Qucs-S a maturing GPL software package for circuit simulation and compact modelling of current and emerging technology devices

Mike Brinson \(^1\), mbrin72043@yahoo.co.uk.
Vadim Kuznetsov \(^2\), ra3xdh@gmail.com

\(^1\) Centre for Communications Technology, London Metropolitan University, UK
\(^2\) Bauman Moscow Technical University, Russia

Presented at the 14th MOS-AK Workshop, ESSDERC/ESSCIRC, Lausanne, September 12, 2016
Qucs-S a maturing GPL software package: Introduction to presentation

- Background and release dates for Qucs-S/RC7
- Qucs-S documentation and circuit simulation capabilities
  - Qus-S Help documentation: User manual and reference material,
  - Qucs-S SPICE style components and model libraries,
  - Qucs-S extended circuit simulation features.
- Qucs-S Equation-Defined Device (EDD) and Verilog-A compact device modelling case studies
  - The Efficient Power Corporation (EPC) GaN EPC2001 power transistor model,
  - The MIT virtual source GaN-RF HEMT 1.0.0 model.
- Qucs-S modelling tool extensions and new features
  - XSPICE ”Code Model” support,
  - Qucs-S model parameter extraction controlled by Qctave script files.
- Qucs-S future developments - the way forward in 2016-2017
Qucs-S a maturing GPL software package: Background and release details

Qucs-0.0.19 is the next release in the Qucs series of software packages. This release is primarily GUI and quasator improvements plus general bug fixes.

Snapshot at https://sourceforge.net/projects/qucs/files/qucs-binary/0.0.19-snapshots/qucs-0.0.19-rc1-win64.exe/download

Extended Qucs circuit simulator package, including:
1. Improved Qucs GUI,
2. Access to qucsator, Ngspice, Xyce and SPICE OPUS
3. Increased number of circuit simulation types, including pole-zero analysis, distortion analysis, Harmonic Balance, steady state shooting method and Monte-Carlo statistical analysis,
4. Full range of SPICE 3f5, Ngspice, Xyce and SPICE OPUS device models,
5. SPICE style component libraries,
6. Equation-defined device modelling,
7. Verilog-A model synthesis and “Turn Key” C++ code compiler,
8. XSPICE Code model “Turn Key” compiler,
9. XSPICE analogue component library,
10. Nutmeg and Xyce script device modelling,
11. Qucs and Octave post-simulation data processing,
12. Qucs-S device parameter extraction with linked Octave packages.

Qucs-S can be found at:
https://github.com/ra3xdh/qucs/releases/tag/0.0.19S-rc7

New test suite for Qucs-S introduced. A set netlist and simulation tests available. Sources are here:
https://github.com/ra3xdh/qucs-test/tree/spice4qucs

New "Qucs-S Help" Documentation:
This is a "live" document which is updated by Qucs-S Developers as the package is improved.
Users can find it at -
Qucs-S a maturing GPL software package: Latest release impact

Description
Qucs is an integrated circuit simulator which means you are able to setup a circuit with a graphical user interface (GUI) and simulate the large-signal, small-signal and noise behaviour of the circuit. After that simulation has finished you can view the simulation results on a presentation page or window. Supports spice simulators.

Changelog
After install run in terminal: snap run qucs-spice.qucs

Permissions
- Home
- Unity7

Info
Support: https://github.com/eldarkg/qucs-spice-snap/issues
Website: http://qucs.sourceforge.net

Version: 0.0.195-RC7-snap2
Updated: Sep 5, 2016
Published: Sep 4, 2016
License: GNU GPLv2
File Size: 89.2 MB
Architectures: i386, amd64
Chapter 1. Introduction
Chapter 2. Basic Ngspice, Xyce and SPICE OPUS simulation
Chapter 3. Spice4Qucs subcircuits, macromodels and device libraries
Chapter 4. Device and component modelling with algebraic equations
Chapter 5. More advanced circuit simulation techniques.
Chapter 6. Ngspice, Xyce and SPICE OPUS post-simulation data processing with Qucs-S and Octave
Chapter 7. Qucs and SPICE simulation models that work with ngspice, Xyce and SPICE OPUS
Chapter 8. Ngspice custom simulation technology
Chapter 9. XSPICE standard components and library
Chapter 10. XSPICE user written device models and library
Chapter 11. Introduction to mixed analogue/digital simulation
Chapter 12. Verilog-A compact semiconductor device modelling
Chapter 13. RF simulation with Ngspice, Xyce and SPICE OPUS
Chapter 14. Qucs-S/Octave circuit simulation and device parameter extraction Interface
Chapter 15. References

