eSim

An open source EDA tool for circuit design, simulation, analysis and PCB design

eSim User Manual

version 1.1.0

Prepared By:
eSim Team
FOSSEE at IIT,Bombay

Indian Institute of Technology Bombay

June 2016

Contents

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 are called `Electronic Design Automation` or `EDA` tools.

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 be 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

A person who builds an electronic system has to first design the circuit, produce a virtual representation of it through a schematic for easy comprehension, simulate it and finally convert it into a Printed Circuit Board (PCB). There are various tools available that will help us do this. Some of the popular EDA tools are those of `Cadence`, `Synopys`, `Mentor Graphics` and `Xilinx`. Although these are fairly comprehensive and high end, their licenses are expensive, being proprietary.

There are some free and open source EDA tools like `gEDA`, `KiCad` and `Ngspice`. The main drawback of these 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`). To the best of our knowledge, there is no open source software that can perform circuit design, simulation and layout design together. eSim is capable of doing all of the above.

eSim is a free and open source EDA tool. It is an acronym for **E**lectronics **Sim**ulation. eSim is created using open source software packages, such as KiCad, Ngspice and

Python. Using eSim, one can create circuit schematics, perform simulations and design PCB layouts. It can create or edit new device models, and create or edit subcircuits for simulation.

Because of these reasons, eSim is expected to be useful for students, teachers and other professionals who would want to study and/or design electronic systems. eSim is also useful for entrepreneurs and small scale enterprises who do not have the capability to invest in heavily priced proprietary tools.

This book introduces eSim to the reader and illustrates all the features of eSim with examples. The software architecture of eSim is presented in Chapter 2 while Chapter 3 gives the user step by step instructions to install eSim on a typical computer system. Chapter 4 gets the user started with eSim. It takes them through a tour of eSim with the help of a simple RC circuit example. Chapter 5 illustrates how to create the circuit schematic in esim and Chapter 6 explains simulating the circuit schematic. The advanced features of eSim such as Model Builder and Sub circuit Builder are covered in Chapter 7 and Chapter 8 respectively. Additional features in eSim like mixed mode simulation and OpenModelica are covered in Chapter 9 and Chapter 10 respectively. Chapter 11 illustrates how to use eSim for solving circuit simulation problems. The last chapter, Chapter 12 explains how eSim can be used to do PCB layout.

The following convention has been adopted throughout this manual.All the menu names, options under each menu item, tool names, certain points to be noted, etc., are given in *italics*. Some keywords, names of certain windows/dialog boxes, names of some files/projects/folders, messages displayed during an activity, names of websites, component references, etc., are given in `typewriter` font. Some key presses, e.g. `Enter` key, `F1` key, `y` for yes, etc., are also mentioned in `typewriter` font.

Chapter 2

Architecture of eSim

eSim is a CAD tool that helps electronic system designers to design, test and analyse their circuits. But the important feature of this tool is that it is open source and hence the user can modify the source as per his/her need. The software provides a generic, modular and extensible platform for experiment with electronic circuits. This software runs on all Ubuntu Linux distributions and some flavours of Windows. It uses `Python`, `KiCad` and `Ngspice`.

The objective behind the development of eSim is to provide an open source EDA solution for electronics and electrical engineers. The software should be capable of performing schematic creation, PCB design and circuit simulation (analog, digital and mixed signal). It should provide facilities to create new models and components. The architecture of eSim has been designed by keeping these objectives in mind.

2.1 Modules used in eSim

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

2.1.1 Eeschema

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. It is the schematic editor tool used in KiCad. Eeschema is intended to work with PCB layout software such as Pcbnew. It provides netlist that 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 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:

1. Design rules check (DRC) for the automatic control of incorrect connections and inputs of components left unconnected. 2. Generation of layout files in POSTSCRIPT or HPGL format. 3. Generation of layout files printable via printer. 4. Bill of materials generation. 5. Netlist generation for PCB layout or for simulation.

This module is indicated by the label 1 in Fig. 2.1.

As Eeschema is originally intended for PCB Design, there are no fictitious components[1] such as voltage or current sources. Thus, we have added a new library for different types of voltage and current sources such as sine, pulse and square wave. We have also built a library which gives printing and plotting solutions. This extension, developed by us for eSim, is indicated by the label 2 in Fig. 2.1.

2.1.2 CvPcb

CvPcb is a tool that allows the user to associate components in the schematic to component footprints when designing the printed circuit board. CvPcb is the footprint editor tool in KiCad. Typically the netlist file generated by Eeschema does not specify which printed circuit board footprint is associated with each component in the schematic. However, 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 that the correct footprint is associated with 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 its footprint. This interactive approach is simpler and less error prone than directly associating footprints in the schematic editor. This is because CvPcb not only allows automatic association, but also allows to see the list of available footprints and displays them on the screen to ensure the correct footprint is being associated. This module is indicated by the label 3 in Fig. 2.1.

2.1.3 Pcbnew

Pcbnew is a powerful printed circuit board software tool. It is the layout editor tool used in KiCad. It is used in association with the schematic capture software Eeschema, which provides the netlist. Netlist describes the electrical connections of the circuit. CvPcb is used to assign each component, in the netlist produced by Eeschema, to a module that is used by Pcbnew. The features of Pcbnew are given below:

[1] Signal generator or power supply is not a single component but in circuit simulation, we consider them as a component. While working with actual circuit, signal generator or power supply gives input to the circuit externally thus, doesn't require for PCB design.

- 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 new components, or by modifying any value (and under certain conditions any reference) of old or new modules, according to the electrical connections appearing in the schematic.

- This tool provides a rats nest display, a hairline connecting the pads of modules 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 automatically generates a copper plane, with or without thermal breaks on the pads.

- It has a simple but effective auto router to assist in the production of the circuit. An export/import in `SPECCTRA` dsn format allows to use more advanced autorouters.

- It 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 such as:
  - display in full or outline.
  - display the track/pad clearance.

This module is indicated by the label 4 in Fig. 2.1.

2.1.4 KiCad to Ngspice converter

Analysis parameters, and the source details are provided through this module. It also allows us to add and edit the device models and subcircuits, included in the circuit schematic. Finally, this module facilitates the conversion of KiCad netlist to Ngspice compatible ones.
It is developed by us for eSim and it is indicated by the label 7 in Fig. 2.1. The use of this module is explained in detail in section (yet to be put).

2.1.5 Model Builder

This tool provides the facility to define a new model for devices such as, 1. Diode 2. Bipolar Junction Transistor (BJT) 3. Metal Oxide Semiconductor Field Effect Transistor (MOSFET) 4. Junction Field Effect Transistor (JFET) 5. IGBT and 6. Magnetic core. This module also helps edit existing models. It is developed by us for eSim and it is indicated by the label 5 in Fig. 2.1.

2.1.6 Subcircuit Builder

This module allows the user to create a subcircuit for a component. Once the subcircuit for a component is created, the user can use it in other circuits. It has the facility to define new components such as, Op-amps and IC-555. This component also helps edit existing subcircuits. This module is developed by us for eSim and it is indicated by the label 6 in Fig. 2.1.

2.1.7 Ngspice

Ngspice is a general purpose circuit simulation program for nonlinear dc, nonlinear transient, and linear ac analysis. 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, MESFETs, and MOSFET. This module is indicated by the label 9 in Fig. 2.1.

2.1.8 NGHDL

NGHDL, a module for mixed mode circuit simulation is also integrated with eSim. It uses ghdl for digital simulation and the mixed mode simulation happens through NgSpice.

2.1.9 OpenModelica

OpenModelica (OM) is an open source modeling and simulation tool based on Modelica language. Two modules of OpenModelica, OMEdit, an IDE for modeling and simulation and OMOptim, an IDE for optimisation are integrated with eSim.

2.2 Work flow of eSim

Fig. 2.1 shows the work flow in eSim. The block diagram consists of mainly three parts:

- Schematic Editor

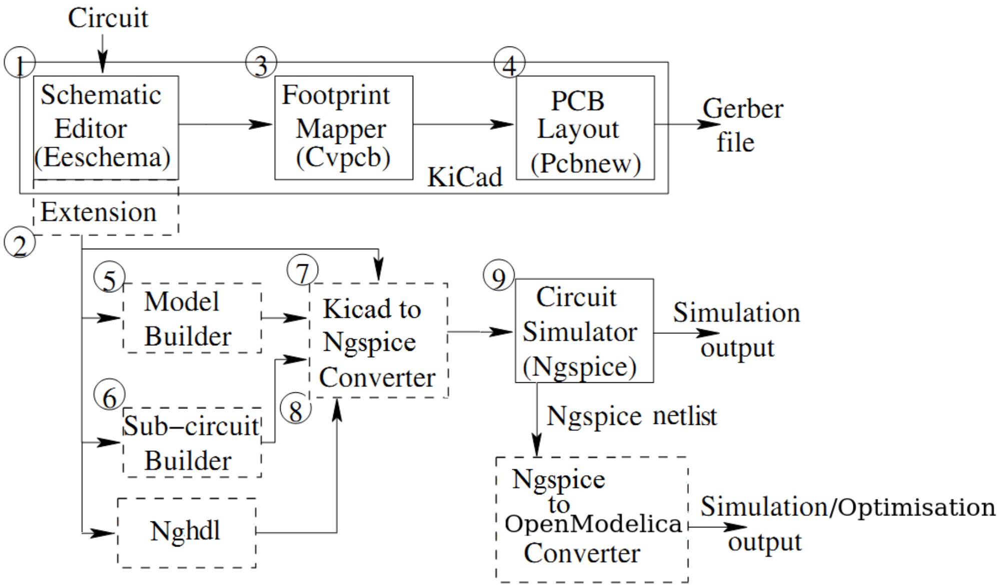


Figure 2.1: Work flow in eSim. (Boxes with dotted lines denote the modules developed in this work).

- PCB Layout Editor
- Circuit Simulators

Here we explain the role of each block in designing electronic systems. 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 eSim. Thus all the functionalities of Eeschema are naturally available in eSim.

Libraries for components, explicitly or implicitly supported by Ngspice, have been created using the features of Eeschema. 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 and square wave, has been added in eSim. A library which gives the functionality of printing and plotting has also been created.

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 perform component to footprint mapping, CvPcb is used. 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 the case of large electronic circuits, breadboard testing is impractical. In such cases, 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 the facility to define a new model for devices and edit existing models. Complex circuit elements can be created by hierarchical modeling. Subcircuit Builder provides an easy way to create a subcircuit.

The netlist generated by Schematic Editor cannot be directly used for simulation due to compatibility issues. Netlist Converter converts it into Ngspice compatible format. The type of simulation to be performed and the corresponding options are provided through a graphical user interface (GUI). This is called KiCad to Ngspice Converter in eSim.

eSim uses Ngspice for analog, digital, mixed-level/mixed-signal circuit simulation. Ngspice is based on three open source software packages

- 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, and PTM models. It is widely used due to its accuracy even for the latest technology devices.

Chapter 3

Installing eSim

3.1 eSim installation in Ubuntu

1. Download eSim installer for Linux from `http://esim.fossee.in/downloads` to a local directory and unpack it.
   You can also unpack the installer through the terminal. Open the terminal and navigate to the directory where this INSTALL file is located. Use the following command to unpack:
   ```
   $ unzip eSim-1.1.2.zip
   ```

2. To install eSim and other dependecies run the following command.
   ```
   $ ../install-linux.sh --install
   ```
   Above script will install eSim along with the dependencies.

3. To run eSim from the terminal, type:
   ```
   $ esim
   ```
   or you can double click on `eSim` icon created on the Desktop after installation.

3.2 eSim installation in Windows

1. Download eSim from website http://esim.fossee.in/downloads
   `$ eSim-Windows-Installer.zip file`.

2. Open eSim-Windows-Installer folder, right click on `Setup file` and select `run as administrator`. Click `Yes` and `Next` to complete the installation.

3. **eSim** icon will be created on deskop. You can double click on the `eSim` icon created on the Desktop after installation.

Chapter 4

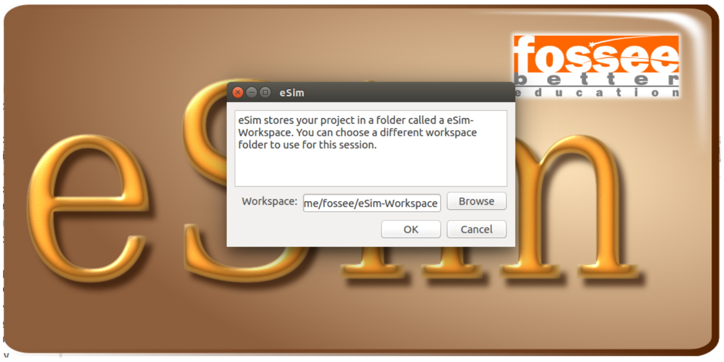
Getting Started

In this chapter we will get started with eSim. Referring to this chapter will make one familiar with eSim and will help plan the project before actually designing a circuit. Lets get started.

4.1 How to launch eSim in Ubuntu?

After the installation of eSim, a shortcut to eSim is created on the Desktop. To launch eSim double click on the shortcut.
Alternately, you can also launch esim from the terminal.
1. Go to terminal.
2. Type **esim** and press `Enter`.

The first window that appears is workspace dialog as shown in Fig. 4.1.


eSim stores your project in a folder called a eSim-Workspace. You can choose a different workspace folder to use for this session.

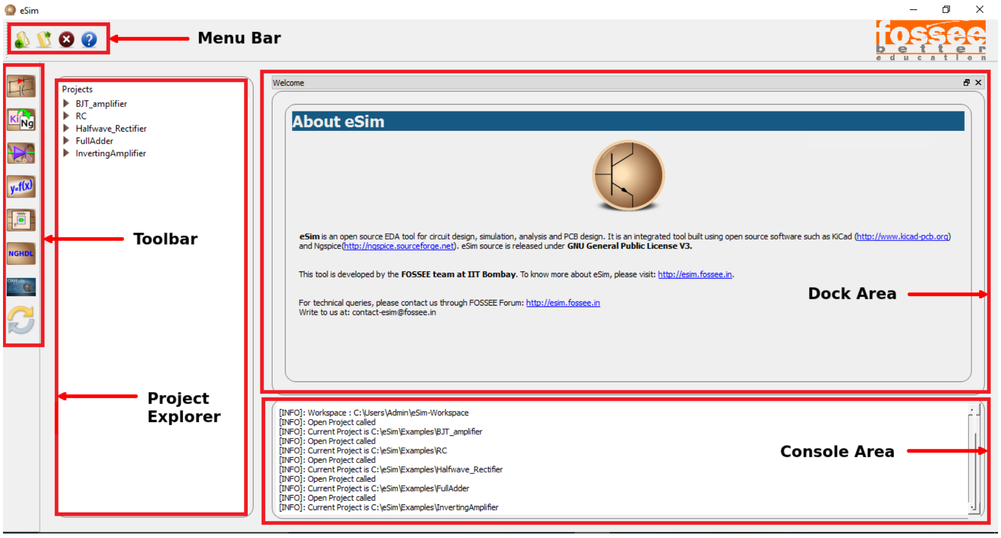
Workspace: me/fossee/eSim-Workspace

Figure 4.1: eSim-Workspace

The default workspace is eSim-Workspace under home directory. To create new workspace use `browse` option.

4.2 eSim User Interface

The main graphic window of eSim is as shown in Fig. 4.2.



eSim is an open source EDA tool for circuit design, simulation, analysis and PCB design. It is an integrated tool built using open source software such as KiCad (http://www.kicad-pcb.org) and Ngspice(http://ngspice.sourceforge.net). eSim source is released under **GNU General Public License V3.**

This tool is developed by the **FOSSEE team at IIT Bombay.** To know more about eSim, please visit: http://esim.fossee.in.

For technical queries, please contact us through FOSSEE Forum: http://esim.fossee.in
Write to us at: contact-esim@fossee.in

```
[INFO]: Workspace : C:\Users\Admin\eSim-Workspace
[INFO]: Open Project called
[INFO]: Current Project is C:\eSim\Examples\BJT_amplifier
[INFO]: Open Project called
[INFO]: Current Project is C:\eSim\Examples\RC
[INFO]: Open Project called
[INFO]: Current Project is C:\eSim\Examples\Halfwave_Rectifier
[INFO]: Open Project called
[INFO]: Current Project is C:\eSim\Examples\FullAdder
[INFO]: Open Project called
[INFO]: Current Project is C:\eSim\Examples\InvertingAmplifier
```

Figure 4.2: eSim Main GUI

The eSim window consists of the following sections.

1. Toolbar
2. Menubar
3. Project explorer
4. Dockarea
5. Console area

4.2.1 Toolbar

The toolbar consists of the following buttons. See Fig. 4.3.

Open Schematic

The first button on the toolbar is the *Schematic Editor*. Clicking on this button will open Eeschema, the KiCad schematic editor. If a new project is being created, one will get a dialog box confirming the creation of a schematic. This is illustrated in See Fig. 4.4. However, if an already existing project is opened, the schematic editor window is opened. To know how to use the schematic editor to create circuit schematics, refer to Chapter 5.

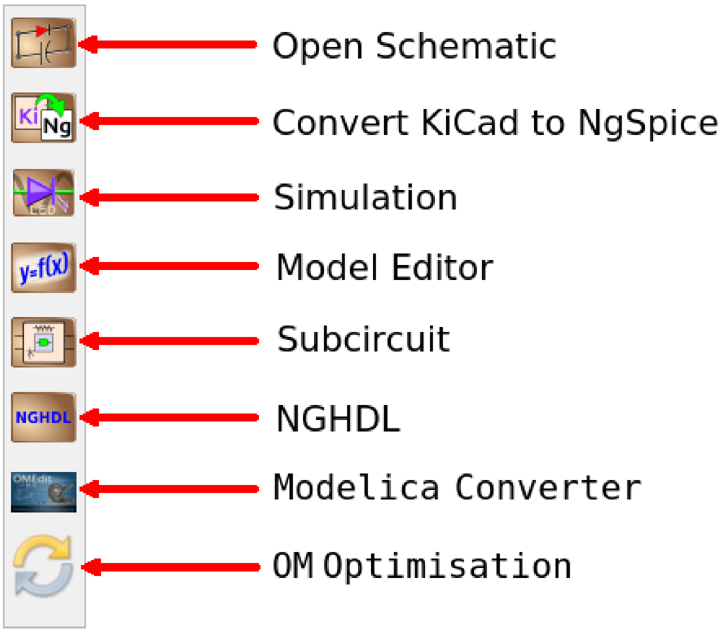
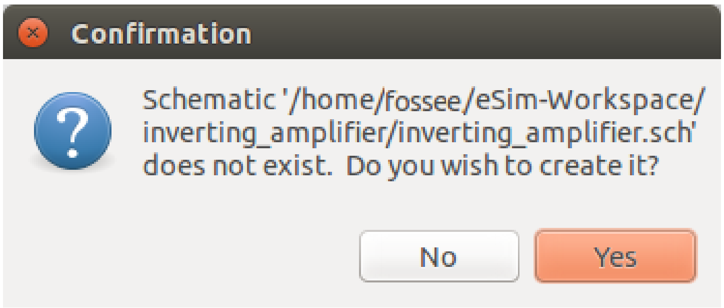


Figure 4.3: Toolbar


Confirmation

Schematic '/home/fossee/eSim-Workspace/inverting_amplifier/inverting_amplifier.sch' does not exist. Do you wish to create it?

Figure 4.4: Confirmation for schematic creation

Convert KiCad to Ngspice

In the Eeschema window, after creating the schematic the netlist is genarated. This tool converts the generated KiCad spice netlist into Ngspice compatible netlist. The KiCad to Ngspice window consists of five tabs namely *Analysis, Source Details, Ngspice Model, Device Modeling and Subcircuits.*

The details of these tabs are as follows.

- *Analysis:* This feature helps the user to insert the parameters for performing different types of analysis such as Operating point analysis, DC analysis, AC analysis, transient analysis, DC Sweep Analysis.

It has the facility to insert

  - Type of analysis such as AC or DC or Transient
  - Values for analysis

- *Source Details:* eSim sources are added from `eSim_Sources` library. Source such as *SINE, AC, DC, PULSE* are in this library. The parameter values to all the sources added in the schematic can be given through 'Source Details'.

- *Ngspice Model:* Ngspice has in-built model such as *flipflop(D,SR,JK,T), gain,summer* etc. which can be utilised while building a circuit. eSim allows to add and modify Ngspice model parameter through Ngspice Model tab.

- *Device Modeling:* Devices like *Diode, JFET, MOSFET, IGBT, MOS* etc used in the circuit can be modeled using device model libraries. eSim also provides editing and adding new model libraries. While converting KiCad to Ngspice, these library files are added to the corresponding devices used in the circuit.

- *Subcircuits:* eSim allows you 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. The subcircuits are added to the main circuits using this facility. We can also edit already existing subcircuits.
  Once the values have been entered, press the `Convert` button. This will generate the `.cir.out` file in the same project directory. Note that *KiCad to Ngspice Converter* can only be used if the KiCad spice netlist `.cir` file is already generated.

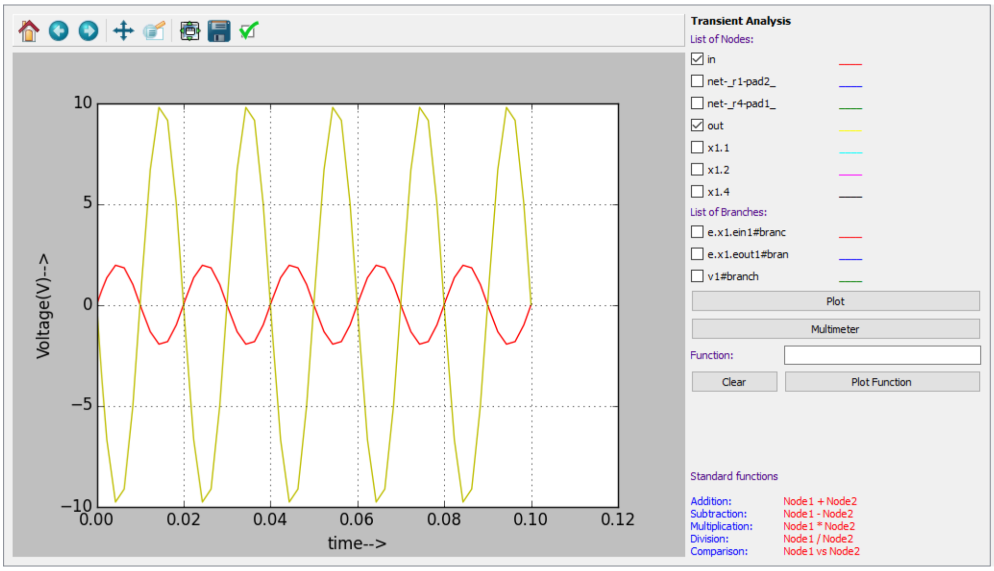
Simulation

The netlist generated using the *KiCad to Ngspice* converter is simulated using *Simulation* button on the eSim left toolbar. This will run the Ngspice simulation for current project. eSim have two options to see the simulation output. The first one is the Python plotting window which opens up in the dock area, as shown in Fig. 4.5. The second is the Ngspice window with the simulation data. The user can type in Ngspice commands to view the plots.

Note: If the user has used the plot components (available under eSim_Plot library) at various nodes in the circuit schematic the Ngspice plots are displayed automatically.

Model Editor

eSim also gives an option to re-configure the model library of a device. It facilitates the user to change model library of devices such as diode, transistor, MOSFET, etc.



Transient Analysis

List of Nodes:

☑ in
□ net-_r1-pad2_
□ net-_r4-pad1_
☑ out
□ x1.1
□ x1.2
□ x1.4

List of Branches:

□ e.x1.ein1#branc
□ e.x1.eout1#bran
□ v1#branch

Function:

Standard functions

Addition:	Node1 + Node2
Subtraction:	Node1 - Node2
Multiplication:	Node1 * Node2
Division:	Node1 / Node2
Comparison:	Node1 vs Node2

Figure 4.5: Simulation Output in Python Plotting Window

Subcircuit

eSim also allows the user to build subcircuits. The subcircuits can again have components having subcircuits and so on. This enable 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 to recreate it. Clicking on *Subcircuit Builder* tool will allow one to edit or create a subcircuit. To know how to make a subcircuit, refer to Chapter 8.

nghdl

Nghdl is an add on to esim for mixed mode circuit simulation. It uses ghdl for digital simulation and the mixed mode simulation is done through the Ngspice. It takes digital values from ghdl using socket programming.

Modelica Converter

OpenModelica (OM) is an open source modeling and simulation tool based on Modelica language. Modelica is an object oriented language. The Modelica Converter in eSim interface, converts the ngspice netlist to Modelica format. This facility will be only available if you have OpenModelica already installed in the system. More details on how to use this module is available in Chapter 10.

OM Optimisation

OMOptimisation (OMOptim) is a powerful and interactive tool for performing design optimisation. It has a good library of electrical components called Modelica Standard Library (MSL). OMOptim is stable and robust. It is very easy to add objective functions to the OMOptim interface.

4.2.2 Menubar

- New Project: New projects are created in the eSim-Workspace. When this menu is selected, a new window opens up with `Enter Project name` field. Type the name of the new project and click on `OK`. A project directory will be created in eSim-Workspace. The name of this folder will be the same as that of the project created. *Make sure that the project name does not have any spaces in between.* This project is also added to the project explorer.

- Open Project: This opens the file dialog of defalut eSim-Workspace where the projects are stored. Select the required project and click on `Open`. The selected project is added to the project explorer.

- Exit: This button closes the opened project.

- Help: It opens user manual in the browser.

4.2.3 Project Explorer

Project explorer contains the list of all the projects previously added to it. Select a project and double click on it, this will display all the files under this project. Right click on any displayed file to open it. To remove or refresh any project file from the project explorer, right click on the main project file.

4.2.4 Dockarea

This area is used to open the following windows.

1. KiCad to Ngspice converter

2. Ngspice plotting

3. Python plotting

4. Model builder

5. Subcircuit builder

Modules/Windows will appear here as per your selection.

4.2.5 Console Area

Console area provides the log information about the activity done during the current session.

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`. eSim 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 Familiarizing the Schematic Editor interface

Fig. 5.1 shows the schematic editor and the various menu and toolbars. We will explain them briefly in this section.

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 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) Page Settings - Set preferences for printing the page.
   (h) Print - Access to print menu (See Fig. 5.2).
   (i) Plot - Plot the schematic in Postscript, HPGL, SVF or DXF format
   (j) Close - Close the schematic editor.

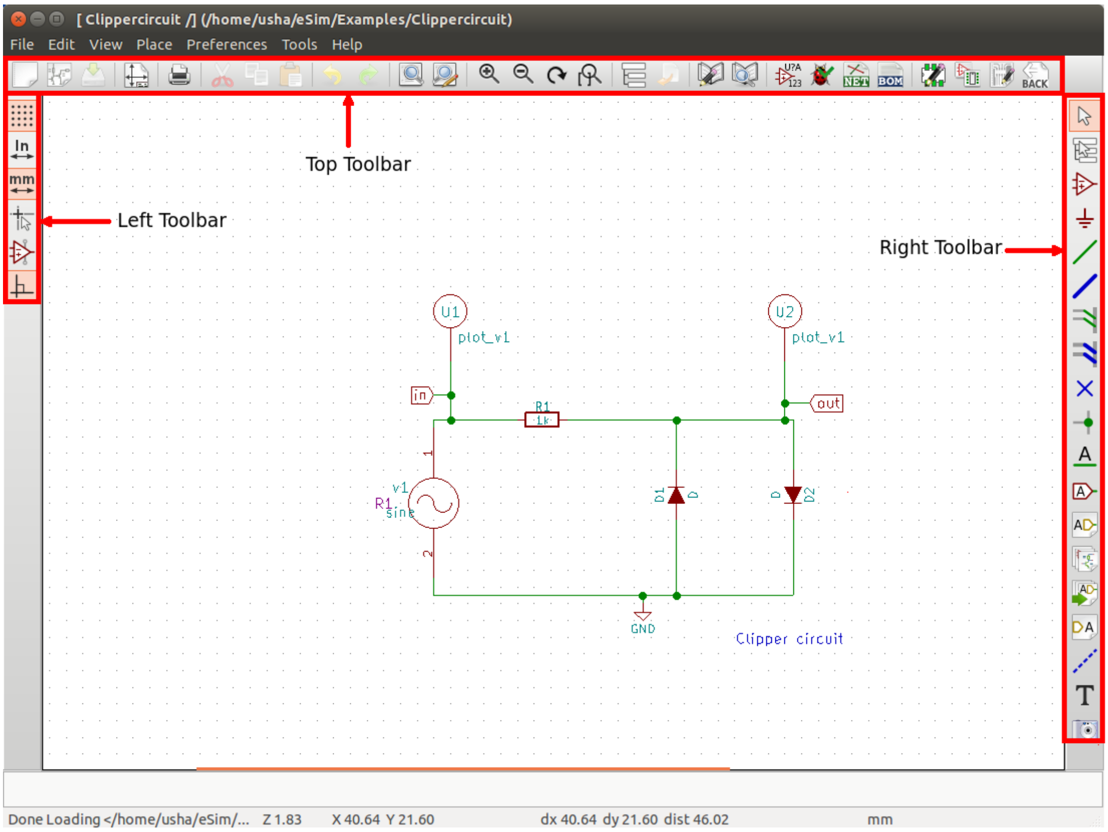
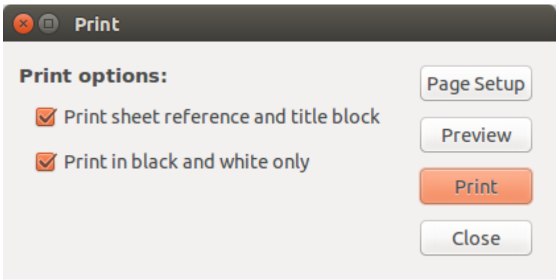

[Clippercircuit /] (/home/usha/eSim/Examples/Clippercircuit)
File Edit View Place Preferences Tools Help
Top Toolbar
Left Toolbar
Right Toolbar
U1
plot_v1
U2
plot_v1
in
out
R1
1k
v1
R1
sine
D1
D
D
D2
GND
Clipper circuit
Done Loading </home/usha/eSim/... Z 1.83 X 40.64 Y 21.60 dx 40.64 dy 21.60 dist 46.02 mm

Figure 5.1: Schematic editor with the menu bar and toolbars marked



Print options:

☑ Print sheet reference and title block

☑ Print in black and white only

Figure 5.2: Print options

2. Place - The place menu has shortcuts for placing various items like components, wire and junction, on to the schematic editor window. See Sec. 5.1.5 to know more about various shortcut keys (hotkeys).
3. Preferences - The preferences menu has the following options:

(a) Component Libraries - Select component libraries and library paths

(b) Schematic Editor Options - Select colors for various items, display options and set hot keys.

(c) Language - Shows the current list of available languages. Use default.

(d) Import and Export - Contain options to load and save preferences and import/ export hot key configuration files. See Sec. 5.1.5 to know about various hotkeys.

