RELEASE NOTES

RELEASE NOTES - VERSION 7.0

OVERVIEW

This document provides details of SIMetrix Version 7.0

WHAT'S NEW

NEW FEATURES AFFECTING BOTH SIMETRIX AND SIMPLIS

1. Product name changes. The SIMetrix and SIMetrix/SIMPLIS products have now been renamed as follows:

Old name	New name
SIMetrix AD Plus	SIMetrix Classic
SIMetrix VX	SIMetrix Pro
SIMetrix Micron VX	SIMetrix Elite
SIMetrix/SIMPLIS	SIMetrix/SIMPLIS
SIMetrix/SIMPLIS VX	SIMetrix/SIMPLIS Pro
SIMetrix/SIMPLIS Micron VX	SIMetrix/SIMPLIS Elite

2. Multiple core Multi-step runs for both SIMetrix and SIMPLIS simulators. This feature speeds up multi-step runs such as Monte Carlo analysis by a factor that is close to the system core count. For example, 3.5X improvement has been seen on 4 core machines. Multi-core capability is available with SIMetrix Pro (max. 4 cores), SIMetrix Elite (max 16 cores), SIMetrix/SIMPLIS Pro (max 4 cores) and SIMetrix/SIMPLIS Elite (max 16 cores). See Performance and Multi-Core.
3. Part selector and part search. This is a window that opens on the right hand side of the schematic editor and provides a hierarchical method of browsing all available parts. A search feature is also available. The part selector can be configured by the user if required. See Part Selector
4. Added Jumper Components. New jumper components have been added for both the SIMetrix and SIMPLIS simulators. The jumpers are configurable in both the number of poles and the number of positions. Once placed on a schematic, the jumper settings can be reconfigured at any time. The device models behind the jumper symbols are ideal zero voltage DC sources and open circuits, allowing the use of these components in any circuit.

NEW FEATURES FOR SIMETRIX

1. Performance improvements for SIMetrix simulator. Much work has been done to improve SIMetrix simulator performance mostly, but not exclusively, by exploiting multi-core architecture. Multi-core capability is available with Pro and Elite products (see 1. above). See Performance and Multi-Core
2. New Matrix solver. This is “KLU” developed by Prof. Tim Davis at the University of Florida and is now the default matrix solver used. KLU is faster for most circuits over around 500 nodes and provides better convergence for some circuits. See New Matrix Solver
3. Arbitrary source improvements. A new LIMITS function has been added. This is the same as LIMIT but has a smooth response at the corners. The smoothness can be controlled using a fourth argument to the function. The IF function has an extra argument that allows its slew rate to be defines. This help discontinuous definitions to converge. There is also an option setting to automatically apply this slew rate even if not specified if the definition is detected as discontinuous. See Arbitrary Source Features
4. NXP Simkit device interface implemented. This is the replacement to the old PCM (Philips compact models) interface. This was added primarily to provide multi-core execution which PCM does not support. But it does provide access to some new models. See SIMKIT Interface
5. R3 Resistor model. See CMC R3 Resistor Model

NEW FEATURES FOR SIMPLIS

1. Improved PWL Sources. SIMPLIS has been improved in its handling of the Piecewise Linear sources. Simulations that either use PWL sources with a large number of PWL segments or use PWL sources to simulate line or load transients may a see substantial reduction in the simulation CPU time.
2. New Digital Gate Dialog. A new digital gate dialog has been designed for SIMPLIS Advanced Digital and Classic (Building Block Library) Digital Gates. This new gate dialog allows the changing of gate type (AND, OR, etc.), number of logic inputs, as well as the ground reference visibility. When editing gates in version 7.0, this new dialog will be used, while editing gates using older versions of SIMetrix/SIMPLIS, the old dialog will be used. Schematics with gates placed using older versions of SIMetrix/SIMPLIS will continue to simulate in version 7.0. The converse is also true. Schematics with gates placed using version 7.0 will simulate in older versions of SIMetrix/SIMPLIS.
3. Added Schottky Diode Placement Options. Placement options for Schottky Diodes have been added to both the Place Menu and to the Parts Selector. The Schematic Menu is located at: Place -> Semiconductors -> Schottky Diode. In the Parts Selector the category is Discretes -> Diodes -> Schottky.
4. New Option for direct placement of Semiconductors. When placing semiconductors via the schematic menu: Place -> Semiconductors or executing keyboard shortcuts for those menu options, the default behavior is to open the "Select Device…" dialog. The old behavior can be invoked by unsetting the option "PSUseSelectDeviceDialogForSemiconductors."

