

Contents

1	Introduction	5
1.1	EDA design flow	5
1.2	EDA Tools	5
1.3	What is Oscad?	6
1.4	Why Oscad?	6
1.5	Structure of the book	6
2	Installing and Setting Up Oscad	7
2.1	Procedure for installing Oscad	8
2.2	Testing	14
3	Architecture Of Oscad	21
3.1	Modules used in Oscad	21
3.1.1	Eeschema – Schematic Editor	21
3.1.2	CvPcb – Component-Footprint mapper	22
3.1.3	Pcbnew – PCB Layout Editor	22
3.1.4	Analysis Inserter	23
3.1.5	KiCad to Ngspice netlist converter	24
3.1.6	Component Model Builder	24
3.1.7	Component Sub-circuit Builder	24
3.1.8	Circuit Simulator – Ngspice	25
3.1.9	Scilab based circuit simulator – SMCSim	25
3.2	Workflow of Oscad	25
4	Getting Started	27
4.1	Schematic Editor	29
4.2	Analysis Inserter	30
4.3	Netlist Converter	32
4.4	Ngspice	33
4.5	Footprint Editor	34
4.6	Layout Editor	34
4.7	SMCSim	36
4.8	Model builder	36
4.9	Subcircuit builder	39

5 Schematic Creation	40
5.1 Familiarising the schematic editor interface	40
5.1.1 Top Menu Bar	40
5.1.2 Top toolbar	42
5.1.3 Toolbar on the right	43
5.1.4 Toolbar on the left	44
5.1.5 Hotkeys	45
5.2 Components and Component libraries	45
5.2.1 Oscad libraries	46
5.2.2 Adding Oscad component libraries to project	46
5.2.3 Library browser	46
5.2.4 Plot component library	47
5.2.5 Power component library	47
5.2.6 Connector library	48
5.2.7 Component references	48
5.3 Schematic creation for Simulation and PCB design - Differences .	49
5.4 Schematic creation for simulation	51
5.4.1 Selection and Placement of components	51
5.4.2 Wiring the circuit	53
5.4.3 Assigning values to components	53
5.4.4 Annotation and ERC	54
5.4.5 Netlist generation	55
6 Simulation	57
6.1 Analysis Inserter	57
6.1.1 Types of Analysis	58
6.1.2 DC Analysis Inserter	59
6.1.3 AC Analysis Inserter	61
6.1.4 Transient Analysis Inserter	62
6.1.5 Saving the analysis file	62
6.2 Modifying KiCad netlist for Ngspice simulation	65
6.2.1 Insert parameters for fictitious components	65
6.2.2 Convert IC into discrete blocks	67
6.2.3 Insert Digital-to-Analog (D-to-A) and Analog-to-Digital (A-to-D) converter at appropriate places	67
6.2.4 Insert plotting and printing statements	67
6.2.5 Insert analysis and option	67
6.2.6 Insert models and subcircuits	67
6.2.7 Simulation	67
6.3 Examples	68
6.3.1 DC Analysis	68
6.3.2 AC Small-signal Analysis	68
6.3.3 Transient Analysis	70

7 PCB Design	74
7.1 Schematic Creation for PCB design	74
7.1.1 Netlist generation for PCB	75
7.1.2 Mapping of Components using Footprint Editor	76
7.1.3 Familiarising the Footprint Editor tool	76
7.1.4 Viewing footprints in 2D and 3D	77
7.1.5 Mapping of components in the RC circuit	77
7.2 Creation of PCB Layout	78
7.2.1 Familiarising the Layout Editor Tool	78
7.2.2 Hotkeys	81
7.2.3 PCB design example using RC circuit	81
8 Model builder and Subcircuit builder	90
8.1 Model builder	90
8.1.1 Example	90
8.2 Subcircuit Builder	92
8.2.1 Example	93
9 Scilab Based Circuit Simulation	102
9.1 Operating Point (DC) Analysis	103
9.2 Transient Analysis	104
9.2.1 Example	105
10 Oscad Spoken Tutorials	107
10.1 Beginner Level	107
10.1.1 Introduction to Oscad	107
10.1.2 Schematic creation and simulation using Oscad	108
10.1.3 Designing Circuit Schematic in KiCad	108
10.1.4 Designing Printed Circuit Board using Oscad	108
10.1.5 Electric rule checking and Netlist Generation in KiCad	108
10.1.6 Mapping components in KiCad	109
10.1.7 Designing PCB in KiCad	109
10.2 Advanced Level	109
10.3 Instruction Sheet	110
10.3.1 The procedure to practice	110
10.3.2 Please ensure	110
10.3.3 Basic Module	110
10.3.4 Schematic creation and Simulation	111
10.3.5 Designing Printed Circuit Board	112
11 OSCAD on Aakash	113
11.1 Installation	114
11.2 Porting on Aakash	114
11.3 Usage	115

A Solved Examples	116
A.1 Diode	116
A.2 BJT	122
A.3 MOSFET	123
A.4 OP-AMP	125
A.5 CMOS	130
B Spoken Tutorial based SELF workshop on Oscad	142
B.1 What is SELF Workshop?	142
B.2 What do we offer in Oscad workshops?	142
B.3 What is expected from the organizing institute	143

Chapter 1

Introduction

Electronic systems are an integral part of human life. They have simplified our lives to a great extent. Starting from small systems made of a few discrete components to the present day integrated circuits (ICs) with millions of logic gates, electronic systems have undergone a sea change. As a result, design of electronic systems too have become extremely difficult and time consuming. Thanks to a host of computer aided design tools, we have been able to come up with quick and efficient designs. These tools are called **Electronic Design Automation** or **EDA** tools.

1.1 EDA design flow

Let us see the steps involved in EDA. In the first stage, the specifications of the system are laid out. These specifications are then converted to a design. The design could in the form of a circuit schematic, logical description using an HDL language etc. The design is then simulated and re-designed, if needed, to achieve the desired results. Once simulation achieves the specifications, the design is either converted to a PCB, a chip layout, or ported to an FPGA. The final product is again tested for specifications. The whole cycle is repeated until desired results are obtained [1].

1.2 EDA Tools

If you are building an electronic system, you would first design your circuit, make its schematic diagram, simulate it and finally convert it into a Printed Circuit Board (PCB) . There are various tools available that would help you do this. Some of the popular EDA tools are those of **Cadence**, **Synopsys**, **Mentor Graphics**, **Xilinx** etc. These are proprietary tools. There are some open source EDA tools like **gEDA**, **KiCad**, **Ngspice** etc.

1.3 What is Oscad?

Oscad is a free and open source EDA tool. It is an acronym for **O**pen source computer **a**ided **d**esign. Oscad is created using several open source software packages namely KiCad, Ngspice, Scilab and Python. Using Oscad, one can create circuit schematics, perform simulation and design PCB layouts. It can create or edit new device models, and create or edit subcircuits for simulation. It also has a Scilab based Mini Circuit Simulator (SMCSim) which is capable of giving the circuit equations for each simulation step. This feature is unique to Oscad.

1.4 Why Oscad?

Proprietary EDA tools are fairly comprehensive and high end. But their licences are very expensive. The main drawback of the available open source tools is that they are not comprehensive. Some of them are capable of PCB design (e.g., KiCad) while some of them are capable of performing simulations (e.g., gEDA). There is no well known open source software that can perform circuit design, layout design and circuit simulation together. Oscad is capable of doing all of the above. This is why Oscad is very important to students, teachers and other professionals who would want to study and/or design electronic systems. Oscad is also very useful for entrepreneurs and small scale enterprises who do not have the capability to invest in heavily priced proprietary tools.

1.5 Structure of the book

This book introduces Oscad to the reader and illustrates all the features of Oscad with examples. Chapter 2 gives step by step instructions to install Oscad on your computer and validate the installation. The software architecture of Oscad is presented in Chapter 3. Chapter 4 gets you started with Oscad. It takes you through a tour of Oscad with the help of a simple RC circuit example.

Chapter 5 explains how to create circuit schematics using Oscad, in detail using examples. Chapter 6 illustrates how to simulate circuits using Oscad. Chapter 7 explains PCB design using Oscad, in detail. The advanced features of Oscad like Model builder and subcircuit builder are covered in Chapter 8. Scilab based circuit simulator is covered in Chapter 8. Chapter 10 describes the spoken tutorials on Oscad and contains instructions to use them. Oscad is a very light weight software and it has been ported to Aakash tablet. Chapter 11 explains about Oscad on Aakash. Appendix A presents examples, that have been worked out using Oscad, from the book *Microelectronic Circuits* by Sedra and Smith [2]. Appendix B explains about Spoken Tutorial based SELF workshops on Oscad.

Chapter 2

Installing and Setting Up Oscad

The step by step instructions to install Oscad are given below.

Before starting the installation, please make sure that all the system and installation requirements and prerequisites are met. Installation script for Oscad is written in Bash. This makes the installation efficient and user friendly.

System Requirements

- Ubuntu 12.04 OS (64-bit/32-bit)
- Oscad
- Scilab 5.4.1

Installation Requirements

- Working Internet connection
- Require to be an admin user to do the installation

Prerequisites

- Basic knowledge of analog and digital electronics
- Basic knowledge of Linux shell commands
- Requires Synaptic Package Manager

Note: The Linux commands typed on Terminal during installation are given in boxes with round corners

2.1 Procedure for installing Oscad

- **Step 1: Download Oscad and examples**

Go to <http://www.oscad.net/downloads>. Figure 2.1 shows the downloads page of Oscad website.

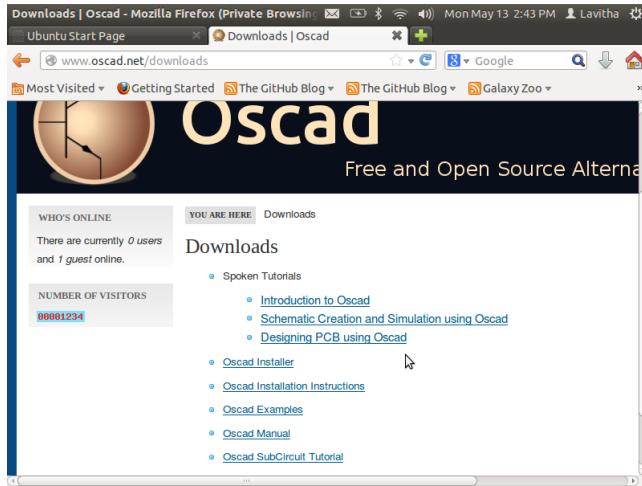


Figure 2.1: Oscad Website

1. Click on **Oscad Installer** and save the file in a folder
2. Click on **Oscad Examples** and save the file in the same folder as above

- **Step 2: Download Scilab**

Go to <http://www.scilab.org/>. Click on the **Download Scilab** option on the home page. Save the file in the same folder where Oscad installer and Examples were saved. Figure 2.2 shows the Download Scilab option in the Scilab webpage.

Note: You can skip this step, if you have scilab 5.4.0 or above in your system



Figure 2.2: Scilab website

- **Step 3: Extract the downloaded files**

Go to the folder where all the three files are saved. Select all, right click and choose **extract here** as shown in Figure 2.3.

- **Step 4: Navigate to Folder**

Go to the directory where all the three folders are saved and extracted. For this open a terminal window and type:

`cd <<folder-where-downloaded-files-are-saved>>`

In the above command, replace *folder – where – downloaded – files – are – saved* with the path of the folder where you have saved and extracted the downloaded files. Press Enter.

Now check whether we have navigted to desired folder. To check please type:

`ls`

Press Enter

Screen shot 2.4 shows navigation to the folder having all files.

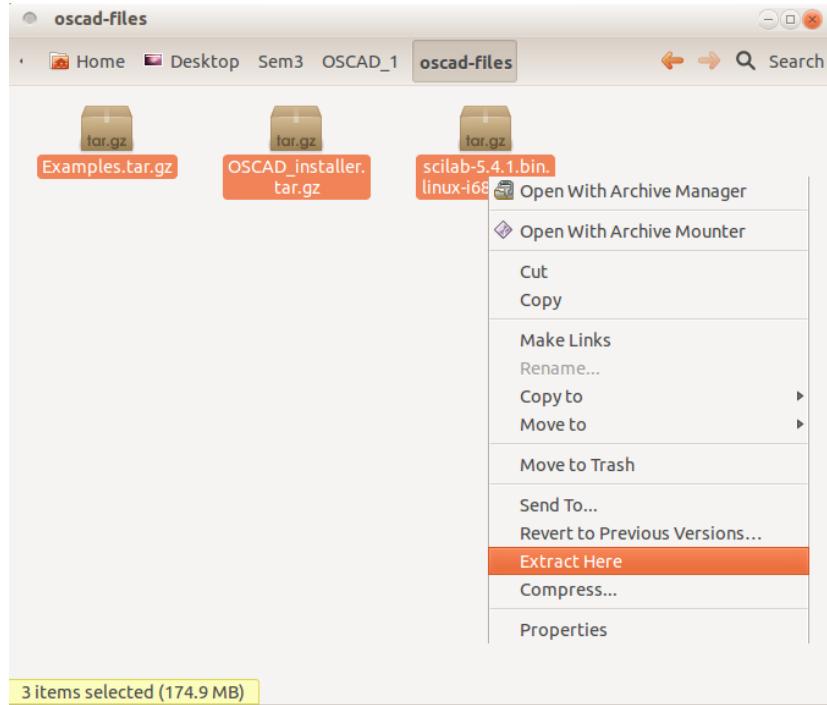


Figure 2.3: Extracting all files

```
lavitha@fossee:~/Downloads$ cd Downloads
lavitha@fossee:~/Downloads$ ls
a.txt scilab-5.4.1 tbc3.sh~ tbc6.sh~ tbc9.sh~
Examples tbc12.sh~ tbc4.sh~ tbc7.sh~ Untitled Document 1~
OSCAD installer tbc2.sh~ tbc5.sh~ tbc8.sh~
lavitha@fossee:~/Downloads$
```

Figure 2.4: Terminal: Navigation to folder with required files

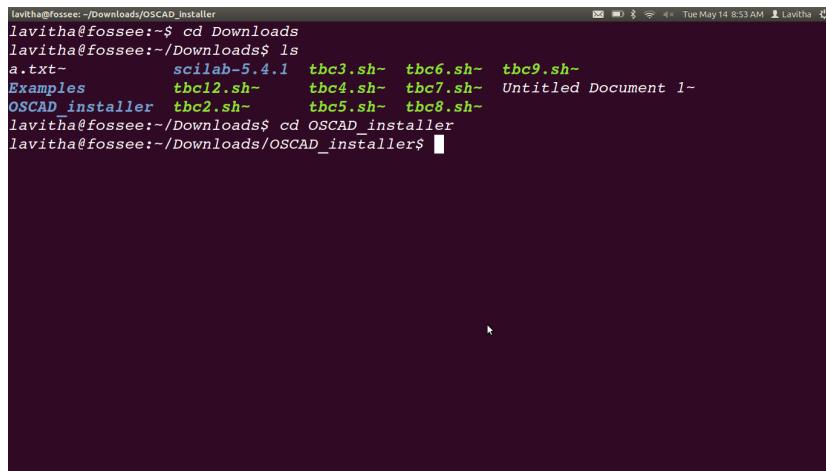
- **Step 5: Navigate to OSCAD_Installer**

Now navigate to the folder `OSCAD_installer`. To do this type:

```
cd OSCAD_installer
```

Press Enter

Figure 2.5 shows that we have navigated to OSCAD_installer folder.



A screenshot of a terminal window titled 'Lavitha'. The terminal shows the following command sequence:

```
lavitha@fossee:~/Downloads$ cd Downloads
lavitha@fossee:~/Downloads$ ls
a.txt scilab-5.4.1 tbc3.sh~ tbc6.sh~ tbc9.sh~
Examples tbc12.sh~ tbc4.sh~ tbc7.sh~ Untitled Document 1-
OSCAD_installer tbc2.sh~ tbc5.sh~ tbc8.sh~
lavitha@fossee:~/Downloads$ cd OSCAD_installer
lavitha@fossee:~/Downloads/OSCAD_installer$
```

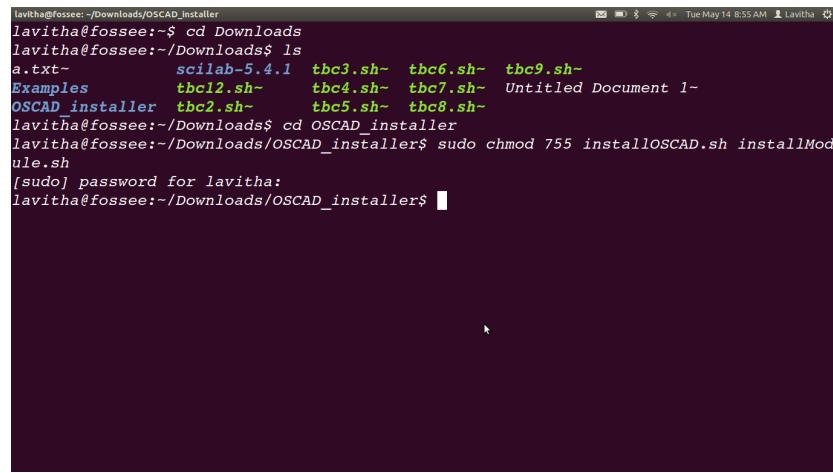
Figure 2.5: Terminal: navigation to OSCAD_installer

- **Step 6: Make the installOSCAD and installModule files executable**

To do this go to terminal and Type:

```
sudo chmod 755 installOSCAD.sh installModule.sh
```

Press Enter. Type the sudo (root) password. The Terminal should look like as shown in figure 2.6.



```
lavitha@fossee:~/Downloads/OSCAD_installer
lavitha@fossee:~/Downloads$ ls
a.txt scilab-5.4.1 tbc3.sh- tbc6.sh- tbc9.sh-
Examples tbc12.sh- tbc4.sh- tbc7.sh- Untitled Document 1-
OSCAD_installer tbc2.sh- tbc5.sh- tbc8.sh-
lavitha@fossee:~/Downloads$ cd OSCAD_installer
lavitha@fossee:~/Downloads/OSCAD_installer$ sudo chmod 755 installOSCAD.sh installModule.sh
[sudo] password for lavitha:
lavitha@fossee:~/Downloads/OSCAD_installer$
```

Figure 2.6: Terminal: make files executable

- **Step 7: Begin installation** To begin installation, type

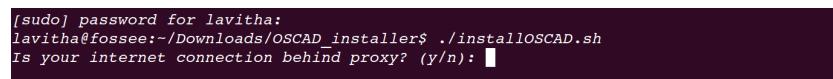
./installOscad.sh

Press Enter

The Terminal prompts for proxy settings as shown in Figure 2.7

Note: If you are not behind proxy type 'n' and press Enter.

You will not get the option to enter proxy parameters.



```
[sudo] password for lavitha:
lavitha@fossee:~/Downloads/OSCAD_installer$ ./installOSCAD.sh
Is your internet connection behind proxy? (y/n):
```

Figure 2.7: Terminal: Installation - Proxy settings

If you typed ‘y’, then enter the proxy details as shown in figure 2.8

```
lavitha@fossee:~/Downloads/OSCAD_installer$ ./installOSCAD.sh
Is your internet connection behind proxy? (y/n): y
Proxy Hostname :netmon.iitb.ac.in
Proxy Port :80
username@netmon.iitb.ac.in:80 :manasi_ghadi
Password :■
```

Figure 2.8: Terminal: Proxy

Now the prompt displays message **Do you want to continue [Y/n]:**
Type ‘y’ and press Enter. KiCad, Ngspice and necessary python modules will be installed.

Note: While installing the python modules, you may get some error messages

In that case, install the missing packages using Synaptic Package Manager

Once this is done, re-run the installOscad.sh script.

- **Step 8: Linking of Scilab** The prompt displays the message: **Do you have scilab 5.4 or above? (y/n)** as shown in figure 2.9

```
0 upgraded, 0 newly installed, 0 to remove and 3 not upgraded.
Checking python Modules.....  

Found python module: wx  

Found python module: re  

Found python module: Image  

Found python module: ImageTk  

Found python module: string  

Found python module: Tkinter  

Found python module: Pmw  

All python modules are available
Checking scilab .....,  

Require scilab version 5.4 or above
Do you have scilab5.4 or above? (y/n) ■
```

Figure 2.9: Terminal: Linking Scilab

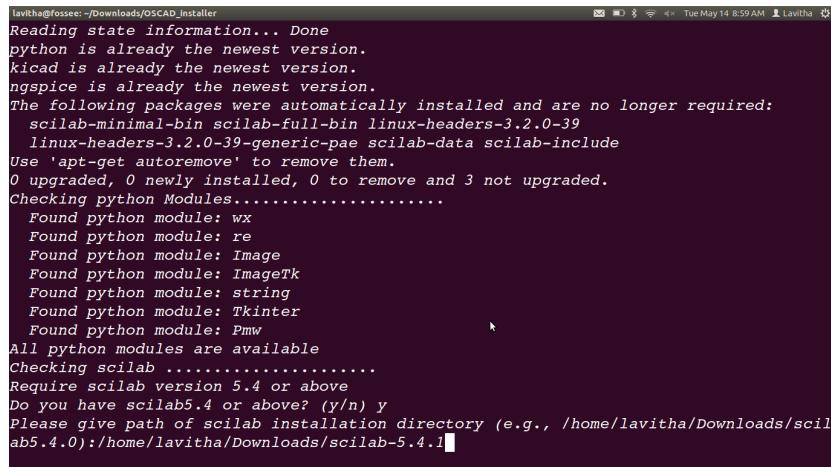
Type 'y' and press Enter. Now give the complete path where you have saved Scilab 5.4.0 or above.

Example:

```
/home/lavitha/downloads/scilab-5.4.1
```

Press Enter. The screen shot is shown in figure 2.10

Note: The metanet library will be installed after this step. It takes a couple of minutes



```
lavitha@tosse:~/Downloads/OSCAD_installer
Reading state information... Done
python is already the newest version.
kicad is already the newest version.
ngspice is already the newest version.
The following packages were automatically installed and are no longer required:
  scilab-minimal-bin scilab-full-bin linux-headers-3.2.0-39
    linux-headers-3.2.0-39-generic-pae scilab-data scilab/include
Use 'apt-get autoremove' to remove them.
0 upgraded, 0 newly installed, 0 to remove and 3 not upgraded.
Checking python Modules.....
  Found python module: wx
  Found python module: re
  Found python module: Image
  Found python module: ImageTk
  Found python module: string
  Found python module: Tkinter
  Found python module: Pmw
All python modules are available
Checking scilab .....
Require scilab version 5.4 or above
Do you have scilab5.4 or above? (y/n) y
Please give path of scilab installation directory (e.g., /home/lavitha/Downloads/scilab5.4.0):/home/lavitha/Downloads/scilab-5.4.1
```

Figure 2.10: Terminal: Scilab instllation

- **Step 9: Final installation** The prompt displays the message **Please select installation directory**

Type the desired location where you want to install Oscad.

For example:

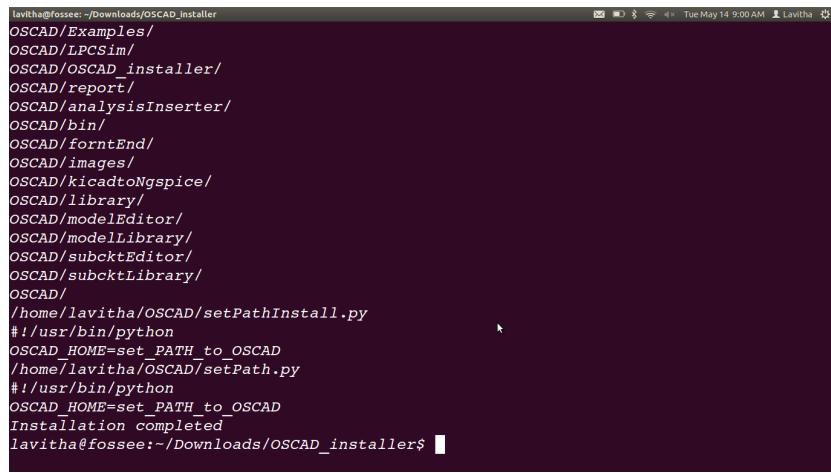
```
/home/lavitha
```

Press Enter

If everything is installed then the message **installation completed** is displayed as shown in screen shot 2.11. This creates **Oscad** shortcut on Desktop.

2.2 Testing

Let us now verify the installation of Oscad.



```
lavitha@fossee:~/Downloads/OSCAD_installer$ OSCAD/Examples/
OSCAD/LPCSim/
OSCAD/OSCAD_installer/
OSCAD/report/
OSCAD/analysisInserter/
OSCAD/bin/
OSCAD/frontEnd/
OSCAD/images/
OSCAD/kicadtoNgspice/
OSCAD/library/
OSCAD/modelEditor/
OSCAD/modelLibrary/
OSCAD/subcktEditor/
OSCAD/subcktLibrary/
OSCAD/
/home/lavitha/OSCAD/setPathInstall.py
#!/usr/bin/python
OSCAD_HOME=set PATH_to OSCAD
/home/lavitha/OSCAD/setPath.py
#!/usr/bin/python
OSCAD_HOME=set PATH_to OSCAD
Installation completed
lavitha@fossee:~/Downloads/OSCAD_installer$
```

Figure 2.11: Terminal: Installation Completed

- Double click on Oscad shortcut created on Desktop as shown in figure 2.12



Figure 2.12: Desktop: Oscad shortcut

- A window is displayed as shown in figure 2.13. Select the option **Run**

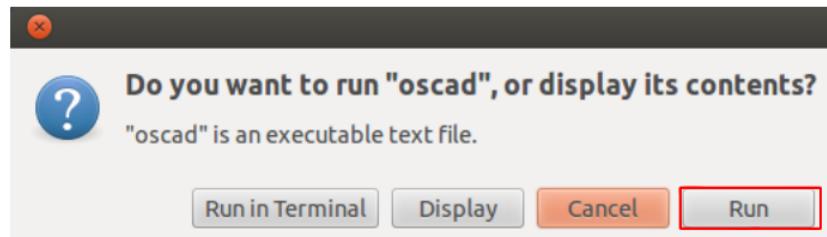


Figure 2.13: Desktop: Oscad - Choose the option ‘run’

- It opens the Oscad window as shown in figure 2.14

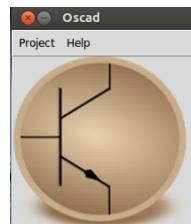


Figure 2.14: Desktop: Oscad

- Select **project** tab at the top left hand corner of Oscad window and then click on **open**

Browse to the folder where Examples are saved as shown in figure 2.15

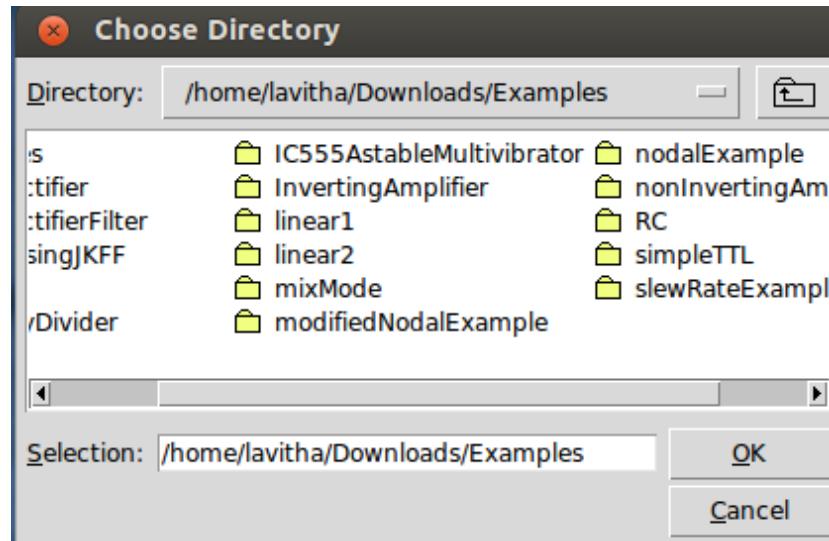


Figure 2.15: Browse to the folder where Oscad Examples are saved

- Select **RC** from the examples by double clicking on RC and then click on **OK**

- Enter Project name dialog box opens up as shown in figure 2.16. Click on **OK**

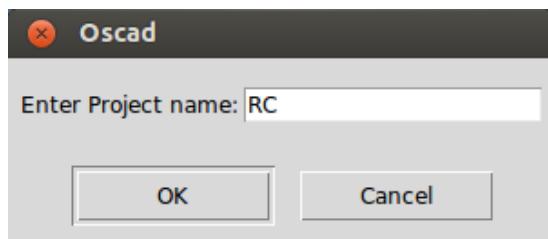


Figure 2.16: Desktop: Oscad

- A tool bar appears as in figure 2.17. Select **Schematic Editor** in the tool bar.

- An error message as shown in figure 2.18 will pop up. Click on **Close**.



Figure 2.17: Desktop: Oscad Toolbar - Schematic editor

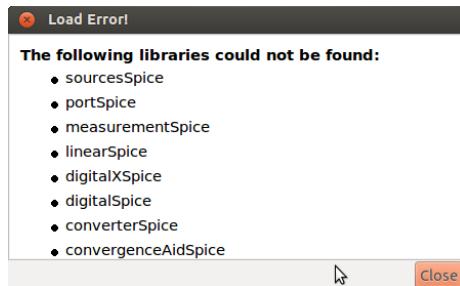


Figure 2.18: Schematic editor Load error

- Scematic Editor window appears. Press **F1** to zoom in and press **F2** to zoom out. RC filter circuit seen on schematic editor window is as shown in Figure 2.19.
- Close Schematic Editor
- Click on Ngspice from Oscad Tool bar as shown in Figure 2.20. This will simulate the netlist using Ngspice.
- The graph and terminal appears as shown in figure 2.21. This shows the result of transient analysis of RC circuit and verifies installation.

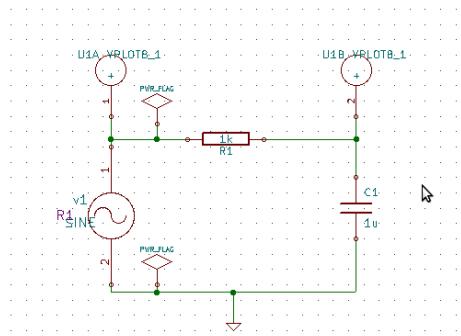


Figure 2.19: Schematic of RC circuit



Figure 2.20: Desktop: Oscad Toolbar - Ngspice

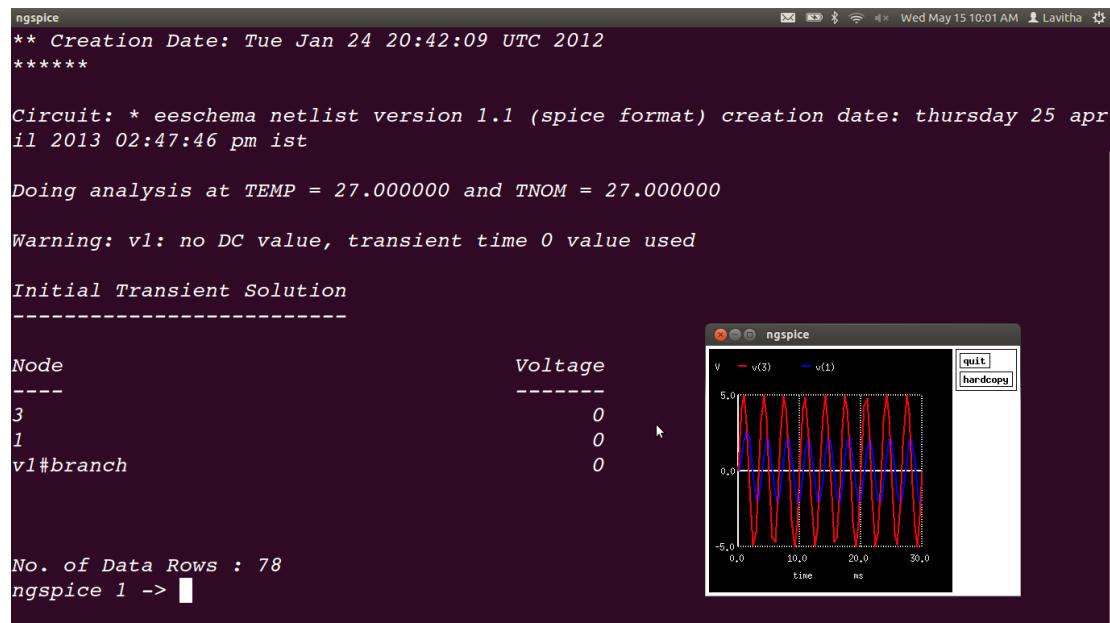


Figure 2.21: Ngspice simulation output

Chapter 3

Architecture Of Oscad

Oscad is a CAD tool which provides the ease of testing circuits for electronic system designers. But the important feature of this tool is that it is Open Source and hence the user can check, modify the source as per their need. The software provides a generic, modular and extensible platform for experiment with electronic circuits. This software runs on all UNIX platforms and uses `python`, `KiCad`, `Ngspice` and `Scilab` 5.4 or above.

The objective behind the development of Oscad is to provide an open source solution for electronics and electrical engineers. The software should be capable of performing Circuit Schematic, PCB Design and Circuit Simulation (Analog, Digital and Mixed signal). It should provide facilities to create new models and components. In addition to this, it should have the capability to explain the circuit by giving symbolic equations and numerical values. The architecture of Oscad has been designed keeping these objectives in mind.

The broad objectives of Oscad are:

- Integrate existing open source tools for Circuit Design and drawing, PCB Layout , Circuit Simulation.
- Create additional modules if required.
- Create user friendly graphical interface.

3.1 Modules used in Oscad

Various open-source tools have been used for the underlying build-up of Oscad. In this section we will give a brief idea about all packages.

3.1.1 Eeschema – Schematic Editor

Eeschema is an integrated software where all functions of circuit drawing, control, layout, library management and access to the PCB design software are carried out within itself [3]. Eeschema is intended to work with printed circuit

software such as Pcbnew. It can provide the netlist file, which describes the electrical connections of the PCB. Eeschema also integrates a component editor which allows the creation, editing and visualization of components. It also allows the user to effectively handle of the symbol libraries i.e; import, export, addition and deletion of library components.

Eeschema also integrates the following additional but essential functions needed for a modern schematic capture software:

- Design rules check (DRC) for the automatic control of incorrect connections and inputs of components left unconnected.
- Generation of layout files in POSTSCRIPT indexPOSTSCRIPTor HPGL format.
- Generation of layout files printable via printer.
- Bill of Material generation.
- Netlist generation for PCB layout or for simulation.

3.1.2 CvPcb – Component-Footprint mapper

CvPcb is a tool that allows user to associate components in the schematic to component footprints used when laying out the printed circuit board [3]. Typically the netlist file generated by Eeschema does not specify which printed circuit board footprint is associated with each component in the schematic. Although this is not always the case as component footprints can be associated during schematic capture by setting the component's footprint field. CvPcb provides a convenient method of associating footprints to components. It provides footprint list filtering, footprint viewing, and 3D component model viewing to help ensure the correct footprint is associated to each component. Components can be assigned to their corresponding footprints manually or automatically by creating equivalence files. Equivalence files are look up tables associating each component with it's footprint. This interactive approach is simpler and less error prone than directly associating the footprints in the schematic editor because as well as allowing for automatic association, CvPcb allows you to see the list of available footprints available and display them on the screen to ensure you are associating the correct footprint.

3.1.3 Pcbnew – PCB Layout Editor

Pcbnew is a powerful printed circuit board software tool [3]. It is used in association with the schematic capture software program Eeschema, which provides the netlist file - this describes the electrical connections of the PCB to design. CvPcb is used to assign each component in the Netlist produced by Eeschema, to a module that is used by Pcbnew.

- It manages libraries of modules. Each module is a drawing of the physical component including its footprint - the layout of pads providing connections to the component. The required modules are automatically loaded during the reading of the netlist produced by CvPcb.
- Pcbnew integrates automatically and immediately any circuit modification, by removal of any erroneous tracks, addition of the new components, or by modifying any value (and under certain conditions any reference) of the old or new modules, according to the electrical connections appearing in the scheme.
- This tool provides a rats nest display, a hairline connecting the pads of modules which are connected on the schematic. These connections move dynamically as track and module movements are made.
- It has an active Design Rules Check (DRC) which automatically indicates any error of track layout in real time.
- It can automatically generates a copper plane, with or without thermal breaks on the pads.
- It has a simple but effective autorouter to assist in the production of the circuit. An Export/Import in SPECCTRA dsn format allows to use more advanced auto-routers.
- This provides options specifically for the production of ultra high frequency circuits (such as pads of trapezoidal and complex form, automatic layout of coils on the printed circuit).
- Pcbnew displays the elements (tracks, pads, texts, drawings and more) as actual size and according to personal preferences:
 - display in full or outline.
 - display of the track/pad clearance.

