ViennaTS User Guide

(Working Draft)

Copyright Notice

Table of Contents

About This Software

ViennaTS

Developer repository for ViennaTS, a C++, OpenMP-parallelized Topography simulator.

ViennaTS is currently in a prototype state.

System Requirements

* C++ compiler

* OpenMP

* Boost C++ Libraries

* SPRNG

** HDF5 – only required to import HDF5 file types*

Currently Supported Operating Systems

* GNU/Linux

Authors and Contact

Current contributors: Lado Filipovic, Paul Manstetten, and Josef Weinbub

Contact us via: viennats@iue.tuwien.ac.at

Founder and initial author was Otmar Ertl; not active anymore.

ViennaTS was developed under the aegis of the 'Institute for Microelectronics' at the 'TU Wien'. To understand more about the software implementation, you are encouraged to read through the dissertation of Otmar Ertl [1].

Getting Started

Building Instructions

To build ViennaTS, clone the repository and issue the following suggested commands:

```
$> cd viennats-dev  # the checked-out GIT folder
$> mkdir build      # the build folder
```

Configure the build, default build type is the 'optimized/release' mode:

```
$> cd build/
$> cmake ..
```

Now build the 'viennats' executable

```
$> make
```

CMake Options

CMAKE_BUILD_TYPE   = debug, (release) # Turn off/on optimizations (default: release, i.e., optimized mode)

Running the Software

./viennats [parameters_file]

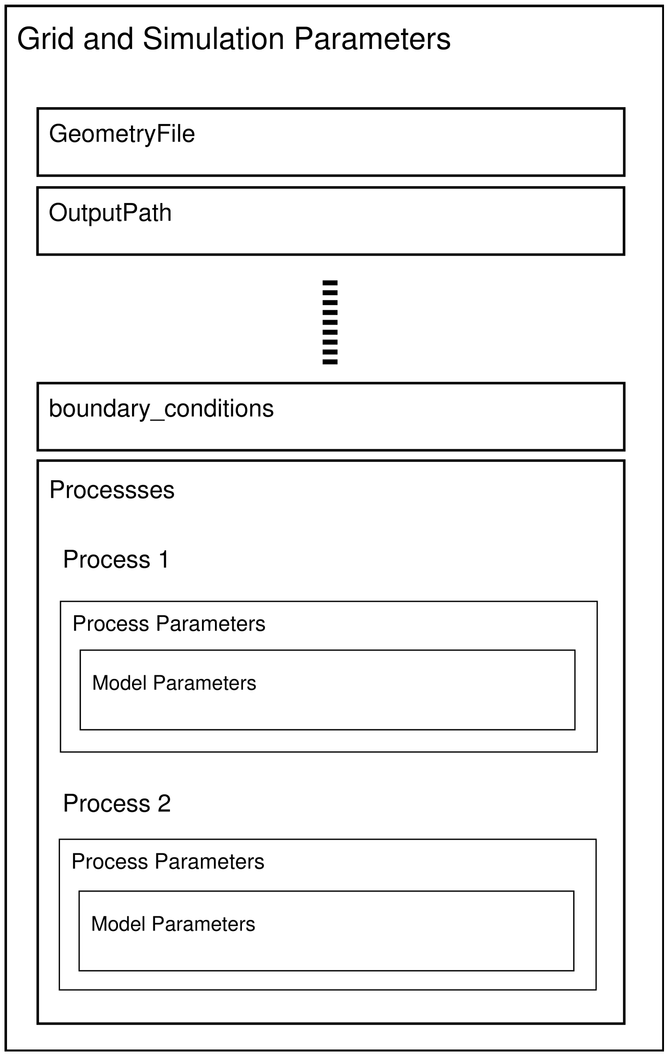
Examples

Several examples are available in the viennats-dev/examples/ folder

All parameters are passed through the parameters file, the format of which is described in the next section.

Populating the Parameters File

The parameters file is build hierarchically. One of the simulation parameters are the processes, which themselves have process-dependent parameters. One of the process parameters are the model-specific parameters.



