Open-source circuit simulation tools for RF compact semiconductor device modelling

Article *in* International Journal of Numerical Modelling Electronic Networks Devices and Fields · September 2014

DOI: 10.1002/jnm.1973

Wladek Grabinski
GMC Reaserch
62 PUBLICATIONS **734** CITATIONS
SEE PROFILE

Mike Brinson
London Metropolitan University
78 PUBLICATIONS **394** CITATIONS
SEE PROFILE

Paolo Nenzi
ENEA
90 PUBLICATIONS **663** CITATIONS
SEE PROFILE

Francesco Lannutti
Synopsys
6 PUBLICATIONS **37** CITATIONS
SEE PROFILE

Open source circuit simulation tools for RF compact semiconductor device modelling

Wladek Grabinski[1] (editor), Mike Brinson[2], Paolo Nenzi[3], Francesco Lannutti[3],
Nikolaos Makris[4], Angelos Antonopoulos[4] and Matthias Bucher[4]

1 MOS-AK Group, email <wladek@mos-ak.org>
2 Centre for Communications Technology, London Metropolitan University, UK.
email mbrin72043@yahoo.co.uk.
3 University of Rome "La Sapienza", Italy; NGSpice Development Team
Email: < nenzi@die.uniroma1.it >
4 School of Electronic and Computer Engineering, Technical University of Crete, 73100 Chania, Greece email
<bucher@electronics.tuc.gr>

SUMMARY

MOS-AK is a European, independent compact modelling forum created by a group of engineers, researchers and compact modelling enthusiasts to promote advanced compact modelling techniques and model standardization using high level behavioral modelling languages such as VHDL-AMS and Verilog-A. This invited paper summarizes recent MOS-AK open source compact model standardization activities and presents advanced topics in MOSEFT modelling, focusing in particular on analogue/RF applications. The paper discusses links between compact models and design methodologies, finally introducing elements of compact model standardization. The open source CAD tools: Qucs, QucsStudio and ngspice all support Verilog-A as a hardware description language for compact model standardization. Latter sections of this paper describe a Verilog-A implementation of the EKV3 MOS transistor model. Additionally, the simulated RF model performance is evaluated and compared with experimental results for 90nm CMOS technology.

Received ………. July 2013

KEY WORDS: CAD; GNU; Qucs; QucsStudio; ngspice; compact modeling; EKV3; RF; MOSFET; Verilog-A

1. INTRODUCTION

MOS-AK (MOS-Modelle und Parameterextraktion Arbeitskreis) is a European, independent compact modelling (CM) forum [1] created by a group of engineers, researchers and compact modelling enthusiasts to promote advanced CM techniques and model standardization using high level behavioral modelling languages such as VHDL-AMS and Verilog-A. MOS-AK aims to encourage interaction and sharing of information related to the CM at all levels of device and circuit characterization, modelling and simulation. MOS-AK also aspires to build a modelling community with global connections by 1. promoting standardization of compact models and their implementation in software tools, 2. connecting national and local modelling groups at a European level and 3. building strong bilateral ties with similar organizations around the world. The group conducts regular meetings with European industry and academia to exchange information on the strengths and weaknesses of the industrialization of compact models [1]. Current activities include 1. drafting of standards and the provision of a center of competence for engineers, designers, managers and decision makers [2], 2. evaluating world-wide best practice and success stories and (3) delivering a comprehensive view of the European CM [3]. MOS-AK believes that the transfer of advanced CM methodologies through open source CAD tools to the semiconductor industry can be accelerated by providing comprehensive reports and reference papers on a range of subjects, such as (1) basic issues and concepts of device characterization and CM, 2. global CM issues embedding advanced CMOS processes, and (3) the provision of examples and analysis centred on best practice for the commercialization of compact models.

2. QUCS GPL OPEN SOURCE CIRCUIT SIMULATOR

The open source General Public Licence (GPL) circuit simulators Qucs [4] and QucsStudio [5] are the brain-child of German Engineer Michael Margraf. Since Qucs was first released for general use in December 2003 the software has evolved into an advanced circuit simulation and device modelling tool with a user friendly “graphical user” interface for schematic circuit capture, investigation of circuit and device properties from DC through RF to optical frequencies and the processing of post-simulation data. During Qucs’ first ten years around twenty engineers and scientists, from all over the world, have invested their expertise and time in support of the Qucs and QucsStudio projects. Since the initial Qucs release, roughly one million downloads of the software have been recorded by sourceforge.net. Interest in the software has resulted in world-wide support for the package and wide spread use by the RF and Integrated Circuit design communities. QucsStudio is similar to Qucs but includes additional communication system simulation and printed circuit layout capabilities. Both Qucs and QucsStudio support Verilog-A as a hardware description language for compact model standardization. The software also supports SPICE standard semiconductor device models and a range of compact semiconductor device models, including HICUM L0 and L2, EKV v2.6, BSIM 3, BSIM 4, FBH HBT and MESFET models.

2.1. QUCS FUNDAMENTAL RF SIMULATION CAPABILITIES

Although the Qucs simulator includes the fundamental DC, AC, and transient simulation features implemented in SPICE 2g6 and 3f5 it is not intended to be a replacement for SPICE. Qucs was originally conceived as a circuit simulation tool built on the main features of SPICE combined with S parameter analysis. As such it would provide engineers with a more comprehensive RF circuit analysis and design tool. Today, Qucs has evolved to a stage where it offers a range of RF simulation facilities, including small signal S parameter/noise analysis and Harmonic Balance simulation, plus a number of device and circuit modelling features which go far beyond traditional SPICE subcircuit and hardware macro-modelling. Central to the current version of Qucs are interactive and compiled modelling facilities for (1) non-linear equation-defined devices (EDD) [6] [7], (2) RF frequency dependent equation-defined devices (REDD) [8], (3) Verilog-A compact semiconductor device models [9] and use of Octave [10] for post simulation data processing and visualization. Category (3) model construction employs the Analogue Device Modelling Synthesiser (ADMS) [11]. Both Qucs and QucsStudio also provide a number of RF circuit synthesis tools which allow, for example, the design of lumped passive, microstrip and coplanar filters, attenuators, RF matching networks and single or coupled transmission lines. The Qucs simulation test circuits and output data plots presented in Figures 1 to 3 demonstrate a number of Qucs fundamental RF device modelling techniques, graphical user interface (GUI) driven S parameter simulation and post simulation data processing equation blocks.

2.2. MORE COMPLEX RF DEVICE MODELLING

Introduced in section 2. are a number of very basic RF device modelling and simulation techniques. With the exception of the GUI driven S parameter/noise and Harmonic Balance simulation these can be found in most circuit simulators derived from SPICE. In contrast to digital circuits, RF circuits mainly consist of a smaller number of active semiconductor devices often with a significant number of other fundamental components operating as linear or weakly non-linear devices.

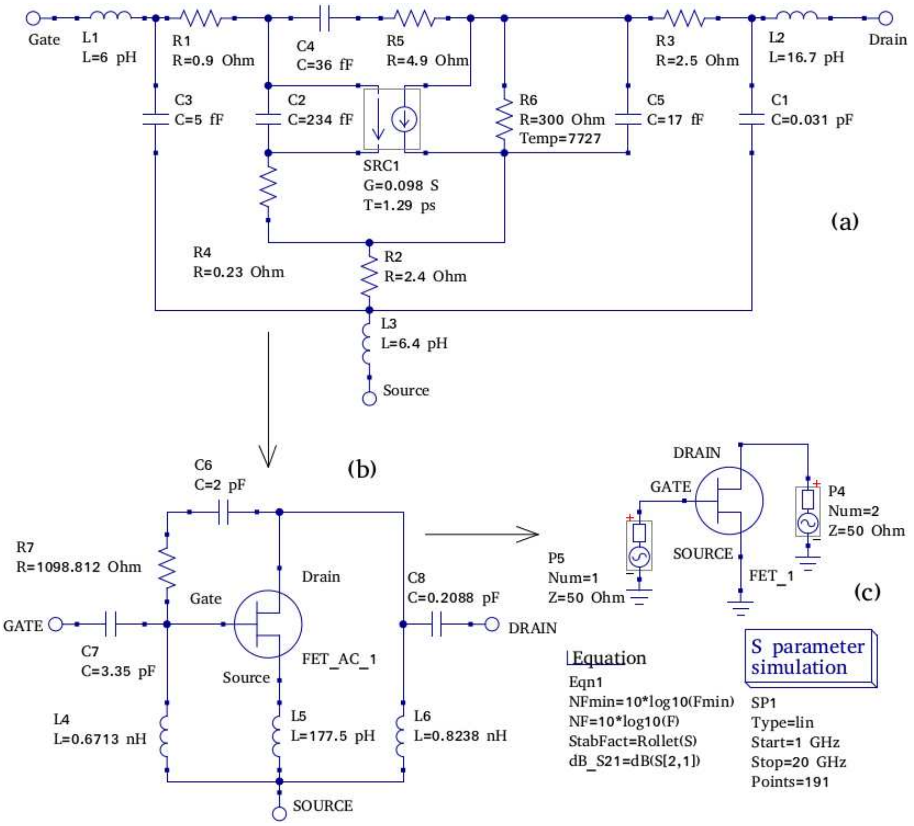


Figure 1. Example small signal RF transistor equivalent circuit and Qucs S parameter test circuit: (a) small signal AC model with package parasitic components, (b) an embedded transistor model with input and output matching networks, (c) a Qucs S parameter simulation test circuit plus post-simulation data processing equations listed in Qucs equation block Eqn1.

