2024 8th IEEE Electron Devices Technology & Manufacturing Conference (EDTM) | 979-8-3503-7152-9/24/$31.00 ©2024 IEEE | DOI: 10.1109/EDTM58488.2024.10511990

FOSS CAD for the Compact Verilog-A Model Standardization in Open Access PDKs

Wladek Grabinski[1], Rene Scholz[2], Jason Verley[3], Eric R Keiter[3], Holger Vogt[4], Dietmar Warning[4], Paolo Nenzi[4], Francesco Lannutti[4], Felix Salfelder[5], Al Davis[5], Mike Brinson[6], Bal Virdee[6], Guilherme Torri[6], Daniel Tomaszewski[6,9], Matthias Bucher [7], Jean-Michel Sallese [7], Markus Müller[8], Pascal Kuthe[8] and Mario Krattenmacher[8]

1 MOS-AK Association 2 IHP GmbH 3 Xyce Developer 4 ngspice Developer 5 GnuCap Developer 6 Qucs Developer 7 EKV Modeling 8 SemiMod GmbH 9 Lukasiewicz IMiF

***Abstract*—The semiconductor industry continues to grow and innovate; however, companies are facing challenges in growing their workforce with skilled technicians and engineers. To meet the demand for well-trained workers worldwide, innovative ways to attract skilled talent and strengthen the local semiconductor workforce ecosystem are of utmost importance. FOSS CAD/EDA tools combined with free and open-access PDKs can serve as a new platform for bringing together IC design newbies, enthusiasts, and experienced mentors.**

Keywords– analog, CAD, CMOS, Compact model, EDA, EKV, FOSS, GnuCap, MOSFET, ngspice, open access PDK, Qucs-S, RF, SPICE, Standardization, Verilog-A, OpenVAF, Xyce.

I. Introduction

Over the last 75 years, the semiconductor industry has continued to grow and innovate. This rapid growth has also been made possible by proactive contributions from the FOSS CAD/EDA community: from SPICE simulations [1], compact modeling, Verilog-A standardization, to advanced IC designs for high-tech applications including IoT, BioMed, Agroelectronics, and others. This paper illustrates the state of FOSS CAD/EDA tools and their relation to SPICE/Verilog-A modeling/standardization. That new initiative discussed in a series of talks produced by the MOS-AK [2], fossee.in [3], FSi Foundation [4], FOSSi (Free and Open Source Silicon) [5] as well as by industrial partners as Google and eFabless promoting open access PDKs. Collaboration on FOSS IC design has recently been enabled by open-access PDKs from SkyWater [6], GF and soon by IHP Microelectronics [7] has the potential to significantly impact the semiconductor industry.

II. FOSS EKV Compact Model

The FOSS EKV2.6 MOSFET model [8] is a scalable compact model for analog-RF design and circuit simulation derived from fundamental physical properties of the MOS device structure. Its development was a joint effort by a number of compact modeling groups, which is aimed at providing the Verilog-A code of the EKV2.6 model of the MOS transistors, and distribute/share it with the FOSS CAD/EDA community to support EKV as the Standard Model for ultra low power analog/RF IC design applications [9]. The EKV 2.6 model is licensed under the ECL-2.0. In order to enable a reliable use in FOSS CAD tools, the EKV2.6 model has been calibrated for a deep-submicrometer CMOS process. The DC, AC, noise parameters of the EKV2.6 model have been extracted for a commercial 180nm CMOS process. The resulting model characteristics are in good agreement with silicon data.

III. FOSS CAD/EDA Simulators

A. ngspice: mixed-level/signal electronic circuit simulator

The *ngspice* [10] circuit simulator numerically solves equations describing (electronic) circuits made of passive and active devices for (time varying) currents and voltages. It is the open source successor of the venerable SPICE3f5 from Berkeley, representing more than 20 years of continuous development. ngspice is coded with C and available for many operating systems, and it's available under the permissive three-clause BCD license.

