Xyce™ Parallel Electronic Simulator
Version 6.7 Release Notes

Sandia National Laboratories

May 8, 2017

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

New Features and Enhancements 2

Defects Fixed in this Release 3

Interface Changes in this Release 5

Known Defects and Workarounds 6

Supported Platforms 10

Xyce Release 6.7 Documentation 10

External User Resources 11

Sandia National Laboratories

New Features and Enhancements

New Devices and Device Model Improvements

- The HICUM Level 0 BJT model version 1.32 was added as Xyce BJT level 230.
- The HICUM Level 2 BJT model version 2.34 was added as Xyce BJT level 234.
- A "solution-dependent capacitor", which allows the device capacitance to depend on solution variables such as node voltages, has been supported since **Xyce** Version 5.1.2. It is now documented in the Reference Guide.
- THE VBIC 1.2 (Level=10) MODEL IS NOW DEPRECATED. It will be removed in version 6.8 of **Xyce**. The VBIC 1.3 model (levels 11 and 12) will be the only supported VBIC model in version 6.8 of **Xyce**. Please update any netlists you have that use the level=10 model to use one of these VBIC 1.3 models instead.
- The Multiplicity Factor (M) is now supported for the R, L and C devices.
- Power output is now supported for the component inductors in both linear and nonlinear mutual inductors (K device).

Enhanced Solver Stability, Performance and Features

- HB analysis now uses the block Jacobi preconditioner with GMRES by default. In previous releases, the default solver for HB was unpreconditioned GMRES.
- Transient adjoint sensitivity analysis now works with trapezoid integration. Previously it only worked with gear.

Interface Improvements

- A "-randseed" command line argument has been added to allow the user to specify a seed for the random number generator that is used for the expression library's "rand," "gauss," and "agauss" functions.
- When any expression includes use of a random number function ("rand," "gauss," or "agauss"), or if "-randseed" is given, Xyce will output the seed that is being used for that run.

Deployment Improvements

- **Xyce** binary releases for Windows are now 64-bit executables. Binaries for all prior releases had been 32-bit executables.
- On Sandia HPC and CEE systems, both serial and parallel versions of **Xyce** have been deployed with shared library plugin support built in, and ADMS has been installed and made accessible through the same module load that provides access to Xyce. This enables **Xyce** users of those systems to build their own models in Verilog-A and test them as plugins. See the **Xyce**/ADMS Users' Guide on the external Xyce web site for details on how to use this feature.

Defects Fixed in this Release

Table 1: Fixed Defects. Note that we have two different Bugzilla systems for Sandia users. SON, which is on the open network, and SRN, which is on the restricted network.

