Modeling of GaN HEMTs With Open Source Qucs-S Circuit Simulation and Compact Device Modeling Technology

Mike Brinson $^1$, mbrin72043@yahoo.co.uk.
Vadim Kuznetsov $^2$, ra3xdh@gmail.com

$^1$Centre for Communications Technology, London Metropolitan University, UK

$^2$Bauman Moscow Technical University, Russia

Presented at IEEE EDS mini-Colloquium on GaN HEMT Technology, Lodz, 22 June 2016
Qucs-0.0.19-S-RC6 Simulation and Compact Device Modelling Tools
Introduction to the Qucs GPL Verilog-A Module Synthesizer
Qucs modelling of the ”Efficient Power Corporation (EPC)” GaN EPC2001 Power Transistor
Qucs Verilog-A Modeling of the ”MIT Virtual Source GaN-RF HEMT Compact Device Model 1.0.0”: Problems Simulating with ADMS; Workarounds and Typical Simulation Data
Qucs-0.0.19-S-RC6 XSPICE Code Modeling package
Qucs-0.0.19-S-RC6/Ngspice/Xyce Circuit Analysis and Compact Device Parameter Extraction from Manufacturers Data or Measurements Controlled by Octave Script Files
Summary
Qucs-0.0.19-S-RC6 includes the Qucs GPL Verilog-A synthesis tool for compact device modelling.

- The Qucs-0.0.19-S-RC6 Verilog-A synthesizer is a fully working version of this new open source ECAD tool.
- It is for test purposes: bugs are likely but it is now more stable than the initial release.
- Verilog-A device models and circuit macromodels can be synthesized from the following Qucs/SPICE built in components:
Structure:

- **Model Specification**
  - Qucs EDD based compact models
  - Device physical and system equations
- **Synthesis**
  - SPICE netlist code
  - ngspice/XYce netlist synthesiser
  - Verilog-A module synthesiser
- **Simulation**
  - ngspice
  - Xyce
  - Verilog-A to C++ synthesizer
  - Quacs/ADMS “Turn Key” Verilog-A to C++ synthesizer
- **Visualisation**
  - Qucs/AEVI model visualization: Qucs and Octave routines
  - Qucs schematic capture
  - Qucs and netlist models
  - Qucsat/XYce simulation
  - EDD, RCL and other components
  - New component model
  - C++ code

**KEY**
- Part of Qucs-0.0.19/S release.
- External GPL SPICE circuit simulators.
- Qucs synthesised SPICE netlist and Verilog-A module code.
- ADMS-2.3.4 Analogue Device Model Synthesiser.
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

**QUCS/ADMS VERILOG-A "TURN KEY" COMPILER**
- Include "disciplines.vams"
- Include "constants.vams"
- module BPF2(P1, P2);
- inout P1, P2;
- electrical P1._net0L1, n1, P2._net0L2, _net0L3;
- analog begin
- @t(initial_model)
- begin
- end
- ![Verilog code snippet]

**VERILOG-A MODEL SYNTHESIS**
- ![Verilog code snippet]

**DEVELOP TEST CIRCUIT, SIMULATE, AND EVALUATE OUTPUT DATA**
- Create circuit schematic and simulate
- ![Simulation data]
- ![Plot of simulation data]
Synthesis of a SPICE-like compact semiconductor diode model: EDD static \( I_d \) and dynamic capacitance model \(-\) \( \rightarrow \) synthesized Verilog-A module code.

\[
\begin{align*}
D1 &= \text{Area} \times \text{I}_{\text{ls}} \times (\text{exp}(V1/(N \times Vt))-1) \\
Q1 &= V1 - (V2 \times \text{Area} \times C0 \times Vj \times (1-M) \times (1-\text{exp}(1-M \times \text{ln}(1-Vj)))) - (Tt \times V2 \times \text{Area} \times C0 \times (F1+1/F2) \times (F3 \times V1 \times Fc \times Vj) + (M \times (2 \times Vj)) \times (V1 \times V1 \times Fc \times Vj \times Vj)) \\
I2 &= 0 \\
Q2 &= 0
\end{align*}
\]
Introduction to the Qucs GPL Verilog-A Module Synthesizer: Part V

