GNU Simulators Supporting Verilog-A Compact Model Standardization

Stefan Jahn[1], Mike Brinson[2], Michael Margraf[3], Hélène Parruitte[4], Bertrand Ardouin[5], Paolo Nenzi[6] and Laurent Lemaitre[7]

Qucs: Quite Universal Circuit Simulator

Qucs is a GPL integrated Circuit Simulator developed by an international group of scientists and engineers [1]

- Qucs has a graphical user interface (GUI), based on Qt® by Trolltech [3]
- simulation engine Qucsator
- Currently supported circuit analysis types are DC, AC, AC noise, S-parameter, S-parameter noise and Transient
- Qucs GUI supports schematic capture, analysis control and simulation data post-processing using equations
- SPICE netlists can be read and simulated by Qucs using conversion program Qucsconv
- Qucs uses FreeHDL [4] and Icarus Verilog [5] for VHDL and Verilog digital simulation
- Qucs links to ASCO [8] - A SPICE Circuit Optimizer
- ADMS is interfaced to Qucs allowing Verilog-A compact device modelling
- Qucs includes a device model and subcircuit library manager
- Qucs is equipped with stand alone software tools for filter synthesis, attenuator and transmission line design
- Qucs also provides interactive routines for the design of 1- and 2-port power and noise matching circuits
- Qucs is multilingual, currently it has been translated into thirteen different languages
- Qucs is available for the GNU/Linux, FreeBSD, Solaris, MacOS, Windows and Cygwin operating systems

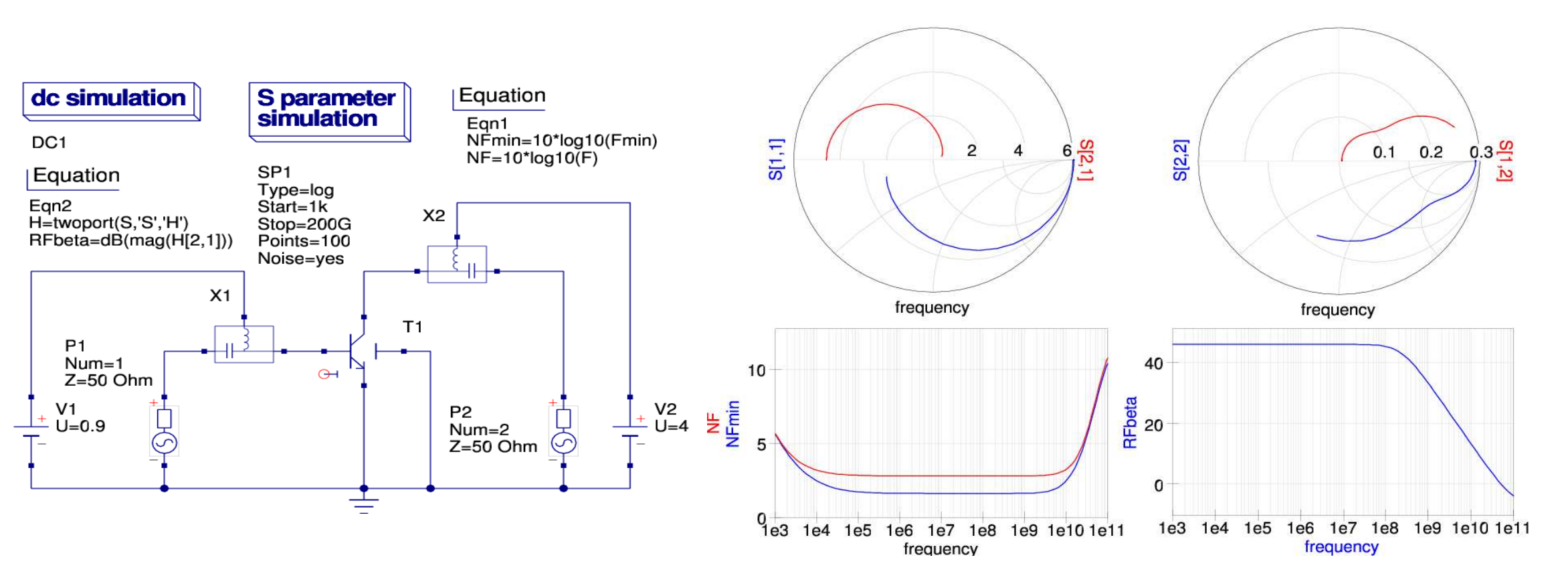

Fig. 1: Test schematic and the S-parameter plots for the Qucs implementation of the HiCUM/L2 V2.11 device model

