

# Ngspice batch and distributed analysis and data postprocessing

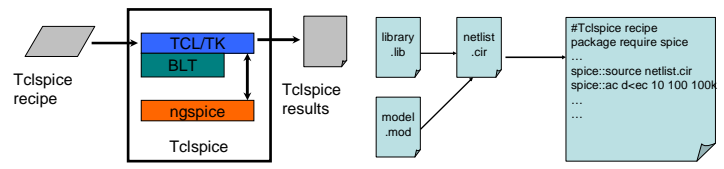
Alberto Lucchini<sup>1</sup>, Paolo Nenzi<sup>2</sup>, Francesco Nizzi<sup>3</sup>, Mauro Olivieri<sup>4</sup> and Lionel Sainte Cluque<sup>5</sup>

## TCLSPICE integration into ngspice

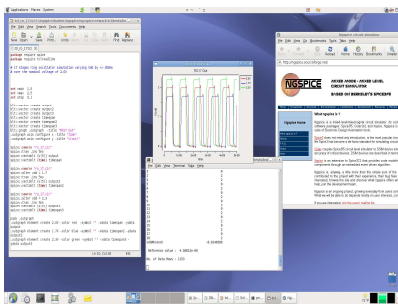
Tclspice is the binding of spice simulator in a shared library. Spice commands can be called in a TCL/TK shell or script. It started as an independent project by Stefan Jones and now is chaired by Lionel Sainte Cluque with the aim of integrate it with ngspice. Tclspice has many advantages:

- **Electronic designers often know TCL:** it is used in many tools. For example ngspice can do post synthesis simulation of digital ASICs using the TCL shell of the digital tool.
- **The TCL scripts are portable on any OS** *tclspice can be compiled on:* For example a Monte Carlo simulator written for tclspice can be used on any computer tclspice is installed on.
- **Tclspice addresses problems regular spice does not:** simulate a circuit and export data in an OpenOffice spreadsheet. Optimize component values, or build a web server on which the visitor can upload a netlist and get an HTML page with simulation output.

### TCLspice structure

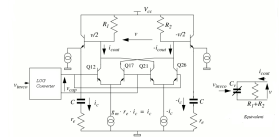


**Fig1. Tclspice structure:** the user interacts with a TCL shell that embeds all the commands of ngspice. The recipe for tclspice embed the original spice netlist and all of its includes.



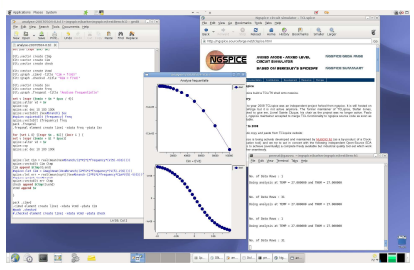
**Fig 2. Tclspice run example (17 stage ring oscillator from BSIM4 test suite):** The original netlist is loaded via the `spice::source` <netlist\_name> command and analyses a performed via simple TCL commands. Results are displayed superimposed on the same BLT plot. On the same script more than one netlist can be loaded and simulation outputs compared.

### TCLspice example: Active Voltage Controlled capacitor



**Fig3. Active variable capacitor:** The circuit on the left is a voltage controlled capacitor<sup>[1]</sup>. We want to extract the capacitance versus voltage.

- The netlist is loaded in the TCL/TK script and a series of AC analyses are performed varying the control voltage.
- The capacitance is extracted by ngspice command:  
`spice::let Cim = real(mean(Vex#branch/(2*Pi*i*frequency*(V(5)-V(6)))))`



**Fig 4. Voltage controlled capacitor simulation.**

### Conclusions:

Tclspice is quite well tested on Linux but there still are some bugs in compilation process and a Windows porting is under development. The compilation process is quite long due to the use of libtool. Tclspice sources are available on ngspice web site (a patched tarball can be downloaded on Sourceforge<sup>[4]</sup>). When all bugs will be fixed, sources will be merged. **There is a need for projects using tclspice like GUI, websites, Monte Carlo analysis, signal and power integrity.**

### References:

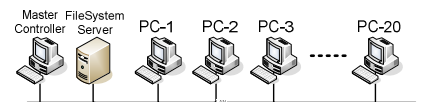
- [1] Marc KODRNJA - Etude des oscillateurs controles en tension pour les circuits a Frequence Intermediaire Analyse et simulation du bruit des oscillateurs, PHD Thesis (Institut National Polytechnique de Grenoble), 1992
- [2] A. G. M. Strollo, E. Napoli, D. De Caro, New Clock-Gating Techniques for Low-Power Flip-flops - Proceedings of ISLPED '00, 2000
- [3] Predictive Technology Model, <http://www.eas.asu.edu/~ptm/>
- [4] Ngspice web site, tclspice section, <http://ngspice.sourceforge.net/tclspice.html>

## Distributed parametric simulation with ngspice

Ngspice has been used as simulator in a distributed simulation system to evaluate and compare the performances of two different implementations of a PET-FF (positive edge triggered flip-flop).