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

**Equation**

\[
\begin{align*}
V_T &= \frac{N\Phi}{k}\ln(1 + \frac{1}{1 + \exp(\frac{1}{V} - V_T)}) \\
F2 &= \exp(1 + \frac{1}{1 + \exp(\frac{1}{V} - V_T)}) \\
F3 &= 1 + \frac{1}{1 + \exp(\frac{1}{V} - V_T)}
\end{align*}
\]

**Synthesized Verilog-A code**

```verbatim
module EDDiode1 (Po, Rd, Rs, n); 
#(parameter real Temp = 273.15, \( N \Phi \) = 4.3e-20, k = 25.7, V_T = 26, a = 1.6, b = 1) 
real V, F1, F2, F3, A, Q; 
\n\begin{align*}
F2 &= \exp(1 + \frac{1}{1 + \exp(\frac{1}{V} - V_T)}) \\
F3 &= 1 + \frac{1}{1 + \exp(\frac{1}{V} - V_T)}
\end{align*}

\begin{align*}
\text{dd} &= \frac{\text{d}V}{\text{d}t} \quad \text{\textbf{Verilog-A}} \quad \text{\textbf{Qucs}}
\end{align*}
\]

**Qucs EDD diode model with temperature effects**
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.

Synthesized Verilog-A module code

```
#include "disciplines.vams"
#include "constants.vams"
module Shot_NoiseR11(P1, P2, P3, P4);
  inout P1, P2, P3, P4;
  electrical nShot, P2, P1, P3, P4;
  analog begin
    @(initial_model)
    begin
      I(nShot) <+ (V(nShot))/1;
      I(nShot) <+ white_noise(1,"shot");
      I(P2,P1) <+ V(P2,P1)/(-1e-3);
      I(P3,P4) <+ sqrt(2*P0*(V(P1,P2)*1e3+1e-20))/V(nShot);
    end
  endmodule
```

---

```
#include "disciplines.vams"
#include "constants.vams"
module Flicker_NoiseR11(P1, P2, P3, P4);
  inout P1, P2, P3, P4;
  electrical P2, P1, nFlicker, P3, P4;
  parameter real Kf=1e-12;
  parameter real Ffe=1;
  parameter real Af=1;
  analog begin
    @(initial_model)
    begin
      I(P2,P1) <+ V(P2,P1)/(-1e-3);
      I(nFlicker) <+ flicker_noise(Kf, Ffe, "flicker");
      I(nFlicker) <+ (-V(nFlicker))/1;
      I(P3,P4) <+ sqrt(exp(Af*ln((V(P1,P2)*1e3+1e-30)))/V(nFlicker));
    end
  endmodule
```

---

Noise model symbols

---

Verilog-A synthesis of semiconductor device shot and flicker noise: EDD models and Verilog-A module code.
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.
Verilog-A synthesis of multi-EDD models: EKV2p6 nMOS

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

**Parameter sweep**

- **Eqn3**
  - Type = in
  - Param = Vds
  - Start = 0
  - Stop = 5
  - Points = 32

- **Eqn2**
  - Type = in
  - Param = Vgs
  - Start = 0
  - Stop = 3
  - Points = 6

**Equation**

\[ I_{ds\_VA} = Pr2.I \]
Qucs modelling of the "Efficient Power Corporation (EPC)" GaN EPC2001 Power Transistor: Part I EDD Subcircuit Compact Device Model


Ported from the EPC mathematical model common to LTSPICE, PSPICE, TSPICE and Spectra netlist models.
Qucs Modeling of the "Efficient Power Corporation (EPC)" GaN EPC2001 Power Transistor: Part II DC Test Bench and Typical Simulation Curves

Parameter sweep

Vgs - Gate to Source Voltage (V)

- SW1
  - Sim=SW2
  - Type=lin
  - Param=Vgs
  - Start=2.0
  - Stop=4.0
  - Points=5

EPC2001 Transfer Curve

- Vgs - Gate to Source Voltage = 3V

Output Curves for EPC2001 model

- Id - Drain Current (A)
- Vds - Drain to Source Voltage (V)

Parameter sweep

- SW2
  - Sim=DC1
  - Type=lin
  - Param=Vds
  - Start=0
  - Stop=3
  - Points=201

Forward and Reverse Current at Vdd/2 (V)

- Vdd (V)
- Current (A)

