SPICECAD
The Schematic Entry for SPICE3,HSPICE and TITAN under X11 R5/R6

SPICECAD Development
www.spicecad.com

December 11, 1998

Contents

Chapter 1

Introduction

SPICECAD is a program for the interactive graphical input and simulation of electronic circuits. Is is in steady development. It contains an interface to the simulator SPICE3, which means, that all devices and all important types of analyses are supported. (.FOUR, .SENS, .TF , .PZ are not supported yet) SPICE3 is linked to SPICECAD (it is not running in background or as a daemon), which means that the interface is very, very fast.

SPICECAD allows hierarchical design with an arbitrary number of hierarchy levels. Netlists which were created by other tools (for example, macro-models for operational amplifiers supplied by silicon manufacturers) can be incorported , too. The user can create his own symbols for subcircuits , and behind this symbol the external netlist is hidden.

SPICECAD allows the usage of user-defined variables for the definition of parameter values like resistors, mos transistor geometries and so on.

The database SPICECAD creates are simple ASCII files which can be easyly understood and edited for own purposes, for example, if you wish to change the name of a model-file (replace-string 'nmos4' 'nbsim2') of all mos transistors in the circuit.

SPICECAD supports the library principle, which means that symbols which describe deeper hierarchy levels (levels under the top level) are stored in subdirectories (this is a can , not a must). If you want to replace all symbols of a certain type by a symbol of another type , you simply change the library search path instead of replacing all elements.

On the other hand, SPICECAD does not support global search and replace operations.

SPICECAD has full backannotation capabilities. DC-OP , DC-TRANSFER, AC and TRAN- analyses - results can be displayed by mouseclicks. DC-Voltages, currents and device information can also displayed by mouseclick. This backanotation process is not only possible when using the SPICE3 simulator. Also when using HSPICE or TITAN, backannotation can be done just as easily as if using the built-in simulator.

Zoomin, Zoomout and a measure function are implemented for the postprocessing of these results. Additionally, a UPN waveform calculator is available which can perform unary and binary calculations on the simulation results : +,-,*,/, FFT, LOG, DIFF, INT, y**2 , and so on are implemented.

Plotting of schematics and simulation results in HPGL and POSTSCRIPT is possible.

SPICECAD gives you three additional simulation tools which are not available in SPICE3 .

These are:

- parametric sweep (sweeping over a user defined variable)
- monte carlo analysis
- circuit optimization employing the gradient method , rapid thermal annealing or exhaustive search .

If you have any idea for other simulation tools, please contact support@spicecad.com

Chapter 2

Installation

The SPICECAD distribution has been split since Version 1.5. There is only one file, spicecad.RELEASE-VERSION.tar.gz which contains a manual, html manual pages, example files, and on-line help files. This file is unpacked with 'gunzip spicecad.RELEASE-VERSION.tgz ; tar xvf spicecad.RELEASE-VERSION.tar '

You can untar this file whereever in you want to. The directory structure which will be created will look like this:

```
3294 ./spicecad/bin
6290 ./spicecad/doc
72 ./spicecad/examples/lib1
2 ./spicecad/examples/mod1
24 ./spicecad/examples/sppar
172 ./spicecad/examples/lowpass_filter
44 ./spicecad/examples/psk_modulator/lib
244 ./spicecad/examples/psk_modulator
770 ./spicecad/examples
114 ./spicecad/help
```

The 'bin' directory will only contain some small supporting executables, but not the SPICECAD binaries themselves.

The files which contain binaries have to be loaded separately, but from the same location where you obtained file spicecad.RELEASE-VERSION.tar.gz

. Depending on whether you want to use SPICECAD on a PC running Linux or on a SUN workstation running SUNOS or SOLARIS, you have to load the file
'spicecad.RELEASE-VERSION.linux-elf.gz '
'spicecad.RELEASE-VERSION.sunos.gz '
'spicecad.RELEASE-VERSION.solaris.gz '
Simply gunzip them and you are almost ready to go. For simplicity, you should unzip the appropriate file in directory

./spicecad/bin .

Okay. Now, modify your PATH variable that it points to '/SPICECAD_INSTALL_PATH/spicecad/bin '

Then, modify your '.cshrc' or '.bashrc' - file and create a new environment variable SPICECAD_HELPDIR. This variable must point to catalog '/SPICECAD_INSTALL_PATH/spicecad/help'. The help files were inside the 'tgz' file you have unpacked, they are called 'help_eng.spi' , 'help_ger.spi' , 'option.help, 'passive.help ' .... For the SPICE3 online help system, a second environment variable SPICE_HELP_DIR is needed. This variable must point to the location where the files 'spice.idx' and 'spice.txt' are stored. 'spice.idx' and 'spice.txt' are also inside the tgz file .

For simplicity, these environment variables should be the same.

Now, you are ready to run SPICECAD. If you have not set the necessary environment variables, SPICECAD will remind you to do that.

Chapter 3

First Steps

This chapter guides you through the first steps to a first simulation. At this point, you should have installed all binary files , documentation and examples by untaring and unzipping the recent SPICECAD version obtained from the internet of wherever from.

WARNING: If you use a mapped keyboard ('xmodmap'), hotkeys will not work as described. Plese read the chapter 'Hotkeys' on page 104

if strange things happen if you use hotkeys.

Please create a new directory and copy the following files into this directory:

- File 'cad.lat' contains the last schematic or symbol you have edited.
- File 'last10.spi' contains the last 10 schematics you have edited. In older program versions, SPICECAD is looking for 'last10.cc' instead of 'last10.spi'. Don't worry, SPICECAD will look for both if you have old 'last10.cc' - files, but will always store new history files as 'last10.spi'. Please remove old files 'last10.cc' by hand. Sorry for this inconvenience.
- File 'setup.spi' contains setup information. In older program versions, SPICECAD is looking for 'setup.cc' instead of 'setup.spi'. Don't worry, SPICECAD will look for both if you have old 'setup.cc' - files, but will always store new setups as 'setup.spi'. Please remove old files 'setup.cc' by hand. Sorry for this inconvenience.

If you have forgotten to copy this file, SPICECAD will remind you immidiately after starting the graphics session to create such a file. Please read chapter 'Options - dialogs' on page 82 which tells you how to create the setup file.

- File 'help_eng.spi' contains online help texts in english.
- File .spiceinit may contain SPICE3 Initialization commands, or can be empty. If not present in the directory where SPICECAD will be started, SPICECAD might crash.

Then, copy the file 'first.scm' into this directory and set the search path to the SPICECAD binaries.

Now you can start the progam.

The output will be :

```
Note: can't find init file.
called freehist -1000
external error:  no graphics interface; please check compiling instructions
don't worry about the previous error message
type 'cad -c cellname' to start schematic entry
Program: Spice, version: 3f5
Date built: Thu Jan 30 21:38:07 GMT 1997
Spice 1 ->
```

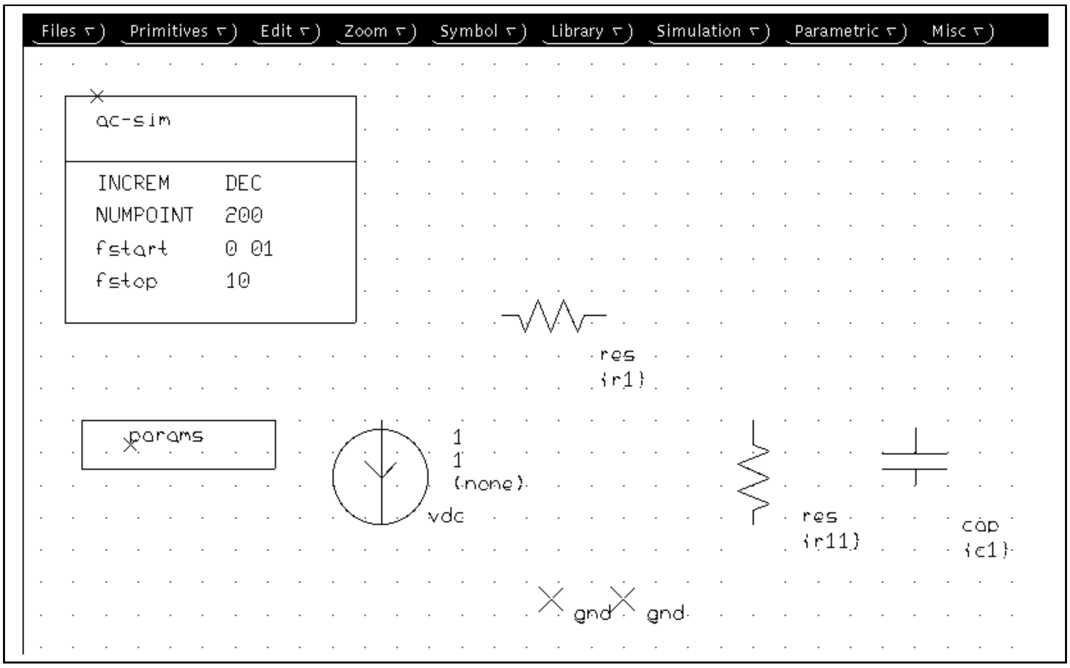
What you see is the well-known SPICE3 command line user interface. If you want, you can stay on this level, running SPICE3 simulations and so on.

If you want to use the schematic entry, now type

```
Spice 1 -> cad -c first
```

Output:

```
calling cad
```


ac-sim

INCREM DEC

NUMPOINT 200

fstart 0 01

fstop 10

Figure 3.1: First schematic

After doing this, a window will pop up which shows you the first example schematic.

We hope it won't happen, but, from time to time, the program crashes with ' internal error - segmentation violation'. Don't worry - try again (once, twice or thrice). The crash happens during the initialization phase of the XVIEW - toolkit which we are not responsible for. If you can't start the program after the third try, something else is wrong.

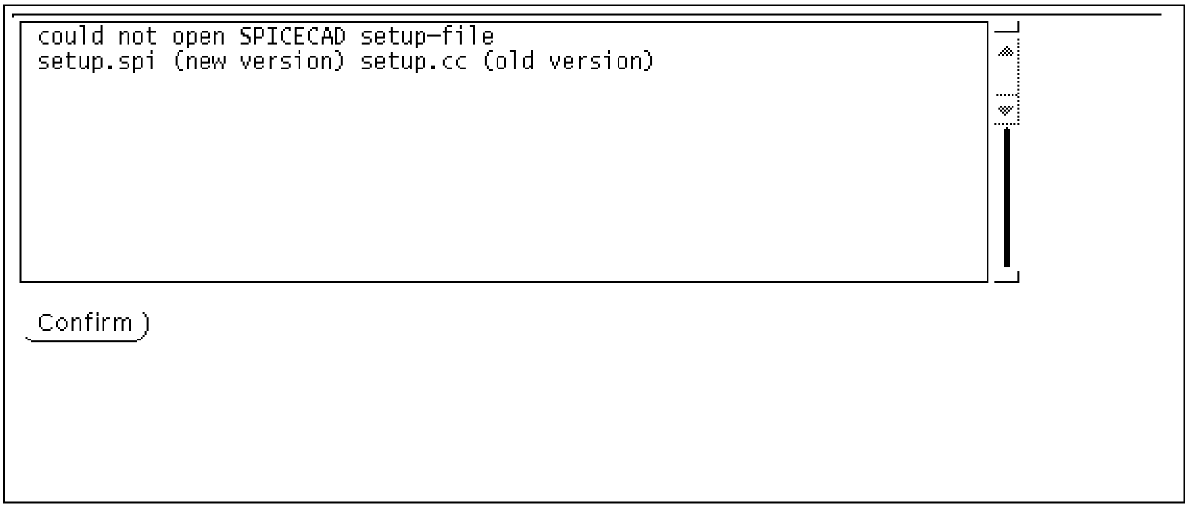
If no crash occurs, a window will come up which is titled 'first' and which contains the version number and the compilation date.

This is your schematic editing window.

If you have forgotten to copy file 'setup.spi' into you working directory, a message box will pop up showing this message:

in a second window. `'could not open setup-file setup.spi'`

Push 'CONFIRM' , and the window will vanish until you make your next error.



```
could not open SPICECAD setup-file
setup.spi (new version) setup.cc (old version)
```

Figure 3.2: Example of an error message

The box you might have seen is the error- ,warning- and information-box.

If you are a SPICECAD starter and if you have installed the freely available version with a reduced functionality, another message will come up:

```
Please register SPICECAD.
Otherwise, parameter sweep, monte carlo simulation and
optimization are not executable.
Press 'registration' button in main menu to register.
```

This feature has been implemented because we are very curious about who is using the free version of SPICECAD. If you press the 'register' button in the main menu, a message file will be created and sent to support@spicecad.com by e-mail.

This file contains only one information: your login name which is obtained using 'who'. Additionally, a file will be created and copied into your HOME directory to tell SPICECAD that you have registered yourself. The next time you start SPICECAD, SPICECAD will look for this file. If SPICECAD could find this file, the 'please register' message will not appear again, and SPICECAD has the full functionality.

The file sent to SPICECAD DEVELOPMENT by e-mail will of course only reach us if your machine is attached to the net. If not, the registration file in the HOME directory will be created, anyway, and you will get full access to the extended SPICECAD simulation functions. But, we would really appreciate if you let us know by other measures that you are using SPICECAD. We really like the feeling that SPICECAD is used world-wide by many people, and the messages we get will keep us motivated to push SPICECAD to the limits.

But now, back to business. If you did everything correct, you still see the schematic window.

Now, you ask yourself which mousebutton to press first. To give you an idea what each mouse button press means, a third window pops up. This window gives you a little context-sensitive help and tells you what will happen if you press the right or left mouse window.

For further information, please read chapter 'The mousecommand window' on page 21.

In that window, you will see a simple RC-low pass circuit schematic.

The circuit is made of a DC voltage source, two resistors and a capacitor. These devices are connected via blue lines, the nets. On the bottom, you can see a cross with a label 'gnd'. This is the ground connector, which means that the net which crosses this connector is the ground net. Nets will be assigned net numbers after the so-called netlisting process . This ground net will have the number 0.

What you can find in the schematic window, too, are two boxes : one is labeled 'params' , the other is labeled 'ac-sim'.

The 'ac-sim' box is a box which must be placed when a small-signal simulation shall be run. The box, if present in a schematic, contains properties which tell the simulator details about frequency range, sweep type, number of points and so on.

If you double-click the property box, the property editor will pop up and grants access to all properties (described later).

Please close the property editor ('Quit editprop') and double-click at the parameter box. Again, the property editor will pop up , but now, you can see different properties: the properties of the parameter box.

Now, double-click at all the other elements in the schematic to look at their values. If you are familiar with SPICE3 or HSPICE, you will know what they mean. For beginners, it will be hard to understand their meaning. Thus, please consult the on-line help (see chapter about online help) .

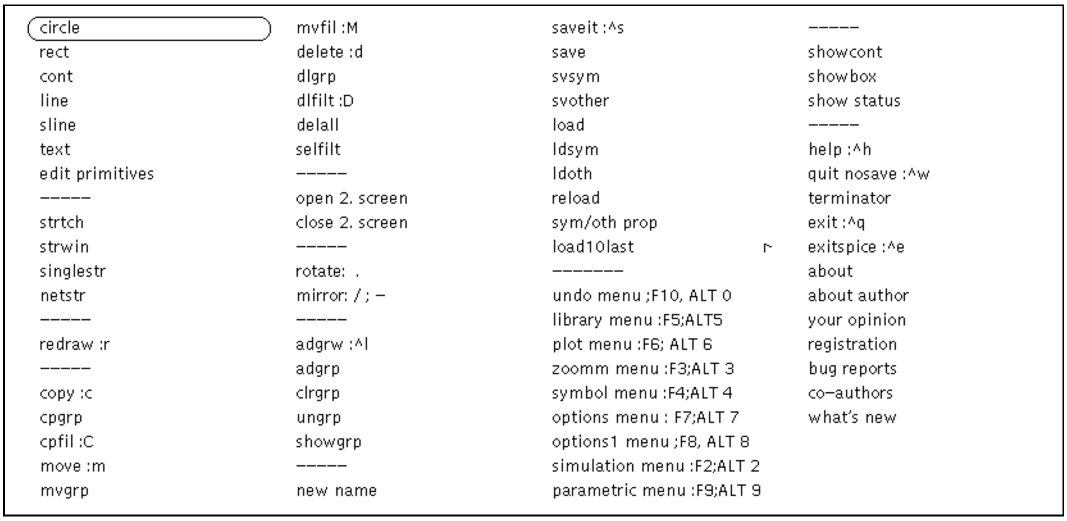


Figure 3.3: The main menu

To give you a fast motivation to find this program nice, simple, and useful, we will do an AC-simulation within one minute.

Push the right mousebutton within the schematic diagram. What appears now is the main menu.

If another menu appears you have pushed a function key F1 ... F10. Select 'leave' or 'F1' or get the main menu next time.

If you have pushed two function keys after another without popping down the first menu by selecting a menu item or pressing a mouse button outside the menu, you will not be able to remove the first menu by selecting a menu item or pressing a mouse button outside the menu. This happens if running SPICECAD on linux PCs. On a SUN SOLARIS 2.X machine, this effect is not observable. To remove dead menus, you must push the function key of the corresponding menu you want to disappear. Then, the menu which appeared to be dead before become alive again and can be popped down. For example, if the main menu remains on the screen, push F1. After doing so, you can hide the menu conventionally by pushing a mouse button outside the menu.

Now, try the right mouse button again.

First, select the 'simulation*' entry in the main menu or press 'F2' or 'ALT ' and '2' after another. Now, you are in the simulation menu level.

Click right mouse button to make the simulation menu appear.

Click left towards the top of the first column and select 'runsim'. This command creates a netlist, reads the netlist into SPICE3 and runs SPICE3. You will get a netlisting statistics in the xterm window where you started SPICECAD. You can read the netlist in an xterm by selecting 'lookat netlist'.

Now, select V(x) to select a net you want to plot.

Notice that until now you probably always selected menu items using the left button. And every time the menu would disappear. To keep the menu window on the screen click left in the menu's upper left corner, over the thumb tack. That frames the menu as a separate window which you can move around and iconify. Note: you can 'kill' that window using your window manager's menus. Do not 'destroy' it though because that kills the entire spicecad session!

Move the cursor over the upper contact (red square) of the capacitor and press the left mousebutton. Notice the adjacent lines become yellow.

Select 'plot acm' in simulation menu.

If you did everything right, a result graphics window will pop up and show you the transmission function of a simple RC lowpass.

When you look at the menu entry 'plot acm', the full entry is 'plot acm : ALT p'. This means that the function 'plot acm' is also reachable via hotkey 'ALT p'. . So, please press ALT and 'p' (at the same time or after another). You can see that the plot window will be updated after pressing the keys. On certain computers, this might not work, and you will get an error message like this:

```
key unknown :  65513 offset 0
```

This message means that the meta key 'ALT' is wrongly coded. Please read chapter 'Hotkeys and key redefinitions' on page 16 . It will tell you how to solve this problem.

Note that the first curve is black on white... If you plot more curves, sometimes their colour can be nearly invisible on a white background. This is machine-dependant. If you want to use other colors, please read the chapter about setting up SPICECAD using file 'setup.spi' (see chapter 'Customization' on page 53). If this does not help, you can also change the background color of the plot window from white to black (more contrast). The setup line in file 'setup.spi' is 'plotwindow_background_is_white yes'. If you set this variable to 'no', the background of the plot window will be black.

Now, move the mouse cursor over this window and push the right button. The menu of the plotwindow will pop up.

Select the 'loglog' entry (first colum, towards the middle) and watch and enjoy.

Now , select 'exit' in the main menu.

I hope , your motivation to explore the features of SPICECAD have risen enourmously.

In the next steps, you should edit the schematics 'second.scm' (cad -c second) , 'third.scm' ... until 'fifth.scm'.

In every schematic, you will find some new features.

Close the plot window using 'close plotwin' (bottom right of simulation menu) and finally close the simulation menu. Open the main menu and 'exit'. In some cases, Terminator, exit, both end up with an 'internal error – segmentation violation'. Exit spice works ok to quit everything.
In the xterm re-launch spicecad and then type
'cad -c second'.

'second': make a dc operating-point simulation and get the node voltages and device status using 'showdevice' and 'showvolt' in the simulation menu.

'third' and 'fourth': hierarchical editing. Push and pop through the hierarchy using 'edit' (hotkey 'e') and 'return' (hotkey 'a') in the symbol-menu (hotkey 'F4' or 'ALT 4')

'fifth': hierarchical editing with local variables. In 'third' and 'fourth' there were also used variables, but all variables were global. Here, we have twice the same block, but their local variable 'symfak' which determines the frequency bahaviour is different in other blocks.

Note : variables are being edited selecting 'editprop' in the library menu or pushing 'g' as a direct hotkey. End of Note.

Perform an AC-analysis , select the output node of each block ('V(x)') , select 'plot acm' and look at the results.

'sixth': Import of external netlists.

The adder in the schematic is something i called 'other' (f_adder.oth). The adder has three nodes , two input and one output. Is you look at the file 'f_adder.m' in the directory './sppar' , you see an arbitrary source which performs the addition.

```
.SUBCKT f_adder 1 2 3
B1 3 0 v=v(1)+v(2)
.ENDS f_adder
```

Now, look at the contents of the file 'f_adder.oth' in directory 'lib1'. There, you will find three objects of type 'contact' (the red boxes) named IN1, IN2 and OUT. At the end of the file, you find the line `'3 IN1 IN2 OUT'.` This line defines the netlist-sequence of the nodes and must be identical to the sequence of nodes in the '.SUBCKT'- command in file 'f_adder.m'.

Thats all .

Simple, isnt it?

Unfortunately, these lines after the 'stop'-statement inthe '*.oth'-file must be created by hand. So, please be careful.

'seventh': parametric sweep, monte carlo simulation and optimization.

Get into the 'parametric' menu (hotkey 'F9', 'ALT 9'). Select 'run paramsweep'. Select a node and watch the output.

Select 'run monte carlo' Select a node and watch the output.

Do a normal simulation ('runsim in the simulation menu '). Select the upper node of the capacitor twice. Plot the AC-result, it must be looking as a RC-lowpass always looks. Push the 'editpop'-window (hotkey 'g') and look at the starting parameters in the 'params'-box : 'res1' = 1.2 , 'cap1' = 0.8. Look at the goal-file 'seventh.goal'. There you find how the results should look like. Select 'run optim (grad)' and watch the optimizer on the result window doing his work.

Now, pop the 'editprop'-window up again and check the parameters in the 'params'-box. Both should be more equal to 1.0 than before.

'eigth': Simulating a transistor circuit. Use 'showdevice' and 'showvolt' and see what happens.

Now , take a break and have a coffee or whatever you prefer. YOU DESERVE IT!

Because, now, it is time to do your first own schematic.

Leave SPICECAD.

Start SPICECAD again.

enter 'cad -c myfirstown'.

Don't worry about the error message. Confirm it and blow it away.

Now, get into the library menu ('F5' or 'ALT 5') and select whatever you want.

Menu 'Library'.

Place it and edit the properties in the property window.

If you think it's enough, connect the devices using 'create new net' or pressing 'n'.

Note: The default net drawing style is ' vertical , then horizontal'. You can change this pressing 'h' for ' horizontal, then vertical ' and 'v' for ' vertical , then horizontal' again.

At the end , place at least one 'gnd'-symbol.

YOU MUST DO THAT!

Now, save your schematic pressing 'CONTROL s' (for 'saveit' in the main menu). Don't worry about the error message, it always appears the first time you save a new design.

WARNING: don't forget to save from time to time. It is not been done automatically , and there is no UNDO function in SPICECAD. If you think that you might start an action, i.e. stretching, which could fail, please save the schematic before beginning this action. Afterwards, there might be a reason to finish SPICECAD wasting the spoiled schematic and start SPICECAD again using the saved schematic version .

Now, change into the simulation menu and place an analysis box.

Edit the parameters .

RUN SPICE.

Here, the introduction into the basic features ends.

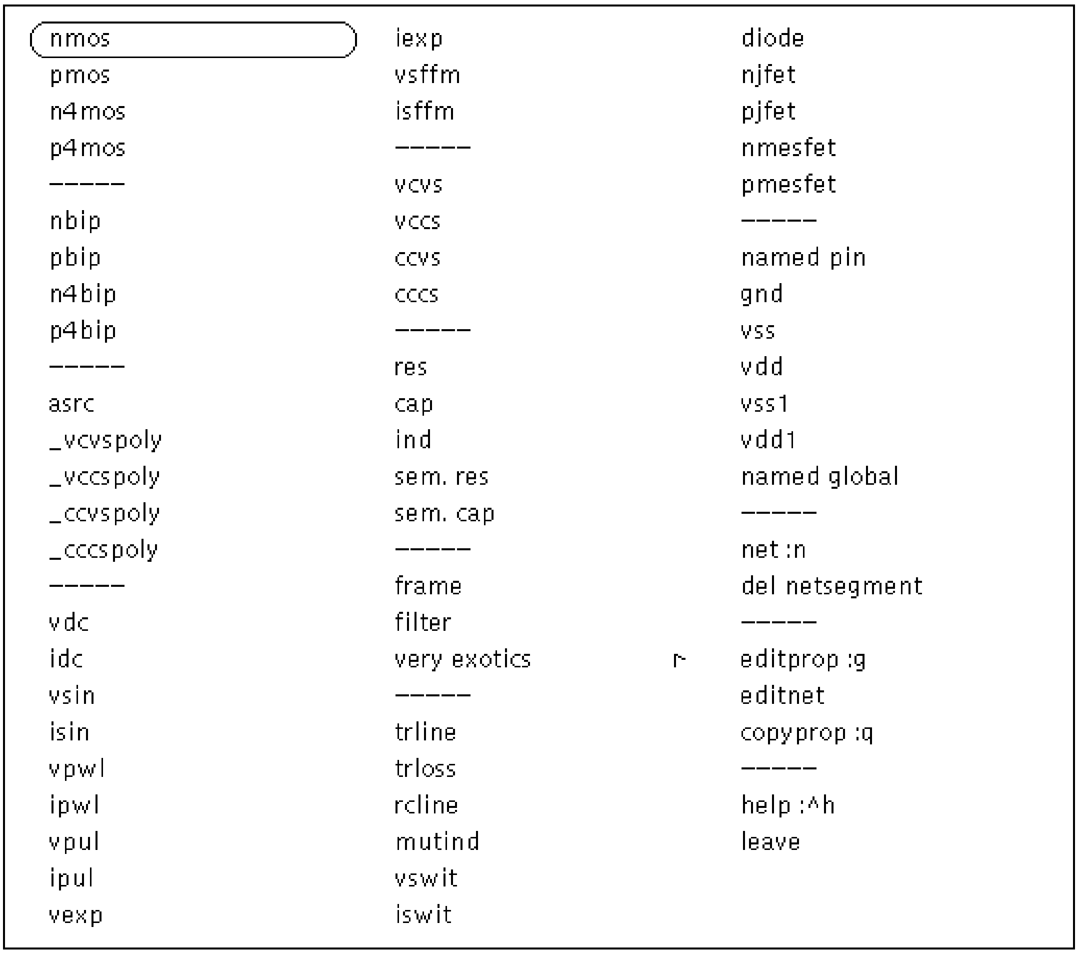
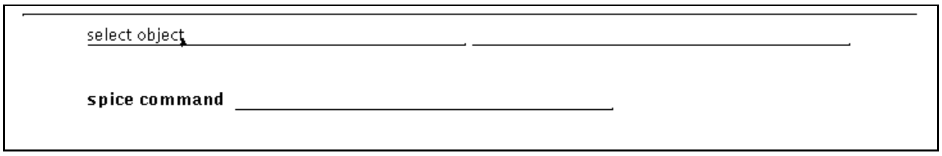
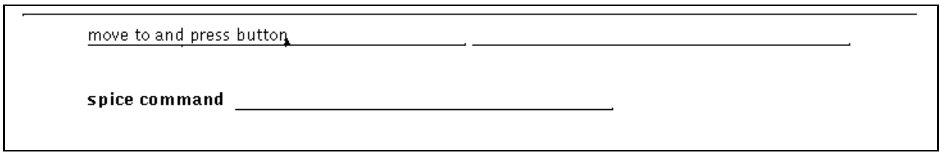


Figure 3.4: Library menu



select object

Figure 3.5: mouse window



move to and press button

If you have took a look at all schematics , you know most features of this program. Many other functions could not be mentioned yet. They will be explained in the following chapters.

3.1 The main window

3.2 The 'mousecommand' window

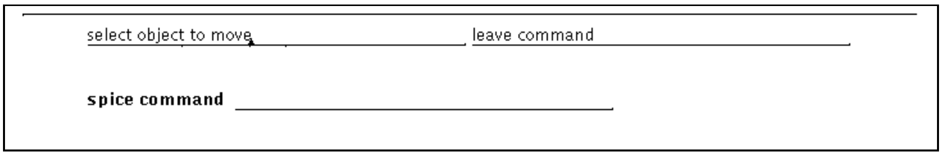
The mousecommand window provides you with information about the meaning of mousebutton clicks within the context.

For example, if you select the 'move unspecific' function, the mouse command window looks like this: it tells you that you have to select the object you want to move with a click of the left mouse button.

After having selected the object to move, the window contents change : it tells you that you can stop moving the object with another click of the left mouse button.

After having moved the object , the window contents change again: it tells you that you can move another object with another left mousebutton click, or you can leave the 'move' event procedure with a click of the right mousebutton.

