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

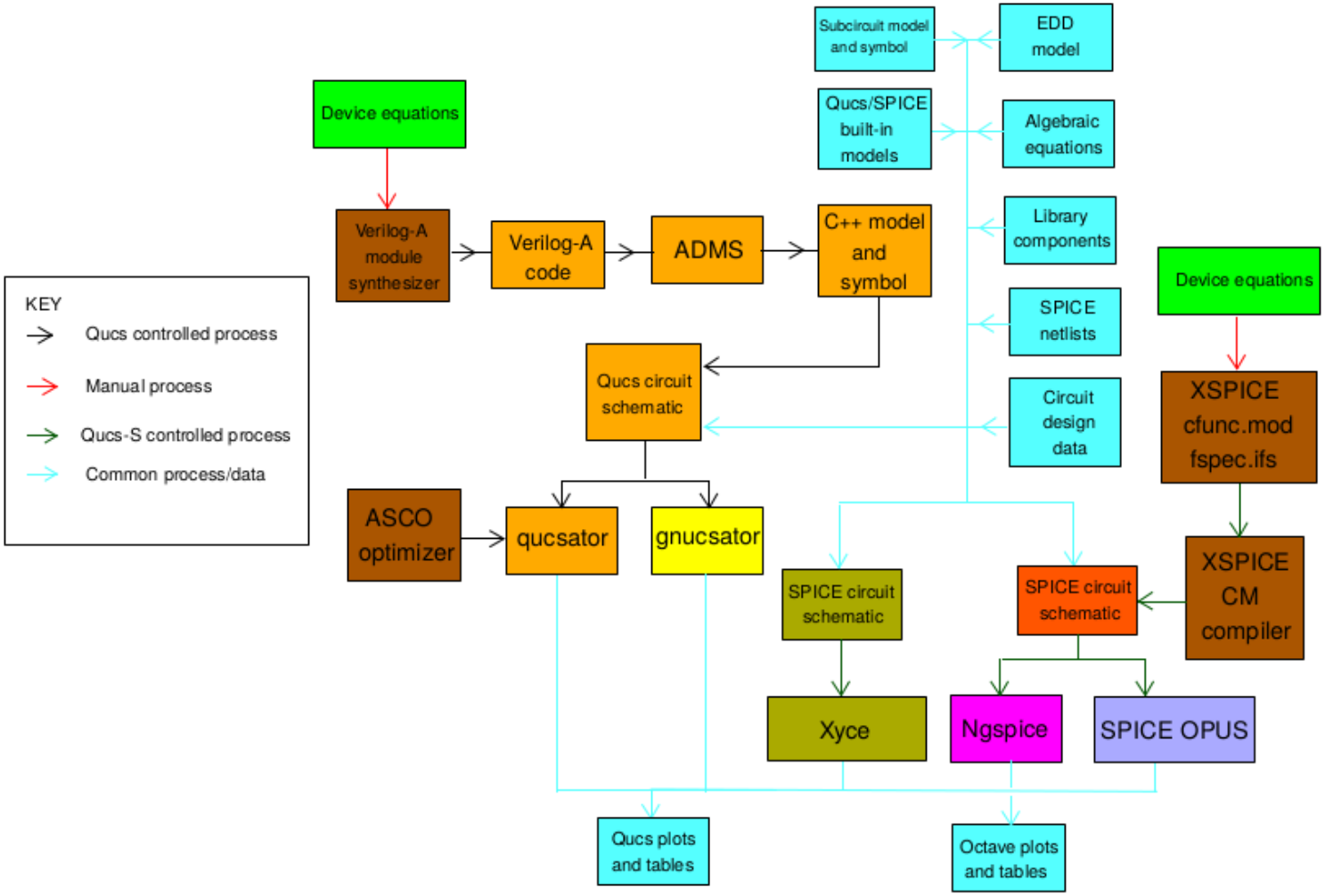
[2]Bauman Moscow Technical University, Russia

Presented at IEEE EDS mini-Colloquium on GaN HEMT Technology , Lodz, 22 June 2016

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

- 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

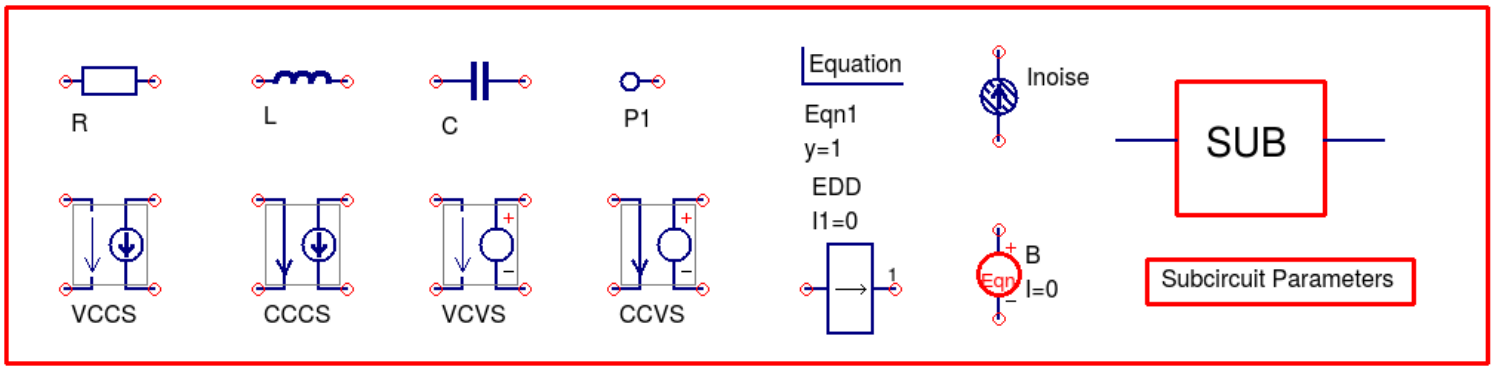
Qucs-0.0.19-S-RC6 Simulation and Compact Device Modelling Tools



Introduction to the Qucs GPL Verilog-A Module Synthesizer: Part I

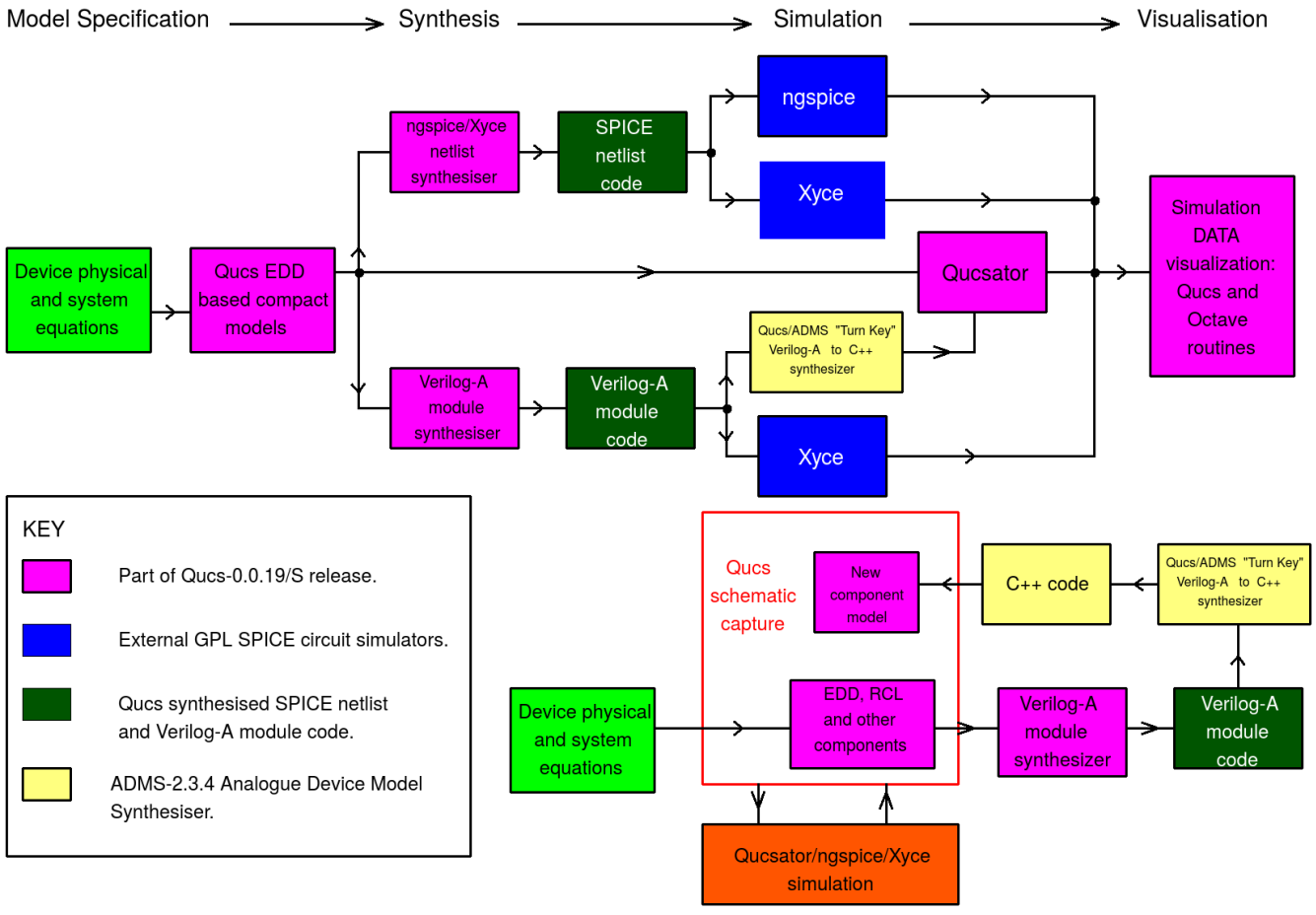
Qucs-0.0.19-S-RC6 includes the Qucs GPL Verilog-A synthesis tool for compact device modelling.

- 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:

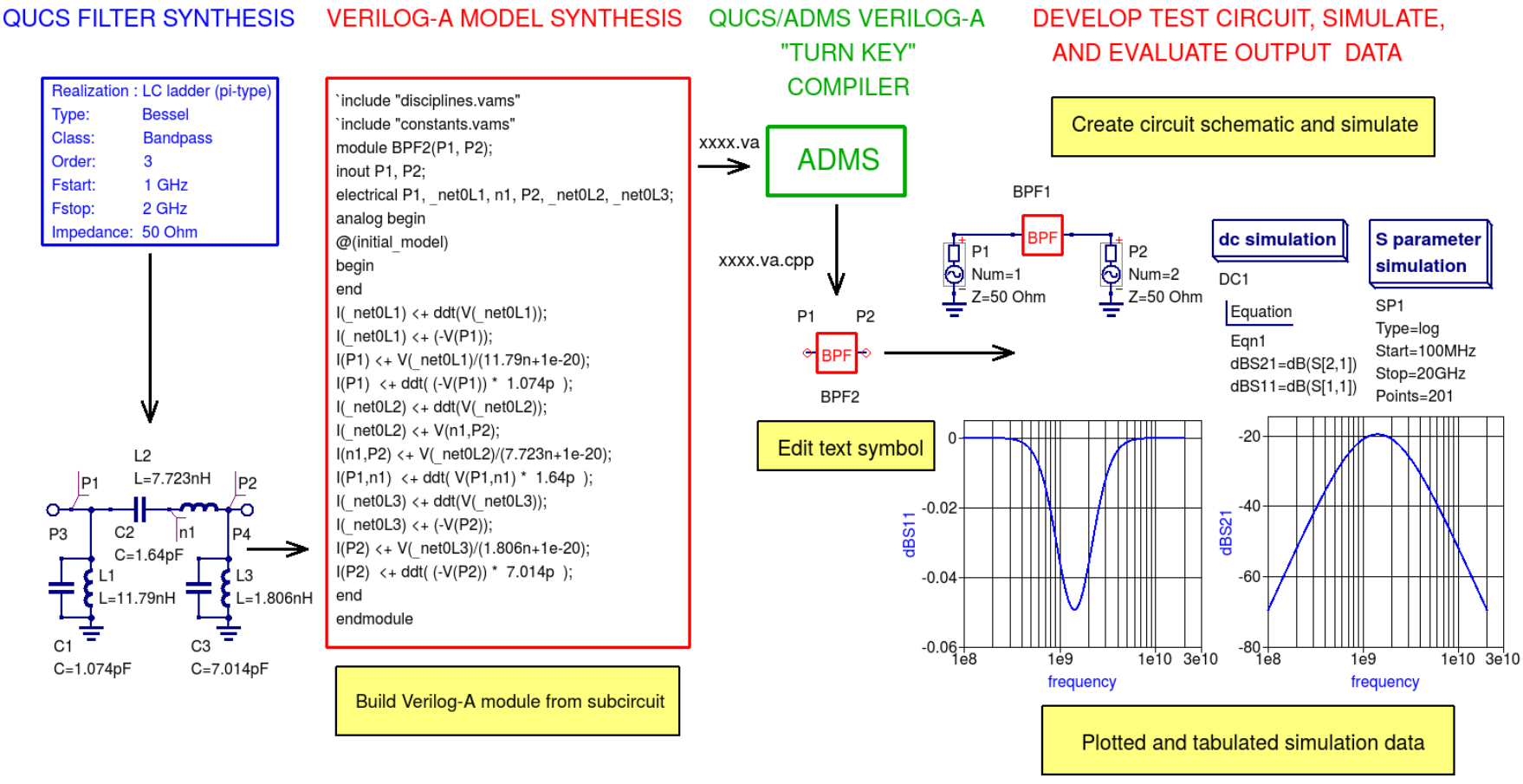


Introduction to the Qucs GPL Verilog-A Module Synthesizer: Part II

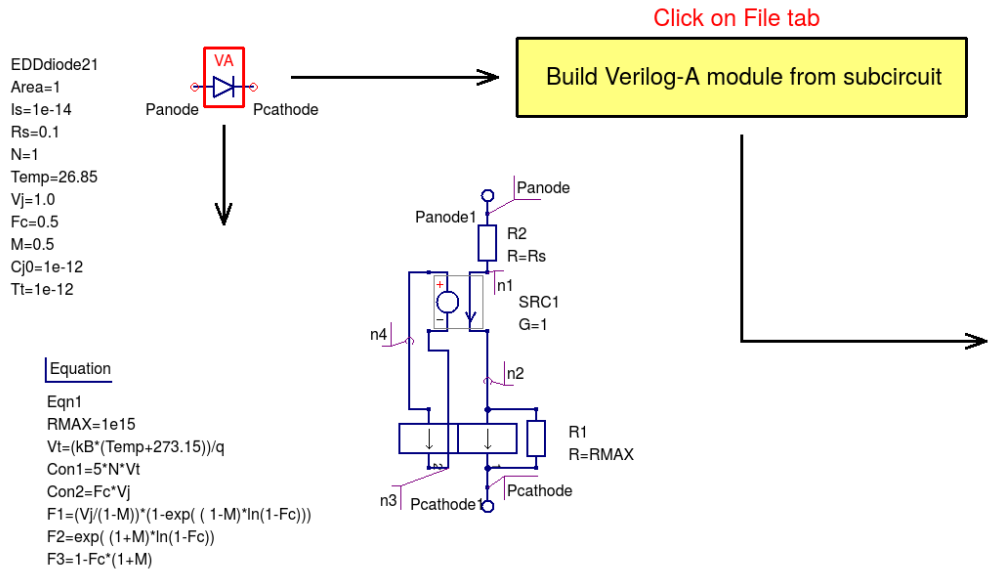
Structure:



Data flow through the Qucs GPL compact device modelling tool set.



