Xyce™ Parallel Electronic Simulator Version 7.7 Release Notes

Sandia National Laboratories

August 3, 2023

The Xyce™ Parallel Electronic Simulator has been written to support the simulation needs of Sandia National Laboratories' electrical designers. Xyce™ is a SPICE-compatible simulator with the ability to solve extremely large circuit problems on large-scale parallel computing platforms, but also includes support for most popular parallel and serial computers.

For up-to-date information not available at the time these notes were produced, please visit the Xyce™ web page at http://xyce.sandia.gov.

Contents

Highlights

- Xyce now handles local variation with the same netlist convention as HSPICE and other simulators. This allows users to more easily simulate things like mismatch using commercial PDKs.
- The parser, expression library and device package have undergone various efficiency improvements with respect to parameter handling. In the case of the parser, this speeds up parsing in many cases. For the device package, it speeds up analysis. In the case of the expression library, it often speeds up both parsing and analysis.

New Features and Enhancements

New Devices and Device Model Improvements

- The Xyce diode now supports sidewall effects supported by other simulators such as ngspice and HSPICE.
- The open source version of Xyce now includes the published version of EKV MOSFET model version 2.6 as the level 260 MOSFET. Previously, this model was only available in our "non free" binaries.

Enhanced Solver Stability, Performance and Features

- The expression library `TABLE` and `SPLINE` operators now support a "downsampling" feature, in which the user can request that a large data file can automatically have the number of points used by the source reduced. For `TABLE` based sources, which are based on a PWL linear approximation, this can dramatically speed up simulations, as the number of required breakpoints is reduced.
- The random operators in the expression library (such as `AGAUSS` and `AUNIF`) now support implicit local and global variation, consistent with other simulators.
- Parameter handling in the parser has been refactored. As a result, the parsing and setup time for large process design kits (PDKs) has been improved.

Interface Improvements

- Subcircuit multipliers (`M=`) previously had to have a fixed value. Now, they can be treated like any other parameter, and can change during a simulation using commands such as `.STEP`.
- Formatting errors in unused subcircuit definitions (`.SUBCKT`) will no longer emit errors, only warnings. This is limited to issues in the (`.SUBCKT`) line of the netlist. If additional parsing issues are found within the subcircuit definition, they may still result in an error being emitted from the simulator.
- Xyce now has a diagnostic option, `.options diagnostic` to aid in debugging of user circuits. This option can be used to output nodes with extreme solution values, voltages, currents and discontinuities during a simulation. Additionally, it can provide information on continuation methods used during the DC Operating point calculation. See the Reference Guide section 2.1.25.2 and chapter 10 the User's Guide for more details.
- The third argument (`num_sigma`) to the random expression operators `AGAUSS` and `GAUSS` is now optional.

- The metrics tracking and reporting feature now has an option to include error messages. When enabled, the tracking data reported by Xyce is encoded as a JSON string. If certain error conditions occur, the JSON string may now include a list of diagnostic messages. These error messages will be encoded as an array under the tag name "ErrorMessages" next to the "audit" field (e.g. "audit":{...},"ErrorMessages":[...]).
- Expressions are now allowed on the following analysis lines: `.TRAN`, `.DC` and `.AC`.
- Data files read in via the expression library operators such as `TABLE` and `TABLEFILE` can now contain comments.
- Data files read in via the expression library operators such as `TABLE` and `TABLEFILE` can now include dashes in the filename.
- When running a frequency domain analysis such as `.HB` or `.AC`, complex-valued expressions on the `.PRINT` line will now automatically output both the real and imaginary parts.
- `.params` can now be complex-valued. When used in a real-valued context, only the real part will be used, but in complex (i.e. frequency domain) contexts, complex arithmetic is performed.

XDM

- Fixed bug where an instance of a SPECTRE device is not translated as an instance or a comment if it appears after a definition of a SPECTRE device.

Defects Fixed in this Release

Table 1: Fixed Defects. The Xyce team has multiple issue trackers, and the table below indicates fixed issues by indentifying both the tracker and the issue number. Further, some issues are reported by open source users on GitHub and these issues may be tracked using multiple issue numbers.

