

Multiphysics Modelling and Simulation in Berkeley MAPP

Tianshi Wang[‡] and Jaijeet Roychowdhury

Department of Electrical Engineering and Computer Science, University of California, Berkeley

[‡]Contact author. Email: tianshi@berkeley.edu

Abstract—We introduce the newly-developed modules for multiphysics modelling and simulation in Berkeley Model and Algorithm Prototyping Platform (MAPP). Using them, developers can write compact models for not only electronic devices, but also those from optoelectronics, spintronics, microelectromechanical systems (MEMS), *etc.* They can also connect these device models into systems using multiphysics netlists and run various simulation algorithms on them. In this paper, we explain the key concepts and techniques behind these modules, and illustrate the usage of them through examples. Our plan is to make them available in open-source form as part of MAPP under the GNU Public License.

I. Introduction

For decades, the development of analog electronic circuit simulators has been playing a crucial role in the advancement of nano-electronic technology. The introduction of Berkeley SPICE [1], [2], as well as the accurate transistor models developed based on it, *e.g.*, BSIM [3], provided designers with enabling tools for implementing large-scale integrated circuits. Ever since then, progress in electronic device modelling and simulation techniques has served as one of the key driving forces behind Moore’s law.

In recent years, as Moore’s law becomes increasingly threatened by the fundamental design and manufacturing limitations of electronic devices, there has been a growing research interest in multiphysics devices and systems [4], *e.g.*, from optoelectronics, spintronics, MEMS, *etc.* Similar to nanoelectronics, the development in these fields calls for accurate device compact models as well as convenient prototyping and simulation at the system level, highlighting the need for a powerful, flexible, yet easy-to-use multiphysics simulation platform.

However, standard electronic simulators available today are not suitable for multiphysics. SPICE and its derivatives, *e.g.*, ngspice [5], were designed from the ground up for electronics. Specifically, the circuit equations are constructed in each device through “stamping”, based on hard-coded Modified Nodal Analysis (MNA) [6] and current right-hand-side (RHS) formulation [7]. While modern open-source post-SPICE circuit simulators, *e.g.*, Gnucap [8], Qucs [9] and Xyce [10], have many advantages over SPICE and its derivatives, they carry with them the same circuit modelling conventions such as MNA that have made them unsuitable for multiphysics.

As a result, much research effort has been devoted to adapting open-source SPICE-like simulators to support non-electronic systems, for MEMS [11], chemical reactions [12], neural systems [13], river networks [14], *etc.* These techniques essentially implement “wrappers” around electronic simulators for other physical domains, and only support the domain for which they are designed. Similarly, for commercial simulators, Verilog-A provides an industry standard “wrapper” for modelling non-electronic systems. However, although Verilog-A models can be written with variables from different “natures” and disciplines [15], these variables have to be either potentials or flows, corresponding to voltages and currents in the underlying circuit formulation in simulators, which still internally use hard-coded MNA and circuit laws (KCLs/KVLs). For systems that don’t obey circuit laws, *e.g.*, optoelectronic [16] and spintronic [17] systems, models are often not intuitive to write and require cumbersome connections in netlists.

In this paper, we outline our implementation of multiphysics simulation in Berkeley MAPP [18], [19]. MAPP is a MATLAB[®]-

based open-source simulation framework whose internal code structure differs markedly from that of SPICE-like simulators. At the core of MAPP is the concept of Differential Algebraic Equations (DAEs), which represent not only electronic circuits, but also general dynamic systems [6]. Device models are also DAEs, specified using the ModSpec format [20]; they describe the relations between unknowns using equations and are thus not limited to electronics either. Device connectivities are specified using MATLAB[®]-structure-based netlists. A component unique to MAPP, namely equation engine, processes netlists and device ModSpec models to construct the system DAEs. As a result, MNA is implemented as only one of the equation engines. Sparse Tableau [6] analysis can be applied by swapping just the equation engine while keeping devices and netlists unchanged [21]. Moreover, domain-specific equation engines, such as optical and chemical ones, have also been demonstrated [19]. It has been speculated [19] that such code structure of MAPP could also facilitate the implementation of a general multiphysics equation engine for systems with multiple physical domains. Recently, we have been able to demonstrate such an equation engine in MAPP. In the process, we have also extended MAPP’s devices and netlists in a modular way for multiphysics. We describe our implementations in more detail in Sec. II.

