FOSDEM'25

Árpád Bűrmen
University of Ljubljana
Faculty of Electrical Engineering

VACASK and Verilog-A Distiller

building a device library for an analog circuit simulator

nlnet FOUNDATION

NGI ZERO

UNIVERZA V LJUBLJANI

VACASK

- Verilog-A Circuit Analysis Kernel
- Analog circuit simulator
- Most of its device models defined in Verilog-A
- Uses OpenVAF Verilog-A compiler
- Separates models and anayses
- C++, clean code, fast
- **https://codeberg.org/arpadbuermen/VACASK**
- https://wiki.f-si.org/index.php?title=VACASK:_a_Verilog-A_Circuit_Analysis_Kernel

VACASK roadmap

- Improve harmonic balance analysis
- Add RF analyses, e.g.
  S-parameters, stability, transient noise,
  shooting methods, cyclostationary noise, ...
- Automatic binning
- Parallel evaluation, parallel linear solver
- Improve DC convergence (i.e. $limit())
- Mixed-mode simulation
- **Support for legacy SPICE3 devices**

VACASK comparison

	Xyce	Ngspice	Gnucap	VACASK
Core lines, language	185500, C++	63800, C	28600, C++	43904, C++
Verilog-A	ADMS, in-house*	OpenVAF, ADMS	ADMS, Modelgen*	OpenVAF
OP	yes	yes	yes	yes
AC	yes	yes	yes	yes
TRAN	yes	yes	yes	yes
TF	-	DC	-	DC/AC
Small-signal NOISE	yes	yes	-	yes
HB	yes	-	-	yes
Sweep analyses, depth	all, arbitrary	OP, 2	OP, 1	all,arbitrary
Parallel	yes (MPI, OpenMP)	partially (OpenMP, CUDA)	-	-
SPICE devices	yes	yes	yes	**TODO**

What is a compact device model?

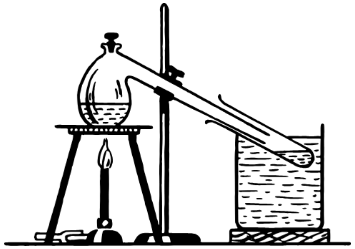
- Equations describing the behavior of a device (e.g. diode, BJT, MOSFET, ...)
- Past de-facto standard: SPICE3 C API
- Must re-implement each compact model in each new non-SPICE3 simulator (1..30k lines of code per model!)
- Current de-facto standard: Verilog-A
- Need a compiler for Verilog-A – e.g. OpenVAF **https://github.com/arpadbuermen/OpenVAF**
- Simulator must support compiled models

Verilog-A

- Analog hardware description language
- Compact models - Verilog-A subset
- Short model code (5 lines/parameter)
- Compiler generates derivatives, avoids error-prone manual coding
- Focus on equations, not implementation
- Write once, compile for each simulator
- No need to re-implement models

Legacy device models

- e.g. BSIM3, BSIM4, Gummel-Poon BJT, ...
- From 1970s (SPICE2) to 2020 (e.g. BSIM4.8.2)
- IC manufacturing processes characterized in terms of legacy device models are **still used** (e.g. TSMC18, Skywater, ...).
- Discrete devices – e.g. 2N2222 transistor is described with its Gummel-Poon parameters
- Educational value, engineering heritage
- Implemented in C using SPICE3 API
- Still maintained in Ngspice



Automatic conversion

- **Verilog-A Distiller, since October 2024**

  https://codeberg.org/arpadbuermen/VADistiller
- Reasonably fast is good enough – Python3
- Uses pycparser for parsing C code
- Library of AST manipulation functions
- Converts Ngspice pre-master-45 models
- Track developments in Ngspice models
- Generated Verilog-A models are validated
- Converted models work in Ngspice and VACASK.
  Should work in other simulators with Verilog-A support.

OK, stop!
Why not write a wrapper?

