Prologue: Some Features of CAPTURE and PROBE

Here is how to set up a project in CAPTURE and use several features of PROBE.

Introduction

Some terminology and grasp of project structure is needed before we can proceed. This document defines some terms and provides a few details about use and organization of files using CAPTURE. Then it describes how to plot circuit behavior as a function of a selected device or circuit parameter using PROBE.

To introduce some terminology, CAPTURE has three very useful functions: it is a file management tool, it is a drafting tool for circuit schematics, and it is a translation tool that converts the schematic to a list of components and connections readable by a simulation engine. The simulation engine of CAPTURE is PSPICE, and the results of simulation are accessible through text in the OUTPUT FILE and through graphics. Both are accessed from the graphical engine, which is PROBE. To learn the rudiments of CAPTURE, PSPICE and PROBE, please consult a handbook like that by Herniter, referenced at the end.

File management

Because it is "only" bookkeeping, file management may not receive the attention it requires. Good organization of a project is very helpful.

A two-node circuit design example

As an example, we will use simulation of a two-node circuit as shown in Figure 1.
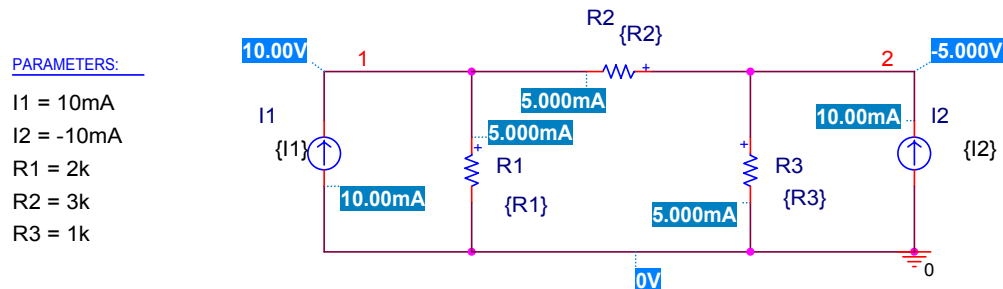

FIGURE 1

A simple two-node circuit (numbered node 1 and node 2) used to illustrate project management

In Figure 1 the value of resistor R1 is a variable R1, denoted as a variable by putting it in curly brackets as {R1}. The value of variable R1 is determined by the value set in the PARAMETERS box, namely R1=2k in this case. Other components are treated similarly. Also two DC current sources appear in Figure 1, with values I1 and I2.

PLACING FILES

When starting a new project using the menu system FILE/NEW/PROJECT, the menu of Figure 2 appears.
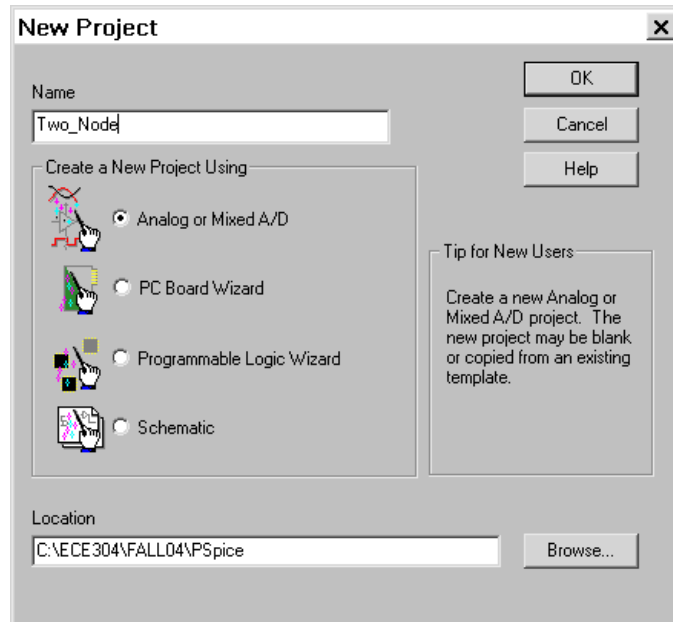

FIGURE 2
The NEW PROJECT menu for selecting the location for the new project

Notice the window at the bottom of this menu named LOCATION. By hitting the BROWSE button at the right of this window you can select where to put the new project file. The BROWSE button leads to the SELECT DIRECTORY menu shown in Figure 3 (see Herniter, p. 5).
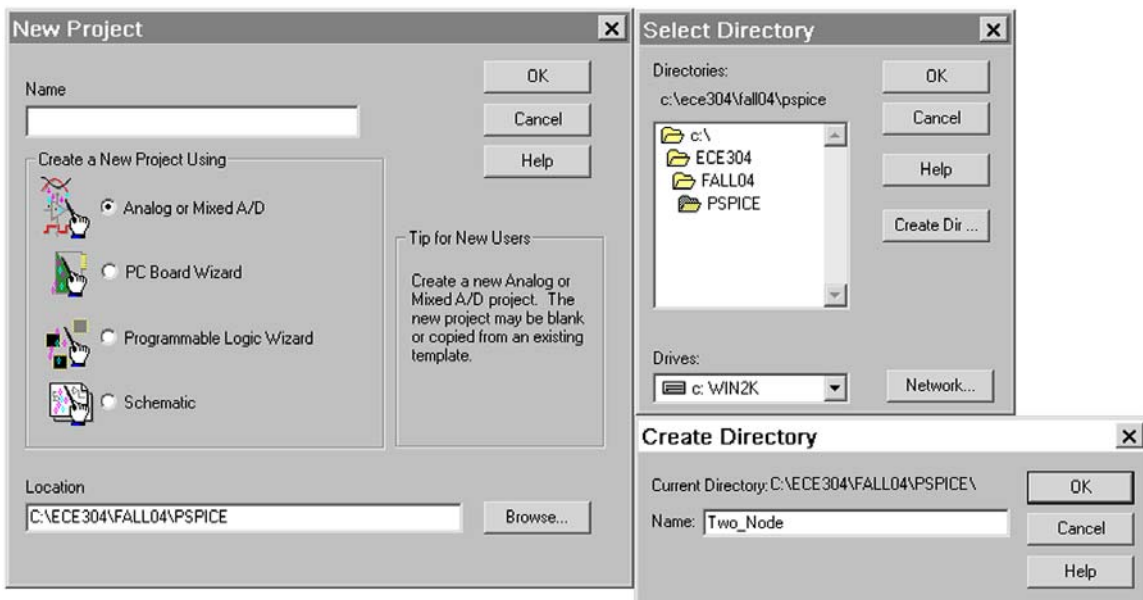

FIGURE 3
Creating a new folder for the project using the CREATE DIRECTORY button on the SELECT DIRECTORY menu.

