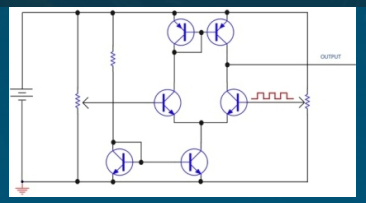
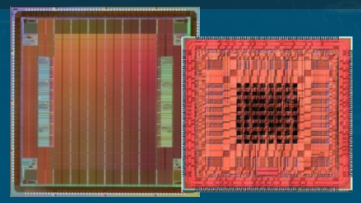
SAND2021-15941C

Sandia National Laboratories

Current Developments for the Xyce Circuit Simulator





PRESENTED BY

Jason C. Verley

U.S. DEPARTMENT OF ENERGY NNSA

Sandia National Laboratories is a multimission laboratory managed and operated by National Technology and Engineering Solutions of Sandia LLC, a wholly owned subsidiary of Honeywell International Inc. for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-NA0003525.

SAND2021-XXXX

Why Develop an Analog Circuit Simulator at Sandia?

Why Develop an Analog Circuit Simulator at Sandia?

Commercial tools cannot simulate the high dose rate radiation environments that are Sandia's primary concern

A known, reliable code base, owned by Sandia

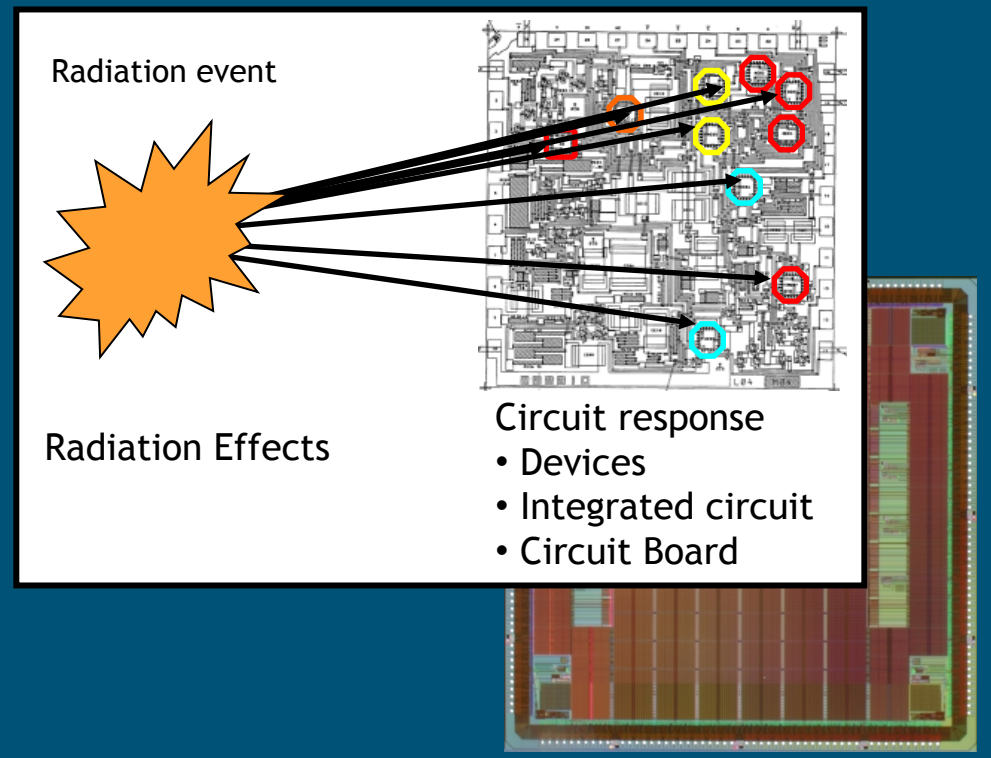
- Prompt response to internal needs
- Trusted code

Specialized compact models for radiation effects

- High-energy photons (X-rays and γ-rays)
- Neutrons

Simulation of entire systems

- Radiation effects are not isolated…
- Need massively parallel simulations



The Xyce Analog Circuit Simulator

Two versions, **Serial** and…

Distributed Memory Parallel (MPI-based)

Unique solver algorithms

Industry standard models

Non-traditional models
- Memristor
- Neuron/synapse
- TCAD (PDE-based)

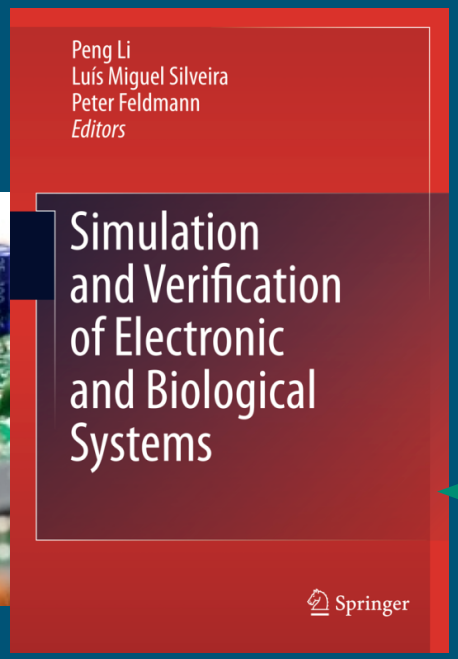
SPICE-Compatible syntax (Berkeley 3f5)

