# Process Modeling Modules

# User's Guide and Reference Manual

Boston University
Process Modeling Group
http://tcad.bu.edu/~srini/PMM

Version: 99.11.08

# Contents

# Chapter 1

# Introduction

The concept of Process Modeling Modules (PMM) came out when we realized that there was a gap to be filled between a process modeling software like SUPREM and a partial differential equation (PDE) solver like DOPDEES or Alamode. Process modelers are not flexible enough to let you choose the equations you want to be solved for, they are hard-coded and you can only change some parameters. On the other hand, when using a PDE solver, you have to specify every single equation and parameter in all input decks, even if some equations and parameters are considered to be well-known.

Thus, one form of software gave you no flexibility, while the other required you to do the same work over and over, giving way to bugs. We attack this dilemma by writing reusable *modules* that can be incorporated into input decks of PDE solvers. This will keep the flexibility of a PDE solver, while providing reasonable defaults for models and parameters.

The resulting software, PMM, can be used both for DOPDEES and Alamode. Alamode users should adopt a few new syntactical rules when specifying their own equations on top of the ones provided by PMM. Other than that we hope that PMM will be easy to learn and use.

## 1.1 Installing PMM

All you need to do for installing PMM is to unpack the compressed archive file and add a line like the following to your `~/.dopdeesrc` or `~/.alamoderc` file:

```
source /home/user/BUSTOP/PMM/scripts/init.tcl
```

Of course, you'll need to change the path of PMM to wherever you have installed it. In the current version of Alamode the `alamoderc` file is sourced only when the program is invoked interactively. Thus you may have to source it explicitly at top of your input deck (or ask Alamode supporters for a patch).

PMM uses a separate package called UNITS, which has commands for handling units and unit systems in Tcl. Please read the documentations for UNITS in the directory `BUSTOP/UNITS/doc`.

## 1.2 PMM commands

- `module <name> -opt1 -opt2`

  The `module` command is used to include a PMM module. A module defines equations for the system. `name` is the name of the module, and `-opt`s are options to the module. Options vary from module to module. If a module includes another module, the arguments for the child module can be specified when calling the parent module. A module defines equations that will be valid only in the *current region*. A module can be included in a given region only once. There is a special command `module clear` which will clear all equations and all defined structure from the memory.

- `param <name> ?value?`

  The `param` command is used to set and read parameter values. If `value` is specified, then the parameter `name` is set to that value; otherwise the value of the parameter `name` is returned.

- `optimize <filename>`

  The `optimize` statement reads a file with parameter names and values, generated by the optimizer program. The file is assumed to have lines of form `name=value`.

## 1.3 Using PMM

An input deck utilizing PMM will consist of the following components:

#### Parameter selection

The user first should specify the simulation temperature (in degrees Kelvin) with a statement like:

```
param TEMPK [unit 1000 C]
```

The temperature doesn't have to be a constant, but can be any expression involving functions. Then, the user must make a paramater module selection:

```
module params-XXX
```

The parameter module will have all values of parameters to be used in the simulation. After loading the parameter module, parameters may be modified with `param` commands; or a set of parameters that are being optimized may be loaded with `optimize` command. If the user wishes to proceed with the parameters as defined in the parameter module, none of the above step is necessary.

### Initialization

This is where the user does grid specification, either using explicit commands or by reading from an external structure file. It is very important to define the initial conditions for dopants in this part of your input file since the equations to be solved will depend on the dopants in your structure. If you define a dopant after you include a module, the module will not act on this dopant.

PMM modules provide reasonable defaults for variables other than dopants. If, for example, the structure is defined without fields for point defects, PMM will initialize them to their equilibrium values. If, however, the user defines initial conditions for point defects, these initial values will be used.

### Module selection

The modules that will be used to solve the system should now be selected. Keep in mind that any defined module will be valid only in the current region, thus appropriate modules should be selected for each region.