> I strongly recommend that each project be placed in its individual folder with a recognizable name.

Following this procedure has the advantage that the dozens of files generated by this project are grouped together and not mixed up with other projects. As a result you can locate, copy, delete, or transfer the files for this project readily.

THE PROJECT MANAGER AND THE PROJECT HIERARCHY

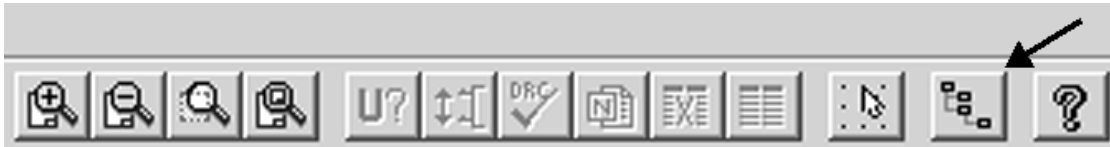Each project in CAPTURE has a PROJECT MANAGER accessed from the toolbar shown in Figure 4.

FIGURE 4
The arrow points to the icon for selecting the PROJECT MANAGER

This manager is more useful than might first appear. The PROJECT MANAGER makes possible the storage of parts that can be copied and pasted into active schematics. It makes possible the separation of different phases of the project, for example small-signal analysis can be separated from large signal, or one circuit variation can be separated from another. This hierarchic separation is more convenient than using completely separate projects.

The PROJECT MANAGER for the two-node design project is shown in Figure 5.
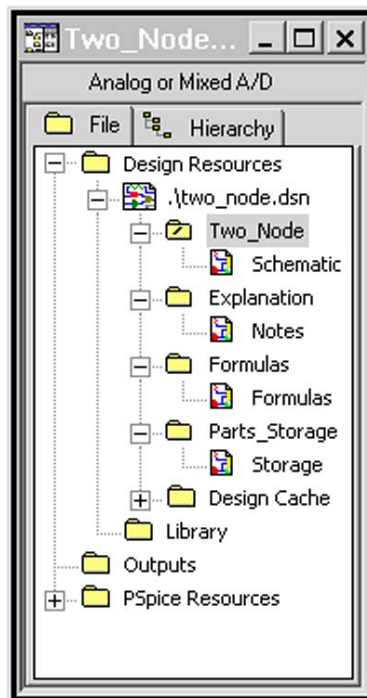

FIGURE 5
Project Manager in CAPTURE

From Figure 5 we see that the PROJECT **two_node.dsn** is broken into four branches named TWO_NODE, EXPLANATION, FORMULAS, and PARTS_STORAGE. Each branch is formed by highlighting **two_node.dsn**, right-clicking, and selecting NEW SCHEMATIC. Then the name of the

new schematic is highlighted, right clicked, and a NEW PAGE selected and named. For this example of a two-node circuit, the circuit schematic is contained in the branch TWO_NODE on the page named SCHEMATIC. There is another branch called EXPLANATION listed in Figure 5 and the page NOTES in this branch is illustrated in Figure 6. While not necessary to the operation of the project, this page is useful if you return to the project when memory has become dim, or if you hand the project to a colleague.
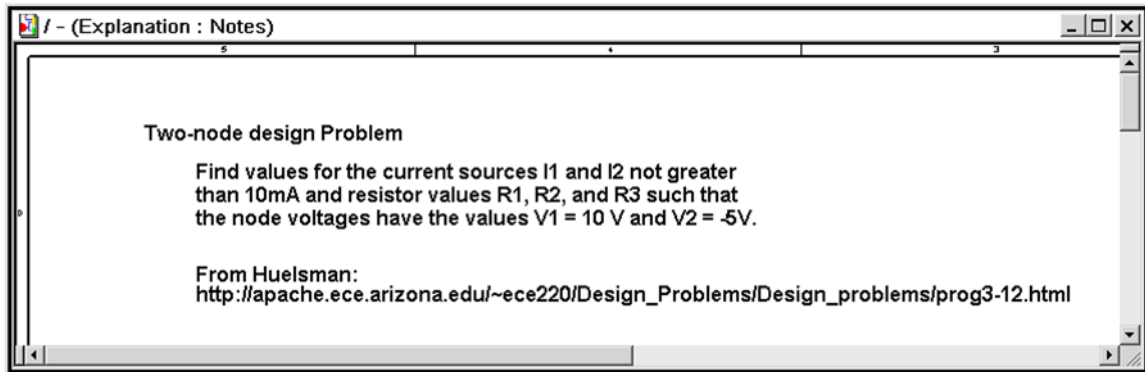

FIGURE 6
The NOTES page in the two-node project reminds the user what the PROJECT is about[1]

The branches created in the PROJECT MANAGER are an hierarchy of <u>independent branches</u>, which is to say, only one branch is simulated at a time, namely the one at the top of the listing in the PROJECT MANAGER. This feature is used to perform independent tasks. For example, the branch called FORMULAS uses the schematic shown in Figure 7
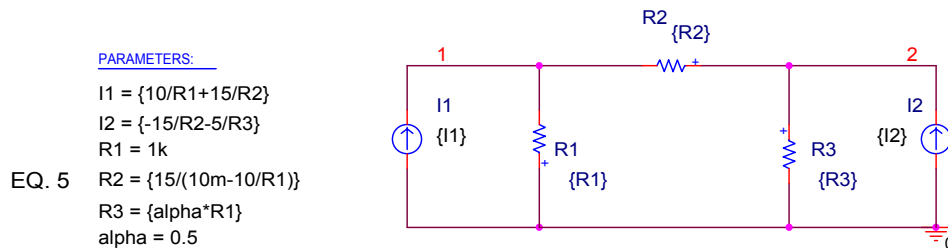

FIGURE 7
Schematic used in the FORMULAS branch of the hierarchy

This schematic uses a formula for I1, I2 and R2 instead of the numerical values of Figure 1. Using formulas can help us decide if we have a good model for how the circuit works. But a simple numerical simulation like that using Figure 1 is more straightforward, and is the ultimate test to see if the design specs are met. So we need both types of simulation.

