International Journal of Electronics
Vol. 00, No. 00, Novmber 2010, 1–14

RESEARCH ARTICLE

Adaptive subcircuits and compact Verilog-A macromodels as integrated design and analysis blocks in Qucs circuit simulation

M.E. Brinson[a]* and H. Nabijou[a]

[a] *Centre for Communications Technology, London Metropolitan University, London, UK*

(*Received 21 June 2010 ; final version received 16 November 2010*)

It is over twenty years since subcircuits were first used in circuit simulation as a means for partitioning parts of an electronic system into identifiable and reusable units. Throughout this period the basic structure of the subcircuit has remained essentially the same. Current developments in open source circuit simulator technology have demonstrated that by merging circuit design algorithms with subcircuit component calculations a new dimension is added to the role played by subcircuits in circuit simulation. This paper proposes the embedding of equation-defined algorithms within a subcircuit for circuit design purposes prior to circuit simulation. To illustrate the structure and operation of the new subcircuit extension the design and simulation of an active band-pass filter is described and its performance reported with data obtained from tests using the"Quite universal circuit simulator" (Qucs).

Keywords: adaptive subcircuits; equation-defined devices; macromodels; Verilog-A; Qucs

1. Introduction

The term "subcircuit" is the name given to a circuit partition that forms part of a hierarchical representation of an electronic system. In most cases it represents either a section of a circuit that performs a well defined function, such as amplification, or it models an integrated circuit. By designing an electronic system using a mixture of fundamental components and subcircuits, it becomes possible to encode a design with a simplified circuit "netlist" that can be easily read by a circuit simulator, as input data. In this context a subcircuit is essentially a wrapper that encloses a section of a circuit. Pins attached to the wrapper connect specific internal subcircuit nodes to other parts of an electronic system. In many respects there is a strong analogy between hardware subcircuits and software subroutines, encouraging the development of modular simulation models (Brinson and Faulkner 1994). The syntax of typical first generation subcircuits, especially the form made popular by the SPICE 2g6 (Vladimirescu, Kaihe Zang, Newton, Pederson and Sangiovanni-Vincentelli 1981) and 3f5 (Newton, Pederson and Sangiovanni-Vincentelli 1992) circuit simulators, only allowed signal transmission from and to external circuits via interface nodes. Except for voltage and current signals no other data were allowed to be passed to or from a subcircuit. It is also worth noting that in most instances subcircuit component values were only defined by numeric quantities. In particular, facilities for calculating component values from algebraic expressions were not implemented. Today, this is considered an unnecessary limitation which

*Corresponding author. Email: mbrin72043@yahoo.co.uk

ISSN: 0020-7217 print/ISSN 1362-3060 online

DOI: 10.1080/0020721YYxxxxxxxx
http://www.informaworld.com

has been addressed through the addition of simulator netlist extensions to the basic subcircuit syntax (MicroSim Corporation 1984). However, at this time there appears to be no universal standardization of the format for passing subcircuit parameters, although the use of a simulator preprocessor for determining subcircuit component values from passed parameters has become a popular technique among current circuit simulators (Schmidt 2010). Parameters have also been successfully employed as input data to simulator specific extensions, such as Laplace (MicroSim Corporation 1984) and control system (Margraf et al. 2003) transfer function blocks. Recent trends in circuit simulation indicate a growing interest in subcircuit parameters and similar simulator technology improvements (Nagal and McAndrew 2009, Allabouteda 2010). This paper introduces an extension to conventional circuit simulator subcircuit technology which merges circuit design with subcircuit functionality, adding a new dimension to the role played by subcircuits in circuit simulation. It also contrasts the performance of extended subcircuits with compact Verilog-A macromodels (Brinson and Jahn 2009). A number of example subcircuit and Verilog-A macromodels are described in the text. These are based on a well known active filter circuit. Extended subcircuit performance testing was undertaken using the GNU General Public Licence (GPL) Qucs (Quite universal circuit simulator) package (Margraf et al. 2003). The reported simulation data illustrates the power of the proposed combined circuit design/subcircuit simulation strategy.

2. Adding design procedures to subcircuits

Subcircuits are important circuit simulator building blocks. They allow a group of components to be combined as an identifiable and reusable entity. Unfortunately, both the SPICE 2g6 and 3f5 simulators, and indeed more recent simulators developed from the SPICE code (Nenzi 2005), do not directly support subcircuit parameters but require a preprocessor to convert a modified form of the conventional SPICE netlist to either the 2g6 format or the 3f5 format, prior to simulation. The ability to pass parameters to a subcircuit is important because it allows subcircuit component values to be calculated from external data. This process presupposes that a circuit simulator includes software code that can evaluate algebraic equations to give the numerical values of components. Often such implemented code is limited to determining variable values embedded as single line expressions associated with individual components. This is an unwanted limitation, making it difficult to develop universal subcircuits where a common template, with different parameter lists, is used to generate subcircuits that have identical internal circuits but different component values. A second and equally important limitation results from the fact that without a facility to evaluate sets of algebraic equations (that include constants, variables, mathematical operators and functions plus calculation flow control statements) it is difficult, if not impossible, to embed and action circuit design procedures within a subcircuit. Design procedures normally require a number of calculations to be performed sequentially, operating on the subcircuit parameters as a starting point. Each of the individual design stages may require the calculation of intermediate variables, the use of constant terms, and the evaluation of a range of scientific functions linked together via mathematical operators. The Qucs circuit simulator has an advanced modelling feature, called an *Equation* block (Jahn and Brinson 2008) that allows multi-line mathematical expressions to be added to any hierarchical level of a circuit being simulated. More than one Qucs *Equation* block per hierarchical level is allowed. Prior to simulation all Qucs *Equation* blocks at a given hierarchy are combined to form one master block. This

