FOSS as an Efficient Tool for Extraction of MOSFET Compact Model Parameters

Daniel Tomaszewski, Grzegorz Głuszko
Division of Silicon Microsystem and Nanostructure Technology
Instytut Technologii Elektronowej
Warsaw, Poland
e-mail: dtomasz@ite.waw.pl

Vadim Kuznetsov
Department of Electronic Engineering
Bauman Moscow State Technical University
Kaluga branch, Russia;
e-mail: ra3xdh@gmail.com

Mike Brinson
Centre for Communications Technology
London Metropolitan University, UK
e-mail: mbrin72043@yahoo.co.uk

Wladek Grabinski
GMC Research, Switzerland
e-mail: wladek@grabinski.ch

***Abstract*—A GNU Octave – based application for device-level compact model evaluation and parameter extraction has been developed. The applications main features are as follows: experimental I-V data importing, generating input data for different circuit simulation programs, running the simulation program to calculate I-V characteristics of the specified models, calculating model misfit and its sensitivity to selected parameter variation, and the comparison of experimental and simulated characteristics. Measured I-V data stored by different measurement systems are accepted. Circuit simulations may be done with Ngspice, Qucs and LTSpiceIV ©. Selected aspects of the application are presented and discussed.**

***Keywords*—CMOS, Parameter extraction, Model evaluation, FOSS, GNU Octave, EDA tools**

I. INTRODUCTION

The complexity of MOSFETs which are central components in state-of-the art integrated circuits (ICs) constantly increases. as the technology matures. The evolution of MOSFET technology suggests the first devices were MOSFETs on bulk Si wafers. Next, Silicon-On-Insulator (SOI) technology was introduced, allowing for fabrication of partially depleted (PD), fully-depleted (FD), and multi-gate MOSFETs [1]. Recent technology improvements used in the design and fabrication of this device family are FinFETs [2], ultra-thin body and buried oxide (UTBB) FDSOI MOSFETs [3] and junctionless MOSFETs [4]. Thanks to the advanced features of these devices, i.e. an improved gate control over the channel conduction, and the current progress in photolithography and image patterning techniques the MOSFET lateral dimensions have shrunk down to the deep sub-100 nm range. However, a further scaling down of the device dimensions is likely to be limited by a number of physical barriers. Amongst these the most relevant ones are uncontrolled paths of a current leakage, increased power dissipation [5], and matter granularity, which via e.g. the so-called line width roughness (LWR) leads to a dramatically increased variability of the device performances and to a degradation of the IC manufacturing yield.

Device evolution is also manifested by an increasing complexity of MOSFET doping profiles. These are used to set correct values of the threshold voltages, to improve the gate control over the channel conduction and suppress current leakage mechanisms (punch-through, drain-induced barrier lowering - DIBL, gate-induced drain leakage - GIDL). In contrast, the effective doping level in the small size devices is subject to random dopant fluctuations (RDF) leading to the variability in threshold voltage and in carrier mobility, which in turn degrade the IC yield. To avoid this source of variability the MOSFETs with undoped channels have been developed [5]. However, this approach requires using the gate stacks with metal electrodes. In such small size devices the variability of a work function resulting from the metal gate granularity leads again to increased device performance spread.

The compound gate stacks mentioned above contain high-k (HK) dielectric layers, which allow to avoid thinning the gate dielectric below the limit, where the gate leakage current becomes high. At the same time they preserve the gate control over the channel conduction. However, due to their polycrystalline morphology the HK dielectrics introduce a dispersion of a the gate leakage to the total MOSFET performance variability [6].

The selection of MOSFET related properties, have been mentioned above to illustrate a variety of problems, which are to be addressed in the device-level models. Due to numerous device variants, electro-physical mechanisms, 2D/3D effects these compact models (CMs) become very complex and contain many features which are enabled/disabled conditionally. An important feature of the state-of-the art CMs is that they must operate in different domains: steady-state, small-signal, transient. They must also take into account temperature variations. They have to allow for noise, high frequency, S-parameter simulations. The CMs have many parameters. Among them there are the parameters which describe topography or have a clear physical meaning. There are parameters introduced in the CMs for a correction purpose. There are also switch-type parameters which control computation flow within the CMs.

