Simulation Standard

TCAD Driven CAD

A Journal for Circuit Simulation and SPICE Modeling Engineers

SmartLib: Product-independent SPICE Model Library

Introduction

Commercial SPICE vendors have traditionally supplied their SPICE simulators by combining the separate functional elements of the simulation process (the parser, the solver and the SPICE models themselves) into a monolithic executable program. This has presented a serious support problem, as most reported bugs are model-related. The typical support cycle takes months from the initial reporting of a bug to the final delivery of the solution: the bug must first be fixed, then that fix must be integrated into the simulator itself, where the fix is tested, along with any other fixes which may happen to have been incorporated at the time, before being re-released in its entirety and shipped to the affected customers; on the customer side, this new executable will now typically be tested again, before it is ultimately installed as a replacement for the previous release.

currently provides more than twenty different non-linear device models, covering MOSFET, bipolar, diode, MESFET, SOI, TFT and HBT technologies.

Silvaco's SPICE models have historically been an integral part of the **SmartSpice** source code, and all model development has therefore occurred primarily within **SmartSpice. UTMOST, SPAYN** and **ATLAS** provide their own implementations of some – but not al – of these models, using equations originally copied from their **SmartSpice** sources, then maintained separately from those contained within **SmartSpice** itself. Unfortunately, as a result of this procedure, the development of these models has, from time to time, lagged behind that of the "reference" models implemented directly in **SmartSpice.**

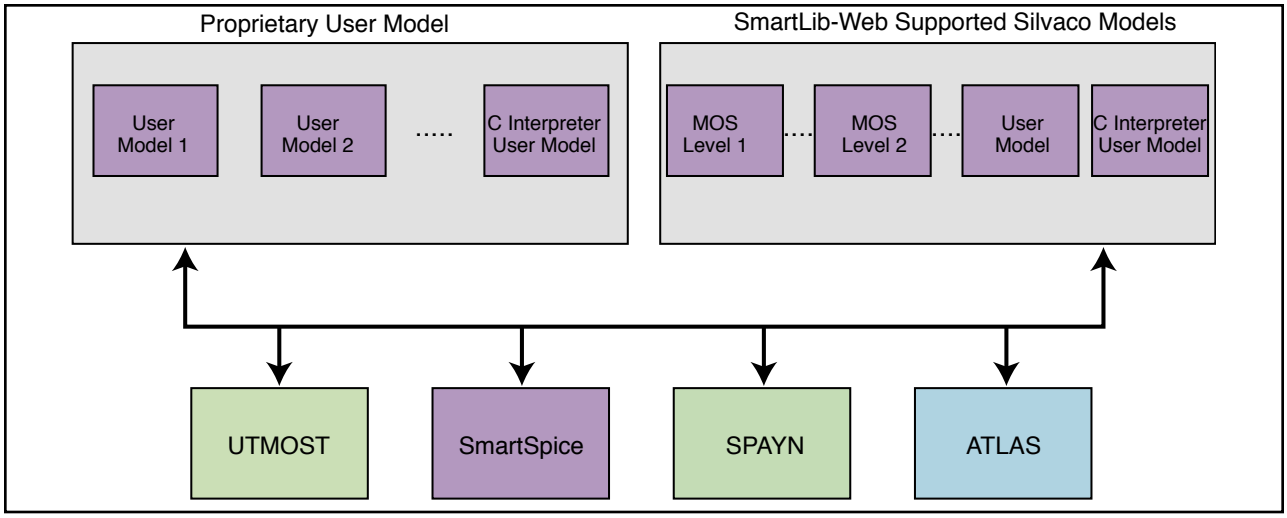


Figure 1. SmartLib architecture.

Continued on page 2....

The evaluation of device-specific SPICE models is a fundamental part of several Silvaco products. It occurs most notably, of course, in **SmartSpice,** but it is also a crucial component of **UTMOST** (in both simulation and interactive "rubberbanding" modes), of the forthcoming model validation product **Validate,** and also of **SPAYN** and **ATLAS** (in mixed device/circuit mode). **SmartSpice**

INSIDE

SILVACO International

This article will describe work, currently nearing completion at Silvaco, which is aimed at eliminating this source of incompatibility, and the associated developmental overheads, through the creation of an independent library of SPICE models, to be known as "**SmartLib**", which will be shared by all Silvaco products. The models which comprise this library will be individually maintained online, through the World Wide Web.

Model Evaluation

A SPICE model consists of a set of parameterized, non-linear equations which allow the calculation of current (and conductance), as a function of voltage, for each node in a particular semiconductor device. **SmartSpice** models provide two fundamental subroutines: the first (the "temperature" function) is responsible for the initialization of the appropriate device parameters, and for the calculation of intermediate temperature- and geometry-dependent variables; the second (the "load" function) evaluates the model equations for the particular device, returning the resulting currents and conductances.

However, this is not the end of the story: while the SPICE model returns currents as a function of voltages, what is invariably required in circuit simulation is actually a set of node voltages which satisfy the particular model equations for a fixed set of currents. To obtain such a solution, all Silvaco products use a variation of the Newton-Raphson root-finding algorithm, which uses the conductances returned by each iteration of the model's load function to converge on a satisfactory set of voltages.

Architecture

As already discussed, with the SPICE models built into **SmartSpice,** model development and support is necessarily a time-consuming process, and can only be achieved by the release of a succession of new **SmartSpice** executables. Furthermore, with the model equations in products such as **UTMOST** and **SPAYN** maintained separately from each other and, most critically, from those built into to **SmartSpice** itself, occasional disparities can occur between the solutions obtained from those models. In order to eliminate this source of potential incompatibility, along with the developmental overhead, the **SmartSpice** model equations have recently been extracted from **SmartSpice** and grouped into an independent collection of shared object libraries; this collection of libraries has been given the name "**SmartLib**". (Under the Solaris operating system–for example–the components of SmartLib have names such as "libbsim3.so", libvbic.so" and "libtft.so".) These libraries can now be dynamically linked with **SmartSpice** (and also with **UTMOST, SPAYN** and **ATLAS**–as will be discussed below); this applies not only to the UNIX platforms currently supported by Silvaco, but also to the PC platform, under both Windows NT and Linux.