### Defining your own operators

As mentioned previously, you may define operators and equations in addition to the ones provided by the modules (but you don't have to). If you want to do so, you should do it at this point. DOPDEES users may simply specify new operators with the `op` command. For compatibility, Alamode users, too, may use DOPDEES' `op` command. The defined operators must have exactly the same syntax and order of arguments as in DOPDEES. The operator will be applied to, as in DOPDEES, the current region, and field names shouldn't have the "InRegion" tags, because all fields are assumed to be in the current region. Read the DOPDEES manual for more info about the operator syntax.

Alamode users who are not familiar with DOPDEES may use Alamode syntax for defining operators. But then, they must attach their new operator to an operator list. The syntax will look like:

```
set myop [operator .... ]
lappend OPLIST(FieldInRegion) $myop
```

Here, `myop` is set to the new operator, `Region` is the name of the region this operator is valid, and `Field` is the name of the field which stands on the left hand side of the equation in which this operator is found. That is, this operator is attached to the list of operators for the field `FieldInRegion`. `OPLIST` is a global variable keeping track of the list of operators. Please preserve the case of the word `In` between the field and the region.

Boundary operators should be attached to another list, BOPLIST, as follows:

```
set mybop [operator .... ]
lappend BOPLIST(Field:Region,Interface) $mybop
```

Here, `Field` and `Region` stand for the FieldInRegion which will be solved for in the equation this operator is attached to. `Interface` is the name of the interface this operator will be active (e.g. `Region1/Region2` or `ExposedRegion1`). Note the colon between `Field` and `Region` and the comma between `Region` and `Interface`.

### Running the PDE solver

DOPDEES users may run the PDE solver with familiar `solver` commands. Alamode users first have to define a *model* before running the simulator. There is no need to use Alamode's `equation`, `systemInRegion` and `model` commands, since all operators have already been collected in lists. Instead PMM provides a `setmodel` command, which has the following syntax:

```
setmodel name
$name -stopTime 10 ...
$name -transientSolve
```

This command will make equations out of operators, systems out of equations and a model out of systems, and set `name` to the defined model. Solution parameters may be specified in the usual fashion using `name` as a key, as in the example above.

# Chapter 2

# Defined modules

Following is a list of defined modules. All modules accept certain options after the command name. Modules usually include other modules, and options are passed down the hierarchy. Thus, you may specify any valid options for the child modules when calling the parent module.

A common option for all modules is `-debug`, which results in extra debugging information, such as the values of used parameters, to be printed on the screen.

A module cannot be included twice in the same region. Trying to do so results in a warning if `-debug` is specified. Some modules may not be used concurrently, and produce an error is such a case.

## 2.1 Diffusion modules

### 2.1.1 The `diffusion` module

The `diffusion` module is the most comprehensive module in PMM. It is used for diffusion of dopants in silicon. It can implement various diffusion models. Options for this module are:

- `-fermi`: Use "fermi" diffusion model, which assumes that point defects are at their equilibrium values. This is analogous to the "fermi" model in SUPREM.

- `-full`: Use "fully coupled" pair diffusion model, which assumes that the point defect diffusion and dopant diffusion are fully coupled. Not only does the non-equilibrium concentration of point defects affect the dopant diffusivites, but also their gradient can move dopants. This model assumes that the concentration of dopant-point defect pairs is much less than the concentration of dopants, but not necessarly much less than point defect concetrations. The model assumes that dopant-point defect pairing reaction is in equilibrium. It is analogous to the "full.cpl" model in SUPREM.

- `-five`: Most comprehensive pair diffusion model. It doesn't assume that dopant-point defect pairing reaction is in equlibrium and keeps track of dopant-defect pairs exclusively. Slowest diffusion model.

The `diffusion` module includes the `carrierconc` module and if `-full` or `-five` has been specified, the `point-defect` module. Thus, options for these modules maybe used as options to `diffusion` module.

