INTERNATIONAL JOURNAL OF NUMERICAL MODELLING: ELECTRONIC NETWORKS, DEVICES AND FIELDS
Int. J. Numer. Model. 2015; **00**:1–20
Published online in Wiley InterScience (www.interscience.wiley.com). DOI: 10.1002/jnm

A new approach to compact semiconductor device modelling with Qucs Verilog-A analogue module synthesis

M. E. Brinson[1]*and V. Kuznetsov[2]

[1] *Centre for Communications Technology, London Metropolitan University U.K.*
[2] *Bauman Moscow State Technical University, Kaluga branch, Russia.*

SUMMARY

Since the introduction of SPICE non-linear controlled voltage and current sources they have become a central feature in the interactive development of behavioural device models and circuit macromodels. The current generation of SPICE based open source General Public License circuit simulators, including Qucs, Ngspice and Xyce©, implement a range of mathematical operators and functions for modelling physical phenomena and system performance. The Qucs Equation-Defined Device is an extension of the SPICE style non-linear B type controlled source which adds dynamic charge properties to behavioural sources, allowing for example, voltage and current dependent capacitance to be easily modelled. Following, the standardization of Verilog-A it has become a preferred hardware description language where analogue models are written in a netlist format combined with more general computer programming features for sequencing and controlling model operation. In traditional circuit simulation the generation of a Verilog-A model from a schematic, with embedded non-linear behavioural sources, is not automatic but is normally undertaken manually. This paper introduces a new approach to the generation of Verilog-A compact device models from Qucs circuit schematics using a purpose built analogue module synthesizer. To illustrate the properties and use of the Qucs Verilog-A module synthesiser the text includes a number of semiconductor device modelling examples and in some cases compares their simulation performance with conventional behavioural device models.

Received . . .

KEY WORDS: Qucs; Verilog-A analogue module synthesis; Equation-Defined Devices (EDD); Compact device modelling; circuit simulation

1. INTRODUCTION

In functional compact device modelling SPICE 2 polynomial [1] and SPICE 3 B current sources [2] are often used to express non-linear terminal currents as a function of applied voltages. Today, open source General Public License circuit simulators, including Qucs [3], ngspice [4] and Xyce©[5], are equipped with a range of mathematical operators and functions that are identical or similar to those found in the Verilog-A analogue hardware description language, allowing straight forward modelling of device physical characteristics. Moreover, the SPICE 3 B non-linear voltage and current sources and the Qucs Equation-Defined Device (EDD) [6,12] introduce compact modelling features which are essentially interpretive, promoting simple construction of functional level models. Unfortunately, this simplicity comes at a price, namely that interpretive functional models are often characterized by slow circuit simulation. When simulating the performance of experimental compact device models or small analogue circuits speed may not be of any

*Correspondence to: Centre for Communications Technology, London Metropolitan University, London N7 8DB. E-mail: mbrin72043@yahoo.co.uk.

significance. Often, the reverse is true if a compact device model is to be used in the design of large scale integrated circuits. Ideally, a compact device model should also be encoded in a standardized hardware description language that is suitable for model interchange and easy input to circuit simulators. Modelling device explicit static current characteristics that are a function of terminal voltages is in most instances straightforward with SPICE B sources. In contrast, the reverse is in many instances not true when modelling semiconductor device dynamic charge. This is due to the fact that a significant number of circuit simulators do not have the differential operator $\frac{d}{dt}$ implemented in their function library, making it unavailable for evaluating non-linear algebraic expressions. Simulator SPICE 3f5 is one example where $\frac{d}{dt}$ is missing. Qucs EDD implement an important extension to the SPICE non-linear B controlled source that adds dynamic charge properties to functional sources, allowing for example, non-linear voltage and current dependent capacitance to be easily modelled. In many respects behavioural device modelling with non-linear controlled sources is a precursor to the recently introduced modelling practices that employ analogue hardware description languages. With the standardization of Verilog-A [7] the latest trends in circuit simulation point to the use of Verilog-A as a preferred analogue hardware modelling language for compact model development [8] and simulator independent model interchange. The Qucs non-linear EDD has been designed so that it not only allows straightforward modelling of device static and dynamic properties but encourages the development of compact device models who's constituent parts can be translated into Verilog-A statements without difficulty. Verilog-A analogue modules are normally written in a netlist style hardware description language with more general computer programming features employed for sequencing and controlling model operation. In classical circuit simulation the generation of a Verilog-A hardware description model from a circuit schematic is not automatic but has to be done manually. This paper introduces a new approach to the generation of Verilog-A compact device models from Qucs circuit schematics using an analogue module synthesizer, specifically designed for this task. To illustrate the properties and use of the Qucs Verilog-A module synthesizer a number of semiconductor device modelling examples are included in the paper and in some cases their simulation performance is compared against data for traditional behavioural device models. The presented modelling examples have also been chosen to illustrate how the Qucs Verilog-A synthesizer deals with a number of different physical properties, including shot and flicker noise.

2. AN OVERVIEW OF THE QUCS COMPACT DEVICE MODELLING FACILITIES

Qucs release 0.0.19S provides for the first time a set of modelling and simulation tools built around the well established Qucs Graphical User Interface (GUI), the Qucs qucsator simulation engine, the Qucs EDD and Radio Frequency Equation-Defined Devices (RFEDD) [9], the Qucs/ADMS [10, 16, 17] Verilog-A "turn key" component modelling system (including a new Verilog-A synthesizer), the Ngspice and Xyce ©circuit simulators (including a new SPICE netlist synthesizer) and the Qucs and Octave [11] numerical analysis data visualization software. An overview of the individual sections of the Qucs-0.0.19S software package are presented as block diagrams in Figure 1, where diagram (a) illustrates the overall structure of the modelling system and the data flow from initial physical model specification to the visualisation of simulation data, and diagram (b) shows a simplified set of the modelling blocks and the data flow associated with new model development. Qucs-0.0.19S allows, again for the first time, the development of non-linear compact device models using a combination of Qucs EDD and SPICE B controlled current sources and other Qucs components connected on the same hierarchical level circuit schematic. This important extension to the Qucs modelling capabilities is made possible by the addition of a SPICE netlist synthesizer to the Qucs software package. In the initial stage of a Qucs circuit simulation sequence the SPICE netlist synthesizer generates an Ngspice or a Xyce netlist from the information provided on a model schematic then passes it to either Ngspice or Xyce for simulation. Table I shows the generated SPICE netlist for the Qucs compact model illustrated in Figure 2. The Qucs Verilog-A synthesizer not only directly translates Qucs subcircuit schematics containing basic components, like R and C into their corresponding Verilog-A statements but it also synthesizes the function of more complex

DOI: 10.1002/jnm

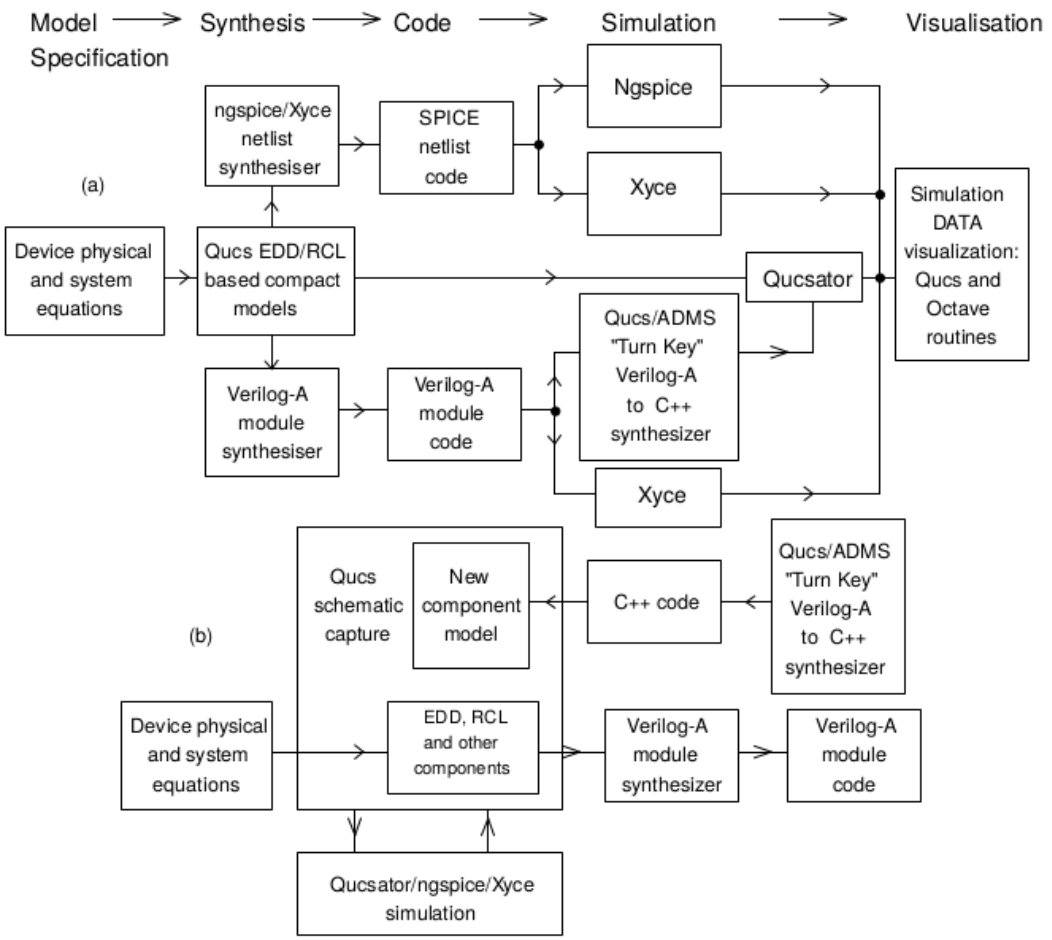

Figure 1. Qucs-0.0.19S block diagram and data flow for the Qucs compact device modelling tool set: (a) Overall structure diagram showing data flow with arrows; (b) Blocks employed in the development of Verilog-A/C++ component models.

components into blocks of in-line Verilog-A analogue statements embedded in a Verilog-A module.

