PALMA DE MALLORCA
7-10 SEPTEMBER
ESSERC 2026

UNIVERSITAT ROVIRA i VIRGILI

UIB Universitat de les Illes Balears

Advances in Open-Source Verilog-A Compilation and Simulation

Árpád Bűrmen
University of Ljubljana,
Faculy of Electrical Engineering - Slovenia
arpad.buermen@fe.uni-lj.si

ODE 4 EC AMS

Chips JU

UNIVERSITY OF LJUBLJANA | FE Faculty of Electrical Engineering

Funded by the European Union

REPUBLIC OF SLOVENIA
MINISTRY OF THE INTERIOR
AND PUBLIC ADMINISTRATION

ARIS

IEEE SSCS Electron Devices Society CAS

Outline

- OpenVAF-Reloaded
  - History
  - Recent progress
  - Missing features
  - LLM-assisted development
- The open-source simulation ecosystem around OpenVAF-Reloaded
- VACASK
  - Performance and scalability
  - API and Python interface
  - Monte Carlo and behavioral sources
- New analyses
  - AC stability and S-parameters
  - Transient noise
  - HBAC and PSS
- Outlook

OpenVAF-Reloaded

- Verilog-A compiler
- Author: Pascal Kuthe, SemiMod GmbH
- Development stopped in 2023
- OpenVAF-Reloaded continues where OpenVAF stopped
- **https://github.com/OpenVAF-Reloaded/OpenVAF**
- Written in Rust, uses LLVM library
- Compiles all CMC models without manual intervention
- Fast compilation, fast models
- Used by Ngspice, VACASK, and several experimental simulators
- Latest semiconductor device models distributed as Verilog-A
- Opens the door for PSP, BSIMBULK, BSIM-IMG, BSIM-CMG, ...

Progress in 2026

- macOS (R. Taylor, J. Verley)
- LLVM 18-21 (Kreijstal, R. Taylor)
- Exposes natures/disciplines – VACASK multiphysics
- Bugs fixed:
  - Better CI (R. Taylor)
  - Loop hoisting
  - Derivatives optimization
  - parameter defaults
- Module flags
- Started work on OSDI 0.5

Missing features

- Silicon photonics
  - absdelay()
  - Vector ports
  - Submodule instantiation
- Memristor models
  - Hidden state support
  - Random generators
  - Proper handling of intial_step and final_step
- Vector parameters
- transition, slew, last_crossing
- Laplace and Z-transform filters
- Events
- Table models, table noise models
- Paramsets
- ...

Using LLMs for development

- OpenVAF code is highly complex, hard to master
- LLMs great for reverse-engineering, can implement new features
- Need precise instructions and supervision
- Can produce inefficient or even wrong code
- Automated tests are not enough!
- Author must
  - be familiar with simulation – simplify API use for simulator,
  - know advanced analyses (PSS, transient noise, ...) - avoid introducing blockers.
- LLMs cannot make design decisions.
- “It compiles and produces a DLL with an API” is just not enough!
- Several repositories attempt to extend OpenVAF using LLMs
- Best and most complete: Meisam Bahadori’s repository
- https://github.com/javaNoviceProgrammer/Ngspice_OpenVAF_Enhancements

