A Demonstrator Platform for Coupled Multiscale Simulation

Carlo de Falco[1,2], Georg Denk[2], and Reinhart Schultz[2]

[1] Bergische Universität Wuppertal
[2] Qimonda AG, München

Summary. In this communication we present the CoMSON Demonstrator Platform (DP), a software tool designed to help researchers in testing and validating models and algorithms for coupled simulation of nanoelectronic circuits and devices. The structure of the DP is presented with an explanation of the motivations behind the critical design choices. A multilevel simulation of a CMOS AND gate using two different coupling algorithms is provided as an application example. The example is intended to demonstrate the suitability of the DP as a flexible prototyping environment and its ability to cope with real life industrial problems. In the numerical simulations both the semi-classical Drift-Diffusion model (DD) and a Quantum Corrected DD model (QCDD) are employed and their predictions are compared.

1 Introduction

Currently, to design new integrated circuits or to port existing designs to a new technological platform, designers follow a path composed of different, almost independent, steps. At each stage of this path different software tools are used to support the design flow. Process simulators are used to predict geometries, doping profiles and other physical parameters of devices that can be produced in a given technological process. Device simulators are then used to predict electrical/thermal behavior of the new devices. Using physical considerations, often based on the drift-diffusion framework with simplifying assumptions on geometry, doping profiles, material parameters, one has to define compact models to describe the device behavior with simple, explicit analytical expressions. Very often a priori considerations lack predictiveness and accurate a posteriori calibration of model parameters based on numerical simulations and experimental data is needed. The compact device models are used in circuit simulations to predict the behavior of new circuit topologies or to evaluate the performance of existing topologies implemented with new technologies. Finally, an optimization step is used to maximize circuit performance by perturbing device parameters in the vicinity of the given values. This design flow presents some disadvantages that are becoming more relevant as CMOS technology is scaled down to its physical limits. To be as accurate as possible, compact models have grown to include several hundreds of parameters (see, for example, [7]) with little or no connection with physical characteristics of the devices. The lack of connection between model parameters and physical properties renders, on one hand, very delicate and cumbersome the parameter calibration stage and, on the other hand, it makes it

almost impossible to perform an optimization of the circuits based on the geometry and doping profiles of the devices. The latter effect is even more evident at the current stage of technological advancement where not only device dimensions are being scaled but completely new device geometries are being considered (DG, Tri-Gate, GAA, FinFET, nanotubes, ... see, for example, [12])
A possible approach to the solution of the problems described above is to create simulation tools where the behavior of the devices is represented not by evaluating the explicit analytical relations given by the compact models but by performing a direct simulation based on more accurate physical models taking the complete 2D/3D device geometry and realistic doping profiles as obtained by process simulation into account. This clearly comes at the cost of a great increase in computational effort, but the advantages are many-fold. First of all the use of few physically based design variables instead of many fitting parameters gives designers a much higher level of understanding which can lead faster to better design decisions and, furthermore, it can greatly help the construction of automatic optimization tools.

2 The CoMSON Demonstrator Platform

To achieve the above goal, many open problems still need to be solved. Apart from the computational cost (which will need to be reduced as much as possible, for example via Model Order Reduction techniques [5], or parallelization, but cannot be expected to be anywhere close to that of compact models) the coupling itself can lead to instability and convergence issues that need to be addressed properly by resorting to suitable numerical schemes. For this reason within the EU RTN project CoMSON (http://www.comson.org) a Demonstrator Platform (http://www.comson.org/dem) will be developed to connect numerical simulation tools available throughout the nodes of the CoMSON consortium through a common interface. In this way, researchers willing to be confronted with the problems arising in the framework of coupled simulation will be given the opportunity to abstract from the implementation of the basic tools (device simulator, circuit simulator, heat transfer simulator, ...) and to concentrate on the coupling itself. The architecture of the Demonstrator Platform will be the main focus of this communication.