3. INTRODUCTION TO QUCS VERILOG-A ANALOGUE MODULE SYNTHESIS

The Qucs EDD is a behavioural modelling element comprising from one to eight two terminal non-linear devices whose physical characteristics are set by branch current I, voltage V, and stored charge Q. At this stage in Qucs development only the explicit form of the EDD block has been implemented. Quantities I, V and Q are given by the following equations:

$$I_j = I_j(V_j), G_j = \frac{dI_j}{dV_j} \tag{1}$$

$$Q_j = Q_j(V_j, I_j) \tag{2}$$

$$C_j = \frac{dQ_j}{dV_j} = \frac{dQ_j(V_j)}{dV_j} + \frac{dQ_j(I_j)}{dVI_j} \cdot G_j \tag{3}$$

where $1 <= j <= 8$, I_j is the current flowing through branch j, V_j is the voltage across branch j, Q_j is branch j internal stored charge, G_j, and C_j are branch conductance and capacitance

DOI: 10.1002/jnm

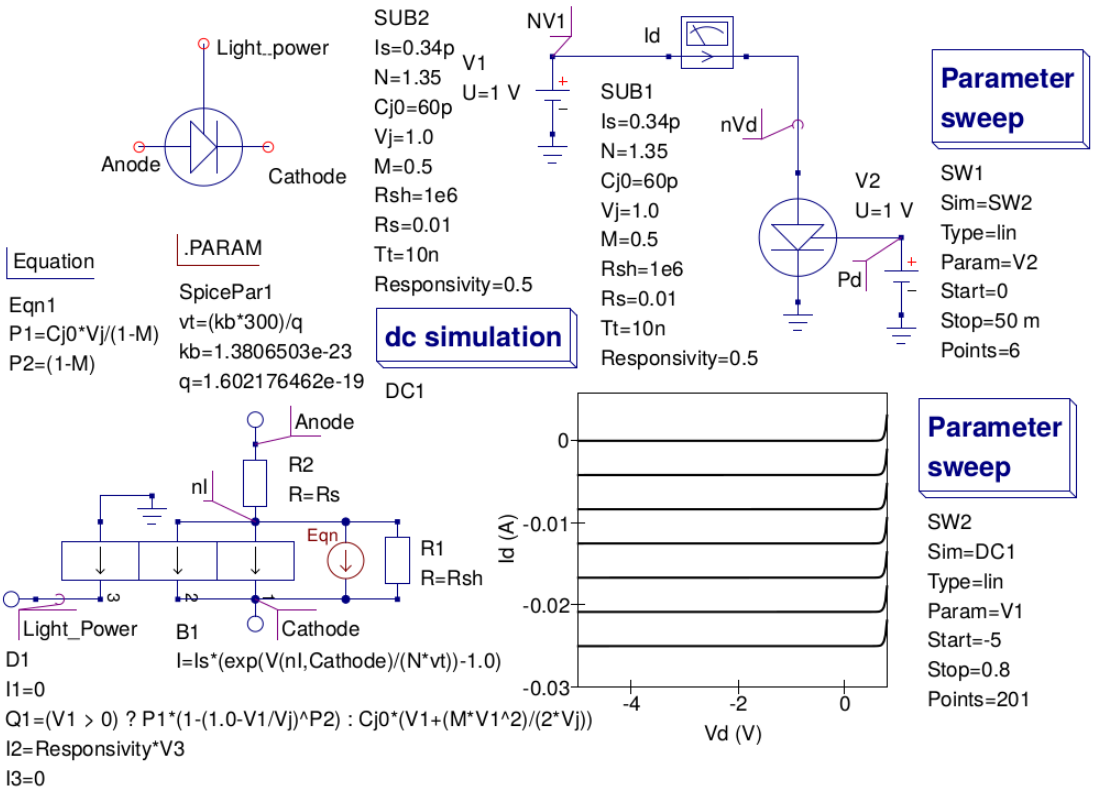

Figure 2. A basic photodiode behavioural model which illustrates Qucs EDD and SPICE B sources combined in the same compact device model: (a) the diode symbol and behavioural model; (b) a test circuit for simulating the photodiode current as a function of reverse DC bias and light power. Here the magnitude of the light power is represented by the value of voltage source V2.

respectively. The Qucs EDD provides a basic set of non-linear circuit elements that are needed for modelling device/circuit static and dynamic properties from two and multi-terminal functional blocks with I_j as a function V_j and Q_j as a function of V_J and I_j, respectively. EDD branch conductance is calculated from the voltage derivative of I_j and the dynamic capacitive current from the time derivative of Q_j. Derivatives are automatically generated in symbolic form by Qucs. Figure 3 shows a diagram of a number of fundamental EDD configurations, plus other blocks, which form the basic building elements used by the Verilog-A synthesizer in the construction of the Qucs standard component models given in Figure 4. Each of the symbols drawn in Figure 3 has a distinct transfer function which relates output current or charge to input voltage or other variables. These are modelled by one, two or multi-line fragments of Verilog-A code. In the past Qucs compact device models had to be translated manually into Verilog-A module code, converted by ADMS to C++, and statically compiled and linked with the main body of the Qucs simulator C++ code each time a new module was added. Qucs-0.0.19S introduces the first release of a new open source GPL Verilog-A synthesis tool for compact device modelling. The Qucs-0.0.19S package includes a basic working version of the synthesis software. It has been released for test purposes. The generated module code can be compiled by ADMS/Qucs to C++ and dynamically linked to Qucs; without having to recompile the main body of the Qucs C++ code each time a new model is added. The overall process is fully automated and does not require that C++ model code be patched to Qucs manually. The flow chart in Figure 5 illustrates a typical set of operations employed when generation and simulating Verilog-A modules. The starting point involves translating a set of mathematical equations that represent the physical properties of a device or circuit into a Qucs subcircuit schematic. Illustrated in Figure 5 is a two input voltage multiplier where $V(Out) = gain \cdot V(In1) \cdot V(In2)$, and the

DOI: 10.1002/jnm

Table I. Example Ngspice synthesized SPICE netlist.

```
* Qucs 0.0.19 VASFig2.sch
.SUBCKT VASFig2 Anode Cathode nLP Is=0.34p N=1.35 Cj0=60p
+ Vj=1.0 M=0.5 Rsh=1e6 Rs=0.01 Tt=10n Responsivity=0.5
.PARAM P1={Cj0*Vj/(1-M)}
.PARAM P2={(1-M)}
.PARAM vt = (kb*300)/q
.PARAM kb = 1.3806503e-23
.PARAM q = 1.602176462e-19
R1 Cathode nI {Rsh}
BD1I0 nI Cathode I=0
GD1Q0 nI Cathode nD1Q0 Cathode 1.0
LD1Q0 nD1Q0 Cathode 1.0
BD1Q0 nD1Q0 Cathode I=-((V(nI,Cathode)> 0) ? P1*(1-(1.0-V(nI,Cathode)/Vj)**P2):
+ Cj0*(V(nI,Cathode)+(M*V(nI,Cathode)**2)/(2*Vj)))
BD1I1 nI Cathode I=Responsivity*(-V(nLP))
BD1I2 0 nLP I=0
B1 nI Cathode I = Is*(exp(V(nI,Cathode)/(N*vt))-1.0)
R2 nI Anode {Rs}
.ENDS
V1 NV1 0 DC 1
VId NV1 nVd DC 0 AC 0
V2 Pd 0 DC 1
XSUB1 nVd 0 Pd VASFig2 Is=0.34P N=1.35 Cj0=60P Vj=1.0 M=0.5
+ Rsh=1E6 Rs=0.01 Tt=10N Responsivity=0.5
.control
...
.endc
.END
```

multiplier output gain is give by parameter *gain*. This simple example has been deliberately chosen to demonstrate the overall Verilog-A module synthesis and testing process because its specification and code are straightforward and easy to understand. Prior to module synthesis it is normal practice to test the operation of a subcircuit, using either qucsator or one of the SPICE simulators. The generation of Qucs Verilog-A modules from subcircuits is fully automated; simply click on the Qucs ”file” drop down menu then click on tab ”Build Verilog-A module from subcircuit” and follow the supplied instructions. The generated Verilog-A code for the multiplier example is listed in Figure 5. Once compiled and linked with the ADMS/GNU C++ software tools Verilog-A modules become available for testing and use in general simulation, again see Figure 5. Further details on the ADMS compile sequence, and more example modules, can be found in references [13,14]. While testing a new model the Ngspice, and indeed Xyce, SPICE netlists can be easily synthesised and their simulation data used to cross check the qucsator output data. For completeness the multiplier Ngspice synthesized netlist has also been included in Figure 5.