5.1.2 Top toolbar

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

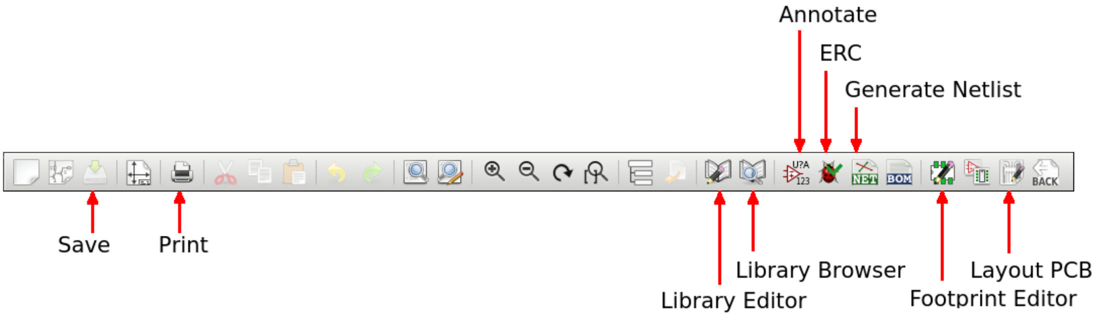


Figure 5.3: Toolbar on top with important tools marked

1. Save - Save the current schematic
2. Print - Print the schematic
3. Navigate schematic hierarchy - Navigate among the root and sub-sheets in the hierarchy
4. Library Editor - Create or edit components.
5. Library Browser - Browse through the various component libraries available
6. Annotate - Annotate the schematic
7. Check ERC - Do Electric Rules Check for the schematic
8. Generate netlist - Generate a netlist for PCB design or for simulation.
9. Create BOM - Create a Bill of Materials of the schematic
10. Footprint editor - Map each component in the PCB netlist to a footprint
11. Layout PCB - Lay tracks between the footprints to get the PCB layout

5.1.3 Toolbar on the right

The toolbar on the right side of the schematic editor window has many important tools. Some of them are marked in Fig. 5.4. Let us now look at each of these tools and their

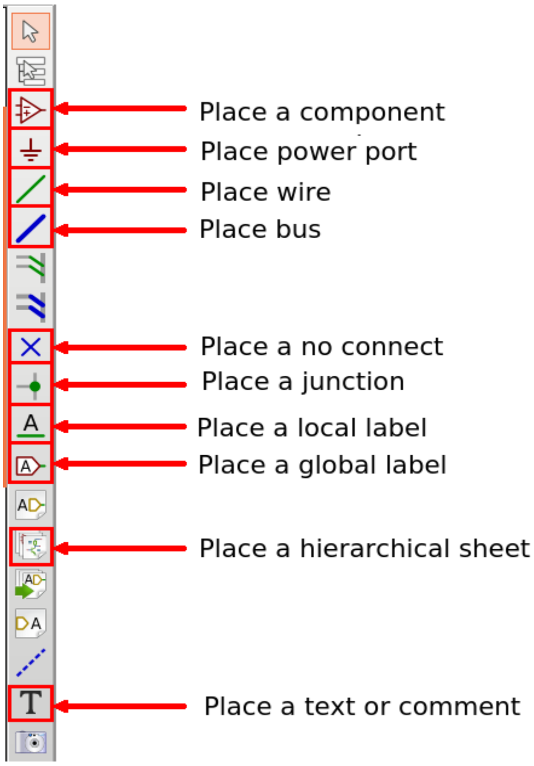


Figure 5.4: Toolbar on right with important tools marked

uses.

1. Place a component - Load a component to the schematic. See Sec. 5.3.1 for more details.
2. Place a power port - Load a power port (Vcc, ground) to the schematic
3. Place wire - Draw wires to connect components in schematic
4. Place bus - Place a bus on the 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 within the root sheet in the hierarchy. Hierarchical schematics is a good solution for big projects
9. Place a text or comment - Place a text or comment in the 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 Fig. 5.5.

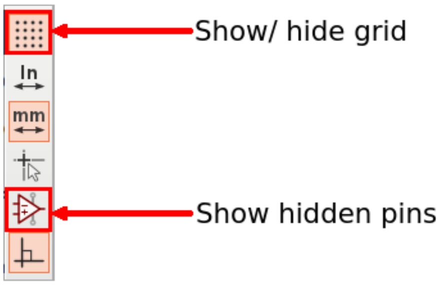


Figure 5.5: Toolbar on left with important tools marked

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

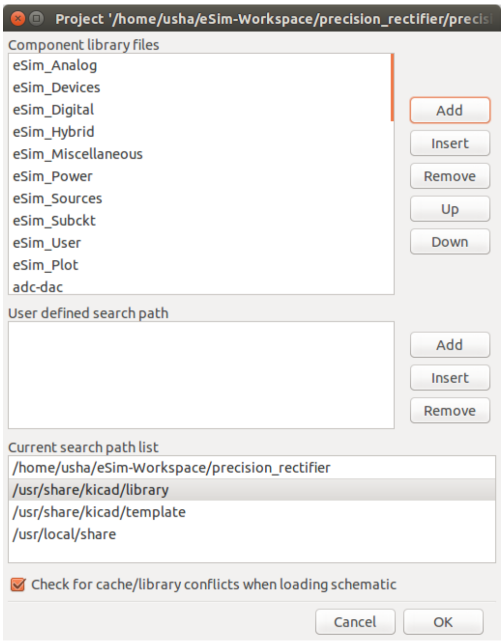
A set of keyboard keys are associated with various operations in the schematic editor. These keys save time and make it easy to switch from one operation to another. The list of hotkeys can be viewed by going to Preferences in the top menu bar. Choose *Schematic Editor Options* and select *Controls* tab. The hotkeys can also be edited here. Some frequently used hotkeys, along with their functions, are given 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: Both lower and upper-case keys will work as hotkeys.

5.2 eSim component libraries

eSim schematic editor has a huge collection of components. All the component libraries in EEschema, on which eSim 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 and current sources). A set of component libraries has been created with such components. If one is using eSim only for designing a PCB, then one might not need these libraries. However, these libraries are essential if one needs to simulate one's circuit. Hereafter, we will refer to these libraries as eSim libraries to distinguish them from libraries already present in EEschema (EEschema libraries) as shown in Fig. 5.6.



Project '/home/usha/eSim-Workspace/precision_rectifier/precis

Component library files

eSim_Analog
eSim_Devices
eSim_Digital
eSim_Hybrid
eSim_Miscellaneous
eSim_Power
eSim_Sources
eSim_Subckt
eSim_User
eSim_Plot
adc-dac

User defined search path

Current search path list

/home/usha/eSim-Workspace/precision_rectifier
/usr/share/kicad/library
/usr/share/kicad/template
/usr/local/share

☑ Check for cache/library conflicts when loading schematic

Figure 5.6: eSim-Components Libraries

The following list shows the various eSim component libraries.

- *eSim_Analog* - Contains Ngspice analog models such as aswitch(analog switch), summer(adder model), Transfo(Transformer), zener
- *eSim_Devices* - Includes elementary components like resistor, capacitor, transistor, mosfet
- *eSim_Digital* - Includes Ngspice digital models such as basic gates (AND, OR, NOR,NAND,XOR), filpflops (SR, D, JK), buffer, inverter

- *eSim_Hybrid* - Includes components like ADC and DAC
- *eSim_Miscellaneous* - Contains components like ic(used for giving initial conditions in circuit) and port(used in creating subcircuits)
- *eSim_Plot* - Contains plotting components like plot_v1(plot voltage at a node), plot_v2(plot voltage between 2 nodes),plot_i2(plot current through branch), plot_log(plot logarithmic voltage at a node)
- *eSim_Power* - Includes power components like DIAC, TRIAC and SCR
- *eSim_Sources* - Contains sources for the ciruits like AC voltage source, DC voltage source, sine source and pulse source
- *eSim_Subckt* - Contains subcircuit components like Op-Amp(UA 741), IC 555, Half adder and fulladder
- *eSim_User* - A repository for all user created components

5.3 Schematic creation for simulation

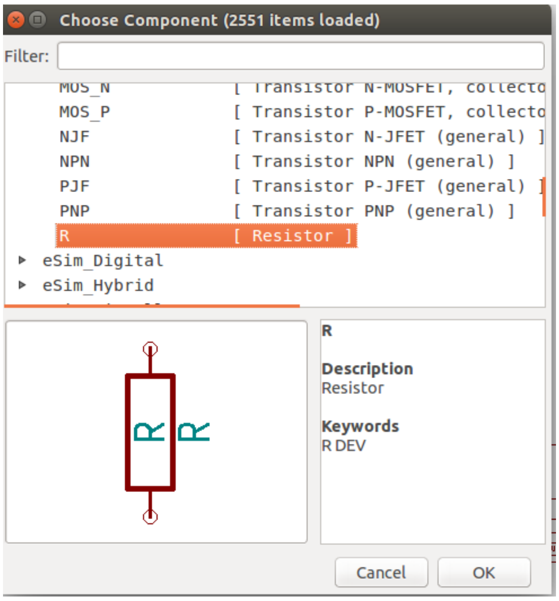
There are certain differences between the schematic created for simulation and that created for PCB design. We need certain components like plots and current sources. For simulation whereas these are not needed for PCB design. For PCB design, we would require connectors (e.g. DB15 and 2 pin connector) for taking signals in and out of the PCB whereas these have no meaning in simulation. This section covers schematic creation for simulation. Refer to Chapter 12 to know how to create schematic for PCB design.

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 Fig. 5.10c and do a transient simulation.

5.3.1 Selection and placement of components

We would need a resistor, a capacitor, a voltage source, ground terminal and some plot components. To place a resistor on the schematic editor window, 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. This action can also be performed by pressing the key A. Choose the *eSim_Devices* library and click on the arrow near it. This will open the *eSim_Devices* library and the resistor component can be found here. Fig. 5.7 shows the selection of resistor component. 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. The capacitor component is also found under *eSim_Devices* library. Select it and then 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. On the component selection window, choose the library *eSim_source*. Select



Choose Component (2551 items loaded)

Filter:

```
MOS_N          [ Transistor N-MOSFET, collecto
MOS_P          [ Transistor P-MOSFET, collecto
NJF            [ Transistor N-JFET (general) ]
NPN            [ Transistor NPN (general) ]
PJF            [ Transistor P-JFET (general) ]
PNP            [ Transistor PNP (general) ]
R              [ Resistor ]
▸ eSim_Digital
▸ eSim_Hybrid
```

R

Description
Resistor

Keywords
R DEV

Figure 5.7: Placing a resistor using the Place a Component tool

the component `SINE` and click on `OK`. Place the sine source on the schematic editor by a single click. Similarly select and place `gnd`, a ground terminal from the `power` library.

The plot components can be found under the eSim_Plot library. Select the plot_v1 component and place the component. Once all the components are placed, the schematic editor would look like as in Fig. 5.8.

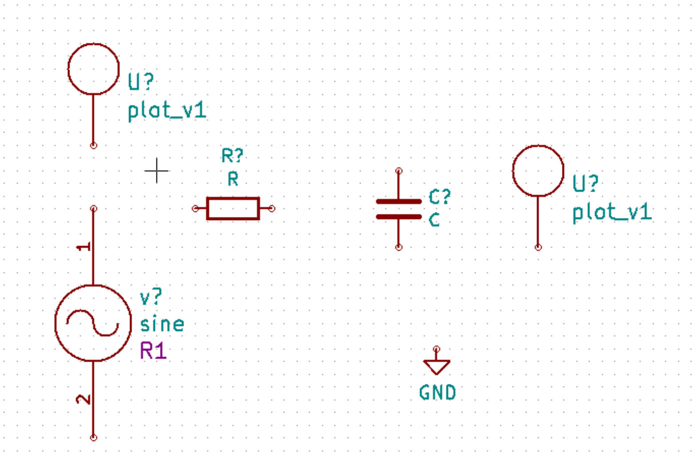


Figure 5.8: All RC circuit components placed

Let us rotate the resistor to complete the circuit. To rotate the resistor, place the cursor on the resistor as shown in Fig. 5.9 and press the key R. This applies to all components.

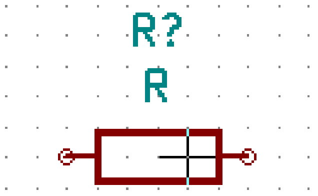
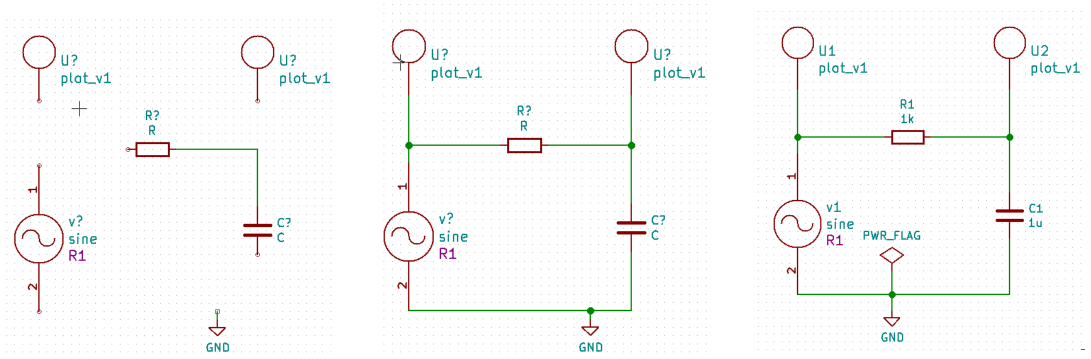


Figure 5.9: Placing the cursor (cross mark) on the resistor component

If one wants 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.

5.3.2 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 to be connected and press the key W. It has now changed to the wiring mode. Alternately, this can also be done by selecting the *Place wire* tool on the right side toolbar. Move the cursor towards the terminal of the capacitor and click on it. A wire is formed as shown in Fig. 5.10a. Similarly connect the wires between all terminals and the final schematic would look like Fig. 5.10b.



(a) Initial stages

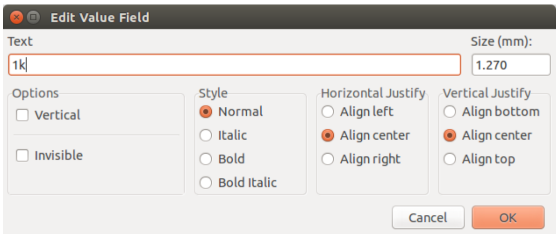
(b) Wiring done

(c) Final schematic with PWR_FLAG

Figure 5.10: Various stages of wiring

5.3.3 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 the 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 Fig. 5.11. 1k means $1k\Omega$. Similarly give the value `1u` for the capacitor. 1u means $1\mu F$.



Edit Value Field

Text: 1k

Size (mm): 1.270

Options
□ Vertical
□ Invisible

Style
● Normal
○ Italic
○ Bold
○ Bold Italic

Horizontal Justify
○ Align left
● Align center
○ Align right

Vertical Justify
○ Align bottom
● Align center
○ Align top

Figure 5.11: Editing value of resistor

5.3.4 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 `annotate`, then click on `OK` and finally click on close as shown in Fig. 5.12. 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.

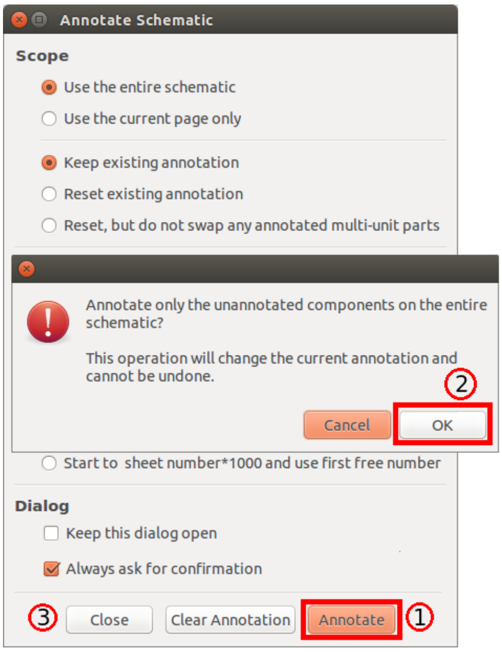
Let us now do `ERC` or `Electric Rules Check`. To do so, click on *Perform electric rules check* tool from the top toolbar. Click on *Test Erc* button. The error as shown in Fig. 5.13 may be displayed. Click on close in the test erc window.

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 Fig. 5.14.

To correct this error, place a `PWR_FLAG` from the Eeschema library *power*. Connect the power flag to the ground terminal as shown in Fig. 5.10c. Repeat the ERC. Now there are no errors. With this we have created the schematic for simulation.

5.3.5 Netlist generation

To simulate the circuit that has been 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. Check the option `Default Format`. Then click



Annotate Schematic

Scope

● Use the entire schematic

○ Use the current page only

● Keep existing annotation

○ Reset existing annotation

○ Reset, but do not swap any annotated multi-unit parts

Annotate only the unannotated components on the entire schematic?

This operation will change the current annotation and cannot be undone.

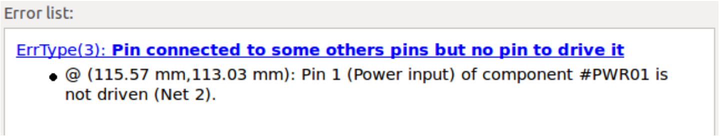
○ Start to sheet number*1000 and use first free number

Dialog

□ Keep this dialog open

☑ Always ask for confirmation

Figure 5.12: Steps in annotating a schematic: 1. First click on Annotation then 2. Click on Ok then 3. Click on close



Error list:

ErrType(3): Pin connected to some others pins but no pin to drive it

- @ (115.57 mm,113.03 mm): Pin 1 (Power input) of component #PWR01 is not driven (Net 2).

Figure 5.13: ERC error

on *Generate*. This is shown in Fig. 5.15. Save the netlist. This will be a `.cir` file. Do not change the directory while saving. Now the netlist is ready to be simulated. Refer to [?] or [?] to know more about Eeschema.

5.4 Tools for creating the PCB layout

The Eeschema top toolbar also has two important tools which can help the user to generate the PCB layout of the created schematic.

5.4.1 Foot Print 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, before using this tool, one should

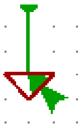
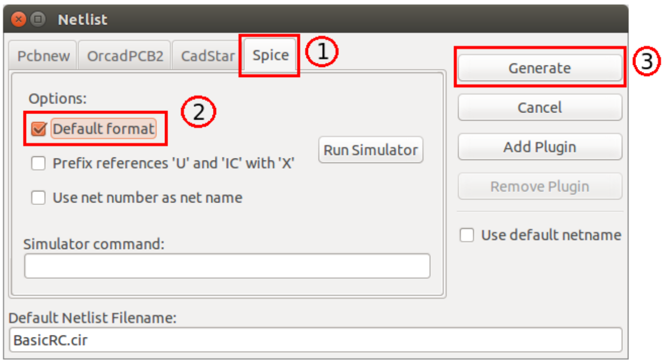


Figure 5.14: Green arrow pointing to Ground terminal indicating an ERC error



Netlist

Options:

☑ Default format

□ Prefix references 'U' and 'IC' with 'X'

□ Use net number as net name

□ Use default netname

Simulator command:

Default Netlist Filename:

BasicRC.cir

Figure 5.15: Steps in generating a Netlist for simulation: 1. Click on Spice then 2. Check the option `Default Format` then 3. Click on Generate

have the netlist for PCB design (a .net file). To know more about how to assign footprints to components, see Chapter 12.

5.4.2 PCB Layout

Clicking on the *Layout Editor* tool will open `Pcbnew`, the layout editor used in eSim. In this window, one will create the PCB. It involves laying tracks and vias, performing optimum routing of tracks, creating one or more copper layers for PCB, etc. It will be saved as a `.brd` file in the current project directory. Chapter 12 explains how to use the *Layout Editor* to design a PCB.

Chapter 6

Simulation

Circuit simulation uses mathematical models to replicate the behaviour of an actual device or circuit. Simulation software allows to model circuit operations. Simulating a circuit's behaviour before actually building it can greatly improve design efficiency. eSim uses `Ngspice` for analog, digital and mixed-level/mixed-signal circuit simulation. The various steps involved in simulating a circuit schematic in eSim are explained in the sections below:

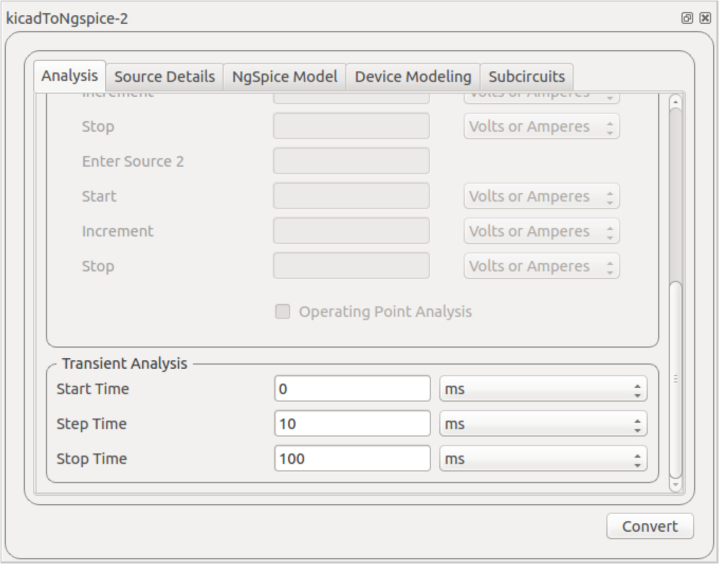
6.1 Kicad to Ngspice Conversion

In the chapter on schematic creation, we have learnt to generate the netlist from circuit schematic. The generated netlist is not compatible with Ngspice. eSim uses Ngspice to simulate the circuit schematic. Hence the netlist i.e. `.cir` file generated should be converted in to a Ngspice compatible file. The *Convert KiCad to Ngspice* tool on eSim left toolbar is used to do this. Let us now see the various tabs and their functions available under this.

6.1.1 Analysis

In order to simulate a circuit, the user must define the type of analysis to be done on the circuit. This tab is used to insert the type of analysis and value of the analysis parameters to the netlist. eSim supports three types of analyses: 1. *DC Analysis* (Operating Point and DC Sweep) 2. *AC Small-signal Analysis* 3. *Transient Analysis* These are explained below.

In the current example for simulating an RC circuit, select the analysis type as `transient` analysis and enter the values as shown in the Fig. 6.1.



Stop | Volts or Amperes

Enter Source 2

Start | Volts or Amperes

Increment | Volts or Amperes

Stop | Volts or Amperes

□ Operating Point Analysis

Transient Analysis

Start Time: 0 ms

Step Time: 10 ms

Stop Time: 100 ms

Figure 6.1: KiCad to Ngspice Window

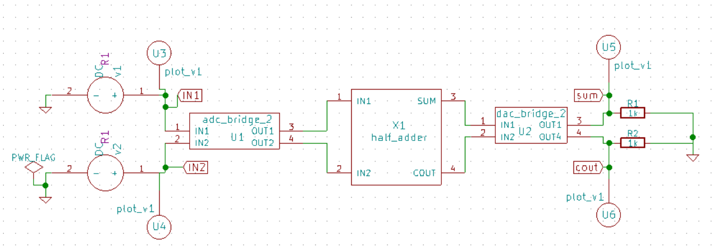


Figure 6.2: Half Adder Schematic

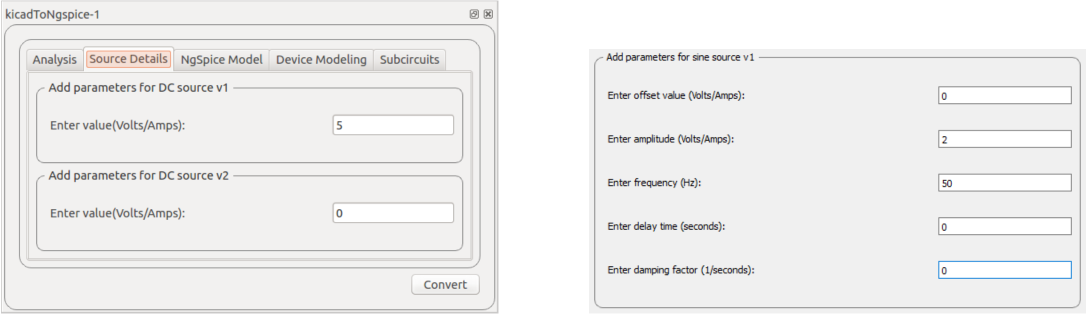
6.1.2 Source Details

The various parameter values of the sources added in the schematic can added using this tool. *Source details* is a dynamic tab, i.e. the fields are added as per the number of sources in the circuit. For example, consider a Half-Adder circuit as shown in Fig. 6.2 Here, we have used two DC input sources are used and hence the source detail GUI would be having two input fields as shown is Fig. 6.3a.

In the current example of the RC circuit, we have a single AC source. Fill in the details as shown in Fig. 6.3b.

6.1.3 Ngspice Model

The component libraries for components like DAC, ADC, transformer etc. which are used in the schematic are directly linked with the corresponding Ngspice models. The



Add parameters for DC source v1

Enter value(Volts/Amps): 5

Add parameters for DC source v2

Enter value(Volts/Amps): 0

Add parameters for sine source v1

Enter offset value (Volts/Amps): 0

Enter amplitude (Volts/Amps): 2

Enter frequency (Hz): 50

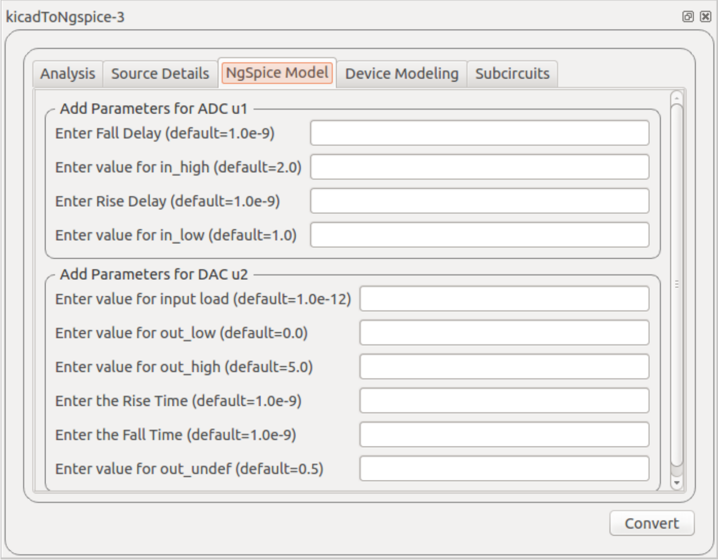
Enter delay time (seconds): 0

Enter damping factor (1/seconds): 0

(a) Source Details of Half-Adder

(b) Source Details of RC circuit

Figure 6.3: Source details interface



Add Parameters for ADC u1

Enter Fall Delay (default=1.0e-9)

Enter value for in_high (default=2.0)

Enter Rise Delay (default=1.0e-9)

Enter value for in_low (default=1.0)

Add Parameters for DAC u2

Enter value for input load (default=1.0e-12)

Enter value for out_low (default=0.0)

Enter value for out_high (default=5.0)

Enter the Rise Time (default=1.0e-9)

Enter the Fall Time (default=1.0e-9)

Enter value for out_undef (default=0.5)

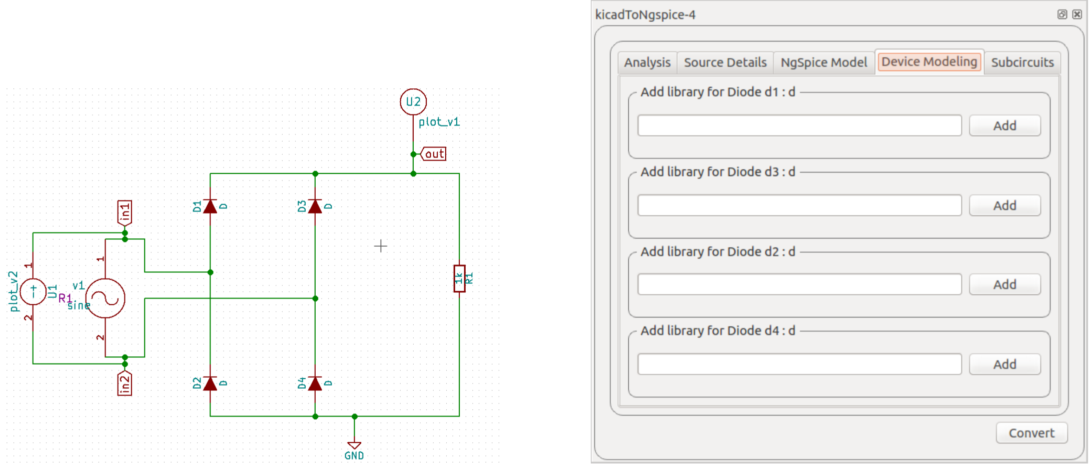
Figure 6.4: Half adder: Ngspice model

user can modify the parameter values using this tab, as shown in Fig. 6.4. If there are no modifications the default values are taken.

6.1.4 Device Modelling

Spice based simulators include a feature which allows accurate modeling of semiconductor devices such as diodes, transistors etc. Model libraries holds these features to define models for devices such as diodes, MOSFET, BJT, JFET, IGBT, Magnetic core etc.

The fields in this tab are added for each such device in the circuit and the corresponding model library is added. In the example of bridgerectifier as shown in Fig. 6.5a for four diodes library files are added as in Fig. 6.5b. If you do not add any library it will take default Ngspice model for diode.



(a) Schematic

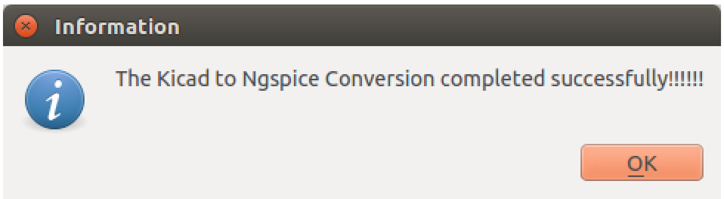
(b) Adding device model

Figure 6.5: Bridge Rectifier

6.1.5 Sub Circuit

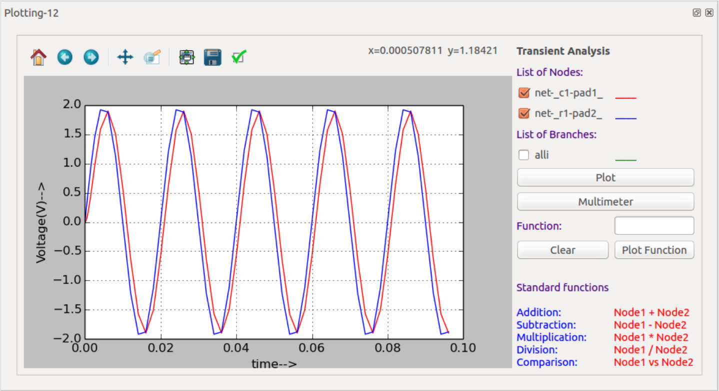
Subcircuit is a way to implement hierarchical modeling. Once a subcircuit for a component is created, it can be used in other circuits.

After Filling up the values in all the above mentioned fields the convert button is pressed. The Ngspice netlist, `.cir.out` file is generated. A message box pops up, as shown in Fig. 6.6. Click on `OK`.



The Kicad to Ngspice Conversion completed successfully!!!!!!

Figure 6.6: Message after successful Ngspice netlist generation



☑ net-_c1-pad1_
☑ net-_r1-pad2_
□ alli

Addition:	Node1 + Node2
Subtraction:	Node1 - Node2
Multiplication:	Node1 * Node2
Division:	Node1 / Node2
Comparison:	Node1 vs Node2

Figure 6.7: Pythonplot for RC circuit

6.2 Simulating the schematic

6.2.1 Simulation

Once the Kicad to Ngspice conversion is successfully completed, press the *Simulation* button on the leftside toolbar on eSim interface. This will display the Python plot window along with the node names. Select the required nodes and click on `Plot`. The simulations are displayed. In the present example of the RC circuit, the plot will be displayed as shown in Fig. 6.7.

