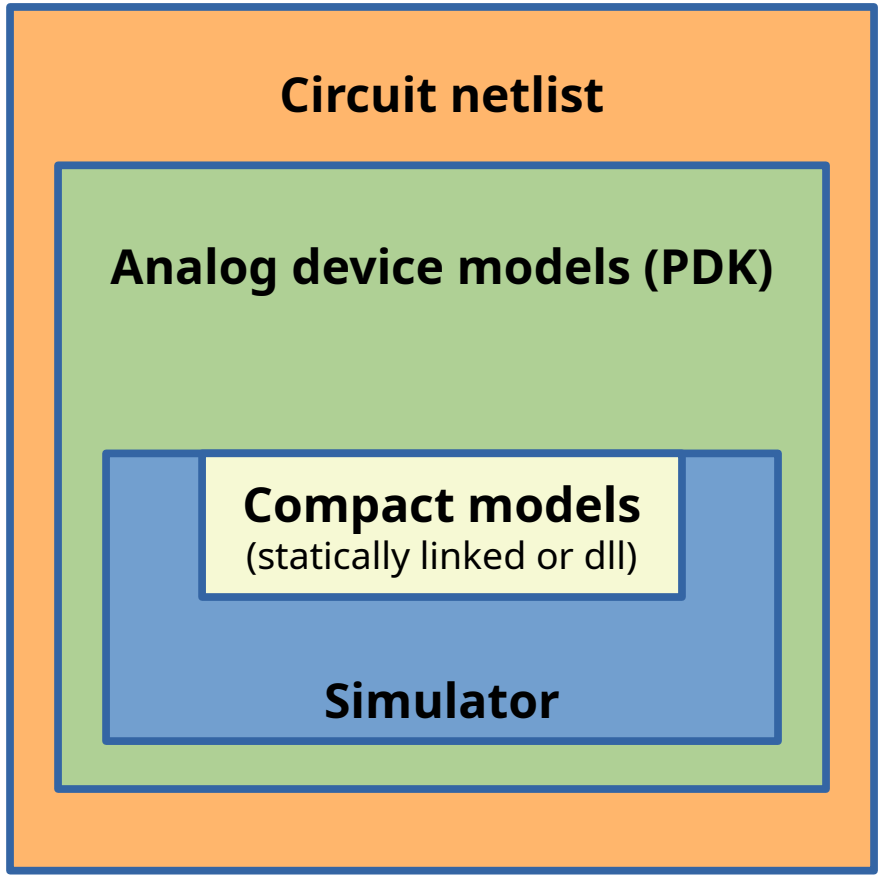
The OpenVAF Verilog-A Compiler for the OpenPDK Ecosystem

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

IHP OpenPDK

- https://github.com/IHP-GmbH/IHP-Open-PDK
- SG13G2 process node - 130nm BiCMOS.
- Ngspice devices:
  - VBIC bipolar transistor model (1.15).
- Verilog-A devices:
  - PSP MOSFET model (103.6),
  - r3_cmc resistor model,
  - MOS varactor.
- PDK designed to be used with the Ngspice simulator.
- OpenVAF provides Verilog-A support in Ngspice.

The big picture




```
.subckt inv (in out vdd vss)
M1 (out in vss vss) nm w=10u l=1u
M2 (out in vdd vdd) pm w=15u l=1u
.ends
```

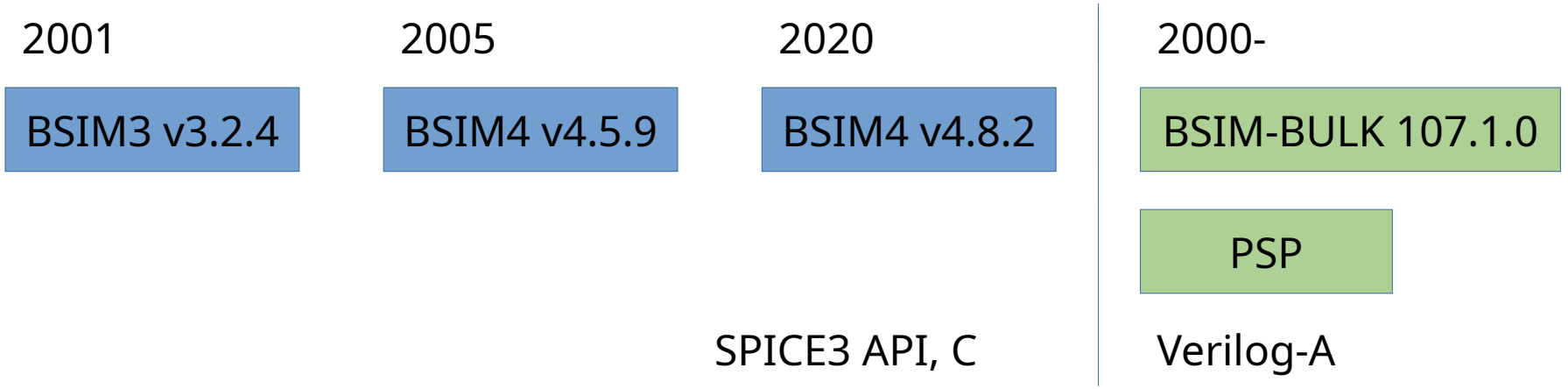
```
.model nm nmos kp=1m vto=1.5 lambda=1m
.model pm pmos kp=0.6m vto=1.4 lambda=1.2m
```

$$I_D = \frac{K}{2}\frac{W}{L}\left(V_{GS} - V_T\right)^2\left(1 + \lambda V_{DS}\right)$$