4. THE DESIGN OF THE QUCS VERILOG-A ANALOGUE MODULE SYNTHESIZER

The Qucs Verilog-A synthesizer is designed to generate a Verilog-A module from the information specified on a subcircuit schematic, such as physical properties, external and internal connection nodes, subcircuit parameters and internal equation blocks. The template in Figure 6 shows the structure of a synthesized module. The subcircuit external connection nodes (P1, P2 and P3 in Figure 6) together with the subcircuit name specify the module name and interface nodes. The

Prepared using jnmauth.cls
DOI: 10.1002/jnm

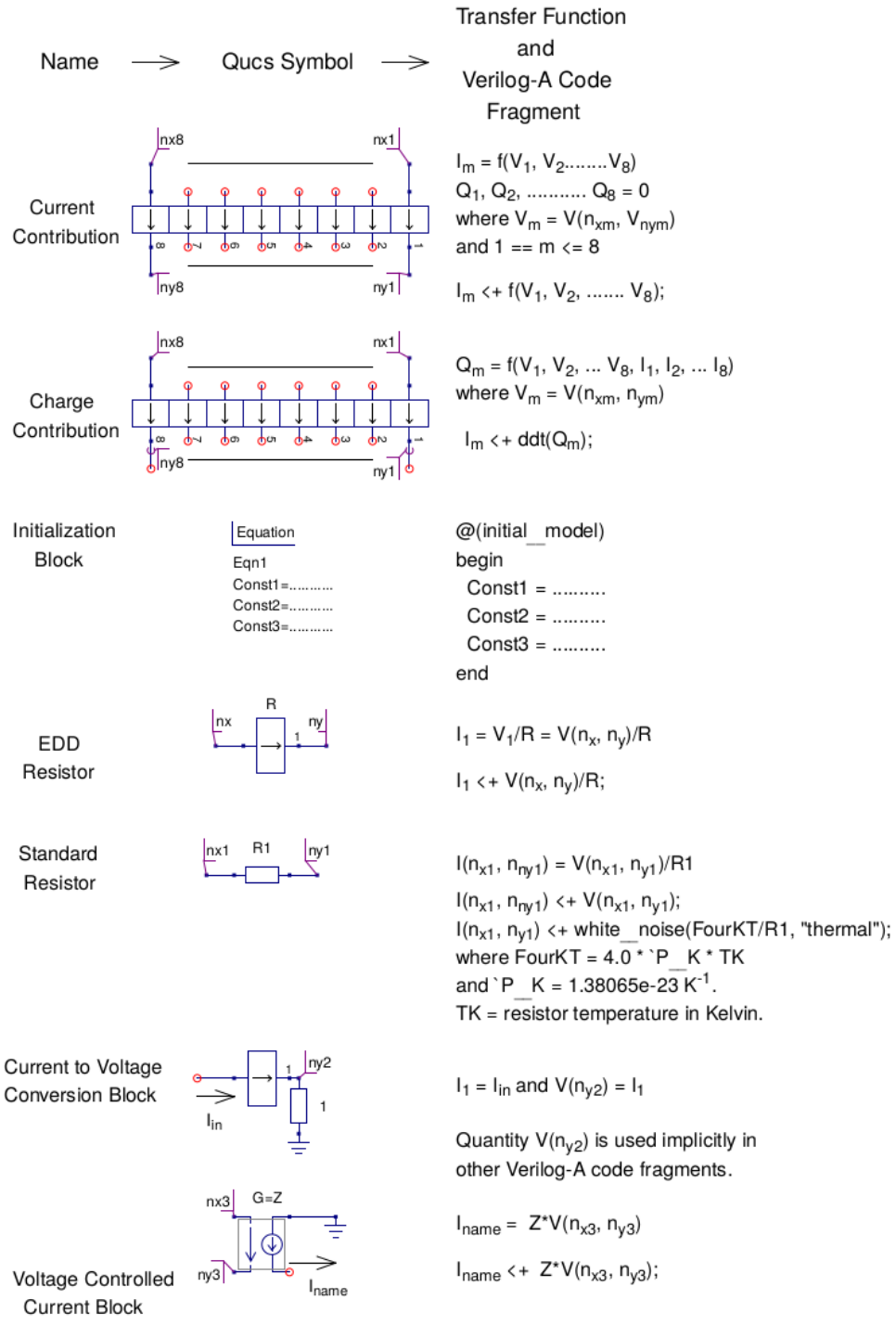

Figure 3. A diagram illustrating the links between Verilog-A in-line code and Qucs EDD models and other fundamental simulation components.

DOI: 10.1002/jnm

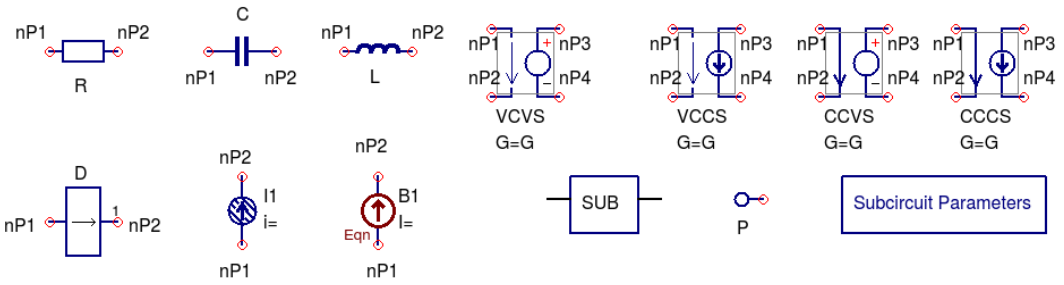

Figure 4. Qucs-0.0.19S and SPICE components used in building device subcircuit models suitable for synthesizing Verilog-A analogue modules.

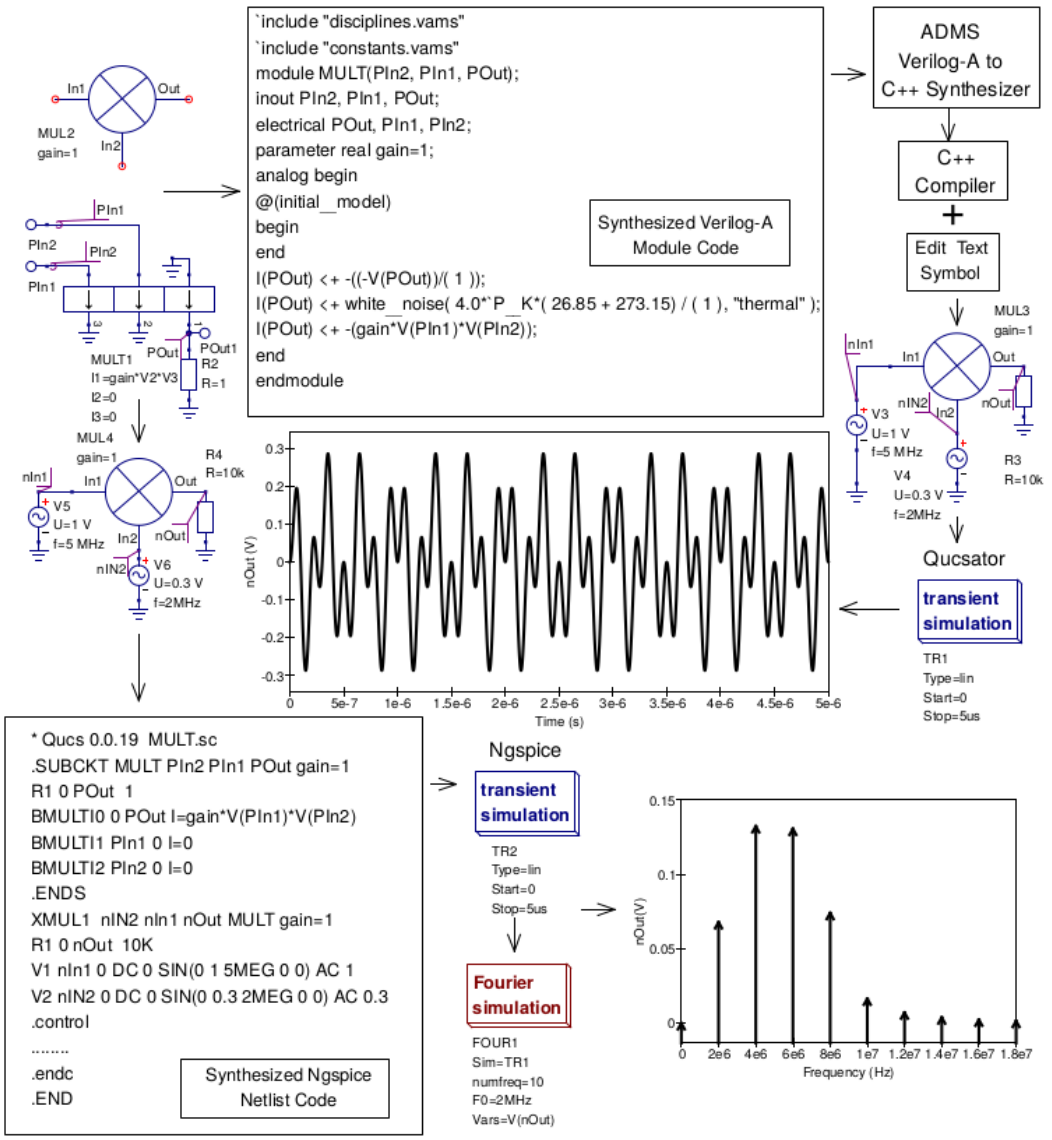

Figure 5. A flow chart showing the basic sequence for synthesizing Verilog-A component modules and test circuit Ngspice netlists from Qucs subcircuits.

DOI: 10.1002/jnm

Table II. Verilog-A current contribution statements for components drawn in Figure 4.

