Simulation Standard

Connecting TCAD To Tapeout

A Journal for Circuit Simulation and SPICE Modeling Engineers

PHILIPS Model 9
New MM9 Extraction routine in *UTMOST III*

Introduction

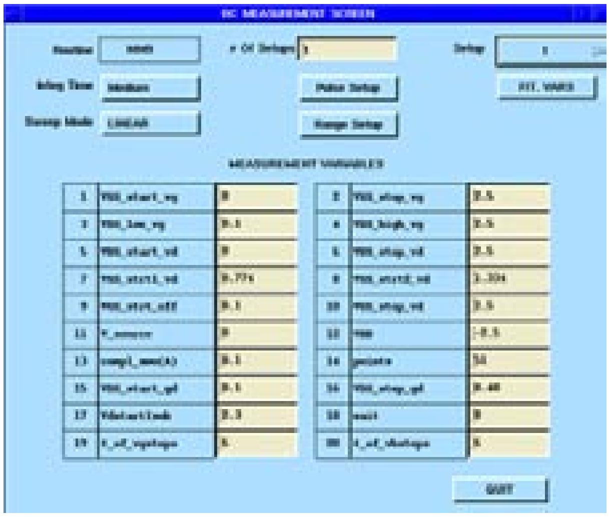
In collaboration with STMicroelectronics Central R&D at Crolles (France), a new routine has been developed in ***UTMOST III*** to provide a complete solution for MOS Philips Model 9 parameter extraction. This methodology[1] is based on the local optimization method; we can determine a limited set of 18 parameters (so called miniset) to describe the electrical behavior of each device, considering it as the reference device. This miniset includes all the electrical parameters for an individual device[1]. Specific local optimization strategies are described in this article to obtain good minisets. From these minisets, we can extract scaling parameters, using simple linear regressions. We will obtain a new complete set of parameters (so called maxisets). All temperature dependant parameters can also be extracted.

Measures

The MM9 extraction algorithm requires measurements from a Long and Large device, Short devices and Narrow devices (asBSIM3_MG routine). For each device a total of six sets of I-V measurements are requested.

Set 1: IDS versus VGS at different VBS and at low fixed VDS.
Set 2: IDS versus VDS at different VGS and at low fixed VBS.
Set 3: IDS versus VGS at different VBS and at high fixed VDS.
Set 4: IDS versus VDS at different VGS and at high fixed VBS.
Set 5: IDS versus VGS at different VDS and at VBS=0V.
Set 6: ISUB versus VGS at different VDS and at VBS=0V.

If these MM9 measurements are stored in a log file and this log file is activated in ***UTMOST III***, then the ALL_DC, ALL_ISUB and AL_IDVGD (ID/VG-VD for several devices) can access these measure.



1	VGS_start_vg	0	2	VGS_stop_vg	2.5
3	VDS_low_vg	0.1	4	VDS_high_vg	2.5
5	VDS_start_vd	0	6	VDS_stop_vd	2.5
7	VGS_start1_vd	0.77s	8	VGS_start2_vd	1.33s
9	VGS_step_vdf	0.1	10	VGS_stop_vd	2.5
11	V_source	0	12	VBB	-0.5
13	compl_mea(A)	0.1	14	points	50
15	VGS_start_gd	0.1	16	VGS_stop_gd	0.40
17	VGstartIsub	2.3	18	wait	0
19	#_of_vgsteps	5	20	#_of_vbsteps	5

Figure 1: MM9 Measurement Setup Screen

The twenty measurement variables are used and described hereafter.

1 VGS_start_vg Starting value for the gate sweep range (ID/VG curves)

2 VGS_stop_vg Stop value for the gate voltage sweep range (ID/VG curves)

3 VDS_low_vg Low fixed VDS bias for the ID/VG-VB linear characteristic

4 VDS_high_vg High fixed VDS bias for the ID/VG-VB saturation characteristic

Continued on page 2....

SILVACO INTERNATIONAL

5	VDS_start_vd	Starting value for the drain voltage sweep range (ID/VD curves)
6	VDS_stop_vd	Stop value for the drain voltage sweep range (ID/VD curves)
7	VGS_strt1_vd	Calculated starting value for VGS steps for ID/VD-VG curve (VBS=0V)
8	VGS_strt2_vd	Calculated starting value for VGS steps for ID/VD-VG curve (high VBS)
9	VGS_strt_off	Offset voltage used to calculate VGS_strt1_vd and VGS_strt2_vd
10	VGS_stop_vd	Stop value for VGS steps (ID/VD curves)
11	V_source	Constant source voltage
12	VBB	Maximum VBS voltage for ID/VG-VB curves and high VBS for ID/VD-VG curve
13	compl_smu(A)	Current SMU's compliance
14	points	Number of sweep data points for each characteristics
15	VDS_start_gd	Starting value for the VDS steps (ID/VG-VD curve at VBS=0V)
16	VDS_step_gd	Step value for the VDS steps (ID/VG-VD curve at VBS=0V)
17	VdstartIsub	Starting value for the VDS steps at VBS=0V (ISUB/VG-VD curve)
18	wait	Wait time in microseconds, between measurements
19	#_of_vgsteps	Number of VGS steps for ID/VD-VG curves
20	#_of_vbsteps	Number of VBS steps for ID/VG-VB curves and for ID/VG-VD curve

Notes:

1. Variables #7 and #8 are calculated as follows:

   VGS_strt1_vd = VTH(VBS=0V) + VGS_strt_off, where VTH(VBS=0V) is the extracted threshold voltage for ID/VGS at VDS low and VBS=0V.

   VGS_strt2_vd = VTH(VBS=VBB) + VGS_strt_off, where VTH(VBS=VBB) is the extracted threshold voltage for ID/VGS at VDS low and VBS=VBB.

2. The number of VDS step for ISUB/VGS curve is constant equal to 3. The stop value for the VDS steps is VDS_high_vg.

3. The maximum number of points is 201 (variable #14).

4. The maximum value for the variable #19 and #20 is 7.

Miniset Extraction Routine

Miniset Definition

The first step of the MM9 extraction methodology is to extract a miniset of parameters for each device, part of the device strategy selection. For this extraction which will be based on various local optimization strategies, we need to adjust 18 model parameters which will be sufficient to describe the electrical behavior of each device, considered as the reference transistor. These 18 parameters are listed hereafter: VTOR, BETSQ, THE1R, THE2R, KOR, KR, VSBXR, MOR, GAMOOR, ZET1R, VSBTR, VPR, ALPR, GAM1R, THE3R, A1R, A2R, and A3R. All the scaling parameters (SLxx, SWxx, and STxx) must be set to 0.