### 2.1.2 The `carrierconc` module

The `carrierconc` module is responsible for calculating the net active carrier concentration and diffusivities of dopants. It is usually used in conjunction with one of the diffusion modules above, and is not intended to be called explicitly by the user. Valid options are:

- `-sss`: Use a simple solid solubility cut-off model to calculate the active portion of the dopant profile. It is important NOT to specify `-sss` when using one of the dopant precipitation modules (e.g. `kpm-BIC` and `kpm-AsV`).
- `-ddp`: Use dopant-dopant pairing module, where dopants may become inactive by pairing with each other (e.g. B-As pairing).
- `-hcd`: Include high concentration diffusivity enhancement effects for As and P. Both dopants show a strong increase in diffusivity above a threshold concentration.

### 2.1.3 The `point-defect` module

The `point-defect` module is used for diffusion and recombination of point defects (interstitials and vacancies). Valid options are:

- `-nosurfrec`: Don't use a constant surface recombination velocity model for point defects.

## 2.2 The {311} module

The {311} module is intented for interstitial type defects and includes a models for both {311} and dislocation loops. Valid options are:

- `-sss`: A simple single cluster model for simulating {311} defects and is to be used only under intrinsic condtions.
- `-analytic`: Analytic model for {311} defects. Solves for the first two moments of the distribution.
- `-loop`: Analytic model for {311} defects including tranformation to dislocation loops.

## 2.3 `boroncluster` module

The `boroncluster` module considers the formation of boron intersititial clusters. Valid options are:

- `-two`: This considers the two species $BI_2$ and $B_3I$.

## 2.4 vcluster module

This module is intended for vacancy cluster formation. The maximum cluster size can be set by `param maxvacancyclustersize`.

## 2.5 The intf-segreg module

The `intf-segreg` is intended to allow for each dopant present in the system to be assigned an interface model independent of the models chosen for the other dopants. Most of the options require an Interface region to be sandwiched between an oxide and silicon regions. Model options must be set for each individual dopant. For example. typing the lines:

```
module params-intf-segreg
param Model.As ''fcm''
```

chooses the finite capacity model for arsenic. Valid model options are:

- `tbm`: The border between the interface and the silicon is handled with a simple linear segregation model while dopant and two separate segregation coefficients.
- `dbm`: A simple model which treats the interface as a separate region with infinite capacity for holding dopant and two separate segregation coefficients.
- `fcm`: Pull substitutional dopant into an interface which holds a finite dopant dose.
- `pair_pull_I`: A slight modification to the `fcm`, but it acts on dopant-interstitial pairs and releases an interstitial for each trapped pair.
- `pair_pull_Iwhole`: A slight modification to the `pair_pull_I`, but interstitials are not released.
- `pair_pull_IV`: A slight modification to the `pair_pull_I`, model, but it acts on both dopant-interstitial and dopant-vacancy pairs and releases interstitials and vacancies. This model has not been rigorously tested.
- `ifl3`: This model pulls out substitutional dopant at a fast rate without a reverse reaction.
- `ifl5`: This model pulls out substitutional dopant, dopant-interstitial pairs and dopant-vacancy pairs at a fast rate without a reverse reaction.

# Chapter 3

# Model Equations

## 3.1 Introduction

All models for diffusion, {311} and loops are described in detail by Alp. H. Gencer in his Ph.D. dissertation. These can obtained from our web site at:
*http : //eng.bu.edu/∼dunham/pubs/*.
Hence in the rest of the sections we will only describe the models not covered by his thesis.

## 3.2 Boron Cluster Model

These are the equations used by the `boroncluster` module. This module considers two species $(B_3I)^-$ and $(BI_2)^+$. $(BI_2)^+$ is important in the initial short times before the formation of {311} defects.

$$\mathrm{B}^- + \mathrm{I} \leftrightarrow \mathrm{BI}, \tag{3.1}$$

$$\mathrm{BI} + \mathrm{I} \leftrightarrow (\mathrm{BI}_2)^+. \tag{3.2}$$

The formation of $(B_3I)^-$ cluster is considered as a two step reaction from a B substitutional $(B^-)$. The only mobile species is $B_i$.