```
*include "disciplines.varns"
*include "constants.varns"
module EPC2001n(Pn, nPd, nPg):
  input nPd, nPg, nPn, electrical ng, ns, nd, npn, nPd, nPg;
  parameter Temp=25; parameter real A1=41.7996; parameter real aTo=5.456026e-3;
  parameter real k2=2.2599866; parameter real k3=1.2e-1;
  parameter real Awg=1077; rpara=4.453059e-3;
  x0=0.975; x0_1=1.10;
  ax07c=0.75e-4; dgs1=4.4e-7;
  dgs2=2.6e-13; dgs3=3.8;
  dgs4=0.23;
  ags1=8.6952e-10; ags1=1.418e-11;
  asd1=3.362e-10; asd2=6.92e-10;
  asd3=1.289e-10;
  asd4=2.296;
  asd5=5.218e-10; asd6=4.59991e-10;
  asd7=2.6038e-10; asd8=2.1475e-10;
  asd9=3.89;
  asd10=5.9551;
  asd11=5.3168e-10; asd12=1.997;
  asd13=2.8377e-10; asd14=1.4751e-10;
  asd15=7.5163;
  asd16=7.121;

  // EPC2001 Synthesized Verilog-A module code
  
  EO1=A1*(1-(x0/Temp)); Rs=1+x011/11; Vgsd=1+x011/11; Vgsd=0.75 para;
  Rs=25 para; A2=0.8*1077/awg; E25=1.6*aTo/Tem/(Temp-25); E25=0.5/awg;
  end
```

Build Verilog-A model from subcircuit
Qucs Modeling of the "Efficient Power Corporation (EPC)" GaN EPC2001 Power Transistor: Part IV AC Gate Matching Network Test Bench and Typical Simulation Results
Qucs Modelling of the "Efficient Power Corporation (EPC)" GaN EPC2001 Power Transistor: Part V Switching Response Test Bench and Typical Simulation Results

Parameter sweep
- SW1: TR1
- Sim: TR1
- Type: in
- Param: Ld
- Start: 0 n
- Stop: 400 n
- IntegrationMethod: Gear
- Order: 4
- MinStep: 1e-20
- MaxIter: 1500
- abs tol: 100 pA
- rel tol: 100 uV
- initialDC: no
The Analogue Device Model Synthesizer (ADMS) version 2.3.5 is used by Qucs, Ngspice, Xyce and GnuCap GPL circuit simulators for Verilog-A compact semiconductor device modeling.

ADMS is based on a subset of Verilog-A HDL selected for compact device modeling.

Although the Verilog-A HDL is standardised there is no guarantee that individual simulator implementations allow the same dialect of Verilog-A for modeling purposes, for example Qucs Verilog-A models can include component noise while Ngspice does not implement thermal, shot or flicker noise.

Normally emerging technology Verilog-A compact models have to be modified, often by hand, to compile without error: specific areas which can cause problems are

- Internal node collapsing,
- Voltage limiting,
- Setting initial conditions,
- Model equations that include complex combinations of analogue functions,
- Thermal effects due to power dissipation.
ADMS parameter statements DO NOT ALLOW reference to previously defined model parameters.

