

# Verilog-A

**Verilog-A Language - Source, Compiled and Encrypted**



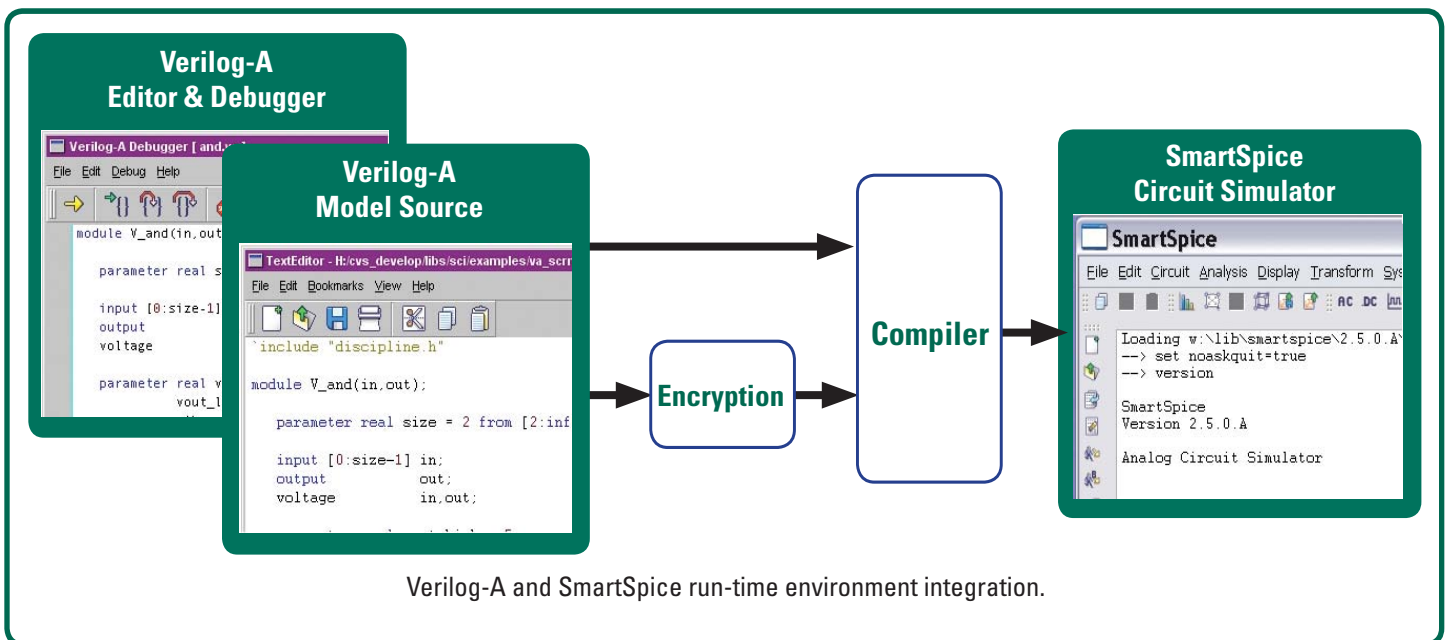
Compiled Verilog-A language combined with SmartSpice provides circuit designers and model developers with an easy-to-use, comprehensive environment for the design and verification of complex analog and mixed-signal circuits and models.

- **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**

**SILVACO**

## Verilog-A Key Features

- 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



## Analog Behavioral Modeling Environment

- 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

## Use of SmartSpice Optimizer with Verilog-A

- 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

The image shows a Verilog-A model for an EKV PMOS transistor. The code defines the model structure, including node definitions, device input variables (length L, width W, multiplier M, series multiplier NS), and process parameters (gate oxide capacitance COX, metallurgical junction depth XJ). A graph plots the drain current  $I_D$  (A) against the drain-source voltage  $v_{dd}$  (V). The current starts at 0A for  $v_{dd} = 0$  and increases, leveling off at approximately -4.5  $\mu$ A for  $v_{dd} > 4$  V. The circuit diagram shows the PMOS transistor connected to a VVIN source (-1V) and a VVDD source (5V), with a load consisting of an inductor L = 10  $\mu$ H and a resistor W = 20  $\Omega$  in series.

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

## 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

The image illustrates a digital PLL design flow. On the left, a Verilog-A module `D_PLL` is defined with parameters for time constants, loop gain, and feedback frequency. The circuit schematic in the center shows the PLL implementation with a phase-locked loop (PLL) block, a voltage-controlled oscillator (VCO), and various control elements. On the right, simulation waveforms are shown for the PLL output at  $t = 0$  and  $t = 1.5 \mu$ s. The waveforms include the reference clock, feedback clock, and VCO input, demonstrating the PLL's locking behavior.

```

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));
end

```

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

# Verilog-A Inputs/Outputs



# SILVACO

## HEADQUARTERS

4701 Patrick Henry Drive, Bldg. 2

Santa Clara, CA 95054 USA

Phone: 408-567-1000

Fax: 408-496-6080

## CALIFORNIA

[sales@silvaco.com](mailto:sales@silvaco.com)

408-567-1000

## MASSACHUSETTS

[masales@silvaco.com](mailto:masales@silvaco.com)

978-323-7901

## TEXAS

[txsales@silvaco.com](mailto:txsales@silvaco.com)

512-418-2929

## JAPAN

[jpsales@silvaco.com](mailto:jpsales@silvaco.com)

## EUROPE

[eusales@silvaco.com](mailto:eusales@silvaco.com)

## KOREA

[krsales@silvaco.com](mailto:krsales@silvaco.com)

## TAIWAN

[twsales@silvaco.com](mailto:twsales@silvaco.com)

## SINGAPORE

[sgsales@silvaco.com](mailto:sgsales@silvaco.com)



[WWW.SILVACO.COM](http://WWW.SILVACO.COM)

Rev 070113\_35