Qucs: A GPL software package for circuit simulation, compact device modeling and circuit macromodeling from DC to RF and beyond

M. E. Brinson mbrin72043@yahoo.co.uk and S. Jahn stefan@gruft.de

Qucs Development Team

- Introduction
- Circuit Simulation
- Compact device modeling and circuit macromodeling
- Summary

Presented on April 4, 2008 at MOS-AK meeting, MiPlaza, High Tech Campus, Eindhoven, The Netherlands

Introduction

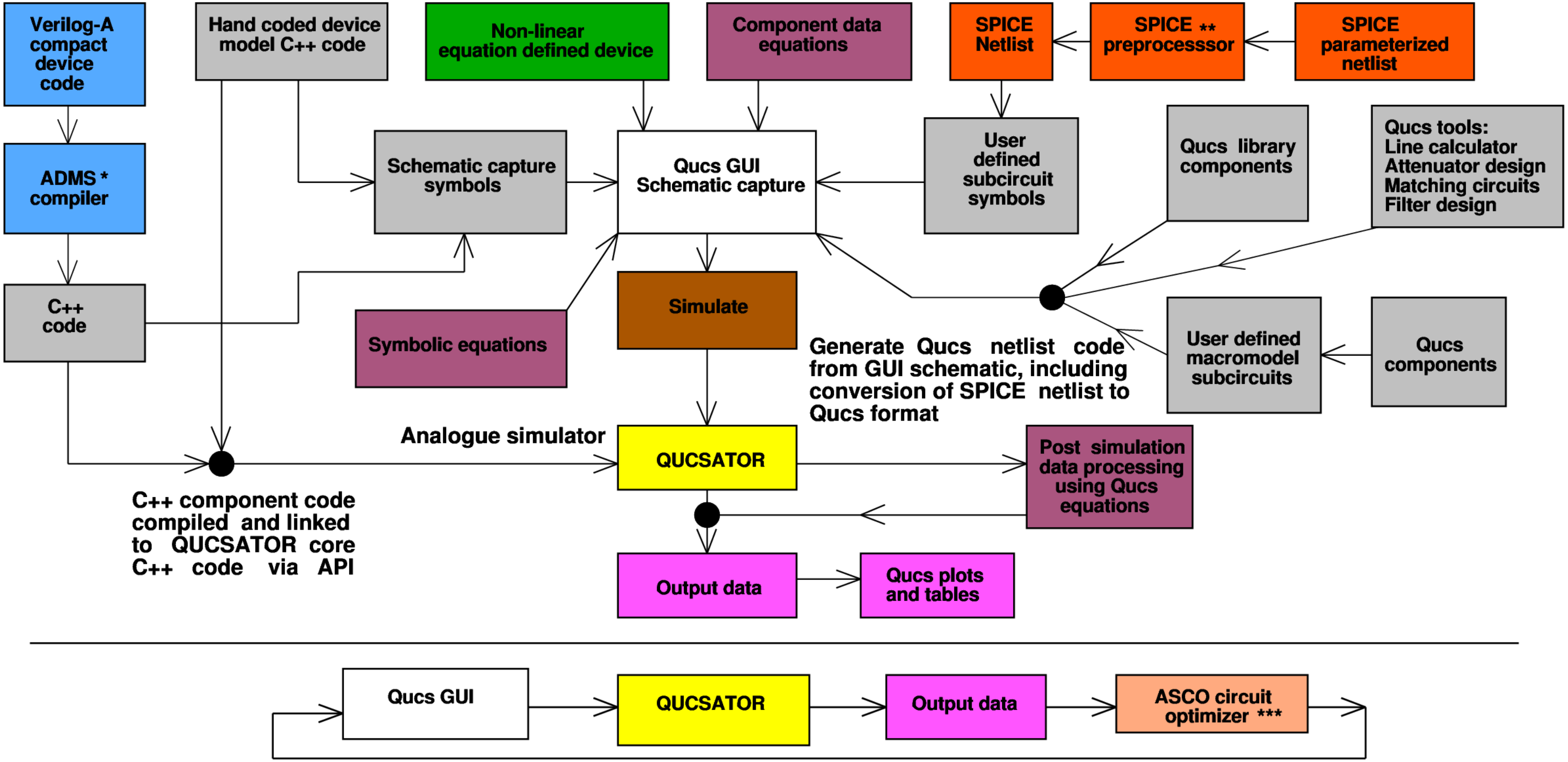
- Open source simulator developed by an international group of scientists and engineers
- Developed using GNU/Linux under the General Public License
- Qucs has been compiled and run on most of the popular operating systems, including GNU/Linux, Windows®, Solaris®, NetBSD, FreeBSD and MacOS®
- Qucs features a graphical user interface, based on Qt® from Trolltech® that supports schematic capture, analysis control, and simulation post-processing using equations
- Qucs graphical user interface also allows input of Verilog/VHDL digital circuits
- Qucs currently supports analogue analysis types: DC, AC, AC noise, S-parameter, S-parameter noise and transient
- ASCO (A SPICE circuit optimizer) is employed for circuit performance optimisation
- Compact device modeling is possible using the ADMS Verilog-A compiler
- Qucs uses FreeHDL and Icarus Verilog for digital simulation

Qucs: Roadmap and current development status

- Stage 5 - Design realization, production and verification: layout editor, Monte Carlo simulation and automated data acquisition/extraction
- Stage 4 - Implementation of industry standard device models: BSIM series, HICUM, MEXTRAM, EKV, VBIC
- Stage 3 - Support for more design and synthesis tools: attenuator design, Smith chart for noise and power matching, filter synthesis, improvements to data conversion tools, optimizer, transmission line calculator, device model and subcircuit library manager, text editor
- Stage 2 - Implementation of additional circuit analysis tools: EM field simulator, transient simulation using convolution for devices defined in the frequency domain, improvements to the GUI, large signal S-parameter simulation based on harmonic balance techniques, symbolic defined devices, digital simulation, Verilog-AMS interface
- Stage 1 - Set up a simple GUI and simulator: support for multiple languages, control support for standard simulations; S-Parameter, AC ,DC, Transient, harmonic balance, AC noise S-Parameter noise, create data visualisation diagrams, implement easy to use schematic editor

Legend: Green text; current feature or being worked on. Blue text; future work.

Qucs: Analogue circuit simulation and device modeling features

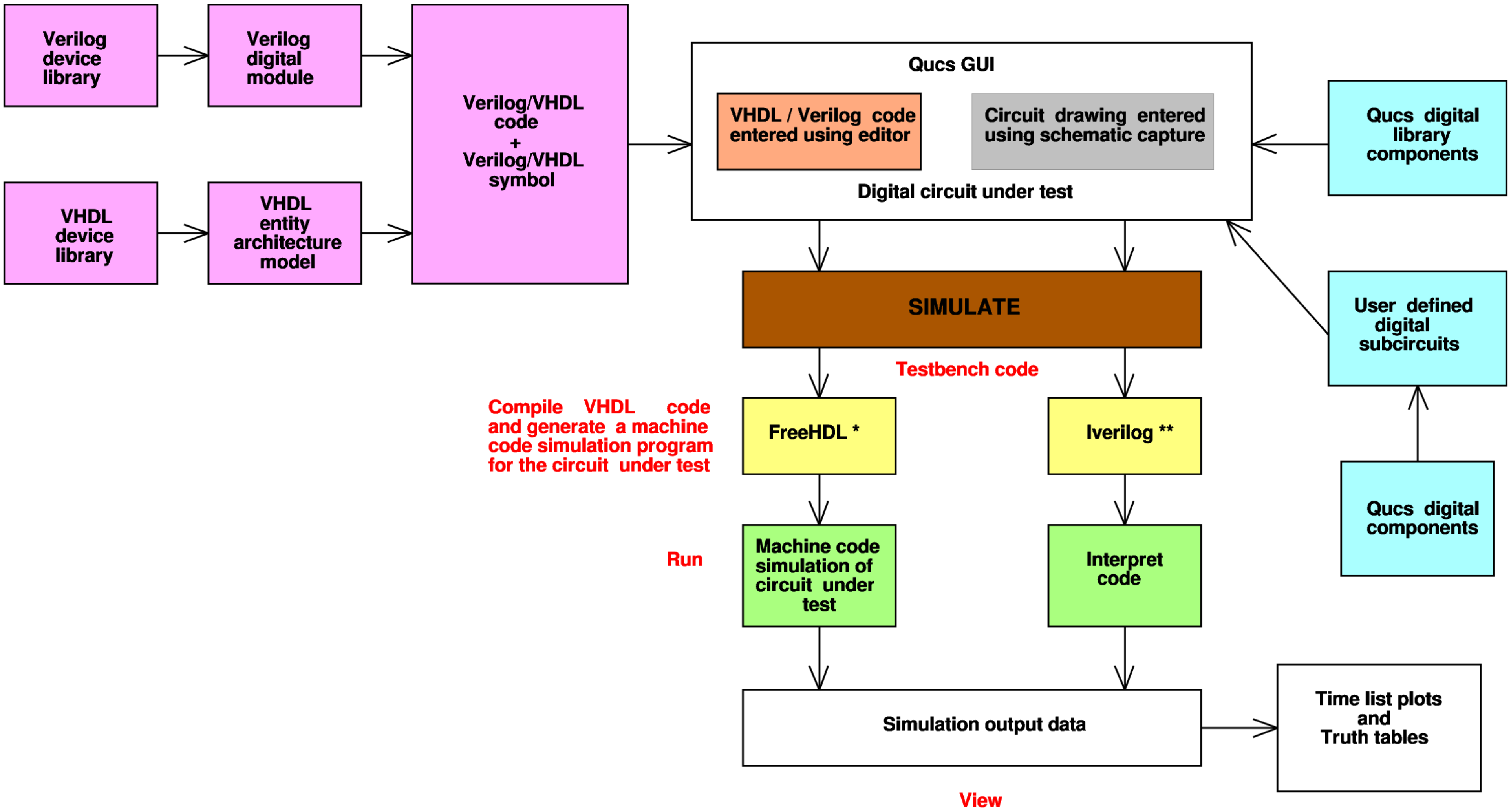


* ADMS – Automatic device model synthesizer, http://sourceforge.net/projects/mot-adms

** PS2SP SPICE PSpice to SPICE preprocessor, http://members.aon.at/fschmid7/

*** ASCO A SPICE circuit optimizer, http://asco.sourceforge.net/

Qucs: Digital circuit simulation

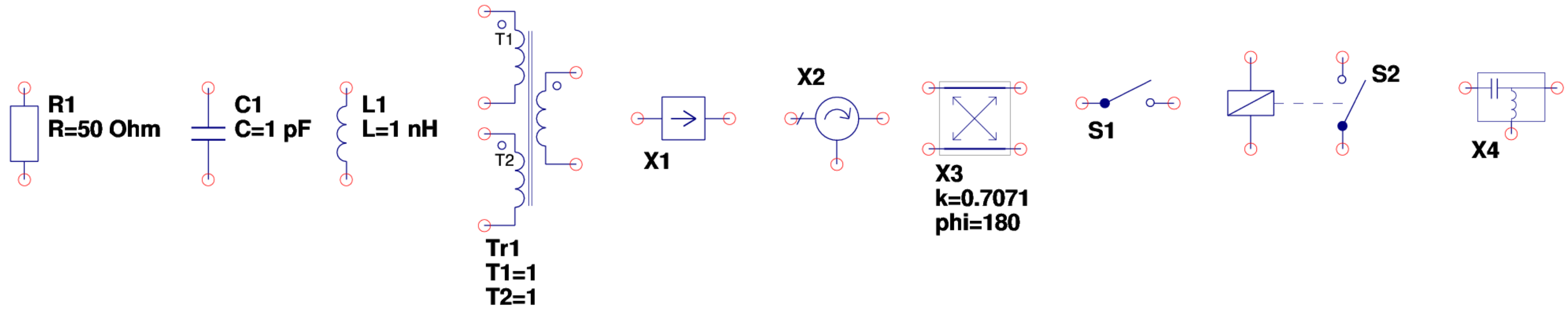


* FreeHDL, http://freehdl.seul.org/

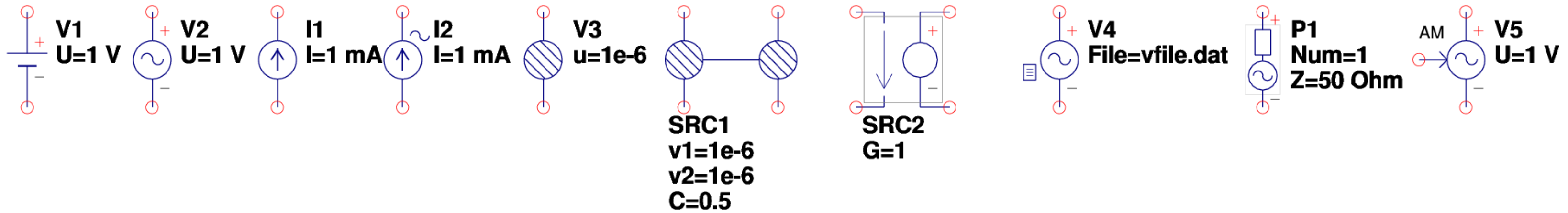
** Icarus Verilog, http://icarus.com/eda/verilog/

