SiMKit
Release Notes

for SiMKit version 2.5

First Edition

NXP Semiconductors
DMS/Tool and Flow Solutions

Eindhoven, April 2007

This document is provided without warranty of any kind, either expressed or implied, including, but not limited to, the particular purpose. The information in this document is subject to change without notice. Whilst every care has been taken in the preparation of this document, some errors may remain. If you find an error or omission, or if you have any comments, please feel free to contact:

NXP Semiconductors
DMS/Tool and Flow Solutions

Location:	High Tech Campus 37 Room: 4-064 5656 AE EINDHOVEN The Netherlands
Post Box:	WAY 41
Telephone:	+31 40 27 41114
E-mail:	edt.helpdesk@nxp.com
Seri:	edthdesk@nlwayhp
Intranet:	http://nww.nxp.com/it/research/dms

SiMKit is a trademark of NXP Semiconductors

Table of contents

Preface

These are the release notes for *SiMKit* version 2.5. All changes with respect to *SiMKit 2.4* are reported in these release notes.

Overview

SiMKit is a simulator-independent compact transistor model library.

Simulator-specific connections are handled through so-called adapters that provide the correct interfacing to:
– *Spectre*, the circuit simulator from *Cadence*
– *Pstar*, the circuit simulator from *Philips*
– *ADS*, the circuit simulator from *Agilent*.

The *SiMKit* library contains the most recent versions of the *Philips* transistor models. The following two tables list the *SiMKit* models. The first lists the *SiMKit* models and the second lists the models for which only a *Pstar* and *Spectre* implementation are available. For a full description please check:

`http://www.nxp.com/Philips_Models/`

In the following tables **e/g** stands for electric/geometric, **t** stands for self-heating and **s** stands for substrate model.

Table 1: Real SiMKit models

Model	Level	Pstar	Spectre	ADS	e/g	t	s
juncap	1	juncap	juncap	juncap	e	no	no
juncap	200	juncap	juncap200	juncap200	e	no	no
psp	102	pspe	psp102e	psp102e	e[a]	no	no
psp	1020	psp	psp1020	psp1020	g[a]	no	no
psp	1021	psp	psp1021	psp1021	g[a]	no	no
pspnqs	102	pspnqse	pspnqs102e	pspnqs102e	e[a]	no	no
pspnqs	1020	pspnqs	pspnqs1020	pspnqs1020	g[a]	no	no
pspnqs	1021	pspnqs	pspnqs1021	pspnqs1021	g[a]	no	no
modella	500	tpl	bjt500	bjt500	e	no	no
modella	500	tplt	bjt500t	bjt500t	e	yes	no
mextram	504	tns/tps	bjt504	bjt504	e	no	yes

Table 1: Real SiMKit models

Model	Level	Pstar	Spectre	ADS	e/g	t	s
mextram	504	tnst/tpst	bjt504t	bjt504t	e	yes	yes
mextram	504	tn/tp	bjtd504	bjtd504	e	no	no
mextram	504	tnt/tpt	bjtd504t	bjtd504t	e	yes	no
mos	1100	mne/mpe	mos1100e	mos1100e	e	no	no
mos	1100	mn/mp	mos1100	mos1100	g	no	no
mos	1101	mne/mpe	mos1101e	mos1101e	e	no	no
mos	1101	mnet/mpet	mos1101et	mos1101et	e	yes	no
mos	11010	mn/mp	mos11010	mos11010	g	no	no
mos	11010	mnt/mpt	mos11010t	mos11010t	g	yes	no
mos	11011	mn/mp	mos11011	mos11011	g	no	no
mos	11011	mnt/mpt	mos11011t	mos11011t	g	yes	no
mos	1102	mne/mpe	mos1102e	mos1102e	e	no	no
mos	1102	mnet/mpet	mos1102et	mos1102et	e	yes	no
mos	11020	mn/mp	mos11020	mos11020	g	no	no
mos	11020	mnt/mpt	mos11020t	mos11020t	g	yes	no
mos	11021	mn/mp	mos11021	mos11021	g	no	no
mos	11021	mnt/mpt	mos11021t	mos11021t	g	yes	no
mos	2001	mne/mpe	mos2001e	mos2001e	e	no	no
mos	2001	mnet/mpet	mos2001et	mos2001et	e	yes	no
mos	2001	mn/mp	mos2001	mos2001	g	no	no
mos	2001	mnt/mpt	mos2001t	mos2001t	g	yes	no
mos	2002[b]	mne/mpe	mos2002e	mos2002e	e	no	no
mos	2002[b]	mnet/mpet	mos2002et	mos2002et	e	yes	no

