

Introduction to simulation of electronic circuits with PSpice.

1. Introduction to simulation of electronic circuits.

The SPICE (*Simulation Program with Integrated Circuit Emphasis*) is a simulation program that allows computer-aided electronic circuit design and simulation, with a high degree of accuracy. SPICE was developed at the University of Berkeley during the 70's and has become the standard in electronic simulation programs. With the development of operating systems with graphical environment and the subsequent development of IDE's (*Integrated Development Environment*) many software companies have developed simulation programs with graphical editors, but with the same engine of simulation, the SPICE program.

This paper explores the foundations of electronic circuits simulation, i.e., the circuit editing, different simulation modes, and above all, the support tools for the analysis of results. Traditionally, after running the simulation, results were presented in a tabular form and were processed with a graphics package. With the development of software, these tools are managed under a single integrated development environment. In particular, this document will be developed with the aid of the CAD system of OrCAD.

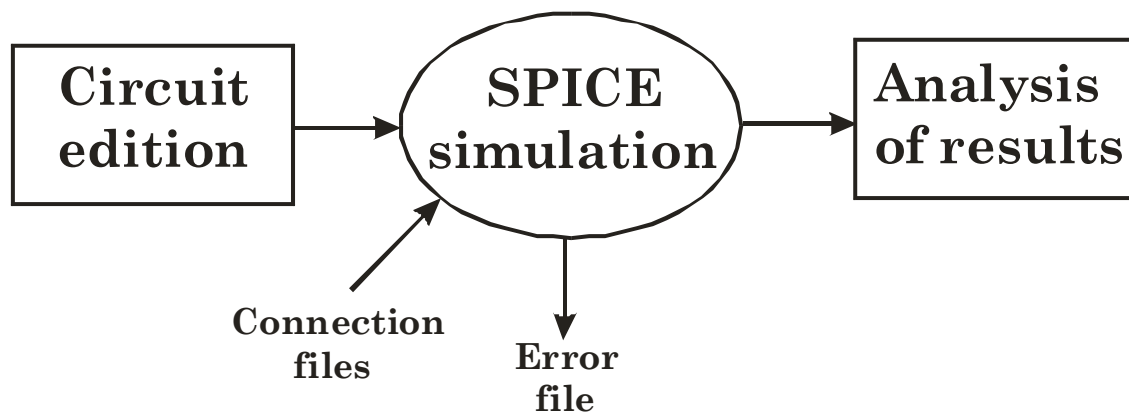


Fig. 1: Simulation process.

The process of circuit simulation consists of drawing the circuit and translating this scheme to a connection file (.CIR and .NET), which is understood by the SPICE simulation engine. Once the circuit is simulated, an output file (.DAT) is generated. This file is understood by many graphics packages. In the case of OrCAD SPICE, this whole process is centralized in the environment *OrCAD Design Manager*.

2. OrCAD Design Manager

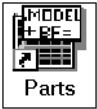
From this CAD system we can edit and simulate electronic circuits, and edit and manage libraries containing models of the devices used in these circuits. Consequently, this system centralizes all the simulation process. This CAD system contains four packages, plus an circuit editor:



Schematics: The schematic editor allows you to draw the circuits. From this package the simulation engine and the graphics package that displays the results can be accessed.



SPICE A/D: This package is the simulation engine, which is responsible of simulating the circuit and preparing the output data for the graphics package.



Model Editor: The model editor lets you change the physical characteristics of the devices used in electronic circuits, for example, the threshold voltage of a semiconductor diode, the width and length of the channel of a MOSFET transistor, etc.



Stimulus Editor: This package allows you to generate different types of input signals, giving the opportunity to view them while they are drawn. In the evaluation version we can only generate sinusoidal signals and clock signals for the application to digital circuits.

As can be seen in Figure 1, the left side of the *PSpice Design Manager* window displays the icons of each of the packages as well as a collection of files that are used during the simulation.