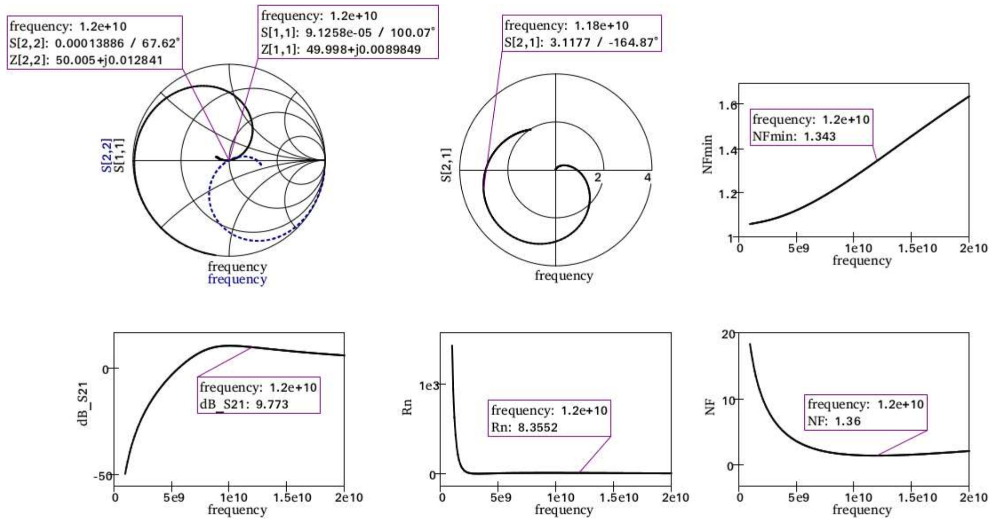
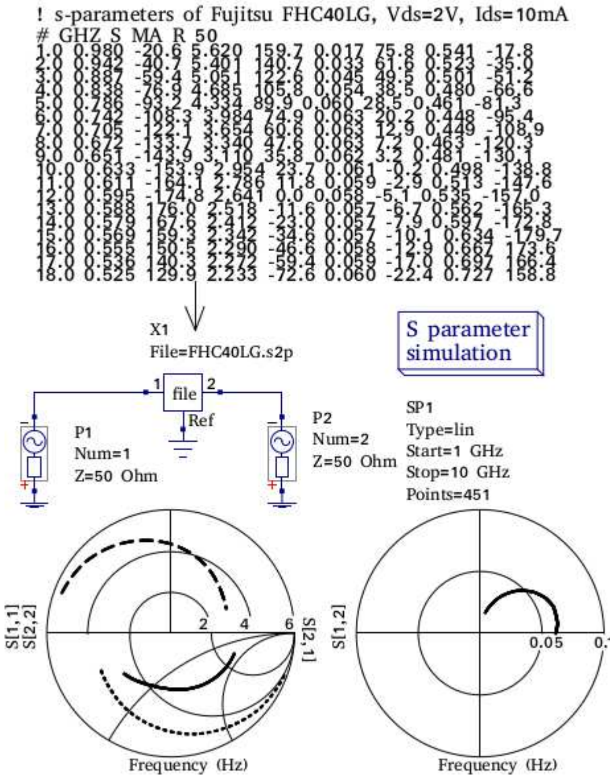


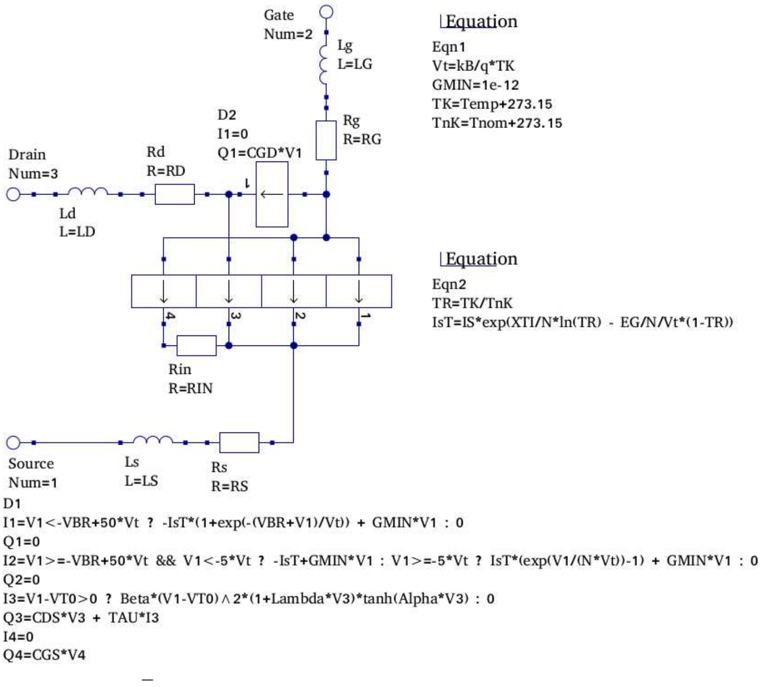
Figure 2. Small signal RF transistor simulation data output by the test circuit illustrated in Figure 1 (c); S and Z parameters (Smith and Polar charts), noise figure NF, noise figure minimum NFmin, equivalent noise resistance Rn and S21 in dB (x/y Cartesian plots).



```
! s-parameters of Fujitsu FHC40LG, Vds=2V, Ids=10mA
# GHZ S MA R 50
1.0 0.980 -20.6 5.620 159.7 0.017 75.8 0.541 -17.8
2.0 0.942 -40.7 5.401 140.7 0.033 61.6 0.523 -35.0
3.0 0.887 -59.4 5.051 122.6 0.045 49.5 0.501 -51.2
4.0 0.838 -76.9 4.685 105.8 0.054 38.5 0.480 -66.6
5.0 0.786 -93.2 4.334 89.9 0.060 28.5 0.461 -81.3
6.0 0.742 -108.3 3.984 74.9 0.063 20.2 0.448 -95.4
7.0 0.705 -122.1 3.654 60.6 0.063 12.9 0.449 -108.9
8.0 0.672 -133.7 3.340 47.6 0.063 7.2 0.463 -120.3
9.0 0.651 -143.9 3.110 35.8 0.062 3.2 0.481 -130.1
10.0 0.633 -153.9 2.954 23.7 0.061 -0.2 0.498 -138.8
11.0 0.611 -164.1 2.786 11.8 0.059 -2.9 0.513 -147.6
12.0 0.595 -174.8 2.641 0.0 0.058 -5.1 0.535 -157.0
13.0 0.588 176.0 2.518 -11.6 0.057 -6.7 0.562 -165.3
14.0 0.579 167.6 2.412 -23.0 0.057 -7.9 0.597 -172.8
15.0 0.569 159.3 2.342 -34.6 0.057 -10.1 0.634 -179.7
16.0 0.555 150.5 2.290 -46.6 0.058 -12.9 0.667 173.6
17.0 0.536 140.3 2.272 -59.4 0.059 -17.0 0.697 166.4
18.0 0.525 129.9 2.233 -72.6 0.060 -22.4 0.727 158.8
```

Figure 3. S parameter test circuit and simulation data for a high frequency transistor modelled by tabular S parameter measured data, in Touchstone file format [12], listed at the top of the figure.

At high frequencies non-linear frequency dependent effects play an increasingly important role in RF circuit operation, for example, one well known physical effect is the frequency dependent skin effect found in conductors. The equation-defined device (EDD) was first implemented in the Qucs circuit simulator in 2007, providing an interactive tool for non-linear semiconductor device equation-defined model construction. An example RF semiconductor device model using multiple EDD is shown in Figure 4. This equivalent circuit diagram represents a non-linear Curtice model [13] of a MESFET transistor. The EDD in Figure 4 are components D1 and D2 with individual two terminal branch currents and stored branch charge given by the I and Q equations. These equations represent the physical properties of the device being constructed. The Qucs EDD element is an important non-linear modelling device in that it allows interpreted experimental device models to be easily developed, de-bugged and tested, via schematic capture, as a preliminary stage to the construction of a compiled Verilog-A hardware description language production model [14]. The Qucs simulator, after passing the current and charge information specified by I and Q, automatically determines the partial derivatives of I and Q that are needed for the non-linear DC, transient and Harmonic Balance simulation of a circuit under test. Figure 4 also shows a number a parasitic L and R components used to model the MESFET device package. Figure 5 presents a typical set of S parameter simulation data for the Qucs EDD MESFET model.



```
Equation
Eqn1
Vt=kB/q*TK
GMIN=1e-12
TK=Temp+273.15
TnK=Tnom+273.15
```

```
Equation
Eqn2
TR=TK/TnK
IsT=IS*exp(XTI/N*ln(TR) - EG/N/Vt*(1-TR))
```

```
D1
I1=V1<-VBR+50*Vt ? -IsT*(1+exp(-(VBR+V1)/Vt)) + GMIN*V1 : 0
Q1=0
I2=V1>=-VBR+50*Vt && V1<-5*Vt ? -IsT+GMIN*V1 : V1>=-5*Vt ? IsT*(exp(V1/(N*Vt))-1) + GMIN*V1 : 0
Q2=0
I3=V1-VT0>0 ? Beta*(V1-VT0)∧2*(1+Lambda*V3)*tanh(Alpha*V3) : 0
Q3=CDS*V3 + TAU*I3
I4=0
Q4=CGS*V4
```

Figure 4. Qucs Curtice non-linear MESFET model with package parasitic components: components D1 (four section model) and D2 (one section model) are non-linear EDD with I and Q equations listed under D1 and D2 respectively. Other L and R components represent linear sections of the MESFET model or package parasitic components.

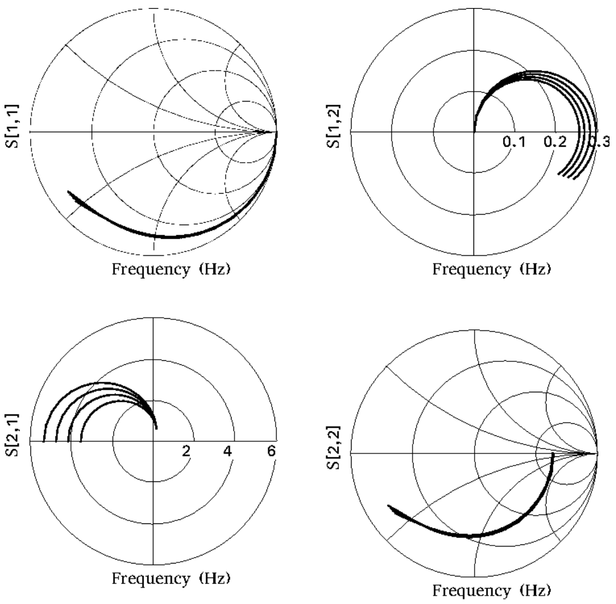
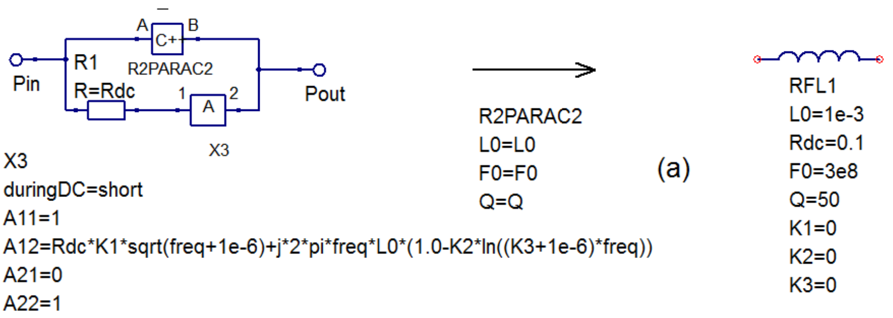


Figure 5. Typical S parameter simulation data for the Qucs EDD MESFET model: device DC bias conditions are Vgs = 0V and Vds =5 to20 V.