`http://xyce.sandia.gov`

`http://github.com/Xyce`

Open Source, GPLv3
- Since September of 2013 (Xyce 6.0)

Xyce Release 7.4
- October, 2021
- >7,600 external downloads since 6.0



Keiter, et al., "Parallel Transistor -Level Circuit Simulation"

Xyce Capabilities

Typical

DC, Transient, AC, Noise
- .DC, .TRAN, .NOISE, .AC (and .STEP)

Post Processing:
- Fourier transform of transient output (.FOUR)
- Post-simulation calculation of simulation metrics (.MEASURE)

Output (.PRINT)
- Text Files (tab or comma delimited)
- Probe (PSPICE)
- Gnuplot, TecPlot, RAW (SPICE 3f5)

Analog Behavioral Modeling

Expressions, functions, parameterizations…

Others

S-Parameter Analysis
- Touchstone file output

Harmonic Balance Analysis (.HB)
- Steady state solution of nonlinear circuits in the frequency domain

Random Sampling Analysis
- Executes the primary analysis (.DC, .AC, .TRAN, etc.) inside a loop over randomly distributed parameters

Polynomial Chaos Expansion (PCE) methods
- Functional dependence of a simulation response on uncertain model parameters

Sensitivity Analysis
- Computes sensitivities for a user-specified objective function with respect to a user-specified list of circuit parameters ($\partial O/\partial p\ldots$)

Xyce-isms

Xyce is not a “plug in replacement” for any other simulator

- …but that’s like trying to hit an invisible moving target
- XDM netlist translator for syntax
- Feature compatibility is significant and improving
  - Still some expected features that we don’t yet support; e.g., .OP functionality is limited

Xyce defaults are conservative

- The industry standard is, “The simulator must never fail to provide an answer…”

  …even if it’s wrong.
- Xyce philosophy: provide a numerically accurate answer, and fail if asked to do something “wrong”
  - Continue to work to provide options that allow problematic circuits to simulate; but usually not on by default

Simulations in Serial vs. MPI Parallel

- Distributed parallelism can take more tuning than shared memory approaches…

  especially device distribution, and direct vs. iterative linear solvers.
- Very large parallel simulations are challenging (need to find the “right” linear solver)
- Leverages Sandia’s Trilinos High-Performance Computing (HPC) solver framework

Xyce Distributions

Binary installers (serial and parallel)

- RHEL 7
- Apple Macintosh
- Windows (serial only)
- Include proprietary compacts models and (eventually) other proprietary features, such as linear solvers
- `http://xyce.sandia.gov`

Source code

- `http://xyce.sandia.gov`
- `http://github.com/Xyce`

Building Xyce is not for the faint of heart

- …usually due to the third-party libraries
- Detailed build instructions on the website
- CMake improvements on the way

Xyce “in the Wild”

Note: Xyce is the simulator (like HSPICE, SmartSpice, Spectre, Eldo…)

- Sandia does not provide a Schematic Design/Capture front end, but…
  there are Qucs-S and gEDA integrations, and…

Typhoon HIL Schematic Editor

- Free (not open-source)
- Includes Xyce integration
  https://www.typhoon-hil.com/products/xyce-integration/
- Primary business is power system emulation, so the editor has a power/distribution emphasis

DARPA POSH/IDEA

- Develop an IC “Compiler”
- Open Source Hardware (POSH)
- Andreas Olofsson has a list of the projects at https://github.com/aolofsson

Simulator Compatibility and PDKs

Or, Recent Xyce Improvements

Xyce Support for PDKs

Most PDKs are written for commercial tools, using their syntax.

So, tool compatibility translates to PDK compatibility

- Netlist syntax
- Features

XDM – A Netlist Translator for Xyce

- Support for
  - HSPICE
  - Spectre
  - PSPICE
- 90% solution
  - Primarily limited by Xyce feature compatibility with the commercial tools

Xyce Support for PDKs

Some success with
- Global Foundries 65/55 nm
- Global Foundries 14/12 nm

Some known deficiencies:
- Multipliers ("M=") with subcircuits
- Verilog-A support
- TMI/OMI (required for TSMC PDK support)
- …others

Translation Procedure
- Pre-process the PDK using XDM
- Fix any remaining issues by hand (most, if not all, are known)
- Translate the netlist for the specific simulation, using XDM
  - Again, fix any remaining issues by hand
  - Some simulation-specific directives are not translatable

Primary Challenge: Foundries work with the EDA vendors to ensure the PDKs are "correct."
Without that relationship, we can only address issues as they come up.

