SPICE to QucsStudio via Qucs: An international project to develop a freely available GNU Public Licence circuit simulator with compact device modelling tools, data processing capabilities, manufacturing features and an analogue/RF design environment for engineers.

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

- Background: Qucs and QucsStudio development road maps
- Circuit simulation: SPICE and Qucs forms of simulation
- Device and circuit modelling with subcircuits, macromodels, equation-defined devices and Verilog-A compiled models
- QucsStudio compact semiconductor modelling extensions
- The development of a QucsStudio Verilog-A MESFET model
- Simulation output data processing with Octave
- QucsStudio system simulation
- QucsStudio – future directions
- Summary
- References

Presented at the MOS-AK/GSA Compact Modelling Workshop, March 16-18 March 2012 at Jaypee Institute of Information Technology (JIIT), India
<table>
<thead>
<tr>
<th>Date</th>
<th>Release</th>
<th>Notes [Highlights]</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dec. 2003</td>
<td>0.0.1</td>
<td>First public version of simulator - very basic.</td>
</tr>
<tr>
<td>Jun. 2004</td>
<td>0.0.2</td>
<td>First MacOS version.</td>
</tr>
<tr>
<td>Sep. 2004</td>
<td>0.0.3</td>
<td>Implemented S-parameters and noise analysis. Added microstrip components and BJT and MOSFET devices.</td>
</tr>
<tr>
<td>Mar. 2005</td>
<td>0.0.5</td>
<td>Implemented AC analysis and basic transient analysis.</td>
</tr>
<tr>
<td>May 2005</td>
<td>0.0.6</td>
<td>Mainly bug fixes and extensions to implemented analysis.</td>
</tr>
<tr>
<td>Jul. 2005</td>
<td>0.0.7</td>
<td>Windows 32 version. Simulator renamed as Qucs. New device library manager. Added post simulation data processing mathematical functions.</td>
</tr>
<tr>
<td>Jan. 2006</td>
<td>0.0.8</td>
<td>Support for pure digital simulation using FreeHDL. Added many new component models. Improved post simulation data plotting features. Added a filter synthesis tool.</td>
</tr>
<tr>
<td>May 2006</td>
<td>0.0.9</td>
<td>New functions in equation solver. Harmonic Balance simulation introduced. Many new components added.</td>
</tr>
<tr>
<td>Sept. 2006</td>
<td>0.0.10</td>
<td>Qucs converter tool improved. Support for nine-valued VHDL logic. Circuit optimization introduced using ASCO. Added attenuator design tool.</td>
</tr>
<tr>
<td>Mar. 2007</td>
<td>0.0.11</td>
<td>Added device parameters to equations and parameters to subcircuits. New models plus improvements to existing models. Using ADMS to translate Verilog-A device models for use in Qucs.</td>
</tr>
<tr>
<td>Nov. 2007</td>
<td>0.0.13</td>
<td>General improvements plus implementation of immediate vectors and matrices in equations.</td>
</tr>
<tr>
<td>Apr. 2008</td>
<td>0.0.14</td>
<td>Implemented multi-port equation-defined RF device (RFEDD) S, Y and Z parameters available. Two-port equation-defined device also supported.</td>
</tr>
<tr>
<td>Apr. 2009</td>
<td>0.0.15</td>
<td>Mainly bug fixes, small improvements and the addition of new models.</td>
</tr>
<tr>
<td>Mar. 2011</td>
<td>0.0.16</td>
<td>Implemented interactive post simulation data processing using Octave. Again many bug fixes, small improvements and the addition of new models.</td>
</tr>
</tbody>
</table>
## Background: QucsStudio development road map