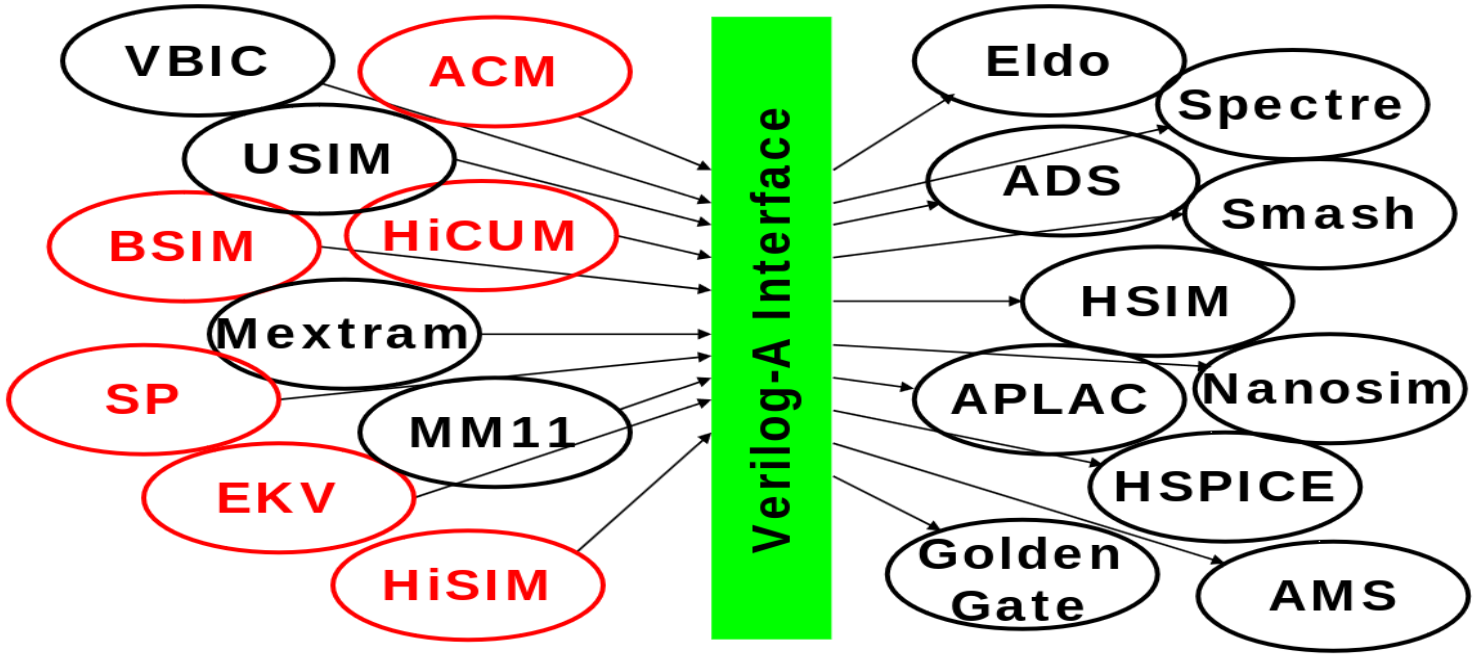
Device equations are coded in Verilog-A.

Compact models - history

- Yesterday – implemented in C using SPICE3 API.
- Today – implemented in Verilog-A.
- Need a compiler.
  Verilog-A → dynamic library → loaded by simulator
- ADMS (old) or OpenVAF (new).
- ADMS (Xyce, Ngspice), OpenVAF (Ngspice, VACASK).



