COMSOL FinFET Model

Contents

Chapter 1. Introduction

This project contains a COMSOL Multiphysics model for a fin field-effect transistor (FinFET), freely available for personal and research use. This documentation aims to help users in using and modifying this model for their purposes by providing details regarding physics contained in the model, numerical details, and basic results.

Thus far, versions of this model has been successfully used in two papers published in the *IEEE Transactions on Electron Devices*, demonstrating its potency. While formal justifications of the choice of model parameters have been discussed in these published works, this documentation will serve as a manual-style resource for users to use and tune the model for their own personal or research projects.

Please note that parts of this documentation is still being updated and subject to change.

Related information

FinFET and TCAD Models *(on page 4)*

Model *(on page 5)*

Copyright *(on page 20)*

Chapter 2. FinFET and TCAD Models

The FinFET is an advanced transistor architecture commonly used in modern computer chips, and has been continuously used in state-of-the-art processors since their first commercial deployment in the Intel Ivy Bridge chips in 2012. The FinFET architecture has been in use for a variety of modern process nodes, such as the Intel 4 node, and the TSMC N3 node.

In order to predict the behavior of future scaled transistors and obtain physical insights to their operation to aid research and development, physics-based technology CAD (TCAD) simulations are important. Compared to fitted models such as BSIM-CMG and alike, TCAD models allows one to examine spatially distributed microscopic quantities such as charge and current, at the cost of slower run speed, which forbids their integration into large scale circuit simulators.

Chapter 3. Model

This section will describe various components of the COMSOL model, including how the geometry is set up, overview of transistor physics, simulation set up and how they can be executed, as well as important numerical considerations.

Related information

Geometry *(on page 5)*
Simulation Setup *(on page 7)*
Simulation Execution *(on page 17)*
Numerical Considerations *(on page 19)*

Geometry

The FinFET transistor model is in 2-D, representing the top view of a single fin. The simulation domain is surrounded by air to accommodate stray fields that will otherwise be out of bounds of the transistor domain itself.

This particular model is modeled after the a 7-nm device. Details of the justifications of these parameters can be found in this published paper.

Table 1. FinFET Key Dimensions

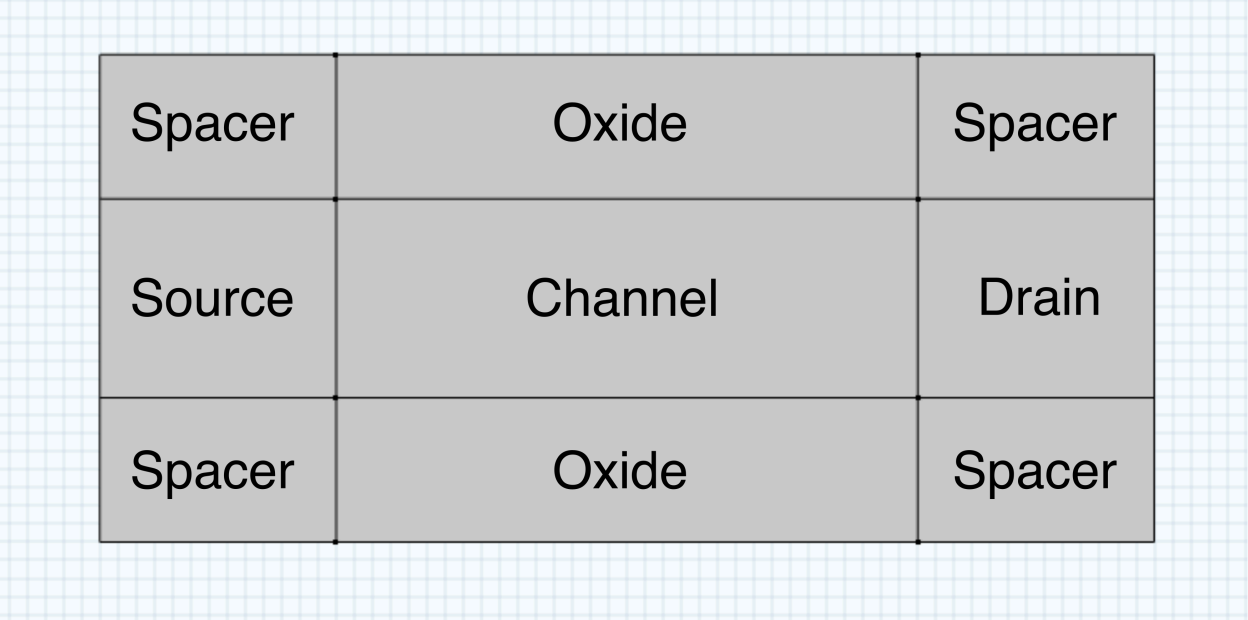
Parameter	Value
Gate Length	16 nm
Fin Width	5.5 nm
Fin Height	50 nm
Spacer Length	6.5 nm
Oxide Thickness	4 nm