<table>
<thead>
<tr>
<th>Date</th>
<th>Release</th>
<th>Notes [Highlights]</th>
</tr>
</thead>
<tbody>
<tr>
<td>Feb. 2011</td>
<td>1.0.0</td>
<td>First public version of simulator- features include more than 100 circuit components, DC analysis, AC analysis (including noise analysis and noise distribution analysis), S-parameter analysis (including noise simulation), transient analysis, Harmonic Balance analysis (including noise simulation), system simulation, parameter sweep and optimization of analogue circuits, digital simulation with ICARUS Verilog, PCB layout using KiCAD, numerical data processing using Octave, RF transmission line calculation (coaxial, microstrip, coupled microstrip, coplanar line, stripline, twisted pair rectangular waveguide etc., filter synthesis (LC ladder, stepped-impedance, microstrip, active filters etc., attenuator synthesis, and GPIB control.</td>
</tr>
<tr>
<td>Mar. 2011</td>
<td>1.1.0</td>
<td>Added VHDL digital simulation with GHDL plus numerous bug fixes.</td>
</tr>
<tr>
<td>Jun. 2011</td>
<td>1.2.0</td>
<td>Many small improvements plus bug fixes.</td>
</tr>
<tr>
<td>Nov. 2011</td>
<td>1.3.0</td>
<td>Added compiled C++ and Verilog-A device and circuit models using MinGW and ADMS, more bug fixes and small improvements. Released QucsStudio-light: QucsStudio without Octave and model Compiler.</td>
</tr>
<tr>
<td>Feb. 2012</td>
<td>1.3.1</td>
<td>Added large signal AC circuit simulation using Harmonic Balance. Many small improvements plus bug fixes. Range of compact semiconductor device models released: Verilog-A models held in binary C++ library.</td>
</tr>
</tbody>
</table>
SPICE and Qucs basic types of simulation facilities

**Schematic capture:** generates circuit diagram

**Circuit simulation:** DC, AC, TRAN, S-parameter, noise and Harmonic Balance

**Post simulation data processing and visualization**

---

**DC**

---

**AC**

---

**TRAN**

---

**S-parameter**

*Implementation 1. Qucs: built-in; 2. SPICE: via RCL networks*
Qucs post-simulation MATLAB®/*Octave** data processing features

**Equation blocks + simulation data sets**

- **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

abs adjoint angle arccos arccosec arc cot arc cosech arcosh arcosh arcsec arctan arg arsech asinh artanh avg besseli0 besselj bessely ceil conj cosec cosech cosh cot coth cumavg cump roduct cums um dB dbm dbm2w deg2rad det dft diff erf erfc erf cvn exp eye f ft fix floor Freq2Time GaCircle GpCircle hypot id ft ifft imag integrate interpolate inverse kbd limexp lins pace In log10 log2 logspace mag max min Mu Mu2 NoiseCircl e norm phase PlotVs polar prod rad2deg random real rms Rollet round rtoswr rtoy rtoz runavg sec se ch sign sin sinc sinh sqr sqrt srandom StabCir cleL StabCirc leS StabFactor StabMeasure stddev step stos stoy stoz sum tan tanh Time2Freq transpose twoport unwrap variance vt w2 dbm xvalue ytor ytos ytoz yvalue ztor zt oz ztoy

**Data processing**

- **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

**Tables and plots**

- Equation blocks + simulation data sets
- Constant values:
  - i, j, pi, e, kB, q
- Immediate values:
  - 2.5, 1.4+j5.1, [1, 3, 4, 5, 7], [11, 12; 21, 22]
- Range values:
  - 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


Limitations: NO user defined functions or control loops
Qucs modelling tools: Subcircuits with parameters

Emitter follower subcircuit

Voltage gain

Input resistance

Output resistance

Parameter sweep

Versus.0001: 3
PLVload: 0.468+j0.00109

DC1

Versus.0001: 888
PLVload: 0.993-j1.63e-06

Equation

Eqn1

Eqn1

V1

U=12 V

VCC

Vload

R1

R=Rload

Parameter sweep

AC1

Type=const

Values=[1M]

Equation

Eqn1

V1

U=12 V

VCC

Vload

R1

R=Rload

V3

U=1 V

Input

V1

U=12 V

VCC

Vload

R1

R=Rload

Output

Voltage gain

ac frequency: 1.41e+05

gain: -0.032

Input resistance

ac frequency: 1e+03

Rin: 3.16e+04-j20.2

Output resistance

ac frequency: 1e+03

RLin=100k

gain=dB(Vload.v/Vinput.v)