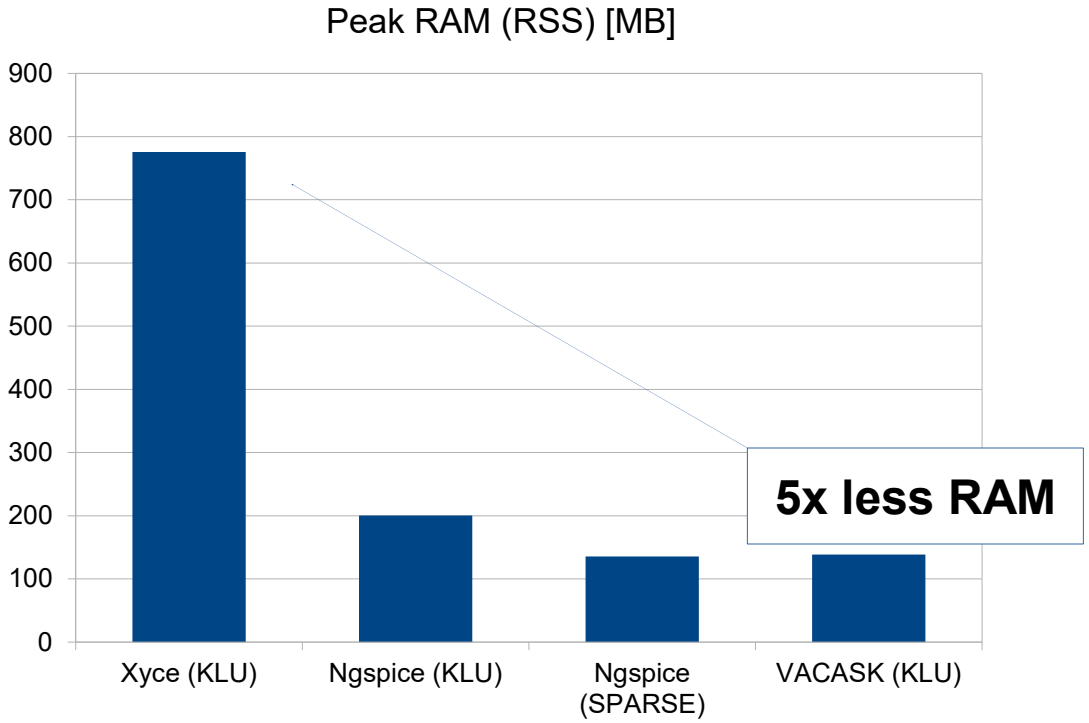
Meisam Bahadori's extensions

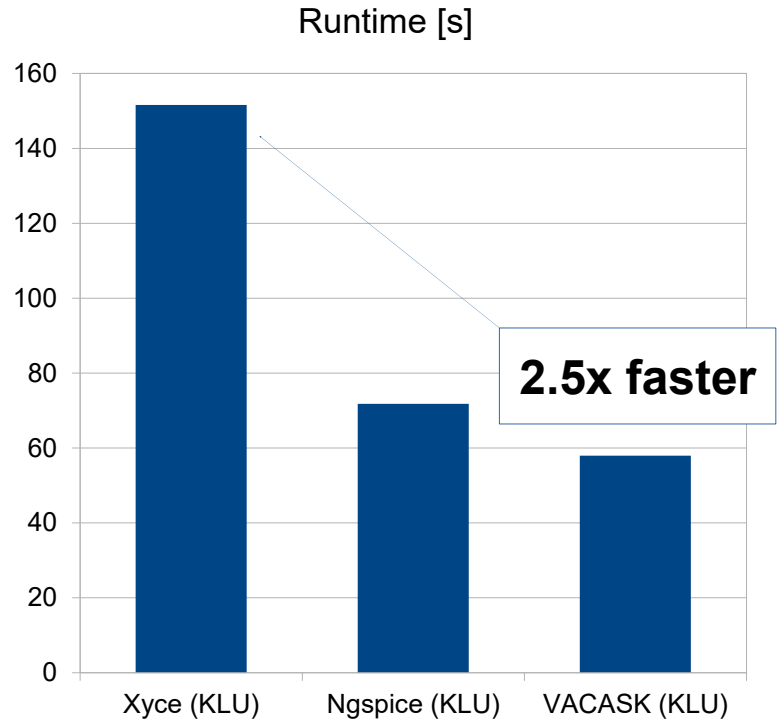
- Based on OpenVAF-Reloaded and Ngspice 46
- 93 OpenVAF enhancements, 150+ robustness improvements
- Used Claude Code as assistant
- Hidden landmines – example: absdelay()
  - Hardcoded structure size in Ngspice instead of sizeof()
  - Does not expose max_delay, cannot implement a LRM compliant simulator
  - History arrays can grow to infinite size
- Blockers
  - Noise handling merges ddt() and noise sources – makes transient noise complicated
- Wrong implementation
  - Hidden state changes between Newton-Raphson iterations
- Still a valuable resource
- Showed how to advance OpenVAF with the help of LLMs.
- Plan: clean up and merge with OpenVAF-Reloaded.

Ecosystem

- Ngspice simulator
  - Established SPICE-based simulator
  - Analog, optional mixed-mode
  - Supports OpenVAF (mostly OSDI 0.3 features)
  - https://ngspice.sourceforge.io/
- VACASK simulator
  - Modern FOSS simulator
  - Analog
  - Based on OpenVAF (uses all OSDI 0.4 features)
  - https://codeberg.org/arpadbuermen/VACASK
- Verilog-A Distiller model converter
  - Converts SPICE device models from C to Verilog-A
  - Most of SPICE device library converted into Verilog-A
  - https://codeberg.org/arpadbuermen/VADistiller

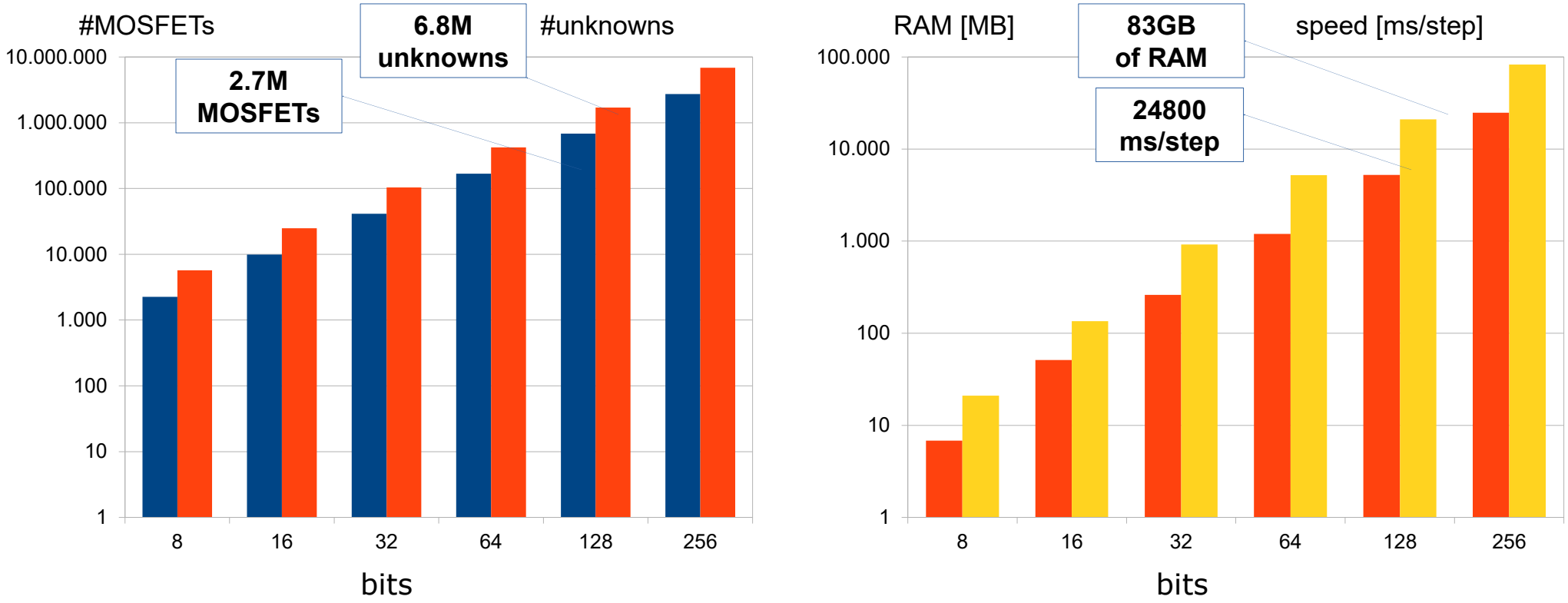
VACASK vs. Xyce and Ngspice – RAM & speed





- Tested on C6288 16x16 multiplier, simulated as an analog circuit
- 10112 transistors, 25380 nodes