2.1 Goals of the CoMSON Demonstrator Platform

The basic idea behind the Demonstrator Platform is to provide an integrated testing framework for researchers interested in new strategies for coupling simulation tools from different physical domains. Within this framework they will be able to implement, test and assess the applicability of their methods to real life problems without having to enter the details of the lower level tools. At the same time, researchers interested in new mathematical models for the basic physical phenomena can asses their relevance for overall system behavior taking advantage of the coupling with system level simulation tools.
It has has been designed to achieve the following objectives:

- providing a fast prototyping environment in which new and existing algorithms can be tested compared and assessed;

- allowing application of the algorithms, once assessed, to real life industrial problems.

2.2 The Structure of the CoMSON Demonstrator Platform

To achieve the results listed above, the structure depicted in Fig. 1 has been devised. The main components of the DP are:

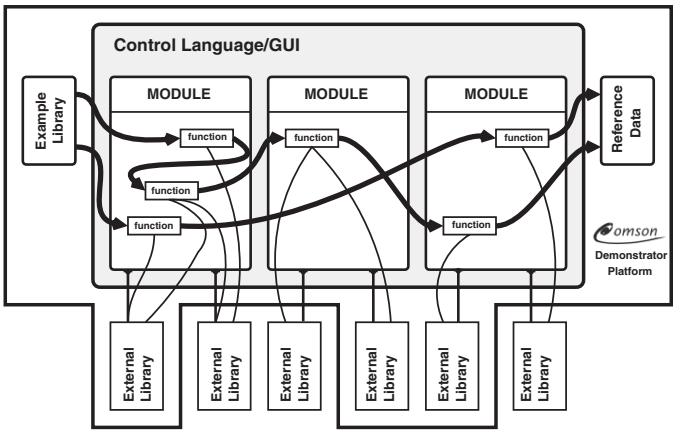


Fig. 1: The structure of the DP

1. a library of test examples and experimental measurements to be used as benchmarks for any new method,
2. a set of modules each consisting of a collection of functions providing the basic functionality of the single domain simulators,
3. a controlling programming language with which the aforementioned functions can be connected to form simulation algorithms. To separate the implementation of the basic functions from that of the coupling algorithms, the single domain simulators are organized as independent external libraries from which the DP functions are obtained via interfaces (bindings for the controlling language to the external libraries).

The test example library will contain both real-life industrial problems from the industrial nodes of the CoMSON consortium (NXP, Qimonda and STMicroelectronics) and simplified academic examples which display the same phenomena but without complications that are not essential for the understanding of the problem. This latter class of examples is especially fit for training purposes.
The initial set of functions in each module will be enriched if new algorithms will be studied that require lower level functions not initially available.
The programming language chosen as a controlling language is `Octave`. The main factors driving this choice were:

- the availability of a free language interpreter, and of a free API for building language extensions in `C, C++, Fortran`;

- the very high level of compatibility of the `Octave` interpreter with the `Matlab` programming language syntax which is the *de-facto* standard for teaching numerical algorithms;
- GPL licensing terms make it simple to distribute a fully functional system based on `Octave` including all needed software dependencies.

To better demonstrate the structure of the Demonstrator Platform and its use we will resort to a practical example. We will consider device/circuit coupling strategies belonging to two different classes:

- based on the extension of the device simulator by considering the network equations as general boundary conditions. Such an approach is used in [2] (in the case of stationary semiconductor equations) and in [1] (in the case of evolutionary semiconductor equations) to derive analytical results for the coupled system.
- Based on extension of the circuit simulator by adding the spatially discretized semiconductor equations to the system of network equations. This approach was applied in [11] for the numerical analysis of the coupled system and, together with a staggered solution approach, in [8, 9] for the simulation of the electrothermal behavior of an operational amplifier.

By implementing solvers based on such different coupling strategies, we demonstrate the flexibility of the Demonstrator Platform architecture. Moreover, we show how the abstraction layer provided by the Demonstrator Platform can be exploited for further generalization of the implemented algorithms by extending the coupling strategies considered to the case where more complex semiconductor models (like the Quantum-Corrected Drift-Diffusion class of models as described in [6]) are used for device simulation.

