The Journal of Open Source Software

DMT: A Python toolkit for electrical engineers

Mario Krattenmacher[*] [1, 2], **Markus Müller**[†] [1, 2], **and Pascal Kuthe**[‡] [1, 2]

1 CEDIC, TU Dresden, 01062 Dresden, Germany **2** SemiMod UG (h.b.), 01159 Dresden, Germany

DOI: DOIunavailable

Software

- Review
- Repository
- Archive

Editor: Pending Editor

Reviewers:

- @Pending Reviewers

Submitted: N/A
Published: N/A

Statement of need

Currently most electrical device modeling is done using proprietary software which results in non reproducible procedures and parameters. Device Modeling Toolkit (`DMT`) aims to fill this gap by offering a standard library of tools to obtain simulation and measurement data to be prepared for standardized and reproducible parameter extraction procedures.

NOTE: mhm we should also make the extraction module public (but maybe later ??)

Summary

`DMT` offers a set of basic classes which are either used directly or intended to be subclassed.

Data inside `DMT` is stored in `DataFrame` objects. This class is derived from `pandas.DataFrame` (**pandas?**) to utilize its speed and existing indexing logic. `DMT` enhances this class by routines from `scikit-rf` (**scikit-rf?**) and own routines to calculate electrical charateristics. Correct naming of the frame columns is enabled by a `DMT` specfic naming scheme using standardized specifiers. This allows to easily and transferable handle data independent from the source.

This data can be obtained from simulation or measurement of either single devices or full electrical circuits. For this `DMT` offers different `DutView` (Device Unter Test) subclasses. In `DutView` common properties of devices are stored and a database access is offered to safely store and load data on the hard drive. The subclasses enhance these possibilities by:

- `DutMeas` allows reading, storing and accessing of measurement data
- `DutCircuit` is an abstract interface for circuit simulators. The interface is implemented for
  - Xyce (Hutchinson et al., 2002) in `DutXyce` and
  - ngspice in `DutNgspice`
- `DutTCAD` is an abstract interface for TCAD device simulators. The interface is implemented for
  - Hdev (Müller et al.) in `DutHdev`

More simulators are easy to implement and can later be used as drop in replacements for the given simulators. This is possible because `DMT` interfaces the simulators by calling them with a generated input file and reading the results into a `DataFrame`. This way only these two steps have to be implemented to allow to use a different simulator (see Figure 1).

[*] co-first author
[†] co-first author
[‡] corresponding author

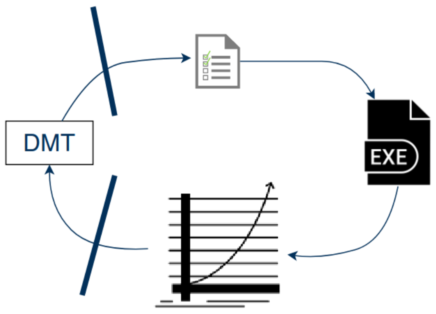

Figure 1: TODO!! DMT interfacing a circuit simulator.

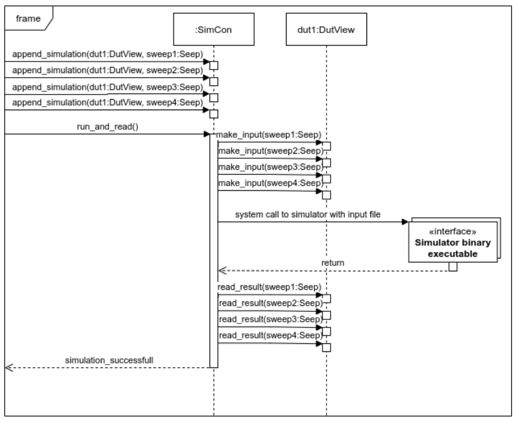
frame
:SimCon
dut1:DutView
append_simulation(dut1:DutView, sweep1:Seep)
append_simulation(dut1:DutView, sweep2:Seep)
append_simulation(dut1:DutView, sweep3:Seep)
append_simulation(dut1:DutView, sweep4:Seep)
run_and_read()
make_input(sweep1:Seep)
make_input(sweep2:Seep)
make_input(sweep3:Seep)
make_input(sweep4:Seep)
system call to simulator with input file
«interface» Simulator binary executable
return
read_result(sweep1:Seep)
read_result(sweep2:Seep)
read_result(sweep3:Seep)
read_result(sweep4:Seep)
simulation_successfull

Figure 2: Alternative: DMT interfacing a circuit simulator.