Pressing the *Simulation* button also opens up the Ngspice terminal and plot windows. The Ngspice plots for all the nodes (where we have used the plot components in the schematic) will be displayed. In the current example, we have used two plot components and the Ngspice simulations for these two nodes are displayed as in Fig. 6.8.

If the *Plot components* are not used in the schematic, the simulations are not displayed automatically. To see the Ngspice simulations, type the following commands in the Ngspice terminal window.

- `plot allv` - Plots all the voltage waveforms.
- `plot v(node-name)` - Plot a waveform of the node-name voltage source.
- `plot alli` - Plots all the current waveforms.

6.2.2 Multimeter

Multimeter is another feature that is available in eSim. Using this facility the user can view the voltage and current values in various nodes and branches respectively. To use the multimeter select the required nodes from the plot window and press `Multimeter` button, shown in Fig. 6.9. Windows equal to the number of selected nodes will open.

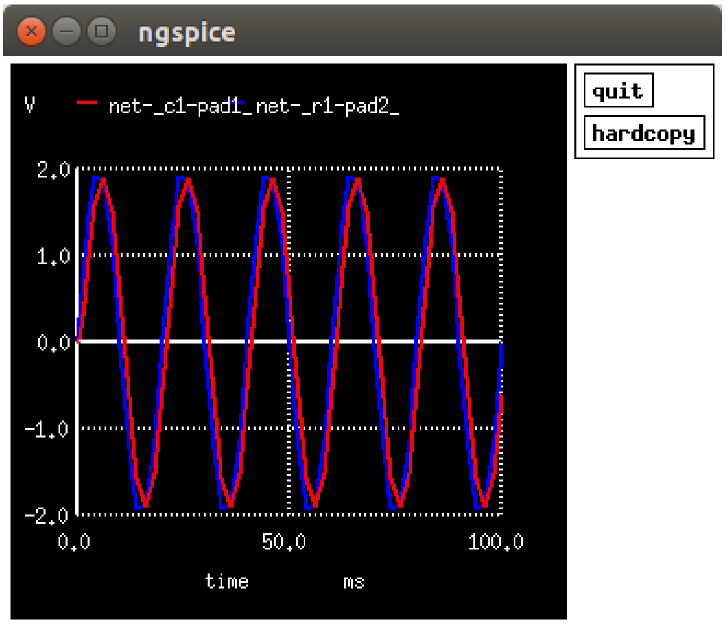


Figure 6.8: Ngspice voltage simulation for RC circuit

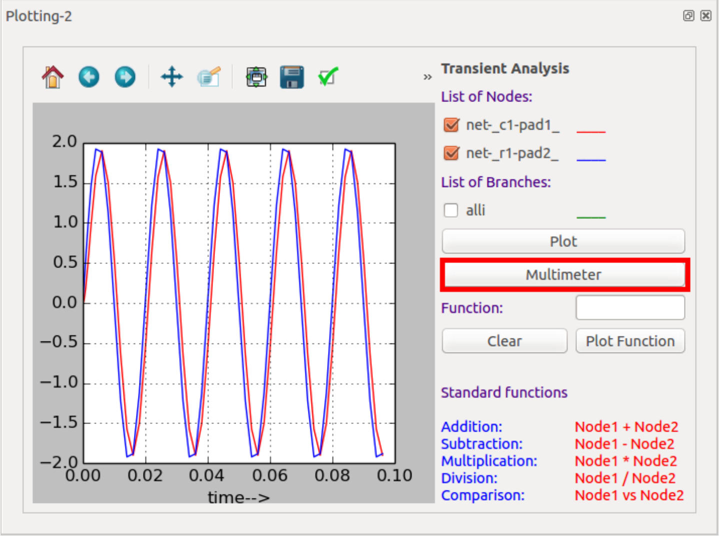


Figure 6.9: Multimeter feature in eSim

Now open the schematic window and place these pop up windows near the appropriate nodes on the schematic to get the voltage of each node. Similarly current through each branches in the schematic can also be found using the multimeter facility.

Chapter 7

Model Editor

Spice based simulators include a feature which allows accurate modeling of semiconductor devices such as diodes, transistors etc. eSim Model Editor provides a facility to define a new model for devices such as *diodes, MOSFET, BJT, JFET, IGBT, Magnetic core* etc. Model Editor in eSim lets the user enter the values of parameters depending on the type of device for which a model is required. The parameter values can be obtained from the data-sheet 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 Editor also provides a facility to edit existing models. The GUI of the model editor is as shown in Fig. 7.1

7.1 Creating New Model Library

eSim lets us create new model libraries based on the template model libraries. On selecting `New` button the window is popped as shown in Fig. 7.2. The name has to be

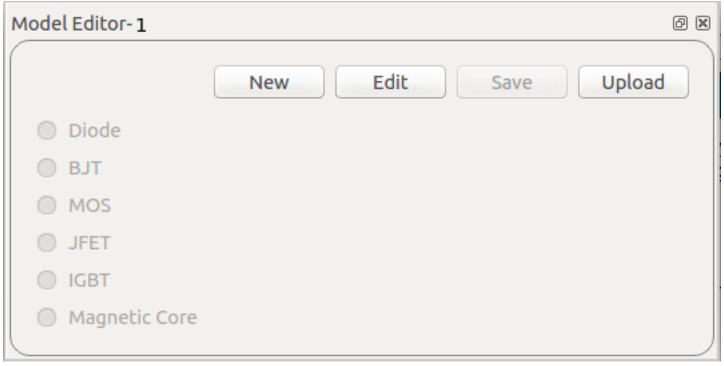
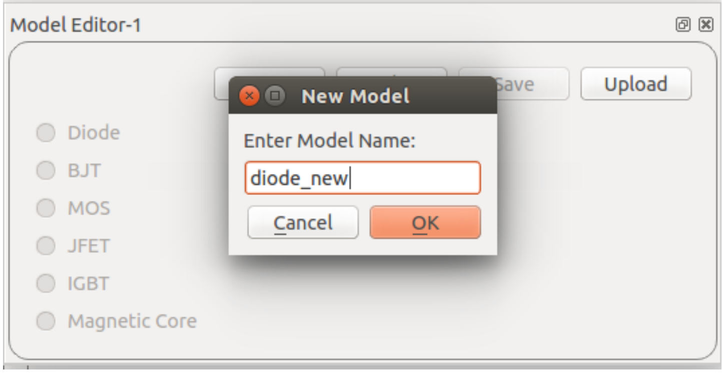
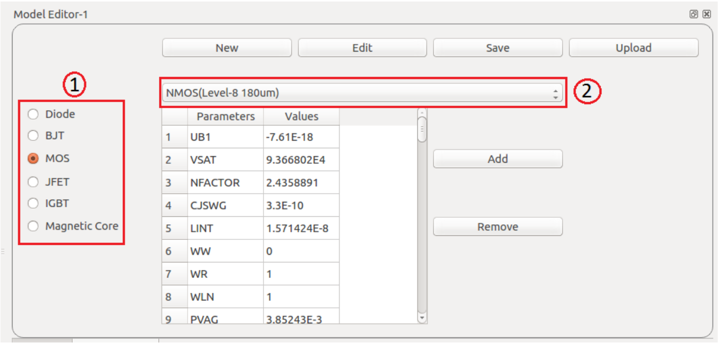


Figure 7.1: Model Editor



Enter Model Name: diode_new

Figure 7.2: Creating New Model Library



○ Diode
○ BJT
● MOS
○ JFET
○ IGBT
○ Magnetic Core

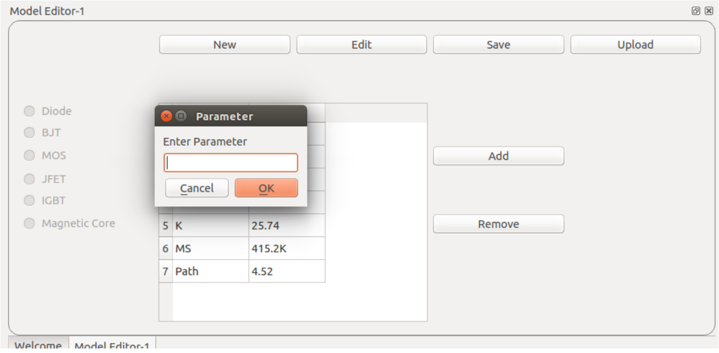
NMOS(Level-8 180um)

	Parameters	Values
1	UB1	-7.61E-18
2	VSAT	9.366802E4
3	NFACTOR	2.4358891
4	CJSWG	3.3E-10
5	LINT	1.571424E-8
6	WW	0
7	WR	1
8	WLN	1
9	PVAG	3.85243E-3

Figure 7.3: Choosing the Template Model Library

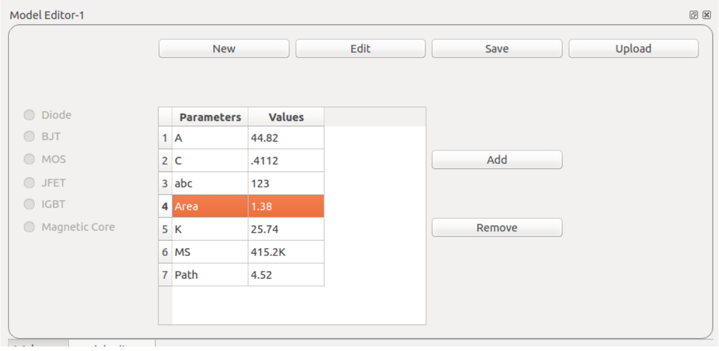
unique otherwise the error message appears on the window.

After the OK button is pressed the type of model library to be created is chosen by selecting one of the types on the left hand side i.e. `Diode`, `BJT`, `MOS`, `JFET`, `IGBT`, `Magnetic Core`. The template model library opens up in a tabular form as shown in Fig. 7.3



5	K	25.74
6	MS	415.2K
7	Path	4.52

Figure 7.4: Adding the Parameter in a Library



	Parameters	Values
1	A	44.82
2	C	.4112
3	abc	123
4	Area	1.38
5	K	25.74
6	MS	415.2K
7	Path	4.52

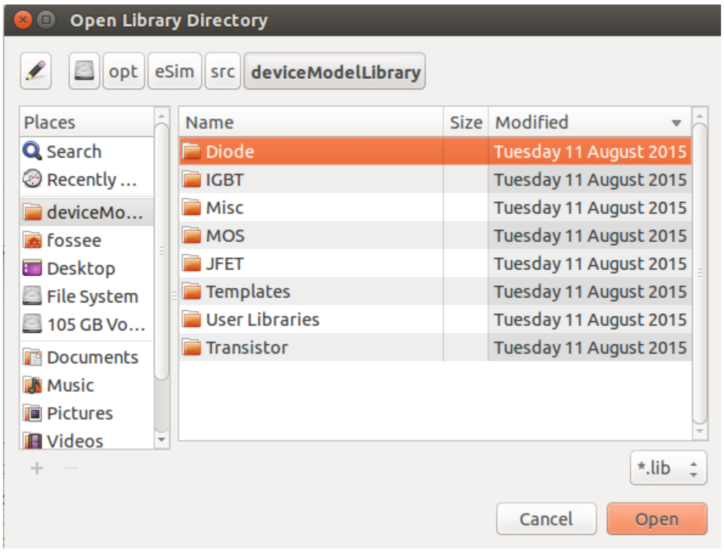
Figure 7.5: Removing a Parameter from a Library

New parameters can be added or current parameters can be removed using ADD and REMOVE buttons. Also the values of parameters can be changed in the table. Adding and removing the parameters in library files is shown in the Fig. 7.4 and Fig. 7.5

After the editing of the model library is done, the file can be saved by selecting the SAVE button. These libraries are saved in the *User Libraries* folder under *deviceModelLibrary* repository.

7.2 Editing Current Model Library

The existing model library can be modified using EDIT option. On clicking the EDIT button the file dialog opens where all the library files are saved as shown in Fig. 7.6. You can select the library you want to edit. Once you are done with the editing, click on SAVE button.



Name	Size	Modified
Diode		Tuesday 11 August 2015
IGBT		Tuesday 11 August 2015
Misc		Tuesday 11 August 2015
MOS		Tuesday 11 August 2015
JFET		Tuesday 11 August 2015
Templates		Tuesday 11 August 2015
User Libraries		Tuesday 11 August 2015
Transistor		Tuesday 11 August 2015

Figure 7.6: Editing Existing Model Library

7.3 Uploading external .lib file to eSim repository

eSim directly cannot use the external .lib file. It has to be uploaded to eSim repository before using it in a circuit. eSim provides the facility to upload library files. They are then converted into xml format, which can be easily modified from the eSim interface. On clicking `UPLOAD` button the library can be uploaded from any location. The model library will be saved with the name you have provided, in the *User Libraries* folder of repository *deviceModelLibrary*.

Chapter 8

SubCircuit Builder

Subcircuit is a way to implement hierarchical modeling. Once a subcircuit for a component is created, it can be used in other circuits. eSim provides an easy way to create a subcircuit. The following Fig. 8.1 shows the window that is opened when the SubCircuit tool is chosen from the toolbar.

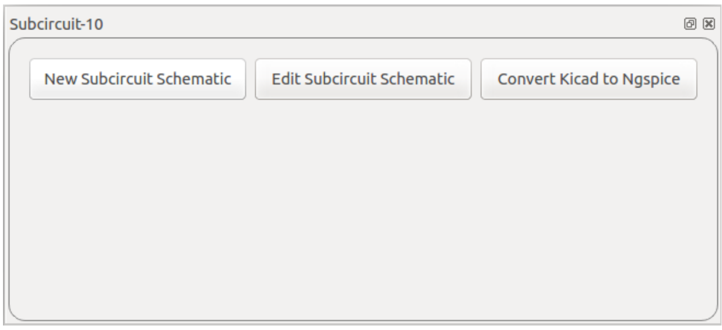


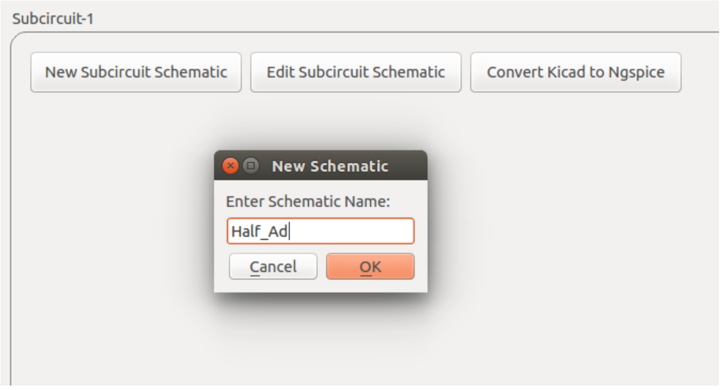
Figure 8.1: Subcircuit Window

8.1 Creating a SubCircuit

The steps to create subcircuit are as follows.

- After opening the Subcircuit tool, click on `New Subcircuit Schematic` button. It will ask the name of the subcircuit. Enter the name of subcircuit (without any spaces) and click `OK` as shown in Fig. 8.2.

- After clicking `OK` button it will open KiCad schematic. Draw your circuit which will be later used as a subcircuit. e.g the Fig. 8.3 shows the half adder circuit.



Subcircuit-1

Enter Schematic Name:

Half_Ad

Figure 8.2: New Sub circuit Window

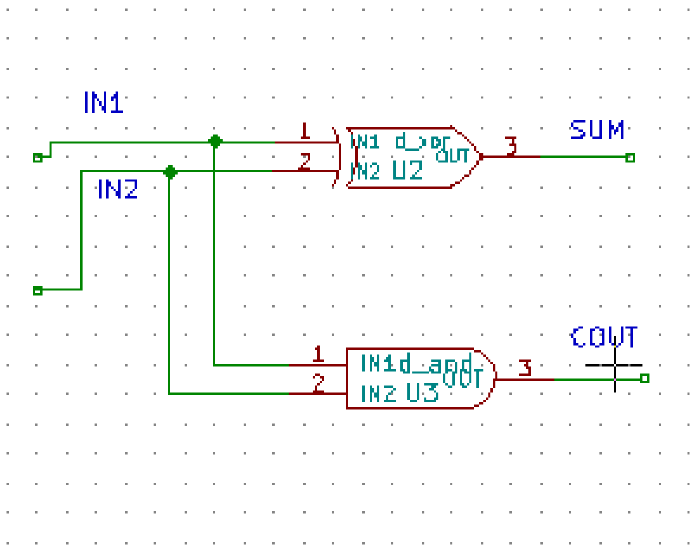


Figure 8.3: New Sub circuit Window

- Once you complete the circuit, assign port to the node of your circuit which will be used to connect with the main circuit. The circuit will look like Fig. 8.4 after adding PORT to it. The PORT symbol can be found in Eeschema as shown in Fig. 8.5.
- Next step is to save the schematic and generate KiCad netlist as explained in Chapter 5.
- To use this as a subcircuit, create a block in KiCad Eeschema by following steps

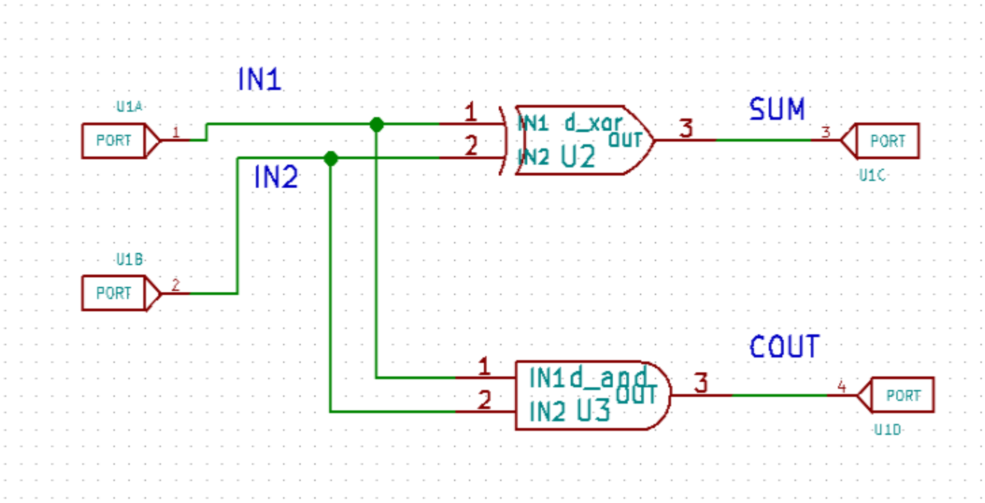


Figure 8.4: Half-Adder Subcircuit

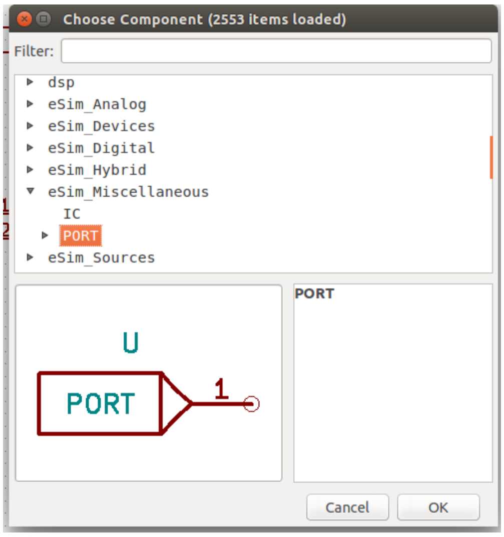
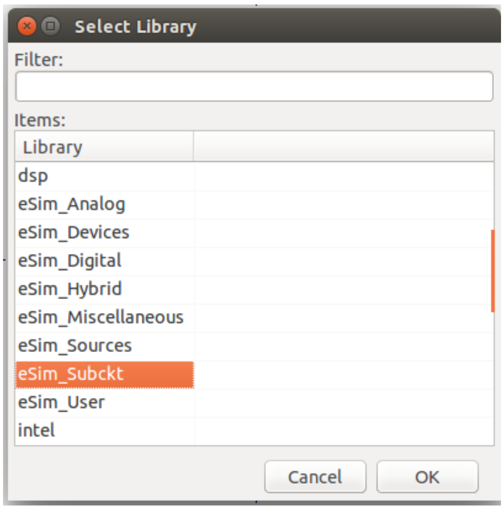


Figure 8.5: Selection of PORT component

given below:

1. Go to library browser of Eeschema.
2. Select the working library as eSim_Subckt as shown in Fig. 8.6



Select Library

Filter:

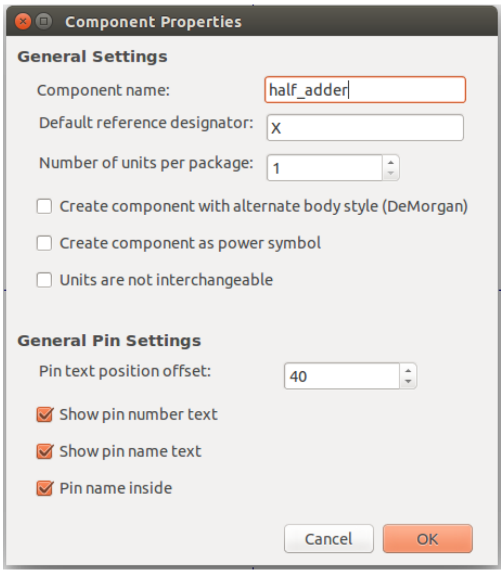
Items:

Library

dsp
eSim_Analog
eSim_Devices
eSim_Digital
eSim_Hybrid
eSim_Miscellaneous
eSim_Sources
eSim_Subckt
eSim_User
intel

Figure 8.6: Selecting Working Library

3. Click on create a new component with reference X as shown in Fig. 8.7



Component Properties

General Settings

Component name: half_adder

Default reference designator: X

Number of units per package: 1

□ Create component with alternate body style (DeMorgan)

□ Create component as power symbol

□ Units are not interchangeable

General Pin Settings

Pin text position offset: 40

☑ Show pin number text

☑ Show pin name text

☑ Pin name inside

Figure 8.7: Creating New Component

4. Start drawing the subcircuit block. Update and save it as shown in Fig. 8.8.

- Close the Eeschema window and click on Convert KiCad to Ngspice button in subcircuit builder tool. This will convert the KiCad spice netlist to Ngspice netlist.

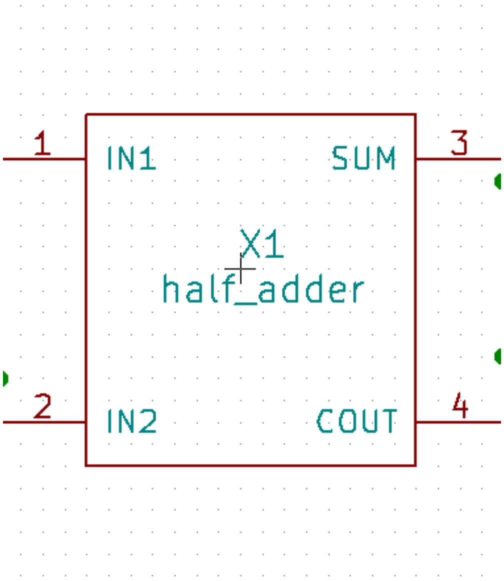


Figure 8.8: Half-Adder Subcircuit Block

And it will save your subcircuit into eSim repository, which you can add in your main circuit.

8.2 Edit a Subcircuit

The steps to edit a subcircuit are as follows.

- After opening the Subcircuit tool, click on `Edit Subcircuit Schematic` button. It will open a dialog box where you can select any subcircuit for editing.
- After selecting the subcircuit it will open it in KiCad Eeschema, where you can edit the subcircuit.
- Next step is to save the schematic and generate KiCad netlist.
- If you have edited the number of ports then you have to change the block in KiCad Eeschema accordingly.
- Close the Eeschema window and click on `Convert KiCad to Ngspice` button in subcircuit builder tool to convert the edited subcircuit KiCad netlist into Ngspice netlist.

Chapter 9

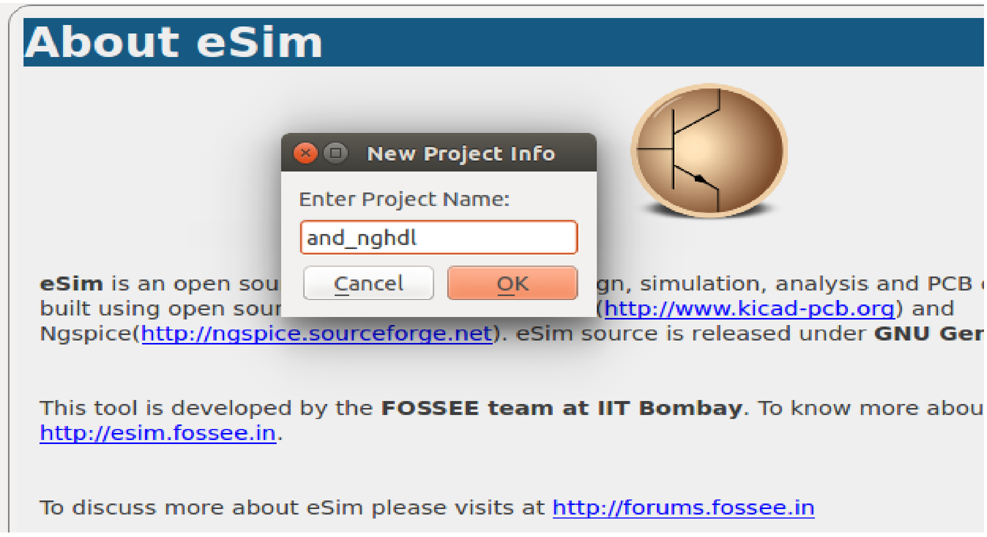
NGHDL-Mixed mode simulation

NGHDL is an add on to eSim for mixed mode circuit simulation, It uses ghdl for digital simulation and the mixed mode simulation happens through Ngspice, it takes digital values from ghdl using socket programming.

9.1 Digital Model creation using NGHDL

The steps to create digital models are given below:

1. Create a new project as shown in figure Fig. 9.1



About eSim

New Project Info

Enter Project Name: and_nghdl

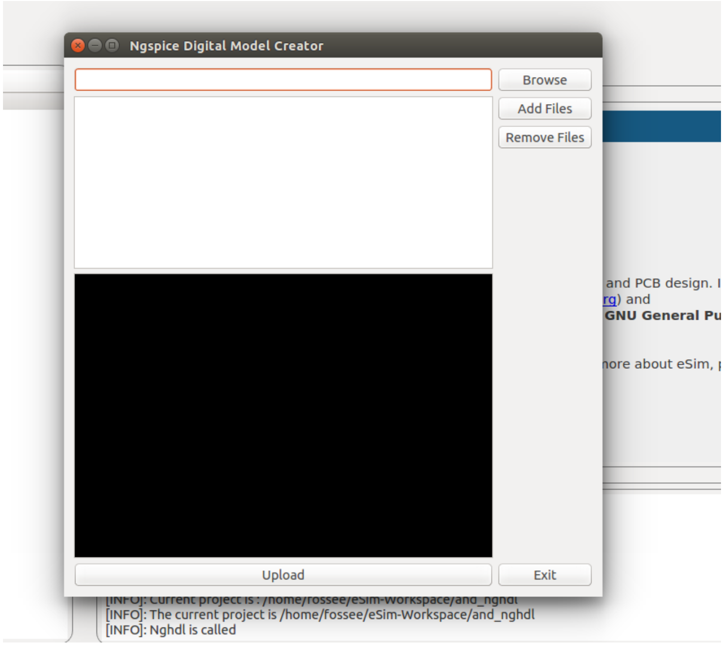
eSim is an open sou gn, simulation, analysis and PCB
built using open sour (http://www.kicad-pcb.org) and
Ngspice(http://ngspice.sourceforge.net). eSim source is released under **GNU Ger**

This tool is developed by the **FOSSEE team at IIT Bombay**. To know more abou
http://esim.fossee.in.

To discuss more about eSim please visits at http://forums.fossee.in

Figure 9.1: Creation of a new project

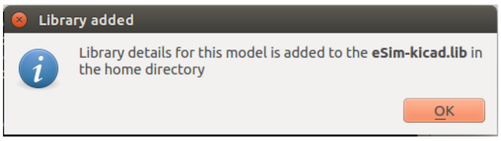
2. Click on NGHDL button on left side pane of main window, a window will appear as shown in figure Fig. 9.2



```
[INFO]: Current project is : /home/fossee/eSim-Workspace/and_nghdl
[INFO]: The current project is /home/fossee/eSim-Workspace/and_nghdl
[INFO]: Nghdl is called
```

Figure 9.2: NGHDL interface

3. Now browse and locate the VHDL code to upload. Select and click upload as shown in figure Fig. 9.3. This process will create Ngspice and KiCad model of digital block.



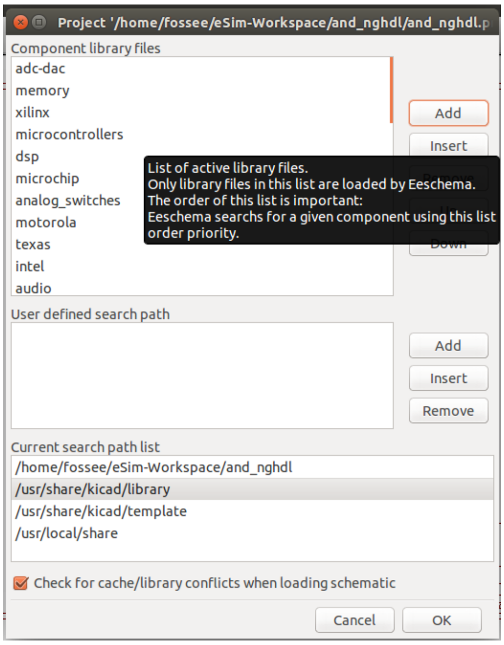
Library details for this model is added to the **eSim-kicad.lib** in the home directory

Figure 9.3: Uploading of digital model

9.2 Schematic Creation

Steps for schematic creation are as follows:

1. After uploading the model, you can create the schematic of your design by clicking on *Open Schematic* button on the left pane of the eSim window. Then go to *Preferences* option on top of the window and click on *Component Libraries* to add the KiCad model created in the previous step. Following window will appear as shown in Fig. 9.4, where you will have to click on *Add* button and select the *esim-kicad* library by browsing to the home folder. Refer Fig. 9.5.



Project '/home/fossee/eSim-Workspace/and_nghdl/and_nghdl.p

Component library files

adc-dac
memory
xilinx
microcontrollers
dsp
microchip
analog_switches
motorola
texas
intel
audio

List of active library files.
Only library files in this list are loaded by Eeschema.
The order of this list is important:
Eeschema searchs for a given component using this list order priority.

