Simulation Standard

TCAD Driven CAD

A Journal for Circuit Simulation and SPICE Modeling Engineers

Temperature Effects in *SmartSpice* LEVEL=6 Ferroelectric Capacitance Model From Ramtron International Corporation

Introduction

Implementation of a new ferroelectric capacitance model from Ramtron International Corporation into ***SmartSpice*** was first described in the April 2002 issue of SILVACO *Simulation Standard*. This model utilizes a new concept of double distribution of domain reversal voltages. The temperature effects were not detailed in the previous article. This application note discusses the implementation of the temperature effects and updates the device syntax.

Features

The updated ferroelectric model is invoked in ***SmartSpice*** by setting LEVEL=6 in the capacitance model card. This model differs from its predecessor (model FCAP LEVEL=5) in that the ferroelectric capacitor is regarded as a non-linear capacitor with Polarization-Voltage (P-V) hysteresis loop, however the biases in the ferroelectric materials are reversed at reversal voltages with double distributions. As compared with the LEVEL=5 model, the LEVEL=6 model demonstrates the following improvements: more accurate simulations of ferroelectric hysteresis loops and sub-loops, improved voltage pulse responses, faster simulation speed (up to six times faster), and added temperature dependence.

Ferroelectric Capacitor Element

SmartSpice device statement syntax:

NOTE: Device syntax is updated from that published in the April 2002 *Simulation Standard*

```
Cxxx n1 n2 mname <V0=val> <P0=val> <A=val>
```

Cxxx:	Capacitor element name; must begin with "C."
n1, n2:	Positive and negative terminal node names, respectively.
mname:	Model name (must be ferroelectric capacitor model).
V0:	Initial voltage across the ferroelectric capacitor (V). Default is 0.0.
P0:	Initial polarization (mC/cm2) from positive node (n1) to negative node (n2). Default is 0.0.
A (AREA):	Capacitor area (m2). Default is 1.0E-12.

One difference to note is that the LEVEL=5 ferroelectric capacitor uses the unit-less parameter, IP, for initial polarization while the LEVEL=6 ferroelectric capacitor uses the initial polarization parameter, P0, in $\mu C/cm^2$. The input for the capacitor size in the LEVEL=6 model is the capacitor area, and can be specified using either A or AREA in m^2.

Temperature Effects in LEVEL=6 Ferroelectric Capacitance Model

SmartSpice model syntax:

```
.MODEL mname C LEVEL=6 <parameter=value> …
```

mname:	Model name.
C:	Specifies capacitance model.
LEVEL=6:	FRMC model.
parameter:	Any model parameter name.

The model parameters were documented in the April 2002 *Simulation Standard* article with the exception of the temperature parameters. Table 1 shows a complete list of the model parameters including the updated default parameter values as well as the temperature coefficient parameters.

Continued on page 2....

INSIDE

SILVACO INTERNATIONAL

LEVEL=6 FERROELECTRIC CAPACITOR MODEL PARAMETERS			
Parameter	**Description**	**Units**	**Default**
VMAX	Maximum voltage	V	3.898e+00
PMAX	Maximum polarization at VMAX	$\mu C/cm^2$	3.787e+01
KPMAX	1st order temperature coefficient of PMAX		-8.803e-03
K2PMAX	2nd order temperature coefficient of PMAX	V	-8.616e-05
VSAT	Saturated model point	V	2.452e+00
VCR	Minimum voltage for domain switching		4.903e-01
AS1	Curve fitting parameter		1.348e+00
BS1	Curve fitting parameter		1.174e+00
CS1	Curve fitting parameter		1.634e+00
AS2	Curve fitting parameter		-1.085e+00
BS2	Curve fitting parameter		1.444e+00
CS2	Curve fitting parameter		1.859e+00
DS0	Curve fitting parameter		5.513e-01
AU1	Curve fitting parameter		1.457e+00
BU1	Curve fitting parameter		1.432e+00
CU1	Curve fitting parameter		1.377e+00
AU2	Curve fitting parameter		-1.109e+00
BU2	Curve fitting parameter		1.820e+00
CU2	Curve fitting parameter		1.574e+00
DU0	Curve fitting parameter		5.695e-01
LEVEL=6 FERROELECTRIC CAPACITOR MODEL PARAMETERS: Temperature Coefficients			
KAS1	1st order temperature coefficient of AS1		-3.178e-03
KBS1	1st order temperature coefficient of BS1		-1.970e-03
KCS1	1st order temperature coefficient of CS1		-4.562e-03
KAS2	1st order temperature coefficient of AS2		3.074e-03
KBS2	1st order temperature coefficient of BS2		-2.539e-03
KCS2	1st order temperature coefficient of CS2		-4.343e-03
KDS0	1st order temperature coefficient of DS0		-1.869e-04
KAU1	1st order temperature coefficient of AU1		-4.090e-03
KBU1	1st order temperature coefficient of BU1		-2.641e-03
KCU1	1st order temperature coefficient of CU1		-4.944e-03
KAU2	1st order temperature coefficient of AU2		4.495e-03
KBU2	1st order temperature coefficient of BU2		-4.626e-03
KCU2	1st order temperature coefficient of CU2		-4.637e-03
KDU0	1st order temperature coefficient of DU0		-4.621e-04
K2AS1	2nd order temperature coefficient of AS1		3.423e-05
K2BS1	2nd order temperature coefficient of BS1		-3.860e-06
K2CS1	2nd order temperature coefficient of CS1		2.176e-05
K2AS2	2nd order temperature coefficient of AS2		-3.725e-05
K2BS2	2nd order temperature coefficient of BS2		-8.596e-06
K2CS2	2nd order temperature coefficient of CS2		1.689e-05
K2DS0	2nd order temperature coefficient of DS0		1.575e-06
K2AU1	2nd order temperature coefficient of AU1		4.337e-05
K2BU1	2nd order temperature coefficient of BU1		-4.328e-06
K2CU1	2nd order temperature coefficient of CU1		1.882e-05
K2AU2	2nd order temperature coefficient of AU2		-5.218e-05
K2BU2	2nd order temperature coefficient of BU2		-4.039e-06
K2CU2	2nd order temperature coefficient of CU2		1.254e-05
K2DU0	2nd order temperature coefficient of DU0		2.099e-06

Table 1. Complete list of the model parameters including the updated default parameter values as well as the temperature coefficient parameters. NOTE: All blank cells indicate a number with no units.