Related information

Active Region *(on page 6)*
Air Domain *(on page 6)*

Active Region

The active region is consisted of a source, drain, channel, oxide, and spacer regions. While the oxide and spacer layers are all separate rectangles, the channel, source, and drain regions are all encapsulated in a single rectangle entity, separated by mesh control lines.

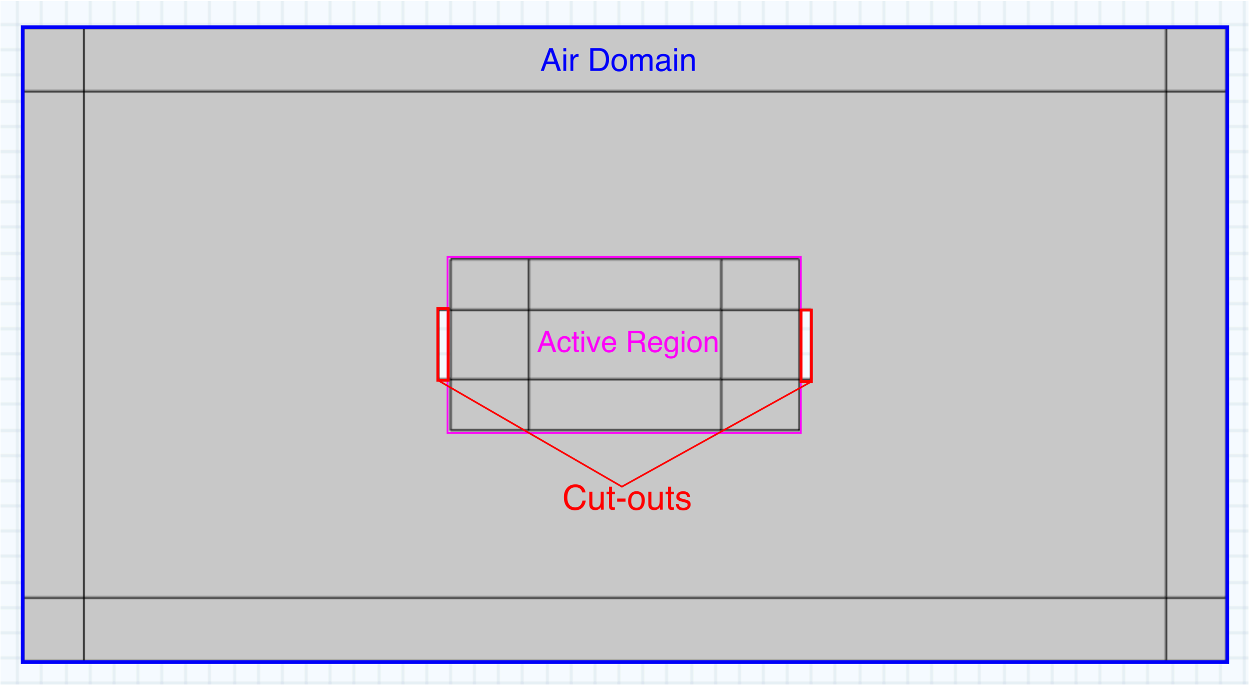


Air Domain

The air domain that surrounds the active region is a simple 100 nm x 50 nm rectangle that acts as the background layer to the active device region. The infinite element domain function in COMSOL is used to model the numerical behaviors of this domain.

Initially, several domains are used for testing purposes, these domains are no longer used.

Two cut-outs had to be constructed beside the source and drain contacts to allow boundary conditions to be properly applied.



Simulation Setup

Basic transport in the transistor is handled by a modified drift-diffusion equation in conjunction with the usual Poisson's equation. Field-dependent Caughey-Thomas electron mobility and quantum confinement via the modified local density approximation (MLDA) have been implemented on top of the well-known drift-diffusion equation.

