SILVACO

Athena

User's Manual

Silvaco, Inc.
4701 Patrick Henry Drive, Bldg. 2
Santa Clara, CA 95054
Phone: (408) 567-1000
Web: www.silvaco.com

August 13, 2015

Notice

The information contained in this document is subject to change without notice.

Silvaco, Inc. MAKES NO WARRANTY OF ANY KIND WITH REGARD TO THIS MATERIAL, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTY OF FITNESS FOR A PARTICULAR PURPOSE.

Silvaco, Inc. shall not be held liable for errors contained herein or for incidental or consequential damages in connection with the furnishing, performance, or use of this material.

AccuCell, AccuCore, Athena, Athena 1D, Atlas, Blaze, C-Interpreter, Catalyst AD, Catalyst DA, Clarity RLC, Clever, Clever Interconnect, Custom IC CAD, DeckBuild, DevEdit, DevEdit 3D, Device 3D, DRC Assist, Elite, Exact, Expert, Expert C++, Expert 200, ExpertViews, Ferro, Gateway, Gateway 200, Giga, Giga 3D, Guardian, Guardian DRC, Guardian LVS, Guardian NET, Harmony, Hipex, Hipex C, Hipex NET, Hipex RC, HyperFault, Interconnect Modeling, IWorkBench, Laser, LED, LED 3D, Lisa, Luminous, Luminous 3D, Magnetic, Magnetic 3D, MaskViews, MC Etch & Depo, MC Device, MC Implant, Mercury, MixedMode, MixedMode XL, MultiCore, Noise, OLED, Optolith, Organic Display, Organic Solar, OTFT, Quantum, Quantum 3D, Quest, RealTime DRC, REM 2D, REM 3D, SEdit, SMovie, S-Pisces, SSuprem 3, SSuprem 4, SDDL, SFLM, SIPC, SiC, Silvaco, Silvaco Management Console, SMAN, Silvaco Relational Database, Silos, Simulation Standard, SmartSpice, SmartSpice 200, SmartSpice API, SmartSpice Debugger, SmartSpice Embedded, SmartSpice Interpreter, SmartSpice Optimizer, SmartSpice RadHard, SmartSpice Reliability, SmartSpice Rubberband, SmartSpice RF, SmartView, SolverLib, Spayn, SpiceServer, Spider, Stellar, TCAD Driven CAD, TCAD Omni, TCAD Omni Utility, TCAD & EDA Omni Utility, TFT, TFT 3D, Thermal 3D, TonyPlot, TonyPlot 3D, TurboLint, Universal Token, Universal Utility Token, Utmost III, Utmost III Bipolar, Utmost III Diode, Utmost III GaAs, Utmost III HBT, Utmost III JFET, Utmost III MOS, Utmost III MultiCore, Utmost III SOI, Utmost III TFT, Utmost III VBIC, Utmost IV, Utmost IV Acquisition Module, Utmost IV Model Check Module, Utmost IV Optimization Module, Utmost IV Script Module, VCSEL, Verilog-A, Victory, Victory Cell, Victory Device, Victory Device Single Event Effects, Victory Process, Victory Process Advanced Diffusion & Oxidation, Victory Process Monte Carlo Implant, Victory Process Physical Etch & Deposit, Victory Stress, Virtual Wafer Fab, VWF, VWF Automation Tools, VWF Interactive Tools, and Vyper are trademarks of Silvaco, Inc.

All other trademarks mentioned in this manual are the property of their respective owners.

How to Read this Manual

Style Conventions		
Font Style/Convention	**Description**	**Example**
•	This represents a list of items or terms.	• Bullet A • Bullet B • Bullet C
1. 2. 3.	This represents a set of directions to perform an action.	To open a door: 1. Unlock the door by inserting the key into keyhole. 2. Turn key counter-clockwise. 3. Pull out the key from the keyhole. 4. Grab the doorknob and turn clockwise and pull.
→	This represents a sequence of menu options and GUI buttons to perform an action.	**File→Open**
`Courier`	This represents the commands, parameters, and variables syntax.	`HAPPY BIRTHDAY`
Times Roman Bold	This represents the menu options and buttons in the GUI.	**File**
New Century Schoolbook Italics	This represents the variables of equations.	$x + y = 1$
Note:	This represents the additional important information.	**Note:** Make sure you save often when working on a manual.

Table of Contents

Chapter 1

Chapter 2

Chapter 3

Chapter 5

Chapter 6

Chapter 1
Introduction

1.1 Athena Overview

Athena is a simulator that provides general capabilities for numerical, physically-based, two-dimensional simulation of semiconductor processing. Athena has a modular architecture that the following licensable tools and extensions:

- **Athena:** This tool performs structure initialization and manipulation, and provides basic deposition and etch facilities
- **SSuprem4:** This tool is used in the design, analysis, and optimization of silicon semiconductor structures. It simulates silicon processing steps such as ion implantation, diffusion and oxidation.
- **Elite:** This tool is a general purpose 2D topography simulator that accurately describes a wide range of deposition, etch and reflow processes used in modern IC technologies.
- **Optolith:** This tool performs general optical lithography simulation including 2D aerial imaging, non-planar photoresist exposure, and post exposure bake and development.

1.1.1 Using This Manual

This chapter is an overview of Athena. For new users, read Chapter 2 “Tutorial”, especially the sections that describe the simulator or modules that you have licensed. This chapter will give you a basic understanding of what Athena can do and how it’s used. The remaining chapters will give you a detailed understanding of Athena’s capabilities and how to access them.

Athena is supplied with a number of example problem descriptions. You can access them through DeckBuild as described in the DeckBuild User’s Manual. These examples demonstrate the capabilities of Athena. The input files provided as part of these examples can provide an excellent starting point for developing your own Athena input files.

1.1.2 Technical Support

If you have difficulties or questions relating using Athena, e-mail Silvaco Support at support@silvaco.com.

When you send us an e-mail message, please:

1. Explain the problem or question in detail.
2. Include any input files that you have created.
3. Provide us with the version number of Athena and the version numbers of the VWF Interactive Tools that you are using.
4. Include your business telephone number and fax number.

Silvaco support will contact you promptly and resolve your problem as quickly as possible.

User feedback helps further develop Athena. Please send your comments on the programs, suggestions for improvements, and additional feature requests to support@silvaco.com.

1.2 Athena Features and Capabilities

Table 1-1 shows the features and capabilities of Athena.

Table 1-1 Athena Features and Capabilities

Features	Capabilities
Bake	• Time and temperature bake specification. • Models photoresist material flow. • Models photo-active compound diffusion.
C-Intepreter	• Allows user-defined models for implant damage, Monte Carlo plasma etching and diffusion in SiGeC.
CMP	• Models Chemical Mechanical Polishing. • Hard and soft models or a combination of both. • Includes isotropical etch component.
Deposition	• Conformal deposition model. • Hemispherical, planetary, and conical metallization models. • Unidirectional or dual directional deposition models. • CVD model. • Surface diffusion/migration effects. • Ballistic deposition models including atomistic positioning effects. • User-definable models. • Default deposition machine definitions.
Development	• Five different photoresist development models.
Diffusion	• Impurity diffusion in general 2D structures including diffusion in all material layers. • Fully coupled point defect diffusion model. • Oxidation enhanced/retarded diffusion effects. • Rapid thermal annealing. • Models simultaneous material reflow and impurity diffusion. • Impurity diffusion in polysilicon accounting for grain and grain boundary components.
Epitaxy	• 2D epitaxy simulation including auto-doping.

Table 1-1 Athena Features and Capabilities

Features	Capabilities
Etch	• Extensive geometric etch capability. • Wet etching with isotropic profile advance. • RIE model that combines isotropic and directional etch components. • Microloading effects. • Angle dependence of etchant source. • Default etch machine definitions. • Monte Carlo plasma etching. • Dopant enhanced etching.
Exposure	• Model is based on the Beam Propagation Method simulating reflections and diffraction effects in non-planar structures with capability to take into account local modification of material optical properties the absorbed dose. • Defocus and large numerical aperture effects.
Imaging	• Two dimensional, large numerical aperture, aerial image formation. • Up to 9th order imaging system aberrations. • Extensive source and pupil plane filtering for enhanced aerial images. • Full phase shift and transmittance variation mask capabilities.

Table 1-1 Athena Features and Capabilities

Features	Capabilities
Implantation	• Experimentally verified Pearson and dual Pearson analytical models. • Extended low energy and high energy implant parameter tables. • Binary Collision Approximation Monte Carlo calculations for crystalline and amorphous materials. • Universal tilt and rotation capability for both analytic and Monte Carlo calculations.
Oxidation	• Compressible and viscous stress dependent models. • Separate rate coefficients for silicon and polysilicon materials. • HCL and pressure-enhanced oxidation models. • Impurity concentration dependent effects. • Ability to simulate the oxidation of structures with deep trenches, undercuts, and ONO layers. • Accurate models for the simultaneous oxidation and lifting of polysilicon regions.
Silicidation	• Models for titanium, tungsten, cobalt, and platinum silicides. • Experimentally verified growth rates. • Reactions and boundary motion on silicide/metal and silicide/silicon interfaces. • Accurate material consumption model.

1.2.1 Using Athena With Other Silvaco Software

Athena is normally used in conjunction with the VWF Interactive Tools. These tools include DeckBuild, TonyPlot, DevEdit, MaskViews and Optimizer. DeckBuild provides an interactive run time environment. TonyPlot supplies scientific visualization capabilities. DevEdit is an interactive tool for structure and mesh specification and refinement, and MaskViews is an IC Layout Editor. The Optimizer supports black box optimization across multiple simulators. For more information about VWF Interactive Tools, see their respective user manuals.

Athena is also frequently used in conjunction with the Atlas device simulator. Athena predicts the physical structures that result from processing. These physical structures are used as input by Atlas, which then predicts the electrical characteristics associated with specified bias conditions. Using Athena and Atlas makes it easy to determine the impact of process parameters on device characteristics.

Athena can also be used as one of the core simulators within Virtual Wafer Fab (VWF). VWF makes it convenient to perform highly automated simulation-based experimentation. VWF is used in a way that closely resembles experimental research and development procedures. Therefore, it links simulation closely to technology development, resulting in greatly increased benefits from simulation use.

For more information about VWF, see the VWF User’s Manual.

1.2.2 The Value Of Physically-Based Simulation

Physically-based process simulators predict the structures that result from specified process sequences. This is done by solving systems of equations that describe the physics and chemistry of semiconductor processes. Detail analysis of various aspects of process simulation can be found in [1] and [2].

Physically-based simulation provides three major advantages: it is predictive, it provides insight, and it captures theoretical knowledge in a way that makes this knowledge available to non-experts.

Physically-based simulation is different from empirical modeling. The goal of empirical modeling is to obtain analytic formulae that approximate existing data with accuracy and minimum complexity. Empirical models provide efficient approximation and interpolation. Empirical models, however, doesn't provide insight, predictive capabilities, or capture theoretical knowledge. Physically-based simulation is an alternative to experiments as a source of data. Empirical modeling can provide compact representations of data from either source.

Physically-based simulation has become very important for two reasons. One, it's almost always much quicker and cheaper than performing experiments. Two, it provides information that is difficult or impossible to measure.

Physically-based simulation has two drawbacks: you must incorporated are that all the relevant physics and chemistry into a simulator and numerical procedures, and you must be implemented to solve the associated equations. But these tasks have been taken care of for Athena users.

Physically-based process simulation tools users must specify the problem to be simulated. Athena users specify the problem by defining the following:

- The initial geometry of the structure to be simulated.
- The sequence of process steps (e.g., implantation, etching, diffusion, exposure) that are to be simulated.
- The physical models to be used.

The subsequent chapters of this manual describe how to perform these steps.

Chapter 2
Tutorial

2.1 Getting Started

This chapter is to help you start using Athena by providing a step-by-step tutorial centered on a typical process simulation sequence. It explains how Athena uses the Interactive tools (i.e., DeckBuild, TonyPlot, MaskViews, DevEdit, and Optimizer). These tools make Athena easier to use and they provide visualization and interface capabilities with other Silvaco tools. This tutorial assumes that you are familiar with the basic features of the Interactive Tools. For more information about these tools, see their respective user manuals.

This chapter begins by explaining how to start Athena and continues with tutorials on how to use the program. We recommend that you read Section 2.3 “Creating a Device Structure Using Athena” before you move on to the section appropriate to the particular tool you will be using. This section explains how to start Athena, how to load and run standard examples, and how to use the Athena online help facility.

The following explanations assume that Athena has been properly installed. See the Silvaco Installation, Migration and Troubleshooting Guide if you encounter installation difficulties.

2.1.1 Running Athena Under DeckBuild

DeckBuild is an interactive graphic environment that is used for the following purposes:

- Generating input files for process or device simulation or both,
- Running simulations interactively,
- Interfacing between different simulators,
- Invoking other Interactive tools.

To start Athena under DeckBuild in interactive mode, enter the following UNIX command:

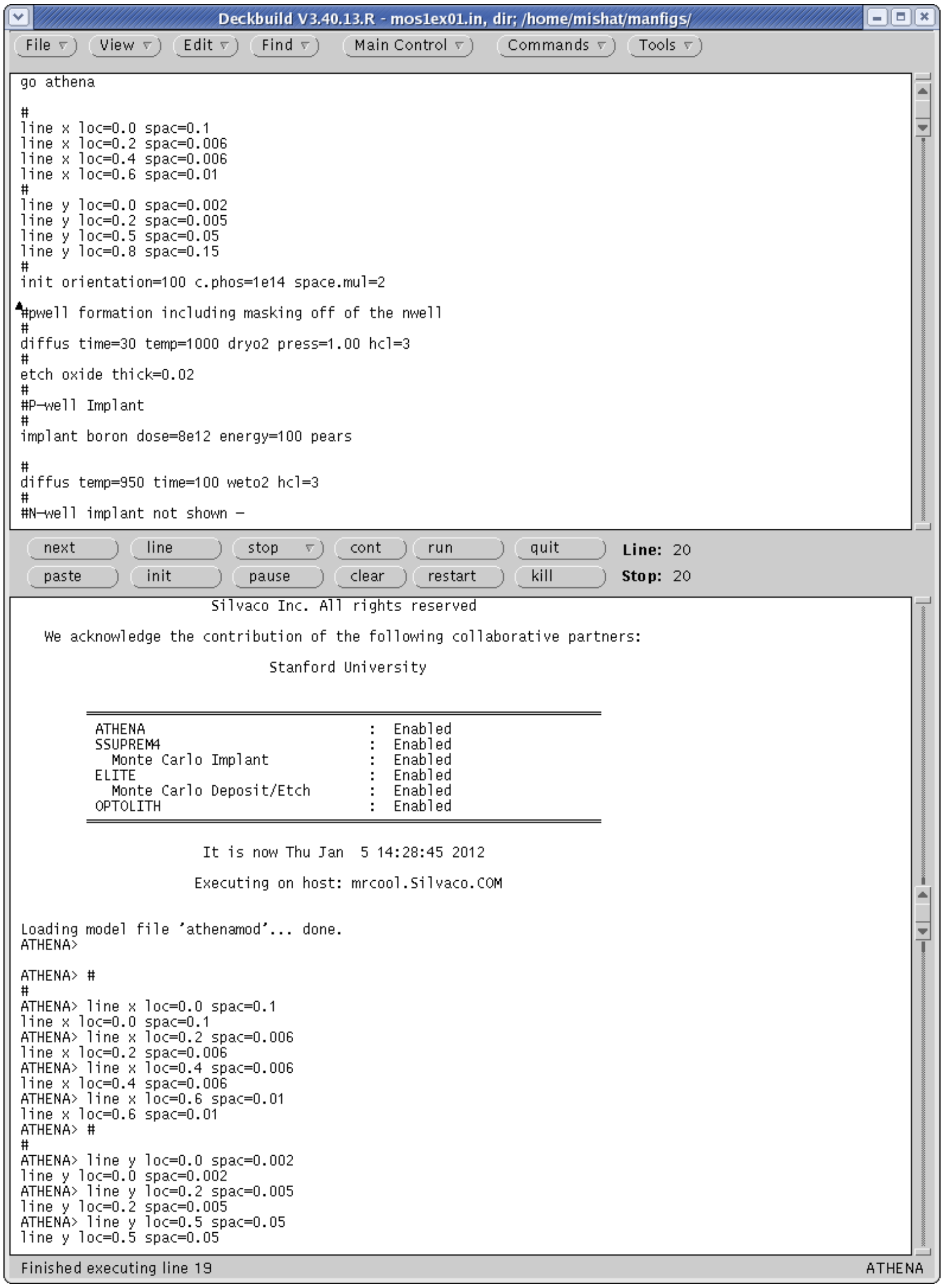
```
deckbuild -an
```

Note: To start the one-dimensional version of Athena (Athena 1D) under DeckBuild, use the following command:

```
deckbuild -an1.
```

After a short delay, the Main DeckBuild Window (See Figure 2-1) will appear. The lower text window of this window will contain the Athena logo and version number, a list of available modules, and a command prompt. Athena is now ready to run. To become familiar with the mechanics of running Athena under DeckBuild, load and run some of the Athena standard examples.

The method described here is the recommended procedure for starting the program. There are other methods and modes of running Athena. Section 2.2 “Operation Modes” or Section 2.8 “Using Advanced Features of Athena” will describe these methods and modes.



```
go athena

#
line x loc=0.0 spac=0.1
line x loc=0.2 spac=0.006
line x loc=0.4 spac=0.006
line x loc=0.6 spac=0.01
#
line y loc=0.0 spac=0.002
line y loc=0.2 spac=0.005
line y loc=0.5 spac=0.05
line y loc=0.8 spac=0.15
#
init orientation=100 c.phos=1e14 space.mul=2

#pwell formation including masking off of the nwell
#
diffus time=30 temp=1000 dryo2 press=1.00 hcl=3
#
etch oxide thick=0.02
#
#P-well Implant
#
implant boron dose=8e12 energy=100 pears

#
diffus temp=950 time=100 weto2 hcl=3
#
#N-well implant not shown -
```

Line: 20
Stop: 20

```
                     Silvaco Inc. All rights reserved

   We acknowledge the contribution of the following collaborative partners:

                            Stanford University


         ==============================================================
         ATHENA                                :  Enabled
         SSUPREM4                              :  Enabled
           Monte Carlo Implant                 :  Enabled
         ELITE                                 :  Enabled
           Monte Carlo Deposit/Etch            :  Enabled
         OPTOLITH                              :  Enabled
         ==============================================================

                    It is now Thu Jan  5 14:28:45 2012

                    Executing on host: mrcool.Silvaco.COM


Loading model file 'athenamod'... done.
ATHENA>

ATHENA> #
#
ATHENA> line x loc=0.0 spac=0.1
line x loc=0.0 spac=0.1
ATHENA> line x loc=0.2 spac=0.006
line x loc=0.2 spac=0.006
ATHENA> line x loc=0.4 spac=0.006
line x loc=0.4 spac=0.006
ATHENA> line x loc=0.6 spac=0.01
line x loc=0.6 spac=0.01
ATHENA> #
#
ATHENA> line y loc=0.0 spac=0.002
line y loc=0.0 spac=0.002
ATHENA> line y loc=0.2 spac=0.005
line y loc=0.2 spac=0.005
ATHENA> line y loc=0.5 spac=0.05
line y loc=0.5 spac=0.05
```

Finished executing line 19 ATHENA

Figure 2-1 Main DeckBuild Window

2.1.2 Loading And Running Athena Standard Examples

DeckBuild makes it possible to load and run a number of example simulation input files. To access the Athena examples, select **Main Control→Examples** and the DeckBuild Examples Window will appear (See Figure 2-2).

Groups of DeckBuild examples are listed in the **Section** menu and are grouped according to the simulator or simulation topic the example demonstrates. The **Sub-section** menu lists individual example input files. To run examples, select one of the sections (e.g., **ATHENA_IMPLANT**) in the Section menu. This will open a list of input file names. Short descriptions of the examples will appear in the Examples Window.

Select one of the input files using the **Sub-section** menu or by double-clicking on the input file name and a description of the selected input file will appear.

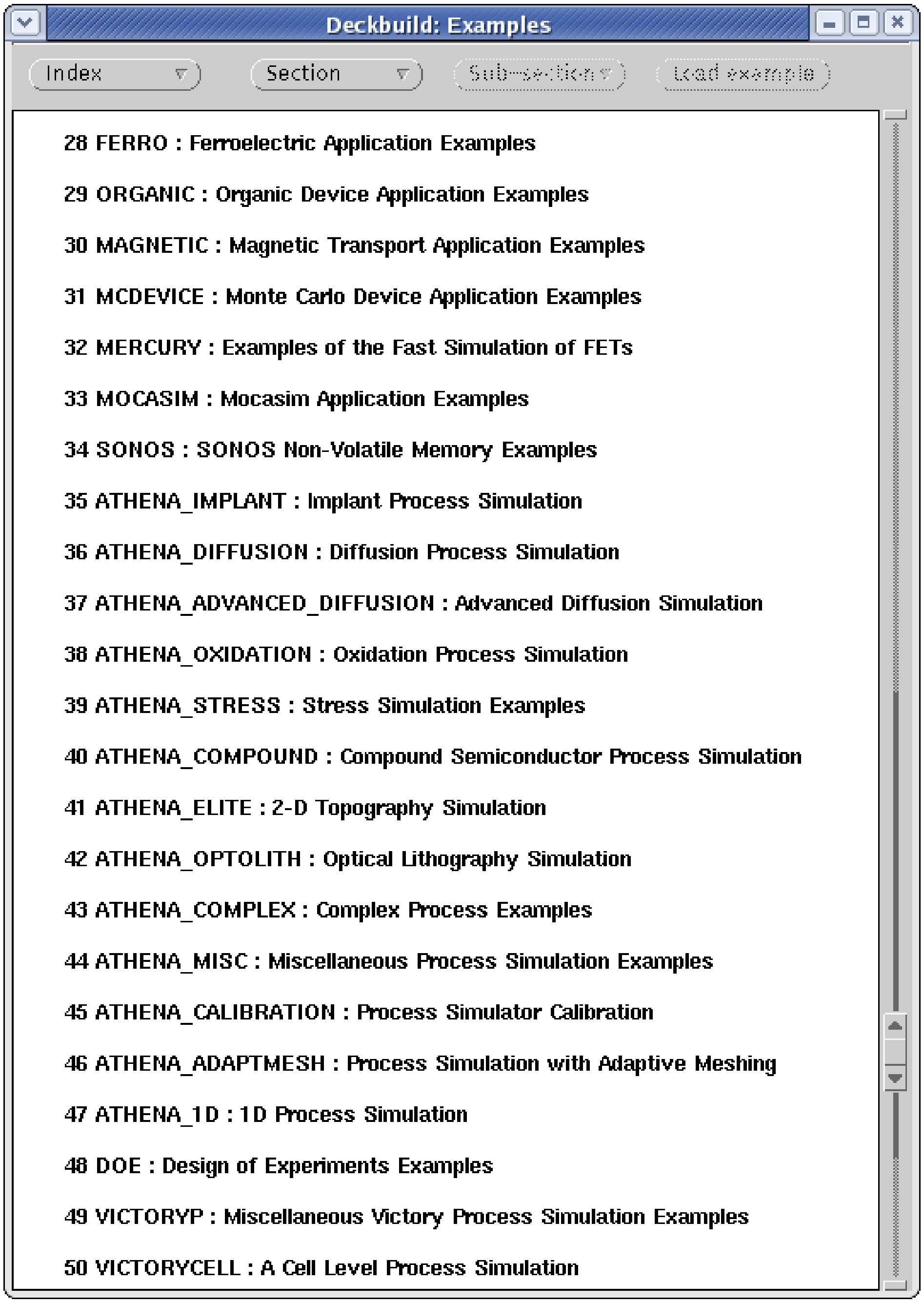
Press the **Load Example** button to load the selected input file into the DeckBuild Text Subwindow (bottom panel of the window). The input file, along with other files associated with the input file (structure files, setfiles for TonyPlot, and layout files for MaskViews), will be copied into your current directory.

Note: To access the examples in PC, select **Help→Examples**.

Once the input file is in the DeckBuild Text Subwindow (bottom panel of the window), press the **Run** button in the Main DeckBuild window or follow the special instructions in the DeckBuild Examples Window to run the input file.

Most of the Athena examples contain preset calls to the graphical postprocessing tool TonyPlot. One or more plots will appear while the selected example is running.

If you are not familiar with DeckBuild, use a simple example to learn the basic DeckBuild features and capabilities. For more information, see the DeckBuild User’s Manual. This will assist you in working through the rest of the tutorial.



28 FERRO : Ferroelectric Application Examples

29 ORGANIC : Organic Device Application Examples

30 MAGNETIC : Magnetic Transport Application Examples

31 MCDEVICE : Monte Carlo Device Application Examples

32 MERCURY : Examples of the Fast Simulation of FETs

33 MOCASIM : Mocasim Application Examples

34 SONOS : SONOS Non-Volatile Memory Examples

35 ATHENA_IMPLANT : Implant Process Simulation

36 ATHENA_DIFFUSION : Diffusion Process Simulation

37 ATHENA_ADVANCED_DIFFUSION : Advanced Diffusion Simulation

38 ATHENA_OXIDATION : Oxidation Process Simulation

39 ATHENA_STRESS : Stress Simulation Examples

40 ATHENA_COMPOUND : Compound Semiconductor Process Simulation

41 ATHENA_ELITE : 2-D Topography Simulation

42 ATHENA_OPTOLITH : Optical Lithography Simulation

43 ATHENA_COMPLEX : Complex Process Examples

44 ATHENA_MISC : Miscellaneous Process Simulation Examples

45 ATHENA_CALIBRATION : Process Simulator Calibration

46 ATHENA_ADAPTMESH : Process Simulation with Adaptive Meshing

47 ATHENA_1D : 1D Process Simulation

48 DOE : Design of Experiments Examples

49 VICTORYP : Miscellaneous Victory Process Simulation Examples

50 VICTORYCELL : A Cell Level Process Simulation

Figure 2-2 DeckBuild Examples Window

Online Help

You can find information on Athena statements and syntax using the online help facility. You can access this facility in interactive mode or through DeckBuild. Typing `help` at the **ATHENA>** prompt either in Interactive Mode or in the DeckBuild Text Subwindow will display a list of valid Athena statements. The syntax for the `help` command is shown below.

```
HELP <command name>
```

This command will give you additional information about parameter names, types, initial values, and a description of the parameters for the specified command.

To obtain more information on Athena default parameters (stored in a special file called a*thenamod*), select **Command→Models...** in the Main DeckBuild Window. This opens *athenamod* in a text editing window, making it possible to read the file or copy and paste statements from the file into a DeckBuild Text Subwindow.

Select **Command→Notes...** to open a special information file that includes the current release notes and a release history.

For more information about Athena syntax, statements, parameters, and their default values, see Chapter 6 “Statements”.

2.2 Operation Modes

Athena is normally used through the DeckBuild run-time environment, which supports both interactive and batch mode operation. We recommend that you always use Athena through DeckBuild. In this section, we present the basic information you need to run Athena in the DeckBuild environment. The DeckBuild User's Manual provides a more detailed description of the features and capabilities of DeckBuild.

2.2.1 Interactive Mode With DeckBuild

To start Athena under DeckBuild, type:

```
deckbuild -an
```

at the UNIX system command prompt. The command line option `-an` instructs DeckBuild to start Athena as the default simulator.

To start with an existing input file, type:

```
deckbuild -an <input filename>
```

The run-time output shows the execution of each Athena command and includes error messages, warnings, extracted parameters and other important output for evaluating each Athena run. When you run Athena interactively, the run-time output is sent to the DeckBuild Text Subwindow of the DeckBuild Application Window and you can save it as needed. You don't need to save the run-time output explicitly. The following command line, however, specifies the file name that will be used for storing the run-time output:

```
deckbuild -an <input filename> -outfile <output filename>
```

In this case, the run-time output is sent to the output file and to the output section of the DeckBuild Window.

2.2.2 Batch Mode With DeckBuild

To use DeckBuild in a non-interactive or batch mode, add the `-run` parameter to the command that invokes DeckBuild. A prepared command file is required for running in batch mode. We advise you to save the run-time output to a file because error messages in the run-time output will be lost when the batch job completes.

```
deckbuild -run -an <input filename> -outfile <output filename>
```

Using this command requires a local X-Windows system to be running. The job runs inside a DeckBuild icon on the terminal and quits automatically when the Athena simulation is complete. You can also run DeckBuild using a remote display. For example:

```
deckbuild -run -an <input file> -outfile <output file> -display
<hostname>:0.0
```

2.2.3 No Windows Batch Mode With DeckBuild

The `-ascii` parameter is required for completely non-X Windows operation of DeckBuild. For example:

```
deckbuild -run -ascii -an <input filename> -outfile <output file-
name>
```

This command directs DeckBuild to run the Athena simulation without any display of the DeckBuild window or icon. This is useful for remote execution without an X windows emulator or for replacing UNIX-based Athena runs within framework programs.

When using batch mode, use the UNIX command suffix `&` to detach the job from the current command shell. To run a remote Athena simulation under DeckBuild without display, and then logout from the system, use the UNIX `nohup` command before the following DeckBuild command line:

```
nohup deckbuild -run -ascii -an <input filename> -outfile <output
filename> &
```

2.2.4 Running Athena inside DeckBuild

Each Athena run inside DeckBuild should start with the following command line:

```
go athena
```

A single input file can contain several Athena runs each separated with a `go athena` line. Input files within DeckBuild can also contain runs from other programs such as Atlas or DevEdit along with the Athena runs.

Running a given version number of Athena

You can modify the `go` statement to provide parameters for the Athena run. To run version 5.8.0.R, the syntax is

```
go athena simflags="-V 5.8.0.R"
```

Starting Parallel Athena

The `-P` option is used to set the number of processors to use in a parallel Athena run (only the MC Implant module is parallelized starting Athena release 5.16.0.R). If the number of processors specified by `-P` is greater than the number of processors available, it is automatically reduced to this cap number. If `-P` parameter is not specified, Athena will run on all available processors automatically. To run Athena on 4 processors, use the command:

```
go athena simflags=" -V 5.16.0.R -P 4"
```

Running Athena with a user-specified default parameter file

Athena supports the use of multiple default parameter files. These files have the default root filename *athenamod*. To start Athena with *athenamod.97*, the syntax is

```
go athena simflags="-modfile 97"
```

2.2.5 Running Athena In Standalone Mode (Without DeckBuild)

You can run Athena outside the DeckBuild environment, but we don't recommended it. If you don't want the overhead of the DeckBuild Window, use the **No Windows Mode**. Many important features such as variable substitution, automatic interfacing to device simulation, and parameter extraction are unavailable outside the DeckBuild environment. To run Athena directly under UNIX, use the following command line:

```
athena <input filename>
```

To save the run-time output to a file, don't use the UNIX redirect command (>). Instead, specify the name of the output file:

```
athena <input filename> -logfile <output filename>
```

Note: Some of the standard examples supplied with Athena will not run correctly outside of DeckBuild.

2.2.6 Running Athena 1D (One-Dimensional Version of Athena)

If you have license for Athena 1D, you can run it exactly as described above by using `athena1d` instead of `athena` and `-an1` parameter instead of `-an` parameter.

2.3 Creating a Device Structure Using Athena

2.3.1 Procedure Overview

Athena is designed as a process simulation framework. The framework includes simulator independent operations and simulator specific functions that simulate different process steps (e.g., implant, RIE, or photoresist exposure). This section describes Athena input/output and the following basic operations for creating an input file:

- Developing a good simulation grid
- Performing conformal deposition
- Performing geometric etches
- Structure manipulation
- Saving and loading structure information
- Interfacing with device simulators
- Using different Interactive tools

These operations are relevant to all individual Athena process simulators. This part of the tutorial should help you if you're new to each of the process simulators.

The three sections of the tutorial: SSuprem4, Elite, and OPTOLITH are devoted to individual simulators and should be read if you're going to use those simulators.

2.3.2 Athena Input/Output

Before proceeding to the Athena operation, we will discuss how to provide Athena with input information and the forms of output information available from Athena.

Input Information

The bulk of input information for Athena is usually provided in the form of input files. An input file is a text file that can be prepared by using DeckBuild (which will be described throughout the rest of the tutorial) or any ASCII text editor (such as *vi* on any UNIX system, or *textedit* on a SUN system). The individual lines of the text file are called statements. Each statement consists of a statement name and a set of parameters that specify a certain step of a process simulation or model coefficients used during subsequent simulation steps. See Section 6.1 "Overview" for details on statement syntax.

The remainder of this tutorial will introduce you to the task of creating good input files.

Since Athena uses a great deal of default information, much of the default information is stored in several non-user-specified files. These files are as follows:

- The *athenamod* file includes default parameters of physical models, diffusion and oxidation coefficients, default parameters of numerical methods, characteristics of predefined deposition and etching machines, and optical parameters of materials for lithography simulation.
- The *std_table* and several *svdp***** files in the *implant_tables* directory contain ion implantation look-up tables.
- Several files with suffix *.mod* in the *pls* and *models* directories contain parameters for advanced diffusion models (Section 3.2 "Advanced Diffusion Models").
- The *athenares* file includes resistivity vs. doping concentration data.

It’s important to be aware that information from the *athenamod* file is loaded into Athena each time it starts. You can override any of the *athenamod* default parameters by specifying an alternative parameter in an input file or by specifying the entire models file using the `-modfile` option.

Output Information

All run-time output generated by Athena will appear in the DeckBuild Text Subwindow when running DeckBuild, or in the current window (or specified output file) when running Athena standalone. Run-time output can be grouped into two categories: Standard Output and Standard Error Output.

Standard Output consists of the output from the `PRINT.1D` statements or the `EXTRACT` statement of DeckBuild or both, and the normal information messages generated by Athena. The number of messages generated depends on the output mode chosen in the `OPTION` statement. The `QUIET` mode is the default. Minimum output is generated in this case: all statements are echoed and the status of a time consuming simulation is reported. The `NORMAL` option produces some additional output information, including information about the current grid (e.g., number of nodes or triangles). `VERBOSE` and `DEBUG` modes are useful for debugging but these options produce too much output for any other purpose.

Standard Error Output consists of the warning and error messages describing syntax errors, file operation errors, system errors, and internal inconsistencies.

Standard Structure File Format

The main channel of information exchange between Athena and other simulators and tools is the Standard Structure File (SSF) format. SSF is a universal file format used by a number of Silvaco simulation programs. The `STRUCTURE` statement of Athena creates a Standard Structure File, which contains mesh and solution information, model information, and other related parameters.

The saved Standard Structure File can be used by the following:

- Athena to re-initialize the structure and continue process simulation.
- Atlas or other device simulators to perform electrical analysis of the structure produced by Athena.
- TonyPlot to graphically display a solution created by Athena.
- DevEdit to generate an updated mesh and export the mesh and doping back to Athena or any other simulator.

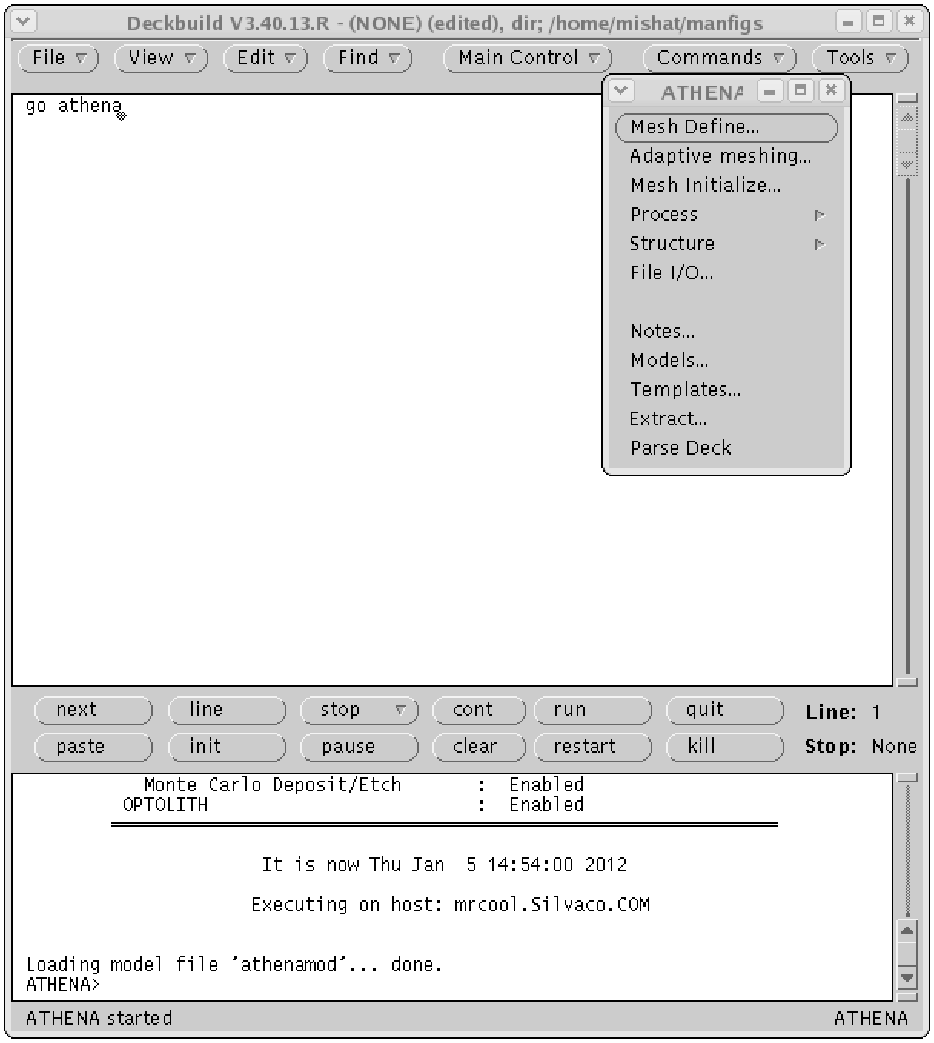
For more information on structure files, see “Saving a Structure File for Plotting or Initializing an Athena Input file for Further Processing” on 50.

2.3.3 Creating An Initial Structure

This section will describe how to use DeckBuild's Commands menu to create a typical Athena input file. The goal of this section is not to design a real process sequence, but to demonstrate the use of specific Athena statements and parameters, as well as some DeckBuild features, to create a realistic input file. You can find many realistic process input files among the examples and use them as a starting point in your process simulation.

Defining Initial Rectangular Grid

Once DeckBuild is running and the current simulator is set to Athena, open and pin the **Commands** menu as shown in Figure 2-3. Then, select **Mesh Define....** and the Athena Mesh Define Menu will appear. We recommend that you pin this popup because it will be used often in designing an initial mesh.



```
go athena
```

```
          Monte Carlo Deposit/Etch       :  Enabled
        OPTOLITH                         :  Enabled

                    It is now Thu Jan  5 14:54:00 2012

                    Executing on host: mrcool.Silvaco.COM


Loading model file 'athenamod'... done.
ATHENA>
```

Figure 2-3 Commands Menu

Now, you can specify the initial rectangular grid. The correct specification of a grid is critical in process simulation. The number of nodes in the grid N_p has a direct influence on simulation accuracy and time. A finer grid should exist in those areas of the simulation structure where ion implantation will occur, where p-n junction will be formed, or where optical illumination will change photoactive component concentration. The number of arithmetic operations necessary to achieve a solution for processes simulated, using the finite element analysis method could be estimated as $(N_p)^{\alpha}$, where α is of order 1.5 - 2.0.

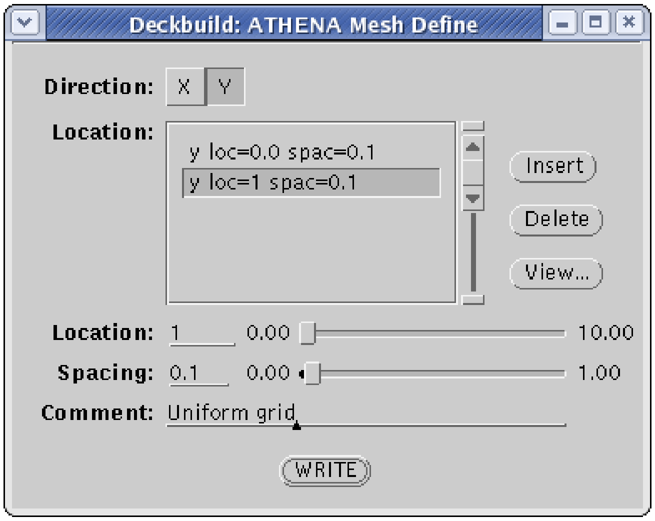
Therefore, to maintain the simulation time within reasonable bounds, the fine grid should not be allowed to spill over into unnecessary regions. The maximum number of grid nodes is 20,000 for Athena simulations, but most practical simulations use far fewer nodes than this limit.

To create a simple uniform grid in a rectangular 1 µm by 1 µm simulation area, click on the **Location** field and enter a value of `0.0`. Then, click on the **Spacing** field and enter a value of `0.10`. Then, click on the **Insert** button and the line parameters will appear in the scrolling list.

Note: Athena coordinate system has positive x axis pointed to the right along the structure surface and positive y axis pointed down to the depth of the structure.

In the same way, set the location of a second X line to `1.0` with a spacing of `0.1`. You can either set the values by dragging a slider or by entering a number directly.

Now, select the **Y** direction and set the lines with the same values as the **X** direction. You can now add the comments at the **Comment** line. The Athena Mesh Define menu should appear as shown in Figure 2-4.

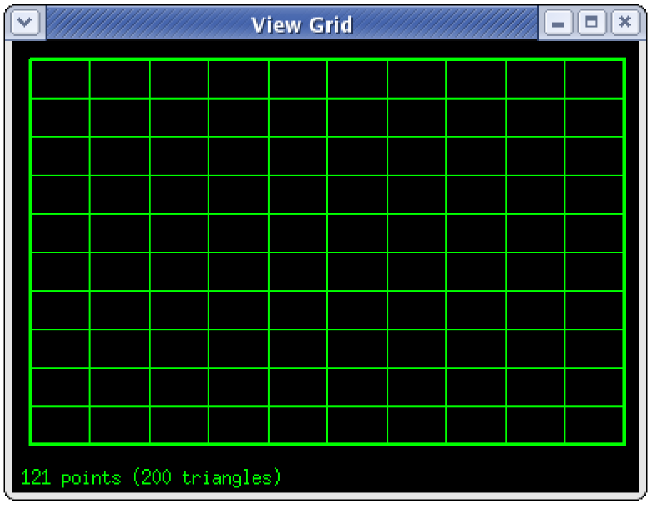


y loc=0.0 spac=0.1
y loc=1 spac=0.1

Location: 1 0.00 10.00
Spacing: 0.1 0.00 1.00
Comment: Uniform grid

Figure 2-4 Athena Mesh Define Menu

You can now write the menu-prepared mesh information into the input file. But first, preview the rectangular grid by selecting the **View...** button and the View Grid window (Figure 2-5) will appear. Notice that vertical and horizontal grid lines are distributed uniformly, and the 121 points and the 200 triangles will be generated.

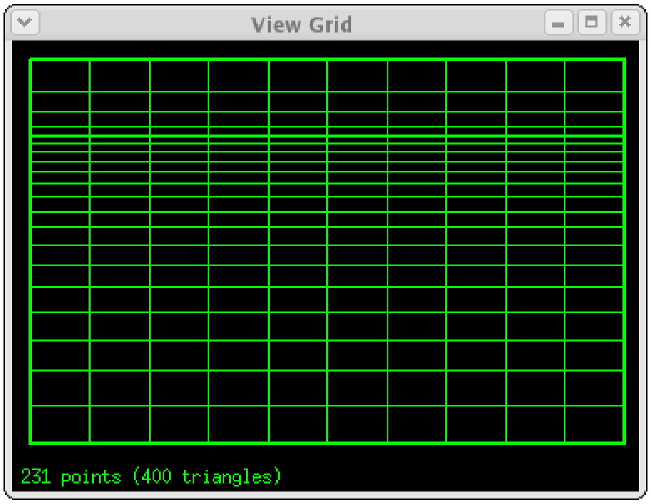


121 points (200 triangles)

Figure 2-5 View Grid Window

A uniform grid such as the one shown in Figure 2-5 is inefficient for performing complex simulations. Therefore, the grid must be improved. First, make a better grid in the y-direction. Usually, it’s necessary to get better resolution for the depth profile after the ion implantation step. When adaptive gridding capability isn’t used, apply preliminary knowledge of the process you are going to simulate.

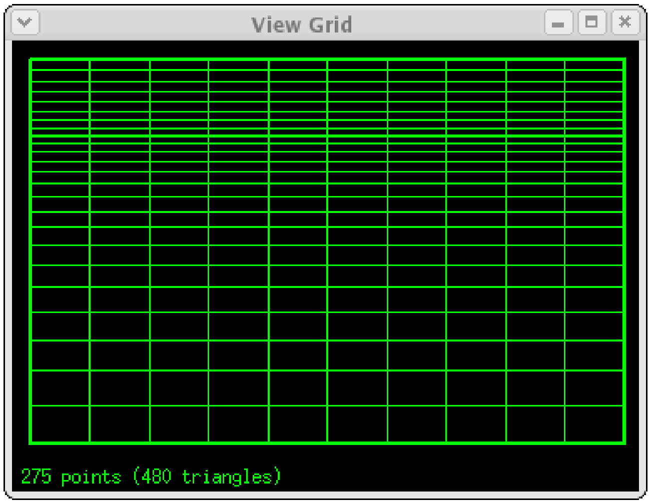
Suppose you want to perform a 60 keV boron implant so that the implant peak would be around 0.2 µm. It is reasonable to make a finer grid at this depth. To achieve this, simply add one more Y-line by setting the **Location** to `0.2` and the **Spacing** to `0.02`. The new rectangular grid (Figure 2-6) will now appear. Notice the number of points and triangles have increased to 231 and 400 respectively.



231 points (400 triangles)

Figure 2-6 New Rectangular Grid

The minimum spacing in the Y-direction is at 0.2 µm, and the spacing gradually increases toward the bottom and the top of the structure. Since the spacing at y=0 is still `0.1`, only 3 grid lines lie between 0 and 0.2 µm. You may want to make a finer grid at the top of the structure. To do this, select the top line of the Y Location scrolling list, change the spacing to `0.03`, and press the **Insert** button. The selected line will be replaced by `Y LOC=0.00 SPAC=0.03`. If you then press the **View...** button, there will be 8 grid lines between y=0 and y=0.2 (Figure 2-7).

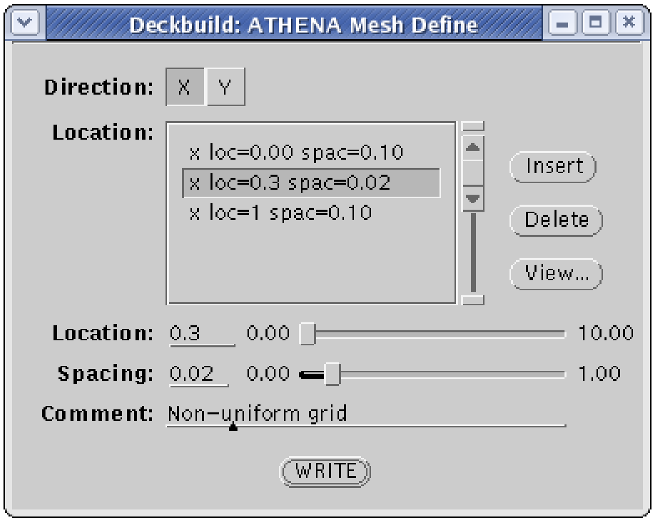


275 points (480 triangles)

Figure 2-7 Inserting New Grid Lines

To improve the initial grid in the x-direction, consider two things. First, make sure that a good 2D profile resolution is specified under the mask edges. Second, make sure the vertical grid lines are placed along future mask edges. To build half of a 0.6 µm MOS structure with the center of the gate at x=0, there must be an additional X line at x=0.3 and spacing at this line must be small enough to obtain good lateral resolution of source/drain implants. To add these items, return to the X direction specification in the **Mesh Define** menu and insert an additional X line at x=0.3 with spacing = 0.02.

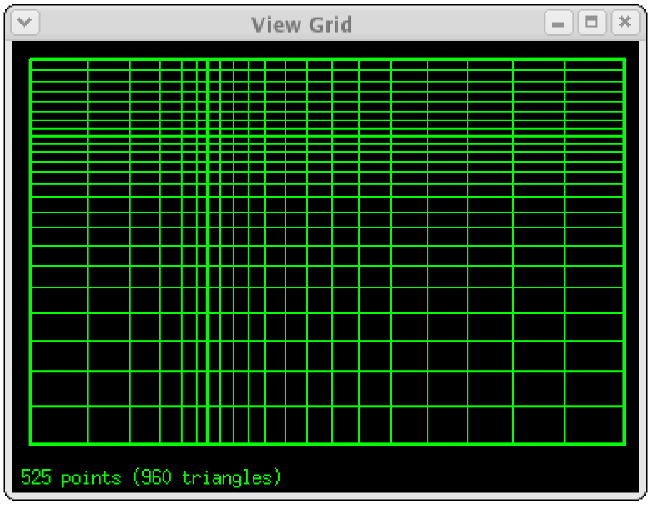
After this final insertion and adding any desired **Comment** information, the **Mesh Define** menu should appear as shown in Figure 2-8. The grid will have 525 points and 960 triangles (see Figure 2-9).



x loc=0.00 spac=0.10
x loc=0.3 spac=0.02
x loc=1 spac=0.10

Location: 0.3
Spacing: 0.02
Comment: Non-uniform grid

Figure 2-8 Athena Mesh Define Menu



525 points (960 triangles)

Figure 2-9 Redefined Grid

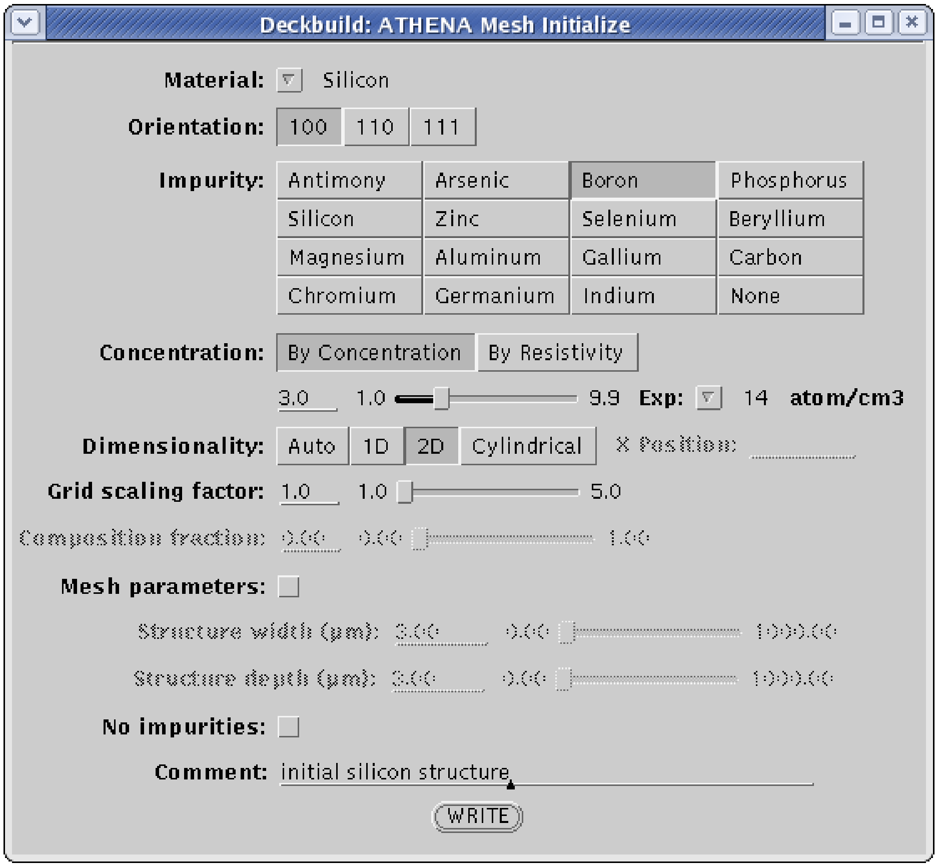
Finally, write the Mesh Define information to the file by pressing the **Write** button. A set of lines like these will appear:

```
GO ATHENA

# NON-UNIFORM GRID

LINE X LOC=0.00 SPAC=0.1

LINE X LOC=0.3 SPAC=0.02

LINE X LOC=1 SPAC=0.1

LINE Y LOC=0.00 SPAC=0.03

LINE Y LOC=0.2 SPAC=0.02

LINE Y LOC=1 SPAC=0.1
```

The first line (`GO ATHENA`) is called an `autointerface` statement and tells DeckBuild that the following file should be run by Athena.

Defining the Initial Substrate

The `LINE` statements specified by the **Mesh Define** menu set only the rectangular base for the Athena simulation structure. The next step is the initialization of the substrate region with its points, nodes, triangles, background doping, substrate orientation, and some additional parameters. To initialize the simulation structure, select **ATHENA Command Menu→Mesh Initialize...** and the Mesh Initialize Menu will appear (see Figure 2-10) .



Deckbuild: ATHENA Mesh Initialize

Material: ▿ Silicon

Orientation: [✓ 100] 110 111

Impurity:			
Antimony	Arsenic	[✓ Boron]	Phosphorus
Silicon	Zinc	Selenium	Beryllium
Magnesium	Aluminum	Gallium	Carbon
Chromium	Germanium	Indium	None

Concentration: [✓ By Concentration] By Resistivity

3.0 1.0 9.9 Exp: ▿ 14 atom/cm3

Dimensionality: Auto 1D [✓ 2D] Cylindrical X Position:

Grid scaling factor: 1.0 1.0 5.0

Composition fraction: 0.00 0.00 1.00

Mesh parameters: □

Structure width (µm): 3.00 0.00 1000.00

Structure depth (µm): 3.00 0.00 1000.00

No impurities: □

Comment: initial silicon structure

WRITE

Figure 2-10 Mesh Initialize Menu

Background doping can be set by clicking on the desired impurity box (e.g., Boron). The background impurity concentration specification will then become active. If the **None** box is checked, the concentration information will become inactive and will appear grayed out from the rest of the menu. Select the desired concentration using the slider (e.g., 3.0) and select an exponent from the **Exp:** menu (e.g., 14). This will give a background concentration of 3.0e14 atom/cm3. You can set background concentration using the **By Resistivity** specification in Ohm•cm. For this tutorial, check the **2D box** in the **Dimensionality** field. This will run the simulation in a two-dimensional calculation. The other items in this menu will be discussed in Section 2.8 “Using Advanced Features of Athena”.

Note: Two-dimensional mode is used in this tutorial to demonstrate **2D** grid generation and manipulation. In most cases, however, it is unnecessary to change the Auto default in the **Dimensionality** item of the **Mesh Initialize** menu. Athena will begin in **1D** and will automatically switch to **2D** mode at the first statement, which disrupts the lateral uniformity of the device structure. This generally results in massive savings of computation time.

You can now write the mesh initialization information into the file by pressing the **Write** button. The following two lines will appear in the DeckBuild Text Subwindow:

```
# INITIAL SILICON STRUCTURE

INIT SILICON C.BORON=3.0E14 ORIENTATION=100 TWO.D
```

Now, run Athena to obtain the initial structure. Press the **Run** button on the DeckBuild control. The following output will appear in the simulator subwindow:

```
ATHENA> # NON-UNIFORM GRID

ATHENA> LINE X LOC=0.00 SPAC=0.10

ATHENA> LINE X LOC=0.3 SPAC=0.02

ATHENA> LINE X LOC=1 SPAC=0.1

ATHENA> #

ATHENA> LINE Y LOC=0.00 SPAC=0.03

ATHENA> LINE Y LOC=0.2 SPAC=0.02

ATHENA> LINE Y LOC=1 SPAC=0.1

ATHENA> # INITIAL SILICON STRUCTURE

ATHENA> INIT SILICON C.BORON=3.0E14 ORIENTATION=100 TWO.D

ATHENA> STRUCT OUTFILE=.history01.str
```

The line `STRUCT OUTFILE=.history01.str` is automatically produced by DeckBuild through the history function. This function provides an important service when debugging new files, performing “what if” simulations, and visualizing the structure at different steps of simulation. This feature will be used throughout the tutorial. Use any of the three methods to visualize the initial structure:

1. Click on the **Tools** menu button. DeckBuild will then automatically save a temporary standard structure file and invoke TonyPlot with this file.
2. Click on the **Main Control** button and the DeckBuild Main Control popup will appear. Then, click on the **Plot Current Structure** button. DeckBuild will then automatically save a temporary standard structure file and invoke TonyPlot with this file.
3. Select (highlight) the name of a structure file (`.history01.str` in this case) and click on the **Tools** or **Plot Current Structure**. DeckBuild will then start TonyPlot with the selected structure file.

After a short delay, TonyPlot will appear. It will have only regional and material information. Click on the **Plot** menu button and the Display (2D Mesh) popup will appear. Select only the two left icons: **Mesh** and **Edges** and the Initial Triangular Grid (Figure 2-11) will appear in TonyPlot.

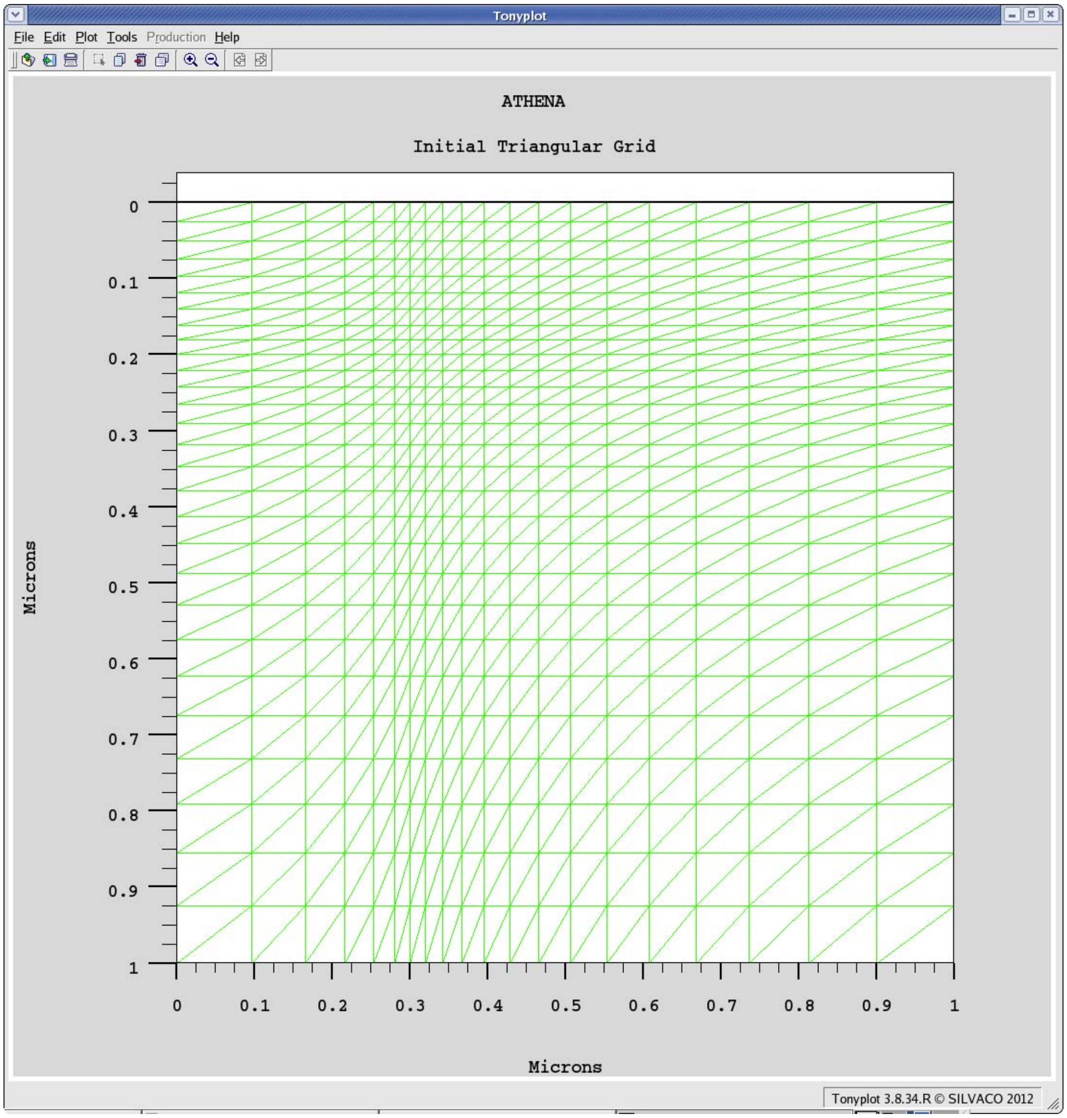


Figure 2-11 Initial Triangular Grid

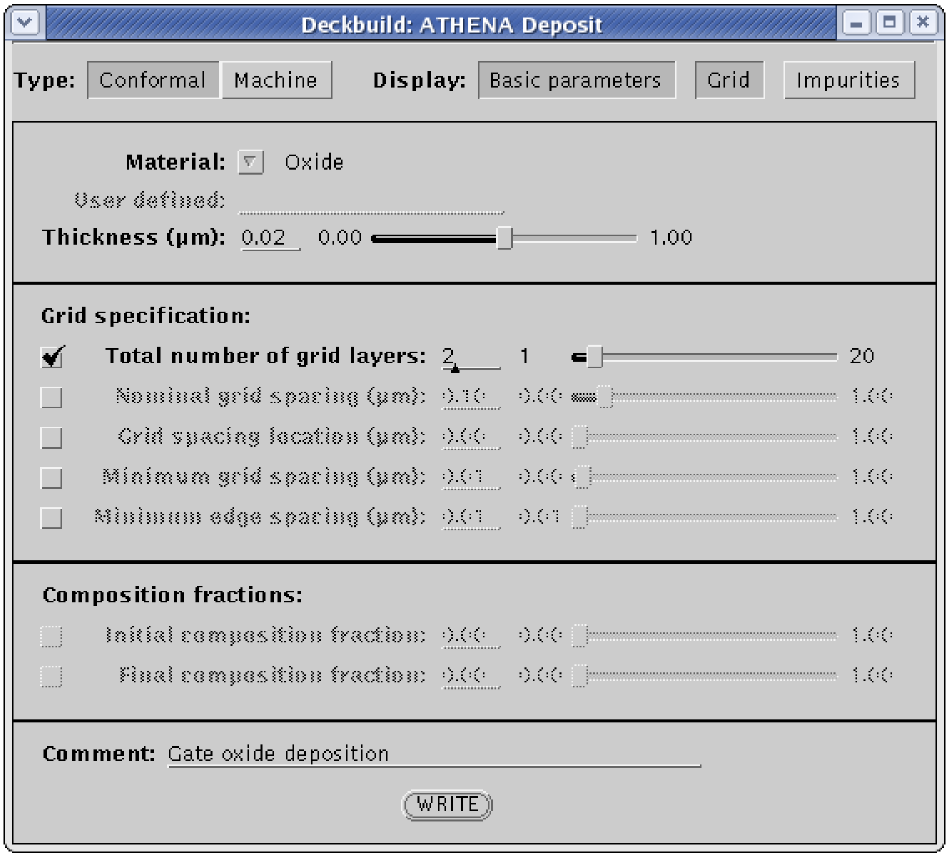
The grid in Athena consists of points connected to form a number of triangles. Each point has one or more nodes associated with it. A point within a material region has one node, while a point which belongs to several regions has several nodes. A node represents the solution (e.g., doping concentration) in a particular material region at the point. For example, a given node may represent solution values in silicon at a point with coordinates (0.0, 0.0); an entirely different node may represent solution values in oxide at the same point (0.0, 0.0).

So, the previous `INIT` statement creates the <100> silicon region of 1.0 µm x 1.0 µm size, which is uniformly doped with boron concentration of 3e14 atom/cm^3. This simulation structure is ready for any process step (e.g., implant, diffusion, Reactive Ion Etching). Before discussing the simulation of physical processing using SSuprem4**,** Elite or OPTOLITH modules, it's important to discuss structure manipulation statements that can precede or alternate with physical process steps.

Simple Film Depositions

Conformal deposition can be used to generate multi-layered structures. Conformal deposition is the simplest deposit model and can be used in all cases when the exact shape of the deposited layer is not critical. Conformal deposition can also be used in place of oxidation of planar or quasi-planar semiconductor regions when doping redistribution during the oxidation process is negligible.

To set the conformal deposition step, select the menu items **Process→Deposit→Deposit...** from the **Commands** menu in DeckBuild and the Athena Deposit Menu (Figure 2-12) will appear.



Deckbuild: ATHENA Deposit

Type: Conformal Machine Display: Basic parameters Grid Impurities

Material: Oxide

User defined:

Thickness (µm): 0.02 0.00 1.00

Grid specification:

☑ Total number of grid layers: 2 1 20

□ Nominal grid spacing (µm): 0.10 0.00 1.00

□ Grid spacing location (µm): 0.00 0.00 1.00

□ Minimum grid spacing (µm): 0.01 0.00 1.00

□ Minimum edge spacing (µm): 0.01 0.01 1.00

Composition fractions:

□ Initial composition fraction: 0.00 0.00 1.00

□ Final composition fraction: 0.00 0.00 1.00

Comment: Gate oxide deposition

WRITE

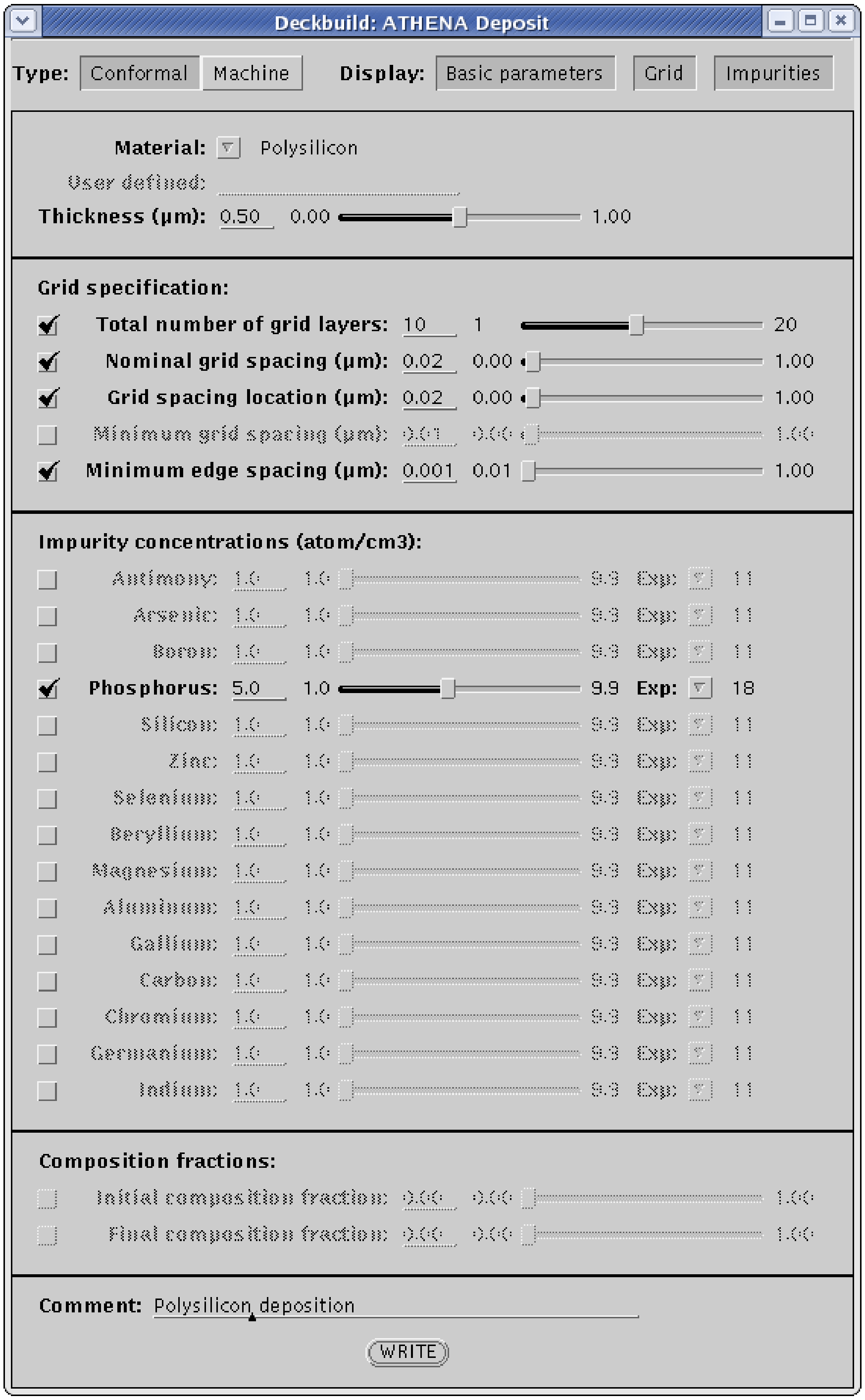
Figure 2-12 Athena Deposit Menu

As shown, Conformal Deposition is the default. If it is known that the oxide layer thickness grown in a process is 200 Angstroms, you can substitute this with conformal oxide deposition. Select **Oxide** from the Material menu and set its thickness to 0.02 µm. It is always useful to set several grid layers in a deposited layer. In this case, at least two grid layers are needed to simulate impurity transport through the oxide layer. In some other cases (e.g., photoresist deposition over a non-planar structure), a sufficiently fine grid is needed to accurately simulate processes within the deposited layer. There are also situations (e.g., spacer formation) when several grid layers in a deposited material region are needed to properly represent the geometrical shape of the region.

The grid in the deposited layer is controlled by **Grid Specification** parameters in the Athena Deposit Menu. Set the **Total number of grid layers** to 2, add a **Comment**, and click on the **Write** button. The following lines will then appear in the DeckBuild Text Subwindow:

```
# GATE OXIDE DEPOSITION

DEPOSIT OXIDE THICK=0.02 DIVISIONS=2
```

The next step will be to deposit a phosphorus doped polysilicon layer of 0.5µm thickness. Select the material Polysilicon, and set the thickness to 0.5. To add doping, select the **Impurities** box. The **Impurity Concentration** section will be immediately added to the Athena Deposit Menu (See Figure 2-13).


Deckbuild: ATHENA Deposit
Type: Conformal Machine
Display: Basic parameters Grid Impurities
WRITE

Material: Polysilicon

User defined:

Thickness (µm): 0.50 0.00 1.00

Grid specification:

☑	Total number of grid layers:	10	1	20
☑	Nominal grid spacing (µm):	0.02	0.00	1.00
☑	Grid spacing location (µm):	0.02	0.00	1.00
□	Minimum grid spacing (µm):	0.01	0.00	1.00
☑	Minimum edge spacing (µm):	0.001	0.01	1.00

Impurity concentrations (atom/cm3):

□	Antimony:	1.0	1.0	9.9	Exp:	11
□	Arsenic:	1.0	1.0	9.9	Exp:	11
□	Boron:	1.0	1.0	9.9	Exp:	11
☑	Phosphorus:	5.0	1.0	9.9	Exp:	18
□	Silicon:	1.0	1.0	9.9	Exp:	11
□	Zinc:	1.0	1.0	9.9	Exp:	11
□	Selenium:	1.0	1.0	9.9	Exp:	11
□	Beryllium:	1.0	1.0	9.9	Exp:	11
□	Magnesium:	1.0	1.0	9.9	Exp:	11
□	Aluminum:	1.0	1.0	9.9	Exp:	11
□	Gallium:	1.0	1.0	9.9	Exp:	11
□	Carbon:	1.0	1.0	9.9	Exp:	11
□	Chromium:	1.0	1.0	9.9	Exp:	11
□	Germanium:	1.0	1.0	9.9	Exp:	11
□	Indium:	1.0	1.0	9.9	Exp:	11

Composition fractions:

□	Initial composition fraction:	0.00	0.00	1.00
□	Final composition fraction:	0.00	0.00	1.00

Comment: Polysilicon deposition

Figure 2-13 Impurity Section of the Athena Deposit Menu

Click on the **Phosphorus** checkbox and set the doping level (e.g., 5.0×10^{19}) using the slider and the **Exp** menu. You can set a non-uniform grid in the deposited layer by changing the **Nominal grid spacing** and the **Grid spacing location** parameters. To create a finer grid at the polysilicon surface, set the total number of grid layers to 10, the **Nominal grid spacing** to 0.02 µm and the **Grid spacing location** to 0.0 (at the surface). Then, click on the **Write** button and the following deposition statement will be written in the input file as:

```
DEPOSIT POLY THICK=0.5 C.PHOSPHOR=5.0E19 DIVISIONS=10 \
        DY=0.02 YDY=0.0 MIN.SPACING=0.001
```

Use the **Cont** button to continue the Athena simulation. This will create the three layer structure shown in the left plot of Figure 2-14. The MIN.SPACING parameter preserves the horizontal mesh spacing for high aspect ratio grids. Athena tries to reduce high aspect ratio grids and MIN.SPACING stops this. To get a finer grid not at the polysilicon surface but in the middle of polysilicon layer, change YDY to 0.2. This puts on a finer grid at a distance of 0.2µm from the surface of the structure. You can do this by positioning the cursor in the input file and backspacing over existing text, or entering new text. For example:

```
DEPOSIT POLY THICK=0.5 C.PHOSPHOR=5.0E19 DIVISIONS=10 \
       DY=0.02 YDY=0.2
```

It is possible to see the effect of changing the YDY. parameter within the polysilicon without rerunning the whole input file. To do this, highlight the previous statement (DEPOSIT OXIDE...), select **Main Control**→**Init from History** button, and press the **Cont** button. The new history file can then be loaded into TonyPlot (see the right plot in Figure 2-14).

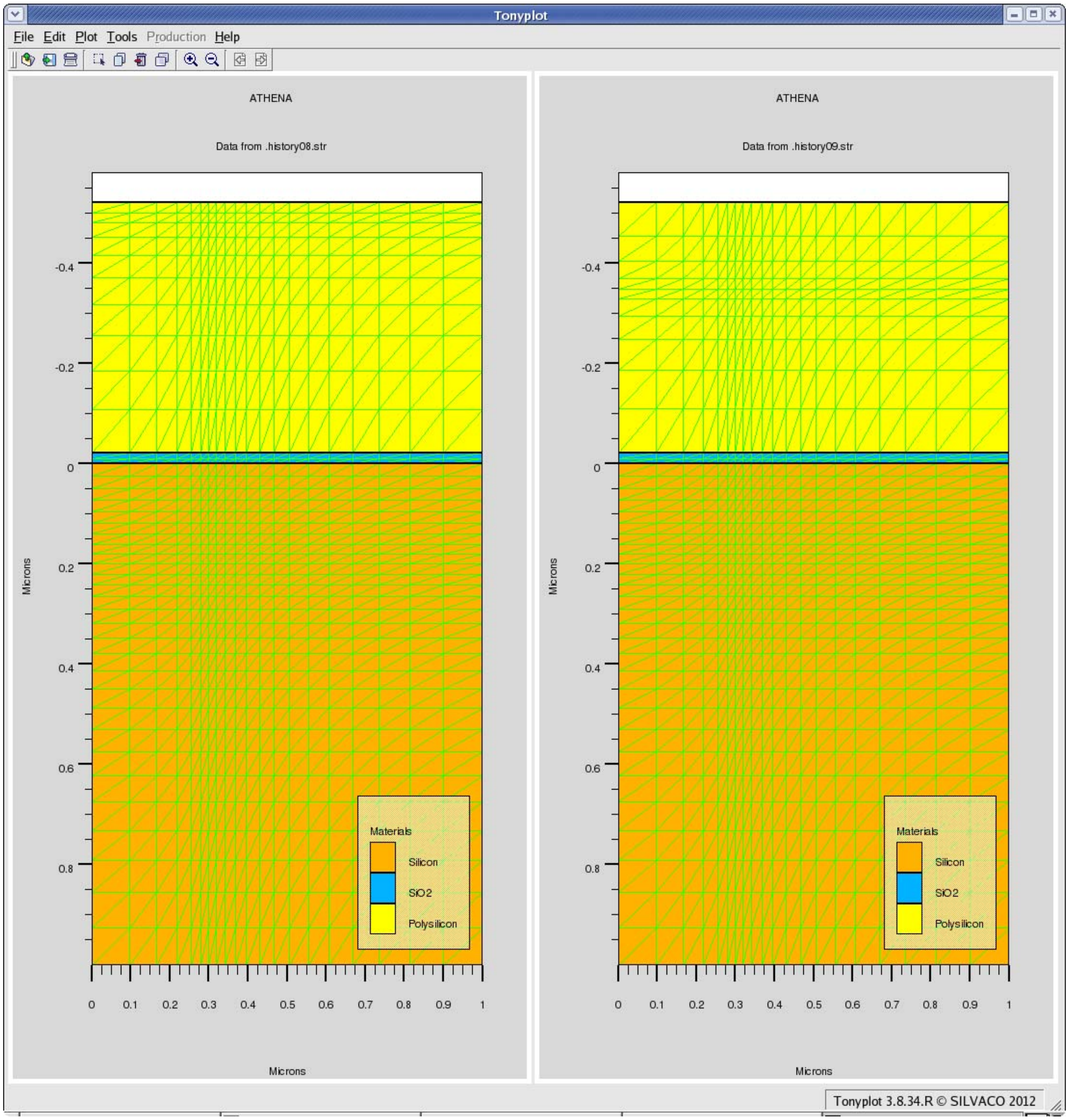
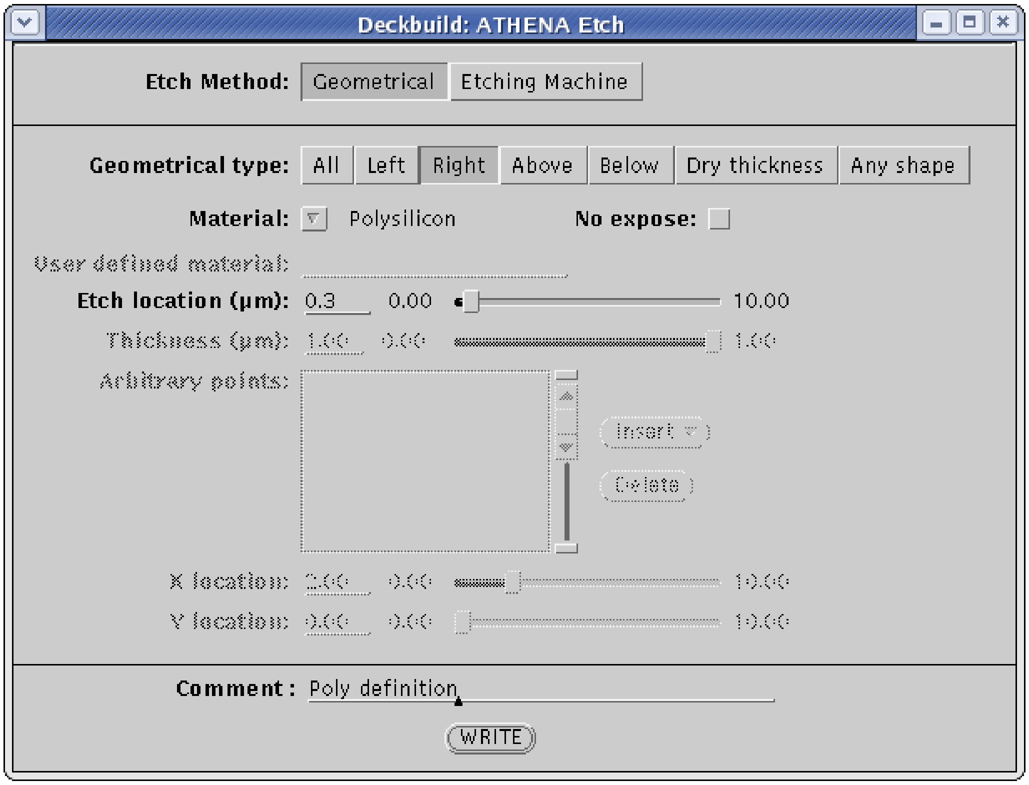


Figure 2-14 Grid Control for Deposition

Simple Geometrical Etches

The next step in this tutorial process simulation is to define the polysilicon gate definition. (Implant and thermal steps will be discussed in Section 2.4 “Choosing Models In SSuprem4”). To set a geometrical etch step, select **Process→Etch→Etch...** from the Command menu of DeckBuild. The Athena Etch Menu (Figure 2-15) will appear.



Etch Method: Geometrical

Geometrical type: Right

Material: Polysilicon

No expose: □

User defined material:

Etch location (µm): 0.3 0.00 10.00

Thickness (µm): 1.00 0.00 1.00

Arbitrary points:

X location: 2.00 0.00 10.00

Y location: 0.00 0.00 10.00

Comment: Poly definition

Figure 2-15 Athena Etch Menu

The **Geometrical** etch is the default method. Other methods will be discussed in Section 2.8.2 “Deposition and Wet/Dry Etching using the Physical Models in Athena/Elite”. Select **Polysilicon** from the **Material** menu. This example will use a polysilicon gate edge at x=0.3 and set the center of the gate at x=0.0 for the initial grid. Therefore, polysilicon should be etched to the right from x=0.3. To do so, select **Right** from the **Geometrical type**, and set the **Etch location** to 0.3. This will give the following statement:

```
# POLY DEFINITION
ETCH POLY RIGHT P1.X=0.3
```

The structure created by this `ETCH` statement is shown in the left hand plot of Figure 2-16.

You can obtain an arbitrary shape of geometrical etching by using the **Any Shape** button. For example, to make a tilted etch, specify X and Y locations of four **Arbitrary points** as shown in Figure 2-17.

The following four etch lines will be inserted into the input file:

```
# POLY DEFINITION
ETCH POLY START X=0.2 Y=-1
ETCH CONT X=0.4 Y=1
ETCH CONT X=2 Y=1
ETCH DONE X=2 Y=-1
```

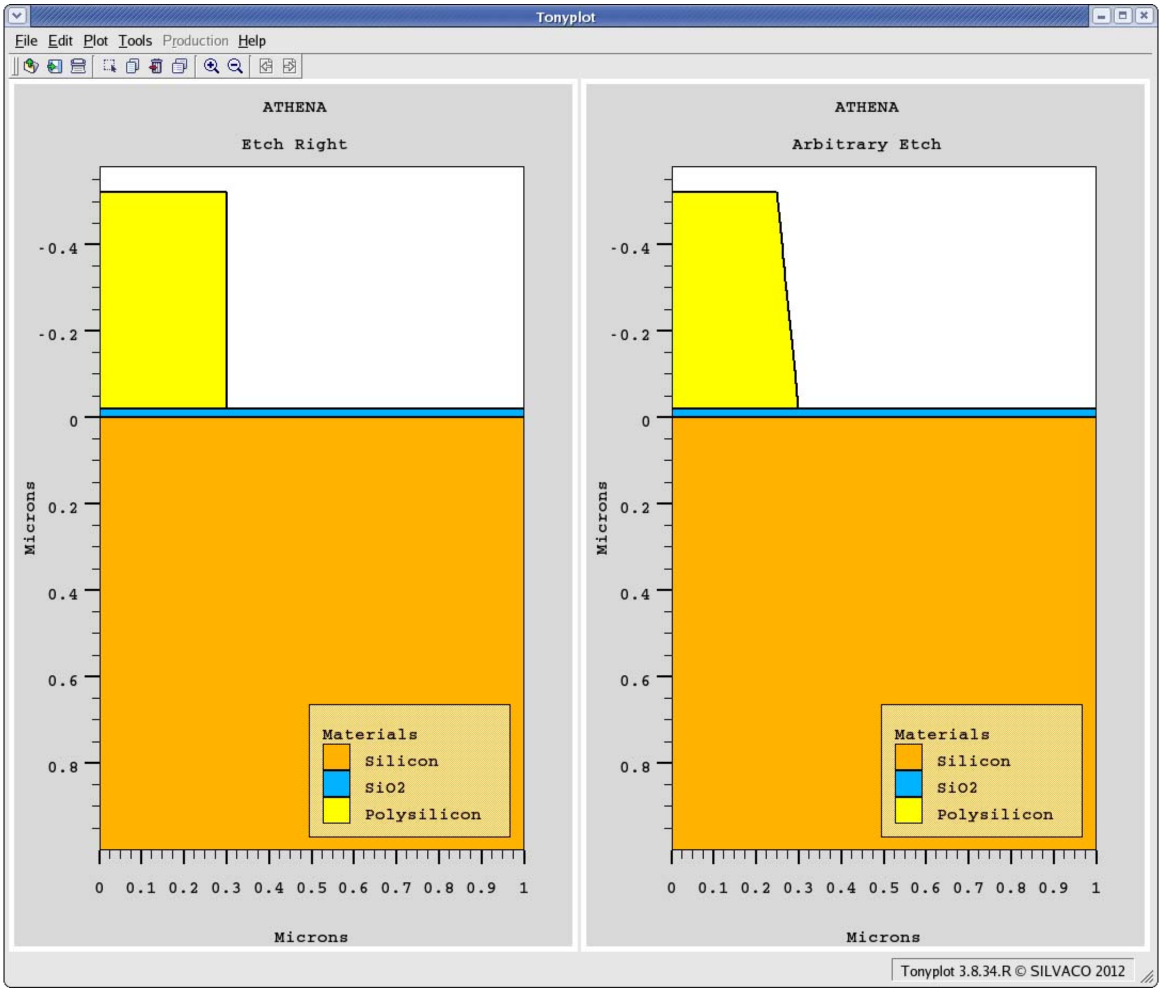
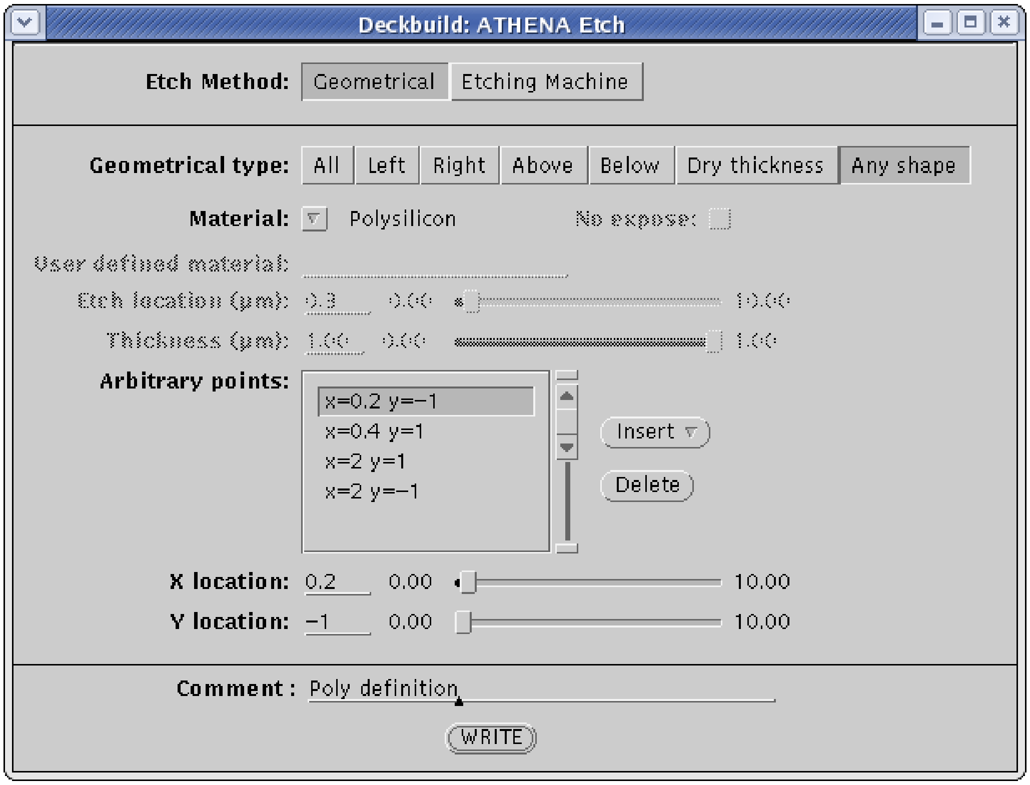


Figure 2-16 Structure Created by Etch Statement



Etch Method: Geometrical

Geometrical type: Any shape

Material: Polysilicon

No expose: □

User defined material:

Etch location (µm): 0.3 0.00 10.00

Thickness (µm): 1.00 0.00 1.00

Arbitrary points:

x=0.2 y=-1
x=0.4 y=1
x=2 y=1
x=2 y=-1

X location: 0.2 0.00 10.00

Y location: -1 0.00 10.00

Comment : Poly definition

Figure 2-17 Arbitrary Etch

If this input file fragment is runned instead of the previous one (using the INIT statement from the **History** capability), the structure after this etch sequence will appear as displayed in the right hand plot in Figure 2-16. Athena etches all polysilicon material within the specified polygon. The polygon etch can consist of any number of points. If you use the **Insert** button, an additional point will appear after the currently selected point.

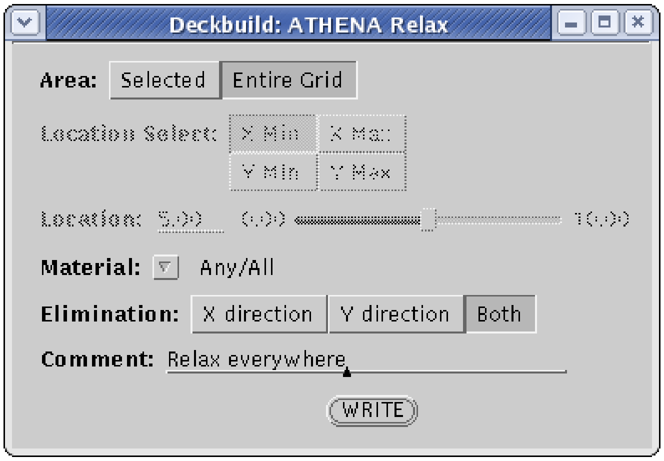
An additional option for geometrical etching is a dry etch with a specified thickness. This can be used for spacer formation as follows: deposit an oxide of a specified thickness (e.g., 0.2µm) and then etch the same thickness again.

```
# CLEAN GATE OXIDE

ETCH OXIDE DRY THICK=0.03

# SPACER DEPOSITION

DEPOSIT OXIDE THICK=0.2 DIVISIONS=8

# SPACER ETCHING

ETCH OXIDE DRY THICK=0.23
```

The dry etching step etches the specified material in the region between the top (exposed) boundary of the structure and a line obtained by translating the boundary line down in the Y direction. The etch distance is specified by the THICK parameter. Figure 2-19 shows the resulting spacer.

Reducing Grid Points in Non-Essential Areas using the Relax Parameter

The previous sections demonstrate that the quality of the grid is extremely important for Athena simulation. The rectangular based grid generated by the `INITIALIZE` or `DEPOSIT` statements may remain intact in those areas not involved in the process steps affecting the grid (e.g., etching or oxidation). The **Grid Relax** capability allows the spacing to be increased in such areas at any point during the simulation. This capability is useful for two reasons. First, the initial small spacings are propagated throughout the structure. For example, the fine grid in the X-direction shown in Figure 2-9 may be needed only in the upper portion of the structure where doping occurs. Eliminating some grid lines and points in the lower portion of the structure will not affect the accuracy of implant and diffusion simulation. Second, it is always necessary to set a fine grid in the area where ion implantation takes place, but the fine grid may be uneccessary after the profile is leveled-off during thermal steps. So, relaxation of an initially fine grid may save simulation time during subsequent steps. Parameters for the `RELAX` statement are set from the Athena Relax Menu (Figure 2-18).



Area: Selected [✓ Entire Grid]

Location Select:

Location: 5.00 0.00 10.00

Material: Any/All

Elimination: X direction Y direction [✓ Both]

Comment: Relax everywhere

Figure 2-18 Athena Relax Menu

To open this menu, select **Structure→Relax...** in the DeckBuild Commands menu. You can preform grid relaxation over the whole structure if you select **Entire Grid**, or within a selected rectangular area, if you choose **Selected** and specify **Xmin**, **Xmax**, **Ymin**, and **Ymax** in **Location Select**. Selecting a material from the **Material** menu specifies which material region will be affected by the grid relax operation. The default is all materials within the specified area. You can perform the grid line elimination either in one direction or in both directions by selecting **X direction**, **Y direction**, or **Both**. The X direction, `Relax`, cannot be performed for individual materials except for the substrate.

To understand how the relax function changes a grid, we will use the structure we have obtained after spacer formation was specified (Figure 2-19). If we relax the entire grid in both directions (Figure 2-20), the following lines will be inserted into the tutorial input file:

```
# RELAX EVERYWHERE

RELAX DIR.X=T DIR.Y=T
```

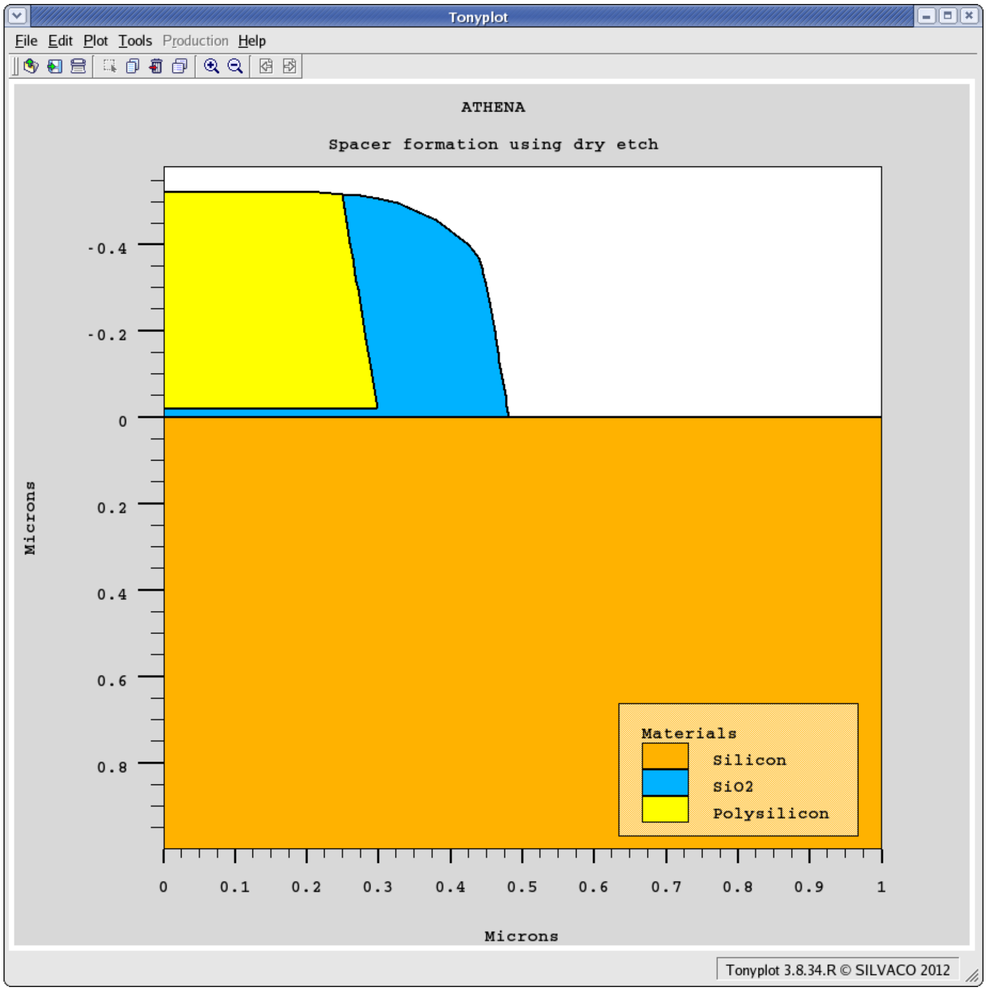


Figure 2-19 Spacer Formation using Dry Etch

The resultant grid is shown in the upper-right corner of Figure 2-20.

The total number of grid points is reduced from 708 to 388. When comparing with the grid before relaxation (upper-left corner of Figure 2-20), note that the grid within the oxide spacer and polygate has not changed. This is due to three factors:

- the relax algorithm works only with rectangular base grid,
- it never eliminates grid lines adjacent to a region boundary,
- the relaxed area should be at least five by five grid points.

Within silicon, each second horizontal line is eliminated. The lower part of each second vertical line is also eliminated. This happens because the algorithm doesn't allow the formation of obtuse triangles.

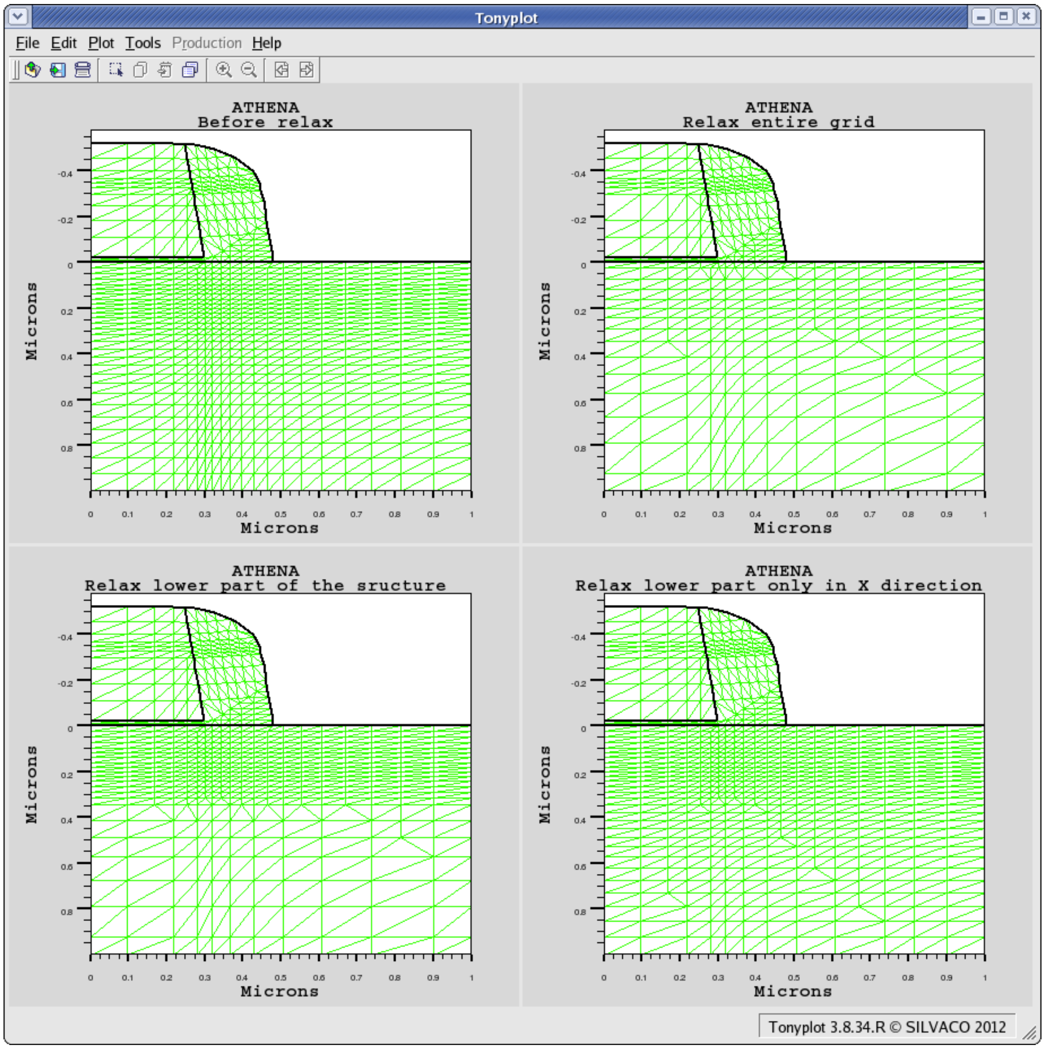

Tonyplot
File Edit Plot Tools Production Help
ATHENA
Before relax
ATHENA
Relax entire grid
ATHENA
Relax lower part of the sructure
ATHENA
Relax lower part only in X direction
Microns
-0.4
-0.2
0
0.2
0.4
0.6
0.8
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1
Tonyplot 3.8.34.R © SILVACO 2012

Figure 2-20 Grids after Various Relax Operations

If you don't want the grid to be relaxed above y=0.3, choose **Selected** and set all four boundaries in the Relax box. This will give the following `RELAX` statement:

```
# RELAX LOWER HALF OF THE STRUCTURE

RELAX X.MIN=0.00 X.MAX=1.00 Y.MIN=0.3 Y.MAX=1.00 DIR.X=T DIR.Y=T
```

In this case, the number of grid points is 567. The grid above y=0.3 remains intact (see the plot in lower-left corner of Figure 2-20) and the elimination in X and Y- directions happens only below y=0.3.

To increase spacing only in the X-direction in the area below y=0.3, select the **X direction** and leave the **Area** and **Location Selections** as before. This will give the following `Relax` statement:

```
 #  RELAX  LOWER  HALF  ONLY  IN  X-DIRECTION  RELAX  X.MIN=0.00
X.MAX=1.00 Y.MIN=0.3 Y.MAX=1.00 DIR.X=T DIR.Y=F
```

Note: The only difference is that instead of DIR.Y=T, the statement contains DIR.Y=F, which prevents elimination in Y-direction. This gives 638 grid points and a different pattern of elimination (see the plot in the lower-right corner of Figure 2-20).

You can also apply several consequent `RELAX` statements to achieve grid elimination in different areas of the structure.

An important thing to remember about the **RELAX** capability is that it allows you to avoid creating obtuse triangles and avoid relaxing directly on the material boundaries. This sometimes results in no relaxation or grid relaxation in a subset of the desired area. The most desirable method for complete control over gridding is by using DevEdit as described briefly in this chapter and in the DevEdit User's Manual.

Reflecting a Structure in the "Y" Plane using the Mirror Parameter

This tutorial process simulation has been building one half of a MOSFET-like structure. At some point in the simulation, you will need to obtain the full structure. This must be done before exporting the structure to a device simulator or setting electrode names. In general, structure reflection should be performed when the structure ceases to be symmetrical (e.g., a tilted implant, an asymmetrical etching, or a deposition takes place), or when a reflecting boundary condition no longer applies to the side, which is going to be the center of the structure.

This example will explain how to mirror the structure at its left boundary. To mirror the structure, select **Structure→Mirror** in the Commands menu (Figure 2-21).

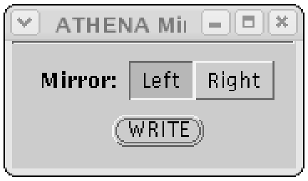


Figure 2-21 Athena Mirror Menu

Then, press the **Write** button to write the following statement to the input file:

```
STRUCT MIRROR LEFT
```

The resulting structure is shown in Figure 2-22.

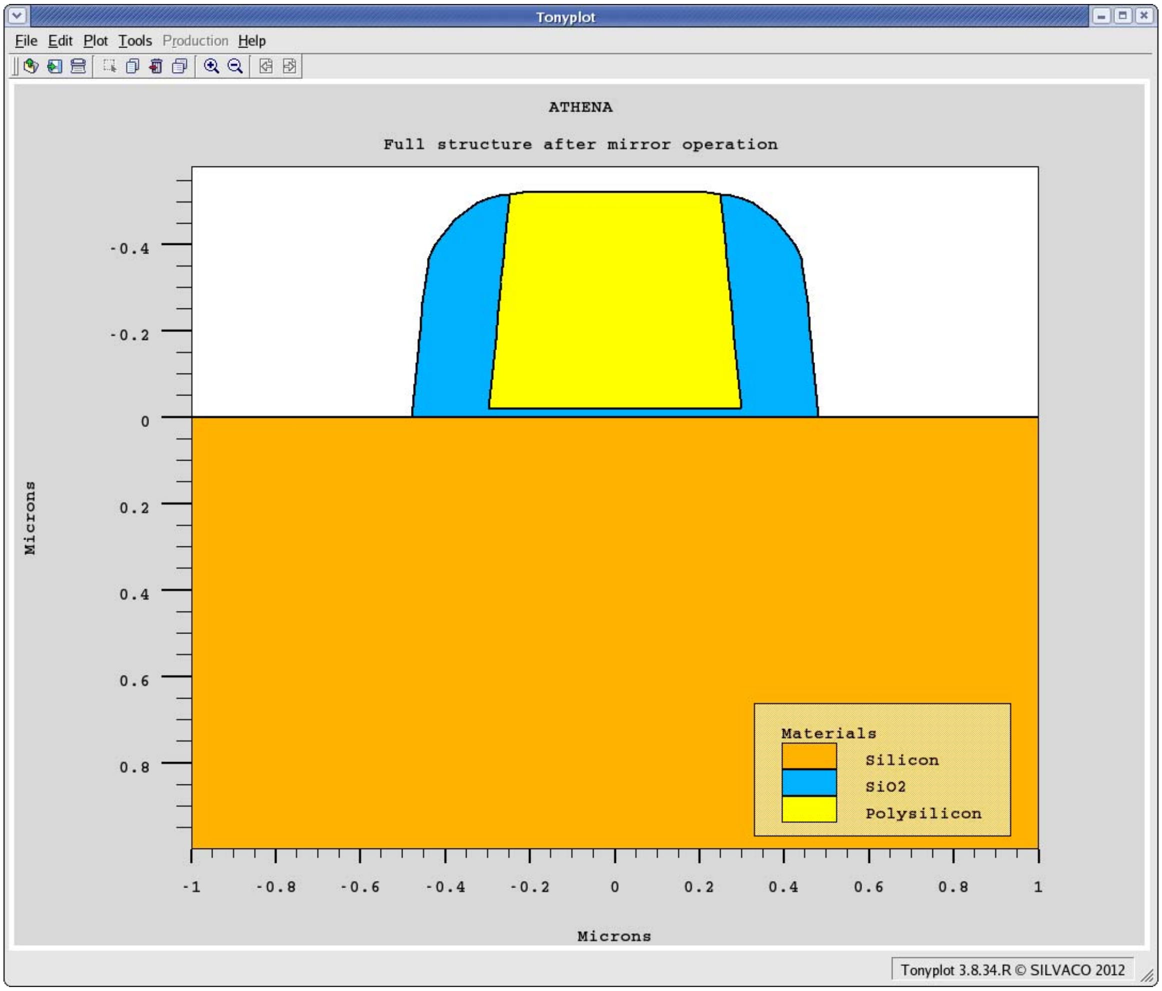


Figure 2-22 Athena Reflect Capability

The left half of the structure is a complete mirror copy of the right part, including node coordinates, doping values, and so on. Beware of rounding errors when mirroring. If the boundary of reflection is not smooth to within 0.1 angstroms, some points will be duplicated.

Specification of Electrodes in Athena

The ultimate goal of an Athena simulation is usually to create a device structure (material layers plus doping), which then can be used by a device simulator (usually Atlas) for electrical characterization. Although Atlas is able to specify the locations of electrodes, in many cases specifying electrodes must be done in Athena. For example, it is impossible to specify an electrode location in Atlas when the electrode does not consist of straight segments. Also, when specifying electrodes in Athena, it is useful to transfer electrode layer information from layout to electrical tests in a device simulator (see the description of the **auto-electrode** capability in the MaskViews User’s manual).

Athena can attribute an electrode to any metal, silicide, or polysilicon region. A special case is the backside electrode, which can be placed at the bottom of the structure without having a metal region there. If you deposit 0.1 µm aluminum layer on the full structure after reflection (Figure 2-22) using:

```
DEPOSIT ALUMIN THICK=0.1
```

and etched the following part of the layer between x=-0.8 and x=0.8, using the **Any Shape** specification in the Athena Etch Menu (See Figure 2-15):

```
ETCH ALUMINUM START X=-0.8 Y=-20
ETCH CONT X=-0.8 Y=20
ETCH CONT X=0.8 Y=20
ETCH DONE X=0.8 Y=-20
```

you will now have the structure shown in Figure 2-23.

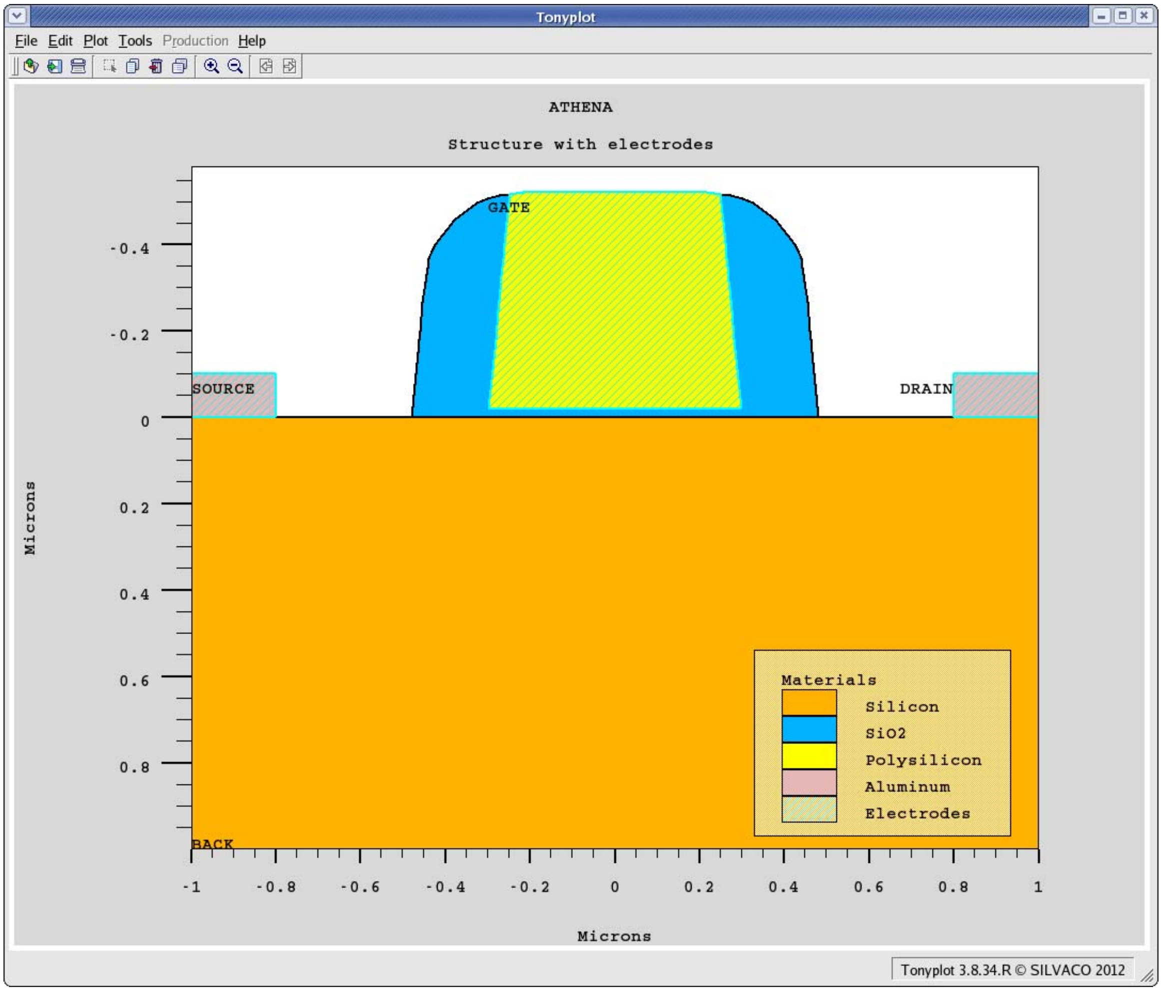


Figure 2-23 MOSFET Structure with Electrodes

You can now use the Athena Electrode menu (see Figure 2-24) by selecting **Commands→Structure→ Electrode...**. To set an electrode at a specified position, select the **Specified Position** button, type in the **X Position** (e.g., -0.9) and **Name** (for example, source), and press the **Write** button. The following statement will appear in the input file:

```
ELECTRODE NAME=SOURCE X=-0.9
```

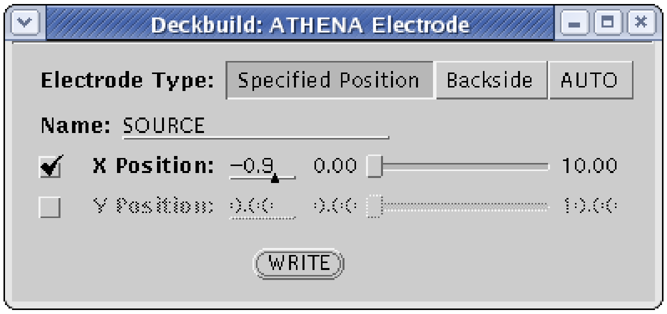
Similarly, specify the drain electrode:

```
ELECTRODE NAME=DRAIN X=0.9
```

The polysilicon gate electrode specification has the same format. For this structure it can be done the same way as for source or drain:

```
ELECTRODE NAME=GATE X=0.0
```

If the polysilicon layer is not the topmost layer at x=0, the **Y Position** can be specified. In this case, check the **Y Position** checkbox and type in a y coordinate within the polygate layer (e.g., -0.2). If Y is not specified and the electrode is not on top, Athena will look for the electrode in the underlying layers. If it fails, an error will be reported. To specify a backside electrode, select **Backside** from the **Electrode Type** field and type in a name (see also Figure 2-24) .



Electrode Type: Specified Position | Backside | AUTO

Name: SOURCE

☑ X Position: -0.9 0.00 10.00

□ Y Position: 0.00 0.00 10.00

Figure 2-24 Athena Electrode Menu

The following backside electrode statement will appear in the input file:

```
ELECTRODE NAME=BACK BACKSIDE
```

If an electrode name is not specified, DeckBuild issues the error message: `NO ELECTRODE NAME SPECIFIED` and the command is not written to the input file.

If an incorrect position for electrode is specified, for example:

```
ELECTRODE NAME=JUNK X=0.6
```

Athena will output the following warning message:

```
Cannot find the electrode for this structure.
Electrode statement ignored and ignores the statement.
```

Saving a Structure File for Plotting or Initializing an Athena Input file for Further Processing

As mentioned in the “Standard Structure File Format” on 27, the DeckBuild history function saves structure files after each process step. In many cases, however, you need to save and initialize structures independently. There are several reason why it’s needed to save and initialize structures independently.

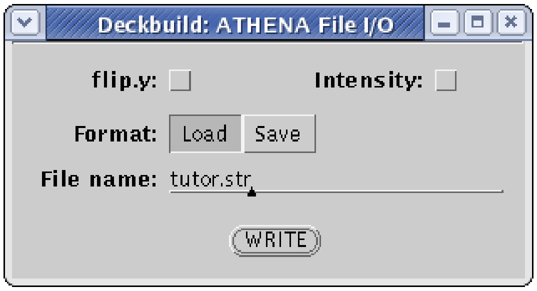
The first reason is because the stack for the history files is limited (50 by default). The second reason is because it is usually undesirable to keep dozens of history files on disc (each of which occupy hundreds of Kbytes) after the DeckBuild session ends. The third reason is because users often want to save the structure information generated after key process steps (e.g., final structure).

To save or load a structure, use the Athena File I/O Menu (See Figure 2-25) by selecting **Commands→File I/O...**. Specify a file name (the file extension .str is recommended for all Athena structure files) and press the **Save** button. The following line will appear in the input file:

```
STRUCT OUTFILE=TUTOR.STR
```

You can reload this file (tutor.str) back into Athena at any time during the current DeckBuild session or during any subsequent session. To reload the structure file, press the **Load** button on the Athena File I/O Menu. The following INIT statement will appear:

```
INIT INFILE=TUTOR.STR
```


Deckbuild: ATHENA File I/O

flip.y: □ Intensity: □

Format: Load Save

File name: tutor.str

Figure 2-25 Athena File I/O Menu

Note: Only the structure will be reloaded if Athena is restarted before this INIT statement. Any parameters or coefficients that were set during previous simulations must be reset if they are needed. This structure file can also be used by any device simulator or DevEdit.

2.4 Choosing Models In SSuprem4

2.4.1 Implantation, Oxidation, RTA, Diffusion and Epitaxy

This section describes how to simulate process steps (e.g., implantation, diffusion/oxidation, epitaxy, and silicidation) specific to the SSuprem4 module of Athena. Also discussed, are the SSuprem4 model statements, such as `METHOD`, `OXIDE`, `MATERIAL`, and `IMPURITY`.

For more information about SSuprem4, see Chapter 3 “SSuprem4 Models”. For more information about the `MODEL` statements, see Chapter 6 “Statements”.

When simulating any process involving dopant or its diffusion or both, it is absolutely critical for simulation accuracy to use the appropriate model. Process steps where correct choice of models are vital include implantation, diffusion, rapid thermal annealing, oxidation, and epitaxy. This section provides specific advice on which models should be used for each process step.

2.4.2 The Reason for Multiple Models for Each Process

The key to simulating any dopant related process is to accurately account for damage in the semiconductor. For example, in silicon processing, typical implantation doses can cause sufficient damage to the substrate to enhance dopant diffusion rates by three orders of magnitude or more, so choosing the wrong model in this instance will result in inaccurate results.

Well known device anomalies such as the Reverse Short Channel Effect in MOS processing or the emitter push effect in bipolar processing are wholly the result of such damage enhanced diffusion.

Other processes that consume the semiconductor, such as oxidation and silicidation also inject damage into the substrate. This must be accounted for if accurate dopant profiles are a requirement.

This section aims to provide you with a set of rules outlined, indicating the correct model that can be used most of the time without you having a detailed knowledge of the physics involved. The usual rules of model selection apply here. The more complicated the model, the greater the simulation time. There is always a compromise between simulation accuracy and simulation time. The following sections describe when to use the hierarchy of models so that the most complicated models are only used when you make a significant difference to the result.

2.4.3 Choosing an Appropriate Model Using the Method Statement

The hierarchy of diffusion and damage models available is broadly related to the maximum level of damage already in the semiconductor or the maximum level of damage that the next process step is likely to introduce at any particular time during the process flow. The level of damage in the semiconductor at any one time is not a static quantity but will depend on when and how much damage was induced by a process step and how much annealing has occurred in subsequent thermal steps. The range of models available to you can account for all of the above effects and allows accurate simulation of dopant diffusion if appropriate models have been chosen.

The choice of model or combination of models for any of the process steps described above is defined in the `METHOD` statement. The `METHOD` statement serves a number of functions but in the context of defining damage models the `METHOD` statement is used for two purposes.

The first purpose is to specify models for how damage is induced during processes such as implantation or oxidation. The second purpose is to specify how that damage anneals and diffuses in subsequent or concurrent thermal processes.

It’s important to realize that the `METHOD` statement must be placed above the line, specifying the process step or steps to which it refers in the input file. Any number of method statements can be used in an input file allowing you to change the models at will during the process flow to optimize the speed and accuracy of the simulation. The models specified in the `METHOD` statement will hold true for all processes that follow it until it’s updated by a subsequent method statement.

Table 2-1 below indicates a recommended `method` statement for typical processes. It should be realized that these statements are hierarchical, so there is no accuracy lost if a more complicated model is used where a simpler one would suffice. The only downside here is a longer simulation time. The table below starts off with the simplest of models and progresses to the more complicated ones.

Table 2-1 Recommended Method Statements for Typical Processes

Method Statement Syntax	Suitability of using this method syntax
`method fermi`	Use only before undamaged silicon diffusions, where doping concentrations are less than 1e20/cm3 and no oxidizing ambient is present.
`method two.dim`	Use before implant doses less than 1e13/cm^2 and for oxidations.
`method full.cpl cluster.dam high.conc`	Use before implant doses greater than 1e13/cm^2.

2.4.4 Changing the Method Statement During the Process Flow

It has previously been stated that the disadvantage of using the most advanced and complex models is the time involved during diffusion cycle simulation. Accordingly, there is an incentive during complex process simulations to switch back to a simpler model during a diffusion cycle when the majority of the damage created by a previous implant has been annealed. We will show you when to switch to a simpler model.

If the process being modeled has involved implantation or oxidation at any stage, we advise not to use the fermi model. An exception to this would be in some power devices with very long diffusion times where the exact nature of surface damage would have little impact on the final distribution of the dopant and simulation time is at a premium.

In reality, for most small geometry processes, the question of switching models becomes one of when to add a new `method` statement that changes from:

```
METHOD FULL.CPL CLUSTER.DAM HIGH.CONC
```

to

```
METHOD TWO.DIM
```

after a high dose implant.

Switching guidelines

A simple guideline to follow when to switch method statements during a process flow is by switching back to the `TWO.DIM` model if the anneal temperature is greater than 900° and the device has been annealed for at least one minute, following an implant where the dose is greater than 1e13/cm^2.

For a more accurate guideline, see the “RTA Diffusion Modelling” on 145. Table 3-6 shows the anneal temperature/time combinations required for 95% of the clusters formed during high dose implants to dissolve. Modeling these dopant/defect clusters requires the fully coupled (`full.cpl`) and cluster damage (`cluster.dam`) models. Only when these clusters have dissolved can the `two.dim` model be used without significant loss of simulation accuracy. As a general rule, we recommend that the `method` statement be changed to `method two.dim` only after a diffusion time that is at least two or three times as long as the values quoted in the table.

If you wish to be certain of when it’s safe to switch models, the recommended procedure is to save a structure file at the point of interest, load the file into Tonyplot and perform a 1D cutline. Plot the clusters and interstitials. If the cluster concentration is still visible, it’s too early to switch models.

For power devices, where simulation time is at a premium, the same method already described should be used. But instead of using the cluster concentration as a guide of when to switch models, the interstitial concentration should be used as the guide as to when to switch models one more time from the `TWO.DIM` model to the basic `FERMI` model. When the interstitial concentration near the surface during a very long anneal has been reduced to only marginally above the background level at the anneal temperature concerned, the `method` statement can be switched to `METHOD FERMI` to greatly reduce the simulation time. The interstitial background level will be the level deep in the substrate where little damage has occurred.

2.4.5 Modelling the Correct Substrate Depth

An important and often overlooked aspect of the correct modeling of dopant diffusion is the choice of substrate depth. It has been mentioned previously that the rate of dopant diffusion is highly dependent on the level of damage in the substrate. Therefore, the accurate modeling of dopant diffusion requires the accurate modeling of substrate damage, particularly the movement of interstitials. In general, the interstitials created directly or indirectly by implantation and oxidation tend to diffuse much greater distances than the dopant. The substrate depth chosen for modeling purposes must therefore be deep enough to allow the interstitial concentrations to return to background levels at the bottom of the simulated substrate, even if no dopant diffusion occurs at this depth.

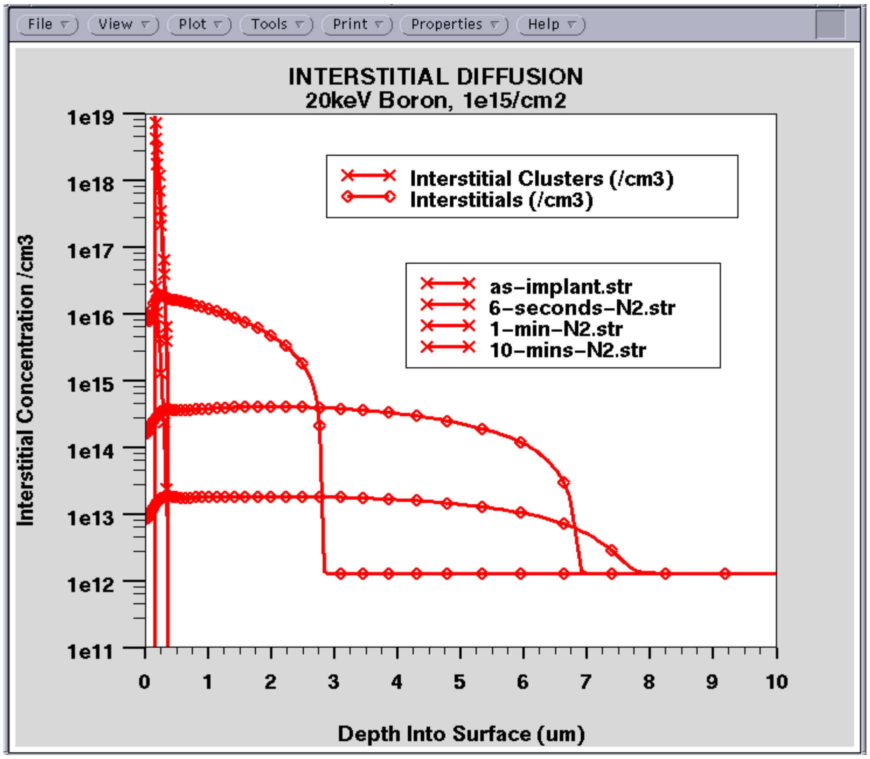


Figure 2-26 Interstitials can move far into the substrate even after a short 10min anneal

Figure 2-26 shows typical diffusion profiles of interstitials after a 1e15/cm3 20keV Boron implant at various anneal times. After only a 10 minute anneal, the interstitials have diffused 8um into the substrate.