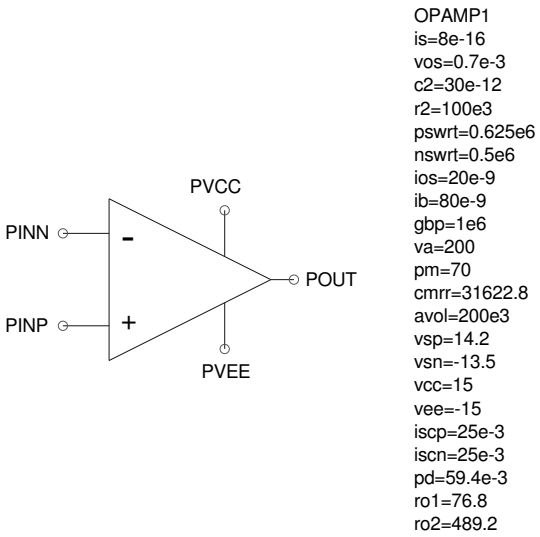


Figure 1. Qucs subcircuit symbol and parameter listing for the Boyle $\mu A741$ macromodel.

has two consequences: firstly the order when entering equations in a Qucs *Equation* block becomes unimportant and secondly an error is flagged if more than one equation for calculating a specific variable occurs in a set of equations. Qucs *Equation* block variables remain fixed in value throughout simulation and as a result cannot be functions of circuit quantities such as node voltages or component currents. Qucs *Equation* blocks are evaluated prior to the start of circuit simulation and again following completion of a simulation sequence. During the first evaluation only those equations which are not functions of circuit output data variables are determined. Any missing data is reported as an error and simulation is halted to allow omissions to be corrected. The previously described process for determining subcircuit component values allows a complete circuit design procedure to be performed before the start of a simulation sequence. One result of this addition to subcircuit functionality is that it allows universal macromodels of integrated circuits to be constructed. Consider the example shown in Figure 1 and Figure 2. This illustrates the original version of the well known Boyle macromodel for a general purpose operational amplifier (Boyle, Pederson, Cohn, and Solomon 1974). Notice that in the schematic diagram given in Figure 2 the majority of the component values are written as algebraic names and not numerical values. Qucs *Equation* block Eqn1 list the equations for calculating the Boyle macromodel component values from the constants and temporary intermediate values given in Eqn1, and the subcircuit parameters listed in Figure 1 and defined in Table 1. Hence, unlike the conventional SPICE subcircuit the proposed extended subcircuit allows macromodels of different operational amplifiers to be generated simply by changing the parameter list attached to the amplifier symbol schematic.

3. Embedding design equations in top level circuit schematics

Qucs *Equation* blocks (or their equivalent) are crucial when adding circuit design algorithms to computer aided circuit simulation software employing subcircuits. This section and later sections of the text introduce a number of approaches for embedding design procedures in Qucs simulation schematics. The techniques introduced are not the only ones possible. Indeed, the flexibility of the equation centred technology provided by Qucs encourages experimentation on the part of the simu-

Table 1. $\mu A741$ Boyle operational amplifier macromodel parameters.

Name	Description	Unit	Default
is	Transistor T1 leakage current	A	8e-16
vos	Input voltage offset	V	7e-4
c2	Frequency compensation capacitor	F	30e-12
pswrt	Positive slew rate	V^{-1}	0.6e6
nswrt	Negative slew rate	V^{-1}	0.5e6
ios	Input offset current	A	20e-9
ib	Input bias current	A	80e-9
r2	Internal gain resistor	Ω	100e3
gbp	Differential gain bandwidth product	Hz	1e6
pm	Phase margin	Deg.	70
cmrr	Common-mode rejection ratio		31622.8
avol	Differntial gain at DC		200e3
vsp	Positive saturation output voltage	V	14.2[a]
vsn	Negative saturation output voltage	V	-13.5[a]
vcc	Positive power-supply voltage	V	15
vee	Negative power-supply voltage	V	-15
iscp	Positive short circuit output current	A	25e-3
iscn	Negative short circuit output current	A	25e-3
pd	Typical power dissipation	W	59.4e-3
ro1	High frequency AC outout resistance	Ω	76.8
ro2	DC output resistance	Ω	498.2

[a] These values assume that vcc=15V and vee=-15V.

lator user. Illustrated in Figure 3 is the popular Delyiannis-Friend active band-pass filter (Delyiannis 1968, Friend 1970). The basic component calculation equations for this circuit are given by Equation (1).

$$f_0 = \frac{1}{2 \cdot \pi \cdot C \cdot \sqrt{R_p \cdot R_3}},\ h_0 = \frac{R_3}{2 \cdot R_1}, bw = \frac{1}{\pi \cdot R_3 \cdot C},\ Q = \frac{f_0}{bw} = \frac{1}{2 \cdot \sqrt{\frac{R_3}{R_p}}} \quad (1)$$

Where, $C1 = C2 = C$, $1/R_p = 1/R_1 + 1/R_2$, f_0 is the filter centre frequency in Hz, bw is the filter bandwidth in Hz, h_0 is the magnitude of the filter voltage gain, and Q is a quality factor. Choosing a suitable value for R_3 allows values for R_1, R_2 and C to be found from the transposed design equations given in Qucs *Equation* block Eqn1. Qucs *Equation* block Eqn3 gives values for the required filter attributes, while Eqn2 lists expressions for extracting the filter gain and phase as functions of frequency. In Figure 3 the filter bandwidth bw is equated to variable *bandwidth* and swept through a range of values. Similarly, Figure 4 shows a set of typical two and three dimension simulation waveform plots. By combining design calculations with simulation not only is it possible to determine component values, and indeed observe the effects of component changes on circuit performance, but it is also possible to generate useful design nomographs. The data given in Figure 5 makes possible the estimation of filter component values by a simple process of interpolation, merging design with circuit simulation.

4. Embedding circuit design procedures in hierarchical subcircuits

The fundamental concepts previously outlined allow component values to be calculated using algebraic equations and circuit parameters attached to a Qucs schematic. In practice it is often more convenient to embed design algorithms within the body of a subcircuit, ensuring that the details of a design process do not obscure its use when constructing a top level Qucs circuit schematic. This approach is similar to designating parts of the data and code within a C++ function as