Grid and Simulation Parameters

GeometryFile

OutputPath

boundary_conditions

Processses

Process 1

Process Parameters

Model Parameters

Process 2

Process Parameters

Model Parameters

Grid and Simulation Parameters

The grid and simulation parameters are listed below in two tables. The first table gives the list of available parameters and their types and the second table gives their description and purpose.

Format of the Parameters File

[parameter] = [parameter_value];

Available Parameters

Parameter	**Parameter values**	**Type**
geometry_file geometry_files GometryFile GeometryFiles	“FileName” “FileName1,FileName2,...,FileNameN”	string
output_path OutputPath	“FolderName”	string
surface_geometry	true/false	bool
report_import_errors	true/false	bool
cfl-condition CFL-Condition	number	double
input_scale InputScale	number	double
grid_delta GridDelta	number	double
input_transformation	(num1, num2, num3)	vector<int>
input_shift	(num1, num2, num3)	vector<double>
default_disk_orientation	(num1, num2, num3)	vector<double>
ignore_materials	(num1, num2, ... numN)	vector<int>
change_input_parity	true/false	bool
random_seed RandomSeed	number	int
num_dimensions Dimensions	2 or 3	int

Parameter	Parameter values	Type
omp_threads OpenMP_threads	number	int
domain_extension Domain_Extension	number	double
receptor_radius ReceptorRadius	number	double
further_tracking_distance FurtherTrackingDistance	number	double
print_vtk	true/false	bool
print_dx	true/false	bool
print_velocities	true/false	bool
print_coverages	true/false	bool
print_rates	true/false	bool
print_materials	true/false	bool
print_statistics	true/false	bool
max_extended_starting_position	number	int
open_boundary	“[+ | -] & [x | y | z]”	char array
remove_bottom	true/false	bool
snap_to_boundary_eps	number	double
process_cycles	number	int
material_mapping	(num1, num2, ... numN)	vector<int>
add_layer	number	int
boundary_conditions	{ {BOUNDARY_x1, BOUNDARY_x2}, {BOUNDARY_y1, BOUNDARY_y2}, {BOUNDARY_z1, BOUNDARY_z2} }	REFLECTIVE, SYMMETRIC, INFINITE, PERIODIC, EXTENDED
processes	{ { PROCESS1 }, { PROCESS2 }, ... { PROCESSn } }	

Description of the Available Parameters

Parameter	Default Value	Description
geometry_file geometry_files GometryFile GeometryFiles	REQUIRED	Location and name of the input geometry; when multiple surfaces are imported, separate with comma; quotation marks are required
output_path OutputPath	REQUIRED	Location an dname of the output folder; quotation marks are required
surface_geometry	false	True if the input geometry describes a surface (or surfaces). False if it describes a volume
report_import_errors	true	True if you wish to be informed of errors during geometry import; will also interrupt simulation in case an error is found; Recommended
cfl-condition CFL-Condition	REQUIRED	Maximum time step for level set method (in grid units); Recommended maximum 0.5 (double)
input_scale InputScale	1	Scale the input geometry by the input_scale
grid_delta GridDelta	REQUIRED	Grid spacing between grid points in each direction
input_transformation	(0, 0, 0)	Transformation for the input geometry for each axial direction
input_shift	(0, 0, 0)	Shift the geometry by the desired values for each axial direction
default_disk_orientation	REQUIRED	Default normal orientation
ignore_materials	-empty-	IDs of materials which are to be ignored
change_input_parity	false	Swap order of vertices in an elemnet in case the transformation has changed the triangle/tetrahedron orientation
random_seed RandomSeed	-empty-	Seed for the random number generator
num_dimensions Dimensions	REQUIRED	Number of dimensions

Parameter	Default Value	Description
omp_threads OpenMP_threads	0	Number of parallel threads to be used during the simulation
domain_extension Domain_Extension	0	Domain extension in case of EXTENDED boundary conditions
receptor_radius ReceptorRadius	0.8	Radius of the receptor disks
further_tracking_distance FurtherTrackingDistance	3	Number of grid cubes for a particle to continue to be tracked after surface intersection is recorded
print_vtk	true	Output surfaces in the vtk file format
print_dx	false	Output surfaces in the dx file format
print_velocities	false	Include the velocities in the output file
print_coverages	false	Include the coverages in the output file
print_rates	false	Include the surface rates in the output file
print_materials	false	Include the materials numbers in the output file
print_statistics	false	Output a statistics file
max_extended_starting_position	1000	Number of grid points by which to extend in case of extended boundaries
open_boundary	REQUIRED	One boundary must be open infinite as it is where the ray tracing particles originate
remove_bottom	true	True if the location of the bottom of the simulation space can be ignored
snap_to_boundary_eps	1e-6	Small number to ensure the simulation snaps to the full extents of the grid space
process_cycles	1	Number of times to perform the same processing sequence
material_mapping	-empty-	Change how materials are mapped by changing their IDs