Related information

Transport and Electrostatics *(on page 7)*
Quantum Confinement Correction *(on page 9)*

Transport and Electrostatics

The transport simulation is split up into domain that handles both transport and electrostatics (the source, channel, and drain regions), and domains that handles only electrostatics (spacer and oxide regions).

Internal boundary conditions uses the insulator interface boundary condition between the semiconductor (source, drain, and channel) and the spacer and oxide, and the continuity boundary condition at the interface that lies between the channel and the source and drain. All contacts have been realized using the metal contact boundary conditions, with the outside of the air domain handled using homogeneous Neumann boundary conditions, which is the default boundary for COMSOL.

Fermi-Dirac statistics along with finite-element quasi-formulation discretization method was used for the transport module.

Doping Profile

Doping in the channel is specified using analytic doping models following a Gaussian profile from values in the source and drain to the background doping of the channel. Specific settings can be found in the modules named "Source Doping" and "Drain Doping" in the model.

Warning:
The sharpness of the Gaussian is controlled by the junction depth parameter in the doping profile. Too sharp of a doping profile (low junction depth) could cause convergence issues.

Table 2. Doping Concentrations

Region	Type	Doping Concentration
Source	n+	$10^{20}\ cm^{-3}$
Channel	p	$5 \times 10^{17}\ cm^{-3}$
Drain	n+	$10^{20}\ cm^{-3}$

Mobility

The only non-constant mobility model utilized in this simulation is the Caughey-Thomas field-dependent mobility model. As an acceptable match with experimental results of similar process node was achieved by only tuning parameters of this model, no further refinement was conducted.

The Caughey-Thomas model equation is as follows

$$\mu(\boldsymbol{r}) = \mu_{eff}\left(\frac{1}{\left(1+\frac{\mu_{eff}F(\boldsymbol{r})}{v_{sat}}\right)^{\beta}}\right)^{\frac{1}{\beta}}$$

$$F(\boldsymbol{r}) = \frac{J(\boldsymbol{r})\cdot E(\boldsymbol{r})}{\left|J(\boldsymbol{r})\right|}$$

$$\boldsymbol{r} \in (x, y)$$

Table 3. Caughey-Thomas Mobility Parameters

Parameter	Value
μ_{eff}	$340\ cm^2/Vs$

Table 3. Caughey-Thomas Mobility Parameters (continued)

Parameter	Value
β	2 (electron) / 1 (hole)
v_{sat}	$2.1\times 10^{7}\ {}^{cm}/_{s}$

Related information

Numerical Considerations *(on page 19)*

Quantum Confinement Correction

The fin of the FinFET is extremely thin, which results in a quantum well like system. This well energetically and spatially confines the charge within the channel, which results in changes to the concentration and spatial distribution of free charges within that is unaccounted for in a purely classical drift-diffusion formulation. To account for this change, the modified local density approximation (MLDA) has been used, which shows good conformation with more accurate Schrödinger-Poisson formulations as shown here.

The MLDA formulation is a modification to the density of states, which is used to compute charge. Typically, charge in a system governed by the Fermi-Dirac distribution can be calculated as follows:

$$\int D(E)f(E-E_f)dE$$

where $D(E)$ is the density of available electron per energy, and $f(E-E_f)$ is the Fermi-Dirac distribution.

Typically, the density of states in a 3-D unconfined system is as follows

$$D(E)=\frac{(m^*)^{\frac{3}{2}}}{2\pi\hbar}\sqrt{2(E-E_c)}$$

where $\hbar$ is the reduced Planck's constant, m^* is the electron effective mass, and E_c is the conduction band edge

In the MLDA formulation, the density of state function $D(E)$ is formulated as follows

$$n_{QC}=\sum_{i=1}^{3}\int_{0}^{\infty}D_i(\varepsilon'-z)f(\varepsilon'-E_F-E_c)d\varepsilon'$$

$$D_i(\varepsilon,z)=\frac{g_i\sqrt{m_x^{*i}m_y^{*i}m_z^{*i}}}{\hbar^2\pi^2}\sqrt{\varepsilon'}\left(1-\mathrm{sinc}(\mathfrak{z}K)-\mathrm{sinc}((t_b-\mathfrak{z})K)+\mathrm{sinc}(t_bK)\right)$$

