

Merging Gnucap and Qucs – the Why and How

Felix Salfelder

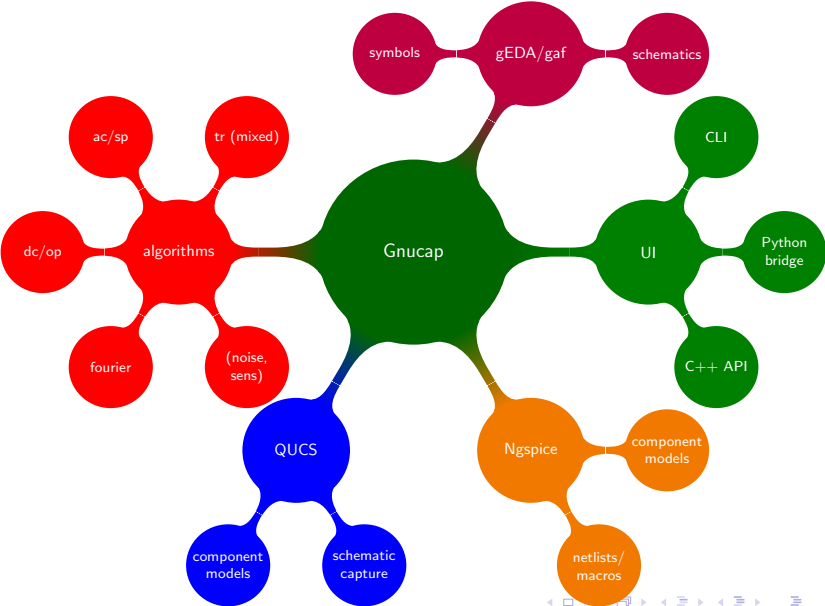
FSIC 22



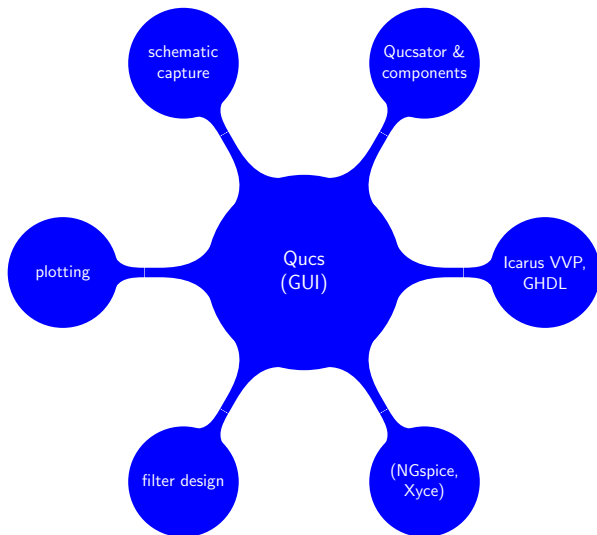
Content

- ▶ Gnucap & QUCS, what is it?
- ▶ Complementary features
- ▶ Motivation for a merge
- ▶ Recent Prototype work
- ▶ Further roadmap
- ▶ Help needed

Gnucap, overview



QUCS – Quite Universal Circuit Simulator



Complementary features

Gnucap

- ▶ Modern architecture (Post-Spice, C++)
- ▶ Plugin defined applications
- ▶ Foreign model support
- ▶ Interactive UI

QUCS

- ▶ Qt GUI, schematic editor
- ▶ run Simulator & alternatives
- ▶ Component library
 - many real world devices
 - macros and parameter sets
- ▶ filter design tools

Complementary features

Gnucap

- ▶ Modern architecture (Post-Spice, C++)
- ▶ Plugin defined applications
- ▶ Foreign model support
- ▶ Interactive UI

QUCS

- ▶ Qt GUI, schematic editor
- ▶ run Simulator & alternatives
- ▶ Component library
 - many real world devices
 - macros and parameter sets
- ▶ filter design tools

Really want

- ▶ The union of the above
- ▶ more

Rationale behind the merge

- ▶ Want the union of features
- ▶ Synergies
 - Qucs with plugins
 - GUI for Gnucap
 - Python bridge for Qucs
- ▶ Stop catch up between Gnucap, Qucs, Ngspice
- ▶ Need some degree of teamwork in the long run
- ▶ Integrate more projects
- ▶ Know what to choose from, what to contribute to

Rationale behind the merge

- ▶ Want the union of features
- ▶ Synergies
 - Qucs with plugins
 - GUI for Gnuicap
 - Python bridge for Qucs
- ▶ Stop catch up between Gnuicap, Qucs, Ngspice
- ▶ Need some degree of teamwork in the long run
- ▶ Integrate more projects
- ▶ Know what to choose from, what to contribute to

Towards the merge of the two

- ▶ Gnucsator
- ▶ modular Qucs

Prototypes towards the merge

Gnucsator

- ▶ Gnucap plugins
- ▶ Wrapper for some Qucsator components
- ▶ Read/write Qucs netlists & data
- ▶ Constitutes Qucsator replacement
- ▶ Few incremental releases