Non-linear high frequency dependent physical effects are often observed in RF passive component characteristics. Both Qucs and QucsStudio include specific equation-defined modelling tools to handle such cases. Illustrated in Figure 6 is a QucsStudio high frequency model for an inductance constructed from functional blocks X3, R2PARAC2 and resistor Rdc. These elements model device inductance, series resistance (Rdc), and inductance self-resonance and frequency dependent resistive and inductive effects. In Figure 6(a) schematic block X3 represents a two port A network, where A11=1, A12 is a series impedance Z, connected between terminals 1 and 2, A21=0 and A22=1. In this example Z is a complex number with real and imaginary components specified by algebraic functions of signal frequency freq. Similarly, the two terminal component R2PARAC2 is represented by a Verilog-A model. See Figure 6(b) for the model code. One point worth emphasizing is the way that components modelled by algebraic equations, which include freq, are represented by Qucs/QucsStudio as two port RF networks and combined with Verilog-A models to form RF subcircuits. In the RFL subcircuit given in Figure 6(a) values for frequency dependent coefficients K1 (series resistance), K2 and K3 (inductance) are passed to X3 via the RFL symbol parameter list. The Verilog-A code shown in Figure 6(b) represents a parallel connection of components Cp and Rp modelled in terms of low frequency inductance L0, self-resonance frequency F0 and inductance Q factor. Figure 7 illustrates a large signal AC test circuit for determining the frequency characteristics of non-linear inductance using swept frequency Harmonic Balance simulation. The high frequency inductance model demonstrates how traditional subcircuit features can be extended with Qucs/QucsStudio non-linear modelling tools, making development of practical RF components straightforward.



(b)

```
//   Verilog-A RF Inductance block Rp||Cp called RpPARACp.va

`include "disciplines.vams"
`include "constants.vams"
`define attr(txt) (*txt*)
module RpPARACp(A, B);
inout A, B ;  electrical A,B;
parameter real L0 = 1e-3 from [1e-20 : inf)    `attr(info="Low frequency inductance value" unit = "H" );
parameter real F0 = 800e3 from [1e-7 : inf)  `attr(info="Self resonant frequency" unit = "Hz");
parameter real Q = 50 from [1 : inf)        `attr(info="Quality factor");
//  Variables
real PIx2xF0, Cp, Rp;
// Model branches
branch (A, B) bAB;
analog begin
@(initial.model)
begin
  PIx2xF0 =`M.TWO.PI*F0;   Cp = 1/(PIx2xF0*PIx2xF0*L0); Rp = Q*PIx2xF0*L0;
  if (Rp >= 1e9)  begin  Rp = 1e9; end
end
// Current contribution etc
`ifdef insideQucsStudio
     R(bAB)  <+ Rp; C(bAB)  <+ Cp;
 `else
     I(bAB)  <+ 1.0/Rp; I(bAB)  <+ ddt(Cp*V(bAB));
 `endif
end
endmodule
```

Figure 6. A high frequency RF inductance model with self-resonance and frequency dependent resistive and inductive effects modelled.

2.3. QUCS/QUCSSTUDIO VERILOG-A COMPACT DEVICE MODELLING

The Analogue Device Model synthesiser (ADMS) is employed by both Qucs and QucsStudio as the primary tool for the construction of compact semiconducor device models and high level functional circuit macromodels. The latest versions of these simulators are distributed with ADMS as one package. ADMS 2.30 is the current GPL version of the Verilog-A to C++ translator. ADMS is designed specifically for analogue compact model development but has only those sections of the Verilog-A subset of Verilog-AMS implemented that are required for analogue model development. The ADMS compiler does however, also include a large percentage of the language extensions specifically designed for compact model construction.

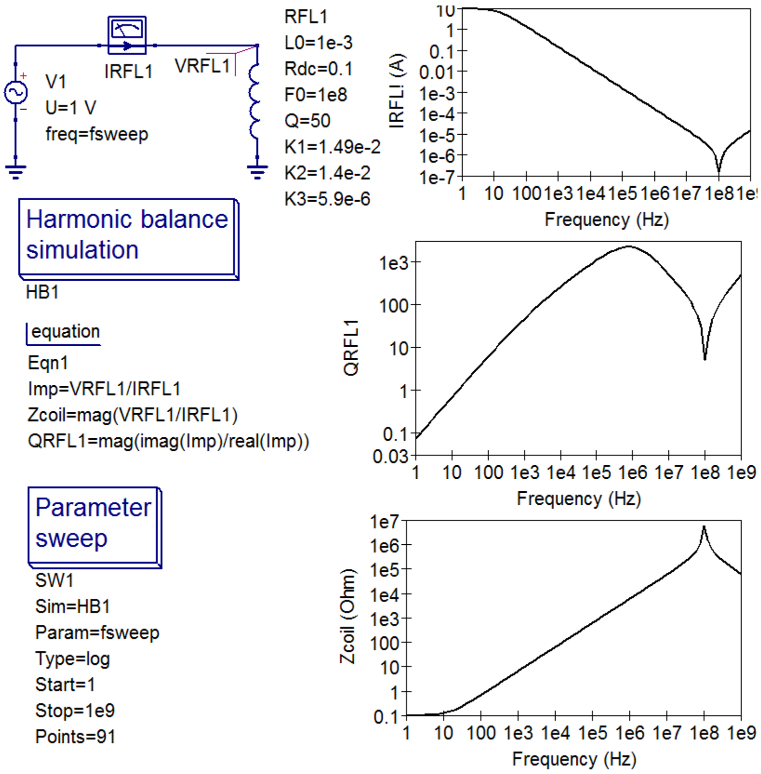


Figure 7. A swept frequency Harmonic Balance non-linear inductance test circuit and typical simulation results for a RF inductance with L0=1mH, Rdc = 0.1 Ω, Fo = 100MHz, Q=50, K1=1.49e-2, K2=1.4e-2 and K3=5.9e-6.

Interfacing ADMS and a circuit simulator is done through XML script files which generate C++ code for the model under construction. These XML scripts are different for each circuit simulator model application interface. After translation from Verilog-A to C++ the ADMS generated C++ is compiled and linked to the main body of a circuit simulator code using the C++ toolset available with a given operating system. With Qucs and QucsStudio additional code needs to be generated for the correct operation of the schematic capture part of the circuit simulation software. Although Qucs and QucsStudio have a similar structure and function the XML scripts provided with each package are different reflecting the underlying structure of the circuit simulator adopted by their developers, for example Qucs uses static code libraries and QucsStudio dynamic linked code libraries. Further details of the Qucs and Qucstudio approaches to compact device modelling using Verilog-A can be found in the following references [15] [16]. However, it is worth noting that the C++ compact models built with either Qucs or QucsStudio and ADMS 2.30 work in all simulation domains, specifically DC, small signal AC (including noise analysis), Transient, S parameter small signal AC (including noise analysis) and Harmonic Balance Simulation. The wide band RF amplifier circuit given in Figure 8 demonstrates these Qucs compact device modelling and multi-domain simulation capabilities. The test circuit shown in Figure 8 is based on a RF transistor specified by a HICUM L0v1.3 BJT model with external biasing and signal stabilisation components. The input and output components also match, approximately, the circuit input and output impedances to 50 Ω. The simulation test circuit and output data plots illustrated in

Figures 8 and 9, clearly illustrate the circuit schematic capture, device modelling, circuit simulation and data visualisation capabilities of a modern circuit simulator.

2.4 THE FUTURE DIRECTION OF QUCS/QUCSTUDIO SIMULATOR DEVELOPMENT

In May 2013 the Qucs development team celebrated the tenth aniversary of the release of Qucs version 0.0.1. Over the last ten years the Qucs GPL software package, and the more recently released QucsStudio software, have develop from a very basic SPICE like circuit simulator to an interactive package with significant RF simulation features and a range of

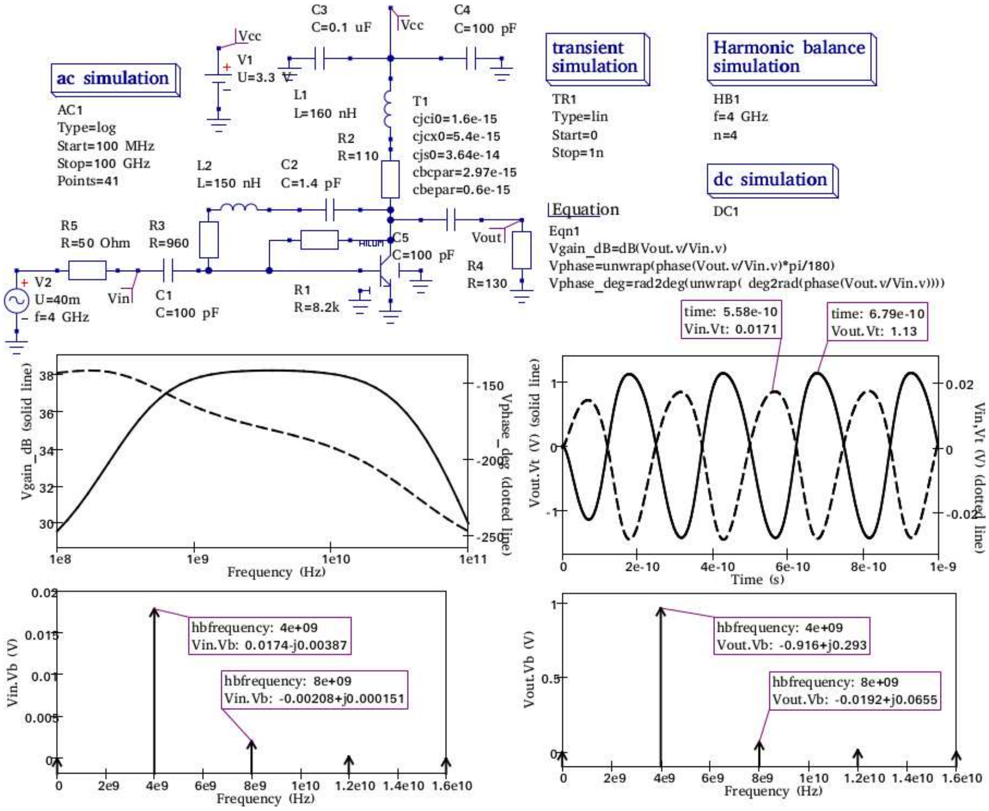


Figure 8. A wide band RF single BJT amplifier test circuit with AC, Transient and Harmonic Balance simulation output plots: the BJT is 9odeled with HICUM L0v1.3 Verilog-A code compiled with ADMS 2.30.