$$\mathrm{B}_i + \mathrm{B}^- \leftrightarrow (\mathrm{B}_2\mathrm{I})^0, \tag{3.3}$$

$$\tag{3.4}$$

$(B_2I)^0$ is found to reach equilibrium (with $B^-$ and $B_i$ ) within a short time. Hence we can assume,

$$C_{(\mathrm{B}_2\mathrm{I})^0} = K^{\mathrm{eq}}_{(\mathrm{B}_2\mathrm{I})^0} C_\mathrm{B} C_{\mathrm{I}^0} (p/n_i)^2 \tag{3.5}$$

$(B_3I)^-$ formation can proceed either by a reaction with a neutral $(B_i)^-$ or with a $(B_i)^0$. Hence we can write the two most probable reactions:

$$\mathrm{B}_i^- + (\mathrm{B}_2\mathrm{I})^0 \leftrightarrow (\mathrm{B}_3\mathrm{I})^- + \mathrm{I}^0, \tag{3.6}$$

$$\mathrm{B}_i^0 + (\mathrm{B}_2\mathrm{I})^0 \leftrightarrow (\mathrm{B}_3\mathrm{I})^- + \mathrm{I}^+. \tag{3.7}$$

It should be noted that under extrinsic conditions Eq. 6 is the more dominant pathway. The rates for these reactions are:

$$R_{\mathrm{B}_i^-} = 4\pi a \left[ C_{\mathrm{B}_i^-} C_{(\mathrm{B}_2\mathrm{I})^0} - \frac{C_{(\mathrm{B}_3\mathrm{I})^-} C_{\mathrm{I}^0}}{K_{\mathrm{B}_i^0}} \right], \tag{3.8}$$

$$R_{\mathrm{B}_i^0} = 4\pi a \left[ C_{\mathrm{B}_i^0} C_{(\mathrm{B}_2\mathrm{I})^0} - \frac{C_{(\mathrm{B}_3\mathrm{I})^-} C_{\mathrm{I}^+}}{K_{\mathrm{B}_i^+}} \right]. \tag{3.9}$$

Assuming ionization reactions are fast and in equilibirum, we know that,

$$\frac{K_{\mathrm{B}_i^-}}{K_{\mathrm{B}_i^+}} = \frac{C_{\mathrm{B}_i^0}}{C_{\mathrm{B}_i^-}} \frac{C_{\mathrm{I}^0}}{C_{\mathrm{I}^+}} = \frac{D_{\mathrm{B}}^+}{D_{\mathrm{B}}^0 K_{\mathrm{I}^+}}. \tag{3.10}$$

We can now calculate the total rate of formation of $(\mathrm{B}_3\mathrm{I})^-$ as,

$$R_{(\mathrm{B}_3\mathrm{I})^-} = R_{\mathrm{B}_i^-} + R_{\mathrm{B}_i^0}, \tag{3.11}$$

$$= 4\pi a \left[ C_{\mathrm{B}_i} C_{(\mathrm{B}_2\mathrm{I})^0} - \frac{C_{(\mathrm{B}_3\mathrm{I})^-} C_{\mathrm{I}^0}}{K_{\mathrm{B}_i^0}} \left( 1 + \frac{D_{\mathrm{B}}^+}{D_{\mathrm{B}}^0} \frac{p}{n_i} \right) \right]. \tag{3.12}$$

## 3.3 Vacancy Cluster Model

This module considers the formation of vacancy cluster of different sizes. This considers equations of the form:

$$V + V_n \Leftrightarrow V_{n+1}, \tag{3.13}$$

$$I + V_{n+1} \Leftrightarrow V_n. \tag{3.14}$$

## 3.4 Interface Segregation Models

These are the equations used by the `intf-segreg` module. Each model is written in terms of fluxes of a particular field from one region to another. The name of the model in the scripts is included in parentheses after each model title.

### Common Hangups

Since this model involves interactions between three regions, it is perhaps the most difficult to implement the proper syntax. It is best to look at the example files provided for this module. Three regions are required, oxide (Ox), interface (Int) and silicon (Si). This module only takes care of boundary operations. In order to maintain generality, it does not take into account required diffusion operators in any of these regions. They must be supplied independently.