Main advantages of Verilog-A



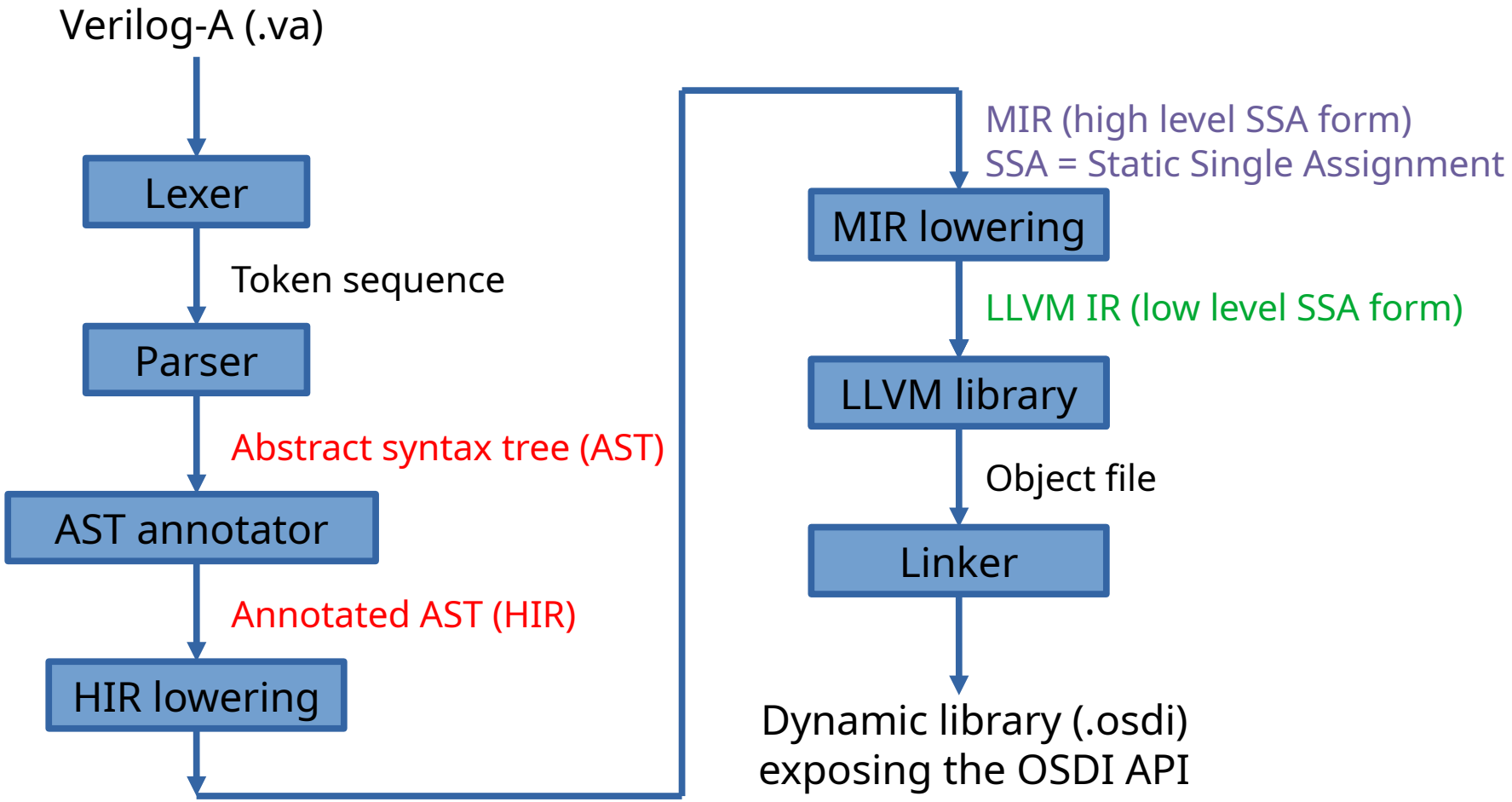
Source: C. McAndrew, *et al*, "Extensions to Verilog-A to Support Compact Device Modeling"

- Write Verilog-A once, deploy to multiple simulators.
- Implement analysis once for all Verilog-A models.

Introduction to OpenVAF-reloaded

- Verilog-A compiler for compact models.
- Author: Pascal Kuthe, now works for Keysight.
- https://github.com/arpadbuermen/OpenVAF
  bleeding edge
- https://github.com/OpenVAF/OpenVAF-Reloaded
  community, lags behind
- https://github.com/pascalkuthe/OpenVAF
  dead, still active discussions
- Compiles all CMC models without manual intervention.
- Fast compilation, fast models.
- Written in Rust, uses LLVM library to create an object file.
- Extended OSDI API for using/accessing the compiled model (0.4).

OpenVAF internals



OpenVAF-reloaded – what is new

- Incorrect $analysis() reporting
- $discontinuity(-1)
- Incorrect handling of voltage contrib when $mfactor!=1
- A bunch of preprocessor bugs
- Empty module not working
- ceil() support
- $limit with one argument
- Singular matrix for single node voltage contrib
- String parameter ranges
- PSP model derivatives issue
- Preprocessed files now compile
- $mfactor handling for voltage noise contributions
- Singular Jacobian for voltage noise contribution nodes
- Noise scaling with $mfactor
- Crash in LLVM under Windows
- List of model inputs
- Jacobian entries output
- Support for HB analysis
- $fatal, $finish, $stop support
- Dumping MIR, LLVM IR, DAE

OpenVAF-reloaded roadmap

- [ ] Improve HB interface.
- [ ] Interface for transient noise.
- [ ] Attribute access (module, parameter).
- [ ] HIR reverse engineering.
- [ ] Timestep control ($bound_step).
- [ ] Discontinuity signaling ($discontinuity).
- [ ] Improve $simparam.
- [ ] Speed improvements (inlining?).
- [ ] Speed improvements (better IR?).
- [ ] Document code as you go.

