Proceedings
of the 3rd International Modelica Conference,
Linköping, November 3-4, 2003,
Peter Fritzson (editor)

Carla Martin, Alfonso Urquia and Sebastian Dormido
Department of Computer Science and Automatic Control, UNED, Spain:
SPICELib - Modeling and Analysis of Electric Circuits with Modelica
pp. 161-170

Paper presented at the 3rd International Modelica Conference, November 3-4, 2003, Linköpings Universitet, Linköping, Sweden, organized by The Modelica Association and Institutionen för datavetenskap, Linköpings universitet

All papers of this conference can be downloaded from
http://www.Modelica.org/Conference2003/papers.shtml

*SPICE*Lib - Modeling & Analysis of Electric Circuits with Modelica

Carla Martin, Alfonso Urquia and Sebastian Dormido
Department of Computer Science and Automatic Control, UNED, Madrid (Spain)
{carla,aurquia,sdormido}@dia.uned.es

Abstract

*SPICE*Lib is a set of model libraries, written in Modelica language, that supports some of the modeling and analysis capabilities of the circuit simulator PSpice. An upgraded version of *SPICE*Lib will be released in the near future: version 1.1. The Modelica code and the documentation of *SPICE*Lib 1.1 will be freely available. The purpose of this contribution is to discuss the capabilities supported by *SPICE*Lib version 1.1, and to illustrate its use by means of two examples.

1 Introduction

*SPICE*Lib is a set of model libraries developed at the Computer Science and Automatic Control Department of UNED for academic purposes. Programming of *SPICE*Lib version 1.0 was completed in January, 2002. It supported OP, AC sweep and TRAN analyses for the following analog device types: linear resistor and capacitor, independent voltage and current sources and NMOSFET (LEVEL1 model).

SPICE models "translation" into Modelica is performed as faithfully as possible. *SPICE*Lib device models are the PSpice models described in [6][4]. In addition, *SPICE*Lib names for device parameters and part types are equal to the PSpice ones [6], so that people used to PSpice models are not required to make an additional model-understanding effort. The procedure used to validate *SPICE*Lib is by comparing the analysis results obtained with *SPICE*Lib with the results obtained using OrCAD PSpice version 9.1 [6].

Two procedures were supported by *SPICE*Lib 1.0 for setting the initial conditions [7]: IC1 and IC2 pseudo-components, and the capacitor IC property. IC1 is a one-pin symbol that allows setting the initial voltage on a circuit node. IC2 is a two-pin symbol that allows setting the initial voltage between two nodes. Capacitor IC property allows setting the initial voltage-drop across the capacitor.

Bias point calculation is the most problematic analysis from the numerical standpoint. PSpice [6] first tries to solve the static model of the circuit using the Newton-Raphson algorithm ("static model iteration" algorithm, abbreviated: SMI). If a solution is not found, and "GMIN stepping" is enabled, the GMIN algorithm is applied . If it also fails or it is not enabled, then "static model ramping" algorithm (abbreviated: SMR) is applied. *SPICE*Lib version 1.0 implemented four algorithms for solving the circuit static model [7]: the three algorithms supported by PSpice [6][3][4] and, in addition, the algorithm "dynamic model ramping" (abbreviated: DMR), proposed in [1]. *SPICE*Lib allows the user to choose among these four algorithms in order to [7] perform the OP analysis, calculate the operating point prior to the AC sweep analysis, and evaluate the steady-state initial condition of the transient analysis.

*SPICE*Lib 1.0 implemented the two transient-analysis initialization methods supported by PSpice [6][7]: with and without bias point calculation. If bias point calculation is not skipped, prior to performing the transient analysis, the bias point of the circuit is computed, with the independent sources set to their respective transient-analysis starting values. The calculated steady-state is the initial condition for the transient analysis. On the other hand, if the bias point calculation is skipped, the bias conditions are fully determined by the IC specifications for capacitors.

An upgraded version of *SPICE*Lib will be released in the near future: version 1.1. The purpose of this contribution is to discuss the capabilities supported by *SPICE*Lib version 1.1, and to illustrate its use by means of two examples.

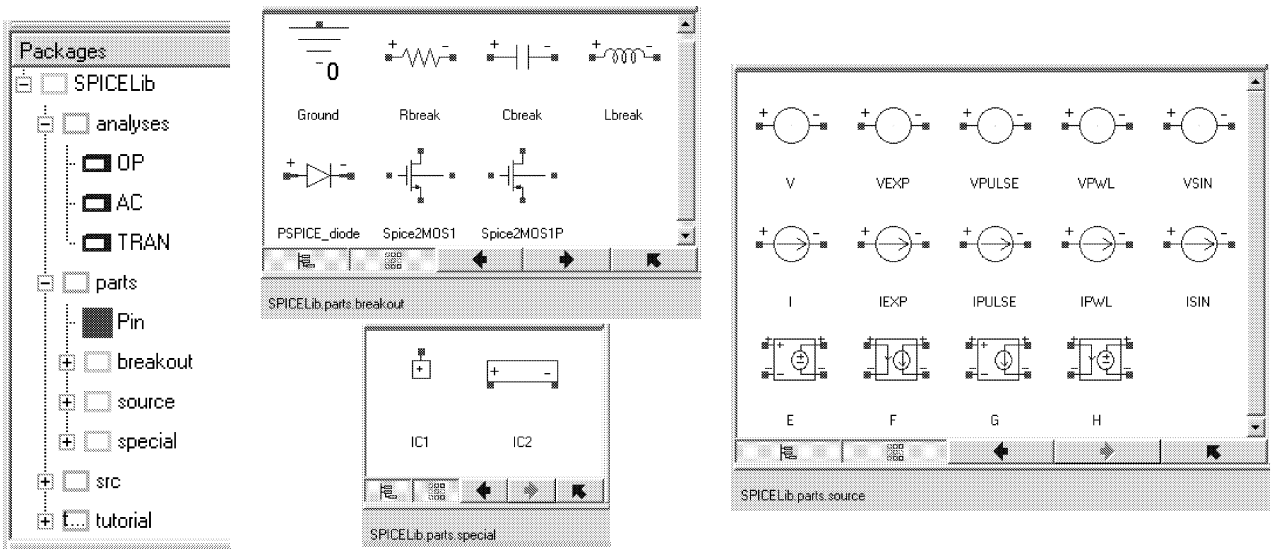

Figure 1: *SPICE*Lib 1.1 libraries. Device models in *SPICELib.parts* library.

