NAME

> simeye2 - simulation signal displayer / editor (old version)

SYNOPSIS

> **simeye2** [X options] [-Lh] [-l|-m|-c|-P|-T] [-a|-b] [cell]

OPTIONS

> Beside the standard X Toolkit options (e.g. to set the display, the geometry, and the foreground color of the program), the following other options may be specified:

> **-high_lowColor** *color*
>> Specifies the color for signal values that are **logic O** or **logic I**.

> **-undefinedColor** *color*
>> Specifies the color for signal values that are **logic X**.

> **-voltsColor** *color*
>> Specifies the color for analog signal values.

> **-pointerColor** *color*
>> Specifies another color for the screen pointer (default: canvas.foreground).

> **-L**   Load window settings each time new signals are read.

> **-h**   Display only help information.

> **-l**   Use **sls-logic** as simulation type (default).

> **-m**   Use **sls-timing** as simulation type.

> **-c**   Use **spice** as simulation type.

> **-P**   Use **pstar** as simulation type.

> **-T**   Use **plato** as simulation type.

> **-a**   Use **analog** representation for **sls** signals.

> **-b**   Use **digital** representation for **sls** signals (default).

DESCRIPTION

> *Simeye* is an X Windows based simulation user interface for viewing the results of the *sls* (1ICD), *spice* (1ICD), *pstar* (Philips) and *plato* (TU-Eindhoven) simulators. The program can be used to inspect the graphical representation of the output signals of the simulators and it can be used to edit input signals that are described in the SLS command file format (see: User's Manual). Moreover, from *simeye2* simulations can be run by starting either the *sls*, *spice*, *pstar* or *plato* simulator (Note: *spice* via the script *nspice* (1ICD)).

> The following program argument may be specified:

> *cell*   This name specifies the name of the cell for which simulation signals are displayed and/or edited.

> The corresponding file from which the signals are read or to which the signals are written has a name that is equal to the cell name with an extension ".res" for sls digital (output) signals, an extension ".plt" for sls analog or waveform (output) signals, an extension ".ana" for spice (output) signals, an extension ".list" for pstar (output) signals, an extension ".sig" for plato (output) signals, and an extension ".cmd" for command file (input) signals.

> Note that the *spice*, *pstar* and *plato* output file should contain the signals of a transient analysis in tabular format. When running the program *spice* directly this is achieved by using the card "**.print tran** ..." in the spice input file. When running *nspice*, this is achieved by using sls "**plot** ..." commands in the command file. When running the program *pstar* directly this is achieved by using "**print:** ..." commands in the pstar input file.

COMMANDS

Commands that can interactively be given are:

CELL   This field specifies the name of the cell to which the signals belong. The cell name is given by the program argument (see above) or it is specified by the user while running the program. To edit it, use the left arrow, right arrow and <Backspace> and/or <Delete> keys of the keyboard. To load the signals of the specified cell, click the read button.

SLS-LOGIC SLS-TIMING SPICE/PSTAR/PLATO

These buttons define the type of simulation. When clicking the run button, an sls simulation at level 1 or 2 is performed when SLS-LOGIC is active, an sls simulation at level 3 is performed when SLS-TIMING is active, and a spice, pstar or plato simulation is performed when SPICE, PSTAR or PLATO is active. Note that previous simulation results of PLATO are not deleted by *simeye2*. When clicking the read button, sls signals are read when SLS-LOGIC or SLS-TIMING is active, and spice, pstar or plato signals are read when SPICE, PSTAR or PLATO is active. For PLATO only the signal names with extension '.v' (voltages) are read.

A/D A/D A

These three buttons, that are positioned immediately to the right of the simulation-type buttons, specify the type of signal representation that is used. For SLS-LOGIC and SLS-TIMING these buttons can have a value "A" (for analog) or "D" (for digital). For SPICE, PSTAR and PLATO the value of the button is always "A". Click on the "A" to switch from SPICE to PSTAR and from PSTAR to PLATO and visa versa.

READ   Load a new signal file. Normally, current signals are cleared from the program. However, if the <Shift> key is held down while pressing the read button the new signals will be appended to the current signals.

RUN   Perform a simulation by using either the *sls*, *spice*, *pstar* or *plato* simulator. For all simulators *simeye2* uses the command file "*cell*.cmd". The *spice* simulator is run by calling the program *nspice*. The *pstar* simulator is (default) called by the *npstar* shell-script. The *plato* simulator is (default) called by the *nplato* shell-script. If any error message occurs it will automatically be displayed by *simeye2*. When the *sls* simulator is run, also intermediate simulation results are displayed. At the end, in all cases, the final simulation results are displayed. A simulation may be aborted by typing <control>C in the simeye2 window (place the cursor in the simeye2 window and type the character 'c' while holding the <Ctrl> key down).

FULL   Draw all present signals from the beginning of the simulation time till the end of the simulation time.

REDRAW

Redraw the current window.

IN   Zoom in on the current window. When the <Shift> key is held down and the window contains only one analog signal, also a zoom-in on the y-axis of the signal is performed.