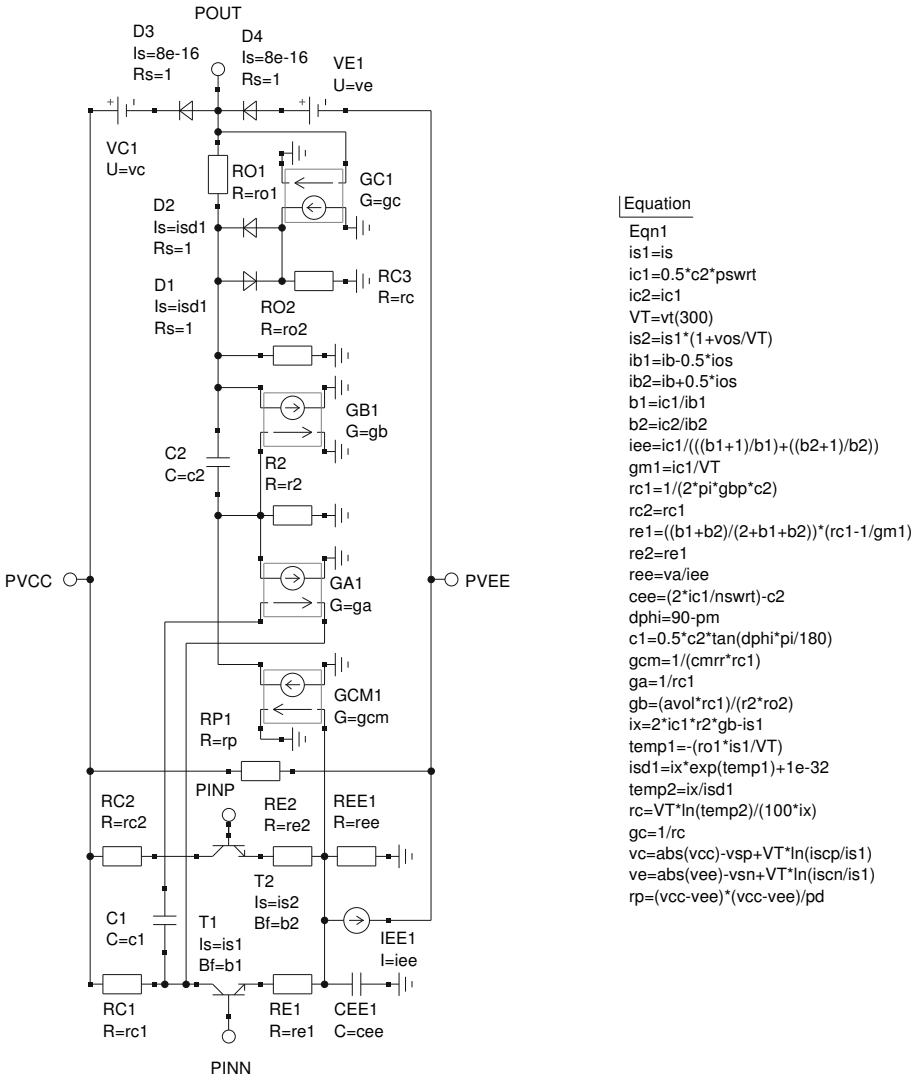


Figure 2. Qucs subcircuit schematic and *Equation* block listing (Eqn1) for the Boyle $\mu A741$ macromodel.

private. Figure 6 shows a symbol for a band-pass filter subcircuit plus the body of the subcircuit and a Qucs *Equation* block (Eqn1) which lists the filter design algorithm. In this example the design algorithm consists of a set of equations for calculating circuit normalised component values (with $\omega_0 = 2 \cdot \pi \cdot f0 = 1$) and for calculating absolute scaled numerical values. The frequency and amplitude scaling factors Kf and Km determine the actual numerical values of the subcircuit components based on parameter data for the filter centre frequency $f0$, filter bandwidth bw and a numerical value for capacitor C. Although the band-pass filter subcircuit only requires three input parameters it can be used to accurately model filters over the frequency band where the $\mu A741$ operational amplifier operates satisfactorily. The ability of a subcircuit model to adapt its component values in response to design parameters fed to the embedded design algorithm illustrates the power of the proposed subcircuit design extension. Figure 7 shows a band-pass filter test example that demonstrates how the filter changes its voltage transfer characteristics as the filter centre frequency is swept through a series of values. The voltage gain waveform given in Figure 7 also clearly shows the effect of the $\mu A741$ limited gain bandwidth product (gbp=1e6 Hz) on the simulated voltage transfer characteristic at $f0$: as the filter centre frequency increases the simulated value of $f0$ becomes