3.1.4 Analysis Inserter

This feature helps the user to perform different type of analysis such as Operating point analysis , DC analysis , AC analysis , transient analysis etc. It has the facility to

- Insert type of analysis
- Option of analysis
- Option of simulator

3.1.5 KiCad to Ngspice netlist converter

It converts KiCad generated netlists to Ngspice compatible format. It has the capability to

- Insert parameters for fictitious components
- Convert IC into discrete blocks
- Insert D-A and A-D converter at appropriate place
- Insert plotting and printing statement in netlist
- Find current through all components

3.1.6 Component Model Builder

- This tool provides a facility to define a new model for devices such as
 - Diode
 - Bipolar Junction Transistor (BJT)
 - Metal Oxide Semiconductor (MOS)
 - Junction Field Effect Transistor (JFET)
 - IGBT
 - Magnetic core
- Provides facility to edit existing model.
- Provides help related to model parameter.

3.1.7 Component Sub-circuit Builder

This feature allows the user to create a sub-circuit for a component. Once the sub-circuit for a component is created, the user can use it for different circuits. It has the facility of

- Provides facility to define new components such as
 - Op-amp
 - IC-555
- Provides facility to edit existing sub-circuit
- Provides help related to components parameters

3.1.8 Circuit Simulator – Ngspice

Ngspice is a general-purpose circuit simulation program for nonlinear dc, non-linear transient, and linear ac analyses [4]. Circuits may contain resistors, capacitors, inductors, mutual inductors, independent voltage and current sources, four types of dependent sources, lossless and lossy transmission lines (two separate implementations), switches, uniform distributed RC lines, and the five most common semiconductor devices: diodes , BJTs , JFETs , MES-FETs , and MOSFET .

3.1.9 Scilab based circuit simulator – SMCSim

The feature of SMCSim is that it gives the system of equations for the circuit under test. The SMCSim works in three modes: normal, symbolic and numerical mode. This is covered in more detail in Chapter 9.

3.2 Workflow of Oscad

Figure 3.1 shows a block diagram of Oscad. The block diagram consists mainly three parts:

- Schematic Editor
- PCB Layout Editor
- Circuit Simulators

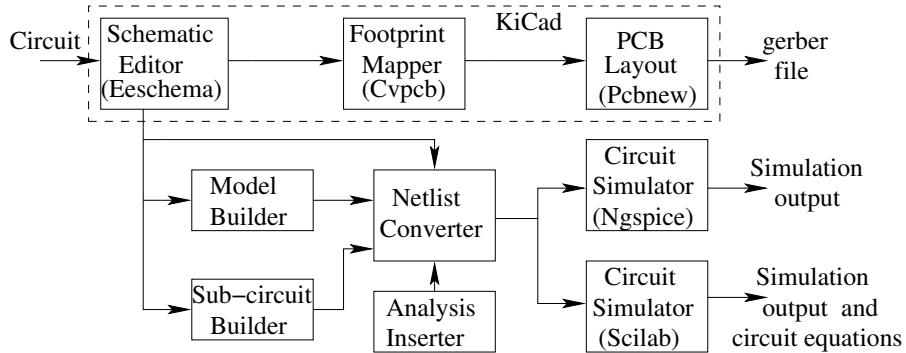


Figure 3.1: Block Diagram of Oscad

Here, we explain the functionality of each block to design electronic systems.

The circuit design is the first step in the design of an electronic circuit. Generally a circuit diagram is drawn on a paper, and then entered into a computer using a schematic editor. Eeschema is the schematic editor for Oscad. Thus all the functionalities of Eeschema are naturally available in Oscad.

Using the functionalities of Eeschema, separate libraries for components explicitly or implicitly supported by Ngspice have been created. As Eeschema is originally intended for PCB Design, there are no fictitious components such as voltage or current sources. Thus, a new library for different types of voltage and current sources such as sine, pulse, square wave, etc. have been added in Oscad. A library which gives the functionality of printing and plotting has also been built.

The schematic editor provides a netlist file, which describes the electrical connections of the design. In order to create a PCB layout, physical components are required to be mapped into their footprints. To create component to footprint mapping, CvPcb is used. It is a part of KiCad. Footprints have been created for the components in the newly created libraries. Pcbnew is used to draw a PCB layout.

After designing a circuit, it is essential to check the integrity of the circuit design. In case of large size electronic circuits, breadboard testing is impractical. Therefore electronic system designers rely heavily on simulation.

The accuracy of the simulation results can be increased by accurate modeling of the circuit elements. Model Builder, provides a facility to define a new model for devices and edit existing models. Complex circuit elements can be created by hierarchical modeling. Sub-circuit Builder provides an easy way to create a sub-circuit. The schematic editor provides a netlist file, which describes the electrical connections between circuit components. But it cannot be directly used for simulation due to compatibility issues. Netlist Converter converts the netlist into an Ngspice compatible netlist. The type of simulation to be performed using the netlist and the corresponding options are provided through a graphical user interface (GUI). This is called Analysis Inserter in Oscad.

Oscad uses Ngspice for mixed-level/mixed-signal circuit simulation. Ngspice is based on three open source software packages [5]:

- Spice3f5 (analog circuit simulator)
- Cider1b1 (couples Spice3f5 circuit simulator to DSIM device simulator)
- Xspice (code modeling support and simulation of digital components through an event driven algorithm)

It is a part of gEDA project. Ngspice is capable of simulating devices with BSIM , EKV , HICUM , HiSim , PSP , PTM models. It is widely used due to its accuracy even for latest technology devices.

In order to provide an explanation capability, Oscad also has a Scilab based circuit simulation capability. It generates equations from the netlist and gets them solved by Scilab, which has many state of the art numerical methods built in. This tool is called Scilab based Mini Circuit Simulator (SMCSim) in Oscad.

Chapter 4

Getting Started

In this chapter we will get started with Oscad. We will run through the various options available with an example circuit. Referring to this chapter will make you familiarize with Oscad thereby helping you plan your project before actually designing a circuit. Lets get started.

After you finish installing Oscad, a shortcut link for Oscad will be created on your desktop. You will click on this link to launch oscad. Oscad main window will open up. It is shown in Figure 4.1. On the menu bar there are two options, **Project** and **Help**. To create a new project or open an existing project, use the project option.

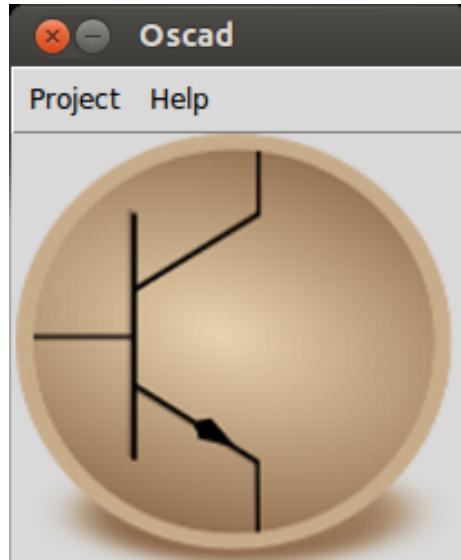


Figure 4.1: Oscad main window

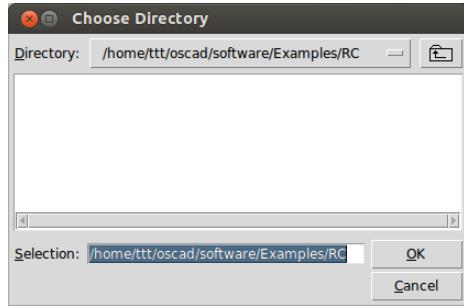


Figure 4.2: Open project directory

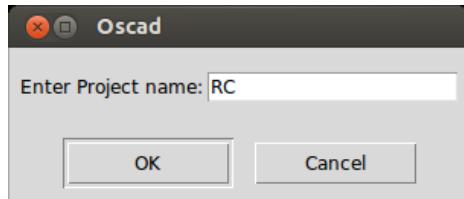


Figure 4.3: 'Enter Project Name' window

Let us open an existing project. Click on Project and select Open. The **Choose Directory** window opens up. This window is shown in figure 4.2. Choose the corresponding directory of the project. Choose RC example from the Examples folder that you have downloaded from the Oscad webpage. Click on OK. Another window will open asking to enter the Project Name. This window is illustrated in figure 4.3. Since we are opening an already existing project, the name will appear in the text box automatically. In case you are creating a new project then you have to write the name of your new project in this window. Click on OK after you finish doing it.

After you finish creating a new project or opening a new project, a vertical tool bar will appear. This tool bar is shown in figure 4.4. This is the **Oscad tool bar**. It contains 9 tools. These tools have images depicting their purpose. If you place your mouse pointer on these tools, the name of the tool appears at the bottom of the mouse pointer. Following is the list of tools, from top to bottom as they appear, in the Oscad tool bar.

1. Schematic Editor
2. Analysis Inserter
3. Netlist Converter
4. Ngspice



Figure 4.4: Oscad Tool bar

5. Footprint editor
6. Layout Editor
7. SMCSim
8. Model builder
9. Subcircuit builder

4.1 Schematic Editor

Click on the first tool on the tool bar i.e. Schematic Editor. Doing so will open EESchema, the schematic editor used in Oscad. If you had started a new

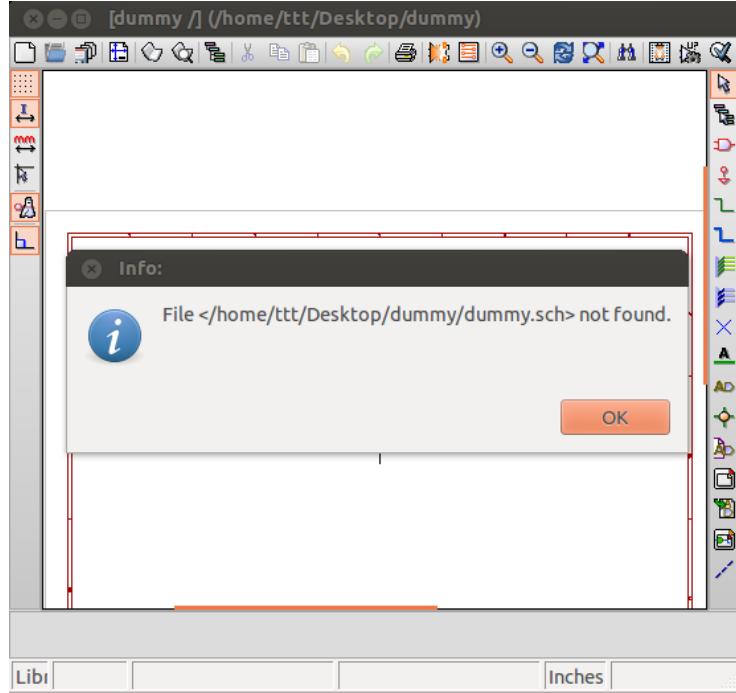


Figure 4.5: Schematic Editor (opening a new project)

project, you will get the schematic editor window with an info dialog box. This is illustrated in figure 4.5. This warning can be safely ignored by clicking on OK.

However, if you had opened an already existing project, you will get the schematic editor window along with a Load error. This is illustrated in figure 4.6. This error occurs because the schematic that you have opened has not been loaded with the libraries mentioned in the Load Error message. Close the Load Error message by clicking on the close button. The RC circuit diagram opens up as shown in figure 4.7. You are now ready to start creating/editing the circuit schematic. To know how to use the schematic editor to create circuit schematics, refer to Chapter 5.

4.2 Analysis Inserter

The second tool on the tool bar is the Analysis Inserter. It is necessary that before you use this tool, you have the spice netlist file (.cir) created. This is because this tool is used to insert analysis commands to the spice netlist file. To know how to generate the spice netlist file, refer to the section 5.4.5.

When you click on this tool, a window named `kicad Ngspice` will open. This is the Analysis inserter GUI. It is shown in figure 4.8. This window will

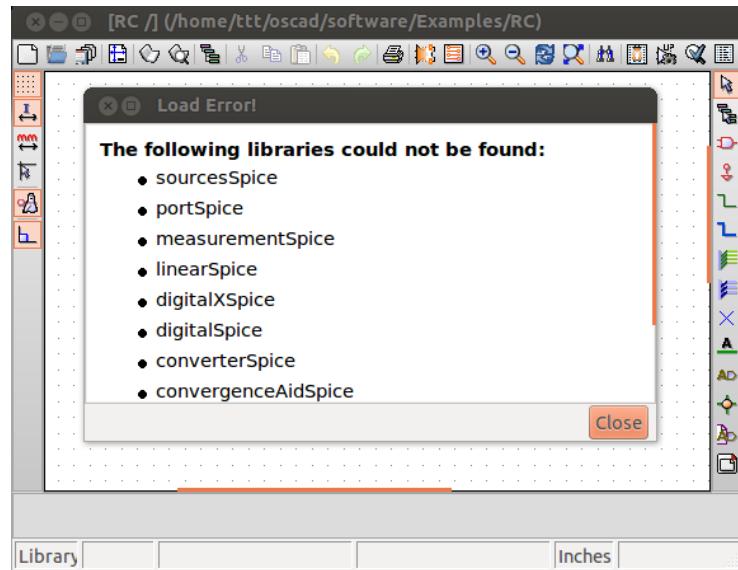


Figure 4.6: Schematic Editor (opening an already existing project)

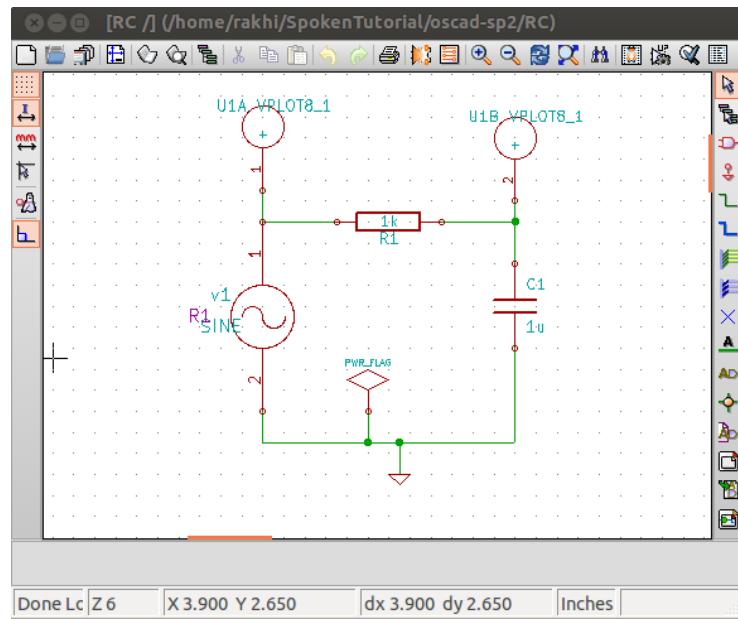


Figure 4.7: Schematic Editor (with RC circuit)

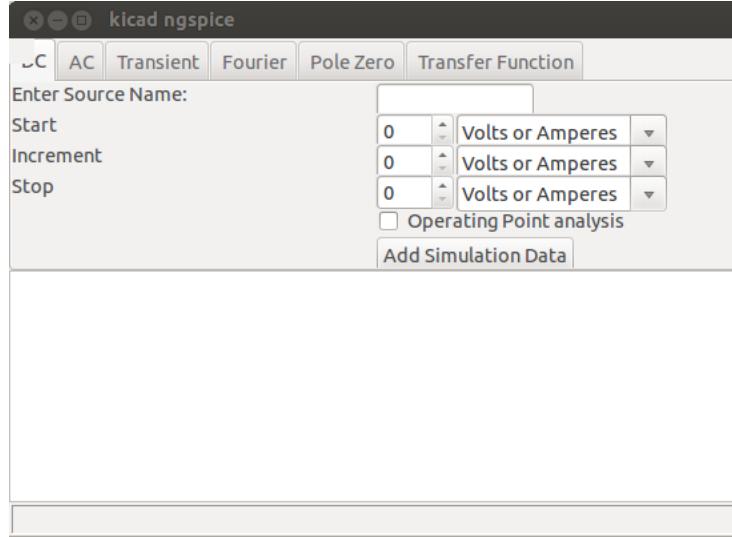


Figure 4.8: Analysis Inserter

allow the analysis commands to be inserted in to the spice netlist file. The use of analysis inserter is explained in detail in section 6.1.

4.3 Netlist Converter

The third tool on the tool bar is the **Netlist Converter**. Before you use this tool, you should have already created the spice netlist file and used analysis inserter to generate analysis commands. To know how to generate spice netlist, refer to the section 5.4.5. This file is not directly usable for simulation. In other words, it is not compatible with Ngspice.

The spice netlist file contains only the component placement information and it says nothing about the magnitude and other parameters (if applicable) of the source components like voltage source, current source etc. When you click on the **Netlist Converter** tool, a terminal window will open up as shown in figure 4.9. If you observe this terminal window, you will notice that it first tells you which .cir file it is referring to. It also asks for the source value. It may ask other parameters too depending upon the type of sources used. Once you are done entering the values, press the ‘Enter’ key. It will generate .cir.out and .cir.ckt files in the same project directory.

```

Terminal
=====
Kicad to Ngspice netlist converter
=====
converting RC.cir
.tran 5e-03 30e-03 0e-00
-----
Add parameters for sine source v1
Enter offset value (Volts/Amps): 
```

Figure 4.9: Netlist Converter

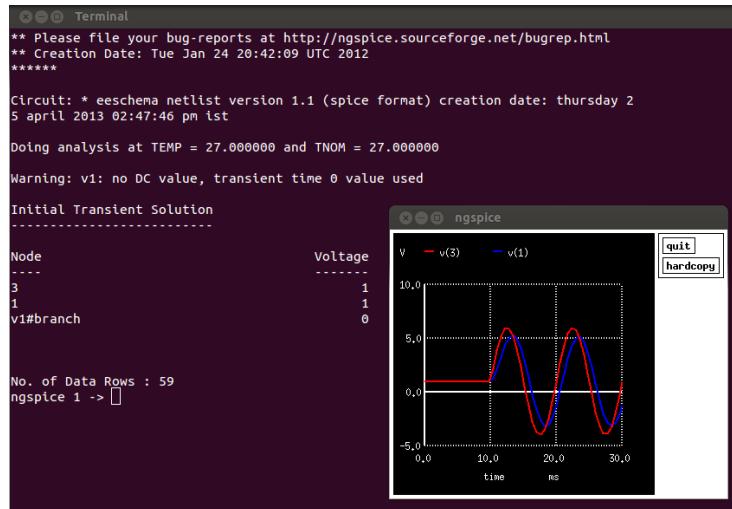


Figure 4.10: Ngspice Simulation

4.4 Ngspice

The sections 4.2 and 4.3 helped to generate a netlist suitable to be simulated using Ngspice. Clicking on the tool **Ngspice** will open a terminal window and a plot window as shown in figure 4.10. You should have the converted netlist file ‘.cir.out’ before using this tool.

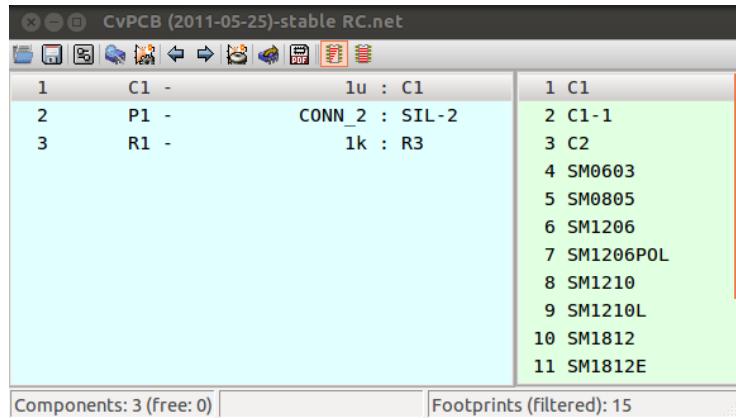


Figure 4.11: CvPcb window

4.5 Footprint Editor

Clicking on the **Footprint Editor** tool will open the CvPcb window. This window will ideally open the .net file for the current project. So for using this tool, you should have the netlist for PCB design (a .net file). To know more about how to create netlist for PCB, refer to the section 7.1.1.

On clicking the Footprint editor tool, we see the corresponding .net file for RC circuit. This window is shown in figure 4.11.

The main purpose of this window is to let you choose the footprints for the various components of your circuit. Let us view the footprint C1 for capacitor C1. Click on C1 from the right hand side of CvPcb window,. Click on **View Selected Footprint** tool from the tool bar of CvPcb window,. This will show the footprint corresponding to C1. This is illustrated in Figure 4.12. To know more about how to assign a footprints to components, see Chapter 7.

4.6 Layout Editor

Clicking on the **Layout Editor** tool will open Pcbnew, the layout editor used in Ocad. In this window, you will create the PCB. It involves laying tracks and vias, performing optimum routing of tracks, creating more than one copper layer for PCB etc. An already made PCB design for RC circuit is shown in figure 4.13. This is how the PCB will look like when you actually print it on a copper-clad board. When you save this design, it will be saved as a .brd file in the same directory. Chapter 7 explains how to use the Layout Editor to design a PCB.

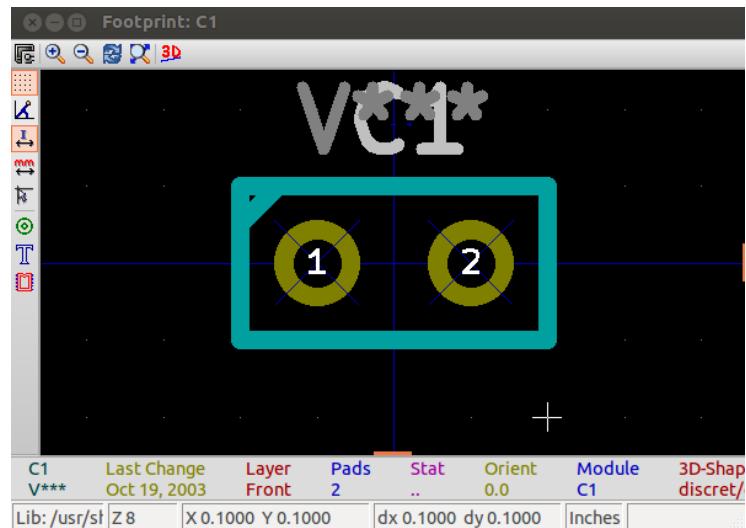


Figure 4.12: Footprint for C1

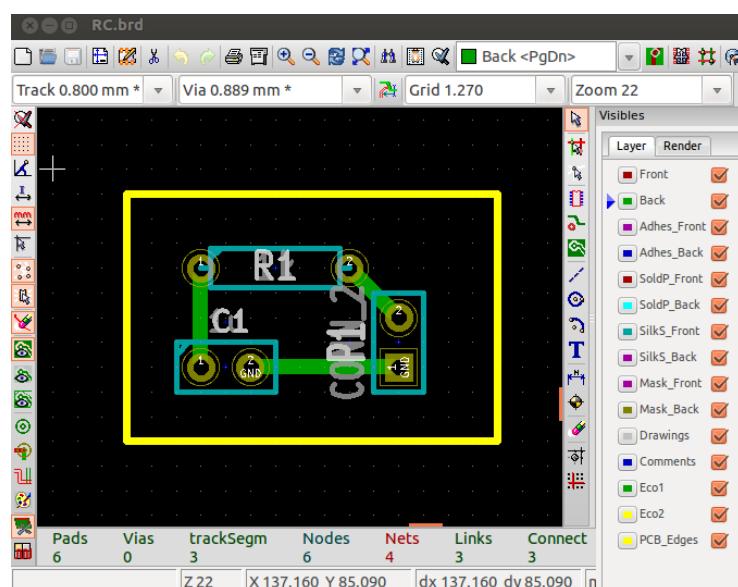


Figure 4.13: PCB design for RC circuit

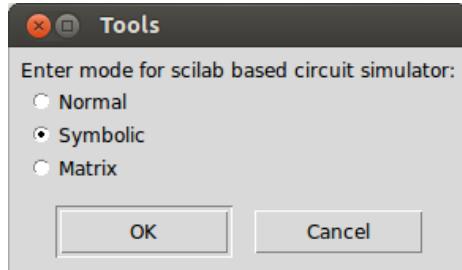


Figure 4.14: Window to choose the mode for Scilab based circuit simulator

4.7 SMCSim

SMCSim stands for Scilab Based Mini Circuit Simulator. This tool will generate mathematical equations for your circuit, thereby helping you gain a better understanding of your circuit. Oscad uses Scilab for this purpose [6]. Clicking on this tool will open a small window where you will have to choose between three options.

1. Normal
2. Symbolic
3. Matrix

These are basically the modes used for circuit simulation. This window is illustrated in figure 4.14. After you select one of them and click ok, Scilab will be launched automatically. Let us choose **Symbolic**. The scilab console will show the set of equations for the circuit as shown in Figure 4.15. A plot window will also open showing the plot of variables of the circuit. Which variables to plot depends upon the placement of plot components in the circuit schematic. The plot window is shown in figure 4.16. To know more about how to use this feature, refer Chapter 9.

4.8 Model builder

Oscad also gives you an option to re-configure the model of a component. It facilitates the user to change the model of components such as diode, transistor, MOSFET etc. When you click on the **Model builder** tool, you will get the window as shown in figure 4.17. You see a blank window because the RC circuit which you have opened does not have any component whose model can be edited. If suppose you had a diode (say 1N4007) in the circuit then the model builder window would have looked like as shown in figure 4.18. We can see that it shows **1n4007** in the window. After choosing **1n4007** and clicking on **OK**, it will confirm saying **Do you want to edit?**. After clicking on **OK**, a window will

4.1 Console

File Edit Control Applications ?

File Browser Scilab 5.4.1 Console

Name

RC

- RC-cache.bak
- RC-cache.lib
- RC.bak
- RC.cir
- RC.cir.ckt
- RC.cir.ckt.sol
- RC.cir.out
- RC.lst
- RC.pro
- RC.proj
- RC.sch
- analysis

Startup execution:
loading initial environment

Start Metanet:
Load macros
Load gateways
Load jar
Load help
Load demos

Simulation of RC.cir.ckt:

System of Equations representing the electrical circuit:

```
(R1)v_1 + (-R1)v_2 + i_V1 = 0
(-R1)v_1 + (R1)v_2 + (C1)dv_2/dt = 0
v_1 = V1
```

Operating Point (DC) Analysis:
All capacitors are open circuited and inductors are short circuited

System of Equations representing the electrical circuit:

```
(R1)v_1 + (-R1)v_2 + i_V1 = 0
(-R1)v_1 + (R1)v_2 = 0
v_1 = V1
```

The complete solution (Operating Point) of the circuit
is written in RC.cir.ckt.sol

Transient Analysis:

System of Equations representing the electrical circuit:

Variable Browser

Name	Dimens...	Type	Visibility
xaxis	1x1	String	local
x	3x1	Double	local
t	1x1	Double	local
i	1x1	Double	local
t_itr	1x1	Double	local
UIC	1x1	Double	local
t_step	1x1	Double	local
t_end	1x1	Double	local
t_start	1x1	Double	local
Asymb	3x3	String	local
Bsymbol	3x1	String	local
isweepA_	N/A/Unknown	local	
vPrintAr_	N/A/Unknown	local	
vPlotArray_	N/A/Unknown	local	
iPrintArr...	N/A/Unknown	local	
iPlotArray_	N/A/Unknown	local	
Wmode	1x1	String	local
Wfname	1x1	String	local
Current	4x1	Double	local

Command History

```
plot2d
// -17/04/2013 12:58:42 // ...
// -17/04/2013 13:07:49 // ...
// -17/04/2013 13:09:13 // ...
// -17/04/2013 13:10:17 // ...
// -17/04/2013 13:11:21 // ...
// -23/04/2013 15:05:46 // ...
exec discretisation.sce
g
// -25/04/2013 12:55:35 // ...
// -25/04/2013 13:02:40 // ...
// -25/04/2013 13:09:53 // ...
// -16/05/2013 13:51:45 // ...
// -16/05/2013 13:46:05 // ...
// -16/05/2013 13:49:05 // ...
// -16/05/2013 13:50:05 // ...
// -16/05/2013 13:51:45 // ...
// -16/05/2013 13:52:21 // ...
// -16/05/2013 13:54:15 // ...
// -16/05/2013 13:56:19 // ...
// -16/05/2013 13:58:52 // ...
```

File/directory Filter Case sensitive Regular expression

Figure 4.15: Scilab Window showing system of equations representing the electrical circuit

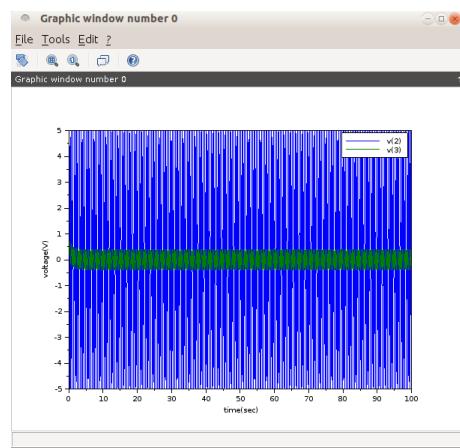


Figure 4.16: Plot window of Scilab showing the result of the circuit simulation

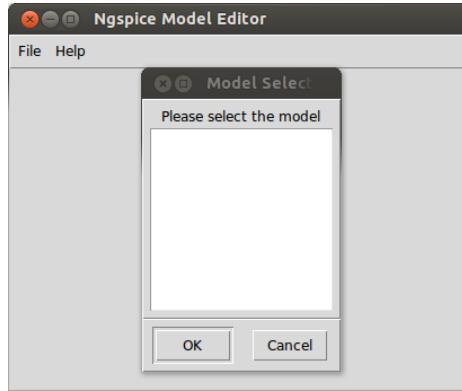


Figure 4.17: Model builder window

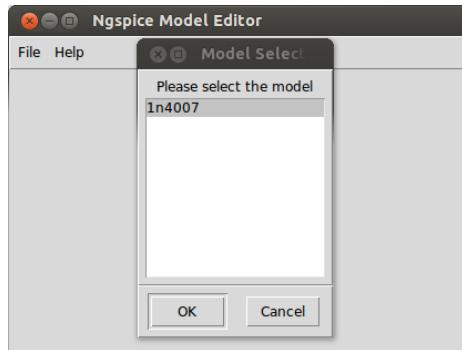


Figure 4.18: Model builder window of circuit having a diode

open. This window will contain the various parameters governing the model of diode, for example reverse breakdown voltage (BV), ohmic resistance (RS) etc. This is illustrated in figure 4.19. To know more about how to use this feature, refer chapter 8.

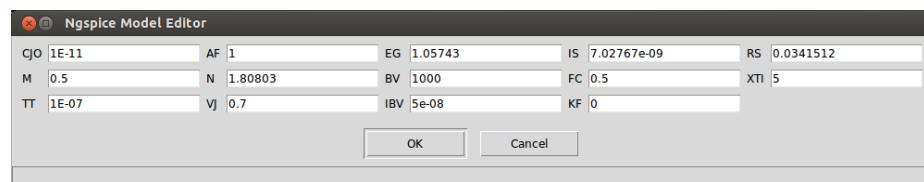


Figure 4.19: Model builder window showing model of diode

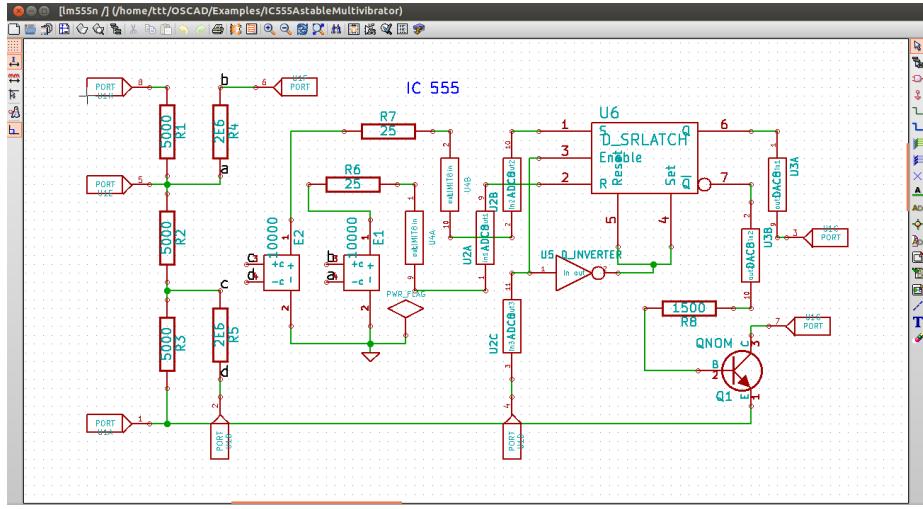


Figure 4.20: Subcircuit of 555 timer IC

4.9 Subcircuit builder

Oscad gives you an option to build subcircuits. The subcircuits can again have components having subcircuits and so on. This enables users to build commonly used circuits as subcircuits and then use it across circuits. For example, one can build a 12 Volt power supply as a subcircuit and then use it as just a single component, across circuits without having the need to recreate it. Clicking on **Subcircuit builder** tool will allow you to edit or create a subcircuit. To know how to make a subcircuit, refer to Chapter 8. Figure 4.20 shows the subcircuit of 555 timer IC.

Chapter 5

Schematic Creation

The first step in the design of an electronic system is the design of its circuit. This circuit is usually created using a **Schematic Editor** and is called a **Schematic**. Oscad uses **EEschema** as its schematic editor. EEschema is the schematic editor of KiCad. It is a powerful schematic editor software. It allows the creation and modification of components and symbol libraries and supports multiple hierarchical layers of printed circuit design.

5.1 Familiarising the schematic editor interface

Figure 5.1 shows the schematic editor and the various menu and tool bars.

5.1.1 Top Menu Bar

The top menu bar will be available at the top left corner. Some of the important menu options in the top menu bar are:

1. File - The file menu items are given below:
 - (a) New - Clear current schematic and start a new one
 - (b) Open - Open a schematic
 - (c) Open Recent - A list of recently opened files for loading
 - (d) Save Whole Schematic project - Save current sheet and all its hierarchy.
 - (e) Save Current Sheet Only - Save current sheet, but not others in a hierarchy.
 - (f) Save Current sheet as - Save current sheet with a new name.
 - (g) Print - Access to print menu (See Figure 5.2).
 - (h) Plot - Plot the schematic in Postscript, HPGL, SVF or DXF format
 - (i) Quit - Quit the schematic editor.

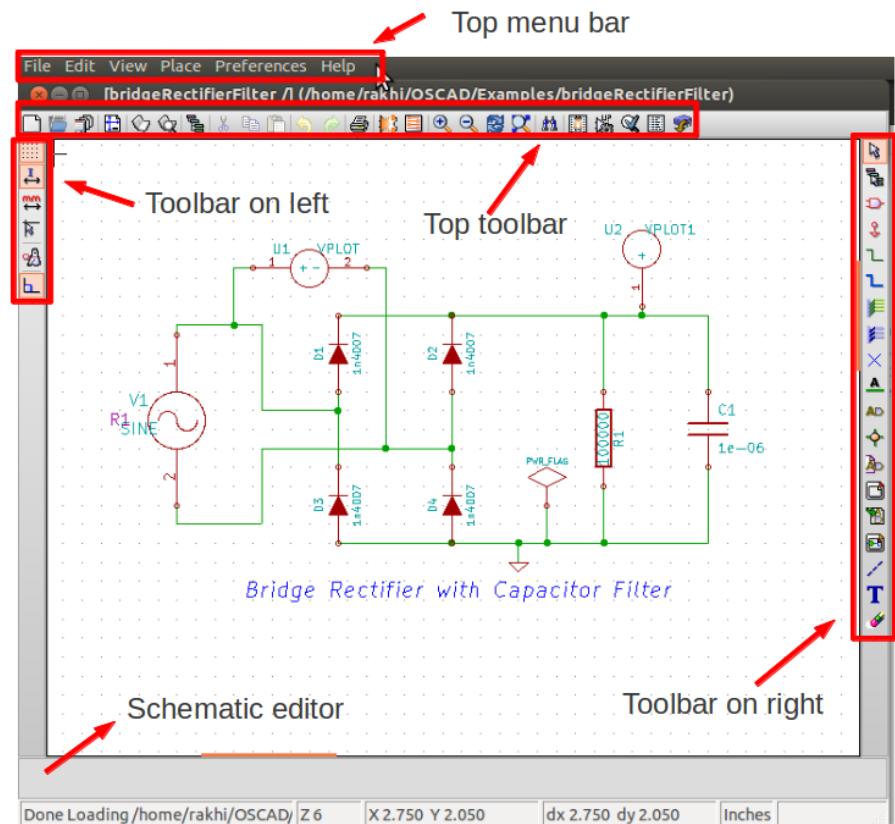


Figure 5.1: Schematic editor with the menu and tool bars shown

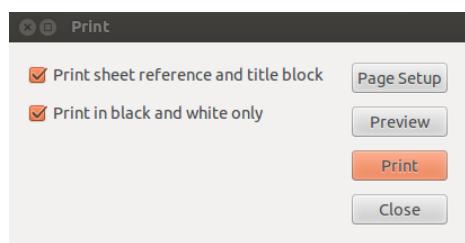


Figure 5.2: Print options



Figure 5.3: Toolbar on top - important tools

2. Place - The place menu has short cuts for placing various items like components, wire, junction etc. onto the schematic editor. See the section 5.1.5 to know more about various short cut keys (hotkeys).
3. Preferences - The preferences menu has the following options:
 - (a) Library - Select libraries and library paths
 - (b) Colors - Select colors for various items.
 - (c) Options - Display schematic editor options (Units, Grid size).
 - (d) Language - Shows the current list of translations. Use default.
 - (e) Hotkeys - Access to the hot keys menu. See the section 5.1.5 about hotkeys.
 - (f) Read preferences - Read configuration file.
 - (g) Save preferences - Save configuration file.