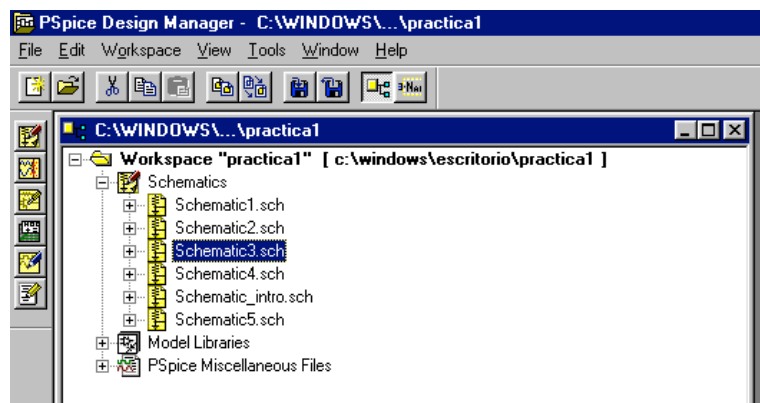


Fig 1.: General aspect of PSpice Design Manager

The contents of each file is identified by its extension:

.SCH: This file extension is created by SCHEMATICS where it stores the circuit diagram. It is the first file that is created.

.IND is the indexfile, and is created when the simulation is done. It makes reference to the libraries where are the elements that have changed some of its parameters.

.LIB is the library file, containing the internal parameters of the models of the elements. You can display it using any text editor.

.SLB is the library file where the data of the symbols of the elements used by the schematic editor are stored.

.VP is the library file where are the data of the cases of the elements of symbol library.

.NET is a file that is created just before the simulation. In this file the results of an electrical check are displayed. Therefore, the check of the circuit must be correct to be generated this file. Otherwise, a window containing the errors detected by the electrical check program is displayed.

.ALS is a file containing a list that identifies the nodes of the circuit with the terminals of the elements.

.CIR is a file that includes data on the type of analysis selected and its specific characteristics. It also refers to the file extensions .NET and .ALS and to the program PROBE.

.DAT is a file generated during the simulation that saves the results of the simulation in a form suitable for graphic interpretation by PROBE.

.OUT is a file generated during the simulation that saves the results of the simulation, i.e. how to interpret the input file .CIR. It lists the names and values of the parameters of the models of devices used in the circuit, the compilation time, the occurred errors (what type and were) and, finally, the operating point of all devices.

3. Schematics

Schematics is the package that allows the creation of electronic circuits and to simulate them. It is composed of five key elements:

- Devices.
- Device parameters.
- Cables for connecting components.
- Simulation settings.
- Graphical environment or PROBE.

Devices

Figure 2 shows several devices we will use in the labs. Among them, there are voltage sources, resistors, capacitors, a signal generator, semiconductor diodes, and MOSFET's transistors.