Table 1: Real SiMKit models

Model	Level	Pstar	Spectre	ADS	e/g	t	s
mos	2002[b]	mn/mp	mos2002	mos2002	g	no	no
mos	2002[b]	mnt/mpt	mos2002t	mos2002t	g	yes	no
mos	3100	mn/mp	mos3100	mos3100	e	no	no
mos	3100	mnt/mpt	mos3100t	mos3100t	e	yes	no
mos	40	mn/mp	mos40	mos4000	e	no	no
mos	40	mnt/mpt	mos40t	mos4000t	e	yes	no

a. For PSP the electrical model is referred to as the local model and the geometrical model as global.
b. Mos 2002 is a test version in SiMKit 2.5.

Table 2: Other (older) models (Pstar and Spectre specific)

Model	Level	Pstar	Spectre
diode	500	d	dio500
mos	3002	mn/mp	mos3002
mos	902	mn/mp	mos902
mos	902	mne/mpe	-
mos	903	mn/mp	mos903
mos	903	mne/mpe	-
mextram	503	tn/tp	bjt503
mextram	503	tns/tps	bjt503
lpnp	301	tpl	bjt301
mos	705	mne/mpe	mos705

Release notes

The release notes can be obtained by entering the following command:

```
cadenv -q simkit
```

1 Improvements

Model improvements

MOS1101

The high clip value of the parameter `NU` is set to `100` to avoid numerical problems.

MOS1102

In 64bit mode a floating point exception could occur in the calculation of the initial values for the iterative process to calculate the surface potential. By re-arranging the calculations, this is now avoided.

PSP and MOS1102

Induced gate noise limiting (i.e. the cut-off of the f^2 dependency of the induced gate noise at high frequency) has been improved in *Spectre* and *ADS*.

With this implementation not all noise problems are solved (see the section titled *Known limitations on page 11*. For PAC and Pnoise calculation problems still exist; these will be solved in (one of) the next *SiMKit* releases.

MOS2001 and MOS2002

The clipping of the effective width of the transistor was not handled correctly, which ended up in an error when the effective width became less than, or equal to, 0. This is solved now by clipping the width in the correct way.

MOS2002

Self-heating has been added to this model.

PSP

In certain cases, the calculation of the surface potential in the gate overlap region was missing in *Spectre* and *ADS*. The derived gate current was therefore incorrect. The problem only applies when the switch parameter `SWIGATE` is `on`.

Mextram 504

When running a 64 bit executable of the simulator, a tremendous slowdown (up to a factor of 1000) was noticed for some examples. This was traced to the power function `pow()`, calculating x^y, with the following specific values for x and y: `y` is `0.5`, and `x` is close to `1`, in the range `[1-1e-15, 1+1e-15]`. The situation is now avoided - the `pow()` function is not something we can change - by using the `sqrt()` function when `y=0.5`.

Juncap level 1

– In the past, differences were seen in the leakage current of the Juncap for different *Spectre* versions. The leakage current in the Juncap is now correct for all *Spectre* versions.

In *Ultrasim* a large difference in results was observed by using the analog mode (`sim_mode=a`) and the *Spice* mode (`sim_mode=s`) for the Juncap device. This has been solved, but it was necessary to introduce three new OP parameters for the Juncap device, which will show up in all simulators:
`lx1`, Total current
`lx3`, Total charge
`lx5`, Total capacitance

These parameters are used by *Ultrasim* in such a way that the performance is guaranteed for the (faster) analog mode.

– In *SiMKit 2.4* there was a DC convergence problem related to RF fringe capacitance used in CMOS5065 with low temperatures (<-25deg). This convergence problem was caused by a Juncap where the conductance became zero. This problem has been solved in *SiMKit 2.5* by placing a parallel `gmin` resistance.