Because it is a <u>separate</u> branch, the FORMULAS branch is <u>not run</u> when the branch called SCHEMATIC is run. In order to run the FORMULAS branch, we have to highlight DESIGN RESOURCES in the PROJECT MANAGER and use FILE/SAVE from the tool bar. Then we can highlight FORMULAS, right-click, and select MAKE ROOT, as shown in Figure 8. The result will be to move the FORMULAS branch to the top of the hierarchy, where it will execute, as shown in Figure 9 below.

---

[1] Formulas or pictures in this page can be introduced using PLACE/PICTURE to paste a bitmap file.
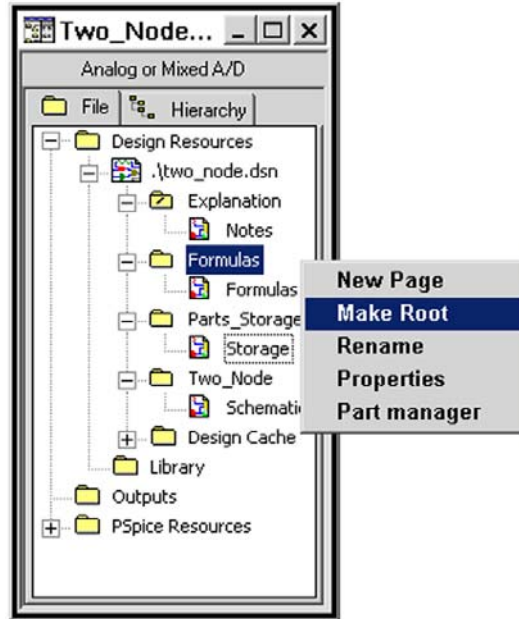
FIGURE 8
Moving branch FORMULAS to the top of the simulation hierarchy using MAKE ROOT
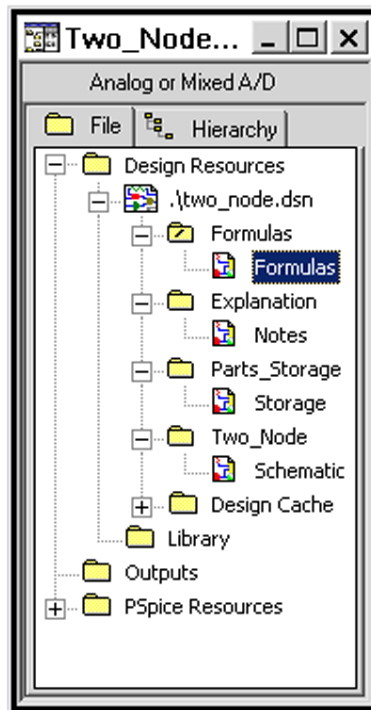
FIGURE 9
PROJECT MANAGER now shows that FORMULAS is at the top of the hierarchy, and so is the branch that will run

Likewise, we can make a branch called PARTS_STORAGE where we simply paste parts that sometimes we want and sometimes not, to use as a library of odds and ends that we can cut and paste into the schematic when it proves useful. For example, we might store a library of different devices that we use from time to time and do not want to set up over and over again. We also can store circuit fragments that are not functional by themselves and would lead to a run time error if

executed. When wanted, parts are copied from Parts Storage and pasted on the main Schematic. When no longer useful, they are deleted, but the copy in the library remains available. Other occasionally useful items for Parts Storage are voltage or current probes, evaluator circuits, and breakout parts.[2]

NAMING FILES

Although CAPTURE recognizes names with spaces, I recommend against it. If you zip the files for the project, or mail the files to another computer, often the spaces get removed. Then *inside* the files the file names have spaces, but their actual names no longer have spaces. Then the project will not run.

BACK-UP FILES

As with most software applications in WINDOWS, system malfunction is a common occurrence, and back up is a necessary annoying activity. With CAPTURE, these system failures take the form of freeze-up, or of failure to locate parts of a project, or automatic closure of a project just as it begins to open.[3] One never knows when the entire project will become inaccessible, so we have to back-up.
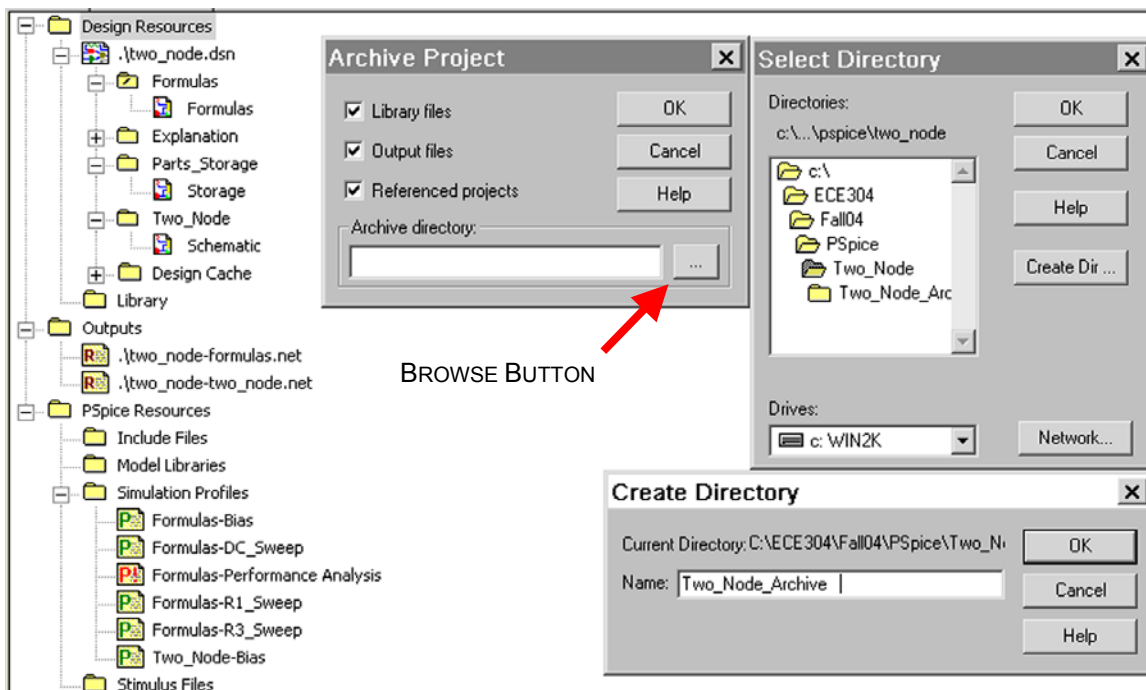