Defect	Description
Xyce Project Backlog/106: Xyce/ADMS generates incorrect derivative code for integer variables	Xyce/ADMS would emit code that would not compile if a Verilog-A module tried to assign a probe-dependent expression into an integer variable.
Xyce Project Backlog/157: Xyce diode does not support sidewall effects	The Xyce diode now supports an instance parameter for the junction perimeter (`PJ`) and a number of model parameters (`JSW`, `NS`, `CJSW`, `PHP`, `MJSW`, `FCS`) supporting the effects. Default values are such that the feature is disabled. This brings the Xyce level 1 and 2 diodes into agreement with other simulators when these parameters are specified.
Xyce Project Backlog/492: Subcircuit multiliplier parameters (`M`) need to be handled as an AST (enabling .STEP to be applied to subcircuit `M`)	This task was related to backlog isssue 494. In the initial implementation of subcircuit multipliers, they were constrained to have fixed values. However, in some cases a user may wish to change multiplier values during a simulation. To enable this feature, the subcircuit multiplier parameters had to be part of the same abstract syntax tree (AST) as other parameters from the netlist. This has been fixed.
Xyce Project Backlog/494: Remove parser error trap for using `.global_param` on the implicit subcircuit multiplier parameter (M)	The subcircuit multiplier parameter, `M`, is treated as a special case in the Xyce parser. Due to its special treatment, the use case of setting this parameter via a mutable parameter (i.e. `.global_param`) did not work correctly. This has been fixed. This issue did not apply to device multipliers, just to subcircuit multipliers.
Xyce Project Backlog/527: Xyce/ADMS aborts on analog functions that have arguments of a type different from their return types	When Xyce/ADMS was refactored to remove use of Sacado, a bit of templated code was not rewritten because it appeared to be good enough for the new implementation. But that version had a restriction that all arguments of an analog function must have the same type as the return value. This code has now been rewritten. Analog functions can now return either reals or integers, and may have arguments of either type irrespective of their return type.
Xyce Project Bugs/36: Xyce/ADMS aborts when an analog function calls $strobe, $bound_step, or other “callfunction”	Due to the way Xyce/ADMS implements $strobe, $bound_step, and other statements that look like function calls and how it implements analog functions themselves, these “callfunctions” cannot be used inside analog functions. Until this release, trying to call $strobe from inside an analog function would cause ADMS to abort with a fatal error about a missing template. Now it will emit a warning that the usage is not implemented, but will simply ignore the callfunction and generate otherwise functional code.
Xyce Project Bugs/35: Xyce/ADMS does not allow ceil and floor in analog functions	A mistake in the code for generating analog function derivatives caused any analog function that used ceil or floor to abort processing of an entire Verilog-A model. This mistake has been fixed and now ceil and floor can be used inside analog functions

Table 1: Fixed Defects. Note that we have two multiple issue tracking systems for Sandia Users. SON and SRN refer to our legacy open- and restricted-network Bugzilla system, and Gitlab refers to issues in our gitlab repositories.

Defect	Description
Xyce Project Bugs/27: Transient adjoint sensitivity doesn't correctly handle scaling	Sensitivity analysis in Xyce can optionally scale sensitivity values by `p/100`, where `p` is the original value of the sensitivity parameter. This particular capability was never implemented for transient adjoints, which was an oversight. This has been fixed.
Xyce Project Backlog/534: Xyce needs to remove deprecated features in C++17, like std::binary_function() and std::unary_function().	These features have been replaced with C++17 compliant features.
Xyce Project Backlog/565: Fix VPWL/IPWL independent sources to work with `.STEP`	Piecewise-linear sources were not instrumented to be updated during `.STEP` loops. This has been fixed.
Xyce Project Backlog/566: Expression library breakpoint handling has a problem with `.STEP`	The breakpoint functions in the expression library were not properly reset at the beginning of each `.STEP` iteration. This was causing breakpoints to be overlooked early in the transient. This has been fixed.
Xyce Project Backlog/575: Expression tables are inefficient when they are large and contain expressions	This issue was inspired by backlog issues 565 and 566. If a PWL source is specified using a Bsrc and the `TABLE` operator (instead of a PWL source), then it was very inefficient if the table was large and the entries of that table were based on expressions rather than pure numbers. This was because the subordinate expressions in the table were being updated too frequently. This has been fixed.
Xyce Project Bugs/41: Expression library inconsistently handles subcircuit nodes for derivatives	When invoked from a Bsrc, the expression library has to provide derivatives as well as function evaluations. In the Bsrc, the needed derivatives are with respect to solution variables, which are usually voltage nodes. When a Bsrc is inside of a subcircuit, it is possible that the original expression contains voltage node dependencies that will need to be replaced with the subcircuit instance nodes. When this happens, the new list of nodes might include a duplicate or ground node, even if the original version did not. The expression library wasn't handling this use case (new duplicates) correctly, and this caused a memory error. This has been fixed.
Xyce Project Bugs/54: subcircuit parameters incorrectly deleted when multiple subcircuit instances (X lines) refer to same subcircuit definition	There was a bug in the parser in the function that resolved subcircuit parameters. When subcircuit instance parameters (from the X line) matched a parameter in the list of unresolved parameters in the subcircuit definition, that parameter was incorrectly being erased from the definition container. This has been fixed.
Xyce Project Backlog/147: The handling of the first failed step out of a DC op was incorrect	When Xyce cannot take the first transient step out of a DC op, it incorrectly reported that Xyce reaches the maximum local error test failures. Xyce does not check local truncation error for the first step. This has been fixed. Xyce now handles this case correctly and it also reports the time when Xyce reaches the maximum number of failures without convergence.