5.1.2 Top toolbar

Some of the important tools in the top toolbar are discussed below. They are marked in Figure 5.3

1. Save - Save your current schematic
2. Library Editor - Create or edit components. See section 5.2 for more details.
3. Library Browser - Browse through the various component libraries available
4. Navigate schematic hierarchy - Navigate between the root and sub-sheets in the hierarchy
5. Print - Print your schematic
6. Generate netlist - Generate a netlist for PCB design or for simulation.

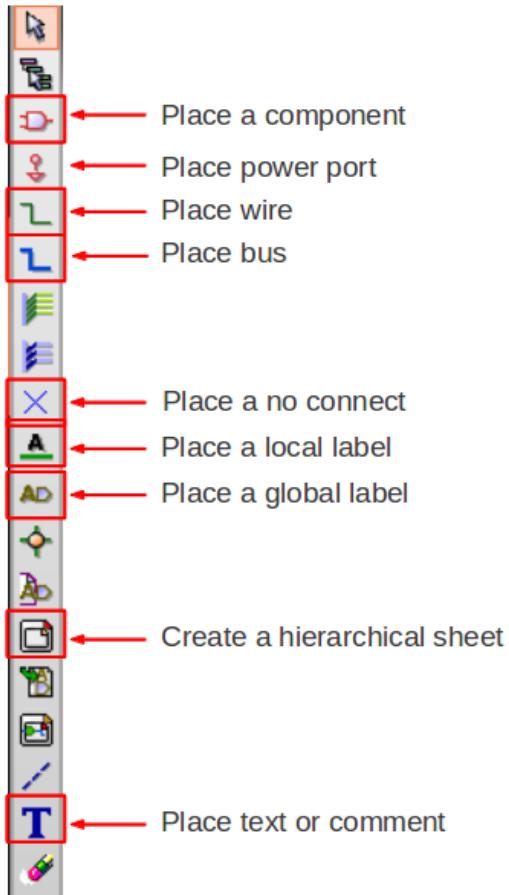


Figure 5.4: Toolbar on right - important tools

7. Annotate - Annotate your schematic
8. Check ERC - Do Electric Rules Check for your schematic
9. Create BOM - Create a Bill of Materials of your schematic

5.1.3 Toolbar on the right

The toolbar on the right side of the schematic editor has many important tools. Some of them are marked in Figure 5.4.

Let us now see these tools one by one.

1. Place a component - Load a component to your schematic. See section 5.4.1 for more details.

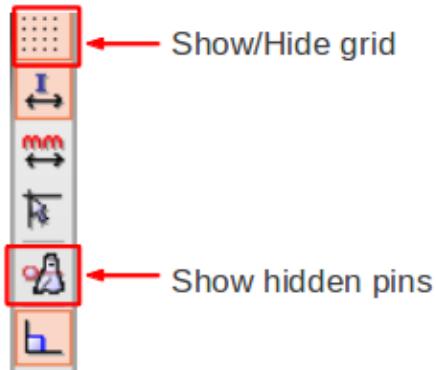


Figure 5.5: Toolbar on right - important tools

2. Place a power port - Load a power port (Vcc, ground) to your schematic
3. Place wire - Draw wires to connect components in schematic
4. Place bus - Place a bus on your schematic
5. Place a no connect - Place a no connect flag, particularly useful in ICs
6. Place a local label - Place a label or node name which is local to the schematic
7. Place a global label - Place a global label (these are connected across all schematic diagrams in the hierarchy)
8. Create a hierarchical sheet - Create a sub-sheet with in the root sheet in the hierarchy. Hierarchical schematics are a good solution for big projects
9. Place a text or comment - Place a text or comment in your schematic

5.1.4 Toolbar on the left

Some of the important tools in the toolbar on the left are discussed below. They are marked in Figure 5.5

1. Show/Hide grid - Show or Hide the grid in the schematic editor. Pressing the tool again hides (shows) the grid if it was shown (hidden) earlier.
2. Show hidden pins - Show hidden pins of certain components, for example, power pins of certain ICs.

5.1.5 Hotkeys

A set of keyboard keys are associated with various operations in the schematic editor. These keys save time and make it easy to move from one operation to another. The list of hotkeys can be viewed by going to Preferences in the top menu bar. Choose *Hotkeys* and select *List current keys*. You can also edit the hotkeys by selecting the option *Edit Hotkeys*. Some of the useful hotkeys are listed below:

- F1 - Zoom in
- F2 - Zoom out
- Ctrl + Z - Undo
- Delete - Delete item
- M - Move item
- C - Copy item
- A - Add/place component
- P - Place power component
- R - Rotate item
- X - Mirror component about X axis
- Y - Mirror component about Y axis
- E - Edit schematic component
- W - Place wire
- T - Add text
- S - Add sheet

Note that both lower and upper-case keys will work as hotkeys.

5.2 Components and Component libraries

Oscad schematic editor has a huge collection of components. All the component libraries in EEschema, on which Oscad schematic editor is based, are available. As EEschema is meant to be a schematic editor to create circuits for PCB, EEschema lacks some components that are necessary for simulation (e.g., plots, current sources, etc.). A set of component libraries has been created with such components. If you are using Oscad only for designing a PCB, then you might not need these libraries. However, these libraries are essential if you need to simulate your circuit. Hereafter, we will refer to these libraries as *Oscad libraries* to distinguish them from libraries already present in EEschema (EEschema libraries).

5.2.1 Oscad libraries

The Oscad libraries (created for simulation) are given below:

1. analogSpice - Discrete components like capacitor, resistor, BJT etc.
2. analogXSpice - Analog Xspice library
3. convergenceAidSpice - To set initial conditions
4. converterSpice - A/D and D/A converters
5. digitalSpice - ICs for digital circuits e.g., the 74 series
6. digitalXSpice - Flip-flops, logic gates etc.
7. linearSpice - 555 timer IC, op amp 741 etc
8. measurementSpice - Plot and print components
9. portSpice - Port
10. sourcesSpice - Current and voltage sources for simulation

Note that the names of all Oscad libraries end with the word *Spice*. Consider the Oscad library *linearSpice*. If you want to simulate a circuit that has a 555 timer IC in it, you should use the 555 timer from this library. This is because only then it will be mapped to the subcircuit of 555 which is required for simulation. Similarly if you use a Flip flop from *digitalXSpice*, the Xspice description of the Flip flop will be mapped to it and enables us to simulate the Flip flop behaviour.

5.2.2 Adding Oscad component libraries to project

Let us see how you can add the Oscad libraries to your project. Go to *Preferences* from the top menu bar. Choose *Library*. You will get the window shown in Figure 5.6. Click on *Add* (marked in red). Browse to the folder where Oscad is installed. Go to the folder *Library*. Select all the **.lib* files as shown in Figure 5.7. Click on *Open*. Now click on *OK* on the window shown in Figure 5.6.

Note: You will have add these to your project each time you create or edit your schematic.

5.2.3 Library browser

You can browse through EEschema and Oscad libraries using the library browser tool from the top menu bar. The components in the *sourcesSpice* library is shown in Figure 5.8 to illustrate this.

Note that you will be able to view the Oscad libraries in the library browser ONLY IF you have added them to your project as described in Section 5.2.2

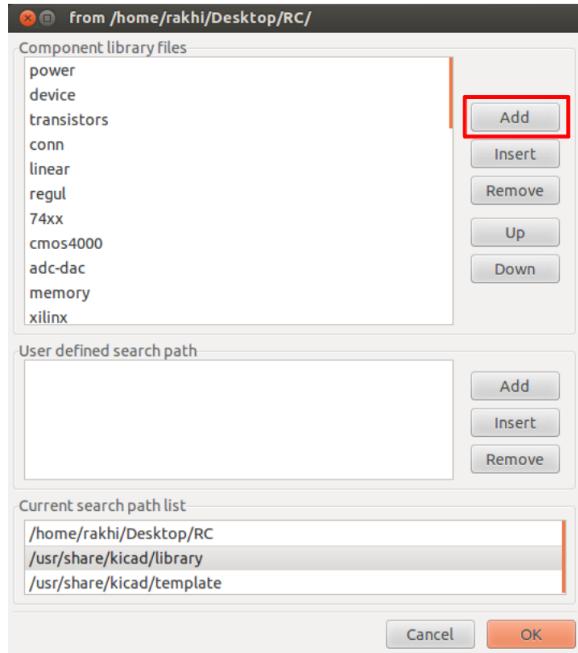


Figure 5.6: Add component library

5.2.4 Plot component library

Plot components are required to view the results of simulation. These are available in the Oscad library *measurementSpice* shown in Figure 5.9. These are used only for simulations. Some of the plots available in this library are:

- IPLOT - Plot the current through a component.
- VPLOT1 - Plot the voltages at nodes in separate graph windows.
- VPLOT8_1 - Plot the voltages at nodes in the same graph window.
- VPLOT - Plot the voltage difference between the two nodes where it is placed.

5.2.5 Power component library

Power components (Vcc and ground) are essential parts of a schematic - both for PCB design and simulation. Power components are available in the EEschema library *power* as shown in Figure 5.10. Another important component in this library is the Power Flag, *PWR_FLAG*. It is a dummy component placed in schematic to tell the schematic editor that the pin/node is driven by a power source and hence prevents ERC errors.

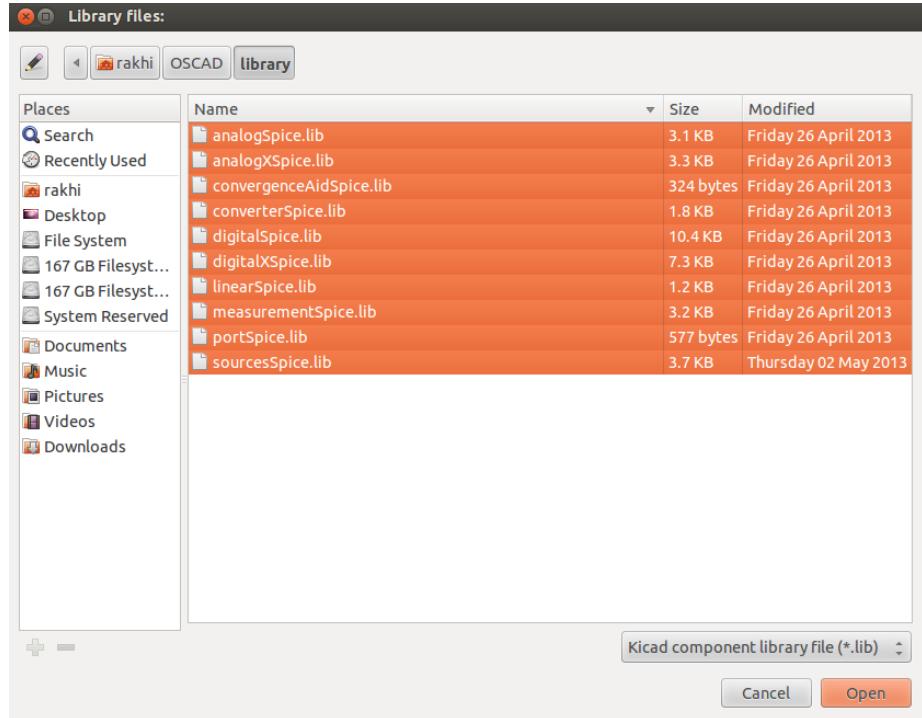


Figure 5.7: Select all the Oscad library (*.lib) files as shown

5.2.6 Connector library

You would want to place connectors in your PCB to take signals in and out of it. These connectors are available in the EEschema library *conn* as shown in Figure 5.11.

5.2.7 Component references

Every component has a unique reference. For e.g., resistor has a reference R, BJTs have a reference Q, MOSFETs have a reference M, ICs have a reference U etc. If a component is to be made a subcircuit, then its reference should be 'X'. When a component is placed in the schematic editor, the reference will be shown with a question mark. This indicates that the component is not annotated. See section 5.4.4 for more information about annotation.

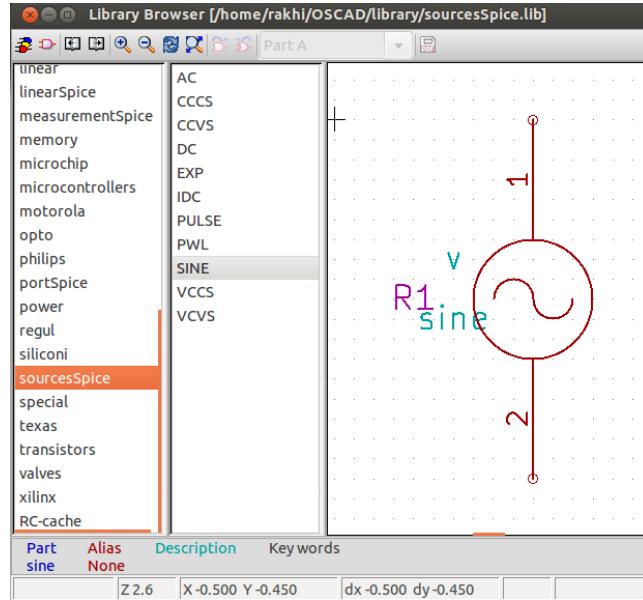


Figure 5.8: Library browser - an example

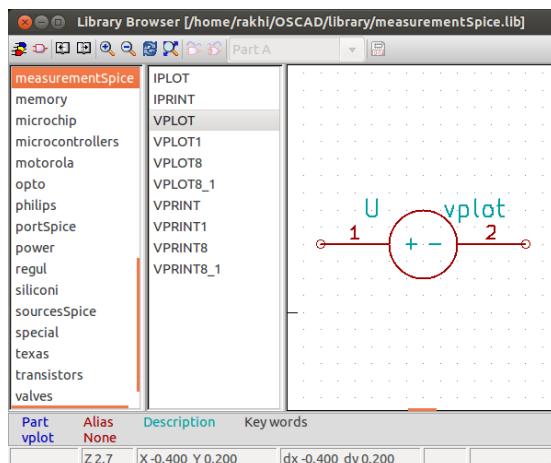


Figure 5.9: The Oscad *measurementSpice* library

5.3 Schematic creation for Simulation and PCB design - Differences

There are certain differences between the schematic created for simulation and that created for PCB design. We need certain components like plots, current

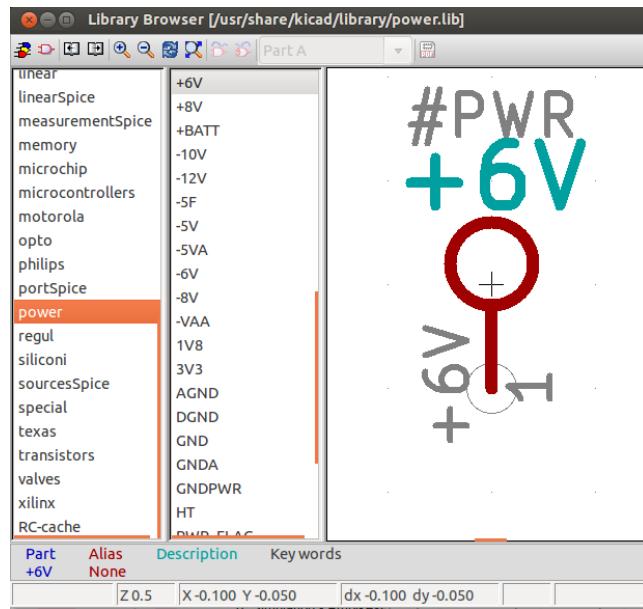


Figure 5.10: The EEschema *power* library

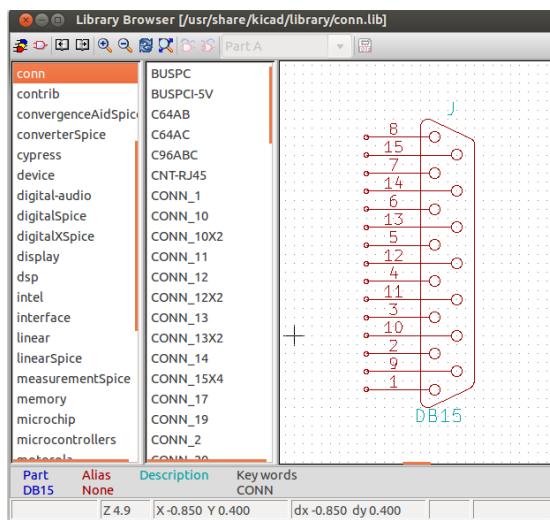


Figure 5.11: The EEschema *conn* library

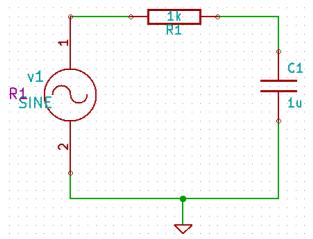


Figure 5.12: RC circuit

sources etc. for simulation whereas these are not needed for PCB design. For PCB design, we would require connectors (e.g., DB15, 2 pin connector etc) for taking signals in and out of the PCB whereas these have no meaning in simulation.

5.4 Schematic creation for simulation

The first step in the creation of circuit schematic is the selection and placement of required components. Let us see this using an example. Let us create the circuit schematic of an RC filter given in Figure 5.12 and do a transient simulation.

5.4.1 Selection and Placement of components

We would need a resistor, a capacitor, a voltage source, ground terminal and some plot components.

Add the Oscad libraries to your project as described in section 5.2.2.

To place a resistor on your schematic editor, select the *Place a component* tool from the toolbar on the right side and click anywhere on the schematic editor. This opens up the component selection window. (The above action can also be performed by pressing the key A.) Type R in the field *Name* of the **component selection** window as shown in Figure 5.13. Click on OK. A resistor will be tied to the cursor. Place the resistor on the schematic editor by a single click.

To place the next component, i.e., capacitor, click again on the schematic editor. Type C in the Name field of component selection window. Click on OK. Place the capacitor on the schematic editor by a single click.

Let us now place a sinusoidal voltage source. This is required for performing transient analysis. To place it, click again on the schematic editor. On the component selection window, click on **List all**. Choose the library **sourcesSpice** by double-clicking on it. Select the component **SINE** and click on OK. Place the sine source on the schematic editor by a single click.

We need to place two plot components. Let us place **vplot8_1** as we need to view input and output waveforms in the same graph window. To do so,

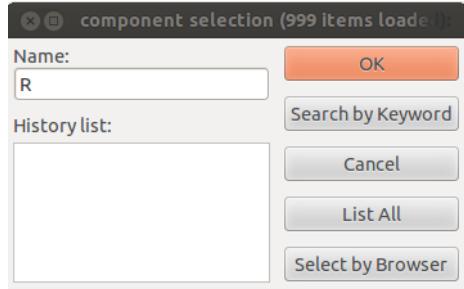


Figure 5.13: Placing a resistor using the Place a Component tool

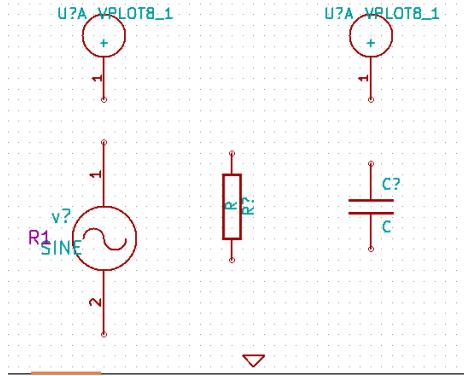


Figure 5.14: All components for RC circuit simulation are placed

choose and place vplot8_1 from the `measurementSpice` library. To place one more vplot8_1, place the cursor on top of vplot8_1 and press the key **C** to copy it. Place the component by clicking on the schematic editor.

Similarly place a ground terminal `gnd` from the library `power`. It can also be placed using the *Place a power port* tool from the toolbar on the right. Click anywhere on the editor after selecting place a power port tool. Click *List all* and choose `gnd`.

Once all the components are placed, the schematic editor would look like the Figure 5.14.

Let us rotate the resistor to complete the circuit as shown in Figure 5.12. To rotate the resistor, place the cursor on the resistor and press the key **R**. Note that if you place the cursor above the letter **R** (not **R?**) on the resistor, you may be asked to clarify selection. Choose the option *Component R*. You can avoid this by placing the cursor slightly away from the letter **R** as shown in Figure 5.15. This applies to all components.

If you want to move a component, place the cursor on top of the component and press the key **M**. The component will be tied to the cursor and can be moved in any direction.

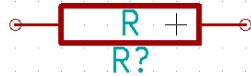


Figure 5.15: Place the cursor (shown by the cross mark) slightly away from the letter R

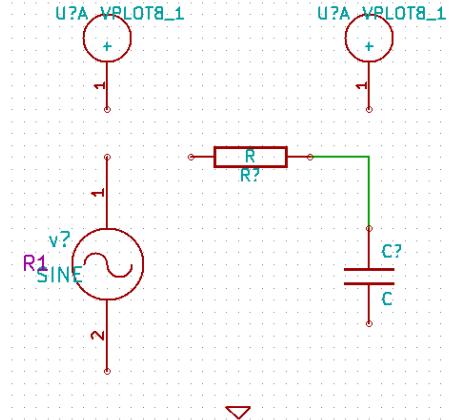


Figure 5.16: Wiring example

5.4.2 Wiring the circuit

The next step is to wire the connections. Let us connect the resistor to the capacitor. To do so, point the cursor to the terminal of resistor you want to connect and press the key W. It has now changed to the wiring mode. Move the cursor towards the terminal of the capacitor and click on it. A wire is formed as shown in Figure 5.16.

Similarly connect the wires between all terminals and the final schematic would look like Figure 5.17.

5.4.3 Assigning values to components

We need to assign values to the components in our circuit i.e., resistor and capacitor. Note that the sine voltage source has been placed for simulation. The specifications of sine source will be given during simulation.

To assign value to resistor, place the cursor above the letter R (not R?) and press the key E. Choose *Field value*. Type 1k in the *Edit value field* box as shown in Figure 5.18. 1k means $1k\Omega$. Similarly give the value 1u for the capacitor. 1u means $1\mu F$.

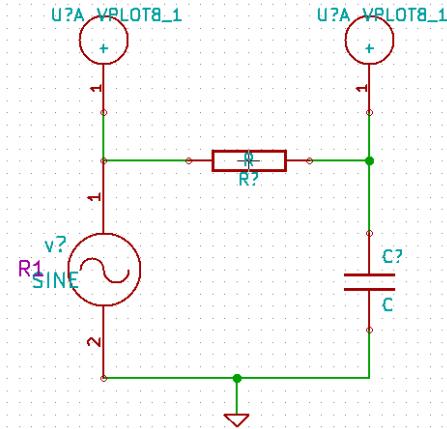


Figure 5.17: Wiring done

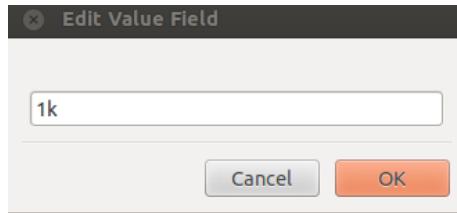


Figure 5.18: Editing value of resistor

5.4.4 Annotation and ERC

The next step is to annotate the schematic. Annotation gives unique references to the components. To annotate the schematic, click on *Annotate schematic* tool from the top toolbar. Click on annotation, then click on OK and finally click on close as shown in Figure 5.19. The schematic is now annotated. The question marks next to component references have been replaced by unique numbers. If there are more than one instance of a component (say resistor), the annotation will be done as R1, R2 etc.

Let us now do ERC or Electric Rules Check. To do so, click on *Perform electric rules check* tool from the top toolbar. Click on *Test Erc* button. You may get an error as shown in Figure 5.20. Click on close in the test erc window. There will be a green arrow pointing to the source of error in the schematic. Here it points to the ground terminal. This is shown in Figure 5.21

To correct this error, place a PWR_FLAG from the EEschema library power. Connect the power flag to the ground terminal as shown in Figure 5.22. More information about PWR_FLAG is given in section 5.2.5. You need to place PWR_FLAG wherever you get the error shown in Figure 5.20. Repeat the ERC. Now there are no errors. With this we have created the schematic for simulation.

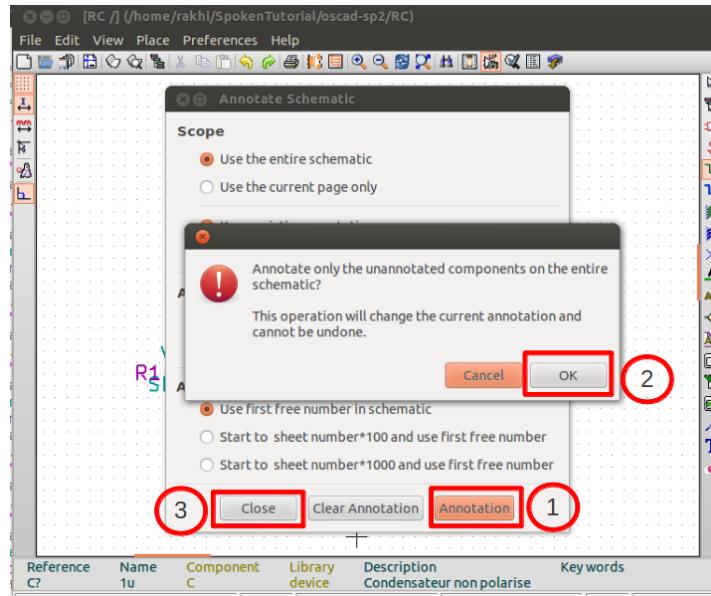


Figure 5.19: 1. First click on Annotation then 2. Click on Ok then 3. Click on close

ErrType(3): Pin connected to some others pins but no pin to drive it
 • @ (5.5500 ",3.2000 "): Cmp #PWR01, Pin 1 (power_in) not driven (Net 1)

Figure 5.20: ERC error

5.4.5 Netlist generation

To simulate the circuit that you created in the previous section, we need to generate its netlist. Netlist is a list of components in the schematic along with their connection information. To do so, click on the *Generate netlist* tool from the top toolbar. Click on spice from the window that opens up. Uncheck the option **Prefix references ‘U’ and ‘IC’ with ‘X’**. Then click on **Netlist**. This is shown in figure 5.23. Save the netlist. This will be a **.cir** file. Do not change



Figure 5.21: Green arrow pointing to Ground terminal indicating an ERC error

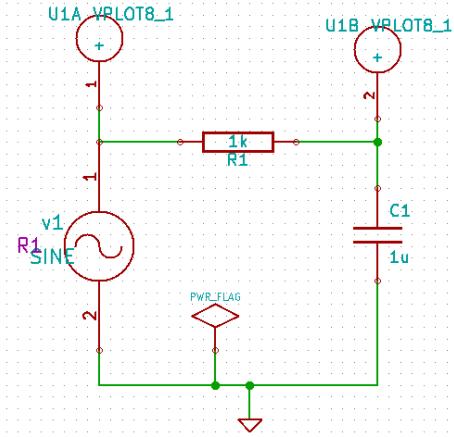


Figure 5.22: Final schematic with PWR_FLAG

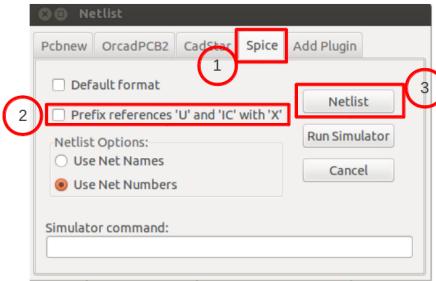


Figure 5.23: 1. Click on Spice then 2. Uncheck the option **Prefix references 'U' and 'IC' with 'X'** then 3. Click on Netlist

the directory while saving.

Now you are ready with the netlist to be simulated. The next chapter will guide you to perform simulation. If you would like to know more about EEschema, please refer [7].

Chapter 6

Simulation

Circuit simulation uses mathematical models to replicate the behaviour of an actual device or circuit. Simulation software allows for modeling of circuit operation. Simulating a circuit's behaviour before actually building it can greatly improve design efficiency by making faulty designs known as such, and providing insight into the behavior of electronic circuit designs. Oscad uses `Ngspice` for mixed-level/mixed-signal circuit simulation.

The various steps involved in simulating a circuit schematic in Oscad is given below:

1. Analysis insertion - This adds the type of simulation to be done to the netlist. This is done by the **Analysis inserter** tool in the Oscad tool bar.
2. Netlist conversion - The netlist created in the Schematic editor will be converted to Ngspice format and analysis options will be appended to it. This is done by the **Netlist Converter** tool in the Oscad tool bar.
3. Ngspice simulation - Ngspice simulation of the netlist is performed. This is done by the **Ngspice** tool in the Oscad tool bar.

In the following sections, we shall describe each of the above steps.

6.1 Analysis Inserter

In order to simulate a circuit, the user must define the type of analysis to be done on the circuit. The types of analysis include Operating point analysis, DC analysis, AC analysis, transient analysis etc. The user should also specify the options corresponding to each analysis. This is facilitated by the **Analysis inserter** tool in Oscad.

Analysis inserter generates the commands for Ngspice. When you click on analysis inserter from the Oscad tool bar, you will get the Analysis inserter GUI as shown in Figure 6.1. It consists of type of analysis on the top, and user needs

to type the details which are needed to perform simulation in the corresponding fields.

6.1.1 Types of Analysis

Oscad supports three types of analyses:

1. DC Analysis (Operating Point and DC Sweep)
2. AC Small-signal Analysis
3. Transient Analysis

Other analyses in the Analysis inserter are currently under progress. The different types of analyses supported in Oscad are explained below [8].

DC Analysis

The dc analysis determines the dc operating point of the circuit with inductors shorted and capacitors opened. The dc analysis options are specified on the `.dc`, and `.op` control lines.

There is assumed to be no time dependence on any of the sources within the system description. The simulator algorithm subdivides the circuit into those portions which require the analog simulator algorithm and those which require the event-driven algorithm. Each subsystem block is then iterated to solution, with the interfaces between analog nodes and event-driven nodes iterated for consistency across the entire system.

Once stable values are obtained for all nodes in the system, the analysis halts and the results may be displayed or printed out as you request them.

A dc analysis is automatically performed prior to a transient analysis to determine the transient initial conditions, and prior to an ac small-signal analysis to determine the linearized, small-signal models for nonlinear devices. The dc analysis can also be used to generate dc transfer curves: a specified independent voltage or current source is stepped over a user-specified range and the dc output variables are stored for each sequential source value.

AC Small-signal Analysis

AC analysis is limited to analog nodes and represents the small signal, sinusoidal solution of the analog system described at a particular frequency or set of frequencies. This analysis is similar to the DC analysis in that it represents the steady-state behavior of the described system with a single input node at a given set of stimulus frequencies.

The program first computes the dc operating point of the circuit and determines linearized, small-signal models for all of the nonlinear devices in the circuit. The resultant linear circuit is then analyzed over a user-specified range of frequencies. The desired output of an ac small-signal analysis is usually a transfer function (voltage gain, transimpedance, etc). If the circuit has only

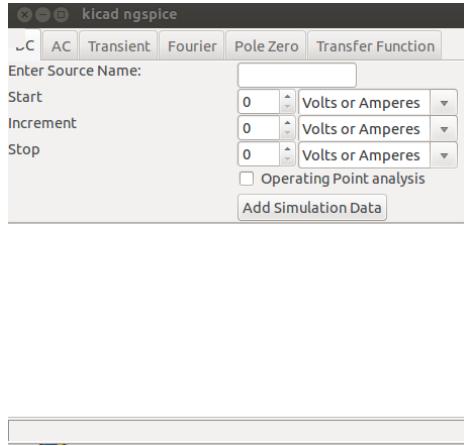


Figure 6.1: Analysis inserter GUI

one ac input, it is convenient to set that input to unity and zero phase, so that output variables have the same value as the transfer function of the output variable with respect to the input.

Transient Analysis

Transient analysis is an extension of DC analysis to the time domain. A transient analysis begins by obtaining a DC solution to provide a point of departure for simulating time-varying behavior. Once the DC solution is obtained, the time-dependent aspects of the system are reintroduced, and the simulator algorithms incrementally solve for the time varying behavior of the entire system. Inconsistencies in node values are resolved by the simulation algorithms such that the time-dependent waveforms created by the analysis are consistent across the entire simulated time interval.

Resulting time-varying descriptions of node behavior for the specified time interval are accessible to you. All sources which are not time dependent (for example, power supplies) are set to their dc value. The transient time interval is specified on a `.tran` control line.

6.1.2 DC Analysis Inserter

By default DC analysis option will appear when you click on analysis inserter. Here we need to give the details of input source name, start value of input, increment and stop value. Then click on “Add Simulation Data”.

The Figure 6.2 gives an example of DC analysis inserter. In this example 'v1' is the input voltage source which starts at '0 Volt', increments by '1 Volt' and stops at '10 Volt'. On clicking “Add simulation data”, the analysis statement is generated and is of the form:

```
.dc sourcename vstart vstop vincr
```

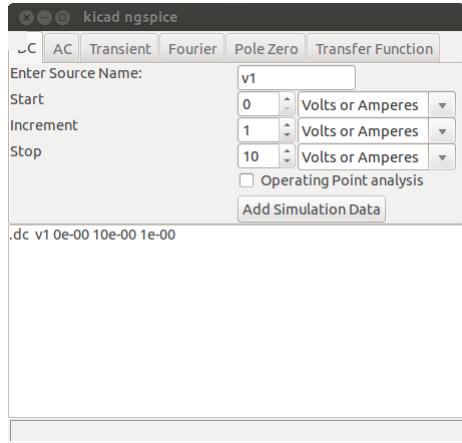


Figure 6.2: DC Analysis option added

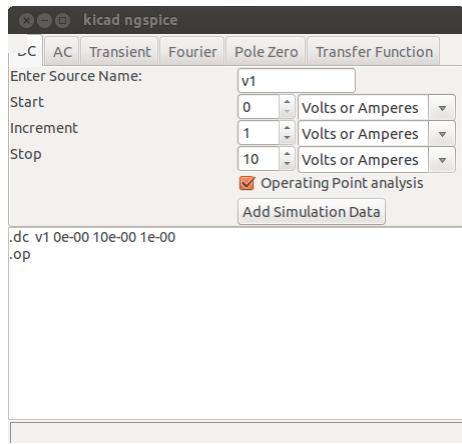


Figure 6.3: OP Analysis

The `.dc` line defines the dc transfer curve source and sweep limits (with capacitors open and inductors shorted). `srcnam` is the name of an independent voltage or current source. `vstart`, `vstop`, and `vincr` are the starting, final, and incrementing values respectively, of the source.

When we check the option `Operating point analysis` on the DC analysis window, `.op` gets appended to the analysis statement. This is shown in Figure 6.3. The inclusion of the line `.op` in the analysis file directs Ngspice to determine the dc operating point of the circuit with inductors shorted and capacitors opened.

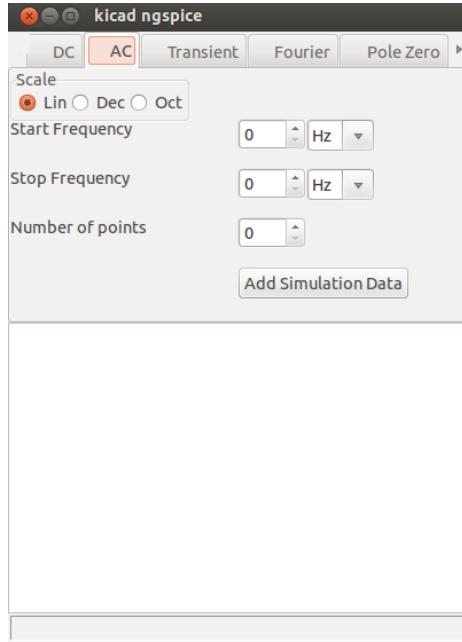


Figure 6.4: AC Analysis GUI

6.1.3 AC Analysis Inserter

When you click on the option “AC” in the analysis inserter GUI, the window given in Figure 6.4 will appear. Here you need to enter the details of **scale**, **start frequency**, **stop frequency** and **Number of points**.

After entering these values, click on “Add Simulation Data”. The analysis statement will be generated. This will be in one of the three forms listed below, depending on the type of **scale** that you choose. The types of **scale** available are dec, oct, and lin.

```
.ac dec nd fstart fstop
.ac oct no fstart fstop
.ac lin np fstart fstop
```

dec stands for decade variation, and **nd** is the number of points per decade. **oct** stands for octave variation, and **no** is the number of points per octave. **lin** stands for linear variation, and **np** is the number of points. **fstart** is the starting frequency, and **fstop** is the final frequency.

If the **.ac** analysis is included in the analysis file, Ngspice performs an AC analysis of the circuit over the specified frequency range. Note that in order for this analysis to be meaningful, at least one independent source must have been specified with an ac value.