Clipping Note

In order to obtain accurate minisets, the MOS level 9 model has been slightly modified. The first modification concerns the parameter clipping. We have introduced a

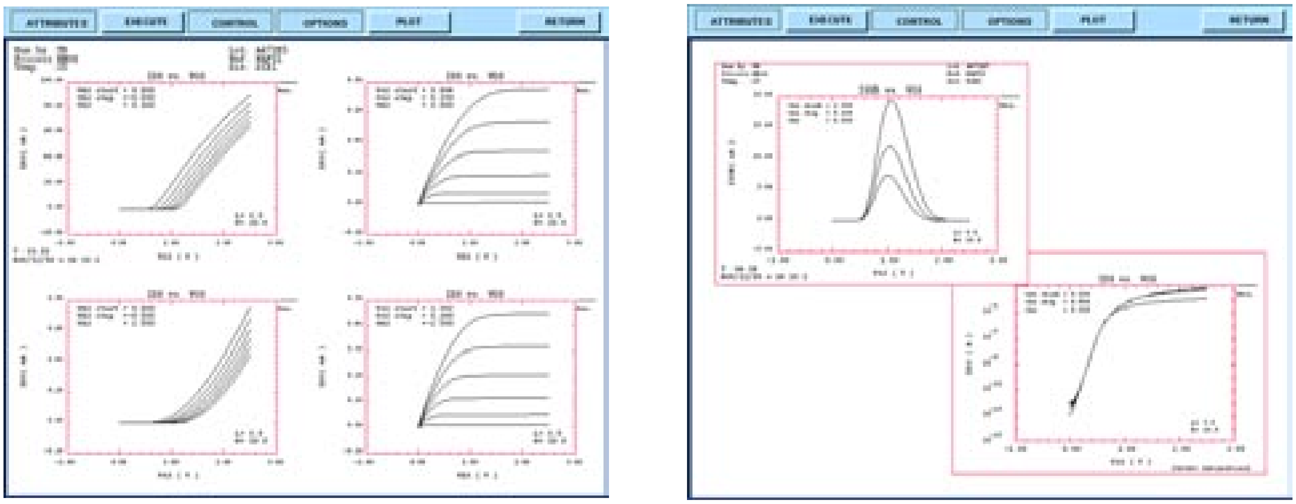


Figure 2: The six measurement characterization for MM9 Routine.

new parameter NOCLIP. If set to 1, clipping on THE3R and GAM1R is removed. Negative value for THE3R can be found for long and wide devices. This NOCLIP has been also introduced for GAM1R, to allow negative values during the optimization; but this almost never appears. The user may take care using NOCLIP=1. If THE3R is too negative, solver may not converge.

The second modification concerns the de-activation of the weak and moderate inversion modeling by using ZET1R value, weak inversion factor, as a switch. When ZET1R>5, there is no more subthreshold current. But, the linear region modeling remains the same. A better optimization of the first-order parameters can be performed.

These two modifications are useful to get an accurate miniset using the local optimization strategies.

User Initial Model Parameter Values

Before launching the extraction procedure, it is important to define the initial value for the model parameters to be optimized. KR and VSBXR model parameters describe the high back bias body effect. These two parameters may not be extracted if only one body effect appears in our technology. For that matter, we can use the ID/VG-VB or LGAMMA routine. Using the "Fit" option, and displaying the:

$$VT \ \ vs. \ \ \sqrt{\phi + |V_{SB}|} - \sqrt{\phi}$$

Parameters	1 Body Effect	2 Body Effect
KOR	0.6	0.6
KR	Not Important	0.3
VSBXR	100	1
ETAMR; not optimized	1	2
ETAGAMR; not optimized	1	2

Table 1. Initial values for miniset #1.

If we do not obtain a straight line, we have to use KR and VSBXR to describe the two body effects. Tables 1, 2 and 3 summarize the initial values of the miniset parameter.

In Table 2, these values can be adapted if one strategy does not give good results. This may happen if initial values are really too far from values to obtain. Minimum and maximum values are also important for local optimization strategies. The table below gives you an example of possible minimum and maximum values.

All the scaling parameters (whose names begin with SL, SW or ST) must be set to 0. TR will be automatically set to the measurement temperature during the miniset optimizations.

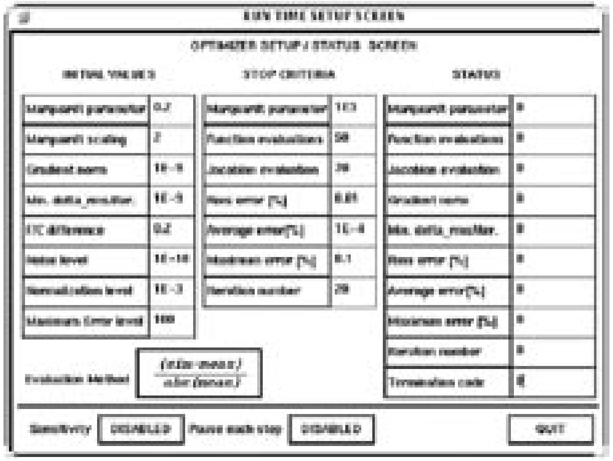
All these initial values must be copied in the User column of the Parameter screen.

Parameters	Initial Value	Parameters	Initial Value	Parameters	Initial Value
VTOR	0.7	VPR	1.5	PHIBR	0.65
BETSQ	1.E-4	ALPR	0.01	LAP	0
THE1R	0.1	THE3R	0.01	WOT	0
THE2R	0.1	VSBTR	100	LER	Mask Length
MOR	0.5	A1R	3	WER	Mask Width
GAMOOR	5.E-4	A2R	20	LVAR	0
ZET1R	2	A3R	1	WVAR	0
GAM1R	0.01	ETADSR	0.6	TR	Temperature

Table 2. Initial values for miniset #2.

Parameters	Minimum	Maximum	Parameters	Minimum	Maximum
VTOR	-2	2	ZET1R	0	5
BETSQ	1.E-6	1.E-3	VSBTR	0	100
THE1R	1.E-3	2	VPR	0.05	100
THE2R	1.E-8	0.5	QLPR	1.E-7	0.1
KOR	0.01	1.5	GAM1R	1.E-5	0.5
KR	0	1	THE3R	-0.05	1
VSBXR	0	100	A1R	0	30
MOR	0	2	A2R	10	100
GAMOOR	1.E-9	0.1	A3R	0.1	10

Table 3. Initial values for miniset #3.