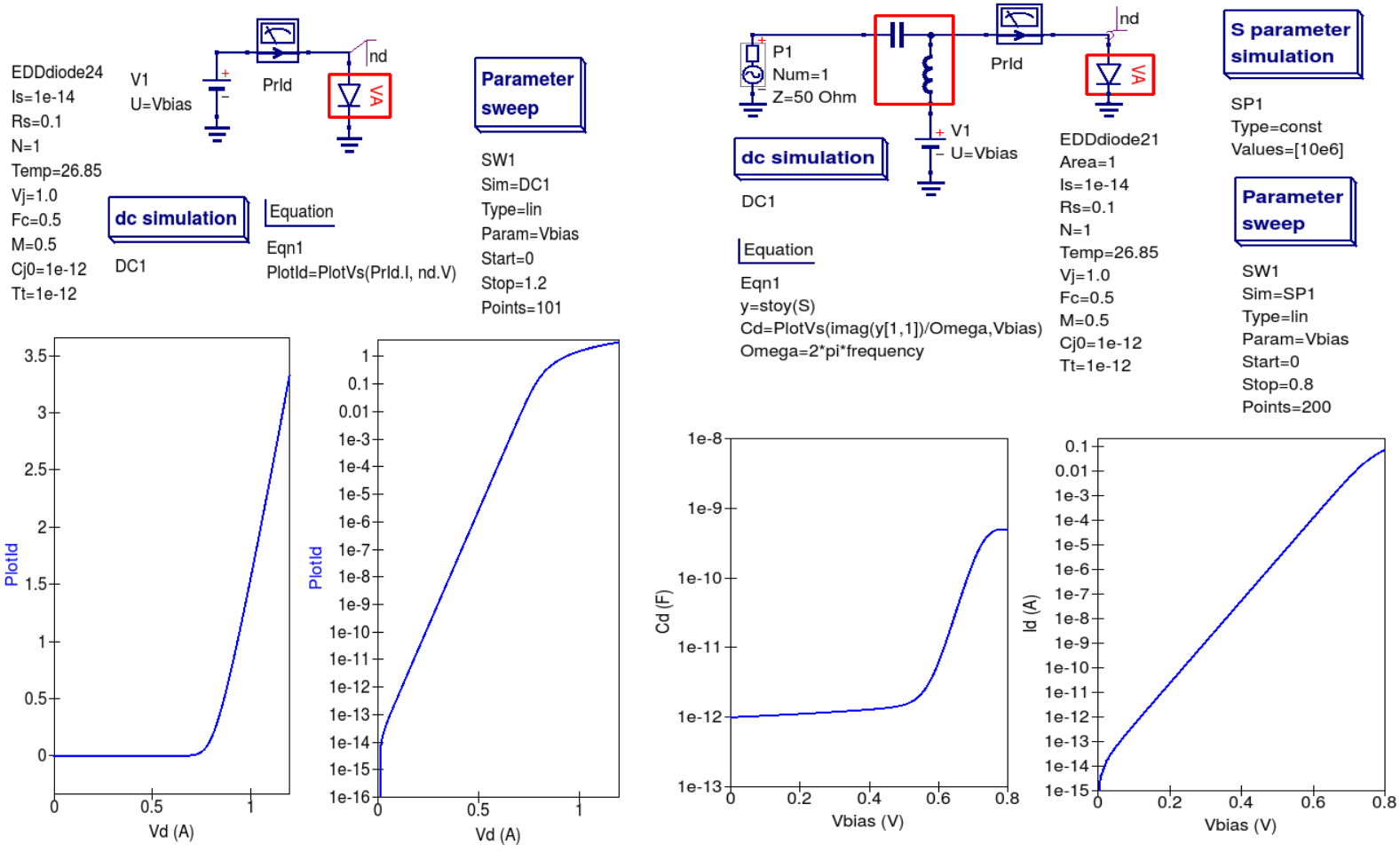
Synthesis of a SPICE like compact semiconductor diode model: EDD static I_d and dynamic capacitance model $------>$ synthesized Verilog-A module code.




```
`include "disciplines.vams"
`include "constants.vams"
module EDDdiode2(Panode, Pcathode);
inout Panode, Pcathode;
electrical Pcathode, n2, n1, Panode, n4, n3;
parameter real Area=1;          parameter real Is=1e-14;
parameter real Rs=0.1;          parameter real N=1;
parameter real Temp=26.85;      parameter real Vj=1.0;
parameter real Fc=0.5;          parameter real M=0.5;
parameter real Cj0=1e-12;       parameter real Tt=1e-12;
real RMAX, Vt, Con1, Con2, F1, F2, F3;
analog begin
@(initial_model)
begin
RMAX=1e15;      Vt=(`P_K*(Temp+273.15))/`P_Q;
Con1=5*N*Vt;    Con2=Fc*Vj;
F1=(Vj/(1-M))*(1-exp((1-M)*ln(1-Fc)));    F2=exp((1+M)*ln(1-Fc));
F3=1-Fc*(1+M);
end
I(Pcathode,n2) <+ V(Pcathode,n2)/( RMAX );
I(n1,Panode) <+ V(n1,Panode)/( Rs );
I(n1,Panode) <+ white_noise( 4.0*`P_K*( 26.85 + 273.15) / ( Rs ), "thermal" );
I(n1,n2) <+ V(n1,n2) * 1e3; I(n3,n4) <+ I(n3,n4) *1e-3;   I(n3,n4) <+ V(n1,n2) * 1e6 * 1 ;
I(n2,Pcathode) <+ Area*Is*(limexp(V(n2,Pcathode)/(N*Vt))-1);
I(n2,Pcathode) <+ ddt( (V(n2,Pcathode)<=Con2) ? Tt*V(n4,n3)+Area*(Cj0*Vj/(1-M))*
                  (1-exp((1-M)*ln(1-V(n2,Pcathode)/Vj)))
                : Tt*V(n4,n3)+Area*Cj0*(F1+(1/F2)*(F3*
                  V(n2,Pcathode)-Fc*Vj)+(M/(2*Vj))*
                  V(n2,Pcathode)*V(n2,Pcathode)-Fc*Fc*Vj*Vj))) );
end
endmodule
```

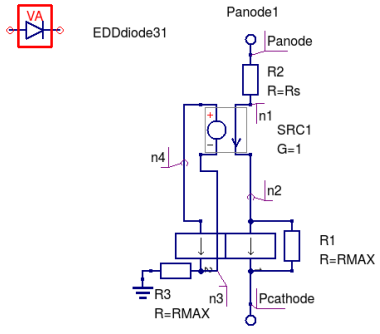
Introduction to the Qucs GPL Verilog-A Module Synthesizer: Part V

Synthesis of a SPICE like semiconductor diode model: simulated static and dynamic characteristics.



Introduction to the Qucs GPL Verilog-A Module Synthesizer: Part VI

Verilog-A synthesis of a SPICE like semiconductor diode model: temperature effects




```
Equation
Eqn1
RMAX=1e12
T1=Tnom+273.15
T2=Temp+273.15
Con1=5*N*Vt
Con2=Fc*Vj
Vt=kB*T2/q
F1=(Vj/(1-M))*(1-exp( ( 1-M)*ln(1-Fc)))
F2=exp( (1+M)*ln(1-Fc))
F3=1-Fc*(1+M)
A=7.02e-4
B=1108
Eg_T1=Eg-A*T1*T1/(B+T1)
Eg_T2=Eg-A*T2*T2/(B+T2)
Vj_T2=(T2/T1)*Vj-(2*Vt)*ln(exp(1.5*ln(T2/T1)))-((T2/T1)*Eg_T1-Eg_T2)
Cj0_T2=Cj0*(1+M*(400e-6*(T2/T1)-(Vj_T2-Vj)/Vj))
Is_T2=Is*exp((XTI/N)*ln(T2/T1))*limexp((-Eg_T1/Vt)*(1-T2/T1))

D1
I1=(V1 > -Con1) ?  Area*Is_T2*(limexp(V1/(N*Vt))-1) : -Is_T2
Q1= (V1 <= Con2) ? Tt*V2+Area*(Cj0_T2*Vj_T2/(1-M))*(1-exp( (1-M)*ln( 1-V1/Vj_T2))) :  Tt*V2+Area*Cj0_T2*(F1+(1/F2)*(F3*(V1-Fc*Vj_T2)+(M/(2*Vj_T2))*(V1*V1-Fc*Fc*Vj_T2*Vj_T2)))
I2=0
Q2=0
```

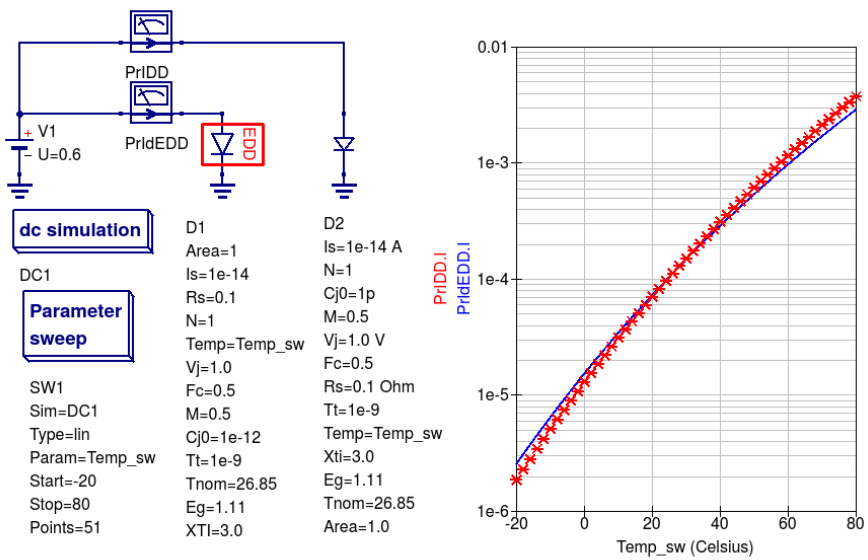
Qucs EDD diode model with temperature effects

Synthesized Verilog-A code

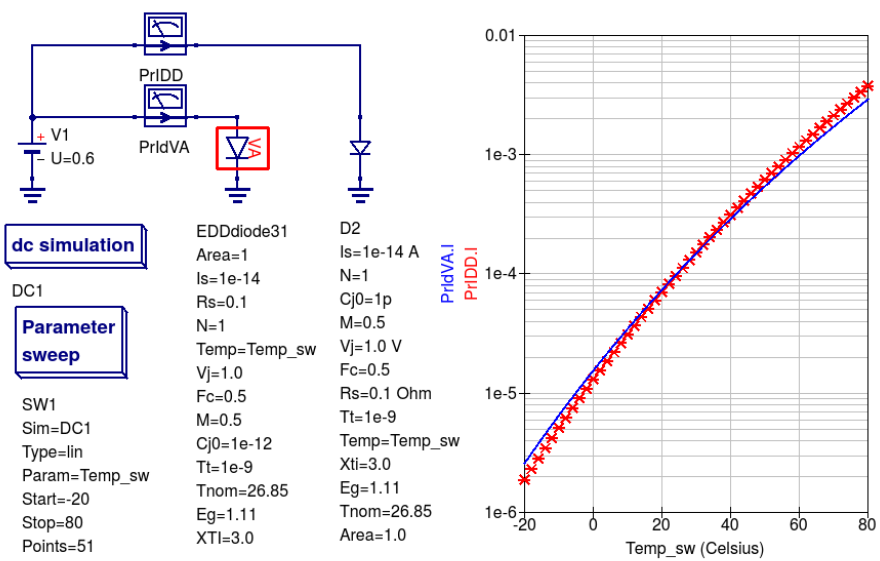
```
`include "disciplines.vams"
`include "constants.vams"
module EDDdiode3(Pcathode, Panode);
inout Pcathode, Panode;
electrical Pcathode, n2, n1, Panode, n4, n3;
parameter real Area=1;   parameter real Is=1e-14;  parameter real Rs=0.1;   parameter real N=1;
parameter real Temp=26.85; parameter real Vj=1.0;  parameter real Fc=0.5;  parameter real M=0.5;
parameter real Cj0=1e-12;    parameter real Tt=1e-9; parameter real Tnom=26.85; parameter real Eg=1.11;
parameter real XTI=3.0;
real RMAX, T1, T2, Con1, Con2, Vt, F1, F2, F3, A, B, EgT1, EgT2, VjT2, Cj0T2, IsT2;
analog begin
@(initial_model)
begin
RMAX=1e12;  T1=Tnom+273.15;  T2=Temp+273.15;  Con1=5*N*Vt;  Con2=Fc*Vj; Vt=`P_K*T2/`P_Q;
F1=(Vj/(1-M))*(1-exp((1-M)*ln(1-Fc))); F2=exp((1+M)*ln(1-Fc)); F3=1-Fc*(1+M); A=7.02e-4; B=1108;
EgT1=Eg-A*T1*T1/(B+T1); EgT2=Eg-A*T2*T2/(B+T2);
VjT2=(T2/T1)*Vj-(2*Vt)*ln(exp(1.5*ln(T2/T1)))-((T2/T1)*EgT1-EgT2);
Cj0T2=Cj0*(1+M*(400e-6*(T2/T1)-(VjT2-Vj)/Vj));
IsT2=Is*exp((XTI/N)*ln(T2/T1))*limexp((-EgT1/Vt)*(1-T2/T1));
end
I(Pcathode,n2) <+ V(Pcathode,n2)/( RMAX );
I(Pcathode,n2) <+ white_noise( 4.0*`PK*( 26.85 + 273.15) / ( RMAX ), "thermal" );
I(n1,Panode) <+ V(n1,Panode)/( Rs );
I(n1,Panode) <+ white_noise( 4.0*`PK*( 26.85 + 273.15) / ( Rs ), "thermal" );
I(n1,n2)  <+  V(n1,n2) * 1e3; I(n3,n4)  <+  I(n3,n4) *1e-3; I(n3,n4)  <+  V(n1,n2) * 1e6 * 1 ;
I(n2,Pcathode) <+ (V(n2,Pcathode)>-Con1)?Area*IsT2*(limexp(V(n2,Pcathode)/(N*Vt))-1):-IsT2;
I(n2,Pcathode) <+ ddt( (V(n2,Pcathode)<=Con2)?Tt*V(n4,n3)+Area*(Cj0T2*VjT2/(1-M))*(1-exp((1-M)*
                       ln(1-V(n2,Pcathode)/VjT2))):
                       Tt*V(n4,n3)+Area*Cj0T2*(F1+(1/F2)*(F3*(V(n2,Pcathode)-Fc*VjT2)+(M/(2*VjT2))*
                       (V(n2,Pcathode)*V(n2,Pcathode)-Fc*Fc*VjT2*VjT2))) );
I(n3) <+ (-V(n3))/( RMAX );
I(n3) <+ white_noise( 4.0*`PK*( 26.85 + 273.15) / ( RMAX ), "thermal" );
end
endmodule
```

Introduction to the Qucs GPL Verilog-A Module Synthesizer: Part VII

Verilog-A synthesis of a SPICE like semiconductor diode model: simulated $I_d - V_d$ temperature effects.