Ngspice

- SPICE3-based circuit simulator.
- Old, but definitely not obsolete.
- OpenVAF-reloaded provides Verilog-A support.
- Verilog-A Distiller project uncovered several bugs in models,
- fixes merged upstream.
- Supports OSDI API 0.3 (old) and 0.4 (OpenVAF-reloaded).
- Support for models generated by Verilog-A Distiller.
- Good performance, especially wrt. Xyce.
- Benchmark on a 16x16 multiplier (C6288) implemented with PSP MOS.

Simulator	Time [s]	Points	Rejected	Evals
Xyce	151.57	1013	37	3559
Ngspice	71.81	1020	1	3474

VACASK

- Verilog-A Circuit Analysis Kernel.
- A novel analog circuit simulator.
- Modern C++.
- Short (44k lines).
- Easy to extend.
- Device models – Verilog-A.
- Leverages OpenVAF.
- KLU linear solver.

https://codeberg.org/arpadbuermen/VACASK

```
CMOS ring oscillator

load “psp103v4.osdi”
include “cmos_models.inc”
model isrc isource
model vsrc vsource

subckt inv(in out vdd vss)
parameters w=10u l=0.2u fac=2
  mp (out in vdd vdd) pmos w=w*fac l=l
  mn (out in vss vss) nmos w=w l=l
ends

subckt ring()
  x1 (1 2 vdd 0) inv
  x2 (2 3 vdd 0) inv
  x3 (3 1 vdd 0) inv
ends

vdd (vdd 0) vsrc dc=1.2

subckt start()
  ipulse (0 1) isrc type=”pulse” \
  v0=0 v1=1u delay=1n rise=1n fall=1n width=1n
ends

control
  elaborate circuit(“ring”, “start”)
  analysis tran1 tran step=0.05n stop=1u maxstep=0.05n
endc
```

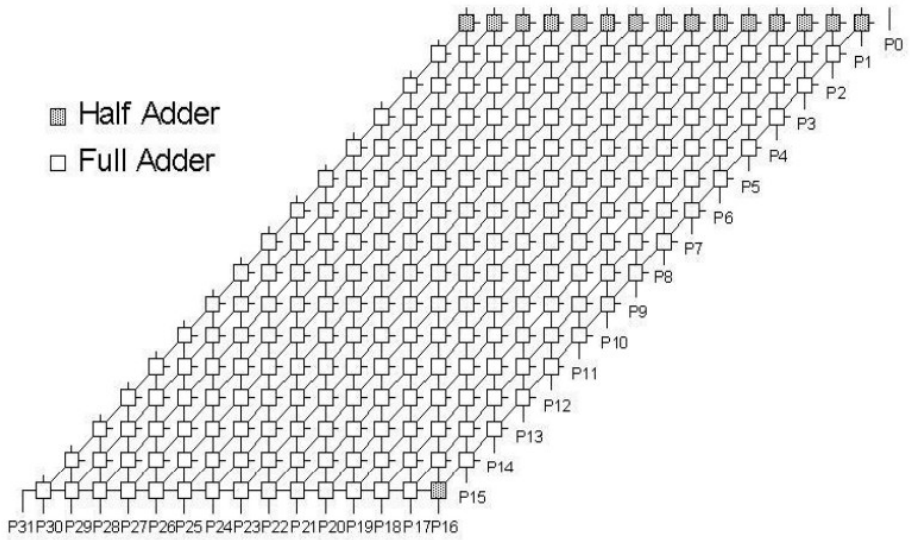
What makes VACASK different

- Model/analysis separation.
- SPICE analyses + HB.
- Parameterized, hierarchical.
- Sweep anything.
- Alter parameters wo. restart.
- Change topology wo. restart.
- Precise (residual checks).
- Fast (designed for modern CPUs).
- RAW file output.
- Embedded files in netlist.

```
// ring oscillator definition here ...

control
  elaborate circuit(“ring”, “start”) // with startup pulse
  analysis tran1 tran step=0.05n stop=1u maxstep=0.05n

  elaborate circuit(“ring”) // no startup pulse
  analysis tran2 tran step=0.05n stop=1u maxstep=0.05n

  postprocess(PYTHON , “runme,py”)
endc

embed “runme.py” <<<FILE
  from rawfile import rawread
  import numpy as np
  import matplotlib.pyplot as plt

  fig1, ax1 = plt.subplots(2, 1)

  tran1 = rawread('tran1.raw').get()
  t = tran1["time"]; v1 = tran1["1"]
  fig1.axes[0].set_title('Oscillator startup, without a pulse')
  fig1.axes[0].plot(t*1e6, v1, color="blue", marker=".", label="v(1)")

  tran2 = rawread('tran1.raw').get()
  t = tran2["time"]; v1 = tran2["1"]
  fig1.axes[0].set_title('Oscillator startup, with a pulse')
  fig1.axes[1].plot(t*1e6, v1, color="blue", marker=".", label="v(1)")

  plt.show()
>>> FILE
```