An example of “lin” scale is given in Figure 6.5. Here the start frequency is “1 Hz”, stop frequency is “10 Meg” and number of points is “10”.

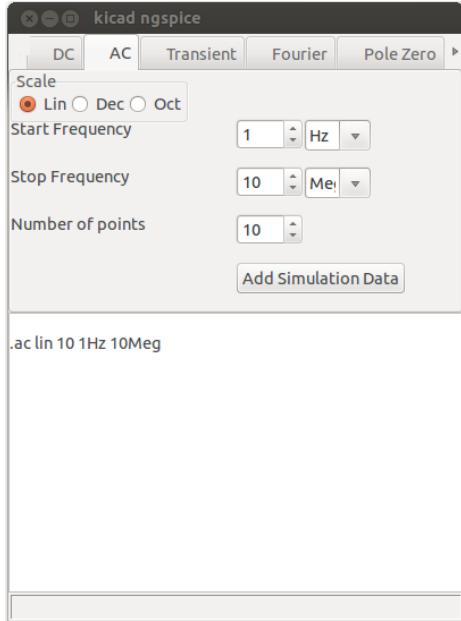


Figure 6.5: AC Analysis options added

6.1.4 Transient Analysis Inserter

When you click on the option “Transient” in the analysis inserter GUI, the window given in Figure 6.6 will appear. Here you need to enter the details of **start time**, **step time**, and **stop time**. After entering these values, click on Add Simulation Data. The analysis statement will be generated. It will be of the form:

```
.tran tstep tstop tstart
```

tstep is the printing or plotting increment for line-printer output. For use with the postprocessor, **tstep** is the suggested computing increment. **tstop** is the final time, and **tstart** is the initial time. If **tstart** is omitted, it is assumed to be zero.

The transient analysis always begins at time zero. In the interval $tstart < tstop$, the circuit is analyzed (to reach a steady state), but no outputs are stored. In the interval $tstart \leq t \leq tstop$, the circuit is analyzed and outputs are stored.

An example of transient analysis inserter is given in Figure 6.7. Here start time is “0 sec”, step time is “1 sec” and stop time is “10 sec”.

6.1.5 Saving the analysis file

After entering the details of analysis we need to save the **analysis** file, which contains the options we added. Save the analysis file as explained below:

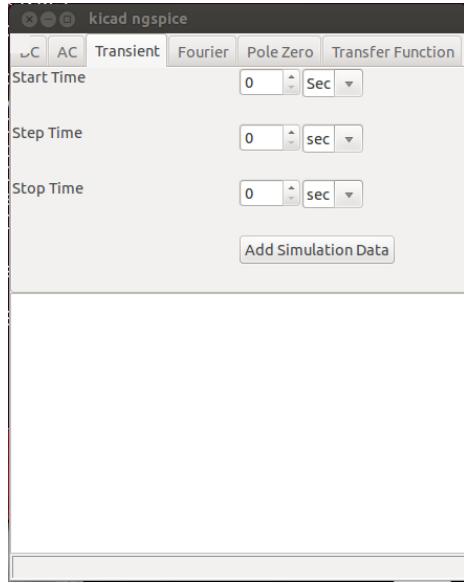


Figure 6.6: Transient Analysis - GUI

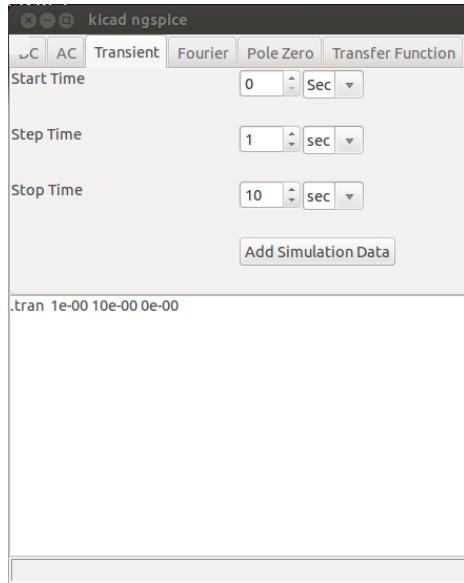


Figure 6.7: Transient Analysis options added

Click on “File” from the top menu bar. Click on “Save”. This is shown in Figure 6.8. Click on Save in the **Choose a File** dialog box as shown in Figure 6.9. The options added in the analysis inserter GUI will be saved in a file named

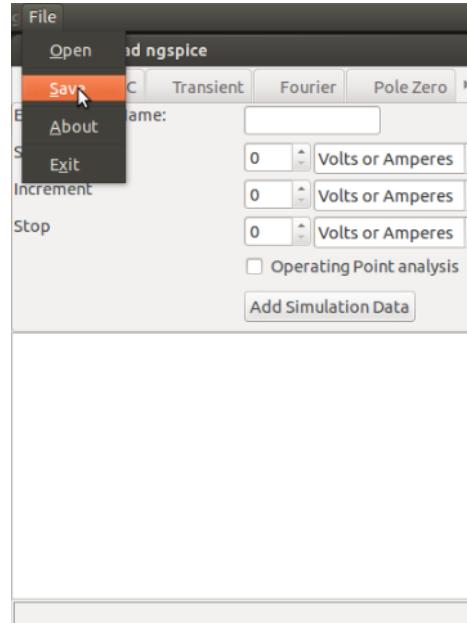


Figure 6.8: Choosing the Save option from File menu

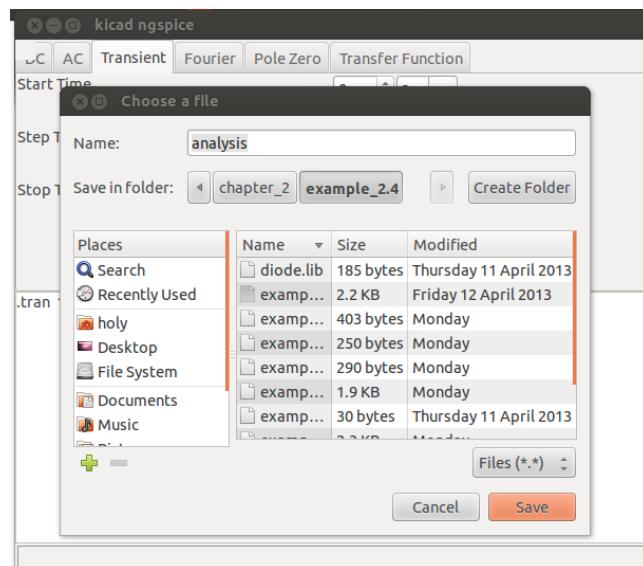


Figure 6.9: Saving the analysis information to netlist

analysis.

```

=====
Kicad to Ngspice netlist converter
=====
converting example_5.6.cir
.tran 20e-03 2e-00 0e-00
-----
Add parameters for sine source v1
Enter offset value (Volts/Amps): 0
Enter amplitude (Volts/Amps): 5
Enter frequency (Hz): 50
Enter delay time (seconds): 0
Enter damping factor (1/seconds): 0
-----
The ngspice netlist has been written in example_5.6.cir.out
The scilab netlist has been written in example_5.6.cir.ckt
Press Enter to quit

```

Figure 6.10: Parameters to be added for sinusoidal source

6.2 Modifying KiCad netlist for Ngspice simulation

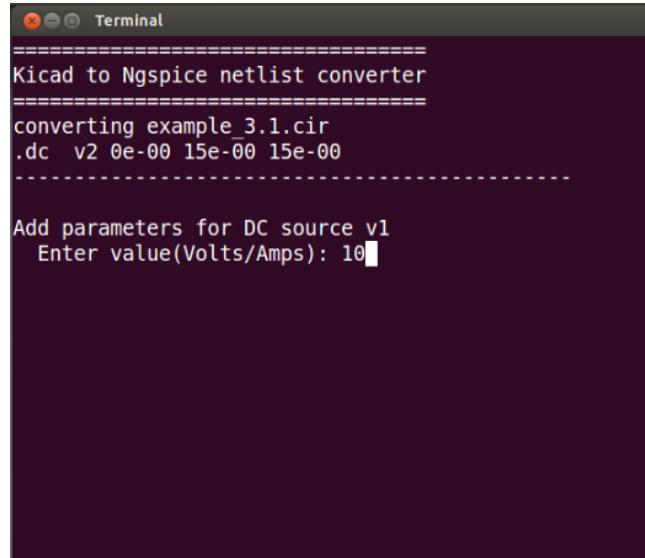
The schematic editor provides a netlist file, which describes the electrical connections between circuit components. This is a SPICE netlist which can not be directly used for Ngspice simulation due to compatibility issues. This needs to be converted to Ngspice format. For this, click on **Netlist converter** tool from the Oscad toolbar after analysis insertion. Oscad netlist converter performs the following operations to generate Ngspice compatible netlist.

6.2.1 Insert parameters for fictitious components

For simulation of the circuit, the values of sources must be specified. For example, for a sine wave voltage source, parameters such as frequency, amplitude, offset etc. are required to be entered.

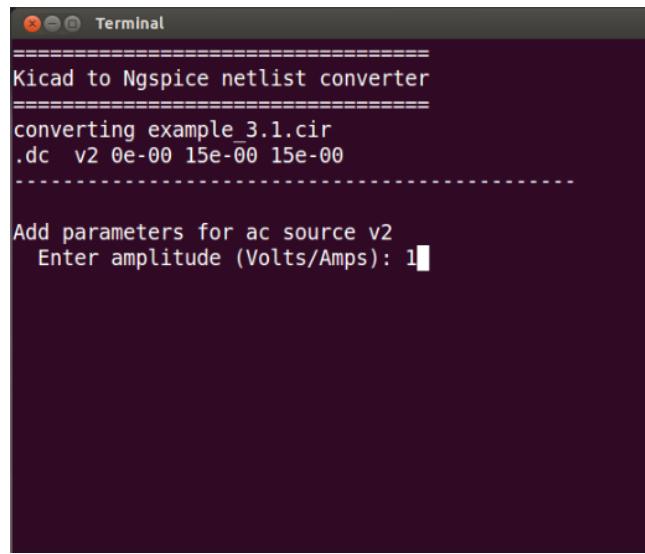
The netlist converter scans the netlist file and asks for parameter values for the sources wherever required. When we click on **Netlist converter**, a terminal window appears. It asks for various parameter values. This varies depending upon the voltage/current sources added in the schematic for simulation.

1. When sinusoidal source is available in schematic - in the terminal the details of sinusoidal source like offset value, amplitude, frequency, delay time, damping factor are asked. This is shown in Figure 6.10.
2. When DC source of unknown value is available in schematic - It will ask for the value of DC source. In the example shown in Figure 6.11, the value of DC source is 10V.



```
Terminal
=====
Kicad to Ngspice netlist converter
=====
converting example_3.1.cir
.dc v2 0e-00 15e-00 15e-00
-----
Add parameters for DC source v1
Enter value(Volts/Amps): 10
```

Figure 6.11: Parameters to be added for dc source



```
Terminal
=====
Kicad to Ngspice netlist converter
=====
converting example_3.1.cir
.dc v2 0e-00 15e-00 15e-00
-----
Add parameters for ac source v2
Enter amplitude (Volts/Amps): 1
```

Figure 6.12: Parameters to be added for ac source

3. When AC source is available in schematic - It will ask for amplitude of AC source, which in this case is 1V. This is shown in Figure 6.12.

6.2.2 Convert IC into discrete blocks

As Eeschema (schematic editor used in Oscad) is intended for PCB Designing, it creates netlist in terms of IC and not components, e.g., if the circuit contains a two-input Nand gate, then in the netlist, IC 7400 appears instead of the Nand gate. Oscad netlist converter converts the IC into discrete blocks by considering proper input and output connections and IC specifications (voltage levels, speed of the operation etc.).

6.2.3 Insert Digital-to-Analog (D-to-A) and Analog-to-Digital (A-to-D) converter at appropriate places

Oscad provides capability to perform mixed mode simulation. Thus circuits with analog and digital components can be analyzed. In order to simulate such kind of circuits, D-to-A and A-to-D converters are inserted at appropriate places. The netlist generated from Eeschema is assumed to have analog connections and for digital components, A-to-D converter for inputs and D-to-A converter for outputs are added.

6.2.4 Insert plotting and printing statements

There is a library for plotting and printing the voltages and currents in the circuit. The netlist converter adds appropriate printing and plotting commands (current or voltage plot, or single or differential plot) in the netlist depending on the print/plot components. Ngspice can find the current through voltage sources only. Netlist converter inserts a zero volt voltage source in series with the component through which current needs to be computed. Thus current through any component can be obtained.

6.2.5 Insert analysis and option

Oscad netlist converter inserts analysis information created in Section 6.1 into the netlist.

6.2.6 Insert models and subcircuits

Oscad netlist converter inserts models or subcircuits for required components into the netlist. To know more about model building and subcircuit creation, please refer Chapter 8.

6.2.7 Simulation

After netlist conversion, a `.cir.out` file will be generated which is compatible with Ngspice simulation software. Let us see how to simulate this file and obtain the results.

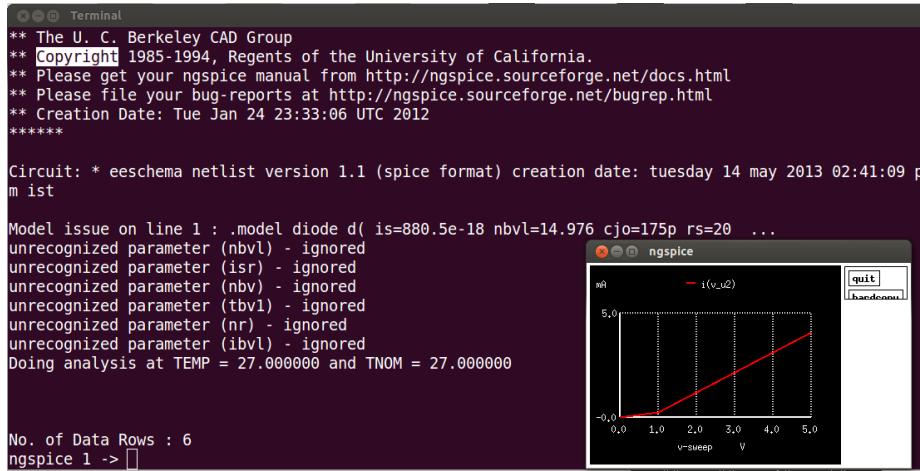


Figure 6.13: Ngspice simulation of .cir.out file

Ngspice simulation in Oscad

Click on Ngspice from the Oscad toolbar. The Ngspice terminal and waveform windows will appear. An example is shown in Figure 6.13.

6.3 Examples

Let us see a few simulation examples which describes the use of Analysis inserter, Netlist converter and Ngspice.

6.3.1 DC Analysis

Consider the `nodalExample` (nodal analysis) given in the `Examples` folder available in the Oscad webpage www.oscad.net. Open Schematic editor and generate spice netlist as shown in Chapter 5. Click on “analysis inserter”. Here we decide which type of analysis we want to do. Let us do **DC Analysis**.

Click on DC and Enter the following details:

`source name = i1, start = 0A, Increment = 1A, stop = 10A` and then click on Add Simulation data. You get the window as shown in Figure 6.14. Save and close the analysis. Now click on Netlist Converter which shows the terminal given in Figure 6.15. Press ‘enter’ key. After this click on Ngspice tool which will show Ngspice terminal with waveform window as shown in the Figure 6.16.

6.3.2 AC Small-signal Analysis

Consider the `RC_ac` example from the `Example` folder available in Oscad website. Open Schematic editor and generate spice netlist as given in Chapter 5. Click

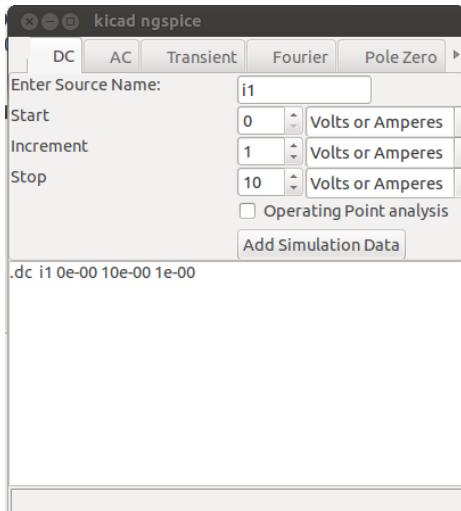


Figure 6.14: DC analysis – adding simulation data

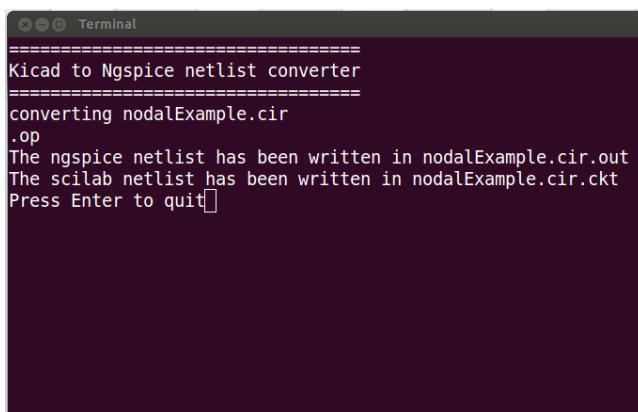


Figure 6.15: Terminal - KiCad to Ngspice netlist converter

on “analysis inserter”. Let us do **AC Analysis**.

Click on AC and add the following details: “scale” = lin, “start frequency” = 1 Hz, “stop frequency” = 10 Meg, “number of points” = 10 and then click on Add simulation data. This is shown in Figure 6.17. Now click on Netlist converter which opens up a terminal. It asks for the amplitude of AC. Type 1 and press ‘enter’ key. A .cir.out netlist file is generated. This is shown in Figure 6.18. After this click on Ngspice tool which will show Ngspice terminal with waveform window as shown in following figure 6.19.

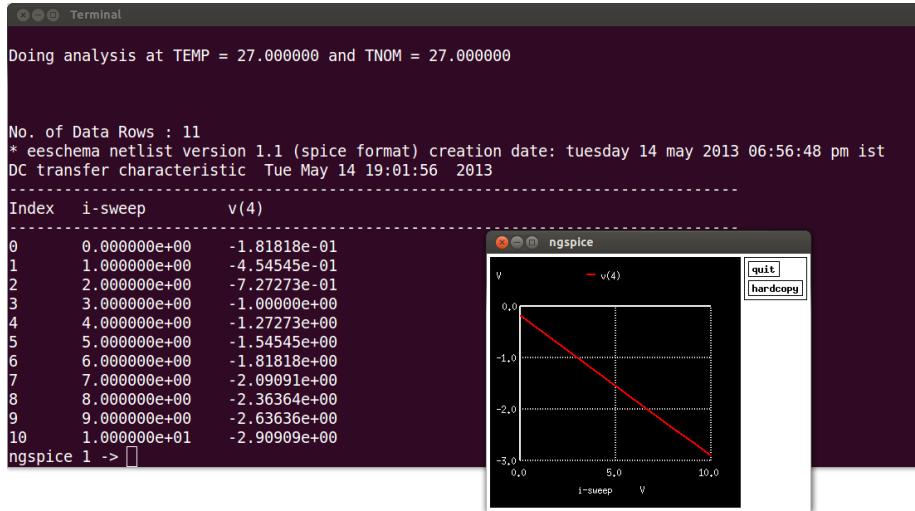


Figure 6.16: Ngspice simulation result

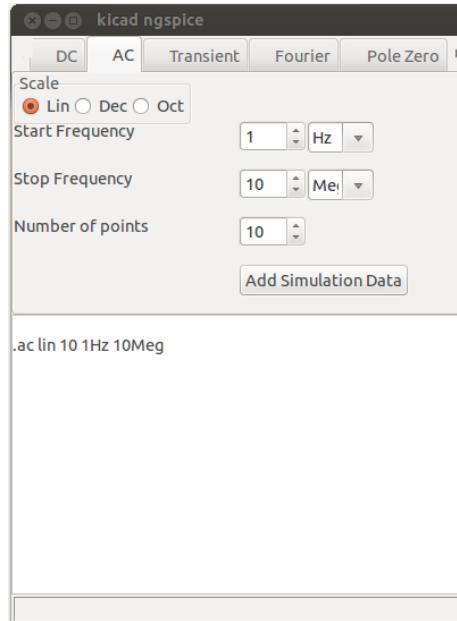


Figure 6.17: AC analysis inserter

6.3.3 Transient Analysis

Let us use the RC circuit project created in Chapter 5 to do transient simulation. Open the **Anlaysis inserter** tool. Click on transient and then add the

```

=====
Kicad to Ngspice netlist converter
=====
converting RC.cir
.tran 5e-03 30e-03 0e-00
-----
Add parameters for ac source v1
  Enter amplitude (Volts/Amps): 1
-----
The ngspice netlist has been written in RC.cir.out
The scilab netlist has been written in RC.cir.ckt
Press Enter to quit

```

Figure 6.18: AC analysis - Netlist converter

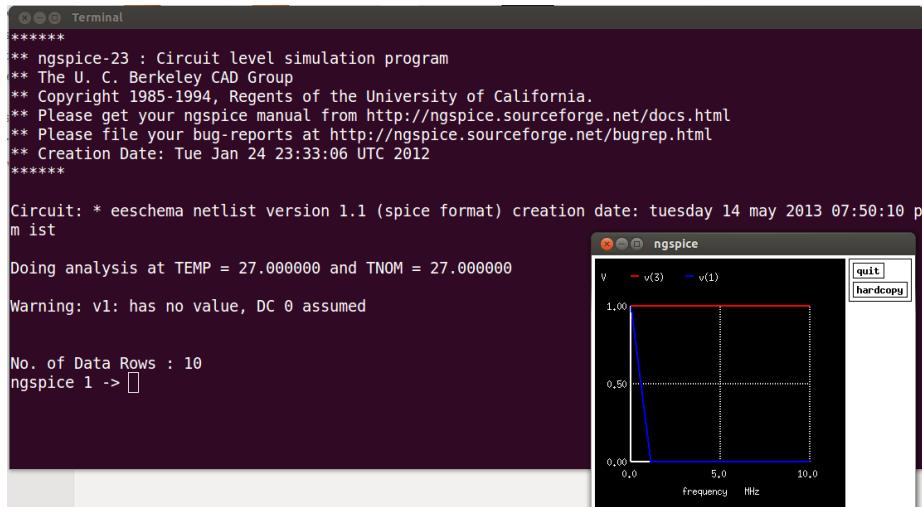


Figure 6.19: AC analysis example - simulation results

following details:

“start time” = 0 sec, “step time” = 1 ms, “stop time” = 20 ms and then click on Add simulation data. This is shown in Figure 6.20. Now click on Netlist converter which opens up a terminal. It asks for various parameters for the sine source. Enter the details as given below:

Offset value = 0, **Amplitude** = 1, **frequency** = 50, **delay time** = 0, **damping factor** = 0 and press ‘Enter’ key. This is shown in Figure 6.21. Now click on Ngspice tool which will shows Ngspice terminal with waveform window as shown in following figure 6.22.

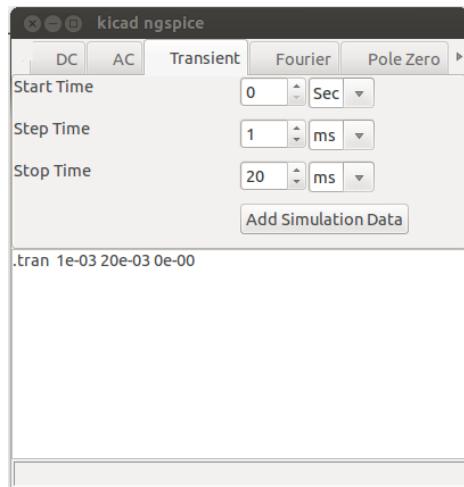


Figure 6.20: Transient analysis inserter

A screenshot of a terminal window titled "Terminal". The window displays the output of a script named "Kicad to Ngspice netlist converter". It shows the conversion of a file named "RC.cir" and includes a ".tran" command with parameters: 1e-03, 20e-03, and 0e-00. The script then prompts for parameters for a sine source "v1", asking for offset, amplitude, frequency, delay time, and damping factor. Finally, it informs the user that the ngspice netlist has been written to "RC.cir.out" and the scilab netlist has been written to "RC.cir.ckt". It ends with a prompt to "Press Enter to quit".

```
=====
Kicad to Ngspice netlist converter
=====
converting RC.cir
.tran 1e-03 20e-03 0e-00
-----
Add parameters for sine source v1
Enter offset value (Volts/Amps): 0
Enter amplitude (Volts/Amps): 1
Enter frequency (Hz): 50
Enter delay time (seconds): 0
Enter damping factor (1/seconds): 0
-----
The ngspice netlist has been written in RC.cir.out
The scilab netlist has been written in RC.cir.ckt
Press Enter to quit
```

Figure 6.21: Transient analysis -Netlist converter

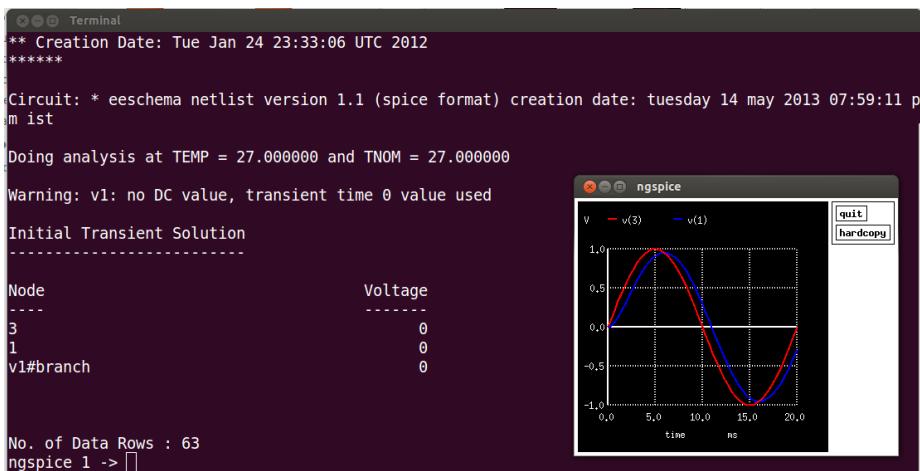


Figure 6.22: Transient analysis example - simulation results

Chapter 7

PCB Design

Printed Circuit Board (PCB) design is a very important step in electronic system design. Every component of your circuit needs to be placed and connections routed to minimize delay and area. Each component has an associated footprint. Footprint refers to the physical layout of a component that is required to mount it on the PCB. PCB design involves mapping footprints for all components, placing them appropriately to minimize wire length and area, connecting the footprints using tracks/vias and finally extracting the required files needed for printing the PCB. Let us see the steps to design PCB using Oscad.

7.1 Schematic Creation for PCB design

In chapter 5, we have seen the differences between schematic for simulation and schematic for PCB design. Let us take the example of an RC low pass circuit. We have used a resistor, capacitor, a sinusoidal voltage source, ground, plots, and power flag. Note that the sinusoidal voltage source and plot components were fictitious components added to let the simulator know what kind of simulation to do and what kinds of signals need to be plotted. These components are not needed for PCB design. You would need to connect an external signal generator instead of the fictitious sinusoidal source. You also do not need a plot component in a PCB. So let us modify the schematic of RC circuit created in chapter 5 for PCB design. Before doing this, you may want to make a back up copy of the files created in Chapter 5 to prevent them getting overwritten.

Open the RC circuit schematic in Schematic editor. Delete the sinusoidal voltage source and plot blocks and place a two pin connector, *CONN_2* from the EEschema library *conn*. You can use the delete key to delete components or you can right-click on the component and choose *Delete component*. See section 5.2.6 to know more about EEschema library *conn*. Connect the components, and make the final schematic as shown in Figure 7.1. Do the annotation and test for ERC. Please refer to chapter 5 to know more about basic steps in schematic creation.

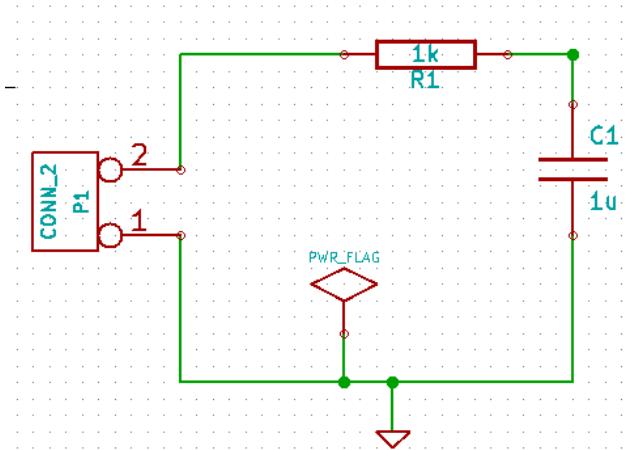


Figure 7.1: Final circuit schematic for RC low pass circuit

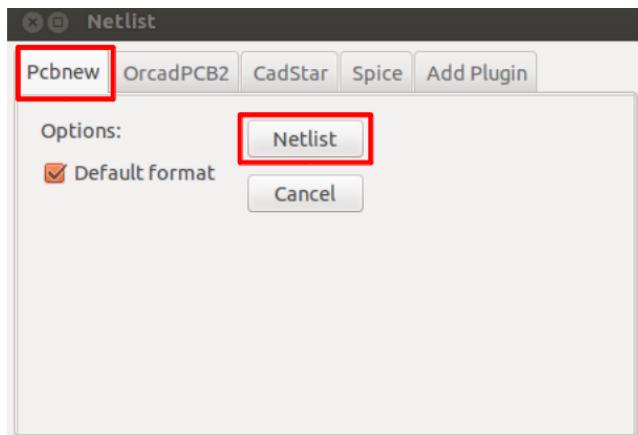


Figure 7.2: Netlist generation for PCB

7.1.1 Netlist generation for PCB

The netlist for PCB is different from that for simulation. To generate netlist for PCB, click on the *Generate netlist* tool from the top toolbar in Schematic editor. Click on the button *Netlist* under the tab *Pcbnew* in the Netlist window. This is shown in Figure 7.2. Click on *Save* in the Save netlist file dialog box that opens up. Do not change the directory or the name of the netlist file.

Note that the netlist has an extension ‘.net’. The netlist created for simulation had an extension ‘.cir’.

Save the schematic and close the schematic editor.

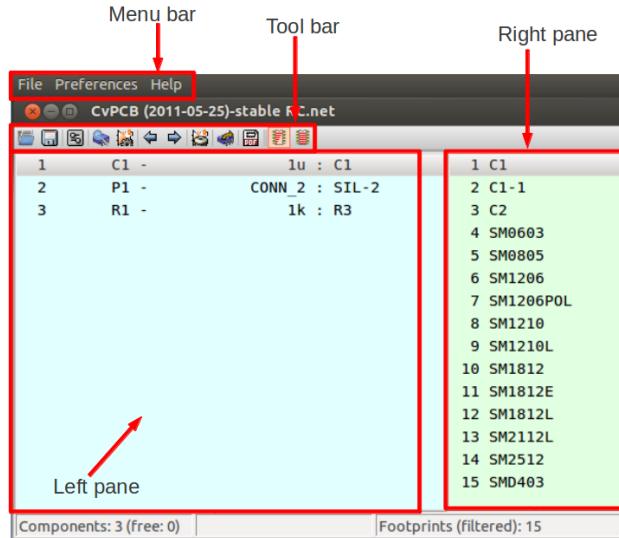


Figure 7.3: Footprint editor with the menu and tool bars and the left and right panes marked

7.1.2 Mapping of Components using Footprint Editor

Once the netlist for PCB is created, you need to map each component in the netlist with a footprint. The tool **Footprint Editor** is used for this. Oscad uses CvPcb as its footprint editor. CvPcb is the footprint editor tool in KiCad.

7.1.3 Familiarising the Footprint Editor tool

If you open the Footprint editor after creating the **.net** netlist file, you will get the Footprint editor as shown in Figure 7.3. The menu and tool bars and the panes are marked in this figure. The menu bar will be available in the top left corner. The left pane has a list of components in the netlist file and the right pane has a list of available footprints for each component.

Note that if you open the Footprint editor before creating a ‘.net’ file, then the left and right panes will be empty.

Tool bar

Some of the important tools in the tool bar are shown in Figure 7.4. They are explained below:

1. Save netlist and footprint files - Save the netlist and the footprints that you have associated with it.
2. View selected footprint - View the selected footprint in 2D. See section 7.1.4 for more details.

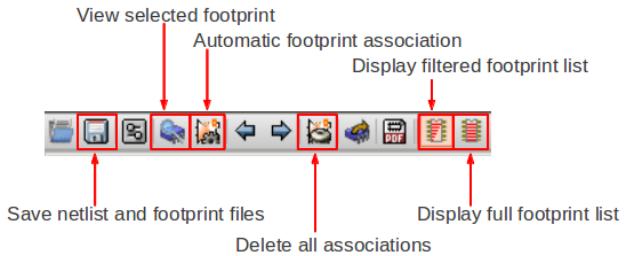


Figure 7.4: Some important tools in the tool bar

3. Automatic footprint association - Perform footprint association to each component automatically. Footprints will be selected from the list of footprints available.
4. Delete all associations - Delete all the footprint associations made
5. Display filtered footprint list - Display a filtered list of footprints suitable to the selected component
6. Display full footprint list - Display the list of all footprints available (without filtering)

7.1.4 Viewing footprints in 2D and 3D

To view a footprint in 2D, choose the footprint from the right pane and click on *View selected footprint* from the menu bar. Let us view the footprint for SM1210. Choose SM1210 from the right pane as shown in Figure 7.5. On clicking the View selected footprint tool, you will get the Footprint window with the view in 2D. Click on the *3D* tool in the footprint window, as shown in Figure 7.6. You will now get a top view of the selected footprint in 3D. Click on the footprint and rotate it using mouse to get 3D views from various angles. One such side view of the footprint in 3D is shown in Figure 7.7.

7.1.5 Mapping of components in the RC circuit

Click on C1 from the left pane. Choose the footprint *C1* from the right pane by double-clicking on it. Click on connector P1 from the left pane. Choose the footprint *SIL-2* from the right pane by double-clicking on it. Similarly choose the footprint *R3* for the resistor R1. The footprint mapping is shown in Figure 7.8. Save the footprint association by clicking on the *Save netlist and footprint files* tool form the toolbar. The *Save Net and component List* window appears. Browse to the directory where you have saved your schematic file for this project and click on *Save*. The netlist gets saved and the Footprint editor window closes automatically.

Note that you need to browse to the directory where you saved the schematic file to save the ‘.net’ file.

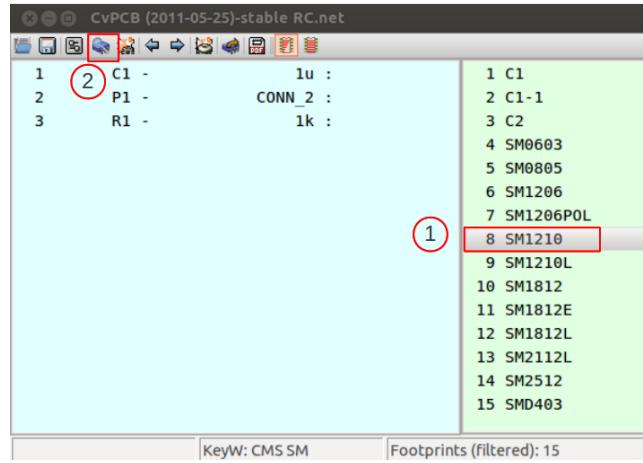


Figure 7.5: 1. Choose the footprint SM1210 from the right pane, then 2. click on View selected footprint

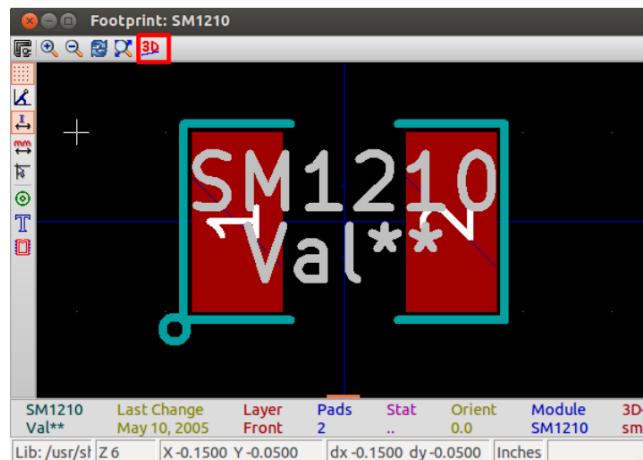


Figure 7.6: Footprint view in 2D. click on 3D to get 3D view

7.2 Creation of PCB Layout

The next step is to place the footprints and lay tracks between them to get the layout. This is done using the **Layout Editor** tool. Oscad uses **Pcbnew**, the layout creation tool in KiCad, as its layout editor.

7.2.1 Familiarising the Layout Editor Tool

The Layout editor with the various menu and tool bars is shown in Figure 7.9.