### Traditional Bulk Model (`tbm`)

This model (Equation 3.15) performs a segegation between silicon and interface regions. In reality, one wants segregation between the silicon and the oxide. Although this is rather cumbersome, the interface is used here to allow this module to assign different models to individual dopants. A dopant assigned this model should have the same diffusion properties in the interface as in the oxide. The model allows the free movement of dopant between the interface and the oxide.

$$\begin{aligned} F^{\mathrm{P}}_{\mathrm{Ox}\Rightarrow\mathrm{In}} &= h_{\mathrm{fast}}(C^{\mathrm{Ox}}_{\mathrm{P}} - C^{\mathrm{In}}_{\mathrm{P}}), \\ F^{\mathrm{P}}_{\mathrm{Si}\Rightarrow\mathrm{In}} &= h_{\mathrm{Si}}(C^{\mathrm{Si}}_{\mathrm{P}} - m_{\mathrm{seg}} C^{\mathrm{In}}_{\mathrm{P}}), \end{aligned} \tag{3.15}$$

where the transport rate is determined by $h_{\mathrm{Si}}$. Equilibrium is defined in terms of the dopant concentrations at the silicon surface ($C^{\mathrm{Si}}_{\mathrm{P}}$), interface surface ($C^{\mathrm{In}}_{\mathrm{P}}$) and the oxide surface ($C^{\mathrm{Ox}}_{\mathrm{P}}$). The segregation coefficient determines the ration of dopant surface concentrations in equilibrium between the silicon and the interface.

$$m_{\mathrm{seg}} = \left(\frac{C^{\mathrm{Si}}_{\mathrm{P}}}{C^{\mathrm{Intf}}_{\mathrm{P}}}\right)^{*}. \tag{3.16}$$

Really, the interface is being treated as the oxide for this model. The parameter in Equation 3.16 is equivalent to the segregation coefficient reported in long term experiments (Equation 3.17).

$$m_{\mathrm{seg}} = \left(\frac{C^{\mathrm{Si}}_{\mathrm{P}}}{C^{\mathrm{Ox}}_{\mathrm{P}}}\right)^{*}. \tag{3.17}$$

### Double Bulk Model (`dbm`)

This model [2] uses two simple segregation equations (Equation 3.18). This model can agree with the traditional bulk model as far as the ration of concentrations at the surfaces of the oxide and silicon in equilibrium. However, it also allows for a pile up of dopant in the interface.

$$\begin{aligned} F^{\mathrm{P}}_{\mathrm{Ox}\Rightarrow\mathrm{In}} &= h_{\mathrm{Ox}}(C^{\mathrm{Ox}}_{\mathrm{P}} - m_{\mathrm{seg1}} C^{\mathrm{In}}_{\mathrm{P}}), \\ F^{\mathrm{P}}_{\mathrm{Si}\Rightarrow\mathrm{In}} &= h_{\mathrm{Si}}(C^{\mathrm{Si}}_{\mathrm{P}} - m_{\mathrm{seg2}} C^{\mathrm{In}}_{\mathrm{P}}), \end{aligned} \tag{3.18}$$

The two segregation coefficients ($m_{\mathrm{seg1}}$ and $m_{\mathrm{seg2}}$) are the key to creating the pile up while still agreeing with the traditional bulk model in equilibrium.

$$\begin{aligned} m_{\mathrm{seg1}} &= \left(\frac{C^{\mathrm{Ox}}_{\mathrm{P}}}{C^{\mathrm{In}}_{\mathrm{P}}}\right)^{*}, \\ m_{\mathrm{seg2}} &= \left(\frac{C^{\mathrm{Si}}_{\mathrm{P}}}{C^{\mathrm{In}}_{\mathrm{P}}}\right)^{*} \end{aligned}$$

The ratio of the two segregation coefficients is the same as the segregation coefficient for the traditional bulk model.

