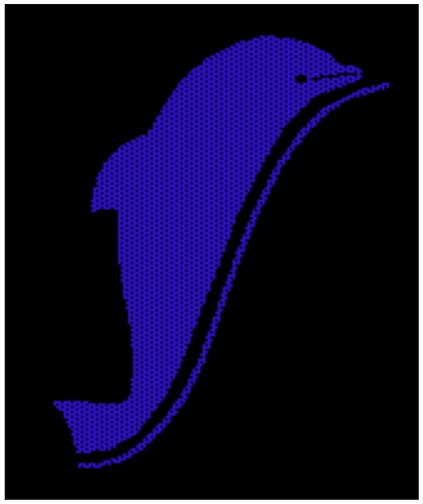
SMASH 3™

True mixed-signal simulation for Windows™

On-line Evaluation Tutorial



Introduction

This on-line tutorial is an overview of what can be achieved with SMASH. It will guide you through the execution of two simple basic simulation examples. The first one is a very simple circuit which will familiarize you with the SMASH simulation environment. The second one demonstrates some of the advanced features of the simulator.

Please read this!

The SMASH evaluation software you have downloaded is actually a limited but fully functional version of the commercial product. What is limited is the maximum size of the circuits you can run (25 analog and 50 digital nodes). It is NOT a slide show or self-running demo. As a consequence, please READ and follow thoroughly the instructions in this tutorial when you run the examples, because activation of commands is “for real.”

For technical support and questions...

USA/Canada: Protel Technology, Inc.
Tech support hotline: 1-800-827-2910
Tech support faxline: 1-408-243-3762
email: helpme@protel.com

Dolphin Integration, BP 65, ZIRST, 8 chemin des Clos,
38242 MEYLAN CEDEX, FRANCE
fax: (33) 76 90 29 65
email: dolphin.int@applelink.apple.com

Installing the SMASH eval software

- You will have already used PKUNZIP to de-compress the software and this documentation, which will be temporarily located in a hard disk directory.
- In the Windows Program Manager, choose Run from the File menu, then (from the temporary directory) type: "A:\INSTALL" and click OK.
- Follow the on-screen instructions, until installation is completed.
- Reboot your computer, then re-start Windows.

Running the RC example

This simple example demonstrates the possible analyses in SMASH. As it may be the first example you will run with SMASH, we chose a simple one, purely analog, using only primitives. However, it will show interesting features like the parameter sweeping capability, and Monte Carlo analysis. It is a simple RC network:

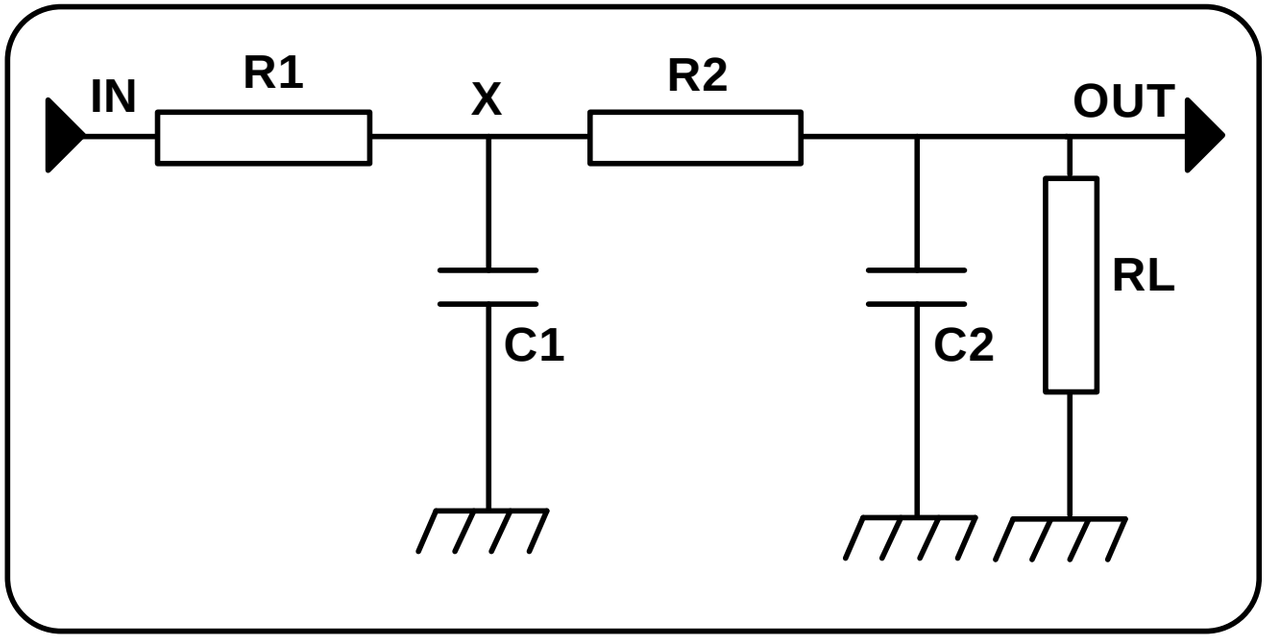


Fig. 1. RC example schematic

Step 1

Once SMASH is installed (see page 2), locate the SMASH icon in the SMASH Evaluation kit program group. This group was created at installation time. Double-click on the SMASH icon to run it. Click in the copyright window to make it disappear. A standard file selection dialog then pops up. Now move to the \smash\booklet directory, then double-click on RC.NSX.

Step 2

Two windows appear side by side, showing RC.NSX and RC.PAT. Use the scroll bars to inspect the contents of the files. RC.NSX is the netlist file, the one which contains the description of the circuit. RC.PAT is the pattern file, it contains the stimulus descriptions and simulation directives. The * character is the comment indicator. Anything following the * character is ignored by SMASH.

Note: SMASH contains a built-in multiple windows text editor. You can perform basic text edition with the commands in the File menu (New, Open..., Save, Save as... etc.)and the Edit menu (Cut, Copy, Paste, Find... etc.).