Parameter	Default Value	Description
add_layer	0	Generate more layers to replicate the top-most surface, required for deposition steps
boundary_conditions	REQUIRED	Define the boundary conditions for the simulation grid; can be REFLECTIVE (SYMMETRIC), INFINITE, PERIODIC, or EXTENDED
Processes	REQUIRED	List of processes to be executed in sequence; the process parameters are described in the section which follows

Process Parameters

Format in the Parameters File

```
processes {

{

        [process_parameter] = [process_parameter_value];

}

};
```

Available Process Parameters

Process parameter	Process parameter values	Type
process_time	number	double
smoothing_max_curvature	number	double
smoothing_min_curvature	number	double
smoothing_material_level	number	int
smoothing_max_iterations	number	int
add_layer	number	int
output_times_periodicity	number	int
output_times_period_length	number	double
initial_output	true/false	bool

Process parameter	**Process parameter values**	**Type**
final_output	true/false	bool
print_velocities	true/false	bool
print_coverages	true/false	bool
print_rates	true/false	bool
print_materials	true/false	bool
max_time_step MaxTimeStep	number	double
iteration_cycles IterationCycles	number	int
start_iteration_cycles StartIterationCycles	number	int
model_name	“ModelName”	string
output_times	{time1, time2, … timeN}	vector<double>
finite_difference_scheme	ENGQUIST_OSHER_1ST_ORDER ENGQUIST_OSHER_2ND_ORDER LAX_FRIEDRICHS_1ST_ORDER LAX_FRIEDRICHS_2ND_ORDER	
dissipation_coefficient	number	double
partition_data_structure	NEIGHBOR_LINKS_ARRAYS, FULL_GRID, UP_DOWN_LINKED_TREE	
partition_splitting_strategy	SPATIAL_MEDIAN, OBJECT_MEDIAN, SURFACE_AREA_HEURISTIC	
partition_surface_area_heuristic_lambda	number	double

parameters	{ [parameter1] = [value];
	[parameter2] = [value];
	...
	[parameterN] = [value]; };
		model-specific parameters

Description of the Available Process Parameters

Process parameter	Default value	Description
process_time	0	Duration of the current process in seconds
smoothing_max_curvature	max<double>	Maximum parameter for smoothing surface level set every time step
smoothing_min_curvature	min<double>	Minimum parameter for smoothing surface level set every time step
smoothing_material_level	0	ID of material to be smoothed, if smoothing is applied
smoothing_max_iterations	100	Number of smoothing iterations to perform, if smoothing is applied
add_layer	0	Generate more layers to replicate the top-most surface, required for deposition steps
output_times_periodicity	1	Periodicity of output with output_times_period_length
output_times_period_length	0	Periodicity of output with output_times_periodicity
initial_output	false	True if initial output before the process simulation should be performed
final_output	false	True if final output after the process simulation should be performed
print_velocities	false	Include the velocities in the output file
print_coverages	false	Include the coverages in the output file
print_rates	false	Include the surface rates in the output file
print_materials	false	Include the materials numbers in the output file
max_time_step MaxTimeStep	max<double>	Maximum allowed time step