Product Independence

Of course, the whole point of creating a standalone library of model equations is so that they can be used directly by all other Silvaco products in which they are needed; to this end, we have also created a functional interface to **SmartLib,** called the "**SmartLib Interface**" (SLI). This interface provides the ability to create, manipulate and destroy the **SmartLib** device and model structures required for model evaluation. By removing all SPICE model code from products such as **UTMOST**, **SPAYN, Validate** and **ATLAS,** we are now able to link these products directly with both **SmartLib** and **SLI,** thereby eliminating all former sources of incompatibility and ensuring that the a single set of model equations will now be evaluated by all Silvaco products. (Each product will, however, retain its own specialized solver, primarily for performance considerations, as these solvers are generally optimized to the needs of the particular product in which they are implemented.) A functional illustration of the **SmartLib** framework is shown in Figure 1.

World Wide Web Support

Functionally speaking, the decomposition of **SmartSpice** into separate model libraries will have no effect on the run-time **SmartSpice** executable: the **SmartLib** libraries are statically linked with the remaining **SmartSpice** object code to produce a finished product indistinguishable from that obtained when the models were built directly into **SmartSpice.** However, with the advent of **SmartLib,** the **SmartSpice** executable itself will be substantially smaller (the model equations typically occupy some 70% of a SPICE simulator), and need only be updated in order to take advantage of improvements in the functionality of the parser, solver or the graphical interface.

Most importantly, however, with **SmartSpice, UTMOST** and related products now dynamically linked with **SmartLib,** it will be possible to update individual model libraries independently of the products with which they are linked. This updating process will be greatly facilitated by use of the World Wide Web, where the **SmartLib** libraries corresponding to the latest versions of all Silvaco SPICE models will be available for immediate download and use in all **SmartLib**-compatible Silvaco products.

Proprietary Model Development

The above scenario is missing only one feature: by the nature of the dynamic linking process, the addition of new, previously unsupported SPICE models will necessarily entail the distribution of new versions of **SmartSpice, UTMOST** and other affected products with hooks for those new models built in. There is, however, a somewhat more sophisticated method of dynamic linking available to these products, which

Continued on page 8....

Parallel SmartSpice for PC

Introduction

Parallel SmartSpice version 1.5.4 is now available on multiple CPU Pentium PCs. The parallel version of **SmartSpice** allows single SPICE simulations to be executed on multiple CPUs. This provides significant run time speed improvements over traditional single CPU execution. (Figure 1)

Parallel SmartSpice for the first time now allows PC users to efficiently run large IC and PCB SPICE simulations in a fraction of the time required for single CPU simulation. This unique software solution provides designers with the ability to take advantage of the available multi-processor PCs.

Parallel SmartSpice for PCs is currently available running under the Linux operating system. Linux is a UNIX clone which runs on Pentium based PCs. It has all the features of a modern UNIX, including multiple CPU capabilities. An NT based version of **Parallel SmartSpice** is planned for release in Q2.

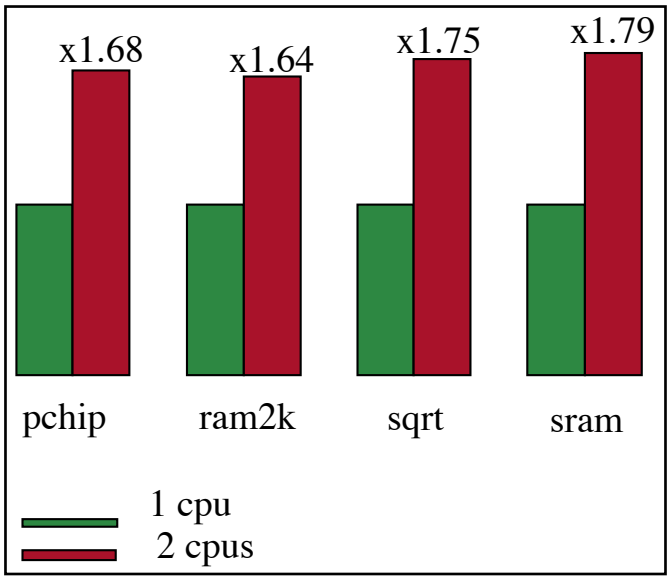


Figure 1. Speedup on two Pentium II CPU's for MCNC benchmark circuits

Parallel SPICE Implementation

The Linux implementation of **Parallel SmartSpice** release 1.5.4 offers the same functionality as **Parallel SmartSpice** running on UNIX machines. The models that are parallelized to take advantage of the availability of multiple CPUs on PC platforms are listed in Table 1. The sparse matrix factorization which occurs at every iteration is also parallelized.

Model	Version	Level
BSIM1	-	level 13
BSIM3	version 2	level 47
BSIM3	version 3.0	level 49,11,8
BSIM3	version 3.1	level 49,11,8
MOS123	-	level 1,2,3
BJT	-	-
TSMC	-	level 18,28

Table1. Parallel models in release 1.5.4

The accuracy of the sequential implementation for all of these models is maintained. **SmartSpice** will exhibit the same behavior irrespective of the number of processors that are used. In other words, the parallelization is entirely transparent as far as the user is concerned.

Parallel SmartSpice Performance

Figure 1 presents the run time speedup for a series of test circuits obtained using a dual processor Pentium II machine running at 300 Mhz. The results show an average speedup of 1.7 on two CPUs. The four circuits are part of the University of North Carolina MCNC benchmark suite and use Silvaco's implementation of the BSIM3v3 model (Level 8) . The size of the circuits range from 942 to 13880 devices. In further experiments, we found that a dual processor Pentium II running at 300Mhz is very competitive with the performance of leading UNIX workstations.

