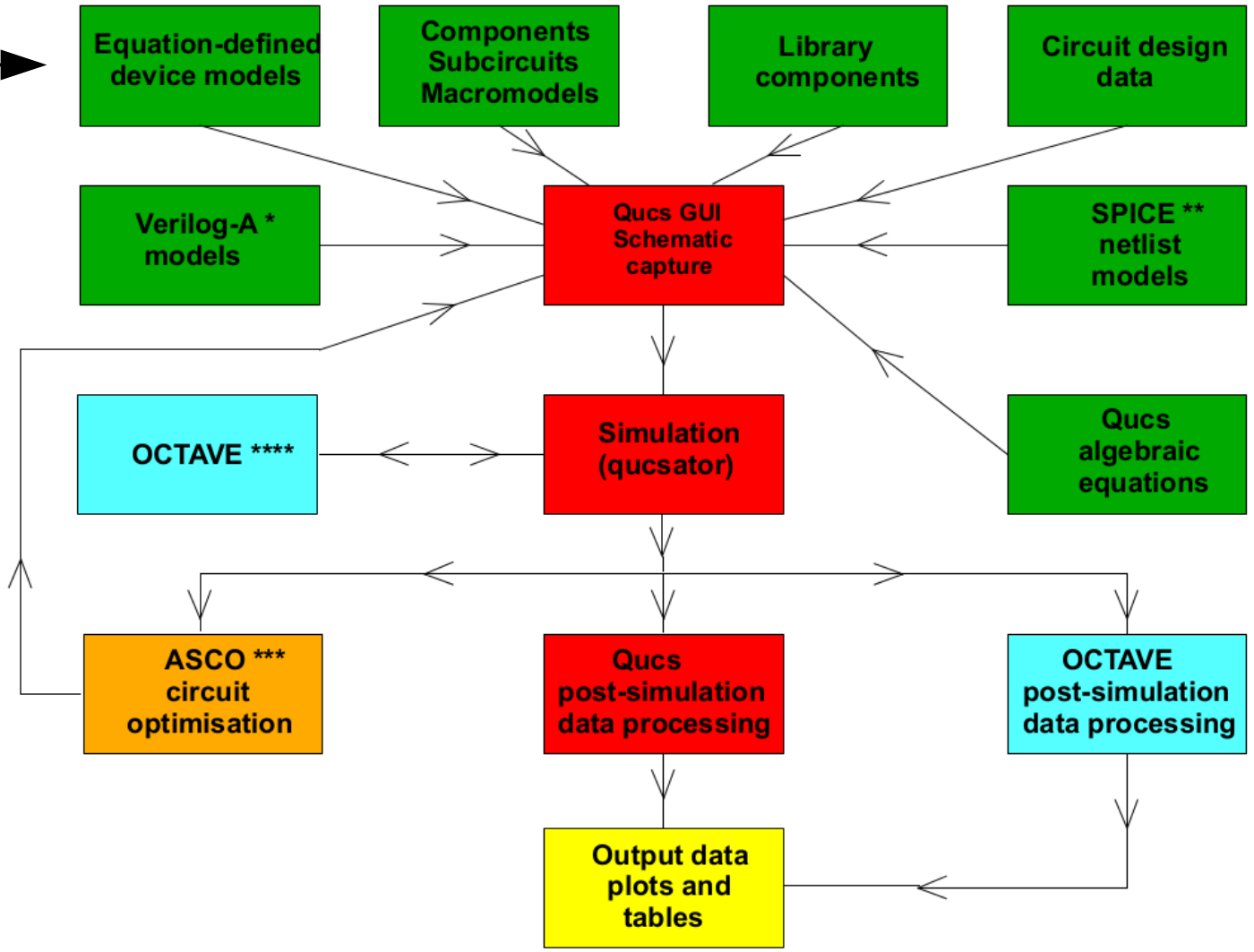
A unified approach to compact device modelling with the open source packages Qucs/ADMS and MAPP/Octave

Mike Brinson, Centre for Communications Technology, London Metropolitan University, UK,
mbrin72043@yahoo.co.uk.

Presented at the IEEE EDS POLAND MINI-COLLOQUIUM - Training Course on Compact Modeling (TCCM), Torun Poland, 24 June 2015

Qucs-0.0.18: structure and basic modelling features

Qucs models and modelling features



Numerical processing linked to qucsator

GPL software used by Qucs-0.0.18

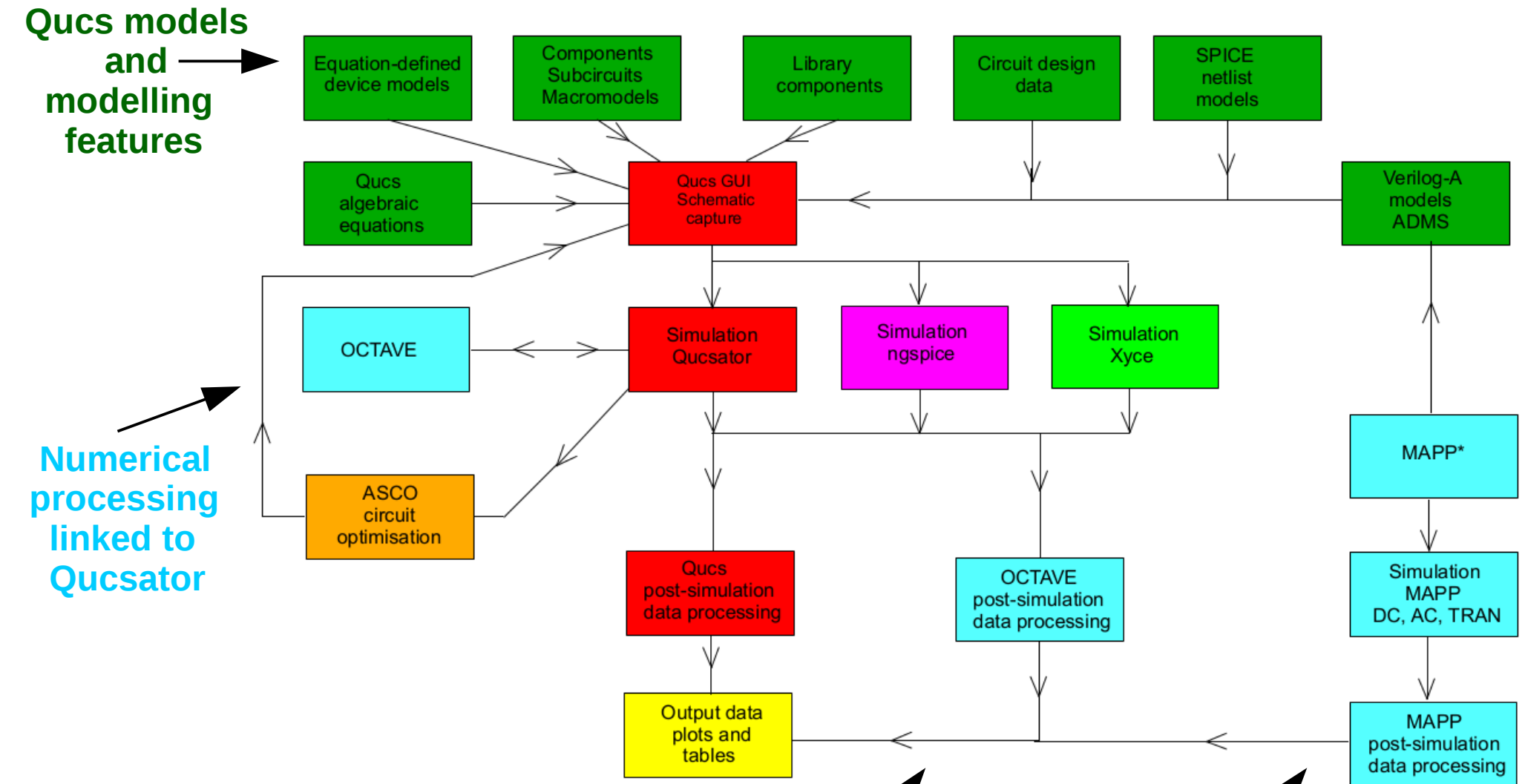
* ADMS – Automatic model synthesiser, http://sourceforge.net/projects/mot-adms

** PS2SP – PSPICE to SPICE preprocessor, http://members.acon.at/fschmid7/

*** ASCO – SPICE circuit optimiser, http://asco.sourceforge.net/

**** OCTAVE – Numerical analysis package, https://www.gnu.org/software/octave/

Qucs-0.0.19: introducing the next generation Qucs simulation and modelling tools



* Berkeley Model and Algorithm Prototyping Platform (MAPP)
http://draco.eecs.berkeley.edu:8765/jr/MAPP/wikis/home

Numerical post-simulation data processing

+ MAPP device modelling and simulation

Qucs circuit simulation and device modelling: Simulation

Schematic capture: generates circuit diagram → **Circuit simulation: DC, AC, TRAN, S-parameter, noise and Harmonic Balance** → **Post simulation data processing and visualization**

Circuit encoding

ngspice and Xyspice netlist

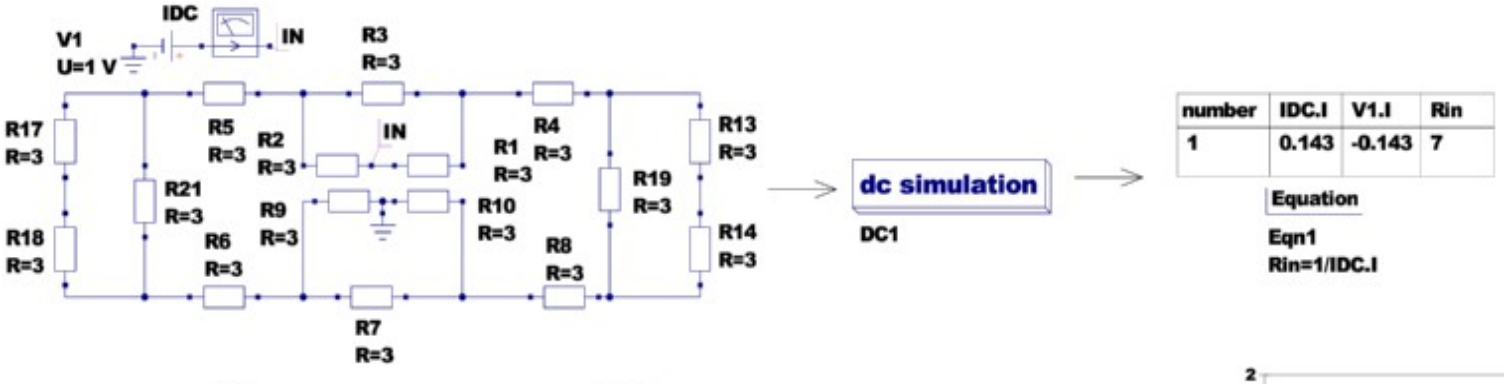
SPICE pre - processor

SPICE 2g6 And 3f5 netlist

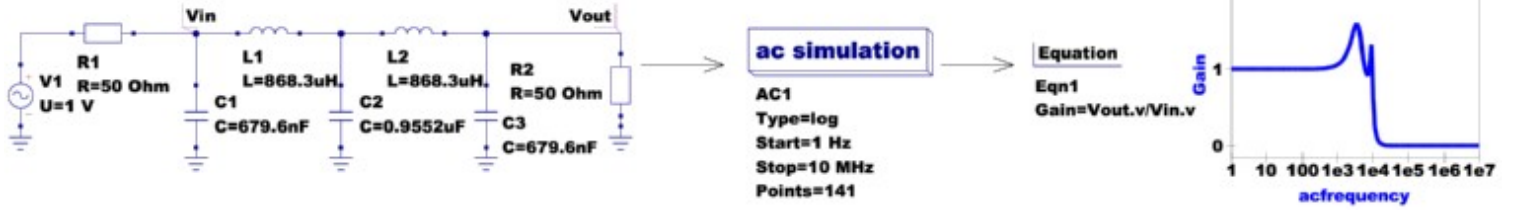
Qucs schematic diagram

Qucs netlist

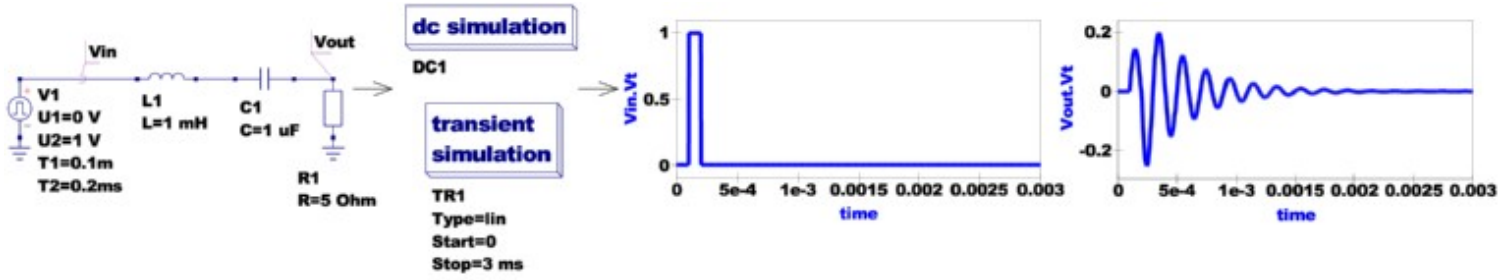
DC



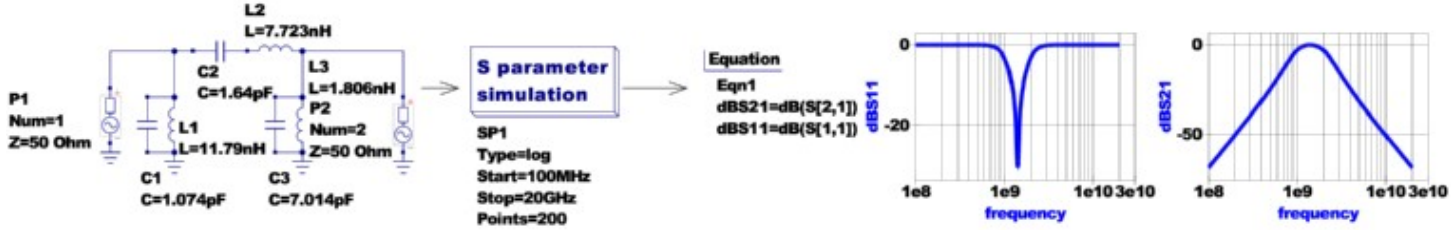
AC



TRAN



S-parameter*



***Implementation : Qucs built-in; SPICE via RCL networks**

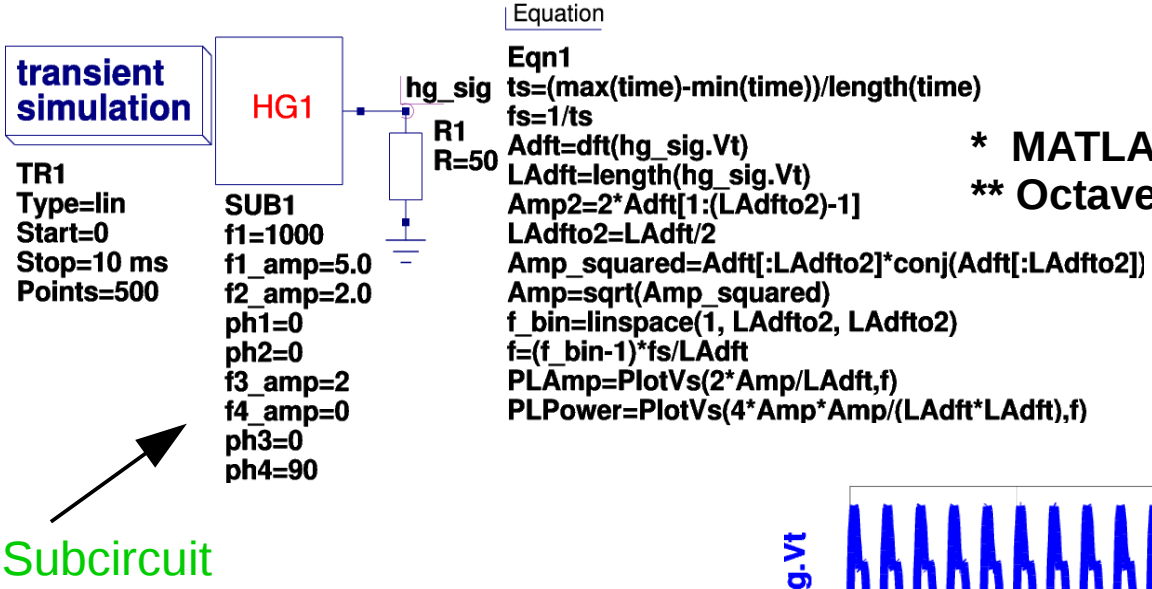
Post-simulation data processing : 1. using Qucs

Equation blocks + simulation data sets → **Data processing** → **Tables and plots**

Constants: i, j, pi, e, kB, q
Immediate: 2.5, 1.4+j5.1, [1, 3, 4, 5, 7], [11, 12; 21, 22]
Ranges: Lo:Hi, :Hi, Lo:, :
Logical operators: !x, x&&y, x||y, x^^y, x?y:z, x==y, x!=y, x<y, x<=y,x>y, x>=y

Number suffixes: E, P, T, G, M, k, m, u, n, p, f, a
Matrices: M, M[2,3], M[:,3]
Arithmetic operators: +x, -x, x+y, x-y, x*y, x/y, x%y, x^y

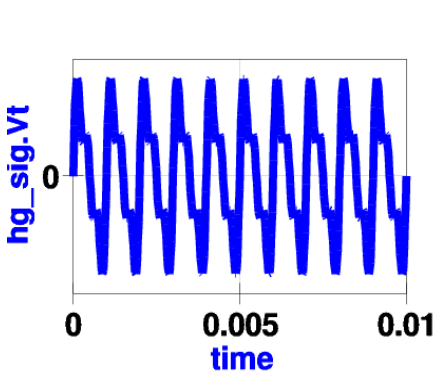
abs adjoint angle arccos arccosec arccot arcosech arcosh arcoth arcsec arcsin arctan arg arsech arsinh artanh avg besseli0 besselj bessely ceil conj cos cosec cosech cosh cot coth cumavg cumprod cumsum dB dbm dbm2w deg2rad det dft diff erf erfc erfcinv erfinv exp eye fft fix floor Freq2Time GaCircle GpCircle hypot idft ifft imag integrate interpolate inverse kbd limexp linspace ln log10 log2 logspace mag max min Mu Mu2 NoiseCircle norm phase PlotVs polar prod rad2deg random real rms Rollet round rtoswr rtoy rtoz runavg sec sech sign sin sinc sinh sqr sqrt srandom StabCircleL StabCircleS StabFactor StabMeasure stddev step stos stoy stoz sum tan tanh Time2Freq transpose twoport unwrap variance vt w2dbm xvalue ytor ytos ytoz yvalue ztor ztos ztoy

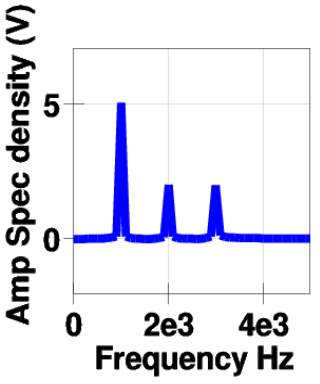


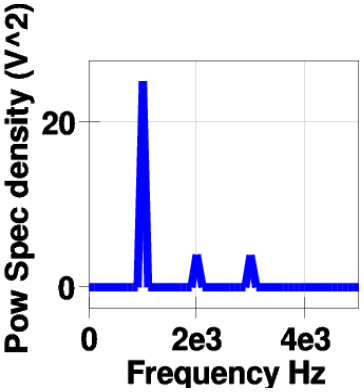
Subcircuit

* MATLAB, Mathworks, http://www.mathworks.com/
** Octave, http://www.gnu.org/software/octave/

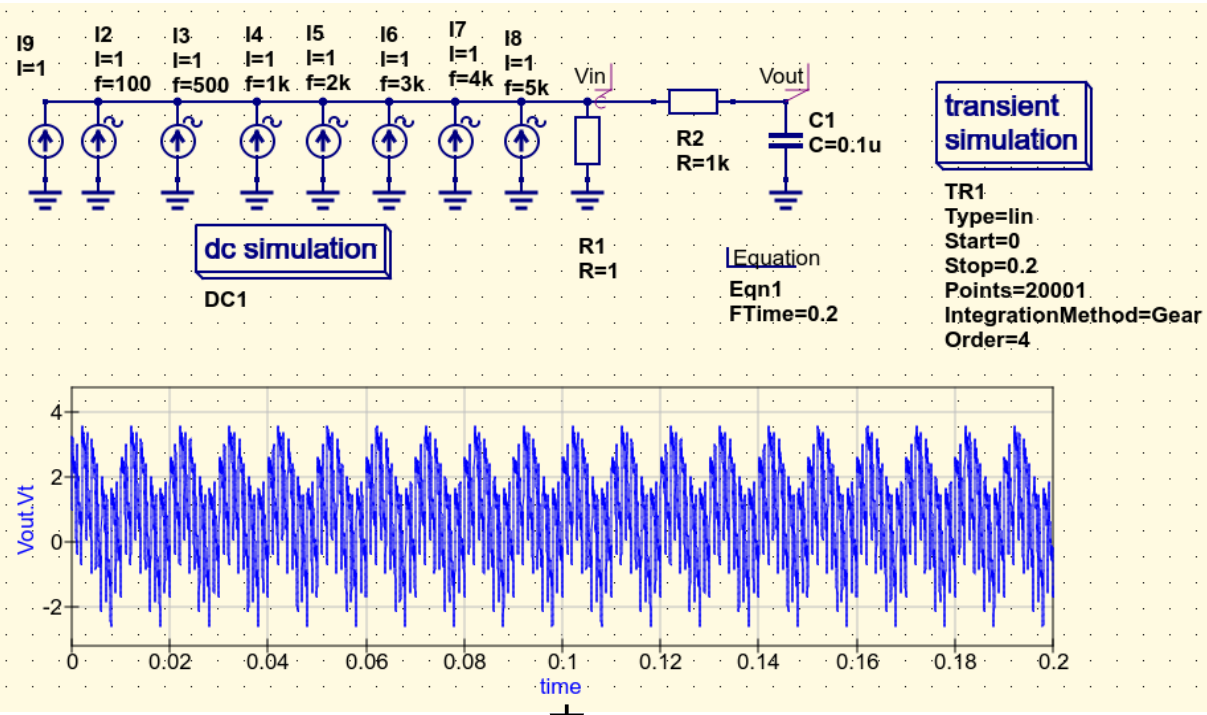
Limitations: NO user defined functions or control loops

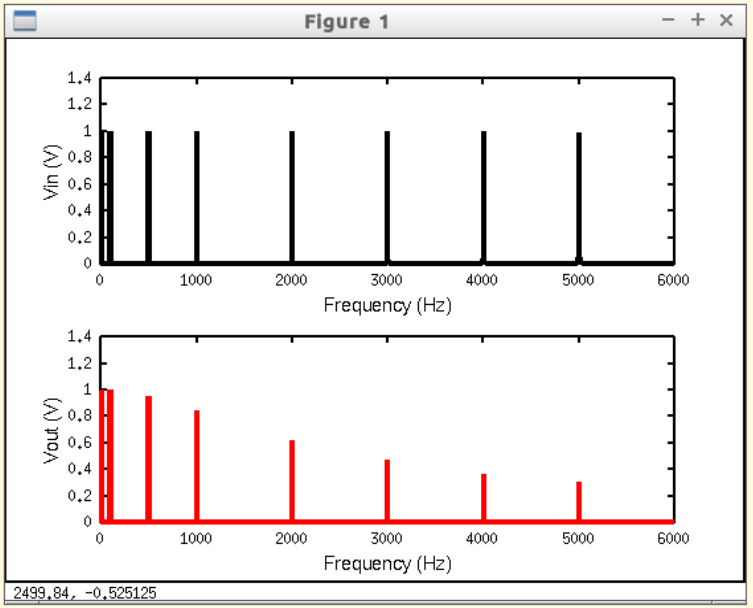


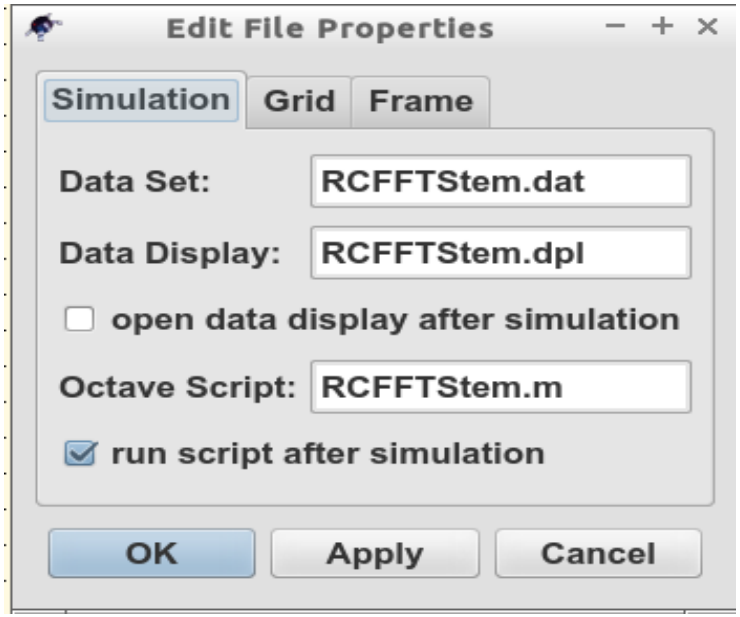




Post-simulation data processing : 2. using Octave



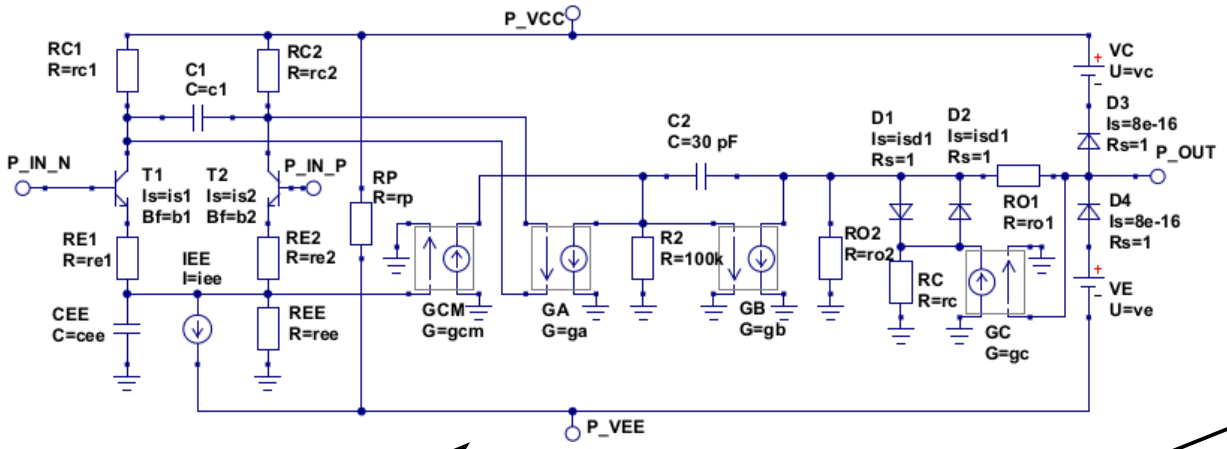


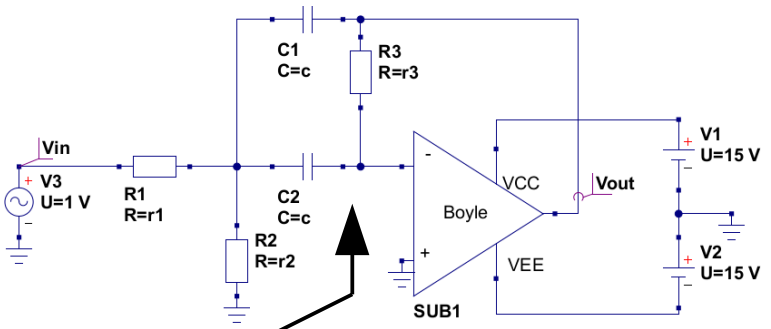


RCFFTStem.m

```
% test RC transient simulation plus FFT of output.
clear;
Data = "RCFFTStem.dat";
qdset = loadQucsDataSet(Data);
[time] = getQucsVariable(qdset,  "time");
[Vin_Vt] = getQucsVariable(qdset,  "Vin.Vt");
[Vout_Vt] = getQucsVariable(qdset,  "Vout.Vt");
[FTime] = getQucsVariable(qdset,  "FTime");
showQucsDataSet(qdset);
[Y1 , Y2, freq] = plotFFT2V( "Stem", "Frequency (Hz)",0,6000,
                             Vin_Vt, "Vin (V)", "black",
                                 Vout_Vt,"Vout (V)", "red", 4, FTime);
```