FIGURE 10
The sequence of menus for naming the back up file, named Two_Node_Archive in this example

To create a back up, the menu sequence is shown in Figure 10. CAPTURE has an ARCHIVE feature accessed by highlighting DESIGN RESOURCES in the PROJECT MANAGER and selecting FILE/ARCHIVE PROJECT. Selecting this option produces the ARCHIVE PROJECT menu in the center of Figure 10, which again has a BROWSE button, but identified only with an ellipsis "...". See Figure 10. Hitting this BROWSE button reveals the SELECT DIRECTORY menu at the right of

---

[2] Evaluator circuits are ancillary circuits used to "measure" circuit properties. Breakout parts are custom devices with parameters selected by you, such as bipolar devices with particular choices for Early voltage or for capacitances. They are discussed later.
[3] Often the failure to open is related to the PROBE files in the project. By deleting all files in the project that are not ordinarily saved by the ARCHIVE feature, sometimes the project can be made to open.

Figure 10, which has a CREATE DIRECTORY button. I recommend creating the archive in a subfolder of the original project folder. Once the file is named, check that it is *open*, which is *not* automatic. Then hit OK to obtain the ARCHIVE PROJECT menu. I leave the default checking of all boxes and hit OK again.
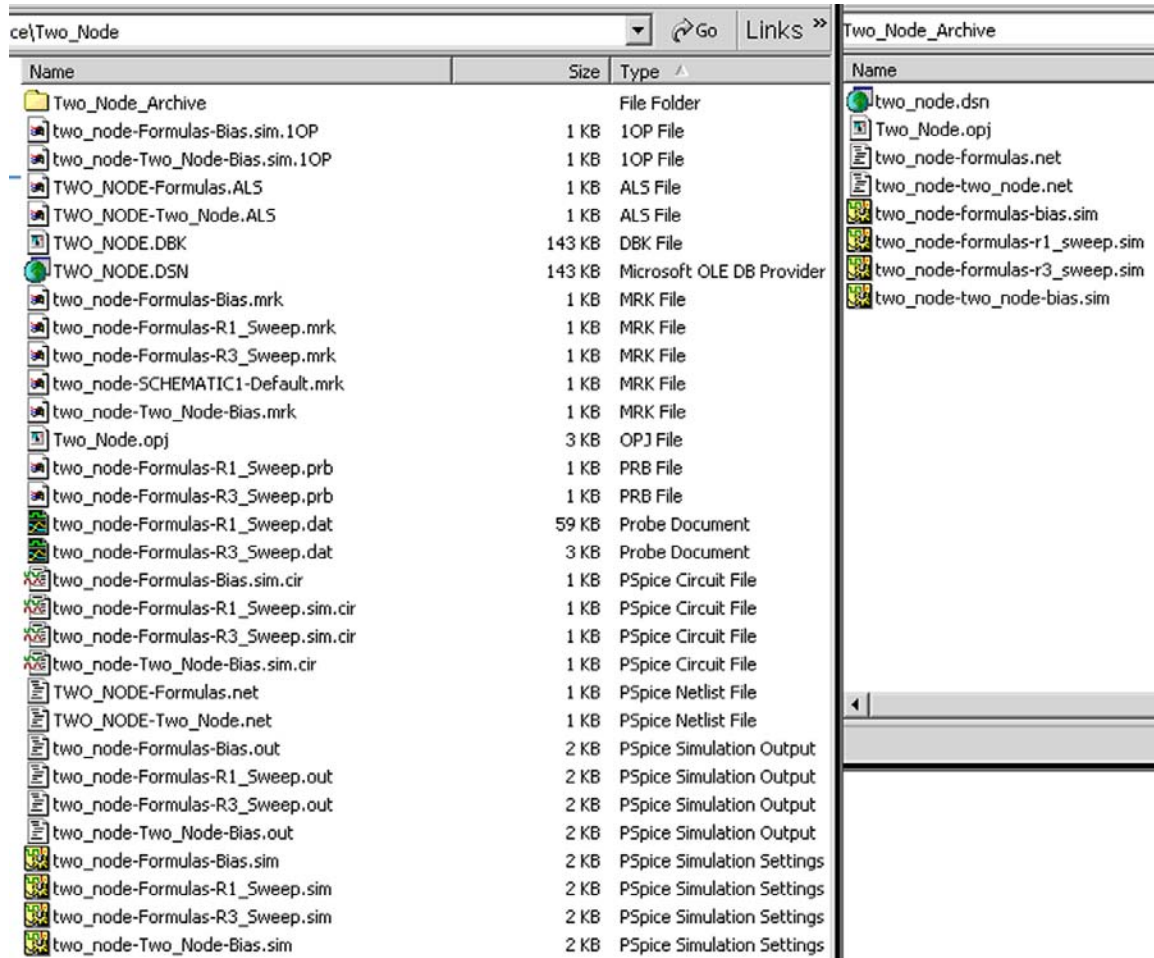
ce\Two_Node			Links »	Two_Node_Archive
Name	**Size**	**Type** △		**Name**
🗀 Two_Node_Archive		File Folder		🌀 two_node.dsn
📄 two_node-Formulas-Bias.sim.1OP	1 KB	1OP File		🔳 Two_Node.opj
📄 two_node-Two_Node-Bias.sim.1OP	1 KB	1OP File		📄 two_node-formulas.net
📄 TWO_NODE-Formulas.ALS	1 KB	ALS File		📄 two_node-two_node.net
📄 TWO_NODE-Two_Node.ALS	1 KB	ALS File		🎮 two_node-formulas-bias.sim
🔳 TWO_NODE.DBK	143 KB	DBK File		🎮 two_node-formulas-r1_sweep.sim
🌀 TWO_NODE.DSN	143 KB	Microsoft OLE DB Provider		🎮 two_node-formulas-r3_sweep.sim
📄 two_node-Formulas-Bias.mrk	1 KB	MRK File		🎮 two_node-two_node-bias.sim
📄 two_node-Formulas-R1_Sweep.mrk	1 KB	MRK File		
📄 two_node-Formulas-R3_Sweep.mrk	1 KB	MRK File		
📄 two_node-SCHEMATIC1-Default.mrk	1 KB	MRK File		
📄 two_node-Two_Node-Bias.mrk	1 KB	MRK File		
🔳 Two_Node.opj	3 KB	OPJ File		
📄 two_node-Formulas-R1_Sweep.prb	1 KB	PRB File		
📄 two_node-Formulas-R3_Sweep.prb	1 KB	PRB File		
🖼 two_node-Formulas-R1_Sweep.dat	59 KB	Probe Document		
🖼 two_node-Formulas-R3_Sweep.dat	3 KB	Probe Document		
📄 two_node-Formulas-Bias.sim.cir	1 KB	PSpice Circuit File		
📄 two_node-Formulas-R1_Sweep.sim.cir	1 KB	PSpice Circuit File		
📄 two_node-Formulas-R3_Sweep.sim.cir	1 KB	PSpice Circuit File		
📄 two_node-Two_Node-Bias.sim.cir	1 KB	PSpice Circuit File		
📄 TWO_NODE-Formulas.net	1 KB	PSpice Netlist File		
📄 TWO_NODE-Two_Node.net	1 KB	PSpice Netlist File		
📄 two_node-Formulas-Bias.out	2 KB	PSpice Simulation Output		
📄 two_node-Formulas-R1_Sweep.out	2 KB	PSpice Simulation Output		
📄 two_node-Formulas-R3_Sweep.out	2 KB	PSpice Simulation Output		
📄 two_node-Two_Node-Bias.out	2 KB	PSpice Simulation Output		
🎮 two_node-Formulas-Bias.sim	2 KB	PSpice Simulation Settings		
🎮 two_node-Formulas-R1_Sweep.sim	2 KB	PSpice Simulation Settings		
🎮 two_node-Formulas-R3_Sweep.sim	2 KB	PSpice Simulation Settings		
🎮 two_node-Two_Node-Bias.sim	2 KB	PSpice Simulation Settings		

