

MSL Electrical Spice3 - Status and Further Development

Kristin Majetta Sandra Böhme Christoph Clauß Peter Schneider
 Fraunhofer Institute for Integrated Circuits IIS, Design Automation Division EAS
 Zeunerstr. 38, 01069 Dresden

{kristin.majetta, sandra.boehme, christoph.clauss, peter.schneider}@eas.iss.fraunhofer.de

Abstract

The Modelica Standard Library was improved by adding a package of a subset of SPICE3 models which are transformed to the Modelica language. This Spice3 library contains basic models, sources, and four semiconductor devices (diode, BJT, MOSFET level 1, resistor). Extensive tests showed the correctness of model characteristics at simple circuits. Further models already prepared will be added to the Spice3 Modelica Library later on.

Keywords: SPICE, Modelica, electronic circuit simulation, semiconductor models, netlist translator

1 Introduction

Beyond the common electric and electronic models available in the Modelica.Electrical.Analog library is has been an early aim to have Spice models for the simulation of advanced electronic circuits. Beyond the MSL two SPICE libraries already had been developed, the SPICELib [1] and the BondLib [2]. The SPICELib, which covers different complex MOSFET models, is a standalone library with its own connectors. The BondLib bases on bond graphs. It offers different levels of models related to HSPICE. At the 7th Modelica Conference the development of another SPICE library was reported as well as test issues and examples [3]. In contrast to the existing SPICE libraries the models of the new SPICE library were directly extracted from the open SPICE3 source code. These models which were improved within the ITEA research projects EURO-SYSLIB and MODELISAR, are now included into the Modelica.Electrical.Spice3 library which is available in the Modelica Standard Library version 3.2, released in October 2010.

In this paper the actual Spice3 library is presented as well as issues of testing. Furthermore, the prepared upgrading is described which concerns further models as well as a netlist translator. After all planning on a commercial extended Spice3 library is discussed.

2 The Spice3 Library of MSL 3.2

SPICE3, a simulation tool for electronic circuits, was developed in the University of California Berkeley in the nineties of the 20th century continuing the successful former version SPICE, and SPICE2 [4], [5]. It is commonly known and widely used. For a very large number of electronic circuits so called SPICE3 netlists are available, that describe the circuits for the simulation with SPICE3. The netlists are built of models of the predefined model pool of SPICE3 exclusively that contains the following models:

- Basic models (resistor, inductor, capacitor, conductor, coupled inductor, linear controlled sources)
- Semiconductor device models (different types of transistors, diode, resistor, capacitor)
- Lines (lossy and lossless transmission lines)
- Sources (different types of both voltage and current sources)

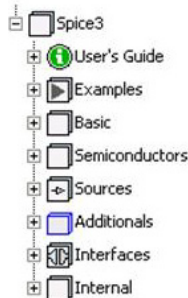
Besides SPICE3, derived simulators are known many of them are commercial, e.g. PSPICE [6], HSPICE [7]. The source code of these derivatives is, in contrast to SPICE3, not open.

The idea to have the SPICE3 models available in Modelica exists already since the beginning of the Modelica development. Between 2007 and 2010 a Spice3 library for Modelica was developed by the Fraunhofer Institute for Integrated Circuits IIS/EAS.

In the following “SPICE3” denominates Berkeley SPICE3 simulator related topics, and “Spice3” the Modelica Spice3 library.

2.1 Offered models

The models of the Modelica.Electrical.Spice3 library are thematically arranged according to the above mentioned grouping of the SPICE3 models (Picture 1). The SPICE3 line models are not yet available in Modelica. Furthermore, MSL typical packages are added (User’s Guide, Interfaces et alt.). The names of the models are composed in the way that the capital coding the model in SPICE3 (e.g. “D” for diode) is followed by an underscore and the common name of the device model, e.g. the name of the diode is *D_Diode* in the Spice3 library. This way each user independently of his knowledge of SPICE3 can indentify the model’s behavior easily.



Picture 1 Spice3 library of MSL3.2

The package *Basic* contains models of the resistor, capacitor, inductor, controlled sources and, additionally to SPICE3, the ground model. They are very similar to basic elements of the Analog library and can be used in the same way. SPICE3 netlists refer to the ground node by using the node number “0”, whereas in Modelica the ground model has to be connected.

