HOW THE
NSPICE PROGRAM
WORKS

S. de Graaf

Circuits and Systems Group
Faculty of Electrical Engineering,
Mathematics and Computer Science
Delft University of Technology
The Netherlands

1. INTRODUCTION

This document describes how the *nspice* utility program works in combination with the *spice* simulator.

The program *nspice* is in real an unix shell-script and consists out of two programs. A preprocessor program *nspice_pp*, which prepares the input for the *spice* simulator program. And second, a back-substitute program *nspice_bs*, which changes the simulator output ".ana" file (the signal node-numbers into node-names).

But first of all, the *nspice* program calls the *xspice* program to extract a spice netlist (".spc" file) of the cell out of the design database. The following figure gives an overview.
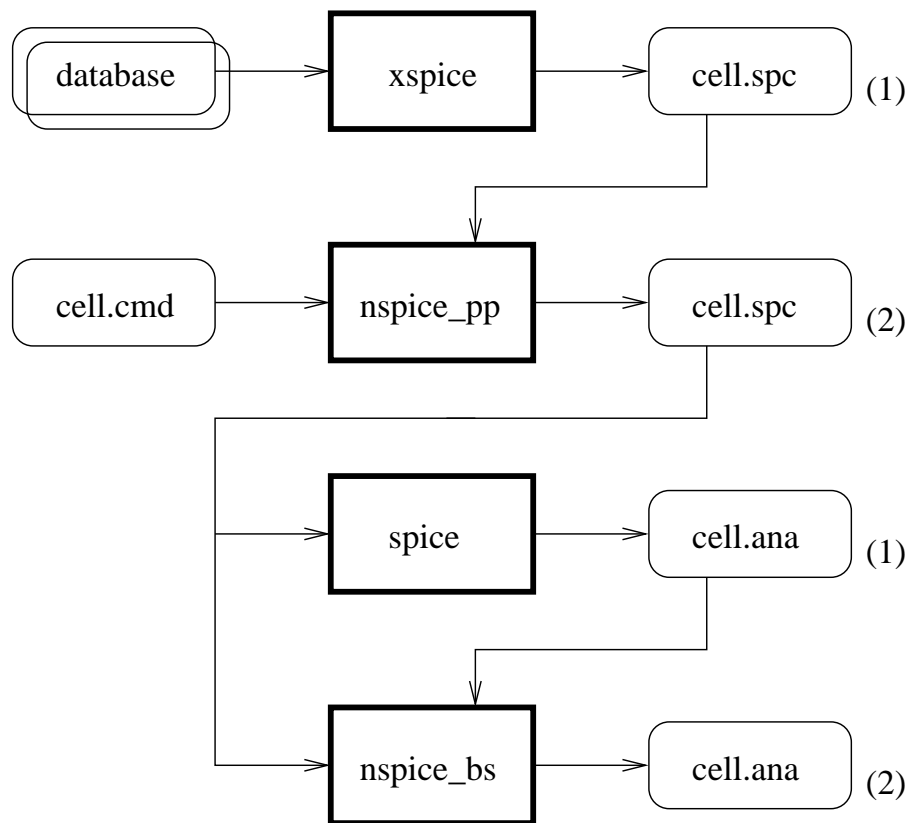
2. NSPICE_PP

The *nspice_pp* program consists out of the following source modules:

main_pp.c	define.h
readCir.c	type.h
readCom.c	extern.h
res.c	
cmd_l.l	
cmd_y.y	

The include (".h") files and "readCir.c" file are also used by *nspice_bs*.

The program contains a lexical analyser and parser to read the ".cmd" file. For detailed information consult the manual page of *nspice*. The ".cmd" file contains *spice* commands, which are added to the spice netlist in the ".spc" file.