Step 3

Now we will simulate our circuit. In the Analysis menu, select the Operating point. item. A dialog pops up. Click on the Run button. A text window appears, named RC.OP. This file contains the bias point description. You can read the bias point voltages on the nodes of the circuit.

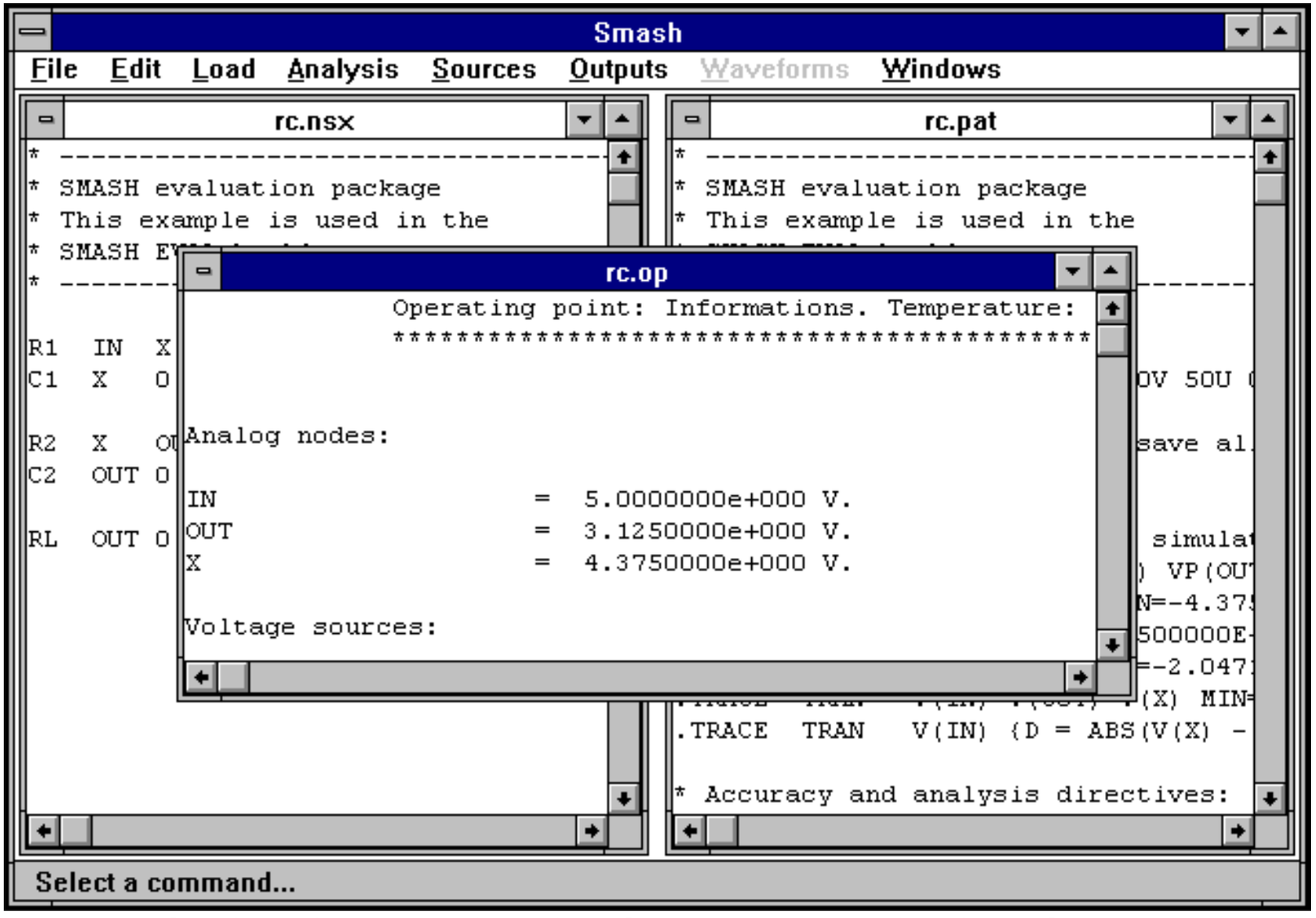


Fig. 2. Netlist (.NSX), pattern (.PAT) and operating point (.OP) windows.

Step 4

In the Analysis menu, select the Transient> Parameters... item. A dialog showing the transient analysis parameters appears. Click on the Run button. Watch the transient analysis running, with the waveforms being displayed as they are computed.