Qucs: Example component, device and source symbols

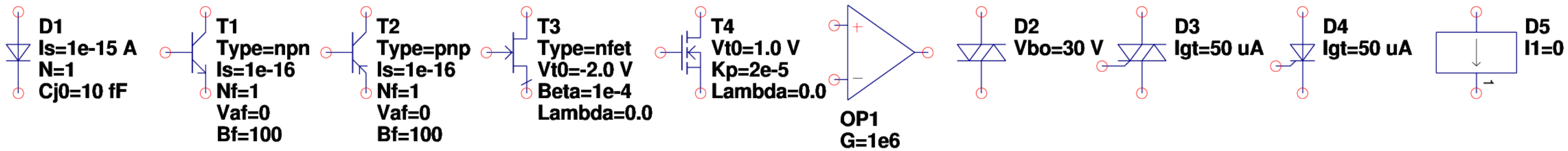
Lumped components



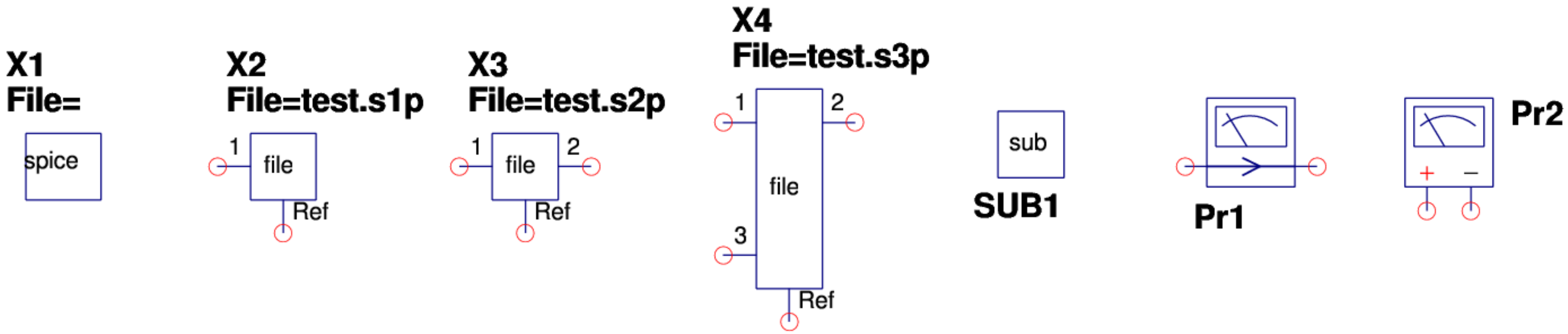
Sources



Non-linear devices



File components and probes

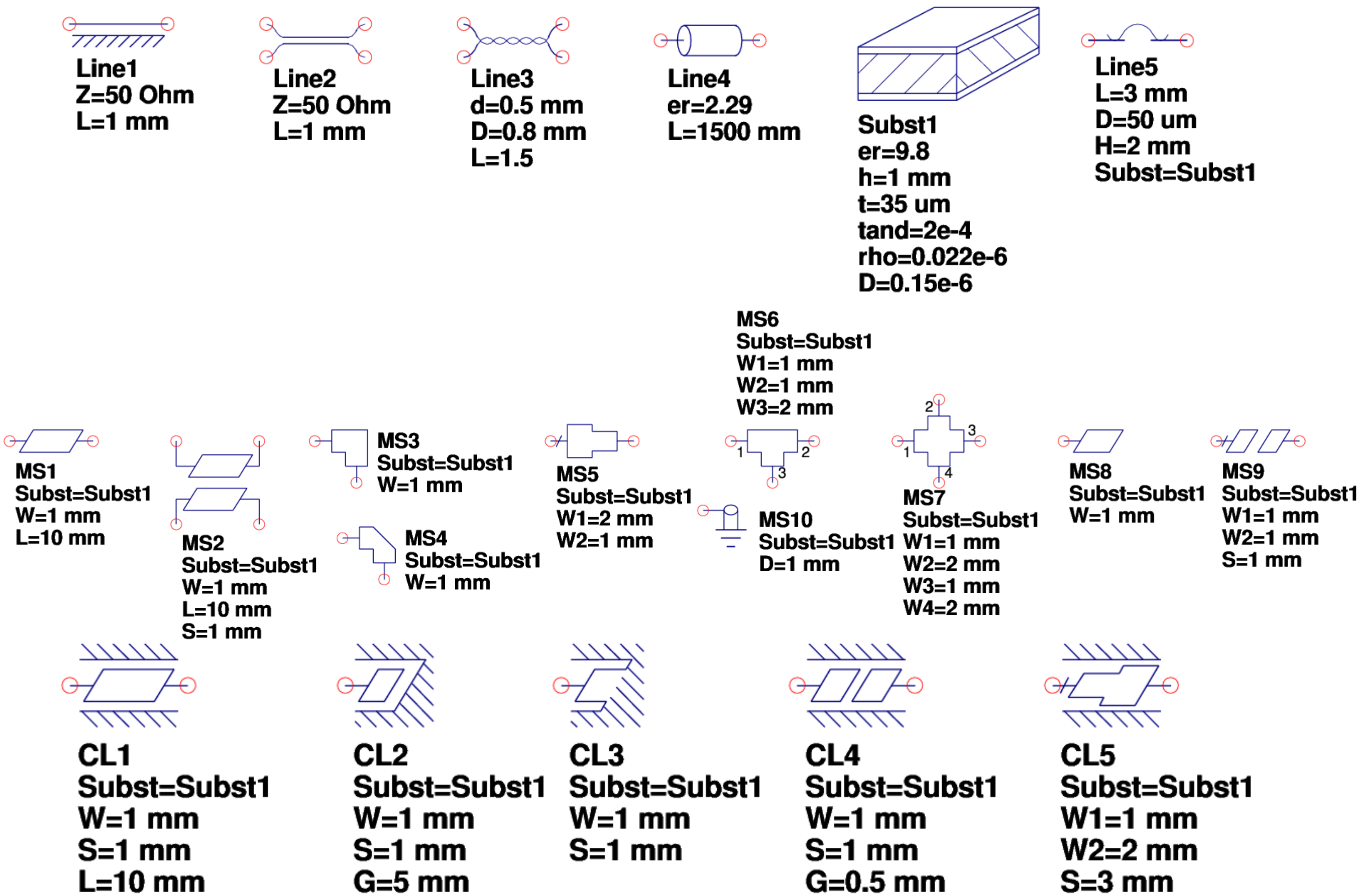


Qucs: Example transmission line, microstrip and coplanar components

Transmission lines

Microstrip components

Coplanar lines



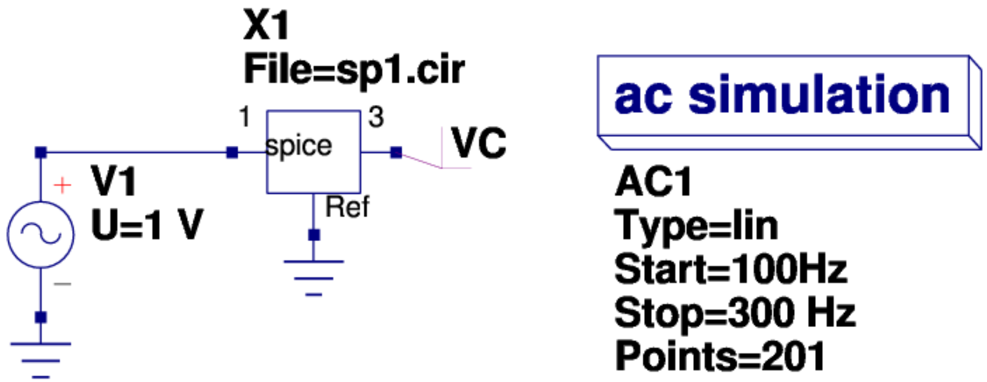
Qucs: Interface with SPICE+

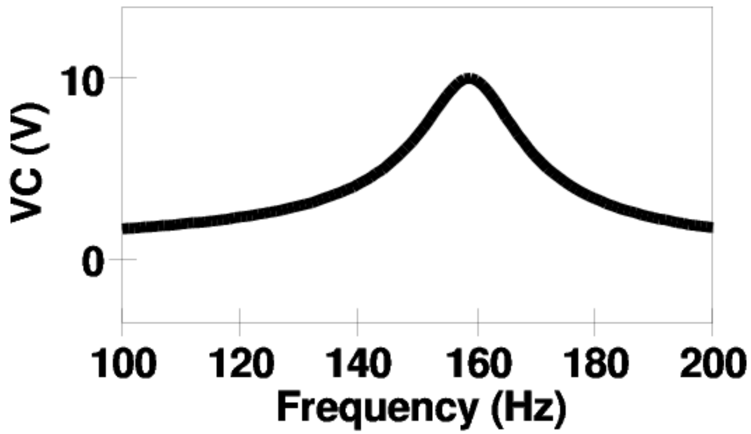
SPICE parameterized netlist ----> SPICE netlist ----> Qucs netlist

```
*RLC test circuit
.param Q=10 L=40m WN=1000
.param l1={L}
.param r1={Wn*L/Q}
.param c1={1.0/(L*WN*WN)}
L1 1 2 {l1}
R1 2 3 {r1}
C1 3 0 {c1}
```

```
*RLC test circuit
l1 1 2 0.04
r1 2 3 4
c1 3 0 2.5e-05
```

