

# Simulation Standard

TCAD Driven CAD

A Journal for Circuit Simulation and SPICE Modeling Engineers

## PHILIPS Model 9 New MM9 Extraction Routine in UTMOST III

Part II

### Introduction

This is the continuation of the same article printed in the previous issue of the *Simulation Standard* (October, 1998).

Philips Level 9 model is widely used as an accurate deep submicron CMOS model. The model is physical, and as such lends itself to accurate extraction methodologies. This paper describes, step by step, an extraction methodology that was developed in collaboration with an industrial partner, and has been validated for its accuracy

**Strategy #8:** Subthreshold region  $\hat{I}$  subth\_highVB\_mm9\_min

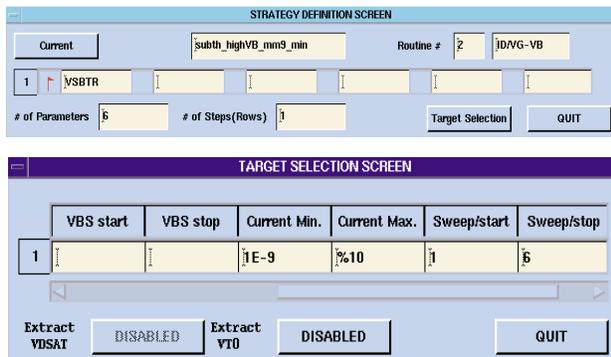


Figure 11. subth\_highVB\_mm9\_min Local Optimization Definition.

This strategy must be used at the end of the local optimization sequence.

The optimized parameter is: VSBTR. This parameter will not be optimized on a long and wide device.

We work on ISUB/VGS for different VDS and at VBS=0V.

**Strategy #9:** Substrate current  $\hat{I}$  sub\_mm9\_min.

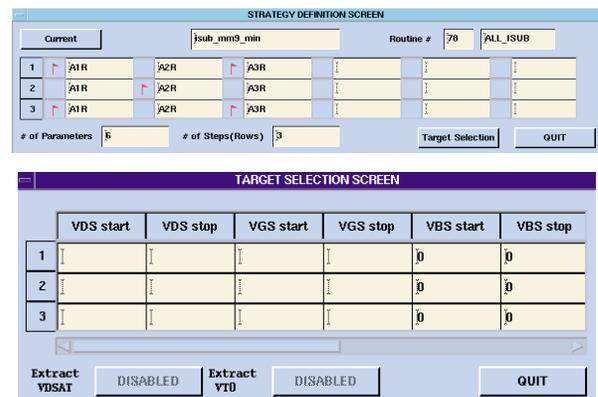


Figure 12. sub\_mm9\_min Local Optimization Definition.

The optimized parameters are: A1R, A2R, A3R. Take care that A1R and A2R are strongly correlated. So, this strategy may need to be modified for each device.

### The Optimization Sequence

An example of optimization sequence is given on the next figures. The order here is important. The first strategy should be #1 or #2, depending on body effect observed on our devices. But only one of these should be present

Continued on page 2...

### INSIDE

New Built-in Dichotomic Search Algorithm in Smartspice .....	5
Enhanced Statistical Features in SPAYN Version 1.7.2 .....	7
New Parameters for TFT model .....	9
Calendar of Events .....	10
Hints, Tips, and Solutions .....	11

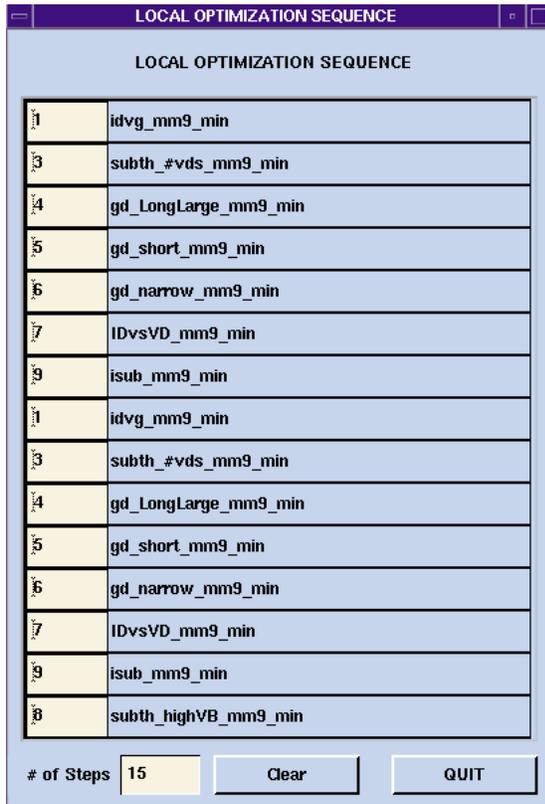


Figure 13. Local Optimization Sequence recommended

in the sequence. Then #3 should be the second strategy. #4, #5, #6 must follow in the sequence. **UTMOST III** will recognize itself the type of device it is displaying, and so which strategy to apply. Only one will be executed for a given device. Then #7 should follow. #9 is optional, but is strongly recommended. #8 must be the last strategy of this sequence.