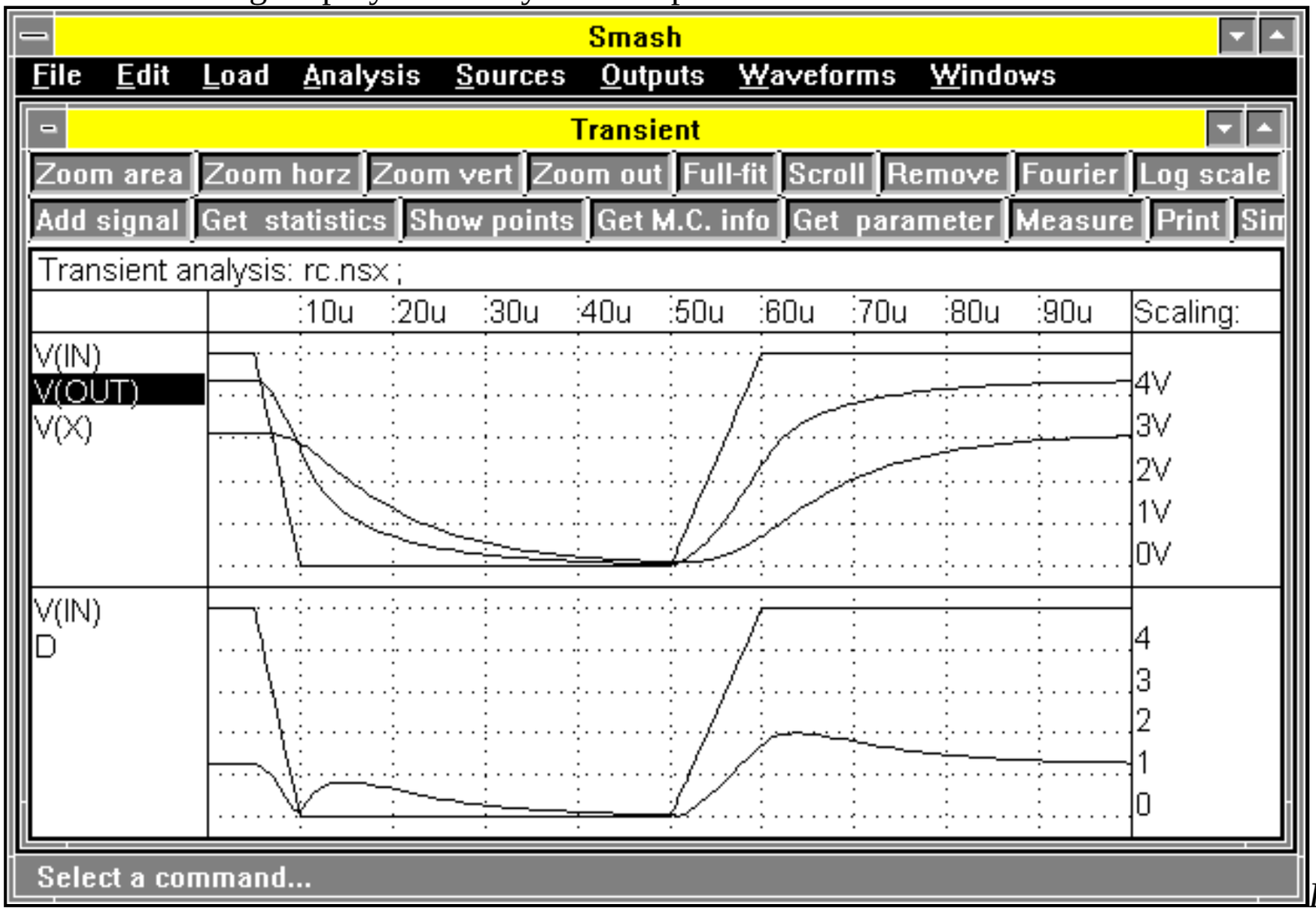


Fig. 3. Transient window.

Graphic windows in SMASH are organized in graphs (a horizontal zone in the window), each graph containing one or several traces (signals). Displayed traces may be voltages, currents, formula

involving several signals, or logic signals. During the simulation you may notice a small cursor moving up and down, on the left side of the screen. This is the current internal timestep.

We drove the IN input with a pulse signal: you can see the shifted responses of nodes X and OUT. Also, you can see the D waveform, which is not a true signal but a formula (search for the .TRACE directive which defines trace D in RC.PAT).

Waveform processing (zooming, measuring etc.) is available *even when a simulation is running*, through the commands under the Waveforms menu. Here are some possibilities:

You may add new signals in the window with the Add> analog signals item (or Add signal button). A dialog box displays a list of available signals. Double-click the name of the signals you want to view. Click the Done button to close the dialog.

You may re-arrange signals and graphs using the drag-and-drop method: in the Transient window, click the name of the signal you want to move, then drag it over the desired destination (inside an other graph), and drop it.

You may choose to isolate a signal by double-clicking its name: only the signal is shown. To come back to the previous view, double click its name again. Any click in the window is reported in the top region of the window, with the click coordinates and deltas.

Note: if you get lost, here is a generic recovery procedure:

- Bring the Transient window to front (Windows menu, Transient item)
- Hit the ESC key (this will cancel the current command).
- If the View All item is active (in the Waveforms menu), select it.
- Select the Full-fit command, in the Waveforms menu.

Step 5

In the Analysis menu, select the Small signal> Parameters... item. A dialog with the small signal parameters pops up. Click on the Run button. Observe the AC analysis running, with the waveforms being displayed as they are computed.

