A Qucs/QucsStudio Swept Parameter Technique for Statistical Circuit Simulation

Mike E. Brinson, *Fellow, IET*

***Abstract*—Qucs and QucsStudio open source circuit simulators have a wealth of built in swept data features, including facilities for linear and logarithmic scans of simulation variables and for setting component values and device parameters. These simulators also allow semicolon separated lists of numerical values to be used as swept data. This little known feature provides a very flexible mechanism for generating component and device parameter statistical data. An outline of a statistical circuit simulation technique is presented in this paper. The proposed technique can be used with any general purpose circuit simulator equipped with swept data capabilities and as such is suitable for the study of device and circuit performance resulting from variations in device parameters and component values. The operation of the proposed simulation technique is illustrated with the results from an investigation of the statistical performance of a simple MOS current mirror integrated circuit cell, modeled with a speed optimized Verilog-A version of a long channel EPFL_EKV v2.6 MOS transistor model.**

***Index Terms*—QucsStudio; Qucs; statistical circuit simulation; swept parameter lists; Verilog-A compact semiconductor device models.**

I. Introduction

CURRENT trends in semiconductor device modeling and circuit simulation research indicate that there is strong interest in the development of both compact device modeling techniques [1] and improved circuit simulation performance, particularly as part of the core features offered by open source, General Public license (GPL), circuit simulation software packages [2] [3] [4] [5]. In the past the device modeling features provided by the widely available SPICE 2g6 [6] and 3f5 [7] circuit simulators were often limited to subcircuit and macromodel model building techniques. Later commercial circuit simulators often extended these basic tools, by adding, for example, the construction of semiconductor device models at C code level. However, such an approach is in practice only suitable for model developers who have a good understanding of the SPICE C code, particularly the SPICE device model application programming interface. Today the Verilog-A hardware description language [8] has evolved to become a leading contender for compact modeling of semiconductor devices. Much of the current published modeling literature concentrates on improvements to existing device models [9] [10] or on the addition of new compact device models constructed using Verilog-A [11]. In parallel to model development important improvements in circuit analysis and simulation have often emerged from basic circuit analysis research; particularly when the research is applied to popular software packages like, for example. the PSpice [12] circuit simulator. PSpice was among the first commercial personal computer circuit simulators to offer extensions for the statistical analysis of circuits, including circuit sensitivity analysis, worst case analysis and Monte Carlo analysis. This paper introduces a flexible swept data technique that supports investigation of the statistical properties of device models, circuit macromodels and integrated circuit designs. The proposed approach is entirely under user control and as such does not require simulator C or C++ code to be modified or indeed that users have an understanding of advanced C++ programming or a detailed knowledge of low level simulator model application programming interfaces. The reported technique has been implemented with the Qucs and QucsStudio simulators. Moreover, it is suitable for use with any circuit simulator which implements swept data processes. In this paper the fundamentals and application of the new statistical simulation technique are introduced through the investigation of the statistical performance of a basic two transistor MOS current mirror integrated circuit cell, based on a speed optimized version of a long channel EPFL-EKV v2.6 MOS model [13].

M.E. Brinson (email mbrin72043@yahoo.co.uk) is with the Centre for Communications Technology Research, London Metropolitan University, Holloway, London, UK.

II. Generation of Swept Parameter Lists

Previous generations of GPL circuit simulators were often released for general use with data entry procedures that only allowed keyboard entry of component or device parameter numerical values prior to the start of a circuit simulation. The effects and advantages of changing component values or device parameters using swept data techniques have only became fully appreciated with the development of second or later generations of circuit simulator; one exception being the SPICE .DC simulation command where the importance of nested .DC sweeps was realized early, as this technique allows a simple direct way to generate data for plotting semiconductor device I/V curves. The current GPL circuit simulators have in many instances been extended to overcome the performance limitations imposed by restricting component values and device parameters to purely numerical quantities. Modern GPL circuit simulators, like for example, Qucs and QucsStudio, allow component and device parameter values to be specified as algebraic named variables, rather than simple numerical quantities. This basic change has had a profound

Copyright © 2013 by Department of Microelectronics & Computer Science, Lodz University of Technology

effect on the way component values and device parameter values can be set prior to simulation. Shown in Fig. 1 is a QucsStudio schematic symbol for a "Parameter sweep" simulation icon. It acts as a platform for defining a named "Parameter sweep" process starting at value Start and finishing at value Stop, where the interval between Start and Stop is divided into a number of values given by parameter Points. The list of "Parameter sweep" Points may be generated as a linear sequence, a logarithmic sequence or it may be constructed from a group of numerical entries each separated by a semicolon (;).