$$\begin{aligned} m_{\text{seg}} &= \frac{m_{\text{seg2}}}{m_{\text{seg1}}} \\ &= \left(\frac{C_{\mathrm{P}}^{\mathrm{Si}}}{C_{\mathrm{P}}^{\mathrm{In}}}\right)^{*} \left(\frac{C_{\mathrm{P}}^{\mathrm{In}}}{C_{\mathrm{P}}^{\mathrm{Ox}}}\right)^{*}. \\ &= \left(\frac{C_{\mathrm{P}}^{\mathrm{Si}}}{C_{\mathrm{P}}^{\mathrm{Ox}}}\right)^{*}. \end{aligned} \tag{3.19}$$

### Original Finite Capacity Model (`fcm`)

This model (Equation 3.20) [1] agrees with the double bulk model (Equation 3.18) for low surface concentrations. The fluxes are determined by the trapping of dopants into a density of empty traps ($N_0$) and the emission of dopants from a density of full traps ($N_f$). Trapping ($t_1$ and $t_2$) and emission ($e_1$ and $e_2$) coefficients not only determine the rate, but their ratios determine the equilibrium dopant concentrations at the surface ($C_{\mathrm{P}}^{\mathrm{Ox}}$ and $C_{\mathrm{P}}^{\mathrm{Si}}$).

$$\mathrm{P}_{\text{bulk}} \Leftrightarrow \mathrm{P}_{\text{intf}}$$

$$\begin{aligned} F_{\mathrm{Ox}\Rightarrow\mathrm{In}}^{\mathrm{P}} &= t_1 N_0 C_{\mathrm{P}}^{\mathrm{Ox}} - e_1 N_f, \\ F_{\mathrm{Si}\Rightarrow\mathrm{In}}^{\mathrm{P}} &= t_2 N_0 C_{\mathrm{P}}^{\mathrm{Si}} - e_2 N_f, \end{aligned} \tag{3.20}$$

### Dopant-Interstital Pair Pull Model with interstital release (`pair_pull_I`)

This model (Equation 3.22) is set to have the same equilibrium condition as Equation 3.18 and 3.20, but it acts on dopant-interstital pairs. An interstital is released for every pair trapped by the interface.

$$(\mathrm{PI})_{\text{bulk}} \Leftrightarrow \mathrm{P}_{\text{intf}} + \mathrm{I}_{\text{bulk}} \tag{3.21}$$

$$\begin{aligned} F_{\mathrm{Ox}\Rightarrow\mathrm{In}}^{\mathrm{P}} &= t_1 N_0 C_{\mathrm{P}}^{\mathrm{Ox}} - e_1 N_f, \\ F_{\mathrm{Si}\Rightarrow\mathrm{In}}^{(\mathrm{PI})} &= t_2 N_0 C_{(\mathrm{PI})}^{\mathrm{Si}} - e_2 N_f \pi_{\mathrm{P,I}} C_{\mathrm{I}^0}, \\ F_{\text{create}}^{\mathrm{I}} &= F_{\mathrm{Si},\Rightarrow\mathrm{In}}^{(\mathrm{PI})} \end{aligned} \tag{3.22}$$

where $\pi_{\mathrm{P,I}}$ is a variable from the five stream model used to account for the charged pairs. If the pairing reaction is in equilibrium then Equation 3.23 holds.

$$C_{(\mathrm{PI})} = \pi_{\mathrm{P,I}} C_{\mathrm{P}} C_{\mathrm{I}^0} \tag{3.23}$$

### Dopant-Interstital Pair Pull Model without interstital release (`pair_pull_Iwhole`)

This model (Equation 3.22) is set to have the same equilibrium condition as Equation 3.18 and 3.20, but it acts on dopant-interstital pairs. It is similar to Equation 3.22 but an interstital is not released when a pair is trapped at the interface.

