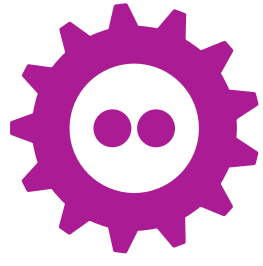
Verilog-AMS in Gnucap

Felix Salfelder

FOSDEM '26

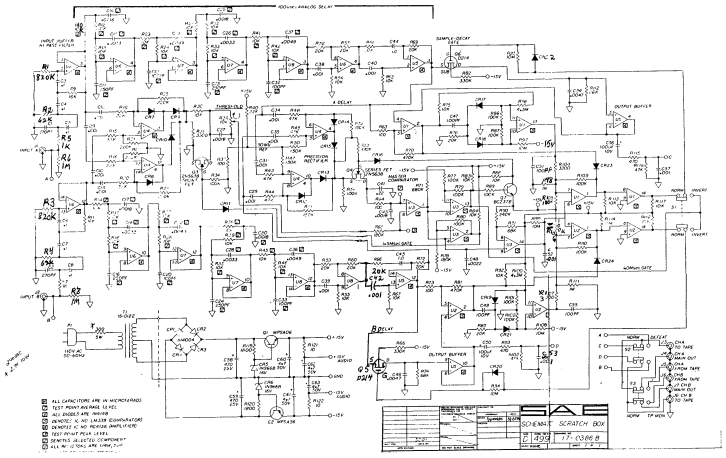


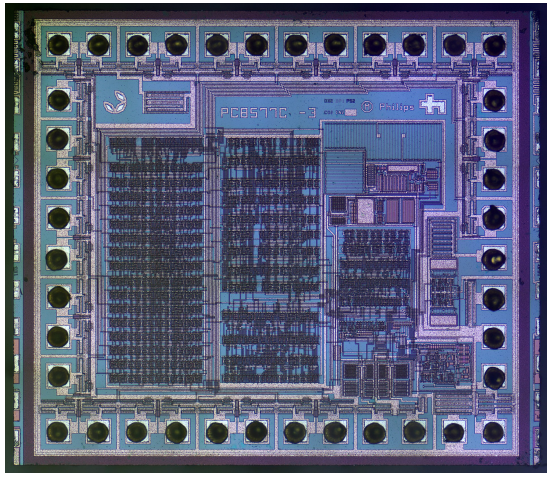
GNU Circuit Analysis Package

- Small C++ library for circuit analysis
- Plugins (devices, algorithms, languages...)
- (Originally) run on hardware too small for SPICE
- Pre-Verilog Model compiler, modelgen.
- Beyond Spice – mixed techniques for fast analog

- 1990. *ACS*, Al's Circuit Simulator
- 2001. *Gnucap*, a GNU project, GPLv3+
- 2008–2010. Move to Plugins
- from 2022: NLnet funding for Verilog-AMS

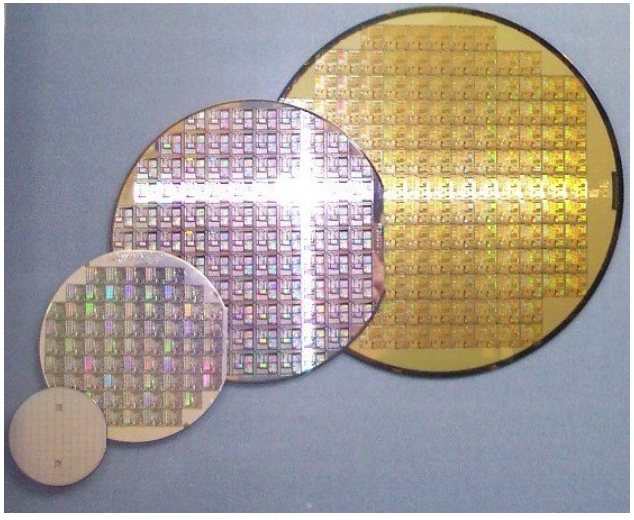
Circuits and why?