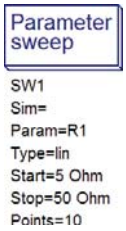


Fig. 1. QucsStudio "Parameter sweep" icon with identifying name (SW1) and initial parameter list, including parameter Sim which identifies the next linked "Parameter Sweep" icon or simulation icon (DC, AC, TRAN, S Parameter or Harmonic Balance).

Both QucsStudio and Qucs allow users to enter by hand a list of semicolon separated numerical values from a keyboard or to automatically generate a list of values using the Octave numerical analysis program. Similarly, the current implementation of the "Parameter sweep" simulation sequence allows "Parameter sweep" icons to either call other sweep icons or any one of the circuit simulation icons in a chain sequence, provided a nested loop does not occur in the chain sequence. Hence, as part of a "Parameter sweep" controlled simulation, component and device parameters are assigned numerical values as a pre-cursor to DC, AC, Transient, S parameter or Harmonic Balance circuit simulation. The "Parameter sweep" list feature is a novel addition to both the QucsStudio and Qucs simulation software. Indeed, it is in many instances not immediately obvious to most users of either circuit simulator why this feature has been included in these packages.

III. User Programmed Sensitivity Circuit Simulation

The combination of nested "Parameter sweep" circuit simulation with user supplied lists of component and device parameter values provides a general purpose circuit simulator with a powerful basic toolset for statistical circuit simulation. Consider the simple resistive voltage divider circuit shown in Fig. 2(a). Conventional non-statistical DC circuit simulation provides information on the transfer characteristics of the circuit characterized by nominal values for resistors R1 and R2. In contrast, by making R1 and R2 lists of component values rather than single numerical values the effect of component variations within a given value range and statistical distribution can be simulated and visualized by plotting two and three dimensional transfer characteristic curves, see Fig. 2 (b). This approach to sensitivity circuit simulation is only possible if a circuit simulator allows component and parameter values to be defined by variable names rather than numerical values. Fig 2 clearly shows the effect of nesting "Parameter sweeps" where both R1 and R2 have values in the range Rmin to Rmax, yielding a box like region with corners defined by (R1min, R2min), (R1min, R2max), (R1max, R2min) and (R1max, R2max) extremes of components R1 and R2.

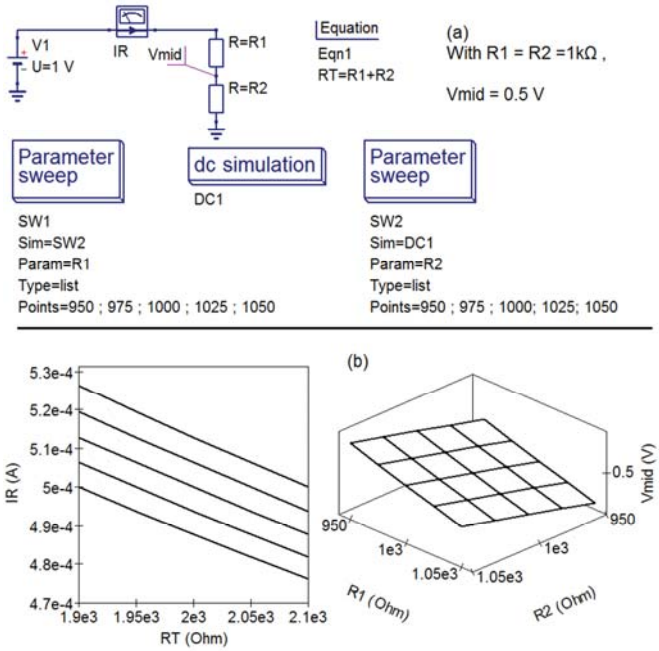


Fig. 2. Sensitivity analysis of a two resistor voltage divider circuit: (a) simulation test circuit and nominal resistor values, (b) example two and three dimensional plots of simulation output data.

IV. Generating Statistical Circuit Simulation Data with Octave

Entering long tables of component data, from a keyboard, into the Points list attached to a "Parameter sweep" icon can be both tedious and error prone. In practice it is better to use a computer program to generate the "Parameter sweep" icons and attached data. Shown in Fig.3 is an Octave [14] script designed to generate a list of normally distributed real numbers linked to a "Parameter sweep" icon. This script requests, as input, a real number for a nominal component value, the required number of values in the semicolon separated list, and finally the component or parameter tolerance as a percentage of the nominal real number. The Octave script constructs a "Parameter sweep" icon netlist file [15] from the user supplied data. This includes code for constructing the "Parameter sweep" icon schematic symbol, its sweep parameters and the attached semicolon separated list of real data values. Library functions are employed by the Octave script to calculate the distribution of the component values. In the example listed in Fig.3 the Octave function "normrnd" is used to generate a univariate normal distribution for each set of component values. Over twenty different types of real number distributions are possible by changing the Octave "normrnd" function to a different mathematical distribution function. These are provided as standard with the