FIGURE 11
> The original project files and the archived files for the two-node design project; the number of files is reduced to 8 and the size by about a factor of two

The files for the project now look like Figure 11. Notice that the ARCHIVE feature saves only a subset of the files. In addition, the storage of masses of simulation data is avoided, resulting in a substantial saving in memory, which can be much more than shown here.

Back up often. If you want to use only one archive file, you will have to make a new one and then delete the old one because CAPTURE will not overwrite an existing archive file.

E-MAIL

The archive serves not only as back up, but also as a convenient abbreviated file for e-mail transfer. If you e-mail the files for a project, it may happen that the files are automatically zipped by your mail utility, or you may zip them yourself. The recipient of the zipped files must move all the zipped files into a new folder upon receipt, and *not* attempt to run the project from *inside* the opening menu of the zip utility. The files of the project all must be able to talk to each other for the project to work, and this requires they be extracted from the zip utility.

Summary of file management

Looking at the large number of files in a project, as seen in Figure 11 for example, the importance of keeping a project in a file separate from other projects is clear. When you add to the sheer number of files the extra problem that every so often a project becomes defunct, inaccessible, and so has to be deleted, separate files are even more important.

Archiving a project is valuable for back up, for e-mail, and for storing a project in a small memory space.

Simulation profiles and PROBE plots

A complete discussion of simulation profiles can be found in Herniter, Chapters 3-6. Here we just provide a brief introduction using the two-node design problem as an example.

There are four types of simulation available in PSPICE, called SIMULATION PROFILES. The types are found using the menu PSPICE/NEW SIMULATION PROFILE/CREATE to obtain the SIMULATION SETTINGS menu. The profile types are listed on this menu as shown in Figure 12.
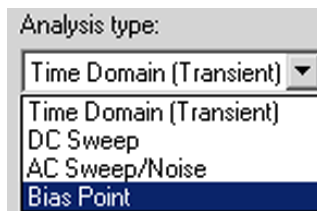
FIGURE 12
The four types of SIMULATION PROFILE available

Here we will focus on two of these types: the BIAS POINT simulation and the DC SWEEP.

The BIAS POINT simulation profile

This simulation simply runs a DC analysis and displays currents and voltages on the circuit schematic. An example of BIAS POINT output is Figure 1. The voltages and currents can be toggled on and off using the toolbar buttons V and I as shown in Figure 13.