CM complexity, including large numbers of model parameter makes model implementation and validation very difficult. In the past model implementation mainly consisted of coding a series of arithmetic expressions in a low-level

 ISBN 978-83-63578-08-4

language, typically C or FORTRAN. This step also required manual coding the model static and dynamic characteristic derivatives, which was a potential source of errors. Checking the code for errors was also a very time consuming procedure. Any change in the model required recompilation of the target circuit simulation program with the new model version. These steps were mandatory for EDA tools and device models statically linked. Often specific simulator implementation features resulted in model implementations that were not identical. This could result in noticeable differences between data from simulations of the same circuit using the same models but different programs. In order to overcome such problems a standardized approach towards compact modeling needed to been introduced. Today this is based on the Verilog-A behavioral hardware description language which is used input for the automatic synthesis of Spice C/C++ models [7]. There are several advantages of such an approach [8]. The model code is portable, i.e. independent from any simulator implementation. Verilog-A provides a means for the coding in a more compact form with automatic generation of differentiation operators, removing the need for manual coding of current and charge derivatives. The coding is more reliable, and model evaluation approaches that achieved by native C/C++ code. Model code efficiency may be done easily by a direct comparison of benchmark circuit simulations using a Verilog-A model and the corresponding Spice synthesized model. Such a functionality is available in a number of EDA tools which have been equipped with interfaces enabling running Verilog-A models. Further progress in this field has been reported in [9,10]. In the Qucs, QucsStudio programs the software not only undertakes automatic synthesis of the C++ code based on Verilog-A, but also dynamically links it with the circuit simulator program core. Furthermore, Verilog-A synthesis based on circuit schematics has been implemented. A number of other useful features, like Equation Defined Device (EDD) objects or GNU Octave interfaces are available. In this way the Qucs program, originally designed as the open-source circuit simulator, has become a very efficient CM development tool.

Until now only a few advanced MOSFET models have been developed, which include the physical effects outlined in previous sections, have reached a satisfactory accuracy, scalability, stability levels, have passed extensive tests comparing their performance against measurement data, and finally have been implemented in EDA tools for use in IC design. One of these models is the so-called threshold voltage based (V_{TH}) MOSFET model family developed at Berkeley University and represented by BSIM3, BSIM4 [11]. These models have been for a number of years industrial standards used in CMOS design kits. A main feature of the V_{TH} based modeling approach is that assumes an abrupt boundary between an accumulation/depletion (OFF) and a strong inversion (ON) ranges of device operation. In order to keep pace with MOSFET scaling numerous corrections have been introduced in such models in order to take into account, for example, of conduction mechanisms in the sub-threshold range, small size effects, realistic doping profiles, and complex gate stacks. However, the V_{TH}-based approach suffers from consequences of unavoidable interpolations used to merge subranges of the MOSFET operation area. There are also problems with passing Gummel symmetry test [12] and MOSFET distortion analysis. Two MOSFET modeling approaches have been developed to cope with the issues mentioned above, namely charge-based and surface potential-based approaches. In the charge-based MOSFET models, e.g. EKV2.6 [13], EKV3.0 [14], the threshold voltage remains an important input parameter. However, the role of the threshold voltage has been changed. It is no longer a boundary between OFF and ON states, but is used to set a so-called pinch-off voltage, which is taken to be a key model variable used both in the weak and strong inversion conditions.. In the surface-potential models, represented by PSP [15] and HiSIM [16], or in the newest member of the BSIM model family, i.e. BSIM6 [17], the threshold voltage is not the model input parameter, but is an output variable, calculated internally within the device model for information purpose.