current Octave release. However, from a circuit simulation point of view the rand (uniformly distributed values) and the "nornrnd" functions are among the most important due to their close agreement with practical component and device parameter distributions. Fig. 4 illustrates a typical set of statistical simulation component values generated by the Octave script using function "normrnd". One of the advantages of using Octave numerical analysis scripts for generating QucsStudio/Qucs schematic symbols is the flexibility it gives when calculating the statistical distribution of component values or device parameters. An interesting and important example of this flexibility is the removal of components from a production batch which are within a specified percentage of the batch nominal value. Simply by adding a few lines of code to the Octave script listed in Fig. 3, see Fig. 5, produces a "Parameter sweep" icon with list values of nominal tolerance tol (in percentage of mean value) and tolerance of the selected values stol (in percentage of mean value), where stol < tol. Fig. 6 shows a typical example distribution for a batch of non-integrated circuit resistors.

Although the Octave scripts shown in Fig. 3 and Fig. 5 are applicable to the QucsStudio circuit simulator only small changes are normally required to these scripts to take into account differences in simulator netlist specifications. This can be done by changing one or more of the fputs statements listed in Fig. 3 and Fig. 5. This is particularly true with the Qucs simulator where only small changes are necessary for the generation of the modified "Parameter sweep" icon code.

```
% QucsStudio tabular normal distribution file generator.
%
% INPUT:  Enter data from keyboard as requested.
% OUTPUT: Script writes file XXXX.sch containing
%         a QucsStudio swept parameter icon with
%         a list of "npoints" items normally distributed
%         around a specified mean and standard deviation.
% Script written with Octave 3.2.4
% Copyright 2012 by Mike Brinson
% Published under GNU General Public License (GPL V2).
% No warranty at all.
filename = input("Enter name of Tabular Distribution File > ", "s");
version  = input("Enter QucsStudio version number > ", "s");
npoints  = input("Enter the number of data points in the tabular sequence > ");
mean     = input("Enter mean value > ");
tol      = input("Enter tolerance [ in percentage ] > ");
std = (tol.*mean)./300.0;
n = normrnd(mean, std, 1, npoints);
fname = strcat(filename,".sch");
fid  =  fopen(fname, "w");
fprintf(fid, "<QucsStudio Schematic %s>\n", version);
fputs(fid, "<Properties>\n");
fputs(fid, "   <View=0,0,3256,800,1,0,0>\n");
fputs(fid, "   <Grid=10,10,1>\n");
fprintf(fid, "   <DataSet=%s.dat>\n", filename);
fprintf(fid, "   <DataDisplay=%s.dpl>\n", filename);
fputs(fid, "   <OpenDisplay=1>\n");
fputs(fid, "   <showFrame=0>\n");
fputs(fid, "   <FrameText0=Title>\n");
fputs(fid, "   <FrameText1=Drawn By:>\n");
fputs(fid, "   <FrameText2=Date:>\n");
fputs(fid, "   <FrameText3=revision:>\n");
fputs(fid, "</Properties>\n");
fputs(fid, "<Symbol>\n");
fputs(fid, "</Symbol>\n");
fputs(fid, "<Components>\n");
fputs(fid, "<.SW SW1 1 90 120 0 83 0 0 ");
fputs(fid, "\"DC1\" 1  \"p1\" 1 \"list\" 1  \"   \" 0  \"   \" 0 \" ");
for i =1:npoints-1
  fprintf(fid," %12.3e ;", n(i));
endfor
fprintf(fid, " %12.3e \" 1>\n", n(npoints));
fputs(fid, "</Components>\n");
fputs(fid, "<Wires>\n");
fputs(fid, "</Wires>\n");
fputs(fid, "<Diagrams>\n");
fputs(fid, "</Diagrams>\n");
fputs(fid, "<Paintings>\n");
fputs(fid, "</Painting>\n");
fclose(fid);
```

Fig. 3. Octave m script for generating a QucsStudio "Parameter sweep" icon schematic with a semicolon separated data list attached. The list consists of a normally distributed set of values specified by a mean and standard deviation.