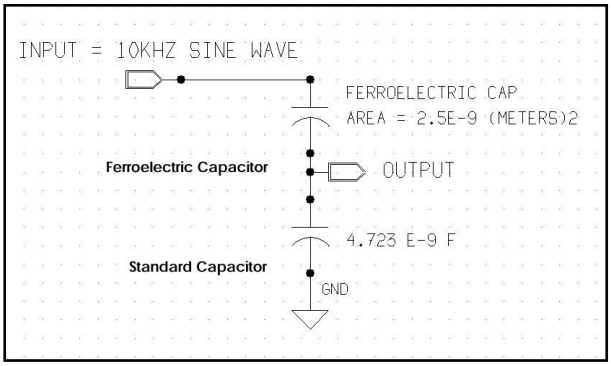

Figure 1. Test circuit for ferroelectric capacitor simulations. (Sawyer-Tower circuit)

Simulation Results

The improved accuracy of the LEVEL=6 model compared to the LEVEL=5 model was documented in the April 2002 *Simulation Standard* article. Here we show comparison of the simulated vs. measured results at three different temperatures to validate the temperature modeling. The test circuit used is shown in Figure 1.

Temperature effects for the ferroelectric capacitor model are calculated in the following form for all model parameters:

$$AS1_{eff} = AS1 + KAS1 * (TEMP - TNOM) + K2AS1 * (TEMP - TNOM)^2$$

Figures 2, 3, and 4 show measured and simulated hysteresis loops at 3V and 5V, for temperatures of 27ºC, 60ºC, and 90ºC, respectively. The input stimulus for these results was a sinusoidal voltage waveform at 10 KHz. This shows very close agreement between simulation and measurement at temperature.

Conclusions

The new ferroelectric capacitor model implemented in ***SmartSpice*** as LEVEL=6, has been shown to accurately simulate hysteresis loops at various temperatures. This model based on double distributions of domain reversal voltages has shown advantages over the LEVEL=5 model as described in the April 2002 *Simulation Standard* article. In this article, we have presented details of the implementation of temperature effects, comparison of simulation results to measured data, and updated the device syntax.

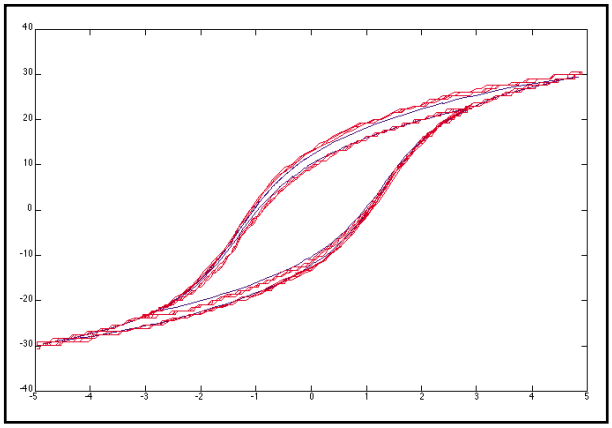

Figure 2. Measured (red) and simulated (blue) hysteresis loops at 27ºC. [*Courtesy of Ramtron International Corporation*]

SILVACO gratefully acknowledges Ramtron International Corporation for the development of the ferroelectric capacitance model and the data presented here.

References

1. "Polarization Reversal Kinetics in Ferroelectric Liquid Crystals", Proceedings of the Sixth International Meeting on Ferroelectricity, Kobe 1985, Yoshihiro Ishibashi, Japanese Journal of Applied Physics,vol. 24 Suppl. 24-2, 126 (1985)
2. *Simulation Standard* Volume 12, Silvaco International, April 2002
3. *ATLAS Users Manual*, Silvaco International. December 2002.

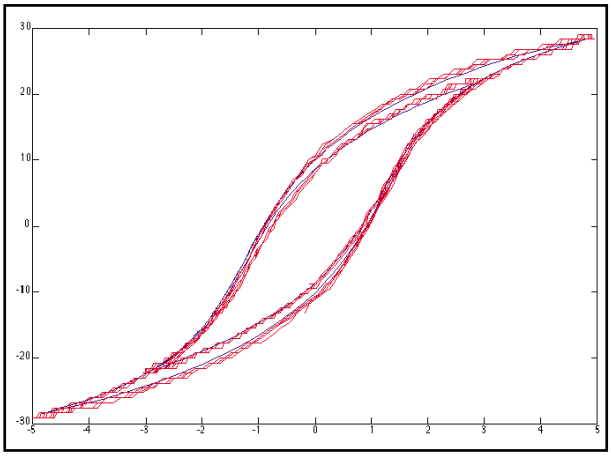

Figure 3. Measured (red) and simulated (blue) hysteresis loops at 60ºC. [*Courtesy of Ramtron International Corporation*]

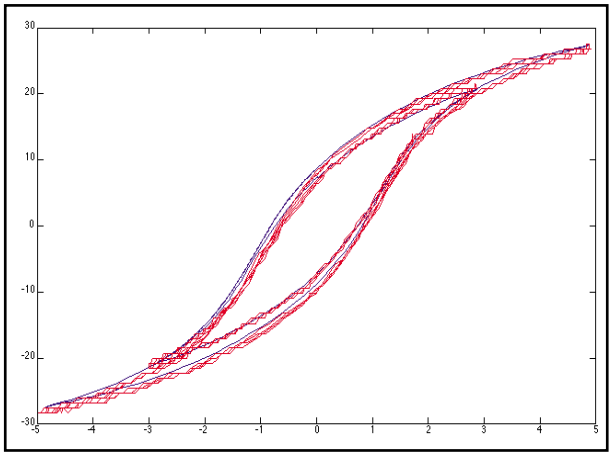

Figure 4. Measured (red) and simulated (blue) hysteresis loops at 90ºC. [*Courtesy of Ramtron International Corporation*]

UFSOI Version 7.0 (UFPDB Version 2.0) Model Released in *SmartSpice*

Introduction

Version 7.0 of the University of Florida Silicon-On-Insulator (UFSOI), released in 2002, is now available with Silvaco SmartSpice by setting LEVEL to 21. ***SmartSpice*** uses version 7 by default, but versions 4.5, 5.0, 5.0 rev 1.0, and 5.0 rev 6.0 are still available through resetting the VERSION and REVISION parameters.

Model Features (prior to version 7.0) [1]

The UFSOI is a physical, charge-based, and process-based SOI MOSFET model that has evolved from the basic modeling of thin-film devices. The charge modeling is physically linked to channel-current modeling. All terminal charges and their derivatives are continuous for all bias conditions. The UFSOI model includes both NFD/PDB (Partially Depleted Bulk) and FD (Fully Depleted) models, which are set with the BODY parameter in version 6.0 and later. In previous versions, FD and NFD models were selected with NFDMOD parameter:

BODY	model
0	FD
1	NFD/PD
2	Bulk-Si

Table 1. BODY values (version 6.0 and later).

NFDMOD	model
0	FD
1	NFD

Table 2. NFDMOD values (previous versions).

The other parameters are process-based and are directly related to the device structure and material properties. This model is charge-based in order to ensure charge conservation and proper accounting for all transcapacitances. The model is extended to account for an accumulated charge in the body that can drive a floating-body mode dynamic bipolar effect in all regions of operation. There is optional accounting for LDD and LDS.