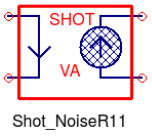
Simulation data for
Qucs EDD model and built-in diode model



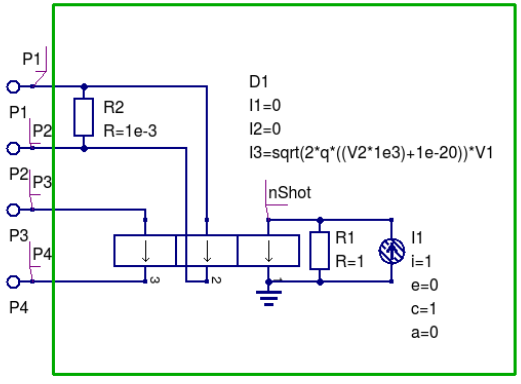
Simulation data for
Verilog-A model and built-in diode model

Introduction to the Qucs GPL Verilog-A Module Synthesizer: Part VIII

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



Shot_NoiseR11

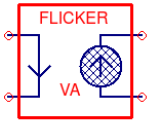


Noise model symbols

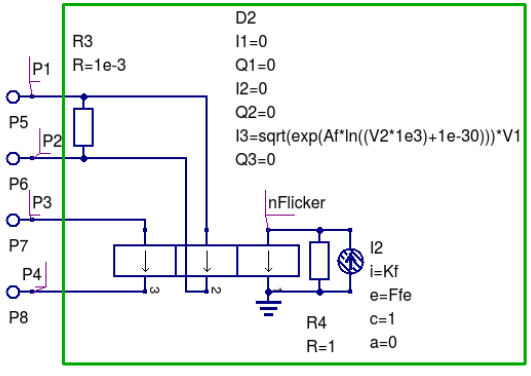
Compact modelling TEMPLATE

```
`include "disciplines.vams"
`include "constants.vams"
module Shot_NoiseR1(P1, P2, P3, P4);
inout P1, P2, P3, P4;
electrical nShot, P2, P1, P3, P4;
analog begin
@(initial_model)
begin
end
I(nShot) <+ (-V(nShot))/( 1 );
I(nShot) <+ white_noise(1,"shot" );
I(P2,P1) <+ V(P2,P1)/( 1e-3 );
I(P3,P4) <+ sqrt(2*`P_Q*((V(P1,P2)*1e3)+1e-20))*V(nShot);
end
endmodule
```

Synthesized Verilog-A module code



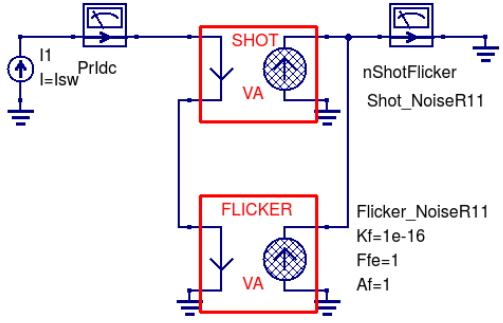
Flicker_NoiseR11
Kf=1e-16
Ffe=1
Af=1




```
`include "disciplines.vams"
`include "constants.vams"
module Flicker_NoiseR1(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
end
I(P2,P1) <+ V(P2,P1)/( 1e-3 );
I(nFlicker) <+ flicker_noise(Kf, Ffe, "flicker" );
I(nFlicker) <+ (-V(nFlicker))/( 1 );
(P3,P4) <+ sqrt(exp(Af*ln((V(P1,P2)*1e3)+1e-30)))*V(nFlicker);
end
endmodule
```

Introduction to the Qucs GPL Verilog-A Module Synthesizer: Part IX

Verilog-A synthesis of semiconductor device shot and flicker noise: small signal AC domain simulation data.



ac simulation

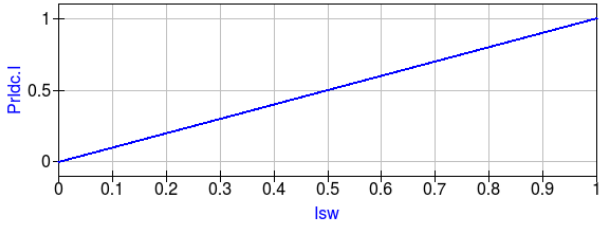
AC1
Type=log
Start=0.01
Stop=100k
Points=141
Noise=yes

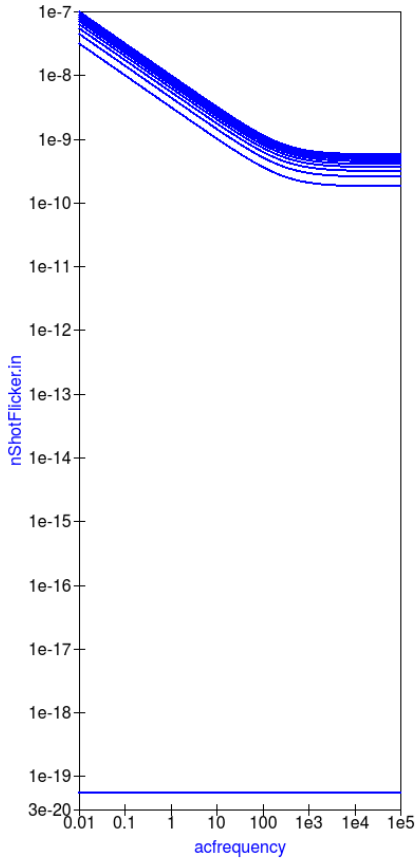
Parameter sweep

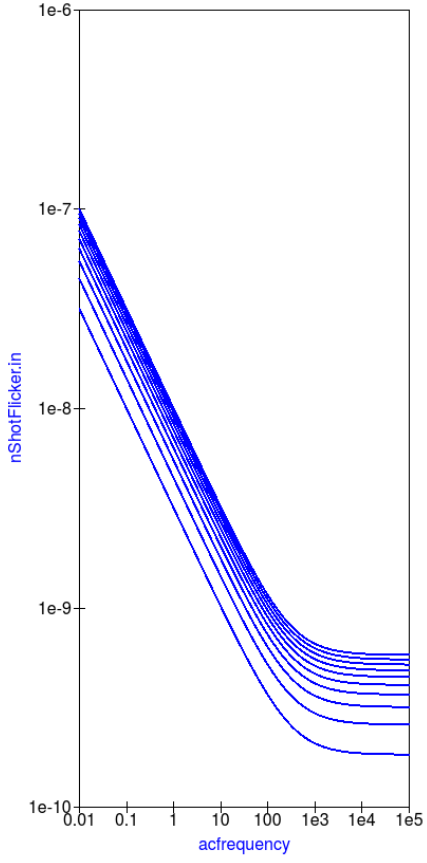
SW1
Sim=AC1
Type=lin
Param=Isw
Start=0
Stop=1
Points=11

dc simulation

DC1



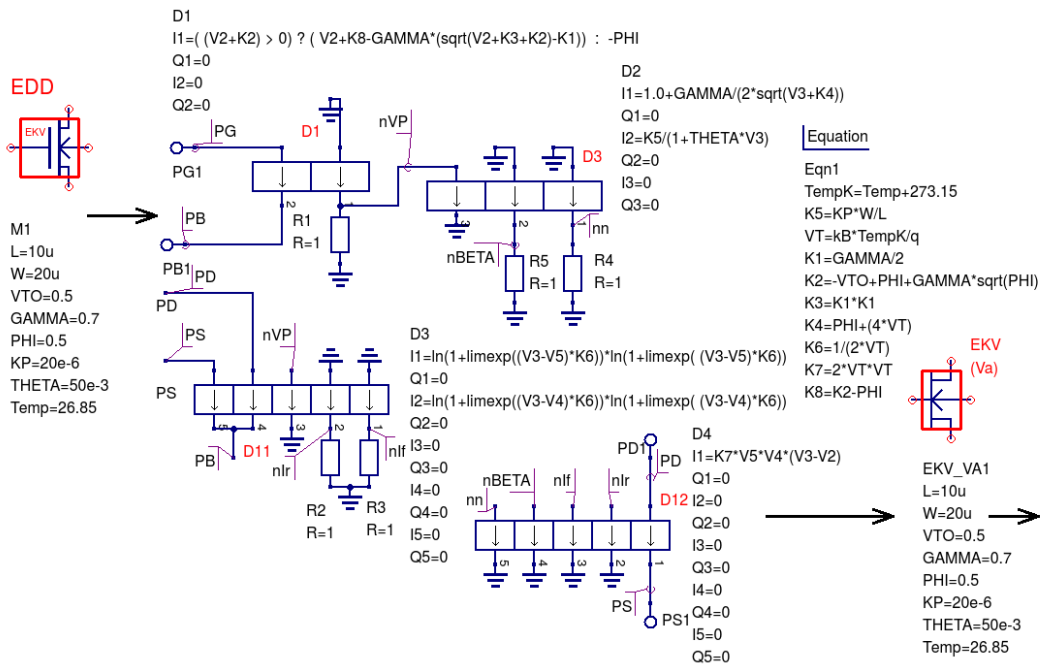




Introduction to the Qucs GPL Verilog-A Module Synthesizer: Part X

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.



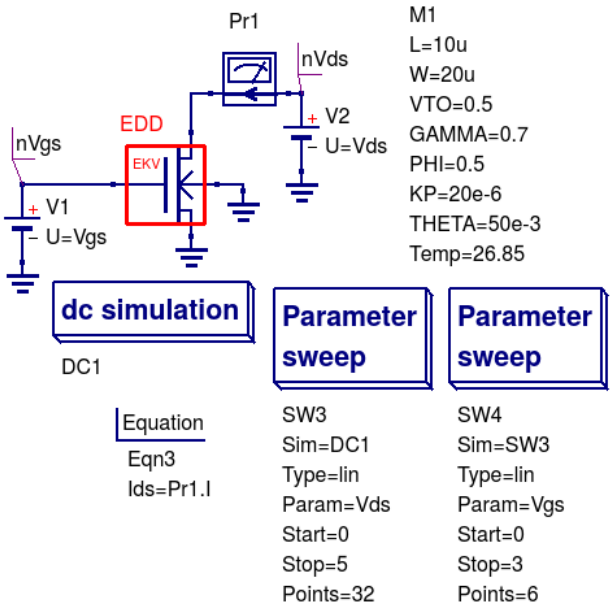
Qucs EDD EKV2p6 Ids=f(Vd, Vg, Vs, Vb) model

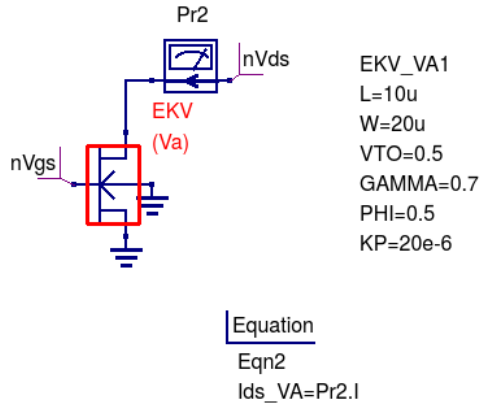
```
`include "disciplines.vams"
`include "constants.vams"
module EKV_VA(PB, PG, PD, PS);
inout PB, PG, PD, PS;
electrical PD, PS, nlr, nlf, nBETA, nn, nVP, PG, PB;
parameter real L=10u;    parameter real W=20u;    parameter real VTO=0.5;
parameter real GAMMA=0.7;    parameter real PHI=0.5;    parameter real KP=20e-6;
parameter real THETA=50e-3;    parameter real Temp=26.85;
real TempK, K5, VT, K1, K2, K3, K4, K6, K7, K8;
analog begin
@(initial_model)
begin
TempK=Temp+273.15;    K5=KP*W/L;    VT=`P_K*TempK/`P_Q;
K1=GAMMA/2;    K2=-VTO+PHI+GAMMA*sqrt(PHI);
K3=K1*K1;    K4=PHI+(4*VT);
K6=1/(2*VT);    K7=2*VT*VT;    K8=K2-PHI;
end
I(PD,PS) <+ K7*V(nn)*V(nBETA)*(V(nlf)-V(nlr));
I(nVP) <+ -(((V(PG,PB)+K2)>0)?(V(PG,PB)+K8-GAMMA*(sqrt(V(PG,PB)+K3+K2)-K1)):-PHI);
I(nVP) <+ V(nVP)/( 1 );
I(nBETA) <+ -((-V(nBETA))/( 1 ));
I(nn) <+ -(1.0+GAMMA/(2*sqrt(V(nVP)+K4)));
I(nBETA) <+ -(K5/(1+THETA*(-V(nVP))));
I(nn) <+ -((-V(nn))/( 1 ));
I(nlr) <+ -((-V(nlr))/( 1 ));
I(nlf) <+ -((-V(nlf))/( 1 ));
I(nlf) <+ -(ln(1+limexp((V(nVP)-V(PS,PB))*K6))*ln(1+limexp((V(nVP)-V(PS,PB))*K6)));
I(nlr) <+ -(ln(1+limexp((V(nVP)-V(PD,PB))*K6))*ln(1+limexp((V(nVP)-V(PD,PB))*K6)));
end
endmodule
```

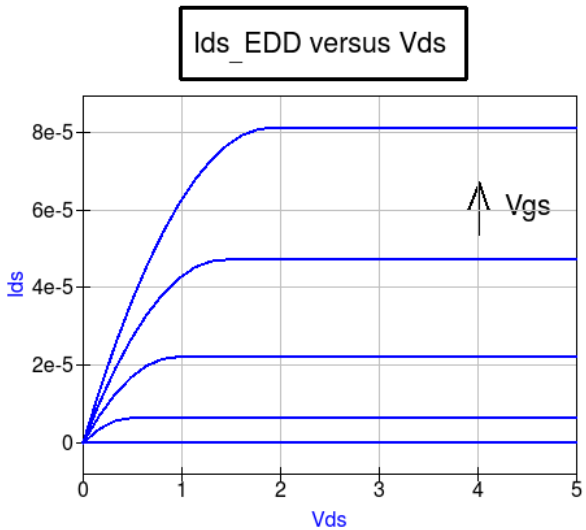
Synthesized EKV2p6 Ids=f(Vd, Vg, Vs, Vb) Verilog-A code

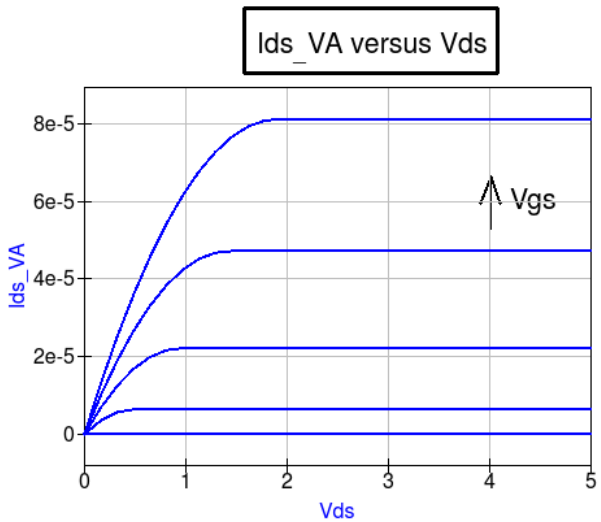
Introduction to the Qucs GPL Verilog-A Module Synthesizer: Part XI

Verilog-A synthesis of multi-EDD models: EKV2p6 nMOS
$I_{ds} = f(V_d, V_g, V_s, V_b)$ swept DC simulation data.