Algebraic equation-defined components and embedded design routines





Equation
Eqn1
Q=f0/bw
r1=r3/(2*h0)
c=1/(pi*r3*bw)
r2=r3/(4*Q^2-2*h0)

Equation
Eqn2
gain=dB(Vout.v)
phase=phase(Vout.v)

Equation
Eqn3
bw=bandwidth
r3=100e3
f0=1e3
h0=1

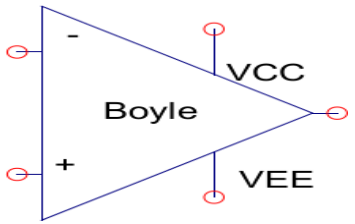
dc simulation
DC1

ac simulation
AC1
Type=lin
Start=800Hz
Stop=1200Hz
Points=21

Parameter sweep
SW1
Sim=AC1
Type=lin
Param=bandwidth
Start=100
Stop=400
Points=10

Equation
Eqn1
is1=is
ic1=0.5*c2*pswrt
ic2=ic1
VT=vt(300)
is2=is1*(1+vos/VT)
ib1=ib-0.5*ios
ib2=ib+0.5*ios
b1=ic1/ib1
b2=ic2/ib2
iee=ic1/((b1+1)/b1)+((b2+1)/b2))
gm1=ic1/VT
rc1=1/(2*pi*gbp*c2)
rc2=rc1
re1=((b1+b2)/(2+b1+b2))*(rc1-1/gm1)
re2=re1
ree=va/iee
cee=(2*ic1/nswrt)-c2
dphi=90-pm
c1=0.5*c2*tan(dphi*pi/180)
gcm=1/(cmrr*rc1)
ga=1/rc1
gb=(avol*rc1)/(r2*ro2)
ix=2*ic1*r2*gb-is1
temp1=-(ro1*is1/VT)
isd1=ix*exp(temp1)+1e-32
temp2=ix/isd1
rc=VT*ln(temp2)/(100*ix)
gc=1/rc
vc=abs(vcc)-vsp+VT*ln(iscp/is1)
ve=abs(vee)-vsn+VT*ln(iscn/is1)
rp=(vcc-vee)*(vcc-vee)/pd

Macromodel design equations

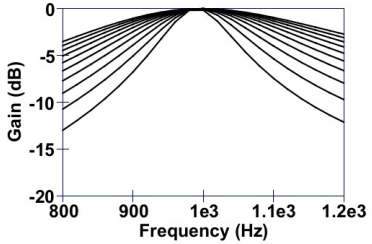


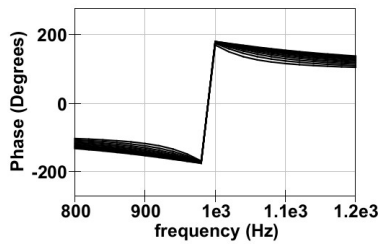
SUB1
is=8e-16
vos=0.7e-3
c2=30e-12
r2=100e3
pswrt=0.625e6
nswrt=0.5e6
ios=20e-9
ib=80e-9
gbp=1e6
va=200
pm=70
cmrr=31622.8
avol=200e3
vsp=14.2
vsn=-13.5
vcc=15
vee=-15
iscp=25e-3
iscn=25e-3

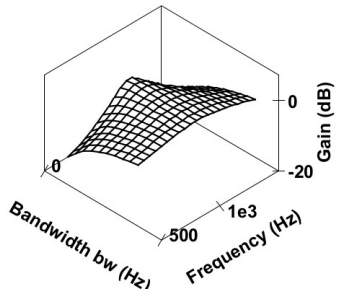
Macromodel parameters

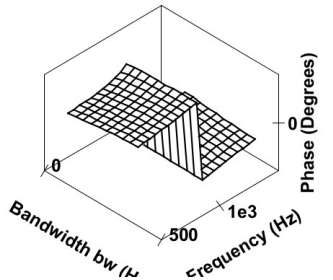
Circuit design equations

Filter specification





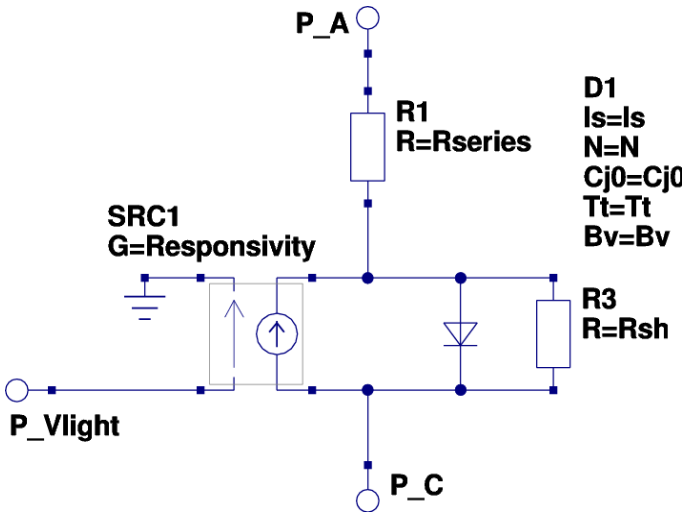




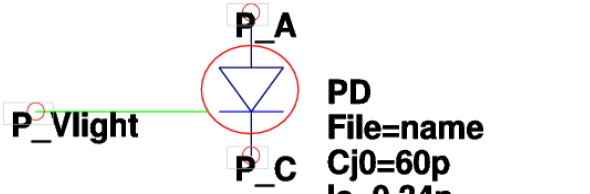
M.E. Brinson and H. Nabijou, Adaptive subcircuits and compact Verilog-A macromodels as integrated design and analysis blocks in Qucs circuit simulation, International Journal of Electronics, Vol. 98, No. 5, June 2011 , 631-645.

Compact Device Modelling 1: Subcircuits and EDD models

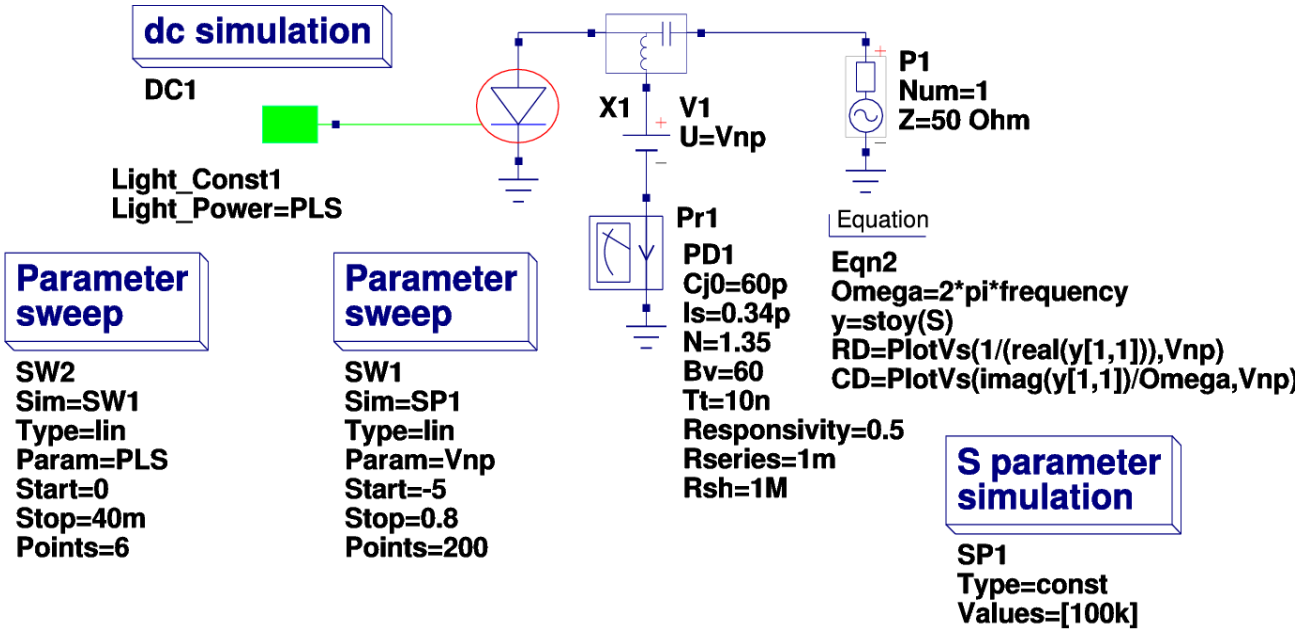
Photodiode subcircuit body

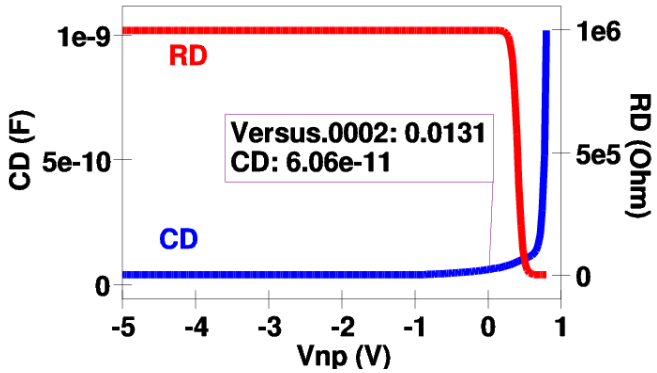


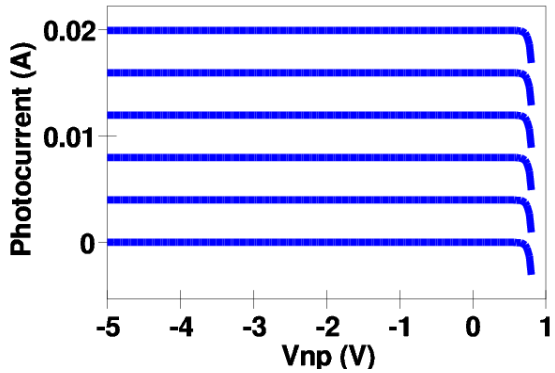
Photodiode symbol



Test circuit and simulation data

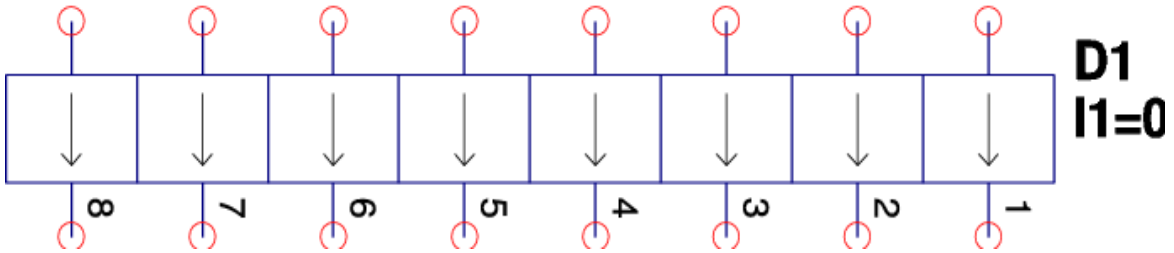






Green denotes light source and light bus

Compact Device Modelling 1: EDD specifications



$$I = I(V)$$

$$g = \frac{dI}{dV}$$

$$Q = Q(V, I)$$

$$c = \frac{dQ}{dV}$$

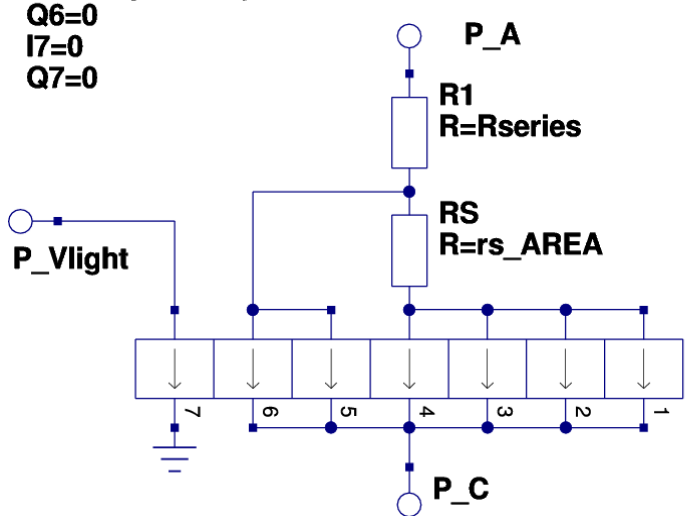
$$c = \frac{dQ(V)}{dV} + \frac{dQ(I)}{dV}$$

- **EDD is a multi-terminal non-linear component with branch currents that can be functions of EDD branch voltage, and stored charge that can be a function of both EDD branch voltages and Currents.**
- **EDD is similar, but more advanced to the SPICE 3f5 B type controlled sources.**
- **EDD can be combined with conventional circuit components and Qucs equation blocks when constructing compact device models and subcircuit macromodels.**
- **EDD is an advanced component, allowing users to construct prototype experimental models from a set of equations derived from physical device properties.**

Jahn S., Brinson M. and Margraf M. Interactive compact device modeling using Qucs equation defined devices, ESSDERC/ESSCIRC MOS-AK Workshop, Munich 2007

Compact Device Modelling 1: EDD Photo-diode model

D1
I1=(V1>-5.0*N*Vt_T2) ? Area*Is_T2*(limexp(V1/(N*Vt_T2))-1.0)+V1*GMIN : 0
Q1=(V1 < Fc*Vj) ? Tt*I1+Area*(Cj0_T2*Vj_T2/(1-M))*(1-(1-V1/Vj_T2)^(1-M)) : 0
I2=((-Bv<V1) && (V1<-5.0*N*Vt_T2)) ? -Area*Is_T2+V1*GMIN : 0
Q2=(V1 >= Fc*Vj) ? Tt*I1+Area*Cj0_T2*(F1+(1/F2)*(F3*(V1-Fc*Vj_T2)+(M/(2*Vj_T2))*(V1*V1-Fc*Fc*Vj_T2*Vj_T2))) : 0
I3=(V1==-Bv) ? -Ibv : 0
Q3=0
I4=(V1<-Bv) ? -Area*Is_T2*(limexp(-(Bv+V1)/Vt_T2)-1.0+Bv/Vt_T2) : 0
Q4=0
I5=V5/(Rsh+1e-20)
Q5=0
I6=-Responsivity*V7
Q6=0
I7=0
Q7=0

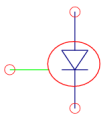


Equation

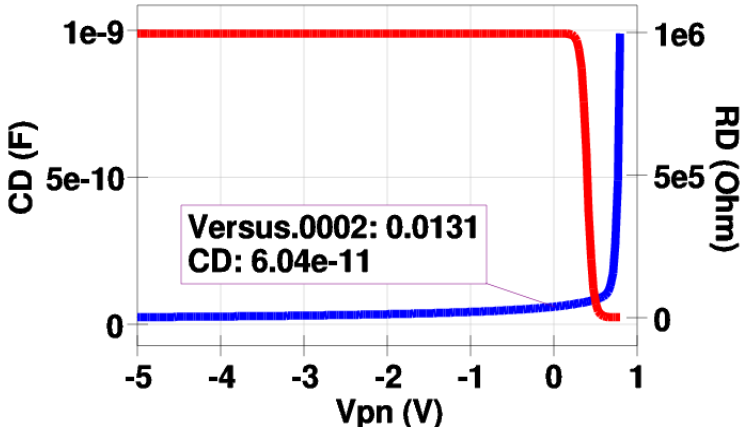
Eqn2
Cj0_T2=Cj0*(1+M*(400e-6*(T2-T1)-(Vj_T2-Vj)/Vj))
Vt_T2=(kB*T2/q)
rs_AREA=Rs/AREA
GMIN=1e-12
A=7.02e-4
B=1108
T1=Tnom+273.15
Vj_T2=(T2/T1)*Vj-(2*kB*T2/q)*ln((T2/T1)^1.5)-((T2/T1)*Eg_T1-Eg_T2)
Is_T2=Is*(T2/T1)^(Xti/N)*limexp((-(q*Eg_T1)/(kB*T2))*(1-T2/T1))
Eg_T1=Eg-A*T1*T1/(B+T1)
Eg_T2=Eg-A*T2*T2/(B+T2)
T2=Temp+273.15

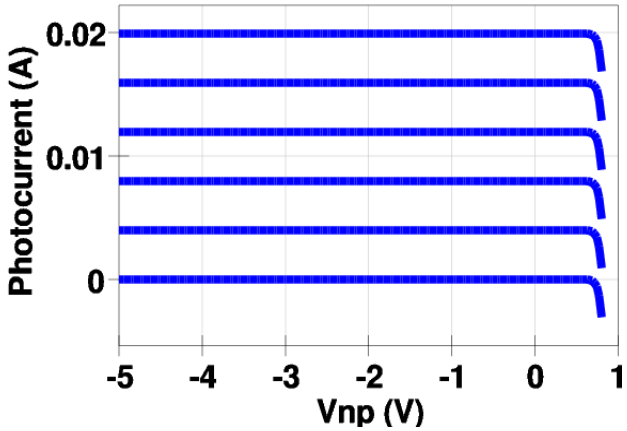
Equation

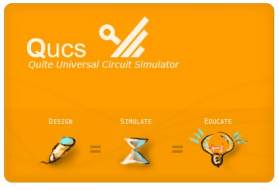
Eqn3
F1=(Vj/(1-M))*(1-(1-Fc)^(1-M))
F2=(1-Fc)^(1+M)
F3=1-Fc*(1+M)



PD1
N=1.35
Is=0.34p
Bv=60
Ibv=1e-3
Vj=1.0
Cj0=60p
M=0.5
Area=1
Fc=0.5
Tt=10n
Xti=3.0
Tnom=26.85
Temp=26.85
Eg=1.16
Responsivity=0.5
Rsh=1M
Rseries=1m
Rs=0.01

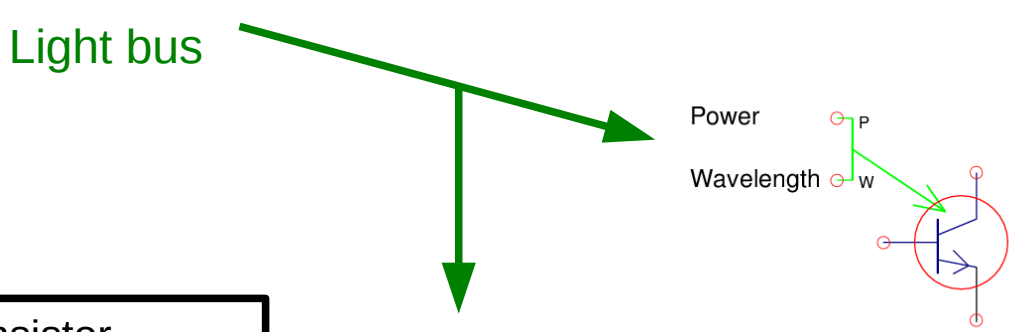


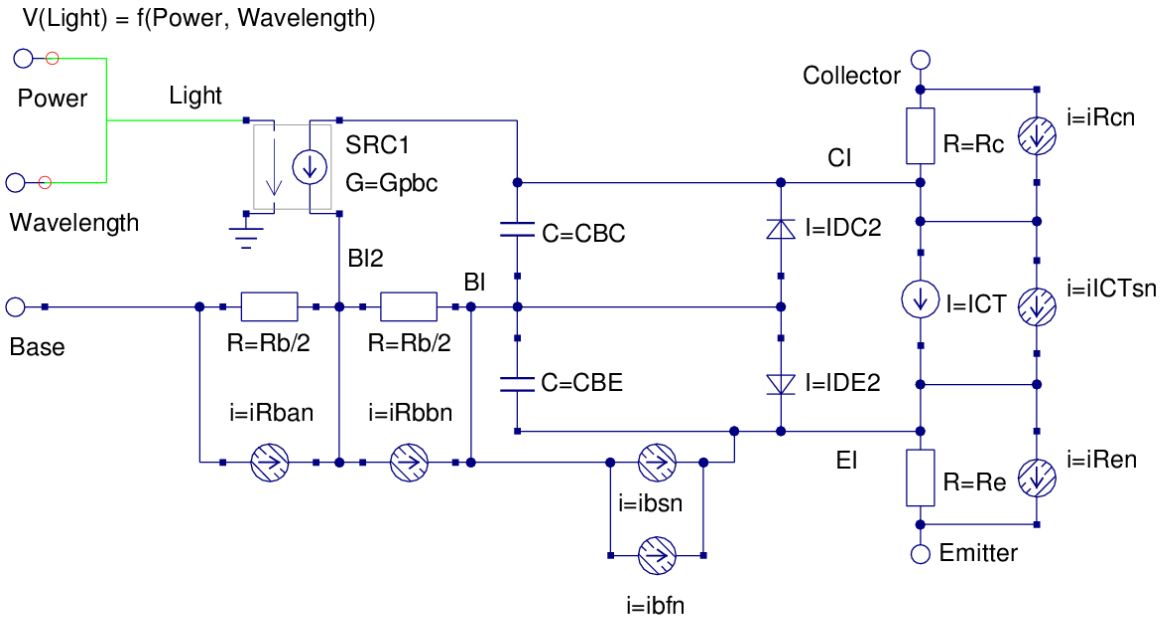




Compact Device Modelling 1: EDD photo-transistor model

Schematic symbol, default parameter list and equivalen circuit of an npn bipolar photo-transistor





The phototransistor consists of an Ebors-Moll bipolar junction transistor model which has been extended to include depletion and diffusion capacitance, forward and reverse Early effects, high current forward and reverse beta degradation, thermal and shot noise, plus a light bus which connects external light signals to the phototransistor

Ptran1
Bf=100
Br=0.1
Is=1e-10
Nf=1
Nr=1
Var=100
Vaf=100
Mje=0.33
Vje=0.75
Cje=1p
Mjc=0.33
Vjc=0.75
Cjc=2p
Tr=100n
Tf=0.1n
Ikf=0.5
Ikr=0.5
Rc=2
Re=1
Rb=100
Kf=1e-12
Ffe=1.0
Af=1.0
Responsivity=1.5
P0=2.6122e3
P1=-1.4893e1
P2=3.0332e-2
P3=-2.5708e-5
P4=7.6923e-9

Compact Device Modelling 1: Photo-transistor specification

Model parameters

Name	Symbol	Description	Unit	Default
BF	βf	Forward beta		100
BR	βr	Reverse beta		0.1
Is	Is	Saturation current	A	1e-10
Nf	*Nf*	Forward emission coefficient		1
Nr	*Nr*	Reverse emission coefficient		1
Var	*Var*	Reverse Early voltage	V	100
Vaf	*Vaf*	Forward Early voltage	V	100
Mje	*Mje*	Base-emitter exponential factor		0.33
Vje	*Vje*	Base-emitter built-in potential	V	0.75
Cje	Cje	Base-emitter zero-bias depletion capacitance	F	1p
Mjc	*Mjc*	Base-collector exponential factor		0.33
Vjc	*Vjc*	Base-collector built-in potential	V	0.75
Cjc	Cjc	Base-colector zero-bias depletion capacitance	F	1p
Tr	*Tr*	Ideal reverse transit time	s	100n
Tf	*Tf*	Ideal forward transit time	s	0.1n
Ikf	*Ikf*	High current corner for forward beta		A 0.5
Ikr	*Ikr*	High current corner for reverse beta		A 0.5
Rc	*Rc*	Collector series resistance	Ω	2
Re	*Re*	Emitter series resistance	Ω	1
Rb	*Rb*	Base series resistance	Ω	100
Kf	*Kf*	Flicker noise coefficient		1e-12
Ffe	*Ffe*	Flicker noise frequency exponent		1
Af	*Af*	Flicker noise exponent		1
Responsivity	*Responsivity*	Responsivity at peak wavelength	A/W	1.5
P0	*P0*	Relative selectivity polynomial coefficient	%	2.6122×10^{3}
P1	*P1*	Relative selectivity polynomial coefficient	%/nm	-1.4893×10^{1}
P2	*P2*	Relative selectivity polynomial coefficient	%/nm^2	3.0332×10^{-2}
P3	*P3*	Relative selectivity polynomial coefficient	%/nm^3	-2.5708×10^{-5}
P4	*P4*	Relative selectivity polynomial coefficient	%/nm^4	7.6923×10^{-9}

Compact Device Modelling 1: Photo-transistor model equations

DC I/V characteristics

$$IEC = Is \cdot \left[\exp\left(\frac{V(BI,CI)}{Nr \cdot vt(300)} \right) - 1 \right] \qquad IDC2 = \frac{IEC}{\beta r} + GMIN \cdot V(BI,CI) \qquad IDE2 = \frac{ICC}{\beta f} + GMIN \cdot V(BI,EI)$$

$$ICC = Is \cdot \left[\exp\left(\frac{V(BI,EI}{Nf \cdot vt(300)} \right) - 1 \right] \qquad ICT = \frac{[ICC - IEC]}{\left(\frac{q1}{2}\right) \cdot (1 + \sqrt{1 + 4 \cdot q2})}$$

$$q1 = 1 + \frac{V(BI,CI)}{Vaf} + \frac{V(BI,EI)}{Var}$$

$$q2 = \frac{ICC}{Ikf} + \frac{IEC}{Ikr} \qquad vt(T) = \frac{K \cdot T}{q}$$

Capacitance

$$CBC = \frac{dQ(BI,CI)}{dV(BI,CI)} = \frac{Cjc}{\left[1 - \frac{V(BI,CI)}{Vjc}\right]^{Mjc}} + Tr \cdot \frac{dIEC}{dV(BI,CI)} \qquad \forall\ V(BI,CI) < \frac{Vjc}{2}$$

$$\forall\ V(BI,CI) >= \frac{Vjc}{2}$$

$$= 2^{Mjc} \cdot Cjc \cdot \left[\frac{2 \cdot Mjc \cdot V(BI,CI)}{Vjc} + (1 - Mjc) \right] + Tr \cdot \frac{dIEC}{dV(BI,CI)}$$

$$CBE = \frac{dQ(BI,EI)}{dV(BI,EI)} = \frac{Cje}{\left[1 - \frac{V(BI,EI)}{Vje}\right]^{Mje}} + Tr \cdot \frac{dICC}{dV(BI,EI)} \qquad \forall\ V(BI,EI) < \frac{Vje}{2}$$

$$= 2^{Mje} \cdot Cje \cdot \left[\frac{2 \cdot Mje \cdot V(BI,EI)}{Vje} + (1 - Mje) \right] + Tr \cdot \frac{dICC}{dV(BI,EI)} \qquad \forall\ V(BI,EI) >= \frac{Vje}{2}$$

Photo current

$$Iopt = Gpbc \cdot Popt \qquad Gpbc = \frac{\mathrm{Re}lSensitivity \cdot \mathrm{Re}sponsivity}{\beta f \cdot 100}$$

$$\mathrm{Re}lSensitivity = P0 + P1 \cdot \lambda + P2 \cdot \lambda^2 + P3 \cdot \lambda^3 + P4 \cdot \lambda^4$$

Noise

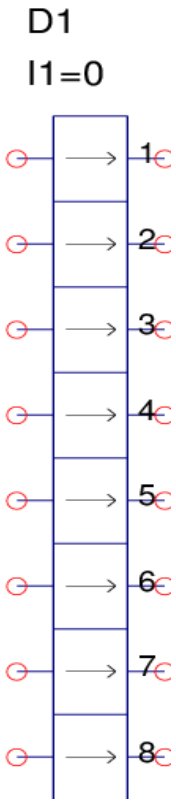
$$iRcn^2 = \frac{4 \cdot K \cdot T}{Rc} \cdot \Delta f \qquad i\mathrm{Re}n^2 = \frac{4 \cdot K \cdot T}{\mathrm{Re}} \cdot \Delta f \qquad iICTsn^2 = 2 \cdot q \cdot IC \cdot \Delta f$$

$$iRban^2 = \frac{8 \cdot K \cdot T}{Rb} \cdot \Delta f \qquad iRbbn^2 = \frac{8 \cdot K \cdot T}{Rb} \cdot \Delta f \qquad ibsn^2 = 2 \cdot q \cdot IB \cdot \Delta f \qquad ibfn^2 = Kf \cdot \frac{IB^{Af}}{f^{Ffe}} \cdot \Delta f$$

Where K is the Boltzmann constant, T is the temperature in Kelvin, q is the electron charge, GMIN is a small admittance in parallel with the device junctions, △f is the noise frequency bandwidth in Hz and λ is the light wavelength in nm. Other symbols and node names are defined in the previous slides.