```
parameter real vxord = 1.30e7 from [0;Inf];  // Source Injection velocity [cm/s]
parameter real VcorD = -2.0;                  // Threshold voltage of drain access transistor[V]
parameter real Cgd = 5.0e-7 from [0;Inf];      // Drain access area capacitance [F/cm²]
parameter real delta1rd = 1.3 from [0;Inf];    // DIBL for drain access transistor
parameter real delta2rd = 0.30 from [0;Inf];   // DIBL for drain access transistor
parameter real Vdibsat = 2.0 from [0;Inf];     // DIBL for drain access transistor
parameter real Srd = 0.35 from [0;Inf];        // Subtheshold slope for drain access transistor [V/Dec]
parameter real zeta = 0.0 from [0;Inf];       // Self heating parameter (scalable)
parameter real beta1rd = 1.3 from [0;Inf];    // Linear to saturation transition parameter
parameter real vthard = 0.05 from [0;Inf];    // Scattering velocity reduction parameter with Vg
parameter real nrd = 0.80 from [0;Inf];       // Punchthrough factor affects slope change in subthreshold

parameter real vxors = vxord from [0;Inf];    // Source Injection velocity [cm/s]
parameter real Vfors = VcorD;                  // Threshold voltage of drain access transistor[V]
parameter real Cgdr = Cgd from [0;Inf];        // Drain access area capacitance [F/cm²]
parameter real delta1rd = delta1rd from [0;Inf]; // DIBL for drain access transistor
parameter real delta2rd = delta2rd from [0;Inf]; // DIBL for drain access transistor
parameter real Srd = Srd from [0;Inf];        // Subtheshold slope for drain access transistor [V/Dec]
parameter real vthard = vthard from [0;Inf];  // Scattering velocity reduction parameter with Vg
parameter real nrd = nrd from [0;Inf];        // Punchthrough factor affects slope change in subthreshold
parameter real beta1rd = beta1rd from [0;Inf]; // Linear to saturation transition parameter
```
ADMS DOES NOT ALLOW voltage contributions of the form $V(n) < +I(n) \times R$, where $R$ is a resistance in $\Omega$,

- OR statements of the form $V(n) < +0.0$,

- Resistors, for example 0.001$\Omega$, are used to short nodes (node collapsing), with $I(n) < +V(N)/0.001$. 

**ADMS synthesis/compile error**

```verilog
if (Rsh > 1e-3 && Ls > 0)
    I(si,src) <+ Idrsrs;
else
    V(src,si) <+ 0;
//Source side contact resistance
if (Rc > 0) begin
    I(src,s) <+ V(src,s) / (Rc / Wg);
else begin
    V(src,s) <+ 0;
end
```

**OK**

```verilog
if (Rsh > 1e-3 && Ls > 0)
    I(si,src) <+ Idrsrs;
else
    I(si,src) <+ V(si, src)/1e-3;
//Source side contact resistance
if (Rc > 0) begin
    I(src,s) <+ V(src,s) / (Rc / Wg);
else begin
    I(src,s) <+ V(src,s)/1e-3;
end
```
Qucs Verilog-A Modeling of the "MIT Virtual Source GaN-RF HEMT compact model 1.0.0": Problems Simulating with ADMS; Workarounds and Typical Simulation Data - Part IV DC Characteristics

**Parameter sweep**

- **SW1**
  - Sim=SW2
  - Type=lin
  - Param=Vgs
  - Start=6
  - Stop=0
  - Points=9

- **SW2**
  - Sim=DC1
  - Type=lin
  - Param=Vds
  - Start=0
  - Stop=3
  - Points=201

**Equation**

Eqn2

- \( gds_{\text{norm}} = \text{diff}(\text{ids}_{\text{norm}}, V_{\text{ds}}) \)
- \( \text{ids}_{\text{norm}} = \text{Pr}_{\text{Ids}}/25e-3 \)

**dc simulation**

- DC1
  - abstol=1 pA
  - vntol=10 uV
  - MaxIter=1500
Qucs Verilog-A Modeling of the "MIT Virtual Source GaN-RF HEMT compact model 1.0.0": Problems Simulating with ADMS; Workarounds and Typical Simulation Data - Part V Simulating Thermal self-Heating Effects Induced by Internal Power Dissipation

- The ADMS dialect of Verilog-A does not implement the pwr(dt) statement,
- Device self-heating is often modelled with a parallel RC network where the volt drop across the RC combination represents the change in device temperature due to internal power dissipation,
- \( T_{th} = R_{th} \cdot P_d + \text{Temp}(P_d = 0) \), where \( T_{th} \) is the device temperature at power dissipation \( P_d \) (W).