2 New features in *SPICE*Lib 1.1

The model libraries have been reorganized, in order to facilitate their use and maintenance (see Fig. 1). A clear separation is made between those libraries to be used by *SPICE*Lib users and those libraries to be used only by *SPICE*Lib designers.

- *SPICE*Lib models describing the structure and behavior of parts and analyses are gathered in the *src* library. This library is intended to be used and modified only by *SPICE*Lib designers. Its documentation is oriented to the designers, explaining implementation details without interest to *SPICE*Lib users.
- Two libraries are defined, *parts* and *analyses*, in order to gather the "final" models that *SPICE*Lib users need to compose or analyze their circuits. These two libraries are a "mirror" of a reduced set of *src* library models, inheriting the structure and the behavior, and adding the class documentation oriented to the library user.
- *tutorial* library has been added (see Fig. 1). It contains some examples discussing how to use the *parts*-library models to compose the circuit schematic, and how to use the *analyses*-library models to analyze this previously defined circuit.

*SPICE*Lib capacitor and NMOS models [7] have been modified, in order to fix some numerical problems and improve the simulation speed of the transient analysis. This point will be discussed in Sections 3.2 and 3.4. In addition, the following device models have been included in the new version of the library (see Fig. 1):

- Linear inductor: *Lbreak* model. It supports the inductor IC property [6]: current through the inductor during the bias point calculation can be set.
- Controlled linear sources (E, F, G, H) [6].
- PSpice diode [4] and SPICE2 LEVEL1 p-channel MOSFET [4].

*SPICE*Lib 1.0 analysis models have not been modified. All the discussions in [7] concerning analyses are valid for *SPICE*Lib 1.1, and therefore they will not be reproduced in this manuscript.

3 Devices models

*SPICE*Lib device models are composed of three different device descriptions: static, AC small-signal and large-signal. Each of these three descriptions has its own set of equations and variables, and its own contribution to the device-model interface. As a consequence, a circuit model built using *SPICE*Lib library contains the static, the large-signal and the AC small-signal descriptions of the circuit [7].

*SPICE*Lib analysis models describe the sequence of control-signal value transitions required to perform the analyses. Control-signals are *inner* variables [5] of the analysis models and *outer* variables of the device models. Analysis models inherit the circuit model, which is composed of device models. *SPICE*Lib control-signals trigger instantaneous changes in the

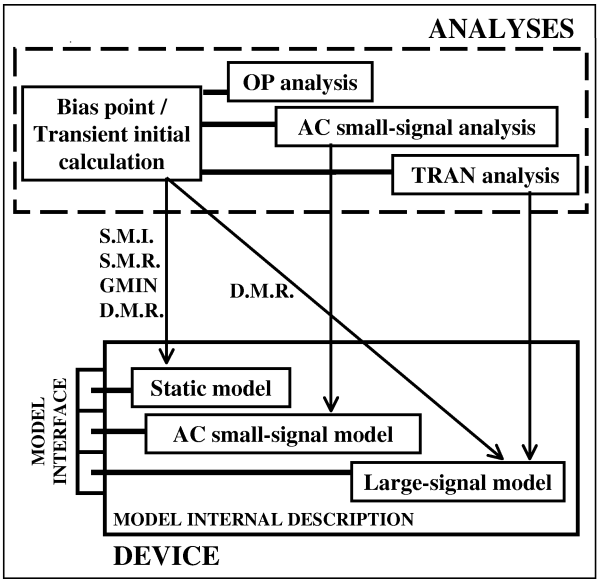

Figure 2: Structure of *SPICE*Lib device descriptions and analyses.

device model state-variables and in the mathematical structure of the device descriptions. These structural changes are carried out by opening and closing the switches internal to the device descriptions [7]. These changes facilitate setting the analysis initial conditions. Also, they facilitate the exchange of information among the device descriptions, required when an analysis use several device descriptions (see Fig. 2) [7]:

- *SPICE*Lib performs an OP analysis before the AC sweep analysis in initiated, in order to calculate the bias-point. Then, *SPICE*Lib calculates the AC model of the circuit by substituting, in the small-signal (i.e., linearized) model of each component, the bias-point current and voltage values.
- If *SKIPBP* parameter is set to false, then *SPICE*Lib carries out an OP analysis prior to the transient analysis, and the calculated bias-point is the initial condition for the transient analysis.
- "Dynamic model ramping" algorithm requires the combined use of the static and large-signal device descriptions.

In addition, control-signals "disable" the model descriptions once they are no more useful during the analysis [7]. As a consequence, the simulation computational effort is not unnecessarily increased.

Relevant aspects of the new models supported by *SPICE*Lib 1.1, and the modifications made on the capacitor and NMOS models, are discussed next. In particular, the large-signal and the static descriptions of the controlled sources, capacitor, inductor, PN-junction diode and MOS transistor will be discussed. The reader interested in all the mathematical details of the descriptions is referred to the documentation included in the *SPICELib.parts* library.

3.1 Controlled linear sources

*Spice*Lib 1.1 supports the following four types of controlled linear sources (see Fig. 1):

- Voltage-controlled voltage source (E).
- Current-controlled current source (F).
- Voltage-controlled current source (G).
- Current-controlled voltage source (H).

*SPICE*Lib controlled sources are devices with four pins: two controlling pins,p1 (+) and n1 (-), placed at the icon left side; and two output pins, p2 (+) and n2 (-), placed at the icon right side. In the three model formulations (i.e., static, AC small-signal and large signal [7]), the source output current or voltage is proportional to its controlling current or voltage. The proportionality constant is a model parameter: *Gain*.