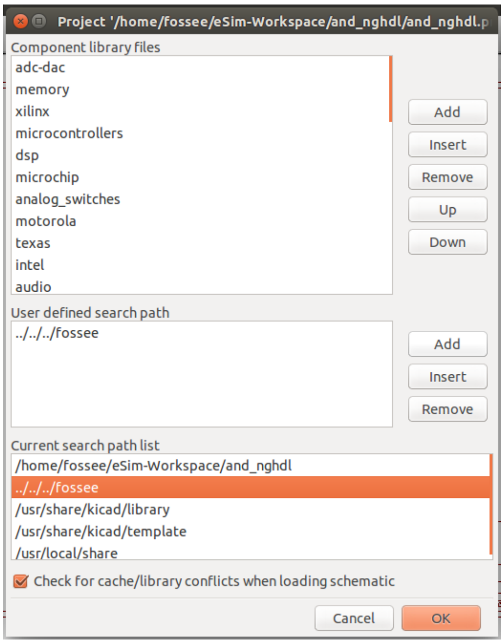
User defined search path

Current search path list

/home/fossee/eSim-Workspace/and_nghdl
/usr/share/kicad/library
/usr/share/kicad/template
/usr/local/share

☑ Check for cache/library conflicts when loading schematic

Figure 9.4: Importing the digital model in KiCad



Project '/home/fossee/eSim-Workspace/and_nghdl/and_nghdl.p

Component library files

adc-dac
memory
xilinx
microcontrollers
dsp
microchip
analog_switches
motorola
texas
intel
audio

User defined search path

../../../fossee

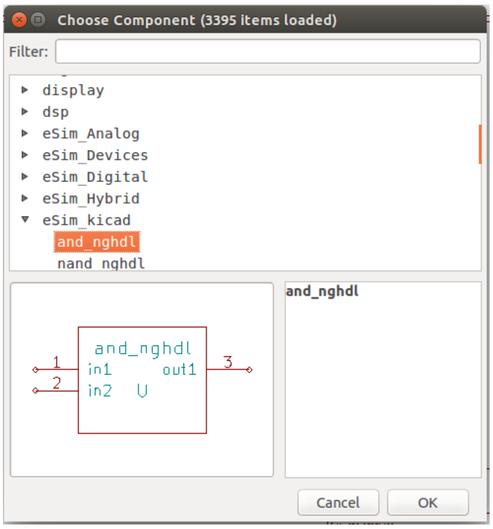
Current search path list

/home/fossee/eSim-Workspace/and_nghdl
../../../fossee
/usr/share/kicad/library
/usr/share/kicad/template
/usr/local/share

☑ Check for cache/library conflicts when loading schematic

Figure 9.5: Selection of library

2. Next step is to locate the component in *kicad-esim* library as shown in figure Fig. 9.6 and place it on the schematic editor as shown in Fig. 9.7.



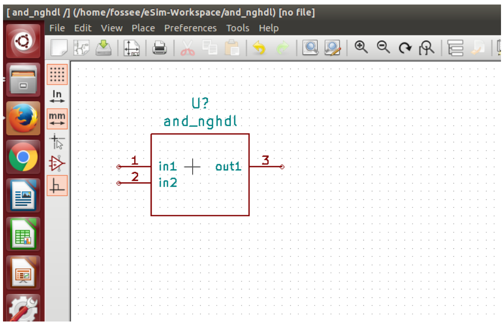
Choose Component (3395 items loaded)

Filter:

▸ display
▸ dsp
▸ eSim_Analog
▸ eSim_Devices
▸ eSim_Digital
▸ eSim_Hybrid
▾ eSim_kicad
and_nghdl
nand_nghdl

and_nghdl

Figure 9.6: Locating the component in library



[and_nghdl /] (/home/fossee/eSim-Workspace/and_nghdl) [no file]
File Edit View Place Preferences Tools Help
In
mm
U?
and_nghdl
1
2
in1
in2
out1
3

Figure 9.7: Placement of component on editor

3. Now create the schematic as shown in figure Fig. 9.8, annotate, perform ERC, create the netlist and save the schematic by following the steps given in Chapter 5.

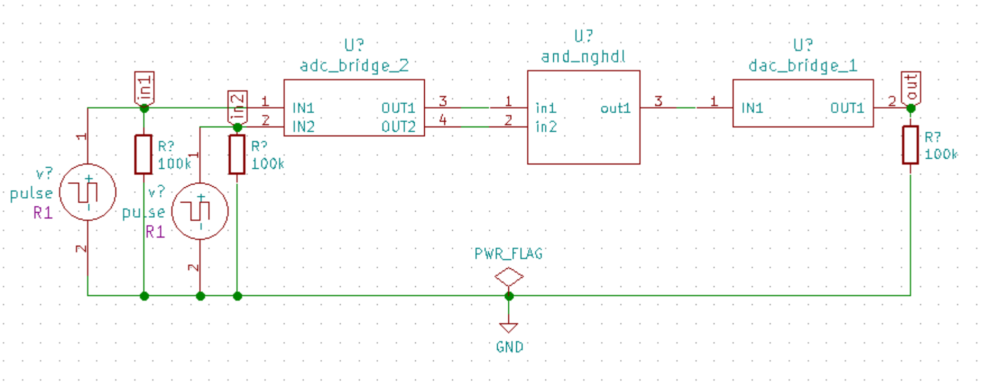
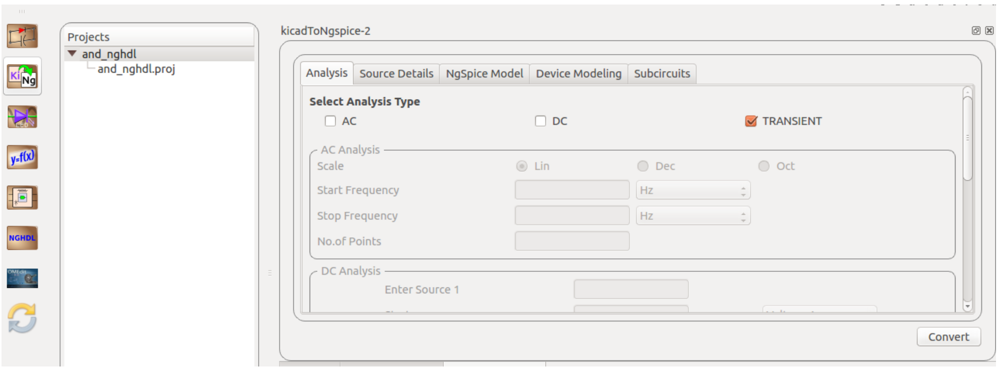


Figure 9.8: Mixed mode circuit

4. After creating the schematic, click on *KiCad to Ngspice converter* and select the type of analysis as transient as shown in figure Fig. 9.9 and set the start, step and stop time as shown in figure Fig. 9.9



Select Analysis Type

□ AC □ DC ☑ TRANSIENT

AC Analysis

Scale ○ Lin ○ Dec ○ Oct

Start Frequency Hz

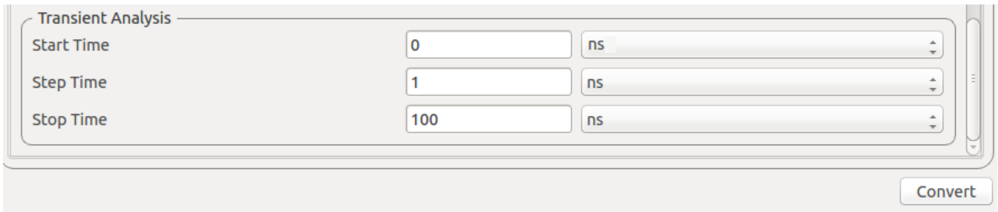
Stop Frequency Hz

No.of Points

DC Analysis

Enter Source 1

Figure 9.9: Analysis Part I

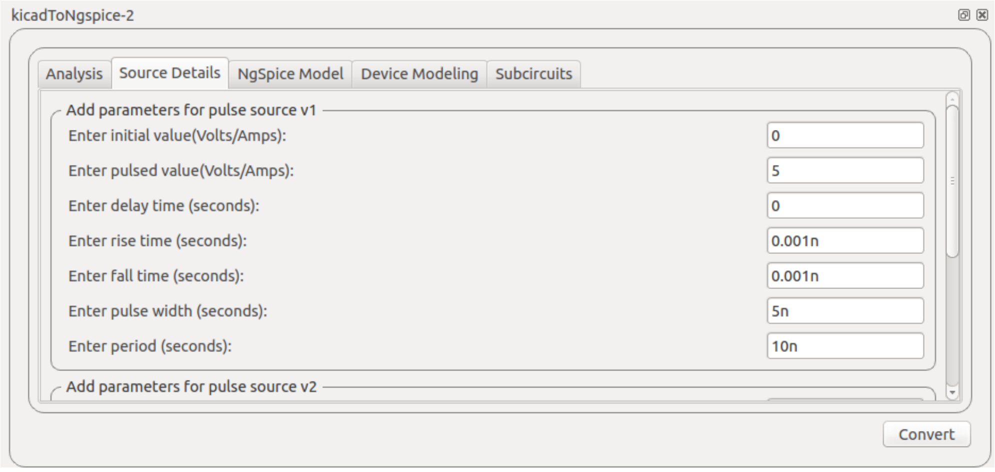


Transient Analysis

Start Time	0	ns
Step Time	1	ns
Stop Time	100	ns

Figure 9.10: Analysis Part II

5. Now click on *Source Details* and enter the values for Source v1 and source v2 as shown in figure Fig. 9.11 and Fig. 9.12



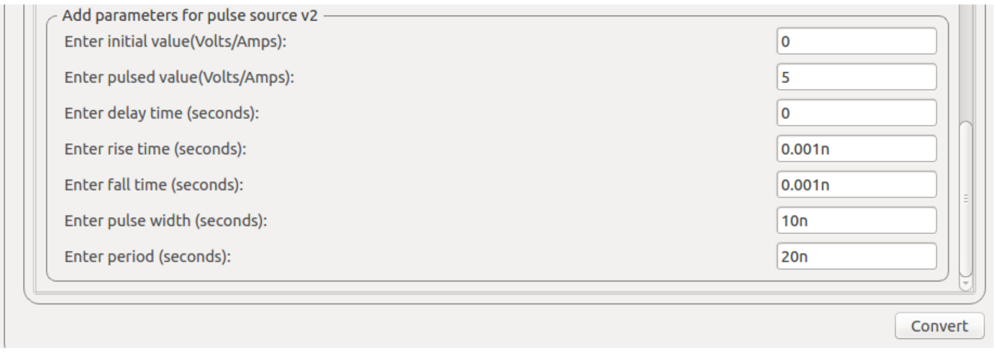
Add parameters for pulse source v1

Enter initial value(Volts/Amps):	0
Enter pulsed value(Volts/Amps):	5
Enter delay time (seconds):	0
Enter rise time (seconds):	0.001n
Enter fall time (seconds):	0.001n
Enter pulse width (seconds):	5n
Enter period (seconds):	10n

Add parameters for pulse source v2

Figure 9.11: Value of Source v1

6. Now select the option Ngspice Model and enter the instance id and leave all



Add parameters for pulse source v2

Enter initial value(Volts/Amps): 0

Enter pulsed value(Volts/Amps): 5

Enter delay time (seconds): 0

Enter rise time (seconds): 0.001n

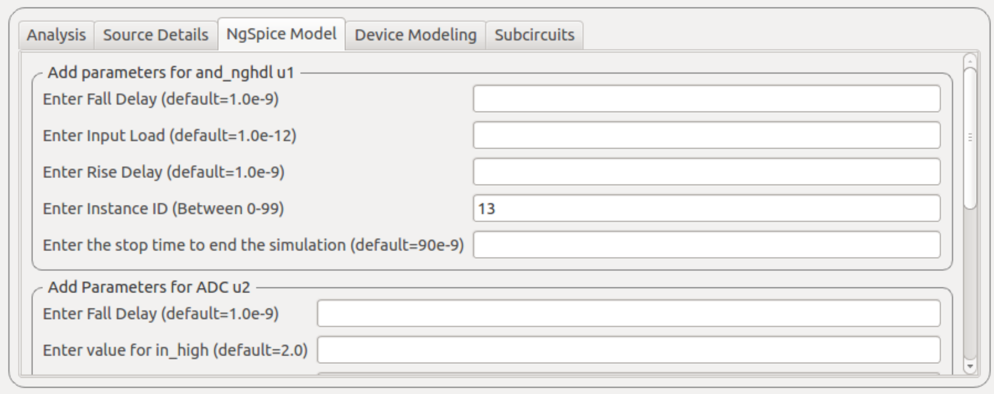
Enter fall time (seconds): 0.001n

Enter pulse width (seconds): 10n

Enter period (seconds): 20n

Figure 9.12: Value of Source v2

other options blank as shown in figure Fig. 9.13. The values of other blocks will be default value. After that click on *convert* button. This step will create the NgSpice netlist.



Add parameters for and_nghdl u1

Enter Fall Delay (default=1.0e-9):

Enter Input Load (default=1.0e-12):

Enter Rise Delay (default=1.0e-9):

Enter Instance ID (Between 0-99): 13

Enter the stop time to end the simulation (default=90e-9):

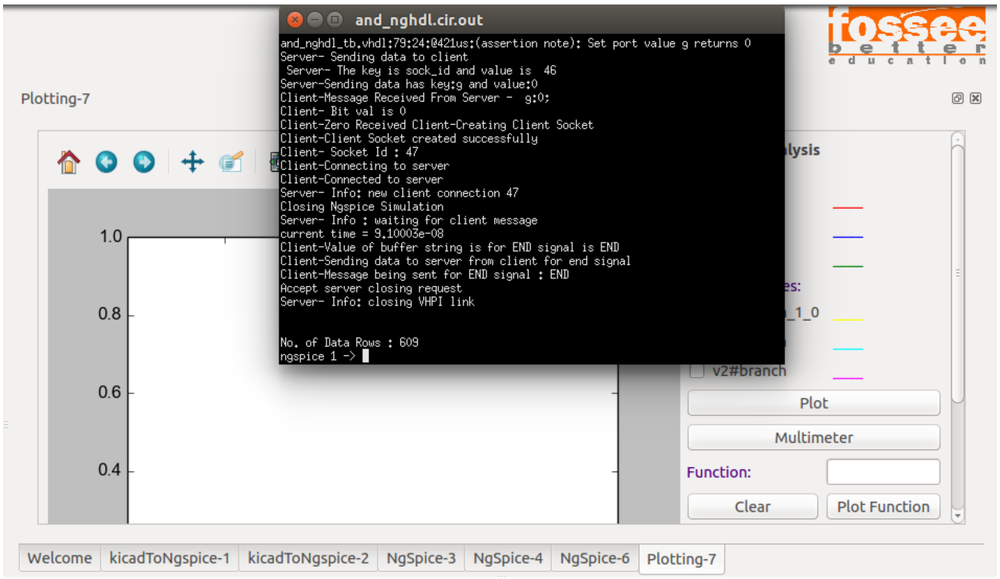
Add Parameters for ADC u2

Enter Fall Delay (default=1.0e-9):

Enter value for in_high (default=2.0):

Figure 9.13: Model Parameters

7. Now click on *Simulation* button, it will display the following windows as shown in figure Fig. 9.14. These are the Ngspice terminal and Python plot window.



```
and_nghdl_tb.vhdl:79:24:@421us:(assertion note): Set port value g returns 0
Server- Sending data to client
 Server- The key is sock_id and value is  46
Server-Sending data has key:g and value:0
Client-Message Received From Server -  g:0:
Client- Bit val is 0
Client-Zero Received Client-Creating Client Socket
Client-Client Socket created successfully
Client- Socket Id : 47
Client-Connecting to server
Client-Connected to server
Server- Info: new client connection 47
Closing Ngspice Simulation
Server- Info : waiting for client message
current time = 9.10003e-08
Client-Value of buffer string is for END signal is END
Client-Sending data to server from client for end signal
Client-Message being sent for END signal : END
Accept server closing request
Server- Info: closing VHPI link


No. of Data Rows : 609
ngspice 1 ->
```

Figure 9.14: Simulation window

8. Now select the required nodes and clicl on *Plot* button. You can see the plots of input source v1, input source v2 and output as shown in Fig. 9.15, Fig. 9.16, and Fig. 9.17 respectively.

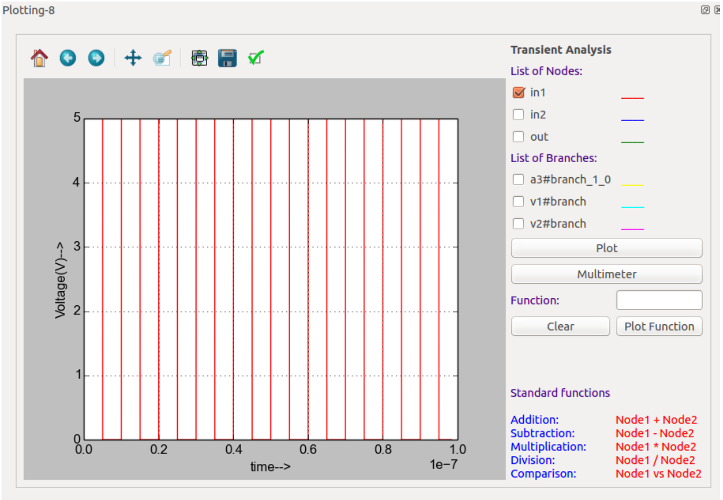


Figure 9.15: Plot of Source V1

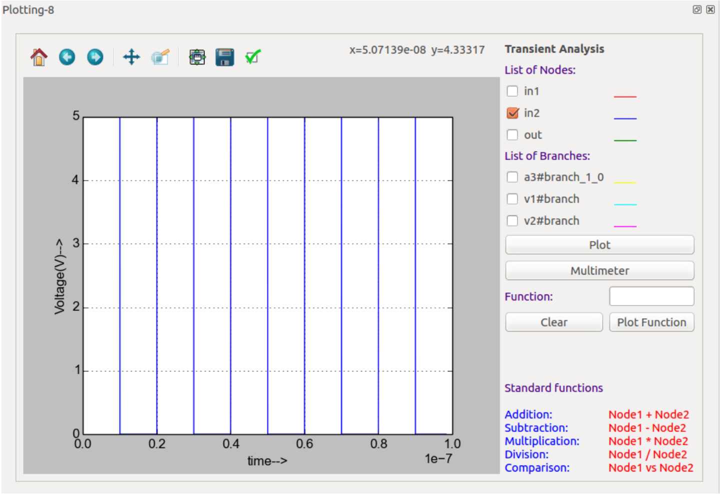


Figure 9.16: Plot of source V2

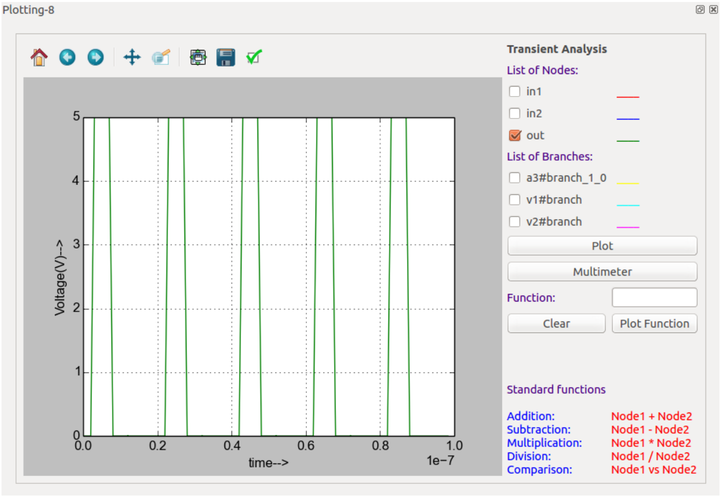


Figure 9.17: Plot of output

Chapter 10

OpenModelica

10.1 Introduction

OpenModelica (OM) is an open source modeling and simulation tool based on Modelica language. Modelica is an object oriented language. As a result, it has all the features of an object oriented language such as inheritance. Models or circuits are defined in the form of classes, with in which there are components, functions, connection and placement information. The OM suite has the following major tools.

10.1.1 OMEdit

An IDE for modeling and simulation. It supports a lot of electrical components. It has a good graphical interface to drag and drop components and create the circuit. One can only do transient simulation using this interface. An attractive feature of OMEdit is the plotting interface. All the parameters in the circuit like voltages and currents through each component, parameters like frequency, delay etc. will be displayed as a list, after simulation. The user can choose the variables to be plotted in an interactive manner from this list. On choosing the variable to plot, it will be plotted on the plot window. One can also create multiple plot windows.

10.1.2 OMOptim

An IDE for optimisation. It lists all the variables in the given model. One can choose the variables to be optimised from the list. Multiple models can be loaded for a given optimisation problem. One can do multi objective optimisations as well. It supports various optimisation algorithms such as Particle Swarm Optimisation (PSO) and Simulated Annealing (SA). The results are displayed graphically.

10.2 OpenModelica in eSim

The above two functionalities can be accessed through the `Modelica Converter` and `OM Optimisation` tools on the eSim left toolbar. The two examples given below illustrates how to use OpenModelica in eSim.

Low Pass Filter circuit

Let us now see how to simulate a low pass filter in OpenModelica.

1. Open the schematic and create the circuit as shown in Fig. 10.1.

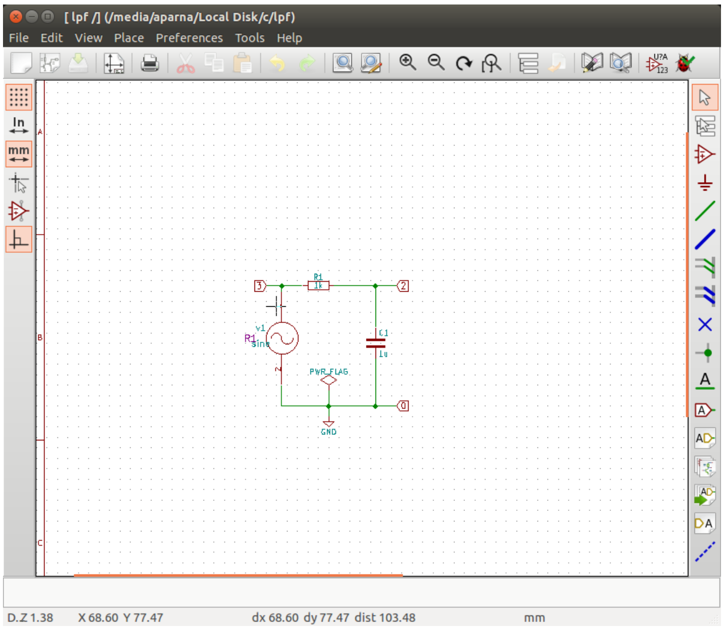

[lpf /] (/media/aparna/Local Disk/c/lpf)
File Edit View Place Preferences Tools Help
R1
v1
sine
C1
PWR_FLAG
GND
D.Z 1.38 X 68.60 Y 77.47 dx 68.60 dy 77.47 dist 103.48 mm

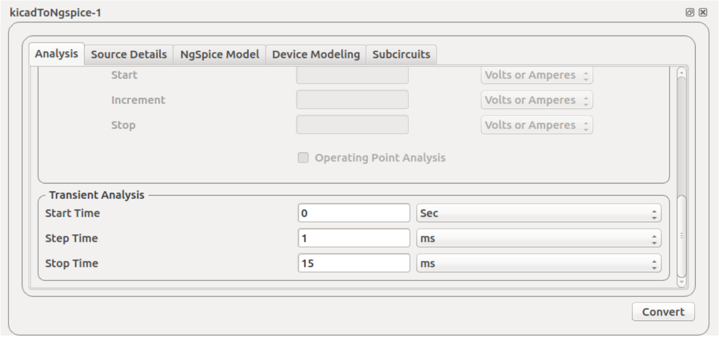
Figure 10.1: Circuit schematic: Low pass filter

2. Create the KiCad netlist. Now the analysis and analysis parameters are given as shown in Fig. 10.2.

3. The source details are given as in Fig. 10.3. The generated KiCad netlist is then converted to ngspice compatible netlist.

4. Simulate the ngspice netlist. The simulation curves are shown in Fig. 10.4.

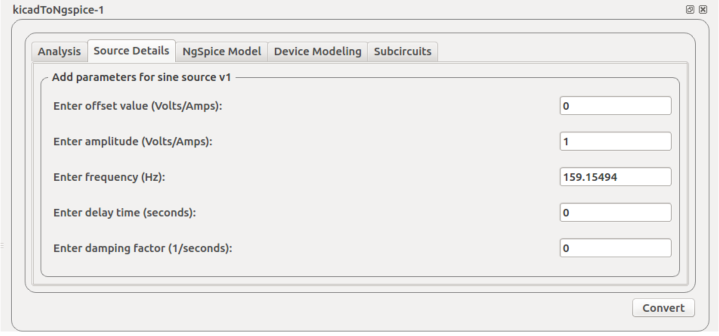
5. Now to use OpenModelica, click on `Modelica Converter` in the bottom left of eSim left toolbar.*M*ake sure you have OpenModelica installed in the system. This converter converts the spice netlist to Modelica format. Click on the LPF in the



Start Volts or Amperes
Increment Volts or Amperes
Stop Volts or Amperes
□ Operating Point Analysis

Transient Analysis
Start Time 0 Sec
Step Time 1 ms
Stop Time 15 ms

Figure 10.2: Analysis parameters: Low pass filter



Add parameters for sine source v1
Enter offset value (Volts/Amps): 0
Enter amplitude (Volts/Amps): 1
Enter frequency (Hz): 159.15494
Enter delay time (seconds): 0
Enter damping factor (1/seconds): 0

Figure 10.3: Source details: Low pass filter

left that is appended in OpenModelica main window. Make sure you are in text view to see the Modelica code as shown in Fig. 10.5 Figure shows that LPF circuit is being used as a model, the initialisation of sources and components are in the beginning followed by the connection information. n3, n0,n2 are the nodes.

Default Modelica libary is used for electrical sources and components. This has to imported so that it can be used in the current circuit. This is available in the left side of main window.

6. Click on Simulation Setup on the toolbar at the top. A window opens as shown in Fig. 10.6. Give start and stop time. Click `OK`.

7. A plotting window opens. Click on the node at the right to display the waveform. The window is shown in Fig. 10.7.

10.2.1 OM Optimisation

Now let us explore how to use OpenModelica for optimisation through an example. Find the value of resistance R2 that maximises the power dissipated through it for the

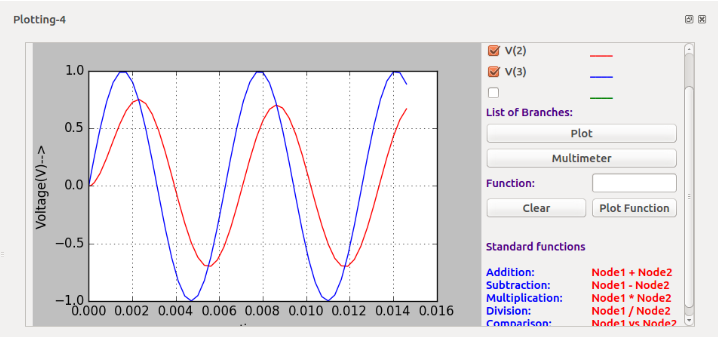
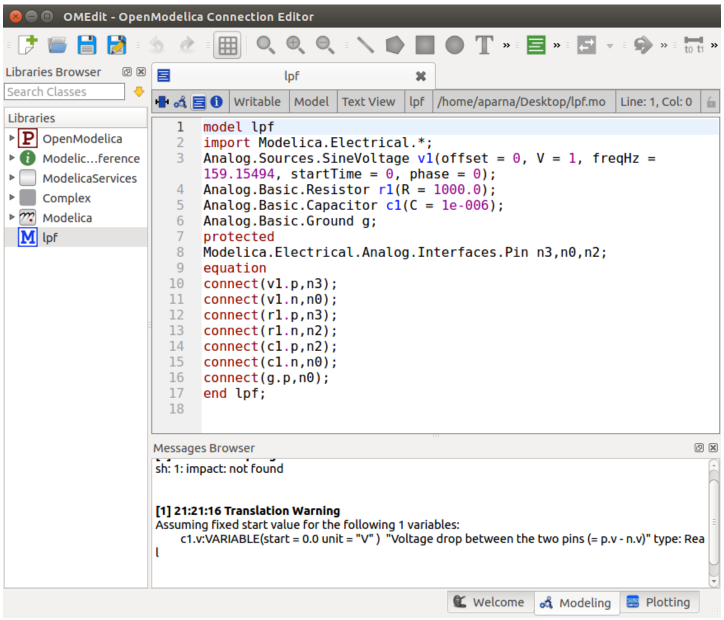


Figure 10.4: Simulation: Low pass filter



```
model lpf
import Modelica.Electrical.*;
Analog.Sources.SineVoltage v1(offset = 0, V = 1, freqHz =
159.15494, startTime = 0, phase = 0);
Analog.Basic.Resistor r1(R = 1000.0);
Analog.Basic.Capacitor c1(C = 1e-006);
Analog.Basic.Ground g;
protected
Modelica.Electrical.Analog.Interfaces.Pin n3,n0,n2;
equation
connect(v1.p,n3);
connect(v1.n,n0);
connect(r1.p,n3);
connect(r1.n,n2);
connect(c1.p,n2);
connect(c1.n,n0);
connect(g.p,n0);
end lpf;
```

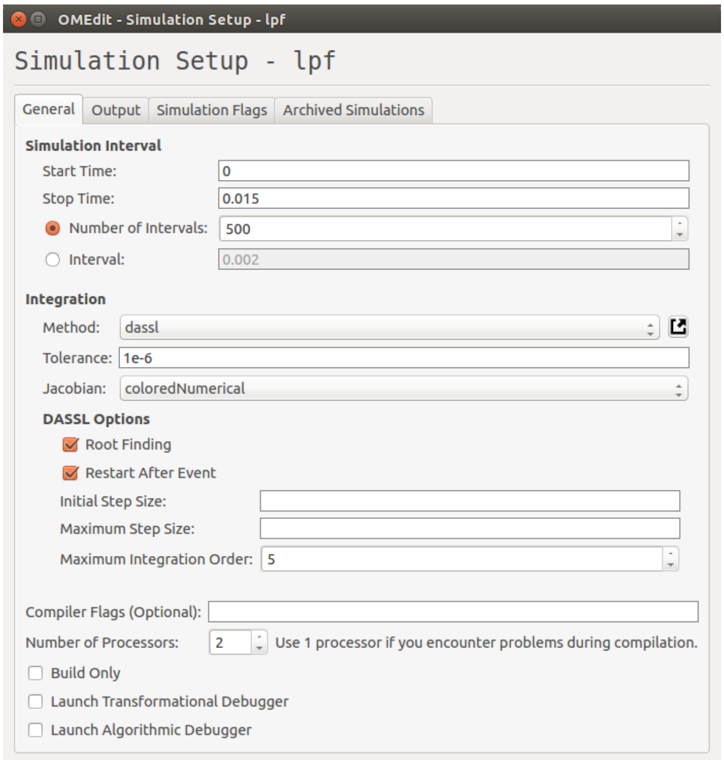
```
sh: 1: impact: not found


[1] 21:21:16 Translation Warning
Assuming fixed start value for the following 1 variables:
     c1.v:VARIABLE(start = 0.0 unit = "V" )  "Voltage drop between the two pins (= p.v - n.v)" type: Rea
l
```

Figure 10.5: OpenModelica: Text view

circuit in Fig. 10.8. This is an illustration of the Maximum Power Transfer Theorem. The power is maximum when R2 = R1, i.e., when R2 = 100. So maximum power would be Pmax = 0.0625. Let us now see the steps to be followed find the value of R2 using eSim.

