SoftwareX 32 (2025) 102453

Contents lists available at ScienceDirect

SoftwareX

journal homepage: www.elsevier.com/locate/softx

ViennaPS: A flexible framework for semiconductor process simulation

T. Reiter*, L. Filipovic

CDL for Multi-Scale Process Modeling of Semiconductor Devices and Sensors at the Institute for Microelectronics, TU Wien, 1040 Vienna, Austria

ARTICLE INFO

Keywords:
Process simulation
Topography evolution
Plasma etching simulation

ABSTRACT

ViennaPS is an open-source software framework for simulating fabrication processes in semiconductor manufacturing, with a focus on topography evolution during etching and deposition. It uses a high-performance level-set method with a hierarchical run-length encoding data structure for efficient and fast geometry representation and evolution. ViennaPS supports both analytical and physics-based process models, including Monte Carlo ray tracing for flux calculation at the feature scale. Designed for flexibility and extensibility, it enables users to prototype new models or apply pre-configured ones. ViennaPS provides a customizable platform for researchers and engineers developing advanced process simulations in both academic and industrial settings.

Metadata

Nr.	Code metadata description	Metadata
C1	Current code version	4.1.2
C2	Permanent link to code/repository used for this code version	https://github.com/ViennaTools/ViennaPS/tree/v4.1.2
C3	Permanent link to Reproducible Capsule	NA
C4	Legal Code License	MIT
C5	Code versioning system used	Git
C6	Software code languages, tools, and services used	C++, Python, CUDA
C7	Compilation requirements, operating environments & dependencies	Linux, Windows, macOS; VTK, Embree
C8	Link to developer documentation/manual	https://viennatools.github.io/ViennaPS/
C9	Support email for questions	vienna-tools@iue.tuwien.ac.at

1. Motivation and significance

Semiconductor fabrication increasingly requires predictive, feature-scale modeling of topography evolution during etching and deposition. Commercial simulators provide robust capabilities, but their closed-source nature limits transparency and the exchange or development of new physical models. Public academic codes are often unavailable or target narrow subproblems without an end-to-end workflow. ViennaPS addresses this gap with an open and extensible framework for both geometric emulation and physics-based simulation, designed for reproducibility, method development, and integration into advanced research workflows.

As device architectures evolve toward higher aspect ratios, tighter design rules, and complex 3D geometries, accurately predicting the effects of manufacturing steps such as etching and deposition becomes essential for ensuring process reliability, design fidelity, and yield optimization [1]. Physical process simulation, often referred to as process TCAD, plays a critical role in enabling virtual prototyping and reducing experimental costs [2]. However, existing topography simulators often suffer from limited flexibility, lack of extensibility, or closed-source licensing, hindering both academic exploration and industrial adaptation.

To address these limitations, ViennaPS provides a flexible, open-source framework for simulating surface and volume evolution during key semiconductor fabrication steps. By combining a sparse-field level-set method with a highly efficient hierarchical run-length encoding (HRLE) data structure, ViennaPS enables accurate and performant simulation of evolving 3D geometries. The software supports both analytical emulation and physics-driven models based on surface flux calculations, allowing users to explore process behavior ranging from idealized test cases to realistic plasma environments.

* Corresponding author.
Email address: reiter@iue.tuwien.ac.at (T. Reiter).

https://doi.org/10.1016/j.softx.2025.102453
Received 25 July 2025; Received in revised form 4 November 2025; Accepted 18 November 2025
Available online 28 November 2025

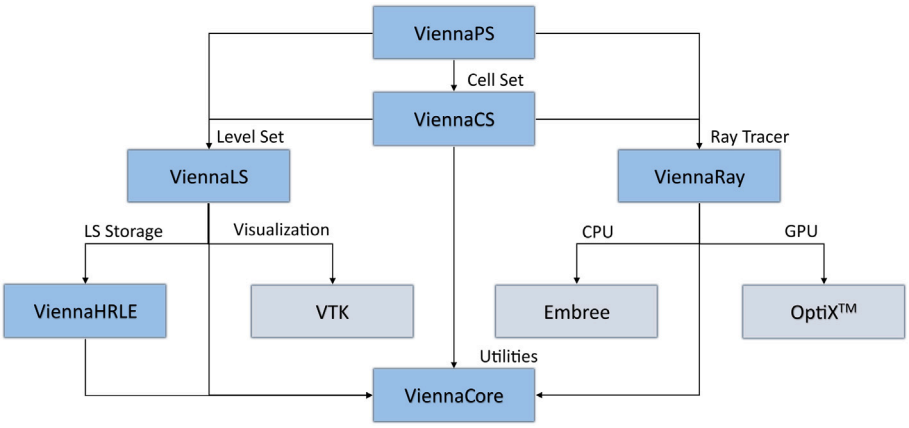

Fig. 1. ViennaPS framework and its core modules. ViennaPS builds on ViennaLS and ViennaCS to represent level-set and cell-based geometries, respectively. For ray tracing, ViennaRay interfaces with either Embree (CPU) or OptiX (GPU) libraries. ViennaLS internally relies on ViennaHRLE for efficient level-set data storage and uses VTK for visualization. Shared utilities and data structures are provided by ViennaCore.