3.2 Capacitor and inductor

*SPICE*Lib 1.1 inductor and capacitor models are analogous. The partial models *Capacitor* and *Inductor* are defined in the library *SPICELib.src.BREAKOUT*. They describe the static, large-signal and AC small-signal behavior of capacitors and inductors respectively, except their large-signal and AC small-signal capacitance and inductance. The linear capacitor (*Cbreak*) and the capacitors internal to the PN-junction diode and MOS transistor models are sub-classes of *Capacitor*. The linear inductor (*Lbreak*) is a sub-class of *Inductor*.

*SPICE*Lib 1.0 linear capacitor model (*Cbreak*) has been modified in the new version of the library: an small-value resistor has been included (by-default value: R_EPS2 $= 2 \cdot 10^{-4}$ohm) . See Fig. 3, in comparison with Figs. 3 and 4 in reference [7]. Analogously, R_BIG2 resistor has been included in the linear inductor model, *Lbreak* (see Fig. 3).

The reason behind this model modification is to avoid that the circuit model index is greater than one. Models with an index greater than one,

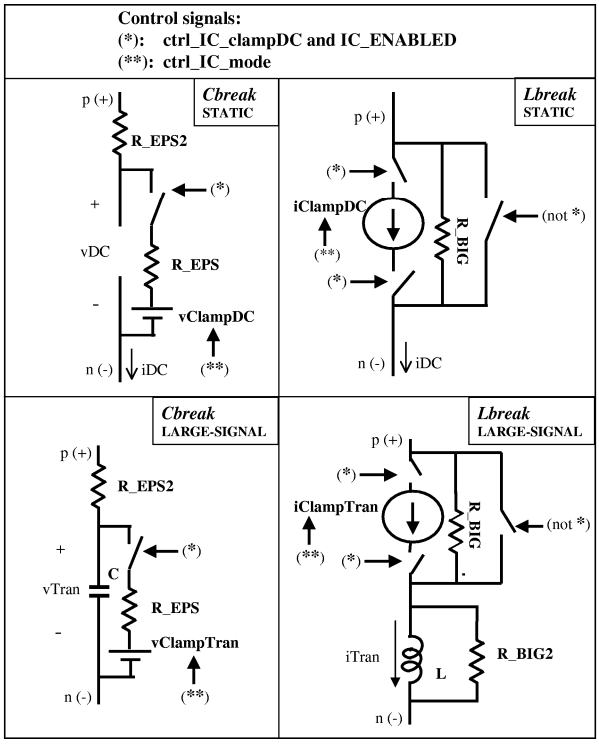

Figure 3: Static and large-signal descriptions of *Cbreak* and *Lbreak*.

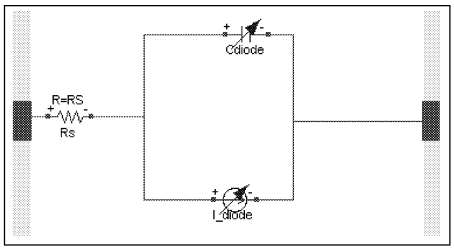

Figure 4: Diagram of *SPICE*Lib diode.

due to the parallel connection of *Capacitor* sub-classes, must be avoided, because index reduction involves differentiating discontinuous equations. In that situation, Dymola [2] shows an error message during the simulation run when the differentiated-function discontinuity is detected.

The static and large signal descriptions of *Cbreak* and *Lbreak* devices are shown in Fig. 3. The purpose of the control signals *ctrl_IC_clampDC* and *ctrl_IC_mode*, and the parameter *IC_ENABLED* is discussed in [7]. Applying an IC property for a capacitor has the same effect as applying one of the pseudocomponents IC2 across its nodes [6]. *SPICE*Lib attaches (closing the switch) a voltage source (*vClampDC* and *vClampTran*) with a 0.0002 ohm series resistance (*R_EPS*) in parallel with the capacitor. In the case of the initial current through the *Lbreak* inductor, the internal implementation is analogous to the capacitor *Cbreak*. *SPICE*Lib attaches a current source (*iClampDC* and *iClampTran*) with a 1 Gohm parallel resistance (*R_BIG*) in series with the inductor (see Fig. 3).

3.3 PN-junction diode

The PN-junction diode model in *SPICE*Lib 1.1 is the PSpice model. This is based on SPICE2 diode model, with new features for modeling the recombination and high-level effects [4]. *SPICE*Lib diode model, *PSPICE_diode*, is composed of the following devices (see Fig. 4): a linear resistor (of *Rbreak* class [7]), a voltage-dependent capacitor, and a voltage-controlled, non-linear current source.

3.3.1 Voltage-dependent capacitor

It extends the partial model *Capacitor*, defining the large-signal and the AC small-signal capacitance (as described in [4]). The mathematical relationship between the large-signal capacitance of the diode, C, and the large-signal voltage drop across the capacitor, v, is a two-branches function, that can be conveniently described using the Modelica expression *if-then-else*:

$$C = \begin{cases} C_d(v) + C_j(0) \cdot \left(1 - \frac{v}{\phi_0}\right)^{-m} & \text{for } v < FC \cdot \phi_0 \\ C_d(v) + \frac{C_j(0)}{F_2} \cdot \left(F_3 + \frac{m \cdot v}{\phi_0}\right) & \text{for } v \geq FC \cdot \phi_0 \end{cases} \quad (1)$$

where $C_d(v)$ is a continuous, highly non-linear function of v (sum of several exponential terms); and $C_j(0)$, ϕ_0, m, FC, F_2 and F_3 are model parameters. By-default value for parameters m and FC is 0.5; and by-default value for ϕ_0 is 1 volt [4]. In order to illustrate the shape of the C-V characteristic described in Eq. (1), the C-V curve of the D1N4002 diode is represented in Fig. 5.

The following consideration is important from the numerical standpoint. Acceptable values for FC guarantee that if $v < FC \cdot \phi_0$, then the term $\left(1 - \frac{v}{\phi_0}\right)$ is greater than zero. However, numerical problems are experienced (i.e., square root of a negative number) unless the *if-then-else* condition is surrounded by *noEvent*, as explained in [2][5].