```verilog
// Self-heating
I(PT) <= ddt( Cth * V(PT) );
I(PT) <= - (I(di,s) * V(di,s) + I(d,drc) * V(d,drc) + I(src,s) * V(src,s) + V(drc,di) * I(drc,di) + V(src,si) * I(src,si) );
I(PT) <= V(PT)/Rth;
```

\( T_{th} \) = \( R_{th} \cdot P_d + \text{Temp}(P_d = 0) \)
Qucs Verilog-A Modeling of the "MIT Virtual Source GaN-RF HEMT compact model 1.0.0": Problems Simulating with ADMS; Workarounds and Typical Simulation Data - Part VI Variation of Thermal Resistance Rth and its Effect on DC Characteristics

Equation
Eqn2
\[ \text{l}_{\text{ds norm}} = \text{Pr}_{\text{l}_\text{ds}}, 1.25 \times 10^{-3} \]
\[ \text{l}_{\text{ds norm2}} = \text{Pr}_{\text{l}_\text{ds1}}, 1.25 \times 10^{-3} \]
\[ \text{l}_{\text{ds norm3}} = \text{Pr}_{\text{l}_\text{ds2}}, 1.25 \times 10^{-3} \]
\[ \text{l}_{\text{ds norm4}} = \text{Pr}_{\text{l}_\text{ds3}}, 1.25 \times 10^{-3} \]
\[ \text{Temp1} = \text{P}B\text{Vs}(\text{nPT1}. \text{V} + 2.7, \text{Pr}_{\text{l}_\text{ds}}.1* \text{V}ds) \]
\[ \text{Temp2} = \text{P}B\text{Vs}(\text{nPT2}. \text{V} + 2.7, \text{Pr}_{\text{l}_\text{ds1}}.1* \text{V}ds) \]
\[ \text{Temp3} = \text{P}B\text{Vs}(\text{nPT3}. \text{V} + 2.7, \text{Pr}_{\text{l}_\text{ds2}}.1* \text{V}ds) \]
\[ \text{Temp4} = \text{P}B\text{Vs}(\text{nPT4}. \text{V} + 2.7, \text{Pr}_{\text{l}_\text{ds3}}.1* \text{V}ds) \]

Parameter sweep
- SW2
- Sim=DC1
- Type=lin
- Param=Vds
- Start=0
- Stop=3
- Points=201

DC simulation
- absetol=1 pA
- vntol=10 uV
- MaxIter=1500
Qucs-0.0.19-S-RC6 includes for the first time a "turn-key" XSPICE Code Modelling package for use with the Ngspice and SPICE OPUS circuit simulators,

Qucs-0.0.19-S-RC6 is being extended to include a new Qucs/Octave integrated tool set for compact device model and circuit macromodel parameter extraction with data fitting and optimization using measured, or manufacturer’s published device data, and simulated circuit data - this new feature is experimental, but should become more stable during the summer 2016 development period.
Qucs-0.0.19-S-RC6 XSPICE Code Modeling package: Part I XSPICE Code Model Subcircuits
The "XSPICE generic device" component is the foundation for:

- Precompiled XSPICE device (*.cm) library support, and

- Dynamic XSPICE Code Models compilation system which allows Code Model sources to be attached to a schematic and compiled automatically at simulation time.

Precompiled Code Model *.cm library attachment data flow diagram:

- Schematic
- Precompiled CodeModel library
- XSPICE Generic device
  - spice_netlist()
- .MODEL
- Component
  - spice_netlist()
- SPICE Netlist
  - .spinit file
- Ngspice simulator
- Qucs GUI
  - Data visualization
The "XSPICE generic device" component is a building block for the construction of user-defined A-devices. It is defined by a comma separated port list, with allowed XSPICE port designators, then attached to a SPICE .MODEL statement.

Source code: cfunc.mod file