Defect	Description
877-SON : Provide capability to force a seed on expression library random number generator	**Xyce**'s expression library has three random number generation functions, `RAND`, `GAUSS`, and `AGAUSS`. In previous releases, **Xyce** would always select a seed for the random number generator that was based on the current time of day. This remains the default behavior, but as of this release, the user may also specify a specific seed to be used by using the "-randseed" command line option. "-randseed" takes a single argument, the seed to be used. Additionally, **Xyce** will output text indicating the seed being used whether "-randseed" is specified or not.
864-SON : Xyce hangs when resolving parameter	An error in the expression library led to **Xyce** entering an infinite loop when a parameter (`.PARAM`) was defined with an expression involving comparison of an undefined parameter with some value using the not-equal ($!=$) operator. After the fix for the bug, **Xyce** will emit an error about being unable to resolve the parameter being defined in the `.PARAM` statement.
838-SON: Shared library plugins do not work in parallel	While binaries of **Xyce** are not distributed supporting this capability, is possible to build **Xyce** from source using options to allow it to load device models as shared library plugins. It was discovered for Xyce 6.6 that this capability did not work correctly in parallel. The loading error was identified and fixed, and parallel runs of **Xyce** can now load shared library device plugins. If both serial and parallel builds of **Xyce** were created with the same base compiler, a shared library created for the serial build can be loaded by the parallel build.
833-SON: Incorrect error message when requesting power for unsupported devices (K, O, U and some Y devices) or non-existent device names	Netlist parsing will fail and produce an error message if power (`P()` or `W()`) is requested either for a device (e.g., `K1`) that does not support power or for a device (e.g., `RBOGO`) that does not exist in the netlist. For a device like `K1`, those error messages will now reference `P(K1)` rather than `I(K1)`. For a device like `RBOGO`, the error message is now clearer that the issue is with the `RBOGO` instance rather than with with power calculations for the `R` device.
839-SON: Add warning/error message (about using default NORM) to ERROR measure	Xyce 6.6 would silently default to using the L2NORM if an `ERROR` measure either did not specify a value for the `COMP_FUNCTION` qualifier, or specified an invalid value. Xyce 6.7 explicitly says what `COMP_FUNCTION` value (`INFORM`, `L1NORM` or `L2NORM`) was used in the descriptive output for each `ERROR` measure.
845-SON: Make the lossless transmission line device work correctly with .STEP	`.STEP` now works correctly with the `TD`, `NL`, `F`, and `Z0` instance parameters for the Lossless Transmission Line device.
848-SON: Make the power grid devices work correctly with .STEP	`.STEP` now works correctly with the appropriate instance parameters for power grid devices. Consult the **Xyce** Reference Guide for more details.
859-SON: Make the thermal resistor work correctly with .STEP	`.STEP` now works correctly with a thermal resistor device (level 2 resistor) that only specifies `L`, and not `A`.

Table 1: Fixed Defects. Note that we have two different Bugzilla systems for Sandia Users. SON, which is on the open network, and SRN, which is on the restricted network.

Defect	Description
874-SON: Incorrect lead current and power values are output, when the `IC` instance parameter is used for a capacitor	If the `IC` instance parameter, which set the initial voltage drop across the capacitor, was used for a capacitor `(C1)`, then the lead current `I(C1)` would only be correct for t=0. It would always be reported as zero for t>0. The power `P(C1)` would always be reported as 0 at t=0. These errors have been fixed.
856-SON: Fix memory bloat in transient adjoint sensitivities	Transient adjoint sensitivities previously used a dense storage scheme for information saved during the forward solve. This has been fixed so that now it uses sparse storage. As a result, the transient adjoint sensitivities capability is much more practical.
862-SON: Update distribution scripts to bundle all three versions of libmkl_avx* for OS X builds	Mac machines with different processors require different versions of the blas functions. Now the OS X installs should work on all machines
2044-SRN: chargeSimpleMPDE.cir fails in parallel build serial run mode	This is due to a failure of the default iterative linear solver to converge in the MPDE solve. However, the MPDE analysis was not acknowledging the linear solver options line so that the solver could be changed. This has been fixed.

Interface Changes in this Release

Table 2: Changes to netlist specification since the last release.

Change	Detail
Some of the default values for the instance parameters for the Power Grid Devices have changed.	Several of the instance parameters for the Power Grid devices were changed from 1 to 0, so those parameters could be omitted from the instance line if they were zero. .STEP compatibility also necessitated a change in the allowed values for the Transformer Type. The **Xyce** Reference Guide has more details.

Known Defects and Workarounds

Table 3: Known Defects and Workarounds.