.. simulate the behaviour

- often part of the design process.
- Then somehow make more of them.



images from wikipedia

Why Verilog, IEEE1364 & friends

A standard language for circuit and device modelling

- "Open Hardware" needs a source language
- Analog, digital modelling and in between
- Verilog-95 and "Verilog-A" have become popular
- Standards define terminology ... and best practice

This one: Widely used in industry

- IEEE1364: 590 pages, 40 contributors, 12 EDA companies
- Verilog-AMS: 442 pages, 74 contributurs from 16 companies
- System-Verilog: 586 pages ...

Now trying not to reinvent the wheel

What are plugins again?

- Dynamically loaded (`dlopen`) extensions
- Linux `insmod`, Python `import`
- Stable public interface
- Decentralised development, no interference
- Unfinished work available for use/testing/completion.

Some existing examples

- Input languages: Verilog, SPICE, "Spectre", gEDA, Qucsator
- Component models: primitives, modelgen, SPICE, Qucsator
- Commands: ac, dc, tran, fourier, pz, sparam, postprocessing..
- Solvers: (C)BS, KLU. Also: node ordering

Your project here.

NLnet funded work & summary

Gnucap Project goals

- Verilog support in the simulator
- Model compiler for Verilog-AMS
- Standardisation and compatibility

Topics for Today:

- I) File formats for data interchange
- II) recent Simulator improvements
- III) Modelgen-Verilog and recent improvements
- IV) Testing, QA and work in progress

I a File format for data interchange

Some facts

- Multiple tools needed to build a chip (or smartphone)
- Yet another file format with every tool
- Getting a circuit across can be a pain
- Most of the time spent validating data exports (and getting used to wild conventions)
- Often conversions are one-way only

Example: Circuit Schematics

- Schematics are circuits with coordinates attached
- .. and text, and drawings.
- Verilog syntax permits meta data
- Always the same circuit, no matter which tool

I b File format for data interchange

Verilog-S, trajectory

- 2012: S. Krishna: gEDA parser for Gnucap (gsoc)
- FOSDEM'16: AI's Talk, Proposal for Data Interchange
- since '23 NLnet funded work on details and spec.
- '25: `same_port` in Gnucap and modelgen-verilog
- mid '25: Verilog-S support in Qucs (David L.)
- late '25: improved non-circuit items in gnucap-geda

More to do

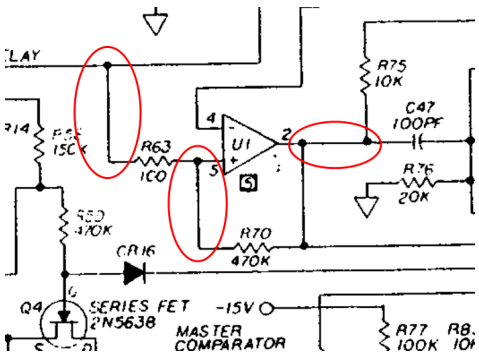
- Native Verilog-S in Lepton (former gEDA)
- Looking for a Qucs maintainer/developper
- Can somebody do xschem, KiCAD please?

I c File format for data interchange

`same_port` device?

- Represent nets (e.g. in Schematics, PCBs, Layout)
- Nets are first class objects
- Part of the standard since Verilog-95
- Available in Icarus-Verilog since .. forever?
- Unavailable in SPICE tools & friends
- Perhaps intended in Qucs, now used in schematic files

```
module same_port (.a(i), .b(i));
  // that's it
endmodule
```



II Simulator

Recent improvements

- Circuit topology (aka union-find)
- Integration method selection
- Data shared across identical devices (automatically)
- Node ordering: now pluggable
- Improved matrix interface (adds: KLU)
- Fix parse performance (esp. SPICE)

Working towards

- Event driven simulation (drafted)
- Nodes carry disciplines (conservative and signal flow)
- Automatic partitioning. Catch up on standard compliance.
- Disciplines determine interconnect models

II a Circuit topology algorithms

Nets

- may be modelled as shorts, resitors, transmission lines etc.
- Depends on the application, accuracy requirements etc.
- `module net (.a(i), .b(i));`
- Often used in Schematic drawings, Verilog-S..
- New: Union-find algorithm finds effective nodes
- Side effect: ground and SPICE style “global” nets