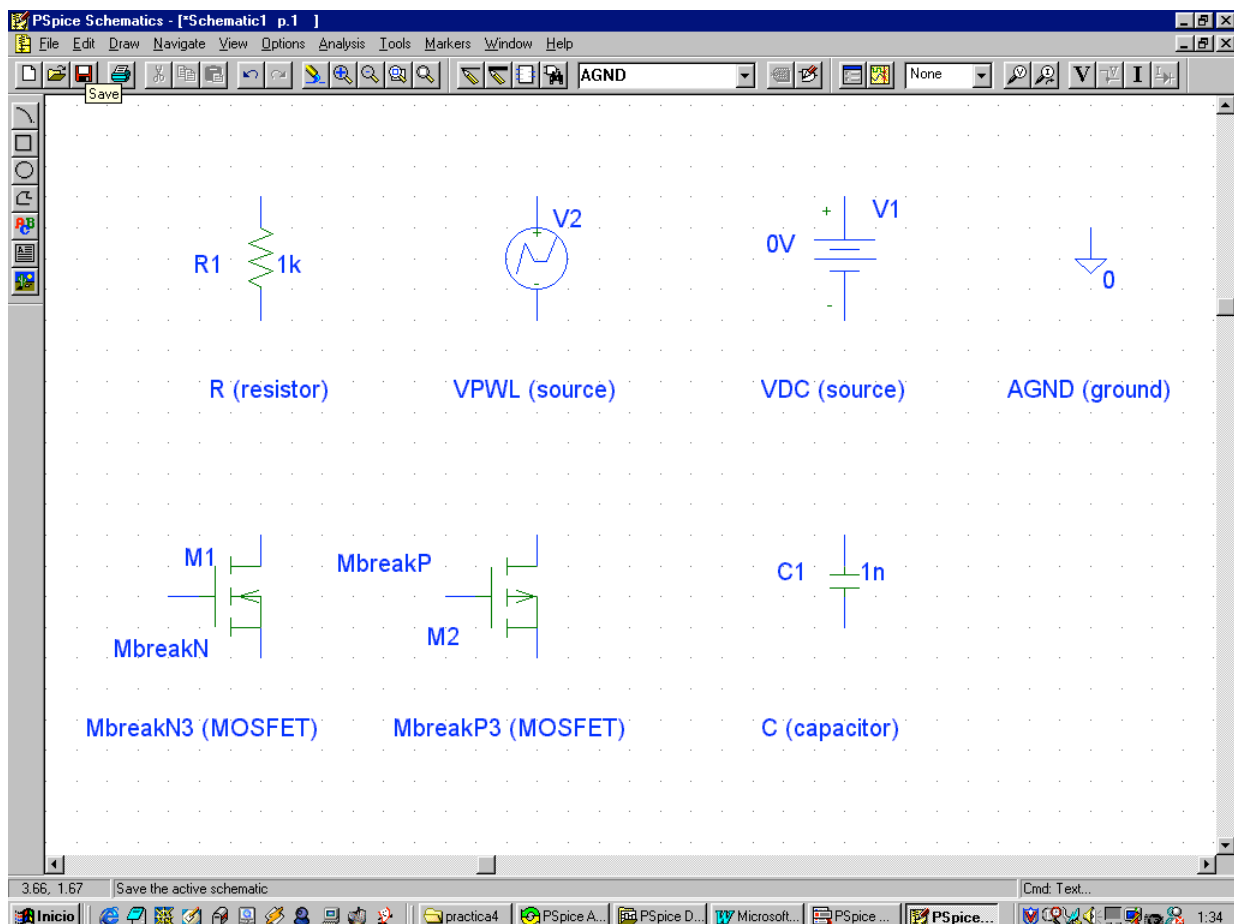


Fig. 2.: SPICE devices.

The access to these devices is as follows:

In the **DRAW** menu, access the library manager (**draw/ get new part**) and find a suitable device (see Figure 3A), employing a description string that can be seen between parenthesis in Figure 2.

Once you have drawn the schematic, the device appears in the cache (see Figure 3B), where can easily be imported without going back to the library manager.

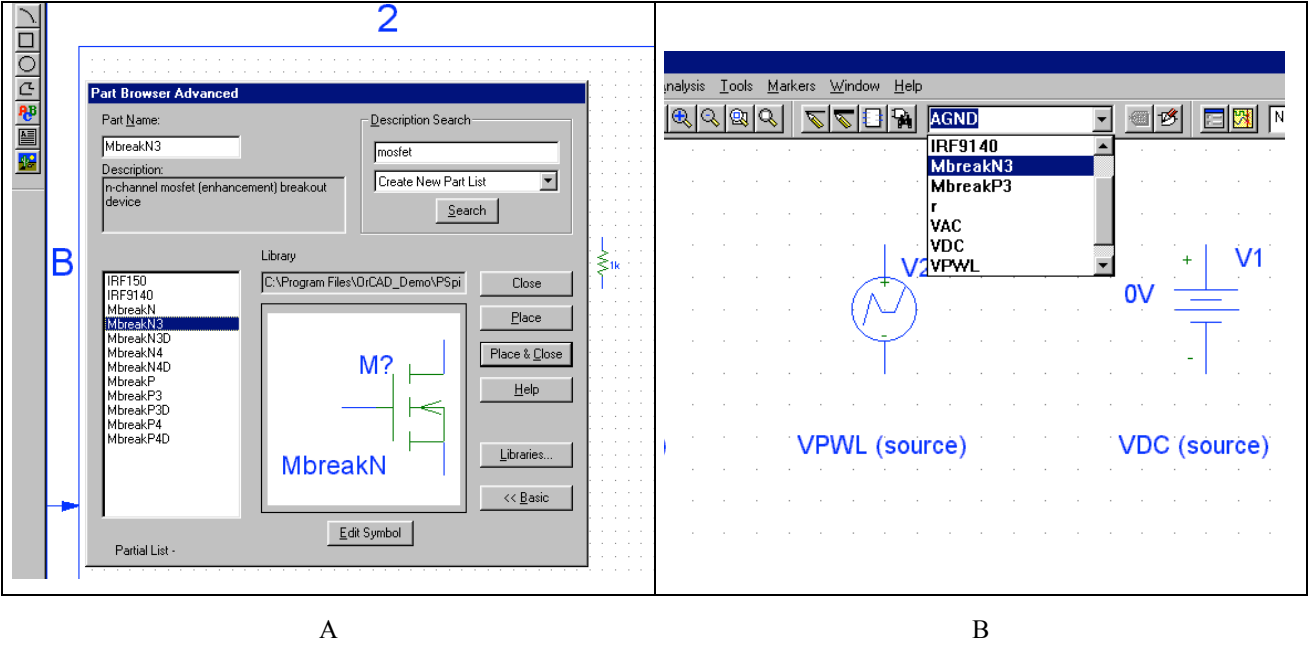
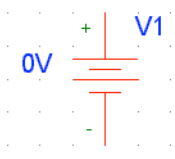
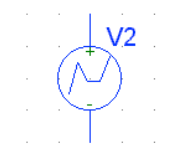
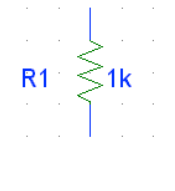