Interstitials, like dopant, require a concentration gradient in order for overall diffusion to take place. For example, if the concentration gradient of interstitials is removed by having too shallow a substrate depth, the concentration of interstitials will start to pile up because they are no longer being removed through diffusion into the bulk of the substrate. If the level of modeled interstitials becomes too high, the diffusion of dopant, even near the surface of the substrate, will also be too high and the simulation will be inaccurate.

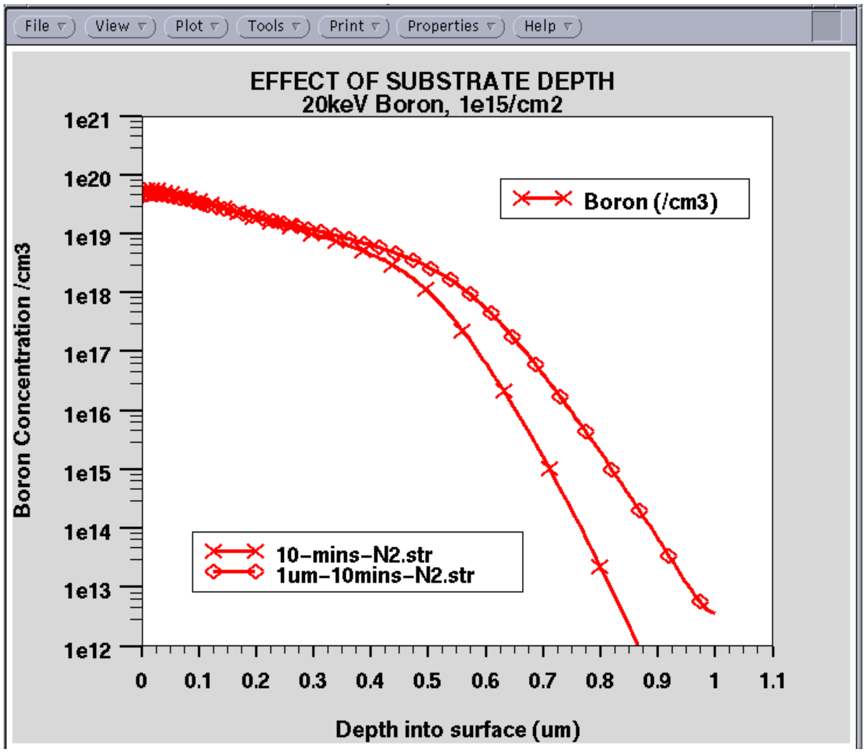


Figure 2-27 Effect on boron diffusion profile when too small a substrate depth is used in the simulation

Figure 2-27 shows the boron profiles for two identical anneals, the only difference is the depth of the simulated substrate. You'll see that a shallow modeled substrate always results in more total diffusion, even though the substrate depth was greater than the total diffusion depth in both cases.

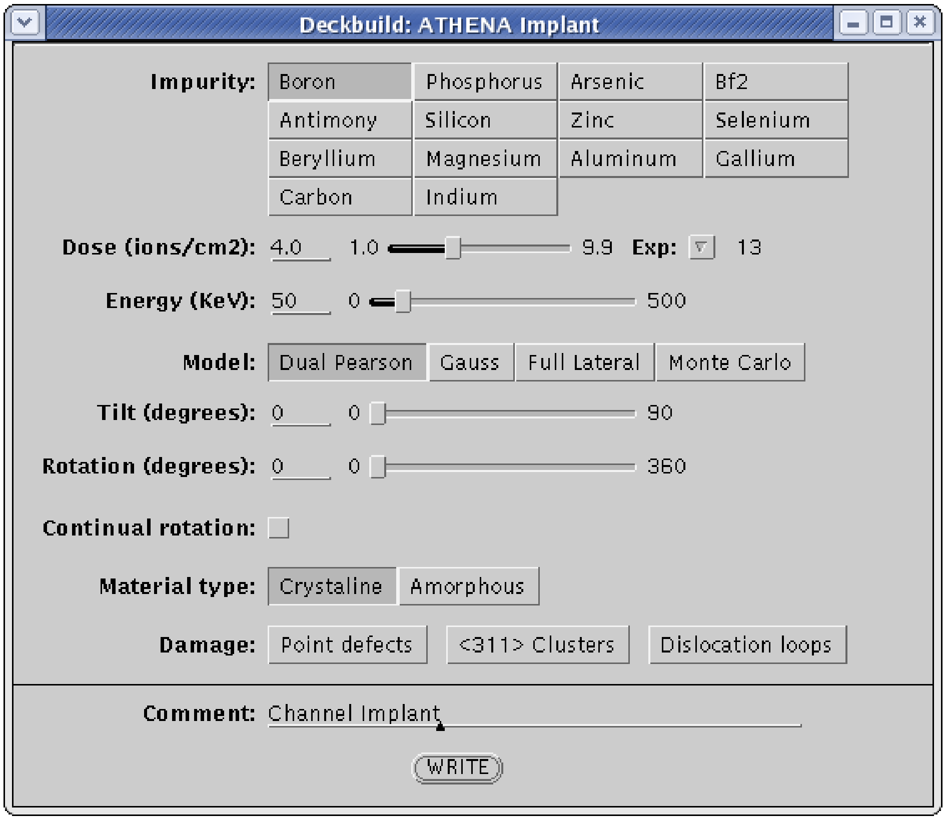
Modeling a deep substrate doesn't need to involve a huge number of extra mesh points, since the mesh points can be placed quite far apart near the bottom of the substrate. All that is required of the mesh points near the bottom of the substrate is that there be sufficient to model the gradient of interstitials in this region. The number of additional mesh points can be further reduced in the X-direction by the using several `RELAX` statements.

For normal small geometry MOSFET/Bipolar processing, a substrate depth of 20 µm should be more than adequate. This depth can be reduced by plotting the vertical interstitial profiles at various points in the process to find the maximum depth of interstitial diffusion. There is little to be gained by reducing the depth of simulation, however, if the combination of large grid spacing is deep in the substrate and the `RELAX` statement is used appropriately.

Simulating Ion Implantation

Ion implantation is the main method used to introduce doping impurities into semiconductor device structures. Adequate simulation of the ion implantation process is very important because modern technologies employ small critical dimensions (CDs) and shallow doping profiles, high doses, tilted implants and other advanced methods.

The `IMPLANT` statement can be set by using the Athena Implant Menu (Figure 2-28). To open this menu, select **Process →Implant...** in the **Commands** menu.



Deckbuild: ATHENA Implant

Dose (ions/cm2): 4.0 Exp: 13

Energy (KeV): 50

Tilt (degrees): 0

Rotation (degrees): 0

Continual rotation: □

Comment: Channel Implant

Figure 2-28 Athena Implant Menu

The following gives the minimum set of parameters that should be specified:

- Name of implant impurity (e.g., boron)
- Implant dose using the slider for the pre-exponential value (e.g., 4.0) and the **Exp** menu for the exponent (e.g., 12)
- Implant energy in KeV (e.g., 60)
- Tilt angle in degrees (e.g., 7°)
- Rotation angle in degrees (e.g., 30°)

All other parameters can use their default values.

Press the **Write** button and the following statement will appear in the input file.

```
# CHANNEL IMPLANT
IMPLANT BORON DOSE=4.0E12 ENERGY=60 PEARSON TILT=7 ROTATION=30 \
CRYSTAL
```

All of the parameters in the statement above are self-explanatory except CRYSTALLINE. The CRYSTALLINE parameter indicates that for all analytical models, the range statistics extracted for a single silicon crystal will be applied (when available). If AMORPHOUS is selected, the range parameters measured in pre-amorphized silicon will be used (when available). The CRYSTALLINE parameter also has another meaning for the Monte Carlo or BCA implant models. It invokes the Crystalline Material Model which takes channeling into account. Note that the latter model is much slower (5 - 10 times) than the Amorphous Material Model. The Crystalline Material Model is the default model for BCA or Monte Carlo simulation.

For a detailed description of ion implant model selection, see Section 3.5 “Ion Implantation Models”.

You can specify tilt and rotation angles of the ion beam. Positive tilt angles correspond to the ion beam coming from the top left. Specifying the rotation angle makes sense only for non-zero tilt angles. Zero rotation means that the ion beam vector lies in the plane parallel to the 2D simulation plane. 90° rotation means that the ion beam vector lies in the plane perpendicular to the simulation plane.

Selecting **Continual rotation** causes SSuprem4 to rotate the wafer, i.e., implantation will be performed at 24 different rotation angles from 0 to 345°, in increments of 15°.

There are several damage models available in SSuprem4. These models allow you to estimate distributions of various defects generated after ion implantation. For more details about the damage models and their effect on subsequent diffusion, see Section 3.5.5 “Ion Implantation Damage”.

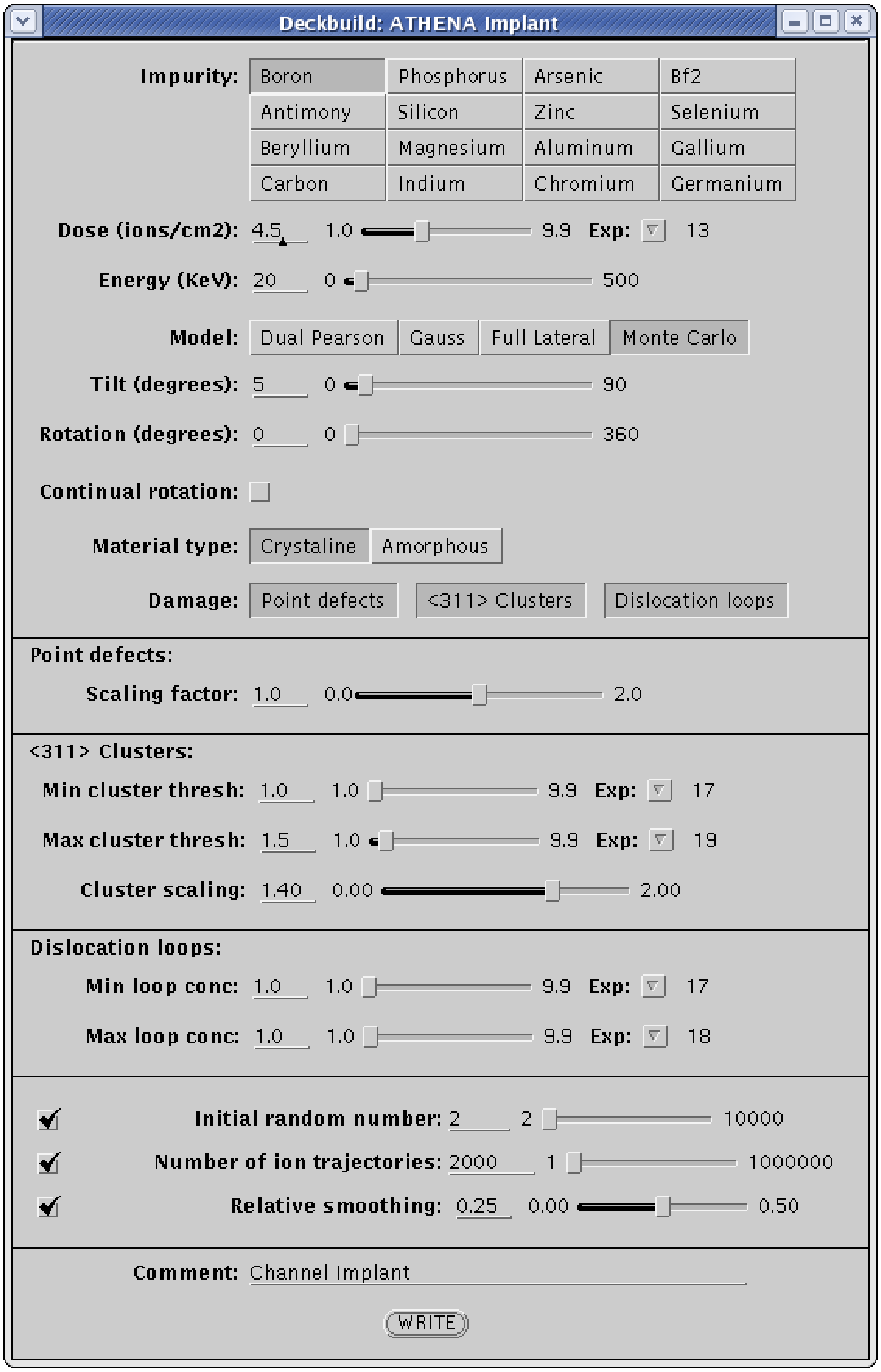
When the Monte Carlo model is selected, you can specify several additional optional parameters (See Figure 2-29). The first three parameters are related to the Damage model (Point defects, {311}-clusters, and dislocation loops). The three others control Monte Carlo calculation (initial random number, number of trajectories, and smoothing). See Table 2-2 for a quick reference of Athena implant models.

Table 2-2 Athena Implant Model Reference

Process	Model	Assumption	Recommendation
Implant	SIMS Verified Dual Pearson (SVDP) - Default	Empirical	See Table 3-7.
	Single Pearson	Analytic	All other cases
	Monte Carlo Monte or BCA	Statistical	Multi-layer structures: angled implants into a structure where many ions could be reflected (trenches), when channeling is not described by SVDP, high or very low energy.

Table 2-2 Athena Implant Model Reference

Process	Model	Assumption	Recommendation
Silicon Type	Amorphous	No channeling effect is included	Most of implant profile is within amorphous materials (oxide, polysilicon, pre-amorphized silicon); channeling is negligible or not important.
	Crystal - Default	Channeling effect is included	When channeling effects are important: light ions (boron, phosphorus), zero or close to 0° tilt, implant through thin amorphous layer into crystalline substrate.



Deckbuild: ATHENA Implant

Impurity: Boron | Phosphorus | Arsenic | Bf2 | Antimony | Silicon | Zinc | Selenium | Beryllium | Magnesium | Aluminum | Gallium | Carbon | Indium | Chromium | Germanium

Dose (ions/cm2): 4.5 1.0 9.9 Exp: 13

Energy (KeV): 20 0 500

Model: Dual Pearson | Gauss | Full Lateral | Monte Carlo

Tilt (degrees): 5 0 90

Rotation (degrees): 0 0 360

Continual rotation: □

Material type: Crystaline | Amorphous

Damage: Point defects | <311> Clusters | Dislocation loops

Point defects:

Scaling factor: 1.0 0.0 2.0

<311> Clusters:

Min cluster thresh: 1.0 1.0 9.9 Exp: 17

Max cluster thresh: 1.5 1.0 9.9 Exp: 19

Cluster scaling: 1.40 0.00 2.00

Dislocation loops:

Min loop conc: 1.0 1.0 9.9 Exp: 17

Max loop conc: 1.0 1.0 9.9 Exp: 18

☑ Initial random number: 2 2 10000

☑ Number of ion trajectories: 2000 1 1000000

☑ Relative smoothing: 0.25 0.00 0.50

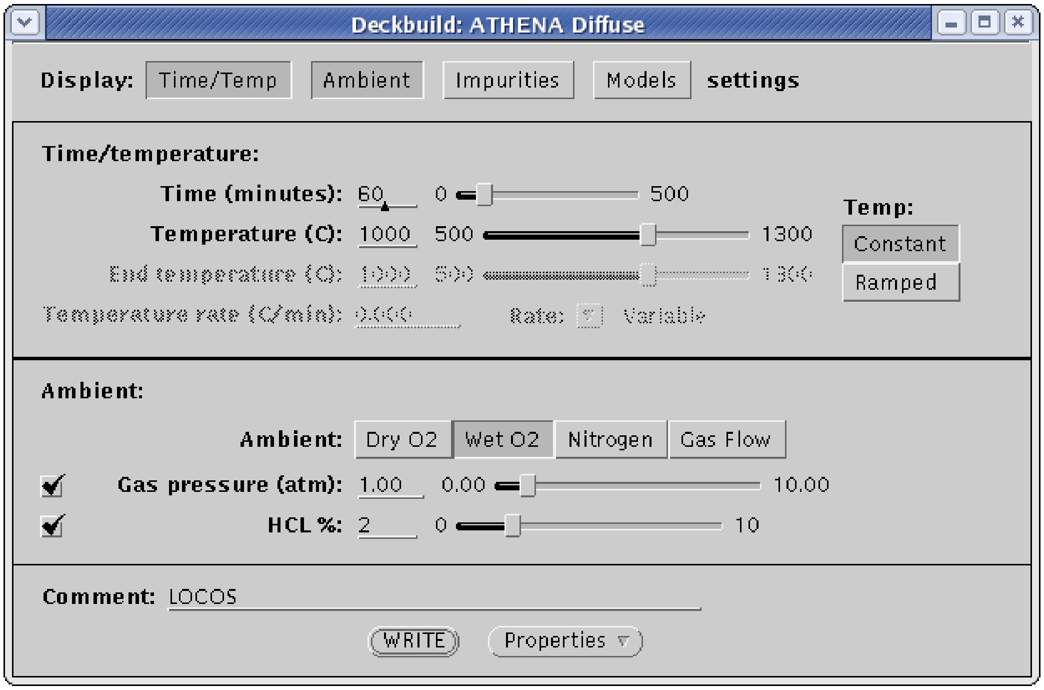
Comment: Channel Implant

Figure 2-29 Athena Implant Window

Simulating Diffusion

Simulation of thermal process steps is a focal point of SSuprem4. The hierarchy of diffusion and oxidation models is described in this chapter and in Sections 3.1 “Diffusion Models” and 3.3 “Oxidation Models”. This section will demonstrate how to set different parameters and models of diffusion, oxidation and silicidation. The last process will take place only if at least one refractory metal or silicide layer is present in the structure.

The parameters and models of a diffusion/oxidation step can be prepared from the Athena Diffuse Menu. (Figure 2-30).


Deckbuild: ATHENA Diffuse
Display: Time/Temp Ambient Impurities Models settings
Time/temperature:
Temp:
Constant
Ramped
Ambient:
Dry O2 Wet O2 Nitrogen Gas Flow
WRITE
Properties

Time (minutes): 60 0 500

Temperature (C): 1000 500 1300

End temperature (C): 1000 500 1300

Temperature rate (C/min): 0.000 Rate: Variable

Ambient: Wet O2

☑ Gas pressure (atm): 1.00 0.00 10.00

☑ HCL %: 2 0 10

Comment: LOCOS

Figure 2-30 Athena Diffuse Menu

To open this menu, select **Process→Diffuse...** in the DeckBuild **Commands** menu. The **Diffuse** menu has four sections. Only the **Time/Temperature** and **Ambient** fields appear initially. The **Impurities** and **Models** fields appear only when the corresponding check boxes are selected.

The minimum set of diffusion step parameters is as follows:

- Time (e.g., 60 minutes)
- Temperature (e.g., 1100° Celsius)
- Gas pressure (1 atmosphere is default)

The following input file statements will appear:

```
# DRIVE-IN
DIFFUSE TIME=60 TEMP=1100 NITRO PRESS=1.00
```

If you choose the **Ramped** box and **End Temperature** or **Temperature rate**, a ramped temperature thermal step is simulated. The temperature rate is a variable by default, but it can be set to a specific constant temperature rate by selecting **Constant** in the **Rate** box. If the **End Temperature** is set to **1000,** the following lines appear:

```
# RAMPING DOWN

DIFFUSE TIME=60 TEMP=1100 T.FINAL=1000 NITRO PRESS=1.00
```

The same pull down menu used for inert diffusions is also used for oxidations described in the “Simulating Oxidation” on 63. But, since there are special considerations for inert diffusions which come under the category of Rapid Thermal Anneals (RTA), the special notes pertaining to this specific set of conditions are described in the next section. These notes are very important for accurate simulation of high temperature, short duration anneals. We recommend that you read these notes before attempting to write the RTA section of the input file.

2.4.6 Simulating Rapid Thermal Anneals (RTA) Notes

The usual reason for employing a Rapid Thermal Anneal (RTA) in a process flow is to anneal out damage in the substrate that has been caused by a previous process step, usually an implant, while at the same time minimizing dopant diffusion. Dopant activation also occurs during this process. These anneals are usually high in temperature and low in duration for sound device physics reasons.

Once again, the key to accurate simulation of RTA lies in the accurate simulation of substrate damage behavior. The role of interstitials in enhanced dopant diffusion has already been explained in Section 2.4 “Choosing Models In SSuprem4” to become familiar with the role of interstitials during process simulation.

The reason why an RTA usually employs high temperatures and short durations is because for a given high dose implant, if an anneal duration is selected so that a fixed percentage of the damage is annealed, the lower the anneal temperature, the more dopant diffusion occurs.

The above statement requires an explanation since intuitively, the opposite would seem more likely. A descriptive explanation of what is happening can be informative if the two extremes of anneal temperature are considered.

For the lowest anneal temperatures, the damage anneal rate is almost zero, so dopant diffusion rates are enhanced by a factor of 1000°C or more for the long time periods required to remove the damage. This results in high total dopant diffusion.

For the highest temperature anneals, a significant percentage of damage removal occurs in a fraction of a second. Almost zero damage enhanced diffusion or total diffusion occurs in this instance, and the anneal time to remove the damage is very short. Extrapolating between these extremes provides a qualitative explanation of what occurs for intermediate temperature anneals.

Two important points have now been established:

1. For sound device physics reasons, most RTA processes consist of high temperature, short duration anneals.
2. Damage-enhanced diffusion will only occur for a few seconds at typical RTA temperatures.

For accurate simulation of RTA, the second point is most important and often wrongly neglected. Suppose an RTA consists of a 10 second ramp up to 1000°C, followed by a 20 second anneal and a 10 second cool down. From the second point, it is apparent that most of the Total Dopant Diffusion would have taken place during the Ramp Up Phase of the RTA.

Therefore, always model the temperature ramp up accurately when simulating an RTA process. In most cases, the ramp down can be neglected, since all the diffusion has already taken place at the beginning when the silicon was still damaged.

2.4.7 Simulating Oxidation

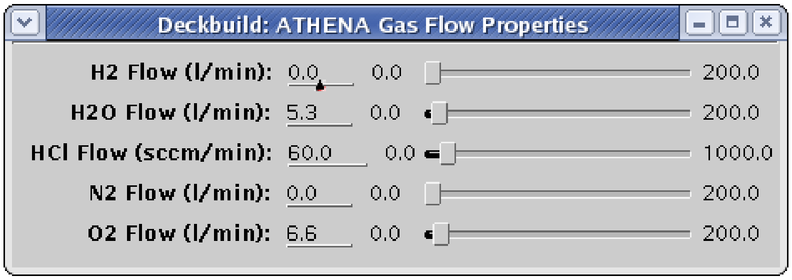
It has already been stated that the pull down menu for simulating oxidations is the same as that for simulating inert diffusions described in the “Simulating Diffusion” on 61. See this section for advice on selecting the appropriate pull down menu from DeckBuild.

The default method for oxidation is **Compress**. In SSuprem4 examples there are a number of examples which illustrate the use of different models for different processes and structures.

In our previous example described in the “Simulating Diffusion” on 61, if the next temperature step is going to be at a constant temperature of 1000°C in dry O2 with 3% of HCL in the ambient, select the Dry O2 box and set HCL% equal to 3 in the **Ambient** section of the **Diffuse** menu. The following input file fragment will appear:

```
# GATE OXIDE
DIFFUSE TIME=60 TEMP=1000 DRYO2 PRESS=1.00 HCL.PC=3
```

If the ambient is a mixture consisting of more than one oxidant, the total oxidation rate will depend on the combined effect of all species in the ambient. To specify the contents of the ambient mixture, select the **Gas Flow** button in the **Ambient** section and an additional Athena Gas Flow Properties Menu (Figure 2-31) will appear.



Deckbuild: ATHENA Gas Flow Properties

H2 Flow (l/min): 0.0 0.0 200.0

H2O Flow (l/min): 5.3 0.0 200.0

HCl Flow (sccm/min): 60.0 0.0 1000.0

N2 Flow (l/min): 0.0 0.0 200.0

O2 Flow (l/min): 6.6 0.0 200.0

Figure 2-31 Athena Gas Flow Properties menu

If the Gas Flow components are selected, as shown in Figure 2-31, the following statement will be generated:

```
# GATE OXIDE
DIFFUSE TIME=60 TEMP=1000 F.H2O=5.3 F.HCL=0.06 F.O2=8.0 \
PRESS=1.00
```

One or several impurities can be present in the ambient. To set ambient in the **Impurity Concentration** section of the Athena Diffuse Menu (See Figure 2-30), check the corresponding checkboxes, and set the values using sliders and the **Exp** menus.

For example, by selecting the appropriate boxes and values, the following `DIFFUSE` statement will be inserted into the input file:

```
# FIELD OXIDE
DIFFUSE TIME=100 TEMP=850 T.FINAL=1060 WETO2 PRESS=1.00 \
HCL.PC=0 C.ARSENIC=9.0E19 C.PHOSPHOR=4.0E20
```

Several other parameters not included on the menu are available in the `DIFFUSE` statement (Section 6.15 “DIFFUSE”). The `DUMP` and `DUMP.PREFIX` parameters can be useful. `DUMP` and DUMP.PREFIX can be used to make a movie using TonyPlot. This provides a good approximation for low temperature processes, such as silicidation.

Several other model specification statements are important for diffusion processes. These are as follows:

- IMPURITY, INTERSTITIAL, and other impurity and point defect statements, which specify model parameters (e.g., diffusivity or segregation) of these species.
- The OXIDE statement, which specifies parameters for different oxidation models.
- The MATERIAL statement, which specifies some basic parameters for all materials.
- The SILICIDE statement, which specifies silicidation coefficients.

Table 2-3 shows the basic diffusion and oxidation models.

Table 2-3 Basic Diffusion and Oxidation Models

Process	Model	Assumption	Recommendation
Diffuse	Fermi- Default	Defect in equilibrium	For undamaged substrates in inert ambients
	two.dim	Transient defect diffusion	During oxidation, and after medium dose implant (e.g., OED)
	full.cpl	Defect and impurity binding energy model	Post high dose implant & co-diffusion effects, but execution time is high
Oxidation	Vertical	Planar	1D oxidation only (should never be used)
	Compress- Default	Non-planar with linear flow	2D oxidation (e.g. birds beak)
	Viscous Elastic	Non-planar with non-linear flow	2D oxidation (e.g. birds beak with thick Si_3N_4, however, execution time is higher.

For a detailed description of all diffusion and oxidation models, see Sections 3.1 “Diffusion Models” and 3.3 “Oxidation Models”.

2.4.8 Simulating the Epitaxy Process

Athena/SSuprem4 can simulate a high temperature silicon epitaxial processes. The epitaxy process is considered as a combination of deposit and diffuse processes. Therefore, processes such as **autodoping** from a highly doped buried layer into a lightly doped epitaxial layer can be simulated. Diffusion parameters for epitaxial silicon, however, are considered the same as for single crystal silicon.

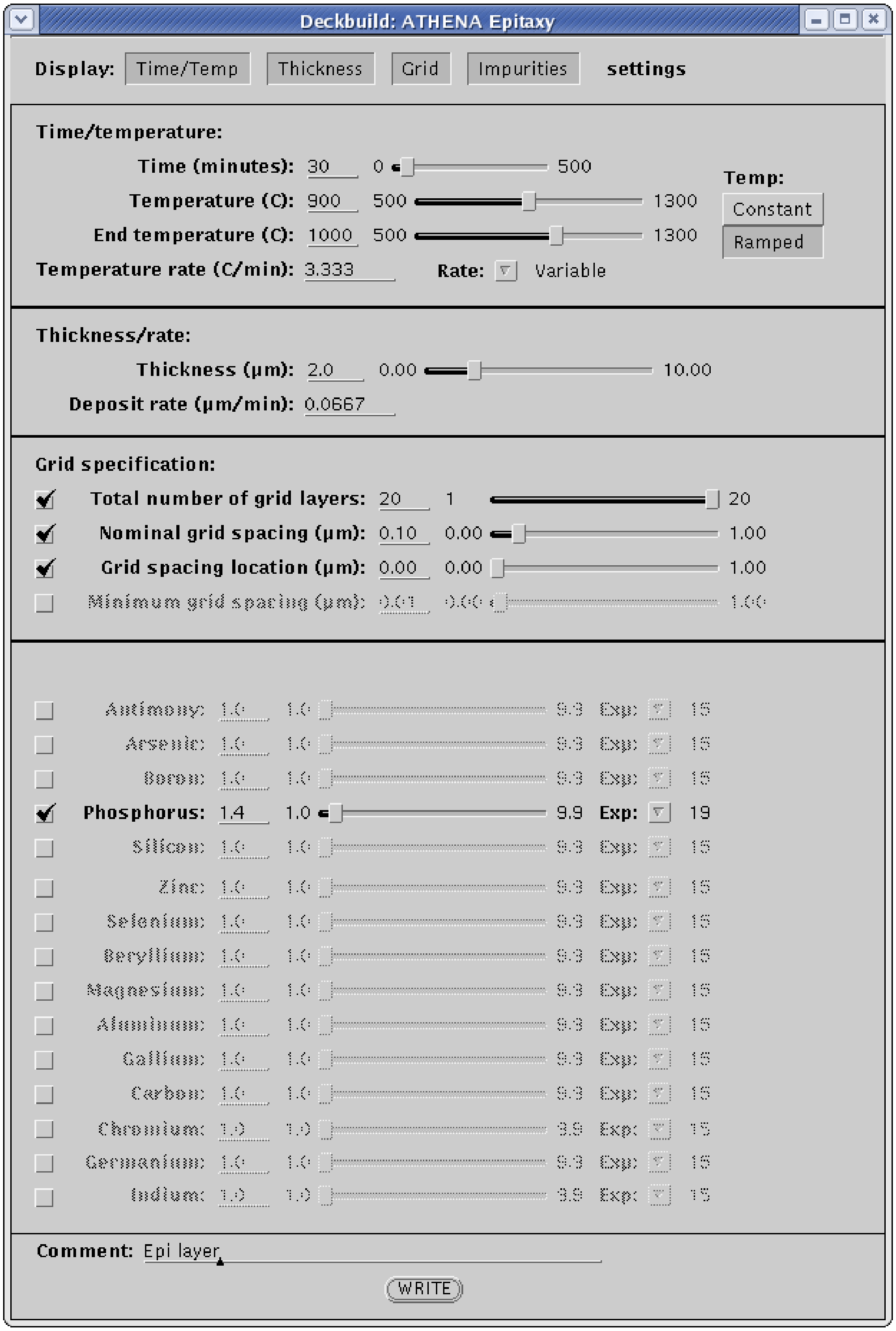
The epitaxy process is defined in the Athena Epitaxy Menu (Figure 2-32). To open this menu, select **Process→Epitaxy** in the **Commands** menu. The Athena Epitaxy Menu consists of four sections:

- The **Time/temperature** section selects temperature step parameters in the same way as in the `DIFFUSE` statement.
- The **Thickness/rate** section selects either the total thickness of the epitaxial layer, or the deposit rate in microns/minute. In the latter case, the total thickness will be determined by the rate and time.
- The **Grid Specification** section specifies the vertical grid structure within the grown epitaxial layer. All grid parameters are equivalent to those of the Athena Deposit Menu (See Figure 2-12).
- The **Impurity Concentrations** section specifies the growing epitaxial layer in the same way as in the `DIFFUSE` statement.

All parameters in the last two groups are optional. If the parameters of an epitaxial step are set exactly as shown in Figure 2-32, the following statement will appear in the input file:

```
# EPI-LAYER

EPITAXY TIME=30 TEMP=900 T.FINAL=1000 THICKNESS=2.0 DIVISIONS=20
\

DY=0.1 YDY=0.00 C.PHOSPHOR=1.4E19
```

Note: The diffusion during the epitaxy process will use the Diffusion Model Set in the most recent `METHOD` statement. If you need another `METHOD` statement, include it before the `EPITAXY` statement.


Deckbuild: ATHENA Epitaxy
Display: Time/Temp Thickness Grid Impurities settings
WRITE

Time/temperature:

Time (minutes): 30 — 0 to 500

Temperature (C): 900 — 500 to 1300

End temperature (C): 1000 — 500 to 1300

Temperature rate (C/min): 3.333 Rate: Variable

Temp: Constant [✓ Ramped]

Thickness/rate:

Thickness (μm): 2.0 — 0.00 to 10.00

Deposit rate (μm/min): 0.0667

Grid specification:

☑ Total number of grid layers: 20 — 1 to 20

☑ Nominal grid spacing (μm): 0.10 — 0.00 to 1.00

☑ Grid spacing location (μm): 0.00 — 0.00 to 1.00

□ Minimum grid spacing (μm): 0.01 — 0.00 to 1.00

□ Antimony: 1.0 — 1.0 to 9.9 Exp: 15

□ Arsenic: 1.0 — 1.0 to 9.9 Exp: 15

□ Boron: 1.0 — 1.0 to 9.9 Exp: 15

☑ Phosphorus: 1.4 — 1.0 to 9.9 Exp: 19

□ Silicon: 1.0 — 1.0 to 9.9 Exp: 15

□ Zinc: 1.0 — 1.0 to 9.9 Exp: 15

□ Selenium: 1.0 — 1.0 to 9.9 Exp: 15

□ Beryllium: 1.0 — 1.0 to 9.9 Exp: 15

□ Magnesium: 1.0 — 1.0 to 9.9 Exp: 15

□ Aluminum: 1.0 — 1.0 to 9.9 Exp: 15

□ Gallium: 1.0 — 1.0 to 9.9 Exp: 15

□ Carbon: 1.0 — 1.0 to 9.9 Exp: 15

□ Chromium: 1.0 — 1.0 to 9.9 Exp: 15

□ Germanium: 1.0 — 1.0 to 9.9 Exp: 15

□ Indium: 1.0 — 1.0 to 9.9 Exp: 15

Comment: Epi layer

Figure 2-32 Athena Epitaxy Menu

2.5 Calibrating Athena for a Typical MOSFET Flow

This section of the manual provides information on which parameters should be tuned in the input file to provide predictive simulations using a typical MOSFET process flow. We assume you are now familiar with the mechanics of making an input file and using the correct methods and models (see Section 2.4 "Choosing Models In SSuprem4"). For example, incorrect use of the `METHOD` statement will invalidate the rest of the following section.

Calibrating an Athena input file for a typical MOS process flow involves using the device simulator, Atlas, since electrical measurements from the MOSFETs in question often represents the majority of the physical data available for calibration. This can be thought as a paradox since Atlas would also have to be correctly calibrated. The reason that this doesn't present a problem is discussed below.

An important point to remember when using Technology Computer Aided Design (TCAD) is that the most critical task is to accurately model the process flow.

Note: For accurate MOSFET simulation, you should invest 90% of the time in achieving an accurate process simulation, while only investing 10% of the time in fine-tuning the device simulation.

The reason for this, especially for silicon technologies, is that the device physics, in general, is understood. For silicon, not only is the physics well understood, it is also well characterized, so most of the default values in Atlas will be correct. Therefore, the calibration of an Athena process file does not involve the calibration of well known quantities such as diffusion coefficients. Instead, the calibration involves variables that are process and production line dependent. For example, the damage caused by an implant cannot be determined exactly, since it is dose rate dependent and can be influenced by beam heating of the substrate, which is dependent on the carousel rotation speed and the efficiency of the cooling system.

Note: If the process has been correctly modeled, the device simulation will also be accurate if appropriate models have been chosen.

If a simulated device exhibits electrical characteristics that are totally inaccurate, you may have done something wrong in the process simulation. Do not make the mistake of changing well known default values in the simulators to make a curve fit one set of results because this will lead to poor predictive behavior. Try and find the cause of a discrepancy.

2.5.1 Input Information

It may seem obvious but must be emphasized that an accurate process flow is vital for simulation accuracy, especially for Rapid Thermal Anneals (see Section 2.4.6 “Simulating Rapid Thermal Anneals (RTA) Notes” for details). Other process information required is an accurate cross-section of the oxide spacer. Modeling the spacer profile accurately ensures the lateral damage distribution due to the subsequent source-drain implants is correctly modeled.

Turning to electrical data, the most important device electrical data is a plot of threshold voltage versus gate length for the NMOS devices. Figure 2-33 shows typical plots of threshold voltage versus gate length. In this figure, the RTA anneal temperature and times were varied to show the various profiles that can be expected. A more typical plot is represented by the 1000°C RTA profile, showing a peak value around 1-2 microns with a tail off for longer or shorter gate lengths.

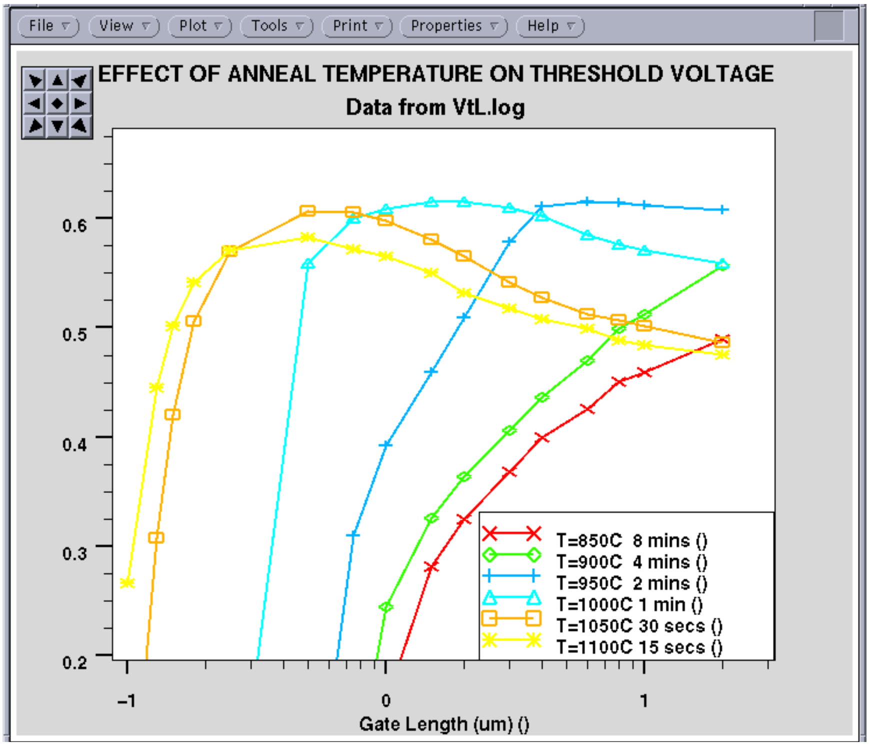


Figure 2-33 A plot of Threshold Voltage vs. Gate Length for NMOS devices

Gate oxide thickness measurements are also required. Be careful here if oxide thickness is measured with capacitance-voltage (C-V) methods, since quantum effects in very thin oxides (less than 5nm) can lead to inaccuracies because the actual location of the peak concentration of the accumulation charge is not at the interface as classic physics predicts but a short distance into the silicon. Use the QUANTUM model in Atlas to match accumulation capacitance with oxide thickness for very thin oxides.

Other useful electrical input information is data that won’t be used now but later for the calibration process itself, testing the predictive nature of the simulation. Typical device characteristics used for predictive testing includes threshold voltage versus gate length measurements for a non-zero substrate bias.

2.5.2 Tuning Oxidation Parameters

During oxidation, interstitials are injected into the silicon substrate by the advancing interface. The first parameter to tune is the fraction of consumed silicon atoms that are re-injected back into the substrate as interstitials. In Athena, the related tuning parameter is called THETA.0 and is defined in the INTERSTITIAL statement. THETA.0 has been found to be slightly different for wet and dry oxides. The default value is reasonably accurate for dry oxides but some tuning may be required for wet oxidation.

The major effect of interstitial injection during gate oxidation is to create enhanced diffusion of the threshold adjust implant. The measured threshold voltage of the final device is very sensitive to the dopant concentration near the silicon-gate oxide interface. Consequently, threshold voltage measurements are a sensitive indicator of interstitial behavior. Oxidation, however, is not the only source of interstitial injection. The source-drain and LDD implants also induce a large concentration of interstitials. In order to isolate oxidation enhanced diffusion, the threshold voltage of a long gate length device is used, preferably where L=20 µm or more, so that the threshold voltage will be little influenced by damage near the source-drain regions.

Interstitials injected by source-drain implant damage can travel up to 10 µm along the surface before recombination takes place. A gate length of 20 µm is recommended as the minimum gate length for calibration so this can allow the interstitials to diffuse 10 µm along the surface from both the source and drain ends without effecting diffusion near the center of the device. In summary, tuning THETA.0 involves the comparison of modeled and measured threshold voltage data for a long gate-length device.

THETA.0 can be rapidly tuned by taking a one dimensional (1D) vertical cutline through the center of the gate and doing a 1D process simulation. You can either tune THETA.0 manually or by using the **Optimize** function in DeckBuild. Theta.0 is tuned until the measured and simulated data of the long channel threshold voltage correspond. The fine tuning of THETA.0 is performed by using a full 2D simulation.

Figure 2-34 shows a typical dependence of extracted threshold voltage on the THETA.0 tuning parameter. Realistic values of THETA.0 correspond to the rising part of the curve. The glitch in the curve is due to rounding errors in the EXTRACT statement used to calculate the threshold voltage due to the automatic and independent mesh generated in the EXTRACT statement. The mesh can be changed from its default value shown here to eliminate this effect. But close examination reveals that the error is only a few millivolts off, which is accurate enough for most process parameter extractions.

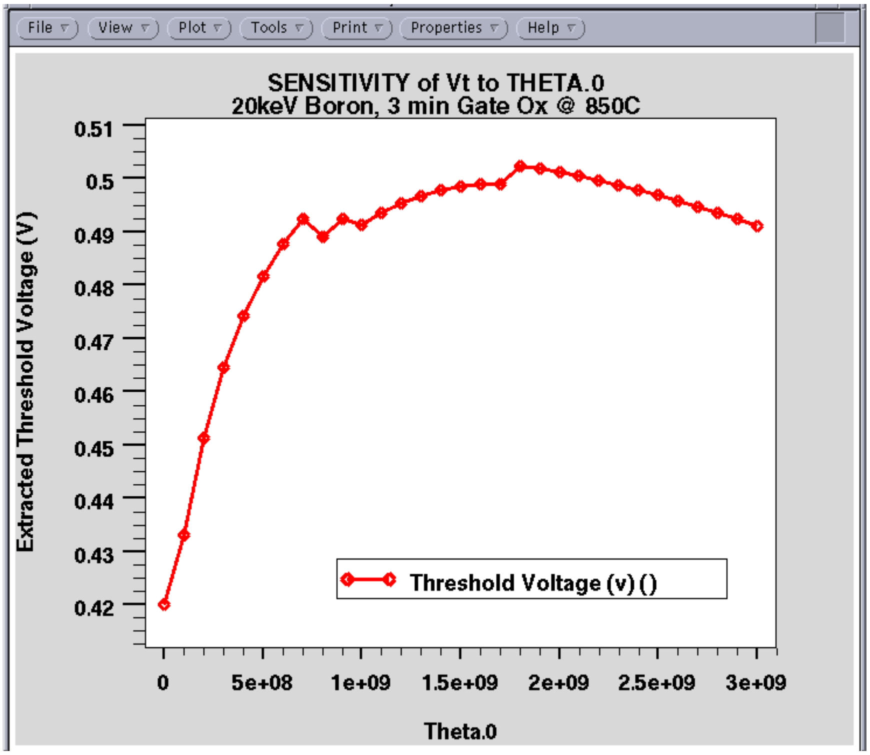


Figure 2-34 A Typical Dependence of Extracted Threshold Voltage

2.5.3 Tuning Implantation Parameters

You can now tune two implantation parameters by using the threshold voltage versus gate length data. The peak value of threshold voltage for a given process flow (the reverse short channel effect) will be a function of the initial implant damage caused by the LDD and source-drain implants. Since these implants have a high total dose and damage, the tuning parameter here is the clustering factor. In Athena, this parameter is called CLUST.FACT and is defined in the CLUSTER statement. The higher the clustering factor, the greater the damage, and the greater the diffusion, the greater the reverse short channel effect.

Figure 2-35 shows the effect on the threshold voltage of changing the CLUST.FACT parameter for a typical process flow.

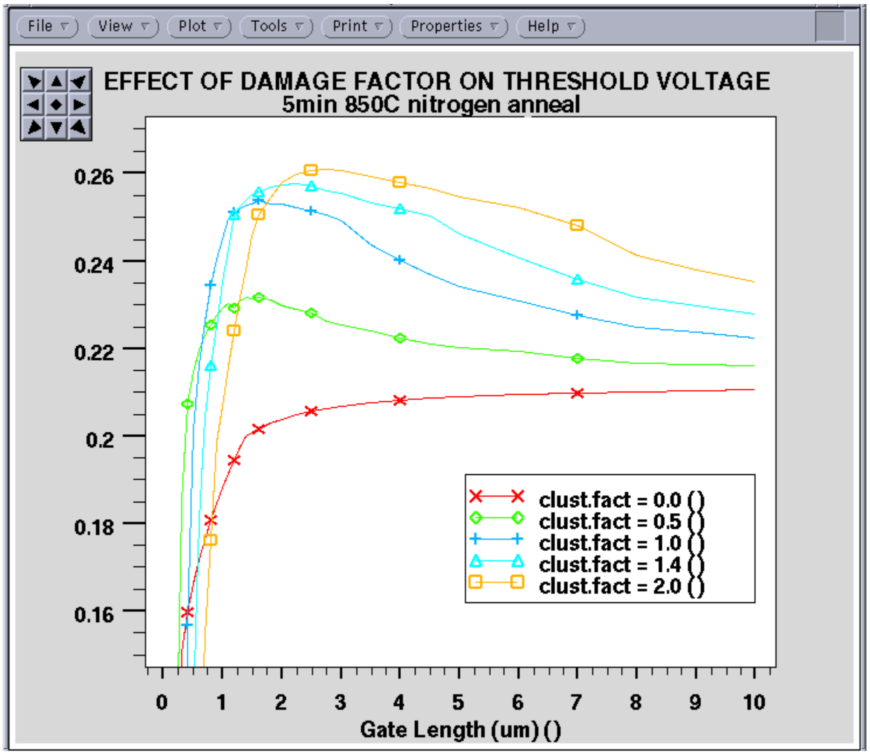


Figure 2-35 How Changing the clust.fact parameter affects the threshold voltage

The second implantation parameter that can now be tuned is the lateral spread of the implant near the surface. In Athena, this parameter is called LAT.RATIO1 and is defined in the IMPLANT statement. The lateral spread of the source-drain and LDD dopant is responsible for the classical short channel effect, where the threshold voltage reduces for very short channel lengths. Simply tune the LAT.RATIO1 parameter until the onset of classical short channel effects of simulated and measured data correspond. If the LAT.RATIO1 is increased, the onset of the classical short channel effect will occur for longer gate lengths.

2.5.4 Tuning Diffusion Parameters

The final part of the threshold voltage versus gate length curve can now be used to tune the surface recombination rate of interstitials. In Athena, this parameter is called KSURF.0 and is specified in the INTERSTITIAL statement. The surface recombination of interstitials will dictate the roll-off rate of threshold voltage from its peak value (reverse short channel effect) to the long gate length value. Once again, simply tune KSURF.0 until the long channel threshold voltage roll off rate matches that of the measured data.

PMOS Tuning

PMOS devices are a special case since the boron doped Source/Drain implants overall tend to absorb interstitials rather than emit them. The reverse short channel effect in buried channel PMOS devices can be caused by high angle implants. If high angle implants are used, the reverse short channel effect can be tuned using the LAT.RATIO1 parameter in the IMPLANT statement.

2.5.5 Related Issues on using the Device Simulator Atlas for MOS Process Tuning

It should now be known that calibrating an Athena process file involves using the device simulator Atlas to a significant extent. Hence, it’s imperative that the use of the device simulator doesn’t create additional errors, rendering the process calibration results invalid.

Fortunately, the device physics involved in simulating the conditions required to extract a threshold voltage are not demanding. The drain voltage required to extract a threshold voltage is only 50-100mV so effects such as impact ionization can be neglected. The field perpendicular to the gate is also relatively low around the threshold voltage so field effects in this direction will do little effect. We recommend, however, using at least the models SRH and CVT during the calculation. Other parameters for silicon are sufficiently well known for silicon to the point that the results from the device simulator are reliable.

The first important point is to ensure that you let the device simulator calculate the work function of the gate electrode from the simulated doping profile rather than assigning a value to it. This means, making sure that the polysilicon gate is not itself defined as an electrode but rather a layer of metal, usually aluminum, is deposited on top of the polysilicon gate. Therefore, this metal layer is the film defined as the electrode. Do not assign a work function to this deposited metal electrode to ensure that it behaves as an ohmic contact rather than a Schottky contact. The effective work function of the poly gate will then be correctly calculated from the doping profile in the polysilicon.

An important area for accuracy in MOSFETs is modeling the inversion region under the gate. As it is, this charge that is responsible for current conduction in the device. The inversion region charge under the gate-only extends approximately 30 Angstroms into the silicon. The inversion region charge density under the gate falls off rapidly with depth into the silicon. It is imperative that there are several mesh points in the Y direction in this inversion region to model the drain current correctly. Accordingly, we recommend that the mesh spacing under the gate be no more than 10 Angstroms (1 nm).

You would think that a 10 Angstrom mesh under the gate would result in a huge number of mesh points. But, there only needs to be approximately three mesh points within the inversion region in the Y direction. The grid spacing can increase rapidly in spacing away from the oxide-silicon interface.

Figure 2-36 shows the effects of changing the mesh spacing at the interface on the simulated drain current. You can see from this figure that too coarse of a mesh always results in too high of a current simulated.

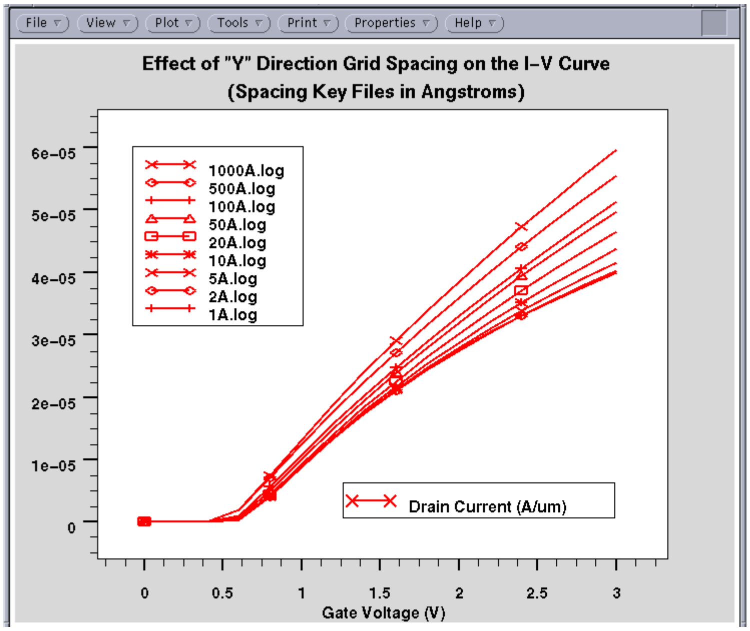


Figure 2-36 The effect of changing the mesh spacing at the interface on the simulated drain current

If contact resistance is a problem, then include it in the CONTACT statement. The resistance added to the CONTACT statement should be the measured resistance per contact divided by the number of contacts on each individual electrode. Obviously for D.C. measurements, the resistance on the gate contact will have no effect on the results since no current flows in this direction.

Checking the Predictive Powers of Tuned Process Parameters

If the process simulation has been correctly tuned, the process and device simulators should have predictive powers. To check the validity of the tuning process, use a new set of electrical data that was not used during the tuning process. For example, a good alternative set of data is to check the threshold voltage versus gate length for a non-zero voltage applied to the MOSFET body contact.

Conclusion

Using just one set of easily obtained measured electrical data, namely a plot of threshold voltage versus gate length, you can obtain most of the tuning parameters required for accurate process simulation. The other most important piece of data required is an accurate measurement of the gate oxide thickness, which is routinely measured in any instance.

You have been given specific advice as to which process and device models to use for each process in order to get the best results out of the simulation software. In particular, the correct use of models for the implantation and diffusion processes is stressed, as this has a dramatic effect on MOSFET characteristics, especially as anneal times and device dimensions decrease.

2.6 Calibrating Athena for a Typical Bipolar Process Flow

As with MOS calibration text, we assume you are familiar with the mechanics of making an input file and using the correct methods and models (see Section 2.4 “Choosing Models In SSuprem4”). For example, incorrect selection of diffusion models defined in the METHOD statement would invalidate the remainder of the following section.

Calibrating a bipolar process flow entails matching the two parameters, base current and collector current versus base emitter voltage to measure results throughout the full operating range of the device. By implication, the current gain of the device (*Ic/Ib*) will also be matched. All of the following paragraphs refer to the standard plot of collector and base currents measured against the base-emitter voltage, *Vbe*, unless it’s specifically stated otherwise. This standard IV graph is usually referred to as the Gummel Plot.

Another way of plotting the same information in a different format that can prove useful is a plot of current gain, *hfe*, versus the log of the collector current. This graph, however, is a derivation of the same information that makes it less clear as to which current is increasing or decreasing for each change. Therefore, a less useful graph when it comes to understanding exactly what is happening to the collector and base currents.

The full operating range of a bipolar junction transistor (BJT) consists of three general regions defined by the current density injected into the base. These three operating regions are usually described as low, medium, and high current injection regimes. The medium injection region is the most important part of the curve to model correctly as this represents the typical operating condition of the BJT. Each of the three operating regions is dominated by a different physical phenomenon. Therefore, successful modeling of a BJT involves matching both the base and collector currents in each of the three general operating regions, making a total of six areas for calibration. The derived parameter, *hfe*, is also a good parameter to monitor, since this is sensitive to errors in the ratio of collector to base current.

The following text suggests an approach and describes which of the six regions are effected by each change. The general technique is to calibrate the parameters that have the greatest effect on device performance in all regions first and then to move on to more subtle phenomenon that effect certain parts of the base or collector currents or both. In general, matching the collector current for all injection regions is less problematic than matching the base current at the extremes of the injection regions. Consequently, there are more sections on tailoring these parts of the curve. The text is divided into the following sections:

1. Tuning Base and Collector Currents – All Regions
2. Tuning the Base Current – All Regions
3. Tuning the Collector Current – All Regions
4. The Base Current Profile – Medium Injection
5. The Base Current Profile – Low Injection
6. Conclusions

If you follow this order, there should be a reasonable correlation between measured and simulated data. Most of the tuning parameters, however, have some degree of interdependency to the extent of which is also device design specific. Therefore, some degree of iteration of the tuning parameters is to be expected.

When tuning bipolar transistors, there is a greater emphasis to accessing tuning parameters by using the device simulator, Atlas, compared to optimizing MOSFETs, where most tuning parameters are process-related. A powerful combination is the tuning of a BiCMOS process where you can use the MOSFET part of the process flow to tune the process parameters, while using the Bipolar part of the flow to tune Atlas. This technique should yield a high degree of predictability in the results.

Tuning the process simulator parameters in Athena are mainly required to model effects, such as the implantation induced defect enhanced diffusion responsible for the Emitter Push Effect, which is essential to obtain the correct depth of the base-collector junction. The correct process modeling of the out diffusion of dopant from the poly-emitter into the mono-crystalline substrate is also critical to obtaining well-matched IV curves. Another critical process modeling area is the base implant, because it is essential to match measured and modeled base resistance for correct modeling of the collector current. These and other issues are discussed in these sections.

2.6.1 Tuning Base and Collector Currents – All Regions

The most important parameter to model the general level of base and collector currents is the device measurement temperature. The base and collector currents are strongly influenced by temperature changes, as small as a few degrees centigrade. A significant effort should be made to determine the exact temperature of the device during measurements before calibration is attempted. This temperature should be input into Atlas in the MODELS statement using the TEMPERATURE=<n> parameter. An increase in temperature will cause an increase in base and collector currents.

2.6.2 Tuning the Base Current – All Regions

A critical region for poly-emitter bipolar devices is the interface between the poly-emitter and the mono-crystalline silicon. This region is difficult to process simulate directly as the interface between the polysilicon emitter and single crystalline silicon usually consists of a thin, uneven and possibly non-continuous film of oxide. This is simulated by calibrating the overall effect of this interface with Atlas. The tuning parameter is the surface recombination velocity at this interface for electrons (VSURFN for PNP devices) or holes (VSURFP for NPN devices). This will only be effective for thin emitters where at least a fraction of the holes (for NPN devices) can reach the emitter before recombination.

The surface recombination velocity parameter not only affects the base current, it also affects the base current in all of the operating regions. Therefore, it is a powerful parameter to approximately match the base current and gain throughout the full operating range. In some cases, the base current may be less affected in the very high and very low injection regions by changes in the surface recombination velocity, and adding some scope to fine tuning the profile of the base current versus base-emitter voltage curve.

It is important to define the poly-emitter as an electrode so it can define the interfacial surface recombination velocity, VSURFN and VSURFP, using the CONTACT statement. This is in contrast to the MOSFET calibration text where we strongly advise you not to define the polygate as an electrode. Be sure not to get these two confused. The parameter that activates the recombination velocity is SURF.REC**,** which is also in the CONTACT statement. For example, an NPN BJT statement would be:

```
CONTACT NAME=emitter N.POLYSILICON SURF.REC VSURFP=1.5e5
```

A lower value of recombination velocity, `VSURFP`, will reduce the base current and increase the gain, *hfe*. The reverse is also true.

2.6.3 Tuning the Collector Current – All Regions

Figure 2-37 shows the parameter that affects the collector current over the entire range is the intrinsic base resistance. The base resistance is primarily determined by the dose of the base implant(s). An increase in the base implant dose will decrease the intrinsic base resistance and decrease the collector current in all injection regions. In some cases, however, the collector current may be affected a little in the very high injection region, giving scope for fine tuning the profile of collector current versus base-emitter voltage.

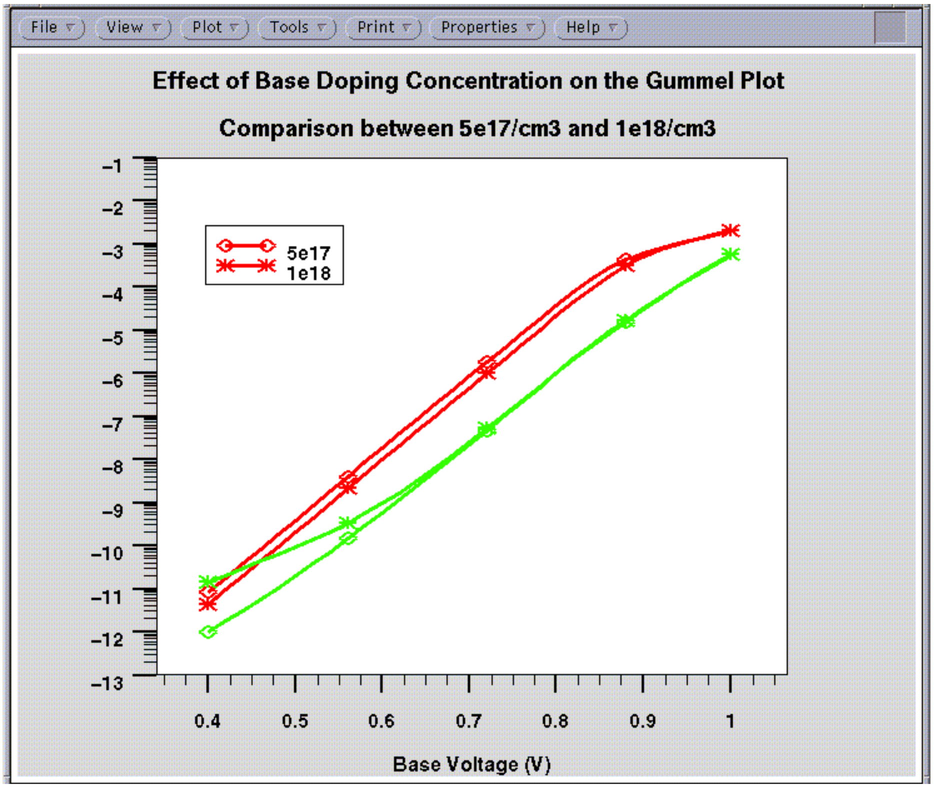


Figure 2-37 Effect of base doping profile on low injection base current in BJT

If the pinched or intrinsic base sheet resistance is a measured parameter, the simplest way to match measured and simulated data is to make slight changes to the base implant dose so that the simulated dose is not outside the expected error in actual implanted dose in conjunction with the error in percentage activation.

In some designs, where the base contact is close to the collector contact or the base contact is the substrate or is generally wide, the collector current can also influence all current injection regions by specifying a surface recombination velocity at the base contact. For a typical design with a buried n+ collector and surface contacts, the surface recombination velocity at the base contact may have little affect on the collector current.

2.6.4 The Base Current Profile – Medium Injection

In Atlas, there are two major parameters that have a significant affect on the base current in the medium injection regime. These parameters are the Poly-emitter Work Function and the Bandgap Narrowing Effect. These parameters are described below.

Poly-emitter work function

If the poly-emitter is described as `N.POLYSILICON` in the `CONTACT` statement for an NPN device, as already described, the Poly-emitter Work Function is then set to 4.17V and is correct for saturation doped n++ polysilicon. But if the poly-emitter is not saturation-doped, the work function will differ from this ideal and have a pronounced affect on the base current and current gain in the medium injection regime as shown in Figure 2-38. The work function of the poly-gate can vary from 4.17V for n++ poly-silicon to ($4.17V + E_g$) for p++ polysilicon, depending on the position of the Fermi-Energy. Changing the work function of the poly-emitter by just 0.1V from 4.17V to 4.27V can often reduce the current gain in half in the medium injection regime, so it's very important to assign the correct value. The `CONTACT` statement below assigns a work function of 4.27eV to the poly-emitter, while keeping the other parameters the same as before.

```
CONTACT NAME=emitter SURF.REC VSURFP=1.5e5 WORKFUN=4.27
```

The poly-emitter work function can be calculated by measuring the position of the Fermi-Energy at the poly-silicon/silicon interface relative to the conduction band and adding this value to 4.17V. For example, if the Fermi-Energy is measured as being 0.1eV from the conduction band edge, the work function of the poly-emitter set in the `CONTACT` statement should be set to 4.17 + 0.1 = 4.27V.

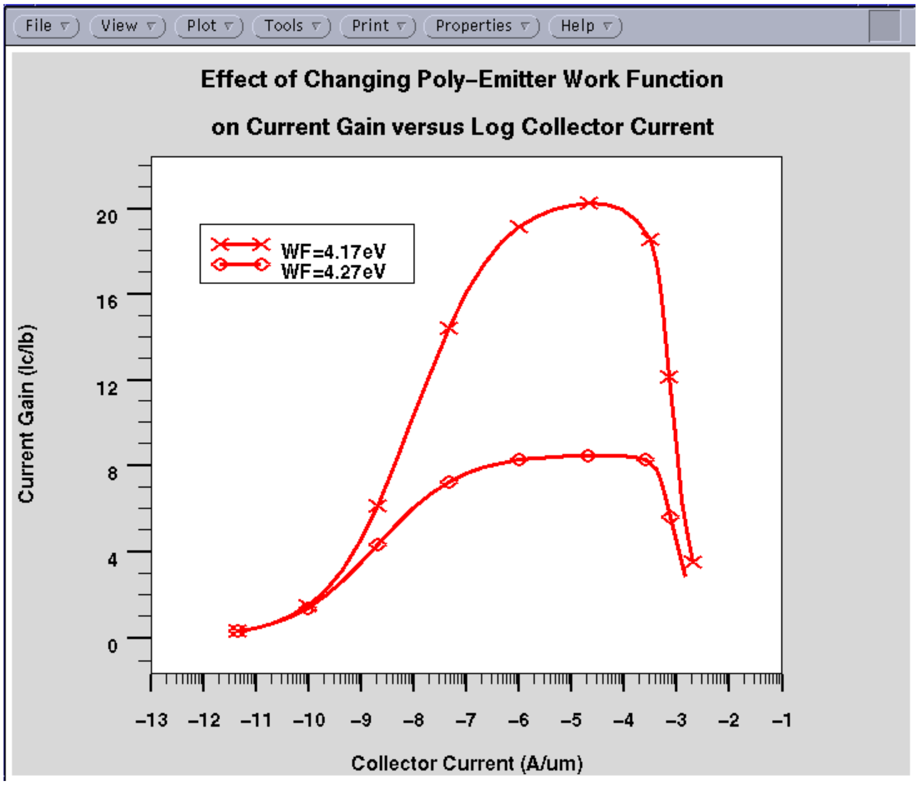


Figure 2-38 Effect of emitter contact work function on bipolar gain

Bandgap Narrowing Effects

If the BIPOLAR parameter is stipulated in the MODELS statement in Atlas, bandgap narrowing is included automatically. The inclusion of bandgap narrowing in the MODELS statement is strongly advised since this phenomenon has a significant effect on the current gain of the device. But, to validate the default Klaassen bandgap narrowing model, you should also use the Klaassen mobility model. Use the additional keyword KLA in the MODELS statement to activate this model. For example:

```
MODELS BIPOLAR KLA
```

The parameters in the Klaassen bandgap narrowing model are user-definable in the MATERIAL statement and described in the Physics chapter of the Atlas User's Manual. There are three user-definable parameters for the Klaassen band gap narrowing model. The BGN.E parameter has a linear dependency on doping concentration and has the default value of 6.92e-3 volts. BGN.C has a square root dependency with doping concentration and has the default value of 0.5. BGN.N is the value of doping where band gap narrowing effectively starts to take effect and has a default value of 1.3e17/cm3. The equivalent default setting consequently should be written as:

```
MATERIAL BGN.E=6.92e-3 BGN.C=0.5 BGN.N=1.3e17
```

You can alter these parameters to modify the current gain of the device in the medium injection regime. For example, reducing the linear parameter from 6.92e-3 to 6.5e-3 is sufficient to cause a significant increase in current gain in the medium injection region. Although the bandgap narrowing parameters affect both collector and base currents, the base current is affected to a greater degree. The most sensitive plot to see the effect of small changes to bandgap narrowing is a plot of current gain versus log of collector current. A reduction in bandgap narrowing will result in an increase in current gain in the medium current injection region.

2.6.5 The Base Current Profile – Low Injection

This is one case where there is an interdependency on one parameter, since the intrinsic base resistance not only affects the collector current in all regions (see the previous section) Figure 2-37, however, also has an effect on the base current in the low injection region.

For a small range of implant doses around the optimum, the base doping concentration will also affect the position of the knee or the rate or both of fall off of the base current in the low injection operating region of the device. This is most noticeable as a loss of current gain in the low injection region for the alternative standard plot of current gain versus collector current. An increase in the base implant reduces the intrinsic resistance and typically increases the base current in the low injection region, resulting in a decrease in current gain for very low currents.

A similar effect to increasing the base doping is observed if the base doping is kept constant but the overall doping is reduced in the mono-crystalline silicon region of the emitter. You can tune the doping profile in the mono-crystalline region of the emitter using three parameters in Athena. The main physical effect of these Athena parameters is to change the doping profile of the emitter in the mono-crystalline silicon. These process parameters are as follows:

- The total interstitial concentration in the poly-emitter.
- The dopant segregation effects in the poly-emitter.
- The dopant velocity across the silicon/polysilicon boundary.

The first process parameter will affect how quickly the dopant in an implanted poly-emitter reaches the silicon/polysilicon boundary during the RTA diffusion and therefore affects the total diffusion of dopant into the single crystalline part of the emitter and the base width doping profile.

The second process parameter affects dopant pile-up at the poly-silicon/silicon boundary and therefore the source doping concentration at the mono-crystalline interface. Once again, this will affect the overall doping profile of the emitter in the mono-crystalline region of the device.

The third process parameter affects the velocity of transport of dopant across the polysilicon/ silicon boundary with similar effects to the parameters above.

You can use these parameters to tailor the emitter doping profile in the mono-crystalline silicon region to match available measured data, usually in the form of SIMS or capacitance information. An accurate profile of dopant in the poly-silicon part of the emitter is not too important if measured data concerning interfacial dopant concentrations is available. This is because the work function of the poly-emitter will be set in Atlas by defining the poly-emitter as an electrode. All you need to calculate the correct work function at the poly-silicon emitter is the interfacial doping concentration at the poly-silicon/silicon interface on the poly side of the junction. See the “Poly-emitter work function” on 77 for setting the correct work function for the poly-emitter .

Conclusions

By using a logical combination of tuning parameters available in both the process simulator (Athena) and the device simulator (Atlas) and with the influence of each parameter, you can get a good match for bipolar transistors for most device designs.

Since it is usually less problematic to match the collector current for all levels of applied base-emitter voltage compared to the matching of base current, you will probably find that more time is spent trying to match the base current for very small and very large values of applied base-emitter voltage. You should, however, spend a good amount of time on making sure that the correct process models are used in the process flow to reduce the overall uncertainty as to which parameters require calibration.

2.7 Using Athena for Simulating SiGe Process

The recommended method for simulating SiGe process is to treat germanium as a dopant in silicon rather than depositing the SiGe material. This is because it allows boron diffusivities to be germanium concentration dependant by using the `model.sige` parameter in the `METHOD` statement. You can also define SiGe related parameters in the `MATERIAL` statement.

The example below show a typical set of statements for depositing a 200A thick trapezoid profile SiGe base region in an HBT may be as follows:

```
METHOD   FULL.CPL MIN.TEMP=600 MODEL.SIGE

MATERIAL SILICON NIFACT.SIGE=100 EAFACT.SIGE=1.5 NO.FLIP

DEPOSIT  SILICON THICK=0.02 DIVISIONS=8 C.BORON=5E18 \

         C.GERMANIUM=1E22 F.GERMANIUM=1E21

DIFFUSE  TIME=4 TEMP=650 NITRO PRESS=1
```

This section will give a brief explanation of the statements and possible variances. A more detailed description of the individual parameters are given in the Athena notes files and in Chapter 6 “Statements”.

2.7.1 METHOD Statement

`FULL.CPL` is the recommended diffusion model for boron to obtain the best accuracy. It is, however, slower than other diffusion models due to the larger number and more complex inter-relations taken into account.

`MIN.TEMP` is required if the deposition temperature is below 700°C, which is the minimum calibrated temperature for Athena in standard processing. Select a `MIN.TEMP` value below the deposition temperature otherwise, no diffusion will be calculated. The `MODEL.SIGE` parameter invokes the silicon germanium models.

2.7.2 MATERIAL Statement

The following statement specifies reasonable user-definable parameters for the SiGe models in the material silicon.

```
MATERIAL SILICON NIFACT.SIGE=100 EAFACT.SIGE=1.5 NO.FLIP
```

`The  NO.FLIP` parameter prevents automatic mesh optimization, which preserves the user-defined x grid spacing. Athena often will try to remove what it believes are excessive grid points during `DEPOSITION/ETCH` statements. It is not advisable to remove mesh points in the base of an HBT, which is the usual SiGe application.

2.7.3 DEPOSIT Statement

This statement is used to specify the material (silicon), thickness, gridding, and doping parameters. Once you specify these parameters, specify the germanium dopant concentration either as a constant or as a graded doping profile as shown in the example.

To calculate the doping required, the total number of atoms in silicon is taken to be 5e22/cm3 for simplicity. Therefore, a doping of 1e22/cm3 of germanium is a 20% concentration. Other concentrations are calculated in proportion to this, so a final concentration at the end of the deposition of 1e21 represents a 2% concentration of germanium. Therefore, the example deposits a 200A thick SiGe film with an initial germanium concentration of 20% and a final germanium concentration of 2%.

2.7.4 DIFFUSE Statement

Since this example above used the DEPOSIT statement, the thermal budget for deposition is simulated by an inert diffusion for the deposition time. A typical deposition temperature is around 650°C.

Generally, a typical SiGe HBT device would have a base profile consisting of boron and non-boron doped regions together with a tapered germanium profile at both ends of the base. In this case, simply specify the DEPOSIT and DIFFUSE statements in several stages. You can also use several EPITAXY statements to do the same thing.

At the end of the process simulation, the germanium dopant profile has to be converted into SiGe material with a variable X composition to pass the device into Atlas. To do this, type in the following STRUCTURE command example.

```
STRUCT OUTFILE=HBT.STR SIGE.CONV
```

This will save a structure call HBT.str and converts the germanium dopant profile into the correct X composition SiGe material.

2.8 Using Advanced Features of Athena

2.8.1 Structure Manipulation Tools

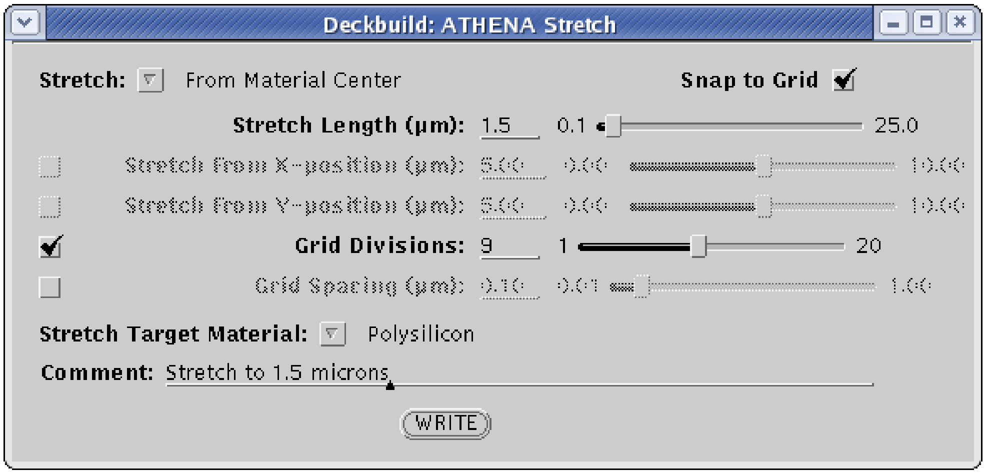
Using the Structure FLIP Capability

The `Structure FLIP` capability allows you to flip the structure in the x axis. The `STRUCT FLIP.Y` statement causes the structure to be vertically flipped.

This operation can be useful if some process steps (e.g., etching, deposition, or implant) take place from the backside of the wafer. By using this statement, you can flip a structure, perform these steps, and then flip it back.

Using the Stretch Capability

In some cases, a device characterization as a function of length is of interest. For example, the drain current characteristics depend strongly on the gate length. The **Stretch** capability makes it possible to generate a number of MOSFET structures with different gate lengths from one Athena simulation. The structure obtained so far in this tutorial (See Figure 2-22) has a gate length of 0.6 µ. To increase the gate length to 1.5µ, use the `STRETCH` command. To use this capability, select **Structure→Stretch** in the **Commands** menu and the Athena Stretch menu (Figure 2-39) will appear.



Stretch: ▽ From Material Center
Snap to Grid ☑
Stretch Length (µm): 1.5 0.1 25.0
□ Stretch from X-position (µm): 5.00 0.00 10.00
□ Stretch from Y-position (µm): 5.00 0.00 10.00
☑ Grid Divisions: 9 1 20
□ Grid Spacing (µm): 0.10 0.01 1.00
Stretch Target Material: ▽ Polysilicon
Comment: Stretch to 1.5 microns

Figure 2-39 Athena Stretch Menu

Then, select **Polysilicon** from the **Stretch Target Material** menu. Next, set **Stretch Length** to 1.5µ, and choose **10** as the number of **Grid Divisions**. Then, press the **Write** button and the following command will appear in the input file:

```
# STRETCH TO 1.5 MICRONS

STRETCH LENGTH=1.5 POLY SNAP DIVISION=10
```

As a result, the polygate will be stretched from its initial length of 0.6 µ (left plot in Figure 2-40) to 1.5*µ* (right plot in Figure 2-40).

Ten additional vertical grid lines will be inserted in the center of the gate area. The `LENGTH` parameter of the `STRETCH` command can serve as a split parameter for the **Virtual Wafer Fab Split Experimentation** capability. For more information about this capability, see the VWF Automation and Production Tools User's Manual.

Another use of the **Stretch** capability is in the simulation of large power device structures, where active areas are uniform everywhere except in close proximity to the mask edges and are separated from each other by long non-active or isolation regions. You can simulate a shrunken structure and then stretch the active or non-active or both areas to the actual widths. This will also save a tremendous amount of simulation time.

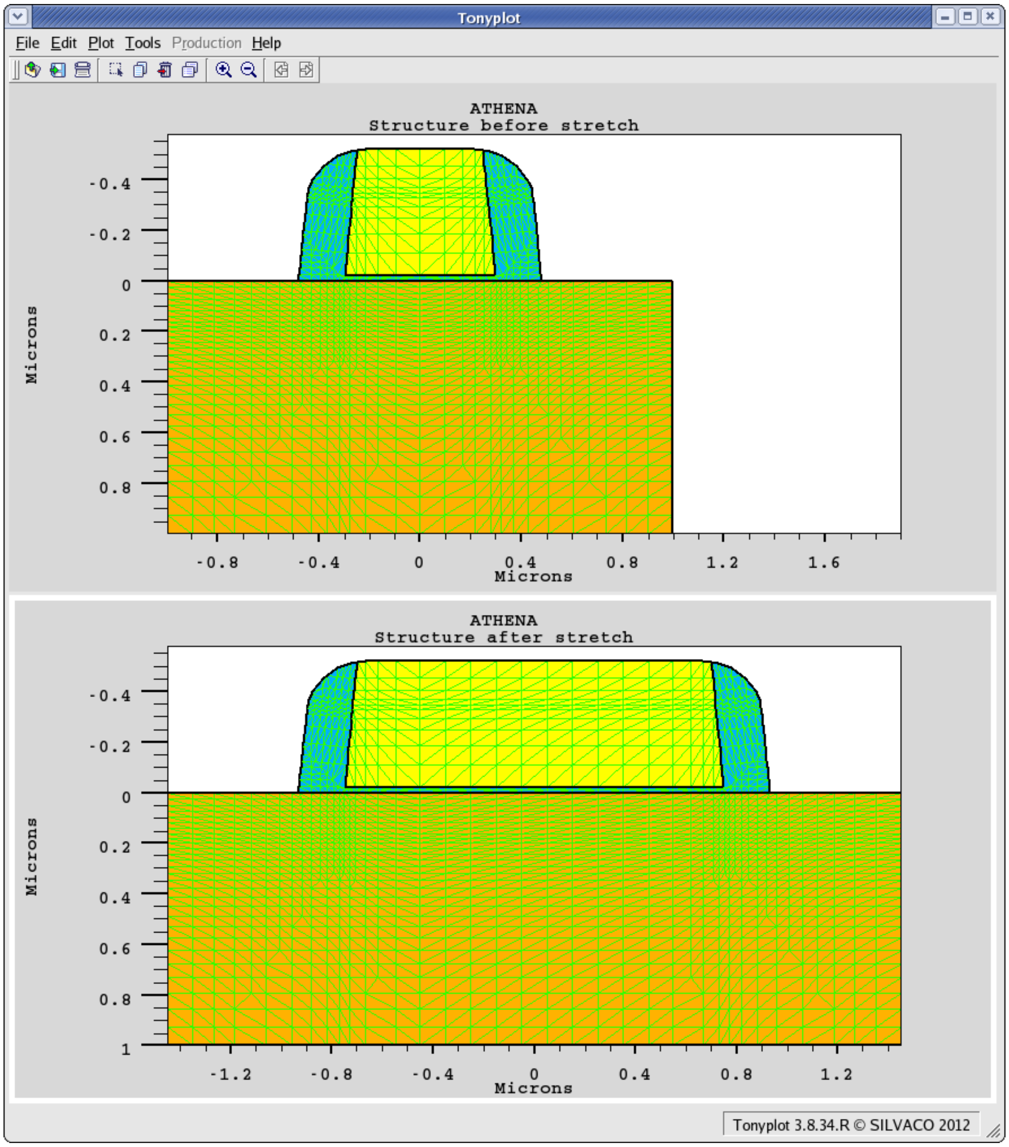


Figure 2-40 Using Stretch Function for a MOSFET Structure

Note: The stretch function can save a great deal of CPU time.

Using Athena In 1D Mode

You can increase the simulation speed by running Athena in 1D mode. Athena automatically runs in 1D mode by default initially. The simulation will automatically converts to 2D mode when you perform a two dimensional simulation process, such as ETCH or EXPOSE. Simple operations such as conformal deposits, oxidation, and diffusion run faster in 1D mode.

The deposition and etch sequences displayed in Figure 2-41 show a sequence of 1D depositions with an automatic conversion to 2D at the first etch.

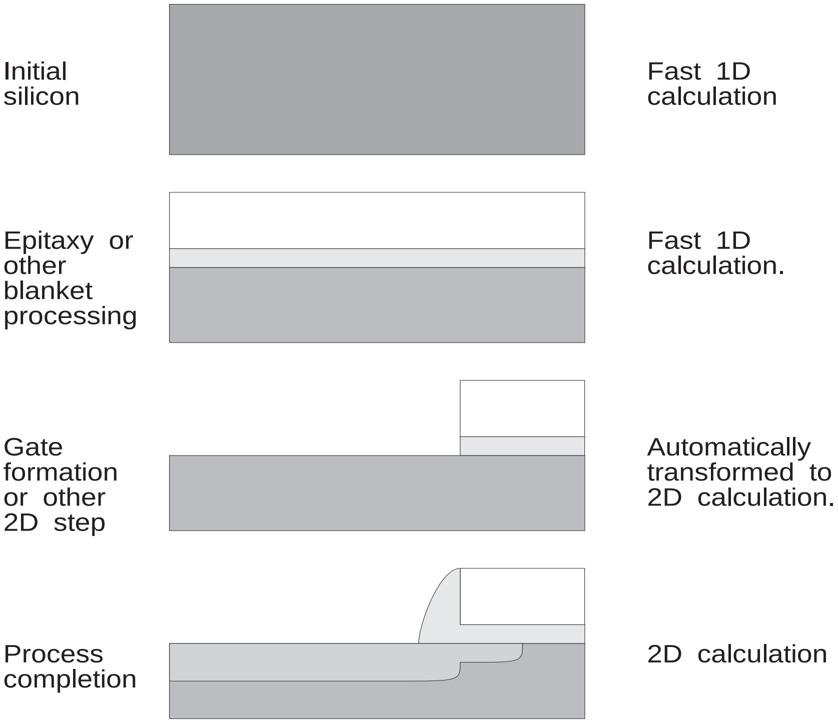
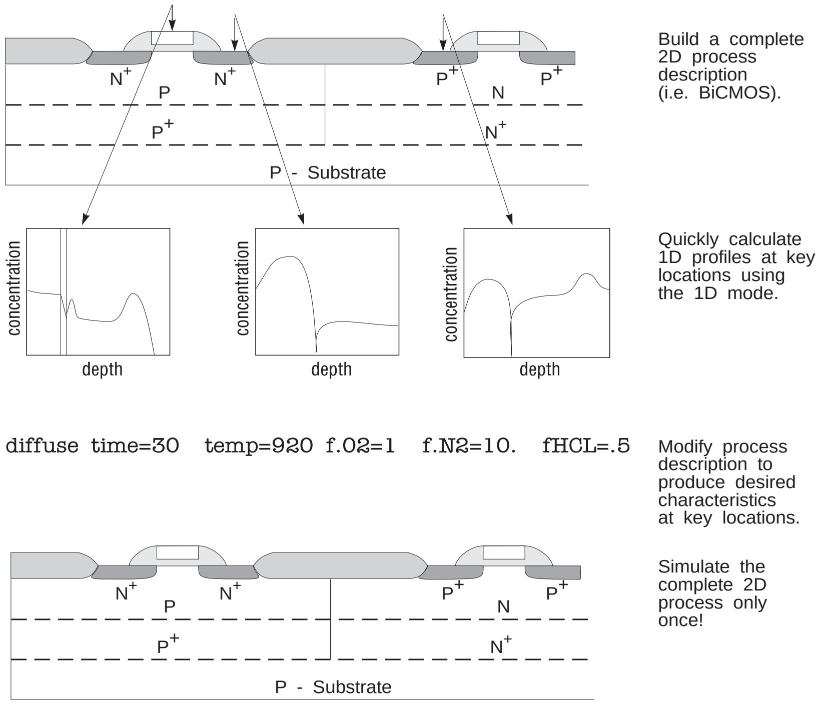

Initial silicon
Fast 1D calculation
Epitaxy or other blanket processing
Fast 1D calculation.
Gate formation or other 2D step
Automatically transformed to 2D calculation.
Process completion
2D calculation

Figure 2-41 Automatic 1D to 2D Conversion

Figure 2-41 shows another aspect of 1D mode. In this case, the INITIALIZE command is specified with the parameters ONE.D and X.LOC=<n> (see Section 6.30 “INITIALIZE”). ONE.D specifies that a one dimensional calculation is to be done at the location X.LOC.

In the case of Figure 2-42, 1D profiles are generated at different X locations of a complicated BiCMOS structure. This allows you to quickly check of the overall process using the 1D mode.



Build a complete 2D process description (i.e. BiCMOS).

Quickly calculate 1D profiles at key locations using the 1D mode.

```
diffuse time=30 temp=920 f.O2=1 f.N2=10. fHCL=.5
```

Modify process description to produce desired characteristics at key locations.

Simulate the complete 2D process only once!

Figure 2-42 Use of One Dimensional Mode

2.8.2 Deposition and Wet/Dry Etching using the Physical Models in Athena/Elite

This section describes the deposition and etch capabilities of the Elite module of Athena using Athena/Elite default machines

To use Athena/Elite's physically based deposition and etch models, use at least one of the following steps:

1. Use one of the predefined machines described in Section 4.4 “Etch Models”.
2. Within your input file, modify the specification of one of these predefined machines to fit your process
3. Define a custom machine
4. Invoke a prepared file that defines machines of interest

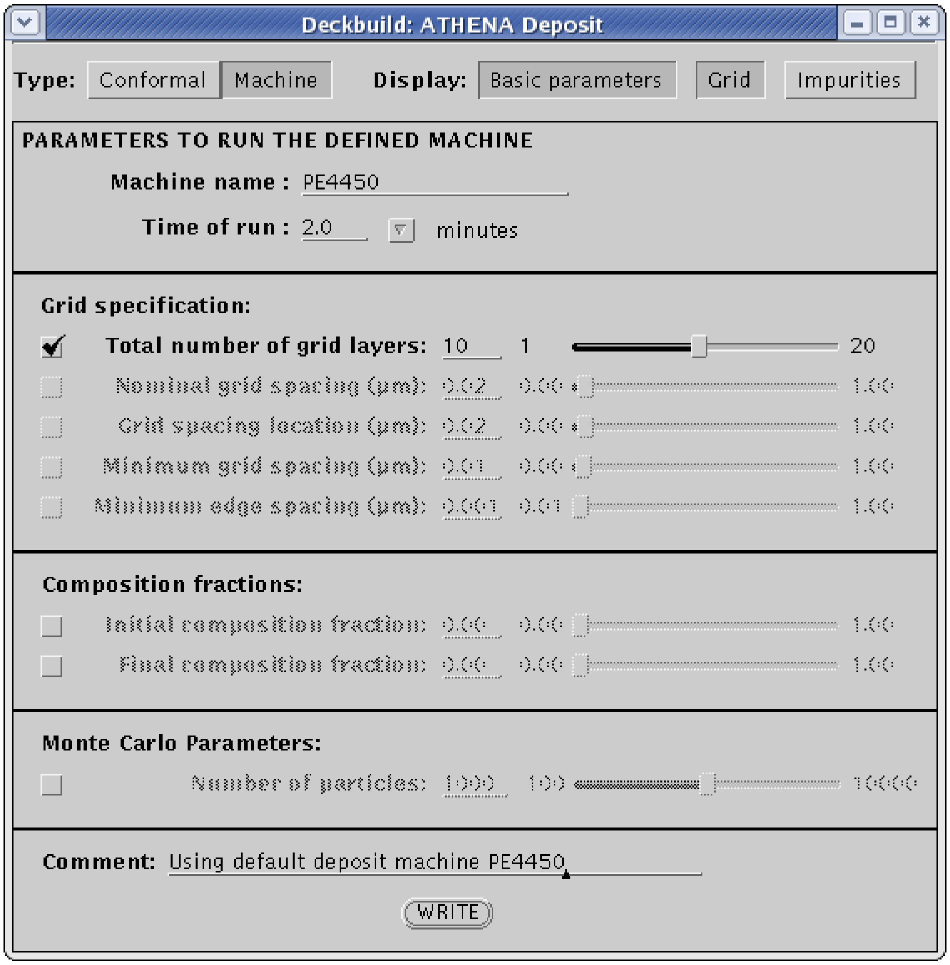
For example, one of the predefined machines is named PE4450. This machine deposits aluminum at a rate of 1 micron/minute from a hemispheric source.

To simulate the effects of two minutes of operation of this machine, open the Athena Deposit Menu and select the **Machine** checkbox. The section **PARAMETERS TO RUN THE DEFINED MACHINE** will appear in the menu (See Figure 2-43). In this section, specify `PE4450` as the **Machine Name**, the **Time of run** (2.0), and the time units (menu box beside the **Time of run** field) as **minutes**.

We recommend that you specify the **Total number of grid layers** in the deposited material region. If you set this number to **10**, it will insert the following Athena/Elite `DEPOSIT` statement:

```
# USING DEFAULT DEPOSIT MACHINE PE4450

DEPOSIT MACHINE=PE4450 TIME=2.0 MINUTES DIVISIONS=10
```

You can specify impurity concentrations in the deposited region in the **Impurity concentration** section of the Athena Deposit Menu by clicking on the **Impurities** box.


Deckbuild: ATHENA Deposit
Type: Conformal Machine
Display: Basic parameters Grid Impurities
WRITE

PARAMETERS TO RUN THE DEFINED MACHINE

Machine name : PE4450

Time of run : 2.0 minutes

Grid specification:

☑ Total number of grid layers: 10 1 20

□ Nominal grid spacing (µm): 0.02 0.00 1.00

□ Grid spacing location (µm): 0.02 0.00 1.00

□ Minimum grid spacing (µm): 0.01 0.00 1.00

□ Minimum edge spacing (µm): 0.001 0.01 1.00

Composition fractions:

□ Initial composition fraction: 0.00 0.00 1.00

□ Final composition fraction: 0.00 0.00 1.00

Monte Carlo Parameters:

□ Number of particles: 1000 100 10000

Comment: Using default deposit machine PE4450

Figure 2-43 Athena Deposit Menu with Machine Section

Modifying Athena/Elite Default Machines

The file *athenamod* defines PE4450 as follows (Notice that a (\ ) is used to concatenate or continue a long input line):

```
RATE.DEPO MACHINE=PE4450 ALUMINUM \
    U.M SIGMA.DEP=.35 HEMISPHE DEP.RATE=1.0 \
    ANGLE1=72 ANGLE2=-70
```

The machine is modeled with a hemispherical deposition model. The deposition rate is 1 micron/minute. The logical parameter, U.M, specifies what units are used, in this case, microns per minute. Finally, the angles of incidence of the hemispherical deposition with respect to the surface normal are specified with the ANGLE1 and ANGLE2 parameters. You can modify these characteristics of the machine PE4450 by copying the specification to the input file and using an ASCII editor. For example:

```
RATE.DEPO MACHINE=PE4450 ALUMINUM \
    U.M SIGMA.DEP=.35 HEMISPHE DEP.RATE=.5 \
    ANGLE1=72 ANGLE2=-70
```

redefines machine PE4450 to have a deposition rate of 0.5 micron/minute.

Defining Elite Deposition Machines

You can define your own deposition machine using the Athena Rate Deposit menu (Figure 2-44). To open this menu, select **Process→Deposit→Rate Deposit** in the Commands menu. Machine definition requires the specification of five general parameters and one or several model-specific parameters. The general parameters that must be specified are the following:

- Machine name (e.g., **TEST01**): This parameter uniquely identifies the machine.
- Material name (e.g., **aluminum**): A user-defined material.
- Machine (model) type (e.g., **unidirectional**): You can select one of six models by pressing the appropriate button.
- Deposition (rate units specifier e.g., **A/min**): You can select one of seven unit specifiers from the menu.
- Deposition rate (e.g., **1000**): This parameter is specified in the user-selected units.

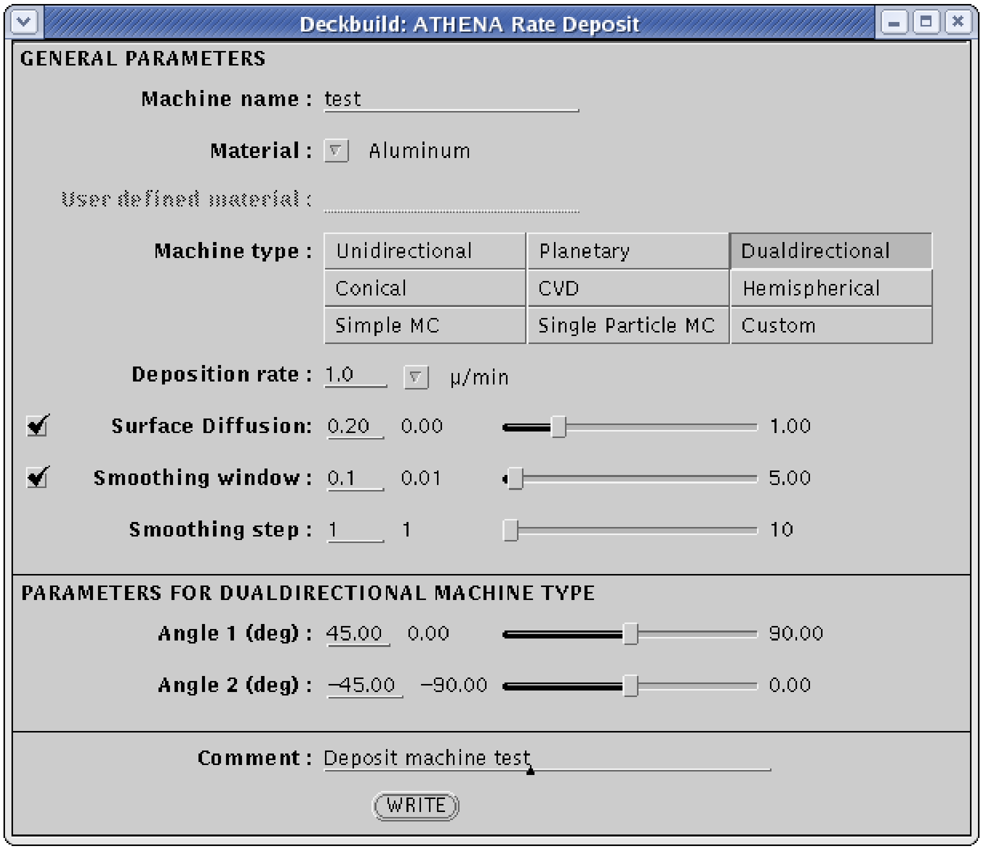
The SIGMA.DEP parameter is optional and defaults to 0.2.

The SMOOTH.WIN and SMOOTH.STEP parameters provides an alternative to a complete reflow calculation. It allows a geometric averaging over a window of width (SMOOTH.WIN) microns that is performed over a number of steps (SMOOTH.STEP). These parameters perform a post-deposition smoothing that effectively emulates a reflow process. The wider smoothing window produces a more intensive surface redistribution of the deposit material. The default number of smoothing operations (1) is adequate for most applications.

One or several model-specific parameters are attributed to each model. For example, only the ANGLE1 parameter is required for the unidirectional model. Table 2-4 indicates which parameters are required for each model. The **Machine Type** section of the Athena Rate Deposit Menu includes only those parameters that are relevant to the selected model. Each parameter has a default value which will be inserted in the input file. If some of the parameters are undefined, the simulation may be invalid or may produce unpredictable results.

If the Athena Rate Deposit menu is set as shown in Figure 2-44, the following RATE.DEPO statement will be inserted into the input file.

```
RATE.DEPO MACHINE=TEST ALUMINUM U.M SIGMA.DEP=0.2
    SMOOTH.WIN=0.1 SMOOTH.STEP=1 DUALDIREC
    DEP.RATE=1.0 ANGLE1=45.0 ANGLE2= -45.0
```



Deckbuild: ATHENA Rate Deposit

GENERAL PARAMETERS

Machine name : test

Material : Aluminum

User defined material :

Machine type : Dualdirectional

Deposition rate : 1.0 μ/min

☑ Surface Diffusion: 0.20 0.00 1.00

☑ Smoothing window : 0.1 0.01 5.00

Smoothing step : 1 1 10

PARAMETERS FOR DUALDIRECTIONAL MACHINE TYPE

Angle 1 (deg) : 45.00 0.00 90.00

Angle 2 (deg) : -45.00 -90.00 0.00

Comment : Deposit machine test

Figure 2-44 Athena Rate Deposit Menu

Table 2-4 Deposition Model Required Parameters

Parameters	Models									
	CVD	UNI	DUAL	HEMI	CONIC	PLANET	MONTE1	MONTE2	CUSTOM1	CUSTOM2
dep.rate	yes	yes	yes	yes	yes	yes	yes	yes	optional	yes
step.cov	yes	no	no	no	no	no	no	no	no	no
angle1	no	yes	yes	yes	yes	yes/no*	yes	yes	no	no
angle2	no	no	yes	yes	no	yes	no	no	no	no

Table 2-4 Deposition Model Required Parameters (Continued)

Parameters	Models									
	CVD	UNI	DUAL	HEMI	CONIC	PLANET	MONTE1	MONTE2	CUSTOM1	CUSTOM2
`c.axis`	no	no	no	no	yes	yes	no	no	no	no
`p.axis`	no	no	no	no	yes	yes	no	no	no	no
`dist.pl`	no	no	no	no	no	yes/no *	no	no	no	no
`sigma.dep`	no	optional	optional	optional	optional	optional	optional	optional	no	yes
`smooth.win`	no	optional	optional	optional	optional	optional	optional	optional	optional	optional
`smooth.step`	no	optional	optional	optional	optional	optional	optional	optional	optional	optional

* To use the planetary model, either the ANGLE1 or the DIST.PL parameter must be specified. These parameters are mutually exclusive.

Defining Elite Etch Machines

An Athena/Elite etch machine can be defined using the Athena Rate Etch Menu (Figure 2-45).

To open this menu, select **Process→Etch→Rate Etch** in the **Commands** menu. The machine definition requires the specification of four general parameters and one or several model-specific parameters.

The general parameters that must be specified are as follows:

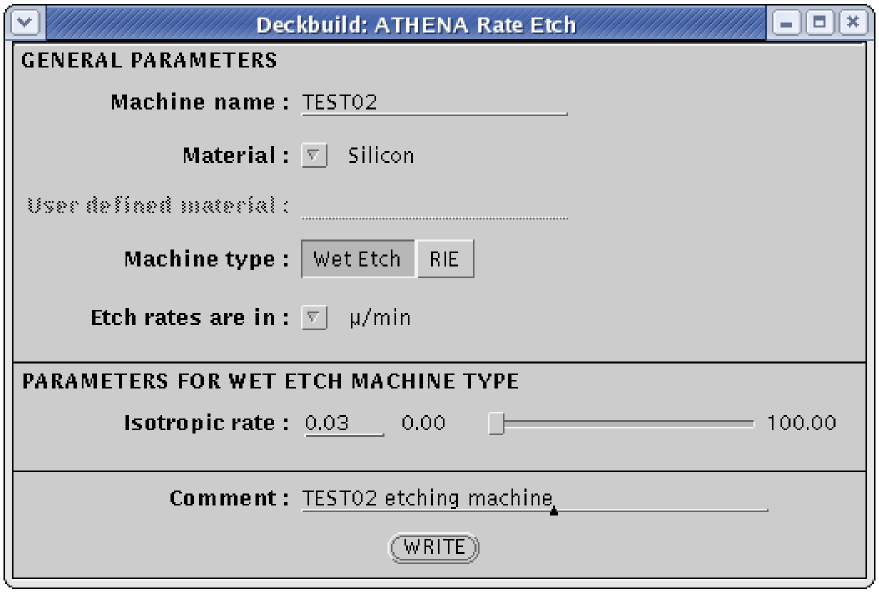
- Machine name (e.g. **TEST02**): This parameter uniquely identifies the machine.
- Material name (e.g., **silicon**): A user -defined material can also be specified.
- Machine type (e.g., **Wet Etch**): You can select one of two models by pressing the appropriate button.
- Etch rate units specifier (e.g., **A/min**): You can select one of seven unit specifiers from the menu.

One or several model-specific parameters are attributed to each model. For example, only the ISOTROPIC rate parameter is required for the Wet Etch model.

Table 2-5 indicates which parameters are required for each of the three models. The **Parameters for Specific Machine Type** section of the Rate Etch menu includes only those parameters, which are relevant to the selected model.

If the Athena Rate Etch Menu is set as shown in Figure 2-45, the following RATE.ETCH statement will be inserted into the input file:

```
# TEST02 ETCHING MACHINE
RATE.ETCH MACHINE=TEST02 SILICON U.M WET.ETCH ISOTROPIC=0.03
```


Deckbuild: ATHENA Rate Etch
GENERAL PARAMETERS

Machine name : TEST02

Material : Silicon

Machine type : Wet Etch RIE

Etch rates are in : µ/min

PARAMETERS FOR WET ETCH MACHINE TYPE

Isotropic rate : 0.03 0.00 100.00

Comment : TEST02 etching machine

Figure 2-45 Athena Rate Etch menu

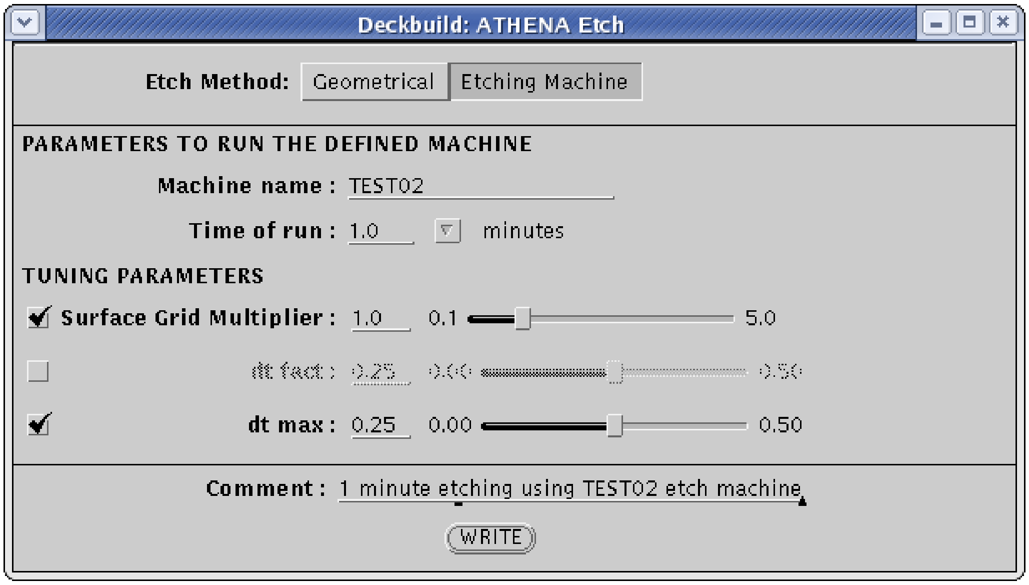
If several materials are present in the structure to be etched, etch rates for each material type should be specified in separate RATE.ETCH statements.

Table 2-5 Allowable Etch Model Parameters

Parameters	wet.etch	rie
isotropic	yes	yes
directional	no	yes
divergence	no	yes
chemical	no	yes

Using A Specified Etch Machine

When etch rates for a specific machine are specified using RATE.ETCH statements, you can simulate the effects of the operation of this machine. To simulate the etch process using a specified etch machine, open the Athena Etch Menu and check the **Etching Machine** box. The **Parameters to Run the Defined Machine** section will appear in the menu (See Figure 2-46).



Etch Method: Geometrical | Etching Machine

PARAMETERS TO RUN THE DEFINED MACHINE

Machine name : TEST02

Time of run : 1.0 minutes

TUNING PARAMETERS

☑ Surface Grid Multiplier : 1.0 0.1 5.0

□ dt fact : 0.25 0.00 0.50

☑ dt max : 0.25 0.00 0.50

Comment : 1 minute etching using TEST02 etch machine

Figure 2-46 Athena Parameters to Run the Define Machine Etch Menu Section

The **Machine Name** (TEST02), the time units (e.g., minutes) and the **Time of run** (e.g., 1.0) must be specified. There are also two tuning parameters that control time stepping during the etch process. To improve the smoothness of the etch surface, decrease the maximum time step parameter, DT.MAX, from its default value of 10 percent of the specified **Time of Run** value.

If you set the Athena Etch Menu as shown in Figure 2-46, the following ETCH statement will appear in the input file when you press the **WRITE** button.

```
# 1 MINUTE ETCHING USING TEST02 ETCH MACHINE

ETCH MACHINE=TEST02 TIME=1.0 MINUTES DT.MAX=0.25
```

A new parameter, DX.MULT, will be added to the ETCH statement to allow enhanced discretization during individual Elite Etch steps. Increasing the value of DX.MULT from its default value of 1.0 will result in larger surface segments and a reduced discretization.

Decreasing DX.MULT will result in better discretization in both space and time during the etch calculation. Reducing the value of this parameter allows realistic modeling of wet etches that previously were poorly resolved. Use the DX.MULT is preferable to the use of DT.MAX.

2.8.3 MaskViews Interface

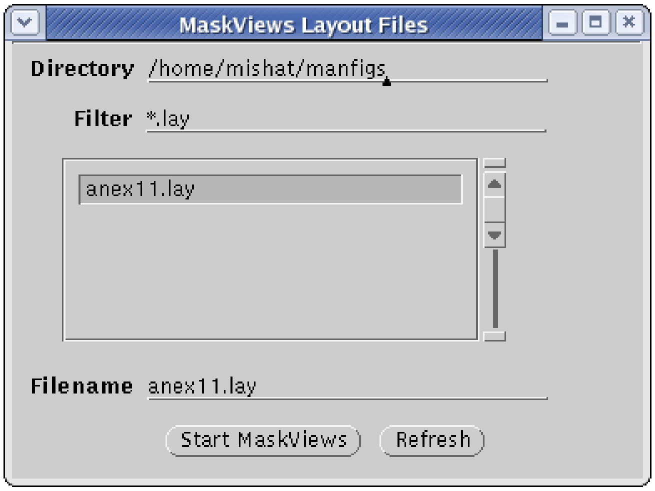
This section describes an alternative to the manual specification of grid and etch steps described in Section 2.3 “Creating a Device Structure Using Athena”.

Defining Initial Rectangular Grid Using MaskViews

An initial rectangular grid can also be defined by using Silvaco’s IC layout editor, MaskViews. MaskViews is designed specifically for interfacing IC layout information with process and device simulators. For more detailed information about MaskViews and its interface with DeckBuild, see the MaskViews User’s manual. This section gives several practical suggestions on how to prepare a good initial grid for Athena.

With MaskViews, you can omit Athena mesh definition statements because you can include the gridding information in the layout file. When using MaskViews to provide line information, DeckBuild will comment out existing line commands when it loads the MaskViews information.

Load the example, *43.11 anex11.in*, from the **ATHENA-COMPLEX** section of the DeckBuild Examples Window (See Figure 2-2). Then, select **MaskViews→Starting MaskViews** from the **Tools** menu of DeckBuild to open MaskViews Layout Files Popup (Figure 2-47).



Directory /home/mishat/manfigs

Filter *.lay

anex11.lay

Filename anex11.lay

Figure 2-47 MaskViews Layout Files Popup

Choose the anex11.lay layout file from the scrolling list and press the **Start MaskViews** button. The MaskViews window will then appear as shown in Figure 2-48.

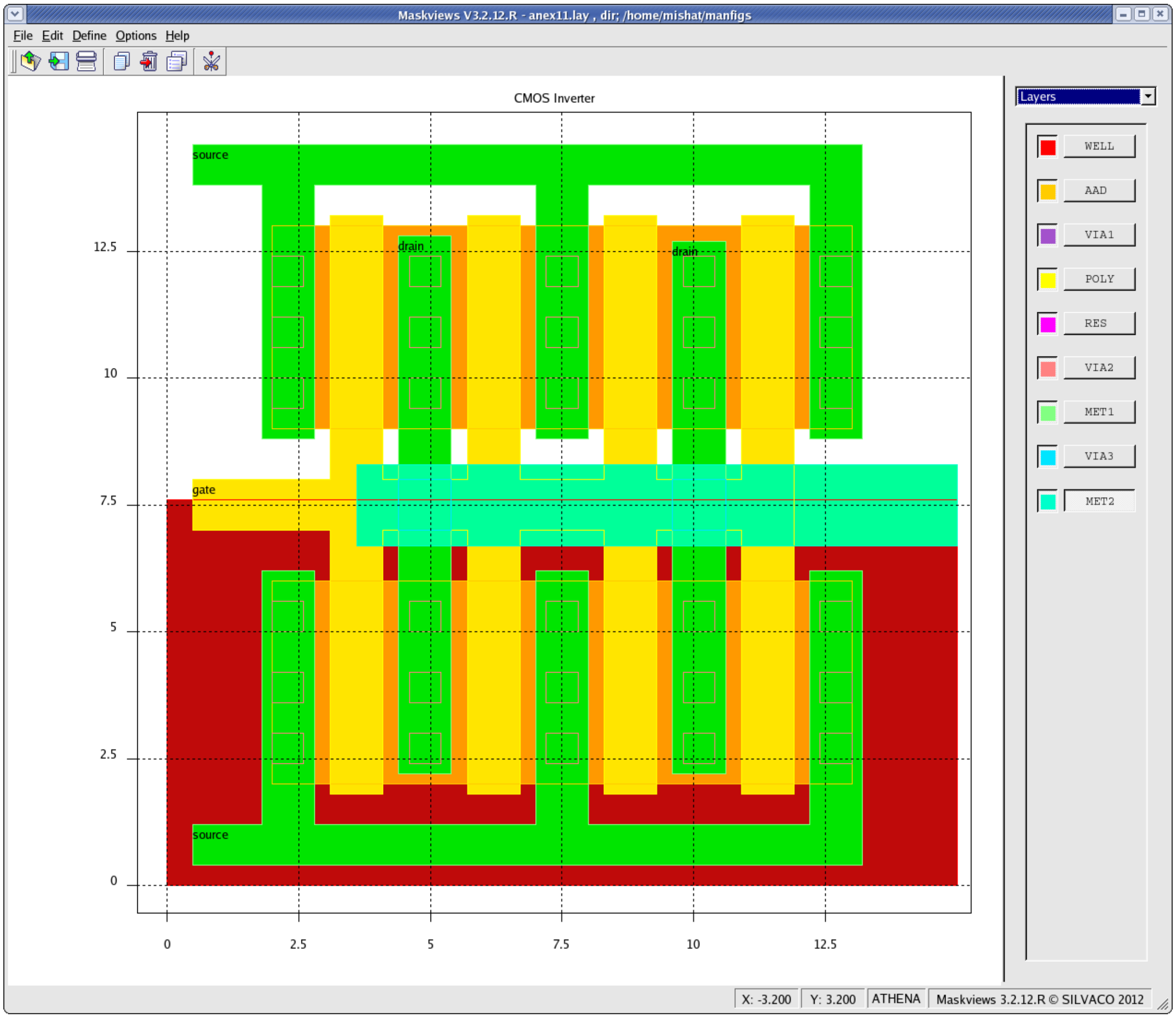
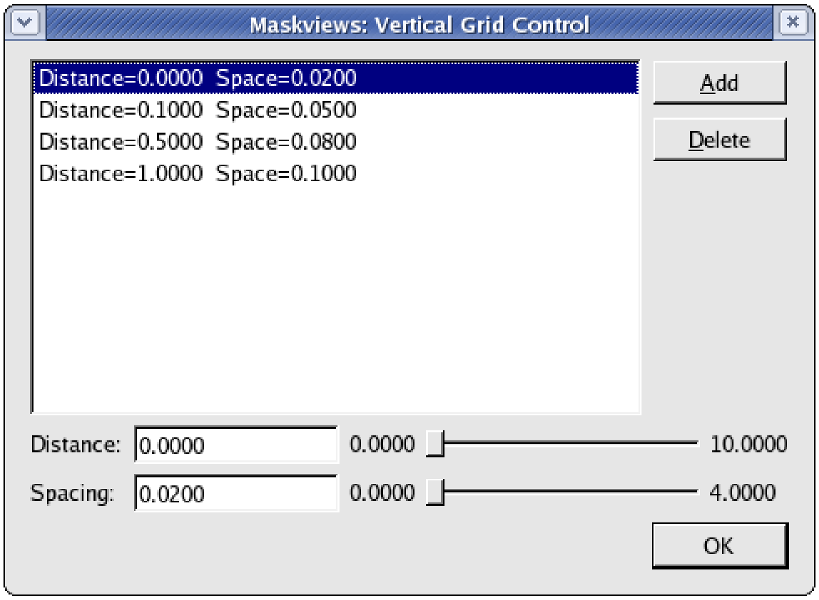


Figure 2-48 MaskViews Window

This section will describe how to modify a grid specification for Athena. First, set the grid in the Y direction by selecting **Grid→Y...** from the **Define** menu. Figure 2-49 shows the Vertical Grid Control popup will appear.

You can add, modify, and delete the lines for Athena initial rectangular grid exactly the same way as using the Athena Mesh Define menu from DeckBuild as previously described. Note that the `Distance` parameter is equivalent to the `location` parameter in Athena. Also, the **Add** button is equivalent to the **Insert** button of the Athena Mesh Define Menu. Then, press **Return** after entering the **Distance** or the **Spacing** values.

If the **Distance** and **Spacing** are set as shown in Figure 2-49, the grid will be the same in the Y-direction as the grid produced using the Athena Mesh Define Menu.



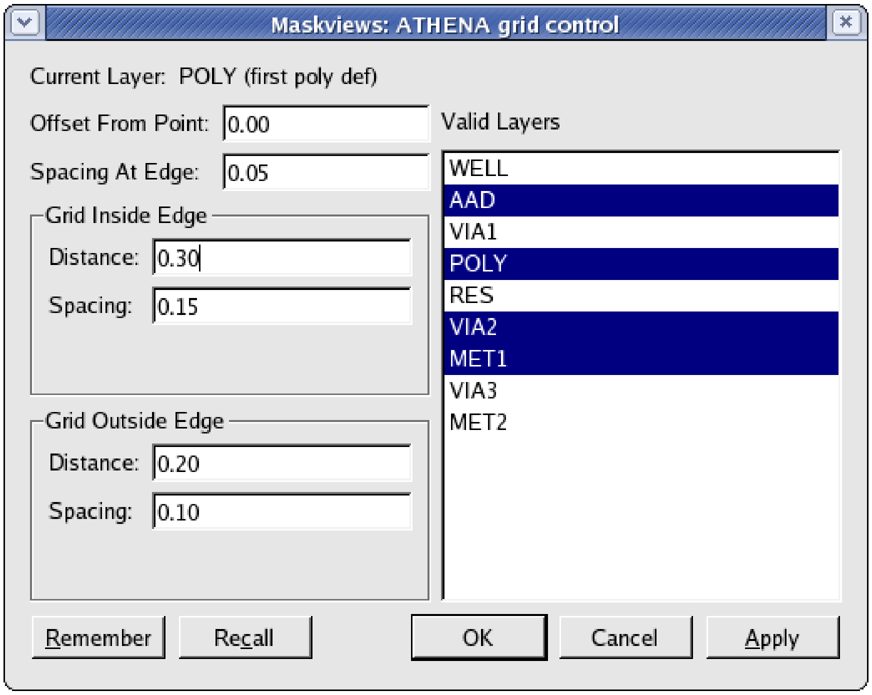
Maskviews: Vertical Grid Control

Distance=0.0000 Space=0.0200
Distance=0.1000 Space=0.0500
Distance=0.5000 Space=0.0800
Distance=1.0000 Space=0.1000

Distance: 0.0000 0.0000 10.0000
Spacing: 0.0200 0.0000 4.0000

Figure 2-49 Vertical Grid Control Popup

MaskViews also controls the initial Athena grid in the X-direction. MaskViews generates Athena line statements for each mask edge on valid layers crossed by a cutline. The grid spacing and the validation of layers can be set by the Athena Grid Control Menu (See Figure 2-50). To open this menu, select **Grid→X...** from the **Define** menu.



Maskviews: ATHENA grid control

Current Layer: POLY (first poly def)

Offset From Point: 0.00

Spacing At Edge: 0.05

Grid Inside Edge
Distance: 0.30
Spacing: 0.15

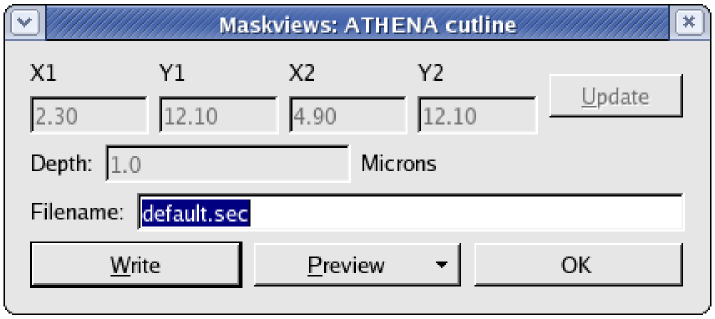
Grid Outside Edge
Distance: 0.20
Spacing: 0.10

Valid Layers
WELL
AAD
VIA1
POLY
RES
VIA2
MET1
VIA3
MET2

Figure 2-50 Athena Grid Control Menu

Figure 2-50 shows the line locations and spacings preset for the POLY layer. This set of parameters means that for each POLY edge crossed by a cutline, three line statements are to be inserted into the Athena input file. The first line will be located exactly at the edge and the spacing will be 0.05µm. The second line will be inside the POLY layer, 0.3µm from the edge, and spacing at this line is 0.15µm. The third line will be outside the POLY layer, 0.2µm from the edge, and its spacing will be set to 0.1µm. You can choose the current edit layer by selecting the **Name** button (located underneath the Layers Menu) for the layer in the key list of MaskViews (Figure 2-48). If you select AAD, then only set one line for an edge of the AAD layer because offset distances are equal to 0.0. We recommend that only one line be set for unimportant layers. It's also important to validate only those layers that are going to be used in Athena MASK statements.

Select **Write Cutline File** from the **File** menu. Then, the prompt **Select first end of the ATHENA cross section** will appear at the bottom panel of the MaskViews window. Press the select (left) mouse button at the desired point in the layout (e.g., within the VIA2 region in the upper left corner of the layout). You will be prompted to select another end of the cross section line. Then, drag the pointer and press the select mouse button on the other end of the selected cutline. Figure 2-51 shows the Athena Cutline Popup, which will then appear. This shows the exact location of the cutline. You can now preview the mask and grid information generated by MaskViews. Press the **Preview** button and the Display Masks Window will appear as shown in Figure 2-52.

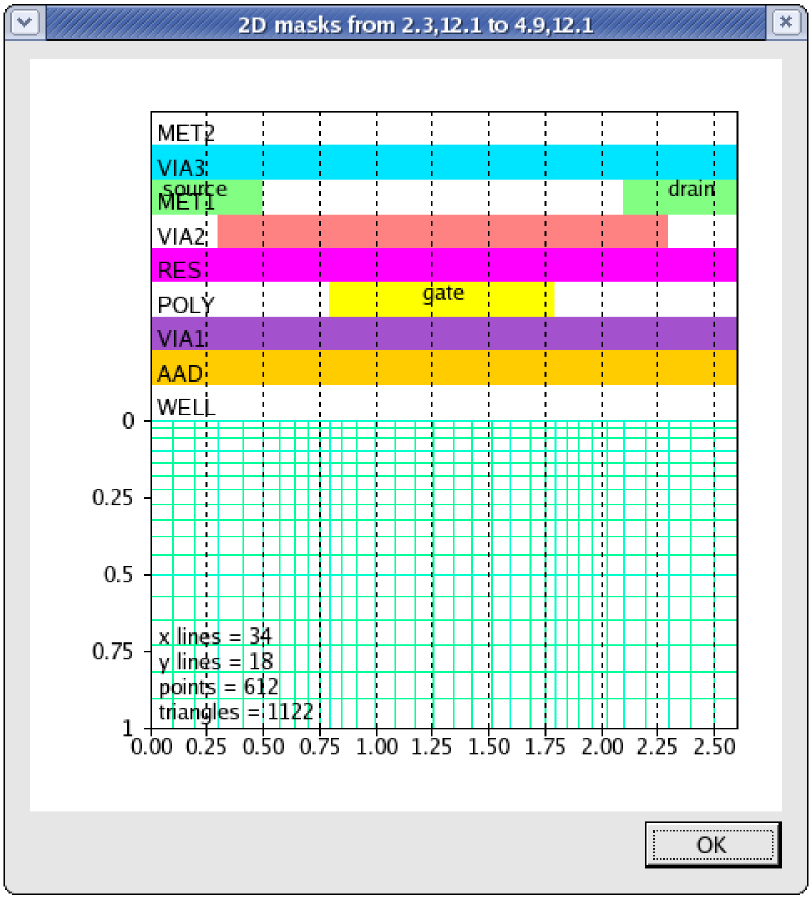


X1	Y1	X2	Y2
2.30	12.10	4.90	12.10

Depth: 1.0 Microns

Filename: default.sec

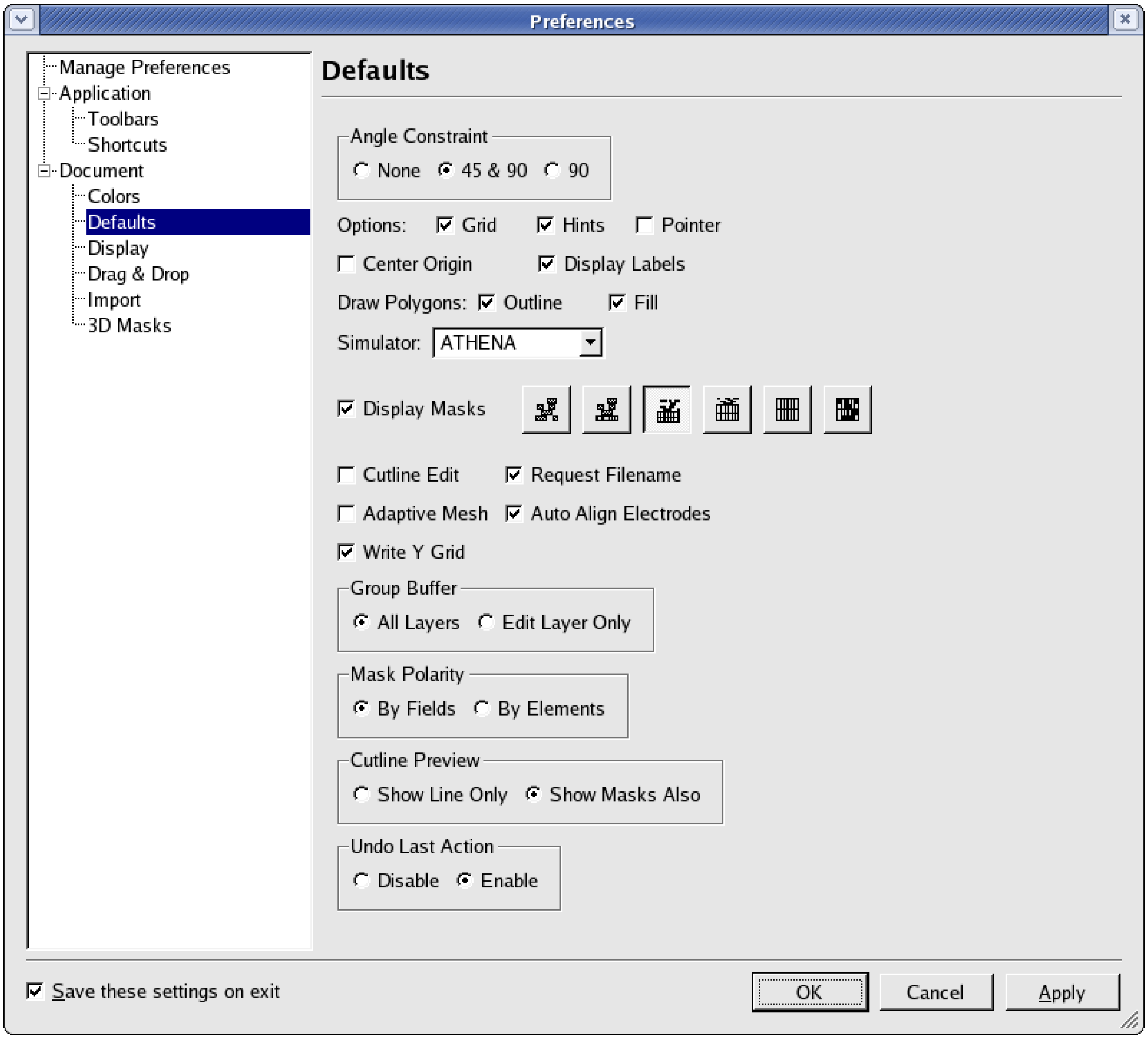
Figure 2-51 Athena Cutline Popup



x lines = 34
y lines = 18
points = 612
triangles = 1122

Figure 2-52 Display Masks Window

The additional information on the number of lines, points, and triangles is also displayed in this window. If the grid does not appear as shown in Figure 2-52. Check **Display Masks** and the third option (**Masks and Grid**) in the **Defaults** menu of the **Preferences** (Figure 2-53).