CM complexity and large number of parameters needed to characterize recent models is challenging when device parameter extraction. The models implemented in the EDA tools are typically accompanied by corresponding parameter extraction recipes (see for example reference [18]). Such procedures usually consist of sequences of "local" fittings of the model and experimental I-V, C-V characteristics measured using the test devices or device series at the given bias and temperature conditions The parameter extraction steps are carried out using excerpts from the complete model corresponding to those conditions. The most advanced integrated system for the semiconductor device measurements, data analysis, modeling and parameter extraction is Keysight IC-CAP [19]. It provides procedures (in a form of bundles) for parameter extraction for a number of the most frequently used device-level models of the MOSFETs, bipolar transistors, diodes. For the modeling of their characteristics built-in Spice simulators are used. IC-CAP provides also a variety of optimization algorithms for the parameters extraction via a minimization of the total misfit between the model and device characteristics.

The fully integrated commercially available tools for the device-level modeling, such as IC-CAP, are expensive. On the other side there are nice Free and Open-Source Software (FOSS) solutions, e.g. [9, 20, 21], or freely available but not open-source ones [22], which provide access to the complete state-of-the art compact models, ready for use for technology characterization and circuit design in research laboratories. Developing a software for the model parameter extraction becomes an important task. In the presented paper we report initial work aimed at development of an application for the modeling of the MOSFETs and possibly other solid-state devices based on the FOSS solutions. Starting from the general description of the project a number of program features are briefly described. The paper ends with a summary pointing out at possible future work.

II. A Structure of the FOSS-based Project for CM Parameter Extraction

The overall structure of the proposed tool is shown in Fig. 1. A core of the program system is a GNU Octave application. Octave offers a high-level programming language supporting data import, processing, graphical visualization and a command shell, which allows for straight forward launching of external programs. FOSS parameter extraction software has

been used as a platform for integration of the freely available circuit simulation programs: Ngspice, LTspiceIV, Qucs. Access to another EDA tools may be easily added to the software package.

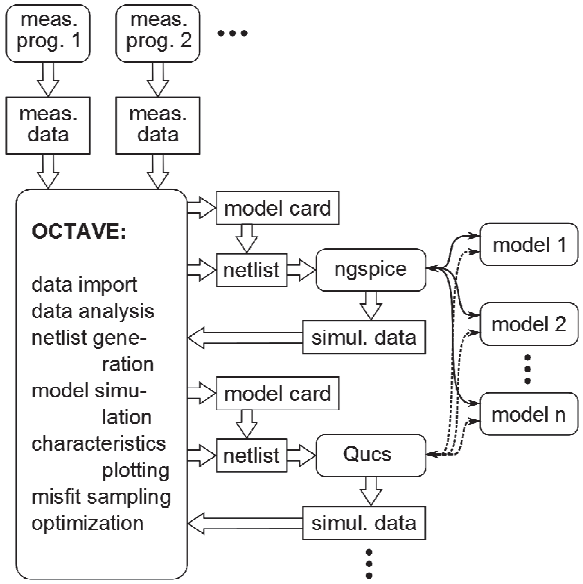

Fig. 1. Overall structure of the FOSS-based tool for modeling and parameter extraction.

The following features are implemented in the program: measurement data import, elements of data analysis, simulation of the MOSFETs using any of the available device CMs implemented in one of the interfaced simulators, sampling of the misfit between the model and experimental characteristics. Fig. 2 shows the text mode menu displayed in the Octave terminal window. This menu requests users to select a given program function.

Information about available simulation programs (i.a. paths of the executables) and about CMs implemented in these programs and enabled by a user (i.a. values of a LEVEL switch) is given in the external configuration file. It is one of aspects of the program portability.

A. *Measurement Data Import*

The program provides an interface to the measurement data saved in text *.res files by one of the measurement programs used in ITE. The measurement data are stored in the internal data structure, which can contain not only different characteristics (e.g. I_D-V_G, I_D-V_D) but characteristics of different devices and many device instances as well. This feature is connected of course with the fact, that the *.res files are created by the program controlling wafer-scale semi-automatic measurements. In Fig. 3 a simple form controlling measurement data import is shown.

