

# MOMENTS: The Modular Monte Carlo Environment for Charge Transport Simulation, Overview and Applications

MARK PESKIN and CHRISTINE MAZIAR

*Microelectronics Research Center, The University of Texas at Austin, Austin, TX 78712, USA*

We present MOMENTS, a newly developed software library for Monte Carlo simulation of semiconductor devices. This library uses object-oriented design principles to provide a flexible, extensible toolset that allows rapid development of a wide variety of Monte Carlo simulation applications. It allows concurrent simulation of multiple particle species (e.g. electrons and holes) with arbitrary user-defined interactions between species (e.g. generation – recombination and carrier – carrier scattering) in arbitrary geometries using either analytic or numerical bandstructure representations. The modular design allows virtually all simulation parameters to be freely varied across the simulation domain. MOMENTS also takes advantage of the parallelism inherent in the ensemble Monte Carlo approach, employing a scheme that can support a wide variety of parallel architectures with active load balancing. To demonstrate some of the library's capabilities, we also present preliminary results from a GaAs avalanche photodiode (APD) simulator based on MOMENTS.

*Keywords:* Simulation, Monte Carlo, object-oriented, semiconductors, transport, parallel, avalanche photodiode

## INTRODUCTION

Conventionally, applications of the Monte Carlo method to semiconductor device simulation have come in the form of highly specialized codes designed specifically to address a relatively constrained and well-defined set of research questions. The scope of these specialized tools has been further limited by the tremendous computational demands of Monte Carlo simulation. However, as both conventional silicon and optoelectronic device designs continue to scale dimensionally, the Monte Carlo method, with its unmatched ability to accurately characterize the nonlocal transport phenomena that dominate the behavior

of devices with very small geometries, will become an increasingly attractive simulation option. In order to satisfy the demands of the TCAD community, future Monte Carlo simulators should possess three key properties: *flexibility* – allowing the code to be easily adapted to explore a variety of devices, geometries, and materials; *extensibility* – permitting the user to add new models and capabilities to the simulation package as needed without requiring modification of existing code; and *performance* – so that the simulator can be used cost effectively on a day-to-day basis.

We have developed a new Monte Carlo simulation environment, MOMENTS, that attempts to address these three key requirements by exploiting

advanced object-oriented and parallel software design in C++. Rather than attempting the (probably) impossible task of creating a single application capable of addressing every anticipated research need, we have instead developed MOMENTS as a library of objects that can be quickly and easily assembled to create a variety of simulation applications. Thus MOMENTS can be thought of as a toolbox or application programming interface (API) for Monte Carlo simulation. The MOMENTS API is designed to be easy to use but has many sophisticated features and is general enough to subsume most existing Monte Carlo models, allowing those models to be leveraged for use in new problems.

## DESIGN PRINCIPLES

MOMENTS comprises a collection of C++ classes designed to represent the functional components of a Monte Carlo simulation. These classes are organized hierarchically, making full use of the inheritance properties of object oriented software. The root level classes have intentionally simple interface specifications. This promotes maximum flexibility, since any child class which supports these very general interfaces can be easily integrated into an existing simulation. Simulations are defined by the interactions between objects based on three primary root level classes: SimObject, SimRegion, and SimEvent (see Fig. 1). The SimObject class is the heart of MOMENTS. Each SimObject represents a self-contained entity that, based on its surroundings, knows how to evolve its own state over a fixed time interval. Classes to represent electrons and holes are descendants of SimObject. In addition, the user may define other particle species based on the SimObject class as long as their time-evolving behavior can be defined in the context of the simulation. Possible examples include photons, phonons, excitons, or mobile ions. This makes MOMENTS useful in a wide variety of simulation applications, including process simulation. Since SimObjects are self-con-

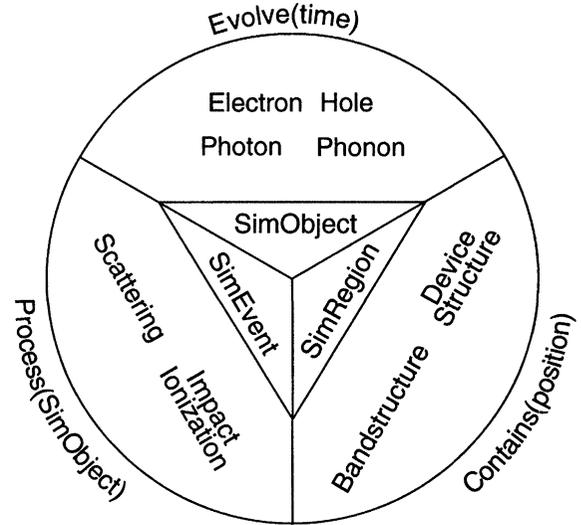


FIGURE 1 Schematic showing the three root functional classes of MOMENTS, examples of simulation elements they are designed to represent, and their primary methods.

tained, several different particle species can easily be simulated concurrently and new instances of simulated particles can be created “on-the-fly”. This allows direct study of multi-species interactions, including direct simulation of generation-recombination processes.