Process parameter	Default value	Description
iteration_cycles IterationCycles	0	The number of iterations (process calculation) before advancing to the next time step
start_iteration_cycles StartIterationCycles	0	The number of iterations at the process simulation start, might be necessary to calculate initial values for coverages
model_name	REQUIRED	The name of the used process model
output_times	-empty-	Vector of times where the output should be printed
finite_difference_scheme	ENGQUIST_OSHER_1ST_ORDER	Finite difference scheme to use for the simulation
dissipation_coefficient	-empty-	Dissipation coefficient to use for the Lax Friedrichs scheme
partition_data_structure	NEIGHBOR_LINKS_ARRAYS	Partition data structure to use for the simulation
partition_splitting_strategy	SURFACE_AREA_HEURISTIC	Partition splitting strategy to use for the simulation
partition_surface_area_heuristic_lambda	0.8	Heuristic lambda for the surface area partition
parameters	REQUIRED	Model parameters, which are passsed ot the model given by “ModelName”

```
}
```

Process Models

Available Process Model	Model Name
Boolean operation	“BooleanOperation”
Mask addition	“Mask”
Chemical mechanical planarization	“Planarization”
Constant rates	“ConstantRates”
Anisotropic wet etching	“WetEtching”
Simple deposition	“SimpleDeposition”
Non-linear deposition	“NonlinearDeposition”
Flux calculation	“CalculateFlux”
Focused ion beam	“FIB”
Palsma etching HBr/O_2	“HBr_O2PlasmaEtching”
Plasma etching SF_6/O_2	“SF6_O2PlasmaEtching”
SiO_2 plasma etching	“SiO2_PlasmaEtching”

Model Parameters

The model parameters are implemented as one section of the process parameters.

Format in the Parameters File

```
processes {
{
        [process_parameter1] = [process_parameter1_value];
        [process_parameter2] = [process_parameter2_value];
                ...
        [process_parameterN] = [process_parameterN_value];
        parameters = {
                [model_parameter] = [model_parameter_value];
        };
}
};
```

Boolean Operation

This model is used to perform boolean operations between two surfaces. From [1]:

If one considers the sets M_A, M_B, and M_C and the corresponding boundaries $S_A = \partial M_A$, $SB = \partial M_B$, and $S_C = \partial M_C$, which are represented by the LS functions Φ_A, Φ_B, and Φ_C, the LS counterparts of Boolean operations are listed in the following.

Union: $M_A = M_B \cup M_C \Leftrightarrow \Phi_A = \min(\Phi_B, \Phi_C)$

Intersection: $M_A = M_B \cap M_C \Leftrightarrow \Phi_A = \max(\Phi_B, \Phi_C)$

Complement: $M_A = R^D \setminus M_B \Leftrightarrow \Phi_A = -\Phi_B$

Relative complement: $M_A = M_B \setminus M_C \Leftrightarrow \Phi_A = \max(\Phi_B, -\Phi_C)$

Model name = “BooleanOperation”

Process parameter	**Process parameter values**	**Type**
geometry_file	“FileName”	string
level	number	int
invert	true/false	bool
surface_geometry	true/false	bool
remove_bottom	true/false	bool

Process parameter	**Default value**	**Description**
geometry_file	REQUIRED	Name of file with which to perform the boolean operation
level	0	1 – union -1 – intersection
invert	REQUIRED	Invert geometry before performing boolean operation
surface_geometry	false	True if the input geometry describes a surface (or surfaces). False if it describes a volume
remove_bottom	true	True if the location of the bottom of the simulation space can be ignored

Mask Addition

A mask is introduced with a pattern transfer using boolean operations.

Model name = “Mask”

Process parameter	Process parameter values	Type
mask_file	“FileName”	string
surface_geometry	true/false	bool
remove_bottom	true/false	bool

Process parameter	Default value	Description
mask_file	REQUIRED	Name of geometry file to be used as a mask layer
surface_geometry	false	True if the input geometry describes a surface (or surfaces). False if it describes a volume
remove_bottom	true	True if the location of the bottom of the simulation space can be ignored

Chemical Mechanical Planarization

Chemical mechanical planarization (CMP) assumes that everything above a certain height is removed.

Model name = “Planarization”

Process parameter	Process parameter values	Type
coordinate	number	double
fill_up	true/false	bool

Process parameter	Default value	Description
coordinate	REQUIRED	Value which determines the level at which the surface should be flattened
fill_up	false	False when no new material is introduced during planarization

Constant Rates

The constant rates model prescribes a given velocity to the level set geometries.