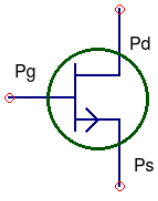




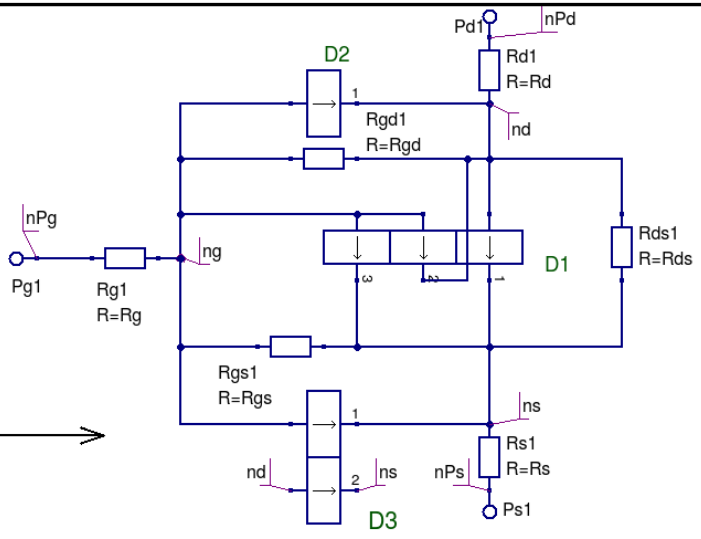


Qucs modelling of the "Efficient Power Corporation (EPC)" GaN EPC2001 Power Transistor: Part I EDD Subcircuit Compact Device Model

Qucs-S EDD subcircuit model of a EPC2001 enhancement GaN power transistor -See Application Note: An005 Circuit Simulation using EPC Device Models, Efficient Power Conversion Corporation,Copyright 2011, www.epc-co.com.



GaN_EPC1
Temp=25
A1=41.7998
alTc=5.486028e-3
k2=2.259866
k3=1.2e-1
aWg=1077
rpara=4.463059e-3
x0_0=0.75
x0_1=1.10
ax0Tc=0.75e-4
dgs1=4.4e-7
dgs2=2.6e-13
dgs3=0.8
dgs4=0.23
ags1=8.6952e-10
agd1=1.4182e-11
asd1=3.3621e-10
asd2=6.308e-10
asd3=1.2803e1
asd4=2.269
asd5=2.518e-10
asd6=-4.0599e1
asd7=2.0638e1
agd2=2.1475e-10
agd3=-3.803
agd4=5.9551
ags2=5.3168e-10
ags3=1.997
ags4=2.8377e-1
ags5=-1.4751e-10
ags6=-7.5163
ags7=7.121



Equation

Eqn1
EQ1=A1*(1-alTc*(Temp-25))
Rgs=1e11/aWg
Rgd=1e11/aWg
Rds=1e11/aWg
Rd=0.75*rpara
Rs=0.25*rpara
Rg=0.6*1077/aWg
EQ2=1.0+ax0Tc*(Temp-25)*(Temp-25)
EQ3=0.5*aWg

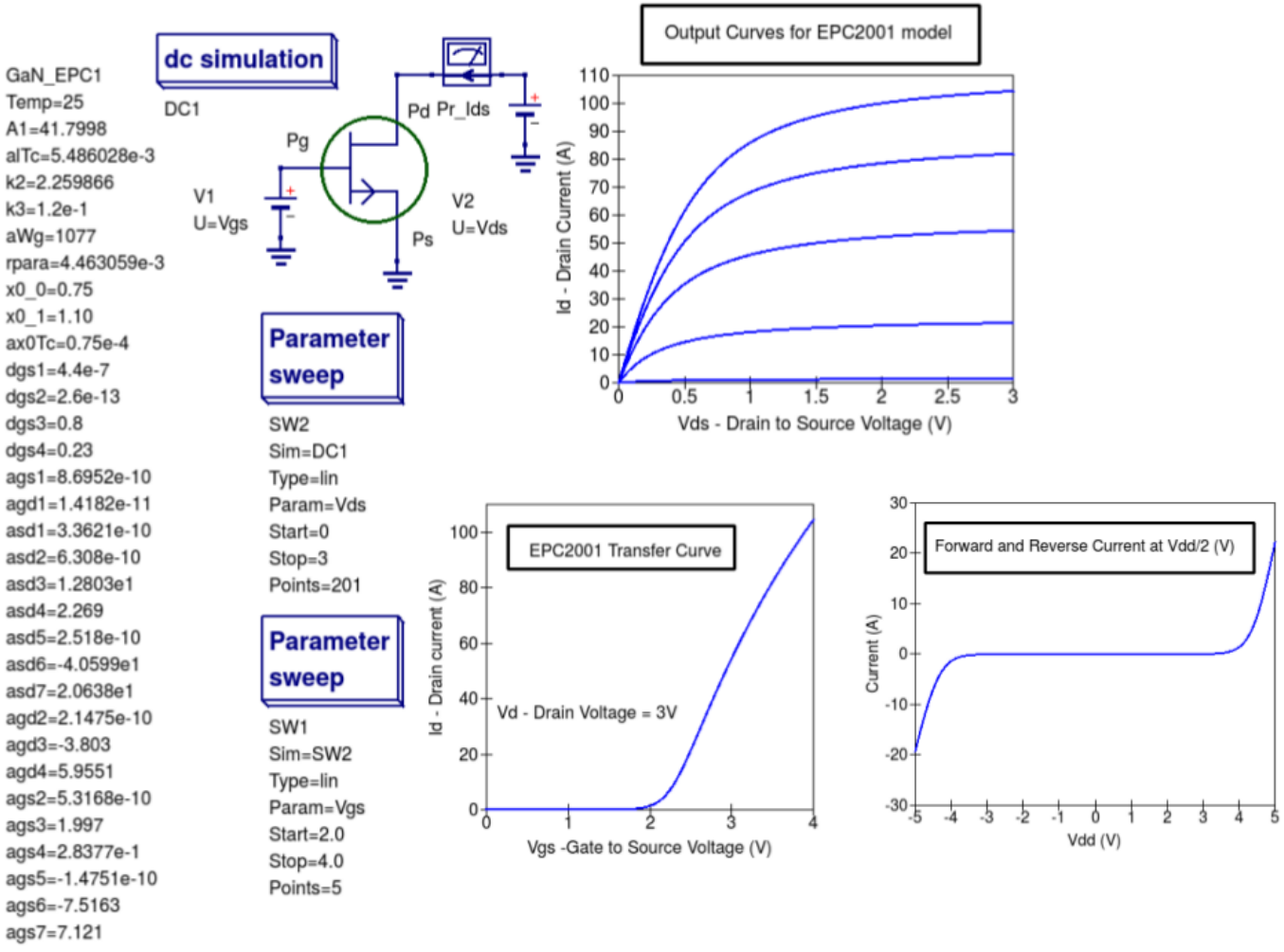
D1
I1=(V1 > 0) ? EQ1*ln(1.0+exp((V3-k2)/k3))*V1/(1.0+max((x0_0+x0_1*V3/EQ2), 0.5)*V1) : -EQ1*ln(1.0+exp((V2-k2)/k3))*(-V1)/(1.0+max((x0_0+x0_1*V2/EQ2), 0.5)*(-V1))
Q1=asd1*V1 + (asd2*asd4*ln(1+exp((V1-asd3) / asd4)) + asd5*asd7*ln(1 + exp((V1-asd6)/asd7)))
I2=0
I3=0

D2
I1=(V1 > 10) ? (EQ3/1077*(dgs1*(exp(10.0/dgs3)-1)+dgs2*(exp(10/dgs4)-1))) : (EQ3/1077*(dgs1*(exp(V1/dgs3)-1)+dgs2*(exp(V1/dgs4)-1)))
Q1=agd1*V1 + (0.5*ags2*ags4*ln(1+exp((V1-ags3) / ags4)) + agd2*agd4*ln(1 + exp((V1-agd3)/agd4)))

D3
I1=(V1 > 10) ? (EQ3/1077*(dgs1*(exp(10.0/dgs3)-1)+dgs2*(exp(10/dgs4)-1))) : (EQ3/1077*(dgs1*(exp(V1/dgs3)-1)+dgs2*(exp(V1/dgs4)-1)))
Q1=ags1*V1 + (0.5*ags2*ags4*ln(1+exp((V1-ags3) / ags4)) + ags5*ags7*ln(1 + exp((V2-ags6)/ags7)))

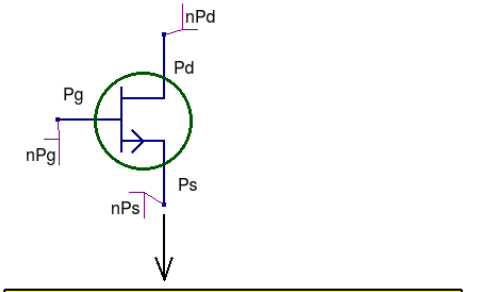
Ported from the EPC mathematical model common to LTSPICE, PSPICE, TSPICE and Spectra netlist models.

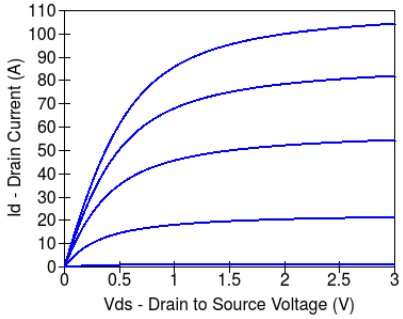
Qucs Modeling of the "Efficient Power Corporation (EPC)" GaN EPC2001 Power Transistor: Part II DC Test Bench and Typical Simulation Curves



Qucs Modeling of the "Efficient Power Corporation (EPC)" GaN EPC2001 Power Transistor: Part III Synthesis of Verilog-A code for a EPC2001 Qucs EDD subcircuit

EPC20011
Temp=25
A1=41.7998
alTc=5.486028e-3
k2=2.259866
k3=1.2e-1
aWg=1077
rpara=4.463059e-3
x0_0=0.75
x0_1=1.10
ax0Tc=0.75e-4
dgs1=4.4e-7
dgs2=2.6e-13
dgs3=0.8
dgs4=0.23
ags1=8.6952e-10
agd1=1.4182e-11
asd1=3.3621e-10
asd2=6.308e-10
asd3=1.2803e1
asd4=2.269
asd5=2.518e-10
asd6=-4.0599e1
asd7=2.0638e1
agd2=2.1475e-10
agd3=-3.803
agd4=5.9551
ags2=5.3168e-10
ags3=1.997
ags4=2.8377e-1
ags5=-1.4751e-10
ags6=-7.5163
ags7=7.121