RUN TIME SETUP SCREEN

OPTIMIZER SETUP / STATUS SCREEN

INITIAL VALUES		STOP CRITERIA		STATUS	
Marquardt parameter	0.2	Marquardt parameter	1E3	Marquardt parameter	0
Marquardt scaling	2	Function evaluations	50	Function evaluations	0
Gradient norm	1E-9	Jacobian evaluation	20	Jacobian evaluation	0
Min. delta_rmultiplier	1E-5	Rms error [%]	0.01	Gradient norm	0
FTC difference	0.2	Average error[%]	1E-4	Min. delta_rmultiplier.	0
Noise level	1E-10	Maximum error [%]	0.1	Rms error [%]	0
Normalization level	1E-3	Iteration number	20	Average error[%]	0
Maximum Error level	100			Maximum error [%]	0
				Iteration number	0
				Termination code	0

Evaluation Method: (n1n-meas) / n1n (meas)

Figure 3. Optimizer Setup recommended.

Optimizer Setup

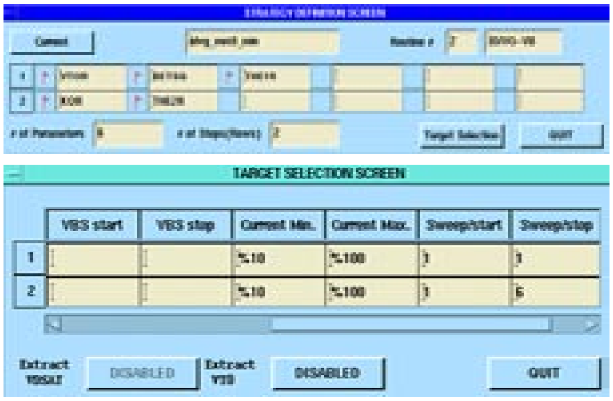
The recommended optimizer setup is illustrated in Figure 3.

Local Optimization Strategies

Nine local optimization strategies are proposed. The name given to the strategies is not important, and can be changed without any problem. The strategy number is important. ***UTMOST III*** will choose by his own which strategy to apply, depending on the type of transistor we are working on. The “Geometry Selected Screen” we can define in the “Target Selection Screen” is not important, as we apply the local optimization only on the device displayed on the “Graphics Screen”. MM9 Routine will recognize the type of device we have selected in the Strategy Screen, and then order them to execute the minisets optimization. First we work on the Long&Large device, then on the short devices, and finally on the narrow devices.

We work on ID/VG-VB at low VDS for several VBS.

Strategy #1: Linear region for one body effect factor: idvg_mm9_min.



STRATEGY DEFINITION SCREEN

Current: idvg_mm9_min   Routine # 2   IDVG-VB

1: VTOR, BETSQ, THE1R
2: KOR, THE2R

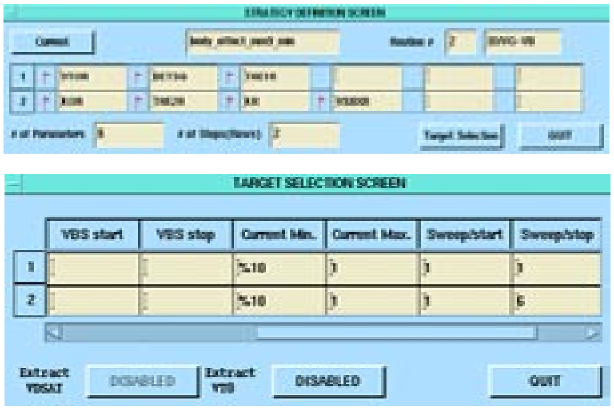
# of Parameters 5   # of Steps(Rows) 2

TARGET SELECTION SCREEN

	VBS start	VBS stop	Current Min.	Current Max.	Sweep/start	Sweep/stop
1			%10	%100	1	1
2			%10	%100	1	6

Figure 4. idvg_mm9_min Local Optimization Definition.

Strategy #2: Linear region for two body effect factors Î body_effect_mm9_min.



STRATEGY DEFINITION SCREEN

Current: body_effect_mm9_min   Routine # 2   IDVG-VB

1: VTOR, BETSQ, THE1R
2: KOR, THE2R, KR, VSBXR

# of Parameters 7   # of Steps(Rows) 2

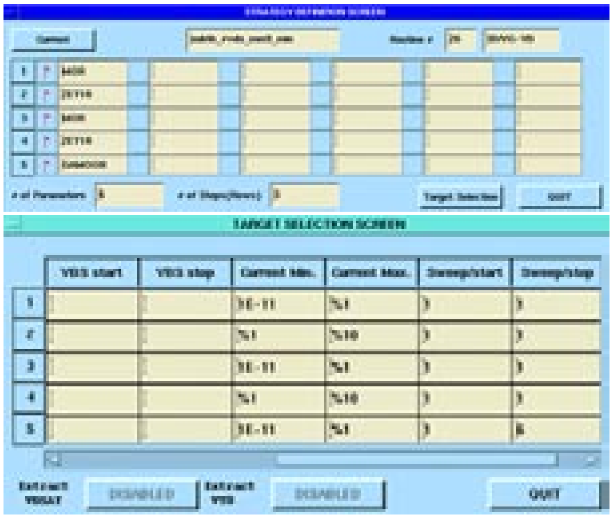
TARGET SELECTION SCREEN

	VBS start	VBS stop	Current Min.	Current Max.	Sweep/start	Sweep/stop
1			%10	1	1	1
2			%10	1	1	6

Figure 5. body_effect_mm9_min Local Optimization Definition.

The optimized parameters are: VTOR, BETSQ, THE1R, KOR, THE2R, KR, VSBXR. During the optimization, ZET1R is automatically set to 100, which will disable the weak and moderate current modeling.

Strategy #3: Weak and moderate inversions: subth_ #vds_mm9_min.