OUT   Zoom out on the current window.

LEFT, RIGHT, UP AND DOWN ARROW

Move the current window leftwards, rightwards, upwards or downwards respectively.

S   Save the current window settings (i.e. start-time, stop-time and the names of the signals that are displayed) in a file (The name of this file is specified in the configuration file).

L   Load the current window settings (i.e. start-time, stop-time and the names of the signals that are displayed) from a file (The name of this file is specified in the configuration file).

MOVE   With this command the order of the signals can be changed or one signal can be placed over another signal in order to compare them. First the signal that is moved or that is placed over another signal is selected, and then the new position of the signal is selected. To place the selected signal over another signal one should hold the <Shift> key down while selecting the new

position. To remove a signal that is placed over another signal, initially select the signal with the <Shift> key held down.

PRINT   Generate a hardcopy of the current screen.

VALUE   Move a vertical scanline over the display window, according to the position of the pointer, and show the value of the corresponding x-position (and y-position). When clicking the middle (or right) button of the mouse, one may toggle between displaying in the right margin (if the number of signals is not too large) the y-values of the signals at the selected x-position. To store the value of a particular position, click the left button of the mouse. This position will then be subtracted from the next positions of the scanline, so as to measure delay times and/or voltage differences.

QUIT   Quit the program.

INPUT   Read the command file "*cell*.cmd" and enable the edit menu. If the command file "*cell*.cmd" does not exist, *simeye2* shall try to read "simeye.def.d", the default command file. First, it tries to read this file from the current directory and second it tries to read this file from the process directory.

The following commands are part of the edit menu:

GRID   Specify the smallest unit for the x-axis (= time axis).

NEW   Create a new signal. The user has to enter the name of the new signal, and the signal will be placed at the bottom of the window.

DELETE   Delete one or more signals by selecting them with the cursor.

CLEAR   Delete all signals.

COPY   Copy the signal description from one signal to another signal.

EDIT   Edit a particular signal by inserting a new logical level for a particular time interval (t1, t2). The signal description may eventually already be defined for this interval. The insertion of the new logical level is done in two steps: During the first click of the mouse the signal, the new logical level and t1 are selected. During the second click t2 is selected and the new signal description is drawn. To insert an interval that has a free-state one should hold the <Shift> key pressed down while making the first selection.

YANK   Store a (part of a) signal description in the buffer.

PUT   Insert a copy of the signal description that is in the buffer onto a particular position. The user has to select the signal to which the contents of the buffer is added and the time from which on the new signal description is valid. The new signal description may (partly) override the existing signal description for the selected signal. Furthermore, the user is asked to type how may periods of the stored signal part are added. If a value -1 is specified, the selected signal will become a periodical signal and (seemingly) an infinite number of periods are added. In that case, when the simulation end time is enlarged, the signal description will automatically be extended according to the periodicity of the signal description. A periodical signal is indicated by means of a tilde (˜) immediately to the right of the signal description.

SPEED   Speed up the signals by some factor. If a value < 1 is specified, the signals will be slowed down.

T_END   Update the end time of the input signals (= end time of simulation).

READY   Disable the edit menu and update the command file "*cell*.cmd" after a confirmation has been given.

COMMAND FILE

For simulation, *simeye2* uses a command file "*cell*.cmd". This file should contain a description of the input signals, values for the simulation control variables, and a listing of the terminals for which output should be generated. This is explained in more detail in the users manuals of the *sls* simulator and the *spice* simulator.

The "**set**" commands in the command file may be edited by enabling the edit menu of *simeye2*. The new signal descriptions will then be written back to the command file when updating the command file. However, when the command file contains a set command that is followed by the keyword "**no_edit**" between comment signs, the signal description of this set command can not be modified. Additionally, when the command file contains on a separate line between comment signs the keyword "**auto_set**", *simeye2* will automatically turn-on or turn-off these 'non-editable' "**set**" commands according to whether or not the node the command refers to is part of the terminal list of the cell that is simulated. This feature may for example be useful to specify default signals for one or more input terminals.

When the command file contains on a separate line between comment signs the keyword "**auto_print**" ("**auto_plot**"), *simeye2* will automatically add a print (plot) statement to the command file for each terminal of the cell that is simulated. The new print (plot) commands will have a keyword "**auto**" between comment signs following the keyword "**print**" ("**plot**").

An example of a default command file that can be used for *sls* and *spice* simulations is:

```
/* auto_set */
set /* no_edit */ vdd = h*~
set /* no_edit */ vss = l*~
set /* no_edit */ phi1 = (l*110 h*80 l*10)*~
set /* no_edit */ phi2 = (l*10 h*80 l*110)*~
option sigunit = 1n
option outacc = 10p
option simperiod = 4000
option level = 3
/*
*%
tstep 0.1n
trise 0.5n
tfall 0.5n
*%
*%
*/
/* auto_print */
/* auto_plot */
```

CONFIGURATION FILE

At start-up of the program, *simeye2* will read some information from a configuration file called ".simeyerc". First, it tries to find the file in the current directory. Second, it tries to open the file in the home directory of the user. Thirdly, it looks for this file in process directory. The configuration file may contain the following keywords, followed by a specification on the same line if the keyword ends with ':';

SLS: Specifies the command for running the *sls* simulator.