The Qucs XML interface

- The ADMS software translates Verilog-A device models into a structured XML tree Qucs/ADMS uses the XML tree to generate ready_to_compile C/C++ code
- The generated C/C++ code is specific to the Qucs API
- The process of transforming Verilog-A model code into C/C++ code is performed by the command line script:

```
$ admsXml <device.va> -e <interface-1.xml> -e <interface-2.xml>
```

Qucs release 0.0.11 is distributed with the following ADMS XML interface transformation scripts :

- qucsMODULEcore.xml : this creates the simulator C/C++ code for a Verilog-A device model
- qucsMODULEdefs.xml : this creates device parameter descriptions
- qucsMODULEgui.xml : this generates a model GUI interface
- qucsVersion.xml : this is a basic library
- analoguefunction.xml : this is used to create analogue function code

A detailed description of the steps needed to translate a Verilog-A model into C/C++ code and link it to the Qucs API are documented [2].

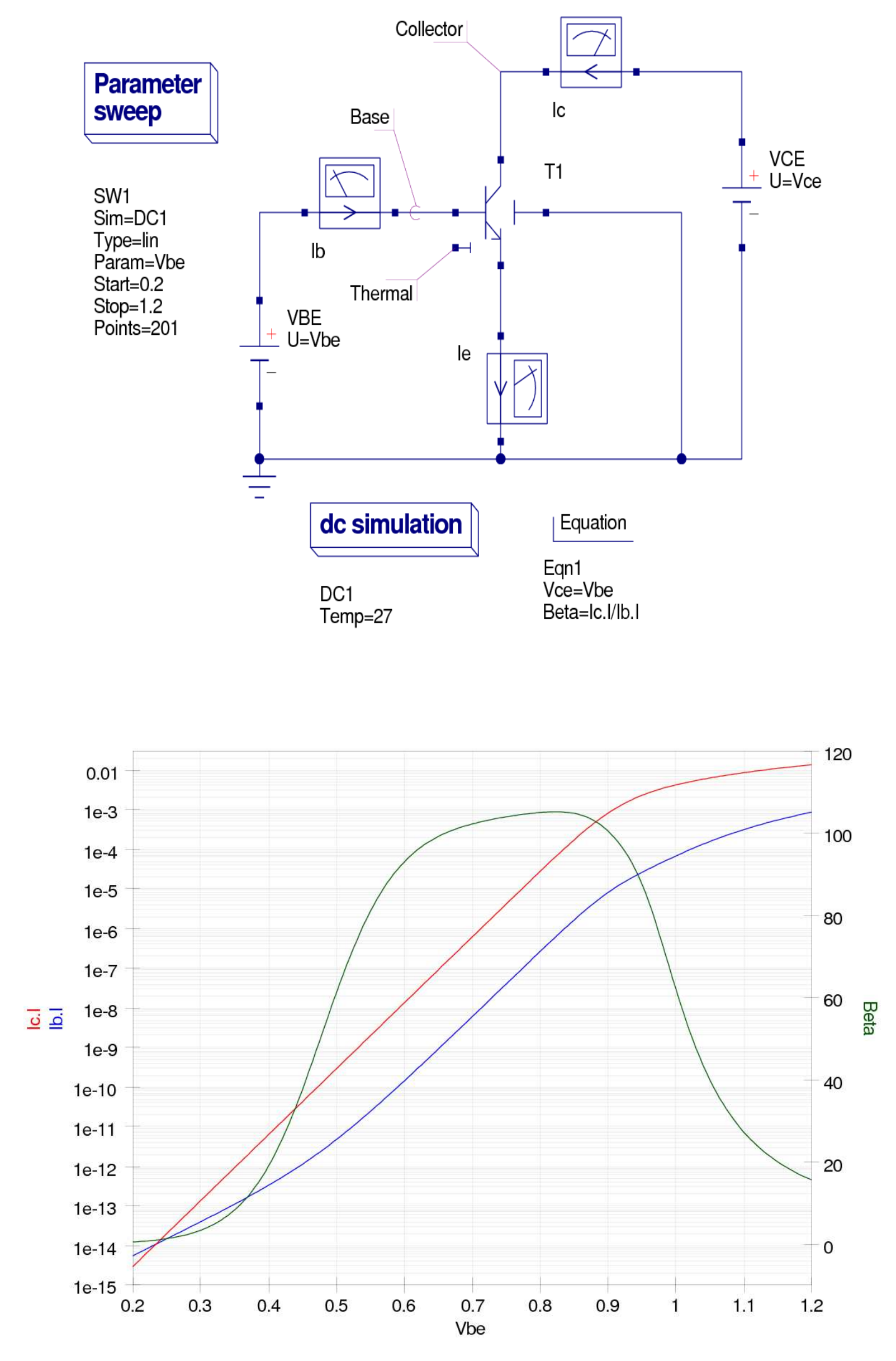

Fig. 2: Test schematic and output plots for the Qucs implementation of the HiCUM/L2 V2.11 device model

ADMS: Verilog-A Compact Model Generator

ADMS is an open-source software tool [9], that supports and simplifies compact model development, implementation, distribution, maintenance, and sharing. adms is a code generator that converts electrical compact device models specified in high-level description language into ready-to-compile c-code for the API of spice simulators. Based on transformations specified in xml language adms transforms Verilog-AMS code into other target languages. The intended audience is people in universities or CAD companies interested in contributing to the development and improvement of the tool. It should also help people whose main interest is using the tool to get a better understanding of how ADMS works internally. ADMS is available on SourceForge.net, one of the most popular open source software development web sites.