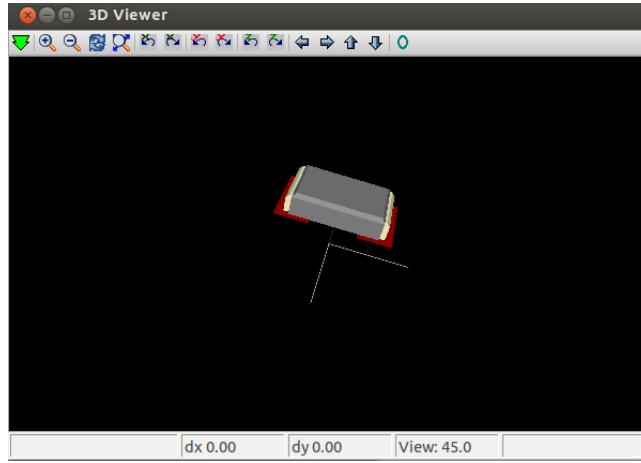


Figure 7.7: Rotate the footprint upwards to get the 3D view from a different angle

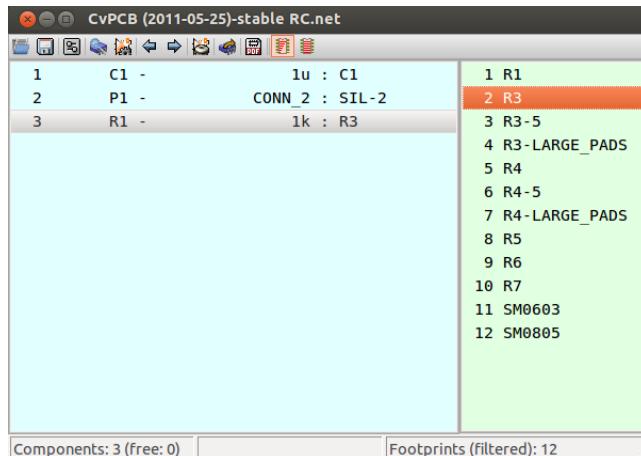


Figure 7.8: Footprint mapping done

Top tool bar

Some of the important menu options in the top menu bar are shown in Figure 7.10. They are explained below:

1. Save board - Save the printed circuit board
2. Module editor - Open module editor to edit footprint modules or libraries
3. Read Netlist - Import the netlist whose layout needs to be created.

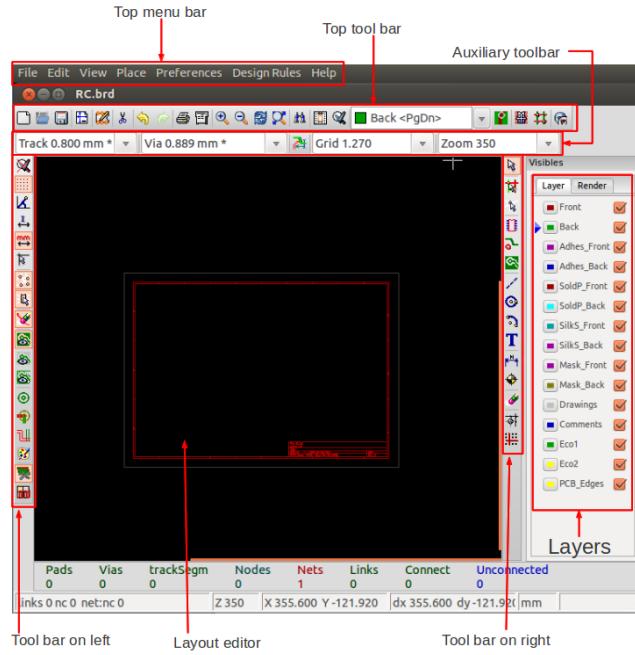


Figure 7.9: Layout editor with menu and tool bars and the layer options marked

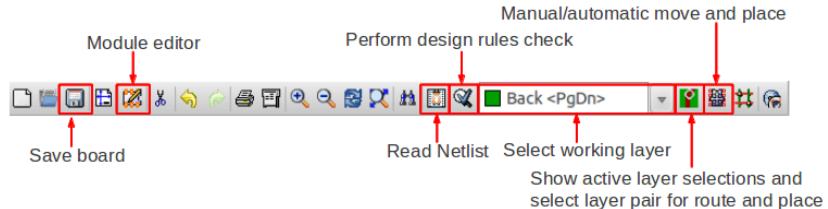


Figure 7.10: Top toolbar with important tools marked

4. Perform design rules check - Check for design rules, unconnected nets etc. in the layout made.
5. Select working layer - Selection of working layer
6. Show active layer selections and select layer pair for route and place - Select layer in top and bottom layers. It also shows currently active layer selections.
7. Mode footprint: Manual/automatic move and place - Move and place modules

7.2.2 Hotkeys

A list of hotkeys are given below:

1. F1 - Zoom in
2. F2 - Zoom out
3. Delete - Delete Track or Footprint
4. X - Add new track
5. V - Add Via
6. M - Move Item
7. F - Flip Footprint
8. R - Rotate Item
9. G - Drag Footprint
10. Ctrl+Z - Undo
11. E - Edit Item

The list can be viewed by selecting *Preferences* from the top menu bar and choosing *List Current Keys* from the option *Hotkeys*.

7.2.3 PCB design example using RC circuit

Click on Layout Editor from the Oscad toolbar. Click on *Read Netlist* tool from the top tool bar. Click on *Browse Netlist files* on the Netlist window that opens up. Select the *RC.net* file that we have saved after assigning footprints. Click on *Open*. Now Click on *Read Current Netlist* on the Netlist window. The message area in the Netlist window says that the *RC.net* has been read. The sequence of operations is shown in Figure 7.11.

The footprint modules will now be imported to the top left hand corner of the Layout Editor. This is shown in Figure 7.12. Zoom in to the top left corner by pressing the key **F1** or using the scroll button of your mouse. The Zoomed in version of the imported netlist is shown in Figure 7.13. Let us now place this in the center of the Layout editor window. Click on *Mode footprint: Manual/automatic move and place* tool from the top tool bar. Place the cursor near the center of the layout editor window. Right click and choose *Glob move and place*. Choose *move all modules*. The sequence of operations is shown in Figure 7.14. Click on yes on the confirmation window, as shown in Figure 7.15, to move the modules. Zoom in using the **F1** key. The current placement of components after zooming in is shown in Figure 7.16.

We need to arrange the modules properly to lay tracks. Rotate the connector P1 by placing the cursor on top of P1 and pressing **R**. Move it by placing the

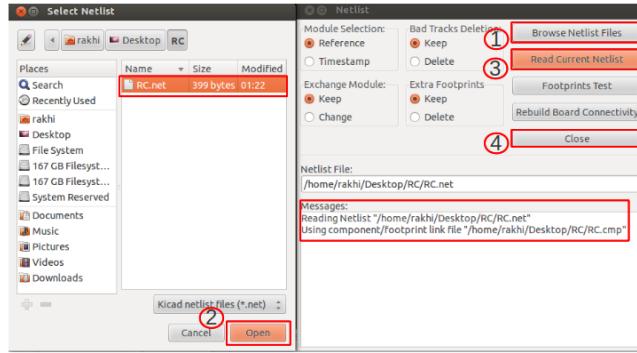


Figure 7.11: Importing netlist file to layout editor. 1. Browse netlist Files 2. Choose the RC.net file 3. Read Netlist file 4. Close

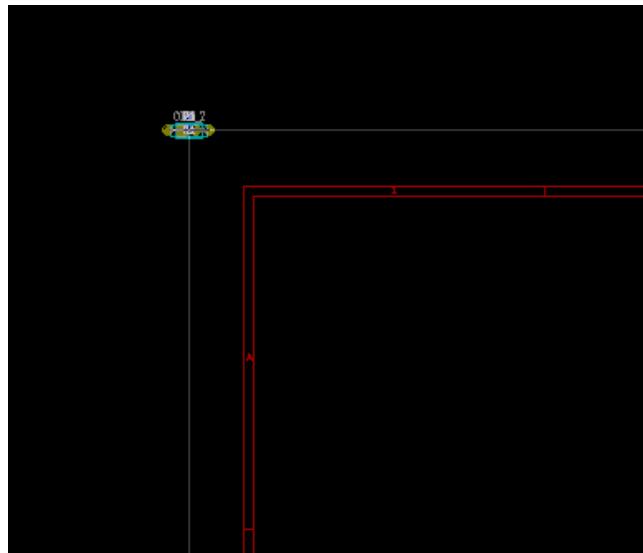


Figure 7.12: Footprint modules imported to top left corner of Layout Editor

cursor on top of it and pressing M. The final placement is shown in Figure 7.17.

Let us now lay the tracks. Let us first change the track width. Click on *Design rules* from the top menu bar. Click on design rules. This is shown in Figure 7.18. The *Design Rules Editor* window opens up. Here you can edit the various design rules. Double-click on the track width field to edit it. Type 0.8 and press Enter. Click on ok. Figure 7.19 shows the sequence of operations.

Click on *Back* from the *Layer* options as shown in Figure 7.20.

Let us now start laying the tracks. Place the cursor above the left terminal of R1 in the layout editor. Press the key x. Move the cursor down and double-

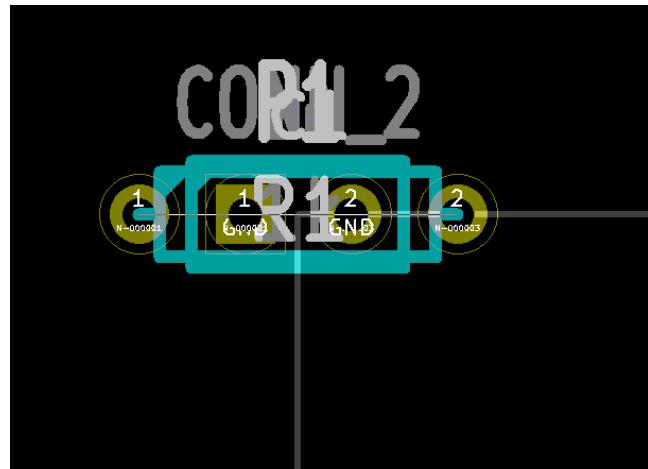


Figure 7.13: Zoomed in version of the imported netlist

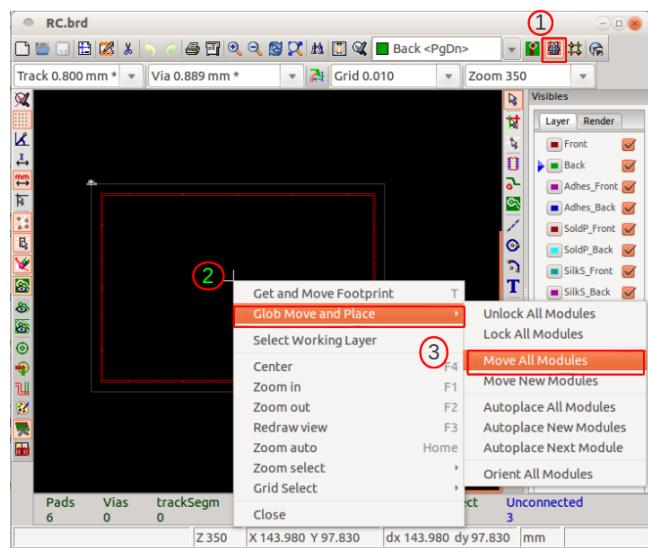


Figure 7.14: Move and place modules to center of layout editor. 1. Click on Mode footprint: Manual/automatic move and place 2. Place cursor at center of layout editor and right click on it 3. Choose Glob move and place and then choose move all modules.

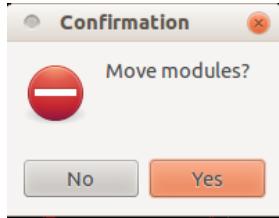


Figure 7.15: Click on Yes in the confirmation window

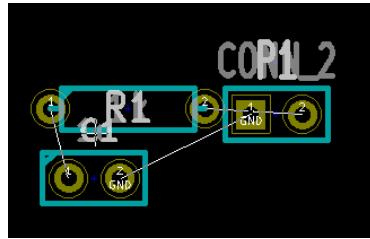


Figure 7.16: Zoomed in version of the current placement after moving modules to the center of the layout editor

click on the left terminal of C1. A track is formed. This is shown in Fig. 7.21.

Similarly lay the track between capacitor C1 and connector P1 as shown in Fig. 7.22. The last track needs to be laid at an angle. To do so, place the cursor above the second terminal of R1. Press the key **x** and move the cursor diagonally down. Double-click on the other terminal of connector. The track will be laid as shown in Fig. 7.23. All tracks are now laid. The next step is create PCB edges.

Choose *PCB_edges* from the Layer options to add edges. Choose *Add graphic line or polygon* from the toolbar on the left. Fig. 7.24 shows the sequence of operations. Let us now start drawing edges for PCB. Click to the left of the layout. Move cursor horizontally to the right. Click once to change orientation. Move cursor vertically down. Draw the edges as shown in Figure 7.25. Double-click to finish drawing edges.

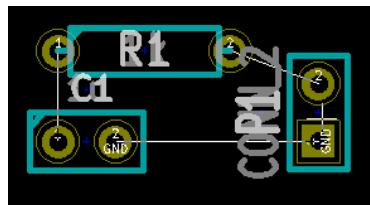


Figure 7.17: Final placement of footprints after rotating and moving P1

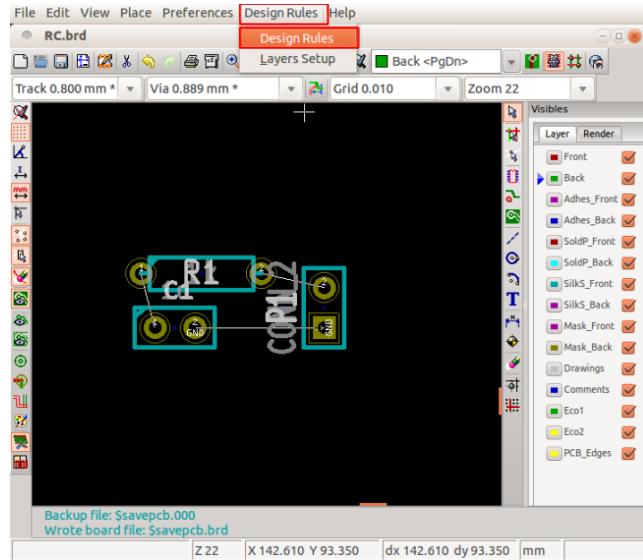


Figure 7.18: Choose Design rules from the top menu bar and choose design rules

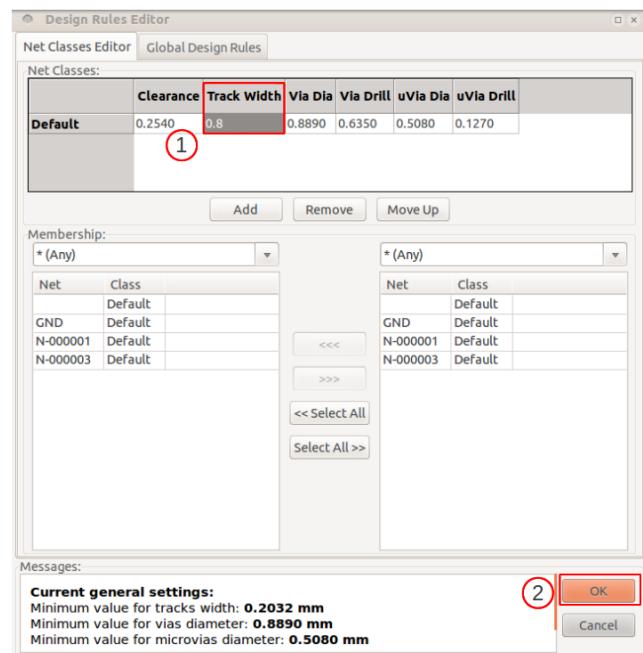


Figure 7.19: 1. Double-click on track width field and type 0.8, 2. Click on ok

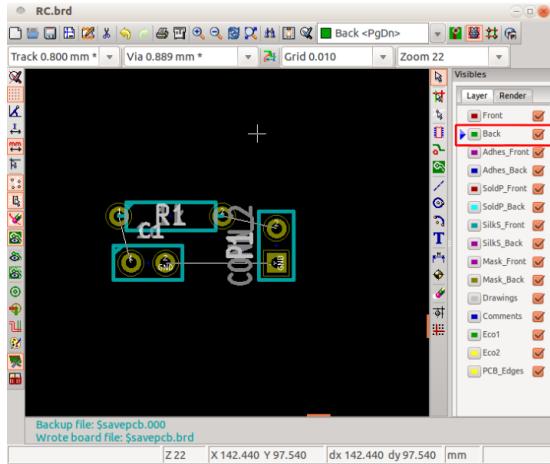


Figure 7.20: Choose the copper layer ‘Back’

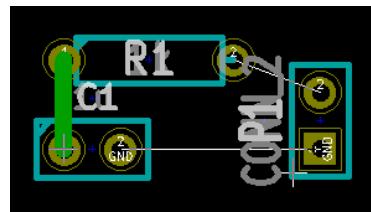


Figure 7.21: A track formed between resistor and capacitor

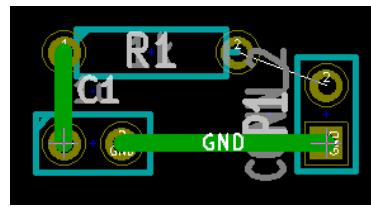


Figure 7.22: A track formed between capacitor and connector

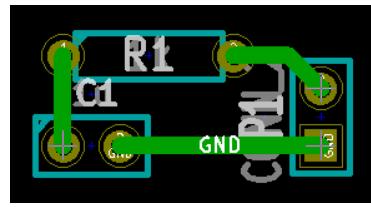


Figure 7.23: A track formed between connector and resistor

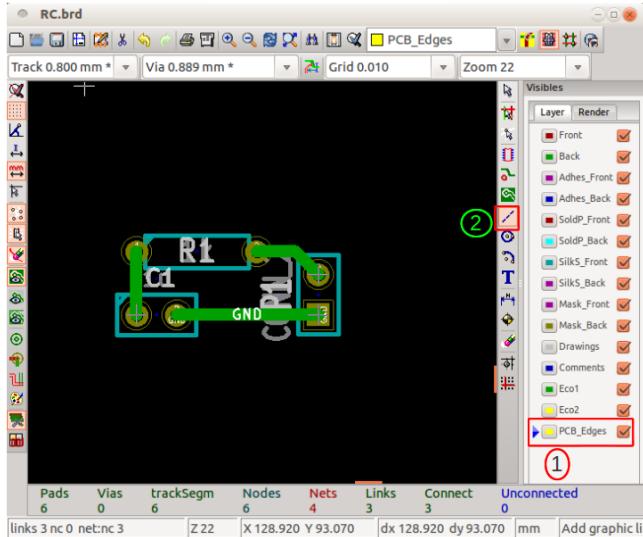


Figure 7.24: 1. Choose PCB_edges from Layer options 2. Choose Add graphic line or polygon from left toolbar

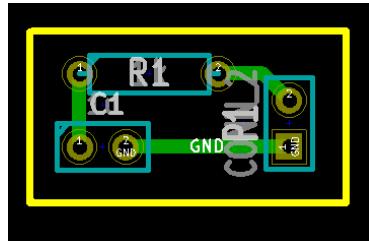


Figure 7.25: PCB edges drawn

Click on *Perform design rules check* from the top tool bar to check for design rules. The *DRC Control* window opens up. Click on *Start DRC*. There are no errors under the Error messages tab. Click on ok to close DRC control window. The figure 7.26 shows the sequence of operations.

Click on *Save board* on the top toolbar. To generate gerber files, click on *File* from the top menu bar. Click on *plot*. This is shown in Figure 7.27. The plot window opens up. You can choose which layers to plot by selecting/deselecting them from the left side. You can also choose the format to plot. Choose Gerber. The output directory of the plots created can also be chosen. By default, it is the project directory. Some more options can be chosen in this window. Click on *Plot*. The message window shows the location in which Gerber files are created. Click on *Close*. This is shown in Figure 7.28.

The PCB design of RC circuit is now complete. If you would like to know more about Pcbnew, please refer [3].

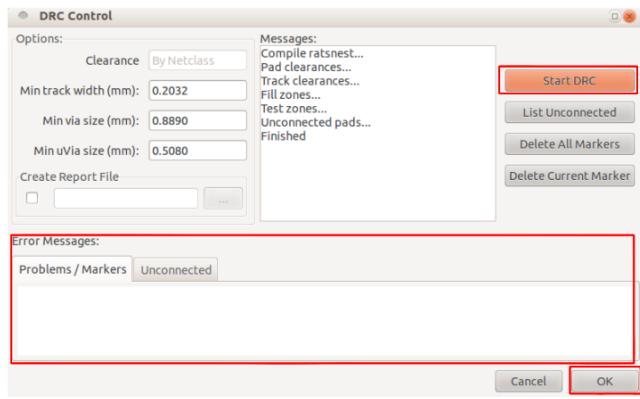


Figure 7.26: 1. Click on Start DRC 2. Click on Ok

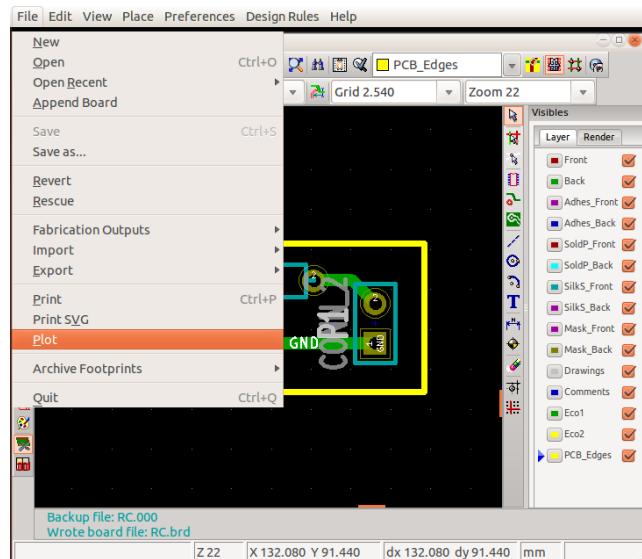


Figure 7.27: Choose plot from the File menu

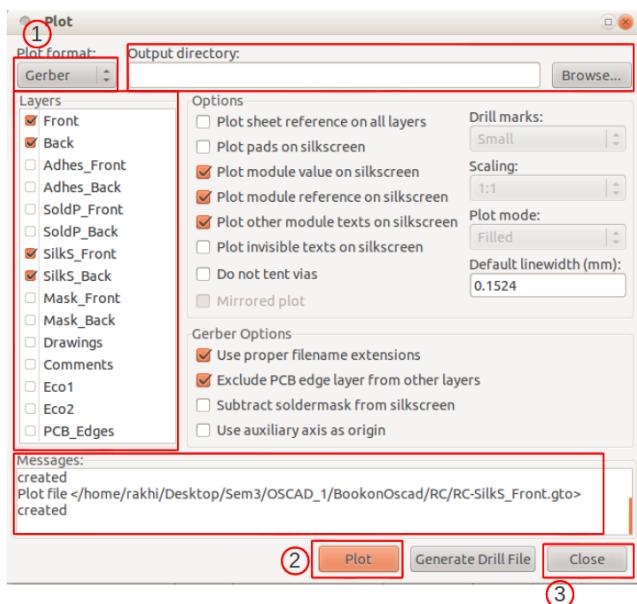


Figure 7.28: 1. Choose Gerber as the plot format 2. Click on Plot. Message window shows location in which Gerber files are created 3. Click on Close

Chapter 8

Model builder and Subcircuit builder

8.1 Model builder

Spice based simulators include a feature which allow accurate modeling of semiconductor devices such as diodes, transistors etc. Oscad Model Builder provides a facility to define a new model for devices such as diodes, MOS, BJT, JFET, IGBT, Magnetic core etc. Model Builder in Oscad asks for the value of parameters depending on the type of the device for which a model is required. The parameter values can be obtained from a datasheet of the device. A newly created model can be exported to the model library and one can import it for different projects, whenever required. Model Builder also provides a facility to edit existing models.

8.1.1 Example

Let us take an example of Bridge rectifier circuit containing diode model 1N4007 and see how Model Builder works in Oscad. Refer figure 8.1 for the Bridge rectifier circuit schematic.

First create the circuit schematic of the Bridge Rectifier shown in figure 8.1 as explained in chapter 5. In this schematic, change reference field of 1n4007 to “D”. Here reference field “D” indicates that it is a diode model. Generate the spice netlist.

Now to build a new model for the diode 1N4007, click on the “Model Builder” from the toolbar of Oscad. It opens up “Model Select” window which shows 1N4007. Since we are going to create a new model, we will click on ”cancel” button as shown in figure 8.2. Then click on “New” from “File” drop-down menu as shown in figure 8.3. An “Ngspice Model Editor” window will open up. In the “Enter Component name” field, type “1N4007”. In the ”Enter type of Component” option, select “Diode” and finally click on “OK” button. This

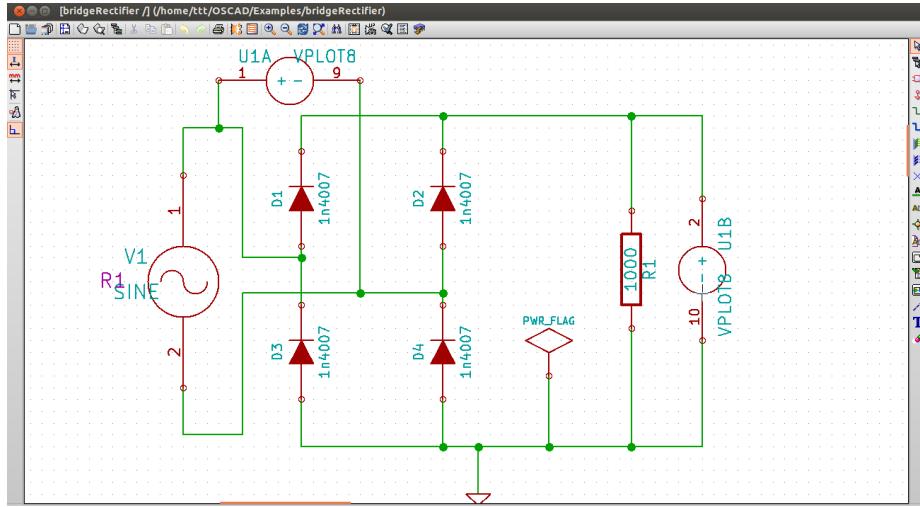


Figure 8.1: Bridge rectifier circuit schematic

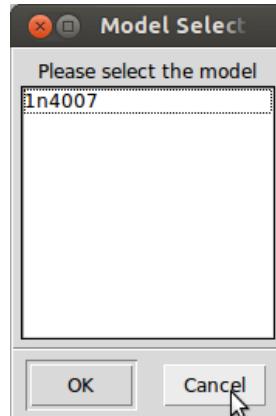


Figure 8.2: Cancel selecting existing model

window is illustrated in figure 8.4. Then you will get a window where it asks for the value of model parameters for diode 1N4008 such as reverse breakdown voltage (BV), ohmic resistance (RS) etc. as shown in Figure 8.5. Here you can change the values of the model parameters for 1N4007 diode model as it is given in the datasheet and then click on OK button to save it.

Once a new diode model of 1n4007 is created it can be exported to the model library as shown in the Figure 8.6. Whenever required, one can also import it for different projects as shown in the figure 8.7 and 8.8.

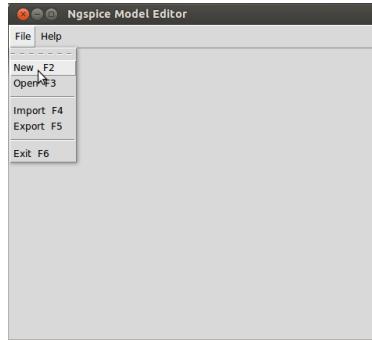


Figure 8.3: Select new Model

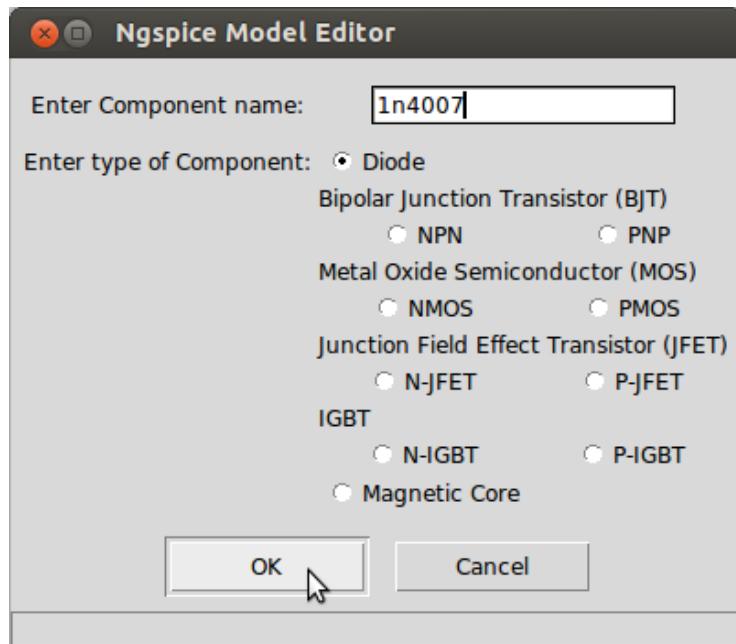


Figure 8.4: Select type of model

8.2 Subcircuit Builder

Subcircuit is a way to implement hierarchical modeling. Once a subcircuit for a component is created, it can be used for different circuits. Oscad provides an easy way to create a subcircuit in steps.

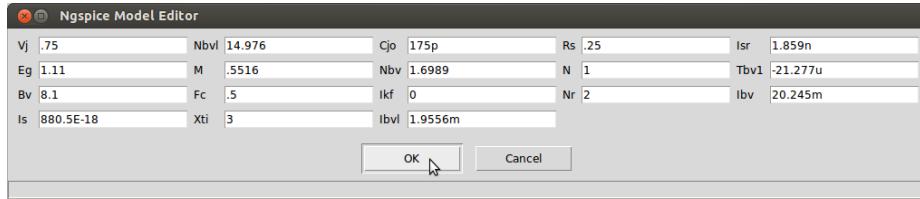


Figure 8.5: Edit model paramters

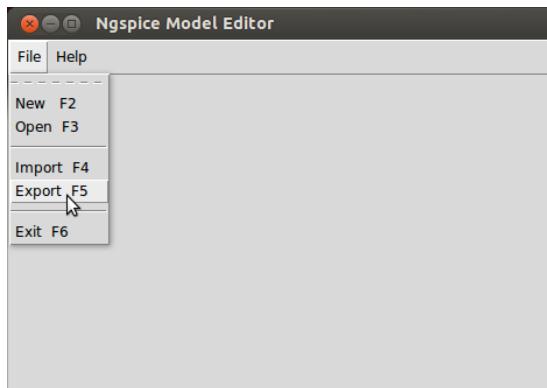


Figure 8.6: Export model

8.2.1 Example

Let us take an example of building a subcircuit of IC 555 timer which is a part of Astable Multivibrator circuit.

Create the schematic of the Astable multivibrator as shown in the figure 8.9. Before making the subcircuit for components, make sure that the “Reference Field” of these component should always be “X”. Let us see how to change “Reference Field” of a component. Right click on U? of the LM555N IC and choose “Field Reference”. This is shown in figure 8.10. Then choose “Edit field” as shown in figure 8.11. Change the reference field of LM555N from U? to X. Here the reference field X denotes that LM555N can have a subcircuit. This is shown in figure 8.12

Now to create Subcircuit, click on the Sub Circuit Builder option from the Tool bar of Oscad. You will get a window displaying the list of “Field Values” of components whose “Reference Field” value is “X”. It is shown in Figure 8.13.

Once you select the subcircuit it opens up the Schematic Editor where you can draw the schematic of the subcircuit. After you finish creating the subcircuit, connect a port to the external pins of the subcircuit. Port can be found under the “Place a component” option. The subcircuit of LM555N is shown in

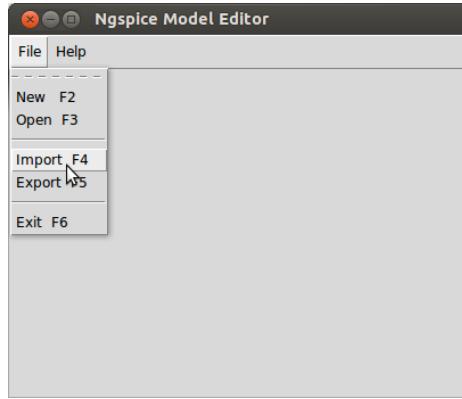


Figure 8.7: Import model-1

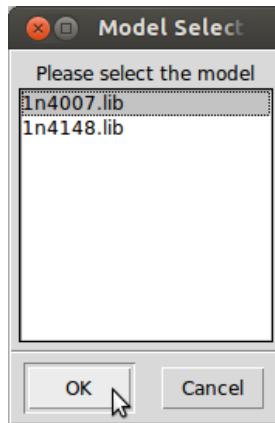


Figure 8.8: Import model-2

Figure 8.14. Once you completed the creation of the subcircuit save it in the respective project folder and close the Schematic editor.

When you close the Schematic Editor, a terminal window will pop-up. It will ask you to enter the value for the different parameters of subcircuit as shown in the Figure 8.15.

At the end of the terminal, after you entered all the parameters, you will see the message:

```
The ngspice netlist has been written in lm555n.cir.out  
The scilab netlist has been written in lm555n.cir.ckt.  
Press Enter to quit
```

This message means that netlist for the subcircuit is created. It is shown in the Figure 8.16.

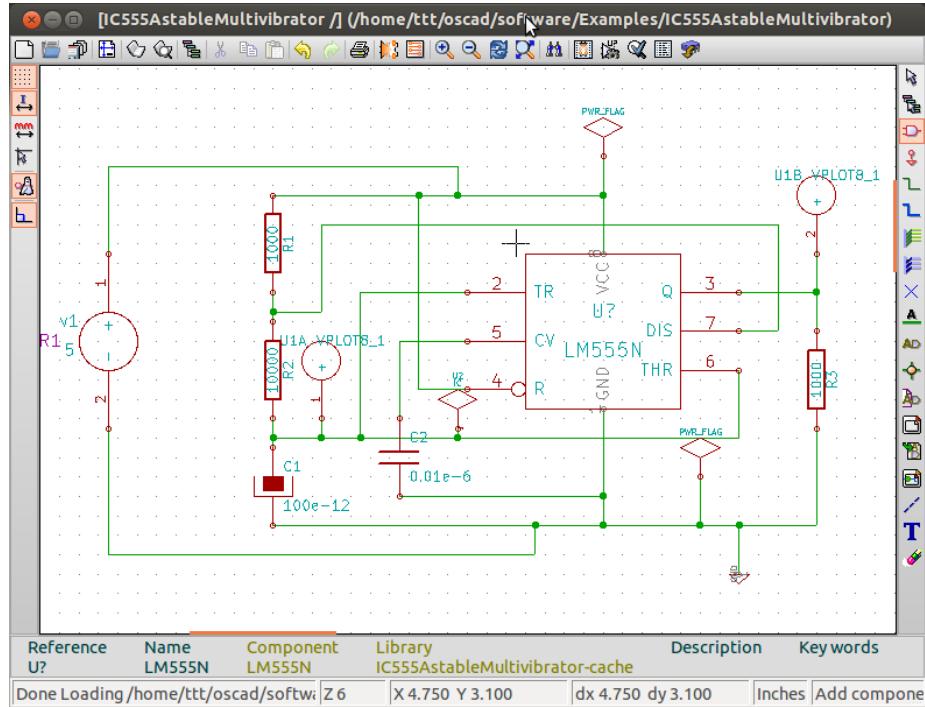


Figure 8.9: Astable multivibrator circuit Schematic

Finally it gives a pop-up saying “Created subcircuit lm555n.sub”. It is shown in figure 8.17. Click on OK. This completes the subcircuit creation.

Once subcircuit is created, it can be exported to the subcircuit library as shown in Figure 8.18 and whenever required one can import it for different projects as shown in the figure 8.19.

Finally close the subcircuit editor window.