3 Two algorithms for coupled circuit-device simulation

In the current section we introduce two different strategies for simulating an electronic circuit where part of the composing elements is described through a full 2D Finite Element model and part is represented by lumped elements.
In Sec. (3.1) we introduce the system of equations stemming from the coupling of circuit and device equations.
The first algorithm is outlined in Sec. 3.2 and is referred to as *circuit-driven* algorithm because it is an approach that could be applied if one were to extend an existing circuit simulator to include distributed device models. The second algorithm, described in Sec. 3.3 is a viable option to extend a device simulation program based on the *Gummel Map* algorithm to include coupled simulation capabilities. In describing the algorithms we will point out which functionalities need to be exposed to the controlling language by the single domain simulators for their implementation.
For more details on implementation we invite the interested readers to download the software code and documentation which will be soon available at http://www.comson.com/dem.

3.1 The Circuit/Device Coupled Problem

Using charge/flux based MNA modeling for the network (see, for example, [10] for more details), we can write

$$\begin{array}{ll} A_q \mathbf{q}_{,t}(t) + \mathbf{f}(\mathbf{x}(t), t) = \mathbf{j}_N \\ \mathbf{q}(t) - g(\mathbf{x}(t)) \qquad\quad = 0 \end{array} \tag{1}$$

where A_q is a constant *incidence matrix*, $\mathbf{f}(\mathbf{x}(t), t)$ and $g(\mathbf{x}(t))$ are non-linear functions, $\mathbf{x}$ is a vector formed by the values of the voltage nodes and of the currents through the inductors and voltage sources and $\mathbf{q}$ is the vector containing the values of the capacitor charges and the magnetic fluxes through the inductors. $\mathbf{j}_N$ represents the currents flowing from the circuit into the contacts of the distributed device. Furthermore note that the subscript $(\cdot)_{,t}$ indicates differentiation with respect to time.
Considering, for sake of brevity, the effect of charge transport due to electron carriers only, a very general form to express the equations for the distributed device which can fit the whole class of Quantum Corrected Drift Diffusion (QCDD, see [6]) is as follows

$$\begin{array}{ll} P(\Phi, n, p) = 0 \text{ in } \Omega & \Phi|_{\Gamma_i} = \phi_i \\ n_{,t} + C_n(\Phi, n) = 0 \text{ in } \Omega & n|_{\Gamma_i} = n_i \end{array} \tag{2}$$

In (2) Φ, n, p are the electric potential, electron density and hole density inside the device computational domain Ω respectively; P and C_n are non-linear differential operators for the Poisson equation and electron current continuity equation respectively; Γ_i is the i_{th} contact of the device and Φ_i and n_i are the values of the electric potential and electron density on each of the contacts.
From the values of Φ and n one can compute the charges q_{S_i} and currents j_{S_i} at the contacts of the device as

$$\begin{array}{ll} \int_{\Gamma_i} \varepsilon \nabla \Phi \cdot \nu \, d\gamma \qquad = q_{s_i} \\ \int_{\Gamma_i} \mathbf{J}_n(\Phi, n) \cdot \nu \, d\gamma = j_{s_i} \end{array}$$

where $\mathbf{J}_n$ represents the current density in the device, and ν being the unit outward normal to the boundary of the device. Finally the circuit and device can be coupled by enforcing charge conservation:

$$\begin{array}{ll} \mathbf{j}_N + A_s(\mathbf{j}_s + \mathbf{q}_{s,t}) = 0 \\ \alpha(\phi_N + \mathbf{V}_{BI}) \qquad = A_s^T \mathbf{x} \end{array} \tag{3}$$

In (3) A_s is an incidence matrix indicating to which nodes in the network the contacts of the distributed device are connected, Φ_N are the voltages of the network nodes connected to the device and $\mathbf{V}_{BI}$ are the corresponding built-in voltages, α is a scaling factor and the vectors $\mathbf{j}_s = [j_{s_1} \dots j_{s_1}]^T$ and $\mathbf{q}_s = [q_{s_1} \dots q_{s_1}]^T$ are the currents and charges flowing through the distributed device contacts.