Compact Device Modelling 1: Basic npn transistor model

Equation defined device

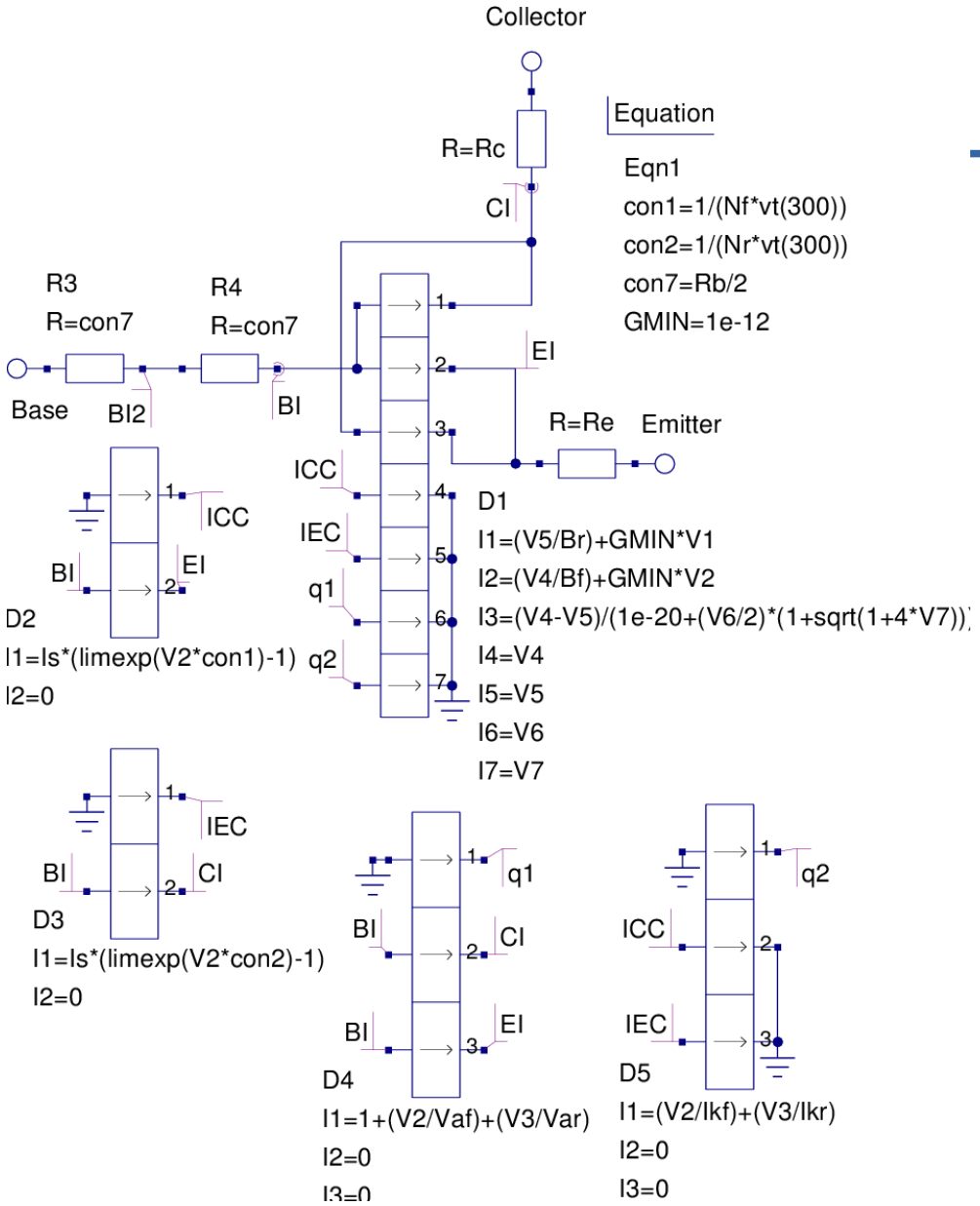


$$I = I(V), g = \frac{dI}{dV}$$

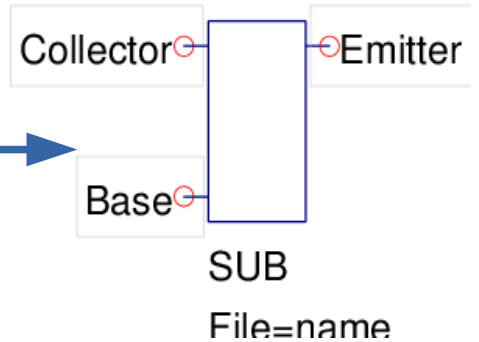
$$Q = Q(I, V)$$

$$C = \frac{dQ}{dV} = \frac{dQ(V)}{dV} + \frac{dQ(I)}{dI} \cdot g$$

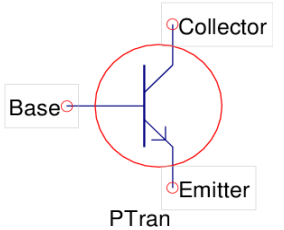
Large signal DC model



Generated symbol

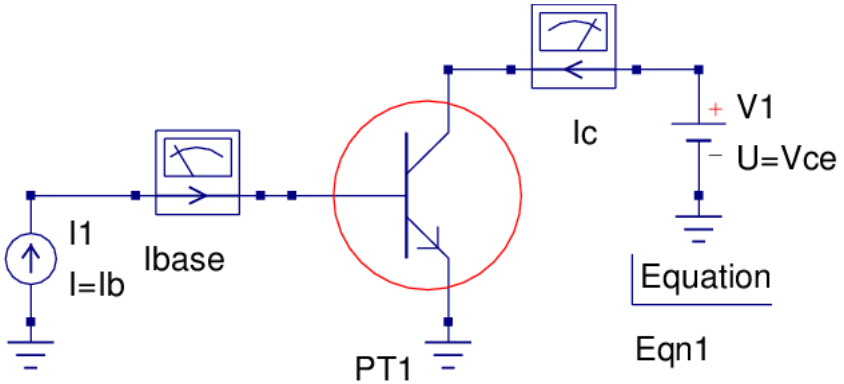


Redrawn symbol



PTran
File=name
Bf=100
Br=0.1
Is=1e-10
Nf=1
Nr=1
Var=100
Vaf=100
Rc=2
Rb=100
Re=1
Ikf=0.5
Ikr=0.5

Compact Device Modelling 1: DC simulation test



Equation

Eqn1

Beta=Ic.I/Ibase.I

PLBeta=PlotVs(Beta, Ic.I)

dc simulation

DC1

PT1

Bf=100

Br=0.1

Is=1e-10

Nf=1

Nr=1

Var=100

Vaf=100

Rc=2

Rb=100

Re=1

Ikf=0.5

Ikr=0.5

Parameter sweep

SW1

Sim=SW2

Type=log

Param=Ib

Start=1e-6

Stop=1m

Points=16

Parameter sweep

SW2

Sim=DC1

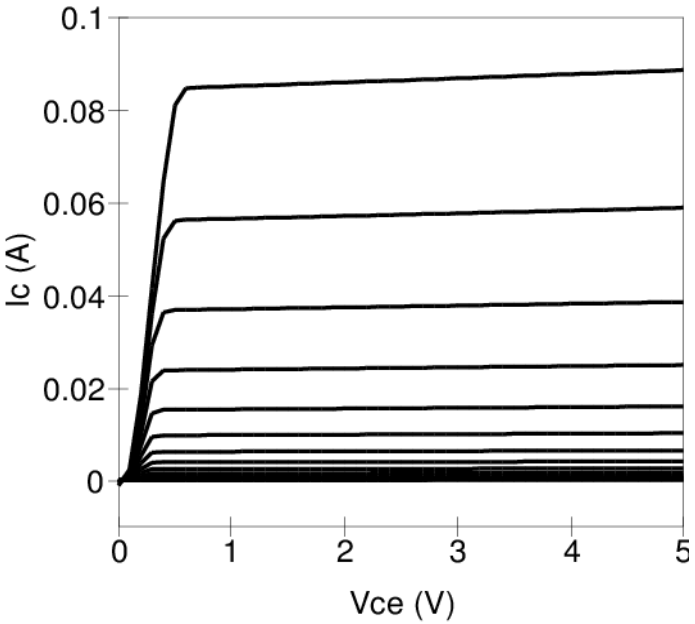
Type=lin

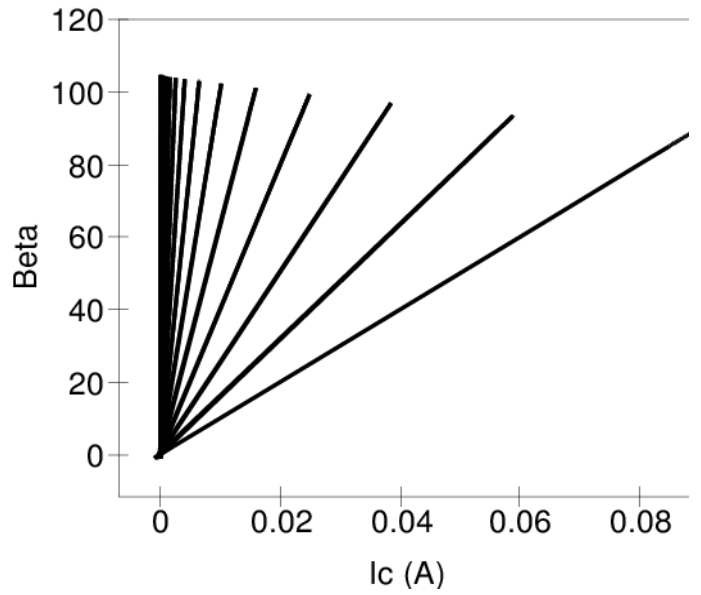
Param=Vce

Start=0

Stop=5

Points=51





Compact Device Modelling 1: Adding capacitance to the phototransistor model

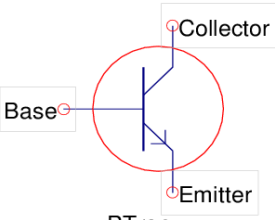
Charge equations

$$Q(BI,CI) = \int_0^{V(BI,CI)} CBC \cdot dV = Tr \cdot IEC + 2^{Mjc} \cdot Cjc \cdot \left[Mjc \cdot \frac{V(BI,CI)^2}{Vjc} + (1 - Mjc) \cdot V(BI,CI) \right] \quad \forall\ V(BI,CI) > \frac{Vjc}{2}$$

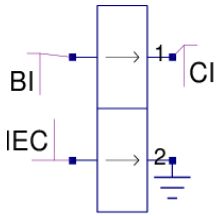
$$= Tr \cdot IEC + \frac{Cjc \cdot Vjc}{1 - Mjc} \left[1 - \left\{ 1 - \frac{V(BI,CI)}{Vjc} \right\}^{1-Mjc} \right] \quad \forall\ V(BI,CI) <= \frac{Vjc}{2}$$

$$Q(BI,EI) = \int_0^{V(BI,EI)} CBE \cdot dV = Tf \cdot ICC + 2^{MjE} \cdot Cje \cdot \left[Mje \cdot \frac{V(BI,EI)^2}{Vje} + (1 - Mje) \cdot V(BI,EI) \right] \quad \forall\ V(BI,EI) > \frac{Vje}{2}$$

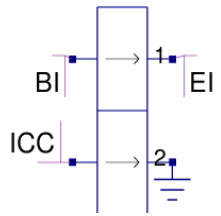
$$= Tf \cdot ICC + \frac{Cje \cdot Vje}{1 - Mje} \left[1 - \left\{ 1 - \frac{V(BI,EI)}{Vje} \right\}^{1-Mje} \right] \quad \forall\ V(BI,EI) <= \frac{Vje}{2}$$



EDD blocks



D6
I1=0
Q1=(V1 >= Vjc/2) ? Tr*V2+Cjc*con6*(Mjc*V1*V1/Vjc+con4*V1) : Tr*V2+Cjc*((Vjc/con4)*(1-(1-V1/Vjc)^con4))
I2=0

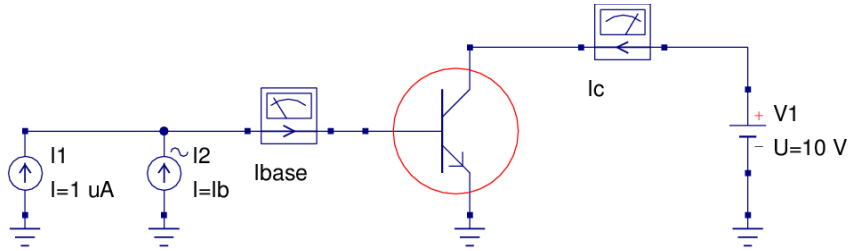


D7
I1=0
Q1=(V1>= Vje/2) ? Tf*V2+Cje*con5*(Mje*V1*V1/Vje+con3*V1) : Tf*V2+Cje*((Vje/con3)*(1-(1-V1/Vje)^con3))
I2=0

PTran
File=name
Bf=100
Br=0.1
Is=1e-10
Nf=1
Nr=1
Var=100
Vaf=100
Rc=2
Rb=100
Re=1
Ikf=0.5
Ikr=0.5
Mje=0.33
Vje=0.75
Cje=1p
Mjc=0.33
Vjc=0.75
Cjc=2p
Tr=100n
Tf=0.1n

Compact Device Modelling 1: Simulating capacitance effects

AC gain



Parameter sweep

SW1
Sim=AC1
Type=list
Param=Ib
Values=[50u; 200u; 500u]

dc simulation

DC1

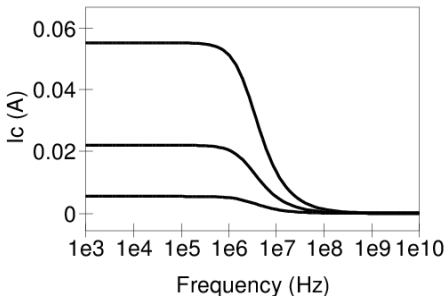
Equation

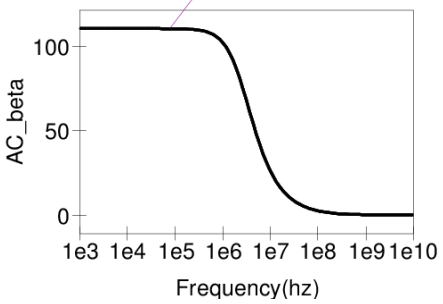
Eqn1
AC_beta=Ic.i/Ibase.i

ac simulation

AC1
Type=log
Start=1 kHz
Stop=10 GHz
Points=101

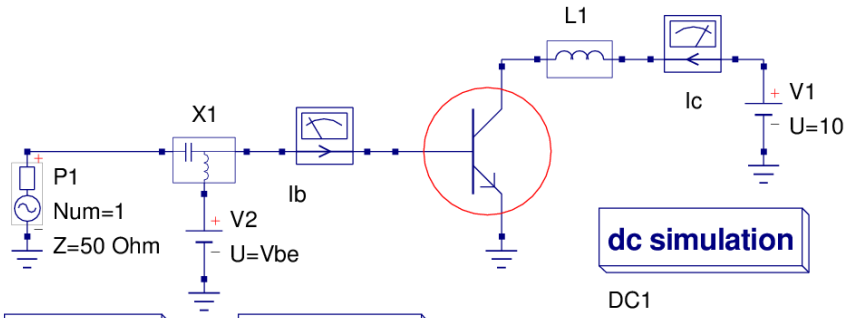
acfrequency: 7.76e+04
Ib: 5e-05
AC_beta: 110-j3.44





PTran1
Bf=100
Br=0.1
Is=1e-10
Nf=1
Nr=1
Var=100
Vaf=100
Rc=2
Rb=100
Re=1
Ikf=0.5
Ikr=0.5
Mje=0.33
Vje=0.75
Cje=1p
Mjc=0.33
Vjc=0.75
Cjc=2p
Tr=100n
Tf=0.1n

Cbe and Rbe extraction



dc simulation

DC1

Parameter sweep

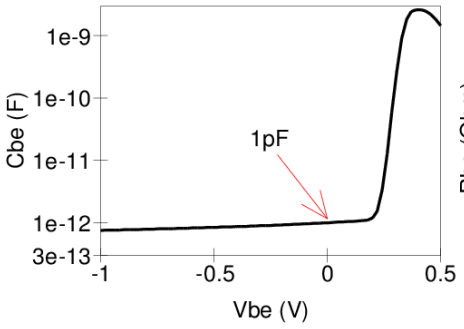
SW1
Sim=SP1
Type=lin
Param=Vbe
Start=-1
Stop=0.5
Points=71

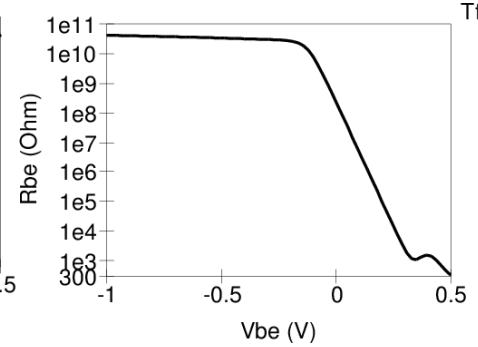
S parameter simulation

SP1
Type=const
Values=[100kHz]

Equation

Eqn1
y=stoy(S)
PL_y11=PlotVs(y[1,1], Vbe)
Cbe=imag(y[1,1])/Omega
Rbe=1/real(y[1,1])
PL_Cbe=PlotVs(Cbe, Vbe)
PL_Rbe=PlotVs(Rbe, Vbe)
Omega=2*pi*frequency

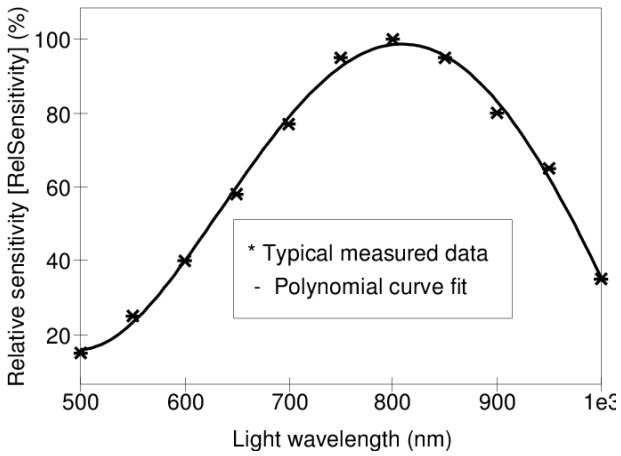




PTran1
Bf=100
Br=0.1
Is=1e-10
Nf=1
Nr=1
Var=100
Vaf=100
Rc=2
Rb=100
Re=1
Ikf=0.5
Ikr=0.5
Mje=0.33
Vje=0.75
Cje=1p
Mjc=0.33
Vjc=0.75
Cjc=2p
Tr=100n
Tf=0.1n

Compact Device Modelling 1: Adding photo-electric effects

Si photo-transistor relative sensitivity data

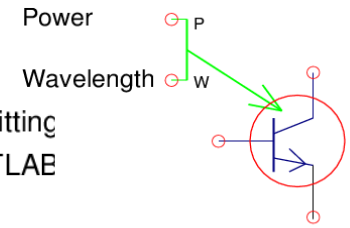


Measured data

Wavelength (nm)	Relative reponsivity (%)
500	15
550	25
600	40
650	58
700	77
750	95
800	100
850	95
900	80
950	65
1000	35

Curve fitting program

```
## Simple phototransistor responsivity fitting
## program - works with Octave or MATLAB
data = load('data.dat');
Ldata = data(:,1);
Sdata = data(:,2);
## Fit polynomial
Order = 4;
p = polyfit(Ldata, Sdata, Order);
## Evaluate the fitted polynomial
x=linspace(min(Ldata), max(Ldata), 101);
y = polyval(p,x);
## Plot
plot(x,y, '-', Ldata, Sdata,'x');
legend('Fitted polynomial', 'Original data');
```

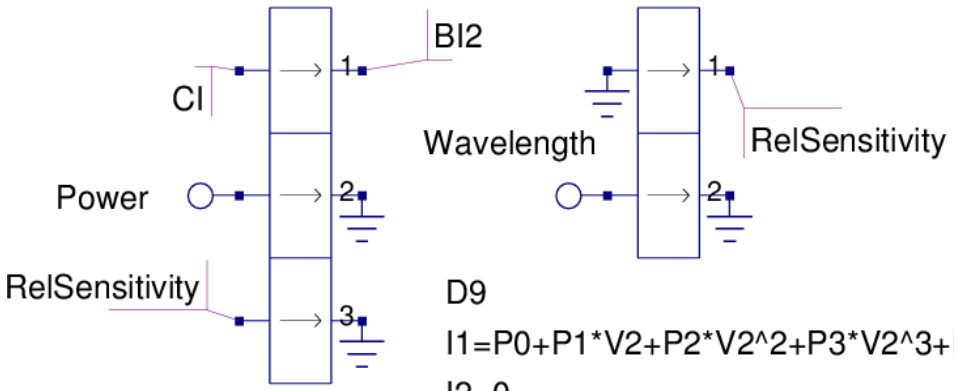



```
Ptran1
Bf=100
Br=0.1
Is=1e-10
Nf=1
Nr=1
Var=100
Vaf=100
Mje=0.33
Vje=0.75
Cje=1p
Mjc=0.33
Vjc=0.75
Cjc=2p
Tr=100n
Tf=0.1n
Ikf=0.5
Ikr=0.5
Rc=2
Re=1
Rb=100
Kf=1e-12
Ffe=1.0
Af=1.0
Responsivity=1.5
P0=2.6122e3
P1=-1.4893e1
P2=3.0332e-2
P3=-2.5708e-5
P4=7.6923e-9
```

$$RelSensitivity = P_0 + P_1 \cdot \lambda + P_2 \cdot \lambda^2 + P_3 \cdot \lambda^3 + P_4 \cdot \lambda^4$$

$$Where\ P_0 = 2.6122e3, P_1 = -1.4893e1, P_2 = 3.0332e\text{-}2, P_3 = -2.5708e\text{-}5, P_4 = 7.69e\text{-}9$$

EDD light bus model



D9

I1=P0+P1*V2+P2*V2^2+P3*V2^3+P4*V2^4

I2=0

D8

I1=((Responsivity*V3)/(Bf*100))*V2

I2=0

I3=V3

Compact Device Modelling 1: Qucs Radio Frequency Defined Device (RFDD)

- Qucs RFEDD is a frequency dependent equation defined device based on two-port and multi-port electrical network structures.
- Two-port RFEDD are similar in structure to the well-known two-port networks commonly employed in the linear analysis of amplifiers and filters.
- Multi-port RFEDD allow, for example, modelling of directional couplers, isolators and other multi-port high frequency devices.
- Admittance (Y), impedance (Z), scattering (S), hybrid (H), inverse hybrid (G), transmission (A) and scattering transmission (T) two-port devices are implemented
  In contrast to the two-port device, the multi-port RFEDD can only be defined for Y, Z or S Networks.
- Qucs RFEDD parameters are numerical or algebraic complex numbers. The latter can be functions of numerical constants, variables from Qucs equation blocks, subcircuit Parameters, mathematical operators and functions defined in the Verilog-A HDL and the Qucs function library.
- The real and imaginary parts of RFEDD parameters may also be functions of small signal simulation frequency (F) or operator S ($S = j \, . \, \omega = j \, . \, 2 \, . \pi \, . \, F$).

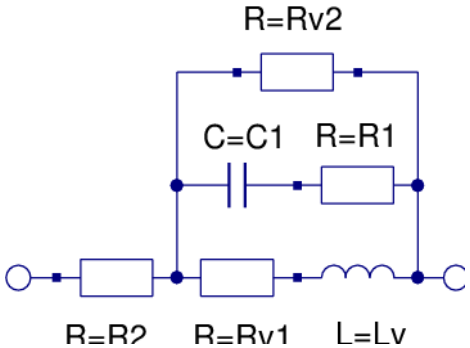
Compact Device Modelling 1: RFDD high frequency inductance

Equations and model

$$Rv1 = K1 \cdot \sqrt{F}, \quad Rv2 = K2 \cdot \sqrt{F}$$

$$Lv = [K3 - K4 \cdot \ln(K5 \cdot F)] \cdot 1\text{e-}6$$

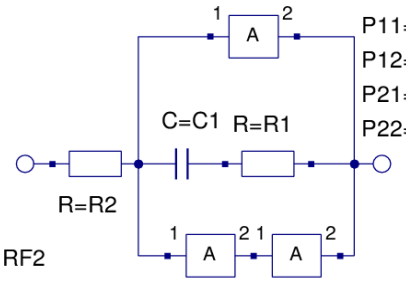
Where K1, K2, K3, K4 and K5 are constants



Qucs RFEDD model

HF Coil 6

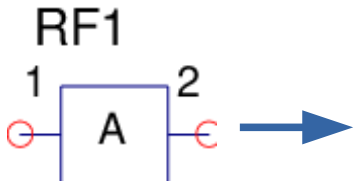
COIL1
R1=6200
R2=1.59
K1=1.49e-5
K2=0.268
K3=1.19
K4=1.4e-2
K5=6.9e-6
C1=0.139e-12



RF1
P11=1
P12=K2*sqrt(F)
P21=0
P22=1

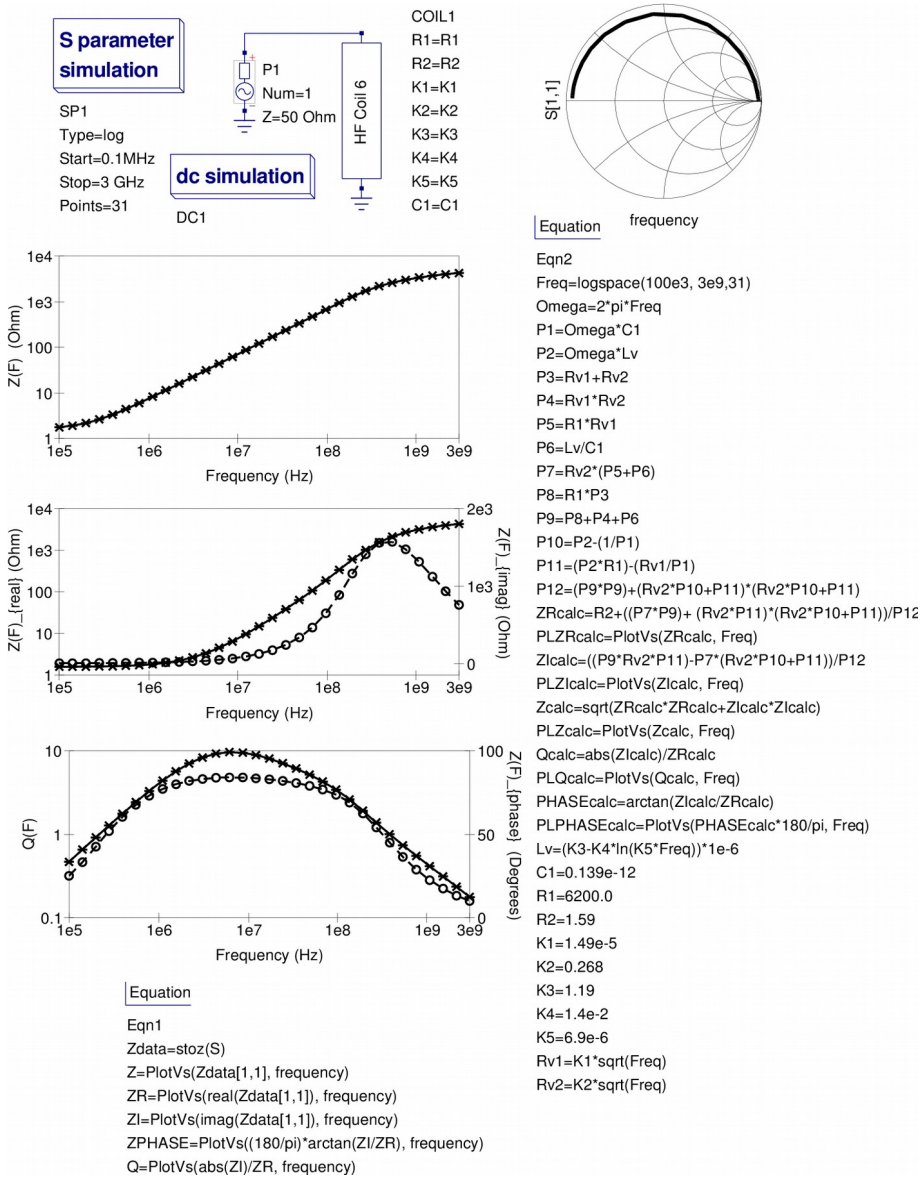
RF2
P11=1
P12=K1*sqrt(F)
P21=0
P22=1

RF3
P11=1
P12=j*2*pi*F*(K3-K4*ln(K5*F))*1e-6
P21=0
P22=1



$$[A] = \begin{bmatrix} P11 & P12 \\ P21 & P22 \end{bmatrix} = \begin{bmatrix} 1 & Z(real) + j \cdot Z(imag) \\ 0 & 1 \end{bmatrix}$$

