Qucs Equation-Defined Device modelling with a Verilog-A Prototyping Platform

Mike Brinson\textsuperscript{1}, mbrin72043@yahoo.co.uk.
Vadim Kuznetsov\textsuperscript{2}, ra3xdh@gmail.com
Wladek Grabinski\textsuperscript{3}, wladek@mos-ak.org

\textsuperscript{1}Centre for Communications Technology, London Metropolitan University, UK

\textsuperscript{2} Bauman Moscow Technical University, Russia

\textsuperscript{3} MOS-AK (EU)

Presented at the 8th International MOS-AK Workshop, Washington DC, December 9 2015
Qucs: An introduction to the new simulation and compact device modelling features implemented in release 0.0.19/0.0.19S of the popular GPL circuit simulator

- Qucs-0.0.19/S structure: overview, spice4qucs initiative tasks and main features
- Compact modelling with Qucs, ngspice, and Xyce
  - EDD support: Current and charge equations
  - B-type SPICE sources
  - Harmonic balance simulation with Xyce and Qucs compact models
- Parametrization features and ngnutmeg scripting introduced with spice4qucs
- Introduction to the Qucs subcircuit to Verilog-A module synthesizer
- Plans for future
Qucs-0.0.19/S structure diagram for simulation and compact device modelling
Overview of spice4qucs structure: – features and current Qucs-0.0.19S version

Spice4qucs features:

- Correct known weaknesses observed with the current Qucs simulation engine qucsator
- Provide Qucs users with a choice of simulator selected from qucsator, ngspice and Xyce
- Extend Qucs subcircuit, EDD, RFEDD and Verilog-A device modelling capabilities
- Access to the additional simulation tools and extra component and device models provided by ngspice and Xyce
- Mixed-mode analogue-digital circuit simulation capability using Qucs/ngspice/XSPICE simulation

Qucs-0.0.19/S:

- Ngspice, Xyce (both serial and parallel) support
- Basic simulations support (.DC, .AC, .TRAN)
- Advanced simulation support (.FOUR, .DISTO, .NOISE, .HB)
- Semiconductor devices with full SPICE specifications
- Qucs equations, parametrization (.PARAM), and ngnutmeg script support
- Custom ngspice simulation – User controlled simulation based on ngnutmeg scripts
- Qucs subcircuit to Verilog-A module synthesizer support

Qucs $\leftarrow \rightarrow$ Ngspice/Xyce interfacing schematic

Consider tunnel diode model represented by

\[
I = I_s \left( e^{\frac{V}{\phi T}} - 1 \right) + I_v e^{k(V-V_v)} + I_p \cdot \frac{V}{V_p} e^{\frac{V_p-V}{V_p}} \tag{1}
\]

With spice4qucs, Qucs EDD charge components can be represented by B-type ngspice/Xyce current sources:
Compact modelling with Qucs and ngspice/Xyce: Part II – Charge equation approach

Nonlinear capacitance current expressed as a function of device voltage can be written as:

\[ I = \frac{dQ}{dt} = \frac{d}{dt} CV \]  \hspace{1cm} (2)

As Xyce and ngspice appear not to support the diff() operator an electrical equivalent circuit is needed to model capacitor charge equations:

- **Nonlinear capacitance equivalent circuit:**

![Equivalent Circuit Diagram]

\[ I = \text{ddt}(C \cdot V(C)) \]
In this example a nonlinear capacitance is simulated with ngspice and Xyce:

\[
Q = C_1 V + \frac{C_2 V^2}{2} + \frac{C_3 V^3}{3} + \ldots + \frac{C_N V^N}{N}
\]  

(3)
Qucs 0.0.19/S introduces a new component: SPICE-compatible equation defined voltage or current sources (SPICE B-type source). The B-type sources allow straightforward construction of compact device models:

$$I = I_s \cdot \exp \left( \frac{V(anode)}{\phi T} \right) - 1.0 + I_V \cdot \exp \left( K \cdot (V(anode) - V_V) \right) + I_P \cdot \frac{V(anode)}{V_P} \cdot \exp \left( \frac{V_P - V(anode)}{V_P} \right)$$