VACASK performance

C6288: 16x16 digital multiplier
Compute 0xFFFF x 0xFFFF
PSP103.4 MOSFETs
10112 transistors, 25380 unknowns



https://codeberg.org/arpadbuermen/VACASK/src/...
...branch/main/benchmark/c6288/vacask/runme.sim

Default VACASK settings,
higher precision than SPICE

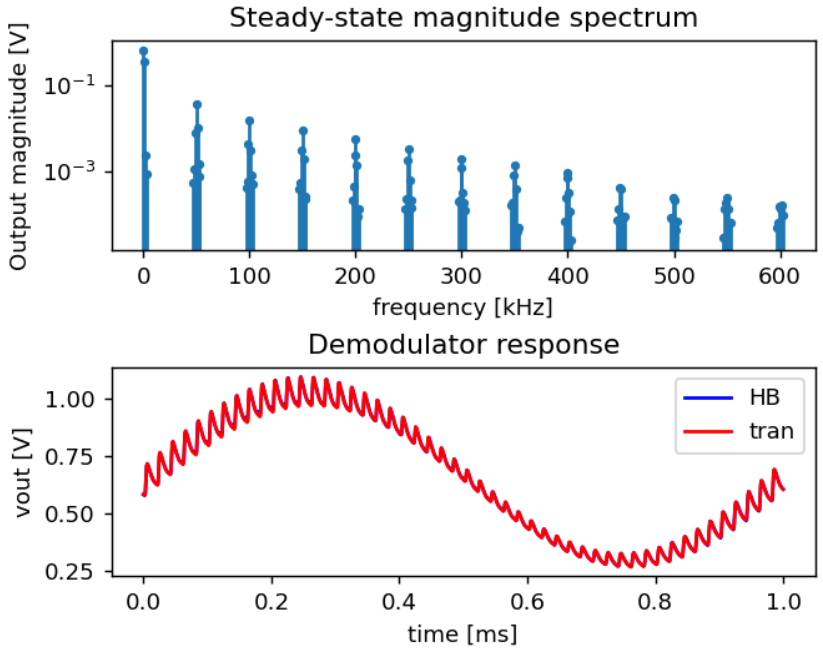
Simulator	Time [s]	Points	Rejected	Evals
Xyce	151.57	1013	37	3559
Ngspice	71.81	1020	1	3474
VACASK	57.98	1021	7	3487

Disabled residual tolerance check,
precision roughly equal to that of SPICE

Simulator	Time [s]	Points	Rejected	Evals
Ngspice	71.81	1020	1	3474
VACASK, no rtc	48.34	1024	8	3090