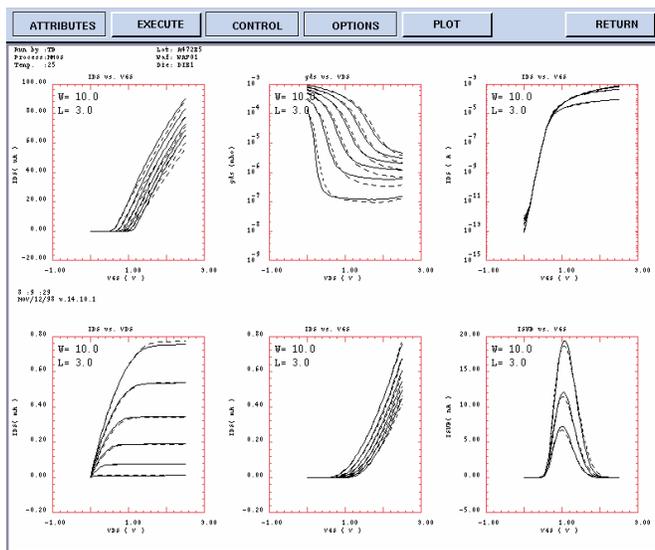


Figure 14. Graphic Screen after a Miniset Extraction on a short device.

At the end of each miniset extraction, the parameters are stored in the Optimized column and also in a library file whose name is specified in the Spice Library File. If that library already exists, **UTMOST III** will append the minisets so extracted to the library. The default library file name, if not set by the user, is "SetupFileName.l". The graphic screen shows you the result of the optimization as you can see on the next figure.

You can also extract the miniset on only one device. You have to know that among the strategies for the gds optimization, #6 for narrow device will be executed whatever can be the type of the device displayed. In that case, you can easily adapt this strategy to obtain a good miniset.

### Maxiset Extraction

Each parameter of the MOS Level 9 model can be expressed as:

$$Param = ParamR + SLParam \times \left( \frac{1}{L_{eff}} - \frac{1}{LER} \right) + SWParam \times \left( \frac{1}{W_{eff}} - \frac{1}{WER} \right)$$

Where LER and WER are respectively the reference effective length and width. The whole set of parameters (parameter for reference transistor and scaling parameters) constitute the maxiset. This equation may be slightly different for some parameters. But in any case, a simple regression will be sufficient to calculate all the scaling parameters.

FIT. VARIABLES		
FITTING VARIABLES		
1	Meas. \$extrac	1
2	Extfromlibfi	0
3	Lref	5E-6
4	Wref	1E-5
5	IR	25
6	PHIBR	0.65
7		0
8		0
9		0
10		0
11		0
12		0

QUIT

Figure 15. Fitting Variables Table

## Fitting Variables

Once the miniset extraction is completed, the maxiset procedure is called. For that matter, we need to fix some fitting variables. These variables are listed below.

1	Meas.&Extract	0 for measurement only; 1 for miniset extraction
2	Extrfromlibfi	1 for maxiset extraction from the library file ; 0 false
3	lref	Reference length for maxiset
4	wref	Reference width for maxiset
5	TR	Reference Temperature
6	PHIBR	Reference fb. Default: 0.65V

## Maxiset Regressions

The first step of the maxiset extraction is to get the length and width reductions from the linear regression of  $1/b$  against the drawn length and  $b$  against the drawn width.  $\beta$  is computed from the minisets using:

$$\beta = BETSQ \times \frac{W_{drawn}}{L_{drawn}}$$

$\Delta L$  and  $\Delta W$  are defined as the intersection of the linear regression with the X axe. From  $D\Delta$  and  $D\Delta$ , we can find LAP and WOT:

$$LER = lref - \Delta L \quad WER = wref - \Delta W$$

LVAR and WVAR represent the difference between the actual and programmed gate length and width. The data come from technologists. Then, LER and WER are calculated as follows:

Once we have determined these geometrical parameters, we can extract the scaling parameters, using simple linear regressions. In the figure below, you have an example of linear regression plot. The parameters finally extracted are: LAP, WOT, BTESQ, VTOR, SLVTO, SL2VTO, SWVTO, KOR, SLKO, SWKO, THE2R, SLTHE2, SWTHE2, THE3R, SLTHE3, SWTHE3, GAM1R, SLGAM1, SWGAM1, ALPR, SLALP, SWALP, VPR, GAMOOR, SLGAMOO, MOR, SLMO, ZET1R, SLZET1, A1R, SLA1, SWA1, A2R, SLA2, SWA2, A3R, SLA3, SWA3. If two body effects are present, we also extract KR, SLK, SWK, VSBXR, SLVBSX, SWVBSX. If extracted during miniset, we will also calculate VSBTR and SLVSBTR. The value of ETAZET is automatically set to 0.5 for NMOS, and 1 for PMOS devices.

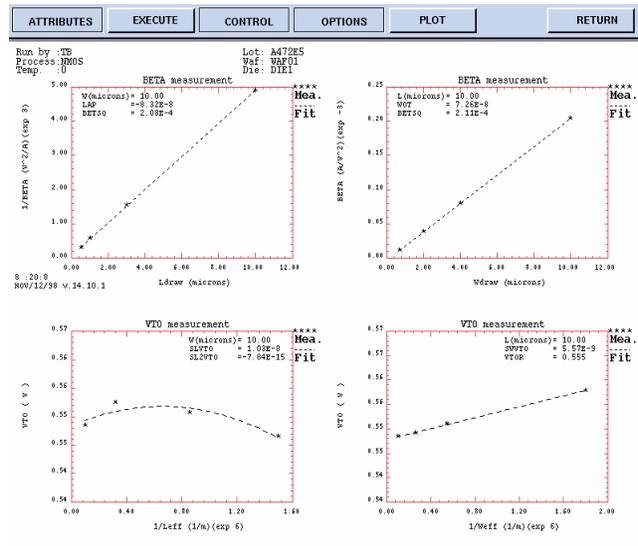


Figure 16. Maxiset Linear Regression Example: LAP, WOT, BETSQ, VTOR, SLVTO, SL2VTO and SWVTO are extracted.

## Maxiset Extraction from Library File

To perform the extraction from the library file generated during the miniset extraction, we must set the Fitting variable #2 to 1. The library file name must be specified correctly in the Spice Library File screen. This file should contain minisets of device with same type (NMOS or PMOS) and from same die, wafer, lot. It is also crucial to have a log file loaded. Else the routine will abort.