FIGURE 13
Toolbar buttons to toggle voltage and current display on circuit schematic

The DC SWEEP simulation profile

Figure 14 shows the menu to set up a sweep of the parameter R1. The variable R1 is specified in a PARAMETER box in Figure 7. In the SIMULATION SETTINGS menu it is identified as a GLOBAL PARAMETER. A logarithmic SWEEP TYPE is selected because the selected range is large (1kΩ to 1 MΩ) and so a log scale seems likely to show the results better than a linear scale.
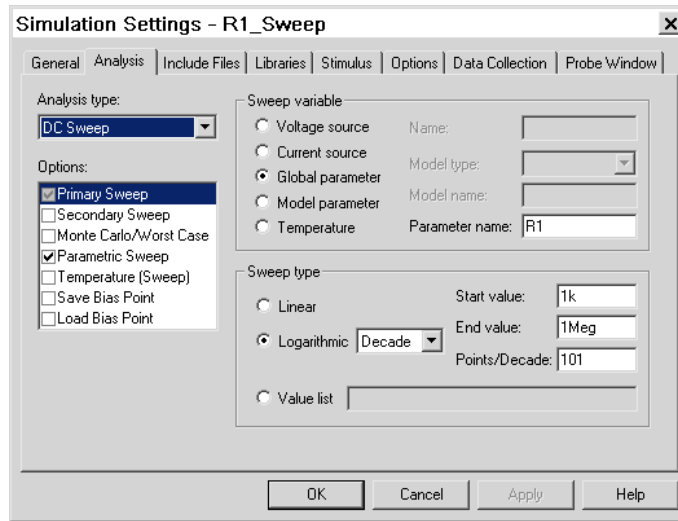
The SIMULATIONS SETTINGS menu for a DC SWEEP simulation profile

Also checked in Figure 14 is the PARAMETRIC SWEEP option. Clicking on this option produces the menu of Figure 15 where the parameter α is identified as a GLOBAL PARAMETER with values chosen from a VALUE LIST. The parameter α also is defined in a PARAMETER box in Figure 7 The menus of Figure 14 and Figure 15 produce a set of three curves, one for each α- value.
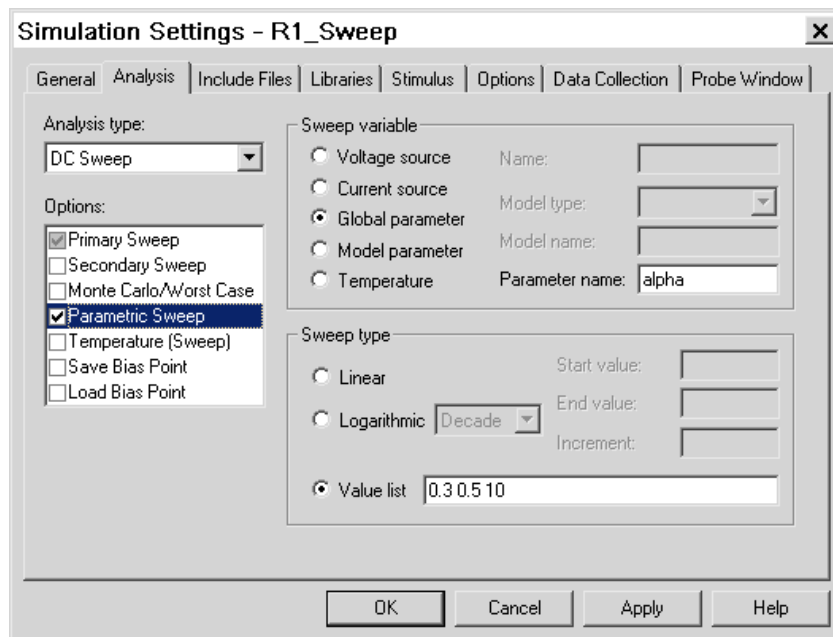

FIGURE 15

The PARAMETRIC SWEEP menu for the DC SWEEP simulation profile

Access to simulation results is provided through PROBE. For example, one can attach CURRENT PROBES to the schematic, as shown in Figure 16. These probes cause PROBE to plot the currents in DC sources I1 and I2 as found using the formula in the PARAMETER BOX.
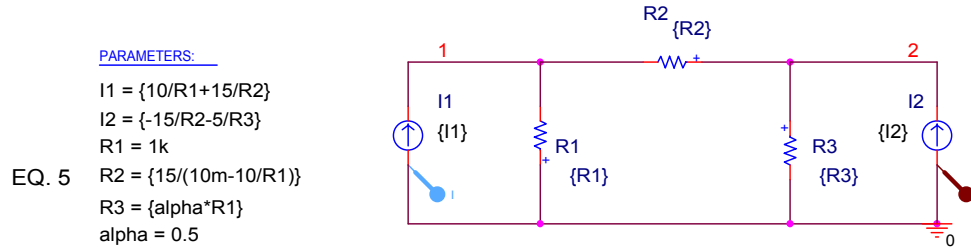
I1 = {10/R1+15/R2}

I2 = {-15/R2-5/R3}

R1 = 1k

EQ. 5   R2 = {15/(10m-10/R1)}

R3 = {alpha*R1}

alpha = 0.5

R2 {R2}

1   I1 {I1}   R1 {R1}   R3 {R3}   2   I2 {I2}

0

FIGURE 16

Attaching CURRENT PROBES to the schematic to produce current plots in the PROBE window

The resulting plots are shown in Figure 17. Notice that symbols and colors are used to identify the curves. Although PROBE automatically chooses colors and symbols, they can be selected by putting the cursor on the symbol for the desired curve and right clicking. Choosing the PROPERTIES menu selection, we obtain the menu shown in Figure 18.