ViennaPS is already applied in research to study feature-scale etching phenomena such as the loading effect in SiGe/Si stacks [3], mask geometry variation effects in plasma etching profiles [4], and the impact of plasma induced damage during the fabrication of 3D NAND flash memory [5].

Compared to commercial process simulation tools such as Synopsys Sentaurus Topography [6], Lam Research SEMulator3D [8] or Silvaco's Victory Process [7], which are closed-source and tailored to production environments, ViennaPS offers open access, full transparency, and the ability to modify or extend physical models at the source level. It combines a performance-focused C++ core with a high-level Python interface, allowing researchers to prototype new models, explore novel process conditions, and integrate simulations into automated design or calibration workflows. The ViennaPS simulation pipeline is fully scriptable, supporting tasks such as parameter sweeps, geometry variations, and the generation of synthetic process data for neural network training [9]. These combined features position ViennaPS as a flexible research-oriented framework that complements existing tools and enables reproducible modeling workflows.

To our knowledge, no publicly available open-source framework currently provides feature-scale topography simulation with modifiable physical models and a fully scriptable CPU/GPU workflow. ViennaPS directly addresses this need by offering a transparent, extensible, and research-oriented alternative to proprietary tools.

2. Software description

The ViennaPS framework is designed for *feature-scale process simulation*, where the focus lies on resolving the detailed geometry and surface interactions at the level of individual device features. This scope contrasts with *reactor-scale* simulations, which model global plasma behavior and process conditions within the entire chamber. To model these feature-scale effects, ViennaPS offers two distinct approaches for continuum modeling, where materials and surfaces are treated as uniform media. These are physical simulation and geometric emulation. Physical *simulation* in ViennaPS involves computing fluxes and solving surface evolution equations based on semi-empirical models. In contrast, *emulation* relies on simplified analytic models that replicate the geometric outcomes of processes without explicitly modeling the underlying physics. These models can incorporate key physical effects such as material-selective masking and geometric shadowing through visibility checks. ViennaPS supports both approaches, offering flexibility in balancing accuracy, speed, and complexity depending on the application.

2.1. Software architecture

ViennaPS is implemented in modern C++ to ensure high performance and scalability. It provides Python bindings for improved usability, scriptability, and integration into existing workflows. The framework follows a modular architecture, consisting of both internal and external components, as illustrated in Fig. 1.

At its core, ViennaPS is composed of the following major submodules:

1. **ViennaLS**
   Responsible for solving the level set equation during topography simulation using a sparse-field level-set method, ViennaLS employs a hierarchical run-length encoded (HRLE) data structure for efficient memory use. It also supports input and output of geometric data using the Visualization Toolkit (VTK) [10], enabling compatibility with tools like ParaView for visualization.
2. **ViennaCS**
   This module extends ViennaPS beyond surface evolution by providing a volumetric cell-set representation. It allows modeling of volume-based processes such as implantation or diffusion, as well as tracking changes in regions above and below material interfaces.
3. **ViennaRay**
   ViennaRay performs feature-scale flux calculations using a top-down Monte Carlo ray tracing approach. Given the high computational demands of this task, the module supports both CPU-based ray tracing via Embree [11] and GPU acceleration through NVIDIA OptiX [12], offering users flexibility in balancing speed and accuracy.

In addition to the core modules, ViennaPS includes **ViennaCore**, a utility library that provides essential shared functionalities such as logging infrastructure, vector operations, and helper methods used across the framework.

ViennaPS is designed to be cross-platform and supports Linux, macOS, and Windows. The build system is based on CMake and automatically checks for required dependencies, including OpenMP, VTK, and Embree. GPU support is available via OptiX, provided that users have a compatible NVIDIA GPU, CUDA toolkit and up-to-date drivers.

To improve accessibility, pre-built Python wheels are provided for major platforms via PyPI,[1] while advanced users can build the bindings

[1] https://pypi.org/project/viennaps

from source, allowing for deeper customization. As an open-source project, ViennaPS is actively maintained on GitHub, with updated documentation and examples provided online. Users are encouraged to report issues, contribute new features, and participate in development through the issue tracker. For precise control over performance and memory, the framework supports both 2D and 3D simulations and allows the underlying numeric precision to be configured (either `float` or `double`).