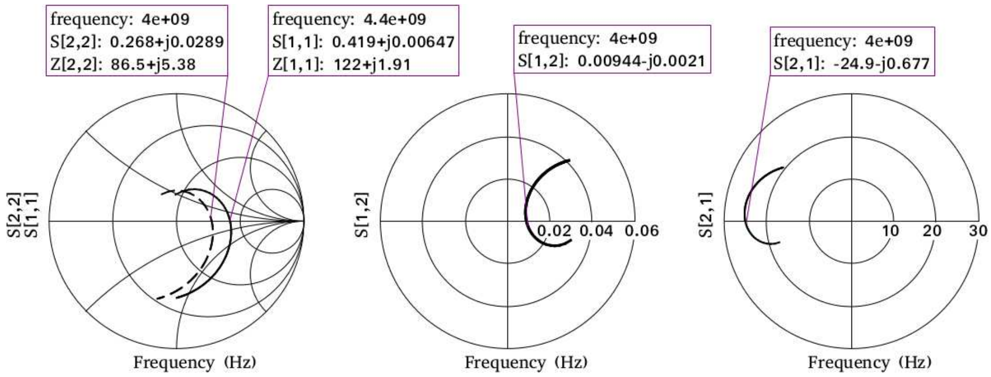


Figure 9. S and Z parameter data for the wide band amplifier test circuit (as in Fig. 8)

compact device modelling tools which allow device modelling engineers and RF circuit design engineers to experiment with emerging circuit technologies and new RF circuit designs. Plans for the future development of these packages include increased numbers of industrial standard device models, see for example Figure 10, an improved GUI based on toolkit Qt4 rather than Qt3 [17] and the addition of more fundamental circuit and system simulation capabilities.

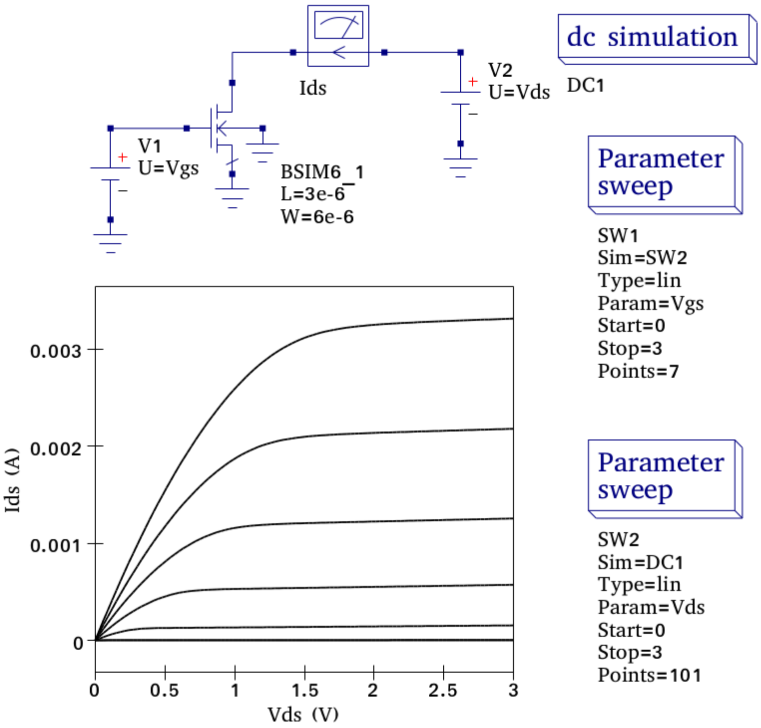


Figure 10. Typical Qucs BSIM 6 bulk nMOS DC I/V characteristics based on BSIM 6.0 Verilog- code A compiled with ADMS 2.30.

3. NGSPICE: THE SPICE HERITAGE; CIRCUIT SIMULATION AND MODELLING

The ngspice [19] open source project started in 1999 with the intention of fixing code errors in, and making improvements to, the Berkeley SPICE 3 [20] circuit simulator. At that time SPICE 3 was no longer under development by it’s original developers. The principle aims of the ngspice project are 1. to fix bugs in the SPICE 3 code, 2. to merge into a single tool the many patches that were available for the SPICE 3 package, and 3. to support new SPICE device models (in 1999 these included the new emerging BSIMSOI models). Over a period of more than ten years a strong engineering community has formed to support and take forward the ngspice project to it’s next level of development. Today, a stable development team has primary responsible for the maintenance and development of the ngspice package. This core team is based in Germany (Robert Larice, Holger Vogt, and Dietmar Warning) with additional new members providing contributions from Italy (Francesco Lannutti and Stefano

Perticaroli). The ngspice simulator integrates within its structure the DSIM [21] device level simulator and the XSPICE [22][23] mixed signal simulator. Ngspice also supports two different hardware description languages, namely the XSPICE code-level modelling interface that allows developers to add C-code models to the package, and the newer ADMS (Automatic Device Model Synthesizer) subsystem, which supports the Verilog-A modelling language. Currently, ngspice includes HICUM level 0 and level 2, MEXTRAM , EKV v2.6, and PSP v102.1 CM models written in Verilog-A.

3.1. NGSPICE FOR RF ANALYSIS AND MODELLING

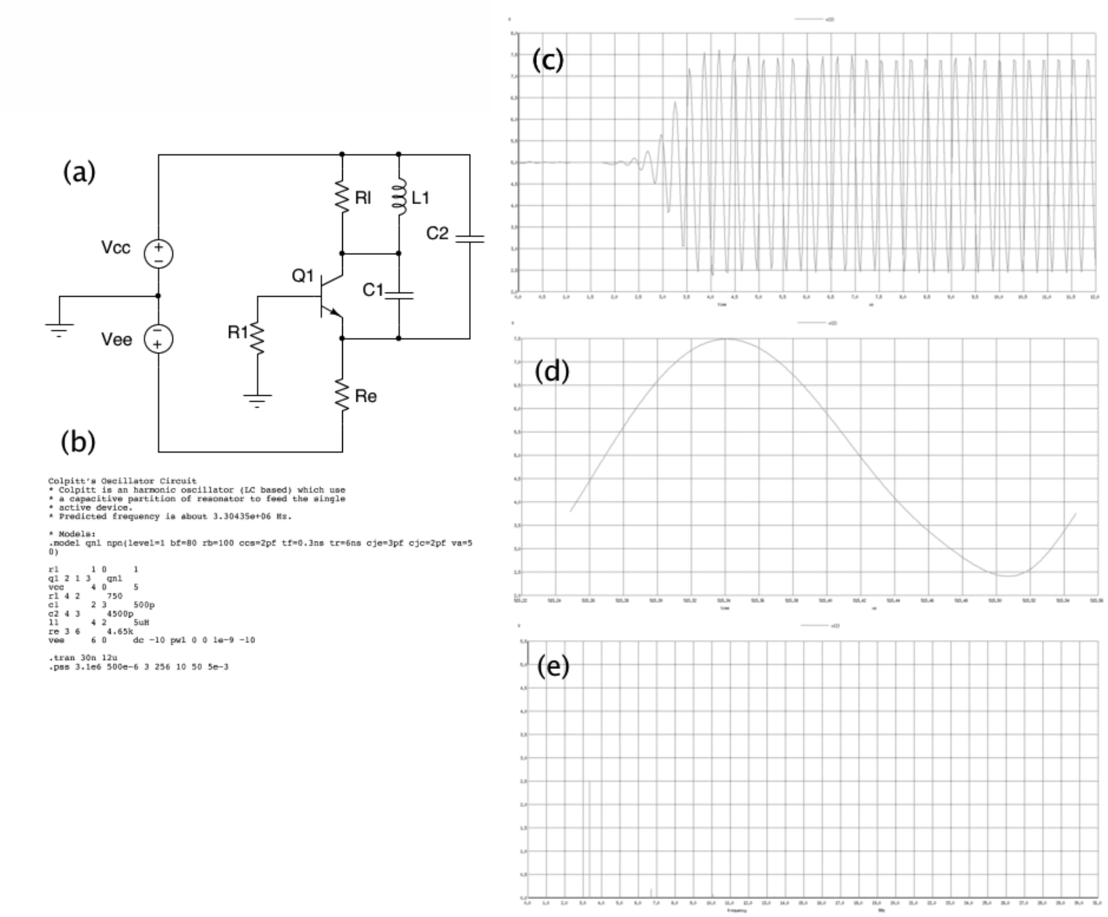
Ngspice supports SPICE 3 DC, AC (linear small signal) and Transient analyses, while simultaneously offering limited support for RF circuit analyses. Periodic Steady State (PSS) analysis has been recently added to the software [24][25] to provide large signal analysis of autonomous systems. The ngspice implementation of PSS is based on the time domain shooting method, allowing the prediction of oscillator output signal frequencies and amplitudes at the fundamental frequency and harmonics of oscillators. An example of a typical ngspice PSS test circuit with a set of simulation output waveforms for a BJT single transistor Colpitts oscillator is presented in Figure 11. The simulation results for the PSS analysis are illustrated as a time domain plot of one or more output waveforms over one period of oscillation, once the output signal has reached a steady state, plus a corresponding frequency domain plot showing the amplitudes of the output signal fundamental frequency component and the harmonic frequency components. S-parameter extraction for arbitrary circuits is another feature implemented in ngspice. It uses the simulators control language to perform a series of small signal AC simulations, extracting the circuit-under-test S-parameters with post processing scripts (the detailed procedure is described in [26]). Figure 12 shows the the extraction results for a simple LC filter. Ngspice also includes facilities which allow S-parameters extracted from a simulation to be written in the Touchstone “.s2p” standard file format to disk, for later use with an RF simulator external to ngspice. At the time of writing this paper, ngspice only supports 2 port S-parameter extraction. Unlike Qucs ngspice does not provide any specific RF modeling tools. However in most practical situations, the XSPICE code-level modelling tool distributed with ngspice and conventional SPICE macro-modeling techniques can be effectively used for designing RF circuits.