It is possible to cancel bad miniset parameters, or removed some doubtful devices from the library, before performing the maxiset extraction.

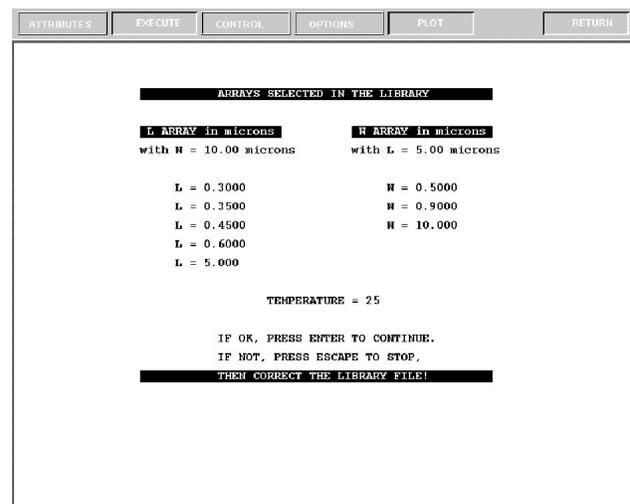


Figure 17. Maxiset Control Screen.

Before beginning the maxiset extraction, a screen displays different information found in the library file.

After the user press ENTER, the routine will proceed as usually reading miniset parameters from the library file, and displaying all regression plots.

### Temperature Extraction

Measurement at different temperature can be performed with this routine. It is clear that we do not have to optimize all the miniset parameters for each temperature, but only those assumed to vary with the temperature. Others would keep their referenced temperature values. But it is easier to extract all miniset parameters for each temperature. During the optimization TR will be automatically set to the current temperature in order not to enable MM9 temperature parameters which are still unknown.

### Extraction from Minisets

If only one device is selected, a miniset optimization is performed at every temperature. When the temperature sequence is finished, MM9 displays the seven parameters (VTOR, BETSQ, THE1R, THE2R, THE3R, MOR, A1R) which vary with temperature. TR Fitting Variable is used to compute  $\sigma$ . If not specified, the default value is 27. If there are no measurement available at 'TR', a default value for b(TR) is taken (1.E-4 for NMOS, 5.E-5 for PMOS). Temperature parameter values are reported in the Parameter Screen with the miniset parameters corresponding to the reference temperature 'TR'. If

there are no measurement at 'TR', parameters are those of the last selected temperature.

The same feature is possible from a library file. On the next figure, you have an example of regression performed during temperature parameters extraction. At the end of these extractions, parameters STVTO, ETABET, STTHE1R, STTHE2R, STTHE3R, STMO, and STA1 are stored.

### Extraction from Maxiset

If several geometries are measured at different temperatures, a maxiset determination is performed at each temperature. Then, the seven temperature parameters mentioned above are extracted. In addition to these, STLTHE1R, STLTHE2R and STLTHE3R are extracted respectively from SLTHE1R, SLTHE2R and SLTHE3R versus temperature plots.

### Acknowledgments

Silvaco would like to thank STMicroelectronics Central R&D at Crolles for his invaluable help and collaboration during the implementation of this new routine. This collaboration allowed a good development, test and validation of this new MM9 routine.

### References

- [1] R.M.D.A. Velghe, D.B.M. Klaassen, F.M. Klaassen " MOS Model 9 Level 902", Unclassified Report NL-UR 003/94, June 1995.

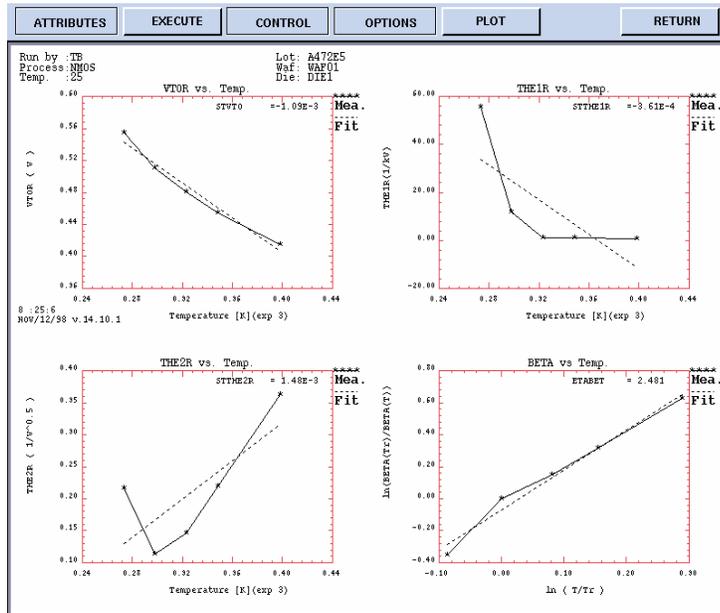


Figure 18. Temperature Maxiset Linear Regression Example: STVTO, ETABET, STTHE1R STTHE2R are extracted.

# New Built-in Dichotomic Search Algorithm in Smartspice

## Introduction

General purpose dichotomic search method is often requested by designers when they want to find a specific element (voltage, current, parameter value ...) in an already sorted set. For example, a typical problem that makes extensive use of this kind of algorithm is the computation of setup and hold time for cell characterization. This problem was already discussed in two articles [1] and [2] and a solution was proposed with the use of the Smartspice Script Language and .MODIF statement.

Continuing on the same kind of example, this article describes a new approach now available in Smartspice which offers significant improvement in terms of speed. The syntax used in the input deck is very close to the one of the optimizer. Therefore the complexity of the parser was absolutely not increased. Of course the powerful .MEASURE statement is also used in conjunction for the definition of the criterions that will lead the dichotomic search.