MAPP’s multiphysics modelling and simulation capabilities are considerably more general and flexible than those of Verilog-A. Specifically, systems unknowns do not have to be potentials or flows. Through the use of Network Interface Layers (NILs) [20] in ModSpec, we allow users to define physical quantities that are suitable and intuitive in their underlying physical domains, especially where circuit laws are not applicable. By writing NIL interfaces for a master equation engine, users can also define system connections, *i.e.*, they specify the multiphysics equivalents of circuit laws by controlling how DAEs are constructed. These interfaces can be easily written by implementing a few API functions that specify how to link NIL quantities to system-level DAE unknowns and equations; they are not limited to using only MNA. Moreover, we leverage MATLAB[®]’s support for complex numbers and vectors in MAPP, simplifying system connections significantly compared with Verilog-A implementations [16], [17]. Some examples of multiphysics systems prototyped in MAPP are provided in Sec. III.

II. Multiphysics Modelling and Simulation in MAPP

Devices in MAPP are modelled based on the mathematical concept of DAEs [20], which describe general equations and unknowns; they do not carry physical meanings such as voltages/currents or KCLs/KVLs by themselves. But as discussed in Sec. I, for equation engines to construct part of the system-level DAEs based on network connectivities, they require the physical meanings of device I/Os and definitions of node connections. This information is provided by a structure attached to the core model equations, namely a Network Interface Layer [20]. Recently, we have extended ModSpec to be able to include an array of NILs for encoding I/O information from multiple physical domains within a single device model.

As an example, Fig. 1 illustrates a thermistor model, where the core model specifies equations using variables v_{pn} , ipn , $temp_t$ and pwr_t . Two NILs, `eeNIL` and `thermalNIL`, are attached to

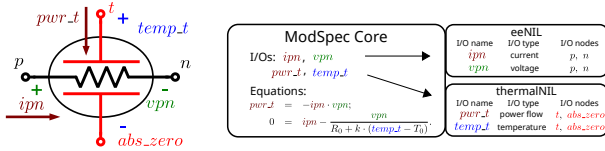


Fig. 1: Thermistor model with both electrical and thermal nodes. eeNIL and thermalNIL associate the I/Os in the ModSpec equations with their physical meanings.

this model to inform the equation engines of the physical meanings of these variables — branch voltage, current, temperature and power flow.

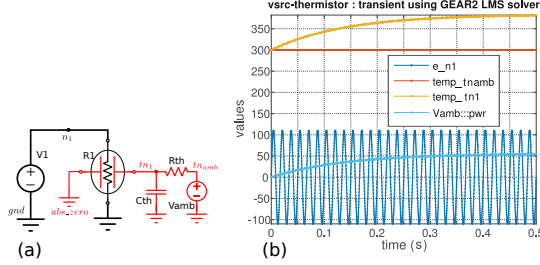


Fig. 2: Multiphysics circuit with a thermistor and its transient simulation results

Such a model can be connected to a system such as the one in Fig. 2 by specifying a multiphysics netlist, *e.g.*, Fig. 3, completely using the MATLAB[®] language.

```
1 function netlist = vsrc_thermistor_netlist()
2 netlist.name = 'vsrc_thermistor';
3
4 % regular voltage source V1
5 netlist = add_element(netlist, vsource(), 'V1', {'electrical', '1', 'gnd'}, {}, {'dc', 1});
6 % thermistor R1
7 netlist = add_element(netlist, thermistor(), 'R1', {'electrical', '1', 'gnd'}, {'thermal', 't', '0'});
8
9 % thermal resistance Rth
10 netlist = add_element(netlist, thermal_res(), 'Rth', {'thermal', 't', 'amb'}, {'Rth', 0.058});
11 % thermal capacitance Cth
12 netlist = add_element(netlist, thermal_cap(), 'Cth', {'thermal', 't', '0'}, {'Cth', 33.6});
13 % environment at ambient temperature
14 netlist = add_element(netlist, tempsource(), 'Vamb', {'thermal', 'amb', '0'}, {}, {'dc', 300});
15
16 netlist = add_domain_add_ons(netlist, {'thermal', 'abszeronodename', '0'});
17 netlist = add_domain_add_ons(netlist, {'electrical', 'groundnodename', 'gnd'});
18 end
```

Fig. 3: Code for constructing a multiphysics netlist for system in Fig. 2.