3.2. NGSPICE VERILOG-A COMPACT DEVICE MODELING

Verilog-A is one of the currently adopted standard hardware description language for compact model development. Ngspice, similar to it’s predecessor SPICE 3, does not natively provide any support for model development with this language but uses the ADMS tool [11] to translate Verilog-A compact model statements into C or C++ model code. ADMS is a powerful model synthesizer that translates Verilog-A coded compact models firstly into an intermediate XML representation and secondly into simulator C code, using scripts written in an ADMSPATH language. Each script transverses the XML representation of a model and writes output to a C code file (in the case of ngspice) containing the model code formatted for linking with a simulator’s API (Application Programming Interface). Ngspice uses a slightly modified version of ADMS 2.30, the latest publicly available version of the ADMS package [27]. The ADMS approach to converting Verilog-A model code to simulator C or C++ requires a set of scripts specific to a simulator in order to translate a high level hardware description model into C or C++ simulator code. In ngspice, ADMS XML scripts are not available for all forms of circuit simulation; in particular SPICE 3 linear noise analysis is missing. However, all the principal forms of simulation are supported, including DC, AC, and transient analysis. Ngspice (as of version 25) implements four device models synthesized from Verilog-A code: HiCUM level 0 and level 2 BJT models, EKV 2.6 MOS models, the MEXTRAM BJT model, and the PSP102.1 MOS model. It is possible to add new models to ngspice given their Verilog-A source code, but this requires modification of the simulator source code due to the fact that ngspice statically links compact model C code with the main body of the ngspice simulator C code. The detailed procedure for adding new compact models to ngspice is described in [27].

3.3. THE FUTURE DIRECTION OF NGSPICE SIMULATOR DEVELOPMENT

The drawbacks that have emerged when using ngspice are all, in general, due to the SPICE 3 heritage. SPICE 3 development ended around 1990 and since then commercial simulators have continually implemented new features and restructured their simulator-to-device API. SPICE 3 was, for a long time after 1990, used as a driver platform for introducing and testing new compact models. For a model to be accepted it needed to be introduced into SPICE 3 for its validity to be accepted by the modeling community. This mitigated the obsolescence of SPICE 3 (and thus ngspice). Today this requirement is no longer true as Verilog-A has been is wdely established as the standard language for compact modeling. In the future further ngspice development to implement Verilog-A models without the need for recompiling the simulator core C source code each time a new Verilog-A model is added to the simulator, In particular CM model development with the existing ngspice API has revealed its limitations during the development of the PSS analysis because it was found to not be possible to isolate linear from non-linear jacobian contributions and static components from dynamic ones. An updated ngspice API, together with a new compiled model interface, is one of the current development lines for ngspice. Once in place this will facilitate the development of new RF analyses and RF component modeling techniques. Ngspice, being tied to its SPICE 3 heritage, encouraged compatibility with past developments, at netlist level and at device level, often remaining in step with SPICE derived commercial simulators. In many instances commercial simulators extend the original SPICE hardware description language through netlist pre-processing (e.g. functions and parameters), improved behavioral models (e.g., table defined controlled generators), and advanced post-processing of data output during simulation (e.g. measure). For ngspice, improving compatibility with other SPICE derived simulators is another development line. Similarly, a third line of development is the improvement of ngspice simulation speed by reducing matrix solve time, particularly for large circuits, by device evaluation (ngspice already implements OpenMP parallelization of device evaluation for some devices). Recently, the KLU solver [28] has been implemented in ngspice [29] to replace the original SPICE Sparse library with development activity ongoing to improve the performance of the circuit simulator. Current ngspice development activities appear to be strongly converging towards the construction of a simulator that is compatible with other SPICE derived simulators which implement industry validated algorithms and emply up to date models. Good Verilog-A support and development of a more powerful netlist processing language are fundamental in the quest to significantly improve the performance of ngspice through allocation of all available development resources. On the other hand, exploitation of multicore and many core CPU architectures to speed-up simulation of large circuits or to perform parallel simulations at once (like in Monte Carlo analysis) remains a second principal line of development for ngspice.



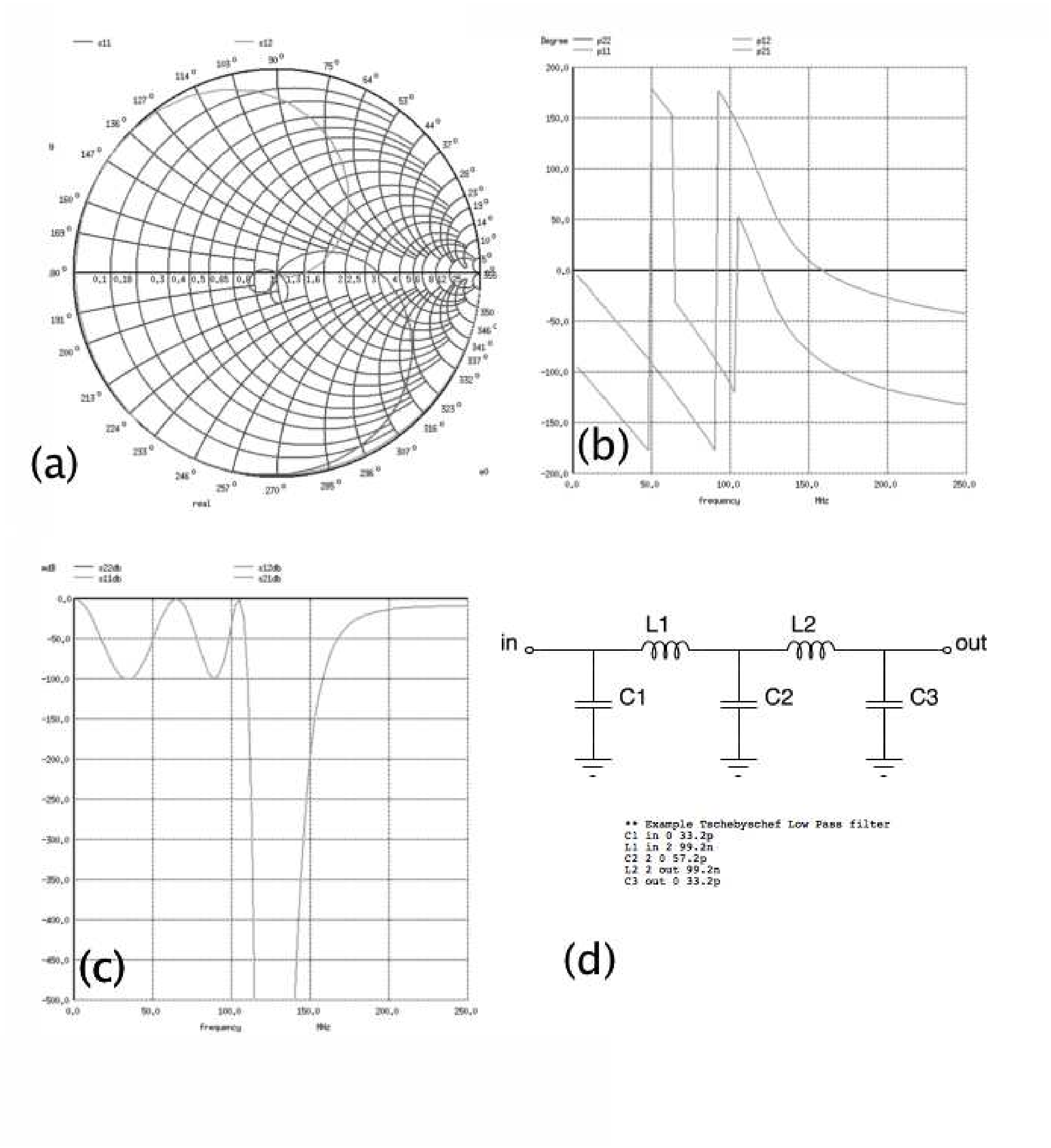
```
Colpitt's Oscillator Circuit
* Colpitt is an harmonic oscillator (LC based) which use
* a capacitive partition of resonator to feed the single
* active device.
* Predicted frequency is about 3.30435e+06 Hz.

* Models:
.model qnl npn(level=1 bf=80 rb=100 ccs=2pf tf=0.3ns tr=6ns cje=3pf cjc=2pf va=5
0)

r1       1 0      1
q1 2 1 3    qnl
vcc      4 0      5
rl 4 2      750
c1       2 3      500p
c2 4 3      4500p
l1       4 2      5uH
re 3 6      4.65k
vee      6 0      dc -10 pwl 0 0 1e-9 -10

.tran 30n 12u
.pss 3.1e6 500e-6 3 256 10 50 5e-3
```

Figure 11. PSS simulation of a Colpitts BJT oscillator: (a) test circuit; (b) test circuit ngspice netlist; (c) transient time domain plot of voltage at the BJT collector node; (d) PSS time domain plot of voltage at the BJTcollector node over one period; and (e) frequency domain plot of the BJT collector node voltage showing the amplitudes of the output signal DC component, fundamental frequency component and harmonic frequency components.



```
** Example Tschebyschef Low Pass filter
C1 in 0 33.2p
L1 in 2 99.2n
C2 2 0 57.2p
L2 2 out 99.2n
C3 out 0 33.2p
```

Figure 12: Ngspice S-parameters extraction for the filter circuit shown in (d) using the procedure highlighted in [26]. In this example S-parameter data are displayed via a Smith chart (a) and as phase and magnitude quantities plotted in (b) and (c), respectively.