NEW FEATURES FOR DVM

1. X64 support. DVM is now fully supported in the 64 bit versions of SIMetrix and SIMetrix/SIMPLIS
2. Jumper manipulation via testplan. New testplan header category allows users to change the position of the jumpers prior to executing a simulation.
3. New test objectives. The following new Test Objectives can be configured from a testplan:
    - PulseLine
    - PulseLoad
    - ACNoise (SIMetrix simulator only)
1. New sources and loads
    - Square
    - Triangle
    - Sawtooth
    - Sine
    - Cosine
    - Pulse
    - Pulse with exponential decay
2. Fixed probe measurements. Probe measurements on fixed probes are now automatically converted into scalars and reported. This is currently implemented on any probe with a name starting with "DVM."
3. Efficiency measurements. The overall efficiency of a converter is now calculated and reported as a scalar for all Steady-State and AC test Objectives. The POP simulation results are analyzed and the power in each managed source and load is measured and reported as well as the overall efficiency
4. Built-in efficiency testplan. A 1 input, 1 output DC/DC Efficiency testplan has been added and can be run via the usual DVM menu: DVM -> Built-In Testplans
5. New measurement testplan entry. A general purpose Measurement post processing entry has been added. This testplan prefix allows the following new functions to be executed after a simulation run.
    - Alias
    - NoSpecs
    - NoScalars
    - NoCurves
    - ArbitraryBodePlot
    - ArbitraryCurve
    - ExtractCurve
    - CreateXYScalarPlot
    - PromoteGraph
    - PromoteScalar
6. New FinalProcess Testplan Entry. Added testplan option for "finalprocess" - scripts called in the finalprocess column are executed after DVM performs measurements and after Measurement functions. This allows a user to make measurements on or manipulate a graph which was autogenerated by DVM.

7. Misceallaneous.
    - Renamed recovery time scalars for transient simulations
        - When output voltages never leaves regulation, changed return scalar from '0.0' to 'never left regulation'
        - When output voltage leaves regulation and doesn't return, changed '-1' to 'left regulation and never recovered'
        - Added logic to detect rising and falling edges for pulseline() and pulseload() objectives, recovery time scalars are named accordingly
    - Added load voltage and current test for "in regulation" to AC Input LineDropout tests.
    - Added screenshot to individual test report, takes a snapshot of the current top-level schematic in PNG format
    - Added original testplan filename to overview report
    - Improved functions for user reporting of scalar and spec values
    - Bug fixes for sensitivity input file syntax variations

PERFORMANCE AND MULTI-CORE

Multi-core execution has been implemented in two ways:

1. Single run. Device equation evaluation is parallelised. This offers speed improvements varying from none at all for simple circuits to almost 3x for complex IC designs simulated using a 4-core processor. See below for further details
2. Multi-step e.g. Monte Carlo analyses. These are speeded up by sharing the multiple steps amongst multiple processes. So for example, a 100 step Monte-Carlo run is split across 4 processes running on 4-cores so that each process runs 25 steps. Each process runs completely independently so there are no data sharing or synchronisation issues. Speed improvements of around 70-80% of the core count are easily achieved using this method. With 8 cores we routinely achieved 6x speed improvement. Unlike single-run multi-core execution, this technique benefits nearly all circuits and analysis types. But runs which normally take only a few seconds fail to benefit from this method as the overhead to launch the additional processes dominates. See below for further details.