Additional, recently-added physical effects are:

- polysilicon-gate depletion
- inversion-carrier energy quantization
- GIDL/GISL (version 6.0 and later)
- narrow-width effects
- junction tunneling (NFD/PD model)
- RSCE/halo effects (NFD/PD model)
- carrier-velocity overshoot (version 5.0 and later)
- Account for hot-carrier effects on the channel thermal noise (version 5.0 and later)

Temperature dependence and accounting for self-heating are implemented in both models, without the need for additional parameters.

A Unified PD/Bulk-Si Model

In Version 6.0, the NFD model is expanded to serve as a unified **process-based compact model** for PD SOI and Bulk-Si MOSFETs that use a single, small set of process-related parameters. This feature, enabled by the process-basis of the model, **allows direct performance comparison of the two mainstream CMOS technologies** without ambiguities in device structure [2].

Figure 1 shows the predicted current-voltage characteristics of PD/SOI and bulk-Si MOSFETs. This reflects the unified feature of UFPDB, and stresses the higher currents and kinks in the PD/SOI characteristics that result from the floating-body effect. The sub-threshold characteristics of both models are compared in Figure 2. The same set of model parameters was used for both devices.

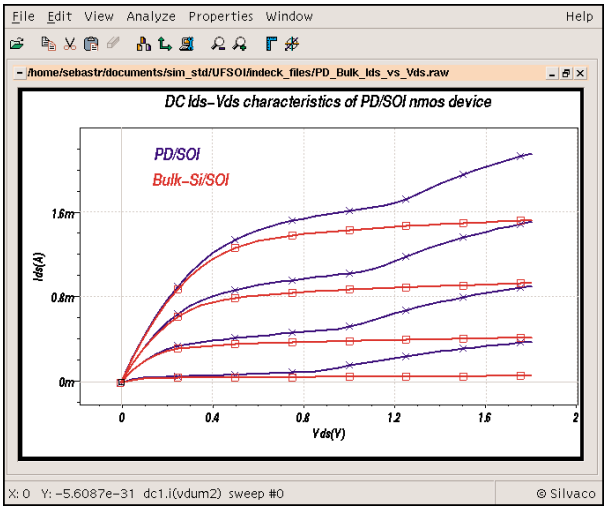

Figure 1. Ids vs Vds characteristics in PD and Bulk-Si mode of the UFPDB model.

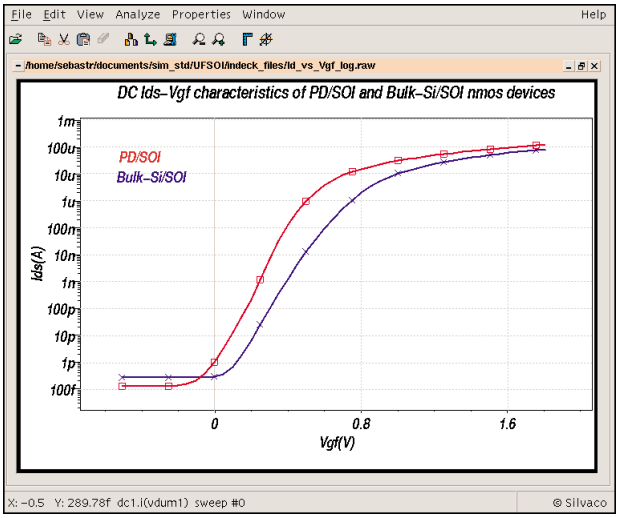

Figure 2. Subthreshold characteristics in PD/NFD and Bulk-Si mode.

The bulk-Si option internally ties the substrate to the body, sets the back-gate oxide thickness to near-zero, and updates the substrate density to reflect the bulk-Si doping. In the Bulk-Si mode, the substrate becomes the well, and this charge reflects the source/drain-junction areal capacitance. In this case, the device has four terminals and the command line in the netlist, like any other MOSFET device, becomes:

```
Mxxx ND NGF NS <NB> ...
```

Improvements in version 7.0 [3]

Version 7.0 includes the upgraded UFPDB version 2.0 model that includes gate-body tunneling current (I_{tung} in Figure 3), exchange energy for inversion carriers, a strained Si/SiGe-channel option, and allowance for arbitrary gate dielectric (KD parameter). With these upgrades, the model is applicable to CMOS devices scaled to the bulk-Si limit (Lgate ~ 50 nm).

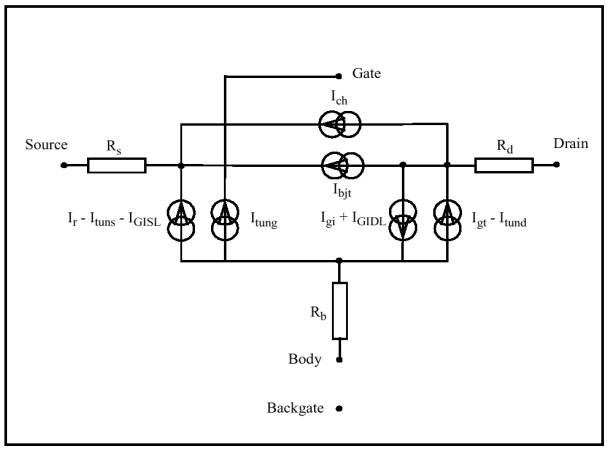

Figure 3. The static UFSOI model.

The current sources in the static model are :

- I_{ch} : channel current
- I_{bjt} : parasitic bipolar effect
- I_r: recombination current
- I_{gi} : impact ionization current
- I_{gt} : thermal generation current
- I_{tund} : drain/body tunneling current
- I_{GIDL} : Gate Induced Drain Leakage current
- I_{GISL} : Gate Induced Source Leakage current
- I_{tuns} : source/body tunneling current
- I_{tung} : gate/body tunneling current

Gate-body tunneling current is only important to floating-body PD/SOI devices (Figure 4), so the feature is selectable only in NFD/PD model. Indirect valence band-conduction components are included that depend on both bandgap narrowing and channel quantization. The modeling is valid for all inversion and accumulation conditions and is applicable only for N+ polysilicon gates on NMOS devices, and P+ polysilicon gates on PMOS devices. This effect is taken into account by setting MOX parameter different from 0 (Table 3)

Parameter	Description	Default	Typical
MOX	Electron effective mass (normalized to free electron mass) in gate dielectric (0 for no gate-body tunneling current)	0.0	0.36

Table 3. Gate-Body tunneling current related parameter.

The following parameters are used for smoothing the Igb (gate-bulk) current. Default values are typically used.

Parameter	Description	Default
SVBE	Smoothing parameter for gate-body current in inversion region (0 for no current)	13.5
SCBE	Smoothing parameter for gate-body cu rent in accumulation region (0 for no current)	0.04 (NMOS) 0.045 (PMOS)
FFACT	Factor for gate-body current smoothing in accumulation region	0.5 (NMOS) 0.8 (PMOS)

Table 4. Gate-Body tunneling current smoothing parameters.