## Example of Dichotomic Search for Latch Setup Time Computation

As already defined in [1], the setup time of a latch is the minimum time, prior to the clock edge, that the input must remain stable to ensure reliable device operation.

If we know the exact time of the clock edge, it is possible to find the setup time by varying the input edge time in a user defined interval. This interval has to be chosen properly : that means that with its minimum value the latch must operate properly (PASS), with its maximum value the latch can't behave properly (FAIL).

To determine whether the test Fails or Passes, one or several .MEASURE statements have to be specified by the user.

Knowing the interval and the .MEASURE criterion the dichotomic search can begin. The initial guess for the parameter is either given by the user or calculated as  $MIN + (MAX - MIN)/2$ . This is the starting point for the dichotomic algorithm. Depending on the result of the first test, PASS or FAIL, then a new interval is calculated with the initial guess becoming the MIN or MAX respectively. Then the second guess is again  $MIN + (MAX - MIN)/2$  and so on... Thus at each step the time interval is divided by 2.

The last parameter that the user needs to specify is the accuracy requested for the result. The algorithm will stop when the interval becomes smaller than the accuracy :

$$(MAX - MIN) / 2^n < ACCURACY$$

In the example below the behavior of the latch is approximated with an A-device. The output V(latch) is equivalent to V(inp) if the parameter is  $> 100e-12$ . This value of 100p is the solution the method should find with the required accuracy :

```
**** DICHOTOMIC ALGORITHM EXAMPLE ****
.PARAM delaytime = 1e-12
.PARAM otherparam = 1
Vin inp 0 PWL 0 5 delaytime 5 `delaytime +
4e-12' 0
Rin inp 0 10K
A1 latch 0 V = if delaytime > 100e-12 then
V(inp) else 0
R1 latch 0 10K

.TRAN 20e-12 3e-8
.MEASURE TRAN result CROSS v(latch) VAL=3

.MODIF
+ OPTIMIZE
+ delaytime = OPT ( 0 5000p 150p )
+ otherparam = OPT ( 1 2 1.5 )          $
parameter not used -> Warning generated
+ TARGETS result = 2
+ OPTIONS passfail = -1e-12

.END
```

All the important parameters are specified in the .MODIF statement with a syntax close to the optimizer one. The difference comes from the OPTIONS parameter : "PASSFAIL". If this option is not specified the normal optimizer is called, if it is specified, the new dichotomic search algorithm is called.

## PARAMETERS :

In this example two parameters are specified : DELAYTIME and OTHERPARAM, but only the first one, DELAYTIME, will be used. The two parameters will be parsed without any problem, but at the run time, a warning is generated explaining that only one parameter is used in this method.

The specification of delaytime requires two or three parameters :

- 1-either Opt ( MIN MAX INIT ) like in this example
- 2-or Opt ( MIN MAX ) and then INIT is then calculated as  $MIN + (MAX - MIN) / 2$

## TARGETS :

Parameters specified as targets must be the result of an existing .MEASURE statement. If, after a simulation, the measurement can't be done, an error message appears and the test is considered FAILED. If the measurement can be achieved, whatever the value found, the test is considered PASSED.

The option "OFF" on the .MEASURE line is not supported with this algorithm. (Option "PROFF" is still available on the modif line and has the same effect).

A non zero value has to be specified for the targets ( here RESULT = 2 ). This value is absolutely never used, but this is kept unchanged to avoid complicating the parser and to give flexibility for future improvements.

Several targets can be specified, and the test will be considered FAILED if all the targets fail, and will be considered PASSED if at least one target passes.

## OPTIONS :

As mentioned above the only important option is PASS-FAIL which makes the difference between traditional optimizer and new dichotomic search method. But, very important is the absolute value of this option ( in this case  $1e-12$  ), which represent the accuracy requested for the final result.

The sign of the PASSFAIL parameter is also very important, it makes the difference between two opposite situations PASS-FAIL or FAIL-PASS. The sign needs to be chosen like the following:

- 1- PASSFAIL > 0 means : for parameter = MIN -> test PASS and for parameter = MAX -> test FAIL
- 2- PASSFAIL < 0 means : for parameter = MIN -> test FAIL and for parameter = MAX -> test PASS

## End of Search

The binary search process stops when the search interval becomes smaller than the accuracy, then the final value is displayed.

Figure 1 shows the 6 first steps of the search. The final value is surrounded step by step more and more accurately.

If, during the process, the tests FAILED ( or PASSED ) at every steps, then a particular warning is generated to inform the user. In fact a situation like that occurs if the sign of the PASSFAIL options is not properly chosen, if the searched value is outside the MIN and MAX boundaries or if there is a problem with the .MEASURE statement. Nevertheless, if the value searched is very close to

the MIN or MAX of the interval, the final value may also be correct. Anyway, even in this case, in addition to the warning, the final result is displayed.

## Conclusion

This article has explained the use of the new SmartSpice dichotomic algorithm through one typical example of latch characterization.

In addition to speed improvement, it is important to mention that all traditional .MODIF general options ( prtbl, proff ... ) are available with the method. All the flexibility of the .MODIF statement is retained. For example, it is still possible to define multiple sets of parameter variations.

The example of Latch Setup time is representative of the use of this feature but of course this new method is general purpose and can be used for many other applications.

## References

- [1] "Cell Characterization using the .MODIF Statement in SmartSpice", Simulation Standard, Vol. 9, No. 7, pp 12-13, January 1998.
- [2] "Advanced Cell Characterization using SmartSpice Scripting Features", Simulation Standard, Vol.9, No. 4, pp 6-7, April 1998.