The package *Semiconductors* contains the Mosfet level 1 (MOS1) model with the two types PMOS and NMOS. It has a set of 41 parameters and can be used for a more detailed simulation than the Mosfet transistor of the Analog package. Furthermore the BJT model, which is a bipolar transistor, was transformed into Modelica, NPN and PNP types are available. A diode model and a semiconductor resistor are also part of the package. The semiconductor resistor can either be parameterized with the usual resistance value or by its geometrical dimension (length, width) as it is in SPICE3.

The package *Sources* contains the SPICE3 voltage and current sources (constant, sinusoidal, exponential, pulse, piece wise linear and single frequency FM).

The package *Additional*s was introduced to contain important models that are not part of SPICE3 but of other SPICE derivatives. At the moment the polynomial sources of SPICE2 are part of the package. They are implemented for polynomials up to the fifth order.

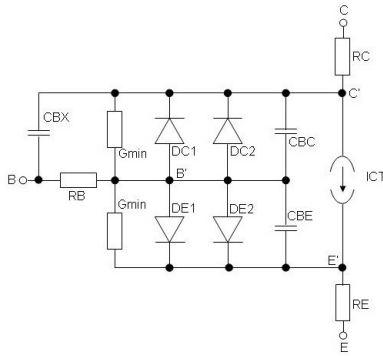
The package *Interfaces* contains only a special partial model, a twoport, that is used in the controlled sources models. The definition of the pins of the other models is used from Electrical.Analog.Interfaces. Therefore, the Spice3 models are compatible to the Electrical.Analog library and to the MSL in general.

Because the semiconductor models are very complex, many functions and data records are needed to describe the semiconductor behavior. These functions and records are collected in the *Internal* package. For the user of the Spice3 library it is not necessary to work with the *Internal* package. It is for developers only.

2.2 Principles of development

Since the SPICE3 source code is open, it is possible to extract the models directly. Years ago this was done [8]. The models were stored in a C++ library in an object oriented programming manner. This C++ library contains the exact SPICE3 models; it is tested very intensively and therefore assumed to be correct. Consequently, it was used as the base for transforming SPICE3 models into Modelica models.

In SPICE3 after linearization is calculated, in each iteration step actual representative values (R, L, G and C) of the device are written to the matrix of a linear system of equations, that basically connects the current vector with the voltage vector. Since in Modelica the terminal behavior of the model has to be described as equations this SPICE3 like linear system of equations is not necessary to be filled in by certain model functions. Instead, in Modelica a so called toplevel model is used. In Picture 2 the toplevel model of the bipolar transistor model is shown as an example, which represents the substitute circuit of the device modeled.



Picture 2 Toplevel model of bipolar transistor

The currents at the different pins (e.g. bipolar transistor: B, C, E) as well as internal currents are calculated using functions that are called in the algorithm part of the toplevel model. These functions are the main part of the model, they came from a nearly one-to-one transformation of the respectively C++ code. Also the data (parameters, variables) structure had to be rewritten into Modelica and is stored in records. The data is stored in a hierarchy in C++, because, as usual in object oriented languages, some models use the same set of parameters besides their extra own parameters. E.g. the MOS1 and MOS2 models both use nearly the same parameters. Therefore they were stored in a superior record MOS. Only the individual parameters of MOS1 and MOS2 are stored in separate records. The data structure is very complex and the number both of parameters and variables is high.

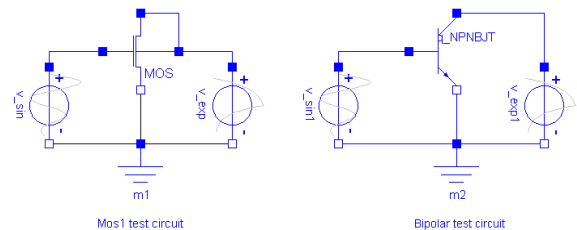
In SPICE3 two kinds of parameters are used, on the one hand the so called device parameters that are adjustable for each single device (e.g. channel length of Mos1 transistor) and on the other hand the technological parameters that are global for a group of devices (e.g. oxide thickness of Mos1 transistors). The technology parameters are set via the so called modelcard in SPICE3. That different parameter handling was transformed to Modelica by introducing modelcard records which collect the technological parameters [9]. Modelcard records are available in the Semiconductors package.