II b Integration method selection

A problem with analog simulation (in a nutshell)

- Different integration methods, e. g.
- *Trapezoidal*: nice for smooth signals
  allows for large timesteps
  but rings after discontinuities
- *Gear*: nicely damps out ringing
  but dissipates energy
- Each device is different. Need to make choices

- Per-device method in Gnucap since .. forever
- Tricks won't work (Has to do with maths)
- Selection now exposed to the user (Verilog extension)
- (prereq for large mixed simulations)

III 1 Modelgen-Verilog

The model compiler for Gnucap

- Generate C++ from model description
- .. to build a device plugin
- `modelgen`: analog only, but predates Verilog.
- Most of Verilog-A covered
- growing Verilog-AMS support, e.g.
  user defined primitives ("lookup tables")

Recently added

- Connectivity overhaul, $same_port...
- Random number geration, "$rdist_" functions
- String and integer support
- Hierarchical reference resolution
- Storage type inference (cf algorithms)

III 2 Modelgen-Verilog, storage type

In Verilog, specifically -AMS, variables

- can represent device state,
- some may be shared across identical instances
- others are temporary/redundant

depending on their assign/use pattern.

Modelgen-Verilog now

- Identifies the storage type (heuristically)
- exact computation is not feasible or useful
- reduces overhead in component storage
- improved cache locality, memory footprint ..
- Prerequisite for mixed & VLSI

IV Testing, QA and work in progress

- Continuous Integration @ codeberg
- Populate models repo with devices and tests
- Numerical cross validation
- Generic testbenches for semicondutor models
- OSDI wrapper
- IHP PDK, Verilog-AMS port

IV 1 Continous Integration

Thanks to Eric @ CCT

- CI runner set up on Codeberg
- “Modernise” official contribution process
- Repo structure needs work
- (Need contributors, still).

IV 2 Models and validation

gnucap-models repo: device collections for Gnucap

- from Spice3f5, jspice3-25 (historic, unmodified)
- ngspice-17 .. current
- devices from berkeley (bsim3, 4)
- psp, bsim4va, rram, .. (Verilog, WIP)

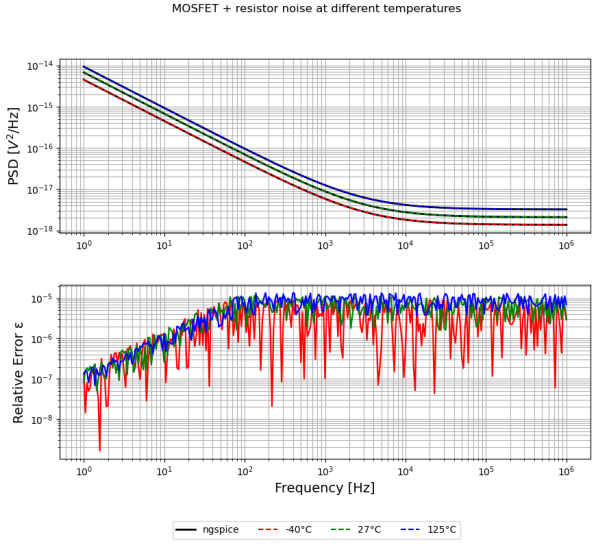
Purpose:

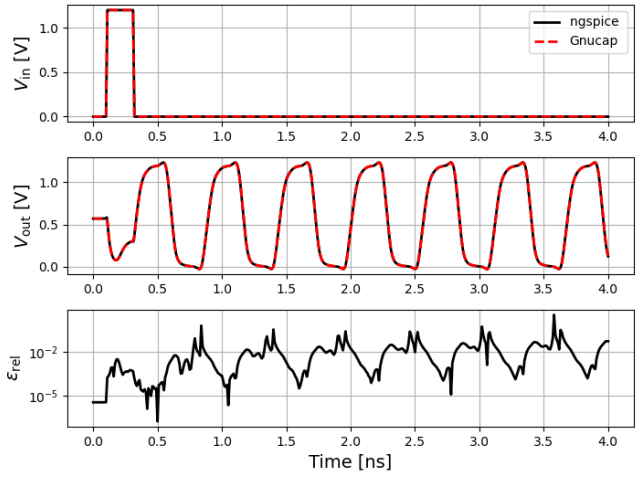
- Archeology – what did you get in 1990?
- Regression testing against reference figures
- Validation (models and new solvers)
- Find bugs (ngSpice: 1, modelgen: 2.)