Previous Silvaco publications including *'Simulation Standard'* (Volume 8, Number 5, May 1997) and *'TCAD Driven CAD'* (Volume 8, Number 7, July 1997) discuss in detail the different steps of the parallelization process, and present performance results on UNIX platforms.

Conclusion

Efficiency and affordability are the key words that best describe **Parallel SmartSpice** for multi-processor PCs. This product uniquely offers a low cost / high speed circuit simulation solution that provides comparable performance, and preserves all of the industry leading functionality and convergence properties of the workstation version.

BSIM3v3.1 Intrinsic Capacitance Modeling in UTMOST

Introduction

The "INT.CAP" routine is now available in MOS technology of **UTMOST** as part of the capacitance measurement module. The "Intrinsic Cap." routine can be used to measure overlap caps, channel capacitance and total gate capacitance. The routine also includes the automatic extraction of **BSIM3V3** capacitance parameters and simulation/optimization capabilities using SPICE.

Hardware Configuration

The "INT.CAP" routine allows users to apply external DC bias to the MOS device during the CV measurements. **UTMOST** controls the DC analyzer and the LCR meter concurrently. The user should select the proper DC analyzer and the LCR meter from the Hardware Configuration screen (Figure 1) and complete the terminal connections before starting the "INT.CAP" routine.

It is recommended to have a large W MOS device (W > 500 μm) for the "INT. CAP" routine measurements. If the W is not large enough the parasitic caps associated with the setup can effect the accuracy of the measurements (Even after the calibration is performed).

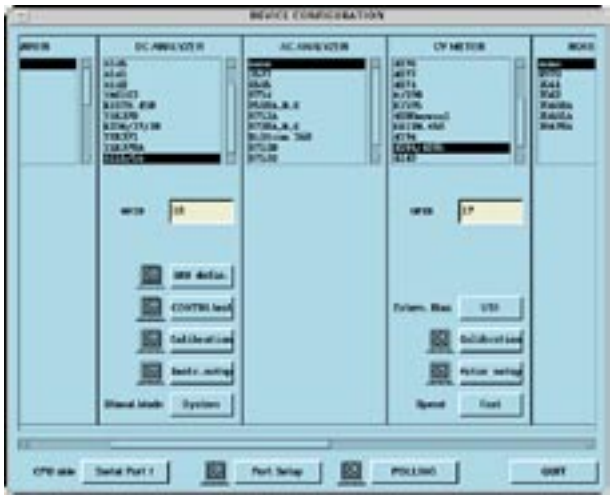


Figure 1. Hardware configuration screen.

Measurement Setup

The "INT.CAP" routine is in the "CAP" analysis section of the UTMOST MOS technology. Open the "Setup and Result" screen and toggle the "Routine Pointer"button until it reads "CAP". The "INT.CAP" routine will appear as routine #67 (Figure 2).

In order to set the bias conditions select the "INT.CAP" routine and press the "Set Measurement" button. This will open the "Capacitance Measurement Screen". (Figure 3.)

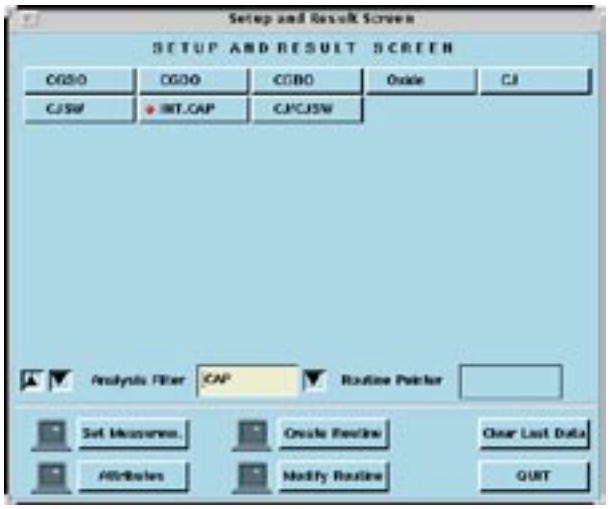


Figure 2. INT.CAP routine in setup screen.

The "c_start_bias", c_stop_bias" and "c_step_bias" measurement variables are belong to the LCR meter's internal DC sweep and they are shared by all types of capacitance measurements performed in the "INTCAP" routine. The remaining DC bias conditions belong to the DC analyzer and they are specific to the capacitance measurement type.

Capacitance measurements offered by the "INT.CAP" routine:

CGS : Gate to source capacitance
CGD : Gate to drain capacitance
CGC : Gate to channel capacitance
CGB : Gate to bulk capacitance
CGG : Total gate capacitance

Figure 3. Capacitance Measurement Screen for the "INT.CAP" routine.

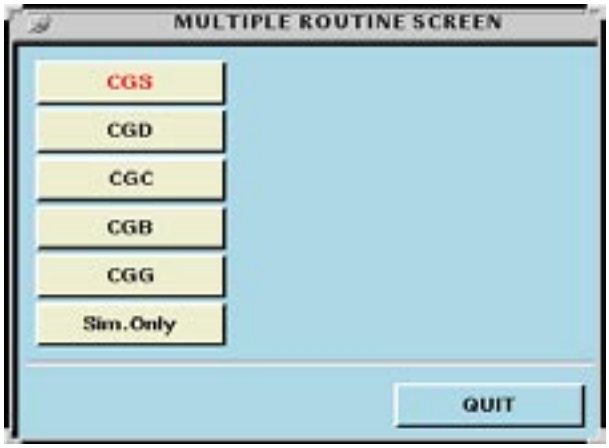


Figure 4 “Multiple Routine Screen” of the “INT.CAP” routine.

In addition to the listed measurements there is an intrinsic capacitance simulation capability which simulates the nine intrinsic capacitances together.

Selection of the type of capacitance measurement can be done by pressing the “Multiple Select.” button in the “Routine Control Screen”.This will open the “Multiple Routine Screen” and the capacitance measurement types will be listed (Figure 4). The buttons in the”Multiple Routine Screen” can be toggled. The selected measurements types will have the red color.