VACASK - scalability

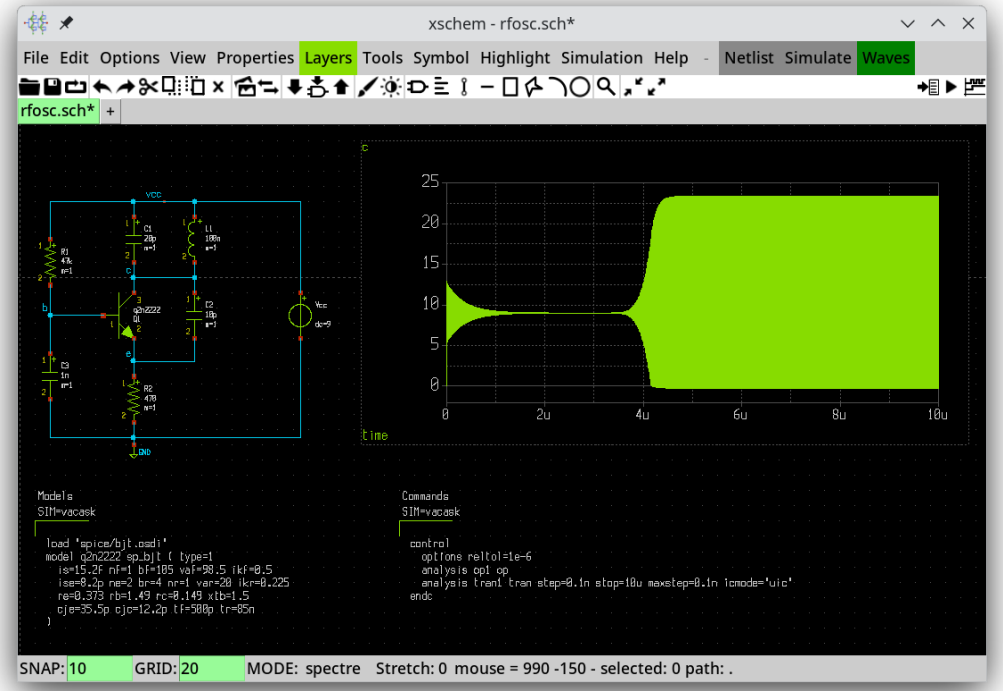


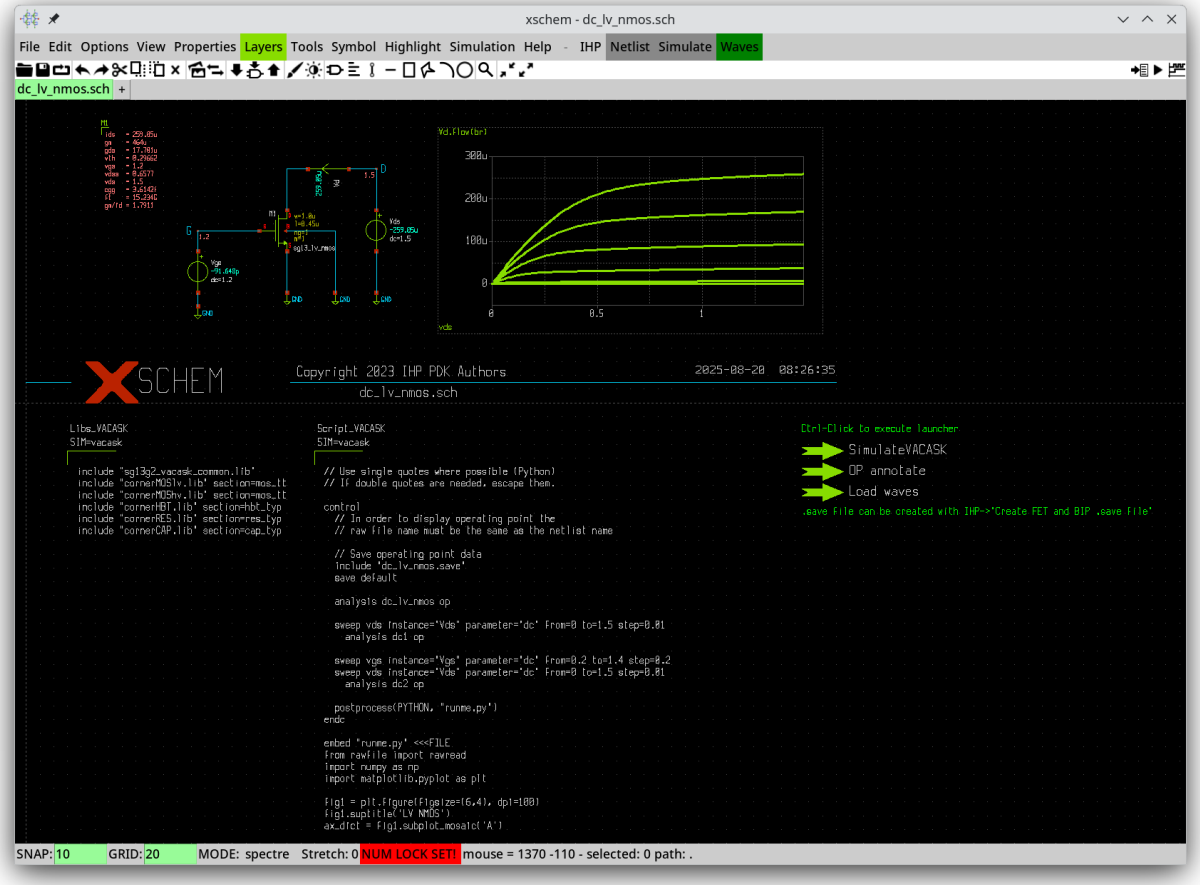
- Multipliers (8 to 256 bits), simulated as analog circuits
- Up to 2.7M MOSFETs, 6.8M unknowns
- Scales almost linearly, no nasty surprises

Netlist, converters, PDKs

- Spectre-like input netlist format
- Ngspice to VACASK converter, primarily for PDKs
- IHP SG13G2 – fully functional
- IHP SG13CMOS5L – fully functional
- Conversion of GF180 open PDK (WIP by Alfonso Cortés Neira)
- Proof of concept: conversion of SKY130 open PDK (Pepijn de Vos)
- Cadnip parsers for common simulators (WIP by Pepijn de Vos)
  Not converters - native parses that use VACASK's API.
  Will support native Ngspice Sky130 PDK.
  Dialects: HSPICE, Spectre, Pspice, Ngspice, ...

Xschem integration





IIC OSIC tools

- Docker container with FOSS chip design tools
- Maintained at JKU (Harald Pretl)
- OpenVAF-Reloaded, Ngspice, VACASK, and many others
- All open source PDKs
- Used extensively, produced many tapeouts
- https://github.com/iic-jku/iic-osic-tools
- **VACASK was used for RF design of a taped-out circuit! https://github.com/iic-jku/SG13CMOS_SPARX**