A special issue is the usage of SI units, as it is usual for the unit check in the MSL. On the one hand there are non SI units, that should not be converted to SI units in order not to use other values in the Spice3 models as it is usual in SPICE3. On the other hand, the C++-sources do not have any units at all, so they have to be added in the future.

2.3 Status of testing

Because of the high number of parameters as well as functions and especially the nonlinear behaviour, testing the Spice3 models is very complex.

To verify the simulation results of the Spice3 models SPICE3 was used as reference simulator and Dymola for the simulation of the Modelica models. Recently tests with SimulationX started. Many tests of device characteristics were developed by varying at least one parameter using a test circuit which is very simple. The device to be tested is connected to an exponential voltage source that provides the operating voltage and a sinusoidal voltage source as input. Picture 3 shows the two test circuits with the Mos1 model as well as the bipolar model.



Picture 3 Test circuits for Mos1 and Bipolar models

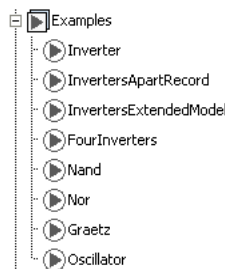
The SPICE3 reference results are stored in a text file and included in the test circuit simulation via the Modelica function CombiTimeTable for automatic comparison. This way it is possible to compare the results directly. Also the error between the Modelica calculation and the SPICE reference results is calculated during simulation. The comparisons are done for all currents of the pins of the models (e.g. for Mos1 the currents from the four pins **D**rain, **G**ate, **S**ource, **B**ulk are compared). Since the Spice3 semiconductor models have very many parameters, a lot of tests of this kind were build (about 1800). The parameters vary from reasonable and common values to completely absurd values. The analyses of the results showed, that in most cases the simulations between Modelica and SPICE3 are in accordance for the reasonable values. In the case of absurd parameter values sometimes the results differ. However, these results are often of no practical relevance (e.g. currents of Mega ampere) but show that also in extreme cases the models work still similar. The differences possibly come from the different implementations of the simulation algorithms SPICE3 and Dymola.

These parameter tests are stored as regression tests. Their simulation instructions are stored in a separate file that runs the simulations of all the parameter tests of the according semiconductor element. The comparison results between of the currents of the Modelica Spice3 model and the imported SPICE3 currents are coded by simply “true” if correct and otherwise “false” for each pin of each test circuit, and printed into a text file for a visual check. The correctness is checked at each internal simulation step by an absolute and relative error criterion the parameters of which are individually adjusted to each test circuit. After changing something at the model, the whole regression test can be repeated for validation easily to find out possibly good or bad influence of the model changes.

Beside these relatively simple tests of model characteristics, more complex circuits were tested. However, the number of transistors in these circuits was still small. These tests were not yet done intensively, but the circuits tested showed correct results compared to SPICE3. Tests of big (transistor) circuits are still necessary. Within the test process it was recognized that numerical issues may arise at larger circuits. Also the performance of the models has to be improved in comparison to other electrical simulators, e.g. SPICE3 or Saber.

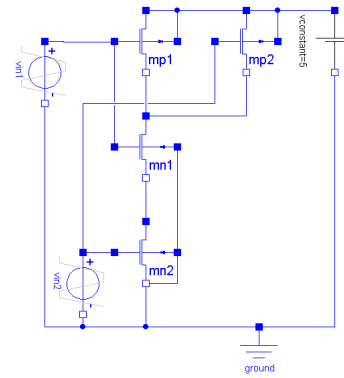
2.4 Examples

In the package *Examples* of the Spice3 library (Picture 4) eight example circuits are prepared to help the user to get an idea of the library and how to use the models.