MULTI-CORE LICENSING

Multi-core capability is available with Pro and Elite products as shown in the table below:

Product (new name)	Number of cores supported
SIMetrix Classic	1
SIMetrix Pro	4
SIMetrix Elite	16
SIMetrix/SIMPLIS	1

Product (new name)	Number of cores supported
SIMetrix/SIMPLIS Pro	4
SIMetrix/SIMPLIS Elite	16

IMPORTANT

Be aware that for single runs, there is little performance enhancement to be gained with more than around 4 cores and that only SIMetrix simulations benefit from this. The benefit with a higher core count is with the Multi-step features such as Monte Carlo analysis and this applies to both SIMetrix and SIMPLIS simulations. With 16 cores we have seen 10-12x speed enhancements for large Monte Carlo runs on modest sized circuits with both SIMetrix and SIMPLIS simulators.

Single Run Details

This feature usually requires no interaction with the user. SIMetrix will automatically choose how many cores to use for the simulation. For simple circuits it will use a single core and beyond a certain level of complexity it will use all the cores available on a single chip. So if you have a 4-core machine where all 4 cores are implemented on a single processor chip, SIMetrix will use all 4 cores as long as the circuit complexity is sufficient to justify it.

If you have a machine with, for example, 8 cores implemented using 2 4-core processor ICs, SIMetrix will use just one of the ICs so therefore 4 cores.

You can override the number of cores using the mpnumthreads .OPTIONS setting. E.g.:

.OPTIONS MPNUMTHREADS=2

will force 2 cores to be used as long as the computer system does actually have 2 cores. SIMetrix will not use more threads than there are physical cores available.

Be aware that hyperthreaded logical processors are not counted as a physical processors. So if you have 4 physical cores and 8 logical cores implemented using hyperthreading, SIMetrix will use a maximum of 4 cores.

Using Multi-step Multi-core Simulation - SIMetrix Simulator

To use this feature, it is currently necessary to explicitly enable it. You do this in the **Define Multi Step Analysis** dialog box where you will see an edit box labelled **Number of Cores**. Set this to the desired number of cores for the simulation.

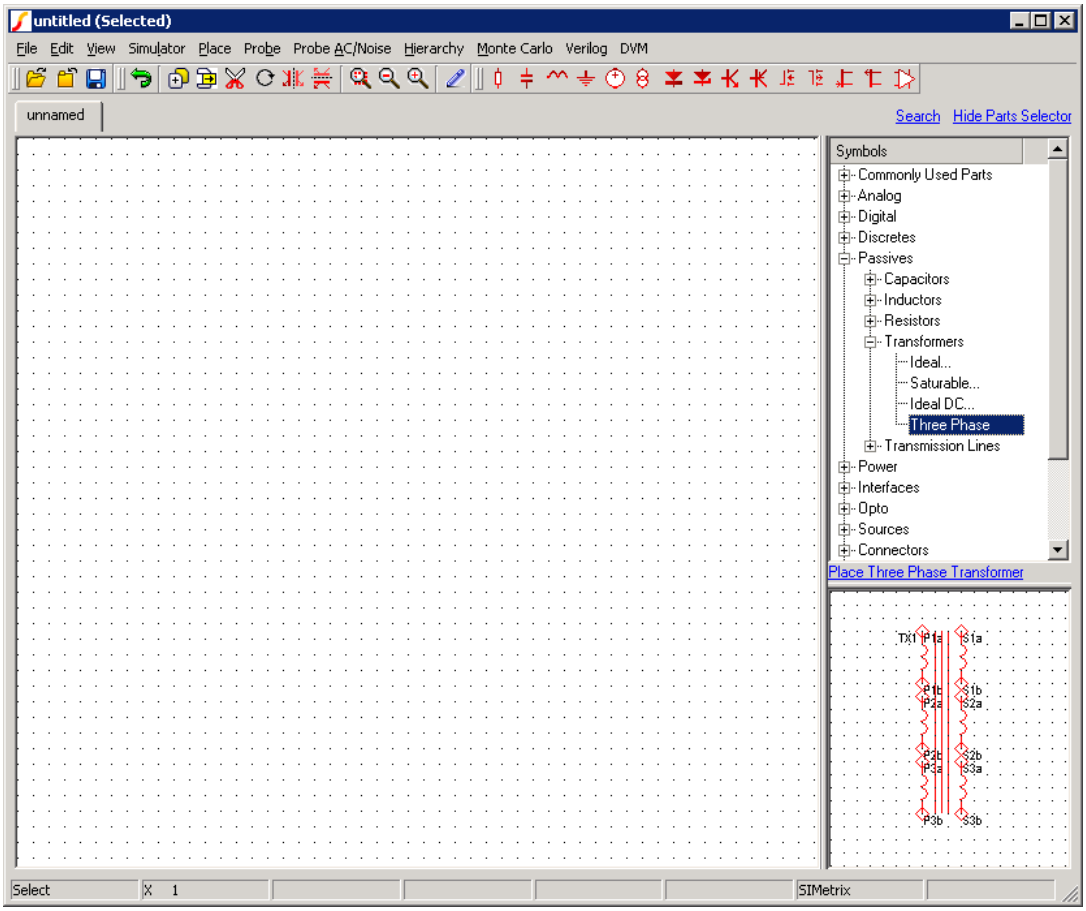
When the run commences you will notice a line for each process appears in the **Analysis Status** window in the simulator status box. Further details are available in the User's Manual.