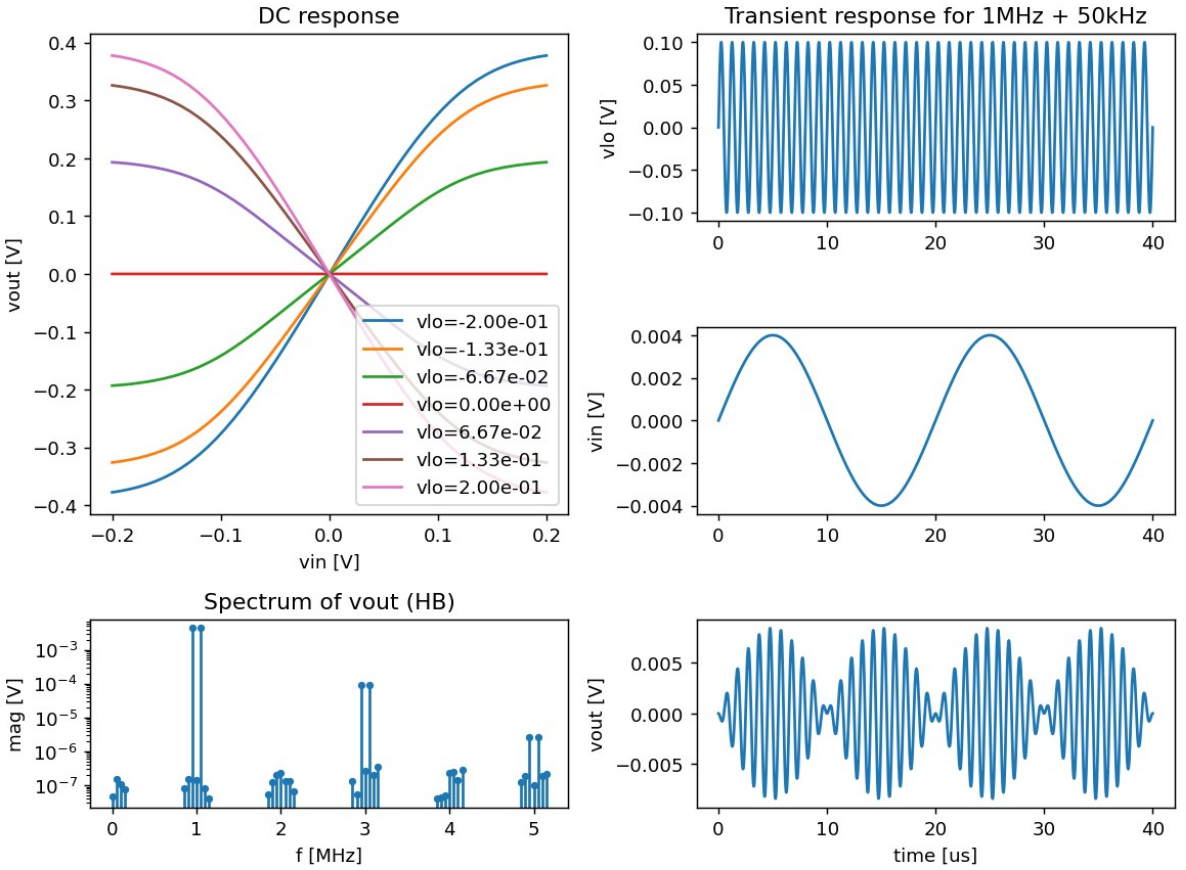
VACASK – harmonic balance

Diode AM demodulator, diode + R + C
2 tone HB, 3 sidebands, incommensurate
50kHz carrier, 1.01kHz signal