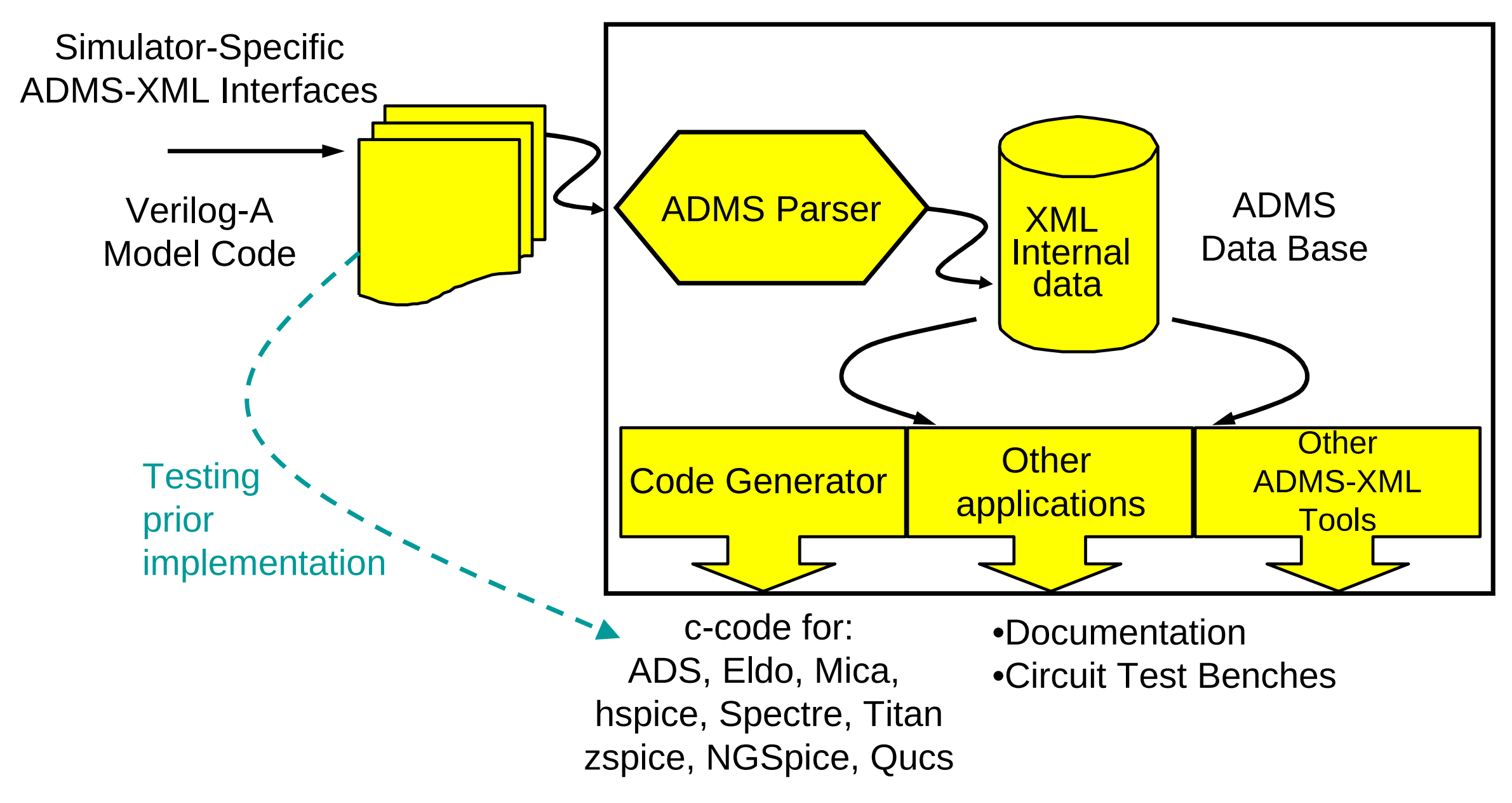

Fig. 3: ADMS data flow diagram

HiCUM Compact Model

HICUM is a semi-physical compact bipolar transistor model [10]. Semi-physical means that for arbitrary transistor configurations, defined by emitter size as well as number and location of base, emitter and collector fingers (or contacts, respectively), a complete set of model parameters can be calculated from a single set of technology specific electrical and technological data. The name HICUM was derived from *HI*gh-*CU*rrent *M*odel, indicating that HICUM initially was developed with special emphasis on modeling the operating region at high current densities which is very important for certain high-speed applications.