Bias Settings:

Below is the description of the measurement variables used in the “INT.CAP”routine setup screen (Figure1.) :

VD_start_cgs : VD (drain voltage) start bias (supplied by DC analyzer) for CGS measurements. Also used as the VD value for “Simulation Only” Intrinsic caps.

VD_step_cgs : VD (drain voltage) step bias (supplied by DC analyzer) for CGS measurements.

#of_step_cgs : Number of VD steps for CGS measurements.

VS_start_cgd : VS (source voltage) start bias (supplied by DC analyzer) for CGD measurements.

VS_step_cgd : VS (source voltage) step bias (supplied by DC analyzer) for CGD measurements.

#of_step_cgd : Number of VS steps for CGD measurements.

VB_start_cgc : VB (bulk voltage) start bias (supplied by DC analyzer) for CGC measurements.

VB_step_cgc : VB (bulk voltage) step bias (supplied by DC analyzer) for CGC measurements.#of_step_cgc : Number of VB steps for CGC measurements.

VD_start_cgb : VD (drain voltage) start bias (supplied by DC analyzer) for CGB measurements.

VD_step_cgb : VD (drain voltage) step bias (supplied by DC analyzer) for CGB measurements.

#of_step_cgb : Number of VD steps for CGB measurements.

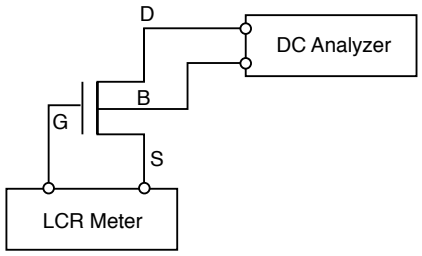
c_start_bias : Starting DC bias voltage for the LCR meter sweep.

c_stop_bias : Stop DC bias voltage for the LCR meter sweep.

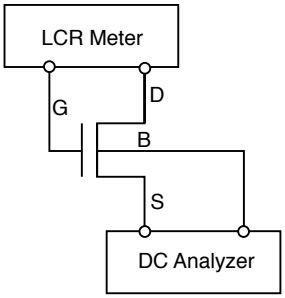
c_step_bias : Step bias voltage for the LCR meter sweep.

Terminal Connections for the “INT.CAP” Measurements

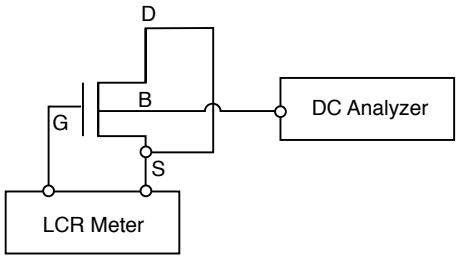
The “INT.CAP” routine allows users to measure various MOS capacitances while the device is under DC bias and conducting current. This requires the DC analyzer and the LCR meter to work jointly. The terminal connections also become an important issue due to the AC and DC current paths which define the total measured capacitance. The following are the terminal connections which are needed for the different types of capacitance measurements available in the “INT. CAP” routine.



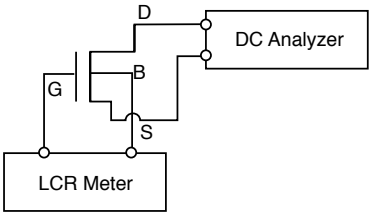
CGS measurement connection



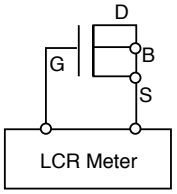
CGD measurement connection



CGC measurement connection



CGB measurement connection



CGG measurement connection

Measurement Results

The measurement curves for CGS, CGD, CGC, CGB, CGG and simulation curves for all nine intrinsic caps are presented below. The CGS, CGD, CGC, CGB and CGG can be simulated and optimized using SPICE. The "Sim Only" selection provides the simulation of 9 intrinsic caps of: CBDB, CBGB, CBSB, CDDB, CDGB, CDSB, CGDB,CGGB and CGSB.

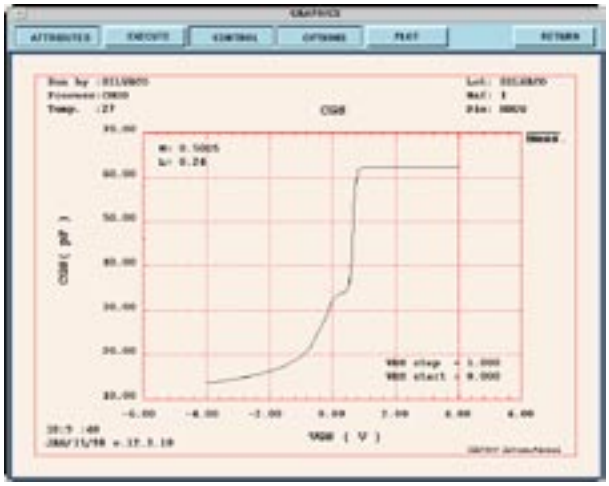


Figure 5. CGS data

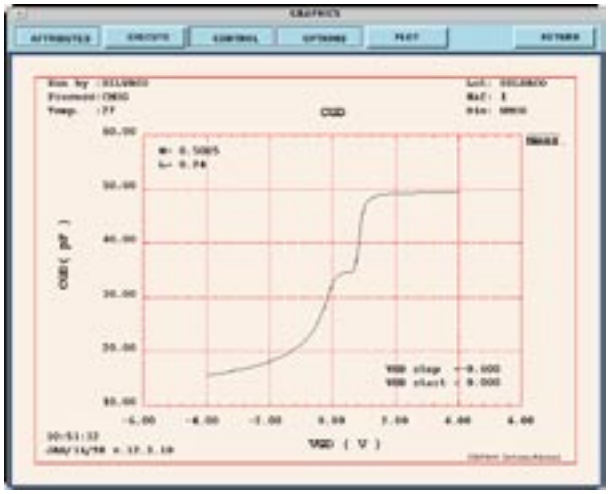


Figure 6. CGD data

Figure 7. CGC data.