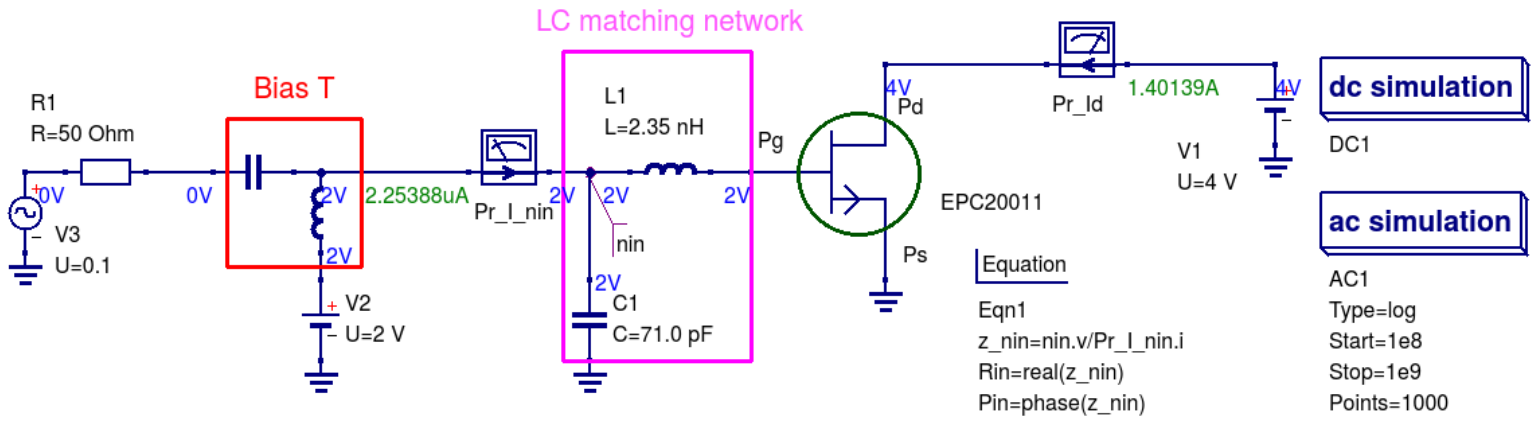


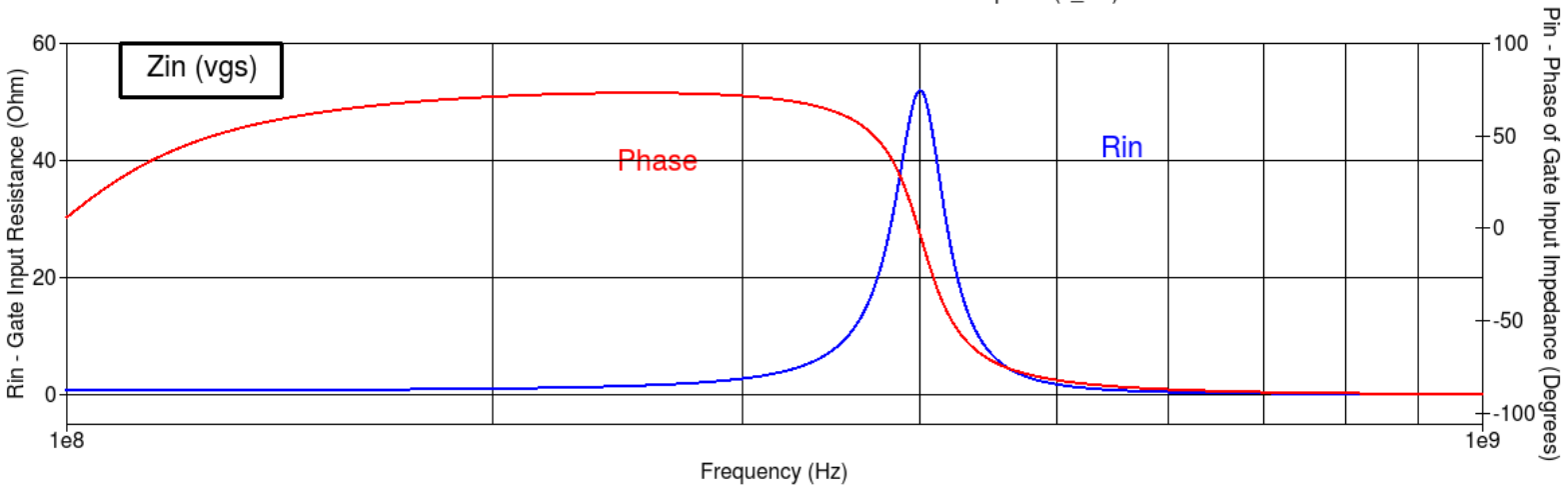


EPC2001 Synthesized Verilog-A module code

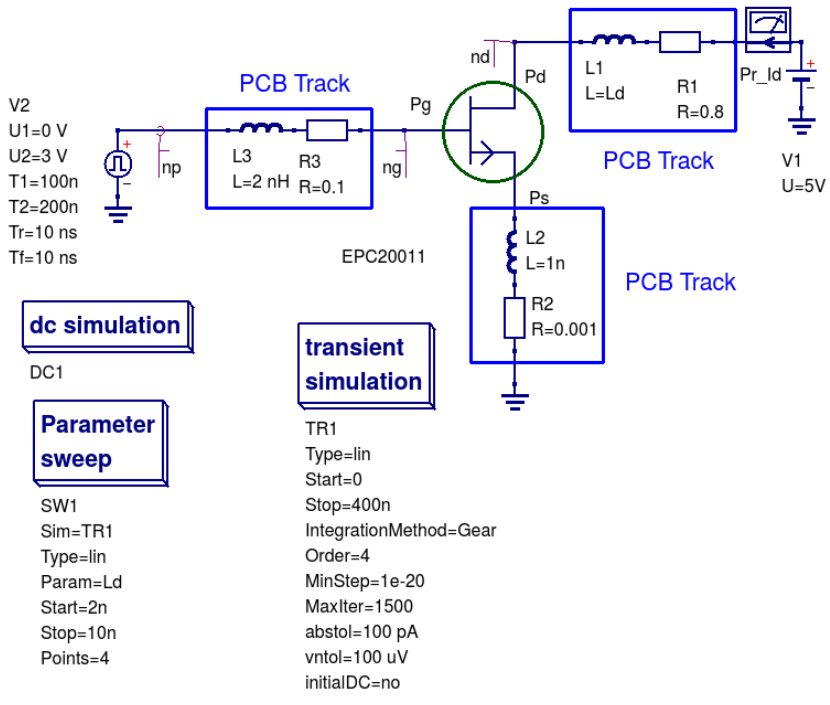
```
`include "disciplines.vams"
`include "constants.vams"
module EPC2001(nPs, nPd, nPg);
inout nPs, nPd, nPg;  electrical ng, ns, nd, nPs, nPd, nPg;
parameter real Temp=25;          parameter real A1=41.7998;     parameter real alTc=5.486028e-3;
parameter real k2=2.259866;      parameter real k3=1.2e-1;      parameter real aWg=1077;
parameter real rpara=4.463059e-3; parameter real x0_0=0.75;     parameter real x0_1=1.10;
parameter real ax0Tc=0.75e-4;    parameter real dgs1=4.4e-7;    parameter real dgs2=2.6e-13;
parameter real dgs3=0.8;         parameter real dgs4=0.23;      parameter real ags1=8.6952e-10;
parameter real agd1=1.4182e-11;  parameter real asd1=3.3621e-10;
parameter real asd2=6.308e-10;   parameter real asd3=1.2803e1;  parameter real asd4=2.269;
parameter real asd5=2.518e-10;   parameter real asd6=-4.0599e1; parameter real asd7=2.0638e1;
parameter real agd2=2.1475e-10;  parameter real agd3=-3.803;    parameter real agd4=5.9551;
parameter real ags2=5.3168e-10;  parameter real ags3=1.997;     parameter real ags4=2.8377e-1;
parameter real ags5=-1.4751e-10; parameter real ags6=-7.5163;   parameter real ags7=7.121;
real EQ1, Rgs, Rgd, Rds, Rd, Rs, Rg, EQ2, EQ3;
analog begin
@(initial_model)  begin
EQ1=A1*(1-alTc*(Temp-25)); Rgs=1e11/aWg; Rgd=1e11/aWg; Rds=1e11/aWg; Rd=0.75*rpara;
Rs=0.25*rpara; Rg=0.6*1077/aWg; EQ2=1.0+ax0Tc*(Temp-25)*(Temp-25); EQ3=0.5*aWg;
end
I(ng,ns) <+ V(ng,ns)/( Rgs );    I(ng,ns) <+ white_noise( 4.0*`P_K*( 26.85 + 273.15) / ( Rgs ), "thermal" );
I(ng,nd) <+ V(ng,nd)/( Rgd );    I(ng,nd) <+ white_noise( 4.0*`P_K*( 26.85 + 273.15) / ( Rgd ), "thermal" );
I(ns,nd) <+ V(ns,nd)/( Rds );    I(ns,nd) <+ white_noise( 4.0*`P_K*( 26.85 + 273.15) / ( Rds ), "thermal" );
I(nd,ns) <+ (V(nd,ns)>0) ?  EQ1*ln(1.0+exp((V(ng,ns)-k2)/k3))*V(nd,ns)/(1.0+max((x0_0+x0_1*V(ng,ns)/EQ2),0.5)*V(nd,ns))
                         :  -EQ1*ln(1.0+exp((V(ng,nd)-k2)/k3))*(-V(nd,ns))/(1.0+max((x0_0+x0_1*V(ng,nd)/EQ2),0.5)*(-V(nd,ns)));
I(nd,ns) <+ ddt( asd1*V(nd,ns)+(asd2*asd4*ln(1+exp((V(nd,ns)-asd3)/asd4))+asd5*asd7*ln(1+exp((V(nd,ns)-asd6)/asd7))) );
I(ng,nd) <+ (V(ng,nd)>10) ? (EQ3/1077*(dgs1*(exp(10.0/dgs3)-1)+dgs2*(exp(10/dgs4)-1)))
                          :  (EQ3/1077*(dgs1*(exp(V(ng,nd)/dgs3)-1)+dgs2*(exp(V(ng,nd)/dgs4)-1)));
I(ng,nd) <+ ddt( agd1*V(ng,nd)+(0.5*ags2*ags4*ln(1+exp((V(ng,nd)-ags3)/ags4))+agd2*agd4*ln(1+exp((V(ng,nd)-agd3)/agd4))) );
I(nPs,ns) <+ V(nPs,ns)/( Rs );
I(nPs,ns) <+ white_noise( 4.0*`P_K*( 26.85 + 273.15) / ( Rs ), "thermal" );   I(nd,nPd) <+ V(nd,nPd)/( Rd );
I(nd,nPd) <+ white_noise( 4.0*`P_K*( 26.85 + 273.15) / ( Rd ), "thermal" );   I(nPg,ng) <+ V(nPg,ng)/( Rg );
I(nPg,ng) <+ white_noise( 4.0*`P_K*( 26.85 + 273.15) / ( Rg ), "thermal" );
I(ng,ns) <+ (V(ng,ns)>10) ? (EQ3/1077*(dgs1*(exp(10.0/dgs3)-1)+dgs2*(exp(10/dgs4)-1)))
                          :  (EQ3/1077*(dgs1*(exp(V(ng,ns)/dgs3)-1)+dgs2*(exp(V(ng,ns)/dgs4)-1)));
I(ng,ns) <+ ddt( ags1*V(ng,ns)+(0.5*ags2*ags4*ln(1+exp((V(ng,ns)-ags3)/ags4))+ags5*ags7*ln(1+exp((V(nd,ns)-ags6)/ags7))) );
end
endmodule
```

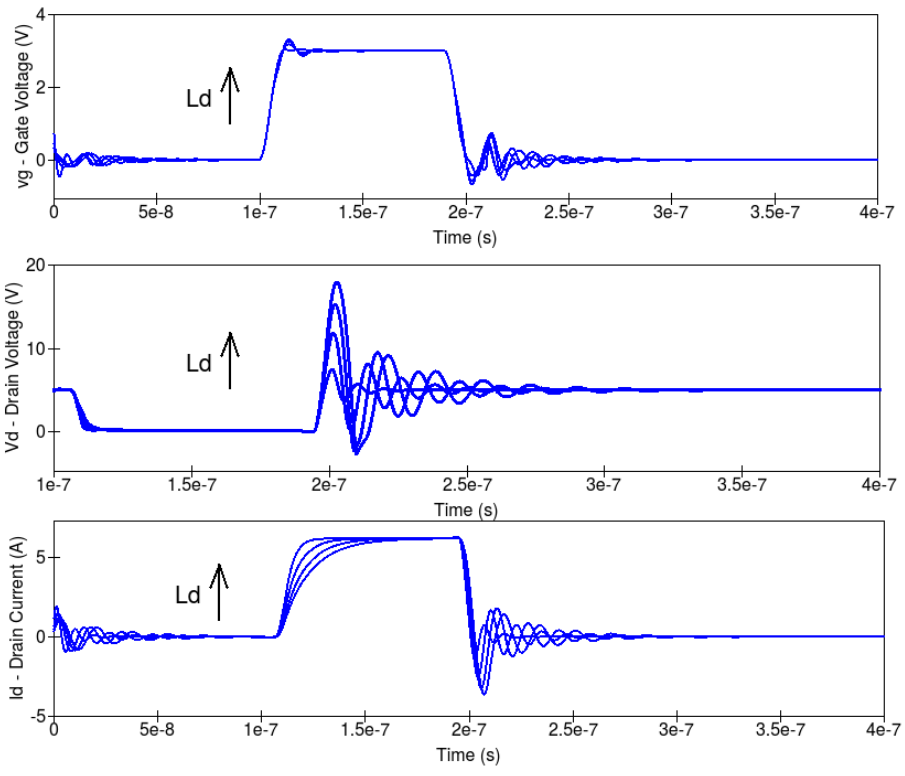
Qucs Modeling of the "Efficient Power Corporation (EPC)" GaN EPC2001 Power Transistor: Part IV AC Gate Matching Network Test Bench and Typical Simulation Results





Qucs Modelling of the "Efficient Power Corporation (EPC)" GaN EPC2001 Power Transistor: Part V Switching Response Test Bench and Typical Simulation Results





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 - Part I Introduction

- 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.

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 - Part II Model Parameter Statement Error Work-Arounds

- ADMS parameter statements DO NOT ALLOW reference to previously defined model parameters.

X ADMS synthesis/compile error

```
parameter real  vxord      = 1.30e7    from [0:inf);        // Source injection velocity [cm/s]
parameter real  VtOrd      = -2.0;          // Threshold voltage of drain access transistor[V]
parameter real  Cgrd       = 5.0e-7    from (0:inf);        // Drain access areal capacitance [F/cm2]
parameter real  delta1rd   = 1.3       from [0:inf);        // DIBL for drain access transitor
parameter real  delta2rd   = 0.30      from [0:inf);        // DIBL for drain access transitor
parameter real  Vdibsat    = 2.0       from [0:inf);        // DIBL for drain access transitor
parameter real  Srd        = 0.35      from [0:inf);        // Subthreshold slope for drain access transitor [V/Dec]
parameter real  zeta       = 0.0       from [0:inf);        // Self heating parameter (scalable)
parameter real  betard     = 1.3       from [0:inf);        // Linear to saturation transition parameter
parameter real  vthetard   = 0.05      from [0:inf);        // Scattering: velocity reduction parameter with Vg
parameter real  ndrd       = 0*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  VtOrs      = VtOrd;                 // Threshold voltage of drain access transistor[V]
parameter real  Cgrs       = Cgrd      from (0:inf);        // Drain access areal capacitance [F/cm2]
parameter real  delta1rs   = delta1rd  from [0:inf);        // DIBL for drain access transitor
parameter real  delta2rs   = delta2rd  from [0:inf);        // DIBL for drain access transitor
parameter real  Srs        = Srd       from [0:inf);        // Subthreshold slope for drain access transitor [V/Dec]
parameter real  vthetars   = 0.05      from [0:inf);        // Scattering: velocity reduction parameter with Vg
parameter real  ndrs       = ndrd      from [0:inf);        // Punchthrough factor affects slope change in subthreshold
parameter real  betars     = betard    from [0:inf);        // Linear to saturation transition parameter
```

OK

```
parameter real  vxord      = 1.30e7    from [0:inf);
parameter real  VtOrd      = -2.0;
parameter real  Cgrd       = 5.0e-7    from (0:inf);
parameter real  delta1rd   = 1.3       from [0:inf);
parameter real  delta2rd   = 0.30      from [0:inf);
parameter real  Vdibsat    = 2.0       from [0:inf);
parameter real  Srd        = 0.35      from [0:inf);
parameter real  zeta       = 0.0       from [0:inf);
parameter real  betard     = 1.3       from [0:inf);
parameter real  vthetard   = 0.05      from [0:inf);
parameter real  ndrd       = 0.80      from [0:inf);

parameter real  VtOrs      = -2.0;
parameter real  Vxors      = 1.30e7;
parameter real  Cgrs       = 5.0e-7    from (0:inf);
parameter real  delta1rs   = 1.3       from [0:inf);
parameter real  delta2rs   = 0.30      from [0:inf);
parameter real  Srs        = 0.35      from [0:inf);
parameter real  vthetars   = 0.05      from [0:inf);
parameter real  ndrs       = 0.80      from [0:inf);
parameter real  betars     = 1.3       from [0:inf);
```

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 - Part III Removing $V(n) <+ statements$

- ADMS DOES NOT ALLOW voltage contributions of the form $V(n) <+ I(n) * R$, where R is a resistance in Ω,
- OR statements of the form $V(n) <+ 0.0$,
- Resistors, for example 0.001Ω, are used to short nodes (node collapsing), with $I(n) <+ V(N)/0.001$.