4. ADMS: AUTOMATIC DEVICE MODEL SYNTHESIZER

This section gives a brief overview of the solutions currently available to port compact models to currently available circuit simulators. Compact models can be viewed as a set of electrical equations that describe, at different levels of precision, the electrical properties of semiconductor micro-structures [30]-[31] MOS transistors and bipolar transistors being amongst the most familiar examples. Recently the MOS-AK Compact Modeling Group has promoted Verilog-AMS [32] as a standard language for describing the physical properties expressed by CM. In the last two years more and more companies and universities have started to use the Verilog-AMS hardware description language to code CM properties. NXP (formerly Philips Semiconductors) and the Arizona State University are, for example, delivering the successive releases of PSP – the new CMC-approved standard MOS transistor model in the Verilog-AMS format. They also provide a release in the C format language automatically derived from the Verilog-AMS code. The Technical University of Dresden (HICUM team [36]) and the Technical University of Delft (MEXTRAM team [37] are proposing a Verilog-A based implementation of their bipolar compact models on their respective download web sites. It is interesting to note that just recently the MEXTRAM team decided to move the source code of MEXTRAM to sourceforge.net - an open-source development web site [37] Using a high-level behavioral description language for compact modeling makes far easier the development of compact models. All the fine-tuning work of bringing compact models into simulators goes to the shoulders of the EDA companies. This allows the compact model developers to better focus on the modeling performances of their models. Today the major EDA companies offer a solution to run on the fly simulations of electrical circuits that include Verilog-A coded files. An open source software tool—called automatic device model synthesizer (ADMS)—that supports and simplifies compact model development, implementation, distribution, maintenance, and sharing has been proposed by industrial partners [33]-[35]. ADMS has been designed to make these tasks simple, efficient, and robust, see TABLE I. The use of the ADMS tool with SPICE derived circuit simulators, including ngspice and Qucs was described in a previous section. To aid the compact modelling community some companies (i.e. Cadence) make freely available the ADMS xml scripts that they wrote for their simulator interfaces.

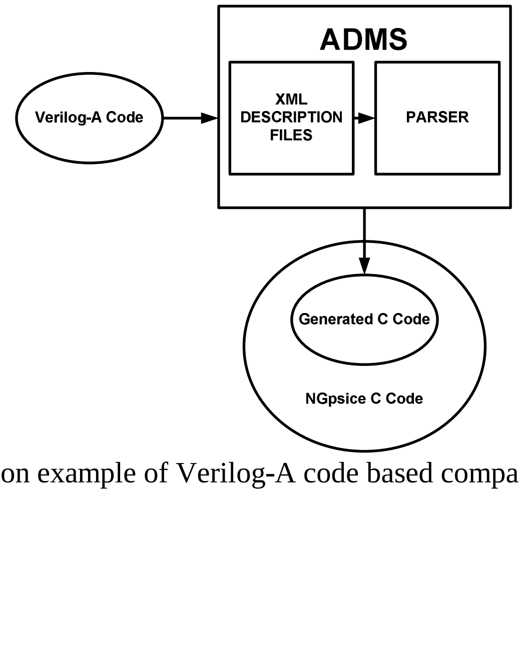


Figure 13: Integration example of Verilog-A code based compact model in ngspice.

TABLE I: List of the free CAD tools supporting ADMS

Tool	License	ADMS Ver.	Available From	Notes
Qucs	GPL	2.30	http://qucs.sourceforge.net/	DC, AC, TRAN, S-par, noise and HB
QucsStudio	GPL	2.30	http://www.mydarc.de/dd6um/QucsStudio/qucsstudio.html	DC, AC, TRAN, S-par, noise and HB
ngspice	GPL[A]	2.20	http://ngspice.sourceforge.net/	DC, AC and TRAN only
GNUCAP	GPL	2.30?	https://github.com/gserdyuk/models-mot-adms	DC, AC and TRAN only.
SymicaDE	Commercial [B]	2.29	http://www.symica.com/products/symica-free-edition	DC, AC, TRAN, noise, S-par and mixed mode AMS extensions

[A] Upverter/ADMS is a development project for ngspice – includes bug fixes and additional Verilog-AMS features.

[B] Symica Free addition: This package is intended for personal non-commercial use in education and scientific research: features SymSpice, circuits with up to 300 elements, single thread simulation, and Verilog-A interface (extended for Verilog- AMS mixed mode simulation). http://www.symica.com/products/symica-free-edition * ADMS is embedded in these circuit simulators.

5. IMPLEMENTATION OF EKV3 VERILOG-A BASED MODEL INTO NGSPICE VIA ADMS

The EKV3 model was integrated in release 25 of ngspice using admsXml version 2.3.0. The code was compiled with gcc 4.6.1 on the Ubuntu 11.04 version of Linux. The EKV3 RF module is called using LEVEL parameter 80. Executable files were produced that are compatible with Linux and Windows distributions. By modifying outitf.c file of the ngspice source code, raw data files are produced that are compatible with Agilent ICCAP. In this work, ngspice RF simulation results are presented for 90nm TSMC CMOS technology. The scattering parameters (S-parameters) are extracted using the theoretical background provided in [37]. The simulation setup was based on Agilent's ICCAP extraction environment and ngspice S-parameters extraction circuits, as shown in Table II. Two instances of EKV3 MOS transistors are used in order to simulate the RF behavior of the model.

Table II. Input file for the simulation of S-parameters in ngspice

Scattering Parameters with EKV3	******Subcircuit ********
.param Rload=50 Vbias_in=0.5 Vbias_out=0.9 Cblock=1K	.SUBCKT S_PARAM 1 2 3 4
Lpass=1K	.param VDCIN=0 VDCOUT=0
.include ekv3_018.par	************************
vs1 s 0 dc 0.0	C1 1 10 'Cblock'
vb1 b 0 dc 0.0	R1 10 20 'Rload'
vs2 s2 0 dc 0.0	VIN 20 0 DC 0 AC 1
vb2 b2 0 dc 0.0	LP1 1 A1 'LPass'
********NMOS instances************	VINDC1 A1 0 DC 'VDCOUT'
m1 d g s b nch	G1 0 3 10 0 2
+W=2u	************************
+L=100n	VI 31 3 DC 0 AC 1
+NF=10	RD1 31 0 1
m2 d2 g2 s2 b2 nch	************************
+W=2u	C2 2 30 'Cblock'

```
+L=100n
+NF=10
***S Parameters Extraction Structures***
X1 d g S22 S12 S_PARAM VDCOUT='Vbias_out'
VDCIN='Vbias_in'
X2 g2 d2 S11 S21 S_PARAM VDCOUT='Vbias_in'
VDCIN='Vbias_out'
```

```
R2 30 0 'Rload'
G2 0 4 30 0 2
LP2 2 A2 'LPass'
VINDC2 A2 0 DC 'VDCIN'
**********************
RD2 4 0 1
**********************
.ends
.control
set noaskquit
set filetype=ascii
op
ac lin 251 100MEG 50.1G
settype s-param S22 S12 S11 S21
set appendwrite
wrdata outresult S11 S12 S21 S22
```

The first part of the code provides some basic definitions and the source / bulk DC sources. Vbias_in and Vbias_out represent the gate and drain DC voltage values. The X1 and X2 devices are the sub-circuits used to extract the Sxx and Sxy parameters. Evaluation of the EKV3 MOSFET model with respect to RF characteristics will be shown in the next Section.

5.1. RF EVALUATION OF THE EKV3 MOSFET MODEL

The EKV3 compact model is a charge-based, physical compact model [41] with emphasis on analog/RF circuit design and simulation [42]-[45], including non-quasistatic (NQS) effects [46], [47] and low-frequency [48], [49] and RF noise [50]. An RF compact model should predict the electrical behavior of MOS devices over a large range of frequency, geometry and bias. Former work showed the EKV3 model application on 180nm, 110nm, and 90nm CMOS technologies, mostly in the frequency range up to 20 GHz. In the present work, the EKV3 compact model is presented up to millimetre-wave frequencies, for the TSMC 90nm RF LP process. For this purpose, multi-finger NMOS devices with varying geometries have been measured over a wide range of bias points. As an example, an NMOS device with channel length L=100nm and channel width W=10x2um is presented here. The device is biased at a constant V_{DS}=1V, with V_{GS} being stepped from 0.3V to 0.5V. Fig. 13 clearly shows that all Y-parameters of the device are well matched by the model up to 30 GHz. Measurements are represented by symbols. The EKV3 model from Spectre simulator is indicated by lines, whereas dotted lines correspond to the EKV3 model from ngspice. Since both simulators give identical results, the EKV3 model will be presented by lines for the rest of the text. Y-parameters are a convenient way to interpret a number of MOS parameters, such as transconductance, series resistance and capacitance [42]-[45]. This can be done by using the analytical equations,

$$\begin{aligned}
Y_{11} &= \omega^2 R_g C_{gg}^2 + j\omega C_{gg} \\
Y_{12} &= -\omega^2 R_g C_{gg} C_{gd} - j\omega C_{gd} \\
Y_{21} &= G_m - \omega^2 R_g C_{gg} \left(C_m + C_{gd}\right) - j\omega(C_m + C_{gd} + G_m R_g C_{gg}) \\
Y_{22} &= G_{ds} + \omega^2 R_g (C_{gg} C_{jd} + C_{gg} C_{gd} + C_{gd} C_m) + j\omega(C_{gd} + C_{jd})
\end{aligned} \tag{1}$$

From (1), one can derive the following relationships, which are used to determine device parameters:

$$\begin{gathered}
G_m = \mathrm{Re}(Y_{21})\big|_{\omega\to 0} \quad , \quad G_{ds} = \mathrm{Re}(Y_{22})\big|_{\omega\to 0} \\
C_{gg} = \frac{\mathrm{Im}(Y_{11})}{\omega}, C_{gd} = -\frac{\mathrm{Im}(Y_{12})}{\omega}, C_{jd} = \frac{\mathrm{Im}(Y_{22})}{\omega} - C_{gd}
\end{gathered} \tag{2}$$

The overall gate capacitance C_{gg} extracted from (2) is plotted in Fig. 14 versus the gate-source voltage (V_{GS}), from inversion to accumulation. The current gain (H_{21}) and unilateral power gain (U) are shown in Fig. 15 (a) and 15 (b) with respect to frequency. Unity gain frequency (f_T) and maximum oscillation frequency (f_{max}) of the device can be calculated by extrapolating the curves and their values are approximately equal to 100 GHz and 130 GHz, respectively. These values are also confirmed in the following, where H_{21} and f_T (Fig. 16), and U and f_{max} (Fig. 17) are presented versus inversion coefficient, IC. The latter is the normalized drain current given as $IC=I_D/I_{spec}$, with I_{spec} standing for the specific current, given in (3):

$$\mathrm{I_{spec}} = 2\mathrm{n}\mathrm{U_T^2}\mu_{\mathrm{eff}}\mathrm{C_{ox}}\frac{\mathrm{W_{eff}}}{\mathrm{L_{eff}}} = I_0 \cdot \frac{\mathrm{W_{eff}}}{\mathrm{L_{eff}}}\,. \tag{3}$$