If you are an experienced user or if you have a small screen, you might not want to display this window. In this case, edit setup file 'setup.spi'



and change the entry

```
want_mousecomwindow yes
```

to

```
want_mousecomwindow no .
```

3.3 The plot result window

3.3.1 Eye diagram plots.

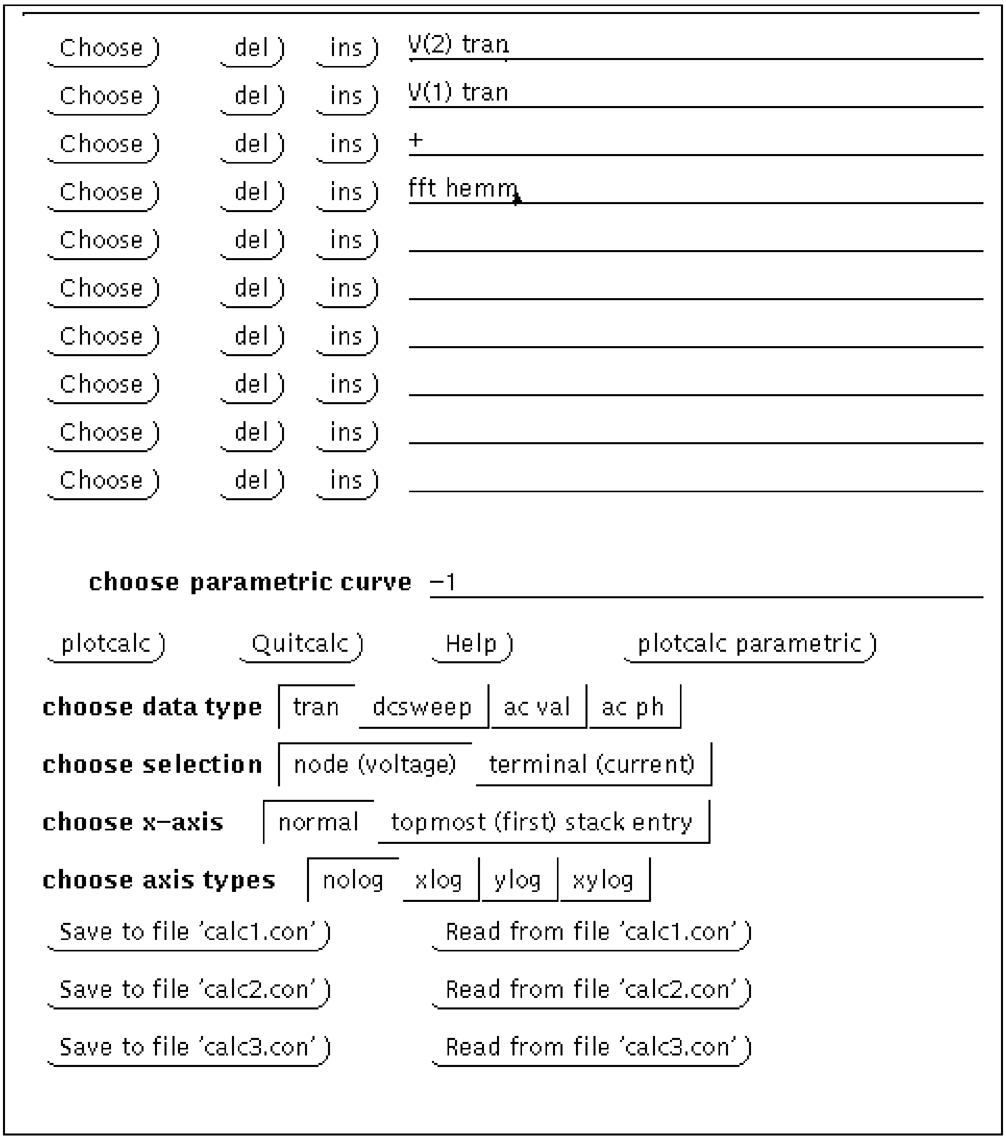
Since V1.7, it is possible to plot eye diagrams. Eye diagram plotting is started when setting the 'eye diagram period' inside the 'plot options' - dialog. Setting this entry to a value unequal '(none)' processes that data which will be plotted in a way that an eye-diagram can be seen on the plot result window.

3.4 The calculator window

The waveform calculator which you can see in the picture below is a UPN calculator which can perform unary and binary calculations on the simulation results (waveforms): +,-,*,/, FFT, LOG, DIFF, INT, y**2 , and so on are implemented.

The calculator can be started by selecting 'calculator*' in the plotwindow menu.

Each line contains either an operator or a simulation result (waveform). Abstract waveforms like ONOISE and INOISE can be used, too, to compute noise results as any other result. Before filling out the lines, make sure that the kind of result you want to compute (tran, dcsweep. ac, ac ph) is chosen correctly in the switchbox in the lower part of the calculator window. By



V(2) tran
V(1) tran
+
fft hemm

choose parametric curve -1

choose data type tran | dcsweep | ac val | ac ph

choose selection node (voltage) | terminal (current)

choose x–axis normal | topmost (first) stack entry

choose axis types nolog | xlog | ylog | xylog

Figure 3.6: Calculator window

default, transient simulation is chosen. Filling out the lines is easy: selecting a voltage or current can be done via mouseclick. First, choose whether you want to compute a voltage or a current in the second switchbox (node or terminal : voltage or current). Then , click the 'choose' button at the left side of the line you want to fill. Afterwards, you may select a net or a terminal. Selecting a terminal means that you can select a voltage source or a device which terminal currents have been saved using 'save cont curs'. In SPICE3, you have to execute 'save cont curs' before simulation to save the currents through elements like resistors and transistors. Otherwise, only the currents through voltage sources are saved, which means that voltage sources may be selected at any time. If you did 'save cont curs' which causes the contact boxes to be displayed in yellow instead in red, you can select these yellow contact boxes to process the current flowing through these contact boxes.

After selecting a node or a terminal, the node or terminal name will be displayed in the appropriate line:

'V(1) tran' will appear if you have chosen to plot voltages, node 1 , and transient simulation.

The outlook of this may vary between different simulators, so, please do not worry if it looks a little bit different.

When you filled out as many lines as you need (I hope, 10 lines are enough), click the 'plot' button. The filled lines will be interpreted . For example, if your input looks like

```
V(1) ac
V(2) ac
```

two lines will appear in the plot window which show the ac magnitude voltages at nodes 1 and 2.

```
V(1)
V(2)
+
```

only one line will appear in the plot window which shows the sum of the voltages at nodes 1 and 2. `V(1) tran`

```
V(2) tran
+
V(3) tran
```

Shows the sum of the transient simulation results voltages at nodes 1 and 2 and the voltage at node 3. `V(1) acph`

```
V(2) acph
+
V(3) ac
V(4) ac
```

Shows the sum of the ac phases at nodes 1 and 2 and the product of the ac magnitudes at nodes 3 and 4. You can mix ac and acph in the same calculator window when you want to plot ac phases and ac magnitudes in the same plot window. What is not allowed is mixing ac and tran simulations, for example, because
a) it makes no sense
b) due to the different vector lengths of ac and tran simulation results, this can lead to unpredictable results.

```
V(1) tran
V(2) tran
-
V(3) tran
V(4) tran
-
```

Shows the product of the difference voltages at nodes 1 and 2 and nodes 3 and 4.

```
V(1) tran
V(2) tran
-
V(3) tran
V(4) tran
-

pow2
FFT hemm
```

Shows the fast fourier transformation of the power of two of the product of the difference voltages at nodes 1 and 2 and nodes 3 and 4. The waveform to be transformed will be Hemming-windowed before the fft is executed.

.
.

.
It is a UPN calculator, indeed.

Since SPICECAD version 1.5, some new features have been added to the calculator.

- Parametric plotting of calculator results is now possible .
- The choice of appropriate x and y axes is now poossible from within the calculator window.
- The calculator lines can be saved in different files (three possible selections) and can be restored afterwards. So, a fast switch between different plot setups is possible.
- Import of ASCII files is possible now. The ASCII files must have two columns, where the first column is the x-value and the second column is the y-value. The syntax is 'import filename' If you want to import files with more than two columns, the syntax is
  import filename numcolumns firstcolumn secondcolumn
- Shifting and sampling simulation vectors is possible, now. If the number of data points exceeds a certain value, the time it takes to display the results may be quite high, especially if you have chosen the 'linespoints' option. Sampling the vector by a fixed sampling period displays only every x-th point, where x is the distance between two adjacent points in the x-axis. All other points are suppressed. Shifting of simulation results is interesting very often to determine the delay between signals. Now, the input signal can be displayed versus the shifted output signal to derive the delay. Or, as an example application, the differential non-linearity of a DAC can be derived by subtracting the shifted DAC output from the unshifted output.
- Smoothing of waveforms: very noisy waveforms can be smoothed now without the need for saving waveforms, smoothing them using C programs and re-importing them. The 'smooth' operator looks as follows
  :
  smooth 'value'
  'value' is the number of samples which are to be averaged to do the smoothing .

- Calling external programs is possible now to do any kind of post-processing you might wish to do (cross-correlation, FIR filtering, etc.) The systax is simple :

  exec 'command'
  or
  texec 'command'

  The 'exec' operator is a unary operator , it works with one vector only (like FFT, sample, smooth etc.) The 'texec' operator is a binary operator, it takes two waveforms (like +,- etc.) .

  The 'exec' routines work as follows :
  First, a file 'spicecad_onestack_export' (exec) or 'spicecad_twostack_export' (texec) is created which contains the x,y (exec) or (x,y1,y2) vectors . Next, a script called 'command' is called, where the script name is the first and only option of the exec/texec operator. Inside this script, which is called using
  system('sh command')
  you can do whatever you want: you can call PERL script, or C programs, or whatever. The only restriction is that SPICECAD expects a result file called 'spicecad_onestack_import' (exec) or 'spicecad_twostack_import' (texec) which contains the same number of lines as the first file. Here is an example of a possible C++ program which do a very simple post-processing.

  File onestack.cc

```
#include <stdio.h>

void main(int argc, char **Argv) {

FILE *inp,*out;
double x[100000],y[100000];
int erg,found;
```

```
inp=fopen("spicecad_onestack_export","r");
found=0;
while((erg=fscanf(inp,"%le %le ",&(x[found]),&(y[found])))==2) {

y[found]+= 5;
found++;




}//while

fclose(inp);
out=fopen("spicecad_onestack_import","w");
for (int i=0;i<found;i++) fprintf(out,"%e %e\n",x[i],y[i]);

fclose(out);

}//main
```

File twostack.cc

```
#include <stdio.h>

void main(int argc, char **Argv) {

FILE *inp,*out;
double x[100000],y[100000],y1[100000];
int erg,found;


inp=fopen("spicecad_twostack_export","r");
found=0;
while((erg=fscanf(inp,"%le %le %le",&(x[found]),&(y[found]),&(y1[found])))==3)

y[found]+= y1[found];
```

```
found++;




}//while

fclose(inp);
out=fopen("spicecad_twostack_import","w");
for (int i=0;i<found;i++) fprintf(out,"%e %e\n",x[i],y[i]);

fclose(out);

}//main
```

Chapter 4

Online help

A simple online help is provided in SPICECAD. If you select 'help' and then select a menu button or an object on your schematics screen, a window will appear which displays an informational text about the function which can be executed from this menu button or about the object you have selected.

The helpfile which contains the texts about menu entries is 'help_eng.spi'. The helpfiles about objects are 'option.help', 'passive.help' and so on.

Old SPICECAD version (obsolete) :
The helpfile 'help_eng.spi' has to be copied into the directory from where you start spicecad. Otherwise, you have to edit the setup file 'setup.spi' where you can enter the full path to the file. The syntax of the help entries in file 'help_eng.spi' is very simple, so, if you want to improve the text, you can do that using a simple ascii editor. If you want, you can send us a version of your improved online help text, so other SPICECAD users can participate from your work.

Recent SPICECAD version:
All help files have to be placed into directory SPICECAD_HELPDIR, where SPICECAD_HELPDIR is an environment variable you have to define before starting SPICECAD.

There is a second on-line help system, the built-in SPICE3 help system. This is not accessible after having opened a schematic window. So, if you want to use this help, please start SPICECAD a second time and type 'help' instead of 'cad'.

The usage of the on-line help system is very easy. To use this help system, the environment variable SPICE_HELP_DIR must be set to the catalog where the files 'spice.idx' and 'spice.txt' are stored.

Chapter 5

Available devices

Devices are placed from within the library menu.

Mos- and Bipolar-transistors, Voltage- and current-sources, controlled sources, sine- pulse-, pwl-, arbitrary- and pwl-sources, resistor, capacitor and coil, globals (gnd, vss, vss1, vdd, vdd1) , named pin, input-, output- and bidir-ports.

Symbol (upper) and Other-Element (lower).

lines, switches, diode, JFET, MESFET

After placing a device, the property- form window pops up and demands you to fill in the values. Some properties are required, others are not. There is no complete interactive check whether you filled the form in a correct manner or not. So, you should have good knowledge of the SPICE3 syntax, otherwise, you have to use the 'trial and error' method, where 'trial' means netlisting.

Whenever you wish to give a property a parameter instead of a value, you may do that. You may use parameters in expressions, too.

Example: The first property for the resistor is called 'res'. You may write '100k' , '{resparam}' or '{wid/leng*sheetres}'.

Things like specifying the kind of variable, 10uA, 26mV , are not allowed. DON'T USE SI-UNITS IN YOUR EXPRESSIONS !

WARNING: Expressions must not be longer than 40 characters and should not contain whitespaces. Whitespaces are replaced by underscores automatically.

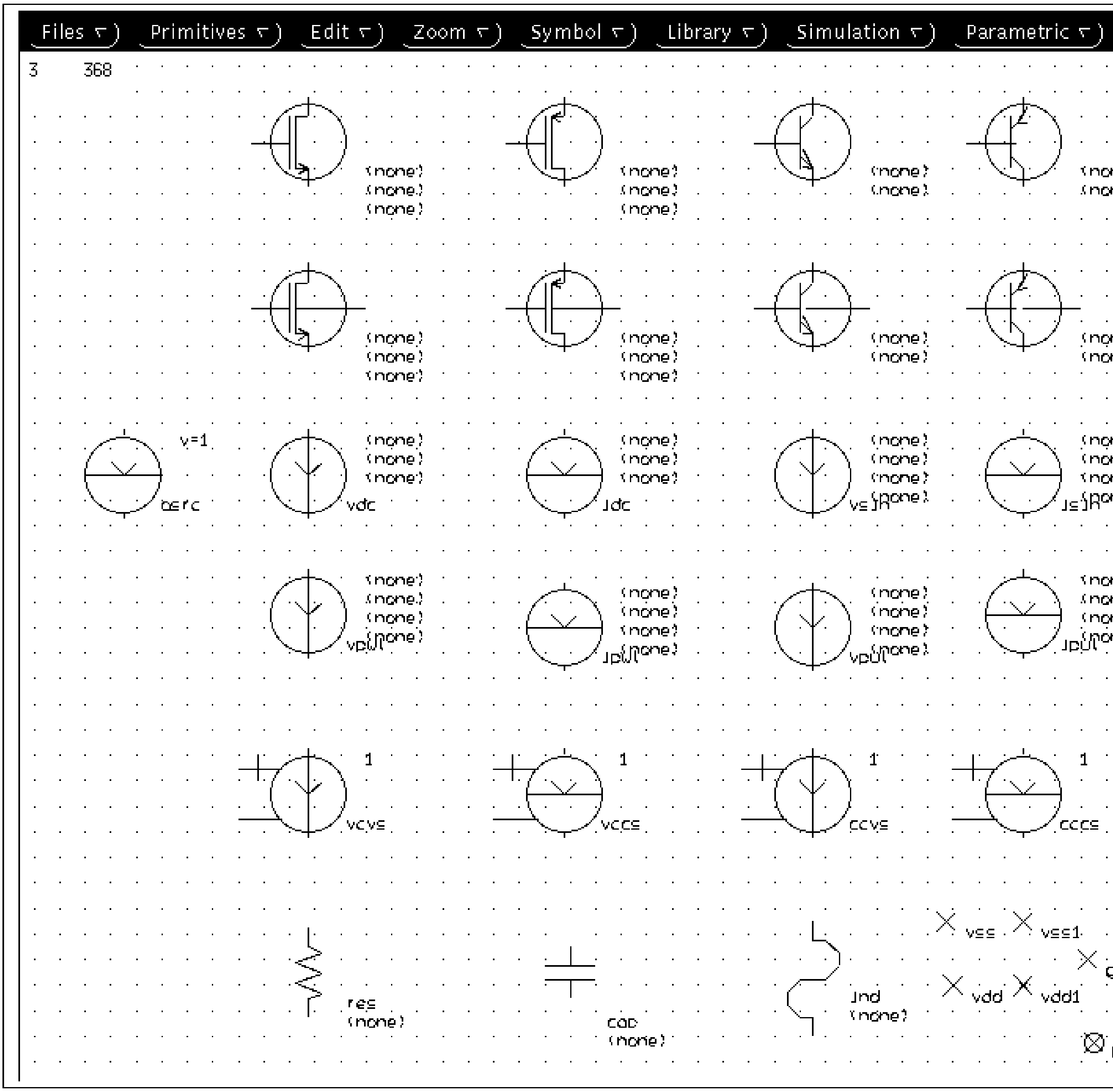


Figure 5.1: Available devices

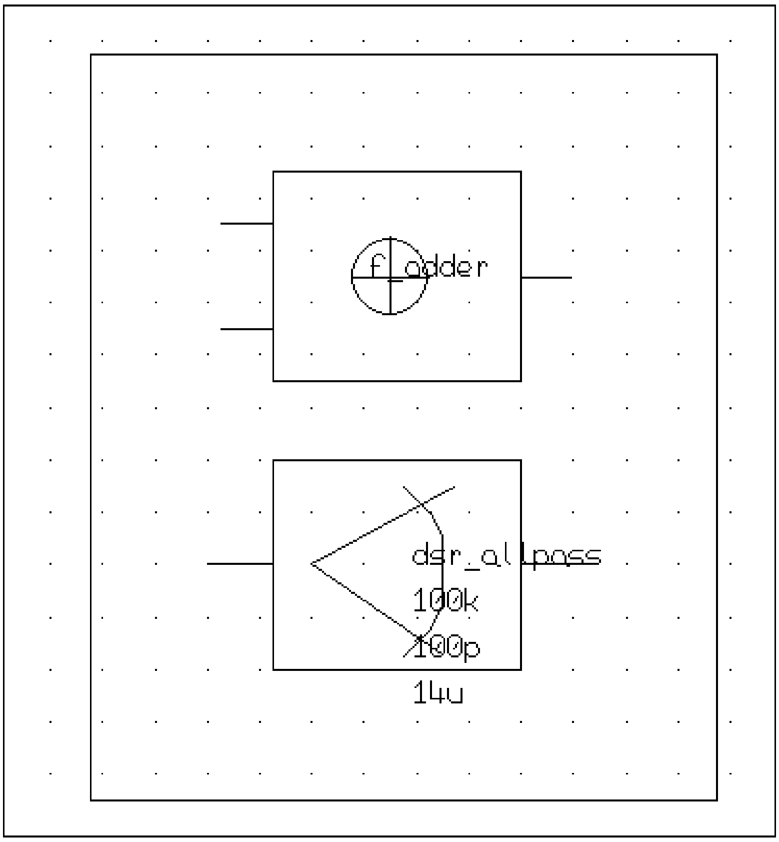


Figure 5.2: Available devices II

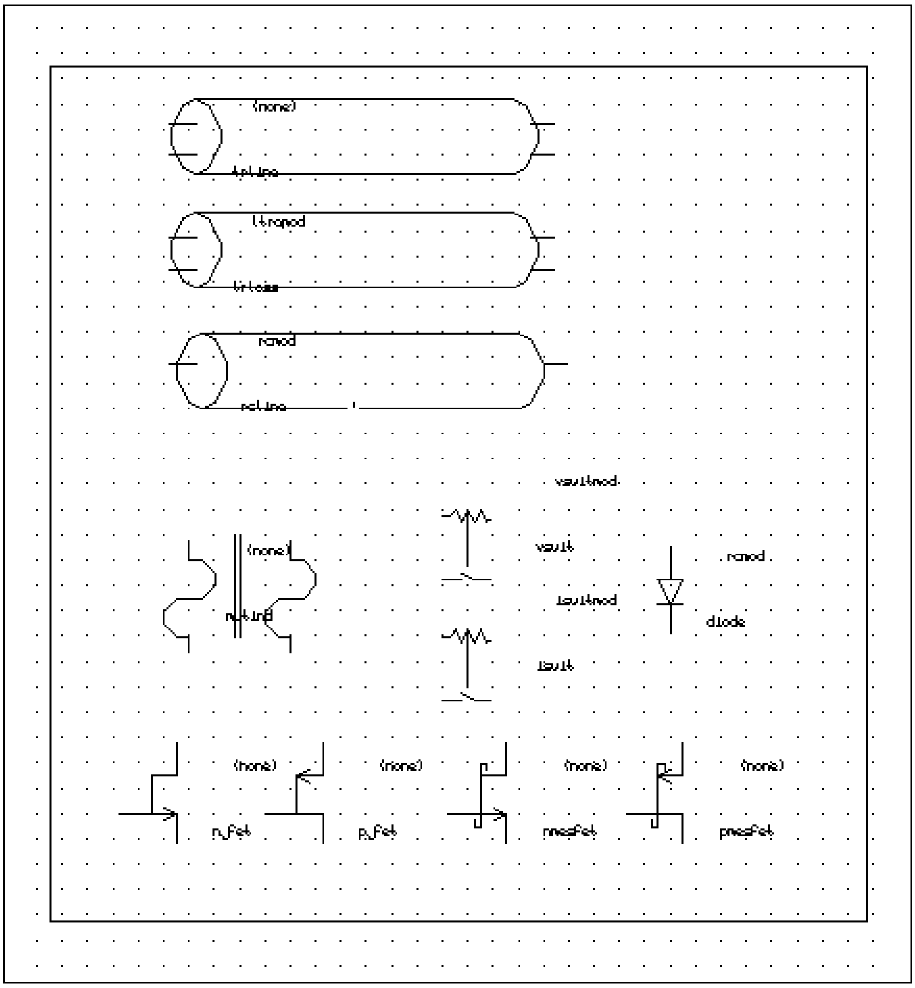


Figure 5.3: Available devices III

Later, if you want to simulate the schematic, you have to place a parameter box (Simulation menu) wherein all parameters of all hierarchy levels will be recursively collected. Editing the properties of the parameter box means editing the values of the parameters.

5.1 Sources

There are some things which has to be told about sources to use them optimal.

All independent sources have the known SPICE3 properties. Besides, they have a property called 'alias'. This alias is the name of the source. The source is referred to by this name in the noise simulation command (equivalent input noise source) and in the dcspweep simulation command. So, if you want to do these types of simulations, be sure that you give the noisy or swept source an alias name. Sometimes, you will forget to do that before placing a simulation box (it often happens to me, too). Then, you will receive an error message that the specified source could not be found. Close the error window, edit the source properties and enter the correct alias. Afterwards, edit the properties of the simulation box once again (pop up the window and close it again). Always after closing the property editor window, a check is done if you have edited things which refer to alias names (nets or sources).

`What is new in SPICECAD V1.7 is the availability of current and voltage controlled current and voltage 'poly' sources.` These sources were available in spice2g6 but have not been ported to spice3. People who want to make use of macromodels supplied by silicon vendors for speeding up simulations have been missing these sources . Many solutions have been found to circumvent this: the availability of a source called 'arbitrary source' can be used to model the behaviour of these controlled poly sources. To convert controlled poly sources to arbitrary spice3-compatible sources, a conversion script has to be run over the macromodel netlist which shall be imported. This conversion process had to be done by the user of the macromodel library before the import of macromodels into the spice3 netlist could be done.

Now, this conversion process is no necessary any longer. For maximum spice2g6 compatibility, the fortran spice2g6 code which derives the output voltage or current of the controlled poly source was translated to C . This work has been done by the creators of XSPICE, developped and written at Geogia Tech.

5.2 Random Sources

For the generation of noisy signals, the 'PULSE' - sources have an additional property, the 'rand'- Property. Setting this property to 'rand' or 'nrand' means that the output of the source is longer a deterministic periodic pulse, it is a pulse train where a random generator determines whether the height of the pulse is the normal height or zero. The 'rand' setting generates a 'return-to-zero'-signal, where the 'nrand'-setting generates a 'no-return-to-zero' - signal. When using 'nrand', the normal PULSE-source-properties 'falll-time' and 'pulse period' are not used for the determination of the pulse train.

This noisy source is available only when using the SPICE3 -Kernel. The introduction of noise could be done by using a PWL-source instead. Unfortunately, the SPCIE3 parser needs an enourmous amount of time to parse PWL sources with more than 1000 lines.

5.3 MOS transistors

SPICECAD has many built-in features concerning MOS transistors.

- each transistor has two model parameters where you can specify models for a Monte-Carlo Simulation. These models affect the geometry of the transistor (gate width and length) .

- Normally, you have to fill in the properties AS.AD,PS.PD,NRS,NRD if you want these parameters to be netlisted. This is very time consuming, especially, if you see that you have to change transistor widths , which means that you have to change all the other properties, too.

  SPICECAD has a built-in formula which calculates and netlists these parameters.

The formulae are:
`AD=AS=gate_width * diff_width`
`PD=PS=gate_width*2 + 2*diff_width`
`NRS=NRD= gate_width/cont_spacing`
diff_width and cont_spacing can be defind in file 'setup.spi'.

Not yet implemented, but soon :
If you have folded transistors, you can specify the folding factor 'm', which is a MOS property. By default, 'm' equals one, which means that the transistor is made of a single gate. Setting m=2 means the transistor is made of two gates, and so on. So,
`AD=AS=gate_width * diff_width / m`
`PD=PS=gate_width*2 + 2*diff_width / m`
`NRS=NRD= gate_width/cont_spacing / m`

Okay, these formulae are not exact, but better than nothing. If you want better ones, tell me. It is easy to change that.

disable

- MOS transistors have an ignore flag (which is saved in the .scm - file) which means that they are not netlisted. You can set this flag by executing the 'disable' command in the simulation menu. 'enable' means the the transistors are netlisted again.

- **only implemented in SPICECAD for linux** Normally, w and l have to be specified in the property list, otherwise, you get an error message. SPICE3 was modified by SPICECAD DEVELOPMENT and can now read in the transistor geometry in the model file. This is similar to PSPICE where you can specify w and l in the model file, too.

- The Transistor property 'm' means that MOS Transistors can be netlisted m times. This feature is useful when precision current mirrors are needed and when, at the same time, sub-micron MOS models are used which have a relationship between the transistor width w and the threshold voltage. So, assuming you need a 1:10 current mirror. Using two MOS transistors of gate lengths 0.5um both and gate widths 2um and 20um may result in a current ratio of 9 instead of 10 due to a threshold mismatch of several mV . When setting the 'm' property of the bigger transistor to 10 and setting both gate widths to 2 um, 11 transistors can be found in the netlist afterwards, and all transistors will have the same threshold voltage, wich gives the desired current ratio. The netlisting of m transistors in parallel can be suppressed by setting the variable 'multiply_mos_w_with_m' to 'yes' in the configuration file 'setup.spi' . In this case, the transistor width will be multiplied with m, and only one transistor is written into the netlist. This reduces the size of the netlist, but also reduces the precision of the current mirrors, as mentioned above.

To locally override the value of the global variable 'multiply_mos_w_with_m', you can use the following syntax at the MOS transistor property:
m '4|' means that m is set to 4 and that 4 transistors are written into the netlist regardless of the value of 'multiply_mos_w_with_m'.
m '4-' means that the transistor width will be multiplied by 4 and that only one transistor is written into the netlist regardless of the value of 'multiply_mos_w_with_m'.

Chapter 6

Simple editing: copying, moving, deleting , rotate and mirror , editing properties, editing nets, editing net properties, deleting parts of nets.

6.1 copy, move, delete , mirror, rotate

The function copy, move, delete and editprop are the most frequently used functions and are executable by hotkey : m=move, c=copy, d=delete, g=editprop. Rotate and mirror have to be executed from the main menu. Deleting of subnets (parts of a net) is done in the library menu ('del netsegment').

Moving, copying and deleting can be applied to all types of devices, to groups of devices created with the 'win add to group' -command (move group, delete group, copy group) or to special types of devices (move filtered,delete filtered, copy filtered) chooseable with 'select filter'.

With a left mouseclick, the device is selected and can be moved (move, copy).

With a second left mouseclick, the moving stops, and the device is placed.

Mirror and rotate require three, respectively two mouseclicks:
1) the first click selects the device to be mirrored or rotated.
2) the second mouseclick defines the point of rotation (rotate)
or defines the first point of the mirror axis (mirror).
3) the third point defines the second point of the mirror axis.

After the last mouseclick, all functions can be executed again :

a left mouseclick selects the next device.

Otherwise, if the right mousebutton is pushed, the command execution stops. Press the right mousebutton one again to pop up the current menu, or push a hotkey to execute the next command directly.

WARNING: THERE IS NO UNDO FUNCTION. YOU SHOULD PRESS THE 'CONTROL s ' HOTKEY AT LEAST EVERY 5 MINUTES TO PREVENT YOU FROM LOOSING DATA.

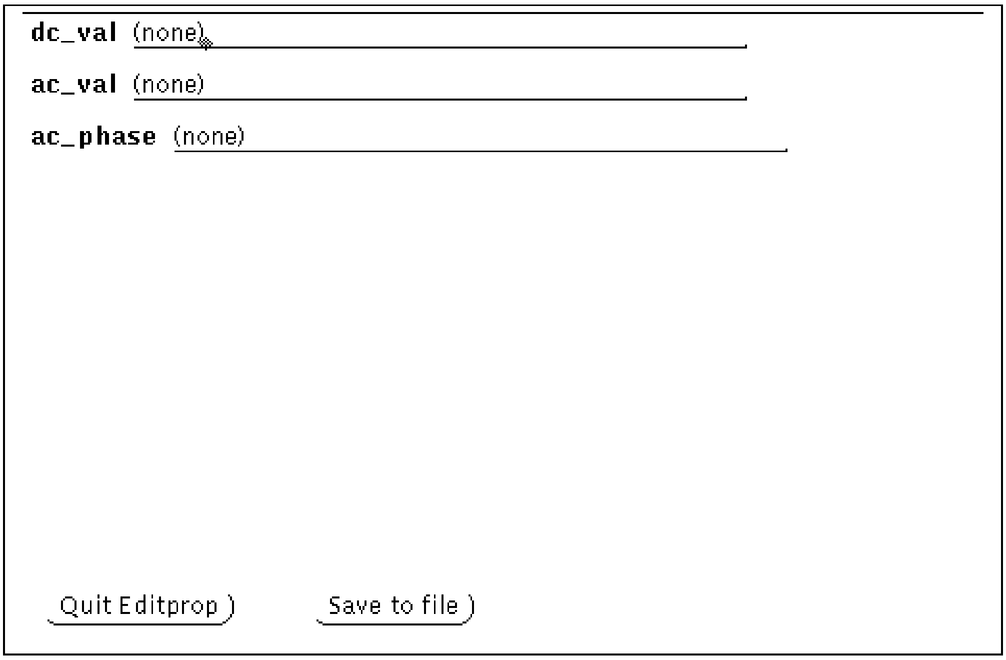
6.2 Edit properties

Editing properties is done via the 'editprop ' -command or via hotkey 'g'. If you execute this command, the property editing box will pop up.

The 'Quit'-Button closes the box if you have finished editing. The properties will be updated internally if you pess 'RETURN' or 'ESC'. **DON'T FORGET TO FINISH EACH INPUT WITH <RETURN> or <ESC>. OTHERWISE, YOUR INPUT WILL NOT BE STORED. TRY IT OUT: MAKE AN EDIT WITHOUT PRESSING 'RETURN', CLOSE THE PARAMETER BOX AND REOPEN IT.**

Pushing the 'Save to file'-Button will save the content of the property box into a file called 'varlist'. These properties can be read later by another design (schematic).

The property editing box will have two buttons 'up' and 'down' if the number of properties exceeds 10. This is the case when you edit parameter boxes or the options- box which defines SPICE3-'.OPTIONS' .



dc_val (none)

ac_val (none)

ac_phase (none)

Figure 6.1: The property editor dialog

The properties are internally held as a string. There is no syntax checking at all. Inputs like '12.0erte-12' instead of '12.0e-12' are stored like any other input.