API and Python wrapper

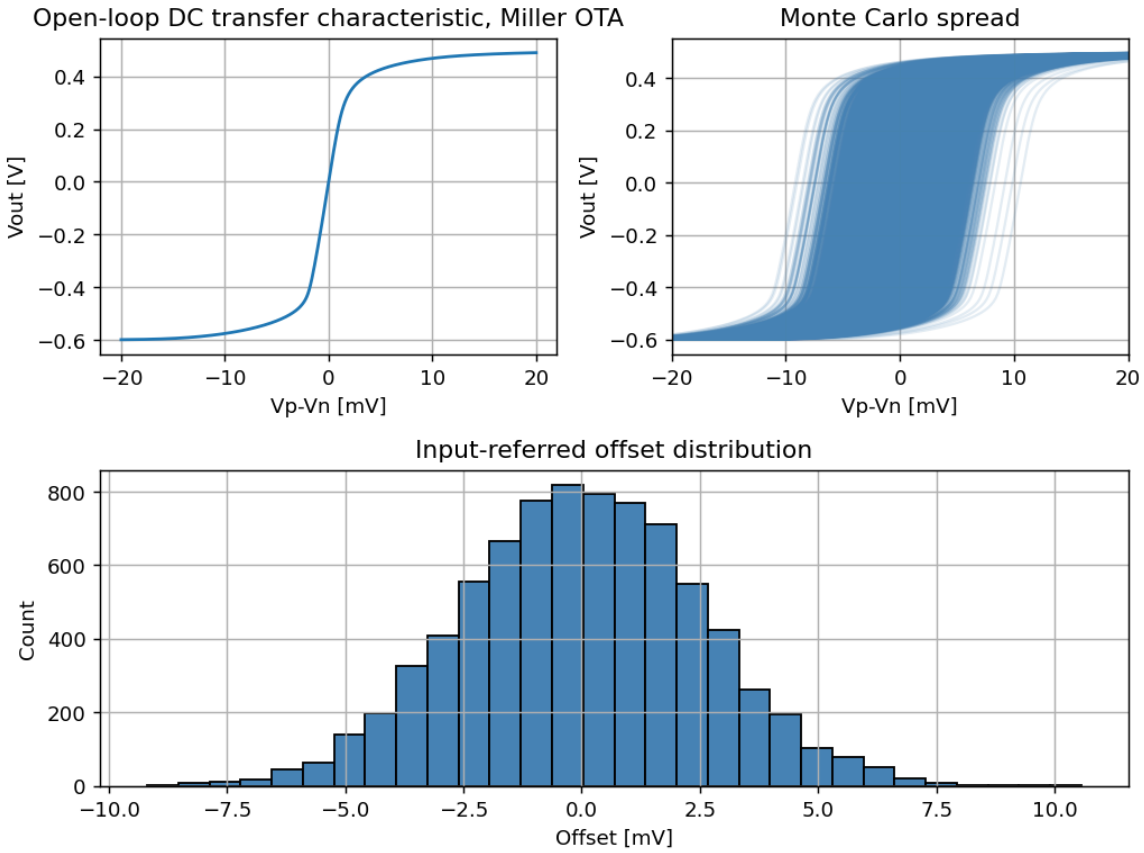
- VACASK as a library
- C++ API
- Simple, intuitive
- All capabilities exposed
- Alternative netlist parsers (HSPICE, Spectre, Pspice, ...)
  WIP by Pepijn de Vos
- Python wrapper
  WIP by Jure Vreča

```
Status s;
ParserTables tab("RC transient");
Parser p(tab);
tab.add(PTLoad("resistor.osdi"))
  .add(PTLoad("capacitor.osdi"))
  .defaultGround()
  .setDefaultSubDef( PTSubcircuitDefinition()
    .add(PTParameters(PVv( PV{"c0", 1e-6}, PV{"v0", 5} ), {}))
    .add(PTModel("res", "resistor"))
    .add(PTModel("cap", "capacitor"))
    .add(PTModel("vsrc", "vsource"))
    .add(PTInstance("r1", "res", {"1", "2"})
      .add(PV{"r", 1000})
    )
    .add(PTInstance("c1", "cap", {"2", "0"})
      .add(PE{"c", p.parseExpression("2*c0")})
    )
    .add(PTInstance("v1", "vsrc", {"1", "0"})
      .add(p.parseParameters(
        "type=\"pulse\" val0=0 val1=v0 delay=1m rise=1u fall=1u width=4m"
    )))
  );

OpenvafCompiler comp;
Circuit cir(tab, &comp, s);
cir.elaborate({}, "__topdef__", "__topinst__", nullptr, s);

auto tranDesc = PTAnalysis("tran1", "tran");
tranDesc.add(PV{"step", 1e-6}).add(PV("stop", 10e-3));
auto tran = Analysis::create(tranDesc, cir, s);
auto [ok, canResume] = tran->run(s);
```

Monte Carlo analysis

```
var ns=8000
mc mc1 samples=ns lh=1
  sweep vid instance="vid1" parameter="dc" \
       from=-10m to=10m mode="lin" points=200
    analysis dc2 op
endmc
postprocess(PYTHON, "runme.py", ns)
```

- Multiple analyses in loop
- Latin hypercube sampling
- Variations: Ngspice style
- Can postprocess sample by sample
- … or all at once after the loop
- Example: offset of a Miller OTA in IHP SG13G2



Behavioral sources

- Used by many PDKs – so we implemented them
- Quick and dirty modeling
- Fast: convert to Verilog-A module, compile with OpenVAF, use module
- Supports ddt() and idt() operators, white and flicker noise
- Voltage source – differentiator with temperature dependence

```
vmult (out 0) v=ddt(1m*(1+1e-3*($temp-27))*v(in))
```

- Current source – simple ad-hoc diode with white noise

```
d1 (a b) i=1e-12*(exp(v(a,b)/26m)-1) + $white_noise(2*P_Q*i(v1), “n1”)
```

- Specify Verilog-A discipline – thermal power source

```
pwr (t 0) flow=-(v(a,b)**2/r0) discipline=["thermal", "Temp", "Pwr"]
```

Analyses

Classical SPICE + HB

- Operating point (op)
- DC small-signal (dcinc)
- DC transfer function (dcxf)
- AC small-signal (ac)
- AC transfer function (acxf)
- AC small-signal noise (noise)
- Transient (tran)
- Harmonic balance (hb)