Defect	Description
855-SON: Missing error message when a netlist uses an operator (e.g., IR or P) that is not supported for .AC analyses	This is related to SON Bug 718. **Xyce** will output all zeroes or all NaNs, for the requested quantity, when a netlist uses an operator (e.g., `IR` or `P`) that is unsupported for .AC analyses. Instead, **Xyce** should report a netlist-parsing warning or error for this case. ***Workaround***: There is none, other than noticing that an output waveform value is unexpectedly all zeroes or all NaNs.
850-SON: Segfault and improper error handling for improperly formated functions	When a user-defined function is called with the wrong number of parameters, **Xyce** is supposed to abort with an error message. However, in some cases Xyce segfaults instead. There are also cases where **Xyce** will improperly run a simulation with an incomplete argument list for a user-defined function. ***Workaround***: None, other than using **Xyce** user-defined functions correctly.
812-SON: Undocumented limitations on, and bugs with, parameter and global parameter names	Based on external customer input and pre-release testing, there are some bugs and undocumented limitations on parameter and global parameter names in **Xyce**. Parameters and global parameters should start with a letter, rather than with a number or "special" character like #. In addition, the use of a single character V as a global parameter name can result in either netlist parsing failures or incorrect results from `.PRINT` lines.
807-SON: BSIM4 convergence problems with non-zero rgatemod value	There have been reports of convergence problems (e.g., the **Xyce** simulation fails part way through and says that the "time step is too small") when the `rgatemod` parameter is non-zero.
805-SON: Parser fails to collect linear mutual inductors in an include file	**Xyce** will fail to collect component inductors together, into a mutual inductor, when they are in an include file rather than in the top-level netlist file. ***Workaround***: Move the relevant L and K device statements into the top-level circuit.
794-SON: Bug in TABLE Form of Xyce Controlled Sources	In some case, a **Xyce** netlist with a controlled source, that uses the TABLE form, will get the correct answer at first. However, it may then "stall" (e.g, keep taking really small time-steps) and never complete the simulation run. ***Workaround***: In some cases, the TABLE specification for the controlled source can be replaced with a Piecewise Linear (PWL) source that uses nested IF statements.
785-SON: Xyce hangs in parallel when passed a directory rather than a file	This bug is related to SON Bug 730 ("Xyce hangs when passed a directory instead of a netlist "), which was fixed in Xyce 6.5. Further testing has shown that **Xyce** will hang (during parallel execution) if a directory is used, rather than a file, in `.LIB` or `.INC` statements. **Xyce** may also hang in parallel if the "output file" is either a directory or a file in a non-existent directory. This can happen with either the `FILE=` keyword on a `.PRINT` line or with the `-o` command line option.
783-SON: Use of ddt in a B-Source definition may produce incorrect results	The `DDT()` function from the **Xyce** expression package, which implements a time derivative, may not function correctly in a B-Source definition. ***Workaround***: None.

Table 3: Known Defects and Workarounds.

Defect	Description
727-SON: Xyce parallel builds hang randomly on OS X	During Sandia's internal nightly testing of the OSX parallel builds, we see that **Xyce** "hangs on exit" with an estimated frequency of less than 1-in-5000 simulation runs. We have not seen this issue with parallel builds for either RHEL6 or BSD. The hang is on exit, whether on a successful exit or on an error exit. The hang occurs after all of the **Xyce** output has occurred though. So, the user will get their sim results, but might have trouble if the individual **Xyce** runs are part of a larger script. ***Workaround***: None.
718-SON: Missing error message for invalid nodes in expressions on .PRINT lines	If an invalid node is specified on a **Xyce** `.PRINT TRAN` line then **Xyce** should return a fatal error during netlist parsing (e.g., `.PRINT TRAN V(BOGONODE)` will produce an error message of `undefined symbol in .PRINT command: node BOGONODE`, if `BOGONODE` does not exist in the netlist). However, if the invalid node is inside a **Xyce** expression (e.g., `.PRINT TRAN {V(BOGONODE)}`) then **Xyce** will not produce an error message during netlist parsing and the output value for `{V(BOGONODE)}` will be zero for all time-steps. ***Workaround***: There is none, other than noticing that an output waveform value is unexpectedly all zeroes, and correcting the `.PRINT` statement.
715-SON: I(*) for subcircuit nodes does not work properly on .PRINT lines	`.PRINT TRAN I(*)` works for nodes at the top-level of the netlist. However, it will fail during netlist parsing if there are nodes in subcircuits. The error message will be something like `Function or variable I(V:X1:1) is not defined`. ***Workaround***: Explicitly put the desired lead or branch currents, using the fully qualified device names, in the `.PRINT` statement.
707-SON: Behavior for invalid nodes on .FOUR lines and in .MEASURE statements	There are issues with `.FOUR` lines and `.MEASURE` statements that accidentally use node names that are not in the netlist. In that case, the `.cir.four` output file will contain a mix of all zero's and NaN's, and **Xyce** will not produce a warning or error message about the invalid node name. Similarly, the measure statement will run without a warning message about the invalid node name. The measure result will then be zero, rather than FAILED.
661-SON Branch Currents and Power Accessors (I(), P() and W()) Do Not Work Properly in .RESULT Statements	There are two issues. First, `.RESULT` statements will fail netlist parsing if the requested branch current is omitted from the `.PRINT TRAN` line. As an example, this statement (`.RESULT I(R1)`) requires either `I(R1)`, `P(R1)` or `W(R1)` to be on the `.PRINT TRAN` line. Second, the output value, in the `.res` file, for the lead current or power calculation will always be zero.