Propertystrings have a length of 40 characters. So, if you want to use variables and expressions, be sure that the variable names and expressions are not too long. But don't worry, you will get an error message if the variable lengths are too long.

6.3 Edit nets, connectivity

Editing nets is started via 'create new net' in the library-menu od via hotkey 'n'. Nets are looking orthogonal. If you move the mouse after the first left-button mouseclick, you will see two orthogonal lines. Clicking the left button fixes the last position of these two lines and gives you two new lines. This way, you can draw a net containing as many lines as you want. The net editing is finished by clicking the right mouse button. If you want to change the edit style, use hotkeys 'h' and 'v'. 'v' means that the first line is drawn vertically, the second horizontally. 'h' means that the first line is drawn horizontally, the second vertically. If you have to draw a very long net, use hotkeys 's' and 'S' to snap the mouse to the next contact or net. 's' means the mouse is being warped to the next net or contact (whatever might be nearer to the actual mouse position). After warping, the program behaves as if you have hit the left mouse button: you will be able to continue net editing with two new lines. 'S' means that the mouse will be warped, and net editing will be finished at the snap position.

You should try 's' and 'S' in a dummy schematic before using it. If you hit 's' several times, and the mouse is warped to the same position every time, netparts of length zero would be created. Of course, this makes no sense, so, these netparts are removed . Every time this happens, an error message will appear on your xterm window like ' net rejected'. Don't worry, what you see on screen is what you get, so , if everything looks fine, it is okay.

The connectivity of schematics is defined by nets, global pins (gnd, vss,vdd,vss1, vdd1), named globals, named pins, ports and symbol contacts. Global pins and named globals may be placed on all hierarchy levels. However, it is necessary to place each used global pin at least once on the top level. For example, if you try to place voltage supply sources in a symbol(schematic) on a deep hierarchy level and connect them via global symbols to the rest of your schematic, you will receive an error message. **You must place all global pins on top level.**

Named pins and ports can also be placed in all levels, but different to global pins, named pins and ports are local. Ports are used to defined connectivity between different schematic levels. Named pins are used to define connectivity inside a single schematic sheet : nets which are attached to named pins of the same name are connected. Ports and pins are of the same object type: they are both of type 'pin'. So, you can use named pins instead of ports and ports instead of pins. Users of SPICECAD versions < 1.7.4 know the concept od named pins, so, they do not need to change their hierarchical schematics if they do not want to do so. Users of SPICECAD versions >= 1.7.4 should use ports to define inter-sheet connectivity and named pins for intra-schemtic connectivity.

Ports have been introduced in SPICECAD version 1.7.4 because they are schematic objects used in all schematic entry EDA tools. If you place a named pin more than once in a schematic, nets on the current hierarchy level will be connected to each other . This allows you to save place which you would need otherwise for large nets running from one end of your schematic window to the other.

6.4 Edit net properties

Nets have several properties. These are:
'alias','nodeset' and 'initial condition'.

'Alias' means that you can assign a name to a net, for example 'IN' , 'DATA1', and so on. This name remains invisible , so you have to keep in mind which nets contain aliases. You have to be careful if you merge nets: it is unpredictable what happens to the properties during merging. So, you should check the net properties after merging.

The alias name is used by commands which refer to nodes, for example, the noise simulation command or the optimization procedure. In both cases, one or two nets have to be defined to compute noise levels or differences between simulation results. These nets do not need to be defined via numbers because these numbers can change between subsequent netlistings (for example, if you decide to execute the 'save cont curs' command).

'Nodeset' is the initial dc voltage to help the simulator during dcop iterations to find a correct dc operating point solution. If 'nodeset' is unequal '(none)', a '.NODESET' - SPICE3 card is added to the netlist.

'Initial condition' is used to override the dc operating point solution at the beginning of a transient simulation . Consult your SPICE3 manual. If 'initial condition' is unequal '(none)', a '.IC' - SPICE3 card is added to the netlist.

Chapter 7

Advanced Editing: Grouping, Stretching, Filtered selection

Groups are groups of objects which can be moved, copied, deleted , rotated and mirrored simultaneously. Groups are defined using 'win add to group' in the main menu (add group in window). Groups are painted voilet and can be flashed several times using 'showgroup' if you are not sure whether an object belongs to a group or not. If a group just exists, single devices and nets can be added using 'add to group' . Single devices can be removed from a group using 'ungroup'. 'clear group' destroys (not deletes) a group, so that each of the former members of the group can be treated individually, again. Groups are destroyed automatically before each save process. 'copy group', 'move group' und 'delete group' copy, move and delete groups. However, you may use the normal copy, move and delete commands. 'Rotate' and 'Mirror' may also be performad on groups like as on single devices.

If the grouping is not sufficient for rearranging groups of devices, the 'stretch' command may help you, especially if there are connections inside a group of objects which have to be preserved. Stretching does not work on groups described before. Stretching is performed on objects inside a stretch area. This stretch area is created immediately after pushing the mainmenu->stretch button.

'Stretch' first defines a set of devices and nets which have to be stretched. Then, you can define a stretch line which indicates the distance by which all devices within the defined stretch set will be moved.

In difference to normal moving, 'stretch' will selectively move parts of nets so that the connectivity is preserved.

This is the theory. However, stretching sometimes fails , so

it is strongly recommended to save the schematic before doing a stretch.

The definition of the stretch set works as follows:

First, if you have selected 'stretch', all device boxes will be highlighted. With the left and right mouse button, you can define a closed polygon around the devices you want to stretch. Note that all four corners of the device box have to lie inside the polygon if the device shall be moved. Net parts which lie totally inside the stretch polygon will be moved. Net parts which lie partly inside, partly outside the strech polygon will be partly stretch, which means, the part inside will be modes, and the part outside remains at its old place. After having closed the polygon, you can define the stretch line. After doing this, the stretch set is moved horizontally or vertically by the horizontal or vertical component of the stretch line.

Nets will be stretched if their orientation is equal to the stretch direction, and if one point lies inside the stretch polygon and the other outside. nets will be moved if both points lie inside the polygon.

You should try this command to learn how it works before you really use it.

Chapter 8

Filter

Copying or moving of devices can be difficult if the schematic is drawn very dense. So , you can define which kind of device you want to copy, move or delete. 'select filter' lets you choose the kind of device, and

'mvfilt' (move filtered),
'cpfilt' (copy filtered) oder
'dlfilt' (delete filtered) move, copy or delete the chosen kind of device.

8.1 cut and paste

Cut and paste can be applied to existing groups. Cut means : moving the existing group into a cut-and-paste buffer. Copy means : copying the existing group into a cut-and-paste buffer.

Paste means: moving the content of the cut-and-paste buffer into the actual list of objects.

The question arises : what is the difference beween ordinary copying and cut-copy-paste?

The answer: Using cut-copy-paste can be used to transfer objects between hierarchy levels. As an example, you have created a symbol containing several analog filters. Later on, you see that there is a demand for one of these analog filters outside this symbol in another part of your circuit.

So, you

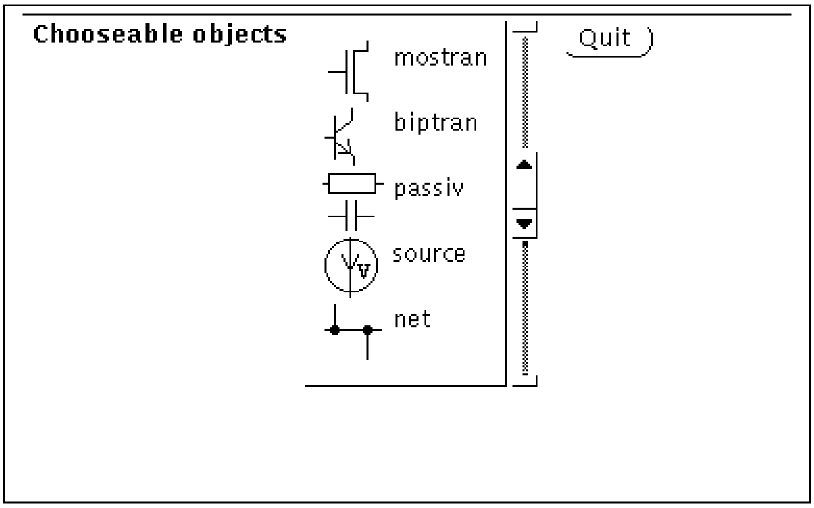


Figure 8.1: Selecting a move/delete/copy filter

- Enter the filter symbol schematic.
- Create a group from this filter.
- Copy the group into the paste buffer.
- Destroy the group.
- Leave the symbol schematic.
- Enter the hierarchy level where the filter is needed.
- Paste the filter into your schematic.
- Destroy the group.
- Save the modified schematic.

Ready.

Chapter 9

Loading and saving

9.1 Loading

If you want to load the schematic, symbol or ’other’ you edited last time, type simply ’cad’ to start the schematic editor. The name and type of this cell is in the file ’cad.lat’.

If you want to load a different existing schematic, a symbol or an ’other’, you can do that

- by typing ’cad -c schematicname’ for editing schematics,

- by typing ’cad -d symbolname’ for editing symbols,

- by typing ’cad -o othername’ for editing other elements.

This requires the name of the schematic in full length.

If you want to load any cell that you have edited in the last few weeks, and if you can’t remember the exact name, or if you are too lazy to type in a very long name, type ’cad -c new’ for editing schematics, where ’new’ is a keyword. The schematic entry screen will appear empty, and you won’t get any error message that file ’new.scm,’ could not be found. Now, klick at load last10’ in the main menu. In file ’last10.spi’, the last 10 different cell names you have edited are saved . Choose one name (or klick ’leave’ if you could not find the desired name and execute load schematic manually in the main menu) and begin editing.

If you want to edit a cell which is not in 'last10.spi', you need the file browser. Type 'cad -c/s/o new ' to start the editor . Then, pop up the load schematic, load symbol or load other- window and select 'Browse'. After a few seconds, a scrolling list pops up and displays all editable cells of the chosen type in the actual directory. Now, you can choose a cell. The window vanishes after choosing a name .

Additionally, you can type in the name of the desired cell in a text line which can be found on the load window. You have to choose that way of loading your cell if it is not on the actual directory, but in a library.

9.2 Saving

Perhaps, you edit a cell for the first time, that means, there is no file on the actual directory like 'name.scm', 'name.sym'or 'name.oth' .

Now, you did some changes and want to save it.

First, you can press CTRL s , which is the hotkey for saveit. This command may be executed for schematics, symbols, and others. It may be executed on top level in a hierarchical design and also on lower levels. If SPICECAD cannot find an existing cell on your actual directory or in the library system with the same name , you will receive a warning message that a new file has been created.

Second, you can pop up the 'save schematic', 'save symbol' or 'save other'-window and type in a new name on the text line. Again, if SPICECAD cannot find an existing cell on your actual directory or in the library system with the name typed in, you will receive a warning message that a new file has been created.

If you type in a name that is just existing, the save process will be stopped, any you will be queried whether you really want to overwrite the file or not.

A third possibility is to select the 'browse'-button and pop up a list of files. You can choose a filename out of this list.

But be careful: Overwriting files is not checked here because it is assumed that you want to do that.

I think, the safest methods for saving is 'CTRL s' and from the text line.

Besides, before saving, a backup file is created. So don't worry, be happy.

Chapter 10

Libraries and model paths

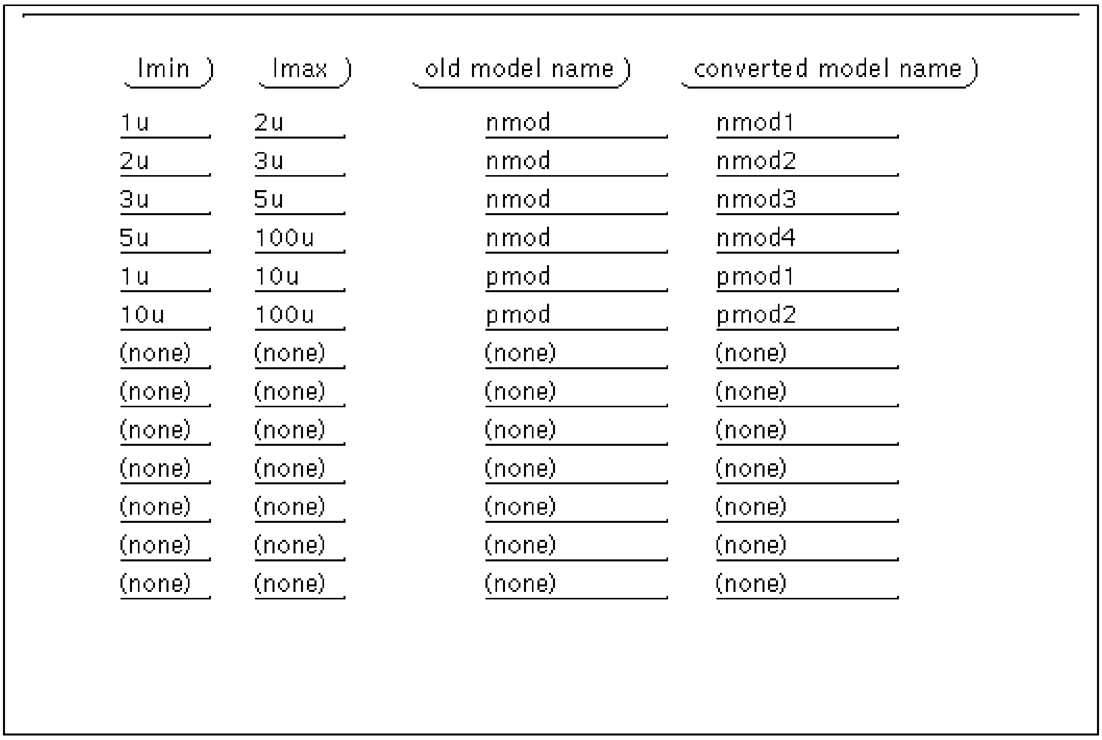
The cells and subcells (symbols and others) of your design can be stored on your

actual directory or in subdirectories, which are called libraries , wherever you want. You can specify the names of the libraries wherein SPICECAD is looking for symbols and others in the 'options'-dialog. The libraries are called 'librarypath1' ... 'librarypath5' . These paths can be absolute or relative. You may change these names during editing. To update the edited design, executed 'reload' in the main menu.

For example, if you want to try utilizing 741 OPAMPS in a design, and you see that they are not fast enough, change the search path from ./opamp_741 to ./opamp_op42 . If both paths contain a file called opamp.scm and opamp.sym , the replacement will be very fast. But do not forget to execute 'reload' to tell SPICECAD that there has been a change . Of course, SPICECAD could detect itself that a change has been made to the library properties. But, this reload operation is a very complex operation which has not been tested very extensively. So, before 'reloading', please save the design.

At netlisting time , SPICECAD includes model files (they always look like 'modelfilename.m') into the netlist. SPICECAD is looking for these files in the model paths. The model paths can be defined in the 'options'-dialog. The modelpaths are called 'modelpath1' ... 'modelpath3' . These paths can be absolute or relative.

It is possible to convert MOS model names during netlisting. This is in-



lmin	lmax	old model name	converted model name
1u	2u	nmod	nmod1
2u	3u	nmod	nmod2
3u	5u	nmod	nmod3
5u	100u	nmod	nmod4
1u	10u	pmod	pmod1
10u	100u	pmod	pmod2
(none)	(none)	(none)	(none)
(none)	(none)	(none)	(none)
(none)	(none)	(none)	(none)
(none)	(none)	(none)	(none)
(none)	(none)	(none)	(none)
(none)	(none)	(none)	(none)
(none)	(none)	(none)	(none)

Figure 10.1: The model conversion dialog

tereresting if you have piece-wise defined models which are valid only in a limited gate length range. Some simulators, like HSPICE and TITAN, have a built-in mechnnism which select the appropriate model if gate length ranges are given. Other modulators, like SPICE3, do not have such a meachnism.

SPICECAD will help you converting your model names automatically.

If you execute 'model conversion' from the simulation menu, a model conversion dialog box will pop up. There, you can enter gate length ranges and old and new MOS model names.

Chapter 11

Setup, Customization and Colors

SPICECAD is widely customizable. Colors, hotkeys and two user defineable menus provide a flexibility that many users require.

A little bit of customization can be done in the 'setup.spi'-file. Some of the variables which are stored here can also be changed interactively when running SPICECAD. Other things like colors cannot be changed interactively, so, you have to make your mind up about how you want SPICECAD to look like before you start SPICECAD. As an example, the net color is 'cornflower_blue'. On many machines, these so defined nets look deep blue. On some machines, these so defined nets are almost invisible. Changing 'color_palette_2' from cornflower_blue to LightCyan1 will change the net color.

The actual (default) contents of this file are:

```
modelpath1 /usr1/martinm/dsrfilt/titpar :  model path1
modelpath2 /usr1/martinm/cad2 :  model path2
modelpath3 /usr1/martinm/dsrfilt :  model path3
modelpath4 /usr1/martinm/dsrfilt :  model path4
modelpath5 /usr1/martinm/dsrfilt :  model path5
librarypath1 ./ :  library path1
librarypath2 ./
librarypath3 ./
librarypath4 ./
```

librarypath5 ./ : library path5
helpfile help_eng.spi : file including short interactive help
linewidth 0 : line width for drawing lines (nets)
plotwin_width 620
plotwin_height 420 : width and height of plotwindow
doubleclickspeed 10 : time distance of two mouse button clicks to be recognized as 'doubleclick'
mainwin_width 800
mainwin_height 500 : width and height of main window
color_palette_1 black : color palette . See comments below this table
color_palette_2 cornflower_blue
color_palette_3 green
color_palette_4 cyan
color_palette_5 red
color_palette_6 magenta
color_palette_7 brown
color_palette_8 gray51
color_palette_9 gray98
color_palette_10 blue
color_palette_11 spring_green
color_palette_12 LightCyan1
color_palette_13 OrangeRed1
color_palette_14 magenta4
color_palette_15 yellow
color_palette_16 white : SPICECAD uses 16 different colors. The color names, which must consist of a single word (colors like 'spring green' can be used by typing 'spring_green'. These names are converted internally. You cannot use names which contain underscores), can be changed here.

plotwindow_color_palette_1 black
plotwindow_color_palette_2 cornflower_blue
plotwindow_color_palette_3 green
plotwindow_color_palette_4 cyan
plotwindow_color_palette_5 red
plotwindow_color_palette_6 magenta

plotwindow_color_palette_7 brown
plotwindow_color_palette_8 gray51
plotwindow_color_palette_9 gray98
plotwindow_color_palette_10 blue
plotwindow_color_palette_11 spring_green
plotwindow_color_palette_12 LightCyan1
plotwindow_color_palette_13 OrangeRed1
plotwindow_color_palette_14 magenta4
plotwindow_color_palette_15 yellow
plotwindow_color_palette_16 white : The 16 colors used in the plot window can be chosen independently from the colors in the schematic window.

box_color 5 : The color number of the contact boxes (5=red)
instance_color 3 : The color which is used for devices and device properties (green)
net_color 2 (cornflower blue)
text_color 9 (light gray)
other_color 14
symbol_color 13
plottype eps : or hpgl, raster
eps_scalefactor 1.0 : Factor with which an EPS plot is scaled with. By default, a plot of the schematic window is scaled up to the size of a DIN A4 (29cm x 21cm) page. You can reduce or increase the size of this scaled plot by this second scaling factor.
printoutfile tmp : beginning of printout file (tmp.eps, tmp.hgl , tmp.ras ...).
simulatorhost wibke : Machine where HSPICE or TITAN will be executed on. SPICE3 will always run on the machine where you start SPICECAD.
simulatorhost1 susan : not implemented yet
simulatorhost2 susan: not implemented yet
simulatorpath /aiv_project/marbic/cad2 : Path where HSPICE or TITAN netlists and simulation results are stored.
plotwindow_background_is_white yes : 'yes' for white plot window background, 'no' for black bachground

sharedlists no :no: all symbols create internal data during load :yes: symbols with the same name share their data The disadvantage is that backannotation of simulation data is not possible on lower hierarchy levels than toplevel.

want_pulldownmenus no : If you want a menu bar and pulldown-menus, type 'yes'.

warning_if_no_mos_l_or_w_given yes : If you forget to specify the geometry of a MOS transistor, you will receive an error message after netlisting. If you use the extended SPICECAD MOS models (only available under linux) wherein you can also specify the geometry, you should set this flag to 'no'.

preserve_spice3_files no : during parametric simulation, the SPICE3 netlist will be overwritten at each step. If you want SPICECAD to copy all netlists to directory /tmp and rename them, set this parameter to 'yes'.

netlist_geometry no : SPICECAD can generate MOS parameters AS,AS,PD,PS, NRD, NRS automatically. Depending of parameters diff_width and cont_spacing, AS,AS,PD,PS, NRD, NRS will be calculated and netlisted if they are not set explicitly and if 'netlist_geometry' is set to 'yes'.

display_geometry_instead_properties no : during netlisting, property values are derived. You can display these values, which is useful whan you have used variables excessively. Say 'yes' if ypu want these derived values, 'no' if you want to see property strings.
display_initial_condition yes : Lets you display the 'initial condition' proterty at nets.

show_elembox_during_mousemove no: is only for debugging want_digital_mos_symbols no : All these Digital designers who find SPICECAD useful and who do not want to miss their beloved MOS transistor symbols should set this parameter to 'yes'.

want_mousecomwindow yes : Novice users will find it helpful to display this window where they can find information about the actual meaning of mouseclicks. Experienced users will consider this window as a waste of spare window area and will want it to disappear. Type 'no' and only the schematics window will appear when starting SPICECAD.

needs_scaling no : I think it had a meaning, but I have forgotten what it is. You better leave this parameter as it is.

pop_titanready_window yes : every time HSPICE or TITAN have finished the simulation, they send a message to SPICECAD. SPICECAD will pop up a window if this parameter is set to 'yes' upon receiving this message.

pop_menu_on_functionkeypress yes : enables the function keys

coordinates yes : shows the mouse position in the upper left corner of the screen. These coordinates are 'world' coordinates, which means that the physical mouse button coordinates are scaled into internal coordinates.

move_orthogonal yes : Moves can only be performed vertically or horizontally. This can be overridden in options1- Menu (F8)

copy_orthogonal yes : Copies can only be performed vertically or horizontally.This can be overridden in options1- Menu (F8).

diff_width 2.000000e-06 : width of diffusion area of mos transistors. With this value, the source and drain diffusion

```
areas are calculated if they are not specified explicitly in
the property-menu.

diff_width_n 2.000000e-06 :  width of diffusion area of nmos
transistors.  Overrides diff_width for NMOS transistors.

diff_width_p 2.000000e-06 :  width of diffusion area of pmos
transistors.  Overrides diff_width for PMOS transistors.

cont_spacing 5.000000e-06 obsolete
resistive_area_width 5.000000e-07 :  distance between contact
edge and poly gate at MOS transistors.  With this parameter,
the number of resistance squares is calculated if they are not
specified explicitly in the property-menu.

bjtsatvolt 3.000000e-01 :  Lets you specify the saturation
voltage of bipolar transistors.  This value is used by the 'Bad
Devices' function in the simulation menu.

font fixed :  used font.
default_editor emacs :  this editor will pop up when you
execute 'lookat netlist'
default_printer hq1e_eng :  when you execute 'send tmp.ps to
printer', the printer hqe1_eng will be chosen (-Phq1e_eng)

stop stop :  End of file
```

Some comments about colors: different colors are used for different elements in the schematic. Contact boxes are red, devices are green, selected nets become yellow, text labels are grey etc. If you do not like these colors, you can specify different colors. All you have to do is edit the 'color_palette' entries. But, which names can you choose? Simply look at file 'rgb.txt' which can be found on you workstation on directory
/usr/openwin/lib/rgb.txt.
Or /usr/X11R6/lib/rgb.txt.

If you cannot find this file, simply look at the ’rgb.txt’ file which is included in the spicecad.tgz file.

The background color of the schematics window cannot be changed. I like it as it is. The background color of the plot window is white by default. Due to the fact that some people have problems with this setup, you can set the background color of the plot window to black, too.
The appropriate entry is
’plotwindow_background_is_white yes’.
Set it to ’no’, and you get a black background.

More customization:

Due to SPICECAD user requests, user-defined menus and hotkeys were introduced.

Two user-defined menus are available now, where each can contain up to 20 entries. The menus are defined in files ’userdef1.spi’ and ’userdef2.spi’ which have to be stored in directory $SPICECAD_HELPDIR . The files have a very simple format. As an example, file userdef1.spi is given below:

```
redraw window
help
gnd
nmos
```

This file provides you with four entries ’redraw’,’help’,’gnd’ and ’nmos’.

These entries must match the entries in the other, fixed menus, with the exception that hotkey definition can be left away.

So, instead of typing

’redraw window:r’ ,

’redraw window’ is sufficient . Of course, if you want to see ’redraw window:r’ in your user defined menu, you can type in that. Whatever you prefer.

Even more customization:

The menu entries which are not reachable by the pre-defined hard-coded hotkeys can now be reached using user-defined hotkeys. The Hotkey definition is done in file 'hotkeys.spi' which has to be stored in directory $SPICECAD_HELPDIR

The file format is very simple again and looks like this:

```
x
n4mos
CTRL j
gnd
p
vdd
w
vdd1
```

In lines with odd numbers, the key is defined , like ' x' ,'CTRL x' and 'ALT x'. (Please do not forget the leading blank when defining single characters, and do not forget the single! blank when defining CTRL and ALT hotkey sequences). In lines with even numbers, the menu entry is defined which is issued when hitting the hotkey. These menu entries must match the names in the fixed menus.

For example, 'CTRL j' starts the placement of a 'gnd' symbol.

Simple but efficient.

Chapter 12

Simulation

You just did some simulation , I hope.

Now, this process will be explained a little bit more detailed.

Before simulation, a netlist must be written which contains all devices, analysis commands and options. The flag 'SPICE3' in the 'options'-dialog has to be set to SPICE3 if you want to use SPICE3 as your simulator. This is the default. If you push 'HSPICE' or 'TITAN' instead, the created netlists will have slightly different formats.

Netlists will be written automatically before simulation if you changed something in your schematic. You will get a netlisting statictics in your xterm-window.

If you want to write a new netlist explicitly, execute 'netlist' in the simulations menu. This has to be done if you change model files in the model directories.

Now, lets assume that the netlist failed.

This happens if you forgot to fill in required properties, or if there are unconnected contacts or contacts which are connected only once.

All devices which are errouneous are marked wich a circle inside. All contacts which are errouneous are marked wich a circle around. Nets which are connected to only a single terminal will be highlighted.

If you did a 'redraw window', you can display all these errors again by using 'showerr' in the simulation menu.

Errors which are not device- or contact specific are displayed in the error window.

Now, lets assume that there is a message ' Net number 34 is connected only once'.

That means that the node number 34 appears only once in the whole netlist.

Now, you need the position of net 34. Ooh, Murphy says that this is always an invisible net on the lowest hierarchy level.

No problem. Choose 'findnode' in the simulation menu, and you get it. Even in the deepest hierarchy levels.

NOTE : correct the error , save with 'CTRL s' and return to the toplevel with 'totop' in the symbol menu or with Hotkey 't'.

Another message is 'transistor M34 has incomplete properties'.

Its just the same: execute 'findname' , type 'M34' , and you get it. Don't forget to save after correcting the error.

Now, lets assume that you have corrected all errors.

What could still be missing?

You might have forgotten the gnd- symbol, the parameter-box, at least one analysis-box.

Correct that.

Maybe, a library path or model path is missing or wrong. You can correct that in the 'options'-dialog.

Okay, now we want to run a simulation.

Execute 'runsim' or type 'CTRL r'. Note: SPICE3 is so fast for small circuits that it takes a longer time to select 'runsim' with the mousebutton than running the simulation itself. You don't believe it ? You will, if you have at least a 486 running at 33MHz or a station.

With 'select V(x)' or 'select I(x)' you can select one or more nets or terminals whose voltages or currents you want to watch. Voltages and currents cannot be, however, displayed in the same plotting window. To tell SPICECAD that you want to plot currents, you have to select 'currents' in the 'What?'-pullright-menu.'voltages' is the default. The selection of nets or terminals is ended with a click at the right mousebutton. To plot the results, select 'plot dc' at DC-transfer simulations, 'plot tran' for transient simulations. 'plot

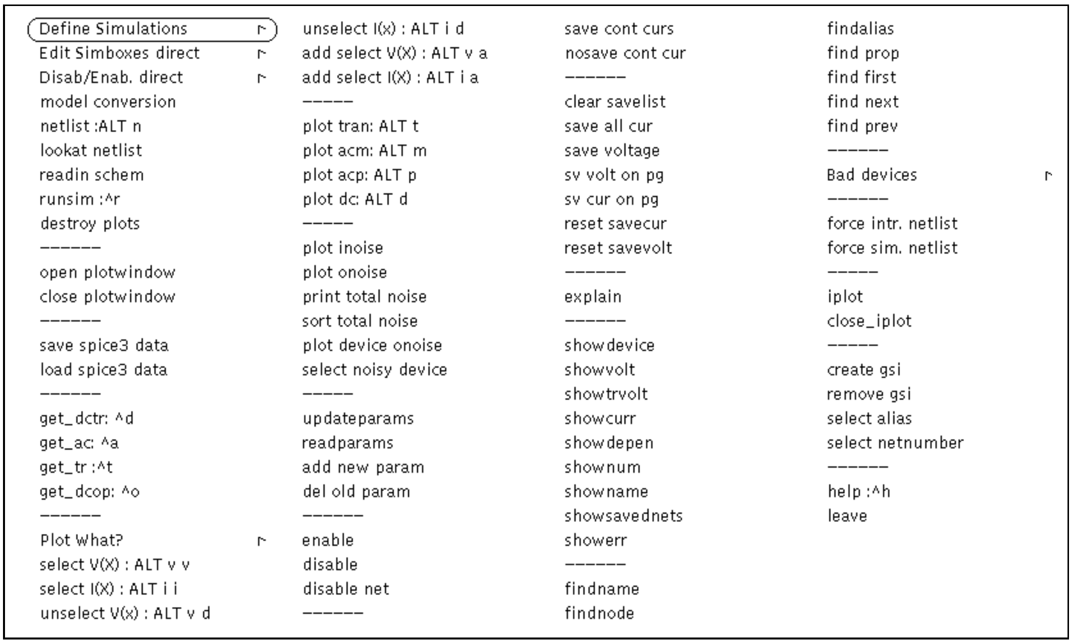


Figure 12.1: The simulation menu

acm' plot the magnitude of the small-signal voltage at AC-simulations, 'plot acp' plots the phase.