Authors Mike Brinson (mbrin72043@yahoo.co.uk) and Vadim Kusnetsov (ra3xdh@gmail.com)

Copyright 2015, 2016

Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.1 or any later version published by the Free Software Foundation. A copy of the license is included in the section entitled "GNU Free Documentation License".
Qucs-S a maturing GPL software package: Ngspice, Xyce and SPICEOPUS built in components
### Qucs-S a maturing GPL software package: Available semiconductor device models

<table>
<thead>
<tr>
<th>Type</th>
<th>Description</th>
<th>Level</th>
<th>Ngspice</th>
<th>Xyce</th>
<th>SPICEOPUS</th>
</tr>
</thead>
<tbody>
<tr>
<td>D</td>
<td>Legacy</td>
<td>1</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td></td>
<td></td>
<td>2</td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>3</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BJT</td>
<td>Legacy</td>
<td>1</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td></td>
<td>VBIC</td>
<td>10,11,12</td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>FEH HBI X</td>
<td>23</td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>MEXTRAM</td>
<td>504,505</td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>JFET</td>
<td></td>
<td>1 (Statz)</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td></td>
<td></td>
<td>2</td>
<td>X</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>MESFET</td>
<td></td>
<td>1 (Ytterdol)</td>
<td>X</td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>MOSFET</td>
<td>Legacy</td>
<td>1</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td></td>
<td></td>
<td>2</td>
<td>X</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>3</td>
<td>X</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>4(BSIM1)</td>
<td>X</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>5(BSIM2)</td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>6</td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>BSIM3v2</td>
<td></td>
<td>47</td>
<td>X</td>
<td></td>
<td>X</td>
</tr>
<tr>
<td></td>
<td></td>
<td>8</td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>9</td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>BSIM3v3</td>
<td></td>
<td>53</td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>49</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BSIM4</td>
<td></td>
<td>60</td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>14</td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>54</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BSIM3SOlv1</td>
<td></td>
<td>55</td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>BSIM0lv2</td>
<td>56</td>
<td>X</td>
<td></td>
<td>X</td>
</tr>
<tr>
<td></td>
<td></td>
<td>58,55,57</td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>10</td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>STAGSOI3</td>
<td></td>
<td>57</td>
<td></td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>UFROI</td>
<td></td>
<td>58</td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>SOI5</td>
<td></td>
<td>60</td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>UFET</td>
<td></td>
<td>7</td>
<td></td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>EKVv2p6</td>
<td></td>
<td>44</td>
<td>X</td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>HISIM2</td>
<td></td>
<td>61,68</td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>HISIM_HV</td>
<td></td>
<td>62,63</td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>VDMOS</td>
<td></td>
<td>18</td>
<td></td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>BSIM 6p1</td>
<td></td>
<td>77</td>
<td></td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>PSP 103p1</td>
<td></td>
<td>103</td>
<td></td>
<td></td>
<td>X</td>
</tr>
</tbody>
</table>

Note: The table lists available semiconductor device models and their compatibility with different simulation software packages (Ngspice, Xyce, SPICEOPUS). The entries marked with 'X' indicate availability, while empty cells indicate unavailability.
Qucs-S a maturing GPL software package: Simulation control icons