3.2 The Circuit-Driven Algorithm

The basic idea behind this approach is to express the complete coupled system in a form as similar as possible to the MNA equations (1).
By using (1) and (3) and discretizing in time by applying Rothe's method and a BDF(m) formula, we can write the coupled problem as

$$\begin{array}{l} \beta_0 \left(A_q \mathbf{q}(t_n) + A_s \mathbf{q}_s(t_n)\right) + \\ \quad + \mathbf{f}(\mathbf{x}(t_n), t_n) + A_s \mathbf{j}_s(A_s^T \mathbf{x}) = - \sum_{k=1 \dots m} \beta_k \left(A_q \mathbf{q}\left(t_{n-k}\right) + A_s \mathbf{q}_s(t_n)\right) \\ \mathbf{q}(t_n) - g(\mathbf{x}(t_n)) \qquad\qquad = 0 \\ \mathbf{q}_s(t_n) - g_s(A_s^T \mathbf{x}(t_n)) \qquad = 0 \end{array} \tag{4}$$

To solve this system with a Newton method we need a function to compute

- Currents and charges flowing through the distributed device contacts as a function of the node voltages
- Derivatives of such currents and charges with respect to the node voltages (local capacitance and conductance matrices)

Such function is implemented along the following lines.

1. Solve the DD equations with the Gummel map algorithm
2. Linearize the Poisson equation around the solution and compute the charges as the flux of $-\varepsilon\nabla\Phi$ through the contacts
3. Linearize the Continuity equation around the solution and compute the currents as the flux of $-\epsilon\mu_n(n\nabla\Phi - V_{th}\nabla n)$ through the contacts
4. Obtain the capacitance and conductance matrices via a Schur complement technique from the linearized Poisson (continuity) equation

The main requirement to implement this algorithm in the framework we described is that, to perform steps 2-3, we need the device simulation module to define functions that, given the contact potentials as input, produce as output the matrices for the linearized Poisson and continuity equation at each integration time point.
Once such matrices are available the computation of conductance and capacitance matrices is very straightforward.
Consider for example the Poisson equation for a device with two contacts. The discrete, linearized Poisson equation has the form

$$\begin{bmatrix} P_{11} & 0 & P_{1I} \\ 0 & P_{22} & P_{2I} \\ P_{I1} & P_{I2} & P_{II} \end{bmatrix} \begin{bmatrix} (\phi_1 + v_{BI_1})\mathbf{1}_{\Gamma_1} \\ (\phi_2 + v_{BI_2})\mathbf{1}_{\Gamma_2} \\ \Phi_I \end{bmatrix} = \begin{bmatrix} q'_{s_1} \\ q'_{s_2} \\ 0 \end{bmatrix} \tag{5}$$

where $\mathbf{1}_{\Gamma_i}$ represents a column vector of all ones with as many elements as there are on the mesh for the i-th contact, $\mathbf{\Phi}_I$ is the vector with the values of the electric potential at the internal mesh nodes, and q'_{s_1} is the vector of the charges at the mesh nodes on the i-th contact.
The total charge at the contacts can be expressed as

$$q_{s_1} = \mathbf{1}^T_{\Gamma_1} q'_{s_1}\,; \quad q_{s_2} = \mathbf{1}^T_{\Gamma_2} q'_{s_2}$$

and by eliminating $\mathbf{\Phi}_I$ one can get a relation for the charges in terms of the contact potentials of the form

$$\begin{pmatrix} q_{s_1} \\ q_{s_2} \end{pmatrix} = \begin{pmatrix} c_{11} & c_{12} \\ c_{21} & c_{22} \end{pmatrix} \begin{pmatrix} \phi_1 \\ \phi_2 \end{pmatrix} + \dots$$

where c_{ij} is the derivative of charge q_i with respect to node voltage Φ_j.