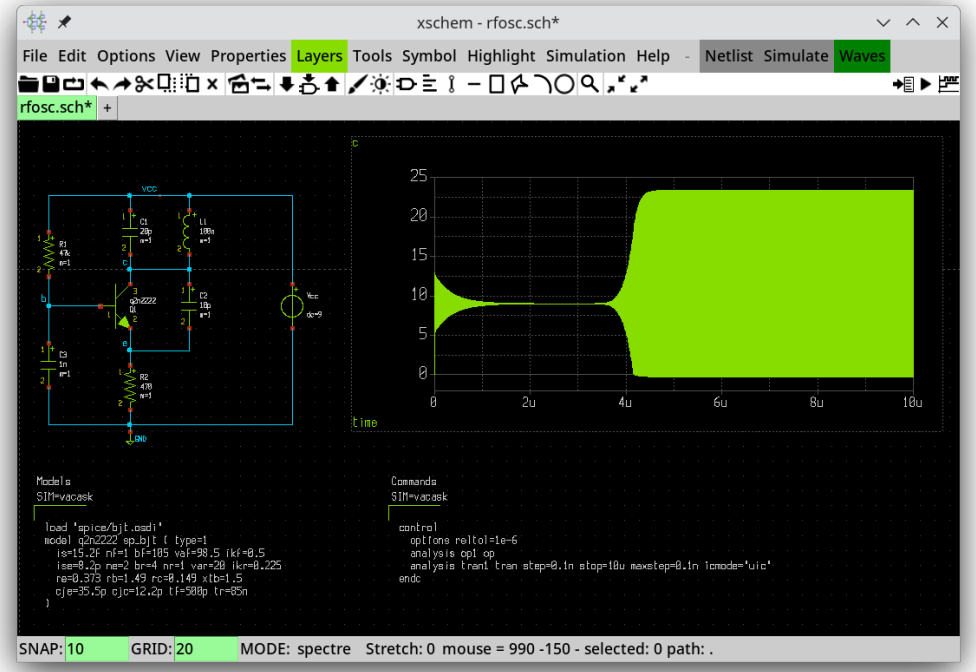


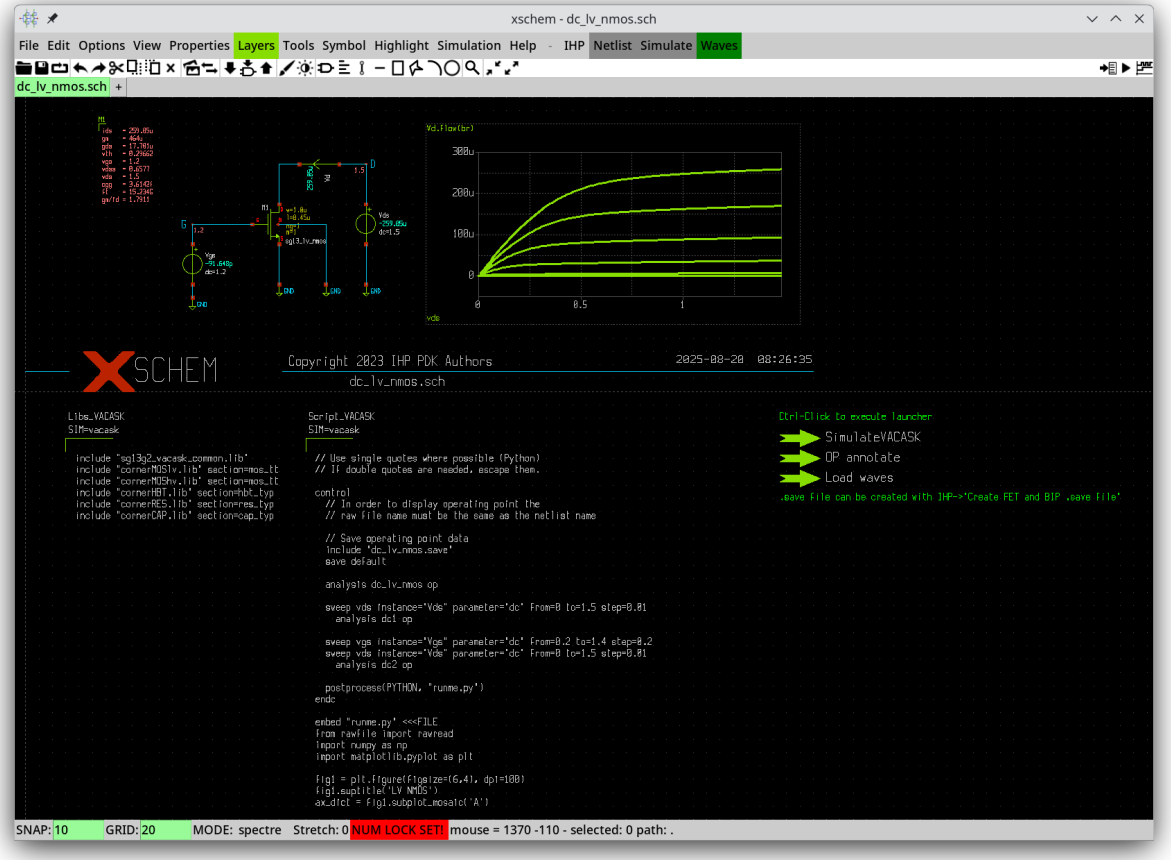
https://codeberg.org/arpadbuermen/VACASK/...
...src/branch/main/test/test_hb5.sim
https://codeberg.org/arpadbuermen/VACASK/...
...src/branch/main/demo/gilbert/gilbert.sim

Gilbert cell mixer, 6 PSP103.4, 3 resistors
2 tone HB, 3 sidebands, commensurate



VACASK – Xschem and IHP PDK integration





- Big thanks to
  - Stefan Schippers for implementing the VACASK interface in Xschem
  - Harald Pretl for sparking the idea of a Ngspice netlist converter

VACASK roadmap

- Convert Skywater PDK.
- Faster NOISE and XF analyses (adjunct circuit equations).
- Stability analysis (STAB) and S-parameter analysis (SP).
- Transient noise analysis.
- Parallel evaluation (OpenMP).
- Parallel linear solver (SUPERLU, MUMPS).
- Shooting (PSS) & small-signal family (PAC, PXF, PNOISE, PSP, PSTAB).
- Improve harmonic balance, use FFTW library.
- HB periodic small-signal family (HBAC, HBXF, HBNOISE, HBSP, HBSTAB).
- Convolutional linear models (touchstone).

Legacy models

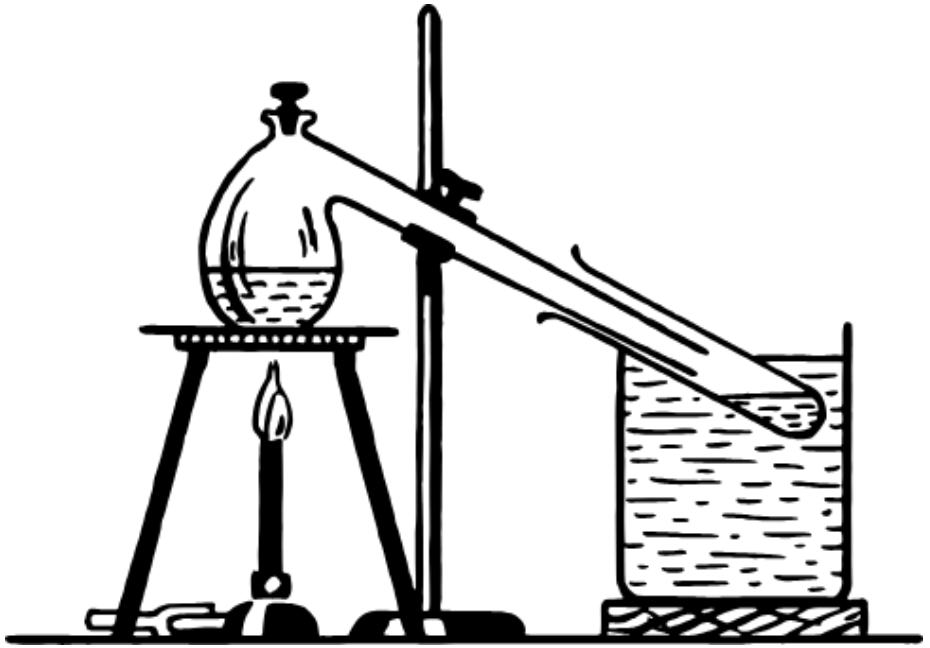
- SPICE3 models written in C, most of them not available in Verilog-A.
- Still used!
  - TSMC18 uses Gummel-Poon BJT, BSIM3v3.
  - Many PDKs use SPICE diode.
  - Discrete devices use MOS3, Gummel-Poon, JFET1, SPICE diode, …
- Educational value
- Options
  - Rewrite SPICE3 models in C/C++ for target simulator – a lot of work.
  - Wrap SPICE3 models – cannot use them in advanced analyses (HB).
  - Convert to Verilog-A – solve the problem once and for all.
- Why convert?
  - Implement once for all simulators, avoid bugs (in derivatives, …).
  - Simplify implementation of new analyses.
- Not available in Verilog-A: BSIM3, BSIM4, BSIM3SOI, BSIMSOI.