1. Follow all the steps as above and generate the Modelica model using the Ngspice to Modelica converter.

2. The objective function is $Power = i^2 \times R2$. To define the objective function, the line $power := i^2 \times R1 + i^2 \times R2$ is added under the keyword algorithm, in the



OMEdit - Simulation Setup - lpf

Simulation Setup - lpf

General | Output | Simulation Flags | Archived Simulations

Simulation Interval

Start Time: 0

Stop Time: 0.015

● Number of Intervals: 500

○ Interval: 0.002

Integration

Method: dassl

Tolerance: 1e-6

Jacobian: coloredNumerical

DASSL Options

☑ Root Finding

☑ Restart After Event

Initial Step Size:

Maximum Step Size:

Maximum Integration Order: 5

Compiler Flags (Optional):

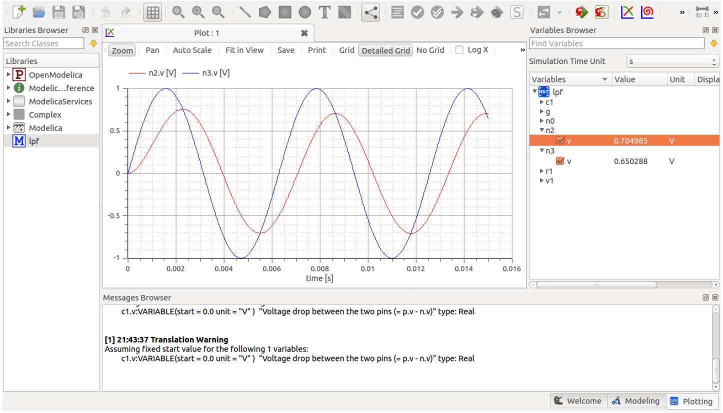
Number of Processors: 2 Use 1 processor if you encounter problems during compilation.

☐ Build Only

☐ Launch Transformational Debugger

☐ Launch Algorithmic Debugger

Figure 10.6: OpenModelica: Simulation setup



Simulation Time Unit: s

Variables	Value	Unit
lpf		
c1		
g		
n0		
n2		
☑ v	0.704985	V
n3		
☑ v	0.650288	V
r1		
v1		

Messages Browser

```
    c1.v:VARIABLE(start = 0.0 unit = "V" )  "Voltage drop between the two pins (= p.v - n.v)" type: Real

[1] 21:43:37 Translation Warning
Assuming fixed start value for the following 1 variables:
    c1.v:VARIABLE(start = 0.0 unit = "V" )  "Voltage drop between the two pins (= p.v - n.v)" type: Real
```

Figure 10.7: OpenModelica: Simulation

Modelica model file.

3. Select `OMOptim` from eSim left toolbar, in the displayed window click on `New`

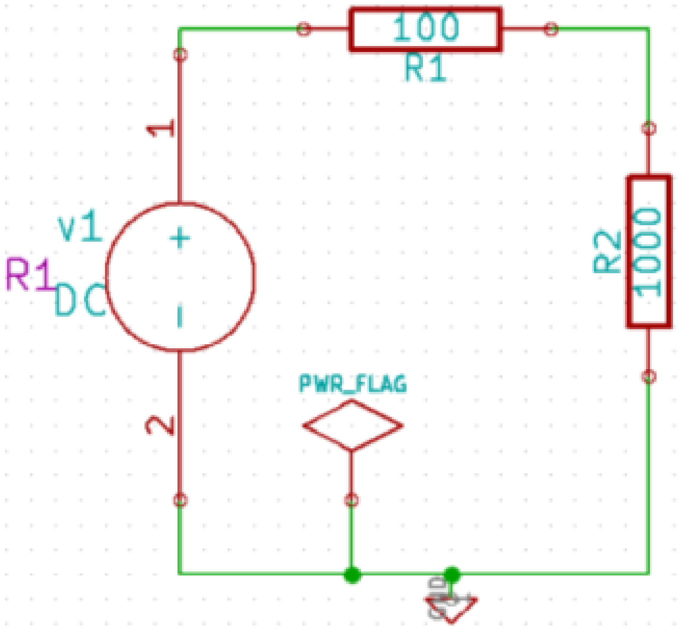


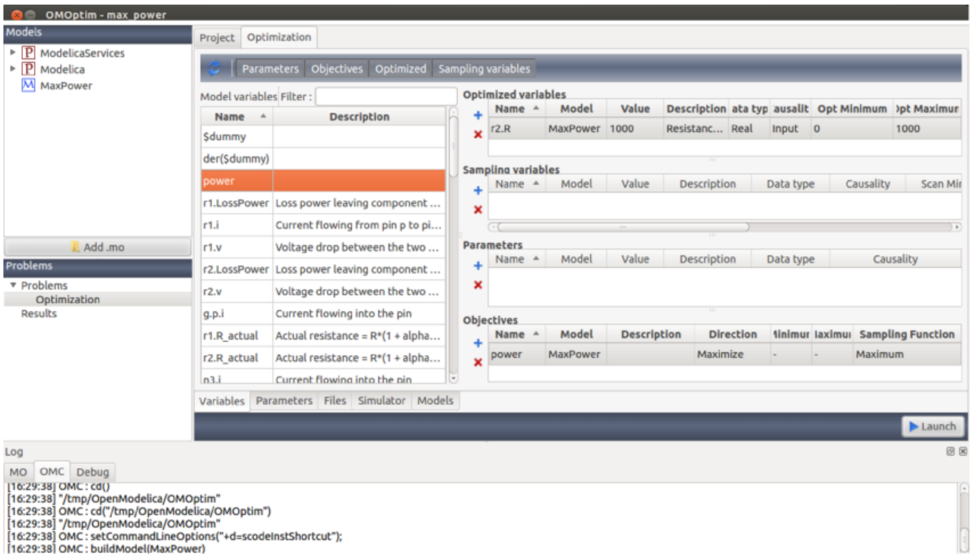
Figure 10.8: Circuit schematic for optimisation

`Project`. Then save the project. It is stored with an extension `.min`. Now select `Models` and then `Load Modelica Library`. Now select `Load mo file` under `Models`. It will be added on the left.

4. Click `Problems` and then `Optimisation`. Select the model to be optimised. *Note that for optimising, that model has to be loaded in OpenModelica as stated before.* Clicking blue turnover icon will display all the variables used in the model. Add details like optimsation variables and objective.

   The OMOptim project for this problem is given in Fig. 10.9. Power is the objective function that has to be maximized. `r2.R` is the variable that will be varied. `r2.R` is limited between 0 and 1000.

5. Click on Parameters tab to select the type of algorithm and its parameters. In this example, the optimisation algorithm used is PSO (Particle Swarm Optimisation). The various parameter values given are as follows: population size as 50, Inertia factor as 1, Learning factor: alpha and beta as 2, Population saving frequency was 1. Iteration limit is also specified. Select the .mo file to be simulated from `Files` tab. Click on `Launch`. The results of optimisation for various values of Iteration Limit are given in Fig. 10.10.



Model variables Filter :

Name	Description
$dummy	
der($dummy)	
power	
r1.LossPower	Loss power leaving component ...
r1.i	Current flowing from pin p to pi...
r1.v	Voltage drop between the two ...
r2.LossPower	Loss power leaving component ...
r2.v	Voltage drop between the two ...
g.p.i	Current flowing into the pin
r1.R_actual	Actual resistance = R*(1 + alpha...
r2.R_actual	Actual resistance = R*(1 + alpha...
n3.i	Current flowing into the pin

Optimized variables

Name	Model	Value	Description	ata typ	ausalit	Opt Minimum	)pt Maximur
r2.R	MaxPower	1000	Resistanc...	Real	Input	0	1000

Sampling variables

Name	Model	Value	Description	Data type	Causality	Scan Mir

Parameters

Name	Model	Value	Description	Data type	Causality

Objectives

Name	Model	Description	Direction	1inimur	laximur	Sampling Function
power	MaxPower		Maximize	-	-	Maximum

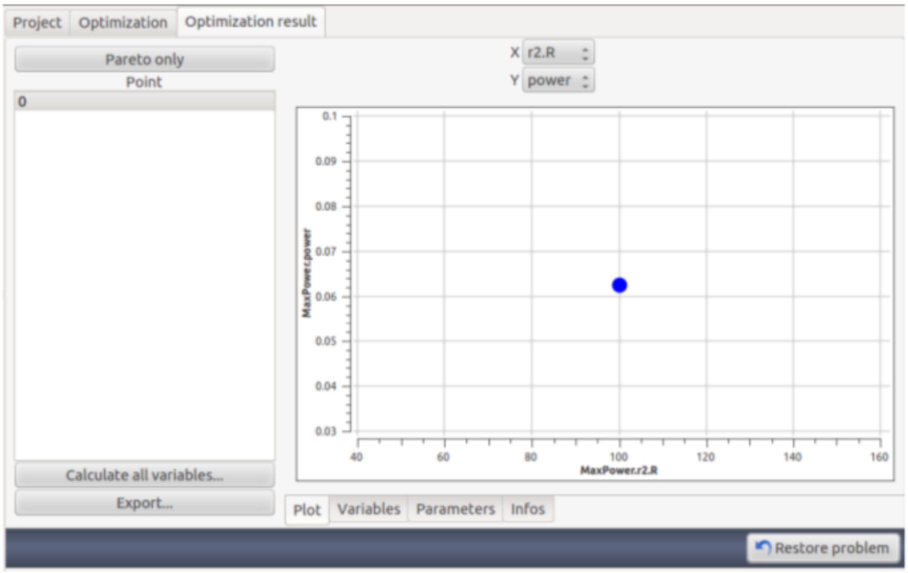
```
[16:29:38] OMC : cd()
[16:29:38] "/tmp/OpenModelica/OMOptim"
[16:29:38] OMC : cd("/tmp/OpenModelica/OMOptim")
[16:29:38] "/tmp/OpenModelica/OMOptim"
[16:29:38] OMC : setCommandLineOptions("+d=scodeInstShortcut");
[16:29:38] OMC : buildModel(MaxPower)
```

Figure 10.9: OMOptim project

Iteration Limit	$R_2(\Omega)$	Power (W)
100	102.652	0.0624893
200	99.9942	0.0625
300	99.9221	0.0625
400	98.6115	0.0624969
500	99.9923	0.0625
600	100.968	0.0624985
700	100.648	0.0624993

Figure 10.10: Optimisation values for various Iteration Limit

6. Depending on the type of algorithm, the time for optimisation varies. Optimised result is graphically displayed as shown in Fig. 10.11.



X r2.R

Y power

Figure 10.11: Optimised value of resistance for maximum power

Chapter 11

Solved Examples

11.1 Solved Examples

11.1.1 Basic RC Circuit

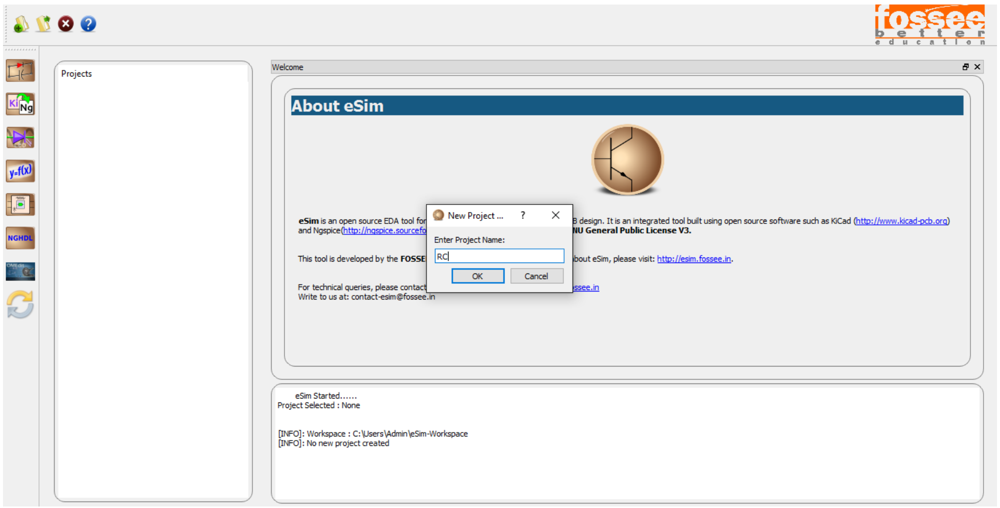
Problem Statement:

Plot the Input and Output Waveform of an RC circuit whose input voltage (Vs) is 50Hz, 3V peak to peak. The values of Resistor (R) and Capacitor(C) are $1k$ and $1uf$ respectively.

Solution:

- Creating a Project: The new project is created by clicking the `New` icon on the menubar. The name of the project is given in the pop up window as shown in Fig. 11.1.
- Creating the Schematic: To create the schematic, click the very first icon of the left toolbar as shown in the Fig. 11.2. This will open KiCad Eeschema.

  To create a schematic in KiCad, we need to place the required components. Fig. 11.3 shows the icon on the right toolbar which opens the component library.



eSim is an open source EDA tool for B design. It is an integrated tool built using open source software such as KiCad (http://www.kicad-pcb.org) and Ngspice(http://ngspice.sourcefo NU General Public License V3.

This tool is developed by the FOSSE bout eSim, please visit: http://esim.fossee.in.

For technical queries, please contact ssee.in
Write to us at: contact-esim@fossee.in

RC

```
     eSim Started......
Project Selected : None


[INFO]: Workspace : C:\Users\Admin\eSim-Workspace
[INFO]: No new project created
```

Figure 11.1: Creating New Project

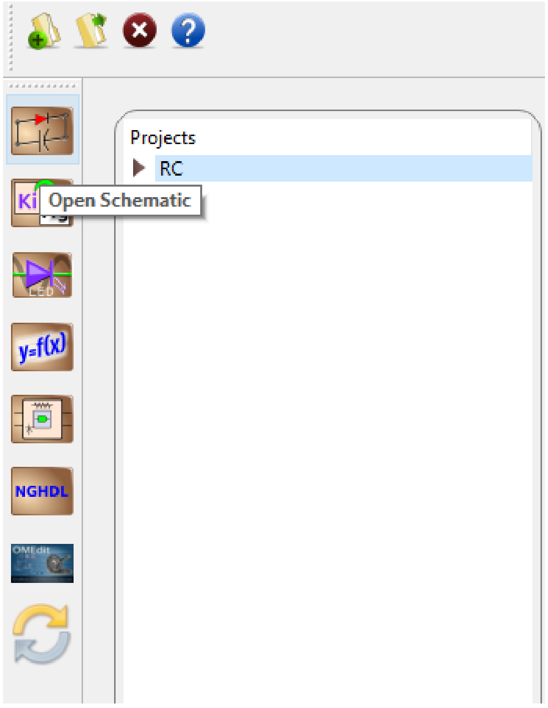


Figure 11.2: Open Schematic Editor

After all the required components of the simple RC circuit are placed, wiring is done using the `Place Wire` option as shown in the Fig. 11.4

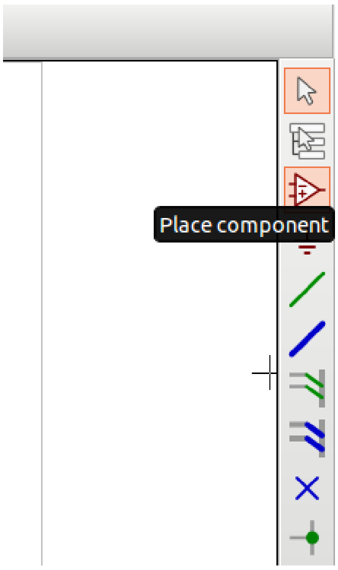


Figure 11.3: Place Component Icon

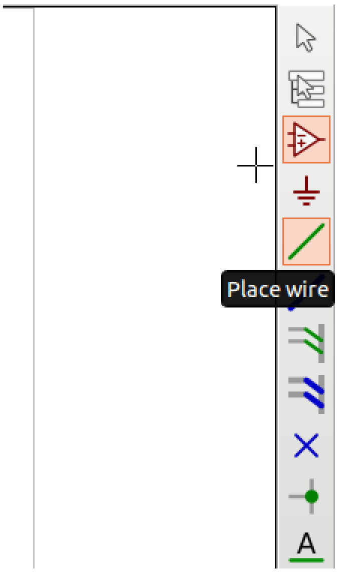


Figure 11.4: Place Wire Icon

Next step is `ERC (Electric Rules Check)`. Fig. 11.5 shows the icon for `ERC`. Fig. 11.6 shows the RC circuit after connecting the components by wire.

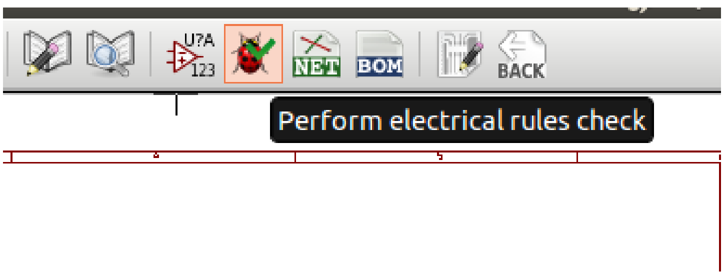


Figure 11.5: Electric Rules Check Icon

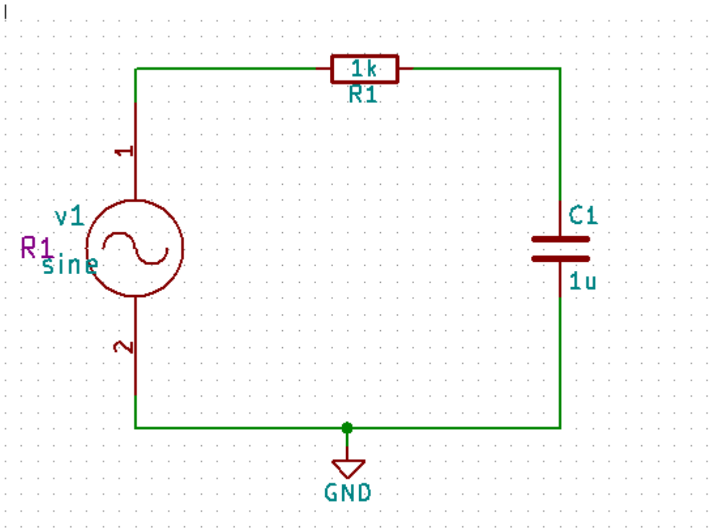
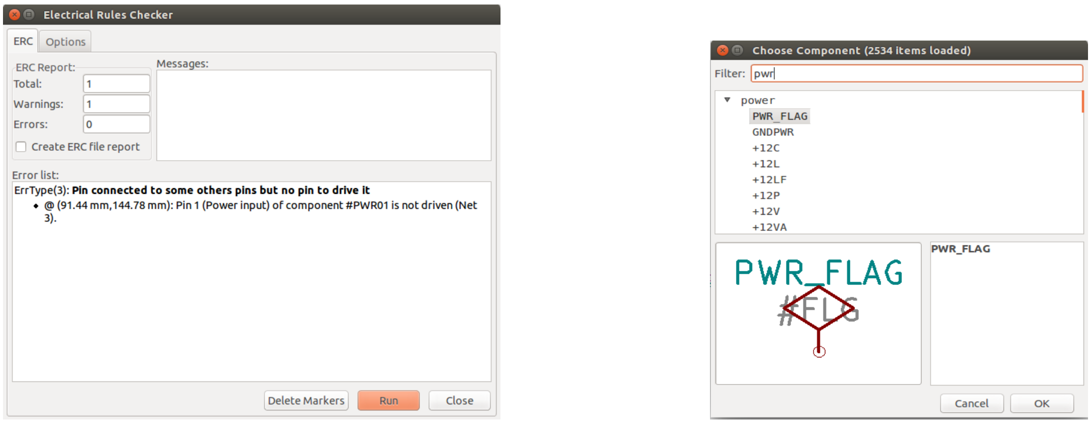


Figure 11.6: RC circuit

After clicking the `ERC` icon a window opens up. Click the `Run` button to run rules check. The errors are listed in as shown in Fig. 11.7a. This error is handled by adding `Power Flag` as shown in Fig. 11.7b.

After adding the `Power Flag` the completed RC circuit is shown in Fig. 11.8a and the netlist is generated as shown in Fig. 11.8b.



Electrical Rules Checker

ERC | Options

ERC Report:

Total: 1

Warnings: 1

Errors: 0

□ Create ERC file report

Messages:

Error list:

ErrType(3): **Pin connected to some others pins but no pin to drive it**

- @ (91.44 mm,144.78 mm): Pin 1 (Power input) of component #PWR01 is not driven (Net 3).

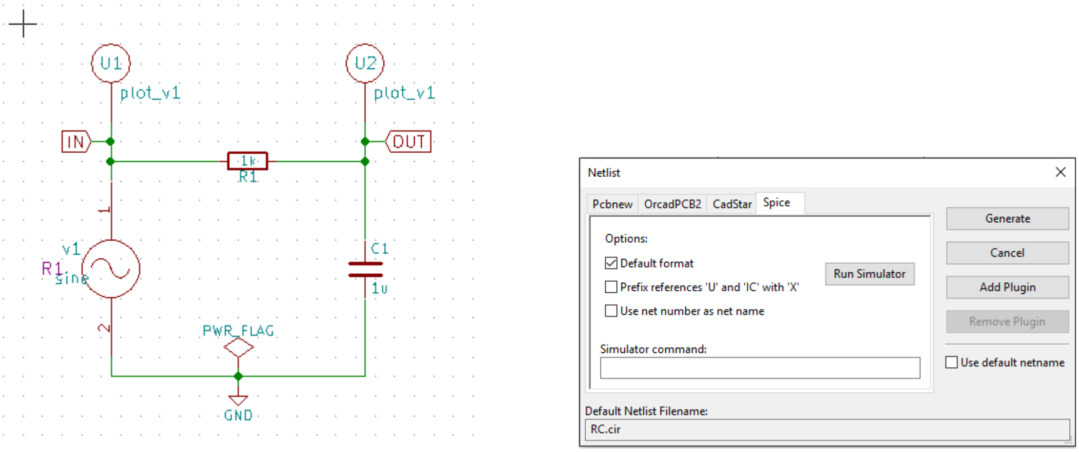
(a) ERC Run

Choose Component (2534 items loaded)

Filter: pwr

(b) Power Flag

Figure 11.7: ERC check and POWER FLAG



(a) Schematic of RC circuit

Netlist

Pcbnew | OrcadPCB2 | CadStar | Spice

Options:

☑ Default format

□ Prefix references 'U' and 'IC' with 'X'

□ Use net number as net name

Simulator command:

□ Use default netname

Default Netlist Filename: RC.cir

(b) Generating KiCad Netlist of RC circuit

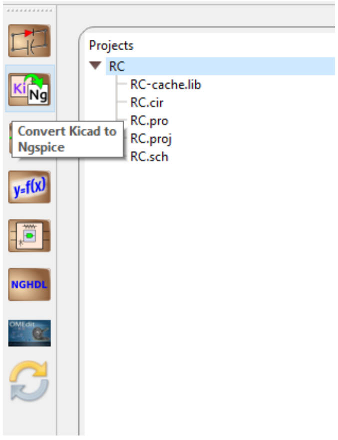
Figure 11.8: RC Schematic and Netlist Generation

- Convert KiCad to Ngspice: To convert KiCad netlist of RC circuit to NgSpice compatible netlist click on KiCad to Ngspice icon as shown in Fig. 11.9.

  Now you can enter the type of analysis and source details as shown in Fig. 11.10a and Fig. 11.10b respectively.

  The other tab will be empty as RC circuit do not use any Ngspice model, device library and subcircuit.

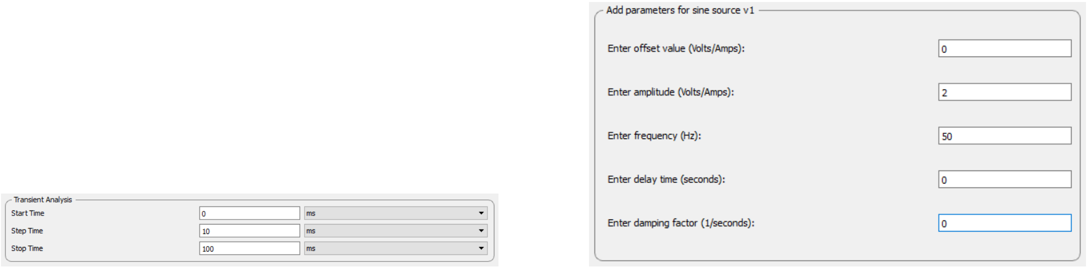
  After entering the value, press the convert button. It will convert the netlist into Ngspice compatible netlist.



Convert Kicad to Ngspice

Projects
- RC
  - RC-cache.lib
  - RC.cir
  - RC.pro
  - RC.proj
  - RC.sch

Figure 11.9: Convert KiCad to Ngspice Icon



Transient Analysis

Start Time	0	ms
Step Time	10	ms
Stop Time	100	ms

(a) RC Analysis

Add parameters for sine source v1

Enter offset value (Volts/Amps):	0
Enter amplitude (Volts/Amps):	2
Enter frequency (Hz):	50
Enter delay time (seconds):	0
Enter damping factor (1/seconds):	0

(b) RC Source Details

Figure 11.10: RC Analysis and Source Detail

- Simulation: To run Ngspice simulation click the simulation icon in the tool bar as shown in the Fig. 11.11.

  In eSim, there are two types of plot. First is normal Ngspice plot and second is interactive python plot as shown in Fig. 11.12a and Fig. 11.12b respectively.

  In the interactive python plot you can select any node or branch to plot voltage or current across it. Also it has the facility to plot basic functions across the node like addition, substraction, multiplication, division and v/s.

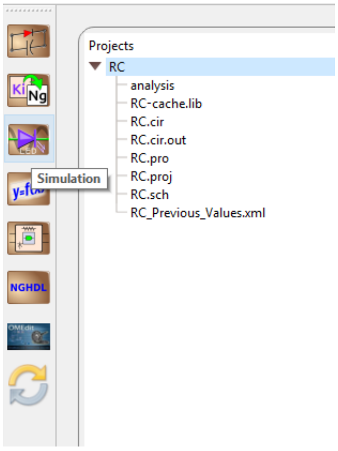
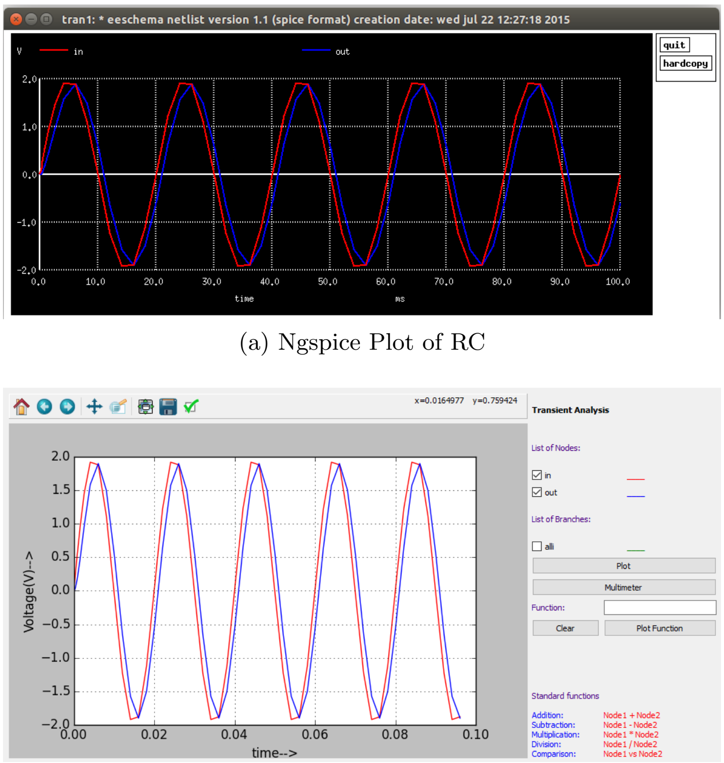


Figure 11.11: Simulation Icon



(a) Ngspice Plot of RC

(b) Python Plot of RC

Figure 11.12: Ngspice and Interactive Python Plotting

11.1.2 Half Wave Rectifier

Problem Statement:

Plot the Input and Output Waveform of Half Wave Rectifier circuit where the input voltage (Vs) is 50Hz, 2V peak to peak. The value for Resistor (R) is 1k.

Solution:

The new project is created by clicking the `New` icon on the menubar. The name of the project is given in the window shown in Fig. 11.1.

- Creating Schematic: To create the schematic, click the very first icon of the left toolbar as shown in the Fig. 11.2. This will open KiCad Eeschema.

  After the KiCad window is opened, to create a schematic we need to place the required components. Fig. 11.3 shows the icon on the right toolbar which opens the component library.

  After all the required components of the simple Half Wave rectifier circuits are placed, wiring is done using the `Place Wire` option as shown in the Fig. 11.4

  Next step is `ERC (Electric Rules Check)`. Fig. 11.5 shows the icon for `ERC`. After completing all the above steps the final Half Wave Rectifier schematic will look like Fig. 11.13.

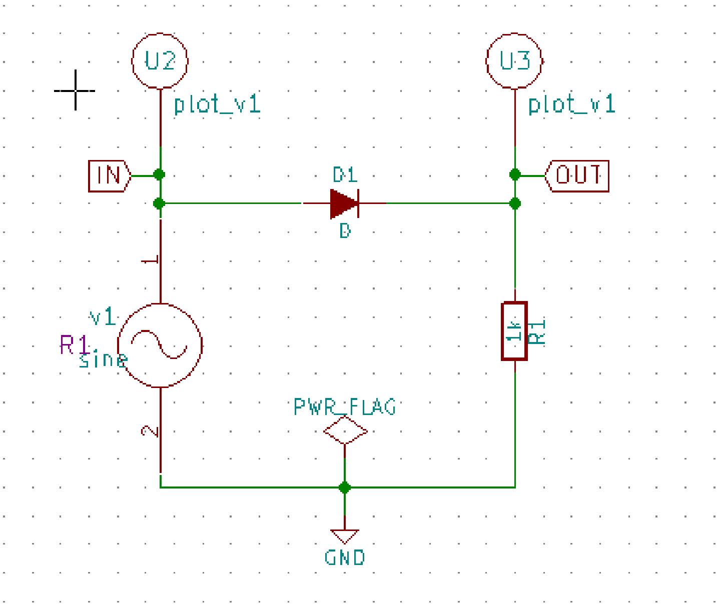
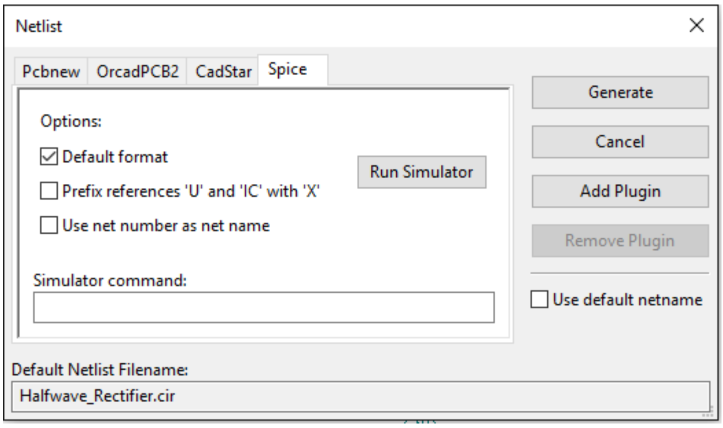


Figure 11.13: Schematic of Half Wave Rectifier circuit

KiCad netlist is generated as shown in the Fig. 11.14



Options:

☑ Default format

☐ Prefix references 'U' and 'IC' with 'X'

☐ Use net number as net name

Simulator command:

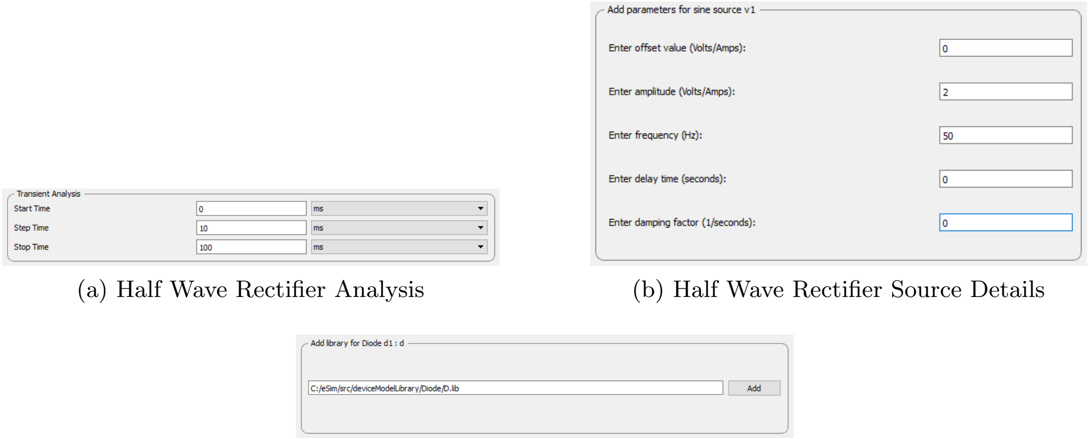
☐ Use default netname

Default Netlist Filename:

Halfwave_Rectifier.cir

Figure 11.14: Half Wave Rectifier circuit Netlist Generation

- Convert KiCad to Ngspice: After creating KiCad netlist, click on the `KiCad-Ngspice converter` button. This will open converter window where you can enter details of Analysis, Source values and Device library.