PSP and Juncap level 200

Several actions were taken to speed up the PSP and juncap200 model evaluation. The C-code for these models is generated from verilog-A, and the tool used to generate this code has been modified to generate more efficient code for the derivatives. Furthermore, some automatic re-grouping of code has been done to make the code more efficient.

Juncap level 200

In *Pstar* and *ADS* a specific bug could cause a memory fault when PAC or Pnoise analysis was used. This bug has been solved.

Spectre specific issues

Spectre releases using CMI5 and older SiMKit versions

SiMKit 2.5 is prepared for *Cadence* versions using *CMI5*, such as *Cadence_mmsim* version 6.1.1.292-ISR10 and higher. *SiMKit* versions previous to and including *SiMKit 2.4* will not automatically work with these *Cadence* versions, but with some minor adjustments to the installation these older *SiMKit* versions can still be used.

1. Go to the directories containing the files:
   `libsimkit_spectre_4.0.so` and **`libsmk_4.0.so`**

   For *Linux* these are:
   `simkit/2.4/lib/Linux/` and **`simkit/2.4/lib/Linux/64bit`**

2. Create the following links to the above-mentioned files:
   `libsimkit_spectre_5.0.so` and **`libsmk_5.0.so`**

Region Parameter

The region parameter for the device models in *Spectre* was not implemented consistently.

A MOS TRANSISTOR has three regions:
– Cut-off or sub-threshold mode
– Triode or linear region
– Saturation.

A BIPOLAR TRANSISTOR has four regions:
– Cut-off or sub-threshold mode
– Forward
– Reverse
– Saturation.

A DIODE has three regions:
– Forward
– Reverse
– Breakdown

For the PSP model you can use all the regions, because it contains the MOS model regions and the Juncap regions. Please note that no changes have been made to the dio500 and nod-

cap devices, since they are not 'real' *SiMKit* models. The defaults for the region parameter have not been changed.

Scalem parameter

In *SiMKit 2.4* we implemented support of the `scalem` parameter for all *SiMKit* devices. The use of `scalem` is meaningless for *SiMKit* devices and will lead to errors. Therefore, we will not support this parameter from *SiMKit 2.5* onwards.

Spectre adapter

The *Spectre* adapter has been modified for performance reasons. By storing some extra information in the *SiMKit* Interface (see software specific items), it has become possible to monitor whether initialization functions really need to be executed, or not. The speed up in time will be visible when doing DC sweeps.

Terminal currents

For *Spectre* versions based on *CMI 5* (*Spectre 6.1* and higher), we implemented terminal currents for the *SiMKit* devices. This means that for a circuit, for which the currents are saved, the number of equations calculated by *Spectre* is far less than for earlier *SiMKit* versions. Depending on the type of circuit and chosen save options, this can lead to an improvement in simulation time.

ADS specific issues

ADS designkit

Symbols for MOS2002 and for PSPNQS have been added to the designkit.

Loadmaps

SiMKit now uses **loadmaps** - a feature offered by the *MINT* interface - according to *Agilent* this should improve the performance of *SiMKit* models in *ADS*.

In the case of *ads 2005A*, we have seen problems in the convergence behavior after introducing the loadmaps. These problems can be solved by using the following options parameter in the netlist:

```
Options:Options1 DC_ConvMode=7
```

To use this parameter in the GUI, select the following in the `OPTIONS` block (available for all analyses):

Convergence -> Advanced -> Advanced DC Convergence Settings -> Mode: Hybrid solver

Gmin

In *ADS*, the `gmin` parameter defined in the simulator will now be used instead of the maximum of the model `gmin` and the simulator `gmin`. This implies that users can set a `gmin` value less than `1e-15` if required.

Pstar specific issues

Juncap level 200

In *Pstar* it was possible to give the type parameter a value between -1 and 1, whereas the definition of the model only allows the exact values -1 and 1. Clipping has been extended for this variable so that intermediate values no longer occur.

PSPNQS

In *SiMKit 2.4* the electrical `PSPNQS` model in *Pstar* was named in an inconsistent way compared to other *Pstar* device model names. In *SiMKit 2.5* we have given the model the name `pspnqse`, which is in line with other *Pstar* names.