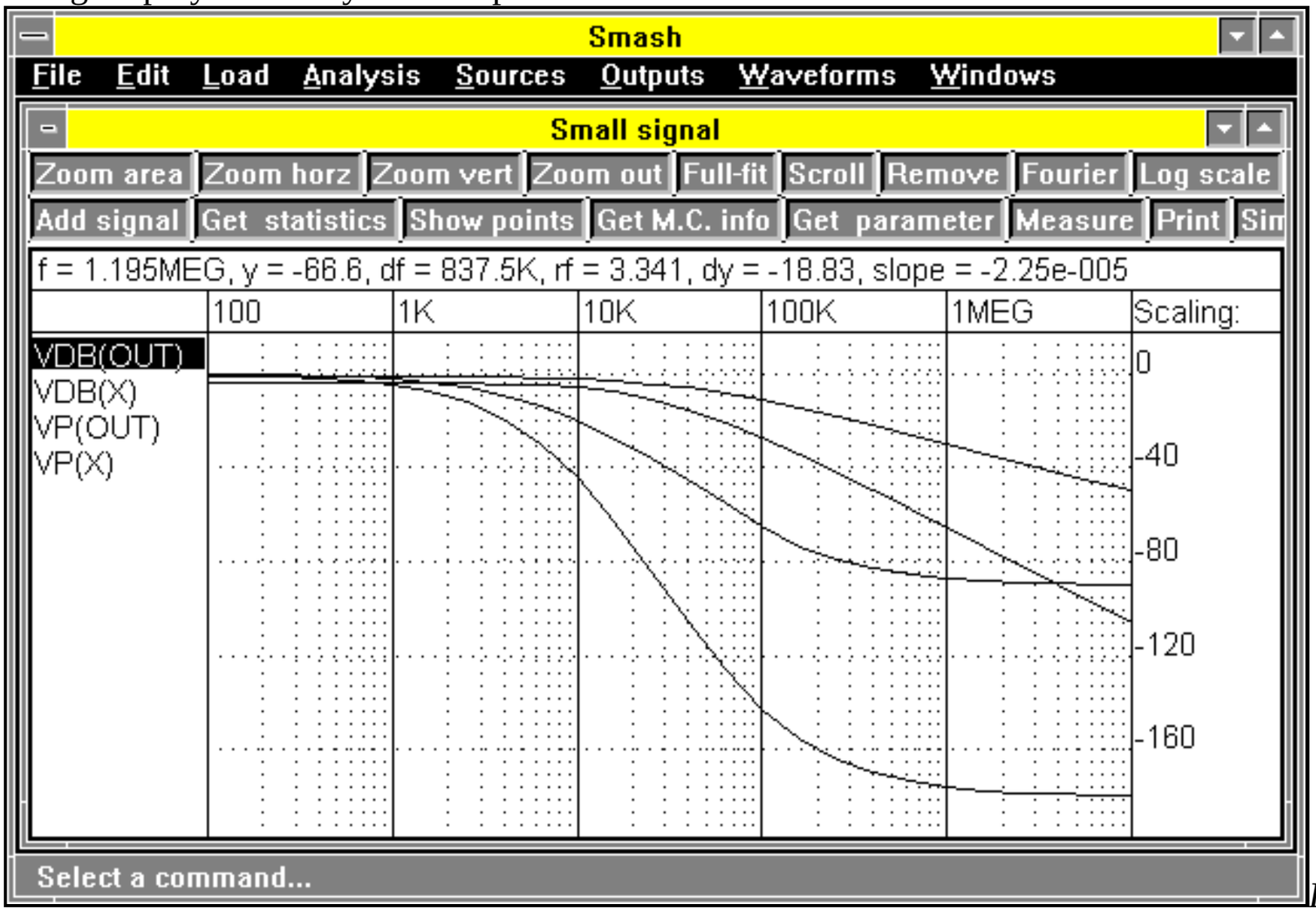


Fig. 4. Small signal window.

The previous window, with results of the transient analysis, is not lost... you can bring it back to the front by clicking in its title bar, or by selecting the Transient item in the Windows menu if it is hidden. You can play with the windows (resize them, move them, close them, etc..). Waveform processing is available as well (see step 4).

Step 6

In the Analysis menu, select the DC transfer> Run item. Watch the DC analysis running, with the waveforms being displayed as they are computed. Notice that this time, by selecting the Run item, we bypassed the parameter dialog, and launched the analysis directly...

As our circuit is a simple RC network, this continuous analysis is not very exciting... we just check that the resistance bridges behave as we would expect! SMASH is also able to run a noise analysis. If you wish, select the Noise> Run item in the Analysis menu. The output noise DB(ONOISE) is shown in the Noise window. The resistors simply generate thermal noise... A text file containing noise analysis results is created (RC.NZE). You may open it with the File Open... command.

Step 7

We've been through the simple analyses. Now we will see what happens when we vary the value of a resistor, how it impacts the response of the circuit.

In the Analysis menu select the Transient> Sweep item. The transient analysis is iterated with the value of a parameter (here the value of resistor R1) changed at each iteration. The results are superimposed in the same window, for easy estimation of the impact of the parameter variation.

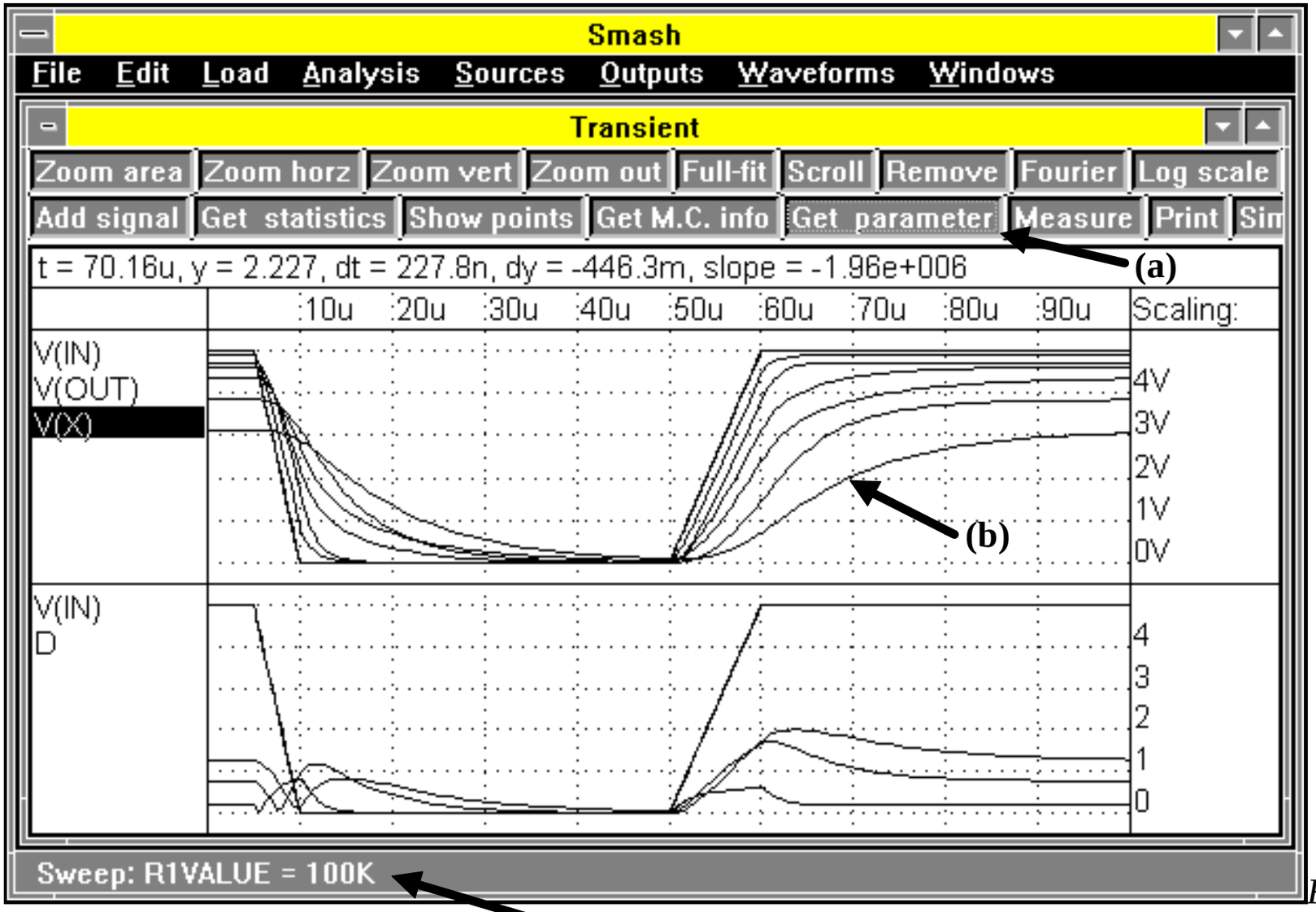


Fig. 5. Transient sweep analysis..