- Gnucap does that...
- Circuit’s system of equations in time domain

$$g(x)+\frac{d}{dt}q(x)=0$$

- Need the resistive/reactive part and the Jacobians

$$g(x) \quad q(x) \qquad\qquad G(x) \quad C(x)$$

- Spice computes

$$g(x)+f(q(x),q_{past}) \qquad G(x)+aC(x)$$

- Works for DC, AC, TRAN, NOISE; but not for HB & co.

Anatomy of a SPICE3 device model (e.g. diode)

Instance/model data structure
diodefs.h

Instance/model setup
diosetup.c

Instance/model parameters
dio.c

Temperature dependence
diotemp.c

Instance parameter access
dioparam.c
dioask.c

Instance resistive/reactive component evaluation
dioload.c

Model parameter access
diompar.c
diomask.c

Instance noise evaluation
dionoise.c

- Can throw away the rest...

Models (under construction) and slideware...

- Conversion engine mostly in place, needs tuning
- Resistor, capacitor, inductor
  Include many geometric and secondary effects. Validated.
- Diode
  Levels 1 and 3. Validated.
- JFET
  Converted, have a look. Working on validation.
- BJT (Gummel-Poon)
  Working on it.
- Coming soon to a Verilog-A capable simulator near you: MOS 1, 2, 3, 6; JFET 2; MESFET; BSIM3; BSIM4

Layout by orngjce223, CC-BY

Data structure size in bytes

Element		SPICE3 C	OpenVAF	Relative size [%]
Resistor	model	176	232	+32
	instance	320	280	-12.5
Capacitor	model	160	200	+25
	instance	208	216	+3.8
Inductor	model	128	144	+12.5
	instance	240	216	-10
Diode	model	568	592	+4.2
	instance	800	648	-19
JFET	model	280	224	-20
	instance	720	864	+20

Generated code size

Element	# parameters	C lines	Verilog-A lines	Reduction (x)
Resistor	29	1200	252	4.8
Capacitor	23	974	225	4.3
Inductor	18	1068	220	4.9
Diode	78	2887	935	3.1
JFET	39	2003	649	3.1

Element	# parameters	C lines/par	Verilog-A lines/par
Resistor	29	41.4	8.7
Capacitor	23	42.3	9.8
Inductor	18	59.3	12.2
Diode	78	37.0	12.0
JFET	39	51.4	16.6

Cogenda BSIM3: 14
Manual BSIM-BULK 107: 4.7

Early results - speed (diode)

- Voltage multiplier: 4 diodes, 4 capacitors, 1 resistor, 1 vsrc
- Disable OpenMP and element bypass for fair comparison
- Use OpenVAF to compile the models
- 100kHz sine, simulate 500 periods

Total runtime (circuit)		
Element	**Time [s]**	**Evaluations**
Native Ngspice	4.414	1 111 349
Verilog-A + Ngspice	5.726	1 205 679
Verilog-A + VACASK	3.982	1 009 365

Eval&load (circuit)			Eval&load (diodes)	
Element	**Time [s]**	**Per eval [us]**	**Time [s]**	**Per eval [us]**
Native Ngspice	1.482	1.33	0.704	0.644
Verilog-A + Ngspice	2.507	2.08	1.557	1.60
Verilog-A + VACASK	1.652	1.64	1.316	1.30

Alternatives, limitations, and roadmap

- JFET level 2, MESFET, and MOS levels 1, 2, 3, 6:
  capacitance models do not conserve charge.
  Cannot be implemented in Verilog-A!
  Use a replacement model, e.g. Sakallah et. al.
  Can only approximate SPICE3 model behavior.
- Alternative - native C/C++ model output: Xyce, VACASK, Gnucap, ...
  Can handle models with charge non-conservation.
- $limit() improvements needed in OpenVAF → simpler models
- $limit() not capable of doing everything SPICE3 models do, need hacks
- Handle BSIM3 data structures, BSIM4 noise model
- Maybe: BSIM3SOI models, various versions of BSIM3 and BSIM4
- Maybe: output VHDL-AMS models
  ... need to write a new backend... anybody out there?

Thank you.

nlnet FOUNDATION NGI ZERO