I_{spec} depends on the effective mobility μ_{eff} and slope factor n, while I_0 is called the technology current [41]. The value of IC directly describes the inversion region. Thus when IC is smaller than 0.1 the device is in weak inversion (W.I.). IC values between 0.1 and 10 indicate operation is moderate inversion (M.I.), whereas for IC values greater that 10 the device is in strong inversion (S.I.). Note, for the NMOS device shown here, I_0 = 740 nA [50].

f_T and f_{max} are extracted at 3 GHz using (4).

$$\begin{gathered}
f_T = \frac{f_o}{\mathrm{Im}(Y_{11}/Y_{21})} \\
f_{\max} = f_o\sqrt{\frac{|Y_{21} - Y_{12}|^2}{4[\mathrm{Re}(Y_{11})\,\mathrm{Re}(Y_{22}) - \mathrm{Re}(Y_{12})\,\mathrm{Re}(Y_{21})]}}
\end{gathered} \tag{4}$$

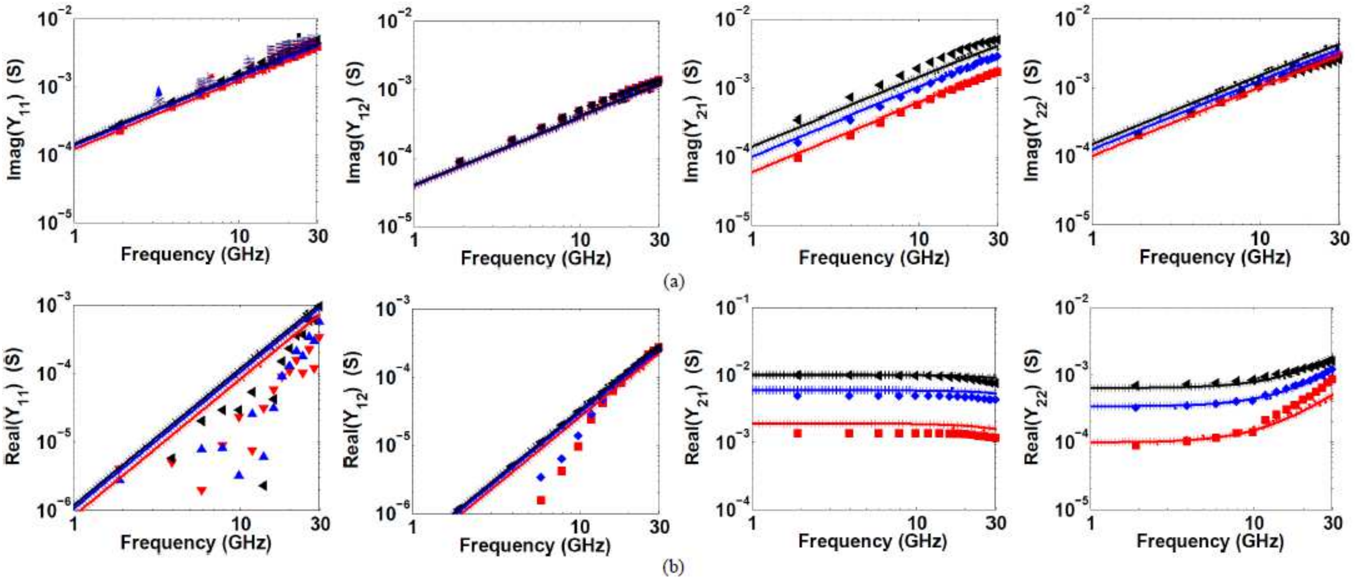


Figure 13. (a) Real and (b) imaginary part of Y-Parameters vs. frequency for a NMOS device of L=100nm and W=10x2um, biased at V_{DS}=1V. Measurements are indicated by symbols for V_{GS}=0.3V (▫), V_{GS}=0.4V (◊) and V_{GS}=0.5V (<). EKV3 model in Spectre is represented by the solid line whereas the dotted line corresponds to EKV3 in ngspice.

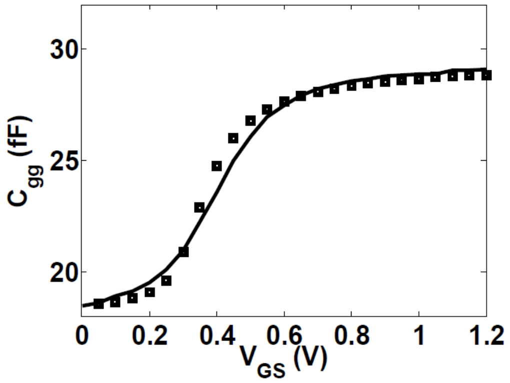


Figure 14. Gate capacitance vs. V_{GS} for an NMOS device of L=100nm and W=40x2um, biased at V_{DS}=0V. Markers: measurements, line: model.

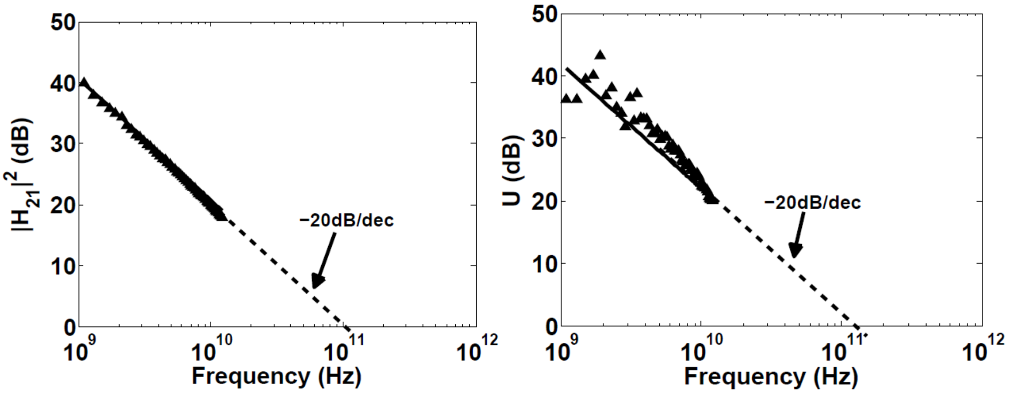


Figure 15 (a) Current gain and (b) unilateral power gain vs. frequency, for an NMOS device of L=100nm and W=40x2um, biased at V_{DS}=V_{GS}=1V. Markers: measurements, line: model.

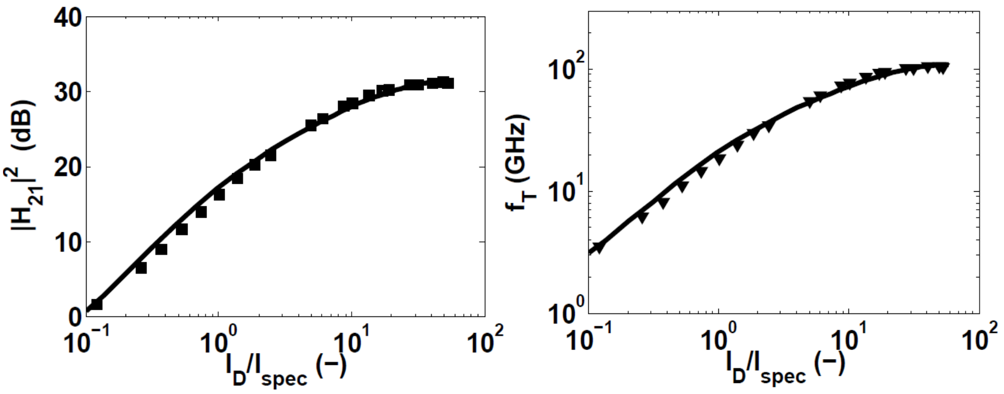


Figure 16. (a) Current gain and (b) unity gain frequency vs. inversion coefficient, for an NMOS device of L=100nm and W=40x2um, biased at V_{DS}=1V, extracted at f=3GHz. Markers: measurements, line: model.

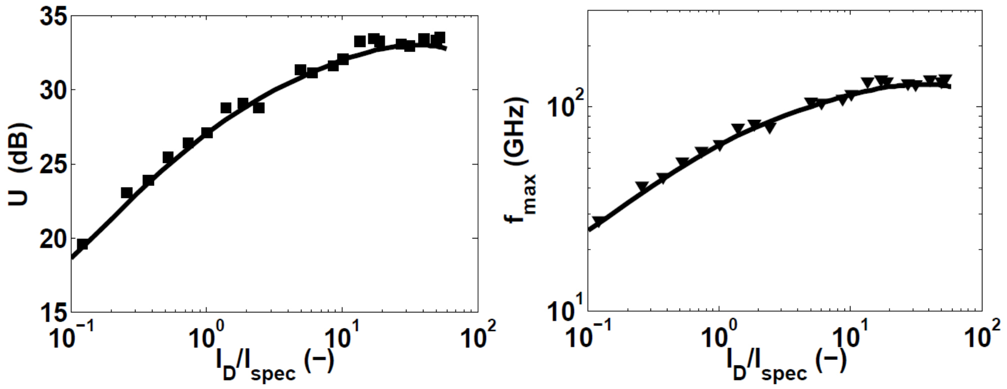


Figure 17 (a) Unilateral power gain and (b) maximum oscillation frequency vs. inversion coefficient for an NMOS device of L=100nm and W=40x2um, biased at V_{DS}=1V, extracted at f=3GHz. Markers: measurements, line: model.

6. CONCLUSIONS

This invited paper summarizes a number of recent MOS-AK open source compact model standardization activities and presents advanced topics on MOSEFT modeling, focusing in particular on the analogue/RF applications. The open source CAD tools: Qucs, QucsStudio and ngspice, which all support Verilog-A as a hardware description language for compact model standardization, have been reviewed. The implementation of the EKV3 Verilog-A compact model in ngspice using ADMS has been discussed. An input file to produce S-parameters in ngspice has been proposed. The EKV3 model ported to ngspice has been found to produce identical results to data obtained with the Spectre circuit simulator up to millimetre wave frequencies. Furthermore, the EKV3 model shows accurate scalability for a 90 nm CMOS technology over large ranges of bias, from weak through moderate to strong inversion, and in frequency ranges up to 30 GHz.

ACKNOWLEDGMENT

A. Antonopoulos acknowledges the support of the Heracleitus II project of the Greek Ministry of Education, co-financed by the European Social Fund (ESF) and Greek national funds, through the operational program "Education and Lifelong Learning", within the National Strategic Reference Framework (NSRF).