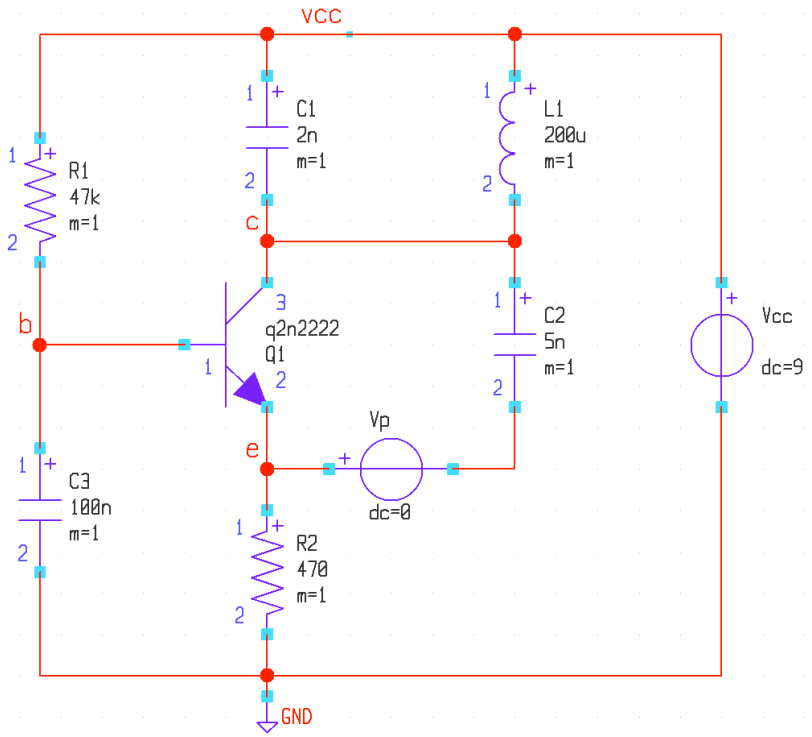
New

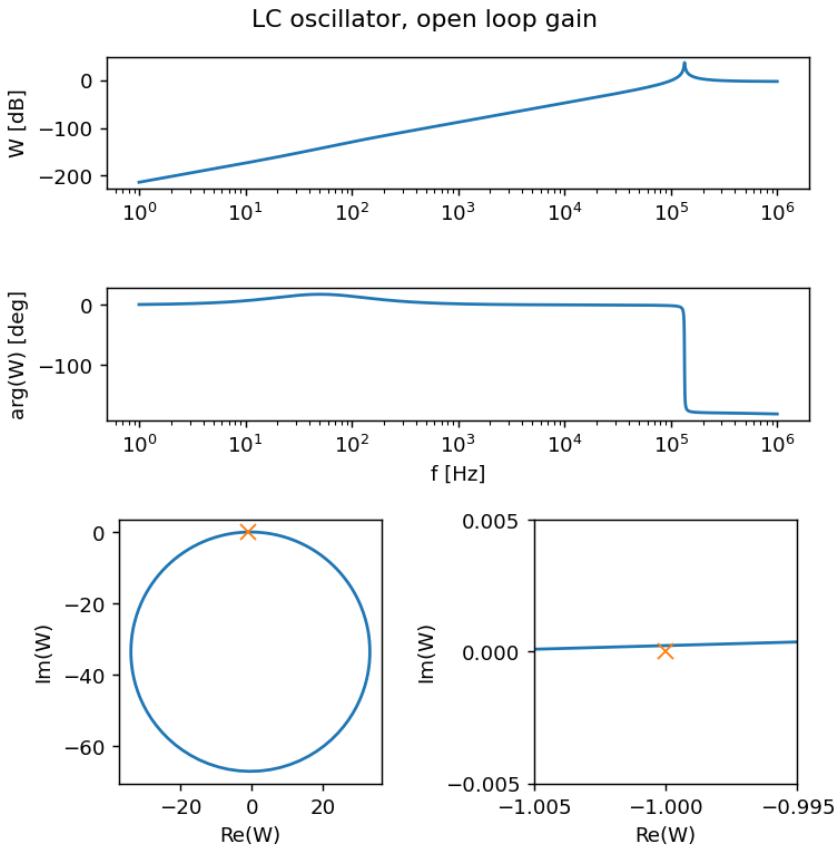
- AC stability - loop gain (acstb)
- AC S-parameters (acsp)
- Transient noise (tran)
- (Quasi)periodic small-signal (hbac)
- Periodic stedy-state (pss)

Planned

- Periodic small-signal (pac)
- Periodic noise – driven (pnoise)
- Periodic noise – oscillators (pnoise)

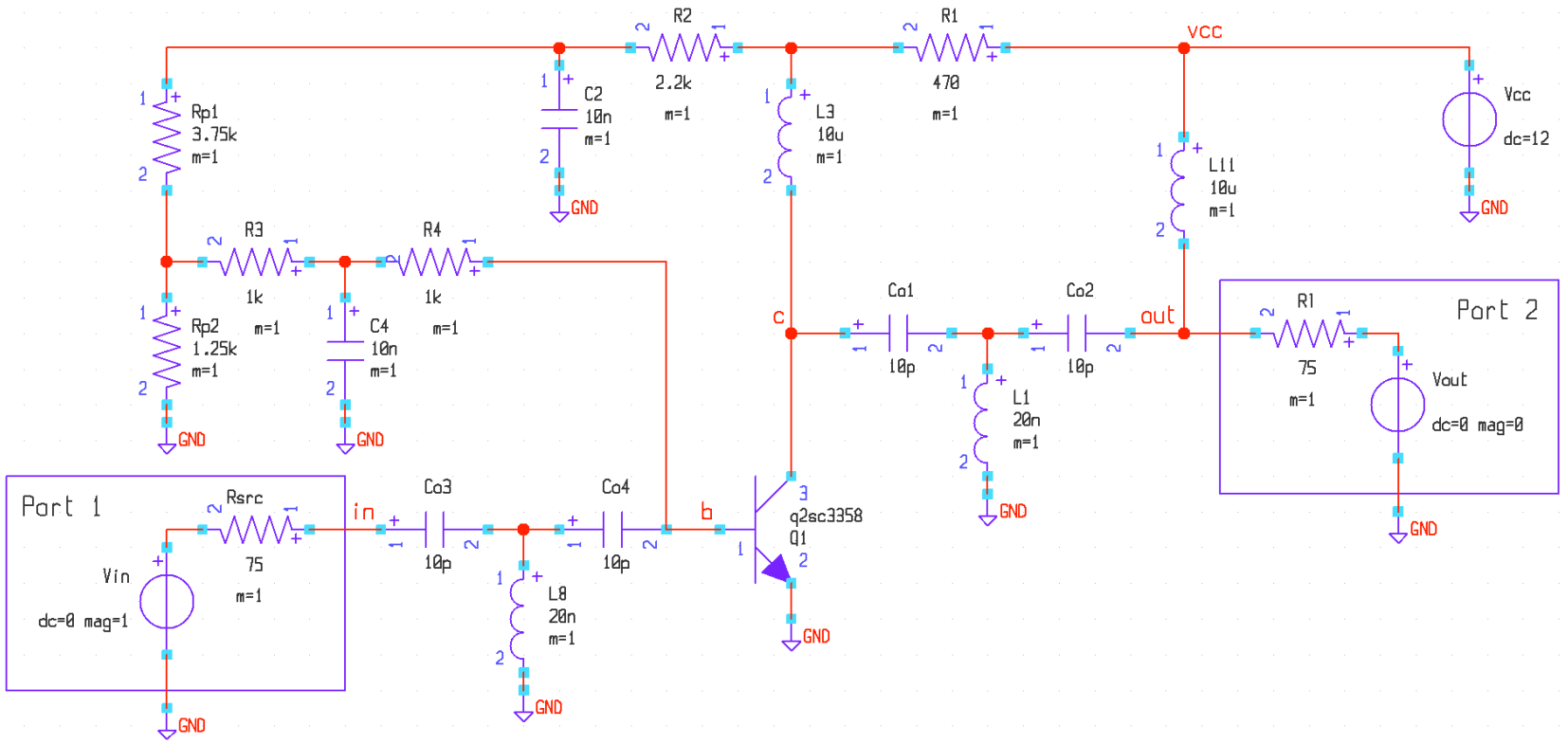
AC stability analysis (oscillator)