Transient Analysis

Start Time: 0 ms

Step Time: 10 ms

Stop Time: 100 ms

(a) Half Wave Rectifier Analysis

Add parameters for sine source v1

Enter offset value (Volts/Amps): 0

Enter amplitude (Volts/Amps): 2

Enter frequency (Hz): 50

Enter delay time (seconds): 0

Enter damping factor (1/seconds): 0

(b) Half Wave Rectifier Source Details

Add library for Diode d1 : d

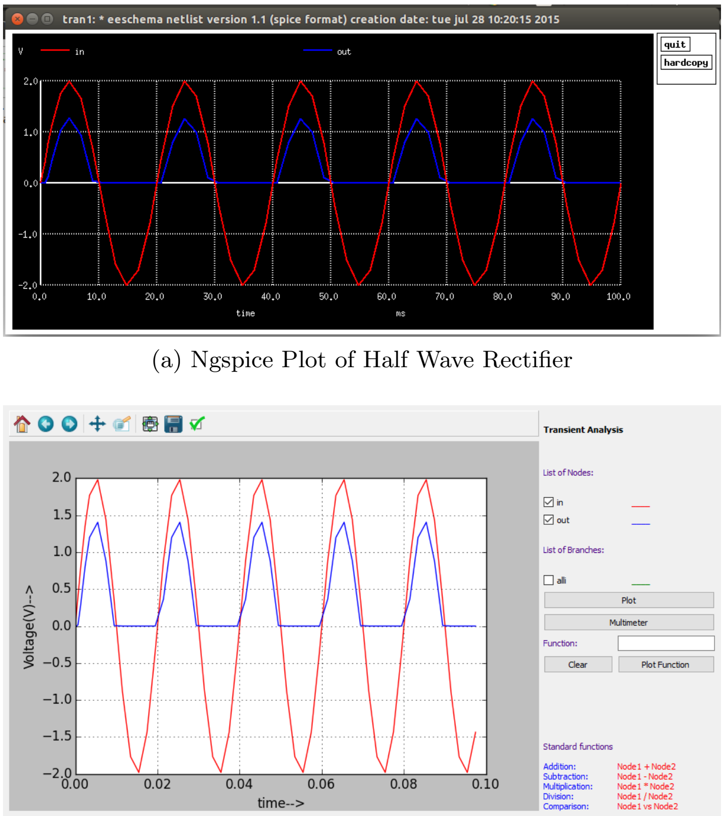
C:/eSim/src/deviceModelLibrary/Diode/D.lib

(c) Half Wave Rectifier Device Modeling

Figure 11.15: Analysis, Source and Device Tab

Under device library you can add the library for diode used in the circuit. If you do not add any library it will take default Ngspice model.

- Simulation: Once the KiCad-Ngspice converter runs successfully, you can run simulation by clicking the simulation button in the toolbar.



(a) Ngspice Plot of Half Wave Rectifier

(b) Python Plot of Half Wave Rectifier

Figure 11.16: Half Wave Rectifier Simulation Output

11.1.3 Inverting Amplifier

Problem Statement:

Plot the Input and Output Waveform of Inverting Amplifier circuit where the input voltage (Vs) is $50Hz$, $2V$ peak to peak and gain is 2.

Solution:

- Creating Schematic: To create the schematic, click the very first icon of the left toolbar as shown in the Fig. 11.2. This will open KiCad Eeschema.
  After the KiCad window is opened, to create a schematic we need to place the required components. Fig. 11.3 shows the icon on the right toolbar which opens the component library.
  After all the required components of the inverting amplifier circuit are placed,

wiring is done using the `Place Wire` option as shown in the Fig. 11.4.
Next step is `ERC (Electric Rules Check)`. Fig. 11.5 shows the icon for `ERC`.

The Fig. 11.17 shows the complete Precision Rectifier schematic after removing the errors.

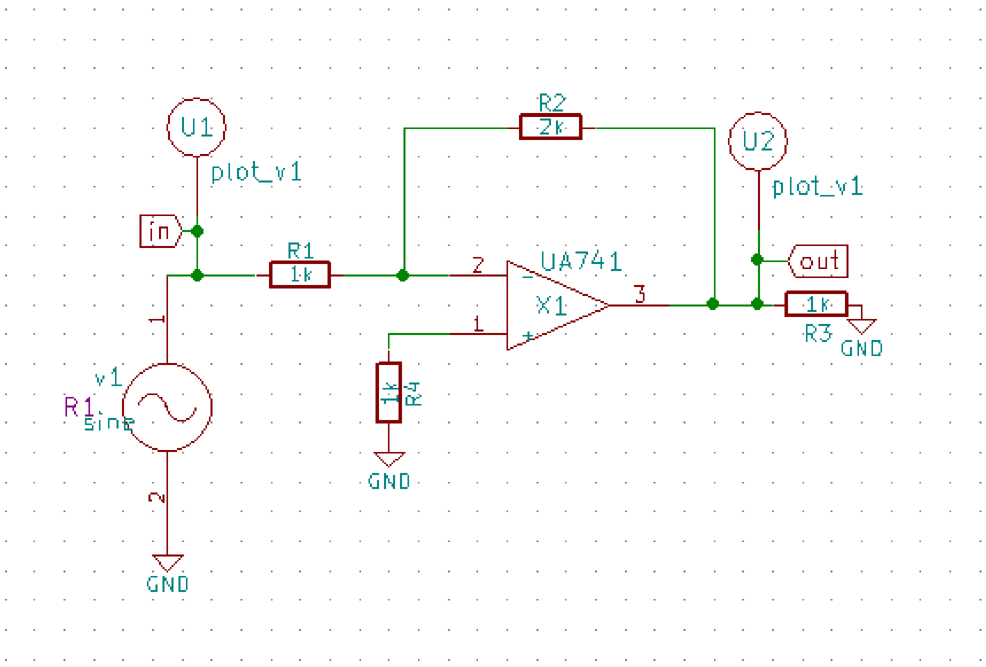


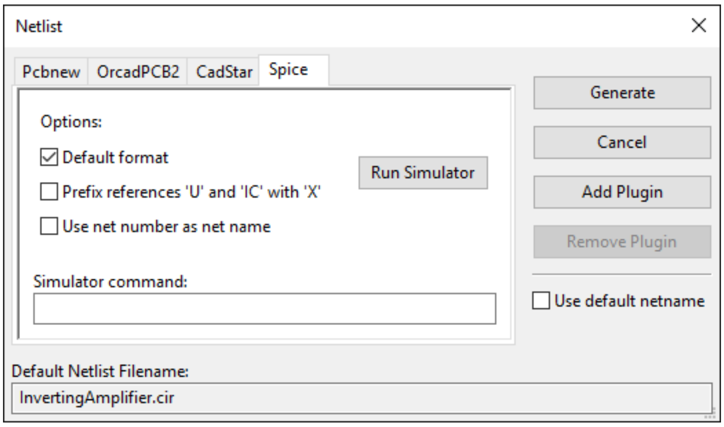
Figure 11.17: Schematic of Inverting Amplifier circuit

The KiCad netlist is generated as shown in Fig. 11.18.

- Convert KiCad to Ngspice: After creating KiCad netlist, click on KiCad-Ngspice converter button.

  This will open converter window where you can enter details of Analysis, Source values, Device library and Subcircuit.

  Subcircuit of Op-Amp is shown in Fig. 11.19d



Netlist

Pcbnew | OrcadPCB2 | CadStar | Spice

Options:

☑ Default format

☐ Prefix references 'U' and 'IC' with 'X'

☐ Use net number as net name

Run Simulator

Simulator command:

Generate

Cancel

Add Plugin

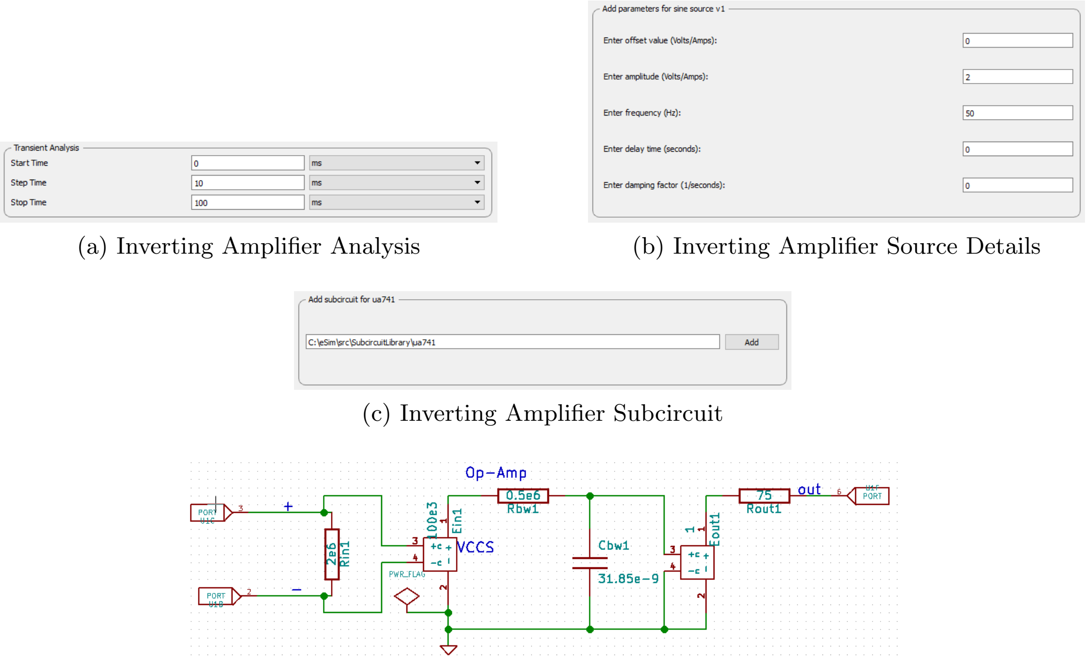
Remove Plugin

☐ Use default netname

Default Netlist Filename:

InvertingAmplifier.cir

Figure 11.18: Inverting Amplifier circuit Netlist Generation



Transient Analysis

Start Time	0	ms
Step Time	10	ms
Stop Time	100	ms

(a) Inverting Amplifier Analysis

Add parameters for sine source v1

Enter offset value (Volts/Amps):	0
Enter amplitude (Volts/Amps):	2
Enter frequency (Hz):	50
Enter delay time (seconds):	0
Enter damping factor (1/seconds):	0

(b) Inverting Amplifier Source Details

Add subcircuit for ua741

C:\eSim\src\SubcircuitLibrary\ua741

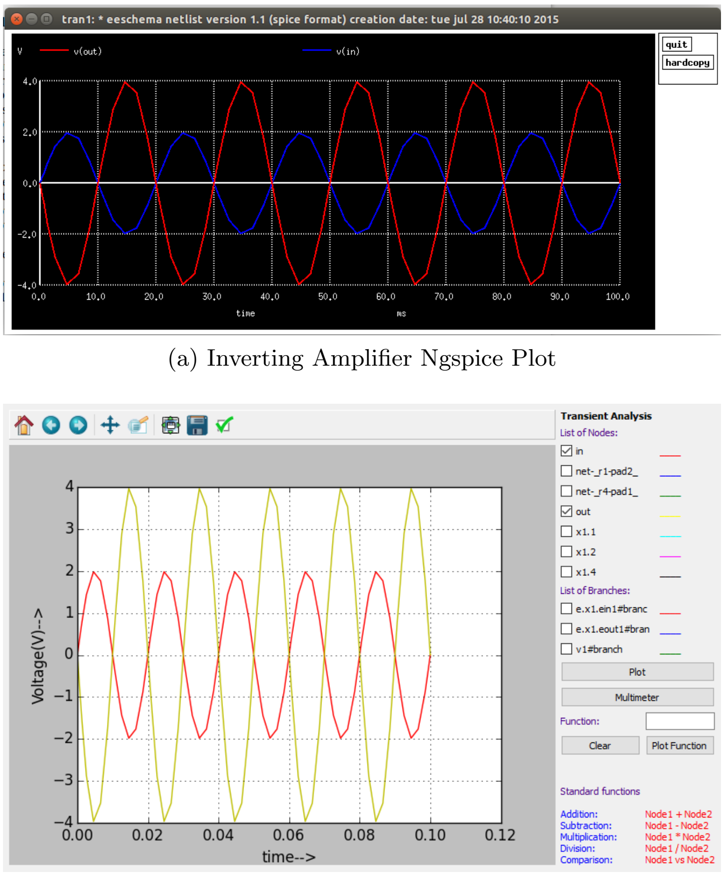
Add

(c) Inverting Amplifier Subcircuit

(d) Sub-Circuit of Op-Amp

Figure 11.19: Analysis, Source, and Subcircuit tab

Under subcircuit tab you have to add the subciruit used in your circuit. If you forget to add subcircuit, it will throw an error.



Transient Analysis

List of Nodes:

☑ in

☐ net-_r1-pad2_

☐ net-_r4-pad1_

☑ out

☐ x1.1

☐ x1.2

☐ x1.4

List of Branches:

☐ e.x1.ein1#branc

☐ e.x1.eout1#bran

☐ v1#branch

Function:

Standard functions

Addition:	Node1 + Node2
Subtraction:	Node1 - Node2
Multiplication:	Node1 * Node2
Division:	Node1 / Node2
Comparison:	Node1 vs Node2

(a) Inverting Amplifier Ngspice Plot

(b) Inverting Amplifier Python Plot

Figure 11.20: Inverting Amplifier Simulation Output

- Simulation: Once the KiCad-Ngspice converter runs successfully, you can run simulation by clicking the simulation button in the toolbar.

11.1.4 Half Adder

Problem Statement:

Plot the Input and Output Waveform of Half Adder circuit.

Solution:

- Creating Schematic: To create the schematic, click the very first icon of the left toolbar as shown in the Fig. 11.2. This will open KiCad Eeschema.
  After the KiCad window is opened, to create a schematic we need to place the required components. Fig. 11.3 shows the icon on the right toolbar which opens the component library.
  After all the required components of the Half Adder circuit are placed, wiring is done using the `Place Wire` option as shown in the Fig. 11.4.
  Next step is `ERC (Electric Rules Check)`. Fig. 11.5 shows the icon for `ERC`.

  The Fig. 11.21 shows the complete Half Adder schematic after removing the errors.

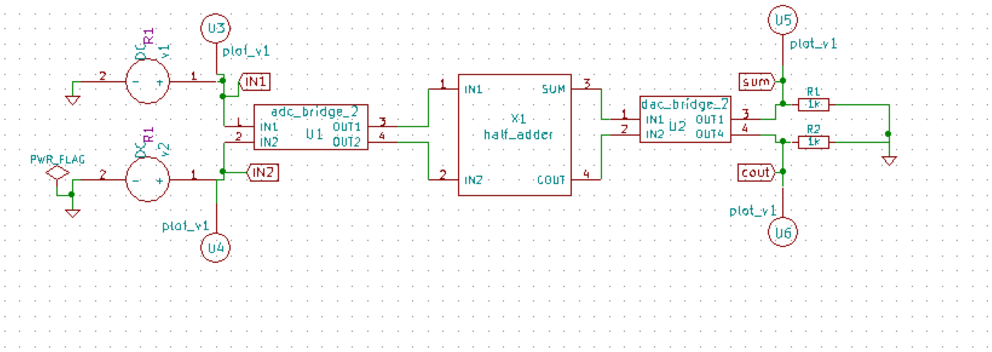
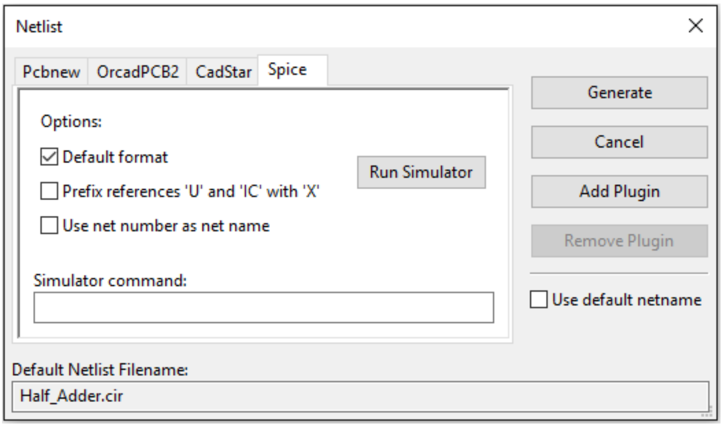


Figure 11.21: Schematic of Half Adder circuit

The KiCad netlist is generated as shown in Fig. 11.22.



Netlist

Pcbnew | OrcadPCB2 | CadStar | Spice

Options:

☑ Default format

□ Prefix references 'U' and 'IC' with 'X'

□ Use net number as net name

Run Simulator

Simulator command:

Generate

Cancel

Add Plugin

Remove Plugin

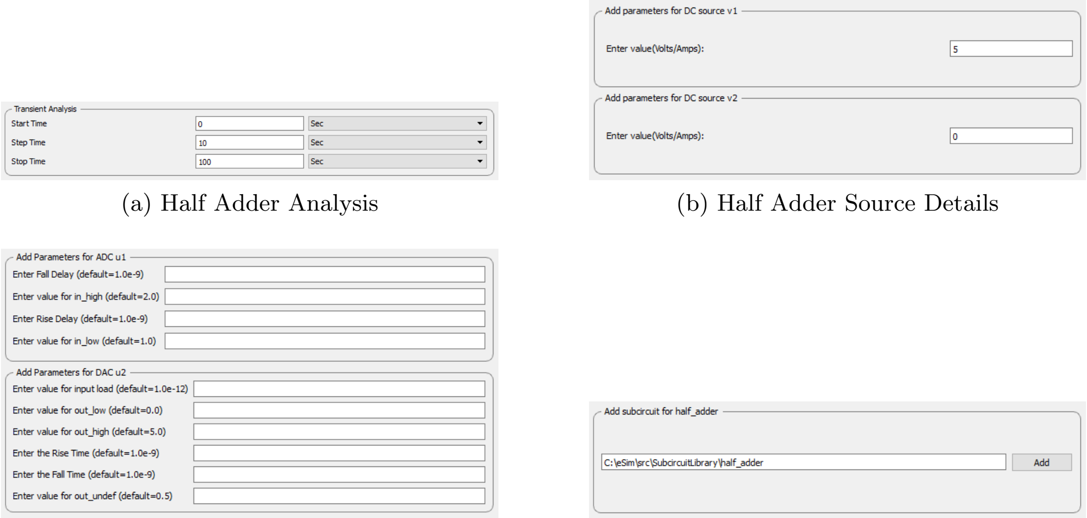
□ Use default netname

Default Netlist Filename:
Half_Adder.cir

Figure 11.22: Half Adder circuit Netlist Generation

- Convert KiCad to Ngspice: After creating KiCad netlist click on KiCad-Ngspice converter button.

  This will open converter window where you can enter details of Analysis, Source values, Ngspice model and Subcircuit.



Transient Analysis

Start Time	0	Sec
Step Time	10	Sec
Stop Time	100	Sec

(a) Half Adder Analysis

Add parameters for DC source v1
Enter value(Volts/Amps): 5

Add parameters for DC source v2
Enter value(Volts/Amps): 0

(b) Half Adder Source Details

Add Parameters for ADC u1
Enter Fall Delay (default=1.0e-9)
Enter value for in_high (default=2.0)
Enter Rise Delay (default=1.0e-9)
Enter value for in_low (default=1.0)

Add Parameters for DAC u2
Enter value for input load (default=1.0e-12)
Enter value for out_low (default=0.0)
Enter value for out_high (default=5.0)
Enter the Rise Time (default=1.0e-9)
Enter the Fall Time (default=1.0e-9)
Enter value for out_undef (default=0.5)

(c) Half Adder Ngspice Model

Add subcircuit for half_adder
C:\eSim\src\SubcircuitLibrary\half_adder

(d) Half Adder Subcircuit Model

Figure 11.23: Analysis, Source, Ngspice Model and Subcircuit tab

Subcircuit of Half Adder in Fig. 11.24

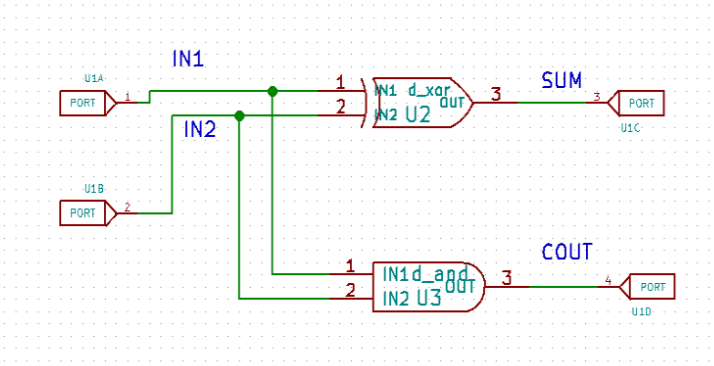
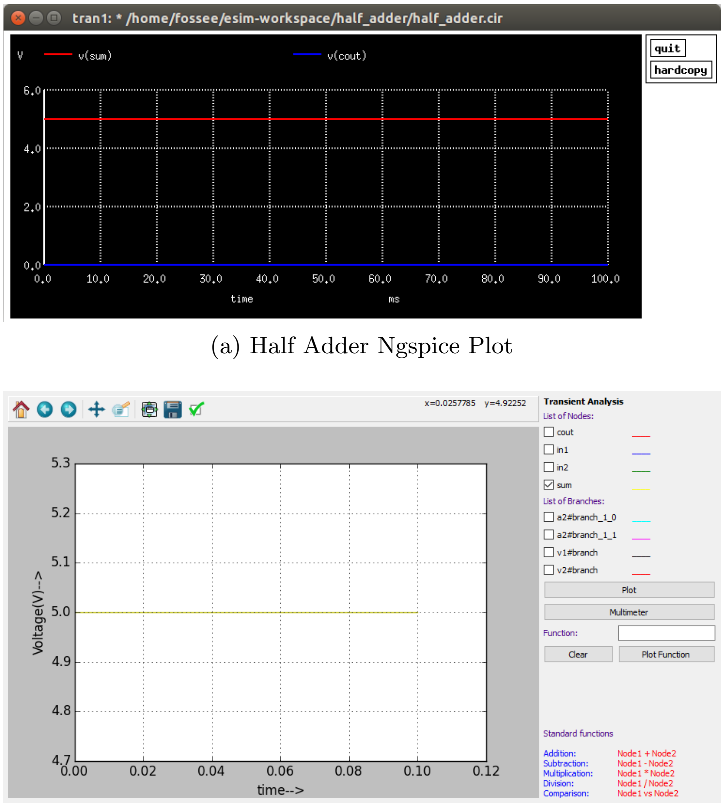


Figure 11.24: Half Adder Subcircuit

- Simulation: Once the KiCad-Ngspice converter runs successfully, you can run simulation by clicking the simulation button in the toolbar.



(a) Half Adder Ngspice Plot

(b) Half Adder Python Plot

Figure 11.25: Half Adder Simulation Output

11.1.5 Full Wave Rectifier using SCR

Problem Statement:

Plot the Input and Output Waveform of Full Wave Rectifier using SCR.

Solution:

- Creating Schematic: To create the schematic, click the very first icon of the left toolbar as shown in the Fig. 11.2. This will open KiCad Eeschema.
  After the KiCad window is opened, to create a schematic we need to place the required components. Fig. 11.3 shows the icon on the right toolbar which opens the component library.
  After all the required components of the Full Wave Rectifier using SCR circuit are placed, wiring is done using the `Place Wire` option as shown in the Fig. 11.4.
  Next step is `ERC (Electric Rules Check)`. Fig. 11.5 shows the icon for `ERC`.

  The Fig. 11.26 shows the complete Rectifier circuit using SCR after removing the errors.

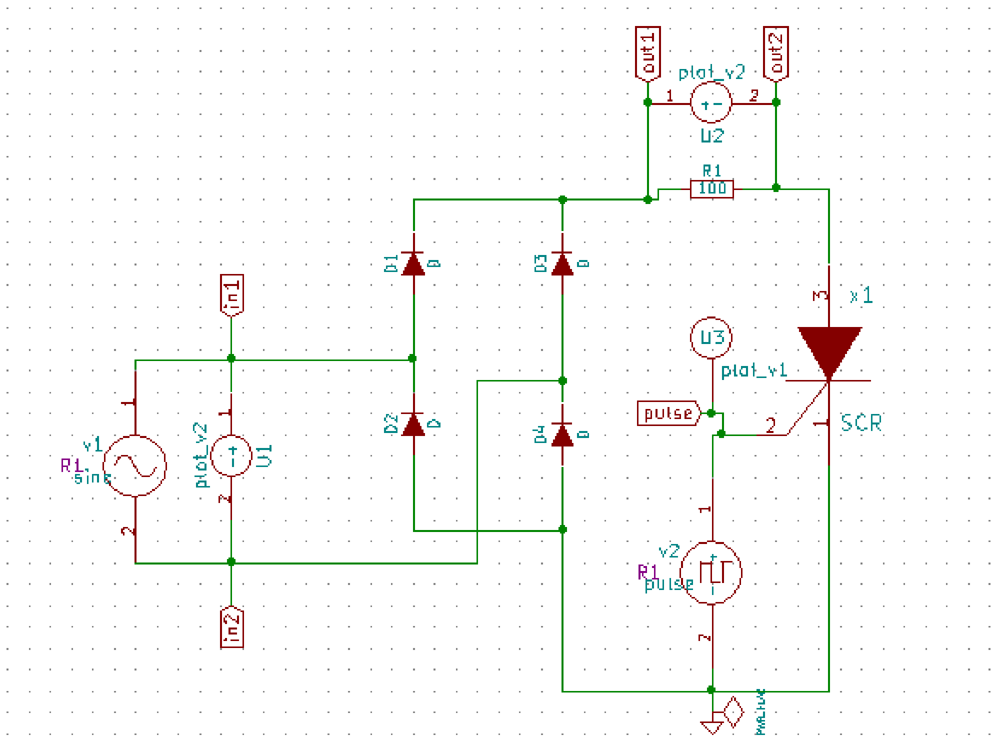
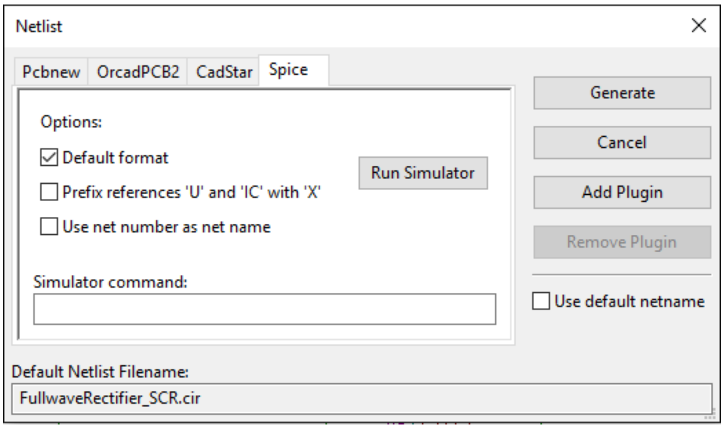


Figure 11.26: Schematic of Full Wave Rectifier using SCR

The KiCad netlist is generated as shown in Fig. 11.27.



Netlist

Pcbnew | OrcadPCB2 | CadStar | Spice

Options:

☑ Default format

□ Prefix references 'U' and 'IC' with 'X'

□ Use net number as net name

Simulator command:

□ Use default netname

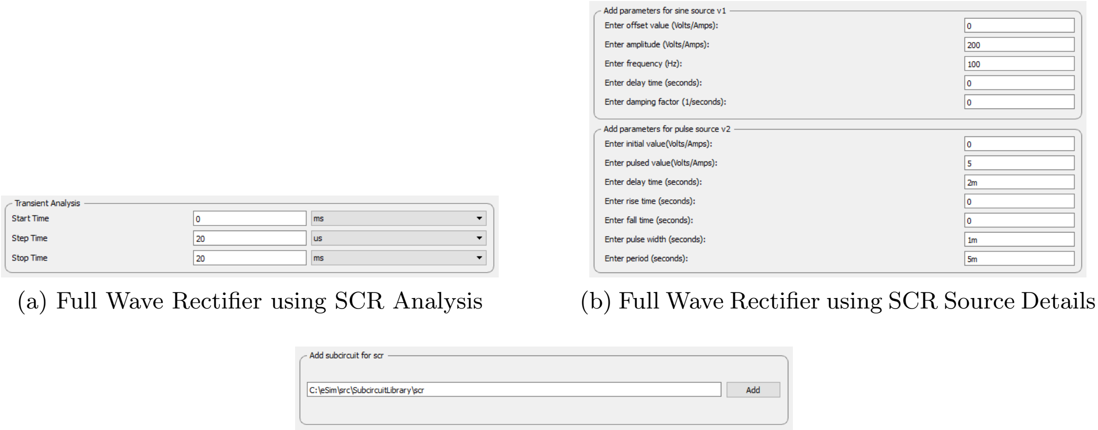
Default Netlist Filename:

FullwaveRectifier_SCR.cir

Figure 11.27: Full Wave Rectifier using SCR Netlist Generation

- Convert KiCad to Ngspice: After creating KiCad netlist click on KiCad-Ngspice converter button.

  This will open converter window where you can enter details of Analysis, Source values, Ngspice model and Subcircuit.



