

SiMKit *Release Notes*

for SiMKit version 2.2

First Edition

Philips
ED&T/Analogue Simulation

Eindhoven, July 2005

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:

Philips
ED&T/Analogue Simulation

Location: High Tech Campus 5
5656 AE EINDHOVEN
The Netherlands
Post Box: WAY 31
Telephone: +31 40 27 41114
E-mail: edt.helpdesk@philips.com
Seri: edthdesk@nlwayhp
Intranet: <http://pww.research.philips.com/edt>

SiMKit is a trademark of Philips Electronics N.V.

© Philips Electronics N.V. 2005

All rights reserved. Reproduction in whole or in part is prohibited without the written consent of the copyright owner.

Table of contents

Table of contents	iii
Overview	vi
Release notes	ix
1 Improvements	1
Performance	2
Performance comparison	2
MOS1101 and MOS1102	3
MOS31 and MOS40	3
Modella	3
PSP	3
JUNCAP2	3
Improvements for users of the SiMKit Interface	4
Modelbooks	4
2 Circuit simulator related issues	5
ADS specific issues	6
Spectre specific issues	6
Pstar specific issues	7
Ultrasim specific issues	8
3 Known limitations	9
Known limitations	10

Preface

These are the release notes for *SiMKit* version 2.2.

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 tables list the *SiMKit* models. The first table lists the *SiMKit* models, the second table lists the models for which only a *Pstar* and *Spectre* implementation are available.

For a full description please check:

http://www.semiconductors.philips.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 (in release 2.2)

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	100	pspe	psp100	psp100	e ^a	no	no
psp	1000	psp	psp1000	psp1000	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
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

Table 1: Real SiMKit models (in release 2.2)

Model	Level	Pstar	Spectre	ADS	e/g	t	s
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	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. Note that for the PSP-model the electrical model is referred to as the local model and the geometrical model as global.

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**

Performance

This release has mainly focussed on the optimization of PSP and JUNCAP 2.

The source code for these two models has been generated automatically from a verilogA description of the models. Part of the optimization has been done in the verilogA code, and part has been done manually in the resulting c-code.

Optimization in speed has been achieved via the following:

- initialisation code now only executed at initialisation
- in JUNCAP2 more efficient limiting functions have been implemented
- the equations that are not required for a specific analysis are no longer calculated
- reduced number of branches in PSP.

Optimization in code structure:

- the verilogA code has been restructured so that the stand-alone JUNCAP2 and the junction model in PSP are now generated from the same source code.

Optimization in the spectre adapter:

- more efficient implementation in DC for all models.

Performance comparison

Table 3: Older analyses models (Pstar and Spectre specific)

Model		SiMKit 2.1.1	SiMKit 2.2
M1101	DC	65.8	60.4
M1102	DC	81.4	72.0
PSP	DC	92.5	73.5
PSP(J)	TRAN	95.0	52.0
PSP	TRAN	68.8	42.6

PSP(J) indicates that the juncaps are switched on.

MOS1101 and MOS1102

The calculation of the power dissipation P_{diss} has been corrected in case of a source drain exchange (inverse bias operation).

MOS31 and MOS40

- The temperature scaling has been extended for the model parameters RON, VSAT and RSAT. Four new parameters have been added to the existing parameter ACH.

These new parameters are:

- ACHMOD=0,1 (default 0) switch between the current (default) scaling and the extended scaling
- ACHRON (default 0, no clipping)
- ACHVSAT (default 0, no clipping)
- ACHRSAT (default 0, no clipping)

This was requested by *ADS* users because they have no other means to influence the temperature scaling.

Modella

- The implementation of the internal diffusion voltage has been changed according to the modelbook.

PSP

The PSP implementation has been extended with the operating point output. The performance has also been improved.

The following minor issues have also been solved:

- Solved issue with MULT (Verilog-code only) (bug fix).
- Solved issue with BGIDL (ADMS issue) (bug fix).
- Added parameter FNT in noise model.
- Added shot-noise in avalanche current.
- Now ϕ_T is used instead of ϕ_{T^*} .
- Corrected value of ϕ_{X^*} (Eq. (4.22) in the documentation in the web page) (bug fix).

JUNCAP2

Efficient limiting functions have been implemented.

Improvements for users of the SiMKit Interface

The noise types in the *SiMKit* interface have been extended with the types SK_NT_THERMAL, SK_NT_SHOT and SK_NT_FLICKER. This was requested by *Aplac*.

Modelbooks

The modelbook updates related to this *SiMKit* release are planned for October.

2 **Circuit simulator related issues**

ADS specific issues

- As of *SiMKit 2.0.1* an *ADS* design kit, which makes it possible to use the *SiMKit* models in the schematic entry for *ADS*, is included in the *SiMKit* package. In *SiMKit 1.3* this design kit was a separate package.