Type	Transfer function	Notes
R	$I(nP1,nP2) < +V(nP1,nP2)/R;$	
	$I(nP1,nP2) < +white\_noise(FOVR,"thermal")$	
	$FOVR = 4.0 \cdot `P\_K \cdot TK/R;$	$`P\_K$=1.3806503e-23 J/K TK=temp in Kelvin
C	$I(np1,nP2) < +ddt(C \cdot V(nP1,nP2));$	Linear C
	$I(np1,nP2) < +ddt(Q);$	Non-linear C
Lx	$I(\_net0Lx) < +ddt(V(\_net0Lx));$	$\_net0Lx$ = Lx internal node
	$I(\_net0Lx) < +V(nP1,nP2);$	Linear/non-linear L
	$I(nP1,nP2) < +V(\_net0Lx)/(Lx + 1e-20);$	
VCVS	$I(nP1,nP2) < +V(nP1,nP2) \cdot 1e-9;$	
	$I(nP4,nP3) < +-(V(nP4,nP3) \cdot 1e3);$	
	$I(nP4,nP3) < +-(V(nP1,nP2) \cdot 1e3 \cdot G);$	G = gain
VCCS	$I(nP1,nP2) < +V(nP1,nP2) \cdot 1e-9;$	
	$I(nP4,nP3) < +V(nP4,nP3) \cdot 1e-9;$	
	$I(nP4,nP3) < +V(nP1,nP2) \cdot G;$	G = conductance S
CCVS	$I(nP1,nP2) < +V(nP1,nP2) \cdot 1e3;$	
	$I(nP4,nP3) < +-(V(nP4,nP3) \cdot 1e3);$	
	$I(nP4,nP3) < +-(V(nP1,nP2) \cdot 1e6 \cdot G);$	G = resistance Ω
CCCS	$I(nP1,nP2) < +V(nP1,nP2) \cdot 1e3;$	
	$I(nP4,nP3) < +V(nP4,n3) \cdot 1e-9;$	
	$I(nP4,nP3) < +V(nP1,nP2) \cdot 1e3 \cdot G;$	G = conductance S
EDD	$I_n < +f(V_1,V_2,....V_8);$	
	$I_m < +ddt(Q_m);$	$Q_m = f(V_1,...V_8,I_1,...I_8)$
Inoise	$I(nP2,nP1) < +white\_noise(i,"shot");$	$i = 1, e = 0, c = 1, a = 0$
	$I(nP2,nP1) < +flicker\_noise(i,e,"flicker");$	$i = Kf, e = Ffe,$ $c = 1, a = 0$
B	$I(nP1,nP2) < +[Equation];$	
	For example, $I(nP1,nP2) < +A + 25 * B - 0.1 * C^4$	where $A = 2, B = 3, C = 4$

synthesizer also generates the required Verilog-A *inout* and *electrical* statements. When used in a module the *electrical* statement has a list of internal nodes appended after the external nodes (n1, n2 and n3 in Figure 6). Similarly, if a subcircuit has one or more parameters these are grouped, with their variable types and default values, in the *parameter* section of a module template specification. Qucs subcircuit *equation* blocks can also contain variables that remain constant during simulation. In such a case these are arranged, in correct order, in the module section headed by Verilog-A statement $@(initial\_model)$, forming a major element in the initialization of a module at the start of a circuit simulation. Variables introduced in a $@(initial\_model)$ block are also declared in Verilog-A *real* or *integer* statements. The main body of a synthesized module consists of one or more Verilog-A current contribution statements. These statements are an essential section of the main analogue block. Each subcircuit component adds at least one DC, capacitive or noise contribution statement to a module. Component contribution statements are synthesized from the Qucs fundamental components shown in Figure 4, using the building elements listed in Figure 3. These are added singly or in combination to specify the Verilog-A in-line transfer functions for each component. Table II gives the details of the individual Verilog-A transfer functions.

DOI: 10.1002/jnm

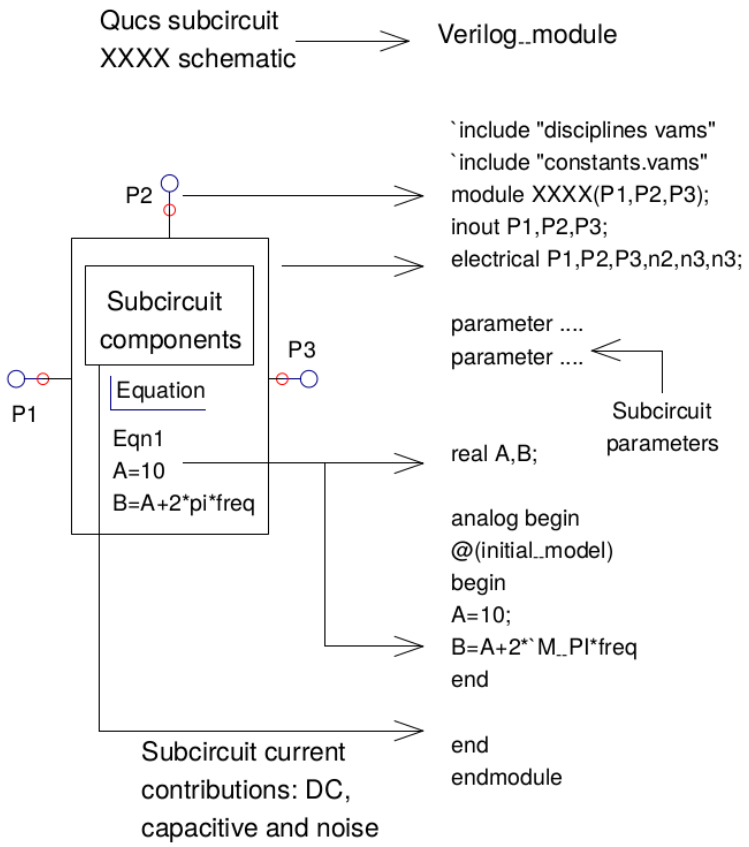

Figure 6. Qucs Verilog-A synthesizer design: structure and data flow paths.

5. ADDING EXTRA COMPONENTS TO THE VERILOG-A SYNTHESIS BUILDING BLOCKS

The Qucs and SPICE components drawn in Figure 4 do not represent a complete set of possible synthesizable components. However, they do form a workable basis for constructing more complex compact device models. The noise models shown in Figure 7 introduce an additional, and practical, building block which is useful in the modelling of semiconductor device noise. In this particular example the additional component models add shot and flicker noise capabilities to the Qucs Verilog-A synthesizer. The data flow represented by the arrows in Figure 7 indicate the steps needed to generate a new synthesized component or circuit model. The first release of the Qucs Verilog-A synthesizer is limited to the synthesis of compact models at the component hierarchical level. This level is enclosed by the dotted box labelled ”Template” in Figure 7. Notice that adding one or more interface pins to either of the shot and flicker templates results in a subcircuit that is suitable for Verilog-A synthesis or direct simulation using qucsator. The ”Template” circuitry enclosed by the dotted boxes can also be copied and pasted onto other component level schematics for use in the construction of new device and circuit models. Once synthesized and compiled with the ADMS ”turn-key” compiler the shot and noise models become available for general simulation or for use as noise components in Qucs subcircuits. The noise sources $I1$ and $I2$ are characterised by current spectral noise density $ISND$ at frequency f that is given by equation

$$ISND(f) = \frac{i}{a + c \cdot f^e} \tag{4}$$

DOI: 10.1002/jnm

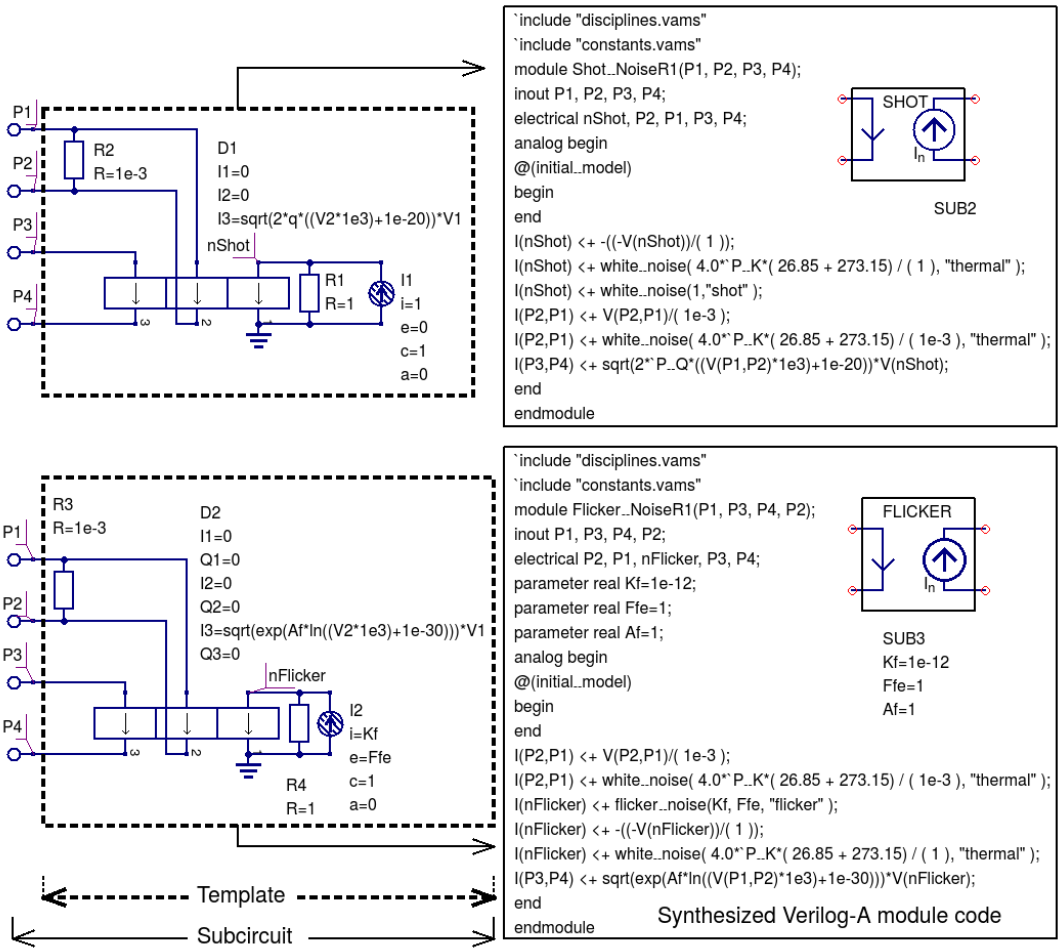

Figure 7. Verilog-A synthesized shot and flicker noise models.

where i is the current spectral density in A^2/Hz at frequency f Hz, as f goes to zero, e is a frequency exponent, c is a frequency coefficient and a is an additive frequency term. $ISND$ parameters i, e, c, a are $i = 1, e = 0, c = 1$ and $a = 0$ for shot noise and $i = Kf, e = Ffe, c = 1$ and $a = 0$ for flicker noise. Parameters Kf and Ffe are a device flicker noise coefficient and a frequency exponent, respectively. The test circuit shown in Figure 8 illustrates a basic test bench for generating and visualizing shot and flicker noise simulation data as a function of noise current. In Figure 7 both noise current sources I_1 and I_2 are multiplied by a scale factor who's value depends on the type of source that is being modelled and the value of the DC current generating the noise. Resistors $R2$ and $R3$ are used to sense this DC current and parameter Af is a flicker noise exponent with a default value of one.