SLS_LOGIC_LEVEL:
         Specifies the level of simulation when "sls-logic" is selected (use 1 or 2).

SLS_LOGIC_SIGNAL:
         Specifies the default signal representation for sls-logic simulations (use A or D).

SLS_TIMING_SIGNAL:
         Specifies the default signal representation for sls-timing simulations (use A or D).

SPICE: Specifies the command for running the *spice* simulator (use *nspice* or a derivative of it).

PSTAR: Specifies the command for running the *pstar* simulator (use *npstar* or a derivative of it).

PLATO: Specifies the command for running the *plato* simulator (use *nplato* or a derivative of it).

XDUMP_FILE:
         Specifies the name of the X Windows dump file that is generated when the print button is clicked.

PRINT: Specifies the command that is executed when the print button is clicked in order to process the X Windows dump file (e.g. to convert the window dump to PostScript and to send the output to a laser-printer).

PRINT_LABEL:
Specifies an optional label that is placed in upper left corner of the display window when an X Windows dump is generated.

SETTINGS_FILE:
Specifies the name of the file in (from) which the window settings are stored (loaded) when using the "S" button ("L" button or **-L** option).

DETAIL_ZOOM_ON
If this keyword is specified in the configuration file, the zoom-in function is also defined for the y-axis of a single analog signal. Thus, you do not need to hold the <Shift> key down (see command IN).

TRY_NON_CAPITAL_ON
If this keyword is specified in the configuration file and simeye2 fails to open a command file that starts with a capital letter, the program will try to open the same command file but now starting with a non-capital letter. If this succeeds and when performing a simulation, simeye2 will first run the program *arrexp* on a copy of the command file to expand one dimensional array node names into single node names (e.g. a[1..3] is converted into a_1_, a_2_, a_3_).

In the above specifications, the string '**$cell**' may be used to refer to the current cell name, '**$file**' may be used to refer to the current file on screen, '**$date**' and '**$time**' may be used to refer to the current date and time, and '**$user**' may be used to refer to your login name.

Example of a configuration file (default values are shown):

```
SLS: sls $cell $cell.cmd
SLS_LOGIC_LEVEL: 2
SPICE: nspice $cell $cell.cmd
PSTAR: npstar $cell $cell.cmd
PLATO: nplato $cell $cell.cmd
XDUMP_FILE: simeye.wd
PRINT: xpr -device ps -output $cell.ps (continued)
    simeye.wd; lp $cell.ps; rm simeye.wd $cell.ps
PRINT_LABEL: $user  $file  $date  $time
SETTINGS_FILE: simeye.set
```

AUTHOR
A.J. van Genderen, S. de Graaf

FILES

`.simeyerc`	(default) configuration file
`HOME/.simeyerc`	(altern.) configuration file
`ICDPATH/share/lib/process/`*process*`/.simeyerc`	
	(altern.) configuration file
`ICDPATH/share/lib/process/`*process*`/.simeyerc`	
	(altern.) configuration file
cell`.res`	(default) input file (SLS logic results)
cell`.plt`	(opt.) input file (SLS analog results)
cell`.ana`	(opt.) input file (SPICE analog results)
cell`.list`	(opt.) input file (PSTAR analog results)
cell`.sig.n`	(opt.) input file (PLATO signal names)

cell`.sig`	(opt.) binary input file (PLATO analog results)
cell`.cmd`	(opt.) command file
`*.cxx`	temporary files
`simeye.def.d`	(default) template for command file
`ICDPATH/share/lib/process/`*process*`/simeye.def.d`	
	(altern.) template for command file
`ICDPATH/share/lib/process/`*process*`/simeye.def.d`	
	(altern.) template for command file
`simeye.set`	(default) window-settings file
`simeye.wd`	(default) X windows dump file
`sim.diag`	simulation diagnostics
`pr.diag`	print diagnostics

X DEFAULTS

Some examples of ".Xdefaults" values are:

simeye∗.borderColor:	white
simeye∗commands∗.foreground:	white
simeye∗commands∗.background:	black
simeye∗canvas.foreground:	white
simeye∗canvas.background:	black
simeye.high_lowColor:	cyan
simeye.undefinedColor:	yellow
simeye.voltsColor:	green
simeye.pointerColor:	red

SEE ALSO

simeye(1ICD), sls(1ICD), spice(1ICD), nspice(1ICD),
SLS: Switch-Level Simulator User's Manual,
User's Guide for SPICE, Univ. of California at Berkeley,
Pstar User Guide & Reference Manual, Philips ED&T,
Plato Users Guide, M.T. van Stiphout, TU-Eindhoven (http://www.es.ele.tue.nl/papers/PWL.html),
X Window System Documentation.