STRATEGY DEFINITION SCREEN

Current: subth_vds_mm9_min   Routine # 2   IDVG-VB

1: MOR
2: ZET1R
3: MOR
4: ZET1R
5: GAMOOR

# of Parameters 5   # of Steps(Rows) 5

TARGET SELECTION SCREEN

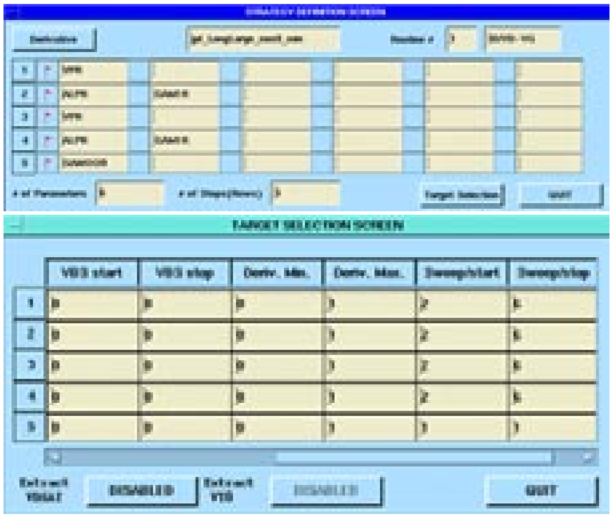
	VBS start	VBS stop	Current Min.	Current Max.	Sweep/start	Sweep/stop
1			1E-11	%1	1	1
2			%1	%10	1	1
3			1E-11	%1	1	1
4			%1	%10	1	1
5			1E-11	%1	1	6

Figure 6. subth_#vds_mm9_min Local Optimization Definition.

The optimized parameters are: MOR, GAMOOR, ZET1R. Before executing this optimization, ZET1R is automatically set to a value lower than 5 to enable weak and moderate current modeling. As MOR and ZET1R are strongly correlated, you may adjust this strategy if the result obtained is not good enough.

We work now on the derivative of ID/VD-VG at VBS=0V for different.

Strategy #4: gds optimization for long&large device: gd_LongLarge_mm9_min.



gd_LongLarge_mm9_min

Routine # 1 ID/VD-VG

1		VPR		
2	*	ALPR		GAM1R
3	*	VPR		
4	*	ALPR		GAM1R
5	*	GAMOOR		

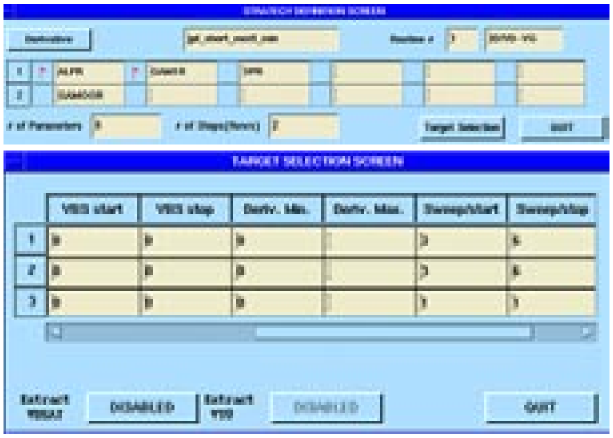
# of Parameters 5   # of Steps(Rows) 5

	VBS start	VBS stop	Deriv. Min.	Deriv. Max.	Sweep/start	Sweep/stop
1	0	0	0	1	2	6
2	0	0	0	1	2	6
3	0	0	0	1	2	6
4	0	0	0	1	2	6
5	0	0	0	1	1	1

Figure 7. gd_LongLarge_mm9_min Local Optimization Definition.

The optimized parameters are: VPR, ALPR, and eventually GAMOOR. For the long&large device, VPR is optimized. LER is automatically set to the length value of the transistor displayed. Like this VPR is identical to Vp. The default GAM1R is 0. GAMOOR has been normally already extracted from the subthreshold region, but may need to be optimized again in that region if more accuracy is needed for low VGS.

Strategy #5: gds optimization for short channel devices: gd_short_mm9_min.



gd_short_mm9_min

Routine # 1 ID/VD-VG

1	*	ALPR	*	GAM1R	VPR
2		GAMOOR			

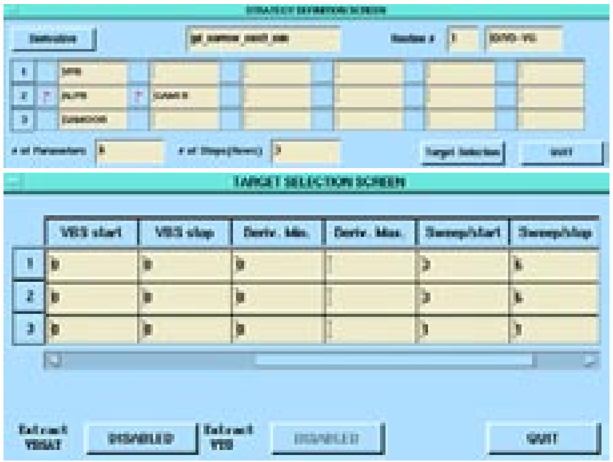
# of Parameters 3   # of Steps(Rows) 2

	VBS start	VBS stop	Deriv. Min.	Deriv. Max.	Sweep/start	Sweep/stop
1	0	0	0		3	6
2	0	0	0		3	6
3	0	0	0		1	1

Figure 8. gd_short_mm9_min Local Optimization Definition.

The optimized parameters are: ALPR, GAM1R and eventually GAMOOR. VPR is calculated using the following formula: As for the long&large device, GAMOOR may need a new optimization especially for low VGS. LER is automatically set to the length of the device displayed.

Strategy #6: gds optimization for narrow channel devices : gd_narrow_mm9_min.



gd_narrow_mm9_min

Routine # 1 ID/VD-VG

1		VPR		
2	*	ALPR	*	GAM1R
3		GAMOOR		

# of Parameters 4   # of Steps(Rows) 3

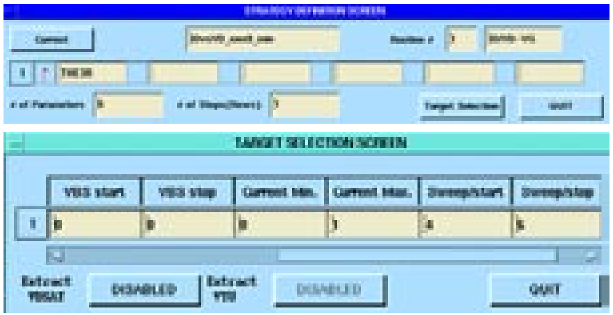
	VBS start	VBS stop	Deriv. Min.	Deriv. Max.	Sweep/start	Sweep/stop
1	0	0	0		3	6
2	0	0	0		3	6
3	0	0	0		1	1

Figure 9. dg_narrrow_mm9_min Local Optimization Definition.

The optimized parameters are: ALPR, GAM1R, and eventually VPR. VPR and LER are calculated as for the short devices. Although there is no scaling rule for Vp, VPR may need new optimization for narrow devices.