Defaults

Angle Constraint: ○ None ● 45 & 90 ○ 90

Options: ☑ Grid ☑ Hints □ Pointer

□ Center Origin ☑ Display Labels

Draw Polygons: ☑ Outline ☑ Fill

Simulator: ATHENA

☑ Display Masks

□ Cutline Edit ☑ Request Filename

□ Adaptive Mesh ☑ Auto Align Electrodes

☑ Write Y Grid

Group Buffer: ● All Layers ○ Edit Layer Only

Mask Polarity: ● By Fields ○ By Elements

Cutline Preview: ○ Show Line Only ● Show Masks Also

Undo Last Action: ○ Disable ● Enable

☑ Save these settings on exit

Figure 2-53 Preferences Menu

To select another cutline location, press the **Done** button in the Athena Cutline Popup and repeat the cutline selection process for the desired cutlines one at a time. If you're not satisfied with the grid, you can modify the **X...** or **Y...** or both settings. You can then preview the modified grid without selecting another cutline. For example, if the **Spacing at edge** in the Athena Cutline Popup (see Figure 2-51) is decreased from 0.05 to 0.025, a finer grid will be obtained at both POLY gate edges.

When the location of a cutline and the corresponding grid are satisfactory, the cutline information can be stored or used either as a **Cut** file or as a cutline object.

You can save the **Cut** file by pressing the **Write** button in the Athena Cutline Popup (see Figure 2-51). You can then load this file into DeckBuild for use in Athena by selecting **MaskViews→Cutfiles...** from the **Tools** menu in the DeckBuild window. The MaskViews Cut Files popup (Figure 2-54 left) will then appear. Select the desired `*.sec*` file and press **Load**.

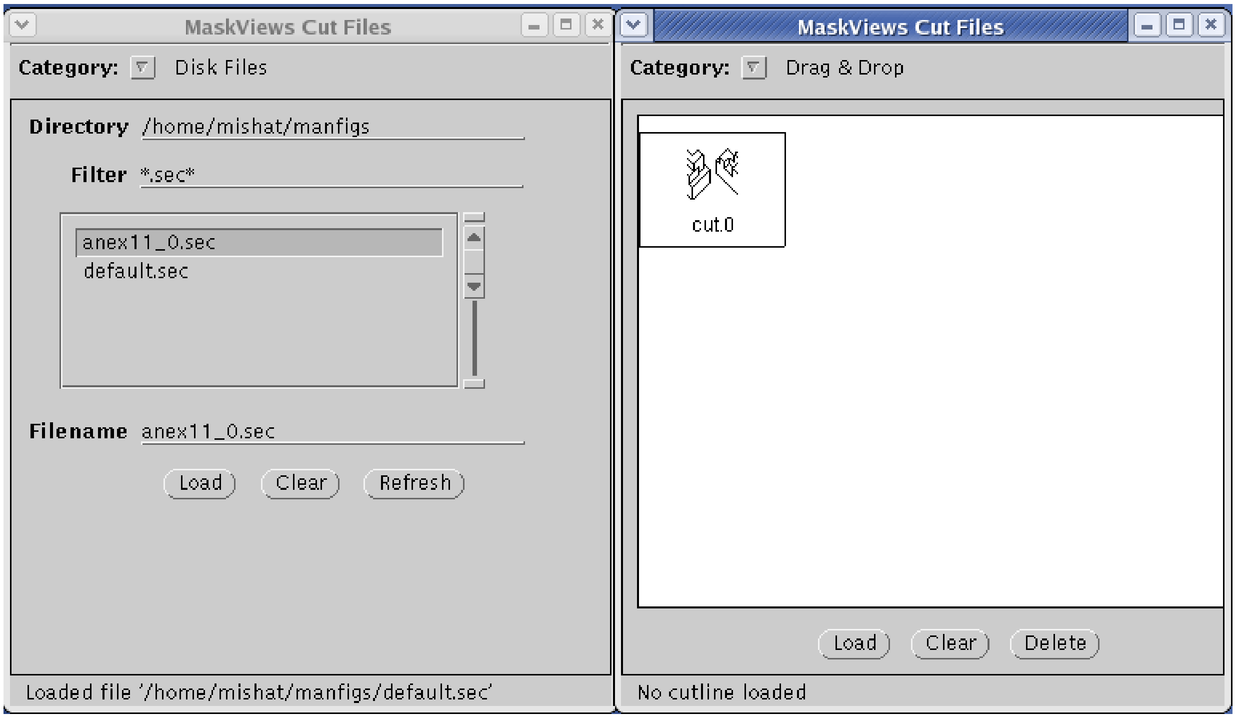


Figure 2-54 MaskViews Cutline Files Window

There is another way of selecting and using cutline information in Athena. This involves select and drag-and-drop operation. To do this, using the Right mouse button, change **Category** in the **MaskViews Cut Files** menu from **Disk Files** to **Drag & Drop** (see Figure 2-54 right). Press the Select mouse button anywhere within the Display Masks window (Figure 2-52) and the cutline icon will appear. Without releasing the Select mouse button, drag the icon into the Drop & Drag window and drop it by releasing the Select mouse button.

Several cutlines with different locations and grids can be dragged and dropped in this fashion. You can then load any of them into DeckBuild by clicking on the **Load** button.

When Athena is loaded with a cutline, DeckBuild will comment out all existing line statements and will automatically run line statements generated by MaskViews. For example, the following output will appear in the DeckBuild Text Subwindow if the `default.sec` generated for the CMOS Inverter is loaded.

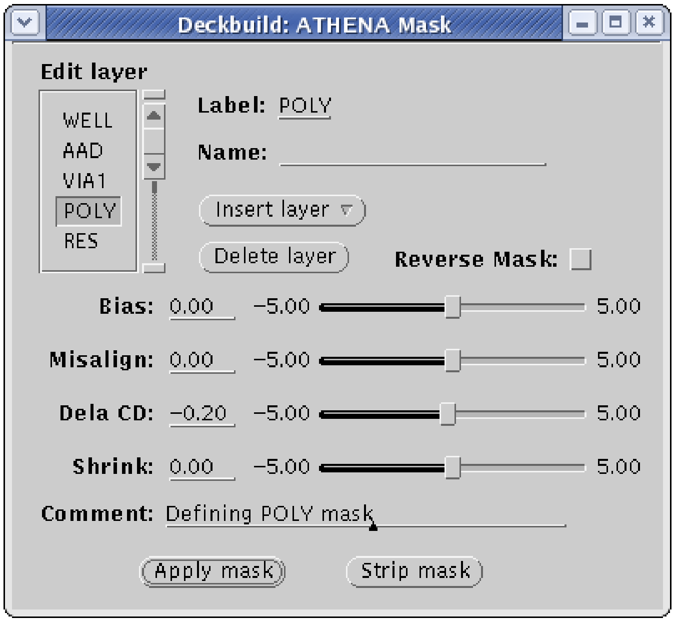
```
ATHENA> LINE X LOC=0.000 SPAC=0.100 TAG=LEFT

ATHENA> LINE X LOC=0.300 SPAC=0.100

ATHENA> LINE X LOC=0.500 SPAC=0.100

ATHENA> LINE X LOC=0.600 SPAC=0.100

ATHENA> LINE X LOC=0.800 SPAC=0.050

ATHENA> LINE X LOC=1.100 SPAC=0.150

ATHENA> LINE X LOC=1.500 SPAC=0.150

ATHENA> LINE X LOC=1.800 SPAC=0.050

ATHENA> LINE X LOC=2.000 SPAC=0.100
ATHENA> LINE X LOC=2.100 SPAC=0.100

ATHENA> LINE X LOC=2.300 SPAC=0.100

ATHENA> LINE X LOC=2.600 SPAC=0.100 TAG=RIGHT

ATHENA> LINE Y LOC=0.00 SPAC=0.03 TAG=TOP

ATHENA> LINE Y LOC=0.20 SPAC=0.02

ATHENA> LINE Y LOC=1.00 SPAC=0.10 TAG=BOTTOM
```

Using MaskViews for Generating Masks in Athena

The dry etching capability of Athena and the physical etching capability of Athena/Elite can be used in conjunction with the mask generating capability provided by DeckBuild and MaskViews. A cutline loaded into DeckBuild has information on the x-location of the photomask edges. You should specify the sequence of mask creation and stripping steps in the Athena input file. This can be done by selecting the **Commands→Process→Photo→Mask** option from DeckBuild when Athena is active, which will open the **DeckBuild: Athena Mask** menu. (Figure 2-55).

When you select a name (e.g., POLY) from the **Edit layer** list, press the **Apply Mask** button and the following lines will appear in the input file.

```
# DEFINING POLY MASK

MASK NAME="POLY"
```



Label: POLY

Name:

Bias: 0.00

Misalign: 0.00

Dela CD: -0.20

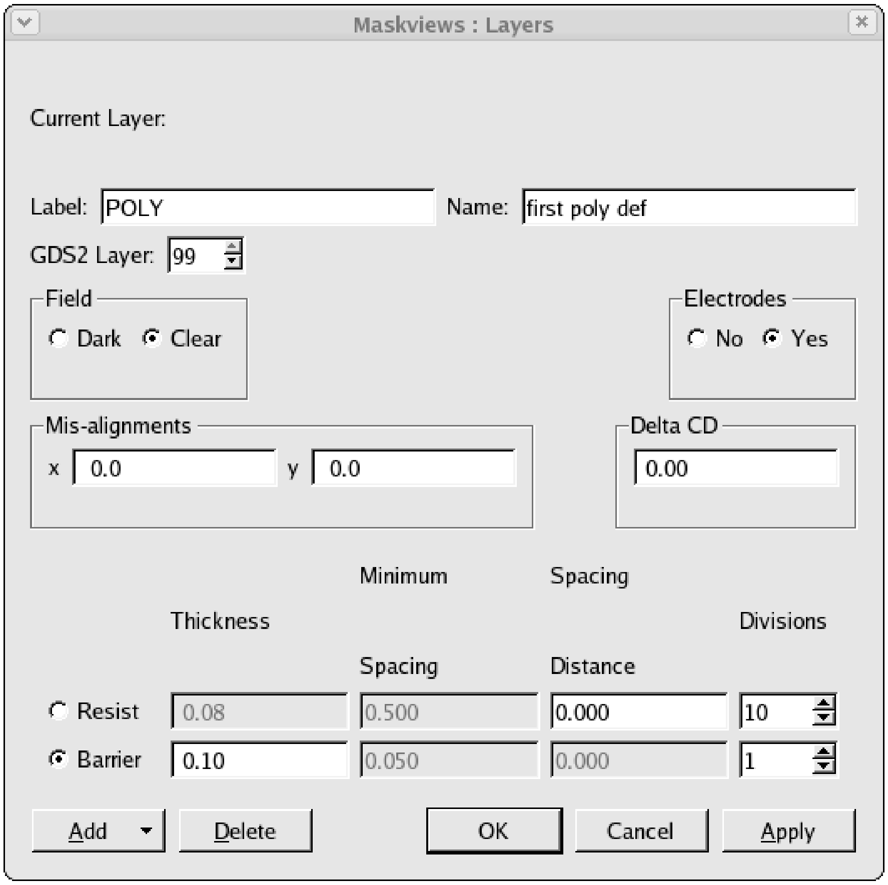
Shrink: 0.00

Comment: Defining POLY mask

Figure 2-55 Athena Mask Popup

Other parameters of the **DeckBuild: ATHENA Mask** menu (**Bias**, **Misalign**, **Delta CD**, and **Shrink**) allow you to modify or experiment with mask size or position or both (see Section 4.9 in the DeckBuild User's Manual)

During runtime, DeckBuild converts the `MASK` statement into a `DEPOSIT` statement, followed by a series of `ETCH` statements. The mask thickness and material type are defined in the MaskViews Layers Popup (Figure 2-56) in the **Define** menu of MaskViews.



Current Layer:

Label: POLY Name: first poly def

GDS2 Layer: 99

Field: ○ Dark ● Clear

Electrodes: ○ No ● Yes

Mis-alignments: x 0.0 y 0.0

Delta CD: 0.00

	Thickness	Minimum Spacing	Spacing Distance	Divisions
○ Resist	0.08	0.500	0.000	10
● Barrier	0.10	0.050	0.000	1

Figure 2-56 MaskViews Layers Menu

Two types of mask material are available: Photoresist and Barrier. The real thickness of a photoresist layer should be specified because it can be used as a mask for implantation. Barrier is a fictitious material. It is impenetrable for any implants and can serve only as a masking material. This material is implemented in Athena for the purpose of simplifying simulation of mask deposition over highly non-flat structures. Therefore, it is recommended to use barrier thickness of 0.1 microns or less in most cases. A region to be etched may be any area not containing a mask on a clear field layer or any area containing a mask on a dark field area. You can specify this in the **Field** section in the MaskViews Layers Popup. In the case of the `POLY` mask and cutline in Figure 2-56, the barrier layer will be etched to the left of x=0.8, and to the right to x=1.8. The following echo output will appear in the DeckBuild Text Subwindow as the result of defining of the `POLY` mask:

```
ATHENA> # DEFINING POLY MASK

ATHENA> ## MASK NAME="POLY"

ATHENA> DEPO BARRIER THICK=0.1 DIVIS=1

ATHENA> STRUCT OUTFILE=.HISTORY.9

ATHENA> ETCH BARRIER LEFT P1.X=0.8

ATHENA> ETCH BARRIER RIGHT P1.X=1.8

ATHENA> STRUCT OUTFILE=.HISTORY.11
```

If the **Reverse Mask** box is checked in the Athena Photo popup, the following lines will be inserted into the input file:

```
# DEFINING POLY MASK

MASK NAME="POLY" REVERSE
```

and the effect of the field attribute is reversed (i.e., the barrier area will be etched between x=0.8 and x=1.8).

When the mask is defined, the Athena dry etch capability can be used to etch the specified thickness of a material not covered by the mask. After the dry etch is complete, strip the mask by clicking the **Strip Mask** button in the Athena Mask popup (Figure 2-55). A typical mask definition fragment should appear as follows:

```
# POLY DEFINITION

MASK NAME="POLY"

ETCH POLY THICK=0.5

STRIP
```

If the cutline from Figure 2-54 is loaded, this will give the structure shown in upper plot of Figure 2-57. If the reverse parameter is added, the structure will appear as shown in the lower plot of Figure 2-57.

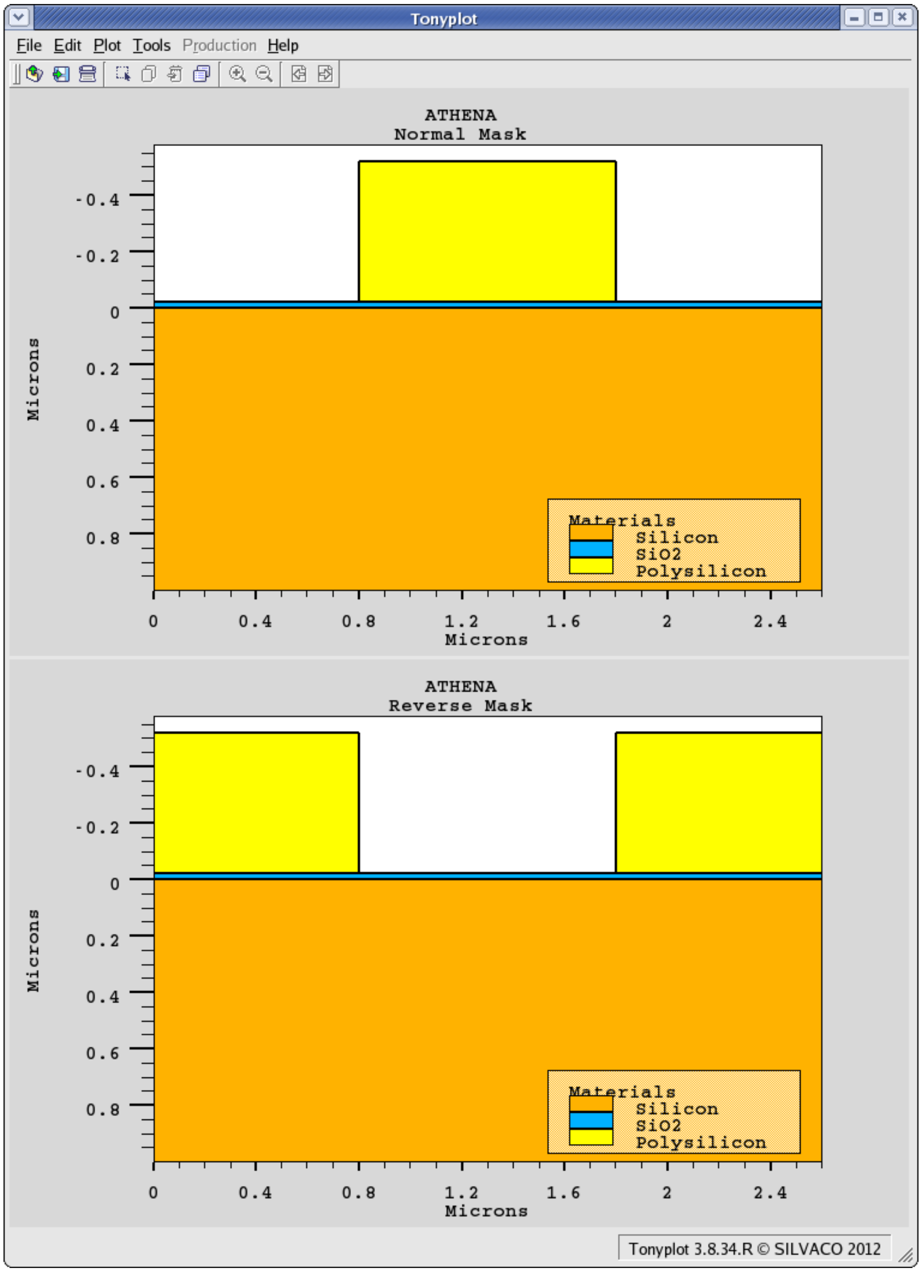


Figure 2-57 Using Mask Capability for POLY Definition

2.9 Using Athena/Optolith

2.9.1 Overview

Athena/OPTOLITH is designed as an optical lithography tool integrated into a complete process framework. Specific functions of Athena/OPTOLITH include 2D aerial image formation, 2D photoresist exposure and development, post exposure bake, and post processing capabilities such as CD extraction for generating SMILE plots. This section of the tutorial describes Athena/OPTOLITH input/output and the following basic operations for creating a typical input file for optical lithography:

- Creating an input mask using MaskViews or the `LAYOUT` command
- Designing custom or standard illumination systems
- Projection Fourier plane filtering
- Imaging controls
- Properties of materials
- Structure exposure, post exposure bake, and development
- CD extraction, SMILE plots, and looping procedures

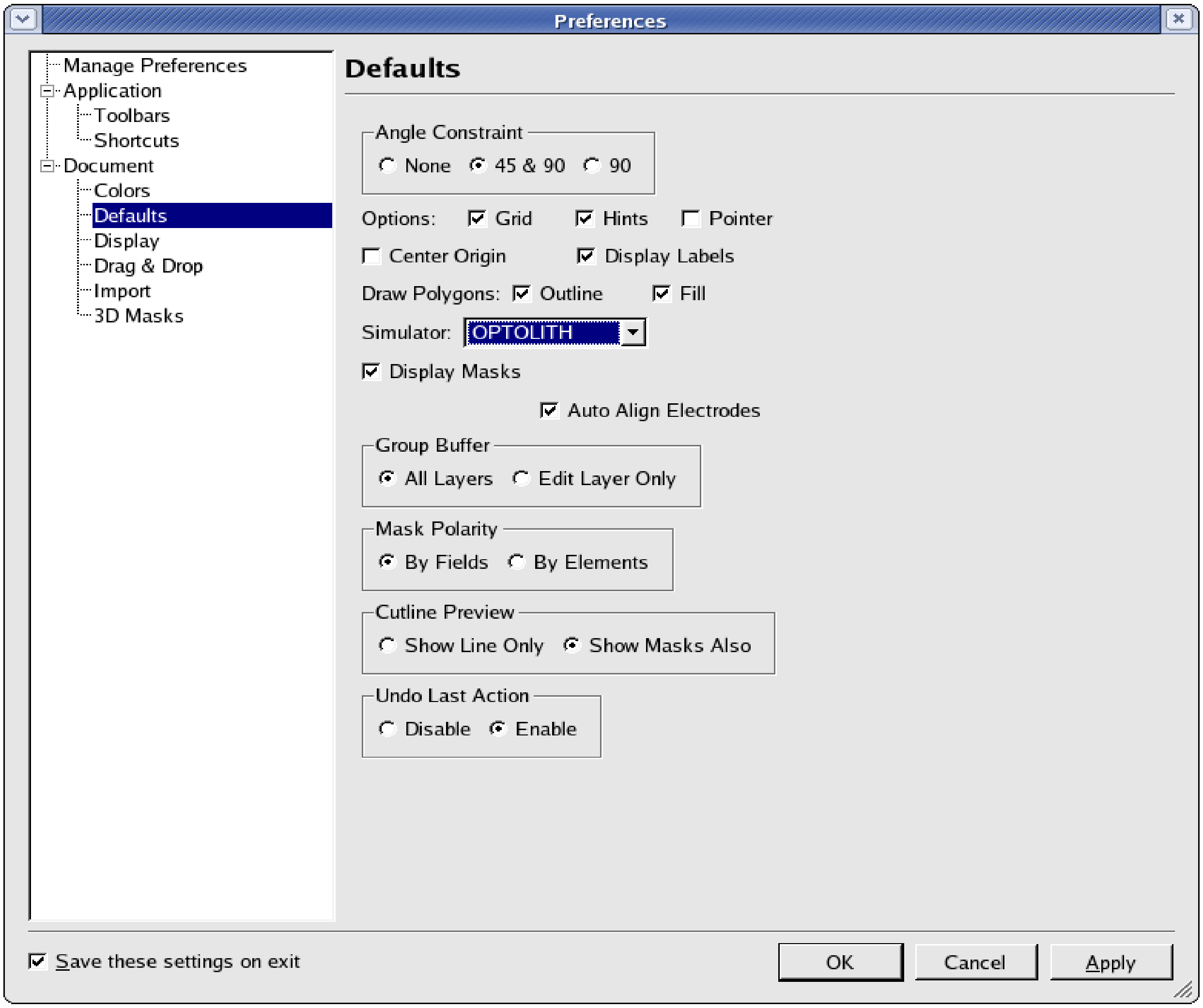
This section of the tutorial assumes that you are familiar with the general operation of Athena. This includes familiarity with the command language used to generate structures, as well as a general knowledge of the use of the Interactive tools. Specific features that refer particularly to OPTOLITH will be explained here.

2.9.2 Creating A Mask

A mask can be created using the MaskViews tool or by using the `LAYOUT` command. MaskViews facilitates the creation of complicated masks and can import different mask data formats, such as the GDS2 stream format. In the case of simple masks containing one or two features, it may be simpler to use the `LAYOUT` command.

MaskViews

Once you select **MaskViews** from the **Tools/MaskViews** menu, press the **Start MaskViews** button and the MaskViews window will appear. Then, press the **Preferences** button and the MaskViews **Properties** popup will appear (See Figure 2-58).



Defaults

Angle Constraint: ○ None ● 45 & 90 ○ 90

Options: ☑ Grid ☑ Hints □ Pointer

□ Center Origin ☑ Display Labels

Draw Polygons: ☑ Outline ☑ Fill

Simulator: OPTOLITH

☑ Display Masks

☑ Auto Align Electrodes

Group Buffer: ● All Layers ○ Edit Layer Only

Mask Polarity: ● By Fields ○ By Elements

Cutline Preview: ○ Show Line Only ● Show Masks Also

Undo Last Action: ○ Disable ● Enable

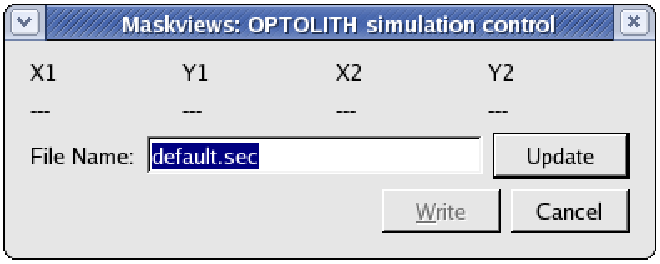
☑ Save these settings on exit

Figure 2-58 MaskViews Preferences Popup showing the Simulation Menu

Open the **Simulator** menu and select the **ATHENA/OPTOLITH** menu item. Customized controls for MaskViews/OPTOLITH will appear in the MaskViews window. The colored buttons on the right side of the window are discrete controls for phase in degrees and intensity transmittance. The buttons first appear as phase. To change to transmittance, open the **Phases** menu above the buttons and select the **Transmittances** menu item. This will change the buttons from phase to transmittance controls. Continuous controls for phase and transmittance are located directly below the colored buttons.

The mask can now be designed using the mouse driven line writer following the description outlined in the MaskViews User's Manual. Once the mask is created, it should be saved to a file with a name ending in a **`.lay`** extension for future editing. It is important to be aware that there are two types of files that can be saved from a MaskViews layout information.

The first type is the layout file. This file includes the information about layers and mask features. To store this information, select the **Files Save** menu item in the MaskViews screen. The second type of file that can be saved from MaskViews is a file that is similar to the layout file but is written to interface with Athena/OPTOLITH. The file to be used by OPTOLITH is created by pressing the **Write File** button in the MaskViews window. The Optolith Simulation Popup will appear (Figure 2-59).



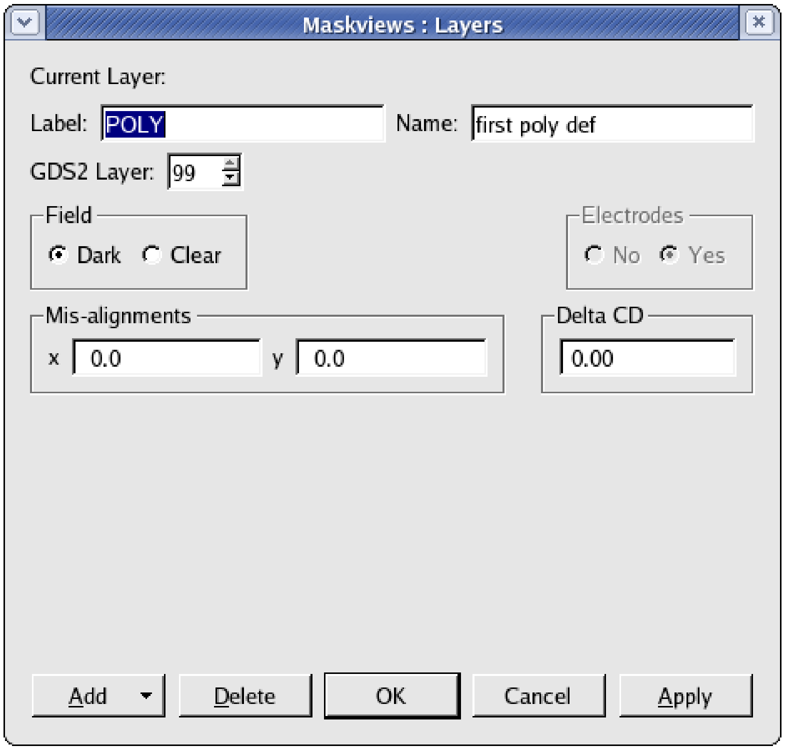
X1	Y1	X2	Y2
---	---	---	---

File Name: default.sec

Figure 2-59 OPTOLITH Simulation Control Popup

Enter the desired file name, which should end with a `.sec` extension, and proceed to the next step. Note that at the bottom of the MaskViews window the message: **Select first corner of OPTOLITH simulation area** will appear. MaskViews is now prepared for the selection of the image window. The image window describes the area where intensity will be calculated. Click on the desired area for intensity calculation to create the first corner of the OPTOLITH simulation area. The message: **Select the other corner of OPTOLITH simulation area** will then appear at the bottom of the MaskViews window. Click on the desired second corner. Once this second point is selected, the coordinates of the image window's lower left and upper right corners will be displayed in the OPTOLITH Simulation Control Popup. Press the **Write** button to save the OPTOLITH mask file. The input file created by MaskViews is loaded into OPTOLITH by the `IMAGE` command, which is described in Section 2.9.5 “Imaging Control”.

To modify the layers, open the **Define** menu in the MaskViews Window (See Figure 2-48) and select **Layers** menu item. The Layers Popup (Figure 2-60) will then appear.



Maskviews : Layers

Current Layer:

Label: POLY

Name: first poly def

GDS2 Layer: 99

Field: ● Dark ○ Clear

Electrodes: ○ No ● Yes

Mis-alignments: x 0.0 y 0.0

Delta CD: 0.00

Figure 2-60 Layers Popup

If you select **Dark**, the field background will be dark and the features will have the intensity transmittance (`T`), where `T` is user-defined. If you select **Clear**, the intensity transmittance automatically becomes `[1 - T]`.

Only rectangular features are used in the imaging module. MaskViews automatically converts triangles or polygons to a set of parallel rectangles. Finer resolution on these rectangles can be obtained by changing the resolution on the **Screen...** popup under the **Define** menu.

Mask Layout

In the `LAYOUT` command, each mask feature is defined with one command line. For example:

```
LAYOUT X.LO=-0.5 Z.LO=-5.0 X.HI=0.5 Z.HI=5.0 TRANS=1 PHASE=0
```

defines a 1*μ* wide line that is 10μ long. The mask has an intensity transmittance of one and a phase of 0°. The `LAYOUT` command can be repeated as often as desired. The number of mask features is limited only to the amount of memory available. The `LAYOUT LAY.CLEAR` command will remove all previous mask features from memory. Overlapping mask features will cause an error. The `OPAQUE` and `CLEAR` parameters can be specified in the `IMAGE` command. This will not reverse polarity as it does in MaskViews.

2.9.3 Illumination System

The Illumination System is defined using two statements: ILLUMINATION and ILLUM.FILTER. ILLUMINATION defines the illuminating wavelength, the possible x and z tilt of the optical system and the relative intensity, which is usually set to 1. ILLUM.FILTER defines the shape of the illumination system. The general shapes available are CIRCLE, SQUARE, GAUSSIAN, ANTIGUASS, and SHRINC. The extent of the source must be defined to be within a square centered at the origin as shown in Figure 2-61.

The extent of the source is defined by the coherence parameter, SIGMA. SIGMA defines the radius for circular sources (CIRCLE, GAUSSIAN, and ANTIGAUSS), the x and y intercepts for square sources, and the radius of each individual SHRINC source element as shown in Figure 2-62.

In all cases, anything outside of the square defined by SIGMA=1 will be ignored. The SHRINC source position is defined by the RADIUS and ANGLE parameters as shown in Figure 2-62. The SHRINC source can be defined by the command:

```
ILLUM.FILTER SHRINC RADIUS=0.25 ANGLE=45 SIGMA=0.1
```

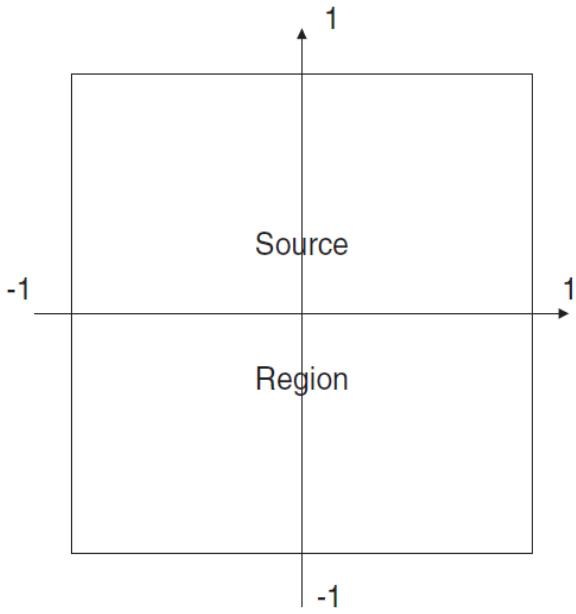


Figure 2-61 Maximum Extent of the Source Region

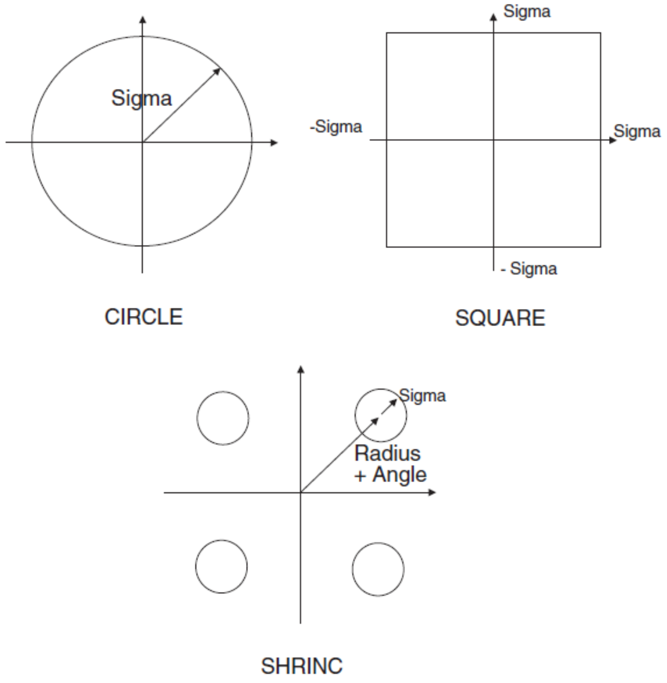


Figure 2-62 Three Different Source Types

Arbitrary sources can be defined by using the ANGLE and RADIUS parameters. Phase and intensity transmittance of each source element are controlled by the parameters PHASE and TRANSMIT. By positioning each source element in the source region, you can simulate any type of source. To simulate a SHRINC source, enter the following command lines:

```
ILLUM.FILTER CIRCLE SIGMA=0.1 RADIUS=0.25 ANGLE=45
ILLUM.FILTER CIRCLE SIGMA=0.1 RADIUS=0.25 ANGLE=135
ILLUM.FILTER CIRCLE SIGMA=0.1 RADIUS=0.25 ANGLE=225
ILLUM.FILTER CIRCLE SIGMA=0.1 RADIUS=0.25 ANGLE=315
```

If overlapping sources are defined, a warning is issued and the most recent source is used. If the overlap is partial, only the overlap area is overwritten by the most recent source.

Annular filters can also be superimposed on the source. There are two types of annular filters: Square and Circle. Annular filters have a multiplicative effect on the source. Because of this, be careful when defining a complex source and a complex filter. The following example of an annular source of inner radius 0.4 and outer radius 0.6 is given below.

```
ILLUM.FILTER CIRCLE SIGMA=0.6
ILLUM.FILTER CIRCLE INNER.RAD=0.0 OUTER.RAD=0.4 TRANSMIT=0.0
```

In the first statement, the SIGMA parameter defines the outer radius. In the second statement, an opaque spot is defined as an annular filter (see Figure 2-63).

The source must be described before the filter or the command will be ignored and a warning will be given. The CLEAR.FILTER parameter is used to remove all pre-existing filters and sources.

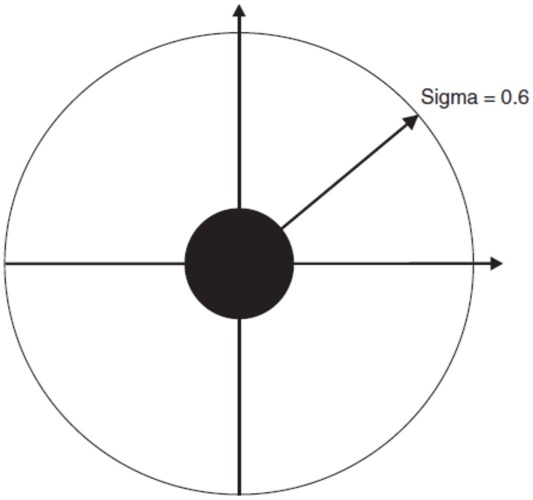


Figure 2-63 Annular Source

2.9.4 The Projection System

The Projection System is defined using two statements: PROJECTION and PUPIL.FILTER. The PROJECTION command is used to define the numerical aperture and flare of the projection system. The PUPIL.FILTER command describes the shape of the projection system and the possible filters of the projection system. The shape of the projector pupil can be square or circular. The circular pupil has the option of having a Gaussian or anti-Gaussian transmittance profile. Filtering of the Fourier spectrum can be performed by using annular filters. The filters have a multiplicative effect on the transmittance and phase in the projector pupil. The following example creates an opaque square at the origin:

```
PUPIL.FILTER SQUARE

PUPIL.FILTER SQUARE INNER.RAD=0.0 OUTER.RAD=0.1 TRANS=0.0
```

This creates the following projection pupil (Figure 2-64):

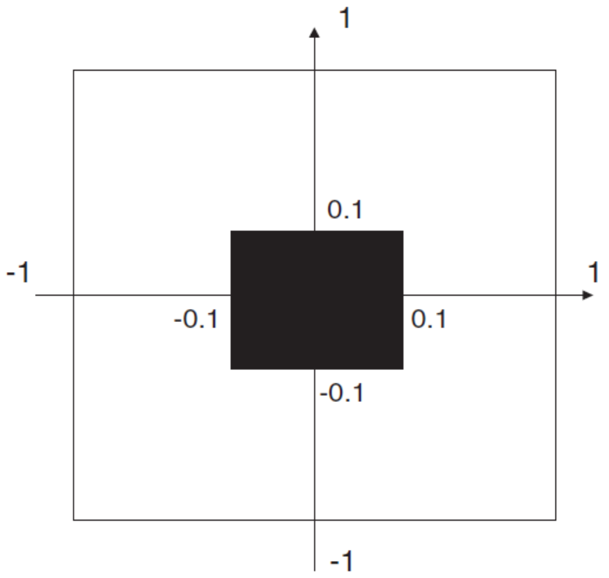


Figure 2-64 Projection Pupil

The maximum extent of the projector pupil plane is +1 or -1 in both dimensions. A filter exceeding these dimensions will be ignored and a warning will be issued.

2.9.5 Imaging Control

The image calculation is done by the IMAGE command and its associated parameters. The mask to be imaged will already be defined either by a MaskViews *.sec file or by the LAYOUT command. If a MaskViews *.sec file is used, the IMAGE command will be of this form:

```
IMAGE INFILE=*.sec ...
```

If the mask is defined using the LAYOUT command, the mask features will be stored in memory and the only required input related to mask features is the OPAQUE/CLEAR specification. OPAQUE specifies the background intensity transmittance to be zero. CLEAR specifies the background intensity transmittance to be one. OPAQUE is the default setting. OPAQUE and CLEAR cannot be used with an input file from MaskViews.

The Image Window (not the Computational Window) is specified with the parameters: WIN.X.LOW, WIN.Z.LOW, WIN.X.HI, and WIN.Z.HI. These parameters define the minimum and maximum range of the x and z values (see Figure 2-65). The aerial image is then calculated only inside this window. This allows for faster computation when you only want a cross section. If you want a simple cross section, set the window parameters for z (WIN.Z.LO, WIN.Z.HI) to the same value for a cross section parallel to the x-axis.

This value (WIN.Z.LOW = WIN.Z.HI) gives the location of the cross-section.

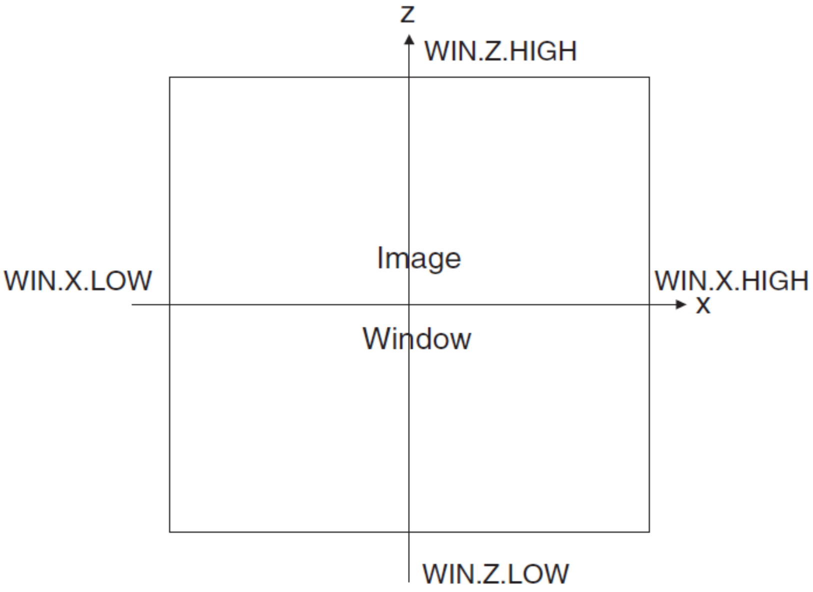


Figure 2-65 The Image Window can be Placed Anywhere in the XZ Window

The resolution in the image window can be controlled by two different sets of parameters. The first set is DX and DZ. DX and DZ are the resolution in micrometers for the x and z directions. The second set, X.POINTS and Z.POINTS, is based on the number of points in each direction. The resolution will be the length of the side of the image window divided by the number of points. To study the defocus of the aerial image, use the DEFOCUS parameter. DEFOCUS uses units of micrometers.

The N.PUPIL parameter specifies the computational window. If N.PUPIL is not specified, it is automatically calculated to a size that encompasses all mask features. In these cases, you can set the computational window manually using the following formula:

$$N.PUPIL = \frac{length \cdot NA}{lambda} * 2 \qquad 2\text{-}1$$

where *length* is the intercept coordinate for the x and z axes of a square centered at the origin that delimits the Computational Window as shown in Figure 2-66.

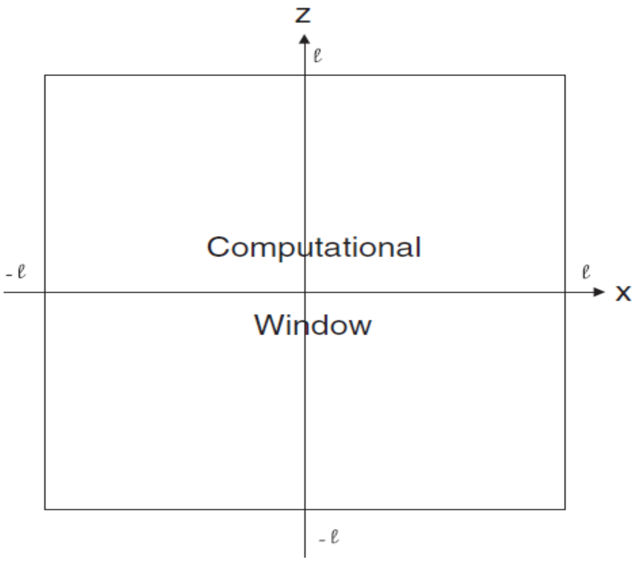


Figure 2-66 The Computational Window is Always Centered at the Origin

Once you calculate the image, you can store it in a standard structure file by using the STRUCTURE OUTFILE=*.STR INTENSITY command.

The INTENSITY modifier identifies the file to be different than a standard structure. This file can later be initialized into memory and used without running the imaging module. To initialize an intensity file, type:

```
INITIALIZE INFILE=*.STR INTENSITY
```

The intensity modifier again specifies the type of file. An intensity file initialized in this fashion is useful only for exposures that use the vertical propagation model.

N.PUPIL also affects the accuracy of the aerial image calculation. A higher N.PUPIL value increases the number of source points by a factor (2*N.PUPIL+1) squared, and increases the accuracy and the computation time.

Note: The Image Window and the Computational Window are not linked. The computational window is automatically adjusted to include all mask features, unless otherwise specified in the IMAGE command. This means that the entire mask will be used in the image calculation. You can use the Image Window to calculate a part of the entire image to increase the simulation speed. You can override the selected image in the IMAGE command by specifying new window coordinates.

Aerial image intensity distributions can be added together by specifying MULT.IMAGE on repetitions of the IMAGE command. You can add any number of images together. The first IMAGE statement shouldn't contain the boolean parameter, MULT.IMAGE, because the preceding aerial images are erased from memory. You can weigh of the aerial images by using the INTENSITY parameter on the ILLUMINATION command.

ONE.DIM is a new parameter that has been added to the IMAGE command. It allows you to calculate one dimensional aerial images. This is used primarily for increasing speed in the exposure calculation for repetitive line width calculations.

2.9.6 Defining Material Properties

There are two statements in Athena/OPTOLITH that relate to material properties: OPTICAL and RATE.DEVELOP. The OPTICAL command sets the complex index of refraction for a single material at a given wavelength. The RATE.DEVELOP command sets development rate parameters for each resist defined in the resist library. Default values for these material parameters are located in the *athenamod* file, which can be viewed in DeckBuild by selecting the **Models...** item from the Command menu. You can change any of these parameters by entering the command with the new values. For example, to change the index of refraction of silicon at the wavelength of 0.365 µm, enter:

```
OPTICAL SILICON LAMBDA=0.365 REFRAC.REAL=6.522 REFRAC.IMAG=2.705
```

To enter resist parameters at wavelength of 0.407µm, use the RATE.DEVELOP command as follows:

```
RATE.DEVELOP NAME.RESIST=AZ1350J LAMBDA=0.407 \
   A.DILL=0.88 B.DILL=0.077 C.DILL=0.018 E1.DILL=5.63 \
   E2.DILL=7.43 E3.DILL=-12.6
```

Photoresist parameters for development or diffusivity (DIX.0, DIX.E) can be entered separately from exposure parameters without specifying the wavelength. The photoresist name must always be specified. When specifying Dill exposure parameters (A, B, and C), specify the wavelength as these parameters vary with wavelength.

2.9.7 Structure Exposure

Exposure, post exposure bake, and development each have separate statements (EXPOSE, BAKE, and DEVELOP) respectively. In order to use these three statements, some initial requirements must be met. First, an intensity cross section or Fourier Spectrum data must be available. Second, you must create a structure including photoresist using the techniques described in Section 2.3 “Creating a Device Structure Using Athena”.

This intensity cross section can come from three different places. The first method is by running the imaging module prior to exposure. This puts the intensity data array into memory. The second method is by initializing with an intensity data array that has been stored in a standard structure file (see Section 2.9.5 “Imaging Control”) using the following command:

```
INITIALIZE INFILE=*.STR INTENSITY
```

The INTENSITY qualifier lets Athena know that this is an intensity file, as opposed to a standard structure file. After this command is entered, the intensity data array will be placed in memory. Wavelength will be stored in this file and can be changed only by re-running the imaging module.

The third method of entering an intensity cross section is through a user data file. The file should contain the wavelength, the number of data points, and the intensity and position of each point. The first line of this file should contain the wavelength in micrometers. The second line should contain the number of points. The following line should contain the position and then the intensity of the first point on the same line. This should be repeated for each point. This input file is read in the EXPOSE command using the following format:

```
EXPOSE INFILE=*.EXP
```

Once the intensity array is initialized or when the Fourier spectrum data is in memory through the IMAGE command, you can expose a structure if it exists in memory and if it has photoresist as its top layer(s). You can either create the structure in the input file or initialize it as described in Section 2.3 “Creating a Device Structure Using Athena”.

The EXPOSE command has many parameters that control the accuracy and speed of the exposure simulation, as well as related imaging parameters. The following parameters control simulation speed and accuracy and are unnecessary for a preliminary simulation.

```
FLATNESS, NUM.REFL, FRONT.REFL, BACK.REFL, ALL.MATS
```

The most important of these parameters is the FLATNESS parameter. If FLATNESS is set equal to zero, the algorithm uses the entire grid for the calculation and may lengthen the simulation time. The remaining parameters refer to the image to be exposed. Both TE and TM modes are available in exposure, but they must be performed separately. Select TE by adding the PERPENDICULAR parameter to the EXPOSE command, or select TM by entering the PARALLEL parameter. TE is the default. The exposure dose is also defined in the EXPOSURE command in units of mJ/cm^2 using the DOSE parameter.

Exposures can be made with either coherent or incoherent sources. Coherent sources are described by SIGMA=0.01 in the IMAGE command. This defines a small enough source so that only one discretization point is included. If a large SIGMA is defined and discretization of the source allows at least three source points in the x (or z) direction, then three points from the source will be used in the bulk image calculation with equal weight given to each point. The points chosen will be the central point and the outermost points, or the dimension of the chosen cross section (x or z). If multiple sources are defined using the ILLUM.FILTER command, then the central point of each SOURCE defined is used for calculating the bulk image in the exposure. The latter allows an arbitrary amount of source points to be simulated for the bulk image calculation. This is done by specifying many small adjacent sources and one point will be taken from each source.

You can add bulk image exposures together by specifying MULT.EXPOSE on repetitions of the EXPOSE command. Any number of exposures can be added together. The first EXPOSE statement should not contain the boolean parameter MULT.EXPOSE because the preceding exposures are erased from memory. You can weigh the exposures by using the DOSE parameter on the EXPOSE command.

The final four parameters: X.CROSS, Z.CROSS, CROSS.VALUE, and X.ORIGIN all refer to the aerial image cross section. The boolean parameters (X.CROSS or Z.CROSS) define the cross-section to be parallel to the x-axis or the z-axis respectively. CROSS.VALUE specifies the z location of X.CROSS, or the x location of Z.CROSS. These parameters are especially useful when several cross sections from the one large, two dimensional aerial image will be simulated. X.CROSS is the default. If CROSS.VALUE is unspecified, the center of the image window, defined in the imaging module, will be used as CROSS.VALUE. X.ORIGIN allows the aerial image cross section to be shifted laterally in the two dimensional exposure simulation.

Post Exposure Bake

The BAKE command can be invoked by using only one parameter DIFF.LENGTH (the diffusion length). For a post exposure bake of 60 seconds at a temperature of 125°C, the recommended diffusion length is between 0.05 and 0.1 micrometers. The BAKE command can also be used by specifying TIME and TEMPERATURE in C°.

Development

The development module offers a choice of six different development models. As mentioned before, model parameters are specified in the RATE.DEVELOP command. After the development model is selected, the three primary parameters for the DEVELOP command are: TIME (in seconds), STEPS, and SUBSTEPS.

TIME is the total development time. STEPS specifies the number of times the structure has to be regridded. SUBSTEPS is the total number of times the development line has to be moved. Each substep is performed for a time increment equal to TIME/(STEPS*SUBSTEPS). After each regridding of the structure, you can dump out a standard structure file to show the progress of the development. To do this, specify the parameter DUMP=1. To name the structure file to be dumped, specify DUMP.PREFIX=<name> and the structure will be created in the local directory with the name <name>**.str, where ** is the current development time.

Post Development Bake

A physically-based reflow of the developed photoresist is available. Specify it by using the BAKE command and the boolean parameter REFLOW along with TIME and TEMPERATURE.

2.10 Adaptive Meshing

2.10.1 Introduction to Mesh Adaption

Athena has a built-in mesh adaption module that automatically adapts the grid to dopant profiles. Used together with implantation and diffusion, the module can achieve more optimized accuracy of a given profile's representation for a given number of grid points. This relieves you to some extent from the time consuming mesh generation task in the simulation structure preparation stage. It will also improve the accuracy and speed of the subsequent diffusion/oxidation/epitaxy stages where impurity profiles change with time.

The algorithm used was suggested in [3], [4]. It uses an efficient local error estimator and a triangulation scheme suitable for complex two-dimensional moving boundary problems.

Adaption During Ion Implantation

Ion implant is a common process step to introduce impurities into the substrate to form active device regions. Prior to the implant step, it is difficult to determine the required mesh density distribution because the exact dopant profile is unknown before processing. Thus, you can only estimate the profile and required mesh. It's a time consuming process to specify mesh generation statements to create the mesh with a density conforming to an estimated profile. Graphical tools, such as Devedit, can make this easier. But it can't totally eliminate the process. With the Adaptive Meshing module, you can overcome these difficulties to a larger extent.

The program uses an iterative algorithm to determine the required mesh density distribution to accurately conform to the implanted profile, and will automatically generate the additional required mesh. The algorithm is illustrated with the flow chart depicted in Figure 2-67.

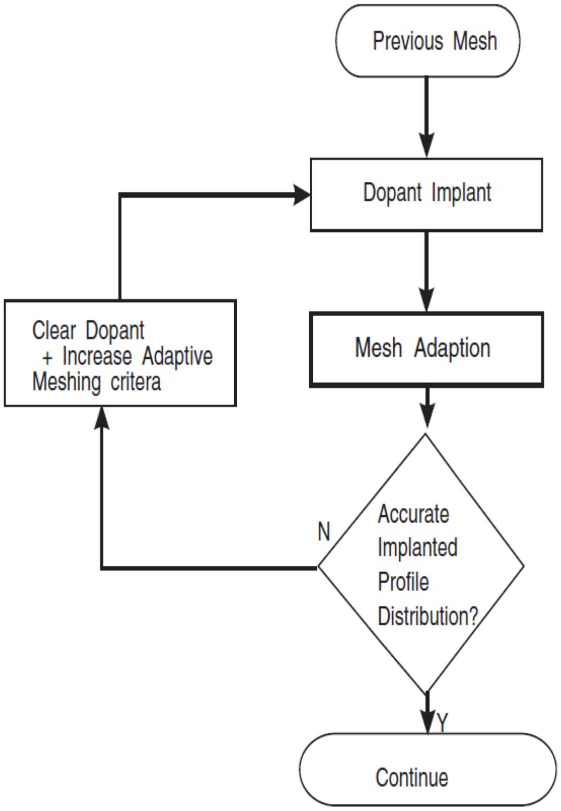


Figure 2-67 Flow Chart of Mesh Adaption Algorithm

Adaption During A Heat Cycle

During the diffusion/oxidation/epitaxy processes, impurity profiles are usually changing continually with each elapsed time step. An initially generated optimal mesh will not conform to the time varying dopant profile. If the impurity profiles change substantially during the process, the mesh density distribution will be different from the dopant contour distribution, causing both accuracy and speed problems.

During simulation, the total time of a diffusion/oxidation/epitaxy process is usually divided into many small time steps, with profiles changing gradually between time steps. Using the Adaptive Meshing module, you can perform a mesh adaption after each time step. This allows the mesh to conform to the dopant after a time step.

The difference between the dopant contours and the change in the mesh density distribution will thus be limited to the difference of dopant profiles between time steps. This difference is substantially smaller than that over total diffusion time. Thus, mesh adaption can allow more accuracy and minimize the mesh density for the dopant representation at any given time.

A Simple Example:

```
GO ATHENA
LINE X LOC=0.00 SPAC=0.1
LINE X LOC=2.00 SPAC=0.1
LINE X LOC=0.00 SPAC=0.1
INIT SILICON C. ARSENIC=10E14
DIFF TIME=50 TEMP=950 DRYO2
DEPOSIT POLY LEFT PL.X=1.2
ETCH POLY LEFT PL.X=1.2
STRUCT OUTF=MOS_0.STR
#PERFORM ADAPTIVE MESHING FOR BOTH IMPLANT AND DIFFUSION
METHOD ADAPT
IMPLANT BORON DOSE=1.0E13 ENGERY=15 PERSON TILT=0
STRUCT OUTF=MOS_1.STR
DEPOSIT OXIDE THICK=.35 DIV=6
ETCH OXIDE THICK=.35
IMPLANT BORON DOSE=1.0E14 ENGERY=15 PERSON TILT=0
STRUCT OUTF=MOS_2.STR
DIFFUSE TEMP=1000 TIME=30
STRUCT OUTF=MOS_3.STR
QUIT
```

LISTING 1: A SIMPLE EXAMPLE OF IMPLANT ADAPTIVE MESHING

This simple example creates a LDD MOS device structure. The initial simple mesh is specified with the four LINE statements. This initial mesh is referred to as a base mesh and options for its formation will be discussed the “The Mechanics of the Base Mesh Formation” on 120. After some initial 1D processing, the adaptive meshing function is invoked. Subsequently, automatically adds mesh that conforms well to the two implanted Boron profiles. During the final DIFFUSE statement, Boron has been driven down into the substrate and tessellated with the initial simple mesh. The mesh adaptation module adapts after each time step. This results in meshing conforming to the Boron profile throughout the diffusion process.

The mesh adaption module is invoked during the simulation by specifying boolean flag ADAPT on the METHOD command preceding IMPLANT, DIFFUSE, or EPITAXY statements. The syntax behind this simple example using the mesh adaption module is shown below.

Three statements are available to access the mesh adaptation module, they are briefly described as the following. The METHOD statement is used to control numerical algorithms. When METHOD ADAPT is specified, the mesh adaptation algorithm will be used. If you specify METHOD ADAPT=false, the mesh adaptation algorithm will be turned off. ADAPT is off by default.

ADAPT specifies that the adaptive meshing should be performed on any of the following IMPLANT, DIFFUSE or EPITAXY statements. Adaptation is performed by following each step on each DIFFUSE/EXPITAXY statement. IMPLANT.MES specifies which adapting algorithm to use on IMPLANT statements. Currently, IMPLANT.MES=0 corresponds to the University of Florida's algorithm. This is the default. Also currently, this is the only recommended algorithm. There are four other parameters on the METHOD statement that specify mesh smoothing.

They are as follows:

- ETCH.SMOOTH specifies that mesh smooth operation will be performed after etch.
- DEPO.SMOOTH specifies that mesh smooth operation will be performed after deposit.
- DIFF.SMOOTH specifies that mesh smooth operation will be performed after diffusion.
- STEP.SMOOTH specifies that mesh smooth operation will be performed after each diffusion time step. These four parameters are currently set as default.

The ADAPT.PAR statement is used to set parameters to adjust the mesh adaptation process. The parameters available on the ADAPT.PAR statement are the following. Specify material regions to be adapted, such as SILICON, OXDIDE, and POLYSILICON. This may be one or several materials at a time. The default impurities include such as I.BORON or I.ARSENIC. Specify impurities to be adapted on. This may be one or several impurities at a time. The DISABLE parameter specifies materials/impurities given disabled to be effective on mesh adapting or smoothing. The MAX.ERR parameter specifies the maximum errors allowed before adding points to the mesh (unitless). Errors calculated above this value cause points to be added. The MIN.ERR parameter specifies the minimum error below which points may be deleted from the mesh (unitless). Error calculated below this value causes points to be removed. Both MAX.ERR and MIN.ERR are calculated using the Bank-Weiser Error Estimator which is defined as

$$c = h^2 * \frac{\Delta(C_i)}{(C_i)} \qquad 2\text{-}2$$

where h is the average of the edge lengths associated with node i. C_i is the impurity concentration at node i. The parameter, CONC.MIN, specifies the minimum impurity concentration below which adapting will stop (units 1.0/cm^3).

The ADAPT.MESH statement is used to do mesh adaptation for a given device structure without coupling implant diffusion/epitaxy to the process. Therefore, the mesh adaptation module can be used to assist the manual mesh generation process. The following parameters are available on the ADAPT.MESH statement:

- ADAPT specifies a stand alone adaptive meshing step should be performed to refine or relax the current mesh based on the material/impurity specification given on the ADAPT.PAR statement (default false).
- ADAPT.COUNT specifies that stand alone annealing be performed during the execution of the ADAPT.MESH statement (default false).
- SMTH.COUNT specifies the number of smooth loops during the smoothing algorithm.

Adaptive Meshing Control

Adaptive meshing may be used in several different modes and has several statements to control it. This section describes the adaptive meshing related statements and how to use them. Table 2-6 list these statements.

Table 2-6 Summary of Adaptive Meshing Control

Parameter	Description
METHOD	Switches the various automated adaption modes on and off.
ADAPT.MESH	Invokes a stand alone adaption of the mesh at a specific point.
ADAPT.PAR	Control both the stand alone adaption and the automatic adaption meshing criteria.
GRID.MODEL	Describes an external template file, containing mesh related statements specific to a general technology or device.
BASE.MESH	Defines the 1D starting point of a mesh for an adaptive mesh based simulation.
BASE.PAR	Specifies the adaption criteria for the base mesh only.

The Mechanics of the Base Mesh Formation

Athena uses adaptive meshing in both 1D and 2D modes. ADAPT.PAR parameters control both these modes. The concept of the Base Mesh, however, needs to be described. A typical simulation (e.g., a MOS) is simulated in 1D initially and then switched to 2D during mid-process flow, perhaps at the Poly Gate definition process step. Here, the mesh is extruded from 1D to 2D and the result is the base mesh. The Base Mesh then forms the basis, and is the starting point, for 2D Adaptive Meshing. The mesh quality of this base mesh is important for success of future adaption for example, during source-drain implants and anneals.

2.10.2 Interface Mesh Control

The Interface Mesh Control is used to control the mesh in the vicinity of a material interface. This function allows you to add grid lines, for example, to run along under the gate of MOSFET at some distance from the Si/Si02 interface. The Interface Mesh Control is often useful for adding mesh as required by highly mesh dependent mobility models during a following device simulation. It is also useful to be able to add mesh for better segregation modeling. The ADAPT ADD.I.LINE=n command controls the addition of a new mesh line. Two materials are specified as parameters to the command defining an interface or a set of interfaces. The mesh line is added to MATERIAL1 as follows:

```
ADAPT.MESH ADD.I.LINE=0.001 MATERIAL1 / MATERIAL2
```

For example, in the case of adding an additional mesh line to the SILICON in the channel region of a MOSFET:

```
ADAPT.MESH ADD.I.LINE=0.001 SILICON / OXIDE
```

The structural transition from 1D to 2D to create a base mesh is controlled by the BASE.PAR parameters. Figure 2-68 indicates the flow of events towards the formation of a base mesh and beyond in the case of MOSFET device.

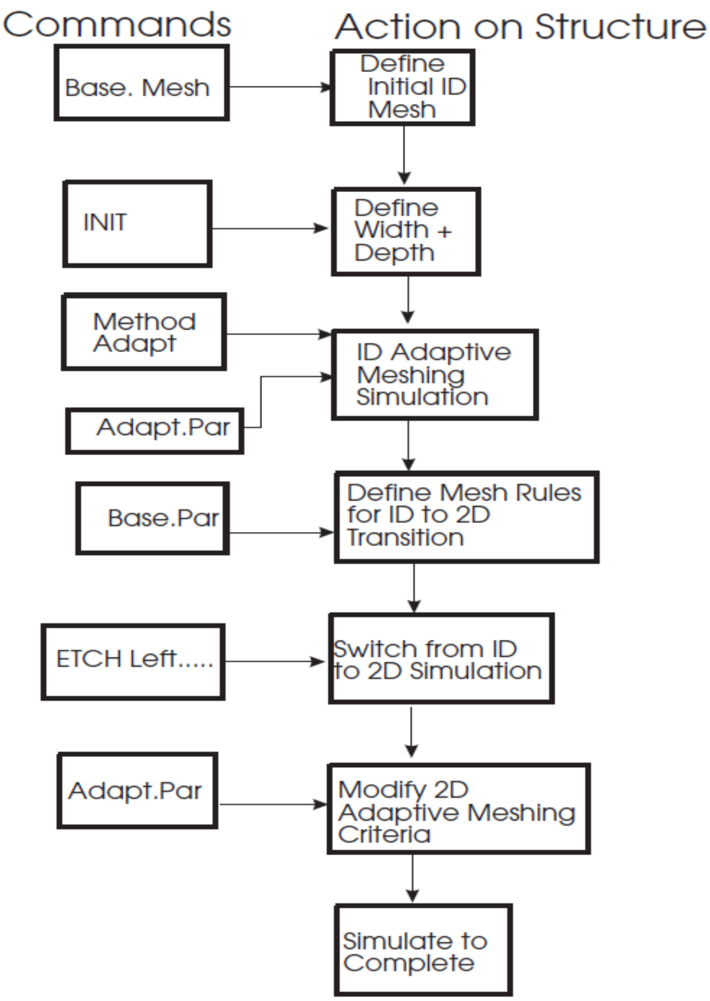


Figure 2-68 MOSFET Device Mesh Formation Flow

The base mesh quality is important to allow a subsequent adaption in **2D**. The adjacent ratio of elements, both in **1D** and **2D** relate directly to the smoothness of the final mesh quality.

The generation of a high quality adapted mesh starts with the `BASE.MESH` command. Here, the 1D mesh is defined from where the final 2D mesh will evolve. The `BASE.MESH` command defines a 1D structure as a stack of up to five layers. Five layers are used to define the five layers of a Bipolar device. Each layer is described as having a thickness: `SURF.LY`, `ACTIVE.LY` `EPI.LY`, `SUB.LY`, and `BACK.LY` and an associated mesh spacing per layer, `SURF.DY`, `ACTIVE.DY`, `EPI.DY`, `SUB.DY`, and `BACK.DY`. The whole structure can also be offset in space with the point of origin determining the top left hand corner of the structure. The `OFFSET.X` and `OFFSET.Y` parameters are used for this purpose.

An example of using the `offset` command might be defining the starting surface of an initial structure an `epi` thickness below the zero position. That way, the subsequent geometrical calculations are made easier.

Figure 2-69 indicates the relationship of the `BASE.MESH` command to the initial 1D structure mesh.

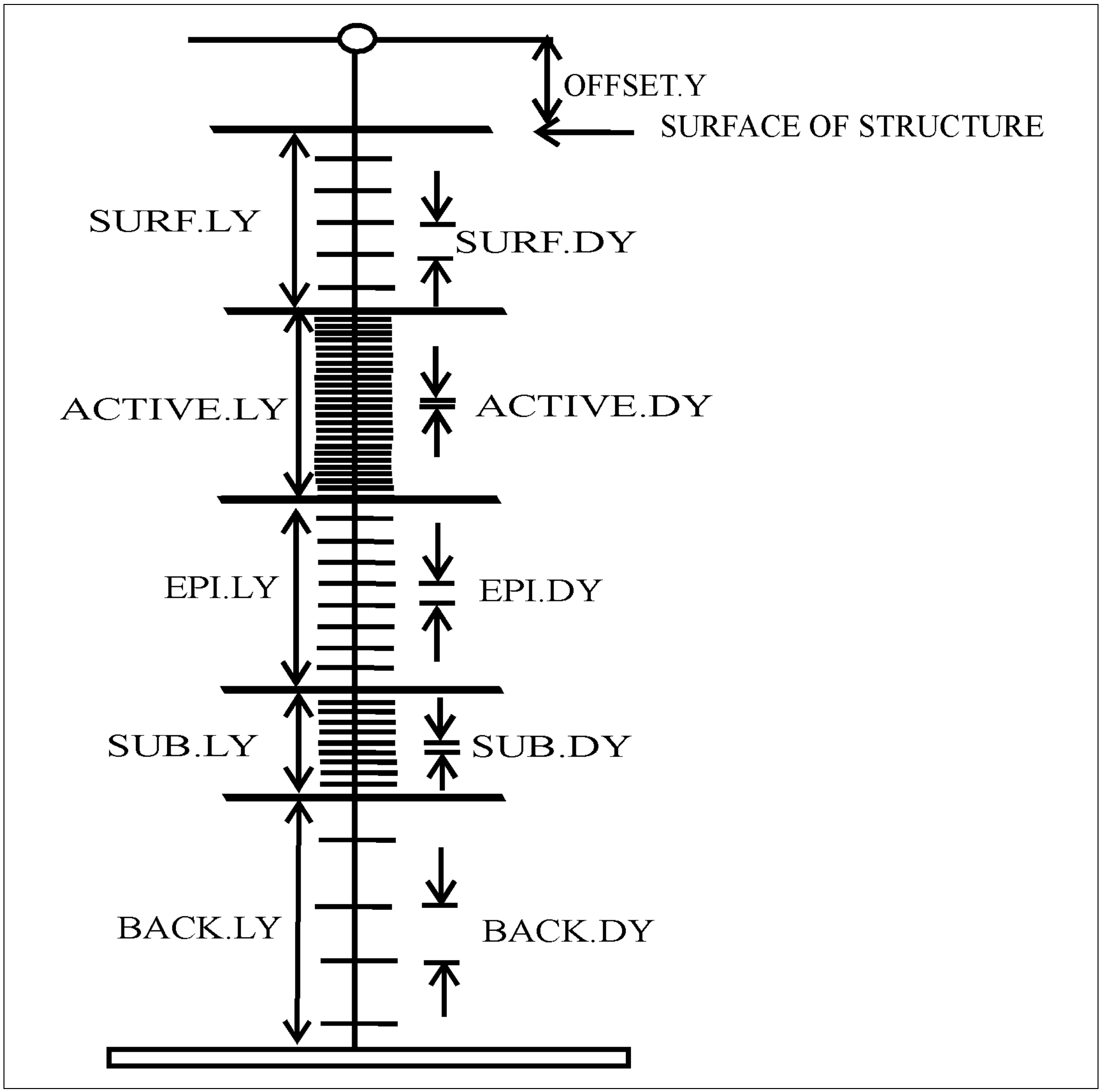


Figure 2-69 Initial 1D Structure Relationship

The `BASE.MESH` parameters should be considered alongside the `BASE.PAR` parameters. When forming a base mesh, there are three objectives to remember regarding the quality of mesh. These objectives are as follows:

- 1D dopant information is neither lost in the 2D transition nor overly refined upon, resulting in overly dense `BASE.MESH` (See Figures 2-70 and 2-71).
- Little or few flat triangles exist in regions and materials of importance. (See Figure 2-74).
- The adjacent triangle ratio, in both X, and Y directions, is not abrupt in spacial regions of importance to the device (See Figure 2-75).

Controlling the quality of the BASE.MESH, formed at the 1D-2D transition, is achieved with the BASE.PAR command parameter. Specific materials can be assigned different parameters. The GRAD.SPACE parameter controls the Vertical Adjacent Triangle Ratio Quality, while the RATIO.BOX parameter controls the lateral Adjacent Triangle Ratio. These two statements can be thought of as operating upon the 1D and 2D simulation segments respectively during 1D simulation. Only the adjacent spacing ratio can be controlled in the vertical profile with the GRAD.SPACE parameter. Subsequently, at the point of 2D transition, the RATIO.BOX parameter is used to trade off mesh quality for mesh density.

The INIT command includes the parameters: WIDTH.STR and DEPTH.STR. These parameters define the size of the initial structure and will truncate the previous BASE.MESH defined structure.

Figures 2-74 and 2-75 show an example of this base mesh and of the subsequent 2D diffusion.

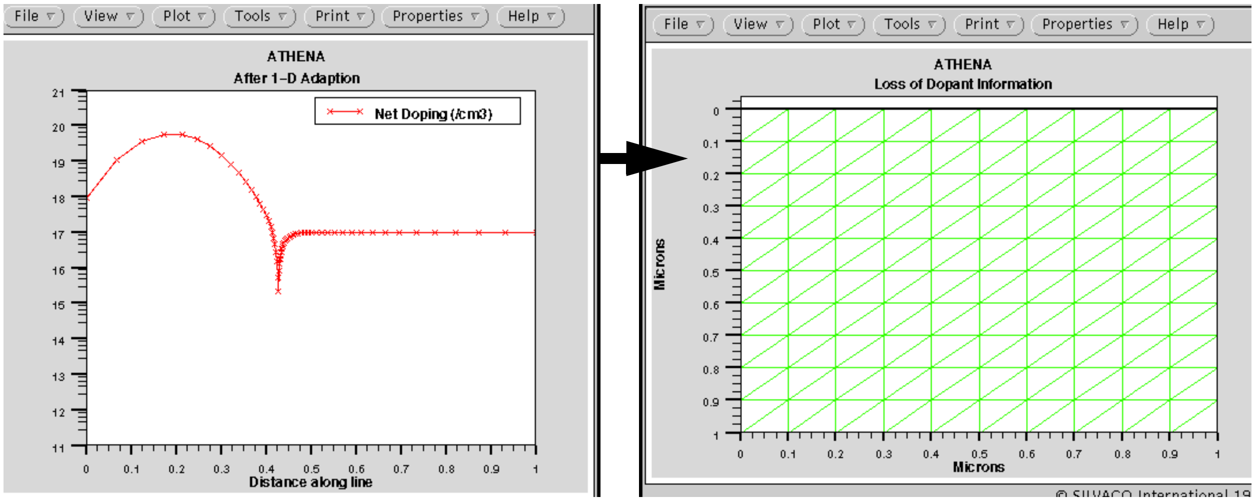


Figure 2-70 Mesh that is too coarse leads to Dopant Information Loss

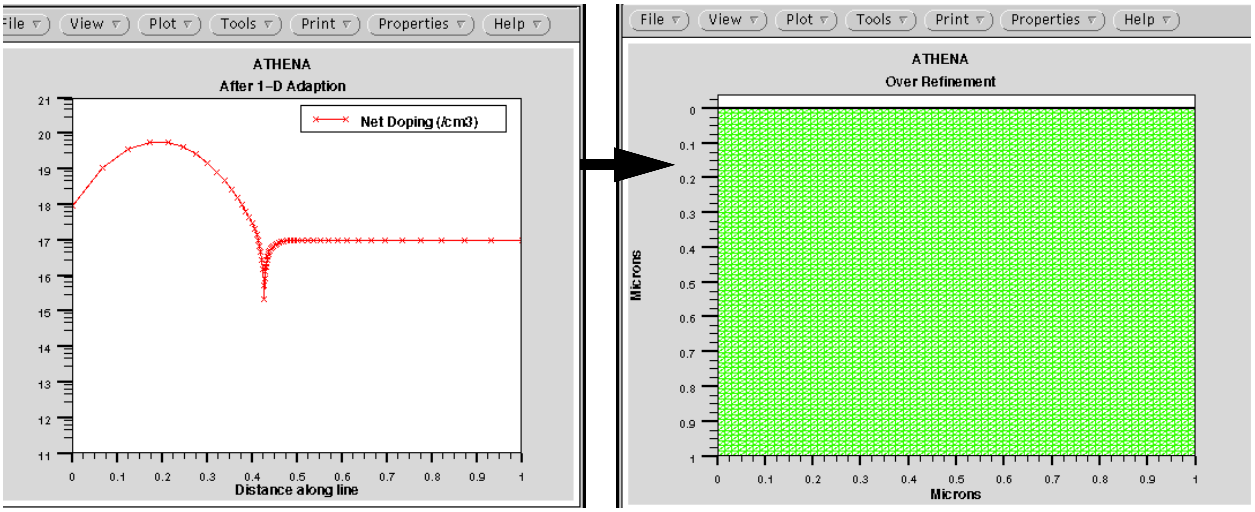


Figure 2-71 Too Dense Mesh Causes Too Much CPU Time during Subsequent Simulation

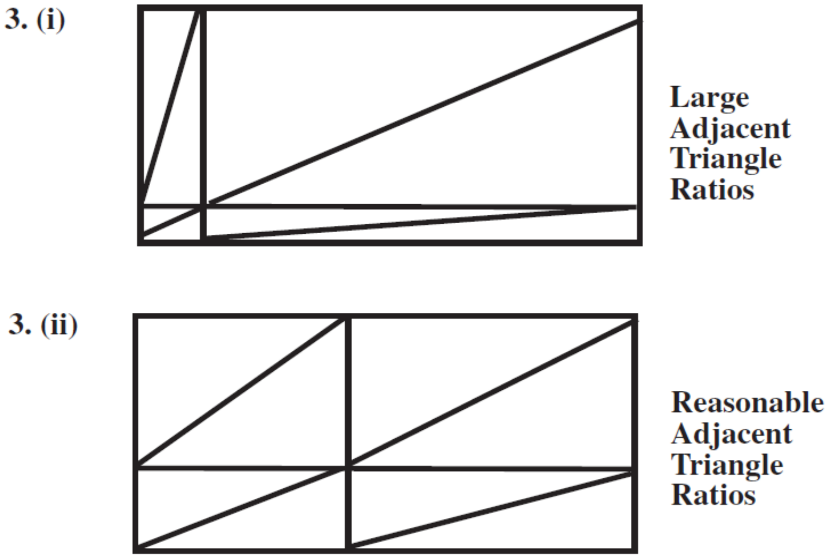


Figure 2-72 Large and reasonable Adjacent Triangle Ratios

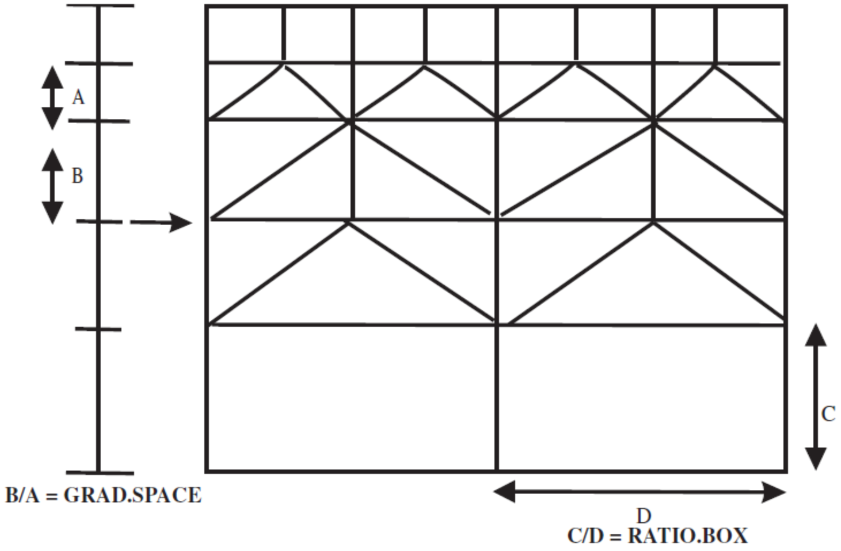


Figure 2-73 Base Mesh Formation

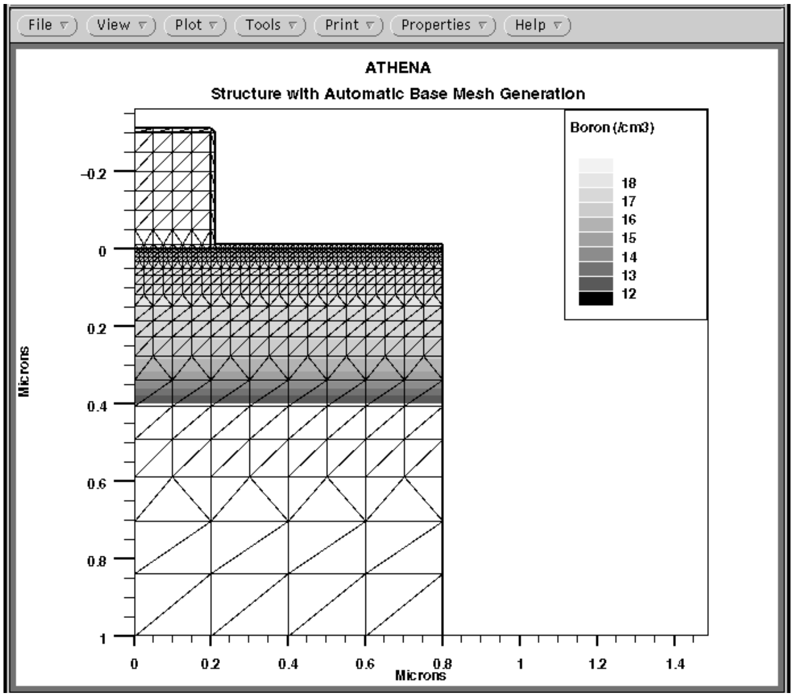


Figure 2-74 Automatic Base Mesh Generation

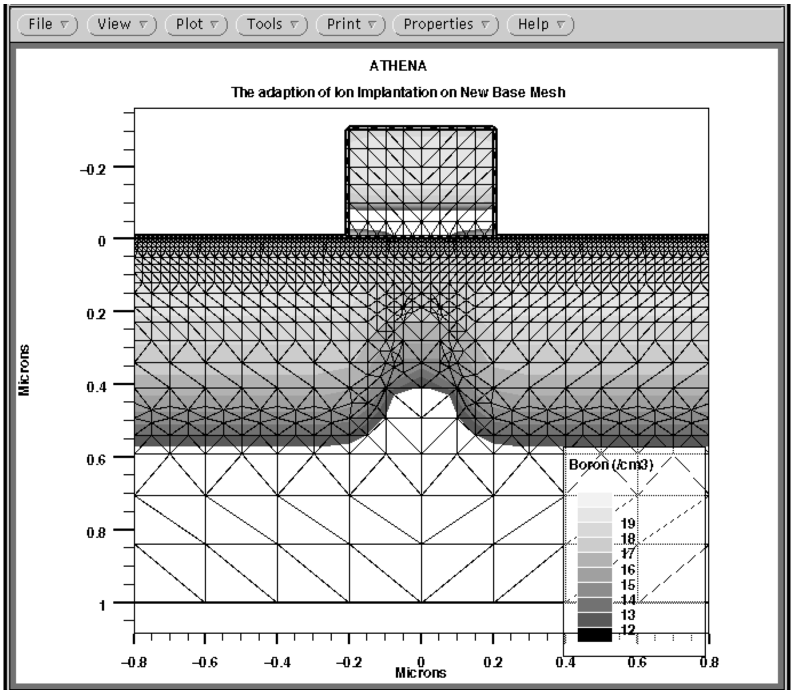


Figure 2-75 Ion Implantation Adaption on New Base Mesh

Chapter 3
SSuprem4 Models

3.1 Diffusion Models

The diffusion models in Athena describe how implanted profiles of dopants/defects (see the Note below) redistribute themselves during thermal treatment, due to concentration gradients and internal electric fields. When modelling the actual diffusion process, there are additional effects to consider such as impurity clustering, activation, and how interfaces are treated. Fundamentals of the models described in this section could be found in [5], [6], and [7].

Note: In the following sections, the terms impurity and dopant shall be used interchangeably, although an impurity doesn't necessarily have to be a dopant. Also the term, defect, shall mean the same as point defect, unless otherwise indicated in the context.

Diffusion of dopants and point defects in SSuprem4 is described by a number of user-specifiable models. The three most basic models are the following:

- The Fermi diffusion model.
- The two dimensional diffusion model.
- The fully coupled diffusion model.

The models are natural extensions of each other in the sense that the Fermi model is included in the two dimensional model, which is included in the fully coupled model. The two significant differences between them are the way point defects are represented and treated throughout the simulation, and how the specific dopant diffusivities are formulated. The selection of which model to use will depend upon the existence or the generation of point defects during the diffusion process and the dopant concentrations within the silicon. Careful reading of the following sections is critical to understanding which model to use.

All three models rely on the concept of Pair Diffusion, which says that a dopant atom cannot diffuse on its own – it needs the assistance of a point defect (a silicon self interstitial or a lattice vacancy) in the near vicinity as a diffusion vehicle. If there is a non-vanishing binding energy between the two, they can move as one entity (a pair) through a number of jumps and inversion cycles before eventually breaking up. When speaking of dopant diffusivity within the scope of these models, one actually means the diffusivity of the pair as a whole. A point defect, however, can either diffuse freely or as a participant in a dopant-defect pair. The diffusivity of a free point defect can actually be different from the diffusivity of a point defect pair.

All diffusion models in Athena also use the concept of Chemical and Active Concentration Values. The chemical concentration is the actual implanted value of the dopant but when dopants are present at high concentrations, clustering or electrical deactivation can occur so that the electrically active concentration may be less than the corresponding chemical concentration. This is described in Section 3.1.6 “Electrical Deactivation and Clustering Models”.

Athena creates structures that can have multiple materials and interfaces such as the polysilicon-oxide-silicon interface in MOSFETs. Each interface within Athena has boundary or interface conditions that model impurity segregation. The model details are described later in this chapter. You should, however, be aware that the gas/solid interface (the surface of the silicon if exposed) and solid/solid interfaces have been strictly modelled within Athena. Effects such as dopant loss from exposed silicon and dopant pile-up at interfaces are simulated.

3.1.1 Mathematical Description

The mathematical definition of a diffusion model includes the following specifications for every diffusing species present:

- a Continuity Equation (often called a Diffusion Equation).
- one or more flux terms.
- a set of boundary and interregional interface conditions.

In the case of impurity diffusion in semiconductors, we need a set of equations for each dopant present and for each type of point defect if point defects are explicitly represented in the model. Since dopants can only diffuse as participants in dopant-defect pairs, the dopant continuity equation is actually a continuity equation for defect-dopant pairs.

The formulation of the continuity equation have a number of built-in assumptions:

- Electronic processes take place on a time scale, which is much smaller than the time scale of all other processes (adiabatic approximation).
- The pairing reaction between dopants and defects is assumed to always be in equilibrium. This may not be the case, especially at a low temperature, but would pose a much harder and more CPU intensive numerical problem to solve.
- Mobile dopants are electrically active and vice versa.

Models that explicitly take pair populations into account have been implemented by various research groups. But all these models suffer from the lack of established experimental data, such as binding energies, or pairing coefficients, for which reason the predictability of these models is questionable. The lack of data, especially for the energy levels of the different charge states of the point defects in the band gap at typical diffusion temperatures, poses a serious gap in our knowledge. Some of these energy levels have been measured in low temperature experiments such as DLTS (deep level transient spectroscopy), but no one knows how these levels adjust themselves relative to the band edges when the band gap narrows as a function of increasing temperature.

Van Vechten [8] has theoretically argued that the acceptor states (0/- and -/=) and the donor states (+/++ and +/0) of the mono vacancy follow the conduction band edge with increasing temperature. Mathiot [5], however, chooses to scale the positions of the energy levels relative to the band edges with the size of the band gap.

In addition to the models described above, which are all specific for dopants and point defects in silicon, there is a smaller number of hard coded models that are used for other materials such as oxide or poly.

In the sections that follow, we apply standard notation used in the literature for dopants, point defects (interstitials and vacancies) and the different charge states as shown in Table 3-1. In Table 3-1, the *x* designates the neutral charge state. - is a single negatively charged state. = is a double negatively charged state.

Table 3-1 Notational standards in diffusion literature

Physical Entity	Generic Symbol	Replacement Values
Dopant	A	B, P, As, Sb,...
Point Defect	X	I, V
Charge State	c	x, -, =, +, ++

Many physical entities or parameters are temperature-dependent. In Athena, this dependence upon temperature is modelled by the Arrhenius expression (unless otherwise indicated):

$$Q(T) = Q.0 exp\left(-\frac{Q.E}{kT}\right) \tag{3-1}$$

where:

- *Q.0* is the pre-exponential factor,
- *Q.E* is the activation energy,
- *k* is the Boltzmann constant,
- *T* is the absolute temperature.

Generic Diffusion Equation

All diffusion models, whether they are the Fermi, the two dimensional or the fully coupled model, follow the same generic mathematical form of a continuity equation. A continuity equation merely expresses particle conservation, that is, the rate of change with time of the number of particles in a unit volume must equal the number of particles that leave that volume through diffusion, plus the number of particles that are either created or annihilated in the volume due to various source and sink terms.

This basic continuity equation for the diffusion of some particle species (*C*) in a piece of semiconductor material is a simple Second Order Fick's Equation [9]:

$$\frac{\partial C_{Ch}}{\partial t} = -\nabla J_A + S \tag{3-2}$$

where C_{Ch} is the total particle (chemical) concentration, J_A is the flux of mobile particles, ?∇ is the gradient operator, and *S* accounts for all source and sink terms. The difference between the total (chemical) concentration and the actual mobile concentration is described in a later section entitled Section 3.1.6 "Electrical Deactivation and Clustering Models". In semiconductor diffusion problems, there are generally two contributors to the particle flux.

The first contributor is an Entropy Driven Term, which is proportional to the concentration gradient of mobile particles. The coefficient of proportionality, D_A, is called the diffusivity. The second contributor is a Drift Term, which is proportional to the local electric field. Notice that if there are several types of electrically charged species present, this term establishes a coupling between them, since all charged particles both contribute to and are influenced by the local electric field.

The Flux Term, J_A, can be written as:

$$J_A = -D_A(C)\,\nabla C_A + C_A \sigma E \qquad \text{3-3}$$

where C_A designates the mobile impurity concentration, σ is the mobility, and E is the electric field. It should also be observed that Equation 3-3 is non-linear, since both the diffusivity D_A and the electric field E in general depend on the concentration of all present species. In thermodynamical equilibrium, the Einstein relation relates mobility and diffusivity through the expression $D = \frac{kT}{q}\sigma$.

Substituting for σ in Equation 3-3 is writing the particle charge as a signed integer, Z_A, times the elementary charge, q, giving us this Flux Expression.

$$J_A = -D_A(C)\left(\nabla C_A - Z_A C_A \frac{qE}{kT}\right) \qquad \text{3-4}$$

In insulator and conductor materials, the electric field is zero. In semiconductor materials, the electric field is given by:

$$E = -\nabla\psi = -\frac{kT}{qn}\nabla n \qquad \text{3-5}$$

where ψ is the electrostatic potential and n is the electron concentration. If charge neutrality is assumed, then the electron concentration may be rewritten as:

$$n = \frac{N_D - N_A}{2} + \sqrt{\left(\frac{N_D - N_A}{2}\right)^2 + n_i^2} \qquad \text{3-6}$$

where N_D and N_A are the electrically active donor and acceptor impurity concentrations, and n_i is the intrinsic carrier concentration calculated as:

$$n_i = NI.0 \cdot exp\left(-\frac{NI.E}{kT}\right) T^{NI.POW} \qquad \text{3-7}$$

where you can specify the `NI.0`, `NI.E`, and `NI.POW` parameters in the `MATERIAL` statement. The electrically active and mobile impurity concentrations are equivalent.

Boundary conditions

Boundary conditions within Athena are of mixed type and are expressed mathematically as:

$$\alpha \cdot C_A + (\beta \cdot \partial_n C_A) = R \qquad \text{3-8}$$

where (α, β) are real numbers and $\partial_n C_A$ designates the flux of C_A across the boundary. The right hand term, R, accounts for all source terms on the boundary. Boundary conditions are applied at two main regions.

The first region is at the top of the simulation region (the surface). The second region is at the inter-regional interfaces for which the species in question only has a meaningful existence in one of the region materials (e.g., an interstitial on a silicon/oxide interface).

Interface conditions

Between any two regions there must be some control on how any impurity species can exist in the vicinity of the interface. For every such interface, you must specify a Concentration Jump Condition and a Flux Jump Condition.

The Concentration Jump Condition accounts for discontinuities in particle concentrations across interfaces and encompasses particle transport across material interfaces due to different solid solubility ratios of the impurity species in the two materials.

The Flux Jump Condition enables the formulation of interface source and sink terms such as surface recombination, particle injection, and particle pile-up at a moving interface.

For all species, no flux boundary conditions are employed on the sides and at the bottom of the simulation structure. This is hardwired into the software, which means it is not user-definable.

3.1.2 The Fermi Model

The Fermi Model assumes that point defect populations are in thermodynamical equilibrium and thus need no direct representation. All effects of the point defects on dopant diffusion are built into the pair diffusivities. The main advantage for using the Fermi Diffusion model is it will greatly improve the simulation speed, since it does not directly represent point defects and only needs to simulate the diffusion of dopants. Also, the Fermi Model usually results in an easier numerical problem due to the avoidance of “numerical stiffness”. But since point defects are not directly simulated, the Fermi model cannot deal with certain process conditions in which the defect populations are not in equilibrium, such as in wet oxidation (where Oxidation Enhanced Diffusion (OED) is important), emitter-base diffusions and wherever implantation results in an initial high level of implant damage.

To use the Fermi Model, specify `FERMI` parameter in the `METHOD` statement.

In the Fermi Model, each dopant obeys a continuity equation of the form:

$$\frac{\partial C_{Ch}}{\partial t} = \sum_{X = I, V} \nabla \bullet D_{AX}\left(\nabla C_A - Z_A C_A \frac{qE}{kT}\right) \tag{3-9}$$

where C_{Ch} is the chemical impurity concentration, Z_A is the particle charge (+1 for donors and -1 for acceptors), D_{AV} and D_{AI} are the joint contributions to the dopant diffusivity from dopant-vacancy and dopant-interstitial pairs in different charge states [5]. C_A is the mobile impurity concentration and E is the electric field. The terms D_{AV} and D_{AI} depend on both the position of the Fermi level as well as temperature and are expressed as:

$$D_{AX}\left(T,\frac{n}{n_i}\right) = D_{AX}^{X} + D_{AX}^{-}\left(\frac{n}{n_i}\right)^{1} + D_{AX}^{=}\left(\frac{n}{n_i}\right)^{2} + D_{AX}^{+}\left(\frac{n}{n_i}\right)^{-1} + D_{AX}^{++}\left(\frac{n}{n_i}\right)^{-2} \tag{3-10}$$

where the temperature dependency is embedded in the intrinsic pair diffusivities, which are specified by Arrhenius expressions of the type:

$$D_{AX}^{c} = D.0_{AX}^{c} exp\left(-\frac{D.E_{AX}^{c}}{kT}\right) \tag{3-11}$$

Table 3-2 shows the names of the pre-exponential factors, *D.0*, and activation energies, *D.E*, for each of the charge states, *c*, of the various intrinsic pair diffusivity terms.

Pair charge states beyond two are unlikely to occur, which is why they have been omitted. Also, for most dopants it is seldom that more than three of the terms above are non-vanishing.

Table 3-2 Table of intrinsic pair diffusivities for different pair types

Pair	**Charge State**	**Pre--exponential factor**	**Activation Energy**
AV	×	`DVX.0`	`DVX.E`
AV	-	`DVM.0`	`DVM.E`
AV	=	`DVMM.0`	`DVMM.E`
AV	+	`DVP.0`	`DVP.E`
AV	++	`DVPP.0`	`DVPP.E`
AI	×	`DIX.0`	`DIX.E`
AI	-	`DIM.0`	`DIM.E`
AI	=	`DIMM.0`	`DIMM.E`
AI	+	`DIP.0`	`DIP.E`
AI	++	`DIPP.0`	`DIPP.E`

Note: Since the point defect populations are by definition assumed to be in equilibrium in the Fermi model, there are no separate continuity or boundary condition equations for these species. Additionally, neither the vacancy concentration, C_I, nor the interstitial concentration, C_V, appear explicitly in Equations 3-9, 3-10, or 3-11.

3.1.3 Impurity Segregation Model

In multilayer structures, dopant segregation across material interfaces must be considered. Such interfaces can represent either a solid/solid interface or a gas/solid interface (the surface).

Interface segregation is modeled empirically by a first order kinetic model for the interregional flux:

$$F_S = h_{12} \cdot \left(\frac{C_1}{M_{12}} - C_2\right) \qquad 3\text{-}12$$

where:

- C_1 and C_2 are the particle concentrations in the immediate vicinity of the interface in the regions 1 and 2.
- h_{12} is the interface transport velocity.
- M_{12} is the segregation coefficient.

The transport velocity and segregation coefficients are temperature-dependent parameters defined through the following Arrhenius expressions:

$$h_{12} = TRN.0 \cdot exp\left(-\frac{TRN.E}{kT}\right) \qquad 3\text{-}13$$

$$M_{12} = SEG.0 \cdot exp\left(-\frac{SEG.E}{kT}\right) \qquad 3\text{-}14$$

You can specify the parameters: `TRN.0`, `TRN.E`, `SEG.0` and `SEG.E` in the `IMPURITY` statement. All parameters are specified for only one direction, which is from region 1 to region 2.

The following is an example of the syntax used to change the segregation coefficients between oxide and silicon. Two material names separated by a / to indicate the combination and the ordering of materials for which these parameters are specified.

```
IMPURITY I.PHOSPHORUS SILICON /OXIDE SEG.0=30 TRN.0=1.66E-7
```

Interface Trap Model (Dose Loss Model)

You can simulate the effect of dose loss at silicon/oxide interface by specifying the `DOSE.LOSS` parameter in the `METHOD` statement. This model is based on the theory that the dopant diffusing through silicon/oxide interface can be trapped into the trap sides located at the interface [10], [11]. A modified equation for impurity flux is used in the Dose Loss Model.

$$F_m = S_m(C_m(1 - C_T/C_{Tmax}) - M_m C_T) \qquad 3\text{-}15$$

where $m=1,2$ correspond to Si and SiO2, $M_m = C_{mss}/C_{Tmax}$, $M_{12} = M_1/M_2$, and $S_1 = 10S_2$. The C_T parameter is the real density of occupied trap sites at the interface and is found by solving the following equation:

$$\frac{\partial C_T}{\partial t} = F_1 + F_2 \qquad 3\text{-}16$$

where C_{Tmax} = 6.8 × 10^{14}cm^{-2} for phosphorus and 2×10^{14}cm^{-2} for other dopants. The dose loss transport coefficient S_1 is calculated through the following Arrhenius expression:

$$S_1 = TRNDL.0 \cdot exp\left(-\frac{TRNDL.E}{kT}\right) \qquad 3\text{-}17$$

where the `TRNDL.0` and `TRNDL.E` parameters can be specified in the `IMPURITY` statement.

3.1.4 The Two Dimensional Model

In this model, the point defect populations are directly represented and evolved in time. If there is a super/supra saturation of point defects, it will affect the dopant diffusivity through a simple scale factor, which goes to unity as the actual defect concentration approaches the equilibrium defect concentration. Therefore with equilibrium defect profiles, the Two Dimensional Model merely reduces to the Fermi Model, although in a more computational inefficient manner, since solving for point defects is not required. The pair coupling between defects and dopants in this model is assumed to be one-way. The diffusion of dopants is highly influenced by the diffusion of point defects, while the diffusion of the point defects, however, is regarded as totally independent of dopant diffusion. Stated in physical terms, this corresponds to a pairing between defects and dopants with zero binding energy.

To turn on the Two Dimensional Model, specify parameter `TWO.DIM` in the `METHOD` statement. The Two Dimensional Model is based on the Fermi Model, so read the Fermi Model description before continuing. The major difference between the Fermi Model and the Two Dimensional Model is the direct representation and evolution of non-equilibrium point defect populations. Therefore, there are three different sets of governing diffusion equations: one for dopants, one for point defect interstitials and one of point defect vacancies. In addition, you also need to take into account the {311} cluster formation and dissolution, bulk and interface recombination, and the generation of point defects through oxidation. Each of these are described in detail in the following sections.

Dopants

The continuity equation for dopants is:

$$\frac{\partial C_{Ch}}{\partial t} = -\sum_{X = I.\,V} \nabla \bullet J_{AX} \qquad 3\text{-}18$$

$$J_{AX} = -f_X D_{AX}\left[\nabla\left(C_A \frac{C_X}{C_X^*}\right) - Z_A\left(C_A \frac{C_X}{C_X^*}\right)\frac{qE}{kT}\right] \qquad 3\text{-}19$$

where C_I and C_V are the actual concentrations of interstitials and vacancies, and C^*_I and C^*_V are the equilibrium concentrations. The f_X factor is an empirical defect factor, which for interstitials is assumed to be temperature-dependent through the following Arrhenius expression:

$$f_I = FI.0\,exp\left(-\frac{FI.E}{kT}\right) \qquad 3\text{-}20$$

where the `FI.0` and `FI.E` parameters can be specified in the `IMPURITY` statement. The value of f_I is clamped to a number between 0 and 1. The equivalent term for vacancies is calculated according to:

$$f_V = (1 - f_I) \tag{3-21}$$

The formulation of equation (Equations 3-18 and 3-19) is similar to the Fermi Diffusion Model except for two elements. The additional term C_X / C_X^{*} has been added to model the enhancement or retardation of diffusion due to non-equilibrium point defect concentrations. The term f_X takes into account the knowledge that some impurities diffuse more by interstitials than by vacancies or vice versa. Although this dependency is of a phenomenological character, it seems reasonable, and is the one used by most diffusion simulators to account for the diffusion enhancement of dopants during oxidation enhanced diffusion (OED) or transient enhanced diffusion (TED).

Remember the point defect equilibrium concentrations are temperature as well as Fermi level dependent and can be calculated from the following expressions:

$$C_X^* = C_X^{*i} \frac{neu + neg\left(\frac{n}{n_i}\right)^{+1} + dneg\left(\frac{n}{n_i}\right)^{+2} + pos\left(\frac{n}{n_i}\right)^{-1} + dpos\left(\frac{n}{n_i}\right)^{-2}}{neu + neg + dneg + pos + dpos} \tag{3-22}$$

where C_X^{*i} represents the equilibrium defect concentrations of interstitials and vacancies under intrinsic conditions, and the weight factors: *neu*, *neg*, *dneg*, *pos*, and *dpos* account for the distribution of defects of different charge states under intrinsic conditions. All of these are assumed to be temperature dependent through Arrhenius expressions of the following type:

$$neu = NEU.0\, exp\left(-\frac{NEU.E}{kT}\right) \tag{3-23}$$

where the pre-exponential factors and activation energies, in this case `NEU.0` and `NEU.E`, can be specified in the `VACANCY` and `INTERSTITIAL` statements. Table 3-3 shows the complete set of corresponding parameters.

For dopants, the boundary and interface conditions are identical to the ones stated in the Fermi Model.

Table 3-3 Parameters for charge statistics and intrinsic point defect concentrations

Entity	Pre-exponential factor	Activation Energy
neu	NEU.0	NEU.E
neg	NEG.0	NEG.E
dneg	DNEG.0	DNEG.E

Table 3-3 Parameters for charge statistics and intrinsic point defect concentrations

Entity	Pre-exponential factor	Activation Energy
pos	`POS.0`	`POS.E`
dpos	`DPOS.0`	`DPOS.E`
C_X^{*i}	`CSTAR.0`	`CSTAR.E`

Interstitials

The interstitial profile evolves with the following continuity equation.

$$\frac{\partial C_I}{\partial t} = -\nabla \bullet J_I - R_B + R_T - R_{\langle 311 \rangle} \qquad 3\text{-}24$$

where J_I is the flux of interstitials, R_B is the bulk recombination rate of interstitials, R_T accounts for the capture or emission of interstitials by traps, and $R_{\{311\}}$ is the recombination rate of {311} clusters. Each of these terms are described below.

The interstitial flux, J_I, is calculated according to [5] with:

$$-J_I = D_I C_I^* \nabla\left(\frac{C_I}{C_I^*}\right) \qquad 3\text{-}25$$

which correctly accounts for the effect of an electric field on the charged portion of the interstitials, taking the gradient of the normalized interstitial concentration, $\frac{C_I}{C_I^*}$. D_I is the diffusivity of free interstitials. Don't confuse it with the *pair* diffusivity D_{AI}, which was mentioned in Section 3.1.2 "The Fermi Model".

D_I is calculated once again with the following Arrhenius expression:

$$D_I = D.0 \exp\left(-\frac{D.E}{kT}\right) \qquad 3\text{-}26$$

where the pre-exponential factor and activation energy (`D.0` and `D.E`) can be set in the `INTERSTITIAL` statement.

The bulk recombination rate (R_B) is a simple reaction between vacancies and interstitials that assumes that any interstitial will recombine with any vacancy, regardless of their charged states. This assumption may overestimate the recombination rate. The equation is expressed as:

$$R_B = K_r(C_I C_V - C_I^* C_V^*) \qquad 3\text{-}27$$

where K_r is the bulk combination coefficient and specified as:

$$K_r = KR.0\,exp\left(-\frac{KR.E}{kT}\right) \tag{3-28}$$

where the parameters KR.0 and KR.E are user-definable in the INTERSTITIAL statement.

The interstitial trap rate (R_T) model was first introduced by Griffin [12] to explain some of the wide variety of diffusion coefficients extracted from different experimental conditions. The Trap Equation, which describes the evolution of the empty trap population in time, is:

$$R = \frac{\partial C_{ET}}{\partial t} = -K_T\left[C_{ET}C_I - \frac{e^*}{1-e^*}C_I^*\ (C_T - C_{ET})\right] \tag{3-29}$$

where:

- C_T is the total trap concentration.
- K_T is the trap capture rate.
- C_{ET} is the empty trap concentration.
- C_I is the interstitial concentration.
- C_I^* is the equilibrium interstitial concentration.
- $e*$ is the equilibrium empty trap to total trap ratio, $e^* ½\ C_{ET}^* / C_T$.

Both K_T and $e*$ are Arrhenius expressions that can be set in the TRAP statement with the total trap concentration, R_T, with the parameters shown in Table 3-4.

Table 3-4 Parameters for interstitial traps

Entity	Pre-exponential factor	Activation Energy
K_T	KTRAP.0	KTRAP.E
e^*	FRAC.0	FRAC.E
C_T	TOTAL	—

The trap equation is either derived from the simple reaction:

$$I + T \Leftrightarrow IT \tag{3-30}$$

or posed as a rate equation:

$$\frac{\partial C_{ET}}{\partial t} = -K_T C_I C_{ET} + K_r(C_T - C_{ET}) \tag{3-31}$$

where K_r is the trap emission rate. In equilibrium, the left hand side of Equation 3-31 must vanish, giving:

$$K_T C_I^{*} \; C_{ET*} \;=\; K_r\left(C_T - C_{ET}^{*}\right) \;\Leftrightarrow\; K_r = \; K_T \frac{C_I^{*}\left(C_{ET}^{*}/C_T\right)}{1-\left(C_{ET}^{*}/C_T\right)} = \; K_T C_I^{*} \; \frac{e*}{1-e*} \qquad \text{3-32}$$

Substituting this value for K_r into Equation 3-31 leads to the expression in Equation 3-32.

The recombination rate of {311} clusters ($R_{\{311\}}$) in Equation 3-24 accounts for the release rate of {311} interstitial clusters, which are small, rod-like defects that have been observed in Transmission Electron Microscopy (TEM) studies after medium to high dose implantation of impurities into silicon. Since a large fraction, if not all, of the excess interstitials after implantation are believed to exist in this form, the time scale for dissolution of {311} clusters plays an important role for the duration of any Transient Enhanced Diffusion (TED).

Think of these volume defects as small pockets of interstitials distributed throughout certain parts of the doped regions, which are released during annealing; thus, acting as bulk sources of point defects.

Note: Actually, {311} defects are believed to be created from excess free interstitials during the earliest part of the annealing cycle through a process called Ostwald ripening. But here, they're considered as existing immediately after the implantation.

The cluster release rate follows a simple exponential decay in time specified by:

$$R_{\langle 311\rangle} \equiv -\frac{\partial C_{\{311\}}}{\partial t} = f(x)\left(\frac{1}{\tau}\right) exp\left(-\frac{t}{\tau}\right) \qquad \text{3-33}$$

where $f(x)$ is the as-implanted profile of {311} clusters, and τ is an Arrhenius type temperature dependent time constant calculated from:

$$\tau \;=\; TAU.311.0\, exp\left(-\frac{TAU.311.E}{kT}\right) \qquad \text{3-34}$$

where the `TAU.311.0` and `TAU.311.E` parameters can be specified in the `CLUSTER` statement. The profile, *f(x)*, of the {311} clusters is created from a previous `IMPLANT` statement. For more information, see Section 3.5.5 "Ion Implantation Damage".

To activate this model, a previous `IMPLANT` statement has to introduce {311} clusters with the `CLUSTER.DAM` flag in the `METHOD` statement. For example:

```
METHOD CLUSTER.DAM

CLUSTER      BORON      MIN.CLUSTER=1.0E17      MAX.CLUSTER=1.0E19
CLUST.FAC=1.4\ TAU.311.0=8.33e-16  TAU.311.E=-3.6 SILICON

IMPLANT  BORON ENERGY=100 DOSE=1E15
```

Here, the `METHOD` statement switches the model on, and the `CLUSTER` (optional) statement decides the location and scaling of the {311} cluster profile. In this example, clusters are present in the regions of the substrate, where the chemical boron concentration is between 1.0e17 cm^{-3} and 1.0e19 cm^{-3}, which are scaled by a factor of 1.4 relative to the boron concentration.

Notice that the activation energy for `TAU.311.E` must be specified as negative, since the time constant decreases with rising temperature.

Interstitial Generation and Recombination at Interfaces

Interfaces present a moving boundary problem during a thermal oxidation. In this instance, there will be a recombination rate at the interface, which will vary as a function of the interface velocity. Also as a consequence of the silicon being consumed, there is a significant injection of interstitials into the substrate. Within Athena, this is modelled by an interstitial flux boundary condition, as described by Hu [13].

$$\partial_n C_I + K_s\left(C_I - C_I{}^*\right) = G_I \qquad \text{3-35}$$

where:

- $\partial_n C_I$ is the projection of the interstitial flux vector on an inward pointing unit vector normal to the boundary.
- K_S is the effective surface recombination rate for interstitials.
- G_I is the generation rate at the interface of interstitials during annealing in an oxidizing ambient.

In other words, Equation 3-35 shows that the number of interstitials generated on the surface, minus the number of interstitials that recombine there, must equal the number of interstitials that diffuse from the surface/interface into the substrate.

The effective surface recombination rate, K_S, depends on the motion of the interface during oxidation according to:

$$K_S = K_{SURF}\left[K_{RAT}\left(\frac{v_i}{v_{i-max}}\right)^{K_{POW}} + 1\right] \qquad \text{3-36}$$

where v_i is the velocity of the interface, $v_{i\text{-}max}$ is the maximum interface velocity, and the parameters K_{SURF}, K_{RAT}, and K_{POW} are calculated according to the following equations:

$$K_{SURF} = KSURF.0\,exp\left(-\frac{KSURF.E}{kT}\right) \qquad \text{3-37}$$

$$K_{RAT} = KRAT.0\,exp\left(-\frac{KRAT.E}{kT}\right) \qquad \text{3-38}$$

$$K_{POW} = KPOW.0\,exp\left(-\frac{KPOW.E}{kT}\right) \qquad \text{3-39}$$

where the pre-exponential factor and exponent terms can be defined in the `INTERSTITIAL` and `VACANCY` statements.

Surface recombination plays an important role in the relaxation of perturbed point defect profiles back to their equilibrium values, which cannot happen by bulk recombination alone.

The surface generation rate, G_I, controls the injection of point defects into the silicon during oxidation.

Two models have been implemented into Athena: the default model, `GROWTH.INJ` and `TIME.INJ`. The moving interface can inject point defects into silicon and polysilicon.

The `GROWTH.INJ` parameter in the `VACANCY` or `INTERSTITIAL` statement will activate or deactivate the growth dependent injection model. By default, this model is always turned on and is described mathematically by the following equation.

$$G_I = \theta \cdot VMOLE \cdot v_i \cdot \left(\frac{v_i}{v_{i-max}}\right)^{G_{POW}} \qquad \text{3-40}$$

where:

- θ is the fraction of silicon atoms consumed during growth that are injected into the bulk as self interstitials.
- `VMOLE` is the lattice density of the consumed material.
- G_{POW} is a power parameter.

The values θ and G_{POW} are calculated from the following equations:

$$\theta = THETA.0 exp\left(-\frac{THETA.E}{kT}\right) \qquad \text{3-41}$$

$$G_{POW} = GPOW.0 exp\left(-\frac{GPOW.E}{kT}\right) \qquad \text{3-42}$$

and the `THETA.0`, `THETA.E`, `GPOW.0`, `GPOW.E`, and `VMOLE` parameters can be specified in the `INTERSTITIAL` and `VACANCY` statements.

As a general rule, the ratio $\theta/(K_{RAT} \cdot K_{SURF})$ should be maintained reasonably constant during calibration. The entities, v_i and v_{i-max} have the same meaning as described for surface recombination. The maximum interface velocity, v_{i-max}, cannot be manipulated directly and will change only when oxidation characteristics change.

The `TIME.INJ` parameter in the `VACANCY` or `INTERSTITIAL` statement activates the time dependent injection model. It is defined as:

$$G_{IV} = A(t + t_0)^{T_{POW}} \qquad \text{3-43}$$

where t is the total diffusion time in seconds, and A, t_0 and T_{pow} are free parameters used for calibration purposes and are calculated according to the following equations:

$$A = A.0 exp\left(-\frac{A.E}{kT}\right) \qquad \text{3-44}$$

$$t_0 = T0.0 exp\left(-\frac{T0.E}{kT}\right) \qquad \text{3-45}$$

$$T_{POW} = TPOW.0 exp\left(-\frac{TPOW.E}{kT}\right) \qquad 3\text{-}46$$

Table 3-5 shows all user-specifiable model parameters for point defect boundary and injection conditions.

Table 3-5 Parameters for specifying point defect boundary and injection conditions

Entity	Pre-exponential factor	Activation Energy
K_{SURF}	KSURF.0	KSURF.E
K_{RAT}	KRAT.0	KRAT.E
K_{POW}	KPOW.0	KPOW.E
A	A.0	A.E
t_0	t0.0	t0.E
T_{POW}	TPOW.0	TPOW.E
Θ	THETA.0	THETA.E
VMOLE	VMOLE	—
G_{POW}	GPOW.0	GPOW.E
K_{SURF}	KSURF.0	KSURF.E

Vacancies

The diffusion and flux equations for vacancies are largely similar to the interstitial equations described above.

$$\frac{\partial C_V}{\partial t} = -\nabla \cdot J_V - R_B \qquad 3\text{-}47$$

Here, J_V is the flux of vacancies and R_B is the bulk recombination rate.

The Vacancy Flux Expression is:

$$-J_V = D_V C_V^* \nabla\left(\frac{C_V}{C_V^*}\right) \qquad 3\text{-}48$$

which correctly accounts for the effect of an electric field on the charged portion of the vacancies by taking the gradient of the normalized concentration $\frac{C_V}{C_V^*}$. The term D_V is the diffusivity of free vacancies, not to be confused with the pair diffusivity D_{AV}, which was mentioned in Section 3.1.2 “The Fermi Model”. The vacancy diffusivity is set according to the following equation:

$$D_V = D.0\,exp\left(-\frac{D.E}{kT}\right) \tag{3-49}$$

where the `D.0` and `D.E` parameters are set on the `VACANCY` statement.

The bulk recombination rate, R_B, is a simple reaction between vacancies and interstitials that assumes that any interstitial is equally likely to recombine with any vacancy, regardless of their charged states. This assumption may overestimate the recombination rate but is a commonly applied assumption. The equation is expressed as:

$$R_B = K_r(C_I C_V - C_I^* C_V^*) \tag{3-50}$$

where K_r is the bulk combination coefficient and is specified as:

$$K_r = KR.0\,exp\left(-\frac{KR.E}{kT}\right) \tag{3-51}$$

where the parameters `KR.0` and `KR.E` are user-definable in the `INTERSTITIAL` statement. This is the same equation for bulk recombination as described earlier for interstitial bulk recombination.

The boundary/interface conditions for vacancies are set similarly to those for interstitials except that the `VACANCY` statements should be used instead of all `INTERSTITIAL` statements.

Dislocation Loop Based Enhanced Bulk Recombination

A topic of some debate in recent literature has been the creation of Dislocation Loops. Currently, the exact physical nature of these defects is still under investigation. It is believed that they arise from amorphizing implants and only exist at the edge of the amorphous layer. It has been suggested that these loops grow through the absorption of interstitials during oxidation and perhaps shrink by the emission of interstitials when annealed in nitrogen.

Due to the lack of proper scientific description, only a simple recombination model has been implemented into Athena. This model introduces an additional sink of interstitials that is described by the following expression:

$$R_{loop} = damalpha\left(C_I - C_I^*\right) \tag{3-52}$$

where *damalpha* is a parameter you can set on the `INTERSTITIAL` statement. This additional recombination is only applied to a region of the amorphizing implant controlled by the parameters `MIN.LOOP` and `MAX.LOOP` in the `DISLOC.LOOP` statement.

As an example, the following statements will produce a region of dislocation loops where the as-implanted phosphorus concentration is between 1e16 cm^{-3} and 1e18 cm^{-3}. The `damalpha` parameter is then set to 1e8 in this region. For example:

```
DISLOC.LOOP MIN.LOOP=1e16 MAX.LOOP=1e18 PHOSPHORUS

INTERSTITIAL SILICON DAMALPHA=1e8
```

```
IMPLANT PHOSPHORUS DOSE=1e15 ENERGY=120
```

This model should only be chosen when either the Two Dimensional or Fully Coupled Model is also used. The two things to apply to this model are as follows:

- an implant has to create an amorphous layer
- immediately after the implant there is an anneal in a wet ambient

Even when these two criteria are met it is suggested to only apply this model when it is needed to match experimental results.

The Steady State Diffusion Model

The Steady State Diffusion Model is a variant of the two dimensional diffusion model, which assumes the point defect profiles are in a steady state. It is turned on with the `METHOD STEADY` command.

Important Notes about Defect Diffusion

Point defects have larger diffusivities than dopants and may therefore diffuse down to the bottom of the structure during a simulation. If the simulation structure is too shallow, you may get an unrealistic high defect concentration in the regions where dopant profiles are present and consequently too much dopant diffusion. Therefore, you may need to extend the depth of the simulation space to provide an adequate sink for the point defects. To determine the depth of the structure, you can estimate the characteristic defect diffusion lengths using:

$$l = \sqrt{D_X \Delta t} \qquad 3\text{-}53$$

where D_X is the defect diffusivity and Δt is the total diffusion time. Simulations show that a depth of 20 to 50 microns is required in most cases. This restriction on the minimum structure depth poses a threat to computational efficiency, whenever diffusion models that include point defects are employed. But since the fine structure of the defect profiles near the bottom of the structure is not a feature of particular interest for processing purposes, you can reduce the computational cost by making the grid very coarse in this region.

Time Step Control

When using diffusion models that include the explicit representation and evolution of point defects, be aware of time stepping issues. Although step size control between iterations is fully automated, you can still specify the size of the initial time step. Different initial time step sizes can be specified for dopants and point defects, respectively, by using the parameters `INIT.TIME` and `PDINIT.TIME`. For example, the command

```
METHOD INIT.TIME=0.001 PDINIT.TIME=0.001
```

would set the initial time step to 1 millisecond for both dopants and point defects. Default values are `INIT.TIME=0.1` seconds and `PDINIT.TIME=1.0E-5` seconds.

Note: There is no guarantee that the program will actually use these values. For this initial time step, the only purpose of these parameters is to make it feasible for you to give the program a hint about an appropriate initial time step size.

3.1.5 The Fully Coupled Model

The Fully Coupled Diffusion Model is identical to the Two Dimensional Model. Be familar with that model before reading any further. The one important difference is that the diffusion of the defects is now influenced by the diffusion of the dopants by the addition of the joint pair fluxes to the flux terms in the governing equation of the defects. Therefore, there is a true two-way interaction between the diffusion of dopants and the diffusion of point defects, which gives this model its name. The fully coupled model is slightly more CPU-intensive than the two dimensional model, but encompasses the capability of reproducing certain important aspects of semiconductor processing such as the Emitter Push Effect in the case of phosphorus diffusion.

From a physical viewpoint, however, this original fully coupled model suffers from the shortcoming of not explicitly representing pairs, and the consequential lack of a subdivision of defects and dopants into paired and non-paired fractions. Therefore, this model cannot reproduce the saturation of the dopant diffusivity that is believed to occur at very high damage concentration due to a total pairing of dopants. In other words, the model relies on the dilute approximation (i.e., the assumption that the concentration of pairs is much smaller than both the dopant and the defect concentrations). To use the Fully Coupled Model, specify parameter `FULL.CPL in` the `METHOD statement`.

The Fully Coupled Model establishes a two-way coupling between the diffusion of dopants and point defects respectively by adding the joint dopant-defect pair fluxes to the flux terms of the defect equations, which then become:

$$\frac{\partial}{\partial t}\left(C_V + \sum_{A,c} C_{AV^c}\right) = -\nabla\cdot\left(J_V + \sum_{A,c} J_{AV^c}\right) - R \qquad \text{3-54}$$

$$\frac{\partial}{\partial t}\left(C_I + \sum_{A,c} C_{AI^c}\right) = -\nabla\cdot\left(J_I + \sum_{A,c} J_{AI^c}\right) - R_B + R_T - R_{\{311\}} \qquad \text{3-55}$$

where summations run over all dopants and pair charge states. The rest of the Fully Coupled Model Equations are identical to those in the Two Dimensional Model described in the previous section. The effect of the correction terms only displays itself at very high dopant or high implant damage conditions or both, where the Fermi level enhancement and point defect supersaturation will increase the dopant diffusivities significantly.

High Concentration Extension to the Fully-Coupled Model

This extension to the fully coupled model takes into account additional higher order defect-dopant/defect pairing the extra point defect recombination mechanisms. This model was developed at Stanford University [14] to include higher order dopant-defect interactions in the cases, where the number of dopant-defect pairs are significant. This is the case for high dopant concentration in silicon. It is activated by the following command:

```
METHOD HIGH.CONC FULL.CPL
```

It is an extension of the basic fully coupled model and may only be used in conjunction with the `METHOD FULL.CPL` command. This model includes two extra bulk recombination reactions and two extra Si/SiO$_2$ interface recombination reactions.

In the bulk, extra terms for point defect recombination apply at high concentrations, where statistically, a high level of dopant-defect pairing is prevalent

$$K_r^{PI-V} = IIFACTOR \cdot \left(\frac{D_{PI} + D_V}{D_I + D_V}\right) \cdot K_r \qquad 3\text{-}56$$

$$K_r^{I-PV} = IVFACTOR \cdot \left(\frac{D_I + D_{PV}}{D_I + D_V}\right) \cdot K_r \qquad 3\text{-}57$$

The extra model parameters calibrate the ratio of effective capture cross sections of dopant-defect/defect to defect/defect recombination mechanisms. The IIFACTOR and IVFACTOR parameters can be set in the INTERSTITAL statement as follows:

```
INTERSTITIAL SILICON IVFACTOR=<n> IIFACTOR=<n>
```

RTA Diffusion Modelling

SSuprem4 has the capability to model rapid thermal annealing (RTA) processes within the framework of existing diffusion models (i.e., the two dimensional model and the fully coupled model). Since RTA is a short time thermal cycle involving steep temperature ramping to high temperatures, Transient enhanced diffusion (TED) will dominate whenever a significant amount of lattice damage is prevalent. Because the amount of dopant diffusion is intimately coupled to the evolution of the point defect populations, you can calibrate these models to RTA conditions by tuning the point defect related parameters. The ratio of interstitial damage in the form of {311} clusters to that in the form of free interstitials and the characteristic time for dissolution of interstitial clusters are important parameters to include when setting up SSPUREM4 for an RTA scenario.