$$(\mathrm{PI})_{\text{bulk}} \Leftrightarrow (\mathrm{PI})_{\text{intf}} \tag{3.24}$$

$$
\begin{aligned}
F^{\mathrm{P}}_{\mathrm{Ox}\Rightarrow\mathrm{In}} &= t_1 N_0 C^{\mathrm{Ox}}_{\mathrm{P}} - e_1 N_f, \\
F^{(\mathrm{PI})}_{\mathrm{Si}\Rightarrow\mathrm{In}} &= t_2 N_0 C^{\mathrm{Si}}_{(\mathrm{PI})} - e_2 N_f \pi_{\mathrm{P,I}} C^{*}_{\mathrm{I}^0},
\end{aligned} \tag{3.25}
$$

where $\pi_{\mathrm{P,I}}$ is a five stream diffusion model variable and is discussed briefly in Section 3.4.

#### Dopant-Defect Pair Pull Model with interstital release (`pair_pull_IV`)

This model (Equation 3.22) is set to have the same equilibrium condition as Equation 3.18 and 3.20, but it acts on dopant-interstital and dopant-vacancy pairs. An interstital is released for every pair trapped by the interface. The same is true for vacancies and dopant-vacancy pairs. This model has not been tested rigorously.

$$
\begin{aligned}
(\mathrm{PI})_{\mathrm{bulk}} &\Leftrightarrow \mathrm{P}_{\mathrm{intf}} + \mathrm{I}_{\mathrm{bulk}} \\
(\mathrm{PV})_{\mathrm{bulk}} &\Leftrightarrow \mathrm{P}_{\mathrm{intf}} + \mathrm{V}_{\mathrm{bulk}}
\end{aligned} \tag{3.26}
$$

$$
\begin{aligned}
F^{\mathrm{P}}_{\mathrm{Ox}\Rightarrow\mathrm{In}} &= t_1 N_0 C^{\mathrm{Ox}}_{\mathrm{P}} - e_1 N_f, \\
F^{(\mathrm{PI})}_{\mathrm{Si}\Rightarrow\mathrm{In}} &= t_2 N_0 C^{\mathrm{Si}}_{(\mathrm{PI})} - e_2 N_f \pi_{\mathrm{P,I}} C_{\mathrm{I}^0}, \\
F^{\mathrm{I}}_{\mathrm{create}} &= F^{(\mathrm{PI})}_{\mathrm{Si}\Rightarrow\mathrm{In}}, \\
F^{(\mathrm{PV})}_{\mathrm{Si}\Rightarrow\mathrm{In}} &= t_2 N_0 C^{\mathrm{Si}}_{(\mathrm{PV})} - e_2 N_f \pi_{\mathrm{P,V}} C_{\mathrm{V}^0}, \\
F^{\mathrm{V}}_{\mathrm{create}} &= F^{(\mathrm{PV})}_{\mathrm{Si}\Rightarrow\mathrm{In}},
\end{aligned} \tag{3.27}
$$

where $\pi_{\mathrm{P,I}}$ is a five stream diffusion model variable and is discussed briefly in Section 3.4.

#### Fast Loss Model - 3 stream version (`ifl3`)

This model pulls out substitutional dopant at a very fast rate. One might use it for modeling fast outdiffusion. This model has not been vigorously tested. It is generally used to test if the interface model (as opposed to the diffusion model) is hindering the amount of dose loss.

$$
F^{\mathrm{P}}_{\mathrm{Si}} = -\sigma_{\mathrm{fast}} C^{\mathrm{Si}}_{\mathrm{P}}
$$

#### Fast Loss Model - 5 stream version (`ifl5`)

This model pulls out substitutional dopant, dopant-interstital pairs and dopant-vacancy pairs at a very fast rate. One might use it for modeling fast outdiffusion. This model has not been vigorously tested. It is generally used to test if the interface model (as opposed to the diffusion model) is hindering the amount of dose loss.