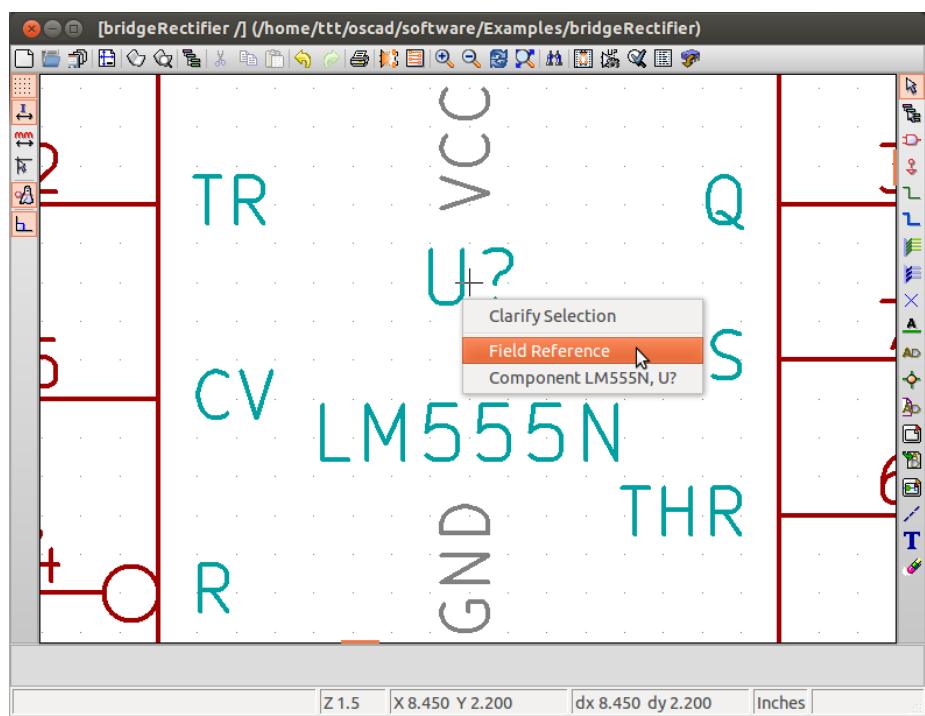


Figure 8.10: Choosing field reference of LM555N

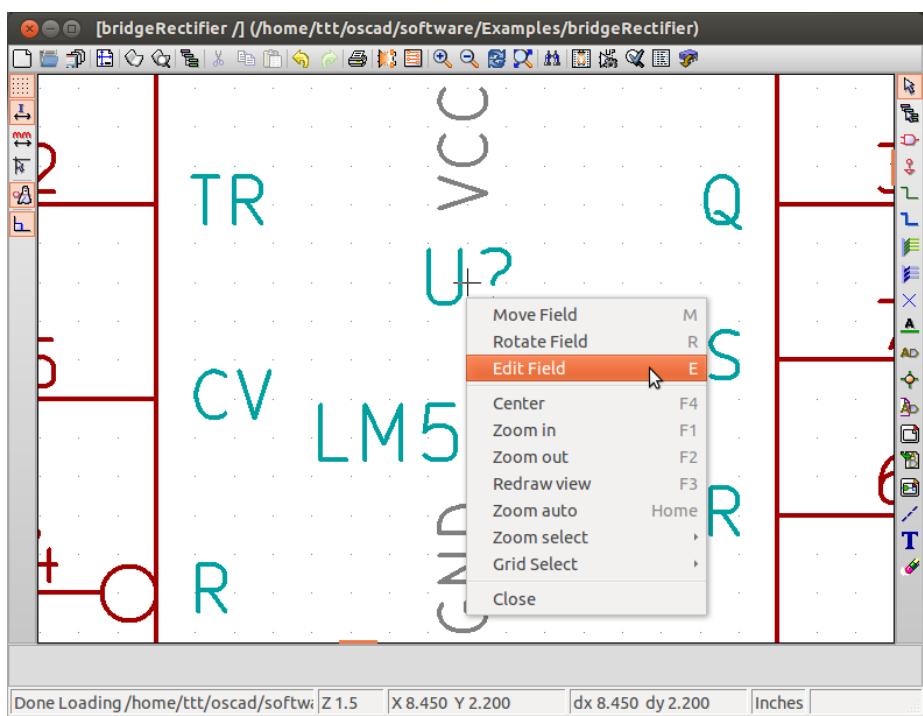


Figure 8.11: Choosing Edit field of LM555N

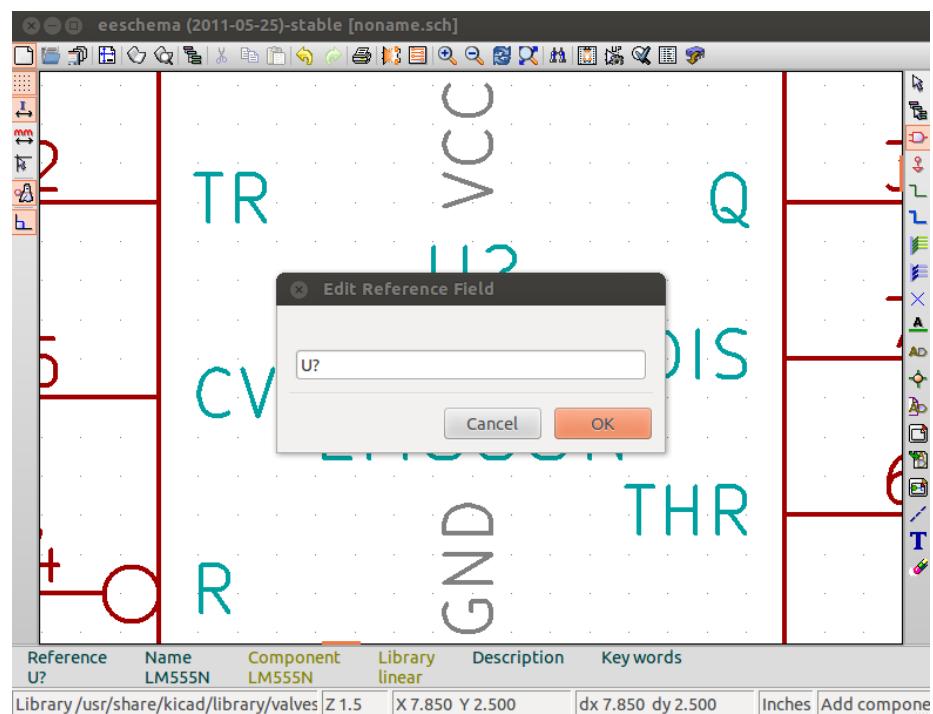


Figure 8.12: Changing reference field of LM555N from U? to X

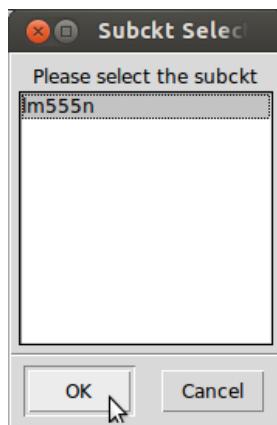


Figure 8.13: Select Model

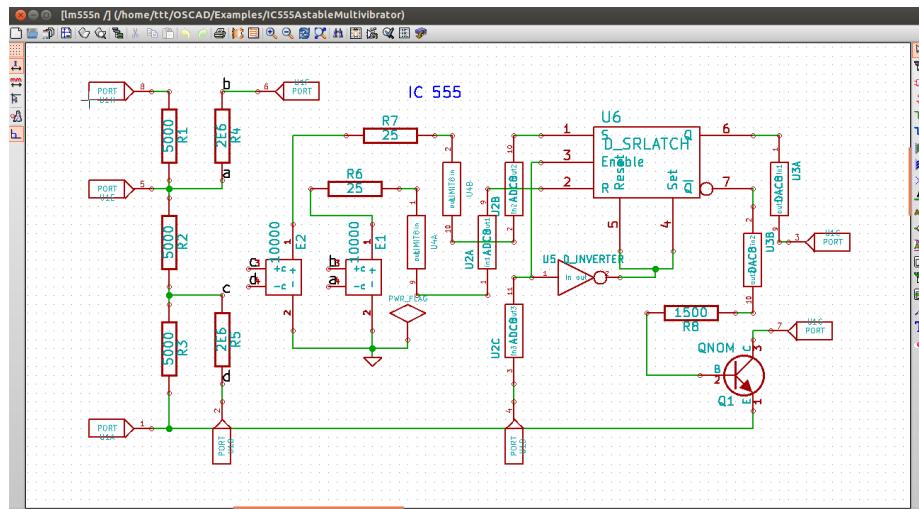


Figure 8.14: IC 555 timer subcircuit

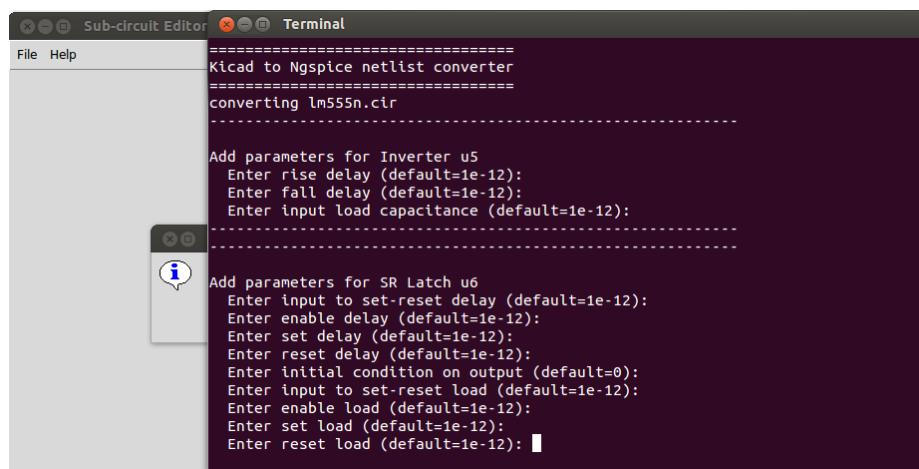


Figure 8.15: IC 555 timer subcircuit - parameters

```
Add parameters for Limiter u4
Enter out lower limit (default=0.0):
Enter out upper limit (default=5.0):
Enter offset for input (default=0.0):
Enter gain (default=1.0):
-----
Add parameters for digital to analog converter u3
Enter output low level voltage (default=0.2):
Enter output high level voltage (default=5.0):
Enter output for undefined voltage level (default=2.2):
-----
Add parameters for analog to digital converter u2
Enter input low level voltage (default=0.8):
Enter input high level voltage (default=2.0):
-----
The ngspice netlist has been written in lm555n.cir.out
The scilab netlist has been written in lm555n.cir.ckt
Press Enter to quit
```

Figure 8.16: Netlist conversion

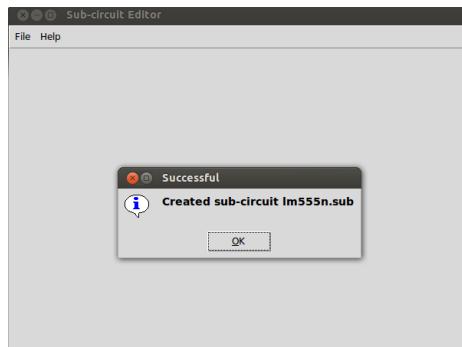


Figure 8.17: Subcircuit creation successfull

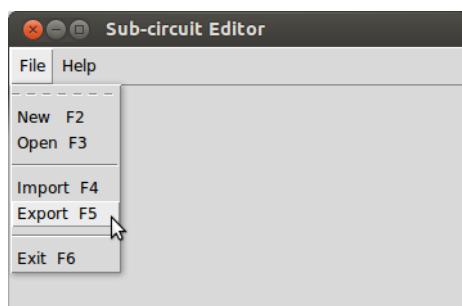


Figure 8.18: Export Subcircuit

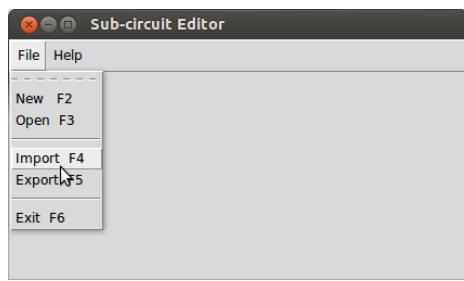


Figure 8.19: Import Subcircuit

Chapter 9

Scilab Based Circuit Simulation

Electronic circuit simulation uses mathematical models to replicate the behavior of an electronic circuit. Unfortunately, no simulator gives the system of equations it solves, in order to understand the simulation. In Oscad there is an option to simulate the circuit using SMCSim (Scilab Based Mini Circuit Simulator). An important feature of SMCSim is that it gives the system of equations for the circuit under test. The SMCSim works in three modes: normal, symbolic and numerical mode. In normal mode, SMCSim solves the circuit and gives the final output. In symbolic mode, it gives symbolic equations along with the result. In numerical mode, it gives symbolic equations, intermediate numerical values of the components and entries in system matrices, and the final output. Here, we present the working and implementation of SMCSim with an example.

Consider Half-Wave Rectifier with Filter shown in Figure 9.1. The circuit is drawn using Eeschema integrated with Oscad and spice compatible netlist is generated using Oscad circuit simulation tools. The generated netlist is given below. Note that this netlist is generated for SMCSim which has a simple model implementation for a diode. Thus, users need to specify only I_s and V_t values.

```
* Half-Wave Rectifier
V1 1 0 sine (5 50)
D1 1 2 mymodel (1e-8 0.026)
R1 2 0 10000
C1 2 0 10e-3
.tran 0 100 0.5
.plot v(1) v(2)
.end
```

SMCSim first reads the netlist and creates a graph corresponding to it using Scilab based graph library metanet. Then circuit is translated into circuit equations using a circuit equations formulation method. All the methods of for-

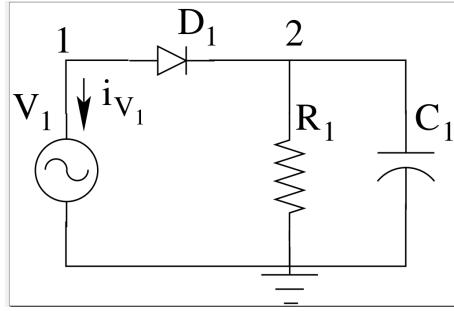


Figure 9.1: Bridge rectifier circuit

mulating circuit equations use Kirchhoffs current and voltage equations (KCE and KVE), and device characteristics constraints but differ in the manner in which these constraints are imposed. We have used Modified Nodal Analysis (MNA) as it is applicable to all kinds of electrical circuits. Using MNA method and graph operations, we have efficiently built the circuit equations. The system of Equations representing the electrical circuit shown in Figure 4 is given below. SMCSim has capability to display these equations.

$$i_{V_1} + D_{1f}(v_1, v_2) = 0 \quad (9.1)$$

$$(R_1)v_2 + (C_1)\frac{dv_2}{dt} + -D_{1f}(v_1, v_2) = 0 \quad (9.2)$$

$$v_1 = V_1 \quad (9.3)$$

$$D_{nf}(v_a, v_b) = I_{sn}(1 - e^{(v_a - v_b)/vt_n})$$

where I_{sn} =reverse saturation current and vt_n =threshold voltage of diode n

Now, we explain how SMCSim performs Operating point (DC) analysis and Transient analysis.

9.1 Operating Point (DC) Analysis

: A circuit can reach an equilibrium point only when stimulus is constant. So, first step of operating point analysis is to configure the independent sources such that they are constant. Since all waveforms are constant-valued at equilibrium points, $dv/dt = 0$ and $di/dt = 0$ and so capacitors act as open circuits and inductors act as short circuits. Thus, for operating point analysis, SMCSim removes the time-dependent components properly. The equations that describe the resulting system are nonlinear and algebraic and solution gives equilibrium point. The equations for the circuit are given below:

$$i_{V_1} + D_{1f}(v_1, v_2) = 0 \quad (9.4)$$

$$(R_1)v_2 - D_{1f}(v_1, v_2) = 0 \quad (9.5)$$

$$v_1 = V_1 \quad (9.6)$$

$$D_{nf}(v_a, v_b) = Is_n(1 - e^{(v_a - v_b)/vt_n})$$

where Is_n =reverse saturation current and vt_n =threshold voltage of diode n

Generally for nonlinear devices, a linear model is constructed that is valid only locally around a point. We have used Newton-Raphson method to construct a linear model for a nonlinear devices. In the example, diode D1 is a nonlinear device. SMCSim constructs the linear model for diode D1 as shown in the Figure 9.2. Note that the value of resistor and current source changes at every iteration and SMCSim allows user to observe the value. This is very useful for debugging the circuit when the simulation is not converging. The system of equations representing the linearized electrical circuit is given below:

$$(R_{D_1})v_1 + (-R_{D_1})v_2 + i_{V_1} = -i_{D_1} \quad (9.7)$$

$$(R_{D_1})v_1 + (R_{D_1} + R_1)v_2 = i_{D_1} \quad (9.8)$$

$$v_1 = V_1 \quad (9.9)$$

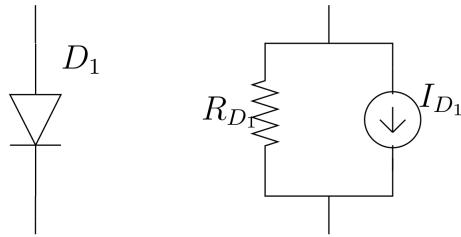


Figure 9.2: Diode model linearisation

9.2 Transient Analysis

In transient analysis, time dependent components are discretized, i.e., for dynamic devices, a static model is constructed, using a numerical integration method, that is valid for a particular time point. The circuit contains capacitor C1 , as a dynamic device. To solve the circuit, SMCSim constructs a static model for the capacitor C1 using Backward Euler method and performs operating point analysis for a time instant t . The operating point solution gives the solution at time instant t . Note that for each time instant, the values of the static model and the voltage source change.

9.2.1 Example

Now let us take an example of Bridge Rectifier circuit shown in the schematic given in Figure 9.3 and see how to do Scilab based circuit simulation

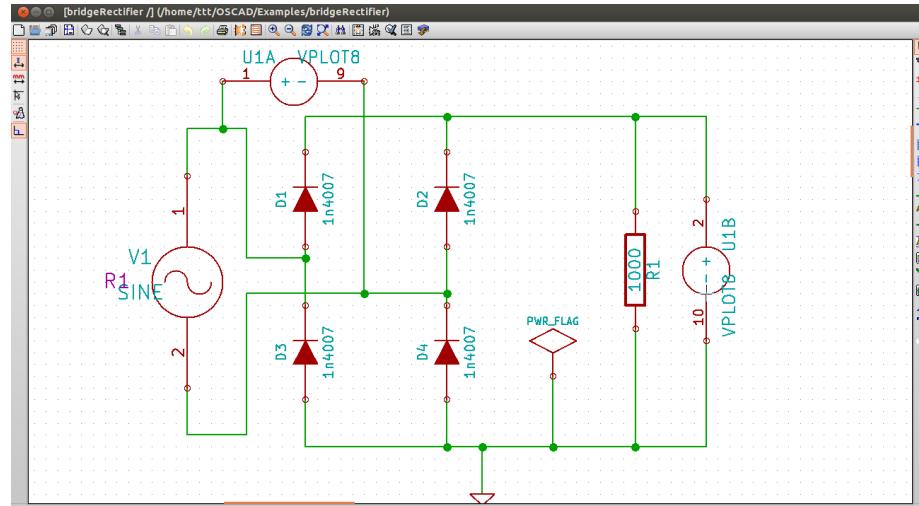


Figure 9.3: Bridge rectifier circuit schematic

To do Scilab based simulation you will have to follow the same steps of simulation that you have followed in Ngspice simulation except for the last step. Here, instead of clicking on the Ngspice, click on the SMCSim from tool bar of Oscad as shown in Figure 9.4.



Figure 9.4: Select SMCSim from Oscad toolbar

Then you will be able to see the output of Scilab based simulations as shown in Figure 9.5. Here you can see that Scilab based simulations not only gives you the plot for the output but also gives the system of equations for the circuit. This is extremely helpful for students in improving their knowledge of circuit simulation.

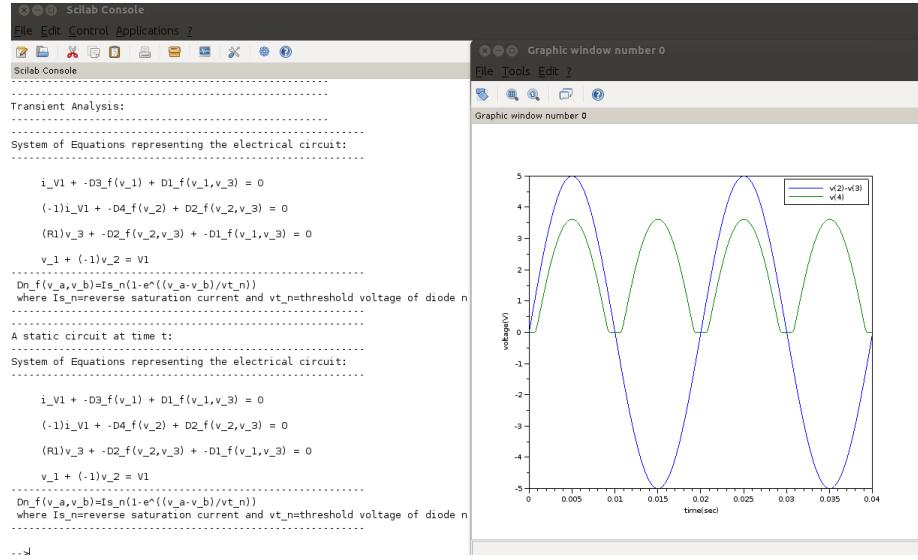


Figure 9.5: Scilab based simulation output

Chapter 10

Oscad Spoken Tutorials

Apart from learning how to use Oscad from this book, one can also refer to spoken tutorials (audio-video tutorials) created for Oscad. They are available at <http://spoken-tutorial.org> & <http://oscad.net>. These tutorials are basically made for self learning and are very clear and detailed. Spoken Tutorial based SELF workshops on OSCAD comprises the following tutorials in the order mentioned below. They are categorized in to two levels, Beginner Level and Advanced Level.

10.1 Beginner Level

The beginner level has a set of tutorials on OSCAD and KiCad. As OSCAD uses KiCad for schematic creation and PCB layout design, the supporting tutorials on KiCad enhances the user's understanding of OSCAD.

10.1.1 Introduction to Oscad

This tutorial aims at making the users get started with OSCAD. It covers the following:

- In this tutorial installation of OSCAD through a shell script is shown.
- This script installs all the requisite software like Ngspice, KiCad, Scilab and Python.
- After installation of all software, an already created schematic of an RC filter is opened.
- A test run of Oscad is done using this circuit.

10.1.2 Schematic creation and simulation using Oscad

This tutorial teaches how to create circuit schematic and simulate it using Oscad. A simple RC filter circuit is used as an example. The following sequence is adopted in the tutorial.

- Required components are chosen from their corresponding libraries and placed in the schematic editor.
- Components are connected together, annotated and values are assigned to them.
- Electric Rules Check is done and erroneous connections are corrected, if any.
- Spice netlist is generated and is converted to NGSpice format.
- Circuit simulation is done using NGSpice.

10.1.3 Designing Circuit Schematic in KiCad

- This tutorial introduces KiCad.
- It teaches how to create a circuit schematic using EESchema and annotate various components in the schematic.
- Astable multivibrator circuit is used as an example.
- An assignment is given in the end for practice.

10.1.4 Designing Printed Circuit Board using Oscad

- Create netlist for PCB from schematic.
- Map footprints to components.
- Generate PCB layout.
- PCB layout of RC filter is created in this tutorial.

10.1.5 Electric rule checking and Netlist Generation in KiCad

This tutorial teaches the following

- To assign values to components in the astable multivibrator circuit schematic created in the previous tutorial
- To perform electric rule check.
- To generate netlist for designing PCB layout

10.1.6 Mapping components in KiCad

- This tutorial explains how to map components in a schematic with corresponding footprints.
- Cvpcb, the footprint editor in Kicad, is used to explain the same.
- Every component in the astable multivibrator circuit schematic is assigned a footprint

10.1.7 Designing PCB in KiCad

- In this tutorial, printed circuit board layout of the astable multivibrator circuit is created.
- It also explains how to lay the tracks, modify the width of the tracks etc.
- Layer selection and track routing are also covered.

10.2 Advanced Level

Advanced level has tutorials on Ngspice, model building using Oscad and sub-circuit creation using Oscad. As Oscad uses Ngspice for simulation, the set of tutorials on Ngspice helps the user to know more about how simulations are done in Oscad.

1. Operating point analysis in Ngspice

This tutorial explains

- How to perform operating point analysis.
- How to verify Kirchoff's voltage law using ngspice in, interactive mode using commandline interface & using command script included in netlist.

2. DC sweep analysis in Ngspice

This tutorial covers the following

- How to perform DC sweep analysis.
- How to perform nested DC sweep analysis using two sweep variables.

3. Model building using Oscad

In this tutorial, we show how to build a model for a diode. This includes

- Opening an already created circuit schematic of bridge rectifier
- Building/editing the 1N4007 diode model present in the bridge rectifier circuit using model builder tool.
- This is explained using a bridge rectifier circuit which contains 1N4007 diode.

4. Subcircuit creation using Oscad

We show how to create and edit a subcircuit. This is explained using astable multivibrator circuit that has 555 timer IC as a subcircuit. The tutorial covers the following:

- An already created astable multivibrator schematic is opened to show the component 555 timer in it.
- As 555 timer will be modelled as a subcircuit, the subcircuit schematic of 555 timer is shown next.
- The tutorial then shows how to edit the 555 timer subcircuit schematic.

10.3 Instruction Sheet

This section should be used as a set of instructions to practice tutorials assuming you have the Oscad spoken tutorials CD/DVD with you.

10.3.1 The procedure to practice

- You have been given a set of spoken tutorials and files.
- You will typically do one tutorial at a time.
- You may listen to a spoken tutorial and reproduce all the commands shown in the video.
- If you find it difficult to do the above, you may consider listening to the whole tutorial once and then practice during the second hearing.

10.3.2 Please ensure

- You have Linux Ubuntu 12.04 OS or above installed on your computer.
- You have a working internet connection on your computer.
- You have basic knowledge about Linux to follow these tutorials.

10.3.3 Basic Module

- Right-click on the file named `index.html`, and choose `Open with Firefox` to open this file in the Firefox web browser.
- Read the instructions given.
- In the left hand side panel you will see `Basic Level`.
- Please click on the module `Basic Level`.
- In this module, there are a few tutorials.

- **Introduction to Oscad** teaches how to install Oscad and test run Oscad using an example.
- Click on it. You will see the video in the centre.
- Click on the play button on the player to play the tutorial.
- To view the tutorial in a bigger player, right-click on the video and choose **View Video** option.
- Adjust the size of the player in such a way that you are able to practice in parallel.
- Follow the tutorial and reproduce all the activities as shown in the tutorial.
- Now you will have Oscad installed and working on your computer.

10.3.4 Schematic creation and Simulation

- Locate the next topic **Schematic creation and Simulation**.
- Click on it. Follow the tutorial and reproduce all the activities as shown in the tutorial.
- Please save your project files that you will create while you practice this tutorial.
- Guidelines for saving your work are as follows-

Instructions for Practice

- Create a folder on the Desktop with your Name-RollNo-Component (Eg. vin-04-Oscad).
- Give a unique name to the files you save, so as to recognize it next time. (Eg. Practice-1-Oscad).
- Remember to save all your work in your folder.
- This will ensure that your files don't get over-written by someone else.
- Remember to save your work from time to time instead of saving it at the end of the task.

Instructions for Assignments

- Attempt all the given assignments.
- Save your work by creating a folder called **Oscad-Assignment** in your main folder.

- At 09:37 the video says that you have to watch KiCad tutorial - **Designing Circuit schematic in KiCad**.
- Locate this tutorial on the left hand panel and watch it.
- Reproduce the astable multivibrator circuit schematic shown in it using Oscad.
- After you finish this tutorial, locate the next tutorial **Designing Printed Circuit Board**.

10.3.5 Designing Printed Circuit Board

- Click on the next topic **Designing Printed Circuit Board**.
- You will need to use the practice files created in the previous tutorial.
- Follow this tutorial and reproduce all the activities as shown.
- At 08:50 the video says that you have to watch KiCad tutorials -
 1. Electric rule checking and netlist generation.
 2. Mapping components in KiCad.
 3. Designing printed circuit board in KiCad.
- Locate these tutorials on the left hand panel and watch it.
- Reproduce the layout of astable multivibrator shown in it using Oscad.

Chapter 11

OSCAD on Aakash

Aakash provides great platform for learning and education. With GNU/Linux now running on Aakash, most of our student developers will find it useful. The best part is, almost all GNU/Linux applications runs on Aakash. As tools such as KiCad, Ngspice and Scilab are already running on Aakash, OSCAD installation procedure is similar to any other desktop running GNU/Linux. With OSCAD running on Aakash, it proves Aakash's capability to run Electronic design tools. It also shows OSCAD's portability from desktops to hand held devices.

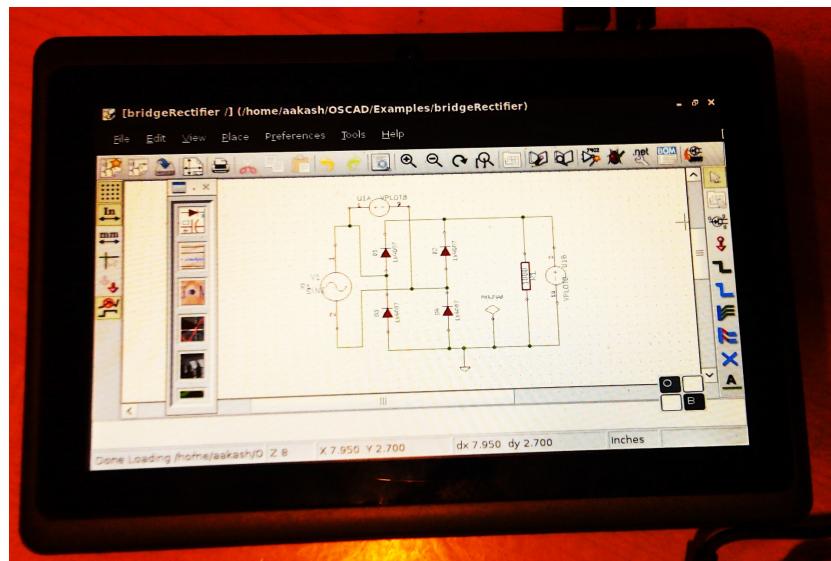


Figure 11.1: Bridge rectifier schematic on Aakash

11.1 Installation

Installation of OSCAD is similar to its GNU/Linux desktop, please follow the same instructions mentioned in Chapter 2. OSCAD installer runs **Synaptic package manager** underneath. This package manager is smart enough to detect your CPU architecture. It downloads and install specific packages to your machine architecture. We are further enhancing the installer so that it could download required version of Scilab for specific machine architecture from the internet with minimum user intervention.

One has to ensure that his/her Aakash tablet is connected to the internet through *WiFi* before starting the installation procedure. OSCAD is currently in development stage on Aakash, but it is completely usable. Once it is ready for Aakash, it will come preinstalled on Aakash.

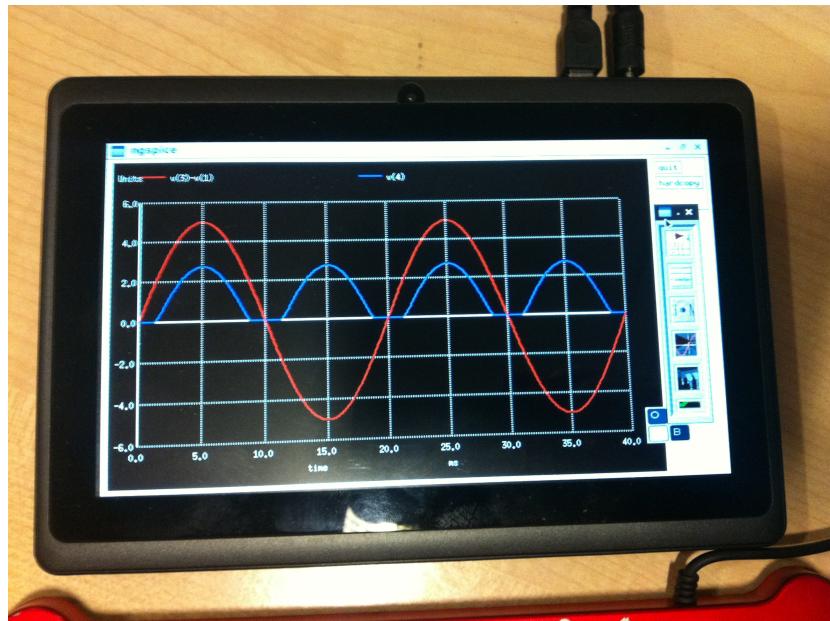


Figure 11.2: Ngspice running on Aakash

11.2 Porting on Aakash

The design framework of OSCAD made it extremely simple to run it on Aakash with minimum modification. As OSCAD was initially developed on an Ubuntu distribution which has GNOME-terminal primarily. This GNOME-terminal was called whenever there was a need for user input using terminal.

On the other hand, GNU/Linux side of Aakash also uses same Ubuntu desktop version but not GNOME as it Desktop Environment. To be ease on memory

and CPU it uses LXDE desktop environment. LXDE is intended for systems with low memory and CPU. LXDE on the other side has LX-terminal which made it difficult for OSCAD process to invoke terminal on demand. Keeping in mind the portability issue, we have dropped both GNOME and LX terminal. Instead we have now opted for `xterm` which is the standard *terminal emulator* for X-Window System.

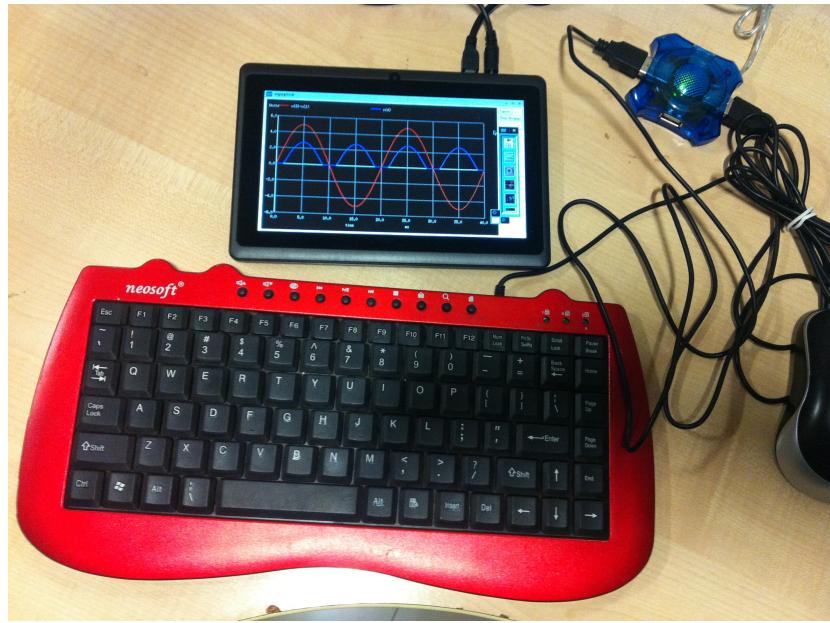


Figure 11.3: OSCAD running with external keyboard and mouse attached

11.3 Usage

Advantage of using OSCAD on Aakash is its portability and performance. Beside having touch on Aakash, one may attach external keyboard and mouse. One should find the similar interface of OSCAD on Aakash as its Desktop version. Sometimes keyboard shortcuts are more handy than using touch controls when running designing and simulation tool.

Appendix A

Solved Examples

Some useful solved examples from the book “Microelectronics Circuits - Theory and Applications”, Fifth Edition by Adel S.Sedra and Kenneth C. Smith, Adapted by Arun N Chandorkar (Oxford University Press) are presented in the Appendix.

A.1 Diode

1. **Example 2.1:** Figure A.1 shows a circuit for charging a 12-V battery. If v_s is a sinusoid with 24-V peak amplitude, find the fraction of each cycle during which diode conducts. Also, find the peak value of the diode current and the maximum reverse-bias voltage that appears across diode.

Solution: Draw the schematic shown in Figure A.1 using schematic editor. Annotate the schematic using the *Annotate* tool from the top toolbar in Schematic editor. Perform Electric Rules check using the *Perfrom electric rules check* tool from the top toolbar. Ensure that there are no errors in the circuit schematic. Now generate Spice netlist for simulation using the *Generate Netlist* tool from the top toolbar. This is shown in Figure A.2.

The next step is to invoke Analysis inserter from the Oscad tool bar. Click **analysis inserter** and select the option **transient** as given in Figure A.3. Enter **start time = 0**, **step time = 1 ms**, **stop time = 10 ms** as in Figure A.4.

Click on **Add Simulation Data** and save the analysis file as in figure A.5.

Now click on **Netlist Converter** in Oscad tool bar and enter the values of DC source and SINE source as shown in following figure A.6 and then press ‘enter’ key. This will generate the Ngspice netlist (.cir.out).

Now click on **Ngspice** from the Oscad tools bar. This will open up the Ngspice terminal with waveform window as shown in figure A.7.