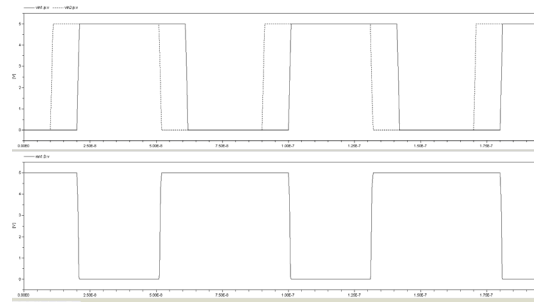
Picture 4 Package Examples of the Spice3 library

The example *Nand* is built in CMOS technology with two NMos1 and PMOS1 models (Picture 5).



Picture 5 Nand circuit of package Examples

The results are plausible (NAND function is realized) and, compared to SPICE3, correct. They are shown in Picture 6. The two upper signals are the input voltages and the lower signal is the output voltage of the NAND circuit.



Picture 6 Simulation results of the NAND circuit

To have the technology parameters available the record *modelcard* is used. In the NAND example above, the modelcards of each single transistor are filled separately. However two other possibilities are imaginable to have the modelcards available for all transistors at the same time. The first way is to make the modelcard available as instance of the circuit and give it, as a parameter, to all occurring transistors in the circuit. The second way is to define the modelcard and extend it to each transistor [3].

2.5 Open issues of the Spice3 library

The described Spice3 library was included into the MSL3.2 which was released in October 2010. The library contains a selection of the SPICE3 models. The test of the library, especially the semiconductor models, in the first instance was done via characteristic curves tests. These tests showed, that the models are working correctly. However, the test of extended electrical circuits with many transistor

models was not done yet, so information about the functionality of such real electric circuits are not yet available.

In SPICE3 it is possible to set *global parameters* to reset control options for specific simulation purpose. E.g. if GMIN is used, very small resistors are added to the circuits at certain nodes to avoid numerical difficulties. Further global parameters are:

- RELTOL (relative error tolerance)
- ABSTOL (absolute current error tolerance)
- VNTOL (absolute voltage error tolerance)
- TRTOL (transient error tolerance)
- TNOM (resetting nominal temperature)

All global parameters are described in [5]. Furthermore, SPICE3 knows *different types of analysis* like DC analysis, AC small-signal analysis, transient analysis, pole-zero analysis, small-signal distortion analysis, noise analysis and analysis at different temperatures. Since the SPICE3 models are closely related to the simulation algorithm, the models and the types of analysis are hardly clearly separable, and models are related to types of analyses. In Modelica the models and the simulation algorithm (part of the simulation tool) are separated from each other, information about the type of analysis is not part of the model but of the simulation tool. The issue is to add further types of analysis (at least AC) to Modelica and allow the models to react on the type of analysis.

In SPICE3 there are possibilities to set initial conditions to start the simulation from different states values (e.g. initial charge of a capacitance). In the actual Spice3 library for Modelica initial conditions are only available for the inductor and the capacitance model of the basic package. The *initial* parameter has to be added to the other models in the Spice3 library.

Another important point is the adding of assertions to *check parameter values* to be reasonable (e.g. temperature has to be greater than 0K, or the channel length must be positive). In SPICE3 the parameters are not checked very intensively.

A further useful feature that is not part of SPICE3 is the conditional heatport that is available in the MSL3.2. It offers the possibility of electric-thermal simulation. The heatport is a

partial model that can be included by any electrical model by inheritance. If it is included, the temperature is available within the model and the loss power has to be calculated. The heatport can be switched off. In this case the loss power flows into an internal ground and the temperature is set to the constant temperature value of the electrical model. If the conditional heatport is used, a connection between the thermal and the electrical network is available and the thermal loss power flows into the thermal network. The conditional heatport should be added in future which will change the temperature handling as modelled using fixed temperatures in SPICE3 if the heatport is switched on.

3 Further development of the MSL Spice3 package

Besides the open issues explained in section 2.5 the MSL Spice3 library will be improved by issues coming from user applications. Especially optimization of numerical issues is expected to become necessary. In this section further SPICE3 models will be described that are prepared to be added to the MSL Spice3 library as well as a netlist translator.