Prototypes towards the merge

Gnucsator

- ▶ Gnucap plugins
- ▶ Wrapper for some Qucsator components
- ▶ Read/write Qucs netlists & data
- ▶ Constitutes Qucsator replacement
- ▶ Few incremental releases

modular Qucs

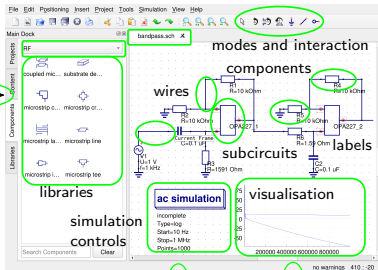
- ▶ Refactor Qucs (long overdue)
- ▶ Overcome Qt3 deadlock
- ▶ Explore plugin architecture
- ▶ Extend interoperability

modular Qucs now

- ▶ Looks like "legacy Qucs", but uses Qt5
- ▶ Explicit circuit underneath the drawing
- ▶ File format plugins: Both, legacy Qucs and Verilog
- ▶ Interaction/dialogs/graphics are plugins
- ▶ Simulation is negotiated through plugins
- ▶ Schematic editor is a plugin
- ▶ Data is managed by plugins
- ▶ Visualisation (plot) plugins
- ▶ "multi view" components are plugin defined.
- ▶ Has a command line interface.
- ▶ Lacks some of the Qucs features.
- ▶ Plugins don't work with Gnucap

modular Qucs now

- Devices
- Legacy (C++)
 - Legacy "lib" files
 - "multi view"
 - (any)



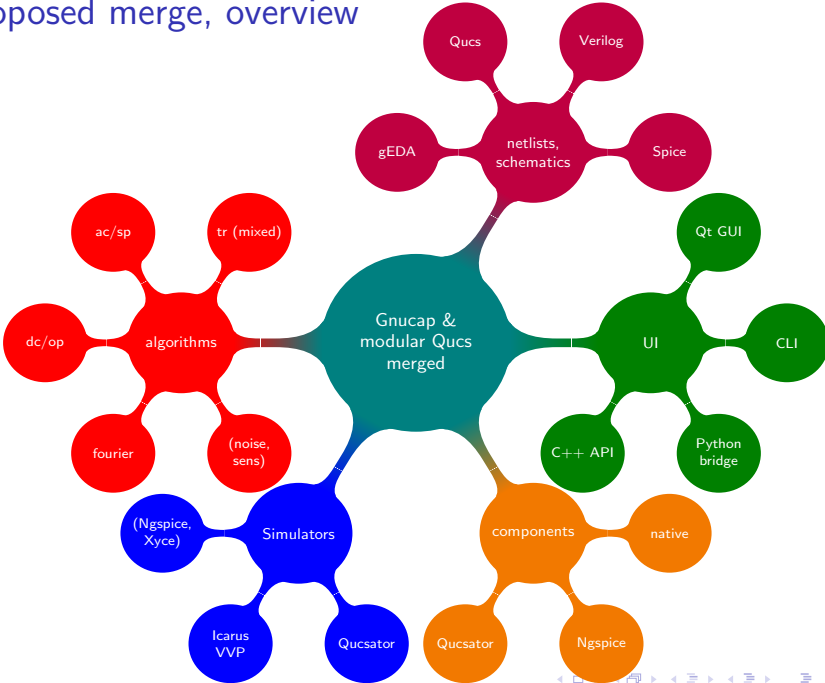
- Schematic file
- Legacy
 - Verilog
 - (any)

- Netlist file
- Qucsator
 - Verilog
 - Spice ...
 - (any)

- Simulation
- Subprocess
 - Shared library
 - (any)

- Data
- Legacy ("dat")
 - hdf5 (wip)
 - shared mem (wip)
 - (any)

Proposed merge, overview



The technical side, how to do it?

- ▶ Tweak Gnucap a bit
 - define interaction patterns
 - add some hooks
- ▶ Decruft/un-hack modular Qucs
 - decouple GUI from library
 - add back some code into plugins
- ▶ conform the libraries
... until modular Qucs equals Gnucap
- ▶ Eventually fold in
 - Gnucsator models
 - code from Qucs forks, as plugins

Thanks to past contributors

- ▶ Gnucsator
 - Fabian Vallon (first steps)
 - Szymon Blachuta (.dat file creation)
 - Dow Drake (loads of testing)
 - Daniel Mulholland (testing)
- ▶ gnuicap-geda
 - Karl Hammar (gEDA symbol insights)
 - Cheng Fei Phung (example schematics)
- ▶ gnuicap-python
 - Henrik Johansson (idea, first steps)
 - Patrick Mulder (early testing)
 - Ruben Undheim (help with Debian package)
- ▶ modular Qucs testing & discussions
 - Michał Walenciak
 - Martin Marmsoler
 - Guilherme Brondani Torri

Help needed

- ▶ Try, report issues
- ▶ write Documentation
- ▶ provide Examples
- ▶ Contribute your own plugins
 - Spice driver(s)
 - gEDA components
 - more file formats
 - [any, really]
- ▶ How to secure funding?
 - Gnuicap/Qucs merge
 - Verilog/AMS support
 - Filling the gaps

What are plugins again?

- ▶ Single file, distinct purpose
- ▶ Mutually independent, no interference
- ▶ Dynamically loaded (`dlopen`)
- ▶ Well known from Linux, Python, Gnucap, ...
- ▶ Even Qt5 plays with plugins
- ▶ Enable decentralised development
- ▶ Avoid license issues
- ▶ Critical: sensible core library underneath