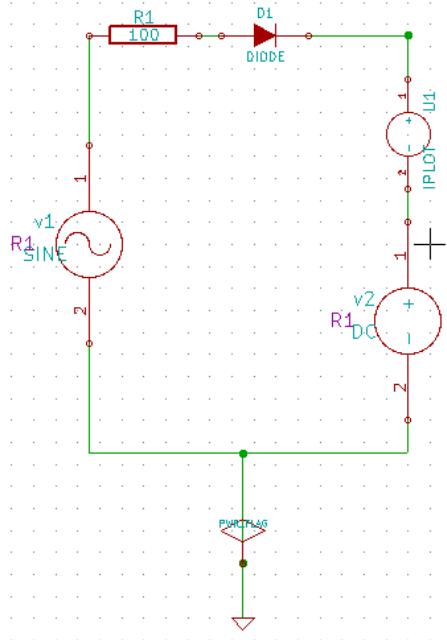


Figure A.1: Schematic of example 2.1

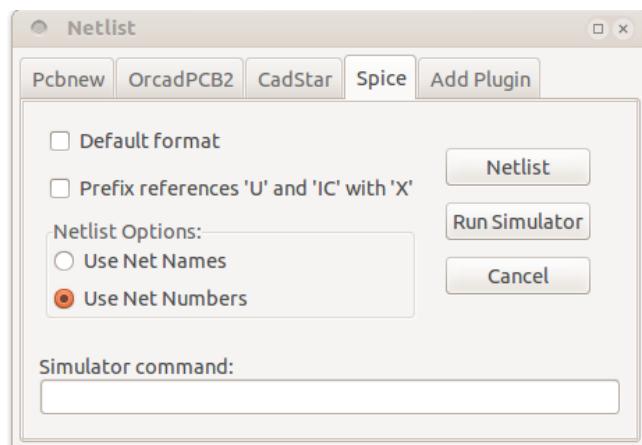


Figure A.2: Netlist generation

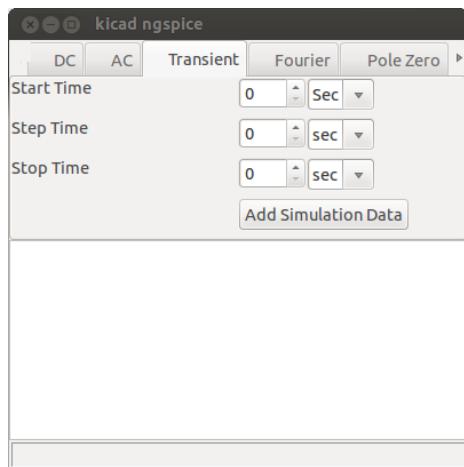


Figure A.3: Analysis Inserter

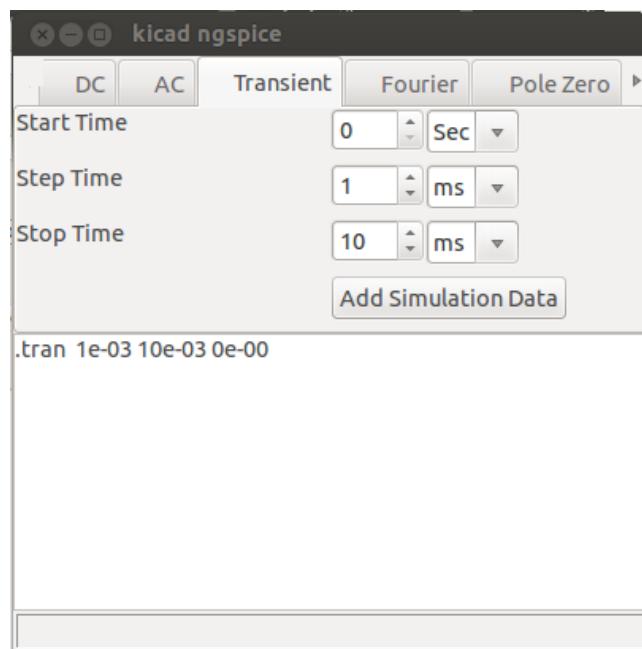


Figure A.4: Analysis Inserter: Transient simulation options

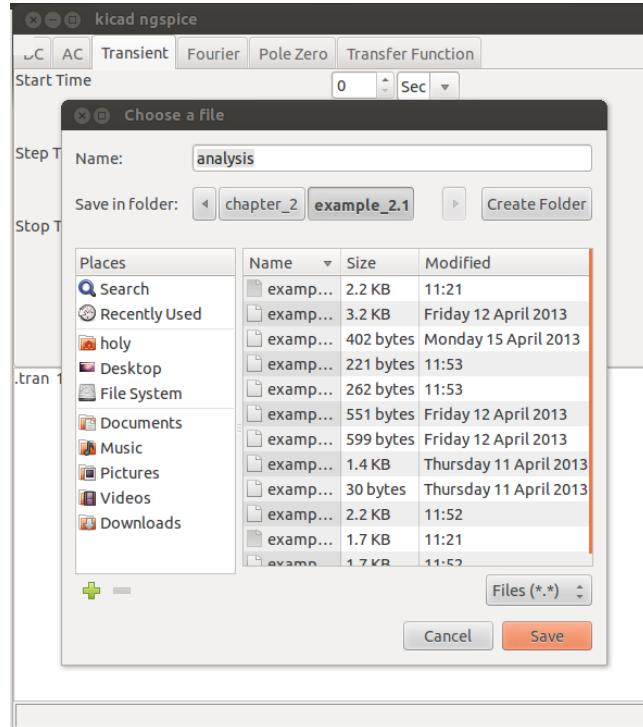


Figure A.5: Analysis Inserter: Save analysis file

2. **Example 2.5:** Find the voltage V_D and current I_D in the circuit shown in Figure A.8, where $V_{DO} = 0.65$, $r_D = 20 \Omega$.

Solution: Create schematic and generate netlist in the same way as given in Example 2.1.

Click on “analysis inserter” from Oscad tool bar. Select “DC” and then enter the following details: `enter source name = v1, start = 0, increment = 1V` and `stop = 5V` as given in Figure A.9. Click on “Add Simulation Data” and save the analysis file.

Now click on **Netlist Converter** and enter the value of DC source as shown in Figure A.10 and then press ‘enter’ key. Now click on “Ngspice” from the Oscad tool bar. The results of Ngspice simulation is shown in Figure A.11. This shows the current and voltage waveforms and Ngspice terminal.

```

Terminal
=====
Kicad to Ngspice netlist converter
=====
converting example_2.1.cir
.tran 10e-03 1e-01 0e-00
-----
Add parameters for DC source v2
Enter value(Volts/Amps): 12
-----
Add parameters for sine source v1
Enter offset value (Volts/Amps): 0
Enter amplitude (Volts/Amps): 24
Enter frequency (Hz): 50
Enter delay time (seconds): 0
Enter damping factor (1/seconds): 0
-----
The ngspice netlist has been written in example_2.1.cir.out
The scilab netlist has been written in example_2.1.cir.ckt
Press Enter to quit

```

Figure A.6: Kicad to Ngspice

```

Terminal
=====
Error on line 5 : d1 5 3 diode
Unable to find definition of model diode - default assumed
Doing analysis at TEMP = 27.000000 and TNOM = 27.000000

Warning: v1: no DC value, transient time 0 value used

Initial Transient Solution
-----
Node          Voltage
-----
1            12
3            12
5            1.201e-09
2            0
v1#branch    1.201e-11
v_u1#branch   -1.201e-11
v2#branch    -1.201e-11

No. of Data Rows : 59
ngspice 1 ->

```

Figure A.7: Ngspice Waveform Window

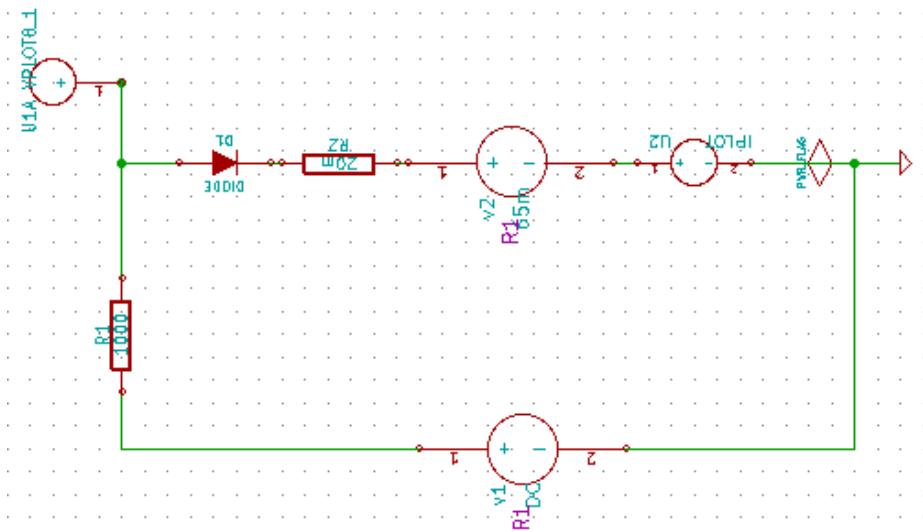


Figure A.8: Schematic

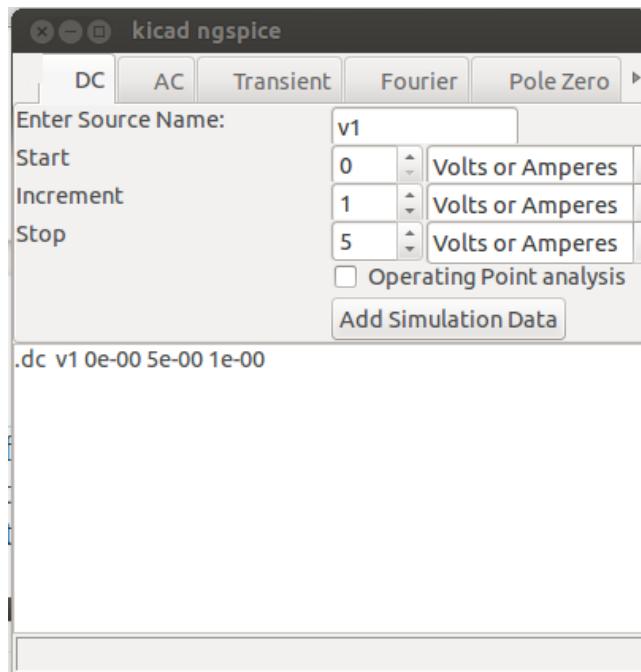
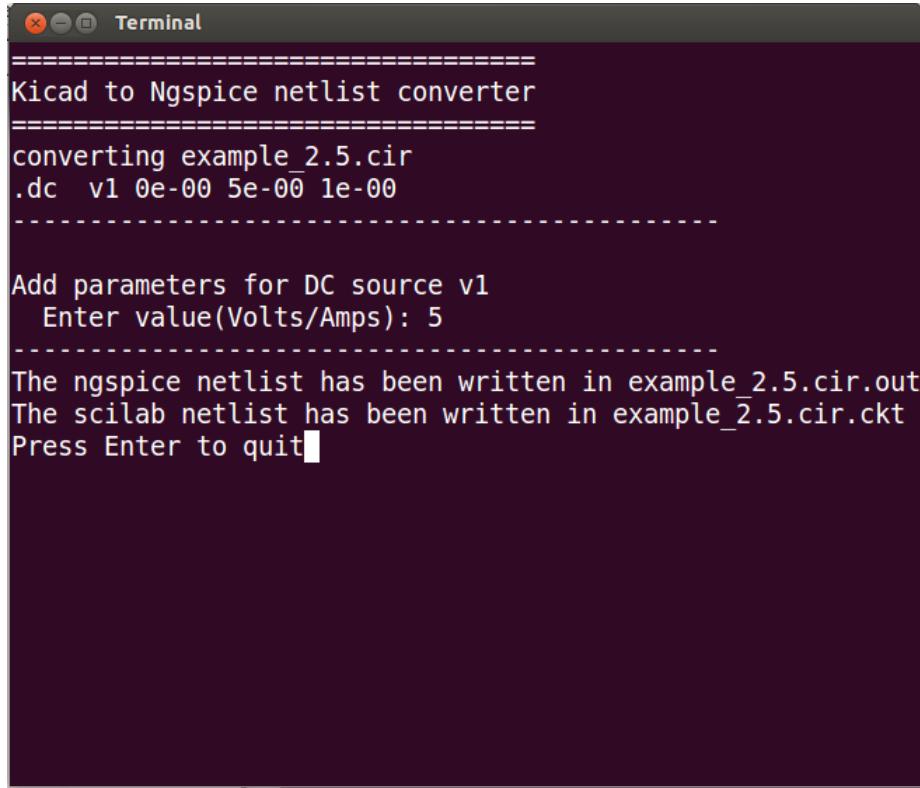


Figure A.9: Analysis Inserter



```
Terminal
=====
Kicad to Ngspice netlist converter
=====
converting example_2.5.cir
.dc v1 0e-00 5e-00 1e-00
-----
Add parameters for DC source v1
Enter value(Volts/Amps): 5
-----
The ngspice netlist has been written in example_2.5.cir.out
The scilab netlist has been written in example_2.5.cir.ckt
Press Enter to quit
```

Figure A.10: Kicad to Ngspice

A.2 BJT

Example 3.1: Find the voltage at all nodes in the transistors shown in Figure A.12. We will assume that β is specified to be 100.

Solution: Create schematic and generate netlist in the same way as given in Example 2.1.

Click on “analysis inserter” from Oscad tool bar. select “DC” and then enter the following details: `enter source name = v1, start = 0, increment = 1V` and `stop = 15V` as given in Figure A.13. Click on “Add Simulation Data” and save the analysis file. Now click on **Netlist Converter** and enter the value of DC source as shown in Figure A.14 and then press ‘enter’ key. Now click on “Ngspice” from the Oscad tool bar. The results of Ngspice simulation is shown in Figure A.15.

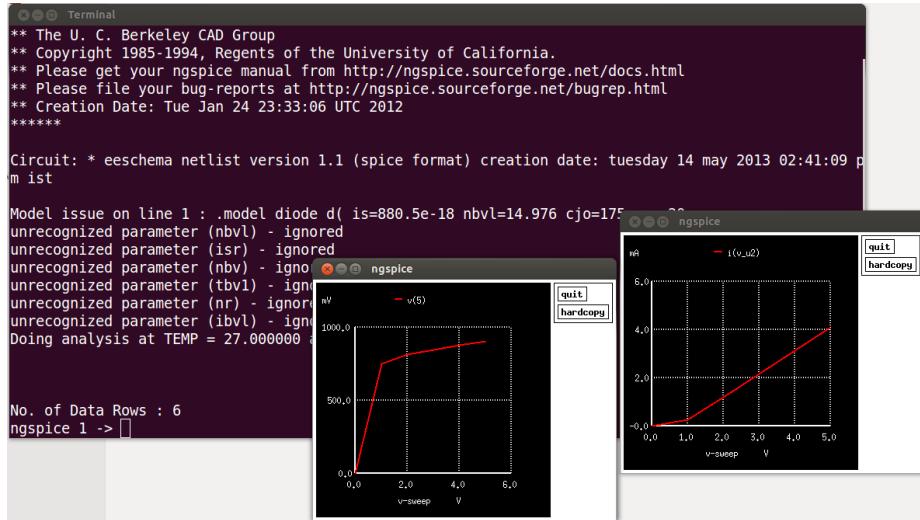


Figure A.11: Ngspice Waveform window

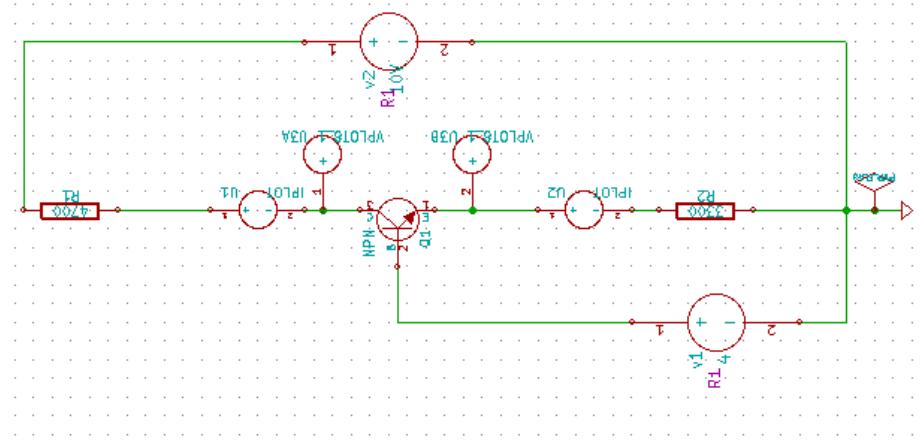


Figure A.12: Schematic

A.3 MOSFET

Example 4.5: Analyze the circuit shown in fig A.16 to determine the voltages at all nodes and the current through all branches.

Solution: Create schematic and generate netlist in the same way as given in Example 2.1.

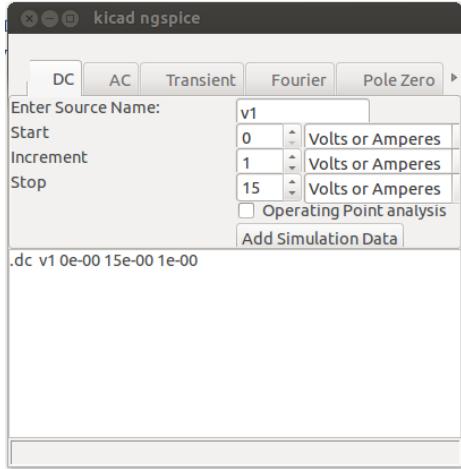


Figure A.13: Analysis Inserter

```
=====
Kicad to Ngspice netlist converter
=====
converting example3.4.cir
.dc v1 0e-00 4e-00 5e-03
The ngspice netlist has been written in example3.4.cir.out
The scilab netlist has been written in example3.4.cir.ckt
Press Enter to quit
```

Figure A.14: Kicad to Ngspice

Click on “analysis inserter” from Oscad tool bar. select “DC” and then enter the following details: `enter source name = v1, start = 0, increment = 1V` and `stop = 10V` as given in Figure A.17. Click on “Add Simulation Data” and save the analysis file. Click on “Add Simulation Data” and save the analysis file.

Now click on `Netlist Converter` and enter the value of DC source as shown in Figure A.18. and then press ‘enter’ key. Now click on “Ngspice” from the Oscad tool bar. The results of Ngspice simulation is shown in Figure A.19.

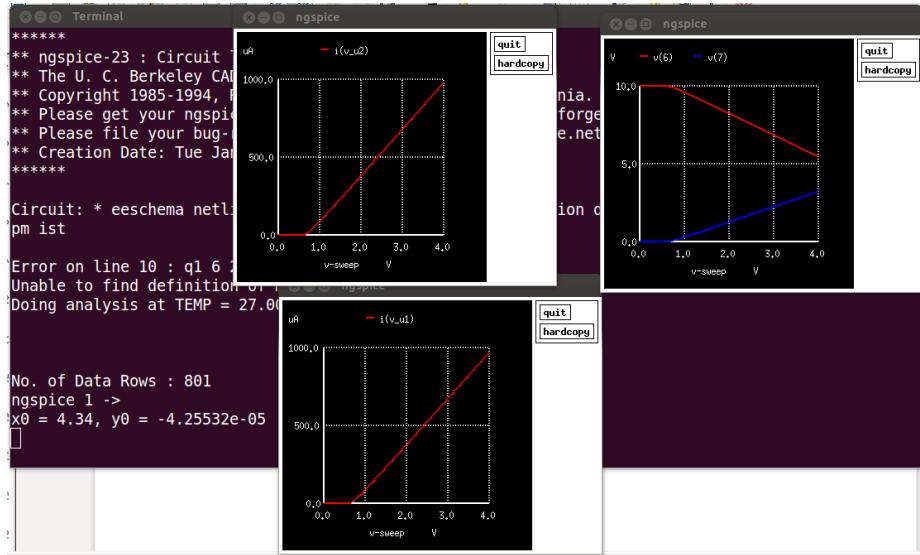


Figure A.15: Ngspice Waveform

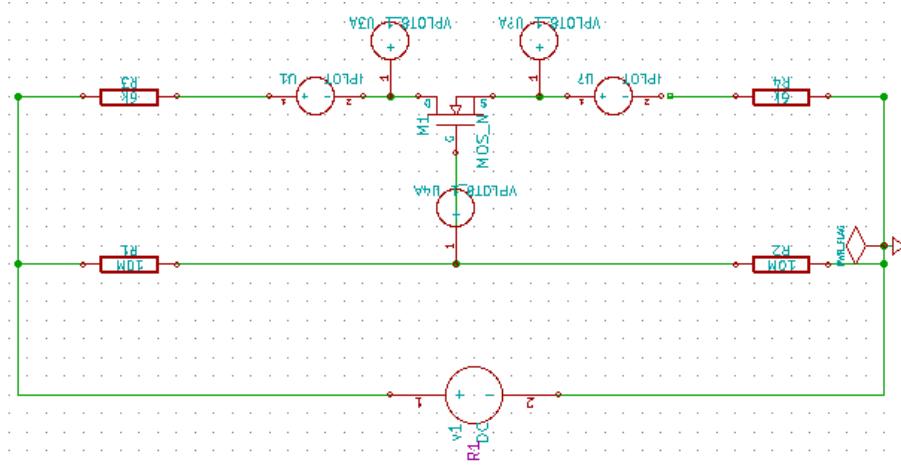


Figure A.16: Schematic

A.4 OP-AMP

1. **Example 5.3:** Figure A.20 shows an op-amp circuit. Find the output voltage where $R_1 = 10 \Omega$, $R_f = 1k\Omega$, $R_L = 100k\Omega$ and input voltage $v_I = 10V$.

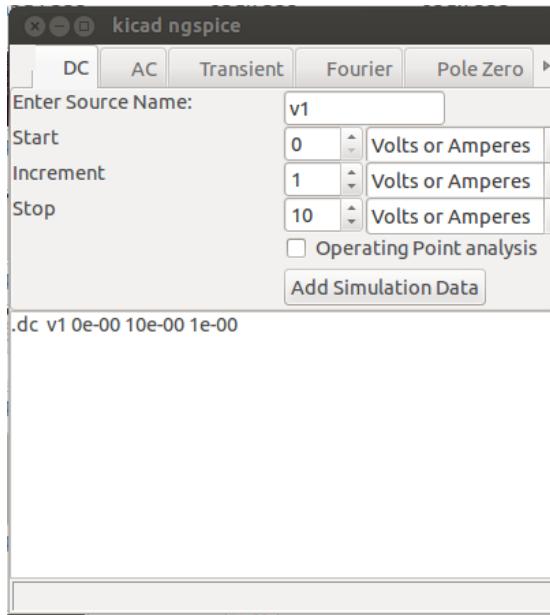


Figure A.17: Analysis Inserter

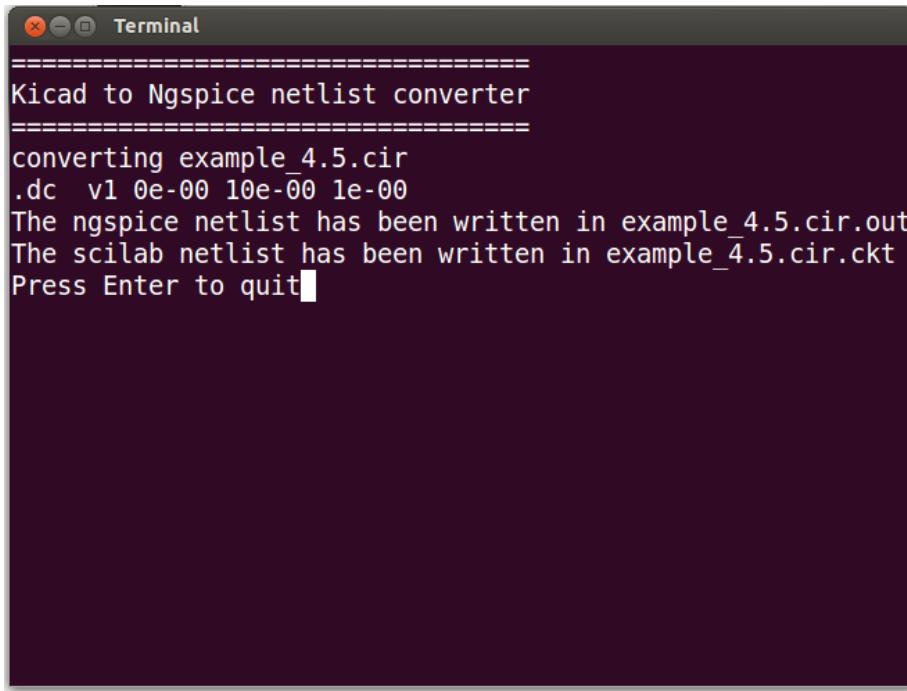
Solution: Create schematic and generate netlist in the same way as given in Example 2.1. While generating netlist, DO NOT uncheck the option **Prefix references ‘U’ and ‘IC’ with ‘X’**.

Click on “analysis inserter” from Oscad tool bar. select “DC” and then enter the following details: **enter source name = v1, start = 0, increment = 1V and stop = 10V** as given in Figure A.17. Click on “Add Simulation Data” and save the analysis file.

Click on “Subcircuit builder” from the Oscad toolbar and click on *Cancel* in the subcircuit selector window. Now, *import* the existing subcircuits. Figure A.21 will appear. Choose **ua741** and click on *ok*. Figure A.22 will now appear.

Click on *ok* in the **Successfully imported** window. Let us open the subcircuit schematic of **ua741**. This can be done as follows: Click on the *File* menu. Choose *open* and then type **ua741** in the *Enter Component name* field as shown in Figure A.23. Click on *ok*. Add the Oscad libraries to the subcircuit schematic to prevent the Load Error. The schematic of the subcircuit is as shown in Figure A.24.

Now click on **Netlist Converter** and enter the value of DC source as shown in Figure A.25 and then press ‘enter’ key. Now click on Ngspice



```

Terminal
=====
Kicad to Ngspice netlist converter
=====
converting example_4.5.cir
.dc v1 0e-00 10e-00 1e-00
The ngspice netlist has been written in example_4.5.cir.out
The scilab netlist has been written in example_4.5.cir.ckt
Press Enter to quit

```

Figure A.18: Kicad to Ngspice

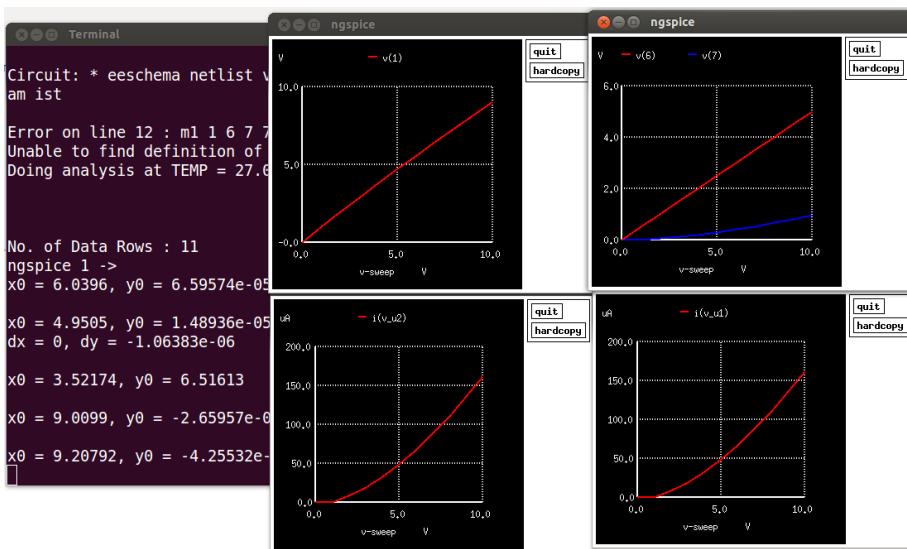


Figure A.19: Ngspice Waveform

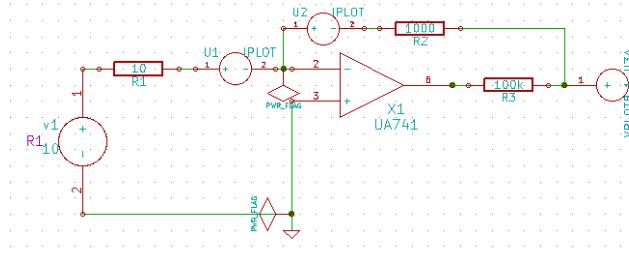


Figure A.20: Schematic

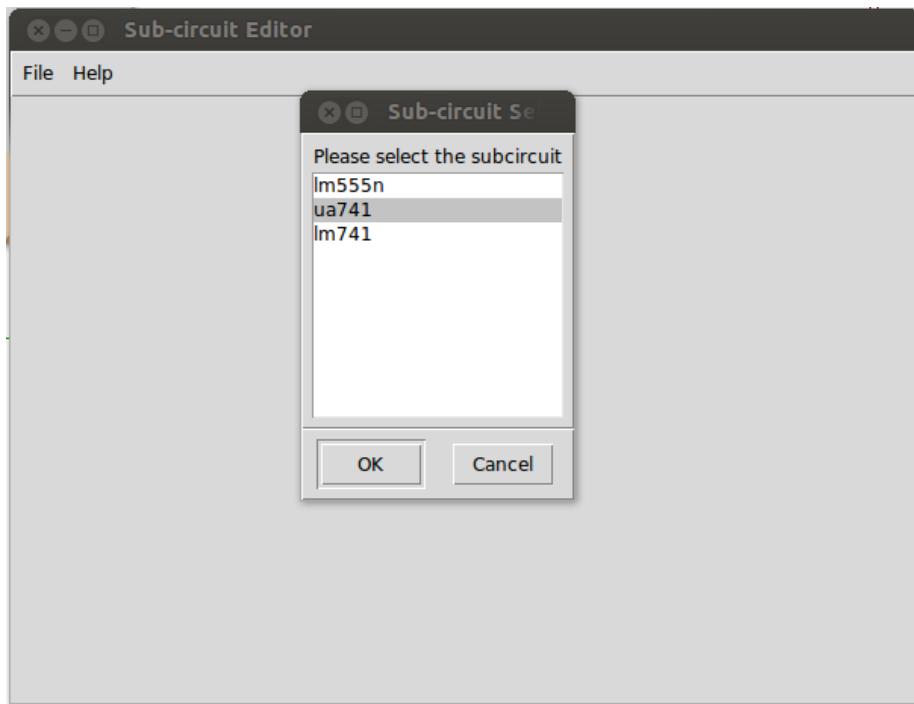


Figure A.21: subcircuit builder 1

button in Toolbar. The figure A.26 will appear.

2. **Example 5.6:** Consider the non inverting amplifier circuit shown in Figure A.27. It is fed with a lower frequency sinusoidal signal of peak voltage $V_i = 1V$, and is connected to load resistance $R_L = 1k\Omega$.

Solution: Create schematic and generate netlist in the same way as given

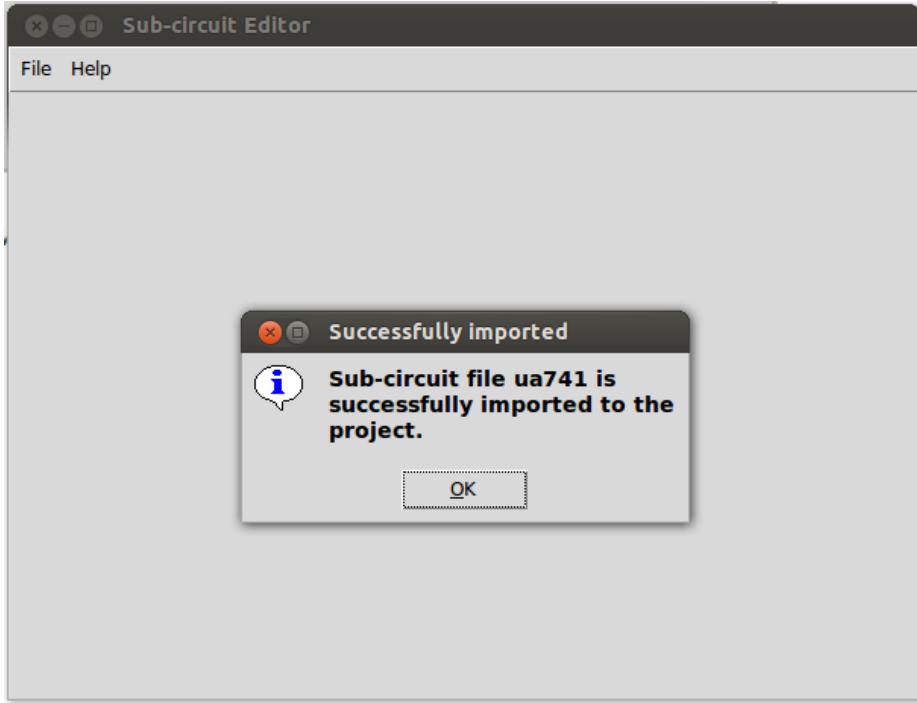


Figure A.22: subcircuit builder 2

in Example 2.1.

Click on “analysis inserter” from Oscad tool bar. select “Transient” and then enter the following details: **Start time** = 0 sec, **Step time** = 1 ms, **stop time** = 10 ms as given in Figure A.28. Click on “Add Simulation Data” and save the analysis file. For subcircuit builder, follow the steps given in Example 5.3.

Now click on **Netlist Converter** and enter the value of sine source as shown in Figure A.29 and then press ‘enter’ key. Now click on Ngspice button in Toolbar. The figure A.30 will appear.

3. **Example 5.7:** For the circuit given in the figure A.31, perform AC analysis where $R_1 = 1\text{k}\Omega$, $R_f = 100\text{k}\Omega$, $C_f = 1.59\text{nF}$ and input voltage $V_i = 1\text{V A.C.}$

Solution: Create schematic and generate netlist in the same way as given in Example 2.1. Click on “analysis inserter” from Oscad tool bar. select “AC” and then enter the following details: **scale** = Lin, **start frequency** = 1, **stop frequency** = 10 Meg, **No. of points** = 10 as given in Figure A.32. Click on “Add Simulation Data” and save the analysis file. For subcircuit builder, follow the steps given in Example 5.3. Now

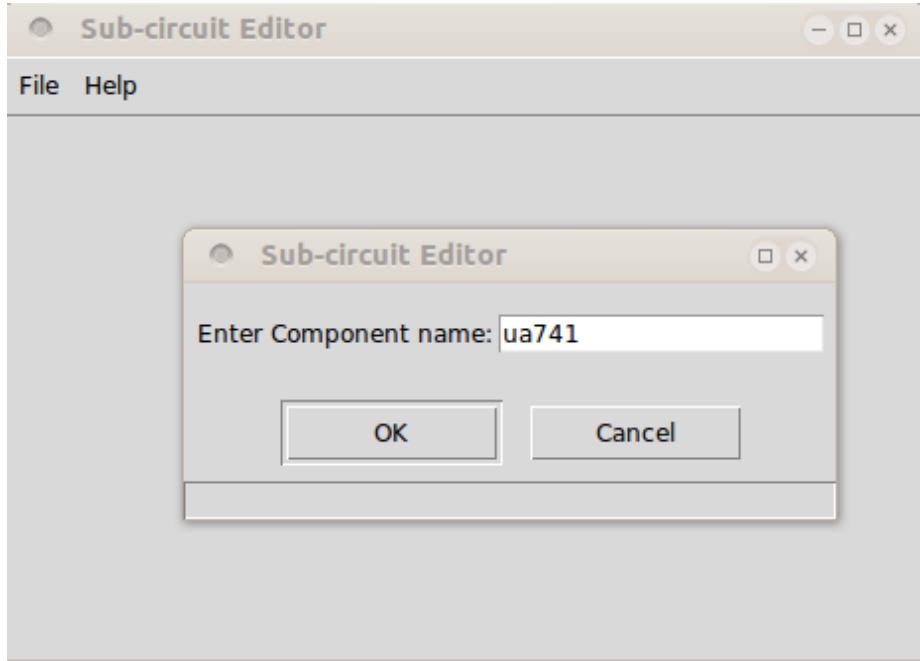


Figure A.23: Open component subcircuit

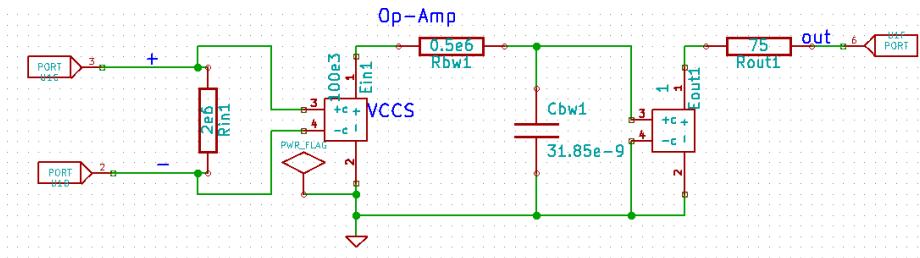
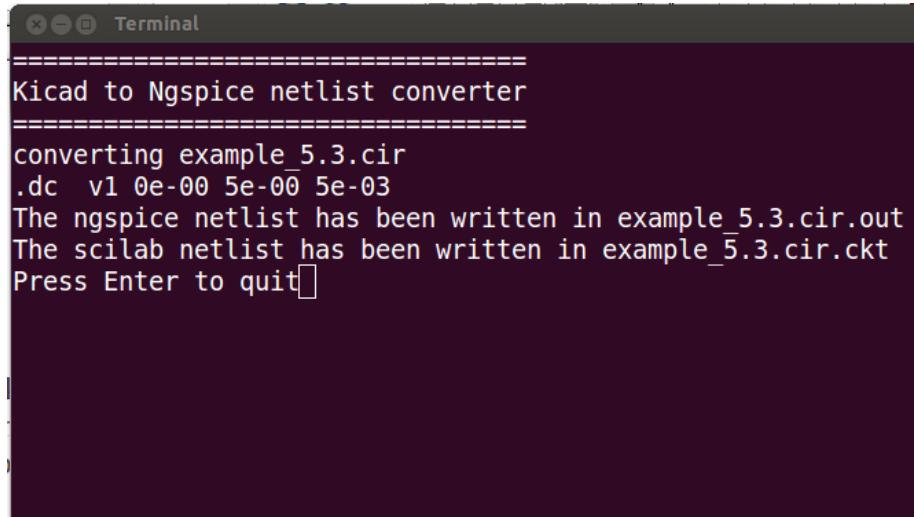


Figure A.24: subcircuit builder 3

click on **Netlist Converter** and enter the value of AC source as shown in Figure A.33 and then press 'enter' key. Now click on **Ngspice** button in Toolbar. The figure A.34 will appear.

