Xyce™ Parallel Electronic Simulator
Version 6.6 Release Notes

Sandia National Laboratories

November 15, 2016

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

New Devices and Device Model Improvements

- All Verilog-A-derived models now support lead currents. In addition, all two-terminal, BJT, and MOSFET devices derived from Verilog-A support power output:
  - Diodes: JUNCAP200
  - BJTs: VBIC, MEXTRAM, FBH HBT
  - MOSFETs: PSP, BSIM6, BSIMCMG, MVS, EKV
- THE VBIC 1.2 (Level=10) MODEL IS NOW DEPRECATED. It will be removed in version 6.7 of **Xyce**. The VBIC 1.3 model (levels 11 and 12) will be the only supported VBIC model in version 6.7 of **Xyce**. Please update any netlists you have that use the level=10 model to use one of these VBIC 1.3 models instead.
- Power output is now supported for the lossless transmission line (T device), Voltage-Controlled Switch (S device), Current-Controlled Switch (W device), Generic Switch (SW device), JFET (Levels 1 and 2), MOSFET (Levels 1,2,3,6,9,10,14 and 18) and MESFET.
- The PSP MOSFET has been updated to version 103.4. Prior versions of **Xyce** had version 103.1.
- The BSIM6 MOSFET has been updated to version 6.1.1. Prior versions of **Xyce** had version 6.1.0.
- The MEXTRAM BJT has been updated to version 504.12.1. Prior versions of **Xyce** had version 504.11.0.
- The Xyce/ADMS Verilog compiler back-end can now produce correct code for models that collapse internal nodes to ground based on model parameter values.
- The legacy version (102.5) of the PSP MOSFET has been added to **Xyce** as a new MOSFET level 102. This version is provided for compatibility with foundry model cards, as the PSP 103 versions are not backward compatible with the 102 versions.
- The BSIM-CMG model version 110.0.0 has been added to **Xyce** as a new MOSFET level 110. Because this is a major version number update, the older BSIM-CMG model version 107.0.0 has been retained as the level 107 MOSFET.
- The Toggle Flip-Flop (TFF) and JK Flip-Flop (JKFF) were added as Behavioral Digital Devices.

Enhanced Solver Stability, Performance and Features

- Improved performance of device evaluation and loading through the separation of device types.
- Improved performance and scalability of parallel netlist parsing and device distribution.
- Improved robustness of the internal KSparse solver by employing KLU for numeric factorization failures.
- NOTE: THE BDF TIME INTEGRATION METHOD (METHOD=BDF or METHOD=6) IS NOW DEPRECATED. It will be removed in version 6.7 of **Xyce**. The Trapezoid (METHOD=TRAP or METHOD=7) and Gear (METHOD=GEAR or METHOD=8) methods will be the only supported time integration methods in version 6.7. Please update any netlists you have that explicitly use the BDF time integration method to use the Gear method instead. If this results in convergence or accuracy issues, please contact the **Xyce** Development Team as soon as possible.
- A new time step error-control method (`MASKIVARS` option), based on types of circuit variables.
- Improved performance and accuracy of transient adjoint sensitivity analysis.

Interface Improvements

- Remeasure now supports `.CSV` files and comma-delimited `.PRN` files.
- New `ERROR` measure calculates the norm between the measured waveform and a “comparison waveform” specified in a file. The supported norms are L1, L2 and INFNORM.

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
804-SON: `outputIC_or_NODESET` is not correct in parallel	`.SAVE` uses this method to output `.IC` or `.NODESET` files after the DCOP calculation. In parallel **Xyce**, this method was only outputting the DCOP solution for one MPI processor, duplicated for the total number of MPI processes. This has been corrected.
836-SON : Segfaults on CSV and Tecplot HB output	**Xyce** would get a segmentation fault and crash if CSV or Tecplot output were used when the number of time points was not equal to the number of frequency points. This would occur if oversampling was used. Standard (PRN) format did not have this problem. **Xyce** 6.6 now handles these alternate output formats correctly when oversampling is employed.
828-SON: “Unable to differentiate” error	All versions of **Xyce** prior to this release would throw a cryptic error, “Unable to diffentiate” when they encountered a use of a user-defined function (one defined via the `.FUNC` mechanism) when the function definition made use of certain “special” variables, such as `TIME`, `TEMP`, or `VT`. This was traced to a bookkeeping error in the expression parser, and is fixed in this release.
547-SON: “Data not in directory” error	A bug in the expression library caused some well-formed expressions in B sources to throw an error. In some versions of **Xyce** this was reported as “Data not in directory” and in others as “Directory node not found”. The issue was tracked down to an error in the handling of terms of the form I(instance) when they were used repeatedly in an expression. The error was a memory access problem, and therefore its behavior was unpredictable — minor variations in how the expression was ordered could change the error or make it go away altogether.
754-SON: Problems with use of dependent parameters for DC source values	The internal handling of global parameters in expressions for device parameters had logic errors that interfered with the handling of DC source sweeps and “source stepping” continuation. The logic errors have been corrected, and these analysis functions now work properly when the devices they impact have their default parmeters set to expressions that depend on global parameters.
813-SON: Fix lead currents in expressions	An error in the ExpressionData class “setup” method led to incorrect data being passed to expressions if the expression involved more than one lead current from the same device (e.g. ID(M1) and IG(M1)). The result was that the value of the first lead current was being reused for subsequent lead currents of the same device in that expression. This error was present in all versions of **Xyce** since at least release 6.2. It is fixed in Release 6.6.