Table 3-6 shows an approximate time for completion of about 95% of the TED at various temperatures.

Table 3-6 The approximate duration of TED at various annealing temperatures

Annealing Temperature (C)	Time for completion of 95% of TED
600	390 hours
700	3.3 hours
750	30 minutes
800	3.7 minutes
850	43 seconds
900	8.3 seconds

Table 3-6 The approximate duration of TED at various annealing temperatures

Annealing Temperature (C)	Time for completion of 95% of TED
950	1.9 seconds
1000	0.48 seconds
1050	0.13 seconds

3.1.6 Electrical Deactivation and Clustering Models

When dopants are present at high concentrations, the electrically active (mobile) concentration, C_{act}, may be less than the corresponding chemical concentration, C_{chem}.

In order for an impurity to become electrically active in a semiconductor material, it must be incorporated into a substitutional lattice site, which then will contribute with a carrier to either the valence band (an acceptor impurity) or the conduction band (a donor impurity). Above certain dopant concentrations, however, it is impossible to incorporate more dopants into substitutional lattice sites. The excess dopants are said to be non-active.

The threshold where the deactivation occurs is often called the solid solubility limit, since impurities can exist in different phases in the crystal. But for this section, we'll call it deactivation threshold. Therefore, it isn't well-defined which phase transition the solid solubility limit might refer to. For example, excess dopants could be participating in small clusters or larger precipitates. Deactivation threshold would be a more proper designation for this limit and will be used throughout the rest of this section. The notation, C_{act}^{th} will be used for the deactivation threshold. Therefore, for all the models described in this section, the following points are assumed for each dopant type:

- Dopants in excess of the deactivation threshold are considered electrically inactive (i.e., they do not contribute to the carrier populations).
- Additionally, dopants in excess of the deactivation threshold are considered to be immobile (i.e., they cannot diffuse).

Electrical Activation Model

The Electrical Activation Model is used to calculate which level of dopant concentration the deactivation occurs. For this purpose, two different Electrical Activation Models are used for all diffusion models:

- An AsV Clustering Model for arsenic or similar models for other impurities.
- A Semi-empirical Activation Model, based on Table B-13 in Appendix B "Default Coefficients", for all other dopants.

These models can be activated in the `IMPURITY` statement by parameters `CLUSTER.ACT` and `SOL.SOLUB` respectively. By default, the `CLUSTER.ACT` model is used only for As in silicon and polysilicon.

The AsV Clustering Model used in SSuprem4 is based on the simple reaction:

$$As^{+} + V^{=} \Leftrightarrow AsV^{-} \tag{3-58}$$

Assuming that Equation 3-58 is always in equilibrium, the following equation describes the relationship between the chemical and the active arsenic concentration.

$$C_{chem} = C_{act} + C_{clust} = C_{act}\left(1 + C_{TN}^{2}\left(\frac{n}{n_i}\right)^{2} \cdot \frac{C_V}{C_V^{*}}\right) \tag{3-59}$$

Since there is a cyclic dependency between the active arsenic concentration, C_{act}, and the carrier concentration, n, an initial guess for the value of n has to be made at the start of the simulation. In addition, C_V/C_V^{*} is (by definition) set to unity when running the Fermi Diffusion Model (see Section 3.1.2 "The Fermi Model").

The clustering coefficient, C_{TN}, is temperature dependent according to the following equation.

$$C_{TN} = CTN.0\,exp\left(+\frac{CTN.E}{kT}\right) \cdot n_i \tag{3-60}$$

Here, the `CTN.0` and `CTN.E` parameters can be defined in the `IMPURITY` statement.

You can specify `CLUSTER.ACT`, `CTN.0` and `CTN.E` parameters for other acceptors in the `IMPURITY` statement. But be aware the model isn't elaborated for other impurities and these parameters are unknown. The model can also be empirically used for acceptors (e.g., indium). The following equation is based on acceptor-interstitial clusters with empirical parameters `CTP.0` and `CTP.E`.

$$C_{chem} = C_{act}\left\{1 + C_{TP}^{2}\left(\frac{p}{n_i}\right)^{2}\frac{C_I}{C_I^{*}}\right\} \tag{3-61}$$

The Semi-empirical Activation Model, based on Table B-13 in Appendix B "Default Coefficients", which is used for all other dopants except arsenic, uses a two-step scheme to calculate the active dopant concentration.

1. The program interpolates into a table of experimental (e.g., temperature and deactivation threshold) data pairs and finds a concentration independent deactivation threshold, C_{act}^{th}, that corresponds to the current simulation temperature. You can set these pairs in the

IMPURITY statement by assigning values to the parameters, SS.TEMP and SS.CONC. The temperature should be specified in Celsius.

2. A logarithmic concentration dependency is incorporated by setting the final deactivation threshold to the value:

$$C_{act}^{th} = \begin{cases} C_{act}^{th} \cdot \left[1 + (1-b)\,ln\dfrac{C_{act}/C_{act}^{th} - b}{1-b}\right]; & C_{act} > C_{act}^{th} \\ C_{act}\,; & C_{act} \le C_{act}^{th} \end{cases} \qquad 3\text{-}62$$

where the parameter b must be in the range of [0.8, 1.0]. Parameter b can be specified as ACT.FACTOR in the IMPURITY statement. The effect of Equation 3-62 is to produce a rounding in the top of the active profile that slightly follows the form of the chemical profile.

Transient Activation Model

The Transient Activation Model assumes that dopants, after an implant, are inactive. A certain time is required before the dopants become active. After an ion is implanted into silicon, this model assumes that all dopants are inactive and may not be activated immediately but become gradually active instead. The Transient Activation Model simulates this behavior and applies it to activating the implant dopants.

The following equation for the active concentration C_A is solved.

$$\frac{\partial}{\partial t}(C_{Chem} - C_A) = \frac{C_C - C_A^{eq}}{\tau_A} \qquad 3\text{-}63$$

C_C is the chemical concentration of the dopant, and $C_A{}^{eq}$ is the equilibrium active concentration calculated either from solid solubility or clustering model (for arsenic) as defined in the previous section. The parameter τ_A is the time constant for activation, which is a function of temperature, and is calculated using the following Arrhenius expression.

$$\tau_A = TRACT.0 \cdot exp\left(-\frac{TRACT.E}{kT}\right) \qquad 3\text{-}64$$

The initial condition at time $t=0$ for Equation 3-63 is specified by

$$C_A\Big|_{t\,=\,0} = min(C_A^{eq}, TRACT.MIN \cdot n_i) \tag{3-65}$$

where n_i is the intrinsic carrier concentration. Therefore, implantation activation will occur immediately up to a level of $TRACT.MIN \cdot n_i$, after the active concentration is calculated according to Equations 3-59 and 3-62.

To activate the transient activation model, set the CLUST.TRANS parameter in the METHOD statement and specify the TRACT.0, TRACT.E, and TRACT.MIN parameters in the IMPURITY statement.

The defaults for B, P, As and Sb are TRACT.0=8e-16sec, TRACT.E=-4.2, and TRACT.MIN=1.0 (TRACT.MIN for Phosphorus is 2.0).

3.1.7 Grain-based Polysilicon Diffusion Model

The mechanism for impurity diffusion in polysilicon is different than that of crystalline silicon. Polysilicon has a micro-structure of small (compared to the interesting device regions) crystalline regions called grains. These grains are separated by grain boundaries which occupy a certain spatial volume and are connected to form a complex network. The texture and morphology of the grain structure depends on the deposition conditions and on subsequent thermal treatment (during which recrystallization can occur). Impurities inside the grain will diffuse differently than those in the grain boundaries. Dopant will also transport through grain and grain boundary interfaces.

A model for impurity diffusion in polysilicon outlined in [18], [19], and [20] is incorporated in Athena. To use polysilicon diffusion, specify the POLY.DIFF parameter in the METHOD statement. Most of parameters, which control the model, have a prefix PD and are specified in the IMPURITY statement. In this model, the concentration of each impurity C_i is split into two components, namely the concentration within grain interior C_i^g and concentration in the grain boundary C_i^{gb}. The impurity diffusion within grain interior is simulated by the standard model used for crystalline silicon (see Equations 3-2 to 3-7):

$$\frac{\partial C_i^g}{\partial t} = \nabla\left(D_i^g\,\nabla C_i^g - ZD_i^g C_i^g \frac{1}{n}\nabla n\right) + G_i \tag{3-66}$$

where diffusivity of impurity *i* within grains D_i^g is calculated exactly as in Equations 3-10 and 3-11.

The diffusion in the grain boundary is assumed to be constant and very rapid:

$$\frac{\partial C_i^{gb}}{\partial t} = \nabla^2(D_i^{gb}C_i^{gb}) - G_i \qquad \text{3-67}$$

where D_i^{gb} is diffusivity of impurity *i* along the grain boundaries:

$$D_i^{gb} = \mathtt{PD.DIX.0} \cdot exp\left(-\frac{\mathtt{PD.DIX.E}}{kT}\right) \qquad \text{3-68}$$

The PD.DIX.0 and PD.DIX.E parameters are specified in the IMPURITY statement. The last term G_i in Equations 3-66 and 3-67 controls impurity segregation between grain interior and grain boundaries:

$$G_i = \left(\frac{C_i^g}{p_{seg}} - C_i^{gb}\right)\tau^{-1} \qquad \text{3-69}$$

where p_{seg} is segregation coefficient and τ is the rate of segregation specified as PD.TAU in the IMPURITY statement. The segregation process brings the ratio between grain interior and grain boundary concentration to a steady state condition where $C_i^g/C_i^{gb} = P_{seg}$. Initial conditions for Equations 3-66 and 3-67 are determined by setting the PD.CRATIO parameter in the IMPURITY statement. This parameter specifies the initial ratio between impurity concentration in the grain boundary C_i^{gb} and total concentration $C_i^{gb} + C_i^g$.

The grain boundary segregation P_{seg} is calculated according the model suggested in [20]

$$P_{seg} = \left[\frac{AQ_S}{L_g(t)N_{Si}} exp\left(\frac{Q_0}{kT}\right)\right]^{-1} \qquad \text{3-70}$$

where Q_S is the density of segregation sites at the grain boundary specified by the PD.SEGSITES parameter in the IMPURITY statement, N_{Si} is the atomic density of crystalline silicon (2.5 10^{22} atoms/cm^3), *A* is the entropy factor specified by the PD.EFACT parameter in the IMPURITY statement, Q_0 is the segregation activation energy specified by the PD.SEG.E parameter in the IMPURITY statement, and $L_g(t)$ is the time-dependent grain size calculated according to the grain growth model suggested in [19]:

$$L_g(t) = \sqrt{L_0^2 + \frac{PD.GROWTH.0}{kT} \cdot exp\left(-\frac{PD.GROWTH.E}{kT}\right) \cdot t} \qquad \text{3-71}$$

where L_0 is the initial polysilicon grain size, which should be specified by GR.SIZE parameter in the DEPOSIT POLYSILICON statement (GR.SIZE.F parameter allows to have linearly graded grain size within deposited polysilicon layer), the grain growth parameters

PD.GROWTH.0 and PD.GROWTH.E are specified in the IMPURITY statement, and *t* is the elapsed diffusion time.

The segregation boundary condition at the polysilicon/silicon boundary is also modified when POLY.DIFF model is used. The default segregation coefficient M_{12} in Equation 3-12 is increased by a factor, which depends on the impurity concentration in the grain boundaries:

$$M^{*}_{12} = M_{12}\left(1 + PD.SEG.GBSI \cdot \frac{C_i^{gb}}{C_i^{gb} + C^{g}}\right) \tag{3-72}$$

where $M*_{12}$ is modified segregation coefficient and the PD.SEG.GBSI parameter is specified in the IMPURITY statement.

3.2 Advanced Diffusion Models

The ultimate goal of TCAD simulation is to compute the electrical characteristics of a given device by using only process-related data as input parameters. Since the electrical characteristics of the device are heavily dependent on the distribution of the electrically active impurity (dopants), resulting from the entire thermal processing sequence, it is important that the diffusion models used in the process simulation are as accurate as possible. This is particularly important for deep sub-micron processes. Therefore for these emerging technologies, 2D or even 3D phenomena are expected to be of growing importance, whereas there is presently no accurate technique to measure multi-dimensional dopant profiles. Consequently, the active dopant 2D distributions can only be obtained by simulation based on models that are as reliable as possible. It has become clear that the "abnormal" behaviors of dopant diffusion in silicon are caused by non-equilibrium point defects. These are induced by the diffusion process itself (emitter push effect caused by high concentration of phosphorus diffusion), or injected into the substrate by external treatments, such as oxidation or silicidation. Otherwise, they result from the ion implantation used to introduce the dopants into the silicon substrate. With the necessary decrease of the thermal budget due to the shrinkage of the device dimensions, these transient phenomena become key issues for accurate dopant diffusion simulation.

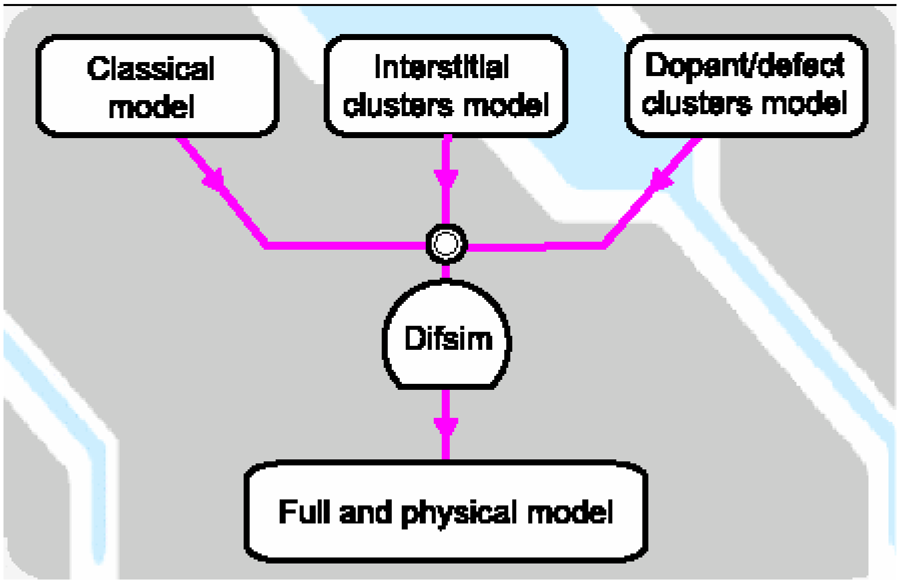


Figure 3-1 The model consists of three parts: the classical dopant diffusion model, the interstitials clusters model and the model of mixed dopant clustering.

The new model of dopant diffusion implemented in Athena is called PLS and was developed in collaboration with CNRS-Phase (Strasbourg, France), CEA-LETI (Grenoble, France) and SILVACO. This model is up to date with actual physical models and contains only physical parameters [21]. It consists of three parts: the classical dopant diffusion (CDD) model, the interstitials clusters (IC) model and the model of mixed dopant-defect clustering (DDC). This section describes the three parts of the model and how the new model differs from the fully coupled model.

The main physical points taken into account in the models are the following:

- Dopant diffusion of all species is assisted by both vacancies (*V*) and self-interstitials (*I*). These point-defects exist in various charge states and their relative concentrations depend on the local Fermi level position (i.e. on the local dopant concentration).
- Both *I* and *V* have strong binding energies with the dopant atoms and consequently the diffusing species are dopant-defect pairs (the isolated substitutional dopants are immobile). These impurity-defect pairs in their various charge states are not assumed to be in local equilibrium with the free substitutional dopant atoms and the free defects. In the PLS model, at high dopant concentrations, the concentrations of these pairs are not considered to be negligible with respect to the substitutional (active) dopant concentration. Therefore, the pair concentrations are explicitly taken into account to compute the total dopant concentration and the Fermi level position (i.e., carrier concentration). Consequently, a partial self-compensation takes place at high doping concentrations, which contributes to the differences between total and active concentrations and affects the variations of the extrinsic diffusivities as a function of the total doping.
- The flux of each diffusing species (dopant-defect pairs and free defects) includes the drift terms caused by the built-in electric field, due to the dopant gradients.
- Both *I* and *V* are not considered to be in local equilibrium but can be annihilated by bimolecular recombination. This feature of the CDD model performs annihilations between not only the free defects but also between the impurity defects pairs, which play the role of recombination centers. Therefore, the *I*-*V* recombination rate is strongly enhanced at high dopant concentration.
- It is now well established that transient enhanced diffusion (TED) is strongly correlated with the evolution of the self-interstitial supersaturation governed by the nucleation and evolution during the high temperature anneal of a variety of extended defect structures, such as the interstitials clusters. Thus, the predictive process modeling of the deep submicron MOSFET technologies requires the development of accurate diffusion models which take into account the full set of interactions between dopants and the point or extended defects (clusters). The PLS model, coupled with the BCA implantation, allows you to calculate the evolution of the clusters. Therefore, it is unneccessary to artificially add {311} or other clusters because the model automatically generates them and calculates their evolutions according to the Ostwald ripening theory.
- At concentrations near the solid solubility limit, a dynamic clustering model is considered. Immobile complexes (As_nV or B_nI_m) are formed, which result in decrease of the effective diffusivity and increase of the inactive dopant concentration. These complexes are not assumed to be in local equilibrium with the other species.
- When the dopant concentration exceeds a few 10^{20}cm^{-3}, the dopant-vacancy pairs can no longer be considered as isolated entities because the vacancies can interact with more than one dopant atom.

3.2.1 Classical Model of Dopant Diffusion (CDD)

The basic idea of the model is isolated substitutional dopant atoms (A_s) are immobile. The dopant diffusion occurs only through the migration of dopant--self-interstitial (*AI*) and of dopant--vacancy (*AV*) pairs. Moreover, in this enhanced model, local equilibrium is not assumed between the pairs and their components (unlike the original CNET model [5]). All possible charge states of the free defects and of the pairs have been considered and their relatives concentrations depending on the local Fermi level position.

To turn on the CDD model, specify `PLS` parameter in the `METHOD` statement. All physical parameters of the model can be modified in the `*.mod` files. To specify the location of these files, use the `B.MOD`, `P.MOD`, `AS.MOD`, `IC.MOD`, and `VI.MOD` parameters in the `DIFFUSE` statement. By default all these files are located in the `$SILVACO/lib/athena/<version_number>/common/pls` directory.

Charge States

Point Defects

The result of diffusion studies in metals and ionic crystals have led to the establishment of several basic atomic diffusion mechanisms. These mechanisms dominate the interpretation of silicon diffusion experiments with the exception that in silicon there is a very wide energy range available to the Fermi level. Therefore, a given lattice defect can appear in a variety of ionized states. The fundamental principles of thermodynamic predict that such defects will exist in equilibrium at all temperatures above 0° K, because the presence of such defects minimizes the free energy of the crystal. The entities $V_{eq}^{(0)}$ and $I_{eq}^{(0)}$ are the equilibrium defect concentrations for vacancies and silicon self-interstitials in their neutral charge state. The weight factors γ and δ account for the different charge states for distribution of point defects under extrinsic conditions. Each γ and δ is assumed to be temperature dependent through Arrhenius expressions.

For point defects (*V* or *I*), five various charge states are considered:

$$I^{(s)} = \gamma^{(s)}\left(\frac{n_i}{n}\right)^s I^0, \quad V^{(s)} = \delta^{(s)}\left(\frac{n_i}{n}\right)^s V^0 \tag{3-73}$$

where *s* is one of the charge states -2, -1, 0,+1, or +2. All parameters γ and δ are specified in the charge state statements of the `defect.mod` file.

With this consideration, the equilibrium concentrations V^{eq} and I^{eq} are estimated as

$$I^{eq} = I^0_{eq}\sum\gamma^{(s)}\left(\frac{n_i}{n}\right)^s, \quad V^{eq} = V^0_{eq}\sum\delta^{(s)}\left(\frac{n_i}{n}\right)^s . \tag{3-74}$$

Under intrinsic conditions (i.e., $n=p=n_i$), the equilibrium concentration can be simply written as:

$$I_i^{eq} = I^0_{eq}\sum\gamma^{(s)}, \quad V_i^{eq} = V^0_{eq}\sum\delta^{(s)}. \tag{3-75}$$

Equilibrium concentration for vacancies and silicon self-interstitials are defined as a simple Arrhenius functions:

$$I_i^{eq} = K_I^0 \, exp\left(-\frac{E_f^I}{kT}\right), \; V_i^{eq} = K_V^0 \, exp\left(-\frac{E_f^V}{kT}\right) \qquad 3\text{-}76$$

where E_f^I and E_f^V represent respectively the formation energy for *I* and *V*, and K_I^0 , K_V^0 are coefficients. These parameters are specified in the `defect.mod` file.

Dopant-defect Pairs

Using the same assumption as for point defects, the dopant-defect pairs are defined by the charge states of -1, 0, +1.

For boron pairs we have

$$BI^{(s)} = \left(\frac{p}{p_{BI}^{(s)}}\right)^s BI^{(0)}, \qquad p_{BI}^{(s)} = \frac{K_{BI}^{\left(\frac{s-1}{2}\right)}}{K_{BI}^{\left(\frac{s+1}{2}\right)}} \cdot \frac{\gamma^{\left(\frac{s+1}{2}\right)}}{\gamma^{\left(\frac{s+3}{2}\right)}} \cdot n_i, \qquad 3\text{-}77$$

$$BV^{(s)} = \left(\frac{p}{p_{BV}^{(s)}}\right)^s BV^{(0)}, \qquad p_{BV}^{(s)} = \frac{K_{BV}^{\left(\frac{s-1}{2}\right)}}{K_{BV}^{\left(\frac{s+1}{2}\right)}} \cdot \frac{\delta^{\left(\frac{s+1}{2}\right)}}{\delta^{\left(\frac{s+3}{2}\right)}} \cdot n_i, \qquad 3\text{-}78$$

The *BV* pairs exist, however, with positive (+1) and neutral (0) charge states only. Moreover, it is well established *BI* pairs exhibit a negative-*U* behavior and neutral state is unstable. Therefore, the three charge states BI^+, BI^0 and BI^- are considered.

For donor impurities, the three charge states for dopant-interstitial pairs are also considered. Though only AV^- and AV^0 are known to exist (*E*-centers).

$$AI^{(s)} = \left(\frac{n}{n_{AI}^{(s)}}\right)^s AI^{(0)}, \qquad n_{AI}^{(s)} = \frac{K_{AI}^{\left(\frac{s+1}{2}\right)}}{K_{AI}^{\left(\frac{s-1}{2}\right)}} \cdot \frac{\gamma^{\left(\frac{s-1}{2}\right)}}{\gamma^{\left(\frac{s-3}{2}\right)}} \cdot n_i, \qquad 3\text{-}79$$

$$AV^{(s)} = \left(\frac{n}{n_{AV}^{(s)}}\right)^s AV^{(0)}, \qquad n_{AV}^{(s)} = \frac{K_{AV}^{\left(\frac{s+1}{2}\right)}}{K_{AV}^{\left(\frac{s-1}{2}\right)}} \cdot \frac{\delta^{\left(\frac{s-1}{2}\right)}}{\delta^{\left(\frac{s-3}{2}\right)}} \cdot n_i, \qquad 3\text{-}80$$

The $K_{AI}{}^{(s)}$ and $K_{AV}{}^{(s)}$ parameters for both donors and acceptors represent the pairing coefficients for the dopant-interstitial and dopant-vacancy pairs and are defined by following formula:

$$K_{AI^{(s)}} = K^{0}_{AI^{(s)}}\ exp\left(-\frac{E_{AI^{(s)}}}{kT}\right), \qquad K_{AV^{(s)}} = K^{0}_{AV^{(s)}}\ exp\left(-\frac{E_{AV^{(s)}}}{kT}\right) \qquad \text{3-81}$$

Fully Coupled Equations

The point defects, dopant-defect pairs and active dopant evolve according to the following continuity equations:

$$\begin{aligned}
\frac{\partial I^{tot}}{\partial t} &= -\nabla\cdot \boldsymbol{J}_I + GR_{IV} + GR_{AI} + GR_{AV-I},\\
\frac{\partial V^{tot}}{\partial t} &= -\nabla\cdot \boldsymbol{J}_V + GR_{IV} + GR_{AV} + GR_{AI-V},\\
\frac{\partial AI^{tot}}{\partial t} &= -\nabla\cdot \boldsymbol{J}_{AI} - GR_{AI} + GR_{AI-V},\\
\frac{\partial AV^{tot}}{\partial t} &= -\nabla\cdot \boldsymbol{J}_{AV} - GR_{AV} + GR_{AV-I},\\
\frac{\partial A_s}{\partial t} &= GR_{AI} + GR_{AV} - GR_{AV-I} - GR_{AI-V}
\end{aligned} \qquad \text{3-82}$$

Here, X^{tot} is the total concentration of the species X, and $[A_S]$ is active dopant concentration in substitutional position. $\boldsymbol{J}$ represents the flux of species X, and GR_X is a generation-recombination term corresponding to the reactions that contain X.

Flux Equations

Point Defects

As the migration rate of a given species may depend on its charge state, two distributions must take into account in the flux equation. The first one is the Fickian term and the second is the Nerstian term. Thus, the flux can be written for a given species X with the charge state s:

$$\boldsymbol{J}_{X^s} = -D_{X^s}\left(\Delta X^S - sX^S\frac{\Delta p}{p}\right). \qquad \text{3-83}$$

For point defects, the PLS model makes the assumption that the diffusivity is independent of the charge state at high temperature. This implies that for vacancy and silicon self-interstitial at any charge state s the following is always true.

$$D_I = D_{I^s}, \qquad D_V = D_{V^s} \qquad \text{3-84}$$

Moreover, it is assumed that diffusivities at high temperature follow the simple Arrhenius law:

$$D_I = D_I^0 \, exp\left(-\frac{E_I^m}{kT}\right), \qquad D_V = D_V^0 \, exp\left(-\frac{E_V^m}{kT}\right) \tag{3-85}$$

where E_I^m and E_V^m represent respectively the migration energy for self-interstitials and vacancies. Parameters D_I^0, D_V^0, E_I^m, and E_V^m are defined in the `defect.mod` file.

Finally, taking into account that $np = n_i^2$ and

$$I^{(s)} = I^{tot}\left[\sum_s \gamma^s\left(\frac{n}{n_i}\right)^s\right]^{-1}, \qquad V^{(s)} = V^{tot}\left[\sum_s \delta^s\left(\frac{n}{n_i}\right)^s\right]^{-1}, \tag{3-86}$$

the flux equations for I^{tot} and V^{tot} are defined as follows:

$$J_{I^{tot}} = \sum_s J_{I^s} = -D_I\left[DI^{tot} - I^{tot}\frac{\sum_s s\gamma^{(s)}\left(\frac{n_i}{n}\right)^s}{\sum_s \gamma^{(s)}\left(\frac{n_i}{n}\right)^s}\frac{\Delta n}{n}\right] \tag{3-87}$$

$$J_{V^{tot}} = \sum_s J_{V^s} = -D_V\left[DV^{tot} - V^{tot}\frac{\sum_s s\delta^{(s)}\left(\frac{n_i}{n}\right)^s}{\sum_s \delta^{(s)}\left(\frac{n_i}{n}\right)^s}\frac{\Delta n}{n}\right] \tag{3-88}$$

Dopant--Defects Pairs

The diffusion of dopant--defect pairs implies on the fact that diffusivities D_A^V and D_A^I are calculated from the basic parameters for the pairs defined by Equation 3-77 through 3-81 as follows:

$$D_A^I = f_I D_{Ai} = \sum_s D_{AI^s} AI^{eq}, \qquad D_A^V = (1-f_I)D_{Ai} = \sum_s D_{AV^s} AV^{eq}, \tag{3-89}$$

where f_I is the interstitialcy component under intrinsic conditions, and D_{Ai} is the intrinsic diffusivity of the impurity A. The f_I and D_{Ai} parameters are defined for each dopant in the corresponding `dopant.mod` file.

It should be emphasized that several relationships exist between this various parameters. Therefore, you can decrease the number of free fitting parameters. From physical point of view, it can be safely assumed the ratio between the diffusivities of the various charge states must be equal. Consequently, we have

$$\frac{D_{AI^-}}{D_{AI^0}} = \frac{D_{AI^0}}{D_{AI^+}} = (\rho_{AI})^S, \qquad \frac{D_{AV^-}}{D_{AV^0}} = \frac{D_{AV^0}}{D_{AV^+}} = (\rho_{AV})^S, \tag{3-90}$$

where $s=1$ for donors and $s= -1$ for acceptors. These diffusivities are free parameters when the various coupling parameters in Equation 3-81 are known. Therefore, you can calculate them through the experimentally known intrinsic diffusivities D_{Ai} and the interstitialcy component f_I. The ρ_{AI} and ρ_{AV} parameters are defined as an Arrhenius functions.

Finally, the flux equations, for instance, for boron--point defect pairs are defined as follows:

$$J_{BI^{tot}} = \sum_s J_{BI^s} = -D_{BI^0} \frac{\sum_s (\rho_{BI})^s K_{BI^s} \left(\frac{p}{n_i}\right)^{s_1} \gamma^{(s_1)}}{\sum_s K_{BI^s} \left(\frac{p}{n_i}\right)^{s_1} \gamma^{(s_1)}} \times \qquad 3\text{-}91$$

$$\left(\Delta BI^{tot} - BI^{tot} \frac{\sum_s sK_{BI^s} \left(\frac{p}{n_i}\right)^{s_1} \gamma^{(s_1)}}{\sum K_{BI^s} \left(\frac{p}{n_i}\right)^{s_1} \gamma^{(s_1)}} \frac{\Delta p}{p} \right),$$

$$J_{BV^{tot}} = \sum_s J_{BV^s} = -D_{BV^0} \frac{\sum_s (\rho_{BV})^s K_{BV^s} \left(\frac{p}{n_i}\right)^{s_1} \delta^{(s_1)}}{\sum_s K_{BV^s} \left(\frac{p}{n_i}\right)^{s_1} \delta^{(s_1)}} \times \qquad 3\text{-}92$$

$$\left(\Delta BV^{tot} - BV^{tot} \frac{\sum_s sK_{BV^s} \left(\frac{p}{n_i}\right)^{s_1} \delta^{(s_1)}}{\sum K_{BV^s} \left(\frac{p}{n_i}\right)^{s_1} \delta^{(s_1)}} \frac{\Delta p}{p} \right),$$

where $s_1=s+1$.

The equations for pairs formed by donor impurities are completely symmetrical to the equations above.

Generation-Recombination Terms

The generation-recombination terms GR_X in Equation 3-82 describe the evolution of particular reactions. For example, GR_{AV} represents the formation of the dopant-vacancy:

$$A_s + V \underset{k^r_{AV}}{\overset{k^f_{AV}}{\Leftrightarrow}} AV \qquad 3\text{-}93$$

Formation of Pairs

In the model, the formation of dopant-defect pairs is taken into account by simulating the following reactions:

$$A_s^- + I^0 \underset{k_{AI}^r}{\overset{k_{AI}^f}{\Leftrightarrow}} AI^- \tag{3-94}$$

$$A_s^- + V^+ \underset{k_{AV}^r}{\overset{k_{AV}^f}{\Leftrightarrow}} AV^0 \tag{3-95}$$

where k^f and k^r are the reaction kinetic constants for each reaction. They are define as

$$k_{AI}^f = 4\pi R^{eff} D_I, \qquad k_{AI}^r = k_{AI^0}^f / K_{AI^0} \tag{3-96}$$

$$k_{AV}^f = 4\pi R^{eff} D_V, \qquad k_{AV}^r = k_{AV}^f / K_{AV^0} \tag{3-97}$$

where $K_{AI}{}^0$, $K_{AV}{}^0$ are defined by Equation 3-81, and R^{eff} is the silicon lattice constant. Thus, the generation-recombination terms in Equation 3-82 are as follows:

$$GR_{AI} = k_{AI}^f A_s^- \; I^0 - k_{AI}^r AI^0, \tag{3-98}$$

$$GR_{AV} = k_{AV}^f A_s^- \; V^+ - k_{AV}^r AV^+ \tag{3-99}$$

Frenckel Pair Recombination

During annealing, many of the interstitials and vacancies recombine either at the surface or in the bulk. The driving force for this reaction is to change both interstitial concentration I^{tot} and vacancy concentration V^{tot} toward their equilibrium concentrations I_{eq}^{tot} and V_{eq}^{tot}. Moreover, it is clearly shown that defect recombination strongly depends on the impurity concentration.

The following reaction is also considered.

$$V^{\alpha} + V^{\beta} \underset{k_{VI}^r}{\overset{k_{VI}^f}{\Leftrightarrow}} \langle 0 \rangle \tag{3-100}$$

The recombination rate can be written as follows:

$$GR_{IV} = k_{BM}(I_{eq}^{tot} \; V_{eq}^{tot} - I^{eq} V^{eq} \quad). \tag{3-101}$$

Here:

$$k_{BM} = k_a\left(\sum_s(\gamma^{(s)} + \delta^{(s)})\left(\frac{n_i}{n}\right)^s - 1\right) \qquad 3\text{-}102$$

and

$$k_a = 4\pi a_0(D_I + D_V), \qquad 3\text{-}103$$

where a_0=2.35°A is the distance between two separated silicon atoms in crystal.

Bimolecular Recombination

As an alternative to the direct recombination of point defects in reaction of Equation 3-100, it is also possible for I and V to recombine through reactions such as

$$AV^0 + I^+ \underset{k^r_{AV-I}}{\overset{k^f_{AV-I}}{\Leftrightarrow}} A^+_s \quad , \qquad AI^0 + V^+ \underset{k^r_{AI-V}}{\overset{k^f_{AI-V}}{\Leftrightarrow}} A^+_s \quad . \qquad 3\text{-}104$$

In these cases, the annihilation of the Frenckel pairs implies a dissociation of a dopant-defect pairs.

Recombination at the Surface

Understanding of the mechanisms that determine the interaction of interstitials and vacancies with the interfaces is getting more important, because the implantation energies of dopants and the temperatures for the thermal treatments become lower and the devices are fabricated closer to the surface. It has been also demonstrated that the fundamentals of the point defect properties are critically important in accurate prediction of device behavior. For example, the reverse short channel effect.

The recombination flux of silicon self-interstitials at a nonoxidizing interface Γ is given by the formula:

$$\mathbf{J}_I\big|_\Gamma = \frac{D_I}{L_I}\left(I^{tot} - I^{tot}_{eq}\right), \qquad \mathbf{J}_V\big|_\Gamma = \frac{D_V}{L_V}\left(V^{tot} - V^{tot}_{eq}\right), \qquad 3\text{-}105$$

where L_V and L_I are called the recombination length at the surface. These parameters can be used for adjusting the recombination rate at the surface and are specified in the `defect.mod` file.

In the case of dopant implantation, an exodiffusion may occur during the thermal treatment. Therefore, you can write the dopant flux at the surface as:

$$\mathbf{J}_{AI}\big|_\Gamma = \sigma_{AI}AI^{tot}, \qquad \mathbf{J}_{AV}\big|_\Gamma = \sigma_{AV}AV^{tot} \qquad 3\text{-}106$$

where σ_{AI} and σ_{AV} are the exodiffusion coefficients for each dopant-defect pair and are defined in `dopand.mod` files. These parameters are defined by the following Arrhenius functions:

$$\sigma_{AI} = \sigma^{0}_{AI} exp\left(-\frac{E_{\sigma AI}}{kT}\right), \qquad \sigma_{AV} = \sigma^{0}_{AV} exp\left(-\frac{E_{\sigma AV}}{kT}\right). \qquad 3\text{-}107$$

3.2.2 Solid Solubility Model

The model for precipitation is assumed to be a constant solid solubility cut-off. This means that all solute atoms above the solid solubility level will form a precipitate almost instantaneously. This model is activated by adding the `SS` parameter to the `METHOD PLS` statement.

The rate equation for the solid solubility model can be formulated as follows [26]:

$$\frac{A_{pr}}{\partial t} = D\lambda A_{pr}(A_s - A_{ss}(T)) + \begin{cases} D\lambda(A_s – A(T)) & for\ \ A_s > A_{ss}(T), \\ 0 & for\ \ A_s < A_{ss}(T), \end{cases} \qquad 3\text{-}108$$

where A_{pr} is the concentration of the precipitate and λ is the effective length of capture. The solid solubility $A_{ss}(T)$ is defined using Arrhenius expressions in the corresponding `dopant.mod` file.

3.2.3 Interstitials Clusters Model (IC)

Point defects in crystalline materials inherently have high free energy. Free interstitials in silicon are thermodynamically unstable because of their unpaired electron orbitals and induced lattice strain. At high concentrations, the interstitials clusters are formed to reduce free energy. Many of interstitial cluster species have been observed for many years (e.g., {311} defects and dislocation loops). The interstitial cluster configurations are believed to occur mainly in ion-implanted silicon. The formation and dissolution of interstitial clusters are simulated to correctly predict TED.

To activate the interstitial clusters models, use the following statement:

```
METHOD PLS IC
```

During a typical rapid thermal annealing, various type of clusters (small clusters, {311} defects, perfect and faulted loops) evolve according to a competitive growth mechanism named Ostwald ripening. The driving force for this evolution is the reduction of the formation energy per interstitial of these clusters as they grow in size and change their crystallographic structures.

In IC model, a cluster containing n interstitials (I_n) evolve to a cluster of size $n+1$ by interaction with a free interstitial according to the following reactions:

$$I^0 + I^0 \underset{k^r_2}{\overset{k^f_2}{\Leftrightarrow}} I_2{}^0, \qquad I_2{}^0 + I^0 \underset{k^r_3}{\overset{k^f_3}{\Leftrightarrow}} I_3{}^0, \qquad \ldots \qquad I^0_{n-1} + I^0 \underset{k^r_n}{\overset{k^f_n}{\Leftrightarrow}} I^0_n, \qquad 3\text{-}109$$

where according to [22]:

$$k^f_n = 4\pi R_{eff} D_I exp\left(-\frac{\Delta g_n}{kT}\right), \quad k^r_n = 6\frac{D_I}{\lambda^2}\theta_n exp\left(-\frac{E^I_f - E_f(n)}{kT}\right) \qquad 3\text{-}110$$

Here, R_{eff} represents the effective capture radius. The elementary jump length λ is equal to the inter-atomic distance a_0=2.35°A. θ_n is the number of dissociating sites, E_f^I is the self interstitial formation energy defined in Equation 3-76, and $E_f(n)$ is the formation energy per interstitial for clusters of size *n*. The number of reactions taken into account is specified in the `ic.mod` file.

The value of effective energy barrier Δg_n in Equation 3-110 can be represented as the sum of two components:

$$\Delta g_n = max\left(-k_B T ln\left(\frac{I^{tot}}{I_{eq}^{tot}}\right) + E_f(n), 0\right). \qquad 3\text{-}111$$

The first of the two terms represents the change in free energy associated to the change in chemical potential when an interstitial jumps from the supersaturated phase to the cluster. The second term is the formation energy per interstitials. This parameter is a function of the size and of the crystallographic structure of the cluster. The values of the formation energy per interstitial are specified in the `ic.mod` file. The first nine parameters corresponding to the formation energy per interstitial for clusters of size 2 through size 10 are defined in [23]. For larger clusters, the energy of formation per interstitial is calculated by the following empirical formula:

$$E_f(n) = Alog\left(1 + \frac{1}{(n+B)^C}\right) + E_f^{\{311\}}(\infty). \qquad 3\text{-}112$$

The empirical parameters of the model *A*, *B*, *C* and $E_f^{\{311\}}$ are specified in the `ic.mod` file. Thus, the system of coupled equations for the IC model can be written as:

$$\begin{aligned}
\frac{\partial I^{tot}}{\partial t} &= -\nabla\cdot \boldsymbol{J}_I + GR_{IV} + GR_{AI} + 2GR_{IC(2)} + \sum_{n=3}^{N} GR_{IC(n)}, \\
\frac{\partial V^{tot}}{\partial t} &= -\nabla\cdot \boldsymbol{J}_V + GR_{IV} + GR_{AV}, \\
\frac{\partial AI^{tot}}{\partial t} &= -\nabla\cdot \boldsymbol{J}_{AI} - GR_{AI} + GR_{AI-V}, \\
\frac{\partial AV^{tot}}{\partial t} &= -\nabla\cdot \boldsymbol{J}_{AV} - GR_{AV} + GR_{AV-I}, \\
\frac{\partial A_s}{\partial t} &= GR_{AI} + GR_{AV} - GR_{AV-I} - GR_{AI-V}, \\
\frac{\partial I_n}{\partial t} &= GR_{IC(n)} - GR_{IC(n+1)}, \qquad n = 2,3,\ldots,N,
\end{aligned} \qquad 3\text{-}113$$

where the generation-recombination term:

$$GR_{IC(n)} = k_n^f I^{tot} I_{n-1} - k_n^r I_n. \qquad 3\text{-}114$$

It is important to notice that the IC model is completely independent of the dopants involved in the process.

3.2.4 Vacancy Cluster Model (VC)

In the VC model, a cluster containing *m* vacancies (V_m) evolves to a cluster of size *m+1* by interaction with a free vacancy according to the following reactions:

$$V^0 + V^0 \underset{k_2^r}{\overset{k_2^f}{\Leftrightarrow}} V_2^0 , \qquad V_2^0 + V^0 \underset{k_3^r}{\overset{k_3^f}{\Leftrightarrow}} V_3^0 , \qquad \ldots \qquad V_{m-1}^0 + V^0 \underset{k_m^r}{\overset{k_m^f}{\Leftrightarrow}} V_m^0 , \tag{3-115}$$

where:

$$k_m^f = 4\pi R_{eff} D_V, \qquad k_m^r = 6\frac{D_V}{\lambda^2}\theta_m exp\left(-\frac{E_f^V - E_f(m)}{kT}\right), \ m = 2, \ldots, M . \tag{3-116}$$

Here, R_{eff} represents the effective capture radius and the elementary jump length λ is equal to the inter-atomic distance. θ_m is the number of dissociating sites. E_f^V is the vacancy formation energy defined in Equation 3-76, and $E_f(m)$ is the formation energy per vacancy for clusters of size *m*.

To take into account the interactions VC with interstitials, the following reactions are added:

$$V_2^0 + I^{(0)} \underset{k_2^r}{\overset{k_2^f}{\Leftrightarrow}} V_2^{(0)}, \qquad V_3^0 + I^0 \underset{k_{V3}^r}{\overset{k_{I4}^f}{\Leftrightarrow}} V_3^0 , \qquad \ldots \qquad V_m^0 + I^0 \underset{k_{Vm}^r}{\overset{k_{Im+1}^f}{\Leftrightarrow}} V_{m-1}^0 \tag{3-117}$$

Therefore, the VC model consists of Equations 3-113 and 3-118.

$$\frac{\partial V_m}{\partial t} = GR_{VC(m)} - GR_{VC(m+1)} \tag{3-118}$$

According to the following reactions, Equations 3-115 and 3-117, there are two kinds of summands in the generation-recombination part.

$$GR_{VC(m)} = k_m^f V^0 V_{m-1} + k_{Im+1}^f I^0 V_m \qquad GR_{VC(m+1)} = k_m^r V_m^0 + k_{m-1}^r V_{m-1}^0 \tag{3-119}$$

The equations for vacanies and intersititials will also contain the additional terms from Equation 3-119.

3.2.5 Electrical Deactivation and Clustering Models (DDC)

At high doses of dopant, the electrically active concentration may be less than corresponding chemical concentration. The impurity atom becomes activated inside semiconductor only if it is incorporated into a substitutional lattice site. In this case, the activated atom will contribute with a carrier to either the valence band (an acceptor impurity), or the conduction band (a donor impurity). It has been observed that, even below solid solubility, a significant dopant concentration can stay inactive. This effect can be explained by the formation of immobile dopant-defect clusters, which is described by the DDC model. This is the third part of the PLS model. The model strongly depends on the nature of the dopant and therefore is presented separately for each type of dopant below. To activate the DDC model, add the `DDC` parameter to the `METHOD PLS` statement.

Boron

In the case of boron, these clusters are named boron interstitial clusters (BIC). The BIC species B_nI_m consists of *n* atoms of boron and *m* atoms of silicon self-interstitials. In absence of any direct experimental data concerning the exact composition of these clusters, BIC structure and charge states are chosen according to recent *ab-initio* theoretical calculation [21]. Various possible path are considered for these clusters: a given cluster can grow or dissolve by the addition or release of a silicon self interstitial or a boron-interstitial pair (Figure 3-2).

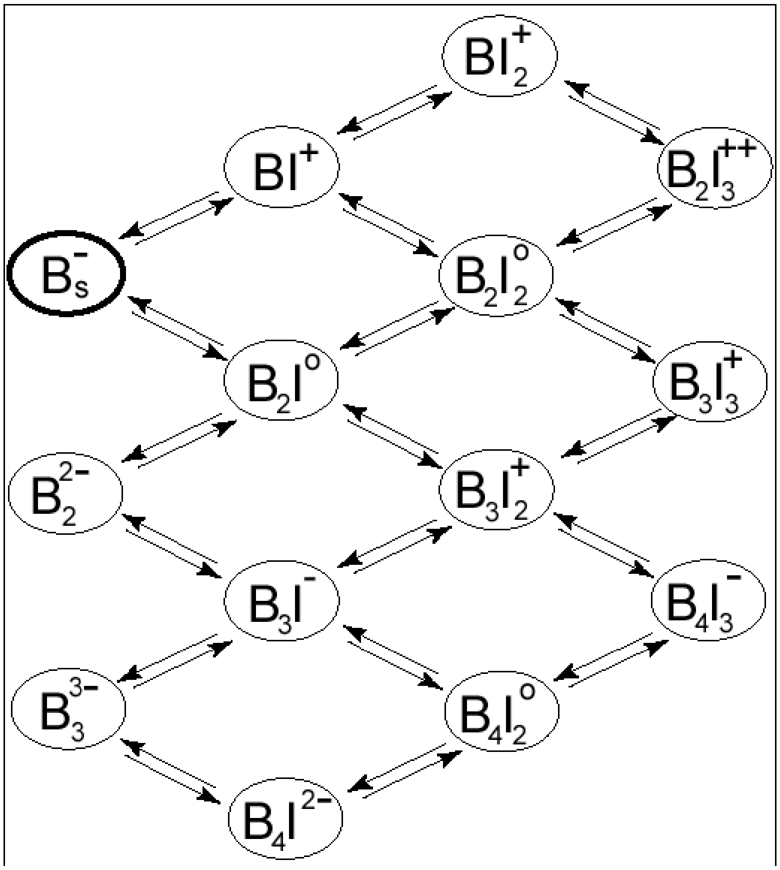


Figure 3-2 BIC reaction paths [5]

By default, the DDC model is based on the formation and dissociation of four BIC’s species: BI_2, B_2I, B_3I_2, and B_4I_3. Thus, the following reaction are added.

$$BI^{+} + I^{0} \overset{k^{f}_{BI_2}}{\underset{k^{r}_{BI_2}}{\Leftrightarrow}} BI_2^{+} \quad , \qquad B_s^{-} + BI^{+} \overset{k^{f}_{B_2I}}{\underset{k^{r}_{B_2I}}{\Leftrightarrow}} B_2I^{0}, \tag{3-120}$$

$$B_2I^{0} + BI^{+} \overset{k^{f}_{B_3I_2}}{\underset{k^{r}_{B_3I_2}}{\Leftrightarrow}} B_3I_2^{+} \ , \qquad B_3I_2^{+} + BI^{-} \overset{k^{f}_{B_4I_3}}{\underset{k^{r}_{B_4I_3}}{\Leftrightarrow}} B_4I_3^{0} \quad . \tag{3-121}$$

By default, the model assumes that dopants just after implantation are inactive. As this model does not assume any local equilibrium between each species, the activation of the dopant will gradually evolve with time.

The kinetic constants k^f and k^r for each reactions are defined as:

$$k^{f}_{B_nI_m} = 4\pi R_{eff} D_{BI}, \qquad k^{r}_{B_nI_m} = 6\frac{D_BI}{\lambda^2}\theta_n exp\left(-\frac{E_b(B_nI_m)}{kT}\right) \tag{3-122}$$

R_{eff} and $E_b(B_nI_m)$ are respectively the effective capture radius and the binding energy for each Boron-Interstitial Cluster and can be specified and modified in the `boron.mod` file.

Phosphorus

For phosphorus, the similar situation can be considered [25]. Therefore, phosphorus atoms can form clusters with self-interstitials. These clusters have been experimentally observed and are called phosphorus interstitials clusters (PIC).

The following reactions are taken into consideration.

$$P_s^{-} + I^{-} \overset{k^{f}_{P_2I}}{\underset{k^{r}_{P_2I}}{\Leftrightarrow}} P_2I^{0}, \qquad PI^{0} + I^{0} \overset{k^{f}_{PI_2}}{\underset{k^{r}_{PI_2}}{\Leftrightarrow}} PI_2^{0}, \tag{3-123}$$

These and additional reactions and their parameters can be specified in the `phosphorus.mod` file.

Arsenic

The case of arsenic is a little bit different. Arsenic migrates through the vacancy and interstitial mechanisms with roughly the same proportion. It is well known that arsenic can form clusters with vacancies of type As_nV. Therefore, the DDC model needs to take into account the following reactions:

$$As_s^+ + AsV^- \underset{k^r_{As_2V}}{\overset{k^f_{As_2V}}{\Leftrightarrow}} AsV^0, \qquad As_2V^0 + AsV^0 \underset{k^r_{As_3V}}{\overset{k^f_{As_3V}}{\Leftrightarrow}} As_3V^0 + V^0, \tag{3-124}$$

$$As_3V^0 + AsV^0 \underset{k^r_{As_4V}}{\overset{k^f_{As_4V}}{\Leftrightarrow}} As_4V^0 + V^0 \tag{3-125}$$

The system of equations for DDC model is

$$\begin{aligned}
\frac{\partial I^{tot}}{\partial t} &= -\nabla\cdot \boldsymbol{J}_I + GR_{IV} + GR_{AI} + GR_{AV-I} + GR_{AIC},\\
\frac{\partial V^{tot}}{\partial t} &= -\nabla\cdot \boldsymbol{J}_V + GR_{IV} + GR_{AV} + GR_{AI-V} + GR_{AVC},\\
\frac{\partial AI^{tot}}{\partial t} &= -\nabla\cdot \boldsymbol{J}_{AI} - GR_{AI} + GR_{AI-V} + GR_{AIC},\\
\frac{\partial AV^{tot}}{\partial t} &= -\nabla\cdot \boldsymbol{J}_{AV} - GR_{AV} + GR_{AV-I} + GR_{AVC},\\
\frac{\partial A_s}{\partial t} &= -\nabla\cdot \boldsymbol{J}_{AV} + GR_{AV} - GR_{AI-V} - GR_{AV-I} + GR_{AIC} + GR_{AVC},\\
\frac{\partial AIC}{\partial t} &= GR_{AIC},\\
\frac{\partial AVC}{\partial t} &= GR_{AVC},
\end{aligned} \tag{3-126}$$

where GR_{AVC}, GR_{AIC} are generation-recombination terms caused by reactions (Equations 3-120 and 3-125).

3.2.6 Typical Examples

The following will show each part of the PLS model validated using specific experimental results.

The CDD model is tested using a simple predeposition step. The interstitial cluster part of the model is validated using the Cowern's experiment [23], and the mixed cluster part of the model is then analyzed using the Pelaz experiment [27]. To illustrate the improvements given by the PLS model, we perform simulations within a very broad range of experimental conditions from a standard implantation and diffusion step to a "state of the art" RTA

Predeposition

Boron

As a first indication that the PLS model is able to handle the complex couplings between boron and the free point defects, we show in Figure 3-3 the result of the conventional predeposition steps simulation (diffusion with a constant surface concentration).

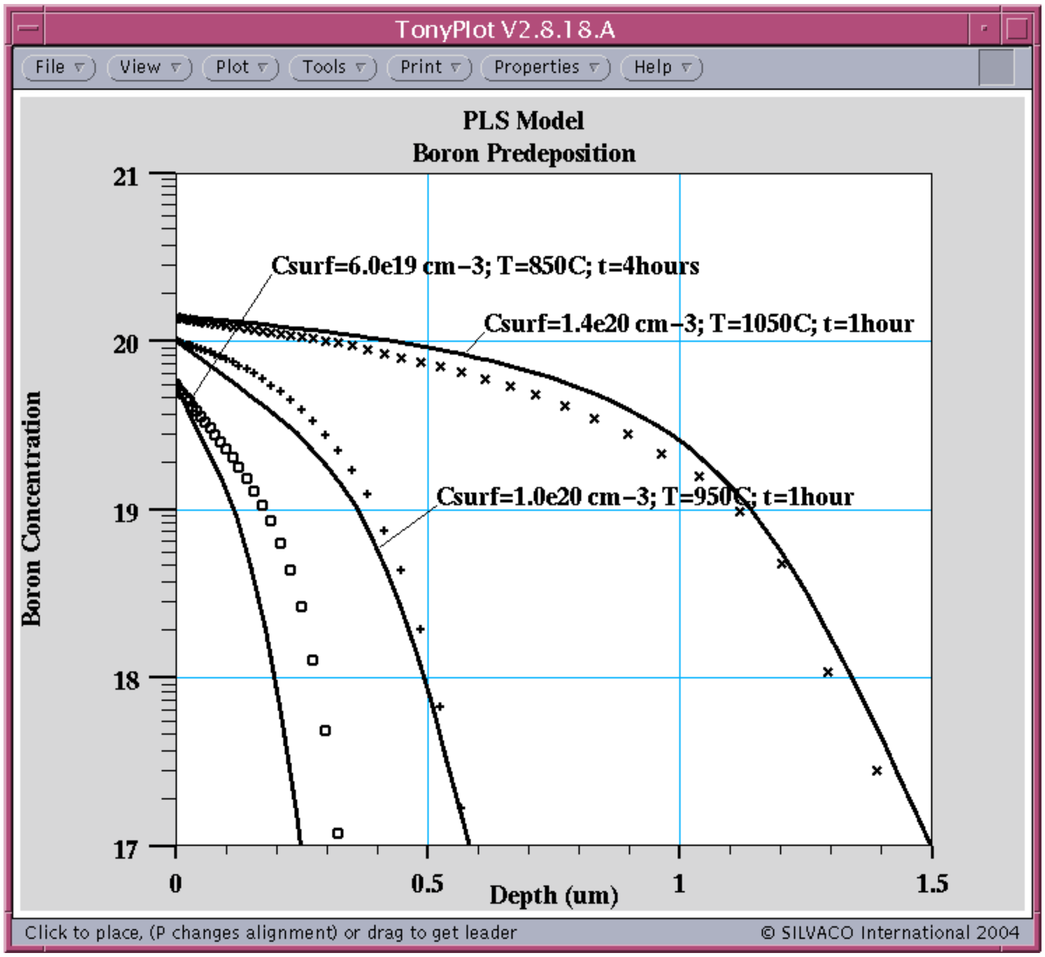

TonyPlot V2.8.18.A
File
View
Plot
Tools
Print
Properties
Help
PLS Model
Boron Predeposition
Boron Concentration
21
20
19
18
17
0
0.5
1
1.5
Depth (um)
Csurf=6.0e19 cm−3; T=850C; t=4hours
Csurf=1.4e20 cm−3; T=1050C; t=1hour
Csurf=1.0e20 cm−3; T=950C; t=1hour
Click to place, (P changes alignment) or drag to get leader
© SILVACO International 2004

Figure 3-3 Simulation of Boron predeposition using the CDD model at various temperature, time annealing and surface concentration. (Crosses, pluses, and squares are experimental data from [28])

Although not corresponding to the modern deep sub-micron technologies, this simulation represents the high dopant concentration features that reveal the complex couplings between dopant and point defects. Therefore, it is considered as a meaningful basic test for any advanced diffusion models.

Although this model has been developed for advanced silicon technologies, it still can be used as the standard diffusion model for any diffusion step. For example in the case of buried layer formation, the TED phenomena become irrelevant. Therefore, you can use only CDD part of the PLS model while ignoring IC and DDC models.

Phosphorus

To illustrate the improvements that result from the CDD model, we show simulations of phosphorus predeposition profiles at high and intermediate concentrations. The simulation results are compared to the SIMS data of Yoshida and Matsumoto. This experiment represents high dopant concentration features that reveal the complex couplings between dopants and point defects. Therefore, it is considered as a meaningful test for advanced diffusion models.

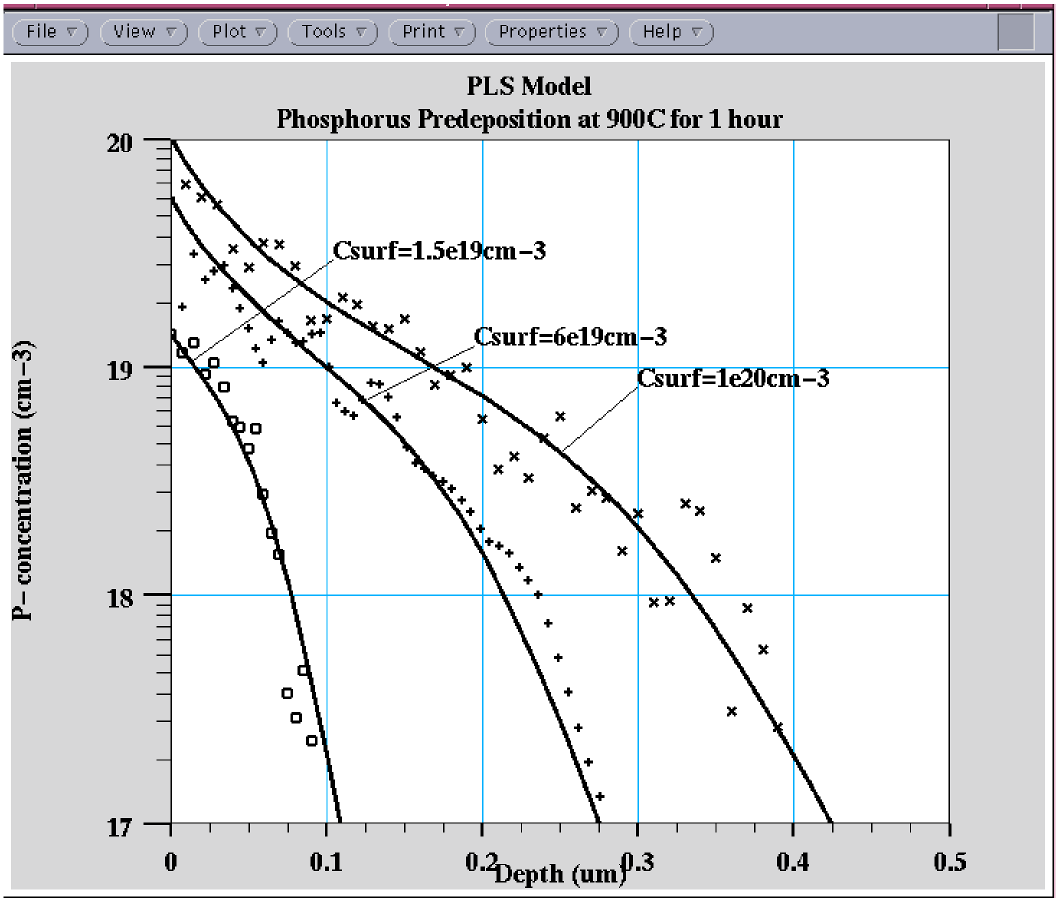


Figure 3-4 Simulation of phosphorus predeposition at 900°C during 1 hour with various surface concentration. (Squares, pluses, and crosses are experimental data from [29])

As our simulations prove, the PLS model accurately reproduces the experimental profiles features. Particularly, the simulated profiles exhibit the enhanced tail with more or less pronounced inflexion in the surface region. This inflexion is the result of the strong coupling between the defect gradients and the dopant.

Cowern’s Experiment

To validate the IC model, we first try to compare our model prediction with the experimental data obtained by Cowern *et al.* [23]. Briefly described, this experiment consists of observing the diffusivity of two boron marker layers after a silicon implantation at 40 keV to a dose of $2 \cdot 10^{13}$ atoms/cm^2 (simulated in the left hand side from Figure 3-5). From these observations, Cowern *et al.* have estimated the diffusion enhancement and the evolution of supersaturation with time. On the right hand side of Figure 3-5, the PLS simulation of the supersaturation evolution is shown.

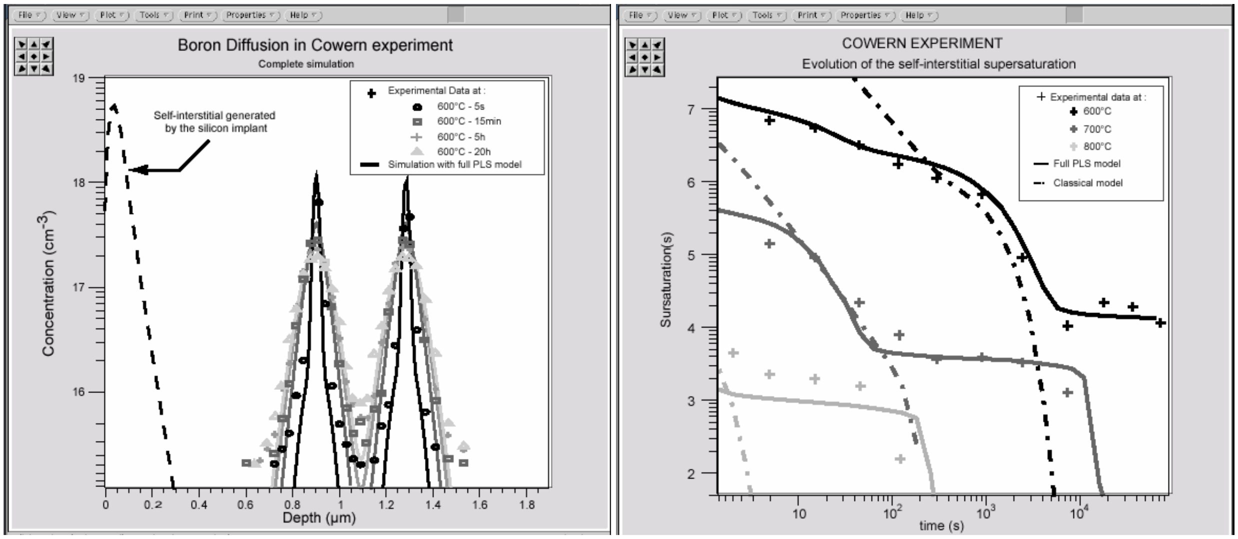


Figure 3-5 Simulation of the Cowern experiment and extraction of the evolution of the supersaturation during the annealing. (Experimental data are from [23])

Figure 3-5 shows the free interstitial supersaturation behavior with time. This quantity is simply calculated using the ratio between local effective interstitial concentration and its equilibrium value. From the experimental point of view, this quantity can be related to the boron diffusion enhancement with respect to its thermal equilibrium diffusion.

The supersaturation evolution curve exhibits three parts:

- The first step characterized by a high supersaturation value corresponds to a large acceleration of the dopant diffusion and can be explained by the presence of small clusters.
- The second step exhibits a plateau slightly decreasing with time. It is explained by the competitive growth between <311> defects known as the Ostwald ripening phenomena.
- The third step characterized by the supersaturation collapse is explained by the entire dissolution of the IC population, due to the recombination at the surface.

It is clear that the CDD model alone (dashed lines in Figure 3-5) cannot reproduce these three steps of the transient enhanced diffusion, since the curves are monotonically decreasing.

Pelaz Experiment

In this experiment, a boron marker is deposited at a depth of 0.15 µm. To observe the boron diffusion, Pelaz *et al.* have performed a silicon implant to generate a high interstitial concentration at the surface. Unlike the Cowern experiment, the boron concentration is high enough to allow the formation of BICs. Thus, this experiment exhibits a particular effects of in the boron diffusion: an immobilization and disactivation of the dopant at high concentrations even under the solid solubility limit.

Figure 3-6 demonstrates simulations of this particular experiment, using various parts of the PLS model.

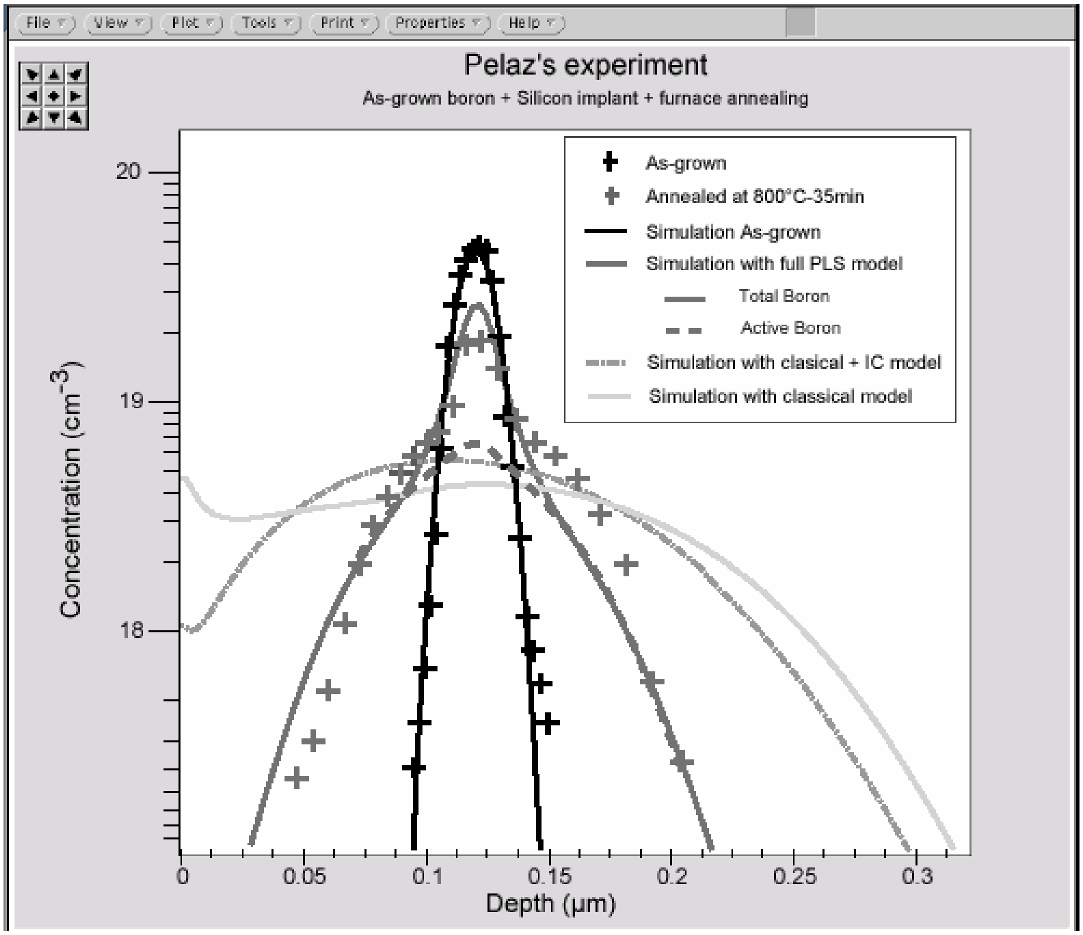


Figure 3-6 Simulation of the Pelaz experiment using various parts of the PLS model. (Experimental Data are extracted from [27])

The results obtained with the full PLS model correspond to the experimental data. As expected, the boron at the highest concentration stay immobile and are still inactive due to the formation of mixed BICs. Moreover, it is clear that only the full model can explain this behavior. Therefore, CDD and IC models cannot simulate this particular phenomenon without DDC model involved.

Implantation Diffusion Experiment

The analysis above proves validity of each part and the entire PLS diffusion model. In addition, we present several simulations relevant to specific processes important in the modern VLSI technologies.

Experiment with boron implanted at 20 keV

Boron implantation at 20 keV is widely used in silicon technologies. Since the experiment setup involves low temperature, such as 800°C, the effects of boron immobilization and disactivation at high concentration take place. Therefore, you must use three parts of the PLS model.

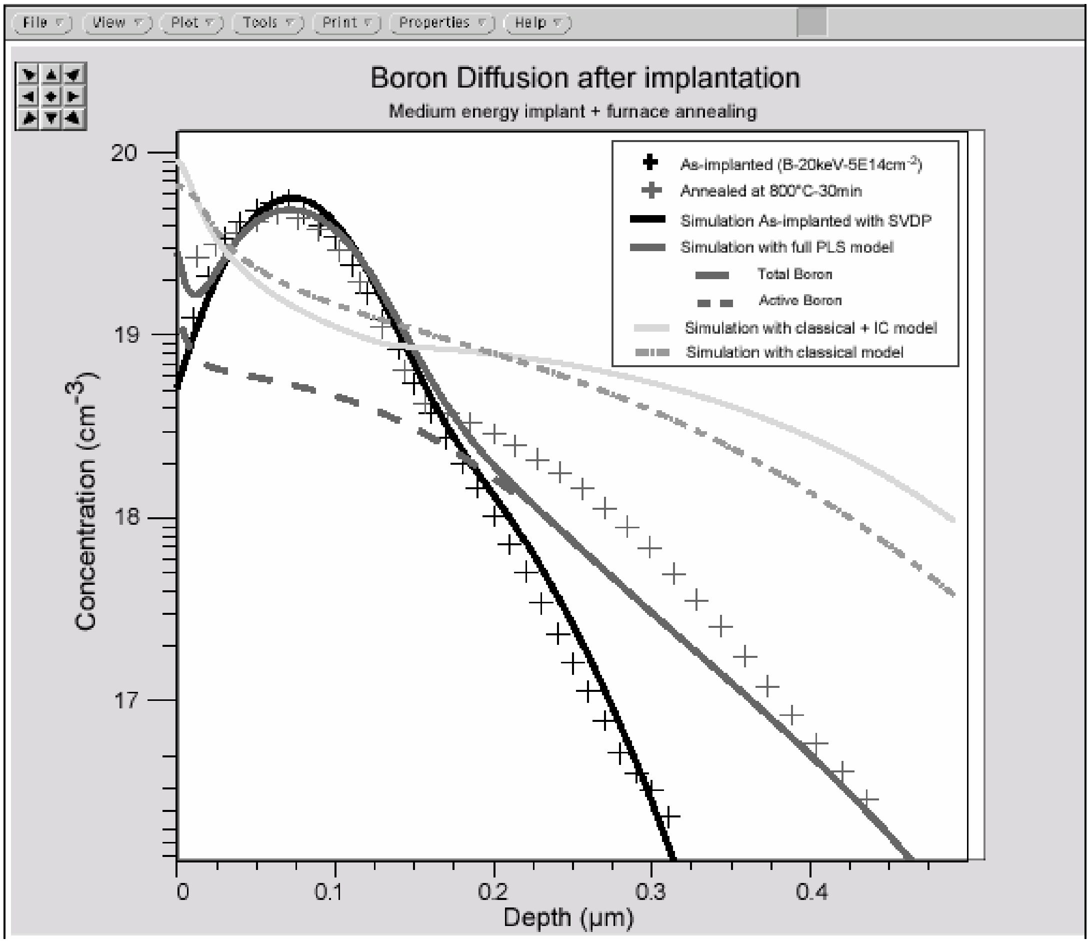


Figure 3-7 Simulation of 35 minutes boron diffusion at 800°C after an implantation at 20 keV with a dose of 5 · 10^{14} cm^{-2}. (Experimental data are extracted from [30])

As shown in Figure 3-7, PLS model results in an excellent fit with the experimental annealed curves. The comparison of chemical and electrically active boron concentration shows that even below the solid solubility, most of the dopant stay inactive due to the formation of mixed dopant-defect clusters. By comparing results obtained with only CDD or IC model with the full PLS (IC + DDC), it can be concluded that this type of clusters also consume free interstitials which leads to reduction of the boron diffusion. This fact justifies one more time that all three parts of the model: CDD, interstitial clusters and mixed dopant-defect clusters are needed in simulation of post-implant diffusion.

Experiment with boron implanted at 2 keV

A most aggressive technology simulation has been carried out with a 2 keV boron implant with a dose of 10^{14} cm^{-2} followed by a RTA at 1000°C during 10 seconds. This type of simulation is difficult because it needs to take into account a large number of phenomena including strong defect recombination at the surface, and the fact that TED duration is function of the implant setup and subsequent diffusion duration. Despite of these difficulties, Figure 3-8 shows a very good fit with experimental annealed data which proves the excellent quality of the model.

Nevertheless, as presented in the literature, the TED effect decreases at very low implant energy and high temperature annealing. This explain why the simulations done using CDD and IC models and only CDD model sometimes give reasonable agreement with experiments.

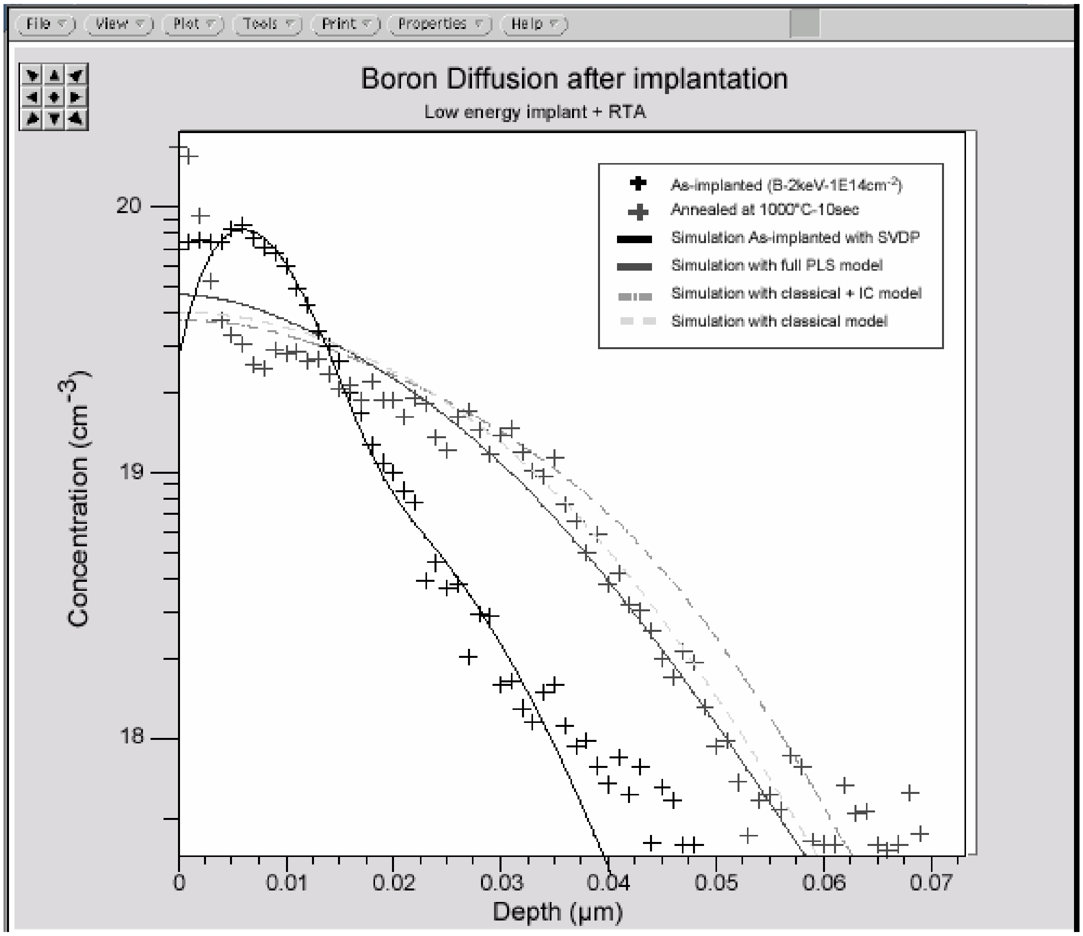


Figure 3-8 Simulation of boron diffusion at 1000°C during 10 s after an implantation at 2 keV with a dose of 1 · $10^{14}$$cm^{-2}$. (Experimental data are from [31])

Experiment with arsenic implanted at 2 keV

The PLS model is not only design for the boron diffusion but also for other common dopant, such as phosphorus or arsenic.

An Arsenic implantation has been performed with a energy of 2 keV at a dose of 10^{15} at/cm^{-2} and followed by a spike RTA at 950°C with a ramp up estimated at 100°C/s. In this typical process example for ultra shallow junction, dopant is implanted at such a high dose that its concentration reaches the solid solubility limit. In this case, most of the dopant at concentration higher that this limits will precipitate in the early stage of the annealing.

Moreover, all other specific characteristics in arsenic diffusion is taken into account. In other words:

- Arsenic migrate both through interstitial and vacancy mechanisms with roughly the same proportion.
- Arsenic atoms form with vacancy any clusters, As_2V or As_4V.

The initial interstitial profile generated by the arsenic implant is modeled though a simple "plus n" model.

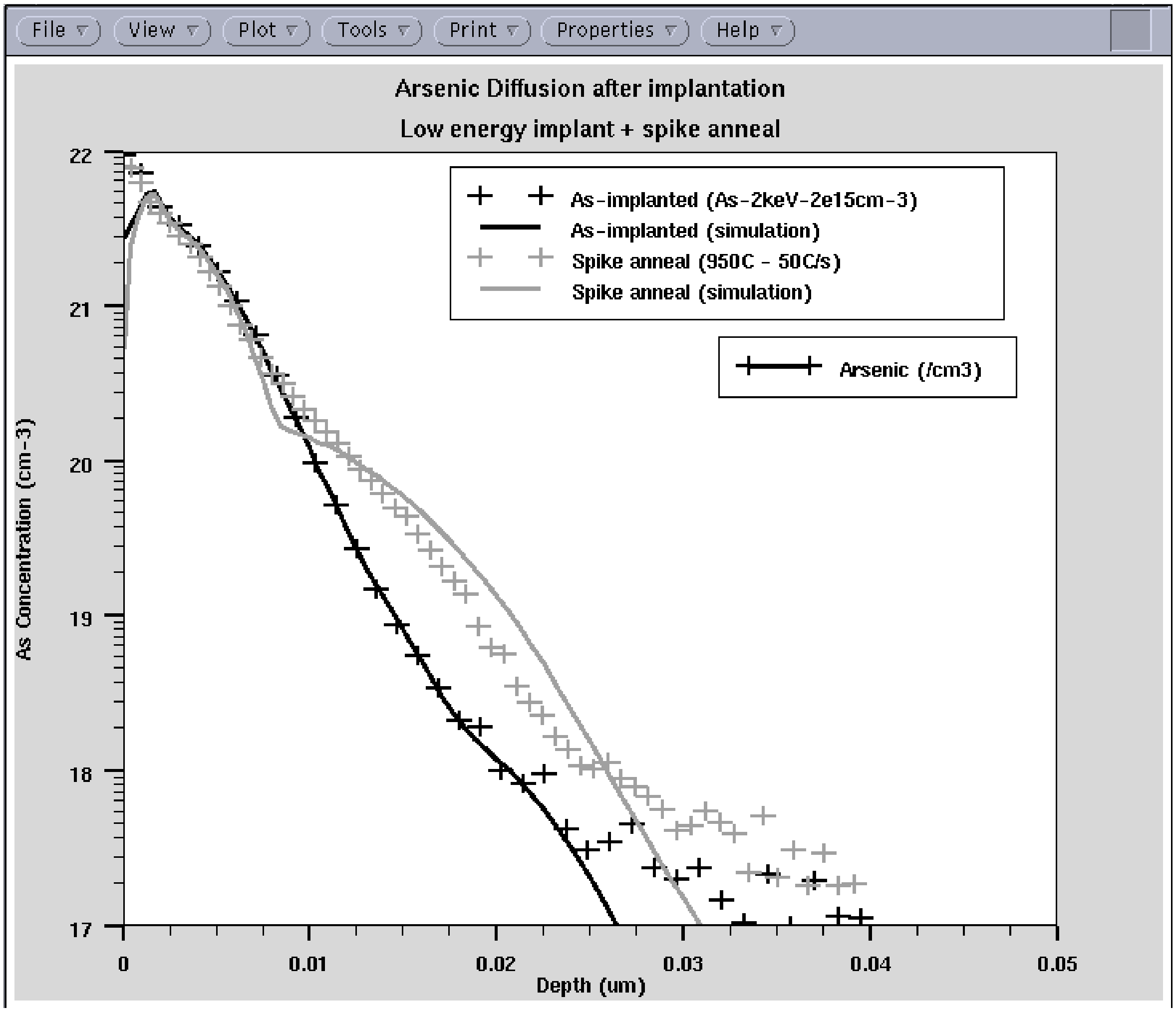


Figure 3-9 Simulation of Arsenic diffusion after an implantation at 2 keV with a dose of 1 · 10^{14} cm^{-2} and a spike anneal at 950°C with a temperature ramp rate estimated at 100°C/s. (Experimental data are from [32])

The result obtain with the full PLS model is in good agreement with experimental data. As expected, most of the arsenic at concentration above the solid solubility limit precipitate quickly and consequently immobilize the dopant.

3.3 Oxidation Models

The fabrication of integrated circuit microelectronic structures and devices vitally depends on the thermal oxidation process for the formation of gate dielectrics, device isolation regions, spacer regions, and ion implantation mask regions. Particularly, the precise control of silicon dioxide thickness as device geometries continue to scale to sub-micron dimensions.

In SSuprem4 silicon thermal oxidation is modeled when a `DIFFUSION` statement contains a `DRYO2`, `WETO2`, `F.O2`, or `F.H2O` parameter. Oxidation takes place when there is an interface between silicon (or polysilicon) and silicon dioxide or a silicon (polysilicon) surface is exposed to an oxidizing ambient. SSuprem4 simulates polysilicon oxidation in a very similar manner as silicon (almost all oxidation parameters for polysilicon are the same as for silicon). SSuprem4 also allows oxidation completely through a silicon (polysilicon) layer. This is very important in processes (e.g., poly buffered `LOCOS`) in which polysilicon regions are completely consumed during oxidation.

Because exposed silicon surfaces usually have a thin native oxide layer, SSuprem4 automatically deposits a thin native oxide layer on all exposed silicon (polysilicon) surfaces at the beginning of oxidation steps. The `INITIAL` parameter in the `OXIDE` statement determines the layer's thickness, which has a default value of 20 Å.

The two-dimensional oxidation models in SSuprem4 are based on the well-known linear-parabolic theory of Deal and Grove [33]. Numerical aspects of the model implementation can be found in [34]. Silicon oxidation is modeled by considering the following three processes:

1. Oxidant (e.g., H_2O or O_2) is transported from the ambient gas into the SiO_2 layer at the gas/SiO_2 interface.
2. Oxidant is transported across the SiO_2 layer until reaching the Si/SiO_2 interface.
3. Oxidant, arriving at the Si/SiO_2 interface, reacts with silicon to form a new layer of SiO_2.

The transport of oxidant across the gas/SiO_2 interface is given by

$$\boldsymbol{F_1} = h(C^* - C_0)\boldsymbol{n_0} \qquad 3\text{-}127$$

where:

- h is the gas-phase mass-transport coefficient.
- C^* is the equilibrium oxidant concentration in SiO_2.
- C_0 is the oxidant concentration in SiO_2 at the gas/SiO_2 interface.
- $\boldsymbol{n_0}$ is a unit vector normal to the gas/SiO_2 interface pointing toward the silicon layer.

The equilibrium oxidant concentration in SiO_2 is linearly related to the partial pressure of the oxidant, P, in the gas by Henry's law:

$$C^* = K \cdot P \qquad 3\text{-}128$$

where K is a constant.

Diffusion of oxidant molecules in the SiO_2 is driven by a concentration gradient and is given by Fick's law as

$$\boldsymbol{F_2} = -D_{eff}\nabla C \qquad 3\text{-}129$$

where D_{eff} is the effective oxidant (H_2O or O_2) diffusivity in the growing SiO_2 layer. C is the oxidant concentration in the oxide.

The reaction at the Silicon (or Polysilicon)/SiO_2 interface between silicon and the oxidant is expressed as

$$\boldsymbol{F_3} = kC_i\boldsymbol{n_i} \qquad \text{3-130}$$

where:

- k is the apparent surface reaction rate constant.
- C_i is the oxidant concentration at the Silicon (or Polysilicon)/SiO_2 interface.
- $\boldsymbol{n_i}$ is a unit vector normal to the Si/SiO_2 pointing toward the silicon layer.

Under steady state conditions, the three fluxes are equal.

$$\boldsymbol{F} = \boldsymbol{F_1} = \boldsymbol{F_2} = \boldsymbol{F_3} \qquad \text{3-131}$$

By dividing the flux by N_1, the number of oxidant molecules incorporated in a unit volume of SiO_2, and considering one dimensional growth, the growth rate of the oxide layer is given by

$$\frac{dx_0}{dt} = \frac{F}{N_1} \qquad \text{3-132}$$

where x_0 is the oxide thickness. From Equation 3-127 and Equations 3-129–3-131, Equation 3-132 can be expressed as

$$\frac{dx_0}{dt} = \frac{B}{A + 2x_0} \qquad \text{3-133}$$

where:

$$A = 2D_{eff}\left(\frac{1}{k} + \frac{1}{h}\right) \qquad \text{3-134}$$

$$B = 2D_{eff}\frac{C^*}{N_1} \qquad \text{3-135}$$

Equation 3-133 is modified for thin oxides (less than 500 Å) as follows:

$$\frac{dx_0}{dt} = \frac{B}{A + 2x_0} + R \qquad \text{3-136}$$

where R is calculated according to [35]:

$$R = THINOX \cdot 0 exp\left(\frac{THINOX \cdot E}{kT}\right) exp\left(\frac{x_0}{THINOX \cdot L}\right) \cdot P^{THINOX \cdot P} \qquad \text{3-137}$$

where *P* is partial pressure. `THINOX.0`, `THINOX.E`, `THINOX.L`, and `THINOX.P` parameters are specified in the `OXIDE` statement.

3.3.1 Numerical Oxidation Models

In the previous section, an introduction to one-dimensional oxidation modeling was presented. This section describes the two-dimensional numerical oxidation models implemented in SSuprem4. The numerical oxidation models build on the Deal-Grove oxidation theory and provide the capability to simulate arbitrary two-dimensional structures.

The numerical oxidation models require solving the oxidant diffusion equation at incremental time steps at discrete grid points in the growing SiO_2 layer. The oxidant diffusion equation is given by

$$\frac{\partial C}{\partial t} = \nabla \cdot \boldsymbol{F} \tag{3-138}$$

where:

- C is the oxidant concentration in SiO_2.
- *t* is the oxidation time.
- ***F*** is the oxidant flux.

Equation 3-138 is solved by substituting Equation 3-129 for ***F*** and defining appropriate boundary conditions at material interfaces with SiO_2. At the gas/SiO_2 interface, Equation 3-127 describes the interface transport flux of oxidant molecules, accounting for the boundary condition at that interface. The boundary condition at the Silicon (or Polysilicon)/SiO_2 interface is described by Equation 3-130. The flux at boundaries between SiO_2 and other materials in the simulation structure is set to zero. By solving Equation 3-138, the oxidant concentration is determined at each grid point in the SiO_2 layer. The SiO_2 growth rate or Si/SiO_2 interface velocity, $\boldsymbol{V_s}$, is determined at each point along the interface by combining Equations 3-130 and 3-132 resulting in the following.

$$\boldsymbol{V}_s = \frac{kC_i\boldsymbol{n}_i}{N_1} \tag{3-139}$$

At each time step, Equation 3-139 is solved. The incremental oxide thickness grown is calculated by multiplying Equation 3-139 by the time step. During the oxidation reaction silicon atoms bond with the oxidant to form the SiO_2 compound. Thus, silicon material is removed during the oxidation process. The ratio of the silicon thickness consumed to form a given thickness of SiO_2 is specified using the `ALPHA` parameter in the `OXIDE` statement.

Equation 3-138 is sufficient to describe the motion of the Si/SiO_2 interface if the oxide flow is in the same direction as the growth (for planar oxidation structures). In most structures of interest, the oxide flow is two dimensional. Therefore, additional equations have to be solved. Both the Compress and Viscous models calculate the two-dimensional flow of oxide elements by solving a simplified hydrodynamic creeping flow equation.

3.3.2 Compress Model

In addition to solving Equations 3-138 and 3-139, a simplification of the hydrodynamic flow equation is solved to obtain the flow of oxide elements [37]. The Compress Model is activated by specifying `COMPRESS` in the `METHOD` statement prior to a `DIFFUSE` statement. The Compress Model is the default oxidation model in SSuprem4. Neglecting the acceleration and gravitational terms in the hydrodynamic flow equation, the creeping-flow equation is given by

$$\mu \nabla^2 \mathbf{V} = \nabla P \qquad \text{3-140}$$

where P is the hydrostatic pressure, $\mathbf{V}$ is velocity of oxide elements and μ is the oxide viscosity.

The oxide viscosity is calculated from the following equation:

$$\mu = \frac{E}{2(1+\nu)} \qquad \text{3-141}$$

where Young's modulus E and Poisson's ratio ν are specified through elastic constants as described in Section 3.10 "Stress Models".

The oxide flow is treated as an incompressible fluid. By doing this, it implies the density of the oxide is constant with respect to time. Applying this fact to the mass continuity equation, the incompressibility condition is given as

$$\nabla \cdot V = 0 \qquad \text{3-142}$$

The incompressibility condition in Equation 3-142 is implemented by allowing a slight compressibility of the flowing oxide. Thus, Equation 3-142 is modified to give the following equation:

$$\nabla \cdot V = \frac{1-2\nu}{\mu} P \qquad \text{3-143}$$

The solution of Equation 3-143 at each time step gives the velocity field of the flowing oxide elements. The Compress Model is recommended for simulations of planar and non-planar structures, where stress effects play a minor role in determining the oxide shape. When stress effects are important, you can use the Viscous oxidation model.

Figure 3-10 shows a two-dimensional cross section of the structure resulting from a LOCOS oxidation using the Compress Model.

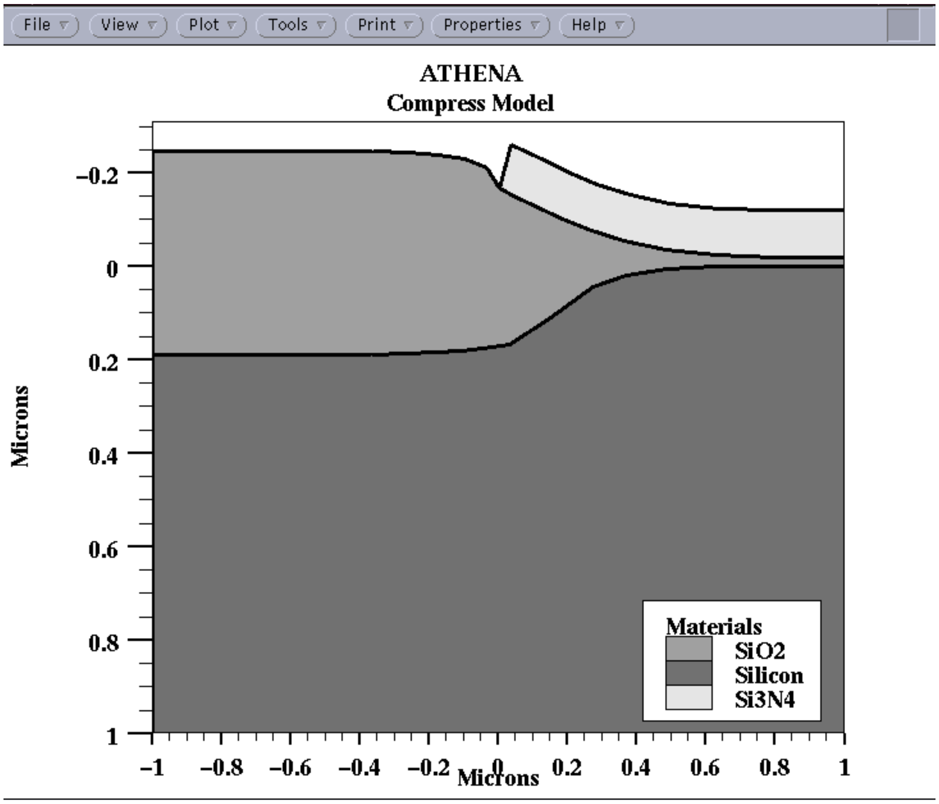


Figure 3-10 Resulting Structure from a LOCOS Oxidation step using the Compress Model

3.3.3 Viscous Model

The Viscous Model solves the same flow equations as described in the previous section. This model is activated by specifying the `VISCOUS` parameter in the `METHOD` statement prior to the `DIFFUSE` statement. The Viscous Model calculates stresses in the growing oxide and creates almost the same shape for the silicon/oxide interface as does the Compress model.

The stresses in the oxide are calculated as follows:

$$\sigma_{xx} + \sigma_{yy} = \frac{2 \cdot VISC.0 \cdot exp\left(\frac{-VISC.E}{kT}\right)}{1 - 2 \cdot \nu}\left(\frac{\partial v_x}{\partial x} + \frac{\partial v_y}{\partial y}\right) \tag{3-144}$$

$$\sigma_{xx} - \sigma_{yy} = 2 \cdot VISC.0 \cdot exp\left(\frac{-VISC.E}{kT}\right)\left(\frac{\partial v_x}{\partial x} - \frac{\partial v_y}{\partial y}\right) \tag{3-145}$$

$$\sigma_{xy} = VISC.0 \cdot exp\left(\frac{-VISC.E}{kT}\right)\left(\frac{\partial v_x}{\partial x} + \frac{\partial v_y}{\partial y}\right) \tag{3-146}$$

where v_x and v_y are the x and y components of flow velocity $\boldsymbol{v}$ respectively. VISC.0 and VISC.E are the pre-exponential and activation energy respectively for viscosity are specified on the MATERIAL statement. ν is Poisson’s ratio specified through elastic constants as described in Section 3.10 “Stress Models”.

The stress-dependent nonlinear model based on Eyring’s work [36] allows a description of the real shape of LOCOS profiles with kinks on the interface. The model is turned on by specifying the STRESS.DEP parameter the OXIDE statement. Using Equation 3-140 and Equations 3-143–3-146, the non-linear solver first finds a linear solution for flow velocities and stresses and then uses the stresses obtained to calculate the reduction factors for oxidant diffusivity, D_{eff}, oxide viscosity, μ, and the interface reaction rate constant k as follows:

$$D^{(i)}_{eff} = D^{(i-1)}_{eff} \cdot exp\left(\frac{V_d(\sigma_{xx}+\sigma_{yy})}{kT}\right) \tag{3-147}$$

$$\mu^{(i)} = \mu^{(i-1)}\frac{\left(\frac{\tau V_c}{2kT}\right)}{sinh\left(\frac{\tau V_c}{2kT}\right)} \tag{3-148}$$

$$k^{(i)} = k^{(i-1)} \cdot exp\left(-\frac{\sigma_r V_r + \sigma_t V_t}{kT}\right) \tag{3-149}$$

where i is the iteration. V_d, V_c, V_r, and V_t are the activation volumes (in Å^3), specified in the OXIDE statement.

τ is the total shear stress:

$$\tau = \frac{1}{2}\sqrt{(\sigma_{xx}-\sigma_{yy})^2 + 4\sigma_{xy}^2} \tag{3-150}$$

σ_r is the normal component of the total stress:

$$\sigma_r = \sigma_{xx}n_x^2 + \sigma_{yy}n_y^2 + 2\sigma_{xy}n_x n_y \tag{3-151}$$

σ_t is the tangential component of the total stress:

$$\sigma_t = \sigma_{xx}n_y^2 + \sigma_{yy}n_x^2 + 2\sigma_{xy}n_x n_y \tag{3-152}$$

where n_x and n_y are the x and y components of the unit vector normal respectively.

The reduced parameters feed back to the next iteration. This process continues until the accuracy criterion is met. Fast convergence of this process is not guaranteed. Oxidation calculations by the stress-dependent model usually take much more CPU time than the Compress Model.

Figure 3-11 shows the resulting structure from a LOCOS oxidation step using the stress dependent Viscous Model.

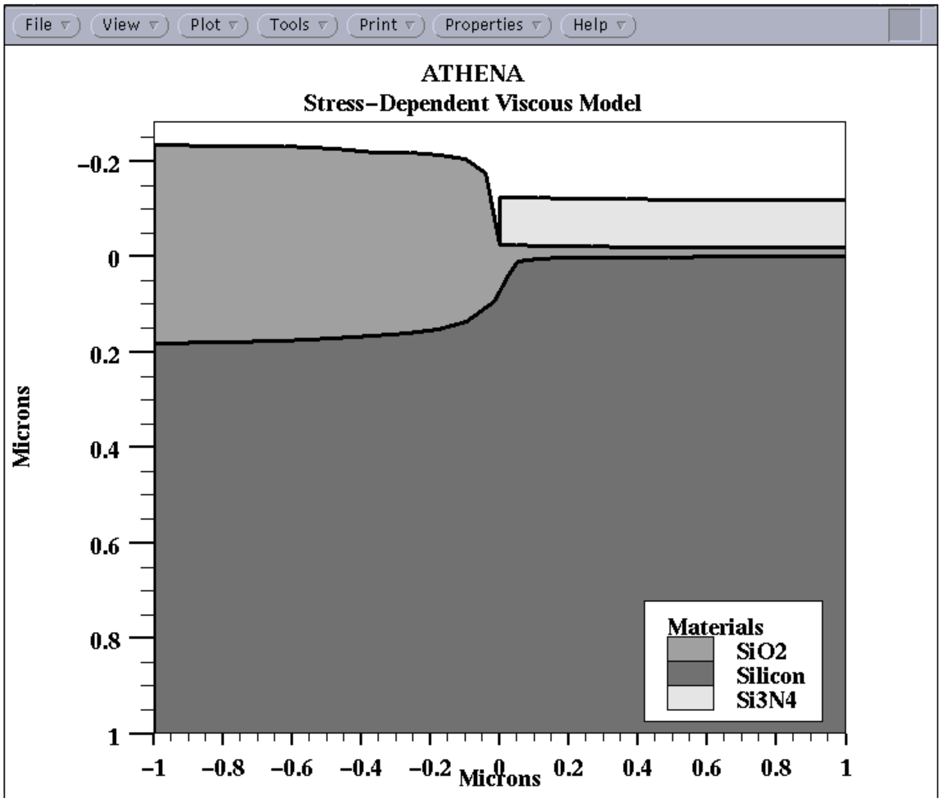


Figure 3-11 Resulting Structure from a LOCOS Oxidation step using the Stress-Dependent Viscous Model

3.3.4 Linear Rate Constant

For short oxidation times and low oxidation temperatures the oxide growth is linearly related to the oxidation time. The interface processes (oxidant transport across the gas/SiO_2 interface and oxidant reaction at the Si/SiO_2 interface) are the determining factor in describing the growth kinetics. In this regime, the oxide thickness can be approximated as

$$x_0 \cong \left(\frac{B}{A}\right)t \qquad 3\text{-}153$$

where (B/A) is called the linear rate constant and is obtained by dividing Equation 3-135 and Equation 3-134, resulting in the following equation:

$$\left(\frac{B}{A}\right) = \frac{C^*}{N_1}\left(\frac{1}{h}+\frac{1}{k}\right) \qquad 3\text{-}154$$

The equilibrium oxidant concentration in the oxide, C^*, is defined by Equation 3-128. K in Equation 3-128 is specified by the HENRY.COEF parameter in the OXIDE statement. The gas-phase mass-transport coefficient, h, is given by the following Arrhenius relation:

$$h = TRN.0 \cdot exp\left(-\frac{TRN.E}{k_b T}\right) \qquad 3\text{-}155$$

where the TRN.0 and TRN.E parameters are specified in the OXIDE statement. The interface reaction rate constant, k, is determined from Equation 3-154 and experimentally determined values of *(B/A)*.

The linear rate constant is composed of several dependencies including orientation, pressure, chlorine additions, and doping effects.

$$\left(\frac{B}{A}\right) = \left(\frac{B}{A}\right)_i \left(\frac{B}{A}\right)_{ori} \left(\frac{B}{A}\right)_P \left(\frac{B}{A}\right)_{Cl} \left(\frac{B}{A}\right)_{doping} \qquad 3\text{-}156$$

$(B/A)_i$ is given by:

$$\left(\frac{B}{A}\right)_i = \begin{cases} LIN.L.0\,exp\left(-\frac{LIN.L.E}{k_b T}\right) & T < L.BREAK \\ LIN.H.0\,exp\left(-\frac{LIN.H.E}{k_b T}\right) & (T \geq L.BREAK) \end{cases} \qquad 3\text{-}157$$

which is the linear rate constant determined for oxidations on lightly-doped substrates annealed at atmospheric pressure with no chlorine content in the ambient. The parameters appearing in Equation 3-157 are specified in the OXIDE statement. The remaining factors in Equation 3-156 are described in the following sections.

Orientation Dependence

The silicon substrate orientation is known to affect the oxidation kinetics [37, 38]. The influence of orientation on the linear rate constant is modeled as $(B/A)_{ori}$ in Equation 3-156. The orientation dependencies for <100> and <110> orientations are modeled by appropriate reduction factors, and $(B/A)_{ori}$ for <111> substrates is unity.

Figure 3-12 shows the silicon dioxide thickness dependence as a function of the substrate orientation for several oxidation temperatures.

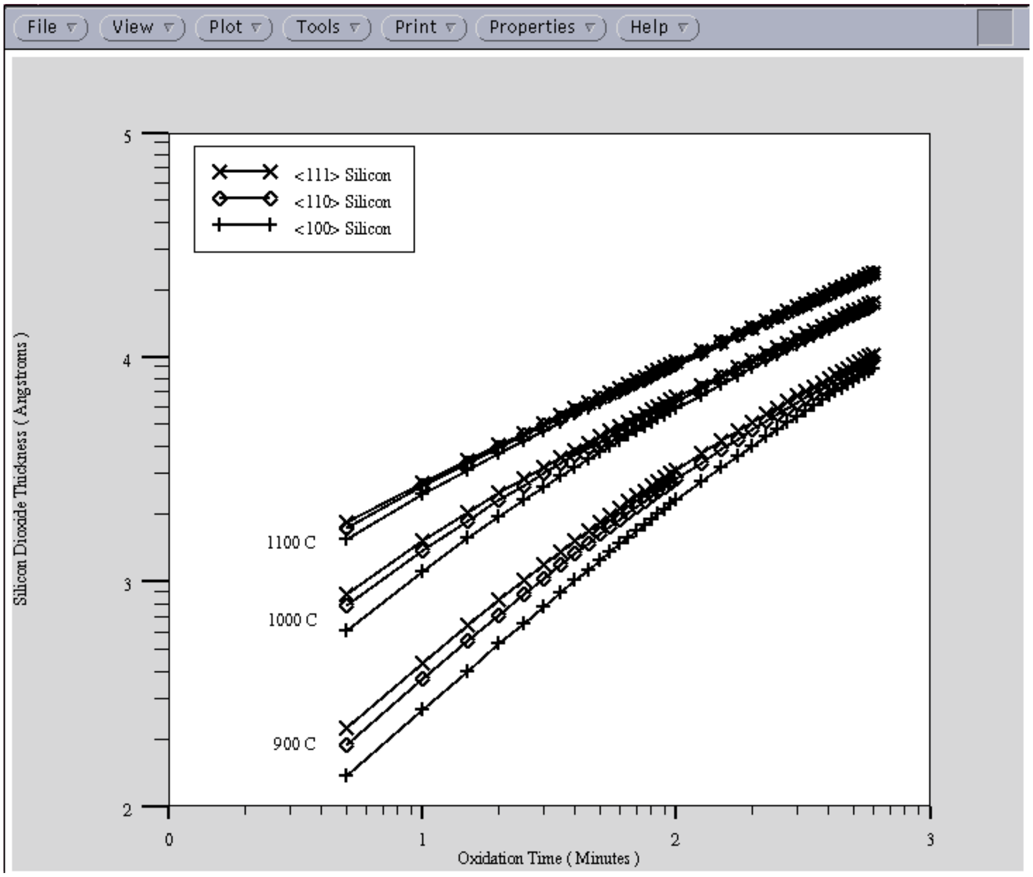


Figure 3-12 Silicon Dioxide Thickness versus Time for Different Substrate Orientations and Temperatures

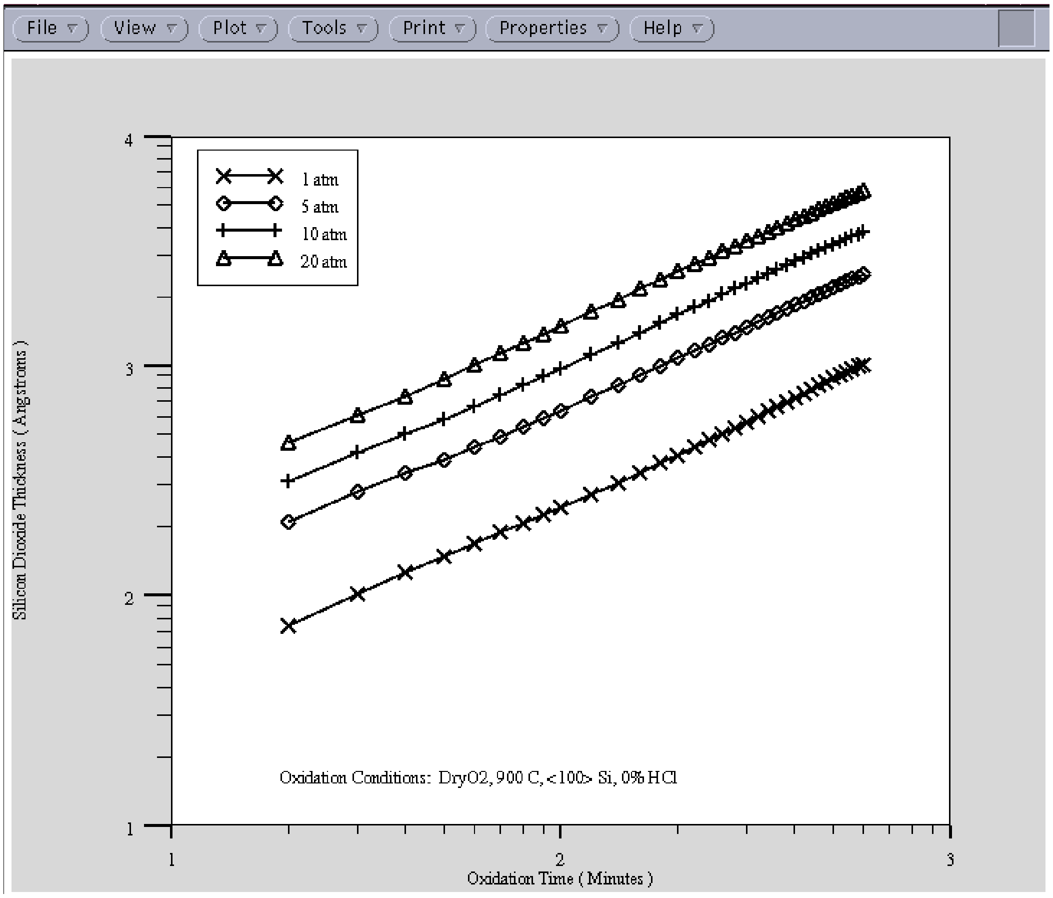
Pressure Dependence

High pressure silicon oxidation allows one to grow relatively thick SiO_2 films while keeping the temperature low so that dopant redistribution is reduced [39]. The pressure dependence in the linear rate constant is given by

$$\left(\frac{B}{A}\right)_P = P^{L.PDEP} \qquad 3\text{-}158$$

where L.PDEP is specified on the OXIDE statement for each oxidant and P is the partial pressure of the oxidizing gas.

Figure 3-13 shows the silicon dioxide thickness versus time with PRESSURE as a parameter.



Oxidation Conditions: DryO2, 900 C, <100> Si, 0% HCl

Figure 3-13 Silicon Dioxide Thickness versus Time with Pressure as a Parameter

Chlorine Dependence

The addition of chlorine to the oxidation system results in better passivation and higher oxide dielectric strength [33, 38]. For a dry oxygen ambient, chlorine introduction gives rise to a higher oxidation rate. It has been suggested [38] that chlorine reacts with O_2 to produce H_2O and Cl_2 as products. The oxidation rate is higher in H_2O ambients than in O_2 ambients because equilibrium concentration of H_2O in the oxide is higher. A look-up table approach is implemented to model the increase in the linear rate constant in Equation 3-156 though the $(B/A)_{Cl}$ term. The table gives an enhancement factor to the linear rate constant as a function of chlorine percentage and temperature. The default values for chlorine dependence are included in Appendix B “Default Coefficients”.

The effects of adding chlorine to the oxidizing ambient is shown in Figure 3-14, where the silicon dioxide thickness increases as more chlorine is added to the ambient.

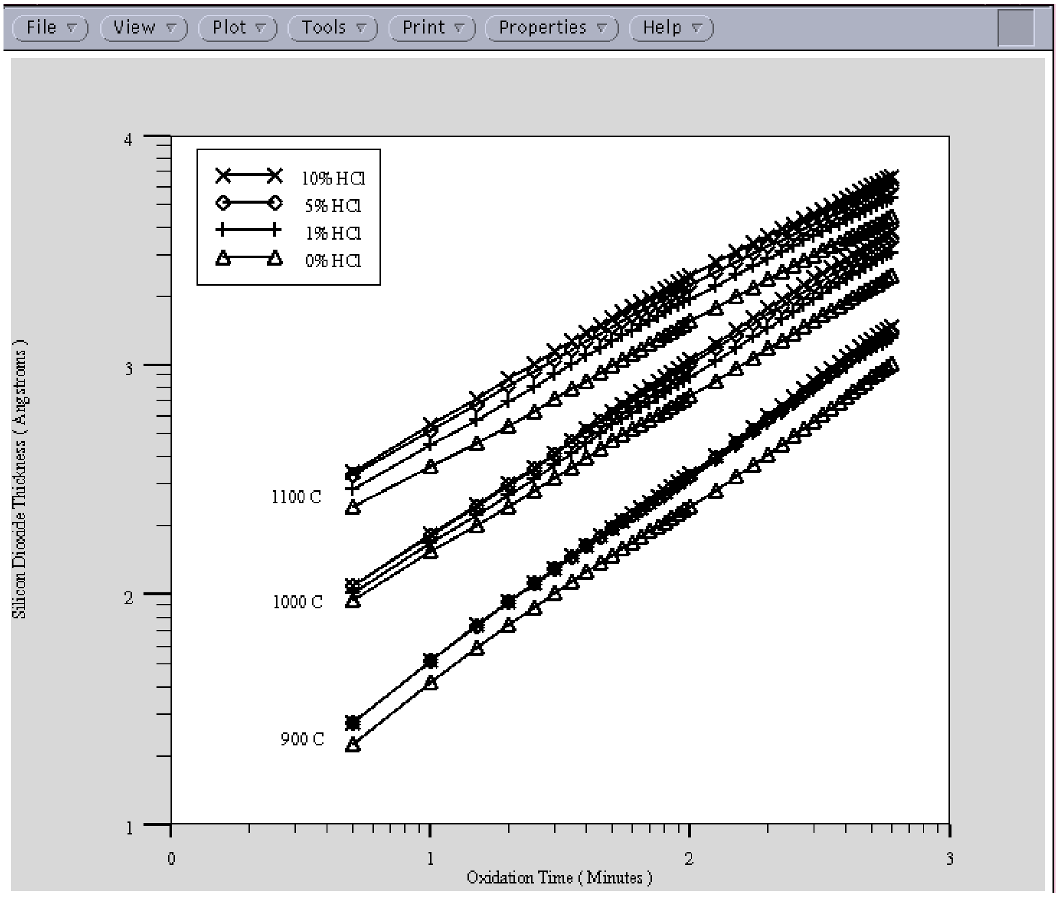


Figure 3-14 Silicon Dioxide Thickness Versus Oxidation Time with HCl Percentage and Temperature as Parameters

Doping Dependence

It is well known that SiO_2 formation on highly-doped n-type and p-type substrates can be enhanced compared to SiO_2 formation on lightly-doped substrates [40]. The dependence of silicon dioxide growth kinetics on doping concentration is manifested as part of the linear rate constant, where the physical significance of the high doping levels has been explained primarily as an electrical effect [40], [41]. This factor in the linear rate constant is given by

$$\left(\frac{B}{A}\right)_{doping} = \left[1 + BAF.K0 \cdot exp\left(\frac{-BAF.KE}{k_bT}\right)\left(\frac{V*}{V_i*} - 1\right)\right] \tag{3-159}$$

where $V*$ is the equilibrium vacancy concentration in silicon at the Si/SiO_2 interface. V_i^* is the equilibrium vacancy concentration in intrinsic silicon. `BAF.K0` and `BAF.KE` are specified on the `OXIDE` statement.

The equilibrium vacancy concentration, composed of vacancy defects in different charged states, depends on the Fermi level location [8], [42] and is given by

$$V* = V_i*\left\{\frac{1 + \left(\frac{n_i}{n}\right)\phi^{+} + \left(\frac{n_i}{n}\right)^2\phi^{++} + \left(\frac{n}{n_i}\right)\phi^{-} + \left(\frac{n}{n_i}\right)^2\phi^{=}}{1 + \phi^{+} + \phi^{++} + \phi^{-} + \phi^{=}}\right\} \tag{3-160}$$

where n is the electron concentration and n_i is the intrinsic carrier concentration, and

$$\phi^{+} = \sqrt{BAF.EBK}\ exp\left(-\frac{BAF.PE}{k_bT}\right) \tag{3-161}$$

$$\phi^{++} = BAF.EBK exp\left(-\frac{BAF.PPE}{k_bT}\right) \tag{3-162}$$

$$\phi^{-} = \sqrt{BAF.EBK}\ exp\left(-\frac{BAF.NE}{k_bT}\right) \tag{3-163}$$

$$\phi^{=} = BAF.EBK exp\left(-\frac{BAF.NNE}{k_bT}\right) \tag{3-164}$$

where ϕ^+, ϕ^{++}, ϕ^-, and $\phi^=$ are fractions of the vacancy concentration which are positively, double positively, negatively, and double negatively charged respectively.

Figure 3-15 shows a plot of V^*/V_i^* at 950°C for common silicon dopants. Notice that for n-type dopants (V^*/V_i^*) increases as the doping concentration increases, but V^*/V_i^* remains essentially constant for the p-type dopant. The increase in V^*/V_i^* for n-type dopants increases the linear rate constant. This ultimately leads to thicker oxides when oxidizing highly-doped n-type substrates due to a higher availability of unoccupied silicon lattice sites (vacancies) for oxidant molecules to be incorporated.

The oxide thickness trend is shown in Figure 3-16, where the SiO_2 thickness is plotted versus doping concentration for common silicon dopants.

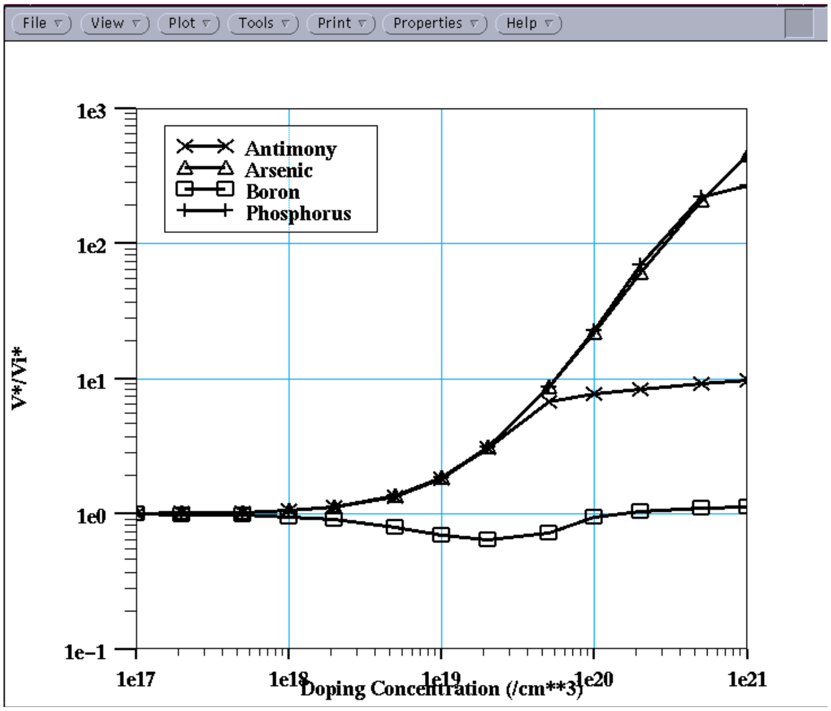


Figure 3-15 V^*/V_i^* **Ratio versus Doping Concentration**

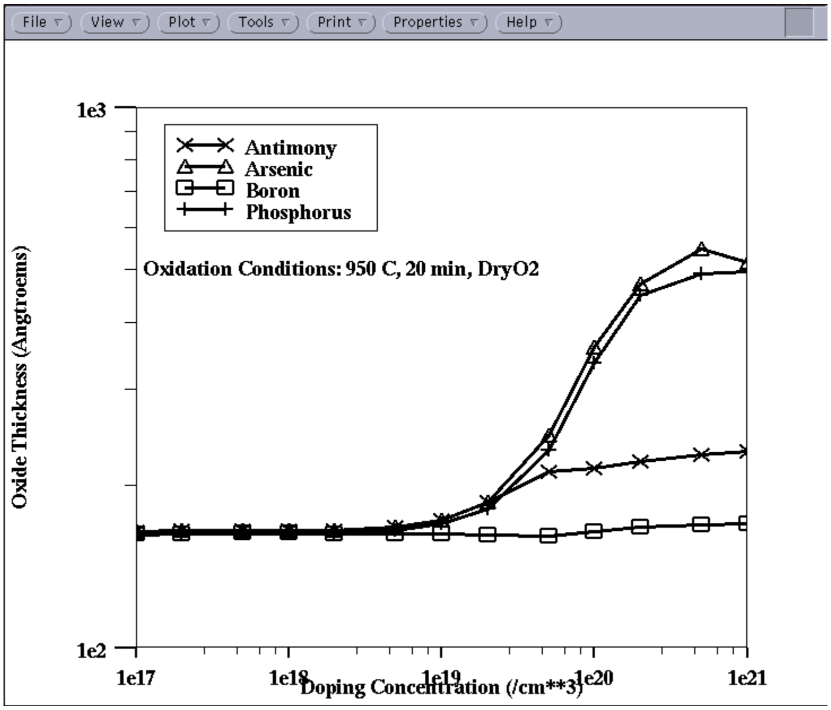


Figure 3-16 Simulated Silicon Dioxide Thickness vs. Doping Concentration for Common Silicon Dopants

3.3.5 Parabolic Rate Constant

For long oxidation times and high temperatures the oxide growth is parabolically related to the oxidation time. The diffusion of oxidant in the oxide is the determining factor in describing the growth kinetics. For these times and temperatures, the oxide thickness can be approximated as:

$$x_0^2 \cong Bt \qquad \text{3-165}$$

where B is called the parabolic rate constant and is given by Equation 3-135. When using this equation, the oxidant diffusivity, D_{eff}, is determined from specified values for C^*, N_1, and experimentally-determined values of B. The parabolic rate constant has been determined to have dependencies on the ambient pressure and the chlorine content during oxidation and is given by

$$B = B_i \cdot B_P \cdot B_{HCl} \qquad \text{3-166}$$

where:

$$B_i = \begin{cases} PAR.L.0\,exp\left(-\dfrac{PAR.L.E}{k_bT}\right) & T < P.BREAK \\ PAR.H.0\,exp\left(-\dfrac{PAR.H.E}{k_bT}\right) & T \geq P.BREAK \end{cases} \qquad \text{3-167}$$

B_i is determined as a function of temperature and time for lightly-doped substrates annealed at atmospheric pressure with no chlorine content in the ambient. The parameters in Equation 3-167 are specified for the appropriate oxidant species using the `OXIDE` statement. The pressure dependence and chlorine dependence are described in the following sections.

Pressure Dependence

The effects of pressure on the kinetics of the silicon oxidation process have been studied by Razouk *et al.* [39] for pyrogenic steam and Lie *et al.* [42] for dry oxygen. The parabolic rate varies with pressure because of its dependence on the oxidant equilibrium concentration in the oxide, C^*, which is directly proportional to the partial pressure of the oxidizing gas. The following relation is used to model this dependency.

$$B_P = P^{P.PDEP} \qquad \text{3-168}$$

Here, P is the partial pressure of the oxidizing gas in atmospheres and `P.PDEP` is specified on the `OXIDE` statement.

See Figure 3-13 for a plot of SiO_2 thickness as a function of time and pressure.

Chlorine Dependence

It has been observed that additions of chlorine during thermal oxidation also affect the parabolic rate constant. One possible explanation is that as chlorine enters the oxide film, it tends to cause the SiO_2 lattice to become strained, which increases the oxidant diffusivity [38].

Chlorine concentration dependence on the parabolic oxidation rate is modeled in a similar manner to that of the linear rate constant. Given an HCl percentage, a look-up table is used to determine an enhancement factor for the parabolic rate constant. Figure 3-14 shows the SiO_2 thickness dependency on HCl percentage.

3.3.6 Mixed Ambient Oxidation

In practice, an oxidizing ambient may be a gas mixture consisting of more than one oxidant and other impurities. The total oxidation rate will be the combined effect of all these species. To simulate oxidation under a multi-gas ambient, SSuprem4 simultaneously calculates the diffusion and oxidation of several ambient gases.

The capability is invoked by specifying the gas flow parameters: `F.O2`, `F.H2`, `F.H2O`, `F.N2`, and `F.HCL` on the `DIFFUSE` statement. From the gas flow, the partial pressure of each gas is calculated as:

$$P_j = P_{total}\frac{F_j}{\sum_i F_i} \qquad \text{3-169}$$

where P_j and F_j are partial pressure and gas flow rate for the j^{th} gas respectively, and P_{total} is the total pressure of the gas mixture (specified by the `PRESSURE` parameter on the `DIFFUSE` statement). If only one oxidant gas is specified in the gas flow (i.e., only O_2 or H_2O with other gases), oxidation is then modeled as previously described. Equation 3-169 determines the pressure of the oxidant gas. If both `F.H2` and `F.O2` are specified, the reaction of H_2 and O_2 to form H_2O is assumed to occur. The partial pressure of H_2O is then calculated before solving the oxidation equations.

For ambients containing more than one oxidant (e.g., O_2 and H_2O), the partial pressure of each oxidant is used to calculate C^* for each species. From C^*, k and D_{eff} for each oxidant species are calculated in a similar manner as that described in the pairs sections, respectively.

Equation 3-138 is solved for each oxidant to obtain each oxidant's concentration distribution in the growing SiO_2. The contributions of each oxidizing species to the Si/SiO_2 interface velocity is calculated with the following equation:

$$\boldsymbol{V_s} = \sum_i \frac{k_j C_{ij}\boldsymbol{n_{ij}}}{N_{1j}} \qquad \text{3-170}$$

where Equation 3-139 has been used and j corresponds to the j^{th} oxidant gas.

The flow equations are also calculated for a mixed ambient where both O_2 and H_2O exist and `COMPRESS` or `VISCOUS` has been specified on the `METHOD` statement. The stress dependence of D_{eff} and k is a function of the composition of dry or wet oxide which depends on oxidation history. Mixed ambient oxidation simulations take longer to solve than simple ambient equations.

3.3.7 Analytical Oxidation Model

You can use the analytical oxidation models to simulate a limited set of simple structures. Possible structures include a silicon substrate with an oxide layer deposited (or grown) on it. Since you can only specify the mask at the left part of a simulated structure, oxidation will only occur to the right of the mask edge. Analytical methods do not account for any real material layer located to the right of the specified mask edge. As the oxide layer thickens, the material is elevated, but growth rate and oxide shape are not affected. In all analytical models, the initial silicon surface must be planar.