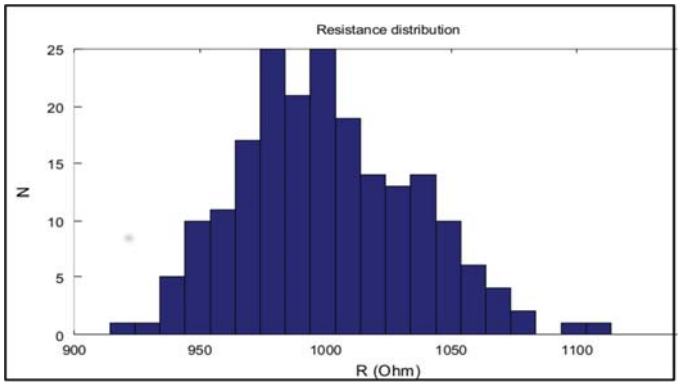


Fig. 4. Distribution of resistance component values generated using the Octave script listed in Fig.3: Nominal value of R = 1k Ω, number of samples = 500 and tolerance = 10%.

```
% Published under GNU General Public License (GPL V2).
% No warranty at all.
%
filename = input('Enter name of Tabular Distribution File [format "name"] > ');
version  = input('Enter QucsStudio version number [format "x.x.xx"] > ');
npoints  = input('Enter the number of data points in the tabular sequence >');
mean     = input('Enter mean value >');
tol      = input('Enter tolerance [ in percentage ] > ');
stol     = input('Enter tolerance for selected values [ in percentage ] > ');
if ( stol >= tol )
   stol = 5;
endif
std = (tol.*mean)./300.0;
nhold = normrnd(mean, std, 1, npoints);  ########## Modified code ##########
%  Remove selected components
vmin = mean.*(1-stol./100);
vmax = mean.*(1+stol./100);
j = 1;
k = 1;
while (j <= npoints)
  if ( (nhold(j) <= vmin ) || (nhold(j)) >= vmax )
     n(k) = nhold(j);
     k++;
  endif
  j++;
endwhile
%                                ########## End modified code #########
fname = strcat(filename,".sch");
%swname = strcat("SW","1");
fid  =  fopen(fname, "w");
```

Fig. 5 QucsStudio normal distribution list generator showing additional Octave code for removing component values or device parameters from within the range nominal value ± stol percent of mean.

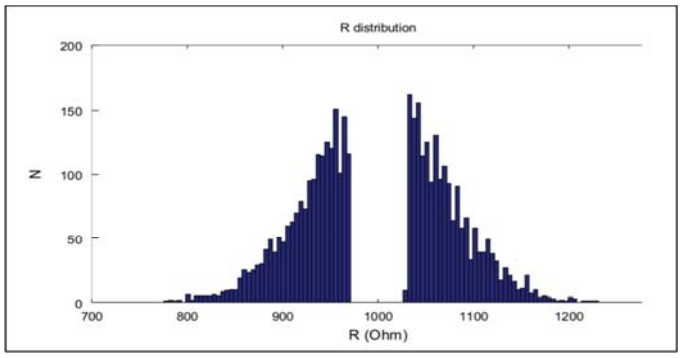


Fig. 6. A 1000 sample histogram showing distributed resistance with R values in the range ± 3% of the nominal value of 1000 Ω (±30Ω) removed.

V. Statistical Circuit Simulation of a MOS Current Mirrror IC Cell

Statistical simulation of a circuit design provides an insight into the causes of circuit failure due to geometric and physical device parameter variations. Fig. 7 shows a basic nMOS current mirror IC cell operating from a single positive voltage supply with equal drain resistances R1 and R2. With perfectly matched transistors, and matched drain resistors, the output current (Iout) to input current (Iin) ratio is given by

$$\frac{Iout}{Iin} = \frac{W2/_{L2}}{W1/_{L1}} \quad (1)$$

where W2=W2n+DW, L2=L2n+DL, W2 and L2 are the gate width and length of the output transistor respectively, W2n and L2n are the nominal values of the output transistor gate width and length respectively, DW is a channel width correction factor and DL a channel length correction factor. W1 and L1 have similar meaning but for the input transistor. In a practical circuit, device parameters are found to vary between circuit fabrication batches and across circuits on the same integrated circuit die. Variations in fabrication process parameters, for example impurity concentration densities, oxide thickness and diffusion depths introduce variations in the transistor physical parameters such as the threshold voltage VTO. Similarly, variations in the dimensions of individual devices, for example photolithography dimensional differences, cause W/L variations in the MOS transistors. Hence, variations in the DC current mirror output to input current ratio will, to a first approximation, depend on the statistical variation of individual transistor L and W dimensions and variations in transistor physical parameters. Fig. 8 shows a typical set of statistical DC simulation data obtained by varying one of the L1, W1, L2 and W2 parameters, via DW and DL, while keeping the remaining dimensional parameters at their nominal values. Similarly, in this example, the values of the transistor physical parameters were kept constant at their nominal values during simulation. The sample data set is 1000 in this test example.