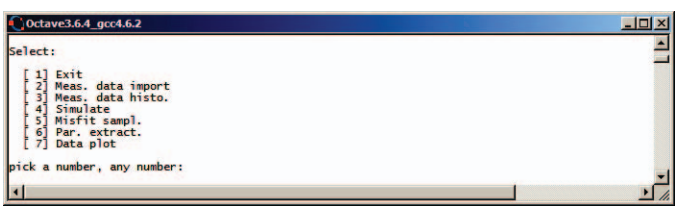

Fig. 2. Menu of the program in the GNU Octave terminal window.

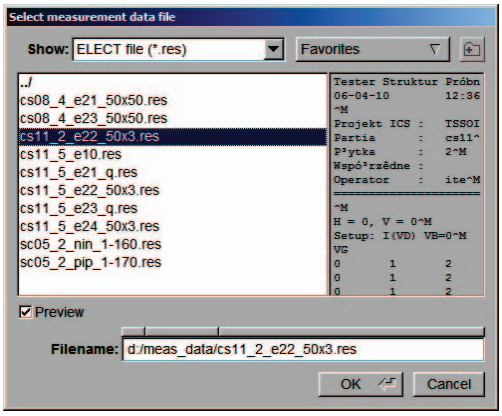

Fig. 3. Dialog window for selecting measurement data file.

After the data import is completed the user receives a report about the contents of the data file. Namely the measurement setups, and a number of dies are listed. Die sections contain the identical sets of the measurement setups. However the software parsing the measurement data is able to recognize correctly transmitted data or data sets with errors. Typically a single measurement setup corresponds to a single family of characteristics. Different setups may correspond to different characteristics of the same device or characteristics of different devices within the same die.

It is worthwhile to mention, that interfaces to data stored in *.kdf files by Keithley Interactive Test Environment (KITE) software operating on a Keithley 4200-SCS parametric analyzer, as well as to data stored in *.mdm files by IC-CAP software are under development. In the case of these formats the internal measurement data is structured to correspond to the instruments data format.

B. *Measurement Data Analysis*

At present only one type of I-V data analysis has been implemented, namely histogram plots of the device current at the maximum values of the bias voltage. Such charts provide very useful information showing the spread of the device characteristics. This function is also useful for identification of the devices, which should not be taken into account in the model parameter extraction task.

Before the histograms are generated and plotted the user receives information about the available measurement setups (Fig. 4) and is requested to specify which data should be displayed graphically. In Fig. 5 there are examples of the histograms of the I_D-V_D characteristics for n- and p-channel FDSOI MOSFETs manufactured at Instytut Technologii Elektronowej, Warsaw, Poland.

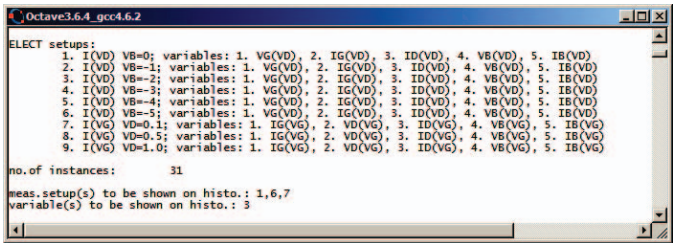

Fig. 4. A selection of the data to be shown on histograms based on the measurement data report.