The input to ngspice is a netlist that describes a circuit. Based on this input it can calculate dc, ac, transient, noise, and other responses of the circuit. ngspice is a command line tool that can be used with third party tools for GUI-based schematic entry (XSCHEM [11], QUCS-S [12], and others). The output of a ngspice simulation may be saved using text files of several formats. Also, plotting capabilities with either an internal graphics tool or gnuplot are offered. Plots can be saved as SVG or PS files.

ngspice serves two major application areas. It enables simulation during circuit development with discrete devices and ICs as an integral part of a PCB design tool. So the ngspice shared library is part of the KiCad distribution. On the other hand, ngspice supports simulation during IC design. It is, for example, closely linked with the schematic editor XSCHEM and is well integrated into the Open Source PDK Google/Skywater [6] design flow. Here, the circuits are mostly made of (MOS) transistors and (parasitic) passive components.

The requirements for a simulator here are compatibility to modern compact models, large circuit capability, simulation speed, and HSPICE PDK compatibility to enable simulation of foundry processes.

Some other features offered by ngspice are: behavioral modelling by C-coded code models or B sources, true mixed-signal capability by integrating event based simulation, transient noise simulation, an integrated control scripting language to provide complex simulation scripts and data evaluation, or random parameters for Monte-Carlo simulation.

New and ongoing developments for ngspice range from an improved integration into new open source design flows, RF capability (S-parameter evaluation, harmonic balance simulation), compatibility with modern Verilog-A compact models by using the OpenVAF compiler as an ADMS replacement, speeding up the IC simulation by the KLU sparse direct linear solver [13], broaden the mixed-signal capability by improving the A/D interfaces, enabling signal integrity simulation with IBIS interfaces, and driven by KiCad developers, a new user interface.

B. The Xyce Parallel Electronic Simulator

The Xyce Parallel Electronic Simulator [14] is a SPICE-compatible circuit simulator, developed at Sandia National Laboratories. In continuous development since 1999, Xyce executes efficiently on a variety of systems, including Linux, MacOS and Windows. Xyce can also perform parallel computations, with no inherent limitation of problem size. As a mature platform for circuit simulation, Xyce supports standard capabilities available from commercial simulators, in addition to a variety of devices and models specific to Sandia's needs.

To support the addition of new compact models, Xyce currently leverages the ADMS Verilog-A translator [15]. While capable, ADMS has limitations that make it unable to translate many modern compact models. To resolve this issue, the Xyce team will be adapting Xyce to interface with models produced by the OpenVAF Verilog-A compiler. This engagement will not only increase Xyce's capability to incorporate more compact models, it will also improve Xyce's compatibility with modern PDKs that leverage Verilog-A.

C. Qucs: Quite Universal Circuit Simulator

Qucs-S [12] is a successor of the Qucs project. Qucs-S doesn't provide its own simulation kernel, but provides an unified GUI for different SPICE and non-SPICE simulation kernels. Qucs-S is written in C++ using Qt framework and operates on both Windows and Linux. Since version 0.0.23 Qucs-S is fully ported to Qt5 framework.

Besides the listed simulation kernel Qucs-S has Verilog-A synthesizer. It also allows exporting subcircuit containing unified equation defined devices (EDD) to the Verilog-A module. The obtained Verilog-A module could be linked to ngspice simulator using OpenVAF compiler.

Qucs-S uses a common schematic file format to provide a unified schematic capture input. For the most of the cases, the schematic could be shared between all simulation kernels without any adaptation. The Nutmeg equation [16] is used for the postprocessing of the ngspice simulation results in Qucs-S.

D. GnuCap: Gnu Circuit Analysis Package

Gnucap is a modular multi-language mixed-signal simulator based on plugins [17]. Gnucap supports several simulation languages, including Verilog, SPICE, and Spectre. The definition of the language is determined by a "language plugin". In addition to what is usually thought of as simulation languages, other formats such as schematics and layout could also be used. Once the plugin exists, there is a migration path from any supported format to any other.