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
691-SON: Add power and lead current support to Xyce/ADMS	Xyce/ADMS now generates lead current support code for all Verilog-A models, and generates code to support P() and W() (power) output for two-terminal devices and those devices imported as Q (BJT) devices. For two-terminal devices, the lead current is accessed with I(devicename). For BJTs, the collector, base, and emitter lead currents are accessed with IC(devicename), IB(devicename), and IE(devicename), respectively. For MOSFETS, the drain, gate, and source currents are accessed with ID(devicename), IG(devicename), and IS(devicename). For BJTs and MOSFETs that have more than three nodes on their instance lines, the additional lead currents (including the substrate and bulk nodes for BJTs and MOSFETs) are printed using I#(devicename), where # is the position of the lead on the instance line (e.g. I4, for substrate and bulk nodes)
736-SON: Error Messages for Lossy Transmission Line (LTRA) Device	**Xyce** 6.5 would sometimes not emit all of the relevant error messages for an invalid instance line for a Lossy Transmission Line (LTRA) device during parallel execution. This is fixed now.
746-SON: Incorrect Information in RAW File Output	The information in the Variables section of the .RAW file output could be incorrect in **Xyce** 6.4, or earlier. The variable names (and subsequent data) in the Values block was correct. However, the "Type" (e.g., current or voltage) for each variable could be incorrect in the Variables section of the .RAW file output. This bug was actually fixed in the **Xyce** 6.5 release. However, it was not fully tested for parallel execution until the **Xyce** 6.6 release.
763-SON: Support remeasure of .CSV files and comma-delimited .PRN files	**Xyce** 6.5 only supported remeasure of `.CSD` files and `.PRN` files delimited with whitespace. **Xyce** now also supports the remeasure of `.CSV` files and comma-delimited`.PRN` files.
802-SON: Fix Core Dumps on Ill-Formed Device Instance Parameters	**Xyce** 6.5 could core dump on some ill-formed instance parameters, like `TC` for the resistor. Examples would be `TC=` where both values for the vector parameter were missing, or `TC=0,` where the second value was missing. This could also occur for non-vector instance parameters. An example would be `TEMP=`. This is fixed now.
817-SON: Fix Error in Lead Current Calculations for Lossless Transmission Line	The lead current calculations for "Terminal 2" of the lossless transmission line (T Device) would have the correct amplitude but the wrong sign. This has been corrected. The polarity conventions are now that positive current flows into the positive node of the specified terminal, and negative current flows out of the positive node of the specified terminal. The Reference Guide section was also updated to correctly refer to Terminals 1 and 2, rather than Terminals A and B. So, the lead currents for the two terminals, of the T device `Line1`, are accessed with `I1(TLine1)` and `I2(TLine1)`.
182-SON: Test case NEURON/HH_Patch.cir fails in paralel testing.	This test case was sensitive to how the device's internal variables were interpolated for output in parallel. Output interpolation routines were improved in an earlier **Xyce** release so that this test now works correctly.

