Qucs: Current and planned circuit simulation and device modelling developments; a review

Mike Brinson 1, mbrin72043@yahoo.co.uk.
Richard Crozier 2, richard.crozier@yahoo.co.uk
Vadim Kuznetsov 3, ra3xdh@gmail.com
Clemens Novak 4, clemens@familie-novak.net
Bastien Roucaries 5, bastien.roucaries@satie.ens-cauchan.fr
Felix Salfelder 6, felix<notifications@github.com>
Frans Schreuder 7, fransschreuder@gmail.com
Guilherme Brondani Torri 4, guitorri@gmail.com

1Centre for Communications Technology, London Metropolitan University, UK
2The University of Edinburgh, UK
3Bauman Moscow Technical University, Russia
4Qucs Developer
5Laboratoire SATIE — CNRS UMR 8929, Université de Cergy-Pontoise, ENS Cachan, FR
6Gnucap and Qucs Developer
7Nikhef, Amsterdam, NL

Plus contributions from the Qucs ”User Community”

Presented at the MOS-AK DATA Workshop, Dresden, 18 March 2016
Qucs: Current and planned circuit simulation and device modelling developments; a review

- Qucs-0.0.19 and Qucs-0.0.19-S-RC4:
  1. Background and release details
  2. Review of changes and improvements

- Qucs Verilog-A modelling: current position and the way forward with AMS and compact semiconductor device models?
  1. Background - free non-GPL model licences?
  2. New release of ADMS under GPL 3
  3. Possible solution to compact device modelling problems

- Qucs development after release 0.0.19: the way forward to Qucs-0.0.20; more improvements and merging of Qucs and Qucs-S branches
  1. Qucs-0.0.19: work in progress
  2. Qucs-0.0.19-S: work in progress
  3. Structure after merging Qucs-0.0.19 and Qucs-0.0.19-S? - (Wish list)
  4. Integrated compact modelling capabilities

- Summary
## Qucs-0.0.19: background, features and release details

### Features
- **GUI/IDE**
- **Schematic capture**
- **Qucsator tools**
- **Optimizer (ASCO)**
- **Icarus-Verilog**
- **FreeHDL (VHDL)**
- **Data visualization**
- **Equation system**
- **Component library**
- **Design/synthesis tools**

### Extensible
- **SPICE netlist import**
- **Verilog-A model builder**
- **Octave/MATLAB support**

### Dependencies
- **C++ compiler**
- **Qt4 (with Qt5 support)**
- **Autoconf / CMake**
- **gperf / flex / bison**
- **ADMS**
- **LaTeX**

### Qucs schematic
- **Schematic to netlist**
- **Schematic to print**
- **Dump components data**

### Custom file formats
- **Schematic**
- **Library**
- **Netlist**
- **Data file**

### Qucsator simulator
- **DC**
- **Transient**
- **AC**
- **AC Noise**
- **S-Parameter**
- **S-Parameter Noise** (Harmonic Balance)

### Qucsconv converter
- **SPICE to Qucs**
- **Qucsdata to Touchstone**
- **Touchstone to Qucsdata**
- **Qucsdata to MATLAB/Octave**

### Release 0.0.19 (February 05, 2016)
- Bug fixing, usability improvements, build system cleanup
- Ongoing port to Qt5 support to Qt4
- New active filter synthesis tool
- Integration of regression tests, qucs-test repository
- Removal of non-GPL models

### Release data and improvements

1) **Source tarball:**
   - http://sourceforge.net/projects/qucs/files/qucs/0.0.19-snapshots/qucs-0.0.19-160204-git-83cc216.tar.gz
   - The latest build instructions can be found at: https://github.com/Qucs/qucs/blob/master/README.md
   - On Windows, perhaps the easiest way is to use MSYS2. The AppVeyor script gives a good idea on how to proceed: https://github.com/Qucs/qucs/blob/master/.appveyor.yml

2) **Binaries for Windows 64bit:**
   - http://sourceforge.net/projects/qucs/files/qucs-binary/0.0.19-snapshots/qucs-0.0.19-160204-git-83cc216-win64.zip
   - http://sourceforge.net/projects/qucs/files/qucs-binary/0.0.19-snapshots/qucs-0.0.19-160204-git-83cc216-win64.exe
# Qucs-0.0.19-S: background, features and release details

**Features**
- Extensible
- SPICE netlist import
- Verilog-A model builder
- Verilog-A model synthesizer
- Octave/MATLAB support