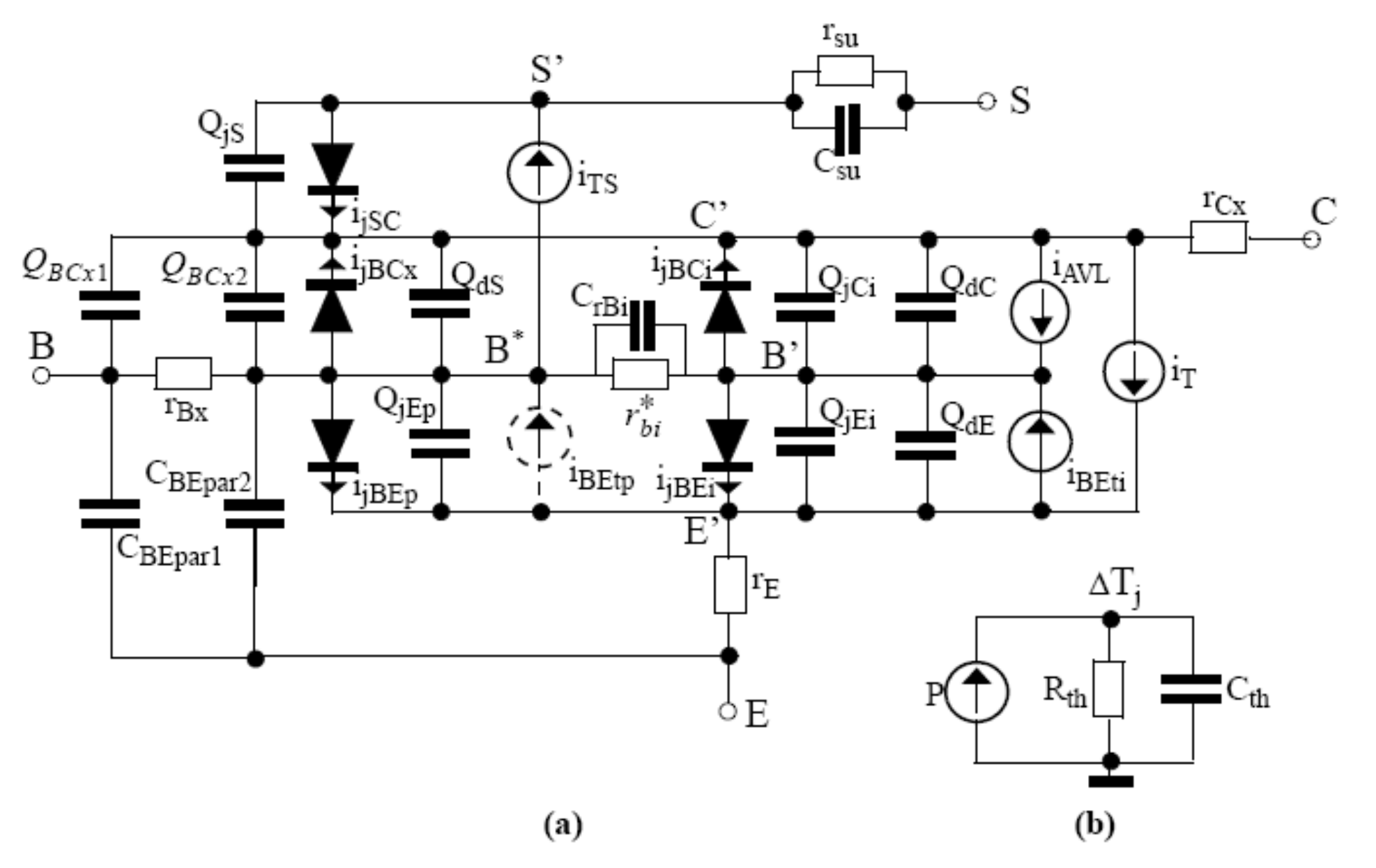

Fig. 4: (a) Large-signal HICUM/Level2 equivalent circuit.
(b) Thermal network used for self-heating calculation.

The important physical and electrical effects taken into account by HICUM/L2 are briefly summarized below:

- high-current effects (incl. quasi-saturation)
- distributed high-frequency model for the external base-collector region
- emitter periphery injection and associated charge storage
- emitter current crowding (through a bias dependent internal base resistance)
- two- and three-dimensional collector current spreading
- parasitic (bias independent) capacitances between base-emitter and base-collector terminal
- vertical non-quasi-static (NQS) effects for transfer current *and* minority charge
- temperature dependence and self-heating
- weak avalanche breakdown at the base-collector junction
- tunneling in the base-emitter junction
- parasitic substrate transistor
- bandgap differences (occurring in HBTs)
- lateral (geometry) scalability

Table 1: Qucs and NGSpice simulation results comparison.
Note numerical compatibility of the simulated base current.
(The Gummel characteristic conditions - refer Fig. 2)

Vbe [V]	Ib (NGSpice) [A]	Ib (Qucs) [A]
0.42	4.800855e-13	4.923047e-13
0.43	6.230791e-13	6.230791e-13
0.44	7.936855e-13	7.936855e-13
0.45	1.018155e-12	1.018155e-12
0.46	1.316080e-12	1.316080e-12
0.47	1.714984e-12	1.714984e-12
0.48	2.253764e-12	2.253764e-12
0.49	2.987657e-12	2.987658e-12
0.50	3.995463e-12	3.995463e-12
0.51	5.390023e-12	5.390023e-12
0.52	7.333482e-12	7.333482e-12