REFERENCES

[1] Web site: www.mos-ak.org
[2] White Paper: http://www.mos-ak.org/hvmos_white_paper.php
[3] W. Grabinski, B. Nauwelaers, D. Schreurs (Eds.), “Transistor Level Modeling for Analog/RF IC Design”, Springer-SBM, 2006, ISBN 1-4020-4555-7
[4] Margraf M. et al. Qucs (Quite Universal Circuit Simulator) Version 0.0.17, 2013. Available from: http://qucs.sourceforge.net/index.html,
[5] Margraf M. et al. QucsStudio Version 1.5.0, Available from http://www.mydarc/dd6um/QucsStudio/qucsstudio.html.
[6] Jahn S. and Brinson M.E. Interactive compact device modelling using Qucs equation defined devices, International Journal of Numerical Modelling: Electronic Networks, Devices and Fields, 2008; 21:335-349.
[7] Brinson M.E. and Jahn S., Qucs: A GPL software package for circuit simulation, compact device modelling and circuit macromodelling from DC to RF and beyond, International Journal of Numerical Modelling: Electronic Networks, Devices and Fields, 2009; 22:297-319.
[8] Qucs5. Brinson M.E. and Jahn S., Modelling of high-frequency inductance with Qucs non-linear radio frequency equation defined devices, International Journal of Electronics, 2008, Vol. 96, NO. 3, 307-321.
[9] Coram G.J How to (and not to) write a compact model in Verilog-A, IEEE International Behavioural Modelling and Simulation Conference (BMAS 2004), San José, CA, USA, 2004;97-106.
[10] Gnu Octave. 1992. Available from: http://www.gnu.org/software/Octave/.
[11] Lemaitre L and Gu B. ADMS – a fully customizable Verilog-AMS compiler approach, MOS-AK, Meeting. Montrex, 2006. Available from: http:www.mos-ak.org/montreux/posters/17_Lamaitre_MOS-AK06.pdf.
[12] Touchstone file format Snp. Available from: en.wikipedia.org/wiki/Touchstone_file.
[13] Curtice W.R. A MESFET model for use in the design of GaAs integrated circuits. IEEE Transactions on Microwave Theory and Techniques 1980; 28:448-456.
[14] Accellera. Verilog-AMS Language Reference Manual. Version 2.2, 2004. Available from: http://www.accellera.org.
[15] Brinson Mike E. and Margraf Michael, Verlilog-A compact semiconductor device modelling and compact macromodelling with the QucsStudio-ADMS “Turn-key” modelling

system. International Journal of Microelectronics and computer science, 2012, Vol.3, No. 1,32-40.
[16] Brinson M.E., Advances in Verilog-A compact semiconductor device modelling with Qucs/QucsStudio, , MOS-AK/GSA, Workshop, Munich 2013. Available from:
[17] Web Site http:www.mos-ak.org/munich_2013/presentation/08_Mike_Brinson_MOS-AK_Munich_2013.pdf.
[18] Qucs14,   Qt is a C++ toolkit for cross-platform application development. Copyright © 2012 Nokia Corporation, see qt.nokia.com.
[19] Ngspice Web Site, Available online http://www.ngspice.org
[20] Spice distribution page, Available online http://embedded.eecs.berkeley.edu/pubs/downloads/spice/index.htm
[21] D. A. Gates, Design-Oriented Mixed-Level Circuit and Device Simulation, Technical Report No. UCB/ERL M93/51, 1993. Available online http://www.eecs.berkeley.edu/Pubs/TechRpts/1993/ERL-93-51.pdf
[22] XSPICE web site, Available online http://users.ece.gatech.edu/mrichard/Xspice
[23] F.L. Cox, III, W. B. Kuhn, J. P. Murray, and S. D. Tynor, Code-level modeling in XSPICE, Proceedings IEEE International Symposium on Circuits and Systems, 1992 (ISCAS 92), vol. 2, pp. 871-874, 10-13 May, 1992
[24] Perticaroli S. and Palma F., Nenzi P., A novel PSS analysis implementation reusing the TRAN analysis of Ngspice circuit simulator, presented at 2011 MOS-AK International CM Meeting, 7-8 April 2011, Paris.
[25] Lannutti F. and Perticaroli S., KLU and PSS Analysis Implementation in NGSPICE, presented at 2012 MOS-AK International CM Meeting, 12 December 2012, San Francisco.
[26] Nenzi P., Vogt H., Ngspice Users Manual, Section 17.9, available online at http://ngspice.sourceforge.net/docs/ngspice25-manual.pdf
[27] Vogt, H. ADMS V2.30 version for Ngspice, available online at http://ngspice.sourceforge.net/admshowto.html
[28] T. Davis and E. P. Natarajan, “Algorithm 907: KLU, a direct sparse solver for circuit simulation problems,” ACM Transactions on Mathematical Software, vol. 37, no. 3, pp. 36:2 - 36:17, 2010
[29] Lannutti, F.; Nenzi, P.; Olivieri, N., "KLU sparse direct linear solver implementation into NGSPICE," Mixed Design of Integrated Circuits and Systems (MIXDES), 2012 Proceedings of the 19th International Conference , vol., no., pp.69,73, 24-26 May 2012
[30] W. Grabinski, L. Lemaitre, C. Mcandrew, “Compact Device Modeling Using Verilog-AMS and ADMS”, Electron Technology vol. 35 (2003), No.3, pp. 1-5
[31] W. Grabinski, D. Tomaszewski, L. Lemaitre, and A. Jakubowski, “Standardization of the compact model coding: non-fully depleted SOI MOSFET example”, Journal of Telecommunications and Information Technology, (2005), No.1, pp.135-141
[32] [ADMS_3] MOS-AK Compact Modeling Group
[33] [ADMS_4]L. Lemaitre, C. McAndrew, S. Hamm, “ADMS-automatic device model synthesizer” Custom Integrated Circuits Conference, 2002. Proceedings of the IEEE 2002, 12-15 May 2002 pp.27 – 30

[34] [ADMS_5]L. Lemaitre, G. Coram, C. McAndrew, K. Kundert, “Extensions to Verilog-A to support compact device modeling”, Behavioral Modeling and Simulation, 2003. BMAS 2003. Proceedings of the 2003 International Workshop on 7-8 Oct. 2003 pp.134 – 138

[35] [ADMS_6]ADMS web site: http://mot-adms.sourceforge.net/

[36] [ADMS_7]M. Schroter, “High-Frequency Circuit Design Oriented Compact Bipolar Transistor Modeling with HICUM”, IEICE Transactions on Electronics 2005 E88-C(6), pp.1098-1113;

[37] [ADMS_8]MEXTRAM web site http://sourceforge.net/projects/mextram

[38] EKV32 M.-A. Chalkiadaki, C. Valla, F. Poullet, and M. Bucher, ”Why – and How To – Integrate Verilog-A Compact Models in SPICE Simulators”, Int. Journal of Circuit Theory and Applications, 2013. EKV3Online Available: http://dx.doi.org/10.1002/cta.1833

[39] EKV33 P. Nenzi and H. Vogt, “Ngspice User Manual - Version 25plus”, June 14, 2013. EKV3 Online Available: http://ngspice.sourceforge.net/docs/ngspice-manual.pdf

[40] EKV34 L. Lemaitre, C. McAndrew, and S. Hamm, “ADMS – Automatic Device Model Synthesizer”, IEEE Custom Integrated Circuits Conference (CICC), pp. 27-30, Florida, USA, 2002.

[41] EKV35 J.-M. Sallese, M. Bucher, F. Krummenacher and P. Fazan, “Inversion Charge Linearization in MOSFET Modeling and Rigorous Derivation of the EKV Compact Model”, Solid-State Electronics, vol. 47, no. 4, pp. 677-683, April 2003.

[42] EKV36 C. Enz and Y. Cheng, “MOS transistor modeling for RF IC design”, IEEE J. Solid-State Circuits, vol. 35, no. 2, pp. 186-201, Feb. 2000.

[43] EKV37 S. Yoshitomi, A. Bazigos, and M. Bucher, “The EKV3 Model Parameter Extraction and Characterization of 90nm RF-CMOS Technology”, Proc. 14th Int. Conf. on Mixed Design of Integrated Circuits and Systems (MIXDES), pp. 74-79, Ciechocinek, Poland, June 21-23, 2007.

[44] EKV38 M. Bucher, A. Bazigos, S. Yoshitomi, and N. Itoh, “A Scalable RF IC Design-Oriented MOSFET Model”, Int. Journal of RF and Microwave Computer-Aided Engineering, vol. 18, no. 4, pp. 314-325, July 2008.

[45] EKV39 A. Bazigos, M. Bucher, P. Sakalas, M. Schroter, and W. Kraus, “High-Frequency scalable compact modelling of si RF-CMOS technology”, Physica Status Solidi (c), vol. 5, no. 12, pp. 3681-3685, Dec. 2008.

[46] EKV310 M. Bucher and A. Bazigos, “An Efficient Channel Segmentation Approach for a Large-Signal NQS MOSFET Model”, Solid-State Electronics, vol. 52, no. 2, pp. 275-281, Feb. 2008.

[47] EKV311 M.-A. Chalkiadaki and M. Bucher, “Large-Signal RF Modeling with the EKV3 Model”, J. of Telecommunications and Information Technology (JTIT), pp. 29-33, Jan. 2010. EKV3Online Available: http://www.nit.eu/czasopisma/JTIT/2010/1/29.pdf

[48] EKV312 N. Mavredakis, A. Antonopoulos, and M. Bucher, “Bias Dependence of Low Frequency Noise in 90nm CMOS”, Proc. NSTI-Nanotech/Microtech, Vol. 2, pp. 805-808, Anaheim, California, June 21-25, 2010.

[49] EKV313 N. Mavredakis, A. Antonopoulos, and M. Bucher, “Measurement and Modeling of 1/f Noise in 180 nm NMOS and PMOS Devices”, Proc. 5th European Conf. on Circuits and Systems for Communications (ECCSC), pp. 86-89, Belgrade, Serbia, Nov. 23-25, 2010.

[50] EKV314 A. Antonopoulos, M. Bucher, K. Papathanasiou, N. Makris, R. K. Sharma, P. Sakalas, and M. Schroter, “CMOS RF Noise, Scaling, and Compact Modeling for RFIC Design”, Proc. IEEE Radio Frequency Integrated Circuits Symposium (RFIC), Seattle, Washington, June 2-4, 2013.