2.1.1. Core simulation components

A ViennaPS simulation is structured around three main components: the `Domain`, `ProcessModel`, and `Process`.

- `Domain`
  The `Domain` class represents the simulation space and stores the geometric data. It manages multiple level sets, each representing the interface between different materials. The enclosed volume corresponds to the associated material region. To model layered structures, level sets are inserted hierarchically, ensuring that each new layer fully encapsulates (wraps) the previous ones. This ensures that the topmost level set represents the current device surface.
- `ProcessModel`
  The `ProcessModel` class defines the physics and configuration of a specific fabrication step. It consists of several modular subcomponents:
  - **Surface Model**: Defines the surface reactions and evolution rates, optionally tracking molecular coverage.
  - **Particles**: Describes the particle species used in the simulation and how their fluxes are computed.
  - **Advection Callback**: Offers pre- and post-step hooks for custom logic around each surface movement.
  - **Velocity Field**: Extends interface velocities to the full level-set domain. For flux-driven models, the surface rate is extended from the interface to nearby grid points. For analytic models, rates can be applied directly to the entire field.

  ViennaPS includes a set of pre-configured process models, which users can adapt through parameter tuning. For advanced use cases, the framework exposes all components to allow users to define custom behaviors, materials, and reaction mechanisms.
- `Process`
  A `Process` executes a given `ProcessModel` on a specified `Domain` for a defined duration. Additional simulation parameters affecting statistical accuracy and numerical stability can be configured to meet the user's requirements.

2.2. Software functionalities

ViennaPS offers a comprehensive set of functionalities tailored for topography simulation in semiconductor manufacturing. These features enable both physical simulation and emulation of fabrication processes with a high degree of flexibility:

- **Level-Set Surface Evolution:** ViennaPS uses a sparse-field level-set method [13] built on a hierarchical run-length encoded (HRLE) data structure [14] to track complex surface geometries over time. This approach supports accurate modeling of dynamic surface changes during etching and deposition.
- **Top-Down Monte Carlo Ray Tracing:** The framework includes a top-down Monte Carlo ray tracing engine to calculate particle fluxes on 3D feature surfaces, accounting for angular and spatial distributions. This method is essential for feature-scale simulation of realistic plasma processes.
- **Cell-Set Based Volume Description:** ViennaPS includes a volumetric representation based on cell sets, which enables the modeling of material regions and phenomena beyond the immediate surface. This is particularly useful for simulating implantation, diffusion, and other volume-based processes.
- **Physical Process Models:** A variety of physical and semi-empirical models are implemented to simulate common semiconductor processes, including:
  - Chemical Vapor Deposition (CVD) based on single- or multi-particle processes
  - Plasma etching using SF_6/O_2 and fluorocarbon chemistries
  - Wet chemical etching and epitaxial growth
  - Atomic layer deposition and etching (ALD/ALE)
  - Deep reactive ion etching (DRIE) via the Bosch process

  Each process model supports user customization through material properties, surface reactions, and particle flux configurations.
- **GDSII Mask Import:** ViennaPS supports layout-based mask import using the GDSII format. This feature allows users to define complex pattern geometries from design data for direct integration into 2D or 3D simulations.
- **VTK-Based Geometry Export:** The framework exports the resulting geometry as a surface or volume mesh in the VTK file format. This enables visualization and post-processing with external tools such as ParaView or other tools that support VTK-based rendering.

3. Illustrative examples

3.1. Emulation examples

The simplest surface evolution models in ViennaPS are based on advection with constant or spatially varying rates. These rates are typically applied iteratively using level set integration schemes, but ViennaPS also supports geometric advection as a faster alternative [15]. Geometric advection updates the surface directly based on the local interface velocity and normal vector, bypassing the need to solve the full level set equation and offering substantial performance gains for purely emulative workflows.

Isotropic processes apply the same etch or deposition rate uniformly in all directions and at all surface points, as illustrated in Fig. 2(a). These models are suitable for processes where directional effects are negligible, such as thermal oxidation or wet etching under homogeneous chemical conditions. Material-specific rates can be defined to account for selectivity, enabling representations of etch masks or differing deposition behavior across materials.

Directional processes, in contrast, assign rates along a fixed direction, typically the wafer normal, and are used to emulate highly anisotropic effects like ion-driven etching or physical sputtering. In their basic form, these models apply the directional rate uniformly to all surface points, regardless of local geometry (see Fig. 2(b)). To more realistically capture directional transport phenomena, ViennaPS optionally performs a visibility check that disables the rate at occluded or shadowed locations. This mimics the limited line-of-sight of ballistic particles, particularly in narrow or deep features, as shown in Fig. 2(c).