The SimRegion class is designed to encapsulate spatial structure data in the most general way possible; the only requirement for SimRegion objects is that they must determine whether or not they contain a given spatial coordinate. The SimRegion class also supports a hierarchy in which each SimRegion object can contain one or more sub-regions. This allows structures of arbitrary complexity and dimensionality to be represented. It also allows the programmer to create anything from bulk up through full 3-D simulations without altering the underlying models in the code. SimRegion objects, which comprise both real-space and  $k$ -space structures (bandstructures), also serve as containers for data (e.g. material data, electric fields, etc.) using the SimRegistry mechanism described below. This data storage is also hierarchical. If a SimObject requests data from a sub-region for which that data has not been

specified, the request will automatically be passed to that sub-region's parent region.

The `SimEvent` class represents stochastically occurring events that alter the state of a `SimObject` (e.g. scattering mechanisms). `SimEvents` related to a given `SimObject` class are collected in `SimEventGroups`, which can automatically handle event selection using an efficient adaptive self-scattering technique. Different `SimEventGroups` can be specified in different regions of the simulation domain. `SimEvents` can also be loaded into and unloaded from `SimEventGroups` at will, even while the simulation is running. This allows transport simulators based on MOMENTS to adaptively select the most appropriate scattering models for a given portion of the simulation. It is fairly straightforward to create new scattering models or modify the behavior of existing ones using C++ inheritance.

MOMENTS derives its flexibility from this modular design. Classes belonging to each of the three functional groups described above can be mixed and matched as needed, even during the course of a running simulation. MOMENTS is also fully extensible; it is relatively easy for users to create new `SimObject`, `SimRegion`, and `SimEvent` subclasses, using the inheritance properties of C++, and integrate them into existing codes. The modular MOMENTS structure encourages collaboration and simplifies code maintenance. It also provides potential performance benefits. Since virtually every aspect of the simulation, including scattering models, self-scattering rates, bandstructure representations, and even interpolation schemes, can be varied freely across different regions of the simulation domain, the user can focus the greatest degree of computational effort in regions of the domain where the greatest accuracy is desired.

## PARALLELISM IN MOMENTS

MOMENTS addresses performance concerns primarily by exploiting the opportunities for parallelism inherent in the Monte Carlo method.

In the MOMENTS system, the collections of `SimObjects` necessary for ensemble Monte Carlo simulations are collected in one or more `SimPopulation` objects. The members of a `SimPopulation` can then be distributed to one or more "contexts" – independent subgroups of `SimObjects` that evolve in parallel under independent threads of control. The members of each context are synchronized and allowed to interact at predetermined intervals in a manner analogous to Bulk Synchronous Parallelism [1] (see Fig. 2). While contexts are busy handling the evolution of their respective `SimObjects`, the main thread is free to perform subsidiary tasks such as re-calculation of electric fields using a Poisson solver or I/O. The MOMENTS context scheme is applicable to a wide variety of parallel architectures, including heterogeneous distributed systems, and supports active load balancing. Such parallelization of the Monte Carlo method raises several interesting issues. For example, the indeterminate timing of calls to a given random number generating sequence by different contexts would tend to prevent reproduction of a given sequence of events – a useful debugging technique. MOMENTS addresses this

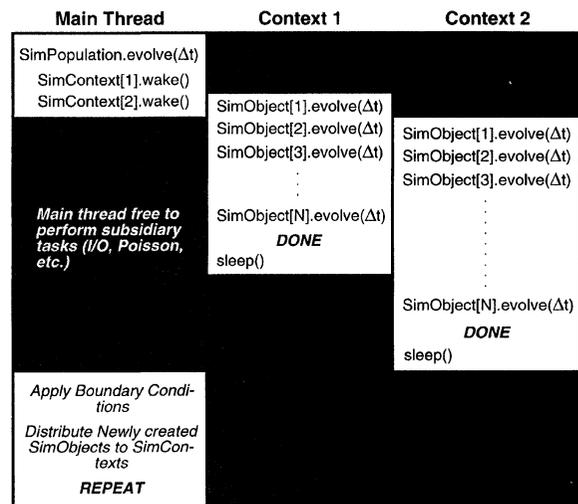


FIGURE 2 Illustration of a simulation cycle for a `SimPopulation` object with two attached contexts.

problem by providing each context with its own random number generating sequence. Calls to the `SimRand()` function are automatically directed to the generating sequence for the calling thread, shielding the user from this complexity. Another problem common to all parallel codes is data contention by multiple threads of control. Many key data structures in MOMENTS have been carefully designed to avoid problems caused by data-contention among multiple contexts.