3.1 Planned models

To enlarge the available Spice3 model pool the Mosfet level 2 model (Mos2), the Jfet model and the coupling factor model for inductors will be added.

Mos2 model

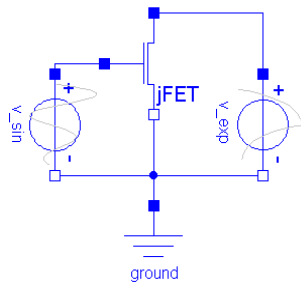
The Mos2 model is similar to the Mos1 model. Since the C++ source code is written in an object oriented manner, many of the functions and parameters of the upper Mos model were used. The Mos2 model is suitable for a smaller channel length. Actual tests of the model show, that it works correctly in principal. However a limiting precondition is, that the capacitance parameters must be set to a nonzero value to avoid numerical difficulties. The reason is, that the internal capacitance value (C_{total}) consists of two parts, the constant parameter value ($C_{parameter}$) given by the user and an internal so called Meyer capacitance value (C_{meyer}):

$$C_{total} = C_{parameter} + C_{meyer}$$

During simulation the Meyer capacitance can become zero, so the total capacitance value is also zero if the constant parameter value for the capacitance is not set. Nevertheless characteristic curve tests show that the Mos2 model is working correctly, if $C_{parameter}$ is set to a non-zero value. The capacity handling is an actual topic of research.

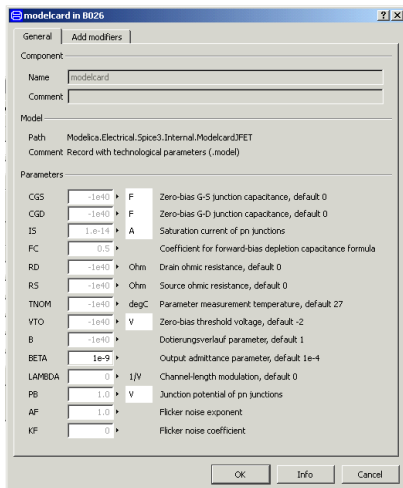
JFet model

The Junction Field Effect Transistor (Jfet, P and N Type) model is also prepared to be added to the Spice3 library. It uses the functions and parameters from the upper Fet class that is also used from the two Mosfet models of level 1 and 2. Its test is also in the first beginning comparing the characteristics using both SPICE3 and Spice3. Picture 7 shows the similar test circuit used for the characteristic curve tests of the JFet model.



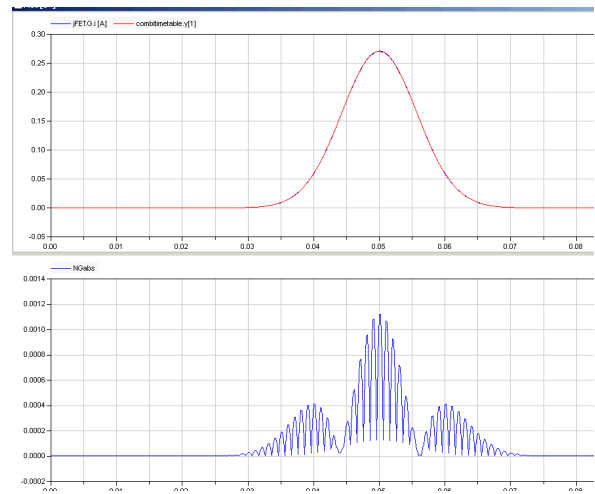
Picture 7 Characteristic curve test circuit for JFET model

As an example the influence of the transconductance parameter $BETA$ is analyzed. $BETA$ is a technology parameter that is specified in the modelcard, as it is usual in SPICE3. An impression of the JFet modelcard gives Picture 8. It can be seen that $BETA$ is given with $1e-9$ and the other parameters have their default values.



Picture 8 Jfet Modelcard

The simulation result of this circuit and the respective SPICE3 result are in accordance. As an example Picture 9 shows some simulation results.