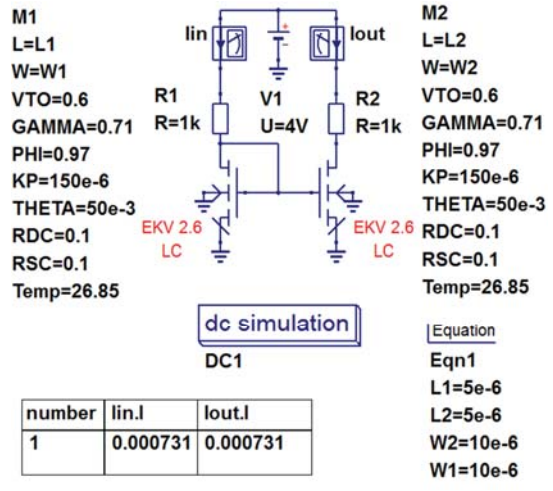
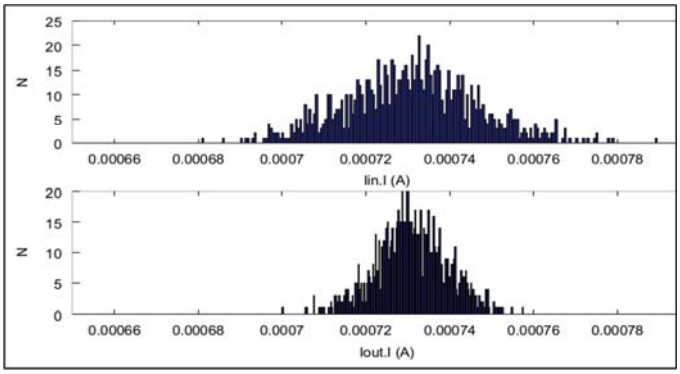
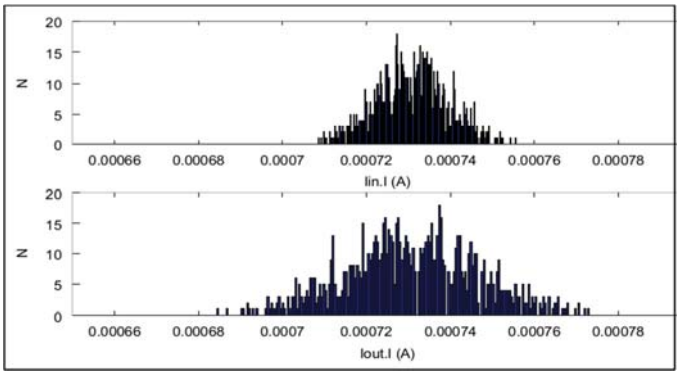


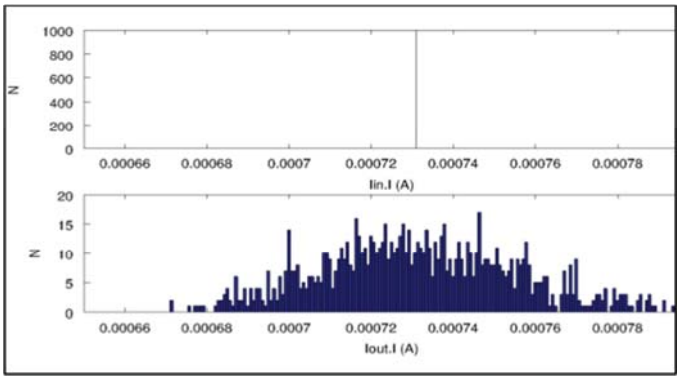
Fig. 7. An nMOS current mirror test circuit with matched device geometries (L1=L1n=5µm, L2=L2n=5µm, W1=W1n=10µm, W2=W2n=10µm) and VTO=0.6V; nMOS transistors are represented by EPFL-EKV 2.6 long channel Verilog-A device models [16]: available in the QucsStudio "extras library or as a non-linear active device in Qucs".



