

Gnucap – Progress and Opportunities

Felix Salfelder

FSIC 19



UNIVERSITY OF LEEDS

Gnucap – Progress and Opportunities

Building bridges

- ▶ Gnucap-Python
- ▶ Gnucsator & QUCS
- ▶ Gnucap-Custom
- ▶ Output plugins
- ▶ Projects & Teaching

Gnucap-Python, a SWIG interface

- ▶ Motivation
 - ▶ Gnucap accessibility
 - ▶ Fun & easier plug'n'play
 - ▶ Escape niche languages
- ▶ Scripting
 - ▶ "notebook" environment
 - ▶ Visualisation, plotting
 - ▶ Modelling / algorithms
- ▶ Interfaces to
 - ▶ Python libraries support numpy
 - ▶ scipy, scikit, pylab, fftw, ...
 - ▶ Anything else possible

Gnucap-Python: details

- ▶ First stub: Henrik Johansson 2009
- ▶ SWIG C++ ↔ Python interface
- ▶ shared symbol space & consistent class hierarchy
- ▶ (SWIG supports 25 more languages... octave?)

Usage Patterns

- ▶ Load python code into simulator kernel
- ▶ Customize and distribute extensions
- ▶ Ready to share, no compilation required
- ▶ Avoid platform issues

Gnucap-Python, examples & numbers

- ▶ S-Parameter analysis command port
 - ▶ quite short and simple
- ▶ Pole/Zero command
 - ▶ Suitable libraries in place
- ▶ Parameter optimisation
 - ▶ It cant be any simpler
- ▶ Possibilities and Contributions
 - ▶ Language support
 - ▶ File formats (Touchstone?)
 - ▶ GUI bindings
 - ▶ more commands, verification?

Gnucsator

- ▶ QUCS: a Qt GUI simulation tool, runs *Qucsator*
- ▶ Gnucsator: a drop-in replacement
- ▶ Gnuicap + a few plugins + component library
- ▶ Issues
 - ▶ not all components available
 - ▶ harmonic balance is missing
 - ▶ some details unresolved
- ▶ Chances
 - ▶ stable solver, mixed signal etc.
 - ▶ run your own components & analysis
 - ▶ extensible and growing
 - ▶ approach is not tailored to QUCS

more about QUCS

- ▶ Qucsator + GUI (Qt)
- ▶ Netlist format close to verilog
- ▶ Model library
 - ▶ Macros and parameter sets
 - ▶ Many real world devices

Opportunities

- ▶ There is nothing like QUCS
- ▶ Based on Qt3, no longer supported
- ▶ Porting to Qt5 is difficult
- ▶ Who is going to do it?

Gnucap-Custom

- ▶ alternative executable (inspired by gnucsator needs etc.)
- ▶ links to libgnucap
- ▶ Per application rc, *freedom of choice*
- ▶ batteries/examples included (all optional)
 - ▶ verilog preprocessor plugin
 - ▶ modified 'include command and path logic
 - ▶ make-attach: compile plugins upon load
 - ▶ readline & history (stub)
- ▶ Not upstream: It doesn't have to be.

Output plugins

almost there

- ▶ original goal: choose output format
- ▶ command sends results to a plugin
- ▶ take it from there, do whatever
hdf5, compress, encrypt ...
- ▶ Play nicely with Python/any UI

side effects, difficulties, opportunities

- ▶ Generic data stash (currently "waves")
- ▶ Command "nesting" (fourier, shooting, pss...)
- ▶ Pluggable and extensible probes (sens, sparam)
- ▶ Set of plugins, but must become default

Projects and Teaching

Students...

- ▶ Strengthen your Software Skills
- ▶ Evade vendor trap and lock-in
- ▶ Implement a plugin, learn by doing
- ▶ Solve a real world problem

Teachers...

- ▶ From electrical level upwards
- ▶ Focus on concepts and algorithms
- ▶ Dissect and rebuild tools
- ▶ Make it simple, use current technology

Opportunities – get involved

Your own extensions

- ▶ Plugin – ready to go
- ▶ No merge window, no review
- ▶ .. and completely yours in your way
- ▶ Anything from parsing to algorithms

Core library contributions

- ▶ Much more difficult
- ▶ Hardly ever needed
- ▶ Bug reports are still welcome

Opportunities – summary

The ecosystem is the sum of its parts

- ▶ SystemC/AMS wrapper
- ▶ Deployment & Packageing
- ▶ QUCS component library
- ▶ Refresh (ng)spice models
- ▶ Supply real world problems or benchmarks
- ▶ Tutorials & walkthroughs
- ▶ Test drive work in progress

Thank You.