We work now on ID/VD-VG at VBS=0V and for different VGS.

Strategy #7: Saturation current: IDvsVD_mm9_min.



IDvsVD_mm9_min

Routine # 1 ID/VD-VG

1	*	THE3R

# of Parameters 1   # of Steps(Rows) 1

	VBS start	VBS stop	Current Min.	Current Max.	Sweep/start	Sweep/stop
1	0	0	0	1	4	6

Figure 10. IdvsVD_mm9_min Local Optimization Definition.

The optimized parameter is: THE3R. Take care that minimum value for THE3R may not be lower than –0.005.

We work on ID/VG-VB for different VBS at low and high VDS.

(to be continued in a future issue)

Parallel .ALTER Statements in *SmartSpice*

Parallel .ALTER

The `.ALTER` statement is designed to allow a SPICE input deck to be re-run with a change in a single parameter. This feature is especially useful in characterization. For this type of work, users will want to run a single deck many times over, changing a single parameter each time. The runs are generally short, taking a few minutes to a few hours to complete. Very often the circuit itself is not so large, but the sequential nature of the analysis means that the characterization takes a long time.

It is increasingly popular to run such simulations on multi-processor machines, with anywhere from two to twelve CPUs, sharing the same memory. A significant speed-up in characterization runs could be achieved if the `.ALTERs` could be spread over all the CPUs.

`Separate` Command

SmartSpice's '`-P <n>`' option, in conjunction with its `separate' command (exclusively developed for ***SmartSpice***), allows the user to distribute a deck with one or more `.ALTERs` over several CPUs. Operating in batch mode, ***SmartSpice*** takes an input deck containing `.ALTER` statements, and farms out each `.ALTER` to a separate CPU. The user can specify the number of CPUs (`<n>`) to be used. In order to maximize efficiency, ***SmartSpice*** will run no more than `<n>` `.ALTERs` at a time. As each CPU finishes, ***SmartSpice*** will give it another `.ALTER`, until all have been processed.

Each `.ALTER` statement will produce a separate RAW file, and output file, whose names are identical to the original input deck, but with'`-#.in`' replacing the extension, where # is the number of the `.ALTER` statement in the input deck.

Performance

The resulting performance improvement is significant. For most decks the parallel speed-up is linear: using four CPUs for example can make your overall simulation finish four times faster. Figure 1 shows the execution time for an input deck with three `.ALTERs` (and therefore, four separate input decks), as a function of the number of CPUs used. Also shown is the theoretical ideal behavior. It is clear that ***SmartSpice*** performance is close to ideal for decks taking longer than a few seconds to run. By specifying the number of CPUs over which the decks are to be processed, a user can, for example, allow a small number of CPUs to share the available memory if the circuit is very large, or run a `.ALTER` on each CPU if the circuit is small. There is a small overhead associated with managing multiple CPUs (typically a few seconds per `.ALTER`), which should affect performance only on the shortest, smallest decks. For most realistic decks, it will be negligible.

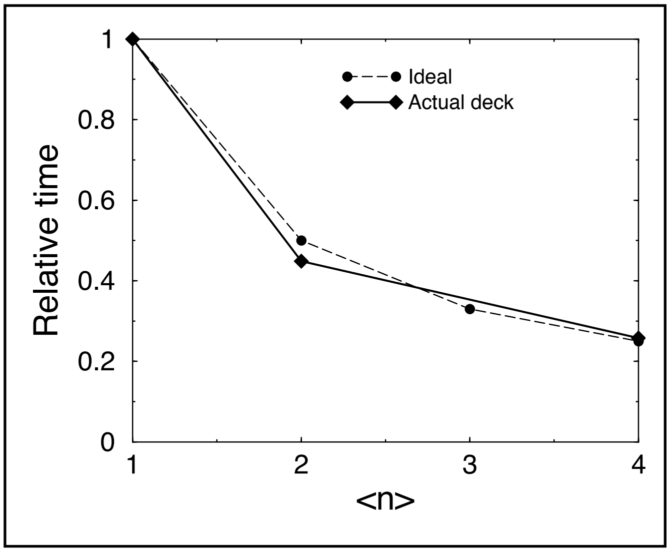


Figure 1. Relative simulation time for a deck containing three `.ALTER` statements as a function of number of CPUs used. Actual performance of SmartSpice is almost ideal. (The test machine is a four-CPU SUN machine running Solaris2. Each `.ALTER` took about one minute to run).

Scholar

An Advanced Hierarchical Schematic Capture

Introduction

The debut of Silvaco's new schematic editor, ***Scholar,*** is drawing closer. ***Scholar*** is a sophisticated design tool which derives its power and flexibility from the fact that it is built on top of the general purpose circuit database. ***Scholar*** has both a fully functional command language, and an easy to use graphical front end. As it has been developed in-house, it will provide unprecedented interoperability with Silvaco's flagship SPICE simulator, ***SmartSpice.*** The planned addition of an EDIF interface will provide complete compatibility with existing circuit libraries.

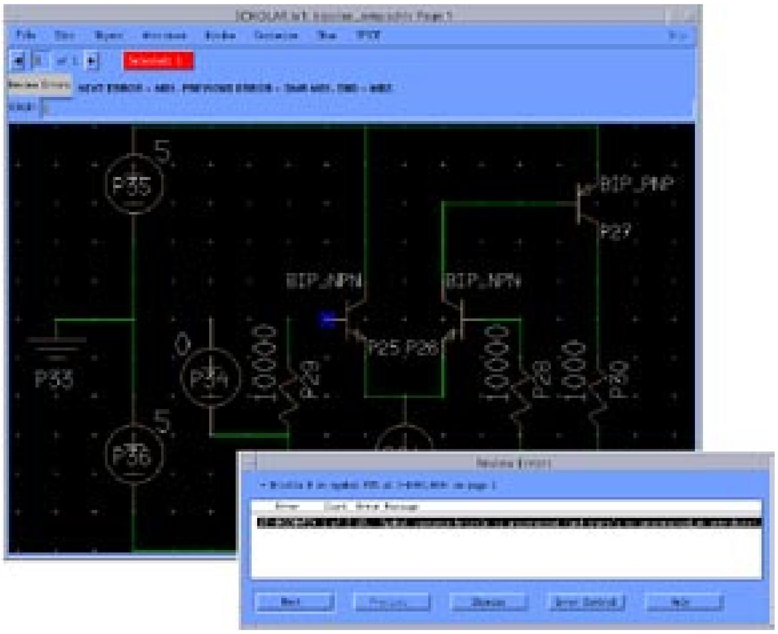


Figure 1. Schematic & Rules checker.