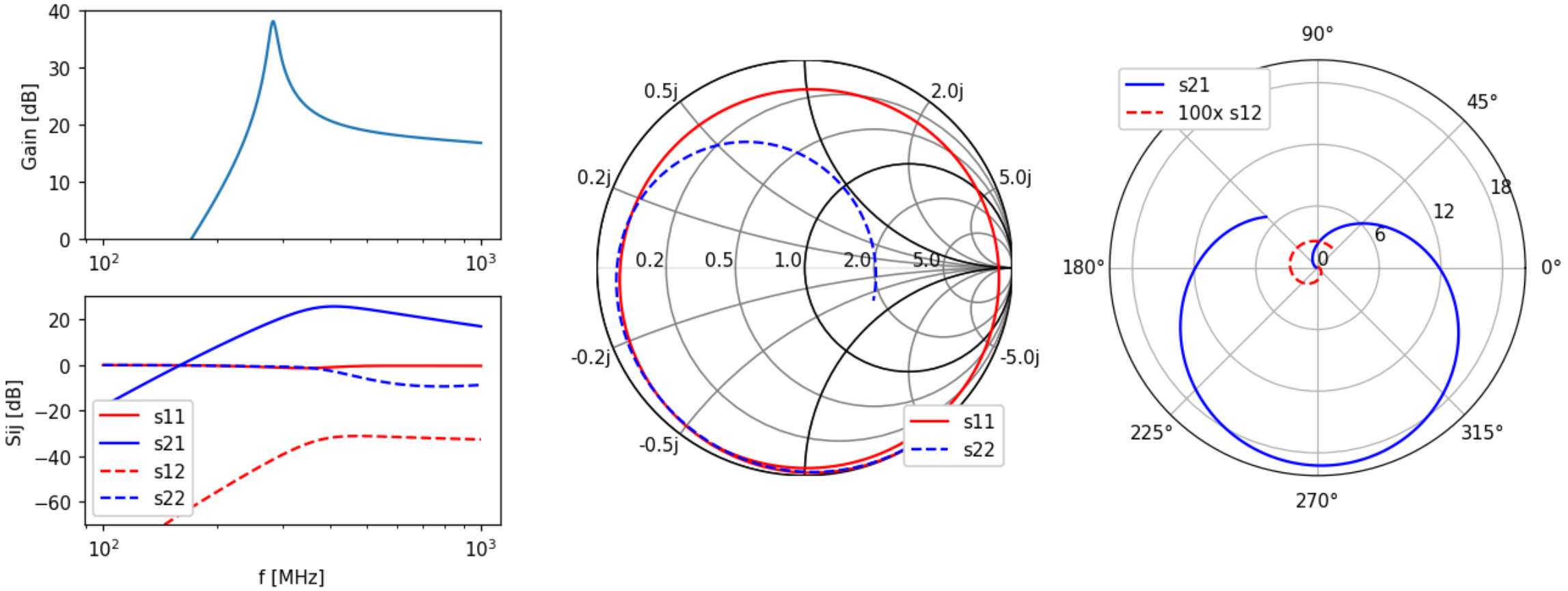
analysis stb1 acstb probe="Vp" from=1 to=1M mode="dec" points=10000

AC s-parameter analysis (UHF amplifier) ...

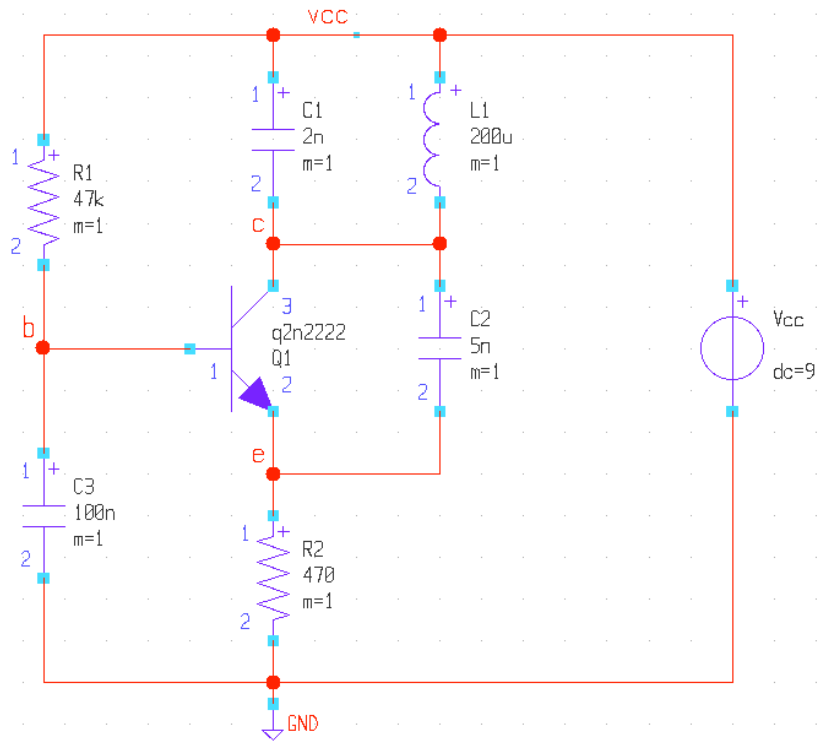


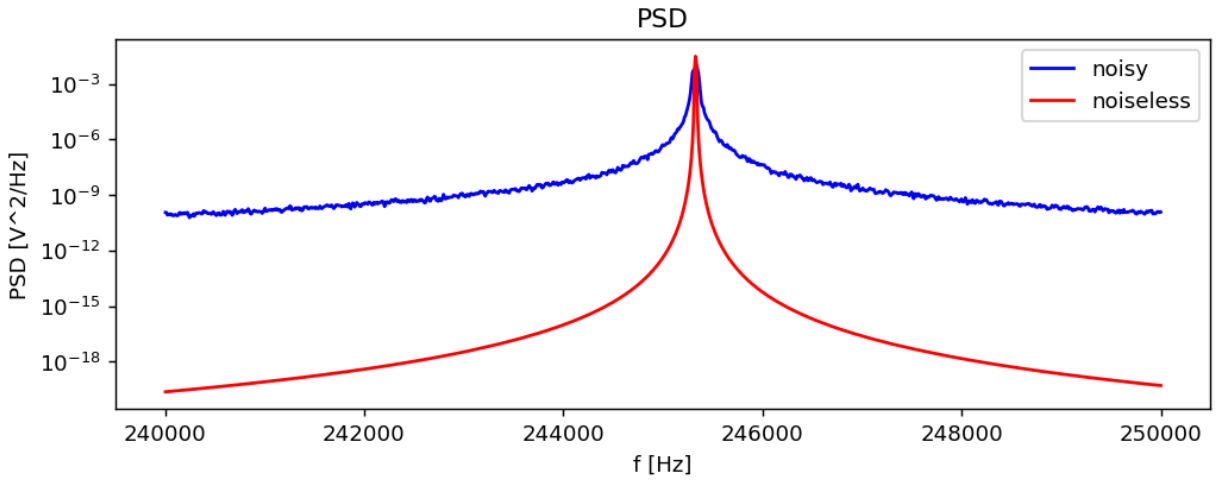

```
analysis sp1 acsp ports=["Vin", "Rsrc", "Vout", "Rl"] \
   from=100M to=10G mode="dec" points=100
```