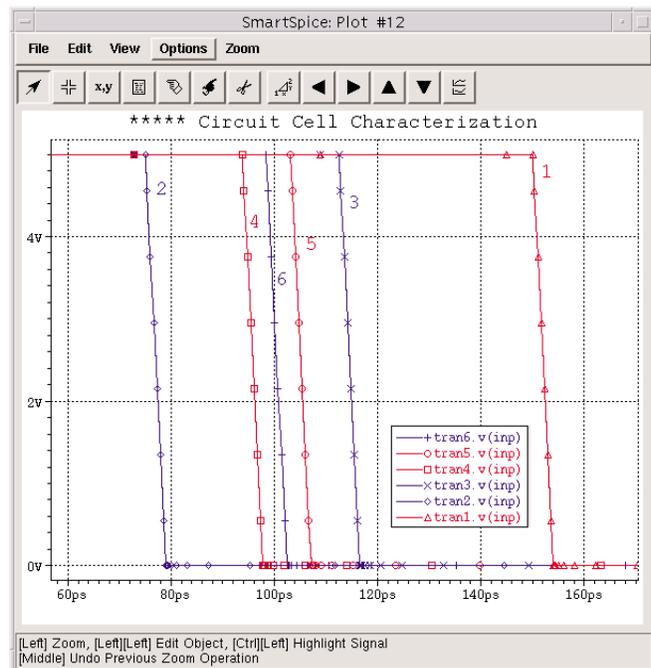


Figure 1. Six first steps of the dichotomic search

# Enhanced Statistical Features in SPAYN Version 1.7.2

## Introduction

When performing statistical analyses it is often necessary to compute transformations of the observed data. This transformation may only involve a single parameter, or it may be a complicated function of several different variables. *SPAYN* has a new feature for generating user-defined parameters as functions of existing variables and/or constants.

A new Factor plot option allows the user to plot the percentage of explained variation versus the number of factors, for a given Principal Components or Factor Analysis. It is extremely useful to have a graphical representation of the contribution each factor makes to explaining the variance in the original data. In particular when deciding how many independent factors to utilize in further calculations.

The *SPAYN* simulation interface currently generates worst-case and Monte Carlo models that can be simulated directly with an external simulator such as *SmartSpice*. The new *SPAYN* /SABER interface generates these models utilizing SABER simulator syntax. Thus providing the flexibility of several different external simulator options.

## Variable Transformation

It is frequently the case when analyzing data that some form of variable transformation is required prior to performing any calculations. Alternatively an modeling engineer may be interested in a particular parameter that is extremely difficult to measure, but can be computed in terms of easily measurable variables. *SPAYN* now has the capability to generate new parameters as functions of existing variables and/or user-defined constants. Once the new parameter has been computed it can be saved as a standard *SPAYN* variable and treated as such for further calculation or analysis.

The Variable Transformation function is available in the Parameter Matching Window, which is accessed by clicking on Analysis->Matching... from the main *SPAYN* window. Parameters P1 and P2 can then be chosen using the two pulldown menus and the scrolled Matching Quantity Window is utilized to select the required transformation function. The transformation may also only involve P1 and a constant. The Generate button will then compute the new variable, and on successful completion a histogram will appear in the graphics window. Pressing the store button allows the new parameter to be saved as a standard *SPAYN* variable. The following example demonstrates the full capabilities of this feature.

## Example

Consider constructing the new variable:

- Select  $P1 = PHI$ , toggle the Parameter Matching quantity  $\sqrt{P1}$ , press the generate button and store the new variable as  $TMP1$ .
- Select  $P1 = VBS$  and  $P2 = TMP1$ , toggle the Parameter Matching quantity  $P1/P2$ , press the generate button and store the new variable as  $TMP2$ .
- Select  $P1 = TMP2$  and Constant = 5.24, toggle the Parameter Matching quantity  $P1+P2$ , press the generate button and store the new variable as  $TMP3$ .
- Select  $P1 = GAMMA$  and  $P2 = TMP3$ , toggle the Parameter Matching quantity  $P1*P2$ , press the generate button and store the new variable as  $TMP4$ .
- Select  $P1 = VBI$  and  $P2 = TMP4$ , toggle the Parameter Matching quantity  $P1+P2$ , press the generate button and store the new variable as  $VTH$  as required.

Figure 1 shows the Parameter Matching window with the selected variables  $UO$  and  $RD$ , and transformation function  $P1 + P2$ . In this case the new parameter is  $NEW\_PARAM = UO + RD$ .

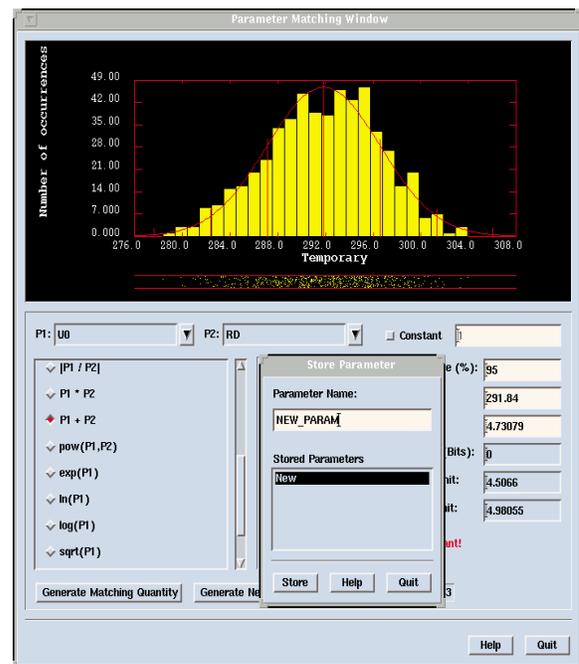


Figure 1. Parameter matching window showing the variable transformation function and a histogram of the new variables.