Table 3: Known Defects and Workarounds.

Defect	Description
652-SON: HB output is buggy	While a straightforward use of `.print HB` works as described in the users and reference guides, several of the documented features do not work as intended. `.print HB_FD` and `.print HB_TD` are intended as a way of specifying variable lists for frequency- and time-domain outputs, respectively. It has been discovered that these only produce output if there are print specifications for *both* frequency and time domain. That is, if only one of `.print HB_FD` or `.print HB_TD` is present in the netlist, no output will be produced at all. ***Workaround***: When performing harmonic balance analysis, always specify enough print lines so that both time- and frequency-domain variables are output. This could be by specifing `.print HB` alone, by specifying both `.print HB` and `.print HB_TD`, or by specifying both `.print HB_FD` and `.print HB_TD`.
583-SON: Switch with RON=0 leads to convergence failure.	The switch device does not prevent a user from specifying `RON=0` in its model, but then takes the inverse of this value to get the "on" conductance. The resulting invalid division will either lead to a division by zero error on platforms that throw such errors, or produce a conductance with "Not A Number" or "Infinity" as value. This will lead to a convergence failure. ***Workaround***: Do not specify an identically zero resistance for the switch's "on" value. A small value of resistance such as 1e-15 or smaller will generally work well as a substitute.
469-SON: Belos memory consumption on FreeBSD and excessive CPU on other platforms	Memory or thread bloat can result when using multithreaded dense linear algebra libraries, which are employed by Belos. If this situation is observed, either build **Xyce** with a serial dense linear algebra library or use environment variables to control the number of spawned threads in a multithreaded library.
468-SON: It should be legal to have two model cards with the same model name, but different model types.	SPICE3F5 and ngspice only require that model cards of the same type have unique model names. They accept model cards of different types with the same name. **Xyce** requires that all model card names be unique.
250-SON: NODESET in Xyce is not equivalent to NODESET in SPICE	As currently implemented, `.NODESET` applies the initial conditions given throughout a full nonlinear solve for the operating point, then uses the result as an initial guess for a second nonlinear solve with no constraints. This is not the same as SPICE, which merely applies the given initial conditions to a single nonlinear solve for the first two iterations, then lets the problem converge with no further constraints. This can lead to a **Xyce** `.NODESET` failing where the same netlist in SPICE might not, if the initial conditions are such that a full nonlinear solve cannot converge with those constraints in place. There is no workaround.
247-SON: Expressions don't work on .options lines	Expressions enclosed in braces ({ }) are handled specially throughout **Xyce**, and may only be used in certain contexts such as in device model or instance parameters or on `.PRINT` lines.
49-SON Xyce BSIM models recognize the model TNOM, but not the instance TNOM	Some simulators allow the model parameter TNOM of BSIM devices to be specified on the instance line, overriding the model parameter TNOM. **Xyce** does not support this.