3.3 The Device-Driven Algorithm

The Device-Driven algorithm we present is a generalization of the well-known Gummel algorithm for the solution of the DD equations where the circuit equations are included as boundary conditions for each of the decoupled problems.
To set up such algorithm we need to decouple the problem into two subproblems, corresponding to the Poisson and continuity equations respectively:

Problem A (Poisson)

$$\begin{aligned} P(A_s^T\mathbf{x}, \Phi_I, \mathbf{q}_s) &= 0 \\ \beta_0(A_q\mathbf{q}(t_n) + A_s\mathbf{q}_s(t_n)) + \mathbf{f}(\mathbf{x}(t_n), t_n) + A_s\mathbf{j}_s &= \\ &= -\sum_{k=1\ldots m} \beta_k(A_q\mathbf{q}(t_{n-k}) + A_s\mathbf{q}_s(t_{n-k})) \\ \mathbf{q}(t_n) - \mathbf{g}(\mathbf{x}(t_n)) &= 0 \end{aligned} \tag{6}$$

Problem B (Continuity)

$$\begin{aligned} C_n\left(A_s^T\mathbf{x}, \Phi_{n_I}, \mathbf{j}_s\right) &= 0 \\ \beta_0(A_q\mathbf{q}(t_n) + A_s\mathbf{q}_s(t_n)) + \mathbf{f}(\mathbf{x}(t_n), t_n) + A_s\mathbf{j}_s &= \\ &= -\sum_{k=1\ldots m} \beta_k(A_q\mathbf{q}(t_{n-k}) + A_s\mathbf{q}_s(t_{n-k})) \\ \mathbf{q}(t_n) - \mathbf{g}(\mathbf{x}(t_n)) &= 0 \end{aligned} \tag{7}$$

In (6) Φ_I is the value of the electrical potential at the internal nodes of the device mesh $\mathbf{x}(t_n)$ is the vector of the network node voltages, the network and device node charges are $\mathbf{q}(t_n)$ and $\mathbf{q}_s(t_n)$ and the current through the device contacts is $\mathbf{j}_s$. In (7) Φ_{n_I} represents the vector of the values of the quasi-Fermi potentials at the internal nodes of the device mesh. As in Sec. 3.2 a BDF(m) formula has been applied for time discretization.
Having defined the two subproblems above, the procedure to be carried out at each time step can be described as follows:

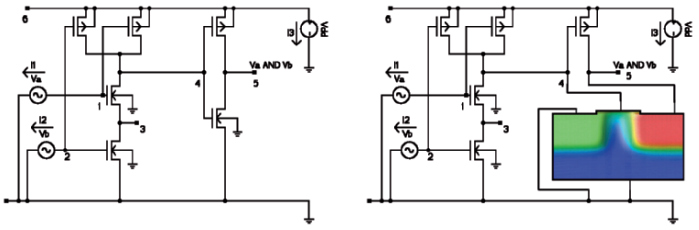
Iterate through steps 1 and 2 below until consistency is reached:
1. Solve the non-linear Poisson equation [A] for the unknowns Φ_I, $\mathbf{x}(t_n)$, $\mathbf{q}(t_n)$, $\mathbf{q}_s(t_n)$ considering $\mathbf{j}_s$ a known quantity.
2. Solve the non-linear continuity equation with unknowns Φ_{n_I}, $\mathbf{x}(t_n)$, $\mathbf{q}(t_n)$, $\mathbf{j}_s$ and considering $\mathbf{q}_s(t_n)$ fixed

Note that both step 1 and step 2 involve the solution of a system of non-linear equations so they require two more Newton loops to be nested within the iteration described above.
To be able to impose the appropriate boundary conditions we need the circuit simulation module to define a function that, given the values of the network unknowns as input, produces as output the matrices for the linearized MNA equations. This is the main requirement to be able to implement the Device-Driven method in our framework.