Furthermore, these elemental models can be combined within a single process step. A directional etch rate, for instance, can be superimposed with an isotropic component. This allows for the emulation of more complex phenomena, such as an ion-driven etch that simultaneously has a chemical, isotropic etching effect.

Together, these emulation models form a computationally efficient toolkit for approximating a wide range of fabrication steps without requiring full particle flux calculations. They can be combined into process flows that resemble real manufacturing sequences. One such example is shown in Fig. 3, demonstrating a simplified FinFET process including fin double patterning, selective epitaxy of source and drain regions, and gate contact formation.

3.2. Simulation examples

To illustrate the capabilities and accuracy of ViennaPS, two representative process simulations are presented. These examples demonstrate the framework's ability to reproduce realistic topography evolution for

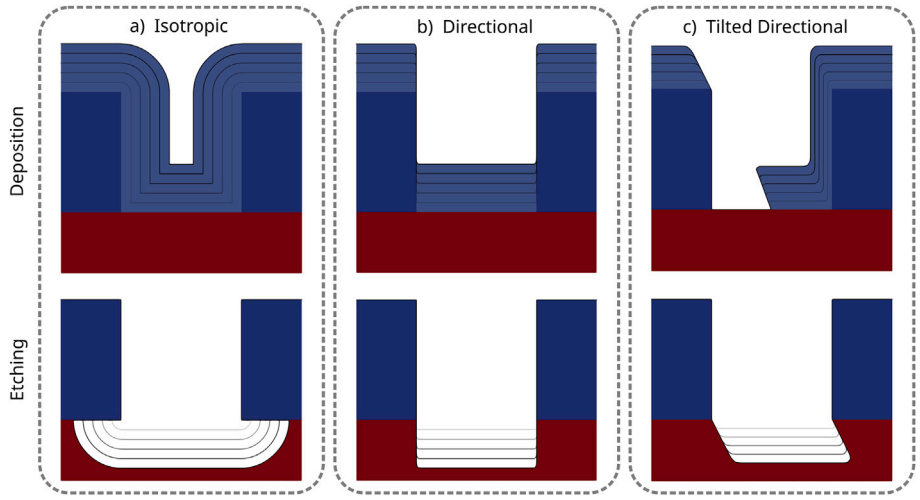

Fig. 2. Examples of process emulation. **a)** Isotropic processes uniformly grow or remove the structure equally in all directions while preserving masked regions; the blue region represents the mask layer. **b)** Directional deposition from a vertical source leads to anisotropic growth along the vertical axis. Directional vertical etching removes unmasked material from the top, preserving the masked sidewalls (mask shown in blue). **c)** Directional deposition from a tilted source includes visibility checking, leading to asymmetric growth. Directional etching with a tilted incidence angle removes material along the incoming direction, with the blue mask protecting the underlying structure.