**Simulation tools**
- Qucs
- Ngspice
- Xyce (serial)
- Xyce (parallel)
- SpiceOpus
- Optimizer (ASCO)
- Icarus-Verilog
- FreeHDL (VHDL)
- Data visualization
- Equation system
- Component library
- Design/synthesis tools

**Qucs schematic**
- Schematic to netlist
- Schematic to print
- Dump components data
- Custom file formats

**Qucsator**
- DC
- Transient
- AC
- AC Noise
- S-Parameter
- S-Parameter Noise
- (Harmonic Balance)

**Quscconv converter**
- SPICE to Qucs
- SPICE to Qucslib
- vcd to Qucsdata
- Qucsdata to csv
- Qucsdata to Touchstone
cdf to Qucsdata
- Touchstone to Qucsdata
csv to Qucsdata
- zvr to Qucsdata
- mdl to Qucsdata
- Qucsdata to MATLAB/Octave

<table>
<thead>
<tr>
<th>Background</th>
<th>Operation</th>
</tr>
</thead>
</table>
| Release 0.0.19-S-rc4 (January, 2016) | 1) Source tarball: https://github.com/ra3xdh/qucs/releases/tag/0.0.19S-rc4qucs-0.0.19S-rc4.tar.gz  
Source code: https://github.com/ra3xdh/qucs/releases/tag/Source code (zip)  
https://github.com/ra3xdh/qucs/releases/tag/Source code (tar.gz)  
Follow official build instructions at https://github.com/Qucs/qucs to build executables from the source tarball.| 2) Binary for Windows 64bit: https://github.com/ra3xdh/qucs/releases/tag/qucs-0.0.19S-rc4-setup.exe  
| Added Pole-Zero analysis with Ngspice | Release date |
| Added XSPICE analogue devices | |
| Added magnetic core models and improvements | |
| Added .MODEL and .INCLUDE directives support | |
| Added new Transformers library for Qucs+Ngspice | |
| Unified SPICE components icons | |
| Added PlotVs() emulation for SPICE | |
| Fixed different bugs including bugfixes from Qucs-0.0.19 | |
Qucs-0.0.19-S additional SPICE and XSPICE component, control and simulation icons
Qucs-0.0.19-S XSPICE standard models: (1) S domain transfer function block

\[ V_1 \quad V = \text{DC 0 AC 1 sin(0 1 3000 0 0)} \quad nV_{\text{Vin}} \quad nV_{\text{Out}} \quad R_1 \quad R = 10k \]

**Nutmeg**

NutmegEq1
Simulation=ac
TF_gain=\text{dB(V(nVout)/V(nVin))}

**S Domain Transfer Function**

SDTF1
A=\text{Cheby\_LP\_3kHz}
A\_Line 2=\text{model Cheby\_LP\_3kHz s\_xfer}
A\_Line 3=+ (in\_offset = 0.0 gain = 1.0)
A\_Line 4=+ num\_coeff = [1.0] denormalized\_freq = 10000
A\_Line 5=+ den\_coeff = [1.0 1.42562 1.51629]
A\_Line 6=+ int\_ic = [0 0 0]

**ac simulation**

AC1
Type=log
Start=0.1 Hz
Stop=100 kHz
Points=121

**transient simulation**

TR1
Type=lin
Start=0
Stop=5 ms

**Fourier simulation**

FOUR1
Sim=TR1
numfreq=10
F0=1 kHz
Vars=V(nvin) V(nvout)
Qucs-0.0.19-S XSPICE standard models: (2) non-linear transformer and magnetic core blocks
Qucs-0.0.19-S XSPICE standard models: (3) non-linear transformer and magnetic core block macromodel example

CORE1
A =Steel_core
A_Li ne 2=.model Steel_core core length = 0.2 area=2e-4
A_Li ne 3++ H_array= [-10000 -9000 -8000 -7000 -6000 -5000 -4000 -3000 -2500 -2000 -1500 -1000 -750 -500 -250 0
A_Li ne 4++ 250 500 750 1000 1500 2000 2500 3000 4000 5000 6000 7000 8000 9000 10000]
A_Li ne 5++ B_array=[-1.506 -1.504 -1.5035 -1.503 -1.502 -1.501 -1.5005 -1.5 -1.48 -1.45 -1.37 -1.0 -0.825 -0.55 -0.3 0
A_Li ne 6++ 0.3 0.55 0.825 1.0 1.37 1.45 1.48 1.5 1.5005 1.501 1.502 1.503 1.5035 1.504 1.506 ]