The different formats could be equal, but Gnucap uses Verilog as its primary format. The main program supports structural Verilog directly, with emphasis on analog circuits. The behavioral part is supported by the "modelgen" model compiler.

Verilog is a big language, so there are really no single tools that support all of the variants. At this time, Gnucap supports most of "Verilog-A", some of the extras in "Verilog-AMS", with work-in-progress on the digital side. In particular, Verilog at the top level is supported as the preferred by the core, including paramset, attributes, and both continuous and discrete signals.

Models are also plugin based. All models are plugins. There are no built-in models in the traditional sense. Even the resistor is a plugin. The plugin interface is mature and well documented, and could be used as a means of interchange.

A basic set of device plugins is included with the main packages. providing most of the devices usually thought of as SPICE devices. For more models, we have "modelgen" the model compiler, and "wrappers" to enable the use of models intended for something else, such as the SPICE C models. Wrappers exists to allow use of the models from several variants of SPICE as plugins. This includes legacy models that are no longer supported by their developers and models written for old variants of SPICE. In most cases, with the appropriate wrapper the models can be used without any changes.

The model compiler, "modelgen" has been the focus of most of the work in the past year. Modelgen is a "data-flow oriented" design, focusing on the three parts: input, processing, and output. The input part is fit to the source language. The output part generates code for the target simulator. The middle "processing" part does processing and optimization.

The recent work has focused on implementation of Verilog-AMS, with the pure analog part first, supporting an event driven simulation engine, and also providing management functions such as those needed to validate a device, control time stepping, and pruning of portions of the model that could be irrelevant with some connections or parameters. Paramset is supported both in modelgen and in the core simulator.

E. Toward an open-source Verilog-A Compiler: OpenVAF

In recent years, Verilog-A has become the de-facto standard for distributing compact models, and is also supported by the compact modeling coalition (CMC) [18]. This allows for faster model development and distribution by providing a unified language. In particular for circuit simulators, the tedious and error-prone work of manual implementation of

models including all derivatives is eradicated by using a tool that converts Verilog-A to a format suitable for the simulator. Some proprietary tools for this purpose exist, however, the open-source tools VAPP [19] and ADMS [15] have serious drawbacks. Both tools operate on an abstract syntax tree (AST), a representation of the Verilog-A source, that is very close to the original code. Simulators must implement (and maintain) a simulator-specific backend for these tools that generates suitable (C) code. The backend implementation is closely tied to the tools' internals, and is hence hard to maintain. ADMS also has no active development community. SemiMod has recently developed the OpenVAF compiler to power its parameter extraction activities [20]. In the following, current development activities toward an open-source Verilog-A compiler for circuit simulators, based on OpenVAF, are described. It is believed that only an open-source tool can find wide adoption in the community and thereby facilitate Verilog-A standardization. OpenVAF operates fundamentally differently than other tools. Instead of generating circuit simulator-specific C code, it directly compiles the Verilog-A code to a shared library using LLVM [21]. By generating machine code directly, OpenVAF can use algorithms that operate on an assembly-like representation called SSA.

This shared library can then be loaded by a simulator at runtime. The interaction between compiler and simulator is reduced to the interface exposed by the shared library. Therefore, the interface can be formally defined and is simulator-independent. This allows easy integration into simulators, compared to implementing an entire simulator-specific compilation backend.

The new compiler is integrated into ngspice. The next step, which is the implementation of noise compilation, has recently been made available to the public. Additionally, a Python backend for parameter extraction, Verilog-AE, is already fully functional and freely available. A complementary parameter extraction environment including standard compact/SPICE model extraction flow is built on Python, Pandas and Verilog-AE as presented in [22].