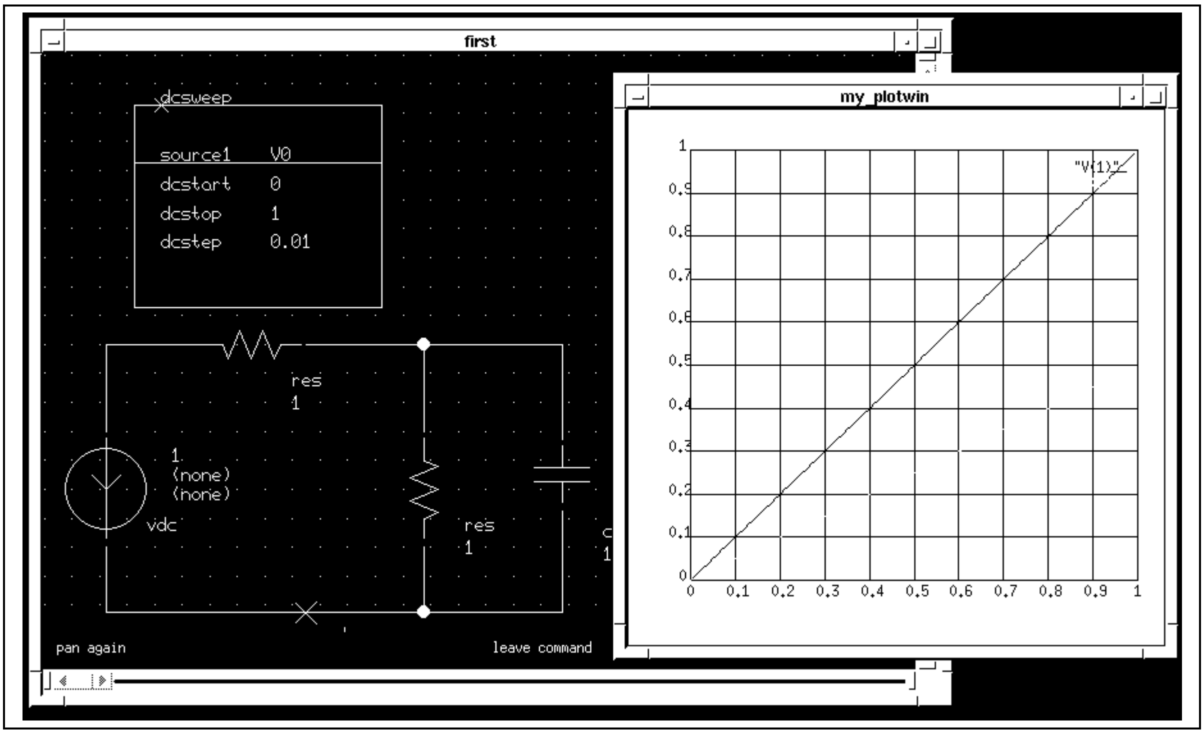
The plotwindow opens automatically. If you want to stretch if , please open it by hand with 'open plotwin' and stretch it before the first plot.

12.1 Simulation types

Simulation types are DC transfer, DC operating point, AC- small signal, transient , temperature and noise. Sensitivity, distortion and fourier are missing.

Each analysis is represented by an analysis-box. If the schematic is becoming too 'overcrowded', you can hide the boxes by selecting 'nosimu' in the 'options'-dialog.

Analyses (boxes) can be enabled and disabled like devices. In the case of disabling , the analysis will not be performed. Some kinds of analysis must be called explicitly, although their analyses-boxes are placed in the schematic.



source1 V0
dcstart 0
dcstop 1
dcstep 0.01

Figure 12.2: The plot window

These analyses are parametric sweep, monte carlo simulation and optimization. Theses analyses can be called from the 'parametric ' menu only.

This type of analyses is no intrinsic SPICE3 - simulation, it is a special SPICECAD- simulation which requires subsequent netlisting and running.

12.2 Parameters

SPICEDCAD understands parameters. If you want to use a parameter within a property, take the name of the parameter and write it between '{}'. Thats all.

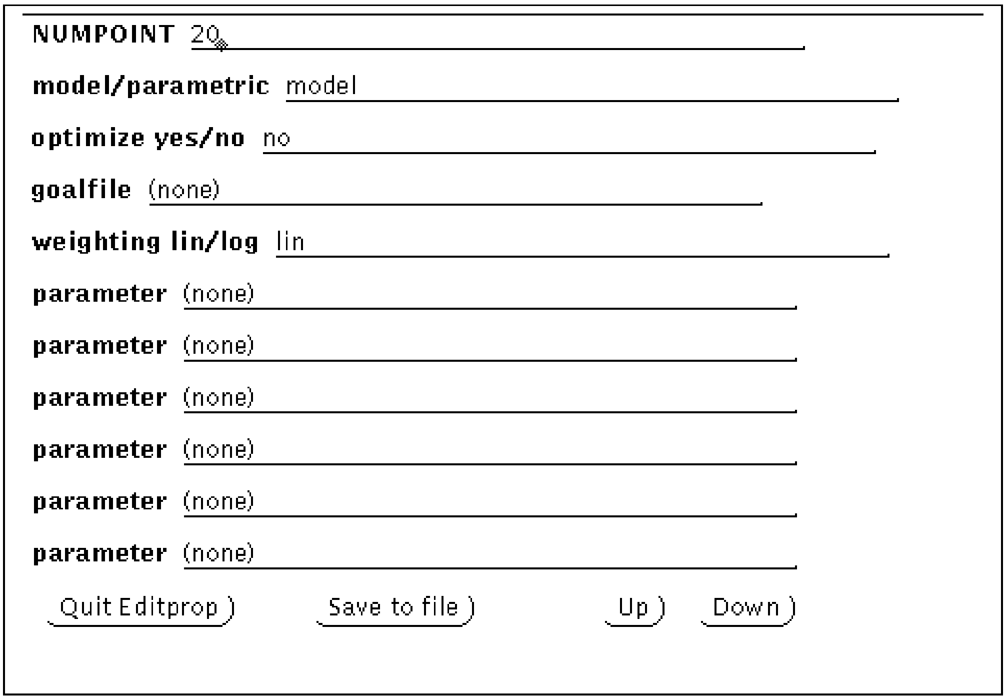
After all, if you don't have a parameter box, place one (use the 'place things' pullright menu entry 'params' within the simulation menu) and edit the parameter values. If you just have a parameter box, execute 'updparams', and you will get the latest parameters after the last update. If you want to add a single new parameter and its value to the parameter box, execute 'add new param' within the simulation menu. A window will pop up, where you can specify the new parameter name and its value. If the parameter just exists, the existing parameter value will be changed, and you will receive a warning message. This command is useful if you specify new parameters which are undetectable by 'updparams', for example monte carlo model specifications (monte carlo capacitor model: 'flat/{stdevcap}') which contain parameters.

Parameter values can be read from a file. The filename is fixed to 'varlist'. Its format is as follows:

param1 value1 param2 value2 param3 value3 .
.
.
param_n value_n (none) (none)

In addition to that, you can copy a schematic file which contains a parameter box to 'varlist'. You can also read in this file. To create a file 'varlist', you can push the button 'Save variables to file varlist' withing the property editing box. The contents will be immediately saved in file 'varlist' for later use in other designs.

NOTE: Reading parameters from a file will only update existing parameters. Parameters which are not present in the actual parameter box will be dropped.



NUMPOINT 20

model/parametric model

optimize yes/no no

goalfile (none)

weighting lin/log lin

parameter (none)

parameter (none)

parameter (none)

parameter (none)

parameter (none)

parameter (none)

Figure 12.3: Properties of the Monte Carlo simulation

Parameters can form expressions, for example , you may write '100k' , '{resparam}' or '{wid/leng*sheetres}'.

There must be only one parameter box inside the design. If you have forgotten a parameter box on a deeper hierarchy level, its contents won't be regarded. Instead, you will get a warning message.

12.3 Monte-Carlo-Simulation

The Monte-Carlo-Simulation is also a paramteric simulation.

To perform such a simulation, you have two possibilities. First, you can edit the 'model'-property of passive components like resistors, capacitors and so on or the model-property of bipolar and MOS transistors. The model property consists of two parts: the kind of statistical distribution (flat or gauss) and the standard deviation. Both values have to be separated with a '/' or a '#', i.e. 'flat/0.1'. 'flat/0.1' means that the standard deviation is

10'flat#0.1' means that the standard deviation of the property is 0.1 .

Setting the model-property to '(none)' means that the device property value remains constant during simulation. It is allowed to use expressions specifying the standard deviation, i.e. 'gauss/stdevcap '.

The second possibility to specify parameters with distribution is to fill out the entrys in the Monte Carlo Simulation Box. The model and standard deviation for all parameters are identical if not set explicitly.

Please check that you don't use the same parameter in the simulation box and as a property parameter inside a device with a specified monte carlo model. There would be unpredictable interference between these two parameters.

To start the Monte Carlo Simulation, there must be a Monte Carlo Simulation Box inside the schematic.

Starting the simulation is done inside the 'parametric*'-menu selecting the 'run monte carlo'-Button.

Selecting 'V(x)' lets you choose a single node you can plot after simulation . Selecting 'I(x)' lets you choose a terminal.

Circuit optimization is the third parametric siulation type and will be treated in a subsequent chapter.

Up to version 1.2, a Monte Carlo Simulation could be performed only on device parameters like gate lengths and widths and component values like resistances, capacitances etc.

12.4 Using the pre-processor cpp, Netlist filtering

Now, beginning with SPICECAD version 1.3, a Monte Carlo Simulation can be performed on almost any parameter, for example MOS or bipolar model parameters. To avoid writing a lot of additional code, I chose the C preprocessor 'cpp' to help SPICECAD to do what the user of SPICECAD wnat to do.

So, running a Monte Carlo Simulation over a MOS model works as follows:

First, place a Monte Carlo Simulation box and fill in the properties :
`NUMPOINT : Number of simulations`

```
model/parametric :  parametric
optimize yes/no:  no
goalfile:  (none)
weighting lin/log :  lin (not used)
std deviation :  {stddev}
parameter1 :  WHATEVER
model1 :  (none) or flat/{stddev1}
parameter2 :  DONTKNOW
model2 :  (none) or flat/{stddev2}


...
modeln :  DOITSO
parametern:  (none) or flat/{stddevn}
modeln+1 :  (none)
parametern+1 :  (none)
```

stddev , stddev1 ... stddevn are the values for the standard deviation, WHATEVER, DONTKNOW ... DOITSO are strings. These strings have to re-appear in the MOS model file like this:

```
.MODEL scnanjss NMOS LEVEL=3
+ UO=3.842E+02
+ VTO= WHATEVER
+ GAMMA=7.930E-01
+ PHI= DONTKNOW
+ NSUB=7.000E+17
+ ETA=0.000E+00
+ DELTA=0.000E+00
+ THETA=6.218E-02
+ VMAX=1.773E+05
+ KAPPA=3.629E-01
+ LD=1.000E-07
+ TOX= LATERNE
+ NFS=2.620E+11
+ CGSO=3.3E-10 CGDO= DOITSO PB=0.6 FC=0.5
+ CJ=0.54E-3 MJ=0.31 CJSW=3.3E-10 MJSW=0.33 KF=7.7E-28
+ AF=0.96
```

WHATEVER, DONTKNOW, LATERNE and DOITSO also have to appear in the parameter box. You have to type in these names and their values by hand, because they are not detected during an 'updateparams' process.

Now, we are ready to run the simulation.

Before doing this, you have to pop up the parametric menu and push 'use cpp (model sweep)'. Now, push 'run monte carlo'.

What will happen , now, is that SPICECAD netlists the ciruit as follows:

```
*default_ueberschrift Wed Sep 11 22:39:39 1996 parametric  0.000000e+00  0.000000e+
#define WHATEVER 6.781304e-01
#define DONTKNOW 6.391833e-02
#define LATERNE 8.648665e-09
#define DOITSO 2.095932e-10
*
*
*

M0 4 4 0 0 scnanjss l=2.000000e-06 w=8.482375e-06 as=3.600000e-11  ad=3.600000e-11
V0 3 0 PULSE ( 1.65  3.3  0 .1n .1n 10n 20n )
V1 1 0 DC 3.3
M1 2 3 1 1 scnapjss l=2.000000e-06 w=4.500000e-05 as=2.700000e-10  ad=2.700000e-10
M2 2 3 4 0 scnanjss l=2.000000e-06 w=6.000000e-06 as=3.600000e-11  ad=3.600000e-11

.MODEL scnanjss NMOS LEVEL=3
+ UO=3.842E+02
+ VTO= WHATEVER
+ GAMMA=7.930E-01
+ PHI= DONTKNOW
+ NSUB=7.000E+17
+ ETA=0.000E+00
+ DELTA=0.000E+00
+ THETA=6.218E-02
+ VMAX=1.773E+05
+ KAPPA=3.629E-01
+ LD=1.000E-07
+ TOX= LATERNE
+ NFS=2.620E+11
```

```
+ CGSO=3.3E-10 CGDO= DOITSO PB=0.6 FC=0.5
+ CJ=0.54E-3 MJ=0.31 CJSW=3.3E-10 MJSW=0.33 KF=7.7E-28
+ AF=0.96

.
.
.
```

You can find this file under '/tmp/orig ' . You should have access to this path, otherwise, funny things will happen. Next, SPICECAD generates a shell script which runs 'cpp' and 'sed'. The name of the shell-script is '/tmp/convert_spice3_file.sh'.

After executing this shell-script, which overwrites the original file, the file which will be read in into SPICE3 looks as it should.

make sure that you do not use parameter names which equal SPICE3 or MOS model keywords. So it is mandatory to use words which do not appear elsewhere inside the SPICE3 netlist ..

12.5 Netlist filtering

A more generalized concept is the usage of self written filter scripts to perform any operation on the netlist whatever you want.

In the 'options'-dialog , the entry 'script file' exists . By default, 'script file' is set to '(none)'. If set to any other value, SPICECAD will interpret this value as the name of a script file. SPICECAD will call this script file via the 'system' command. The call is as follows:
system('sh $script_file $netlistfile').

So, inside this script , you can do whatever you want: calling other 'perl' scripts, C programs, unlimited .

But, if you want to change the contents of the netlistfile, make sure you do not destroy it. A script could look like this:

```
cat $1 | awk -f /usr1/martinm/replace_caps_by_resistors.awk > /tmp/whatever1
cp /tmp/whatever1 $1
```

By the way, if you run 'cpp' AND a script, 'cpp' is executed first, the script afterwards.

12.6 Circuit optimization

There are three optimization methods implemented, the classical gradient method (steepest descent) , 'rapid thermal annealing' or 'Monte Carlo' optimization and, last but not least, exhaustive search .

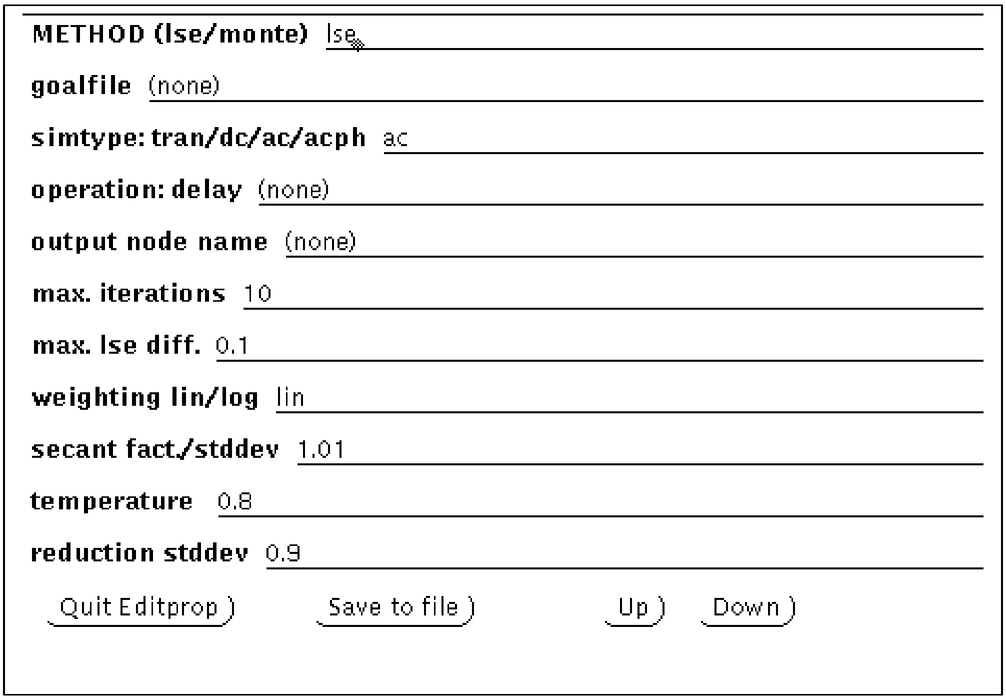
The optimization can optimize parameters defined by the user.

At the beginning of an optimization, the user has to place an optimization box in the schematic, on the top level of the hierarchy . It is a prerequisite that all devices which shall be optimized have parameter properties instead of real value properties:
{res} {optres}
instead of
{res} 100k
to optimize a resistor value.

A circuit optimization can optimize small-signal, dc-transfer or transient simulation results. Also, group delay optimization is possible.

To determine the error between simulated and desired result, the user has to create a so-called goal-file. This file has four columns . The first column is the independent variable time, frequency or sweep value. The next two columns are the lower and upper bounds of the desired results. If the simulated results lie within these bounds, they do not produce any errors. 'Error' means that the cost-function which measures the difference between the achieved result and the desired result is the classical quadratic error function: the sum of the squared differences between achieved and desired result .

Upper and lower bound of the goal definition inside the goal file may be identical. The fourth column of the goal file is a weighting factor with which the local error contribution will be multiplied (weighted).



METHOD (lse/monte) lse

goalfile (none)

simtype: tran/dc/ac/acph ac

operation: delay (none)

output node name (none)

max. iterations 10

max. lse diff. 0.1

weighting lin/log lin

secant fact./stddev 1.01

temperature 0.8

reduction stddev 0.9

Figure 12.4: Properties of the Optimization processor

The determination of the optimization algorithm is done in line 'method' in the property box.

Valid methods are 'lse' which means the gradient method or 'monte' which means 'rapid thermal annealing'. When placing an 'exhaustive search' optimization box, the method is 'exhaustive'.

Next, the simulation type has to be specified. 'tran', 'acm' , 'acph' or 'dc' are valid.

Next, a reference node has to be specified . The voltage at this node will be taken by the error function to derive the quadratic error. NOTE: Don't forget to edit the alias name of the desired net with 'editnet'.

Next, the maximum number of iterations ('max. Iterations') and the minimum quadratic error ('max. lse diff.') have to be specified . These parameters control the end of the optimization interations on success or on exceeding the allowed number of iterations : no success.

A little preprocessing of the simulation and goal results is possible specifying the 'weighting lin/log' parameter. 'lin' is default , 'log' should be chosen during small-signal optimizations.

Last, the optimized parameters have to be specified. A maximum of 20 parameters is allowed within the iteration loop.

The simplest method is the steepest descent or 'gradient search' where a set of new coefficients is derived from the first derivatives of the squared error by the optimization parameter.

A more difficult to handle method is the 'rapid thermal annealing' or 'Monte Carlo' method where new sets of coefficients are determined by using a random number generator. After applying these random numbers to the initial optimization parameters, the squared error is determined after a new simulation. If the squared error is smaller than before, the new coefficient set is accepted. If the result is worse, then, depending on another stochastic experiment, the likelihood that this set of new coefficients is accepted, too, is greater zero. This way, there is a certain chance that the optimizer, when trapped in a local minimum, can still find the global minimum is it exists. The likelihood with which a 'bad' set of coefficients is accepted is directly proportional to (1-exp(-T)) , the temperature T. When T is very high, it is very likely that a bad set of coefficients is accepted. Whether it is accepted or not is determined by the equation

rand() < (1-exp(-T))

rand() is the outcome of a random variable which lies between 0 and +1 . When T is zero, rand() < (1-exp(-T)) is always false, and bad coefficient sets are not accepted. T has to be set by the user to an appropriate value. It is always difficult to say which is a useful starting value, so, it is highly recommended to play with this type of optimization before seriously using it. If T is too high, too many bad coefficient sets are accepted, and it can even be possible that convergence is not achieved. If T is too low, the optimizer can be trapped in a local minimum.

After a certain number of successful reductions of the optimization error, T is reduced step by step, so that the achieved results are 'frozen'. This reduction factor can be defined by the user, too. Again, it must not be too high or too low.

The last optimization method is exhaustive search. Here, up to 10 parameters can be optimized. By specifying lower and upper bound of an n-dimendional orthogonal search space, SPICECAD will loop over all dimensions between lower and upper bounds and determine the squared error. This method is very primitive, but using today's available Pentium II PCs or SUN ULTRA II workstations, results are achievable overnight or over the weekend.

Chapter 13

Backannotation of simulation results

The plotting results of ac-,dc- and tran- simulations are displayed in a second window called 'my plotwin'. Pushing the right mousebutton pops up a menu.

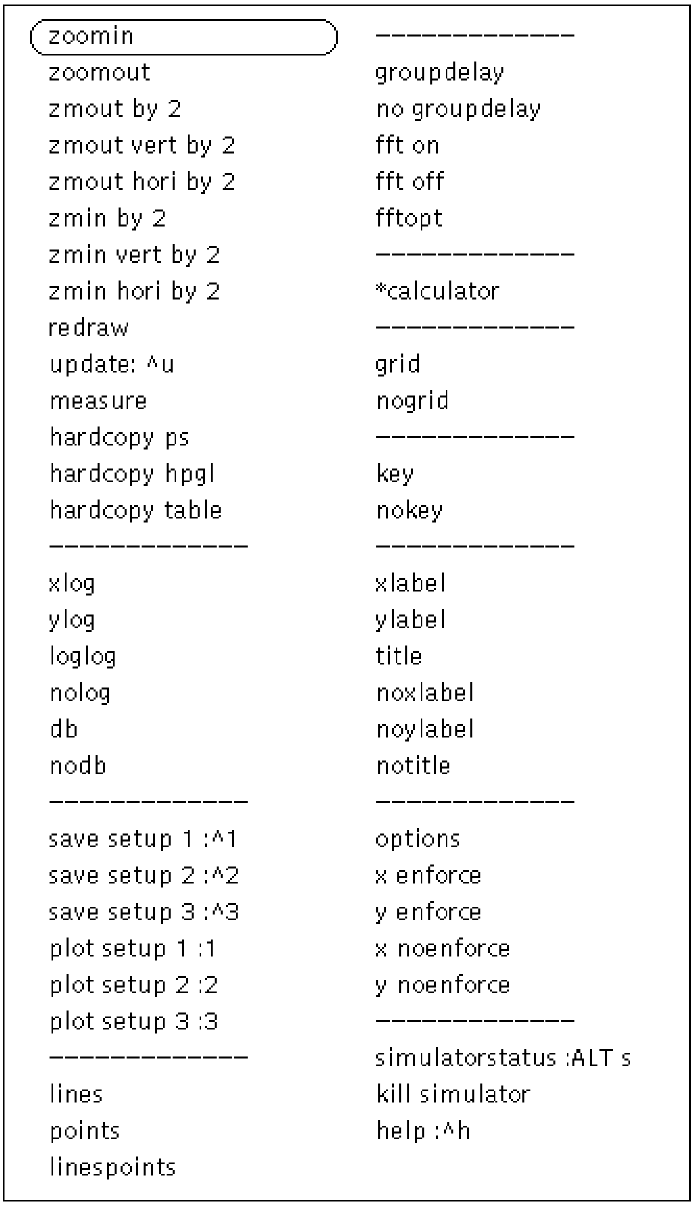


Figure 13.1: The plot window menu

Command	Meaning
’zoomin’ :	zoom in to get more detailed view
’zoomout’ :	zoom out (zoom to fit) to get less detailed view
’zmout by 2 :	zoom out by a size factor of 2
’zmout vert by 2’:	zoom out vertical by a size factor of 2
’zmout hori by 2’:	zoom out horizontal by a size factor of 2
’zmin by 2’:	zoom in by a size factor of 2
’zmin vert by 2’:	zoom in vertical by a size factor of 2
’zmin hori by 2’:	zoom in horizontal by a size factor of 2
’redraw plot’:	Repaint currrent window
’update: û’ :	Is only useful when running the SIEMENS simulator TITAN. The command fetches the latest simulation results if there is a simulation ongoing in the background
’measure’ :	Clicking twice at a result curve in the plotting window will show you the (x1,y1) , (x2,y2) values of that curve at these points.
’hardcopy ps’ :	plots the current plot window into a file ’tmp.ps’ using POSTSCRIPT format.
’hardcopy hpgl’ :	plots the current plot window into a file ’tmp.hgl’ using HPGL format.
’hardcopy table’ :	plots the current plot window into a file ’tmp.txt’ in rows and columns for later post-processing.
’save as goal’ :	saves the current plot as a goal file for later use in an optimization , which means, in a format x - y -y - w where x is the current x-axis (independent variable), y is the y-axis (dependent variable) and w is the weighting factor. y is saved twice, so that the lower and upper tolerance band for use inside an optimization are equal. The weighting factor w is set to one.
’xlog’ :	logarithmic x-axis.
’ylog’ :	logarithmic y-axis.
’loglog’ :	logarithmic x -and y-axes.
’nolog’ :	linear axes.
’save setup n’ :	save your current plotting setup (line style, ffft on/off) in buffer no. n
’plot setup n’ :	restoring the saved plotting setup from buffer no. n and plotting.
’style lines’ :	connecting the simulation points with lines.
’style points’ :	using symbols to display the simulation points
’style linespoints’ :	connecting the simulation points with lines and using symbols to display the simulation points
’groupdelay’ :	determines the group delay when displaying ac results.
’groupdelay off’ :	no further group delay calculation.
’fft on’ :	
’fft off’ :	

Table 13.1: Plotwindow menu commands

'key' :	Displays information about plotted variables in the upper right corner of the plt window.
'nokey:	Suppresses display of Information about plotted variables in the upper right corner of the plot window
'fftopt' :	FFT- options. You can choose a start point, and an ending point for the calculation of the FFT.You can also choose a window type. Available are triangular, Hanning and Hemming.
'plot options' :	Pops us dialog box where setting FFT and text options is possible.
'x enforce' :	Enforce x axis range.
'y enforce' :	Enforce y axis range.
'x noenforce' :	Do not enforce x axis range.
'y noenforce' :	Do not enforce y axis range.
'grid' :	Displays a grid.
'nogrid' :	No display of a grid.
'xlabel' :	Text output at the x-axis.
'noxlabel' :	No text output at the x-axis.
'ylabel' :	Text output at the y-axis.
'noylabel' :	No text output at the y-axis.
'title' :	Output of a heading.
'notitle' :	No output of a heading.

Table 13.2: Continuation of plotwindow menu commands

Chapter 14

Back-Annotation of DC-Operational-Point-Simulations in the schematic

It is possible to display net voltages, terminal currents and device parameters in the schematic.

Net voltages are displayed after selecting the menu entry 'showvolt' and selecting a net. With a second left mouseclick, you can place the text label whereever you want. This is different compared with schematic entries from EDA companies where the position of these labels is chosen automatically. This is easier for the user, but the schematic window is then filled with huge amounts of information from which only 1% is really needed. In SPICECAD , the user decides what he wants to know.

Displaying of terminal currents works identical: chose 'showcurr' and klick at a terminal contact of a voltage source. If you have executed 'select cont curs' and chosen several objects inside the schematic window in whose terminal currents you are interested in before running the simulation, you can also click at these terminals. These terminals are printed in yellow instead of in red to remind you that the currents through these terminals have been stored.

If you want to display device information (where device means sources, transistors, resistors ,capacitors etc.), execute 'showdevice'. Each time you select a device, a 'SPICE3F5-show'-command is issued by SPICECAD which displays the desired information in the xterm-window in which you have started SPICECAD.

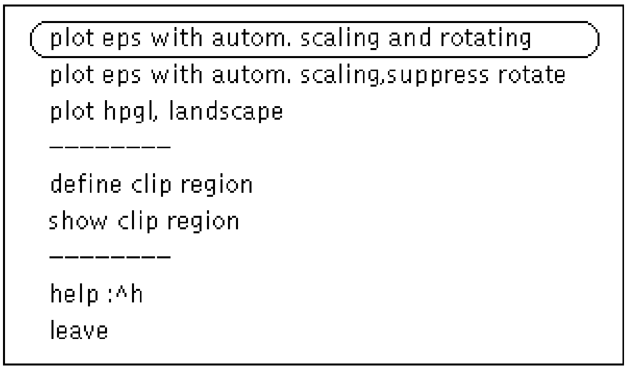
If you use other simulators than SPICE3, the 'showdevice' command is a little bit limited. Here, 'showdevice' only displays the dc operating point of MOS and bipolar transistors.

Chapter 15

Plotting schematics

The plotmenu 'Plot*' is reachable via hothey 'F6' or 'ALT 6' . Plotting of schematics is possible in HPGL and POSTSCRIPT-format.

Selecting the format is done in the 'options*'- box which is reachable in the main menu or via hotkey 'F7'. The default plotting format is Postscript. The filename which is created afterwards begins with 'tmp' and ends with '.eps' or with '.hgl'. You can select other filenames in the 'options*'-Box by changing the entry 'printoutfilename'. Postscript- plots are always being fitted to A4 - format (29cm x 21cm) and are being rotated if this is necessary. To prevent the plot from being rotated, you can choose 'supress rotate' in the plot-menu. An additional scaling factor can always be supplied in the 'options*'-box . By default, this factor is set to 1.0. The resulting plot has a size of (29cm x 21cm) times this scaling factor.