**dc simulation**
- DC1

**transient simulation**
- TR1
  - Type=lin
  - Start=0
  - Stop=1 ms

**ac simulation**
- AC1
  - Type=lin
  - Start=1 Hz
  - Stop=10 GHz
  - Points=100

**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
  - m=4

**XYCE script**
- XYCESCR1
  - SpiceCode=
  - .AC LIN 2000 100 10MEG
  - .PRINT AC format=raw file=ac.txt V(1)

**Nutmeg script**
- Nutmeg
  - Equation
    - Eqn1
      - y=1
  - PARAM
    - Eqn1
      - SpicePar1
        - y=1

**Pole-Zero simulation**
- PZ1
  - Input=0
  - Output=0
  - TF_type=val
  - PZ_mode=pz

**Parameter sweep**
- SW1
  - Type=lin
  - Param=R1
  - Start=5 Ohm
  - Stop=50 Ohm
  - Points=100
  - Output=v(node1)

**Noise simulation**
- NOISE1
  - Type=lin
  - Start=1 Hz
  - Stop=10 kHz
  - Points=100
  - Output=v(node1)

**GLOBAL PARAM**
- SpGlobPar1
  - y=1

**OPTIONS**
- SpiceOptions1
  - GMIN=1e-12

**IC**
- SpicelC1
  - v(node1)=1

**INCLUDE**
- SpiceInclude1
  - File=/home/user/library.inc

**MODEL**
- SpiceModel1
  - Line_1 = .MODEL DIODE1 D(BF=50 Is=1e-13 Vbf = 50)
Qucs-S a maturing GPL software package: XSPICE analogue component models
Qucs-S a maturing GPL software package: Qucs-S extended circuit simulation Part 1. SPICE .OP to visual DC by pressing key "F8"

Diagram of a circuit with components R1, R2, R3, R4, R5, C1, C2, and V1. The circuit includes a noise simulation with parameters:

- NOISE1
  - Type=ac
  - Start=1
  - Stop=1e9
  - Points=181
  - Output=v(nout)
  - Source=V2

Parameters for simulation:

- Nutmeg
  - Type=ac
  - Start=1
  - Stop=1e9
  - Points=181
  - Simulation=ac
  - gain=\text{mag}(V(\text{nout})/V(\text{nin}))
  - phase=\text{cph}(V(\text{nout})/V(\text{nin}))*180/\pi

Graphs showing output vs. frequency and noise spectrum.

![Diagram showing circuit simulation results]

- **R1**: R=0.2
- **D1**: D=DMOD
  - D_Lin 2=.model DMOD D(is=1e-15 n=1 cj0=1p rs=0.1 bv=0.7)
- **V1**: V=dc 0 ac 0 sin(0.7 0.5 0.96e6 0 0)
- **V2**: V=dc 0 ac 0 sin(0.2 1.04e6 0 0)

**Harmonic balance simulation**
- HB1
  - f=0.95e6 1.05e6
  - n=3,3

**Transient simulation**
- TR1
  - Type=lin
  - Start=0
  - Stop=25 us
Qucs-S a maturing GPL software package: Qucs-S extended circuit simulation Part 4. Direct support for SPICE libraries
Qucs-S a maturing GPL software package: Compact device modelling tools
Part 1. Equation-Defined behvioural modelling and Verilog-A model code synthesis
Qucs-S a maturing GPL software package: Compact device modelling Part 2. Qucs-S EDD model of the Efficient Power Corporation (EPC) GaN EPC2001 power transistor

---


---

Ported from the EPC mathematical model common to LTSPICE, PSPICE, TSPICE and Spectra netlist models.
Qucs-S a maturing GPL software package: Compact device modelling Part 5. AC gate matching network, test bench and typical simulation results
Qucs-S a maturing GPL software package: Compact device modelling Part 6. Switching response test bench and typical simulation results
The Analogue Device Model Synthesizer (ADMS) version 2.3.5 is used by Qucs/Qucs-S, Ngspice, Xyce and Gnucap GPL circuit simulators.

ADMS is based on a subset of Verilog-A HDL selected for compact device modelling.

Although the Verilog-A HDL is standardised there is no guarantee that individual simulator implementations allow the same dialect of Verilog-A for modelling purposes, for example Qucs/Qucs-S Verilog-A models can include component noise while Ngspice does not implement thermal, shot or flicker noise.