Underlying Structure

Scholar is built on top of Silvaco's in-house wirelist database library. The wirelist has been designed to be a superset of a SPICE netlist, extending the concept to include everything from straight forward analog simulation to parasitic extraction. It forms the backbone for Silvaco's forthcoming EDA tools, enhancing interoperability and enabling an unprecedented level of integration.

Scholar is comprised of several functional blocks. A comprehensive netlisting module converts wirelists, stored in the wirelist database, into hierarchical or flat netlists. The interpreter module uses a fully featured programming language tailored for use with the schematic editor to provide customization and a command line interface to most of Scholar's functionality.

Current Development

The low-level libraries which underpin ***Scholar's*** command-line functionality are virtually complete. Work is now progressing on implementing the graphical user interface. A complete set of basic schematic editing features are already implemented, along with a library of standard components. Completed designs can be verified with a design rule checker which automatically detects common problems such as short or open nets or dangling wires. Figure 1 shows the current ***Scholar*** front-end, along with a diagnostic error caught by the rule checker.

Work is also in progress on fine-tuning the interface between ***Scholar*** and ***SmartSpice. SmartSpice*** analysis options and include files are supported through ***Scholar*** menus, and ***Scholar*** will run ***SmartSpice*** automatically when all required control cards have been generated.

Future Direction

The next milestone for ***Scholar*** is the provision of comprehensive back annotation and cross-probing features. Back annotation allows the results of a simulation to be displayed directly in the schematic window. Cross-probing gives the user an easy way to query circuit parameters and output waveforms by selecting wires and components directly on the schematic. Further tightening of the interface to ***SmartSpice*** will bring even more parameters under the control of ***Scholar.***

The foundations of ***Scholar's*** design management layer are in place, supporting design locking and versioning. The finishing touches will futher improve functionality, providing a fully automated environment for maintaining multiple branches on a design tree.

Conclusion

The imminent release of ***Scholar*** will provide Silvaco's customers with a complete schematic design and simulation solution. The tight integration with ***SmartSpice*** and other tools enabled by the common database, in conjuction with the support of an industry-standard EDIF interface, make ***Scholar*** an extremely useful new tool in Silvaco's product line.

SPAYN
Recent Developments

Introduction

An important aspect of statistical process control in IC production is the ability to predict circuit performance variation in the manufacturing process. Two new features in ***SPAYN*** allow the user, for a particular circuit performance parameter, to rapidly calculate an estimate of the standard deviation and also generate a yield distribution utilising Monte Carlo simulations, thus allowing a full statistical analysis of the circuit performance parameter distribution.

For any given data base containing SPICE model parameters, an initial exploratory data analysis usually consists of trying to identify relationships and interdependencies between the specified variables. This is usually achieved by looking at the correlation structure, and examining histogram and scattergram plots of the various circuit parameters. However due to the large number of SPICE model parameters that are generally extracted, it is of great benefit to be able to generate these plots automatically. The new ***SPAYN*** **"**Macro Command" feature allows the user to do just that.

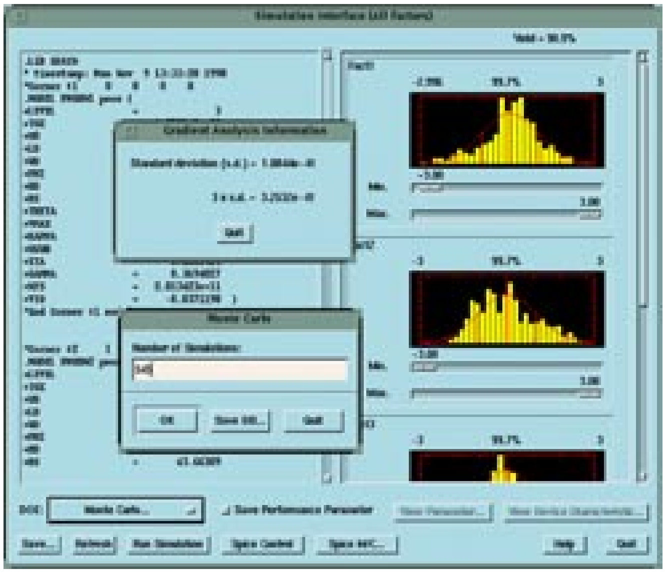


Figure 1. SPAYN simulation interface window, with gradient analysis standard deviation estimate and Monte Carlo simulation dialog (inset).

Statistical Analysis of Circuit Performance Parameters

One of the primary objectives of statistical ``worst-case" yield modelling is to predict best and worst circuit performance due to fluctuations in the IC manufacturing process. The aim is to isolate those parameters causing the largest variation in circuit performance, so that they can be strictly monitored and controlled. An understanding of the sources of variation provides an insight, at the design stage, into where any problems are likely to occur during production.

There are two new features in ***SPAYN*** that allow the user to statistically analyse a particular circuit performance parameter. The circuit simulations are performed by connecting ***SPAYN*** to an external simulator such as ***SmartSpice***, ***SPAYN*** is then used to analyse the results. The first new feature, known as the ``Gradient Analysis" method [1], computes an accurate estimate of the standard deviation for a given circuit performance parameter. The second new feature allows the statistical investigation of a performance parameter generated through Monte Carlo simulations.

The ``Gradient Analysis" approach [1] permits designers to rapidly calculate an estimate of the circuit performance standard deviation utilising independent or quasi-independent reference design parameters such as principal component factors or dominant parameters. The Gradient Analysis standard deviation estimate is based on a linear approximation by considering the circuit performance parameter as a function of independent reference design parameters.

Monte Carlo simulations of a given circuit can be performed using ***SmartSpice***, and the performance parameter stored as a new ***SPAYN*** variable for further analysis. Thus allowing an engineer to examine the distribution of a particular circuit performance parameter, calculate various descriptive statistics or utilise any of ***SPAYN***'s other statistical capabilities.

As an example of the above new features (Figure 1), consider a circuit consisting of a single MOS transistor biased with -2.5 volts on the gate, and device dimensions W/L = 20/2.5 microns. The ***SmartSpice*** control file then specifies the drain source voltage to be ramped from 0 -> -5 volts in steps of 0.2 of a volt. For this analysis let the circuit performance parameter be the maximum value of the drain current (ID_MAX) for a given set of parameter values. In this case the independent reference design parameters are taken as the first four principal components, which account for 89% of the variation in the data.