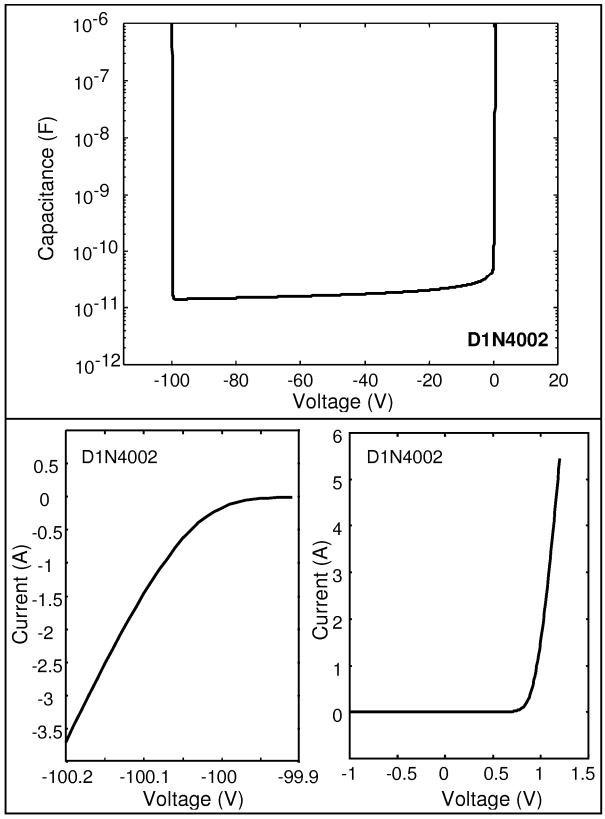

Figure 5: C-V and I-V characteristics of D1N4002.

3.3.2 Voltage-controlled current source

The static and large-signal constitutive relation of the source is (see I-V characteristic of diode D1N4002 in Fig. 5) [4]:

$$\begin{aligned} i_d &= K_{hli}(v)\cdot I_S\cdot\left(e^{\frac{q\cdot v}{n\cdot k\cdot T}}-1\right) \\ &+ K_{gen}(v)\cdot I_{SR}\cdot\left(e^{\frac{q\cdot v}{n_R\cdot k\cdot T}}-1\right) \\ &- IBV\cdot e^{-\frac{q\cdot(BV+v)}{k\cdot T}} \end{aligned} \quad (2)$$

$$K_{hli}(v) = \begin{cases} \sqrt{\dfrac{I_{KF}}{I_{KF}+I_S\cdot\left(e^{\frac{q\cdot v}{n\cdot K\cdot T}}-1\right)}} & \text{for } I_{KF}>0 \\ 1 & \text{otherwise} \end{cases} \quad (3)$$

$$K_{gen}(v) = \sqrt{\left[\left(1-\frac{v}{\phi_0}\right)^2+0.005\right]^m} \quad (4)$$

Transient analysis. R_S resistor guarantees that the capacitor voltage can always be a state variable of the large-signal model (see Fig. 4). As a consequence, the large-signal constitutive relation of the current source is used, during the transient analysis, to calculate the current generated by the source.

The numerical efficient of the *SPICE*Lib transient analysis has been significantly improved by including in the diode model an equation setting that the voltage drop across the capacitor (state variable) is equal to the voltage drop across the current source. The large-signal constitutive relation of the current source is formulated in terms of this voltage drop, directly calculated from a state variable, instead of in terms of the difference between the voltage at the source pins. Although mathematically these two modeling approaches are equivalent, they are not from the numerical standpoint. In the first case, the current generated by the source is directly calculated from a state variable. In the second case, it is necessary to calculate the voltage values at the two source pins in order to obtain the current generated by the source. This second approach commonly leads to non-linear algebraic loops.

DC analysis. As the static model of *Cdiode* capacitor is an open circuit, the computational causality of the source constitutive relation is not known in advance. In fact, numerical problems can be experienced when the voltage drop, v, needs to be calculated from Eq. (2). The current-source conductance $\frac{di_d}{dv}$ falls to zero in the reverse-bias region of operation (see Fig. 5): the current is constant and no longer a function of the diode voltage. Often, very small conductance values force the Newton-Raphson algorithm to significantly overshoot the correct solution voltage, and a large number of algorithm iterations are required to work back to the correct solution voltage. Worse yet, it the conductance value ever reaches zero, the next Newton iteration will cause a floating divide-by-zero error.

This problem is solved in SPICE by placing a shunt resistor in parallel with the PN-junction [3]. This resistor has a default conductance value of $GMIN = 10^{-12}$mhos. The GMIN resistor is built into the source model, and always provides a small voltage-dependent current for the devices. The constitutive relation of the current source, with the GMIN resistor term included, is the following: $i = i_d + v\cdot GMIN$. Consequently, the conductance under reverse-bias conditions is $\frac{di}{dv} = 0 + GMIN$.

To help avoid nonconvergence problems, *GMIN* should be set as large as possible without affecting the accuracy of the simulation output. The larger the value of *GMIN*, the faster the Newton-Raphson algorithm will converge on a solution. *GMIN* is a global parameter for the entire *SPICE*Lib circuit, so when selecting the value of *GMIN*, the most current