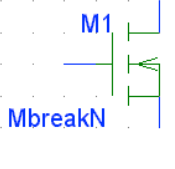
Fig.3: Mechanisms to search for devices.

All devices have parameters that can be modified (for example, the resistor value). There are two places to make this change depending on the type of parameter: by double clicking the component or from **edit menu/model/edit instance**, where we can see the parameters to be modified. Below are some examples:

Examples of device parameters:

	<p>Voltage source, battery, voltage generator.</p> <p>Parameters:</p> <p>DC, specifies the voltage between the positive and negative poles.</p>
	<p>Signal or function generator.</p> <p>Parameters:</p> <p>In pairs T1, V1, T2, V2 ...</p> <p>(Set Vn volts at the instant Tn seconds)</p>

	<p>Resistor:</p> <p>Parameters:</p> <p>VALUE specifies the value of resistance, for example, a 1kΩ resistor is specified as 1000 or 1k.</p>
-----------------------------------------------------------------------------------	---------------------------------------------------------------------------------------------------------------------------------------------

	<p>N channel MOSFET</p> <p>Parameters (EDIT/model/edit instance model ...).</p> <p>vto = 3, l = 10u, w = 20u, kp = 100u</p> <p>VTO: threshold voltage, l, w: length and width of the channel, kp: transconductance</p> $K = \frac{1}{2} \frac{W}{L} K_p$
-----------------------------------------------------------------------------------	-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

We can add suffixes to values of each parameter to refer to multiples and submultiples. For example, if we want to indicate that resistance has a value of 10000Ω, it can be specified as 10000 or 10k. Table 1 summarizes these multiples and submultiples:

Multiple	Suffix	Value
FEMTO	F	10E-15
PICO	P	10E-12
NANO	N	10E-9
MICRO	U	10E-6
MILI	M	10E-3
KILO	K	10E+3
MEGA	MEG	10E+6
GIGA	G	10E+9
TERA	T	10E+12

Table 1: Symbols for multiples and submultiples.

Linking components using cables.

For editing the cables that connect the devices you can press the button on the toolbar with the pencil icon, or press **Ctrl + W**.

Watch out for the cable connection! The cables have to be connected only at the end of the terminals of each device. To finish drawing cables, press the **ESC** key or the right mouse button.

Simulation settings.

The electronic circuits can be analysed in four ways: two different DC analysis (operating point and DC analysis), a transient analysis and a frequency response.

Operating point (OP): This analysis solves a circuit in DC, i.e. obtains the current in all branches, and the voltages at all nodes.

DC Analysis (DC): It is similar to the above analysis, but allows the circuit simulation by parameterizing the value of voltage sources, i.e., it performs the OP analysis for different values of voltage of one or two specific sources.

Transient analysis (TRANS): It allows you to calculate the evolution of the voltages and currents in one or more points on the circuit as a function of time.

Frequency analysis (AC): It calculates the response of the circuit to different frequencies. This analysis is performed on each point indicated.

To set the type of simulation, go to **Analysis-Setup** menu in **Schematics** program, as illustrated in Figure 4.