Summary: The ADMS tool offers an excellent compact modeling environment. It allows faster development of advanced compact models and faster implementation into IC design tools. Combined with the GNU based CAD tools creates coherent and highly reliable modeling framework simplifying compact model evaluation and verification task across different simulation platforms and operating systems

NGSpice: mixed-level/mixed-signal circuit simulator

NGSpice is a mixed-level/mixed-signal circuit simulator [6]. Its code is based on three open source software packages: Spice3f5, Cider1b1 and Xspice and is one of the simulators in the **gEDA project**.

Spice3 does not need any introduction, is the most popular circuit simulator. In over 30 years of its life Spice3 has become a de-facto standard for simulating circuits.

Cider couples Spice3f5 circuit level simulator to DSIM device simulator to provide greater simulation accuracy of critical devices. DSIM devices are described in terms of their structures and materials.

Xspice is an extension to Spice3C1 that provides code modeling support and simulation of digital components through an embedded event driven algorithm.

NGSpice is a general-purpose circuit simulator program. It implements three classes of analysis:

- Nonlinear DC analyses
- Nonlinear Transient analyses
- Linear AC analyses

NGSpice implements the usual circuits elements, like resistors, capacitors, inductors (single or mutual), transmission lines and a growing number of semiconductor devices like diodes, bipolar transistors, mosfets (both bulk and SOI), mesfets, jfet and HFET.

Codemodels are available through the Xspice co-simulator o with the new ADMS Verilog-AMS codemodeling system.

Netlists can contain parameters and expressions. Parametric macromodels, often released by manufacturers, can be imported as-is into NGSpice.