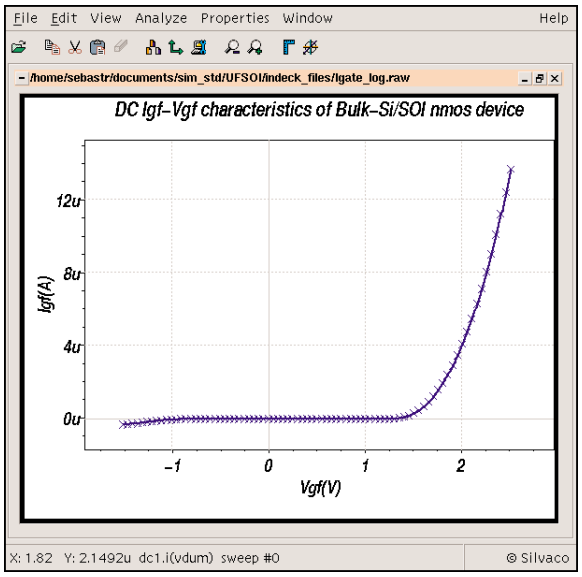

Figure 4. Itung vs Vgate characteristic.

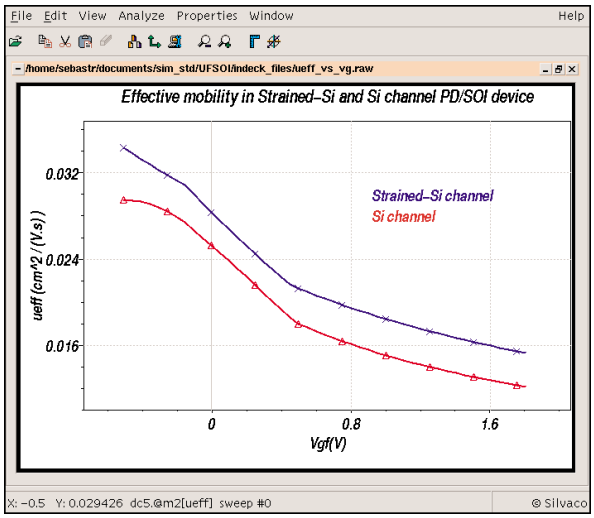

Figure 5. Field-effect mobility with strained- and unstrained-Si channel on a L = 60nm device.

Strained-Si channel property is a consequence of the pseudomorphic nature of a Si film on a SiGe layer. The deposited Si layer conformd atom-by-atom to the underlying SiGe lattice pattern and results in an enhanced-mobility device (Figure 5). The strained-Si option in UFSOI Version 8.0 is selected by setting the DEG parameter to the correct value and is deselected by setting DEG to zero (default value, cf table 5). This effect has an impact on other parameters (VO, VSAT, U0 and NBL to control threshold voltage) that must be evaluated in order to compare strained- and unstrained-Si devices.

A comparison of strained- and unstrained-Si channel SOI ring-oscillator with equivalent modelcards shows a period of 100ps for strained-Si, and 140ps for unstrained-Si (Figure 6).

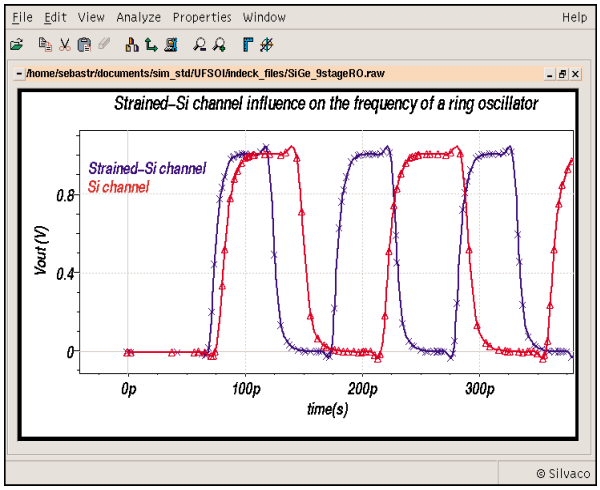

Figure 6. strained- and unstrained-Si devices ring oscillator.

Speed improvements are implemented in version 7.0. All of these improvements make the UFSOI model even closer to the real behavior of the device and make the simulation faster and more accurate. Strong and weak inversion threshold voltages are no longer iteratively calculated.

Parameter	Description	Default	Typical
DEG	Bandgap narrowing in strained Si channel (0 for no SiGe)	0.0 eV	0.1 eV

Table 5. Strained-Si channel related parameter.

References

[1] UFSOI MOSFET MODELS (Vers. 6.0), User's Guide, SOI group, University of Florida, June 2001.

[2] J.G. Fossum, "A Unified Process-Based Compact Model for Scaled PD/SOI and Bulk-Si MOSFETs", MSM 2002, Univ. Florida, april 2002.

[3] UFSOI MOSFET MODELS (Vers. 7.0), User's Guide, SOI group, University of Florida, June 2002.

Managing *SmartSpice / SmartView* Simulation Output Raw Files

Storing Waveform Data to Disk Instead of Holding in Ram

Silvaco ***SmartSpice*** simulation results are typically stored in RAM. Since transient simulations of large circuits often exceed 1G, a large swap-space partition is required prior to simulation. Constant disk access may dramatically decrease simulation speed, therefore shifting some or all of the load to the system's memory helps to alleviate this problem.

Unfortunately, over-reliance on memory may also produce negative results and slow simulation time. In addition, it may be impossible to view intermediate results if a simulation runs for several days in RAM. These intermediate results help the user to decide whether or not to abort a simulation early.

To adjust the amount of simulation data held in memory, use this SPICE deck option:

```
.option rawpts= <val> post
```

<val> is the maximum number of simulation time points that are used during a simulation. As soon as this number is reached and th epost option active, this block of data is written to disk. A very small <val> results in the frequent saving to disk and use of less memory.

It is important to strike a balance between using minimum memory and speed. Since each system is different, this "sweet spot" is found through experimentation. Start a test simulation that uses relatively small value and increase the value gradually. A suggested starting value is between 100-1000.

Since a circuit with many saved nodes generates more results, a smaller <val> is required. A small-<val> save cycle is also useful to those who need marching waveforms of the simulations results. The user can monitor the simulation results while the simulation is running and can abort at any time.

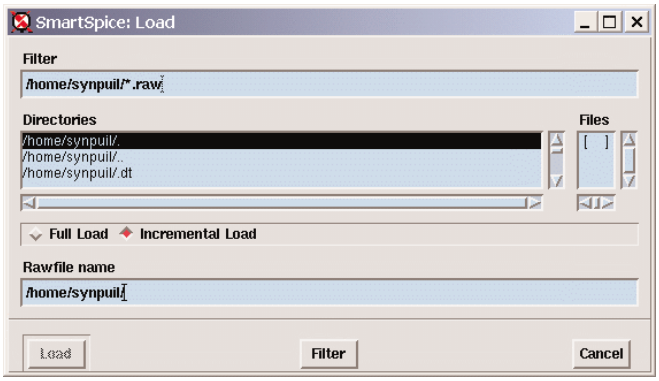

Figure 2.

