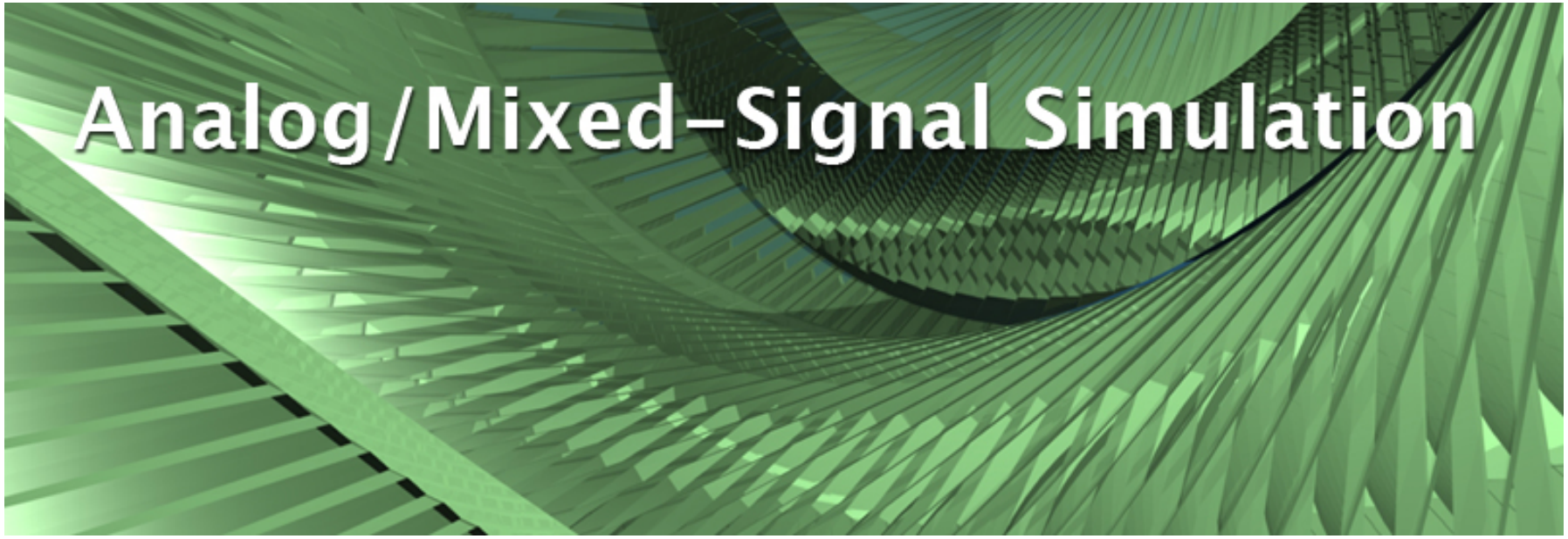
SmartSpice – Functionality Development



2004 Development

MULTI CORE PDK TOKEN WORLD LICENSE

SILVACO

SmartSpice Functionality Agenda

- General Functionality Improvements
- Performance Enhancements
- HSPICE™ Compatibility

SmartSpice Functionality Agenda

General Functionality Improvements

- Performance Enhancements
- HSPICE™ Compatibility

General Functionality: Multiple Nested Sweep

- Powerful tool for the multi-variable parametric analysis
- Enables complex functional dependence of variables and circuit parameters
- Developed as an extension for the standard sweep methodology
- All sweep levels have the same syntax
- Multiple nested sweep expands capabilities of .MODIF, .ST and .ALTER by nesting (theoretically) unlimited number of parameters in a loop, executing them one at a time

General Functionality: Multiple Nested Sweep Capability

- Currently used in .DC, .AC, .TRAN analyses
- To perform multiple nested sweep:
  - Save current variable states and initializes all sweep variables
  - Perform the specified analysis stepping through each parameter for the first sweep variable
  - When the loop of the first sweep variable is finished, SmartSpice initializes the first sweep variable and steps the second sweep variable
  - Repeat the above until the loop of the second sweep variable is finished

General Functionality: Multiple Nested Sweep Depth

Multiple nested sweep can be nested to any depth, such as:

for (,,)
  for (,,)
    for (,,)
      for (,,)

- The total number of the analysis runs is a product of LOOP1 * LOOP2 * LOOP3 * … : LOOP1, LOOP2, LOOP3 … are the loop numbers for the corresponding sweep variables
- After completing all loops, SmartSpice restores the states of all sweep variables

General Functionality: .OVERSHOOT – Applications and Setup

- .OVERSHOOT is a new SmartSpice function used for detecting and reporting voltage spikes on node voltage waveforms
- Enables designers to verify all waveforms over the entire .TRAN simulation window
- Reports nodes with voltages exceeding limits specified by minimum and maximum voltage thresholds
- To minimize false errors, designers can specify minimum spike duration and a list of nodes to be ignored

General Functionality: .OVERSHOOT – Reporting Output

- .OVERSHOOT reports include
  - For each node with overshoot, the report lists
    - V(x) > Vmax or
    - V(x)< Vmin
    - Listing of spikes detected
  - For each spike detected, the report lists
    - Spike start time
    - Spike end time
    - Spike duration
    - Peak voltage

