstitial clustering	Clustering of interstitials. Clustered interstitials are assumed to be immobile and do not promote the diffusion of impurities. In TSUPREM-4, the clustering model is used to simulate the effects of {311} defects.
interstitial trap model	A model describing the absorption and release of interstitials at a trapping site.
ion pairing	Coupling of dopant atoms of opposite charge to form an immobile pair.
ion implantation	Introduction of dopants into a wafer by a beam of accelerated ions.

L

local oxidation	Oxidation of a portion of the silicon surface, achieved by masking the portion that is not oxidized with a material such as silicon nitride.

M

mesh	The array of points within the structure at which solution values are numerically calculated. (The term is used interchangeably with *grid*.)
Monte Carlo implant model	A physics-based model that uses statistical techniques to calculate impurity and damage profiles produced by ion implantation.

N

numerical oxidation model	A model that uses the numerical solution of the physical equations for oxide growth and flow to predict the results of arbitrary oxidation steps.

O

oxidation	Creation of silicon dioxide by reaction of silicon or polysilicon with oxygen. Oxidation is modeled in TSUPREM-4 whenever diffusion occurs in an oxidizing ambient.
point defects	A localized defect (as opposed to an extended defect) in a silicon lattice structure. In TSUPREM-4, the term *point defects* refers to interstitials (i.e. silicon self-interstitials) and vacancies.
polycrystalline model	A model for polycrystalline materials used primarily for modeling polycrystalline silicon. It includes models for grain growth and the effect of grain boundaries on dopant diffusion.

P

PMEI	Physical model and equation interface. PMEI is intended to be an universal language for scripting models and equations for semiconductor process and device simulation.

S

segregation	Transport of an impurity across an interface between two materials due to a difference in chemical potential. This transport results in an equilibrium ratio of impurity concentration across a material boundary.
silicidation	Interaction of silicon to form a metal silicide. Modeled in TSUPREM-4 by the `DIFFUSION` statement whenever necessary materials are present in the structure.
solid solubility	The concentration at which dopant atoms precipitate, becoming immobile and electrically active.
stress	TSUPREM-4 includes models for calculating the physical stresses produced during fabrication. The stress history model simulates the stresses produced by oxidation, thermal mismatch between materials, intrinsic strain in deposited layers, and surface tension.

T

TSUPREM-4	The name originated from a combination of TMA (now the TCAD Business Unit of *Synopsys*) and SUPREM-IV, the Stanford University Process Engineering Models.
TED	Transient-Enhanced Diffusion. Enhanced diffusion caused by implant damage.

U

USEIT	User-specified equation interface is an optionally available capability for TSUPREM-4 that helps to quickly develop and evaluate process models.

Acronyms

LDD	Lightly Doped Drain
MOS	Metal-Oxide Semiconductor
NMOS	N-channel Metal-Oxide Semiconductor
OED	Oxidation-Enhanced Diffusion
PMEI	Physical Model and Equation Interface
TED	Transient-Enhanced Diffusion
USEIT	User-Specified Equation Interface

Index

A

absorption of interstitial traps 3-66
activation of dopants 3-37
ACTIVE 7-9
active region simulation
    LDD implant 6-8
    modeling polysilicon 6-7
    oxide spacer and source/drain implant 6-9
    plots 6-11
    source/drain contacts 6-9
adaptive grid parameters, default coefficients A-28
adaptive gridding
    advanced examples of 6-4, 6-29
    analytical implant models 4-224
    disable 4-224
    overview 3-10
    tutorial example of 5-4
adding expressions to equations 7-6
advanced examples
    CMOS process 6-32
    DMOS power transistor 6-42
    MOSFET with self-aligned silicides 6-54
    NMOS LDD process 6-2
    poly-buffered LOCOS 6-28
    polysilicon emitter study 6-59
    SOI MOSFET 6-49
    trench implant simulation 6-18
advantages of intermediates 7-8
AMBIENT 4-233
    additional notes 4-250
    chlorine 4-248
    coefficients 4-248
    coefficients chlorine examples 4-248
    COMPRESS model 4-246
    description 4-245
    ERFC model 4-245
    ERFG model 4-245
    examples 4-249
    INITIAL parameter, 1D bipolar example 5-5
    orientation 4-249
    oxidation models 4-245
    oxidation models, polycrystalline silicon 4-245
    oxidizing species 4-249
    parameter dependencies 4-249
    specified material 4-249
    specified units 4-249
    stress dependence 4-246
    VERTICAL model 4-246
    VISCOELA model 4-246
    VISCOUS model 4-246
ambient gas
    chlorine example 3-17
    chlorine, coefficient tables 3-17
    defaults 3-16
    oxidation of materials 3-16
    pressure 3-15
    specifying characteristics 3-15
amorphization, damage dechanneling 3-143
amorphous implant calculation 3-137
    electronic stopping 3-138
    electronic stopping at high energies 3-139
    ion beam width 3-140
    nuclear stopping 3-138
    total energy loss and ion deflection 3-139
analytic damage model 3-129
    damage distribution calculations 3-129
    recommended usage and limitations 3-130
analytic ion implant models 3-119
    analytic damage model 3-129
    depth-dependent lateral distribution 3-128
    dose-dependent implant profiles 3-124
    dual Pearson distribution 3-124
    Gaussian distribution 3-123
    implant moment tables 3-121
    implanted impurity distributions 3-120
    lateral distribution 3-128

- multilayer implants 3-127
- Pearson distribution 3-123
- tilt and rotation tables 3-127
- wafer tilt and rotation 3-129

analytical oxidation models 3-87
- **ERF1** model 3-91
- **ERF2** model 3-92
- **ERFC** model 3-90
- **ERFG** model 3-92
- linear rate 3-87
- overview 3-87
- oxide growth rate 3-87
- parabolic rate 3-88
- thin regime 3-87
- usage 3-89

annealing
- damage 3-143
- *in situ* 3-143
- self-annealing 3-143

ANTIMONY 4-337
- additional notes 4-343
- description 4-342
- examples 4-343

application of user-specified models 7-8

Arora mobility model 3-174

array values
- effect due to redefinition 4-36
- getting array values 4-35
- redefinition of array elements 4-35

ARSENIC 4-344
- additional notes 4-350
- description 4-349
- examples 4-350

ARSENIC parameter, **EPITAXY** statement
- 1D bipolar example 5-6

ASSIGN 4-23
- additional **ASSIGN** notes 4-28
- ASSIGN and optimization 4-28
- ASSIGN with mathematical expressions 4-27
- assign with mathematical expressions, using **N.EXPRESS** 4-28
- description 4-26
- expansion of **ASSIGN**ed variable 4-28
- overriding variables 4-29
- reading the external data file 4-29
- varying during statement looping 4-27

authorization file *s4auth* 2-10

automatic grid generation
- custom defaults 3-7
- overview 3-5
- parameters, default coefficients A-27
- tutorial examples of 5-4, 5-20

AXES parameter, **PLOT.1D** statement
- 1D bipolar example 5-7

AXES parameter, **PLOT.2D** statement
- local oxidation example 5-19

B

BF2 implant 3-129

BF2 implant profile *Fig.* 3-126

binary scattering theory 3-133
- Coulomb potential 3-135
- dimensionless form 3-134
- energy loss 3-134
- scattering angle 3-134
- universal potential 3-136

bipolar emitter structure and as-implanted arsenic profiles, as plotted using *s4ex11c.inp Fig.* 6-61

Boltzmann statistics 3-171

BORON 4-351
- additional notes 4-357
- description 4-356
- examples 4-357

boron diffusion
- enhancement due to fluorine 3-109
- enhancement in oxides 3-106
- enhancement in thin oxides 3-107

BORON parameter, **INITIALIZE** statement
- 1D local oxidation example 5-20

boron-interstitial clustering model with small clusters 7-21

BOTTOM parameter, **PLOT.1D** statement
- local oxidation example 5-25

BOUNDARY 4-62
- description 4-62
- example 4-63
- limitations 4-63

boundary conditions, ion implantation 3-149

built-in keywords 7-10

C

C.COMPRE parameter, **PLOT.2D** statement
- local oxidation example 5-17

C.GRID parameter, **PLOT.2D** statement
- local oxidation example 5-15

C.TENSIO parameter, **PLOT.1D** statement

  local oxidation example 5-17
capacitance calculation 3-176
  DC method 3-176
  MOS capacitances 3-177
Caughey mobility model 3-175
CENTER parameter, **LABEL** statement
  CMOS process example 6-39
channel and source/drain doping profiles for SOI MOSFET *Fig.* 6-54
channel doping profile for NMOS transistor *Fig.* 6-37
Channeling 4-126
channeling 3-140
Chapter 6 7-1
charge state fractions 3-60
chemical predeposition 3-17
CLEAR parameter, **PLOT.1D** statement
  1D bipolar example 5-7
CLEAR parameter, **PLOT.2D** statement
  local oxidation example 5-27
clustering and solid solubility, default coefficients A-7
clustering model 3-37
CM parameter, **LABEL** statement
  1D bipolar example 5-8
CMOS process example
  0.8 micron device 6-39
  1.2 micron device 6-40
  arsenic profiles in gate 6-39
  channel doping plot 6-36
  COMPRESS model 6-36
  contacts 6-36
  end of main loop 6-37
  END statement 6-37
  final mesh 6-39
  FOREACH loop 6-32, 6-37
  FOREACH statement 6-36
  INITIALIZE statement 6-37
  lightly doped drain structure 6-36
  main loop 6-34
  mesh generation 6-34
  models 6-36
  overview 6-32
  PD.TRANS model 6-36
  plotting results 6-37
  processing 6-36
  SAVEFILE statement 6-37
  saving the structure 6-37
  VERTICAL oxidation model 6-36
coefficient default values A-1
COLOR 4-170
  description 4-171
  examples 4-171
  plot device selection 4-171
COLOR parameter, **PLOT.1D** statement
  1D bipolar example 5-7
command input files 2-4
COMMENT 4-7
  description 4-7
  examples 4-7
  notes 4-7
complete NMOS structure, plotted by input file *s4ex4c.inp Fig.* 6-14
completing the active region simulation 5-9
  field oxidation 5-11
  reading a saved structure 5-11
COMPRESS model 3-96
  boundary conditions 3-97
  compressible viscous flow 3-96
  material parameters 3-97
  recommended usage 3-97
compressible viscous flow 3-96
concentration dependence 3-94
concentration of point defects
  vs. depth, as plotted by input file *s4ex2b.inp Fig.* 5-25
  vs. width, as plotted by input file *s4ex2b.inp Fig.* 5-26
CONTOUR 4-168
  additional notes 4-169
  description 4-169
  example 4-169
  line type and color 4-169
contours of boron
  after analytic implant *Fig.* 6-21
  after Monte Carlo implant *Fig.* 6-26
  concentration produced by input file *s4ex2b.inp Fig.* 5-24
contours of hydrostatic pressure
  final poly-buffered LOCOS structure *Fig.* 6-31
  plotted by statements in input file *s4ex2a.inp Fig.* 5-19
contours of interstitial concentration, as plotted by input file *s4ex2b.inp Fig.* 5-27
contours of net doping for 1-micron and 2-micron emitter stripes *Fig.* 6-63
contours of total arsenic concentration and poly grain size after RTA *Fig.* 6-63
conventions
  typographical 1-iv
CPULOG 4-50

  - description 4-50
  - examples 4-50
  - limitations 4-50
- creating the initial structure
  - adaptive gridding 6-4
  - masking information 6-4
  - setting the grid density 6-3
- creating the test structure
  - automatic grid generation 5-31
  - outline of example 5-31
- **CRYSTAL** parameter, **IMPLANT** statement
  - trench implant simulation example 6-24, 6-26
- crystalline implant model 3-140
  - BF2 implantation 3-144
  - channeling 3-140
  - damage annealing 3-143
  - damage dechanneling 3-143
  - lattice damage 3-142
  - lattice temperature 3-142
  - number of ions 3-143

D

- damage annealing 3-143
- damage dechanneling
  - amorphization 3-143
  - Monte Carlo implant 3-143
- DC method 3-176
- **DEEP** parameter, **ELECTRICAL** statement
  - NMOS LDD process example 6-16
- default coefficient notes A-34
- default coefficient tables
  - adaptive grid parameters A-28
  - automatic grid generation parameters A-27
  - clustering and solid solubility A-7
  - electrical parameters A-21
  - impurity parameters A-2
  - material coefficients A-23
  - Monte Carlo implant A-26
  - notes A-34
  - numerical methods A-26
  - oxidation A-11
  - point defect parameters A-8
  - polysilicon grain segregation A-7
  - references A-29
- default file names 2-3
- **DEFINE** 4-46
  - description 4-46
  - examples 4-46
  - format and syntax 4-46
  - usage notes 4-47
- **DEPOSITION** 4-99
  - additional notes 4-105
  - description 4-102
  - examples 4-104
  - photoresist 4-103
  - polycrystalline materials 4-103
  - structure bottom 4-102
- deposition
  - incorporation of impurities 3-155, 3-156
  - layer thickness 3-155
  - overview 3-155
  - photoresist type 3-156
- deposition parameters 4-104
- **DEVELOP** 4-108, 4-109
  - description 4-108
  - example 4-108
- **DEVICE** parameter, **OPTION** statement
  - 1D bipolar example 5-7
- device structure specification, input statements
  - **BOUNDARY** 4-62
  - **ELECTRODE** 4-96
  - **ELIMINATE** 4-59
  - **INITIALIZE** 4-66
  - **LINE** 4-57
  - **LOADFILE** 4-71
  - **MASK** 4-84
  - **MESH** 4-53
  - **PROFILE** 4-86
  - **REGION** 4-64
  - **SAVEFILE** 4-74
  - **STRUCTURE** 4-79
- diagnostic output file *s4dia* 2-6
- **DIFFUSION** 4-131
  - ambient gas 4-134
  - ambient gas parameters 4-135
  - description 4-134
  - DRY02 parameter, 1D bipolar example 5-6
  - examples 4-136
  - oxidation limitations 4-135
  - reflow 4-136
  - STEAM parameter, 1D bipolar example 5-5
  - THICKNES parameter, 1D bipolar example 5-6
- diffusion
  - activation of impurities 3-34
  - along grain boundaries 3-163
  - anisotropic 3-163
  - clustering model 3-37
  - diffusion of impurities 3-18

  DIFFUSION statement 3-15
  grain boundary structure 3-162
  grain interiors 3-162
  impurities 3-154
  impurity fluxes 3-18
  injection and recombination of point defects at interfaces 3-66
  interstitial traps 3-70
  overview 3-14
  point defects 3-58, 3-59
  segregation of impurities 3-54
  solution of equations 3-17
  temperature 3-15
  tutorial example of 5-20
diffusion along boundaries 7-7
diffusion enhancement in thinner oxides 7-25
diffusion in **PD.TRANS** 7-19
diffusion of point defects
  absorption of interstitial traps 3-66
  charge state fractions 3-60
  equations 3-61
  equilibrium concentrations 3-59
  interstitial and vacancy diffusivities 3-63
  net recombination rate of interstitials 3-64
  reaction of pairs with point defects 3-64
diffusivities 3-22
dislocation loop model 3-82, 7-17
  creation of 3-83
  effects of 3-84
DMOS power transistor example
  after p-type body diffusion *Fig.* 6-46
  COMPRESS oxidation model 6-43
  mesh generation 6-43
  overview 6-42
  PD.TRANS point defect model 6-48
  processing DMOS structure 6-43
  summary 6-48
documentation and control, input statements
  ASSIGN 4-23
  COMMENT 4-7
  CPULOG 4-50
  DEFINE 4-46
  ECHO 4-40
  FOREACH/END 4-14
  HELP 4-51
  IF/ELSEIF/ELSE/IF.END 4-21
  INTERACTIVE 4-11
  INTERMEDIATE 4-30
  L.MODIFY 4-20
  LOOP/L.END 4-16
  OPTION 4-41
  PAUSE 4-12
  RETURN 4-9
  SOURCE 4-8
  STOP 4-13
  UNDEFINE 4-49
dopants
  activation of 3-37
  advanced tutorial examples of 6-21, 6-24
  diffusion of atoms equations 3-18
  dislocation loop model 3-82
  paired fractions 3-25
  polycrystalline materials 3-161
  solid solubility 3-34
  transient clustering model 3-37
DOSE parameter, **IMPLANT** statement
  trench implant simulation example 6-23
DRYO2 parameter, **DIFFUSION** statement
  1D bipolar example 5-6
dual Pearson distribution 3-124
DY parameter, **DEPOSITION** statement
  SOI MOSFET example 6-50
DY.OXIDE parameter, **METHOD** statement
  local oxidation example 5-15
DY.SURF parameter, **MESH** statement
  CMOS process example 6-34

