

New Capabilities for Verilog-A Implementations of Compact Device Models

M. Mierzwiński,^{*} P. O'Halloran,^{*} B. Troyanovsky,^{*} K. Mayaram,^{**} R. Dutton^{***}
^{*}Tiburon Design Automation, Inc., Santa Rosa, CA, ^{**}Oregon State University,
^{***}Stanford University

ABSTRACT

Historically, analog models and the simulators in which they are embedded form a single analog simulation kernel. This was true of SPICE and its predecessors and is true of most commercial and proprietary analog simulators in use today. As a consequence, while model interfaces generally serve the same function (allowing the model to define the differential equations of a system), the mechanics of each interface is specific to its simulator and the interface is complex and tightly interwoven with the analysis engine. Adding a new model to any analog simulator is a task that is measured in engineer-months requiring intimate knowledge of the simulator's architecture and, in some cases, thousands of lines of C code (BSIM4 is ~15k lines in SPICE3). Verilog-A, a language originally intended for behavioral modeling, has been shown to be a viable alternative to C-code [1-3] with comparable implementations typically providing an order of a magnitude reduction in the number of lines of code necessary.

Acceptance of a model requires its availability in mainstream simulators. Simulation vendors generally do not add unproven, immature models because the return on investment is never guaranteed. For these reasons the cycle of analog model development and maturation has never flourished in analog CAD – an area of CAD where perhaps having accurate predictive models is most crucial. The capability of Verilog-A to describe compact model behavior in a concise and portable fashion can only be realized if commercial simulators incorporate the interface in a consistent way.

In this paper we present simulation results using a new modular architecture implemented in both commercial and research simulators. We use industry standard models, including BSIMSOI and BSIM3, as well as MEMS models coupled into complex harmonic balance and device level simulators. This is the first demonstration of multiple commercial simulators sharing the same model binaries.

1 INTRODUCTION

Previously [1], we described a new model-simulator architecture that provided both Verilog-A OVI 2.0 language compliance as well as simulation speed of complex models comparable to C-coded models (figure 1). The architecture

is comprised of two key elements: a stand-alone Verilog-A compiler and a simulator-specific interface. The compiler generates an object file (shared library) from the Verilog-A source code. The simulator-specific interface, or run-time environment (RTE), provides the application program interface to each unique interface. The technology provides a way to effectively decouple the model specific description from the analysis requirements. The compiler and the RTE work together to provide the support necessary for a particular simulator's capabilities.

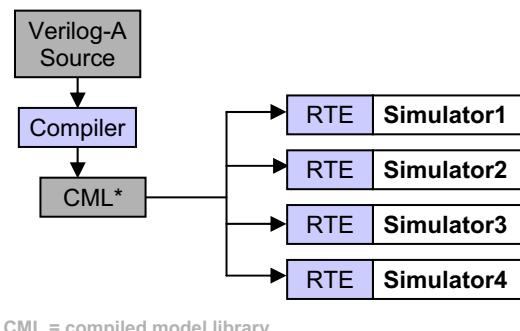


Figure 1: Tiburon Design Automation uses simulator-specific interfaces (Run Time Environments, or RTEs) to couple the same compile model library into each simulator.

The architecture is engineered to be easily embedded into existing analog simulation engines and is currently being deployed in such environments as UC Berkeley SPICE, Agilent Technologies RF/MW design environments (ADS and RFDE) [5], Eagleware's GENESYS simulator product [6], in addition to other industry-proprietary and academic simulation engines such as Oregon State University's CODECS. The Verilog-A language has been shown to be fully capable of simulating complex models. Popular compact models such as BSIM3, BSIM4, BSIMSOI, HICUM, MEXTAM, MOS9, MOS11, and Angelov have been coded in Verilog-A and validated against their C-coded counterparts. Verilog-A has been demonstrated to be a fully-capable language for compact modeling and an active committee exists to address issues to improve the language for compact models. Figure 2 shows the output of a ring oscillator circuit comparing both the simulator's C-coded version of the UC Berkeley

industry-standard BSIM3 model with a Verilog-A coded version. The results are identical, indicating the correct representation of the core model and parasitics. The simulation time is on the same order as the built-in version.