The ERFG model simulates the bird’s beak oxide shape under nitride masks of different thicknesses [43]. The ERFG model consists of two models: ERF1 and ERF2. The ERF1 model describes the oxide growth under a thin nitride layer, where the stress from the nitride mask layer is negligible. ERF2 model describes the oxide growth when nitride layer thicknesses are large enough to cause stress in the oxide, which can result in the oxide layer being pinched. When ERFG is specified, either the ERF1 or ERF2 model will be automatically selected based on the structure under consideration. Both models are based on the error-function shape of the oxide/silicon and oxide/ambient or oxide/nitride interfaces:

$$Z = A\ erfc\ (By + C) + D \qquad 3\text{-}171$$

The *A*, *B*, *C*, and *D* parameters are complex functions of several geometric parameters:

- initial thickness of oxide l_{ox} and nitride l_n,
- current thickness E_{ox} of oxide given by the Deal-Grove Model (Equation 3-132),
- the length of lateral oxidation under the nitride layer L_{bb},
- and the lifting of the mask during oxidation H.

These functions are specified in the `OXIDE` statement. All defaults are taken from [43].

3.3.8 Recommendations for Successful Oxidation Simulations

Achieving successful oxidation simulations can be a frustrating task for a novice user of process simulation software. This section highlights some of the most common barriers encountered using process simulation to model oxidation steps, and describes how to overcome those barriers with the proper methods for simulating these oxidation steps.

One of the most common errors made in simulating oxidation steps is improperly gridding of the oxide structure. Improper gridding can result in jagged oxide shapes and errors in resolving impurity distributions. As the oxide layer is growing, grid points are added at predefined spacings. As silicon is being consumed, dopants are transported across the Si/SiO_2 interface. It is important to obtain a well gridded oxide to properly account for dopant redistribution during the oxidation step.

Growing Thin Oxides

A typical application where thin oxide growth is important is during a gate oxidation step of a MOSFET which has a highly-doped polysilicon gate. By default, SSuprem4 uses a grid spacing of 0.1 microns in the growing oxide layer. Thus, one grid layer will be added in the growing oxide every 0.1 microns (or 1000 angstroms). This grid spacing is appropriate for field oxidations, and hence the reason it is the default grid spacing in the growing oxide layer. Using the default grid spacing in the oxide for typical gate oxidations in today’s MOS technology results in no grid being added in the interior of the SiO_2 layer. With no grid present in the oxide to resolve the dopant diffusion in the oxide during subsequent processing, the polysilicon dopant can penetrate into the underlying silicon substrate. This simulation artifact can cause threshold voltages to be very different than expected.

To rectify this simulation artifact, you can control the number of grid layers added during the oxidation with the `GRID.OXIDE` and `GRIDINIT.OXIDE` parameters in the `METHOD` statement. You should place this statement before the gate oxidation diffusion step. Setting these parameters to a value, which results in three or four grid layers in the gate oxide (e.g., 15 angstroms for a 60 angstrom gate oxide thickness), can alleviate this problem. We suggest you to set back these parameters to the default values after the gate oxidation step.

Figure 3-17 shows a cross section of an NMOSFET with a highly-doped phosphorus polysilicon gate. The default grid spacing in the oxide is used in Figure 3-17(b), while the grid spacing is adjusted properly for Figure 3-17(a). By comparing these two figures, it is obvious that phosphorus has penetrated through the gate oxide for Figure 3-17(b) but does not penetrate through the gate oxide in Figure 3-17(a).

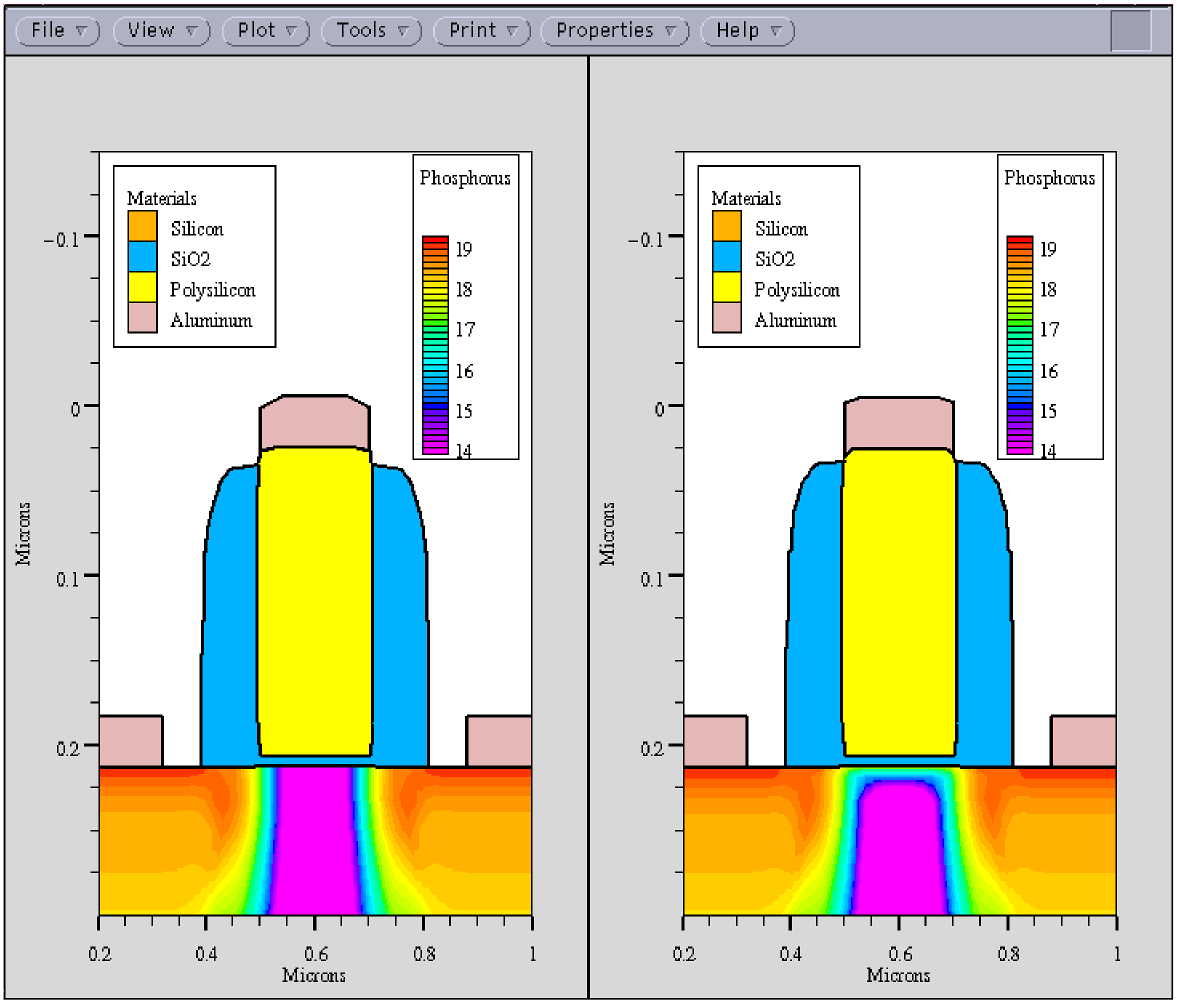


Figure 3-17 (a) MOSFET Structure with Proper Gridding in Gate Oxide, (b) MOSFET Structure with Default Grid Spacing in Gate Oxide

Implantation Through Thermally-Grown Oxides and Dopant Loss During Subsequent Annealing

Frequently, dopants are implanted through thermally grown oxide layers. It is important to have a proper grid spacing in the oxide through which the dopant is implanted for two reasons. First, this will aid in determining the proper dopant profile in the oxide layer and the underlying silicon. Secondly, proper gridding is required to resolved the dopant diffusion in the oxide during subsequent processing steps. During annealing, the dopant will diffuse in SiO_2 and silicon and eventually evaporate into the ambient at the gas/SiO_2 interface. If proper gridding is not supplied in the growing oxide layer, the amount of dopant evaporating can be underestimated, yielding a larger dose retained in the silicon substrate. The mechanism is similar to what was described in the earlier sections. There may not be any grid points in the interior of the growing SiO_2 layer. The problem is again remedied by specifying more grid layers to be added as the SiO_2 layer grows.

Figure 3-18 shows a comparison of the resulting arsenic profiles in silicon using the default grid spacing and a corrected grid spacing in the growing SiO_2 layer. For this experiment, a silicon dioxide layer was thermally grown. Arsenic was ion implanted through the SiO_2/Silicon structure. A subsequent annealing step followed which results in the profiles shown in Figure 3-18.

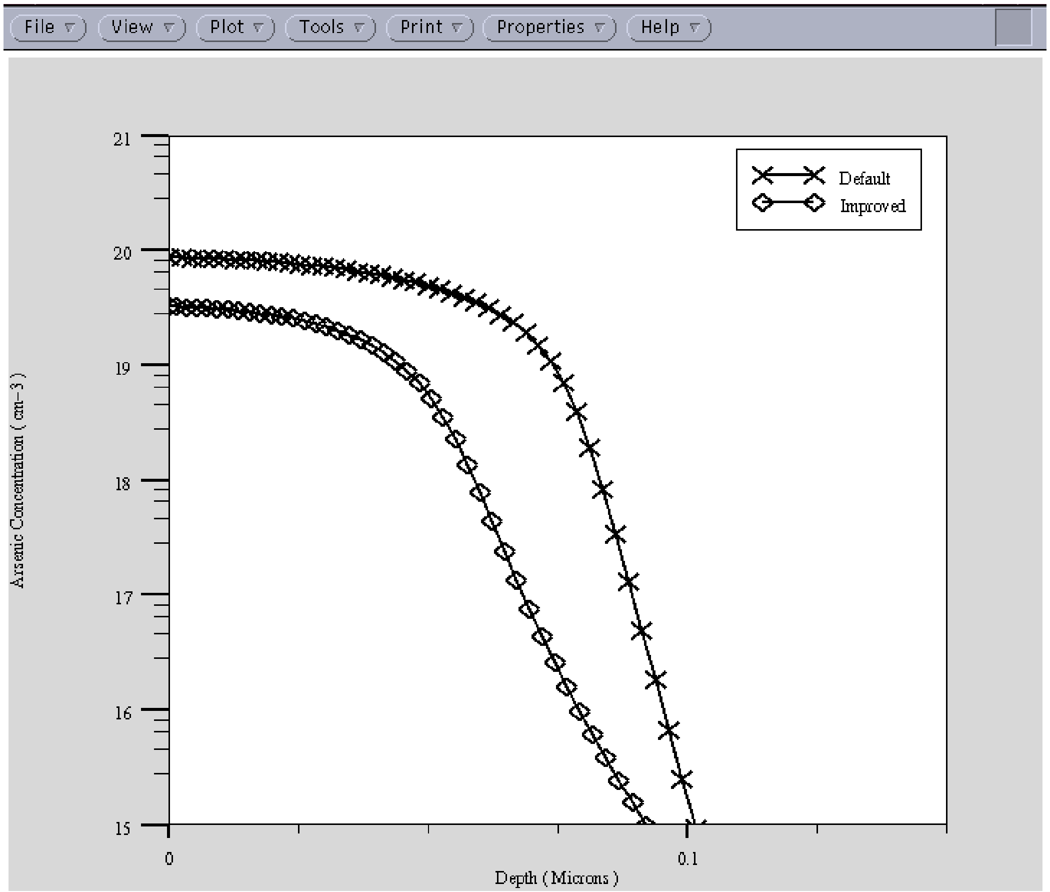


Figure 3-18 Comparison of Arsenic Profiles in Silicon with Default Grid Spacing and Improved Grid Spacing in the Growing SiO_2 layer.

Oxidation Enhanced Diffusion (OED) / Oxidation Retarded Diffusion (ORD)

During silicon thermal oxidation, some of the dopant in silicon gets incorporated into the growing SiO_2 layer and some remains in silicon where it diffuses. As oxidation proceeds, silicon lattice atoms become interstitial (interstitials are injected into silicon at the Si/SiO_2 interface) as oxygen molecules are incorporated into the lattice to form SiO_2. Due to the injection of interstitial defects during oxidation, you can enhance dopant diffusivities. To properly simulate this effect, you must include the creation and movement of point defects, vacancies and interstitials in the simulation. By specifying `TWO.DIM` in the `METHOD` statement before the oxidation step, non-equilibrium point defect concentrations (including injection and recombination at the Si/SiO_2 interface) are included in the simulation. For more information on point defect diffusion kinetics, see Section 3.1 “Diffusion Models”.

Note: Figure 3-19 compares the boron concentration profiles after an oxidation step when point defects remain at their equilibrium values (FERMI) and when point defects are allowed to obtain non-equilibrium values (TWO.DIM). It is evident from Figure 3-19(a) that boron diffusion is enhanced for the TWO.DIM case. The corresponding interstitial concentrations are shown in Figure 3-19(b). The interstitial concentration is above the equilibrium interstitial concentration for the TWO.DIM case (thus, allowing oxidation enhanced diffusion to be observed) but remains at equilibrium for the FERMI case.

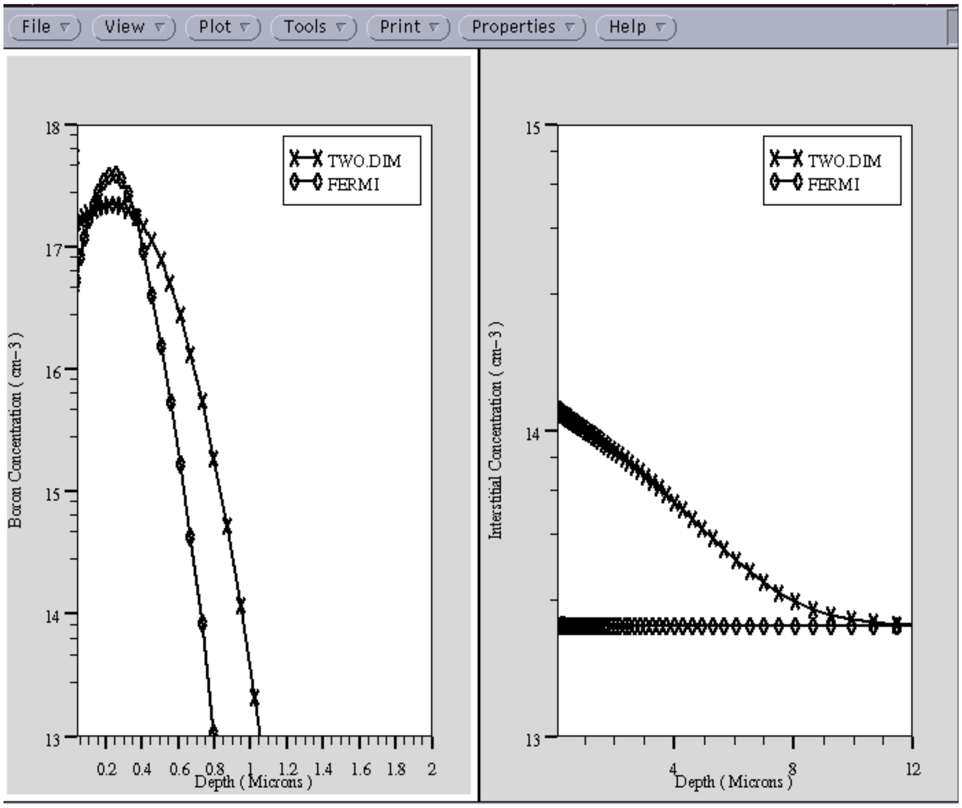


Figure 3-19 (a) Boron Concentration Versus Depth (b) Corresponding Interstitial Concentration Versus Depth

You can also have a diffusion retardation effect during thermal oxidation. For dopants diffusing primarily through a vacancy mechanism, you can reduce their diffusivities during oxidation because of the recombination of vacancies with injected interstitials at the SiO_2/ Silicon interface. Figure 3-20 shows an example of this phenomenon.

Figure 3-20(a) shows the resulting antimony concentration profiles after an oxidation step where the `FERMI` and `TWO.DIM` models were used. In contrast to boron (Figure 3-19(a)), the resulting antimony concentration profile is shallower for the `TWO.DIM` case when compared to the `FERMI` case.

Figure 3-20(b) shows the reduced vacancy concentration, displaying the results from the `TWO.DIM` and `FERMI` models.

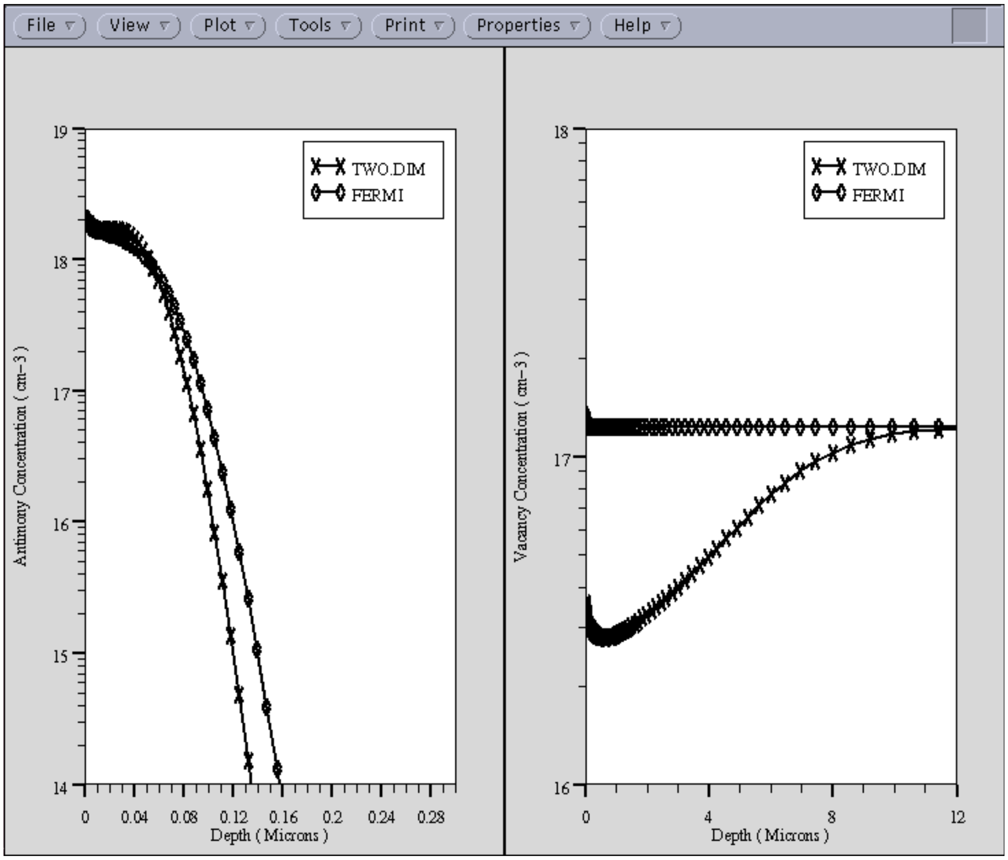


Figure 3-20 (a) Antimony Concentration Versus Depth (b) Corresponding Vacancy Concentration Versus Depth

3.4 Silicidation Model

Silicide modeling capability is implemented in SSuprem4. Silicides are formed when a metal reacts with silicon or polysilicon to create an intermediate phase. The conductivity of silicides is typically orders of magnitude greater than that of highly doped n$^+$ and p$^+$ regions. Modern CMOS technologies use silicides to reduce contact and interconnect resistances. Also, the use of SALICIDE technology (self-aligned silicides) is a practical way to reduce poly gate resistance and source and drain sheet resistance.

Silicidation process is invoked by depositing refractory metal layers on the exposed silicon/ poly surface and then specifying a thermal cycle in the `DIFFUSE` statement. There are four standard refractory metals in Athena: Titanium, Tungsten, Platinum, and Cobalt. Corresponding silicides are called TISIX, WSIX, PTSIX, and COSIX. User-defined metal and corresponding silicide can be also specified using parameters `MTTYPE` and `/MTTYPE` in the `SILICIDE` statement.

The modeling and understanding of silicide growth is nowhere near as developed as for oxidation. But, you can consider simulation of silicidation process similar to that of oxidation. It starts with insertion of a thin (0.002 microns) initial layer of silicide on the boundary between silicon (or polysilicon) and corresponding metal. During each time step, growth velocities are calculated for each point at both metal-silicide and silicon (or polysilicon)-silicide interfaces. The growth velocity at the i^{th} interface point is calculated as follows:

$$\frac{dx_i}{dt} = k_i C_i \frac{n_{in}}{N_{1i}} \qquad 3\text{-}172$$

where k_i is the interface reaction rate coefficient, N_{1i} is the number of silicon or metal molecules per unit silicide material, and C_i is the silicon or metal concentration. n_{in} is the interface normal vector which points towards the silicon-poly or metal side. Similarly to oxidation, this equation can be solved by applying an initial boundary condition $x_i = x_0$ at $t = 0$. The solution is

$$\frac{x_i^2}{B} + \frac{x_i}{B/A} = t + \tau \qquad 3\text{-}173$$

where parameters $B = (2D_iC_i)/N_{1i}$ and $B/A = (k_iC_i)/N_{1i}$ are equivalent to Deal-Grove coefficients of classical oxidation model. The silicide growth data indicates that for most silicides the rate limiting step is diffusion of silicon. This simplifies the Equation 3-174 to

$$x_i^2 = Bt \qquad 3\text{-}174$$

The silicide growth rates parameters are extracted from experimental data for $TiSi_2$ [44], [45] and CoSi2 [46]. For two other standard silicides PTSIX and WSIX as well as for user-defined Silicides the TiSi2 growth rates are used. The silicide growth rates can be modified by varying parameters `D.0` and `D.E` for silicon (or interstitial) diffusivity in silicide D_i, which are specified in the `INTERSTITIAL` statement. Defaults are `D.0=1.96` and `D.E=1.81` for all silicides.

Silicide formation usually leads to a large volume decrease. The ratio between consumed volumes of silicon and metal and resultant volume of silicide are specified by ALPHA parameters in the SILICIDE statement. The default values for the ALPHA parameter are taken from [47].

The 2D movement of growth interfaces and volume change cause the viscous flow of the silicide layer. This silicide flow is modeled analogously to the compress model of oxidation, where the equations solved are:

$$\mu \nabla^2 \mathbf{V} = \nabla P \qquad 3\text{-}175$$

$$\nabla \cdot \mathbf{V} = -\left(\frac{1-2v}{\mu}\right) P \qquad 3\text{-}176$$

$$\mu = \frac{E}{2(1+v)} \qquad 3\text{-}177$$

where:

- $\mathbf{V}$ is the velocity.
- P is the pressure.
- μ is the viscosity.
- v is Poisson’s ratio.
- E is Young’s modulus.

The parameters v and E are specified through elastic constants as described in Section 3.10 “Stress Models”.

3.5 Ion Implantation Models

Athena uses analytical and statistical techniques to model ion implantation. By default, the analytic models are used. Analytical models are based on the reconstruction of implant profiles from the calculated or measured distribution moments. The statistical technique uses the physically based Monte Carlo calculation of ion trajectories to calculate the final distribution of stopped particles.

3.5.1 Analytic Implant Models

Athena uses spatial moments to calculate ion implantation distributions. This calculation method is based on range concepts from “Range Concepts and Heavy Ion Ranges” [48] in which an ion-implantation profile is constructed from a previously prepared (calculated or measured) set of moments. A 2D-distribution could be essentially considered a convolution of a longitudinal (along the implant direction) 1D-distribution and a transverse (perpendicular to implant direction) 1D-distribution.

In the rest of this section, we will first describe three 1D implant models and the method used to calculate 1D profiles in multi-layered structures. Then, two models of transverse (lateral) distribution and a method of construction of 2D implant profiles will be outlined. Finally, three methods of implant parameter specification will be described.

Gaussian Implant Model

There are several ways to construct 1D profiles. The simplest way is using the Gaussian distribution, which is specified by the `GAUSS` parameter in the `IMPLANT` statement:

$$C(x) = \frac{\phi}{\sqrt{2\pi}\Delta R_{\rho}} \, exp\frac{-(x-R_p)^2}{2\Delta R_p^2} \tag{3-178}$$

where ϕ is the ion dose per square centimeter specified by the `DOSE` parameter. R_p is the projected range. ΔR_p is the projected range straggling or standard deviation.

Pearson Implant Model

Generally, the Gaussian distribution is inadequate because real profiles are asymmetrical in most cases. The simplest and most widely approved method for calculation of asymmetrical ion-implantation profiles is the Pearson distribution, particularly the Pearson IV function. Athena uses this function to obtain longitudinal implantation profiles.

The Pearson function refers to a family of distribution curves that result as a consequence of solving the following differential equation:

$$\frac{df(x)}{dx} = \frac{(x-a)f(x)}{b_0 + b_1 x + b_2 x^2} \tag{3-179}$$

in which *f(x)* is the frequency function. The constants *a*, b_0, b_1 and b_2 are related to the moments of *f(x)* by:

$$a = -\frac{\Delta R_p \gamma(\beta + 3)}{A} \tag{3-180}$$

$$b_0 = -\frac{\Delta R_p^2 (4\beta - 3\gamma^2)}{A} \tag{3-181}$$

$$b_1 = a \tag{3-182}$$

$$b_2 = -\frac{2\beta - \gamma^2 - 6}{A} \tag{3-183}$$

where $A = 10\beta - 12\gamma^2 - 18$, γ and β are the skewness and kurtosis respectively.

These Pearson distribution parameters are directly related to the four moments ($\mu_1, \mu_2, \mu_3, \mu_4$) of the distribution $f(x)$:

$$R_p = \mu_1 \qquad \Delta R_p = \sqrt{\mu_2} \qquad \gamma = \frac{\mu_3}{R_p^3} \qquad \beta = \frac{\mu_4}{R_p^4} \tag{3-184}$$

μ_i is given by:

$$\mu_1 = \int_{-\infty}^{\infty} x f(x) dx \tag{3-185}$$

$$\mu_i = \int_{-\infty}^{\infty} (x - R_p)^i f(x) dx \qquad i = 2, 3, 4 \tag{3-186}$$

The forms of the solution of the Pearson Differential Equation depend upon the nature of the roots in the equation $b_0+b_1x+b_2x^2=0$. There are various shapes of the Pearson curves. You can find the complete classification of various Pearson curves found in “Atomic and Ion Collision in Solids and at Surfaces” [49]. Obviously, only bell-shaped curves are applicable to ion implantation profiles. It is readily shown by Ashworth, Oven, and Mundin [50] that $f(x)$ has a maximum when $b_0+b_1x+b_2x^2 < 0$. You can reformulate this as the following relation between β and γ:

$$\beta > \frac{9\left\{(6\gamma^2 + 5) + [(9\gamma^6)/16 + 8\gamma^4 + 25(\gamma^2 + 1)]^{1/2}\right\}}{50 - \gamma^2} \tag{3-187}$$

with the additional constraint that $\gamma^2 < 50$.

Only Pearson type IV has a single maximum at $x = a + R_p$ and monotonic decay to zero on both sides of the distribution. Therefore, Pearson type IV is usually used for ion implantation profiles; it is the solution of Equation 3-178 when the following conditions are satisfied:

$$\beta = \frac{39\gamma^2 + 48 + 6(\gamma^2 + 4)^{3/2}}{32 - \gamma^2} \qquad and \qquad 0 < \gamma^2 < 32 \tag{3-188}$$

This gives the following formula for Pearson IV distribution:

$$f(x) = K[b_0 + b_1(x - R_p) + b_2(x - R_p)^2]^{\frac{1}{2b_2}} exp\left[-\frac{\frac{b_1}{b_2} + 2a}{\sqrt{4b_0b_2 - b_1^2}} atan\left(\frac{2b_2(x - R_p) + b_1}{\sqrt{4b_0b_2 - b_1^2}}\right)\right] \tag{3-189}$$

where K is defined by the constraint:

$$\int_{-\infty}^{\infty} f(x)dx = 1 \ . \tag{3-190}$$

In the narrow area of $\beta - \gamma^2$ plane where Pearson IV type criterion (Equation 3-188) is not satisfied while bell-shaped profile criterion (Equation 3-187) holds Athena, by default, uses other than type IV Pearson functions. These functions are bell-shaped but they are not specified over the whole $([-\infty, \infty])$ interval. Usually, this doesn't affect the quality of calculated profiles because the limits of these functions are situated far from their maximums. If you want to use only Pearson-IV distribution, set the `ANY.PEARSON` parameter to `FALSE`. In all cases when β and γ do not satisfy one of the mentioned criteria, Athena will automatically increase β up to the value that satisfies the criterion used. In the standard Pearson model, the longitudinal dopant concentration is proportional to the ion dose ϕ:

$$C(x) = \phi f(x) \tag{3-191}$$

This single Pearson approach (method) has been proved to give an adequate solution for many ion/substrate/energy/dose combinations. But, there are many cases when the channeling effects make the Single Pearson Method inadequate.

Dual Pearson Model

To extend applicability of the analytical approach toward profiles heavily affected by channeling, Al Tasch [51] suggests the dual (or Double) Pearson Method. With this method, the implant concentration is calculated as a linear combination of two Pearson functions:

$$C(x) = \phi_1 f_1(x) + \phi_2 f_2(x) \qquad \text{3-192}$$

where the dose is represented by each Pearson function $f_{1,2}(x)$. $f_1(x)$ and $f_2(x)$ are both normalized, each with its own set of moments. The first Pearson function represents the random scattering part (around the peak of the profile) and the second function represents the channeling tail region. Equation 3-191 can be restated as:

$$C(x) = \phi[\Re f_1(x) + (1-\Re) f_2(x)] \qquad \text{3-193}$$

where $\phi = \phi_1 + \phi_2$ is the total implantation dose and $\Re = \phi_1 / \phi$.

To use dual Pearson distribution, supply nine parameters— four moments for each Pearson function with the dose ratio $\Re$. The dual Pearson model will be used only when all nine parameters are present (see “Specification of Implant Parameters in the Moments Statement” on 208) and the `AMORPHOUS` parameter is not specified in the `IMPLANT` statement (the default is `CRYSTAL`). Otherwise, the single Pearson formula will be used.

SIMS-Verified Dual Pearson (SVDP) Model

By default, Athena uses SIMS-Verified Dual Pearson (SVDP) implant models. These are based on the tables from the University of Texas at Austin. These tables contain dual Pearson moments for **B**, **BF2**, **P**, and **As** extracted from high quality implantation experiments are also conducted by the University of Texas at Austin. Table 3-7 show these implantation tables contain dose, energy, tilt, rotation angle, and screen oxide thickness dependence.

Table 3-7 Range of Validity of the SVDP Model in Athena

Ions	**Energy** (keV)	**Dose** (cm^{-2})	**Tilt Angle**(°)	**Rotation Angle**(°)	**Screen Oxide** (Å)
B	1 — 100[a]	10^{13} — 8×10^{15}	0 — 10	0 — 360	native oxide — 500[b]
BF$_2$	1 — 80[c]	10^{13} — 8×10^{15}	0 — 10	0 — 360	native oxide
P	12 — 200[d]	10^{13} — 8×10^{15}	0 — 10	0 — 360	native oxide
As	1 — 200[e]	10^{13} — 8×10^{15}	0 — 10	0 — 360	native oxide

[a] Experimentally verified for *5-80keV*. For energy range, *1-5keV*, an interpolation between *5keV* and *0.5keV* calculated with `UT-MARLOWE`, is used; an extrapolation is used for energy range *80± 100keV*.

[b] Only for *15-80keV*.

[c] Experimentally verified for *5-65keV*. For energy ranges, *1-5keV* and *65-80keV*, the same procedures is used for boron.

[d] Experimentally verified for *15-80keV*. Numerical extrapolation is outside this energy range.

e Experimentally verified for *5-180keV.* Interpolation between *5keV* and UT-MARLOWE calculated profile at *0.5keV.*

If you choose a simulation outside the parameter ranges, described in Table 3-7, Athena will not use the Dual Pearson Implant SVDP Models but will use the standard tables instead. When using the Dual Pearson model, remember the following:

- For implant energies below *15keV,* for boron, BF_2 and arsenic, the simulation predicts the dopant profiles for implants into a bare silicon surface (i.e., silicon wafer subjected to an HF etch less than two hours before implantation). Low energy implant profiles at such low implant energies are found to be extremely sensitive to the presence of a thin (0.5-1.5*nm*) native oxide layer or disordered silicon layer on the wafer surface [52]. Remember this fact when using the model for the simulation of low energy ion implantation and when performing implantations.
- For implant energies between 10*keV* and 15*keV,* the simulations are performed for boron, BF_2, and arsenic by using an interpolation between the Dual Pearson Model parameters at 15*keV,* and the Dual Pearson Model parameters at *10keV.* The parameters at 15*keV* correspond to implantation through a native oxide layer (0.5-1.5*nm*), while the parameters at 10*keV* correspond to implantation into a bare silicon surface (i.e., silicon wafer subjected to an HF etch less than two hours before implantation).
- For implant energies below 5*keV,* the models for boron, BF_2, and arsenic have not been verified experimentally. The simulations in this range of implant energy are performed using an interpolation between experimentally verified Dual Pearson parameters at 5*keV* and parameters based on UT-MARLOWE estimates at 0.5*keV.*
- The SIMS measurements upon which these profiles are based have a concentration sensitivity limit in the order of 5×10^{15} to $2 \times 10^{16} cm^{-2}$, increasing with dose from the implant. The profiles have been extended below these limits, following the trends that occur within the sensitivity limits of the SIMS.
- The screen oxide thickness range has been verified from 1.5 to 40*nm* (only for boron and *15-80keV* energy range). But the oxide range has been extended to 50*nm*.

Screen Oxide Thickness Parameter (S.OXIDE)

To specify screen oxide, use the S.OXIDE (*µm*) parameterin the IMPLANT statement. This thickness is specified independently of any actual surface oxide in the structure. It is, however, possible to automate the extraction of the surface oxide thickness for use with the IMPLANT statement. An example is supplied demonstrating this.

S.OXIDE is another parameter for dual Pearson moments selection from the implant tables. It's up to you to select its value accordingly. The effect of this parameter is that it represents ion implantation through a thin (0÷50*nm*) surface oxide layer. The present algorithm in Athena when encountering a multi-layered structure (see Section 3.5.2 "Multi-Layer Implants"). For example, oxide/silicon switches to multi-material scaling technique for evaluating the depth profile. This technique will combine two profiles — single Pearson for the oxide and dual Pearson for silicon with S.OXIDE preferably set to the thickness of the oxide.

There are two reasons why this separation between the surface oxide is present in the structure before the IMPLANT statement and the S.OXIDE parameter. The first reason is because the flexibility of using this parameter for different thin surface layers other than oxide (with appropriate scaling of their thickness for stopping). The second reason the currently restricted availability of moments with screen oxide in the tables (0÷50*nm*, 15÷80keV and for boron only). If you need a more precise dependence of the implantation profiles on the surface screen oxide, use a single layer of silicon with S.OXIDE set to an appropriate value.

3.5.2 Multi-Layer Implants

To apply any of the described analytical distribution functions for structures that are comprised from several different material layers, use a scaling method that's mentioned in this section. This is because stopping powers and range parameters are different in different materials. This section will describe the implant scaling methods available in Athena.

DOSE.MATCH

The Dose-Matching Method was historically the first and is the most widely used [53] method. The Dose-Matching Method is selected by the DOSE.MATCH parameter (default) in the IMPLANT statement. With this method, the segment of the profile within *i*th layer is calculated by:

$$C_i(x) = \phi f(x - x_t + x_{eff}) \qquad 3\text{-}194$$

where *f(x)* is the distribution function specified for this implant (Gaussian, Pearson, or Dual Pearson) with moments corresponding to the *i*th layer; x_t is the distance from the surface to the top of the *i*th layer:

$$x_t = \sum_{k=1}^{k \le i-1} t_k \qquad 3\text{-}195$$

x_t is the thickness of the *k*th layer; x_{eff} is the effective thickness evaluated from:

$$\int_0^{x_{eff}} C_i(x)dx = \sum_{k=1}^{k \le i-1} \Phi_k \qquad 3\text{-}196$$

where ϕ_k is the portion of the total implant dose, which is consumed in the *k*th layer. Obviously, for the first layer $x_{eff} = 0$ and $x_t = 0$.

RP.SCALE and MAX.SCALE

The other two methods for analytical calculation of implantation profiles in the layered structures are projected range depth scaling (set by RP.EFF or RP.SCAL in the IMPLANT statement) and maximal depth scaling (set by the MAX.SCALE parameter). These two methods differ from the dose-matching method in the way the effective depth x_{eff} is calculated and in the normalization of the partial profiles in the layers. Like in the dose-matching method, the distribution in the first layer is calculated directly from the moments corresponding to the first layer without any corrections. For subsequent layers, the implant distribution is calculated by the formulae:

$$C_i(x) = Nf(x - x_t + x_{eff}) \qquad 3\text{-}197$$

and

$$N = \left(\phi - \sum_{k = 1}^{i-1} \Phi_k \right) / \phi_i \qquad \text{3-198}$$

where N is the normalization factor, ϕ is the total implantation dose, and x_{eff} is the effective depth calculated as follows. In the case of projected range scaling, x_{eff} for the i^{th} layer is:

$$x_{eff} = \sum_{k = 1}^{i-1} \frac{t_k}{R_p^k} R_p^{k+1} \qquad \text{3-199}$$

where R_p^k is the projected range of the specified ion in the material of the k^{th} layer. For the case of the maximal range scaling, x_{eff} is calculated as:

$$x_{eff} = \sum_{k = 1}^{i-1} \frac{t_k}{R_p^k + 3\Delta R_p^k} \left(R_p^{k+1} + 3\Delta R_p^{k=1} \right) \qquad \text{3-200}$$

where ΔR_p^k is the projected range straggling in the k^{th} layer. In this approximation, the estimated maximum ion range $R_p + 3\Delta R_p$ is taken as the measure of the ion penetration into the corresponding material.

MOM.SCALE

In all three models described above, the range parameters in each layer are considered independent of the presence of other layers. But obviously, the distribution of ions stopped in the deeper layers may depend on the thickness and stopping characteristics of the upper layers because each ion trajectory passes through these upper layers. The Moment Correction Method set by the `MOM.SCALE` parameter of the `IMPLANT` statement partially accounts for this effect. In the `SCALE.MOM`, method the projected range and range straggling in the layer are normalized according to the probability for the ion to penetrating into the layer. The only available measure of the probability is the portion of the dose accumulated in the specific layer. Therefore, the corrected projected range R_{pc} and range straggling ΔR_{pc} in the i^{th} layer are calculated as follows:

$$R_{pc} = \phi_i R_p^i + \sum_{k = 1}^{i-1} \phi_k R_p^k \qquad \text{3-201}$$

$$R_{pc} = \phi_i R_p^i + \sum_{k = 1}^{i-1} \phi_k R_p^k \qquad \text{3-202}$$

where

$$\phi_i = \phi - \sum_{k = 1}^{i - 1} \phi_k \qquad \text{3-203}$$

You can use the `SCALE.MOM` parameter together with any of three depth matching methods.

3.5.3 Creating Two-Dimensional Implant Profiles

Convolution Method

Athena calculates 2D implant profiles using a convolution method described as follows. First, it calculates the implantation direction within the simulation plane using the `TILT` θ and `ROTATION` φ angle parameters specified in the `IMPLANT` statement. θ is the angle between the ion beam direction and *y*-axis, φ is the angle between ion beam direction and the simulation plane. For example, φ = 0° and θ > 0° correspond to an ion beam parallel to the simulation plane and directed toward the lower-right corner of the simulation area. The case of φ = 90° and θ > 0° correspond to an ion beam in the plane perpendicular to the simulation plane and directed from behind the simulation plane. The effective implantation angle in the simulation plane could be found from

$$tan(\alpha) = tan(\theta)cos(\varphi) \qquad \text{3-204}$$

When the `FULLROTAT` parameter is specified in the `IMPLANT` statement Athena calculates superposition of 24 implants with rotation angles equal to (15*n)*° and doses equal to φ /24.

The implantation front (perpendicular to the α direction) is divided into a number of slices N_s (usually > 100) of width *a*. The implant concentration in each grid point, *i*, with coordinates (x_i, y_i) is calculated by the summation of contributions from each slice, *k*:

$$C(x_i, y_i) = \sum_{1 \le k \le N_s} C_k(x_i, y_i) \qquad \text{3-205}$$

The contribution from each slice C_k is calculated by integration of the point source 2D frequency function $F_{2D}(x,y)$ (with the starting point at the intersection of the normal ***n*** to the central of the slice with the structure surface) over slice width:

$$C_k(x_i, y_i) = \phi \int_{-a/2}^{a/2} f_{2D}(d_i, t_i)dt \qquad \text{3-206}$$

where d_i is the depth along implant direction (i.e., distance between the starting point and the projection of the point *i* on the vector ***n***) and t_i is the transversal distance (i.e., distance between the point i and the vector ***n***). See Figure 3-21.

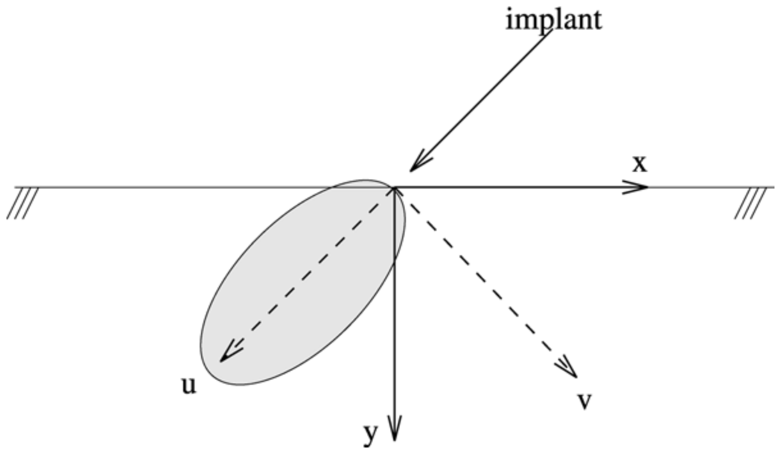


Figure 3-21 Integration Geometry for the Convolution Method

Depth-Independent Lateral Distribution

The simplest type of the 2D frequency function is a product of *longitudinal* function $f_l(x)$, which can be a Gaussian (Equation 3-178), Pearson (Equation 3-191), Dual-Pearson (Equation 3-193), and depth-independent *transversal* function $f_t(y)$. See Equation 3-207.

$$f_{2D}(x, y) = f_l(x)f_t(y) \tag{3-207}$$

This approximation is used in Athena by default. Obviously, the function $f_t(y)$ must be symmetrical and have a bell shape.

Gaussian Lateral Distribution Function

The traditional selection for this function is a Gaussian. Athena uses the Gaussian approximation unless the transversal kurtosis β_y (KURTT in the MOMENTS statement) is specified to be different from its default value of 3.0. In this case, Equation 3-206 can be easily integrated into the following equation:

$$C(x_i, y_i) = \frac{1}{2}\phi f_l(d_i)\left[erfc\frac{t_i - a/2}{\sqrt{2}\Delta Y} - erfc\frac{t_i + a/2}{\sqrt{2}\Delta Y}\right] \tag{3-208}$$

where ΔY is the transversal (lateral) standard deviation defined from:

$$\Delta Y = \iint_{-\infty}^{\infty} f_{2D}(x, y)y^2 dxdy \tag{3-209}$$

Specification of Lateral Standard Deviation

You can specify Lateral Standard Deviation (LSTD.DEV or LDRP) together with other moments in the MOMENTS statement (see “Specification of Implant Parameters in the Moments Statement” on 208). You can also control it with the LAT.RATIO1 parameter in the IMPLANT statement. LAT.RATIO1 is the ratio between ΔY and ΔR_p which is equal to 1.0 by default. This means that if the lateral standard deviation and LAT.RATIO1 are not specified it will be equal to projected range straggling ΔR_p. In the case of dual Pearson model for longitudinal profile, corresponding parameters, SLSTD.DEV or SLDRP and LAT.RATIO2 are used. The only difference is that the default for LAT.RATIO2 is 0.2. This is because the channelled portion of a 2D profile is obviously very narrow.

Parabolic Approximation of Depth-Dependent Lateral Distribution

It has been shown [54], [55], [56], and [57] that in general, the transversal function $f_t(y)$ is not independent of depth because there is considerable correlation between transversal and longitudinal motion of the implanted ions. This correlation could be taken into account by using a transversal function with the depth-dependent lateral standard deviation $\sigma_y(x)$. As it was shown in [55] and [57], if the spatial moments up to fourth order are used, the best approximation for $\sigma_y(x)$ is the parabolic function:

$$\sigma_y^2(x) = c_0 + c_1(x - R_p) + c_2(x - R_p)^2 \tag{3-210}$$

In order to find the coefficients of the function two additional spatial moments should be used. These are mixed skewness:

$$\gamma_{xy} = \iint\limits_{-\infty}^{\infty} f_{2D}(x, y)(x - R_p)y^2 dxdy \tag{3-211}$$

and mixed kurtosis:

$$\beta_{xy} = \iint\limits_{-\infty}^{\infty} f_{2D}(x, y)(x - R_p)^2 y^2 dxdy \tag{3-212}$$

The c_0, c_1, and c_2 parameters can be found by substituting Equation 3-207 into Equations 3-209, 3-211, and 3-212 and taking into account in Equations 3-180—3-186, while integrating over x. This results in the system of equations where you can find the following relations.

$$c_0 = \Delta Y^2(1 - B) \tag{3-213}$$

$$c_1 = \frac{\Delta Y^2}{\Delta R_p}(\gamma_{xy} - \gamma B) \tag{3-214}$$

$$c_2 = \frac{\Delta Y^2}{\Delta R_p^2}B \tag{3-215}$$

where:

$$B = \frac{\beta_{xy} - 1 - \gamma\gamma_{xy}}{\beta - 1 - \gamma^2} \qquad 3\text{-}216$$

This parabolic approximation for depth-dependent f_t will be used if the FULL.LAT parameter is used in the IMPLANT statement and when mixed spatial moments: γ_{xy} (SKEWXY parameter) and β_{xy} (KURTXY parameter) are non-zeros. In the case of the Dual Pearson longitudinal function, the mixed spatial moments for the second Pearson, SSKEWXY and SKURTXY, can be also specified. The values of spatial moments are not yet included in the default moments tables and should be specified in the MOMENTS statement (see “Specification of Implant Parameters in the Moments Statement” on 208).

Non-Gaussian Lateral Distribution Functions

Detailed Monte Carlo simulations [58] and [59] also show that in most cases, transversal distribution function, $f_{t,}$ is not Gaussian. In other words, the transversal kurtosis β_y is calculated as

$$\beta_y = \iint_{-\infty}^{\infty} f_{2D}(x, y)y^4 dxdy \qquad 3\text{-}217$$

and is not always equal to 3.0 and also depends on depth. Several non-Gaussian transversal distribution functions were examined in [49]. Their conclusions were as follows. The symmetrical Pearson functions (type II for $\beta_y \le 3$ and type VII when β_y > 3) are acceptable, providing an agreement with amorphous Monte Carlo simulations and have computational advantage because they can be integrated over x in a closed form through incomplete beta functions [60].

Another good alternative for transversal distribution function is the Modified Gaussian Function (MGF) suggested in [58]. It is shown in [60] that it also can be integrated in the close form through the incomplete gamma function. Selection of transversal distribution function is subjective because it is based on comparison with the lateral cross-section of the 2D Monte Carlo distributions, which cause accuracy to diminish further away from its maximum. The analysis of [59] based on the BCA simulation (see Section 3.5.4 “Monte Carlo Implants”) showed that when $\beta_y \le 2.5 \div 2.8$, which usually happens for random part of the 2D distribution or for amorphous implants, the Pearson type II function slightly underestimates concentrations obtained in the BCA calculations while the MGF slightly overestimates these concentrations. Therefore, it was decided to use in Athena an average between the Pearson type II and the MGF for all $\beta_y < 3$. When $\beta_y = 3$ both functions reduce to standard Gaussian. Finally, in the case of higher values of lateral kurtosis it was found [59] that the MGF appears to be a better approximation, so it is used in Athena.

It is very difficult to find $\beta_y(x)$ as was done for $\sigma_y(x)$ already mentioned, because the spatial moments of fifth and sixth order would be needed to build analytical functions for $\beta_y(x)$. Therefore, Athena uses constant β_y (the KURTT and SKURTT parameters for the first and second Pearson functions, correspondingly) when you specify the FULL.LAT model in the

IMPLANT statement. The generic approximations [59] for $\sigma_y(x)$ (instead of Equation 3-210) and for $\beta_y(x)$ will be implemented in future when more complete tables of lateral parameters will be generated using Monte Carlo simulations.

Specification of Implant Parameters in the Moments Statement

As mentioned previously, the analytical ion implantation simulations strongly depend on the input parameters (moments). Athena provides several ways of implant parameter specification. They are: look-up tables, user-defined look-up tables, and the MOMENTS statement.

Two types of look-up tables are currently provided with Athena. The files containing the tables are in ASCII format and can be found in the <install>/lib/athena/<>version>/common/implant-tables directory. The first type are standard tables (std-tables file) containing parameters for most ion-material combinations used in Athena. These are longitudinal parameters for the single Pearson distribution in the energy interval 10keV to 1MeV. The energy interval is extended to 1keV÷8MeV for B, P, and As in silicon, silicon oxide, polysilicon, and silicon nitride. These tables also include a limited set of parameters for the dual-Pearson function (only for B and BF_2 in the energy interval 10÷100*keV*, tilt angle 7°, rotation angle 30°, and native oxide as well as with simple interpolation of the dose ratio parameter $\Re$ between different doses). Parameters for the FULL.LAT model are provided only for B implants in silicon. They are based on the spatial moment calculations in amorphous silicon as in [61], [62]. The auxiliary file *userimp* in the *<install>/lib/athena/<version>/common* directory provides a template for specifying implant parameters in the format of standard tables. The second type of look-up tables are SVDP tables described in the “Dual Pearson Model” on 200. The format of these tables is much more flexible than the format of the standard tables. It also allows parameters for lateral distribution to be added easily.

The SVDP tables are used by default. If no moments are found, Athena will search through standard tables. If it cannot find parameters for a specified energy/ion/material combination, a warning message is issued, which will tell you a very small projected range and straggling will be used in simulation for this combination. The message will also suggest that you use the Monte Carlo method in order to find the right moments. This is the sequence of Athena actions in the case when no MOMENTS statement precede the current IMPLANT statement. The MOMENTS statement serves to control the moment parameters/tables to be used in subsequent IMPLANT statements. If you specify the STD_TABLES parameter, Athena will skip searching through SVDP tables and proceed directly to the standard tables. If you specify the USER_STDT or USER_SVDPT parameter, then the user-defined file specified with the USER.TABLE=<filename> parameter will be used as the first choice. Of course, if the moments are not found in the specified file, Athena will proceed to the standard tables. Finally, the set of MOMENTS statement can be used to specify all spatial moments for any ion/material/energy/dose combination. Athena will use parameters from this set before proceeding to a standard search sequence. If the moments for certain implant conditions are unavailable, then you can use the Monte Carlo simulation for these conditions.

Using PRINT.MOM for Extraction of Spatial Moments

The PRINT.MOM parameter in the IMPLANT statement prints the calculated (or extracted from the tables) moments into output and also saves the moments in the standard structure file. The last capability allows you to use the extract (EXTRACT statement) and substitution functions of Deckbuild for automatic generation of the MOMENTS statement. If spatial (lateral and mixed) moments need to be found from Monte Carlo calculation, use the IMPCT.POINT parameter because it forces all trajectories to be started in one point. This not only allows the spatial moments to be found, but also the building of a Monte Carlo calculated source-point 2D distribution function, which can be useful for comparison purposes.

3.5.4 Monte Carlo Implants

The analytical models described in the previous section give very good results when applied to ion implantation into simple planar structures (bare silicon or silicon covered with thin layer of other material). But for structures containing many non-planar layers (material regions) and for the cases, which have not been studied yet experimentally requires more sophisticated simulation models. The most flexible and universal approach to simulate ion implantation in non-standard conditions is the Monte Carlo Technique. This approach allows calculation of implantation profiles in an arbitrary structure with accuracy comparable to the accuracy of analytical models for a single layer structure.

Athena contains two models for Monte-Carlo simulation of ion implantation: Amorphous Material Model and Crystalline Material Model. Both of them are based on the Binary Collision Approximation (BCA) and apply different approximations to the material structure and ion propagation through it.

Nature of the Physical problem

A beam of fast ions (energy range, approximately 50 eV/amu to 100 keV/amu) entering crystalline or amorphous solid is slowed down and scattered due to nuclear collisions and electronic interaction. Along its path, an individual projectile may create fast recoil atoms that can initiate collision cascades of moving target atoms. These can either leave the surface (be sputtered) or deposited on a site different from their original one. Together with the projectiles being deposited in the substrate, this results in local compositional changes, damage creation and finally amorphization of the target. Depending on the crystal orientation or the direction of the beam or both, the implanted projectiles and the damage created by them has different spatial distribution. With even more higher fluency, these phenomena will cause collisional mixing in a layered substances, changes of the surface composition due to preferential sputtering, and the establishment of a stationary range profile of the implanted ions.

Method of Solution

The paths of the individual moving particles and their collisions are modeled by means of the binary collision approximation for a crystalline, polycrystalline and amorphous substance, using a screened Coulomb potential for the nuclear collisions and a combination of local and non-local free-electron-gas approximation for the electronic energy loss. For each nuclear collision, the impact parameter and the Azimuthal Deflection Angle are determined according to the crystal structure using its translational symmetry. For amorphous materials, the impact parameter and the azimuthal deflection angle are determined from random numbers. A proper scaling is chosen so that each incident projectile (pseudo-projectile) represents an interval of implantation dose. Subsequent to the termination of each pseudo-projectile and its associated collision cascades, the local concentrations of the implanted species, created vacancies and interstitials are calculated according to the density of the matrix.

Nuclear Stopping

As mentioned before, during their passage through matter ions interact not only with the atoms from the lattice but also with the electrons. Figure 3-22 shows the scattering geometry of two particles in the Laboratory Co-ordinate System. In the computational model, it is assumed that ions from one deflection point to the next move along straight-line segments, these being the asymptotes of their paths. At each collision, ion loses energy through quasielastic scattering by a lattice atom and by an essentially separate electron energy loss part.

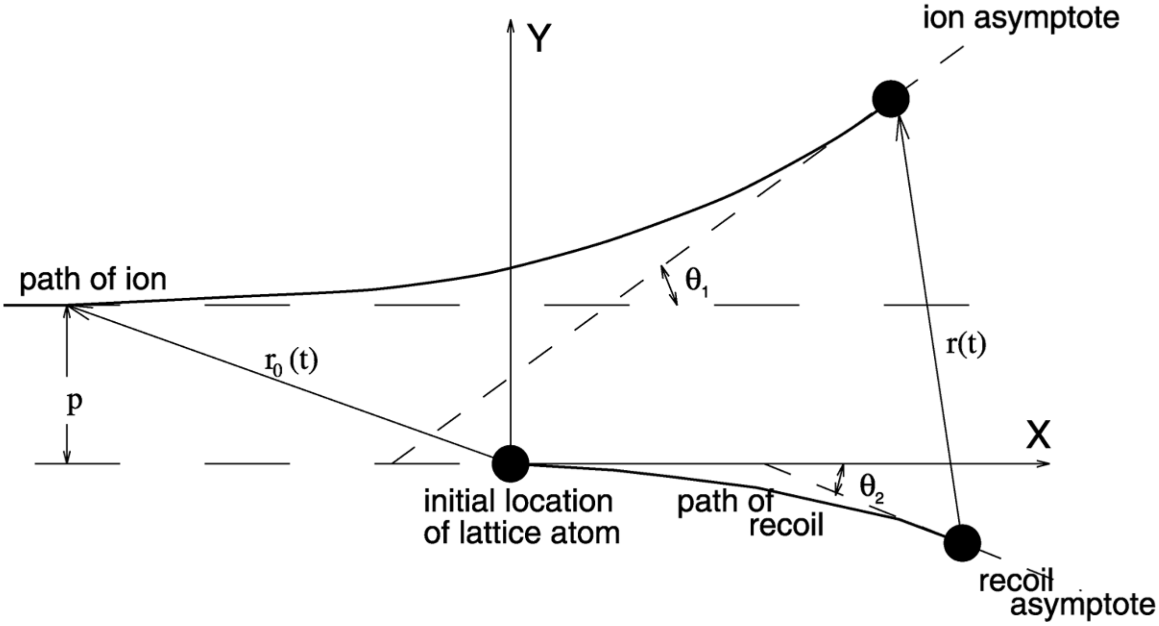


Figure 3-22 The trajectories of the ion (projectile) and the lattice atom (recoil).

The scattering angles of the projectile and the recoil are as follows:

$$\tan \vartheta_1 = Af \sin\theta / (1 + Af \cos\theta) \qquad 3\text{-}218$$

$$\tan \vartheta_2 = f \sin\theta/(1 + f \cos\theta) \quad \text{3-219}$$

where:

$$f = \sqrt{1 - Q/E_r} \quad \text{3-220}$$

Q is the energy lost by electron excitation.

$A = M_2/M_1$ is the ratio of the mass of the target (scattering) atom to that of the projectile (implanted ion).

θ is the barycentric scattering angle calculated as follows:

$$\theta = \pi - 2p\int_R^{\infty} \frac{1}{r^2 g(r)} dr \quad \text{3-221}$$

where:

$$g(r) = \sqrt{1 - \frac{p^2}{r^2} - \frac{V(r)}{E_r}}$$

where:

- p is the impact parameter,
- $E_r = AE_0/(1 + A)$ is the relative kinetic energy,
- E_0 is the incident energy of the projectile,
- r is interatomic separation,
- $V(r)$ is the interatomic potential,
- R is defined from equation $g(R) = 0$.

In Athena, the intersections of the incoming and outgoing asymptotes are evaluated with the hard core approximation of the time integral:

$$x_1 = p\tan(\theta/2) \quad \text{3-222}$$

$$x_2 = 0 \quad \text{3-223}$$

Interatomic Potential

Athena uses two-body screened Coulomb potentials with a screening function, which is a numerical fit to the solution given by Firsov [63]. It also preserves the same analytic form as for the isolated atom:

$$V(r) = \frac{Z_1 Z_2 e^2}{r}\chi\left(\frac{r}{a_0}\right) \quad \text{3-224}$$

where Z_1 and Z_2 are the atomic numbers of the two atoms and a_0 is the screening length defined by

$$a_0 = 0.8853 a_B \bar{Z}^{-1/3} \tag{3-225}$$

where $\bar{Z}$ is an ‘average’ atomic number of the two atoms calculated as

$$\bar{Z}^{-1/3} = \left(Z_1^{0.23} + Z_2^{0.23}\right)^{-1} M \quad . \tag{3-226}$$

The main drawback of these two-body potentials is their relatively slow decay as $r \rightarrow \infty$. The screening parameter, a_0, is often regarded as an adjustable parameter for each two-body combination, which can be matched either to self-consistent field calculations or to experimental data. Athena uses the screening function in the form

$$\chi = \sum_{i=1}^{4} a_i exp(-b_i x) \tag{3-227}$$

where a_i and b_i are taken from [64].

Electronic Stopping

Electronic stopping used in the simulation consists of two essentially separate mechanisms for inelastic energy losses: local and non-local. These two types of electronic stopping are quite different in nature and behavior -- they have different energy and spatial dependencies [65]. The local inelastic energy losses are based on the model proposed by Firsov [66]. In this model, the estimation of the electronic energy loss per collision is based on an assumption of a quasi-classical picture of the electrons (i.e., the average energy of excitation of electron shells, and electron distribution and motion according to the Thomas-Fermi model of the atom).

In this quasi-classical picture, the transfer of energy, ΔE, from the ion, to the atom, is due to the passage of electrons from one particle to the other. Thus, resulting in a change of the momentum of the ion (proportional to its velocity ,υ, and a rising of a retarding force acting on the ion). When ions move away from the atom (despite being trapped by ions) electrons will return to the atom. There is no transfer of momentum calculated back, because the electrons fail in higher energy levels. The energy loss in the Firsov's Model is calculated as follows:

$$-\Delta E = \frac{0.05973 \times (Z_1 + Z_2)^{5/3} \sqrt{E/M_1}}{\left(1 + 0.31(Z_1 + Z_2)^{1/3} R_0\right)^5} eV \tag{3-228}$$

where:

- R_0 is their distance of closest approach in $\overset{o}{A}$, which is approximately equal to the impact parameter in case of small-angle collisions.
- E is the energy of the moving atom (the ion) in eV.

- M_1 is its mass in a.m.u.

In a binary collision, the scattering angles are affected by the inelastic energy loss ΔE (see Equation 3-228) through the parameter f.

The non-local electronic energy losses are based on the model proposed by Brandt and Kitagawa [67]. Their stopping power, $S = -\frac{dE}{dx}$, of the medium for an ion is in the first approximation proportional to a mean-square effective ion charge. They derive the effective stopping power charge of a projectile, Z_1^* from a given ionization state, q. If a fractional effective charge of an ion with the given ionization state, q is defined as

$$\zeta \text{½} \frac{Z_1^*}{Z_1} = \left[\frac{S_q}{S_{q=1}}\right]^{1/2} \qquad \text{3-229}$$

where $S_{q=1}$ is the stopping power for bare nucleus. Brandt and Kitagawa theories produces the following simple expression for the fractional effective charge of an ion:

$$\zeta \approx q + C(k_F)(1-q)\ln(1 + (2\Lambda v_F a_0 v_0)^2) \qquad \text{3-230}$$

where:

- $q = (Z_1 - N)/Z_1$ is the fractional ionization,
- N is the number of electrons still bond to the projectile nucleus,
- a_0 and r_0 are Bohr's radius and velocity,
- k_F and v_F are Fermi wave vector and velocity.

For the screening radius Λ, Brandt and Kitagawa assume exponential electron distribution, which becomes:

$$\Lambda = \frac{0.48N^{2/3}}{Z_1(1 - N/(7N_1))} \qquad \text{3-231}$$

The only undefined quantity, C, is of about 0.5 and somewhat depends on the target. The degree of ionization, q, can be expressed as

$$q = 1 - exp\left(\frac{-0.92v_r}{v_0 Z_1^{2/3}}\right) \qquad \text{3-232}$$

where $v_r \equiv \langle|\mathbf{v}_1 - \mathbf{v}_e|\rangle$ is the relative velocity between the projectile and the target electrons, which are calculated as follows:

$$v_r = \frac{3v_F}{4}\left[1 + \left(\frac{2v_1^2}{3v_F^2}\right) - \frac{1}{15}\left(\frac{v_1}{v_F}\right)\right] \qquad for \qquad v_1 < v_F \qquad \text{3-233}$$

$$v_r = v_1\left(1 + \frac{v_F^2}{5v_1^2}\right) \qquad for \qquad v_1 \geq v_F \tag{3-234}$$

Damage Accumulation Model

The present model includes dynamic processes of the transformation from crystalline to amorphous state as ion implantation proceeds. Each pseudo-projectile in the simulation represents a portion of the real dose Φ, where N is the number of projectiles.

$$\Delta\Phi = \frac{\Phi}{N} \tag{3-235}$$

The deposited energy is accounted for each grid point of the target and accumulated with the number of projectiles. As the implantation proceeds, deposited energy increases and the crystalline structure gradually turns into an amorphous structure. This is quantified by the Amorphization Probability Function as follows:

$$f(\mathbf{r}) = 1 - exp\left[\frac{\Delta E(\mathbf{r})}{E_c}\right] \tag{3-236}$$

Here, $\Delta E(\mathbf{r})$ is the energy deposited per unit volume at the grid point **r**, and E_c is the critical energy density, which represents the deposition energy per unit volume needed to amorphize the structure in the relevant volume.

It is defined as:

$$E_c(T) = E_{c0}\left\{1 - exp\left[\frac{E_0(T - T_\infty)}{2k_B T T_\infty}\right]\right\}^{-2} \tag{3-237}$$

where E is activation energy, k_B is Boltzmann's constant, and T_∞ is the temperature at and above which the infinite dose is required for crystalline to amorphous transition.

Some experimental values for $E_c(E_{c0})$ are given by F. L Vook [68]. In the BCA module, the value *f(**r**)=0.6* corresponds to a fully amorphized state and any additional energy deposited at point *r* does not contribute to the amorphization process.

Implantation Geometry

Figure 3-23 shows the orientation of the ion beam, relative to the crystallographic orientation of the substrate. There three major planes regarding ion implantation in crystalline materials, mainly:

- the implantation plane α,
- the surface plane Σ,
- and the simulation plane β.

The implantation plane is where the initial beam of incoming ions lays in. It equivocally defines the direction of the incoming beam -- tilt and rotation. If the orientation of the surface plane is [100], which is the *only* substrate orientation available currently in the Binary Collision approximation implantation module (`BCA` or `CRYSTAL` parameters), the offset of the rotation angle is the direction `<101>` on this plane. This means that the tilt angle, ϑ, specified by the `TILT` parameter in the `IMPLANT` statement will be the polar angle in laying this plane, while the rotation angle, φ, specified by the rotation parameter will be the difference of azimuths of the line where the implantation plane, α, crosses the surface plane, Σ, and the direction <101>. See Figure 3-23.

Note: Presently, the surface orientation (the ORIENT parameter in the INITIALIZE statement) does not have any affect in the crystal Monte-Carlo module and the surface orientation is always {100}.

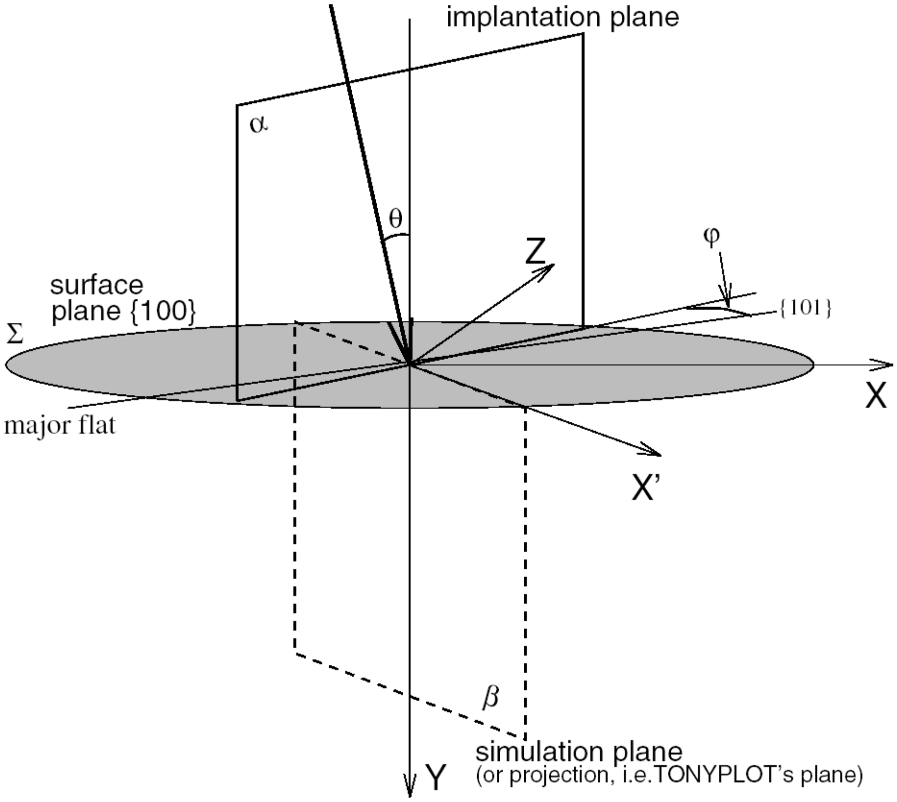


Figure 3-23 Implantation geometry

The simulation (projection) plane is where all data regarding the simulation is projected on, which is what finally goes into Athena's structure. The orientation of the simulation plane is specified by the `ROT.SUB` parameter in the `INITIALIZE` statement.

In summary, the laboratory coordinate system used in the BCA implant simulation is right-hand-sided -Y is depth, X is the other co-ordinate and Z is from the observer. The azimuth angle is measured in the X-Z plane. `ROT.SUB` is relative to X. Simulation plane is X'-Y.

The simulation plane (where `TONYPLOT` displays results and Athena calculates) is always perpendicular to the wafer surface and its orientation is specified by the `ROT.SUB` parameter in the `INITIALIZE` statement relative to the wafer major flat. By default, it's `ROT.SUB= = 45°`.

The implant calculation in bulk is 3D. Ray-tracing for BCA (i.e., calculation of ion impacts, scattering from walls and re-implantation) is 2.5D. In other words, structure is infinitely extended in the third dimension (along Z). All simulation results (doping, damage) are projected on the simulation plane and appropriately scaled.

`ROTATION` is measured from the *major flat* and in Athena’s case from the simulation plane because it is coupled with the wafer’s major flat.

For silicon (and other crystalline materials), you can think of `TILT`/`ROTATION` as always relative to the simulation plane giving *the same shadowing effects* while `ROT.SUB` defines which direction the simulation plane will slice the crystal structure through (in the laboratory coordinate system).

Specifying different `ROT.SUB` will have an effect on channeling. But remember, that ion propagation is three dimensional and there are some channeling patterns that remain the same or become stronger/weaker because of favorable/unfavorable initial impact conditions (`TILT`/`ROTATION`/`ROT.SUB` combination).

Implantation Geometry in SiC

Monte Carlo module of Athena allows accurate simulation of ion implantation in 4H-SiC and 6H-SiC [15]. It is known that channeling effects are very pronounced in SiC since its lattice has several wide open channels (e.g., <0001>, <11$\bar{2}$0>, and <11$\bar{2}$3> for 4H-SiC). Therefore, implant profiles strongly depend on wafer orientation and ion beam direction. Another important aspect of ion implantation simulation in SiC is the fact that most of commercially available standard <0001> wafers are offered with miscut of 3.5°–8.5° usually toward <11$\bar{2}$0> direction. The miscut is introduced primarily to facilitate epitaxial growth. In the same time, the miscut alters ion implantation because implanting normal to the wafer surface is equivalent to a tilt of the beam in the (11$\bar{2}$0) plane.

In order to perform Monte Carlo implant simulation in Athena, you should first specify simulation plane. The Monte Carlo module of Athena simulates ion trajectories in 3D and then registers the stopping points and consequently doping concentration by projecting 3D result onto simulation plane. The module also assumes that configuration of material interfaces don’t change in direction normal to this simulation plane. The simulation plane is always perpendicular to the wafer surface and its orientation is specified by the `ROT.SUB` parameter in the `INITIALIZE` statement relative to the wafer major flat (see Figure 3-24). Most commercial 4H-SiC and 6H-SiC wafers have major flat parallel to (1$\bar{1}$00) plane as in Figure 3-24.

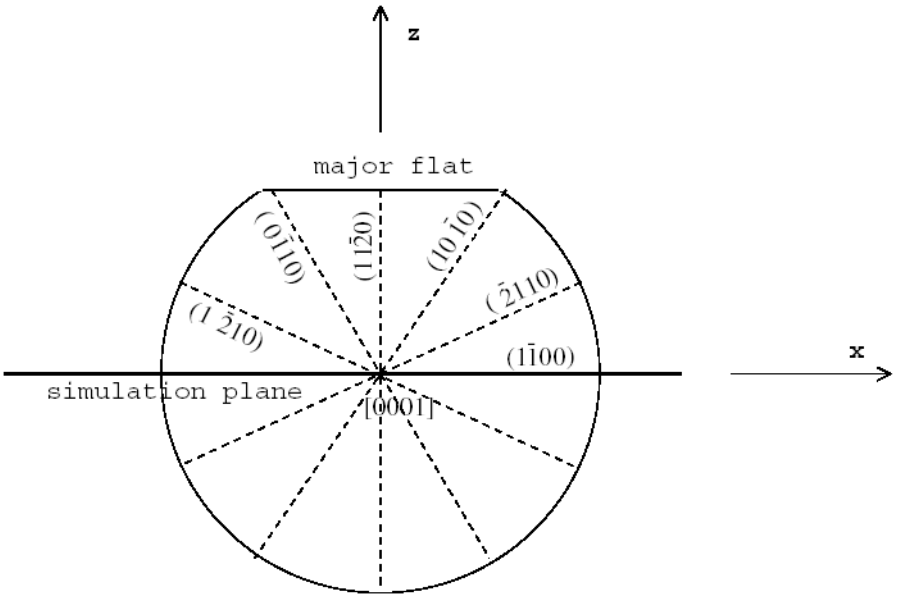


Figure 3-24 SiC wafer orientation relative to simulation plane in Athena BCA Module. Also shown are major plane projections on the wafer surface.

So if you want your Athena simulation plane to be parallel to the major flat, <$1\bar{1}00$>, you have to specify ROT.SUB=0. If you want your simulation plane to be parallel to <$11\bar{2}0$> crystallographic direction, you should specify ROT.SUB=90.

The ion beam direction is specified by the TILT and ROTATION parameters in the IMPLANT statement. The TILT is the angle (in degrees) between the ion beam vector and the normal to the wafer surface. The ROTATION is the angle between the ion beam vector plane perpendicular to the wafer surface and the simulation plane. For example, if the ion beam vector plane is parallel to <$\bar{2}110$> crystallographic direction in Figure 3-24, you have to specify ROTATION=30 (see Figure 3-25). Alternatively, if the ion beam vector plane is perpendicular to simulation plane, you have to specify ROTATION=90.

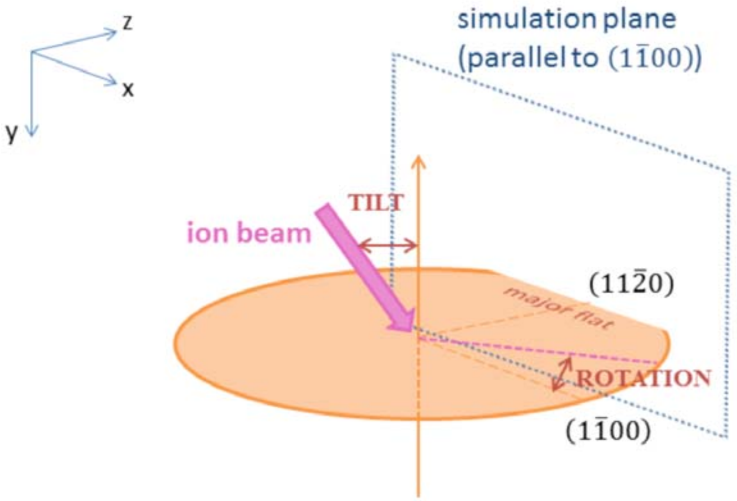


Figure 3-25 3D SiC wafer and simulation plane orientation. ROT.SUB=0 for simulation plane parallel to (1$\bar{1}$00)

The wafer miscut is specified by the MISCUT.TH and MISCUT.PH parameters. MISCUT.TH specifies the angle between actual wafer surface and the XZ plane (see Figure 1). Parameter MISCUT.PH specified the angle (measured counter-clockwise from the major flat direction) toward which the wafer was cut. So if the wafer is cut toward <1100> (i.e., X direction in Figure 3-24), you need to specify MISCUT.PH=0. This type of miscut is not used very often. The majority of commercial wafers (see, for example, [16] and [17]) have miscut toward <11$\bar{2}$0>. So practically in all cases, you need to set MISCUT.PH=90 and MISCUT.TH=4 or MISCUT.TH=8, depending of “off axis” angle in commercial wafer specifications (see [16] and [17]).

So, a typical initialize of Monte Carlo implant statements will be as follows:

```
init sic_4h rot.sub=0

implant aluminum energy=50 dose=1e13 bca n.ion=10000 tilt=4

rotation=90  miscut.th=4 miscut.ph=90
```

This will correspond to implant to a wafer with standard miscut usually specified as “Off axis 4° toward <11$\bar{2}$0> ± 0.5°” in commercial wafer specifications. The simulation plane will be parallel to the major flat and ion beam will be tilted by 4° in the plane perpendicular to the simulation plane.

The following examples closely demonstrate equivalent implant directions based on miscut MISCUT.TH/PH or TILT/ROTATION specifications.

Amorphous Material Monte Carlo

In the doping of semiconductors, the rest distribution of the implantations is of principal importance. The penetration of ions into amorphous targets is most simply described by using a Statistical Transport Model, which is the solution of Transport Equations or Monte Carlo Simulation. Among the two approaches, Monte Carlo is more convenient for multiple components and two or three dimensional targets, which is partly possible because the Monte Carlo method treats an explicit sequence of collisions, so the target composition can change on arbitrary boundaries in space and time.

The rest of the distribution is built up from a vast number of ion trajectories and the statistical precision of which depends directly on this number: $\propto \sqrt{N}$. As the ion penetrates a solid, it undergoes a sequence of collisions with the target atoms until it comes to rest. A simplified model of this interactions is a sequence of instantaneous binary nuclear collisions separated by straight line segments (free flight path lengths) over which the ion experiences continuous (non-local) electronic energy loss. The collisions are separated (i.e., the state of an ion after a collision depends solely on the state of the ion before the collision).

The model assumes that the arrangement of the target atoms is totally randomized after each collision (i.e., the target has no structure and no memory). As a result, a sequence of collisions is described by randomly selecting the location of the next collision partner relative to the pre-flight location and velocity direction of the ion. This means that this model cannot simulate the anomalous tail penetration observed for implanted ions into aligned single crystal targets. The model adequately describes the ion penetration into multilayer non-planar structures.

Crystalline Material Monte Carlo

The crystalline model used in Athena is based on the program `CRYSTAL` described elsewhere, [69]. In order to calculate the rest distribution of the projectiles, Athena simulates atomic collisions in crystalline targets using the Binary Collision Approximation (BCA). The algorithm follows out the sequence of an energetic atomic projectiles (ions) launched from an external beam into a target. The targets may have many material regions, each with its own crystal structure, (crystalline or amorphous) with many kinds of atoms. The slowing-down of the projectiles is followed until they either leave the target or their energy falls below some predefined cut-off energy.

The crystal model is invoked with the `MONTE` parameter in the `IMPLANT` statement. Athena will then choose which model to use depending on the predefined crystal structure of the material. Specifying `CRYSTALLINE` has no affect on the implantation and the BCA parameter is just a synonym for `MONTE`. You can manipulate the implantation module to consider all materials amorphous by adding the `AMORPHOUS` parameter in the `IMPLANT` statement. At that moment, the materials with predefined crystal structure are Si, Ge, GaAs, SiGe, InP as well as three types of silicon carbides (3C-SiC, 4H-SiC, and 6H-SiC) and two types of superconductors (Ba2YCu3O7 and Ba2NdCu3O7). All remaining materials in Athena are considered amorphous.

Statistical Sampling

In order to reduce calculation time and improve statistical quality of simulated profiles, Athena implements a three-dimensional rare event algorithm. An implantation profile can differ significantly in concentration values across implantation depth. Low concentrations in the profile are due to low probability of implanted species (rare events) to reach that point in space. Therefore, the number of cascades simulated to get good statistics profile depends on the desired number of orders of magnitude of accuracy. Even in real experiments, depending on device size, implant distributions below some threshold concentration value could exhibit significant statistical noise.

The algorithm uses trajectory splitting to achieve increased occurrence of the rare target events by generating several independent sub-trajectories from less rare events. The original idea [70], [71], and [72] was first developed into a refined simulation technique by Villién-Altamirano *et al.*, [73]. They call their version of this approach, Restart. The basic idea is to identify subspaces from where it is more likely to reach the target subspace where the rare event occurs. Each time these subspaces are reached, current events sequence are split in a number of replicas all continuing from the splitting state. The number of rare events will then increase, depending on the number of restart thresholds defined and the number of replicas generated.

The trajectory splitting algorithm naturally fits into the problem of Monte Carlo simulation of stopping and ranges (i.e., ion implantation). A similar method was first used by Phillips and Price to simulate hot electron transport [74]. Yang *et al.* were the first ones to apply the rare event algorithm was applied to simulation of transport phenomena of ions in matter [75]. Then, Beardmore *et al.* significantly refined this algorithm [76].

A brief but comprehensive review of trajectory splitting methods used in modelling of ion implantation is given in [77].

With the rare event trajectory splitting technique, the speed-up is due to changes in the statistical behavior so that rare events are provoked to occur more often. The rare event algorithm in Athena achieves this by identifying subspaces from where it is more likely to observe given collision event, followed by making replicas of the cascade sequences that reach these subspaces.

Figure 3-26 illustrates the trajectory splitting and restart of events (replicas) as a new threshold is reached. When applying splitting to collision cascades (or any other specific system), the two things that need to be determined are when to split and how many sub-trajectories to create when splitting.

There are different criteria that can be used to obtain the threshold states when splitting need to occur. For example, Bohmayr *et al.* use a trajectory split method based on checking the local dopant concentration at certain points [78]. Beardmore *et al.* rare event algorithm uses the integrated dose as a criterion when to split [76].

Athena uses the same criterion to determine the splitting depths. Dose integration is carried out along the radius vectors of ions' co-ordinates thus, roughly taking into consideration the three-dimensionality of the ion distribution.

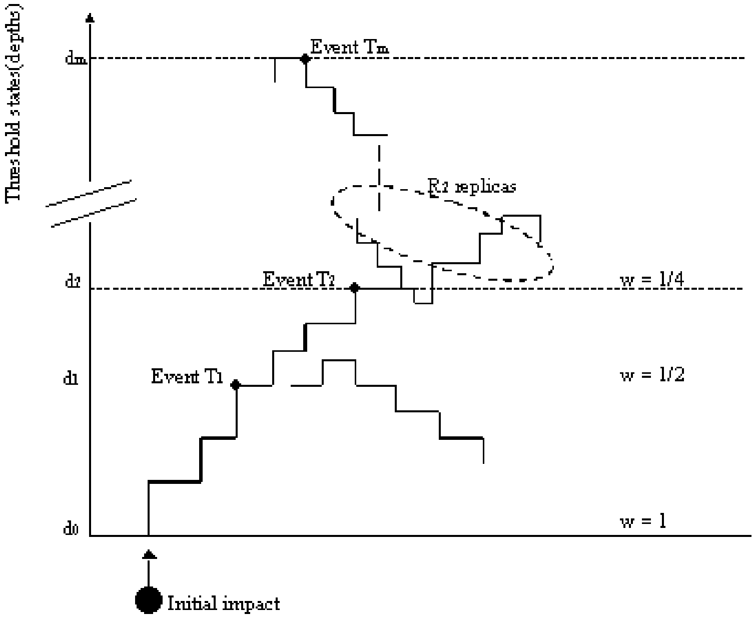


Figure 3-26 Restarting Collision Events by splitting at “m” thresholds

Due to the discrete nature of collision cascades, the number of sub-trajectories created at each split depth should be an integer number greater or equal to two. Suppose T_i is the event at each threshold state (i.e., this is the event of ion passing through a split depth). Suppose also the probability of an ion being in state T_{i-1} to reach the state T_i is $p_i=P(T_i|T_{i-1})$. Then, the recommended number of replications at each threshold (a split depth) is $R_i=1/P$. This relation gives the link between the number of replications at each split and the criterion to identify the threshold states (i.e., the split depths).

If $R_i=2$, then the number of ions passing through split depth will be twice smaller the number of particles passing through split depth. In Athena, the criterion to determine the split depths is the integrated dose along the radius vectors of stopped particles, d_i (i.e., split depths d_1, d_2, d_3, and so on will be at doses 0.5ϕ, 0.75, 0.875ϕ, and so on, where ϕ is the total retained implant dose).

In Athena, trajectory splitting is turned on with the `SAMPLING` parameter in the `IMPLANT` statement. For more information this statement, see Section 6.28 “IMPLANT”.

Theoretically, sampling estimators are unbiased and consistent. In practice, however, the estimate is obtained as the average of finite number of samples, as opposed to the theoretical expectation, which is obtained from the whole ensemble of sample paths, including even the very unlikely ones. Overbiasing can occur if the only goal in mind is to increase the probability of the event that needs to be analyzed further as is the case of trajectory splitting ion implantation simulations. The result of overbiasing is usually the underestimation of the probability to be evaluated (dopant concentration in case of ion implantation). In fact, it has been reported in [79] that when the splitting parameters are not consistent with the system's large deviations behavior, the probability in question may be underestimated by several orders of magnitude. This situation is almost present in ion implant simulators when treating multi-layered targets and two-dimensional layouts. Therefore, use splitting with caution.

In conclusion, you should bare in mind that:

- increasing the probability of occurrence of the event to be analyzed is not always enough to guarantee variance reduction.
- trajectory splitting should be used carefully.
- complex implantation geometries could lead to large deviations behavior of the system, thus overbiasing and underestimating the relevant statistics.

3.5.5 Ion Implantation Damage

Ion implantation induced crystal damage can play an important role in the various mechanisms related to diffusion and oxidation. Athena includes several different types of damage formation, which can be used in a subsequent diffusion calculation. Implantation induced damage results from cascades of atomic collisions. If these collisions cascades are dense, it may result in the crystal lattice becoming locally amorphized. Accurate simulation of collision cascades with simultaneous estimation of generating various types of point defects, clusters, and spatial defects can be done only in elaborated Binary Collision Approximation (BCA) or Molecular Dynamics (MD) simulators. Such simulations are usually time consuming and impractical within general-purpose process simulators. Generally, the amount of damage and distribution of defects associated with it depend on the energy, species, and dose of implanted ions.

Athena includes several simple models that link various types of defect distributions with ion implantation distributions calculated using any of the models described in previous sections.

The following types of defects can be estimated:

- Interstitial profiles
- Vacancy profiles
- {311} Clusters
- Dislocation Loops

You can describe the damage types to the simulator during the analytical ion implantation process simulation step by scaling their distribution densities to the implanted profile.

Plus 1 Model

The first damage model is related to free point defects. Here, interstitials are scaled to the as implanted dopant profile with the scaling parameter `DAM.FACTOR=<n>`. This model is invoked with the `UNIT.DAM` flag on the `IMPLANT` line.

For example:

```
IMPLANT PHOS DOSE=5E14 ENERGY=45 UNIT.DAM DAM.FACTOR=0.001
```

This model is known as the Plus 1 model. In the case of low implantation doses, the value for `DAM.FACTOR` has been suggested to be equal to unity. Although perhaps valid at low doses, the related and subsequent diffusion mode `METHOD FULL.CPL` is not required in most cases. Therefore, this combination is an impractical approach. Recent research on RTA diffusion models (e.g., Stanford’s {311} Cluster model) has introduced other forms of damage. Thus, lowering the dependency of free point defects being initially set at a Plus 1 scaled profile.

The DAM.FACTOR parameter, when used with the {311} Cluster model, should have a far lower value in the order of 0.001. Note that this is an extremely sensitive parameter when studying shallow junction formation so use it carefully.

{311}Cluster Model

The {311} Cluster model introduces a bulk injection source of interstitials in addition to any other free point defects sources. Clusters are introduced during ion implantation, scaled to the dopant, and within two user-defined concentration thresholds. For example, you can scale clusters to 1.4 times the dopant concentration but only exists between the dopant concentrations of 1e19 and 1e17 cm^{-3}. This allows a scalable approach, where clusters will follow implanted dopant as energies and doses vary (see Figure 3-27). The following syntax to both switch on and control the cluster model damage scaling.

```
METHOD FULL.CPL CLUSTER.DAM
```

```
CLUSTER CLUST.FACT=1.4 MIN.CLUST=1e17 MAX.CLUST=1e19 PHOS
```

See “RTA Diffusion Modelling” on 145 on how to use the {311} clusters during RTA.

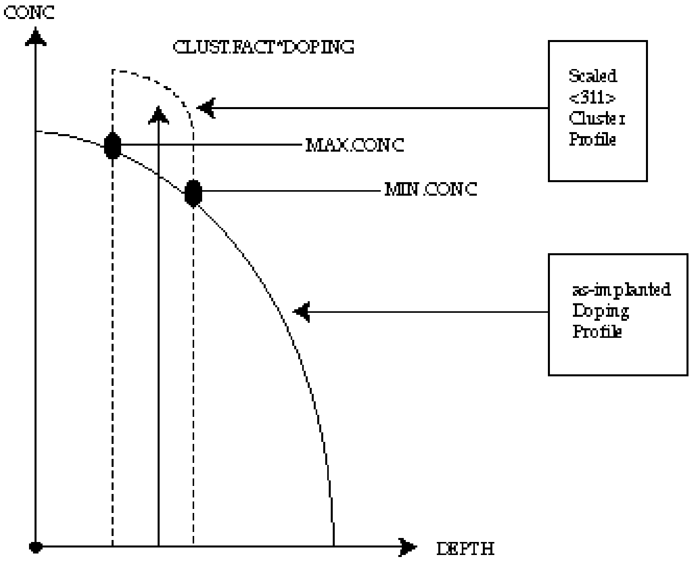


Figure 3-27 Cluster Damage Control

Dislocation Loops Model

Dislocation loops can also be scaled to the as-implanted dopant profile. Loops are introduced as a simple static band to act as an interstitial sink. Here, interstitials will be recombined at an enhanced rate according to

$$Rate = damalpha\left(\left(C_I - C_I^*\right)\right) \tag{3-238}$$

Here, C_I is interstitial concentration and C_I^* is equilibrium interstitial concentration.

Loops are placed in a band scaled to dopant concentration with the following command before implantation.

```
DISLOC.LOOP MIN.LOOP=1e16 MAX.LOOP=1e18 PHOSPHORUS
```

The recombination rate within the loop band is controlled as follows:

```
INTERSTITIAL SILICON DAMALPHA=1e8
```

C-Interpreter Model

The C-Interpreter capabilities shown in Appendix A “C-Interpreter” allows you to extend control over the damage formation models described. The template for the implant damage model function is also shown in Appendix A “C-Interpreter”. The function is introduced by setting the `DAM.MOD` parameter in the `IMPLANT` statement. The user-defined damage model introduced in the function will be used only within the current `IMPLANT` statement. All subsequent implants will use the default damage models.

3.5.6 Stopping Powers in Amorphous Materials and Range Validation

Stopping powers in amorphous materials have been validated against available experiments. Figure 3-28 shows a validation of boron and phosphorus ranges in amorphous silicon where compiled experimental data are taken from [64].

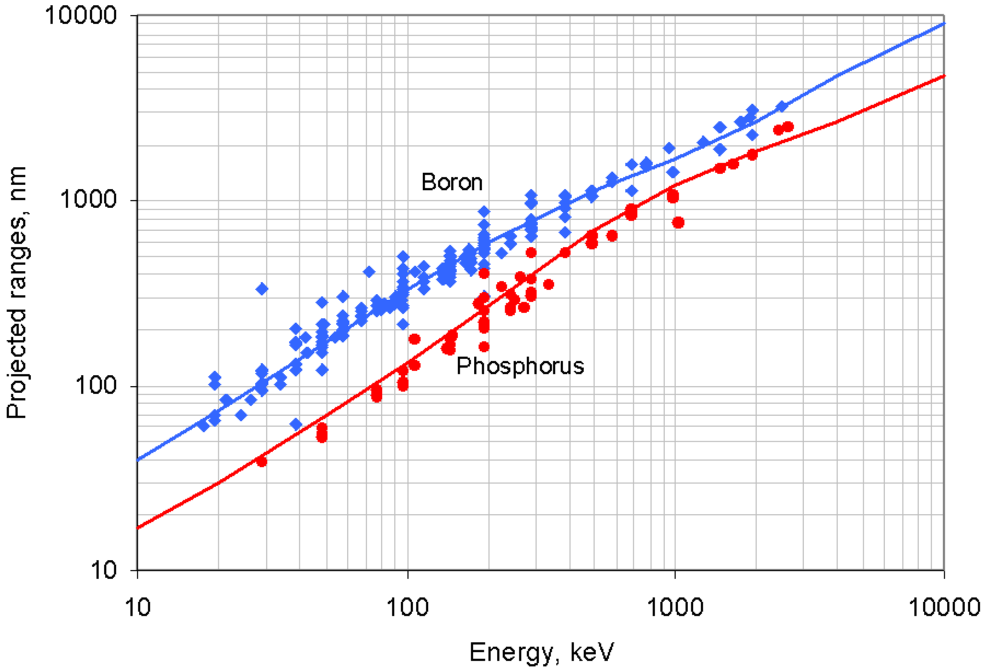


Figure 3-28 Comparison of Monte Carlo simulated project ranges (lines) and measured ranges (dots) for Boron and Phosphorus in Silicon. Experiments are from [**64**].

The solid lines were calculated with Athena's Monte-Carlo Module. The spread of the experimental points in Figure 3-28 is typical and cannot be avoided. For example, systematic errors due to the depth calibrations and memory effects in SIMS measurements if accounted improperly would yield less accurate (usually longer) ranges. Therefore, the Monte-Carlo module in Athena is calibrated to give overall agreement with the available experimental data. The figure also demonstrates that there can be a possible disagreement with individual set of measurements. Similar stopping powers validations were performed for other important materials. The accuracy of the calculated ranges in Athena is within 10% for majority of ion/ material combinations, which is close to the best possible achievements of today's theory of stopping and ranges.

3.6 Deposition Models

A deposition step is simulated by the DEPOSIT statement where the material and the thickness (THICKNESS parameter) of the layer to be deposited must be specified. The deposited layer is constructed by a simple algorithm that describes conformal deposition. In this algorithm, the whole layer is divided into a number of sublayers with thicknesses equal to grid spacings calculated according to the grid control algorithm (see Section 3.6.2 “Grid Control During Deposit”). Each sublayer is deposited and triangulated separately.

More complete physically based models for deposition are available in the Elite module as described in Chapter 4 “Elite Models”.

3.6.1 Deposition of Doped Layers

You can add the uniform or graded concentration of impurities or defects or both to each node of the deposited material by using the C.BORON, F.BORON, C.INTERST parameters in the DEPOSIT statement.

3.6.2 Grid Control During Deposit

You can control the grid inside deposited layer. The grid distribution along normal direction is controlled by a number of divisions (the user-defined parameter, DIVISIONS) in a uniform vertical grid. If non-uniform vertical grid is used, then specify the DY and YDY parameters in the DEPOSIT statement.

DY specifies nominal spacing and YDY specifies the position where to apply the nominal spacing. The spacings further from the nominal position YDY increase or decrease according to geometrical series. The coefficients of the geometrical series are calculated so that total number of the spacings will be equal to the DIVISIONS parameter.

When the conformal deposition algorithm fails to deposit next sublayer (which happens when deposition occurs on the structure with narrow trenches or undercuts or both), the current spacing divides into two and thinner layer is checked. This spacing division algorithm is applied recursively because in these situations, the number of sublayers (divisions) actually deposited can be higher than the number specified in DIVISIONS.

3.6.3 Epitaxy Simulation

SSuprem4 models high temperature deposition of single crystal silicon through the EPITAXY statement. This statement combines deposit and diffusion steps and parameters. See Section 6.18 “EPITAXY” for more information.

3.7 Etching Models

Although etching is an integral process step in silicon technology, SSuprem4 lacks a complete physical description of etching steps. To circumvent this problem, SSuprem4 considers etching simulation as a purely geometrical problem. Etching is simulated as a low-temperature process. Impurity redistribution is ignored during the etching process.

Etch steps are simulated using the `ETCH` statement in which the material to be etched and the geometrical shape of the etch region are specified. It is not necessary that material to be etched be exposed or at the top surface of the structure. There are five different ways to define an etch region:

1. A polygonal region may be defined by specifying the *x* and *y* coordinate of each vertex in the polygon. Etching will confined to that polygon only.
2. A region to the left or right of a line segment may be defined by specifying the x and y coordinates of the end points of the line segment. Etching will then proceed from the left or right of the line segment to the edge of the structure.
3. A region between the top boundary of the structure and a line obtained by translating exposed portion of the top boundary down in the y-direction may be defined by specifying the `DRY` parameter in the `ETCH` statement. The `THICKNESS` parameter will determine the distance to etch in the y-direction.
4. An extension of the `DRY` etch produces the etch region with slopped sidewalls and undercuts under the mask. The `UNDERCUT` parameter specifies the top boundary's extension of the etch region under the mask. The `ANGLE` parameter defines the slopes of sidewalls of the region. The bottom line of the etch region is defined by vertical translation of the top boundary with undercut taken into account.

   The `ANGLE` less than 90° results in trenches narrowing to the bottom. The `ANGLE` greater than 90° produces retrograde sidewalls. The `UNDERCUT` length is measured along the boundary line between etched material and masking layer.

   In a special case when the etched material layer is sandwiched between two other layers, the `THICKNESS` parameter is ignored and `UNDERCUT` is applied to both the upper and lower boundaries of the etched material layer.
5. All regions of a particular material may be etched by specifying the `ALL` parameter of the `ETCH` statement.

When a region is defined in one of the first three ways. By default, all materials in the defined region will be etched. Specifying a material in the `ETCH` statement limits etching to only that material within the defined region. For a complete description of physically based etch models, see Chapter 4 “Elite Models”. Elite is a complete 2D topography simulator included in the Athena framework.

3.8 Compound Semiconductor Simulation

ATHENA allows you to simulate basic technological processes in compound semiconductors. The set of standard compound materials includes: GaAs, AlGaAs, InGaAs, and InP. Additional user-defined materials can be also specified. The following chemical elements are typical dopants in these compound semiconductors: Si, C, Se, Be, Mg, Ge, and Zn.

3.8.1 Diffusion Models

The default diffusion model in compound semiconductors is the same as the Fermi Model with electric field effect used for silicon in SSUPREM4 (see Section 3.1.2 “The Fermi Model”). All diffusivity parameters from Table 3-2 can be specified for each dopant in all compound materials. But, only reasonably calibrated set of diffusion parameters exist for GaAs, [80]. You should perform calibration or all other materials.

Note: More advanced diffusion models TWO.DIM and FULL.CPL can be potentially specified for compound semiconductors. The point defects kinetics, however, is largely unknown which means that extensive research and calibration is needed

It was determined in [80], that n-type dopants, Si, Se, and Ge in GaAs, diffuse through the Ga Vacancy Mechanism, while p-type dopants, Be, Mg, Zn, and C, diffuse through the Ga interstitial mechanism. This means that for donors in GaAs the diffusivity is calculated as follows:

$$D_{donor} = D^{x}_{AV} + D^{-}_{AV}\frac{n}{n_i} + D^{=}_{AV}\left(\frac{n}{n_i}\right)^2 \qquad \text{3-239}$$

To look up these diffusivity terms, see Table 3-2.

The intrinsic carrier concentration n_i is calculated by Equation 3-9 with the parameters `NI.0`, `NI.E`, and `NI.POW` taken from [81]. The experimental data cited in [80] show that diffusivity of Si and Se in GaAs can be considered as concentration independent and therefore only the first term of this equation is non-zero for these impurities. It appears that diffusion of Ge in GaAs is proportional to the second power of (n/n_i).

The diffusivity for acceptors is the following:

$$D_{acceptor} = D^{x}_{AI} + D^{-}_{AI}\frac{p}{n_i} + D^{=}_{AI}\left(\frac{p}{n_i}\right)^2 \qquad \text{3-240}$$

Different terms are dominant for different acceptors. Carbon diffusivity is considered as concentration independent. Be and Mg diffusivities are proportional to p/n_i, while diffusivity of Zn is proportional to $(p/n_i)^2$.

It’s important to know that some dopants in compound semiconductors are amphoteric and can be either donor or acceptors under certain conditions. This means, you can use the `DONOR` and `ACCEPTOR` parameters in the `IMPURITY` statement to specify the type of dopant.

Boundary and interface condition for impurities in compound semiconductors are specified using the transport velocity parameters `TRN.0` and `TRN.E` and the segregation coefficients `SEG.0` and `SEG.E`.

The impurity activation in compound semiconductors is calculated using solid solubility model with default value for solid solubility limit for all impurities set at 10^{19} cm^{-3}.

3.8.2 Implantation Models

Ion implantation models for compound semiconductors are essentially the same as those for silicon. The Pearson analytical approximation uses look-up tables derived from experiments [82] and calculations [62]. These tables correspond to amorphous approximation and ignore effects of material crystalline structure.

When accuracy of simulated as-implant profiles is important and ion implant channeling is pronounced, use the Monte Carlo BCA implant model. The Monte Carlo model takes into account both crystalline structure and composition of compound materials. The actual composition and density of the default ternary compounds, InGaAs and AlGaAs, as well as the user-defined materials must be specified in the `MATERIAL` statement.

3.9 SiGe/SiGeC Simulation

Several experiments (e.g., [83], [84], and [85]) revealed that boron diffusion in $Si_{1-x}Ge_x$ or $Si_{1-x}Ge_xC_y$ alloys may differ from diffusion in pure Si substrates. To simulate effects of Ge concentration fraction, x, and carbon concentration fraction, y, on boron diffusion several models were introduced in SSuprem4. To activate these models, set the `MODEL.SIGEC` parameter in the `METHOD` statement. For example

```
METHOD MODEL.SIGEC FULL.CPL
```

If only `MODEL.SIGEC` is set, the `MODEL.SIGEC FERMI` option will be taken by default.

3.9.1 Deposition of SiGe/SiGeC Epitaxial Layer

In this model, $Si_{1-x}Ge_x$ or $Si_{1-x}Ge_xC_y$ alloys are considered as heavily doped with Ge or Ge and C. Usually, layers with either constant or graded germanium content are formed by a special epitaxy process.

You can simulate the formation of the $Si_{1-x}Ge_xC_y$ layer with a constant Ge and C content using the `DEPOSIT` statement with a `C.GERMANIUM` parameter set equal to $y*N_{Si}$, where N_{Si} is the atomic density of undoped silicon equal to 5.0 10^{22} cm^{-3}. For layers with graded germanium content, use an additional parameter, `F.GERMANIUM`, in the `DEPOSIT` statement. The following statement:

```
DEPOSIT  SILICON  THICK=0.1  DIN=10  C.GERMANIUM=1e20  F.GERMA-
NIUM=5e21 C.CARBON=1e19
```

will create the 0.1µm layer of $Si_{1-x}Ge_xC_y$ with constant carbon concentration of 1.0 10^{19} cm^{-3} and with germanium concentration varying from 1.0 10^{20} cm^{-3} at the bottom of the layer to 5.0 10^{22} cm^{-3} at the top of the layer. This corresponds to Ge content *x* varied from 0.2% to 10%.

3.9.2 Boron Diffusion in SiGe/SiGeC

The special model takes empirically into account experimental facts that boron diffusivity apparently decreases with germanium and carbon content. The total boron diffusivity (see Equation 3-10) decreases exponentially with the Ge content, x, and carbon content, y.

$$D_B[Si_{1-x-y}Ge_xC_y] = D_B[Si]exp\left(-\frac{x \cdot EAFACT \cdot SiGe + y \cdot EAFACT \cdot SIGE}{N_{Si}kT}\right) \tag{3-241}$$

Another effect taken into account in this model is variation of intrinsic carrier concentration n_i with x and y. It is presumed that n_i increased with x and decreased with y.

$$n_i[Si_{1-x-y}Ge_xC_y] = n_i[Si]\left(1 + \frac{x \cdot NIFACT \cdot SIGE}{N_{Si}}\right) \Big/ \left(1 + \frac{y \cdot NIFACT \cdot SIC}{N_{Si}}\right) \tag{3-242}$$

The user-defined calibration parameters (`EAFACT` and `NIFACT`) for the above equations are specified in the `MATERIAL` statement for silicon.

As an alternative to Equations 3-241 and 3-242, you can use different dependencies for diffusivity and intrinsic carrier concentration through C-Interpreter function specified by `SIGECDF.MOD` and `SIGECNI.MOD` parameters in the `METHOD` statement.

3.9.3 Boron Transient Diffusion Suppression by Carbon Incorporation Models

There are experimental indications [85] that interstitials diffuse slowly and tend to disappear or get trapped more intensively in SiGe layers with substitutional carbon. These effects result in suppressing of the boron transient diffusion when carbon is incorporated into SiGe layer. The following equations shows the diffusivity of interstitials, D_I, is controlled by the DCARBON.E parameter.

$$D_I[Si_{1-x-y}Ge_xC_y] = D_I[Si] \cdot exp\left(-\frac{y(DCARBON \cdot E)}{N_{Si}kT}\right) \quad 3\text{-}243$$

This model also introduces an additional sink for interstitials in the layers with high carbon concentration. Intensity of the sink is proportional to the carbon concentration and is controlled by the recombination parameters, KCARBON.0 and KCARBON.E, specified in the "INTERSTITIAL SILICON" statement.

This means that the following recombination term will appear on the right-hand side of the interstitial transport equation Equation 3-24.

$$R_{CARBON} = KCARBON \cdot 0\, exp\left(-\frac{KCARBONE}{kT}\right) \cdot \frac{y}{N_{Si}}(C_I - C_I^*) \quad 3\text{-}244$$

3.10 Stress Models

Athena provides several capabilities to calculate stresses generated during various process steps. Simulation of the stresses generated during viscous oxidation is discussed in Section 3.3.3 “Viscous Model”. Three other sources of stresses could be simulated in Athena. These stresses are generated by:

- the intrinsic stress in deposited films
- the stresses induced by local lattice distortion in presence of impurity atoms
- the stresses due to thermal expansion mismatch between material layers

To calculate the stresses generated in the current structure, use the `STRESS` statement. You can do this at any moment of the processing sequence usually after deposition or etch or both of the layer with specified intrinsic stresses or after formation of a highly doped layer or area by implantation or epitaxy process. Alternatively, Athena could follow the stress history if you specify the `STRESS.HIST` parameter in the `METHOD` statement. To calculate resulting stresses, Athena solves the following system of equations using a finite element analysis.

$$\frac{\partial\sigma_{xx}}{\partial x}+\frac{\partial\sigma_{xy}}{\partial y} = 0 \tag{3-245}$$

$$\frac{\partial\sigma_{yy}}{\partial y}+\frac{\partial\sigma_{xy}}{\partial x} = 0 \tag{3-246}$$

$$\sigma_{xx}+\sigma_{yy} = \frac{E}{(1+\upsilon)(1-2\upsilon)}\left(\frac{\partial u_x}{\partial x}+\frac{\partial u_y}{\partial y} - 2(1+\upsilon)\int_{T_1}^{T_2} LCTE\cdot dt\right)+\sigma_{xx}^{int}+\sigma_{yy}^{int}+2\sigma^{int} \tag{3-247}$$

$$\sigma_{xx}-\sigma_{yy} = \frac{E}{(1+\upsilon)}\left(\frac{\partial u_y}{\partial y}-\frac{\partial u_x}{\partial x}\right)+\sigma_{xx}^{int}-\sigma_{yy}^{int} \tag{3-248}$$

$$\sigma_{xy} = \frac{E}{2(1+\upsilon)}\left(\frac{\partial u_x}{\partial y}+\frac{\partial u_y}{\partial x}\right) \tag{3-249}$$

This system of equations corresponds to 2D formulation of the linear elastostatics problem for an isotropic homogeneous media. This means that all forces applied to the elastic structure sum to zero and displacements are not a function of time. Equations 3-245 and 3-246 are equations of motion in equilibrium conditions and Equations 3-247 through 3-249 are constitutive equations for elasticity formulated through displacements. Here, σ_{xx}, σ_{yy}, and σ_{xy} are calculated components of the stress tensor, u_x and u_y are displacements in x and y directions.

Note: Displacements u_x and u_y are saved in the structure file as velocity X and velocity Y respectively and can be used in `EXTRACT` commands or visualized in TonyPlot.

The elastic properties of each material in the simulation structure are specified through two elastic moduli: Young modulus E and Poisson's ratio υ. In the case of cyrstalline materials, the values of E and υ depend on the crystallographic plane defining material surface by the parameters `100`, `110`, or `111` in the `INITIALIZE` statement) and on specific orientation within the plane or angle measured from the wafer flat direction (which is specified by `ROT.SUB` in the `INITIALIZE` statement). For the cubic lattice, these dependencies are as follows [140]:

for surface plane {100}

$$E = \frac{1}{s_{11} - 2S(1-\cos^2\alpha)\cos^2\alpha}$$
$$\upsilon = -\frac{s_{12} + 2S\cos^2\alpha(1-\cos^2\alpha)}{s_{11} - 2S\cos^2\alpha(1-\cos^2\alpha)} \qquad 3\text{-}250$$

for surface plane {110}

$$E = \frac{1}{s_{11} - \frac{1}{2}S\cos^2\alpha(4-3\cos^2\alpha)}$$
$$\upsilon = -\frac{s_{12} + \frac{3}{2}S\cos^2\alpha(1-\cos^2\alpha)}{s_{11} - \frac{1}{2}S\cos^2\alpha(4-3\cos^2\alpha)} \qquad 3\text{-}251$$

for surface plane {111}

$$E = \frac{4}{2s_{11} + 2s_{12} + s_{44}}$$
$$\upsilon = -\frac{s_{11} + s_{12} - \frac{1}{2}s_{44}}{3\left(s_{11} + s_{12} + \frac{1}{2}s_{44}\right)} \qquad 3\text{-}252$$

where s_{11}, s_{12}, and s_{44} are elastic constants (compliances) specified by the parameters `S11.ELAST`, `S12.ELAST`, and `S44.ELAST` in the `MATERIAL` statement. $S = s_{11} + s_{12} - \frac{1}{2}s_{44}$ and α is the substrate rotation angle (`ROT.SUB`).

In case of amorphous material, the relations between E and υ and constants are simplified as follows: S=0, E=1/s_{11}, and υ=-s_{12}/s_{11}. Therefore, the following formulae can be used for obtaining elastic constants from Young modulus E and Poisson's ratio υ:

$$s_{11} = 1/E \qquad 3\text{-}253$$

$$s_{12} = -\upsilon/E \qquad 3\text{-}254$$

$$s_{44} = 2\cdot(1+\upsilon)/E \qquad 3\text{-}255$$

The second term of Equation 3-247 describes extrinsic stress generated by thermal expansion mismatch between material layers, for example, substrate and overlying films. The linear expansion coefficient, LCTE, is specified as a function of absolute temperature T in the MATERIAL statement. This function is integrated between temperature T_1 and T_2, which are either specified as TEMP1 and TEMP2 in the STRESS statement or correspond to temperatures at the start and end of process steps if STRESS.HIST is used (see Section 3.10.2 "Stress History").

The term σ^{int} in Equation 3-247 corresponds to the initial intrinsic stress in a deposited material layer. This intrinsic stress in grown or deposited films depends on many factors including thickness, deposition method, rate, temperature, and ambient.

Athena cannot simulate all these effects. You can, however, specify the initial intrinsic stress value INTRIN.SIG in the MATERIAL statement.

3.10.1 Dopant-Induced Intrinsic Stresses

The last terms σ_{xx}^{int} and σ_{yy}^{int} in Equations 3-247 and 3-248 represent local intrinsic stresses induced by dopants. The size of impurity atoms can differ significantly from that of the substrate silicon. Smaller dopants like B and P result in local lattice contraction, while bigger atoms. For example, *Ge* in a substitutional site causes lattice expansion. The lattice expansion or contraction increases with doping concentration. The local strain can be expressed as $\varepsilon = \Delta a/a_{\mathrm{Si}}$, where Δa is local change of lattice constant and a_{Si} is the silicon lattice constant. According to the Vegard's law, the lattice constant is approximately proportional to the concentration of the constituent elements in "alloyed" materials. To take into account a possible deviations from the linear dependency, Athena allows quadratic dependence of intrinsic stress on local impurity concentration as follows:

$$\sigma_{xx}^{int} = E\varepsilon_{xx} = E(A_{xx} + B_{xx}x + C_{xx}x^2) \tag{3-256}$$

$$\sigma_{yy}^{int} = E\varepsilon_{yy} = E(A_{yy} + B_{yy}x + C_{yy}x^2) \tag{3-257}$$

where $x = C_A/N$, C_A is local impurity concentration and N is atomic density of the material (for example, 5 *10^{22} cm^{-3} for silicon). You can specify the parameters A_{xx}, B_{xx}, C_{xx}, A_{yy}, B_{yy}, and C_{yy} as follows in the IMPURITY statement.

The dopant-induced intrinsic stresses are included only when you specify the DOPANT.STRESS in the METHOD statement.

Note: If there are several local impurities in the structure, the intrinsic stresses are calculated by summation of above equations for each impurity.

High concentration of impurity may also affect elastic properties of the bulk material. For example, it is known that the Young modulus of SiGe is lower than that of pure Silicon and that it can be approximated as follows:

$$E = E_0 + A_y x \tag{3-258}$$

where E_0 is the Young modulus of Silicon, A_y is linear coefficient specified by parameter A.YOUNG in the IMPURITY statement, and x is Ge fraction in SiGe. Athena allows the same linear dependence for all impurities.

3.10.2 Stress History

If the STRESS.HIST method is specified, Athena then calculates stresses when the simulation structure changes after etching, deposition, epitaxy, and diffusion processes. The temperature (including ramp) specified for current process step is used in the calculation. The room temperature is used if the temperature isn't specified in the ETCH or DEPOSIT statement. The final stresses from the previous step is used as a initial condition for the subsequent step. If the temperature is changed between the end of one step and the start of another, the stress calculation with corresponding temperature ramp is automatically inserted.

When the STRESS.HIST specification is active, all oxidation steps will be simulated with VISCOUS model and stresses inside silicon will be calculated in the end of each oxidation step.

The STRESS.HIST will be switched off when stresses are calculated explicitly using the STRESS statement or when material reflow is simulated by using the REFLOW parameter in the DIFFUSE statement.

3.10.3 Stress-Induced Diffusion Enhancement/Retardation

Several studies have clearly demonstrated that mechanical stresses in silicon may affect impurity diffusion. Most of those studies were related to diffusion in SiGe and strained Silicon near the SiGe layers (for example, see [137]). Experiments and accompanying models show diffusion retardation in case of compressive stresses. It was also shown that diffusivity dependence on local strain could reasonably be approximated by an Arrhenius formula. Sheu et.al. [138] suggested to use this approximation for any stresses generated during Si processing.

To use the stress-induced diffusion model in Athena, specify the STRAIN.DIFF parameter in the METHOD statement. The local impurity diffusion coefficient D_S in presence of stresses is calculated as follows:

$$D_s(x, y) = D_I exp\left[-\frac{E_s(\varepsilon_{xx}(x, y) + \varepsilon_{yy}(x, y))}{kT}\right] \qquad \text{3-259}$$

Here:

- D_I is the local diffusivity calculated without stresses.
- E_S is the activation per volume change ratio (or strain), which is specified by the E.STRAIN parameter in the IMPURITY statement.
- $\varepsilon_{xx}=\sigma_{xx}/E$ and $\varepsilon_{yy} =\sigma_{yy}/E$ are the components of strain.

Sheu et.al. [138] have found that diffusion appears to be retarded in the presence of compressive stresses (or when $\varepsilon_{xx} + \varepsilon_{yy}$ is negative). Therefore, the values for E_S are negative for B, P, and A_s impurities.

Chapter 4
Elite Models

4.1 Overview

The Elite module of Athena allows the use of sophisticated models for deposition and etch processes. These processes are modeled by defining a machine and invoking the machine to perform either deposit or etch. Elite also includes a model for material reflow. Elite can also be licensed with modules for Monte Carlo deposition, Monte Carlo etching, and Chemical Mechanical Polishing (CMP).

In Elite, a number of default machines are defined so that specifying any process reasonably close to the standard is especially simple. Process modifications or additions are easily implemented by changing or adding individual machines without affecting the remainder of the simulator.

For all models except Monte Carlo deposition and Monte Carlo Etching, Elite uses a string algorithm to describe topographical changes that occur during deposition and etching processes. This chapter describes the models and techniques used in Elite and the command language used to access model parameters.

4.2 String Algorithm

The Elite simulation regime consists of a set of triangles that hold information on the materials that are being simulated. The string algorithm treats each of these interfaces as a set of segments that move in response to a particular process calculation.

As microfabrication technology becomes more complex, modeling each step of the manufacturing process is increasingly important for predicting the performance of the technology. Etching is a step that is universal in microfabrication. It may take place as the dissolution of a photoresist by an organic solvent, the etching of an oxide by an alkali, or the plasma etching of an electron resist. Whatever its physical details, the etching process can in many cases be modeled as a surface etching phenomenon. Etching simulation starts from an initial profile that moves through a medium in which the speed of etching propagation can be a function of position and other variables that determine the final profile.

Two major assumptions limit the generality of the string algorithm in Elite. First, the pattern to be etched is uniform in one dimension, so the problem can be solved using only two dimensions. For most microfabrication problems, the important cases involve the cross-sections of lines, so this model is directly applicable. In certain other cases, such as round holes, the symmetry of some cross-sections is such that the algorithm is still valid. The second major assumption is that the etch rate is a scalar function of position, and is independent of the direction of local etch front motion and the history of the front. In some real situations, this does not hold. PMMA, for instance, has been found to have a gel region at the resist-solvent interface during development, so the etch rate is a function of the history of the adjacent regions as well as of the exposure. Another case where the second assumption does not hold is in the so-called “preferential etching” where etching proceeds more quickly along certain crystal directions, making the etch anisotropic.

The algorithm described here is known as a “string algorithm” [86]. The etch front is simulated by a series of points joined by straight line segments, forming a string. During each time increment, each point advances perpendicularly to the local etch front as in Figure 4-1. A major portion of the algorithm adjusts the number of segments to keep them approximately equal in length. Other subroutines input the data and output the etch front.

Choosing suitable criteria for segment length was a major problem in developing the algorithm. It seemed that segments must be short enough so that any curve that developed would be well defined, i.e., there should be some maximum angle between adjacent segments, perhaps 0.1 radians. This criterion, however, led to a proliferation of segments in regions where the front was either expanding or contracting. The algorithms in Elite attempt to maintain approximately equal segment lengths. This results in position errors of about one-half segment length. The error can be reduced by decreasing the average segment length with a proportional increase in computation time.

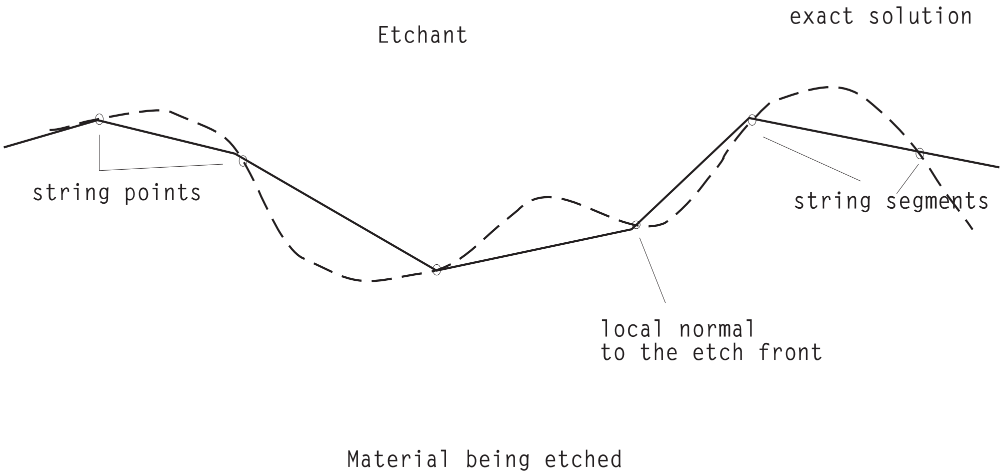


Figure 4-1 String Model approximation to the Etch Front

For the most cases of interest, the etch rate varies with position. This leads to some errors in the position and in the direction of each point on the string. Errors in position arise from the use of a rather simple integration algorithm. The local rate at the start of each time step is assumed to be constant throughout the step. This can easily lead to position errors as large as the distance covered in one step. Consider, for example, an etch front in a photoresist approaching an unetchable substrate. A point, which is barely outside the substrate at the start of the time step, will advance into the substrate at the rate associated with the resist. Thin layers of alternating fast and slow etch rates could spawn errors in position. With too large a time step, a point could jump over a slow region.

Errors in direction arise from non-uniform rates along the string and from certain boundary conditions. During each step perpendicularity to the front, which is defined below, is assumed to be constant in direction. If two adjacent points have greatly differing rates, however, the quickly moving point cannot start turning towards the slower point until the end of the time step. This mechanism tends to introduce relatively small errors in position because the error is roughly proportional to the cosine of the angle error.

4.3 Deposition Models

Elite provides a set of deposition models that correspond to different physical deposition techniques. Most of the models were first developed at UC Berkeley [87], [88], [89], and [90] and were originally implemented in the topography simulator SAMPLE [91]. Any one of these models can be selected to define a "machine" for simulating processes on the structure. In addition, Elite provides a conformal deposition capability that can be used to define initial structures.

In most integrated-circuit processes, at least one layer of interconnect is formed by depositing and patterning an Al or Al alloy film. The trend toward lower temperature processing, combined with the very steep edge profiles produced by anisotropic dry etching processes, results in sharp step profiles, which are difficult to cover with a uniform film of metal.

4.3.1 Conformal Deposition

You can perform conformal deposition by specifying a material to deposit, a thickness, and a number of vertical grid spacings on the `DEPOSIT` statement. The conformal deposition model produces unity step coverage.

4.3.2 CVD Deposition

To use this model, specify the `CVD` parameter in the `RATE.DEPO` statement as well as the material type, the deposition rate `DEP.RATE`, and step coverage, `STEP.COV`.

The local deposition rate R(x,y) for the `CVD` model is given by

$$R(x, y) = DEP.RATE[(1 - STEP.COV)\cos\theta + STEP.COV] \quad 4\text{-}1$$

where θ is the angle between the surface segment and the horizontal.

4.3.3 Unidirectional Deposition

To specify this model, specify the `UNIDIRECT` parameter in the `RATE.DEPO` statement.

As shown in Figure 4-2, the region of the substrate not shadowed sees the arrival of the vapor streams in one direction only. The growth rate of the deposited film in the shadowed region is equal to zero. According to these assumptions, growth rate on the substrate `R(x,y)` can be expressed as:

$$R(x,y) = 0\text{, if point (x,y) is shadowed} \quad 4\text{-}2$$

$$R(x,y) = C\sin\omega\,\boldsymbol{i} + C\cos\omega\,\boldsymbol{j} \quad 4\text{-}3$$

where:

- ω is the angle between the y-axis and the direction of the vapor stream.
- *i* and *j* are the unit vectors in the x and y direction respectively.
- C is the growth rate of an unshadowed surface normal to the vapor stream.
- Angle ω is specified as `ANGLE1` on the `RATE.DEPO` command.

The following is a short description of the cosine law deposition. This is a simple model that accounts for metallization, due to evaporation. The cosine law deposition model is based on the following assumptions:

- The mean free path of atoms or particles is much larger than the distance between the source and the substrate.
- The source to substrate distance is large compared to the surface topography.
- The film grows in the direction toward the vapor flux.
- Shadowing effects must be included.

The magnitude of the film growth rate follows the cosine distribution law, which says that deposited film thickness grows at a rate proportional to *cos(*ω*)*, where ω is the angle between the vapor steam and the normal surface.

The sticking coefficient can be used as a tuning parameter. It is assumed to be 1.0 for deposition on the cold substrates (at 300 K).