Auto-generated SPICE netlist:

```
.PARAM kB = 1.38e-23
.PARAM q = 1.6e-19
.PARAM Vv = 0.4
.PARAM Iv = 1e-6
.PARAM Ip = 1e-5
.PARAM Is = 1e-12
.PARAM Vp = 0.1
.PARAM K = 5
.PARAM Temp0 = 300
.PARAM phiT = \{(kB*Temp0)/q\}

VPr1 _net0 anode DC 0 AC 0
V1 _net0 0 DC 1
B1 anode 0 I = Is*(exp(V(anode)/phiT)-1.0)+Iv*exp(K*(V(anode)-Vv))+Ip*(V(anode)/Vp)*exp((Vp-V(anode))/Vp)

.control
set filetype=ascii
echo "" > spice4qucs.cir.noise
DC V1 -0.05 0.4 0.009
write tunn-Bsrc_dc.txt VPr1#branch v(anode)
.destroy all
.reset
.exit
.endc
.END
```
Compact modeling with Qucs and ngspice/Xyce: Part V – NPN BJT compact model used for Harmonic balance analysis of a one-stage BJT amplifier

- Spice4qucs and Xyce allow large signal steady state AC Harmonic Balance simulation, for example the simulation of an experimental NPN BJT compact macromodel:

\[
\begin{align*}
I_1 &= I_s (\exp(Delta_f V_2) - 1) \cdot \text{stp}(-Delta_f V_2 + X_{crit_f}) + I_s X_{crit_f} \cdot (1 + (Delta_f V_2 - X_{crit_f}) \cdot (1 + (Delta_f V_2 - X_{crit_f})/2)) \cdot \text{stp}(Delta_f V_2 - X_{crit_f}) \\
I_2 &= 0 \\
I_3 &= I_s (\exp(Delta_r V_2) - 1) \cdot \text{stp}(-Delta_r V_2 + X_{crit_r}) + I_s X_{crit_r} \cdot (1 + (Delta_r V_2 - X_{crit_r}) \cdot (1 + (Delta_r V_2 - X_{crit_r})/2)) \cdot \text{stp}(Delta_r V_2 - X_{crit_r}) \\
I_4 &= 0 \\
I_5 &= 0
\end{align*}
\]

\[
\begin{align*}
T_{Kelvin} &= T_{bulk} + 271.15 \\
Delta_f &= q / (N_f k_B T_{Kelvin}) \\
Delta_r &= q / (N_r k_B T_{Kelvin}) \\
X_{crit_f} &= V_{crit} \cdot Delta_f \\
X_{crit_r} &= V_{crit} \cdot Delta_r \\
E_{crit_f} &= \exp(X_{crit_f}) \\
E_{crit_r} &= \exp(X_{crit_r}) \\
\end{align*}
\]

\[
\begin{align*}
Q_1 &= T_{r} V_4 + P_{Cj} (V_1 - V_{maxc}) \cdot (1 + (V_1 - V_{maxc}) \cdot (0.5 + (V_1 - V_{maxc})/6)) \\
I_2 &= V_4 / Bf \\
Q_2 &= T_{f} V_5 + P_{Cje} (V_2 - V_{maxe}) \cdot (1 + (V_2 - V_{maxe}) \cdot (0.5 + (V_2 - V_{maxe})/6)) \\
I_3 &= (V_4 - V_5) \\
I_4 &= 0 \\
I_5 &= 0
\end{align*}
\]
Compact modelling with Qucs and ngspice/Xyce: Part VI – HB analysis

SPICE netlist and output data for a BJT amplifier

Xyce harmonic balance simulation data and auto generated netlist for one-stage BJT amplifier; see http://www.mixdes.org/Mixdes3/:
Compact modelling with Qucs and ngspice/Xyce: Part VII – XSPICE macromodels usage