Table 3: Known Defects and Workarounds.

Defect	Description
37-SON: Connectivity checking is broken for devices with more than 10 leads	The diagnostic code used by the **Xyce** setup that checks circuit topology for basic errors such as a node having no DC path to ground or a node being connected to only one device has a bug in it that causes the code to emit a cryptic error message, after which the code will exit. This error has so far only been seen when a user has attempted to connect a large number of inductors together using multiple mutual inductor lines. The maximum number of non-ground leads that can be used without confusing this piece of code is 10. If your circuit has that type of large, highly-connected mutual inductor and the code exits with an error message, this bug may be the source of the problem. The error message now includes a recommendation to use the workaround below. ***Workaround***: Disable connectivity checking by adding the line `.OPTIONS TOPOLOGY CHECK_CONNECTIVITY=0` to your netlist. This will disable the check for the basic errors such as floating nodes and improperly connected devices, but will allow the netlist to run with a highly-connected mutual inductor.
27-SON: Fix handling of .options parameters	When specifying .options for a particular package, what gets applied as the non-specified default options might change.
1962-SRN: Voltages from interface nodes for subcircuits may not work correctly in expressions on `.PRINT` lines	An expression that uses a voltage from an interface node to a subcircuit on a `.PRINT` line may only work if that voltage node is also used outside of the expression on the `.PRINT` line. A simple example is as follows. The expression `{V(X1:a)*I(X1:R1)}` prints out as 0, unless `V(X1:a)` is also on the `.PRINT` line.
1923-SRN: LC lines run out of memory, even if equivalent (larger) RLC lines do not.	In some cases, circuits that run fine using an RLC approximation for a transmission line, exit with an out-of-memory error if the (supposedly smaller) LC approximation is used.

1903-SRN: Xyce fails to collect several inductors into a linear mutual inductor	In some rare cases with complex include file usage, the mutual inductor syntax with multiple couplings can fail to work. **Xyce** will return an error message that it can not find `L_L1`:
	`L_L1 node1 node1 inductance1`
	`L_L2 node3 node4 inductance2`
	`L_L3 node5 node6 inductance3`
	`L_L4 node7 node8 inductance4`
	`K_K1 L_L1 L_L2 L_L3 L_L4 .999`

1595-SRN: **Xyce** won't allow access to inductors within subcircuits for mutual inductors external to subcircuits	It is not possible to have a mutual inductor outside of a subcircuit couple to inductors in a subcircuit. ***Workaround***: Put all inductors and mutual inductance lines that couple to them together at the same level of circuit hierarchy.

Supported Platforms

Certified Support

The following platforms have been subject to certification testing for the **Xyce** version 6.7 release.

- Red Hat Enterprise Linux® 6, x86-64 (serial and parallel)
- Microsoft Windows 7® and Windows 10®, x86-64 (serial)
- Apple® OS X El Capitan, x86-64 (serial and parallel)

Build Support

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

- Red Hat Enterprise Linux® 7, x86-64 (serial and parallel)
- FreeBSD 9.x and 10.x on Intel x86 and x86-64 architectures (serial and parallel)
- Distributions of Linux other than Red Hat Enterprise Linux
- Microsoft Windows under Cygwin and MinGW.

Xyce Release 6.7 Documentation

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

- **Xyce** Version 6.7 Release Notes (this document)
- **Xyce** Users' Guide, Version 6.7
- **Xyce** Reference Guide, Version 6.7
- **Xyce** Mathematical Formulation
- Power Grid Modeling with Xyce
- Application Node: Using Open Source Schematic Capture Tools with **Xyce**

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 Department,
Sandia National Laboratories
P.O. Box 5800, M.S. 1177
Albuquerque, NM 87185-1177

Sandia National Laboratories is a multi-mission laboratory managed and operated by Sandia Corporation, a wholly owned subsidiary of Lockheed Martin Corporation, for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-AC04-94AL85000.