SiMKit
Release Notes

for SiMKit version 3.6

NXP Semiconductors
A&M/Tool and Flow Solutions

Eindhoven, January 2011



NXP Semiconductors
A&M/Tool and Flow Solutions

Location: High Tech Campus 46
5656 AE EINDHOVEN
The Netherlands
Post Box: WDA 2
E-mail: dm.helpdesk@nxp.com
Intranet: http://nww.wiki.nxp.com/display/rbstdsgn/SiMKit



## Preface

These are the release notes for SiMKit version 3.6. All changes with respect to SiMKit 3.5.2 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, APS, etc.: the Cadence circuit simulators
- Pstar, the NXP circuit simulator
- ADS, Agilents circuit simulator

Note that several other simulators (e.g. AFS by BDA) provide an adapter for the simkit models.

The SiMKit library contains the most recent versions of the NXP transistor
models. The following two tables list the SiMKit models. The first table lists the 'real' SiMKit models, the second table lists the pre-SiMKit models,
for which only a Pstar or Spectre implementation is available.
For a full description please check:

http://www.nxp.com/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/APS/AFS | ADS | e/g | t<br>s |
|---|---|---|---|---|---|---|
| juncap | 1 | juncap | juncap | juncap | e | no<br>no |
| juncap | 200 | juncap | juncap200 | juncap200 | e | no<br>no |
| psp | 102 | pspe | psp102e | psp102e | e* | no<br>no |
| psp | 1020 | psp | psp1020 | psp1020 | g* | no<br>no |
| psp | 1021 | psp | psp1021 | psp1021 | g* | no<br>no |
| pspnqs | 102 | pspnqse | pspnqs102e | pspnqs102e | e* | no<br>no |
| pspnqs | 1020 | pspnqs | pspnqs1020 | pspnqs1020 | g* | no<br>no |
| pspnqs | 1021 | pspnqs | pspnqs1021 | pspnqs1021 | g* | no<br>no |
| psp | 103 | psp | psp103 | psp103 | eg | no<br>no |
| pspnqs | 103 | pspnqs | pspnqs103 | pspnqs103 | eg | no<br>no |
| modella | 500 | tpl | bjt500 | bjt500 | e | no<br>no |
| modella | 500 | tplt | bjt500t | bjt500t | e | yes<br>no |
| mextram | 504 | tns/tps | bjt504 | bjt504 | e | no<br>yes |
| mextram | 504 | tnst/tpst | bjt504t | bjt504t | e | yes<br>yes |
| mextram | 504 | tn/tp | bjtd504 | bjtd504 | e | no<br>no |
| mextram | 504 | tnt/tpt | bjtd504t | bjtd504t | e | yes<br>no |

```
mextram      3500    tns/tps       bjt3500        bjt3500       e     no
yes
mextram   3500    tnst/tpst     bjt3500t       bjt3500t      e     yes
yes
mextram   3500    tn/tp         bjtd3500       bjtd3500      e     no
no
mextram   3500    tnt/tpt       bjtd3500t      bjtd3500t     e     yes
no
mos       903     mne/mpe       mos903e        mos903e       e     no
no
mos       903     mn/mp         mos9030        mos9030       g     no
no
mos       903     mnt/mpt       mos9030t       mos9030t      g     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       2002    mne/mpe       mos2002e       mos2002e      e     no
no
mos       2002    mnet/mpet     mos2002et      mos2002et     e     yes
no
mos       2002    mn/mp         mos2002        mos2002       g     no
no
mos       2002    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 | | | | | | |
| rfldmos | 600 | most | rfldmos600t | rfldmos600t | g | yes |
| no | | | | | | |

* For PSP 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 | - |
| mextram | 503 | tn/tp | btj503 |
| mextram | 503 | tns/tps | bjt503 |
| lpnp | 301 | tpl | bjt301 |
| mos | 705 | mne/mpe | mos705 |

# 1 - Improvements

## SiMKit general:

The work to make the SiMKit threadsafe (TS) has been continued in this release.

For the phased transformation of SiMKit to a TS version the model library interface has been extended with some extra data fields and with TS versions of eval_model and other functions. In this release, these extensions are backward compatible which means that the old non-TS interface can still be used for all the models. The number of device models which have implemented the TS function versions will increase in the coming releases. When all models have been transformed, the new functions will replace the non-TS versions: for example eval_model_reentrant() will then be renamed to eval_model().

The ADS and Spectre adapters have been updated so that they can handle some of the models in a threadsafe way. For Pstar nothing has changed.

SiMKit 3.6 contains two extra models that are TS: psp102 (including pspnqs102) and mextram504.

Implementing mextram504 in a thread-safe way can have some minor consequences
for the results. The changes are in the limiting of the device ev's:
- it is no longer possible to use ev-values of a previous Newton iteration,
  a feature that was only used by Pstar. This can have an influence on
  the convergence process for better or for worse. We have observed both.
- the  information about linearisations of exponentials in the diode currents
  (when converged) is less specific than before.

The following models are also TS: mos1101, juncap, mos1102, juncap200, mos903, mos3100, mos40.

For APS and ADS the models that are not yet threadsafe are labeled as such,
thus informing the simulator not to use them in multi-thread mode.
For ADS, if one of the SiMKit models used in the netlist is not threadsafe, the