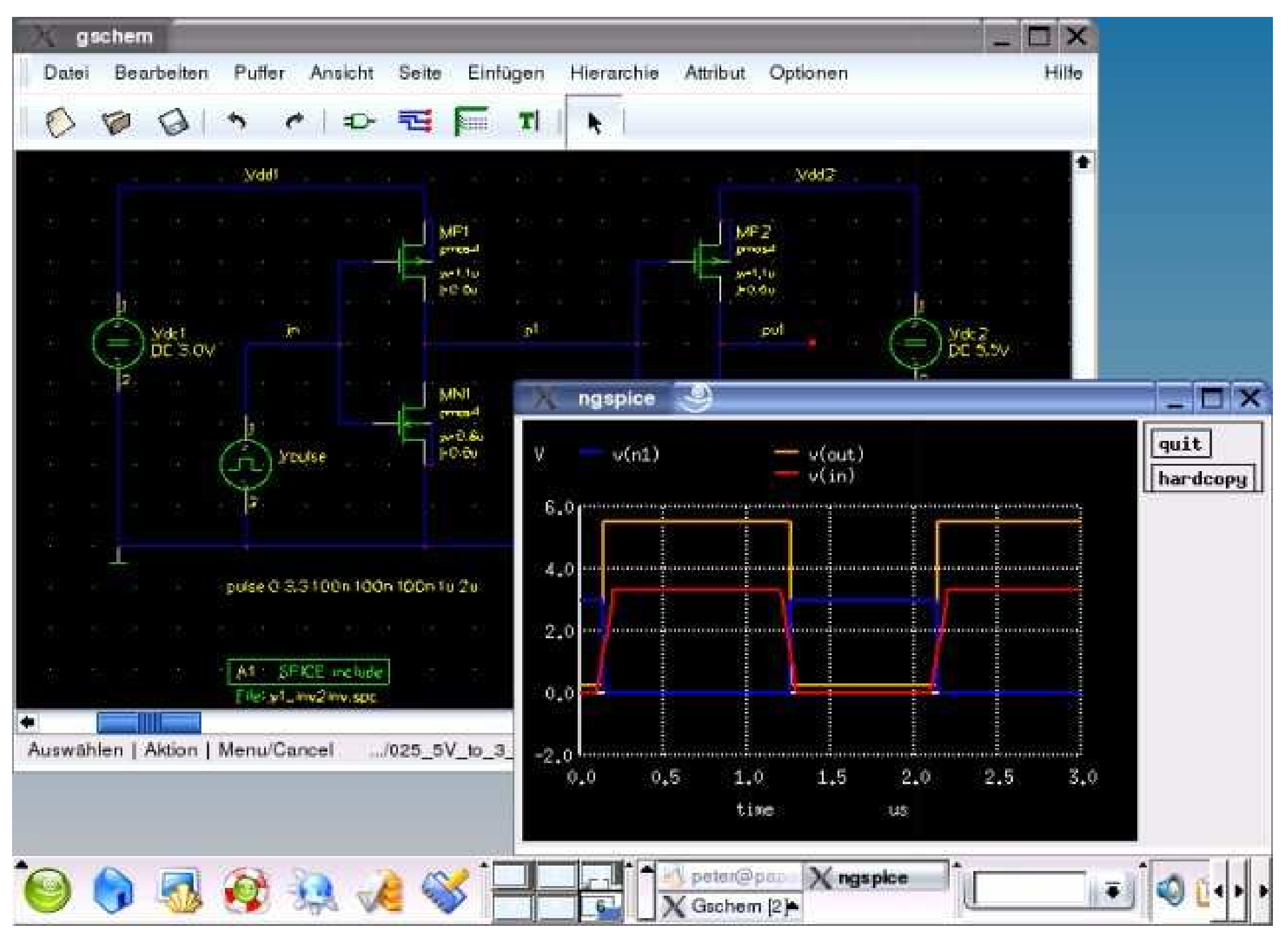

Fig. 5: NGSpice and gSchem running on Linux with KDE

The NGSpice XML interface

- ADMS is not (yet) included into NGSpice and must be downloaded separately.
- Verilog-AMS devices are compiled statically into the simulator and code must be present at *configure* time.
- Adms templates and codemodels devices are grouped under a common directory.
- Adms templates are used to translate Verilog-AMS code and fill with the appropriate code NGSpice model structure.
- There exist a template file for each file to be created.
- Spice noise analysis is not (yet) supported.
- There exist a "special" template file needed to generate the Makefile.am needed by NGSpice to build C code from XML and this file is processed first.