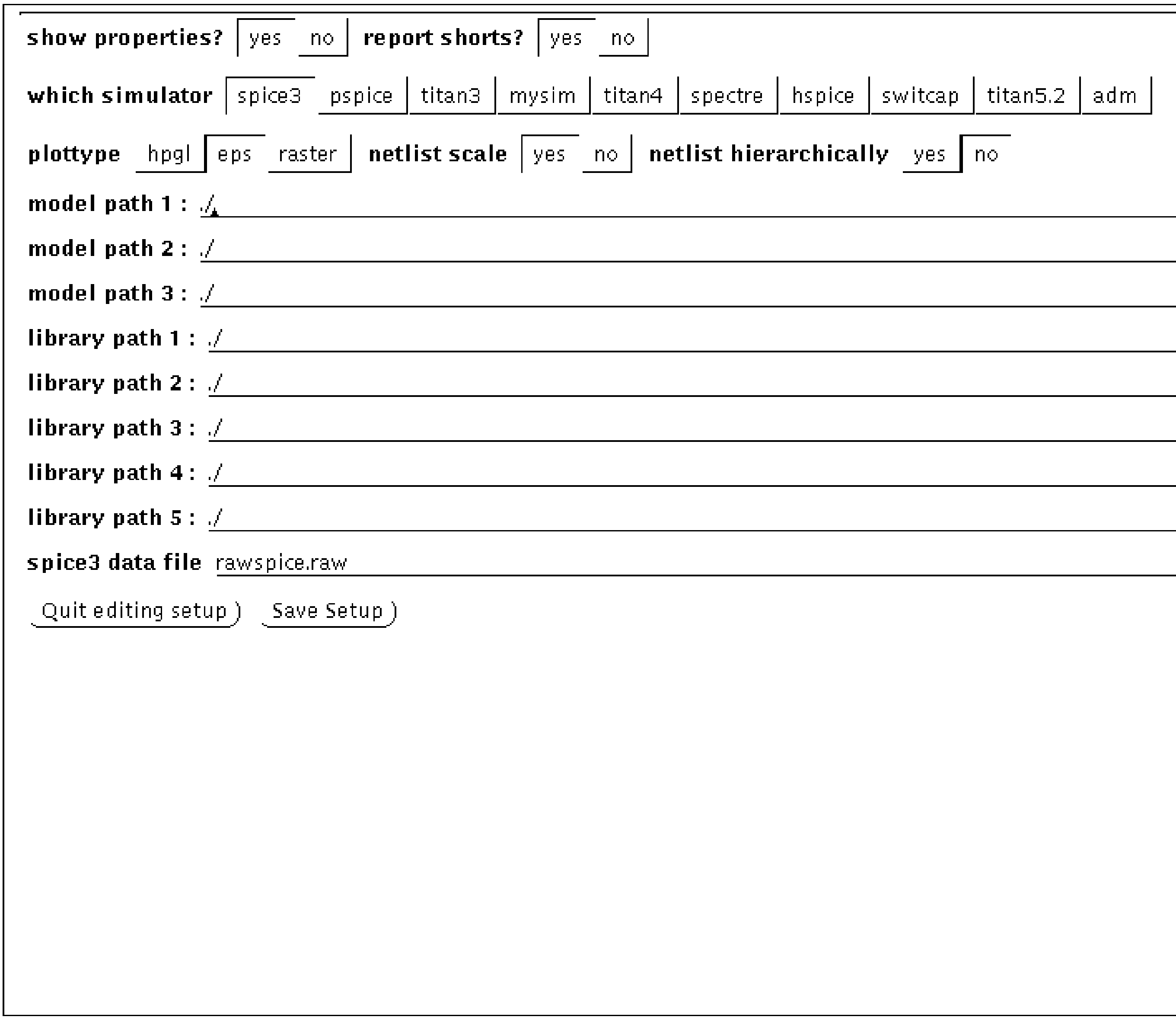
Chapter 16

Options - dialogs

The behaviour of SPICECAD as well as library and model search paths can be influenced a lot. The three options dialogs (can be started using the 'option 2/3' menu entry in the main menu) are the interfaces to modify everything which can be modified. Below, you can find the screen dumps of the option dialogs.

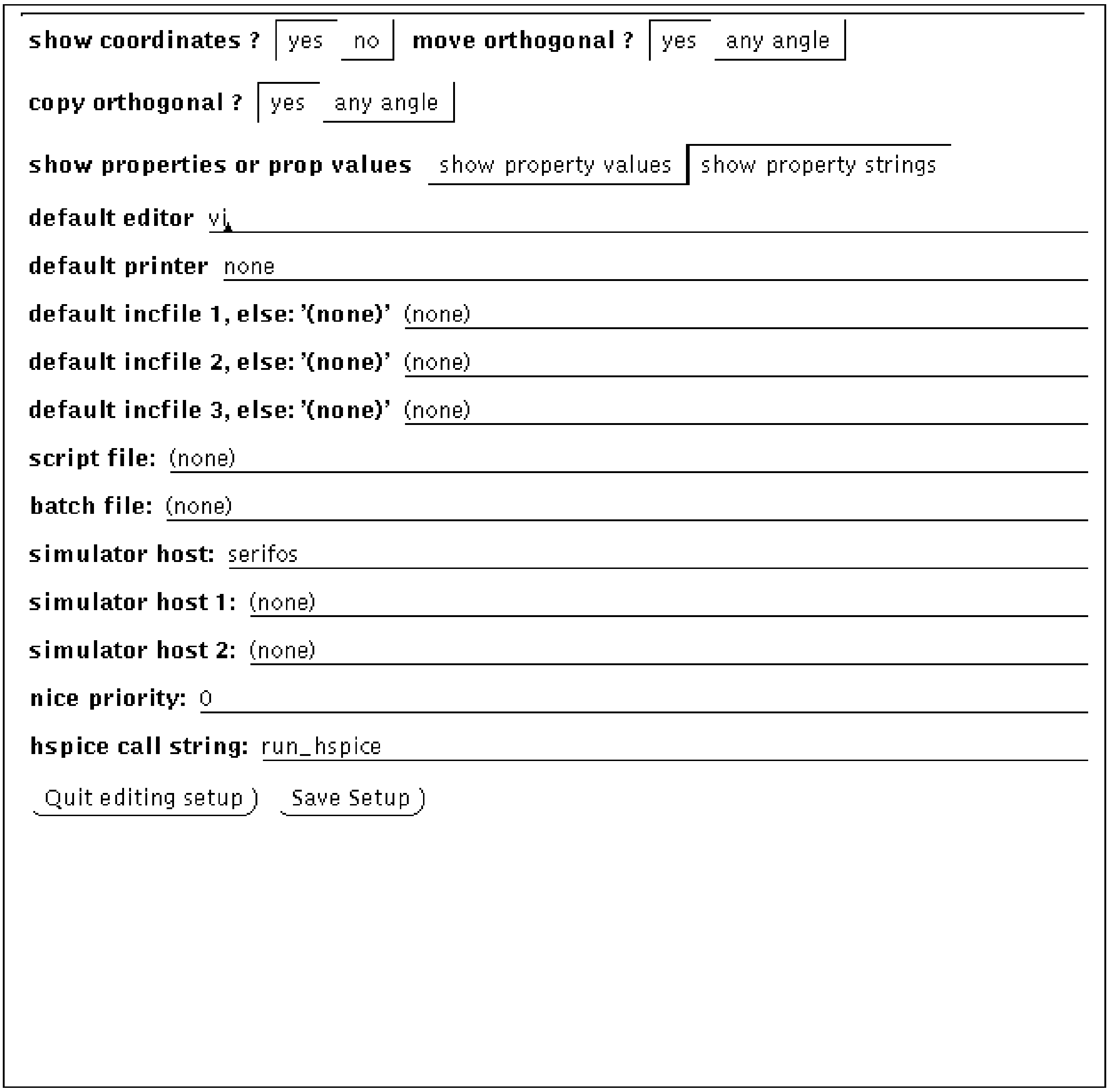
Every dialog box has two buttons 'Quit editing setup' and 'save setup'. 'Quit editing setup' finished the editing session. All edits which were done by editing a string have to be finished pressing 'RETURN' to copy the contents of the edited line into the corresponding internal string variable. 'save setup' saves the currrent setting into the setup file 'setup.spi'. .

Here is the meaning of the dialog entries:



model path 1 : ./
model path 2 : ./
model path 3 : ./
library path 1 : ./
library path 2 : ./
library path 3 : ./
library path 4 : ./
library path 5 : ./
spice3 data file rawspice.raw

Figure 16.1: First options dialog



show coordinates ? yes no **move orthogonal ?** yes any angle

copy orthogonal ? yes any angle

show properties or prop values show property values show property strings

default editor vi

default printer none

default incfile 1, else: '(none)' (none)

default incfile 2, else: '(none)' (none)

default incfile 3, else: '(none)' (none)

script file: (none)

batch file: (none)

simulator host: serifos

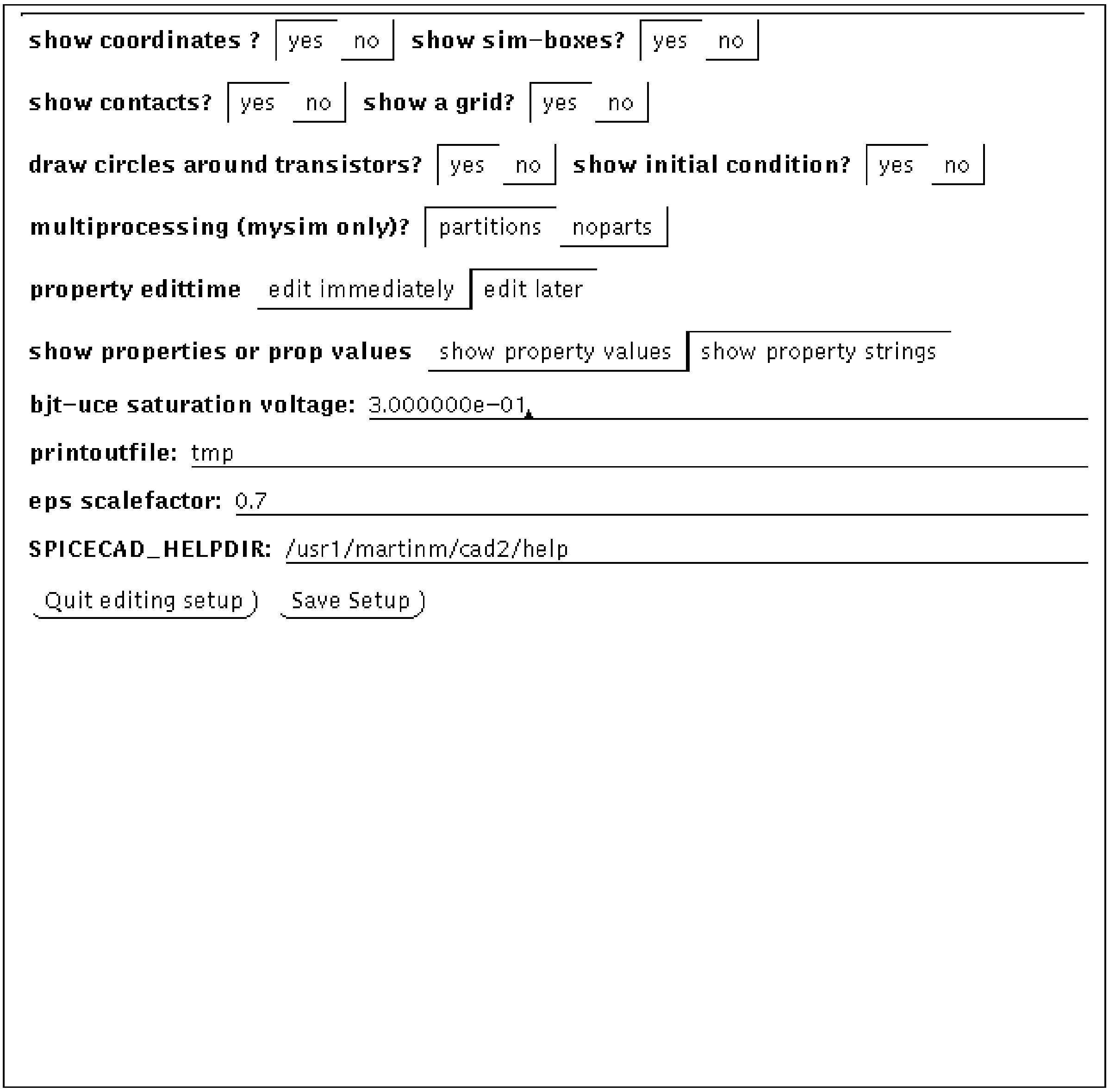
simulator host 1: (none)

simulator host 2: (none)

nice priority: 0

hspice call string: run_hspice

Figure 16.2: Second options dialog



show coordinates ? yes no show sim-boxes? yes no

show contacts? yes no show a grid? yes no

draw circles around transistors? yes no show initial condition? yes no

multiprocessing (mysim only)? partitions noparts

property edittime edit immediately edit later

show properties or prop values show property values show property strings

bjt-uce saturation voltage: 3.000000e-01

printoutfile: tmp

eps scalefactor: 0.7

SPICECAD_HELPDIR: /usr1/martinm/cad2/help

Figure 16.3: Third options dialog

entry	meaning
'show properties' :	properties are printed in the schematic yes/no
'which simulator' :	'spice3' , 'pspice' , 'hspice', 'titan' , 'mysim', 'adm' etc. Choice of a simulator. SPICE3 is built-in. A PSPICE interface is not yet available. Any volunteers out there who want to create one? Contact me!
'plottype' :	'hpgl' and 'eps' are available
'netlist scale' :	determines whether the SCALE entry at resistors, capacitors and inductors shall be netlisted or not.
'netlist hierarchically' :	For LVS (layout-versus-schematic) reasons, it might be useful to generate a hierarchical netlist instead of the default flat netlist. During hierarchical netlisting mode, cross-probing does not yet work properly.
'model path 1..3' :	The model paths can be changed here
'library path 1..5' :	The library (SPICECAD search) paths can be changed here
'spice3 data file':	Here, the filename into which SPICE3 results are written into can be changed. By default, this filename is set to 'rawspice.raw'.

Table 16.1: Meaning of the entries of the 'options' dialog window

entry	meaning
'move orthogonal' :	If selected, moving of devices can be done in horizontal or vertical direction exclusively. If not set to 'yes', devices can be moved in any direction
'copy orthogonal' :	If selected, moving of devices during device copying can be done in horizontal or vertical direction exclusively. If not set to 'yes', devices can be moved in any direction
'netlist nrs,nrd,as,ad' :	If set to 'yes', for each MOS transistor a set of geometry information will be derived and added to the netlist if the properties nrs,nrd,as,ad etc. are NOT set. This geometry information is derived from the MOS transistor gate width, the diffusion width and the contact spacing which is defined in the setup-file 'setup-spi'
'use digital/analog MOS symbols' :	Here, the style (look) of the MOS transistors inside the schematic can be chosen.
'show contactnames':	If set to 'yes', contact names are displayed.
'default editor':	Several functions will come up with an editor window which diplays useful information (for example, the 'lookat netlist' . The editor which will be used then can be defined here.

Table 16.2: Meaning of the entries of the 'options1' dialog window

'default printer' :	The function 'tmp.eps to lpr' will send the file 'tmp.eps' to the printer using the UNIX command 'lpr -P 'default printer' tmp.eps
'default incfile 1..3, else: '(none)' :	During netlisting, additional files can be added to the netlist. These files might contain simulation or storage commands which are not supported by SPICECAD.
'script file':	The netlist can be filtered by a UNIX script before it is executed. This process is described on page 70, section 'Netlist Filtering'. If filtering is not desired , this entry should be set to '(none)'
'batch file' :	SPICECAD can be used interactively or in batch mode. Using SPICECAD in batch mode is described on page 123 in chapter 'batch execution'
'netlistpath (absolute)'	This entry is obsolete
'simulator host' :	Remote simulator simulations using HSPICE or TITAN are carried out on machines which names can be defined here.
'simulator host 1..2':	It is planned to allow simultaneous remote simulations in the future. This feature is not yet implemented.
'nice priority':	Remote simulations can be assigned a nice piority to reduce the CPU power consumption. The nice priority value must be a positive integer between 1 and 19. 1 reduces the CPU power consumption only very little, 19 reduces it largely.
'hspice call string':	When HSPICE is used as a simulator engine, the 'run_hspice' script is called by default. If you want to use different HSPICE versions, you can copy and rename, customize and modify the script. The script name you wish to use can be defined here.

Table 16.3: Meaning of the entries of the 'options1' dialog window: continuation

'show coordinates ?' :	Display of coordinates in the upper left corner of the schematic display
'show sim-boxes' :	analysis boxes are shown in the schematic yes/no.
'show contacts' :	device-contact-boxes are shown in the schematic yes/no
'show a grid' :	grid in schematic is shown yes/no
'draw circles around transistors' :	circles around transistors are shown yes/no
'show initial condition ' :	the initial condition of a net is shown inside the schematic yes/no
'multiprocessing (mysim only)' :	should always be set to 'no' . Otherwise, the netlist created will not be SPICE3-conform.
'property edittime' :	'edit immediately','edit later' Properties can be edited directly after placing a new device or later.
'show properties or prop values' :	If set to 'show property values', the property expressions are evaluated during netlisting, and they are shown correctly after the next redraw. If set to 'show property strings', the expression strings are shown all the time.
'bjt-uce saturation voltage:' :	The function 'bad devices' is looking for MOS transistor in triode region and bipolar transistors in saturation. Although saturation is operating point dependent, the search for bipolar transistors in saturation is done simply by looking at their collector-emitter voltages. If this voltages is below 'bjt-uce saturation voltage', the bipolar transistor is considered to be in saturation
'printoutfile' :	by default set to 'tmp'. Printoutfiles, which contain the EPS plot of the schematic, can be renamed this way. The file postfix, '.eps', can't be changed.
'eps scalefactor' :	Normally, the EPS plot of a schamatic is scaled in a way that it fills a whole A4 page. To reduce the size of a plot , this scaling factor can be used. Any scaling factor of less than 1.0 reduces the size of the schematic plot.
'SPICECAD_HELPDIR' :	Here, you can set the path where SPICECAD can find the help files. By default, the environment variable is read and copied into this position.

Table 16.4: Meaning of the entries of the 'options2' dialog window

Chapter 17

Hierarchical design

SPICECAD allows fully hierarchical design over as many hierarchy levels as you need. This is made possible due to recursive programming of the netlisting and load functions. Hierarchy levels are represented by symbols, which can be created by the user him/herself or automatically.

17.1 Generation of symbols

17.1.1 Generation of symbols from scratch

First, leave SPICECAD and start it again typing

```
Spice 1 –> cad -s firstsymbol
```

The '–s' tells SPICECAD that you want to edit a symbol instead of a schematic, and that the name of the symbol is 'firstsymbol'.

A symbol is made up of graphics primitives like lines, polylines, circles, rectangles and text. All these primitives can be placed wherever you want. What is needed , additionally, are contacts to provide connections to the underlying schematic levels which the symbol represents.

There is one graphics primitive that has a special meaning. It is a text object, and the textstring has to be '@prop'. If such an object appears at the list of objects, the symbol name and symbol properties will be drawn at the (x,)–coordinate of the text objet. The text object itself will be hidden.

If you don't place such a text object, the coordinate where the symbol name and properties will be printed are chosen by default, which does not look so good very often. If you have forgotten to place this text object, you can do it afterwards, this does not corrupt the database integrity.

Up to now, you know how to create the look of a symbol. Now, we connect the symbol to the underlying schematic. For each contact in the symbol, there has to be a port (SPICECAD version < 1.7.4 : named pin) on the corresponding schematic level.

So, if you have finished painting your symbol, don't forget to place an appropriate number of contacts with appropriate names using the 'editprop'– command.

After saving the symbol (CONTROL s or 'save sym') ,let's call it 'firstsymbol' , in the file 'firstsymbol.sym', this file has to be edited by hand if you want to add local properties to the symbol.

What are local properties?

Local properties are useful in a design which contains several symbols of the same type. For example, consider a MOS inverter chain wherein the widths of the transistors increase by a constant factor from stage to stage.

Now, you have two possibilities.

First, you can create as many different inverter symbols as you need for the chain, and each of them has different transistor properties. Second, you create an inverter symbol with local symbols 'wpmos' and 'wnmos'. You place this symbol as often as you need and edit the local symbol properties afterwards.

The property syntax is as follows (Example):

```
.
.
.
stop
; End of graphics primitives section {wn}
; Begin of local property section {ln}
{wp}
{lp}
(none)
; End of local property section , End of file '*.sym'.
```

To give your symbol local properties, you have to add the lines after 'stop' in the symbol file. The number of lines is unlimited, but don't forget to add '(none)' at the end of the local property section. You should be careful when choosing the number of properties when you create the symbol. If you want to increase this number afterwards or if you see that this would be useful, you have to edit the schematics files wherein this symbol appears by hand to add the missing properties. Sorry for this inconvenience.

This could be a possible point for a future enhancement.

17.1.2 Automatic creation of symbols

Symbols can also be created automatically if you are editing a schematic which you want to use in deeper hierarchy levels. Symbols can be created from groups or from schematic views.

Automatic creation of symbols from schematics

First, pop up the symbol menu and choose 'gen autom. symbol'. If your schematic contains less than two ports (SPICECAD version < 1.7.4 : named pins) , you will receive an error message.

If your schematic seems to be correct, a menu will pop up .

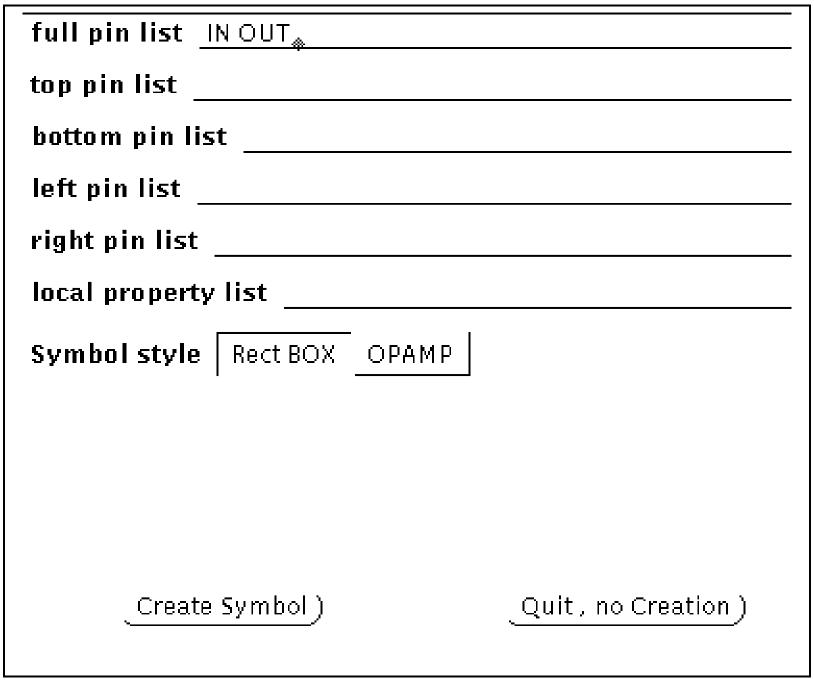
Here, you can decide where to place contacts (left, right, upper, lower pins) and how the symbol should look like: like a box or like an opamp. If you choose the opamp, there must not be more than two right contacts. If you want to give local properties to the symbols, you can add them in the last line.

The creation of the symbol works very fast, so you don't have to wait for some actions after having pushed the creation button.

Afterwards, you should check the look of the new symbol and correct it whereeever you want.

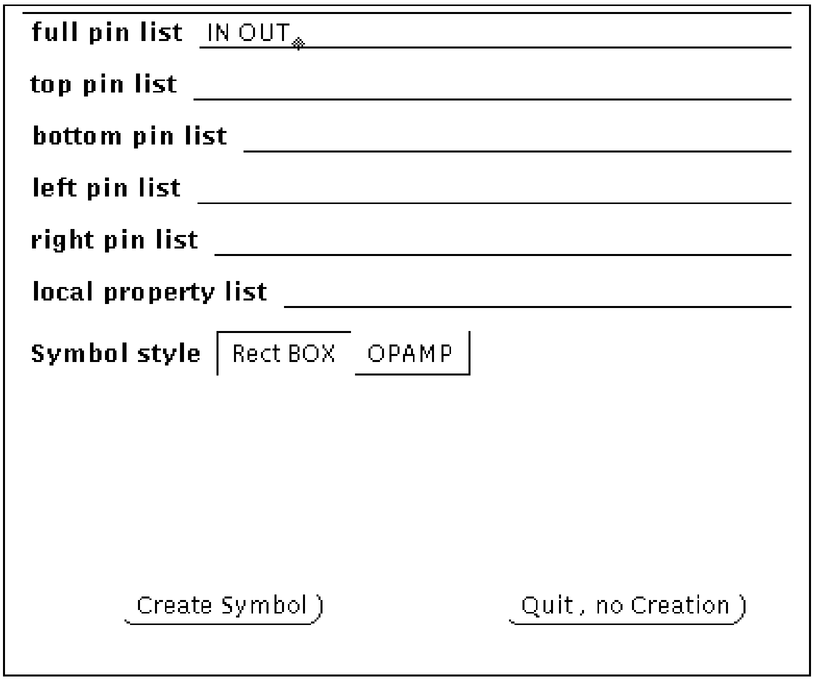
Automatic creation of symbols from groups

First, create a group using 'win add to group' in the main menu. The group should contain at least two ports (SPICECAD version < 1.7.4 : named pins



full pin list IN OUT

Figure 17.1: Automatic symbol generation dialog


full pin list IN OUT

Symbol style Rect BOX

Figure 17.2: Automatic symbol generation dialog

). Then, pop up the symbol menu and choose 'gen symbol from group'. If your group contains less than two ports, you will receive an error message.

If the group seems to be correct to SPICECAD, a menu will pop up .

Here, you have to give a name to the new symbol/schematic , and you can decide where to place contacts (left, right, upper, lower pins) and how the symbol should look like: like a box or like an opamp. If you choose the opamp, there must not be more than two right contacts. If you want to give local properties to the symbols, you can add them in the last line.

The creation of the symbol works very fast, so you don't have to wait for some actions after having pushed the creation button.

After pushing the creation button, the content of the group will be written into a file called 'newname.scm', where 'newname' is the name you have chosen in the menu. A second file, 'newname,sym', will be created which contains the symbol and property information. After doing this, the group will be erased from your schematic and will be replaced by the symbol you have created.

Afterwards, you should check the look of the new symbol and correct it whereever you want.

If you do not want that the group disappears from the schematic, you should copy the group to a second screen where you can perform the symbol creation.

17.2 Rules for local and global connections within schematics.

The connectivity of schematics is defined by nets, global pins (gnd, vss,vdd,vss1, vdd1), named globals, ports, named pins and symbol contacts.

Global pins and named globals may be placed on all hierarchy levels. However, it is necessary to place each used global pin at least once on the top level. For example, if you try to place voltage supply sources in a symbol(schematic) on a deep hierarchy level and connect them via global symbols to the rest of your schematic, you will receive an error message. **You must place at least one global pin of each kind or name on top level if you use them on deeper hierarchy levels.**

Named pins and ports can also be placed in all levels, but different to global pins, named pins and ports are local. When a port (SPICECAD version < 1.7.4 : named pin) is placed on a hierarchy level n and is related to a symbol contact with the same name, the net on level n attached to this port will be connected to a net on level n-1 which is connected to the symbol contact. For upwards-compatibility, SPICECAD does not differ between named pins and ports. So, every named pin can be replaced by a port, and each port can be replaced by a named pin, both with the same name. To make schematics easier understandable, ports, however, should be used to define inter-sheet connectivity, and named pins should be used to define intra-sheet connectivity.

If you have placed a named pin more than once on different nets, these nets (on the same hierarchy level) will be connected to each other. This allows you to save place which you would need otherwise for large nets running from one end of your schematic to the other.

What is also not allowed on deep hierarchy levels are property, option and analysis boxes. Erase them if you change a top level schematic to a deeper level. Otherwise, you will get error messages .

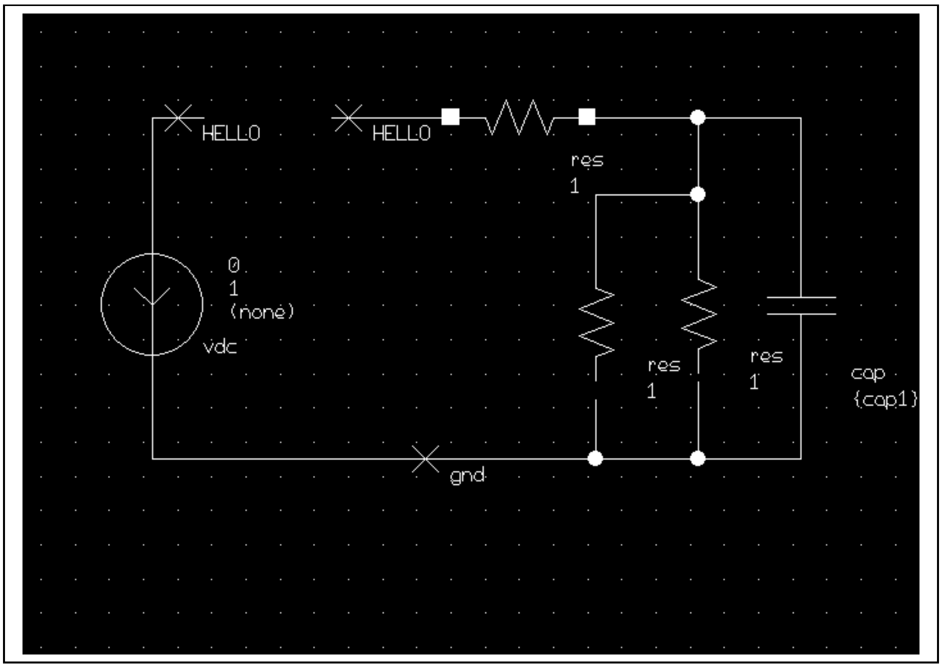


Figure 17.3: Usage of 'named pins'

17.3 Incorporating external netlists : 'other '-elements

In another chapter, I described hierarchies of schematics. There is another method to introduce hierarchy, it is the incorporation of external netlists.

This method is useful if you want to use netlists created not by yourself but by other people or companies (for example, opamp macromodels or power transistor device models).

Let us assume that you are doing a design which uses discrete operational amplifiers commercially available . If you want to simulate your design, you want to use ready-to-use libraries which contain netlists and describe the behaviour of your opamp.

To insert these netlists into your SPICE3 netlist, you have to place an 'other'-element. This element has to be created by the SPICECAD user and looks like a symbol. However, the 'other' element has at least one property, the 'model'-property. This property contains the name of the netlist file which has to be inserted into the simulation netlist during netlisting. Such an elements appears like a subcircuit in your netlist.

```
X12 n1 n2 n3 f_adder
```

The subcircuit definition is in file 'f_adder.m' which looks as follows:

```
.SUBCKT f_adder 1 2 3
B0 3 0 v=v(1)+v(2)
.ENDS f_adder
```

This behaviour describes the hierarchical netlisting mode.

There is another mode, the flat netlisting mode.

In this mode, the subcircuit will be expanded and incorporated into the simulation netlist once for every appearance of an 'other'-element. **In this mode, variables and expressions may be used within the subcircuit definition which looks as follows.**

This is an example for file f_adder.m :

```
V1 1 0 param1
I2 2 0 param2
R1 3 0 param2+param4
L1 3 0 param4*param1
C1 3 0 param2/param4
M1 1 2 3 4 modelfile w=20u l=name5
M2 1 2 3 4 modelfile1 l=20u w=name5
Q1 2 3 4 5 modelfile_bjt area
```

In the flat mode, only these elements are allowed which also appear in the example with exactly the same syntax. This means, no AC , SIN or PWL sources ,no resistor temperature coefficients, no NS, ND or AREA parameters.

For the node numbers, the following restrictions have to be obeyed:

a) 0 ist the ground-node, the global ground. b) Nodes 1..n are external nodes . They will be replaced by external node numbers during netlisting. There must not be gaps between 1..n, each nodenumber between 1 and n must appear in the netlist.