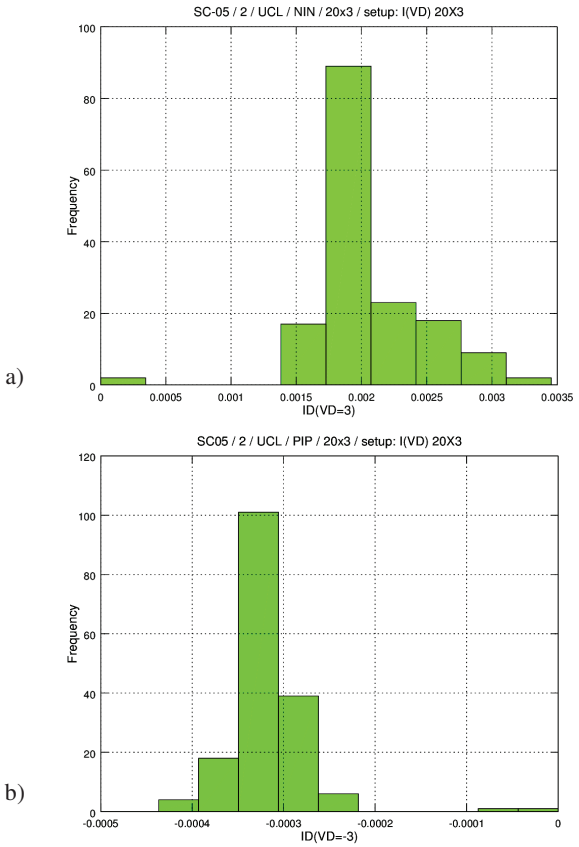

Fig. 5. Histograms of the drain current measured for a) 160 FDSOI NMOSFETs, b) 170 FDSOI PMOSFETs; W=20µm, L=3µm.

C. Modeling

The simulation of a device-level model characteristic is a key functionality of the new FOSS application. For users to be able to select the required circuit simulator for a specific set of simulations a dialogue is displayed by the Octave software. This is illustrated in Fig. 6.

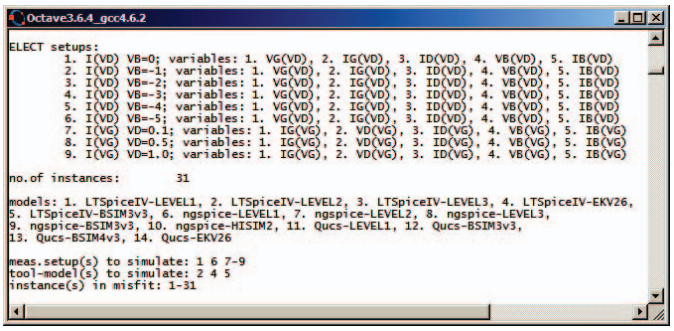

Fig. 6. Specification of the measurement setups, models and setup instances in the Octave terminal window.

The user is requested to select the required measurement setup, the simulator/model pair and the setup instances, which should be taken into account in a comparison with the simulated characteristics. It is worthwhile noticing that multiple setups, multiple simulator/model pairs as well as any available number of measured dies may be specified in the dialogue mentioned above.

Based on the given data the program prepares and outputs text files with simulation netlists and bias conditions taking into account differences existing between the input netlist syntax implemented in different Spice derived simulators. In the netlists the device instance parameters are specified, whereas the model parameters are given in the model card files. In this way the model parameters can be easily modified by Octave application or manually by the user during the program run while the netlist remains constant.

After generation of the simulation data for all the specified measurement setups and simulator/model pairs the circuit simulations are initiated. For this purpose the Octave function "system(<command>)" is used, where <command> contains the given simulator executable path together with parameters and options. For example in the case of ngspice the <command> is as follows: **"<executable path> -o <log_file> -b -r <output_text_file> <netlist_file>**". The files corresponding to each simulation have unique identifiers defined by the measurement setup, the simulation program and the model.

After circuit simulations are completed, simulation output data stored in text files are parsed and buffered in the Octave internal data structure for further processing and visualization. Shown in Fig. 7 are an example of a set of model simulation results. Octave visualization functionality allows overlaying on the same chart the measured I-V characteristics for a specified earlier number of dies together with the corresponding I-V characteristics of a number of available compact models.

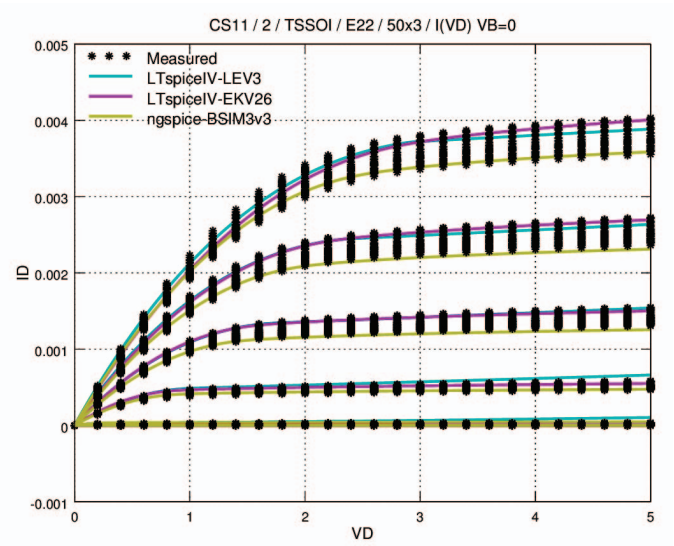

Fig. 7. I_D-V_D characteristics generated by the Octave application using two different simulators and three different models.