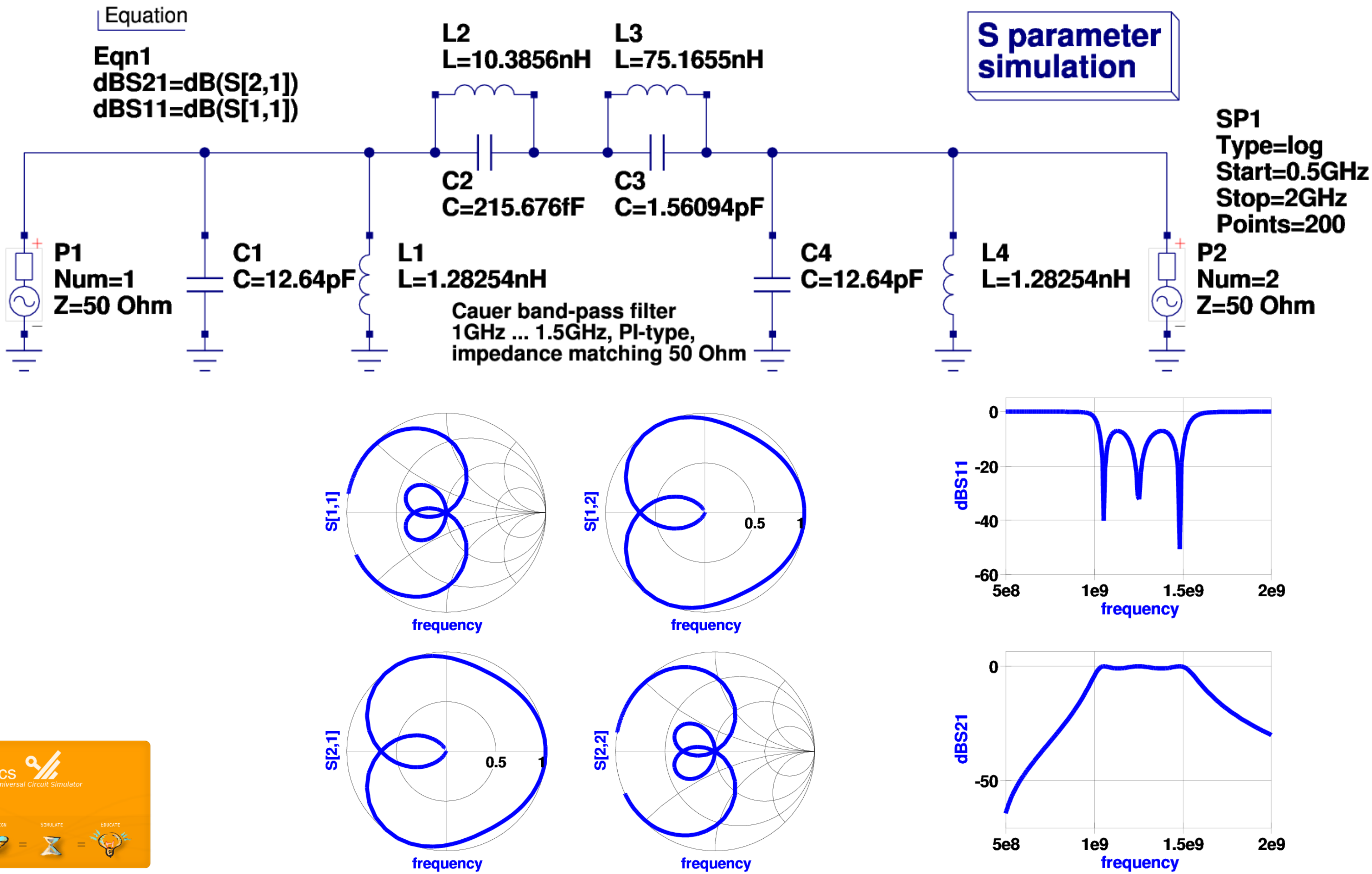
```
#Qucs 0.0.13
.Def: sp1_cir _net _net3 _ref
C:C1 _net3 _ref C=”2.5e-05”
R:R1 _net2  _net3 R=”4”
L:L1 _net _net2 L=”0.04”
.Def: End
Sub:X1 _net0 VC gnd type=”sp1_cir”
Vac:V1 _net0 gnd U=”1V” f=”100Hz” Phase=”0” Theta=”0”
.AC:AC1 type=”lin” Start=”100Hz” Stop=”300Hz” Points=”201”
```





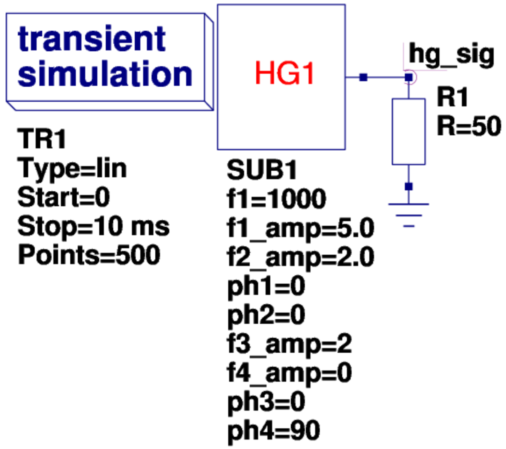
+ Brinson M., *Qucs: A Tutorial; Qucs simulation of SPICE netlists*, 2007,
http://qucs.sourceforge.net/docs.html

Qucs: Circuit design tools: Text editor; Filter Synthesis; Line Calculator; Matching Circuits; Attenuator Synthesis; Component Library



Qucs: Post simulation data processing – MATLAB®*/Octave** style equations

Equation blocks > data processing >Plots/Tables



Equation
```
Eqn1
ts=(max(time)-min(time))/length(time)
fs=1/ts
Adft=dft(hg_sig.Vt)
LAdft=length(hg_sig.Vt)
Amp2=2*Adft[1:(LAdfto2)-1]
LAdfto2=LAdft/2
Amp_squared=Adft[:LAdfto2]*conj(Adft[:LAdfto2])
Amp=sqrt(Amp_squared)
f_bin=linspace(1, LAdfto2, LAdfto2)
f=(f_bin-1)*fs/LAdft
PLAmp=PlotVs(2*Amp/LAdft,f)
PLPower=PlotVs(4*Amp*Amp/(LAdft*LAdft),f)
```

Simulation data: X.V, X.I, X.v, X.i, X.vn, X.in, X.Vt, X.It

Constants: i, j, pi, e, kB, q

Number suffixes: E, P, T, G, M, k ,m, u, n, p, f, a

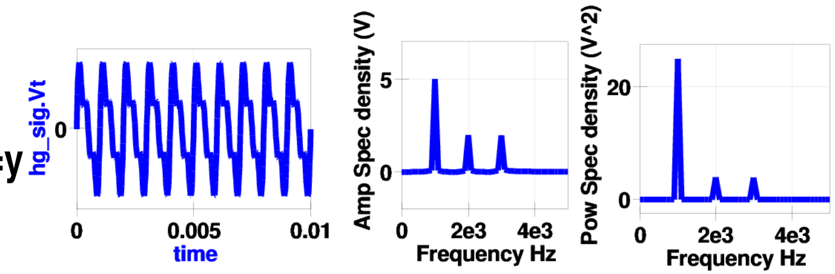
Immediate: 2.5, 1.4+j5.1, [1, 3, 4, 5, 7], [11, 12; 21, 22]

Matrices: M, M[2,3], M[:,3]

Ranges: Lo:Hi, :Hi, Lo:, :

Arithmetic operators: +x, -x, x+y, x-y, x*y, x/y, x%y, x^y

Logical operators: !x, x&&y, x||y, x^^y, x?y:z, x==y, x!=y, x<y, x<=y,x>y, x>=y



Functions

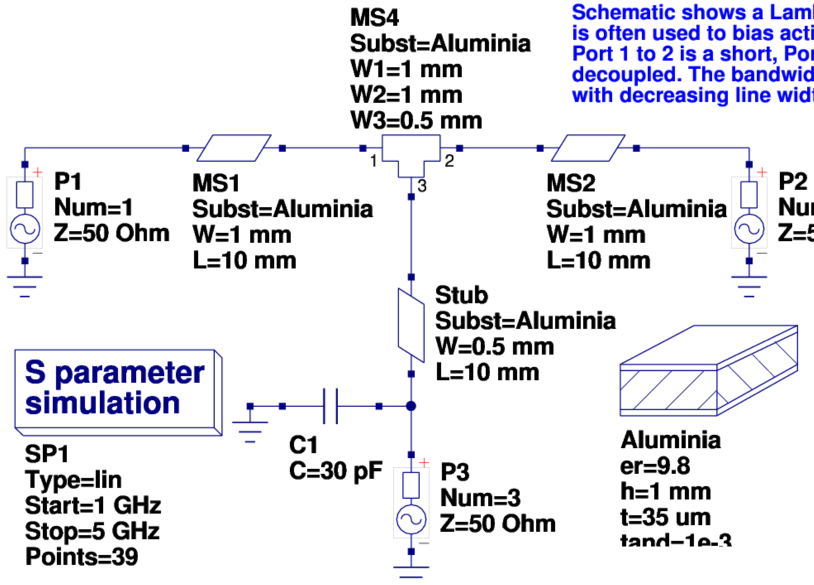
abs adjoint angle arccos arccosec arccot arcosech arcosh arcoth arcsec arcsin arctan arg arsech arsinh artanh avg besseli0 besselj bessely ceil conj cos cosec cosech cosh cot coth cumavg cumprod cumsum dB dbm dbm2w deg2rad det dft diff erf erfc erfcinv erfinv exp eye fft fix floor Freq2Time GaCircle 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

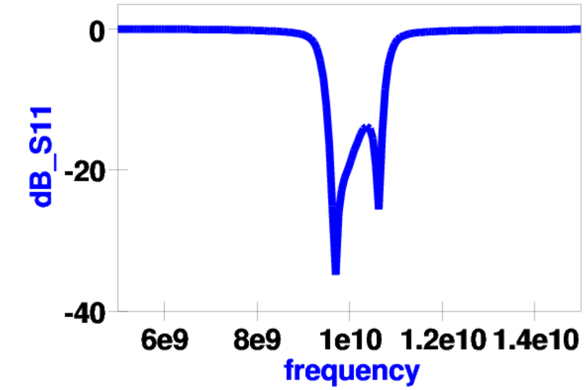
* MATLAB, Mathworks, http://www.mathworks.com/

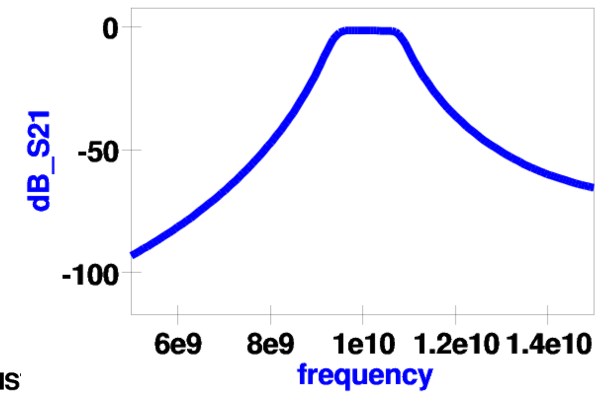
** Octave, http://www.gnu.org/software/octave/

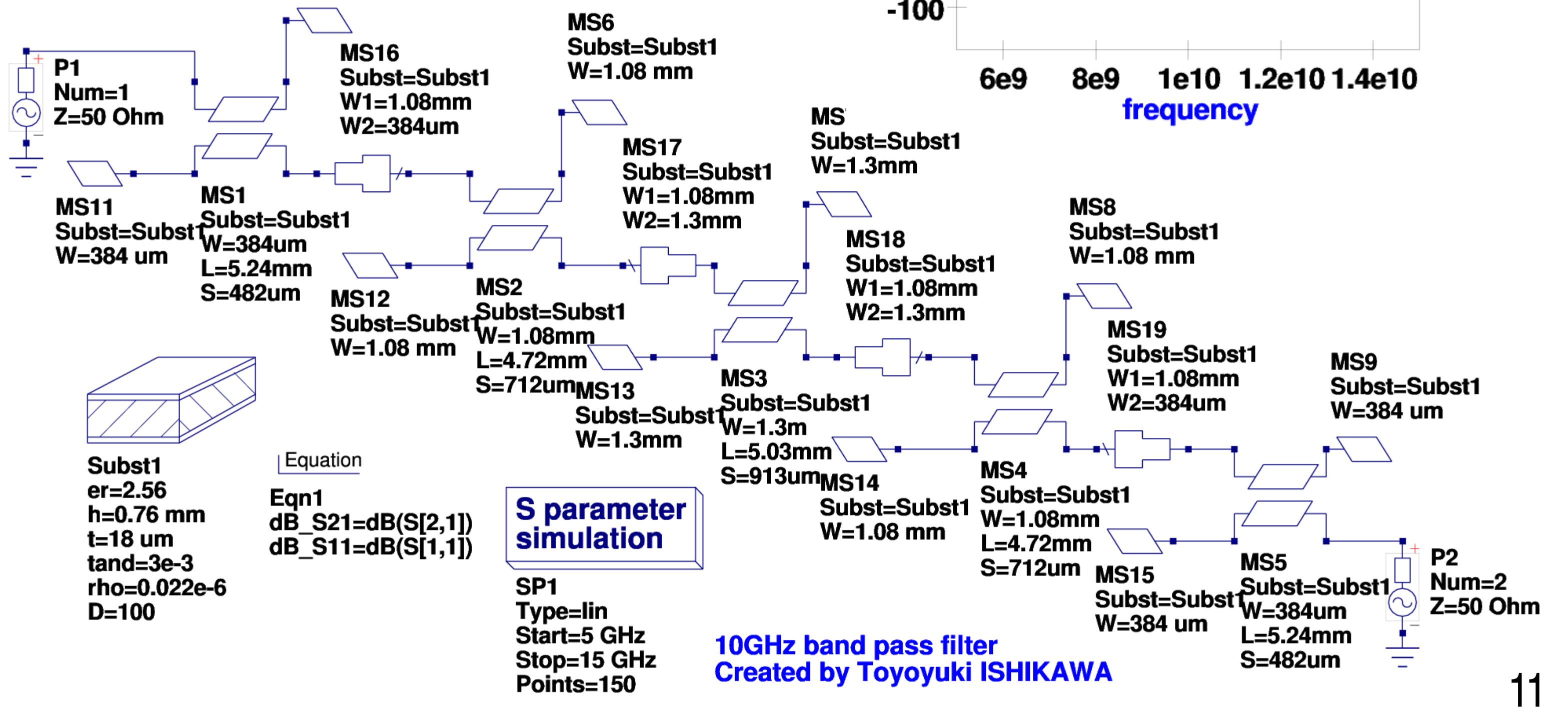
Qucs: RF examples 1. - Microstrip designs

Schematic shows a Lambda/4 stub that is often used to bias active elements. Port 1 to 2 is a short, Port 1 and 3 is decoupled. The bandwidth increases with decreasing line width of "Stub".