A.5 CMOS

Example 9.4: Operation of CMOS Inverter



```

=====
Kicad to Ngspice netlist converter
=====
converting example_5.3.cir
.dc v1 0e-00 5e-00 5e-03
The ngspice netlist has been written in example_5.3.cir.out
The scilab netlist has been written in example_5.3.cir.ckt
Press Enter to quit

```

Figure A.25: Kicad to Ngspice

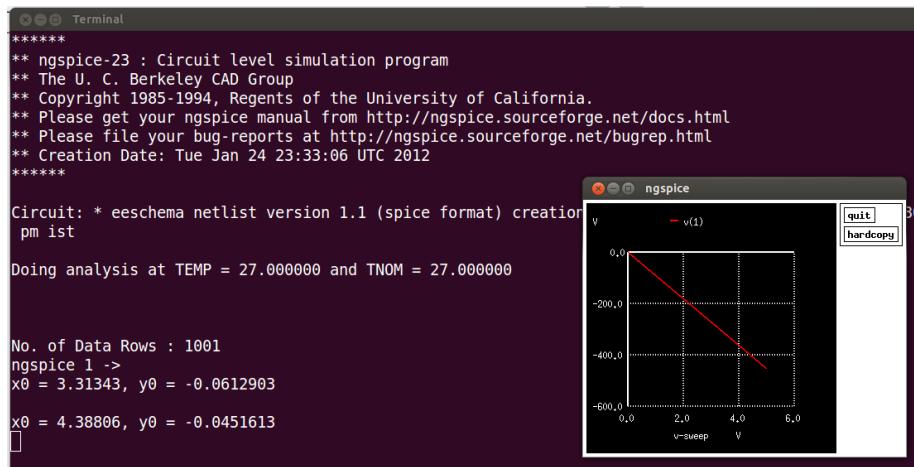


Figure A.26: Ngspice Waveform

Solution: Create schematic and generate netlist in the same way as given in Example 2.1. Click on “analysis inserter” from Oscad tool bar. select “DC” and then enter the following details: `enter source name = v1, start = 0, increment = 1 V, stop = 15V` as given in Figure A.36. Click on “Add Simulation Data” and save the analysis file.

Now click on **Netlist Converter** and enter the value of DC source as shown

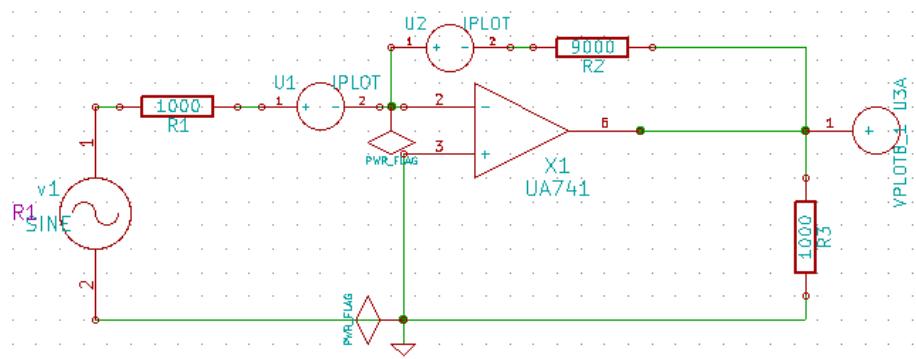


Figure A.27: Schematic

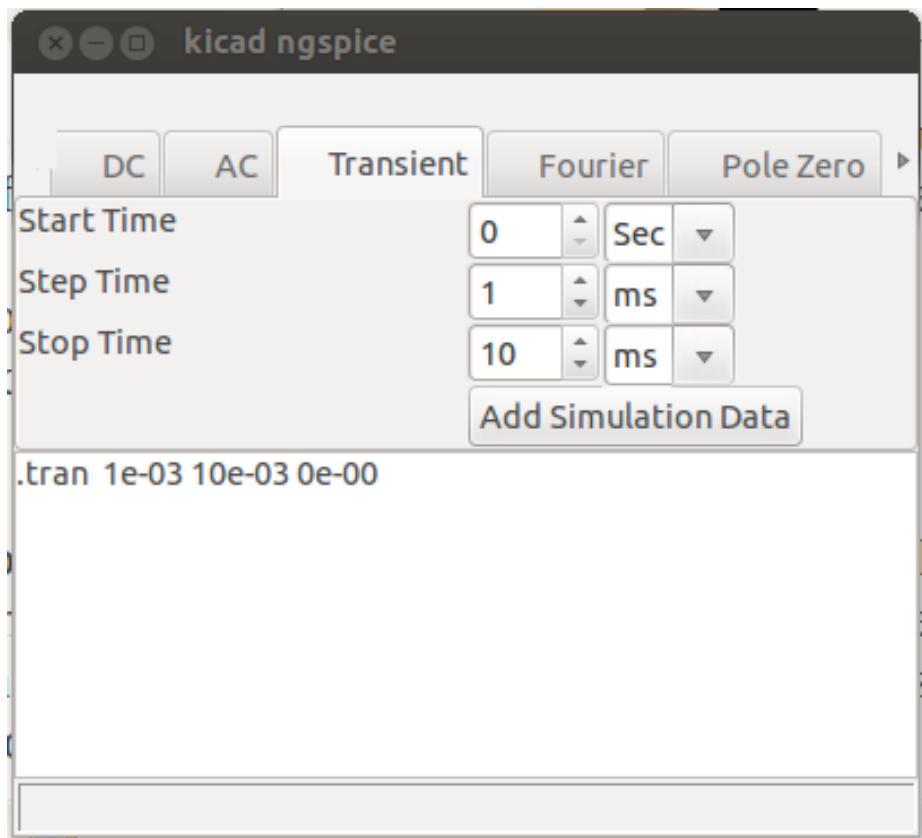
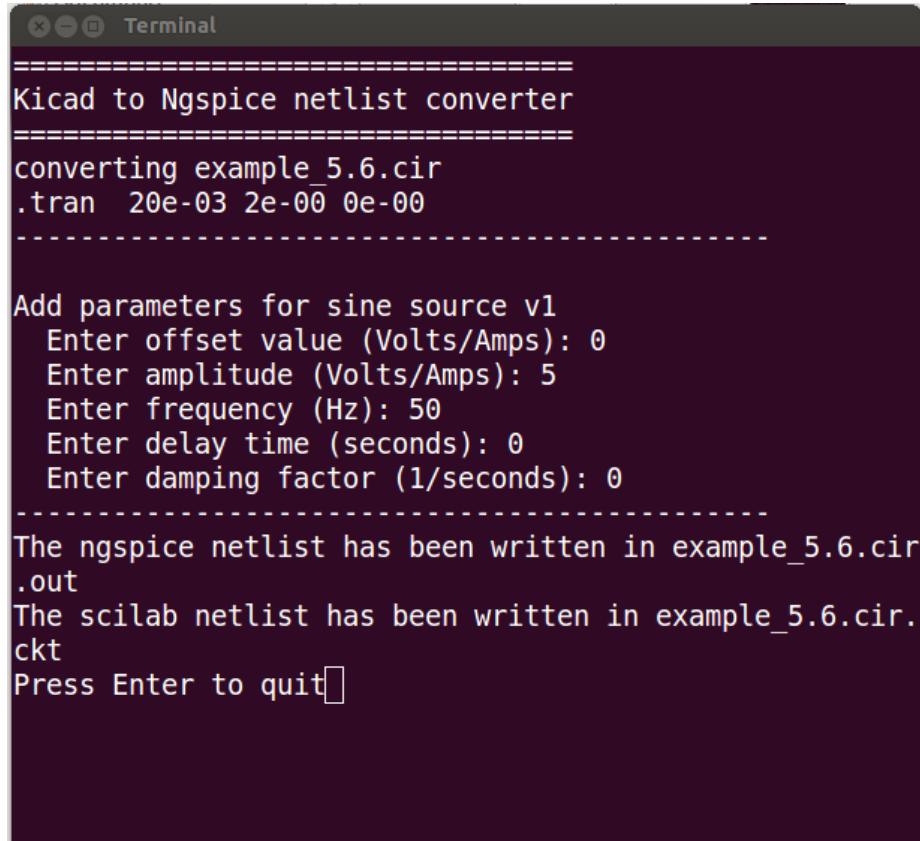


Figure A.28: Analysis Inserter



```
=====
Kicad to Ngspice netlist converter
=====
converting example_5.6.cir
.tran 20e-03 2e-00 0e-00
-----
Add parameters for sine source v1
Enter offset value (Volts/Amps): 0
Enter amplitude (Volts/Amps): 5
Enter frequency (Hz): 50
Enter delay time (seconds): 0
Enter damping factor (1/seconds): 0
-----
The ngspice netlist has been written in example_5.6.cir
.out
The scilab netlist has been written in example_5.6.cir.
ckt
Press Enter to quit
```

Figure A.29: Kicad to Ngspice

in Figure A.37 and then press 'enter' key. Now click on Ngspice button in Toolbar. The figure A.38 will appear.

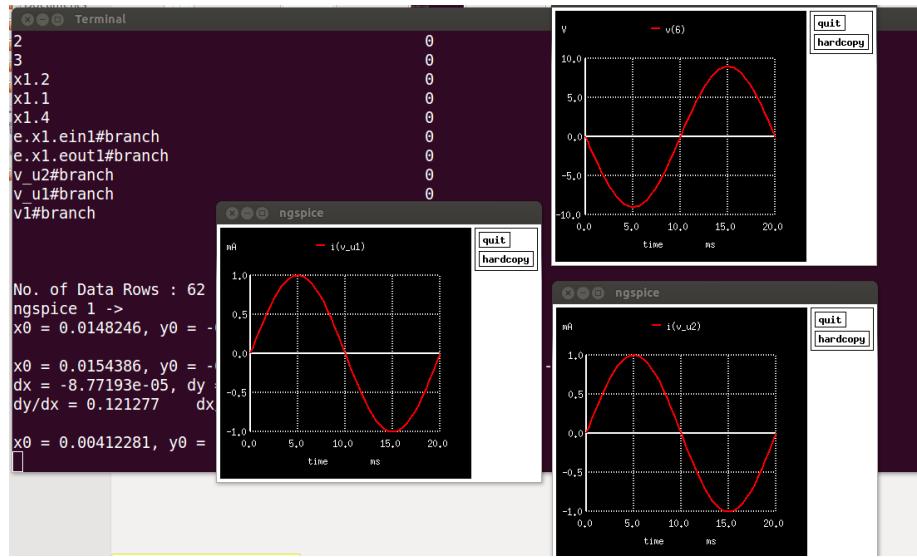


Figure A.30: Ngspice Waveform

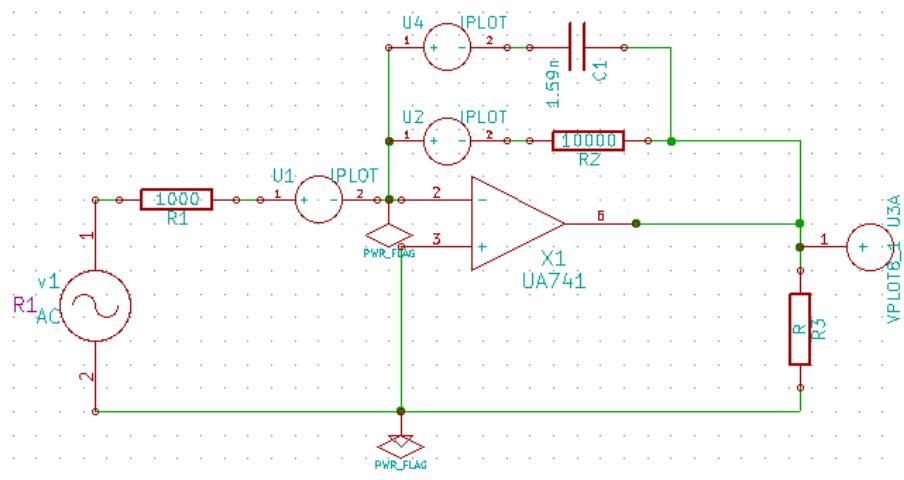


Figure A.31: Schematic

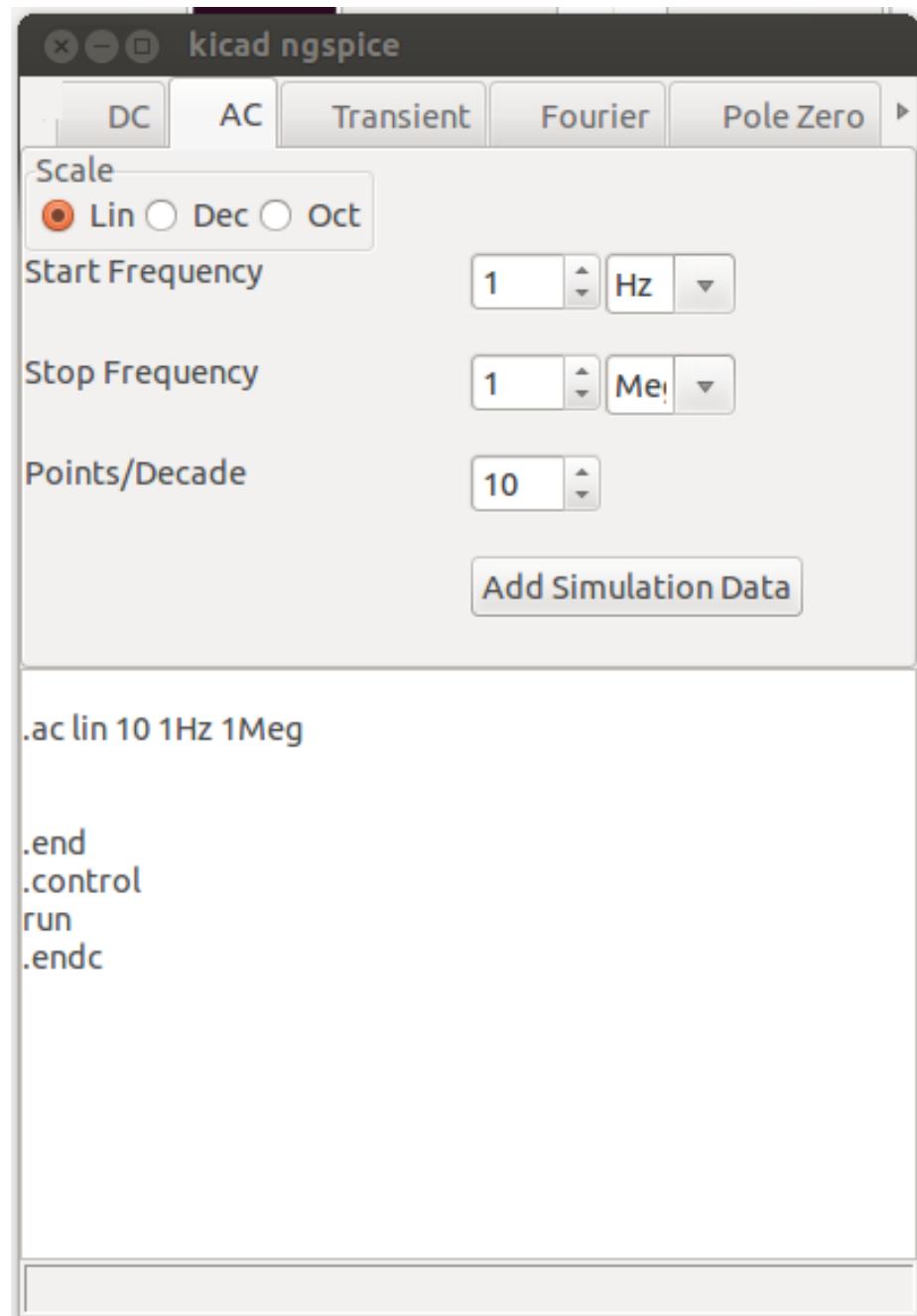
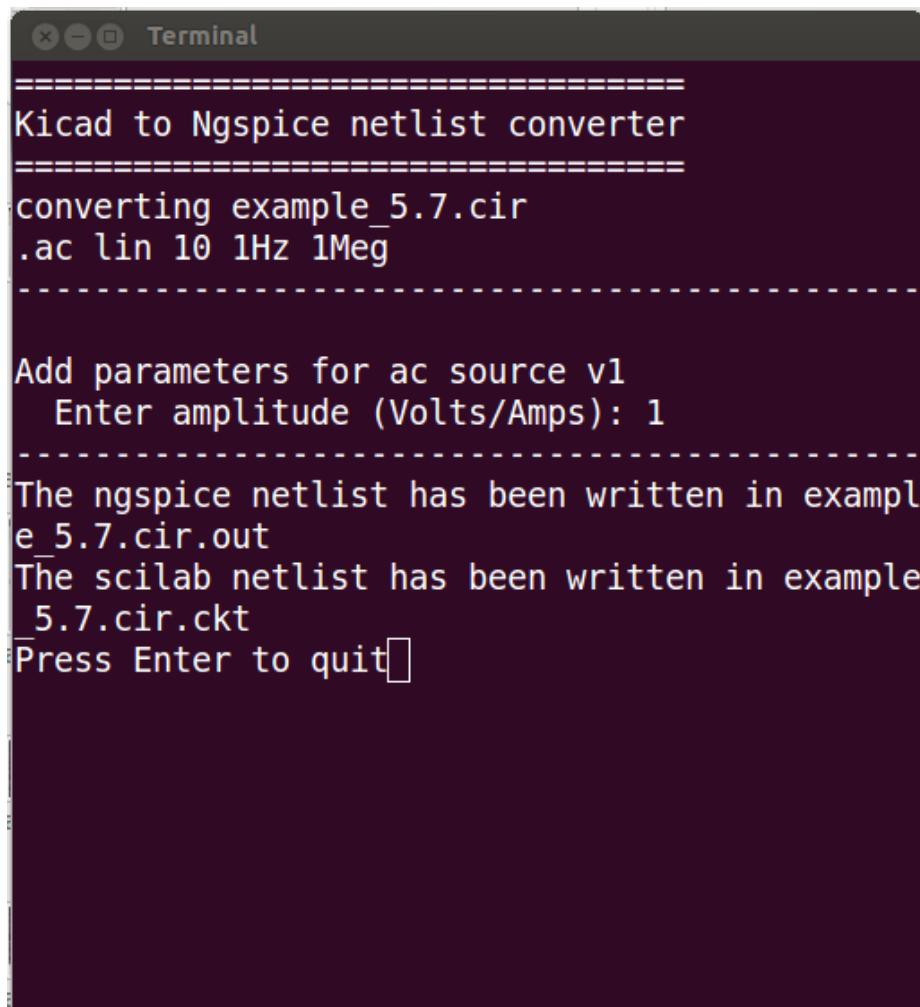


Figure A.32: Analysis Inserter



```
Terminal
=====
Kicad to Ngspice netlist converter
=====
converting example_5.7.cir
.ac lin 10 1Hz 1Meg
-----
Add parameters for ac source v1
Enter amplitude (Volts/Amps): 1
-----
The ngspice netlist has been written in example_5.7.cir.out
The scilab netlist has been written in example_5.7.cir.ckt
Press Enter to quit
```

Figure A.33: Kicad to Ngspice

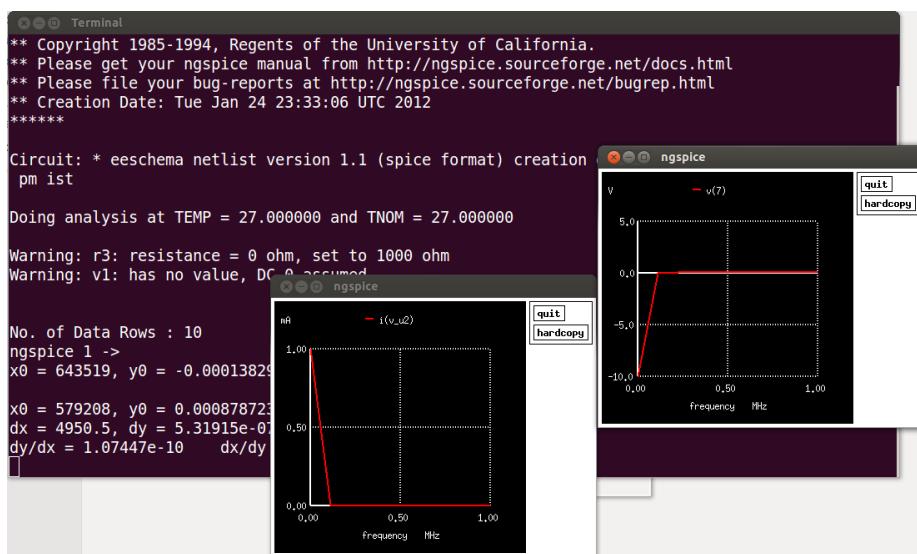


Figure A.34: Ngspice Waveform

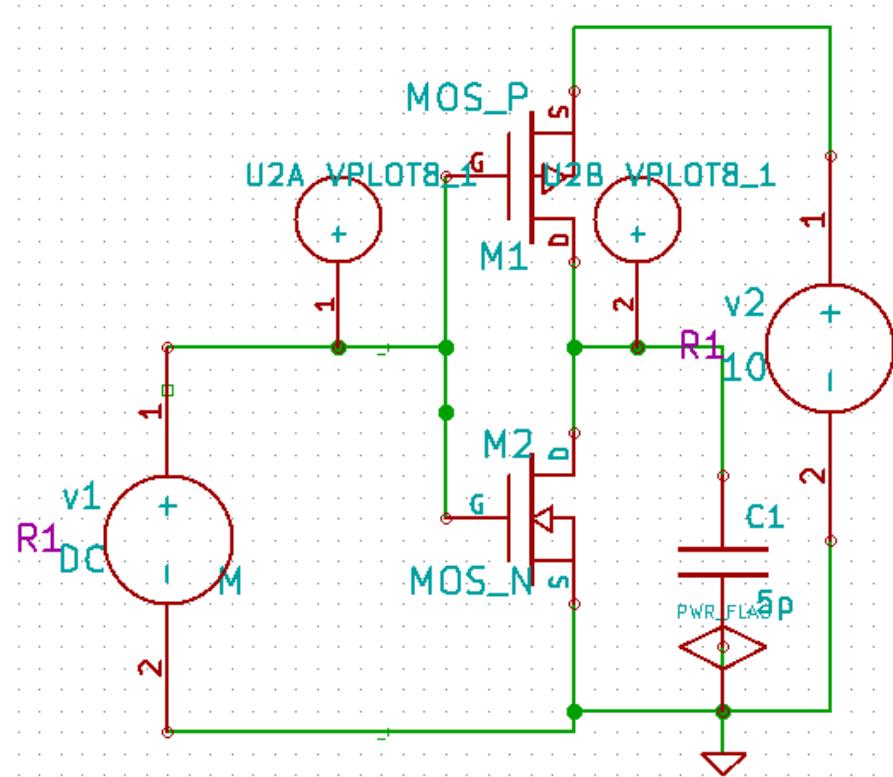


Figure A.35: Schematic

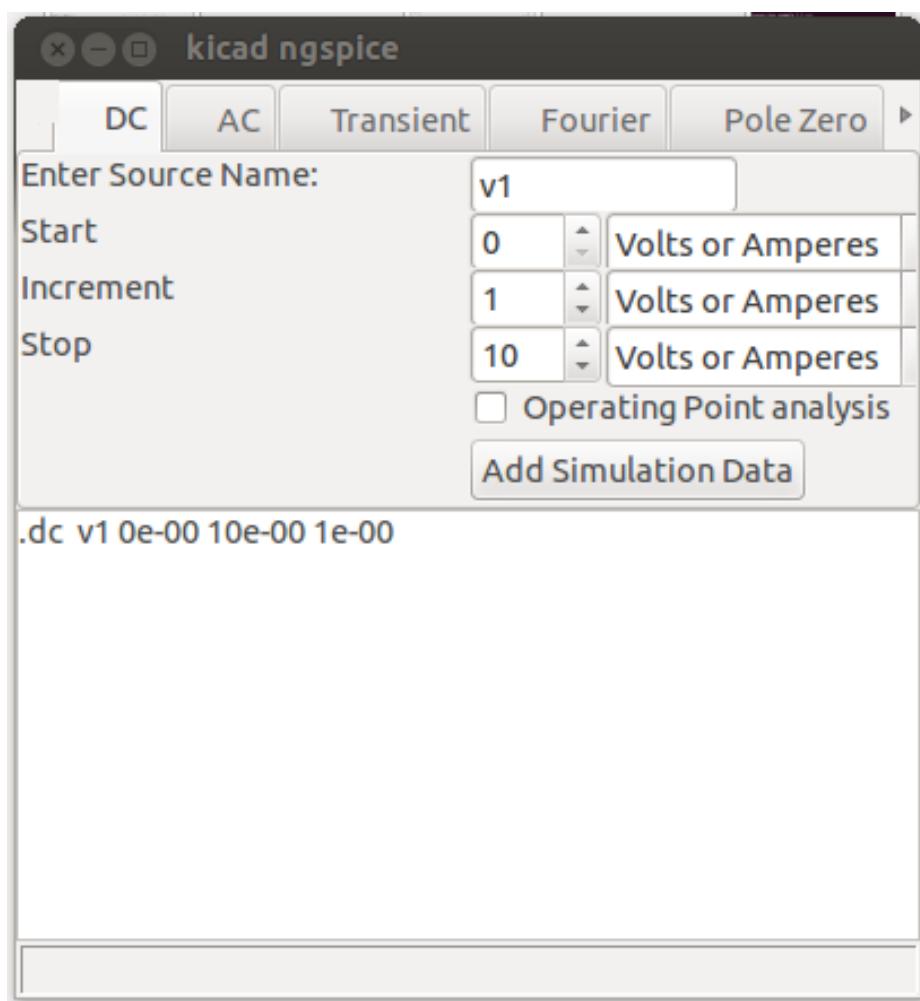
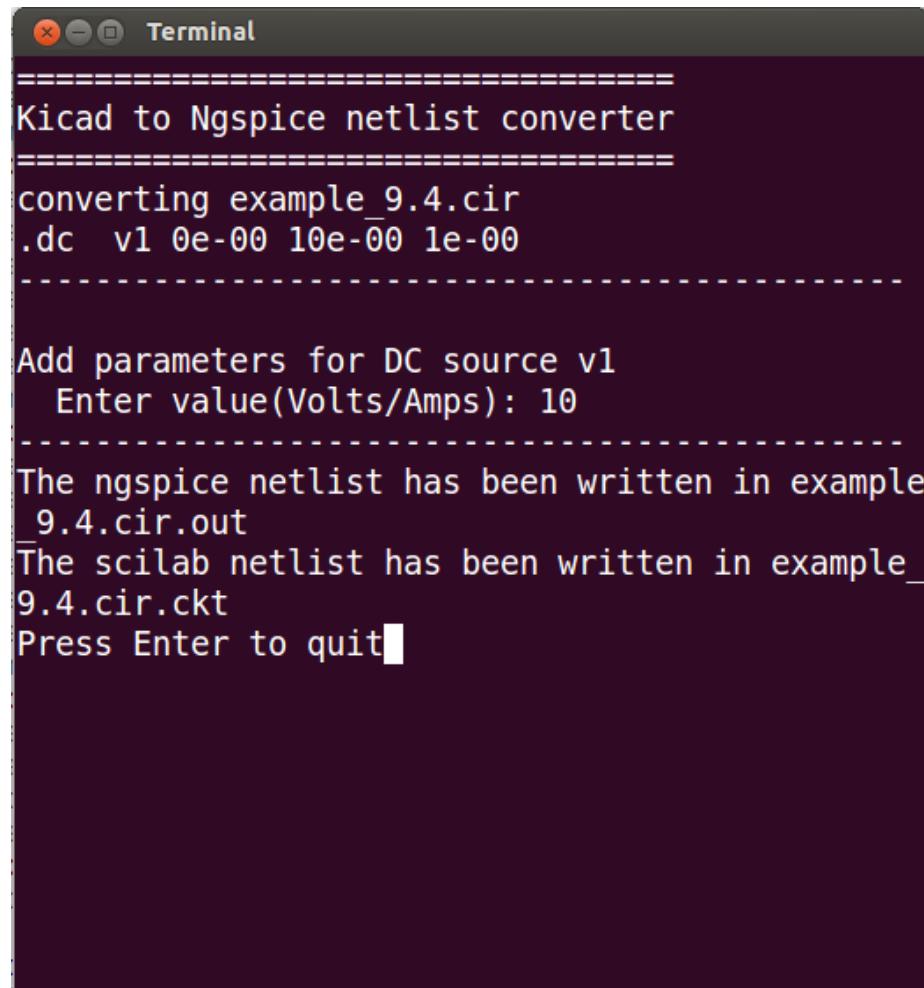


Figure A.36: Analysis Inserter



```
Terminal
=====
Kicad to Ngspice netlist converter
=====
converting example_9.4.cir
.dc v1 0e-00 10e-00 1e-00
-----
Add parameters for DC source v1
Enter value(Volts/Amps): 10
-----
The ngspice netlist has been written in example_9.4.cir.out
The scilab netlist has been written in example_9.4.cir.ckt
Press Enter to quit
```

Figure A.37: Kicad to Ngspice

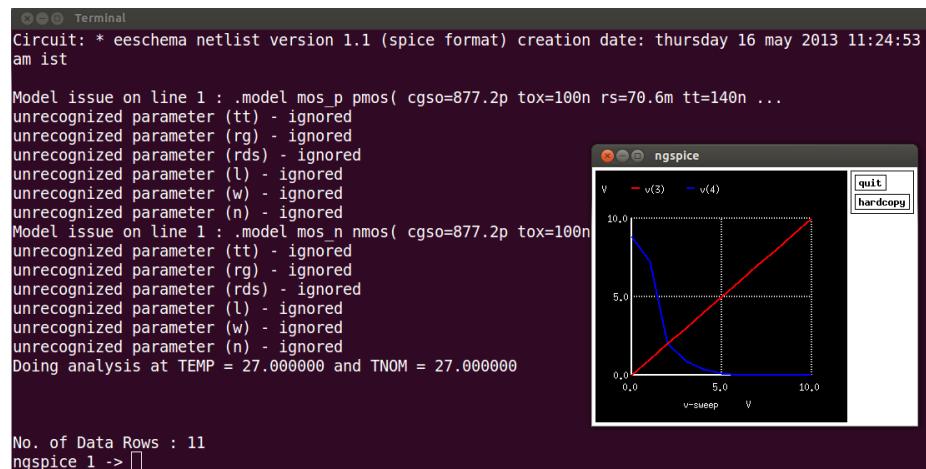


Figure A.38: Ngspice Waveform

Appendix B

Spoken Tutorial based SELF workshop on Oscad

B.1 What is SELF Workshop?

The Spoken Tutorial project is about teaching and learning a particular FOSS (Free and Open Source Software) like Linux, Scilab, LaTeX, PHP & MySQL, Java, C/C++, LibreOffice, Oscad etc. via an easy video tool- Spoken Tutorials. Some of the salient features of SELF (Spoken Tutorial based Education and Learning through Free FOSS study) workshops are -

- This method is highly conducive to self learning. Once one get started any student or faculty can master the FOSS.
- UG, PG or research scholar students and even teachers of Science, IT, Engineering, Commerce, Management disciplines can learn any of the FOSS.
- The training workshops are conducted remotely via a video conferencing tool (Skype). Typically, the first workshop at any college is of 2 hour duration.

B.2 What do we offer in Oscad workshops?

- Ubuntu Linux live CD with Oscad software and Spoken Tutorials on Oscad, KiCad and Ngspice.
- Instruction sheet to practice all tutorials.

B.3 What is expected from the organizing institute

1. To start of, any college which is organizing the workshop needs to get a computer lab ready with machines that support sound with Internet connectivity.
2. One computer per user, with audio/video functionality and DVD drive in working condition is required.
3. One Headphone per user is required.
4. There is no lower or upper limit on the number of participants. This just depends on the number of computers available.
5. For additional participants, laptops if available can be used.
6. The main organizer's computer should have a webcam via which Spoken Tutorial team can monitor the workshop.
7. On a convenient day, participants take an assessment test and on clearing it get certificates from Spoken Tutorial project, IIT Bombay.

Useful Links

1. Spoken Tutorials: <http://spoken-tutorial.org>
2. Oscad: <http://oscad.net>
3. FOSSEE project: <http://fossee.in>

Contact

For Spoken Tutorial related queries please write to: contact@spoken-tutorial.org
To contribute to Oscad project OR to know more about Oscad project: info@fossee.in

Bibliography

- [1] (2013, May). [Online]. Available: http://en.wikipedia.org/wiki/Electronic_design_automation
- [2] A. S. Sedra, K. C. Smith, and A. N. Chandorkar, *Microelectronics Circuits - Theory and Applications*. Oxford University Press, 2009.
- [3] (2013, May). [Online]. Available: <http://www.kicad-pcb.org/display/KICAD/KiCad+EDA+Software+Suite>
- [4] (2013, May). [Online]. Available: <http://ngspice.sourceforge.net/>
- [5] S. M. Sandler and C. Hymowitz, *SPICE Circuit Handbook*, 2006.
- [6] (2013, May). [Online]. Available: <http://www.scilab.org/>
- [7] J.-P. Charras and F. Tappero. (2013, May). [Online]. Available: <http://www.kicad-pcb.org/display/KICAD/KiCad+Documentation>
- [8] P. Nenzi and H. Vogt, *Ngspice Users Manual version 25plus*, 2013.

Index

- .ac, 61
- .dc, 58, 59
- .op, 58
- .tran, 62
- Aakash, 113
- AC Analysis Inserter, 61
- AC Small-signal Analysis, 23, 58, 68
- Analysis Inserter, 30, 57
- Analysis insertion, 57
- analysis option, 67
- Annotate, 54
- Backward Euler, 104
- BJT, 25
- BSIM, 26
- CAD, 21
- Circuit simulation, 57
- Component, 45
 - footprint, 74
 - mapping, 76
 - library, 46
 - connector, 48
 - Oscad libraries, 46
 - plot, 47
 - power, 47
 - move, 52
 - place, 51
 - reference, 48
 - rotate, 52
 - values, 53
- CvPcb, 22, 34
- DC Analysis, 23, 58, 68
- Design rules check, 22
- diode, 25
- EDA
 - tools, 5
 - design flow, 5
- EESchema, 40
 - toolbar
 - left, 44
 - right, 43
 - top, 42
- EKV, 26
- ERC, 54
- Footprint editor, 76
- Footprints, 34, 74
 - mapping, 76
 - move and place, 82
 - view
 - 2D, 77
 - 3D, 77
- gEDA, 26
- HICUM, 26
- hierarchical modeling, 92
- HiSim, 26
- Hotkeys
 - Layout editor, 81
 - Schematic Editor, 45
- HPGL, 22
- installModule.sh, 11
- installOSCADC.sh, 11
- JFET, 25
- KiCad to Ngspice conversion, 57
- Layout editor, 78
- Linux, 7

Load error message, 30
MES-FET, 25
metanet, 102
model library, 91
models and subcircuits, 67
MOSFET, 25

Netlist, 18, 55
 for PCB, 75
 for simulation, 55
Netlist converter, 32, 65
Newton-Raphson, 104
Ngspice, 33, 57
Ngspice simulation, 57, 68
normal mode, 102
numerical mode, 102

Operating point analysis, 23, 60
Oscad, 6
Oscad Installer, 8
Oscad tool bar, 28

PCB, 5, 74
PCB design
 design rules, 82
 design rules check, 87
 gerber, 87
 lay tracks, 82
 pcb edges, 84
PCB Layout, 21
 creation, 78
Pcbnew, 22, 34
plotting and printing statements, 67
Power Flag, 54
proxy, 12
PSP, 26
PTM, 26
python, 13

reference field, 90

Schematic, 40
 annotate, 54
 editor, 29, 40
 ERC, 54
 toolbar
left, 44
right, 43
top, 42
wiring, 53
Scilab, 102
Simulation, 67
simulation examples, 68
sinusoidal source, 65
SMCSim, 25, 102
SPECCTRA, 23
SPICE netlist, 65
symbolic mode, 102

Terminal, 11
Transient Analysis, 23, 58, 59, 70
Transient Analysis Inserter, 62
Types of Analysis, 57, 58

Ubuntu, 7