CORE2
A =Steel_core
A_Li ne 2=.model Steel_core core length = 0.2 area=2e-4

CORE3
A =Steel_core
A_Li ne 2=.model Steel_core core length = 0.2 area=2e-4

ICOUPLE1
A =P1couple
A_Li ne 2=.model P1couple icouple (num_turns={npi})

ICOUPLE2
A =P1couple
A_Li ne 2=.model P1couple icouple (num_turns={npi})

ICOUPLE3
A =P1couple
A_Li ne 2=.model P1couple icouple (num_turns={npi})

ICOUPLE4
A =S1couple
A_Li ne 2=.model S1couple icouple (num_turns={nts1})

ICOUPLE5
A =S1couple
A_Li ne 2=.model S1couple icouple (num_turns={nts1})

ICOUPLE6
A =S1couple
A_Li ne 2=.model S1couple icouple (num_turns={nts1})
Qucs-0.0.19-S two port network analysis: new probes, parameter conversion subcircuits and nutmeg parameter conversion blocks

**ac simulation**
AC1
Type=log
Start=1e8
Stop=3e10
Points=1487

**dc simulation**
DC1

**Nutmeg**
Y_Param_Extraction1
Simulation=ac
Y11_phase=cph(y11)*180/pi
Y11_mag=mag(y11)
Y12_mag=mag(y12)
Y12_phase=cph(y12)*180/pi
Y22_phase=cph(y22)*180/pi
Y22_mag=mag(y22)
Y21_phase=cph(y21)*180/pi
Y21_mag=mag(y21)
Qucs-0.0.19-S Ngspice and Xyce new simulation features: (1) An example Pole-Zero analysis with Qucs and Ngspice
Qucs-0.0.19-S Ngspice and Xyce new simulation features: (2) SPICE .MODEL directive support

- The SPICE .MODEL directive allows use of unmodified SPICE modelcards provided by electronic devices manufacturers
- Place a .MODEL directive on a schematic then copy a SPICE model from a component datasheet, finally pasting it on the .model directive
- An example of .MODEL attachment for a JFET
Qucs-0.0.19-S Ngspice and Xyce new simulation features: (3) Qucs PlotVs() support

- Ngspice has no PlotVs() equivalent for the generation of user defined data plots
- With Qucs-0.0.19-S the PlotVs() processing function has been moved to the GUI level in order to provide this feature. Qucs-0.0.19-S makes use of the @ symbol to specify X-variable
Qucs-0.0.19-S Ngspice and Xyce new simulation features: (3) Qucs PlotVs() support continued

- An example of PlotVs() usage for frequency and time-domain simulation
Qucs-0.0.19-S Ngspice and Xyce new simulation features: (4) FFT spectrum analysis with Nutmeg scripting

- Use nutmeg scripts with "Ngspice Custom simulation" to obtain the output spectrum
- .INCLUDE directive allows attachment of unchanged SPICE libraries to a schematic with the new SPICE-compatible devices symbols
Qucs-0.0.19-S Ngspice and Xyce new simulation features: (5) Single ended thermionic valve (tube) amplifier; demonstration of SPICE model usage and the new XSPICE transformer library

Output transformer library model

6V6 pentode model

ac simulation
AC1
Type=log
Start=1 Hz
Step=100 kHz
Points=101

transient simulation
TR1
Type=lin
Start=0
Step=5 ms
Qucs Verilog-A modelling: current position and the way forward with ADMS and compact semiconductor device models?

1. Background - free non-GPL model licences?
Problems: Free model code but not GPL because, for example:

**agree not to charge**, under GPL the code is free, but people can charge for a nicely packaged source, as long there are other ways to get the source for free;

**agree to acknowledge ... in the documentation**, this originates in the original BSD license, the advertising clause, which is incompatible with GPL and has been removed in new BSD licenses.

**agree to obey all government restrictions**, which government? this adds on top of what is allowed by GPL, which makes it incompatible;

*Non-GPL Verilog-A device models removed from Qucs-0.0.19* as a temporary measure until a solution can be found. The same action will take place for Qucs-0.0.19-S when it is finally formally released or merged with Qucs-0.0.19.

2. New release of ADMS under GPL 3 - version adms-2.3.5
Add new simplified constants.vams and disciplines.vams. Tested to work with models currently in use by Qucs, Ngspice, Xyce and Gnucap. Whenever these headers are used, adms informs the user about the availability of the standard headers at: http://accellera.org/downloads/standards/v-ams.