10GHz band pass filter
Created by Toyoyuki ISHIKAWA

Qucs: RF examples 2. - BJT and FET noise

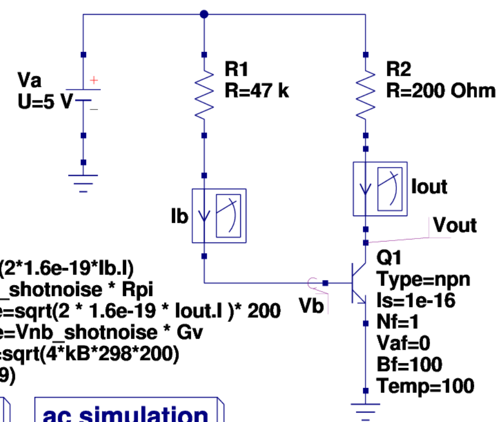
Simple Flicker Noise Simulation

Equation

BTJ_Biasing
gm=Q1.gm
Rpi=1 / Q1.gpi
Gv=gm * 200

Equation

Noise
Inb_shotnoise=sqrt(2*1.6e-19*Ib.I)
Vnb_shotnoise=Inb_shotnoise * Rpi
Vnout_Ic_shotnoise=sqrt(2 * 1.6e-19 * Iout.I)* 200
Vnout_Ib_shotnoise=Vnb_shotnoise * Gv
Vnout_R2_thermal=sqrt(4*kB*298*200)
fc=0.1e-15/(2*1.6e-19)



dc simulation

DC1

ac simulation

AC1
Type=log
Start=1 Hz
Stop=10 GHz
Points=400
Noise=yes

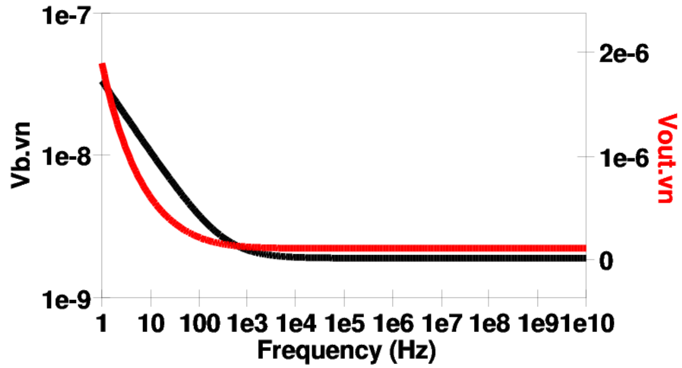
To display Flicker Noise plot a cartesian graph of Vb.vn or Vout.vn against frequency.

Note that the circuit noise is dominated by the BJT base current noise (shot + flicker). Moreover, the collector shot noise (Vnout-Ic-shot) is at least one order of magnitude lesser and R1 and R2 (Vnout-R2-thermal) noises are even smaller.

Often we don't know the Kf (Flicker coeff.) from a manufacturers datasheet. However, if the noise corner frequency is known an approximation for Kf can be calculated from:

$$Kf = 2 \cdot q \cdot fc \cdot Ib^{Af-1}$$

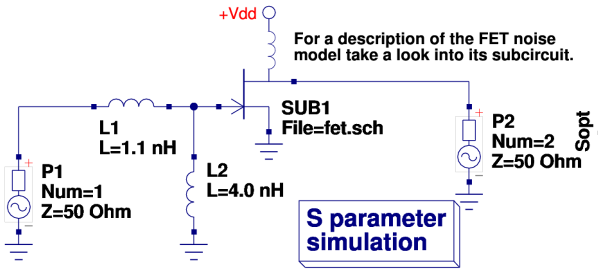
where:
- Af is the flicker current exponent (typically 1 ~ 2),
- q is the electron charge,
- fc is the noise corner frequency, and
- Ib is the base current.



Turn on the noise simulation in the "Properties" tab of the AC simulation, otherwise the noise is not calculated.

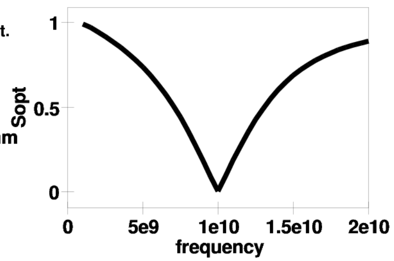
FET noise optimization

For a description of the FET noise model take a look into its subcircuit.

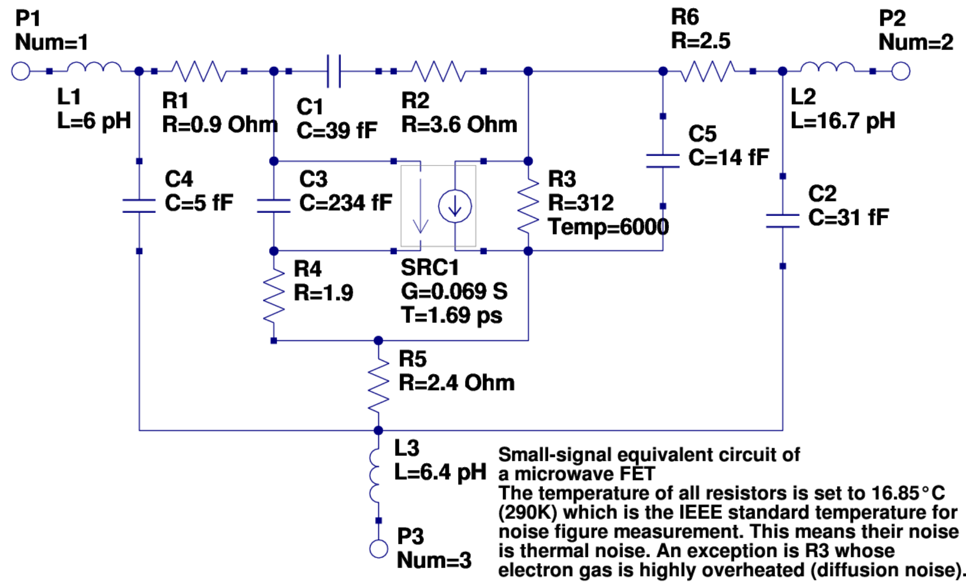


S parameter simulation

SP1
Type=lin
Start=1 GHz
Stop=20 GHz
Points=39
Noise=yes



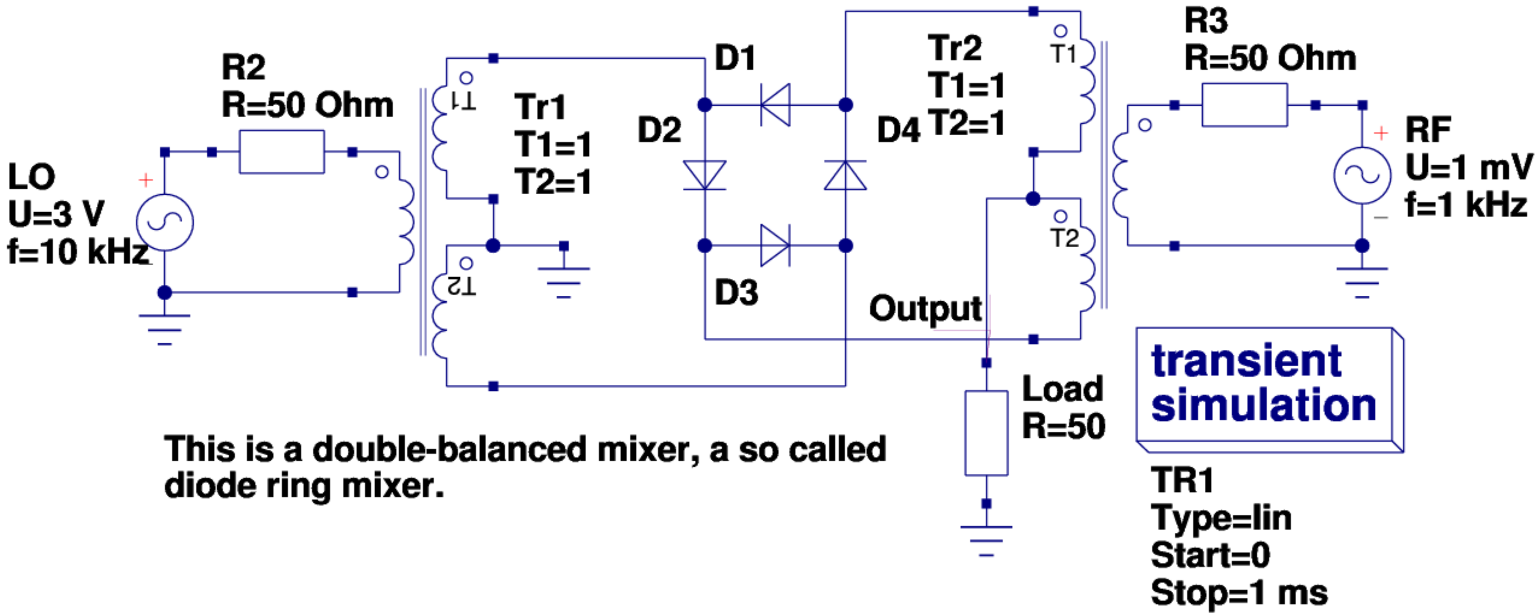
This amplifier is noise matched at the input for a frequency of 10GHz.
Thus, the noise optimum Sopt of the overall circuit is zero. (Note :- basic matching circuit)

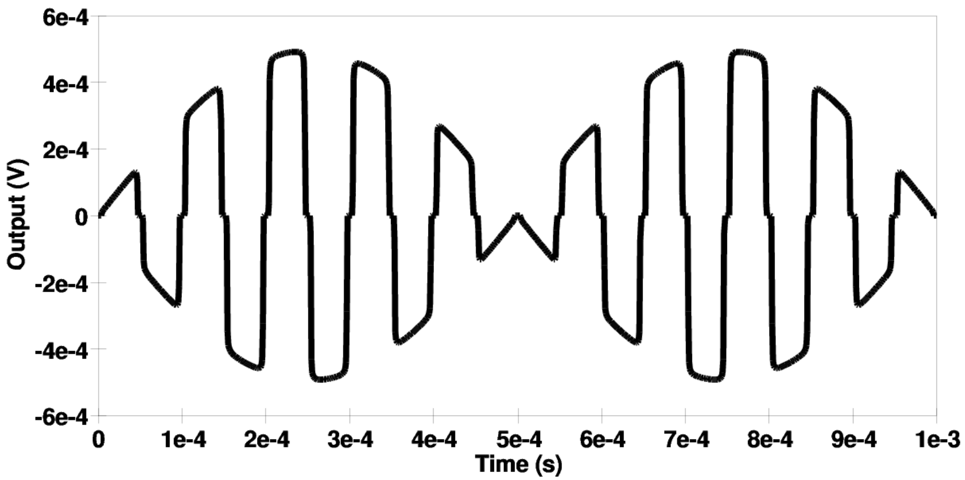


Small-signal equivalent circuit of a microwave FET
The temperature of all resistors is set to 16.85°C (290K) which is the IEEE standard temperature for noise figure measurement. This means their noise is thermal noise. An exception is R3 whose electron gas is highly overheated (diffusion noise).