$$K=2\sqrt{\frac{2m_q^{*i}\varepsilon'}{\hbar^2}}$$

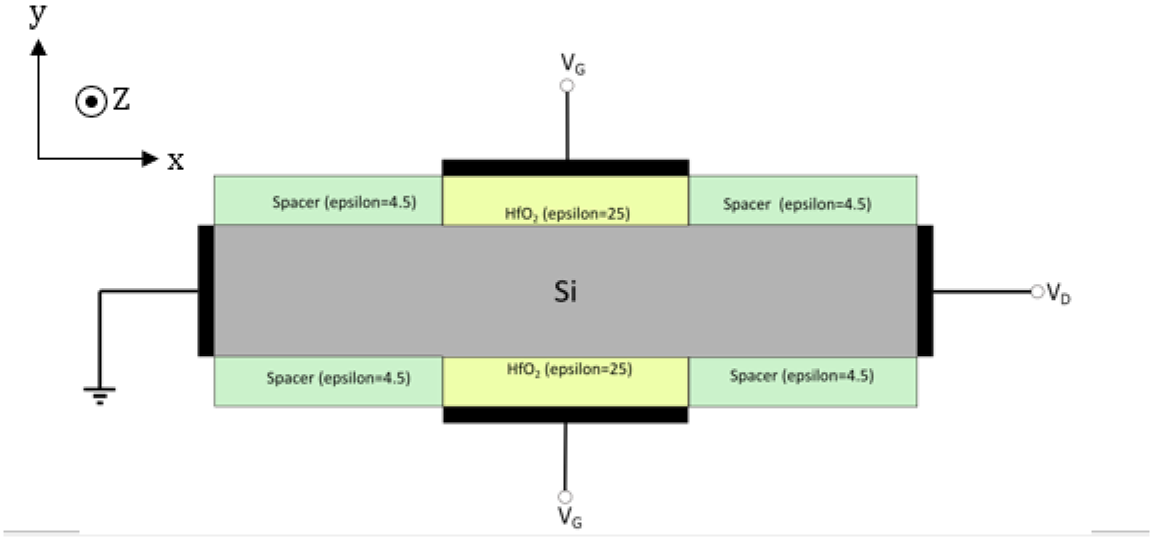
$$\varepsilon'=\varepsilon+\frac{h^2\pi^2}{2m_q^{*i}t_b^{2}}$$

where $\mathfrak{z}$ is the distance to the semiconductor-insulator interface at $\mathfrak{z} = 0$, and effective masses are defined in the table below for Si

Table 4. Effective Masses

Valley	m^*_x	m^*_y	m_z^*	m_q^*
[1 0 0]	$0.196\ m_e^*$	$0.196\ m_e^*$	$0.914\ m_e^*$	$0.196\ m_e^*$
[0 1 0]	$0.196\ m_e^*$	$0.914\ m_e^*$	$0.196\ m_e^*$	$0.914\ m_e^*$
[0 0 1]	$0.914\ m_e^*$	$0.196\ m_e^*$	$0.196\ m_e^*$	$0.196\ m_e^*$

The device is assumed to have the following orientation, along with a wafer direction of [1 0 0], *i.e.*, [1 0 0] in the direction of the fin height (z)



Related information

Implementation of Quantum Confinement *(on page 10)*

Custom Functions *(on page 11)*

Implementation of Quantum Confinement

The MLDA formalism is implemented via an auxiliary equation that calculates the ratio of the quantum-confined density vs. the classically calculated charge density. This calculation is done using the "Domain ODE and DAE" module, with a variety of custom functions. Inside this module, the source term implements the calculation of this ratio N_{QC} .

Note:
The command "nojac" on integral terms were used to assist convergence of the formulation and decrease computational time, as the Jacobian of these terms are not needed.

Note:
This correction is **only** applied to electrons, and not holes, as they do not account for a large portion of transport current in this n-MOSFET setup.

The solved term, N_{QC} is coupled into the semiconductor module using the degeneracy factor that was originally used to correct differences between the Fermi-Dirac and the Boltzmann distribution. This modification can be found in the "Equation View" of the main semiconductor module.

Note:
A lower bound was implemented to improve convergence, where if the quantum-corrected charge is less than 0.001% of the classical charge, the charge is simply 0.001%. This is because near-zero charge causes convergence issues in the simulator.

Warning:
A value smaller than 0.001% (1e-5 in the formula) may cause non-convergence.