The first quantitative information received from the simulations is the model misfit with respect to the device characteristics. If for the given device more setups are simulated (e.g. input and output I-V characteristics) then partial misfit and total misfit values are calculated. In the misfit calculation the characteristics of any number of dies (measurement setup instances) are taken into account. Besides the misfit its sensitivities S_{Pi} to the selected parameter P_i variation are calculated. These parameters as well as their step

values necessary for the misfit sensitivity calculation are specified in additional model configuration files, one for each model. The sensitivities are determined using a simple formula (1), in which the parameters other than P_i remain constant.

$$S_{P_i} = \frac{misfit(..., P_i + \delta P_i, ...) - misfit(..., P_i - \delta P_i, ...)}{2 \cdot \delta P_i} \quad (1)$$

Hence, for a given model misfit sensitivity calculation a model card file is modified twice for each "active" parameter. An example of an output report generated by this functionality is shown in Fig. 8. The misfit sensitivity estimation is a useful option, which can help in manual tuning of the model parameter set. Currently it may be done by editing the model card files, which can be done on the fly.

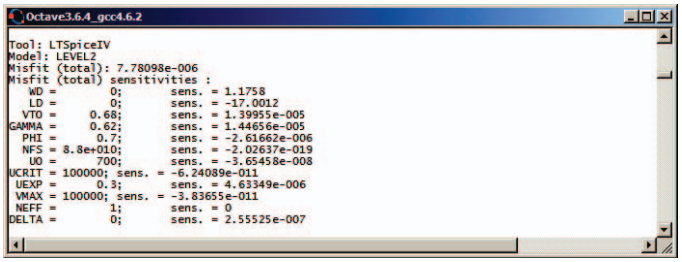

Fig. 8. The NMOS transistor model LEVEL=2 misfit and its sensitivities to parameter variations in the GNU Octave terminal window.

D. Model Misfit Sampling

In the CM parameter extraction task the misfit between the model and real device characteristics is subject to minimization. One of the method has been described in [23]. It consists in random sampling (MC method) of the parameters in the model parameter space. There are numerous variants of this approach. The samples may be followed by a deterministic "local" optimization or by e.g. an evolutionary algorithm. This compound approach, though time-consuming leads to a "global" minimization task. Analysis of the behavior of an error function (in logarithmic scale) in different cross-sections (along subsequent parameter axes) looking similarly to "tornados", gives valuable information about a position and character of the error function, thus providing an optimum model parameter set indicator.

The simplest version of this approach has been implemented in the Octave application. Currently the selected parameter sampling is carried out according to uniform distributions. The parameters as well as the borders of these distributions (the min./max. values of the parameters) are specified in the configuration files mentioned in the section C. The mechanisms for the model card file updating, netlist generation and simulator launch process are also described in the section C. An example of this functionality, based on the model LEVEL=1, is illustrated in Fig. 9. A minimum of the error function against the transconductance factor KP is clearly visible. Using such a value of KP a quite good fitting between the simplest MOSFET model and the I-V characteristics has been obtained (Fig. 10).