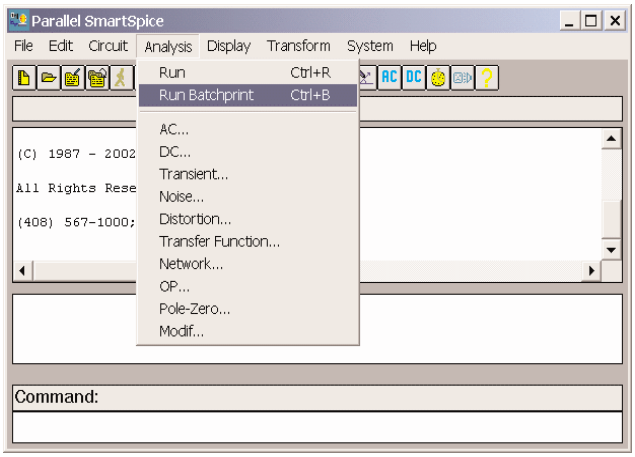

Figure 1.

Saving *SmartSpice* Results for Interactive Mode

While in interactive mode, ***SmartSpice*** simulation results are not saved to disk when Analysis -> Run is executed. This follows the original Berkley implementation and is true even if there is an option post specified in the input deck or the 'r' flag has been used to invoke ***SmartSpice***. Use menu Analysis -> Run Batch Print to allow the possibility of saving the simulation output data.

Speeding up Loading of Raw File in *SmartSpice/SmartView*.

When using ***SmartSpice / SmartView*** to load a large (1-2G) simulation results file, activating the incremental loading of waveforms is recommended. This dramatically reduces the time needed in order to see the first waveform from approximately 45 minutes to 5 minutes for a 1G file. This also allows the user to load a data file bigger than 2 Gigs which is normally the limit on a 32 bit operating system.

In the UNIX version of ***SmartSpice***, incremental loading is activated with the dialog shown in Figure 2.

Incremental load is activated in the Windows version through the Properties -> Drawing Options menu, shown in Figure 3.

Click the pull-down menu next to Incremental File Loading and select "On" as shown in Figure 4.

Continued on page 10....

Defining Voltage Controlled Oscillator in *SmartSpice*

1. Introduction

One of the methods to define a Voltage Controlled Oscillator (VCO) in a SPICE simulation is to make use of a voltage-controlled voltage source (E element). A typical VCO expression makes use of a sinusoidal function as shown in Formula 1.

$$Vol = V_0 + A \cdot \sin(k \cdot V_{control} \cdot t) \qquad (1)$$

When expression (1) is applied, it is easy to assume that the frequency of VCO is proportional to the absolute value of control voltage. Sometimes, however, simulations return unexpected results when $V_{control}$ is an inconstant function of time. An example is shown in Figure 1. The control voltage in this example is a piecewise linear function, and sinusoidal voltage in Figure 1(a) is the corresponding VCO.

The "E" element syntax can be found in the examples shown in the *SmartSpice User's Manual.*

This result is unexpected since Figure 1(a) is not symmetrical to Figure 1b. Intuitively, one expects the frequency in Figure 1(a) to reduce gradually after t = 50ns, but the simulation result in Figure 1(a) is correct. The rest of this article will show why the result in Figure 1a is correct, and how to use the ***SmartSpice*** E element to achieve an expected VCO response.

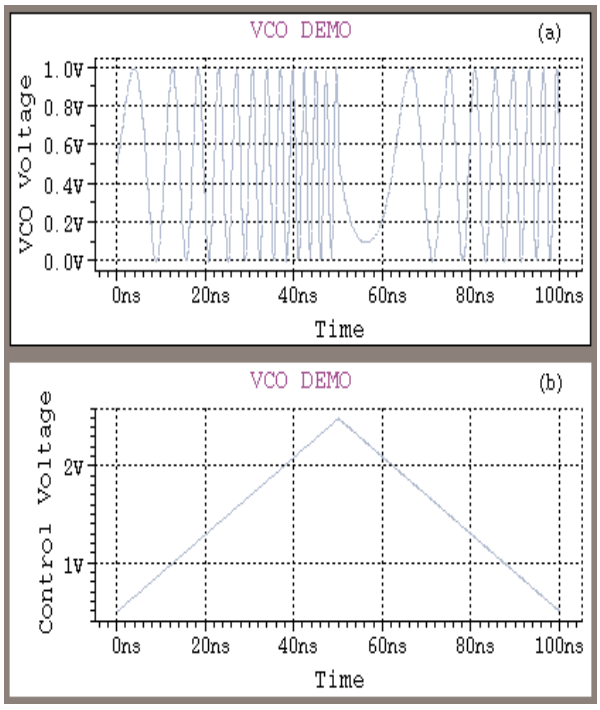

Figure1 An example of VCO frequency controlled by piecewise linear voltage

2. Problem Diagnosis

In order to explain the behavior of the VCO shown in Figure 1a, we rewrite Equation 1 in a more general form for a given time-varied control voltage V(t):

$$Vol = V_0 + A \cdot \sin(k \cdot v(t) \cdot t) \qquad (2)$$

The frequency in Equation 2 is not Kv(t), but rather is calculated based on the cyclic numbers at time t as shown in Equation 3:

$$N_{cycle}(t) = k \cdot v(t) \cdot t \qquad (3)$$

By definition, angular velocity or frequency at *t* can be calculated using:

$$f(t) = \frac{dN_{cycle}(t)}{dt} = k \cdot [v(t) + v'(t) \cdot t] \qquad (4)$$

Equation 4 gives the true frequency of a VCO defined by Equation 2. Obviously, KV(t) in Equation 2 is not always equal to the actual frequency, and it equals the frequency only if *V(t)* is a constant with respect to time. By applying Equation 4 to the control voltage displayed in Figure 1b, we find that the frequency of the VCO defined by Equation 1 is

$$f(t) = \begin{cases} 0.08kt + 0.05k & (0 \le t < 50 \text{ ns}) \\ k4.5 - 0.08kt & (50\text{ns} \le t \le 100 \text{ ns}) \end{cases} \qquad (5)$$

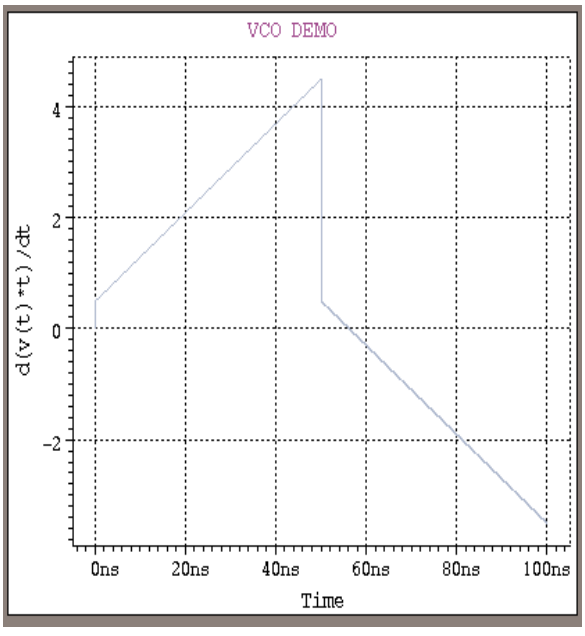

Figure 2. VCO frequency varies with time.