Equation

Eqn1

RLin=100k

Vaf=100.0

Cin=10u

Cout=100u

Equation

Eqn1

Rload=100k

Vaf=100.0

Cin=10u

Cout=100u

Equation

Eqn1

Rload=100k

Vaf=100.0

Cin=10u

Cout=100u

Equation

Eqn1

Rload=100k

Vaf=100.0

Cin=10u

Cout=100u

Equation

Eqn1

Rload=100k

Vaf=100.0

Cin=10u

Cout=100u

Equation

Eqn1

Rload=100k

Vaf=100.0

Cin=10u

Cout=100u

Equation

Eqn1

Rload=100k

Vaf=100.0

Cin=10u

Cout=100u

Equation

Eqn1

Rload=100k

Vaf=100.0

Cin=10u

Cout=100u

Equation

Eqn1

Rload=100k

Vaf=100.0

Cin=10u

Cout=100u

Equation

Eqn1

Rload=100k

Vaf=100.0

Cin=10u

Cout=100u
Qucs modelling tools: Macromodels with parameters

Single pole OP AMP macromodel specification:
- \( V_{\text{off}} \) = Input offset voltage
- \( R_d \) = differential input resistance
- \( C_d \) = differential input capacitance
- AOLDC = DC open loop differential voltage gain
- GBP = Gain bandwidth product
- \( V_{\text{limit}} \) = output voltage saturation limit
- \( R_o \) = Output resistance

**Closed loop amplifier with x10 gain**

**Equation**
- \( \text{Eqn1} \)
- \( R_p = 10^8 (\text{AOLDC} / 20) \)
- \( C_p = 1 / (2 \pi f \cdot \text{GBP}) \)

Macromodel body

Closed loop amplifier with x10 gain

Macromodel symbol

Three OP AMP state variable filter
Qucs modelling tools: Equation-defined device (EDD) modelling

Relationships between Qucs schematic symbols and Verilog-A code fragments

Fundamental EDD blocks

Qucs symbol | Quantity equations | Verilog-A code fragment | Quantity equations | Verilog-A code fragment
---|---|---|---|---
[Image of Qucs symbols and logic diagrams]

- **n_x1**
  - Iname = f(V2, V3, ..., V8)
  - l2, l3, ..., l8 = 0 and Q1, Q2, ..., Q8 = 0.
  - Where Vm = V(n_xm, n_ym) or Vm = V(n_xm), and 2 ≤ m ≤ 8.

- **n_y1**
  - Iname = f(V2, V3, ..., V8);
  - Or
  - Iname <= f(V2, V3, ..., V8);

- **n_x8**
  - l2, l3, ..., l8 = 0 and Q1, Q2, ..., Q8 = 0.
  - Where Vm = V(n_xm, n_ym) or Vm = V(n_xm), and 1 ≤ m ≤ 8.

- **n_y8**
  - Iname = X * V(n_x, n_y);

- **n_x1**
  - Q1 = f(V1, V2, ..., V8, l1, l2, ..., l8)
  - l2, ..., l8 = 0.
  - Where Vm = V(n_xm, n_ym) or Vm = V(n_xm), and 1 ≤ m ≤ 8.

- **n_y1**
  - Iname = ddt(Q1);
  - Or
  - Iname = ddt(Q1);

- **n_x8**
  - l2, l3, ..., l8 = 0 and Q1, Q2, ..., Q8 = 0.
  - Where Vm = V(n_xm, n_ym) or Vm = V(n_xm), and 2 ≤ m ≤ 8.

- **n_y8**
  - Iname = V(n_x);

---

(a) Model initialisation block

```verilog
@ (initial_model)
begin
  con1 = ......;
  con2 = ......;
  con3 = ......;
end
```

(b) Standard resistors

```verilog
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 = 1.38066505e-23 K^-1, $temperature is the resistor temperature in Kelvin.
```

(c) Noise free resistors

```verilog
I(n_x, n_y) <+ V(n_x, n_y)/R;
```

(d) Voltage controlled current block

```verilog
Iname = X * V(n_x, n_y);
```

(e) Current to voltage conversion block

```verilog
Iname = I1 = V1
```
Qucs modelling tools: Verilog-A compact device modelling

Generating Verilog-A code: From Qucs equation-defined devices via Verilog-A code fragments to a Verilog-A standardised template

1. Compile Verilog-A template code with ADMS
2. Add new model to Qucs by patching C++ code
3. Add new model symbol to Qucs C++ code
4. Compile and link Qucs static C++ code to generate new version of Qucs
Qucs: Analogue circuit simulation and device modelling features

Qucs and QucsStudio also allow digital simulation with VHDL and Verilog

QucsStudio changes
- QUCSATOR replaced by new analogue circuit simulator
- Octave simulation output data processing added
- KiCAD PCB layout software added
- Communications system simulation added
- New C++ component added which allows “turn-key” Verilog-A compact model development

** PS2SP - SPICE PSpice to SPICE preprocessor, http://members.aon.at/fschmid7/
QucsStudio: Compact semiconductor device and circuit macromodel construction using ADMS and MinGW dynamically linked models