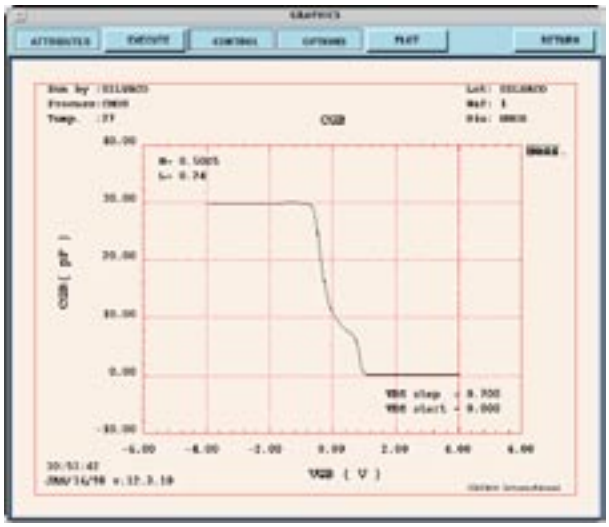


Figure 8. CGB data

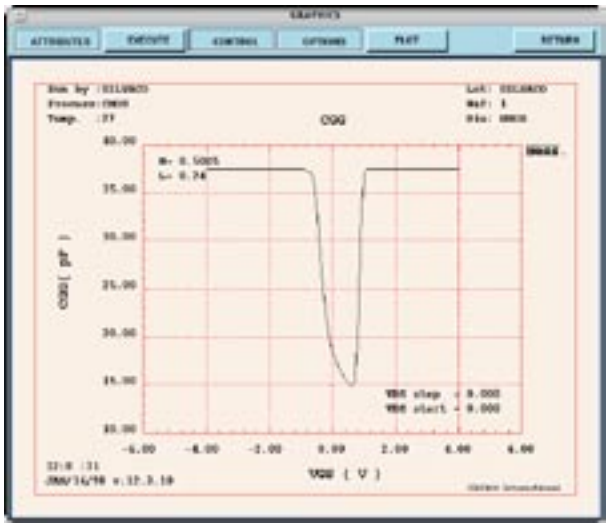


Figure 9. CGG data.

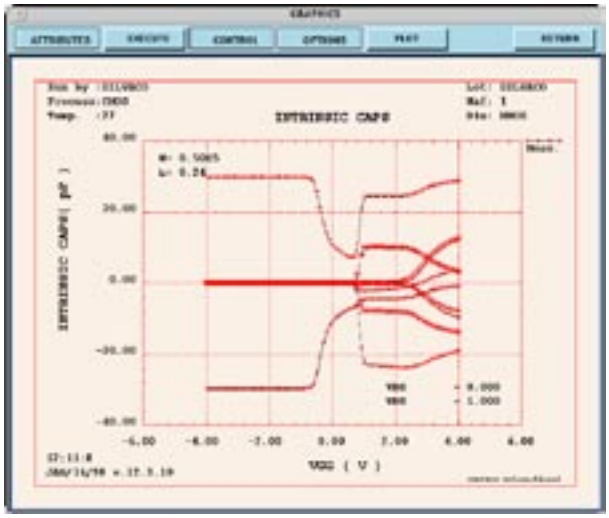


Figure 10. "Sim. Only" data. Simulation results of: CBDB, CBGB, CBSB, CDDB, CDGB, CDSB, CGDB, CGGB and CGSB.

Continued on page 11...

A New C-Interpreter

Introduction

The **C-Interpreter** is a general purpose tool that allows users to prototype new model equations. A new **C-Interpreter** with a more sophisticated architecture and advanced features will be available in a number of Silvaco's products including **SmartSpice, ATLAS** and **UTMOST.** The following describes a number of its advanced features and summarizes the benefits it will bring to end users.

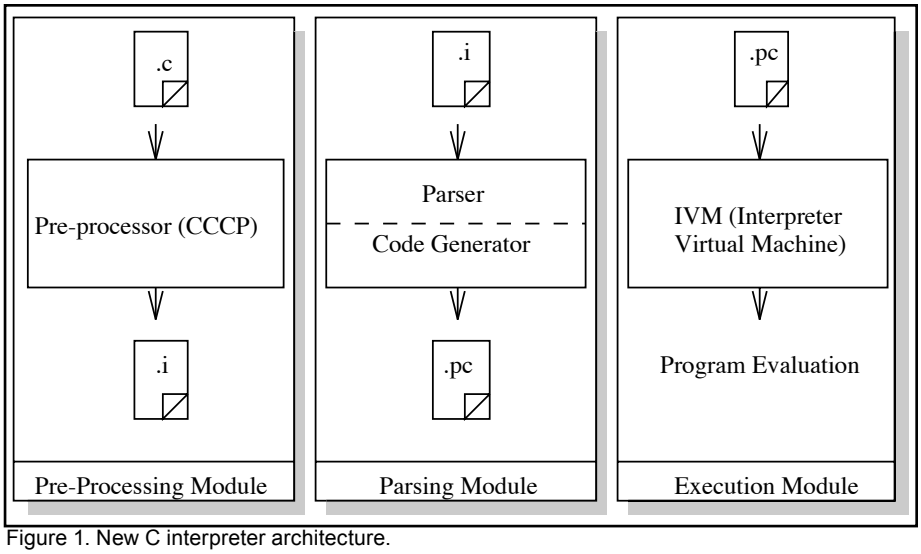


Figure 1. New C interpreter architecture.

New C Interpreter Architecture and Features

The new **C-Interpreter's** architecture, shown in Figure 1., uses state of the art compiler techniques in order to compile a program into an internal machine code representation which is then interpreted by a software virtual machine. This technique gives the new C-Interpreter far superior execution times over its predecessor.