6. A STRUCTURED MODULAR APPROACH TO BUILDING QUCS EDD/VERILOG-A COMPACT DEVICE MODELS

The compact device modelling examples previously introduced in this paper each have a maximum of one EDD and one B source. This is convenient for describing the basic features of the Qucs Verilog-A synthesizer but does not reflect how straightforward, or complex, it is to model device physical properties which require multiple EDD or B type sources. In this section a more complex model of a phototransistor is presented which uses EDD to model electrical and optical device

Prepared using jnmauth.cls
DOI: 10.1002/jnm

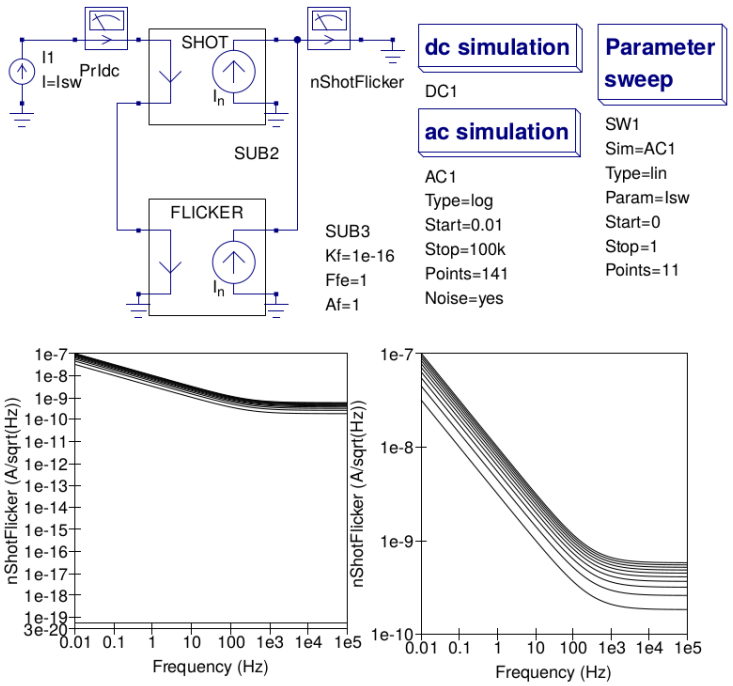

Figure 8. Qucs shot and flicker noise test bench plus typical simulated noise date: DC input current $0 <= Isw <= 1$ A; frequency range $0.01 <= f <= 100k$ Hz.

properties. Qucs-0.0.19S offers a fully integrated set of modelling tools plus access to the Octave numerical software for design calculations and data fitting. The multiple EDD example has been chosen to demonstrate a number of these modelling features, concentrating on non-linear DC device characteristics, S parameter modelling and simulation, and the modelling of photoelectric effects with wavelength response data fitted from measured optical characteristics. Figure 9 gives the details of a phototransistor subcircuit constructed from multiple EDD. EDD $D1$ to $D5$ represent an Ebers-Moll npn BJT equivalent circuit that has been extended to include Early effects and high current forward and reverse bias beta degredation. Diffusion and depletion capacitance and their variation with transistor DC bias conditions are modelled by EDD $D6$ and $D7$. The transistor photelectric properties are modelled by EDD $D8$ and $D9$. The individual EDD non-linear equations used to construct the phototransistor model are also listed, under each EDD D number, in Figure 9. Further DC and capacitance model details, including parameter names and definitions, can be found in Antognetti [15]. EDD $D2$, $D3$, $D4$, $D5$ and $D9$ use one ohm resistors to convert branch currents into node voltages, see for example EDD $D2$ where $D2 : I1$ is converted to the voltage at node Icc. These voltages can be input to other EDD branches as model equation values, provided the connecting branch currents are set to zero, again see EDD $D1 : V4$ ($D1 : I4 = 0$) and $D7 : V2$ ($D7 : I2 = 0$). Figure 10 shows a plot of the phototransitor relative responsivity as a function of incident light wavelength. An equation expressing the relationship between photo current, incident light power, relative responsivity, and values for a fourth degree polynomial fitted to measured photoelectric response data is given in the following equation:

$$I_p = I(nIC, Base) = D8 : I1 = Responsivity \cdot RelativeResponsivity \cdot Power/(100 \cdot Bf)] \tag{5}$$

where

$$RelativeResponsivity = P0 + \lambda \cdot P1 + \lambda^2 \cdot P2 + \lambda^3 \cdot P3 + \lambda^4 \cdot P4 \tag{6}$$

Responsivity is a phototransistor subcircuit parameter, *Power* is the incident light power in watts and $P0$=2.6122e3, $P1$=-1.4893e1 m^{-1}, $P2$=m^{-2}, $P3$= -2.5708e-5 m^{-3}, and $P4$=7.68e-9^{-4}. In order to

DOI: 10.1002/jnm

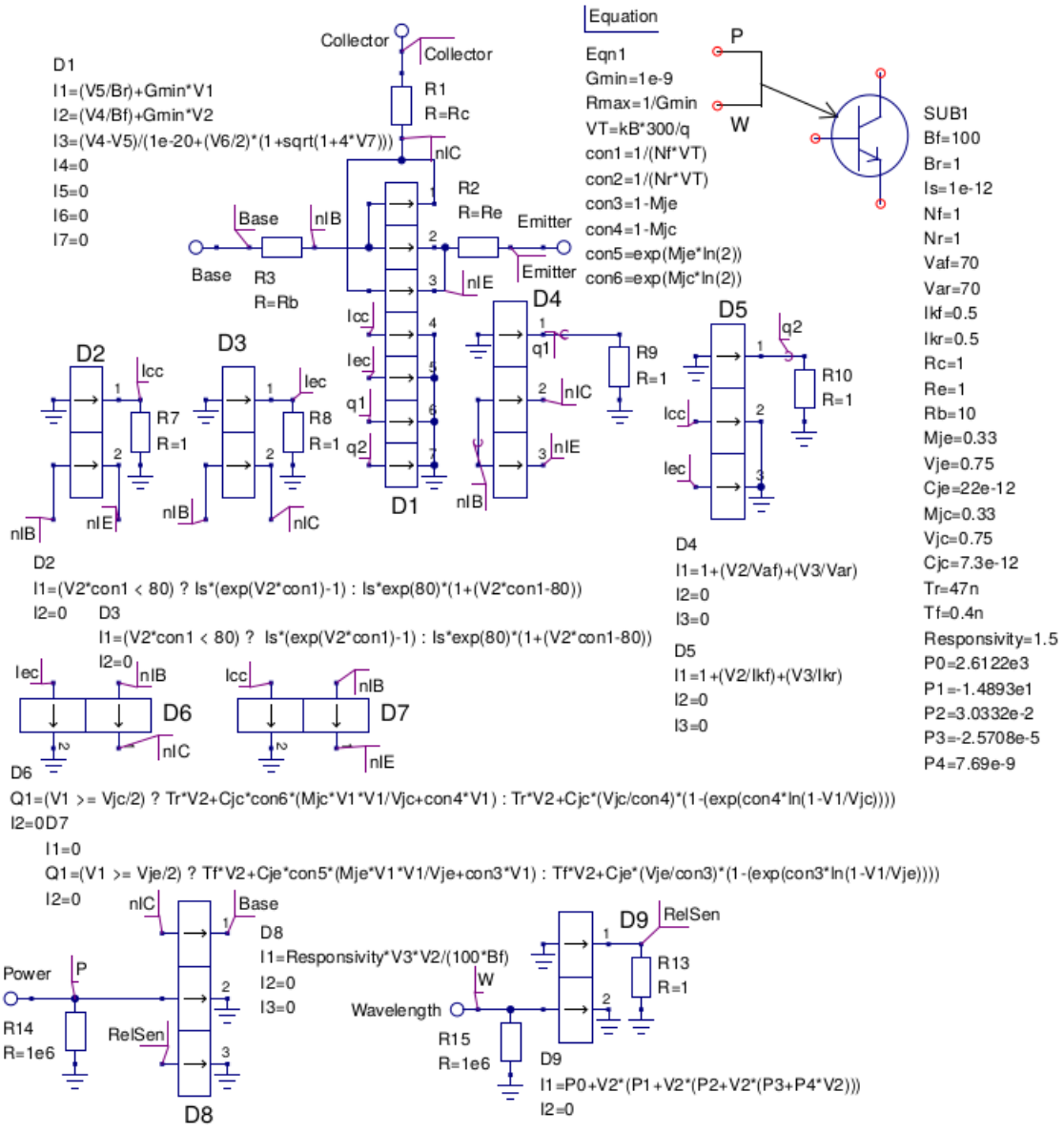

Figure 9. A Qucs non-linear compact device model for an npn phototransistor: the device symbol and EDD phototransistor subcircuit model.

keep the photoelectric model relatively simple temperature effects and device noise properties have not been included in the demonstration model. Table III lists the Verilog-A module code generated by the Qucs synthesizer for the phototransistor multi-EDD subcircuit given in Figure 9. For clarity very long current contribution statements are displayed on more than on line.

7. TESTING THE PERFORMANCE OF THE PHOTOTRANSISTOR SYNTHESIZED VERILOG-A MODULE

Figures 11, 12 and 13 show three test circuits which are suitable for simulating the phototransistor DC characteristics, S parameters and wavelength related properties. In each test circuit the

DOI: 10.1002/jnm

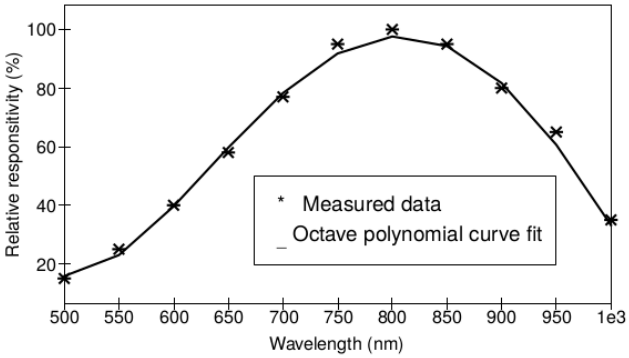

Figure 10. The phototransistor EDD block for modelling photo current as a function of input light power and wavelength: Light power and wavelength values are represented, for modelling purposes, by DC voltages.