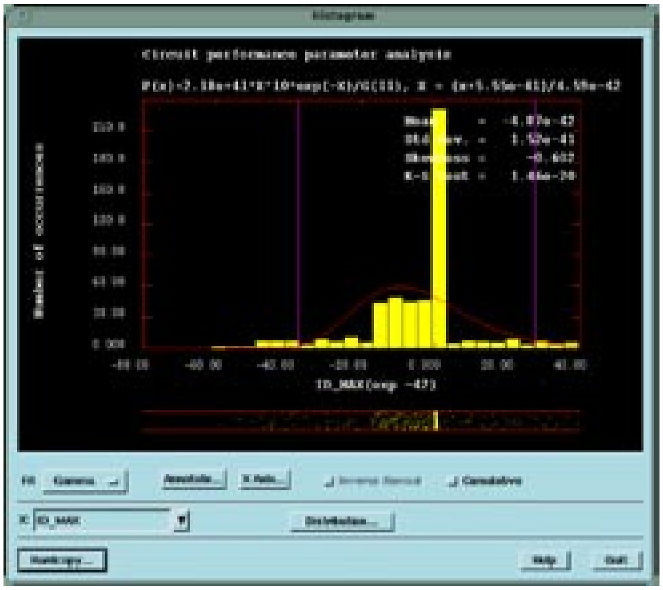


Figure 2. Histogram of the circuit performance parameter ID.MAX.

Utilising the Gradient Analysis method, the estimated standard deviation of ID_MAX is 1.10e-41 compared to the sample standard deviation of 1.52e-41 generated from the Monte Carlo simulations. Figure 2 shows a histogram plot of the Monte Carlo simulations for the circuit performance parameter ID_MAX. The vertical lines on the plot indicate the +/- 3 σ limits of the Gradient Analysis standard deviation estimate. Using the distribution fit option in the histogram window, a Gamma distribution best describes this particular set of circuit performance parameter data.

Generating multiple Histogram and Scattergram plots

With any statistical data analysis an applications engineer is interested in examining possible relationships between the various parameters in a particular circuit design. This is especially important with the extraction of a large number of interdependent circuit parameters. The identification of such relationships is crucial to the understanding of how individual variables affect circuit performance. In particular, if a user it attempting to locate sources of variation, isolate outliers, or investigate marginal distributions in a given data base.

The histogram and scattergram are two of the key exploratory data analysis tools used to investigate the graphical relationships between variables in a given data base. It is now possible, within ***SPAYN***, to rapidly produce multiple histogram and scattergram plots in a postscript file format. The postscript files can then be directly incorporated into technical reports, presentations or product documentation. This is accomplished utilising the ``Macro Commands'' facility, which is accessed from the ``Analysis'' menu of the main ***SPAYN*** window (Figure 3).

Consider, for example, a data base where the NMOS threshold voltage parameter VTO_N is of particular interest. The user would like to investigate the relationship between VTO_N and the other variables in the data base. An examination of the correlation matrix will reveal which parameters are correlated with VTO_N, this can then be confirmed graphically by generating multiple scattergram plots of those sets of parameters using the ``Macro Commands'' option. Alternatively, suppose a user needs a graphical verification that all of the parameters in a given data base can be modelled with a Gaussian distribution. Rather than producing plots individually, using the ***SPAYN*** histogram window, the ``Macro Commands'' option can be employed to automatically generate any number of plots in a postscript file format, ready for comparison. In both of these examples the user has the flexibility to select the destination directory. All of the plots are uniquely identified with a plot type label, the system PID (***SPAYN*** process identification number) and a plot number tag.

References

[1] "Realistic Statistical Worst-Case Simulations of VLSI Circuits" by M. Bolt, M. Rocchi and J. Engel. IEEE Transactions on Semiconductor Manufacturing Vol. 4. August 1991.

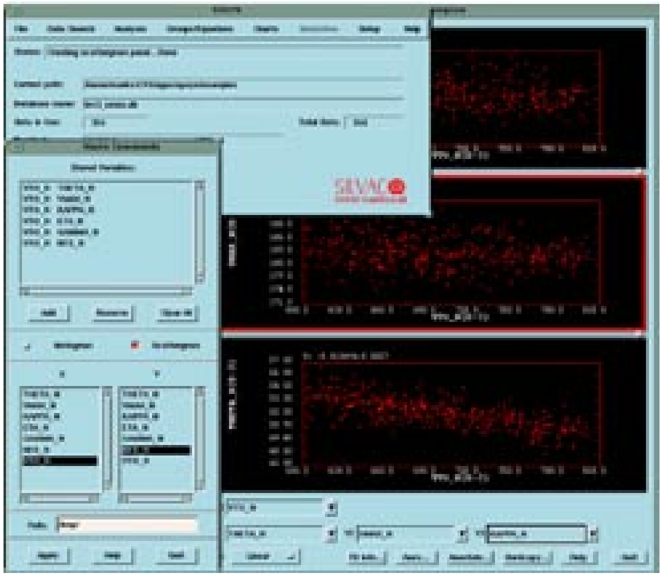


Figure 3. Macro command window for generating multiple histograms and scattergrams.

Calendar of Events

October

1
2
3
4 EOS/ESD - Reno, NV
5 EOS/ESD - Reno, NV
6 SOI - Stuart, FL
7 SOI - Stuart, FL
8 SOI - Stuart, FL
9
10
11
12 ISCS - Nara, Japan
13 ISCS - Nara, Japan
14 ISCS - Nara, Japan
15 ISCS - Nara, Japan
16 ISCS - Nara, Japan
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31

November

1 GaAs IC - Atlanta, CA
2 GaAs IC - Atlanta, CA
3 GaAs IC - Atlanta, CA
4 GaAs IC - Atlanta, CA
5
6
7
8 ICCAD - San Jose, CA
9 ICCAD - San Jose, CA
10 ICCAD - San Jose, CA
11 ICCAD - San Jose, CA
12
13
14
15
16
17
18 W/S Tokyo - Celebrity
19 W/S Tokyo - Discovery
20 W/S Tokyo - UTMOST III
21
22
23
24
25
26 Happy Thanksgiving
27
28
29
30

Bulletin Board

Silvaco Japan Expands

Silvaco's Japan office has signed a new lease that will effectively double the office space. To accommodate the recent rapid growth (almost 30 employees), and to accommodate for 1999 expansion (expecting to double the staff) the new premises will provide for maintaining efficient engineering support and sales operation.