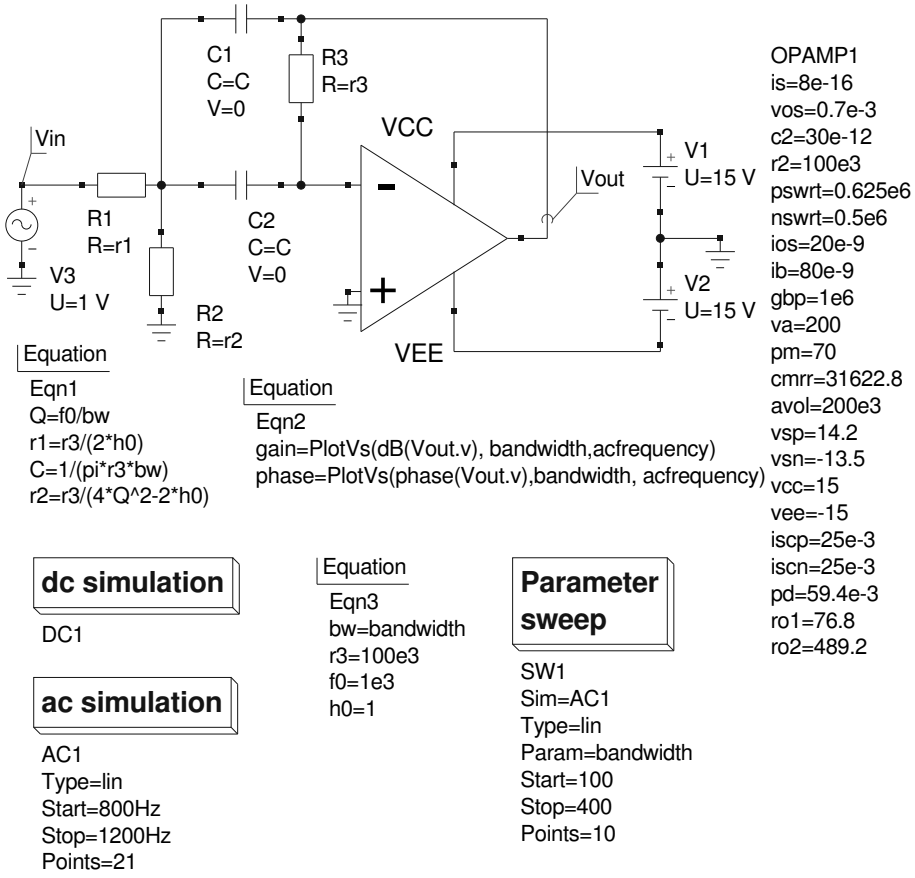


Figure 3. Delyiannis-Friend active band-pass filter circuit

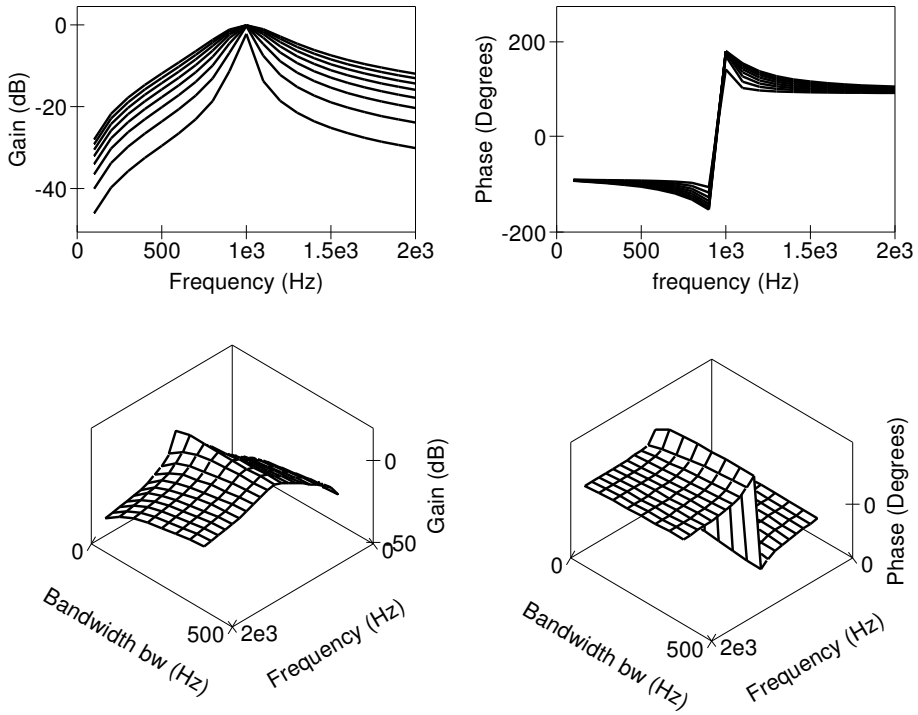
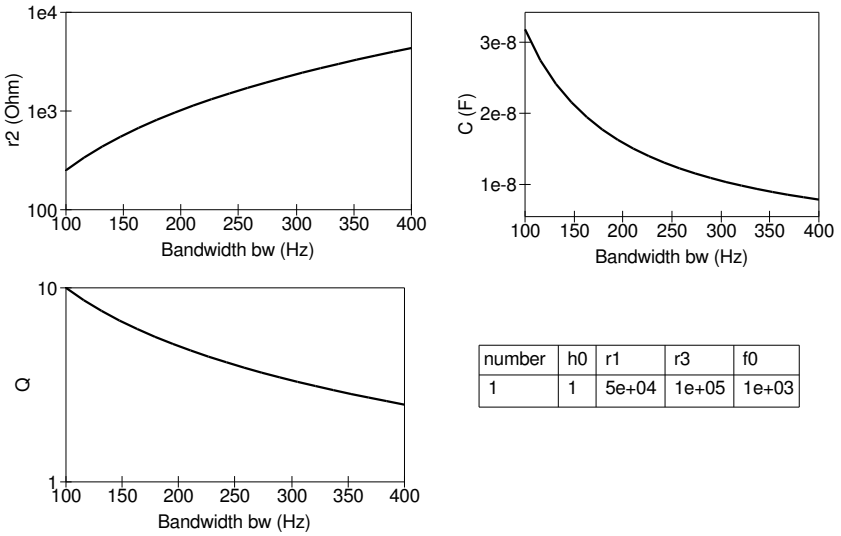


Figure 4. Band-pass filter gain and phase plots.



number	h0	r1	r3	f0
1	1	5e+04	1e+05	1e+03

Figure 5. Delyiannis-Friend active band-pass filter design nomographs.

increasingly lower than the theoretical value; making *gbp* infinite eliminates this error.

5. Embedding circuit design procedures in Verilog-A compact macromodels

Undoubtedly three of the reasons why the Verilog-A hardware description language (Accellera 2004) has become so popular among compact model developers is firstly the inherent power offered by the language, secondly its simplicity, and thirdly its flexibility when writing model code. Like Qucs subcircuits it is possible to extend Verilog-A compact macromodels to include embedded circuit design algorithms. In many respects the procedure for adding design algorithms to Verilog-A models is very similar to that proposed for the Qucs extended subcircuits. However, compact macromodelling using Verilog-A tools, for example the Advanced Device Model Synthesiser (ADMS) (Lemaitre and Gu 2006, Lamaitre and Gu 2009), offers a number of additional features which are not yet implemented with Qucs *Equation*

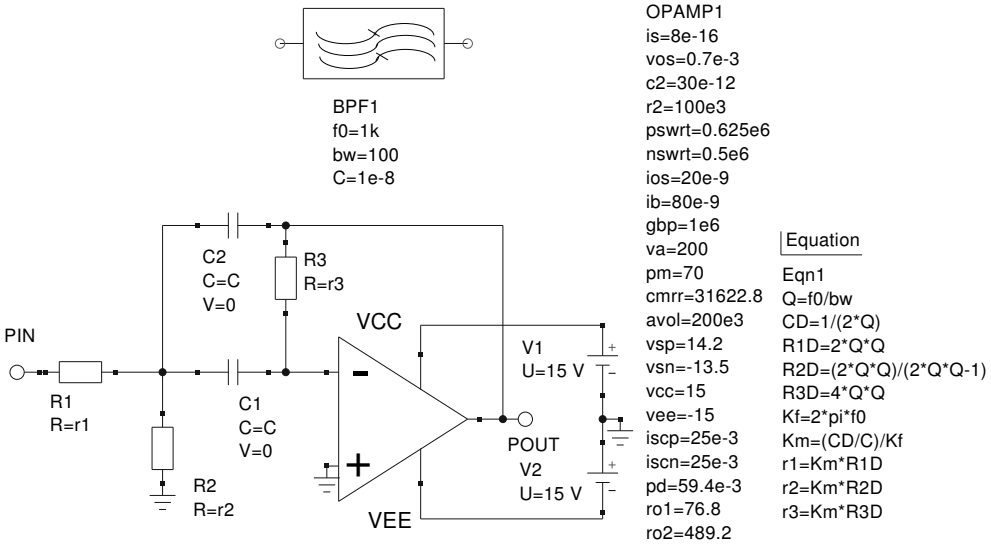


Figure 6. Band-pass filter subcircuit macromodel with embedded design algorithm: symbol and body of macromodel.