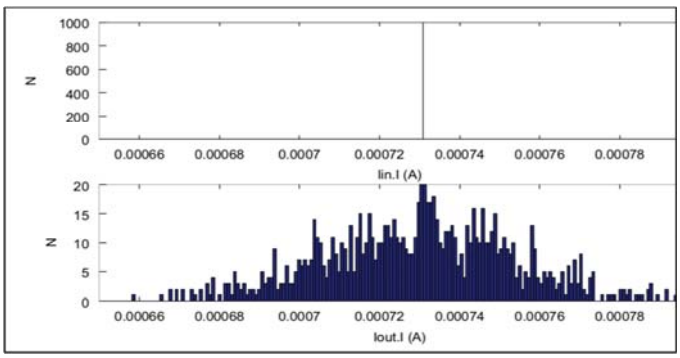
(a): L1=5µm±10%=L1n+DL, W1=10µm, L2=5µm and W2=10µm.



(b): W1=10µm±10%=W1n+DW, L1=5µm, L2=5µm and W2=10µm.



(c): L2=5µm±10%=L2n+DL, L1=5µm, W1=10µm and W2=10µm.



(d): W2=10µm±10%=W2n+DW, L1=5µm, L2=5µm and W2=10µm.

Fig. 8. Typical statistical simulation results for an nMOS current mirror test circuit with L1, W1, L2 and W2 as specified in (a) to (c). All other parameters are listed in Fig.7.

VI. Optimizing the Current Mirror Model for Minimum Simulation Run Time

Statistical circuit simulation is in general computationally expensive, for example varying three component values, or device parameters, over a range set by ten values requires 10 by 10 by 10 or 1000 individual simulations, Obviously, each time a parameter is added to a statistical circuit analysis the number of simulations increases proportionally. Simulation run times also strongly depend on the type of simulation being undertaken; transient analysis taking the longest time. Simulation times also increase when the circuit under test includes nonlinear active devices. A significant computational overhead in circuit simulation is the time taken to calculate the equations which form the basic non-linear model of an active device. In general, these calculations can be broken down into two distinct groups; firstly equations which are functions of variables which vary with, for example, circuit voltage, current or temperature and secondly those variables which are constant throughout a simulation. The first group needs to be recalculated each time a model is called by a circuit simulator. In contrast the second group has only to be determined once at the start of a circuit simulation. Simulation times are increased further when the circuit under test contains a high number of identical transistors. In such a case the variables which are constant throughout a simulation get calculated the same number of times as there are identical devices, instead of only once. One way to reduce this overhead is to calculate the transistor variables which are constant during a circuit simulation at a higher abstraction level than the non-linear device models and to pass the values of the calculated variables to each transistor model rather than the basic device parameters. A simple way of achieving this is to add an additional level of abstraction to a subcircuit and to pass the values of the calculated variables from the inner level to a device model. Fig. 9 shows a modified current mirror test circuit where the current mirror is represented by a subcircuit with three levels of abstraction. The first inner level of abstraction is shown in Fig. 10. Here variables P1 to P8 are passed as parameters to the bottom level of abstraction. This contains modified Verlog-A EPFL-EKV v2.6 nMOS device code. With this structure the variables in equation block eqn1,

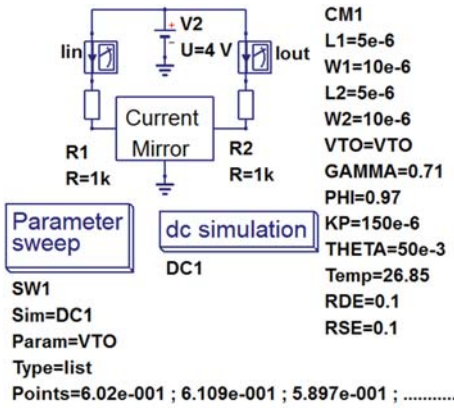


Fig. 9. A QucsStudio test bench for simulating the performance characteristics of a three level MOS DC current mirror subcircuit model with L1, W1, L2 and W2 set at nominal values and VTO in the range 0.6V ± 60mV.