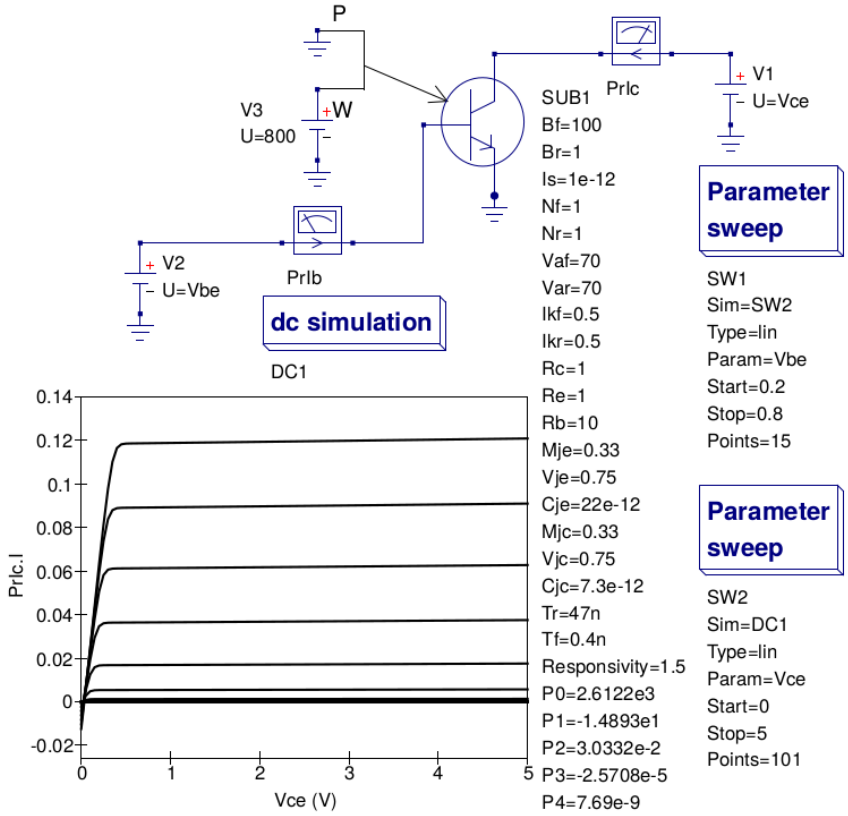

Figure 11. Phototransistor DC test circuit and Verilog-A model collector current plotted as a function of V_{ce} and I_b.

plotted curves were generated from data obtained using the synthesised Verilog-A model. Similar simulation data was recorded with the EDD phototransistor model introduced in Figure 9. In the DC test circuit the input light power is set at zero watts, at a wavelength of 800 nm, which ensures that the BJT DC output characteristics are determined without the influence of an external light source. The phototransistor equivalent parallel capacitance and resistance between the base and emitter terminals (called C_{be} and R_{be} in Figure 12, respectively) are easily extracted from simulated S parameters. Qucs *Equation* block *Eqn1* contains a series of algebraic statements which convert the value of the simulated $S[1,1]$ data into y parameter $y[1,1]$. Values for C_{be} and R_{be} are extracted

DOI: 10.1002/jnm

Table III. Qucs synthesized phototransistor Verilog-A module code.

```
‘include ”disciplines.vams”
‘include ”constants.vams”
module Phototransistor(Emitter, P, W, Collector, Base);
inout Emitter, P, W, Collector, Base;
electrical nIE, Emitter, Iec, Icc, q1, q2, nIB, nIC, Base, P, RelSen, W, Collector;
parameter real Bf=100; parameter real Br=1; parameter real Is=1e-12; parameter real Nf=1;
parameter real Nr=1; parameter real Vaf=70; parameter real Var=70; parameter real Ikf=0.5;
parameter real Ikr=0.5; parameter real Rc=1; parameter real Re=1; parameter real Rb=10;
parameter real Mje=0.33; parameter real Vje=0.75; parameter real Cje=22e-12;
parameter real Mjc=0.33; parameter real Vjc=0.75; parameter real Cjc=7.3e-12;
parameter real Tr=47n; parameter real Tf=0.4n; parameter real Responsivity=1.5;
parameter real P0=2.6122e3; parameter real P1=-1.4893e1; parameter real P2=3.0332e-2;
parameter real P3=-2.5708e-5; parameter real P4=7.69e-9;
real Gmin, Rmax, VT, con1, con2, con3, con4, con5, con6;
analog begin
@(initial_model) begin
Gmin=1e-9; Rmax=1/Gmin; VT=‘P_K*300/‘P_Q; con1=1/(Nf*VT); con2=1/(Nr*VT);
con3=1-Mje; con4=1-Mjc; con5=exp(Mje*ln(2)); con6=exp(Mjc*ln(2));
end
I(nIE,Emitter) <+ V(nIE,Emitter)/( Re ); I(Iec) <+ -((-V(Iec))/( 1 ));
I(Icc) <+ -((-V(Icc))/( 1 )); I(q1) <+ -((-V(q1))/( 1 )); I(q2) <+ -((-V(q2))/( 1 ));
I(q1) <+ -(1+(V(nIB,nIC)/Vaf)+(V(nIB,nIE)/Var)); I(q2) <+ -((V(Icc)/Ikf)+(V(Iec)/Ikr));
I(Icc) <+ -((V(nIB,nIE)*con1<80)?Is*(exp(V(nIB,nIE)*con1)-1):Is*exp(80)*
(1+(V(nIB,nIE)*con1-80)));
I(Iec) <+ -((V(nIB,nIC)*con1<80)?Is*(exp(V(nIB,nIC)*con1)-1):Is*exp(80)*
(1+(V(nIB,nIC)*con1-80)));
I(nIB,nIC) <+ (V(Iec)/Br)+Gmin*V(nIB,nIC); I(nIB,nIE) <+ (V(Icc)/Bf)+Gmin*V(nIB,nIE);
I(nIC,nIE) <+ (V(Icc)-V(Iec))/(1e-20+(V(q1)/2)*(1+sqrt(1+4*V(q2))));
I(nIB,nIC) <+ ddt( (V(nIB,nIC)>=Vjc/2)?Tr*V(Iec)+Cjc*con6*(Mjc*V(nIB,nIC)*
V(nIB,nIC)/Vjc+con4*V(nIB,nIC)):
Tr*V(Iec)+Cjc*(Vjc/con4)*(1-(exp(con4*ln(1-V(nIB,nIC)/Vjc)))) );
I(nIB,nIE) <+ ddt( (V(nIB,nIE)>=Vje/2)?Tf*V(Icc)+Cje*con5*(Mje*V(nIB,nIE)*
V(nIB,nIE)/Vje+con3*V(nIB,nIE)):
Tf*V(Icc)+Cje*(Vje/con3)*(1-(exp(con3*ln(1-V(nIB,nIE)/Vje)))) );
lI(nIC,Base) <+ Responsivity*V(RelSen)*V(P)/(100*Bf);
lI(P) <+ -((-V(P))/( 1e6 )); I(RelSen) <+ -((-V(RelSen))/( 1 ));
lI(RelSen) <+ -(P0+V(W)*(P1+V(W)*(P2+V(W)*(P3+P4*V(W)))));
I(W) <+ -((-V(W))/( 1e6 )); lI(nIC,Collector) <+ V(nIC,Collector)/( Rc );
I(Base,nIB) <+ V(Base,nIB)/( Rb );
end
endmodule
```

from the imaginary and real parts of the $y[1,1]$ data. The extracted values shown plotted in Figure 12 are only accurate to a first order of magnitude because it is assumed in the extraction process that C_{be} consists of the base to emitter and base to collector diode capacitors in parallel and that R_{be} has value that is largely determined, especially in the forward biased region of operation, by the resistance of the base emitter diode plus R_b. During the phototransistor S parameter simulation the incident light power was once again set to zero watts. In contrast both the incident light power and wavelength have values determined by voltage sources $V4$ and $V3$ in Figure 13, respectively. One of the aims of the photoelectric test circuit is to demonstrate that with Qucs synthesized compact models it is possible to represent, and vary by sweeping, non-electrical circuit properties. The test

DOI: 10.1002/jnm

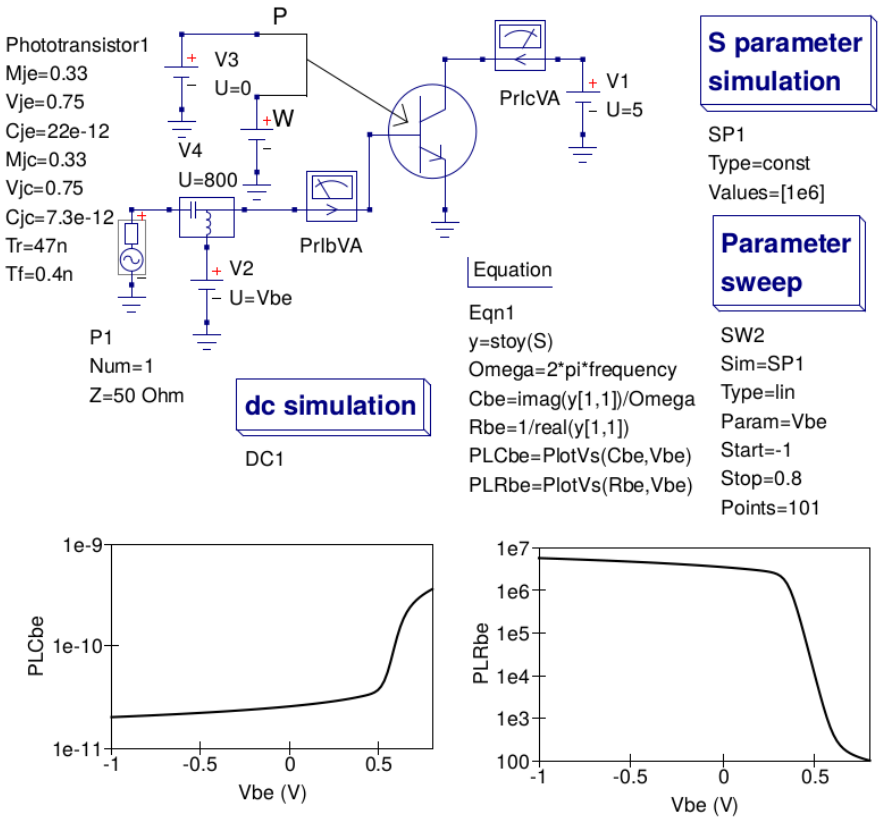

Figure 12. Phototransistor S parameter test circuit and Verilog-A model C_{be} and R_{be} plotted as a function of V_{be} with V_{ce} = 5 Volts.

