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 command like this:

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

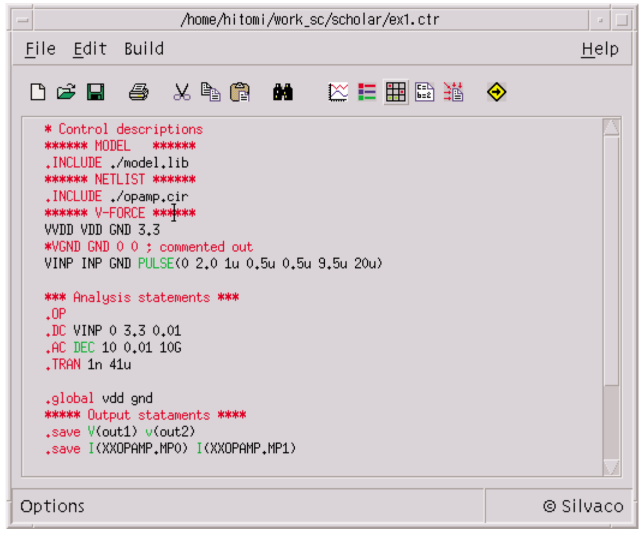

Figure 1. SpiceDeck editor with a color-coded syntax checker.

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.

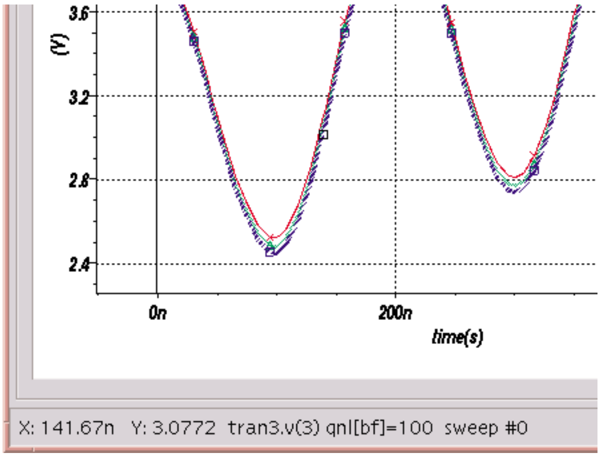

Figure 2. Parameter value shown when pointing a mouse cursor on a waveform.

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