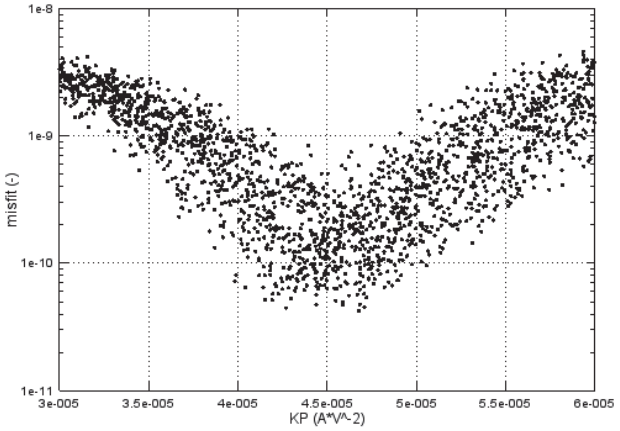

Fig. 9. Misfit sampling "tornado" for the NMOS transistor model LEVEL=1 based on I_D-V_G characteristics @ V_D=-0.1V, V_B=0,-1,…,-5V; W×L=50x50µm

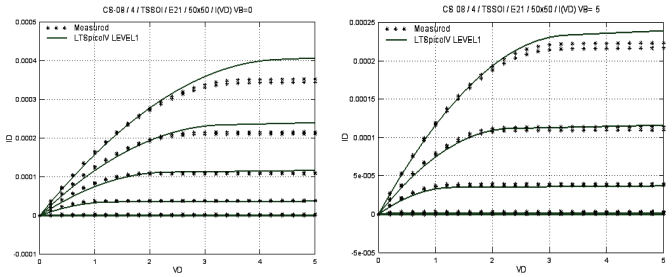

Fig. 10. Fitting of the the NMOS transistor model LEVEL=1 to I_D-V_D characteristics; V_B=0 ,-5V; W×L=50x50µm

III. SUMMARY

In the presented paper recent developments of a FOSS based application system for the compact model parameter extraction are briefly described. The system consists of a series of freely available simulators and a GNU Octave based program controlling their operation, processing the simulation results and model measurement misfit data. An important reason for using the simulators introduce earlier in the text is that they provide a range of ready for use state-of-the art compact models for MOSFET and bipolar transistors. These are either C/C++ compiled models or Verilog-A synthesized models. The Octave application system can be easily extended, firstly by adding additional EDA tools, like for example, the Xyce © and SPICE OPUS circuit simulators, secondly by further integrating the Octave FOSS software with graphical user interface software, like for example, the Qucs schematic capture package and thirdly by extending the parameter extraction process to include macromodels/ICs of current production devices.

Three main challenges for future development are worth mentioning. Firstly, development of a series of efficient interfaces to low cost general purpose measurement instrumentation is needed, like for example, an RS232 data transfer link to digital oscilloscopes, spectrum analyzers and network analyzers, secondly the addition of software procedures for a "local" sequential parameter extraction, like for example, adding new models not implemented by the

linked circuit simulators, and finally much more work is needed to improve the implemented optimization methods.

Although the FOSS software project reported in this paper is at an early stage in its development it should generate significant interest among the CM community [24, 25] to make its future development worth while

REFERENCES

[1] J.-P. Colinge, "Silicon-on-Insulator Technology: Material to VLSI", 3rd Ed., Kluwer, Boston, 2004

[2] J.-P. Colinge (Ed.), "FinFETs and Other Multi-Gate Transistors", Springer Science+Business Media, LLC, 2008

[3] J.-P. Noel, et al., "Multi- V_T UTBB FDSOI Device Architectures for Low-Power CMOS Circuit", IEEE Trans. Electron Devices, Vol. 58, No. 8, August 2011, pp. 2473-2482

[4] J. P. Colinge, et al., "Junctionless Nanowire Transistor (JNT): Properties and design guidelines", Solid-State Electronics, Vol. 65–66 (2011), Nov-Dec 2011, pp. 33–37

[5] K.J. Kuhn, "Considerations for Ultimate CMOS Scaling", IEEE Trans. Electron Devices, Vol. 59, No. 7, July 2012, pp. 1813-1828

[6] J. R. Watling, et al., "Impact of High- Gate Stacks on Transport and Variability in Nano-CMOS Devices", J. Computational and Theoretical Nanoscience, Vol.5, 2008, pp. 1072–1088