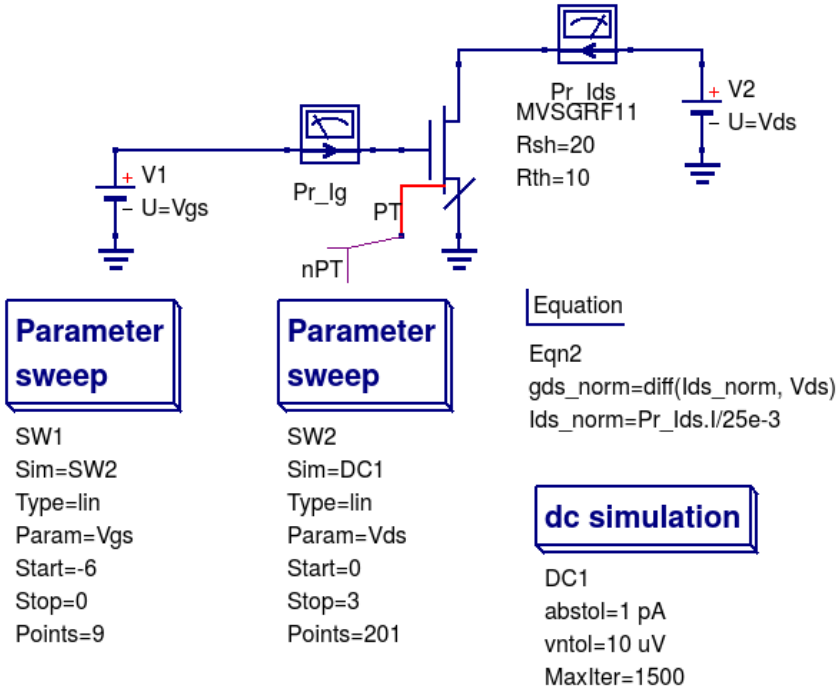
X ADMS synthesis/compile error

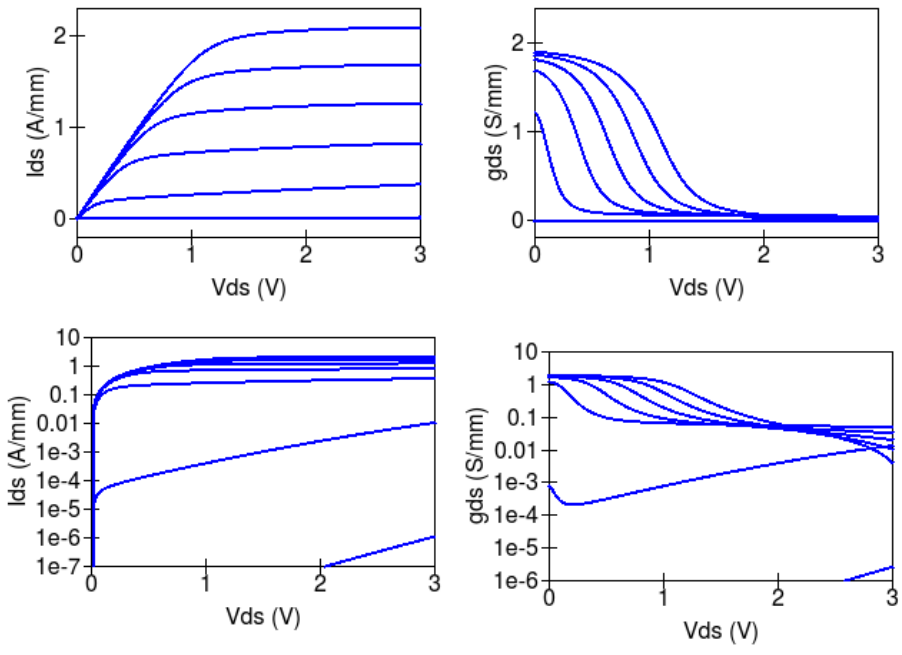
```
if (Rsh > 1e-3 && Ls > 0)
    I(si,src) <+ Idsrs;
else
   V(src,si) <+ 0;
//Source side contact resistance
if (Rc > 0) begin
  I(src,s)    <+ V(src,s) / ( Rc / Wg );
end else begin
  V(src,s)    <+ 0;
end
```

OK

```
if (Rsh > 1e-3 && Ls > 0)
    I(si,src) <+ Idsrs;
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 );
end else begin
  I(src,s)    <+  V(src,s)/1e-3;
end
```

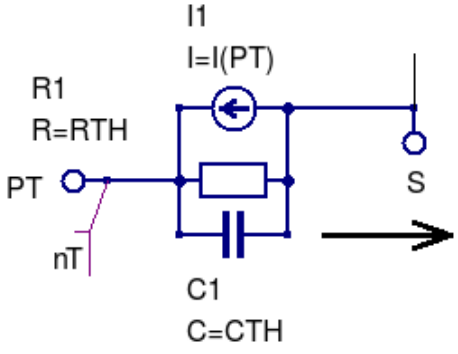
Qucs Verilog-A Modeling of the "MIT Virtual Source GaN-RF HEMT compact model 1.0.0": Problems Simulating with ADMS; Workarounds and Typical Simulation Data - Part IV DC Characteristics





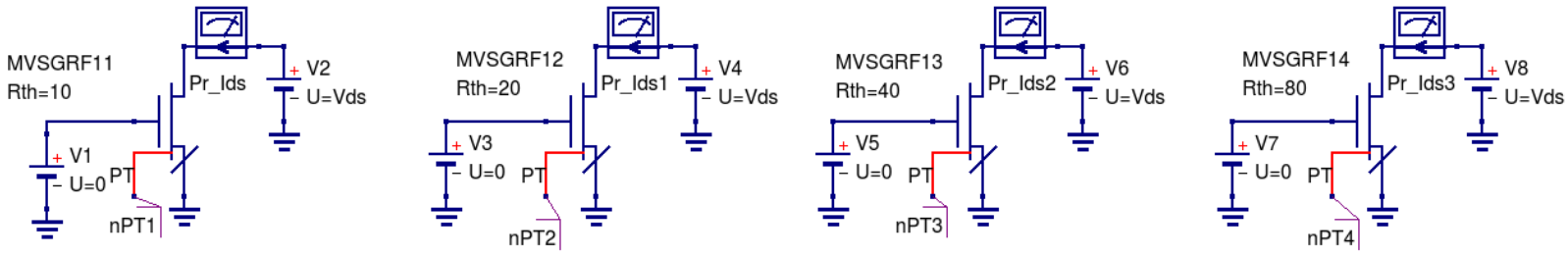
Qucs Verilog-A Modeling of the "MIT Virtual Source GaN-RF HEMT compact model 1.0.0": Problems Simulating with ADMS; Workarounds and Typical Simulation Data - Part V Simulating Thermal self-Heating Effects Induced by Internal Power Dissipation

- 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,
- $Tth = Rth \cdot Pd + Temp(Pd = 0)$, where Tth is the device temperature at power dissipation Pd (W).



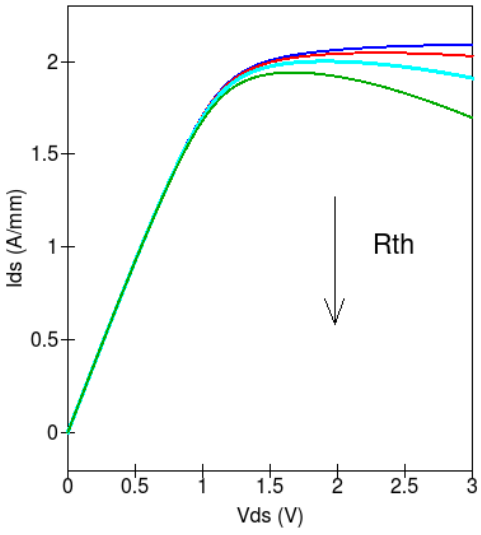

```
// Self-heating
    I(PT)  <+ ddt( Cth * V(PT) );
    I(PT)  <+ -( I(di,si) * V(di,si) + I(d,drc) * V(d,drc) + I(src,s) * V(src,s) + V(drc,di) * I(drc,di) + V(src,si) * I(src,si) );
    I(PT)  <+ V(PT)/Rth;
```

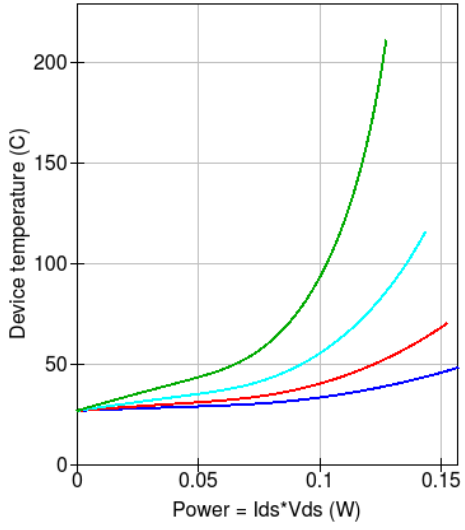
Qucs Verilog-A Modeling of the "MIT Virtual Source GaN-RF HEMT compact model 1.0.0": Problems Simulating with ADMS; Workarounds and Typical Simulation Data - Part VI Variation of Thermal Resistance Rth and its Effect on DC Characteristics



Equation

Eqn2
Ids_norm=Pr_Ids.I/25e-3
Ids_norm2=Pr_Ids1.I/25e-3
Ids_norm3=Pr_Ids2.I/25e-3
Ids_norm4=Pr_Ids3.I/25e-3
Temp1=PlotVs(nPT1.V+27, Pr_Ids.I*Vds)
Temp2=PlotVs(nPT2.V+27, Pr_Ids1.I*Vds)
Temp3=PlotVs(nPT3.V+27, Pr_Ids2.I*Vds)
Temp4=PlotVs(nPT4.V+27, Pr_Ids3.I*Vds)





Parameter sweep

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

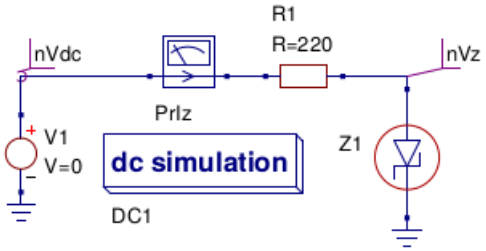
dc simulation

DC1
abstol=1 pA
vntol=10 uV
MaxIter=1500

Qucs-0.0.19-S-RC6 Modeling Tool Additions and New Features Currently Under Development: Moving Forward to the Next Generation of Qucs-S Circuit Simulation and compact Device Modeling Capabilities

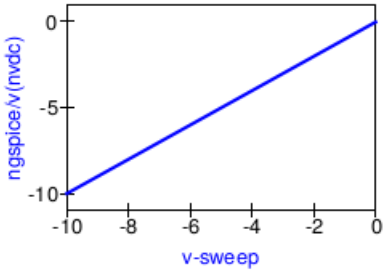
- Qucs-0.0.19-S-RC6 includes for the first time a "turn-key" XSPICE Code Modelling package for use with the Ngspice and SPICE OPUS circuit simulators,
- Qucs-0.0.19-S-RC6 is being extended to include a new Qucs/Octave integrated tool set for compact device model and circuit macromodel parameter extraction 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.

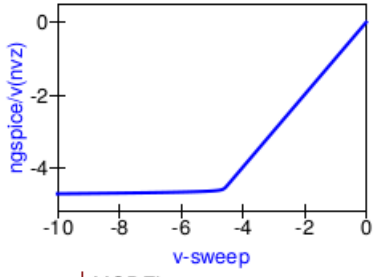
Qucs-0.0.19-S-RC6 XSPICE Code Modeling package: Part I XSPICE Code Model Subcircuits

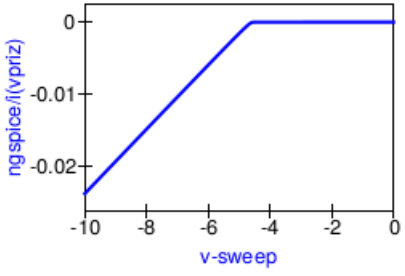


Parameter sweep

SW1
Sim=DC1
Type=lin
Param=V1
Start=0
Stop=-10
Points=201

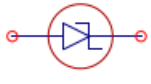


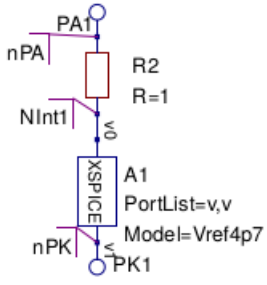




.MODEL

zener1
Line_1 =.MODEL Vref4p7 zener(V_breakdown=4.7 i_breakdown=20m
Line_2=+ r_breakdown=1
Line_3=+ i_rev=1e-6
Line_4=+ i_sat=1e-12)





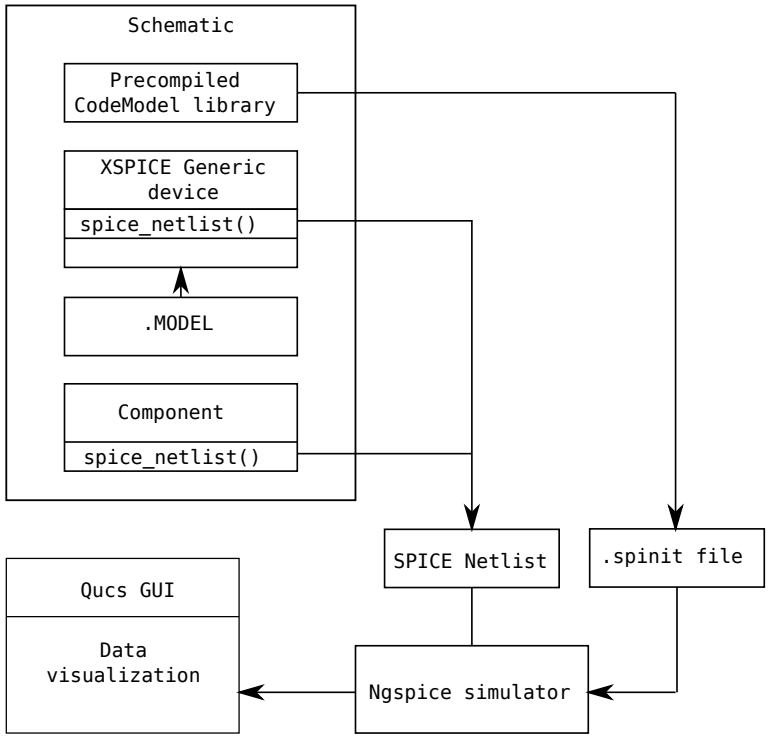

```
* Qucs 0.0.19  Zener4p7.sch
.SUBCKT Zener4p7 nPA nPK
A NInt1 nPK Vref4p7
R1 NInt1  nPA 1

.MODEL Vref4p7  zener(V_breakdown=4.7
+   i_breakdown=20m
+   r_breakdown=1
+   i_rev=1e-6
+   i_sat=1e-12 )
.ENDS
VPrIz nVdc _net0 DC 0 AC 0
R1 nVz  _net0 220
V1 nVdc  0 0
XZ1  nVz 0 Zener4p7
.control
set filetype=ascii
echo "" > spice4qucs.cir.noise
echo "" > spice4qucs.cir.pz
dc v1 0 -10 -0.0497512
write Test_zener4p7_dc.txt VPrIz#branch v(nVdc) v(nVz)
destroy all
reset
exit
.endc
.END
```

Qucs-0.0.19-S-RC6 XSPICE Code Modeling Package: Part II XSPICE CodeModel Support Subsystem

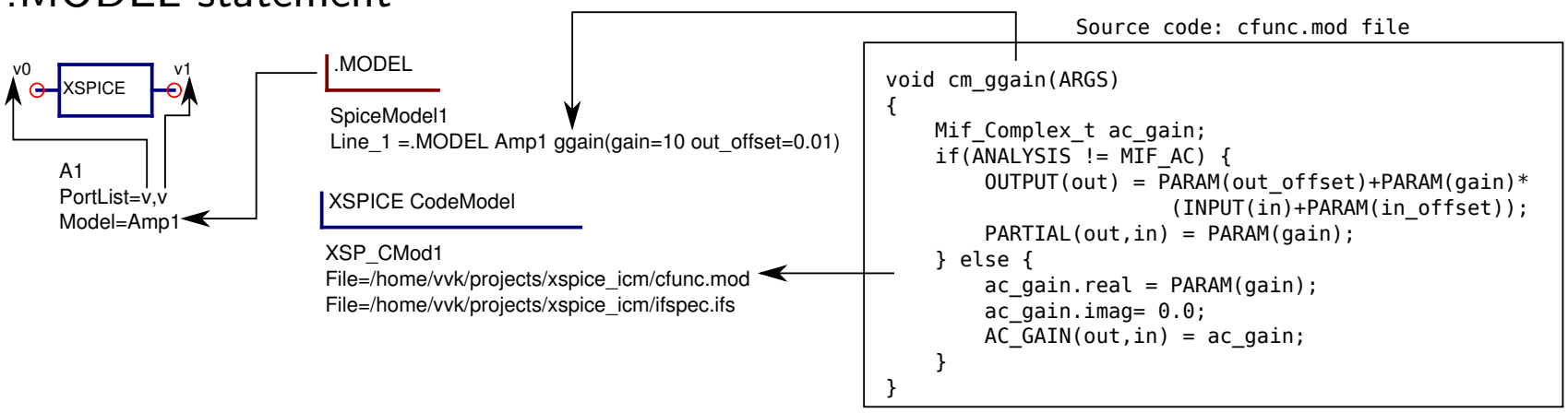
- The "XSPICE generic device" component is the foundation for
  - Precompiled XSPICE device (*.cm) library support, and
  - Dynamic XSPICE Code Models compilation system which allows **Code Model sources to be attached to a schematic and compiled automatically at simulation time**.