```c
void cm_ggain(ARGS)
{
    Mif_Complex_t ac_gain;
    if(ANALYSIS != MIF_AC) {
        OUTPUT(out) = PARAM(out_offset)+PARAM(gain)*
                  (INPUT(in)+PARAM(in_offset));
        PARTIAL(out,in) = PARAM(gain);
    } else {
        ac_gain.real = PARAM(gain);
        ac_gain.imag= 0.0;
        AC_GAIN(out,in) = ac_gain;
    }
}
```
Qucs-0.0.19-S-RC6 XSPICE Code Modeling Package: Part IV XSPICE "Turn-Key" Model Generation; Compiler System Dataflow Diagram

- cfunc.mod
- ifspec.ifs
- XSPICE CodeModel
- .MODEL
- XSPICE Generic device
- spice_netlist()
- Component
- spice_netlist()

Extract CodeModels

*.mod and *.ifs

$WORKDIR/qucs_cmlib/

Make rules file *.mk

Compilation

SPICE Netlist

qucs_xspice.cm

.datainit file

Qucs GUI

Data visualization

Ngspice simulator

CodeModels library

source tree

XSPICE Generic device

spice_netlist()
Qucs-0.0.19-S-RC6 XSPICE Code Modeling Package: Part V XSPICE Diode Model - (a) The Qucs-S subcircuit Symbol and Model Circuit

XSPICE diode model based on:
Qucs-0.0.19-S-RC6 XSPICE Code Modeling Package: Part V XSPICE Diode Model - (b) The XSPICE Diode/func.mod Code

/*
   diode cm model.  4 March 2016  Mike Brinson

   This file contains the mode code for an experimental semiconductor diode model. This
   is used as a test bench for constructing compact device models using the Qucs-0.0.19-S automatic XSPICE CodeModel compiler system.

   This is free software; you can redistribute it and/or modify
   it under the terms of the GNU General Public License as published by
   the Free Software Foundation; either version 2, or (at your option)
   any later version.
*/
#define DERIVE 0
#include <math.h>
void cm_diode(ARGS)
{
    double Vt, temp, Vd, P1, P3, P4, PTNOM, PTEMP;
    double PIS, PAREA, PXTI, PEG, PN;
    double Tr, Is_temp, Id;
    double *derive;
    double exp80 = 5.4066334e34;
    double GMIN = 1e-12;

    PTNOM = PARAM(nom)+273.15;
    PTEMP = TEMPERATURE+273.15;
    Vt=temp = 8.65387156e-5*PTEMP;
    PEG = PARAM(eg);
    PIS = PARAM(is);
    PN = PARAM(n);
    PAREA = PARAM(area);
    PXTI = PARAM(xti);
    P1 = 1/(P1*Vt*temp);
    P4 = exp80;
    P3 = exp80;

    #ifndef INIT
          cm_analog.albof(DERIVE, sizeof(double));
          derive = (double *)&cm_analog.dev_ptr(DERIVE, 0);
          *derive = 0.0;
    #endif

    if (Vd > P3*Vt*temp) {
        P1 = exp(P1*Vd)-1.0 + GMIN * Vd;
        OUTPUT(diode) = Is_temp*exp|P1*Vd|+GMIN;
        PARTIAL(diode, diode) = *derive;
    } else {
        id = Is_temp*exp80*(1+(P1*Vd-80))|GMIN*Vd;
        OUTPUT(diode) = id;
        *derive = P1*P4*GMIN;
        PARTIAL(diode, diode) = *derive;
    }

    if (Vd <= P3*PN*Vt*temp)
        OUTPUT(diode) = id;
        *derive = GMIN;
        PARTIAL(diode, diode) = *derive;
}


Semiconductor diode non-linear I_d / V_d characteristics, including Verilog-A limexp function and temperature effects.
Qucs-0.0.19-S-RC6 XSPICE Code Modeling package: Part VI XSPICE diode model - (c) The XSPICE Non-Linear Diode Capacitance dnlcap/func.mod Code