SkyWater Open Source PDK

https://github.com/google/skywater-pdk

A collaboration between Google and SkyWater Technology Foundry
- Targets the SKY130 process node

As-released PDK targeted ngspice
- Many compatibility issues have been addressed
- Some reports that Xyce is faster than ngspice (related to parsing?)

Known remaining compatibility issues
- Multipliers ("M=") with subcircuits
- Device model enhancements (BSIM3, e.g.)
- Some HSPICE-like syntax incompatibilities
- Other smaller issues…

Unlike closed PDKs, we are able to work with the community to improve compatibility and performance.

Xyce Interfaces

Various Xyce Interfaces

Application Programming Interfaces (APIs)

- Mixed Signal
- Xyce General External Interface (C++)
- App Notes on the website

PySPICE on GitHub

- Xyce and ngspice
- https://github.com/FabriceSalvaire/PySpice

And…

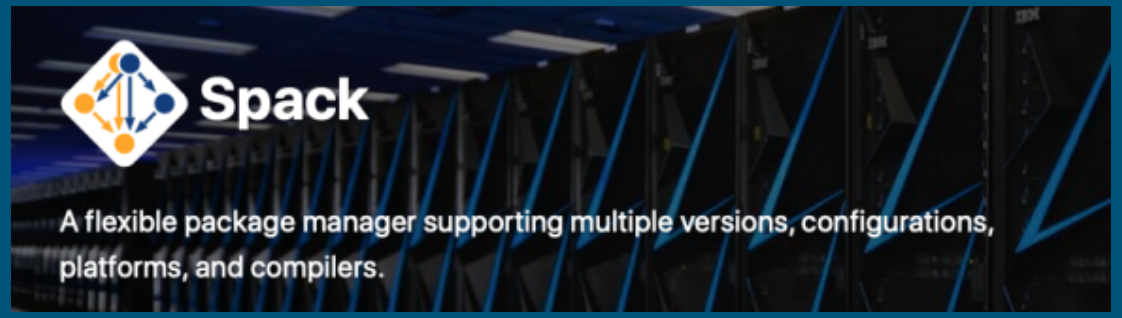
Xyce-PyMi

Application Note:

- “An embedded Python model interpreter for Xyce (Xyce-PyMi)”
- Provides compilation and use instructions (see the Xyce website)

Available via Spack, “a package manager for supercomputers, Linux, and macOS”

- https://spack.io
- (also another way to install Xyce, though not officially supported)



Current focus: Surrogate Modeling and Model Order Reduction (MOR) approaches

No publications, yet, but see, e.g.:

- K. Aadithya, P. Kuberry, et al., **Data-driven compact models for circuit design and analysis**, *Proceedings of Machine Learning Research*, 107, 555-572, 2020.

Verilog-A and Xyce

A Selection of Recent Tools

ADMS
- Still the most capable open-source model compiler available

VAPP (Verilog-A Parser and Processor)
- Translates Verilog-A device models into ModSpec, a device modeling specification & compiler
- Used with the NEEDS project (https://nanohub.org/groups/needs/)

VAMPyRE (Verilog-A Model Pythonic Rule Enforcer)
- Verilog-A compact model parser and checker
- Supported by the Compact Model Coalition (CMC) (https://si2.org/standard-models/)

VerilogAE
- Verilog-A Compiler for Compact Model Parameter Extraction

Xyce/ADMS

What is Xyce/ADMS…

- XML templates that provide a code-generating “back-end” to ADMS

Can compile many standard models for built-in use

Verilog-A can be compiled as a shared-library plug-in

- Requires a special build of Xyce
- Many models need hand-modifications before they will compile

…there are limitations

- Requires @(initial_model) and @(initial_instance) to indicate blocks that should be executed only once
- Missing data types
- Limited support for expressions (such as ddt)
- Missing analog behaviors
- …see the Xyce/ADMS Users Guide at https://xyce.sandia.gov/

Full support of modern compact models requires modification of the ADMS source code (C)

Xyce Verilog-A Path Forward

Develop an in-house Verilog-A compiler targeted at Xyce
- A known, reliable code base, owned by Sandia
- Specialized capabilities for multi-physics, including radiation effects

Leverage work from DAGADO
- a DAG-based differentiation library, written in Python, developed at Sandia

allows for…

Ease development of compact models
- Simplify the shared library approach taken by Xyce/ADMS

Xyce compatibility with modern PDKs
- Dynamic compilation (eventually)

Will release under GPL3 as part of Xyce

Timeline… to be announced, but work is commencing now.

Xyce Team Acknowledgements

Eric R. Keiter

Thomas V. Russo

Richard L. Schiek

Heidi K. Thornquist

Ting Mei

Jason C. Verley

Karthik V. Aadithya

Josh Schickling

Gary J. Templet, Jr. (XDM)

Garrick Ng (XDM)

…and many others

Contact:

`http://xyce.sandia.gov`
`http://github.com/Xyce`
`xyce@sandia.gov`

Google Group Forum:

`https://groups.google.com/forum/#!forum/xyce-users`