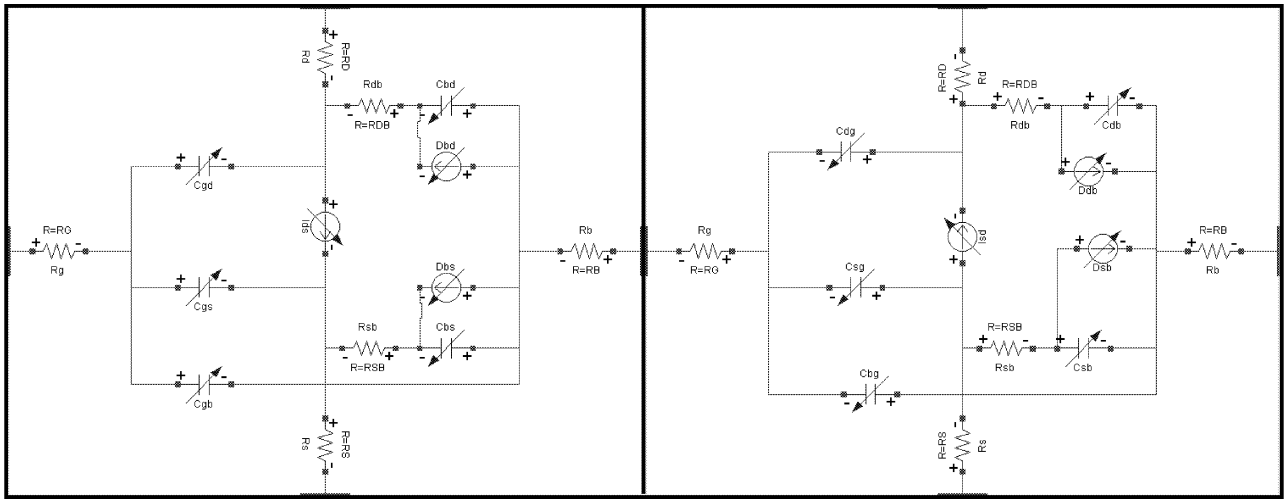

Figure 6: *SPICE*Lib NMOS (left) and PMOS (right) model diagrams.

sensitive parts of the circuit should be considered. The following recommendation is found in [3]: to set *GMIN*, determine the smallest parasitic resistance value (R_p) which could be placed across any two nodes without influencing the behavior of the circuit. Set $GMIN = \frac{1}{R_p}$.

"GMIN stepping" algorithm attempts to find a solution for the static model by starting with a large value of *GMIN*, initially 10^{10} times the nominal value. If a solution is found at this setting, *GMIN* is reduced by a factor of 10 and a new solution is found. This continues until *GMIN* is back to the nominal value, or the repeated cycle fails to converge [6][3]. *SPICE*Lib implementation of "GMIN stepping" algorithm is described in [7]. The *scaleGMIN* variable is included in the static constitutive relation of the *SPICE*Lib current source: $i = i_d + v \cdot GMIN \cdot scaleGMIN$.

Just as the Newton-Raphson algorithm has problems with very small values of conductance, very large values of conductance can lead to problems also [3]. Under these conditions, the next voltage iteration (v_{n+1}) will be only slightly different than v_n, and if the device is biased far from the proper solution voltage, many iterations will be required to work back to the solution. Conductance values become unrealistically large when PN-junctions have a forward bias of more than 0.8 volts (see Eq. (2) and Fig. 5). If a high forward bias is applied to the diode during the iterative process, then the Newton-Raphson algorithm may fail to converge. SPICE (and *SPICE*Lib) addresses this problem including a series resistance in the diode model (see R_S in Fig. 4) [3]. Under high forward-bias conditions, the series resistance dominates the device conductance and helps reduce the occurrence of nonconvergence.

3.4 MOS transistors

PSpice provides six MOSFET device models, which differ in the formulation of the I-V characteristic [6]. *SPICE*Lib implements LEVEL1 model [4]. The *SPICE*Lib model diagrams of the NMOSFET and PMOSFET devices are shown in Fig. 6. They are modeled as intrinsic MOSFETs using ohmic resistances in series with the drain (R_g), source (R_s), gate (R_g) and bulk (R_b). The model of the two substrate junctions is composed of a linear resistor, a voltage-dependent capacitor and a voltage-controlled, non-linear current source (see Fig. 6). For the sake of conciseness, only NMOSFET model will be discussed in this manuscript: PMOSFET model is completely analogous (all the details can be found in *SPICE*Lib library documentation).

The drain-to-source current (I_{DS}) and the gate capacitances (C_{GB}, C_{GS} and C_{GD}) are a function of the variables V_{DS}, V_{GS} and V_{BS}. However, V_{GS} and V_{BS} cannot be calculated from the terminal variables of the I_{DS} source model (i.e., the voltage and current at their pins). Gate capacitors modeling presents a similar problem. This situation is solved in *SPICE*Lib defining the variables V_{DS}, V_{GS} and V_{BS} as *inner* variables [5] of the transistor model, and as *outer* variables of its components: the I_{DS} current-source and the gate capacitors. These variables are defined as

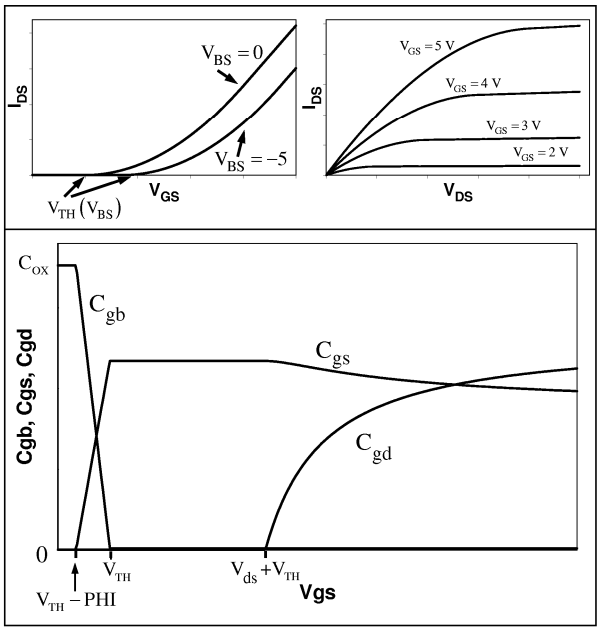

Figure 7: Transfer and output characteristics (above), and gate capacitance (below) of LEVEL1 MOS implemented in *SPICE*Lib. Axes scaling is linear.

follows: $V_{DS} = \text{noEvent}\left(\text{abs}\left(\text{C}_{\text{GS}}.v - \text{C}_{\text{GD}}.v\right)\right)$, $V_{GS} = \max\left(\text{C}_{\text{GS}}.v, \text{C}_{\text{GD}}.v\right)$, and $V_{BS} = \max\left(\text{C}_{\text{BS}}.v, \text{C}_{\text{BD}}.v\right)$; where C.$v$ represents the voltage drop cross the capacitor C. As all terms C.v are state-variables of the large-signal MOSFET model, these definitions tear the computational-causality loops of the MOSFET large-signal description.

