Compiled Verilog-A language combined with SmartSpice provides circuit designers and model developers with an easy-to-use, comprehensive environment for the design and verification of complex analog and mixed-signal circuits and models.

- SmartSpice Verilog-A is within 2x runtime performance of C-compiled ADMS models
- Supports top-down design via behavioral modeling and bottom-up verification of analog and mixed-signal designs
- Enables compact model engineers to easily develop proprietary models for any semiconductor technology
- Enables executable specification to connect analog and digital engineers in a single design project
- Provides secure, transportable method for analog IP distribution and evaluation through encryption full/partial and binary files
- A Verilog-A debugger is available under GUI mode of SmartSpice to aid model code development. Verilog-A parameter evaluations are shown enabling user to step through model code
- SmartSpice Verilog-A is within 2x runtime performance of C-compiled ADMS models
- Compatible with all analog features of the Verilog-AMS 2.3.1 language specification
- Implements small-signal and noise sources in multiple distributions (Gaussian, exponential, Poisson, chi-square, Student's T, and Erlang)
- Executes analog operators, including time integral/derivative, partial derivative, transition, slew, Laplace transform and Z-transform
- SmartSpice can accept any combination of netlist, C, C++, and Verilog-A for mixed mode execution
- Verilog-A devices can be referenced using the subcircuit X call
- Verilog-A modules can be referenced using the .MODEL statement
- Memory usage and runtime have been significantly reduced by generating a sparse matrix for individual Verilog-A modules
- Support for encrypting or partial encryption of Verilog-A source allows distribution of proprietary models without contents disclosure
- Supports Single Event Upset (SEU) analysis

- Enables analog designers to build executables for designs of phase-locked loops, VCOs, A/D, D/A, etc. for prototyping purposes before detailed circuit design
- Allows designers to describe digital components as sub-circuits for mixed-signal designs such as Sigma-Delta converters
- Powerful graphical post-processor allows waveform overlays to speed up mixed-signal debugging

- SmartSpice Optimizer can be used for an input deck which includes Verilog-A module(s)
- Optimization targets can be a combination of:
  - delay, rise time, fall time, and power
  - DC, AC, transient combination of curves
- This can be achieved by optimizing parameters such as:
  - Transistor lengths and widths
  - Device model parameters
Compact Model Development

- Verilog-A compact models are compiled into binary code for rapid execution
- Enables easy development of proprietary SPICE models for specific technologies
- Integrated development and debugging environment accelerates compact model development
- Enables mixing of SPICE model statements and Verilog-A modules in the same SmartSpice netlist

Verilog-A environment allows development of compiled models for any .DC, .TRAN, .AC, .NOISE, or .TEMP capability.

```
module D_PLL(RF, OUT, ref);
inout RF, OUT, ref;
electrical RF, OUT, ref;
parameter real tau = 1m from (0:inf);
parameter real loop_gain = 1 from (0:inf);
parameter real fc = 2.5k from (0:inf);
real c_nom;
real r_nom;
electrical LO, IF;
PD #(.gain(2), .type(`mult)) pd(LO, RF, IF);
VCO #(.gain(loop_gain/2), .fc(fc)) vco(OUT, LO);
initial begin
  c_nom = 170n;
  r_nom = tau/c_nom;
end
analog begin
  V(OUT, IF) <+ I(OUT, IF)*r_nom;
  I(OUT, ref) <+ ddt(c_nom*V(OUT, ref));
```

A digital PLL example showing a design flow using Verilog-A.
Verilog-A Inputs/Outputs

- Verilog-A source files
- Encrypted Verilog-A source file
- Berkeley/HSPICE/Spectre netlist
- Precompiled Verilog-A model
- Verilog-A debugger

Verilog-A For SmartSpice

- User compiled Verilog-A model
- Waveforms in SPICE formats
- Simulation history
- Raw files, output listings log and error files