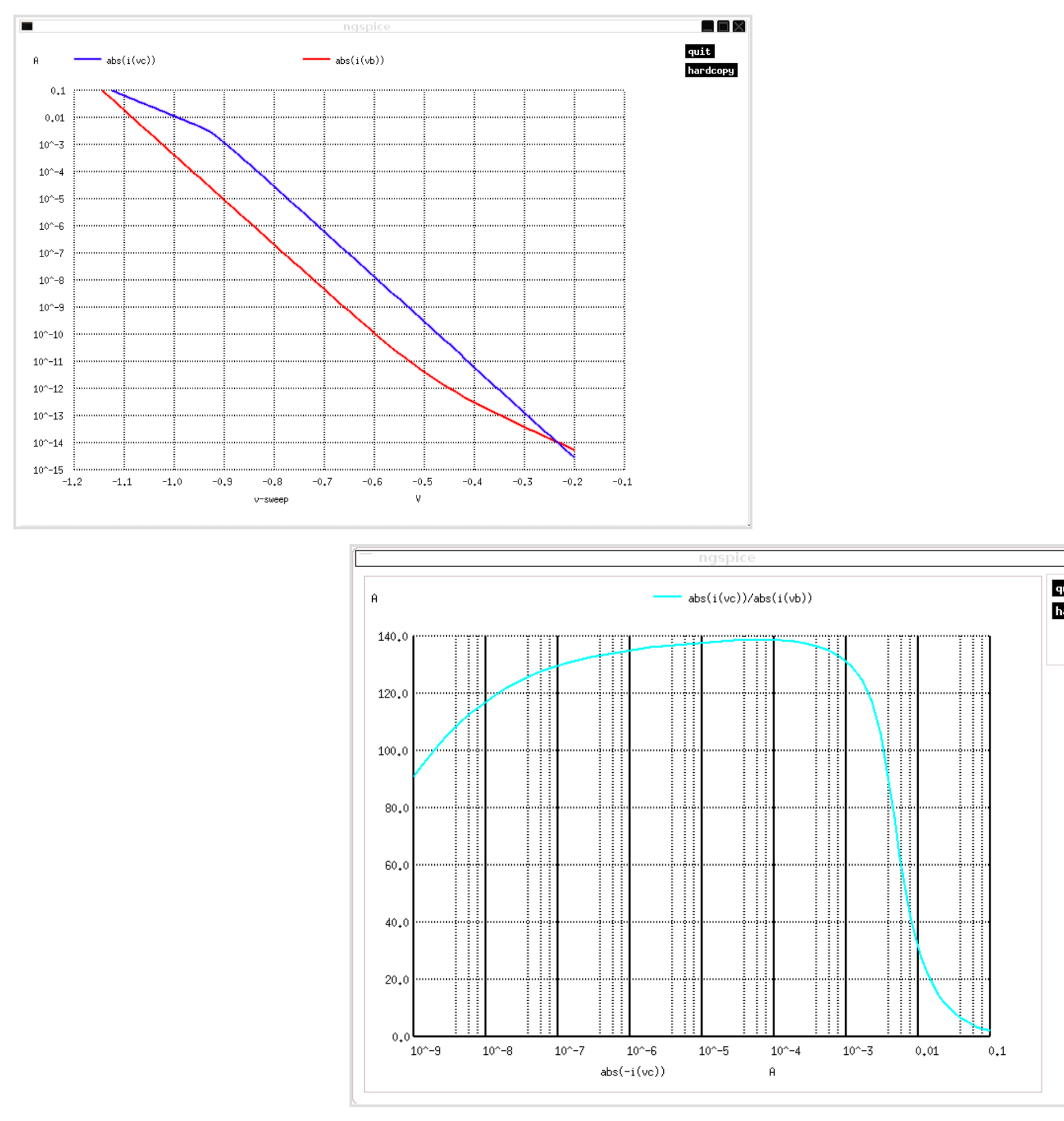

Fig. 6: Output plots for the NGSpice implementation of the HiCUM/L2 V2.22 device model

References:
[1] Qucs: http://qucs.sourceforge.net
[2] Stefan Jahn and Hélène Parruitte, "Qucs, A description, Verilog-AMS interface"
[3] Qt® is a registered trademark of Trolltech® ,Norway; http://www.trolltech.com
[4] FreeHDL: http://www.freehdl.seul.org
[5] Icarus Verilog: http://www.icarus.com/eda/verilog
[6] NGSpice web site: http://www.ngspice.org
[7] Latest code is available on CVS: http://ngspice.sourceforge.net/files/ng-spice-rework_CVS.tar.gz
[8] ASCO web site: http://sourceforge.net/projects/asco
[9] ADMS web site: http://mot-adms.sourceforge.net/
[10] HiCUM web site: http://www.iee.et.tu-dresden.de/iee/eb/hic_new/hic_intro.html

Authors' contacts: 1. Stefan Jahn <stefan@gruft.de>, 2. Mike Brinson <mbrin72043@yahoo.co.uk>, 3. Michael Margraf <michael.margraf@alumni.tu-berlin.de>, 4. Hélène Parruitte <parruitt@enseirb.fr>, 5. Bertrand Ardouin <ardouin@xmodtech.com>, 6. Paolo Nenzi <p.nenzi@ieee.org>, 7. Laurent Lemaitre <laurent.lemaitre@freescale.com>, Editor: Wladek Grabinski <wladek@grabinski.ch>

Presented on April 20, 2007 at MOS-AK Meeting in Schloss Premstaetten