E

E.RVCAP parameter, **ELECTRICAL** statement
  NMOS LDD process example 6-16
ECHO 4-40
  description 4-40
  examples 4-40
effective range model 3-127
electric field 3-20
ELECTRICAL 4-194
  additional notes 4-203
  description 4-199
  E.RVCAP parameter, NMOS LDD process example 6-16
  examples 4-199
  files and plotting 4-199
  JUNCTION parameter, NMOS LDD process example 6-16
  MOSCAP parameter, NMOS LDD process example 6-14, 6-15
  optimization examples 4-201
  quantum effect in CV plot 4-201

    - RESISTAN parameter, NMOS LDD process example 6-14
    - THRESHOLD parameter, NMOS LDD process example 6-14
- electrical calculations
    - Arora mobility model 3-174
    - Boltzmann statistics 3-171
    - carrier mobility 3-173
    - Caughey mobility model 3-175
    - Fermi-Dirac statistics 3-171
    - ionization of impurities 3-172
    - overview 3-170
    - Poisson’s equation 3-171
    - solution methods 3-173
    - tabular form 3-174
- electrical characteristics, plotted by input file *s4ex4d.inp Fig.* 6-17
- electrical data output files 2-8
- electrical extraction
    - MOS capacitance 6-15
    - plotting results of electrical extraction 6-17
    - source/drain junction capacitance 6-16
    - threshold voltage 6-14
- electrical parameters, default coefficients A-21
- **ELECTRODE** 4-96
    - additional notes 4-97
    - description 4-96
    - examples 4-97
- **ELIMINATE** 4-59
    - description 4-59
    - examples 4-60
    - initial structure generation 4-60
    - overlapping regions 4-60
    - reducing grid nodes 4-60
- enable/disable user-specified models 4-223
- **ENERGY** parameter, **IMPLANT** statement
    - trench implant simulation example 6-23
- environment variables 2-4
- epitaxial growth
    - layer thickness 3-154
    - modeling 3-153
    - overview 3-153
- epitaxial regrowth 3-167
- **EPITAXY** 4-137
    - ARSENIC parameter, 1D bipolar example 5-6
    - description 4-139
    - example 4-141
    - mobility tables 4-141
    - SPACES parameter, 1D bipolar example 5-6
    - TEMPERAT parameter, 1D bipolar example 5-6
    - TIME parameter, 1D bipolar example 5-6
- epitaxy
    - selective 3-154
- **EPITAXY** parameter, **TEMPERAT** statement
    - 1D bipolar example 5-6
- **EQUATION** 4-230
    - adding expressions to equations 7-6
    - diffusion along boundaries 7-7
    - flux at interfaces 7-6
    - initialization 7-4
    - new solution variable 7-3
    - solving method 7-3
    - update solution 7-7
- equilibrium concentrations 3-59
- **ERF1** model 3-91
    - initial structure 3-91
    - parameters 3-90
- **ERF2** model 3-92
    - initial structure 3-91
    - parameters 3-90
- **ERFC** model 3-90
- **ERFG** model 3-92
    - initial structure 3-91
    - parameters 3-90
    - recommended usage 3-92
- **ERR.FAC** parameter, **MESH** statement
    - NMOS LDD process example 6-4
- **ERR.FAC** parameter, **METHOD** statement
    - 1D bipolar example 5-4
- error messages 2-3
- **ETCH**
    - description 4-111
    - examples 4-113
    - generate simulation structure 4-112
    - removing regions 4-111
- etching
    - complex structures 3-161
    - defining the region 3-157
    - examples 3-159
    - material removal 3-158
    - overview 3-157
    - simple structure 3-159
    - structure with overhangs 3-160
    - trapezoidal model 3-158
    - tutorial example 5-5
- examples
    - mask data file *Fig.* C-2
    - sensitivity plot for target with multiple data *Fig.* 4-19

trapezoidal etch model *Fig.* 3-160

EXPOSE 4-106

description 4-106

OFFSET parameter 4-106

extended defects AAM

dislocation loop model 3-82

transient clustering model 3-37

EXTRACT 4-180

description 4-185

error calculation 4-188

examples 4-189

extraction procedure 4-186

file formats 4-188

optimization examples 4-191

solution variables 4-186

targets for optimization 4-188

extract output files 2-8

F

Fermi-Dirac statistics 3-171

field isolation simulation

displaying the plot 6-5

file specification 2-3

files

command input 2-4

default names 2-3

environment variables 2-4

input 2-4

library 2-8

mask data 2-5

profile 2-5

terminal output 2-6

types 2-3

final grid

LDD NMOS example, produced by input files *s4ex4a.inp* and *s4ex4b.inp Fig.* 6-11

SOI MOSFET *Fig.* 6-53

final mesh

0.8 micron NMOS structure *Fig.* 6-39

DMOS simulation, showing the result of adaptive gridding *Fig.* 6-48

final profiles produced by input file *s4ex1b.inp Fig.* 5-12

final structure

0.8 micron, NMOS structure, plotted by *s4ex7b.inp Fig.* 6-39

1.2 micron, NMOS, plotted by input file *s4ex7c.inp Fig.* 6-41

after removal of remaining titanium *Fig.* 6-58

DMOS power transistor, produced by input file *s4ex8.inp Fig.* 6-48

NMOS plotted by file *s4ex4b.inp Fig.* 6-12

showing contours of net doping for SOI MOSFET *Fig.* 6-53

fine grids, final check 6-1

first part of input file

s4ex10.inp, NMOS transistor processing *Fig.* 6-55

s4ex11c.inp, for plotting structure and contours of as-implanted arsenic concentration *Fig.* 6-62, 6-66, 6-68, 6-69, 6-73, 6-75

s4ex2a.inp, for determining LOCOS shape *Fig.* 5-13

s4ex2b.inp, showing processing steps *Fig.* 5-21

s4ex3.inp, showing processing and plotting using **PD.FERMI** point defect model *Fig.* 5-29

s4ex4a.inp, setting up grid for simulating NMOS process *Fig.* 6-3

s4ex4b.inp, showing polysilicon gate formation *Fig.* 6-8

s4ex4d.inp, showing threshold voltage extraction *Fig.* 6-15

s4ex5.inp, showing grid setup *Fig.* 6-18

s4ex6.inp, Poly-buffered LOCOS process *Fig.* 6-29

s4ex7a.inp, to set up grid for simulating CMOS process *Fig.* 6-32

FLOW parameter, **PLOT.2D** statement

local oxidation example 5-16

flux at interfaces 7-6

flux at interfaces *Fig.* 7-7

FOREACH/END 4-14

description 4-14

examples 4-14

notes 4-15

format

input statements 4-2

mask data files C-1, D-1

statement description 4-4

G

gas

see **DIFFUSION** statement, ambient gas

Gaussian distribution 3-123

grain size model 3-165

concentration dependence 3-166

grain growth 3-166

grain surface energy 3-167

segregation drag 3-167

graphical output files 2-7
graphics devices
  default device B-7
  determining which to use B-1
  modifying *s4pcap* B-7
  output B-2
  supported devices B-2
grid
  after formation of isolation region, plotted by *s4ex4a.inp Fig.* 6-6
  plot produced by first **PLOT.2D** statement input file *s4ex2b.inp Fig.* 5-22
  poly-buffered LOCOS application *Fig.* 6-30
  trench implant example *Fig.* 6-19
grid generation parameters, automatic, default coefficients A-27
grid lines
  automatic generation 3-5
  eliminating 3-5
  generated 3-4
  specifying 3-4
GRID parameter, **PLOT.2D** statement
  local oxidation example 5-15, 5-22
grid parameters, adaptive, default coefficients A-28
grid structure
  1D simulation of simple structures 3-13
  adaptive gridding 3-10
  automatic generation 3-5
  changes to mesh during processing 3-7
  defining 3-3
  explicit specification 3-3
  generated grid lines 3-4
  initial impurity concentration 3-13
  initial impurity concentration, sheet resistance 3-14
  LINE statement 3-4
  mesh 3-3
  nodes 3-3
  overview 3-3
  triangular elements 3-3
GRID.FAC parameter, **MESH** statement
  CMOS process example 6-34
  DMOS power transistor example 6-43
  local oxidation example 5-14
  NMOS DD process example 6-3

H

H.RECTAN parameter, **LABEL** statement
  local oxidation example 5-19
HELP 4-51
  description 4-51
  example 4-51
  notes 4-51
HIGH parameter, **ELECTRICAL** statement
  NMOS LDD process example 6-16

I

IF/ELSEIF/ELSE/IF.END 4-21
  conditional operators 4-22
  description 4-21
  expression for condition 4-22
IMPLANT 4-114
  boundary conditions 4-126
  channeling effects 4-127
  CRYSTAL parameter, trench implant simulation example 6-24
  description 4-123
  DOSE parameter, trench implant simulation example 6-23
  ENERGY parameter, trench implant simulation example 6-23
  examples 4-127
  extended defects 4-126
  Gaussian and Pearson distributions 4-124
  Monte Carlo implant model 4-125
  MONTECAR parameter, trench implant simulation example 6-20
  N.ION parameter, trench implant simulation example 6-24
  point defect generation 4-125
  table of range statistics 4-124
  TILT parameter, trench implant simulation example 6-20
implant damage model 3-144
  using 3-148
implant, Monte Carlo, default coefficients A-26
impurities
  activation of 3-34
  clustering of 3-34
  diffusion 3-154
  diffusion of 3-18
  implantation 3-57
  incorporation 3-154
  interface trap model 3-55
  ionization of 3-172
  moving-boundary flux 3-55
  segregation coefficient 3-55

  - segregation flux 3-54
  - segregation of 3-54
  - specifying 4-2
  - transport coefficient 3-55
- **`IMPURITY`** 4-273
  - description 4-296
  - examples 4-298
  - further reading 4-298
  - impurity type 4-297
  - multiplication to diffusivity 4-297
  - other parameters 4-297
  - solution options 4-297
- impurity distributions in bipolar structure at end input file *s4ex1a.inp Fig.* 5-8
- impurity parameters, default coefficients A-2
- **`IN.FILE`** parameter, **`INITIALIZE`** statement
  - 1D bipolar example 5-11
- incorporation of impurities 3-154
- informational output file *s4inf* 2-6
- initial conditions 3-116
- initial grid
  - accuracy 6-1
  - for simulating DMOS power transistor *Fig.* 6-43
  - for simulating SOI MOSFET *Fig.* 6-50
  - for the 0.8 micron NMOS transistor, produced by *s4ex7a.inp Fig.* 6-35
- **`INITIAL`** parameter, **`AMBIENT`** statement
  - 1D bipolar example 5-5
- initialization 7-4
- initialization input file *s4init* 2-9
- **`INITIALIZE`** 4-66
  - crystalline orientation 4-69
  - description 4-68
  - examples 4-70
  - IN.FILE parameter, 1D bipolar example 5-11
  - Masetti’s mobility table 4-65, 4-70
  - mesh generation 4-69
  - previously saved structure files 4-69
  - specifying initial doping 4-65, 4-69, 4-70, 4-72
- injection rate 3-68
- input files 2-4
  - command 2-4
  - mask data 2-5
  - other 2-5
  - profile 2-5
  - *s4ex1a.inp*, simulating buried layer and epitaxial deposition, bipolar transistor structure *Fig.* 5-3
  - *s4ex4c.inp*, plotting final LDD NMOS structure *Fig.* 6-13
  - *s4ex7b.inp*, plotting results *Fig.* 6-38
  - *s4init* 2-9
  - *se4ex10p.inp*, plotting results *Fig.* 6-56
- input statement descriptions
  - format 4-2
  - parameter types 4-3
  - specifying materials and impurities 4-2
  - statement description format 4-4
  - syntax 4-2
- **`INTERACTIVE`** 4-11
  - description 4-11
  - example 4-11
  - interactive input mode 4-11
- interface functions 7-14
- interface oxide break-up and epitaxial regrowth
  - oxide break-up 3-167
- interface trap model 3-57, 7-20
- interface trap model, impurities 3-55
- **`INTERMEDIATE`** 4-30
  - advantages of **`INTERMEDIATES`** 4-31
  - advantages of intermediates 7-8
  - array values 4-34
  - comparison to **`ASSIGN`** 4-36
  - value type 4-33
- **`INTERSTITIAL`** 4-311
  - additional notes 4-326
  - bulk and interface parameters 4-325
  - description 4-324
  - examples 4-325
- interstitial and vacancy diffusivities 3-63
- interstitial clustering model 3-71, 7-15
- interstitial traps 3-70
- interstitial traps, enabling, disabling, and initialization 3-71
- interstitials
  - absorption of traps 3-66
  - net recombination rate 3-64
- intrinsic stress in deposited layers
  - stress history model 3-116
  - stress model 3-116
- introduction 1-iii to ??
  - typographical conventions 1-iv
- introduction to TSUPREM-4 1-1
- ion implant data file *s4imp0* 2-9
- ion implantation
  - advanced example of 6-18
  - analytic models 3-119
  - boundary conditions 3-149
  - implant damage model 3-144

- into silicon carbide 3-152
- Monte Carlo model 3-131
- overview 3-119

ion pairing and mobile impurities 3-19

ION parameter, **IMPLANT** statement
- trench implant simulation example 6-24

J

JCAP parameter, **ELECTRICAL** statement
- NMOS LDD process example 6-14

JUNCTION parameter, **ELECTRICAL** statement
- NMOS LDD process example 6-16

K

key files *s4fky0* and *s4uky0* 2-10

L

L.MODIFY 4-20
- description 4-20

LABEL 4-175
- CM parameter, 1D bipolar example 5-8
- color 4-179
- description 4-178
- examples 4-179
- label placement 4-178
- line, symbol, and rectangle 4-178
- X parameter, 1D bipolar example 5-8
- Y parameter, 1D bipolar example 5-8

LAYERS parameter, **PRINT.1D** statement
- DMOS power transistor example 6-45
- local oxidation example 5-9

library files 2-8
- authorization file *s4auth* 2-10
- initialization input file *s4init* 2-9
- ion implant data file, *s4imp0* 2-9
- key files *s4fky0* and *s4uky0* 2-10
- plot device definition file *s4pcap* 2-10

LINE 4-57
- additional notes 4-58
- default regions and boundaries 4-58
- description 4-57
- example 4-58
- maximum number of nodes and grid lines 4-58
- placing grid lines 4-57
- structure depth and point defect models 4-58

LINE.TYP parameter, **CONTOUR** statement
- 1D local oxidation example 5-24

LINE.TYPE parameter, **PLOT.1D** statement
- 1D bipolar example 5-7

listings
- input file *s4ex11a.inp* for simulating bipolar emitter structure *Fig.* 6-60
- input file *s4ex1b.inp*, showing statements simulating the field oxide, base, and emitter region processing for bipolar transistor *Fig.* 5-10
- mask information read from file *s4ex4m.tl1 Fig.* 6-4
- MINIMOS 5 command file *EX2D.INP Fig.* E-7
- MINIMOS 5 command file *EX2D.INP*, modified to read doping profiles produced by TSUPREM-4 *Fig.* E-8
- **PRINT.1D** command in file *s4ex1a.inp Fig.* 5-9
- produced by **PRINT.1D** statement in input file *s4ex3.inp Fig.* 5-33