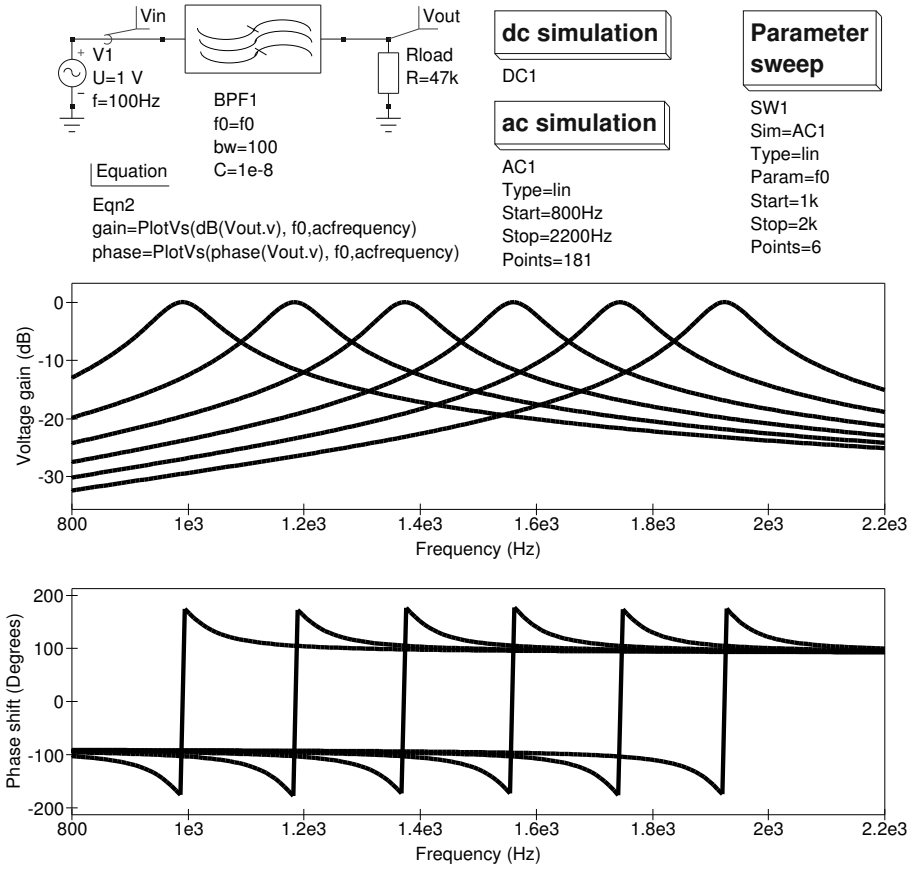


Figure 7. Sub circuit band-pass filter test circuit with gain and phase simulation waveforms.

blocks. This part of the paper describes how circuit design algorithms can be embedded in Verilog-A analog modules, highlighting the Verilog-A hardware description language features that make Verilog-A compact macromodels more powerful than Qucs extended subcircuits. Verilog-A code for the Delyiannis-Friend band-pass filter is listed in Table 2. In the filter example the Verilog-A code comprises a fundamental analog *module* with four major internal sections: (1) definition of internal node and branch names (listed in code lines [6[to [9], (2) the definition of model parameter values (listed in code lines [10] to [15]), (3) code actioned prior to circuit simulation (listed in lines [18] to [21]), and (4) circuit current contributions which are evaluated each time the Verilog-A analogue *module* is called during simulation (listed in lines [23] to [27]). The Verilog-A @($initial\_model$) statement performs a function similar to Qucs *Equation* blocks because it allows circuit design algorithms to be embedded within the Verilog-A module code. The @($initial\_model$) statement is computationally efficient in that it is only evaluated once at the start of a simulation sequence. Traditionally, this statement has in the past been used to calculate the value of algebraic expressions which result in variables and component values that are constant during simulation. It also offers the compact modeller the opportunity to develop design algorithms that include iterative equation solutions which depend on for-loop, case and repeat-while Verilog-A statements. If-then-else statements are also allowed. The addition of a range of programming control statements to Verilog-A design routines significantly increases the power of compact modelling. At this stage in its development Qucs only allows the single line form of the if-then-else statement in an *Equation* block. Looping style statements are not implemented. As a result the Qucs circuit simulator provides modellers with a sophisticated subcircuit modelling medium but one which