Simulation results



Equation

Eqn1
Zdata=stoz(S)
Z=PlotVs(Zdata[1,1], frequency)
ZR=PlotVs(real(Zdata[1,1]), frequency)
ZI=PlotVs(imag(Zdata[1,1]), frequency)
ZPHASE=PlotVs((180/pi)*arctan(ZI/ZR), frequency)
Q=PlotVs(abs(ZI)/ZR, frequency)

Equation

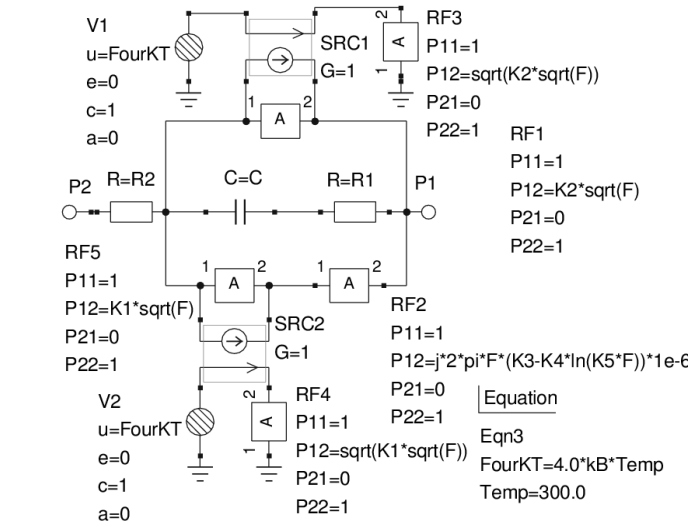
frequency

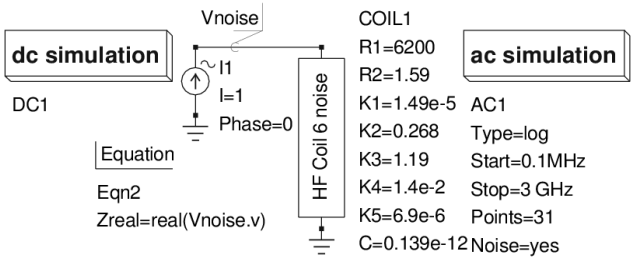
Eqn2
Freq=logspace(100e3, 3e9,31)
Omega=2*pi*Freq
P1=Omega*C1
P2=Omega*Lv
P3=Rv1+Rv2
P4=Rv1*Rv2
P5=R1*Rv1
P6=Lv/C1
P7=Rv2*(P5+P6)
P8=R1*P3
P9=P8+P4+P6
P10=P2-(1/P1)
P11=(P2*R1)-(Rv1/P1)
P12=(P9*P9)+(Rv2*P10+P11)*(Rv2*P10+P11)
ZRcalc=R2+((P7*P9)+ (Rv2*P11)*(Rv2*P10+P11))/P12
PLZRcalc=PlotVs(ZRcalc, Freq)
ZIcalc=((P9*Rv2*P11)-P7*(Rv2*P10+P11))/P12
PLZIcalc=PlotVs(ZIcalc, Freq)
Zcalc=sqrt(ZRcalc*ZRcalc+ZIcalc*ZIcalc)
PLZcalc=PlotVs(Zcalc, Freq)
Qcalc=abs(ZIcalc)/ZRcalc
PLQcalc=PlotVs(Qcalc, Freq)
PHASEcalc=arctan(ZIcalc/ZRcalc)
PLPHASEcalc=PlotVs(PHASEcalc*180/pi, Freq)
Lv=(K3-K4*ln(K5*Freq))*1e-6
C1=0.139e-12
R1=6200.0
R2=1.59
K1=1.49e-5
K2=0.268
K3=1.19
K4=1.4e-2
K5=6.9e-6
Rv1=K1*sqrt(Freq)
Rv2=K2*sqrt(Freq)

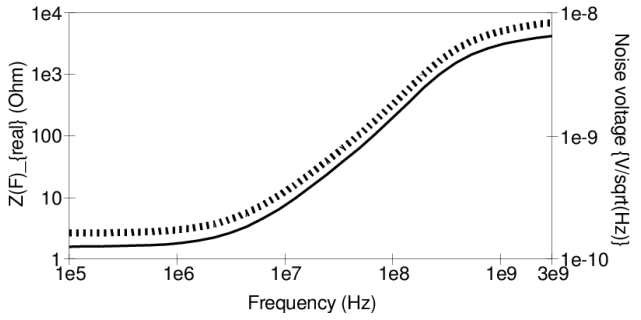
M.E. Brinson and S, Jahn, **Modelling of high-frequency inductance with Qucs non-linear radio frequency defined devices**, International Journal of Electronics, in press, 2009, 96:, 307-321

Compact Device Modelling 1: RFDD high frequency skin effect noise

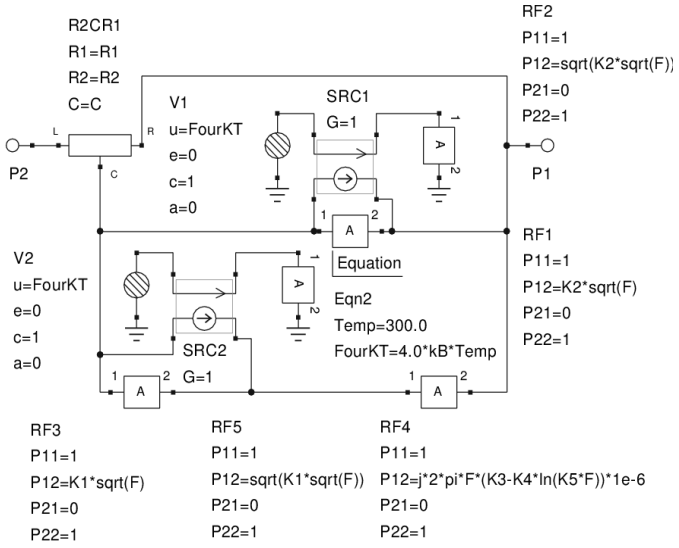
Noise model: standard plus RFEDD non-linear components

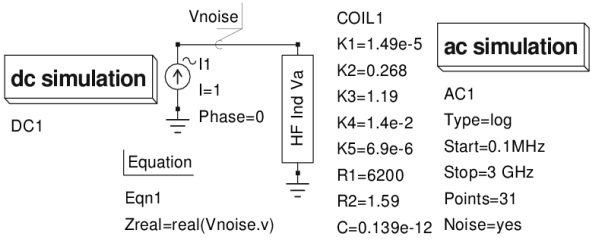


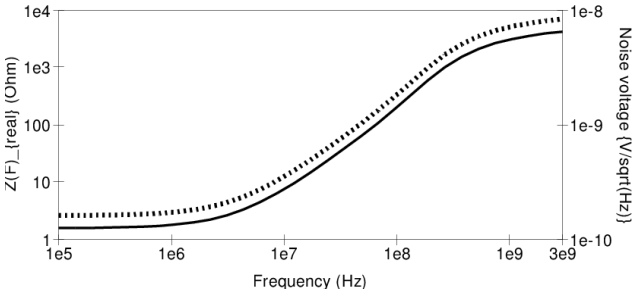




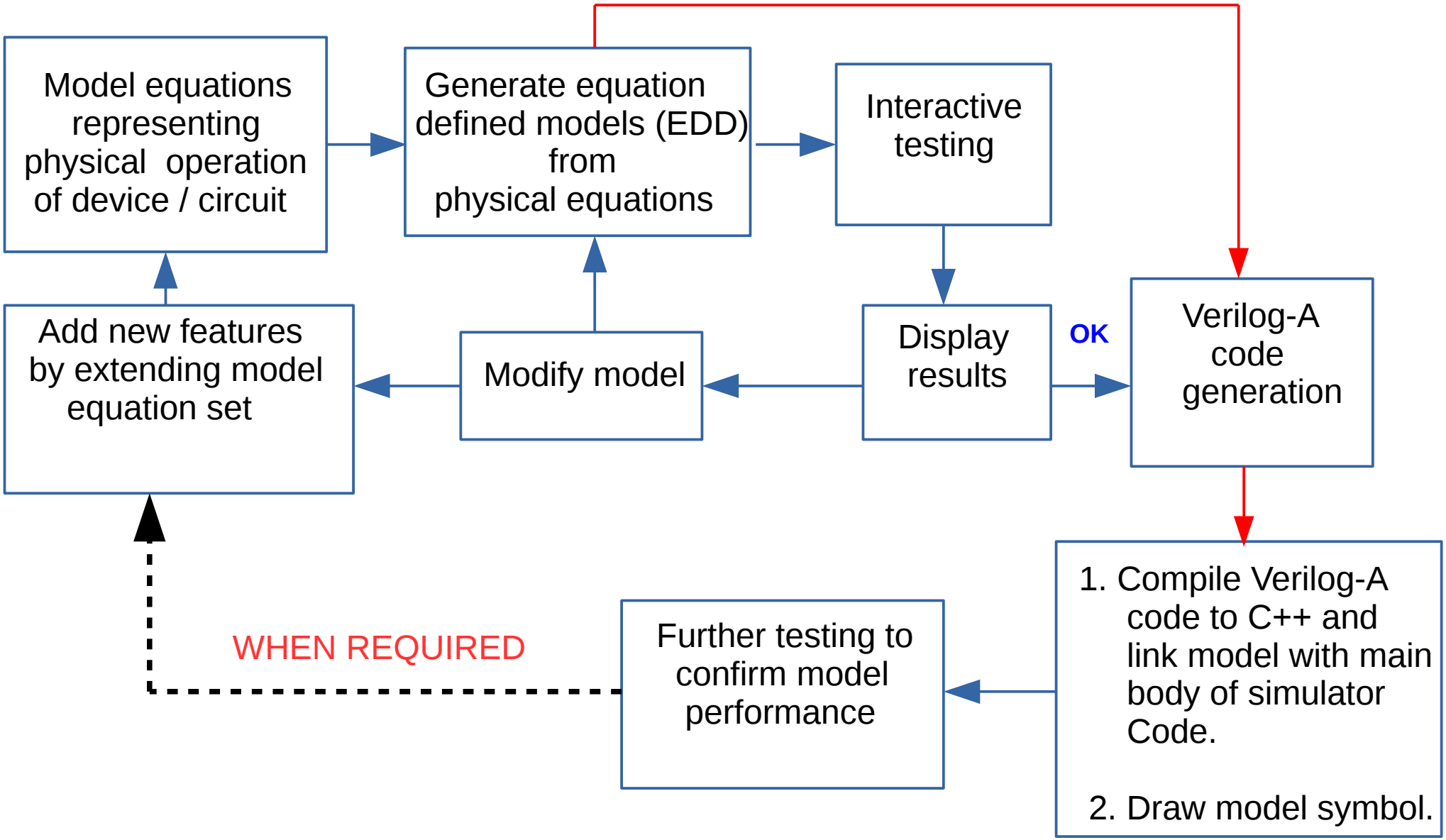
Noise model: Verilog-A block plus RFEDD non-linear components





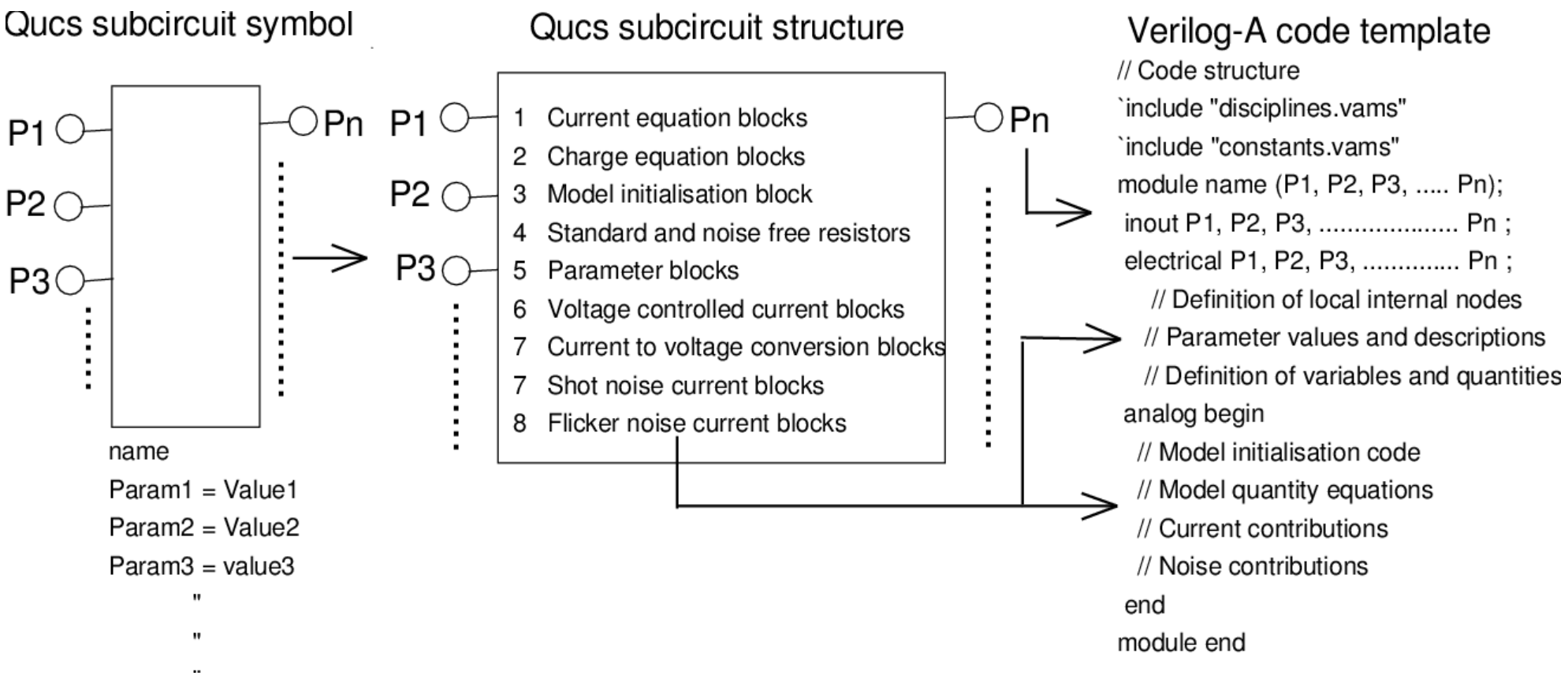


Compact Device Modelling 3: Verilog-A model generation



Compact Device Modelling 3: Route for generating Verilog-A compact device models

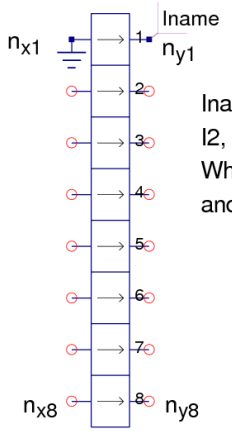
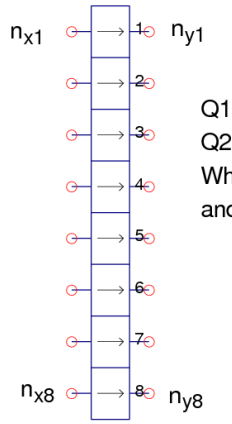
The following diagram illustrates the initial stage in the construction of a Qucs Verilog-A compact device model

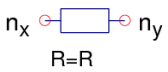
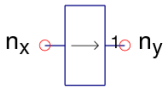
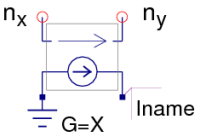
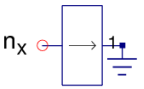


Compact Device Modelling 3: Qucs EDD to Verilog-A code

Relationships between Qucs schematic symbols and Verilog-A code fragments

Fundamental EDD blocks

Qucs symbol	Quantity equations	Verilog-A code fragment
	Iname = I1 = f(V2, V3, ... V8) I2, I3, ... I8 = 0 and Q1, Q2, ... Q8 = 0. Where Vm = V(n_{xm}, n_{ym}) or Vm = V(n_{xm}), and 2 ≤ m ≤ 8.	Iname = f(V2, V3, ... V8); Or Iname <+ f(V2, V3, ... V8);
	Q1 = f(V1, V2, ... V8, I1, I2, ... I8) Q2, ... Q8 = 0. Where Vm = V(n_{xm}, n_{ym}) or Vm = V(n_{xm}), and 1 ≤ m ≤ 8.	I(n_{x1}, n_{y1}) <+ ddt(Q1); Or I(n_{x1}, n_{y1}) = ddt(Q1);

	Quantity equations	Verlog-A code fragment
(a) Model initialisation block	Equation Eqn1 con1= ..... con2= ..... con3=	@(initial_model) begin con1 = ...; con2 = ...; con3 = ...; end
(b) Standard resistors		I(n_x, n_y) <+ V(n_x, n_y)/R; I(n_x, n_y) <+ white_noise((FourKT/R, "thermal"); Where FourKT = 4.0 · `P_K · $temperature, and `P_K = 1.3806505e-23 K^{-1}, $temperature is the resistor temperature in Kelvin.
(c) Noise free resistors	 I1 = V1/R = V(n_x, n_y)/R	I(n_x, n_y) <+ V(n_x, n_y)/R;
(d) Voltage controlled current block	 Iname = X · V(n_x, n_y)	Iname = X*V(n_x, n_y);
(e) current to voltage conversion block	 Iname = I1 = V1	Iname = V(n_x);

Compact Device Modelling 3: The Qucs/“Analog Device Model Synthesiser” Verilog-A subset

The “Analog Device Model Synthesiser” (ADMS) software is supplied with little documentation!
These brief notes provide a basic introduction to the Qucs/ADMS Verilog-A subset

- Verilog-A is a case sensitive language.
- Comments: (1) single line comments start with ***//***,
  (2) block comments begin with ***/**** and end with ****/***
- Identifiers are sequences of letters, digits, dollar signs '**$**' and the underscore '_'; the first letter of an identifier must not be a digit.
- Qucs/ADMS version 2.34 keywords: **parameter, aliasparameter, aliasparam, module, endmodule, function, endfunction, discipline, potential, flow, domain, ground, enddiscipline, nature, endnature, input, output, inout, branch, analog, begin, end, if, while, case, endcase, default, for, else, integer,real, string, from, exclude, inf, INF.**
- Compiler directives: **`define,`undef, `ifdef, `else, `endif, `include.**
- Data types: **integer**s, **real**s and **string**s.
- Predefined constants in “constants.vams”: **`M_PI, `M_TWO_PI, `M_PI_2, `M_PI_4, `M_1_PI, `M_2_PI, `M_2_SQRTPI, `M_E, `M_LOG2E, `M_LOG10E, `M_LN2, `M_LN10, `M_SQRT2, `M_SQRT1_2, `P_Q, `P_C, `P_K, `P_H,`P_EPS0, `P_U0, `P_CELSIUS0.**
- Variables are named objects that contain a value of a particular type. They are initialised to zero or uknown. They retain their value until changed by an assignment statement.

Compact Device Modelling 3: The Qucs/“Analog Device Model Synthesiser” Verilog-A subset continued

- Parameters are declared with statements of the form:
  parameter integer size=16; parameter real period = 1.0 from (0:inf);
  parameter integer dir = 1 from [-1:1] exclude 0;
- Verilog-A natures and disciplines ae listed in file “disciplines.vams”
- Port, net and node examples in Verilog-A:

```
module amp(out1, in1);
  input in1;
  output out1;
  electrical out1, in1;
```

- Branches are declared with statement **branch (n1,n2) b1;**
- Signal access function examples: **V(n2), I(n), V(b1), I(b1), V(n,m), I(n,m)**
- Current contribution examples:

```
I(diode) <+ Is*(limexp(V(diode)/$vt)-1);
I(diode) <+ ddt(-2*cj0*phi*sqrt(1-V(diode)/phi));
```

- Qucs/ADMS allows an extensive range of Verilog-AMS operators and mathematical functions.
- Environmental Functions: **$temperature, $vt, $strobe, $finish, $given, $parameter_given.**
- Analogue operators: **@(initial_step), @(final_step), @(initial_model), @(initial_instance).**

Compact Device Modelling 3: The Qucs/"Analog Device Model Synthesiser" Verilog-A subset continued

Analogue behavioural statements:

1. Analog process/procedural block;

```
analog begin
    I(diode) <+ Is*(limexp(V(diode)/$vt)-1);
    qd       =  tf*I(diode) -2*cj0*phi*sqrt(1-V(diode)/phi);
   I(diode) <+ ddt(qd);
end
```

2. Assignment statements consist of a variable followed by = and an expression
3. Conditional operator cond ? Val1 : Val2, for example

```
State = (V(d) > 0.0) ? 1 : -1;
```

TRUE = non-zero value

4. if-else statement:

```
If (V(ps,ns) > thresh)
    I(p,n) <+ 1;
else
    I(p,n) <+ 0;
```

5. Case statement:

```
case (select)
  0 : out = I(in0);
  1 : out = I(in1);
  2 : out = I(in2);
  default : out = 0;
endcase
```

6. While statement:

```
test = 4;
While( test) begin
  A = A+1;
  B = B-1;
  test = test-1;
end
```

7. For loops:

```
for (i=0; I <10; i=i+1) begin
                ..
end
```

8. User defined functions and function calls:

More details of Qucs/ADMS Verilog-A statement covourage is given in:
http://www.mos-ak.org/london_2014/presentations/09_Mike_Brinson_MOS-AK_London_2014.pdf

Compact Device Modelling 3: The Qucs/“Analog Device Model Synthesiser” Verilog-A subset; special features and problems

- Although Verilog-A is standardised, published versions of ADMS implement different subsets of verilog-A.
- Simulator interface facilities may vary: for example **$strobe** is replaced/added to by **$error** and **$warning** in QucsStudio.
- The inclusion mechanism for Verilog-A predefined constants, natures and discipines files can change from simulator to simulator: for example use of **`include constants.h** and **`include disciplines.h** by SymicaDE.
- Parameter statement descriptive elements vary significantly amoung circuit simulator implementations: for example Qucs and QucsStudio use **`attr(info = “description”)**, or **`attr(desc=”description”)** and **unit=”value”** other simulators **use`P(.....)**.
- Verilog-A compact modelling extensions: special features taylored to compact device modelling have been added to the MOT-ADMS 2.30 software, including **desc, unit, $parameter_given, $given**, **aliasparameter** and **string** type in **parameters**. In addition the function **ddx** is now adopted in the language standard, where **ddx** takes a variable and a node potential (for example **gm = ddx(Ids,V(g));**) and returns the symbolic partial derivative of the variable with respect to the node potential, holding all other potentials constant. Qucs and QucsStudio implement these extensions but other circuit simulators may not.
- Each circuit simulator uses one or more backend XML script files for C++ code generation and model interfacing to the main body of C++ code: as these XML scripts are specific to each simulator API there are likely to be differences in the C++ model generated. One example being electrical noise: Verilog-A statements **white_noise(pwr, <name>), flicker_noise(pwr, <exp>,<name>)** are implemented in Qucs and QucsStudio but not, for example, in ngspice.
- In addition to the above are the results from new types of circuit simulation, like for example
- Harmonic Balance, correct or not?

Compact Device Modelling 3: Building a Qucs EDD/Verilog-A model; part 1 – DC characteristics

EPFL-EKV v2.6 long channel nMOS equations

$$Vg = V(gate) - V(bulk),\quad Vs = V(source) - V(bulk),\quad Vd = V(drain) - V(bulk)$$

$$VGprime = Vg - VTO + PHI + GAMMA \cdot \sqrt{PHI}$$

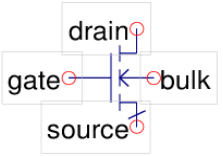
$$VP = VGprime - PHI - GAMMA \cdot \left[\sqrt{VGprime + \left(\frac{GAMMA}{2}\right)^2} - \frac{GAMMA}{2} \right]$$

$$n = 1 + \frac{GAMMA}{2} \cdot \sqrt{VP + PHI + 4 \cdot Vt},\quad BETA = KP \cdot \frac{W}{L} \cdot \frac{1}{1 + THETA \cdot VP}$$

$$X1 = \frac{Vp - Vs}{Vt},\quad If = \left[\ln(1 + limexp(X1/2))\right]^2,\quad X2 = \frac{Vp - Vd}{Vt},\quad Ir = \left[\ln(1 + limexp(X2/2))\right]^2$$

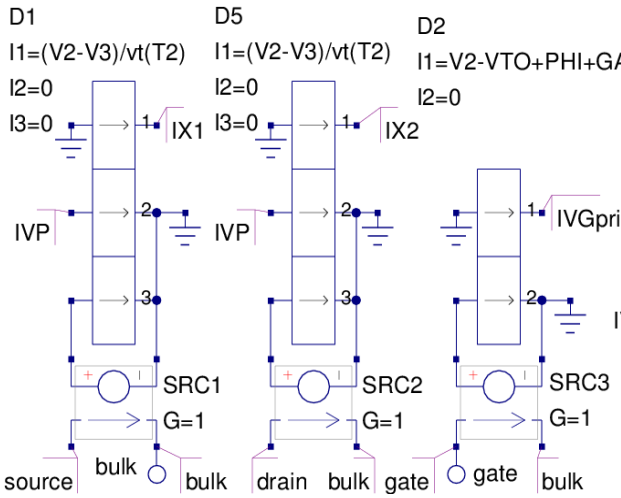
$$Ispecific = 2 \cdot n \cdot BETA \cdot vt^2,\quad Ids = Ispecific \cdot (If - Ir)$$

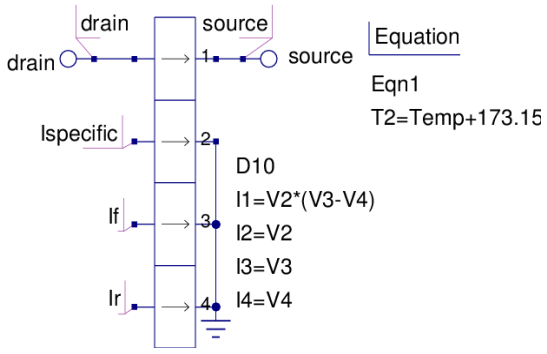
Symbol

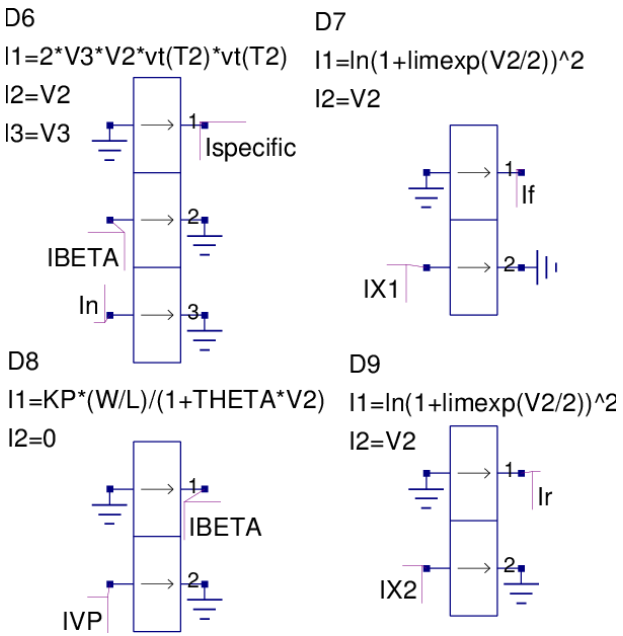


EKVLC
File=name
L=10e-6
W=10e-6
VTO=0.5
GAMMA=0.7
PHI=0.5
KP=50e-6
THETA=50e-3
Temp=26.85

Model







Compact Device Modelling 3: Building a Qucs EDD/Verilog-A model; part 2 – intrinsic charge characteristics

EPFL-EKV v2.6 long channel nMOS equations

$$nq = 1 + \frac{GAMMA}{2 \cdot \sqrt{VP + PHI + 1e\text{-}6}}$$

$$Xf = \sqrt{0.25 + If}\,, \quad Xr = \sqrt{0.25 + Ir}$$

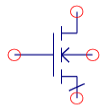
$$qI = -nq \cdot \left[\frac{\frac{4}{3} \cdot (Xf^2 + Xf \cdot Xr + Xr^2)}{Xf + Xr} - 1 \right]$$

$$qB = \frac{-GAMMA \cdot \sqrt{VP + PHI + 1e\text{-}6} \cdot 1}{vt} - \left[\frac{nq - 1}{nq} \right], when \;\; VGprime > 0$$

$$qB = \frac{-VGprime}{vt}, when \;\; VGprime \le 0$$

$$qG = -qI - qB\,, \quad Cox = COX \cdot W \cdot L\,, \quad Q(I,B,D,S,G) = Cox \cdot vt \cdot q(I,B,D,S,G)$$