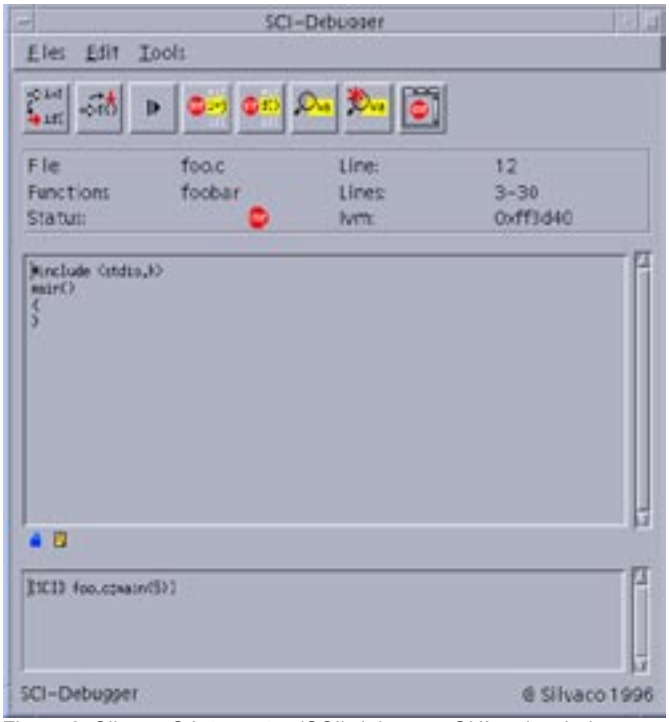


Figure 2. Silvaco C Interpreter (SCI) debugger GUI main window.

Not only is the new **C-Interpreter** ANSI/ISO compliant, but it also supports a number of C++ extensions, additional extensions for concurrency and can be embedded within a multi-threaded application allowing multiple virtual machines to interpret more that one program at a time. It also has a new debugger GUI providing user friendly access to all features. Figure 2 shows the new **C-Interpreter**'s debugger GUI main window.

SmartSpice - C interpreter Model Development

As an example, Figure 3 shows the relationship between the **C-Interpreter** debugging environment and **SmartSpice** for model development.

The debugging environment shields the user from unnecessary code allowing a focus solely on model development. The integration with **SmartSpice** allows the results of model modifications to be seen instantly within the **SmartSpice** simulation as a whole. The new environment preserves the current ability for users to compile model code with the native host compiler once debugged for full model performance optimization.

Performance Summary

The new **C-Interpreter** architecture gives much faster execution speeds allowing faster simulations of interpreted code and faster debug and test cycles. In **SmartSpice,** early benchmarks have shown a significant improvement over the existing **C-Interpreter**. Circuits

containing single devices including BJTS, JFETS, MESFETS and MOSFETS have shown total analysis time ratio improvements varying from 3.8 to 7.12. Comparisons with native compiled code Sun Solaris 2.5.1 C compiler have indicated the interpreter to be only four to five times slower which is a much improved performance increase over the current **SmartSpice C-Interpreter**.

Conclusion

In summary the new **C-Interpreter** currently under development is very powerful and offers a number of sophisticated features. This translates to a number of benefits to the end user. These benefits include much faster interpreted execution times, faster debug/test cycles, improved debugging environment features and cross tool transparency between SILVACO products.

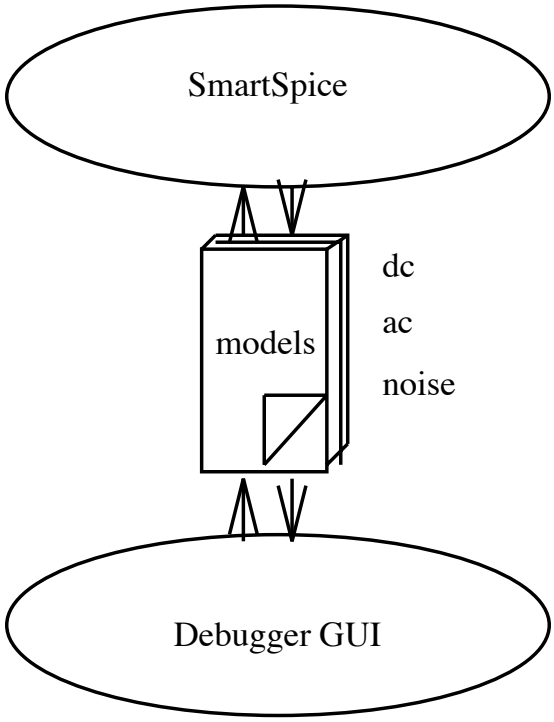


Figure 3. SmartSpice - C interpreter debugging relationship.

...continued from page 2

will enable arbitrary shared object libraries to be opened and closed at will. This implementation will enable the run-time linking of **SmartSpice, UTMOST** and all other affect products with arbitrary model libraries; this will allow the development of proprietary user models which, on compilation with Silvaco-supplied header files, will be functionally indistinguishable from the **SmartLib** components supplied by Silvaco. Customers will thus be free to develop, and use their own models on an equal footing with those already implemented inside **SmartLib.**

Summary

To summarize, the advantages of **SmartLib** over the former, monolithic architecture are:

- *Efficiency:* all SPICE models are maintained within a single repository (**SmartLib**), eliminating the need for separate development of models in different products.
- *Compatibility:* since the same SPICE models will now be linked with all affected products, there will be no possibility of incompatibility between different products.
- *Maintainability:* new SPICE models, and updates to existing models, can be downloaded from Silvaco's Web site as soon as they are released.
- *Flexibility:* user-developed proprietary models can be easily implemented alongside Silvaco-supplied **SmartLib** models, and dynamically linked with **SmartSpice** through the same mechanism.
- *Speed:* there is some preliminary evidence that SPICE models evaluated from **UTMOST** through **SmartLib** are slightly faster than those previously implemented directly in **UTMOST,** and it is likely that this improvement will extend to similarly affected products.