Transient Analysis

Start Time: 0 ms

Step Time: 20 us

Stop Time: 20 ms

(a) Full Wave Rectifier using SCR Analysis

Add parameters for sine source v1

Enter offset value (Volts/Amps): 0

Enter amplitude (Volts/Amps): 200

Enter frequency (Hz): 100

Enter delay time (seconds): 0

Enter damping factor (1/seconds): 0

Add parameters for pulse source v2

Enter initial value(Volts/Amps): 0

Enter pulsed value(Volts/Amps): 5

Enter delay time (seconds): 2m

Enter rise time (seconds): 0

Enter fall time (seconds): 0

Enter pulse width (seconds): 1m

Enter period (seconds): 5m

(b) Full Wave Rectifier using SCR Source Details

Add subcircuit for scr

C:\eSim\src\SubcircuitLibrary\scr

(c) Full Wave Rectifier using SCR Subcircuit Model

Figure 11.28: Analysis, Source and Subcircuit tab

Subcircuit of SCR in Fig. 11.29

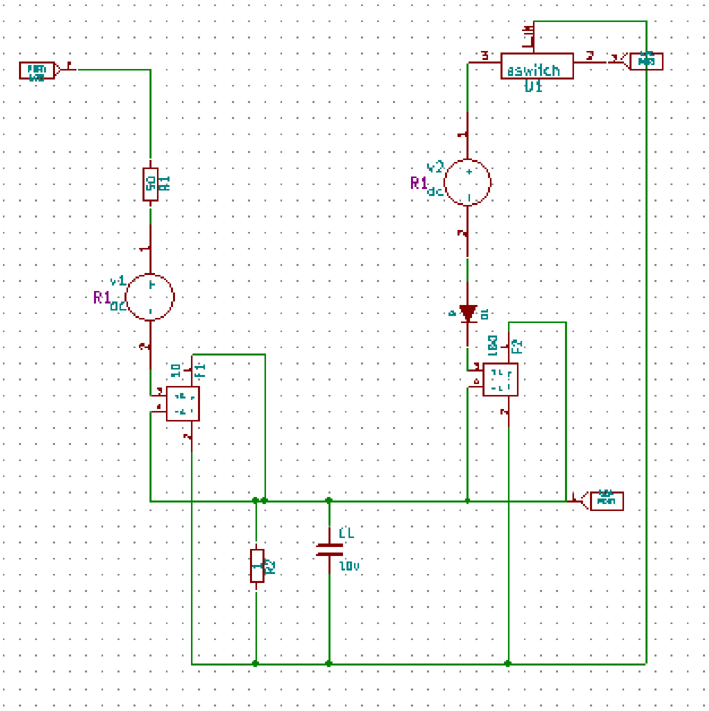
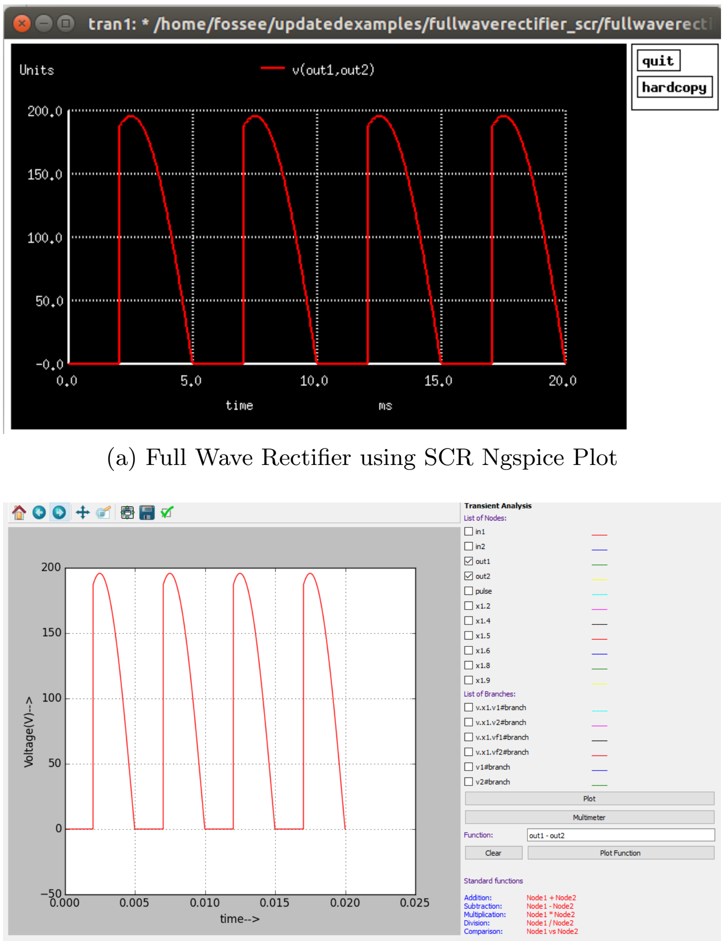


Figure 11.29: SCR Subcircuit

- Simulation: Once the KiCad-Ngspice converter runs successfully, you can run simulation by clicking the simulation button in the toolbar.



(a) Full Wave Rectifier using SCR Ngspice Plot

Transient Analysis

List of Nodes:

□ in1
□ in2
☑ out1
☑ out2
□ pulse
□ x1.2
□ x1.4
□ x1.5
□ x1.6
□ x1.8
□ x1.9

List of Branches:

□ v.x1.v1#branch
□ v.x1.v2#branch
□ v.x1.vf1#branch
□ v.x1.vf2#branch
□ v1#branch
□ v2#branch

Function: out1 - out2

Standard functions

Addition:	Node1 + Node2
Subtraction:	Node1 - Node2
Multiplication:	Node1 * Node2
Division:	Node1 / Node2
Comparison:	Node1 vs Node2

(b) Full Wave Rectifier using SCR Python Plot

Figure 11.30: Full Wave Rectifier using SCR Simulation Output

11.1.6 Oscillator Circuit

Problem Statement:

Plot the Oscillation Waveforms for Phase Shift Oscillator circuit.

Solution:

The new project is created by clicking the `New` icon on the menubar. The name of the project is given in the window shown in Fig. 11.1.

- Creating Schematic: To create the schematic, click the very first icon of the left toolbar as shown in the Fig. 11.2. This will open KiCad Eeschema.

After the KiCad window is opened, to create a schematic we need to place the required components. Fig. 11.3 shows the icon on the right toolbar which opens the component library.

After all the required components of the Oscillator circuits are placed, wiring is done using the `Place Wire` option as shown in the Fig. 11.4
sss

Next step is `ERC (Electric Rules Check)`. Fig. 11.5 shows the icon for `ERC`. After completing all the above steps the Oscillator schematic will look like Fig. 11.31.

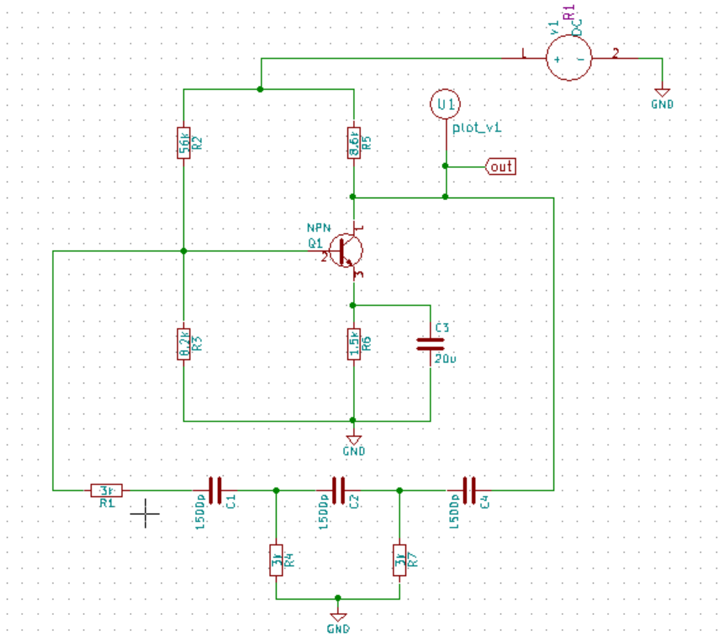
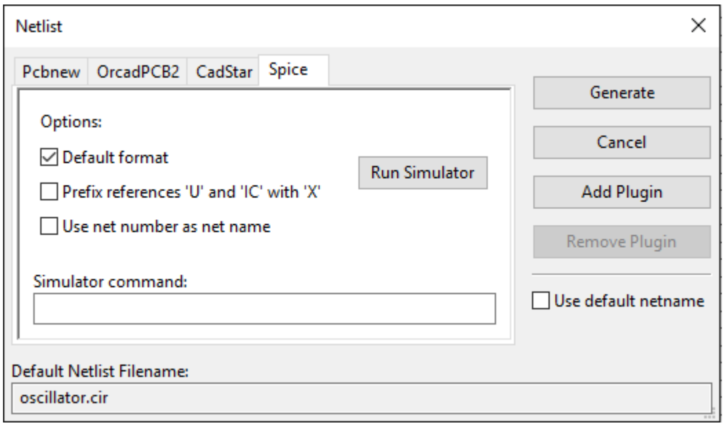


Figure 11.31: Schematic of Phase Shift Oscillator circuit

KiCad netlist is generated as shown in the Fig. 11.32



Options:

☑ Default format

□ Prefix references 'U' and 'IC' with 'X'

□ Use net number as net name

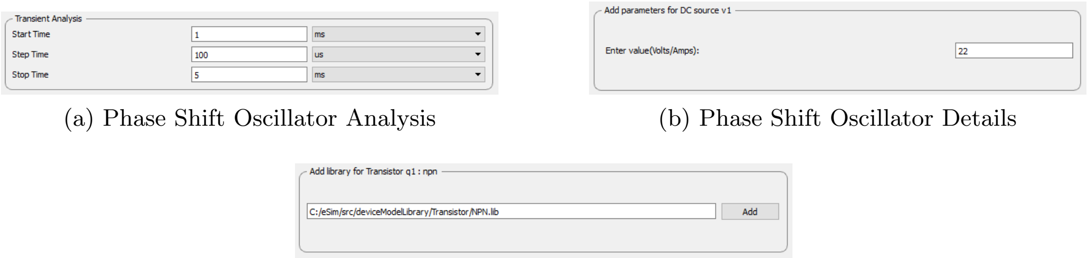
Simulator command:

□ Use default netname

Default Netlist Filename: oscillator.cir

Figure 11.32: Phase Shift Oscillator circuit Netlist Generation

- Convert KiCad to Ngspice: After creating KiCad netlist, click on the `KiCad-Ngspice converter` button. This will open converter window where you can enter details of Analysis, Source values and Device library.



Transient Analysis

Start Time	1	ms
Step Time	100	us
Stop Time	5	ms

(a) Phase Shift Oscillator Analysis

Add parameters for DC source v1

Enter value(Volts/Amps): 22

(b) Phase Shift Oscillator Details

Add library for Transistor q1 : npn

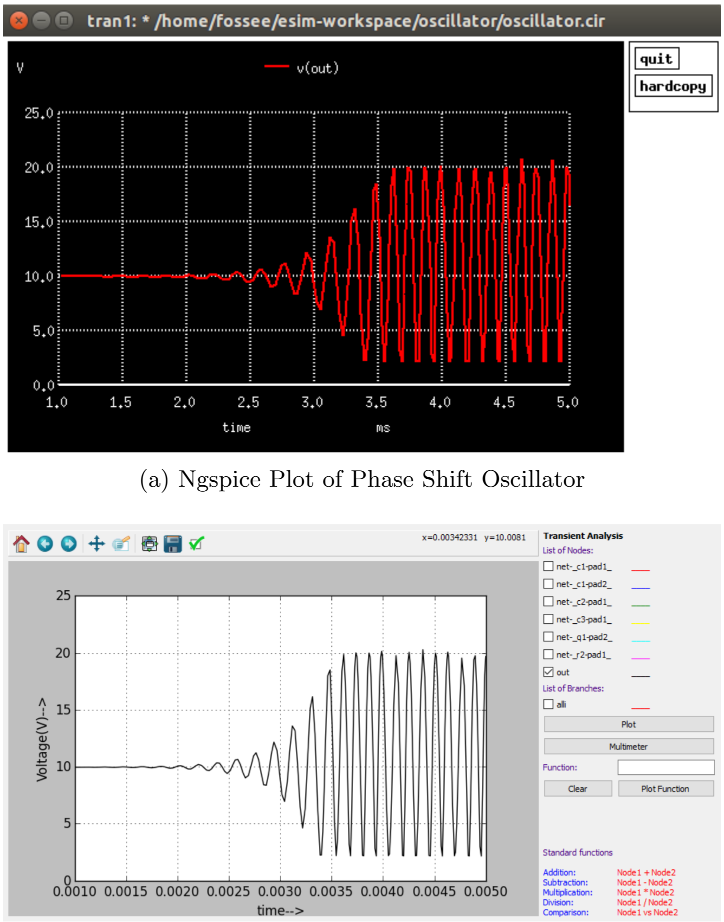
C:/eSim/src/deviceModelLibrary/Transistor/NPN.lib

(c) Phase Shift Oscillator Device Modeling

Figure 11.33: Analysis, Source and Device Tab

Under device library you can add the library for diode used in the circuit. If you do not add any library it will take default Ngspice model.

- Simulation: Once the KiCad-Ngspice converter runs successfully, you can run simulation by clicking the simulation button in the toolbar.



(a) Ngspice Plot of Phase Shift Oscillator

(b) Python Plot of Phase Shift Oscillator

Figure 11.34: Phase Shift Oscillator Simulation Output

11.1.7 Characteristics of BJT in Common Base Configuration

Problem Statement:

Plot Characteristics of BJT in Common Base Configuration.

Solution:

The new project is created by clicking the `New` icon on the menubar. The name of the project is given in the window shown in Fig. 11.1.

- Creating Schematic: To create the schematic, click the very first icon of the left toolbar as shown in the Fig. 11.2. This will open KiCad Eeschema.

After the KiCad window is opened, to create a schematic we need to place the required components. Fig. 11.3 shows the icon on the right toolbar which opens the component library.

After all the required components of the simple Half Wave rectifier circuits are placed, wiring is done using the `Place Wire` option as shown in the Fig. 11.4

Next step is `ERC (Electric Rules Check)`. Fig. 11.5 shows the icon for `ERC`. After completing all the above steps the BJT in CB Configuration schematic will look like Fig. 11.35.

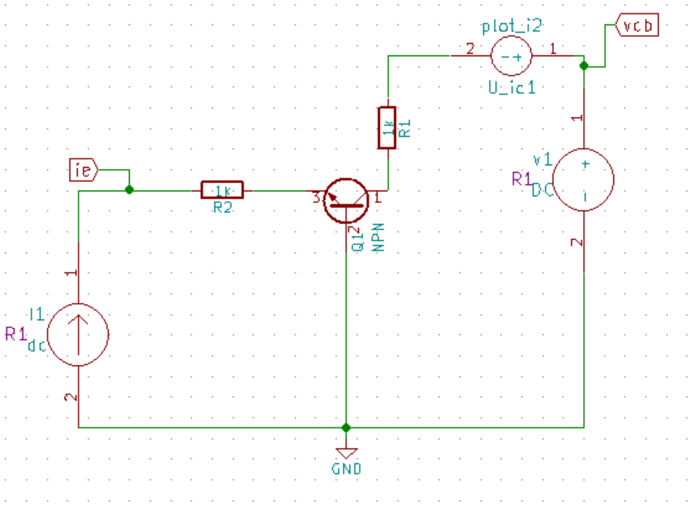
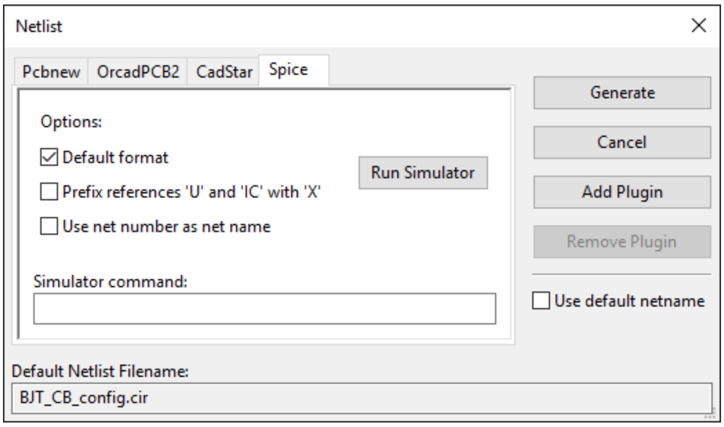


Figure 11.35: Schematic of BJT in CB Configuration circuit

KiCad netlist is generated as shown in the Fig. 11.36



Options:

☑ Default format

☐ Prefix references 'U' and 'IC' with 'X'

☐ Use net number as net name

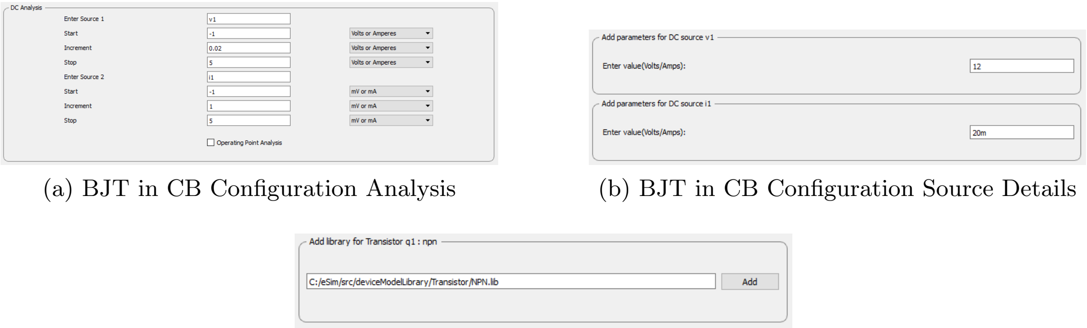
Simulator command:

☐ Use default netname

Default Netlist Filename: BJT_CB_config.cir

Figure 11.36: BJT in CB Configuration circuit Netlist Generation

- Convert KiCad to Ngspice: After creating KiCad netlist, click on the `KiCad-Ngspice converter` button. This will open converter window where you can enter details of Analysis, Source values and Device library.



DC Analysis

Enter Source 1: v1

Start: -1 Volts or Amperes

Increment: 0.02 Volts or Amperes

Stop: 5 Volts or Amperes

Enter Source 2: i1

Start: -1 mV or mA

Increment: 1 mV or mA

Stop: 5 mV or mA

☐ Operating Point Analysis

(a) BJT in CB Configuration Analysis

Add parameters for DC source v1

Enter value(Volts/Amps): 12

Add parameters for DC source i1

Enter value(Volts/Amps): 20m

(b) BJT in CB Configuration Source Details

Add library for Transistor q1 : npn

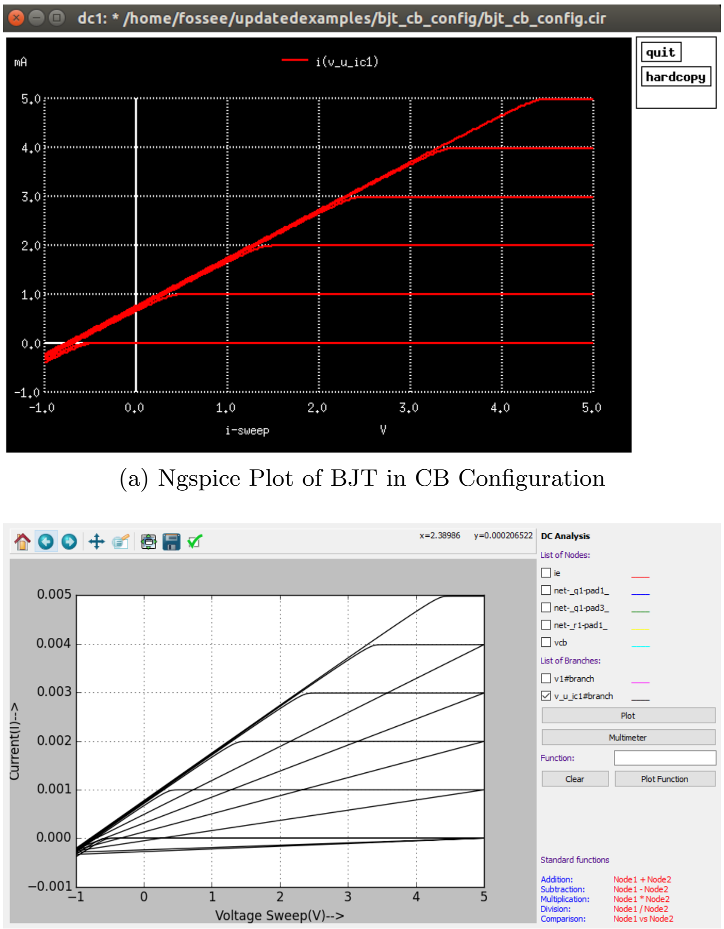
C:/eSim/src/deviceModelLibrary/Transistor/NPN.lib

(c) BJT in CB Configuration Device Modeling

Figure 11.37: Analysis, Source and Device Tab

Under device library you can add the library for diode used in the circuit. If you do not add any library it will take default Ngspice model.

- Simulation: Once the KiCad-Ngspice converter runs successfully, you can run simulation by clicking the simulation button in the toolbar.



(a) Ngspice Plot of BJT in CB Configuration

(b) Python Plot of BJT in CB Configuration

Figure 11.38: BJT in CB Configuration Simulation Output

Chapter 12

PCB Design

Printed Circuit Board (PCB) design is an important step in electronic system design. Every component of the circuit needs to be placed and connections routed to minimise 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 associating footprints to all components, placing them appropriately to minimise 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 eSim.

12.1 Schematic creation for PCB design

In Chapter 5, we will see the differences between schematic for simulation and schematic for PCB design. Let us design the PCB for a RC circuit. A resistor, capacitor, ground, power flag and a connector are required. Connectors are used to take signals in and out of the PCB.

Create the circuit schematic as shown in Fig. 12.1. The two pin connector (*CONN_2*) can be placed from the Eeschema library *conn*. Do the annotation and test for ERC. Refer to Chapter 5 to know more about basic steps in schematic creation.

12.1.1 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. In the Netlist window, under the tab *Pcbnew*, click on the button *Netlist*. This is shown in Fig. 12.2. 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. Save the schematic and close the schematic editor. *Note that the netlist for PCB has an extension* `.net`. *The netlist created for simulation has an extension* `.cir`.

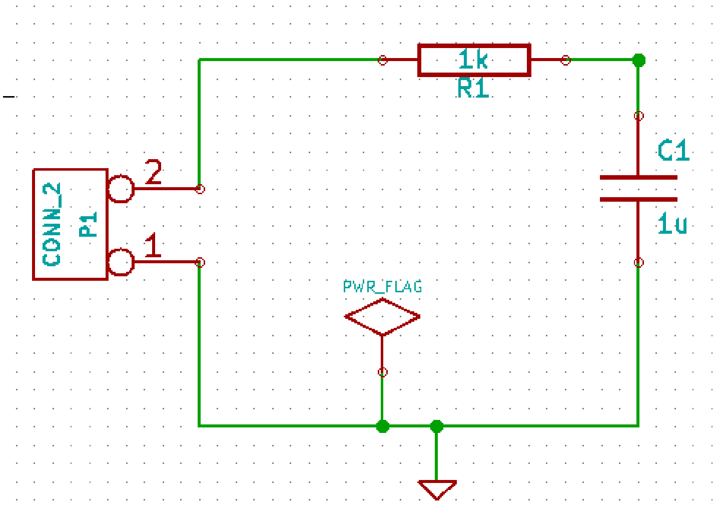
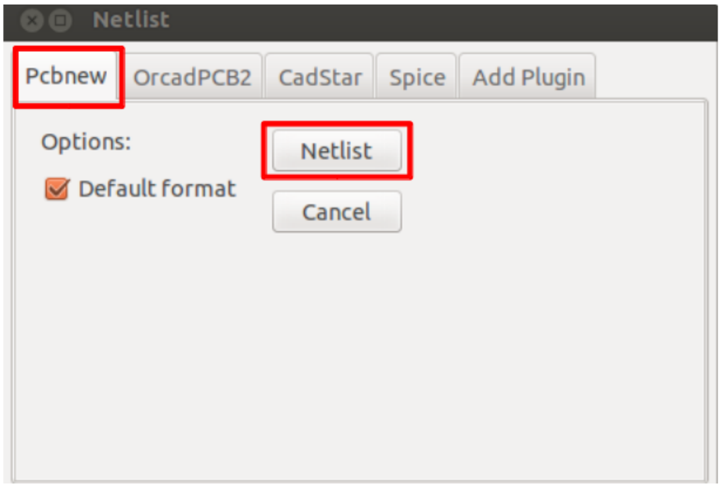


Figure 12.1: Final circuit schematic for RC low pass circuit



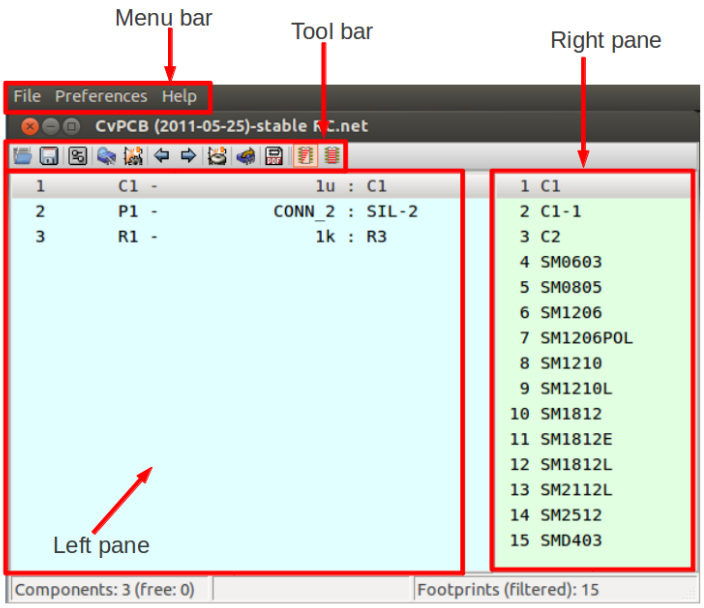
Options:

☑ Default format

Figure 12.2: Netlist generation for PCB

12.1.2 Mapping of components using Footprint Editor

Once the netlist for PCB is created, one needs to map each component in the netlist to a footprint. The tool *Footprint Editor* is used for this. eSim uses `CvPcb` as its footprint editor. `CvPcb` is the footprint editor tool in KiCad.



```
1        C1  -              1u  :  C1
2        P1  -          CONN_2  :  SIL-2
3        R1  -              1k  :  R3
```

```
 1  C1
 2  C1-1
 3  C2
 4  SM0603
 5  SM0805
 6  SM1206
 7  SM1206POL
 8  SM1210
 9  SM1210L
10  SM1812
11  SM1812E
12  SM1812L
13  SM2112L
14  SM2512
15  SMD403
```

Components: 3 (free: 0)

Footprints (filtered): 15

Figure 12.3: Footprint editor with the menu bar, toolbar, left pane and right pane marked

12.1.3 Familiarising the Footprint Editor tool

If one opens the *Footprint Editor* after creating the `.net` netlist file, the Footprint editor as shown in Fig. 12.3 will be obtained. The menu bar and toolbars 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 the Footprint Editor is opened before creating a '.net' file, then the left and right panes will be empty.*

Toolbar

Some of the important tools in the toolbar are shown in Fig. 12.4. They are explained below:

1. Save netlist and footprint files - Save the netlist and the footprints that are associated with it.
2. View selected footprint - View the selected footprint in 2D. See Sec. 12.1.4 for more details.
3. Automatic footprint association - Perform footprint association for each component automatically. Footprints will be selected from the list of footprints available.
4. Delete all associations - Delete all the footprint associations made

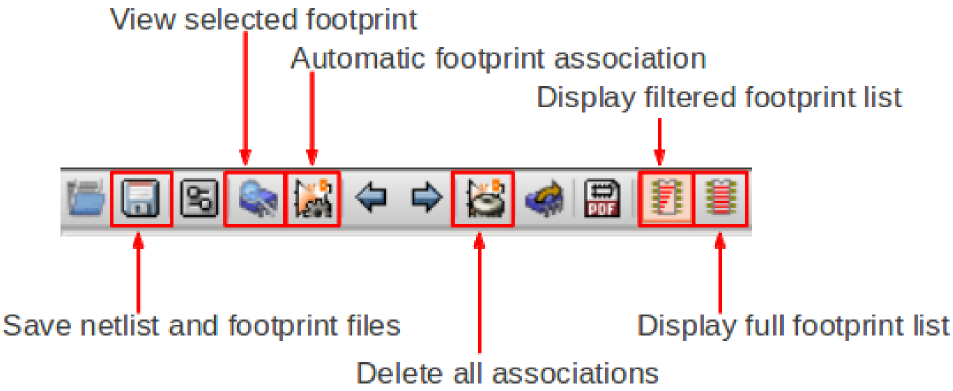

View selected footprint
Automatic footprint association
Display filtered footprint list
Save netlist and footprint files
Display full footprint list
Delete all associations

Figure 12.4: Some important tools in the toolbar

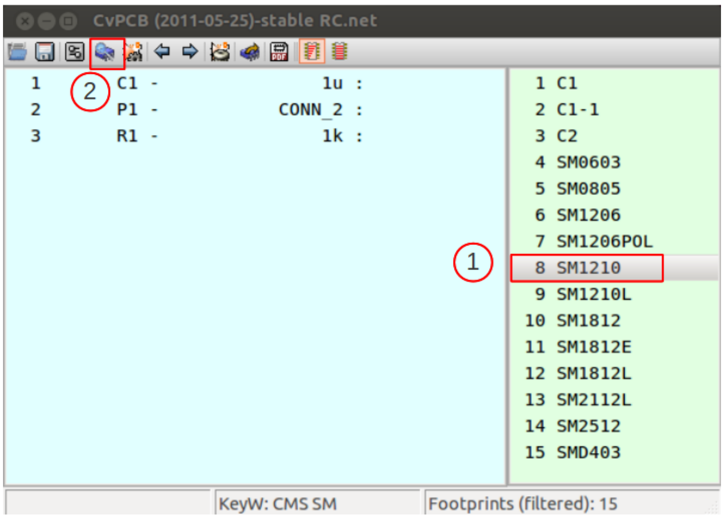
5. Display filtered footprint list - Display a filtered list of footprints suitable to the selected component
6. Display full footprint list - Display the list of all footprints available (without filtering)

12.1.4 Viewing footprints in 2D and 3D

To view a footprint in 2D, select it 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 Fig. 12.5. On clicking the *View selected footprint* tool, the `Footprint` window with the view in 2D will be displayed. Click on the *3D* tool in the `Footprint` window, as shown in Fig. 12.6. A top view of the selected footprint in 3D is obtained. 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 Fig. 12.7.