Normally emerging technology Verilog-A compact models have to be modified, often by hand, to compile without error: specific areas which can cause problems are

- Internal node collapsing,
- Voltage limiting,
- Setting initial conditions,
- Model equations that include complex combinations of analogue functions,
- Thermal effects due to power dissipation.
Qucs-S a maturing GPL software package: Verilog-A Modeling of the MIT Virtual Source GaN-RF HEMT 1.0.0: Problems simulating with ADMS; workarounds and typical simulation data - Part 2. Model parameter statement error workarounds

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

```
X
ADMS synthesis/compile error
```

```
OK
```

```plaintext
parameter real vxord = 1.30e7 from [0:inf];  // Source injection velocity [cm/s]
polygon real VOrd = -2.0;                     // Threshold voltage of drain access transistor[V]
polygon real Cgrd = 5.0e-7 from [0:inf];     // Drain access area capacitance [F/um2]
polygon real deltaxrd = 1.3 from [0:inf];     // DBIL for drain access transistor
parameter real delta2rd = 0.30 from [0:inf];  // DBIL for drain access transistor
parameter real Vdibsat = 2.0 from [0:inf];    // DBIL for drain access transistor
parameter real Srd = 0.35 from [0:inf];       // Subthreshold slope for drain access transistor [V/Dec]
polygon real zeta = 0.0 from [0:inf];        // Self heating parameter (scaleable)
polygon real betard = 1.3 from [0:inf];      // Linear to saturation transition parameter
parameter real vphetard = 0.06 from [0:inf];  // Scattering velocity reduction parameter with Vg
parameter real ndrd = 0.80 from [0:inf];      // Punchthrough factor affects slope change in subthreshold

parameter real vxors = vxord from [0:inf];    // Source injection velocity [cm/s]
polygon real VOrs = VOrd;                     // Threshold voltage of drain access transistor[V]
polygon real Cgrs = Cgrd from [0:inf];       // Drain access area capacitance [F/um2]
polygon real deltaxrs = deltaxrd from [0:inf]; // DBIL for drain access transistor
parameter real delta2rs = delta2rd from [0:inf]; // DBIL for drain access transistor
parameter real Srs = Srd from [0:inf];        // Subthreshold slope for drain access transistor [V/Dec]
polygon real vphetars = 0.06 from [0:inf];   // Scattering velocity reduction parameter with Vg
parameter real ndrs = ndrd from [0:inf];      // Punchthrough factor affects slope change in subthreshold
parameter real betars = betard from [0:inf];  // Linear to saturation transition parameter
```

```markdown
QucS model parameters:
- `vxord`: Source injection velocity [cm/s]
- `VOrd`: Threshold voltage of drain access transistor [V]
- `Cgrd`: Drain access area capacitance [F/um2]
- `deltaxrd`: DBIL for drain access transistor
- `delta2rd`: DBIL for drain access transistor
- `Vdibsat`: DBIL for drain access transistor
- `Srd`: Subthreshold slope for drain access transistor [V/Dec]
- `zeta`: Self heating parameter (scaleable)
- `betard`: Linear to saturation transition parameter
- `vphetard`: Scattering velocity reduction parameter with Vg
- `ndrd`: Punchthrough factor affects slope change in subthreshold

QucS-S model parameters:
- `vxors`: Source injection velocity [cm/s]
- `VOrs`: Threshold voltage of drain access transistor [V]
- `Cgrs`: Drain access area capacitance [F/um2]
- `deltaxrs`: DBIL for drain access transistor
- `delta2rs`: DBIL for drain access transistor
- `Srs`: Subthreshold slope for drain access transistor [V/Dec]
- `vphetars`: Scattering velocity reduction parameter with Vg
- `ndrs`: Punchthrough factor affects slope change in subthreshold
- `betars`: Linear to saturation transition parameter
```
ADMS DOES NOT ALLOW voltage contributions of the form \( V(n) < +I(n)R \), where \( R \) is a resistance in \( \Omega \),

OR statements of the form \( V(n) < +0.0 \),

Resistors, for example 0.001, are used to short nodes (node collapsing), with \( I(n) < +V(N)/0.001 \).

**ADMS synthesis/compile error**

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

**OK**

```plaintext
if (Rsh > 1e-3 && Ls > 0)
    I(si,src) <- ldsrs;
else
    I(si,src) <- V(si, src)/1e-3;
//Source side contact resistance
if (Rc > 0) begin
    I(src,s) <- V(src,s) / ( Rc / Wg );
end else begin
    I(src,s) <- V(src,s)/1e-3;
end
```
Qucs-S a maturing GPL software package: Verilog-A Modeling of the MIT Virtual Source GaN-RF HEMT 1.0.0: Problems simulating with ADMS; workarounds and typical simulation data - Part 4. DC characteristics