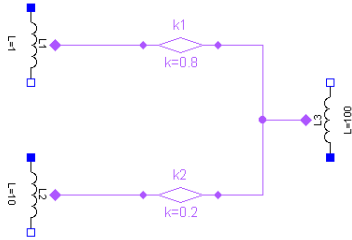
Picture 9 Simulation results of JFet test circuit

The upper picture shows the Modelica and the SPICE3 results of the gate current. These curves coincide in principle. In the lower picture the differences are shown which are in the range of mA while the maximum gate current is about 0.27A. The maximum error between the two gate currents is 0.4%. Reasons for that are the different simulation algorithms of SPICE3 and Dymola as well as the interpolation due to different step sizes.

Inductive coupling factor

In SPICE3 it is possible to model an inductive coupling between inductances via the inductive coupling factor K , which is a separate element in the SPICE3 netlist with its own element line (key letter K). It refers to two conductors which have to be defined in the netlist. To model the coupling factor in a similar way in Modelica it is suggested to introduce a new pin, the *InductiveCouplePin*, which has three variables, the inductance, the derivation of the current and, for adding the induced voltages of several couplings as a flow variable, the voltage. This pin is added to the existing inductance and to the also new introduced element *K_CoupledInductors*. Within this model, the mutual inductance M is calculated for two inductances via $M = K \sqrt{L_1 L_2}$ and with that the induced voltages $\tilde{v}_1 = M \cdot \dot{i}_2$ and $\tilde{v}_2 = M \cdot \dot{i}_1$ that are needed to complete both the voltages

$v_1 = L_1 \cdot \dot{i}_1 + M \cdot \dot{i}_2$ and $v_2 = L_2 \cdot \dot{i}_2 + M \cdot \dot{i}_1$ of the first and second inductance. This approach is still under discussion. Picture 10 gives an impression from a circuit schematic with coupled inductors.

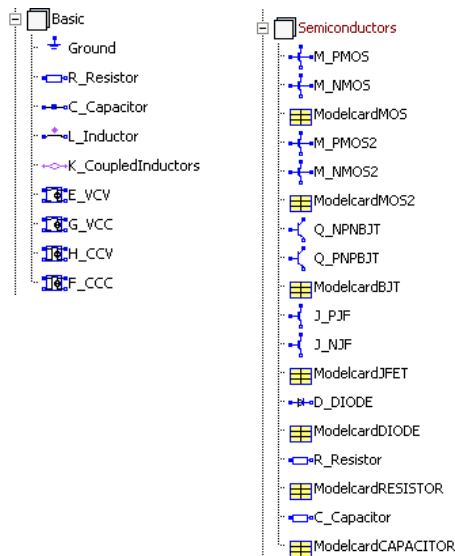


Picture 10 Circuit with coupled inductors

Semiconductor capacitance

The prepared semiconductor capacitance *C_Capacitor* is modeled like the other semiconductor devices. For technological parameters the *ModelcardCapacitor* is available which contains junction capacitances, the default width, and a narrowing factor.

Concluding the presentation of models to be added Picture 11 shows the icons of the future extended Spice3 subpackages:



Picture 11 Extended Spice3 library

3.2 Netlist translator

An important improvement will be the netlist translator [10]. It allows SPICE3 netlists to be translated to Modelica models which refer to the Models of the Modelica.Electrical.Spice3

library. The translator itself is a special Modelica model which uses essentially string function capabilities of the MSL.

As an example a SPICE3 netlist which describes a rectifier (Picture 12). The netlists file name may be rectifier.cir.

```
Rectifier
vsin 1 0 sin(0 5 50)

ri 1 2 0.1
di 2 3 diode area=1.5
c 3 0 30u
rl 3 0 1k

.model diode d cjo=3pF

.tran 0.001 0.1

.control
run
set options no break
.endc

.end
```