Model name = “ConstantRates”

Process parameter	Process parameter values	Type
direction	{num1, num2, num3}	double[3]
constant_rates	{num1, num2, … numN)	vector<double>
Isotropic_rates	{num1, num2, … numN)	vector<double>
directional_rates	{num1, num2, … numN)	vector<double>
vector_rates	{num1, num2, … numN)	vector<double>

Process parameter	Default value	Description
direction	-empty-	Direction in which the “directional_rates” should proceed
constant_rates	-empty-	Anisotropic velocity of the surface, defined for each material
Isotropic_rates	-empty-	Isotropic velocity of the surface, defined for each material
directional_rates	-empty-	Directional velocity of the surface, defined for each material; combined with the “direction” parameter
vector_rates	-empty-	Vector velocity of the surface, defined for each material

Anisotropic Wet Etching

The implemented model for the anisotropic wet etching of silicon is described in [2].

Model name = “WetEtching”

Process parameter	**Process parameter values**	**Type**
direction100	{num1, num2, num3}	vector<double>
direction010	{num1, num2, num3}	vector<double>
r100	number	double
r110	number	double
r111	number	double
r311	number	double
has_mask	true/false	bool

Process parameter	**Default value**	**Description**
direction100	REQUIRED	Normal vector for the crystallographic direction [100]
direction010	REQUIRED	Normal vector for the crystallographic direction [010]
r100	REQUIRED	Etch rate in the [100] crystallographic direction
r110	REQUIRED	Etch rate in the [110] crystallographic direction
r111	REQUIRED	Etch rate in the [111] crystallographic direction
r311	REQUIRED	Etch rate in the [311] crystallographic direction
has_mask	false	True if the etch should be performed around a mask

Simple Deposition

The simple deposition model is used when a single neutral depositing species is required for the process. Details of the implementation can be found in [1] and [3].

Model name = “SimpleDeposition”

Process parameter	Process parameter values	Type
direction	{num1, num2, num3}	double[3]
flux	number	double
sticking_probability	number	double
star_angle_distribution	number	double
reemitted_angle_distribution	number	double
stop_criterion	number	double
yield	number	double
statistical_accuracy	number	double

Process parameter	Default value	Description
direction	REQUIRED	Average direction of the incoming particles
flux	REQUIRED	Flux of the depositing species
sticking_probability	REQUIRED	Sticking probability of the depositing species on the surface
start_angle_distribution	1	Angular distribution of the incoming particles
reemitted_angle_distribution	1	Angular distribution of the particles which are reflected from the surface
stop_criterion	REQUIRED	Particle tracking continues until its probability reaches the stop_criterion
yield	REQUIRED	Relates the surface rates to the final velocity
statistical_accuracy	REQUIRED	Number of particles to use for each time step

Non-Linear Deposition

The implemented non-linear deposition model is described in more details in [4].

Model name = “NonlinearDeposition”

Process parameter	Process parameter values	Type
direction	{num1, num2, num3}	double[3]
deposition_rate	number	double
sticking_probability	number	double
reaction_order	number	double
stop_criterion	number	double
statistical_accuracy	number	double

Process parameter	Default value	Description
direction	REQUIRED	Average direction of the incoming particles
deposition_rate	REQUIRED	Average rate for the deposition
sticking_probability	REQUIRED	Sticking probability of the depositing species on the surface
reaction_order	REQUIRED	Order of the surface reaction; used to combine the effects of the depositing species coverage with the reflection coefficient and rate
stop_criterion	REQUIRED	Particle tracking continues until its probability reaches the stop_criterion
statistical_accuracy	REQUIRED	Number of particles to use for each time step

Flux Calculation

The flux calculation model is used to calculate the flux on a given geometry for various reflection models. The model will output two CSV files: rates.csv and rates_griddelta.csv.

rates.csv	–	Output file of the surface rates for all active grid points in grid spacing units defined by the grid_delta parameter
rates_griddelta.csv	–	Output file of the surface rates for all active grid points in global units (nm, µm, cm, m, ...)