The result in Equation 5 is plotted in Figure 2. This plot indicates a dramatic drop in frequency at $t = 50$ ns that continues to decrease after 50 ns. From $t = 56.25$ ns the frequency increases along with the absolute value of f(t) increases. This proves the ***SmartSpice*** simulation result shown in Figure 1a.

3. A New Approach

For a time-variant control voltage in Figure 1 (b), using Equation 1 in E element definition does not result in a symmetrical frequency distribution over time. To achieve this, we construct function $w(t)$ in Equation 6, so that the frequency of $y(t)$ is always equal to the absolute value of a control voltage.

$$y(t) = \sin[w(t) \cdot t] \tag{6}$$

Clearly, $w(t)$ must satisfy the following relation,

$$w(t) + w(t) \cdot t = k \cdot v(t) \tag{7}$$

where $v(t)$ is the control voltage, and k is a coefficient used to convert voltage to frequency. Equation 7 is a linear differential equation of the first order, and its solution is

$$w(t) = \frac{1}{t} + \frac{k}{t} \int_0^t v(\tau) d\tau \tag{8}$$

Here, the initial condition is arbitrarily assumed in order to achieve a simpler form. Plugging Equation 8 into Equation 6, we obtain

$$y(t) = \sin\left[1 + k \int_0^t v(\tau) d\tau\right] \tag{9}$$

Equation 9 is the sinusoidal function with a frequency of $kv(t)$. The result in Equation 9 can be directly used to define a VCO, and the frequency of this VCO is proportional to the value of control voltage.

4. Implementation in *SmartSpice*

SmartSpice features the numerical integration function necessary in order to implement Equation 9. This function is denoted as integral(x) or s(x). Considering k is a large number, we may neglect 1 in Equation 9, and write Equation 9 in SmartSpice format:

$$y(t) = \sin(k \cdot s(v(t))) \tag{10}$$

Applying Formula 10 to the problem in Figure 1 obtains a correct result (Figure 3). The frequency is proportional to absolute value of control voltage.

Figure 4 is an similar example that uses a pulse control voltage. The input deck is listed in appendix.

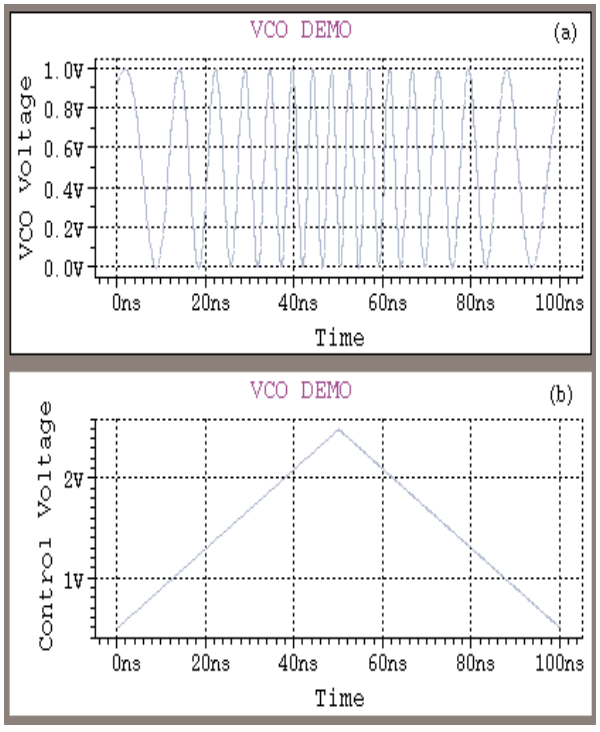

Figure 3. Simulated VCO frequency (a) controlled by piecewise linear voltage (b) using new method

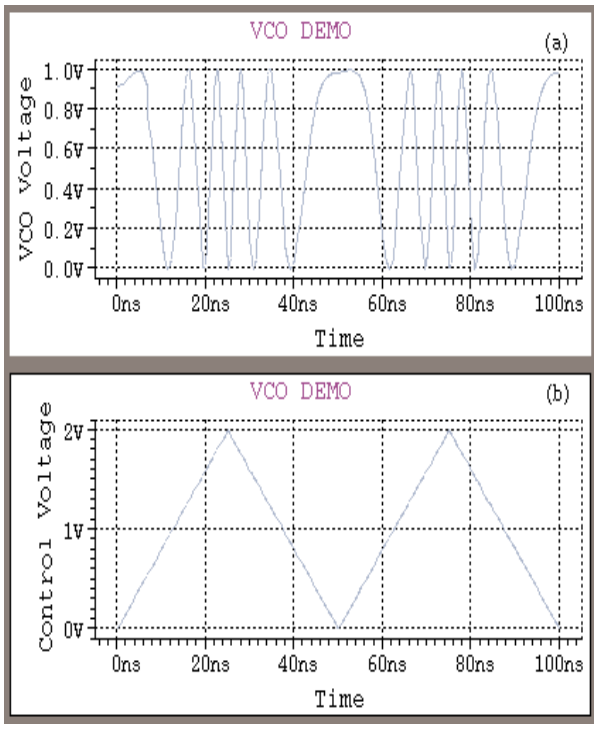

Figure 4. Simulated VCO frequency (a) controlled by pulse voltage (b) using new method

5. Summary

A conventional SmartSpice VCO definition $\sin(k*V_{control}*t)$ may generate an unexpected VCO frequency if the control voltage varies with respect to time. A general method is available for the use of E element to define a VCO that results in a frequency that is proportional to the absolute value of a control voltage:

$$Vol \propto \sin(k\int_0^t v(\tau)d\tau]$$

In ***SmartSpice*** it will be vol = $\sin(k.s(v(t)))$.