Table 1: Fixed Defects. Note that we have two multiple issue tracking systems for Sandia Users. SON and SRN refer to our legacy open- and restricted-network Bugzilla system, and Gitlab refers to issues in our gitlab repositories.

Defect	Description
Xyce Project Backlog/595: Expression library does not recognize device names that include square brackets and a bunch of other unusual (but supported) characters	The expression library could not tokenize a device name such as `V_B[0]`. The use of brackets and other unusual characters was allowed in other tokens such as voltage nodes. For this bug only device name tokens, used in the expression library, were affected. This was an oversight in the expression library lexer code. This has been fixed.
Xyce Project Bugs/55: Null pointer in expression	In the resolution of parameters in subcircuits, expressions could be deleted before they were stored in parameter objects resulting in a null pointer for the expression. This issue has been fixed.
Xyce Project Bugs/59: Mixed-Signal Segmentation Fault	The Python interface to Xyce was incorrectly using an ADC base name when trying to get DAC devices. This has been fixed.
Xyce Project Bugs/60: Error building with Xyce_GRAPH_DEBUG=ON	The bracket operator used in a block of debugging code was incorrect for use on the std::unordered_map object. This code was fixed to use the correct accessor function for the map.
Xyce Project Backlog/611: The devConMap is not implemented in any of the BSIM CMG devices	The devConMap is a data structure provided by each device model to enable the topology package to perform the ”no DC path to ground” diagnostic. The BSIMCMG devices did not have this set up, and so circuits using these devices would sometimes produce spurious warning messages. This has been fixed.
Xyce Project Bugs/61: Circuit causes Segfault	Xyce would get a segmentation fault when the netlist specified a Bsrc that depended on the current thru another Bsrc, that happened to be a current-source style Bsrc. Behavioral sources can either be current or voltage sources, depending on the netlist specification. When specified as a voltage source, they have an internal current variable, which is accessible by other behavioral sources. When specified as a current source, they do not. Xyce was not catching this mistake early enough and this was resulting in a memory error. This has been fixed.

Supported Platforms

Certified Support

The following platforms have been subject to certification testing for the Xyce version 7.7 release.

- Red Hat Enterprise Linux® 7, x86-64 (serial and parallel)
- Microsoft Windows 10®, x86-64 (serial)
- Apple® macOS, x86-64 (serial and parallel)

Build Support

Though not certified platforms, Xyce has been known to run on the following systems.

- FreeBSD 12.X on Intel x86-64 and AMD64 architectures (serial and parallel)
- Distributions of Linux other than Red Hat Enterprise Linux 6
- Microsoft Windows under Cygwin and MinGW

Xyce Release 7.7 Documentation

The following Xyce documentation is available on the Xyce website in pdf form.

- Xyce Version 7.7 Release Notes (this document)
- Xyce Users' Guide, Version 7.7
- Xyce Reference Guide, Version 7.7
- Xyce Mathematical Formulation
- Power Grid Modeling with Xyce
- Application Note: Coupled Simulation with the Xyce General External Interface
- Application Note: Mixed Signal Simulation with Xyce 7.2

Also included at the Xyce website as web pages are the following.

- Frequently Asked Questions
- Building Guide (instructions for building Xyce from the source code)
- Running the Xyce Regression Test Suite
- Xyce/ADMS Users' Guide
- Tutorial: Adding a new compact model to Xyce

External User Resources

- Website: http://xyce.sandia.gov
- Google Groups discussion forum: https://groups.google.com/forum/#!forum/xyce-users
- Email support: xyce@sandia.gov
- Address:

  Electrical Models and Simulation Dept.
  Sandia National Laboratories
  P.O. Box 5800, M.S. 1177
  Albuquerque, NM 87185-1177

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.

SAND2023-07328 O