Qucs: RF examples 3. - Double-balanced mixer

Balanced mixer





Qucs: Overview of component and circuit modeling capabilities

Hand crafted C/C++ models

Equation defined components – use of design equations

Subcircuit macromodels

Non-linear equation defined device (EDD) models

Verilog-A compact device and circuit macromodels

Qucs: Equation defined components – use of design equations

Qucs equation blocks can be used

as a design aid to calculate component

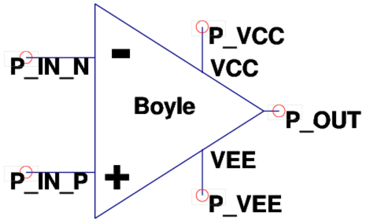
values at the start of a simulation sequence.

The order of the equations in a block is

not important .

Multiple blocks are combined by Qucs.

Adds a design element to subcircuits.

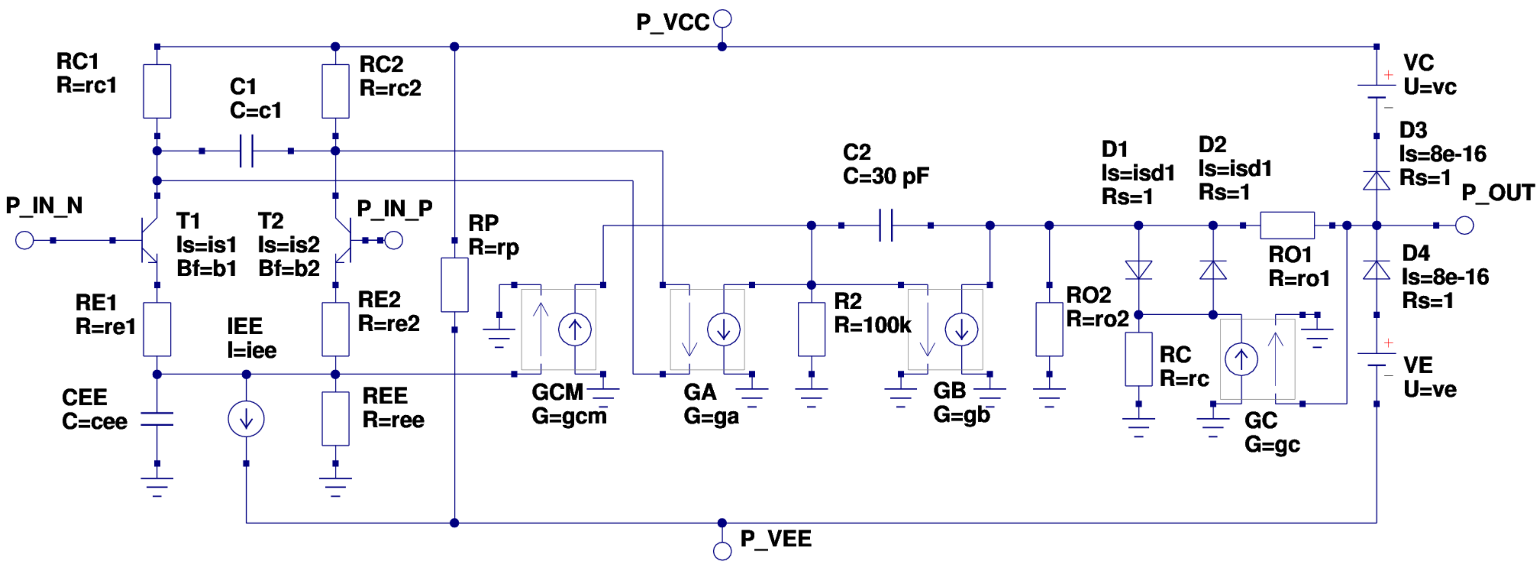



```
SUB
File=name
is=8e-16
vos=0.7e-3
c2=30e-12
r2=100e3
pswrt=0.625e6
nswrt=0.5e6
ios=20e-9
ib=80e-9
gbp=1e6
va=200
pm=70
cmrr=31622.8
avol=200e3
vsp=14.2
vsn=-13.5
vcc=15
vee=-15
iscp=25e-3
iscn=25e-3
pd=59.4e-3
ro1=76.8
ro2=489.2
```

```
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
```

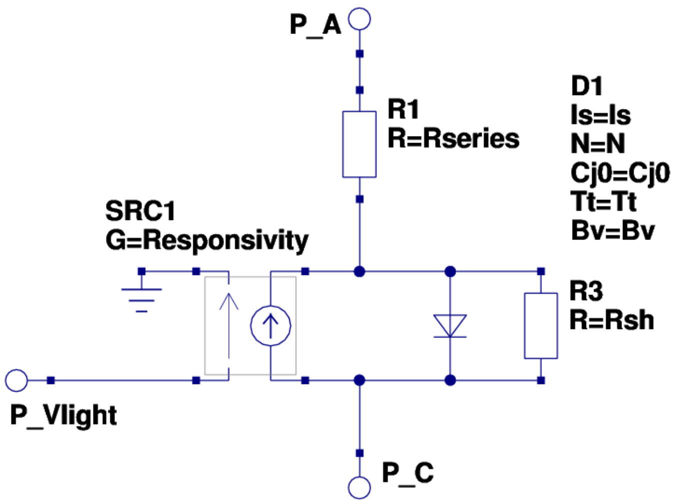
Boyle OP AMP

macromodel

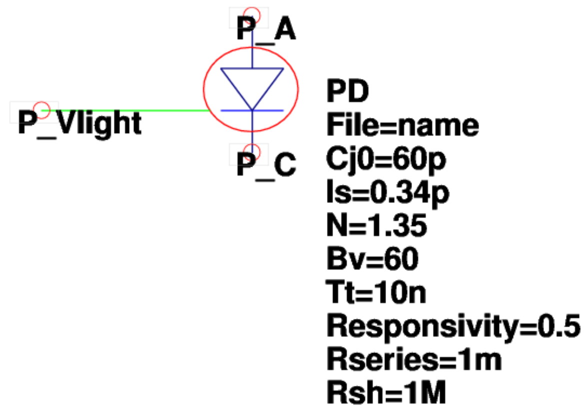


Qucs: Subcircuit macromodels

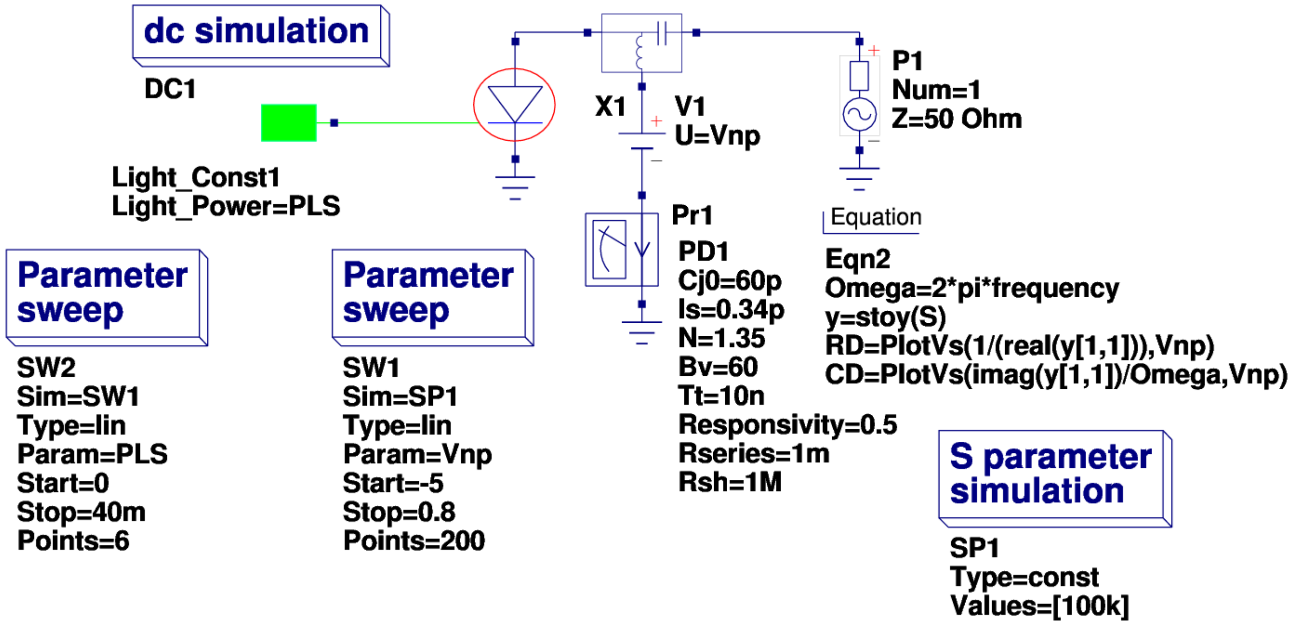
Photodiode subcircuit body

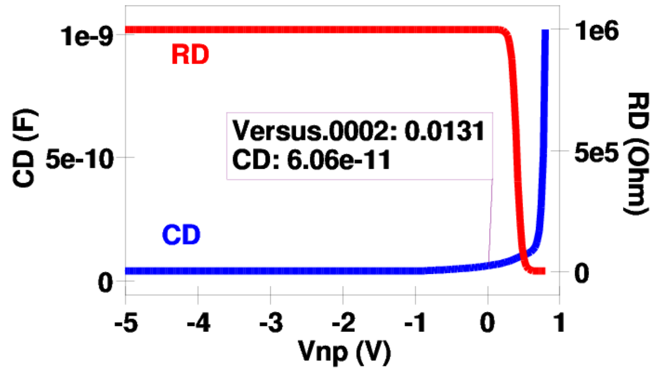


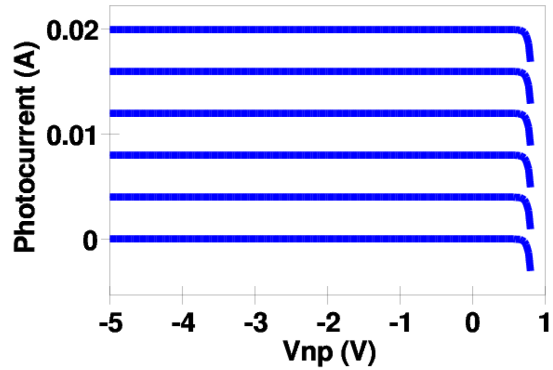
Photodiode symbol



Test circuit and simulation data

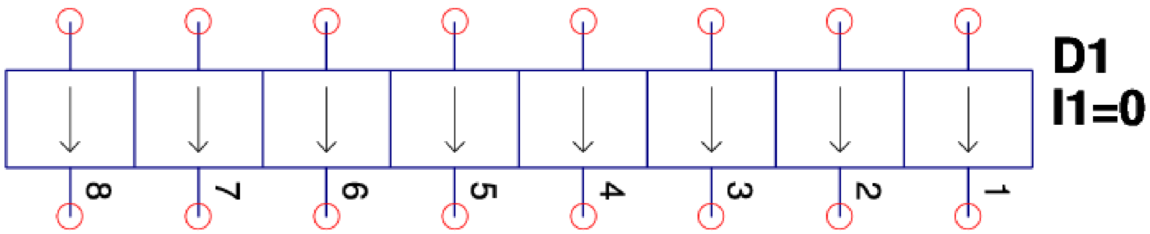






Green denotes light source and light bus

Qucs: Non-linear equation defined devices (EDD*)



$$I = I(V)$$

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

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

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

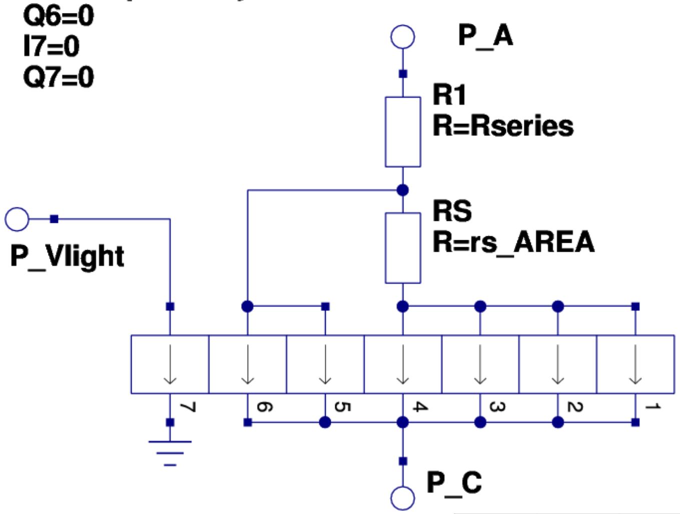
$$c = \frac{dQ(V)}{dV} + \frac{dQ(I)}{dI} \cdot g$$