The next release of both **SmartSpice** and **UTMOST** will be a **SmartLib** implementation. Other Silvaco products will follow.

RPI* Polysilicon and Amorphous TFT Implemented in SmartSpice

fect in polysilicon TFTs and affect threshold voltage, the field-effect mobility, the subthreshold/leakage currents and the frequency dispersion of the capacitance in both a-Si:H and polysilicon TFTs.

Accurately modeling these effects becomes increasingly important as the device dimensions are scaled down and design margins are reduced. TFT models that satisfy the above requirements have been developed by the Center for Integrated Electronics and Electronics Manufacturing at Rensselaer Polytechnic Institute[1,2]. The physically based analytical TFT models are fully scalable " with device geometry Model parameters can be easily extracted using **UTMOST.**

Description

In the crystalline silicon device, the knee of the log(Id)-Vgs curve is very sharp, and the Id-Vgs characteristic quickly becomes linear. The result is that the threshold voltage (Vt) and the on voltage (Von) of the device are nearly equal. In contrast, the Von is larger than Vt in the non-crystalline transistor due to the bandgap states. This results in a comparatively small current at threshold and the current transition from the the exponential to the linear regine is much more gradual. In addition to the leakage, subthreshold, and above threshold conduction regimes, the polysilicon model also covers the kink regime. The kink effect in polysilicon TFT occurs at large biases when the TFT is in saturation. Electron-hole pairs generated through impact ionization recombine in the channel via boundary trap states and the feedback is caused by the recombination. The feedback effect does not occur and therefore the current increase is not nearly as dramatic in crystalline silicon transistors. The non-crystalline effects have been currently modeled in the Rensselaer TFT models.

The Rensselaer TFTs models have the following features:

- Universal charge control model, which guarantees stability and conversion
- AC models accurately reproduces Cgc frequency dispersion
- Temperature dependence included
- Automatic scaling of model parameters to accurately model a wide range of device geometries
- A minimum number of physically based parameters that can easily be extracted from experimental data and related back to the fabrication steps
- Verified using a wide variety of TFT structures from many foundries

In addition, the a-Si:H TFT model uses a unified Ids equation which covers all regimes of transistor operation. Therefore, the derived conductance gds and gm have no discontinuity problems through the regime boundaries.

Both of these models are implemented in **SmartSpice** and are be available on Unix or PC.

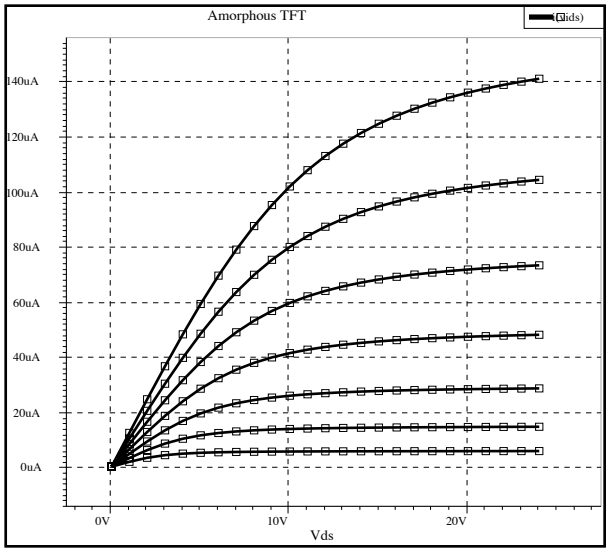


Figure 1. A typival Ids vs VDS simulation data using the RPI TFT model.

Acknowledgements

The TFT models described in this article were developed by the Semiconductor Devices Research Group at RPI, under the direction of Professor Michael Shur. Silvaco would like to acknowledge RPI's contribution of model code and documentation, which greatly facilitated implementation of these models within **SmartSpice.**

References

1. Michael S. Shur, H. C. Slade, et al, "Modeling and scaling of a-Si: H and Poly-Si Thin film transistors", MRS Spring Meeting, San Francisco, March 31 - April 4, 1997.
2. Michael S. Shur, Mark D. Jacunski, et al.,"SPICE models for amorphous silicon and polysilicon thin film transistors", Elec.Chem. Soc. Proc., Vol 96-23, p242-259, 1996.

* RPI - Rensselaer Polytechnic Institute, Troy New York

Calendar of Events

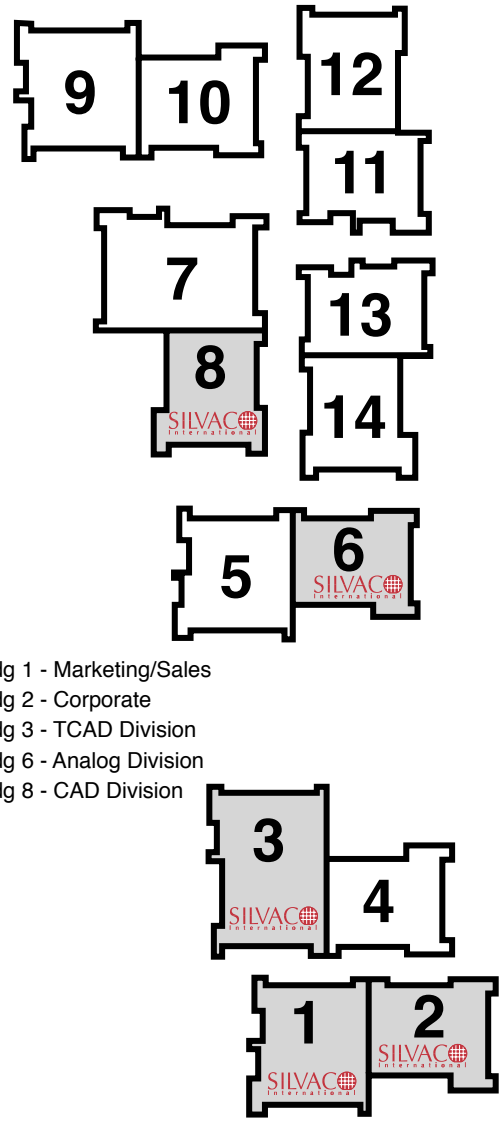
October

1	
2	
3	
4	
5	
6	Intl. SOI Conference
7	Intl. SOI Conference
	European Symposium on Reliability
8	Intl. SOI Conference
	European Symposium on Reliability
9	Intl. SOI Conference
	European Symposium on Reliability
10	
11	
12	IEEE Intl Rel. W/S
13	IEEE Intl Rel. W/S
14	IEEE Intl Rel. W/S
15	IEEE Intl Rel. W/S
16	IEEE Intl Rel. W/S
17	
18	
19	
20	
21	
22	
23	
24	SmartSpice W/S - Tokyo
25	
26	
27	
28	
29	
30	
31	