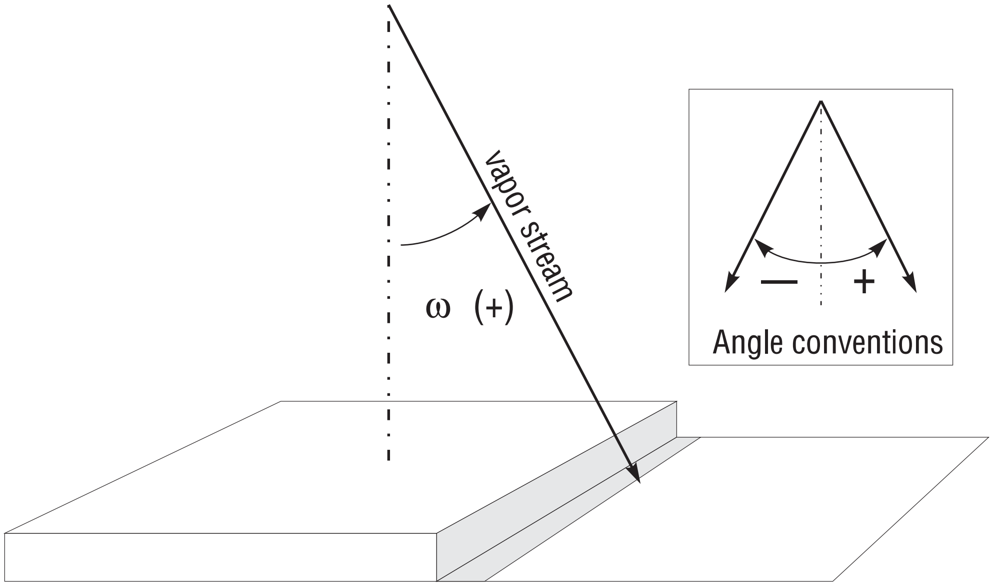


Figure 4-2 Step Profile with a Unidirectional Source

4.3.4 Dual Directional Deposition

This model is invoked by specifying the `DUALDIRECT` parameter in the `RATE.DEPO` statement. In this type of source, each point in the unshadowed region views the vapor streams arriving from two different directions, and assumes the diffusion length of deposited material large compared to the features (see Figure 4-3). Growth rate is given as:

$$R(x, y) = 0, \quad if \quad point\ (x, y) \quad is\ shadowed \tag{4-4}$$

$$R(x, y) = C\sin\omega_1\,\boldsymbol{i} + C\sin\omega_1\,\boldsymbol{j} \quad or \quad R(x, y) = C\sin\omega_2\,\boldsymbol{i} + C\sin\omega_2\,\boldsymbol{j} \tag{4-5}$$

if point (x,y) is partially shadowed.

$$R(x, y) = C(\cos\omega_1 + \cos\omega_2)\boldsymbol{i} + C(\sin\omega_1 + \sin\omega_2)\boldsymbol{j} \tag{4-6}$$

if point (x,y) is unshadowed, where ω_1 and ω_2 are the incident angles. ω_1 and ω_2 are specified on the `RATE.DEPO` command by `ANGLE1` and `ANGLE2`, respectively.

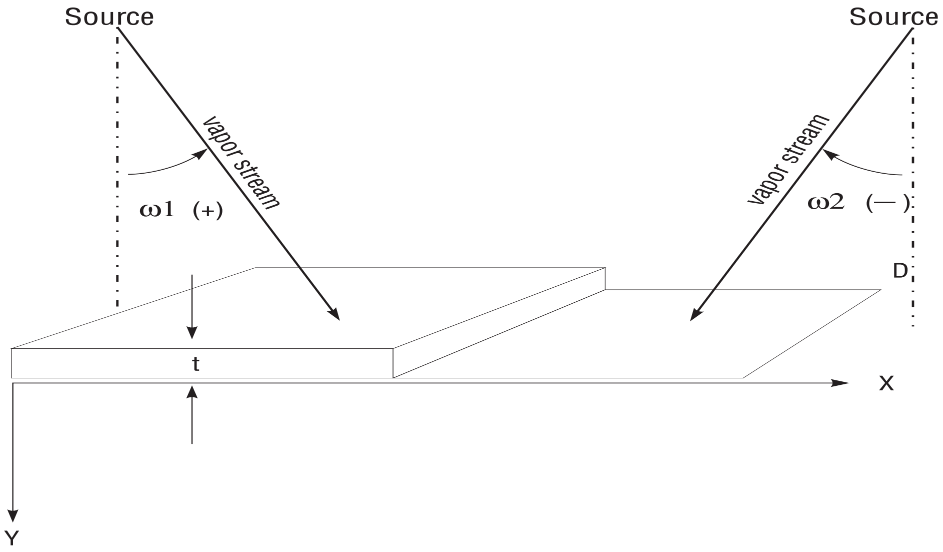


Figure 4-3 Step Profile with Dual Source

4.3.5 Hemispheric Deposition

To use this model, specify the HEMISPHERIC parameter in the RATE.DEPO statement

The flux of vapor is continuously distributed in a range of directions (see Figure 4-4). The growth rate can be calculated as:

$$R(x, y) = C(\cos\omega_1 - \cos\omega_2)\boldsymbol{i} + C(\sin\omega_1 - \sin\omega_2)\boldsymbol{j} \qquad 4\text{-}7$$

where ω_1 and ω_2 are the lower and upper bounds, respectively, of the incident angles of the vapor streams set by parameters ANGLE1 and ANGLE2, respectively.

To avoid step-coverage problems, planar sputtering is often used to achieve better film profiles. The ideal sputtering source is modeled by means of a hemispheric vapor source with atoms impinging on the substrate from all angles.

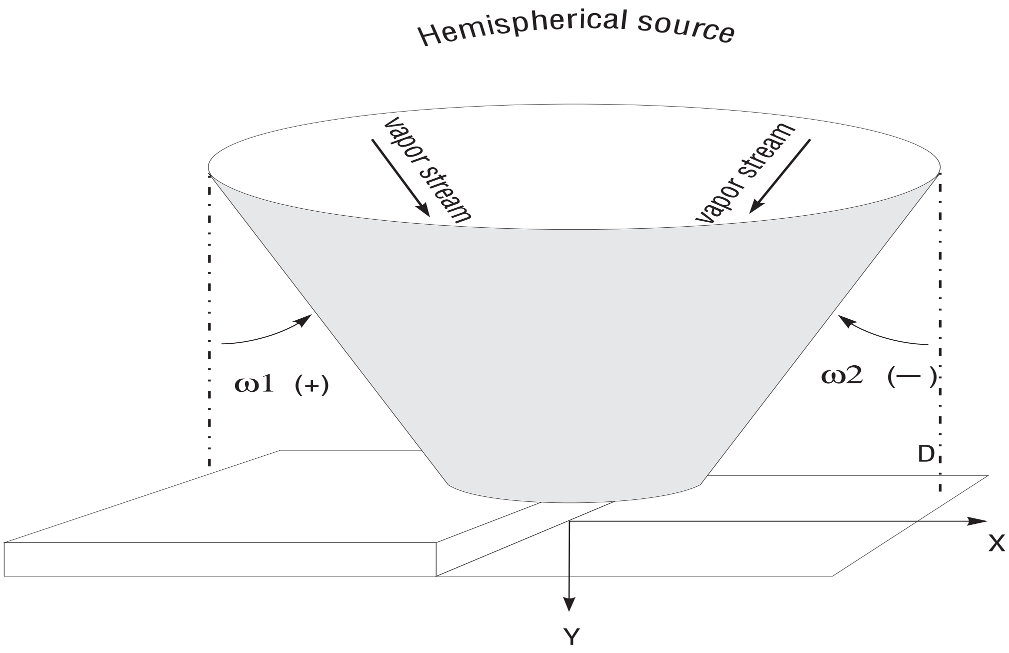


Figure 4-4 Step Profile with a Hemispherical Vapor Source

4.3.6 Planetary Deposition

To use this model, specify the `PLANETARY` parameter in the `RATE.DEPO` statement.

Figure 4-5 illustrates the planetary evaporation system. By inspecting this system, you can be convinced that the rotation of the planet along the system central axis has no effect on the deposition rate. For simplicity's sake, you can calculate the growth rate by holding the planet stationary and by rotating only the source along the axis of the planet (see Figure 4-5). The growth rate is derived according to the following equations:

$$R_x(x,y) = \int \frac{[R^2 - r^2 - rL\tan\Delta + LW][-L\sec(\Delta)^2][L\tan\Delta\sin\beta - L\cos\beta]\tan\delta}{[R^2 - r^2 + L^2 - 2rL\tan\Delta]^2\sqrt{(R^2 + W^2)[R^2 - (r + L\tan\Delta)^2]}}d\delta \qquad 4\text{-}8$$

$$R_x(x,y) = \int \frac{[R^2 - r^2 - rL\tan\Delta + LW][-L\sec(\Delta)^2][L\tan\Delta\sin\beta - L\cos\beta]}{[R^2 - r^2 + L^2 - 2rL\tan\Delta]^2\sqrt{(R^2 + W^2)[R^2 - (r + L\tan\Delta)^2]}}d\delta \qquad 4\text{-}9$$

where:

- δ is the incident angle of the vapor stream.
- β is the tilt angle of the planet plane.
- $\Delta = \delta - \beta$.
- r is the distance between the position of the wafer and the planet axis.
- R, L, and W are the parameters dependent on the system dimensions.

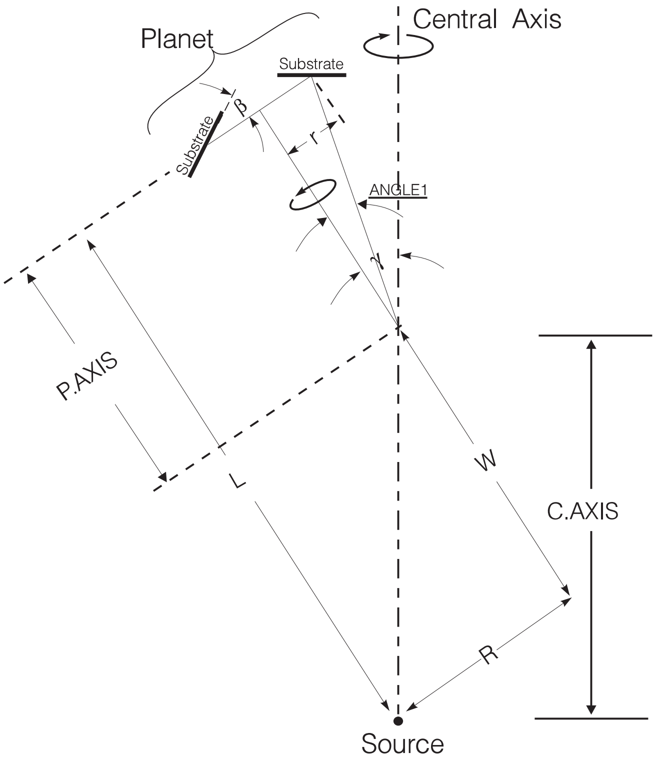


Figure 4-5 Illustration of Planetary Evaporator

Using the planetary model of Elite, you can observe asymmetries both in edge coverage and the depth of cracks produced by the particular location and orientation of a specimen in a planetary system.

Figure 4-6 shows the following planetary model parameters:

γ = `ANGLE2`, β = `ANGLE3`,

r = `DIST.PL`, `P.AXIS`, `C.AXIS`, and `ANGLE1`

`ANGLE1` is used to calculate `DIST.PL` = `P.AXIS` * tan(`ANGLE1`).

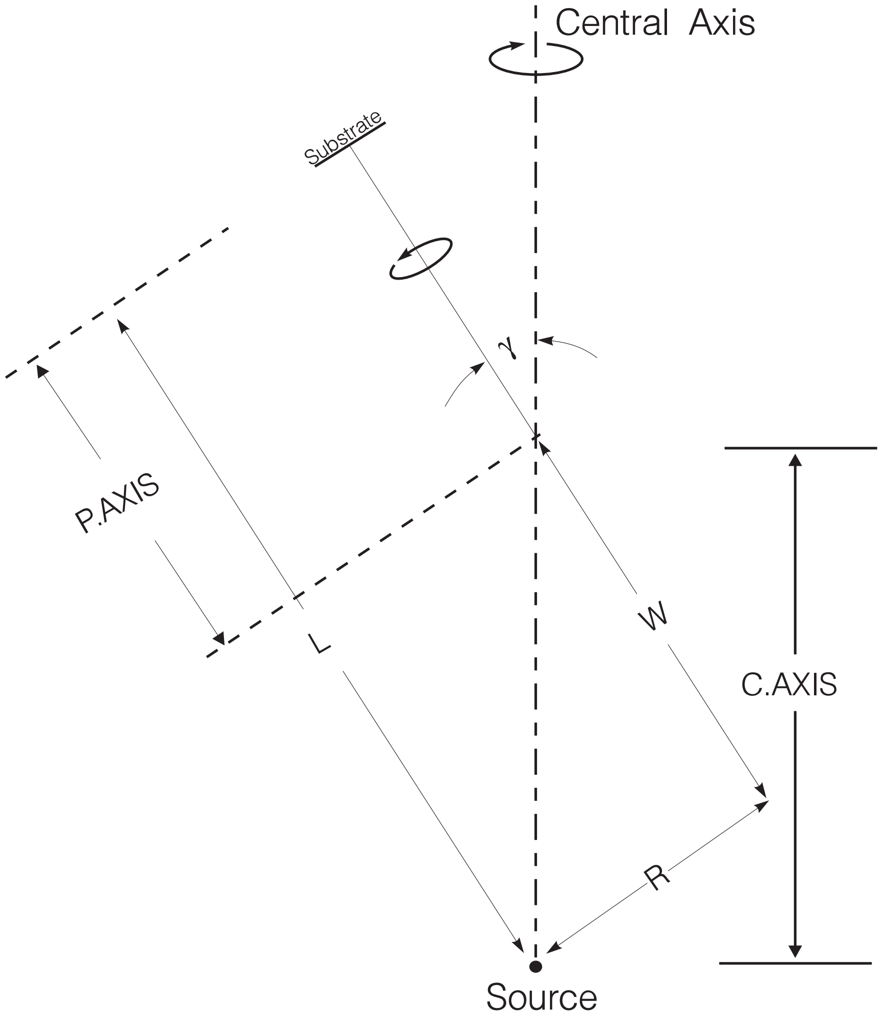


Figure 4-6 Geometric of Source to Substrate in a Conical Evaporator

4.3.7 Conical Deposition

To use this model, specify the `CONICAL` parameter in the `RATE.DEPO` statement.

The Conical model is a simplified version of the Planetary model with β and r = 0, the substrate always sees a symmetrical cone source. In this type of configuration, the integral of the above two equations can be evaluated analytically and expressed in the following simple closed form:

$$R_x(x,y) = \frac{-R(R^2+LW)}{\sqrt{R^2+W^2}\cdot(R^2+L^2)^2}\left\{\sqrt{1-\left(\frac{L}{R}\tan\delta_{max}\right)^2}-\sqrt{1-\left(\frac{L}{R}\tan\delta_{min}\right)^2}\right\} \tag{4-10}$$

If expression under either square root in this formula is less than 0 it is set to exact 0.

$$R_z(x,y) = \frac{L(R^2+LW)}{\sqrt{R^2+W^2}\cdot(R^2+L^2)^2}\left\{\operatorname{asin}\left(\frac{L}{R}\tan\delta_{max}\right)-\sqrt{\operatorname{asin}\left(\frac{L^2}{R^2}\tan\delta_{min}{}^2\right)}\right\} \tag{4-11}$$

In the Conical model, the parameter, `ANGLE1`=γ, and other parameters are `C.AXIS` and `P.AXIS` as shown in Figure 4-6.

4.3.8 Monte Carlo Deposition

There are two models that are invoked by specifying the MONTE1 or MONTE2 parameters in the RATE.DEPO statement. The parameters SIGMA.DEP, DEP.RATE, and ANGLE1. MONTE1 invokes the Monte Carlo based deposition model, which you can use to model low-pressure chemical vapor deposition (LPCVD) [92], [93].

Since the radicals are incident on the substrate with non zero thermal velocities, they may be re-emitted from the surface before they react. Therefore, the probability of their sticking is considered. You can define the sticking coefficient using STICK parameter in the RATE.DEPO statement.

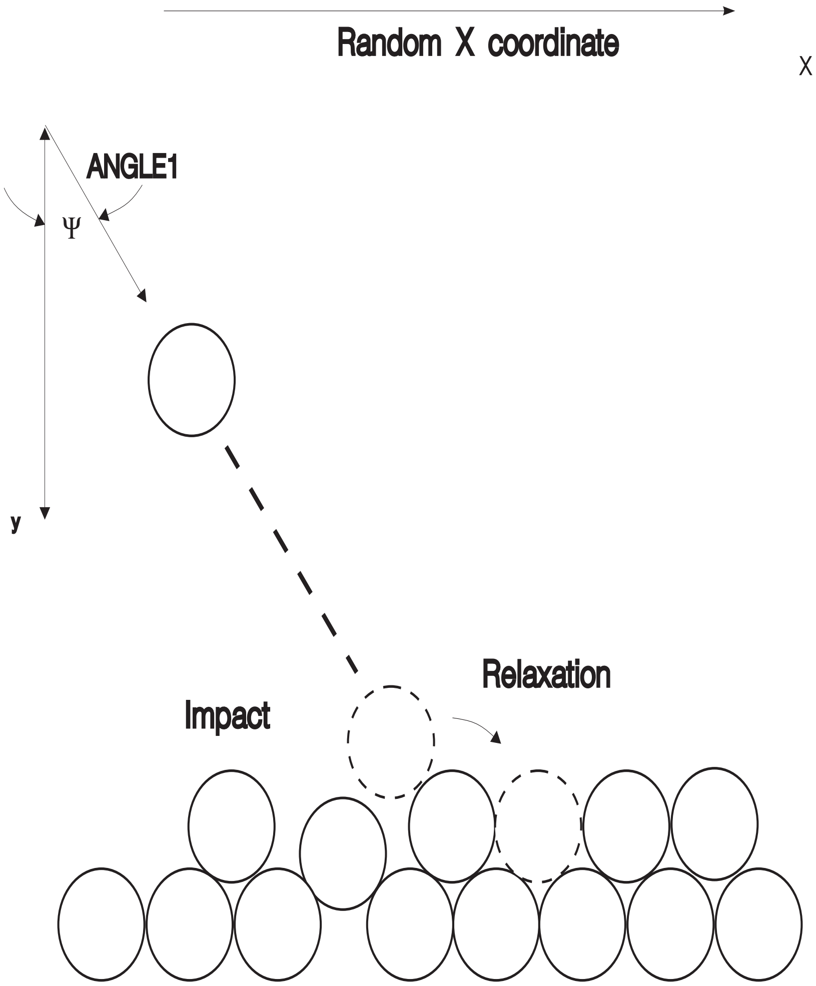


Figure 4-7 Deposition and Relaxation Model used in Ballistic Deposition model (MONTE2)

The model uses an analytical approach to calculate a surface diffusion through a normalized gaussian distribution *nd*:

$$nd = \exp\left(-\frac{x^2}{\mathbf{SIGMA.DEP}^2}\right) \qquad \text{4-12}$$

where x is the point of contact with the surface as shown in Figure 4-7.

MONTE2 invokes a ballistic deposition model, which simulates film growth by the random irreversible deposition of hard two-dimensional discs launched with linear trajectories from a random point at the top of the simulation area towards the structure surface.

At the point of contact with the growing film, the incident discs are relaxed to the nearest cradle point with the highest coordination number (contacting the largest number of neighbor discs) within a radius equal to SIGMA.DEP, which is four disc diameters by default.

The profile was initialized using a series of discs. In order to inhibit unrealistic “epitaxial” growth from a closest-packed surface [94] and [95], the initial series of discs was spaced with centers approximately 1.3 diameters apart.

This relaxation process simulates limited surface diffusion that occurs in films to reduce the surface energy associated with areas of high curvature.

A prediction of the trends in local film density can be achieved. Plot with discs can be obtained using the parameter OUTFILE=<filename> in the DEPOSIT statement. Figure 4-7 shows the vapor flux distribution arriving can be defined using the ANGLE1 parameter describing the angle measured between the vertical from the source and the wafer normal.

To use multiple steps for both MONTE1 and MONTE2 models, set the DIVISION parameter in the DEPOSIT statement. The number of incoming particles can be defined by the N.PARTICLE parameter in the DEPOSIT statement.

4.3.9 Custom Deposition Models

Elite implements two slightly different custom deposition models. In both models, the angle is *d*.

4.4 Etch Models

Elite provides a set of etch models that correspond to different physical etching techniques [96], [97], and [98]. Any one of these models can be selected to define a machine that can then be invoked to perform processing on the structure. In addition, Elite provides a primitive etching capability that can be used to define initial structures.

4.4.1 Isotropic Etch Model

To use the model, specify the `WET.ETCH` parameter in the `RATE.ETCH` statement. In wet etching and simple plasma etching, the substrate is immersed in a fluid (liquid or gas), which chemically reacts with the exposed surface. In wet etching, the liquid attacks the surface and removes soluble products. This reaction produces volatile by-products, which are removed by a vacuum pump. Physical reactions do not take place. A barrel plasma reactor achieves such conditions, usually at low powers and moderate pressures. Due to the chemical reaction, isotropic profiles develop with mask undercutting and circular cross-sections.

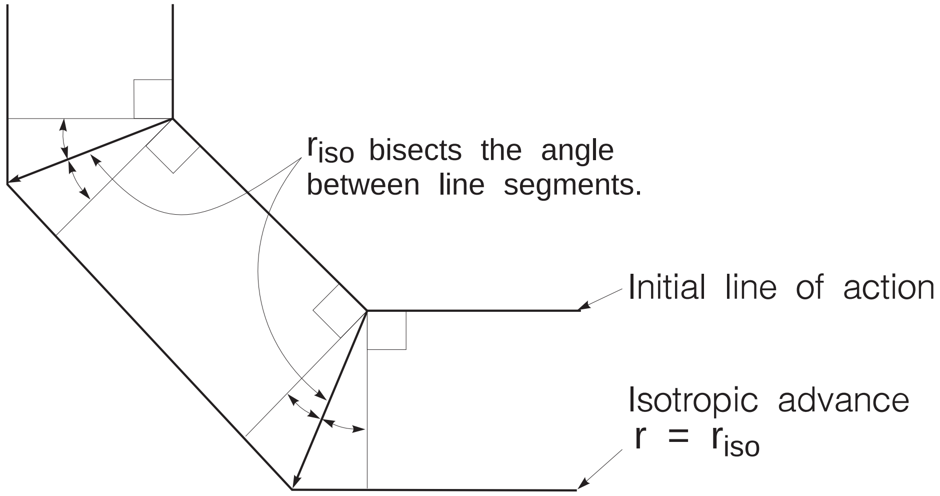


Figure 4-8 Segment Point in Case of Isotropic

4.4.2 RIE Model

In the Reactive Ion Etching (RIE) model, the etching process is divided into the two adjustable components isotropic etching and anisotropic etching. Each of these components is characterized by empirical etch rates (r_{iso} and r_{dir}).

The ratio:

$$A = \frac{r_{dir}}{r_{riso} + r_{dir}} \qquad 4\text{-}13$$

defines the measure of anisotropy.

The isotropic component r_{iso} models chemically reactive etching which results in profiles with undercut and circular cross-sections. For *A=0*, the process is completely isotropic. Under isotropic conditions, the string-points are advanced at the constant rate r_{iso}, in the direction of the perpendicular bisector of the adjacent segments (see Figure 4-8).

The anisotropic etch rate component r_{dir} is proportional to the cosine of the angle between the flux direction and the surface normal (the perpendicular bisector of adjacent segments). For *A=1*, the process is anisotropic yielding vertical sidewalls (see Figure 4-9).

Figure 4-10 illustrates the regions of significance for each component in the RIE model. The shadowing effect is accounted for by the r_{iso} component in the shadowed area.

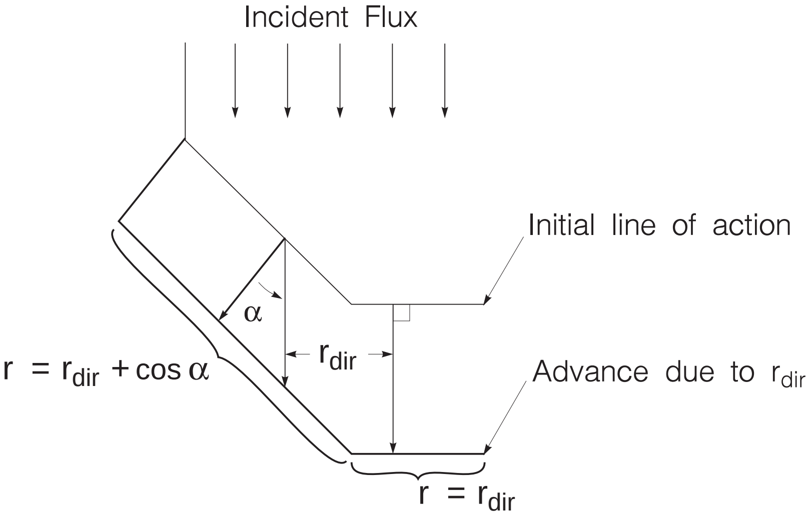


Figure 4-9 Point Advance due to Directional Influence

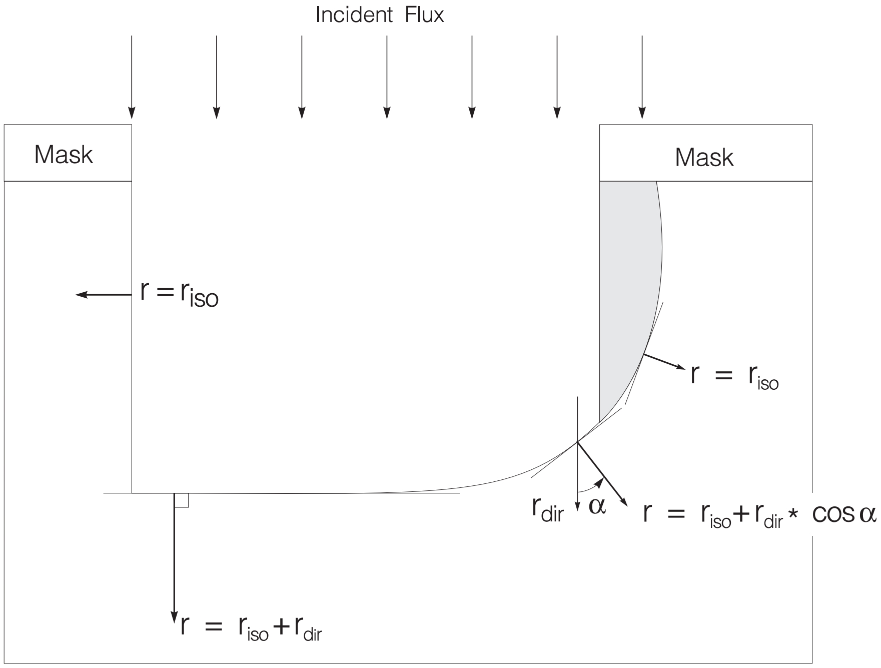


Figure 4-10 Regions of Significance of r_{dir} and r_{isc}

4.4.3 Dopant Enhanced Etching

Dopant enhanced etching is a feature included in Elite and allows the etch rate at any point on the surface to be changed depending on the value of any solution variable present. The etch rate at any point is then given by the formula

$$ER_{enh} = (1 + enh)ERM \qquad \text{4-14}$$

where ER_{enh} is the enhancement or retardation due the presence of particular dopant. All impurities as well as interstitials, vacancies, and stress solutions S_{xx}, S_{xy}, and S_{yy} can be specified in the model.

This enhancement is calculated using the formula

$$enh = 0.5\mathtt{ENH.MAX}(tanh(\mathtt{ENH.SCALE}(S - \mathtt{ENH.MINC}) + 1)) \qquad \text{4-15}$$

where `ENH.MAX` is the maximum value of enhancement or retardation, `ENH.MINC` gives the solution value below which enhancement decays and `ENH.SCALE` gives the spread of the enhancement over solution values. In other words, how quickly does the enhancement or retardation factor reach its maximum. *S* is the dopant value. The positive value of `ENH.MAX` corresponds to enhancement, while negative value corresponds to retardation.

For exponentially varying solutions, e.g., oxidation stress and dopant concentrations, both S and `ENH.MINC` are taken to be log base 10 of their respective value. Parameters of the model are specified in the `RATE.DOPE` statement.

4.4.4 Plasma Etch Model

The plasma etch model in Athena is based on a Monte Carlo simulation of the ion transport from the neutral plasma or bulk, denoted by its glow, through the dark sheath surrounding the electrodes and walls and isolating the plasma. Ions enter the sheath from the plasma and are then accelerated through the sheath due the electrical potential drop between the plasma and the electrodes. The Monte Carlo simulation follows a large number ions in their transport through the sheath including collisions with other gaseous species present in the etch chamber. The number of collisions encountered by a particular ion depends on both the ion mean free path, a calculated quantity, and the sheath thickness, an user-specified quantity. To reduce the computation time, ion trajectories are calculated independently and inter-ion interactions are not considered in this version of the code.

In the current version of Athena, the simulated Monte Carlo distributions are used to calculate an ion flux incident on the substrate surface. This flux is then used to calculate an etch rate by integrating this flux over the window of visibility at each point on the surface. The window of visibility is, for point on a flat surface, simply from 0 to 2π. But for more complicated structures (e.g., trenches), points on the surface are shadowed and the “window of visibility” is reduced. Currently, only a simple linear surface kinetic model for etching is supported.

See Chapter 3 “SSuprem4 Models” for a description of the `RATE.ETCH` parameters required for plasma etch simulation.

4.4.5 Monte Carlo Etching Model

The shrinking critical dimensions of modern technology place a heavy requirement on optimizing the etching of narrow mask opening. In addition, the aspect ratio of etches has been increased, requiring deeper etches along with the small kc/s. The simulation of these process requires more advanced techniques than the analytical rate-based etching models described above. A more complete treatment involving calculation of the plasma distribution and direct interaction of plasma particles with substrate materials is required.

The Monte Carlo etch module is implemented into Athena/Elite. The main application of the module is simulation of plasma or ion assisted etching. The module can take into account the redeposition of the polymer material generated as a mixture of incoming ions with etched (sputtered) molecules of substrate material. Also, the module has interface to the C-Interpreter, which allows simulation of several other processes such as wet etch and deposition, ion milling and sputtering deposition of various materials. The Monte Carlo etch module was successfully used by Toshiba researchers for simulations of reactive ion etching of narrow deep trenches in oxide [99].

Simulation of Incoming Ions and Neutrals

Direct modeling of the plasma sheath is not included into this release and will be added later. It is assumed that ions and neutrals fluxes leaving plasma sheath are represented by bimaxwell velocity distribution function along the direction determined by user specified incident angle:

$$f(\ \upsilon_{\parallel}\ ,\ \upsilon_{\perp})\sim I\cdot exp\left(-\frac{\upsilon_{\parallel}}{T_{\parallel}}-\frac{\upsilon_{\perp}}{T_{\perp}}\right) \tag{4-16}$$

where:

$\upsilon_{\parallel}$ is the ion velocity component parallel to the incident direction.

$\upsilon_{\perp}$ is the ion velocity component perpendicular to the incident direction.

I ion (or neutral) current density specified by parameters, MC.ION.CU1 or MC.ION.CU2 in the RATE.ETCH statement.

$T_{\parallel}$ is the dimensionless parallel temperature specified by parameters, MC.NORM.T1 or MC.NORM.T2.

$T_{\perp}$ is the dimensionless lateral temperature specified by parameters, MC.LAT.T1 or MC.LAT.T2.

The incident angles are specified by the MC.ANGLE1 and MC.ANGLE2 parameters.

Calculation of Ion and Neutral Fluxes

During each time step, the simulation consists of the three stages. These stages are as follows:

1. Calculation of ion, neutral, and polymer fluxes
2. Calculation of etch, polymer ejection and redeposition rates
3. Surface movement

On the first stage, the fluxes of incoming and reflected ions and neutrals are calculated on the each segment of the surface. Computation of the ion fluxes is done by tracing the user-defined number of particles (Figure 4-11, model a). Each particle is generated at random positions on top of the simulation area, with normal and lateral velocities randomly determined from the bimaxwell distribution function (Equation 4-1). Then, each particle trajectory is traced until the ion is either absorbed by the surface or back scattered out of the simulation area.

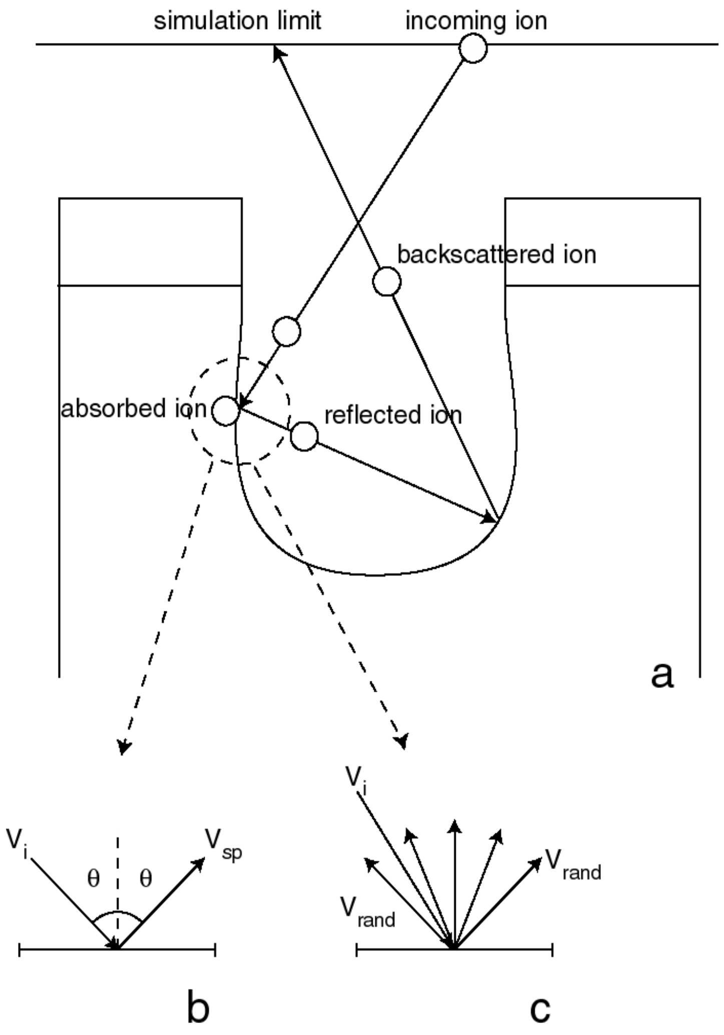


Figure 4-11 Diagram of Plasma Flux algorithm: (a) including zoom-in of ion reflection models (a & b)

The interaction of the ion with material surface is governed by two factors. The first is the reflection coefficient P_{refl}, which is specified by the MC.ALB1 and MC.ALB2 parameters for two types of plasma particles. The second is MC.PLM.ALB for polymer particles and roughness of the surface R, which is specified by the MC.RFLCTDIF parameter. Both factors depend on the surface material and the type of ion. Reflection coefficient is the probability of the particle to be reflected from the surface. Roughness determines how the ion is reflected. If R = 0 the reflection is specular (Figure 4-11, model b). If R = 1, the reflection is random with uniform angular distribution (Figure 4-11, model c). In general, the velocity $\boldsymbol{v}_{refl}$ of the ion after a collision with a surface segment could be presented as follows:

$$v_{refl} = 0(\text{ion is absorbed}) , \ \text{if}\, x > P_{refl} \tag{4-17}$$

$$v_{refl} = v_{sp} \cdot (1 - R) + v_{rand} \cdot R, \text{if} \quad x > P_{refl} \tag{4-18}$$

where:

- $\boldsymbol{v}_{sp}$ is the ion velocity after specular reflection.
- $\boldsymbol{v}_{rand}$ is the ion velocity after random reflection.
- x is a random number.
- $|\boldsymbol{v}_{sp}| = |\boldsymbol{v}_{rand}| = |\boldsymbol{v}_i|$, where $\boldsymbol{v}_i$ is the velocity of incident ion.

Each absorbed ion is used to compute the incoming flux F_i at the surface segment. The following characteristics describe the flux.

$$N_{norm} = N_{abs} \cdot I / N_{traj} \tag{4-19}$$

where:

- N_{norm} is the normalized number of absorbed particles,
- N_{abs} is the number of absorbed particles,
- N_{traj} is the number of trajectories specified by the MC.PARTS1 and MC.PARTS2 parameter for each type of plasma particles, and by the MC.POLYMPT parameter for polymer particles,
- normalized normal $v_{\perp abs}$ and tangential $v_{\| abs}$ velocity components of the absorbed particle before the encounter with the surface:

$$v_{\perp abs} = \frac{I}{N_{traj}} \cdot \sum_{N_{abs}} v_{\perp} \tag{4-20}$$

$$v_{\| abs} = \frac{I}{N_{traj}} \cdot \sum_{N_{abs}} v_{\|} \tag{4-21}$$

- normalized kinetic energy of absorbed particles:

$$v^2{}_{abs} = \frac{I}{N_{traj}} \cdot \sum_{N_{abs}} \boldsymbol{v}^2 \tag{4-22}$$

Calculation of Polymer Fluxes

After ion and neutral fluxes are determined, the fluxes of the polymer particles are calculated as follows. As the result of ion flux interaction with the surface segment, the polymer particles are generated. The angular distribution of the polymer particles is uniform and the current density of these particles is determined by the etch model (see “Linear Etch Model” on 253) and the sum of the fluxes from incoming ions, neutrals, and from polymer particles ejected from other surface segments. Obviously, the latter flux needs to be pre-calculated.

This flux is computed as follows. First, the configuration (or geometrical) factors, are calculated. These factors are the fractions of the number of particles ejected from one segment and absorbed by the other one. These are calculated using the same trajectory tracing algorithms, which are described above for the incident ions and neutrals with the only one difference: starting points are not at the upper boundary of the simulation area but at the surface segments. Then, an iteration process is initialized. At the first iteration, only the incoming ion and neutral fluxes are used for calculation of the ejection rates from each surface segment. Knowing the current densities of ejected particles and the configuration factors, the polymer fluxes are calculated. At subsequent iterations, the polymer fluxes calculated at the previous iteration are used to update the etch and ejection rates. The iterations are repeated until etch and ejection rates converge.

Calculation of Rates

The second stage involves calculation of the etching rates as well as ejection and redeposition rates of the polymer particles. During each time step, the two processes simultaneously take place on each surface segment. The first is redeposition of the polymer with the rate equal to the polymer flux. The second is etching by incoming ions and neutrals. The combination of these two processes can be treated as deposition of a virtual polymer layer with subsequent etching of the two-layer structure. If the etch rate of polymer by incoming ions and neutrals is less than the polymer deposition rate, the result is the redeposition of a polymer layer on the surface. If the etch rate of polymer by incoming ions and neutrals is larger than the polymer deposition rate, the result is actual etch of the underlying material.

Linear Etch Model

In the case of the linear model, the etch rate *ER(m)* of each material *m* is calculated as

$$ER(m) = \sum_n EP(m,i) \cdot v_{abs} \qquad \text{4-23}$$

where *n* is the number of plasma ion types specified by the parameter `ION.TYPES` (*n* could be equal to 1 or 2), *EP(m,i)* is the etch parameter for material *m*, and ion type *i* specified by parameters `MC.ETCH1` and `MC.ETCH2`,v_{abs} is the ion velocity as calculated in Equation 4-22.

If calculated, *ER(polymer)* is less than the polymer flux (redeposition rate), *PF* the actual etch rate, and *ER* is negative which corresponds to redeposition.

$$ER = ER(polymer) - PF < 0 \qquad \text{4-24}$$

The corresponding ejection rate *EJR* is equal to the etch rate of polymer.

$$EJR = ER(polymer) \qquad \text{4-25}$$

When calculated *ER(polymer)* is larger than polymer flux, the actual etch rate is positive.

$$ER = ER(m) - PF \cdot \sum_{n} \frac{EP(m, i)}{EP(polymer, i)} \qquad 4\text{-}26$$

The corresponding ejection rate is calculated as follows.

$$EJR = PF + ER \qquad 4\text{-}27$$

C-Interpreter

You can use the C-Interpreter to introduce different etch and ejection models. The following parameters are passed to the C-Interpreter file and can be used for implementing the models: number of ion types, the four characteristics of ion fluxes for each ion type (Equations 4-19-4-22), *PF*, and surface material, *m*. Returned parameters are *ER* and *EJR*.

For example, you can simulate the wet etching by setting the etch rate to a constant positive value depending only on the surface material. In this case, the trajectory tracing part of the model is not needed. The number of trajectories can be set to one.

Uniform deposition can be simulated by the setting of a negative constant etch rate and by specifying the redeposited material other than polymer in the `ETCH` statement. If the fluxes are not used, as in the wet etching simulation, the void formed will eventually be filled with the deposited material, because inside the C-Interpreter there is no way to determine if the current surface segment belongs to the void or not. This obstacle can be overcome by simulating ion fluxes and by setting the etch rate to zero if the flux on the surface segment is less than some small threshold value.

Surface Movement

A sophisticated string algorithm is used to move all segments, according to the rates (positive or negative) calculated at each time step. If the rate is negative, the surface moves outside and the area is filled with redeposited material (by default, polymer). If the rate is positive, the surface moves inwards and the area is filled with vacuum.

4.5 Reflow Model

A two-dimensional viscous reflow capability is included in Elite**.** The vitreous silica (e.g., oxide, BPSG,) are modeled as the viscous incompressible fluids, which are dynamically deformed under the driving force of surface tension. The finite-element method is used to solve the creeping flow equations for the chosen materials. With a 7-node triangle element as the basic discretization unit, arbitrarily shaped 2D regions and surface curvatures are automatically described. Using the built-in user defined material capability, you can simulate multiple material combinations. The flow equation solver can be coupled with impurity diffusion to simulate the impurity redistribution and oxide growth.

The reflow is invoked by setting the reflow flag in the `DIFFUSE` statement and by setting the `REFLOW` flag in the `MATERIAL` statement to choose a specific material. Physical parameters that are specific for various materials, such as viscosity and surface tension, are also given in `MATERIAL` statement. Reflow will proceed according to the time and temperature given in the `DIFFUSE` statement. The finite element solver are invoked by specifying the `VISCOUS` parametrs and various numerical control parameters in the `METHOD` statement.

The viscous creep flow equations solved are as follows (see [100]):

$$\mu\nabla^2\boldsymbol{V} = \nabla P \tag{4-28}$$

$$\nabla\cdot\boldsymbol{V} = -\left(\frac{1-2\nu}{\mu}\right)P \tag{4-29}$$

$$\mu = \frac{E}{2(1+\upsilon)} \tag{4-30}$$

where V is the velocity, P the pressure, μ the viscosity, ν the Poisson's ratio, and E the Young's modulus. The parameters v and E are specified through elastic constants as described in Section 3.10 "Stress Models".

4.6 Chemical Mechanical Polish (CMP)

Chemical Mechanical Polish (CMP) is a module in Athena. To run CMP, you need to have the license to use Elite. CMP is used to model wafer planarization using polishing pad and chemical slurry characteristics. CMP is also used to circumvent two major problems. First, the depth of focus of high numerical aperture lithography systems. Second, metal thinning that can occur over non-planar topographies

The CMP module that is incorporated into Athena has two distinct models. The first is the Hard Polish or Buzz Saw Model [101]. The second is the Soft Polish Model based on the work of J.Warnock [102]. To access these models, use the Athena statements: `RATE.POLISH` and `POLISH`. These statements are similar to those used for the Elite deposition, `RATE.DEPO` and `DEPOSIT`, and the Elite etching, `RATE.ETCH` and `ETCH`. The `RATE.POLISH` statement sets up the parameters for a particular machine, while the `POLISH` statement executes the actual polishing step using the machine.

4.6.1 Hard Polish Model

The Hard Polish Model [101] simulates the grinding down of the topography based on a rate calculated as a function of the pattern factor *Pf* of the surface. The higher the pattern factor, the lower the polishing rate. Use the following formula.

$$R(x,y) = MAX.HARD \cdot (1 - Pf) + MIN.HARD \cdot Pf \qquad 4\text{-}31$$

The hard polish model parameters are `MAX.HARD` and `MIN.HARD`. $Pf = 1$ corresponds to a flat surface. *Pf* is calculated from the topography by the formula:

$$Pf = \left(\sum_i (x_i - x_{i-1})\Big|_{y > Y_{max} - \Delta Y}\right) / X_{total} \qquad 4\text{-}32$$

where (x_i,y_i) are points on the polished material surface, *y* denotes both y_i and y_{i-1}, Y_{max} is the highest point of the structure and ΔY is the rate effective height calculated by the previous rate multiplied by the current time step value. For Figure 4-12, a pattern factor will be as follows:

$$Pf = \frac{\Delta X_1 + \Delta X_2}{X_{total}} \qquad 4\text{-}33$$

The rate for points at height Y_{max} are equal to the *R* calculated in Equation 4-31. Points below Y_{max} have a rate cause the structure to polish to the *y* coordinate $Y_{max} - \Delta Y$. Therefore, the structure becomes more planar as shown in Figure 4-13.

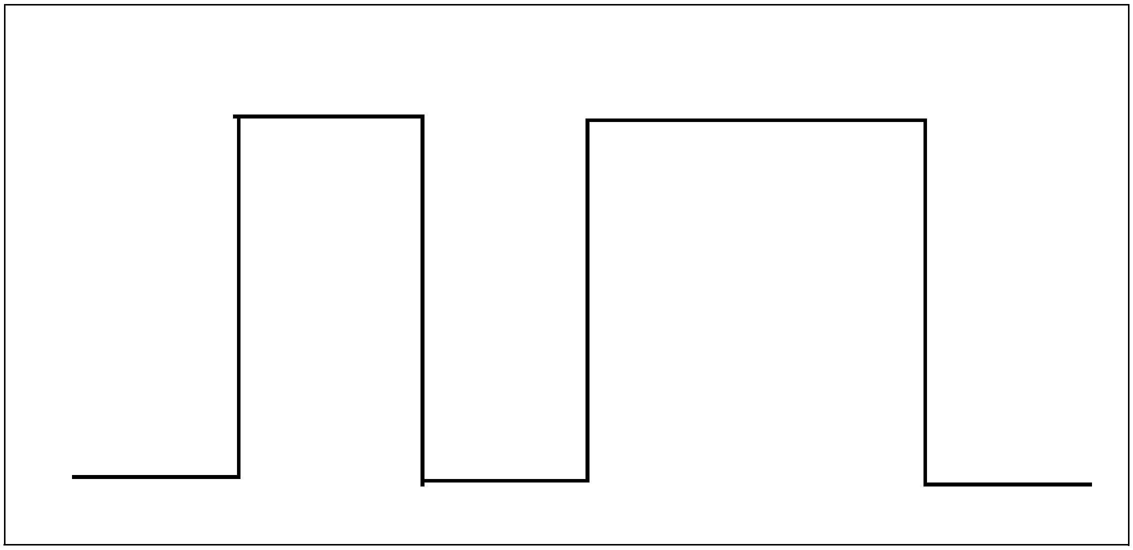


Figure 4-12 Illustration of the Hard Polish Model: Structure before Planarization

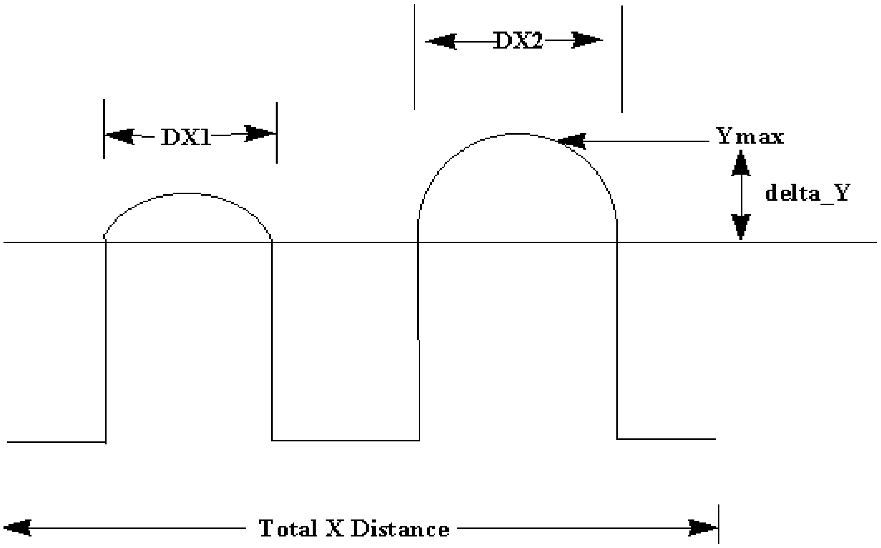


Figure 4-13 Illustration of Hard Polish Model: Structure after Planarization

A total amount of ΔY is always removed at each time step in the above fashion.

You can mix the hard polish model with the soft polish model and isotropic etch component by specifying the `ISOTROPIC` parameter of the `RATE.POLISH` statement.

4.6.2 Soft Polish Model

The soft polish model is based on the work of J.Warnock [102]. It has four parameters: `SOFT`, `LENGTH.FAC`, `HEIGHT.FAC`, and `KINETIC.FAC`. `SOFT` is the polish rate on a flat surface. `LENGTH.FAC` is the horizontal deformation scale in microns. It is a measure of the polishing pad's flexibility. It describes the distance at which shadowing will be felt by a "tall" feature. `HEIGHT.FAC` is the vertical deformation scale in (microns). This measures how much the polishing pad will deform with respect to the height of the feature. `KINETIC.FAC` increase the vertical polish rate as the surface becomes more vertical.

The following formula gives the polishing rate.

$$P_i = \frac{K_i A_i}{S_i} \tag{4-34}$$

K_i is the kinetic factor or horizontal component of the polish removal rate at point i. A_i is the accelerating factor of point i and is large for points that are higher and shadow other points. S_i is the shadow factor and decreases the polish rate as a function of the points that are above point i. For a flat surface, $K_i\ A_i/S_i = 1$. Following the work of Warnock, these three factors are calculated using the following set of equations.

The shadow factor is the one for flat surfaces. But it is generally calculated basedon one or two points that shadow point i and is given by the equations below.

$$S_i = exp\left(\frac{\overline{\Delta z_i}}{HEIGHT.FAC}\right) \tag{4-35}$$

$\overline{\Delta z_i}$ >= 0.0, so S_i > 1. $\overline{\Delta z_i}$ is obtained by integration over the surrounding topography.

$$\overline{\Delta z_i} = \sum_{i_{left}}^{i_{right}} z_i \big/ cosh\left(\frac{r_i}{LENGTH.FAC}\right) \tag{4-36}$$

In these equations, i_{left} and i_{right} refer to the two points that can possibly shadow point i. The effect of these shadow points depends on the two parameters `LENGTH.FAC` and `HEIGHT.FAC` as shown in the equations. The variable z_i is the vertical distance between the point i and the point i_{left}/i_{right}. The variable r_i is the horizontal distance between the point i and the point i_{left}/i_{right}.

The acceleration factor, A_i, is given by the equations below. A_i is calculated for the two points that shadow point i. In this manner, multiple shadowing effects are taken into account through the term A_{ileft}/A_{iright}. This is the acceleration factor for the point(s) that shadow. If point i shadows some other point j in the system, it will increase A_i by a similar factor. This increase is then passed on to A_{ileft}.

$$A_{ileft} = A_{ileft} + A_i \cdot B_i \cdot (1 - 1/S_i) \tag{4-37}$$

$$A_{iright} = A_{iright} + A_i \cdot B_i \cdot (1 - 1/S_i) \tag{4-38}$$

The constant B_{ileft}/B_{iright} is a weighing factor based on the amount of shadowing at point *i* due to shadow point i_{left}/i_{right}.

The kinetic factor is based on the following equation.

$$K_i \;=\; 1 + KINETIC.FAC \cdot \tan\alpha_i \qquad \text{4-39}$$

This shows the effect of the parameter, `KINETIC.FAC`, on the polishing rate. The angle α_i is the local angle that is tangent to the polished surface. The maximum allowable angle α_i is 89.9544° (1.57 radians) will avoid calculation errors.

Figure 4-14 demonstrates three regions where each of the components of the polishing rate would be large.

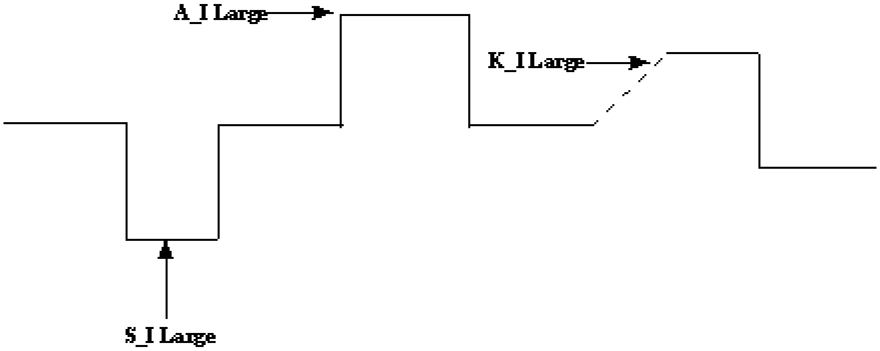


Figure 4-14 Soft Polishing Model: Areas where different components dominate

Chapter 5
Optolith Models

5.1 Overview

The Optolith module of Athena allows the use of sophisticated models for imaging, photoresist exposure, photoresist bake, and photoresist development. Optolith includes a library of photoresists with default characterizations for development and optical properties. These default characterizations can easily be tuned to adjust for variations that very typically occur from one facility to another. This chapter describes the models and capabilities of Optolith.

5.2 The Imaging Module

Optolith includes an imaging module that utilizes the Fourier series approach.

The theoretical resolution (*RES*) and Depth Of Focus (*DOF*) of a microlithographic exposure system are approximated by:

$$RES = k_1 \cdot \frac{\lambda}{NA} \tag{5-1}$$

and

$$DOF = k_2 \cdot \frac{\lambda}{(NA)^2} \tag{5-2}$$

where λ is the wavelength of the exposing radiation, NA is the Numerical Aperture of the imaging system, and k_1 and k_2 are process dependent constants. Typical values for k_1 are 0.5 for a research environment and 0.8 for a production process; the value usually assigned to k_2 is 0.5.

We shall discuss the basic assumptions upon which the model rests. Next, we shall derive the principal equations used for calculation of the image irradiance distribution for objects illuminated by partially coherent light.

The treatment presented here assumes the radiation incident on the object to be quasi-monochromatic, which means that the spectral bandwidth is sufficiently narrow so that wavelength-dependent effects in the optics or in diffraction angles are negligible. The source is of a finite spatial extent so that the advantages of spatial incoherence are realized in imaging.

The mask is completely general in that phase and transmission are variable, but it must be composed of rectangular features.

The calculation of the diffraction phenomena is based upon the scalar Kirchhoff diffraction theory. Since the dimensions of the mask are almost the same as the illumination wavelength, we can ignore any polarization taking place as the radiation propagates through the mask.

We assume scalar diffraction, which means neglecting the vector nature of the radiation. This is acceptable if all convergence angles are small.

According to Watrasiewicz [103], who experimentally investigated the limiting numerical aperture, the breakdown of the scalar theory occurs at angles of convergence greater than 30°, which corresponds to a numerical aperture of 0.5. Similar results were published by Richards and Wolf [104], who used theoretical calculations to investigate the electromagnetic field near the focus produced by an aplanatic system working at a high convergence angle. They also found appreciable departures from scalar theory for convergence angles larger than 30°. Since the convergence angles are calculated in air, we can assume that the accuracy of this model is even better inside the photoresist, where angles are reduced in accordance with Snell’s law.

Consequently, it can be stated that the scalar diffraction theory gives a reliable limit for imaging system numerical apertures of 0.5.

The approach used for calculating the image irradiance distribution is based on the work of Hopkins [105] and [106], which showed the partially coherent illumination of the object structure can be simulated by the incoherently illuminated exit pupil of the condenser. The exit pupil serves as an effective source, which produces the same degree of coherence in the illuminated object plane as the actual condenser system. The degree of coherence in the object plane is therefore determined by the shape and angular size of the effective source. The condenser system is assumed to be diffraction limited, that is, free of aberrations. Residual aberrations of the illuminator do have an appreciable influence on the final image for Koehler type illumination systems as shown by Tsujiuchi [107].

Figure 5-1 shows a schematic diagram of a generalized optical system. The actual source and the condenser system are replaced by the equivalent effective source having an irradiance distribution of $g\ (x_0, z_0)$. The effective source for the object plane U is taken to lie in the exit pupil reference sphere of the condenser lens. This means that directing from arbitrary points (x_0, z_0) on the effective source, plane waves propagate towards the object plane U having irradiance values of $\gamma\,(x_0, z_0)$.

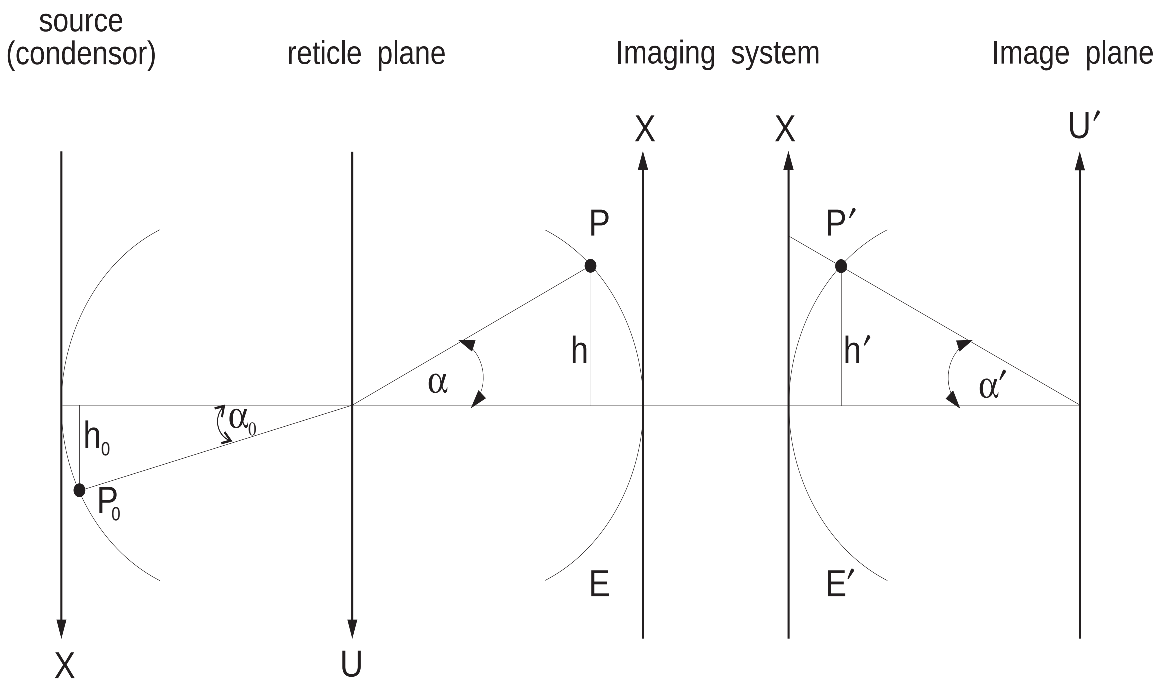


Figure 5-1 Schematic Diagram of a Generalized Optical System

The reduced coordinates [106] on the object plane are defined as follows:

$$u = \frac{2\pi}{\lambda} \cdot n \cdot \sin\alpha \cdot \xi \tag{5-3}$$

$$v = \frac{2\pi}{\lambda} \cdot n \cdot \sin\alpha \cdot \eta \tag{5-4}$$

where ξ and η are the Cartesian coordinates of the object plane, $2\pi/\lambda$ is the absolute value of the wave vector, and $n\text{-}sin\alpha$ is equal to the numerical aperture (NA) of the imaging system. Primed quantities indicate the corresponding coordinates and angles in the image space of the projection system. The fractional coordinates on the object pupil spheres are defined as follows:

$$x = \frac{\xi}{h} \tag{5-5}$$

$$z = \frac{\eta}{h} \tag{5-6}$$

where *h* is the radius of the pupil. The fractional coordinates of the exit pupil of the condenser are given by

$$x_0 = \frac{x}{\sigma} \tag{5-7}$$

$$z_0 = \frac{z}{\sigma} \tag{5-8}$$

In these equations:

$$\sigma = \frac{n_0 \cdot \sin\alpha_0}{n \cdot \sin\alpha} \tag{5-9}$$

where α_0 and α are angular semi-apertures of the condenser and the objective respectively. n_0 and n are the refractive indices in the image space of the illuminator and the object space of the imaging system, usually both are set to one. The ratio σ is the radius of the effective source referred to the aperture of the objective and governs the degree of spatial coherence in the object plane. The limits $\sigma \to 0$ and $\sigma \to \infty$ correspond respectively to coherent and incoherent illumination.

The object is taken to be infinitely thin. Therefore, a complex amplitude transmission function can describe the object, which gives the change in magnitude and phase produced on the radiation passing through it. The object has the complex transmission *A(u,v)*. Its real part is given by

$$\Re(A(u,v)) = \{ \begin{matrix} 1 \ \ in \ \ transparent \ \ areas \\ 0 \ \ in \ \ opaque \ \ areas \end{matrix} \tag{5-10}$$

The complex amplitude of the Fraunhofer diffraction pattern on the entrance pupil reference sphere at *E* of the imaging system is given, apart from a constant factor, by

$$a(x,z) = \frac{1}{2\pi}\int\int A(u,v)\cdot exp(-i(ux+vz))dudv \qquad 5\text{-}11$$

which is the inverse Fourier transform of the complex amplitude transmission of the object. If not stated otherwise, integration ranges from to – ∞ to + ∞.

If the object is illuminated by an element dx_0, dz_0 of the effective source at (x_0, z_0) with its amplitude proportional to $\sqrt{\gamma(x_0,z_0)}$, the object spectrum *a(x,z)* is then shifted by a corresponding amount. In this instance, the complex amplitude distribution on the entrance pupil sphere of the objective is

$$\sqrt{\gamma(x_0,z_0)}\cdot a(x-x_0,z-z_0) \qquad 5\text{-}12$$

The complex amplitude on the exit pupil reference sphere at *E'* will be given by

$$a'(x,z) = \sqrt{\gamma(x_0,z_0)}\cdot a(x-x_0,z-z_0)f(x,z) \qquad 5\text{-}13$$

In this equation, *f(x,z)* denotes the pupil function of the optical system. If the system has an annular aperture, where the central circular obstruction has the fractional radius ε, the pupil function has the form:

$$f(x,z) = \begin{cases} 0 & x^2+z^2<\varepsilon^2 \\ \tau(x,z)\cdot exp(i\cdot k\cdot W(x,z)) & x^2+z^2\le 1 \\ 1 & x^2+z^2\ge 1 \end{cases} \qquad 5\text{-}14$$

t(x,y) is the pupil transmission, which is usually set to one, and *W(x,z)* denotes the wave-front aberration. For an entirely circular aperture, ε becomes zero. Note that the approach taken here is somewhat similar to the one used in the investigations on phase contrast microscopy [108].

The function *W(x,z)* gives the optical path difference between the real wave-front and the exit pupil reference sphere. Commonly, the wave-front aberration is expanded into a power series [106], giving

$$W(x,z) = \sum_{l,m,n} W_{l,m,n}(\chi^2+\zeta^2)+(x\chi+z\zeta)+(x^2+z^2)^n \qquad 5\text{-}15$$

for a particular position *(x,z)* in the exit pupil. χ and ξ denote the fractional coordinates of the image field. The values of l, m, and n describe the order of aberrations, while the coefficients *W(l,m,n)* determine the magnitude of the aberrations.

The aberration coefficients up to the ninth order of aberration are specified in the `ABERRATION` statement.

For third order aberrations `l`, `m`, and `n` take the values:

```
l=0 , m=0 , n=2 : spherical aberration
l=0 , m=1 , n=1 : coma
l=0 , m=2 , n=0 : astigmatism
l=1 , m=0 , n=1 : field curvature
l=1 , m=1 , n=0 : distortion
l=0 , m=0 , n=1 : defocus
```

where isoplanatism is assumed for the particular section of the image field for which the irradiance distribution is calculated. The coefficient W_{001} can be determined from

$$W_{001} = \delta \cdot \frac{(n' \cdot \sin\alpha')^2}{2 \cdot \lambda} \qquad 5\text{-}16$$

where δ refers to the distance of the defocused image plane to Gaussian image plane.

The resulting amplitude in the image plane due to a wave coming from the point x_0, z_0 of the effective source is

$$A'(x_0, z_0; u', v') = \frac{1}{2\pi} \cdot \sqrt{\gamma(x_0, z_0)} \cdot \int a'(x - x_0, z - z_0) \cdot exp(i(u'x + v'z))dxdz \qquad 5\text{-}17$$

where (u',v') refers to a point in the image plane. The irradiance distribution associated with the illuminating wave of the effective source will then be represented by

$$dI'(x_0, z_0, ;u', v') = |A'(x_0, z_0, ;u', v')|^2 dx_0 dz_0 \qquad 5\text{-}18$$

Since, by definition, the effective source is equivalent to a self-luminous source, the total irradiance at (u',v') can be obtained by integrating over the entire source Σ.

$$I'(u', v') = \int_{\Sigma} \int |A'(x_0, z_0; u', v')|^2 dx_0 dz_0 \qquad 5\text{-}19$$

where Σ indicates the area of the effective source for which $\gamma(x_0,z_0)$ has non-zero values.

For this purpose, Equation 5-19 is put into the form:

$$I'(u', v') = \iint_{\Sigma} \gamma(x_0, z_0) \cdot |\Phi(x_0, z_0; u', v')|^2 dx_0 dz_0 \qquad 5\text{-}20$$

where:

$$\Phi(x_0, y_0; u', v') = \frac{1}{2\pi} \cdot \iint a'(x - x_0, y - y_0) \cdot f(x, y) \cdot exp(i(u'x + v'y))dxdy \qquad 5\text{-}21$$

$\Phi(x_0, z_0 \; ; \; u', v')$ is proportional to the intensity at the point (u',v') due to a wave of unit irradiance passing through (x_0, z_0) of the effective source.

In the case of an annular shaped source, x_0, z_0 has the form:

$$\gamma(x_0, z_0) = \begin{cases} 0 \ for \ x_0^2 + z_0^2 < \varepsilon_0^2 \\ 1 \ for \ x_0^2 + z_0^2 \le 1 \\ 0 \ for \ x_0^2 + z_0^2 > 1 \end{cases} \qquad 5\text{-}22$$

where ε_0 is the fractional radius of the centered circular obstruction in the exit pupil of the condenser lens. For a circular exit pupil, ε_0 becomes zero.

Equation 5-20 is the principle relation of a generalized Abbe theory, where the image formation under partially coherent illumination of the object is accounted for by a combination of coherent imaging processes for perpendicular and obliquely incident illuminating plane waves on the object. Since only the image irradiance is of interest, it can be determined without using of coherence theory [106]. For the computation, the whole source is divided into a number of luminous point sources considering the imaging due to each source as an independent coherent image formation process. The contributions from each point source do not interfere, so the net image irradiance is the sum of the irradiance from each source point.

The normalization used throughout this investigation is that the mask is illuminated with unit irradiance so that the ideal image has unit irradiance, where unit magnification is assumed. Therefore, the brightness of the source decreases as its size increases. Equation 5-20 is the principle equation of the algorithm, which is used for studying the influence of annular apertures.

The object spectrum (see Equation 5-11) is calculated analytically, and the coherent image (see Equation 5-18) is calculated using a Fourier Series approach.

The shape of a single mask feature must be rectangular. This is because the Fourier transform for a rectangular feature is calculated based on an analytical formula.

Since the Fourier transform is linear, you can compose arbitrary shaped mask features from the rectangular components. The object spectra of the single mask features (components) are simply added up. The treatment can then be considered as appropriate, and no numerical discretization errors in the size and placement of the mask features can occur.

Note: You may use MaskViews to create or import masks of any arbitrary shape. The mask layout will be sliced (divided) on rectangular elements when it is imported into Optolith. Optolith can import masks containing any number of mask elements.

5.3 Optical System

Figure 5-2 shows the optical system used by Optolith. The meshes in the Fourier and Image planes are totally independent. There is no mesh in the object or reticle plane.

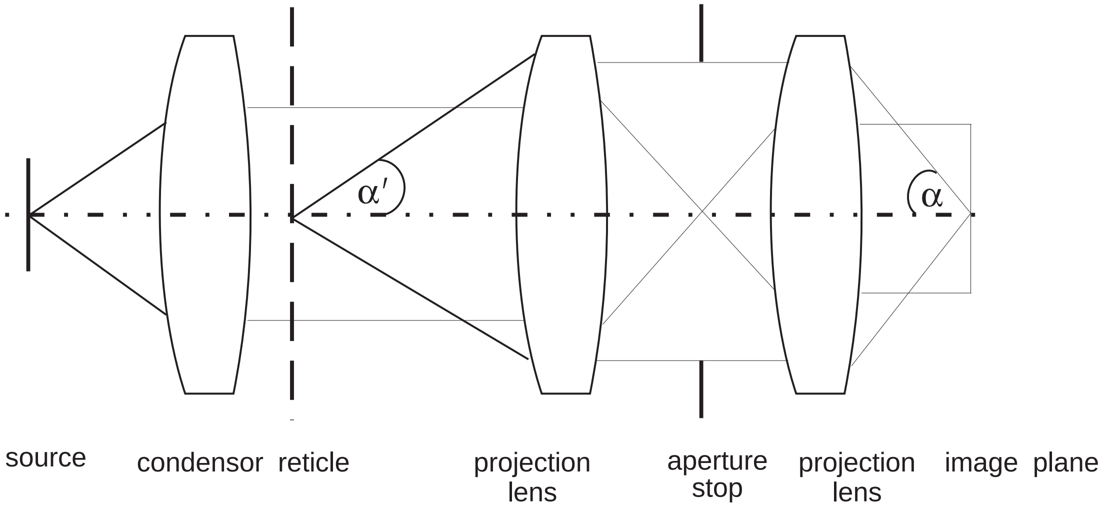


Figure 5-2 The Generated Optical System

5.3.1 Discretization Errors

The size of the window in the reticle plane is determined by the number of mesh points in the projector pupil, the numerical aperture, and by the chosen wavelength:

$$CW = NP \cdot lambda / NA \tag{5-23}$$

where:

- *CW* is a computational or sampling window (mask or image cell) in the object or reticle plane.
- *NP* is the number of mesh points in the projector pupil.
- *NA* is the numerical aperture of the stepper.
- *lambda* is the chosen wavelength.

For an i-line stepper with NA = 0.54, the size of the sampling window is the square whose side length is equal to 6.8 µm (10 · 0.365/0.54). No mask feature should exceed this dimension.

You can increase the size of the sampling window for this particular stepper to any size by increasing the number of mesh points in the projector pupil. This will be done automatically to accommodate the mask and image windows that were specified.

Mask features cannot be placed outside of the sampling window. As mentioned earlier, the image mesh is totally independent of the mesh in the Fourier plane. This allows you to arbitrarily specify the number and distance of image points.

5.3.2 Mesh

The size of the computational window is determined by Equation 5-23) and the position of the mask points. By positioning this window so that the mask cell in the object plane is covered, multiple image cells can be calculated.

5.3.3 Computation Time

To increase computation, first use a very coarse mesh for screening-type simulations and then refine the mesh as you approach specific points of interest. Computation time is linearly dependent on the number of source points, which is determined by the coherence factor.

5.4 The Exposure Module

The Exposure Module computes the intensity distribution in the photoresist through the numerical solution of the Helmholtz equation (Equation 5-24).

$$\nabla^2 E + k^2 n^2 E = 0 \qquad 5\text{-}24$$

Here, *E* is the electric field, *n(x,y)* is the complex refractive index of media, and *k* is the wave number. The specific solution of this general equation is determined by a set of boundary conditions. Generally, the target (substrate coated with the resist film) consists of an arbitrary number of different materials with *n* in Equation 5-24 and interfaces between material regions. Also, material refractive index can depend on the absorbed dose (i.e., *n* can vary with the exposure time).

According to the electromagnetic theory, any field distribution can be represented by unique set of plane waves. Generally, such a set has an infinite number of terms. But, the main contributions are the incident wave and reflections from interfaces. Therefore, the calculation of the field inside the complex structure can be divided into several subsequent calculations of the plane wave propagation through the structure. The final field distribution is the sum of distributions obtained for all separate plane waves. The simulation algorithm is outlined as below.

1. Simulation of the incident field propagation through the target (resist + substrate)
2. For each segment of each interface:
   - compute the direction of the reflection,
   - align the coordinate system and simulation domain with the direction of the reflection,
   - obtain the field distribution over the segment,
   - recursion of the propagation simulation procedure for the field generated over the surface segment.
3. Summation of the distributions obtained with all recursive steps.

The reflection from each segment of interface is computed, using the preliminary obtained field on the segment as the initial (incident) field. The recursion depth can be specified as a simulation parameter.

The Beam Propagation Method (BPM) [109] is used to simulate the field propagation. The BPM can be used for different types of radiation (e.g., UV, EUV, X-ray) as well as for multi exposure processes and multilayer and non-linear resists.

There are three reason why we choose this method. The first reason is because the diffraction of the field along the propagation is automatically taken into account. The second reason is because it includes a capability to simulate non-linear effect of the intensity distribution on the local optical properties of the resist material. The third reason is because it provides a good “accuracy to run time” ratio.

The BPM is used to solve the Helmholtz equation for electromagnetic field inside the structure. During the simulation, the field distribution is formed as the superposition of incident light with all the reflections from all elements of the resist-substrate interface and secondary reflection(s) from the upper resist surface. [109] shows the formal descriptions of the BPM.Papers [110], [111], and [112] also describe some applications using BPM.

In this model, the Helmholtz equation (Equation 5-24) for the electric field, *E*, in the media with complex refractive index, *n(x,y)*, is solved in two main stages. The first stage, the diffraction over a small spatial step along the propagation is calculated. Thus, obtaining the new field amplitude distribution without absorption taken into account. Then, the actual field distribution is computed as a product of this amplitude distribution and the distribution of the complex absorption over the step. Let the wave propagate along the y-axis.

We find the solution as a quasi-plane wave *E=A(x,y)exp(inky)* with a slowly varied amplitude *A*. *A* is then modified with *y*, which is slower than phase term *inky*. In this case, the Fourier image of current distribution *A* in the plane $y=y_0$ is defined as follows.

$$F_A(k_x) = \int A(x, y_0)\,exp(-ik_x x)\,dx \tag{5-25}$$

After propagating over a small step, Δ*y*, each component of F_A obtains additional phase shift corresponding to the value of $k_y = \sqrt{k^2 - k_x^2}$. Thus, the amplitude distribution at new *y* (without accounting for absorption) can be written as follows.

$$A(x, y_0 + \Delta y) = \int F_A(k_x)\,exp(ik_y \Delta y)\,exp(ik_x x)\,dk_x \tag{5-26}$$

Since actual material optical properties differ from the properties of vacuum, the field at the new plane is computed simply as follows.

$$E(x, y_0 + \Delta y) = A(x, y)\,exp\{ik[n(x, x) - 1]\Delta y\} \tag{5-27}$$

The algorithm is repeated recursively step-by-step over all simulation domains. The same calculations are then applied to reflections from all interface segments. The current intensity distribution is calculated from the field distribution as

$$I(x, y) = |E(x, y)|^2 \tag{5-28}$$

During the exposure, the resist structure is modified. So, the dissolution inhibitor is converted to the photo reaction product. The initial normalized concentration of photoactive compound (PAC) is defined by local intensity magnitude as:

$$M_{PAC} = exp[-CI(x, y)] \tag{5-29}$$

where *I(x,y,)* is the current intensity distribution, and *C* is Dill’s *C*-parameter. Accordingly, the optical properties of the resist (complex refraction index *n*, which includes both refractive and absorption indices) are modified too. The capability to take into account is the effect of dose (intensity) on the refraction index *n*, which is implemented into the module and the following approach is realized. Non-linear dependence of *n* on dose is defined through the PAC concentration as

$$= n_{unexposed} + \Delta n[1 - M_{PAC}(x, y)] \tag{5-30}$$

Here, $n_{unexposed}$ is the complex refraction index of the unexposed resist. $\Delta n = n_{exposed} - n_{unexposed}$ is the difference between values of n for completely exposed and unexposed resists.

Current intensity distribution is calculated after each simulation of direct propagation and all the reflections from interfaces with BPM. Then, current M_{PAC} and *n(x,y)* are calculated using (Equation 5-29) and (Equation 5-30) respectively for each point of the resist volume. The new values of *n(x,y)* are used during next recursion of the field and intensity simulations. Thus, the resulting intensity distribution is obtained as an accumulation of intermediate results.

You can specify the optical properties of the simulated material in the `OPTICAL` statement. You can specify the refraction and absorption indices for unexposed resist (and/or for any other material) using the `REFRAC.REAL` and `REFRAC.IMAG` parameters respectively. To specify the difference of the refraction index for the completely exposed resist from the unexposed one, use the `DELTA.REAL` and `DELTA.IMAG` parameters. If this difference isn't specified, the effect of intensity on the resist refraction index will not be taken into account during the simulation. The number of recursions to obtain the intensity distribution in both cases with and without taking into account dose-to-*n* effect is specified in the `EXPOSE` statement as `NUM.REFL`. The maximum dose that corresponds to completely exposed resist is specified with `DOSE` parameter.

The examples below show how to use the `OPTICAL` and `EXPOSE` statement to specify parameters for the exposure module.

Example1: Resist exposure with accounting dose effect

```
OPTICAL    NAME.RESIST=CURREN_RESIST    I.LINE    REFRAC.REAL=1.4
REFRAC.IMAG=0.02 DELTA.REAL=-0.2 DELTA.IMAG=0.01

EXPOSE DOSE=200 NUM.REFL=5
```

Example2: Resist exposure without accounting dose effect

```
OPTICAL    NAME.RESIST=CURREN_RESIST    I.LINE    REFRAC.REAL=1.4
REFRAC.IMAG=0.02

EXPOSE DOSE=200 NUM.REFL=5
```

5.5 Photoresist Bake Module

Post Exposure Baking (PEB) of the photoresist has been demonstrated to dramatically reduce standing wave fringes of the developed resist image resulting from optical interference of monochromatic illumination. This effect is generally accepted to be a result of bulk diffusion of the PAC and photo reaction products.

The simple physical model, which is adopted here to describe the PEB, is that just one chemical constituent of the resist diffuses. This constituent is generally assumed to be PAC or the dissolution inhibitor, which diffuses according to the diffusion equation with the diffusion constant *D* being independent of time, concentration and location.

$$\frac{dM}{dt} = \nabla \cdot (D \nabla M) \tag{5-31}$$

M is the PAC concentration and *t* is the PEB time. For a more general discussion, see [113]. *M* is calculated by solving the two dimensional diffusion (Equation 5-31).

The diffusion length can be related to the bake time t and the diffusion coefficient D.

$$2tD = \sigma^2 \tag{5-32}$$

For a PEB of 60 seconds at 125ºC, a diffusion length in the range of $0.04 < \sigma < 0.06$ microns would be appropriate.

PEB can also be specified with parameters temperature and time. The diffusivity *D* is given by the equation:

$$D = D_o exp(-D_E \diagup kT) \tag{5-33}$$

You can specify the D_0 and D_E parameters in the `RATE.DEVELOP` command.

“Reflective” boundary conditions at the air/resist interface and at the resist/substrate interface must be incorporated to ensure that the total amount of dissolution inhibitor in the resist is conserved. *M(x,y,t)* is extended into regions outside the resist by reflection at the planar interfaces.

A post development bake is also available. It models a physically based reflow of the photoresist.

5.6 The Development Module

The development model is based on the knowledge of the PAC distribution or dissolution inhibitor in the resist layer after exposure and post-exposure bake. In classical Novolac resists, the dissolution inhibitor and the PAC are usually part of the same molecule. In chemically amplified resists, the reaction kinetics are more complicated. The inhibitor concentration still, however, is considered to be the key quantity for the development process.

In positive tone Novolac resists, to determine the inhibitor concentration from exposure simulations, use Dill’s model as previously described. This model applies when the resist material undergoes a transition between two chemical states during the exposure step. The actual development process is treated as a surface limited etching process, which is dependent on the particular resist-developer chemistry and on the local concentration of the dissolution inhibitor at the surface of the resist that has been decomposed to a degree during the exposure step. If the resist developer chemistry is held constant, the dissolution rate is assumed to be a function of the inhibitor concentration only.

The rate function *r(x,y)* is determined experimentally and usually fitted by an empirical function to experimental development rate data as a function of the remaining PAC concentration *M(x,y).*

You can use one of the following models to simulate the development process for the specific resist-developer combination.

- Dill
- Kim
- Mack
- Trefonas
- Hirai

Each model assumes a specific rate function type to describe the rate-inhibitor concentration relation. These models are described in the following sections.

5.6.1 Dill’s Development Model

The Dill model [114] uses the parameters E_1, E_2, and E_3. Surface induction effects are not considered. The bulk development is given by:

$$R(x, y) = exp(E_1 + E_2 \cdot M(x, y) + E_3 \cdot M(x, y)^2) \quad 5\text{-}34$$

and for $M(x,y) \leq 0.4$:

$$R(x, y) = exp(E_1 + E_2 \cdot 0.4 + E_3 \cdot 0.16) \quad 5\text{-}35$$

5.6.2 Kim's Development Model

The Kim model [115] describes the development rate through the function

$$R_{Induction} = 1-(1-(R_5-(R_5-R_6)M(x,y,z)))\cdot exp-\frac{d(y)}{R_4} \tag{5-36}$$

$$R_{Bulk}(x,y) = \frac{1}{\frac{1-M'(x,y)}{R_1}+\frac{M'(x,y)}{R_2}} \tag{5-37}$$

$$M'(x,y) = M(x,y)\cdot exp(-R_3(1-M(x,y))) \tag{5-38}$$

$$R(x,y) = R_{Induction}\cdot R_{Bulk} \tag{5-39}$$

$$R(x,y) = R_{Induction}\cdot R_{Bulk} \tag{5-40}$$

where R_{Bulk} is the bulk development rate and $R_{Induction}$ is the surface induction factor. The limiting development rate values are R_1 and R_2 respectively for completely exposed and unexposed resist. The function $R_{Induction}(x,y)$ is an empirical relationship describing the reduced dissolution rate at the surface of a resist layer and is a function of the normal distance from the original surface of the resist $d(y)$ and the amount of remaining PAC, $M(x,y)$. The parameter R_4 is the characteristic length along this path for the induction effect. The parameters R_5 and R_6 are respectively the ratio of the surface rate to the bulk rate for a completely exposed resist, and the ratio of surface rate to bulk rate for an unexposed resist.

5.6.3 Mack's Development Model

The Mack model [116] describes the development rate through the function

$$R(x,y) = R_{max}\cdot\frac{(a+1)(1-M(x,y))^n}{a+(1-M(x,y))^n}+R_{min} \tag{5-41}$$

$$a = \frac{n+1}{n-1}\cdot(1-M_{th})^n \tag{5-42}$$

where the parameter n is a selectivity parameter describing the sensitivity of the developer to the exposed photoresist. The M_{th} parameter is the threshold PAC concentration. The R_{max} parameter is the development rate of a completely exposed resist. The parameter R_{min} is the development rate of totally unexposed resist.

5.6.4 Trefonas' Development Model

The Trefonas development rate model [117] requires only two parameters.

$$R(x,y) = R_0\cdot(1-M(x,y))^q \tag{5-43}$$

where R_0 is the development rate for unexposed photoresist and q is sensitivity.

5.6.5 Hirai's Development Model

The development rate model by Hirai [118] is very similar to the one by Trefonas. The rate function of the Hirai model is given by

$$R(x,y) = R_0 \cdot (1 - M(x,y))^{\alpha} + R_C \tag{5-44}$$

where R_0 is the development rate for fully exposed photoresist, R_C is the rate for unexposed resist material and α is a reaction constant.

5.7 Proximity Printing

Alongside with the “standard” projection imaging simulation, Optolith includes an additional module for simulation of proximity printing. The proximity 1×1 printing (i.e., the imaging without any reduction lens) is used to print relatively big features of a micron scale. This method is still practical and in some cases could be cost effective. Internally, projection and proximity printing modules use different simulation techniques. A common syntax and user interface, however, allow you to switch easily from one module to another. Some specifics of the proximity lithography and a brief theory used in the proximity printing module are described below.

5.7.1 General Description of Proximity Lithography

Proximity lithography is used to print images of masks without expensive projection systems. It has internal resolution limits but still applicable for relatively large objects. Figure 5-3 shows the optical scheme of proximity lithography.

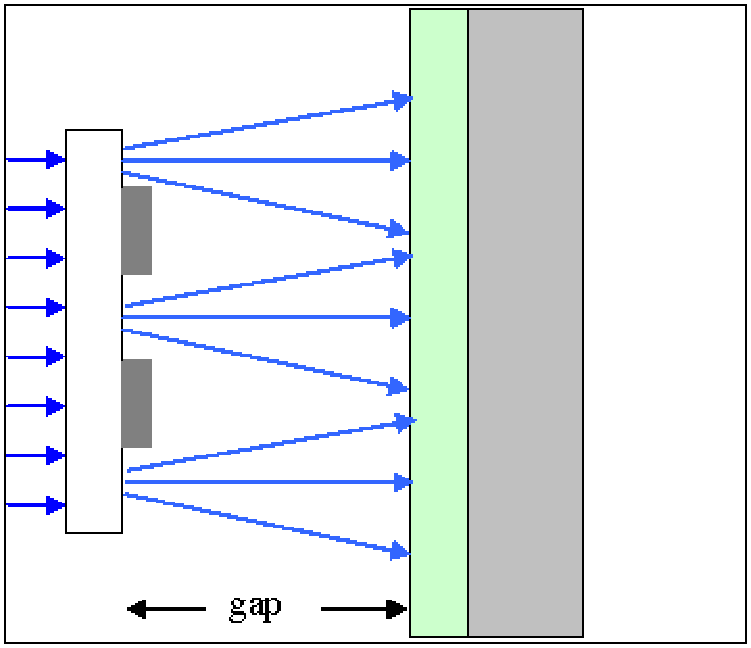


Figure 5-3 Scheme of proximity optical system.

Light illuminating the mask creates a diffraction image in resist film, placed on some distance (gap) from the mask. Due to diffraction effects in the gap, the image in the resist film is a distorted mask image. Distortion depends on the gap size and on the radiation wavelength and size of printed features. The reliability and longevity of the mask also depends on the gap. Therefore, identification of the optimal printing conditions is a very important technology design task.

5.7.2 Theory of Proximity Printing

A formal description of diffraction on a gap can be found in any classical book on physical optics. The Fresnel diffraction is applicable here. Usually, the Fresnel diffraction integral is obtained by applying Huygens diffraction approximation. It can be derived from the “first principles” by applying specific limitation to common wave equation. The goal is to obtain distribution of the light intensity over the resist surface. The real resist exposure with the light absorption into resist, reflections from materials interfaces, and following photo-chemical resist modifications are considered the same way as for the projection printing.

The common wave equation is

$$\nabla^2 E + k^2 E = \frac{\partial E}{\partial x^2} + \frac{\partial^2 E}{\partial y^2} + \frac{\partial^2 E}{\partial z^2} + k^2 E = 0 \qquad 5\text{-}45$$

where E is the electromagnetic field and k is the wave-vector.

Because the normal incidence on the mask is considered and propagation of light from the mask to resist surface is to be calculated, it is convenient to represent

$$E(r) = A(r)\,exp(ikz) \qquad 5\text{-}46$$

where A is amplitude of electromagnetic field and z is direction along the light propagation.

Substitution from Equation 5-46 into Equation 5-45 gives

$$\frac{\partial^2 A}{\partial x^2} + \frac{\partial^2 A}{\partial y^2} + \frac{\partial^2 A}{\partial y^2} + 2ik\frac{\partial A}{\partial z} = 0 \qquad 5\text{-}47$$

Obviously, A is modified with z weakly. Therefore, you can neglect the third term in Equation 5-47. As the result, Equation 5-47 can be rewritten as

$$\frac{\partial^2 A}{\partial x^2} + \frac{\partial^2 A}{\partial y^2} + 2ik\frac{\partial A}{\partial z} = 0 \qquad 5\text{-}48$$

Now Equation 5-48 looks as a diffusion equation with complex diffusion coefficient, where z replaces diffusion time. It means that the propagation along the z-direction can be formally considered as a “diffusion” of complex amplitude A in the x,y-plane with a complex “diffusion coefficient”.

The known δ-solution of traditional diffusion equation

$$\frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2} - \frac{1}{\Upsilon^2} + 2ik\frac{\partial f}{\partial t} = 0 \qquad 5\text{-}49$$

is as follows

$$f_0 = \frac{1}{t}exp\left(-\frac{x^2 + y^2}{4\Upsilon^2 t}\right) \qquad 5\text{-}50$$

The substitution in Equation 5-48 results in:

$$A_0 = \frac{1}{z}exp\left(-ik\frac{x^2 + y^2}{2z}\right) \qquad 5\text{-}51$$

The solution for given "initial conditions" (i.e., amplitude *A* immediately after (thin) mask should be equal to the mask transparency *T(x,y)*) can now be expressed as a convolution of A_o with *T*:

$$A(x,y,z)\frac{1}{z}\int T(X_m, Y_m)\,exp\left(-ik\frac{(x-X_m)^2+(y-Y_m)^2}{2z}\right)dX_m dY_m \qquad \text{5-52}$$

This expression exactly corresponds to the Fresnel diffraction.

5.7.3 Simulation Method

The Beam Propagation Method (BPM) is very efficient for tasks where a "one-way" propagation of electromagnetic field is considered. Common description of BPM can be found in [119], [120]. A simplified approach applicable to proximity printing conditions (i.e., a gap space filled with homogeneous matter, so refractive index does not depend on coordinates in the gap) is described here.

The initial distribution of amplitude *A* in *x* and *y* (in the plane perpendicular to the propagation direction *z*) can be obtained easily as a complex mask transparency function:

$$A_0 = A(x,y,z=0) \qquad \text{5-53}$$

Then by applying the 2D Fourier transform over *x* and *y*, you can obtain the "angular" distribution:

$$A(k_x,k_y,k_z=0) = \int A(x,y,z=0)\,exp[i(k_x x + k_y y)]dxdy \qquad \text{5-54}$$

In this angular-spectrum domain, the relation between *A* and its z-derivative from Equation 5-48 is given by the following expression [121]:

$$\frac{\partial A}{\partial z}(k_x,k_y,z) = i\sqrt{k^2-k_x^2-k_y^2}A(k_x,k_y,z) \qquad \text{5-55}$$

It means that for each particular "direction", a specific phase shift has to be applied to get the value of *A* at a new z-position:

$$A(k_x,k_y,z+\Delta z) = A(k_x,k_y,z)\,exp(ik_z\Delta z) \qquad \text{5-56}$$

In this equation, the z-component of the wave vector is derived by:

$$k_z = \sqrt{k^2-k_x^2-k_y^2} \qquad \text{5-57}$$

Now the inverse Fourier transform will produce the distribution of the field amplitude over x, y plane at *z*=Δ*z* (i.e., at the gap distance from the mask).

Finally, the required intensity distribution is the square of the module of the field amplitude *A*.

Chapter 6
Statements

6.1 Overview

Athena executes a file that describes the process, meshing, and models to be used in a simulation. The contents of the file are statements, each of which prompts an action or sets a characteristic of the simulation. This chapter is a reference to the command language that can be used to control Athena.

Throughout this manual, we will refer to commands, statements, and parameters. A line in an input file is referred to as a statement (or statement line).

An Athena statement is specified in the general format

```
<COMMAND>   <PARAMETERS>=<VALUE>
```

where <COMMAND> is the command name, <PARAMETER> is the parameter name, and <VALUE> is the parameter value. Four types of parameters are used in Athena — Real, Integer, Logical, and Character. The space character is used to separate parameters from a command or from other parameters.

Table 6-1 Types of Parameters

Type	Description	Value Required	Example
Character	An alphabetic, alphanumeric, or numeric string	Yes	OUTFILE=MOS.STR
Integer	Any whole number	Yes	DIVISIONS=10
Boolean	A true or false condition	No	OXIDE or OXIDE=f
Real	Any real number	Yes	C.BORON=1.5e14

Any parameter that does not have a logical value must be specified in the form PARAM=VAL, where PARAM is the name of the parameter and VAL is the value of the parameter. Boolean parameters must be separated from other parameters or commands with a space.

For example, in the statement line:

```
DEPOSIT NITRIDE THICK=0.35
```

the NITRIDE parameter has a Boolean value (true) and the THICK parameter has a value of 0.35(real).

Many parameters are provided default values. If a parameter is not specified, its default value will be used. Table 6-1 explains the different types of parameters which may be used when preparing an Athena input deck. The command language of Athena is not case sensitive and can be entered using either upper case or lower case letters.

6.1.1 Abbreviations

It is not always necessary to input the entire statement or parameter name. Athena only requires that you input enough letters to distinguish that command or parameter from other commands or parameters. For example, DEPO can be used to indicate the DEPOSIT command.

6.1.2 Continuation Lines

Since it may be necessary for a statement line to contain more than 256 characters, Athena allows you to specify continuation lines. If a statement line ends with a backslash (\), the next line will be interpreted as a continuation of the previous line.

6.1.3 Comments

Comments are indicated by the `COMMENT` statement or a number sign (#). All characters on a line which follow a comment indicator (`COMMENT` or `#`) will not be processed by Athena. The comment symbol $ is not supported anymore. The `$` should be avoided for use as a character in strings since it is used as part of shell capabilities included in DeckBuild.

6.1.4 General Syntax Description

An Athena statement is a sequence of words starting with a statement name and followed by some or all of the statement's parameters. This manual describes the syntax for each statement in the following way:

```
STATEMENT NAME

DESCRIPTION OF PARAMETER 1

DESCRIPTION OF PARAMETER 2 . . .
```

Parameters are described in the following form:

```
PARAM=<n>... a real valued parameter

PARAM=<c>... a string valued parameter

PARAM ... a Boolean parameter
```

Boolean parameters are those that recognize the Boolean values `TRUE` and `FALSE` as valid values. In Athena, Boolean parameter values are automatically set to true if the name of the Boolean parameter appears by itself in a statement. A Boolean parameter can be set to false using the syntax: `PARAM=FALSE` or `PARAM=F`.

A mutually exclusive choice among parameters is indicated by parentheses around the parameters and vertical bars between each parameter `(PAR1|PAR2)`. Only one parameter in such a group may be specified at a time. Specifying more than one parameter in a mutually exclusive group is an invalid operation and will generally prompt a warning or error message.

Parameters that are optional to a statement are enclosed by brackets [ ]. Most parameters are assigned default values and so defining them is optional. All parameters and parameter values, however, should be checked in the context of the actual process that will be simulated before relying on the results of any simulation.

String-valued parameters can be specified as a single word, e.g., `INFILE=FILE1`, or as a sequence of words surrounded by double quotes, e.g., `MATERIAL="Nickel Silicide"`. Real-valued parameters can be specified as expressions involving numbers, numerical constants, the operators `+`, `-`, `*`, `/`, and the functions listed in Table 6-2. If an expression contains spaces, then enclose it in parentheses.

Table 6-2 Functions

Function	Description
abs5	Absolute value
active	Active portion of the specified dopant
erf	Error function
erfc	Complimentary error function
exp	Exponential
gradx	Computes the approximate gradient in the x direction
grady	Computes the approximate gradient in the y direction
log	Logarithm
log10	Logarithm base 10
mat1@mat2	Returns the y value of the interface between mat1 and mat2 along a vertical slice at the given location
mat1|mat2	Returns the x value of the interface between mat1 and mat2 along a horizontal slice at the given location
scales	Scales the value given by the maximum value
sqrt	Square root
xfn	Takes y and z and finds a matching x
yfn	Takes x and z and finds a matching y
zfn	Takes x and y and finds a matching z

Examples:

```
PAR1=<n>
```

PAR1 is a required numeric valued option.

```
PAR1=( 4.0 * EXP( -2.0 / (8.62E-5 * 1173.0) ) )
```

PAR1 is a required numeric valued option, assigned a real number expression.

```
[PAR2=<c>]
```

PAR2 is an optional character variable.

For further examples of expressions, see SET and EXTRACT in DeckBuild User’s Manual.

6.1.5 Command Line Parsing

Athena supports expressions on the command line. For example:

```
DIFFUSE TIME=10/60 TEMP=1000
```

Be careful when using parentheses as the precedence of arithmetic operators, as in programming languages, is not guaranteed in all cases.

6.2 Athena Statements List

This chapter contains a complete description (in alphabetical order) of every statement used by any of the Athena products. The following documentation is provided for each statement:

- The statement name
- A list of all of the parameters of the statement and their type
- A description of each parameter or group of similar parameters
- An example of the correct usage of each statement

The Athena command language encompassed by this document describes each of the modules of Athena, namely: Elite, Optolith, SSuprem4 and their submodules. Depending on which of the Athena modules have been purchased, some of the capabilities described may not be available as part of the Athena installation.

Note: You can print a summary of statement names and parameters by using the `HELP` statement.

The following list classify Athena statements and provide their brief description and use.

6.2.1 Structure and Grid Initialization Statements

These statements define the dimensions, boundary conditions, grid density, and material type of the initial structure. Typically, only `LINE` and `INITIALIZE` statements are required.

- **BASE.MESH** specifies parameters of the base mesh used for initial grid generation.
- **BOUNDARY** specifies which lines in a rectangular grid are exposed to gas.
- **INITIALIZE** sets up the initial grid and specifies background doping concentrations and material type.
- **LINE** specifies the positioning of x and y grid lines for a rectangular mesh.
- **REGION** specifies corresponding sections of the rectangular mesh and material.

6.2.2 Structure and Mesh Manipulation Statements

These statements manipulate the geometry or attributes of the structure or create output files.

- **ADAPT.MESH** enables the adaptive meshing algorithm.
- **ADAPT**.**PAR** specifies adaptive meshing parameters.
- **BASE**.**PAR** defines adjacent mesh characteristics of an automated base mesh.
- **ELECTRODE** names electrode regions.
- **GRID**.**MODEL** defines a template file containing adaptive meshing commands.
- **PROFILE** causes Athena to read in an ASCII file of depth and doping data.
- **RELAX** loosens the grid within a user-specified area.
- **STRETCH** allows changes in structure geometry by stretching at a horizontal or vertical line.
- **STRUCTURE** writes the mesh and solution information into a file. This is the main output statement for generating program data to be plotted.

6.2.3 Simulation Statements

These statements apply physically based models for processing operations to the structure.

- **BAKE** performs post-exposure or post-development photoresist bake.
- **DEPOSIT** deposits a material layer.
- **DEVELOP** performs photoresist development.
- **DIFFUSE** performs a time/temperature step on the wafer and calculates oxidation and diffusion of impurities.
- **EPITAXY** performs high temperature silicon epitaxial growth.
- **ETCH** performs a geometric or machine type etch on the structure.
- **EXPOSE** models photoresist exposure.
- **IMAGE** calculates a 2D or 1D aerial image.
- **IMPLANT** models ion implantation.
- **POLISH** simulates chemical mechanical polishing in the Elite module.
- **STRESS** computes the thermal elastic stresses.
- **STRIP** removes photoresist or another user specified material.

6.2.4 Model Statements

These statements are used to change model parameters and coefficients. The parameters are described in the statement descriptions. When starting-up, Athena executes the model statements in the file named `athenamod` located in the `$SILVACO/lib/athena` subdirectory corresponding to the version number and system type of Athena that you are running. This file contains the default parameters for most model statements.

- **ABERRATION** defines aberration parameters of the optical projection system.
- **CLUSTER** specifies parameters of {311} cluster model.
- **ILLUMINATION** describes the photolithographic illuminating system.
- **ILLUM**.**FILTER** defines filters used in the illumination source for photolithography.
- **IMPURITY** sets the coefficients of impurity kinetics.
- **INTERSTITIAL** sets the coefficients of interstitial kinetics.
- **LAYOUT** describes the mask reticle for imaging.
- **MATERIAL** sets the coefficients of various materials.
- **METHOD** sets the numerical options or models for solving the equations.
- **MOMENTS** specifies moments for Pearson implant model.
- **OPTICAL** specifies the coefficients of reflection and refraction.
- **OXIDE** specifies oxidation coefficients.
- **PROJECTION** defines the photolithographic projection system.
- **PUPIL**.**FILTER** defines filters in the pupil plane.
- **RATE**.**DEPO** specifies deposition rates for machine type deposits.
- **RATE**.**DEVELOP** specifies development rates and other photoresist parameters.
- **RATE**.**ETCH** specifies the etch rate for machine etches.
- **RATE**.**POLISH** specifies polishing parameters for definition of a polishing machine.
- **SILICIDE** sets the coefficients for silicidation reactions.
- **TRAP** sets the coefficients of trap kinetics.
- **VACANCY** sets the coefficients of vacancy kinetics.

6.2.5 Special DECKBUILD Statements

These statements invoke special operations when running under DeckBuild. For more information on these statements, see the DeckBuild User's Manual.

- **AUTOELECTRODE** defines layout-based electrodes.
- **EXTRACT** extracts parameters.
- **GO** indicates interfacing between simulators.
- **MASK** performs photoresist deposition and etching through the MaskViews interface.
- **SET** sets the value of a user-defined variable.
- **SYSTEM** allows execution of any UNIX C-shell command within an input file.
- **TONYPLOT** creates a plot using TonyPlot.

6.2.6 Post-processing Statements

Starting from version 4.0, all internal plotting capabilities of former `SUPREM-IV` have been depreciated. Enhanced superior capabilities are available through TonyPlot and other VWF Interactive Tools. Only the following two post-processing statements remain.

- **PRINT.1D** is used to print the values (data points and profile information).
- **SELECT** allows a variable to be chosen as the z coordinate for the `PRINT.ID` command to follow.

6.2.7 Execution Control Statements

These statements control some execution capabilities. Some of them are useful only in a batch mode when Athena is run outside DeckBuild.

- **COMMENT** is used to document the input file.
- **CPULOG** instructs Athena to output CPU statistics.
- **FOREACH/END** specifies the command looping facility.
- **HELP** prints summary of statement names and parameters.
- **OPTION** specifies the level of run-time output.
- **PRINTF** parses a string or expression and places result into standard output.
- **QUIT** terminates execution of Athena.
- **SETMODE** sets execution mode parameters.
- **SOURCE** causes Athena to read statements from the specified file.
- **UNSETMODE** sets execution mode parameters to false.

6.2.8 Obsolete Statements

The following statements existed in earlier versions of Athena. Their capabilities were substituted either by superior capabilites of VWF Interactive Tools or included in other more advanced or generic statements.

- **ANTIMONY** is substituted by the `I.ANTIMONY` parameter in the `IMPURITY` statement.
- **ARSENIC** is substituted by the `I.ARSENIC` parameter in the `IMPURITY` statement.
- **BORON** is substituted by the `I.BORON` parameter in the `IMPURITY` statement.
- **COLOR** - all plotting capabilities are now provided by TonyPlot.
- **CONTOUR -** all plotting capabilities are now provided by TonyPlot.
- **ECHO** is a synonym of the `PRINTF` statement

- **DEFINE** is substituted by the SET capability in DeckBuild.
- **INDIUM** is substituted by the I.INDIUM parameter in the IMPURITY statement
- **LABEL -** all plotting capabilities are now provided by TonyPlot
- **PAUSE** is substituted by the **Pause** button in DeckBuild.
- **PHOSPHORUS** is substituted by I.PHOSPHOR parameter in the IMPURITY statement
- **PLOT.1D -** all plotting capabilities are now provided by TonyPlot
- **PLOT.2D -** all plotting capabilities are now provided by TonyPlot.
- **PLOT.3D -** all plotting capabilities are now provided by TonyPlot.
- **UNDEFINE** is substituted by the SET capability in DeckBuild.
- **VIEWPORT -** all plotting capabilities are now provided by TonyPlot.

6.2.9 Standard and User-Defined Materials

Different materials can be specified as parameters in various statements. Both standard and user-defined materials are available in Athena. The generic name MATERIAL appeared in a statement syntax description signifies that you can only specify one of standard material names from the list below or user-defined material. The generic name MATERIALS appeared in a statement syntax description signifies that you can specify one or several standard and user-defined materials.

The following shows the standard material names currently available in Athena.

Semiconductors

SILICON, POLYSILICON, GAAS, ALGAAS, INGAAS, SIGE, INP, GERMANIUM, SIC_6H, SIC_4H, SIC_3C

Note: Since Athena parser doesn’t recognize parameter names that begin with numerals, non-standard names are used for Silicon Carbides: SIC_6H, SIC_4H, and SIC_3C. Standard names: 6H-SiC, 4h-SiC, and 3C-SiC are used for these materials outside Athena (e.g. TonyPlot and Atlas).

Insulators

OXIDE, OXYNITRIDE, NITRIDE, PHOTORESIST

Metals

ALUMINUM, TUNGSTEN, TITANIUM, PLATINUM, COBALT

Silicides

WSIX (Tungsten Silicide), TISIX (Titanium Silicide), PTSIX (Platinum Silicide), COSIX (Cobalt Silicide)

Special Materials

GAS is used only in the IMPURITY, INTERSTITIAL and VACANCY statements to specify some parameters (i.e., segregation) at exposed boundaries.

BARRIER is a fictitious material. It can be specified only in DEPOSIT and ETCH statements and serve as a masking material.

User-Defined Materials

User-defined materials can be specified by MATERIAL = <c>, where <c> could be a single word MATERIAL=OXIDE1 or any string in double quotes as MATERIAL="MY INSULATOR". The user-defined material with the names exactly corresponding to Silvaco standard material names are saved in Silvaco Structure Files as those standard materials and will be recognized as such by other tools, e.g., DevEdit and Atlas. The following is the list of those Silvaco standard materials that can be used in Athena as previously described:

“Fictive GaAs”, AlInAs, AlAs, "Alpha Si 1", "Alpha Si 2", "Alpha Si 3","Alpha Si 4", "AlxGa1_xAs_x_0.25", "AlxGa1_xAs_x_0.5", "AlxGa1_xAs_x_0.75", "InxGa1_xAs_x_0.50 Unstr", "InxGa1_xAs_x_0.33 Str GaAs", "InxGa1_xAs_x_0.75 Str InP", "AlxIn1_xAs_x_0.50", Diamond, AlP, AlSb, GaSb, GaP, InSb, InAs, ZnS, ZnSe, ZnTe, CdS, CdSe, CdTe, HgS, HgTe, PbSe, PbTe, SnTe, ScN, GaN, AlN, InN, BeTe, InGaP, GaSbP, GaSbAs, InAlAs, InAsP, GaAsP, HgCdTe, CdZnTe, InGaAsP, AlGaAsP, AlGaAsSb, SiN, Si, CuInGaSe, InGaN, AlGaN, InAlGaN, InGaNAs, InGaNP, AlGaNAs, AlGaNP, AlInNAs, AlInNP, InAlGaAs, InAlGaP, InAlAsP, Pentacene, Alq3, TPD, PPV, Organic, InAlP,CdZnTe, Tetracene, ZnO, InGaSb, InAlSb, GaAsSb, AlAsSb, InGaAsSb, InAlAsSb, InPSb, InPAsSb, AlPSb, AlPAs, AlGaP, IGZO, NPB, InAlN,CuPc, NPD, CBP, Irppy, BAlq, CdO, CDZnO, MgCdO, MGCdZnO, Sapphire, Vacuum, TEOS, BSG, BPSG, PMMA, SOG, Polyimide, "Cooling package material", Ambient, Air, Insulator, Polymer, ITO, HfO2, HfSiO4, ZrO2, TiO2, Al2O3, PML, Lens, SixNyHz, In2O3, Tio, SnO2, Gold, Silver, AlSi, Palladium, Molybdinum, Lead, Iron, Tantalum, AlSiTi, AlSiCu, TiW, Copper, Tin, Nickel, WSix, NiSix, TaSix, PdSix, MoSix, ZrSix, AlSix, Conductor, Contact, Ba2YCu3O7, Ba2NdCu3O7, TiON, GST, GsSBTe, and PCM.

The generic name /MATERIAL specifies the second material in those statements, which specify parameters related to the boundary between two materials.

6.2.10 Standard Impurities

Different impurities can be specified as parameters in various statements. The generic name IMPURITY appeared in a statement syntax description signifies that you can only specify one of impurity names from the list below. The generic name IMPURITIES appeared in a statement syntax description signifies that you can specify one or more impurity names from the list below in the statement simultaneously. The impurity names below can appear as is or as a part of a parameter name, e.g. I.BORON, C.BORON and F.BORON. The following is the list of standard impurity names currently available in Athena:

ALUMINUM, ANTIMONY, ARSENIC, BERYLLIUM, BORON, CARBON, CHROMIUM, FLUORINE, GALLIUM, GERMANIUM, GOLD, HELIUM, HYDROGEN, INDIUM, MAGNESIUM, NITROGEN, OXYGEN, PHOSPHORUS, SELENIUM, SILICON, and ZINC

6.3 ABERRATION

ABERRATION defines aberration parameters of the optical projection system.

Note: This statement is not available in Athena 1D.

Syntax

```
ABERRATION
[X.FIELD=<n>] [Z.FIELD=<n>]  [SPHERICAL=<n>]  [COMA=<n>]
[ASTIGMATISM=<n>]  [CURVATURE=<n>]  [DISTORTION=<n>]
[FIFTH|SEVENTH|NINTH]
C1=<n>] [C2=<n>] [C3=<n>] [C4=<n>] [C5=<n>]
[C6=<n>] [C7=<n>] [C8=<n>] [C9=<n>] [C10=<n>] [C11=<n>]
[C12=<n>] [C13=<n>] [C14=<n>] [C15=<n>] [C16=<n>] [C17=<n>]
[C18=<n>] [C19=<n>] [C20=<n>]
```

Description

This statement specifies the aberration coefficients in the power series expansion of the wave aberration function. Each coefficient is entered in fractions of a wavelength in the range $0 \leq C \leq 0.5$.

X.FIELD and **Z.FIELD** define or change the position in the image field for which the irradiance distribution is to be computed. Note that the position is expressed in fractional field coordinates, so that the values for the x and z directions vary between -1.0 and 1.0.

SPHERICAL specifies 0C40, the amount of third order spherical aberration present in the power series expansion of the wave aberration function of the optical projector.

COMA specifies 1C31, which is the amount of third order coma present in the power series expansion of the optical projector.

ASTIGMATISM specifies 2C22, which is the amount of third order astigmatism present in the power series expansion of the optical projector.

CURVATURE specifies 2C20, which is the amount of third order field curvature present in the power series expansion of the optical projector.

DISTORTION specifies 3C11, which is the amount of third order distortion present in the power series expansion of the optical projector.

FIFTH, **SEVENTH**, and **NINTH** specify the aberration order. Coefficients for only one aberration order can be specified on a single statement.

C1, C2, C3, C4, C5, C6, C7, C8, C9, C10, C11, C12, C13, C14, C15, C16, C17, C18, C19, and **C20** are described in Table 6-3. Coefficients for fifth, seventh, and ninth order aberrations must be entered in separate ABERRATION commands for each order. Each of these parameters represents a particular aberration coefficient depending on the order specified by parameters FIFTH, SEVENTH, or NINTH.

Table 6-3 Aberration Coefficients

Parameter	Fifth	Seventh	Ninth
C1	4C20	6C20	8C20
C2	2C40	4C40	6C40
C3	0C60	2C60	4C60
C4	5C11	0C80	2C80
C5	3C31	7C11	0C100
C6	1C51	5C31	9C11
C7	4C22	3C51	7C31
C8	2C42	1C71	5C51
C9	3C33	6C22	3C71
C10		4C42	1C91
C11		2C62	8C22
C12		5C33	6C42
C13		3C53	4C62
C14		4C44	2C82
C15			7C33
C16			5C53
C17			3C73
C18			6C44
C19			4C64
C20			5C55

Examples

If high order aberrations are to be studied, they must be entered on a separate command line for each order.

```
ABERRATION X.FIELD=.5 SPHERICAL=.25

ABERRATION FIFTH C1=.25 C2=.5

ABERRATION SEVENTH C1=.3 C4=.4
```

For more examples, see IMAGE, ILLUMINATION, PROJECTION, ILLUM.FILTER, PUPIL.FILTER, LAYOUT.

6.4 ADAPT.MESH

`ADAPT.MESH` enables the adaptive meshing algorithm.

Note: This statement is not available in Athena 1D.

Syntax

```
ADAPT.MESH

[SMOOTH]      [SMTH.COUNT=<n>]      [ADAPT]      [ADAPT.COUNT=<n>]
[ADD.I.LINE=<n>]

[SENSITIVITY] [MATERIAL] [/MATERIAL]
```

Description

This statement runs the adaptive meshing algorithm or the smoothing algorithm in standalone mode.

SMOOTH flag to do mesh smoothing.

SMTH.COUNT specifies the number of smooth loops during the smooth operation. The default is 1.

ADAPT flag to do stand alone mesh adapting. Specifies that a stand alone adaptive meshing step should be performed to refine or relax the current mesh based on the material/impurity specification given on `ADAPT.PAR` command The default is `False`.

ADAPT.COUNT specifies the number of adapting loops during the stand alone adaptive meshing operation The default is 1.

ADD.I.LINE depth of the shadow interface mesh line in microns. Add the mesh line at the interface between two materials as defined by the booleans `MATERIAL` and `/MATERIAL`. The line is added in `MATERIAL` at a distance `ADD.I.LINE` from `/MATERIAL`.

SENSITIVITY specifies sensitivity of adaptation algorithm. The lower value leads to grid with more triangles. The default is 1.0.

MATERIAL one of standard materials or user specified material (see Section 6.2.9 “Standard and User-Defined Materials” for the list of materials).

/MATERIAL one of standard materials or user specified material (see Section 6.2.9 “Standard and User-Defined Materials” for the list of materials).

Examples

The following statement will add a set of lines close to the silicon/silicon dioxide interface. For this to work, the existing grid spacing at the interface must be greater than 0.005 microns. Note that since the lines are added only between existing mesh lines and the interface the lines must be specified in this order (i.e., getting closer to the surface)

```
ADAPT.MESH SILICON /OXIDE ADD.I.LINE=0.005

ADAPT.MESH SILICON /OXIDE ADD.I.LINE=0.001

ADAPT.MESH SILICON /OXIDE ADD.I.LINE=0.0005

ADAPT.MESH SILICON /OXIDE ADD.I.LINE=0.0001
```

For more examples, see `ADAPT.PAR`.

6.5 ADAPT.PAR

ADAPT.PAR specifies adaptive meshing parameters.

Note: This statement is not available in Athena 1D.

Syntax

```
ADAPT.PAR

[MATERIALS] [I.IMPURITIES] [I.INTERST] [I.VACANCY] [DISABLE]

[MAX.ERR=<n>]    [MIN.ERR=<n>]    [CONC.MIN=<n>]    [AREA.MIN=<n>]
[AREA.MAX=<n>]

[EDGE.MIN=<n>] [EDGE.MAX=<n>] [MIN.ADD=<n>]

[MAX.POINT=<n>]    [MAX.LOOP=<n>]    [IMPL.SMOOTH]    [DIFF.SMOOTH]
[IMPL.SUB]

[DOSE.ERR=<n>] [DOSE.MIN=<n>] [DIFF.LENGTH=<n>] [ANISOTROPIC]
```

Description

ADAPT.PAR specifies parameters used during adaptive meshing enabled by the ADAPT.MESH statement.

MATERIALS specify standard materials or user specified material regions in which mesh adaptation takes place (see Section 6.2.9 “Standard and User-Defined Materials” for the list of materials). One or several materials can be specified at a time.

I.IMPURITIES specify the impurities to be used for the grid adaptation (see Section 6.2.10 “Standard Impurities” for the list of impurity names that can be used, e.g. I.BORON). You can specify one or several impurities at a time.

I.INTERST specifies that interstitials to be used for the grid adaptation.

I.VACANCY specifies that vacancies to be used for the grid adaptation.

DISABLE specifies that the materials/impurities combinations given are disabled to be effective on mesh adapting or smoothing. Default is false.

MAX.**ERR** specifies the maximum error allowable before adding points to the mesh, (unitless). Error calculated above this value cause points to be added.

MIN.**ERR** specifies the minimum error below which points can be deleted from the mesh, (unitless). Error calculated below this value will remove points. Both MAX.ERR and MIN.ERR are calculated using the Bank-Weiser error estimator.

CONC.**MIN** specifies the minimum impurity concentration below which adapting will stop. Units are cm^{-3}. Default is 1.0 10^{14}cm^{-3}.

AREA.**MIN** specifies the minimum triangle area below which adding points will stop. Units are cm^{2}. Default is 1.0 10^{-15}

AREA.**MAX** specifies the maximum triangle area below which deleting points will stop. Units are cm^{2}. Default is 1.0 10^{-11}.

EDGE.MIN specifies the minimum edge length below which adding points will stop. Units are cm. Default is 1.0 10^{-6}.

EDGE.MAX specifies the maximum edge length below which deleting points will stop. Units are cm. Default is 1.0 10^{-5}.

MIN.ADD percent criteria to turn off implant adapt loop. MIN.ADD stops point addition in IMPLANT when the number of points added in the current loop is less than MIN.ADD * (total number of points). The default value for MIN.ADD = 0.05.

MAX.POINT specifies the maximum number of points above which adapting will stop. Default is 20000.

MAX.LOOP specifies the maximum loop count above which adapting will stop. This is only effective with implant. Default is 10.

IMPL.SMOOTH specifies which annealing algorithm to use after each adaption step. Currently, IMPL.SMOOTH=0 corresponds to no annealing during IMPLANT. IMPL.SMOOTH=1 corresponds to Laplacian smoothing and dose conservation interpolation algorithm. The default is IMPL.SMOOTH=1.

DIFF.SMOOTH specifies which annealing algorithm to use after each adaption step. Currently, DIFF.SMOOTH=0 corresponds to no annealing during DIFFUSE. DIFF.SMOOTH = 1 corresponds to Laplacian smoothing and dose conservation interpolation algorithm. The default is DIFF.SMOOTH=0.

IMPL.SUB flag to do grid subtracting in implant adapt. IMPL.SUB is a boolean flag that stops point removal during IMPLANT adaptive meshing. The default value for IMPL.SUB=false signifies that points are not being removed.

DOSE.ERR specifies dose error for the refinement/unrefinement.

DOSE.MIN specifies minimum of dose level for grid refinement during adaptation.

DIFF.LENGTH used to limit the activity of adaptation of grid during the simulation of dopant diffusion. This parameter will allow the mesh to adapt only after a given diffusion length for a given dopant and will override any other adaptation triggers based upon gradient error estimates. This is a useful control to limit the number of time steps. Units are microns.

ANISOTROPIC is the flag used to maintain the mesh to be anisotropic. The flag is material dependent.

Examples

The following is an example of setting the adaptive meshing parameters during diffusion for Boron.

```
IMPLANT BORON DOSE=1E15 ENERGY=60

ADAPT.PAR DIFF.LEN=0.1 SILICON I.BORON

DIFFUSE TEMP=1000 TIME=100 NITROGEN
```

For more examples, see ADAPT.MESH.

6.6 BAKE

BAKE performs post-exposure or post-development photoresist bake.

Note: This statement is not available in Athena 1D.

Syntax

```
BAKE
[DIFF.LENGTH=<n>] [TEMERATURE=<n>] [REFLOW]
[TIME] [SECONDS|MINUTES|HOURS]
[DUMP=<n>] [DUMP.PREFIX=<c>]
```

Description

This command runs a bake process using the diffusion length as the parameter that incorporates the bake temperature and bake time.

DIFF.LENGTH specifies the diffusion length for the post exposure bake. Default is 0.05 micrometers

TEMPERATURE specifies the temperature of the bake process in ^{o}C.

REFLOW specifies that material flow should be calculated during the bake process. Default is False.

TIME specifies the amount of time for the bake step in specified units. Default is MINUTES.

HOURS, **MINUTES**, and **SECONDS** specify the units of the TIME parameter.

DUMP and **DUMP.PREFIX** specify that a structure file be output at every DUMPth time step. The files are readable with the STRUCTURE statement or can be displayed using TonyPlot. The names will be of the form DUMP.PREFIX<time>.str, where <time> is the current total time of the simulation.

Examples

The BAKE command is entered with the user-specified diffusion length for post exposure bake:

```
BAKE DIFF.LENGTH=0.05
```

BAKE can also be entered with time and temperature parameters for post exposure bake:

```
BAKE TIME=45 TEMP=120
```

For photoresist reflow (post development bake), the above command is entered with the REFLOW parameter:

```
BAKE REFLOW TIME=45 TEMP=120
```

For more examples, see DIFFUSE and RATE.DEVELOP.

6.7 BASE.MESH

BASE.MESH specifies parameters of the base mesh used for initial grid generation.

Note: This statement is not available in Athena 1D.

Syntax

```
BASE.MESH
[SURF.LY=<N>] [SURF.DY=<N>] [ACTIVE.LY=<N>]
[ACTIVE.DY=<N>] [EPI.LY=<N>] [EPI.DY=<N>]
[SUB.LY=<N>] [SUB.DY=<N>] [BACK.LY=<N>] [BACK.DY=<N>]
```

Description

This command specifies parameters of the base mesh used for initial grid generation.

SURF.LY specifies the location of surface. The default is y=0.0 µm.

SURF.DY specifies the local grid spacing in y-direction at SURF.LY.

ACTIVE.LY, EPI.LY, SUB.LY specify another three base line location at some critical region of the device structure to be fabricated.

ACTIVE.DY, EPI.DY, SUB.DY specify the local grid spacing at the ACTIVE.LY, EPI.LY, SUB.LY. The units are all in microns.

BACK.LY and **BACK.DY** are the location of bottom in the structure to be fabricated and the BACK.DY is the local grid spacing in the BACK.LY location.

Examples

The following example assigns the initial base line for the substrate materials. It places the base line at y=1.0, y=2.0, y=10.0 with a local grid spacing of 0.01 µm, 0.5 µm, 1.0 µm and 10 µm separately. The device dimension in the y-direction is specified as y.main=0. y=max=100. This depth, however, is only for reference. The real depth and width of the device structure will be assigned in the INITIALIZE command.

```
BASE.MESH SURF.LY=0.0 SURF.DY=0.01\
ACTIVE.LY=1.0 ACTIVE.DY=0.5 EPI.LY=2.0 EPI.DY=1.0\
SUB.LY=10.0 SUB.DY=10.0 BACK.LY=500 BACK.DY=100
```

For more examples, see BASE.MESH and INITIALIZE.

6.8 BASE.PAR

`BASE.PAR` runs the base mesh for generating the initial grid.

Note: This statement is not available in Athena 1D.

Syntax

```
BASE.PAR

[MATERIAL] [GRAD.SPACE] [RATIO.BOX]
```

Description

MATERIAL one of standard materials or user specified material (see Section 6.2.9 “Standard and User-Defined Materials” for the list of materials).

GRAD.SPACE specifies the gradient of the adjacent grid spacing in the y-direction of this material. Default is 1.5.

RATIO.BOX specifies the approximate aspect ratio of triangle element after base mesh generation in this material. Default is 2.0.

Examples

The following example generates a good quality base mesh for each related material region.

```
BASE.PAR OXIDE       GRAD.SPACE=5    RATIO.BOX=2

BASE.PAR SILICON     GRAD.SPACE=1.5  RATIO.BOX=2

BASE.PAR POLYSILICON GRAD.SPACE=5    RATIO.BOX=2

BASE.PAR OXIDE       GRAD.SPACE=5    RATIO.BOX=2

BASE.PAR SILICON     GRAD.SPACE=1.5  RATIO.BOX=2

BASE.PAR POLYSILICON GRAD.SPACE=5    RATIO.BOX=2
```

For more examples, see `BASE.MESH`.

6.9 BOUNDARY

BOUNDARY specifies boundary conditions for the initial material.

Note: For most typical boundary conditions, Athena has defaults that eliminate the need for BOUNDARY statements. The BOUNDARY statement can be used to modify the treatment of the surfaces for special purpose simulations.

Syntax

```
BOUNDARY
[REFLECTING|EXPOSED|BACKSIDE]
[XLO=<c>] [YLO=<c>)] [XHI=<c>] [YHI=<c>]
```

Description

EXPOSED surfaces correspond to the top of the wafer. Only exposed surface have deposition or oxidation on top of them. A surface created by etching will also be exposed unless the ETCH NO.EXPOSE syntax is used.

REFLECTING surfaces correspond to the sides of the device and are also applicable to the backside as long as defects are not being simulated. All surfaces default to REFLECTING..

BACKSIDE surfaces are physically identical to the reflecting surface with special meaning only when backside electrode is specified in the ELECTRODE statement.

XLO, YLO, XHI, and **YHI** set the left, right, top, and bottom bounds of the rectangle being specified. The value string should be one of the tags specified in one of preceding line statements.

Examples

The following lines define the top of the mesh to be an exposed surface and the bottom to be the backside.

```
BOUNDARY EXPOSED XLO=LEFT XHI=RIGHT YLO=SURF YHI=SURF
BOUNDARY BACKSIDE XLO=LEFT XHI=RIGHT YLO=BACK YHI=BACK
```

For more examples, see REGION and INITIALIZE.