3. Possible solution to compact device modelling problems
In the future the Qucs team will try to work around the model license issues and provide Qucs users with a way to load these models dynamically. No solution ready yet.
Qucs development after release 0.0.19: the way forward to Qucs-0.0.20; Part 1 Linking Qucs-0.0.19 with GNUCAP

QUCS + GNUCAP by Felix Salfelder

- Proof of concept based on old code fragments by Fabian Vallon,
- **gnucsator**, a gnucap based qucsator supplemented by means of:
  - Input deck parser (qucsator input),
  - A few compatible components,
  - Command and semantics emulation (noninteractive, one-shot, probe-placement)
  - GNUCAP simulation data translated into Qucs dataset format (‘.dat’)
- Independent implementation, work in progress, see https://www.github.com/QUCS/gnucsator
- For proper integration: more work on both the front and back ends of the package needs to be done.
Work in progress: Qucs+GNUCAP;$ QUCSATOR=qucsator qucs -i rc.sch
Work in progress: `Qucs+GNUCAP; $ QUACSATOR=gnucsator.sh qucs -i rc.sch`
Qucs-S development after release 0.0.19-S: the way forward to Qucs-0.0.20; Part 1 DC bias display on a schematic - press key F8
Qucs-S development after release 0.0.19-S: the way forward to Qucs-0.0.20;
Part 2 XSPICE code model subcircuits
The "XSPICE generic device" component is the foundation for:

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

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

Precompiled CodeModel *.cm library attachment dataflow diagram:

1. **Schematic**
2. **Precompiled CodeModel library**
3. **XSPICE Generic device**
4. **spice_netlist()**
5. **.MODEL**
6. **Component**
7. **spice_netlist()**
8. **Qucs GUI**
9. **Data visualization**
10. **SPICE Netlist**
11. **.spinit file**
12. **Ngsimpce simulator**
The "XSPICE generic device" component is a building block for the construction of user-defined A-devices. It is defined by a comma separated port list, with allowed XSPICE port designators, then attached to a SPICE .MODEL statement.

```plaintext
void cm_ggain(ARGS) {
    Mif_Complex_t ac_gain;
    if(ANALYSIS != MIF_AC) {
        OUTPUT(out) = PARAM(out_offset)+PARAM(gain)*
            (INPUT(in)+PARAM(in_offset));
        PARTIAL(out,in) = PARAM(gain);
    } else {
        ac_gain.real = PARAM(gain);
        ac_gain.imag= 0.0;
        AC_GAIN(out,in) = ac_gain;
    }
}
```

Source code: cfunc.mod file
Qucs-S development after release 0.0.19-S: the way forward to Qucs-0.0.20; Part 5 XSPICE ”turn-key” model generation; compiler system dataflow diagram
Qucs-S development after release 0.0.19-S: the way forward to Qucs-0.0.20; Part 6 XSPICE diode model - (a) The Qucs-S subcircuit symbol and model circuit

XSPICE diode model based on:
/*
  diode cm model.  4 March 2016  Mike Brinson

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

This is free software; you can redistribute it and/or modify
it under the terms of the GNU General Public License as published by
the Free Software Foundation; either version 2, or (at your option)
any later version.
*
#define DERIVE 0
#include <math.h>
void cm_diode (ARGS)
{
  double Vt, temp, Vd, P1, P3, P4, PTNOM, PTEMP;
  double PIS, PAREA, PTXI, PEG, PN;
  double Tr, ls_temp, Id;
  double *derive;
  double exp80 = 5.5406334e34;
  double GMIN = 1e-12;

  PTNOM = PARAM(nom)-273.15;
  PTEMP = TEMPERATURE-273.15;
  Vttemp = 8.65387156e-5*PTEMP;
  PEG = PARAM(e);
  PIS = PARAM(s);
  PN = PARAM(n);
  PAREA = PARAM(area);
  PTXI = PARAM(xti);