USING MULTI-STEP MULTI-CORE SIMULATION - SIMPLIS SIMULATOR

This is enabled by setting the number of cores to use in the Multi-step setup dialog box. When the run commences you will notice multiple tabs appear in the SIMPLIS status box. Each of these shows the progress of each process. Further details are available in the User's Manual.

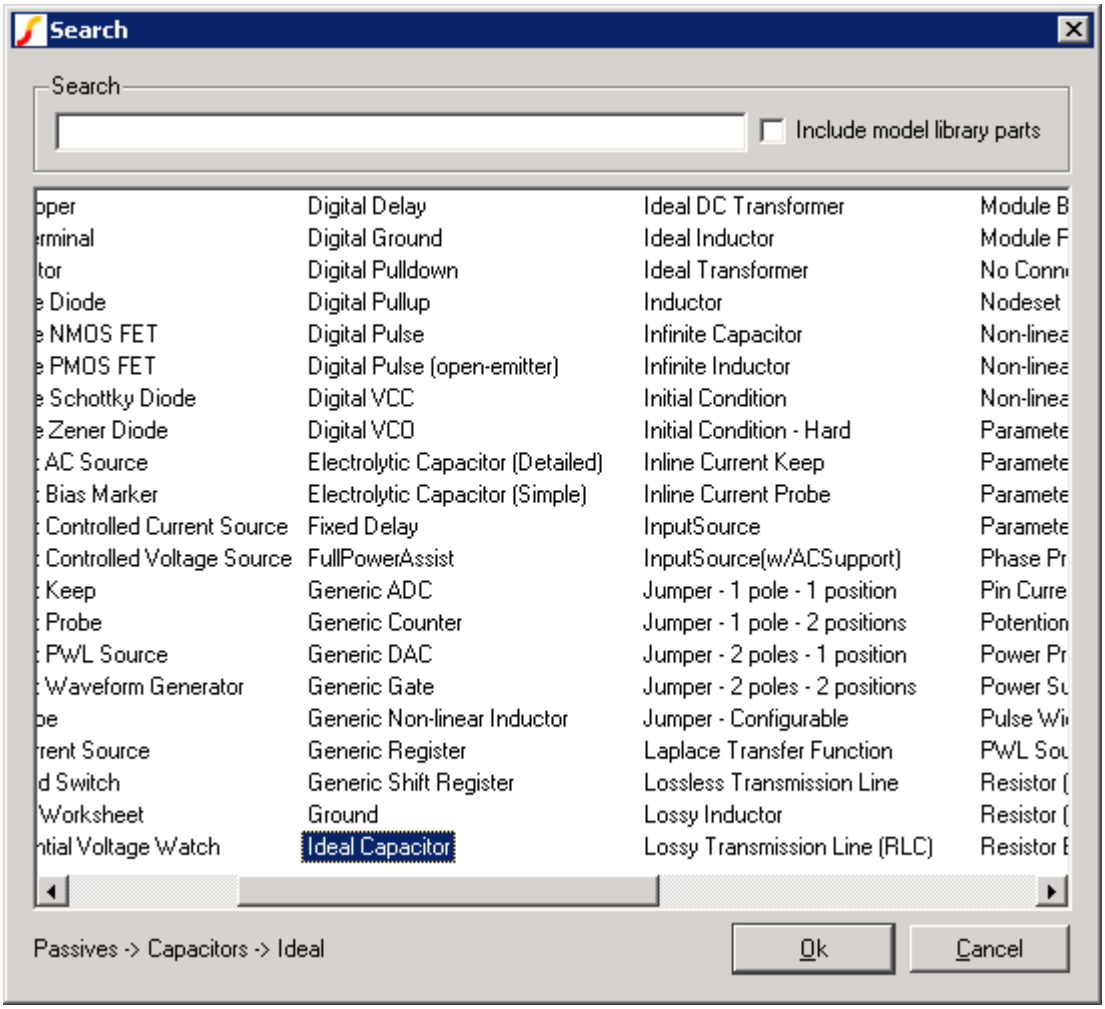
PART SELECTOR

See picture below



The part selector can be hidden when not required simply by pressing the Hide Parts Selector link above it.

There is also a search facility:



The search feature searches the part selector including a list of keywords associated with each part. The list is updated as you type in the search box.

NEW MATRIX SOLVER

OVERVIEW

The SIMetrix simulator now uses a new matrix solver with the original still available as an option. This is called “KLU” and was developed by a research group at the University of Florida under Prof. Tim Davis. This was developed for circuit simulation and was moulded to perform well for the type of matrix that circuit simulators tend to generate. The solver makes use of more modern techniques than the original SPICE3 solver which was developed over 20 years ago.

In our tests, we have found that KLU provides a worthwhile speed benefit for circuits with more than around 500 nodes. It is also showing a convergence improvement for some circuits. We have not found a downside except for very small circuits which run slightly slower. Here is an example

of the results of one benchmark with 2785 nodes run with the 64 bit version:

Version 6.1: run time = 321 seconds
Version 6.2, 4 core, old matrix solver: run time = 177 seconds
Version 6.2, 4 core, KLU matrix solver: run time=126 seconds

Using KLU

The KLU solver is enabled by default and no action is needed to use it.

If you wish to use the old solver, use this option setting:

.options spsolver=ksparse

Note that KLU does not currently support sensitivity analysis and the default SPICE3 solver (known as “KSparse”) will be used for sensitivity runs regardless of the above option setting. All other analysis modes including AC, noise and transfer function are supported.

Matrix Options