## Factor Plot Function

When performing a Principal Components Analysis one of the questions that confronts a modeling engineer is how many factors (independent uncorrelated variables) to select for further calculations. The idea is to select a small number of factors so as to capture as much of the variation in the original data as possible. It is therefore useful to have a graphical representation of the contribution that each factor makes towards explaining the variation.

The new factor plot function allows the user to plot the percentage of explained variation against the number of factors, for a particular Principal Components or Factor Analysis. A graphical representation shows at-a-glance what percentage of variation in the original data is explained by individual factors. The user can then select those factors which provide a significant contribution to explaining the variation.

The Plot Factors button is accessed by selecting Groups/Equations→PCA/PFA→Loadings/Coefficients from the main *SPAYN* menu. Figure 2 shows an example of a factor plot using 10 factors. It is clear from the plot that only the first 3 factors provide a significant contribution to explaining the variation in the data. In fact in this case the first 3 factors explain about 90% of the variation.

Note: When generating plots for different numbers of factors the Rotation Type toggle should be set to None. This is so plots generated for different numbers of factors using the same measured variables can be directly compared.

## SPAYN/SABER Interface

One of the features of *SPAYN* is its ability to perform statistical worst-case and Monte Carlo yield modeling. This is extremely important for the IC manufacturing process as it allows circuit performance variation to be monitored and controlled. It also provides an insight

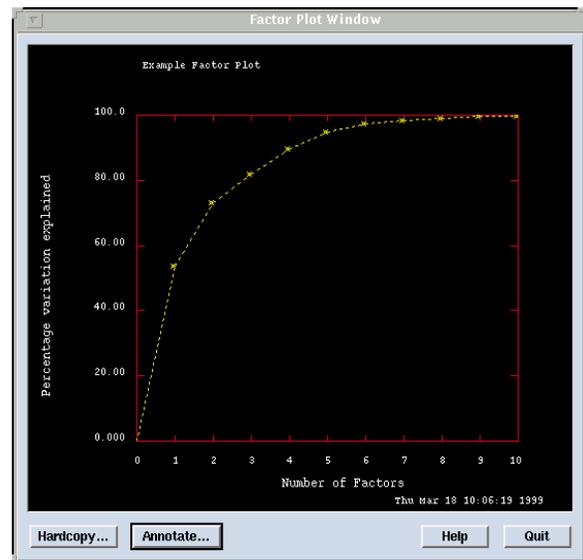


Figure 2. Factor plot with the percentage variation explained by the first 10 factors of a Principal Component Analysis.

into those model parameters that have the greatest effect on circuit performance. Once *SPAYN* has generated a worst-case or Monte Carlo model an external simulator such as *SmartSpice* can be used to perform the simulations. The new *SPAYN* /SABER interface allows the user to generate worst-case or Monte Carlo models utilizing SABER syntax.

The SABER Control window is accessed by pressing the SABER Control button in the *SPAYN/SmartSpice* Simulation Interface window, which is located under the Simulation menu of the main *SPAYN* window. The SABER Control window facilitates the generation of a SABER netlist (.sin) file from within *SPAYN*, for a particular worst-case model or Monte Carlo simulation. After generating a particular DOE or Monte Carlo model the four text fields in the SABER Control window are completed as follows:

- Model name: Enter a name for this particular set of parameters e.g. `n_parameters_test`.
- Model level: Enter the model level using SABER syntax, including any extension for that particular model. For example a *BSIM3v3* (level 8) model would be entered as `mbsim3v3.dev`.
- Type: Enter the device type e.g. `_p` for pmos or `_n` for nmos.
- Destination: Enter the destination directory where the file .sin will be placed.

Click on the Apply button to generate the SABER netlist file `n_parameters_test.sin`. The file will consist of all the simulation points in the scrolled window of the Simulation Interface window converted into SABER syntax. Figure 3 shows a demonstration of the *SPAYN* Simulation Interface window with the *SPAYN* /SABER control window inset.

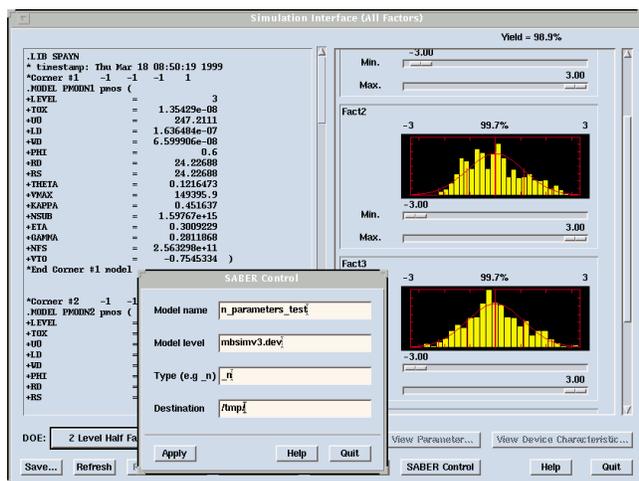


Figure 3. *SPAYN* simulation interface with the SABER control window inset.

# New Parameters for TFT model: Amorphous (Level=35) and Poly-Silicon (Level=36) TFT

## Introduction

Several enhancements have been made to the TFT models in SmartSpice. New parameters have been added to improve accuracy and allow similar sophisticated modeling options to MOS.

## Noise

The noise description in this model is based on that in BSIM3. Three new model parameters have been added to this model.