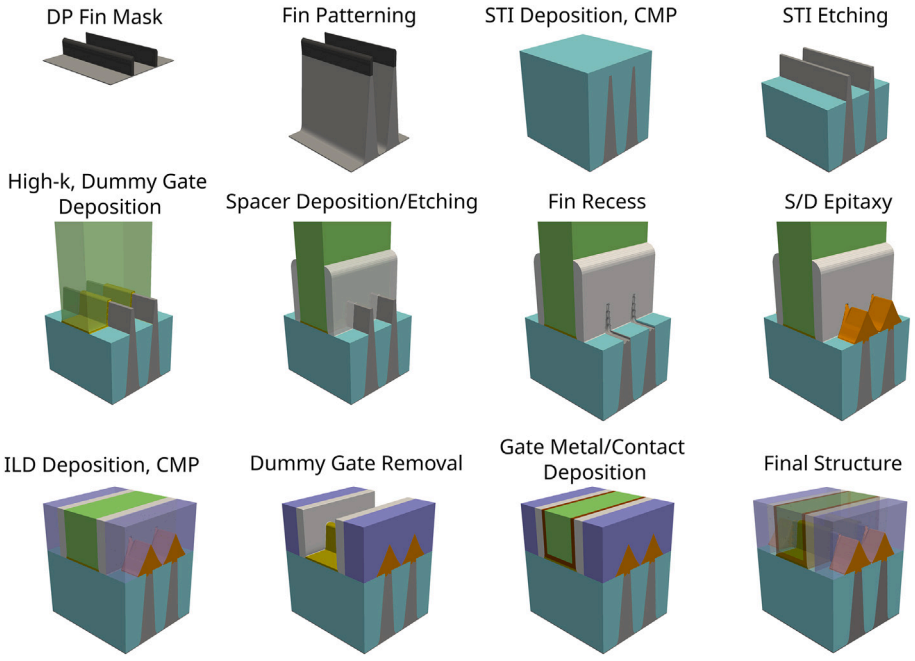

Fig. 3. Simplified FinFET process emulation using ViennaPS. The flow demonstrates key topography steps modeled using analytic surface velocities and material-based masking. From top left to bottom right: (1) spacer-defined fin double patterning, (2) fin transfer etch into the substrate, (3) shallow trench isolation (STI) deposition and planarization, (4) fin reveal, (5) dummy gate stack formation, (6) sacrificial oxide and nitride spacer deposition and etch, (7) source and drain recess etch, (8) selective epitaxy of source and drain regions, (9) interlayer dielectric (ILD) deposition and planarization, (10) dummy gate removal, (11) metal gate stack deposition and planarization, (12) final structure including conformal and anisotropic etching steps. All steps are emulated without flux-based modeling, using directional or isotropic advection rates, geometric advection, and material-selective masks. The corresponding simulation script is available at: https://github.com/ViennaTools/ViennaPS/blob/v4.0.0/examples/emulation/FinFET.py.

deposition and etching processes and to achieve quantitative agreement with established commercial simulators.

This first example compares a particle-flux based deposition process on trenches with varying aspect ratios simulated in Silvaco's Victory Process and in ViennaPS using identical flux and surface reaction models. The close match between both results confirms the physical consistency and numerical reliability of ViennaPS. In addition, the performance benchmark in Fig. 4(c) demonstrates its computational efficiency, showing predictable scaling behavior with decreasing grid spacing while maintaining high accuracy in feature-scale topography simulation.

As a further demonstration of ViennaPS's physical simulation capabilities, we present the SF_6/O_2 plasma etching model for silicon. This model captures key feature-scale phenomena such as ion-enhanced etching, dynamic surface passivation, and competitive adsorption, and serves as a representative example of the physical process models

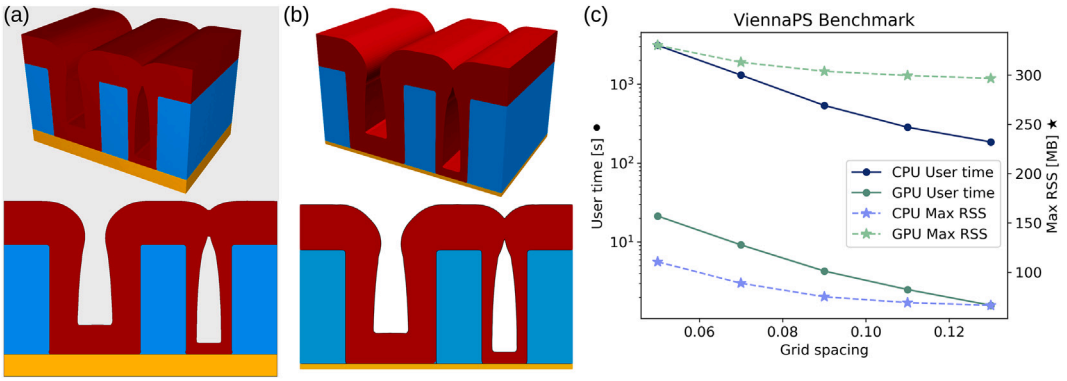

Fig. 4. (a) Particle-flux based deposition on trenches with different aspect ratios simulated in Silvaco's Victory Process. (b) Equivalent simulation performed in ViennaPS using the same flux-based surface model. (c) Performance benchmark of the ViennaPS simulation for varying grid spacings. A smaller grid spacing corresponds to a finer resolution with more grid points used to represent the geometry. The *Max RSS* denotes the maximum resident set size, representing the peak physical memory usage of the simulation process.

implemented in ViennaPS. In addition to SF_6/O_2 etching, ViennaPS supports a variety of other surface- and volume-based models, enabling a broad range of semiconductor fabrication processes to be studied.

Modern plasma etching involves complex interactions between surface chemistry and incoming species. Surface coverage-based models are particularly well suited to capture these effects, as they account for the local chemical environment, adsorption competition, and time-dependent passivation dynamics. To capture these complex surface interactions, the plasma etching models in ViennaPS use a Langmuir–Hinshelwood-type site balance. This approach tracks the fraction of the surface covered by different chemical species, which dynamically changes based on incoming particles and surface reactions. In the case of SF_6/O_2 etching, reactive species such as fluorine (F) and oxygen (O) adsorb onto a limited number of available surface sites. Their coverages evolve in response to incident fluxes, sticking probabilities, and surface reactions, including adsorption, desorption, sputtering, and etching.