$$
\begin{aligned}
F^{\mathrm{P}}_{\mathrm{Si}} &= -\sigma_{\mathrm{fast}} C^{\mathrm{Si}}_{\mathrm{P}} \\
F^{(\mathrm{PI})}_{\mathrm{Si}} &= -\sigma_{\mathrm{fast}} C^{\mathrm{Si}}_{(\mathrm{PI})} \\
F^{(\mathrm{PV})}_{\mathrm{Si}} &= -\sigma_{\mathrm{fast}} C^{\mathrm{Si}}_{(\mathrm{PV})}
\end{aligned}
$$

# Bibliography

[1] F. Lau, L. Mader, C. Mazure, Ch. Werner, M. Orlowski, "A Model for Phosphorus Segregation at the Silicon-Silicon Dioxide Interface", *Applied Physics A*, Vol. **A49**, 1989, pp. 671-675

[2] M. Akazawa, T. Aoki, S. Tazawa, Y. Sato, "Phosphorus Pile-Up Model for Si$O_2$-Si Interface of p-Channel MOSFETS", International Conference on Simulation of Semiconductor Process and Devices, 1996, pp. 29-30

# Appendix A

# Naming conventions in PMM

Most modules in PMM expect certain fields and regions with certain names. This allows the scripts to determine the type of a field and region. We will first describe the default naming convention and then explain how it can be changed.

## A.1 Default names in PMM

**Single species concentrations:** All concentrations start with a capital `C` and is followed by the name of the chemical species it stands for. Thus `CB` is boron concentration and `CAs` is arsenic concentration. The species `I` is interstitials and `V` is vacancies, making `CI` interstitial concentration and `CV` vacancy concentration.

**Pair concentrations:** Concentrations of pairs of species is obtained in a similar way, by adding two species names to the capital `C`. The order is important here. If one of the species is a dopant and the other a point defect, the dopant comes first. If both species are dopants, the acceptor comes first. Thus `CBI` is the concentration of boron-interstitial pairs, `CAsV` is the concentration of the arsenic-vacancy pairs and `CBAs` is the concentration of boron-arsenic pairs.

**Grain boundary concentrations:** Concentrations of species within a poly silicon grain boundary are specified similar to above, but by appending a `_gb` to the end. Thus `CAs_gb` is the concentration of arsenic within a grain boundary. This is used only by the `poly` module.

**KPM moments:** KPM moments have names starting with `M_` and ending with the name of the precipitate. Thus, `M_P` stands for the moments of phosphorus precipitates and `M_BIC` stands for boron-interstitial clusters. The only exception is {311} defects and dislocation loops, which have the name `M_I`.

**Region names:** Region names are only used by PMM modules that have a segregation operator, and therefore need to check the type of the neighboring region. Only the first couple of letters are checked in the region names, thus the user has the freedom to use long names, or differentiate between, say, two silicon regions. Here is a list of default region names:

| **Name starts with** | **Type of region** |
|---|---|
| `Si` | Silicon |
| `Ox` | Silicon Dioxide ($SiO_2$) |
| `Poly` | Poly-silicon |
| `Int` | Generic interface region |

## A.2 Changing default names

Since names are a matter of preference, different users might want to use different naming conventions. In such a case you'll need to follow a two simple steps to use your own convention:

**Create a new names file:** You should copy the module file `names-default.tcl` to a different name, and then make modifications on it for the names of your liking. Try to be as consistent as possible for your own sake.

**Source the file:** Call the file you created on top of every file you want to use it with a `module` command. If you really want to use it with every file you create, you may add the `module` command to your `~/.dopdeesrc` or `~/.alamoderc` file AFTER the call to PMM initialization.

# Appendix B

# Contacts

For additional information or bug reports contact one of the following individuals:

Srinivasan Chakravarthi
Boston University
8 St Mary's Street
Boston, MA 02215.
tel: 617-353-5885
email:srini@bu.edu

Brendon Murphy
Boston University
8 St Mary's Street
Boston, MA 02215.
tel: 617-353-5885
email:murphyb@bu.edu

Alp H Gencer
*Presently Working at*
TCAD Business Unit
Avant! Corporation.

Scott T Dunham
Boston University
8 St Mary's Street
Boston, MA 02215.
tel: 617-353-9845
email:dunham@bu.edu