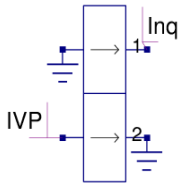
Symbol



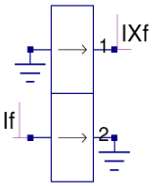
EKVLC1
L=10e-6
W=10e-6
VTO=0.5
GAMMA=0.7
PHI=0.5
KP=50e-6
THETA=50e-3
Temp=26.85
COX=3.45e-3
Xpart=0.6

Model

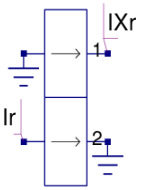
D11
I1=1+GAMMA/(2*sqrt(V2+PHI+1e-6))
I2=0

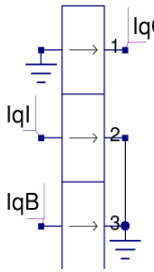


D12
I1=sqrt(0.25+V2)
I2=0



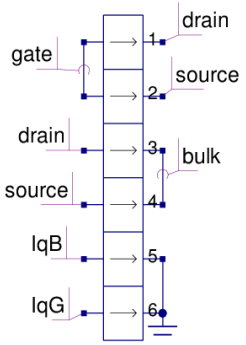
D13
I1=sqrt(0.25+V2)
I2=0





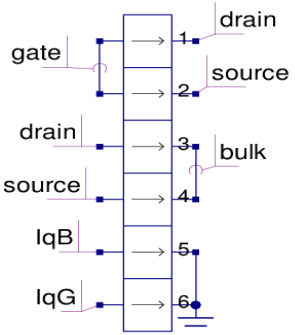
D14
I1=-(V2+V3)
I2=V2
I3=V3

Equation
Eqn1
Spart=1-Xpart
Qox=COX*W*L*vt(T2)



D17
I1=0
Q1=Xpart*V6*Qox
Q2=Spart*V6*Qox
Q3=Xpart*V5*Qox
Q4=Spart*V5*Qox
I5=0
I6=V6

Equation
Eqn1
Spart=1-Xpart
Qox=COX*W*L*vt(T2)



D17
I1=0
Q1=Xpart*V6*Qox
Q2=Spart*V6*Qox
Q3=Xpart*V5*Qox
Q4=Spart*V5*Qox
I5=0
I6=V6

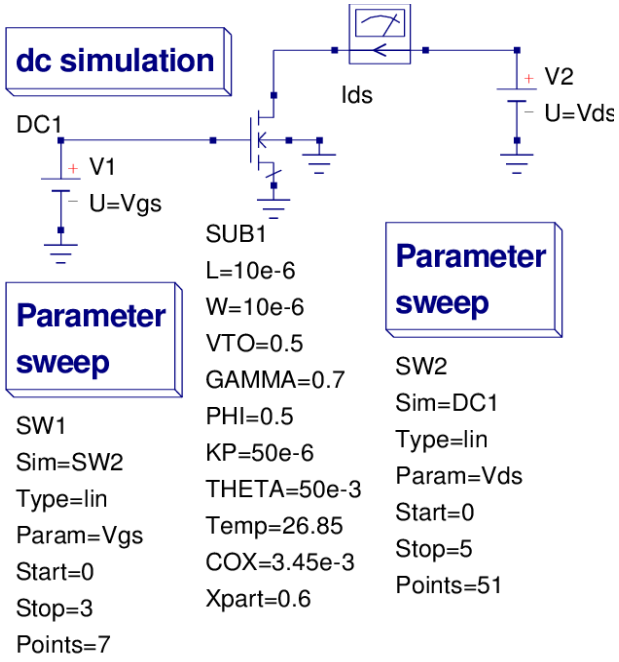
Compact Device Modelling 3: Building a Qucs EDD/Verilog-A model; part 3 – Verilog-A model code

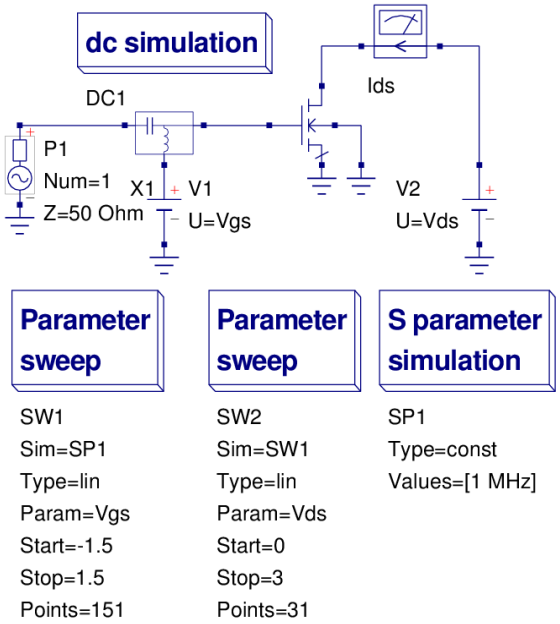
Verilog-A code derived from Qucs EDD model

```
`include “disciplines.vams”
`include “constants.vams”
module EKVLC(drain, gate, source, bulk);
inout drain, gate, source, bulk;
electrical drain, gate, source, bulk;
`define attr(txt) (*txt*)
parameter  real L=1e-6 from [1e-20 :inf] `attr(info=”Channel length”);
..
..
real VGprime,VP, n, BETA, X1, If, X2, Ir, Ispecific, Ids;
real nq, XF, Xr, ql, qB,qG;
analog begin
@(initial_model) begin
   Spart=1-Xpart; Qox=COX*W*L; T2=Temp+273.15; vt=`P_K*T2/`P_Q;
end
Vg=V(gate)-V(bullk); Vs=V(source)-V(bulk); Vd=V(drain)-V(bulk);
VGprime=Vg-VTO+PHI+GAMMA*sqrt(PHI);
VP=VGprime-PHI-GAMMA*(sqrt(VGprime+(GAMMA/2)^2)-GAMMA/2);
n=1+GAMMA/(2*sqrt(VP+PHI+4*vt);
BETA=KP*(W/L)*1/(1+THETA*VP);
X1=(VP-Vd)/vt; If=ln(1+limexp(X1/2))^2;
X2=(VP-Vs)/vt; Ir=ln(1+limexp(X2/2))^2;
Ispecific=2*n*BETA*vt*vt;
Ids=Ispecific*(If-Ir);
I(drain,source)  <+ Ids;
nq=1+GAMMA/(2*sqrt(VP+PHI+1e-6);
Xf=sqrt(0.25+If); Xr=sqrt(0.25+Ir)
ql=-nq*( (4/3)*(Xf^2+Xr*Xf+Xr^2)/(Xf+Xr)^2 -1);
if (VGprime > 0)
     qB=-GAMMA*sqrt(VP+PHI+1e-6);
else qB=-VGprime/vt;
qG=-ql-qB;
I(gate, drain)<+ddt(Xpart*qG*Qox;
I(gate, source)<+ddt(Spart*qG*Qox);
I(drain, bulk)<+ddt(Xpart*qB*Qox);
I(source, bulk)<+ddt(Spart*qB*Qox);
end
endmodule
```

Compact Device Modelling 3: Building a Qucs EDD/Verilog-A model; part 4 – model test circuits and typical simulation results

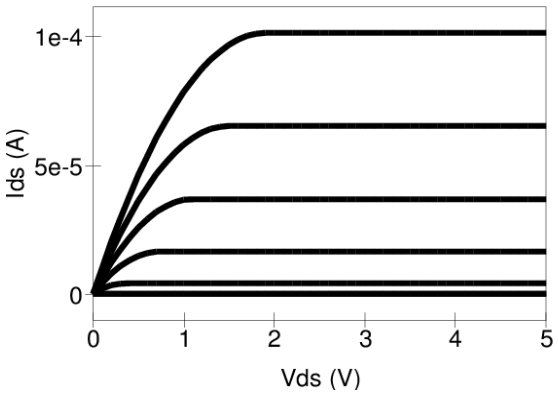
Interactive testing and parameter extraction

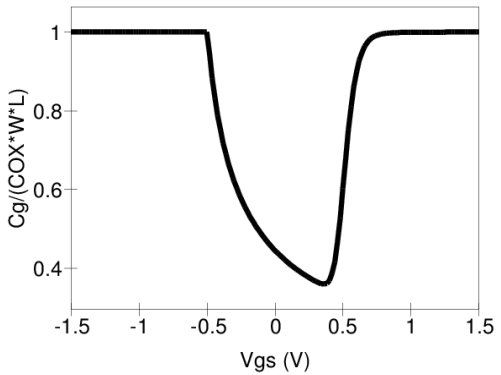


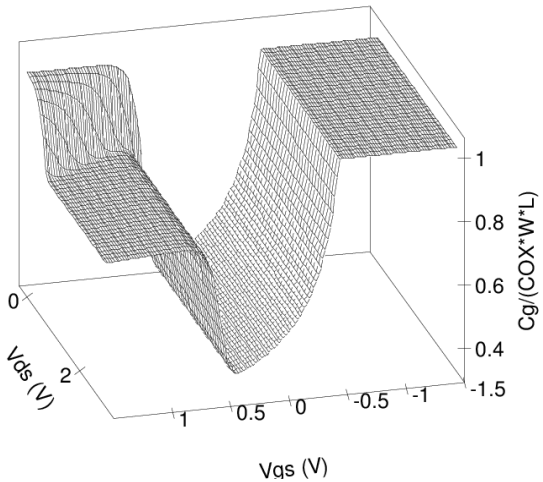


EKVLC1
L=10e-6
W=10e-6
VTO=0.5
GAMMA=0.7
PHI=0.5
KP=50e-6
THETA=50e-3
Temp=26.85
COX=3.45e-3
Xpart=0.6

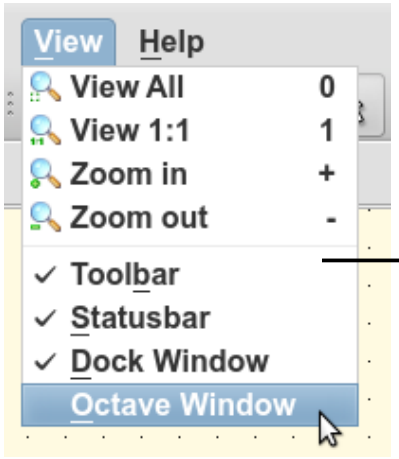
Equation
Eqn1
L=10e-6
W=10e-6
COX=3.45e-3
Cg_norm=COX*W*L
Omega=2*pi*frequency
Y=stoy(S)
Cg=imag(Y[1,1]/Omega)
Cg_2D_plot=PlotVs(Cg/Cg_norm,Vgs)
Cg_3D_plot=PlotVs(Cg/Cg_norm,Vds, Vgs)
Cg_a=PlotVs(Cg,Vgs)

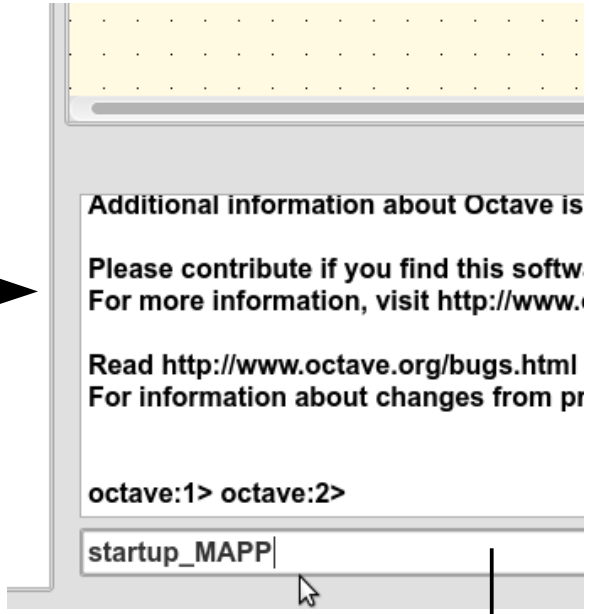






Compact device modelling: 3. Expanded compact device modelling capabilities with the Berkeley Model and Algorithm Prototyping Platform (MAPP); part 1

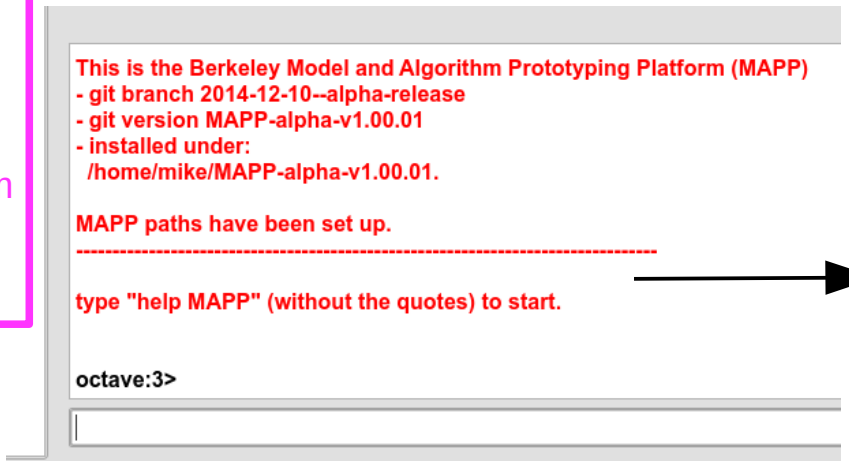




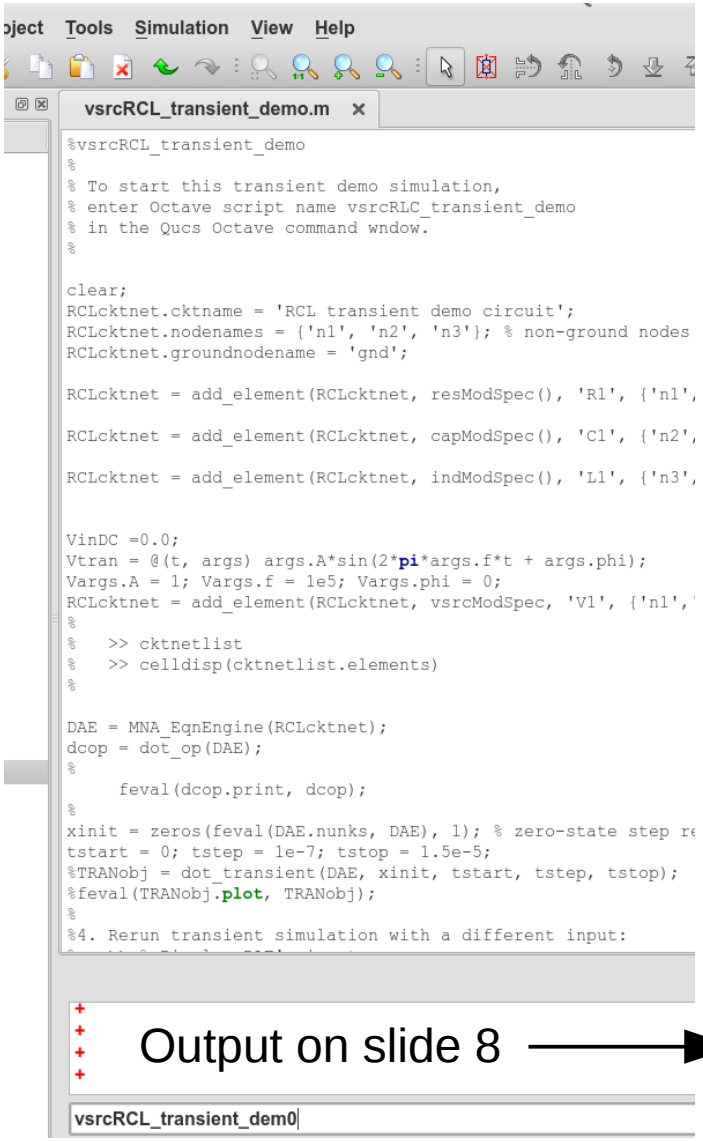
Qucs project:

MAPPExamples_prj

Startup_MAPP.m
vsrcRLC_transient_demo.m
..
..



MAPP code in Qucs edit window

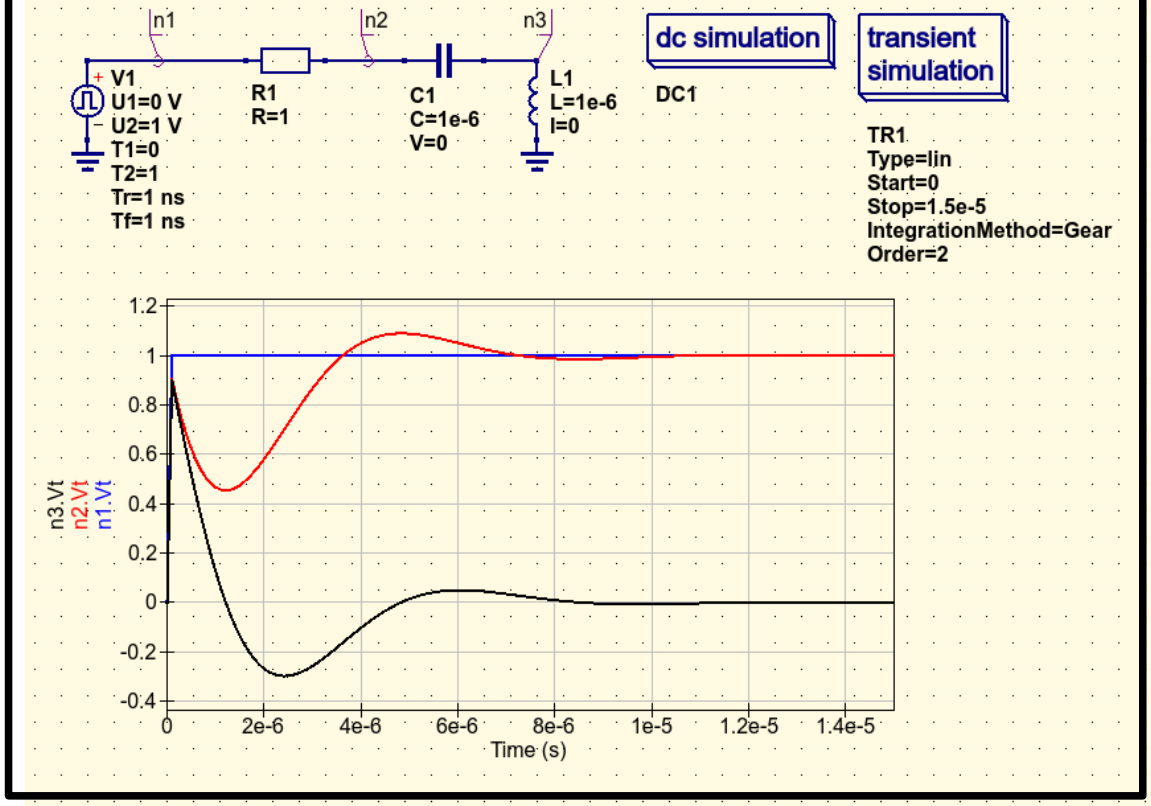


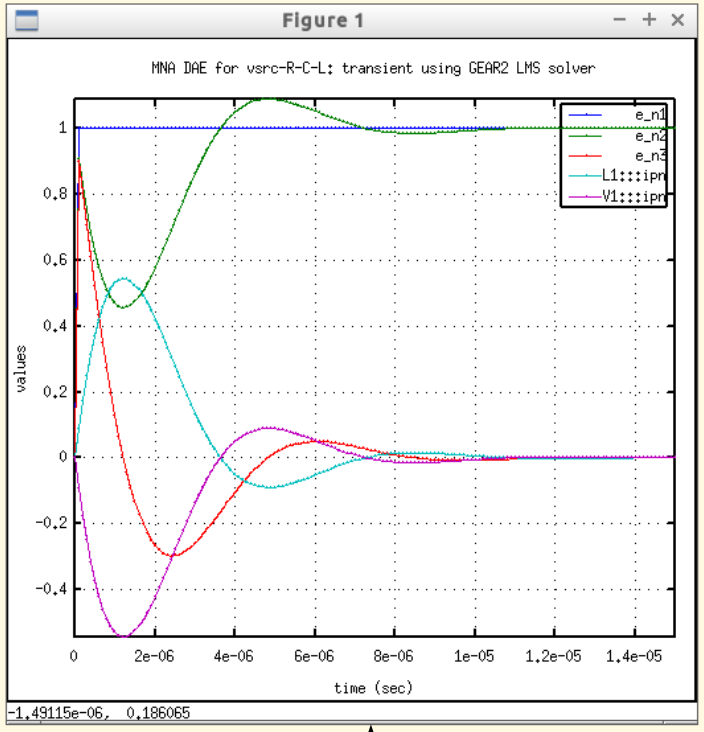
Background to MAPP:

http://www.mos-ak.org/berkeley_2014/presentations/07_Jaijeet_Roychowdhury_MOS-AK_Berkeley_2014.pdf
http://www.mos-ak.org/berkeley_2014/presentations/07_Tianshi_Wang_MOS-AK_Berkeley_2014.pdf

Compact device modelling: 3. Expanded compact device modelling capabilities with the Berkeley Model and Algorithm Prototyping Platform (MAPP); part 2

Qucs circuit schematic and output data





From slide 33

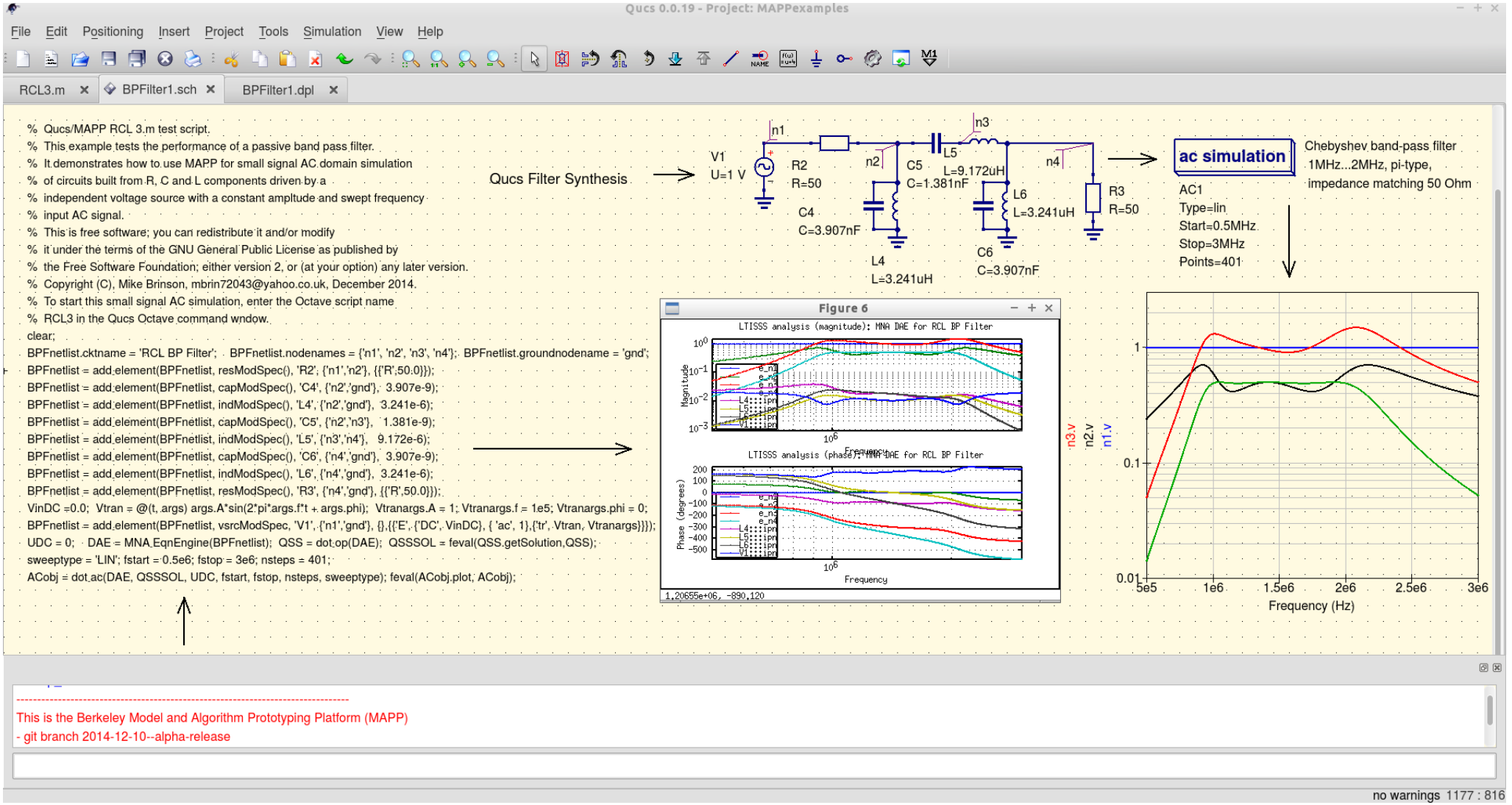
MAPP output data

Background to MAPP:

http://draco.eecs.berkeley.edu/dracotiki/tiki-index.php?page=MAPPfeatures

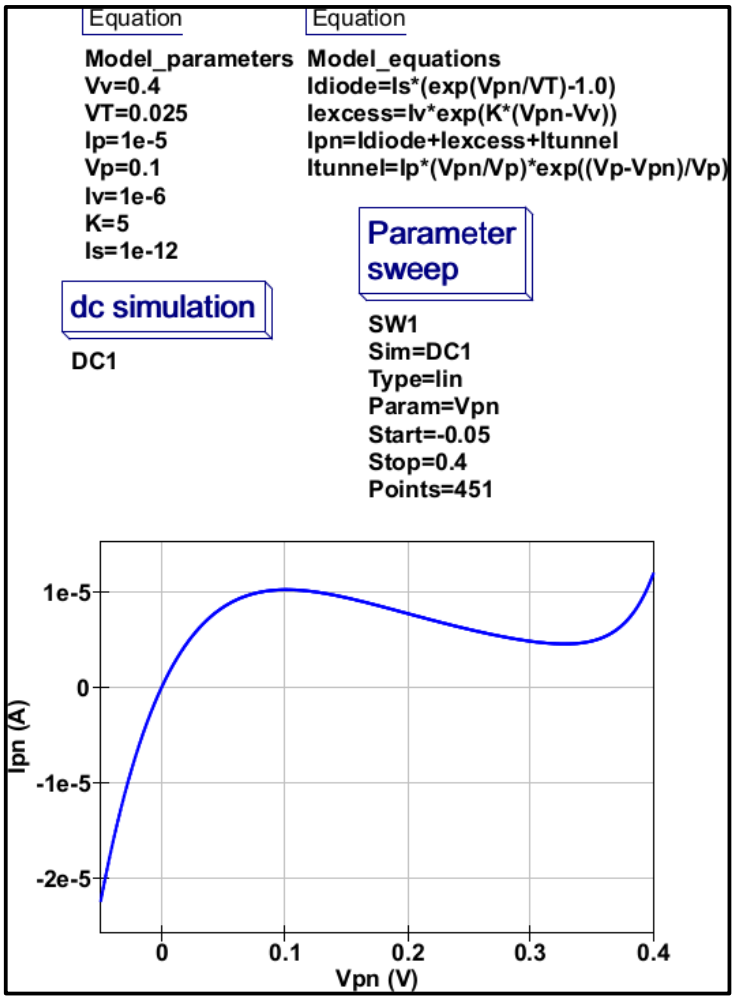
Jaijeet Roychowdhury, Numerical simulation and modelling of electronic and biochemical systems, Foundations and Trends in Electronic Design Automation 3:2-3, NOW, 2009

Compact device modelling: 3. Expanded compact device modelling capabilities with the Berkeley Model and Algorithm Prototyping Platform (MAPP); part 3



Compact device modelling: 3. Expanded compact device modelling capabilities with the Berkeley Model and Algorithm Prototyping Platform (MAPP); part 4

Qucs evaluation of tunnel diode model equations



MAPP tunnel diode test wrapper

```
function plotIV_tunnelDiode_ModSpec_wrapper()
    MOD = tunnelDiode_ModSpec_wrapper();

    vs = -0.05:0.001:0.4;
    is = zeros(size(vs));

    S = ee_model_parm2struct(MOD);
    for idx = 1:1:size(is,2)
        S.vpn = vs(1,idx);
        is(1,idx) = MOD.fe_of_S(S);
    end

    figure();

    plot([min(vs), max(vs)], [0, 0], 'Color', 'red', ...
      'LineWidth', 1.25, 'LineStyle', '--');
    hold on;
    plot([0, 0], [min(is), max(is)]*1e6, 'Color', ...
       'red', 'LineWidth', 1.25, 'LineStyle', '--');
    h = plot(vs, is*1e6, 'Color', 'blue', 'LineWidth', 1.75);

        axis tight;
        box on;
        grid on;
        set(gca,'FontName','Times New Roman','FontSize', ...
           15,'FontWeight','bold');
        xlabel('vpn (V)','FontName','Times New Roman',...
           'FontSize',18,'FontWeight','bold');
        ylabel('ipn (uA)','FontName','Times New Roman',...
           'FontSize',18,'FontWeight','bold');
        title(['I/V curve of a tunnel diode'],'FontName',...
            'Times New Roman','FontSize',18,'FontWeight','bold');
        set(gcf,'color','white');
end
```