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

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

  if(ANALYSIS != AC)
    { Vd = INPUT(diode);
      if (Vd > P3*Vt_temp) {
        if (P1*Vd < 0.0) { 
          Id = ls_temp*exp80*(1+(P1*Vd)-1.0) + GMIN * Vd;
          OUTPUT(diode) = Id;
          *derive = P1*ls_temp*exp80*P1*Vd+GMIN;
          PARTIAL(diode, diode) = *derive;
        }
        else {
          Id = ls_temp*exp80*ln((P1*Vd)-1.0) + GMIN*Vd;
          OUTPUT(diode) = Id;
          *derive = P1*ls_temp*exp80*P1*Vd+GMIN;
          PARTIAL(diode, diode) = *derive;
        }
      } else {
        Id = -ls_temp*exp80*ln(-P1*Vd)+GMIN*Vd;
        OUTPUT(diode) = Id;
        *derive = GMIN;
        PARTIAL(diode, diode) = *derive;
      }
    }
}
*/
Qucs-S development after release 0.0.19-S: the way forward to Qucs-0.0.20; Part 6 XSPICE diode model - (c) The XSPICE non-linear diode capacitance dnlcap/func.mod code
Qucs-S development after release 0.0.19-S: the way forward to Qucs-0.0.20; Part 6 XSPICE diode model - (d) The diode small signal AC performance; Y parameter, Rd and Cd extraction
Qucs-S development after release 0.0.19-S: the way forward to Qucs-0.0.20; Part 6 XSPICE diode model - (e) The diode Id/Vd temperature variation

**dc simulation**

DC1

**Parameter sweep**

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

![Diode Model Diagram](image)

```
CM_D1
is=1e-14
n=1
tnom=27
area=1
rs=0.01
xti=3.0
eg=1.11
temp=27
cj0=1e-12
vj=0.7
m=0.5
fc=0.5
tt=1e-10
```
Qucs-S development after release 0.0.19-S: the way forward to Qucs-0.0.20; Part 6 XSPICE diode model - (f) Diode transient response
Qucs Structure after merging Qucs-0.0.19 and Qucs-0.0.19-S? - Wish list

- How and when Qucs-0.0.19 and Qucs-0.0.19-S are merged is not decided yet! Indeed they may never merge but continue to function as two separate packages side-by-side!

- GUI and simulator:
  - Refactor/rewrite, (Qt4) Qt5, plug-ins, API...; Standard file formats, exchangeable

- Powerful circuit analysis tools:
  - Robust algorithms (Eigen, KLU); API, high level interface (SWIG); improved Harmonic-Balance
  - EM field simulation / extraction (openEMS, NEC2++); improved SPICE flavors compatibility/converter
  - Co-simulation (analogue + Verilog/VHDL), interface (icarus, GHDL); Monte-Carlo simulation
  - Solvers: Ngspice, Xyce, GnuCap, SpiceOpus

- Design and synthesis tools:
  - Data import / export

- Industry standard device models:
  - MEXTRAM, VBIC, HiSIM, IGBT, UTSoI, ...

- Hardware implementation:
  - Output layout data for input to PCB and IC packages, for example KiCad and Klayout
Integrated Qucs and Qucs-S compact modelling capabilities

**KEY**
- Qucs controlled process
- Manual process
- Qucs-S controlled process
- Common process/data

**Diagram:**
- **Device equations**
  - Verilog-A module synthesizer
  - Verilog-A code
  - ADMS
  - C++ model and symbol
  - Subcircuit model and symbol
  - EDD model
  - Algebraic equations
  - Library components
  - SPICE netlists
  - Circuit design data
  - Device equations

- **ASCO optimizer**
- Qucsator
- Gnucsator

- **SPICE circuit schematic**
- **XSPICE cfunc.mod fspec.ifs**
- **XSPICE CM compiler**

- **Qucs plots and tables**
- **Octave plots and tables**
Qucs 0.0.19 is a major release of the circuit simulator package with the extended features introduced in this presentation included. Qucs 0.0.19 has benefited from the significant amount of work done by the Qucs Development Team to remove bugs, restructure the software, port the GUI from Qt3 to Qt4, improve the performance of qucsator, add new circuit design and modelling features and make the Qucs GUI more user friendly and productive. As the Qucs Development Team moves on to release 0.0.20 it is difficult to say what the structure of the next release will be. However, whatever the final decision is concerning merging Qucs-0.0.19 and Qucs-0.0.19-S the Qucs project will continue to provide a freely available modern full featured circuit simulator under GPL.

Stable and development versions of Qucs-0.0.19 and Qucs-0.0.19-S can be downloaded from:
1. **Qucs-0.0.19**: http://qucs.sourceforge.net/ and https://github.com/Qucs/qucs/
2. **Qucs-0.0.19-S**: https://github.com/ra3xdh/qucs/releases/tag/0.0.19S-rc4