Picture 12 SPICE3 netlist of a rectifier

The translator is a Modelica model called *SpiceToModelica.mo* (Picture 13). It has to be adapted by inserting the name of the SPICE3 netlist file. Further parameters are the path to the Modelica Spice3 library, which will be used as import variable in the Modelica resulting model, the version which is prepared to enable different SPICE netlist derivatives in future, as well as the maximum number of modelcard, which is a temporary parameter of the actual solution.

```
model SpiceToModelica
"Start translation here. Enter file name,
switch to simulation mode and simulate.
The resulting .mo file will be created
in the working directory."
/-- Enter file name here: ---
parameter String fileName = "rectifier.cir"
"Name of file which shall be translated";
/-- Parameter Section ---
parameter String SpicePackage =
"Modelica.Electrical.Spice3.*"
"Location of Spice Package in Modelica Li-
brary";
parameter String version = "spice3"
"Which Spice Version shall be interpreted";
// not implemented yet
constant Integer iMax = 100
"Maximum number of modelcards in the file.";
/-- Main Program ---
protected
Boolean ok "Return value";
algorithm
ok := translate(fileName, SpicePackage,
help.upper(version), iMax);
end SpiceToModelica;
```

Picture 13 Translator call

Then the model `SpiceToModelica` has to be simulated. As a result the file `rectifier.mo` (Picture 14) contains the following Modelica model of the rectifier which can be simulated using the `Modelica.Electrical.Spice3` library:

```

model rectifier "Rectifier"
  import Modelica.Electrical.Spice3.*;

  parameter Semiconductors.ModelcardDIODE
    DIODE(CJO=3e-012);
  Sources.V_sin VSIN( VO=0, VA=5, FREQ=50);
  Basic.R_Resistor RI(R=0.1);
  Semiconductors.D_DIODE
    D1(modelcarddiode=DIODE,AREA=1.5);
  Basic.C_Capacitor C(C=3e-005);
  Basic.R_Resistor RL(R=1000);
  Basic.Ground g;

protected
  Modeica.Electrical.Analog.Interfaces.Pin
    n1;
  Modelica.Electrical.Analog.Interfaces.Pin
    n0;
  Modelica.Electrical.Analog.Interfaces.Pin
    n2;
  Modelica.Electrical.Analog.Interfaces.Pin
    n3;

equation
  connect(g.p,n0);
  connect(VSIN.p, n1);
  connect(VSIN.n, n0);
  connect(RI.p, n1);
  connect(RI.n, n2);
  connect(D1.p, n2);
  connect(D1.n, n3);
  connect(C.p, n3);
  connect(C.n, n0);
  connect(RL.p, n3);
  connect(RL.n, n0);

  annotation (uses(Modelica(version="3.2")),
    experiment(StopTime=0.1, Interval=0.001));
end rectifier;

```

Picture 14 Translated Modelica model of the rectifier circuit

Once the Modelica “netlist model” is available it can be used for simulation or adapted to further modelling. Especially connectors can be added as well as an icon to use the SPICE netlist as building block in further models. This way SPICE3 netlists can be integrated in Modelica modelling easily.

The Modelica function *translate* reads the input SPICE3 netlist as a string from the file given by the parameter `FileName`. Then preprocessing simplifies the netlist, e.g. by deleting comments. The resulting intermediate netlist is parsed to recognise several kinds of tokens which describe component names, node names, parameters and others. Since no Backus Naur Form of the SPICE3 input language exists, and the description in the manual is not as exact as expected, many tests were necessary to find out details. The SPICE3 input language for netlists is a context sensitive language therefore no prepared parser (e.g. `lex` [11]) can be applied. The used parser was constructed by taking into consideration the basic element notification of the input language as well as the specifics of many elements. The Modelica string functions were intensively used.

It is planned to add the translator which is adapted to the offered SPICE3 model pool, to `Spice3` Modelica subpackage utilities.

4 Commercial fully extended Spice3 Library

There are still SPICE3 device models remaining which are not yet transformed to Modelica:

- MOSFET Level 3
- MOSFET Level 4 (BSIM 1)
- MOSFET Level 5 (BSIM 2)
- MOSFET Level 6 (modified Level 3)
- MESFET

These models can be transformed to Modelica in the same way as described in section 2.2. It is planned to have these models available in a fully extended `Spice3` library which will be no more open source.

The transformation of further semiconductor models which are offered e.g. by HSPICE depends on the availability of the model source code. Usually the source code of models is not available in simulation tools but e.g. in publications, which need not describe identical to implementation model code. If model equations are available, and models are asked for further models can be added to the fully extended library. A challenge not discussed yet is the modeling of up-to-date MOSFET models like e.g. the EKV [12] model.