Variable	Description	Values	Default
nlev	Noise model selector	0, 1, 2	0
kf, fnccoef	Flicker noise coefficient	$10^{25}$ to $10^{-19}$	0
af, fnexp	Flicker noise exponent	1	

## Geometry

The effective channel length and width calculations have been implemented. The new model parameters are:

Variable	Default	Variable	Default	Variable	Default
lmlt	1	ld	0.75 xj	xw	0
wmlt	1	del	0	wd	0
xl	0	scaln	1	xj	0.15E-6

## ACM

This part is used to calculate effective drain/source resistance. The ACM parameter is used to select one of three available models. The new parameters are :

Variable	Default	Variable	Default	Variable	Default
acm	0	rdc	0	geo	0
m, mult	1, not effective	hdif	0	meto	0
rsc	0	ldif	0		

## Expert Debug Options

To help resolve the convergence problem, expert options similar to those from MOS Level 9 have been implemented. The syntax is:

```
.OPTION EXPERT=nnn
```

where n is a number from 0 to 7 (8 and 9 are assumed to be equivalent to 7).

*Most significant digit:* This is the selector for negative-value detection:

Value	Description
1	Activate negative-conductance warnings.
2	Activate negative-capacitance warnings.
4	Limit the total number of messages to 10.

*Middle digit:* This is the selector for NaN and voltage-limitation warnings:

Value	Description
1	Allow message when NaN is detected.
2	Activate warnings when a high voltage is detected before voltage limitation.

i.e. 1,3,5,7 are equivalent (and 0,2,4,6 in the same way).

*Least significant digit:* This selects the type of diode warnings (not useful for TFT because there is no diode in the model):

Value	Description
1	Activate warnings when a diode is ON in IC.
2	Activate warnings when a diode is ON during iterations.
4	Limit the total number of messages to 10 (not for warnings on IC).

Example : EXPERT = 370

- All negative-capacitance and negative-conductance warnings are displayed.
- NaN warnings are displayed.
- All high-voltage warnings are displayed, the number of messages during iterations is limited to 10.

## New Charge Model (only for Poly-TFT model Level=36)

In level 36 the charges qgs and qgd are numerically calculated from the capacitances cgs and cgd. This leads to problems of charge non-conservation. A charge based approach to transient modeling is being implemented with encouraging results.

# Calendar of Events

## January

1	Happy New Year
2	
3	
4	
5	
6	
7	
8	
9	
10	
11	
12	
13	
14	
15	
16	
17	
18	
19	
20	
21	
22	
23	
24	
25	
26	
27	EDA TechnoFair - Japan
28	EDA TechnoFair - Japan
29	EDA TechnoFair - Japan
30	
31	

## February

1	
2	DesignCon '99 - Santa Clara
3	DesignCon '99 - Santa Clara
4	
5	
6	
7	
8	
9	
10	
11	
12	
13	
14	
15	
16	
17	ISSCC - San Francisco
18	ISSCC - San Francisco Analog '99 - Munich, Germany
19	Analog '99 - Munich, Germany
20	
21	
22	
23	
24	
25	
26	
27	
28	

## Bulletin Board



### See Us at the Land of the Rising Sun

Silvaco International will have a major presence at the EDA TechnoFair in Yokohama, Japan. Join our largest offshore branch for a conference full of demonstrations, fun, and progress.



### Industry Leaders Train at Silvaco

Leading players in the semiconductor industry will train at Silvaco in February. To find out how you can participate in one of our sessions, contact us at 408.567.1000 or on the web at [www.silvaco.com](http://www.silvaco.com).



### 1999 Marks 15 Years of Silvaco

1999 is the 15th anniversary of Silvaco International. In 1984, a foundation of hard work, great ideas, and determined optimism was established. Join us throughout the year to celebrate our success.

For more information on any of our workshops, please check our web site at <http://www.silvaco.com>

The Simulation Standard, circulation 18,000 Vol. 12, No. 1, January 1999 is copyrighted by Silvaco International. If you, or someone you know wants a subscription to this free publication, please call (408) 567-1000 (USA), (44) (1483) 401-800 (UK), (81)(45) 820-3000 (Japan), or your nearest Silvaco distributor.

Simulation Standard, TCAD Driven CAD, Virtual Wafer Fab, Analog Alliance, Legacy, ATHENA, ATLAS, MERCURY, VICTORY, VYPER, ANALOG EXPRESS, RESILIENCE, DISCOVERY, CELEBRITY, Manufacturing Tools, Automation Tools, Interactive Tools, TonyPlot, TonyPlot3D, DeckBuild, DevEdit, DevEdit3D, Interpreter, ATHENA Interpreter, ATLAS Interpreter, Circuit Optimizer, MaskViews, PSTATS, SSuprem3, SSuprem4, Elite, Optolith, Flash, Silicides, MC Depo/Etch, MC Implant, S-Pisces, Blaze/Blaze3D, Device3D, TFT2D/3D, Ferro, SiGe, SiC, Laser, VCSELS, Quantum2D/3D, Luminous2D/3D, Giga2D/3D, MixedMode2D/3D, FastBlaze, FastLargeSignal, FastMixedMode, FastGiga, FastNoise, Mocasim, Spirit, Beacon, Frontier, Clarity, Zenith, Vision, Radiant, TwinSim, , UTMOST, UTMOST II, UTMOST III, UTMOST IV, PROMOST, SPAYN, UTMOST IV Measure, UTMOST IV Fit, UTMOST IV Spice Modeling, SmartStats, SDDL, SmartSpice, FastSpice, Twister, Blast, MixSim, SmartLib, TestChip, Promost-Rel, RelStats, ReLib, Harm, Ranger, Ranger3D Nomad, QUEST, EXACT, CLEVER, STELLAR, HIPEX-net, HIPEX-r, HIPEX-c, HIPEX-rc, HIPEX-crc, EM, Power, IR, SI, Timing, SN, Clock, Scholar, Expert, Savage, Scout, Dragon, Maverick, Guardian, Envoy, LISA, ExpertViews and SFLM are trademarks of Silvaco International.