- 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

Qucs: EDD photodiode 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

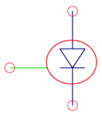


Equation

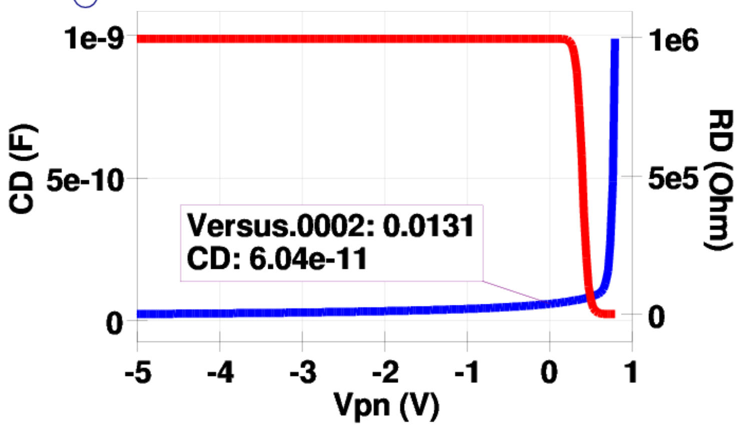
Eqn2
Cj0_T2=Cj0*(1+M*(400e-6*(T2-T1)-(Vj_T2-Vj)/Vj))
Vt_T2=(kB*T2/q)
rs_AREA=Rs/AREA
GMIN=1e-12
A=7.02e-4
B=1108
T1=Tnom+273.15
Vj_T2=(T2/T1)*Vj-(2*kB*T2/q)*ln((T2/T1)^1.5)-((T2/T1)*Eg_T1-Eg_T2)
Is_T2=Is*(T2/T1)^(Xti/N)*limexp((-(q*Eg_T1)/(kB*T2))*(1-T2/T1))
Eg_T1=Eg-A*T1*T1/(B+T1)
Eg_T2=Eg-A*T2*T2/(B+T2)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

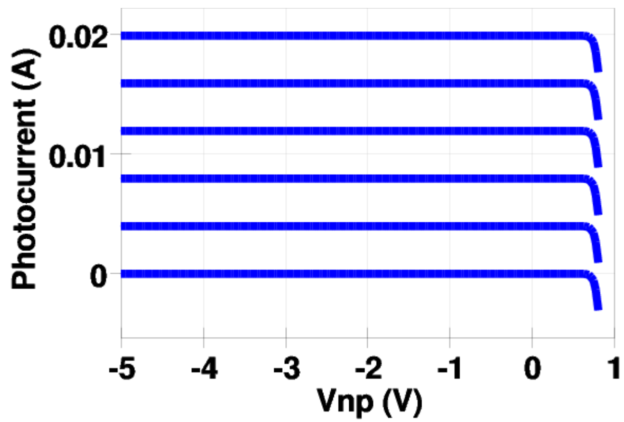
Equation

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





Qucs: Verilog-A compact device and circuit macromodeling interface*

- The ADMS compiler translates Verilog-A models into a structured XML tree. Qucs/ADMS uses this tree to generate ready-to-compile C++ code
- Generated C++ code is specific to the Qucs API
- The process of transforming Verilog-A model code into C++ is performed by a command line script:

  $admsXml <device.va> -e <interface-1.xml> -e <interface-2.xml>

- Qucs release 0.0.13 is distributed with the following XML scripts:

  qucsMODULEcore.xml ; creates simulator C++ code for a Verilog-A device model

  qucsMODULEdefs.xml ; creates device parameter descriptions

  qucsMODULEgui.xml ; creates a model GUI interface

  qucsVersion.xml ; a basic library

  analoguefunction.xml ; creates analogue function code

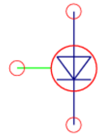
- Qucs has standardised on Verilog-A for compact device and circuit macromodel development. EDD models often being employed as prototypes for Verilog-A models. Recent work has improved the Qucs/ADMS interface making compiling and linking of Verilog-A models to Qucs more straightforward.

* Jahn S. and Parruite H., *Qucs: A description; Verilog-AMS interface*, 2006, http://qucs.sourceforge.net/docs.html

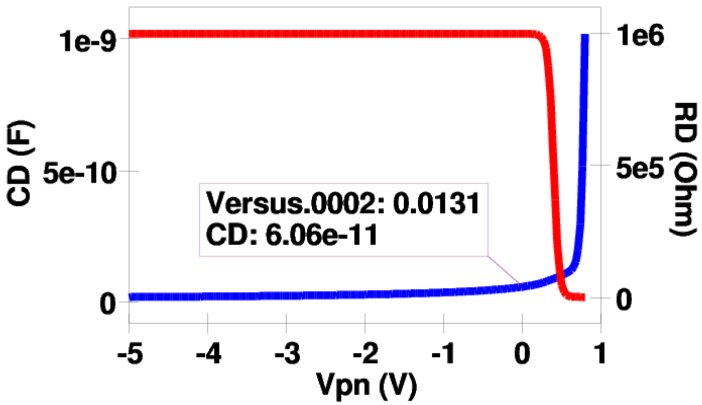
Qucs: Verilog-A compact device model for a photodiode

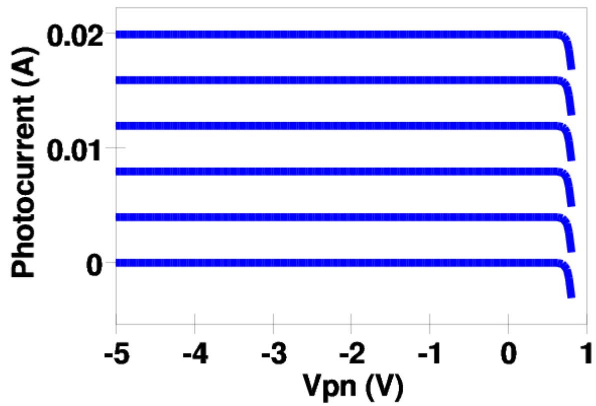
Verilog-A code fragment

```
analog begin
A=7.02e-4;  B=1108.0;  GMIN=1e-12;  T1=Tnom+273.15;  T2=$temperature;  Tr=T2/T1;  con1=pow(Tr, 1.5);
F1=(Vj/(1-M))*(1-pow((1-Fc), (1-M)));   F2=pow((1-Fc), (1+M));   F3=1-Fc*(1+M);    Rs_Area=Rs/Area;
Eg_T1=Eg-A*T1*T1/(B+T1);    Eg_T2=Eg-A*T2*T2/(B+T2);
Vt_T2=$vt;    Vj_T2=(Tr*Vj)-(2*Vt_T2*ln(con1)) - ( Tr*Eg_T1-Eg_T2);    Cj0_T2=Cj0*(1+M*(400e-6*(T2-T1)-(Vj_T2-Vj)/Vj));
Is_T2=Is*pow( (T2/T1), (Xti/N))*limexp(-(`P_Q*Eg_T1)*(1-T2/T1)/(`P_K*T2));
I1 = (V(pdb4) > -5.0*N*Vt_T2) ? Area*Is_T2*( limexp(V(pdb4)/(N*Vt_T2)) -1.0) + V(pdb4)*GMIN : 0;
I2 = ( (-Bv < V(pdb4)) && (V(pdb4) < -5.0*N*Vt_T2) ) ? -Area*Is_T2 +V(pdb4)*GMIN : 0;
I3 = (V(pdb4) == -Bv)  ? -Ibv : 0;  I4 = (V(pdb4) < -Bv )  ? -Area*Is_T2*(limexp(-(Bv+V(pdb4))/Vt_T2)-1.0+Bv/Vt_T2) : 0;
Q1 =  (V(pdb4) < Fc*Vj)  ? Area*(Cj0_T2*Vj_T2/(1-M))*(1 - pow( (1-V(pdb4)/Vj_T2), (1-M) ) )  : 0;
Q2 =  (V(pdb4) >= Fc*Vj) ? Area*Cj0_T2*(F1+(1/F2)*(F3*(V(pdb4)-Fc*Vj_T2)+(M/(2.0*Vj_T2))*(V(pdb4)*V(pdb4)-Fc*Fc*Vj_T2*Vj_T2))) : 0;
Id=I1+I2+I3+I4;   I(pdb4)  <+ Id;   I(pdb4)  <+ ddt(Q1+Q2);   I(pdb4)  <+ ddt(Tt*Id);   I(pdb4)  <+ V(pdb4)/Rsh;
I(pdb1)  <+ V(pdb1)/Rseries;   I(pdb2)  <+ V(pdb2)/Rs_Area;   I(Light) <+ V(Light)/1e10; I(pdb3)  <+ -V(Light)*Responsivity;
end
endmodule
```