6.10 CLUSTER

CLUSTER specifies parameters of {311} cluster model.

Syntax

```
CLUSTER
[I.IMPURITY] [MATERIAL]
[CLUST.FACT=<n>] [MIN.CLUST=<n>] [MAX.CLUST=<n>]
[TAU.311.0=<n>] [TAU.311.E=<n>]
```

Description

This command specifies the scaling of {311} clusters during a subsequent IMPLANT step and the time constant for the dissolution of clusters into free interstitials.

Note: This command will only work if you switch on the {311} cluster model with the METHOD CLUSTER.DAM command.

I.IMPURITY specifies an impurities to be used for the {311} cluster scaling (see Section 6.2.10 "Standard Impurities" for the list of impurity names that can be used, e.g. I.BORON).

MATERIAL specifies a material in which the scaling takes place (see Section 6.2.9 "Standard and User-Defined Materials" for the list of materials). Default is SILICON.

MIN.CLUST and **MAX.CLUST** define two values of implanted dopant concentration. Clusters will be placed between these two dopant concentration levels only. These parameters are used to control the scaled position of clusters during ion implantation. Typically, MIN.CLUST is the background doping level. MAX.CLUST is the dopant concentration required to amorphize the substrate

CLUST.FACT specifies the ratio between the concentration of clustered interstitials and the implanted dopant concentration.

TAU.311.0=<n> and **TAU.311.E=<n>** specify the time constant in seconds for the dissolution of clusters into free interstitials. TAU.311.0 is the pre-exponential linear coefficient and TAU.311.E is the exponential coefficient used to control temperature dependence.

Examples

The following command introduces clusters during ion implantation. The clusters will have an effective interstitial concentration of 1.4 times the concentration of implanted boron. The clusters will lie in the region where Boron is between 10^{15} and 10^{19} cm^{-3}.

```
METHOD CLUSTER.DAM

CLUSTER    I.BORON    SILICON    MIN.CLUS=1e15    MAX.CLUST=1e19
CLUST.FACT=1.4

IMPLANT DOSE=1e14 ENERGY=50 BORON
```

The example goes on to define the cluster dissolution time and a short thermal cycle. Results for each timestep of the diffusion cycle will be stored in files RTA_*.

```
CLUSTER I.BORON SILICON TAU.311.0=10 TAU.311.E=0.24
DIFFUSE TEMP=1000 TIME=10/60 NITRO DUMP=1 DUMP.PREF=RTA_
```

For more examples, see `METHOD`, `DISLOC.LOOP`, `INTERSTITIAL`, and `VACANCY`..

6.11 COMMENT

`COMMENT` is used to specify character strings for documenting the input deck and Athena output.

Syntax

```
COMMENT
```

```
#
```

or

```
$
```

Description

The `COMMENT` statement, `#` or `$` are used to document the input file. You can insert them in the beginning of any line of the input deck.

6.12 CPULOG

CPULOG instructs Athena to output CPU statistics.

Syntax

```
CPULOG

[LOG] [CPUFILE = <c>]
```

Description

The CPULOG statement logs the CPU time used in various internal operations. The CPU time information appears in the standard output or in the DeckBuild Text Subwindow unless it is re-directed into CPUFILE.

LOG enables logging of CPU usage when true, and disables CPU logging when false. The default is true.

CPUFILE specifies a name of the file to which CPU log is written. The default is the standard output.

Examples

The following example enables Athena to gather CPU statistics and store it in the file timeusage.out.

```
CPULOG LOG CPUGFILE=timeusage.out
```

Note: The accuracy of time statistics depends on the computer and operating system. It is usually around 0.01 sec.

6.13 DEPOSIT

DEPOSIT deposits a layer of specified material. DEPOSITION is a synonym for this statement.

Note: Unless the Elite module is used, all deposition steps in Athena are 100% conformal. This means deposition on all surfaces with a step coverage of 1.0.

Syntax

```
DEPOSIT
MATERIAL [NAME.RESIST=<c>] THICKNESS=<n>
[SI_TO_POLY] [TEMPERATURE=<n>]
[DIVISIONS=<n>] [DY=<n>] [YDY=<n>] [MIN.DY=<n>] [MIN.SPACE=<n>]
[C.IMPURITIES=<n>]   [F.IMPURITIES=<n.]   [C.INTERST=<n>]
[F.INTERST=<n>]
[C.VACANCY=<n>]  [F.VACANCY=<n>]  [X.COMP=<n>]  [FX.COMP=<n>]
[C.FRACTION=<n>]
[F.FRACTION=<n>]  [Y.COMP=<n>]  [FY.COMP=<n>]  [GR.SIZE=<n>]
[F.GR.SIZE=<n>]
[MACHINE=<c>] [TIME=<n>] [HOURS|MINUTES|SECONDS]
[N.PARTICLE=<n>] [OUTFILE=<c>] [SUBSTEPS=<n>] [VOID]
```

Description

This statement is used to simulate deposition of specified material on the exposed surface of the current structure.

MATERIAL specifies the material to be deposited (see Section 6.2.9 “Standard and User-Defined Materials” for the list of materials).

NAME.**RESIST** specifies the type of photoresist to be deposited.

THICKNESS specifies the deposited layer thickness in microns.

SI_TO_POLY specifies that crystalline silicon will be deposited only over crystalline silicon, while polysilicon will be deposited elsewhere.

TEMPERATURE specifies deposition temperature used by STRESS.HIST model. The temperature is also used for surface diffusion simulation during Elite deposition.

Grid Control Parameters

DIVISIONS specifies the number of vertical grid spacings in the layer. In some cases, it is important to control the number of grid points in a conformally deposited layer since this also controls the accuracy of subsequent processes. SPACES is an alias for DIVISIONS. The default is 1.

Note: The default for DIVISIONS is 1. This typically needs to be increased for all deposition steps. If DIVISIONS is set too low to maintain grid integrity in a non-planar deposition, Athena will attempt to recover by increasing DIVISIONS automatically. Athena will echo the number of DIVISIONS finally used to the run-time output.

DY specifies the nominal spacing in the layer. Units are microns.

YDY specifies the depth where the nominal spacing will be applied. YDY is calculated relative to the top of the newly deposited layer. Units are microns.

MIN.DY specifies the minimum spacing in microns allowed between grid lines in the y direction in the new material. The default is 0.001 microns (10 Angstroms).

MIN.SPACE specifies a minimum spacing between points on the surface of each sub-layer. Increasing this parameter will reduce the number of points on arced deposited surfaces. Units are microns. ARC.SPACE is a synonym for this parameter.

Parameters Specific to Depositing Doped Layers

C.IMPURITIES specify the concentration of the impurity in the deposited layer in cm^{-3}. You can specify more than one of these parameters to define materials doped with multiple impurities.

Note: There are several additional methods to specify impurity concentration in deposited material. These methods are introduced to provide TSUPREM4 compatibility and described in Section C.8 “Changes in the DEPOSIT statement”.

F.IMPURITIES can only be specified together with the corresponding C.IMPURITY (e.g., F.BORON and C.BORON). This parameter generates the linearly graded concentration of the specified impurity in the deposited layer, where C.IMPURITY specifies concentration at the bottom of the layer and F.IMPURITY specifies concentration at the top of the layer. Units are cm^{-3}.

C.INTERST specifies the concentration of interstitials in deposited layer. Units are cm^{-3}.

F.INTERST can only be specified together with C.INTERST. This parameter generates the linearly graded interstitial concentration in the deposited layer, where C.INTERST specifies concentration at the bottom of the layer and F.INTERST specifies concentration at the top of the layer.

C.VACANCY specifies the concentration of vacancies in deposited layer. Units are cm^{-3}.

F.VACANCY can only be specified together with C.VACANCY. This parameter generates the linearly graded vacancy concentration in the deposited layer, where C.VACANCY specifies concentration at the bottom of the layer and F.VACANCY specifies concentration at the top of the layer. Units are cm^{-3}.

X.COMP specifies the composition fraction of the cation listed first in ternary or quaternary compound semiconductor to be deposited. For example, Al is usually first cation of AlGaAs. The full list of compound semiconductors that can be specified in Athena and the rules of their naming can be found in Appendix B of Atlas User’s Manual. For backward compatibility, C.FRACTION is the alias for X.COMP.

FX.COMP can only be specified together with X.COMP. This parameter allows you to generate the compound semiconductor layer with linearly graded composition fraction of the cation. If both X.COMP and FX.COMP are specified, the cation composition fraction will be X.COMP at the bottom of the deposited layer and FX.COMP at the top of the new layer. For backward compatibility, F.FRACTION is the alias for FX.COMP.

Y.COMP specifies the composition fraction of the anion listed first in quaternary compound semiconductor to be deposited. For example, As is first anion in InGaAsP. The full list of compound semiconductors that can be specified in Athena and the rules of their naming can be found in Appendix B of Atlas User's Manual.

FY.COMP can only be specified together with `Y.COMP`. This parameter allows you to generate the compound semiconductor layer with linearly graded composition fraction of the anion. If both `Y.COMP` and `FY.COMP` are specified, the anion composition fraction will be `Y.COMP` at the bottom of the deposited layer and `FY.COMP` at the top of the new layer.

GR.SIZE specifies grain size in deposited polysilicon layer. This parameter is recognized only when `POLY.DIFF` model is specified in the `METHOD` statement. Units are microns.

F.GR.SIZE can only be specified together with `GR.SIZE`. This parameter deposits polysilicon layer with grains linearly graded with their sizes, where `GR.SIZE` specifies grain size at the bottom of the layer and `F.GRAIN.SIZE` specifies grain size at the top of the layer. Units are microns.

Parameters Specific to Elite Depositions

MACHINE specifies the name of the machine to be run for Elite deposits. The machine name must be specified in a previous `RATE.ETCH` statement.

TIME sets the time in specified units the etch machine will be running.

HOURS, MINUTES, and **SECONDS** specifies the units of the `TIME` parameter. Default is `MINUTES`.

N.**PARTICLE** specifies the number of particle trajectories to calculate for the Monte Carlo deposit model.

OUTFILE specifies the name of the file to be written with Monte Carlo particle positions.

SUBSTEPS specifies the number of timesteps made for each division of the deposit in the Elite module.

VOID specifies that the voids formed during deposition are to remain unfilled with deposit material.

Conformal Deposition Example

The following statement deposits a conformal layer of silicon dioxide, 1000 Angstroms thick, on the surface of the simulation structure. It will contain 4 vertical grid points.

```
DEPOSIT OXIDE THICK=0.1 DIVISIONS=4
```

Example Depositing Doped User-defined Material

The following deposits a layer of a user defined material BPSG doped with boron and phosphorus.

```
DEPOSIT   MATERIAL=BPSG   THICKNESS=0.1   DIV=6   C.BORON=1e20
C.PHOS=1e20
```

Grid Control Example

The following statement deposits a conformal layer of silicon nitride with a thickness of 0.3µm. The grid spacing at the bottom of the layer is 0.01µm and the layer will include 10 vertical sublayers.

```
DEPOSIT NITRIDE THICK=0.3 DY=0.1 YDY=0.3 DIVISIONS=10
```

Elite Machine Deposition Example

The following statements define a machine named MOCVD and use it to deposit tungsten with a thickness of 0.1µm on planar areas and step coverage of 0.75.

```
RATE.DEPO MACHINE=MOCVD DEP.RATE=.1 u.m STEP.COV=.75 TUNGSTEN
```

```
DEPOSIT MACHINE=MOCVD TIME=1 MINUTE
```

For more examples, see RATE.DEPO.

6.14 DEVELOP

DEVELOP runs the development module in Optolith.

Note: This statement is not available in Athena 1D.

Syntax

```
DEVELOP
[MACK|DILL|TREFONAS|HIRAI|KIM|EIB]
[TIME=<n>] [STEPS=<n>] [SUBSTEPS=<n>]
[DUMP=<n>] [DUMP.PREFIX=<c>]
```

Description

This command runs the development module and enables the use of the option to select a development model.

MACK, DILL, TREFONAS, HIRAI, KIM and **EIB** specify the development model to be used.

TIME, STEPS, and **SUBSTEPS** are related parameters that control the string algorithm in development. TIME is the total development time in seconds. STEPS gives the number of times ETCH is to be performed. SUBSTEPS controls string movement. Each substep or string movement has a time duration of TIME/STEP*SUBSTEPS.

DUMP determines whether a structure is saved after each step of the development is completed.

DUMP.PREFIX specifies the prefix name for the structure file to be saved. The number of steps will be equal to the number of output files. The files are readable with the STRUCTURE statement or can be displayed using TonyPlot. The names of the files will be of the form DUMP.PREFIX***.***.str, where ** is the current development time.

Examples

The following example dumps out five structure files to show the evolution of development using the KIM development model.

```
DEVELOP KIM DUMP=1 TIME=60 STEPS=5
```

For more examples, see RATE.DEVELOP.

6.15 DIFFUSE

DIFFUSE runs a time temperature step on the wafer and calculates oxidation, silicidation and diffusion of impurities. DIFFUSION is a synonym for this statement.

Syntax

```
DIFFUSE
TIME=<n> [HOURS|MINUTES|SECONDS]
TEMPERATURE=<n> [T.FINAL=<n>|T.RATE=<n>]
 [DRYO2|WETO2|NITROGEN|INERT] [HCL.PC=<n>] [PRESSURE=<n>]
 [F.02=<n>|F.H2=<n>|F.H20=<n>|F.N2=<n>|F.HCL=<n>]
 [C.IMPURITIES=<n>]
 [DUMP][DUMP.PREFIX=<c>] [TSAVE=<n>] [TSAVE.MULT=<n>]
 [B.MOD=<c>] [p.MOD=<c>] [AS.MOD=<c>] [IC.MOD=<c>] [VI.MOD=<c>]
 [REFLOW]
```

Description

This command specifies diffusion and/or oxidation/silicidation steps. Any impurities present in the wafer are diffused if they have non-zero diffusivities. The oxidation and diffusion control parameters are contained in the associated METHOD, OXIDE, and SILICIDE statements. Default coefficients are in the ATHENAMOD file available from the DeckBuild **Commands** menu under **Models**. To change model coefficients, refer to the appropriate IMPURITY statement for information.

Parameters to Define the Diffusion Step

TIME specifies the amount of time for the diffusion step in specified units. If TIME is set to zero, only stress simulation will be performed when STRESS.HIST is specified .

HOURS, **MINUTES**, and **SECONDS** specify the units of the TIME parameter. Default is MINUTES.

TEMPERATURE specifies the ambient temperature in ^{o}C. This temperature should fall within the range between 700 and 1200^{o}C. Outside of this range, the diffusion coefficients may be inaccurate and numerical difficulties may occur during simulation. For ramped thermal step, a synonym T.START can be used.

T.FINAL specifies the final temperature for ramped thermal steps. Synonym is T.STOP.

T.RATE specifies the ramp rate in ^{o}C/minute for ramped thermal steps.

Parameters to Define the Diffusion Ambient

DRYO2, WETO2, INERY and **NITROGEN** specify the type of ambient during the diffusion step. DRYO2 specifies that ambient is dry oxygen, WETO2 specifies that ambient is wet oxygen, NITROGEN specifies that ambient is inert. INERT is a synonym for NITROGEN.

HCL.PC specifies the percentage of HCl in the oxidant gas stream.

PRESSURE specifies the partial pressure of the active species in atmospheres. Units are atmospheres. The default is 1.

F.02, F.H2, F.H20, F.N2, and **F.HCL** specifies the relative flow rate of the components of oxygen, hydrogen, water, nitrogen, and HCl in the ambient. If these parameters are used, the DRYO2, WETO2, NITROGEN set or HCL.PC should not be specified.

C.IMPURITIES specifies concentration of the impurities in the ambient gas. (see Section 6.2.10 "Standard Impurities" for the list of impurity names that can be used, e.g. I.BORON). Units are atoms/cm^3. You can define multiple impurity parameters for ambients with multiple impurities. You can only use boron, phosphorus and arsenic if you specify the advanced diffusion model (PLS) in the METHOD statement.

Note: There are several additional methods to specify impurity concentration in ambient gas. These methods are introduced to provide TSUPREM4 compatibility and described in Section C.9 "Changes in the DIFFUSE statement".

Parameters Related to File Output

DUMP and **DUMP.PREFIX** specify that a structure file be output at every DUMPth time step. The files are readable with the STRUCTURE statement or can be displayed using TonyPlot. The names will be of the form DUMP.PREFIX<time>.str, where <time> is the current total time of the simulation in minutes.

TSAVE and **TSAVE.MULT** specify that intermediate structure files be output when the advanced PLS diffusion model is used. The structure files named DUMP.PREFIX<time>.str will be output at time = TSAVE * TSAVE.MULT^k, *k=0,1,2,...* where time is in seconds. The default value for the parameter DUMP.PREFIX is "at". The parameter TSAVE.MULT should be greater than 1.0.

Parameters Related to the Model Files for Advanced Diffusion Models

B.MOD, P.MOD, AS.MOD, IC.MOD, and **VI.MOD** specify direct paths to boron.mod, phosphorus.mod, arsenic.mod, i.mod, and defect.mod files correspondingly. By default, these files are in $SILVACO/lib/athena/<version_number>/common/pls directory. You can modify your own *.mod files inside directories specified by these parameters.

Miscellaneous Parameters

REFLOW specifies that a surface tension based reflow of the material is to be performed during the diffusion step.

Predeposition Example

The following statement specifies a 1000°, 30 minute boron pre-deposition.

```
DIFFUSE TIME=30 TEMP=1000 C.BORON=1.0E20
```

Oxidation Example

The following statement instructs the simulator to grow oxide for 30 minutes in a dry oxygen ambient.

```
DIFFUSE TIME=30 TEMP=1000 DRYO2
```

Gas Flow Example

The following command performs diffusion with a mixed ambient with relative components of oxygen, hydrogen, and HC1 of 10, 10, and 0.1 respectively.

```
DIFFUSE TIME=10 TEMP=1000  F.02=10 F.H2=10 F.HCl=.1
```

Hydrogen and Oxygen are combined in a ratio 2:1 to form the ambient WETO2. Any excess hydrogen is considered inert. Any excess oxygen is considered as the ambient DRYO2. Since the total pressure of the gas flow is defined (or defaults to one atmosphere) the partial pressure of WETO2 will be reduced if any excess hydrogen or oxygen is present.

File Output Example

The following commands perform diffusion in dry oxygen ambient for 30 minutes at 1000 ^{o}C. After every second timestep a structure file is written with a name prefix TEST. Following the diffusion, the TonyPlot statement plots each timestep output file in a manner suitable for creating a diffusion movie. A SYSTEM command is used to execute a UNIX command prior to the diffusion step to remove all TEST*.str files from previous runs.

```
SYSTEM rm -rf TEST*.str

DIFFUSE TIME=30 TEMP=1000 DRYO2 DUMP=2 DUMP.PREFIX=TEST

TONYPLOT -st TEST*.str
```

Advanced Diffusion Model Example

The following command performs the boron pre-deposition at 950°C for 1 hour. The boron concentration in the ambient gas is 10^{20}cm^{-3}. As a result, the output files predep1.str, predep10.str, predep100.str, and predep1000.str will be saved.

```
METHOD PLS

DIFFUSE TIME=1 HOUR TEMP=950 C.BORON=1E20 TSAVE=1 TSAVE.MULT=10

DUMP.PREFIX=predep
```

For more examples, see IMPURITY, INTERSTITAL, MATERIAL, METHOD, OXIDE, TRAP, and VACANCY.

6.16 DISLOC.LOOP

DISLOC.LOOP defines the scaling parameters and position of dislocation loops.

Syntax

```
DISLOC.LOOP

MATERIAL I.IMPURITY MIN.LOOP.CO =<n> MAX.LOOP.CO=<n>
```

Description

This command specifies the scaling of dislocation loops during a subsequent IMPLANT step. Dislocation loops are used as interstitial sinks whose recombination rate can be determined with the INTERSTITIAL DAMALPHA=<n> command.

Note: This command will only work if you switch on the dislocation loop model with the METHOD I.LOOP.SINK command.

MATERIAL specifies material for which dislocation loops parameters are set (see Section 6.2.9 “Standard and User-Defined Materials” for the list of materials). Default is SILICON.

I.IMPURITY specifies an impurities to be used for the dislocation loop scaling (see Section 6.2.10 “Standard Impurities” for the list of impurity names which can be used, e.g. I.BORON).

MIN.LOOP.CO and **MAX.LOOP.CO** define the upper and lower bounds of the dopant concentrations where the loops are placed.

Dislocation Loop Generation Example

The following example switches on the loop model and then places loops in the position where indium concentrations lie between 1e16 and 1e15 cm^{-3}.

```
METHOD I.LOOP.SINK

DISLOC.LOOP MIN.LOOP.CO=1e15 MAX.LOOP.CO=1e16 I.INDIUM SILICON

IMPLANT INDIUM DOSE=1e15 ENERGY=45
```

For more examples, see METHOD, CLUSTER, INTERSTITIAL, VACANCY, DIFFUSE, and IMPLANT.

6.17 ELECTRODE

ELECTRODE defines electrodes and names for Atlas or other device simulation.

Note: This statement is not available in Athena 1D.

Syntax

```
ELECTRODE

NAME=<c> [X=<n> | Y=<n> | BACKSIDE | LEFT | RIGHT]
```

Description

This statement defines a whole material region as an electrode.

NAME gives a name to the electrode that can be plotted or referenced in TonyPlot or Atlas.

X specifies the horizontal location or x-coordinate of the region, which will be defined as an electrode.

Y specifies the vertical location or y-coordinate of the electrode being defined. If no value of Y is specified, the top of the structure is assumed.

BACKSIDE specifies that a flat (zero height) electrode will be placed on the bottom of the simulation structure. This is the one exception to whole regions being defined as electrodes. If a metal region is present on the bottom of the structure, this parameter will not be used and the XY coordinates used instead. BOTTOM is a synonym for this parameter.

LEFT specifies that the top left region of the structure will be defined as an electrode.

RIGHT specifies that the top right region of the structure will be defined as an electrode.

Note: The ELECTRODE statement recognizes the regions made of polysilicon, standard metals (see.Section 6.2.9 “Standard and User-Defined Materials” for the list of standard metals) or user-defined materials with the following standard names: Gold, Silver, AlSi, Palladium, Molybdenum, Lead, Iron, Tantalum, AlSiTi, AlSiCu, TiW, Copper, Tin, Nickel, NiSix, TaSix, PaSix, MoSix, ZrSix, AlSix, Conductor, Contact.

Note: Atlas contains syntax that makes use of the common electrical names for highly preferred terminals. These are anode, cathode, emitter, base, collector, gate, source, drain, bulk and substrate.

Metal Region Electrode Definition Example

The following gives the name source to the metal or polysilicon region at location x=1 micron on the top of the current structure.

```
ELECTRODE X=1.0 NAME=SOURCE
```

Substrate Definition Example

The following gives the name well to a flat electrode along the bottom edge of the current structure There is no metal required at this location.

```
ELECTRODE BACKSIDE NAME=WELL
```

For more examples, see STRUCTURE.

6.18 EPITAXY

EPITAXY specifies an epitaxial deposition process step.

Syntax

```
EPITAXY

TIME=<n>  [HOURS|MINUTES|SECONDS]

TEMPERATURE=<n> [T.FINAL=<n> | T.RATE=<n>]

[THICKNESS=<n> | GROWTH.RATE=<n>]

[C.IMPURITIES=<n>]        [F.IMPURITIES=<n.]        [C.INTERST=<n>]
[F.INTERST=<n>]

[C.VACANCY=<n>]  [F.VACANCY=<n>]

[DIVISIONS=<n>] [DY=<n>] [MIN.DY=<n>] [YDY=<n>] [SI_TO_POLY]
```

Description

This statement simulates the epitaxial deposition of silicon. This model is limited to silicon on silicon applications and should not be used when other materials are present. The model is inherently 1D and isn't suitable for selective epitaxial deposition processes.

Parameters to Define the Epitaxial Step

TIME specifies the amount of time for the epitaxial step in specified units.

HOURS, **MINUTES**, and **SECONDS** specify the units of the TIME parameter. Default is MINUTES.

TEMPERATURE specifies the ambient temperature, in ^{o}C. This temperature should fall within the range between 700 and 1200^{o}C. Outside of this range, the diffusion coefficients may be inaccurate and numerical difficulties may occur during simulation. For ramped thermal step, a synonym T.START can be used.

T.FINAL specifies the final temperature for ramped thermal steps. Synonym is T.STOP.

T.RATE specifies the ramp rate in ^{o}C/minute for ramped thermal steps.

THICKNESS specifies thickness of epitaxially grown layer. Units are microns.

GROWTH.RATE specifies epitaxial growth rate. It is applicable only when THICKNESS is not specified. Units are µm/minute.

Doping Related Parameters

C.IMPURITIES specify the concentration of the impurity in the epitaxially grown layer in cm^{-3}. You can specify more than one of these parameters to define materials doped with multiple impurities.

F.IMPURITIES can be specified only together with the corresponding C.IMPURITY (e.g. F.BORON and C.BORON). This parameter generates the linearly graded concentration of the specified impurity in the epitaxially grown layer, where C.IMPURITY specifies concentration at the bottom of the layer and F.IMPURITY specifies concentration at the top of the layer.

C.INTERST specifies the concentration of interstitials in the epitaxially grown layer in cm^{-3}.

F.INTERST can be specified only together with C.INTERST. This parameter generates the linearly graded interstitial concentration in the epitaxially grown layer, where C.INTERST specifies concentration at the bottom of the layer and F.INTERST specifies concentration at the top of the layer.

C.VACANCY specifies the concentration of vacancies in the epitaxially grown layer in cm^{-3}.

F.VACANCY can be specified only together with C.VACANCY. This parameter generates the linearly graded vacancy concentration in the deposited layer, where C.VACANCY specifies concentration at the bottom of the layer and F.VACANCY specifies concentration at the top of the layer.

Gridding Parameters

DIVISIONS controls the number of vertical grid points in the resulting epitaxial layer. This is an optional parameter since it will be generated automatically by default and is related to the surface grid spacing of the original simulation structure before epitaxial process. SPACES is an alias for this parameter. The default is 10.

DY specifies the nominal spacing in microns in the epitaxial layer.

YDY specifies the depth where the nominal spacing will be applied. YDY is calculated relative to the top of the newly grown epitaxial layer.

MIN.DY specifies the minimum spacing allowed between grid lines in the y direction in the new material. The default is 0.001 microns (10 Angstroms).

SI_TO_POLY specifies that the crystalline silicon layer will be grown only over Silicon, while Polysilicon will be grown elsewhere.

Deposition Rate Example

The following statement will simulate the growth of boron doped silicon on top of silicon at a rate of 0.5 µm per minute. The deposit thickness is time x rate=5 µm.

```
EPITAXY TIME=10 TEMP=1150 C.BORON=5E14 GROWTH.RATE=0.5
```

Time and Temperature Example

The following statement will deposit 6µm of epitaxial silicon on top of silicon over 10 minutes. Phosphorus is out-diffused during the processing. The number of vertical grid points in the completed epitaxial layer is set with the DIVISIONS parameter. The syntax is similar to the DEPOSIT statement.

```
EPITAXY THICK=6 TIME=10 TEMP=1180 C.PHOS=1.5E14 DIVISIONS=20
```

Non-uniform Grid Control Example

The following statement performs epitaxial process with a non-uniform vertical grid spacing. The vertical grid spacing will be 0.5 µm at a distance of 5 µm below the final surface. The epitaxial layer will be subdivided into 40 sublayers:

```
EPITAXY THICK=10 TIME=30 TEMP=1100 DY=.5 YDY=5.0 DIVISIONS=40
```

For more examples, see DEPOSIT and DIFFUSE.

6.19 ETCH

ETCH simulates an etch process.

Syntax

```
ETCH
[MATERIAL]  [NAME.RESIST]
[ALL|DRY]  [THICKNESS=<n>]  [ANGLE=<n>]  [UNDERCUT=<n>]
[LEFT|RIGHT|ABOVE|BELOW]    [P1.X=<n>]    [P1.Y=<n>]    [P2.X=<n>]
[P2.Y=<n>]
[START|CONTINUE|DONE]  [X=<n>] [Y=<n>]
[INFILE=<c>]  [TOP.LAYER]  [NOEXPOSE]  [DIRECT.ANGLE=<n>]
[MACHINE=<c>]  [TIME=<n>]  [HOURS|MINUTES|SECONDS]
[DT.FACT=<n>]  [DT.MAX=<n>]  [DX.MULT=<n>]
[MC.REDEPO]  [MC.SMOOTH = <n>]  [MC.DT.FACT = <n>]  {MC.MODFNAME =
<c>]
```

Description

Athena provides two different etch simulation methods. The first is geometrical etching available within any Athena module. The second is physical etching available only in Elite.

Parameters used for Geometrical Etching

MATERIAL specify the material to be etched (see Section 6.2.9 “Standard and User-Defined Materials” for the list of materials). If a material is specified, only that material is etched even if other materials lie within the etch region. If no material is specified, all materials in the etch region are removed.

NAME.RESIST specify the type of photoresist to be etched.

ALL specifies that all of the specified materials are removed.

DRY indicates that the resulting surface will replicate the exposed surface and will simply be lowered by a fixed depth of THICKNESS microns below the exposed surface. If ANGLE or UNDERCUT or both is specified, the shape of DRY etched region is modified accordingly. TRAPEZOI is a synonym for this parameter.

THICKNESS specifies the thickness to be etched for the dry etch type. Units are microns.

ANGLE specifies sidewall slopes in degrees (90° corresponding to vertical slope is the default).

UNDERCUT specifies the distance in microns that ETCH extends under a mask when dry etch is performed (the default is 0).

LEFT, RIGHT, ABOVE, and **BELOW** provide a quick means of etching with a trapezoidal cross section. The etch region will be to the specified side (left/right/above/below) of the line specified by the coordinates given in P1.X, P1.Y and P2.X, P2.Y.

P1.X, P1.Y, P2.X, and **P2.Y** allow you to specify a line for left/right/above/below etching. The P1 parameters are always required if left/right/above/below are used. The P2 parameters are required when the etch angle is non-vertical. Units are microns.

START, CONTINUE, and **DONE** specify an arbitrarily complex region to be etched. You can combine several lines to specify the several points that make up the region. See the examples.

X and **Y** specify a point in the `start/continue/done` mode of etch region specification. Units are microns.

INFILE specifies that the etch profile will be taken from the filename specified by the `INFILE` parameter. The specified file must have the following format:

X_1 Y_1

X_2 Y_2

X_3 Y_3

. . .

X_n Y_n

This will etch the region enclosed by the boundary coordinates within the file. You can define any number of coordinates within the file. This command is often useful for inputting data from digitized experimental profiles or external programs. The closing line is automatically drawn from the final coordinate point to the initial point.

TOP.LAYER specifies that only the top layer of the etched material should be etched.

NOEXPOSE specifies that the new surface is not exposed for subsequent oxidation or deposition after geometrical etch. Use this parameter to remove a part of the structure from the bottom or side of simulation.

DIRECT.ANGLE specifies angle (in degrees) of a directional etch. This angle is measured from the vertical. Default is 0 degrees.

Parameters used only with physical etching in the Elite module

MACHINE specifies the name of the etch machine that is to be run.

TIME specifies the time the etch machine is to be run.

HOURS, MINUTES, and **SECONDS** specify the units of the `TIME` parameter.

Parameters used only with RIE, WET.ETCH, and PLASMA models

DT.FACT is used with Elite type etch calculations. By default, the movement of a string node is limited to less than or equal to one quarter of the median segment length. This is a good compromise between simulation speed and the danger of loop formation. The optimization factor `DT.FACT` must not exceed 0.5. You can, however, decrease it if necessary for more accuracy.

DT.MAX is used to limit timesteps size. By default, the upper limit for the maximum timestep is one tenth of the total etch time specified. This is a good compromise between calculation accuracy and calculation time. But sometimes, it is useful to adapt this value to the specific simulation problem. Allowing the time steps to become greater gives a higher simulation speed but the accuracy may suffer. For smaller time steps, the simulation speed will decrease but the accuracy may be greater.

DX.MULT is the accuracy multiplier for Elite etches. The discretization size used for the etch calculation will be multiplied by `DX.MULT`. For improved accuracy at the cost of extra simulation time, decrease the value of `DX.MULT`.

Parameters used only with MC.PLASMA model

MC.REDEPO specifies that redeposition of polymer should be simulated. Default is `true`.

MC.SMOOTH specifies level of smoothing of the surface.

MC.DT.FACT specifies time step control for Monte Carlo etching and redeposition.

MC.MODFNAME specifies name of the C-Interpreter file with user-defined Monte Carlo etching and redeposition models.

Simple Geometrical Etch Example

The following command etches all the nitride to the left of a vertical line located at x=0.5:

```
ETCH NITRIDE LEFT P1.X=0.5
```

Arbitrary Geometrical Shape Etch Example

The following set of commands etch the oxide in the square defined at (0,0), (1,0), (1,1), and (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
```

Be careful when using this style of syntax that the list of coordinates forms a regular polygon. The closing line from the last coordinate pair to the initial point is automatically added.

Anisotropic Geometrical Etch Example

The following command finds the exposed surface and lowers it straight down 0.1 microns. This line will be the new surface.

```
ETCH DRY THICK=0.1
```

Physical Etch Example

The following sequence defines an etch machine named `PLASMA1` that performs reactive ion etching of silicon. The machine is applied to etch the current structure for 10 minutes.

```
RATE.ETCH   MACHINE=PLASMA1   SILICON   U.M   RIE   ISOTROPIC=0.1
DIRECT=0.9

ETCH MACHINE=PLASMA1 TIME=10 MINUTES
```

Note: The program can be sensitive to grid placement. It often helps to prepare the initial grid by having a vertical grid line exactly at the etch coordinate for geometric etches.

For example, see `RATE.ETCH`.

6.20 EXPOSE

EXPOSE runs the exposure module of Optolith.

Note: This statement is not available in Athena 1D.

Syntax

```
EXPOSE
[INFILE=<c>] [PERPENDICUL|PARALLEL] [X.CROSS|Z.CROSS]
[CROSS.VALUE=<n>] [DOSE=<n>] [X.ORIGIN=<n>
[FLATNESS=<n>] [NUM.REFL=<n>]
[FRONT.REFL=<n>] [BACK.REFL=<n>] [ALL.MATS=<n>]
[MULT.EXPOSE] [POWER.MIN=<n>]
```

Description

This command defines the parameters associated with and performs two dimensional exposure.

INFILE is the name of an input file that contains a user aerial image cross section data file. This file has the form:

```
<wavelength in microns>
<number of data pairs>
<x location in structure>          <relative image intensity>
```

PERPENDICUL and **PARALLEL** specify TE mode or TM mode respectively. PERPENDICUL is the default.

X.CROSS and **Z.CROSS** specify that the cross section is parallel to the x-axis (z=constant) and parallel to the z-axis (x=constant), respectively. X.CROSS is the default.

CROSS.VALUE specifies the x or z coordinates of the cross section of the aerial image. The default will be centered in the image window. Units are microns.

DOSE specifies the exposure dose in mJ/cm^2.

X.ORIGIN locates the beam relative to the structure. This allows the aerial image to be shifted if necessary. Units are microns. The default is 0.0.

FLATNESS specifies the accuracy of the change in surface topography in degrees. A value of zero specifies that all grid points will be calculated. The default value is 0.25. In any case, maintain the limits 0 ≤FLATNESS ≤ 1.

NUM.REFL specifies the number of reflections to be considered.

FRONT.REFL specifies that front surface reflection should be considered in the calculation. The default is no front reflection.

BACK.REFL specifies the back surface reflection. The default is no back reflection.

ALL.MATS specifies that intensity be displayed in all materials. The default is photoresist only.

MULT.EXPOSE is used to make multiple exposures. MULT.EXPOSE is specified on the second EXPOSE command for addition of exposures. If MULT.EXPOSE is not specified, previous exposures will be erased.

POWER.MIN sets the minimum power accounted for in multiple reflections. POWER.MIN is used in a multiplicative format. In other words, if power attenuation due to 10 reflections is less than POWER.MIN it will not be counted for calculation.

Examples

The following statement loads a cross section of an aerial image that you can input. It then runs the exposure module. The number of reflections increases calculation time when it is set to a value greater than one.

```
EXPOSE INFILE=CROSS.SECT NUM.REFL=3
```

The following command runs the exposure module for the Z.CROSS section of a two dimensional aerial image that has been previously generated. The x value of the cross section is 0.1.

```
EXPOSE Z.CROSS CROSS.VAL=0.1
```

Note: The DEFOCUS parameter on the IMAGE statement must be used to do defocus exposure calculations. The image command must be used in conjunction with the EXPOSE command for a defocussed bulk image.

For more examples, see INITIALIZE and IMAGE.

6.21 EXTRACT

The EXTRACT command is used to analyze the current structure or a previously saved file. It can extract important parameters such as material thickness, junction depth and peak doping levels. It also includes electrical extractions such as sheet resistance, threshold voltage and CV curves.

Note: The EXTRACT statement is supported under DeckBuild and is fully documented in the DeckBuilD User's Manual.

6.22 FOREACH

FOREACH specifies the command looping facility.

Syntax

```
FOREACH(NAME)(LIST)

(COMMANDS)

END
```

Description

This command is used to specify input loops. FOR is equivalent to FOREACH. When the loop executes, NAME will consecutively take on each value in LIST and exit the loop after assuming the last value. COMMANDS will be executed once for each value in LIST. NAME is set to a value in LIST using the shell define function.

LIST is a set of strings separated by commas or spaces. The values in LIST can be delimited by either commas or spaces. LIST can also take the following numerical operator form:

```
START TO END STEP VAL
```

where START is a numerical start value, END is the last value, and VAL is the amount to increment at each iteration.

Examples

The following statement will increment val from 1.0 to 10.0 in steps of 0.5. This loop will be executed 19 times.

```
FOREACH VAL ( 1.0 TO 10.0 STEP 0.5 )

ECHO VAL

END
```

Note: Command line continuation using the backslash character “\” indicator is not supported in the FOREACH statement.

Note: FOREACH, ECHO, and END statements are supported only in the batch mode. DeckBuild provides much more powerful looping capabilities, which is recommended.

6.23 GO

GO starts the simulator. Each Athena input file should begin with a GO statement.

Note: The GO command is executed by DeckBuild and documented in the DeckBuild User's Manual.

Examples

Two useful features of the GO command are shown here.

This command initializes Athena with a specified version number

```
go athena simflags="-V 4.3.0.R"
```

This command initializes Athena with a model file `<install>/lib/athena/<version>/common/athenamod.97a`.

```
go athena simflags="-modfile 97a"
```

Note: If DeckBuild encounters a GO statement and there is no change in the version or model file, Athena will continue running.

6.24 HELP

`HELP` prints summary of statement names and parameters syntax.

```
HELP [<command>]
```

or

```
? [<command>]
```

Description

HELP lists the parameters of the specified statement and provides a short description of each. If there is no statement name given, `HELP` will show an introductory help message and will list all statements.

Examples

The following will print a list of valid Athena commands to the standard output.

```
HELP
```

The following will print a description of the `DIFFUSE` command and its parameters.

```
HELP DIFFUSE
```

6.25 ILLUM.FILTER

ILLUM.FILTER specifies the illumination source shape and illumination source filtering in Optolith.

Note: This statement is not available in Athena 1D.

Syntax

```
ILLUM.FILTER
[CIRCLE|SQUARE|GAUSSIAN|ANTIGAUSS|SHRINC]
[GAMMA=<n>] [RADIUS=<n>] [ANGLE=<n>] [SIGMA=<n>]
[IN.RADIUS=<n>] [OUT.RADIUS=<n>] [PHASE=<n>]
[TRANSMIT=<n>] [CLEAR.FIL]
```

Description

This statement specifies the illumination source options as well as illumination source filtering.

CIRCLE, SQUARE, GAUSSIAN, ANTIGAUSS, and **SHRINC** define or change the shape of the exit pupil of the illumination system. SHRINC can be used to define the illumination system only, not annular filters.

GAMMA defines or changes the GAMMA value for GAUSSIAN or ANTIGAUSS source transmittance. GAMMA is a parameter that defines the truncation of the GAUSSIAN by the pupil. In the limit of GAMMA→0, the source will be uniform.

RADIUS specifies the radius of a single source if you define the SHRINC illuminator concept. This parameter must be entered in fractions of unity.

ANGLE specifies the angular location for the SHRINC illuminator.

SIGMA defines or changes the filling factor for the combination of the illumination and projection systems. The value of SIGMA is expected to vary, but it will not be reset. Also, specifies the radius of a single source if you specify the SHRINC illuminator concept. This parameter must be entered in fractions of unity (assuming a unit pupil radius).

IN.RADIUS and **OUT.RADIUS** define or change the intensity transmittance and phase transmittance of an annular zone inside the exit pupil of the illumination system. This qualifier is used to simulate spatial filtering techniques. IN.RADIUS and OUT.RADIUS are used to define an annular zone in the exit pupil having the pupil transmittance equal to TRANSMIT and producing the phase angle equal to PHASE. Radius values are specified in fractions of unity, and phase is specified in degrees. Note that the annular zones should not overlap. The outer radius of an inner zone must be smaller than the inner radius of an outer zone.

PHASE specifies the phase shift in degrees produced by the illumination source filter ($-180^o \leq$ PHASE $\leq 180^o$).

TRANSMIT specifies the intensity transmittance produced by the illumination filter ($0 \leq$ TRANSMIT ≤ 1).

CLEAR.**FIL** resets the illumination source filter list.

Example

The following example defines a SHRINC illumination source, where the quadruple circular illumination sources are located at 45° to the x-axis with the center at a radius of 0.2 from the origin and a circle radius of SIGMA=0.2.

```
ILLUM.FILTER SHRINC RADIUS=.2 SIGMA=.2 ANGLE=45
```

6.26 ILLUMINATION

ILLUMINATION specifies the basic illumination parameters in Optolith.

Note: This statement is not available in Athena 1D.

Syntax

```
ILLUMINATION
[I.LINE|G.LINE|H.LINE|KRF.LASER|DUV.LINE|ARF.LASER|F2.LASER|
LAMBDA=<n>]
[X.TILT=<n>][Z.TILT=<n>]
[INTENSITY=<n>]
```

Description

I.LINE, **G.LINE**, **H.LINE**, **KRF.LASER** (alias **DUV.LINE**), **ARF.LASER**, and **F2.LASER** specify that the standard wavelengths of the illumination to be used. The corresponding wavelengths are 0.365, 0.436, 0.407, 0.268, 0.193, and 0.157 microns.

LAMBDA defines or changes the source wavelength. Only monochromatic sources are assumed for simulation, that is only one wavelength can be specified. The units are microns.

X.TILT and **Z.TILT** specify the tilt of the illumination system with respect to the optical axis of the projection system. All values are to be entered in degrees.

INTENSITY defines or changes the absolute value (usually set to one) of the complex amplitude, that is the intensity in the mask or reticle plane.

Examples

The following statement defines i.line illumination with X and Z tilt of 0.1° and an intensity of 1.

```
ILLUMINATION I.LINE X.TILT=0.1 Z.TILT=0.1 INTENSITY=1
```

For more examples, see IMAGE, PROJECTION, ILLUM.FILTER, PUPIL.FILTER, ABERRATION, LAYOUT.

6.27 IMAGE

IMAGE calculates a one or two dimensional aerial image.

Note: This statement is not available in Athena 1D.

Syntax

```
IMAGE

[INFILE=<c>] [DEMAG=<n>] [GAP=<n>]

[OPAQUE|CLEAR] [DEFOCUS=<n>] [CENTER]

[WIN.X.LOW=<n>]          [WIN.X.HIGH=<n>]          [WIN.Z.LOW=<n>]
[WIN.Z.HIGH=<n>]

[DX=<n>] [DZ=<n>] [X.POINTS=<n>] [Z.POINTS=<n>] [N.PUPIL=<n>]

[MULT.IMAGE] [X.CROSS|Z.CROSS] [ONE.DIM]
```

Description

This statement calculates a 2D aerial image and sets parameters that control the accuracy, input, and output of the imaging module. The IMAGE statement accepts layout information created by MaskViews.

INFILE is the name of the mask data file from MaskViews. It contains coordinates of rectangular mask features as well as the transmittance and phase of each feature. This file name usually ends with the extension.sec.

Note: For more information on the alternative method of loading MaskViews layout information for image calculations, see the LAYOUT statement.

DEMAG specifies demagnification factor. If specified, all elements of layout as well as all parameter of image window and grid will decrease.

GAP specifies the mask-to-wafer gap for the case of contact printing. The units are microns.

OPAQUE and **CLEAR** specify the type of mask to be used. The background will be opaque if you select OPAQUE, while the mask features will be clear. The background will be clear if you select CLEAR and the mask features will be opaque.

DEFOCUS is a user specified defocus parameter. If < 0, above the resist. If > 0, below the resist surface.

CENTER specfies that layout loaded using the INFILE parameter will be shifted so its center is in the point (0,0)-- the origin of coordinates for the computational window.

WIN.X.LOW, WIN.X.HIGH, WIN.Z.LOW, and **WIN.Z.HIGH** set the minimum and maximum x and z values that define the image window. If unspecified, default values from the mask file will be used. The units are microns.

DX specifies the mesh resolution for the image window in x. If DX is not specified, X.POINTS and Z.POINTS will be used. The units are microns.

DZ specifies the mesh resolution for the image window in z. The default is DZ=DX. The units are microns.

X.POINTS and **Z.POINTS** are the number of x and z coordinate points in the image window, respectively. These parameters are used only if DX is not specified. Default value is 10 for both coordinates.

N.PUPIL defines or changes the number of mesh points in the projector's exit pupil used in imaging simulations. The value of N.PUPIL sets the number of mesh points along the exit pupil's radius. Larger values provide better accuracy. The default setting should be adequate for accuracy. N.PUPIL also sets the size of the mask or image cell for imaging simulations. Finally, N.PUPIL affects the discretization of the source. This means that if a very fine source discretization is required, N.PUPIL should be set to a larger value. Note that computation time grows linearly with the number of pupil mesh points and source points used in the simulation.

MULT.IMAGE specifies that the preceding and current images will be added. You can add any number of images as long as the IMAGE command contains this boolean.

X.CROSS and **Z.CROSS** specify if the one dimensional image is parallel to the x-axis or z-axis respectively.

ONE.DIM use a one dimensional image module that images a line drawn across a 2D layout. This is the best method when the image will be used for subsequent EXPOSE statements.

Example

This statement loads a mask named MASK.SEC and specifies x resolution in the image window of DX=0.1 micrometers. It then runs the imaging module.

```
IMAGE INFILE=MASK.SEC DX=0.1
```

For more information, see ILLUMINATION, PROJECTION, ILLUM.FILTER, PUPIL.FILTER, ABERRATION, LAYOUT, and EXPOSE.

6.28 IMPLANT

IMPLANT specifies an ion implantation process step.

Syntax

```
IMPLANT
[GAUSS|PEARSON|FULL.LAT|MONTECARLO|BCA] [CRYSTAL|AMORPHOUS]
[PROFILE] [MEASURED.SIMS=<c> | EXTRACT.SIMS=<c>]
IMPURITY ENERGY=<n> DOSE=<n> [FULL.DOSE]
[TILT=<n>] [ROTATION=<n>]
[PLUS.ONE] [DAM.FACTOR=<n>] [DAM.MOD=<c>] [PRINT.MOM]
[X.DISCR=<n>] [LAT.RATIO1] [LAT.RATIO2] [S.OXIDE=<n>]
[MATCH.DOSE|RP.SCALE|MAX.SCALE] [SCALE.MOM] [ANY.PEARSON]
[IMPL.TAB=<c>] [FULLROTATION]
[N.ION=<n>] [MCSEED=<n>] [TEMPERATURE=<n>] [DIVERGENCE=<n>]
[IONBEAMWIDTH=<n>]
[IMPACT.POINT=<n>]     [SMOOTH=<n>]     [SAMPLING]     [DAMAGE]
[CRIT.ENERGY=<n>]
[MISCUT.TH] [MISCUT.PH]
[TRAJ.FILE=<n>] [N.TRAJ=<n>]
[Z1=<n>] [M1=<n>] [REFLECTIVE]
```

Description

This statement simulates ion implantation using different analytical and Monte Carlo models.

Model Selection Parameters

GAUSS, **PEARSON**, **FULL.LAT**, **MONTECARLO**, and **BCA** specify the implant model that is being used. GAUSS selects a Gaussian distribution. PEARSON selects the Pearson-IV distribution or where available dual Pearson-IV distributions. FULL.LAT is the same as PEARSON with lateral component of the 2D distribution calculated using all available moments instead of just a lateral standard deviation. MONTECARLO (synonym is BCA) activates the Monte Carlo Implant Module, which based on the Binary Collision Approximation.

CRYSTAL and **AMORPHOUS** specify whether or not the silicon lattice structure is to be taken into account during implant steps. The statements are mutually exclusive and CRYSTAL is true by default. For implants through thick screen materials, you often need to specify AMORPHOUS to avoid incorrect channeling profiles.

- For analytical implant models, these parameters select which set of tables are used for silicon implant ranges. The CRYSTAL model uses the SVDP tables where available and is the default.
- For MONTECARLO or BCA models these parameters control whether the crystalline lattice structure is considered or not.

PROFILE specifies that a profile data to be used in silicon instead of analytical model (GAUSS or PEARSON). The data might come either from measured SIMS profile or from "SIMS" profile extracted by the EXTRACT command of DeckBuild.

MEASUREAD.SIMS specifies name of the file that contains measured SIMS profile used in profile model indicated by the PROFILE statement. The file consist of two columns: depth in microns and concentration in ion/cm^3.

EXTRACT.SIMS specifies the name of the file that contains extracted SIMS profile.

Parameters Applicable for All Implant Models

IMPURITY specifies the impurity to be implanted (see Section 6.2.10 “Standard Impurities” for the list of impurities). **BF2** is also available.

ENERGY specifies the implant energy in keV.

DOSE specifies the dose of the implant. Dose is calculated in a plane normal to the implant direction. The units are in cm^{-2}.

FULL.DOSE specifies that the implanted dose is adjusted to compensate for the tilt angle. This type of dose specification is often used for high tilt implants.

```
   Adjusted Dose  = DOSE/cos(TILT)
```

TILT specifies the tilt with respect to the vertical of the implantation ion beam. The units are degrees. The default is 7°.

ROTATION specifies the angle of rotation of the implant relative to the plane of the simulation. The units are degrees.The default is 30°.

PLUS.ONE (synonyms are **UNIT.DAMAGE** and **D.PLUS**), and **DAM.FACTOR** (synonym is **D.SCALE**) specify the implant damage calculation. UNIT.DAMAGE specifies that the interstitial profile should be a scaled version of the doping profile from the implant. DAM.FACTOR specifies the scaling factor to be used for the UNIT.DAMAGE model. At a depth, the interstitial concentration from the UNIT.DAMAGE model will equal to the implanted ion concentration multiplied by DAM.FACTOR.

PRINT.MOM prints out moments for all ion/material combinations used in the analytical model. In the case of Monte Carlo simulation, it prints out moments calculated from the coordinates of ion in the standard structure file and can extracte them by the EXTRACT function.

DAM.MOD specifies the name of the C-Interpreter file, which can be used to modify defect concentration models.

Parameters Applicable Only for Analytical Implant Models

X.**DISCR** specifies the width of slices along the direction of the ion beam used to calculate the implanted profile. The value used is scaled relative to the lateral straggling of the current implant. By default, a slice width of between 0.1 and 0.2 of the average lateral straggle will be used. This parameter allows you to override the internal selection of discretization along the implant front. If the value of X.DISCR decreases, simulation accuracy and simulation time will increase.

LAT.RATIO1 specifies a factor by which all lateral standard deviations for the first Pearson distribution would be multiplied. Default is 1.0.

LAT.RATIO2 specifies a factor by which all lateral standard deviations for the second Pearson distribution would be multiplied. Default is `0.2`.

Note: The `LAT.RATIO*` parameters provide simple scaling of the default lateral standard deviation. Use the `MOMENTS` statement for more complete lateral standard deviation modifications.

S.OXIDE specifies screen oxide parameter for the SVDP implant model. Default is `0.001` microns. The screen oxide thickness is **not** determined from the structure and must be user-specified. See Section 3.5 "Ion Implantation Models" for more details and the on-line examples on how to set this parameter automatically.

MATCH.DOSE, RP.SCALE (synonym is **RP.EFF)**, and **MAX.SCALE** specify the method for implant calculations in multi-material structures (see Section 3.5.2 "Multi-Layer Implants"). Default is `MATCH.DOSE`.

SCALE.MOM specifies that moment scaling to be used with selected multilayer implant model.

ANY.PEARSON specifies no restrictions on the combinations of allowed skewness and kurtosis. This is true by default as required for the SVDP models. See Section 3.5 "Ion Implantation Models" for details on potential problems with this setting. Athena versions earlier than 4.0 had this parameter set to false by default.

IMPL.TAB specifies the table from the T-SUPREM4 default implant tables file.

FULLROTATION specifies that the implant be performed at all rotation angles.

Parameters Applicable Only for Monte Carlo/BCA Implant Models

N.ION specifies the number of ion trajectories to be calculated for the Monte Carlo method. When the `SAMPLING` is not specified, the default is `N.ION` is 1,000 for 1D structures and 10,000 for 2D structures.

MCSEED specifies a seed for the random number generator used for the Monte Carlo calculation.

TEMPERATURE specifies the temperature of the substrate during implantation.

DIVERGENCE specifies the implant beamwidth in degrees. When the `BEAMWIDTH` angle is specified the `TILT` angle is varied between `TILT +/- DIVERGENCE/2.0`. Each ion will have an angle somewhere in this range decided by a random number generator. Distribution of the ions is uniform across the defined angular distribution. Correct specification of `DIVERGENCE` is generally required for accurate zero degree implant ranges. Default is 1 degree.

IMPACT.POINT specifies (only in the Monte Carlo method) that the ion beam enters the surface in the point with lateral coordinate `x=(left+IMPACT.POINT*L)`, where `left` is the x-coordinate of the left boundary of the structure and `L` is the length of the structure. This parameter would be used for calculation of the point-source 2D distribution and spacial moments using Monte Carlo method.

IONBEAMWIDTH specifies the ion beam width in nanometers. It can only be used with the `IMPACT.POINT` parameter.

SMOOTH specifies that a special Gaussian convolution smoothing to be applied to the Monte Carlo results. SMOOTH multiplied by estimated standard deviation of the whole profile serves as the standard deviation for the Gaussian formula.

SAMPLING specifies that statistical sampling to be used in the Monte Carlo method.

Note: When you specify SAMPLING parameter, the multi-threading will be switched off and all ion trajectories will be simulated on one CPU. In many cases, the statistical sampling method can be more accurate than running larger amount of trajectories using multiple CPUs.

DAMAGE specifies that damage (in displacements/cm^3) is calculated and saved into the structure file during Monte Carlo implant simulation. This damage distribution is served only for estimation of depth and lateral spread of amorphization area.

CRIT.ENERGY specifies critical deposition energy for amorphization. If the value of this parameter is less than 1000, then it is used as a scaling factor for calibrated default value. Otherwise, it will be user-specified value in units of eV/cm^3.

MISCUT.TH and **MISCUT.PH** specify the wafer's miscut. See “Implantation Geometry in SiC” on 216 for more information.

TRAJ.FILE specifies the name of the file in which ion trajectories calculated with the Monte Carlo (BCA) method are to be saved.

Note: This parameter switches off statistical sampling if it’s specified by the SAMPLING parameter.

Note: When TRAJ.FILE parameter is specified, the Monte Carlo implant simulation will be performed on a single processor, even if the multiple processors are available and parallel capability is specified by the -P parameter in the GO ATHENA statement.

N.TRAJ specifies the number of ion trajectories to be saved in the TRAJ.FILE. The default is minimum of N.ION and 2000.

Z1 specifies the atomic number of an "inert" ion used only for damage or amorphization of substrate. No new impurity will be introduced into the structure except "damage", which will affect subsequent Monte Carlo implants.

M1 specifies the atomic weight of the "inert" ion with atomic weight of Z1. If M2 is not specified, the atomic weight of the main isotope will be used.

REFLECTIVE specifies that reflective boundary conditions will be used at the left and right sides of the structure. The reflective boundary conditions are applicable for either TILT=0 or for tilted implant with FULLROTATION. The reflective boundary conditions are also applicable with non-zero ion beam DIVERGENCE.

Analytical Implant Example

This example specifies that a 100keV implant of phosphorus to be done with a dose of 1.0 e14 and with a tilt angle of 15° to the surface normal. The Pearson model is to be used to calculate the doping profile.

```
IMPLANT PHOSPH DOSE=1E14 ENERGY=100 TILT=15
```

SVDP Boron Implant Example

This example shows the syntax for a zero tilt and 50keV boron implant through 5nm of screen oxide. The oxide is defined by S.OXIDE and this definition is independent of any actual oxide in the structure itself.

```
IMPLANT BORON DOSE=1E13 ENERGY=50 TILT=0 S.OXIDE=0.005
```

Monte Carlo Implant Example

This example specifies a 300keV boron implant at zero degrees tilt and rotation. Accurate modeling of such implants is only possible in the BCA model. Since ion channeling is highly dependent on the tilt angle.

```
IMPLANT BORON DOSE=1E13 ENERGY=300 BCA TILT=0 ROTATION=0
```

Implant Damage Example

This example implants phosphorus and invokes the unit damage model. The UNIT.DAMAGE model creates an interstitial profile scaled to the implant doping profile. DAM.FACTOR is used here to specify that the interstitial concentration will be ten times less than the doping throughout the depth of the implant profile.

```
IMPLANT PHOSPHORUS DOSE=1E14 ENERGY=50 UNIT.DAMAGE DAM.FACTOR=0.1
```

SiC Implant Examples

Example 1

This example demonstrates equivalent implant directions with wafer’s major flat, <1$\bar{1}$00>.

```
init sic_4h rot.sub=0
```

```
implant aluminum n.ion=40000 dose=0.63e13 tilt=0 rot=0 energy=60 bca
```

The equivalent implant direction with miscut wafer 4° <11$\bar{2}$0> toward will be

```
init sic_4h rot.sub=0
```

```
implant aluminum n.ion=40000 dose=0.63e13 tilt=4 rot=90 miscut.th=4 miscut.ph=90 energy=60 bca
```

The two profiles will be statistically equivalent with slight shift due to cos(4°) projection of the second profile.

Example 2

This example demonstrates equivalent implant directions with wafer’s major flat, <11$\bar{2}$0>. MISCUT.TH/PH are measured from the internal crystallographic coordinate system, not the laboratory one defined by ROT.SUB.

```
init sic_4h rot.sub=90
```

```
implant aluminum n.ion=40000 dose=0.63e13 tilt=0 rot=0 energy=60 bca
```

The equivalent implant direction with miscut wafer 4° toward ?1120? will be

```
init sic_4h rot.sub=90
```

```
implant  aluminum  n.ion=40000  dose=0.63e13  tilt=4  rot=0  mis-
cut.th=4 miscut.ph=90 energy=60 bca
```

The two profiles will be statistically equivalent with slight shift due to cos(4°) projection of the second profile.

Example 3

The following example demonstrates equivalent implant conditions for 3° tilted off [0001] direction toward <11$\bar{2}$0> implant for ideal, and for miscut 4° toward <11$\bar{2}$0> 4H-SiC wafer.

```
init sic_4h rot.sub=0
```

```
implant aluminum n.ion=40000 dose=0.63e13 tilt=3 rot=0 energy=60
bca
```

The equivalent implant direction with miscut wafer 4° toward <11$\bar{2}$0> will be

```
init sic_4h rot.sub=90
```

```
implant  aluminum  n.ion=40000  dose=0.63e13  tilt=1  rot=0  mis-
cut.th=4 miscut.ph=90 energy=60 bca
```

The two profiles will be statistically equivalent with slight shift due to both profiles projection on the vertical, Y, coordinate.

6.29 IMPURITY

`IMPURITY` specifies impurity parameters.

Note: This statement supersedes the older syntax using separate statements for each impurity type. The `ARSENIC`, `ANTIMONY`, `BORON`, `INDIUM` and `PHOSPHORUS` statements should no longer be used.

Syntax

```
IMPURITY
I.IMPURITY [DONOR|ACCEPTOR|NEUTRAL] MATERIAL
[AT.NUMBER=<n>][AT.MASS=<n>]
[DIX.0=<n>] [DIX.E=<n>] [DIP.0=<n>] [DIP.E=<n>]
[DIPP.0=<n>] [DIPP.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>]
[DVP.0=<n>] [DVP.E=<n>] [DVPP.0=<n>] [DVPP.E=<n>]
[SOL.SOLUB|CLUSTER.ACT] [PROP.ACTIV=<n>]
[CTN.0=<n>] [CTN.E=<n>] [CTP.0=<n>][CTP.E=<n>]
[SS.CLEAR][SS.TEMP=<n>][SS.CONC=<n>]
[ACT.FACTOR=<n>] [TRACT.0=<n>] [TRACT.E=<n>] [TRACT.MIN=<n>]
[/MATERIAL] [SEG.0=<n>][SEG.E=<n>][TRN.0=<n>][TRN.E=<n>]
[TRNDL.0=<n>][TRNDL.E=<n>]
[PD.DIX.0=<<n>>] [PD.DIX.E=<n>] [PD.EFACT=<n>] [PD.SEG.E=<n>]
[PD.TAU=<n>][PD.SEGSITES=<n>]                   [PD.GROWTH.0=<n>]
[PD.GROWTH.E=<n>]
[PD.CRATIO=<n>][PD.SEG.GBSI=<n>]

[AXX.STRAIN] [BXX.STRAIN] [CXX.STRAIN] [AYY.STRAIN] [BYY.STRAIN]
[CYY.STRAIN] [E.STRAIN][A.YOUNG]
```

Description

This statement allows to specify coefficients of impurity diffusion, transport, segregation, and so on.

Generic Parameters

I.IMPURITY is the name of impurity which parameters to be specified (see Section 6.2.10 “Standard Impurities” for the list of impurities).

DONOR, ACCEPTOR and **NEUTRAL** specify the type of the impurity in the given material. Default is `NEUTRAL`.

MATERIAL specify the material in which the impurity parameters apply as well as MATERIAL1 for the segregation and transport parameters on the boundary between two materials (see Section 6.2.9 “Standard and User-Defined Materials” for the list of materials).

AT.NUMBER and **AT.MASS** specify the atomic number and atomic mass of the impurity respectively. This parameters are used in Monte Carlo/BCA implant calculations.

Diffusion Parameters

The units for all pre-exponential diffusion constants are cm^2/sec, while the units for activation energies are eV.

DIX.0 and **DIX.E** specify the diffusion coefficient for the impurity diffusing with neutral interstitials. DIX.0 is the pre-exponential constant and DIX.E is the activation energy.

DIP.0 and **DIP.E** specify the diffusion coefficient for the impurity diffusing with single positive interstitials. DIP.0 is the pre-exponential constant. DIP.E is the activation energy.

DIPP.0 and **DIPP.E** specify the diffusion coefficients for the impurity diffusing with double positive interstitials.

DIM.0 and **DIM.E** specify the diffusion coefficient for the impurity diffusing with single negative interstitials. DIM.0 is the pre-exponential constant. DIM.E is the activation.

DIMM.0 and **DIMM.E** specify the impurity diffusing with doubly negative interstitials. DIMM.0 is the pre-exponential constant. DIMM.E is the activation energy.

DVX.0 and **DVX.E** specify the impurity diffusing with neutral vacancies. DVX.0 is the pre-exponential constant. DVX.E is the activation energy.

DVM.0 and **DVM.E** specify the impurity diffusing with single-negative vacancies. DVM.0 is the pre-exponential constant. DVM.E is the activation energy.

DVMM.0 and **DVMM.E** specify the impurity diffusing with double-negative vacancies. DVMM.0 is the pre-exponential constant. DVMM.E is the activation energy.

DVP.0 and **DVP.E** specify the impurity diffusing with single-positive vacancies. DVP.0 is the pre-exponential constant. DVP.E is the activation energy.

DVPP.0 and **DVPP.E** specify the impurity diffusing with double-positive vacancies. DVPP.0 is the pre-exponential constant. DVPP.E is the activation energy.

FI.0 and **FI.E** are the fractional interstitialcy parameters that determine whether the impurity diffuses through interaction with interstitials or vacancies. Once the expression for total FI is evaluated from these coefficients, the value of total FI can vary between 0 and 1. FI equal to 1 corresponds to a pure interstitial-based diffusion, while value of 0 corresponds to a pure vacancy mechanism.

Activation Model Parameters

SOL.SOLUB specifies that solid solubility model and solid solubility tables will be used for calculation of active concentration of the specified impurity in the specified material. This is default for all cases except Arsenic in Si and Poly.

CLUSTER.ACT specifies that cluster activation will be used. It is default only for AS in Si and Poly.

SS.CLEAR, **SS.TEMP**, and **SS.CONC** are the parameters for solid solubility data. SS.CLEAR clears the currently stored solid solubility data for the specified impurity in the specified material. SS.TEMP and SS.CONC add a single temperature and an associated solid solubility concentration point to those already stored for the impurity. Units for SS.TEMP are ^{o}C. Units for SS.CONC are cm^{-3}.

CTN.0 and **CTN.E** specify the vacancy clustering coefficients for the impurity. CTN.0 is the pre-exponential coefficient and CTN.E is the activation energy. By default, these parameters are only used for Arsenic.

CTP.0 and **CTP.E** specify the vacancy clustering coefficients.

ACT.FACTOR specifies parameter for concentration dependent solid solubility activation model. The value of ACT.FACTOR must be between 0.8 and 1.0.

PROP.ACTIVE specifies coefficient of the proportional activation model. If this parameter is specified, then in every point of specified material, the active concentration of the specified impurity will be equal to the chemical concentration multiplied by the value of this parameter. This model is mainly used for silicon carbides. The value must be more or equal to 0.0 and less or equal to 1.0.

TRACT.0, **TRACT.E**, and **TRACT.MIN** specify parameters of the transient activation model. Units for TRACT.0 and TRAT.MIN are seconds. Units for TRACT.E are eV.

Interface Transport Parameters

/MATERIAL specify MATERIAL2 for the segregation and transport parameters on the boundary between two materials (see Section 6.2.9 “Standard and User-Defined Materials” for the list of materials).

SEG.0 and **SEG.E** allow the computation of the equilibrium segregation concentrations. SEG.0 is the unitless pre-exponential constant. SEG.E is the activation energy in eV.

TRN.0 and **TRN.E** allow the specification of the transport velocity across the interface given. TRN.0 is the pre-exponential constant (units are cm/sec). TRN.E is the activation energy (units are eV).

TRNDL.0 and **TRNDL.E** specify parameters of the Interface Trap Model, describing dose loss at silicon/oxide interface. Units for TRNDL.0 are cm/sec; units for TRNDL.E are eV.

Polysilicon Diffusion Model Parameters

PD.DIX.0 and **PD.DIX.E** specify impurity diffusivity along grain boundaries. PD.DIX.0 is the pre-exponential factor of grain boundary diffusivity (units are cm^{2}/sec). PD.DIX.E is the activation energy for grain boundary diffusivity (units are eV).

PD.EFACT specifies entropy factor of grain boundary segregation coefficient.

PD.SEG.E specifies the activation energy of grain boundary segregation coefficient. Units are eV.

PD.TAU specifies the grain boundary time constant. Units are seconds.

PD.SEGSITES specifies density of segregation sites at grain boundary. Units are sites/cm^{2}.

PD.GROWTH.0 specifies the grain growth rate pre-exponential coefficient. Units are eV*cm^{2}/sec.

PD.GROWTH.E specifies the grain growth rate activation energy. Units are eV.

PD.CRATIO specifies initial ratio between impurity concentration in grain boundaries and total concentration.

PD.SEG.GBSI specifies the factor which controls segregation between polysilicon grain boundaries and Silicon.

Parameters Related to Stress Models

AXX.STRAIN specifies the constant term of the doping dependent XX component of intrinsic strain.

BXX.STRAIN specifies the linear term of the doping dependent XX component of intrinsic strain.

CXX.STRAIN specifies the quadratic term of the doping dependent XX component of intrinsic strain.

AYY.STRAIN specifies the constant term of the doping dependent YY component of intrinsic strain.

BYY.STRAIN specifies the linear term of the doping dependent YY component of intrinsic strain.

CYY.STRAIN specifies the quadratic term of the doping dependent YY component of intrinsic strain.

E.STRAIN specifies activation energy per volume change ratio for the stress-dependent diffusion model.

A.YOUNG specifies the linear coefficient of doping dependent Young's modulus. The units are dyne/cm^2.

Examples

The following statement changes the neutral interstitial diffusivity component of phosphorous in silicon.

```
IMPURITY I.PHOSPHORUS SILICON DIX.0=3.85 DIX.E=3.85
```

The following statement changes the segregation parameters at the silicon-silicon dioxide interface. The concentration of phosphorous in silicon will be 30 times the concentration of phosphorous in oxide at equilibrium.

```
IMPURITY I.PHOSPHORUS SILICON /OXIDE SEG.0=30.0 SEG.E=0.0
```

The following syntax sets the temperature dependent impurity activation of Indium in Silicon.

```
IMPURITY I.INDIUM SILICON SS.TEMP=800 SS.CONC=<VAL1> SS.CLEAR
IMPURITY I.INDIUM SILICON SS.TEMP=900 SS.CONC=<VAL2>
IMPURITY I.INDIUM SILICON SS.TEMP=950 SS.CONC=<VAL3>
```

Note: The transport and segregation coefficients, `TRN.0`, `TRN.E`, `SEG.0`, and `SEG.E`, are known to be inaccurate for some values of concentration, material combinations, and temperature ranges. If the simulation is inaccurate, consider these coefficients for calibration.

For more examples, see `DIFFUSE`, `METHOD`, `INTERSITITIAL`, and `VACANCY`.

6.30 INITIALIZE

INITIALIZE specifies the initial starting material and background doping levels. LOADFILE is a synonym for this statement.

Syntax

```
INITIALIZE

[MATERIAL]     [ORIENTATION=<n>]     [ROT.SUB=<n>]     [X.COMP=<n>]
[Y.COMP=<n>]

[C.FRACTION=<n>] [C.IMPURITIES=<n>|RESISTIV-
ITY=<n>] [C.INTERST=<n>]

[C.VACANCY=<n>]   [BORON|PHOSPHORUS|ARSENIC|ANTIMONY]   [NO.IMPU-
RITY]

[ONE.D|TWO.D|AUTO]  [X.LOCAT=<n>]  [CYLINDRICAL]

[INFILE=<c>]  [STRUCTURE|INTENSITY]

[SPACE.MULT=<n>]    [INTERVAL.R=<n>]    [LINE.DATA]    [SCALE=<n>]
[FLIP.Y]

[DEPTH.STR=<n>]  [WIDTH.STR=<n>]
```

Description

This command sets up the mesh from either a rectangular specification or from a previous structure file. The statement also initializes the background doping concentration in all regions.

Material Related Parameters

MATERIAL specifies the material to be initialized (see Section 6.2.9 “Standard and User-Defined Materials” for the list of materials).

ORIENTATION specifies the substrate orientation. Only 100, 110 and 111 are recognized. The default is 100.

ROT.SUB specifies the major flat of the silicon substrate. It is measured in degrees from the external x-axis of the crystallographic coordinate system. By default, ROT.SUB=-45, i.e., it represents the [104] plane. This parameter is used only in BCA implantation module.

X.COMP specifies the composition fraction of the cation listed first in ternary or quaternary compound semiconductor to be deposited. For example, Al is usually first cation of AlGaAs. The full list of compound semiconductors that can be specified in Athena and the rules of their naming can be found in Appendix B of Atlas User’s Manual. For backward compatibility, C.FRACTION is the alias for X.COMP.

Y.COMP specifies the composition fraction of the anion listed first in quaternary compound semiconductor to be deposited. For example, As is first anion in InGaAsP. The full list of compound semiconductors that can be specified in Athena and the rules of their naming can be found in Appendix B of Atlas User’s Manual.

Dopant Related Parameters

C.IMPURITIES specify the uniform impurity concentration in substrate material (see Section 6.2.10 “Standard Impurities” for the list of impurities). Multiple parameters can be used to define compensated doping in the substrate material.

RESISTIVITY specifies the resistivity of the initial substrate material. If RESISITVITY is specified, the impurity concentration specified by C.*** parameter will be ignored and calculated from the resistivity vs. concentration tables, which are available only for boron, phosphorus, arsenic, and antimony. The units are Ω-cm.

BORON, PHOSPHORUS, ARSENIC, ANTIMONY specify the type of impurity when initial doping is defined by RESISTIVITY parameter.

Note: There are several additional methods to specify impurity concentration in substrate material. These methods are introduced to provide TSUPREM4 compatibility and described in Section C.7 “Changes in the INITIALIZE statement”.

C.INTERST specifies the uniform interstitial concentration in substrate material. Units are cm^{-3}.

C.VACANCY specifies the uniform vacancy concentration in substrate material. Units are cm^{-3}.

NO.IMPURITY specifies that the calculation be performed without impurities. No impurities will be introduced during the simulation. This speeds calculation and allows quick analysis of oxidation, deposit, and etch results.

Parameters Related to Dimensionality of Simulation

ONE.D, **TWO.D, AUTO** set whether the run will be in 1D, 2D, or the dimensionality automatically determined from the process flow. ONE.D specifies that calculation will be performed in 1D at a location indicated by the X.LOCAT parameter. TWO.D specifies that all process steps will be done in a full two dimensional calculation. If name of these parameters are specified or AUTO is used, Athena will perform 1D calculation until a two-dimensional calculation is required. This typically happens at the first ETCH statement, which doesn’t remove material across the whole width of the structure.

X.LOCAT specifies the position within the defined 2D mesh for performing 1D simulation.

CYLINDRICAL specifies the boundary conditions for cylindrically symmetrical structure. In this case, the axis of rotation is X=0.0 and no negative x coordinates are allowed.

Parameters Related to Initialization from a File

INFILE specifies a file name for reading. This file must contain a previously saved structure or intensity distribution (see Section 6.63 “STRUCTURE”). IN.FILE is a synonym for this parameter.

STRUCTURE and **INTENSITY** specify which type of file is to be initialized. STRUCTURE is the default.

Grid and Structure Related Parameters

SPACE.MULT specifies a global spacing multiplier to be applied to the spacings defined on the previously specified LINE statements.

INTERVAL.**R** is the maximum ratio between the distances of adjoining mesh lines. The default is 1.5. RATIO is a synonym for this parameter.

LINE.**DATA** specifies that locations of mesh lines be printed during execution.

SCALE allows an incoming mesh to be scaled. The default is 1.0.

FLIP.Y is a Boolean parameter that dictates the mesh should be mirrored about the x axis.

DEPTH.**STR** and **WIDTH**.**STR** specify the depth and width of the initial substrate structure dimension for use with the Process Adaptive Meshing algorithm. Units are microns.

Example Starting from a file

The following statement reads in a previously saved structure from the TEST.STR file.

```
INITIALIZE INFILE=TEST.STR
```

Example Using an GaAs Substrate

The following statement creates GaAs substrate doped with Selenium concentration of 1×10^{15}cm^{-3}.

```
INITIALIZE GAAS C.SELENIUM=1E15
```

For more examples, see BOUNDARY, LINE, REGION, STRUCTURE, and BASE.MESH.

6.31 INTERSTITIAL and VACANCY

`INTERSTITIAL` specifies coefficients of interstitial diffusion, recombination, and generation.

`VACANCY` specifies coefficients of vacancy diffusion, recombination, and generation.

Note: These two statements are almost equivalent. Most parameters that exist in the `INTERSTITIAL` statement are also on the `VACANCY` statement.

```
INTERSTITIAL|VACANCY
[MATERIAL] [D.0=<n>] [D.E=<n>] [CSTAR.0=<n>] [CSTAR.E=<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>]
[KR.0=<n>] [KR.E=<n>] [IVFACTOR=<n>] [IIFACTOR=<n>]
[KTRAP.0=<n>] [KTRAP.E=<n>] [DAMALPHA=<n>]
[/MATERIAL=] [TIME.INJ] [GROWTH.INJ] [RECOMB]
[KSURF.0=<n>] [KSURF.E=<n>] [KRAT.0=<n>] [KRAT.E=<n>]
[KPOW.0=<n>] [KPOW.E=<n>]
[VMOLE=<n>] [THETA.O=<n>] [THETA.E=<n>] [GPOW.0=<n> [GPOW.E=<n>]
[WET02|DRY02] [REC.STR=<n>] [INJ.STR=<n>]
[A.0=<n>] [A.E=<n>]                           [T0.0=<n>] [T0.E=<n>]
[TPOW.0=<n>] [TPOW.E=<n>]
[DCARBON.E=<n>] [KCARBON.0=<n>] [KCARBON.E=<n>]
```

Description

These two equivalent commands specify transport and generation/recombination coefficients for interstitials and vacancies. The statements allow you to specify coefficients for any material, though it is only practical for semiconductors. Athena has measured (or calibrated) default values only for silicon and some interfaces with silicon.

MATERIAL specify the material for which the interstitial (or vacancy) parameters apply as well as `MATERIAL1` for the segregation and transport parameters on the boundary between two materials (see Section 6.2.9 “Standard and User-Defined Materials” for the list of materials). Default is `Silicon`.

Defect Diffusion Parameters

D.0 and **D.E** specify the interstitial (or vacancy) diffusion coefficient. `D.0` is the pre-exponential constant (the units are cm^2 /sec), and `D.E` is the activation energy (the units are eV).

CSTAR.0 and **CSTAR.E** specify of the total equilibrium concentration of interstitials (or vacancies) in intrinsically-doped conditions. `CSTAR.0` is the pre-exponential constant (the units are cm^{-3}), and `CSTAR.E` is the activation energy (the units are eV).

NEU.0, NEU.E, NEG.0, NEG.E, DNEG.0, DNEG.E, POS.0, POS.E, DPOS.0 and **DPOS.E** specify the relative concentration of interstitials (or vacancies) in the various charge states (neutral, negative, double negative, positive, double positive) under intrinsic doping conditions. All `*.0` parameters are unitless. All `*.E` parameters are in eV.

Bulk Defect Recombination Parameters

KR.0 and **KR.E** specify the interstitial (or vacancy) bulk recombination rate. `KR.0` is the pre-exponential constant (the units are cm^{-3}sec^{-1}), and `KR.E` is the activation energy in eV.

IVFACTOR and **IIFACTOR** specify I/V Bimolecular recombination ratios in `HIGH.CONC` model. These parameters are valid only for the `INTERSTITIAL` statement.

KTRAP.0 and **KTRAP.E** specify the interstitial trap reaction rate. `KTRAP.0` is the pre-exponential constant (the units are cm^{-3}sec^{-1}), and `KTRAP.E` is the activation energy in eV.

Note: At present, it is very difficult to extract exact values for these parameters. The default values assume the trap reaction is limited by the interstitial concentration.

DAMALPHA specifies the interstitial recombination rate in the dislocation loop region. The units are sec^{-1}.

Interface Defect Generation and Recombination Parameters

/MATERIAL specify `MATERIAL2` for setting generation and recombination parameters on the boundary between two materials (see Section 6.2.9 “Standard and User-Defined Materials” for the list of materials).

TIME.INJ, GROWTH.INJ and **RECOMB** specify the type of reactions occurring at the specified interface. The `TIME.INJ` parameter means that a time dependent injection model should be chosen. The `GROWTH.INJ` parameter ties the injection to the interface growth velocity. The `RECOMB` parameter indicates a finite surface recombination velocity

KSURF.0, KSURF.E, KRAT.0, KRAT.E, KPOW.0 and **KPOW.E** specify the interstitial (or vacancy) surface recombination rate. `KSURF.0` is the pre-exponential constant for surface recombination rate under inert conditions (the units are cm/sec). `KSURF` is corresponding activation energy in eV. `KRAT.0` is the pre-exponential constant for the growth rate dependent component of the surface recombination rate (unitless). `KRAT.E` is the corresponding activation energy in eV. `KPOW.0` is the pre-exponential constant of the power parameter in the surface recombination rate formula (unitless), and `KPOW.E` is the corresponding activation energy in eV.

VMOLE, THETA.0, THETA.E, GPOW.0 and **GPOW.E** specify interstitial (or vacancy) generation parameters of the growth dependent generation model. `VMOLE` is the lattice density of the consumed material (the units are cm^{-3}). `THETA.0` specifies the pre-exponential constant for the fraction of consumed atoms injected as interstitial (or vacancy). `THETA.E` specifies the activation energy for the fraction (the units are eV). `GPOW.0` and `GPOW.E` specify pre-exponential constant and activation energy of the power parameter of the growth injection formula.

WETO2, DRYO2 specify whether the parameters `THETA.0` and `THETA.E` are for wet oxidation or dry oxidation. The default is `DRYO2`.

REC.STR and **INJ.STR** allow you to specify experimental models for interstitial (or vacancy) recombination or injection at interfaces. Three macros are defined for use: T is the time in seconds, and X and Y is the coordinates. If these are specified, they are used in place of any default model.

A.0, A.E, T0.0, T0.E, TPOW.0 and **TPOW.E** specify parameters for time-dependent injection model `TIME.INJ`. `A.0` is the pre-exponential constant for the injection rate and `A.E` is the corresponding activation energy. `T.0` and `T.E` are the pre-exponential constant and activation energy for the time constant in the time-dependent injection formula. `TPOW.0` and `TPOW.E` are the pre-exponential constant and activation energy for the power constant in the time-dependent injection formula.

Parameters for Carbon Effects in SiGeC

KCARBON.0 and **KCARBON.E** specify interstitial recombination rate in carbon sink. `KCARBON.0` is the pre-exponential constant for the rate in sec^{-1}, and `KCARBON.E` is the corresponding activation energy in eV.

DCARBON.E specifies the coefficient of interstitial diffusion retardation in SiGe in presence of carbon impurity. The units are eV.

Basic Example

The following statement specifies the silicon diffusion and equilibrium values for interstitials.

```
INTERST SILICON DI.0=5.0E-7 D.E=0.0 CSTAR.0=1.0E13 CSTAR.E=0.0
```

Defect Injection during Oxidation Example

The following statement specifies the oxide-silicon interface injection for DRYO2 ambient is to be computed using the oxide growth velocity and with 1% of consumed silicon injected as interstitials.

```
INTERST SILICON /OXIDE GROWTH VMOLE=5.0E22 \

          THETA.0=0.01 THETA.E=0.0
```

Surface Recombination Example

The following statement specifies the surface recombination velocity at the nitride silicon interface is 3.5×10^{-3} cm/sec.

```
INTERST SILICON /NITRIDE KSURF.0=3.5E-3 KSURF.E=0.0 KRAT.0=0.0
```

Experimental Injection Model Example

The following statement describes an injection at the silicon oxide interface that exponentially decays in time.

```
INTERST SILICON /OXIDE INJ.STR = (10.0E4*EXP( T / 10.0 ))
```

General Comments

The equivalence of `INTERSTITIAL` and `VACANCY` statement syntax is done regardless of physical meaning. For example, you can define vacancy injection during oxidation although default parameters are zero.

The models used here are involved in ongoing research. Many of the parameters have unknown dependencies, such as stress, temperature, starting silicon material, and stacking fault density.

For more examples, see `DIFFUSE` and `TRAP`.

6.32 LAYOUT

`LAYOUT` describes manual input of mask features for Optolith.

Syntax

```
LAYOUT

[LAY.CLEAR=<n>]

[X.LOW=<n>] [Z.LOW=<n>] [X.HIGH=<n>] [Z.HIGH=<n>] [X.TRI=<n>]

[Z.TRI=<n>]

[HEIGHT=<n>] [WIDTH=<n>] [ROT.ANGLE=<n>] [X.CIRCLE=<n>] [Z.CIR-
CLE=<n>]

[RADIUS=<n>] [RINGWIDTH=<n>] [MULTIRING] [PHASE=<n>] [TRANSMIT=<n>]

[INFILE=<c>] [MASK=<c>] [SHIFT.MASK=<c>]
```

Description

This command is used to enter mask information for Optolith. Several `LAYOUT` statements can be used in sequence to define complete mask patterns. All coordinate and size parameters are in microns.

LAY.CLEAR specifies that the currently defined layout should be deleted prior to the execution of the new layout definition.

X.LOW specifies the minimum x coordinate of the rectangular feature.

Z.LOW specifies the minimum z coordinate of the rectangular feature.

X.HIGH specifies the maximum x coordinate of the rectangular feature.

Z.HIGH specifies the maximum z coordinate of the rectangular feature.

X.TRI specifies the x coordinate of the right angle corner of the triangular feature.

Z.TRI specifies the z coordinate of the right angle corner of the triangular feature.

HEIGHT specifies the height of the right angle triangle feature.

WIDTH specifies the base width of the right angle triangle feature.

ROT.ANGLE specifies the angle of rotation of the feature (-180°≤`ROT.ANGLE`≤180°) with respect to the x-axis. The default value is 0°. The center of rotation is at the center of the rectangle and at the right angle corner of the triangle, respectively.

Note: You can only use the `X.TRI`, `Z.TRI`, `HEIGHT`, `WIDTH`, and `ROT.ANGLE` parameters for projection printing model.

X.CIRCLE specifies the x coordinate of the center of the circular or ring feature.

Z.CIRCLE specifies the z coordinate of the center of the circular or ring feature.

RADIUS specifies the radius of the circular feature or the outermost radius of the ring or multiring feature.

RINGWIDTH specifies the width of ring(s) of the mask feature shaped as a single ring or a series of concentrated rings.

MULTIRING specifies that mask feature consists from series of concentered rings with ring widths and distances between rings specified by the RINGWIDTH parameter. The number N of the rings in the feature should satisfy the following relation 2N * RINGWIDTH<RADIUS.

Note: You can only use the RINGWIDTH and MULTIRING parameters for proximity printing model specified by the GAP parameter in the IMAGE statement.

PHASE specifies the phase shift produced by the feature (-180°≤PHASE≤180°). The default value is 0°.

TRANSMIT specifies the intensity transmittance of the feature (0≤TRANSMIT≤1). The default value is unity.

INFILE specifies the name of the MaskViews layout file. This file contains the mask information. The MaskViews mask information is a set of polygons for each mask layer with attributes (transmittance and phase shift) for each polygon.

MASK specifies the name of mask to be used for image calculation.

SHIFT.MASK specifies the name of the additional mask layer - usually phase shifting layer.

Examples

The following statement describes a mask feature that is 2 microns in the x dimension and 0.4 microns in the z direction and rotated by 45° with respect to the x-axis.

```
LAYOUT X.LO=-1 X.HI=1 Z.LO=-0.2 Z.HI=0.2 ROT.ANGLE=45 \

       TRANSMIT=1
```

For more examples, see IMAGE, ILLUMINATION, PROJECTION, ILLUM.FILTER, PUPIL.FILTER, ABERRATION, and the MaskViews User's Manual.

6.33 LINE

LINE specifies a mesh line during grid definition.

Syntax

```
LINE
X|Y LOCATION=<n> [SPACING=<n>] [TAG=<c>] [TRI.LEFT|TRI.RIGHT]
```

Description

This statement defines the position and spacing of mesh lines. All LINE statements should come before the REGION and BOUNDARY statements, which should then be followed by an INITIALIZE statement.

X and **Y** specify whether a mesh line is horizontal or vertical.

LOCATION specifies the location along the chosen axis (in microns) at which the line should be positioned. The **x** coordinate increases from left to right; the **y** coordinate increases progressing from top to bottom going into the substrate. This is the opposite of normal Cartesian y-axis progression which increases going upward.

SPACING specifies the local grid spacing (in microns). Athena adds mesh lines to the ones given according to the following recipe. Each user line has a spacing whether user-specified or inferred from the nearest neighbor. These spacings are then smoothed out so no adjacent intervals have a ratio greater than the value given by INTERVAL.R on the INITIALIZE statement (default is 1.5). New grid lines are then introduced so that the line spacing varies geometrically from one end of the interval to the other. Refer to the example below.

TAG labels lines for later reference by BOUNDARY and REGION statements. The tag label can be any word.

TRI.LEFT and **TRI.RIGHT** can be specified in the LINE X statement to control triangle orientation in the initial grid. Initial simulation area is divided into non-uniform rectangular elements defined by the LINE X and LINE Y statements. By default each box element is then divided into two rectangular triangles by a diagonal going from the bottom-left to upper-right corner of the box. In some applications it is preferable to have a symmetrical grid triangle orientation locally.

One of the examples is etching of a non-vertical trench. If TRI.RIGHT (default) is specified all boxes between this line and the line with TRI.LEFT (or the last line) will be divided by a "bottom-left to upper-right" diagonal. If TRI.LEFT is specified the boxes will be divided by a diagonal going from upper-left to bottom-right corner of the box.

Example

In the following specifications, there are 3 user-specified x lines and 2 user-specified y lines. Spacing of the x lines is finer in the center than at the edges. After processing, Athena produces a mesh with x lines at 0.0, 0.42, 0.69, 0.88, 1.0,1.12, 1.31, 1.58, 2.0. Around the center, the spacing is 0.12, approximately what was requested. At the edge, the spacing is 0.42 because that was as coarse as the line spacing could get without having an interval ratio greater than 1.5. If the interval ratio is set to 9, then we would have one interval of 0.9 and one interval of 0.1 on each side. In this example, specifying a spacing of 1 would produce an x line at 0.0 and 1.0.

```
LINE X LOC=0 SPA=1 TAG=LEFT
LINE X LOC=1 SPA=0.1

LINE X LOC=2 SPA=1 TAG=RIGHT

LINE Y LOC=0 SPA=0.02 TAG=SURF

LINE Y LOC=3 SPA=0.5 TAG=BACK
```

Note: It is difficult to predict how many lines are going to be generated in each interval. The initial mesh specification is quite important to the success of the simulation. Use the geometric mode by specifying the NO.IMP parameter on the INITIALIZE statement to perform a fast simulation without impurities to determine if the grid spacings are appropriate.

For more examples, see INITIALIZE, REGION, BASE.MESH, and BASE.PAR.

6.34 MASK

MASK deposits and patterns photoresist or artificial masking material barrier through the MaskViews interface.

Syntax

```
MASK
NAME=”<c>” [REVERSE] [DELTA=<n>]
```

Description

MASK is used in DeckBuild to provide interface to Silvaco’s general purpose layout editor, MaskViews. When you specify a mask statement, Athena will deposit photoresist and pattern it by etching. The etched pattern is determined by selected cut line in MaskViews. See the MaskViews User’s Manual for a complete description of this feature.

NAME specifies the name of the layer that defines the photoresist patterning. Mask names must appear inside of double quotes. This name must correspond to a mask name contained in the layout file invoked using DeckBuild. The mask names are case sensitive and cannot be abbreviated.

REVERSE specifies that the mask polarity should be reversed or that negative type photoresist should be modeled.

DELTA specifies an offset in mask size. The offset corresponds to a change in CD (critical dimension) of the mask. Each edge of the mask is moved by a distance DELTA to enlarge or contract the mask feature.

Examples

The following statement deposits photoresist on the top of the simulation structure and etches it with the pattern prescribed by the MaskViews layout. The layout file must be specified using the MaskViews interface as described in the MaskViews User’s Manual or in Chapter 2 “Tutorial”.

```
MASK NAME="CONT"
```

For more examples, see STRIP.

6.35 MATERIAL

MATERIAL sets the coefficients for materials.

Syntax

```
MATERIAL
 [MATERIAL] [NI.0=<n>] [NI.E=<n>] [NI.POW=<n>] [EPS=<n>] [E.FIELD]
 [VISC.0=<n>] [VISC.E=<n>] [VISC.X=<n>] [WETO2|DRY02]
 [S11.ELAST=<n>] [S12.ELAST=<N>]
 [S44.ELAST=<n>] [LCTE=<c>] [INTRIN.SIG=<n>]
 [OXIDIZABLE] [IMMOBILE] [DENSITY=<n>] [COMPONENTS]
 [AT.NUM.1=<n>] [AT.NUM.2=<n>] [AT.NUM.3=<n>] [AT.NUM.4=<n>]
 [AT.MASS.1=<n>] [AT.MASS.2=<n>] [AT.MASS.3=<n>] [AT.MASS.4=<n>]
 [ABUND.1=<n>] [ABUND.2=<n>] [ABUND.3=<n>] [ABUND.4=<n>]]
 [REFLOW] [GAMMA.REFLO=<n>] [NO.FLIP]
 [NIFACT.SIGE=<n>] [EAFACT.SIGE=<n>] [NIFACT.SIC] [EAFACT.SIC]
```

Description

MATERIAL specify the material for which all parameters apply (see Section 6.2.9 “Standard and User-Defined Materials” for the list of materials).

Parameters Related to Impurity Diffusion

NI.0, **NI.E**, and **NI.POW** specify parameters of the intrinsic electron concentration as a function of temperature. NI.0 is the preexponential constant in the intrinsic electron concentration formula. NI.E is the corresponding activation energy. NI.POW is the unitless power constant. These parameters are used only in diffusion calculation and not in EXTRACT electrical calculations

EPS specifies the relative dielectric permittivity of the material. This value is used to calculate electric field in semiconductors during diffusion simulation. This value isn’t used in EXTRACT electrical calculations.

E.FIELD specifies that electric field term will be included in the impurity diffusion equations for this material. The default is true.

Parameters Related to Material Stress and Viscosity

VISC.0, VISC.E, and **VISC.X** specify the material viscosity parameters. VISC.0 is the pre-exponential coefficient, in g/(cm·s). VISC.E is the activation energy, in eV. VISC.X is the incompressibility factor.

WETO2 and **DRYO2** specify whether the viscosity parameters are for wet or dry oxidation. These parameters are valid only if the specified material is oxide.

LCTE specifies the linear coefficient of thermal expansion as a function of temperature (T). It is expressed as a fraction rather than a percentage.

INTRIN.SIG specifies the initial uniform stress state of a material, such as a thin film of nitride deposited on the substrate. The units are dyne cm^{-2}.

S11.ELAST, **S12.ELAST**, and **S44.ELAST** are elastic compliances used to calculate Young's modulus and Poisson's ratio for stress calculations and compless oxidation.

Parameters Related to Oxidation

OXIDIZABLE specifies that the material could be oxidized. If you set OXIDIZABLE to TRUE, then all oxidation related parameters for this material will be set equal to those specified earlier for Silicon. You can specify different values in the OXIDE statement.

IMMOBILE specifies that this material will not move during oxidation. The default is FALSE. This is useful when a non-oxidizable material(s) is a continuation of a substrate, so it is supposed to be immobile and not distorted or lifted as in a case when it serves as a "floating" mask.

Parameters Related to Material Characteristics used by Monte Carlo Implant

DENSITY specifies the density of the material in g/cm^3.

COMPONENTS specifies number of atomic components in the material.

AT.NUM.1, AT.NUM.2, AT.NUM.3, and **AT.NUM.4** specify the atomic numbers of the constituent atoms of the material. The number of AT.NUM parameters specified must correspond to COMPONENTS.

AT.MASS.1, AT.MASS.2, AT.MASS.3, and **AT.MASS.4** specify the atomic masses of the constituent atoms of the material in atomic mass units. The number of AT.MASS parameters specified must correspond to COMPONENTS.

ABUND.1, ABUND.2, ABUND.3, and **ABUND.4** specify the relative fraction of the constituent atoms of the material. The number of ABUND parameters specified must correspond to COMPONENTS.

Note: At least one parameter from each of the four lines above are required to define materials for Monte Carlo implants.

Parameters Related to REFLOW Calculations

REFLOW specifies that the material will flow when a DIFFUSE step including REFLOW is defined.

GAMMA.REFLO specifies the surface tension parameter used in the reflow calculation. Units are dyne/cm. The material viscosity (VISC.* parameters) will also affect the reflow rate.

Parameters Related to the Grid Control

NO.FLIP specifies that triangle flipping procedure should not be applied to the specified material.

Parameters Related to the Boron Diffusion Model in SiGe/SiGeC

NIFACT.SIGE specifies the linear coefficient for Ge dependency formula of intrinsic carrier concentration for Boron diffusion model in SiGe/SiGeC.

EAFACT.SIGE specifies the linear coefficient for Ge dependency formula of intrinsic carrier concentration for Boron diffusion model in SiGe/SiGeC.

NIFACT.SIC specifies linear coefficient for Ge content dependency formula of intrinsic carrier concentration for Boron diffusion model in SiGe/SiGeC.

EAFACT.SIC specifies linear coefficient for Ge content dependency formula of intrinsic carrier concentration for Boron diffusion model in SiGe/SiGeC.

Example

The following statement defines some properties of a material called BPSG. The material is composed of silicon, oxygen, boron, and phosphorus with fraction composition 0.3, 0.6, 0.05, and 0.05 respectively. Monte Carlo Implants could be performed into this material based on this definition.

```
MATERIAL MATERIAL=BPSG  COMPONENTS=4  DENSITY=2.362 \
         AT.NUM.1=14 AT.NUM.2=8 AT.NUM.3=5 AT.NUM.4=15 \
         AT.MASS.1=28.0 AT.MASS.2=16.0 AT.MASS.3=10.8 \
         AT.MASS.4=31.0 \
         ABUND.1=0.3 ABUND.2=0.6 ABUND.3=0.05 ABUND.4=0.05
```

For more information, see `OXIDE`, `STRESS`, and `DIFFUSE`.

6.36 METHOD

METHOD selects numerical methods and models for diffusion and oxidation.

Syntax

```
METHOD

[FERMI|TWO.DIM|STEADY|FULL.CPL]

[PLS] [IC] [VC] [DDC] [SS]

[CLUSTER.DAM] [HIGH.CONC] [I.LOOP.SINK] [POLY.DIFF]

[CLUST.TRANS]      [DOSE.LOSS]      [BDOSE.LOSS]      [PDOSE.LOSS]
[ASDOSE.LOSS]

[SiGECDF.MOD=<c>]        [SiGECNI.MOD=<c>]        [MODEL.SIGEC]
[MIN.TEMP=<n>]

[IMPURITY|INTERST|VACANCY|OXIDANT|VELOCITY|TRAPS|PSI|PAC]

[REL.ERROR=<n>] [ABS.ERROR=<n>]

[FE.RELERR=<n>] [FE.ABSERR=<n>] [TD.RELERR=<n>] [TD.ABSERR=<n>]

[ST.RELERR=<n>] [ST.ABSERR=<n>] [FU.RELERR=<n>] [FU.ABSERR=<n>]

[MIN.FILL] [MIN.FREQ=<n>] [GAUSS|CG] [BACK=<n>] [BLK.ITLIM=<n>]

[TIME.STR|ERROR|NEWTON] [DIAG|KNOT|FULL.FAC] [TRUNC.DEF=<n>]

[INIT.TIME=<n>] [PDINIT.TIME] [T.DEFECT=<n>]

[OXIDE.GDT=<n>] [REDO.OXIDE=<n>] [TRBDF|FORMULA]

[ERFC|ERFG|ERF1|ERF2|COMPRESS|VISCOUS]

[LIFT.POLY] [LIFT.OXIDE] [LIFT.NITRID] [OX.THRESH=<n>] [SKIP.SIL]

[GRID.OXIDE=<n>] [GRIDINIT.OX=<n>]

[GRID.SILICI=<n>] [GRIDINIT.SI=<n>

[GLOOP.IMAX=<n>] [GLOOP.EMIN=<n>] [GLOOP.EMAX=<n>]

[OXIDE.EARLY=<n>] [OXIDE.LATE=<n>] [OXIDE.REL=<n>]

[OX.OBFIX=<n>] [FILL] [PERIMETER=<n>] [MINTAREA.OX=<n>]

[ETCH.EPS=<n>] [DEPO.ACCU.FACT]

[ADAPT] [DEPO.SMOOTH] ETCH.SMOOTH] [DIFF.SMOOTH]

[STEP.SMOOTH] [STRESS.HIST] [DOPANT.STRESS] [STRAIN.DIFF]
```

Description

This statement is used to set flags to select the various mathematical algorithms that will be used to produce the simulation and to select the desired diffusion and oxidation model complexity. Appropriate defaults for the numerical parameters are included in the *athenamod* file so that you only need to specify the desired diffusion and oxidation model. The numerical methods used in Athena for the solution of the diffusion equations are described in [7].

Parameters Related to DIFFUSION models

FERMI, TWO.DIM, STEADY, and **FULL.CPL** specify the type of diffusion equations to be solved with particular regard to the point defect models (see Section 3.1 "Diffusion Models"). The FERMI parameter specifies the defects are assumed to be a function of the Fermi level only. The TWO.DIM parameter specifies that a full time dependent transient simulation should be performed. The STEADY parameter specifies that the defects are assumed to be in a steady state. The FULL.CPL parameter specifies that full coupling between defects and dopants should be included. The default is FERMI. PD.FERMI is an alias for FERMI. PD.TRANS is an alias for TWO.DIM. PD.FULL is an alias for FULL.CPL. PD.FERMI is an alias for FERMI. PD.TRANS is an alias for TWO.DIM. PD.FULL is an alias for FULL.CPL.

Note: Chapter 2 "Tutorial" shows a complete description of the use of these diffusion models for typical applications.

PLS, IC, VC, DDC, and **SS** specify advanced diffusion models (see Section 3.2 "Advanced Diffusion Models"). If only the PLS parameter is specified, the classical dopant diffusion model will be used. The parameters IC and VC will invoke additional interstitial and vacancy clustering models. The DDC parameter switches on the dopant-defect clustering model. The SS parameter will include the solid solubility model.

Note: These advanced diffusion models can be used only for boron, phosphorus and arsenic in silicon technologies Also, these models cannot be used when oxidation or silicidation or both occur during the simulated diffusion step. There are also some limitation on complexity of the 2D structures, which can be handled by the solver. In most cases, when the solver cannot handle the structure, materials, impurities or other conditions, it returns control to standard diffusion models.

CLUSTER.DAM specifies that the Stanford {311} cluster model is enabled, allowing a scaled profile of {311} clusters during a subsequent implant. Only use this model when FULL.CPL is also specified. It further causes a transient dissolution of the {311} clusters leading to bulk interstitial injection. The CLUSTER statement is used to set parameters for this model.

Note: For correct operation, set METHOD CLUSTER.DAM FULL.CPL before the IMPLANT statement that generates the {311} clusters.

HIGH.CONC specifies that extra dopant concentration dependent point defect recombination model is enabled. The IIFACTOR and IVFACTOR parameters on the INTERSTITIAL command are used when METHOD HIGH.CONC is enabled.

I.LOOP.SINK specifies that a dislocation loop band can be specified during a subsequent implant and that the loops may behave as an interstitial sink during diffusion. The DISLOC.LOOP command is used to set parameters for this model.

POLY.DIFF specifies that the two-stream polysilicon diffusion model should be used. To operate accurately, set this model before the deposition of the polysilicon material. See

Section 6.29 “IMPURITY” and Section 3.1.7 “Grain-based Polysilicon Diffusion Model” for more information.

CLUST.TRANS enables the Transient Activation Model.

DOSE.LOSS specifies that Interface Trap Model for dose loss at silicon/oxide interface is enabled for all three impurities B, P, and AS.

BDOSE.LOSS specifies that Interface Trap Model for dose loss at silicon/oxide interface is enabled for boron.

PDOSE.LOSS specifies that Interface Trap Model for dose loss at silicon/oxide interface is enabled for phosphorus.

ASDOSE.LOSS specifies that Interface Trap Model for dose loss at silicon/oxide interface is enabled for arsenic.

MODEL.SIGEC enables special B diffusion model in SiGeC/SiGeC.

SIGECDF.MOD specifies the name of the C-Interpreter file for boron diffusivity model in SiGe.

SIGECNI.MOD specifies the name of the C-Interpreter file instrinic carrier concentration model used in boron diffusion mode in SiGe/SiGeC.

MIN.TEMP specifies the minimum temperature for which impurity diffusion is considered. At temperatures below `MIN.TEMP`, the impurities are considered immobile. The default is 700°C. With caution, you can set this parameter to a lower value for certain diffusion steps.

Parameters Related to Numerics of Diffusion/Oxidation

IMPURITY specifies impurity for which one or several bound (tolerance) parameters will be applied during diffusion/oxidation simulation (see Section 6.2.10 “Standard Impurities” for the list of impurities).

INTERST, VACANCY, OXIDANT, VELOCITY, TRAPS, PSI and **PAC** specifies type of solution for which one or several bound (tolerance) parameters will be applied during diffusion/oxidation simulation.

REL.ERR indicates the precision with which the impurity solution must be solved. In general, the actual error will be less than half of the indicated error. The defaults are 0.01 for all impurities except the potential, which is solved to 0.001. If this parameter is used an impurity should also be specified.

ABS.ERR specifies the error tolerance absolute value. For dopants, the absolute error defaults to 1.0×10^{9}. For defects, the absolute error defaults to 1.0×10^{5}. For the potential, the error defaults to 1.0×10^{-6}. If this parameter is used, an impurity should also be specified.

FE.RELERR and **FE.ABSERR** specifies the relative error and absolute errors for the `FERMI` model.

TD.RELERR and **TD.ABSERR** specifies the relative and absolute errors for the `TWO.DIM` model.

ST.RELERR and **ST.ABSERR** specifies the relative and absolute errors for the `STEADY` model.

FU.RELERR and **FU.ABSERR** specifies the relative and absolute errors for the `FULL.CPL` model.

MIN.FILL and **MIN.FREQ** specify a minimum fill. It defaults to true. This is a highly recommended option since it can reduce the matrix sizes by a factor of two or more and operation speed is a function of the size of the matrix. MIN.FREQ is a parameter that controls the frequency of the minimum fill reorderings. It is only partially implemented and has no effect on the calculation.

GAUSS and **CG** specify the type of iteration performed on the linear system as a whole. CG specifies that a conjugate residual should be used.

BACK specifies the number of back vectors that can be used in the CG outer iteration. The default is three and the maximum possible value is six.

Note: A higher value of BACK will give faster convergence at the cost of more memory usage.

BLK.ITLIM is the maximum number of block iterations that can be taken. The block iteration will finish at this point independent of convergence.

TIME.STE, ERROR, and **NEWTON** specify the frequency with which the matrix should be factored. The default is TIME. The TIME parameter specifies that the matrix should be factored twice per time step. This option takes advantage of the similarity in the matrix across a time integration. The ERROR parameter indicates that the matrix should be factored whenever the error in that block is decreasing. The NEWTON parameter forces factorization at every NEWTON step.

DIAG, KNOT, and **FULL.FAC** specifies the amount of fill to be included in the factorization of the matrix. FULL.FAC indicates that the entire amount of fill is to be computed. The DIAG parameter indicates that only the diagonal blocks should be factored in the matrix. The KNOT parameter is inactive. DIAG is the default parameter. Although under certain conditions (one-dimensional stripes), FULL.FAC will perform better.

TRUNC.DEF specifies that defect concentrations that become negative due to numerical difficulties be forced to a positive value.

Parameters Related to Timestep Control

INIT.TIME specifies the initial timestep value. The default is 0.1 seconds.

PDINIT.TIME specifies the initial time step for point defect diffusion. Point defects are held fixed for the first timestep. The default is 10^{-5} seconds.

T.DEFECT specifies time in seconds for which point defect injection will be neglected during an oxidation. The default is 5 seconds.

OXIDE.GDT limits the timestep during oxidation to a fraction (OXIDE.GDT) of the time required to oxidize the thickness of one grid layer (GRID.OXIDE). The timestep may be limited by oxidation and by diffusion, and the value of OXIDE.GDT will limit the timestep if it is more stringent than the limits imposed by diffusion. OXIDE.GDT<<1 is recommended to improve resolution of oxidizing diffusions. The default is 0.25.

REDO.OXIDE saves time by not computing the oxide flow field every time the diffusion equation for impurities is solved. The REDO.OXIDE parameter specifies the percentage of the time required to oxidize the thickness of one grid layer, which should elapse before resolving the flow field. Usually REDO.OXIDE is much less than OXIDE.GDT, which is an upper bound on how long the solution should wait. It is mainly intended to exclude solving oxidation at each and every one of the first few millisecond time steps when defects are being tracked.

TRBDF and **FORMULA** specify the time integration method to be used. The TRBDF parameter indicates that a combination trapezoidal rule/backward difference should be used. The error is estimated using Milne's device. The FORMULA method allows you to specify the time step directly as a function of time (t), previous time step (dt), and grid time (gdt). This option is primarily for testing. The TRBDF method is the default.

Parameters related to OXIDATION models

ERFC, ERFG, ERF1, ERF2, COMPRESS, and **VISCOUS** are oxidation models (see Section 3.3 "Oxidation Models"). The ERFC parameter indicates that a simple error function approximation to a bird's beak shape should be used. The ERF1 and ERF2 models are analytic approximations to the bird's beak from the literature. The ERFG model chooses whichever of ERF1 or ERF2 is most appropriate. All erf models are applicable only to the simplest case of oxidation to the right of the mask edge. All relevant parameters in the OXIDE statement must be explicitly specified when using any of the ERF* models. The COMPRESS model regards the oxide as a compressible liquid. The VISCOUS model treats the oxide as an incompressible viscous liquid. Oxide is actually believed to be incompressible, but the compressible model runs faster. The default is the COMPRESS model.

Note: For Hints on the use of the different oxidation models, see Chapter 2 "Tutorial".

Note: Use of the VERTICAL model is not recommended in Athena

LIFT.POLY, LIFT.OXIDE, and **LIFT.NITRID** specifies that the polysilicon, oxide, and nitride materials can be lifted by oxidation or silicidation processes. These are true by default, but you set them to false to eliminate the lifting portion of the calculation for geometries where lifting is not expected to occur.

OX.THRESH specifies that the oxidation threshold model is enabled. This doesn't allow oxidation when the concentration of oxidant drops below a critical threshold value set by MIN.OXIDANT on the OXIDE statement.

SKIP.SIL is a Boolean parameter which controls the computation of stress in silicon. SKIP.SIL defaults to true. stress can only be computed when the VISCOUS oxide model is used. The silicon substrate is treated as an elastic solid subject to the tensions generated by the oxide flow. Indiscriminate use is not recommended. The silicon grid is usually much larger than the oxide grid and stress computation is correspondingly more lengthy.

Parameters related to Grid Control during Oxidation

Many grid related problems during oxidation are related to the initial oxide deposition. See Section 6.40 "OXIDE" for more about initial oxides.

GRID.OXIDE specifies the desired thickness, in microns, of grid layers to be added to the growing oxide. It has an effect on time steps (refer to OXIDE.GDT). The default is 0.1 microns.

GRIDINIT.OX specifies the initial oxide grid spacing (in microns). The default is 0.1 microns.

GRID.SILICI specifies the maximum silicide grid spacing (in microns). The default is 0.1 microns.

GRIDINIT.SI specifies the initial silicide grid spacing (in microns). The default is 0.1 microns.

GLOOP.EMIN, GLOOP.EMAX, and **GLOOP.IMAX** controls loop detection during grid manipulation. The default value is `GLOOP.IMAX=170°`. Loop detection checks for intrusions and extrusions in the boundary. The intrusion-fixing algorithm is triggered by angles greater than `GLOOP.IMAX`. A larger value means that more extreme intrusions can develop and increases the possibility of a tangled grid. A smaller value leads to earlier intrusion-fixing; too small a value will lead to inaccuracy due to premature intervention. Similar concerns apply to the other parameters. The values are a compromise between safety and accuracy. The extrusion-fixing algorithm is always triggered by angles greater than `GLOOP.EMAX`. It may be triggered by lesser extrusions, anything greater than `GLOOP.EMIN`, if the extrusion is a single-triangle error in the boundary. The default value is `GLOOP.EMIN=130°`. Neither of these parameters should be less than 90° because the rectangular edges of the simulation space would be smoothed.

OXIDE.EARLY, OXIDE.LATE, and **OXIDE.REL** should not normally be modified. They relate to internal numerical mechanisms and are described here only for the sake of completeness. A node whose spacing decreases proportionally by more than `OXIDE.LATE` is marked for removal. Also, if any nodes are removed, then all nodes greater than `OXIDE.EARLY` are also removed. For earlier node removal (fewer obtuse triangles), try `OXIDE.LATE=0.3` and `OXIDE.EARLY=0.1`. Though not logical, it is harmless for `OXIDE.EARLY` to be greater than `OXIDE.LATE`. The `OXIDE.REL` parameter is the percentage error in velocities for the non-linear viscous model. The default is 1.0×10^{-2} (that is, a 1.0 percent error). `OXIDE.REL` can be increased for a faster solution.

OX.OBFIX specifies the cosine squared of the worst angle allowed during oxidation.

FILL and **PERIMETER** specify which action to apply to voids that may form during oxidation. `FILL` specifies that you must fill the voids with oxide materials. Default is false. `PERIMETER` specifies the maximum perimeter of the voids to fill. Default is 0.2 microns.

MINTAREA.OX specifies the minimum area of triangles generated during oxidation, unitless. The default is zero. If `MINTAREA.OX` is greater than zero, then the oxide triangles with area smaller than `MINTAREA.OX` is multiplied by the total area will be marked for elimination. The typical safe value for `MINTAREA.OX` is around 1e-06. The use of `MINTAREA.OX` is recommended mostly when small (< 0.01 microns) values for `GRID.OX` is used for non-planar complex oxidations.

Parameters related to Grid Control during ETCH and DEPOSITION

ETCH.EPS sets a tolerance on the grid movement during `ETCH` statements. This parameter is defined in relative units. The default is 10^{-6} that corresponds to about 10 Angstroms. Reducing this number will allow sub-10A etches to be exact. But the possibility of small triangles being created during etches is high if the parameter is set too low. This parameter should not be set to zero.

DEPO.ACCU.FACT specifies a scaling factor that allows you to improve accuracy of geometrical deposition and epitaxy. The default value is 1. Values of 10 or more are recommended for device structures with individual features of few nanometers or smaller sizes.

Parameters used in the Adaptive Meshing Module

ADAPT specifies that the adaptive meshing should be performed on the IMPLANT, DIFFUSE, or EPITAXY statements (the default is false).

DEPO.SMOOTH specifies that the mesh smoothing should be performed after each DEPOSIT statement.

ETCH.SMOOTH specifies that the mesh smoothing should be performed after each ETCH statement.

DIFF.SMOOTH specifies that the mesh smoothing should be performed after each DIFFUSE statement.

STEP.SMOOTH specifies that the mesh smoothing should be performed after each time step on each DIFFUSE statement.

Parameters related to Stress Simulation

STRESS.HIST specifies that stresses to be calculated during etching, deposition, diffusion, and epitaxy process steps. Default is False.

DOPANT.STRESS specifies that dopant-induced stress simulation model should be used either with the STRESS statement or when STRESS.HIST is specified. Default is False.

STRAIN.DIFF specifies that stress induced diffusion model should be used when stresses have been calculated previously (i.e., STRESS.HIST is specified).

Example setting tolerances

The following statement specifies that the arsenic equation should be solved with a relative error of 1% and concentrations below 1 x 10^9 can be ignored.

```
METHOD ARSEN REL.ERR=0.01 ABS.ERR=1.0E9
```

Example setting numerical techniques

The following statement specifies that minimum fill reordering should be done and that the entire system should be solved using a conjugate residual technique with three back vectors. The initial time step should be 0.1 seconds and time should be integrated using the TRBDF parameter. The FERMI model should be used for diffusion and the COMPRESS model for the oxide growth.

```
METHOD MIN.FILL CG BACK=3 INIT.TI=0.1 TRBDF FERMI COMPRESS
```

Example setting diffusion model for power devices

The following step specifies that a simple diffusion model should be used appropriate for power electronic devices.

```
METHOD POWER

DIFFUSION TEMP=1000 TIME=300 NITROGEN
```

Example setting diffusion models for RTA

The following statement invokes all {311} cluster models for RTA simulation. It must be set before the IMPLANT statement that generates the cluster damage.

```
METHOD NEWTON FULL.CPL CLUSTER.DAM I.LOOP.SINK HIGH.CONC BACK=6

IMPLANT ....

DIFFUSE ....
```

6.37 MOMENTS

MOMENTS specifies tables and spacial moments used in analytical implant models.

Syntax

```
MOMENTS

[SVDP_TABLES | STD_TABLES | USER_STDT] [USER_TABLE=<c>]

[MATERIAL] [I.IMPUTITY] [DOSE=<n>] [TMA_TABLES] [TBL.PATH=<c>]
[ENERGY=<n>]

[RANGE=<n>] [STD.DEV=<n>] [GAMMA=<n>] [KURTOSIS=<n>

[LSTD.DEV] [LGAMMA] [LKURTOSIS] [SKEWXY]

[SRANGE=<n>] [SSTD.DEV=<n>] [SGAMMA=<n>] [SKURTOSIS=<n>]

[LSSTD.DEV] [SSKEWXY] [SKURTXY] [SKURTT] [DRATIO=<n>]

[IGNORE_MOM]
```

Description

Parameters Used to Select Moment Tables

SVDP_TABLES specifies that the SIMS Verified Dual Pearson (SVDP) moments tables will be used with dual Pearson implant model. Default is true (See Section 3.5.1 “Analytic Implant Models”).

STD_TABLES specifies that SVDP_TABLES are ignored and standard tables are used with the subsequent implant statements.

USER_STDT specifies the user-defined moments file (see the USER_TABLE parameter) will be used with standard format. You can find a template for the user-defined moments file in <install.area>/lib/athena/<version>/common/userimp.

USER_TABLE=<c> specifies the file that contains user-defined look-up implant parameter tables.

TMA_TABLES specifies that moments from T-SUPREM4 format implant tables will be used.

TBL_PATH specifies the full path to a user-defined T-SUPREM4 implant tables file. By default, the T-SUPREM4 tables file is to be located in the [install]/lib/athena/<version>/common/implant_tables folders.

Implant Definition Parameters

MATERIAL specifies the material for which the implant moments are set (see Section 6.2.9 “Standard and User-Defined Materials” for the list of materials).

I.IMPURITY specifies the implanted impurity for which the moments are set (see Section 6.2.10 “Standard Impurities” for the list of impurities). I.BF2 can be also specified.

DOSE is an incident ion dose (/cm^2).

ENERGY sets the incident ion energy (keV).

Parameters Used for Specification of Spacial Moments

RANGE (RP) specifies the projected range. Units are microns.

STD.DEV (**DRP**) specifies the standard deviation Units are microns.

GAMMA (SKEWNESS) specifies the third moment. Default is 0.0.

KURTOSIS specifies the fourth moment. Default is 3.01.

LSTD.DEV(LDRP) specifies the lateral standard deviation Units are microns.

SKEWXY specifies the mixed third moment.

KURTXY specifies the lateral mixed fourth moment.

KURTT specifies the lateral fourth moment. Default is 3.0.

SRANGE (SRP) specifies the projected range for second Pearson Units are microns.

SSTD.DEV(SDRP) specifies the standard deviation for second Pearson Units are microns.

SGAMMA (SSKEW) specifies the third moment for second Pearson function. Default is 0.0.

SKURTOSIS specifies the fourth moment for second Pearson function. Default is 3.01.

LSSTD.DEV(LSDRP) specifies the lateral standard deviation for second Pearson Units are microns.

SSKEWXY specifies the mixed third moment for second Pearson. Default is 0.0.

SKURTXY specifies the mixed fourth moment for second Pearson. Default is 0.0

SKURTT specifies the lateral fourth moment for second Pearson. Default is 3.0.

DRATIO specifies the dose ratio R in the double Pearson function. Default is 0.9

Reset Parameter

IGNORE_MOM specifies that all previous MOMENTS statements will be ignored.

Examples

The MOMENTS statement is used to define user moments through a convenient command language. The following example sets the moments for boron implantation into the user-defined material SAPPHIRE.

```
MOMENTS MATERIAL=SAPPHIRE I.BORON DOSE=1.6e12 ENERGY=25 \

        RANGE=0.098 STD.DEV=0.045 GAMMA=-0.04 KURTOSIS=3.5
```

For more examples, see IMPLANT.

6.38 OPTICAL

OPTICAL sets the optical parameters of materials for Optolith.

Note: This statement is not available in Athena 1D.

Syntax

```
OPTICAL
[MATERIAL] [NAME.RESIST=<c>]
[LAMBDA=<n>] [I.LINE|G.LINE|H.LINE|DUV.LINE]
[REFRAC.REAL=<n>] [REFRAC.IMAG=<n>]
[DELTA.REAL=<n>] [DELTA.IMAG=<n>]
```

Description

This command sets the optical parameters, reflective index, and extinction coefficient or REFRAC.REAL and REFRAC.IMAG for each material at a particular wavelength. If photoresist is used, NAME.RESIST must also be specified.

MATERIAL specifies the material for which the optical parameters to be set (see Section 6.2.9 “Standard and User-Defined Materials” for the list of materials)

NAME.RESISTspecifies the name of the photoresist.

I.LINE, G.LINE, H.LINE, DUV.LINE, and **LAMBDA** specifies the line or the wavelength. Units for LAMBDA are microns. WAVELENGTH is an alias for LAMBDA.

REFRAC.REAL specifies the real component of the refractive index.

REFRAC.IMAG specifies the imaginary component of the refractive index.

DELTA.REAL specifies the difference between the real components of the refractive index for completely exposed and unexposed resist. This value is used when dose effect on the refractive index is simulated.

DELTA.IMAG specifies the difference between the imaginary components of the refractive index for completely exposed and unexposed resist. This value is used when dose effect on the refractive index is simulated.

Examples

The OPTICAL statement is used to load refractive index values into Athena for each wavelength. The following shows a typical statement.

```
OPTICAL SILICON WAVELENGTH=.365 REFRAC.REAL=4.5 REFRAC.IMAG=5.2
```

You can enter user-defined materials in the following format.

```
OPTICAL MATERIAL=XXX WAVELENGTH=.365 REFRAC.REAL=1.4 \
REFRAC.IMAG=.3
```

For more examples, see EXPOSE and IMAGE.

6.39 OPTION

OPTION specifies the level of run-time output.

Syntax

```
OPTION
```

```
[QUIET|NORMAL|VERBOSE|DEBUG|WARNING]
```

Description

This statement specifies the level of information sent to the TTY Terminal Window of DeckBuild.

QUIET, NORMAL, VERBOSE, DEBUG, and **WARNING** determines the amount of information that is output about errors, CPU times, and behavior of the algorithms. The default is QUIET. The VERBOSE and DEBUG modes are intended mainly for debugging by developers.

Examples

The following statement sets the routine output to include more information.