This implementation of quantum confinement have been calibrated with an in-house non-equilibrium Green's function (NEGF) simulator for a the same FET, results within an acceptable error margin were obtained.

Related information

Custom Functions *(on page 11)*
Numerical Considerations *(on page 19)*
Quantum Confinement Correction *(on page 9)*

Custom Functions

Various custom functions have been implemented to assist the overall implementation of the quantum confinement correction formalism. These functions are defined both globally (under Global Definitions) and within the simulator (under "Variables_domain" in the simulation component comp1).

The final function executed in the calculation for the quantum corrected charge ratio N_{QC} is int196 *(on page 16)* and int 914 *(on page 17)*. It is suggested that the users start from there in order to understand the full implementation.

Related information

Implementation of Quantum Confinement *(on page 10)*
J0 *(on page 12)*
E_prime *(on page 13)*
K_func *(on page 13)*
J0Func *(on page 14)*
fermi *(on page 14)*
Int_Di *(on page 15)*
lim *(on page 16)*
int196 *(on page 16)*
int914 *(on page 17)*

J0

Implements the sinc(x) function with a lower bound

Syntax

J0(*x*)

Note:
For x<0.01, the value is automatically evaluated to 1. Otherwise, the value is evaluated as sin(x)/x

Variables

Table 5.

Parameter	Description
x	input

Related information

Custom Functions *(on page 11)*
Quantum Confinement Correction *(on page 9)*

E_prime

Calculates ε' as described in the MLDA formalism

Syntax

E_prime(*E*,*mi_q*,*tb*)

Variables

Table 6.

Parameter	Description
E	Energy to be integrated over
mi_q	Confinement effective mass for the ith direction
tb	FinFET fin width

Related information

Custom Functions *(on page 11)*

Quantum Confinement Correction *(on page 9)*

K_func

Implements K as described in the MLDA formalism

Syntax

K_Func(*E_prime*,*mi_q*)

Variables

Table 7.

Parameter	Description
E_Prime	ε' as described in the MLDA formalism
mi_q	Confinement effective mass for the ith direction

Related information

Custom Functions *(on page 11)*

Quantum Confinement Correction *(on page 9)*

J0Func

Implements the integrand of the MLDA formalism with a lower bound.

Syntax

J0Func(*E*,*z*, *mi_q*,*tb*)

Note:
If the output of this function is smaller than 1e-10, it is rounded to 1e-10

Variables

Table 8.

Parameter	Description
E	Energy to be integrated over
z	Distance from a semiconductor-insulator interface at z=0
mi_q	Confinement effective mass for the ith direction
tb	FinFET fin width

Related information

Custom Functions *(on page 11)*

Implementation of Quantum Confinement *(on page 10)*

fermi

Implements the Fermi-Dirac Distribution

Syntax

fermi(*E*,*mi_q*,*tb*,*nu*,*vth*)

Variables

Table 9.

Parameter	Description
E	Energy to be integrated over

Table 9. (continued)

Parameter	Description
mi_q	Confinement effective mass for the ith direction
tb	FinFET fin width
nu	Distance between fermi level and conduction band edge
Vth	Thermal voltage kBT (~25 meV at room temperature)

Related information

Custom Functions *(on page 11)*

Implementation of Quantum Confinement *(on page 10)*

Int_Di

Implements the density of states multiplied by Fermi-Dirac distribution

Syntax

Int_Di(*gi,mi_x,mi_y,mi_z,mi_q,tb,E,nu,Vth*)

Variables

Table 10.

Parameter	Description
gi	Valley degeneration parameter (2 is used to account for both spins)
mi_x	x-direction effective mass
mi_y	y-direction effective mass
mi_z	z-direction effective mass
mi_q	Confinement effective mass for the ith direction
tb	FinFET fin width
E	Energy to be integrated over

Table 10. (continued)

Parameter	Description
nu	Distance between fermi level and conduction band edge
Vth	Thermal voltage kBT (~25 meV at room temperature)

Related information

Custom Functions *(on page 11)*

Implementation of Quantum Confinement *(on page 10)*

lim

Contained within "Variables_domain" of the component, limit of integration for charge density integration

Note:

The default upper bound of $10k_BT$ was used as a reasonable upper bound, it was tested to ensure that there are noticeable differences beyond this value

Syntax

None

Related information

Custom Functions *(on page 11)*

Implementation of Quantum Confinement *(on page 10)*

int196

Contained within "Variables_domain" of the component, implements the whole MLDA integral with a confinement effective mass of $0.196m_e^*$