- Qucs-0.0.19/S allows embedding of SPICE netlist models in Qucs libraries
- An example application of this feature is shown below
  - Direct simulation of SPICE defined components
  - XSPICE macromodel usage
- LM358 XSPICE macromodel usage example (noninverting amplifier):

```
LM358 XSPICE macromodel

.OP1
    ac simulation
    AC1
    Type=log
    Start=1 Hz
    Stop=10 MHz
    Points=141

.R2
    R=10k

.TR1
    Type=lin
    Start=0
    Stop=1 ms
    Points=200

.V2
    U=5 V

.V1
    U=1 V
    f=5 kHz

.TR1
    V2
    ac simulation
    LM358_test_ngspice:tran.v(in)
    LM358_test_ngspice:tran.v(out)

.TRAN
    1 10 100 1e3 1e4 1e5 1e6 1e7
    0 10 100 1e3 1e4 1e5 1e6 1e7

.END
```

```
LM358 XSPICE macromodel

.OP1
    ac simulation
    AC1
    Type=log
    Start=1 Hz
    Stop=10 MHz
    Points=141

.R2
    R=10k

.TR1
    Type=lin
    Start=0
    Stop=1 ms
    Points=200

.V2
    U=5 V

.V1
    U=1 V
    f=5 kHz

.TR1
    V2
    ac simulation
    LM358_test_ngspice:tran.v(in)
    LM358_test_ngspice:tran.v(out)

.TRAN
    1 10 100 1e3 1e4 1e5 1e6 1e7
    0 10 100 1e3 1e4 1e5 1e6 1e7

.END
```

```
LM358 XSPICE macromodel

.OP1
    ac simulation
    AC1
    Type=log
    Start=1 Hz
    Stop=10 MHz
    Points=141

.R2
    R=10k

.TR1
    Type=lin
    Start=0
    Stop=1 ms
    Points=200

.V2
    U=5 V

.V1
    U=1 V
    f=5 kHz

.TR1
    V2
    ac simulation
    LM358_test_ngspice:tran.v(in)
    LM358_test_ngspice:tran.v(out)

.TRAN
    1 10 100 1e3 1e4 1e5 1e6 1e7
    0 10 100 1e3 1e4 1e5 1e6 1e7

.END
```
Qucs equation support in spice4qucs

An example for evaluating the total $S$, active $P$, and reactive $Q$ power in an RC passive electrical network:

$$S = \text{abs}(U \cdot \bar{I}) \quad P = \Re[U \cdot \bar{I}] \quad Q = \Im[U \cdot \bar{I}]$$  \hspace{1cm} (4)
The following Qucs “equation” style icons introduce model parametrization and simulation data postprocessing:

- **SPICE .PARAM section icon**
- **ngnutmeg equation icon**
New analysis-simulation types implemented with spice4qucs: SPICE small signal distortion, SPICE small signal AC domain and large signal time domain noise, and SPICE Fourier analysis

![Circuit Diagram](image)

### Results

<table>
<thead>
<tr>
<th>number</th>
<th>BJT-disto_ngspice:inoise_total</th>
<th>BJT-disto_ngspice:onoise_total</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1.48e-10</td>
<td>3.91e-09</td>
</tr>
</tbody>
</table>

**Fourier Analysis Results**

- **BJT-disto_ngspice:ac.h2**
- **BJT-disto_ngspice:ac.h3**

**Distortion Analysis Results**

- **NOISE1**
  - Type=lin
  - Start=1 Hz
  - Stop=10 MHz
  - Points=1000
  - Output=v(out)
  - Source=V2

- **DISTO1**
  - Type=lin
  - Start=1 Hz
  - Stop=10 kHz
  - Points=1000

**Noise Analysis Results**

- **NOISE1**
  - Type=lin
  - Start=1 Hz
  - Stop=10 MHz
  - Points=1000
  - Output=v(out)
  - Source=V2

**Transient Analysis Results**

- **TR1**
  - Type=lin
  - Start=0
  - Stop=1 ms

**Fourier Simulation**

- **FOUR1**
  - Sim=TR1
  - numfreq=20
  - F0=4kHz
  - Vars=V(out)
Ngspice custom simulation techniques: Part I – Main features

Main features:

- Embedding user defined ngnutmeg scripts in a Qucs schematic
- Full ngnutmeg operator and function support
- User defined variables for plotting simulation data
- User defined raw ASCII SPICE3f5 style output