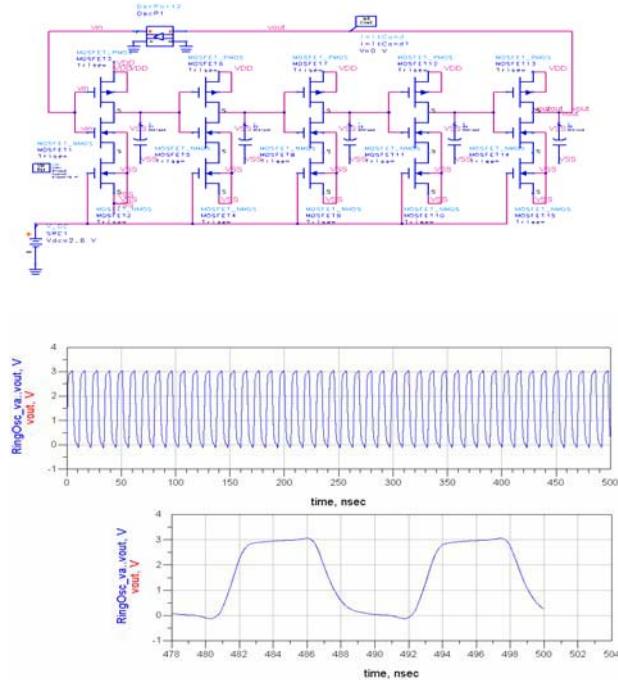
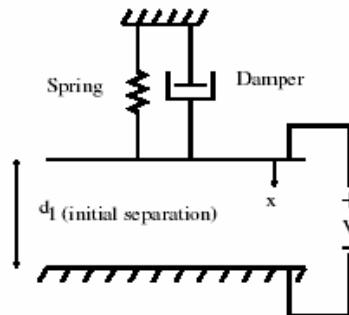


Figure 2: BSIM3 ring oscillator circuit and results showing identical behavior for the simulator “built-in” c-coded model as compared to the Verilog-A coded version.

2 VERILOG-A MODELING OF MEMS IN DEVICE SIMULATOR

CODECS is a coupled device and circuit simulator that allows accurate and detailed simulation of semiconductor circuits [4]. The simulation environment of CODECS enables modeling of critical devices within a circuit by physical (numerical) models based upon the solution of Poisson's equation and the current-continuity equations. CODECS incorporates SPICE3 for the circuit-simulation capability and for analytical models of semiconductor devices. Analytical models can be used for the non-critical devices while numerical models are provided by a one- and two-dimensional device simulator. The numerical models in CODECS include physical effects such as bandgap narrowing, Shockley-Hall-Read and Auger recombinations, concentration- and field-dependent mobilities, concentration-dependent lifetimes, and avalanche generation.

Adding Verilog-A simulation capability to CODECS gives the user the ability to abstract certain models while simulating other key models at the physical level. In this example, development of a MEMS varactor diode (figure 3)



$$m \frac{d^2x}{dt^2} + r \frac{dx}{dt} + k x = -c_0 AV^2 / 2(d_1 + x)^2$$

Figure 3: MEMS implementation of a varactor diode. The MEMS structure is modeled as a parallel plate where one plate can move and its motion is controlled by the balance of the static electric field and the equations of motion for a mass on a damped spring.

is aided by the use of differing levels of abstraction. It also allows the device to be characterized easily in more than one simulator. Figure 4 shows the setup and output for an S-parameter simulator sweep used to characterize the capacitance of the device over bias.

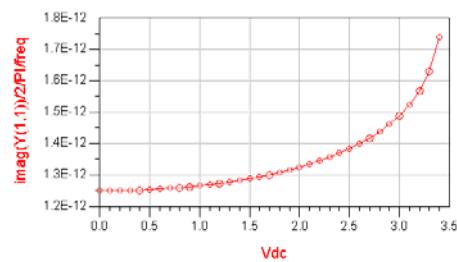
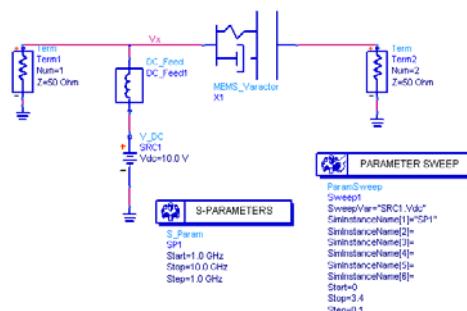


Figure 4: MEMS characterization using ADS for parameter swept S-parameter analysis.

The characterization is then used to develop a voltage-controlled oscillator circuit employing the varactors. The critical MOSFET devices are modeled at the numerical level while the MEMS device and other non-critical MOSFETs are modeled using Verilog-A. With CODECS the designer can simulate the MOS device at the numerical level, using the SPICE c-code implementation, or using a Verilog-A implementation. The Verilog-A implementation will give the same results as the C-coded version, but will

give the developer the ability to quickly and easily edit the model to add or subtract model behavior. Figure 5 shows the results of the simulations.

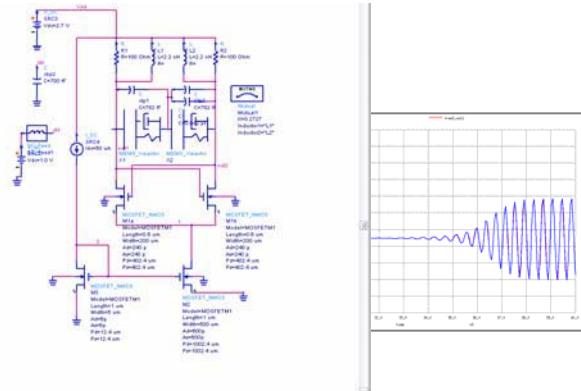


Figure 5: VCO circuit incorporating MEMS varactor diode and transient simulation of the VCO using a Verilog-A abstraction of a MEMS varactor device. These results show both MOSFETs simulated using the physical (numerical) equations as well as their Verilog-A compact model representation.