LOADFILE 4-71
- description 4-71
- examples 4-72
- Taurus-Lithography files 4-72
- user-defined materials and impurities 4-72

local oxidation example
- **COMPRESS** oxidation model 5-20
- **CONTOUR** statement 5-18, 5-24
- **DIFFUSION** statement 5-15, 5-20
- diffusion with point defects 5-20
- **END** statement 5-18, 5-24
- **FOREACH** statement 5-18, 5-24
- **INITIALIZE** statement 5-14
- **LABEL** statement 5-25, 5-26
- mesh generation 5-14
- model selection 5-15
- overview 5-13
- oxide shape 5-13
- pad oxide and nitride layers 5-15
- **PLOT.1D** statement 5-26
- plotting
  - mesh 5-15
  - results 5-16
  - stresses 5-17
- **PRINT.1D** statement 5-19
- **SAVEFILE** statement 5-15
- **SELECT** statement 5-18
- summary 5-27

LOOP/L.END 4-16
- advantages 4-19

  - dependence and variability 4-17
  - description 4-17
  - example 4-18
  - parameter sensitivity 4-17
  - termination of optimization looping 4-17

LOW parameter, **ELECTRICAL** statement
  - NMOS LDD process example 6-16

LY.ACTIV parameter, **MESH** statement
  - CMOS process example 6-34

LY.SURF parameter, **MESH** statement
  - CMOS process example 6-34

M

MASK 4-84
  - description 4-84
  - examples 4-85

mask data files
  - format C-1
  - overview 2-5, C-1

MATERIAL 4-258
  - description 4-270
  - examples 4-270
  - POLYCRYS parameter, NMOS LDD process example 6-7
  - viscosity and compressibility 4-270

material coefficients, default coefficients A-23

materials, specifying 4-2

mathematical functions 7-11

Medici
  - **MEDICI** parameter 4-78
  - saved structure files overview 2-7
  - structure file format D-9

Medici structure file *Fig.* D-11

MESH 4-53
  - 1D mode 4-56
  - description 4-54
  - examples 4-56
  - grid creation methods 4-54
  - GRID.FAC parameter, 1D bipolar example 5-4
  - horizontal grid generation 4-55
  - scaling the grid spacing 4-56
  - scaling the grid spacing, **GRID.FAC** 4-56
  - vertical grid generation 4-55

mesh
  - changes during processing 3-7
  - **DEPOSITION** statement 3-7
  - **DEVELOP** statement 3-8
  - **EPITAXY** statement 3-7
  - **ETCH** statement 3-8
  - overview 3-3
  - oxidation and silicidation 3-9
  - structure extension 3-8

mesh generation
  - depositing a layer with nonuniform grid spacing 6-50
  - DMOS power transistor, from input file *s4ex8.inp Fig.* 6-42
  - SOI MOSFET, from input file *s4ex9.inp Fig.* 6-49
  - tutorial example 5-4

mesh used for oxidation simulation *Fig.* 5-14

METHOD 4-207
  - adaptive gridding 4-224
    - analytical implant models 4-224
    - disable 4-224
  - block solution 4-227
  - customizing the point defect models 4-223
  - description 4-221
  - enable/disable user-specified models 4-223
  - ERR.FAC parameter, 1D bipolar example 5-4
  - error tolerances 4-228
  - examples 4-228, 4-231
  - fine control 4-225
  - grid spacing in oxide 4-221
  - initial time step 4-225
  - internal solution methods 4-225
  - internal solution methods, changing 4-225
  - matrix refactoring 4-227
  - matrix structure 4-227
  - minimum-fill reordering 4-226
  - oxidation models 4-221
  - PD.FERMI model 4-222
  - PD.FULL model 4-222
  - PD.TRANS model 4-222
  - point defect modeling 4-222
  - rigid vs. viscous substrate 4-221
  - solution method 4-227
  - system solutions 4-226
  - time integration 4-225

MINIMOS 5 interface E-1

miscellaneous functions 7-15

MOBILE 7-9

mobile impurities and ion pairing 3-19

MOBILITY 4-305
  - analytic models 4-309
  - description 4-308
  - example 4-309
  - tables and analytic models 4-309
  - tables or model selection 4-309

- models and coefficients, input statements
  - **AMBIENT** 4-233
  - **ANTIMONY** 4-337
  - **ARSENIC** 4-344
  - **BORON** 4-351
  - **EQUATION** 4-230
  - **IMPURITY** 4-273
  - **INTERSTITIAL** 4-311
  - **MATERIAL** 4-258
  - **METHOD** 4-207
  - **MOBILITY** 4-305
  - **MOMENT** 4-251
  - **PHOSPHORUS** 4-358
  - **REACTION** 4-300
  - **VACANCY** 4-327
- modification of built-in functions 7-9
  - **ACTIVE** 7-9
  - **MOBILE** 7-9
- **MOMENT** 4-251
  - additional notes 4-257
  - description 4-255
  - examples 4-256
  - optional and required model parameters 4-255
  - using 4-255
- Monte Carlo implant
  - amorphization 3-143
  - damage dechanneling 3-143
  - default coefficients A-26
  - model 3-131
  - model, binary scattering theory 3-133
  - overview 6-23
  - using 6-23
- **MONTECAR** parameter, **IMPLANT** statement
  - trench implant simulation example 6-20, 6-23
- MOS capacitance 3-177
- **MOSCAP** parameter, **ELECTRICAL** statement
  - NMOS LDD process example 6-14
- MOSFET with self-aligned silicides example
  - **COMPRESS** model 6-55
  - overview 6-54
  - **PD.FERMI** model 6-55
  - preparation for silicidation 6-54
  - silicidation 6-55
- MOSFET, quantum mechanical model 3-175
- moving-boundary flux 3-55, 3-70
- multilayer implants 3-127
  - dose matching 3-127
  - effective range model 3-127

N

- N20 oxidation
  - nitrogen trap and generation 3-105
- N2O oxidation 3-105
- net recombination rate of interstitials 3-64
- new solution variable 7-3
- nitrogen trap and generation 3-105
- NMOS LDD process example
  - active region simulation 6-7
  - **COMPRESS** oxidation model 6-5
  - creating initial structure 6-2
  - **DEPOSITION** statement 6-7, 6-9
  - **DEVELOP** statement 6-5, 6-7
  - electrical extraction 6-14
  - **ELIMINATE** statement 6-2
  - **ETCH** statement 6-7
  - formation of complete NMOS transistor 6-12
  - **INITIALIZE** statement 6-2, 6-7
  - **LINE** statement 6-2
  - **MASK** statement 6-2, 6-7
  - **MESH** statement 6-2, 6-7
  - **METHOD** statement 6-7
  - overview 6-2
  - **PD.FERMI** model 6-4
  - **PD.TRANS** model 6-5
  - **PLOT.2D** statement 6-5
  - remove photoresist 6-4
  - **SAVEFILE** statement 6-11
  - **TRAP** etch model 6-5
  - **VERTICAL** model 6-4
  - **VISCOELA** model 6-5
- **NMOS** parameter, **ELECTRICAL** statement
  - NMOS LDD process example 6-14
- NMOS structure
  - after LDD implant, as plotted by file *s4ex4b.inp Fig.* 6-9
  - to transfer to MINIMOS 5 *Fig.* E-3
- nodes
  - addition in growing oxide 3-9
  - maximum number 3-3
  - overview 3-3
  - regions where oxide is forming 3-10
  - removing in consumed silicon 3-9
- numerical methods, default coefficients A-26
- numerical oxidation models 3-93
  - **COMPRESS** model 3-96
  - compressible viscous flow 3-96
  - concentration dependence 3-94
  - oxide growth rate 3-93

  - thin regime 3-95
  - usage 3-95
  - **VERTICAL** model 3-95
  - **VISCOELA** model 3-99
  - **VISCOUS** model 3-97

O

- operators 7-11
- optimization
  - **ASSIGN** statement 4-23
  - **ELECTRICAL** statement 4-201
  - **EXTRACT** statement 4-188
  - **LOOP** statement 4-16
- **OPTION** 4-41
  - DEVICE parameter, 1D bipolar example 5-7
  - echoing and execution of input statements 4-43
  - examples 4-45
  - informational and diagnostic output 4-43
  - printed output 4-42
  - redirecting graphics output 4-42
  - selecting a graphics device 4-42
  - version compatibility 4-43
- other silicides 3-114
- output files 2-5
  - electrical data 2-8
  - extract 2-8
  - graphical 2-7
  - listing 2-6
  - saved structure 2-7
  - terminal 2-6
- output listing files 2-6
  - *s4dia* 2-6
  - *s4inf* 2-6
  - *s4out* 2-6
- output, input statements
  - **COLOR** 4-170
  - **CONTOUR** 4-168
  - **ELECTRICAL** 4-194
  - **EXTRACT** 4-180
  - **LABEL** 4-175
  - **PLOT.1D** 4-155
  - **PLOT.2D** 4-163
  - **PLOT.3D** 4-172
  - **PRINT.1D** 4-151
  - **SELECT** 4-146
  - **VIEWPORT** 4-204
- output, program
  - graphical 2-2
  - overview 2-2
  - printed 2-2
- oxidation
  - advanced example of 6-29
  - analytical models 3-87
  - default coefficients A-11
  - N20 3-105
  - numerical models 3-93
  - overview 3-84
  - polysilicon 3-104
  - surface reaction rate 3-106
  - theory of 3-85
  - thin oxidation rate 3-106
  - tutorial example 5-4, 5-13
- oxidation and plotting of impurity profiles
  - doping and layer information 5-33
  - **PD.FERMI** model 5-32
  - **PD.FULL** model 5-32
  - **PD.TRANS** model 5-32
  - printing junction depth 5-33
  - simulation procedure 5-31
- oxidation and silicidation 3-9
- oxide growth rate 3-93
- oxide reflow
  - see reflow

P

- **P1.X** parameter, **ETCH** statement
  - 1D local oxidation example 5-15
- **P1.Y** parameter, **ETCH** statement
  - 1D local oxidation example 5-15
- **P2.X** parameter, **ETCH** statement
  - local oxidation example 5-15
- **P2.Y** parameter, **ETCH** statement
  - 1D local oxidation example 5-15
- paired fractions of dopant atoms 3-25
- parameters
  - definition table 4-4
  - deposition 4-104
  - list syntax 4-4
  - types 4-3
- parameters, **AMBIENT** statement
  - **/AMBIENT** 3-93, 4-241
  - **/MATERIA** 4-240
  - **/NITRIDE** 4-241
  - **/OXIDE** 4-240
  - **/OXYNITR** 4-241
  - **/POLYSIL** 4-241

/SILICON 4-240
<100> 3-87, 3-88, 3-106, 4-235
<110> 3-87, 3-88, 3-106, 4-235
<111> 3-87, 3-88, 3-106, 4-235
ALPHA 3-104, 4-241
AMB.1 3-15, 4-234
AMB.2 3-15, 4-234
AMB.3 3-15, 4-234
AMB.4 3-15, 4-234
AMB.5 3-15, 4-234
AMBIENT 4-239
CLEAR 4-245
CM.SEC 4-245
COLUMN 3-88, 4-239
D.0 3-93, 4-240
D.E 3-93, 4-240
DRYO2 3-15, 4-234
ERF.DELT 3-90, 4-244
ERF.H 3-90, 3-91, 4-244
ERF.LBB 3-90, 3-91, 4-244
ERF.Q 3-90, 4-244
ERF1 3-92
ERF2 3-92
F.H2 3-15, 4-235
F.H2O 3-15, 4-235
F.HCL 3-15, 4-235
F.N2 3-15, 4-235
F.N2O 4-235
F.O2 3-15, 4-234
GAMMA.0 3-94, 4-238
GAMMA.E 3-94, 4-238
GB.FAC.0 4-238
GB.FAC.E 4-238
H.LIN.0 3-88, 4-236
H.LIN.E 3-88, 4-236
H.PAR.0 3-89, 4-237
H.PAR.E 3-89, 4-237
H2O 4-235
HCL 3-15, 4-235
HENRY.CO 3-93, 4-240
INERT 3-15, 4-234
INITIAL 3-89, 3-90, 4-244, 5-5
L.LIN.0 3-88, 4-236
L.LIN.E 3-88, 4-236
L.PAR.0 3-89, 4-237
L.PAR.E 3-89, 4-237
LIN.BREA 3-88, 4-237
LIN.CLDE 3-88, 4-238
LIN.PCT 3-88, 4-238
LIN.PDEP 3-88, 4-238
MASK.EDG 3-87, 4-244
MATERIAL 4-239
N.0 4-236
N.E 4-237
N.G.0 4-241
N.G.E 4-242
N.GPOW.0 4-242
N.GPOW.E 4-242
N.THIN.0 4-237
N.THIN.E 4-237
N.V.0 4-242
N.V.E 4-242
N2O 4-234
NIT.THIC 3-90, 3-91, 4-244
NITRIDE 4-239
O2 4-235
ORIENTAT 4-236
OXIDE 3-93, 4-239
OXYNITRI 4-239
PAR.BREA 3-89, 4-237
PAR.CLDE 3-89, 4-238
PAR.PCT 3-89, 4-238
PAR.PDEP 3-89, 4-238
POLYSILI 3-87, 3-88, 3-106, 4-236, 4-239
PRESSURE 3-15, 4-235
SEG.0 4-241
SEG.E 4-241
SILICON 4-239
SPREAD 3-90, 4-244
STEAM 3-15, 4-234
STRESS.D 3-98, 3-104, 4-242, 4-247
TABLE 3-88, 4-239
TEMPERAT 3-88, 4-244
THETA 3-93, 4-240
THINOX.0 3-87, 4-236
THINOX.E 3-87, 4-236
THINOX.L 3-87, 4-236
TRANS.0 3-93, 4-241
TRANS.E 3-93, 4-241
VC 3-99, 4-240
VD 3-99, 4-243
VDLIM 3-99, 4-244
VR 3-99, 3-104, 4-242, 4-247
VR100 4-242
VR110 4-243
VR111 4-243
VT 3-99, 3-104, 4-243, 4-247
WETO2 3-15, 4-234
parameters, **ANTIMONY** statement
/AMBIENT 4-340

- **/MATERIA** 4-340
- **/NITRIDE** 4-340
- **/OXIDE** 4-340
- **/OXYNITR** 4-340
- **/POLYSIL** 4-340
- **/SILICON** 4-340
- **AMBIENT** 4-338
- **CM.SEC** 4-342
- **DIM.0** 3-22, 4-338
- **DIM.E** 3-22, 4-338
- **DIMM.0** 3-22
- **DIMM.E** 3-22
- **DIP.E** 3-22
- **DIP.O** 3-22
- **DIPAIR.0** 3-25, 4-338
- **DIPAIR.E** 3-25, 4-338
- **DIX.0** 3-22, 4-338
- **DIX.E** 3-22, 4-338
- **DVM.0** 3-22, 4-338
- **DVM.E** 3-22, 4-338
- **DVMM.0** 3-22
- **DVMM.E** 3-22
- **DVP.0** 3-22
- **DVP.E** 3-22
- **DVPAIR.0** 3-25, 4-338
- **DVPAIR.E** 3-25, 4-338
- **DVX.0** 3-22, 4-338
- **DVX.E** 3-22, 4-338
- **E.I.S** 3-32, 4-339
- **E.IP.V** 3-32, 4-339
- **E.V.S** 3-32, 4-339
- **E.VP.I** 3-32, 4-339
- **ES.100** 4-342
- **ES.110** 4-342
- **ES.F.100** 4-342
- **ES.F.110** 4-342
- **ES.F.RAN** 4-341
- **ES.RAND** 4-341
- **IP.V.HIG** 4-339
- **IP.V.LOW** 4-339
- **IP.V.MED** 4-339
- **KSIP.0** 4-341
- **KSIP.E** 4-341
- **KSVP.0** 4-341
- **KSVP.E** 4-341
- **MATERIAL** 4-337
- **NITRIDE** 4-337
- **OXIDE** 4-337
- **OXYNITRI** 4-337
- **POLYSILI** 4-337
- **R.I.S** 3-32, 4-339
- **R.IP.V** 3-32, 4-339
- **R.V.S** 3-32, 4-339
- **R.VP.I** 3-32, 4-339
- **SEG.0** 4-341
- **SEG.E** 4-341
- **SILICON** 4-337
- **SS.CLEAR** 4-340
- **SS.CONC** 4-340
- **SS.TEMP** 4-340
- **TRANS.0** 4-341
- **TRANS.E** 4-341
- **VP.I.HIG** 4-340
- **VP.I.LOW** 4-340
- **VP.I.MED** 4-340

parameters, **ARSENIC** statement
- **/AMBIENT** 4-347
- **/MATERIA** 4-347
- **/NITRIDE** 4-347
- **/OXIDE** 4-347
- **/OXYNITR** 4-347
- **/POLYSIL** 4-347
- **/SILICON** 4-347
- **AMBIENT** 4-345
- **CM.SEC** 4-349
- **CTN.0** 4-347
- **CTN.E** 4-347
- **CTN.F** 4-347
- **DIM.0** 3-22, 4-345
- **DIM.E** 3-22, 4-345
- **DIMM.0** 3-22
- **DIMM.E** 3-22
- **DIP.0** 3-22
- **DIP.E** 3-22
- **DIPAIR.0** 3-25, 4-345
- **DIPAIR.E** 3-25, 4-345
- **DIX.0** 3-22, 4-345
- **DIX.E** 3-22, 4-345
- **DVM.0** 3-22, 4-345
- **DVM.E** 3-22, 4-345
- **DVMM.0** 3-22
- **DVMM.E** 3-22
- **DVP.0** 3-22
- **DVP.E** 3-22
- **DVPAIR.0** 3-25, 4-345
- **DVPAIR.E** 3-25, 4-345
- **DVX.0** 3-22, 4-345
- **DVX.E** 3-22, 4-345
- **E.I.S** 3-32, 4-346
- **E.IP.V** 3-32, 4-346