[7] L. Lemaitre, C. McAndrew, S. Hamm, "ADMS-automatic device model synthesizer", Proc. IEEE Custom Integrated Circuits Conf. (CICC 2002), pp. 27-30

[8] M.-A. Chalkiadaki, C. Valla, F. Poullet, and M. Bucher, "Why- and how- to integrate Verilog-A compact models in SPICE simulators", Int. J. Circ. Theor. Appl., 2013; 41, pp. 1203-1211, DOI: 10.1002/cta.1833

[9] M. E. Brinson, M. Margraf, "Verilog-A Compact Semiconductor Device Modelling and Circuit Macromodelling with the QucsStudio-ADMS "Turn-Key" Modelling System", Proc. 19th Int. Conf. Mixed Design of Integrated Circuits & Systems (MIXDES), 2012, Warsaw, May 24-26, 2012, pp. 94-99

[10] M. E. Brinson, V. Kuznetsov, "A new approach to compact semiconductor device modelling with Qucs Verilog-A analogue module synthesis", Int. J. Numer. Model., Article published online: April 15, 2016, doi:10.1002/jnm.2166.

[11] BSIM model web site: http://bsim.berkeley.edu/

[12] C. C. McAndrew, "Validation of MOSFET model Source-Drain Symmetry", IEEE Trans. Electron Devices, Vol. 53, No. 9, Sept 2006, pp. 2202-2206

[13] C. C. Enz, F. Krummenacher, E. A. Vittoz, "An analytical MOS transistor model valid in all regions of operation and dedicated to low-voltage and low-current applications", Analog Integrated Circuits and Signal Processing, J. Low-Voltage and Low-Power Des., vol. 8, pp. 83-114, July 1995

[14] A. Bazigos, M. Bucher, F. Krummenacher, J.-M. Sallese, A. Roy, C. Enz, M. A. Chalkiadaki, and N. Mavredakis, "EKV3 MOSFET Compact Model Documentation, Model Version 301.06", Tech. Rep., Technical University of Crete, June 23, 2011

[15] G. Gildenblat, et al., "PSP: An Advanced Surface-Potential-Based MOSFET Model for Circuit Simulation", IEEE Trans. Electron Devices, Vol. 53, No. 9, Sept 2006, pp. 1979-1993

[16] M. Miura-Mattausch, et al., "HiSIM2: Advanced MOSFET Model Valid for RF Circuit Simulation", IEEE Trans. Electron Devices, Vol. 53, No. 9, Sept 2006, pp. 1994-2007

[17] Y. S. Chauhan, .et al., "BSIM6: Analog and RF Compact Model for Bulk MOSFET", IEEE Trans. Electron Devices, Vol. 1, No. 2, Feb 2014, pp. 234-244

[18] M. Bucher, C. Lallement, C. C. Enz, "An Efficient Parameter Extraction Methodology for the EKV MOST Model", Proc. 1996 IEEE Int. Conference on Microelectronic Test Structures, Trento, March 26-28, 1996, pp. 145-150

[19] IC-CAP website: http://www.keysight.com/en/pc-1297149/ ic-cap-device-modeling-software-measurement-control-and-parameter-extraction

[20] ngspice project website: http://ngspice.sourceforge.net/

[21] Xyce project website: https://xyce.sandia.gov/

[22] LTspiceIV project website: http://www.linear.com/designtools/software/#LTspice

[23] J. Arabas, Ł. Bartnik, S. Szostak, D. Tomaszewski, "Global extraction of MOSFET parameters using the EKV model: Some properties of the underlying optimization task", Proc. 16th Int. Conf. "Mixed Design of Integrated Circuits and Systems", MIXDES 2009, June 25-27, 2009, Lodz, Poland, pp. 67-71

[24] W. Grabinski, D. Tomaszewski, "FOSS CAD for Compact/SPICE Modeling", Brussels, Jan 31 – Feb 1, 2015

[25] W. Grabinski, et al., "FOSS EKV 2.6 parameter extractor", Proc. 22nd Int. Conf. Mixed Design of Integrated Circuits & Systems (MIXDES), 2015, Torun, June 25-27, 2015, pp. 181-186