Generate Verilog-A code with the QucsStudio colour highlighted text editor

“Turn-Key” Verilog-A compact model development system WHERE the Verilog-A code is automatically recompiled ONLY if it has been changed prior to the start of a simulation

Other QucsStudio components may be included in a subcircuit with one or more compiled C++ models
QucsStudio: RF Curtice MESFET compact model
Part 1: Verilog-A code

// Verilog-A Curtice MESFET: hyperbolic tangent model
// with fixed capacitance and noise ; Curtice.va.
// This is free software; you can redistribute it and/or modify
// it under the terms of the GNU General Public License as published by
// the Free Software Foundation; either version 2, or (at your option)
// any later version.
//
// Copyright (C), Mike Brinson, mbrin72043@yahoo.co.uk
// QucsStudio version September 2011.
//
`include "disciplines.vams"
`include "constants.vams"

module Curtice(Drain, Gate, Source);
inout Drain, Gate, Source;
electrical Drain, Gate, Source;

`define attr(txt) (*txt*)
`define CTOK 273.15
`define K1 7.02e-4
`define K2 1108.0
`define K3 400e-6

`define GMIN 1e-12

parameter real Area = 1 from (1 : inf)        `attr(info="area factor");
parameter real Vto = -1.8 from (-inf : inf)   `attr(info="pinch-off voltage" unit = "V");
parameter real Beta = 3e-3 from [1e-9 : inf)   `attr(info="transconductance parameter" unit = "A/(V*V)");
parameter real Alpha = 3.25 from [1e-9 : inf)  `attr(info="saturation voltage parameter" unit="1/V");
parameter real Lambda = 0.05 from [1e-9 : inf) `attr(info="channel length modulation parameter" unit="1/V");
parameter real Vtotc = 0 from (-inf : inf)    `attr(info="Vto temperature coefficient");
parameter real Betatc = 0 from (-inf : inf)   `attr(info="Beta temperature coefficient" unit = "/Celsius");
parameter real Alphatc = 0 from (-inf : inf) `attr(info="Alpha temperature coefficient" unit = "/Celsius");
parameter real Eg = 1.11 from [1e-6 : inf)    `attr(info="energy gap" unit = "eV");
parameter real Tau = 1e-9 from [1e-20 : inf)  `attr(info="transit time under gate" unit = "s");
parameter real Is = 1e-14 from [1e-20 : inf) `attr(info="diode saturation current" unit = "A");
parameter real N = 1 from [1e-9 : inf)       `attr(info="diode emission coefficient");
parameter real Xti = 3.0 from [1e-9 : inf)   `attr(info="diode saturation current temperature coefficient");
parameter real Af = 1 from [0 : inf]         `attr(info="flicker noise exponent");
parameter real Kf = 0 from [0 : inf]         `attr(info="flicker noise coefficient");
parameter real Gdsnoi = 1 from [0 : inf]     `attr(info="shot noise coefficient");
parameter real Bv = 1e9 from (-inf : inf)    `attr(info="drain-gate junction reverse bias breakdown voltage" unit = "V");
parameter real R1 = 1e9 from [1e-9 : inf)    `attr(info="breakdown slope resistance" unit = "Ohms");
parameter real Nsc = 1 from [1e-9 : inf)     `attr(info="subthreshold conductance parameter");
parameter real Temp = 26.85 from [-273 : inf) `attr(info="circuit temperature" unit = "Celsius");
parameter real Tnom = 26.85 from [-273 : inf) `attr(info="parameter measurement temperature" unit = "Celsius");
QucsStudio: RF Curtice MESFET compact model
Part 1: Verilog-A code continued

real T1, T2, Vt_T2, Vto_T2, Rg_T2, Rd_T2, Rs_T2, Vf, Ah, Beta_T2, Ids;
real Tr, con1, Eg_T1, Eg_T2, Qds;
real Cgs_T2,Cgd_T2, Vbi_T2; real lgs1, lgs2, Is_T2;
real con2, con3, VfDC; real fourkt, gm, An, thermal_pwr, flicker_pwr, Alpha_T2;

// Model branches
branch (Drain, Source) bDS; branch (Gate, Source) bGS; branch (Gate, Drain) bGD;