This convention is like

```
.SUBCKT f_adder 1 2 3 .  .  .  n
```

in a SPICE3 subcircuit definition.

n must be smaller or equal 100.

c) Nodes n+1 .. m are internal nodes. There may be gaps between n+1 and m. The number of different internal nodes is 100.

d) Device names, property names and model file names must not be longer than 40 characters.

17.4 Creating 'other'- elements

The picture of an 'other'-element can be created like a symbol.

So, leave SPICECAD and start it again typing
Spice 1 -> cad -o firstother

Like in a symbol, contacts have to be placed to connect the subcircuit to the schematics world. Don't forget to give names to the contacts (edit their properties)!

The element is saved using 'sv oth' in the main menu.

After saving, you have to add additional information to the just created file 'firstother.oth' to tell SPICECAD the contact sequence and the netlisting mode (hierarchical/flat).

In the following lines, there are two examples of such an information trailer.

hierarchical netlisting mode: .

```
.
.
stop
;end of graphics primitives section subckt
X
3 IN1 IN2 OUT
notused
1 model
```

Line entry	Meaning
. . .	
stop	End of picture data
subckt	keyword . It means that the incorporated netlist contains analog elements
X	Prefix (first Letter) , 'X' for subcircuits , 'Y' for digital parts (just under implementation)
3 IN1 IN2 OUT	Number and name of contacts. The sequence of names will be the netlist sequence later. Example : X123 IN1_node IN2_node OUT_node f_adder
hierarchical	keyword for hierarchical netlisting
1 model	number of properties and their names. In the hierarchical netlisting mode, only 'model' has a meaning (it's a keyword). In flat netlisting mode, additional model names may be used.

flat netlisting mode:

```
.
.
.
stop
subckt
X
3 IN1 IN2 OUT
flat
4 model {par1} {par2} {par3}
```

Chapter 18

Transistors and 'globals'

MOS-Bulk : 'gnd' ... 'vss'

No time to write something here.

Chapter 19

Multi-frame editing

Multi-frame editing is editing two different schematics at the same time in two different windows. You can start multi-frame editing by selecting ’open 2. screen’ on the main menu.

Multi-frame editing is useful to move or copy groups of devices (created by ’win add to group’) from one schematic to an other one. For example, you have designed a bias stage and want to use it in another design.

If you have created a symbol from the bias stage, you can import a symbol rather than a schematic. This way, you can re-use designs. If you don’t want to create a symbol, you need multi-frame editing to copy the bias stage from one schematic to the other. But be careful: properties will not be copied, too. If you have used them and you don’t know the values any longer, you have to change to the top level of the bias stage design, edit the property box and save the parameter list in file ’varlist’ using the appropriate button. After having copied the bias stage into the second design, update the parameter box of this second design and import the missing parameters from the file you have just created before.

After having copied everything, you should close the second frame using ’close 2. screen’ in the main menu. **Warning: Multi- frame SIMULATION is not allowed. So, close the second design frame after copying or moving devices.**

The multi-frame editing is just a very new function and does not behave as it should. Nevertheless, you can copy and move , and this is all this function is intended to do.

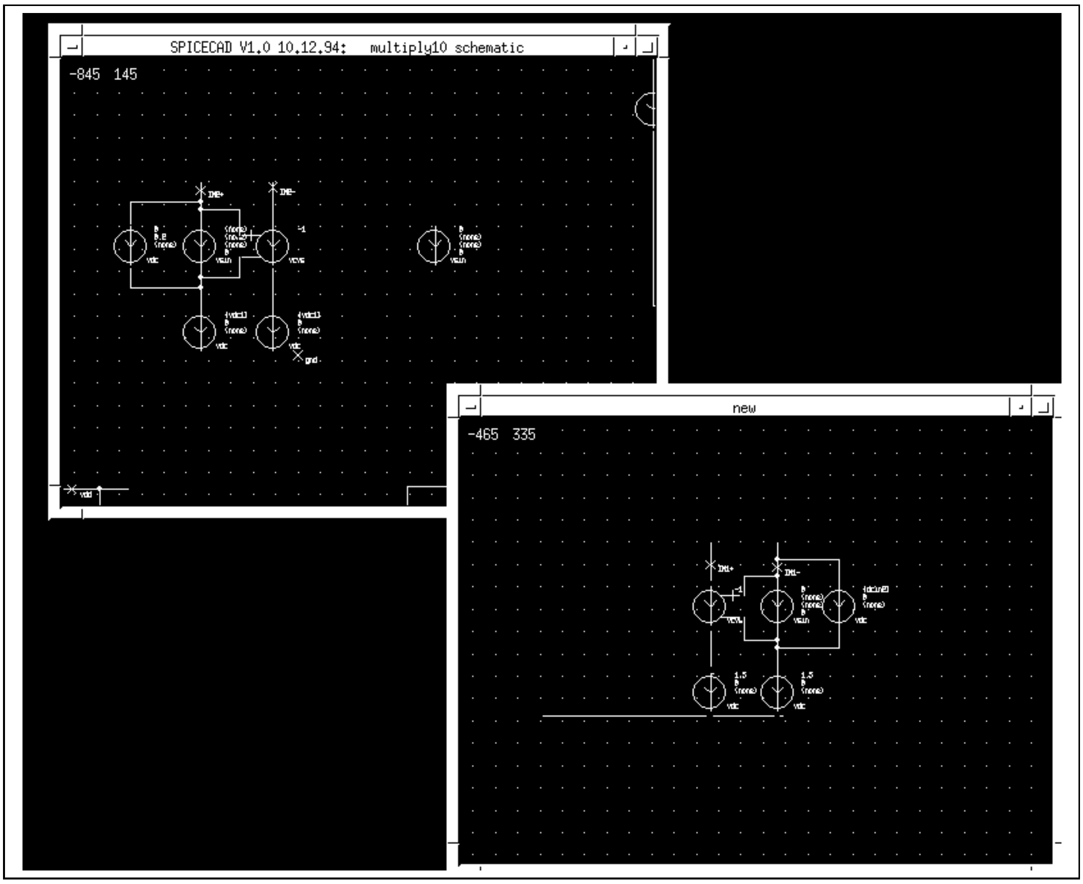


Figure 19.1: Multi-frame editing

Some steps have to be followed to copy or move.

1. Create a group using ’win add to group’.
2. Select copy or move or use the hotkeys in the source screen.
3. Move the mouse into the target screen. (As you will see , nothing will happen.)
4. Select copy or move or use the hotkeys in the target screen. Now, the group will appear in the target screen.

If the group appeared in the target screen, you should save the tagret screen.

Chapter 20

Hotkeys and key redefinitions

Hotkeys are very useful for fast access of often-used functions like redraw, save etc. Using hotkyes requires that SPICECAD recognizes these keys well. Many X11 users customize the behaviour of their keyboards by using the program 'xmodmap'. The usage of this program may change some keycodes. This might cause the decoding of the hitted kotkey to fail. To circumvent this problem, you can define the new keycodes in file 'keys.spi'. The file has to look as follows, where the numbers given here are an example:

```
CTRL_KEY_R 65507
CTRL_KEY_L 65508
ALT_KEY_R 44567
ALT_KEY_L 44568
SHIFT_KEY_L 34565
SHIFT_KEY_R 34566
ESC_KEY 45452
CURUP_KEY 34456
CURDOWN_KEY 34457
CURLEFT_KEY 34458
CURRIGHT_KEY 34459
DEL_KEY 34460
INSERT_KEY 34461
```

The numbers behind the key names have to changed individully according to your keyboard map table. For example, if CTRL_KEY_R is not recognized (the right Control key), press this key and find the number of this key in the error message appearing in the x-terminal window:

```
key unknown :  65513 offset 2000
```

The offset is debugging information for SPICECAD DEVELOPMENT. So, take the 65513 and write it behind CTRL_KEY_R in file 'keys.spi'.

```
CTRL_KEY_R 65513
CTRL_KEY_L 65507
ALT_KEY_R 44567
ALT_KEY_L 44568
SHIFT_KEY_L 34565
SHIFT_KEY_R 34566
ESC_KEY 45452
```

If you do not get a message when a key remains undetected, start the schematic entry using
'cad -key -c cellname'
.

Then, you will get the keycode in the X11 terminal window where you started SPICECAD each time you hit a key.

Only those keys which are not recognized have to be re-defined in file 'keys.spi'. Simple keys like function keys, number keys (0 1 2 3 4 .. 9) and letter keys (a b c .. z A B C ... Z) can be re-defined, too:

```
CTRL_KEY_R 65513
CTRL_KEY_L 65507
ALT_KEY_R 44567
ALT_KEY_L 44568
XK_a 65
XK_e 68
XK_z 91
XK_A 92
XK_F 102
XK_Y 112
XK_3 123
XK_F1 165
XK_F11 174
```

Here is a list of all used hotkeys:

a: Pop one hierarchy level (return from symbol or schematic editing)
b:
c: copy without filter
d: delete without filter
e: Push one hierarchy level (go to symbol or schematic editing)
f: full (fit window to schematic size)
g: Edit properties
h: plot next net segment in 'horizontal -then - vertical' mode
i
j
k
l
m: move without filter
n: Begin net editing
o: zoomout (default zoom coordinate system. Primarily useful for debugging purposes.)
p: pan
q: copy properties between devices of the same kind. nmos->nmos4 is possible, too.
r: redraw
s: Snap to next net or contact, continue net editing.
t: Pop all hierarchy levels (return from symbol or schematic editing, go to top level)
u
v: plot next net segment in 'vertical -then - horizontal' mode
w
x
y
z
A
B
C: copy with filter (default filter : no nets)
D: delete with filter(default filter : no nets)
E
F
G

H
I
J
K
L
M: move with filter(default filter : no nets)
N
O
P
Q
R
S : Snap to next net or contact, finish net editing T
U
V
W
X
Y
Z
1 : plot setup 1 (plot the plot-setup which you have saved using CTRL 1)
2 : plot setup 2
3 : plot setup 3
4 : plot setup 4
5 : plot setup 5
6
7
8
9
0
CTRL 1 : save plot setup 1
CTRL 2 : save plot setup 2
CTRL 3 : save plot setup 3
CTRL 4 : save plot setup 4
CTRL 5 : save plot setup 5
CTRL 6
CTRL 7
CTRL 8
CTRL 9
CTRL 0

ALT a
ALT b
ALT c
ALT d
ALT e
ALT f
ALT g
ALT h
ALT i i: select terminal current to plot
ALT j
ALT k
ALT l
ALT m
ALT n: netlist
ALT o
ALT p
ALT q
ALT r
ALT s
ALT t
ALT u d : undo delete
ALT v v: select voltage to plot
ALT w
ALT x
ALT y
ALT z
ALT 1: pop up main manu
ALT 2: pop up simulation menu
ALT 3: pop up zoom menu
ALT 4: pop up symbol menu
ALT 5: pop up library menu
ALT 6: pop up plot menu
ALT 7: pop up first options box
ALT 8: pop up second options box
ALT 9: pop up parametric simulation menu
ALT 0: pop up undo menu
CTRL a : load ac simulation data (not necessary in SPICE3)
CTRL b

CTRL c
CTRL d: load dc transfer simulation data (not necessary in SPICE3)
CTRL e: exit SPICECAD. SPICECAD will ask you for saving your schematic if you did some changes after the last saving.
CTRL f
CTRL g
CTRL h : Enter help mode. After doing this, you can select menu entries or objects on the schematics screen. If available, a help text will appear.
CTRL i
CTRL j
CTRL k: run monte carlo simulation
CTRL l
CTRL m
CTRL n
CTRL o: load dc op point simulation data (not necessary in SPICE3)
CTRL p
CTRL r: run ordinary simulation (no paramsweep, no monte carlo, no optimization)
CTRL s: save all what you are currently editing CTRL t: load transient simulation data (not necessary in SPICE3)
CTRL u
CTRL v
CTRL w: exit SPICECAD. All changes you did after the last saving will be lost, and SPICECAD will not ask you to save before leaving.
CTRL x
CTRL y
CTRL z
F1: pop up main manu
F2: pop up simulation menu
F3: pop up zoom menu
F4: pop up symbol menu
F5: pop up library menu
F6: pop up plot menu
F7: pop up first options box
F8: pop up second options box
F9: pop up parametric simulation menu
F10: pop up undo menu
Cursor right: Scroll into right direction in screen and plot window.

Cursor left: Scroll into left direction in screen and plot window.
Cursor up: Scroll to top in screen and plot window.
Cursor down: Scroll to bottom in screen and plot window.
. : Rotate
- : Mirror
/ : Mirror

Chapter 21

Extensions of SPICE3

21.1 arbitrary sources : user defined functions

The ASRC- arbitrary source has some new functions it can parse. These functions are time(): current time in a transient-simulation. rand(): random function to simulate noise in transient simulations user0(): user defined function0 user10(): user defined function10

User defined functions are implemented in file ptfuncs.c which can be compiled and liked to SPICECAD. See makefile.

21.2 arbitrary sources : link to net alias names

Node references can be given in an arbitrary source. There references are resolved during netlisting. Note: there will be a warning message that the symbol list has been resetted during netlisting. This is caused by resolving the node reference.

Node references to net alias names look like this: `B0 1 0 v=` $5.0 - sin(34.6 * v\{refnode\})$ `(example)` 'refnode' is a net alias name.

21.3 scaling of passive devices: constant or timec or frequency dependent

A useful feature of HSPICE is the possibility of having frequency -, time- or temperature- dependent component values. As an example, the modelling of the skin effect during small-signal simulations can be performed only using a frequency-dependent scaling constant. Thus, the SPICE3 syntax has been extended by a SCALE-statement similar to the HSPICE syntax.

The predefined variable names are 'hertz', 'time' and 'temper'.

A passive element definition thus looks like this:

C1 1 2 100n SCALE=1.0+ATANH[hertz*3.13159] R1 2 3 100k SCALE=1.0+time*11.3e6 L1 3 4 1u SCALE=1.0+temper/273.0

SPICECAD provides a 'scale' property at resistors, capacitors and inductors. It is not necessary to use " or " when using variables or expressions inside the scaling factor definition.

The following functions are allowed inside the scale expressions:

ASINH[] ACOSH[] ATANH[] ASIN[] ACOS[] ATAN[] SINH[] COSH[] TANH[] SIN[] COS[] TAN[] SQRT[] EXP[] LN[] LOG10[] ABS[] GAMMA[] ERFC[] ERF[]

It is mandatory to use '[' and ']' as brackets, and to write the function name in upper-case letters.

Inside the scale expression, variables used and defined in the parameter box may be used, too.

It is possible to suppress the netlisting of the SCALE statement. Inside the options1-dialog (hotkey F8), a button 'netlist scale' exists, which can be set to 'yes' or 'no'.

21.4 MOS models containing geometry information

MOS transistor models may contain geometry information, which means the definition of gate lengths and widths. This is important if discrete semiconductor libraries are used instead of integrated transistors. This way, SPICECAD is as useful for discrete circuit designs as for integrated circuit designs.

As an example, the following MOS model definition is allowed now:

.model n NMOS l=1u w=100u

21.5 New options

A few new options has been added to the list of SPICE3 options.

These are :

- cptime : Limits the CPU time to the given number of seconds. This is useful especially when it shall be avoided to kill the whole SPICECAD process when the simulation time exceeds useful limits
- gminsteps : the number of steps during gmin stepping is now adjustable. Its default has been increased from 10 to 100. It can be overridden using :
  .option gminsteps=200 .
- srcsteps : the number of steps during source stepping is now adjustable. Its default has been increased from 10 to 100. It can be overridden using :
  .option srcsteps=200 .
- delmin : The normal spice3 kernel sets delmin to a constat value which may be too high during runtime. Although convergence could be achieved, spice3 halts the simulation saying 'Timestepp too small'. Now, delmin is set by default to 1e-15. If this still causes 'Timestepp too small' errors, it can be overridden using :

.option delmin=1e-16 .

21.6 importing x-y files during simulation

In the past, it was possible to import x-y data by using picewise linear voltage or current sources. Unfortunately, the length of such a data set was limited to 1000 .. 5000 points depending on the speed of the employed computer. Now , the SPICE3 kernel has been extended. It is possible to import x-y data by telling SPICE3 where to read the data . The syntax is as follows:

V0 1 0 DC 1 AC 1 filepwl filename= ¡filename¿

Such a netlist entry is generated after filling in the 'filename' entry of a piecewise linear voltage source. When using HSPICE as simulator engine, the contents of the file containing the x-y data is written into the netlist instead.

Chapter 22

Interfaces to other simulators

SPICECAD can produce netlists for other simulators, e.g. HSPICE and TITAN.

22.1 HSPICE

The integration of HSPICE in SPICECAD works as follows.

The simulator can be chosen in the 'options*'-box in the main menu. The 'HSPICE' button must be selected to tell SPICECAD that HSPICE–compatible netlists shall be written.

HSPICE netlists are written automatically before each simulation. To force netlisting if a model file has changed since the last netlisting, the netlisting can be started explicitly by hitting the 'netlist' button. The netlist file ends with '.sp'. If the edited schematic topcell is called 'topcell', the netlist file will be called 'topcell.sp'.

To start the simulator, a script 'run_hspice' will be called. A sample can be found in the installation path inside the 'bin'-directory. This script needs to be customized .
A sample script looks like this:

```
#!/bin/csh -f
# users' own setup in cad_hspice
```

```
/cad/bin/cad_hspice
mv -f $1.sim $1.sim.bak
hspice $1.sp > $1.sim
```

The script calls another script (cad_hspice) which sets environment variables. These settings can be done in the 'run_hspice'-script , instead. Afterwards, the output file is saved (this line can be deleted), and as a last step, HSPICE is called.

HSPICE can be run in two modes: in one of these modes, HSPICE will run in background. This means, immediately after executing the line
'hspice $1.sp > $1.sim' ,
the script 'run_hspice' will be finished. This means that a message is sent to SPICECAD telling that HSPICE has finished. Of cource, this might be wrong, because HSPICE might still be running in background. So, if customizing the 'run_hspice'-script , make sure that you do not put HSPICE in the background run mode.

The script 'run_hspice' will be executed by a batch job.

Here, some more customization is possible. You can choose (in the option-dialog) on which machine HSPICE shall be executed, and with which nice priority HSPICE shall run. .

After receiving the message that HSPICE has finished, simulation results will be read from 'topcell.ac0', 'topcell.sw0', 'topcell.tr0' and 'topcell.sim'. Error and warning messages are found in 'topcell.sim'.

To start the simulation, 'runsim' has to selected in the simulation menu (or use hotkey 'CTRL r'). If selected, a file called 'run' will be opened in the current directory. The file will be executed on a remote host (called simulatorhost) on path 'simulatorpath' using the command
rsh simulatorhost 'cd simulatorpath; run_hspice topcell'

The variables 'simulatorhost' and 'simulatorpath' have to be specified in the 'options1*'-menu accessible from the main menu. 'simulatorhost' is the host where HSPICE will be run, and 'simulatorpath' has to be set to your current path. The latter is no longer necessary, during the SPICECAD start phase, this variable is set to the current working directory $PWD.

The 'run_hspice'-script is called from another script, which contains all the customization info which you have specified before.

Here is the script:

```
rm -f hspicemsg.simulatorhost.*
echo "begin of job"
date
nice -0 run_hspice $topcell
echo "end of job"
date
grep  error $topcell.sim  > hspicemsg
sendmessage $simulatorhost 5218
rsh $your_own_host kill -1 5218
```

The SPICECAD process (PID 5218) will receive a signal (SIGNAL 1, HANGUP) . This signal tells SPICECAD that HSPICE is ready.

SPICECAD will now open the file 'hspicemsg' and look for the contents. If the HSPICE simulation was successful, which means 'hspicemsg' contains only one line, 'hspice ready', a window pops up in SPICECAD which contains this line. Afterwards, the simulation results will be read in by SPICECAD automatically. If the simulation was unsuccessful, which means that 'hspicemsg' contains several error messages, the error messages will be displayed in the error-and-warning-window , but no simulation results will be read in.

Besides .ac, .tran and .dc simulation results, dc operating point results (in file topcell.sim) are being read in, too. The whole file 'topcell.sim' is read in and stored in internal data structures. These data structures can be accessed and processed via different functions. dc op point voltages and currents can be displayed using 'show voltage' and 'show current'. Device information about bipolar and MOS transistors can be displayed using 'showdevice'. Devices in triode region or in saturation can be displayed using 'show bad devices'.

Parametric sweep, monte carlo simulation and optimization are working.

DC operation point results can be displayed (showdevice, showvolt) in the schematic window. (showcurrent is not yet ready).

To be able to display results from deeper hierarchy levels which are included in the netlist as subcircuits, the 'create gsiwindow' function was implemented. This function displays all node names in a schematic window. To determine where these names are to be placed, create a text object '@gsi' and place it where you want to see these names.

To be able to select a net and display the simulation result, you have to execute 'select alias' before. This means that instead of displaying the result according to the netnumber of the selected net, the alias name of this net is taken instead and will be displayed. To switch back to normal mode , execute 'select netnumber'.

By the way, currents can be displayed this way, too. All displayed nets have aliases, some are currents, most are voltages.

22.2 TITAN

The integration of TITAN in SPICECAD works as follows.

The simulator can be chosen in the 'options*'-box in the main menu. TITAN3 (Versiond 3.2) and TITAN4 (TITAN4.0) buttons can be selected.

TITAN netlists are written automatically before each simulation. To force netlisting if a model file has changed since the last netlisting, the netlisting can be started explicitly by hitting the 'netlist' button. The netlist file ends with '.sp'. If the edited schematic topcell is called 'topcell', the netlist file will be called 'topcell.sp'.

Simulation results will be written in 'topcell.ac', 'topcell.dc', 'topcell.tr' and 'topcell.sim'. Error and warning messages are found in 'topcell.msg'.

By default, all simulation results are saved using the '.PRINT'-card :
`.PRINT V(*)` is default. Currents are not saved, this has to be done explicitly in the simulation menu. To start the simulation, 'runsim' has to be selected in the simulation menu (or use hotkey 'CTRL r'). If selected, a file called 'run' will be created in the current directory. The file will be executed on a remote host (called 'simulatorhost') on path 'simulatorpath' using the command

```
rsh simulatorhost 'cd simulatorpath; run_titan topcell'
```

The variables 'simulatorhost' and 'simulatorpath' have to be specified in the 'options1*'-menu accessible from the main menu. 'simulatorhost' is the host where TITAN will be run, and 'simulatorpath' has to be set to your current path.

File content of 'run_titan':

```
rm -f titanmsg titmess.sunny.*
run_titan topcell
grep -n -v ERRORS topcell.msg | grep  ERROR  > titanmsg
grep -n ERROR  topcell.sim >> titanmsg
sendmessage sunny 7696
rsh sunny kill --1 7696
```

First, some (old) files will be deleted wich contain information about the last TITAN run. Then, a shell script 'run_titan' will be called which has to be installed in your current search path. 'run_titan' looks as follows:

```
#!/bin/csh -f
/marbic/titan/bin/titan.ip_sun4  <<! > /dev/null
$1.sp
$1.msg
zcd.$1
!
/marbic/titan/bin/titan.ak_sun4  <<!
zcd.$1
$1.sim
!
rm -f zcd.$1
```

This shell script starts the preprocessor and the simulator using the handled variable '$1'. The the above example, '$1' will be set to 'topcell'.

After executing 'run_titan', the 'run' - shellscript will be continued as follows:

```
grep -n -v ERRORS topcell.msg | grep  ERROR  > titanmsg
```

```
grep -n ERROR  topcell.sim >> titanmsg
sendmessage sunny 7696
rsh sunny kill -1 7696
```

The error messages will be extracted from the files topcell.msg and 'topcell.sim' and stored in file titanmsg. Afterwards, a shell-scipt called 'sendmessage' will be executed. 'sendmessage' contains the following lines:

```
#!/bin/csh -f
echo "titan ready" > titmess.$1.$2
```

A file called 'titmess.sunny.7696' (this example) will be created which contains the line 'titan ready'. 7696 is the process id of the SPICECAD process.

The last line of shell-script 'run' is the line

```
rsh sunny kill -1 7696
```

A remote shell is started on host sunny, the same host SPICECAD is running on. The SPICECAD process (PID 7696) will receive a signal (SIGNAL 1, HANGUP) . This signal tells SPICECAD that TITAN is ready.

SPICECAD will now open the file 'titanmsg' and look for the contents. If the TITAN simulation was successful, which means the 'titanmsg' contains only one line, 'titan ready', a window pops up in SPICECAD which contains this line. Afterwards, the simulation results will be read in by SPICECAD automatically. If the simulation was unsuccessful, which means that 'titanmsg' contains several error messages, the error messages will be displayed in the error-and-warning-window , but no simulation results will be read in.

If you have specified a '.op'–simulation (dc operating point), the file 'topcell.sim' will be scanned, and all dc operating point voltages will be written in a file called 'nodes.m' . This file looks as follows:

```
.NODESET V(1) = 0.000000e+00
```

```
.NODESET V(2) = 0.000000e+00
.NODESET V(3) = 1.000000e+00
.NODESET V(4) = 4.999898e+00
.NODESET V(5) = 5.000000e+00
```

If you repeat the simulation with slightly different parameters, it would speed up the simulation if this file could be included in the netlist. To do this, open the 'options1*'-box in the main menu and set the entry 'default includefile 1' to 'nodes'.

Then, file 'nodes.m' will be included in the TITAN netlist during subsequent simulations.

After reading in the simulation results, you can select nodes or terminals to display them in the plot window.

Besides .ac, .tran and .dc simulation results dc operating point results (in file topcell.sim) are being read in, too. The whole file 'topcell.sim' is read in and stored in internal data structures. These data structures can be accessed and processed via different functions. dc op point voltages and currents can be displayed using 'show voltage' and 'show current'. Device information about bipolar and MOS transistors can be displayed using 'showdevice'. Devices in triode region or in saturation can be displayed using 'show bad devices'.

22.2.1 Parametric sweep, Monte Carlo simulation and circuit optimization

Additionally to the built-in simulation types, SPICECAD emulates so-called parametric simulations. For example, you want to perform a parameter sweep over the variable 'rload'. It has to be varied between 100Ohm and 500Ohm in 6 steps. First, place a 'paramsweep' -box in the schematic and fill out the entrys. Then, enter the 'parametric*'- menu from the simulation menu or use hotkey 'F9'. Then, execute 'run paramsweep' . **The swep variable name in the parameter sweep box has to be set to 'rload' and not to 'rload'.**

Now, six netlists (for each sweep step one netlist) will be written and called one after another in shell-script 'run'. After the last run, SPICECAD receives a SIGNAL and reads in all simulation files except the dc op point information files. (No sweep over dc operation points).

After selecting a **single** node or terminal, all results (dc, ac or transient) of the different simulation runs can be displayed in the same window.

Circuit optimization , dito.

22.2.2 Device currents

Currents through contacts of passive elements, independent and dependent sources, bipolar, MOS and JFET transistors can be saved during dcsweep and transient simulations . During ac simulations, these currents cannot be saved except currents through voltage sources. To keep the currents through all types of devices during ac simulations, a mechanism was created which adds a voltage source in series to each device contact which has to be saved. To keep the number of additional nodes small, transient and ac simulations should be executed separately. (Disabling the ac simulation box is recommended when a transient simulation is performed. Then, the transient simulation box should be disabled and the ac simulation box enabled.)

The currents through symbols and 'other'-elements can also be saved using the additional voltage source netlisting mechanism.

22.2.3 Required Hardware Platforms

SPICECAD is available under SUN SPARC running SUN OS 4.1.1 .. 4.1.3 and X11 R5 and under IBM PC and compatibles running LINUX and X11 R5.

Chapter 23

Batch execution

SPICECAD has a simple command interpreter which allows for execution of command files. The command syntax is simple (no variables, no branches, no loops, no if-then-else). The file which is executed can be defined in the 'options1' - dialog, entry 'batch file'. Lines beginning with a '#' are skipped, they can be used to insert comments.

In tables 23.1 ... 23.5, the available commands are described in short. Additionally, three batch file examples will show how to use these commands.