The SF_6/O_2 etching model for silicon, based on the work of Belen et al. [16], incorporates chemical etching by fluorine, physical sputtering by ions, and ion-enhanced etching. Surface fluxes of fluorine, oxygen, and ions are computed using top-down Monte Carlo ray tracing, which enables spatially resolved coverage calculations. The surface coverages θ_F and θ_O evolve according to:

$$\sigma_{Si}\frac{d\theta_F}{dt} = \gamma_F\Gamma_F(1-\theta_F-\theta_O) - k\sigma_{Si}\theta_F - 2Y_{ie}\Gamma_i\theta_F, \tag{1}$$

$$\sigma_{Si}\frac{d\theta_O}{dt} = \gamma_O\Gamma_O(1-\theta_F-\theta_O) - \beta\sigma_{Si}\theta_O - Y_O\Gamma_i\theta_O. \tag{2}$$

Assuming a pseudo-steady state condition, the total etch rate can be expressed as:

$$\mathrm{ER} = \frac{1}{\rho_{Si}}\left(\underbrace{\frac{k\sigma_{Si}\theta_F}{4}}_{\text{Chemical}} + \underbrace{Ysp\Gamma_i}_{\text{Sputtering}} + \underbrace{Yie\Gamma_i\theta_F}_{\text{Ion-enhanced}}\right). \tag{3}$$

Fluorine coverage promotes chemical etching and enhances ion-driven reactions by lowering the energy barrier for sputtering, while oxygen competes for surface sites, inhibiting fluorine adsorption and reducing the etch rate. Neutral species are modeled with diffuse reflection, whereas ions follow a coned specular distribution to reflect their collimated angular spread.

Fig. 5 shows the results of a hole etching simulation under varying fluorine and oxygen flux conditions. Higher fluorine flux accelerates etching, whereas elevated oxygen flux promotes sidewall passivation, leading to narrower and more vertical profiles. The resulting interplay between etching and passivation enables tuning of the profile shape from isotropic to highly anisotropic depending on process parameters.

The model parameters have been calibrated to experimental data reported by Belen et al. [16], ensuring quantitative agreement with observed etch rates and profile shapes under various plasma conditions.

Furthermore, the model can be extended to include polymer deposition mechanisms present in fluorocarbon-based chemistries. By incorporating additional polymer material, it becomes possible to simulate time-multiplexed etching schemes such as the Bosch process, including alternating deposition and etch steps that enable deep, anisotropic profiles with sidewall protection.

4. Impact

ViennaPS advances the field of semiconductor process simulation by offering a high-performance, open-source platform tailored to the needs of topography modeling. Its architecture supports a wide range of applications, from analytic emulation to physically motivated feature-scale simulations, making it suitable for early stage prototyping as well as detailed process analysis. Researchers can build on ViennaPS to investigate complex interactions in etching and deposition, including aspect ratio dependent etching and deposition (ARDE/ARDD), angular and energy dependent etch yields, multi-material interfaces with varying sticking coefficients and reaction rates, as well as advanced models for atomic layer processing.

A key differentiator of ViennaPS is its open-source nature, which enables transparent, reproducible simulation workflows. Complete simulation scripts and configuration files are made available alongside published examples, allowing other researchers to replicate results, validate model behavior, and adapt workflows to their own needs. This level of openness is not possible with commercial tools, making ViennaPS especially valuable for academic and collaborative research environments. This transparency also enables reproducible workflows, collaborative model development, and community-driven extensions, which are difficult to achieve with proprietary simulation environments.

The framework's capability in modeling advanced processes across a wide range of applications is demonstrated by its contribution to several peer-reviewed publications. One example is the investigation of loading effects during the selective isotropic etching of SiGe in stacked SiGe/Si structures, relevant for gate-all-around transistor fabrication [3]. ViennaPS was used to calibrate a physical etch model that accurately reproduced lateral etch profiles under varying structure densities and process conditions. The study identified restricted particle diffusion as the main contributor to reduced etch uniformity in dense regions, highlighting the strength of ViennaPS in predicting complex feature-scale behavior.