```
OPTION NORMAL
```

6.40 OXIDE

OXIDE specifies coefficients for use during oxidation steps. AMBIENT is a synonym for OXIDE.

Syntax

```
OXIDE

DRY02|WET02 [ORIENT=<n>]

[LIN.L.0=<n>][LIN.L.E=<n>] [LIN.H.0=<n>][LIN.H.E=<n>

[L.BREAK=<n>][L.PDEP=<n>]

[PAR.L.0=<n>][PAR.L.E=<n>] [PAR.H.0=<n>][PAR.H.E=<n>]

[P.BREAK=<n>][P.PDEP=<n>]

[ORI.DEP][ORI.FAC=<n>]

[HCL.PC=<n>][HCLT=<n>][HCLP=<n>][HCL.PAR=<n>][HCL.LIN=<n>]

[THINOX.0=<n>][THINOX.E=<n>][THINOX.L=<n>][THINOX.P=<n>]

[BAF.DEP][BAF.EBK=<n>][BAF.PE=<n>][BAF.PPE=<n>]

[BAF.NE=<n>][BAF.NNE=<n>][BAF.K0=<n>][BAF.KE=<n>]

[STRESS.DEP][VC=<n>][VR=<n>][VD=<n>][VT=<n>][DLIM=<n>]

[MATERIAL][/MATERIAL]

[DIFF.0=<n>][DIFF.E=<n>][SEG.0=<n>][SEG.E=<n>]
[TRN.0=<n>][TRN.E=<n>]

[HENRY.COEFF=<n>] [THETA=<n>] [ALPHA=<n>] [MIN.OXIDANT=<n>]

[INITIAL=<n>] [SPLIT.ANGLE=<n>]

[SPREAD=<n>][MASK.EDGE=<n>][NIT.THICK=<n>][ERF.Q=<n>]

[ERF.DELTA=<n>][ERF.LBB=<n>][ERF.H=<n>]
```

Description

All parameters relating to oxidation are specified in this statement. Oxidation models are specified in the METHOD statement. All oxidation models are described in Section 3.3 “Oxidation Models”. To properly set values for most coefficients, you need to know whether to use wet or dry oxidation and to know the substrate orientation.

Note: If a required parameter is omitted (e.g., orientation when a linear rate coefficient is being specified), then the statement is ignored without warning.

Oxide Growth Rate Parameters

DRYO2, WETO2 specifies the type of oxidation to which specified coefficients apply. Required for everything except for one-dimensional coefficients and the volume ratio.

ORIENT is the substrate orientation the coefficients specified apply to the required for orientation factor (see ORI.FAC) and thin oxide coefficients. Only 100, 110, and 111 are recognized. The default is 100.

LIN.L.0, LIN.L.E, LIN.H.0, LIN.H.E, L.BREAK, and **L.PDEP** specifies the linear rate coefficients (B/A). A doubly activated Arrhenius model is assumed. `L.BREAK` is the temperature breakpoint between the lower and higher ranges in degrees Celsius. `LIN.L.0` is the pre-exponential factor in microns/min. `LIN.L.E` is the activation energy in eV for the low temperature range. `LIN.H.0` and `LIN.H.E` are the corresponding high temperature numbers. `L.PDEP` is the exponent of the pressure dependence. The value given is taken to apply to <111> orientation and later adjusted by `ORI.FAC` according to the substrate orientation present.

PAR.L.0, PAR.L.E, PAR.H.0, PAR.H.E, P.BREAK and **P.PDEP** specifies the parabolic rate coefficients (B).

ORI.FAC is the ratio of B/A on the specified orientation to the orientation.

ORI.DEP specifies whether the local orientation at each point on the surface should be used to calculate B/A. The default is true. If it is false, the substrate orientation is used at all points.

THINOX.0, THINOX.E, and **THINOX.L** specifies coefficients for the thin oxide model proposed by Massoud [14]. `THINOX.0` is the pre-exponential factor in microns/min, `THINOX.E` is the activation energy in eV, and `THINOX.L` is the characteristic length in microns.

THINOX.P is the thin oxide model pressure dependence.

HCL.PC, HCLT, HCLP, HCL.PAR, and **HCL.LIN** is where the numerical parameter, `HCL.PC`, is the percentage of HCl in the gas stream. It defaults to 0. The HCl dependence of the linear and parabolic coefficients is obtained from a look-up table specified in the model file. The table rows are indexed by HCl percentage. Specify the row entries with the parameter `HCLP`, which is an array of numerical values surrounded by double quotes and separated by spaces or commas. The columns are indexed by temperature. Specify the column entries with the parameter `HCLT`, which is an array of numerical values surrounded by double quotes and separated by spaces or commas. Specify the dependence of B/A with the parameter `HCL.LIN`, which is an array of numerical values surrounded by double quotes and separated by spaces or commas. The number of entries in `HCL.LIN` must be the product of the number of entries in `HCLP` and `HCLT`. Specify the dependence of B with the parameter `HCL.PAR`, which is an array of numerical values surrounded by double quotes and separated by spaces or commas. The number of entries in `HCL.PAR` must be the product of the number of entries in `HCLP` and `HCLT`.

BAF.DEP, BAF.EBK, BAF.PE, BAF.PPE, BAF.NE, BAF.NNE, BAF.K0, and **BAF.KE** relates to the doping dependence of the oxidation rate. The doping dependence is activated when `BAF.DEP` is true (default). `MATERIAL1` must be specified with these parameters (only `SILICON` and `POLYSILICON` make sense here).

STRESS.DEP, VC, VR, VD, VT, and **DLIM** controls the stress dependence of oxidation, which is only calculated under the `VISCOUS` model. `STRESS.DEP` turns on the dependence. `VC` is the activation volume of viscosity. `VR` is the activation volume of the reaction rate with respect to normal stress. `VT` is the activation volume of the reaction rate with respect to tangential stress. `VD` is the activation volume of oxidant diffusion with respect to pressure. `DLIM` is the maximum increase of diffusion permitted under tensile stress.

MATERIAL specifies `MATERIAL1` for which parameters to be set (see Section 6.2.9 “Standard and User-Defined Materials” for the list of materials).

/MATERIAL specifies `MATERIAL2` for which parameters to be set (see Section 6.2.9 “Standard and User-Defined Materials” for the list of materials).

DIFF.0, DIFF.E, SEG.0, SEG.E, TRN.0, and **TRN.E** specifies the diffusion coefficients of oxidant in MATERIAL1 and the boundary coefficients (“transport” and “segregation”) from MATERIAL1 to MATERIAL2 as defined above. DIFF.0 is the diffusivity pre-exponential factor in cm^2/sec. DIFF.E is the energy in eV. The transport coefficient represents the gas-phase mass transfer coefficient in terms of concentrations in the solid at the oxide-gas interface, the chemical surface-reaction rate constant at the oxide-silicon surface, and a regular diffusive transport coefficient at other interfaces. The segregation coefficient is 1 at the oxide-gas interface, it is infinity at the oxide-silicon interface, and is a regular segregation coefficient at other interfaces.

Note: Oxidant in materials other than oxide is allowed to diffuse and segregate but its concentration is then ignored (for instance, no oxynitridation).

HENRY.COEFF (Henry’s coefficient) is the solubility of oxidant in MATERIAL1 measured in cubic centimeters (cm^3) at one atmosphere. THETA is the number of oxygen atoms incorporated in a cubic centimeter of oxide. In the case of dry oxidation, it is equal to THETA and in the case of wet oxidation, it is equal to 2 THETA. Usually, the Deal-Grove coefficients should be changed instead of HENRY.COEFF.

THETA specifies the concentration of O_2 atoms incorporated in the material. Units are cm^{-3}.

ALPHA specifies the volume expansion ratio between MATERIAL1 and MATERIAL2. Only SILICON, POLYSILICON and OXIDE make sense here.

MIN.OXIDANT specifies the minimum oxidant concentration for oxidation to occur. Units are cm^{-3}. This parameter is active only if METHOD OX.THRESH is used.

Parameters Related to Grid Control

INITIAL specifies the thickness of the native (initial) oxide at the start of oxidation step. If any oxidizable surface of the structure is bare, an oxide layer of this thickness is deposited before oxidation begins. Units are microns. Default is 0.002.

Note: The oxidation algorithm requires selective deposition of a native oxide onto all exposed silicon or polysilicon areas prior to oxidation. Grid problems can result in complex structures. To resolve these problems, adjust INITIAL or use the DEPOSIT statement to create the native oxide.

SPLIT.ANGLE governs the minimum angle at which the oxide will split open one more grid spacing when oxidizing at a triple point (i.e., where silicon, oxide, and nitride coincide together at a point). The default for the split angle is 22.5°. The SPLIT.ANGLE parameter for triple point oxidation is material dependent. Specify the oxidizing MATERIAL1 without a "/" and MATERIAL2 with a "/" using the following format: OXIDE SPLIT.ANGLE=35 SILICON /NITRIDE. There are only three possible combinations and they are SILICON/NITRIDE, SILICON/POLYSILICO, and POLYSILION/NITRIDE. You can use this to control lateral encroachment during oxidation.

Parameters of the Analytical Oxidation Models (ERF*)

SPREAD and **MASK.EDGE** are used only in the error-function approximation to a bird's beak shape. SPREAD is the relative lateral to vertical extension, which defaults to 1. The fitting parameter makes the "erfc" bird's beak look realistic. MASK.EDGE is the position of the mask edge in microns and defaults to negative infinity. Oxide grows to the right of the mask edge.

ERF.Q and **ERF.DELTA** are the DELTA and Q parameters for the "erfg" model. Normally, you don't need to change them but they are available if necessary.

ERF.LBB is the length of the bird's beak and applies to the "erfg" model only. It can be specified as an expression in Eox (the field oxide thickness (µm)), eox (the pad oxide thickness (µm)), Tox (the oxidation temperature (Kelvin)), and en (the nitride thickness, µm). The published expression can be found in the models file. Specifying ERF.LBB=EOX, for instance, would give a lateral spread equal to the field thickness, similar to the Hee-Gook Lee model with a spread of one.

ERF.H is the ratio of the nitride lifting to the field oxide thickness. It corresponds to the Guillemot "H" parameter except that it is normalized to the field oxide thickness. It is specified as an expression of Eox, eox, Tox, en.

NIT.THICK specifies the nitride thickness to substitute for the parameter EN.

Note: The ERFG model uses both oxide and nitride thickness. These values are not inferred from the structure. Instead, the nitride thickness is user-specified in the OXIDE statement and the oxide thickness is computed by adding the total oxide grown and the initial user-specified oxide thickness. If the structure has more than 20 angstroms (the default) of native oxide on it when diffusion begins, then thickness must be specified. Beware of this when continuing a diffusion by any means (e.g., after reading in a previous structure).

Examples

The following modifies the parabolic oxidation rates for {100} silicon in a dry oxygen ambient.

```
OXIDE DRY ORI=100 PAR.L.0=283.333 PAR.L.E=1.17
```

Note: If a required parameter is omitted (e.g., orientation when a linear rate coefficient is being specified), then the statement is ignored without warning.

The following set the native oxide thickness at 10 Angstroms.

```
OXIDE INITIAL=0.001
```

The following defines that stress-dependent oxidation rates will be used with the viscous oxidation model.

```
METHOD VISCOUS

OXIDE STRESS.DEP=t
```

For more examples, see DIFFUSE and METHOD.

6.41 POLISH

POLISH runs the chemical mechanical polishing (CMP) module.

Note: This statement is not available in Athena 1D.

Syntax

```
POLISH
MACHINE=<c> [TIME=<n>] [HOURS|MINUTES|SECONDS]
[DX.MULT=<n>] [DT.FACT=<n>] [DT.MAX=<n>]
```

Description

This statement executes the chemical mechanical polishing module of Elite. The POLISH statement must be preceded by a RATE.POLISH statement to define the polishing machine.

MACHINE specifies the name of the polish machine.

TIME specifies the time the machine is to be run.

HOURS, MINUTES, and **SECONDS** specifies the units of the TIME parameter.

DX.MULT is the accuracy multiplier for Elite polishes. The discretization size used for the polish calculation will be multiplied by DX.MULT. For improved accuracy, decrease the value of DX.MULT. For improved speed, increase the value of DX.MULT.

DT.FACT controls the timestep size. By default, the movement of a string node is limited to less than or equal to one quarter of the median segment length. This is a good compromise between simulation speed and the danger of loop formation. The optimization factor DT.FACT must not exceed 0.5 but can decrease if necessary for more accuracy.

DT.MAX is used with Elite type polish calculations. By default, the upper limit for the micro timestep DT.MAX is one tenth of the total etch time specified. This is a good compromise between calculation accuracy and calculation time. But, sometimes it is useful to adapt this value to the specific simulation problem. Allowing the time steps to become greater gives a higher simulation speed but the accuracy may suffer. For smaller time steps, the simulation speed will decrease but the accuracy may be greater.

Examples

The following statements illustrate running the chemical mechanical polish module. A RATE.POLISH statement sets the values for the polish model and must precede the POLISH statement.

```
RATE.POLISH OXIDE MACHINE=cmp u.s MAX.HARD=0.15 MIN.HARD=0.03 \
ISOTROPIC=0.001
POLISH MACHINE=cmp TIME=5 MIN
```

For more examples, see RATE.POLISH and ETCH.

6.42 PRINT.1D

PRINT.1D prints values along a one–dimensional cross section or an material interface.

Note: Use of this statement is not recommended. All functions are available using the EXTRACT command within DeckBuild.

Syntax

```
PRINT.1D
[X.VALUE=<n>|Y.VALUE=<n>]
[MATERIAL] [/MATERIAL]  [ARCLENGTH] [LAYERS]
[X.MIN=<n>] [X.MAX=<n>] [FORMAT=<c>]
```

Description

This command prints the values along cross sections through the device. You can also use to integrate along a specified line. The value printed is the value that has been selected (see Section 6.55 “SELECT”).

X.VALUE specifies the x coordinate of a vertical cross-section along which the selected values are to be printed. Units are microns.

Y.VALUE specifies the y coordinate of a vertical cross-section along which the selected values are to be printed. Units are microns.

MATERIAL specifies the selected values in the named material at the interface with another material named by /MATERIAL are to be printed (see Section 6.2.9 “Standard and User-Defined Materials” for the list of materials).

ARCLENGTH is only relevant when printing along an interface. If ARCLENGTH is chosen, the printed ordinate is the arclength, measured in microns, along the boundary from the left most point of the curve. If ARCLENGTH is not chosen, the **x** value of the interface location is printed. The coordinate of the left most point is equal to its x coordinate in the mesh layers.

LAYERS instructs the selected print variable to integrate in each material it crosses. The integrated value and material width is reported. Zero crossings of the variable are treated the same as material interfaces.

X.MIN and **X.MAX** specify the minimum and maximum positions along the cross-section to be printed.

FORMAT changes the print format for the variable, using standard format expressions of the C-language. Default is “%-16e”.

Examples

The following statement prints the selected value at x equal to one micron between the top of the mesh and the 3.0 micron point.

```
PRINT.1D X.VAL=1.0 X.MAX=3.0
```

The following prints the selected variable along the silicon side of the silicon/oxide interface.

```
PRINT.1D SILICON /OXIDE
```

For more examples, see SELECT and PRINTF.

6.43 PRINTF

PRINTF is a string printer and desk calculator.

Note: Functions of this statement have been replaced by the EXTRACT statement

Description

The ECHO statement merely prints the string given to it. This is useful for placing comments in an output file. The statement attempts to parse the string to a legal real number if possible. It has a regular expression parser built-in. This allows ECHO to be used as a desk calculator.

Examples

The following command will send the string "Athena Is My Favorite Process Simulator" to standard output.

```
ECHO Athena Is My Favorite Process Simulator
```

The following command will print 4096.

```
ECHO (2^3^4)
```

The following command will print 8.373, which is the solution to the arithmetic expression.

```
ECHO ( 15.0 - 12.0 * EXP( 4.0 - 2.0 / 6.0 ) )
```

6.44 PROFILE

PROFILE reads a 1D doping profile into Athena.

Syntax

```
PROFILE
[INFILE=<c>] [MASTER | XY.SIMS | EXTRACT.SIMS]
[IMPURITY|INTERST|VACANCY|CLUSTER.DAM|DIS.LOOP]
[LAYER1.DIV=<n>] [LAYER2.DIV=<n>]....[LAYER20.DIV=<n>)
```

Description

This statement can be used to load a 1D stream of doping data into an Athena structure. The data might come from a Secondary Ion Mass Spectroscopy (SIMS) profile, from the “SIMS” profile extracted by the EXTRACT command of DeckBuild, or from a 1D simulation in SSuprem3. Data is applied in 1D across the width of the mesh for subsequent 2D simulation.

INFILE specifies the name of the profile data file or Standard Structure File to be loaded.

MASTER or **SSF** indicates that the file to be loaded is an Silvaco Standard Format file. Files generated by SSuprem3 are in this format.

XY.SIMS specifies that the profile file is in a SIMS format described in example below. This is the default.

EXTRACT.SIMS specifies that the profile file is obtained by EXTRACT command of DeckBuild (see DeckBuild User’s Manual for more information).

IMPURITY specifies the impurity type for profile data file. Corresponding active impurity will be also added. See Section 6.2.10 “Standard Impurities” for the list of impurities.

INTERST, VACANCY, CLUSTER.DAM and **DIS.LOOP** specify that profile data file includes a profile of interstitials, vacancies, {311} clusters or dislocation loops, correspondingly.

LAYER1.DIV, LAYER2.DIV,..., LAYER20.DIV specifies the number of subdivisions for each layer when loading SSuprem3 Structure files.

Examples

An example of a PROFILE statement is given below.

```
PROFILE INF=BORON.SIMS BORON
```

In this case, the PROFILE statement specifies that only boron information will be added to the current working silicon structure. The data file BORON.SIMS should be in the following format.

```
#THIS IS SIMS DATA
0.01   1E15
0.02   1.1E15
0.04   1.3E15
0.06   1.5E15
0.1    1.7E15
0.2    1.9E15
```

```
0.4   2.6E15
. . .
```

In the following example, the PROFILE statement will read in a 1D Silvaco's standard format (SSF) file. All doping and layer information will be preserved. This allows you to start a simulation in, for example, SSuprem3 and finish it in Athena. The Athena grid must be set up in the conventional manner first. The PROFILE statement will then include any overlying layers that may have been deposited or grown in creating the SSuprem3 structure. The value LAYER<n>.DIV controls the number of grid points in the overlying layers. The default grid spacing generated for overlying layers is 0.05 µm.

```
PROFILE MASTER INF=SSUPREM3.STR LAYER1.DIV=3 LAYER2.DIV=6
```

The first layer above the substrate will have 3 vertical grid spacings and the second layer above the substrate will have 6 vertical grid spacings. The file SSUPREM3.STR must be a SSF file.

The following is a list of special cases and their solutions.

- If a SSuprem3 structure is deeper than the Athena structure, the PROFILE statement will extend the value of the bottom grid point.
- If a SSuprem3 structure is shallower than the Athena structure, the PROFILE statement will clip the Athena profile.
- Loading a SSF file works only with a bare silicon wafer as a starting point. If you try to use some other material for a substrate, the results are unreliable and unpredictable.
- Any concentrations of dopant initialized in Athena will be overwritten if a PROFILE statement is used to load a SSF file.

6.45 PROJECTION

PROJECTION specifies the basic optical projection parameters for Optolith.

Note: This statement is not available in Athena 1D.

Syntax

```
PROJECTION
[NA=<n>] [FLARE=<n>]
```

Description

This statement specifies the numerical aperture NA, the defocus distance, and the possible flare in the optical or resist systems.

NA is the numerical aperture of the optical projection system.

FLARE is the amount of flare for the particular imaging problem. FLARE must be expressed in percentages.

Examples

The following statement sets the numerical aperture and flare value for the projection system.

```
PROJECTION NA=.5    FLARE=2
```

For more examples, see IMAGE, ILLUMINATION, ILLUM.FILTER, PUPIL.FILTER, LAYOUT, and ABERRATION.

6.46 PUPIL.FILTER

PUPIL.FILTER specifies the projection pupil type and filtering for Optolith.

Note: This statement is not available in Athena 1D.

Syntax

```
PUPIL.FILTER
CIRCLE|SQUARE|GAUSSIAN|ANTIGAUSS
[GAMMA=<n>] [IN.RADIUS=<n>] [OUT.RADIUS=<n>] [PHASE=<n>]
[TRANSMIT=<n>] [CLEAR.FIL]
```

Description

This command allows you to specify four different pupil types and allows spatial filtering in the Fourier transform plane.

CIRCLE, SQUARE, GAUSSIAN, and **ANTIGAUSS** defines or changes the shape of the exit pupil of the projection system. The shape of the pupil must be entered as a character string.

GAMMA defines or changes the GAMMA value for GAUSSIAN and ANTIGAUSS pupil transmittance. GAMMA is a parameter that defines the truncation of the GAUSSIAN by the pupil. In the limit of GAMMA→0, the pupil transmittance will be uniform.

IN.RADIUS and **OUT.RADIUS** defines or changes the intensity transmittance and phase transmittance of an annular zone inside the exit pupil or either the illumination or the projection system. This qualifier is used to simulate spatial filtering techniques. IN.RADIUS and OUT.RADIUS are used to define an annular zone in the exit pupil having the pupil transmittance equal to TRANSMIT and producing the phase angle equal to PHASE. Radius values are specified in fractions of unity and phase is specified in degrees. Note that the annular zones should not overlap. The outer radius of an inner zone must be smaller than the inner radius of an outer zone. The shape of the annular zone is specified by the shape parameter above. The maximum radius is one.

PHASE specifies the phase shift in degrees produced by the pupil filter-180° ≤ PHASE ≤180°).

TRANSMIT specifies the pupil transmittance caused by the pupil filter.

CLEAR.**FIL** resets the projection filter list.

Examples

This set of commands defines a square aperture in the projection pupil that is opaque over a square annular region.

```
PUPIL.FILTER SQUARE
PUPIL.FILTER IN.RADIUS=.1 OUT.RADIUS=.2 PHASE=0 TRANSMIT=0
```

For more examples, see IMAGE, ILLUMINATION, PROJECTION, ILLUM.FILTER, LAYOUT, and ABERRATION.

6.47 QUIT

QUIT terminates execution of Athena. The EXIT, STOP, and BYE statements are synonyms of the QUIT statement.

Syntax

```
QUIT
```

Description

All statements after a QUIT statement will not be checked or executed.

6.48 RATE.DEPO

RATE.DEPO specifies the deposit rates of a machine, which is used in a subsequent DEPOSIT statement.

Note: This statement is not available in Athena 1D.

Syntax

```
RATE.DEPO
MACHINE=<c> MATERIAL | NAME.RESIST=<c>]
CONICAL|CVD|PLANETAR|UNIDIRECT|DUALDIRECT|
HEMISPHERIC|MONTE1|MONTE2|CUSTOM1|CUSTOM2
DEP.RATE=<n> [INFILE=<c>]
[A.H|A.M|A.S|U.S|U.M|U.H|N.M]
[STEP.COV=<n>] [ANGLE1=<n>][ANGLE2=<n>][ANGLE3=<n>]
[C.AXIS=<n>] [P.AXIS=<n>] [DIST.PL=<n>]
[SIGMA.DEP=<n>] [SIGMA.0][SIGMA.E]
[SMOOTH.WIN=<n>][SMOOTH.STEP=<n>][MCSEED=<n>][STICK.COEF=<n>
```

Description

This statement is used to define deposition parameters and the machine name for one of ten deposition models available in Elite.

MACHINE specifies the machine name for the RATE.DEPO statement.

MATERIAL specifies material to be deposited by the deposit machine (see Section 6.2.9 "Standard and User-Defined Materials" for the list of materials).

NAME.RESIST specifies the name of photoresist to be deposited.

CONICAL, CVD, PLANETARY, UNIDIRECT, DUALDIRECT, HEMISPHERIC, MONTE1, MONTE2, CUSTOM1 and **CUSTOM2** specify a particular model for the machine definition

DEP.RATE specifies the deposition rate used by the models CONICAL, CVD, UNIDIREC, DUALDIREC, HEMISPHE, PLANETAR, MONTE1, and MONTE2. DEP.RATE is a rate multiplier for the CUSTOM1 and CUSTOM2 models.

INFILE specifies the name of a file containing angle and deposition rate information for the CUSTOM model.

A.H, A.M, A.S, U.H, U.M, U.S, and **N.M** specify that the deposition rate DEP.RATE is in Angstroms per hour, Angstroms per minute, Angstroms per second, microns per hour, microns per minute, microns per second, and nanometers per minute, respectively. Default is A.S.

STEP.COV specifies the step coverage used by the model CVD.

ANGLE1 specifies the angle parameter used by the models HEMISPHE, CONICAL, UNIDIREC, DUALDIREC, and PLANETAR.

ANGLE2 specifies the angle parameter used by the models DUALDIREC, PLANETAR, and HEMISPHE.

ANGLE3 specifies the angle parameter used by the model PLANETAR.

C.AXIS specifies the central axis length used by the models CONICAL and PLANETAR.

P.AXIS specifies the planetary axis length used by the models PLANETAR and CONICAL.

DIST.PL specifies the distance from wafer to planetary axis used by the model PLANETAR.

SIGMA.DEP specifies the surface diffusion parameter used by the models UNIDIRECT, DUALDIRECT, HEMISPHERIC, PLANETARY, CONICAL, MONTE1, and MONTE2.

SIGMA.0 and **SIGMA.E** specify pre-exponential coefficient and activation energy of temperature dependent surface diffusion. The temperature is specified on the DEPOSIT command.

SMOOTH.WIN and **SMOOTH.STEP** specifies a window size in microns and a number of smoothing passes for the simple geometric deposit smoothing algorithm.

MCSEED specifies a seed to be used for random number generation in the Monte Carlo deposit models: MONTE1 and MONTE2.

STICK.COEF specifies the sticking coefficient for the MONTE1 model. Unitless, which must be between 0.0 and 1.0.

Examples

The following statement defines a machine named TEST that deposits silicon nitride with a rate of 1500 A/minute using the CVD model with step coverage of 75%.

```
RATE.DEPO   MACHINE=TEST   NITRIDE   DEP.RATE=1500   A.M   CVD
STEP.COV=.75
```

For more examples, see DEPOSIT.

6.49 RATE.DEVELOP

`RATE.DEVELOP` sets development rate and exposure bleaching parameters for each type of photoresist in Optolith.

Note: This statement is not available in Athena 1D.

Syntax

```
RATE.DEVELOP
[NAME.RESIST=<c>] [G.LINE|H.LINE|I.LINE|DUV.LINE|LAMBDA=<n>
[A.DILL=<n>] [B.DILL=<n>] [C.DILL=<n>]
[E1.DILL=<n>] [E2.DILL=<n>] [E3.DILL=<n>
[RMAX.MACK=<n>] [RMIN.MACK=<n>] [MTH.MACK=<n>] [N.MACK=<n>]
[RO.TREFONAS=<n>] [Q.TREFONAS=<n>
[RO.HIRAI=<n>] [RC.HIRAI=<n>] [ALPHA.HIRAI=<n>
[R1.KIM=<n>] [R2.KIM=<n>] [R3.KIM=<n>] [R4.KIM=<n>] [R5.KIM=<n>]
[R6.KIM=<n>] [R7.KIM=<n>] [R8.KIM=<n>] [R9.KIM=<n>] [R10.KIM=<n>]
[C0.EIB=<n>] [C1.EIB=<n>] [C2.EIB=<n>] [C3.EIB=<n>]
[DIX.0=<n>] [DIX.E=<n>]
```

Description

This command sets the development rate parameters and exposure parameters for each type of photoresist. These statements can be entered into the *athenamod* file, so that the parameters are loaded each time Athena is started.

NAME.RESIST is the photoresist name for this set of parameters.

G.LINE, H.LINE, I.LINE, DUV.LINE, and **LAMDBA** specify the wavelength for each set of photoresist parameters.

A.DILL, B.DILL, and **C.DILL** are the A, B, and C constants for the Dill exposure model.

E1.DILL, E2.DILL, and **E3.DILL** defines the `E1`, `E2`, or `E3` parameter for Dill's development rate function. These parameters are dimensionless.

RMAX.MACK, RMIN. MACK, MTH.MACK, and **N.MAC**K are the constants for the Mack development model. `RMAX.MACK` specifies the development rate of the fully exposed resist. `RMAX.MACK` must be specified in microns/sec. `RMIN.MACK` specifies the development rate of the unexposed resist. `RMIN.MACK` must be specified in microns/sec. `MTH.MACK` is the threshold normalized PAC concentration. `MTH.MACK` is dimensionless. `N.MACK` specifies the developer sensitivity. `N.MACK` is dimensionless.

R0.TREFONAS and **Q.TREFONAS** are constants for the Trefonas development model. `R0.TREFONAS` specifies a development rate constant. `R0.TREFONAS` must be specified in microns/sec. `Q.TREFONAS` specifies a development rate constant.

R0.HIRAI, RC.HIRAI, and **ALPHA.HIRAI** are constants for the Hirai development model. R0.HIRAI specifies the development rate of the fully exposed resist material. R0.HIRAI must be specified in microns/sec. RC.HIRA specifies a development rate for unexposed resist. RC.HIRAI must be specified in microns/sec. ALPHA.HIRAI specifies a dimensionless reaction constant.

R1.KIM, R2.KIM, R3.KIM, R4.KIM, R5.KIM, R6.KIM, R7.KIM, R8.KIM, R9.KIM, and **R10.KIM** are constants for the Kim development model. R1.KIM corresponds to the dissolution rate of the resist material if it has been fully exposed, that is if all the PAC has been decomposed. R1.KIM must be expressed in microns/sec. R2.KIM corresponds to the dissolution rate of the unexposed resist material. R2.KIM must be expressed in microns/sec. R3.KIM corresponds to the dissolution sensitivity of the resist material. R3.KIM is dimensionless. R4.KIM corresponds to a specific depth into the resist film for surface retardation effects. R4.KIM must be specified in microns. R5.KIM describes extraordinary retardation effects. R5.KIM is dimensionless, positive, and less than one. R6.KIM describes extraordinary retardation effects. R6.KIM is dimensionless, positive, and less than one. R7.KIM describes extraordinary retardation effects. R7.KIM is dimensionless, positive, and less than one. R8.KIM describes extraordinary retardation effects. R8.KIM is dimensionless, positive, and less than one. R9.KIM describes extraordinary retardation effects. R9.KIM is dimensionless, positive, and less than one. R10.KIM describes extraordinary retardation effects. R10.KIM is dimensionless, positive, and less than one.

CO.EIB, C1.EIB, C2.EIB, and C3.EIB are the parameters for the Eib development model.

DIX.0 and **DIX.E** specify pre-exponential constant in cm^2/sec and activation energy in eV for diffusion of photoactive compound that are used in the post exposure bake.

Examples

The following statement defines the Dill development parameters for a user-defined resist called SECRETX.

```
RATE.DEVELOP NAME.RESIST=TEST E1.DILL=1 E2.DILL=0.5 E3.DILL=.003
```

For more examples, see EXPOSE, BAKE, and DEVELOP.

6.50 RATE.DOPE

RATE.DOPE specifies the enhancement parameters for dopant enhanced etching in Elite.

Note: This statement is not available in Athena 1D.

Syntax

```
RATE.DOPE

MACHINE = <c>  MATERIAL I.IMPURITY

[ENH.MAX = <n>] [ENH.SCALE = <n>] [ENH.MINC = <n>]
```

Description

This statement is used to define dopant enhanced etching and can be applied to an etch machine defined using the RATE.ETCH statement.

Note: Dopant enhanced etching is not applicable to MC.PLASMA etch model.

MACHINE specifies the machine name for which the dopant enhanced model to be applied.

MATERIAL specifies material in which the dopant enhanced model to be used (see Section 6.2.9 “Standard and User-Defined Materials” for the list of materials).

IMPURITY, INTERST, VACANCY, SXX, SYY, and **SXY** specify impurity (or other solution) which concentration is used in the dopant enhanced etching model (see Section 6.2.10 “Standard Impurities” for the list of impurities).

ENH.MAX specifies the maximum enhancement.

ENH.MINC specifies the solution value below which enhancement decays.

ENH.SCALE specifies the spread of the enhancement over solution values (i.e., how quickly the enhancement factor reach its maximum).

Note: For exponentially varying solutions, e.g., oxidation stress and dopant concentrations, both C and ENH.MINC are taken to be log base 10 of their respective value.

6.51 RATE.ETCH

RATE.ETCH specifies the etch rate parameters for a machine, which is used in a subsequent ETCH statement in Elite.

Note: This statement is not available in Athena 1D.

Syntax

```
RATE.ETCH

MACHINE=<c> MATERIAL | NAME.RESIST=<n>

WET.ETCH | RIE | PLASMA | MC.PLASMA

A.H | A.M | A.S | U.H | U.M | U.S | N.M

[DIRECTIONAL=<n>] [ISOTROPIC=<n>] [CHEMICAL=<n>] [DIVERGENCE=<n>]

[PRESSURE=<n>] [TGAS=<n>] [TION=<n>] [VPDC = <n>] [VPAC=<n>]

[LSHDC=<n>] [LSHAC=<n>] [FREQ=<n] [MGAS=<n>] [MION=<n>] [QIO=<n>]

[QCHT= <n>]

[CHILD.LANGM|COLLISION|LINEAR|CONSTANT] [IONS.ONLY]

[NPARTICLES=<n>] [ENERGY.DIV = <n>] [OUTF.TABLE = <<n>>]

[OUTF.ANGLE = <c>]

[ER.LINEAR|ER.INHIB|ER.COVERAGE|ER.THERMAL]

[K.I=<n>] [K.F=<n>] [K.D=<n>] [SPARAM=<n>] [THETA=<n>]

[IONFLUX.THR=<n>] [MAX.IONFLUX=<n>] [MAX.CHEMFL=<n>] [MAX.DEPOFL=<n>]

[ION.TYPES = <n>] [MC.POLYMPT = <n>] [MC.RFLCTDIF = <n>]

[MC.ETCH1 = <n>] [MC.ETCH2 = <n>] [MC.ALB1 = <n>] [MC.ALB2 = <n>]

[MC.PLM.ALB = <n>] [MC.NORM.T1 = <n>] [MC.NORM.T2= <n>]

[MC.LAT.T1 = <n>] [MC.LAT.T2= <n>] [MC.ION.CU1= <n>]

[MC.ION.CU2= <n>]

[MC.PARTS1 = <n>] [MC.PARTS1 = <n>] [MC.ANGLE1=<n>]
[MC.ANGLE2=<n>]
```

Description

This statement is used to define parameters and the machine name for one of four etch models available in Elite.

MACHINE specifies the machine name for the RATE.ETCH statement.

MATERIAL specifies material for which parameters of the etch machine to be applied (see Section 6.2.9 “Standard and User-Defined Materials” for the list of materials).

NAME.RESIST specifies the name of photoresist to be etched.

WET.ETCH,**RIE, PLASMA,** and **MC.PLASMA** specify a particular model for the machine definition.

Parameters used for RIE and WET.ETCH models

A.H, A.M, A.S, U.H, U.M, U.S, and **N.M** specifies that the etch rates are in Angstroms per hour, Angstroms per minute, Angstroms per second, microns per hour, microns per minute, microns per second, and nanometers per minute respectively.

DIRECTIONAL specifies the directional component of the etching rate used by the `RIE` model. The ionic etch rate is the contribution of the ions to the chemically oriented etching mechanisms. The ions are assumed to have an anisotropic angular distribution specified by divergence parameter.

ISOTROPIC specifies the isotropic etch rate used by the `WET.ETCH` and `RIE` models. The isotropic etch rate is the contribution of thermal atoms, radicals, and molecules coming out of the plasma. These are assumed to have an isotropic angular distribution. Therefore, the isotropic etching may lead to an underetching of the mask.

CHEMICAL, DIVERGENCE: `CHEMICAL` is the etch rate in the `RIE` model normal to the ion beam when the `DIVERGENCE` is specified as non-zero. `DIVERGENCE` specifies the beam divergence used by the `RIE` model. The angular distribution of the ions coming down to the wafer is Gaussian.

Parameters used for Plasma Etch Model

PRESSURE specifies the plasma etcher reactor pressure. Units are mTorr. Default 50 mTorr.

TGAS specifies the plasma etcher reactor gas temperatures. Units are ^{o}K. Default is 300 ^{o}K.

TION specifies the plasma etcher reactor ion temperatures. Units are ^{o}K. Default is 300 ^{o}K.

VPDC specifies the DC bias in the plasma sheath. Units are V. Default is 32.5 V.

VPAC specifies the AC voltage in the sheath-bulk interface. Units are V. Default is 32.5 V.

FREQ specifies frequency of the AC current Units are Mhz. Default is 13.6 Mhz.

LSHDC specifies the mean sheath thickness. Units are mm. Default is 0.005 mm.

LSHAC specifies the AC component of the sheath thickness. Units are mm. Default is 0.0.

MGAS specifies the atomic mass of the gas atoms. Default is 40.

MION specifies the atomic mass of the plasma ions. Default is 40.

QIO specifies the momentum transfer cross-section. Units are m^{2}. Default is 1.7e-19

QCHT specifies the charge exchange cross-section. Units are m^{2}. Default is 2.1e-19.

CHILD.LANG, **COLLISION**, **LINEAR**, and **CONSTANT** specify a model used in calculation of the voltage drop in the plasma sheath. Default is `LINEAR`.

IONS.ONLY specifies that neutrals to be ignored in plasma simulation. Default is false

NPARTICLES specifies number of particles used for Monte Carlo calculation of the ion flux coming from plasma. Default is 10,000.

ENERGY.DIV specifies number of energy divisions used for calculation of the plasma ion flux. Default is 50.

OUTF.TABLE specifies the name of an output file in which complete table of simulated plasma ions and neutral distributions is saved. The table cannot be loaded using TonyPlot. The meanings of the columns in the table are:

- **i -** index of energy from `0` to `Nrow=ENERGY.DIV -1`, where `ENERGY.DIV` is the number of energy divisions specified in the `RATE.ETCH` statement (default is 50).
- **k -** index of angle from `0` to `Ncol=14`. The interval [0, 90] degrees is divided into 15 intervals and each of them are divided into 4 sub-intervals each of 1.5 degrees wide.
- **Cergy<n> -** number of ions in each sub-intervals, `n=0. .3`.
- **Cergyn<n> -** number of neutrals in each sub-intervals, `n=0. .3`.
- **Angadd -** the sum of `Cergy<n>` and `Cergyn<n>` for each (`i`,`k`) pair. It corresponds to the energy-angle distribution of the particles.
- **Angtot -** the sum of `Angadd` for each `k`, it corresponds to the angle distribution of particles.
- **Erel -** the result of `i/Nrow +1/(2*Nrow)` corresponding to the medium energy in eV of the 50 normalized intervals `[0.0, 0.02]`, `[0.02, 0.04] . . . [0.98, 1.0]`.

OUTF.ANGLE specifies the name of an output file in which energy-angular ion flux distribution is saved. The distribution can be plotted using TonyPlot.

ER.LINEAR, ER.INHIB, ER.COVERAGE, and **ER.THERMAL** specify surface kinetics model to be used: Simple linear, Adsorbed inhibiting layer, threshold coverage, and thermal spike models, correspondingly. Default is `ER.LINEAR`.

K.I specifies the plasma etch rate linear coefficient related to the ion flux.

K.F specifies the plasma etch rate linear coefficient related to the chemical flux.

K.D specifies the plasma etch rate linear coefficient related to the deposition flux.

SPARAM specifies S-parameter of threshold coverage and thermal spike models.

THETA specifies theta parameter of threshold coverage and thermal spike models.

IONFLUX.THR specifies the flux threshold value below which the flux is not considered for etching. Default is 0.0.

MAX.IONFLUX specifies a multiplier for ion flux generated by the plasma etching machine. Default is 1.0.

MAX.CHEMFL specifies a multiplier for chemical flux generated by the plasma etching machine. Default is 1.0.

MAX.DEPOFL specifies a multiplier for deposition flux generated by the plasma etching machine. Default is 1.0.

Parameters used for Monte Carlo Plasma Etch Model

ION.TYPES specifies the number of different ions in etching plasma.

MC.POLYMPT specifies the number of MC simulated polymer particles normalized to the volume of the ejected material.

MC.RFLCTDIF specifies the reflection diffusiveness. 1 corresponds to completely diffusive reflection, 0 corresponds to ideal mirror reflection.

MC.ETCH1 specifies the etch rate parameter for the first type of ions, unitless.

MC.ETCH2 specifies the etch rate parameter for the second type of ions, unitless.

MC.ALB1 specifies the reflection parameter for the first type of ions, unitless. This coefficient can vary from 0 (no reflection) to 1 (100% reflection).

MC.ALB2 specifies the reflection parameter for the second type of ions, which is unitless. This coefficient can vary from 0 (no reflection) to 1 (100% reflection).

MC.PLM.ALB specifies the reflection parameter for polymer particles, which is unitless. This coefficient can vary from 0 (no reflection) to 1 (100% reflection).

MC.NORM.T1 specifies the plasma normal temperature for the first type of ions, which is unitless.

MC.NORM.T2 specifies the plasma normal temperature for the second type of ions, which is unitless.

MC.LAT.T1 specifies the plasma lateral temperature for the first type of ions, which is unitless.

MC.LAT.T2 specifies the plasma lateral temperature for the second type of ions, which is unitless.

MC.ION.CU1 specifies the plasma ion current density for the first type of ions, ions/second/cm^2.

MC.ION.CU2 specifies the plasma ion current density for the second type of ions, ions/second/cm^2.

MC.PARTS1 specifies the number of MC simulated particles for the first type of ions.

MC.PARTS2 specifies the number of MC simulated particles for the second type of ions.

MC.ANGLE1 specifies the incident angle for the first type of ions. The default is 0 (normal incidence).

MC.ANGLE2 specifies the incident angle for the second type of ions. The default is 0 (normal incidence).

Wet Etch Example

The following example defines an etch machine that attacks silicon with wet etch characteristics and an etch rate of 0.1 micron/minute.

```
RATE.ETCH MACHINE=TEST SILICON WET.ETCH ISOTROPIC=.1 U.M
```

Monte Carlo Plasma Etch Example

The following statement defines parameters of Monte Carlo Plasma Etch machine as well as etching characteristics of Silicon associated with this machine.

```
RATE.ETCH MACHINE=MCETCH SILICON MC.PLASMA ION.TYPES=1 \
     MC.PARTS1=20000 MC.NORM.T1=14.0 MC.LAT.T1=2.0 \
     MC.ION.CU1=15 MC.ETCH1=1e-05 MC.ALB1=0.2 MC.PLM.ALB=0.5 \
     MC.POLYMPT=5000 MC.RFLCTDIF=0.5
```

6.52 RATE.POLISH

RATE.POLISH specifies the polishing parameters for a chemical mechanical polishing (CMP) module.

Note: This statement is not available in Athena 1D.

Syntax

```
RATE.POLISH
MACHINE = <c> MATERIAL | NAME.RESIST=<n>
[A.H|A.M|A.S|U.S|U.M|U.H|N.M]
[SOFT.RATE] [HEIGHT.FAC=<n>] [LENGTH.FAC=<n>] [KINETIC.FAC=<n>]
[MAX.HARD=<n>]  [MIN.HARD=<n>]
[ISOTROPIC=<n>]
```

Description

This command sets the parameters for the CMP machine used in the POLISH statement. The parameters must be set for each material to be polished. There are two polish models, hard and soft, that can be used together or separately. Define these models by specifying their parameters.

MACHINE specifies the machine name.

MATERIAL specifies material for which parameters of the CMP machine to be applied (see Section 6.2.9 “Standard and User-Defined Materials” for the list of materials).

NAME.**RESIST** is the user-defined photoresist to be polished.

A.H, A.M, A.S, U.H, U.M, U.S, and **N.M** specifies that the rates are in Angstroms per hour, Angstroms per minute, Angstroms per second, microns per hour, microns per minute, microns per second, and nanometers per minute respectively.

SOFT.RATE is the rate for the soft polish model.

HEIGHT.**FAC** is the vertical deformation scale for the soft polish model. Units are microns.

LENGTH.**FAC** is the horizontal deformation scale for the soft polish model. Units are microns.

KINETIC.**FAC** is the Kinetic factor (soft polish model). The vertical polish rate increases as the surface becomes more vertical.

MAX.**HARD** is the maximum rate for the hard polish model. Corresponds to a pattern factor of zero.

MIN.**HARD** is the minimum rate for the hard polish model. Corresponds to a pattern factor of one.

ISOTROPIC specifies the isotropic etch rate used by the POLISH model.

Examples

The following statements describe a polishing machine named CMP for nitride and oxide.

```
RATE.POLISH MACHINE=cmp NITRIDE SOFT=4 N.M HEIGHT.FAC=0.02 \
     LENGTH.FAC=80 KINETIC.FAC=10 \
RATE.POLISH MACHINE=cmp OXIDE SOFT=25 HEIGHT.FAC=0.02 \
     LENGTH.FAC=30 KINETIC.FAC=10
```

For more examples, see `POLISH` and `RATE.ETCH`.

6.53 REGION

REGION specifies a material to be assigned to a defined mesh region.

Note: Typically, the REGION statement is not required since initial substrate material is specified on the INIT statement.

Note: This statement is not available in Athena 1D.

Syntax

```
REGION
MATERIAL [XLO=<c>] [YLO=<c>] [XHI=<c>] [YHI=<c>]
```

Description

This command specifies the material in a rectangular mesh. REGION statements should follow LINE statements. Material must be specified for every triangle in a mesh. Therefore for each rectangular mesh, there must be at least one REGION statement specifying, which material is included within the mesh. If you do not include REGION statement between the LINE statement and the INITIALIZE statement, you can define the material on the INITIALIZE statement.

MATERIAL specifies the material in a region (see Section 6.2.9 “Standard and User-Defined Materials” for the list of materials).

XLO, YLO, XHI, and **YHI** specifies the bounds of the region rectangle. The value <string> should be one of the tags created in a preceding LINE statement.

Examples

The following REGION statement specifies silicon as the material for the entire mesh.

```
LINE X LOC=0 SPA=1 TAG=LEFT
LINE X LOC=1 SPA=0.1
LINE X LOC=2 SPA=1 TAG=RIGHT
LINE Y LOC=0 SPA=0.02 TAG=SURF
LINE Y LOC=1 SPA=0.1 TAG=BACK
REGION SILICON XLO=LEFT XHI=RIGHT YLO=SURF YHI=BACK
INIT
```

Note: If you do not use REGION statement and no material appears on the INIT statement, then Athena assumes Silicon is the starting material. If you do not specify enough regions to describe the material at every part of the grid, it may not be detected until the execution of a subsequent command.

For more examples, see INITIALIZE.

6.54 RELAX

RELAX loosens the grid in an Athena mesh.

Note: This statement is not available in Athena 1D.

Syntax

```
RELAX
[MATERIAL]  [X.MIN=<n>] [X.MAX=<n>] [Y.MIN=<n>] [Y.MAX=<n>]
[DIR.X|DIR.Y]  [SURFACE] [DX.SURF=<n>
```

Description

This statement allows you to increase grid spacing. You can place the RELAX statement anywhere within the input file. RELAX commands, however, are ignored if Athena is in 1D mode. The RELAX statement also includes an algorithm for relaxing grid on the surface of the simulation structure.

MATERIAL specifies that RELAX will only apply to the regions of this MATERIAL (see Section 6.2.9 “Standard and User-Defined Materials” for the list of materials). If MATERIAL is not specified, RELAX will be applied to all materials in the box.

X.MIN, X.MAX, Y.MIN, and **Y.MAX** specifies the corner coordinates of the RELAX box. Units are microns. Default is bounding box of the current simulation structure.

DIR.X or **DIR.Y** allow the direction of the grid relax to be controlled. DIR.X and DIR.Y are true by default (i.e., when the RELAX statement is encountered, the grid is relaxed in both directions by default). When DIR.X or DIR.Y is selected as false (i.e., DIR.X=F or DIR.Y=F), then the grid is only relaxed in the direction that is left as true.

SURFACE specifies to relax the surface grid.

DX.SURF specifies a minimum size for surface segments.

Examples

```
RELAX SILICON X.MAX=1 Y.MIN=0
```

This statement changes a grid over a rectangular area in silicon from the left side of a structure to 1 and from y=0 to the bottom of the silicon.

Note: RELAX will not make any changes to a grid if obtuse triangles would result from the mesh relaxation. Consequently, RELAX will typically only work on meshes that were initially defined using LINE statements in Athena. For other structures, you can use DevEdit. For more information about DevEdit, see the DevEdit User’s Manual.

6.55 SELECT

SELECT selects the variable for printing using the PRINT.1D statement.

Note: This command has been superseded for use with PRINT.1D by the EXTRACT command. See the DeckBuild User's Manual for more information.

Syntax

```
SELECT
[Z=<c>] [TEMPERATURE=<n>]
```

Description

SELECT specifies the variable that will be printed by the PRINT.1D statement. You can only use one variable at a time. Each SELECT statement overrides any previous statements.

Z is set equal to the selected variable. The operators *, /, +, -, ^ all work as standard algebraic operators would. Z can be set to any of the vector variables shown on the next page.

Table 6-4 Select Operator Variables.

Vector Variables	Description
ANTIMONY	antimony concentration
ARSENIC	arsenic concentration
BORON	boron concentration
CI.STAR	equilibrium interstitial concentration
CV.STAR	equilibrium vacancies concentration
DOPING	net active concentration
ELECTRONS	electron concentration
INTERSTITIAL	interstitial concentration
NI	intrinsic electron concentration
OXYGEN	oxygen concentration
PHOSPHORUS	phosphorus concentration
Sxx, Sxy, Syy	components of stress in rectangular coordinates
TRAP	unfilled interstitial trap concentration
VACANCY	vacancy concentration
X	x coordinates
Y	y coordinates
X.V	x velocity
Y.V	y velocity

Table 6-5 Select Functions.

Function	Description
abs	absolute value
active	active portion of the specified dopant
erf	error function
erfc	complimentary error function
exp	exponential
gradx	numerically differentiates the argument with respect to x location
grady	numerically differentiates the argument with respect to y location
log	logarithm
log10	logarithm base 10
<mat1>@<mat2>	returns the y value of the interface between <mat1> and <mat2> along a vertical slice at the given location
scale	scales the value given by the maximum value
sqrt	square root

TEMPERATURE specifies the temperature at which expressions are evaluated. It defaults to the last diffusion temperature. This parameter has to be specified (by default or explicitly) when printing a net active concentration or preparing a Atlas structure file.

Examples

The following example will choose the base 10 logarithm of the arsenic concentration as the `PRINT.1D` variable.

```
SELECT Z=LOG10(ARSEN)
```

The following chooses the difference between the phosphorus and an analytic profile as the `PRINT.1D` variable.

```
SELECT Z=(PHOS - 1.0E18 * EXP (Y * Y / 1.0E-8))
```

The following chooses the excess vacancy interstitial product as the `PRINT.1D` variable.

```
SELECT Z=(INTER * VACAN - CI.STAR * CV.STAR)
```

Note: When using log or log10 functions, make sure the argument is positive and non-zero. For example, always use log10(abs(doping)+1).

For more examples, see the `PRINT.1D`

6.56 SET

SET specifies strings or numbers for variable substitution.

Note: This commands executed under DeckBuild and is documented fully in the DeckBuild User’s Manual.

Syntax

```
SET

variable = <value>
```

Numerical Variable Example

The following statement defines a variable and performs an expression on it for use later within the Athena processing syntax.

```
SET MYDOSE=1e13

SET HALFMYDOSE=$"MYDOSE"/2

IMPLANT BORON DOSE=$"HALFMYDOSE"
```

String Variable Example

The following uses SET to define a string variable. The saved file will be called mosfet_fred.str.

```
SET MYNAME=fred

STRUCTURE OUTFILE=mosfet_$"myname".str
```

For more examples, see EXTRACT.

6.57 SETMODE

SETMODE specifies execution mode parameters.

Syntax

```
SETMODE
[NOEXECUTE | ECHO]
```

Description

This command turns on the following execution mode parameters. The UNSET statement allows the same parameters to be turned off.

NOEXECUTE puts all entered statements into a check only mode. If this flag is on, Athena will only check the legality of the input syntax and not execute any statements.

ECHO instructs Athena to echo all input lines to the run-time output. Note that in DeckBuild, this is not required as all lines are echoed to the bottom run-time window or run-time output file by default.

Examples

The following statement causes Athena to echo each command it receives.

```
SETMODE  ECHO
```

For more examples, see UNSETMODE.

Note: The parser does not recognize abbreviated forms of these commands. It requires that you enter NOEXECUTE and ECHO verbatim.

6.58 SILICIDE

`SILICIDE` specifies the silicidation coefficients.

Syntax

```
SILICIDE

[SILICON|POLYSILICON|TUNGSTEN|TITANIUM|PLATINUM|COBALT|

WSIX|TISIX|PTSIX|COSIX|MATERIAL=<c>]

[/SILICON|/POLYSILICO|/TUNGSTEN|/TITANIUM|/PLATINUM|/COBALT

/WSIX|/TISIX|PTSIX|/COSIX|/MATERIAL=<c>]

[MTTYPE=<c>] [/MTTYPE=<c>] [ALPHA=<n>]
```

Description

SILICON, POLYSILICON, TUNGSTEN, GAAS, TITANIUM, PLATINUM, COBALT, WSIX, TISIX, PTSIX, and **MATERIAL** specify the first material to which the parameters apply.

/SILICON, /POLYSILICO, /TUNGSTEN, /TITANIUM, /PLATINUM, /COBALT, / WSIX, /TISIX, /PTSIX, / COSIX, and **/MATERIAL** specify the second material to which parameters apply.

MTTYPE specifies the type (metal or silicide) of the user-defined `MATERIAL`.

/MTTYPE specifies the type (metal or silicide) of the user-defined `/MATERIAL`.

ALPHA specifies the volume expansion ratio between `MATERIAL` and `/MATERIAL`.

Examples

The following example specifies the volume expansion between user-defined material TiSi2 and standard material titanium.

```
SILICIDE  MATERIAL=TISI2  MTTYPE=SILICIDE  /MATERIAL=TITANIUM
ALPHA=0.4
```

6.59 SOURCE

SOURCE executes statements from the specified file.

Syntax

```
SOURCE
```

```
<filename>
```

Description

SOURCE reads statements from an input file. Statements are read from the file until an end-of-file marker is found. SOURCE is especially useful for executing a large group of statements. SOURCE places the named file in the current input stream. SOURCE statements can be nested up to the limit of open file descriptors (system dependent).

Examples

The following statement causes the contents of a file named test.in to be included into the input stream.

```
SOURCE TEST.IN
```

Note: To support the use of this function when running under the VWF Automation Tools, place the file to be sourced into a directory directly visible to the simulation run, regardless where the simulator is executing.

6.60 STRESS

STRESS calculates elastic stresses.

Note: This statement is not available in Athena 1D.

Syntax

```
STRESS
[TEMP1=<n>] [TEMP2=<n>] [NEL=<n>]
```

Description

This command calculates stresses due to thin film intrinsic stress or thermal mismatch.

TEMP1 and **TEMP2** are the initial and final temperatures in ^{o}C for calculating thermal mismatch stresses.

NEL is the number of nodes per triangle element. Both linear approximation with 3 or 4 nodes and quadratic approximation with 6 or 7 nodes are allowed. Linear approximation with 3 nodes is the fastest and is recommended for very large structures. Default is quadratic approximation with 6 nodes.

Examples

The following calculates the stresses in the substrate and film arising from a nitride layer, which has an intrinsic stress of 1.4×10^{10} dynes cm^{-2} when deposited uniformly.

```
MATERIAL NITRIDE INTRIN.SIG=1.4E10
STRESS
```

The following calculates thermal mismatch stress in the whole structure as the result of a temperature change from 1000 to 100 ^{o}C.

```
STRESS TEMP1=1000 TEMP2=100
```

For more examples, see MATERIAL.

6.61 STRETCH

STRETCH stretches structures about a specified location.

Note: This statement is not available in Athena 1D.

Syntax

```
STRETCH

MATERIAL=<c>                [LENGTH=<n>]                  [X.VAL=<n>]
[Y.VAL=<n>] [STRETCH.VAL=<n>]

[SPACING=<n>] [DIVISION=<n>] [SNAP]
```

Description

This statement specifies that the device is to be stretched about a specified location. If device characterization as a function of length is of interest, the stretch function will save massive amounts of CPU time in generating multiple gate length structures. The stretch capability is also useful for power devices.

MATERIAL specifies material that defines the stretch region (see Section 6.2.9 “Standard and User-Defined Materials” for the list of materials). Default is SILICON.

LENGTH specifies the final value in microns to which the specified material region is stretched. Alternatively, you can specify X.VAL using STRETCH.VAL to specify the position of a vertical cut line and the distance to be stretched respectively. The grid spacing within the stretched region is defined either by spacing or by division.

X.VAL and **Y.VAL** specifies the horizontal or vertical position in microns at which stretch occurs. LENGTH overrides the STRETCH.VAL, X.VAL, and Y.VAL parameters. If LENGTH is specified, the cut line stretch location defaults to the center of the specified material. The default material is polysilicon.

SPACING specifies the grid spacing within the stretched region. Units are microns.

DIVISION specifies the number of grid divisions within the stretched region.

SNAP indicates that X.VAL should “snap” (change value or locate) to the nearest grid point before stretching. SNAP is recommended to minimize the potential for obtuse triangle generation. SNAP is set to true by default.

Stretch Examples

The following statement will stretch a device about the center of its polysilicon region. This device can have been a MOSFET with a polysilicon gate 1 micron long. The STRETCH command creates a 1.8 micron-long MOSFET in this case.

```
STRETCH LENGTH=1.8
```

The following example will stretch an oxide isolation structure from the x position of 2.3 microns by a value of 1.3 microns. The stretched region contains 14 grid spaces. This case can be useful for generating large isolation regions that take too long to simulate numerically.

```
STRETCH OXIDE X.VAL=2.3 DIVISIONS=14 STRETCH.VAL=1.3
```

Note: The stretch function may not be valid or physically correct in the case of very short initial structures (e.g. with RSCE effect in MOSFETs). The location selected for stretching should correspond exactly to a grid line for best results. It will provide best grid quality if the stretch location does not touch areas in which the grid has been relaxed. The `STRETCH` command often results in grid failure for complex structures and is not recommended for complex topographies. DevEdit provides a superior stretch feature for these cases.

6.62 STRIP

STRIP removes all photoresist and barrier materials.

Syntax

```
STRIP
[MATERIAL]
```

Description

MATERIAL specifies the material to be stripped (see Section 6.2.9 “Standard and User-Defined Materials” for the list of materials). If no material is specified, the STRIP command removes both photoresist and barrier materials.

Examples

The following sequence of statements deposits photoresist patterned with the mask level named CONT, etches oxide through the mask, and removes the photoresist with the STRIP statement.

```
MASK NAME="CONT"
ETCH OXIDE DRY THICK=.2
STRIP
```

This example requires the use of MaskViews.

For more examples, see MASK and ETCH.

6.63 STRUCTURE

STRUCTURE writes the mesh and solution information, aerial image information, or flips or mirrors the structure. SAVEFILE is a synonym for this statement.

Syntax

```
STRUCTURE

[OUTFILE=<c>]

[FLIP.Y] [MIRROR] [LEFT|RIGHT|TOP|BOTTOM]

[INTENSITY] [MASK] [REMOVE.GAS]  [SIGE.CONV] [TWO.DIM]
```

Description

This statement writes the entire mesh and solution set to a file. The saved data is from the current set of solution and impurity values.

OUTFILE specifies the name of the file to be written. Existing files with the same name are overwritten by newly specified files. OUT.FILE is an alias for this parameter.

FLIP.Y indicates that the structure should be flipped around the x axis. This is used to invert structures for backside processing.

MIRROR, LEFT, RIGHT, TOP, and **BOTTOM** mirrors the grid about its left or right, top or bottom boundary respectively. This is useful for turning half of a MOSFET simulationstructure into full structure for subsequent Atlas simulation. The default reflection is about the right axis.

INTENSITY specifies the aerial image intensity distribution to be saved in the output file.

MASK specifies layout mask information to be saved in the output file.

REMOVE.GAS specifies that the gas region is to be removed from the output structure. Currently, the overlaying gas region is automatically added to the structure for Monte Carlo etch and BCA implant simulations.

SIGE.CONV converts the layer of silicon that is highly doped with Ge into a Si_{1-x} Ge_x layer so it can be used in Atlas.

TWO.DIM specifies that the structure to be transformed into 2D when simulation is performed in AUTO mode (see Section 6.30 “INITIALIZE”) and the structure is still 1D.

Examples

The following statement writes the current structure to a file called test.str.

```
STRUCTURE OUTFILE=TEST.STR
```

The following statement saves an aerial image and masks calculated by Optolith to a file called test.str.

```
STRUCTURE OUTFILE=TEST.STR INTENSITY MASK
```

The following statement mirrors the structure about its left boundary.

```
STRUCTURE MIRROR LEFT
```

Note: The STRUCTURE command will only save all mesh and solution information. It will not save any defined model or machine methods. If you exit a simulator the middle of an input file, you may need to manually parse the preceding METHOD and IMPURITY commands to reinitialize specified parameters. This function is handled automatically when running under the VWF Automation Tools.

For more examples, see INITIALIZE.

6.64 SYSTEM

SYSTEM allows execution of any UNIX C-shell command within an input file

Note: The SYSTEM statement is executed by DeckBuild and is fully documented in the DeckBuild User's Manual.

Note: The SYSTEM command must be enabled using an option on the DeckBuild Main Control menu under **Category→Options.**

Examples

The following command will remove all files named test*.str before a DIFFUSE statement where the DUMP parameter is used.

```
system \rm -rf test*.str
DIFFUSE .... DUMP=1 DUMP.PREF=test
```

The SYSTEM command and UNIX commands are case sensitive.

UNIX commands can be concatenated on a single line using the semicolon (;) operator. For example, to run a third party program that reads and writes Silvaco format files with the fixed names input.str and output.str.

```
STRUCTURE OUTF=mysave.str
system mv mysave.str input.str; source myprog.exe; mv output.str
myrestart.str
INIT INF=myrestart.str
```

The UNIX re-direct symbol, >, is not supported by the system command. The UNIX echo and sed syntax can be used instead to output values or variables to a given filename. For example, to save the extracted value of the variable, $myvariable, to the file called myfile.

```
system echo $"myvariable" | sed -n "w myfile"
```

6.65 TONYPLOT

`Tonyplot` starts the graphical post-processor TonyPlot.

Note: The `Tonyplot` statement is executed by DeckBuild, which is fully documented in the deckBuild User's Manual.

Examples

All graphics in Athena is performed by saving a file and then loading the file into TonyPlot. The command

```
tonyplot
```

causes Athena to automatically save a file and plot it in TonyPlot. The TonyPlot window will appear displaying the material boundaries. Use the **Plot:Display** menu to see more graphics options.

The following command will display the `myfile.str` file.

```
tonyplot -st myfile.str
```

The following command will overlay the results of `myfile1.str` and `myfile2.str`.

```
tonyplot -overlay myfile1.str myfile2.str
```

Note: For documentation of the extensive features of TonyPlot for graphical display and analysis, consult the TonyPlot chapter of the VWF Interactive Tools User's Manual, Vol. I.

6.66 TRAP

TRAP sets the coefficients of interstitial traps.

Syntax

```
TRAP

MATERIAL [ENABLE] [TOTAL=<n>] [FRAC.0=<n>] [FRAC.E=<n>]
```

Description

This statement allows you to specify values for coefficients of the interstitial traps. The statement allows coefficients to be specified for each of the materials. Athena has default values only for silicon. Polysilicon parameters default to those for silicon.

MATERIAL specifies the material for which the parameters apply (see Section 6.2.9 “Standard and User-Defined Materials” for the list of materials).

ENABLE indicates that traps should be enabled in the material specified.

TOTAL specifies the total number of traps, in cm^{-3}. The default for silicon is 5.0×10^{17} cm^{-3}. This value is appropriate for Czochralski silicon material.

FRAC.0 and **FRAC.E** allows the specification of the equilibrium empty trap ratio.

Examples

The following statement turns on interstitial traps and sets the total to 5.0×10^{17} and the fraction to a half.

```
TRAP SILICON TOTAL=5.0E17 FRAC.0=0.5 FRAC.E=0.0 ENABLE
```

Note: The trap concentration depend upon the thermal history of the wafer, starting material, stress and temperature. This history is not considered in the trap model in Athena.

For more examples, see INTERSTITIAL and VACANCY.

6.67 UNSETMODE

UNSETMOD unsets execution mode parameters defined in the SETMODE statement.

Note: When SET variable=value is used in DeckBuild, it is impossible to UNSETMODE the variable

Syntax

```
UNSETMODE
[NOEXECUTE | ECHO]
```

Description

This command turns off the following execution mode parameters. The SETMODE statement allows you to turn on the same parameters.

NOEXECUTE puts all entered statements into a check only mode. If this flag is on, Athena will only check the syntax of the input commands and not actually run them.

ECHO instructs Athena to echo all input lines.

Examples

The following turns off statement echoing.

```
UNSETMODE ECHO
```

Note: UNSET is a synonym for this command.

Note: The parser does not recognize abbreviated forms of these commands. It requires that you enter NOEXECUTE and ECHO verbatim.

For more examples, see SETMODE.

Appendix A
C-Interpreter

A.1 C-Interpreter Overview

Athena has a C language interpreter (C-Interpreter) that allows you to modify the models contained in Athena. In order to use this capability, write a C language analytical function describing the model. If you're not familiar with the C language, then we suggest that you read any of the popular C language books such as [122].

The function arguments of the C-Interpreter functions are fixed in Athena. Thus, you need to make sure that the arguments and return values match those expected by Athena. To help you, a set of templates for functions available for the current release of Athena can be obtained by typing:

```
athena -T filename
```

The `filename` is the name of the file where you want the template to be copied. You can also obtain the C-Interpreter templates by selecting **Commands→Templates...** in Deckbuild. The following example shows how to use the C-Interpreter function `get_damage_values` to modify the default Plus One Model (See Chapter 3 "SSuprem4 Models") for interstitials generated during ion implantation.

A.2 Example

```
/*
  Template for  the C-Interpreter function for defect formation
  during ion implantation
*/
void get_damage_values
     (
     /* input parameters */
     int imp, /* impurity index: As - 2; P - 3; Sb - 4; B - 5;
etc. */
     int mater,  /* material index: Si - 3 */
     double x,   /* x-coordinate in micron*/
     double y,   /* y-coordinate in microns */
     double implanted_conc,/* implanted concentration in 1/cm**3
*/
     double implanted_dam,  /*accumulated damage in eV/cm**3),
                           do not use without Monte Carlo BCA */
     /* return parameters */
     double *I_val,      /* Interstitial concentration */
     double *V_val,      /* Vacancy concentration */
     double *CL_val,     /* 311-Cluster concentration */
     double *DL_val      /* Dislocation loops concentration */
     )
     {
       if ( mater == 3 ) /* only in Silicon */
          {
          /* The function modifies +1 interstitial generation model.
         The interstitials are generated only in unamorphized layer
          where damage is less than 0.1 of atomic density of Si */
          if ( implanted_dam < 5e21)
            *I_val = implanted_conc;
    }
          return;
     }
```

The function receives the following input values:

- material index,
- *x* and *y* coordinates of the point in the structure where damage is calculated,
- implant concentration at this point,
- implant damage but only when the Monte Carlo BCA model is used for the current implant calculations.

This allows you to return the values of interstitial and vacancy concentration, and concentrations of {311} clusters and dislocation loops. If one or few of return values are not modified in the function the corresponding concentrations will remain unchanged after the implant.

The function then needs to stored as a file (i.e., `damage.lib`). The model stored in the function can then be activated by specifying the `DAM.MOD=DAMAGE.LIB` parameter in the `IMPLANT` statement.

Note: Prior to Athena version 5.4.0.R, `DAMAGEMOD.FN` (`DAM.MOD`) was a parameter in the `MOMENTS` statement

When you execute the `IMPLANT` statement using analytical or Monte Carlo models, the specified C-Interpreter function will be used in place of the build in function.

Appendix B
Default Coefficients

This appendix contains the list of impurity and material default coefficients, default model parameters, and other parameters used in Athena calculations. Most of these coefficients are initialized in the *athenamod* file. The file *athenamod* will appear when you select **Commands**→**Models** in DeckBuild while Athena is the current simulator. Almost all of these coefficients can be modified to match measured results.

You should check the contents of *athenamod* for updates to default values that may be more current than those shown in the following lists.

B.1 Oxidation Rate Coefficients

B.1.1 Dry Ambient For <111> Orientation

These parameters are from the bibliography reference [37].

Table B-1 Parabolic and Linear Rate Constants for Dry Ambient

Parameter	Value
`PAR.H.0(µm2/min)`	`12.8667`
`PAR.H.E (eV)`	`1.23`
`P.BREAK (°C)`	`0.`
`LIN.H.0 (µm/min)`	`1.038×105`
`LIN.H.E (eV)`	`2.0`
`L.BREAK (°C)`	`0`

B.1.2 Wet Ambient for <111> Orientation

These parameters are from the bibliography reference [39].

Table B-2 Parabolic and Linear Rate Constants for Wet Ambient

Parameter	Value
`PAR.L.0 (µm2/min)`	`283.333`
`PAR.L.E (eV)`	`1.17`
`PAR.H.0 (µm2/min)`	`7.0`
`PAR.H.E (eV)`	`0.78`
`P.BREAK (°C)`	`950.`
`LIN.L.0 (µm/min)`	`3.45×104`
`LIN.L.E (eV)`	`1.6`
`LIN.H.0 (µm/min)`	`2.95×106`
`LIN.H.E (eV)`	`2.05`
`L.BREAK (°C)`	`900.`

B.1.3 Orientation Factors For Linear Coefficients (both Ambients)

Table B-3 Linear Coefficient Orientation Factors

Orientation	Value
For <100> orientation (unitless)	ORI.FAC = 0.595
For <110> orientation (unitless)	ORI.FAC = 0.833
For <111> orientation (unitless)	ORI.FAC = 1.0

B.1.4 Pressure Dependence

For Dry Oxidation, DRY: L.PDEP = 0.75 and P.PDEP = 1.0. For Wet Oxidation, WET: L.PDEP = 1.0 and P.PDEP = 1.0.

Table B-4 Thin oxide coefficients (only for dry ambient)

Orientation	THINOX.0 (μ2/min)	THINOX.E (eV)	THINOX.L (μ)	THINOX.P [14]
<111>	5.87×106	2.32	0.0078	1.0
<110>	5.37×104	1.80	0.0060	1.0
<100>	6.57×106	2.37	0.0069	1.0

B.1.5 Chlorine Dependence

Table B-5 Chlorine dependence of dry coefficients for three temperatures: 900, 1000, 1100ºC

HCL.PC	HCL.LIN			HCL.PAR		
	900	1000	1100	900	1000	1100
0.0	1.0	1.0	1.0	1.0	1.0	1.0
1.0	1.75	1.25	1.621	1.083	1.658	1.355
3.0	1.75	1.486	2.207	1.25	1.840	1.490
5.0	1.75	1.486	2.207	1.444	2.075	1.641
7.0	1.75	1.486	2.207	1.639	2.332	1.816
10.0	1.75	1.486	2.207	2.028	2.759	2.102

B.1.6 Doping Dependence Of Oxidation Rate

Table B-6 Doping Dependence of Oxidation Rate

Parameter	Value
BAF.EBK	241.6
BAF.PE	0.46
BAF.PPE	1.0
BAF.NE	0.145
BAF.NNE	0.62
BAF.K0	2.6e3
BAF.KE	1.1

See the bibliography reference [40] for more details.

B.1.7 Coefficients for the Analytical Guillemot Model

Table B-7 Coefficents for the Analytical Guillemot Model

Parameter	Value
Spread	1.0
INITIAL	0.002
MASK.EDGE	-200
ERF.Q	0.05
ERF.DELTA	0.04
ERF.LBB	(8.25e-3*(1580.3 - Tox)* (Eox ^0.67)*(eox ^0.3)*exp (-(en -0.08)^2)/0.06))
ERF.H	(402*(0.445 - 1.75*en)*exp (- Tox /200)

See the bibliography reference [43] for more details.

B.1.8 Numerical Oxidation Coefficients

For Dry Oxidation, HENRY.COEFF=5E16,THETA= 2.2E22, TRN.0= 1E+3.

For Wet Oxidation, HENRY.COEFF= 3e19,THETA= 2.2e22,TRN.0= 1e+6.

B.1.9 Stress-dependent Growth Model Coefficients

```
Vc = 78,  Vr = 9.7, Vd = 1.9, Vt = 0.0,  Dlim = 1.0
```

B.1.10 Linear Coefficients Of Thermal Expansion

These parameters can be accessed by specifying the LCTE parameter in the MATERIAL statement.

Table B-8 Linear Coefficients of Thermal Expansion

Parameter	Value
SILICON	LCTE = 3.052e-6 + 2 * 6.206e-10 * (T - 293)
OXIDE	LCTE = 1.206e-7 + 2 * 2.543e-10 * (T - 293)
ALUMINUM	LCTE = 2.438e-5 + 2 * 6.660e- 9 * (T - 293)
NITRIDE	LCTE = 3.0e-6
POLY	LCTE = 3.052e-6 + 2 * 6.206e-10 * (T - 293)

B.1.11 Volume Expansion Ratio

The volume expansion ratio, ALPHA, can be set in the OXIDE statement.

Table B-9 Volume Expansion Ratio

Ratio	Value
silicon/oxide (unitless)	0.44
poly/oxide (unitless)	0.44
Other combinations (unitless)	1.00

B.2 Impurity Diffusion Coefficients

Table B-10 Impurity Diffusion Coefficients

Parameter	Antimony	Arsenic	Boron	Phosphorus
Silicon [124]				
DIX.0 (cm^{2}/s)	0.214	8.0	0.037	3.85
DIX.E (eV)	3.65	4.05	3.46	3.66
DIP.0 (cm^{2}/s)	0.0	0.0	0.72	0.0
DIP.E (eV)	0.0	0.0	3.46	0.0
DIM.0 (cm^{2}/s)	15.0	12.8	0.0	4.44
DIM.E (eV)	4.08	4.05	0.0	4.00
DIMM.0 (cm^{2}/s)	0.0	0.0	0.0	44.2
DIMM.E (eV)	0.0	0.0	0.0	4.37
CTN.0 (cm^{2}/s)		5.19×10^{-24}		
CTN.E (eV)		0.60		
FI (unitless)	0.05	0.20	0.94	
Polysilicon				
DIX.0 (cm^{2}/s)	21.4	6.6	3.66	385.0
DIX.E (eV)	3.65	3.44	3.46	3.66
DIP.0 (cm^{2}/s)	0.0	0.0	72.0	0.0
DIP.E (eV)	0.0	0.0	3.46	0.0
DIM.0 (cm^{2}/s)	1500.0	1200.0	0.0	443.9
DIM.E (eV)	4.08	4.05	0.0	4.05
DIMM.0 (cm^{2}/s)	0.0	0.0	0.0	4420.0
DIMM.E (eV)	0.0	0.0	0.0	4.37
CTN.0 (cm^{2}/s)		$5.19 \text{ X } 10^{-24}$		
CTN.3(eV)		0.60		
Oxide				
DIX.0 (cm^{2}/s)	1.31×10^{16}	1.75	3.16×10^{-4}	7.6×10^{-3}
DIX.E (eV)	8.75	4.89	3.53	3.5
Tungsten Silicide [125]				
DIX.0 (cm^{2}/s)	2.6	2.6	1.0×10^{-4}	4.2
DIX.E (eV)	2.11	2.11	1.17	2.14
Titanium Silicide [125]				
DIX.0 (cm^{2}/s)	4.8	4.8	1.5×10^{-7}	392.0
DIX.0 (eV)	2.13	21.3	2.0	2.64
Platinum Silicide [125]				
DIX.0 (cm^{2}/s)	2.6	2.6	1.0×10^{-3}	4.2
DIX.0 (eV)	2.11	2.11	1.17	2.14

All other coefficients for refractory metals and their silicides are set to 0.0.

B.3 Impurity Segregation Coefficients

Table B-11 Impurity Segregation Coefficients

Parameter	Antimony	Arsenic	Boron	Phosphorus
Silicon/oxide				
SEG.0 (unitless)	30.0	30.0	1126.0	30.0
SEG.E (eV)	0.0	0.0	0.91	0..0
Poly/oxide				
SEG.0 (unitless)	30.0	30.0	1126.0	30.0
SEG.E (eV)	0.0	0.0	0.91	0..0
Other Impurities and Pairs of Materials				
SEG.E (eV)	0.0			

B.4 Interface Transport Coefficients

Table B-12 Interface Transport Coefficients

Parameter	Antimony	Arsenic	Boron	Phosphorus
Silicon/gas [126]				
TRN.0 (cm/s)	2.5×10^{-3}	1.5	27.9	1.5
TRN.E (eV)	1.04	1.99	2.48	1.99
Poly/gas				
TRN.0 (cm/s)	2.5×10^{-3}	1.5	27.9	1.5
TRN.E (eV)	1.04	1.99	2.48	1.99
Other Impurities and Pairs of Materials				
TRN.0 (cm/s)	1.55×10^{-7}			
TRN.E (eV)	0.0			

B.5 Solid Solubility In Silicon

Solubility can be modified for a particular temperature using the SS.TEMP and SS.CONC parameters in each of the impurity statements.

Table B-13 Solid Solubility in Silicon [127], [128]

Temperature (ºC)	Boron [cm^3]	Phosphorus [cm^3]	Antimony [cm^3]
800	3.4499×10^{19}	2.3000e^{19}	
825	4.1291×10^{19}		
850	4.9027×10^{19}	2.7943×10^{20}	

Table B-13 Solid Solubility in Silicon [127], [128]

Temperature (ºC)	Boron [cm^3]	Phosphorus [cm^3]	Antimony [cm^3]
875	5.7777×10^{19}		
900	6.7615×10^{19}	3.1585×10^{20}	3.0000×10^{19}
925	7.8610×10^{19}		
950	9.0832×10^{19}		
975	1.0435×10^{20}		
1000	1.01922×10^{20}	3.3981×10^{20}	4.0000×10^{19}
1025	1.03552×10^{20}		
1050	1.5331×10^{20}		
1075	1.7263×10^{20}		
1100	1.9356×10^{20}	3.7943×10^{20}	4.8000×10^{19}
1125	2.01613×10^{20}		
1150	2.04041×10^{20}		
1175	2.6643×10^{20}		
1200	2.9423×10^{20}		
1225	3.2387×10^{20}		
1250	3.5536×10^{20}		6.6200×10^{19}
1275	3.8876×10^{20}		

B.6 Point Defect Parameters

These parameters are for silicon only.

Table B-14 Point Defect Parameters [129]

Bulk Parameters	Interstitial	Vacancy
D.0	600.0	13.0
D.E (eV)	2.44	2.92
CSTAR.0 (cm^{-3})	5.0×10^{22}	2.31×10^{21}
CSTAR.E (eV)	2.36	1.08
KR.0	8.16×10^{-4}	8.16×10^{-4}
KR.E (eV)	3.19	3.19

Table B-15 Point Defect Parameters [129]

Charge State Information	Interstitial	Vacancy
NEU.0	1.0	1.0
NEU.E (eV)	0.0	0.0
NEG.0	5.68	5.68

Table B-15 Point Defect Parameters [129]

Charge State Information	Interstitial	Vacancy
NEG.E (eV)	0.50	0.145
DNEG.0	0.0	32.47
DNEG.E (eV)	0.0	0.62
POS.0	5.68	5.68
POS.E (eV)	0.26	0.45

B.7 Defect Interface Recombination Parameters

Table B-16 Defect Interface Recombination Parameters [129]

Silicon/oxide	Interstitial	Vacancy
KSURF.0	4.65×10^{5}	4.18×10^{2}
KSURF.E (eV)	2.35	2.47
KRAT.0	4.47×10^{6}	2.63×10^{12}
KRAT.E (eV)	0.94	2.88
KPOW.0	0.84	1.79×10^{-07}
KPOW.E (eV)	0.05	-1.91
Silicon/nitride		
KSURF.0	1.0×10^{-05}	1.0×10^{-09}
Silicon/oxynitride		
KSURF.0	1.0×10^{-02}	1.0×10^{-05}
Silicon/gas		
KSURF.0	1.0×10^{-04}	7.0×10^{-08}
KSURF.E (eV)	0.0	4.08
KPOW.0	1.0	1.0

All parameters for other combinations are 0.0.

B.8 Defect Growth Injection Interface Parameters

Table B-17 Defect Growth Injection Interface Parameters [129]

Silicon/oxide	Interstitial	Vacancy
THETA.0 [129]	4.58×10^{8}	0.0
THETA.E (eV) [129]	2.36	0.0
GPOW.0	0.0	1.0
GPOW.E (eV)	0.0	0.0
VMOLE	5.0×10^{22}	5.0×10^{22}

All parameters for other combinations are 0.0.

B.9 Material Parameters

B.9.1 Diffusion Related

Table B-18 Diffusion Related Material Parameters

Parameter	Silicon	Poly	Oxide	Oxynitride	Nitride	Photo	Alumin
NI.0 [130]	3.9×10^{16}	3.9×10^{16}	1.0	1.0	1.0	1.0	1.0
NI.POW [130]	1.5	1.5	0.0	0.0	0.0	0.0	0.0
NI.E [130]	0.605	0.605	0.0	0.0	0.0	0.0	0.0
EPS	11.9	11.9	3.9	7.5	7.5	1.0	1.0

B.9.2 Stress and Oxidation Related

Table B-19 Viscosity Parameters

Parameter	VISC.0	VISC.E	VISC.X
OXIDE (wet)	1.99×10^{-7}	5,292	0.499
OXIDE (dry)	3.1×10^{-3}	7.405	0.499
NITRIDE	1.8×10^{15}	0	0.499
SILICON	1×10^{30}	0	0.499
POLY	5×10^{16}	0	0.499
OXYNI	5×10^{12}	0	0.499

Table B-20 Material Elastic Constants (Compliancies) in units of 10^{-12} dyn/cm^2 [136],[137]

Material	S11.ELAST	S12.ELAST	S44.ELAST
Oxide	1.515	-0.303	3.636
Nitride	0.257	-0.077	0.668
Silicon	0.768	-0.214	1.26
Polysilicon	0.535	-0.150	1.37
Oxynitride	0.257	-0.077	0.668
Aluminum	1.43	-0.500	3.86
6H_SiC	0.134	-0.06	0.388
4H_SiC	0.134	-0.06	0.388
3C_SiC	1.06	-0.476	1.82
Germanium	0.964	-0.26	1.49
SiGe	0.858	-0.236	1.366

Appendix C TSUPREM-4 and TSUPREM-3 Compatibility Features

The following changes in Athena syntax and functionalities are implemented in order to achieve better compatibility with TSUPREM-4 and TSUPREM-3 simulators.

C.1 General Syntax Capabilities

- Added capability to specify that default values of some parameters correspond to those of TSUPREM-4. The modified keyfile `athenakey.tma` with some modified default values is introduced. To run Athena with default parameters specified in athenakey.tma file the syntax is:

  ```
  go athena simflags="-tma"
  ```

  For example, TSUPREM-4 defaults for `TILT` and `ROTATION` parameters in the `IMPLANT` statement are 0°, while Athena uses 7 and 30° respectively.
- A plus character `'+'` can be used as a line continuation sign instead of standard backslash `'\'`.
- Boolean parameters can be set to false by preceding the parameter name with `'^'` or `'!'` character.
- The `'$'` character can be used to specify the comment line. This should only be used at the beginning of the line, because the `'$'` character can be used for substitution of parameters defined by `SET` or `DEFINE` statements of DeckBuild.
- The maximum length of parameter names has extended from 12 to 16 characters. (some TSUPREM-3 names are longer than 12 characters).
- The first character of a parameter name can be a numeral now.

C.2 Execution Control Capabilities Provided by DeckBuild

The detailed description on these new functionalities will be published in the DeckBuild manual. Here, we highlight only key features related to the compatibility issues.

C.2.1 DEFINE Statement and Substitutions Capability

`DEFINE` statement specifies strings for substitution in the following input statements until the `UNDEFINE` statement is encountered. The following `DEFINE` statements and corresponding substitutions are allowed.

```
DEFINE dconditions temp=1000 time=10 dry

DIFFUSE dconditions


DEFINE t1 5.0

DEFINE t2 10


DIFFUSE temp=900 time = ( t1 + t2 )
```

or

```
DIFFUSE temp=900 time = $t1 + $t2
```

or

```
DIFFUSE temp=900 time = @t1 + @t2
```

In case when you have to redefine a string, you should use the %DEFINE statement.

C.2.2 IF/ELSEIF/ELSE/IF.END Capability.

This allows you to perform segments of input deck depending on conditions set in the IF COND=(condition) or IFELSE COND=(condition). For example, the following sequence extracts the gate oxide thickness. If it is greater than required 100 A, then the extra oxide thickness is etched. Otherwise, the lacking oxide thickness is deposited.

```
extract name="gateox" thickness material="SiO~2" mat.occno=1
x.val=0.5
extract name="gateoxdiff" 1.e-5*($gateox - 100.0)
IF cond=($gateoxdiff > 0.0)
   etch oxide thick=$gateoxdiff
ELSE
   deposit oxide thick=-($gateoxdiff)
IF.END
```

C.2.3 LOOP/L.END/ASSIGN/L.MODIFY Capability

LOOP and L.END statements defines the beginning and end of an input deck segment, which will be processed several times. The number of passes is specified by the STEPS parameter of the LOOP statement.

The ASSIGN statement allows you to assign a numerical or character value to a name. It is similar to the DEFINE statement. The main difference is the capability to vary the assigned value within the LOOP cycle using RATIO and DELTA parameters. Also, only '$' and '@' characters can be used for substitution.

The L.MODIFY statement allows you to alter the processing of the current LOOP cycle. For example, if you use BREAK parameter, the current LOOP cycle gets interrupted and control comes to the first line after the L.END statement.

The following example shows LOOP/L.END/ASSIGN/L.MODIFY capability. It demonstrates a simple way to estimate a diffusion time needed to grow the gate oxide with the thickness of 100 A.

First, the initial value 0.0 is assigned to the name 'tt'. Within the loop, variable value is assigned to the name 't'. Its inital value is 1.0 and increases by a factor of 1.2 in each subsequent loop iteration. The current value of 't' is used as diffusion time.

The next ASSIGN statement calculates the total diffusion time.

The EXTRACT statement finds current oxide thickness 'gateox'. If the required condition is that extracted 'gateox' exceeds 100 A, the LOOP cycle is interrupted and the total diffusion time is printed.

```
ASSIGN name=tt n.val=0.0
LOOP steps=20
    ASSIGN name=t n.val=1.0 ratio=1.2
    diffuse time=$t temperature=950 dry
    ASSIGN name=tt n.val=$tt+$t
    extract name="gateox" thickness material="SiO~2" mat.occno=1
x.val=0.1
IF cond = ( $gateox > 100.0 )
   L.MODIFY  break
IF.END
L.END
echo $tt
```

C.3 MESH Statement

The new MESH statement provides an alternative to standard mesh generation using the LINE statements. It also specifies some parameters used in automatic grid generation when layout information is provided by the Mask Data File generated by Taurus Layout and loaded in the MASK statement.

DX.MAX specifies the maximum grid spacing in the horizontal direction. It is used when the grid in the x-direction is specified using the Mask Data File.

DX.MIN specifies the minimum grid spacing in the horizontal direction. It is used when the grid in the x-direction is specified using the Mask Data File.

DX.RATIO specifies the maximum interval ratio between adjacent grid points in the horizontal direction. You can also specify this parameter in the INITIALIZE statement. The default is 1.5.

DY.ACTIV specifies the grid spacing in y-direction at the bottom of the active region.

DY.BOT specifies the grid spacing in y-direction at the bottom of the structure.

DY.RATIO specifies maximum interval ratio between adjacent grid points in vertical direction. This parameter could be also specified in the INITIALIZE statement. The default is 1.5.

DY.SURF specifies the grid spacing in the surface region (i.e., between y=0 and y=LY.SURF).

FAST is equivalent to the AUTO (default) parameter in the INITIALIZE statement.

Note: The MESH statement with LY.SURF and other related parameters cannot be used together with LINE Y statements.

GRID.FAC specifies a global spacing multiplier, which will be applied to all spacing parameters when you generate a grid with the INITIALIZE statement. This parameter is equivalent to the SPACE.MULT parameter of the INITIALIZE statement.

LY.SURF specifies the depth of of the surface region in the default vertical grid (y-direction).

LY.ACTIV specifies the bottom of the active region in the default vertical grid (y-direction).

LY.BOT specifies the depth of the bottom of the structure in the default vertical grid.

C.4 Using MASK statement with the parameter IN.FILE and XLINES for Automatic grid generation in the horizontal direction

This capability can be used only if Athena runs within DeckBuild. If DeckBuild encounters the MASK statement with the parameters IN.FILE=<maskfile.tl1> and XLINES, it recognizes that the file should be in Taurus Layout Mask Data format.

The following information from the maskfile.tl1 are used to build the grid:

- The scale factor (units per micron).
- The minimum and maximum coordinates of mask specification, which define the left and right of simulation space.
- The number of mask in the file.
- The first line of each mask description includes the mask name (e.g., Poly) and number of opaque segments. Each subsequent line gives the minimum and maximum coordinates of each of these segments.

DeckBuild will generate the LINE X statements, which are used by Athena according the following rules.

1. The lines will be generated at each mask edge.
2. The grid spacing at these lines will be equal to DX.MIN specified in the MESH statement.
3. If none of the mask edges coincides with left or right boundary of the simulation space, the LINE statement corresponding to such boundaries will be without spacing.
4. Additional one or two LINE statements will be generated between the lines corresponding to mask edges. The SPAC parameters at these additional lines will be minimum of DX.MAX and DX.MIN*(DX.RATIO)**n, where DX.MIN, DX.MAX and DX.RATIO are parameters specified in the MESH statements. This will guarantee that grid spacings in the horizontal grid will be increased far from mask edges.

These rules are illustrated by the following example of a structure with two POLY gates. If the Mask Data File has the following fragment

```
1.000000E+03

     0    3000

1

POLY           2

     800     1200

     1800    2200
```

and the MESH statement

MESH dx.min=0.01 dx.max=0.1 dx.ratio=2 follows, then DeckBuild will generate the next sequence of the LINE X statements.

```
line x loc=0.000000 tag=left
line x loc=0.800000 spac=0.010000
line x loc=0.950000 spac=0.080000
line x loc=1.050000 spac=0.080000
line x loc=1.200000 spac=0.010000
line x loc=1.350000 spac=0.080000
line x loc=1.650000 spac=0.080000
line x loc=1.800000 spac=0.010000
line x loc=1.950000 spac=0.080000
line x loc=2.050000 spac=0.080000
line x loc=2.200000 spac=0.010000
line x loc=3.000000 tag=right
```

Note: MASK IN.FILE=<maskfile.tl1> cannot be used together with LINE X statements.

Note: .sec files generated by SILVACO's MaskViews tools provide superior capabilities in generating grid and mask processing.

C.5 Using mask information with the EXPOSE MASK=<maskname> statement.

This capability can be used only if Athena runs within DeckBuild. If DeckBuild encounters the statement EXPOSE with the parameter MASK=<maskname>, it provides Athena with a sequence of ETCH statements, which will remove photoresist below all transparent regions of the specified mask. The EXPOSE MASK=<maskname> statement should be preceded by a DEPOSIT PHOTO statement and followed by a DEVELOP (without parameters) statement. For example, if you load the same Mask Data File as used above in the MASK statement, the following input deck fragment appears in the input deck.

```
DEPO PHOTO THICK=1
EXPOSE MASK=POLY
DEVELOP
```

DeckBuild will then generate the next sequence of the ETCH statements.

```
etch photo start x=-0.100000 y=-1000
etch cont x=-0.100000 y=1000
etch cont x=0.800000 y=1000
etch done x=0.800000 y=-1000
etch photo start x=1.200000 y=-1000
etch cont x=1.200000 y=1000
etch cont x=1.800000 y=1000
etch done x=1.800000 y=-1000


etch photo start x=2.200000 y=-1000
etch cont x=2.200000 y=1000
etch cont x=3.100000 y=1000
etch done x=3.100000 y=-1000
```

If you use negative photoresist, the photoresist will be removed underneath all opaque regions.

```
DEPO PHOTO NEGATIVE THICK=1
EXPOSE MASK=POLY
DEVELOP
```

Note: Because this is a special DeckBuild capability the DEPOSIT statement in above sequence has some lexical limitations. The following parameter names/abbreviations must be used:

```
DEPOSIT/DEPO
NEGATIVE/NEG
POSITIVE/POS
THICKNES/THICK
PHOTORES/PHOTO
```

In this case, DeckBuild will generate an alternative sequence of the ETCH statements.

```
etch photo start x=0.800000 y=-1000
etch cont x=0.800000 y=1000
etch cont x=1.200000 y=1000
etch done x=1.200000 y=-1000


etch photo start x=1.800000 y=-1000
etch cont x=1.800000 y=1000
etch cont x=2.200000 y=1000
etch done x=2.200000 y=-1000
```

Note: .sec files generated by Silvaco's MaskViews tool provide superior capabilities in the simulation grid generation and mask processing control.

C.6 Aliases and substitutions for some statements

AMBIENT is alias for OXIDE.

DIFFUSION is alias for DIFFUSE.

Note: PRINT.1D issues a warning message if there is no SELECT statement prior to it.

ELIMINATE is not used in Athena and therefore is ignored. A warning is then issued. The RELAX statement should be used instead. The RELAX capability is similar but more flexible since it can be used in anywhere in the input deck.

ELECTRICAL and **MOBILITY** are not used in Athena and therefore are ignored. A warning is issued. The EXTRACT capabilities of DeckBuild should be used instead.

EXTRACT is not used within Athena and therefore is ignored. DeckBuild has superior extract capabilities. The EQUATION, REACTION, and INTERMEDIATE statements are not part of Athena. Therefore, they are ignored.

LOADFILE is alias for INITIALIZE.

PLOT (in TSUPREM-3 decks) is ignored and TonyPlot should be used instead.

PLOT.1D, **PLOT.2D**, **PLOT.3D**, **CONTOUR**, **LABEL**, and **COLOR** statements are depreciated in Athena. Warnings are issued. TonyPlot should be used instead.

PRINT (in TSUPREM-3 decks) is alias for PRINT.1D.

SAVEFILE is alias for STRUCTURE.

C.7 Changes in the INITIALIZE statement

Boolean parameters <100>, <110>, and <111> that specify crystalline orientation of the silicon substrate are aliases for ORIENTATION=100, ORIENTATION=110, and ORIENTATION=111.

DX.RATIO (new parameter) specifies maximum ration between adjacent mesh lines in x-direction. It is equivalent to the same parameter in the MESH statement.

DY.RATIO (new parameter) specifies maximum ration between adjacent mesh lines in y-direction. It is equivalent to the same parameter in the MESH statement.

IN.FILE is alias for INFILE.

RATIO is an alias for INTERVAL.R.

There are several additional ways to specify initial substrate doping. You can specify the impurity name by using IMPURITY=<impname> parameter, where <impname> could be TS4.BORON=<n>, TS4.PHOSPHORUS=<n>, TS4.ARSENIC=<n>, and TS4.ANTIMONY=<n>. You can specify the corresponding doping either by I.CONC=<conc> or I.RESIST=<resistivity>. You can also specify initial concentration or corresponding resistivity for an individual impurity by using TS4.BORON=<n>, TS4.PHOSPHORUS=<n>, TS4.ANTIMONY=<n>, and TS4.ARSENIC=<n>, while boolean parameters TS4.CONCENTR and TS4.RESISTIV will indicate that the value <n> will be impurity concentration or resistivity, correspondingly.

A one-dimensional grid structure can now be specified without using LINE statements (or their equivalents MESH, MASK or loading of a .sec file from MaskViews). The syntax of the INITIALIZE statement of SSuprem3 and its derivatives can be used:

- **DX** specifies the nominal grid spacing in the initial grid.
- **MIN.DX** specifies the minimal grid spacing.
- **SPACES** specifies the number of the grid spaces in the initial structure.
- **THICKNESS** specifies total thickness of the initial structure.
- **TIF** is ignored. Athena uses Silvaco Structure File (SSF) format.
- **XDX** specifies the distance from the top of the initial structure at which nominal grid spacing is placed.

C.8 Changes in the DEPOSIT statement

There are several additional ways to specify doping in the deposited layer. You can specify the impurity name by the IMPURITY=<impname> parameter, where <impname> could be TS4.BORON=<n>, TS4.PHOSPHORUS=<n>, TS4.ARSENIC=<n>, and TS4.ANTIMONY=<n>. You can specify the corresponding doping either by I.CONC=<conc> or I.RESIST=<resistivity>. You can also specify concentration or corresponding resistivity for an individual impurity in deposited material by using TS4.BORON=<n>, TS4.PHOSPHORUS=<n>, TS4=<n>.ANTIMONY, and TS4.ARSENIC=<n>, while boolean parameters TS4.CONCENTR and TS4.RESISTIV will indicate that the value <n> will be impurity concentration or resistivity, correspondingly.

ARC.SPACE is an alias for MIN.SPACE.

To provide compatibility with SSuprem3, the following aliases have been introduced:

- DX for DY
- XDX for YDY
- MIN.DX for MIN.DY

C.9 Changes in the DIFFUSE statement

There are several additional ways to specify impurity concentration in the ambient gas. You can specify the impurity name using the IMPURITY=<impname> parameter, where <impname> could be TS4.BORON=<n>, TS4.PHOSPHORUS=<n>, TS4.ARSENIC=<n>, and TS4.ANTIMONY=<n>. You can specify the corresponding concentration by I.CONC=<conc>. Alternatively, you can specify concentration of an individual impurity by using TS4.BORON=<conc>, TS4.PHOSPHOR=<conc>, and so on.

The new parameter **SS.IMPURITY** (where the generic name IMPURITY could be substituted by any standard impurity name) specifies that concentration of the named impurity in the ambient gas is set to its solid solubility in silicon at the current temperature.

C.10 Changes in the ETCH statement

TRAPEZOI is an alias for the DRY.

If THICKNESS is not specified, it assumed to be infinite.

C.11 Changes in the STRUCUTURE (SAVEFILE) statement

OUT.FILE is an alias for OUTFILE.

TIF, **DEVICE**, and **MEDICI** are ignored because Athena and other Silvaco TCAD tools use the universal SSF data format.

C.12 Changes in the IMPLANT statement.

D.PLUS is an alias for PLUS.ONE and UNIT.DAMAGE.

D.SCALE is an alias for DAM.FACTOR.

IMPL.TAB specifies the table from the T-SUPREM4 default implant tables file to be used.

C.13 Changes in the ELECTRODE statement

BOTTOM is an alias for SUBSTRATE.

C.14 Changes in the METHOD statement

OX.ADAPT, **IMP.ADAPT**, and **DIF.ADAPT** are aliases for ADAPT.

PD.FERMI is an alias for FERMI.

PD.TRANS is an alias for TWO.DIM.

PD.FULL is an alias for FULL.CPL.

C.15 Changes in the MATERIAL statement

E.FIELD specifies that the electric field terms are to be accounted for in the diffusion calculations. This parameter is always set to TRUE in semiconductors.

POLYCRYS is an alias for the POLY.DIFF parameter in the METHOD statement.

Bibliography

1. J.D. Plummer, M.D. Deal, and P.B. Griffin, *Silicon VLSI Technology. Fundamentals, Practice and Modeling*, Prentice Hall, 2000.
2. G.F. Carey, W.B. Richardson, C.S. Reed, and B. J. Mulvaney. *Circuit, Device and Process Simulation. Mathematical and Numerical Aspects*, John Wiley and Sons, 1996.
3. C. C. Lin and M. E. Law, "2-D Mesh Adaption and Flux Discretizations for Dopant Diffusion Modeling", *IEEE Trans. CAD*, v. 15, p. 194, 1995.
4. C. C. Lin and M. E. Law, "Mesh Adaptation and Flux Discretizations for Dopant Diffusion Modeling", *NUPAD-V*, p. 151, 1994.
5. D. Mathiot, and J.C. Pfister. "Dopant Diffusion in Silicon: A Consistent View Involving Nonequilibrium Defects", *J. of Appl. Phys*, v. 55, p. 3518, 1984.
6. P.M. Fahey, *Point Defects and Dopant Diffusion in Silicon*, Ph.D Thesis, Integrated Circuits Laboratory, Department of Electrical Engineering, Stanford University, June 1985.
7. M.E. Law, *Two Dimensional Numerical Simulation of Dopant Diffusion in Silicon*, Ph.D. Thesis, Department of Electrical Engineering, Stanford University, 1988.
8. J. A. Van Vechten and C. D. Thurmond, "Entropy of Ionization and Temperature Variation of Ionization Levels of Defects in Semiconductors," *Phys. Rev. B*, v. 14, p. 3539, 1976.
9. R.B. Fair, (ed. F.F.Y. Wang). "Concentration Profiles of Diffused Dopants in Silicon", *Impurity Doping Process in Silicon*. North Holland, New York, 1981.
10. F.Lau *et.al.*, "A Model for Phosphorus Segregation at the Silicon-Silicon Dioxide Interface", *Appl. Phys. A*, v. 49, p 671, 1989.
11. Y.S. Oh, D. Ward, "A Calibrated Model for Trapping of Implanted Dopants at Material Interface During Annealing, *IEDM Tech. Digest*, p. 509, 1998
12. P.B. Griffin, and J.D. Plummer. "Process Physics Determining 2-D Impurity Profiles in VLSI Devices", *IEDM Tech. Digest*, p. 522, 1986
13. S.M. Hu, "On Interstitial and Vacancy Concentration in Presence of Injection", *J. Appl. Phys*., v. 57, p. 1069, 1985.
14. S. Crowder, *Processing Physics in SOI Material*, Ph.D Thesis, Department of Electrical Engineering, Stanford University, 1995.
15. Chakarov, I. and M. Temkin, "Modeling of Ion Implantation in SiC Crystals", *Nucl. Instr. And Meth. In Phys. Res.* v. B-242 (2005):690.
16. http://www.qualitymaterial.net/products.html. Date Accessed: July 3, 2013.
17. http://www.semiconductorwafers.net/4H-Semi-insulating-SiC.html. Date Accessed: July 3, 2013.
18. B.J.Mulvaney, W.B.Richardson, and T.L.Crandle, "PEPPER - A Process Simulator for VLSI", *IEEE Trans. on Computer-Aided Design*, v. 8, p. 336, 1989.
19. L.Mei, M.River, Y.Kwart, and R.W.Dutton, "Grain Growth Mechanism in Polysilicon", *Proc. 4th Intern. Symp. on Silicon Materials and Technology*, v. 81, p. 1007, 1981.

20. L. Mei and R.W. Dutton, “A Process Simulation Model For Multilayer Structures Involving Polycrystalline Silicon”, *IEEE Trans. Electron Devices*, v. ED-29, p. 1726, 1982.

21. F. Boucard, Dopant diffusion modelling in silicon for shallow junctions processing.Ph. d. thesis, Louis Pasteur University, September 2003.

22. C. Ortiz and D. Mathiot, "A new kinetic model for the nucleation and growth of selfinterstitialclusters in silicon," Mat. Res. Soc. Symp. Proc. "Si Front-end processingphysics and technology of dopant defect interactions lll", edited by E.C.Jones, K.S. Jones, M.D. Giles, P.Stolk, J.Matsuo, vol. 669, no. -, pp. J5-6, 2001.

23. N. Cowern, G. Mannino, P. A. Stolk, F. Roozeboom, H. G. A. Huizing, J. G. M. van Berkum, F. Cristiano, A. Claverie, and M. Jaraíz, "Energetics of self-interstitial clusters in si," *Phys. Rev. Lett.*, vol. 82, no. 22, p. 4460, 1999.

24. T. J. Lenosky, B. Sadigh, S. K. Theiss, M.-J. Caturla, and T. D. de la Rubia, "Ab initio energetics of boron-interstitial clusters in crystalline si," *Appl. Phys. Lett.*, vol. 77, no. 12, p. 1834, 2000.

25. P. H. Keys, *Phosphorus-defect interactions during thermal annealing of ion implanted silicon*. Ph.d. thesis, University of Florida, 2001.

26. A. H. Gencer, *Modelling and simulation of transient enhanced diffusion based on interaction of point and extended defects*. Ph. d. thesis, Boston University, 1999.

27. L. Pelaz, M. Jaraiz, G. Gilmer, H. Gossmann, C. Rafferty, D. Eaglesham, and J. M. Poate, "B diffusion and clustering in ion implanted si: The role of b cluster precursors," *Appl. Phys. Lett.*, vol. 70, no. 17, p. 2285, 1997.

28. W. Orr Arienzo, R. Glang, R. F. Lever, R. K. Lewis, and F. F. Morehead, "Boron diffusion in silicon at high concentrations," *J. Appl. Phys*, vol. 63, no. 1, p. 116, 1988.

29. M. Yoshida, E. Arai, H. Nakamura, and Y. Terunuma, "Excess vacancy generation mechanism at phosphorus diffusion into silicon," *J.Appl.Phys.*, vol. 45, no. 4, p. 1498, 1974.

30. S. Solmi, F. Barualdi, and R. Canteri, "Diffusion of boron in silicon during postimplantation annealing," *J. Appl. Phys.*, vol. 69, no. 4, p. 2135, 1991.

31. B. Colombeau, *Interactions entre les défauts étendus et anomalies de diffusion des dopants dans le silicium : modéle physique et simulation prédictives*. Thèse de doctorat, Universite Paul-Sabatier (Toulouse), Septembre 2001.

32. P. Fastenko, *Modeling and simulation of arsenic activation and diffusion in silicon*. Ph. d. thesis, University of Washington, 2002.

33. B.E. Deal, and A.S. Grove. “General Relationship for the Thermal Oxidation of Silicon”, *J. Appl. Phys*. v. 36, p. 3770, 1965.

34. D. Chin, *Two Dimensional Oxidation. Modeling and Applications.*, Ph.D Thesis, Department of Electrical Engineering, Stanford University, 1983.

35. H.Z. Massoud, *Thermal Oxidation of Silicon in Dry Oxygen-Growth Kinetics and Charge Characterization in the Thin Regime*, Technical Report, Stanford Electronic Laboratories, Stanford University, 1983.

36. H. Eyring, “Viscosity, Plasticity, and Diffusion as Examples of Absolute Reaction Rate”, *J. Chem. Phys*. v. 4, p. 283, 1936.

37. B.E. Deal, “Thermal Oxidation Kinetics of Silicon in Pyrogenic H2O and 5% HCl/H2O Mixtures”, *J. Electrochem. Soc*. v. 125, p. 576, 1978.

38. D.W. Hess and B.E.Deal. “Kinetics of the Thermal Oxidation of Silicon in O2/HCl Mixtures”, *J. Electrochem. Soc*. v. 124, p. 735,1977.

39. R.R. Razouk, L.N.Lie, and B.E.Deal. “Kinetics of High Pressure Oxidation of Silicon in Pyrogenic Steam”, *J. Electrochem. Soc*. v. 128, p. 2214, 1981.

40. 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*. v. 126, p. 1516, 1979.

41. B. E. Deal and M. Sklar, “Thermal Oxidation of Heavily Doped Silicon,” *J. ELectrochem. Soc.*, v. 112, p. 430, 1965.

42. L. N. Lie, R. R. Razouk, and B. E. Deal, “High Pressure Oxidation of Silicon and Dry Oxygen,” *J. Electrochem. Soc.*, v. 129, p. 2828, 1982.

43. N. Guillemot, “A New Analytical Model of the Bird’s Beak”, *IEEE Trans. on Electron Devices*. v. ED-34, p. 1033, 1987.

44. C.L. L.S.Hung, J.Gyulai, J.W.Mayer, S.S.Lau, and M-A.Nicolet, "Kinetics of TiSi2 Formation by Thing Ti Films on Si", *J. Appl. Phys*., v. 54, p. 5076, 1983

45. C.A.Pico, and M.G.Lagally, "Kinetics of Titanium silicide Formation on Single-Crystal Si: Experiment and Modeling", *J. Appl. Phys.*, v. 64, p. 4957, 1988.

46. S.Sen-Hou Ko, Sh. Murarka, A.R. Sitaram, "Ellipsometric measurements of the CoSi2 formation from very thin cobalt films on silicon", *J. Appl. Phys*., v.71, p. 5892, 1992.

47. M.A. Nicolet and S.S. Lau, in *VLSI Handbook*, Ed. Norman G. Einspruch, Academic Press, p.430, 1985.

48. J. Lindhard, M. Scharff, and H.E. Schiott. “Range Concepts and Heavy Ion Ranges”, *Kgl. Dan. Vid. Selsk. Mat.-fys. Medd.*, v. 33, 1963.

49. R. Smith (Ed.), *Atomic and Ion Collisions in Solids and at Surfaces*, Cambridge University Press, 1997.

50. D.G. Ashworth, R. Oven and B. Mundin, “Representation of Ion Implantation Profiles be Pearson Frequency Distribution Curves”, *J. Phys. D*, v. 23, p. 870, 1990.

51. A. F. Tasch, “An Improved Approach to Accurately Model Shallow B and BF2 Implants in Silicon”, *J. Electrochem. Soc*., v. 136, p. 810, 1989.

52. K.B. Parab *et.al.*, “Analysis of Ultra-Shallow Doping Profiles Obtained by Low Energy Ion Implantation”, *J. Vac. Sci. Technol*., v. B14, p. 260, 1996.

53. G.A.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. Electron. Dev*. v. ED-32, p. 1899, 1985.

54. A.F. Burenkov, F.F.Komarov, and M.M.Temkin. “Analytical Calculation of Ion Implantation through Mask Windows” (in Russian), *Microelektronika*, v. 16, p. 15, 1987.

55. A.F. Burenkov, A.G. Kurganov, and G.G. Konoplyanik, “Two-Dimensional Local Ion Implantation Distribution” *(in Russian), Povekhnost (Surface Sciences)*, v.8, p.52, 1989.

56. J. Lorenz, W. Kruger, and A. Barthel “Simulation of the Lateral Spread of Implanted Ions: Theory”, *NASECODE-VI*, Ed. J.J.H.Miller, Boole Press, p.513, 1989.

57. D.G. Ashworth, M.D.J. Bowyer, and R. Oven, “Representation of Ion Implantation Distributions in Two and Three Dimensions”, *J. Phys. D*, v. 24, p. 1120, 1991.
58. G. Hobler, E. Langer, and S. Selberherr, “Two-Dimensional Modeling of Ion Implantation with Spatial Moments”, *Solid-State Electronics*, v. 30, p. 445, 1987.
59. M. Temkin and I. Chakarov, “Computationally Effective Model for 2D Ion Implantation Simulation”, *Semiconductor Process and Device Performance Modeling*, Eds, S.T. Dunham, J.S. Nelson, MRS, p. 27, 1998.
60. R.Oven, D.G.Ashworth, and M.D.J. Bowyer, “Formulas for the Distribution of Ions Under an Ideal Mask”, *J. Phys. D*,. v. 25, p.1235, 1992.
61. J.E. Gibbons, W. S. Johnson, and S. W. Mylroie. *Projected Range Statistics (2nd edition)*, Stroudsburg, Pennsylvania: Dowden, Hutchinson, & Ross, Inc., 1975.
62. A.F. Burenkov, F.F.Komarov, M.A.Kumakhov, and M.M.Temkin. *Tables of Ion Implantation Spatial Distributions*. Gordon & Breach Science Publishers, 1986.
63. O. B. Firsov, *Soviet Physics JETP*, v. 33, p. 696, 1957
64. J. F. Ziegler, J. P. Biersack, U. Littmark, *The stopping and range of ions in solids*, v. 1, Pergamon Press, 1985.
65. *UT-Marlowe Version 5.0 User manual*, University of Texas, Austin, USA, 1999.
66. O. B. Firsov, *Soviet Physics JETP*, v. 36, p. 1517, 1959.
67. W. Brandt and M. Kitagawa, “Effective Stopping-Power Charges of Swift Ions in Condensed Matter.”, *Phys. Rev. B*, v. 25, p. 5631, 1982.
68. F. L. Vook, *Defects in Semiconductors*, p. 60, 1972
69. I. R. Chakarov and R. P. Webb, ”CRYSTAL -- Binary Collision Simulation of Atomic Collision and Damage Buildup in Crystalline Silicon”, *Radiation Effects*, v. 130-131, p. 447, 1994.
70. H. Kahn and T. E Harris, “Estimation of Particle Transmission by Random Sampling”, *National Bureau of Standards Applied Mathematics Series*, v. 12, p. 27, 1951.
71. B. A. Bayes, “Statistical Techniques for Simulation Models”, *The Australian Computer Journal* v. 2, p. 190, 1975.
72. J. M. Hemmersley and D. C. Handscomb, *Monte Carlo Methods*, Methuen and Co., Ltd., London, 1964.
73. M. Villién-Altamirano and J. Villién-Altamirano,. “RESTART: A Method for Accelerating Rare Event Simulations”, *Proc. 13th Int. Teletraffic Congress, ITC 13 (Queueing, Performance and Control in ATM)*, P. C. Cohen J. W., Ed., North-Holland, Copenhagen, Denmark, p. 71, 1991;
74. A. Phillips and P. J. Price, “Monte Carlo Calculations on Hot Electron Energy Tails”, *Appl. Phys. Lett.*, v. 30, 1977
75. S.H.Yang, D. Lim, S. Morris, and A. F. Tasch, “A More Efficient Approach for Monte Carlo Simulation of Deeply-Channeled Implanted Profiles in Single-Crystal Silicon”, *Proc. NUPAD*, p. 97, 1994.
76. K. M. Beardmore and N. Gronbech-Jensen, “Efficient Molecular Dynamics Scheme for the Calculation of Dopant Profiles Due to Ion Implantation”, *Phys. Rev. E*, 57, 1998, p. 7278.

77. J. M Hernández-Mangas, J. Arias, M. Jaraiz, L. Bailón, and J. Barbolla, “Algorithm for Statistical Noise Reduction on Three-Dimensional Ion Implant Simulations”. *Nucl. Instr. Meth. in Physics Research B*, 174, 2001, p, 433-438.

78. W. Bohmayr, A. Burenkov., J. Lorenz, H. Ryssel, and S. Selberherr, “Trajectory Split Method for Monte Carlo Simulation of Ion Implantation”, *IEEE Transactions on Semiconductor Manufacturing*, v. 8, p. 402, 1995.

79. P Glasserman, P. Heidelberger, P. Shahabuddin,and T.Zajic, *A look at Multilevel Splitting*, Technical report RC-20692, IBM Research Division, T. J. Watson Research Center, Yorktown Heights, New York, 1997.

80. S.E. Hansen and M. Deal, “SUPREM-IV.GS: Two-Dimensional Process Simulation for Silicon and Gallium Arsenide”, Integrated Circuits Laboratory, Stanford University, 1993.

81. O. Madelung (Ed.). *Semiconductors - Basic Data.* Springer Verlag, 1996.

82. R. Anholt *et.al.*, ”Ion Implantation Into Gallium Arsenide”, *J. Appl. Phys*. v. 64, p.3429, 1988.

83. K. Rajendran, "Simulation of Boron Diffusion in Strained SiGe Epitaxial Layers", *Proc. SYSPAD*, p. 206, 2000.

84. R.F. Lever, "Boron Diffusion across Silicon-SiGe Boundaries", *J. of Appl. Phys.*, v. 83, p. 1988, 1998.

85. J.L. Ngau, P.B. Griffin, and J.D. Plummer, “Modelling the Suppression of Boron Transient Enhanced Diffusion in Silicon by Substitutional Carbon Incorporation”, *Japanese. of Appl. Phys.*, v. 90, p. 1768, 2001

86. R.Jewett, *A String Model Etching Algorithm*, M. S. Thesis, University of California, Berkeley, 1979.

87. W.G.Oldham *et.al.*, “A General Simulator For VLSI Lithography And Etching Processes: Part II - Application To Deposition And Etching”, *IEEE Trans. on Electron Devices*, v. ED-27, p. 1455,1980.

88. A.R.Neureuther, C.H.Ting, and C.Y.Lin, “Application of Line-Edge Profile Simulation to Thin-film Deposition Process”, *IEEE Trans. on Electron Devices*, v. ED-27, p. 1449, 1980.

89. A.R.Neureuther, “Basic Models And Algorithms For Wafer Topography Simulation”, P*roblems and New Solutions for Device and Process Modeling*, Ed. J.J.H. Miller, Boole Press, Dublin, p. 99, 1985.

90. A.R.Neureuther, “Algorithms For Wafer Topography Simulation”, *NASECODE IV*, Dublin, Ireland, p.58, 1985.

91. *SAMPLE User Guide*, Department of Electrical Engineering and Computer Sciences, UC Berkeley, 1991.

92. M. Sikkens, *et.al.*, *Opt. Eng.*, v. 25, p. 142, 1986.

93. R.N. Tait, T. Smy and M.J. Brett “A Ballistic Deposition Model for Films Evaporated Over Topography”, *Thin Solid Films*, v. 187 p. 375, 1990.

94. R.N. Tait, S.K. Dew, T. Smy and M.J. Brett “Ballistic Simulation of Optical Coatings Deposited Over Topography”, *SPIE Proc.* v. 1324, p. 112, 1990.

95. R.N. Tait, T. Smy and M.J. Brett “Simulation and Measurement of Density Variation in Mo Films Sputter Deposited Over Oxide Steps”, *J. Vac. Sci. Technol.* v. A8, p. 1593, 1990.

96. S.F.Meier, *Etching Simulation Of Nonplanar Layers*, M.S. Thesis, UC Berkeley, 1987.

97. J.L.Reynolds, A.R.Neureuther, and W.G.Oldham, “Simulation of Dry Etched Line Etched Profiles”, *J. Vac. Sci. Technol*., v. 16, p. 1772, 1979.

98. A.R.Neureuther, C.Y.Liu, and C.H.Ting, “Modeling Ion Milling”, *J. Vac. Sci. Technol.*, p. 1167, 1979.

99. S. Takagi, K. Iyanagi, S. Onoue, T. Shinmura, and M. Fujino, "Topography Simulation of Reactive Ion Etching Combined with Plasma Simulation, Sheath Model, and Surface Reaction Model", J*apanese J. Appl. Phys.*, v. 41, no. 6A, p. 3947, 2002.

100.P.Sutardja, Y.Shacham-Diamand, and W.G.Oldham, “Two Dimensional Simulation of Glass Reflow And Silicon Oxidation”, *VLSI Technology Technical Digest*, p. 39, 1986.

101.P.A.Burke, “Semi-Empirical Modeling of SiO2 Chemical-Mechanical Polishing Planarization”, *Proc. VMIC Conf.*, p. 379, 1991.

102.J.Warnock, “A Two-Dimensional Process Model for Chemimechanical Polish Planarization”, *J. Electrochem. Soc.* v. 138, p. 2398, 1991.

103.B.M. Watraslewicz, “Image Formation in Microscopy at High Numerical Aperture”,. *Optical Acta*, v. 12, p. 167, 1965.

104.B. Richards, E.Wolf, “Electromagnetic Diffraction in Optical Systems II. Structure of the Image Field In An Aplanatic System”, *Proc. Phys Soc*., v. A 253, p. 358, 1959.

105.H.H. Hopkins, “On the Diffraction Theory of Optical Images”, *Proc. Roy. Soc.*, v. A 217, p. 408, 1953.

106.H.H. Hopkins, “Applications of Coherence Theory In Microscopy And Interferometry”, *J. Opt. Soc.* Am., v. 47, p. 508, 1957.

107.J.Tsujiuchi, “Image Forming Performance of Projection Systems”, *Japanese. J. of Appl. Phys*., v. 4, Suppl. I., Proc. Conf. On Photographic And Spectroscopic Optics, p. 251, 1965.

108.W.H.A. Fincham, M.H.Freeman, *Optics*, Butterworths, London, Ch. 15, 1980.

109.J. Van Roey, J.van der Donk, P.E. Lagasse. “Beam-Propagation Method: Analysis and Assessment”. *J. Opt. Soc. Am.*, v. 71, p. 803, 1981.

110.J. Z. Y. Guo, F. Cerrina. “Modeling X-ray Proximity Lithography”. *IBM J. Res. Develop*., v. 37, p. 331, 1993.

111.A. Erdmann, C. L. Henderson, C. G. Wilson, W. Henke. “Influence of Optical Nonlinearities of the Photoresist on the Photolithographic Process: Basics”. *SPIE Proc.*, v. 3051, p. 529, 1997.

112.A. Erdmann, C. L. Henderson, C. G. Wilson, R. R. Dammel. “Some aspects of thick film resist performance and modeling”. *SPIE Proc.*, v. 3333, p. 1201, 1998.

113.D.A. Bernard, “Simulation Of Post Exposure Bake Effects On Photolithographic Performance of a Resist Film”, *Phillips Journal of Research*, v. 42, p. 566, 1987.

114.F.H. Dill, W.P. Hornberger, P.S. Hauge, J.M.Shaw, “Characterization of Positive Photoresist”, *IEEE Trans. Electron Devices*, v. ED-22, p. 445, 1975.

115.D.J. Kim, *et. al.*, “Development of Positive Photoresist”, *IEEE Trans. Electron Devices*, v. ED-31, p. 1730, 1984.

116.C.A. Mack, “PROLITH: A Comprehensive Optical Lithography Model”, *SPIE Proc.*, v. 538, p.207, 1985.

117.P. Trefonas III, *et al.*, “New Principle For Image Enhancement In Single Layer Positive Photoresist”, *SPIE Proc.*, v. 771, p. 194, 1987.

118.Y. Hirai et.al., “Process Modeling For Photoresist Development And Design Of Drl/sd (double Resist Layer) by a Single Development Process”, *IEEE Trans. on CAD,* v. CAD-6, p. 403, 1987.

119.J.Van Roey, J. vander Donk, P.E. Lagasse, “Beam-propagation method: analysis and assessment”, *J. Opt. Soc. Am.*, Vol. 71, No. 7, p.803, July 1981.

120.J.Z.Y.Guo, F.Cerrina, “Modeling X-ray proximity lithography”, *IBM J. Res. Develop*., Vol. 37, No. 3, p.331, May 1993.

121.J.W.Goodman, *Introduction to Fourier Optics*, McGraw-Hill, New York, 1969.

122.B. Kernighan and D.Ritchie, *The C Programming Language*, Prentice-Hall, 1988.

123.D.B. Kao, J.P.McVittie, W.D.Nix, and K.C.Saraswat. “Two Dimensional Thermal Oxidation of Silicon I. Experiments”, *IEEE Trans. Electron. Dev*., ED-34, p.1008, 1987.

124.R.B. Fair, *Impurity Doping Process in Silicon*, Ed. F.F.Y. Wang, North Holland, Amsterdam, 1981.

125.C.L. Chu, “Characterization of Lateral Diffusion in Silicides”, *IEDM Tech. Digest*, p. 245, 1990.

126.P.H. Langer and J.I. Goldstein. “Boron Autodoping During Silane Epitaxy”, *J. Electrochem. Soc*., v. 124, p. 591, 1977.

127.G.L. Vick and .M. Whittle. “Solid Solubility and Diffusion Coefficients of Boron in Silicon”, *J. Electrochem. Soc*. v. 116, p. 1142, 1969.

128.F.A. Trumbore “Solid Solubilities of Impurity Elements in Germanium and Silicon”, *Bell System Tech. J*. v. 39, p. 205, 1960.

129.H. Park and M. Law, “Point Defect Based Modeling of Low Dose Silicon Implant Damage and Oxidation Effects on Phosphorus and Boron Diffusion in Silicon”, *J. of Appl. Phys*. v. 72, p. 3431, 1992

130.F.J. Morin and J.P. Maita. “Electrical Properties of Silicon Containing Arsenic and Boron”, *Phys. Rev*. v. 96, p. 28, 1954.

131.M. Giles, “Transient Phosphorus Diffusion Below the Amorphization Threshold”, *J. Electrochem. Soc*. v. 138, p. 1160, 1991.

132.P.Gas *et.al.*, "Diffusion of Sb, Ga, Ge, and As in TiSi2," *J.Appl.Phys.*, v. 63, p. 5335, 1988.

133.P.Gas *et.al.*, "Boron, Phosporus, and Arsenic Diffusion in TiSi2," *J.Appl.Phys.*, v. 60, p.1634, 1986.

134.C.M.Osburn *et.al.*, "The Effect of Titanium Silicide Formation on dopant Redistribution," *J.Electrochem. Soc.*, v. 135, p.1490, 1988.

135.Chi On Chui *et.al.*, “Activation and Diffusion Studies of Ion-implanted p and n Dopants in Germanium”, *Appl. Phys. Lett.*, v.83, p. 3275, 2003

136.S. Uppal, “Diffusion of Ion-implanted Boron in germanium”, *J. Appl. Phys.*, v.90, p. 4293, 2001.

137.N.R. Zandenberg, *et.al.*, “Boron and Phosphorus Diffusion in Strained and Relaxed Si an SiGe”, *J. Appl. Phys.*, v.94, p.3883, 2003.

138.Yi-M. Sheu *et.al.*, “Modeling Mechanical Stress Effect on Dopant Diffusion in Scaled MOSFETs”, *IEEE Trans. Electron Devices*, v. 52, p.30, 2005.

139.V. Senez, P. Ferreira, and B. Baccus, “2-D Simulation of Local Oxidation of Silicon: Calibrated Viscoelastic Flow Analysis,” *IEEE Trans. Electron Devices*, v. 43, p. 720, 1996.

140.W.A. Brantley, J. Appl. Phys., Vol. 44, p. 534, 1973.

Index