Introducing Scholar

Silvaco is completing the finishing touches on its own schematic capture program scheduled for release in January 1999. ***Scholar*** will be used for tight integration with ***SmartSpice.*** Given its hierarchical nature and ability to handle unlimited number of components, it is expected that ***Scholar*** will quickly become very popular.

Silvaco Present at Int'l SOI Conference

Silvaco will demonstrate the power of its tools at the International SOI Conference in Stuart, Florida this October. Eric Verett from the Austin team and Micheal Ridinger from our Boston office are bringing the message of Silvaco's superior technology for the physics and modeling of SOI devices to the forefront of the silicon-on-insulator field.

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

The Simulation Standard, circulation 17,000 Vol. 9, No. 10, October 1998 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) 341-7220 (Japan), or your nearest Silvaco distributor.

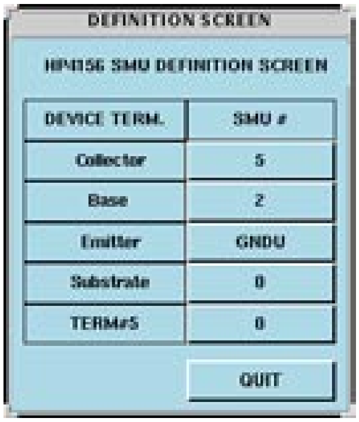
Hints, Tips and Solutions

Mustfa Tanner, Applications and Support Engineer

Q. How can I make high current or voltage measurements using HP4155/56 and UTMOST?

A. The HP4155/56 DC Analyzer can be configured with SMU and Pulse Generator expander box frame called HP41501A. The expander box can have different measurement unit options. The option unit for the expander box include GNDU (GrouND Unit), PGU (Pulse Generator Unit), MPSMU (Mid Power Source Monitoring Unit) and HPSMU (High Power Source Monitoring Unit).

In order to measure a device with currents higher than 100mA, the expander box with HPSMU and GNDU unit is required. The Power limit of the HPSMU is 20W (up to 200V or 1A) The Mid Power SMUs which are part of the main 4155/56 frame have power limit of 2W (up to 100V or 100mA). The HPSMU and GNDU units have both force and sense terminals. The GNDU unit can sink currents up to 1A and must be used with HPSMU for high current measurements.

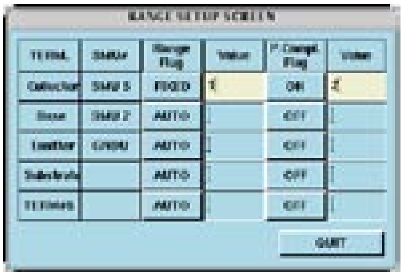


DEFINITION SCREEN

HP4156 SMU DEFINITION SCREEN

DEVICE TERM.	SMU #
Collector	5
Base	2
Emitter	GNDU
Substrate	0
TERM#5	0

Figure 1. ***UTMOST*** SMU configuration screen for high current Bipolar device measurement.



RANGE SETUP SCREEN

TERM.	SMU#	Range Flag	Value	P Compl. Flag	Value
Collector	SMU 5	FIXED	1	ON	2
Base	SMU 2	AUTO		OFF	
Emitter	GNDU	AUTO		OFF	
Substrate		AUTO		OFF	
TERM#5		AUTO		OFF	

Figure 2. Measurement Range and Power Compliance definition screen.

In order to measure a bipolar device with collector current greater than 100mA the following hardware configuration is needed:

HPSMU Force connect to COLLECTOR
HPSMU Sense connect to COLLECTOR (optional)
MPSMU connect to BASE
GNDU Force connect to EMITTER
GNDU Sense connect to EMITTER

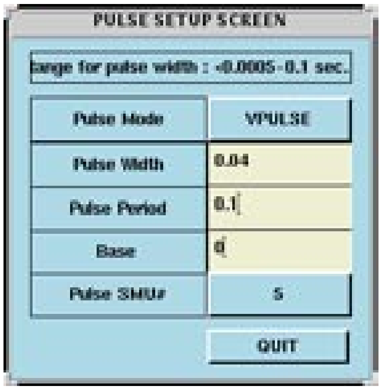
UTMOST SMU configuration screen supports the expander box unit configurations. The HP4155/56 main frame contains 4 MPSMUs therefore the 5th SMU is the HPSMU. The selection of "0" in the SMU configuration screen corresponds to "unused SMU". Figure 1 demonstrates the ***UTMOST*** SMU configuration for the high current bipolar device measurement.

Q. How can I set the measurement range and power compliance of HP4155/56 using *UTMOST*?

A. ***UTMOST*** supports automatic/fixed range definition and power compliance setting of HP4155/56. Both settings are ***UTMOST*** routine dependent and can be set separately for each routine. The "Range Setup" button is located in each routine's measurement setup screen (Figure 2).The default settings are: Automatic Range and Power Compliance Off. For further information on Measurement Ranging and Power Compliance please refer to the "User's Dictionary Reference of 4155/56".

Q. Is it possible to make pulsed measurements using HP4155/56 and UTMOST?

A. Yes, each SMU in 4155/56 can be used as Voltage or Current pulse source. (MPSMUs and HPSMUs). There can be only one SMU pulsed for a given configuration. In order to pulse a SMU the Stimulus Mode of HP4155/56 in the Hardware Configuration screen should be set to "Pulse 1". The selection of which SMU to pulse and type of pulsing (VPULSE or IPULSE) is made in the "Pulse Setup" screen for each routine (Figure 3.) The "Pulse Setup" button is located in the Measurement Setup Screen of each routine.



PULSE SETUP SCREEN

Range for pulse width : -0.0005-0.1 sec.

Pulse Mode	VPULSE
Pulse Width	0.04
Pulse Period	0.1
Base	0
Pulse SMU#	5

Figure 3. Pulse Setup Screen for HP4155/56.

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***

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

Join the Winning Team!

Silvaco Wants You!

- ● Process and Device Application Engineers
- ● SPICE Application Engineers
- ● CAD Applications Engineers
- ● Software Developers

fax your resume to:
408-496-6080, or
e-mail to:
jobs@*silvaco.com*

Opportunities worldwide for apps engineers: Santa Clara, Phoenix, Austin, Boston, Tokyo, Guildford, Munich, Grenoble, Seoul, Hsinchu. Opportunities for developers at our California headquarters.

SILVACO
INTERNATIONAL

USA Headquarters:

Silvaco International
4701 Patrick Henry Drive, Bldg. 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*