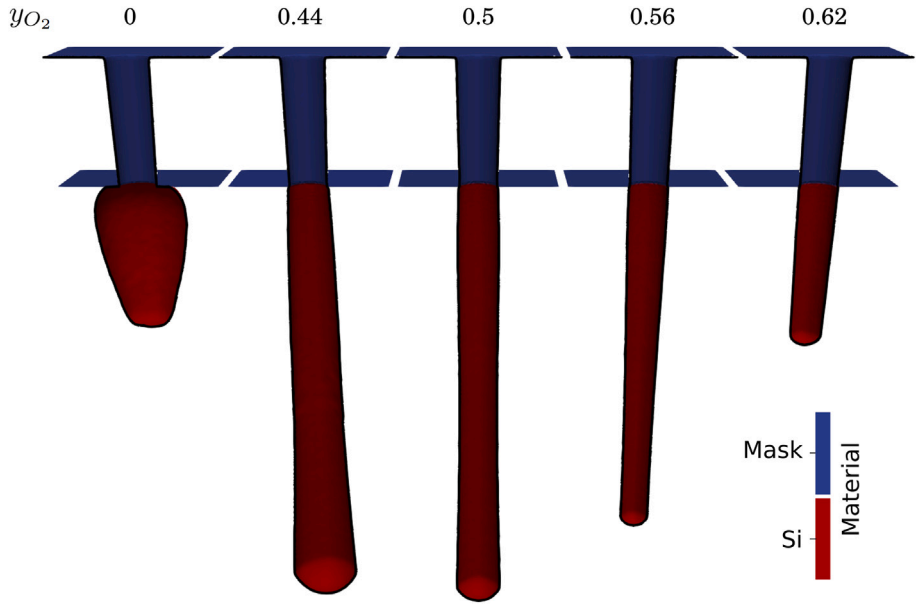

Fig. 5. SF_6/O_2 hole etching with varying flux ratios. The figure shows a clipped view of the etched hole geometry, with the blue region indicating the mask layer and the red region representing the silicon substrate. The different profiles demonstrate how the balance between SF_6 and O_2 fluxes y_{O_2} influences the etching outcome. Higher SF_6 flux enhances chemical etching and increases etch rates, while higher O_2 flux promotes sidewall passivation, leading to more anisotropic profiles. The corresponding simulation script is available at: https://github.com/ViennaTools/ViennaPS/blob/v4.0.0/examples/holeEtching/testFluxes.py.

In another study [4], ViennaPS was used to develop and validate a physical SF_6/O_2 plasma etching model based on Langmuir adsorption for the fabrication of vertical three-dimensional features. The simulations explored how variations in mask geometry, such as taper angle, thickness, and etch rate, influence profile depth and bowing. These results provided predictive guidance for mask optimization in advanced plasma processes.

ViennaPS also enabled the simulation of plasma-induced damage during vertical channel hole etching in 3D NAND flash memory [5]. A model was implemented to track ion-induced damage as a surface property, which was shown to degrade subsequent selective epitaxial growth and cause void formation. By integrating a binary collision-based damage model [17], the framework could perform transient etch simulations and match experimental observations, offering valuable insights into damage mitigation strategies.

In addition, ViennaPS was applied as the foundational tool for research into physics-informed deep learning for plasma etch optimization [9]. The framework was used to generate a series of plasma etch datasets for training and testing machine learning models. This process involved running numerous simulations to extract key parameters that describe the etch profile's evolution, specifically the surface velocity and the level-set function. To meet the specific needs of the research, the author supplemented ViennaPS's capabilities by introducing direct code modifications, adding new parameters for mask density and influx angle.

These examples illustrate just a subset of the publications that have used ViennaPS as their core simulation framework. By enabling accurate modeling of subtle process variations and their impact on nanoscale features, ViennaPS provides insights that are often inaccessible through experiments or less flexible simulation environments.

Its efficient C++ core, combined with accessible Python bindings, has attracted a growing user base from both academic and industrial research communities. While the user base is still developing, usage has recently gained momentum, with ongoing adoption in new projects and increasing visibility through citations and collaborations. The modular design of the software enables customization and extension, making it possible for the community to develop and share their own models, algorithms, and workflows. This flexibility allows ViennaPS to be adapted to a wide range of research needs, including new physical models, surface interaction mechanisms, and coupled simulation environments.

ViennaPS thus not only addresses current modeling challenges, but also provides a platform for future research directions. These include multi-scale modeling approaches, such as coupling with equipment- or reactor-scale plasma simulations to provide more realistic input at the feature scale. Additionally, integration with atomistic simulations can offer deeper physical insights into material properties and surface interaction parameters.

The framework's scripting interface and modular design support automated parameter tuning and process optimization strategies, making it suitable for use in data-driven process development or hybrid simulation pipelines that combine physics-based models with surrogate or machine learning components.

5. Conclusion

In this work, we introduce ViennaPS, an open-source framework for topography simulation in semiconductor manufacturing. Combining an efficient sparse-field level-set method with a high-performance HRLE data structure, ViennaPS accurately models complex surface evolution for a wide range of fabrication processes, including directional and isotropic deposition, as well as advanced plasma etching techniques.

The integration of Monte Carlo ray tracing enables realistic particle flux computation at the feature scale. ViennaPS supports both emulation and physics-based simulation through a modular design, allowing custom model development and offering pre-configured options for common use cases. Python bindings further enhance accessibility and integration with existing workflows.

Overall, ViennaPS offers a flexible and extensible alternative to commercial tools, combining physical accuracy with performance. Ongoing developments aim to expand multi-scale capabilities and integration with reactor-scale simulations.