IV. Outlook/Recommendations

To meet the growing demand for well trained engineers, software developers and IC designers globally, we must find innovative ways to attract skilled talent and strengthen the local semiconductor workforce ecosystem. FOSS CAD/EDA tools developments and its application to open source Multi-Project Wafer (MPW) shuttles with open access PDKs enable a complete open source manufacturing supply chain for analog/RF ICs and custom application specific integrated circuits (ASICs). That new initiative discussed at a series of event organized by the MOS-AK, fossee.in, FSi Foundation, FOSSi as well as by industrial partners as Google and eFabless opening Skywater and GF PDKs. A new open access PDKs recently announced by IHP Microelectronics will innovate this process in Europe.

References

[1] Laurence W. Nagel. "A Computer Program to Simulate Semiconductor Circuits, ERL-M520, UCB, May 1975". (2023), [Online]. Available: http://www.eecs.berkeley.edu/Pubs/TechRpts/1975/9602.html.

[2] MOS-AK. "Enabling Compact Modeling RD Exchange". (2023), [Online]. Available: https://www.mos-ak.org/.

[3] eSIM. "Free and open-source EDA tool". (2023), [Online]. Available: https://esim.fossee.in/.

[4] FSiF. "Free Silicon Foundation". (2023), [Online]. Available: https://wiki.f-si.org/.

[5] FOSSi. "FOSSi Foundation". (2023), [Online]. Available: https://fossi-foundation.org/.

[6] SkyWater. "Open Source SkyWater PDK". (2022), [Online]. Available: https:// github. com/ google/ skywater-pdk.

[7] IHP. "Open Source IHP PDK". (2022), [Online]. Available: https://github.com/IHP-GmbH/IHP-Open-PDK.

[8] GitHub. "FOSS EKV2.6 Compact Model". (2019), [Online]. Available: https://github.com/ekv26/model.

[9] W. Grabinski *et al.*, "FOSS EKV2.6 Verilog-A Compact MOSFET Model", in *49th ESSDERC 2019*, IEEE, 2019, pp. 190–193.

[10] ngspice. "for mixed-level/mixed-signal". (2023), [Online]. Available: http://ngspice.sourceforge.net/.

[11] xschem. "gEDA schematic capture ". (2023), [Online]. Available: http://repo.hu/projects/xschem/.

[12] Qucs-S. "Qucs with SPICE". (2023), [Online]. Available: https://ra3xdh.github.io/.

[13] F. Lannutti, P. Nenzi and M. Olivieri, "Klu sparse direct linear solver implementation into ngspice", in *19th MIXDES*, IEEE, 2012, pp. 69–73.

[14] Xyce. "Parallel electronic simulation". (2023), [Online]. Available: https://xyce.sandia.gov/.

[15] L. Lemaitre, W. Grabin´ski and C. Mcandrew, "Compact device modeling using Verilog-AMS and ADMS", *Electron Technology*, vol. 35, 2003, ISSN: 00709816.

[16] Nutmeg. "SPICE3 Post-Processor". (2023), [Online]. Available: http://sourceforge.net/projects/ngspice/.

[17] GnuCap. "New features in 0.35". (2023), [Online]. Available: https://www.gnu.org/software/gnucap/.

[18] SI2, Silicon Integration Initiative, *Compact modeling coalition*, 2020.

[19] A. Gokcen, M. X. Wang, J. Roychowdhury, A. G. Mahmutoglu and X. Wang, "New Generation Verilog-A Model Development Tools: VAPP and VALint", 2018.

[20] P. Kuthe, M. Muller and M. Schroter, "VerilogAE: An open source Verilog-A compiler for compact model parameter extraction", *J-EDS*, vol. 8, pp. 1416–1423, 2020, ISSN: 21686734.

[21] LLVM Foundation, *The LLVM Compiler Infrastructure Project*.

[22] M. Krattenmacher, "DMT-extraction: A FOSS Python framework for parameter extraction showcased by a PSP extraction workflow", [Accessed 13-11-2023], Paris, France: 35th ArbeitsKreisBipolar, 2023.