Parameter sweep

SW1
Sim=SW2
Type=lin
Param=Vgs
Start=-6
Stop=0
Points=9

Parameter sweep

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

Equation
Eqn2
\[ gds_{\text{norm}} = \text{diff}(lds_{\text{norm}}, Vds) \]
\[ lds_{\text{norm}} = Pr_{\text{Ids}} / 25e3 \]

dc simulation

DC1
abstol=1 pA
vtol=10 uV
MaxIter=1500
The ADMS dialect of Verilog-A does not implement the `pwr(dt)` statement,

Device self-heating is often modelled with a parallel RC network where the volt drop across the RC combination represents the change in device temperature due to internal power dissipation,

\[ T_{th} = R_{th}P_d + Temp(P_d = 0) \]

where \( T_{th} \) is the device temperature at power dissipation \( P_d(W) \).

```vhdl
// Self-heating
I(P) <= dt( Cth * V(P) );
I(P) <= -( I(di,si) * V(di,si) + I(d,drc) * V(d,drc) + I(src,s) * V(src,s) + V(drc,di) * I(drc,di) + V(src,si) * I(src,si) );
I(P) <= V(P)/Rth;
```
Qucs-S a maturing GPL software package: Verilog-A Modeling of the MIT Virtual Source GaN-RF HEMT 1.0.0: Problems simulating with ADMS; workarounds and typical simulation data - Part 6. Variation of thermal resistance $R_{th}$ and its effect on DC characteristics.
Qucs-S a maturing GPL software package: Modelling tool additions and new features

- Qucs-S includes for the first time a turn-key XSPICE "code level modelling" package for use with the Ngspice and SPICE OPUS circuit simulators,

- Qucs-S has also been extended to include a new Qucs/Octave integrated tool set for compact device model and circuit macromodel parameter extraction. The technique employed is based on data fitting and optimization using measured, or manufacturers published device data, compared against simulated circuit data.
The XSPICE generic device component is the foundation for
- Precompiled XSPICE device (*.cm) library support, and
- Dynamic XSPICE “Code Model” compilation system which allows Code Model sources to be attached to a schematic and compiled automatically at simulation time.
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, plus XSPICE port designators. These are attached to a SPICE .MODEL statement.
Qucs-S a maturing GPL software package: XSPICE Turn-Key Model Generation; compiler system dataflow diagram
Qucs-S a maturing GPL software package: Ngspice/Xyce/SPICEOPUS
device parameter extraction from manufacturers data, or measurements,
controlled by Octave Script Files; structure diagram
Qucs-S is a relatively stable circuit simulator and compact modelling tool, incorporating the best features of Qucs, Ngspice, Xyce and SPICE OPUS. The next development phase will concentrate on the following:

- Testing the package and improving the "feature coverage" by the Qucs-S test suit,
- Adding the missing sections to the "Qucs-S Help" documentation,
- Introduction of XSPICE and Xyce digital models as the first step towards making Qucs-S a true mixed-mode electronic system and circuit simulation and modelling package,
- Improvements to XSPICE CodeModel development system, including new XXX.mod and XXX.ifs templates generated from model schematics,
- Continue development of the Qucs-S built-in libraries.

It is expected that the next development phase will last around one year with a series of release candidates published at regular intervals.
Qucs-S a maturing GPL software package: Endnote - A brief look into the future

Qucs-S True mixed-mode simulation and modelling

ANALOGUE and DIGITAL device and IC models

Qucsator
Ngspice
SPICE OPUS
Xyce

Transistor level digital models

In the next development phase

Ngspice and SPICE OPUS XSPICE digital IC models

Xyce behavioural digital IC models

X1 Device=BUF
VCC=5

X2 Device=INV
VCC=5

X5 Device=AND2
VCC=5

X7 Device=NAND2
VCC=5

X9 Device=OR2
VCC=5

X11 Device=NOR2
VCC=5

X20 Device=FADDER
VCC=5

X18 Device=PG01234567
D0=0
D1=0
D2=0
D3=0
D4=0
D5=0
D6=0
D7=0
VCC=5

X23 Device=MUX2TO1
VCC=5

X22 Device=MUX4TO1
VCC=5

X24 Device=PG0123
D0=0
D1=0
D2=0
D3=0
VCC=5

PG 0 1 2 3 4 5 6 7

33 / 33