# Hints, Tips and Solutions

Mustafa Taner, Applications and Support Engineer

## Q. How can I obtain Fmax, Gmax, Upg and K factor plots from s-parameter measurements.

A. Typically, *UTMOST III* s-parameter module is used to measure the FT versus IC (collector current) characteristic of bipolar transistors. The FT\_CE routine can also be utilized to extract important parameters such as Fmax, Gmax, Upg and K factor. These parameters provide a clear indication of the performance of the device as an amplifier.

Brief description of these parameters as follows:

**GMAX** (Maximum Available Gain):

GMAX is the forward power gain when the input and output are simultaneously conjugately matched. GMAX is only defined for an unconditionally stable device. ( $K > 1$ ).

**FMAX** : The frequency where GMAX gain is equal to 1.

**K** (Stability Factor):

K is a measure of oscillatory tendency. For  $K < 1$  the device is potentially unstable and can oscillate by the application of some combination of passive load and source admittances. For  $K > 1$  the device is unconditionally stable. K plays an important role in amplifier design.

**UPG** (Unilateral Power Gain):

UPG is defined as the forward power gain in a feedback amplifier whose reverse gain has been adjusted to zero by a loss less reciprocal feedback network.

The relations between the measured s-parameters and K, GMAX and UPG are given as:

$$D = s_{11} \times s_{22} - s_{12} \times s_{21}$$

$$K = (1 + |D|^2 - |s_{11}|^2 - |s_{22}|^2) / (2 \times |s_{12}| \times |s_{21}|)$$

$$GMAX = |s_{21} / s_{12}| \times (K - \sqrt{K^2 - 1})$$

The FT\_CE routine is set as default to measure four s-parameters (s11, s12, s21 and s22) and display the h21 vs frequency. The Ft is defined as the frequency point where  $|H_{21}|$  is equal to 1. The user can view the GMAX, UPG and K factor plots and extract Fmax from the GMAX versus frequency plots. In order to view the GMAX and K factor plots the the Fit Variable "Gmax=1, Upg=2" should be set to 1. (Figure 1). Same variable should be set to 2 to view UPG versus frequency and K factor plot. If set to 0 (default) the H21 versus frequency plots will be displayed and FT will be extracted for each DC bias point. The Fmax will be extracted from the GMAX vs. freq. (Figure 2) or UPG vs. freq. plots (Figure 3).

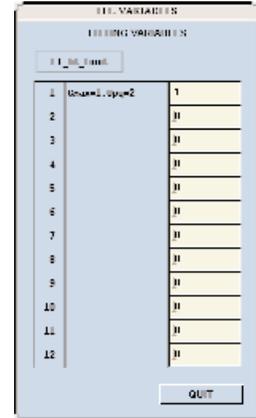


Figure 1. Fit Variables screen of the FT\_CE routine.

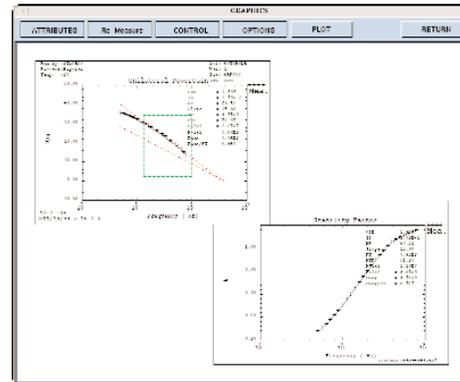


Figure 2. GMAX vs Frequency and K Factor plots.

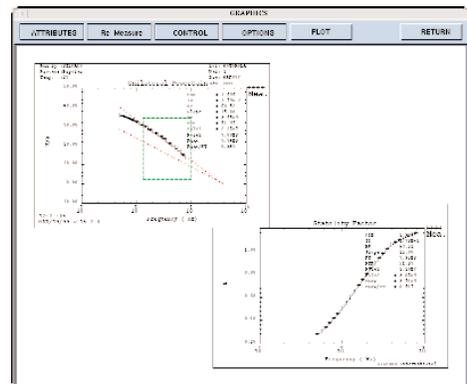


Figure 3. UPG vs. Frequency and K Factor plots.

### Call for Questions

If you have hints, tips, solutions or questions to contribute, please contact our Applications and Support Department  
 Phone: (408) 567-1000 Fax: (408) 496-6080  
 e-mail: [support@silvaco.com](mailto:support@silvaco.com)

### Hints, Tips and Solutions Archive

Check our our Web Page to see more details of this example plus an archive of previous Hints, Tips, and Solutions  
[www.silvaco.com](http://www.silvaco.com)

# Your Investment in Silvaco is SOLID as a Rock!!

While others faltered, Silvaco stood SOLID for 15 years. Silvaco is NOT for sale and will remain fiercely independent. Don't lose sleep, as your investment and partnership with Silvaco will only grow.

## **SILVACO** INTERNATIONAL

### USA HEADQUARTERS

**Silvaco International**  
**4701 Patrick Henry Drive**  
**Building 2**  
**Santa Clara, CA 95054**  
**USA**

**Phone: 408-567-1000**  
**Fax: 408-496-6080**

**sales@silvaco.com**  
**www.silvaco.com**

### CONTACTS:

**Silvaco Japan**  
jpsales@silvaco.com

**Silvaco Korea**  
krsales@silvaco.com

**Silvaco Taiwan**  
twsales@silvaco.com

**Silvaco Singapore**  
sgsales@silvaco.com

**Silvaco UK**  
uksales@silvaco.com

**Silvaco France**  
frsales@silvaco.com

**Silvaco Germany**  
desales@silvaco.com

*Products Licensed through Silvaco or e\*ECAD*