## SUPPORT CLASSES

In addition to the three main class hierarchies described above, MOMENTS includes a number of support classes. Grids can be defined using the `SimGrid`, `SimGridPoint`, and `SimGridCell` classes. These grid classes are very general and can support all types of structured and unstructured grids in one, two, or three dimensions. Boundary conditions are handled by the `SimSource` and `SimSink` classes, which can emulate all types of conventional boundaries. In addition, these classes can be used to simulate absorption (recombination) or emission (photoemission, thermal generation, etc.) throughout entire regions of the simulation domain. The `SimRecorder` class can be used to automatically record the characteristics of one or more `SimObjects` at specified intervals. MOMENTS is also equipped with a variety of objects to handle thread control in parallel implementations, including a number of thread-safe data structures.

One of the key support classes is the `SimRegistry` – a construct that allows members of a “client” class to transparently create storage in every member of a `SimRegistry` “host” class. The client can then address the “registered” data item for a given object of the host class using an array-style notation. Storage for this registered data is automatically created in any new objects based on the host class. The `SimRegistry` concept is important because it allows users to extend the

functionality of existing host classes without actually having to modify the source code of the host class. Also, any number of client classes can use the same host class for storage. `SimRegion`, `SimGridPoint`, `SimBandStruct`, and `SimThread` are all examples of `SimRegistry` host classes.

## GAAS APD SIMULATION

As an example of the capabilities of MOMENTS, we present preliminary results from a GaAs avalanche photodiode (APD) simulator based on the MOMENTS library. This simulator will be used to study the anomalous multiplication noise-suppression recently detected in III – V compound separate absorption and multiplication (SAM) APDs with very thin ( $\leq 200$  nm) multiplication regions [2, 3]. Conventional APD theory relates the multiplication noise to the ratio of hole to electron ionizations ( $k$ ) in the multiplication region of the device. Smaller values of  $k$  are associated with less noise. The lower observed noise values in devices with very thin multiplication regions may be associated with phenomena related to the nonlocality of the impact ionization process. One such phenomenon is known as the “dead space effect”, referring to the fact that charge carriers with low initial energies must be accelerated through a high field for a sufficient distance to acquire enough energy to ionize.

Obtaining a detailed understanding of the dead space effect and its relationship to APD device geometry and noise figures requires simultaneous simulation of both electrons and holes, including the ability to simulate new carriers as they are generated by impact ionization events. Moreover, accurate characterization of transport and impact ionization in high-field regions entails the use of realistic numerically calculated bandstructure data for both carrier types. The capabilities of MOMENTS make it ideally suited for the study of this complex problem. Concurrent simulation of multiple carrier species is trivial in MOMENTS, thanks

to the SimObject abstraction. The library’s ability to generate new carriers on the fly allows a rigorous simulation of the impact ionization process. Finally, the modular MOMENTS design simplifies incorporation of numerical bandstructure data based on nonlocal pseudopotential calculations. The MOMENTS package was used to create a simple 1-D GaAs APD simulator. The simulator employs realistic nonlocal pseudopotential bandstructure representations for both carrier types. In addition to impact ionization, intra and inter-valley scatterings by both optical and acoustic phonons are included. The simulator, assembled from MOMENTS components, requires less than a page of computer code.

Preliminary results from this new simulator are encouraging. Figure 3 is a graph of gain and  $k$  for several multiplication region thicknesses with a fixed field of 700 kV/cm in the multiplication region. The  $k$  values demonstrate the same rapidly decreasing trend for thinner multiplication regions seen in experimental devices. Figure 4 shows the relative ionization rate for each carrier type as a function of position along a 100 nm multiplication layer. In this graph, electrons are injected at the right boundary ( $x=0$ ) and accelerate to the left, while generated holes are accelerated back to the right. There is a clear “dead space” for each carrier

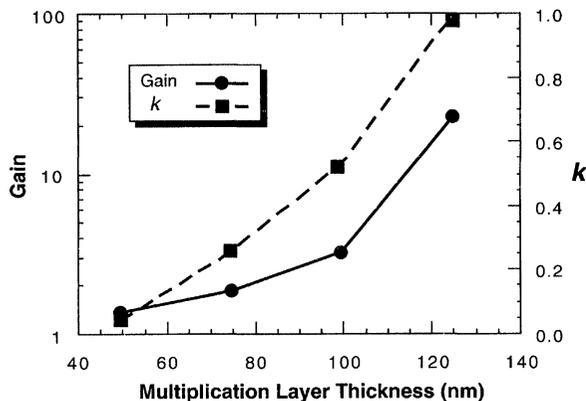


FIGURE 3 Simulated gain and  $k$ -ratio for several GaAs APD multiplication layer thicknesses with an applied field of 700 kV/cm.