First batch file example :

```
set simulator spice3
#set simulator hspice
#set xrange 0 10
set xlabel time [s]
set ylabel Vout [V]
set title example of a batch file
set param cap1 1.0
set param res1 2.0
set calculatorline 0 V(1) tran
open calculator
runsim
# for correct line count!
plot calc
#generate hardcopy of plot result window, into file tmp.ps
hardcopy ps
#
#
#
system mv tmp.ps tmp1.ps
set param cap1 .2
set param res1 20.0
runsim
plot calc
hardcopy ps

#stop

#open calculator
set calculatorline 0 V(2) tran
#import measurement
set calculatorline 1 import tmp.txt
set calculatorline 2 -
plot calc
#
#
#
```

```
hardcopy ps
system mv tmp.ps tmp2.ps

#stop

set calculatorline 2 -
plot calc
hardcopy ps
system mv tmp.ps tmp3.ps
#
set param cap1 .4
set param res1 10.0
runsim
#
plot calc
hardcopy ps
system mv tmp.ps tmp4.ps
#

close calculator
close plotwindow


stop
```

Second batch file example: calls an external SPICE3 simulation core and reads its

results afterwards

```
set simulator spice3
netlist
system spice3 -b rctest1.sp -raw rawspice.raw
load spice3 rawspice.raw
stop
```

Third batch file example : sweep over different technology files, temperatures and special command files

```
set simulator spice3
#set simulator hspice
#set xrange 0 10
set xlabel input voltage [V]
set ylabel output voltage [V]
set title CMOS inverter transfer function using different
technology sets
set modelpath 1 /usr1/martinm/models/hspice/035/ff
runsim
select net outnet
open plotwindow
plot dcsweep
hardcopy ps
system mv tmp.ps ff.ps
#
set modelpath 0 /usr1/martinm/models/hspice/035/ss
runsim
plot dcsweep
hardcopy ps
system mv tmp.ps ss.ps
#
set modelpath 0 /usr1/martinm/models/hspice/035/fs
runsim
plot dcsweep
hardcopy ps
system mv tmp.ps fs.ps
#
set modelpath 0 /usr1/martinm/models/hspice/035/sf
runsim
plot dcsweep
hardcopy ps
system mv tmp.ps sf.ps

set modelpath 0 /usr1/martinm/models/hspice/035/ty
set temp 100
```

```
runsim
plot dcsweep
hardcopy ps
system mv tmp.ps ty_100.ps
#
set temp -40
runsim
plot dcsweep
hardcopy ps
system mv tmp.ps ty_-40.ps
#
set temp 25
set includefile 0 specialcommands1
runsim
plot dcsweep
hardcopy ps
system mv tmp.ps ty_special1.ps
#
set includefile 0 specialcommands2
runsim
plot dcsweep
hardcopy ps
system mv tmp.ps ty_special2.ps
stop
```

stop :	stops execution
no_execution:	test the script, the commands are not executed
redo_execution:	exit test mode, execute commands starting here
run batchfile :	call another batchfile.
run monte :	run Monte Carlo simulation
run paramsweep :	run parameter sweep simulation
run optimgrad :	run optimization, use gradient search
run optimmonte :	run optimization, use Monte Carlo approach (rapid thermal annealing)
plot voltages :	plot voltages only (no mix of voltages and currents)
plot currents :	plot terminal currents only (no mix of voltages and currents)
plot tran:	Plot selected voltage or terminal currents from transient simulation
plot dcsweep:	Plot selected voltage or terminal currents from dc sweep simulation
plot acm:	Plot selected voltage or terminal currents from ac sweep simulation (magnitudes)
plot acp:	Plot selected voltage or terminal currents from ac sweep simulation(phase)
plot inoise:	plot input referred noise from noise simulation
plot onoise:	plot output referred noise from noise simulation
plot calc normal :	plot the calulator contents in normal (non-parametric) mode
plot calc parametric :	plot the calulator contents in parametric mode
clear xrange:	sets plot xrange to default (=autoscale)
clear yrange:	sets plot yrange to default (=autoscale)
clear xlabel:	sets plot xlabel string to default = "(none)"
clear ylabel:	sets plot ylabel string to default = "(none)"
clear title:	sets plot title string to default = "(none)"
clear selectednet :	clear list of selected net voltages to be plotted
clear selectedterm :	clear list of selected terminal currents to be plotted

Table 23.1: Begin of batch command list

select net 'alias':	Select the net with the attached 'alias' shall be plotted after the next 'plot' command
select terminal 'alias' 'name':	Select the terminal 'name' of the device with the attached 'alias' to be plotted after the next 'plot' command. The terminal names are : 'B' 'D' 'S' 'G' for MOS transistors 'S' 'D' 'G' for JFET and MESFET transistors 'B' 'C' 'E' 'S' for bipolar transistors 'coil1','coil2' for coupled inductors No terminal names are required for two - terminal devices like sources and passive elements
unselect net 'alias':	Deselect the net with the attached 'alias' which shall no longer be plotted after the next 'plot' command
unselect terminal 'alias' 'name':	Select the terminal 'name' of the device with the attached 'alias' which shall no longer be plotted to be plotted after the next 'plot' command. The terminal names are : 'B' 'D' 'S' 'G' for MOS transistors 'S' 'D' 'G' for JFET and MESFET transistors 'B' 'C' 'E' 'S' for bipolar transistors 'coil1','coil2' for coupled inductors No terminal names are required for two - terminal devices like sources and passive elements
set calculatorline 'n' 'string' :	'n' is the number of the calculator line you want to set, 'string' is the string which shall appear in this line. 'n' starts at 0.
set modelpath 'n' 'path' :	Set model path no. 'n'. 'n' starts at 0.
set host 'n' 'host' :	Set simulator no. 'n'. 'n' starts at 0.
set scriptfile 'scriptfile' :	Set 'scriptfile' to the name of a script file which can filter the SPICECAD generated netlist
set nice 'prio' :	Sets the nice priority of the simulator task.
set librarypath 'n' 'path':	Set library path no. 'n'. 'n' starts at 0.
set includefile 'n' 'file' :	Set include file name no 'n' to 'file'. 'n' starts at 0.

Table 23.2: Continuation of batch command list

set param ’name’ ’value’ :	Sets parameter ’name’ inside parameter box to value ’value’
set axis ’look’:	’look’ may be ’nolog’,’xlog’,’ylog’,’loglog’.
set grid :	Switch grid in plotwindow on
set nogrid :	Switch grid in plotwindow off
set key :	Displays plot description in plotwindow
set nokey:	Do not display plot description in plotwindow

Table 23.3: Continuation of batch command list

set style 'look':	'look' may be 'lines','points' or 'linespoints'.
set xrange 'xmin' 'xmax' :	Define plot x-range.
set yrange 'ymin' 'ymax' :	Define plot y-range.
set xlabel :	Defines label of x-axis in plotwindow
set ylabel :	Defines label of y-axis in plotwindow
set title :	Defines title in plotwindow
set simulator 'sim' :	'sim' can be 'spice3','hspice' or 'titan'
set tran 'prop' 'val' :	set properties inside the TRAN simulation box 'prop' may be one of 'tstep' 'tstop' 'tstart' 'tmax' 'UIC' 'MONTE'
set dcop 'prop' 'val' :	set properties inside the DCOP simulation box 'prop' may be one of 'when' 'MONTE' (only when using HSPICE!)
set dcsweep 'prop' 'val' :	set properties inside the DCSWEEP simulation box 'prop' may be one of 'source1' 'start1' 'stop1' 'step1' 'source2' 'start2' 'stop2' 'MONTE'
set acsweep 'prop' 'val' :	set properties inside the ACSWEEP simulation box 'prop' may be one of 'inctype' 'num' 'fstart' 'fstop' 'MONTE'
set temp 'prop' 'val' :	set properties inside the TEMP simulation box 'prop' may be one of 'temp'
set paramsweep 'prop' 'val' :	set properties inside the PARAMETER SWEEP simulation box 'prop' may be one of 'varname1' 'sweeptype1' 'start1' 'stop1' 'steps1' 'varname2' 'start2' 'stop2' 'steps2'
set option 'prop' 'val' :	set properties inside the OPTIONS simulation box 'prop' may be one of abstol reltol method gmin tnom probe post gmindc converge gshunt interpolation delmax vntol trtol limpts scale
set hbfreq 'prop' 'val' :	set properties inside the harmonic balance analysis box 'prop' may be one of startfile freqfile hbstab
set hbtran 'prop' 'val' :	set properties inside the harmonic balance analysis box 'prop' may be one of source start stop inc hbstab
set startup 'prop' 'val' :	set properties inside the startup analysis box 'prop' may be one of ianmod epscon muldaf ngsmax ntimpo ninper nenvwr ninwri

Table 23.4: Continuation of batch command list

load spice 'file' :	load spice3 results file
load schematic 'schemname' :	Load schematic. Use very carefully!
read params :	Read parameter file 'varlist'
read calcfile n :	Load the contents of file 'calcn.con' , where n is equal or between 1 and 3. This commands sets the lines of the calculator window.
read tran :	Read simulation results from transient simulation
read dc :	Read simulation results from dcsweep simulation
read ac :	Read simulation results from small-signal simulation
spice3command 'command':	with this command, you can send commands directly to the spice3 command interpreter. Using this command, you can control which simulations you want to save in which file etc.
save spice3 :	save all SPICE3 simulation results into file 'rawspice.raw'.
open calculator :	opens the calculator window
open plotwin :	opens the plot results window
close calculator :	closes the calculator window
close plotwin :	closes the plot results window
hardcopy ps :	plot plotwindow contents into postscript file tmp.ps
hardcopy hpgl :	plot plotwindow contents into hpgl file tmp.hgl
hardcopy table :	save plotwindow contents in an ascii file 'tmp.txt' in tabular format
save as goal :	save plotwindow contents in goal file format in an ascii file 'tmp.txt' for later use in an optimization.
system 'command' :	executing 'command' using 'system "command"'
runsim :	run simulator
netlist :	run netlisting explicitly. Althogh SPICECAD netlists automatically when library or model paths have been changed, it is better to run 'netlist' before each simulation explicitly.
reload :	reloads schematic after changing library path

Table 23.5: Continuation of batch command list

Chapter 24

What's new and very hot features

24.1 UNDO

A primitive UNDO function is under construction: UNDELETE is possible now, 10 elements which have been deleted are stored in a queue where they can be retrieved again by selecting 'undelete' in the UNDO menu.

Things like unmove (what happens if two nets are merged after a move?) are more complicated, and I have to do a big thinking job to implement these features so that they a running stable (do I hear hysteric laughter in the background?) and future-compatible.

Pulldown menus have been requested by various friends of SPICECAD. They are available now. Due to problems in SUNOS, the pulldown menu inside the plot window must be installed explicitly by selecting the 'create pulldown-menu' menu entry inside the plotwindow pop-up menu.

In SUNOS and Linux it works fine, but under SUNOS, crashes occurr which could not fixed, even with extensive debugging. This may be a problem inside the used XVIEW toolkit provided by SUN Microsystems.

24.2 Faster Editing

Direct Edit means that you do not have to look for simulation or parameter boxes , which is quite difficult if thay are placed on a large schematic.

Instead, pull the pullright menu Direct edit and select what you want to edit. The property editor will pop up immediately.

24.3 threads in SPICECAD: run SPICE3 in background

On SUN SOLARIS machines and Linux machines, SPICE3 can be run in background. This means, that

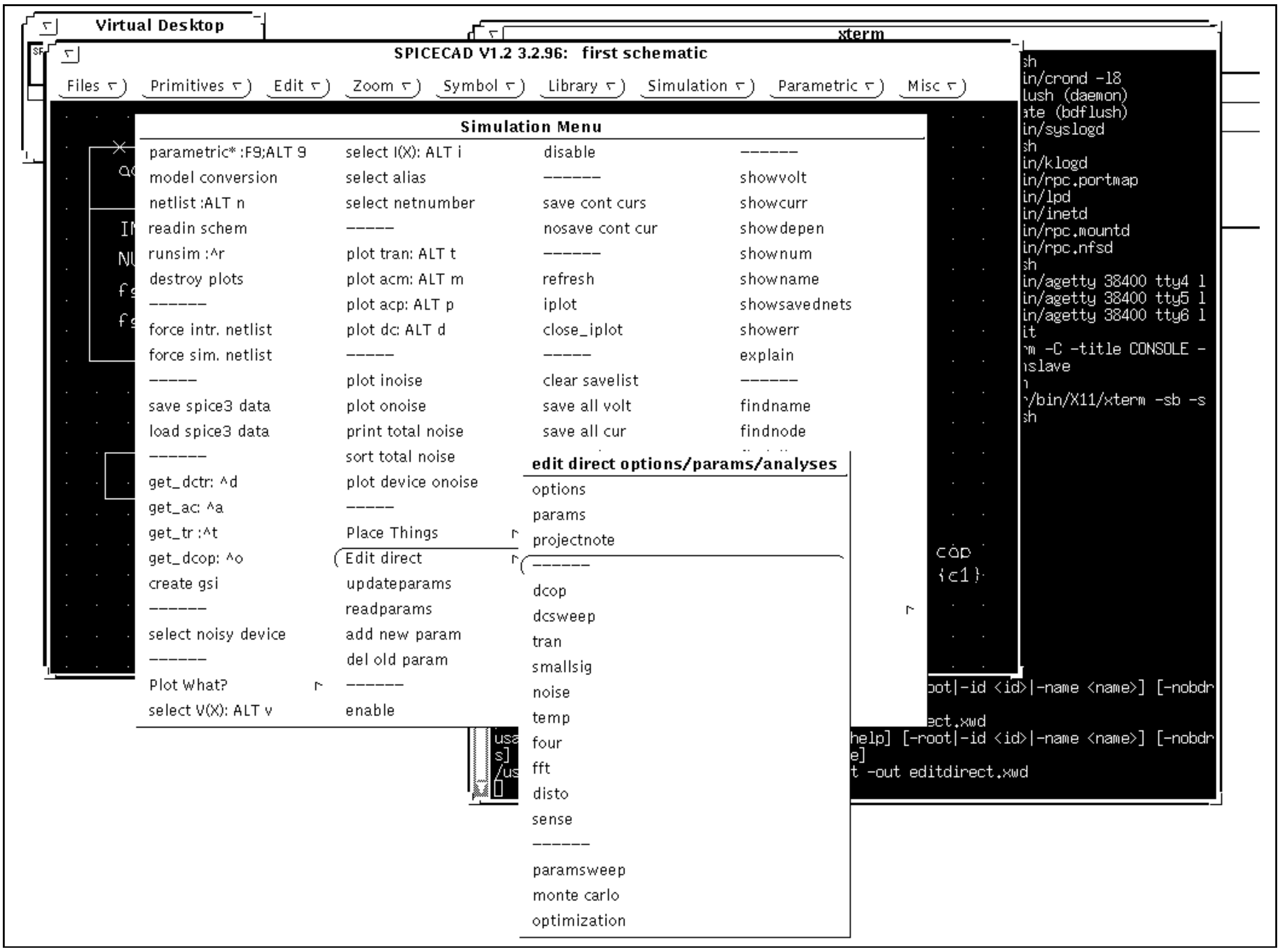
instead of calling directly the function which executes SPICE3: `function$runsim()`

it is called this way:

```
pthread_create(&thread,(pthread_attr_t *) NULL,(void * (*)(void *)) function$runsim
```

So, instead of blocking the SPICECAD graphics entry when running SPICE3 using 'runsim', you can say 'run in background' and carry on editing your schematics. After a few moments after starting the simulation, you can select a net which you want to plot and plot it. In the plotwindow, the curve you wanted will appear . If you run a longer transient simulation, the curve will stop somewhere before the expected final simulation timepoint. When you want to display what has happened a few seconds later, you can use 'zoomout' to redisplay the plot. You will discover that the last timepoint of the simulation curve approached the final timepoint. This way, you can check (for example) whether the circuit that you have designed is oscillating.

`If you run SPICE3 in background, is is possible to start a second simulation, in foreground or in background. You should NOT do that because this will result in unpredictable behaviour of SPICE3 and SPICECAD.`



```
sh
in/crond -l8
lush (daemon)
ate (bdflush)
in/syslogd
sh
in/klogd
in/rpc.portmap
in/lpd
in/inetd
in/rpc.mountd
in/rpc.nfsd
sh
in/agetty 38400 tty4 l
in/agetty 38400 tty5 l
in/agetty 38400 tty6 l
it
m -C -title CONSOLE -
slave
h
/bin/X11/xterm -sb -s
sh
oot|-id <id>|-name <name>] [-nobdr
ect.xwd
help] [-root|-id <id>|-name <name>] [-nobdr
e]
t -out editdirect.xwd
```

Figure 24.1: 'Edit direct' menu

Chapter 25

Command Index

25.1 Place commands

Here is a list of all commands which you can find inside the menus. The following entries have to be used to place devices like lines, circles, text labels, transistors, resistors, pins etc. :

$\boxed{\text{text}}$: Place text label. $\boxed{\text{circle}}$,$\boxed{\text{rect}}$,$\boxed{\text{line}}$,$\boxed{\text{cont}}$,$\boxed{\text{econt}}$,$\boxed{\text{sline}}$: Placing circles, rectangles, lines, contacts, extended contacts , multiple segment lines.
$\boxed{\text{nmos}}$, $\boxed{\text{pmos}}$, $\boxed{\text{n4mos}}$, $\boxed{\text{p4mos}}$: MOS Transistors
$\boxed{\text{nbip}}$, $\boxed{\text{pbip}}$, $\boxed{\text{n4bip}}$, $\boxed{\text{p4bip}}$: Bipolar Transitors
$\boxed{\text{arsc}}$: SPICE3 arbitrary source
$\boxed{\text{vdc}}$, $\boxed{\text{idc}}$: DC current and voltage source
$\boxed{\text{vsin}}$, $\boxed{\text{isin}}$: current and voltage source with sinusoidal transient characterictics.
$\boxed{\text{vpul}}$, $\boxed{\text{ipul}}$: current and voltage source with pulse transient characterictics.
$\boxed{\text{vpwl}}$, $\boxed{\text{ipwl}}$: current and voltage source with piecewise linear transient characterictics.

$\boxed{\text{vexp}}$, $\boxed{\text{iexp}}$: current and voltage source with exponential transient characterictics.

$\boxed{\text{vsffm}}$, $\boxed{\text{isffm}}$: current and voltage source with frequency modulation transient characterictics.

vcvs, vccs, cccs, ccvs: Linear voltage and current controlled voltage and current sources.

_vcvspoly, _vccspoly, _cccspoly, _ccvspoly: Voltage and current controlled voltage and current sources with POLY characteristics (old SPICE2 stuff), but necessary for the inclusion of opamp of other macromodels.

res, cap, ind : linear resistor, capacitor and inductance.
sem. cap, sem. res : linear semiconductor capacitor and resistor.

mutind, trline, trloss, rcline : mutual coupled inductance, lossy and lossless lines
vswit, iswit, diode: voltage and current controlled switch, diode
njfet, pjfet, nmesfet, pmesfet : JFETs and MESFETs.

frame: Nice frame to generate good looking schematic plots for customers.

inport : create input port to define connections between different schematic hierarchy levels .
outport : create input port to define connections between different schematic hierarchy levels .
ioport : create input port to define connections between different schematic hierarchy levels .

named pin : create pin for local named connections (connections between different nets which are attached to named pins of the same name)

vss, vdd, vss1, vdd1: Global pins.

gnd: Global pin ground (node 0), must appear at least once in the schematic hierarchy (and on the top level).

named global : If the number of pre-defined globals vss, vdd, vss1, vdd1 is not sufficient for your design, you can create more globals with arbitrary names. The only restriction is that every global must appear at least once on the top level schematic. To satisfy this requirement, you can connect the named global on top level with 1GOhm resistor to whatever you want .

options : Places simulation box which generates a .options - statement during netlisting.

params: Box which stored user-defined design variables.

projectnote: Box stores notes.

dcop: Places simulation box which generates a .op -statement during netlisting.

dcsweep: Places simulation box which generates a .dc -statement during netlisting.

smallsig: Places simulation box which generates a .ac -statement during netlisting.

noise: Places simulation box which generates a .noise -statement during netlisting.

tran: Places simulation box which generates a .tr -statement during netlisting.

four: Places simulation box which generates a .four -statement during netlisting.

paramsweep: Places simulation box which controls a variable parametric sweep.

monte carlo: Places simulation box which controls a monte carlo simulation.

optimization: Places simulation box which controls an optimization .

temp: Places simulation box which controls a temperature sweep .

25.2 Schematic menu Commands

- `'add new param' : Adding a new user -defined variable to the parameter box.`
- `'add symprop' : Symbols can have local properties which override global properties. (Like inverters with different transistor geometries). If you are in symbol editing mode (cad -s symbolname), you can add local properties to the symbol.`
- `'add select v(x)' : This command enables to to add nets to the nets which voltages you want to plot.`
- `'add to group' : By selecting a group with a click of the left mouse button, you can define the group into which more elements shall be moved. With a second click of the left mouse button, you can define the element which shall be moved.`
- `'all devices' : Find and display all devices which could cause problems because they are not saturated (MOS) or if they are saturated (bipolar). Concerning bipolar transistors, use the 'options' -dialog to define the bipolar saturation threshold voltage. Also, look at 'bsim in linear region','jfet in linear region' 'mesfet in linear region', 'bjt saturated'.`
- `'about' : Displays the SPICECAD logo. Hope you like it.`

- 'about author' : Displays a photo of mine.
- 'bjt saturated' : Find and display all bipolar transistors which are saturated. Use the 'options' -dialog to define the bipolar saturation threshold voltage.
- 'bsim in linear region' : Find and display all BSIM MOS transistors which are not saturated (SPICE3 only).
- 'bug reports' : needed!
- 'check schematic' : Schematics can be checked whether they will be netlisted without causing netlisting errors. This is useful if you do some editing on deep hierarchy levels. Without the need to go to the top level and issue a 'netlist' command, you can check locally whether everything seems to be okay. This command hasn't been tested thoroughly and might cause segmentation violations.
- 'clear group' : This command flattens the group you have defined before using 'win add to group' . 'Flatten' means that the object called 'group' , which is nothing but a container of different schematic objects, is destroyed after having moved the contents of the container back into the drawing area.
- 'close plotwindow' : Closes Plot Window.
- 'copy unspecific' : copies elements of the schematic which you select with a click of the left mouse button. You can copy everything.
- copy filtered' : copies elements of the schematic which you select with a click of the left mouse button. The elements which are selectable can be defined by using the 'select filter' command before. The default filter is 'no nets', which means, if you use 'copy filtered', you can copy everything but nets. This is very useful if you want to copy pins and globals.

- ’copy group’ : copies groups only.

- ’copy group in buffer’ : This command copies the group you have defined before into the cut--and--paste buffer. The elements of the group will remain on the schematics screen, but a copy of these elements also exists in the buffer. cut--and--paste buffer

- ’copy property’ : With this function, property values can be copied from one schematic element to one or more elememts of the same kind.

- ’create gsi’ : HSPICE only. When you place ’others’ on the schematic and want to plot node voltages inside its netlist, ’create gsi’ generates a lot of nets with the net alias property set to the names of the nodes stored in the HSPICE results file. If you select such a net, SPICECAD will not take the node number and convert it into a node name, SPICECAD will take the alias name and will search for the appropriate reults in the results database. To tell SPICECAD where to display the nets, you have to place a text element ’@gsi’ on the schematic. To tell SPICECAD that it has to use the net alias instead of the node number, use the function ’select alias’ (see there). To switch back to default mode (take net number), use ’select netnumber’ (see there).

- ’create new net’ : Start creating and editing a new. Place net , start command also with hotkey ’n’. Command execution stops when pressing right mouse button. Hotkeys ’v’ and ’h’ change drawing style to ’first vertical , then horizontal’ or ’first horizontal , then vertical’. Pressing ’s’ warps the mouse to the next contact or net, whichever is nearer, and simulates a left mouse button press (continiue command execution) -> snap function. Pressing ’S’ warps the mouse to the next contact or net, whichever is nearer, and simulates a right mouse button press (stop command execution) -> snap function.

- 'currents' : If you want to plot currents instead of voltages, use this command. Use 'voltages' to switch back to voltage plot mode (the default).

- 'cut group in buffer' : This command moves the group you have defined before into the cut--and--paste buffer. The elements of the group will disappear from the schematics screen, but they still exist. cut--and--paste buffer

- 'define clip region' : If you want to plot portions of the schematic window you are editing at present, use 'define clip region' to define a rectangle which you wish to see on the plot. Things outside this clip region will not be plotted if they lie completely outside. This they are partly inside and partly outside the clip region, the part which lies inside will be visible.

- 'delete unspec.' : deletes elements from the schematic which you select with a click of the left mouse button. You can delete everything. The 'deleted' elements are not destroyed, they are moved into a buffer from where you can access them using the 'undelete' command.

- 'delete filtered' : deletes elements of the schematic which you select with a click of the left mouse button. The elements which are selectable can be defined by using the 'select filter' command before. The default filter is 'no nets', which means, if you use 'delete filtered', you can delete verything but nets. This is very useful if you want to delete pins and globals.

- 'delete group' : deletes groups only.

- 'delete all' : deletes everything from the schematic (big cleanup).

- 'del old param' : Delete obsolete user defined variable from the parameter box .

- 'del netsegments' : Delete a part (segment) of a net.

- 'destroy' : Delete all SPICE3 simulation results . Due to the fact that these results are stored in the RAM, this function should be executed if your computer begins swapping.

- 'disable' : Disables netlisting for a special device. If a device is disabled, it will not be netlisted. This is especially useful to control the excution of certain simulations (disable simulation box).

- 'disable net' : Disables netlisting for a net. If a device is disabled, it will not be netlisted. This function is useful only if a net is connected to contacts of devices which are disabled. A non-disabled net would cause unnecessary netlisting errors.

- 'don't use cpp' : Stop using the preprocessor cpp to post-process netlists. See chapter 'Simulation', section 'Using the pre-processor cpp'.

- 'double view' : Magnifies the viewport of the schematic window by a factor 2.0 (zoom in).

- 'down' : Moves the viewport of the schematics window vertically. The same function can be executed using the cursor keys.

- 'edit net props' : Start net property editing dialog after selecting a net which you want to edit.

- 'edit primitives' : Lets you edit primitives like lines, slones, circles and rectangles after having placed them.

- 'edit property' : Start property editing dialog after selecting a schematic element which you want to edit. Nets cannot be selected, for editing net properties, use function 'edit net props'.

- 'edit schematic' : Going down into the next schematic hierarchy level. You will be asked to save the schematic you are editing at present.

- 'edit schematic nosave' : Going down into the next schematic hierarchy level. You will not be asked to save the schematic you are editing at present.

- 'edit symbol' : When clicking at a symbol with the left mouse button, you can edit the symbol itself in--place. You will be asked to save the schematic you are editing at present.

- 'edit symbol nosave' : When clicking at a symbol with the left mouse button, you can edit the symbol itself in--place. You will not be asked to save the schematic you are editing at present.

- 'enable' : Enables netlisting for a special device. If a device is disabled, it will not be netlisted. This is especially useful to control the excution of certain simulations (disable simulation box).

- 'exit ' : Exits SPICECAD. You will be asked to save the schematic if necessary.

- 'explain' : Gives you the error message of a device which could not be netlisted without faults.

- 'findnode' : If you get simulator or netlister errors referencing nodes, you can use this command to find these nodes. Even if the nodes is in a deeper hierarchy level, it will be found and displayed.

- 'findalias' : Many schematic elements contain properties called 'alias'. With this command, you can find all elements which have a property called 'alias' and which is set to a certain value.

- 'find prop' : With this command, you can find elements which have a property set to a certain value. Only the value can be defined using this search function.

- ’findname’ : If you get simulator or netlister errors referencing element names, you can use this command to find the elements which have these names.

- ’find first’ : Each ’find’ command can result in many search results. A search list will be created in this case where all elements which match the search criterion will be stored in. A pointer points to the current element inside the list. ’find first’ resets the pointer to the beginning of the list.

- ’find next’ : Each ’find’ command can result in many search results. A search list will be created in this case where all elements which match the search criterion will be stored in. A pointer points to the current element inside the list. ’find next’ advances the pointer to the next element of the list.

- ’find prev’ : Each ’find’ command can result in many search results. A search list will be created in this case where all elements which match the search criterion will be stored in. A pointer points to the current element inside the list. ’find prev’ advances the pointer to the previous element of the list.

- ’force intr. netlist’ : When symbols are used with local properties attached, the transistor geometry inside the schematic below changes between different symbols. Thus, the .param - statement cannot be used to define design variables because these so defined variables are global. When using HSPICE, it is necessary to replace expressions like

  ```
  M12 1 2 1 0 nmod l='lmax-0.5u' w='wft*imax/10u'
  ```

  by

  ```
  M12 1 2 1 0 nmod l=1.5u w=13u
  ```

  if this transistor geometry is defined by local symbol properties. ’force intr. netlist’ tells SPICECAD no to use the .param -statement when the parameter box is netlisted. Instead, SPICECAD will evaluate geometry expressions and netlist the value which results from the evaluation. You do not need to think about this mechanism when you start simulations. SPICECAD will detect local properties attached to symbols automatically , will pop up a warning for you and will switch automatically into 'intrinsic netlisting' mode.

- 'force sim. netlist' : This is the opposite to 'force intr. netlist'. When netlisting the parameter box , .param - statements will be generated :
  .param lmax=10u
  .param wft=100u
  etc.
  when you use HSPICE as a simulator. For other simulators like SPICE3 and TITAN which do not understand this .param - function, a netlist will be generated where all expressions will be resolved.
- 'full view' : Magnifies/fits the viewport of the schematics window so that all elements will be visible afterwards.
- 'gen oth from file' : 'other' elements serve as containers for subcircuit netlists. For example, if you have a file called 'ad432.m' containing an opamp subcircuit (can be a transistor level model or a macromodel or whatever), you can let SPICECAD parse this file looking for node names. SPICECAD assumes that there is a line inside the file which looks like this:
  .subckt ad432 1 2 3 4
  SPICECAD will identity the nodes 1, 2, 3 and 4 and will display them in a dialog box. There, you can define the outlook of the 'other' element which will be created automatically.

  If you do not like the outlook of the 'other' element, you can edit it using 'cad -o othername' .
- 'gen autom. symbol' : This command starts a dialog where you create and define symbols. The whole schematic you are editing will be searched for 'ports' . The port

names will be displayed in the dialog box. You can choose whether you want them to be displayed in the symbol afterwards. For each port, a contact will be created inside the symbol view with the port name attached to it. This way, ports in the schematic view and contacts inside the symbol view serve as interfaces between different hierarchy levels, which means, nets connected to the symbol pins will automatically be connected to the nets attached to ports insid ethe underlying schematic . Port names which do not appear at the symbol view will have no connection to higher hierarchy levels.

If you do not like the outlook of the symbol, you can edit it using 'cad -s symbolname' .

- 'gen sym. from group' : Normally, symbols are generated from schematics. If you want to make a part of a schematic a symbol, use this command. First, you have to use the 'win add to group' and 'add to group' commands to create a group. The group should contain some ports (SPICECAD version < 1.7.4 : named pin) which will serve as electrical interfaces to the higher hierarchy level afterwards. After having searched the group for ports, a dialog box will pop up displaying the names of the ports. Now, you can define the outlook of the symbol.

  After having finished the dialog, the group you have defined disappears from the screen, and the new symbol will appear at the same space.

  If you do not like the outlook of the symbol, you can edit it using 'cad -s symbolname' .

- 'get dctr results' : Get DC transfer simulation results. Normally, SPICECAD is able to detect when a simulation has finished, and reads the simulation results automatically. If you wish to read in simulation results from later simulations, use this command to read results explicitly.

- 'get ac results' : Get small signal simulation results. Normally, SPICECAD is able to detect when a simulation has finished, and reads the simulation results automatically. If you wish to read in simulation results from later simulations, use this command to read results explicitly.

- 'get tr results' : Get transient simulation results. Normally, SPICECAD is able to detect when a simulation has finished, and reads the simulation results automatically. If you wish to read in simulation results from later simulations, use this command to read results explicitly.

- 'get dcop results' : Get DC operating point and noise simulation results. Normally, SPICECAD is able to detect when a simulation has finished, and reads the simulation results automatically. If you wish to read in simulation results from later simulations, use this command to read results explicitly.

- 'get temp results' : Get the results of a temperature sweep when setting different temperatures inside the TEMP simulation box. This feature is available in HSPICE , only. Normally, SPICECAD is able to detect when a simulation has finished, and reads the simulation results automatically. If you wish to read in simulation results from later simulations, use this command to read results explicitly.

- 'half view' : Magnifies the viewport of the schematic window by a factor 0.5 (zoom out).

- 'help' : Displays help. After having selected this item, you can click on a menu entry of a schematic element. Depending on what you do, a help text will appear .

- 'jfet in linear region' : Find and display all jfet transistors which are not saturated.

- ’leave’ : Leaves a menu level. Due to the fact that only two menu levels exist, the main menu will pop up if you hit the right mouse button.

- ’lhand’ : Moves the viewport of the schematics window horizontally. The same function can be executed using the cursor keys.

- ’library menu’ : Switch to the ’library’ sub-menu. Here, you can place all the things you need inside an electronic circuit.

- ’load schematic’ : Load schematic. The schematic window you are editing at present should be empty if you issue this command because the elements to be loaded are added to the current schematic. If the schematic window is not empty, use $\boxed{\texttt{delete all}}$.

- ’load other’ : Load other element.

- ’load symbol’ : Load symbol.

- ’load group’ : If you want to import elements from other schematics, you can use ’save group’ to export them and ’load group’ to import them.