General issues

On *Linux* the *SiMKit* is generated with different compiler flags than before. The result is that for all models the model evaluation has become 20 to 30% faster. Some minor numerical differences may also be observed. How the more efficient model evaluation works out in the total simulation time, depends on how much the model evaluations contribute to the total simulation time - e.g. if the model evaluations take 25% of the total simulation time, and the speedup of the model is 20%, the total simulation time will be reduced by 5%.

Software specific issues

– Due to an intense cleanup of the Ptar/SiMKit interface, several areas of the *SiMKit* interface have been changed.

  Instead of the types boolean, integer and real the model library interface now uses it's own types: `min_real, min_integer` and `min_boolean.`

  The *SiMKit* types `real`, `boolean`, `integer` and `unint` have been renamed to `sk_real`, `sk_boolean`, `sk_integer` and `sk_unint`.

  The new type names are used in all model files and all adapters.

  The *SiMKit* interface is runtime compatible, but at compile time, the interfaces are not compatible.

  Please check the `simkit_interface.html` document for all details.

– Starting from *SiMKit 2.5* the common adapter code is compiled into a separate library. In earlier versions the `common_adapter.c` file had to be included in the adapter code directly via an **#include** statement. As of *SiMKit 2.5* this include must be replaced by adding the common adapter code library (`libadapter_common.a`) to the link line for your own adapter.

  The source code for the common adapter library is stored in the `simkit/2.5/source/adapter_common` directory. The interface to the library is defined in the `adapter_common.h` header file.

  It is not obligatory to use the `adapter_common` library for your own *SiMKit* adapter. However, if you do not use the library, you will have to implement most of the functionality it provides yourself.

– The *SiMKit* interface version is now registered in the define `#define SK_INTERFACE_VERSION` in the file `sk.h`. The adapters at loading time use the `SK_load_models()` function in the `adapter_common` library to check whether the interface version is correct. If it is not they deliver a command line error message:

`Failed to load library 'libnxp_models.so'.`
`The adapter expected the model library to use interface version` `'<version_a>'`**`, but the model library reported version`** `'<version_b>'`**`.`**

– Since *SiMKit 2.4* the adapters and models have been divided into two libraries. The robustness of this division has been improved. When the adapter library expects more models than are actually found in the model library a warning message will be displayed:

`Model` `<modelname>` **`is not available in the model library.`**

When specific environmental variables are used to change the order of loading the libraries, the loading of the model library no longer fails.

The error message when the models library is not found has been improved:

`"Failed to load library 'libnxp_models.so'.`
`Error message: '<path_where_simkit_is looking>`
`/libnxp_models.so: cannot open shared object file: No such file or directory'.`
`Maybe you forgot to set LD_LIBRARY_PATH or SHLIB_PATH."`

– The name of the device models library has been changed from `libphilips_models` to `libnxp_models`. *SiMKit* users will generally not notice this name change, unless something goes wrong.

2 Known limitations

Known limitations

PSP1020

The geometrical model of PSP level 102 can show convergence problems, using the default values of the parameters. This can be solved by using another value for the overlap capacitance `LOV`. If this value is set to approx. `5n`, the model shows better convergence behavior. This value will be used as a default value for a new level of PSP. In order to prevent any problems, it's always better to use realistic model parameters that come from an actual process technology kit.

ADS2005A

In the case of *ads 2005A*, we have seen problems in the convergence behavior after introducing the loadmaps. These problems can be solved by using the following options parameter in the netlist:

```
Options:Options1 DC_ConvMode=7
```

To use this parameter in the GUI, select the following in the `OPTIONS` block (available for all analyses):

Convergence -> Advanced -> Advanced DC Convergence Settings -> Mode: Hybrid solver

Noise

Although problems with induced gate noise limiting have been solved for PSP and MOS1102, a few more issues related to noise have been reported: in SP-noise, pnoise and transient noise. There was not enough time to solve these problems in *SiMKit 2.5*, but we will try to solve them in the next *SiMKit* release.

Cadence_ius

Not all versions of *cadence_iuc* (needed for *AMS* designer and/or *Ultrasim*) work well with *SiMKit*; you must have at least *cadence_ius 5.7.** or higher to use the *SiMKit*.