In accordance to the extended `Spice3` library the netlist translator has to be adapted. If net-

lists from other SPICE simulators shall be translated which are not identical to SPICE3 netlists a thoroughly check and reprogramming of the translator will be necessary.

5 Conclusions

The MSL Electrical.Spice3 was presented which has been available since October 2010. It contains most of the basic SPICE3 models as well as low level semiconductor device models. Transistor characteristics are intensively tested at small circuits with varying parameters.

The next development step presented will add coupled inductors, further semiconductor models as well as a netlist translator which completes the library to support given SPICE3 netlists without any intermediate composition step.

It is planned to create a commercial Spice3 electronics library which offers the full SPICE3 model pool as well as models of other SPICE derivatives, furthermore additional dedicated device models.

Additional to the models a method was developed of transformation existing model descriptions, which are sequentially programmed using e.g. the C or C++ language, to Modelica. This knowledge could be applied to the transformation of further open source libraries.

Open issues in the development of electronics simulation using SPICE3 features in Modelica are

- Testing extended complex circuits
- Finding a solution of further methods of analysis (AC, sensitivity)
- Improvements to performance as well as numerical stability (e.g. homotopy operator)
- Adding the conditional heat port
- Handling of some SPICE3 options and similar issues like the initial conditions and global parameter settings

The Modelica Spice3 libraries allow more advanced electronic simulation than the Modelica.Electrical.Analog library, nevertheless electronics simulation with Modelica is to be regarded as promising, but at its initial stage.

Acknowledgement

This research was founded by the European ITEA2, research projects EUROSYSLIB and MODELISAR.

References

- [1] Urquia, A.; Martin, C.; Dormido, S.: Design of SPICELib: a Modelica Library for modeling and analysis of electric circuits. *Mathematical and Computer Modelling of Dynamical Systems*, 11(1)2005, 43-60.
- [2] Cellier, F. E.; Nebot, A.: The Modelica bond graph library. *Proc. 4th Int. Modelica Conference, Hamburg-Harburg, Germany, 1, 2005, 57-65.*
- [3] Majetta, K.; Böhme, S.; Clauß, C.; Schneider, P.: SPICE3 Modelica Library. *7th International Modelica Conference, Como, 2009.*
- [4] Nagel, L.W.: SPICE2: A computer program to simulate semiconductor circuits. *Electronic Research Laboratory Rep. No. ERL-M520, University of California, Berkeley, 1975.*
- [5] SPICE Version 3e Users Manual, 1991
- [6] Robert Heinemann: PSPICE – Einführung in die Elektronik-Simulation. Carl Hanser Verlag München, 2009. <http://www.spicelab.de/index.htm>
- [7] <http://www.synopsys.com/Tools/Verification/AMSVerification/CircuitSimulation/HSPICE/>
- [8] Leitner, T.: Entwicklung simulatorunabhängiger Modelle für Halbleiter-Bauelemente mit objektorientierten Methoden. Chemnitz, Technische Universität, Diss., 1999.
- [9] Majetta, K.: Entwicklung und prototypische Umsetzung eines Konzeptes für eine Modelica-Bibliothek von SPICE-Halbleiterbauelementen und Erarbeitung einer Teststrategie. Dresden, Berufsakademie Sachsen, Dipl., 2008.
- [10] Boneß, K.: Konzeption und prototypische Implementierung eines Übersetzers von SPICE3-Netzlisten nach Modelica basierend auf Modelica-inhärenter String-Verarbeitung. Hochschule für

Technik und Wirtschaft, Dresden, Dipl.,
2011, under preparation.

- [11] Lesk, M. E.; Schmidt, E.: LEX – A
Lexical Analyzer Generator. UNIX Pro-
grammer's Manual. 7th Ed., Vol. 2A.
Murray Hill, New Jersey: Bell Tele-
phone Laboratories, Inc. 1978.
- [12] Enz, C.; Krummenacher, F.; Vittoz,
E.A.: 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
8(1995)83-114.