To support these multiphysics devices and netlists in MAPP, the key technique, as illustrated in Fig. 4, is to separate the components that are dependent on physical domains from those that are not. Specifically, we design a master equation engine to handle all the structures not associated with physical domains, *e.g.*, parameters, all the non-I/O unknowns, implicit equations, *etc.* Then it uses components named NIL interfaces to handle the domain-specific parts of the devices, *i.e.*, the I/Os. More specifically, NIL interfaces set up the relations between device's I/Os and system-level unknowns. Each NIL interface is an add-on component to the master equation engine, designed to handle the NIL in one physical domain. In this way, MAPP can be conveniently extended to support new physical disciplines, simply by adding new types of NILs at the device level, and the corresponding NIL interfaces for the master equation engine, without changing the device's ModSpec format or simulation algorithms.

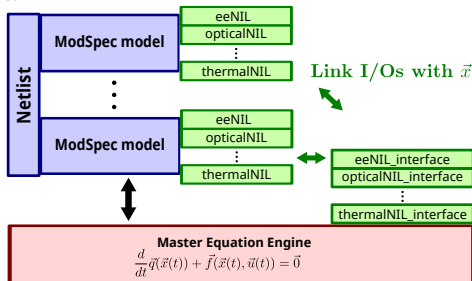


Fig. 4: MAPP uses a master equation engine with multiple NIL interfaces to construct system-level DAEs from device models and netlist structure.

III. Examples

With the help of the master equation engine and NIL interfaces, MAPP now supports all the “natures” defined in Verilog-A’s standard disciplines.vams file, including disciplines labelled as electrical, magnetic, thermal, kinematic, *etc.* All these disciplines are modelled with potentials and flows.

MAPP also supports multiphysics systems beyond this potential/flow formulation. Table 1 summarizes MAPP’s formulations for some selected physical domains. Among them, optical connections are modelled with incoming/outgoing waves, which can be thought of as time-varying phasors capturing the modulated envelopes of light. Variables at each optical ports consist of vectors of complex numbers representing these phasors at all simulation frequencies. There is no direct circuit analogy to this formulation. In chemical reaction networks, each node represents the concentration of a reactant. “KCL” at a node involves both the summation of reaction rates and a differential term for the node concentration. Using our infrastructure, such “differential-KCLs” can be defined in MAPP conveniently. To illustrate the simulation capabilities of MAPP on these multiphysics systems, we run DC and transient analyses on a spin valve and a chemical oscillator in MAPP, and show the results in Fig. 5 and Fig. 6 respectively.

domain NILs	I/Os	network connectivity
optical NIL	incoming/outgoing waves (vector of complex numbers)	two optical ports are connected at each node, the incoming waves on one side are equal to the outgoing waves on the other, and vice-versa.
spin NIL	spin voltages and currents (size-3 vectors)	spin “KCLs” and “KVLs” in vector form
chemical NIL	concentrations (C) and reaction rates (R) (scalars)	node connections represent concentrations of reactants; the sum of reaction rates at a node is equal to the change in corresponding concentration, <i>i.e.</i> , $\sum R = -\frac{d}{dt} C$.

Table 1: NIL descriptions of some selected physical domains supported in MAPP.

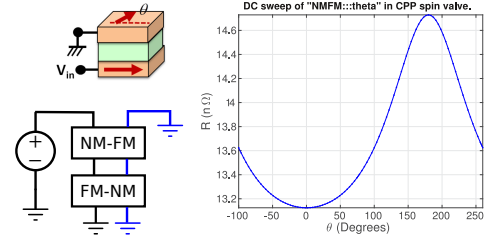


Fig. 5: DC sweep on a spin valve system in MAPP. The network contains both electrical and spintronic connections (blue wires). Device diagram is adapted from [17].

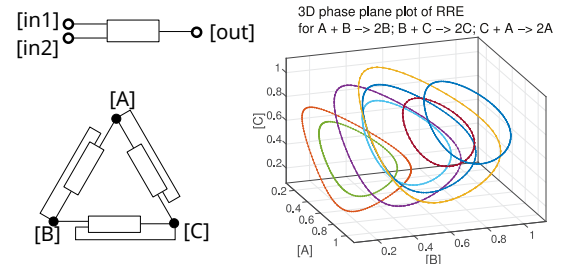


Fig. 6: Diagrams of a 2-to-1 chemical reaction, a network containing three of such reactions. Transient simulations with different initial concentrations (plotted in 3-D) show that the reaction network constitutes a chemical oscillator.