6. Appendix

A. Input deck for the example in Figure 1.

```
VCO DEMO

.OPTIONS post accurate
.options RMAX=2

Evco 2 0 vol='0.5+0.5*sin(6.28*1e8*V(1)*time)'

Vcontrol 1 0 pwl (0 0.5v 50n 2.5V 100n
0.5v)

R 1 0 10MEG

.TRAN .1n 100n

.END
```

B. Input deck for the example in Figure 3.

```
VCO DEMO

.OPTIONS post accurate
.options RMAX=2

Evco 2 0 vol='0.5+0.5*sin(6.28*1e8*sv(1)+1)'

Vcontrol 1 0 pwl (0 0.5v 50n 2.5V 100n
0.5v)

R 1 0 10MEG

.TRAN .1n 100n

.END
```

C. Input deck for the example in Figure 4.

```
VCO DEMO

.OPTIONS post accurate

Evco 2 0 vol='0.5+0.5*sin(6.28*1e8*sv(1)+1)'

Vcontrol 1 0 pulse (0 2v 0 25n 25n 0 50n)

R 1 0 10MEG

.TRAN .1n 100n

.END
```

...continued from page 7

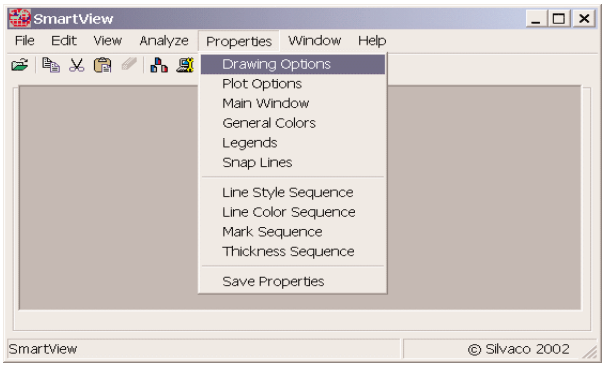

Figure 3.

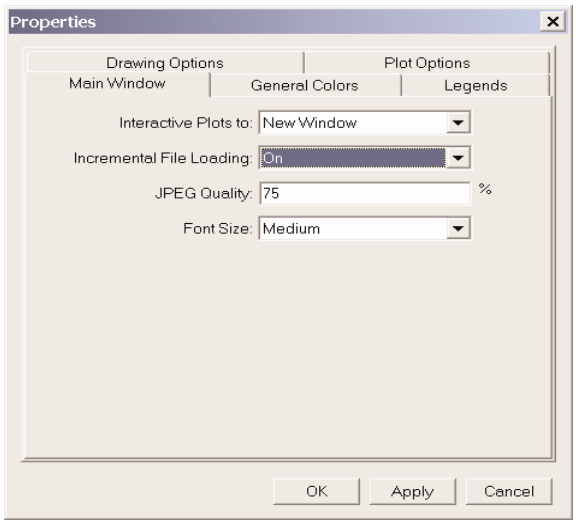

Figure 4.

Viewing Data Across Different Platforms

Data generated on one operating system is inherently different to data generated on another system (such as Linux or Solaris). ***SmartSpice*** has an option to make data file cross platform compatible. The following line needs to be in the input deck:

```
.option post=4
```

This creates the output data in XDR binary format. The data can therefore be generated by ***SmartSpice*** on say a Linux box and the waveforms viewed on a solaris box using ***SmartView***.

Calendar of Events

October

1 BCTM - Monterey, CA
2 Non-Stoichiometric IIIV Compounds - Monterey, CA
3 Non-Stoichiometric IIIV Compounds - Monterey, CA
4 Non-Stoichiometric IIIV Compounds - Monterey, CA
5
6
7 IEEE SOI Conf. - Williamsburg, VA
8 IEEE SOI Conf. - Williamsburg, VA
9 IEEE SOI Conf. - Williamsburg, VA
10 IEEE SOI Conf. - Williamsburg, VA
11
12
13
14
15
16
17
18
19
20
21 GaAs IC Symposium Monterey, CA
22 GaAs IC Symposium Monterey, CA
23 GaAs IC Symposium Monterey, CA
24
25
26
27
28
29
30 LCD/PDP 2002 - Japan
31 LCD/PDP 2002 - Japan

November

1 LCD/PDP 2002 - Japan
2
3
4
5
6
7
8
9
10 ICCAD - San Jose, CA
LEOS Annual Meeting-Scotland
11 ICCAD - San Jose, CA
LEOS Annual Meeting-Scotland
CS-MAX - San Jose, CA
12 ICCAD - San Jose, CA
LEOS Annual Meeting-Scotland
CS-MAX - San Jose, CA
13 ICCAD - San Jose, CA
LEOS Annual Meeting-Scotland
CS-MAX - San Jose, CA
14 ICCAD - San Jose, CA
LEOS Annual Meeting-Scotland
15
16
17
18 EDMO - Manchester, UK
19 EDMO - Manchester, UK
20
21
22
23
24
25
26
27
28
29
30

Bulletin Board

AustriaMicrosystems Supports SmartSpice Models at 0.35um

austriamicrosystems Business Unit Full Service Foundry and Silvaco International announced the support of the ***SmartSpice*** circuit simulator with the latest spice-parameters for austriamicrosystems 0.35um technology platform. The supported processes include mixed-signal 0.35um CMOS, which is fully TSMC CMOS-compatible, as well as 0.35um BiCMOS and by the end of this year 0.35um Silicon Germanium. austriamicrosystems AG, with headquarters in Unterpremstätten near Graz (Austria), is one of the world's leading designers and manufacturers of custom specific mixed signal IC´s. This company has 820 employees and offices in 14 countries worldwide.

Silvaco at ICCAD with Windows

Come see Silvaco at ICCAD 2002 November 11-14 at the Doubletree Hotel in San Jose showing a complete suite of circuit simulation and IC CAD tools on the Windows and Linux platforms.

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

Hints, Tips and Solutions

Kunio Hitomi, Applications and Support Engineer

Q. Is automatic geometry binning for TFT models currently available?

A. Level=36 RPI poly-Si TFT and other original TFT models do not support automatic geometry binning. The enhanced model included with ***SmartSpice*** 1.9.7.C or later now supports the selecting of a suitable model with both LMIN/LMAX and WMIN/WMAX, as well as with other MOSFET models.

Q. How can I run an input deck in batch mode using a PC/Windows version of *SmartSpice*?

A. Both the Windows and UNIX versions of ***SmartSpice*** support batch mode, but users familiar with the Interactive ***SmartSpice*** GUI may prefer not to use it. Batch simulation jobs are executed, with starting flags, from a Windows command line. Setting a path to Silvaco's binary directory (<install_directory>/bin) is recommended.

To invoke ***SmartSpice*** at a command line prompt, type this command with variable options as follows:

```
Smartspice -b <input_file.in> -o
<output_file.out> -r <data_file.raw>
```

The '-b' flag executes a ***SmartSpice*** simulation in batch mode. If –b is omitted, the main ***SmartSpice*** window appears on the desktop. The '-o <output_file>' (create a simulation logfile) and the '-r <data_file.raw>' (create a raw data file). The '-r' and '-o' flags are optional and can be omitted.