First the command file is parsed by routine readCom to make a list of all signals, which are specified with "set" commands. And it makes also a list of all found "plot" commands. It also reads a number of "option" commands. Of which "option sigunit" is most important, this option is used to set the time-scaling. With "option simperiod" can the simulation end-time be set. If not specified, the end-time is calculated from the found "set" commands. Note that the end-less simuli cannot be used for it. The "option simlevel=3" is needed to generate "**.print** ..." cards and a "**.tran** ..." card. The level must be '**3**' to get them.

The spice section of the command file is divided in two parts. The parts begin and are separated from each other with the '*%' marks. The whole spice section must be a comment for the sls parser. Thus, this section is not read with routine readCom.
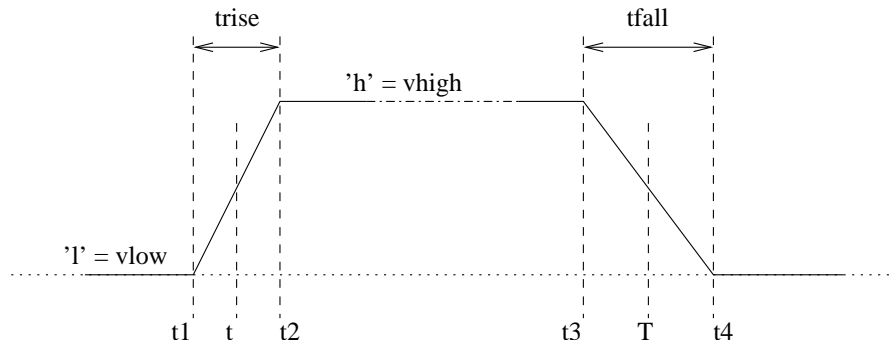
Second, routine readCir is used to read the name-list of the cell netlist, which specifies the relation between node-names and node-numbers. The program *xspice* should generate this name-list after the cell netlist, else the *nspice_pp* program cannot work. The format of this name-list must be:

```
* namelist cellname
*
*    nodenr name ...
```

Third, routine writeSpiceCom is used to write to "stdout" the spice commands which must be added to the ".spc" file. This routine shall first read the ".cmd" file for the spice section. It reads in first part the keywords, which specify "tstep", "trise", "tfall", etc. The value "tstep" must be specified to generate the a valid "**.tran** ..." card. Also, when "trise" and/or "tfall" are missing, the value of "tstep" is used instead. You can also specify a "vhigh" voltage, to use another value then 5 volt.

Thus, the sls "set" commands are converted to spice "pwl" stimuli commands. But only for those signal nodes, for which the node number is not equal to zero (the ground node).

Each transition from logic 'l' to 'h' at a time 't' is changed into two time points. The first time point 't1' is equal to (t - trise / 2) and the second time point 't2' is equal to (t + trise / 2). Thus, each transition from logic 'h' to 'l' at a time 'T' is changed into (T - tfall / 2) and (T + tfall / 2). See the following figure.

Of coarse the values of "tstep", "trise" and "tfall" must be small enough in accordance with the specified "option sigunit" value.

3. NSPICE_BS

The *nspice_bs* program consists out of the following source modules:

main_bs.c	define.h
readCir.c	type.h
	extern.h

The include (".h") files and "readCir.c" file are also used by *nspice_pp*.

First, routine readCir is used to read the name-list of the cell netlist (from the ".spc" file), which specifies the relation between node-names and node-numbers.

Second, routine backsubSpc is used to read the ".ana" spice output file. First, it scans the ".ana" file, to see if there are "*error*" or "*ERROR*" messages. If found, it print these error messages and exits the program.

When there are no errors, the program shall start to back-substitute the node numbers to node names. The node names can be found in the name-list. For spice3 output, the node numbers can be found on "Index time" lines. Between "v(..)" are these numbers found and changed into names. Everything is copied to an ".axa" temporary file.

If nothing goes wrong, (a) at least one "Index time" line found, and (b) all node numbers are found in the name-list. In that case, the ".axa" file is moved to the ".ana" file and overwrites the old ".ana" file.