# New Device Model Card Approach

## Introduction

Normally a single active device model is extracted to cover a range of device geometries and temperature. Sometimes this single scaleable model is not sufficient to describe all the changes in output characteristics over the range of geometry and temperature required. The total range of geometry and temperature is then broken up into regions and a model produced for each of these sub-sets of devices. This is the basis of a binned model and can lead to discontinuities at the bin boundaries as the model card is changed. To get round this problem ***SmartSpice*** has introduced a new functionality to allow the user to go back towards a single scaleable model card via the use of a function rather than a single parameter value. This new powerful algorithm allows the user to specify a formula linking in other device model card parameters as so giving a continous multi-dimensional function. This then gets round the problems of binning and gives the potential of a more accurate model fit to the device output characteristics.

## Setting up the Model's Parameter

***SmartSpice*** offers new algorithm for setting up a model for SOI, TFT, MOSFET, JFET, MESFET, bipolar and diode. This new feature is intended to substitute for the binning scheme and gives the user a powerfull mechanism for device modeling. The new algorithm allows the user to specify a formula for any model's parameter. The formula describes the model across a wide range of data and target model parameters and is represented by a continuos multi-dimensional function.

***Syntax***:

```
m0 B0_x10 adr7 B0_x11 vss nenh L=5.1e-06 W=2.11e-06
.model nenh NMOS
 ...
+VERSION=3.1 +Vth0='0.9302 + 0.125E-012/(W*L) +0.001*L-0.012*W'
...
```

In the example above the model card parameter Vth0 is driven by width and length which comes from a set of instance parameters. For device m0 ***SmartSpice*** will use a model with parameter Vth0 calculated using L=5.1e-06 and W=2.11e-06.

## Building up the Model

An abstract model nenh is used to generate either a table of Vth0 values or create a copy of absract model values. ***SmartSpice*** uses both algorithms. Table generation algorithm is used by default and is recommended. There is a variable "modelalg" which must be specified in the initialization file(.SmartSpice.in(Unix) and smspice.set(Windows) ) to switch the internal algorithm. If modelalg=0(default) then ***SmartSpice*** builds up the table with the parameter's values. If modelalg=1 then ***SmartSpice*** creates a separate model for each instance.

## Perfomance in the New Model Approach

modelalg=0 is a memory conserving approach and has a higher speed during the parser phase. ***SmartSpice*** bypasses the regular model creation procedures and builds up an optimized (quick selection) table. During the simulation phase ***SmartSpice*** operates with the table to set up a correct model parameter value. The overhead for modelalg=0 is near 3%-6% of total simulation time. The post-processing phase takes less time. Use ***SmartSpice*** shell command "cmcstat" to get the absolute overhead during the simulation.

modelalg=1 is a more memory consuming approach. If the netlist contains a lot of different geometries (a lot of groups) ***SmartSpice*** will allocate memory for each group. The parser phase can take 4-5 times longer in comparision with the modelalg=0.

During simulation modelalg=1 is more preferable then modelalg=0 because ***SmartSpice*** does not need to make a selection of the model's parameter from table. Post processing takes more time.

## *SmartSpice* shell Command for Checking up the Model's Parameter table

All devices are grouped with respect to the arguments of the formula used for the model parameter. To see how many different groups are created and the model parameter target value ***SmartSpice*** offers the shell command "cmcstat". The output after the use of "cmcstat" is shown below with modelalg=0:

```
--> cmcstat

****internal information about CMC format model use

****

Total Overhead During Simulation = 0.000000

Model nenh

vth0=0.9302 + 0.125e-012/(w*l) +0.001*l-0.012*w

group 0 W=1.050000e-05 L=1.440000e-06 9.384671e-01
```

group 1 W=1.050000e-05 L=1.450000e-06 9.384101e-01

group 2 W=1.050000e-05 L=1.460000e-06 9.383538e-01

group 3 W=2.100000e-06 L=1.100000e-06 9.843125e-01

group 4 W=2.110000e-06 L=2.100000e-06 9.584103e-01

group 5 W=2.120000e-06 L=1.100000e-06 9.838020e-01

group 6 W=2.130000e-06 L=1.100000e-06 9.835504e-01

group 7 W=2.140000e-06 L=1.100000e-06 9.833011e-01

group 8 W=2.150000e-06 L=1.100000e-06 9.830541e-01

group 9 W=2.160000e-06 L=1.100000e-06 9.828094e-01

Output after the use of "cmcstat" is shown below when modelalg=1:

--> cmcstat

****internal information about CMC format model use****

[holder model]=nenh

group 0 W=1.050000e-05 L=1.440000e-06

group 1 W=1.050000e-05 L=1.450000e-06

group 2 W=1.050000e-05 L=1.460000e-06

group 3 W=2.100000e-06 L=1.100000e-06

group 4 W=2.110000e-06 L=2.100000e-06

group 5 W=2.120000e-06 L=1.100000e-06

group 6 W=2.130000e-06 L=1.100000e-06

group 7 W=2.140000e-06 L=1.100000e-06

group 8 W=2.150000e-06 L=1.100000e-06

group 9 W=2.160000e-06 L=1.100000e-06

## User's Oriented Errors and Warning Messages

### Conflict of Names in the .PARAM and Formula Arguments

New model approach algorithm reports about errors. In the case of using the same names for parameter in .param statement and arguments in formula model's parameter (Berckely approach) ***SmartSpice*** will issue the warning.