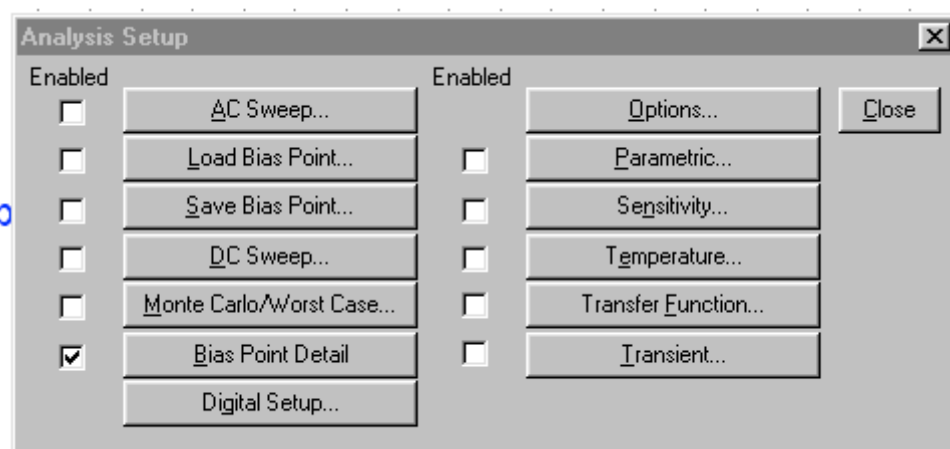


Fig.4: Setting the type of simulation.

Operating point (OP) simulation, by pressing the activation cell corresponding to **Bias Point Detail**. To view the values of voltages and currents in the circuit, the easiest way is to activate in the toolbar the V, I buttons. This will provide the voltage and current in all elements of the circuit. If you want to make a selective analysis of a subset of specific components, you can use the voltage probes (**voltage/level marker**) and current probes (**current marker**), placed next to the V, I buttons.

You can also go to the menu (**Analysis/Display Results on Schematic/Enable Voltage Display**) to see the voltage on all nodes, and to the menu (**Analysis/Display Results on Schematic /Enable Current Display**) to view the currents. Another alternative way to see the results of this type of simulation is to open from the window of **Schematics** the .OUT file (**analysis/ examine output**).

Remember that this only sets up the simulation and how to see the results. To see the final results is mandatory to simulate the circuit and, therefore go to the menu (**Analysis/Simulate**).

DC simulation, by pressing the corresponding activation cell of **DC Sweep** and the corresponding button, as shown in Figure 5.

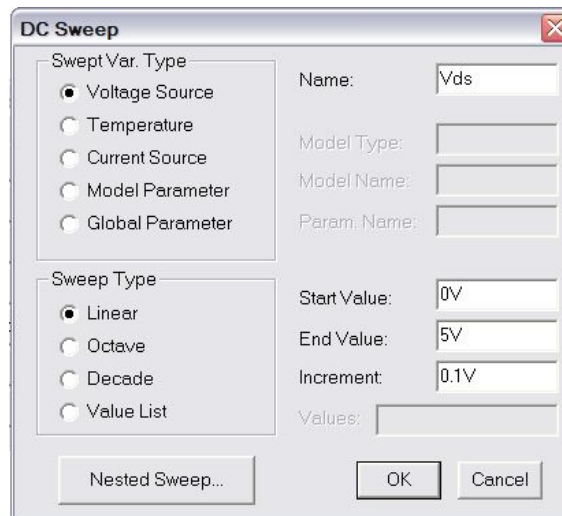


Fig.5: DC Sweep analysis.

It replaces the value of supply voltage specified in the cell (Vds), and makes the analysis of the working point for the specified voltage range between *start* and *end* value with the increasing value *increment*. By pressing **Nested Sweep** analysis, a chained analysis can be performed. When using this issue, it appears a window similar to the previous one, which specifies the name of a second source. (Caution! in this second window we must activate the cell **enable nested sweep**).

Instead of seeing the results on the circuit itself, it will appear the window of the machine simulation that enables to visualize the evolution of different variables, depending on the values of the voltages of each of the sources specified in DC analysis. The PROBE program is described briefly afterwards.

TRAN simulation, by clicking the activation cell that corresponds to *Transient*, and the corresponding button (Fig. 6).

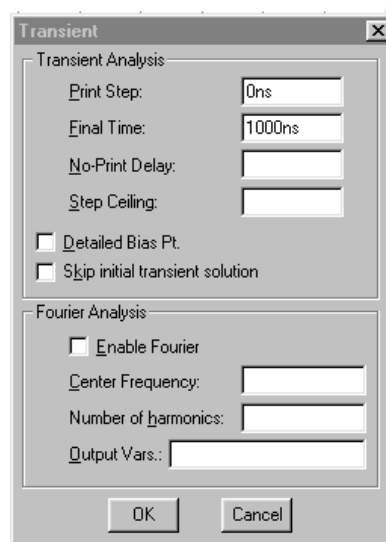


Fig.6: Transient analysis.