Model name = “CalculateFlux”

Process parameter	Process parameter values	Type
direction	{num1, num2, num3}	double[3]
reflection_model	NONE DIFFUSIVE_SINGLE DIFFUSIVE_MULTIPLE SPECULAR DIFFUSIVE_COMBINED	-
reflection_diffusive_upperbound	number	int
flux	number	double
sticking_probability	number	double
star_angle_distribution	number	double
reemitted_angle_distribution	number	double
stop_criterion	number	double
yield	number	double
statistical_accuracy	number	double

Process parameter	Default value	Description
direction	REQUIRED	Average direction of the incoming particles
reflection_model	DIFFUSIVE_SINGLE	Applies the desired particle reflection model
reflection_diffusive_upperbound	4	Maximum number of new particles to generate when DIFFUSIVE_MULTIPLE reflection is selected
flux	REQUIRED	Flux of the depositing species

Process parameter	**Default value**	**Description**
sticking_probability	REQUIRED	Sticking probability of the depositing species on the surface
start_angle_distribution	1	Angular distribution of the incoming particles
reemitted_angle_distribution	1	Angular distribution of the particles which are reflected from the surface
stop_criterion	REQUIRED	Particle tracking continues until its probability reaches the stop_criterion
yield	REQUIRED	Relates the surface rates to the final velocity
statistical_accuracy	REQUIRED	Number of particles to use for each time step

Focused Ion Beam

The level set model for 2D FIB simulations is given in [5], while the 3D extension is described in further details in [1].

Model name = “FIB”

Process parameter	Process parameter values	Type
position	{num1, num2, num3}	double[3]
direction	{num1, num2, num3}	double[3]
FWHM	number	double
yield_0	number	double
yield_max	number	double
angle_max	number	double
current	number	double
num_reemited_particles	number	int
target_density	number	double
stop_criterion	number	double
sticking_probability	number	double
sputter_yield_file	“FileName”	string
num_simulated_particles	number	int

Process parameter	Default value	Description
position	REQUIRED	Location of the source of the incoming particles
direction	REQUIRED	Average direction of the incoming particles
FWHM	REQUIRED	Full width at half maximum for the incoming particle beam
yield_0	-empty-	Required Yamamura model parameter
yield_max	-empty-	Required Yamamura model parameter
angle_max	-empty-	Required Yamamura model parameter
current	REQUIRED	Flux of the incoming particles

Process parameter	Default value	Description
num_reemited_particles	1	Number of particles to be re-emitted after surface reflection
target_density	REQUIRED	Density of the targeted material
stop_criterion	0	Particle tracking continues until its probability reaches the stop_criterion
sticking_probability	0	Sticking probability of the depositing species on the surface
sputter_yield_file	-empty-	If a file should be used instead of the Yamamura model, then a file should be given
num_simulated_particles	REQUIRED	Number of particles to use for each time step

Plasma Etching – HBr/O_2

The model implemented within the level set framework is explained in more details in [6], while the parameters used in the model are given in [7].

Model name = “HBr_O2PlasmaEtching”

Process parameter	Process parameter values	Type
direction	{num1, num2, num3}	double[3]
flux_ion	number (/cm^3)	double
flux_oxygen	number (/cm^3)	double
flux_bromine	number (/cm^3)	double
a_oxygen	number	double
statistical_accuracy	number	int
min_ion_energy	number	double
delta_ion_energy	number	double

Process parameter	Default value	Description
direction	REQUIRED	Average direction of the incoming particles
flux_ion	REQUIRED	Incoming ion flux
flux_oxygen	REQUIRED	Oxygen flux at the feature surface
flux_bromine	REQUIRED	Bromine flux at the feature surface
a_oxygen	REQUIRED	Model parameter related to the oxygen yield
statistical_accuracy	REQUIRED	Number of particles to use for each time step
min_ion_energy	REQUIRED	Minimum energy of the incoming ions
delta_ion_energy	REQUIRED	Delta energy of the incoming ions, used to generate an ion particle energy distribution

Plasma Etching – SF_6/O_2

The model implemented within the level set framework is explained in more details in [1], while the parameters used in the model are given in [8].