E.V.S 3-32, 4-346
E.VP.I 3-32, 4-346
ES.100 4-349
ES.110 4-349
ES.F.100 4-349
ES.F.110 4-349
ES.F.RAN 4-348
ES.RAND 4-348
IP.V.HIG 4-346
IP.V.LOW 4-346
IP.V.MED 4-346
KSIP.0 4-348
KSIP.E 4-348
KSVP.0 4-348
KSVP.E 4-348
MATERIAL 4-344
NITRIDE 4-344
OXIDE 4-344
OXYNITRI 4-344
POLYSILI 4-344
R.I.S 3-32, 4-346
R.IP.V 3-32, 4-346
R.V.S 3-32, 4-346
R.VP.I 3-32, 4-346
SEG.0 4-348
SEG.E 4-348
SILICON 4-344
TRANS.0 4-348
TRANS.E 4-348
VP.I.HIG 4-347
VP.I.LOW 4-347
VP.I.MED 4-347

parameters, **ASSIGN** statement
ARRAY 4-26
C.COUNT 4-26
C.FILE 4-24
C.VALUE 4-24
C1 4-25
C10 4-26
C2 4-25
C3 4-25
C4 4-25
C5 4-25
C6 4-25
C7 4-25
C8 4-25
C9 4-26
COLUMN 4-26
DATA 4-26
DELETE 4-23
DELTA 4-24
IN.CVALU 4-26
IN.FILE 4-26
IN.NVALU 4-26
LINE 4-24
LOG 4-24
LOWER 4-24
N.EXPRES 4-23
N.VALUE 4-24
NAME 4-23
PRINT 4-23
PROMPT 4-23
RATIO 4-24
ROW 4-26
TIF 4-26
UPPER 4-24

parameters, **BORON** statement
/AMBIENT 4-354
/MATERIA 4-354
/NITRIDE 4-354
/OXIDE 4-354
/OXYNITR 4-354
/POLYSIL 4-354
/SILICON 4-354
AMBIENT 4-352
CM.SEC 4-356
DIM.0 3-22
DIM.E 3-22
DIMM.0 3-22
DIMM.E 3-22
DIP.0 3-22, 4-352
DIP.E 3-22, 4-352
DIPAIR.0 3-25, 4-352
DIPAIR.E 3-25, 4-352
DIX.0 3-22, 4-352
DIX.E 3-22, 4-352
DVM.0 3-22
DVM.E 3-22
DVMM.0 3-22
DVMM.E 3-22
DVP.0 3-22, 4-352
DVP.E 3-22, 4-352
DVPAIR.0 3-25, 4-352
DVPAIR.E 3-25, 4-352
DVX.0 3-22, 4-352
DVX.E 3-22, 4-352
E.I.S 3-32, 4-353
E.IP.V 3-32, 4-353
E.V.S 3-32, 4-353
E.VP.I 3-32, 4-353

  ES.100 4-356
  ES.110 4-356
  ES.F.100 4-356
  ES.F.110 4-356
  ES.F.RAN 4-355
  ES.RAND 4-355
  IP.V.HIG 4-353
  IP.V.LOW 4-353
  IP.V.MED 4-353
  KSIP.0 4-355
  KSIP.E 4-355
  KSVP.0 4-355
  KSVP.E 4-355
  MATERIAL 4-351
  NITRIDE 4-351
  OXIDE 4-351
  OXYNITRI 4-351
  POLYSILI 4-351
  R.I.S 3-32, 4-353
  R.IP.V 3-32, 4-353
  R.V.S 3-32, 4-353
  R.VP.I 3-32, 4-353
  SEG.0 4-355
  SEG.E 4-355
  SILICON 4-351
  SS.CLEAR 4-354
  SS.CONC 4-354
  SS.TEMP 4-354
  TRANS.0 4-355
  TRANS.E 4-355
  VP.I.HIGH 4-354
  VP.I.LOW 4-354
  VP.I.MED 4-354
parameters, **BOUNDARY** statement
  EXPOSED 4-62
  REFLECTI 4-62
  XHI 4-62
  XLO 4-62
  YHI 4-62
  YLO 4-62
parameters, **COLOR** statement
  ALUMINUM 4-170
  COLOR 4-170
  MATERIAL 4-170
  MAX.VALU 4-170
  MIN.VALU 4-170
  NITRIDE 4-170
  OXIDE 4-170
  OXYNITRI 4-170
  PHOTORES 4-170
  POLYSILI 4-170
  SILICON 4-170
parameters, **CONTOUR** statement
  COLOR 4-168
  LINE.TYP 4-168, 5-24
  SYMBOL 4-168
  VALUE 4-168, 5-24
parameters, **CPULOG** statement
  LOG 4-50
  OUT.FILE 4-50
parameters, **DEPOSITION** statement
  ALUMINUM 3-155, 4-99
  ANISOTRO 4-102
  ANTIMONY 3-156, 4-100
  ARC.SPAC 3-155, 4-101, 4-139
  ARSENIC 3-156, 4-100
  BORON 3-156, 4-101
  CONCENTR 4-101
  DY 3-7, 4-101, 6-50
  GSZ.LIN 3-165, 4-102
  I.CONC 3-156
  I.RESIST 4-100, 4-138
  IMPURITY 3-156
  MATERIAL 4-99
  NEGATIVE 4-100
  NITRIDE 3-155, 4-99
  OXIDE 3-155, 4-99
  OXYNITRI 3-155, 4-99
  PHOSPHOR 3-156, 4-101
  PHOTORES 3-155, 4-99
  POLYSILI 3-155, 4-99
  POSITIVE 4-99
  RESISTIV 4-101
  SILICON 3-155, 4-99
  SPACES 3-8, 4-101, 6-50
  STRAINED 4-102
  TEMPERAT 4-101
  THICKNES 3-7, 4-101, 5-6
  TOPOGRAP 4-102
  YDY 3-8, 4-101, 6-50
parameters, **DEVELOP** statement
  NEGATIVE 3-156
  POSITIVE 3-156
parameters, **DIFFUSION** statement
  AMB.1 4-132
  AMB.2 4-132
  AMB.3 4-132
  AMB.4 4-132
  AMB.5 4-132
  ANTIMONY 3-17, 4-133

  ARSENIC 3-17, 4-133
  BORON 3-17, 4-133
  CONTINUE 4-131
  D.RECOMB 4-134
  DRYO2 3-84, 4-131, 5-6
  DUMP 4-134
  F.H2 4-132
  F.H2O 4-132
  F.HCL 4-133
  F.N2 4-133
  F.N2O 4-132
  F.O2 4-132
  HCL 3-17, 4-134
  I.CONC 3-17
  IMPURITY 3-17
  INERT 4-132
  MOVIE 4-134, 4-139
  N2O 3-84, 4-132
  P.FINAL 3-15, 4-134
  P.RATE 3-15, 4-134
  P.SIDE 4-142
  P.TOP 4-142
  PHOSPHOR 3-17, 4-133
  PRESSURE 4-133
  STEAM 3-84, 4-132, 5-5
  T.FINAL 3-15, 4-131, 4-142
  T.RATE 3-15, 4-131, 4-142
  TEMPERAT 3-15, 4-131, 4-142
  TIME 3-15, 4-131, 4-142
  WETO2 3-84, 4-131
parameters, **ELECTRICAL** statement
  ANGLE 4-194
  BIAS.REG 4-195
  BULK.LAY 4-196
  BULK.REG 4-196
  DEEP 4-196, 6-16
  DEPTH 4-195
  DISTRIB 4-197
  E.RVCAP 6-16
  EXT.REG 4-195
  GATE.ELE 4-196
  GATE.WF 4-196
  HIGH 4-196, 6-16
  JCAP 4-195, 6-14
  JUNCTION 4-195, 6-16
  LOW 4-196, 6-16
  MIN.ABS 4-199
  MIN.REL 4-199
  MOSCAP 4-196, 6-14
  NAME 4-197
  NMOS 4-196, 6-14
  OUT.FILE 3-170, 4-197
  PITCH 4-195
  PMOS 4-196
  POINT 4-195
  PRINT 4-197
  QM 3-176, 4-196
  QSS 4-196, 6-14
  RESISTAN 4-195, 6-14
  SENSITIV 4-197
  SRP 4-194
  T.COLUMN 4-198
  T.FILE 4-197
  T.LOWER 4-198
  T.TRANSF 4-198
  T.UPPER 4-198
  TARGET 4-197
  TEMPERAT 3-172, 4-197
  THRESHOL 4-196
  THRESHOLD 6-14
  TOLERANC 4-199
  V 4-195
  V.COLUMN 4-197
  V.LOWER 4-198
  V.SELECT 4-197
  V.TRANSF 4-198
  V.UPPER 4-198
  VB 4-196
  VSTART 4-195
  VSTEP 4-195
  VSTOP 4-195
  WEIGHT 4-199
  X 4-194, 6-16
  Y.SURFAC 4-195
  Z.VALUE 4-198
parameters, **ELECTRODE** statement
  ALL 4-96
  BOTTOM 4-96
  CLEAR 4-96
  MERGE 4-96
  NAME 4-96
  PRINT 4-96
  X 4-96
  Y 4-96
parameters, **ELIMINATE** statement
  COLUMNS 4-59
  ROWS 4-59
  X.MAX 4-59
  X.MIN 4-59
  Y.MAX 4-59

Y.MIN 4-59
parameters, **EPITAXY** statement
ANTIMONY 3-154, 4-138
ARC.SPAC 4-139
ARSENIC 3-154, 4-138, 5-6
BLANKET 4-139
BORON 3-154, 4-138
CONCENTR 4-138
DY 3-7, 4-139
I.CONC 3-154
I.RESIST 4-138
IMPURITY 3-154
PHOSPHOR 3-154, 4-138
RESISTIV 4-138
SELECTIV 4-139
SPACES 3-8, 4-139, 5-6
T.FINAL 3-153, 3-154, 4-137
T.RATE 3-153, 3-154, 4-137
TEMPERAT 3-153, 4-137, 5-6
THICKNES 3-7, 4-138
TIME 3-153, 4-137, 5-6
YDY 3-8, 4-139
parameters, **EQUATION** statement
/MATERIAL 4-230
ADDTOEXP 4-230
BOUNDARY 4-231
INITIAL 4-230
INTERFACE 4-231
MATERIAL 4-230
MODEL 4-230
TIF.NAME 4-230
TRANSPOR 4-231
UPDATE 4-230
VARIABLE 4-230
parameters, **ETCH** statement
ALL 3-157, 4-111
ALUMINUM 3-157, 4-109
ANGLE 3-157, 4-110
CONTINUE 3-157, 4-110
DONE 3-157, 4-110
ISOTROPI 3-157, 4-110
LEFT 3-157, 4-110
MATERIAL 4-109
NITRIDE 3-157, 4-109
OLD.DRY 3-157, 4-111
OXIDE 3-157, 4-109
OXYNITRI 3-157, 4-109
P1.X 3-157, 4-110, 5-15
P1.Y 3-157, 4-110, 5-15
P2.X 3-157, 4-110, 5-15
P2.Y 3-157, 4-110, 5-15
PHOTORES 3-157, 4-109
POLYSILI 3-157, 4-109
RIGHT 3-157, 4-110, 5-15
SILICON 3-157, 4-109
START 3-157, 4-110
THICKNES 3-157, 4-110
TOPOGRAP 4-111
TRAP 6-36
TRAPEZOI 3-157, 4-109
UNDERCUT 3-157, 4-110
X 3-157, 4-111
Y 3-157, 4-111
parameters, **EXPOSE** statement
MASK 4-106
OFFSET 4-106
SHRINK 4-106
parameters, **EXTRACT** statement
/ALUMINU 4-181
/AMBIENT 4-181
/MATERIA 4-181
/NITRIDE 4-181
/OXIDE 4-181
/OXYNITR 4-181
/PHOTORE 4-181
/POLYSIL 4-181
/SILICON 4-181
ALUMINUM 4-181
APPEND 4-185
AREA.EXT 4-182
ASSIGN 4-183
AVG.EXTR 4-182
CLOCKWIS 4-182
CLOSE 4-185
D.EXTRAC 4-182
DISTANCE 4-182
INT.EXTR 4-182
MATERIAL 4-180
MAXIMUM 4-182
MIN.ABS 4-185
MIN.REL 4-185
MINIMUM 4-182
NAME 4-183
NITRIDE 4-180
OUT.FILE 4-185
OXIDE 4-180
OXYNITRI 4-180
P1.X 4-181
P1.Y 4-181
P2.X 4-181