// analog begin
T1=Tnom+CTOK; T2 = $temperature; Tr=T2/T1; Vt_T2 = $vt;
Eg_T1=Eg-`K1*T1*`K2+T1; Vto_T2=Vto+Vtotc*(T2-T1);
Beta_T2=Area*Beta*pow(1.01, Betatc*(T2-T1));
Is_T2=Area*Is(pow(Tr, (Xti/N))*limexp(-(`P_Q*Eg_T1)*(1-Tr)/(`P_K*T2));
con2 = -5.0*N*Vt_T2; con3 = 1.0/(N*Vt_T2); fourkt=Area*4.0*P_K*T2;
Alpha_T2=Alpha*(pow( 1.01, Alphatc*(T2-T1))); // Drain to source current with subthreshold modification
VfDC = V(bGS)-Vto_T2; Ah = 1/(2*Vt_T2*Nsc); Vf = ln(1+exp(Ah*VfDC))/Ah;
Ids = Beta_T2*Vf*Vf*(1+Lambda*V(bDS))*tanh(Alpha*V(bDS));

// Charge equations
Qds = Tau*Ids;

// Diode DC equations
Igs1 = (V(bGS) > con2) ? Is_T2*(limexp(V(bGS)*con3) -1.0) : -Is_T2;
Igs2 = (V(bGS) < -Bv)  ? (V(bGS)+Bv)/R1 : 0.0;

// Current contributions
I(bGS) <+ Igs1+Igs2+`GMIN*V(bGS); I(bDS) <+ Ids; I(bDS) <+ ddt(Qds);

// Model noise equations
gm = 2*Ids/VfDC;
if ( V(bDS) < 3/Alpha )
begin
    An=1-V(bDS)/VfDC;
    thermal_pwr= (8*`P_K*T2*gm/3)*((1+An+An*An)/(1+An))*Gdsnoi;
end
else
    thermal_pwr=(8*`P_K*T2*gm/3)*Gdsnoi;
I(bDS) <+ white_noise(thermal_pwr, "thermal"); flicker_pwr = Kf*pow(Ids,Af);
I(bDS) <+ flicker_noise(flicker_pwr,1.0, "flicker");
end
QucsStudio: RF Curtice MESFET compact model
Part 2: Model Schematic

Verilog-A Curtice compact semiconductor device model for an n type MESFET
Properties:
1. Curtice quadratic model - basic plus subthreshold improvement [1]
2. Gate - source current model - exponential diode model plus linear reverse breakdown model.
3. Series terminal resistors Rd, Rg and Rs
4. Thermal noise generated by Rd, Rg and Rs
5. Channel noise
6. Gate noise
7. Flicker noise
8. Bias independent capacitors Cgse, Cgde and Cdse.


Verilog-A model parameters are listed and described in file Curtice.va.

MESFET Symbol and parameters

MESFET subcircuit body
QucsStudio: RF Curtice MESFET compact model
Part 3: Test simulations

Parameter sweep
SW1
Sim=SW2
Param=Vgs
Type=lin
Start=-2
Stop=0
Points=11

Parameter sweep
SW2
Sim=DC1
Param=Vds
Type=lin
Start=0
Stop=5
Points=51

DC1
dc simulation

Vds (V)
0 1 2 3 4 5
Ids (A)
0 0.005 0.01 0.015

CURTICE1
Area=1
Vto=-1.8
Beta=3e-3
Alpha=2.25
Lambda=0.05
Vtotc=0
Betatc=0
Alphatc=0
Eg=1.11
Tau=1e-9
Is=1e-14
N=1
Xti=3.0
Af=1
Kf=0
Gdsnoi=1
Bv=1e9
R1=1e9
Temp=26.85
Tnom=26.85
Cgse=300f
Cgde=30f
Cdse=300f
Lge=0.0
Lde=0.0
Lse=0.0
Nsc=1
Rg=5.1
Rd=1.3
Rs=1.3
Rin=1e-3
QucsStudio: RF Curtice MESFET compact model
Part 3: Test simulations

**Equation**

Eqn1

\[ V_{gain} = dB(V_{out}/V_{in}) \]
\[ V_{phase} = \text{wphase}(V_{out}/V_{in}) \]
QucsStudio: RF Curtice MESFET compact model
Part 3: Test simulations continued