If you have the *SiMKit_designkit 1.3* installed and you cadenv *SiMKit 2.2*, some files must be modified in order to make the design kit work. First, cadenv *SiMKit 2.2* and then change the **DESIGN_KIT_MODEL_PATH** variable in the following files:

- **de_sim.cfg** (located in the startup directory)
- **de_sim.cfg** (located in your project directory)
- **hpeesofsim.cfg** (located in your project directory)

The correct value of that variable is:

```
DESIGN_KIT_MODEL_PATH=$CADENV_HOME/.caddata/simkit/  
adsdesignkit/bin/hpux11
```

where **hpux11** can also be **linux_x86** depending on the operating system in use. Typing **hpeesofarch** in the command line gives you the operating system that you are using.

- To use *SiMKit* with *ADS*, you need at least *ADS2003C*.

Known problem

- For HB and Noise analyses, the simulation is faster using the built in models provided by *ADS* than the *SiMKit* models.

Spectre specific issues

- If you want to use *Spectre* stand-alone, the order in the cadenv of *SiMKit* and *cadence_ic* is very important. You must first cadenv *cadence_ic* and then *SiMKit*. This is because *SiMKit* sets a wrapper around the *Spectre* start-up script in the cadenv package *cadence_philips*. The version of the *cadence_philips* package should be 2004.3.2, or higher. A cadenv of *cadence_ic*, or some of the flows like AMSDE, overwrites these wrappers.

We have tested the following *cadence_ic* versions in combination with *SiMKit 2.2*:

- On *Linux*: 4.4.6.100.*, 5.0.33.500.1.11, 5.10.41.500.0.7
- On *HP* : 4.4.6.100.*, 5.0.32.500.9 , 5.10.41.500.0.7

✓ Note _____
Do not use *cadence_ic 5.0.0.500.38*. It does not work with *SiMKit* due to a small error in *Spectre*.

✓ Note _____
With *cadence_ic 5.0.33.500.0.6* we have seen problems in combination with *SiMKit 1.3, 1.3.1.1.1* and *2.0.1*. These problems do not occur with *SiMKit 2.2*.

Known problems:

- For MOS11 devices the noise parameters are zero only for *cadence_ic* versions that use CMI3.0 (*cadence_ic 5.0.33.500.0.6_ads* and *cadence_ic 5.0.33.500.0.6*). The *cadence_ic* versions with CMI 1.0 and 4.0 show the right values (*cadence_ic 5.10.41.500.0.7*).
- If you change the gender of a device with an alter statement (for example: `alter gender alter mod=MM3100 param=type value=n`), a warning is displayed informing you that the device parameters are set to their default values.
- The following *Spectre* statements are not supported by the *SiMKit*:
 - alter group statements for *SiMKit* parameters
 - scale statements.
- For Juncap1 it is currently not possible to use the `type` statement. Existing netlists will have to be converted before they will work correctly.

Pstar specific issues

- In some cases, the convergence of jobs with *Pstar 4.8* and *SiMKit 2.0.1* was slower than for the same jobs with *Pstar 4.7* and *SiMKit 1.3*. This was caused by differences in the limiting functions in Juncap level 1 and a change in *Pstar*'s strategy for finding a DC solution.
- For Juncap2 and PSP there is no `print_scaled_parameters` output.

Ultrasim specific issues

- *Ultrasim* generates a simplified model according with the simulation settings from the original model (built in model, *SiMKit* models, ...). This helps to speed up the simulation.

If *SiMKit* is cadenv-ed, *Ultrasim* is not able to simplify the *SiMKit* models, meaning that the simulation will be done with the complete model independent of the simulation settings. Therefore no speed up of the simulation is achieved.

3 **Known limitations**

Known limitations

The following known limitations are in *SiMKit 2.2*:

- For HB and Noise analyses, the simulation time is faster using the built-in models provided by *ADS* than the *SiMKit* models.
- For *cadence_ic* versions with CMI3.0, the noise parameters of MOS11 devices are zero. (The *cmi* version can be checked by entering: `spectre -cmiversion`).
- The following *Spectre* statements are not supported by the *SiMKit*:
 - alter group statements for *SiMKit* parameters
 - scale statements.
- For Juncap level 1 it is currently not possible to use the `type` statement in *Spectre*. Existing netlists will have to be converted before they will work correctly.
- Due to the parameter structure change introduced in *SiMKit 2.1.1* the output produced when the *Pstar* parameter `print_scaled_device_parameters` is set to `true` changed. This means that the set of scaled parameters that is printed may vary from *SiMKit 2.0.1* to *SiMKit 2.1.1*.

For the next release we aim to have a clear description of the set of parameters that are printed for all *SiMKit* models.