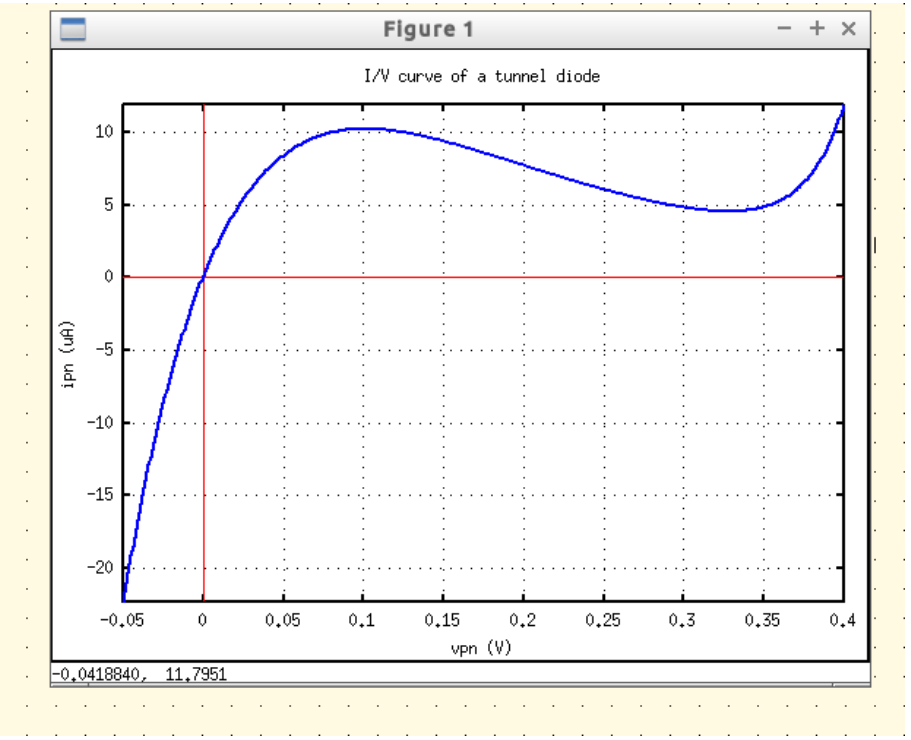

S. Jahn and M.E. Brinson, Interactive compact device modelling using Qucs equation-defined devices, Int. J. Numer. Model. 2008; 21:335-349. http://onlinelibrary.wiley.com/doi/10.1002/jnm.676/abstract
http://www.mos-ak.org/berkeley_2014/presentations/07_Tianshi_Wang_MOS-AK_Berkeley_2014.pdf

Compact device modelling: 3. Expanded compact device modelling capabilities with the Berkeley Model and Algorithm Prototyping Platform (MAPP); part 5

MAPP tunnel diode compact device model

```
function MOD = tunnelDiode_ModSpec_wrapper()
    MOD = ee_model();
    MOD = add_to_ee_model(MOD, 'external_nodes', {'p', 'n'});
    MOD = add_to_ee_model(MOD, 'explicit_outs', {'ipn'});
    MOD = add_to_ee_model(MOD, 'parms', {'Is', 1e-12, 'VT', 0.025});
    MOD = add_to_ee_model(MOD, 'parms', {'Ip', 1e-5, 'Vp', 0.1});
    MOD = add_to_ee_model(MOD, 'parms', {'Iv', 1e-6, 'Vv', 0.4, 'K', 5});
    MOD = add_to_ee_model(MOD, 'parms', {'C', 0});
    MOD = add_to_ee_model(MOD, 'f', @f);
    MOD = add_to_ee_model(MOD, 'q', @q);
    MOD = finish_ee_model(MOD);
end

function out = f(S)
    v2struct(S);
    I_diode = Is*(exp(vpn/VT)-1);
    I_excess = Iv * exp(K * (vpn - Vv));
    I_tunnel = (Ip/Vp) * vpn * exp(-1/Vp * (vpn - Vp));
    out = I_diode + I_tunnel + I_excess;
end

function out = q(S)
    v2struct(S);
    out = C*vpn;
end
```

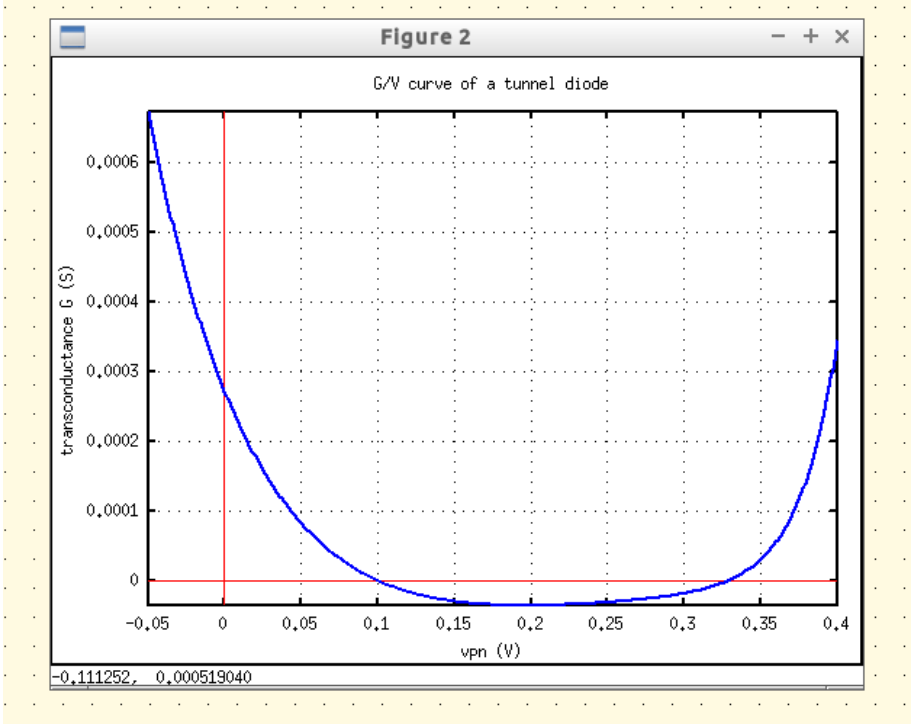
Tunnel diode I/V curve



MAPP post-simulation output

Compact device modelling: 3. Expanded compact device modelling capabilities with the Berkeley Model and Algorithm Prototyping Platform (MAPP); part 6

```
function out = plotGV_tunnelDiode_ModSpec_wrapper()
    MOD = tunnelDiode_ModSpec_wrapper();

    vs = -0.05:0.001:0.4;
    gs = zeros(size(vs));

    for idx = 1:1:size(gs,2)
        gs(1,idx) = MOD.dfe_dvecX(vs(idx), [], [], [], MOD);
    end

    figure();

    plot([min(vs), max(vs)], [0, 0], 'Color', 'red', 'LineWidth', …
        1.25, 'LineStyle', '--');
    hold on;
    plot([0, 0], [min(gs), max(gs)], 'Color', 'red', 'LineWidth', …
       1.25, 'LineStyle', '--');
    h = plot(vs, gs, 'Color', 'blue', 'LineWidth', 1.75);

          axis tight;
          box on;
          grid on;
          set(gca,'FontName','Times New Roman','FontSize',...
            15,'FontWeight','bold');
          xlabel('vpn (V)','FontName','Times New Roman','FontSize',...
            18,'FontWeight','bold');
          ylabel('transconductance G (S)','FontName','Times New Roman',...
            'FontSize',18,'FontWeight','bold');
          title(['G/V curve of a tunnel diode'],'FontName','Times New Roman', …
          'FontSize',18,'FontWeight','bold');
          set(gcf,'color','white');

end
```



Tunnel diode G/V curve

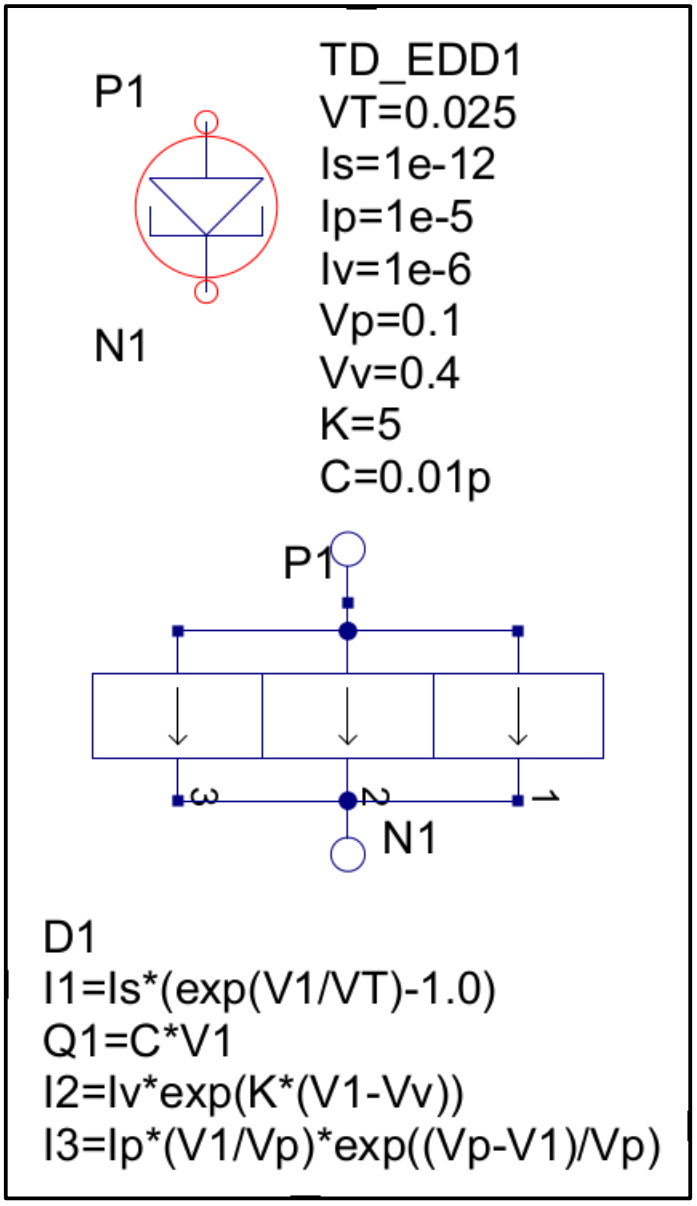
Compact device modelling: 3. Expanded compact device modelling capabilities with the Berkeley Model and Algorithm Prototyping Platform (MAPP); part 7

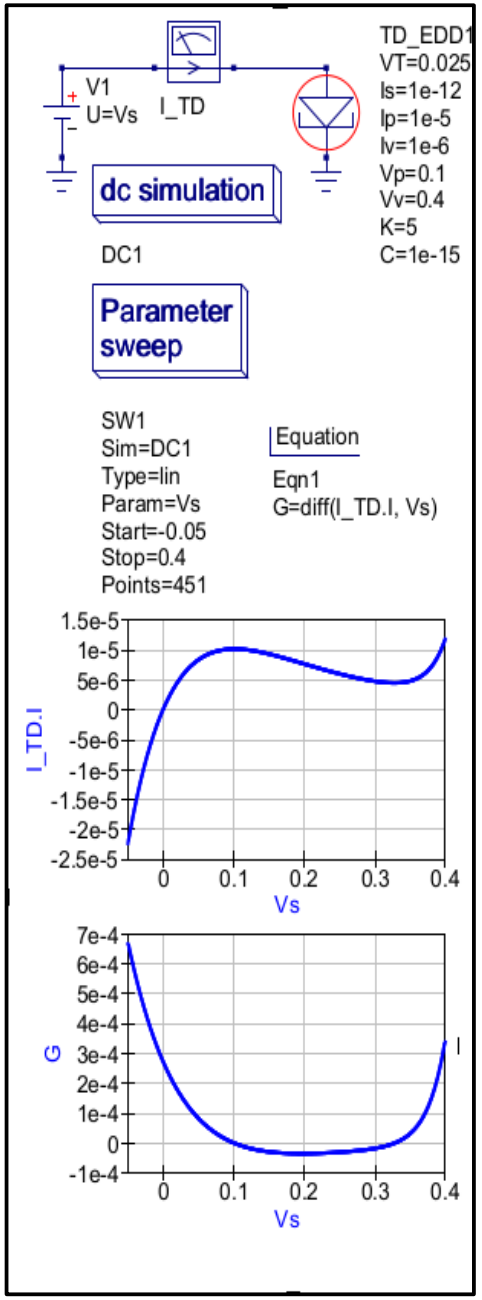
Equation

Model_parameters
Vv=0.4
VT=0.025
Ip=1e-5
Vp=0.1
Iv=1e-6
K=5
Is=1e-12

Equation

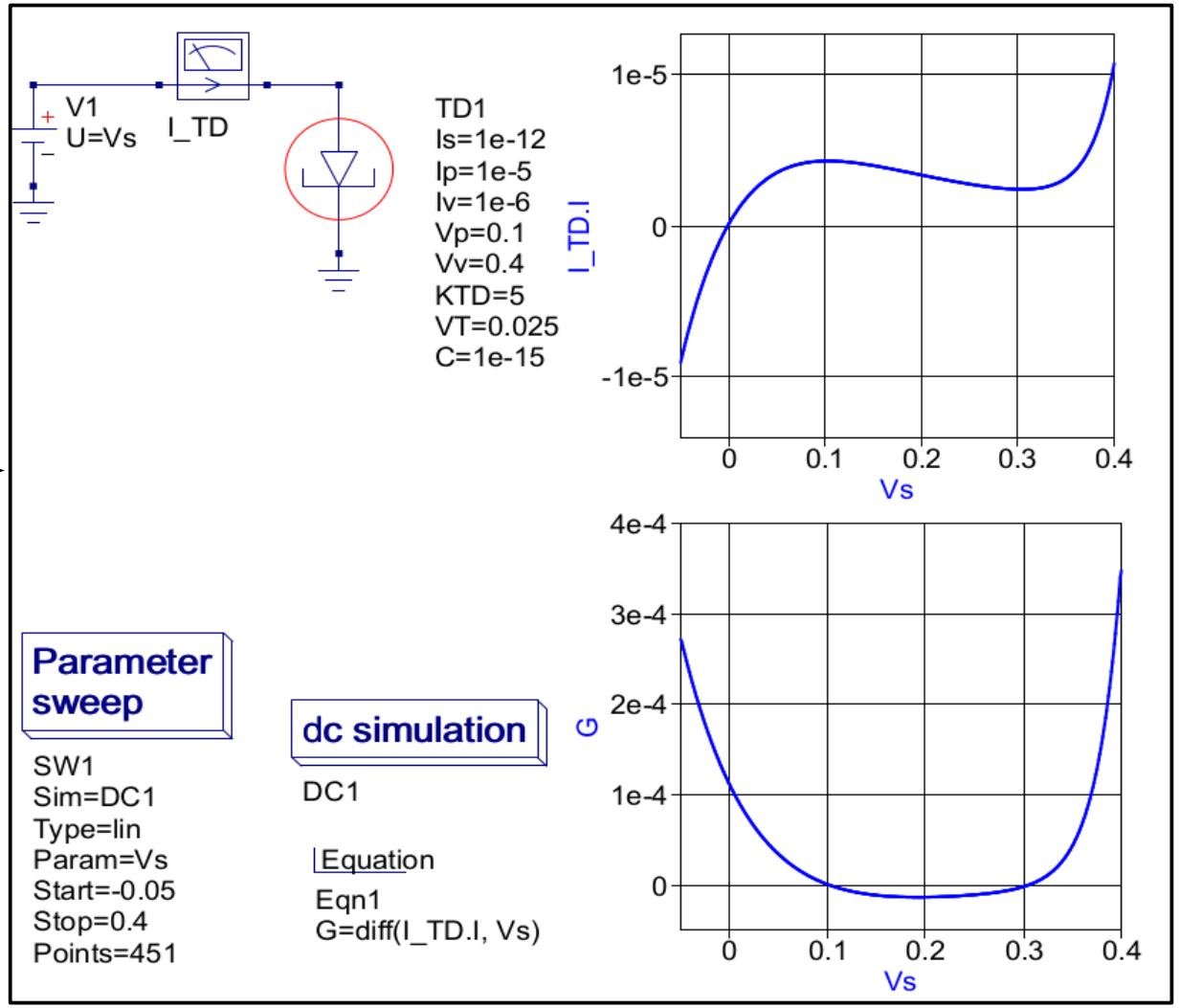
Model_equations
Idiode=Is*(exp(Vpn/VT)-1.0)
Iexcess=Iv*exp(K*(Vpn-Vv))
Ipn=Idiode+Iexcess+Itunnel
Itunnel=Ip*(Vpn/Vp)*exp((Vp-Vpn)/Vp)



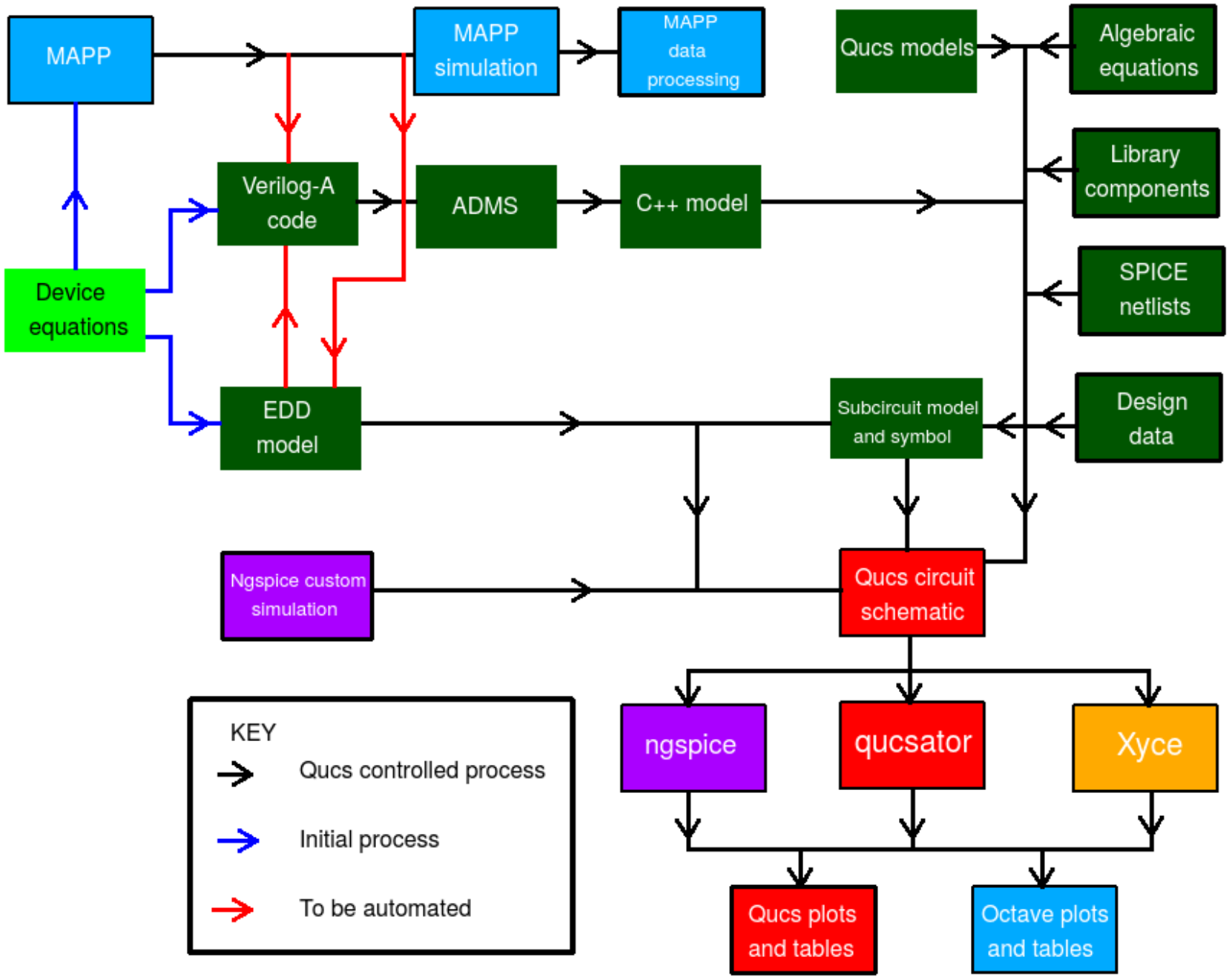


Compact device modelling: 3. Expanded compact device modelling capabilities with the Berkeley Model and Algorithm Prototyping Platform (MAPP); part 8

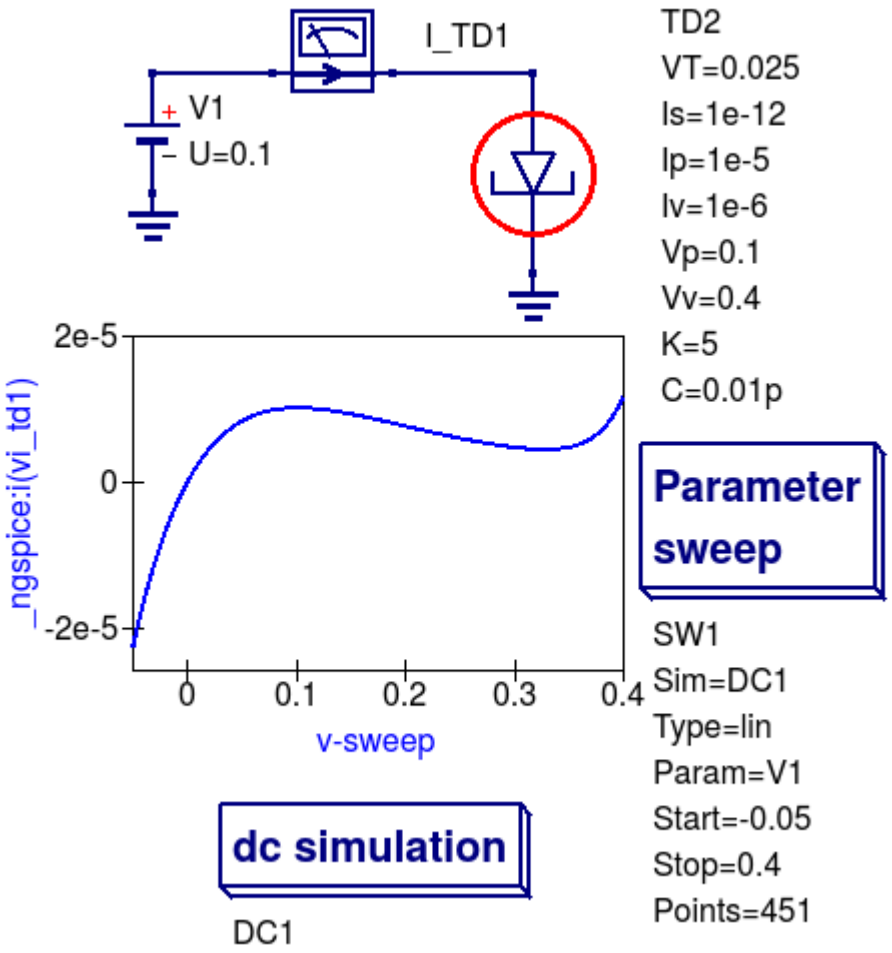
```
// Tunnel diode Verilog-A compact device model.
// Verilog-A code translated, by hand, from
// MAPP model.
//
`include "disciplines.vams"
`include "constants.vams"
//
module TD(p,n);
inout p,n;
electrical p,n;
//
parameter Is=1e-12 from [1e-20 : inf);
parameter Ip=1e-5  from [1e-20 : inf);
parameter Iv=1e-6  from [1e-20 : inf);
parameter Vp=0.1   from [1e-20 : inf);
parameter Vv=0.4   from [1e-20 : inf);
parameter KTD = 5    from [1e-20 : inf);
parameter VT=0.025 from [1e-20 : inf);
parameter C=1e-15  from [1e-20 : inf);
//
real Idiode, Iexcess, Itunnel;
//
analog begin
 Idiode = Is*(exp(V(p,n)/VT)-1.0);
 Iexcess = Iv*exp(KTD*(V(p,n)-Vv));
 Itunnel = Ip*(V(p,n)/Vp)*exp(Vp-V(p,n)/Vp);
 I(p,n) <+ Idiode+Iexcess+Itunnel;
 I(p,n) <+ ddt(C*V(p,n));
end
endmodule
```



A unified GPL compact device modelling and simulation platform



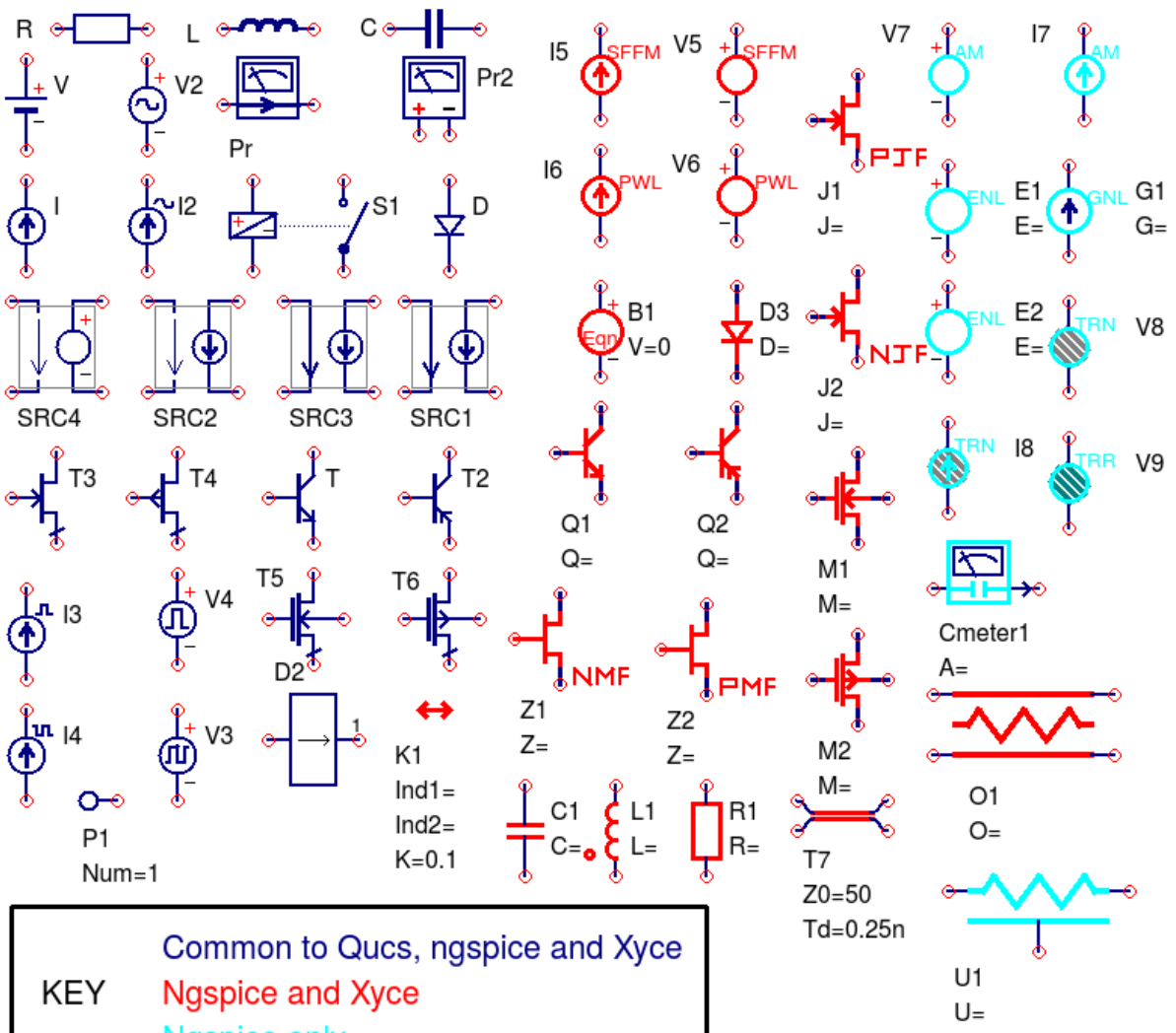
Qucs development: 1. A unified GPL compact device modelling and simulation platform; SPICE netlist synthesis




```
* Qucs 0.0.19
* Qucs 0.0.19  TD.sch
.SUBCKT TD _net0 _net1 VT=0.025 Is=1e-12 Ip=1e-5 Iv=1e-6 Vp=0.1 Vv=0.4 K=5 C=0.01p
BD1I0 _net0 _net1 I=Is*(exp((V(_net0)-V(_net1))/VT)-1.0)
GD1Q0 _net0 _net1 nD1Q0 _net1 1.0
LD1Q0 nD1Q0 _net1 1.0
BD1Q0 nD1Q0 _net1 I=-(C*(V(_net0)-V(_net1)))
BD1I1 _net0 _net1 I=Iv*exp(K*((V(_net0)-V(_net1))-Vv))
BD1I2 _net0 _net1 I=Ip*((V(_net0)-V(_net1))/Vp)*exp((Vp-(V(_net0)-V(_net1)))/Vp)
.ENDS
XTD2  _net0 0 TD VT=0.025 Is=1E-12 Ip=1E-5 Iv=1E-6 Vp=0.1 Vv=0.4 K=5 C=0.01P
VI_TD1 _net1 _net0 DC 0 AC 0
V1 _net1 0 DC 0.1
.control
set filetype=ascii
DC V1 -0.05 0.4 0.000997783
write _dc.txt VI_TD1#branch
destroy all
reset

exit
.endc
.END
```