CURTICE1
Area=1
Vto=1.8
Beta=3e-3
Alpha=2.25
Lambda=0.05
Vt=0
Beta_t=0
Alpha_t=0
Eg=1.11
T=1.0e-9
L=1.0e-14
M=1
Xi=3.0
Af=1
Kf=0
Gds=1
Bv=1e9
R1=1e9
Temp=26.85
T_n=26.85
Cgs=300f
Cgd=30f
Cds=300f
Lg=0.0
Ld=0.0
Ls=0.0
Nsc=1
Rg=5.1
Rd=1.3
Rs=1.3
Rin=1e-3

Vin
C2=10uF
R5
R=4.7k
V1
U=5
Vout
C1=C=10u
R6
R=47k
R7
R=1k
C3
C=10u
V3
U=-1 V
V4
U=0.2
freq=100kHz

transient simulation
TR1
Stop=50u
Points=1001
method=Gear4
reitol=0.01
abstol=10 μV
initialDC=yes

time
Vout
Vs
0
1e-5
2e-5
3e-5
4e-5
5e-5
0
0
0
0.439
0.439
0.439

Vout Vs
0
1
2
3
4
5
0
1
2
3
4
5

QucsStudio: RF Curtice MESFET compact model
Part 3: Test simulations continued
QucsStudio: Octave post-simulation data processing

Harmonic Balance

Post-simulation data processing control file

FFT

Octave plot of Vout amplitude spectra against frequency

Includes “User defined functions”
Qucs/QucsStudio: Physical System simulation.  
Part 1: Continuous systems

radioActiveDecay simulation:

\[-\frac{dA}{dt} = 0.347 \cdot A\]
\[-\frac{dB}{dt} = -0.347 \cdot A + 0.174 \cdot B\]
\[-\frac{dC}{dt} = -0.174 \cdot B\]

Initial conditions at \( t = 0 \) s, \( A = 1 \), \( B = 0 \), \( C = 0 \).
A switched capacitor inverting integrator example

Qucs/QucsStudio: Physical System simulation.
Part 2: sampled data systems
QucsStudio: System simulation. Part 3 communications systems

**Channel with noise**
- X4 samples = numBits * num_per_bit
  - variance = 1/Q_factor^2
- X5

**Delay line interferometer**
- X6
- X7
- X8 phi = Phase
- X9
- X10

**Detector**
- X11 Func = norm
- X12 Func = norm
- X13 Cutoff = Bandwidth/BaudRate/num_per_bit
- G = 1
- Order = 5
- Umax = 1e4 destructive

**System simulation**
Eqn2
- BaudRate = 20G
- Bandwidth = 40GHz
- FSR = 20GHz
- Q_factor = 10
- Phase = 45
- numBits = 64
- num_per_bit = 32

NRZ-QPSK transmission as used in optical telecommunication systems.

The schematic shows one receiver arm only. A second one with a phase shift of 135° is needed to receive the Q channel.

QucsStudio example, Michael Margraf, 2010
QucsStudio: System manufacturing printed circuit boards

simple 1.8 kHz oscillator

Press F10 (Tools->Create PCB netlist) in order to modify PCB layout.

The Gerber files can be viewed with Tools->Gerber Viewer

QucsStudio example, Michael Margraf, 2010

Simulation

PCB layout by KiCAD
## Qucs/QucsStudio: Additional features

<table>
<thead>
<tr>
<th>Feature</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Digital simulation: VHDL</td>
<td>Qucs → FreeHDL</td>
</tr>
<tr>
<td></td>
<td>QucsStudio → GHDL</td>
</tr>
<tr>
<td></td>
<td>Both using ICARUS</td>
</tr>
<tr>
<td>SPICE netlist defined components</td>
<td></td>
</tr>
<tr>
<td>Small signal AC and S-parameter noise</td>
<td></td>
</tr>
<tr>
<td>Harmonic Balance and noise and noise parameter calculations</td>
<td></td>
</tr>
<tr>
<td>Tabular two and multiport S-parameter components</td>
<td>Qucs → User library only</td>
</tr>
<tr>
<td>Tabular voltage and current sources</td>
<td>QucsStudio → User and binary libraries</td>
</tr>
<tr>
<td>Component libraries</td>
<td></td>
</tr>
<tr>
<td>RF components: transmission line, microstrip line and coplanar line</td>
<td></td>
</tr>
<tr>
<td>Import and export data of different formats</td>
<td></td>
</tr>
<tr>
<td>GPIB device control</td>
<td></td>
</tr>
<tr>
<td>Create and extract project packages</td>
<td></td>
</tr>
<tr>
<td>Circuit performance optimization using ASCO</td>
<td></td>
</tr>
<tr>
<td>Templates: Octave function Octave_function.m, S-parameters.sch, skeleton.cpp, skeleton.va, skeleton.vhdl and symbols.sch</td>
<td></td>
</tr>
<tr>
<td>Design tools: Text editor, Filter synthesis, Line calculations, Matching circuits</td>
<td>Both Qucs and QucsStudio</td>
</tr>
<tr>
<td>Gerber Viewer</td>
<td>QucsStudio only</td>
</tr>
</tbody>
</table>