```
* dnlcap cm model 4 March 2010 Mike Srinivasan
This file contains the model code for an experimental
semiconductor diode capacitance: both Gdc and Gdf are modelled.
This is used as a test bench for constructing compact device models
with the Qucs-0.0.19-S automatic XSPICE Code/Model compiler system
This is free software; you can redistribute it and/or modify
it under the terms of the GNU General Public License as published by
the Free Software Foundation; either version 2, or (at your option)
any later version.
*
#define CV0 0
#include cnash.h4
void cm_dnlcap(ARG3s)
| Complex_t ac; gain;
  static double PCJ0, PVJ, PM, PFC, PTI, PIS, PN;
  double PI, V0, partial, Vtemp;
  double PTEMP, WI, Wi, Wd, Rd;
  double Vd;
  static double cap, F2, F3, cdc, derive, Id, PI, Pi;
  double Rp = 1.0e12;
  double expg0 = 0.5918340624;
  double GMIN = 1e-12;
  PTEMP = TEMPERATURE / 273.15;
  Vtemp = 8.65368719e-5*PTEMP;
  PI = 1/(Vtemp);
  #INIT(1) {
    cm_dnlcap.aloc(CVC, size0(double));
    cvo = (double *) cm_dnlcap.get_ptr(CVC, 0);
    *cvo = 0.0;
    cap = 1e-10;
    derive = 1e-10;
    PCJ0 = PARAM(0);
    PVJ = PARAM(1);
    PM = PARAM(2);
    PFC = PARAM(3);
    PTI = PARAM(4);
    PIS = PARAM(5);
    PN = PARAM(6);
    F2 = exp(1+PM*log(1-PFC));
    F3 = 1-PFC*(1-PM);
    PI = 1e-10
    P4 = 1e-10;
    }
    }
  if (ANALYSIS = DC) {
    cvo = INPUT(dnlcap)/cap;
    OUTPUT(dnlcap) = "cvo;-
    PARTIAL(dnlcap, dnlcap) = Rp;
    }
    if (ANALYSIS = TRANSIENT) {
      cm_dnlcap.integrate(dnlcap)/(cap + 1e-17), cvo, &paren);
      partial = cap;
      OUTPUT(dnlcap) = "cvo;-
      PARTIAL(dnlcap, dnlcap) = partial;
    }
    if (ANALYSIS = AC) {
      Rd = 1/(derive);
      WI = 1/Rd*REO1*REO2*REO3*REO4*cap;
      WI = Rd/REO1; REO2*REO3*REO4*cap;
      ac.gain.real = Rd;
      ac.gain.real = 1.0/RI;
      AC = GAIN(dnlcap, dnlcap) = ac.gain;
    }
```
Qucs-0.0.19-S-RC6 XSPICE Code Modeling Package: Part VII XSPICE Diode Model - (d) The Diode Small Signal AC performance; Y parameter, Rd and Cd Extraction

ac simulation

Paramter sweep

\[ V2 = \text{DC 0.8 AC 0.1} \]

\[ V1 = \text{DC 0.8 AC 0.1} \]

\[ \text{Params-V1} \]

\[ \text{Type=lin} \]

\[ \text{Start=0} \]

\[ \text{Stop=6.8} \]

\[ \text{Points=21} \]

\[ \text{gm}_d_1 \]

\[ \text{ls}=1e-15 \]

\[ \text{a}=1 \]

\[ \text{th}=27 \]

\[ \text{area}=1 \]

\[ \text{rs}=0.1 \]

\[ \text{n}=3 \]

\[ \text{eg}=1.11 \]

\[ \text{q}=2e-12 \]

\[ \text{m}=0.7 \]

\[ \text{b}=0.5 \]

\[ \text{t}=1e-10 \]

\[ \text{temp}=27 \]
Qucs-0.0.19-S-RC6 XSPICE Code Modeling Package: Part V XSPICE Diode Model - (e) The Diode Id/Vd Temperature Variation

**dc simulation**

**Parameter sweep**

**Graph:**
- **Axes:**
  - x-axis: temp-sweep
  - y-axis: ngs/psi/vwrd
- **Graph Data:**
  - Points=101
  - Start=-20
  - Stop=80
  - Sim=DC1
  - Type=lin
  - Param=temp

**Diagrams:**
- **Diagram 1:**
  - Symbol: ns
  - Symbol: nd
  - Node: PrId
  - Symbol: XSPICE
- **Diagram 2:**
  - Symbol: V1
  - Symbol: CM_D1
  - Symbol: V= DC 0.6 AC 0.1
  - Param:
    - is=1e-14
    - n=1