There are three matrix options that apply to the original SPICE3 solver. These are PIVTOL, PIVREL and doInductorMNAPreorder. PIVTOL has no effect with KLU while PIVREL has essentially the same function. The default value for PIVREL (1e-3) provides good convergence and speed in most cases. We have found that KLU can provide better performance with PIVREL set to 1e-4 but this will degrade convergence and should be used with care. doInductorMNAPreorder does not work with KLU and should not be enabled.

Technical Details

Details of KLU itself may be obtained from here: http://www.cise.ufl.edu/research/sparse/klu/

Although generally licensed under the LGPL, we have obtained a commercial license for KLU.

Arbitrary Source Features

LIMITS() Function

Same as LIMIT but has a smooth response when moving towards the limiting values. Defined using a single equation that is smooth in all orders of derivative. This gives better convergence than the LIMIT function. LIMITS has a fourth argument that defines the sharpness of the transition to the limiting region. See the Simulator Reference Manual for full details.

IF() Function

The IF function now has an optional fourth argument that defines its maximum slew rate. This helps resolve convergence problems caused by discontinuous definitions which typically occur when both true and false values are constants. A new option setting - discontinuousIfSlewRate - can also be defined to automatically set the slew rate for all instances of IF that have discontinuous definitions. See the Simulator Reference Manual for full details.

SIMKIT INTERFACE

This supersedes the old Philips compact models (PCM) interface, however, the older devices are still available if needed. Versions of devices that are available through both interfaces (e.g MOS 9.03) will now default to the SIMKIT interface but the old version is still available by selecting a different level number. PCM devices do not support multi-core execution and so usually the Simkit versions will run faster.

See the Simulator Reference Manual for full details.

CMC R3 RESISTOR MODEL

The CMC R3 resistor model is incorporated in this version. It has been built from its Verilog-A description using the SIMetrix Verilog-A compiler.

See the Simulator Reference Manual for full details.