Qucs schematic → Synthesis → Ngspice/Xyce netlist

A unified GPL compact device modelling and simulation platform: ngspice and Xyce component symbols



KEY

Common to Qucs, ngspice and Xyce

Ngspice and Xyce

Ngspice only

.PARAM
SpicePar1
y=1

.OPTIONS
SpiceOptions1
GMIN=1e-12

.GLOBAL PARAM
SpGlobPar1
y=1

Noise simulation

NOISE1
Output=v(node1)
Source=V1

Ngspice custom simulation

CUSTOM1
SpiceCode=

ac simulation

AC1
Type=lin
Start=1 GHz
Stop=10 GHz
Points=19

transient simulation

TR1
Type=lin
Start=0
Stop=1 ms

Fourier simulation

FOUR1
Sim=TR1
numfreq=10
F0=1kHz
Vars=V(1)

Distortion simulation

DISTO1
Type=lin
Start=1 Hz
Stop=10 kHz
Points=100

Harmonic balance simulation

HB1
n=4

dc simulation

DC1

Parameter sweep

SW1
Sim=
Param=R1

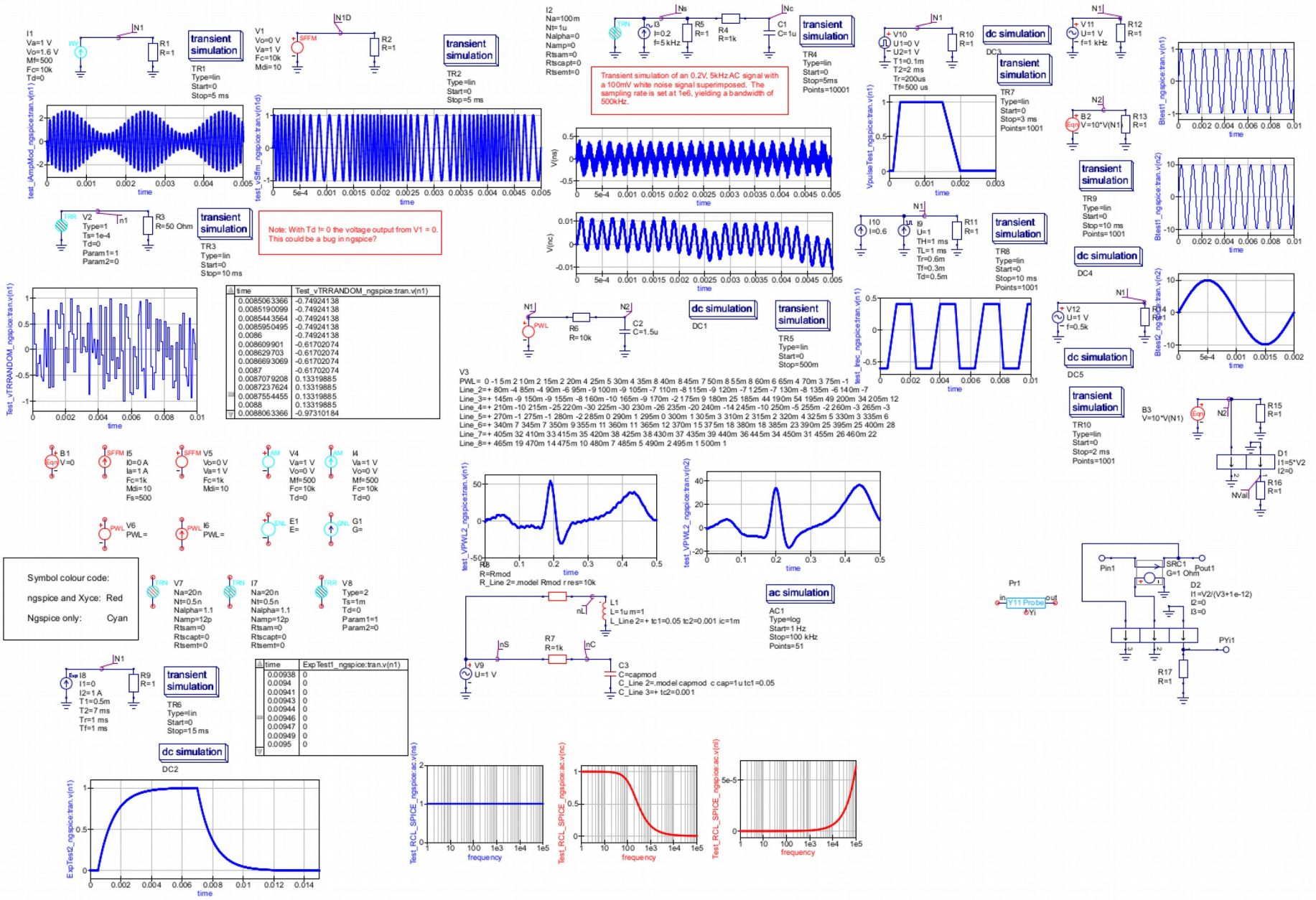
.NODESET
Nodeset1
v(node1)=1

.IC
SpiceIC1
v(node1)=1

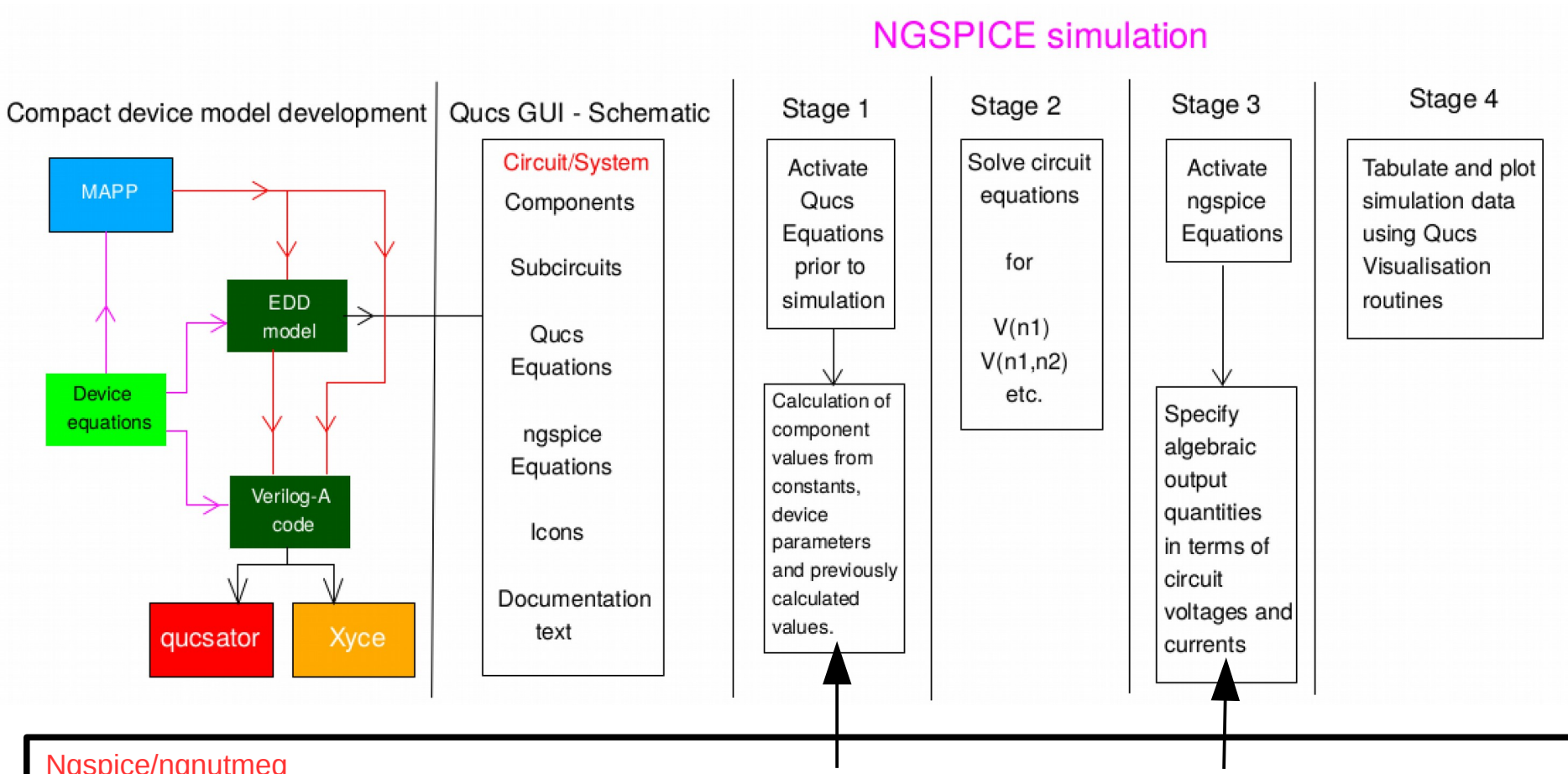
Nutmeg
NutmegEq1
Simulation=ac
y=1

Equation
Eqn1
y=1

A unified GPL compact device modelling and simulation platform: Signal generator test examples



A unified GPL compact device modelling and simulation platform: Integrating Qucs and ngspice



Ngspice/ngnutmeg

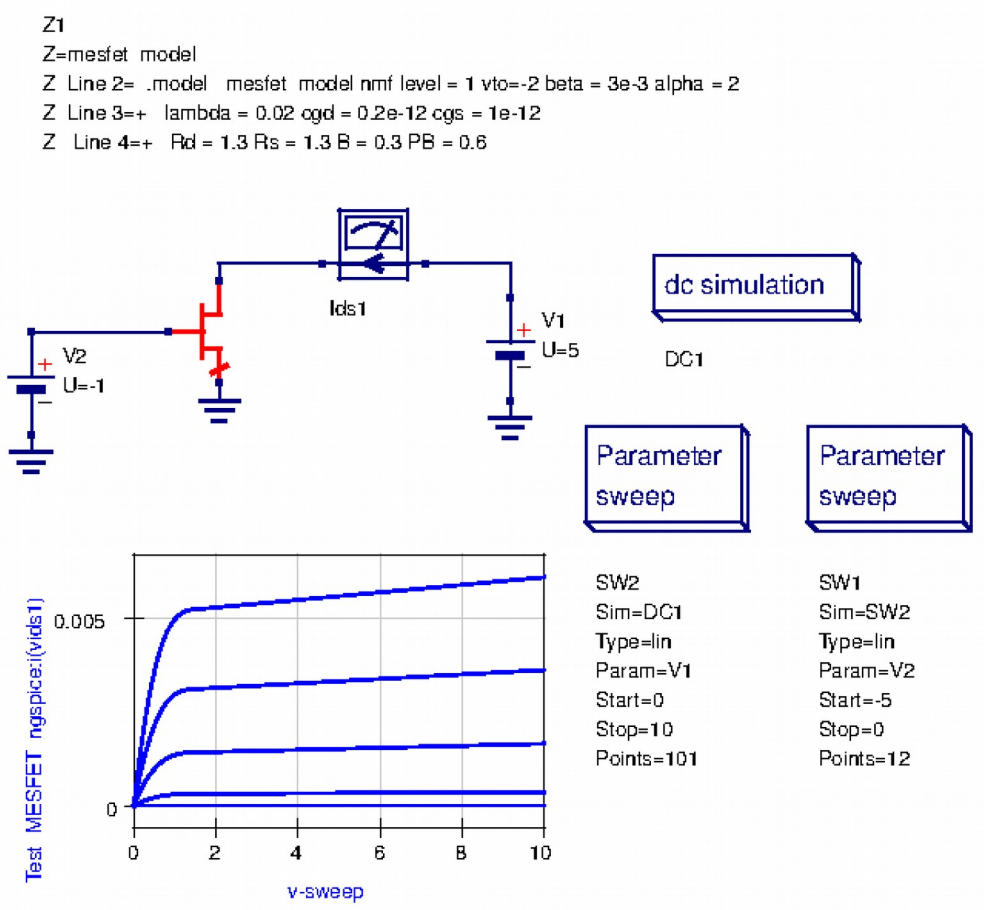
Stores data in the form of vectors:time, voltage etc. Each type of vector can be operated on and combined algebraically. Expressions are algebraic formula involving vectors and scalars.

Ngnutmeg defines the following functions: mag, ph,cph,j,real,imag,db,log, ln, exp,abs,sqrt,sin,cos,tan,atan,sinh,cosh, tanh,floor,ceil, norm, mean avg,group_delay,vector,unitvector.length,interpoate,deriv,vecd,vecmax.rnd,sgauss,sunif, poisson,exponential.

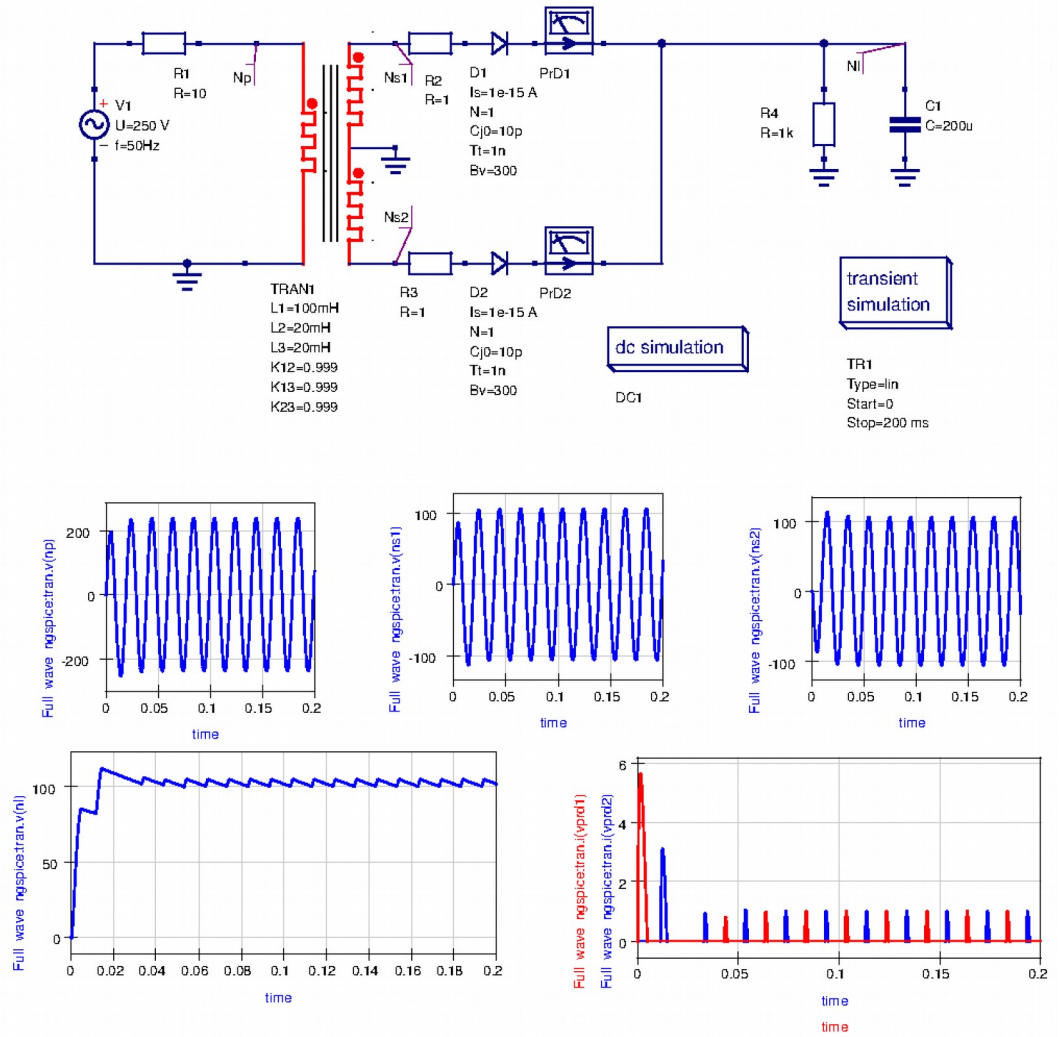
Allowed operators: +,-,*,/,^,%. , ,>, <, >=, <=, <>, &, |, !, =

Constants: pi, e, c, I, kelvin, echarge, boltz, plank, yes, no, TRUE, FALSE

A unified GPL compact device modelling and simulation platform: Typical simulation examples

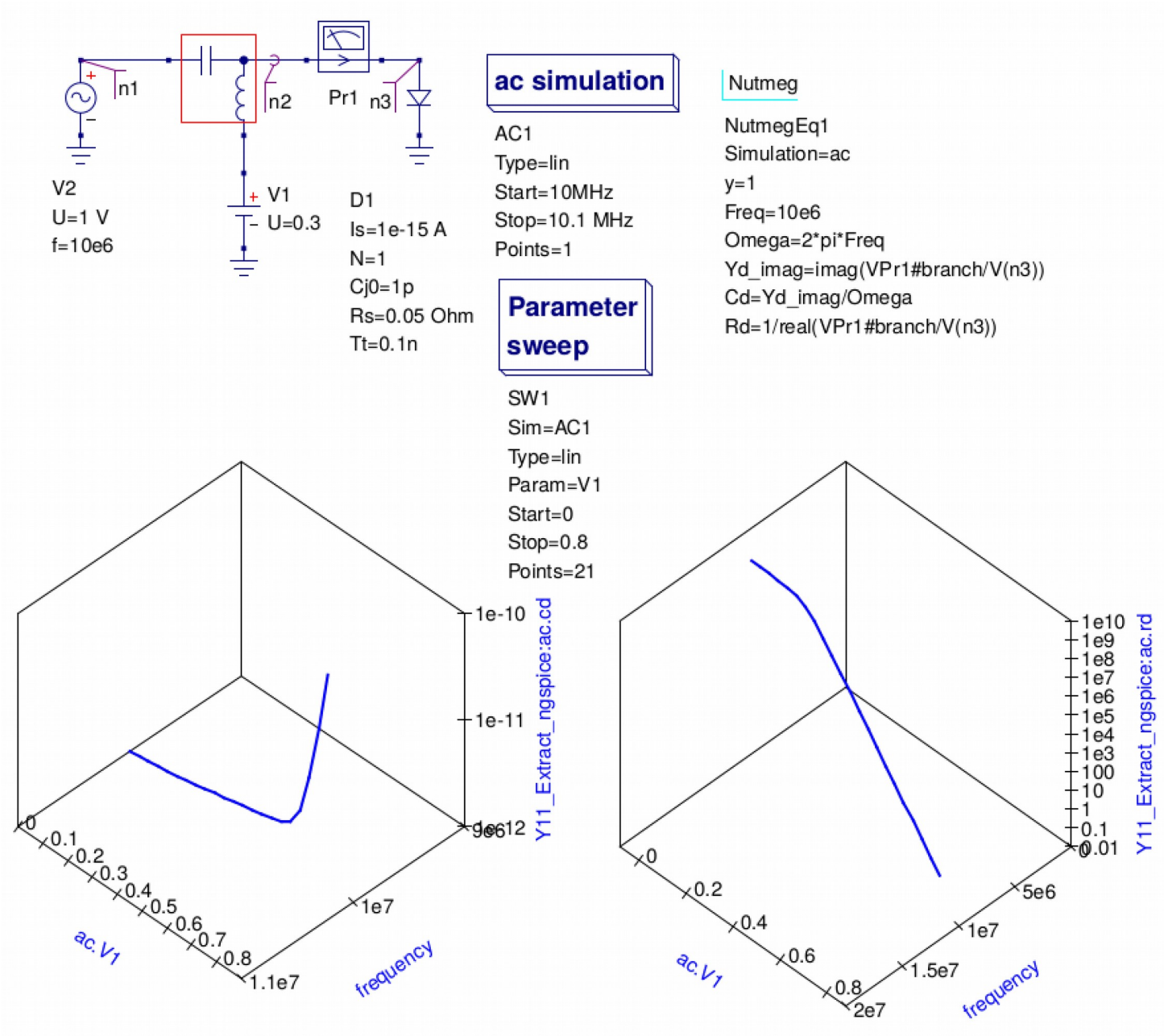


(1) MESFET DC characteristic test bench using nested parameter sweep

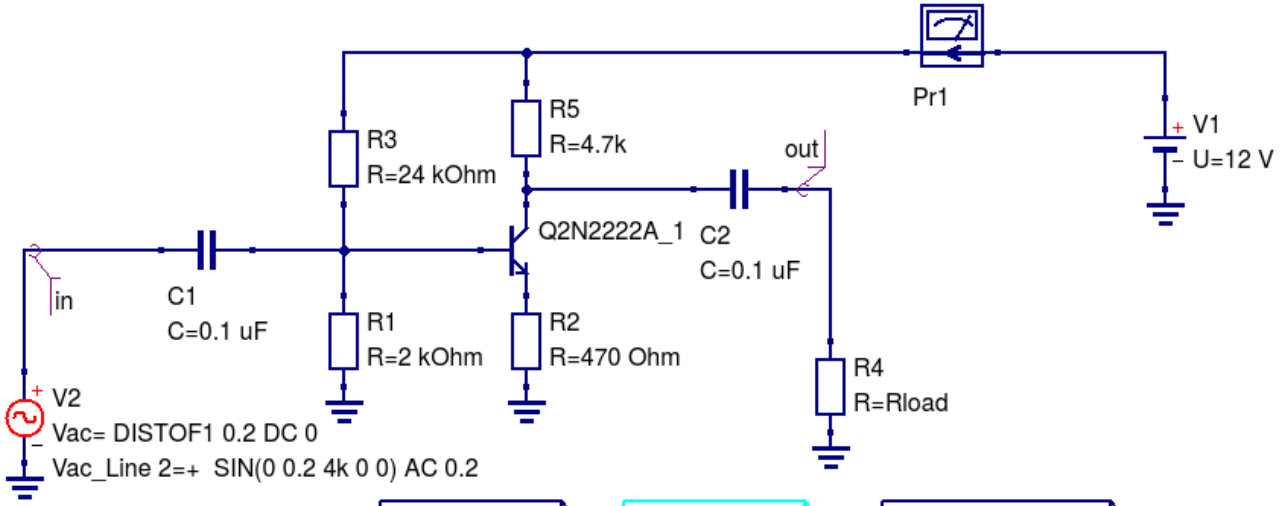


(2) Basic full wave rectifier with transformer modelled by SPICE K type inductive coupling

A unified GPL compact device modelling and simulation platform: Semiconductor diode Cd and Rd extraction using Y parameters



A unified GPL compact device modelling and simulation platform: More ngspice simulation types; small signal distortion



Equation

Eqn1
Rload=47k
K=out.v/in.v

Nutmeg

NutmegEq1
Simulation=disto
H2=disto1.v(out)
H3=disto2.v(out)

transient simulation

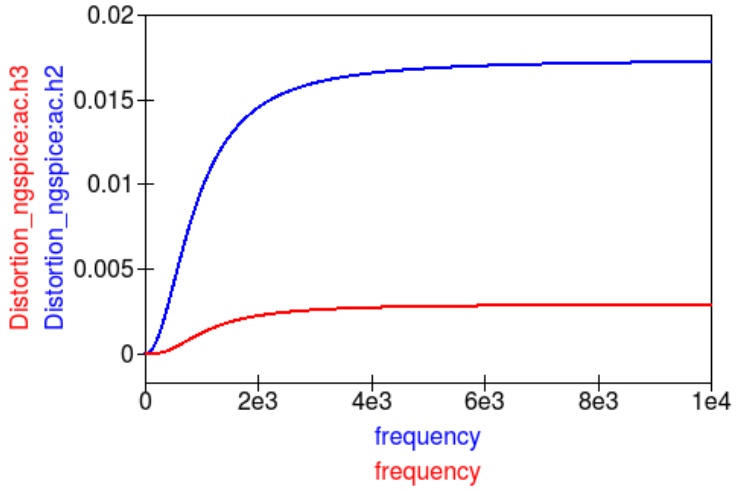
TR1
Type=lin
Start=0
Stop=1 ms

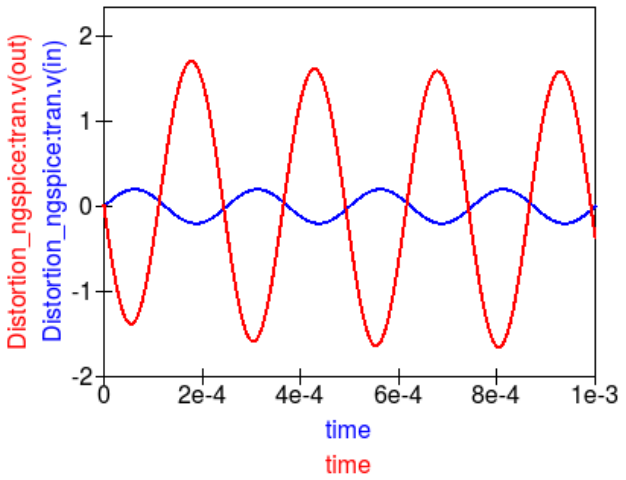
Distortion simulation

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

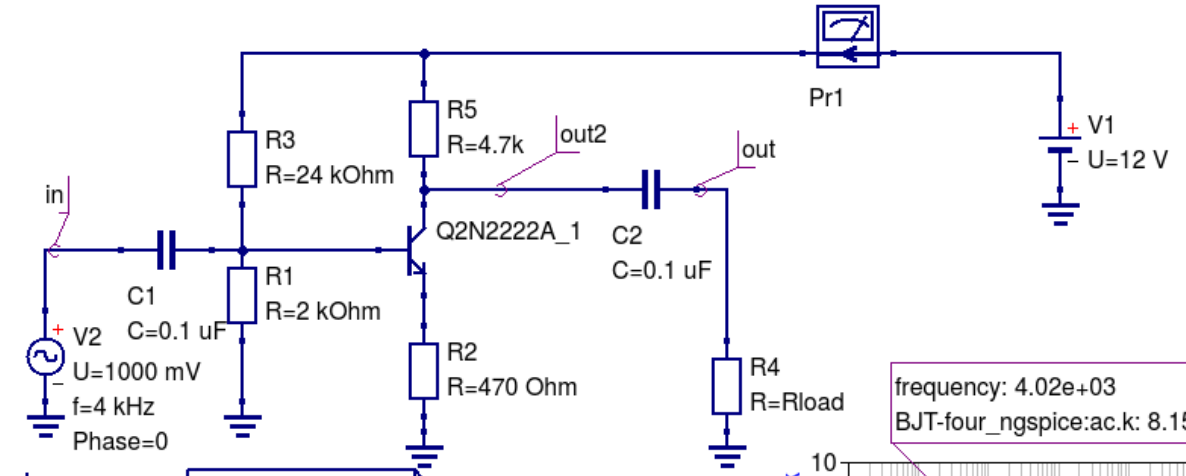
dc simulation

DC1





A unified GPL compact device modelling and simulation platform: More ngspice simulation types; Fourier analysis



Equation

Eqn1
Rload=47k
K=out.v/in.v
Kp=(cph(out.v) - cph(in.v))*180/pi

dc simulation

DC1

ac simulation

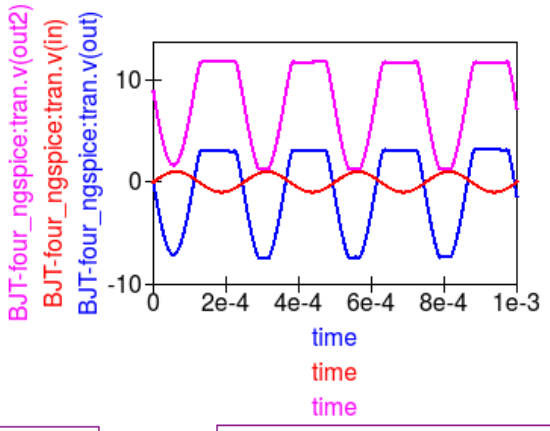
AC1
Type=log
Start=100 Hz
Stop=100 MHz
Points=601