CRediT authorship contribution statement

T. Reiter: Writing – original draft, Software, Conceptualization. **L. Filipovic:** Writing – review & editing, Supervision, Software, Funding acquisition, Conceptualization.

Declaration of generative AI and AI-assisted technologies in the writing process

During the preparation of this work the authors used ChatGPT 4o in order to improve readability and language. After using this tool, the authors reviewed and edited the content as needed and take full responsibility for the content of the publication.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

Financial support by the Federal Ministry of Labour and Economy, the National Foundation for Research, Technology and Development, and the Christian Doppler Research Association is gratefully acknowledged.

The authors acknowledge TU Wien Bibliothek for financial support through its Open Access Funding Programme.

References

[1] Hargrove M, Wen S, Yim D, Ruegger KE, Nanja P, Sarkar S, et al. Review of virtual wafer process modeling and metrology for advanced technology development. J Micro/Nanopatterning Mater Metrol 2023;22(3):031209–17. https://doi.org/10.1117/1.JMM.22.3.031209

[2] Aboud S. What is TCAD and why it is essential for the semiconductor industry. In: Semiconductor Engineering, 16 November 2023. Available from: https://semiengineering.com/what-is-tcad-and-why-it-is-essential-for-the-semiconductor-industry/ [cited 14 July 2025].

[3] Shao H, Reiter T, Chen R, Li J, Hu Z, Wei Y, et al. Loading effect during sige/si stack selective isotropic etching for gate-all-around transistors. ACS Appl Electron Mater 2024;6(11):8124–9. https://doi.org/10.1021/acsaelm.4c01462

[4] Bobinac J, Reiter T, Piso J, Klemenschits X, Baumgartner O, Stanojevic Z, et al. Effect of mask geometry variation on plasma etching profiles. Micromachines 2023;14(3):665. https://doi.org/10.3390/mi14030665

[5] Reiter T, Klemenschits X, Filipovic L. Impact of plasma induced damage on the fabrication of 3d NAND flash memory. Solid State Electron 2022;192:108261–9. https://doi.org/10.1016/j.sse.2022.108261

[6] Synopsys, Inc. Sentaurus topography; 2025. Available from:. https://www.synopsys.com/manufacturing/tcad/process-simulation/sentaurus-topography.html [Accessed 30 October 2025].

[7] Silvaco, Inc. Victory process; 2025. Available from:. https://silvaco.com/tcad/victory-process-3d/ [Accessed 30 October 2025].

[8] Coventor, Inc. A lam research company. SEMulator3D; 2025. Available from: https://www.lamresearch.com/products/semulator3d/ [Accessed 30 October 2025].

[9] Dighamber M. Physics-informed deep learning for plasma etch optimization Master of Engineering Thesis, Massachusetts Institute of Technology; 2024. Available from: https://dspace.mit.edu/handle/1721.1/157256.

[10] Schroeder W, Martin K, Lorensen B. The visualization toolkit (VTK). Version 9.0.0. software. 2024.

[11] Wald H, Woop S, Benthin C, Johnson G, Ernst M. Embree: a kernel framework for efficient CPU ray tracing. ACM Trans Graph 2014;33(4):143–8. https://doi.org/10.1145/2601097.2601199

[12] Parker SG, Bigler J, Dietrich A, Friedrich H, Hoberock J, Luebke D, et al. Optix: a general purpose ray tracing engine. ACM Trans Graph 2010;29(4):1–13. https://doi.org/10.1145/1778765.1778803

[13] Osher S, Fedkiw R. Level set methods and dynamic implicit surfaces. New York: Springer New York; 2003.

[14] Houston B, Nielsen MB, Batty C, Nilsson O, Museth K. Hierarchical RLE level set: a compact and versatile deformable surface representation. ACM Trans Graph 2006;25(1):151–75. https://doi.org/10.1145/1122501.1122508

[15] Klemenschits X, Selberherr S, Filipovic L. Geometric advection and its application in the emulation of high aspect ratio structures. Comput Methods Appl Mech Eng 2021;387:114–96. https://doi.org/10.1016/j.cma.2021.114196

[16] Belen RJ, Gomez S, Cooperberg D, Kiehlbauch M, Aydil ES. Feature-scale model of SI etching in SF_6/o_2 plasma and comparison with experiments. J Vac Sci Technol A 2005;23(5):1430–9. https://doi.org/10.1116/1.2013317

[17] Reiter T, Klemenschits X, Filipovic L. Modeling plasma-induced damage during the dry etching of silicon, In: Proceedings of the 2022 IEEE International Integrated Reliability Workshop (IIRW); South Lake Tahoe, CA, USA: IEEE. https://doi.org/10.1109/IIRW56459.2022.10032764 [9-14 October 2022].