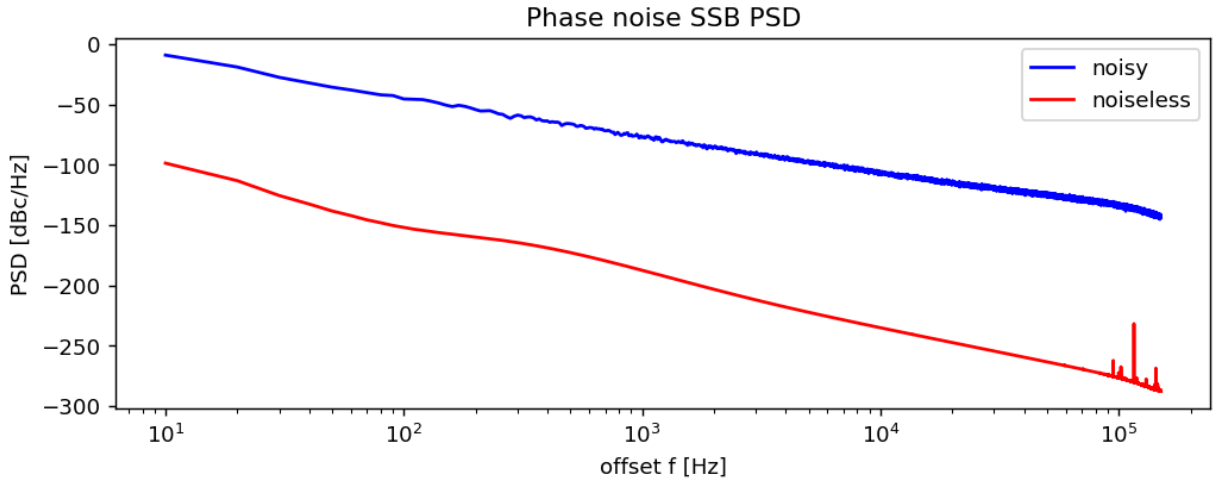
… AC s-parameter analysis (UHF amplifier)



Transient noise analysis ...