Note:

This variable is contained within the component such that simulation variables such as semi.eta_n can be used as a part of its argument

Related information

Custom Functions *(on page 11)*

Implementation of Quantum Confinement *(on page 10)*

int914

Contained within "Variables_domain" of the component, implements the whole MLDA integral with a confinement effective mass of $0.914m_e^*$

Note:
This variable is contained within the component such that simulation variables such as semi.eta_n can be used as a part of its argument

Related information

Custom Functions *(on page 11)*

Implementation of Quantum Confinement *(on page 10)*

Simulation Execution

There are two studies in embedded in this model, "DC Solution" and "SS Perturbation", corresponding to a general DC sweep and a DC small-signal perturbation around a bias point. This section will describe basic settings used in setting up these simulations. These settings are mostly common across both studies, but settings specific to a single study will be described where applicable.

The typical run time for a DC sweep is ~18 hours, the run time for a small-signal perturbation simulation can be up to 20+ hours. Simulations of a full I_{ds}-V_{ds} family of curves could take up to one week. These simulations cannot be paused, hence, an always-on computer is recommended. One component that contributes heavily to the long run time is the custom modified local density approximation formulation – if the simulator is re-formed using the newer density gradient approximation available in COMSOL 5.5 or above, it may be possible to have a much faster simulation.

Sweeps

Each solver utilizes sweeps that gradually increases voltage and other parameters from the equilibrium, with each step using the previous step as an initial guess. This scheme allows for the simulations to properly converge.

The DC study utilizes two sweep. The first sweep is used to achieve a desired drain voltage, and the second sweep obtains data at all gate desired gate voltages for the single desired drain voltage. This scheme is used as usually the I_{ds}-V_{gs} curves are desired.
The small signal perturbation study simply runs five sequential points around a central bias point, each with a perturbed drain or gate voltage of dV.

Common to both studies, there is a final sweep of the parameter CT_P, this gradually ramps up the effect of the Caughey-Thomas mobility model, which is necessary for the convergence of the simulation, as this model causes the governing equations to become extremely non-linear.

⚠ **Warning:**
Small ramp steps of the CT_P parameter near 1 is needed to ensure solution accuracy. The current ramp is set to 0.1, 0.5, 0.8, and 1 and it is not recommended to use ramps that are coarser than what is currently set. One symptom of faulty data from too large of a ramp step are spikes in the output conductance as a function of gate voltage.

Solver Setup

A direct-segregated approach was taken to minimize RAM usage and allow for more simulations to be executed in parallel. A scaling factor of 1e-10 was used to aid the convergence, and can be found under "Dependent Variable" settings. Beyond these default scaling, the scaling for the quantum correction variable N_{QC} is set to 1, which can be found under the N_{QC} tab that is a sub-component to "Dependent Variables" for each solver.

The CT_P parameter ramp is implemented under "Previous Solution", which is nested under "Parametric 1" for each solver.

⚠ **Warning:**
Ensure this value is present in all simulations that needs to ramp up the Caughey-Thomas mobility. The simulation would not converge despite seemingly-correct setup in COMSOL 5.4 if this was not set.

The segregated solver first solves for charge and electrical driving force for the Caughey-Thomas mobility, followed by the quantum correction factor N_{QC}, and finally followed by overall electrical potential. The maximum number of iterations have been changed to 100 for each step to allow for convergence in some extreme cases.

Numerical Considerations

Several important points must be kept in mind in order for the model to converge with the current formulation.

Carriers

Currently, the model is set to run with electron and hole solutions. The simulation **will** not converge if set to single carrier only, even if the minority carrier does not contribute to transport.

Quantum-Corrected Densities

As discussed in quantum confinement correction *(on page 10)*, the quantum corrected carrier densities should be truncated at low values to avoid non-convergence issues.

Doping

As discussed in doping profile *(on page 8)*, a sharp transition of dopant concentration would lead to non-convergence.

Caughey-Thomas Mobility

The implementation of Caughey-Thomas mobility is a major contribution factor to non-convergence as its formulation highly impacts the linearity of the drift diffusion equations. If the non-linearity of this effect is incorrectly ramped up, non-convergence could occur.

Chapter 4. Copyright

This model is delivered "as-is", with no warranty expressed or implied, and with no guarantee for accuracy or applicability for any purpose.