- **P2.Y** 4-181
- **PHOTORES** 4-181
- **POLYSILI** 4-180
- **PREFIX** 4-183
- **PRINT** 4-183
- **SENSITIV** 4-185
- **SEPARAT** 4-183
- **SILICON** 4-180
- **SUFFIX** 4-183
- **T.COLUMN** 4-184
- **T.FILE** 4-183
- **T.LOWER** 4-184
- **T.TRANSF** 4-184
- **T.UPPER** 4-184
- **TARGET** 4-183
- **TOLERANC** 4-185
- **V.COLUMN** 4-183
- **V.LOWER** 4-184
- **V.TRANSF** 4-184
- **V.UPPER** 4-184
- **VAL.EXTR** 4-182
- **VALUE** 4-182
- **WEIGHT** 4-185
- **WRITE** 4-183
- **X** 4-182
- **X.EXTRAC** 4-182
- **Y** 4-182
- **Y.EXTRAC** 4-182
- **Z.VALUE** 4-184

parameters, **IMPLANT** statement
- **ALSO.TAB** 4-118
- **ANTIMONY** 3-119, 4-114, 4-253
- **ARSENIC** 3-119, 4-114, 4-253
- **BACKSCAT** 3-120, 4-117
- **BEAMWIDT** 3-140, 4-120
- **BF2** 3-119, 4-115, 4-253
- **BF2** 3-144
- **BORON** 3-119, 4-114, 4-253
- **CRIT.110** 3-141, 4-117, 4-121
- **CRIT.F** 3-141
- **CRIT.PRE** 3-141
- **CRYSTAL** 3-140, 4-120, 6-24
- **D.P1** 4-116
- **D.PLUS** 3-148, 4-116
- **D.RAN.F** 4-254
- **D.RECOMB** 4-116
- **D.SCALE** 3-148, 4-116
- **D.SIG.F** 4-254
- **D.SUP.E** 4-254
- **D.SUP.R** 4-254
- **DAM.MOD** 4-121
- **DAM.SUFFIX** 4-120
- **DAM.TABL** 4-254
- **DAMAGE** 3-129, 4-116
- **DATA.PAT** 4-120
- **DATA.SUFFIX** 4-119
- **DOSE** 3-119, 4-115, 6-23
- **E.LIMIT** 3-142, 4-123
- **ENERGY** 3-119, 4-115, 6-23
- **GAUSSIAN** 3-120, 4-117
- **IMPL.TAB** 3-121, 4-118
- **IMPURITY** 4-114, 4-253
- **IN.FILE** 3-121, 4-119
- **INTEREST** 4-123
- **ION** 6-24
- **L.DENS** 3-83, 4-116
- **L.DMAX** 4-117
- **L.DMIN** 3-83, 3-84, 4-116
- **L.FRAC** 4-117, 4-119
- **L.RADIUS** 3-83, 4-116
- **L.THRESH** 4-117
- **MAX.DAMA** 3-143, 4-123
- **MAX.SPLI** 4-121
- **MOMENTS** 4-118
- **MONTECAR** 3-131, 3-133, 4-121, 6-20
- **N.ION** 3-137, 4-120
- **OLD.MC** 4-122
- **PAI** 4-117
- **PAI.THRESH** 4-117
- **PD.FULL** 3-148
- **PD.PTIME** 3-148
- **PEARSON** 3-120, 4-117
- **PERIODIC** 3-149, 4-115, 4-116
- **PHOSPHOR** 3-119, 4-115, 4-253
- **POINT.RE** 4-121
- **POLY.GSZ** 4-123
- **PRINT** 4-118, 4-119
- **RANDOM** 4-121
- **RANGE.F** 4-254
- **RANGE.SHIFT** 4-120
- **REC.FRAC** 3-142, 4-123
- **REC.OXYG** 4-121
- **REFLECT** 3-149, 4-116
- **ROTATION** 3-129, 4-115
- **RP.EFF** 3-127, 4-119
- **SAV1D** 4-122
- **SAV1D.FILE** 4-122
- **SAV1D.HISTO** 4-122
- **SAV1D.SHIFT** 4-122
- **SAV1D.UNITS** 4-122

SEED 4-120
SIGMA.F 4-254
TABLE 4-254
TEMPERAT 3-142, 4-121
THRESHOL 3-142, 4-123
TILT 3-120, 4-115, 6-20
TS4.REGRID 4-117
VACUUM 3-149, 4-116
VIBRATIO 3-142, 4-122
X.LENGTH 4-121
X.RMS 3-142, 4-122
parameters, **IMPURITY** statement
/MATERIA 3-55, 4-290
/RATIO.0 3-58, 4-291
/RATIO.E 3-58, 4-292
/SEG.0 3-58, 4-291
/SEG.E 3-58, 4-291
/SEG.EQ2 3-58, 4-292
/SEG.SS 4-292
/TRANS.0 4-291
/TRANS.E 4-291
ACCEPTOR 4-276
ACT.AMOR 4-288
ACT.MIN 3-38, 4-277
ACT.NI 4-288
ACTIVE 4-290
ALPHA 3-164
AMOR.PAR 4-296
AT.NUM 4-274, 4-276
AT.RADIU 4-276
AT.WT 4-276
C.I.ALL 4-282
C.INTERS 4-282
C.IP.ALL 4-282
C.IPAIR 4-282
C.STATE 4-280
C.V.ALL 4-282
C.VANC 4-282
C.VP.ALL 4-282
C.VPAIR 4-282
C0.INI.F 4-287, 4-288
C1.INI.F 4-287
CASC.AMO 4-296
CASC.DIS 4-296
CASC.SUR 4-296
CG.MAX 3-164, 4-288
CHAN.CRI 3-141, 4-295
CHAN.FAC 3-142, 4-295
CHARGE 4-276
CL.INI.A 4-283
CM.SEC 4-277
CO.TF.E 4-285
CO.TF.O 4-285
CO.TR.E 4-285
CO.TR.O 4-285
CTN.0 3-37, 4-283
CTN.E 3-37, 4-283
CTN.F 3-37, 4-283
D.PF 4-277, 4-321
D.PNORM 4-277
DC1FRAC.A 4-287
DDC.F.0 4-283
DDC.F.E 4-283
DDC.KF.E 4-286
DDC.KF.O 4-286
DDC.KR.E 4-286
DDC.KR.O 4-286
DDC.N.D 4-275
DDC.N.I 4-275, 4-276
DDC.T.0 4-283
DDC.T.E 4-283
DDCF.D.N 4-283, 4-284
DDCF.I.N 4-283, 4-285
DDCF.N.N 4-283
DDCFRACA 4-286
DDCFRACC 4-286
DDCR.I.N 4-284, 4-285
DDCR.N.N 4-283
DDCS.0 4-284
DDCS.E 4-284
DDCS.N 4-284
DDCTOT.A 4-288
DI.FAC 4-290
DI.PAIR.E 3-25
DIC.0 3-22, 4-280
DIC.E 3-22, 4-280
DIFFUSE 4-277
DIM.0 4-278
DIM.E 4-278
DIMM.0 4-278
DIMM.E 4-279
DIP.0 4-278
DIP.E 4-278
DIPAIR.0 3-25, 4-279
DIPAIR.E 4-279
DISP.FAC 3-143, 4-295
DISP.THR 4-296
DIV.ANG 4-295
DIX.0 3-112, 4-278
DIX.E 3-112, 4-278

DONOR 4-276
DV.FAC 4-290
DVC.0 3-22, 4-280
DVC.E 3-22, 4-280
DVM.0 4-279
DVM.E 4-279
DVMM.0 4-279
DVMM.E 4-279
DVP.0 4-279
DVP.E 4-279
DVPAIR.0 3-25, 4-279
DVPAIR.E 3-25, 4-279
DVX.0 4-279
DVX.E 4-279
E.I.S 4-280
E.IP.V 4-281
E.V.S 4-280
E.VP.I 4-281
ES.100 3-141, 4-294
ES.110 3-141, 4-294
ES.BREAK 3-139, 4-294
ES.F.100 3-141, 4-294
ES.F.110 3-141, 4-294
ES.F.H 3-139, 4-294
ES.F.RAN 3-139, 4-293
ES.RAND 3-139, 4-293
FGB 3-170, 4-278, 4-288
FVELB.0 4-289
FVELB.E 4-289
GSEG.0 4-289
GSEG.E 4-289
GSEG.INI 3-164, 4-289
I.DDC 4-286
IFRACM 4-284, 4-285
IFRACS 4-284
IMP.GB 4-275
IMP.IT 4-275
IMPL.TAB 4-276
IMPURITY 4-274
IP.DDC 4-286
IP.FRACA 4-281
IP.V.HIGH 4-281
IP.V.LOW 4-281
IP.V.MED 4-281
KSIP.0 4-292
KSIP.E 4-293
KSVP.0 4-293
KSVP.E 4-293
LOC.FAC 3-139, 4-295
MATERIAL 3-37, 4-277
MAX.AMOR 4-288
MOBILE 4-290
MODEL 4-289
NEGATIVE 4-276
NEW 4-274
NLOC.EXP 3-139, 4-294, 4-295
NLOC.K 3-139, 4-294, 4-295
NLOC.MAX 3-139, 4-294
NLOC.PRE 4-294, 4-295
NLOC.PREN 3-139
PD.PAIR 4-276
Q.INI.0 3-58, 4-292
Q.INI.E 3-58, 4-292
Q.MAX.0 3-58, 4-292
Q.MAX.E 3-58, 4-292
Q.SITES 3-164, 4-288
R.I.S 4-280
R.IP.V 4-280
R.V.S 4-280
R.VP.I 4-281
RATIO.0 3-58, 4-291
RATIO.E 3-58, 4-291
REC.FRAC 4-296
S.FRACA 4-282
SEG.0 3-55, 4-290
SEG.CPOW 4-293
SEG.E 3-55, 4-290
SEG.EQ2 3-58, 4-292
SEG.EQ3 4-292
SEG.IFRA 4-293
SEG.IPOW 4-293
SEG.O 3-58
SEG.SEGSS 3-58
SEG.SS 4-291
SEG.VPOW 4-293
SOLVE 4-276
SS.CLEAR 3-36, 4-282
SS.CONC 3-36, 4-282
SS.TEMP 3-36, 4-282
STEADY 3-112, 4-276
STS.VOL 4-278
SUR.RAT 4-295
T.ACT.0 3-37, 4-277
T.ACT.E 3-37
TIF.NAME 4-275
TRANS.0 3-55, 4-290
TRANS.E 3-55, 4-291
TWO.PHAS 3-58, 4-292
VD 4-277, 4-278
VELIF.0 4-289

- **VELIF.E** 4-289
- **VP.FRACA** 4-282
- **VP.I.HIG** 4-281
- **VP.I.LOW** 4-281
- **VP.I.MED** 4-281

parameters, **INITIALIZE** statement
- **<100>** 4-67
- **<110>** 4-67
- **<111>** 4-67
- **ANTIMONY** 4-68
- **ARSENIC** 4-68
- **BORON** 4-68, 5-20
- **CONCENTR** 4-68
- **DX** 3-5, 3-6, 4-67
- **FLIP.Y** 4-66
- **I.CONC** 4-68, 4-100, 4-133, 4-137
- **I.RESIST** 4-68
- **IMPURITY** 4-68, 4-100, 4-133, 4-137
- **IN.FILE** 3-7, 4-66, 5-11
- **INTERMED** 4-66
- **KEEP** 4-67
- **LINE.DAT** 4-67
- **MATERIAL** 4-68
- **NEW.MATE** 4-66
- **ORIENTAT** 4-67
- **PHOSPHOR** 4-68
- **RATIO** 3-4, 4-67
- **RESISTIV** 4-68
- **ROT.SUB** 3-141, 4-67
- **SCALE** 4-66
- **TEMPERAT** 4-67
- **TIF** 4-66
- **WIDTH** 3-5, 3-6, 4-67, 5-20
- **X.ORIENT** 3-141, 4-67
- **ZIP.FILE** 4-66

parameters, **INTERMEDIATE** statement
- **DEFAULT** 4-30
- **DELETE** 4-30, 4-31
- **EXPRESS** 4-30
- **MININDEX** 4-30
- **NAME** 4-30
- **PRINT** 4-30
- **SIZE** 4-30

parameters, **INTERSTITIAL** statement
- **/AMBIENT** 4-322
- **/MATERIA** 4-321
- **/NITRIDE** 4-321
- **/OXIDE** 4-321
- **/OXYNITR** 4-321
- **/POLYSIL** 4-321
- **/SILICON** 4-321
- **A.0** 3-70, 4-323
- **A.E** 3-70, 4-323
- **AMBIENT** 4-312, 4-320
- **C.ALL** 4-315
- **C.STATE** 3-60, 4-315
- **C.V** 4-315
- **CEQUIL.0** 3-59, 4-313
- **CEQUIL.E** 3-59, 4-313
- **CEQUIL.F** 4-313
- **CL.ALPHA** 4-319
- **CL.CF** 3-73, 4-318
- **CL.CR** 3-73, 4-318
- **CL.CSL.0** 4-319
- **CL.CSL.E** 4-319
- **CL.CSS.0** 4-319
- **CL.CSS.E** 4-319
- **CL.IFC** 3-73, 4-318
- **CL.IFI** 3-73, 4-317
- **CL.ISFC** 3-73, 4-318
- **CL.ISFI** 3-73, 4-317
- **CL.K0** 4-318
- **CL.K1** 4-318
- **CL.K2** 4-318
- **CL.KFC.0** 3-73, 4-317
- **CL.KFC.E** 3-73, 4-317
- **CL.KFCI** 4-318
- **CL.KFI.0** 3-73, 4-317
- **CL.KFI.E** 3-73, 4-317
- **CL.KR.0** 3-73, 4-318
- **CL.KR.E** 3-73, 4-318
- **CL.LAM0** 4-319
- **CL.LAM1** 4-319
- **CL.MODEL** 4-314
- **CL.NCRIT** 4-319
- **CL.NONE** 4-314
- **CM.SEC** 4-324
- **D.0** 3-63, 4-312, 4-320
- **D.E** 3-63, 4-312
- **D.F11** 4-312
- **D.F22** 4-312
- **D.FACTOR** 4-312
- **DC.0** 4-315
- **DC.E** 4-315
- **DNEG.0** 4-314
- **DNEG.E** 4-314
- **DPOS.0** 4-315
- **DPOS.E** 4-315
- **ECLUST.0** 4-316
- **ECLUST.E** 4-316

- **ECLUST.N** 4-321
- **F.TRAP.0** 3-71, 4-317
- **F.TRAP.E** 3-71, 4-317
- **FRAC.0** 4-315
- **FRAC.E** 4-316
- **GPOW.0** 3-69, 4-324
- **GPOW.E** 3-69, 4-324
- **GROWTH** 3-68, 4-323
- **K.TRAP** 3-71
- **K.TRAP.0** 4-317
- **K.TRAP.E** 3-71, 4-317
- **KB.0** 3-65, 4-313
- **KB.E** 3-65, 4-313
- **KB.HIGH** 3-66, 4-313
- **KB.LOW** 3-66, 4-313
- **KB.MED** 3-66, 4-313
- **KCV.0** 4-316
- **KCV.E** 4-316
- **KIV.0** 4-316
- **KIV.E** 4-316
- **KIV.NORM** 4-313
- **KLOOP.0** 3-82, 4-321
- **KLOOP.E** 3-82, 4-321
- **KPOW.0** 3-67, 4-324
- **KPOW.E** 3-67, 4-324
- **KSRAT.0** 3-67, 4-322
- **KSRAT.E** 3-67, 4-323
- **KSURF.0** 3-67, 4-322
- **KSURF.E** 3-67, 4-322
- **KSVEL.0** 3-67, 4-322
- **KSVEL.E** 3-67, 4-322
- **MATERIAL** 4-211, 4-312
- **N.MAX.0** 4-324
- **N.MAX.E** 4-324
- **NEG.0** 4-314
- **NEG.E** 4-314
- **NEU.0** 4-314
- **NEU.E** 4-314
- **NITRIDE** 4-312
- **OXIDE** 4-312
- **OXYNITRI** 4-312
- **POLYSILI** 4-312, 4-320
- **POS.0** 4-314
- **POS.E** 4-315
- **RLMIN** 4-321
- **SILICON** 4-312, 4-319, 4-320
- **T.EQPAIR** 4-211
- **T0.0** 3-70, 4-323
- **T0.E** 3-70, 4-324
- **THETA.0** 3-69, 4-323
- **THETA.E** 3-69, 4-323
- **TRAP.CON** 3-71, 4-316
- **V.INITOX** 3-67, 4-322
- **V.MAXOX** 3-67, 4-322
- **V.NORM** 3-67, 4-322
- **VCEQUIL** 4-313
- **VD** 4-313
- **VMOLE** 3-69, 4-314
- **VNORM.0** 3-68, 4-323
- **VNORM.E** 3-68, 4-323

parameters, **L.MODIFY** statement
- **BREAK** 4-20
- **NEXT** 4-20
- **STEPS** 4-20

parameters, **LABEL** statement
- **C.LINE** 4-176
- **C.RECTAN** 4-177
- **C.SYMBOL** 4-177
- **CENTER** 4-176, 6-39
- **CM** 4-175, 5-8
- **COLOR** 4-176
- **H.RECTAN** 4-177, 5-19
- **LABEL** 4-176
- **LEFT** 4-176
- **LENGTH** 4-176
- **LINE.TYP** 4-176
- **RECTANGL** 4-177
- **RIGHT** 4-176, 6-39
- **SIZE** 4-176, 5-19
- **SYMBOL** 4-177
- **W.RECTAN** 4-177, 5-19
- **X** 4-175, 5-8
- **X.CLICK** 4-175
- **Y** 4-175, 5-8
- **Y.CLICK** 4-175

parameters, **LINE** statement
- **LOCATION** 3-4, 4-57
- **SPACING** 3-4, 4-57, 5-14
- **TAG** 4-57, 6-50
- **X** 4-57, 5-14
- **Y** 4-57, 5-14