Mutliple `DutView` objects can be collected in a `DutLib` to be handled collectivly. This is usefull for technology charateriziations and model extractions, since in there processes many different devices are measured and then processes in parallel and dependent on each other.

The simulations of circuit and TCAD simulators are managed together in the `SimCon` class. This allows to call many simulations in parallel and utilize the high core count of many modern computers, although the simulators are often still single threaded. Also the simulations can be easily compared, because both circuit and TCAD simulations consists of two parts, one is the device, which is described by a `DutView`, and secondly the operation condition. The operation condition consists of for example ambient temperature and applied voltages to the dut contacts. These conditions are swept inside of a simulation and hence `DMT` offers the class `Sweep` to describe the conditions in python.

To make the electrical circuit description in `DMT` transferable from one `DutCircuit` to another, the `Circuit` class offers a strongly type tested way. The type testing is done there to ease the implementation of the interfaces. This reduces the number of possibilities of circuit topologies, but since `DMT` focuses on single devices or small circuits this trade off was chosen.

This leads to the final feature `DMT` offers. As many device models have a defined set of model parameters each with special valid ranges `MCard` allows the easy handling of the model parameters. It can read model codes, carry, save and load a list of parameters and finally be used as `Circuit` elements in circuit simulations. The parameters inside a `MCard` are always valid and can be limited to be in the valid range. These parameters can be used later in parameter extraction procedures which often are badly conditioned numerical optimizations.

Finally, for an everyday use of model engineers, `DMT` implements `Plot` to easily display device characteristics using different back-ends. The Back-ends range from `matplotlib` or `pyqtgraph` for direct to `LaTeX:pgfplots` for saving and using the plots in technical documentations or scientific publications.

Mentioned

The DMT project is already used internally by CEDIC in research and SemiMod in production. Furthermore DMT has been used by selected partners for their work.

The project is mentioned in the following papers

- (Weimer et al., 2021)
- There was a conference paper from Wladek (or someone else) where he said: If they release others can shut down ??
- others?

Related projects

DMT uses two other open source projects in order to handle Verilog-AMS files for models

- DMT uses verilogae (Kuthe et al., 2020) to access data in Verilog-AMS files. Verilog-AMS is a standardized programming language for device models. verilogae is used so the implemented model can be used simulataniously in DMT and also in circuit simulation. The circuit simulation is possible for example using the Xyce (Hutchinson et al., 2002) interface DutXyce.
- DMT is used as a front end for the open-source TCAD device simulator Hdev (Müller et al.). DutHdev implements the input generation, simulation call and result reading. The results then can be used in further calculations and plotted using the DMT plot tools.

Acknowledgements (TODO)

We acknowledge contributions from Christoph Weimer and Prof. Schröter

References

Hutchinson, S., Keiter, E., Hoekstra, R., Watts, H., Waters, A., Russo, T., Schells, R., Wix, S., & Bogdan, C. (2002). THE xyce PARALLEL ELECTRONIC SIMULATOR ? AN OVERVIEW. In *Parallel computing* (pp. 165–172). PUBLISHED BY IMPERIAL COLLEGE PRESS AND DISTRIBUTED BY WORLD SCIENTIFIC PUBLISHING CO. https://doi.org/10.1142/9781860949630_0021

Kuthe, P., Müller, M., & Schröter, M. (2020). VerilogAE: An Open Source Verilog-A Compiler for Compact Model Parameter Extraction. *IEEE Journal of the Electron Devices Society*, *8*, 1416–1423. https://doi.org/10.1109/JEDS.2020.3023165

Müller, M., Mothes, S., Claus, M., & Schröter, M. *Hdev: A 1D and 2D Hydrodynamic/Drift-Diffusion Solver for SiGe and III-V HBTs*.

Weimer, C., Sakalas, P., Müller, M., Fischer, G. G., & Schröter, M. (2021). An experimental load-pull based large-signal RF reliability study of SiGe HBTs. *2021 IEEE BiCMOS and Compound Semiconductor Integrated Circuits and Technology Symposium (BCICTS)*, 1–4. https://doi.org/10.1109/BCICTS50416.2021.9682473