Note: Windows supports the use of file names that contain spaces. ***SmartSpice*** 2.2.0.R or later supports these filenames. Filenames that contain spaces must be contained in quotation marks ("") when typed in the Windows command prompt. For example:

```
smartspice -b "abc def.in" -o "abc def.out"
```

Q. May I specify an initialization file (.SmartSpice.ini) on Windows as well as on a UNIX OS?

A. ***SmartSpice*** 2.2.0.R now supports the "-startupfile" option, which specifies an user-defined initialization file within a directory. For example, to find the "smartspice.set" in a ***SmartSpice*** library directory, you might phrase the commend like this:

```
<inst_dir>\lib\smartspice
\<smartspice_version>\x86-NT
```

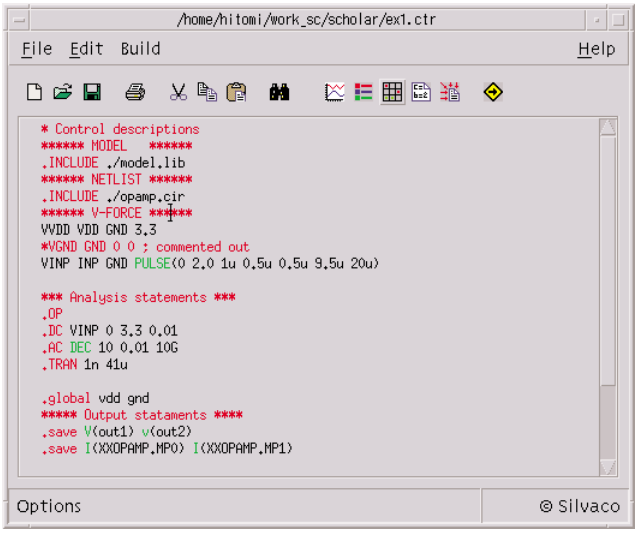

Figure 1. SpiceDeck editor with a color-coded syntax checker.

SmartSpice automatically loads an initialization file at startup by default, but the user can specify a separate custom startup file from both UNIX and Windows commend lines:

```
smartspice -startupfile <filename> -b
<input_file> [other option flag(s)]
```

Q. How can I generate Hspice data files?

A. ***SmartSpice*** supports both Hspice compatible command syntaxes and data generations in ***SmartSpice*** format. By default, ***SmartSpice*** generates a raw data file (<filename>.raw) by the use of the '.option post' or '-r' startup flags in batch mode.

SmartSpice also optionally generates the data files in Hspice format (*.tr?, *.sw?, and *.ac?). In order to generate the data files, describe .options POST is placed in the input deck and the '-hspice' command flag is used when starting ***SmartSpice***:

```
smartspice -hspice -b <input_file> [other
startup option flag(s)]
```

Hspice-format data files are generated with an appropriate file that depends on the type of commend executed. ***SmartSpice*** also supports the parsing of Hspice analysis commands and syntaxes (such as element/device syntaxes and functions). The use of the startup option flag also prevents compatibility conflicts between different ***SmartSpice*** and Hspice syntaxes.

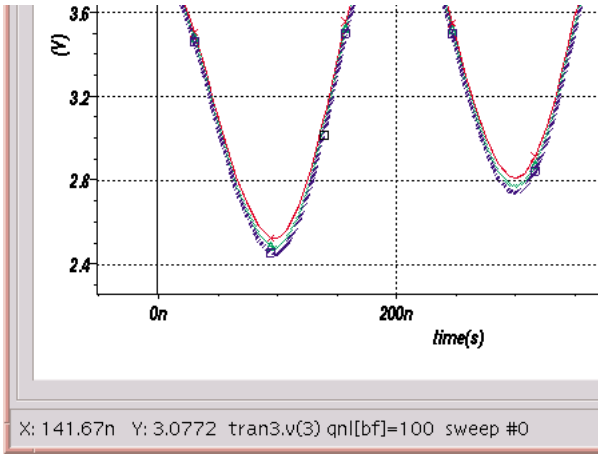

Figure 2. Parameter value shown when pointing a mouse cursor on a waveform.

SmartSpice also supports compatibility with other commercial SPICE or SPICE-like simulators, such as PSpice, ELDO and Spectre. A different command line flag is necessary for alternate simulator compatibility:

```
smartspice -[simulator] -b <input_file>
[other startup option flag(s)]
```

The –[simulator] option supports these formats: hspice, pspice, eldo, and spectre.

Q *SmartSpice* sometimes continues simulating even when some error messages are reported. Can I stop the simulation if an error is found?

A. *SmartSpice* will simulate an input deck as long as the circuit's topology is adequately expanded to a solutions matrix and at least one analysis statement is executable resolved syntax are ignored. For example, if .dc, .ac, and .tran analyses are defined in a deck and two are syntactically incorrect, the remaining correct analysis will still execute. Users are able to instruct ***SmartSpice*** to stop the simulation in one of two ways if any incorrect description is detected in the input deck. One way is to set the 'stoponfatalerrors' variable in .SmartSpice.ini to true..SmartSpice.ini or the corresponding .option STOPERR in an input deck:

```
set stoponfatalerrors = 'true'
```

The user can also add the '.option STOPERR' statement to an input deck:

```
.option STOPERR
```

This option is useful for checking descriptions of circuits and syntaxes before running a time-consuming circuit or parametric analysis.

Note: ***SmartSpice*** supports several variables that permit different feature settings of ***SmartSpice***. These are usually set in a user defendant .SmartSpice.ini file or inserted in the input deck as control loop eg.

```
# the first line is reserved for a title
    .control
    <other SmartSpice variables>
    .endc
```

Q. How do i guard against high voltages in my circuit?

A. You can use the option VSTA to limit the voltage charge between 2 successive time points in the simulation. The default value is 1000 volts eg. .options VSTA=10 limits voltage change to 10v and so traps excessive charge. You can also guard against reverse bias of active device junctions (in BSIM3/4 models) by using these model parameters:

VGS MAX Maximum limit Vgs (gate to source voltage)

VGS MIN Minimum limit Vgs (gate to source voltage)

Q. How do I obtain parameter values for each curve when executing a parametric analysis, such as .st, .modif, or nested sweeps, with a basic analysis (.dc, .tran, .ac)?

A. ***SmartSpice*** typically generates raw data in a simple format (based on Berkeley SPICE) and suppresses large data generation by default. ***SmartSpice*** 2.2.0.R has an option to generate additional parameterized data as a vector into a raw data file. To enable this output, set the following variable in the current user's SmartSpice.ini:

```
set parametrized_data_in_raw
```

Once the variable is set and an analysis is run, the new parameter name appears as a vector along with the analysis type in the Vectors tab window (Display Spice) of ***SmartView***. After a new plot is created, the vector is referred to by pointing the mouse cursor at a waveform and selecting node voltages, such as V(2) and V(3). Figure 2 shows the analysis type (transient), vector name (V3), the modified parameter value (100), and parameter name (qnl[bf]).

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

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