If there are voltage sources that vary with time, we must perform a temporal analysis, i.e., we have to calculate the voltage and current of all the nodes and branches as function of time. The most important parameter in this case, is how long will be the analysis. It is specified in the

Final Time cell. Normally the duration of the simulations in our labs will be the order of nanoseconds (ns) or microseconds (us).

4. PROBE Graphical representation environment. Running the Simulation.

Once you have configured the type of simulation, it is necessary to launch it. To do so, go to the menu **Analysis/Simulate**. After this, the simulation engine will start. This engine has a graphical representation of signals in a similar way as does an oscilloscope (see Figure 7). This window is divided into three blocks, the block where the signals are graphically displayed, a window with messages about the operation and the simulation results, and a third window showing the status of the simulation, i.e., the time elapsed, the voltage of the power supplies of the simulated circuit, etc.

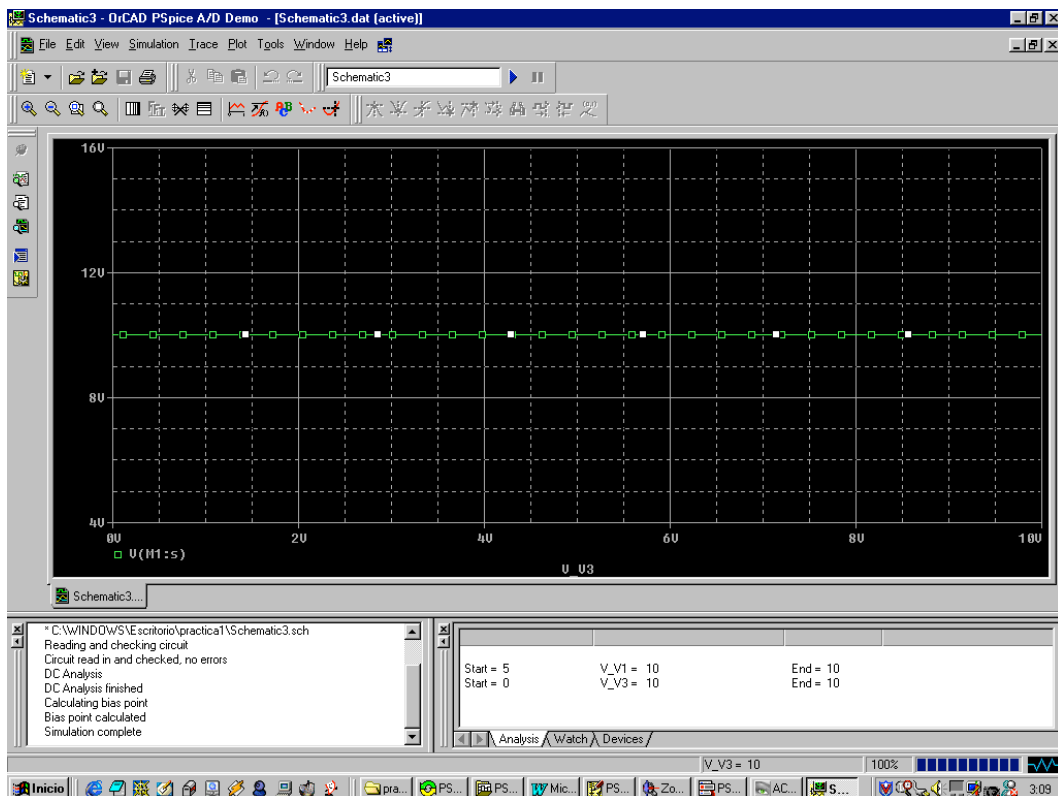


Fig.7: PROBE Graphical representation environment.

In the simulation window there are two important menus:

View/Output, that shows the output file (.OUT), i.e., a file with all the information concerning simulation. It is very helpful to analyse this output file if any error is produced during the simulation.

View/Simulation Results, that shows the window necessary to represent the signals.

Trace/Add Trace that lets you to choose the function to be represented in the **Simulation Result** window (see Figure 8).