Click the Get parameter button in the toolbar (a), then click on waveforms (b). Each time you click, the value of the parameter which corresponds to the curve is displayed in the prompt bar (c).

Step 8

We can launch the Small signal> Sweep analysis, the DC transfer> Sweep analysis, and also the Noise> Sweep analysis. For each of the sweep analyses, the results are immediate and superimposed in the corresponding analysis window.

Step 9

Now we will activate the Monte Carlo analyses. The value of components is varied at random, according to specified statistical distributions and tolerances, and the analyses are re-run a number of times. So we can simulate how the response is modified when we pick the components off the shelf. In the Analysis menu, select Transient> Monte Carlo. As usual, the results are superimposed in the same window. The current run number is indicated in the prompt bar. You can run the other Monte Carlo analyses if you like (Small signal> MonteCarlo for ex.).

Now, bring the transient window to the front. Click the Get M.C. Info button in the toolbar, then click on waveforms. The parameter values for the run corresponding to the clicked waveform are displayed in a window in the bottom of the screen. Deviations from the nominal values are shown in this window. When you are done, close the RC.MC window, and click the Full-fit button

Step 10

We've seen a lot of things. You probably wonder where SMASH was told to do these things. In the Windows menu, select RC.PAT. This file, which we call the pattern file, is read by SMASH upon a Load Circuit... operation, together with the netlist (RC.NSX). It contains the user directives to setup simulations. Detailing all features of the pattern file is beyond the scope of this manual. However, the file contains a lot of comments, so you can take a look at it if you are interested. For example, look at the .PARAM statements to see how the values of the components are parametrized.

Let us answer your next question, which probably is: how was this pattern file created? Well, it depends. Some people want to edit the pattern file, and write the directives by hand. Some other people don't want to manually edit anything... SMASH lets you work the way you like. You can edit

the pattern file and rerun simulations (the SPICE way...), or use the dialogs to specify simulation directives interactively. If the Update pattern file box is checked in any dialog you close by clicking on OK, any data you entered in the dialog fields will be backed up in the pattern file, so it is not lost for the next session. If you work this way, you can enter most of the directives through dialogs, never edit the pattern file manually, and forget about the syntax of the commands. It is up to you...

If you want to test this feature, try the following: in the RC.PAT window, search for a line starting with .AC, which specifies the AC analysis parameters. Remember this line. Activate the Small signal> Parameters item in the Analysis menu. In the dialog that pops up, modify the field "Number of points" (enter twice the current value), and modify the Stop frequency field (enter 10 times the current value). Verify that the "Update the pattern file" checkbox is checked, then click on Run. A new AC analysis is run. Now bring the RC.PAT window back to the front (Windows menu, RC.PAT item), and search for the .AC line again. It should have been modified, to reflect the changes you made in the dialog, and thus these changes are saved for the next time you will simulate this RC circuit.

Running the MIX example

This example demonstrates the mixed signal (analog plus digital) and multi level (structural plus behavioral) capabilities of SMASH.