bench illustrated in Figure 13 has the incident light power held constant at 10 mW and the light wavelength changed from 500 nm to 1000 nm. Plotting the phototransistor collector current against light wavelength results in a curve which represents the relative responsivity of the semiconductor device. The 100k ohm resistor $R1$ in Figure 13 performes a DC self-bias function.

8. OPTIMIZATION OF SYNTHESIZED VERILOG-A MODULE CODE FOR SPEED

Although the Qucs synthesizer generates Verilog-A modules that compile to C++ with ADMS, dynamically link with the main body of simulator C++ code and simulate faster than interactive EDD models, the synthesized code is by no means optimized for simulation speed. Inspection of the synthesized code listed in Table III suggests how module simulation speed can be improved. A central feature in the synthesis process is the use of a one ohm resistor, driven from a current source, to convert a model equation value from a current to a voltage. This step introduces an unwanted by product in that it adds an additional internal node for each current to voltage conversion. Ideally, current contributions only need to be included for those nodes in the connection net linking external nodes. In a high percentage of modules the connection net will include one or more internal nodes. The Verilog-A module header statement lists all the external nodes defined in a module, making it obvious which nodes in the *electrical* statement are internal. In Table III the internal nodes are labelled nIE, Iec, Icc, $q1$, $q2$, nIB, nIC,$RelSen$. The voltages at five of these nodes, Iec, Icc, $q1$, $q2$ and $RelSen$ represent calculated model values and can be eliminated by expressing the equations they represent as real variables, rather than internal voltages. The resulting optimized Verilog-A

Prepared using jnmauth.cls
DOI: 10.1002/jnm

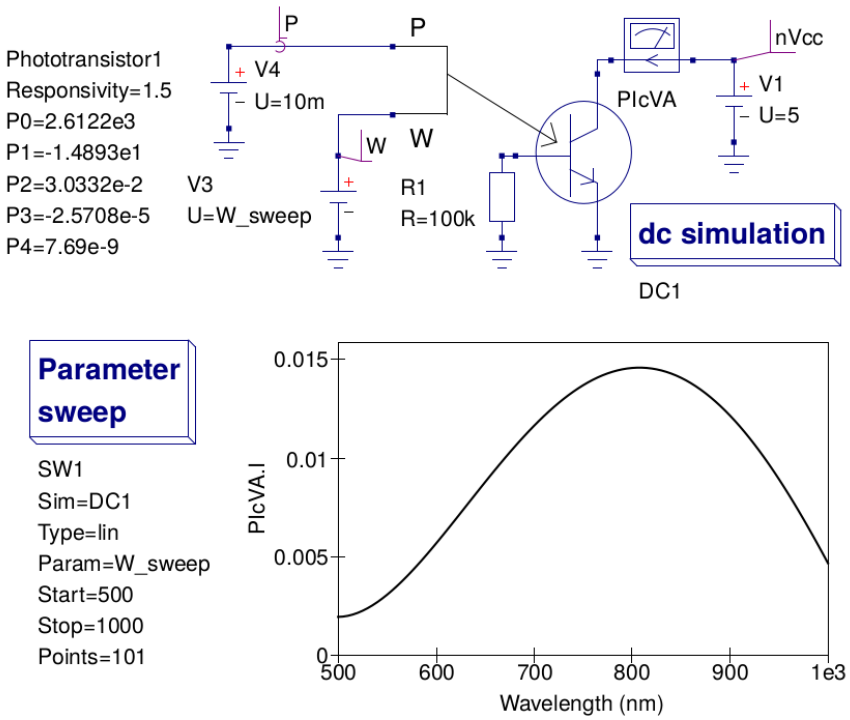

Figure 13. Phototransistor photoelectric test circuit and device collector plotted as a function of illumination light wavelength and V_{ce} = 5 volts.

module code for the phototransistor is given in Table IV. At this stage in the development of the Qucs Verilog-A synthesizer the final optimization process must, when required, be done manually.

9. A COMPARISON OF QUCS EDD BEHAVIOURAL AND VERILOG-A SYNTHESIZED PHOTOTRANSISTOR MODELS IN THE TIME DOMAIN.

The final example in this paper introduces a transient test bench and the simulated performance data for a single stage phototransistor amplifier modelled by the EDD and Verilog-A models described in previous sections. Figure 14 gives details of the test bench circuit and a typical set of simulation data. The phototransistor is self-biased via base resistor $R1$ and collector load $R2$. A light source with a power of 5.5 mW (signal P in Figure 14) and a sinusoidally modulated waveform (signal W in Figure 14) in the range 600 nm to 1000 nm excites the phototransistor. The modulating signal frequency is set at 1k Hz. Due to the shape of the phototransistor relative responsivity curve the simulated amplifier output waveforms not only exhibit amplitude modulation but also clearly indicate that frequency doubling also occurs. The simulation data obtained with the EDD and Verilog-A synthesized models were found to be very similar. However, one important difference was very noticeable, namely that the time taken by the three phototransistor models to complete the transient analysis, using the Qucs simulation engine qucsator, were not the same. In relative terms, with the optimized Verilog-A model as a base line, these were (a) the synthesised Verilog-A model around five times slower and (b) the Qucs EDD model approximately fifteen times slower, confirming as was expected, that the synthesized and optimized Verilog-A models are significantly faster than the interactive EDD version of the phototransistor behavioural model. The difference in simulation time between the synthesized Verilog-A model and its optimized version can to some extent be attributed to the lower number of nodes in the optimized model.

DOI: 10.1002/jnm

Table IV. Qucs synthesized phototransistor Verilog-A module code optimized for speed.

```
‘include ”disciplines.vams”
‘include ”constants.vams”
module PTOpt(Emitter, P, W, Collector, Base);
inout Emitter, P, W, Collector, Base;
electrical nIE, Emitter, nIB, nIC, Base, P, W, Collector;
parameter real Bf=100; parameter real Br=1; parameter real Is=1e-12;
parameter real Nf=1; parameter real Nr=1; parameter real Vaf=70;
parameter real Var=70; parameter real Ikf=0.5; parameter real Ikr=0.5;
parameter real Rc=1; parameter real Re=1;
parameter real Rb=10; parameter real Mje=0.33; parameter real Vje=0.75;
parameter real Cje=22e-12; parameter real Mjc=0.33; parameter real Vjc=0.75;
parameter real Cjc=7.3e-12; parameter real Tr=47n;
parameter real Tf=0.4n; parameter real Responsivity=1.5; parameter real P0=2.6122e3;
parameter real P1=-1.4893e1; parameter real P2=3.0332e-2;
parameter real P3=-2.5708e-5; parameter real P4=7.69e-9;
real Gmin, Rmax, VT, con1, con2, con3, con4, con5, con6;
real Iec, Icc, q1, q2, RelSen;
analog begin
@(initial_model)
begin
Gmin=1e-9; VT=‘P_K*300/‘P_Q; con1=1/(Nf*VT); con2=1/(Nr*VT);
con3=1-Mje; con4=1-Mjc; con5=exp(Mje*ln(2)); con6=exp(Mjc*ln(2));
end
I(nIE,Emitter) <+ V(nIE,Emitter)/( Re );
Icc = ((V(nIB,nIE)*con1<80)?Is*(exp(V(nIB,nIE)*con1)-1):
Is*exp(80)*(1+(V(nIB,nIE)*con1-80)));
Iec = ((V(nIB,nIC)*con1<80)?Is*(exp(V(nIB,nIC)*con1)-1):
Is*exp(80)*(1+(V(nIB,nIC)*con1-80)));
q1 = (1+(V(nIB,nIC)/Vaf)+(V(nIB,nIE)/Var)); q2 = Icc/Ikf+Iec/Ikr;
RelSen = (P0+V(W)*(P1+V(W)*(P2+V(W)*(P3+P4*V(W)))));
I(nIB,nIC) <+ (Iec/Br)+Gmin*V(nIB,nIC); I(nIB,nIE) <+ (Icc/Bf)+Gmin*V(nIB,nIE);
I(nIC,nIE) <+ (Icc-Iec)/((1e-20+q1/2)*(1+sqrt(1+4*q2)));
I(nIB,nIC) <+ ddt( (V(nIB,nIC)>=Vjc/2)?Tr*Iec+Cjc*con6*(Mjc*V(nIB,nIC)*
V(nIB,nIC)/Vjc+con4*V(nIB,nIC)):Tr*Iec+Cjc*(Vjc/con4)*
(1-(exp(con4*ln(1-V(nIB,nIC)/Vjc)))) );
I(nIB,nIE) <+ ddt( (V(nIB,nIE)>=Vje/2)?Tf*Icc+Cje*con5*(Mje*V(nIB,nIE)*
V(nIB,nIE)/Vje+con3*V(nIB,nIE)):Tf*Icc+Cje*(Vje/con3)*
(1-(exp(con3*ln(1-V(nIB,nIE)/Vje)))) );
I(nIC,Base) <+ Responsivity*RelSen*V(P)/(100*Bf);
I(P) <+ -((-V(P))/( 1e6 )); I(W) <+ -((-V(W))/( 1e6 ));
I(nIC,Collector) <+ V(nIC,Collector)/( Rc ); I(Base,nIB) <+ V(Base,nIB)/( Rb );
end
endmodule
```