General Functionality: Stop-Continue Algorithm

- Algorithm is used for pausing simulation and analyzing the intermediate results during long-running .TRAN analysis. There is no limitation on how many times to pause and restart simulation
- Three new commands implement Stop-Continue algorithm:
  - Pause .TRAN analysis
  - Continue .TRAN analysis
  - Cancel .TRAN analysis
- Additional feature under development (release after Jan. 2004)
  - A file with SmartSpice “image” will be created on the hard drive
  - The user can later go back to that simulation timepoint, re-parse netlist, load “image” and continue simulation from that timepoint
  - This preserves all simulation data in the case of a computer or network failure

General Functionality: Safe Mode Option

- New option to monitor SmartSpice main memory and disk space usage. It will notify a user through a pop-up window when system resources are exhausted. At that moment a user can (in order to continue simulation):
  - Free up RAM by closing other applications
  - Free up disk space by deleting files
  - Increase virtual memory or swap space

General Functionality: Raw-file Parametric Data Output Option

- New option to allow writing of parametric analysis data into raw-file
- Results from executing .ST, SWEEP, .MODIF, .ALTER analyses can be written and analyzed with SmartView

General Functionality: Flattened Sweep and Multi-Sweep Options

- New options to provide concise representation of the simulated swept data in the raw-file
- All curves from sweep family are represented as one long vector. Headers are eliminated from the raw-file to save space and improve viewing speed

General Functionality: .EQUIV Statement

- New .EQUIV statement provides aliases for model names located in different libraries. This statement increases flexibility to share model libraries and netlists among many users
- Syntax
  - .EQUIV NewName1=OldName1 <NewName2=OldName2 …>
  - NewName is model name used in .MODEL statements. OldName is model name used in device statements of the netlist

General Functionality: Portability of the Raw-File

- SmartSpice contains internal raw-file converters for different operating systems
  - UNIX
  - Linux
  - Windows
- Any conversion combination is supported

Agenda: Performance Improvements

General Functionality Improvements

- Performance Improvements
- HSPICE™ Compatibility

Performance Improvements: Distributed .ALTER

- An option that allows user to distribute .ALTERs on remote UNIX computers
  - Each “distributed alter” can operate in the multi-CPU mode
  - Feature gives significant increase in the simulation speed by utilizing all available networked hardware
- Invoked automatically by using “remote shell” UNIX command

Performance Improvements: Dynamic Hierarchical Parsing Engine

- New option to speed up parsing by
  - taking advantage of the hierarchy of the simulated circuit
  - detecting expressions in the subcircuits (cells) for R, C, L, A devices and places these expressions into the special structure
- Dynamic Hierarchical Parsing Engine is highly efficient for circuits with deep hierarchy when basic cell contains expressions
- The parsing phase speed increases by 10 to 25 times for large hierarchical circuits

Performance Improvements: Multithreading, Pool of Threads

- New option to reduce runtime when performing multithreaded parallel simulation
- The option creates a user-specified number of threads at the beginning of the run and re-uses them from the pool of threads during the entire run as needed, rather than creating and destroying a thread each time when needed

Performance Improvements: Circuit Topology Change Detection (release Jan. 2004)

- New algorithm to detect any change of the circuit topology at each step during execution of any parametric analyses statement: .ALTER, .MODIF, .ST, SWEEP, MULTI-SWEEP,.TEMP.
- Based on the detection result, if topology did not change, the internal data structures are not rebuilt (as it is traditionally done), and simulation proceeds without interruption
- Simulation speed increases on average 8-16%

Performance Improvements: Split of Foreground/ Background Operations

- A unique capability to enhance monitoring and interrogating SmartSpice results during interactive runtime by splitting foreground/background operations
- With this capability, SmartSpice can be run in the background mode, freeing the command line
- User can utilize this free window and SmartSpice postprocessing full capability to interrogate and monitor dynamically built data. For example, .PRINT or .MEASURE, etc. can be used in real time on the data that is coming out from the simulation

Agenda: HSPICE™ Compatibility

- General Functionality Improvements
- Performance Enhancements

HSPICE™ Compatibility

HSPICE Compatibility: .BIASCHK Capability

- .BIASCHK statement monitors the voltage bias between specified terminals of the particular device during .TRAN analysis and detects if the voltage bias is too large and breakdown can occur.
- .BIASCHK reports the following information:
  - Type of the device
  - Terminals
  - Time at which overreach had happened
  - Voltage bias overreaching the limit
  - Model name
  - Device name
  - Number of times the bias surpassed the limit
- SmartSpice implementation of .BIASCHK supports active and passive devices

HSPICE Compatibility: .BIASCHK Capability (con't)

- SOA – Safe Operating Area is a new capability to allow monitoring of the total power consumption of a specified active device and issuing a warning if the total power exceeds the specified limit

HSPICE Compatibility: Parameter Scoping Capability

- Very important feature to be used when simulating very large designs with very deep hierarchy
- It's very likely in the very large circuits with many levels of hierarchy to have the same names for different parameters at different levels of hierarchy. This creates a conflict when a value is assigned to a parameter name
- Parameter Scoping Capability resolves this conflict and allows different parameter values to be assigned to the same parameter name at different hierarchical level, or parameters that are located on the same hierarchical level but belong to different subcircuit
- The burden to create unique names for everything in a very large circuit is removed from the user

HSPICE Compatibility: AvanWave™ and CSCOPE™ Compatible Outputs

- SmartSpice produces compatible output files in 9601, 9007, 2001 formats and supports long vector node names
- All files can be produced in binary or ASCII formats