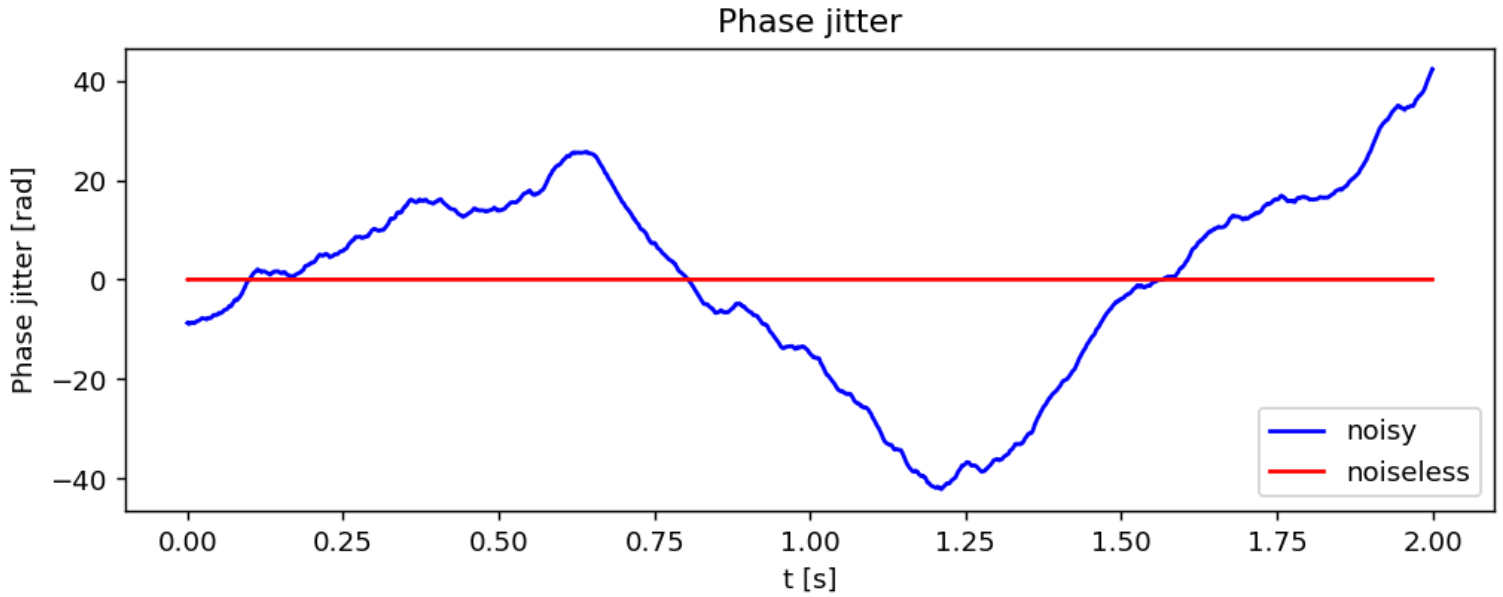






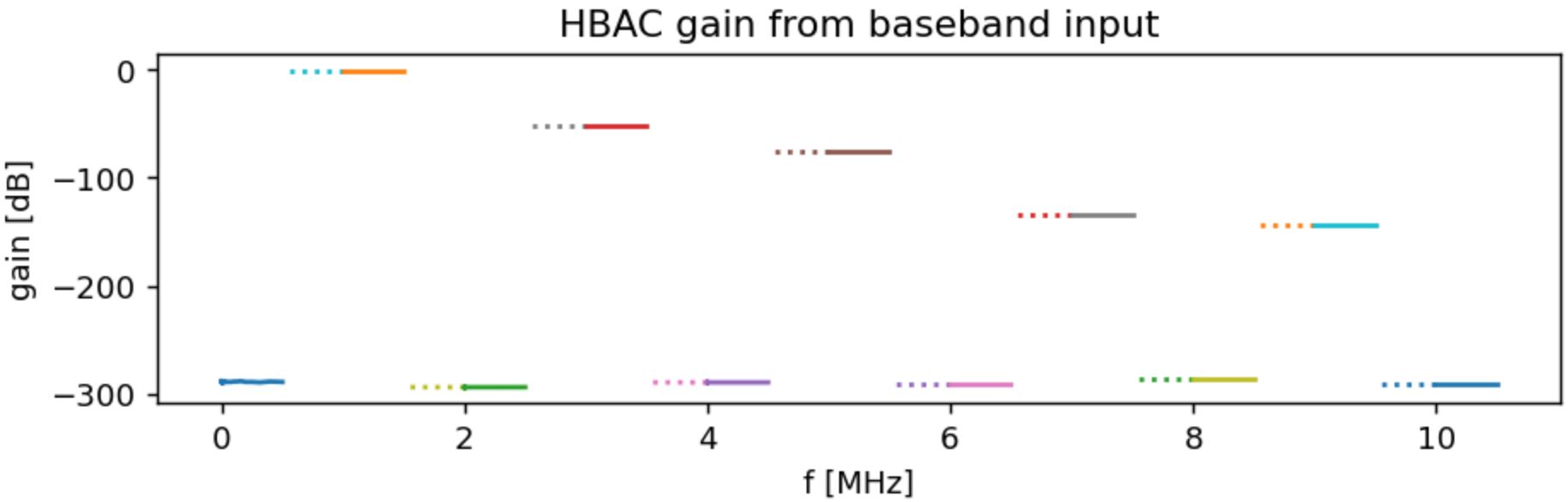
- Expensive!
- 245.4kHz oscillator, 10Hz..400kHz
- 81M points, 5 minutes.

… transient noise analysis



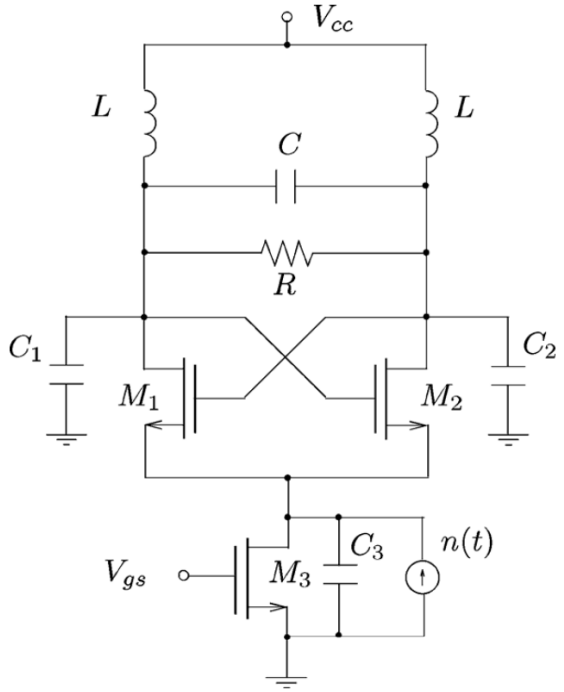
- FIR bandpass around fosc (BW=50kHz)
- Hilbert xform
- Subtract linear phase … and we have jitter
- Thanks to Filip Maksimović for evaluation and SNR improvement

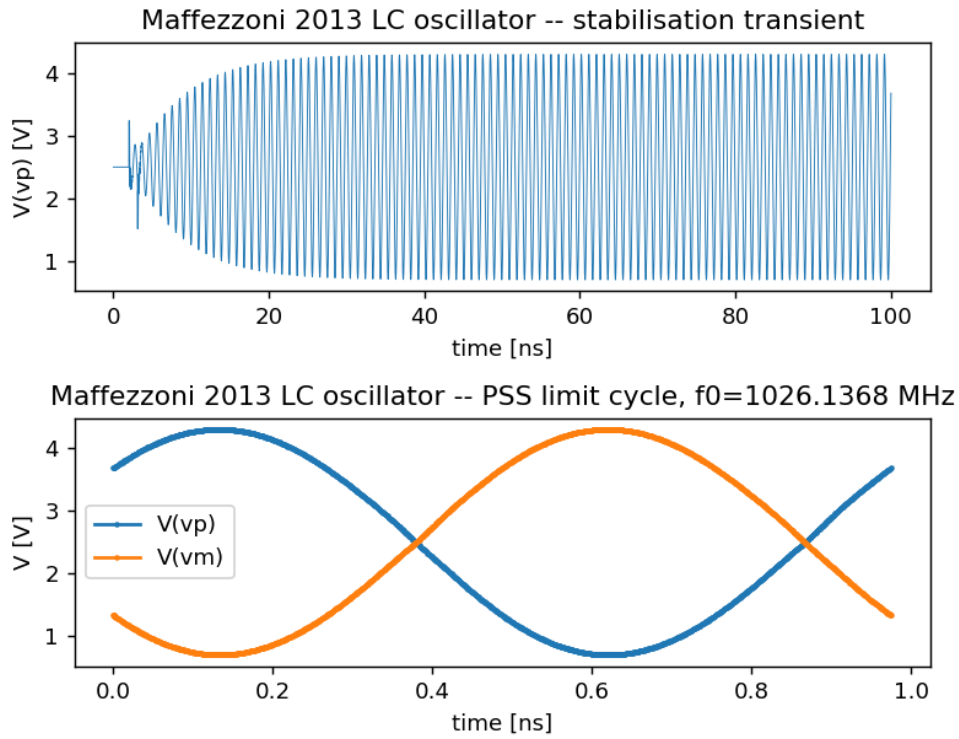
(Quasi)periodic small-signal analysis (HBAC)



- Gilbert cell upmixer operating in multiplier mode
- flo=1MHZ, HBAC spurs
- Switching mode fails because HB is not appropriate for such circuits.
- Will reuse code in PAC/PNOISE

Periodic steady-state analysis (PSS)





- Implemented by Simȯn Jaramillo during his internship at INRIA
- Supervisor: Filip Maksimović

Outlook

OpenVAF-Reloaded

- absdelay()
- Vector ports
- Vector parameters
- Submodule instantiation
- Review and include all Meisam Bahadori's extensions

VACASK analyses

- PAC
- PNOISE (driven and autonomous)
- Improve HB (use FFTW)

VACASK features

- Parallel evaluation
- Parallel matrix construction
- Parallel linear algebra
- SuperLU (OpenMP)
- Trilinos Basker (OpenMP)
- Support all open PDKs
- Foreign netlist format support
- Mixed-mode simulation (Icarus Verilog, Verilator, GHDL)
- Iterative solvers (gmres)

Thank you.

This work has received funding from the Chips Joint Undertaking (Chips JU) and its members through the ODE4EC-AMS project (Open Design Environment for European Chips – Analog and Mixed Signal), Grant Agreement No 101252962. Funded by the European Union. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or the Chips Joint Undertaking. Neither the European Union nor the granting authority can be held responsible for them.