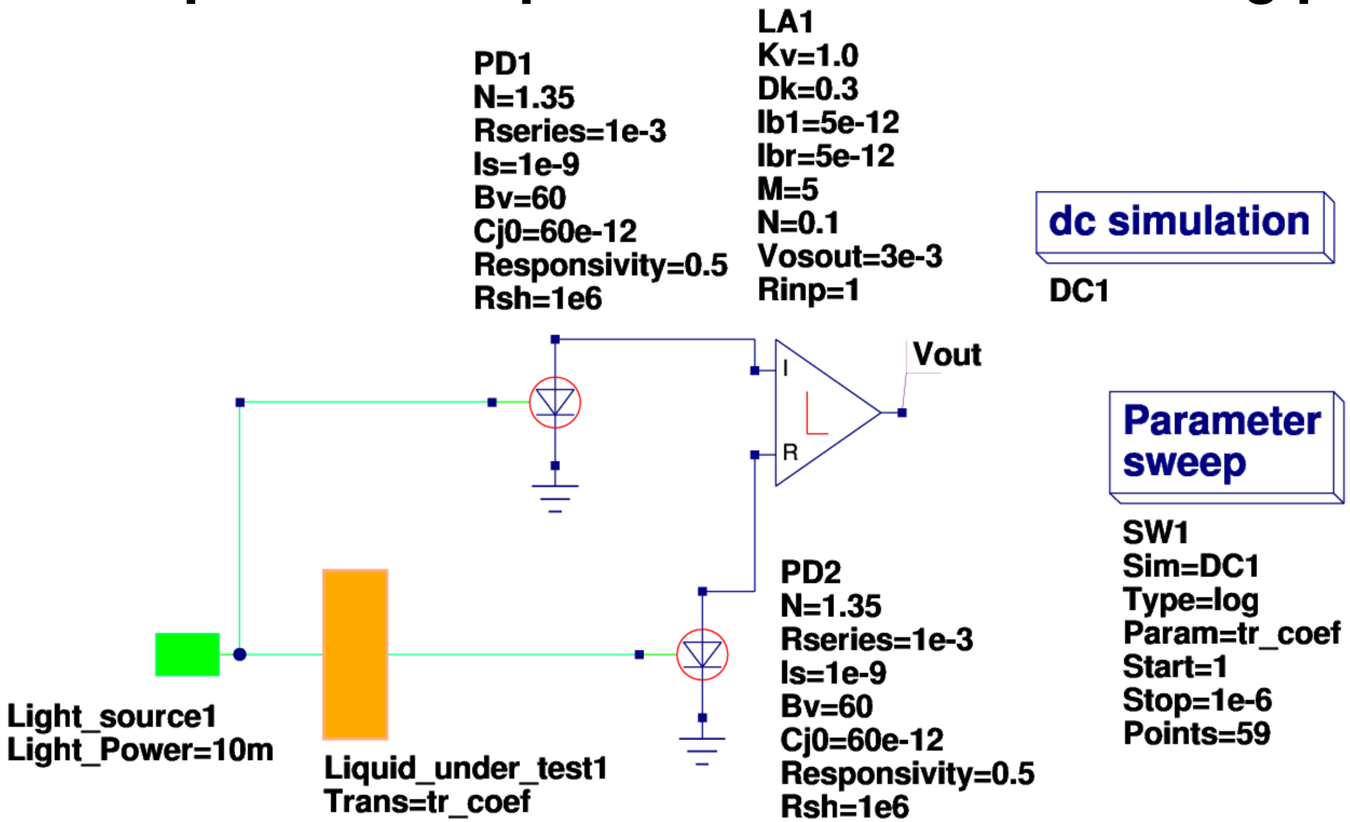


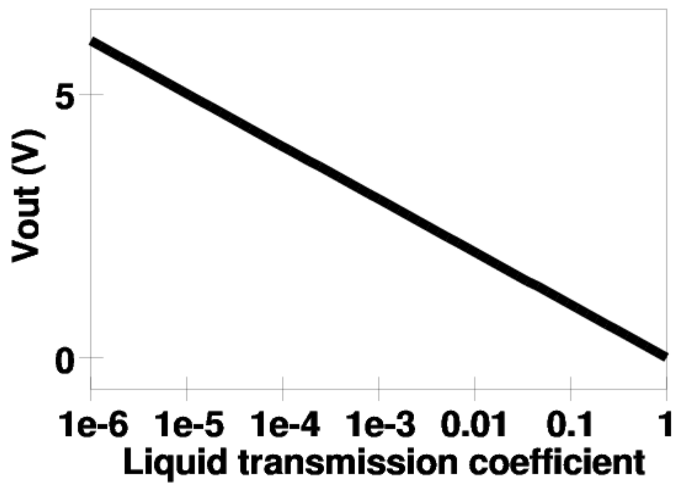
Verilog-A and EDD equations identical





Qucs: Optical absorption measurements using photodiodes and a log amp





Log Amp

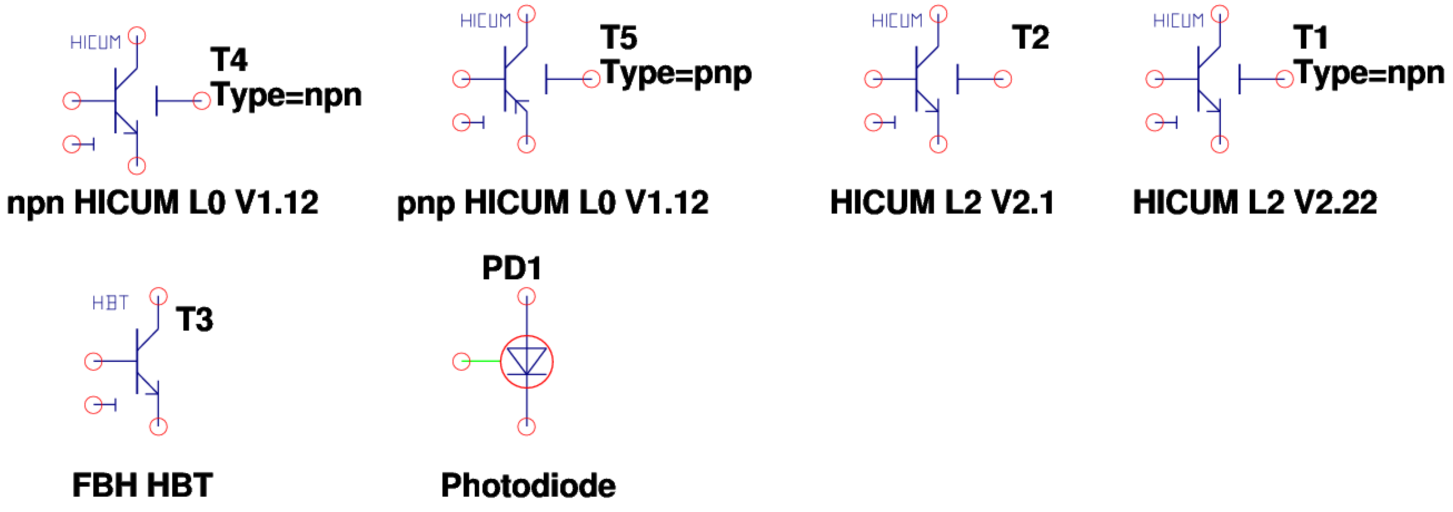
$$Vout = K_v \cdot (1 \pm \Delta K) \cdot \log\left(\frac{I_i - I_{b1}}{I_r - I_{br}}\right) \pm 2 \cdot K_v \cdot N \cdot m \pm Vosout$$

Green denotes light source and light path

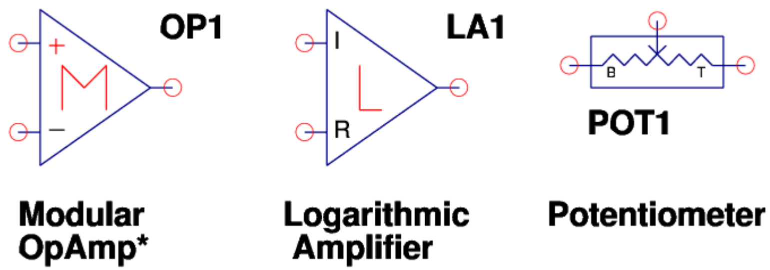
Qucs: Verilog-A compact device models and circuit macromodels

Qucs 0.0.13/0.0.14

Compact device models



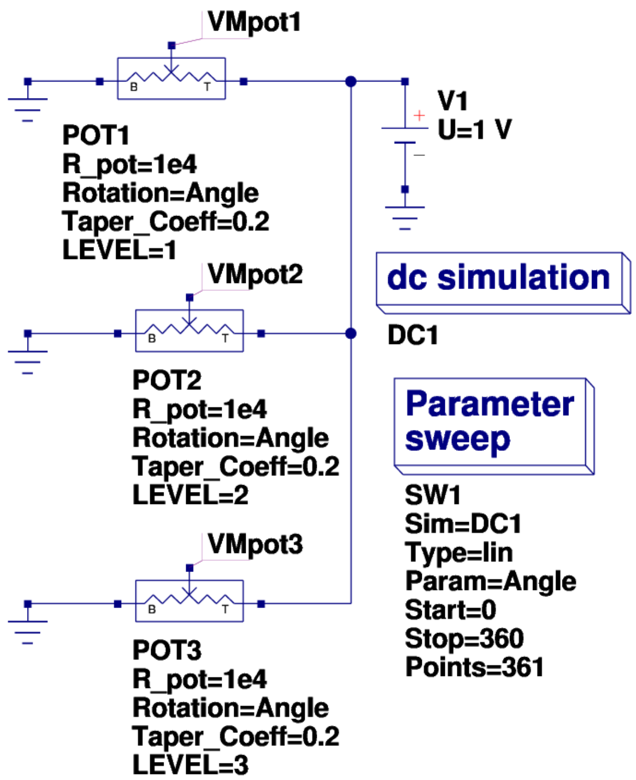
Compact macromodels

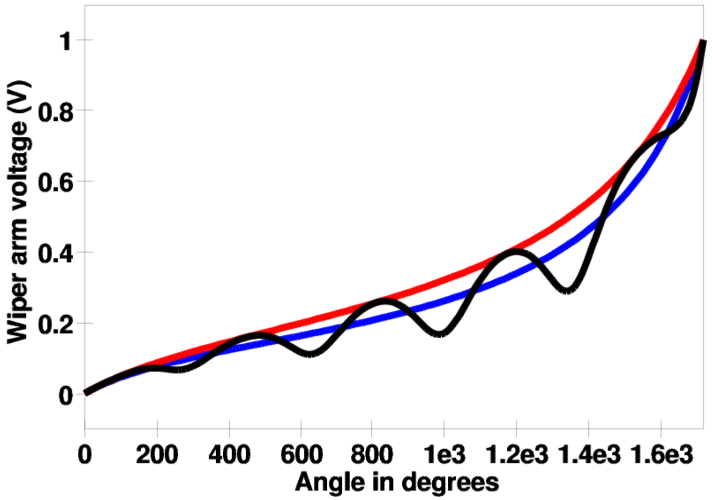


* Brinson M. E. and Faulkner D.J., (1994), *Modular SPICE macromodel for operational amplifiers,* IEE Proc.-Circuits Devices Sys., 141, 417-420

Qucs: Verilog-A compact macromodel of a resistive potentiometer*

Parameters	Description	Unit	Default
Rpot	Nominal device resistance	Ω	1e4
Rotation	Wiper arm rotation	degrees	120
Taper_coeff	Resistive law taper coeff.		0
LEVEL	Device type selector		1
Max_rotation	Maximum wiper rotation		240
Conformity	Conformity error	%	0.2
Linearity	Linearity error	%	0.2