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

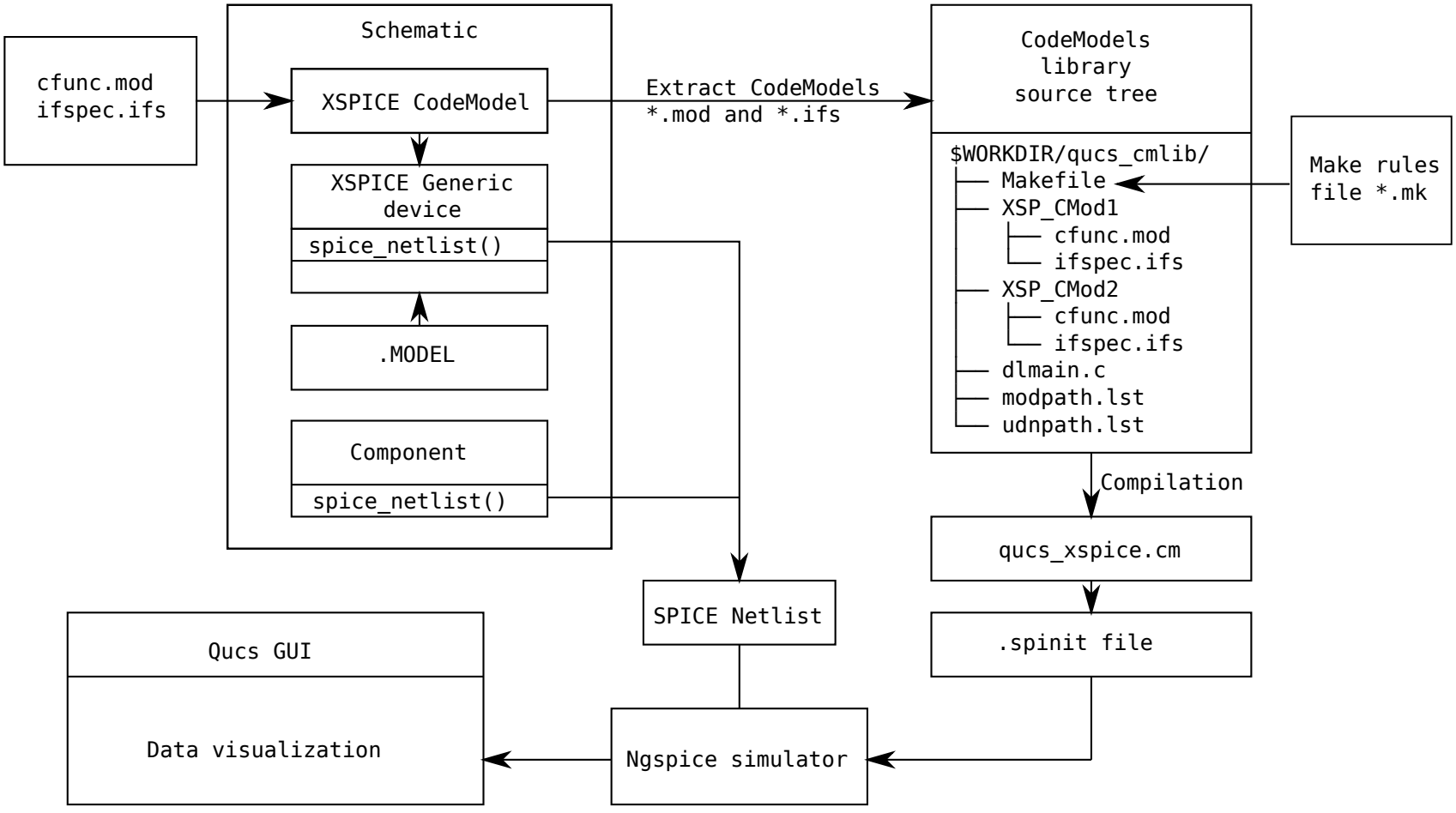


Qucs-0.0.19-S-RC6 XSPICE Code Modeling package: Part III "XSPICE Generic Device" Component

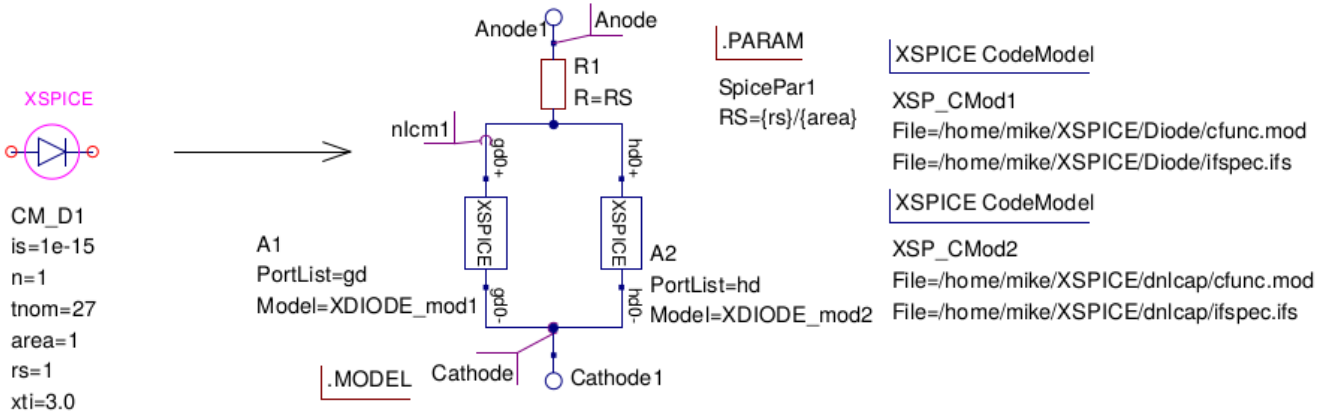
- The "XSPICE generic device" component is a building block for the construction of user-defined A-devices. It is defined by a comma separated port list, with allowed XSPICE port designators, then attached to a SPICE .MODEL statement



Qucs-0.0.19-S-RC6 XSPICE Code Modeling Package: Part IV XSPICE "Turn-Key" Model Generation; Compiler System Dataflow Diagram



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:

1. P. Antognetti and G. Massobrio (Editors), "Semiconductor device modeling with SPICE, 1988, McGraw-Hill Book Componey, New York, pp1-32.
2. S. Jahn amd M.E. Brinson, "Interactive device modelling using Qucs equation-defined devices., 2008, International Journal of Numerical Modelling: Electrical Networks, Devices and Fields, 21:335-249, DOI: 10.1002/jnm.676.

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 = 5.5406334e34;
   double GMIN = 1e-12;

    PTNOM = PARAM(tnom)+273.15;
    PTEMP = TEMPERATURE+273.15;
    Vt_temp = 8.65387195e-5*PTEMP;
    PEG = PARAM(eg);
    PIS   = PARAM(is);
    PN    = PARAM(n);
    PAREA = PARAM(area);
    PXTI  = PARAM(xti);

    P1 = 1/(PN*Vt_temp);
    Tr = PTEMP/PTNOM;
    Is_temp = PAREA*PIS*exp( (PXTI/PN)*log(Tr))*exp( (-PEG/Vt_temp)*(1.0-Tr));
    P3 = -5*PN;
    P4 = Is_temp*exp80;

    if(INIT) {
       cm_analog_alloc(DERIVE, sizeof(double));
       derive = (double *)cm_analog_get_ptr(DERIVE, 0);
       *derive = 0.0;
     }
    else {
         derive = (double *)cm_analog_get_ptr(DERIVE, 0);
       }