Drain-to-source current is described by Eq. (5) when the transistor is in the linear region and by Eq. (6) when it is in the saturation region. Equations (5) and (6) are valid for $V_{DS} > 0$ (normal mode). For $V_{DS} < 0$ (inverted mode), *SPICE*Lib switches the source and drain in the above equations.

$$I_{DS} = \frac{KP \cdot W}{L - 2 \cdot X_{jl}} \cdot \left(V_{GS} - V_{TH} - \frac{V_{DS}}{2}\right) \cdot V_{DS} \cdot (1 + \lambda \cdot V_{DS}) \quad (5)$$

$$I_{DS} = \frac{KP \cdot W}{2 \cdot \left(L - 2 \cdot X_{jl}\right)} \cdot (V_{GS} - V_{TH})^2 \cdot (1 + \lambda \cdot V_{DS}) \quad (6)$$

Gate capacitance models used by *SPICE*Lib are represented in Fig. 7 (as described in [4]). The three gate capacitances (i.e., C_{GB}, C_{GS} and C_{DS}) are nonlinear, continuous functions of V_{GS}. In addition, C_{GS} and C_{DS} are functions of V_{DS} when the transistor operates in the linear region (i.e., when $V_{GS} > V_{TH} + V_{DS}$). For $V_{DS} < 0$ (inverted mode), *SPICE*Lib switches the source and drain in the capacitance calculations. The transition between C_{GS} and C_{DS} at $V_{DS} = 0$ is discontinuous in the SPICE2 model described in [4]. The model introduced in *SPICE*Lib differs from that described in [4], because a continuous link has been introduced in *SPICE*Lib between C_{GS} and C_{GD} in the vicinity of $V_{DS} = 0$.

The substrate junctions model differs slightly from the PN-junction model discussed in Section 3.3 [4]. The series resistances (see R_{db} and R_{sb} in Fig. 6) guard against extremely large values of the conductance and to avoid that the MOSFET model index is greater than one.

4 Modeling and analysis of an AC to DC quadrupler

This first example discusses the modeling of an AC to DC quadrupler (i.e., full wave rectifier in series with a low pass filter) using *SPICELib.parts* library. The results of OP, AC sweep and TRAN analyses using *SPICELib.analyses* and OrCAD PSpice are compared.

4.1 Circuit modeling

The AC to DC quadrupler circuit is composed of one independent voltage source, four diodes, one capacitor and one inductor (see Fig. 8). The steps to build the model are the following:

- To open *SPICE*Lib library, select in the Dymola window "*File/Open*". Open the file *SPICELib/package.mo*.
- Create a new package. Enter *circ1* as name of the new package. The circuit model and the analysis models will be inserted in this package.
- Create a new partial-model. It will describe the circuit schematic. Enter *Schematic* as name of the new model, and insert it in package *circ1*.
- Drag the voltage source model from *SPICELib.parts.source* library and drop it into *Schematic* model window. Drag and drop the other circuit components. Resistor, capacitor, inductor, diode and ground models can be found in *SPICELib.parts.breakout* library.
- Connect the components as shown in Fig. 8.
- Double-click on the component icons to enter the component parameters.

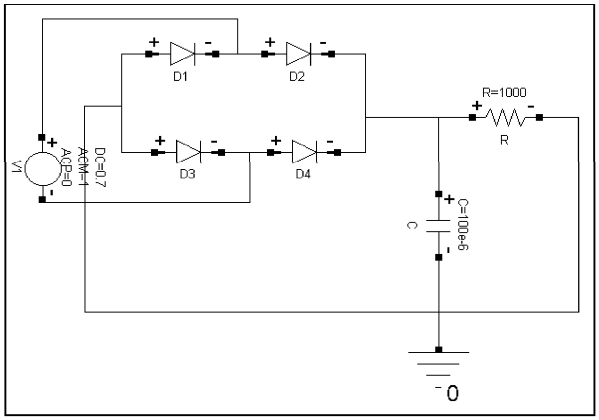

Figure 8: *SPICE*Lib model of an AC to DC quadrupler.

Device model documentation is included in *SPICE*Lib. To get documentation for a device model, place the cursor on it, press the right button and select "*Info*".

4.2 Bias point analysis

Once the circuit schematic has been described, follow these steps to perform the OP analysis:

- Create a new model, and insert it in *circ1* package. Enter *circ1_OP* as name of the new model.
- Drag *OP* model from *SPICELib.analyses* library, and drop it into *circ1_OP* model window.
- Double-click on the component icon. Enter the name of the circuit model to analyze (*Schematic*) and the value of the OP analysis parameters: *TimeScale*, *LogResults* and *SOLVE_STATIC*. The meaning of these parameters in discussed in [7]. *TimeScale* parameter value is intended for providing an approximate value of the circuit time-constant (in this example, $TimeScale = CLOCK = 0.1s$). *LogResults* value sets the amount of information to be logged during the bias point analysis. The bias-point algorithm to be used in determined by *SOLVE_STATIC* parameter value (0, 1, 2 or 3).
- Change to the Dymola window "*Simulation*" and translate *circ1_OP* model.
- Set Dymola's variable *StopTime* to a value large enough not to interfere with *SPICE*Lib analysis [7]. A valid setting is $StopTime = 10 \cdot TimeScale$. In this example $StopTime = 1s$.
- Run the simulation. *SPICE*Lib forces the simulation to end when the analysis is finished (always before *StopTime* is reached, otherwise increment *StopTime* value and repeat the analysis simulation) [7]. Once he simulation is finished, the analysis results are logged out to *dslog.txt* file.