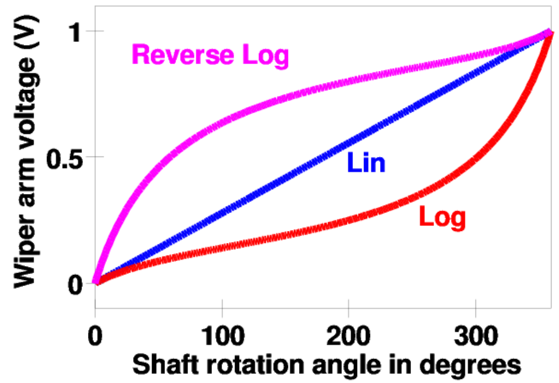




Blue: conformity = 0%, Linearity = 0%

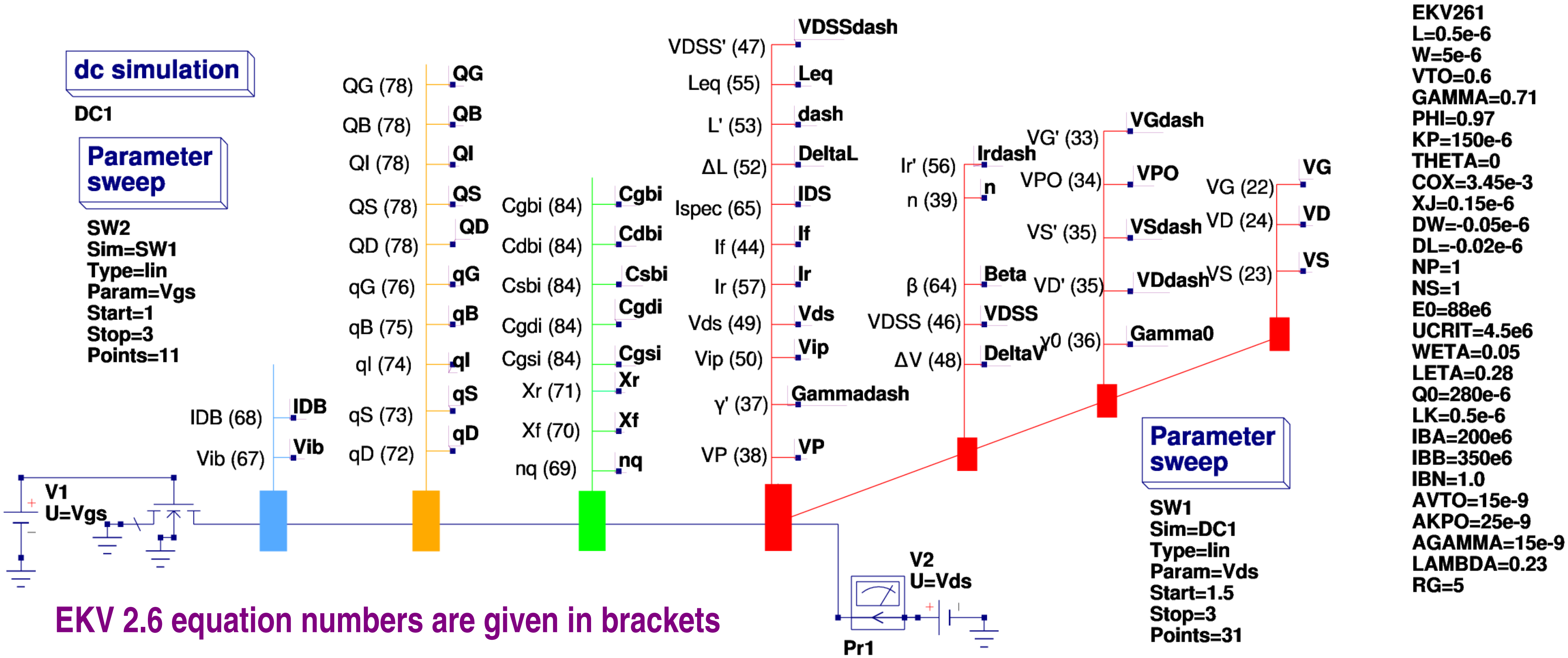
Red: conformity = 10%, Linearity = 0%

Black: conformity = 0%, Linearity = 10%



* Brinson M., *Qucs: A Report; Verilog-A macromodel for resistive potentiometers*, 2008, http://qucs.sourceforge.net/docs.html

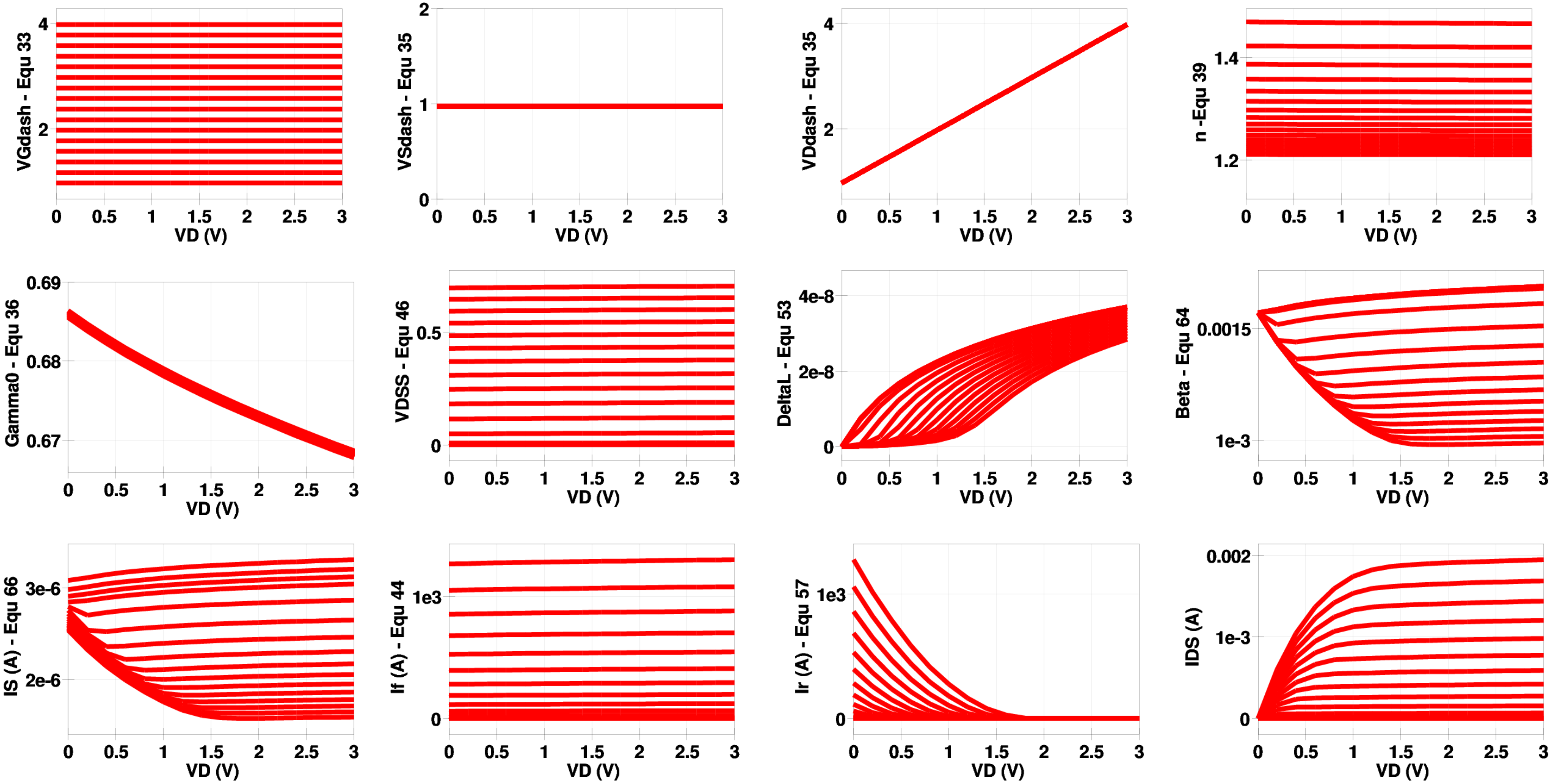
Qucs: EPFL-EKV 2.6* MOSFET EDD macromodel with equation monitoring bus



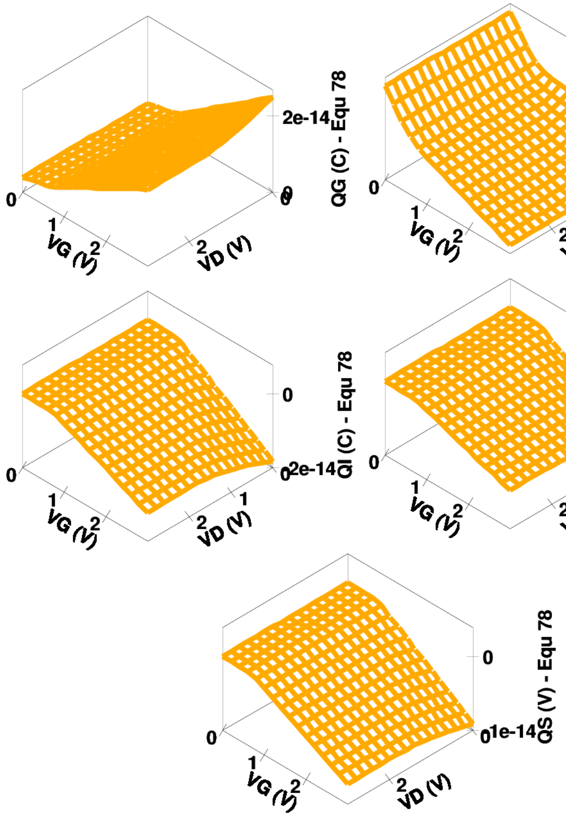
The schematic is characterized by two dominant features: firstly a standard symbol for a MOSFET and secondly a diagnostic bus which outputs model properties. The diagnostic bus comprises four sections: (1) a long and short channel device data bus [red], (2) a capacitance bus [green], (3) a charge data bus [orange] and (4) an ionization current data bus [blue].

* Matthias Bucher *et al*, *The EPL-EKV MOSFET model equations for simulation*, Technical Report, Revision II, July 1998, Electronics Laboratories, EPFL, Switzerland

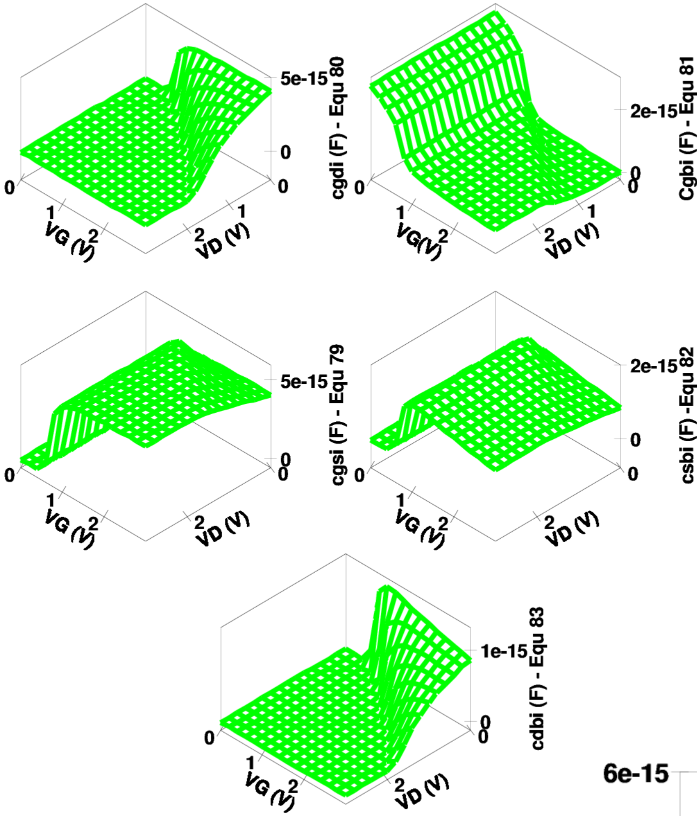
Qucs: Typical EPFL-EKV 2.6 MOSFET I/V characteristics



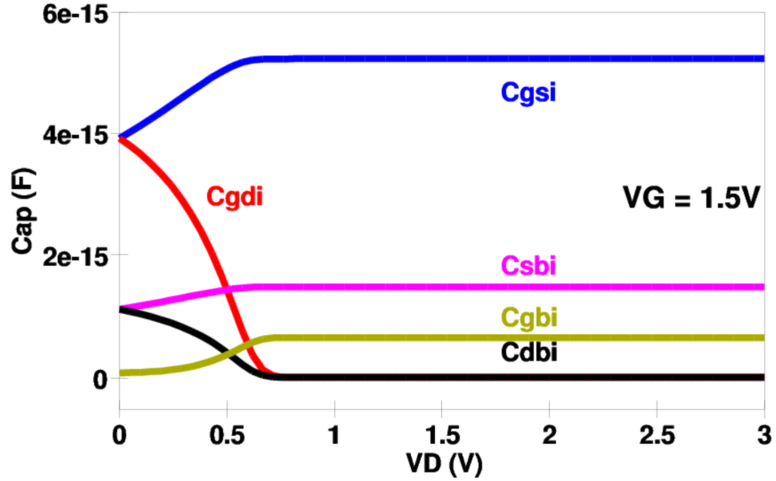
Qucs: Typical EPL-EKV 2.6 MOSFET charge and capacitance plots



Charge



Capacitance



Qucs: Future Developments

Short term: Next release (0.0.14) around June/July 2008

- More hand crafted C/C++ models; Diac, Thyristor (SCR) and Triac
- Improved PlotVs facility
- Sub- and super-scripts in graphical text paintings
- Multi-port equation defined RF device (RF-EDD): S, Y, Z parameters available
- Two-port equation defined RF device: H-, G-, T- and A- parameters available
- More Verilog-A models: including Log Amp, potentiometer and photodiode
- ?

Medium term: Work started but not complete

- Porting Qucs GUI from Qt3 to Qt4
- Initial planning for addition of an EM field simulator* to Qucs
- ?

Long term: More sections of the Qucs roadmap to be implemented

- See roadmap and Qucs todo lists
- ?

* See *EM field simulator blog* by Fredy Konig, 2008, http://emtheory.blogspot.com