simulator is forced to switch to single threaded mode.

Spectre:
--------
As a consequence of implementing SiMKit devices in a thread safe way, the behavior of the 'info' statement in Spectre has changed.

Up till SiMKit 3.5.2 'what=output' or 'what=all' in an info line would print
both the instance parameters that can be set by the user and the set of electrical parameters.

As of SiMKit 3.6 for thread safe devices the set of electrical parameters is not
part of output anymore when 'what=output' or 'what=all' is specified in an
'info' statement. The only way to get acces to the set of electrical parameters
for a thread safe device is to ask for the operating point info ('what=oppoint')
and set 'printscaled=1' at the instantiation of the device.

ADS designkit:
--------------
Only for those working with ADS GUI.

In order to avoid further confusion between SiMKit designkit names and ADS
built-in names for symbols, the SiMKit designkit symbols have been renamed
and they now all have the prefix SIMKIT_ added to their names.

For projects started with SiMKit 3.5.2 or earlier please follow
the description below that will guide you through the necessary
steps in the conversion of your project.

In the SiMKit 3.6 disribution we provide a script to help you do the
conversion. This script, convert_dsn.pl, simply replaces all occurences
of SiMKit designkit names as they were until now with the new names,
where the prefix SIMKIT_ is put in front of very name.

Go to the directory where the root of the ADS project is located
Assume the project is called <ads_project>.
Make sure you have write access to all files in the project

Find all dsn files and put them in a file called dsnfiles:

> find <ads_project> -name *.dsn > dsnfiles

Execute the conversion:

> for i in `cat dsnfiles`; do convert_dsn.pl $i; done

This will create for every dsn file <file> in the list a new file, with
name <file>_out. You can inspect the differences, e.g. with

> for i in `cat dsnfiles`; do echo $i; diff $i $i"_out" ; done

When you are satisfied with the results, move the new files to the
original names and you project conversion is done.

> for i in `cat dsnfiles`; do mv $i"_out" $i ; done

Now you can start ADS with SiMKit 3.6 on your existing project.

Model Improvements
==========

General:
---
Some mathematical functions in PSP102, PSP103, Juncap200 and
RFLDMOS600 have been protected against numerical problems.

MM1101 and MM1102:
---
The scaling rule for KO has been updated.
The scaling rule used to be
ko = kor * [1 + LEN/LE*SLKO + (LEN/LE)^2*SL2KO] * [1 + WEN/WE*SWKO]
and has now become
ko = kor * [1 + LEN/LE*SLKO + (LEN/LE)^2*SL2KO + (LEN/LE)^SL3KOEXP*SL3KO]
*
[1 + WEN/WE*SWKO]
where

SL3KO is the third coefficient of the length dependence of KO,
default value is 0.0 and
SL3KOEXP is the exponent of the third coefficient of the length
dependence
of KO, default value is 1.0

RFLDMOS600T
---
A bug in the implementation of the selfheating has been fixed.

MM903, MM1100, MM1101, MM1102, MM3100, MM40, Mextram504, juncap1:
---
A new instance parameter PRINTSCALED has been added to these models.
When PRINTSCALED is set to any value other than zero (the default) the
model
will print the scaled parameters of the model when the DC operating point
information is printed.

MM1102:
---
For Dcmatch calculations the parameter BET has been added to the OP
output.

PSP102
---
For Dcmatch calculations the parameter BETN has been added to the OP
output.

Juncap and Modella
---
When running a 64 bit executable of the simulator, a tremendous slowdown
was
noticed for some examples. As before in Mextram504 (see SiMKit 2.5) this
was
due to the power function "pow()", calculating x^y, with the following
specific values for x and y: y is 0.5, and x is approximately 1.
These situations are now avoided in Juncap and Modella by using the
sqrt()
function when y==0.5.

Known limitations
---

None for the models.
Overvoltage checks for Pstar will only work for Pstar version 6.0.1 or
higher.
Overvoltage checks are not implemented for ADS.

Flexible Topology in ADS and Spectre:
For both Spectre and ADS a model can only choose one topology at a time.
This
topology must remain fixed throughout the simulation. So, e.g. a sweep
of the

parameter RGO (gate resistor) in PSP going from zero to another value, or a
sweep over SWNQS in PSPNQS is not possible. The simulator will stop with an
appropriate message - because continuation would result in erroneous results.

Internal node names in ADS for models with a flexible topology (in SiMKit
3.5 those are PSP, PSPNQS, M1101 and M1102) might be wrong in the output.

Pstar oppar print:
As a consequence of making the SiMKit threadsafe, the use of intermediate variables for OP output in Pstar does not work anymore from SiMKit 3.6 onwards.
For example the output statements

```
oppar = OP(vds.mn_1);
print: vds.mn_1;
print: op(vds.mn_1);
print: oppar;
```

would give the same output in SiMKit versions before 3.6, but from SiMKit 3.6
onwards oppar will be zero.