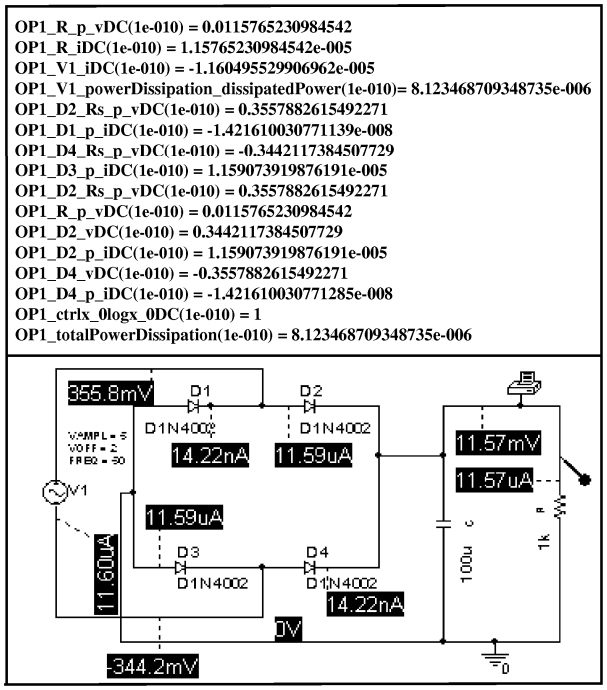

Figure 9: OP analysis results. *SPICE*Lib using SMI algorithm (above). ORCAD PSpice (below).

The results of *SPICE*Lib OP analysis using SMI algorithm are compared in Fig. 9 with the results obtained using OrCAD PSpice [6].

*SPICE*Lib OP analysis is repeated using SMR algorithm. The static-description value of the independent power supply is ramped by *SPICE*Lib from zero to its target value, during *TIME_SCALE* seconds. Then, *Spice*Lib logs the analysis results out to *dslog.txt* file. *CLOCK* seconds are elapsed before *SPICE*Lib triggers the simulation end. The evolution in time of the power supply voltage and the evolution of the voltage drop across the resistor $(\times 10)$ are shown in Fig. 10.

Continuing with the example, *SPICE*Lib GMIN algorithm is applied. The results are shown in Fig. 11. *SPICE*Lib solves the circuit static formulation starting with a large value of *GMIN*. Every *CLOCK* seconds *SPICE*Lib reduces *GMIN* by a factor of 10, until GMIN nominal value is reached. Then, *SPICE*Lib logs the voltages and currents of the circuit static formulation out to *dslog.txt* file.

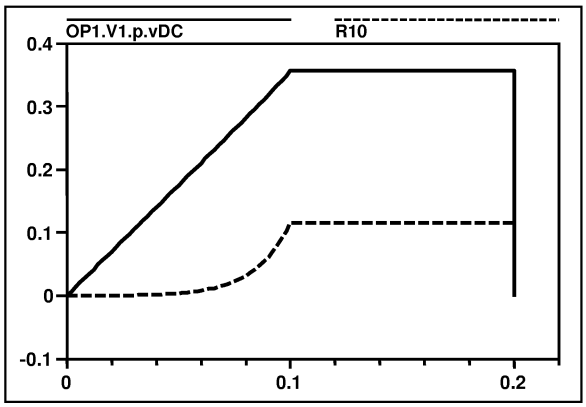

Figure 10: Power supply ramping and voltage drop across the resistor ($\times 10$).

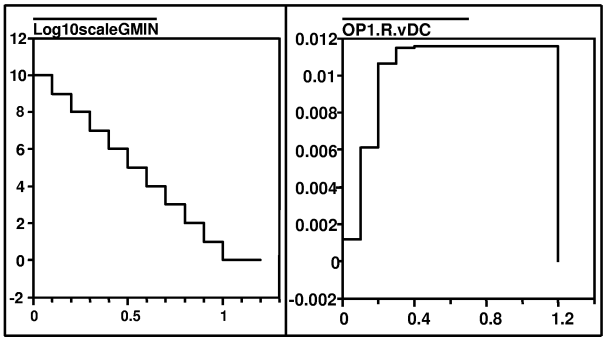

Figure 11: log10 $(scaleGMIN)$ (left). Voltage drop across the resistor (right).

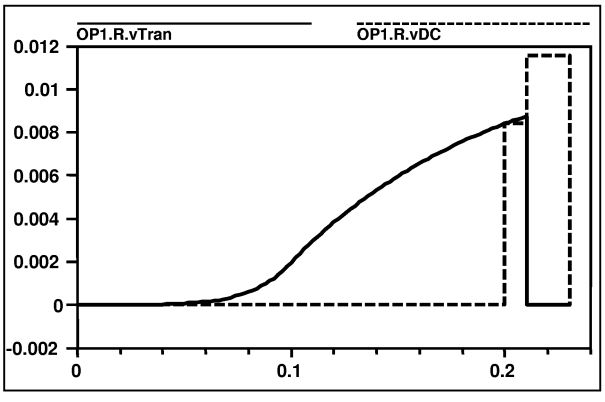

Figure 12: OP analysis using DMR algorithm.

Finally, DMR algorithm is applied (see Fig. 12). The value of *CLOCK* is set to 0.01. The initial condition to iterate the circuit static description is obtained by simulating the circuit large-signal description [7][1]. A transient analysis is performed: the power supply is ramped from zero up to its initial value during

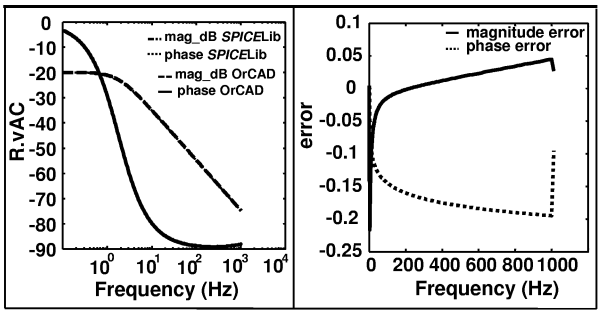

Figure 13: AC sweep analyses.

$TIME\_SCALE = 0.1s$ and this value is held for $TIME\_SCALE = 0.1s$ to allow the circuit to stabilize. Then (at $time = 0.2s$), the large-signal description voltages are transferred to the static description. This static circuit setting is held for $CLOCK = 0.01$ second (until $time = 0.21s$). Then, the circuit static formulation is solved. The OP analysis results are logged out to *dslog.txt* file at $time = 0.22s$. *CLOCK* seconds latter, *SPICE*Lib triggers the simulation end.