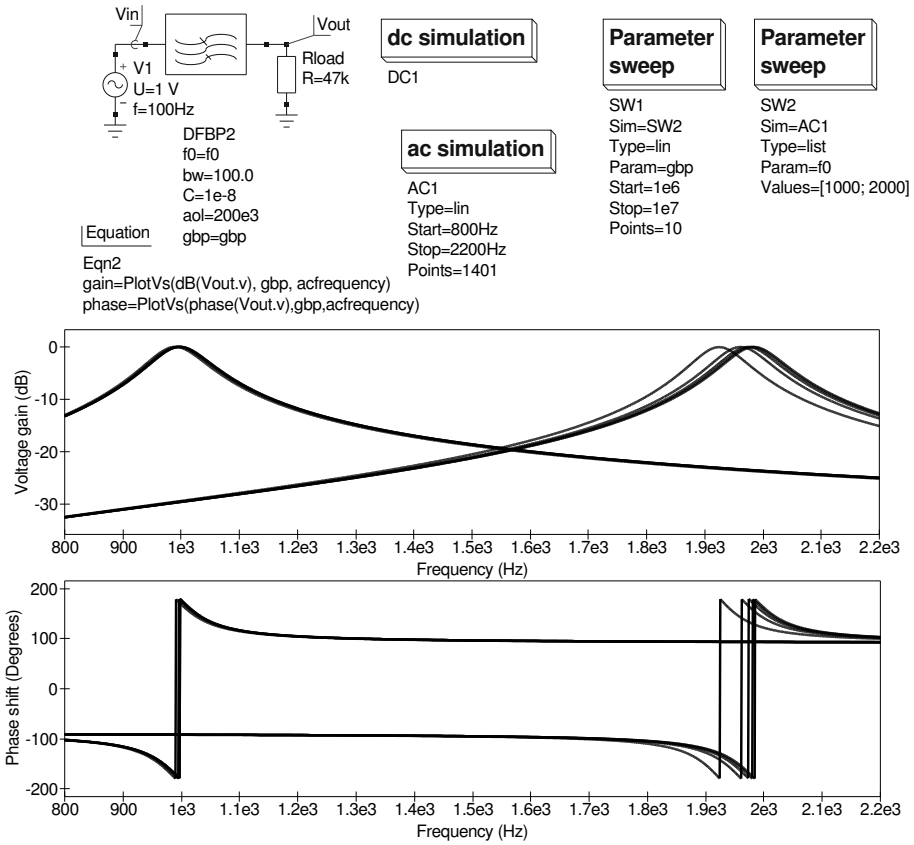


Figure 8. Verilog-A macromodel band-pass filter test circuit with gain and phase simulation waveforms

is slightly less powerful than, but compatible with, the Verilog-A modelling route. Figure 8 presents the same circuit as the one used to test the subcircuit Delyiannis-Friend active band-pass filter. Although the test circuits shown in Figures 7 and 8 are identical the Verilog-A macromodel and extended subcircuit models are slightly different. The test procedures are also slightly different. In the Verilog-A model the Boyle $\mu A741$ operational amplifier model has been replaced by a simple single pole amplifier model with an open loop voltage gain *aol* (value = 200e3) and gain bandwidth product *gbp* (value = 1e6 Hz), giving a similar open loop voltage transfer characteristic to the Boyle model but with simpler Verilog-A code. In Figure 8 both the filter centre frequency $f0$ and the amplfier *gbp* are changed by the Qucs parameter sweep operations. This technique adds a new approach to circuit simulation in that by changing filter specification parameter $f0$ the filter components are recalculated to give essentially a different filter. In contrast to changes in $f0$, changes in *gbp* allow the effects this amplifier parameter has on each of the filter designs, to be investigated. The waveforms illustrated in Figure 8 clearly show that at $f0 = 1000$Hz changing *gbp* from 1MHz to 10MHz has little impact on the filter characteristics. However, the reverse is true when $f0 = 2000$Hz where low values of *gbp* indicate that the simulated value of $f0$ is lower than the expected theoretical value. This is in agreement with the subcircuit results discussed in section 5.

Table 2. Delyiannis-Friend active band-pass filter Verilog-A analog module code (excluding noise contributions). Code line numbers are listed on left hand-side enclosed in [ ] brackets.

```
[1] ‘include ”disciplines.vams”
[2] ‘include ”constants.vams”
[3] module DF_BP_filter(PIN, POUT);
[4] inout PIN,POUT;
[5] electrical PIN,POUT;
[6] electrical n1, n2, n3;
[7] branch (n1) b_n1; branch (n2) b_n2; branch (n3) b_n3; branch (PIN,n1) b_PIN_n1;
[8] branch (POUT,n2) b_POUT_n2; branch (n1,POUT) b_n1_POUT; branch (n1,n2) b_n1_n2;
[9] branch (POUT) b_POUT;
[10] ‘define attr(txt) (*txt*)
[1] parameter real f0=1000.0 from [1e-20 : inf] ‘attr(info=”filter centre frequency” unit=”Hz”);
[12] parameter real bw=100.0 from [1e-20 : inf] ‘attr(info=”filter bandwidth” unit=”Hz”);
[13] parameter real C=1e-8 from [1e-20 : inf] ‘attr(info=”frequency determining capacitance” unit=”F”);
[14] parameter real aol=200e3 from [1 : inf] ‘attr(info=”amplifier DC open loop gain”);
[15] parameter real gbp=1e6 from [1 : inf] ‘attr(info=”amplifier gain bandwidth product” unit=”Hz”);
[16] real Qf, CD, R1D, R2D, R3D, Kf, Km, r1, r2, r3, cp1;
[17] analog begin
[18] @(initial_model) begin
[19]        Qf=f0/bw; CD=1/(2*Qf); R1D=2*Qf*Qf; R2D=R1D/(R1D-1); R3D=2*R1D; Kf=2*‘M_PI*f0;
[20]        Km=(CD/C)/Kf; r1=Km*R1D; r2=Km*R2D; r3=Km*R3D; cp1 = 1/(2*‘M_PI*gbp);
[21] end
[22] // Current contributions
[23] I(b_PIN_n1) <+ V(b_PIN_n1)/r1; I(b_n1) <+ V(b_n1)/r2; I(b_POUT_n2) <+ V(b_POUT_n2)/r3;
[24] I(b_n1_POUT) <+ ddt(C*V(b_n1_POUT)); I(b_n1_n2) <+ ddt(C*V(b_n1_n2));
[25] // Single pole operational amplifier with open loop voltage transfer function similar to Boyle model
[26] I(b_n3) <+ V(b_n2); I(b_n3) <+ V(b_n3)/aol; I(b_n3) <+ ddt(cp1*V(b_n3));
[27] I(b_POUT) <+ -V(b_n3); I(b_POUT) <+ V(b_POUT);
[28] end
[29] endmodule
```

6. Extracting design information and component values from extended subcircuits and compact Verilog-A macromodels

Conventional subcircuits interface with external components and other subcircuits via a set of nodes which are visible to both the external circuitry and the internal body of a subcircuit. These nodes correspond to the pins connected to a subcircuit, allowing current and voltage signals to be applied to or read from a subcircuit. When debugging the operation of a standard subcircuit extra pins are often added to the interface pin list to allow subcircuit internal signals to be made available