20mA

0A

-20mA

1.0K   3.0K   10K   30K   100K   300K   1.0M

□ ◇ ✕ I(I1)   △ ◇ ▽ I(I2)

R1
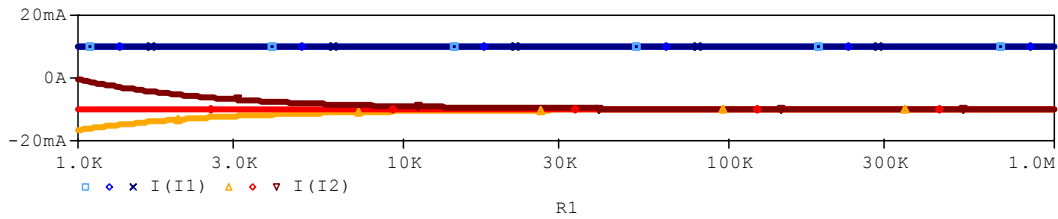
FIGURE 17

Results of the DC SWEEP simulation profile R1_SWEEP

Trace Properties ✕

Color

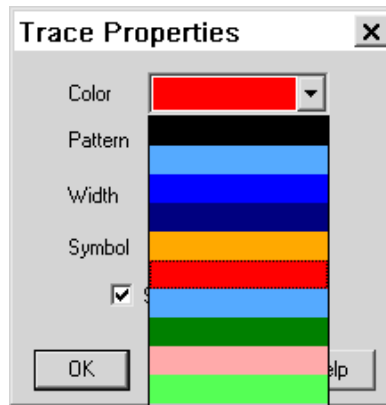Pattern

Width

Symbol

☑ S

OK   elp

FIGURE 18

The PROPERTIES menu showing the color scroll down palette

Likewise, we can display the parameter value for α corresponding to any curve by placing the cursor on the curve symbol, right-clicking, and choosing the INFORMATION menu shown in Figure 19.
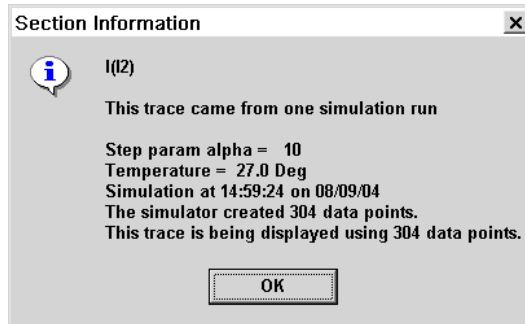
The INFORMATION menu obtained by right-clicking on a curve symbol in Figure 17

If we selected the SECONDARY SWEEP option in Figure 14 instead of the PARAMETER SWEEP option, we get similar curves, but less flexibility in displaying and identifying the curves.

Performance Analysis

Using both the primary DC sweep of Figure 14 and a parameter sweep using the menu of Figure 20, we can produce a very useful type of plot called a PERFORMANCE ANALYSIS.
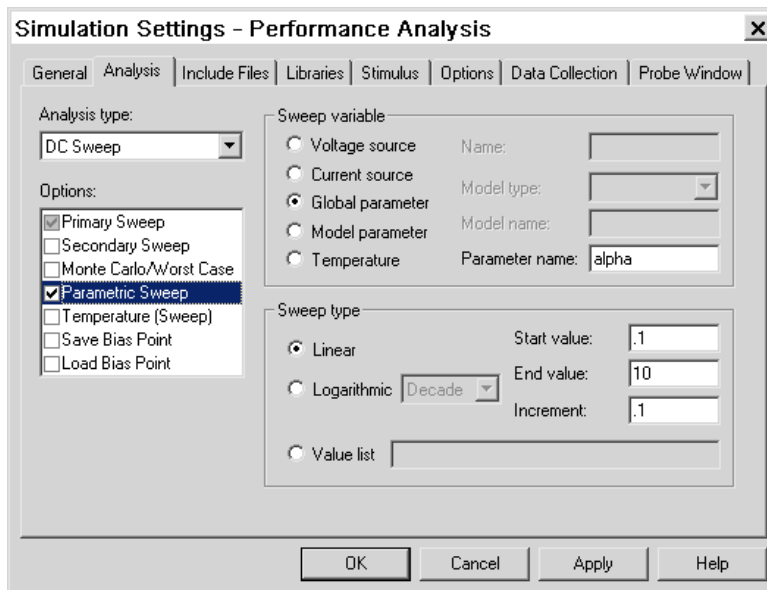

FIGURE 20

The PARAMETRIC SWEEP setting for a PERFORMANCE ANALYSIS

To illustrate this approach, we run the simulation with no probes on the schematic. PROBE then provides a blank window for the plot. We then use the menu PLOT/AXIS SETTINGS to get the menu of Figure 21.
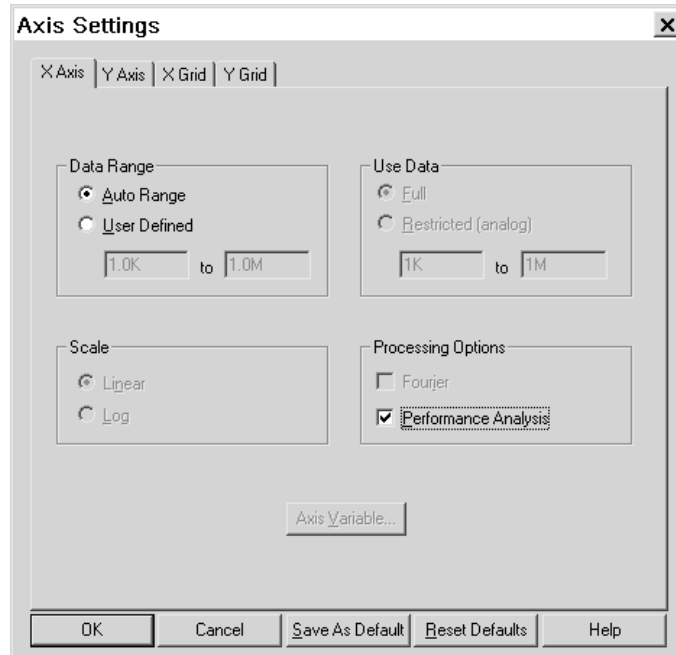
FIGURE 21
      AXIS SETTINGS menu with the PERFORMANCE ANALYSIS box checked