4 An Application Example

As an application for the algorithms described above, the test circuit in Fig.2(a), representing a CMOS AND gate, has been considered. For sake of simplicity only the n-type MOSFET in the output stage has been simulated using a full 2D simulation as shown in Fig. 2(b). The simulated device is a very aggressively scaled MOSFET

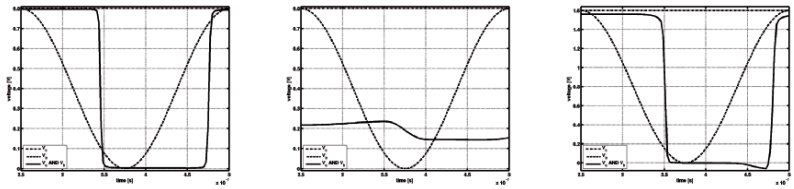
with a gate length of $15nm$. For such small devices, according to traditional scaling rules a V_{dd} voltage of $0.8V$ should be appropriate.
Fig. 3(a) displays the switching behavior of the AND gate computed with a DD Model for the distributed device and using both the Circuit and the Device-Driven algorithms. As stated in the previous section both coupling approaches can be applied



(a) The CMOS AND gate simulated (b) The set-up for coupled simulation of the AND gate

Fig. 2:

with no effort to more advanced semiconductor models. To demonstrate this we repeated the simulation using a Density Gradient model (See [3], [4] for a description of the model). The impact of quantum correction on the performance of the circuit is shown in Fig. 2(b). Essentially the circuit does not behave as a digital gate at all. This is mainly due to a shift in the threshold voltage of the device connected to the increased *Equivalent Oxide Thickness* (see [6] for a description of this effect). Indeed, as shown in Fig. 3(c), if $V_{dd} = 1.6V$ is applied, the circuit displays better performance.



(a) Simulation results using a DD model with $V_{dd} = 0.8V$ (b) Simulation results using a QDD model with $V_{dd} = 0.8V$ (c) Simulation results using a QDD model with $V_{dd} = 1.6V$

Fig. 3:

Acknowledgment

The work described here is partly financially supported by the European Commission in the framework of the CoMSON RTN project, grant number MRTN-2005-019417.

References

1. G. Alì, A. Bartel, and M. Günther, *Parabolic differential-algebraic models in electrical network design*, SIAM J. Mult. Model. Sim. **4** (2005), no. 3, 813–838.
2. G. Alì, A. Bartel, M. Günther, and C. Tischendorf, *Elliptic partial differential-algebraic multiphysics models in electrical network design*, Mathematical Models and Methods in Applied Sciences **9** (2003), no. 13, 1261–1278.
3. M. G. Ancona and G. J. Iafrate, *Quantum Correction to the Equation of State of an Electron Gas in a Semiconductor*, Phys. Rev. B **39** (1989), 9536–9540.
4. M. G. Ancona and H. F. Tiersten, *Macroscopic physics of the silicon inversion layer*, Phys. Rev. B **35** (1987), no. 15, 7959–7965.
5. T. Bechtold, E. B. Rudnyi, and Jan G. Korvink, *Fast simulation of electro-thermal mems: Efficient dynamic compact models*, Springer, 2006.
6. C. de Falco, A. L. Lacaita, E. Gatti, and R. Sacco, *Quantum–corrected drift–diffusion models for transport in semiconductor devices*, J. Comp. Phys. **204** (2005), no. 2, 533–561.
7. G. Denk, *Circuit simulation for nanoelectronics*, Proceedings of Scientific Computing in Electrical Engineering (SCEE), Springer–Verlag, 2004, pp. 13–20.
8. T. Grasser, *Mixed-mode device simulation*, Ph.D. thesis, Institut für Mikroelektronik, TU/Wien, 1999.
9. T. Grasser and S. Selberherr, *Fully coupled electrothermal mixed-mode device simulation of SiGeHBT circuits*, IEEE Transactions on Electron Devices **48** (2001), no. 7, 1421–1427.
10. M. Günther, U. Feldmann, and J. ter Maten, *Modelling and discretization of circuit problems*, Handbook of Numerical Analysis (P.G. Ciarlet, W.H.A. Schilders, and E.J.W. ter Maten, eds.), vol. XIII, Elsevier North-Holland, 2005, pp. 523–659.
11. C. Tischendorf, *Coupled systems of differential algebraic and partial differential equations in circuit and device simulation. Modeling and numerical analysis*, Habilitationsschrift, Inst. für Math., Humboldt-Univ. zu Berlin, 2003.
12. H.-S. P. Wong, *Beyond the conventional transistor*, IBM J. Res. & Dev. **46** (2002), no. 2/3, 133–169.