IV 3 Numerical cross validation

Thanks to Lukas D & NLnet

- Comparison against other FOSS tools





- Goal: standard test procedures that work with known simulators.
- Method: Verilog-S, map-to-application.
- Separate NLnet funded project intended

IV 4 OSDI wrapper (OpenVAF)

OpenVAF: another model compiler

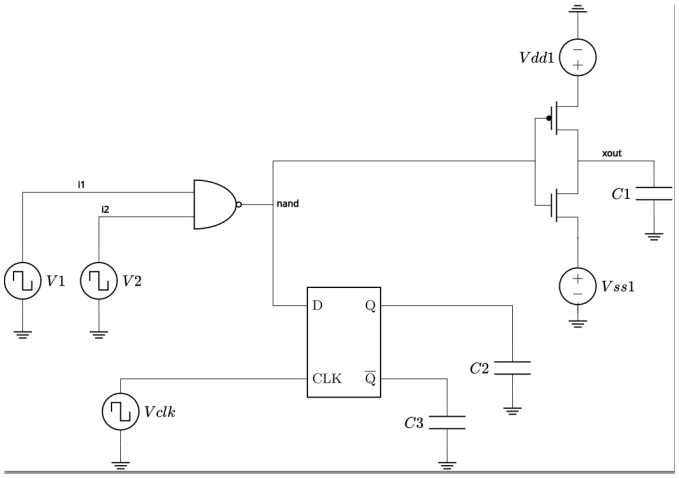
- Pretty fast compilation.
- Obnoxiously big ("compact") models
- Basically, SPICE subset of "Verilog-A"
- Very popular in analog simulation

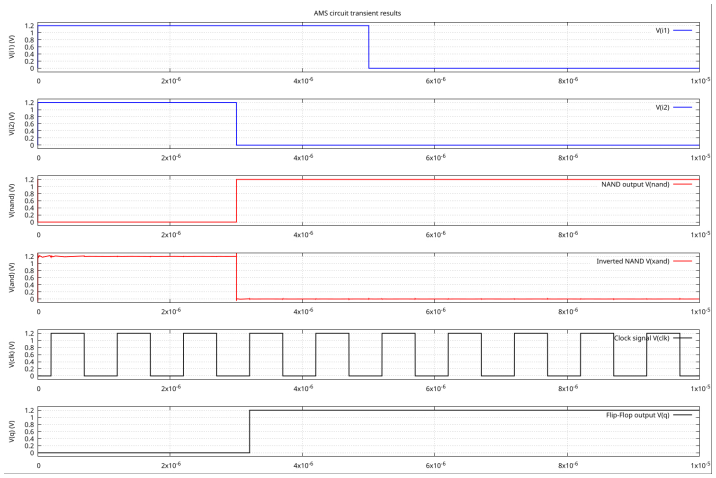
Use OSDI in Gnucap (Lukas, WIP)

- Devices are plugins, use one or the other
- Mechanism modelled after spice-wrapper
- All features combined in one framework

IV 5 IHP PDK Verilog-AMS port

- Work in progress with Krzystof @ IHP
- Initiative: Pepijn D (IHP/#725)





- Semiconductor device library
- Mixed signal cell library
- Statistical modelling, mismatch analysis
- Towards official support for Gnucap

Thanks

Get started

- Wiki `http://gnucap.org`
- Mailing list `gnucap-devel@gnu.org`
- Chat `#gnucap:matrix.org`

More work to do

- Plugins, wrappers, extensions
- Data exchange, standard compliant tools
- Funding (still) available