Acknowledgments

Support for the development of multiphysics simulation in MAPP has been provided by the NSF- and SRC-sponsored NEEDS (Nano-Engineered Electronic Device Simulation Node) project.

REFERENCES

- [1] L.W. Nagel. *SPICE2: a computer program to simulate semiconductor circuits*. PhD thesis, EECS department, University of California, Berkeley, Electronics Research Laboratory, 1975. Memorandum no. ERL-M520.
- [2] D. O. Pederson and A. Sangiovanni-Vincentelli. *SPICE 3 Version 3F5 User's Manual*. Dept. EECS, Univ. California, Berkeley, CA, 1994.
- [3] B. Sheu, D. Scharfetter, P.-K. Ko, and M.-C. Jeng. Bsim: Berkeley short-channel igfet model for mos transistors. *Solid-State Circuits, IEEE Journal of*, 22(4):558–566, 1987.
- [4] Semiconductor Industry Association et al. ITRS: International Technology Roadmap for Semiconductors, 2015.
- [5] P. Nenzi and H. Vogt. Ngspice Users Manual Version 26 (Describes ngspice-26 release version). 2014.
- [6] J. Roychowdhury. Numerical simulation and modelling of electronic and biochemical systems. *Foundations and Trends in Electronic Design Automation*, 3(2-3):97–303, December 2009.
- [7] A. Vladimirescu. *The SPICE book*. John Wiley & Sons, Inc., 1994.
- [8] A. Davis. The gnu circuit analysis package. 2006. <http://www.gnu.org/software/gnucap>.
- [9] M. E. Brinson and S. Jahn. Quacs: A GPL software package for circuit simulation, compact device modelling and circuit macro-modelling from DC to RF and beyond. *International Journal of Numerical Modelling: Electronic Networks, Devices and Fields*, 22(4):297–319, 2009.
- [10] E. R. Keiter, T. Mei, T. V. Russo, R. L. Schiek, H. K. Thornquist, J. C. Verley, D. A. Fixel, T. S. Coffey, R. P. Pawlowski, C. E. Warrender, et al. Xyce parallel electronic simulator users' guide, Version 6.0. 1. Technical report, Raytheon, Albuquerque, NM; Sandia National Laboratories (SNL-NM), Albuquerque, NM (United States), 2014.
- [11] E. Zhu. SUGAR 3.0: A MEMS Simulation Program (Users Guide). 2002.
- [12] J. Wyatt, D. Mikulecky, and J. DeSimone. Network modelling of reaction-diffusion systems and their numerical solution using SPICE. *Chemical Engineering Science*, 35(10):2115–2127, 1980.
- [13] R. Schiek, C. Warrender, C. Teeter, J. Aimone, H. Thornquist, T. Mei, and A. Duda. Simulating Neural Systems with Xyce. No. SAND2012-10628. Sandia National Laboratories, 2012.
- [14] F. Liu and B. Hodges. Dynamic river network simulation at large scale. In *Proceedings of the 49th Annual Design Automation Conference*, pages 723–728. ACM, 2012.
- [15] Accellera. Verilog-AMS Language Reference Manual, version 2.4, 2014.
- [16] E. Kononov. *Modeling photonic links in Verilog-A*. PhD thesis, Massachusetts Institute of Technology, 2013.
- [17] K. Camsari, S. Ganguly, and S. Datta. Modular approach to spintronics. *Scientific reports*, 5, 2015.
- [18] MAPP: The Berkeley Model and Algorithm Prototyping Platform. <http://mapp.eecs.berkeley.edu>.
- [19] T. Wang, K. Aadithya, B. Wu, J. Yao, and J. Roychowdhury. MAPP: The Berkeley Model and Algorithm Prototyping Platform. In *Proc. IEEE CICC*, 28-30 Sept. 2015.
- [20] D. Amsallem and J. Roychowdhury. ModSpec: An open, flexible specification framework for multi-domain device modelling. In *Computer-Aided Design (ICCAD), 2011 IEEE/ACM International Conference on*, pages 367–374. IEEE, 2011.
- [21] T. Wang, K. Aadithya, B. Wu, and J. Roychowdhury. MAPP: A platform for prototyping algorithms and models quickly and easily. In *2015 IEEE MTT-S International Conference on Numerical Electromagnetic and Multiphysics Modeling and Optimization (NEMO)*, pages 1–3. IEEE, 2015.