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
704-SON: **Xyce** segfaults when attempting to print current outputs like IP (phase) on lead currents in AC analysis.	The underlying data for lead current output is not available during AC analysis. However, the code did not safely check for the existence of the lead current data structure before trying to use it to output a current. The code now safely checks the lead current data structure before using it for output. Since lead currents are not calculated under AC analysis, **Xyce** will output zero for the `IR`, `II`, `IP` and `IM` lead current quantities. **Xyce** will output a meaningless value (e.g., `NAN`) for the `IDB` lead current quantity.
667-SON: Make `.IC` and `.NODESET` usable inside subcircuits	`.IC` and `.NODESET` statements within subcircuits were silently ignored by **Xyce**. This has been corrected. Now `.IC` and `.NODESET` statements inside subcircuits will be resolved by the parser, eliminating the need to move the statements to the top-level of the circuit.
815-SON: Spice strategy for DCOP solve doesn't properly enforce `.IC/.NODESET` conditions	The internal Spice strategy for performing the DCOP calculation was ignoring `.IC` and `.NODESET` statements if the initial Newton solve failed. This has been corrected.
830-SON: HB does not work with `.IC`	HB analysis would exit with an error when `.IC` was used in the netlist. This has been corrected. The `.IC` statements only apply to the transient phase of HB, which is used to generate a good initial guess for HB analysis.
1823-SRN: Implement collapse of nodes to ground in the Xyce/ADMS back-end	Xyce/ADMS can now handle Verilog-A models that collapse internal nodes to ground via contributions of the form "`V(node) <+ 0;`". In version 6.5 of **Xyce** this appeared to work, but generated incorrect code that could crash **Xyce** under some circumstances. In versions of **Xyce** prior to 6.5, Xyce/ADMS would have generated a fatal error refusing to proceed upon encountering such a contribution. Xyce/ADMS is not able to collapse *external* nodes to ground, and cannot be made to do so at this time.
797-SON: Transient Adjoint sensitivity calculation incorrect when dQdx-terms are non-constant	There was a book-keeping mistake that prevented the correct dQdx matrix terms from being used during the reverse time integration required by adjoints. This has been fixed.
799-SON: Transient adjoint sensitivity analysis has "glitch" at the last time point when using Backward Euler	An indexing mistake that was specific to Backward Euler time integration caused the t=tmax time point to be evaluated incorrectly. This has been fixed.
842-SON: Transient adjoint sensitivity analysis has incorrect answers near time=0	There were a few mistakes related to saving the solution and function derivative histories. These histories are stored during the forward solve, and then used during the reverse time integration used by transient adjoint sensitivity analysis. The mistakes were related to the DCOP history, and thus had the most impact on the adjoint calculation near time=0. This has been fixed.

Known Defects and Workarounds

Table 2: Known Defects and Workarounds.

Defect	Description
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 has been discovered that this capability does not work correctly in parallel. There is currently no workaround, and the issue was discovered too late to fix it for Release 6.6. In Release 6.6, it is only possible to use shared library plugins in serial builds, or in single-processor runs of parallel builds.
833-SON: Incorrect error message when requesting power for unsupported devices (K, O, U and some Y devices)	Netlist parsing will fail and produce an error message if power (`P()` or `W()`) is requested for a device that does not support power. However, for example for device `K1`, those error messages will reference `I(K1)` rather than `P(K1)` or `P(K1)`.
819-SON: Transient sensitivity analysis output is not interpolated when used with .options timeint output_interval	Normally, any transient output is automatically interpolated if the user requests outputs at specific times. The adjoint sensitivity calculation does not do this, and will simply output sensitivities from the time point nearest the requested one. ***Workaround***: In general, transient adjoint sensitivity calculations are not efficient when applied to entire waveforms, so in practice this isn't the best way to get transient sensitivities for specific time points. A better way to get sensitivities from specific time points is to use the `.options sensitivity adjointTimePoints` parameter, which allows the user to specify a list of specific time points at which to compute sensitivities.
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 cases, 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.

Table 2: Known Defects and Workarounds.

Defect	Description
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 version 6.5 of **Xyce**. 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.
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 simulation 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 a 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 zeros and meaningless values (e.g., NaN), and **Xyce** will not produce a warning or error message about the invalid node names. Similarly, the measure statement will run without a warning message about the invalid node names. 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 2: Known Defects and Workarounds.

Defect	Description
652-SON: HB output is buggy	While a straightforward use of `.print HB` works as described in the **Xyce** 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 `.NODESET` failing in **Xyce** 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 2: 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 6.6 release.

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

Note that the **Xyce** team has dropped Certified Support for Red Hat Enterprise Linux® 5. RHEL5 is now in the "Build Support" category.

Build Support

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

- 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 6
- Microsoft Windows under Cygwin and MinGW.

Xyce Release 6.6 Documentation

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

- **Xyce** Version 6.6 Release Notes (this document)
- **Xyce** Users' Guide, Version 6.6
- **Xyce** Reference Guide, Version 6.6
- **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.