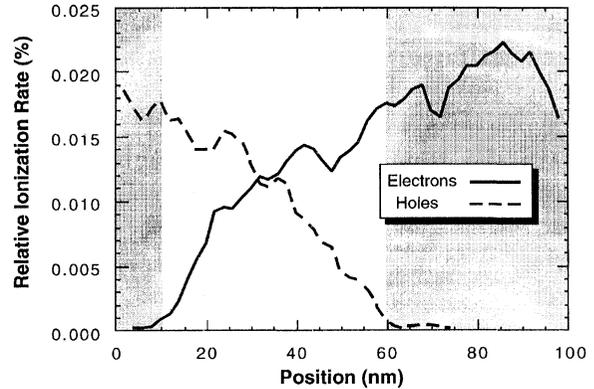


FIGURE 4 Simulated relative ionization rates as a function of position across a 100 nm multiplication layer with an applied field of 700 kV/cm.

type, indicated by the shaded regions. We believe this is the first quantification of dead space in APDs. Note that the dead space region for holes is approximately 3 – 4 times longer than that for electrons, indicating that as the multiplication layer shrinks, hole ionizations will be preferentially suppressed. Thus it appears that dead space may indeed explain the anomalously low noise levels observed in thin multiplication layer APDs, although further study will be necessary for confirmation.

## CONCLUSION

In conclusion, we have developed MOMENTS, a modular library of C++ objects that serves as a framework for development of high-performance Monte Carlo simulation applications. The library supports parallel execution in an architecture-independent manner, and provides tremendous flexibility, extensibility, maintainability, and ease-of-use – properties lacking in most conventional Monte Carlo simulators. To demonstrate the advanced capabilities of MOMENTS, we have presented preliminary results from a GaAs APD simulator based on the MOMENTS library. To the best of our knowledge this Monte Carlo tool is the first to concurrently simulate the transport of

both electrons and holes using realistic numerical bandstructure representations for both carrier species with direct simulation of the carrier generation process. We believe that MOMENTS will help make the Monte Carlo method much more accessible to the TCAD research community as a whole. This should be a tool that will be especially attractive to researchers working on novel device structures or in novel material systems.

### **Acknowledgements**

This work was supported, in part, the Joint Services Electronics Program (F49620-95-C-0045), the Texas Advanced Technology Program (TATP) and Motorola. The United States Government is authorized to distribute reprints for governmental purposes notwithstanding any copyright notation hereon.

### **References**

- [1] Valiant, L. G., "A Bridgion Model for Parallel Computation", *Communications of ACM*, August 1990.
- [2] Anselm, K. A. *et al.* (1996). "A Resonant Cavity, Separate-Absorption-and-Multiplication, Avalanche Photodiode with Low Excess Noise Factor", *IEEE Electron Device Lett.*, 17(3).
- [3] Hu, C. *et al.* (1996). "Noise Characteristics of Thin Multiplication Region GaAs Avalanche Photodiodes", *Appl. Phys. Lett.*, 69(24).

### **Authors' Biographies**

**Mark Peskin** received a B.A. in Physics (Magna cum Laude) from Wesleyan University in 1992 and an M.S.E. in Electrical Engineering in 1995 from the University of Texas at Austin, where he is currently completing a Ph.D. His research interests include semiconductor device physics, Monte Carlo and Hydrodynamic simulation, Monte Carlo parameter extraction and model development, optoelectronic simulation, non-linear optimization techniques, parallel simulation systems, and advanced simulation software development.

**Christine Maziar** received the B.S.E.E. (with Highest Distinction), the M.S.E.E. and the Ph.D., degrees from Purdue University in 1981, 1984, and 1986. In January of 1987, she joined the faculty of the University of Texas at Austin, where she is a Professor of Electrical and Computer Engineering and Vice Provost. She holds the Archie W. Straiton Endowed Teaching Fellowship in Electrical and Computer Engineering. Her research interests include modeling of charge transport in high performance device structures, semiconductor device physics, and device simulator enhancement. She has authored or coauthored over 100 journal or conference publications. She is a member of IEEE, ASEE, APS, Eta Kappa Nu, Tau Beta Pi and Sigma Pi Sigma.



# Hindawi

Submit your manuscripts at  
<http://www.hindawi.com>