November

1	
2	
3	
4	
5	
6	
7	
8	
9	
10	ICCAD'97, Santa Clara
11	ICCAD'97, Santa Clara
12	ICCAD'97, Santa Clara
13	
14	
15	
16	
17	
18	
19	
20	
21	TCAD W/S, Tokyo
22	
23	
24	
25	
26	ATHENA W/S, Taiwan
27	UTMOST W/S, Taiwan
28	
29	
30	

Silvaco Headquarters Expands!!



Bldg 1 - Marketing/Sales
Bldg 2 - Corporate
Bldg 3 - TCAD Division
Bldg 6 - Analog Division
Bldg 8 - CAD Division

Patrick Henry Drive

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

...continued from page 6

Parameter Extraction

The CGS, CGD, CGC and CGG measurements have the capability of extracting the BSIM3v3 capacitance parameters. For parameter extraction press the "Fit" button from the "Options" menu. For example from the CGD measured data, parameters CGSO, CGDL and CKAPPA can be extracted and displayed in the graphics screen (Figure 11). The extracted parameters are also copied into the parameters screen.

The CGG measured data can be used to extract oxide thickness (TOX). After the measurement is completed, press the "Fit" button and the extracted value of TOX will be displayed on the graphics screen (Figure 12.)

Figure 11. CGD measured data and extracted parameters CGDO, CGDL and CKAPPA.

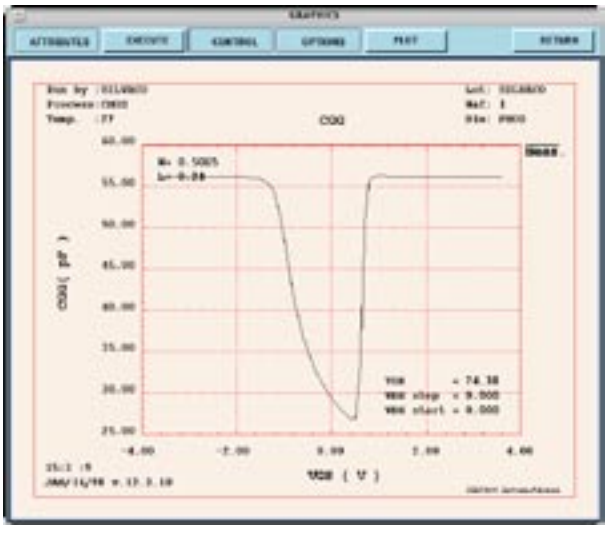


Figure 12. CGG measured data and extracted parameter TOX.

Simulation and Optimization

The "INT.CAP" routine allows **UTMOST** users to simulate and optimize the BSIM3v3 capacitance parameters with SmartSpice. The optimization of the model with SPICE using the measured data will improve the accuracy of the capacitance model. The improvement on the AC model will translate into better fits for the ring oscillator circuit simulations.

In the following example the CGC (channel capacitance) was measured when Drain and Source terminals were shorted and the Substrate was connected to DC ground. The parameters CGDO, CGSO, CGDL, CGSL and CKAPPA were optimized in the "accumulation region" and parameters CLC, CLE and DLC were optimized in the "inversion region" (Figure 13.)

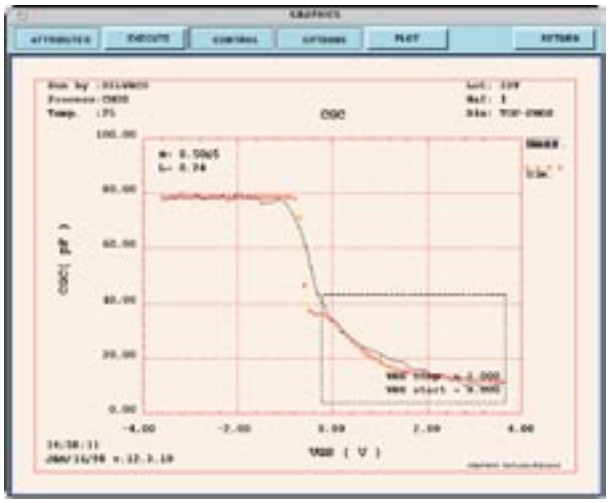


Figure 13. PMOS device CGC measured and optimized simulation results. The optimization box includes the accumulation region.

Conclusion

The addition of the "INT.CAP" routine enables **UTMOST** users to measure any combination of MOS capacitances. An independent bias from a DC analyzer can also be applied to the MOS device while the CV measurements are performed. The "INT.CAP" routine utilizes **SmartSpice** for capacitance simulations. Using SPICE in capacitance simulation and optimization will enhance the accuracy of the AC modeling with **BSIM3v3.**

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

http:://www.silvaco.com

Join the Winning Team!

Silvaco Wants You!

- SPICE Application Engineers
- TCAD Application Engineers
- CAD Application Engineers
- Software Developers

fax your resume to:
408-496-6080, or
e-mail to:
personnel@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

4701 Patrick Henry Drive, Building 2
Santa Clara, CA 95054

Telephone: (408) 567-1000
Fax: (408) 496-6080
URL: http://www.silvaco.com