- ’load spice3 data’ : You can load a spice3 file ’rawspice.raw’ into the memory.

- ’lookat netlist’ : Pops up an editor window which displays the netlist. You can choose the editor using the ’option1’ menu.

- ’measure’: Define pairs of points on the plot result curve with the mouse -> get (x,y),(x1,y1) SPICECAD will calculate x1-x, y1-y, (y1-y)/(x1-x) etc.

- ’mesfet in linear region’ : Find and display all mesfet transistors which are not saturated.

- ’mirror’ : mirrors elements. Simulation boxes cannot be mirrored. Elements can be mirrored vertically and horizontally. The mirror axis is defined using the mouse.

- ’model conversion’ : Opens the ’model conversion’ dialog where you can map your schematic transistor models to netlist models depending on transistor geometries (MOS transistors only). This feature is useful if your MOS model consists of several sub-models which are valid only for a limited range of MOS gate lengths and widths.

- ’mos in linear region’ : Detects and display MOS transistors where uds < (ugs-uth) .

- ’move unspecific’ : moves elements of the schematic which you select with a click of the left mouse button. You can move everything .

- ’move filtered’ : moves elements of the schematic which you select with a click of the left mouse button. The elements which are selectable can be defined by using the ’select filter’ command before. The default filter is ’no nets’, which means, if you use ’move filtered’, you can move everything but nets. This is very useful if you want to move pins and globals.

- ’move group’ : moves groups only.

- ’netlist’ : Starts the netlisting routine. Normally, SPICECAD itself detects when it is necessary to generate a new netlist. But, if yopu have changed a model file, SPICECAD will not check that, and if you have changed nothing but this model file, the netlist will not be redone which means you will run a wrong simulation afterwards.

- ’net stretch’ : If you run out of space inside your schematic window, you can use this command to get space where you need it. See chapter ’Advanced Editing:

Grouping, Stretching, Filtered selection'. This command is easier to use than the 'stretch' function because only a single net segment can be moved preserving the connectivity.

- 'new name' : renames the schematic you are currently editing. The new name will be displayed on the top of the schematic window.
- 'nosave cont curs' : This function resets the 'save contact currents' flag from a device of symbol.
- 'open plotwindow' : Opens the plot window.
- 'plot options' : Pops up the plotwindow option dialog box.
- 'option dialog' : Pops up the first of two option dialog boxes.
- 'option1 dialog' : Pops up the second of two option dialog boxes.
- 'pan' : Moves the schematic viewport by a vector which will be defined using the mouse.
- 'parametric menu' : Switch to the 'parametric' sub-menu. This menu is a simulation menu, here, you can run parametric simulations.
- 'paste group from buffer' : This command takes the contents of the cut--and--paste buffer cut--and--paste buffer and copies it into the drawing area . The buffer itself is unchanged afterwards.
- 'paste group from buffer and clear' : This command takes the contents of the cut--and--paste buffer cut--and--paste buffer and moves it into the drawing area . The buffer itself is empty afterwards. The buffer contents are stored as a 'group', so, you can move them over the screen simultaneously. Use 'clear group' to flatten the group .

- 'place other' : Starts dialog for placing an 'other' element .

- 'place symbol': Starts dialog for placing a symbol .

- 'plot tran' : Plot transient simulation results. It is necessary to select nets or terminals before you issue this command.

- 'plot dc' : Plot dc-sweep simulation results. It is necessary to select nets or terminals before you issue this command.

- 'plot acm' : Plot small signal simulation results (magnitude). It is necessary to select nets or terminals before you issue this command.

- 'plot acp' : Plot small signal simulation results (phase). It is necessary to select nets or terminals before you issue this command.

- 'plot eps with autom. scaling and rotating' : Generates an EPS plot of the schematics window you are editing at present. Depending on the aspect ratio, the result will be rotated by 90 degrees if its height is larger than its width. (landscape orientation versus portrait orientation) The plot results will be written into file 'tmp.eps' indexfile tmp.eps

- 'plot eps with autom. scaling,suppress rotate' : Generates an EPS plot of the schematics window you are editing at present. The plot orientation will always be landscape, which makes it easier to incorporate these plots into documents. The plot results will be written into file 'tmp.eps' indexfile tmp.eps

- 'plot hpgl' : Generates a HPGL plot of the schematics window you are editing at present.

- 'plot inoise' : Plot input refered noise after running a noise simulation. It is not necessary to select nets or terminals before you issue this command.
- 'plot menu' : Switch to the 'plot' sub-menu.
- 'plot onoise' : Plot output refered noise after running a noise simulation. It is not necessary to select nets or terminals before you issue this command.
- 'readin schem' : SPICE3 only. SPICE3 will read the actual netlist if you execute this command. SPICECAD will not check whether the netlist differs from the schematic. SO, if necessary, execute 'netlist' before.
- 'readparams' : Reads user defined variables from file 'varlist' .
- 'redraw window' : Redraws the schematic.
- 'redraw plot' : Redraws the plot window.
- 'registration' : Sends a mail to support@spicecad.com The mail tells us your user name, hostname and platform (Linux, SUNOS, Solaris). We like to know how many people use the freely available version of SPICECAD.
- 'reload' : Reloads all symbols and underlying schmatics. This function is useful if you have edited symbols or schematics in a parallel SPICECAD session and want to update the session you are working on at present.
- 'remove gsi' : HSPICE only. After having created nets using 'create gsi' , you can remove all this stuff using this command.
- 'report reds' : Display schematic hierarchy.
- 'return' : Go to the next higher schematic hierarchy level. You will be asked to save the schematic you are editing at present.

- 'return nosave' : Go to the next higher schematic hierarchy level. You will not be asked to save the schematic you are editing at present. After leaving SPICECAD, all edits you have done will be lost.

- 'rhand' : Moves the viewport of the schematics window horizontally. The same function can be executed using the cursor keys.

- 'rotate': rotate elements. Simulation boxes cannot be rotated.

- 'runsim' : Start simulation.

- 'run in background' : SOLARIS only. By using the pthread library, SPICE3 can run in the background. In foreground, you are able to make schemtaic edits or plot intermediate results.

- 'run monte' : Start Monte Carlo Simulation.

- 'run optim(grad)' : Start Optimization. A gradient search algorithm is used to find the optimum.

- 'run optim(monte)' : Start Optimization. A random search algorithm is used to find the optimum.

- 'run paramsweep' : Start parametric sweep.

- 'save schematic' : saving schematic window. You can save the schematic under a new name. The schematic name you are editing at present will not be changed. This can be done explicitly using 'new name'.

- 'save group' : If you want to import elements from other schematics, you can use 'save group' to export them and 'load group' to import them.

- 'saveit' : Saves the current schematic, symbol or other element .

- 'save edif' : Under construction.

- ’save other’ : saving other. You can save the other element under a new name. The other name you are editing at present will not be changed. This can be done explicitly using ’new name’.

- ’save symbol’ : saving symbol. You can save the symbol under a new name. The symbol name you are editing at present will not be changed. This can be done explicitly using ’new name’.

- ’save cont curs’ : By default, only voltage source currents are saved by the simulators. This function adds zero volts dc voltage sources in series to the contacts in whose currents you are interested in. The contact boxes of the selected device or symbol will be drawn in yellow afertwards.

- ’save spice3 data’ : Save SPICE3 simulation results in file ’rawspice-raw’. .

- ’select alias’ : For HSPICE only. When you place ’others’ on the schematic and want to plot node voltages inside its netlist, you can use the ’create gsi’ function (see tehre) which generates a list of all nodes stored in the simulation result files. ’select alias’ means that SPICECAD takes the net alias (equals the node name) and plots the according voltage.

- ’select filter’ : Lets you choose betwen different filters. This filter affects the ’copy filtered’, ’move filtered’ and ’delete filtered’ functions: only those schematic elements can be selected (and afterwards moved, copied of deleted) if they match (pass) the filter.

- ’select netnumber’ : For HSPICE only. The default mode. You have to execute this command if you have executed ’select alias’ before and want to go back to the default mode.

- 'select noisy device ' : This function displays the noise contribution of a selected device (resistors and transistors)
- 'select v(x)' : This command can be used to select net voltages you want to plot. A click with the left mouse button selects which nets you are interested in.

  Several different nets can be selected with left mouse button clicks. The selection has to be finished with a click of the right mouse button.
- 'select i(x)' : This command can be used to select terminal currents you want to plot. A click with the left mouse button selects which terminal you are interested in.

  By default, you can select voltage source terminals. If you have told the simulator to save the contact currents of elements like transistors and symbols (by using the command 'save cont curs'), you can select the yellow drawn terminals of these schematic elements.

  Several different terminals can be selected with left mouse button clicks. The selection has to be finished with a click of the right mouse button.
- 'show clip region' : Displays the plot clip region. See 'define clip region'.
- 'showbox' : For debugging purposes only. Displays boundary box of schematic elements.
- 'showcont' : For debugging purposes only.
- 'showdepen' : tells you where you can find voltage sources which are referred to in noise simulation of dcsweep simulation boxes. Click at the box, and a line will be drawn between the box and the source(s). Looks very impressive.
- 'show status' : Display internal information of schematic elements.

- ’showdevice’ : Displays DC-operating point information about a device. WHen using SPICE3, the information will be displayed inside the xterm window where you have started SPICECAD. When using remote simulators like HSPICE or TITAN, the well known information box pops up.

- ’showgroup’ : The groups inside the schematic window will flash several times. This makes it easier to find them if you have forgotten where you defined them.

- ’showcurr’ : This command can be used to display DC--operating point soure currents. A click with the left mouse button selects which terminal you are interested in, and with the next click , you can define where you want to display the current value inside the schematic window. After a redraw, this value will vanish.

  By default, you can select voltage source terminals. If you have told the simulator to save the contact currents of elements like transistors and symbols (by using the command ’savec ont curs’), you can select the yellow drawn terminals of these schematic elements.

- ’showerr’ : Displays devices which could not be netlisted correctly or where errors occurred during netlisting.

- ’showname’ : Lets you display the instance name of a device.

- ’shownum’ : Lets you display the net number of a net. Both analog and digital net number are displayed. Topmost is analog , bottom is digital.

- ’showvolt’ : This command can be used to display DC--operating point voltages. A click with the left mouse button selects which net you are interested in, and with the next click , you can define where you want to display the voltage value inside the schematic window. After a redraw, this value will vanish.

- 'showtrvolt' : This command can be used to display the net voltage of the last transient simulation timestep . A click with the left mouse button selects which net you are interested in, and with the next click , you can define where you want to display the voltage value inside the schematic window. After a redraw, this value will vanish.

- 'simulation menu' : Switch to the 'simulation' sub-menu. Here, you can place simulation-, parameter- and options boxes, define and start simulations, select results which you want to plot.

- 'single stretch' : If you run out of space inside your schematic window, you can use this command to get space where you need it. See chapter 'Advanced Editing: Grouping, Stretching, Filtered selection'. This command is easier to use than the 'stretch' function because only a single element can be moved preserving the connectivity.

- 'stretch' : If you run out of space inside your schematic window, you can use this command to get space where you need it while preserving the connectivity of the devices you want to move. See chapter 'Advanced Editing: Grouping, Stretching, Filtered selection'. This command is the most flexible one, but also difficult to use.

- 'stretch window' : If you run out of space inside your schematic window, you can use this command to get space where you need it while preserving the connectivity of the devices you want to move. See chapter 'Advanced Editing: Grouping, Stretching, Filtered selection'. This command is easier to use than the 'stretch' function because the elements which shall be moved must lie inside a rectangle.

- 'symbol menu' : Switch to the 'symbol' sub-menu. Here, you can place symbols and others, generate symbols and others, and more things.

- ’take over’ : After an optimization run, it must be decided whether the results are better than before or not. If the results are better, ’take over’ copies the optimized design variables into the parameter box. . You do not need to worry if you have forgotten to say ’take over’, or if SPICECAD crashes during optimization: all intermediate results will be written into file ’opti.log’ . Thus, you can restore the optimized design variables by hand. A second set of files will be opened during optimization, called ’varlistN’, where N is the optimization iteration number. . By copying file ’varlistN’ to ’varlist’, you can easily read in the design variables of the N-th optimization iteration.

- ’tmp.eps to lpr’ : The plot results of the schematics window (found in file tmp.eps are written to the printer. The printer can be defined using the ’options1’ -menu.

- ’totop’ : Go immediately to the top of the schematic hierarchy. You will be asked to save the schematic you are editing at present.

- ’totop nosave’ : Go immediately to the top of the schematic hierarchy. You will not be asked to save the schematic you editing at present. If you leave SPICECAD, all edits will be lost.

- ’undo menu’ : Switch to the ’undo’ sub-menu.

- ’ungroup’ : This command can be used if objects which are inside a group shall be removed from this group. First, selecting the group from which an object shall be extracted is necessary (left mouse button). Afterwards, a second click with the left mouse button selects the object which shall be extracted from the group.

- ’unselect v(x)’ : This command enables you to remove nets from the list of net voltages you want to plot. Each left click at a net removes it from the list of nets to plot. A right click stops the unselection process.

- 'unzoom' : Set the schematic viewport to the default value :
  viewport= (0 0 window_width window_height).

- 'up' : Moves the viewport of the schematics window vertically. The same function can be executed using the cursor keys.

- 'updateparams' : Searches for new user defined variables, adds them into the parameter box and pops up the property editor if necessary.

- 'use cpp (model sweep)' : You can use the preprocessor 'cpp' to convert the netlists generated by SPICECAD before they are read into the chosen simulator. This way, you can perform sweeps even over model parameters. Look at chapter 'Simulation', section 'Using the pre-processor cpp' how to use cpp.

- 'voltages' : The default. By default, only voltages are plotted. If you want to plot currents instead, use 'currents' .

- 'win add to group' : By drawing a window, you can define the elements which shall be moved inside a container called 'group'. This container can be moved, copied or deleted, mirrored and rotated. All elements inside this group will be affected by all these mentioned commands simultaneously.

- 'zoom in' : Define a zoom-in window using the mouse.

- 'zoom menu' : Switch to the 'zoom' sub-menu.

25.3 Plotwindow menu commands

- '*calculator' : This function pops up the waveform calculator.

- ’db’ : Do apply the ’db’ function to the y values of the plotted curves before plotting.

- ’fftopt’ : This function pops up a dialog where you can define title, xlabel, ylabel, FFT window type, xrange, yrange, FFT start, FFT end, number of FFT points etc.

- ’fft on’ : An FFT can be run over the plot results. Different parameters can be set to achieve optimum results: you can choose the window type, the start and end points of the fft (x variable). The window type can be defined by popping up the ’options’ dialog.

- ’fft off’ : Do not run FFTs any longer.

- ’grid’ : Display a grid.

- ’groupdelay’ : If small signal simulation data are available, the groupdelay can be plotted without the need to start the waveform calculator.

- ’hardcopy hpgl’ : The plotwindow will be written as hpgl into file ’tmp.hgl’.

- ’hardcopy ps’ : The plotwindow will be written as postscript into file ’tmp.ps’.

- ’hardcopy table’ : The plotwindow will be written as an ASCII table into file ’tmps.txt’. Using this file, results can be post-processed using tools like MATLAB or MATHCAD.

- ’hori zoomin’ : Zoom into the plot results by defining a zoom viewport using the mouse. Only the x-range of the viewport will be changed, the y-range will be left unchanged.

- ’key’ : The meaning of the curves (which line type, which variable name) is displayed in the plot window in the upper right corner. This is the default. ’no key’ disables the display of this information. ’key’ enables the display.

- ’kill simulator’ : Kill remote TITAN or HSPUCE runs.

- ’loglog’ : Set x axis to logarithmic and y axis to logarithmic scaling.

- ’measure’ : Try it, it is hard to say what it does.

- ’mult-y-axis’ : Plot all curves in different coordinate systems.

- ’nodb’ : Do not apply the ’db’ function to the y values of the plotted curves before plotting.

- ’nogrid’ : Do not display a grid.

- ’nogroupdelay’ : If small signal simulation data are available, the groupdelay can be plotted without the need to start the waveform calculator. ’nogroupdelay’ switches back to normal AC display mode.

- ’nokey’ : The meaning of the curves (which line type, which variable name) is displayed in the plot window in the upper right corner. This is the default. ’no key’ disables the display of this information. ’key’ enables the display.

- ’nolog’ : Set x and y axis to linear scaling.

- ’notitle’ : The display of a title will be suppressed.

- ’noxlabel’ : The display of an x-label will be suppressed.

- ’noylabel’ : The display of an y-label will be suppressed.

- ’options’ : Pops up a dialog where you can define title, xlabel, ylabel, FFT window type, xrange, yrange, FFT start, FFT end, number of FFT points etc.

- 'plot setup ' : The settings of a plot window can be saved. The settings are: 'db/nodb', 'grid/nogrid', 'fft/nofft', 'nolog/xlog/ylog/xylog', 'key/nokey' , zoom viewport , plot variables etc. Four different buffers are available which store the setup information. By using the 'plot setup' function, the plot setups which have been saved before are restored, and the plot will be executed.

- 'repeat action': SPICECAD will behave again as if you have selected the last menu entry that you have selected before. Very useful if you have to place several devices of the same type (gnd pins, MOS transistors)

- 'save setup ' : The settings of a plot window can be saved. The settings are: 'db/nodb', 'grid/nogrid', 'fft/nofft', 'nolog/xlog/ylog/xylog', 'key/nokey' , zoom viewport , plot variables etc. Four different buffers are available which store the setup information. By using the 'plot setup' function, the plot setups which have been saved before are restored, and the plot will be executed.

- 'single-y-axis' : Plot all curves into a single coordinate systems. This is the default.

- 'style lines' : Plot curves as solid lines only.

- 'style linespoints' : Plot curves as solid lines and using points, too.

- 'style points' : Plot curves using points.

- 'title' : This function pops up a dialog where you can define title, xlabel, ylabel, FFT window type, xrange, yrange, FFT start, FFT end, number of FFT points etc.

- 'tmp.ps to lpr' : Send previously written file 'tmp.ps' to printer. The printer can be defined in the 'options1'-menu.

- ’update’ : During simulations which run in background (SPICE3 when using SOLARIS, HSPICE or TITAN) , the results file can be read and the results plot will be updated if additional data are available.

- ’vert zoomin’ : Zoom into the plot results by defining a zoom viewport using the mouse. Only the y-range of the viewport will be changed, the x-range will be left unchanged.

- ’x enforce’ : When you use the ’options’ dialog to define an xrange, this xrange will be used as the xrange for the plot window.

- ’xlabel’ : This function pops up a dialog where you can define title, xlabel, ylabel, FFT window type, xrange, yrange, FFT start, FFT end, number of FFT points etc.

- ’x noenforce’ : This function clears the pre-selected xrange.

- ’xlog’ : Set x axis to logarithmic and y axis to linear scaling.

- ’y enforce’ : When you use the ’options’ dialog to define an yrange, this yrange will be used as the yrange for the plot window.

- ’ylabel’ : This function pops up a dialog where you can define title, xlabel, ylabel, FFT window type, xrange, yrange, FFT start, FFT end, number of FFT points etc.

- ’y noenforce’ : This function clears the pre-selected yrange.

- ’ylog’ : Set x axis to linear and y axis to logarithmic scaling.

- ’zoomin’ : Zoom into the plot results by defining a zoom viewport using the mouse.

- 'zoomout' : Display the full view of the selected simulation results.

- 'zmin by 2' : The plot window viewport will be magnified by a factor of 0.5. The aspect ratio will be left unchanged.

- 'zmin hori by 2' : The plot window viewport x-range will be magnified by a factor of 0.5. The y-range of the viewport will be left unchanged.

- 'zmin vert by 2' : The plot window viewport y-range will be magnified by a factor of 0.5. The x-range of the viewport will be left unchanged.

- 'zmout by 2' : The plot window viewport will be magnified by a factor of 2.0. The aspect ratio will be left unchanged.

- 'zmout hori by 2' : The plot window viewport x-range will be magnified by a factor of 2.0. The y-range of the viewport will be left unchanged.

- 'zmout vert by 2' : The plot window viewport y-range will be magnified by a factor of 2.0. The x-range of the viewport will be left unchanged.

Chapter 26

FAQ

Here are some questions (and answers) of the past. Some are obsolete now, but it might be due to changes in the code, that some could reappear.

- `When I try to run spicecad (i.e.  cad -c) the following message appears:`

  ```
  Spice 1 -> cad -c called freehist -1000 calling cad
  sorry, you are not allowed to run spicecad contact
  support@spicecad.com
  ```

  This is due to a little license function which has been left in the code. Please set an environment variable SPICECAD_VALIDATE to 'unddassollallessein'

-
  ```
  When exiting spicecad, exit:  CTRL q and quit nosave:
  CTRL w produce
  leaving cad


  internal error -- segmentation violation


  exitspice:  CTRL e works no problem
  ```

  This error seems to be hard to reproduce: on some schematics and machines, crashes can be produced hitting these keys, on other machines and operating systems, crashes never happen. So, if you have a schematic which causes crashes, please send it to me for debugging.

- `Is there anyway to exit spicecad and have the vectors remain to check plots in spice? (Your plotting routine is much much better but I wanted to check transient parmsweep plot data)`

  You can use the 'save spice3 data' function in the simulation menu to store the last simulation. If you want to store earlier simulations, you can type in SPICE3 commands directly from within the mouse window (the lengthy window which appears first).

- `Transient Parmsweep problem: Curves which should lie on top of each other do not appear this way in the plot window.`

  This problem is solved now.

- `Tran plot questions The lines are labelled '0', 'V1', 'V2', 'V3', and 'V4' for the five step parameter sweep I did in the inverter.scm file (attached at end). Why does '0' correspond to the 5th step and not the first. Is my perception backwards or are the transient plots labelled in reverse order?`

  This is changed , now. After a paramsweep simulation, the variable names and values will appear in the plot window.

- `Plot feature questions When zooming in is it possible to show the zoom rectangle before the zoom takes place?`

  This is implemented now.

- `In the parameter sweep plots is it possible to plot more than one voltage?`

  Definitely not. This requires major changes in SPICECAD. What is more, plotting more than one variable makes is difficult to see what is what. I am thinking of multiple plot widows. What do you think about this?

- Is it possible to update all edit fields when the editprop menu is exited? I usually forget to hit the enter key after modifying a field - so it does not get updated.

  This is a feature of XVIEW: you have to hit a pre-defined key to confirm that the menu field you filled in is correct now and can be sent to the application program. In this case, ESCAPE and RETURN are the keys used for confirmation.

- The FET symbols I would like to have are in the ps file below (basic.eps). I think this would help the popularity in the "digital" world.

  Analog and digital MOS symbols are available now.

- I think the manual could use a little beefing up - a little more explaining and a few more detailed example. So that folks like me can follow a little better!!!

  This is always true. Thanks to everyone sending suggestions, bug reports and error messages which helps me improving the program and the manual and fixing the errors which are still undetected.

- When a param name is changed and updateparam ran, the old param still appears in the parameters box. Manually editing the scm file is the only way I can find to delete/remove parameters. Is there any other way??

  This bug is fixed now.

- Is there any way to remove the error circles and is there any way to use CTRL s or something other than CTRL r to run the simulator??

  Error circles can be removed by netlisting the circuit successfully.

- A definite don't: I hit CTRL c in the xterm window. This action caused spicecad to be "dead" - no input possible. Is there a way to reactivate spicecad? I had to exit spice and restart.

  Sorry, no way to re-activate SPICECAD. Yes, don't kill SPICECAD hitting CTRL c. Use CTRL z and kill -9 PID instead.

- `Alt key - The Alt key is the source of most of my 'unknown key' errors. I did change my keymap so that the alt keys also have the mod1 modifier bits. This seems to work much better!!! The remaining error is peculiar - after running the simulation Alt-d (plots dc) and then Alt-t (plots trans). However, if the same alt key is used with d the dc plot does not return. Using the other Alt key with the d does bring the dc plot back up. Is there something going on here?`

  Undetecable key errors reside from mapped keyboards.

  To circumvent this, you can re-define the key codes in file 'keys.spi'. Plase read chapter 'Hotkeys'.

- `Are there any future plans to extend the 40 character limit of a property string?`

  It could be made customizable due to a setting in the 'setup.spi'–file. Any more requests?

- `F4 then F5 sequence brings up the symbol menu then the library menu and leaves the symbol menu dead in the X background. Is there any way to remove "dead" menus?`

  Dead menus only appear under linux, not under SUN SOLARIS. A way to remove them is first removing the last menu (F5). Klick with the right mouse button outside the menu to make it disappear. Then, press the function key to redraw the dead menu (F4). Klick with the right mouse button outside the menu to make it disappear, and everything will be fine again.

- `Pin menus. Neat feature to keep menu visible but how to remove. When I closed the pinned menu spicecad exited!?! Is there a way to remove pinned menus without exiting spicecad?`

  This depends on the window manager used. OLWM,OLVWM and TWM can be used, FVWM not.

- `Hierarchy. Neat use of subcircuits - Are symbol files connected to the underlying scm files, i.e. When the` `scm file is changed are the symbol instances in all other hierarchial designs also updated?`

This depends on whether you have the 'sharedlists' entry in the file 'setup.spi' set to 'yes'. 'yes' means that all symbols with the name in your schematic share the same physical list of objects (trasistors, sources). 'no' means that each time a symbol is created during loading or after a symbol placement, a new list of objects is created and loaded from the apropriate symbol files. This means that , if sharedlists is set to 'yes', editing an object in a symbol affects all symbols, because the object you have edited only exists once. If sharedlists is set to 'no', editing an object in a symbol affects no other object in any other symbol, because the object you have edited is unique and is not related to any other objects in any other symbols. This is why you should use properties is symbols, it reduces the necessarity to leave and re-enter SPICECAD due to changed symbol object properties.

Setting 'sharedlists' to 'yes' also means the simulation result backannotation only works on the top hierarchy level or if you use symbols only once. Using symbols several times means that node number and device names are incorrect: node numbers and device names are those of the last symbol in the netlisting sequence. As an example, is you create a FLIPFLOP and use it 20 times, an internal node might have the node numbers 4, 23, 45 , 78 ... 121 during netlisting. If you want to look at this internal node after simulation, you can only look at node 121. The other nodes are invisible. So, if you develop cells, you should always set 'sharedlists' to 'no'.

- `setup.spi - Every time I start spicecad I get the following warning/error: an error has occurred during setup file 'setup.spi' readin near . sharedlists What is the problem?`

  The file 'setup.spi' has two rows (columns?). Each line of this file is read by reading two strings at a time , where the first string is the name and the second the value of a variable to be set. If a value string is missing (because it has been set to an empty string, perhaps), the next variable name is read and interpreted as the value. After the next readed line, a variable value is read and interpreted as a variable name. This results in the mentioned error message. So, please check you file

’setup.spi’ whether all lines contain exactly two strings. Please add the appropriate variable value if one is missing, or delete the whole line if the variable name is missing.

- ’s’ and ’S’ - What is the secret??? Sometimes it works wonderfully other times not at all as expected. I get alot of "reject point of net" messages and alot of node # has only one connection. To remove only one node connection I have to quit spicecad and restart and reread scm file - then errors disappear??? I am getting better but sometimes the ’s’ button restarts from the last node connected - even after I have exited the Library menu. Is there a science or is more of an acquired art?

  It has to be taken care that the mouse is near to the desired position inside a net or a contact you want to snap to. If not, it might happen that it will be snapped to the same position two or more times after another. Then, nets of length zero are created and removed immediately, which results in the described error message. Nets with only one connection which require leaving an re-entering SPICECAD : this is still a bug which has to be solved.

- Automatic symbol generation - seems neat!! However, if symbol is placed in scm file with same name then the following error/warning message appears: analysis box in hierarchy found in path Are all sources included in symbol?

  What is described there is recursive placement: a symbol, which contains itself in its own description.

  Please do not place symbols into their own schematic.

- SpiceCAD seems to be thoroughly developed and I was wondering if you would consider adding a netlist to schematic converter - this would provide a full LVS method in cooperation with a layout CAD program such as Magic.

  If there is anyone outside there who has written a LVS program, please contact me. I could rewrite the netlisting procedures of SPICECAD to provide the LVS with additional information.

Netlist to schematic converters are only useful in digital design. In analog design, too many information is lost when making a netlist from a schematic: the reverse way is very difficult to go.

- I get a segmentation violation and spicecad exits when I select simulation --> show bad devices --> MOSFETs linear region

  Did you run a simulation before ? Please check whether you have at least a DCOP box inside the schematic.

- I was not lucky - last night the machine locked up when I did a rename and then tried to save (CTRL s). The mouse still moves fine but no keyboard or mouse button iput is taken. I had to reboot and then X would not start - a two reboot error. Is there a way to have a hot button for the terminate now command? That is, when I need the terminate command the keyboard and mouse are locked up - mouse movement and hardware reset buttons are all thats left - what can I do?

  There is a button called 'terminate' which terminates SPICECAD immediately. But, the described behaviour above could not be reproduced by me. Rename works fine on my machine.

- Another SpiceCAD don't: 'c' copy symbol. When I type 'c' and then select a symbol the following results: segmentation violation and spicecad exits. Is this a definite no-no? Should I always use "Place Symbol"?

  Copying symbols is allowed. Again, the described behaviour could not be reproduced by me.

- Suggestion - When spicecad reaches general public - maybe have some sort of novice, intermediate, and advanced menuing system?!?

  Sounds interesting. Detailed suggestions welcome.

- win add to group - By the way grouping and setfill are great!!! When adding to a group in a dense schematic I accidentally selected something that was already in the group!!! Guess what happened - segmentation fault and an abrupt exit. This is when I became wise to setfill!!! Is there any way to prevent selection of items already in a group?

- Parameter sweep labelling looks good!!! Is it possible to sweep a parameter that is local to a symbol? I am trying and would like to see all the .sp files for each step in the parameter sweep. Is it possible to save all the .sp files generated during a parameter sweep?

  Is implemented now.

- I finally figured out my ’s’ and ’S’ problems -- user error! Whenever netting I found that I must move the mouse for ’s’ to work properly after starting a new net. That is, consider the following procedure: ’n’ then move near contact then ’s’ move to end ’S’ -- everything okay, continue with new net ’n’ then fun begins -- if mouse located before ’n’ and then ’s’ pressed then the previous net is connected to the new net! However, if mouse moved after ’n’ and then ’s’ pressed a new net is created -- try it out!!!

  No comments.

Chapter 27

Announcements

XSPICE will soon appear as part of SPICECAD for interactive mixed-mode simulation and extended analog behavioural modelling. XSPICE was written at Georgia Tech.

Chapter 28

Acknowledgements

This work was only possible by using software from different sources. These sources are:

SPICE3 , written at the University at Berkeley

GNUPLOT
Copyright(C) 1986 - 1993 Thomas Williams, Colin Kelley

The scalable font was provided by Michael Minnhaar.

The program to read in the HSPICE binaries was written by Dennis Ciplickas.

XSPICE , written at Georgia Tech.

Index

List of Figures

List of Tables