3 DEVICE SIMULATION IN MULTIPLE SIMULATORS

Since different simulators offer different analysis capabilities, modern circuit designs often require the use of multiple simulator platforms for development and verification. Agilent's Advanced Design System and Eagleware's GENESYS product are good examples of simulators that provide key simulation capabilities to RF IC and microwave designers. Sharing models between commercial simulators, even "standard" models, typically requires a great deal of verification.

Using the Tiburon-DA compiled interface, two separate commercial simulators can share the same compiled model library file (figure 6) as was used in CODECS. This dramatically reduces the chance of model implementation errors and differences as the model equations are for the most part independent of the simulator implementation. It also provides almost immediate access to new models, such as those developed using CODECS, to be distributed to end-users without loss of analysis capability or simulation performance. Compiling gives the added benefit that the shared libraries provide a good level of intellectual property protection by means of the compile process.

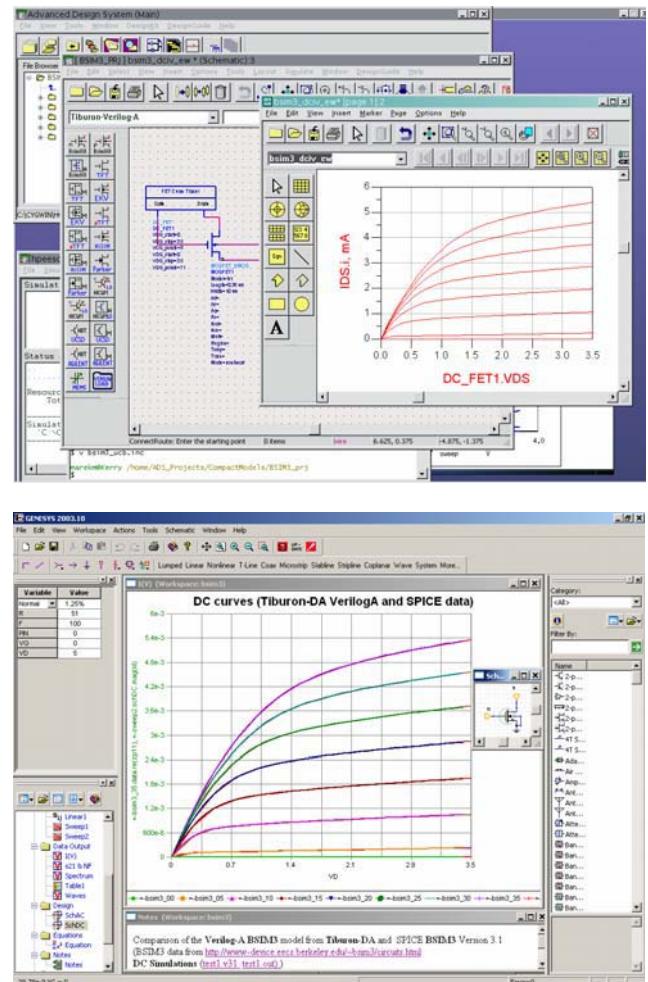


Figure 6: Agilent's ADS and Eagleware's GENESYS design suites sharing the same Verilog-A version of the BSIM3 MOSFET model that was used to simulate the CODECS MEMS VCO.

CONCLUSIONS

The ability to model at high levels of abstraction in Verilog-A, as well as to efficiently simulate complex transistor models, allows developers to easily shift between physical, compact, and abstract model domains. Verilog-A's acceptance is growing and most CAD vendors support or plan to support the language. The implementation described here provides a Verilog-A OVI 2.0 compliant solution in commercial products with simulation speeds close to that of traditional C-based models. For the first time, model developers now have a convenient development and release process for a wide range of analog models.

REFERENCES

- [1] "Changing the paradigm for compact model integration in circuit simulators using Verilog-A," M. Mierzwinski, P. O'Halloran, B. Troyanovsky, and R. Dutton, Workshop on Compact Modeling at the 6th International Conference on Modeling and Simulation of Microsystems, 2003.
- [2] "ADMS - Automatic Device Model Synthesizer," L. Lemaitre, C. McAndrew, S. Hamm, CICC 2002.
- [3] "Portable high-performance analog models using Verilog-A," Troyanovsky B., et al., presented at the workshop "Fundamentals of Nonlinear Behavioral Modeling: Foundations and Applications" at MTT 2003 in Philadelphia, June 8 2003.
- [4] "CODECS: a fixed mixed-level device and circuit simulator," Mayaram, K., Pederson, D.O., Computer-Aided Design, 1988, ICCAD-88 Digest of Technical Papers, IEEE International Conference on 7-10 Nov. 1988, pg. 112 -115.
- [5] Agilent Technologies: <http://eesof.tm.agilent.com>.
- [6] Eagleware Corporation: <http://www.eagleware.com>.