parameters, **LOADFILE** statement
- **FLIP.Y** 4-71
- **IN.FILE** 4-71
- **INTERMED** 4-71
- **NEW.MATE** 4-71
- **REGION** 4-71
- **SCALE** 4-71
- **TIF** 4-71
- **ZIP.FILE** 4-71

parameters, **LOOP/L.END** statement
  DNORM 4-16
  DSSQ 4-16, 4-220
  INDEX 4-16
  OPTIMIZE 4-16, 4-220
  PLOT 4-16
  START 4-16
  STEPS 4-16, 4-219
parameters, **MASK** statement
  DX.MAX 3-6
  G.EXTENT 4-84
  GRID 3-6, 4-84
  IN.FILE 4-84
  PRINT 4-84, 6-4
  SCALE 4-84
parameters, **MATERIAL** statement
  "CL.IMI.A 3-38
  AFFINITY 3-173, 4-262
  ALPHA 4-266
  ALUMINUM 3-121, 4-259
  AMBIENT 4-259
  AMOR.THR 4-269
  ANISOEL 4-261
  AT.NUM 3-112, 4-268
  AT.WT 3-112, 4-268
  BANDGAP 3-172, 4-262
  BOLTZMAN 3-171, 4-263
  C11.0 4-262
  C11.1 4-262
  C12.0 4-262
  C12.1 4-262
  C44.0 4-262
  C44.1 4-262
  CONDUCTO 4-263
  D.PCONST 4-264, 4-265
  DAM.GRAD 4-264
  DEBYE.T 4-269
  DENSITY 3-110, 4-260
  DLGX.0 3-168, 4-267
  DLGX.E 3-168, 4-267
  DSIM.0 4-266
  DSIM.E 3-166, 4-266
  DSIMM.0 3-166, 4-266
  DSIMM.E 3-166, 4-266
  DSIP.0 3-166, 4-266
  DSIP.E 3-166, 4-266
  DSIX.0 3-166, 4-266
  DSIX.E 3-166, 4-266
  DY.DEFAU 4-259
  E.ACCEP 3-173, 4-263
  E.DONOR 3-173, 4-263
  E.FIELD 4-260
  E.SEG 3-164
  EAVEL.0 3-168, 4-267
  EAVEL.E 3-168, 4-267
  EGALPH 4-262
  EGBETA 4-262
  EPSILON 3-171, 4-260
  F11 3-163, 4-265
  F22 3-163, 4-265
  FRAC.TA 3-165, 4-265
  G.ACCEP 3-173, 4-263
  G.DENS 3-163, 4-265
  G.DONOR 3-173, 4-263
  GAMMA.0 3-166, 4-266
  GAMMA.E 3-166, 4-266
  GB.THICK 4-267, 4-268
  GBE.0 3-167, 4-267
  GBE.1 3-167, 4-267
  GBE.H 3-167, 4-267
  GEOM 3-166, 4-266
  GRASZ.0 3-165, 4-265
  GRASZ.E 3-165, 4-265
  GSEG.0 3-164
  IMPL.TAB 3-123, 4-259
  INTRIN.S 3-116, 4-261
  ION.PAIR 3-20, 4-260
  IONIZATI 3-172, 4-263
  IP.OMEGA 3-20, 4-260
  LCTE 3-115, 4-261
  MATERIAL 4-259
  MAX.DAMA 4-264
  MD.INDEX 4-259
  MIN.GRAI 3-165, 4-265
  MOL.WT 3-110, 4-268
  N.CONDUC 3-172, 4-262
  N.VALENC 3-172, 4-263
  NC.F 4-263
  NEW 4-259
  NI.0 3-21, 4-260
  NI.E 3-21, 4-260
  NI.F 3-21, 4-260
  NITRIDE 3-121, 4-259
  NSEG 3-167, 4-267
  NV.F 4-263
  OXIDE 3-121, 4-259
  OXYNITRI 4-259
  PHOTORES 3-121, 4-259
  POISS.R 3-97, 4-261
  POLY.FAC 4-269

- **POLYCRYS** 3-170, 4-265, 4-269, 4-270, 6-7
- **POLYSILI** 3-121, 4-259
- **QM.BETA** 3-176, 4-263
- **QM.YCRIT** 3-176, 4-263
- **SEMICOND** 4-262
- **SILICON** 3-121, 4-259
- **STRENGTH** 4-261
- **SURF.DIS** 4-269
- **SURF.TEN** 3-104, 4-261
- **TBU.0** 3-168, 4-267
- **TBU.E** 3-168, 4-267
- **TEMP.BRE** 3-165, 4-265
- **TEMPERAT** 4-261
- **TIF.NAME** 4-259
- **TOXIDE** 3-168, 4-267
- **VC** 4-261, 6-29
- **VELIF.0** 3-164
- **VELIF.E** 3-164
- **VISC.0** 3-98, 4-260
- **VISC.E** 3-98, 4-261
- **VISC.X** 3-98, 4-261
- **WORKFUNC** 3-173, 4-263
- **YOUNG.M** 3-97, 4-261

parameters, **MESH** statement
- **DX.MAX** 3-6, 4-53
- **DX.MIN** 3-6, 4-53
- **DX.RATIO** 3-6, 4-53
- **DY.ACTIV** 3-7, 4-54
- **DY.BOT** 3-7, 4-54
- **DY.RATIO** 3-7, 4-54
- **DY.RATION** 3-7
- **DY.SURF** 3-7, 4-53, 6-34
- **ERR.FAC** 6-4
- **FAST** 3-13, 4-54
- **GRID.FAC** 3-3, 4-53, 5-14, 6-3, 6-34
- **LY.ACTIV** 3-6, 3-7, 4-53, 6-34
- **LY.BOT** 3-7, 4-54
- **LY.SURF** 3-6, 4-53, 6-34

parameters, **METHOD** statement
- **ABS.ADAP** 3-11, 4-216
- **ABS.ERR** 3-18, 4-215
- **ACT.EQUI** 3-54, 4-212
- **ACT.FULL** 4-212
- **ACT.TRAN** 4-212
- **ADVANCED** 4-219
- **ALUMINUM** 4-216
- **ANTIMONY** 4-214
- **ARSENIC** 4-214
- **BACK** 4-213
- **BACKSCAT** 4-220
- **BLK.ITLI** 4-213
- **BORON** 4-214
- **CG** 4-212
- **CO.ANALY** 4-219
- **CO.HYBRI** 4-219
- **CO.NUMR** 4-219
- **COMPRESS** 3-96, 4-208
- **CONTIN.M** 4-217
- **DAN,TABL** 4-220
- **DEPO.MIS** 4-209
- **DIF.ADAP** 3-12, 4-211
- **DIFF.MIS** 4-209
- **DY.EXACT** 4-209
- **DY.LOCAL** 4-209
- **DY.OXIDE** 3-9, 4-209, 5-15
- **E.AVCAP** 4-217
- **E.DSURF** 4-218
- **E.ITMAX** 3-173, 4-217
- **E.ITMIN** 3-173, 4-217
- **E.REGRID** 3-171, 4-218
- **E.RELERR** 3-173, 4-217
- **E.RSURF** 4-218
- **E.RVCAP** 4-217
- **E.TSURF** 4-218
- **E.USEAVC** 4-218
- **ENABLE** 4-219
- **ERF1** 3-90, 4-207
- **ERF2** 3-90, 4-207
- **ERFC** 3-87, 4-207
- **ERFG** 3-87, 4-208
- **ERR.FAC** 3-11, 4-212, 5-4
- **ERROR** 4-215
- **FT.FULL** 4-211
- **FULL** 4-215
- **GRID.OXI** 3-9, 4-209
- **II.TOL.DY** 4-219
- **II.TOL.F** 4-219
- **IMP.ADAP** 3-12, 4-211
- **IMPURITY** 4-214
- **INIT.TIM** 3-17, 4-212
- **INTERSTI** 4-214
- **IT.ACT** 3-56, 4-218
- **IT.CPL** 3-56, 4-218
- **IT.STEAD** 3-57, 4-218
- **IT.THERM** 3-57, 4-218
- **IT.ZERO** 3-57, 4-218
- **ITRAP** 3-55, 4-218
- **KO.FULL** 4-211, 4-213
- **MATERIAL** 4-216
- **MF.DIST** 4-213, 4-214

- **MF.METH** 4-213
- **MF.P.LEV** 4-214
- **MIN.FILL** 4-213
- **MIN.FREQ** 4-213
- **MIN.SPAC** 3-11, 3-12, 4-216, 4-217
- **MOB.AROR** 3-175, 4-218
- **MOB.CAUG** 3-175, 4-218
- **MOB.TABL** 3-174, 4-218
- **MODEL** 4-219
- **NEWTON** 4-215
- **NITRIDE** 4-216
- **NONE** 4-215
- **NSTREAMS** 3-24, 4-210
- **OX.ADAP** 3-12
- **OX.ADAPT** 4-212
- **OX.REL** 4-217
- **OXIDANT** 4-215
- **OXIDE** 4-216
- **OXYNITRI** 4-216
- **PAIR.GRA** 3-34, 4-210
- **PAIR.REC** 3-24, 4-210
- **PAIR.SAT** 3-24, 4-210
- **PART** 4-215
- **PD.FERMI** 3-111, 4-210
- **PD.FULL** 3-24, 4-210
- **PD.NREC** 4-211
- **PD.PFLUX** 3-34, 4-211
- **PD.PREC** 3-59, 4-211
- **PD.PTIME** 3-59, 4-211
- **PD.TRANS** 3-24, 4-210
- **PD.TRANS** 5-20
- **PHOSPHOR** 4-214
- **PHOTORES** 4-216
- **POISSON** 4-219
- **POLYSILI** 4-216
- **PRINT** 4-220
- **REL.ADAP** 3-11, 4-216
- **REL.ERR** 3-18, 4-215
- **SILICON** 4-216
- **SKIP.SIL** 3-101, 4-209
- **SPLU** 4-212
- **ST.CG1** 4-208
- **ST.CG2** 4-208
- **ST.DIFF** 4-208
- **ST.DIREC** 4-208
- **ST.HISTO** 3-98, 4-208, 4-209
- **ST.LEVEL** 4-208
- **ST.PDEQ** 4-209
- **STN.MIS** 4-209
- **SYMMETRY** 4-215
- **TIME** 3-17
- **TIME.STE** 4-215
- **TRAP** 4-215
- **VACANCY** 4-214
- **VE.SMOOT** 3-102, 4-217
- **VERTICAL** 3-95, 4-208
- **VISCOELA** 3-99, 4-208, 5-15
- **VISCOUS** 3-98, 4-208

parameters, **MOBILITY** statement
- **ALPHAN** 4-308
- **ALPHAP** 4-308
- **AN** 4-306
- **AP** 4-307
- **CN** 4-306
- **CP** 4-307
- **ECN.MU** 3-174, 4-306
- **ECP.MU** 3-174, 4-306
- **EXN** 4-306
- **EXN2** 4-306
- **EXN3** 4-306
- **EXN4** 4-307
- **EXP1** 4-307
- **EXP2** 4-307
- **EXP3** 4-307
- **EXP4** 4-307
- **GSURFN** 3-174, 4-306
- **GSURFP** 3-174, 4-306
- **KELVIN** 4-305
- **MUN.MAX** 4-307
- **MUN.MIN** 4-307
- **MUN1** 4-306
- **MUN2** 4-306
- **MUP.MAX** 4-308
- **MUP.MIN** 4-308
- **MUP1** 4-307
- **MUP2** 4-307
- **NREFN** 4-307
- **NREFP** 4-308
- **NUN** 4-308
- **NUP** 4-308
- **TAB.CLEA** 4-305
- **TAB.CONC** 4-305
- **TAB.E.MU** 4-305
- **TAB.H.MU** 4-305
- **TAB.TEMP** 4-305
- **XIN** 4-308
- **XIP** 4-308

parameters, **MOMENT** statement
- **ALL.MATE** 4-251
- **CLEAR** 4-251

- **D.FRAC** 4-252
- **D.GAMMA** 4-253
- **D.KURTOS** 4-253
- **D.LSIGMA** 4-253
- **D.LSLOPE** 4-253
- **D.RANGE** 4-252
- **D.SIGMA** 4-252
- **GAMMA** 4-252
- **KURTOSIS** 4-252
- **LSIGMA** 4-252
- **LSLOPE** 4-252
- **MATERIAL** 4-251
- **NITRIDE** 4-251
- **OXIDE** 4-251
- **OXYNITRI** 4-251
- **PHOTORES** 4-251
- **POLYSILI** 4-251
- **RANGE** 4-251
- **SIGMA** 4-252
- **SILICON** 4-251

parameters, **OPTION** statement
- **DEBUG** 4-41
- **DEVICE** 4-41, 5-7
- **DIAGNOST** 4-41
- **ECHO** 4-41
- **EXECUTE** 4-42
- **INFORMAT** 4-41
- **NORMAL** 4-41
- **PLOT.OUT** 4-41
- **QUIET** 4-41
- **V.COMPAT** 3-144, 4-42
- **VERBOSE** 4-41
- **ZIP.EXE** 4-42
- **ZIP.FILE** 4-42
- **ZIP.SUFF** 4-42

parameters, **PHOSPHORUS** statement
- **/AMBIENT** 4-362
- **/MATERIA** 4-361
- **/NITRIDE** 4-362
- **/OXIDE** 4-362
- **/OXYNITR** 4-362
- **/POLYSIL** 4-362
- **/SILICON** 4-362
- **AMBIENT** 4-359
- **CM.SEC** 4-363
- **DIM.0** 3-22, 4-359
- **DIM.E** 3-22, 4-359
- **DIMM.0** 3-22, 4-359
- **DIMM.E** 4-359
- **DIP.0** 3-22
- **DIP.E** 3-22
- **DIPAIR.0** 3-25, 4-360
- **DIPAIR.E** 3-25, 4-360
- **DIX.0** 3-22, 4-359
- **DIX.E** 3-22, 4-359
- **DVM.0** 3-22, 4-359
- **DVM.E** 3-22, 4-359
- **DVMM.0** 3-22, 4-360
- **DVMM.E** 3-22, 4-360
- **DVP.0** 3-22
- **DVP.E** 3-22
- **DVPAIR.0** 3-25, 4-360
- **DVPAIR.E** 3-25, 4-360
- **DVX.0** 3-22, 4-359
- **DVX.E** 3-22, 4-359
- **E.I.S** 3-32, 4-360
- **E.IP.V** 3-32, 4-361
- **E.V.S** 3-32, 4-360
- **E.VP.I** 3-32, 4-361
- **ES.100** 4-363
- **ES.110** 4-363
- **ES.F.100** 4-363
- **ES.F.110** 4-363
- **ES.F.RAN** 4-363
- **ES.RAND** 4-363
- **IP.V.HIG** 4-361
- **IP.V.LOW** 4-361
- **IP.V.MED** 4-361
- **KSIP.0** 4-362
- **KSIP.E** 4-362
- **KSVP.0** 4-363
- **KSVP.E** 4-363
- **MATERIAL** 4-358
- **NITRIDE** 4-358
- **OXIDE** 4-358
- **OXYNITRI** 4-358
- **POLYSILI** 4-358
- **R.I.S** 3-32, 4-360
- **R.IP.V** 3-32, 4-360
- **R.V.S** 3-32, 4-360
- **R.VP.I** 3-32, 4-361
- **SEG.0** 4-362
- **SEG.E** 4-362
- **SILICON** 4-358
- **SS.CLEAR** 4-361
- **SS.CONC** 4-361
- **SS.TEMP** 4-361
- **TRANS.0** 4-362
- **TRANS.E** 4-362
- **VP.I.HIG** 4-361