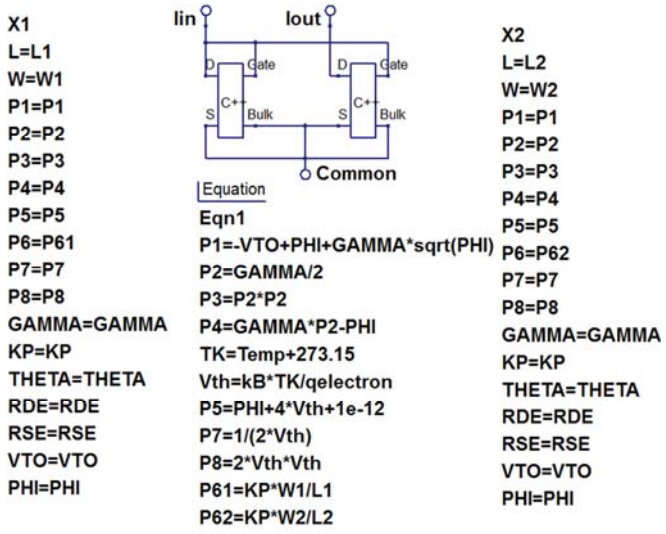


Fig. 10. The internal structure of a three level MOS DC current mirror subcircuit constructed using two speed optimized EPFL-EKV v2.6 nMOS long channel compact device models.

```
//   QucsStudio/Qucs EPL-EKV 2.6 nMOS long channel DC model: NMOSLC.va
//   Copyright (C), Mike Brinson, mbrin72043@yahoo.co.uk, January 2013.
//   Published under GNU General Public License (GPL V2).
`include "disciplines.vams"
`include "constants.vams"
//
 module NMOSLC (D, Gate, S, Bulk);
 inout D, Gate, S, Bulk;
 electrical D, Gate, S, Bulk; electrical Drain, Source;
`define attr(txt) (*txt*)
// Device dimension parameters
 parameter real L = 2e-6 from [0.0 : inf);
 parameter real W = 2e-6 from [0.0 : inf);
 parameter real P1 =0.0 `attr(info="super-subcircuit parameters P1 to P8");
 parameter real P2 =0.0;
 parameter real P3 =0.0;
 parameter real P4 =0.0;
 parameter real P5 =0.0;
 parameter real P6 =0.0;
 parameter real P7 =0.0;
 parameter real P8 =0.0;
 parameter real GAMMA = 0.71 from [0.0 : 2.0);
 parameter real KP = 150e-6;
 parameter real THETA = 50e-3 from [0 : inf);
 parameter real RDE = 0.01 from [1e-6 : inf);
 parameter real RSE = 0.01 from [1e-6 : inf);
 parameter  real Temp = 26.85 from [-150.0 : 200.0];
 parameter  real VTO = 0.6;
 parameter  real PHI = 0.97;
//
real VGprime, VP, n, Beta, iff1, iff, irr1, irr;
// Branches
branch (Gate, Bulk) Bgb;   branch (Source, Bulk) Bsb;
branch (Drain, Bulk) Bdb;  branch (Drain, Source) Bds;
branch (D, Drain) BDDrain; branch (Source, S) BSourceS;
analog begin
VGprime = V(Bgb)+P1;
if (VGprime > 0.0)
 begin
        VP = P4+VGprime-GAMMA*sqrt(VGprime+P3);
  end
else  begin VP = -PHI; end
n = 1+P2/sqrt(VP+P5); Beta = P6/(1+THETA*VP);
iff1 = ln(1+limexp((VP-V(Bsb))*P7)); iff = iff1*iff1;
irr1 = ln(1+limexp((VP-V(Bdb))*P7)); irr = irr1*irr1;
I(Bds) <+ P8*n*Beta*(iff-irr);
// External resistances
  I(BDDrain)  <+ V(BDDrain)/RDE; I(BSourceS) <+ V(BSourceS)/RSE;
end
endmodule
```

Fig. 11. Verilog-A code for a EPFL-EKV v2.6 MOS DC long channel model modified to allow equation-derived device parameters P1 to P8.

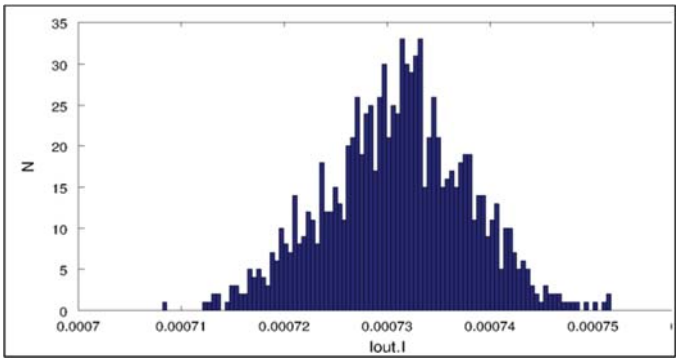


Fig. 12. Histogram of current mirror output current for a 1000 VTO samples normally distributed in the range 0.6V ±60mV.

Fig. 10, are calculated only once, rather than twice. Obviously for current mirrors, or other circuits, which are constructed from larger numbers of transistors the number of calculations would be reduced proportionally. Fig. 11 outlines the Verilog-A code for the modified EPFL-EKV v2.6 nMOS model. A histogram of the statistical distribution of current mirror output current Iout.I due to variations in VTO is also illustrated in Fig. 12.