Table 3. Example Verilog-A code for component and equation data extraction bus network: Output bus variables are in Volts with PVn = component bus pin number n and PEn = equation bus pin number n.

```
branch (PV1) b_PV1; branch (PV2) b_PV2; branch (PE1) b_PE1;
branch (PE2) b_PE2; branch (PE3) b_PE3; branch (PE4) b_PE4;
branch (A) b_A;
I(b_PV1) <+ -V(b_A)*r1; I(b_PV1) <+ V(b_PV1);          // Component r1
I(b_PV2) <+ -V(b_A)*r2; I(b_PV2) <+ V(b_PV2);          // Component r2
I(b_PE1) <+ -V(b_A)*Q; I(b_PE1) <+ V(b_PE1);           // Equation Q
I(b_PE2) <+ -V(b_A)*CD; I(b_PE2) <+ V(b_PE2);          // Equation CD
I(b_PE3) <+ -V(b_A)*Kf; I(b_PE3) <+ V(b_PE3);          // Equation Kf
I(b_PE4) <+ -V(b_A)*Km; I(b_PE4) <+ V(b_PE4);          // Equation Km
```

for external diagnostic testing. Expanded subcircuits not only allow design procedures to be embedded within the body of a subcircuit but by using the improved equation handling capabilities of Qucs *Equation* blocks they naturally promote the development of more advanced diagnostic tools. Such tools allow component values and equation data to be extracted post circuit design, providing the data needed to generate circuit design nomographs or lists of component values. Figure 9 illustrates an example set of component and design algorithm values extracted using two electrical buses (in Figure 9 long dashes represent the component bus and dots represent the equation bus) driven from a data collection network attached to the Delyiannis-Friend band-pass subcircuit model, see Figure 10. This network consists of a number of voltage controlled current sources with their gain factors set to component or equation values. Filter input signals, at the node labelled A, are scaled to give the required bus output node and parameter values expressed in volts. For transient simulation add an appropriate signal source at the circuit input. Extraction of component and equation data from a subcircuit is only possible using the network illustrated in Figure 10 provided a circuit simulator allows the *gm* values of the voltage controlled current sources to be algebraic variables rather than fixed numerical values. The structure of the data extraction network was chosen to allow simple generation of the equivalent Verilog-A code. Table 3 list the Verilog-A code for extracting the example band-pass filter component and equation values. Merging this code with the Verilog-A model code listed in Table 2 generates a compact macromodel with a similar function to the Qucs extended subcircuit illustrated in Figure 10.

7. Conclusions

It is over twenty years since subcircuits were first used in circuit simulation as a means for partitioning parts of an electronic system into identifiable and reusable units. Throughout this period the basic structure of the subcircuit has remained essentially the same. However, recent generations of circuit simulators have employed pre-simulation processing software for passing parameters to subcircuits and for the calculation of internal component values. In the past the fact that subcircuit component values could, in general, only be defined as numerical values largely limited the use of circuit design algorithms within a subcircuit to minor design tasks. This paper proposes the embedding of equation-defined circuit design algorithms within a subcircuit for circuit design purposes prior to circuit simulation. This approach is

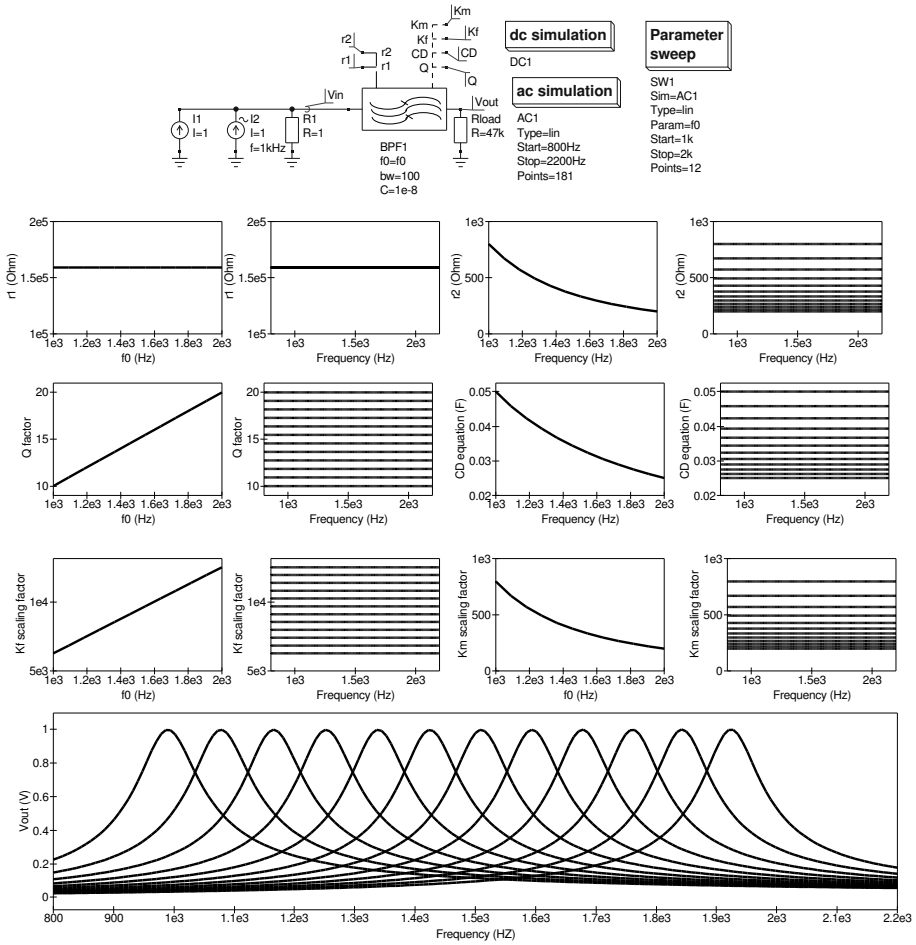


Figure 9. Subcircuit band-pass filter test circuit with component and design equation data extracted and plotted as two dimensional graphs.