VP.I.LOW 4-361
VP.I.MED 4-361
parameters, **PLOT.1D** statement
  /ALUMINU 4-157
  /AMBIENT 4-157
  /MATERIA 4-156
  /NITRIDE 4-156
  /OXIDE 4-156
  /OXYNITR 4-156
  /PHOTORE 4-157
  /POLYSIL 4-156
  /REFLECT 4-157
  /SILICON 4-156
  ALUMINUM 4-156
  AXES 4-159, 5-7
  BOTTOM 4-160, 5-25
  BOUNDARY 4-158
  CLEAR 4-159, 5-7
  COLOR 4-159, 5-7
  COLUMN 4-157
  CURVE 4-159
  ELECTRIC 4-158
  IN.FILE 4-157
  LEFT 4-160
  LINE.TYP 4-159
  LINE.TYPE 5-7
  MATERIAL 4-155
  NITRIDE 4-156
  OXIDE 4-156
  OXYNITRI 4-156
  PHOTORES 4-156
  POLYSILI 4-156
  RIGHT 4-160, 5-25
  ROW 4-157
  SILICON 4-156
  SYMBOL 4-159
  T.SIZE 4-160
  TIF 4-157
  TOP 4-160
  X.AXIS 4-157
  X.COLUMN 4-157
  X.LABEL 4-158
  X.LENGTH 4-160
  X.LOG 4-158
  X.OFFSET 4-160
  X.ROW 4-158
  X.SCALE 4-158
  X.SHIFT 4-158
  X.SIZE 4-160
  X.VALUE 4-155, 5-7
  Y.AXIS 4-157
  Y.COLUMN 4-157
  Y.LABEL 4-158
  Y.LENGTH 4-160
  Y.LOG 4-158
  Y.OFFSET 4-160
  Y.ROW 4-158
  Y.SCALE 4-158
  Y.SHIFT 4-158
  Y.SIZE 4-160
  Y.VALUE 4-155, 5-7
parameters, **PLOT.2D** statement
  AXES 4-164, 5-19
  BOUNDARY 4-164
  C.BOUND 4-164
  C.COMPRE 4-165, 5-17
  C.GRID 4-164, 5-15
  C.TENSIO 4-165, 5-17
  CLEAR 4-163, 5-27
  DIAMONDS 4-165
  FLOW 4-164, 5-16
  GRID 4-164, 5-15
  L.BOUND 4-164
  L.COMPRE 4-165
  L.GRID 4-164
  L.TENSIO 4-165
  SCALE 4-163, 5-15
  STRESS 4-164, 5-17
  T.SIZE 4-166
  VLENG 4-164, 5-16
  VMAX 4-164
  X.LENGTH 4-165
  X.MAX 4-163, 5-16
  X.MIN 4-163, 5-16
  X.OFFSET 4-165
  X.SIZE 4-165
  Y.LENGTH 4-165
  Y.MAX 4-163, 5-16, 6-35
  Y.MIN 4-163, 5-16
  Y.OFFSET 4-165
  Y.SIZE 4-166
parameters, **PLOT.3D** statement
  BOUNDARY 4-173
  C.BOUND 4-173
  CLEAR 4-172
  COLOR 4-173
  L.BOUND 4-173
  LINE.TYP 4-173
  NUM.CNTR 4-173
  PHI 4-172

- **THETA** 4-172
- **X.MAX** 4-172
- **X.MIN** 4-172
- **Y.MAX** 4-172
- **Y.MIN** 4-172
- **Z.MAX** 4-173
- **Z.MIN** 4-173

parameters, **PRINT.1D** statement
- **/ALUMINU** 4-152
- **/AMBIENT** 4-152
- **/MATERIA** 4-152
- **/NITRIDE** 4-152
- **/OXIDE** 4-152
- **/OXYNITR** 4-152
- **/PHOTORE** 4-152
- **/POLYSIL** 4-152
- **/REFLECT** 4-152
- **/SILICON** 4-152
- **ALUMINUM** 4-152
- **APPEND** 4-152
- **CM** 4-152
- **LAYERS** 4-153, 5-9, 6-45
- **MATERIAL** 4-151
- **NITRIDE** 4-151
- **NM** 4-152
- **OUTFILE** 4-152
- **OXIDE** 4-151
- **OXYNITRI** 4-151
- **PHOTORES** 4-151
- **POLYSILI** 4-151
- **SILICON** 4-151
- **SPOT** 4-153, 5-33
- **UM** 4-152
- **X.MAX** 4-153
- **X.MIN** 4-153
- **X.VALUE** 4-151
- **Y.VALUE** 4-151

parameters, **PROFILE** statement
- **ANTIMONY** 4-86
- **ARSENIC** 4-86
- **BORON** 4-86
- **CM** 4-87
- **I.COLUMN** 4-88
- **I.SCALE** 4-87
- **I.SOURCE** 4-88
- **IMPURITY** 4-86
- **IN.FILE** 4-87
- **MIN.WT** 4-88
- **NM** 4-87
- **OFFSET** 4-88
- **ONE.DIM** 4-87
- **PHOSPHOR** 4-86
- **REPLACE** 4-87
- **SIGMA** 4-88
- **TIF** 4-88
- **TS4** 4-88
- **TWO.DIM** 4-88
- **UM** 4-87
- **X.MAX** 4-87
- **X.MIN** 4-87
- **X.OFFSET** 4-88
- **Y.MAX** 4-87
- **Y.MIN** 4-87
- **Y.OFFSET** 4-88
- **ZIP.FILE** 4-87

parameters, **REACTION** statement
- **/E.FAC** 4-302
- **/EI.L** 4-301
- **/EI.R** 4-301
- **/IMP.L** 4-301
- **/IMP.R** 4-301
- **/MAT.L** 4-300
- **/NI.L** 4-301
- **/NI.R** 4-301
- **/NM.L** 4-301
- **/R.FAC** 4-302
- **DELETE** 4-300
- **E.FAC** 4-302
- **EI.L** 4-301
- **EI.R** 4-301
- **EQUIL.0** 4-302
- **EQUIL.E** 4-302
- **IMP.L** 4-300
- **IMP.R** 4-301
- **MAT.BULK** 4-300
- **MAT.NEW** 4-302
- **MAT.R** 4-300
- **MODEL** 4-300
- **NAME** 4-300
- **NI.L** 4-300
- **NI.R** 4-301
- **NM.R** 4-301
- **R.FAC** 4-302
- **RATE.0** 4-301
- **RATE.E** 4-302
- **REPLACE** 4-300
- **THICKNES** 4-302

parameters, **REGION** statement
- **ALUMINUM** 4-64
- **KEEP** 4-65

- **MATERIAL** 4-64
- **NEW.MATE** 4-65
- **NITRIDE** 4-64
- **OXIDE** 4-64
- **OXYNITRI** 4-64
- **PHOTORES** 4-64
- **POLYSILI** 4-64
- **SILICON** 4-64
- **X** 4-65
- **XHI** 4-64
- **XLO** 4-64
- **Y** 4-65
- **YHI** 4-64
- **YLO** 4-64

parameters, **SAVEFILE** statement
- **ACTIVE** 4-76
- **CHEMICAL** 4-76
- **DEFECT** 4-76
- **DX.MIN** 4-76
- **DY.MIN** 4-76
- **ELEC.BOT** 4-75
- **FLIP.Y** 4-74
- **FULL.DEV** 4-75
- **HALF.DEV** 4-75
- **INTERMED** 4-74
- **MEDICI** 4-75
- **MINIMOS5** 4-75
- **MISC** 4-77
- **OUT.FILE** 4-74
- **OXID** 4-77
- **POLY.ELE** 4-75
- **SCALE** 4-74
- **TEMPERAT** 4-74
- **TIF** 4-75
- **TIF.VERS** 4-75
- **WAVE** 4-76
- **X.CHANNE** 4-76
- **X.MASK.D** 4-75
- **X.MASK.S** 4-75
- **X.MAX** 4-76
- **X.MIN** 4-76
- **Y.MAX** 4-76
- **Y.MIN** 4-76
- **ZIP.FILE** 4-74

parameters, **SELECT** statement
- **LABEL** 4-146
- **TEMPERAT** 4-146
- **TITLE** 4-146
- **Z** 4-146, 5-8, 6-30

parameters, **STRESS** statement
- **TEMP2** 4-142

parameters, **STRUCTURE** statement
- **BOTTOM** 4-79
- **COLUMNS** 4-80
- **DX** 3-8, 4-80
- **EXTEND** 3-8, 4-80
- **LEFT** 4-79
- **REFLECT** 4-80, 6-19
- **REPEAT** 4-80
- **RIGHT** 4-79
- **ROWS** 4-80
- **SPACES** 4-80
- **TEMPERAT** 4-81
- **TOP** 4-79
- **TRUNCATE** 4-79, 6-35
- **UNREFINE** 4-80
- **WIDTH** 3-8, 4-80
- **X** 4-79, 4-80, 4-81
- **XDX** 3-8, 4-80
- **Y** 4-79
- **Y.ELIM** 4-80

parameters, **VACANCY** statement
- **/AMBIENT** 4-331
- **/MATERIA** 4-331
- **/NITRIDE** 4-331
- **/OXIDE** 4-331
- **/OXYNITR** 4-331
- **/POLYSIL** 4-331
- **/SILICON** 4-331
- **A.0** 3-70, 4-333
- **A.E** 3-70, 4-333
- **AMBIENT** 4-327
- **C.ALL** 4-330
- **C.I** 4-330
- **C.STATE** 3-60, 4-330
- **CEQUIL.0** 3-59, 4-328
- **CEQUIL.E** 3-59, 4-328
- **CEQUIL.F** 4-328
- **CM.SEC** 4-334, 4-335
- **D.0** 3-63, 4-328
- **D.E** 3-63, 4-328
- **D.F11** 4-328
- **D.F22** 4-328
- **D.FACTOR** 4-328
- **DC.0** 4-330
- **DC.E** 4-330
- **DNEG.0** 4-329
- **DNEG.E** 4-329
- **DPOS.0** 4-329
- **DPOS.E** 4-330

  - **ECLUST.0** 4-331
  - **ECLUST.E** 4-331
  - **ECLUST.N** 4-331
  - **FRAC.0** 4-330
  - **FRAC.E** 4-330
  - **GPOW.0** 3-69, 4-334
  - **GPOW.E** 3-69, 4-334
  - **GROWTH** 3-68, 4-333
  - **K.SURF.0** 3-67
  - **KIC.0** 4-330
  - **KIC.E** 4-331
  - **KPOW.0** 3-67, 4-334
  - **KPOW.E** 3-67, 4-334
  - **KSRAT.0** 3-67, 4-332
  - **KSRAT.E** 3-67, 4-332
  - **KSURF.0** 4-332
  - **KSURF.E** 3-67, 4-332
  - **KSVEL.0** 3-67, 4-332
  - **KSVEL.E** 3-67, 4-332
  - **MATERIAL** 4-327
  - **NEG.0** 4-329
  - **NEG.E** 4-329
  - **NEU.0** 4-329
  - **NEU.E** 4-329
  - **NITRIDE** 4-327
  - **OXIDE** 4-327
  - **OXYNITRI** 4-327
  - **POLYSILI** 4-327
  - **POS.0** 4-329
  - **POS.E** 4-329
  - **SILICON** 4-327
  - **T0.0** 3-70, 4-333
  - **T0.E** 3-70, 4-333
  - **THETA.0** 3-69, 4-333
  - **THETA.E** 3-69, 4-333
  - **V.INITOX** 3-67, 4-332
  - **V.MAXOX** 3-67, 4-332
  - **V.NORM** 3-67, 4-332
  - **VCEQUIL** 4-329
  - **VD** 4-328
  - **VMOLE** 4-329
  - **VNORM.0** 3-68, 4-333
  - **VNORM.E** 3-68, 4-333
- parameters, **VIEWPORT** statement
  - **X.MAX** 4-204
  - **X.MIN** 4-204
  - **Y.MAX** 4-204
  - **Y.MIN** 4-204
- **PAUSE** 4-12
  - description 4-12
  - example 4-12
- **PD.FERMI** model 3-24
- **PD.FULL** model 3-24
- **PD.TRANS** model 3-24
- **PD.TRANS** parameter, **METHOD** statement
  - local oxidation example 5-20
- Pearson distribution 3-123
- **PHOSPHORUS** 4-358
  - description 4-363
  - examples 4-364
- photoresist
  - masking, exposure, and development 3-156
- physical functions 7-13
- physical model and equation interface 7-3
- plot device definition file *s4pcap* 2-10
- plot produced by
  - **PLOT.2D FLOW** statement input file *s4ex2a.inp Fig.* 5-17
  - **PLOT.2D STRESS** statement input file *s4ex2a.inp Fig.* 5-18
- **PLOT.1D** 4-155
  - AXES parameter, 1D bipolar example 5-7
  - CLEAR parameter, 1D bipolar example 5-7
  - COLOR parameter, 1D bipolar example 5-7
  - description 4-161
  - examples 4-162
  - **IN.FILE** parameter 4-161
  - line type and color 4-161
  - LINE.TYP parameter, 1D bipolar example 5-7
  - X.VALUE parameter, 1D bipolar example 5-7
  - Y.VALUE parameter, 1D bipolar example 5-7
- **PLOT.2D** 4-163
  - description 4-166
  - examples 4-166
  - line type and color 4-166
  - X.MIN parameter, local oxidation example 5-16
- **PLOT.3D** 4-172
  - additional notes 4-174
  - description 4-173
  - examples 4-174
  - line type and color 4-174
- plotting results, Monte Carlo method
  - boron contours 6-24
  - vertical profiles 6-25
- plotting the results 5-6
  - after implant 6-61
  - doping and grain size 6-63
  - doping vs. stripe width 6-63
  - labels 5-8

- **PLOT.1D** statement 5-7
- **SELECT** statement 5-7
- specifying a graphics device 5-7

point defect diffusion equations 3-61

point defect models example
- choosing 5-34
- creating the test structure 5-31
- **LABEL** statement 5-32
- **LOADFILE** statement 5-31
- overview 5-28
- oxidation and plotting of impurity profiles 5-31
- **PD.FERMI** model 5-28
- **PD.FULL** model 5-28
- **PD.TRANS** model 5-28
- point defect profiles 5-33

point defect parameters, default coefficients A-8

point defects
- diffusion of 3-58, 3-59
- enhancement 3-24
- injection and recombination at interfaces 3-66
- injection rate 3-68
- profiles plotted by *s4ex3.inp Fig.* 5-34
- reaction of pairs 3-64
- tutorial example of 5-20

Poisson's equation 3-171, 4-197, 4-217

Poisson's ratio 3-97, 4-246

poly-buffered LOCOS example
- **COMPRESS** model 6-29
- **METHOD** statement 6-29
- overview 6-28
- plotting results 6-30
- structure generation 6-28
- **VISCOEL** model 6-29
- **VISCOUS** model 6-29

POLYCRYS parameter, **MATERIAL** statement
- NMOS LDD process example 6-7

polycrystalline materials, modeling
- advanced example of 6-28
- epitaxial regrowth 3-167
- grain interior and boundaries, segregation 3-164
- grain size model 3-165
- interface oxide breakup 3-167
- overview 3-161
- using the model 3-58, 3-169

polysilicon
- enhancement 3-22, 3-23
- oxidation 3-104

polysilicon emitter study example
- overview 6-59
- **PD.FERMI** diffusion model 6-59
- **PD.FULL** model 6-59
- plotting results 6-61
- process simulation 6-59

polysilicon grain segregation, default coefficients A-7

polysilicon Monte Carlo implant model
- ion implantation into silicon carbide 3-152