    - tnom=27
    - area=1
    - rs=0.01
    - xti=3.0
    - eg=1.11
    - temp=27
    - cj0=1e-12
    - vj=0.7
    - m=0.5
    - fc=0.5
    - tt=1e-10
Qucs-0.0.19-S-RC6/Ngspice/Xyce Circuit Analysis and Compact Device Parameter Extraction from Manufacturer’s Data or Measurements Controlled by Octave Script Files: Part I Structure Diagram
The main purposes of Octave integration are:

- Parameter substitution in Qucs and SPICE netlists,
- Simulation process control from Octave,
- Simulator output dataset (SPICE3f5-raw and Qucs XML) loading into Octave matrix structures,
- Compact model parameter extraction from simulation and manufacturer’s, or measured, data using curve fitting and optimization with the ASCO package.

Example Octave package functions:

- `subs_spice_netlist(FILE, PARAM, VALUE),`
- `subs_qucs_netlist(FILE, PARAM, VALUE),`
- `subs_spice_model_netlist(FILE, MODEL, PARAM, VALUE),`
- `DATA = read_spiceraw(FILE).`

Where FILE – represents SPICE or Qucs netlist files
PARAM – represents a SPICE or Qucs variable or .MODEL parameter name
VALUE – represents a parameter value to replace its original quantity
Qucs-0.0.19-S-RC6/Ngspice/Xyce Circuit Analysis and Compact Device Parameter Extraction from Manufacturer’s Data or Measurements Controlled by Octave Script Files: Part III Simple Ngspice example
Qucs-0.0.19-S-RC6/Ngspice/Xyce Circuit Analysis and Compact Device Parameter Extraction from Manufacturer’s Data or Measurements Controlled by Octave Script Files: Part IV Simple Qucs example

transient simulation

Equation
Eqn1
Cp=1000pF
Rs=1k

RCQ.sch

RCQ.m

# Qucs 0.0.19 RCQ.sch
R: R1 in out R="Rs" Temp="26.85" Tc1="0.0" Tc2="0.0" Tnom="26.85"
Eqn: Eqn1 Cp=1000pF Rs=1k Export="yes"
Vcc: V1 in gnd U=1 V f=10 kHz Phase="0" Theta=0"

TR1 Type="lin" Start=0 Stop=0.5 ms Points="1001" IntegrationMethods="Trapezoidal" Order="2"
InitialStep="1 ns" MinStep="1e-16" MaxIter="150" reltol="1e-001" abstol="1 pA" vntol="1 uV"
Temp="26.85" LTEntol="1e-03" LTEabstol="1e-06" LTEfactor="1.0" Solver="CroutLU"
relaxTSR="no" initIC="yes" MaxStep="0"
C: C1 out G="Cp" V=0"

Octave-Log

Figure 1

Rs = 2k
Cp = 5000pF
This presentation has attempted to show that open source compact modelling technology offers engineers and scientists viable tools for investigating the properties of emerging technology devices at a cost which is acceptable to all,

Verilog-A models for GaN RF and power devices have been introduced and the problems involved in evaluating their performance demonstrated with a series of circuit simulation test benches,

A short outline to the current state of Qucs-S development provided those attending the IEE EDS mini-Colloquium with a brief look at possible future directions in GPL circuit simulation and compact device modelling.

Linux and Windows versions of Qucs-0.0.19-S-RC6 can be downloaded from: https://github.com/ra3xdh/qucs/releases/tag/0.0.19S-rc6

Documentation is available here: https://qucs-help.readthedocs.org/en/spice4qucs/

Octave packages from https://github.com/ra3xdh/octave-circuittools/