- The evaluation space is defined by the *technology node*, *temperature* and *switching activity* axes. Simulations of both circuits have been run in different point of the evaluation space.
- Data generated by the simulation have been post-processed to generate reports in a spreadsheet compatible format.

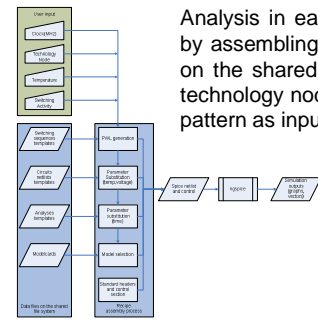
The huge design space forced the development of a distributed simulation strategy and the deployment of such system on the available computing resources, which consisted in networked GNU/Linux PCs with a shared file system. Simulation were run during nights to not interfere with daily use of the PCs.



**Fig 5. Available network infrastructure:** PCs are connected on a LAN and share file system on NFS.

The engine of the distributed simulations is made by portable bash shell scripts and uses of standard UNIX text tools only (awk, etc).

### Parametric analysis and netlist assembly



Analysis in each point of the evaluation space is performed by assembling the netlist from different parametric templates on the shared file system. The assembler shell script takes technology node, clock frequency, temperature and switching pattern as input and substitute parameters accordingly.

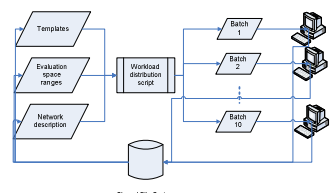
For each point in the space the data extracted are:

- Postscript plot of input and output waveforms.
- Power dissipated from the FFs.
- Propagation delay of the FFs.

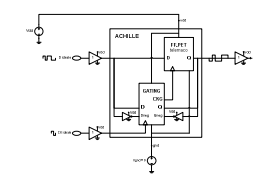
Data extraction is performed by scripts written in the spice control language. Simulations are performed by adding a standard `.control` section to the circuit netlist.

### Distribution of batches over the network

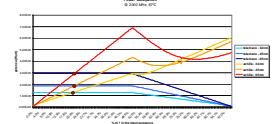
One PC in the network is assigned the role of Master Controller (MC). The MC spawns simulations to other PCs using secure shell and does data post-processing after all simulations completed. Each computer accesses a shared data directory on an NFS share to retrieve templates and write data. A tree of directories in the share holds the templates of switching sequences, model cards, circuit netlists and headers common to all the simulations.



**Fig 7. Batch distribution:** A script running on the master controller generates batches for each PC and send them over the network via SSH.



Simulation	Time (ns)	Power (mW)	Delay (ns)
1	10	0.5	1.5
2	20	1.0	1.8
3	30	1.5	2.1
4	40	2.0	2.4
5	50	2.5	2.7
6	60	3.0	3.0
7	70	3.5	3.3
8	80	4.0	3.6
9	90	4.5	3.9
10	100	5.0	4.2



**Fig 8. Circuits and simulation results:** The circuit on the left was simulated<sup>[2]</sup> against the non gated version to evaluate the performances of the clock gating. An example of the data output is shown (imported in a commercial spreadsheet).

### Conclusions:

Ngspice has been used to perform distributed parallel simulation and data extraction at a rate of 1600 simulations per night (on 20 PCs). The circuits under evaluation employed the BSIM4 model and the parameters for the 65, 42 and 35 nm technologies were retrieved from PTM<sup>[3]</sup>. This distribution computing environment is made only by shell scripts employing standard UNIX commands and can be rapidly deployed in any GNU/Linux networked without requiring additional software or dedicated hardware or infrastructure.

### Authors' Contact:

1. Alberto Lucchini <a.lucchini@gmail.com>.
2. Paolo Nenzi <p.nenzi@ieee.org>.
3. Francesco Nizzi <francesco.nizzi@gmail.com>.
4. Mauro Olivieri <olivieri@die.uniroma1.it>.
5. Lionel Sainte Cluque <saintel@users.sourceforge.net>.