4.3 AC sweep and TRAN analyses

Follow these steps to perform an AC sweep analysis:

- Create a new model, and insert it in *circ1* package. Enter *circ1_AC* as name of the new model.
- Drag *AC* model from *SPICELib.analyses* library, and drop it into *circ1_AC* model window.
- Double-click on the component icon. Enter the name of the circuit model to analyze (*Schematic*) and the value of the AC analysis parameters: *TYPE_AC_SWEEP* (linear or logarithmic by decades), *POINTS_NUMBER*, *START_FREQUENCY* and *END_FREQUENCY* [7]. In addition, the parameters of the OP analysis prior to the AC sweep have to be set.

The results of the AC sweep analysis using *SPICE*Lib are compared in Fig. 13 with the results obtained using OrCAD PSpice.

The steps to perform a TRAN analysis are analogous. In this case, drag *TRAN* model instead of *AC* model. *SPICE*Lib TRAN analysis of the AC to DC quadrupler, with bias point calculation, is shown in Fig. 14, and it is compared with OrCAD PSpice analysis.

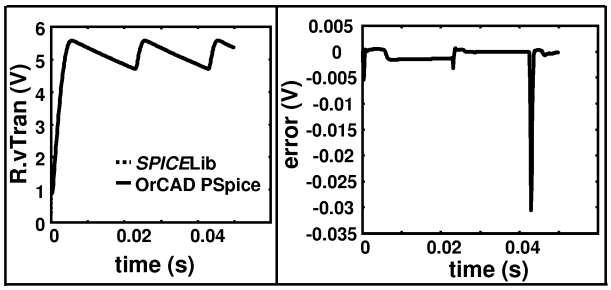

Figure 14: TRAN analyses with bias point calculation.

5 Analysis of digital circuits

The precision obtained from LEVEL1 model is limited; however, the simplicity of these equations is useful in many situations, for instance the analysis of digital circuits. NAND and NOR logical gates have been modeled using *SPICE*Lib (see Fig. 15), and the results of a transient analysis with bias point calculation have been compared with the results obtained using OrCAD PSpice (see Figs. and 16 and 17).

Conclusions

The new capabilities of *SPICE*Lib 1.1 have been discussed, and illustrated by means of two examples.

References

[1] Cellier, F. E.: *Continuous System Modeling*. Springer-Verlag. 1991.

[2] Elmqvist, H. et al.: *Dymola. User's Manual.* Dynasim AB. Version 5.0a, 2002.

[3] Kielkowski, R. M.: *Inside SPICE.* McGraw-Hill. 2nd edition, 1998.

[4] Massobrio, G. and Antognetti P.: *Semiconductor Device Modeling with SPICE.* McGraw-Hill. 1993.

[5] Modelica Association: *Modelica. Language Specification.* Version 2.0, July 10, 2002.

[6] OrCAD, Inc.: *OrCAD PSpice A/D. Reference Guide & User's Guide.* OrCAD, Inc. 1999.

[7] Urquia, A. and Dormido, S.: *DC, AC Small-Signal and Transient Analysis of Level 1 N-Channel MOSFET with Modelica.* In proc: 2nd Modelica Conf. pp. 99-108.

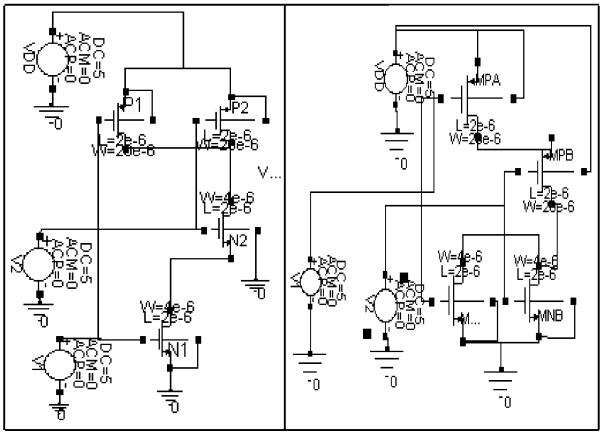

Figure 15: *SPICE*Lib NAND and NOR models.

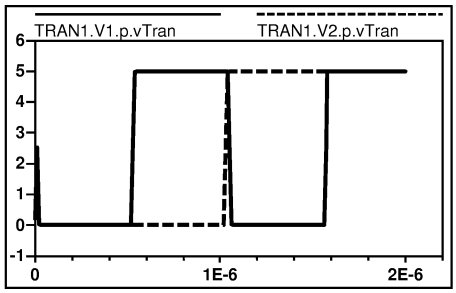

Figure 16: Input stimuli (equal for both gates).

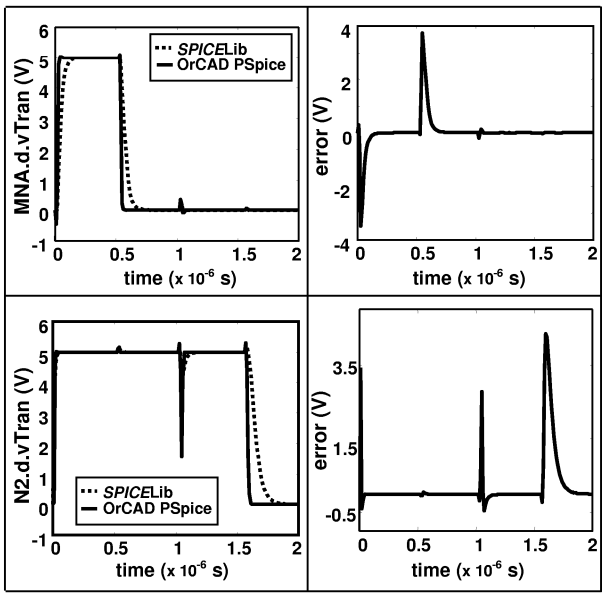

Figure 17: NOR and NAND output voltages.