PRINT parameter, **MASK** statement
- NMOS LDD process example 6-4

PRINT.1D 4-151
- description 4-153
- examples 4-154
- interface values 4-153
- layers 4-153
- LAYERS parameter
  - 1D bipolar example 5-9
  - DMOS power transistor example 6-45
  - point defect models example 5-33

printed output 2-2

printing layer information 5-8
- **PRINT.1D** statement 5-8
- using **PRINT.1D** layers 5-9

process simulation
- models 6-59
- processing 6-59

process steps, input statements
- **DEPOSITION** 4-99
- **DEVELOP** 4-108
- **DIFFUSION** 4-131
- **EPITAXY** 4-137
- **ETCH** 4-109
- **EXPOSE** 4-106
- **IMPLANT** 4-114
- **STRESS** 4-142

processing of SOI MOSFET, from input file *s4ex9.inp Fig.* 6-52

processing steps 1-2, 5-5
- buried layer 5-5
- buried layer masking oxide 5-5
- epitaxial layer 5-6
- pad oxide and nitride mask 5-6

processing the DMOS power transistor
- gate processing 6-45
- source processing 6-46

processing, CMOS process example 6-36

PROFILE 4-86
- description 4-88
- example 4-90
- IMPURITY parameter 4-89

- interpolation 4-89
- OFFSET parameter 4-89

profiles
- active and total arsenic concentration through the poly gate *Fig.* 6-40
- with **`PD.FERMI`** and **`PD.FULL`** models, from *s4ex3.inp Fig.* 5-32

program execution and output 2-1

Q

`QSS` parameter, **`ELECTRICAL`** statement
- NMOS LDD process example 6-14

quantum effect 3-175

quantum effect in MOS capacitance *Fig.* 4-202

quantum mechanical model for MOSFET 3-175

R

`REACTION` 4-300
- defining and deleting 4-303
- description 4-302
- effects 4-304
- insertion of native layers 4-303
- parameters 4-303
- reaction equation 4-303

reaction of pairs with point defects 3-64

reaction rate constants 3-31

references
- default coefficients A-29
- TSUPREM-4 models 3-178

`REFLECT` parameter, **`STRUCTURE`** statement
- trench implant simulation example 6-19

reflow 3-104, 4-136

`REGION` 4-64
- description 4-65
- example 4-65

related publications
- *Data Visualizer Programming Guide* 2-7
- *TCAD Products and Utilities Installation Manual* 1-iv

`RESISTAN` parameter, **`ELECTRICAL`** statement
- NMOS LDD process example 6-14

`RETURN` 4-9
- description 4-9
- example 4-10
- exiting interactive input mode 4-9
- returning from batch mode 4-9

`RIGHT` parameter, **`LABEL`** statement
- CMOS process example 6-39

`RIGHT` parameter, **`PLOT.1D`** statement
- local oxidation example 5-25

S

saved structure files
- Medici D-9
- MINIMOS 5 2-7
- older versions of TSUPREM-4 D-7
- overview 2-7, D-1
- TIF 2-7
- TSUPREM-4 D-1
- *wave* 2-7

`SAVEFILE` 4-74
- description 4-77
- examples 4-78
- Medici files 4-78
- MINIMOS 4-78
- Taurus-Lithography files 4-78
- temperature 4-78
- TIF files 4-77

saving the structure 5-6

`SCALE` parameter, **`PLOT.2D`** statement
- local oxidation example 5-15

Schrödinger equation 3-175

second part of input file
- *s4ex10.inp*, silicide growth *Fig.* 6-56
- *s4ex2a.inp*, showing statements plotting results of LOCOS process *Fig.* 5-16
- *s4ex2b.inp*, showing statements for plotting the results of diffusion simulation *Fig.* 5-23
- *s4ex3.inp*, using the full two-dimensional point defect model *Fig.* 5-30
- *s4ex4a.inp*, for simulating an NMOS process *Fig.* 6-6
- *s4ex4b.inp*, showing source/drain processing and metallization *Fig.* 6-10
- *s4ex4d.inp*, showing the MOS capacitance extraction *Fig.* 6-16
- *s4ex5.inp*, showing tilted implantation using analytic implant model *Fig.* 6-20
- *s4ex6.inp*, plotting final poly-buffered LOCOS structure *Fig.* 6-31
- *s4ex7a.inp*, showing statements for simulating CMOS process *Fig.* 6-33
- *s4ex8.inp*, processing of DMOS power transistor, through body diffusion *Fig.* 6-44

segregation and transport coefficients, default

        coefficients A-5
- segregation coefficient 3-55
- segregation flux 3-54
- **SELECT** 4-146
  - description 4-146
  - examples 4-149
  - mathematical operations and functions 4-148
  - solution values 4-146
  - **TEMPERAT** specified 4-149
- **SELECT** statement 5-7
  - **Z** expression 5-7
- selective epitaxy 3-154
- sidewall profiles
  - after Monte Carlo implant *Fig.* 6-28
  - produced by analytic implant *Fig.* 6-23
- SiGe model 7-38
- silicidation and oxidation 3-9
- silicide models
  - advanced example of 6-54
  - impurities and point defects 3-111
  - other silicides 3-114
  - overview 3-109
  - $TiSi_2$ growth kinetics 3-109
  - titanium silicide model 3-111
  - tungsten, cobalt, and nickel silicides 3-114
- simple grids, initial checkout 6-1
- simulation structure 1-2
  - advanced examples of 6-2, 6-19, 6-28
  - coordinates, overview 3-2
  - initial structure, restrictions 3-2
  - materials, overview 3-2
  - regions, overview 3-2
- **SIZE** parameter, **LABEL** statement
  - local oxidation example 5-19
- SOI MOSFET example
  - **COMPRESS** model 6-53
  - **INITIALIZE** statement 6-51
  - **LINE** statement 6-51
  - mesh generation 6-49
  - overview 6-49
  - process simulation 6-53
  - **REGION** statement 6-49
  - **VERTICAL** model 6-53
  - **VISCOEL** model 6-53
- solid solubility clustering model 3-57
- solid solubility model 3-36
- solid solubility tables 3-36
- solving method 7-3
- **SOURCE** 4-8
  - description 4-8
  - examples 4-8
  - generating templates 4-8
  - reusing combinations of statements 4-8
- **SPACES** parameter, **DEPOSITION** statement
  - SOI MOSFET example 6-50
- **SPACES** parameter, **EPITAXY** statement
  - 1D bipolar example 5-6
- **SPACING** parameter, **LINE** statement
  - 1D local oxidation example 5-14
- specifying silicide models and parameters
  - dopants 3-114
  - impurities 3-112
  - materials 3-111
  - reactions 3-112
- **SPOT** parameter, **PRINT.1D** statement
  - local oxidation example 5-33
- standard output file *s4out* 2-6
- statements
  - **AMBIENT** 4-233
  - **ANTIMONY** 4-337
  - **ARSENIC** 4-344
  - **ASSIGN** 4-23
  - **BORON** 4-351
  - **BOUNDARY** 4-62
  - **COLOR** 4-170
  - **COMMENT** 4-7
  - **CONTOUR** 4-168
  - **CPULOG** 4-50
  - **DEFINE** 4-46
  - **DEPOSITION** 4-99
  - **DEVELOP** 4-108
  - **DIFFUSION** 4-131
  - **ECHO** 4-40
  - **ELECTRICAL** 4-194
  - **ELECTRODE** 4-96
  - **ELIMINATE** 4-59
  - **EPITAXY** 4-137
  - **EQUATION** 4-230
  - **ETCH** 4-109
  - **EXPOSE** 4-106
  - **EXTRACT** 4-180
  - **FOREACH/END** 4-14
  - **HELP** 4-51
  - **IMPLANT** 4-114
  - **IMPURITY** 4-273
  - **INITIALIZE** 4-66
  - **INTERACTIVE** 4-11
  - **INTERMEDIATE** 4-30
  - **INTERSTITIAL** 4-311

- **LABEL** 4-175
- **LINE** 4-57
- **LOADFILE** 4-71
- **LOOP/L.END** 4-16
- **MASK** 4-84
- **MATERIAL** 4-258
- **MESH** 4-53
- **METHOD** 4-207
- **MOBILITY** 4-305
- **MOMENT** 4-251
- **OPTION** 4-41
- **PAUSE** 4-12
- **PHOSPHORUS** 4-358
- **PLOT.1D** 4-155
- **PLOT.2D** 4-163
- **PLOT.3D** 4-172
- **PRINT.1D** 4-151
- **PROFILE** 4-86
- **REACTION** 4-300
- **REGION** 4-64
- **RETURN** 4-9
- **SAVEFILE** 4-74
- **SELECT** 4-146
- **SOURCE** 4-8
- **STOP** 4-13
- **STRESS** 4-142
- **STRUCTURE** 4-79
- **UNDEFINE** 4-49
- **VACANCY** 4-327
- **VIEWPORT** 4-204

STEAM parameter, **DIFFUSION** statement
- 1D bipolar example 5-5

STEAM parameter, **ETCH** statement
- 1D local oxidation example 5-15

STOP 4-13
- description 4-13
- example 4-13

STRESS 4-142
- description 4-142
- example 4-144
- printing and plotting of stresses and displacements 4-144

stress history model, initial conditions 3-116

STRESS parameter, **PLOT.2D** statement
- local oxidation example 5-17

STRUCTURE 4-79
- description 4-81
- examples 4-82
- order of operations 4-82
- REFLECT parameter, trench implant simulation example 6-19
- truncation cautions 4-82

structure
- after silicide growth step *Fig.* 6-58
- immediately before silicide growth step *Fig.* 6-57
- with contours of boron concentration, after first p-well diffusion *Fig.* 6-45
- with contours of boron concentration, after formation of isolation region, plotted by file *s4ex4a.inp Fig.* 6-7

surface reaction rate 3-106

surface recombination rate depending on nitrogen 7-27

surface recombination velocity models 3-67

surface tension 3-104

syntax
- input statements 4-2
- parameter lists 4-4
- parameters 4-3

T

TAG parameter, **LINE** statement
- SOI MOSFET example 6-50

Taurus Layout, mask data files C-1

TCAD Products and Utilities Installation Manual 1-iv

Technology Interchange Format
- *see TIF* files

terminal output 2-6

thermal stress model equations 3-115, 3-116
- boundary conditions 3-115
- effect of etching on stress 3-117
- initial conditions 3-116
- limitations 3-117
- stress history model 3-117

THICKNES parameter, **DEPOSITION** statement
- 1D bipolar example 5-6

thin oxidation rate 3-106

thin regime 3-95

third part of input file
- *s4ex4d.inp*, showing junction capacitance extraction *Fig.* 6-17
- *s4ex5.inp*, using Monte Carlo implantation model *Fig.* 6-25
- *s4ex7a.inp*, for simulating CMOS process *Fig.* 6-34
- *s4ex8.inp*, final processing and plotting *Fig.* 6-47

THRESHOLD parameter, **ELECTRICAL** statement
- NMOS LDD process example 6-14

TIF files

- overview 2-7
- **TIF** parameter 4-77
- TSUPREM-4 version limitations 4-77
- user defined materials or impurities 4-77

tilt and rotation tables, accuracy after implantation 3-127

TILT parameter, **IMPLANT** statement
- trench implant simulation example 6-20

TIME parameter, **EPITAXY** statement
- 1D bipolar example 5-6

$TiSi_2$ growth kinetics 3-109
- diffusion of silicon 3-110
- initialization 3-110
- material flow 3-111
- reaction at $TiSi_2$/Si interface 3-110

titanium silicide model 3-111

transient clustering model 3-37, 7-16

transport coefficient 3-55

TRAP parameter, **ETCH** statement
- CMOS process example 6-36

trapezoidal etch model 3-158
- etch examples 3-159
- etch steps 3-158
- parameters 3-158

trench implant simulation example
- analytic implant 6-20
- **COLOR** statement 6-21
- **ELIMINATE** statement 6-19
- **END** statement 6-21
- **ETCH** statement 6-19
- **FOREACH** statement 6-21
- **LABEL** statement 6-21
- **LINE** statement 6-19
- Monte Carlo implant 6-23
- Monte Carlo summary 6-28
- overview 6-18
- **PLOT.1D** statement 6-21
- **PLOT.2D** statement 6-21
- plotting results of Monte Carlo method 6-24
- **SAVEFILE** statement 6-19
- **SELECT** statement 6-21
- structure generation 6-19

triangles
- flipping 3-10
- maximum number 3-3
- numerical integrity 3-10
- overview 3-3

TRUNCATE parameter, **STRUCTURE** statement
- CMOS process example 6-35

TSUPREM-4 models
- deposition 3-155
- diffusion 3-14
- electrical calculations 3-170
- epitaxial growth 3-153
- etching 3-157
- grid structure 3-3
- ion implantation 3-119
- modeling polycrystalline materials 3-161
- oxidation 3-84
- photoresist 3-156
- polysilicon Monte Carlo implant model 3-151
- references 3-178
- silicide models 3-109
- simulation structure 3-2
- stress history 3-115

tungsten silicide model 3-114

tutorial examples
- 1D bipolar example 5-2
- local oxidation, 2D simulation 5-13
- point defect models 5-28

typographical conventions 1-iv

U

UNDEFINE 4-49
- description 4-49
- example 4-49
- redefined parameter names 4-49

update solution 7-7

USEIT 7-1, 7-27
- examples 7-15
- overview 7-1
- physical model and equation interface (PMEI) 7-3
- using 7-3
- working diagram *Fig.* 4-232, 7-2

USEIT examples
- 5-stream diffusion model 7-27
- boron-interstitial clustering model with small clusters 7-21
- diffusion enhancement in thinner oxides 7-25
- diffusion in **PD.TRANS** 7-19
- dislocation loop model 7-17
- interface trap model 7-20
- interstitial clustering model 7-15
- SiGe model 7-38
- surface recombination rate depending on nitrogen 7-27
- transient clustering model 7-16

User-Specified Equation Interface (USEIT) 7-1

using TSUPREM-4
    file specification 2-3
    input files 2-4
    introduction 2-1
    library files 2-8
    output files 2-5
    program output 2-2
    starting TSUPREM-4 2-1
using USEIT 7-3
    application of user-specified models 7-8
    built-in keywords 7-10
    EQUATION 7-3
    interface functions 7-14
    INTERMEDIATE 7-8
    mathematical functions 7-11
    modification of built-in functions 7-9
    operators 7-11
    physical functions 7-13

V

V.INITOX model 3-68
V.MAXOX model 3-68
V.NORM model 3-68
VACANCY 4-327
    additional notes 4-336
    bulk and interface parameters 4-335
    description 4-335
    examples 4-335
VALUE parameter, **CONTOUR** statement
    local oxidation example 5-24
VC parameter, **MATERIAL** statement
    poly-buffered LOCOS example 6-29
VERTICAL model 3-95
VERTICAL model, recommended usage 3-96
vertical profiles
    after Monte Carlo implant *Fig.* 6-27
    produced by analytic implant *Fig.* 6-22
VIEWPORT 4-204
    description 4-204
    examples 4-205
    scaling plot size 4-204
VISCOELA model 3-99
    parameters 3-101
    recommended usage 3-103
    viscoelastic flow 3-100
VISCOELA parameter, **METHOD** statement
    local oxidation example 5-15
VISCOUS model 3-97
    incompressible viscous flow 3-98
    recommended usage 3-99
    stress dependence 3-98
VLENG parameter, **PLOT.2D** statement
    local oxidation example 5-16

W

W.RECATN parameter, **LABEL** statement
    1D local oxidation example 5-19
warnings 2-3
WIDTH parameter, **INITIALIZE** statement
    1D local oxidation example 5-20

X

X direction
    automatic grid generation 3-6
    column elimination 3-6
X grid
    MASK statement 3-6
    WIDTH parameter 3-6
X parameter, **ELECTRICAL** statement
    NMOS LDD process example 6-16
X parameter, **LABEL** statement
    1D bipolar example 5-8
X parameter, **X** statement
    local oxidation example 5-14
X.MAX parameter, **PLOT.2D** statement
    local oxidation example 5-16
X.VALUE parameter, **PLOT.1D** statement
    1D bipolar example 5-7
    local oxidation example 5-25

Y

Y parameter, **LABEL** statement
    1D bipolar example 5-8
Y parameter, **LINE** statement
    local oxidation example 5-14
Y.MAX parameter, **PLOT.2D** statement
    CMOS process example 6-35
    local oxidation example 5-16
Y.MIN parameter, **PLOT.2D** statement
    local oxidation example 5-16
Y.VALUE parameter, **PLOT.1D** statement

  1D bipolar example 5-7

`YDY` parameter, **`DEPOSITION`** statement
  SOI MOSPET example 6-50

Z

`Z` parameter, **`METHOD`** statement
  1D bipolar example 5-8

`Z` parameter, **`SELECT`** statement
  poly-buffered LOCOS example 6-30