    if(ANALYSIS != AC) {
       Vd = INPUT(diode);
       if  ( Vd > P3*Vt_temp ) {
            if (P1*Vd <= 80)  {
              Id = Is_temp*(exp(P1*Vd)-1.0) + GMIN * Vd;
              OUTPUT(diode) = Id;
              *derive = P1*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 <= -5*PN*Vt_temp){
          Id = -Is_temp+GMIN*Vd;
         OUTPUT(diode) = Id;
          *derive = GMIN;
          PARTIAL(diode, diode) = *derive;
       }
    }
}
```

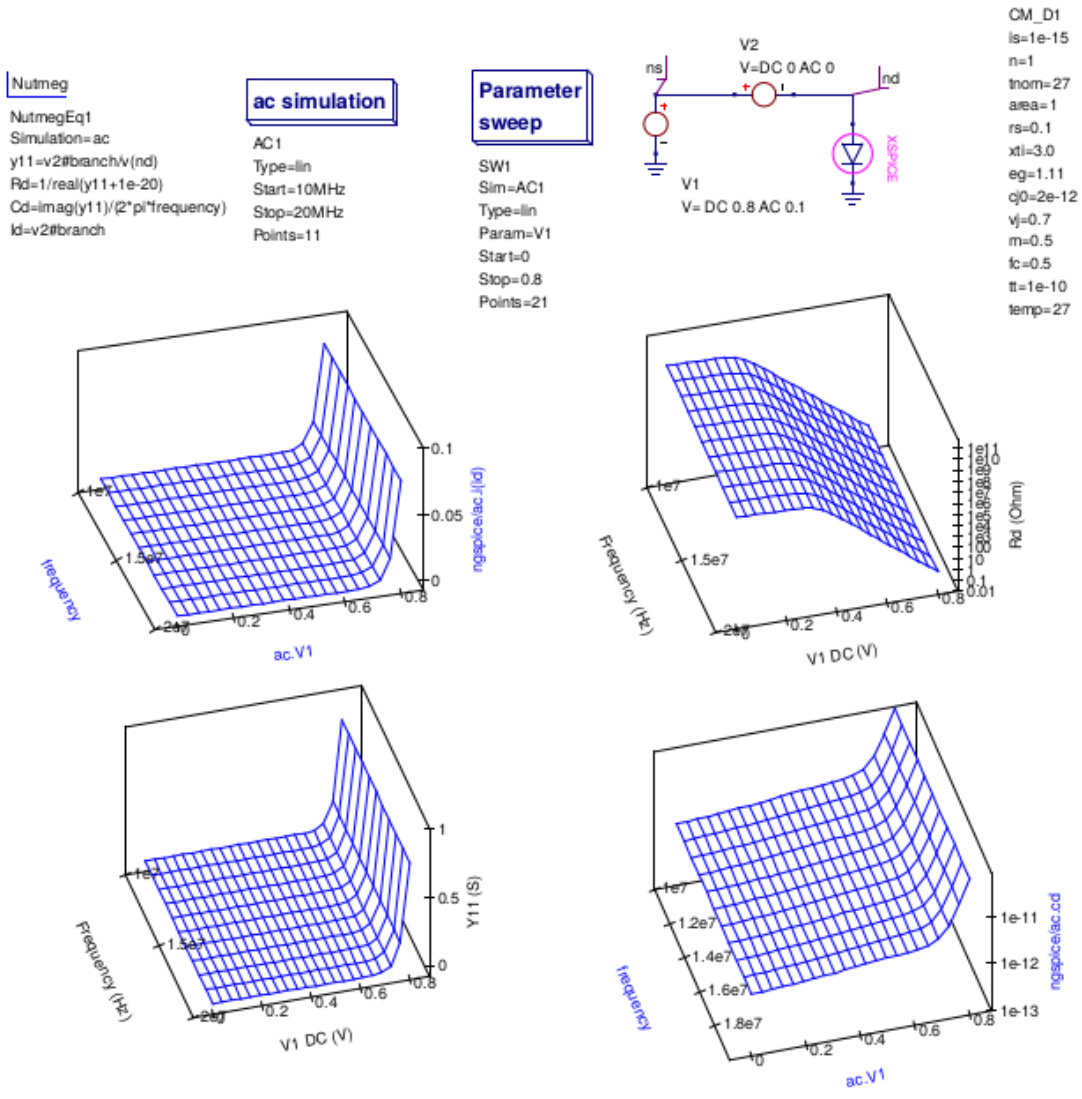
Semiconductor diode non-linear I_d / V_d characteristics, including Verilog-A limexp function and temperature effects.

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

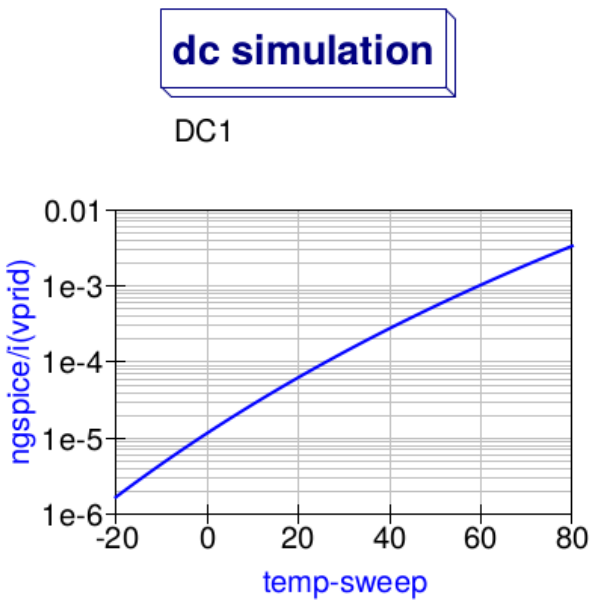
```
/*   dnlcap cm model.     4 March 2016    Mike Brinson
   This file contains the model code for an experimental
   semiconductor diode capacitance: both Cdep and Cdiff are modelled.
   This is  used as a test bench for constructing compact device models
   with the Qucs-0.0.19-S automatic XSPICE 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 CVC 0
#include <math.h>
void cm_dnlcap(ARGS)
{
  Complex_t ac_gain;
  static double PCJ0, PVJ, PM, PFC, PTT, PIS, PN;
  double P1, Vd, partial, Vt_temp;
  double PTEMP, W1, Wr, Wi, Rd;
  double *cvc;
  static double cap, F2, F3, cdep, derive, Id, P3, P4;
  double Rp = 1.0e12;
  double exp80 = 5.5406334e34;
  double GMIN = 1e-12;
  PTEMP = TEMPERATURE+273.15;
   Vt_temp = 8.65387195e-5*PTEMP;
   P1   = 1/(Vt_temp);
    if(INIT==1) {
      cm_analog_alloc(CVC, sizeof(double));
      cvc = (double *) cm_analog_get_ptr(CVC, 0);
      *cvc = 0.0;
      cap = 1e-16;
      derive = 1e-20;
      PCJ0 = PARAM(cj0);
      PVJ  = PARAM(vj);
      PM   = PARAM(m);
      PFC  = PARAM(fc);
      PTT  = PARAM(tt);
      PIS  = PARAM(is);
      PN   = PARAM(n);
      F2 = exp( (1+PM)*log(1-PFC) );
      F3 = 1-PFC*(1+PM);
      P3 = -5*PN;
      P4 = PIS*exp80;
    }
   else {
     cvc = (double *) cm_analog_get_ptr(CVC,0);
    }
   if (ANALYSIS != AC) {
    Vd =  *cvc;
    if  ( Vd > P3*Vt_temp ) {
          if (P1*Vd <= 80)  {
            Id = PIS*(exp(P1*Vd)-1.0) + GMIN * Vd;
            derive = P1*PIS*exp(P1*Vd)+GMIN;
           }
           else {
             Id = P4*(1+(P1*Vd-80))+GMIN*Vd;
             derive = P1*P4+GMIN;
           }
    }
    else {
       Id = -PIS+GMIN*Vd;
       derive = GMIN;
    }
    if ( Vd < PFC*PVJ ) {
        cdep = PCJ0/exp(PM*log(1.0 - (Vd/PVJ)) );
    }
    else {
        cdep = (PCJ0/F2)*(F3+(PM*Vd/PVJ));
    }
    cap = PTT*Id/Vt_temp + cdep;
   }
   if (ANALYSIS == DC) {
     *cvc = INPUT(dnlcap)*Rp;
     OUTPUT(dnlcap) =  *cvc;
     PARTIAL(dnlcap, dnlcap) =  Rp;
   }
   if (ANALYSIS == TRANSIENT) {
     cm_analog_integrate(INPUT(dnlcap) / (cap + 1e-17), cvc, &partial);
     partial /= cap;
     OUTPUT(dnlcap) = *cvc;
     PARTIAL(dnlcap, dnlcap) = partial;
   }
   if (ANALYSIS == AC) {
     Rd = 1/derive;
     W1  = 1+RAD_FREQ*RAD_FREQ*Rd*Rd*cap*cap;
     Wr  =  Rd/W1;
     Wi  =  RAD_FREQ*cap*Rd*Rd/W1;
     ac_gain.real = Wr;
     ac_gain.imag = -1.0*Wi;
     AC_GAIN(dnlcap, dnlcap) = ac_gain;
   }
}
```

Semiconductor diode non-linear capacitance characteristics, including depletion and diffusion components

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

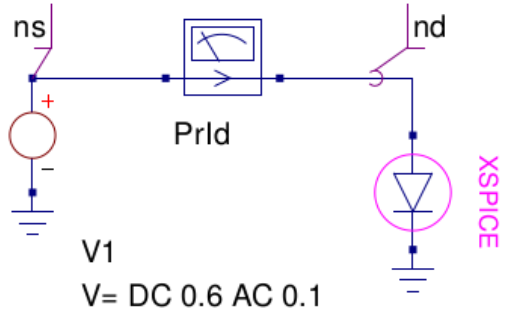


Qucs-0.0.19-S-RC6 XSPICE Code Modeling Package: Part V XSPICE Diode Model - (e) The Diode Id/Vd Temperature Variation



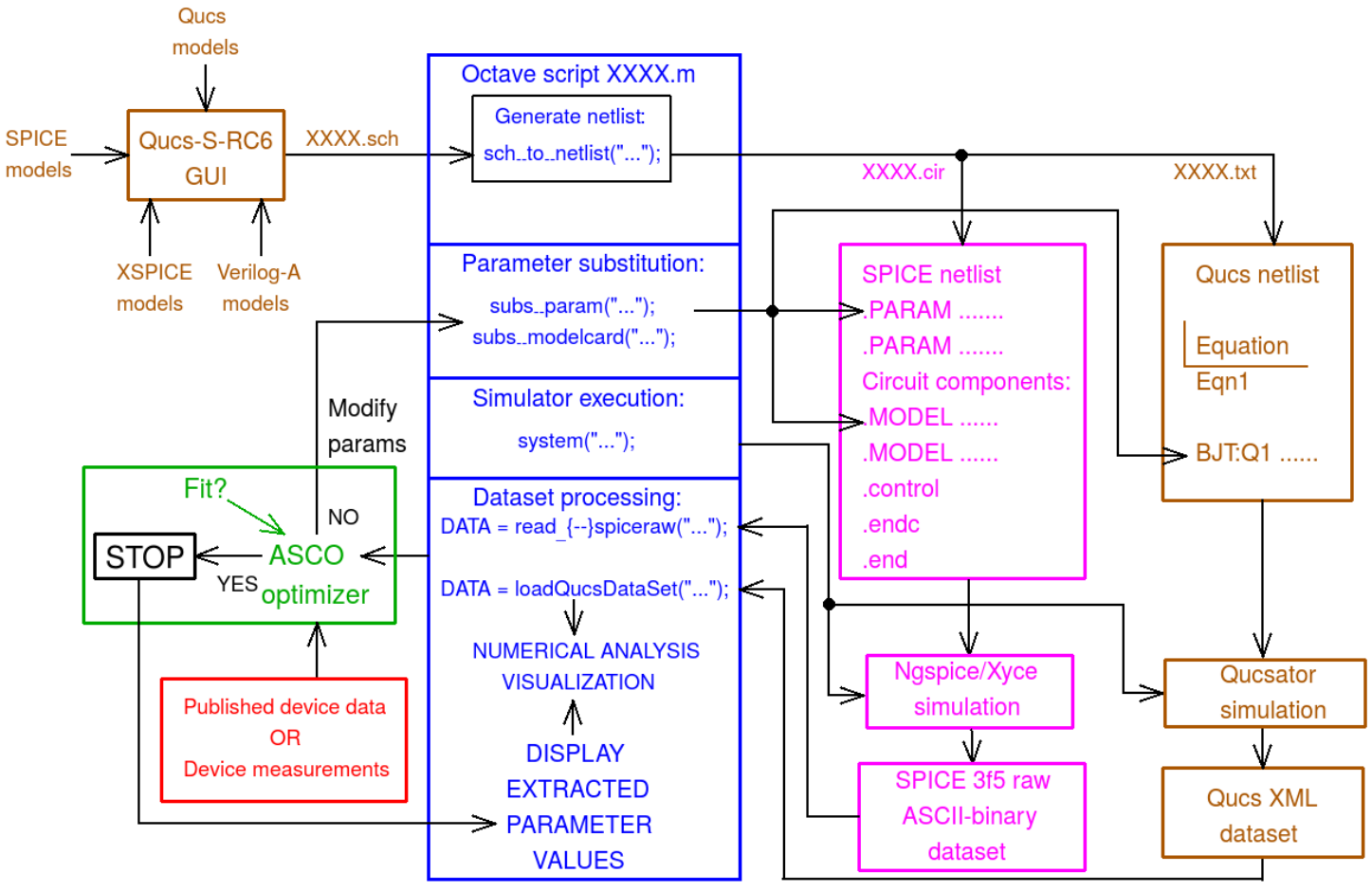
Parameter sweep

SW1
Sim=DC1
Type=lin
Param=temp
Start=-20
Stop=80
Points=101



CM_D1
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 Maufacturer's Data or Measurements Controlled by Octave Script Files: Part I Structure Diagram



Qucs-0.0.19-S-RC6/Ngspice/Xyce Circuit Analysis and Compact Device Parameter Extraction from Maufacturer's Data or Measurements Controlled by Octave Script Files: Part II Octave Package

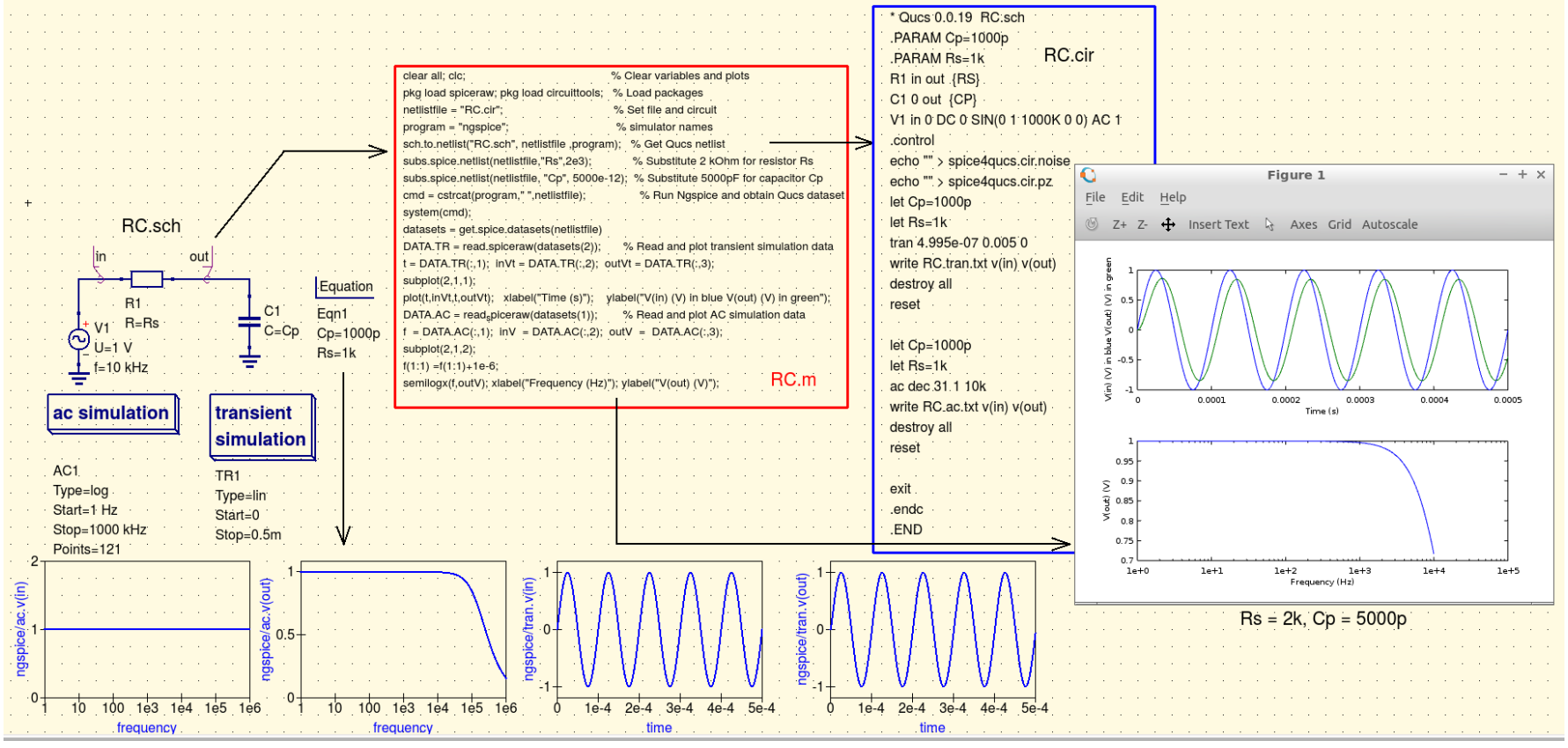
- The main purposes of Octave integration are:
  - Parameter substitution in Qucs and SPICE netlists,
  - Simulation process control from Octave,
  - Simulator output dataset (SPICE3f5-raw and Qucs XML) loading into Octave matrix structures,
  - Compact model parameter extraction from simulation and 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

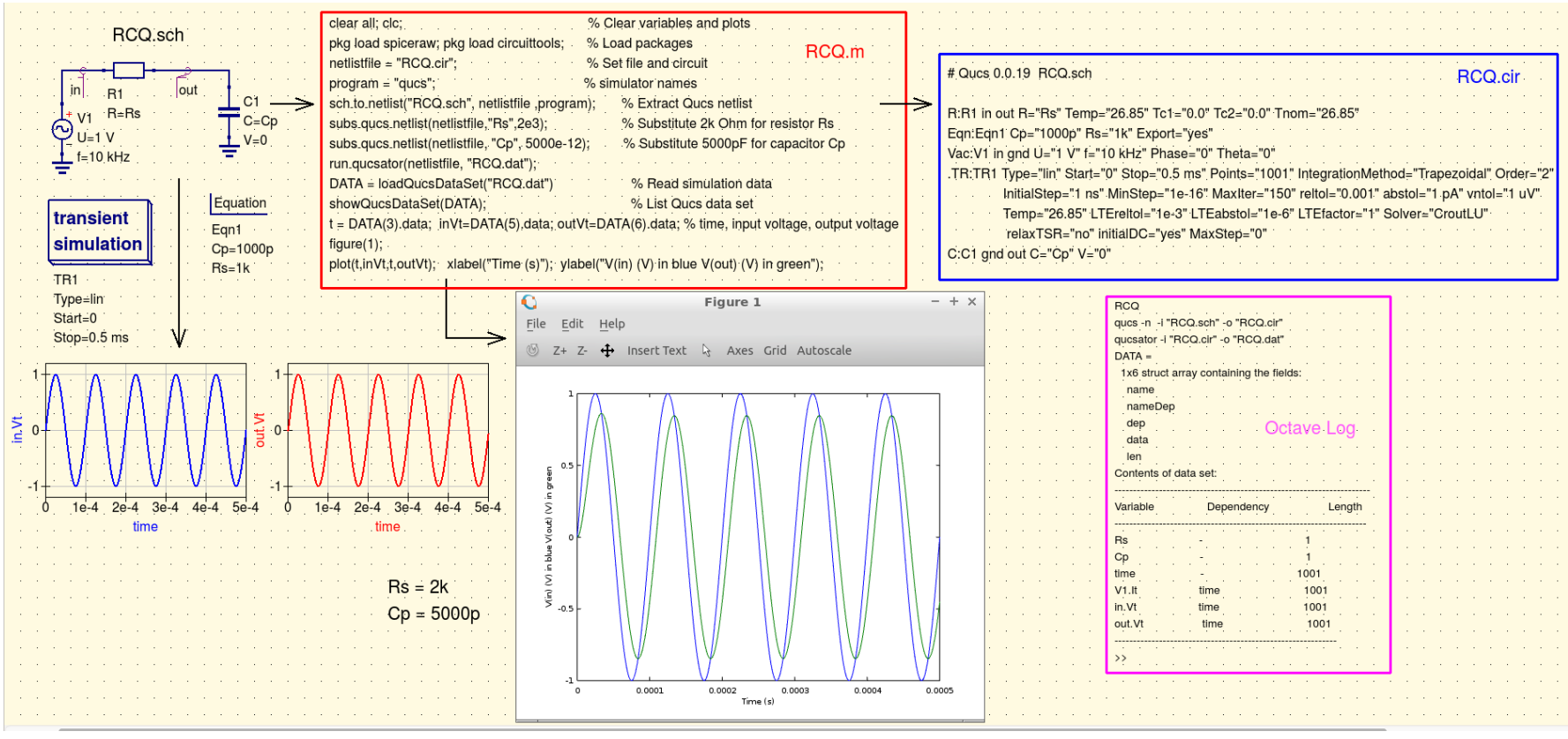
PARAM – represents a SPICE or Qucs variable or .MODEL parameter name

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 Maufacturer'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 Maufacturer's Data or Measurements Controlled by Octave Script Files: Part IV Simple Qucs example



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

- 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*/