NGSpice is part of the gEDA family of free, open-source tools for electronic design. NGSpice itself is a command-line tool that reads SPICE files and writes data files. NGNutmeg is a waveform viewer that is used to plot the results from NGSpice. Installation instructions for NGSpice can be found here.

Contents

Quick Start

There are three steps in running an NGSpice simulation:

1. Create a SPICE circuit description file.
2. Run the NGSpice program.
3. Run NGNutmeg to display the results.

DC .OP Simulation

Create a simple SPICE circuit file, such as this voltage divider circuit with a fixed DC source:

```
Example Circuit 1
R1 2 1 1kOhm
R2 1 0 1kOhm
V1 2 0 DC 1v
.op
.end
```

Save this file as **example1.sp**. Now run NGSpice using the command

```
ngspice -b example1.sp -o example1.out
```

In this command, we used the following options:

- **-b** : Tells NGSpice to run in *batch mode* rather than launching its own interpreter shell.
- **-o example1.out** : Output from .op and .print statements will go to this text file.

To view the results, type **less example1.out** and press **>** to skip to the bottom.

Transient or DC Sweep Simulation

For simulations using sweeps or transient data, you may generate a raw file and view the results using **ngnutmeg**. Create a simple SPICE circuit file, such as this voltage divider circuit with a sinusoidal voltage source:

```
# This is a simple SPICE file
v1 1 0 SIN(0 1 1k)
R1 1 0 1k
R2 1 2 100
R3 2 0 100

.tran 50u 10m
.print tran v(1) v(2)

.END
```

Now run NGSpice using the command

```
ngspice -b simple.sp -o simple.out -r simple.raw
```

In this command, we used the following options:

- **-b** : Tells NGSpice to run in *batch mode* rather than launching its own interpreter shell.
- **-o simple.out** : Output from .print statements will go to this text file.
- **-r simple.raw** : Raw binary simulation data will be stored here. NGNutmeg uses the raw file for plotting and printing results.

Now launch NGNutmeg:

```
ngnutmeg simple.raw
```

This will launch the ngnutmeg shell. To list the available variables, type `display`. You will see a list of nodes, branches, and the time variable:

```
    V(1)                : voltage, real, 208 long
    V(2)                : voltage, real, 208 long
    time                : time, real, 208 long [default scale]
    v1#branch           : current, real, 208 long
```

Plot these variables using the `plot` command, e.g.:

```
plot V(1) V(2)
```

This creates a plot showing the voltages at nodes 1 and 2. To output this plot to a postscript file, enter these commands:

```
set hcopydevtype=postscript
hardcopy myplot.ps V(1) V(2)
```

The first command tells NGNutmeg to use the *postscript* file type. The `hardcopy` command is the same as the plot command, only it sends output to a file instead of your screen. To view your file, run the `evince` program from a terminal (not from within NGNutmeg):

```
evince myplot.ps &
```

If you are using the KDE desktop, use **kghostview** in place of **evince**.

Note: The plot window in NGNutmeg has a "hardcopy" button. This button has a few bugs and is not the recommended way to save your plots.

For more information on using NGSpice and NGNutmeg, type `help`. To exit NGNutmeg, type `quit`.

Plotting With Octave or Matlab

ngnutmeg doesn't always produce the nicest looking graphs and has a few quirks that get a little annoying. If you'd rather use something like Matlab to plot your graphs -- you can! There is an m file which will read in your raw file and allow you to plot it in Matlab which can be found here. If you use Octave, see this guide. Both methods are fairly easy and will allow you to use powerful tools to manipulate your data and produce better plots.

Matlab: To plot your data in Matlab, first download this file and **place it in your project directory**:

- ReadNGSpice.m (Updated 2010).

Then run ngspice and generate a raw file output. In Matlab, navigate to the project directory where your raw file is located. Then run the command

```
    sim = ReadNGSpice('filename.raw');
    name=sim{1}(1,1);name=name{1};
    data=sim{1}(1,2);data=data{1};
    labels=sim{1}(1,3:end);
```

This will load the simulation results into the *rows* of **data**, and the label for each row is provided in **labels**.

Note that there are at least two different versions of the ReadNGSpice script, and you can find them at MatlabCentral. You may use a different one if you want.

Octave: To plot your data in Octave, first download this file into your project directory:

- spice_readfile.m

Then run ngspice and generate a raw file output. In Matlab, navigate to the project directory where your raw file is located. Then run the command

```
    [data, labels] = spice_readfile('filename.raw');
```

This will load the simulation results into the *columns* of **data**, and the label for each row is provided in **labels**.

See Also

- SPICE
- User's Manual
- `man ngnutmeg`
- `man ngspice`
- ngspice Linux Installation Instructions

SoftwareTools

ElectronicsWiki: NGSpice (last edited 2014-08-28 18:55:45 by A00348692)