Verilog-A Distiller

- Funded by NLNET NGI ZERO.
- Semi-automatic converter.
- Must clean up code before conversion (mostly on AST level).
- Written in Python3, uses pycparser library.
- Converts Ngspice pre-master-45 models.
- Can track developments in Ngspice models.
- Generated Verilog-A models are validated.
- Models work in Ngspice and VACASK simulators.
- Should also work in others.
- 12k lines of Python code, 10k lines of config.
- 51k lines of C → 18k lines of Verilog-A.
- https://codeberg.org/arpadbuermen/VADistiller

Verilog-A Distiller status

- Converter almost finished.
- Converted and validated:
  - Passives: R, C, L
  - SPICE diode (L1, L3)
  - Gummel-Poon BJT
  - JFET (L1, L2)
  - MESFET L1
  - MOSFET (L1-3, L6, L9)
  - BSIM3
- Working on BSIM4.



Converted models – code size

Element	C lines	Verilog-A lines	Reduction (x)
Resistor	1201	329	3.7
Capacitor	974	306	3.2
Inductor	1068	285	3.7
Diode	2891	1002	2.9
BJT	4607	1458	3.2
JFET 1	2004	728	2.8
JFET 2	1864	983	1.9
MESFET 1	1670	617	2.7
MOS 1	4217	1275	3.3
MOS 3	4732	1613	2.9
BSIM3.3.0	12478	4211	3.0

Converted models – data structure size

Element	SPICE3		Verilog-A + OpenVAF		Change wrt. SPICE3 [%]	
	Model	Instance	Model	Instance	Model	Instance
Resistor	176	320	232	280	32	-13
Capacitor	160	208	200	216	25	4
Inductor	128	240	144	216	13	-10
Diode	568	800	592	680	4	-15
BJT	1128	1712	1048	2096	-7	22
JFET 1	280	720	224	904	-20	26
JFET 2	408	728	360	1480	-12	103
MESFET 1	224	680	136	840	-39	24
MOS 1	320	1240	360	1824	13	47
MOS 3	424	1240	440	1984	4	60
BSIM3.3.0	3600	1304+968	3512	2472	-2	9
PSP103.4	-	-	6320	6904	-	-

Converted SPICE3 models - speed (1/2)

- Evaluation bypass disabled, no OpenMP, use KLU linear solver.
- Verilog-A cases - if possible, use Verilog-A models.
  - Multiplier (4x diode, 4x capacitor, 1x resistor, 1x vsrc)
  - Ring oscillator (5x JFET1, 10x capacitor, 15x resistor, 1x vsrc)
  - Ring oscillator (9x BJT, 9x capacitor, 18x resistor, 1x vsrc, 1x isrc)
  - C6288 (10112x converted BSIM3 - no NQS, 32x resistor, 33x vsrc)
  - C6288 (10112x Cogenda BSIM3 - no NQS, 32x resistor, 33x vsrc)
- Measure
  - total runtime (per iteration),
  - eval&load time (per iteration).
- Use Ngspice & VACASK simulators.
- Compare converted BSIM3 with Cogenda BSIM3 Verilog-A model.

Converted SPICE3 models - speed (2/2)

	NR iteration time [us]			Eval&load time [us]		
	Ngspice		VACASK	Ngspice		VACASK
Circuit	**Native**	**Verilog-A**	**Verilog-A**	**Native**	**Verilog-A**	**Verilog-A**
Diode multiplier	1.160	1.611	1.103	0.291	0.619	0.505
JFET1 ring oscillator	1.750	2.586	2.199	0.413	1.072	0.906
BJT ring oscillator	3.328	6.016	4.749	1.117	2.749	2.593
C6288 BSIM3	6580	12180	11950	4180	9650	10270
C6288 BSIM3 Cogenda	x	8690	8730	x	7300	8020

- VACASK Verilog-A interface is faster than that of Ngspice.
- Converted models are roughly 2x slower than native models.
- There is room for improvement in OpenVAF and VA Distiller.

Thank you!