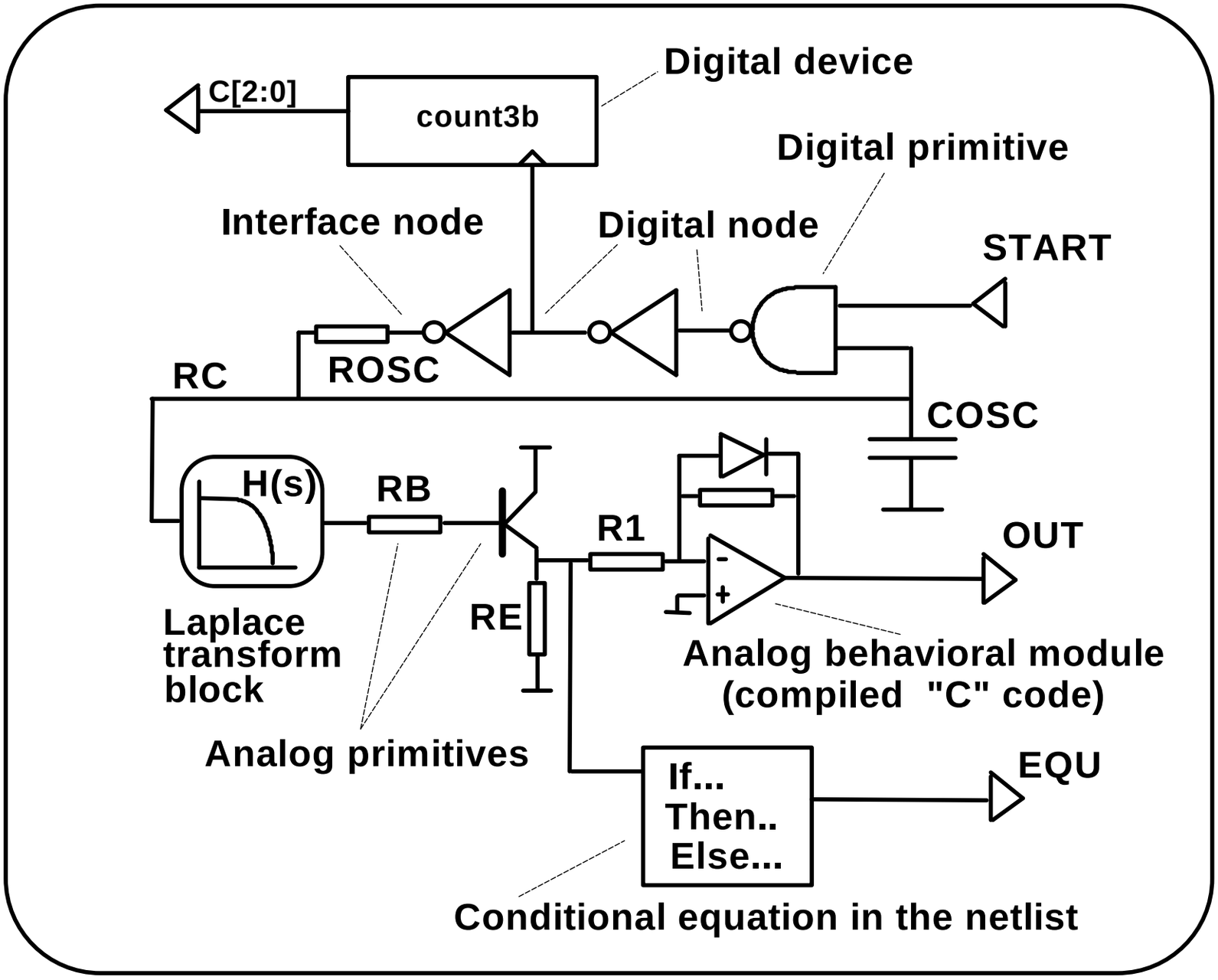


Fig. 6. MIX example schematic.

The circuit contains a simple RC oscillator, a three bit counter, and analog components to perform some basic analog signal processing. The oscillator and the counter are built with digital gates (described in Verilog-HDL), and analog components (resistor and capacitor). It generates an RC-like waveform, the period of which is set by ROSC and COSC. A Laplace-defined block filters this signal to extract its fundamental. Thus, the bipolar transistor is fed with a (distorted) sinewave. The operational amplifier amplifies this signal by a factor -2, and the diode passes the negative alternances to the output. In parallel, the EEQU block performs the same operation as the amplifier/diode section, with a conditional equation.

Caution: If you just run the previous RC example, and you are currently running SMASH, please quit SMASH now (File Quit), before proceeding.

Step 1

If not already done, it is time to install the software (see page 2). Once SMASH is installed, locate the SMASH icon in the «SMASH Evaluation kit» group in Windows. This group was created by the installation procedure. Double-click on the SMASH icon to launch SMASH. Click in the copyright window to make it disappear...

A standard file selection dialog pops up. Move to the \smash\booklet directory and double-click on MIX.NSX. SMASH loads the MIX.NSX netlist, together with its associated pattern file, MIX.PAT

Step 2

Use the scrollbars to inspect the content of the MIX.NSX file. This is the netlist file, which contains the circuit description. This evaluation manual is not a replacement for the 600 page SMASH documentation, and is no space for a detailed description of the syntax, but you may try to locate the statements corresponding to the devices in figure 6. You will find standard SPICE statements, along with SMASH extensions, as for digital gates (Verilog-HDL), and conditional equations.

Step 3

Now we will simulate our circuit. In the Analysis menu, select the Operating point... item. A dialog pops up. Click on the Run button. A text window appears, named MIX.OP. This file contains the bias point description. You can see the bias point voltages and logic levels on the nodes of the circuit. Also in the .OP file, you will find transistor bias information (here QBJT, see figure below). The SMASH model is the standard SPICE model (modified Gummel-Poon)

```
.Bipolar transistors:
====================
QBJT :
  IE = 5.73230e-004, IC = 5.69342e-004, IB = 3.88873e-006, RB = 1.00000e+001
  CBE= 4.49399e-011, CBC= 3.88810e-012, CBX= 0.00000e+000, CSC= 0.00000e+000
  RPI= 7.44929e+003, RMU= 1.00000e+012, R0 = 1.37059e+005, GM = 2.19638e-002
  VBE= 6.29894e-001, VBC=-4.00336e+000, VCE= 4.63325e+000, RC = 1.00000e+000
 xVBE= 6.29933e-001,xVBC=-4.00389e+000,xVCE= 4.63382e+000, RE = 0.00000e+000
  Bdc= 1.46408e+002, Bac= 1.63615e+002, FT = 7.15912e+007
```

Fig. 7. Bias currents and small signal parameters in the .OP file.

This circuit being a mixed signal circuit, SMASH has to deal with both continuous analog voltages and discrete logic levels. Whenever a node is connected to both analog components and digital components, it becomes an interface node.

Step 5

In the Analysis menu, select the Transient> Run item:

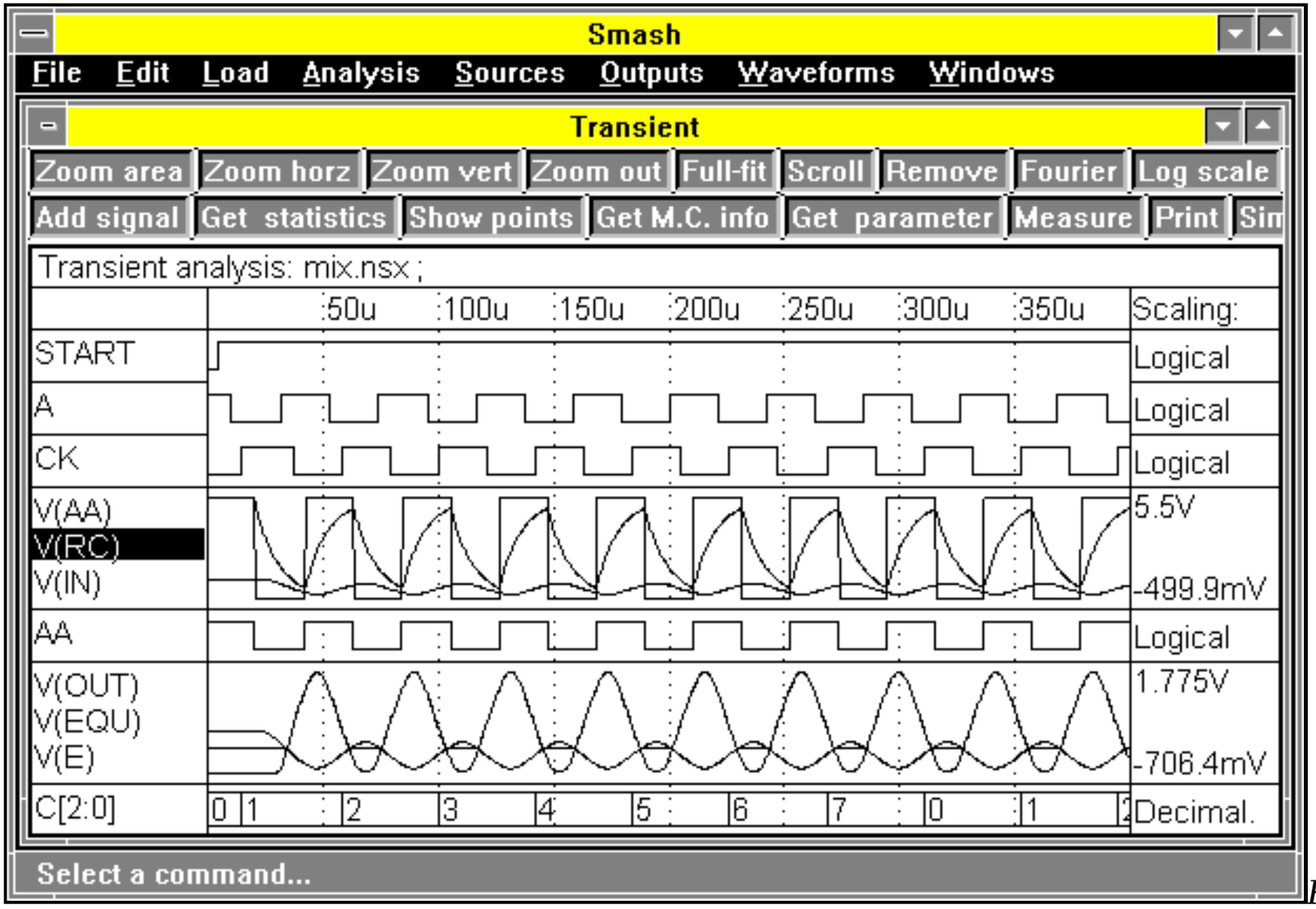


Fig. 8. Transient window.

You can see the transient analysis running, with the waveforms being displayed as they are computed. Waveforms on digital nodes are flagged as Logical for bit signals (A), or with their radix for bus signals (C[2:0]), in the Scaling zone of the window. The START signal is used to start and

stop the oscillator. The OUT signal and the EQU signal are almost identical. OUT is generated with the opamp. and the diode, and EQU is generated with a conditional equation (see the MIX.NSX file). The difference between OUT and EQU is due to the diode.

Step 6

Now we would like to analyze the spectrum of the transient waveforms. We want to compute the spectrum of signal AA with a FFT analysis. This time, you will need to enter a few parameters yourself...

- Bring the Transient window to front,
- In the name zone, select the V(AA) signal (click its name once),
- Click the Fourier button in the toolbar, or select the DSP functions>FFT... item in the Waveforms menu: the Fourier transform dialog appears.
- Enter `10N` in the "Sampling step" field,
- Enter `500` in the "Start point index" field,
- Enter `2048` in the "Number of points" field,
- Select a Blackmann-Harris 4 windowing function,
- Click on OK. The FFT of signal V(AA) is computed, and a new FFT window is created, displaying the spectrum of signal V(AA). The figure on next page shows the low frequency part of the FFT plot.
- Using the same procedure, compute other FFTs on other signals (FFT is available for both analog and digital signals) if you like, they will be added in the FFT window. Waveform processing is available for the FFT plots, as for normal simulation windows (such as the transient window).