- Ngnutmeg script editing dialogue:
Ngspice custom simulation technique: Part II – Application example: Monte-Carlo simulation controlled via a ngnutmeg script

```
CUSTOM1
SpiceCode=
set filetype=ascii
let mc_runs = 5
let run = 0
define unif(nom, rvar) (nom + (nom*rvar) * sunif(0))
define aunif(nom, avar) (nom + avar * sunif(0))
define gauss(nom, rvar, sig) (nom + (nom*rvar)/sig * sgauss(0))
define agauss(nom, avar, sig) (nom + avar/sig * sgauss(0))
define limit(nom, avar) (nom + ((sgauss(0) >= 0) ? avar : -avar))
*
dowhile run < mc_runs $ loop starts here
  alter c1 = unif(1e-09, 0.1)
alter l1 = unif(10e-06, 0.1)
alter c2 = unif(1e-09, 0.1)
alter l2 = unif(10e-06, 0.1)
alter l3 = unif(40e-06, 0.1)
alter c3 = limit(250e-12, 25e-12)
  *
  ac oct 100 250K 4Meg
  set run = "$&run" $ create a variable from the vector
  let K = db(v(out))
  write MonteCarlo_ac.txt v(out) K
  set appendwrite
  let run = run + 1
end $ loop ends here
```
Qucs-0.0.19S includes the first release of a GPL Verilog-A synthesis tool for compact device modelling.

- The Qucs-0.0.19S Verilog-A synthesizer is a basic working version of this new open source ECAD tool.
- It is for test purposes: bugs are likely and it may not be very stable.
- Generated synthesized Verilog-A code is relatively basic and has to be optimized manually for speed. However, it is expected that in the future its operation will improve as development of the Qucs synthesizer progresses.
- Circuits and Verilog-A synthesized models can be constructed from the following Qucs/SPICE built in components:
Data flow through the Qucs GPL compact device modelling tool set.

**QUCS FILTER SYNTHESIS**

- **Realization**: LC ladder (pi-type)
- **Type**: Bessel
- **Class**: Bandpass
- **Order**: 3
- **Fstart**: 1 GHz
- **Fstop**: 2 GHz
- **Impedance**: 50 Ohm

**VERILOG-A MODEL SYNTHESIS**

- Include "disciplines.vams"
- Include "constants.vams"
- module BPF2(P1, P2);
- inout P1, P2;
- electrical P1._net0L1, n1, P2._net0L2, _net0L3;
- analog begin
  @$(initial_model)
  begin
  end
  l(_net0L1) += ddt(V(_net0L1));
  l(_net0L1) += -ddt(V(P1));
  (P1) <= V(_net0L1)/((11.79*1e-20);
  (P1) <= ddt(V(P1)) * 1.64p;
  l(_net0L2) <= ddt(V(_net0L2));
  l(_net0L2) <= V(n1,P2);
  (n1,P2) <= V(_net0L2)/((7.723*1e-20);
  (P1,n1) <= ddt(V(P1,n1)) * 1.64p;
  l(_net0L3) <= ddt(V(_net0L3));
  l(_net0L3) <= -(P2);
  (P2) <= V(_net0L3)/((1.806*1e-20);
  (P2) <= ddt(-(P2)) * 7.014p;
  end
  endmodule

**QUCS/ADMS VERILOG-A "TURN KEY" COMPILER**

- Create circuit schematic and simulate

**DEVELOP TEST CIRCUIT, SIMULATE, AND EVALUATE OUTPUT DATA**

- Build Verilog-A module from subcircuit

- Edit text symbol

- Plotted and tabulated simulation data
Synthesis of a SPICE-like compact semiconductor diode model: static $I_d$ and dynamic capacitance model plus synthesized Verilog-A module code.
Synthesis of a SPICE like semiconductor diode model: simulated static and dynamic characteristics.
Verilog-A synthesis of a SPICE like semiconductor diode model: temperature effects