Model name = “SF6_O2PlasmaEtching”

Process parameter	Process parameter values	Type
direction	{num1, num2, num3}	double[3]
flux_ion	number (/cm^3)	double
flux_oxygen	number (/cm^3)	double
flux_fluorine	number (/cm^3)	double
a_oxygen	number	double
statistical_accuracy	number	int
min_ion_energy	number	double
delta_ion_energy	number	double

Process parameter	Default value	Description
direction	REQUIRED	Average direction of the incoming particles
flux_ion	REQUIRED	Incoming ion flux
flux_oxygen	REQUIRED	Oxygen flux at the feature surface
flux_fluorine	REQUIRED	Fluorine flux at the feature surface
a_oxygen	REQUIRED	Model parameter related to the oxygen yield
statistical_accuracy	REQUIRED	Number of particles to use for each time step
min_ion_energy	REQUIRED	Minimum energy of the incoming ions
delta_ion_energy	REQUIRED	Delta energy of the incoming ions, used to generate an ion particle energy distribution

SiO_2 Plasma Etching

The model implemented within the level set framework is explained in more details in [6], while the parameters used in the model are given in [9].

Model name = "SiO2_PlasmaEtching"

Process parameter	Process parameter values	Type
direction	{num1, num2, num3}	double[3]
flux_ion	number (/cm^3)	double
flux_polymer	number (/cm^3)	double
flux_etchant	number (/cm^3)	double
statistical_accuracy	number	int
min_ion_energy	number	double
delta_ion_energy	number	double
temperature	number	double

Process parameter	Default value	Description
direction	REQUIRED	Average direction of the incoming particles
flux_ion	REQUIRED	Incoming ion flux
flux_polymer	REQUIRED	Flux of the polymer which is simultaneously deposited along the profile during the etching step
flux_etchant	REQUIRED	Flux of the etchant used to chemically etch the oxide layer
statistical_accuracy	REQUIRED	Number of particles to use for each time step
min_ion_energy	REQUIRED	Minimum energy of the incoming ions
delta_ion_energy	REQUIRED	Delta energy of the incoming ions, used to generate an ion particle energy distribution
temperature	REQUIRED	Temperature of the simulation; The effective etchant flux has an Arrhenius relationship to the incoming flux

Application Programming Interface

In the */src/viennats.cpp* file, the *main* directory is found. The only argument which needs to be passed during execution is the parameters file.

Please look at the */docs/ViennaTS_documentation.html* for a more detailed documentation of the ViennaTS code.

References

[1] Ertl, Otmar; Numerical Methods for Topography Simulation, Dissertation, Institute for Microelectronics, TU Wien (2010).

[2] B. Radjenovic and M. Radmilovic-Radjenovic. 3D simulations of the profile evolution during anisotropic wet etching of silicon. *Thin Solid Films*, **517**(14): 4233–4237, 2009.

[3] D. Adalsteinsson and J. A. Sethian. A fast level set method for propagating interfaces. *Journal of Computational Physics*, **118**(2): 269–277, 1995.

[4] T. S. Cale and G. B. Raupp. A unified line-of-sight model of deposition in rectangular trenches. *Journal of Vacuum Science and Technology B*, **8**(6): 1242–1248, 1990.

[5] H.-B. Kim, G. Hobler, A. Steiger, A. Lugstein, and E. Bertagnolli. Level set approach for the simulation of focused ion beam processing on the micro/nano scale. *Nanotechnology*, **18**(26): 265307, 2007.

[6] Filipovic, Lado; Topography Simulation of Novel Processing Techniques, Dissertation, Institute for Microelectronics, TU Wien (2012).

[7] R. J. Belen, S. Gomez, M. Kiehlbauch, and E. S. Aydil. Feature scale model of Si etching in $SF_6/O_2/HBr$ plasma and comparison with experiments. *Journal of Vacuum Science & Technology A: Vacuum, Surfaces, and Films*, **24**(2): 350–361, 2006.

[8] R. J. Belen, S. Gomez, D. Cooperberg, M. Kiehlbauch, and E. S. Aydil. Feature-scale model of Si etching in SF6 /O2 plasma and comparison with experiments. *Journal of Vacuum Science and Technology A*, **23**(5): 1430–1439, 2005.

[9] A. La Magna and G. Garozzo. Factors affecting profile evolution in plasma etching of SiO2 modeling and experimental verification. *Journal of The Electrochemical Society*, **150**(10): F178–F185, 2003.