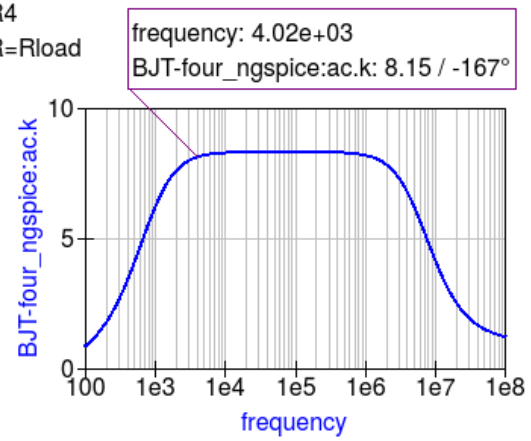
transient simulation

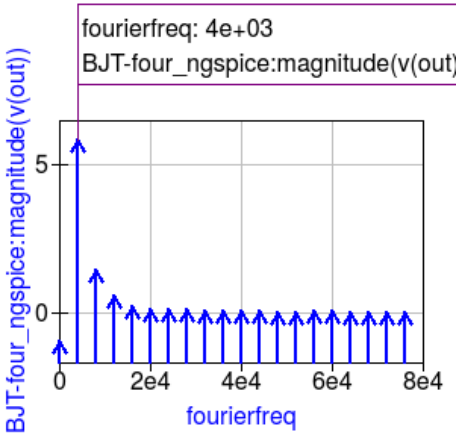
TR1
Type=lin
Start=0
Stop=1 ms

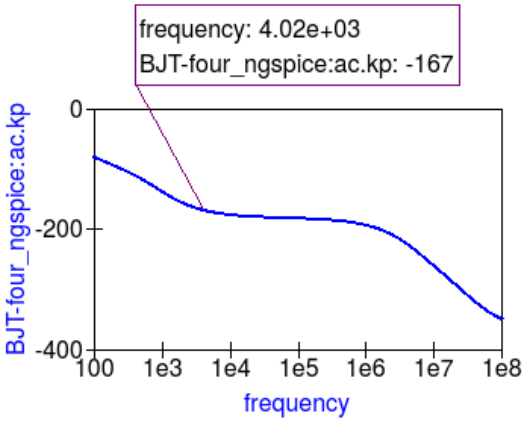
Fourier simulation

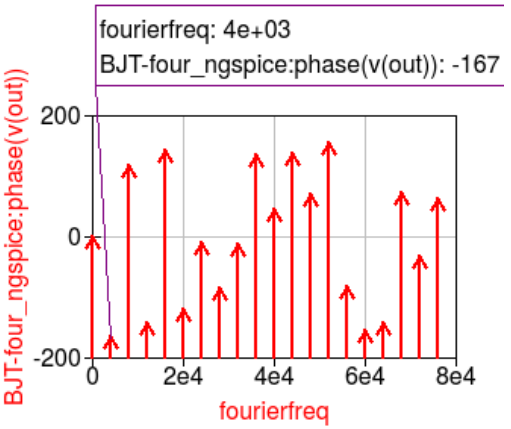
FOUR1
Sim=TR1
numfreq=20
F0=4kHz
Vars=V(out) V(out2)





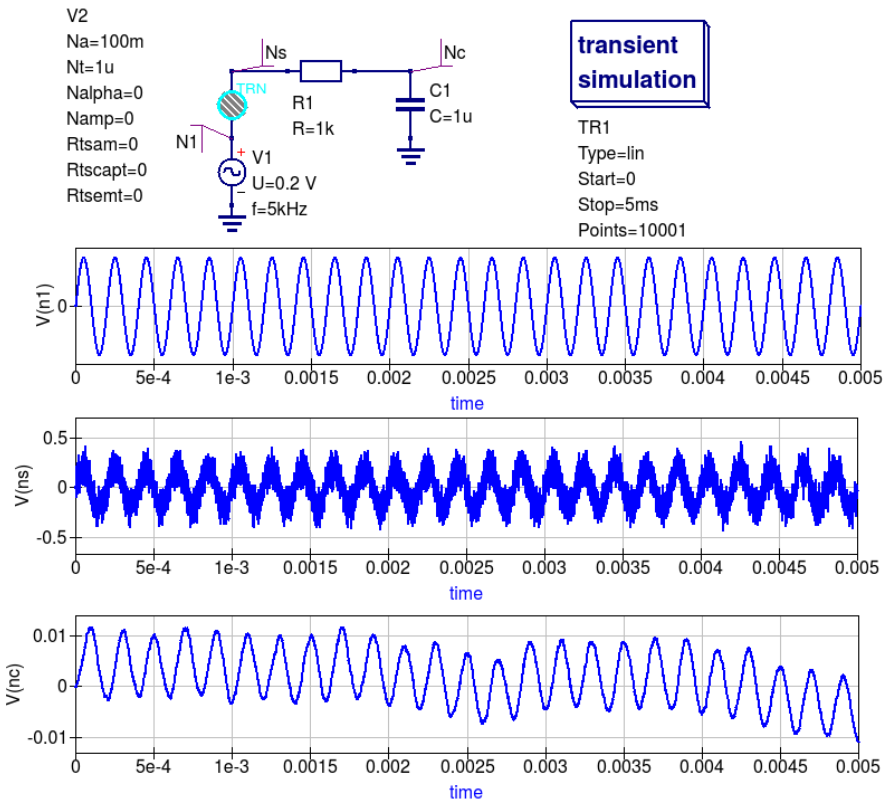




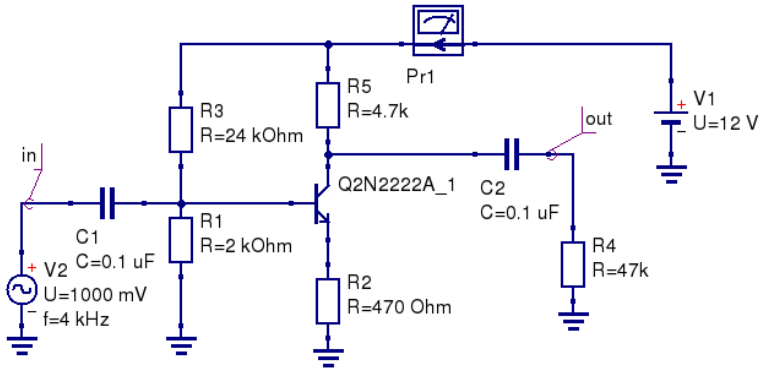


A unified GPL compact device modelling and simulation platform: More ngspice simulation types; Noise analysis

Large signal transient noise



Small signal AC noise



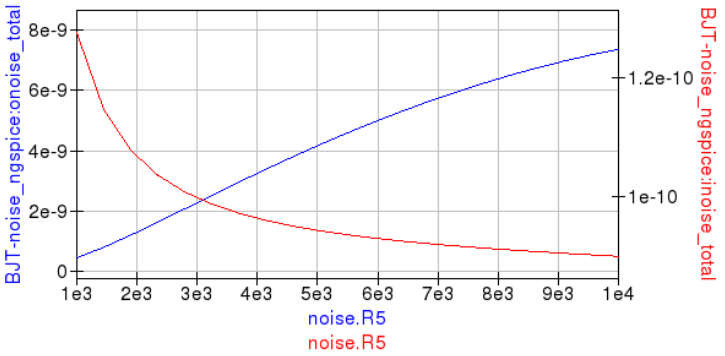
Parameter sweep

SW1
Sim=NOISE1
Type=lin
Param=R5
Start=1k
Stop=10k
Points=20

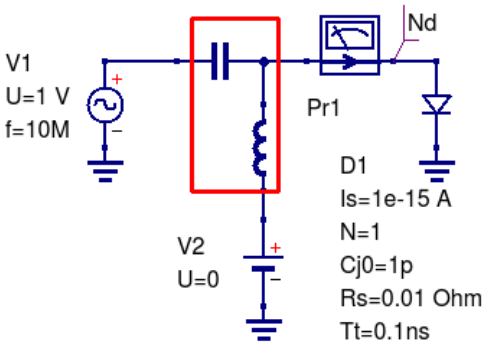
Noise simulation

NOISE1
Type=lin
Start=1 Hz
Stop=10 MHz
Points=100
Output=v(out)
Source=V2

noise.R5	BJT-noise_ngspice:inoise_total	BJT-noise_ngspice:onoise_total
1e3	1.28e-10	4.58e-10
1.45e3	1.15e-10	8.09e-10
1.9e3	1.08e-10	1.21e-09
2.35e3	1.04e-10	1.64e-09
2.8e3	1.01e-10	2.09e-09
3.25e3	9.88e-11	2.53e-09
3.7e3	9.72e-11	2.97e-09
4.15e3	9.6e-11	3.4e-09

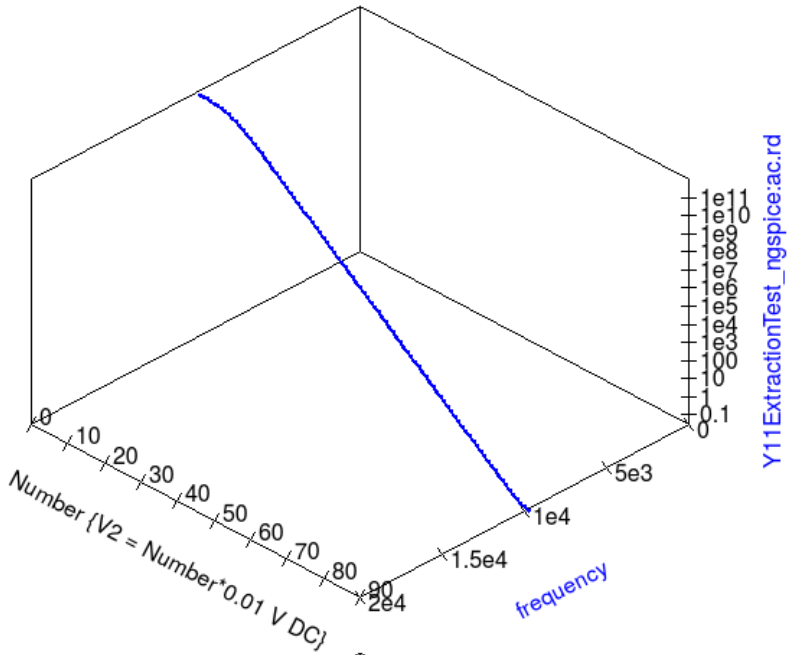


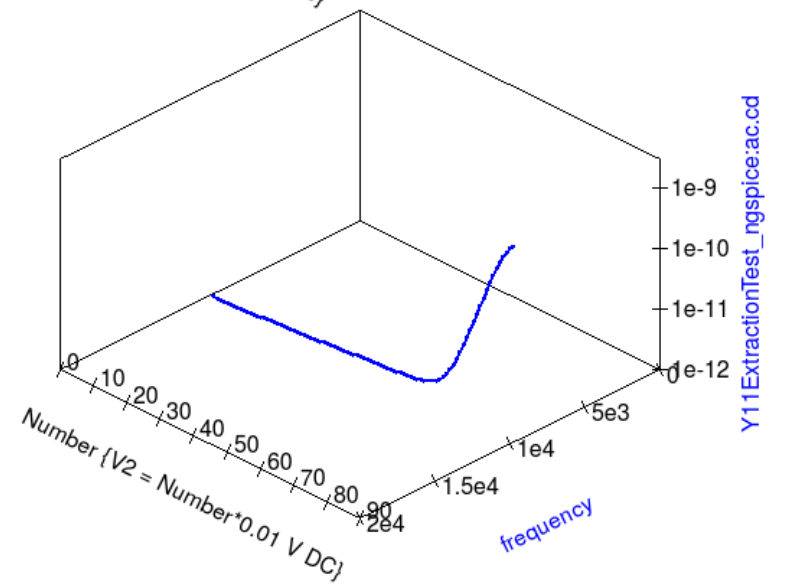
A unified GPL compact device modelling and simulation platform: More ngspice simulation types; Custom ngspice simulation



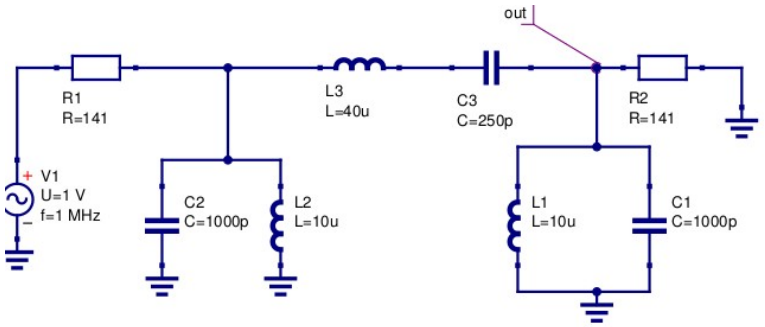
Ngspice custom simulation

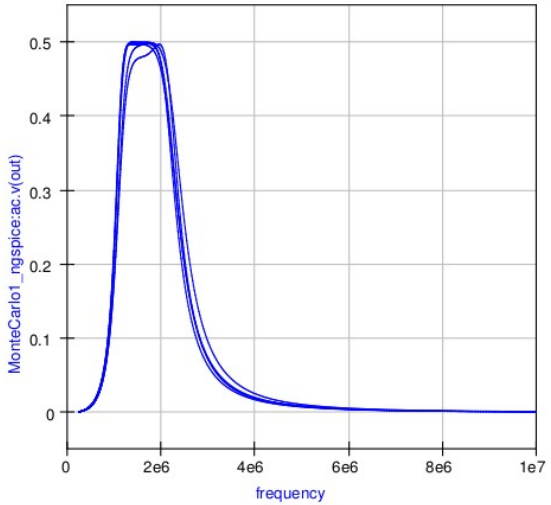
```
CUSTOM1
SpiceCode=
** Extraction of diode Cd and Rd values for 0 <= Vd <=0.9 V DC.
*********************************************************************************
* Initialise data
destroy all
set filetype=ascii
let count = 0
let count_Fin = 90
*********************************************************************************
** Run simulation and extract Cd and Rd values for each value of V2 DC.
while count le count_Fin
  alter V2 = count*0.01
  ac lin 1  1e4 2e4
  let Omega  = 2*pi*frequency
  let Cd = imag(VPr1#branch/V(Nd))/Omega
  let Rd = 1.0/(real( VPr1#branch/V(Nd))+1e-15 )
  write Y11Test_ac.txt Cd Rd VPr1#branch V(Nd)
  set appendwrite
  let count=count+1
end
*********************************************************************************
```

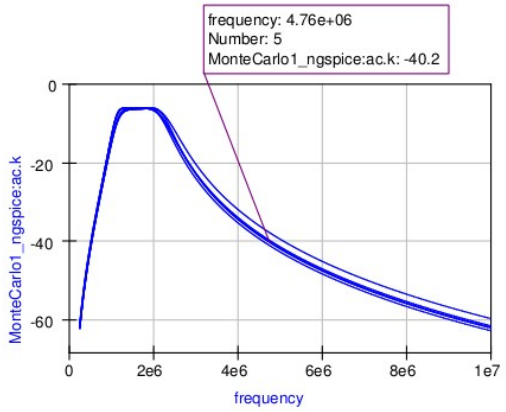




A unified GPL compact device modelling and simulation platform: More ngspice simulation types; Monte Carlo analysis







Ngspice
custom simulation

```
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) ge 0) ? avar : -avar))
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 c1 = aunif(1e-09, 100e-12)
* alter c1 = gauss(1e-09, 0.1, 3)
* alter c1 = agauss(1e-09, 100e-12, 3)
*
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 10Meg

set run ="$&run" $ create a variable from the vector

let K = db(v(out))
write MonteCarlo1_custom.txt v(out) K
set appendwrite
let run = run + 1
end $ loop ends here
```

Summary

This presentation outlined the compact modelling and simulation features implemented in the current and next generation versions of the Qucs GPL circuit simulation tool set.

Qucs version 0.0.19 will be released on, or before, 1 September 2015.

The latest Qucs development version can be downloaded using git from https://github.com/Qucs/qucs/

If you wish to follow the development of Qucs see the regularly updated spice4qucs-help document at http://qucs-help.readthedocs.org/en/spice4qucs/DModel.html

Beyond Qucs-0.0.19

- Addition of SPICE .PZ etc to Qucs
- Improvements in Qucs/Xyce link
- Synthesis of SPICE code for Qucs RFEDD models
- Improved RF device modelling and simulation
- Synthesis of Verilog-A model code from Qucs EDD compact device models
- More work on the Qucs/MAPP link

Acknowledgement: My thanks to all the members of the Qucs Development Team for their input and support during the recent intense software development phase.

References

Brinson M. and Jahn S., Interactive compact device modelling using Qucs equation defined devices, International Journal of Numerical Modelling: Electrical Networks, Devices and Fields, 21(5) pp. 335-349, September/October 2008. DOI : 10.1002/jnm.676.

Brinson M. and Jahn S., Qucs: A GPL software package for circuit simulation, compact device modelling and circuit macromodelling from DC to RF and Beyond, International Journal of Numerical Modelling: Electrical Networks, Devices and Fields, 22(4) pp. 297-319, July/August 2009. DOI : 10.1002/jnm.702.

Brinson M. and Jahn S., Compact macromodelling of operational amplifiers with equation defined devices, International Journal of Electronics, 96(2), pp. 109-122, February 2009, DOI:10.1080/00207210802580288, ISSN :0020-7217.

Brinson M. and Jahn S., Modelling of high-frequency inductance with Qucs non-linear radio frequency equation-defined devices, International Journal of Electronics, 96(3), March 2009, DOI:10.1080/00207210802640603, ISSN : 0020-7217.

Brinson M.E., Jahn S. and Nabijou H., Z Domain delay subcircuits and compact Verilog-A macromodels for mixed-mode sampled data circuit simulation, Test Technology Technical Council (TTTC) of the IEEE Computer Society, Radioelectronics & Informatics Journal, Vol. 45, No. 2, pp. 14-20, April/June 2009. ISSN 1563-0064.

Brinson M.E. and Nabijou H., Adaptive subcircuits and compact Verilog-A macromodels as integrated design and analysis blocks in Qucs circuit simulation,International Journal of Electronics, Vol. 98 (5), pp. 631-645, May 2011. DOI: 10.1080/00207217.2011.562452.

Brinson M.E., Jahn S. and Nabijou H., A tabular source approach to modelling and simulating device and circuit noise in the time domain, International Journal of Numerical Modelling: Electronic Networks, Devices and Fields, Vol 26(6), pp. 555-567, November/December 2011. DOI: 10.1002/jnm801.

Brinson M. and Jahn S., Compact device modelling for established and emerging technologies with the Qucs GPL circuit simulator, Mixed design of Integrated Circuits and Systems (MIXDES) 2009, Proceedings of the 16 International Conference, pp. 39-44, Lodz, Poland, June 2009. ISBN 978-1-4244-4798-5. INSPEC Accession Number: 10928855. Available from: http://ieeexplore.org .

References

Brinson M.E.and Nabijou H., Adaptive EPFL-EKV long and short channel MOS device models for Qucs, SPICE and Modelica circuit simulation, Mixed design of Integrated Circuits and Systems (MIXDES) 2011, Proceedings of the 18 International Conference, pp. 65-70, Gliwice, Poland, June 2011, ISBN 978-1-4577-0804-1, INSPEC Accession Number: 12219696. Available from http://ieeexplore.org .

Brinson M.E.and Margraf M., Verilog-A compact semiconductor device modelling and circuit macromodelling with the QucsStudio-ADMS “Turn-Key” modelling system , Mixed design of Integrated Circuits and Systems (MIXDES) 2012, Proceedings of the 19 International Conference, pp. 65-70, Warsaw, Poland, May 2011. ISBN 978-83-62954-43-8, INSPEC Accession Number: 12219696. Available from http://ieeexplore.org

Brinson M.E. From Qucs to QucsStudio: An international project to develop a freely available GU Public Licence circuit simulator with compact device modelling tools, data processing capabilities, manufacturing features and an analogue/RF design environment for engineers, MOS-AK/GSA International workshop on Device modeling for Microsystems, Jaypee Institute of Information Technology, Nodia, March 2012. Available from: http://www.mos-ak.org/india/presentations/Brinson_MOS-AK_India12.pdf .

Brinson M.E., Jahn S. and Nabijou H., A hybrid Verilog-A and equation-defined subcircuit approach to MOS switched current analog cell simulation, IETE Journal of research, Vol 58(3), pp. 177=185, May-June 2012.

Mike E. Brinson, and Hassan Nabijou, Adaptive EPFL-EKV long and short channel MOS device models for Qucs, SPICE and Modelica circuit simulation, International Journal of microelectronics and Computer Science, Vol.3, NO. 1, 2012, pp. 1-6. ISSN 2080-8755.

Mike E. Brinson, and Michael Margraf, Verilog-A compact semiconductor device modelling and circuit macromodelling with the QucsStudio-ADMS “Turn-Key” modelling system, International Journal of microelectronics and Computer Science, Vol.3, NO. 1, 2012, pp. 32-40. ISSN 2080-8755.

References

Brinson M.E., Jahn S. and Nabijou H., A hybrid Verilog-A and equation-defined subcircuit approach to MOS switched current analog cell modeling and simulation in the transient and large signal AC domains, Mixed design of Integrated Circuits and Systems (MIXDES) 2010, Proceedings of the 17 International Conference, pp. 3-48, Wroclaw, Poland, June 2010.ISBN 978-1-4244-7011-2, INSPEC Accession Number: 11487844. Available from:http://ieeexplore.ieee.org/xpl/freeabs_all.jsp?arnumber=5551306

Brinson M.E., Jahn S. and Nabijou H., Adaptive EPFL-EKV long and short channel MOS device models for Qucs, SPICE and Modelica circuit simulation, Mixed design of Integrated Circuits and Systems (MIXDES) 2011, Proceedings of the 18 International Conference, pp. 65-70, Gliwice, Poland, June 2011. ISBN 978-1-4577-0304-1. INSPEC Accession Number: 12219696. Available from http://ieeexplore.ieee.org/xpl/freeabs_all.jsp?arnumber=6016035

Jahn S., Brinson M.E., Margraf M., Parruitte H., Ardouin B., Nenzi P., and Lemaitre L., GNU simulators supporting Verilog-A compact model standardization, MOS-AK International Meeting, Premstaetten, Germany, March 2007. Available from: http://www.mos-ak.org/premstaetten/papers/MOS-AK_QUCS_ngspice_ADMS.pdf

Brinson M. and Jahn S., Building device models and circuit macromodels with the Qucs GPL simulator : A demonstration, Presentation to the European Network on Compact Modelling (COMON), Frankfurt(O), Germany, 2 April 2009. Available from: http://www.mos-ak.org/frankfurt_o/papers/M_Brinson_Qucs_COMON_April_2_2009_final.pdf

Brinson M., Jahn S. and Cullinan M., Advances in compact semiconductor device modelling and circuit macromodelling with the Qucs GPL circuit simulator, MOS-AK International Meeting, Frankfurt(O), Germany, 3 April 2009. Available from: http://www.mos-ak.org/frankfurt_o/papers/P_7_Brinson_MOS-AK_April_2009_final.pdf

Brinson M.E., Jahn S. and Nabijou H., Qucs, SPICE and Modelica equation-defined modelling techniques for the construction of compact device models based on a common model template structure, MOS-AK/GSA International workshop on the frontiers of compact modeling for advanced analog/RF applications, Université Pierre et Marie Curie, Paris, April 2011. Available from: http://www.mos-ak.org/paris/papers/P06_Brinson_MOS-AK_Paris.pdf

References

Brinson M.E.and Nabijou H., Adaptive EPFL-EKV long and short channel MOS device models for Qucs, SPICE and Modelica circuit simulation, Mixed design of Integrated Circuits and Systems (MIXDES) 2011, Proceedings of the 18 International Conference, pp. 65-70, Gliwice, Poland, June 2011, ISBN 978-1-4577-0804-1, INSPEC Accession Number: 12219696. Available from http://ieeexplore.org .

Brinson M.E.and Margraf M., Verilog-A compact semiconductor device modelling and circuit macromodelling with the QucsStudio-ADMS “Turn-Key” modelling system , Mixed design of Integrated Circuits and Systems (MIXDES) 2012, Proceedings of the 19 International Conference, pp. 65-70, Warsaw, Poland, May 2011. ISBN 978-83-62954-43-8, INSPEC Accession Number: 12219696. Available from http://ieeexplore.org

Brinson M.E. From Qucs to QucsStudio: An international project to develop a freely available GU Public Licence circuit simulator with compact device modelling tools, data processing capabilities, manufacturing features and an analogue/RF design environment for engineers, MOS-AK/GSA International workshop on Device modeling for Microsystems, Jaypee Institute of Information Technology, Nodia, March 2012. Available from: http://www.mos-ak.org/india/presentations/Brinson_MOS-AK_India12.pdf .

Brinson M.E., Jahn S. and Nabijou H., A hybrid Verilog-A and equation-defined subcircuit approach to MOS switched current analog cell simulation, IETE Journal of research, Vol 58(3), pp. 177=185, May-June 2012.

Mike E. Brinson, and Hassan Nabijou, Adaptive EPFL-EKV long and short channel MOS device models for Qucs, SPICE and Modelica circuit simulation, International Journal of microelectronics and Computer Science, Vol.3, NO. 1, 2012, pp. 1-6. ISSN 2080-8755.

Mike E. Brinson, and Michael Margraf, Verilog-A compact semiconductor device modelling and circuit macromodelling with the QucsStudio-ADMS “Turn-Key” modelling system, International Journal of microelectronics and Computer Science, Vol.3, NO. 1, 2012, pp. 32-40. ISSN 2080-8755.

References

Mike Brinson, Richard Crozier, Clemens Novak, Bastien Roucaries, Frans Schreuder, Guilherme Brondani Torri, Building a second generation Qucs GPL circuit simulator: package structure, simulation features and compact device modelling capabilities, MOS-AK Compact Device Modelling Workshop, London Metropolitan University, London(UK), March 28-29, 2014. Available from:
http://www.mos-ak.org/london_2014/presentations/09_Mike_Brinson_MOS-AK_London_2014.pdf

Mike Brinson, Richard Crozier, Clemens Novak, Bastien Roucaries, Frans Schreuder, Guilherme Brondani Torri, QUCS Roadmap: background to the new features in release 0.0.18 and an outline of future software development directions,12th MOS-AK ESSDERC/ESSCIRC Workshop, Venice(IT), September 26, 2014. Available from:
http://www.mos-ak.org/venice_2014/publications/T_4_Brinson_MOS-AK_Venice_2014.pdf

Mike Brinson, Richard Crozier, Clemens Novak, Bastien Roucaries, Frans Schreuder, Guilherme B. Torri, W. Grabinski, QUCS/ADMS/Verilog-A Update, 7th International MOS-AK at UC Berkeley (US), Dec.12, 2014. Available from:
http://www.mos-ak.org/berkeley_2014/presentations/04_Wladek_Grabinski_MOS-AK_Berkeley_2014.pdf

Mike Brinson, Richard Crozier, Vadim Kuznetsov, Clemens Novak, Bastien Roucaries, Frans Schreuder, Guilherme Brondani Torri, Qucs: improvements and new directions in the GPL compact device modelling and circuit simulation tool, MOS-AK Workshop at DATE Grenoble (F), April 2015. Available from:
http://www.mos-ak.org/grenoble_2015/presentations/T4_Brinson_MOS-AK_Grenoble_2015.pdf

Grabinski W, Brinson M, Nenzi P, Lannutti F, Makris N, Antonopoulos A, Bucher M. Open-source circuit simulation tools for RF compact semiconductor device modelling. Invited paper. International Journal of Numerical Modelling: Electronic Networks, Devices and Fields Volume 27, Issue 5-6, September-December 2014, Pages: 761–779.
DOI: 10.1002/jnm.1973.

NEW BOOK: Open Source/GNU CAD for Compact Modelling, Editors: Wladek Grabinski and Daniel Tomaszewski. Publisher: Mark de Jongh [Mark.de.Jongh@springer.sbm.com], www.springer-sbm.com.
Chapter 5: M.E. Brinson, Schematic entry and circuit simulation with Qucs.
Chapter 6: M.E. Brinson, Qucs modelling and simulation of analogue/RF devices and circuits.

Qucs is a Project of the Week on SourceForge: see
http://sourceforge.net/blog/project-of-the-week-june-1-2015/