Qucs EDD diode model with temperature effects

```verilog
include "disciplines.vms"
include "constants.vms"
module EDDiode31(Poathode, Panode, Pnode, Pdode);
input Poathode, Panode; 
electrical Poathode, n2, n1, Panode, n4, n3;
parameter real Area=1.; parameter real I0=1e-14; parameter real Rs=0.1; parameter real N=1;
parameter real Temp=30;5; parameter real Vt=1.6; parameter real F0=0.5; parameter real M=0.5;
parameter real Cj=1e-12; parameter real Th=1e-9; parameter real Tn=26.85; parameter real Eg=1.11;
parameter real XT=2.0;
real RMAX, T1, T2, Cor, V2, F1, F2, A, B, E,g1, E,g2, V2, G2y2, I2;
begin 
@printall_mode 
begin 
RMAX=1e12; T1=273.15; T2=Temp+273.15; Con1=5*N*VI; Con2=2*F*I; VI=P*V/2; P; 
P1=Imag(V11/1-M1*)(1-Exp(1-M1*N1*F1)); F2=2*Exp(1-M1*N1*F1); F3=1-F2*(1-M1) 
A=7.0E-4; 
B=110; 
Eg1=1-Eg1*1/(1+T1); Eg2=2*Exp(A*1/T1/2); 
V2=2*T1(V11/T1)**2*Exp(1.5*N*1*1/T1)**2; V2=V2*1/V11; 
Is=2*Exp((1/3)*A)*2*1**1**2**1; 
I2=2*Exp((1/3)*A)*2*1**1**2**1; 
end 
endmodule
```
Verilog-A synthesis of a SPICE like semiconductor diode model: simulated $I_d - V_d$ temperature effects.

Simulation data for
Qucs EDD model and built-in diode model

Simulation data for
Verilog-A model and built-in diode model
Verilog-A synthesis of semiconductor device shot and flicker noise: EDD models and Verilog-A module code.

```
module Shot_NoiseR11(P1, P2, P3, P4);
input P1, P2, P3, P4;
input nShot, Sh, Ph, P1, P2, P3, P4;
output nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, n Shot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, n Shot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, n Shot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nShot, nSho...```
Verilog-A synthesis of semiconductor device shot and flicker noise: small signal AC domain simulation data.
**Verilog-A synthesis of multi-EDD models: EKV2p6 nMOS**

\[ I_{ds} = f(V_d, V_g, V_s, V_b) \]

model for a transistor operating in long channel mode.

```
```

Qucs EDD EKV2p6 \( I_{ds} = f(V_d, V_g, V_s, V_b) \) model

Synthesized EKV2p6 \( I_{ds} = f(V_d, V_g, V_s, V_b) \) Verilog-A code
Verilog-A synthesis of multi-EDD models: EKV2p6 nMOS

\[ I_{ds} = f(V_d, V_g, V_s, V_b) \] swept DC simulation data.

A comment on the Qucs simulation process:
Simple simulation run time tests indicate that the optimized EKV2p6 Verilog-A model simulation speed is at least 30X faster than the interactive EDD model.

1. At this stage in the development of the Qucs synthesizer optimized Verilog-A module code is done manually.
2. General procedure:
   2.1 Reduce current contribution statements to a minimum. This can be done by representing model equation quantities by real variables rather than internal node voltages.
   [one I(a) << ... in the EKV nMOS example]
   2.2 Eliminate as many as possible internal model nodes and remove current to voltage one Ohm conversion resistors.
   [zero left in EKV nMOS example]
Conclusion

Summary:

- Version 0.0.19 is a major release of the Qucs circuit simulator, updating the popular RF package while simultaneously adding a new software tool, Qucs 0.0.19S, which provides Qucs users with an experimental software package that links legacy Qucs with ngspice and Xyce GPL SPICE.

In the future the main Qucs development directions are likely to be:

- Further integration of Qucs with ngspice and Xyce: including improvement of the existing ngnutmeg support, an RFEDD synthesizer implementation, additional analysis support for SPICE .SENS and .PZ etc, and a range of new SPICE compatible components, for example magnetic core models.

- Improvements to the Verilog-A module synthesizer.

- Implementation of mixed signal simulation with ngspice/XSPICE and Xyce.

Qucs-0.0.19S-RC3 from

https://github.com/ra3xdh/qucs/releases/download/0.0.19S-rc3/qucs-0.0.19Src3.tar.gz (linux source)
https://github.com/ra3xdh/qucs/releases/download/0.0.19S-rc3/qucs-0.0.19Src3-setup.zip (Windows installer)