10. CONCLUSIONS

The synthesis of Verilog-A analogue hardware description language modules from compact model schematics or circuit schematics is a facility which appears not to have been previously implemented in open source GPL circuit simulators. Moreover, to the authors knowledge the availability of Verilog-A synthesizers for use with commercial circuit simulators has not been reported in the published literature. This paper introduces a Qucs Verilog-A synthesizer for the generation of

DOI: 10.1002/jnm

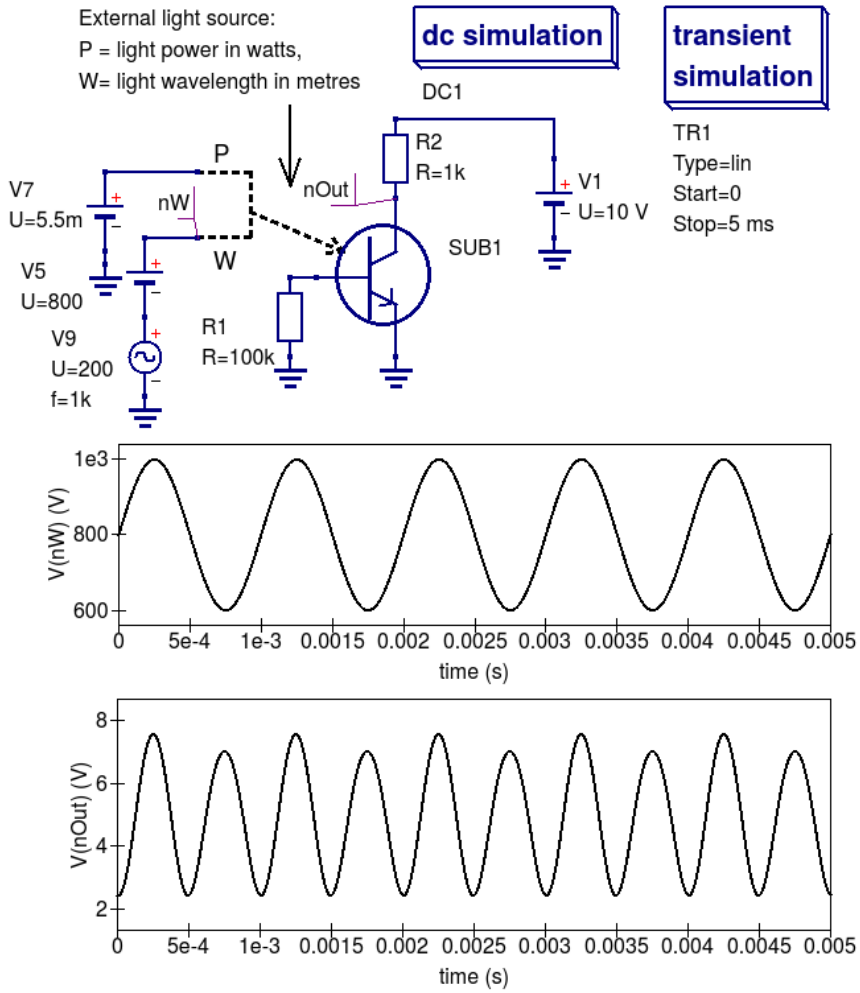

Figure 14. A Qucs single stage phototransistor amplifier test bench and typical transient simulation output voltage and collector current waveforms for an incident power of 5.5 mW and a sinusoidal wavelength modulated light source varying between 600 nm and 1000 nm.

Verilog-A analogue module code from subcircuit schematics. The reported synthesis technique acts as a pre-process to ADMS. The synthesizer generates Verilog-A module code that compiles with ADMS, which in turn outputs C++ code that compiles with a GNU C++ compiler prior to dynamic linking with the main body of Qucs C++ code. The synthesis process is fully automatic and does not require users to manually patch the generated C++ code before object code linking and simulation by Qucs. Although the reported Verilog-A synthesizer was developed for use by Qucs a similar approach to generating Verilog-A analogue module code should be possible with other GPL circuit simulators that have access to ADMS. The Qucs Verilog-A synthesizer introduced in this paper is released under the General Public licence, making it freely available for use by anyone interested in circuit simulation and compact device modelling. The version described in the text is released primarily for test purposes. Although the described Qucs Verilog-A synthesiser is in the early stages of development it's generated modules have been found to simulate significantly faster than conventional non-linear behavioural device models.

REFERENCES

DOI: 10.1002/jnm

1. Newton AR, Pederson DO, Sangiovanni-Vincentelli A. *SPICE Version 2g User's Guide.* Department of Electrical Engineering and Computer Sciences, University of California, Berkeley, CA, 1981.
2. Johnson B, Quarles T, Newton AR, Pederson DO, Sangiovanni-Vincentelli A. *SPICE Version 3f User's Manual.* Department of Electrical Engineering and Computer Sciences, University of California, Berkeley, CA, 1992.
3. Brinson M, Crozier R, Kuznetsov V, Novak C, Roucaries B, Schreuder F, Torri GT, *Qucs (Quite Universl Circuit Simulator).* 2015. Available from http;//qucs.sourceforge.net. Accessed November 2015.
4. Nenzi P, Vogt H, *Ngspice-26 (Next generation SPICE version 26).* 2015. Available from http://ngspice.sourceforge.net. Accessed November 2015.
5. Sandia National Laboratories, US, *Xyce parallel electronic simulator version 6.3.* 2015. Available from http://xyce.sandia.gov. Accessed November 2015.
6. Jahn S, Brinson ME, Interactive compact device modelling using Qucs equation-defined devices, *International Journal of Numerical Modelling: Electronic Networks, Devices and Fields* **2008; 21(5)**, pp. 335-349, DOI: 10.1002/jnm.676.
7. Accellera, *Verilog-AMS Language Reference Manual. Version 2.3.1.* 2009. Available from http://www.accellera.org. Accessed November 2015.
8. Silicon Integration Initiative (Si2), *Compact Model Coalition*. 2015. Available from http;//www.si2.org. Accessed November 2015.
9. Brinson ME, Jahn S, Modelling high-frequency inductance with Qucs non-linear radio frequency equation defined devices, *International Journal of Electronics* **2009; 96(3)**, pp. 307-321, DOI:10.1080/00207210802640603.
10. Lemaitre L, Gu B, *ADMS - a fully customizable Verilog-AMS compiler approach*. MOS-AK, Meeting, Montreux, 2006. Available from http://www.mos-ak.org/montreux/posters/17_Lemaitre_MOS_AK06.pdf. Accessed November 2015.
11. Eaton JW, *GNU Octave. Version 4.0*. 2015. Available from https:www.gnu.org/software/octave/. Accessed November 2015.
12. Brinson ME, 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(4)**, pp 297-319, DOI: 10.1002/jnm.702.
13. Brinson M, Crozier R, Novak C, Roucaries B, Schreuder F, Torri GT, *Building a second generation Qucs GPL circuit simulator: package structure, simulation features and compact device modelling capabilities.* MOS-AK Workshop, London, 28 March 2014. Available from http://www.mos-ak.org/london_2014/presentations/09_Mike_Brinson_MOS-AK_London_2014.pdf. Accessed November 2015.
14. Brinson M, Crozier R, Kuznetsov V, Novak C, Roucaries B, Schreuder F, Torri GT. *Qucs: An introduction to the new simulation and compact device modelling features implemented in release 0.0.19/0.0.19Src2 of the popular GPL circuit simulator*. MOS-AK ESSDERC/ESSCIRC Workshop, 18 September Graz, 2015. http://www.mos-ak.org/graz_2015/presentations/T_5_Brinson_MOS-AK_Graz_2015.pdf. Accessed November 2015.
15. Anognetti P, Massobrio G, *Semiconductor Device Modeling with SPICE.* New York: McGraw-Hill Inc, 1988.
16. Lemaitre L, Grabinski W, McAndrew C, Compact device modelling using Verilog-AMS and AMS. *Electron Technology (Internet Journal)*, June 6 2003, pp 1-5. Available from http://www.ite.waw.pl/etij/pdf/35-03p.pdf. Accessed November 2015.
17. Lemaitre L, McAndrew CM, Hamm S. *Automatic Device Model Synthesis*. CICC, 2002, Florida, USA.

AUTHORS' BIOGRAPHIES

Mike Brinson received a first class honours BSc degree in the Physics and Technology of Electronics from the United Kingdom Council for National Academic Awards in 1965, and a PhD in Solid State Physics from London University in 1968. Since 1968 Dr. Brinson has held academic posts in Electronics and Computer Science. From 1997 till 2000 he was a visiting Professor of Analogue Microelectronics at Hochschule, Breman, Germany. Currently, he is a visiting Professor at the Centre for Communication Technology Research, London Metropolitan University, UK. He is a Chartered Engineer (CEng) and a Fellow of the Institution of Engineering and Technology (FIET), a Chartered Physicist (CPhys), and a member of the Institute of Physics (MInstP). Prof. Brinson joined the Qucs project development team in 2006, specializing in device and circuit modelling, testing and document preparation.

Vadim Kuznetsov was born in Kaluga, Russia in 1988. He received a dipl. engineer degree from Moscow Bauman State University (BMSTU) in 2010. He received a PhD degree from Higher school of Economics in 2014. He is Associate Professor of Electronic Engineering in the department of Kaluga Branch of BMSTU. His research field is electrostatic discharge simulation methods. His field of interest is electronic design automation (EDA) CAD open-source software development. He is core member of Qucs circuit simulator development team.

DOI: 10.1002/jnm