**Example:**

**.param w=1**

.model nenh NMOS

**+Vth0='0.9302 + 0.125E-012/(w*l) +0.001*l-0.012*w'**

---

Warning:

New model format in use : member [ w ] in the expression [ 0.9302 + 0.125e-012/(w*l) +0.001*l-0.012*w ] of model [nenh](file ./bsim3v3.mod

/home/user/MYINDECK/CMC/bsim3v3.mod) is treated as device geometry and overrides the value of parameter [w] in the .PARAM

---

User must check either formula in the .model card or parameter in the .PARAM.

### Arguments in the Formula for Model Card Parameter do not Correspond to the Set of Device's Parameters

**Example:**

q1 cq bq gnd1 gnd QNLREF m=1 areac =45

.modelqnlrefnpn (

...

+ BF='0.5+area+1'

Fatal Error: CMC model: Model qnlref can not find instance parameter 'area' in the specified device's line

q1 cq bq gnd1 gnd qnlref m=1 areac =45

used in the expression '0.5+area+1'

Error on lines:

29 : .modelqnlrefnpn (

30 : *+BF=100 31 : +BF='0.5+area+1'

...

cannot evaluate expression '0.5+area+1'

### SmartSpice's System Internal Messages

***SmartSpice*** prints the error message "Device type "CAP32" is not supported in the enchanced modeling algorithm" if the parse finds inconsistencies in the parser phase. In this case user must check the .modelcard for device CAP32.

System message "CMC model: module found fatal error, parameter table and corresponding value table are not syncronized" is issued by ***SmartSpice*** if internal engine fails to find correspondences in the internal tables. User must report that message to the ***SmartSpice*** support.

## Multithreading Support in the New Model Approach

modelalg=0 and modelalg=1 support multithreading in ***SmartSpice.*** The user does not need to specify any extra input for ***SmartSpice,*** only -P n, where N-is a number of CPUs.

## Enchancements of the New Modeling Approach

The set of allowed instance's parameters which can be used in the formula for a model card parameter is the subject for increase later. The core of the algorithm is scalable.

| Typegroup | Technology | Internal name | Info | Level | the list of supported parameters |
|---|---|---|---|---|---|
| npn<br>pnp<br>lpnp | BJT | BJT | Bipolar Junction Transistor | 1, 2 | area, areb, areac |
| | | VBIC | VBIC Bipolar Junction Transistor | 5 | area, areab, areac |
| | | HICUM | HICUM Bipolar Junction Transistor | 6 | area, areab, areac |
| | | PBJT | Mextram BJT (Philips) | 503, 504 | area, areab, areac |
| | | MODELLA | Philips TPL500 Bipolar Transistor | 500 | |
| | | HBT | Hetero-Junction Bipolar Transistor | 20 | area, areab, areac |
| nmos<br>pmos<br>ntft<br>ptft | SOI | BSIM31SOI | Berkeley SOI MOSFET model version 1 (level 25) | 25 | w, l |
| | | BSIM3SOI2DD | Berkeley SOI MOSFET model version 2 (level 27) | 27 | w, l |
| | | BSIM3SOI2FD | Berkeley SOI MOSFET model version 2 (level 26) | 26 | w, l |
| | | BSIM3SOI2PD | Berkeley SOI MOSFET model version 2 (level 29) | 29 | w, l |
| | | BSIM3SOI3 | Berkeley SOI MOSFET model version 3 (level 33) | 33 | w, l |
| | | UFS | University of Florida SOI Model (level 21) | 21 | w, l |
| | | LETISOI | CEA/LETI SOI MOSFET model | 32 | w, l |
| | TFT | TFT | MOS field-effect transistor | 15 | w, l |
| | | PTFT | PolySi TFT model | 16 | w, l, area |
| | | MOS15 | MOS15 TFT Model | 35 | w, l |
| | | MOS16 | RPI Poly-Si TFT Model | 36 | w, l |
| | MOSFET | MOS123 | MOS field-effect transistor | 1, 2, 3 | w, l |
| | | BSIM1 | Berkeley Short Channel IGFET Model | 4, 13 | w, l |
| | | BSIM3 | Berkeley Short Channel IGFET Model ) Version-3 (level 81 | 81 | w, l |
| | | BSIM3v3 | Berkeley Short Channel IGFET Model Version-3 (level 8, 49, 53) | 8, 49, 53 | w, l |
| | | BSIM3M | Modified Berkeley Short Channel IGFET Model Version 3 (level 7,10,47) | 7, 10, 47 | w, l |
| | | BSIM4S | Berkeley Short Channel IGFET Model-4 (level 14, 54) | 14, 54 | w, l |
| | | MOS11 | Philips MOS11 model | 11,63 | w, l |
| | | MOS31 | MOS31 MOSFET Model | 30, 31, 40 | w, l |
| | | MOS20 | Philips MOS20 LDMOS model | 20 | w, l |
| | | EKV | EKV MOSFET Model | 44 | w, l |
| | | BSIM3H | High-Voltage MOSFET Model (level 88) | 88 | w, l |
| | | HISIM | Hiroshima University STARC IGFET Model (level 111) | 111 | w, l |
| njf<br>pjf<br>nmf<br>pmf | JFET/ MESFET | JFET | Junction/Schottky contact field-effect transistor | 1, 2, 3, 4, 5, 6 | w, l, area |
| d | Diode | DIO | Junction Diode | 1, 3 | w, l, area |
| | | DIO2 | Fowler-Nordheim Diode | 2 | w, l , area |
| | | DIO500 | Diode Level 500 | 500 | |
| | | JCAP | Junction Capacitor | 9 | area |
| | | LAS1 | VCSEL model | 4 | area |
| c<br>cap | Capacitance | FCAP | Ramtron Ferroelectric Capacitance Model | 5 | |
| | | FRMC | Ramtron Ferroelectric Capacitance Model | 6 | |

Table 1. Correspondence between different types of devices and device's parameters which can be used in the model's parameter.