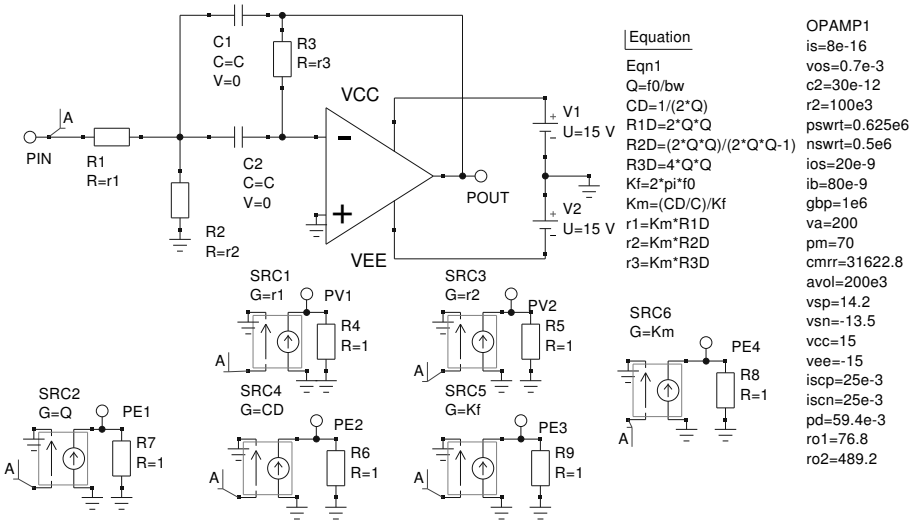


Figure 10. Body of a Qucs extended subcircuit for the Delyiannis-Friend active band-pass filter with component and design equation data extraction buses added.

only possible with circuit simulators that allow the evaluation of blocks of algebraic equations attached to the body of a subcircuit or allow the development of compact Verilog-A macromodels. In the latter case computationally efficient models are possible by embedding design algorithms within the Verilog-A @(*initial_model*) block statement. Conventional subcircuits are normally limited in their ability to pass data from and to their internal circuit, for example passing component values is difficult. By adding a small diagnostic electrical network to the proposed extended subcircuit, or to a compact Verilog-A macromodel, it becomes possible to pass component and equation data from inside a subcircuit to the outside world. This adds a new and important function to subcircuits which allows the construction of useful circuit design nomographs and advanced data extraction/performance diagnostic tools. The proposed extended design/simulation subcircuit reported in this paper demonstrate the progress that is being made in the development of GPL circuit simulators and their modelling capabilities.

Acknowledgement

The material reported in this paper forms part of the work done in response to the MOS Modeling and Parameter Extraction Working Group (MOS-AK) Verilog-A compact modeling standardization initiative (Jahn, Brinson, Margraf, Parruitte, Ardouin, Nenzi and Lemaitre 2007).

References

Accellera (2004),"*Verilog-AMS Language Reference Manual, version 2.2*",http://www.accellera.org (accessed 13 June 2010).

Allabouteda (2008),"*Circuit Simulation = the current state of play*",http://www.allabouteda.com/circuit-simulation-the-next-generation/ (accessed 12 November 2010).

Boyle, G.R., Pederson, D.O., Cohn, B.M. and Solomon J.E., (1974),"Macromodeling of integrated operational amplifiers." in *IEE Journal of Solid State Circuits*, pp. 353-363.

Brinson, M.E., and Faulkner, D.J., (1992),"Modular SPICE macromodel for operational amplifiers," in *IEE Proceedings: Circuits, Devices and Systems*, pp. 417-420.

Brinson, M.E., and Jahn, S., (2009),"Compact macromodelling of operational amplifiers with equation defined devices," in *International Journal of Electronics*, pp. 109-122.

Delyiannis, T., (1968), "High Q factor circuit with reduced sensitivity," in *Electronic Letters*, p 577.

Friend, J.J., (1970), "A single operational-amplifier biquadic filter section," in *IEEE ISCT Digest Technical Papers*, p. 189.

Jahn, S., Brinson, M.E., Margraf M., Parruitte, H., Ardouin, B., Nenzi, P. and Lemaitre, L., (2007),"GNU simulators supporting Verilog-A compact model standardization," *MOS-AK Meeting* , Premstaetten, http://www.mos-ak.org/premstaetten/papers/MOS-AK_QUCS_ngspice_ADMS.pdf (accessed 14 June 2010).

Jahn, S., and Brinson, M.E., (2008),"Interactive compact device modelling using Qucs equation-defined devices," in *International Journal of Numerical Modelling: Electric Networks, Devices and Fields*, pp. 335-349.

Lemaitre, L., and Gu, B., (2006),"ADMS–a fully customizable Verilog-

AMS compiler approach," *MOS-AK Meeting, Montreux*, http://www.mos-ak.org/montreux/posters/17_Lemaitre_MOS-AK06.pdf (accessed 14 June 2010).

Lemaitre, L., and Gu, B., (2009),"ADMS–a fully customizable compact model compiler," *MOS-AK Meeting, Frankfurt(O)*, http://www.mos-ak.org/ftankfurt_o/posters/P_4_Lemaitre_MOS-AK_April_2009.pdf (accessed 14 June 2010).

Margraf, M., Jahn, S., Flucke, J., Jacob, R., Habchi, V., Ishikawa, T., Gopala, K.A., Brinson M.E., Parruitte, H., Roucaries, B., and Kraut, G. (2003), "Qucs (Quite universal circuit simulator)," http://qucs.sourceforge.net/index.html (accessed 13 June 2010).

MicroSim Corporation (1984), *PSpice*, (now included in OrCad 16.0), San José, California: Cadence Design Systems.

Nagel, L., and McAndrew (2009), "SPICE and models - pefect together," http://www.mos-ak.org/athens/Laurence_Nagel_MOS_AK09_athens.pdf (accessed 12 November 2010).

Newton, A.R., Pederson, D.O., and Sangiovanni-Vincentelli. (1992), *SPICE3 Version 3f User's Manual*, Berkeley, California: Department of Electrical Engineering and Computer Sciences, University of California.

Nenzi, P., et al., "ngspice," http://ngspice.sourceforge.net/index.html (accessed 13 June 2010).

Schmidt, F. (2010), "ps2sp.pl a SPICE preprocessor to convert PSpice to SPICE3/XSPICE," http://member.aon.at/fschmidt7/page_3_1.html (accessed 13 June 2010).

Vladimirescu, A., Kaihe Zhang, Newton, A.R., Pederson, D.O., and Sangiovanni-Vincentelli. (1981), *SPICE 2G User's Guide*, Berkeley, California: Department of Electrical Engineering and Computer Sciences, University of California.