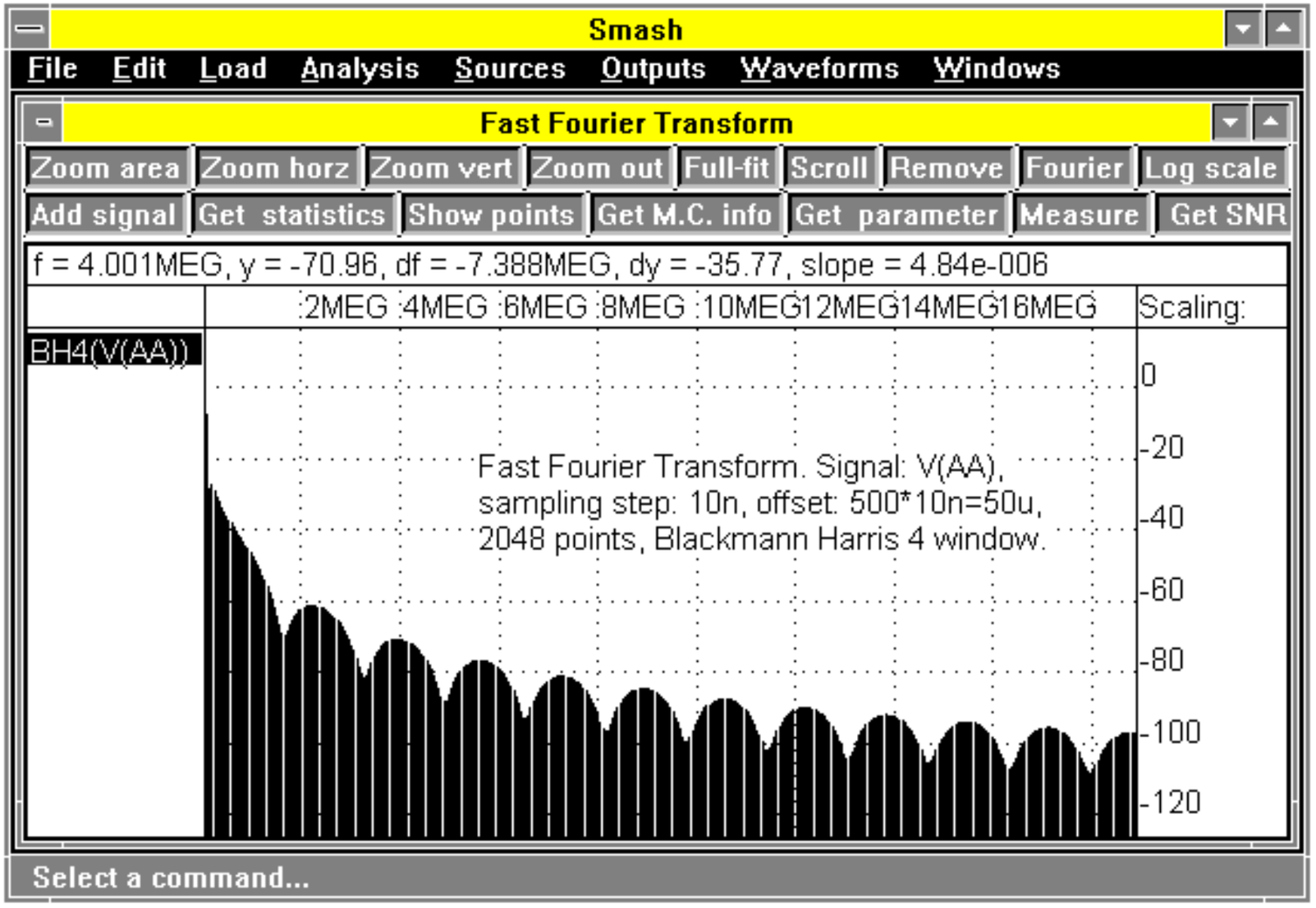


Fig. 9. FFT window.

A few words about Verilog-HDL...

SMASH uses the IEEE standard Verilog-HDL for digital descriptions. This ensures that you can import libraries into SMASH with no recoding or rewriting into some non-portable proprietary language. Verilog-HDL support in SMASH is based on the OVI specifications (OVI stands for "Open Verilog International", and LRM for "Language Reference Manual"). Both structural and behavioral constructs are supported. The complete strength model is supported, as opposed to most Verilog clones, which only support an over-simplified (0,1,X,Z) model. Verilog-HDL behavioral constructs are transparently translated to C, compiled, and dynamically linked, which provides the best speed performance for large digital circuits. For optimum productivity at an early stage of the design, when you need several iterations before you are satisfied, the whole process is incremental. Structural constructs (gate-level) do not require C compilation.

The digital parts in a circuit, described in Verilog-HDL, are handled and simulated by SMASH with specific data structures and algorithms (tailored to efficient logic simulation). They are not simply merged into the analog equations (this technique would not allow simulation of large circuits). Thanks to these advanced techniques, there is no performance penalty to pay when running either a pure analog simulation or a pure digital simulation. When SMASH is used for either pure analog or pure digital simulations, there is simply no mixed signal overhead.

SMASH is the only professional solution for mixed signal simulation which is based on standards (SPICE for analog, and Verilog-HDL for digital), and does not use any inefficient simulation backplane...

Running other examples

This evaluation kit contains several examples (pure analog, pure digital, mixed...) in the \smash\examples directory, which you can easily run, now that you are an expert... All netlist and pattern files are commented, and they illustrate the different capabilities of the simulator. Particularly:

- the PLL_BHV and NE555 directories both contain examples of mixed signal simulations,
- the BDF_ALGO directory contains an example to illustrate the accuracy of the BDF integration algorithm,
- the PROTEL directory contains a sample Protel schematic, which is configured for SMASH simulation (instructions are in the Protel schematic (.sch binary file)),
- the VERILOG directory contains examples of digital simulations, to illustrate the use of Verilog-HDL,
- the EKV directory contains examples using the EPFL-EKV model,
- the FILTERS and DIG_SINE directories contain examples with (resp.) analog and digital behavioral models.

Also, to understand why SMASH is definitely the premier reference in mixed signal, multi-level simulation, double click the «Read Me!» icon. The associated file contains information about behavioral modeling, schematic entry, libraries, etc.

The SMASH solution...

Supported platforms:

- PC under Windows 3.1, 95 and NT. Recommended minimum configuration: 486DX/8Mb.
- Sun Sparc, Hp 700 under Unix.
- Apple Power Macintosh (native PowerPC application)

SMASH for analog design

- SPICE simulation (the industry standard)
- Transient, AC, DC, Noise and Powerup analyses
- Superior convergence and algorithms (BDF* for transient simulation)
- Easy and intuitive interactive waveform viewer (graphics co-processor) with toolbars
- Support for all classical SPICE primitives
- Support for look-up tables and equation-defined sources
- Support for parameters with conditional form
- Support for parametrized subcircuits
- Additional relaxation algorithm for CMOS circuits
- MonteCarlo analysis
- MOS transistor models (1,2,3, BSIM, EPFL (for low-power))
- TRANS* add-on for inclusion of proprietary MOS models
- Laplace transforms
- Z-domain modeling
- PRISM* library for PCB design
- EMBLEM* library for electro-mechanical simulations

SMASH for digital design

Verilog-HDL simulation (IEEE-standard) at gate level, RTL* and behavioral* levels

- Full support for the Verilog-HDL strength model
- Timing analysis (setup and hold verifications) and pin-to-pin delays
- Toggle-test analysis
- Input stimuli description language
- Conversion of output results into input stimuli
- Interface to test machines
- Logic analyzer functions (edge/change tracking)
- C-based behavioral modeling*

Analog behavioral modeling*

- Genuine analog behavioral modeling based on the C language (ABCD)
- Intimately mixed with SPICE-style descriptions
- Modeling style preserves investment in SPICE models
- x1000 speed-up compared to circuit level simulation
- Dynamically linked models
- Uses industrial standard C compilers
- May model virtually any behavior
- Allows system-level simulation and pin-to-pin simulation of complete designs (ASICs or PCBs) made feasible

*Note: features available as SMASH options.