Choosing PERFORMANCE ANALYSIS causes the *x*-axis to become the variable of the PARAMETER SWEEP, α in this case, as shown in Figure 22.
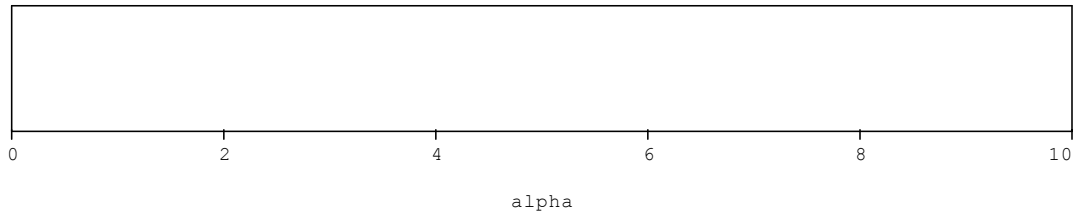

    0   2   4   6   8   10

alpha

FIGURE 22
      With the PERFORMANCE ANALYSIS box checked, the *x*-axis variable becomes α

To plot a curve we use the menu TRACE/ADD TRACE, shown in Figure 23.
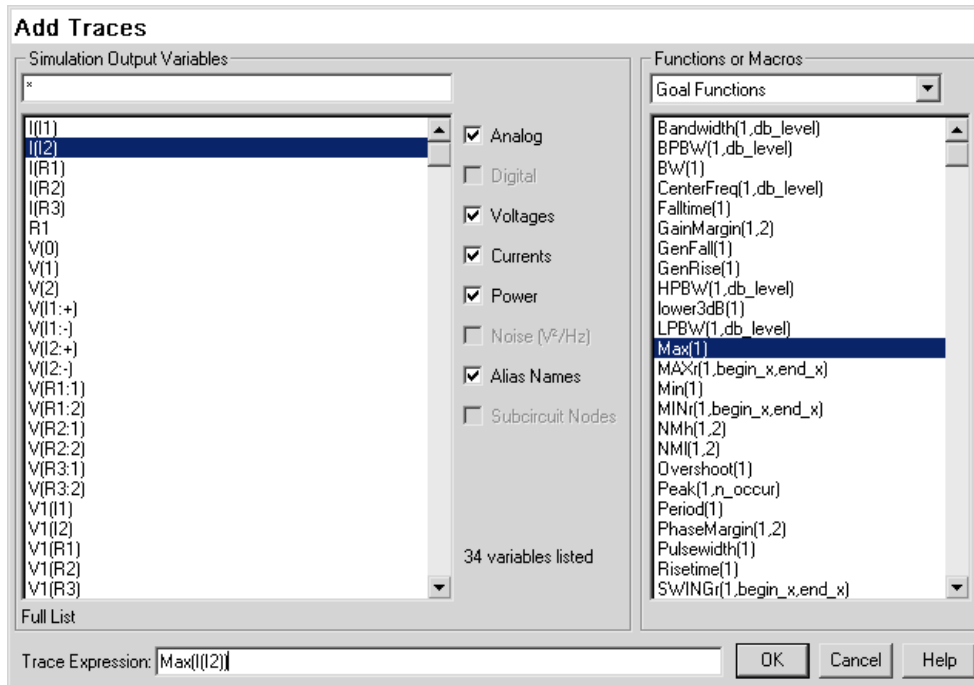
FIGURE 23
The TRACE/ADD TRACE menu with the GOAL FUNCTION selected MAX()

In the case of a PERFORMANCE ANALYSIS we use a GOAL FUNCTION from the right column of Figure 23. Each GOAL FUNCTION scans the complete range of R1 used in the primary sweep to find the corresponding goal at each value of α. In Figure 23 the GOAL FUNCTION to find the maximum value of I2 at each α is chosen. In Figure 24 both the maximum and the minimum values of I2 are shown for each value of α. For example, the plot shows that we must keep α > 0.5 to keep MIN(I2) > –10 mA.
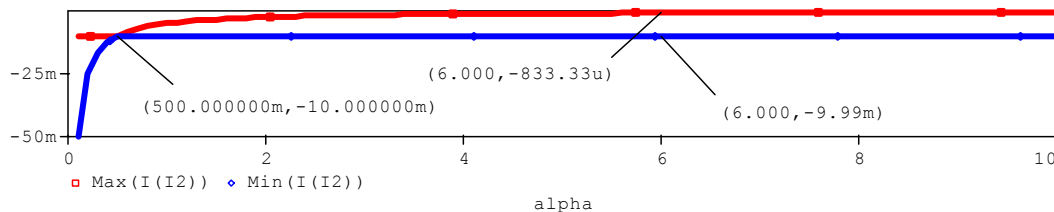

FIGURE 24
Maximum and minimum currents I2 as a function of PARAMETER SWEEP variable α

The PSPICE.ini file

You may have noticed that the PROBE plots in this document are plotted on a white background. However, the default background for PROBE is black, which does not look too good when results are pasted into reports like this one. To change the background to white we have to edit the PSPICE.ini file, located in a standard installation of Orcad 9.2 at C:\Program Files\OrcadLite\PSpice\PSpice.ini. Opening this file using NOTEPAD, we change the following entries from the default values to the values listed in Figure 25. The colors for the curves have to be changed too, because dark colors show up better on a white background than the default colors, which are selected for use with a black background. The line TRACEWIDTH=4 is added to make the plotted curves a heavier weight, which looks a bit clearer in print. These changes have been made already in the I^{2}TL.

```
[PROBE DISPLAY COLORS]
NUMTRACECOLORS=12
BACKGROUND=Brightwhite
FOREGROUND=Black
TRACE_1=LightBlue
TRACE_2=Blue
TRACE_3=DarkBlue
TRACE_4=Orange
TRACE_5=Red
TRACE_6=DarkRed
TRACE_7=DarkGreen
TRACE_8=BrightGreen
TRACE_9=Pink
TRACE_10=LightGray
TRACE_11=DarkGray
TRACE_12=Purple
TraceWidth=4
```

FIGURE 25
Modification of the PSPICE.ini file to obtain dark colored curves against a white background for import into printed documents

References

1. Marc E Herniter, *Schematic Capture with Cadence PSPICE,* Second Edition, Prentice Hall, 2003.