VII. Conclusions

Statistical simulation of a circuit design provides an insight into the causes of circuit failure due to geometric and physical device parameter variations. This paper introduces a flexible swept data technique that supports investigation of the statistical properties of device models, circuit macromodels and integrated circuit designs. The reported technique has been implemented with the Qucs and QucsStudio simulators and does not require an extensive knowledge of the internal structure and workings of C++ simulator software, making it suitable for use by all the Qucs/QucsStudio simulator user community. Moreover, it is also suitable for use with any circuit simulator which implements swept data processes. To demonstrate the fundamentals and application of the proposed swept parameter statistical simulation technique the simulation of a nMOS current mirror is introduced in the text. This example also introduces the concept of a three level subcircuit Verilog-A compact device model with equation variable data passing and improved simulation run times.

Acknowledgement

The material reported in this paper forms part of the work done in response to the MOS Modelling and Parameter Extraction Working Group (MOS-AK) Verilog-A compact modelling standardization initiative [17].

References

[1] G. J. Coram, “How to (and how not to) write a compact model in Verilog-A”, IEEE International Behavioural Modeling and Simulation Conference (BMAS 2004), San Josě, CA, USA. 2004, 97-106.

[2] M. Margraf, S. Jahn, J. Flucke, R. Jacob, V. Habchi, T. Ishikawa, A. Gopala Krishna, M. Brinson, H. Parruitte, B. Roucaries , G. Kraut and F. Schreuder. Qucs (Quite universal circuit simulator), Version 0.0.17, http://qucs.sourceforge.net/index.html, 2013. [accessed January 2013].

[3] M. Margraf, QucsStudio, Version 1.4.3. http://www.mydarc.de/DD6UM/QucsStudio/qucsstudio.html, 2013. [accessed January 2013].

[4] P. Nenzi, ngspice release 25, http://ngspice.sourceforge.net/index.html, 2013. [accessed January 2013].

[5] A. Davis, Gnucap, Version 0.35, http://www.gnu.org/software/gnucap/, 2009. [accessed January 2013].

[6] A. R. Newton, D. O. Pederson, A. Sangiovanni-Vincentelli, “A SPICE 2g User’s Guide”, Department of Electrical Engineering and Computer Science, University of California, Berkeley, CA, 1981.

[7] B. Johnson, T. Quarles, A. R. Newton, D. O. Pederson, A. Sangiovanni-Vincentelli,”A SPICE3 Version 3f User’s Manual”, Department of Electrical Engineering and Computer Science, University of California, Berkeley, CA, 1992.

[8] Accellera “Verilog-AMS Language Reference Manual, Version 2.2”, 2004, http//www.accellera.org. 2010. [accessed January 2013].

[9] L. Lemaitre, G. J. Corum, C. McAndrew and K Kundert, “Extensions to Verilog-A to support compact device modeling”, Proceedings of the IEEE International Workshop on Behavioural Modeling and Simulation, BMAS, 7-8 Oct. 2003, pp, 134-138.

[10] Yogesh Chauhan, “Transitioning BSIM4 to BSIM6”, University of California, Berkeley, CA, http://www.mos-ak.org/india/presentations/Chauhan_BSIM6_MOS-AK_India12.pdf.

[11] Compact Model Council, TechAmerica, Arlington, VA. http://www.geia.org , 2009. [accessed January 2013].

[12] Cadence, “PSpice A/D and Advanced Analysis”, http://www.cadence.com/products/orcad/pspice_simulation/pages/default.aspx, 2013. [accessed January 2013].

[13] M. Bucher et al., “The EPFL-EKV MOSFET V2.6 model equations for simulation”, Technical Report. Electronics Laboratories, Swiss Federal Institute of Technology (EPFL), Lausanne, Switzerland, June 1997.

[14] GNU Octave. 1992-2013, Available from: http://www.gnu.org/software/octave/.

[15] QucsStudio, Version 1.4.3. Help file: section – Description of Qucs file formats. http://www.mydarc.de/DD6UM/QucsStudio/qucsstudio.html, 2013. [accessed January 2013].

[16] M.E. Brinson ands. Jahn, “Compact device modeling for established and emerging technologies with the Qucs GPL circuit simulator”, MIXDES 2009, 16 International Conference, Mixed Design and Integrated Circuits, Łódź, June 25-27,2009.

[17] S. Jahn, M. Brinson, H. Parruitte, B. Ardouin, P. Nenzi and L. Lemaitre, ”GNU simulators supporting Verilog-A compact model standardization“, MOS-AK meeting, Premstaetten 2007, http://www.mosak.org/premstaetten/papers/MOSAK_QUCS_ngspice_ADMS.pdf, [accessed July 2012].

M.E. 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, Bremen, 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, specialising in device and circuit modelling, testing and document preparation.