Qucs and QucsStudio are freely available under the open source General Public Licence

Download from: Qucs version 0.0.16 → [http://qucs.sourceforge.net](http://qucs.sourceforge.net)

QucsStudio from [http://mydarc.de/DD6UM/QucsStudio/qucsstudio.html](http://mydarc.de/DD6UM/QucsStudio/qucsstudio.html)

[Currently QucsStudio supports Windows® only: QucsStudio-1.3.1.zip or QucsStudio-1.3.0_light.zip {without Octave and model compiler}]
QucsStudio: Latest additions: Large signal AC simulation

**RF Diode Switch Model**

**Test circuit**

- **RF Switch 1**
  - \( N = 1.0 \)
  - \( Is = 1e^{-15} \)
  - \( Tn = 26.85 \)
  - \( Temp = 26.85 \)
  - \( Rs = 0.1 \)
  - \( Vp = 1.0 \)
  - \( M = 0.5 \)
  - \( Fc = 1.0 \)
  - \( Cj0 = 60\text{ff} \)
  - \( Tt = 0.0n \)

- **HBDiode 1**
  - \( N = 1.0 \)
  - \( Is = Is \)
  - \( Rs = Rs \)
  - \( Tn = Tnom \)
  - \( Temp = Temp \)
  - \( Vp = Vp \)
  - \( M = M \)
  - \( Fc = Fc \)
  - \( Cj0 = Cj0 \)
  - \( Tt = Tt \)

Equation

- **On State:**
  - \( V1 \)
  - \( U = VState \)
  - \( VState = 2+4 \text{ State} \)

Equation

- **Off State:**
  - \( V \text{State} = 0 \)

**Graphs:**

- **Vin vs Vout**
  - Frequency: \( 1e6 \text{ to } 1e1 \)

- **Value(Vin, Vout) vs Frequency**
  - Frequency: \( 1e6 \text{ to } 1e1 \)

- **Vout vs Vout**
  - Frequency: \( 1e6 \text{ to } 1e1 \)

- **Plot(Vin, Vout)**
  - Frequency: \( 1e6 \text{ to } 1e1 \)
The following changes to the QucsStudio software are being actively planned:

- A full range of non-linear semiconductor device models, including EKV, BSIM, HiCUM etc.
- Model improvements for different electrical and physical domains and systems.
- Circuit oscillation simulation by Harmonic Balance analysis.
- Periodic steady state analysis.
- Mixed-mode simulation.
- EM field simulation.
- Upgrade of the QucsStudio GUI to Qt4.
- A range of Octave functions for simulation data analysis and visualisation.
- Many smaller improvements and extensions.

Individuals or groups interested in contributing to the development of QucsStudio should contact Michael Margraf (michael.margraf@alumini.tu-berlin.de) or Mike Brinson (mbrin72043@yahoo.co.uk)
Summary

1. Qucs and QucsStudio are freely available circuit simulators distributed as open source software under the GNU General Public Licence (GPL).

2. This presentation has attempted to outline the history and the fundamental features of the packages, the available components, libraries, built in design aids, analysis types and post-simulation data analysis and visualisation capabilities.

3. The presentation also introduced a number of basic approaches to circuit simulation with Qucs and QucsStudio.

4. A series of slides also showed how the compact semiconductor modelling and circuit macromodeling features implemented in the current QucsStudio release can be used to develop equation-defined component models of established and emerging technology devices.

5. A “turn-key” approach to compact device modelling using the Verilog-A hardware description language was introduced and the proposed modelling system demonstrated via the development of a MESFET RF device simulation model.

6. The final section of the presentation briefly introduced the use of QucsStudio for system simulation and PCB development.
References


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. 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), Frankfurth(O), Germany, 2 April 2009. Available from: http://www.mos-ak.org/frankfurt_o/papers/M_Brinson_Qucs_COMON_April_2_2009_final.pdf