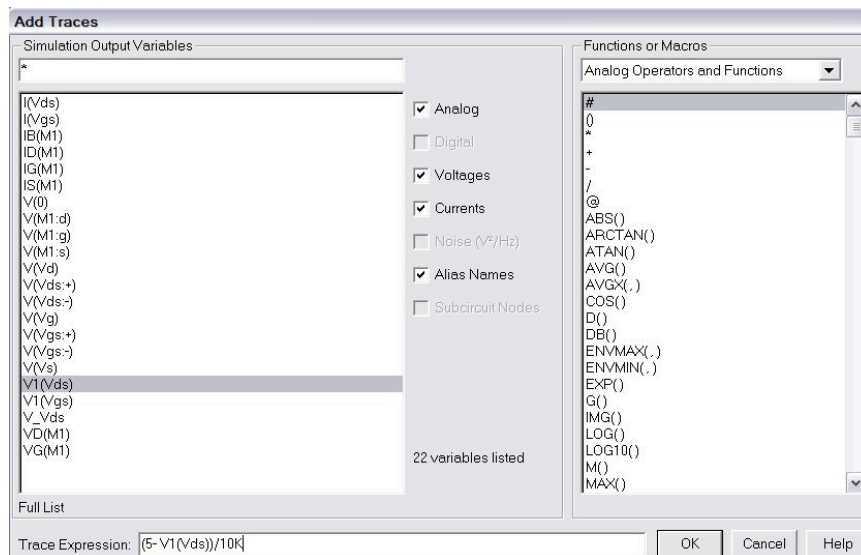


Fig.8: Election of the magnitude to plot, using Add Trace.

It also allows expressions that represent values dependent on voltage/ current values obtained during the simulation of the circuit.

Trace/Plot/Axis settings, that enables to set the independent variable, i.e. which variable is represented on the X axis. It can also be a complex expression, as the voltage difference between two nodes. To set the X-axis variable, press the menu option **Plot/Axis settings** and then click Axis Variable button (see Figure 9).

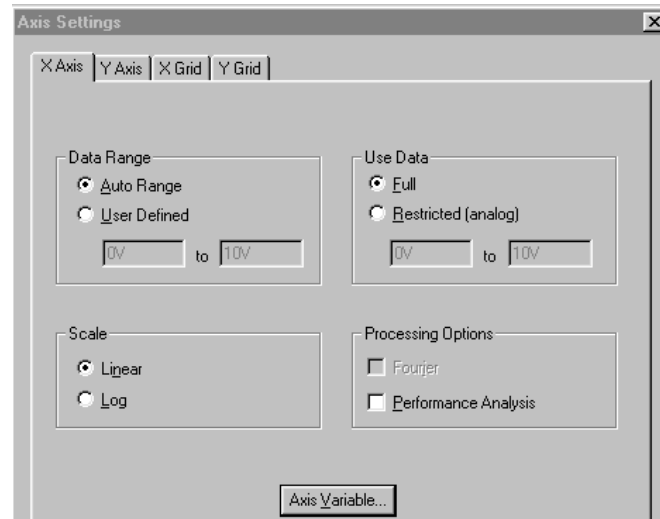


Fig.9: X axis variable specification.

On the window that is shown, we specify the variable or expression of the X axis, for example the voltage difference between drain and source in a MOSFET transistor (see Figure 10).

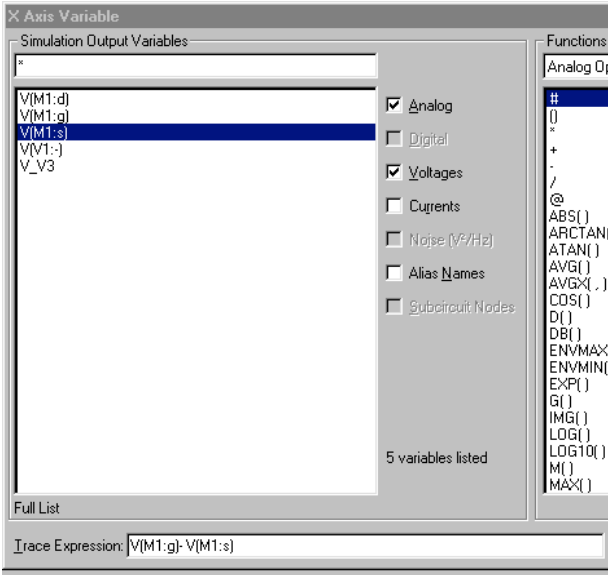


Fig.10: Example of specifying the X-axis variable.