12.1.5 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 Fig. 12.8. Save the footprint association by clicking on the *Save netlist and footprint files* tool from the `CvPcb` toolbar. The `Save Net and component List` window appears. Browse to the directory where the schematic file for this project is saved and click on *Save*. The netlist gets saved and the *Footprint Editor* window closes automatically. *Note that one needs to browse to the directory where the schematic file is saved and save the '.net' file in the same directory.*



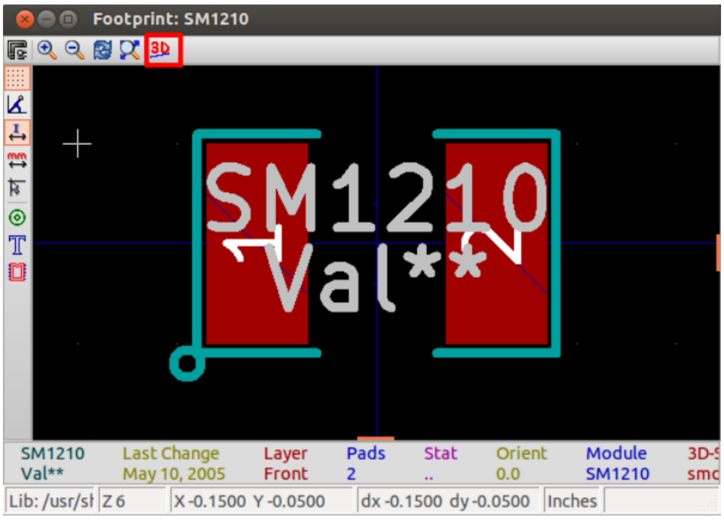
CvPCB (2011-05-25)-stable RC.net

```
1   C1 -         1u :
2   P1 -     CONN_2 :
3   R1 -         1k :
```

```
 1 C1
 2 C1-1
 3 C2
 4 SM0603
 5 SM0805
 6 SM1206
 7 SM1206POL
 8 SM1210
 9 SM1210L
10 SM1812
11 SM1812E
12 SM1812L
13 SM2112L
14 SM2512
15 SMD403
```

KeyW: CMS SM | Footprints (filtered): 15

Figure 12.5: Viewing footprint for SM1210: 1. Choose the footprint SM1210 from the right pane, 2. Click on *View selected footprint*



Footprint: SM1210

SM1210	Last Change	Layer	Pads	Stat	Orient	Module	3D-S
Val**	May 10, 2005	Front	2	..	0.0	SM1210	smc

Lib: /usr/sl | Z 6 | X -0.1500 Y -0.0500 | dx -0.1500 dy -0.0500 | Inches

Figure 12.6: Footprint view in 2D. Click on *3D* to get 3D view

12.2 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. eSim uses `Pcbnew`, the layout creation tool in KiCad, as its layout editor.

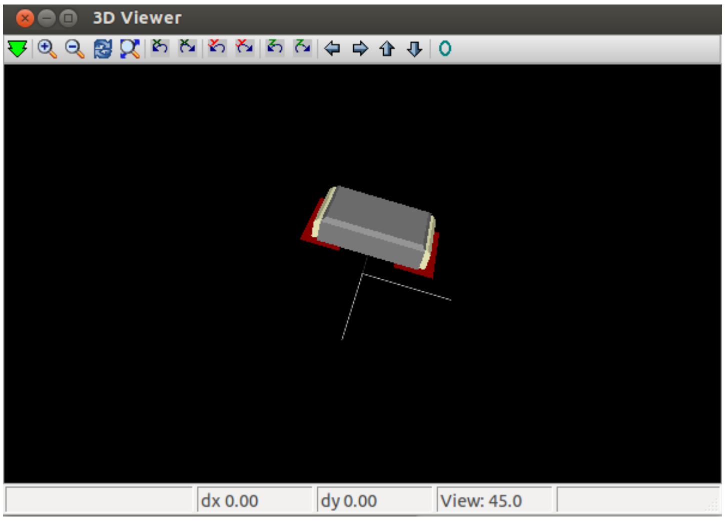
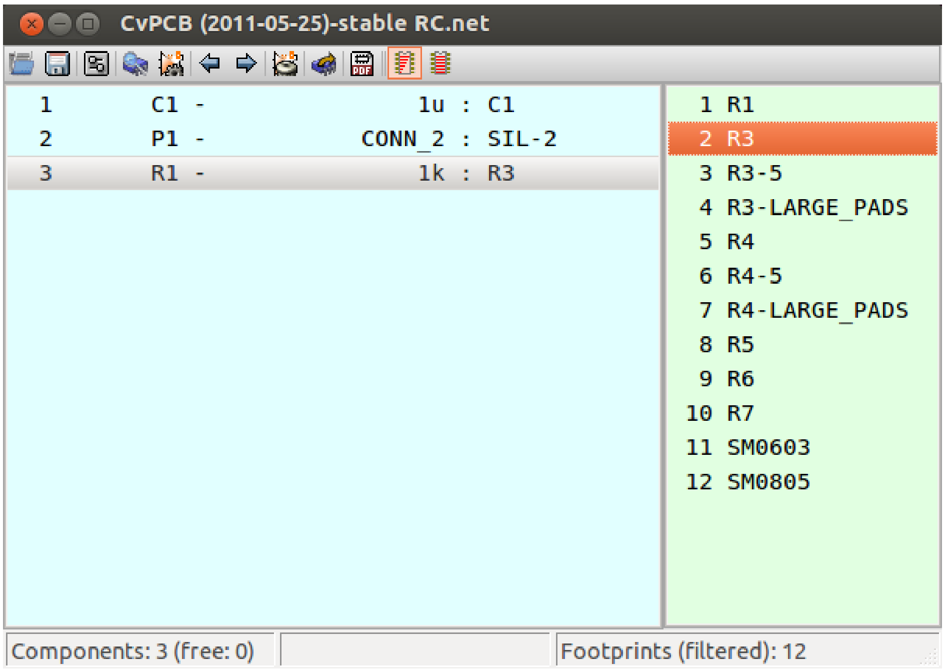


Figure 12.7: Side view of the footprint in 3D


CvPCB (2011-05-25)-stable RC.net

```
1        C1  -           1u  :  C1
2        P1  -       CONN_2  :  SIL-2
3        R1  -           1k  :  R3
```

```
 1  R1
 2  R3
 3  R3-5
 4  R3-LARGE_PADS
 5  R4
 6  R4-5
 7  R4-LARGE_PADS
 8  R5
 9  R6
10  R7
11  SM0603
12  SM0805
```

Components: 3 (free: 0)

Footprints (filtered): 12

Figure 12.8: Footprint mapping done

12.2.1 Familiarizing the Layout Editor tool

The layout editor with the various menu bar and toolbars is shown in Fig. 12.9.

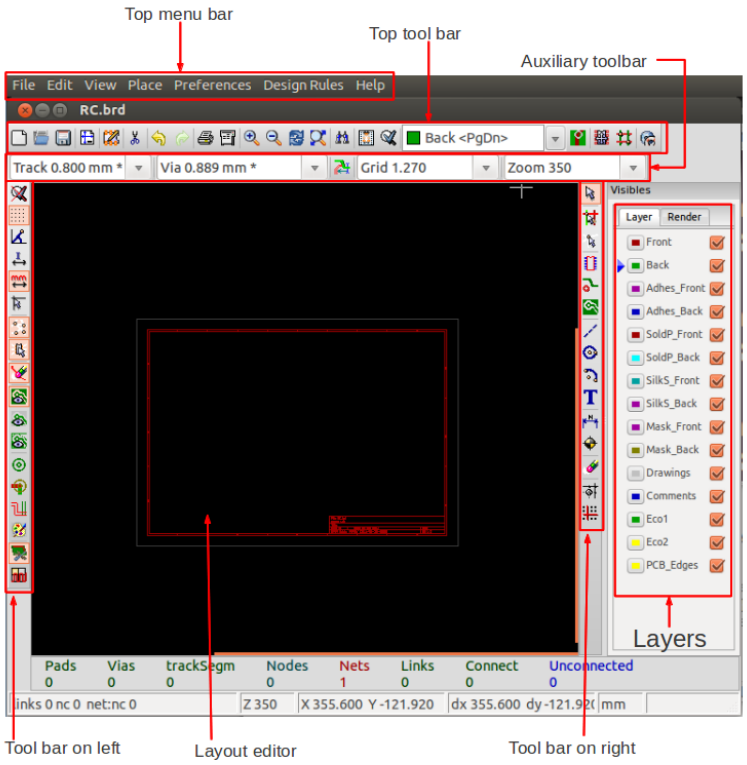


Figure 12.9: Layout editor with menu bar, toolbars and layer options marked

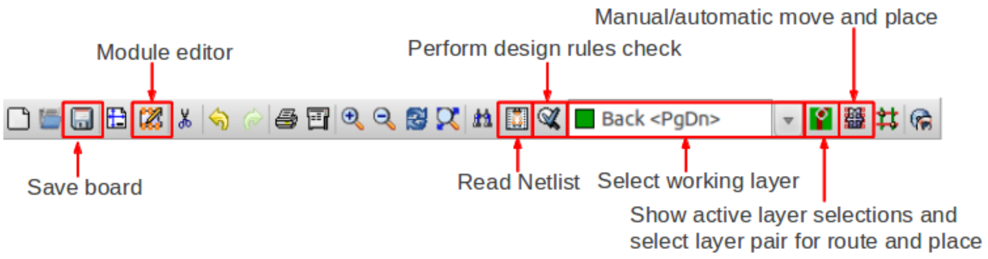

Manual/automatic move and place
Module editor
Perform design rules check
Back <PgDn>
Save board
Read Netlist
Select working layer
Show active layer selections and select layer pair for route and place

Figure 12.10: Top toolbar with important tools marked

Top toolbar

Some of the important menu options in the top menu bar are shown in Fig. 12.10. 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.

4. Perform design rules check - Check for design rules, unconnected nets, etc., in the layout.
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 the currently active layer selections.
7. Mode footprint: Manual/automatic move and place - Move and place modules

12.2.2 Hotkeys

A list of hotkeys are given below:

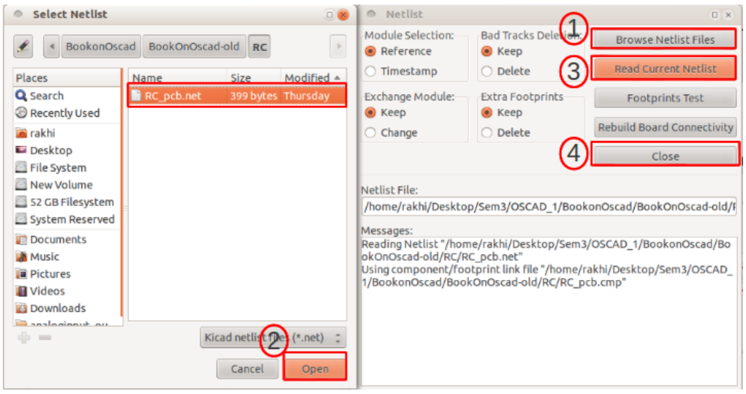
1. F1 - Zoom in
2. F2 - Zoom out
3. Delete - Delete Track or Footprint
4. X - Add new track
5. V - Add Via
6. M - Move Item
7. F - Flip Footprint
8. R - Rotate Item
9. G - Drag Footprint
10. Ctrl+Z - Undo
11. E - Edit Item

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

12.2.3 PCB design example using RC circuit

Click on *Layout Editor* from the eSim toolbar. Click on *Read Netlist* tool from the top toolbar. Click on *Browse Netlist files* on the Netlist window that opens up. Select the `.net` file that was modified 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_pcb.net has been read. The sequence of operations is shown in Fig. 12.11. The footprint modules will now be imported to the top left hand corner of the layout editor window. This is shown in Fig. 12.12. Zoom in to the top left corner by pressing the key `F1` or using the scroll button of the mouse. The zoomed in version of the imported netlist is shown in Fig. 12.13.

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 toolbar. 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 Fig. 12.14. Click on *Yes* on the confirmation window to move the modules. Zoom in using the F1 key. The current placement of components after zooming in is shown in Fig. 12.15a. We need to arrange the modules properly to lay tracks. Rotate the connector P1 by



Name	Size	Modified
RC_pcb.net	399 bytes	Thursday

Module Selection: ● Reference ○ Timestamp

Bad Tracks Deletion: ● Keep ○ Delete

Exchange Module: ● Keep ○ Change

Extra Footprints: ● Keep ○ Delete

Netlist File:
/home/rakhi/Desktop/Sem3/OSCAD_1/BookonOscad/BookOnOscad-old/I

Messages:
```
Reading Netlist "/home/rakhi/Desktop/Sem3/OSCAD_1/BookonOscad/Bo
okOnOscad-old/RC/RC_pcb.net"
Using component/footprint link file "/home/rakhi/Desktop/Sem3/OSCAD_
1/BookonOscad/BookOnOscad-old/RC/RC_pcb.cmp"
```

Figure 12.11: Importing netlist file to layout editor: 1. Browse netlist Files, 2. Choose the RC_pcb.net file, 3. Read Netlist file, 4. Close

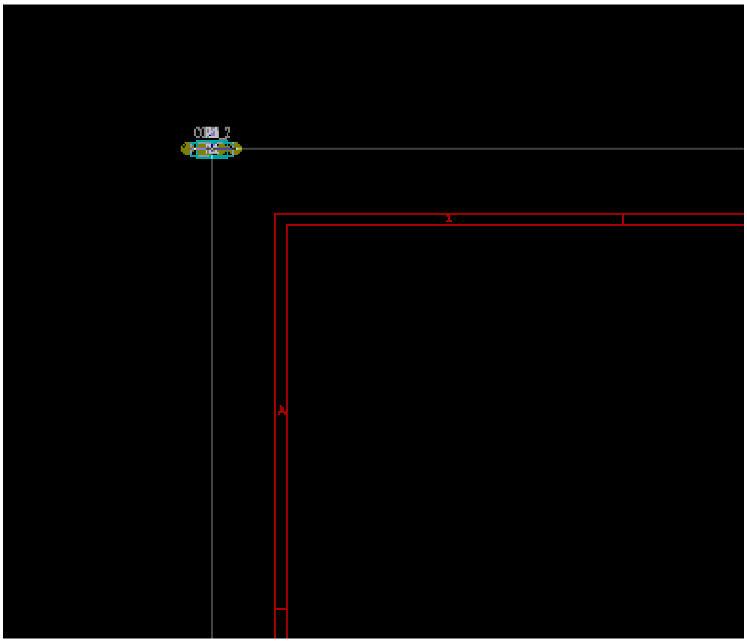


Figure 12.12: Footprint modules imported to top left corner of layout editor window

placing the cursor on top of P1 and pressing R. Move it by placing the cursor on top of it and pressing M. The final placement is shown in Fig. 12.15b.

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 Fig. 12.16. The *Design Rules Editor* window opens up. Here one 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. Fig. 12.17

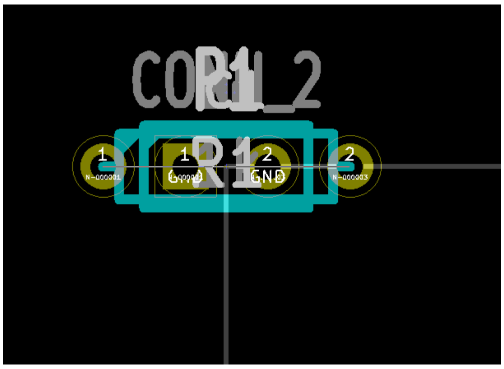


Figure 12.13: Zoomed in version of the imported netlist

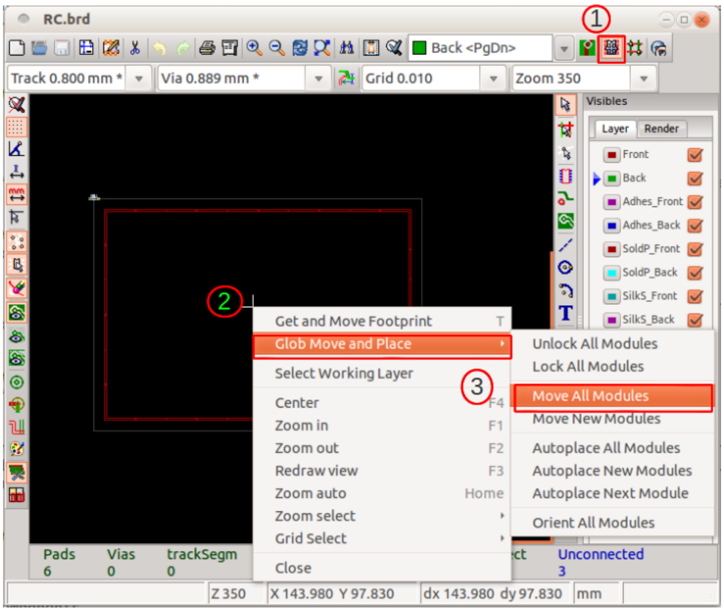
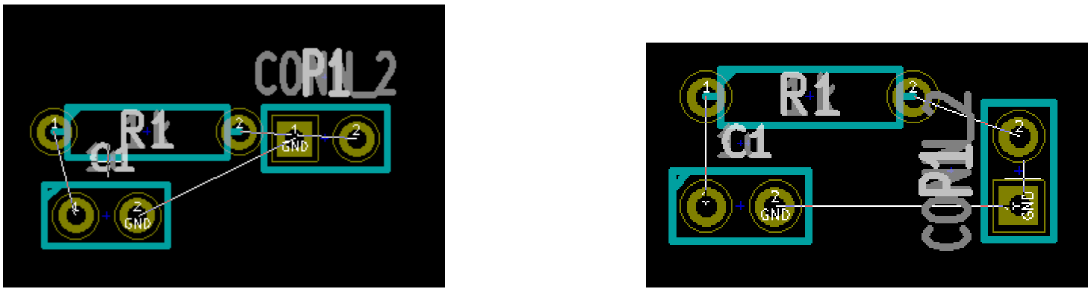


Figure 12.14: Moving and placing modules to the center of layout editor. 1. Click on *Mode footprint: Manual/automatic move and place*, 2. Place cursor at center of layout editor and right click on it 3. Choose *Glob Move and Place* and then choose *Move All Modules.*

shows the sequence of operations.

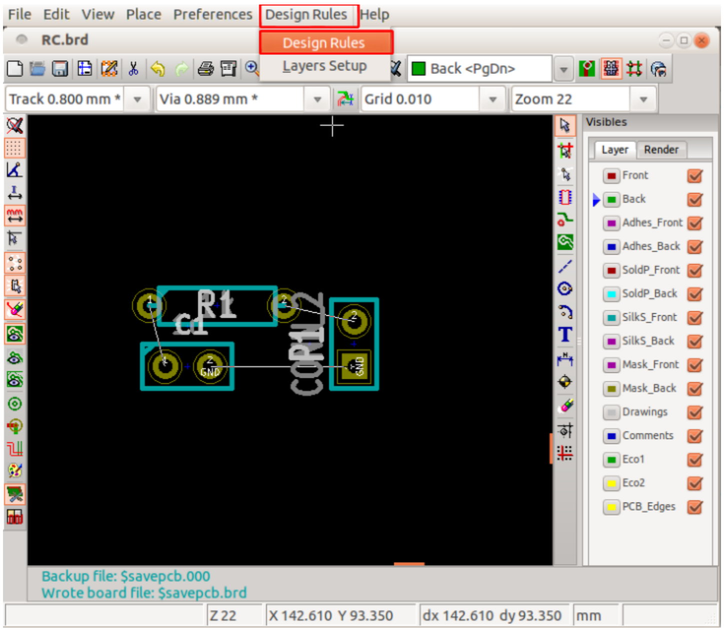
Click on *Back* from the *Layer* options as shown in Fig. 12.18. Let us now start laying the tracks. Place the cursor above the left terminal of R1 in the layout editor window. Press the key `x`. Move the cursor down and double click on the left terminal of C1. A track is formed. This is shown in Fig. 12.19a. Similarly lay the track between



(a) Zoomed in version of the current placement after moving modules to the center of the layout editor

(b) Final placement of footprints after rotating and moving P1

Figure 12.15: Different stages of placement of modules on PCB



Track 0.800 mm *
Via 0.889 mm *
Grid 0.010
Zoom 22

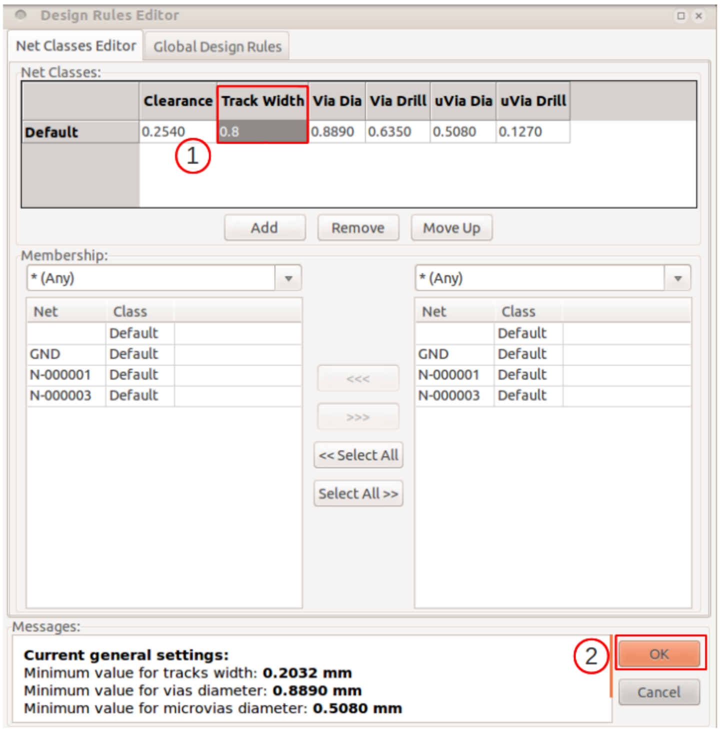
Backup File: $savepcb.000
Wrote board file: $savepcb.brd

Z 22 | X 142.610 Y 93.350 | dx 142.610 dy 93.350 | mm

Figure 12.16: Choose *Design Rules* from the top menu bar and *Design Rules* again

capacitor C1 and connector P1 as shown in Fig. 12.19b. 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 the connector. The track will be laid as shown in Fig. 12.19c. All tracks are now laid. The next step is to create PCB edges.

Choose *PCB_edges* from the *Layer* options to add edges. Click on *Add graphic line or polygon* from the toolbar on the left. Fig. 12.20 shows the sequence of operations.



	Clearance	Track Width	Via Dia	Via Drill	uVia Dia	uVia Drill
Default	0.2540	0.8	0.8890	0.6350	0.5080	0.1270

Net	Class
	Default
GND	Default
N-000001	Default
N-000003	Default

Net	Class
	Default
GND	Default
N-000001	Default
N-000003	Default

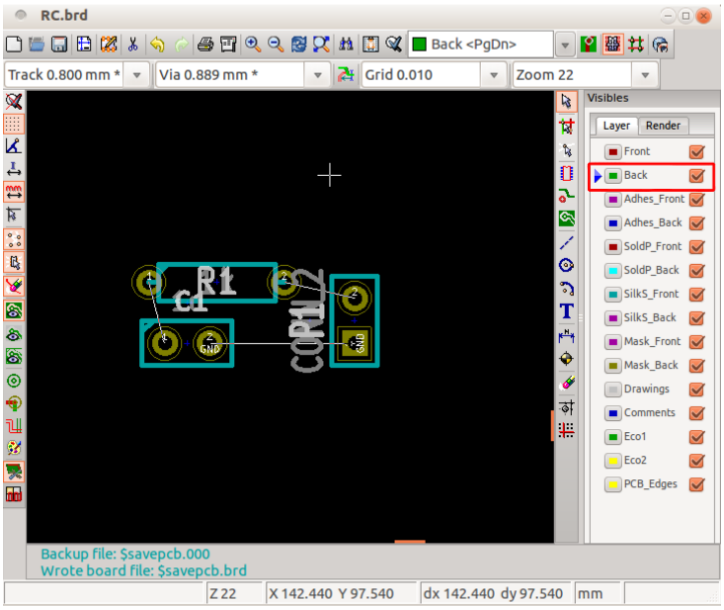
Current general settings:
Minimum value for tracks width: **0.2032 mm**
Minimum value for vias diameter: **0.8890 mm**
Minimum value for microvias diameter: **0.5080 mm**

Figure 12.17: Changing the track width: 1. Double click on *Track Width* field and type 0.8, 2. Click on *OK*

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 Fig. 12.21. Double click to finish drawing the edges.

Click on *Perform design rules check* from the top toolbar 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. Fig. 12.22 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 Fig. 12.23. The plot window opens up. One can choose which layers to plot by selecting/deselecting them from the `Layers` pane on the left side. One can also choose the format used to plot them. 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 the Gerber files are created. Click on *Close*. This is shown in Fig. 12.24. The PCB design of RC circuit is now complete.



RC.brd

Back <PgDn>

Track 0.800 mm *

Via 0.889 mm *

Grid 0.010

Zoom 22

Backup file: $savepcb.000
Wrote board file: $savepcb.brd

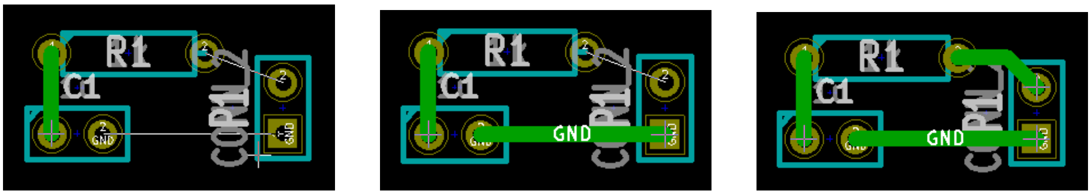
Z 22

X 142.440 Y 97.540

dx 142.440 dy 97.540

mm

Figure 12.18: Choosing the copper layer *Back*

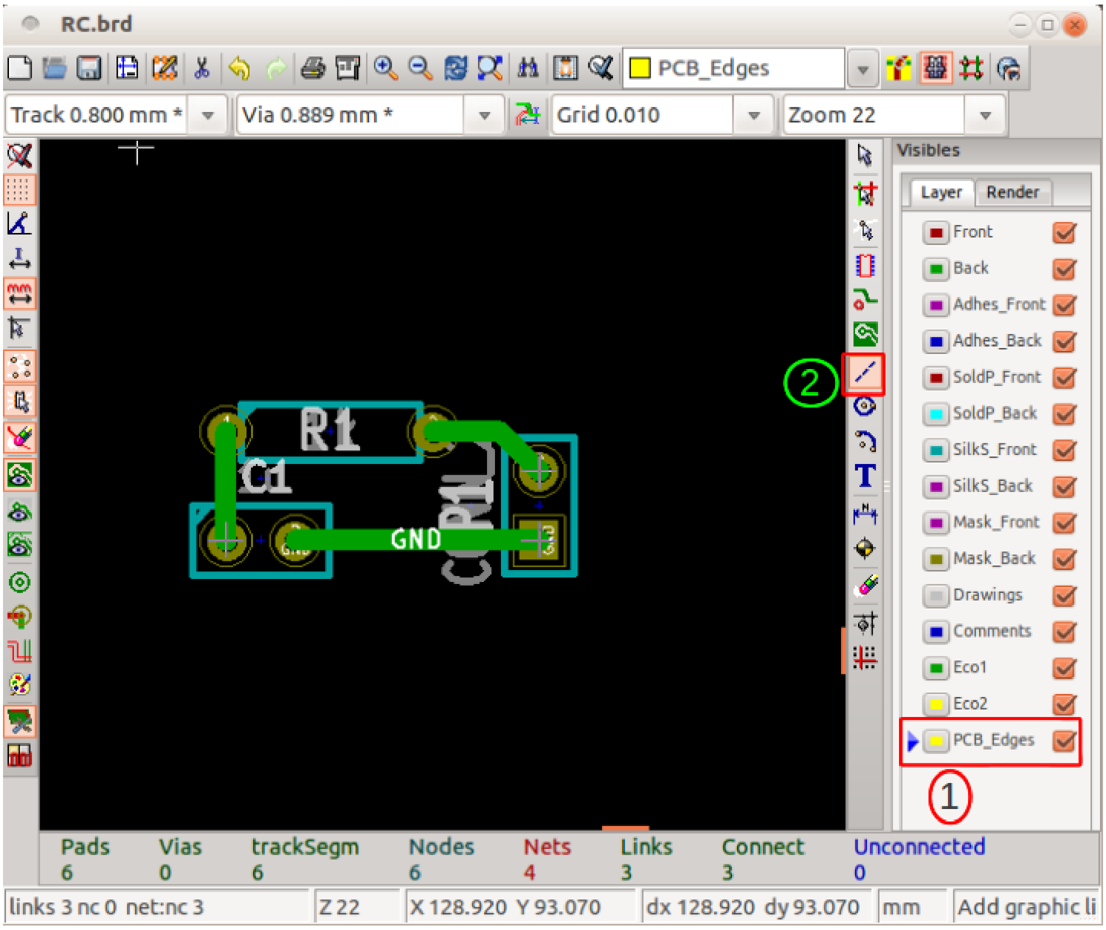


(a) A track formed between resistor and capacitor

(b) A track formed between capacitor and connector

(c) A track formed between connector and resistor

Figure 12.19: Different stages of laying tracks during PCB design



Track 0.800 mm * | Via 0.889 mm * | Grid 0.010 | Zoom 22

Pads	Vias	trackSegm	Nodes	Nets	Links	Connect	Unconnected
6	0	6	6	4	3	3	0

links 3 nc 0 net:nc 3 | Z 22 | X 128.920 Y 93.070 | dx 128.920 dy 93.070 | mm | Add graphic li

Figure 12.20: Creating PCB edges: 1. Choose *PCB_Edges* from *Layer* options 2. Choose *Add graphic line or polygon* from left toolbar

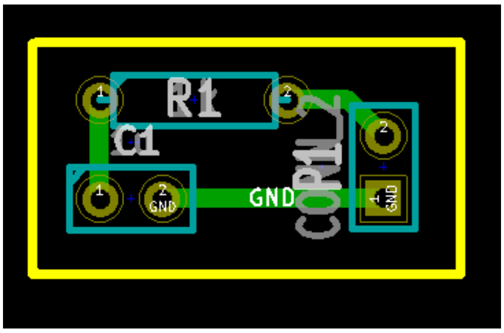
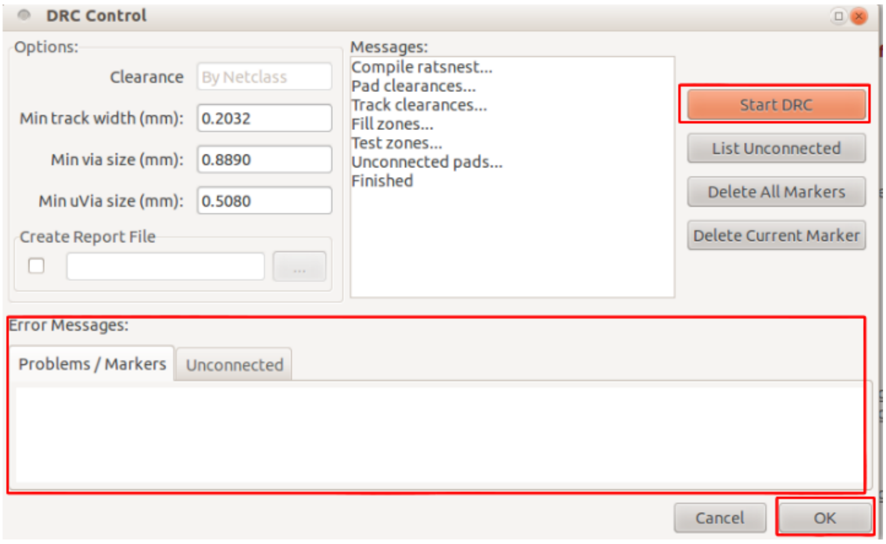


Figure 12.21: PCB edges drawn



Clearance: By Netclass

Min track width (mm): 0.2032

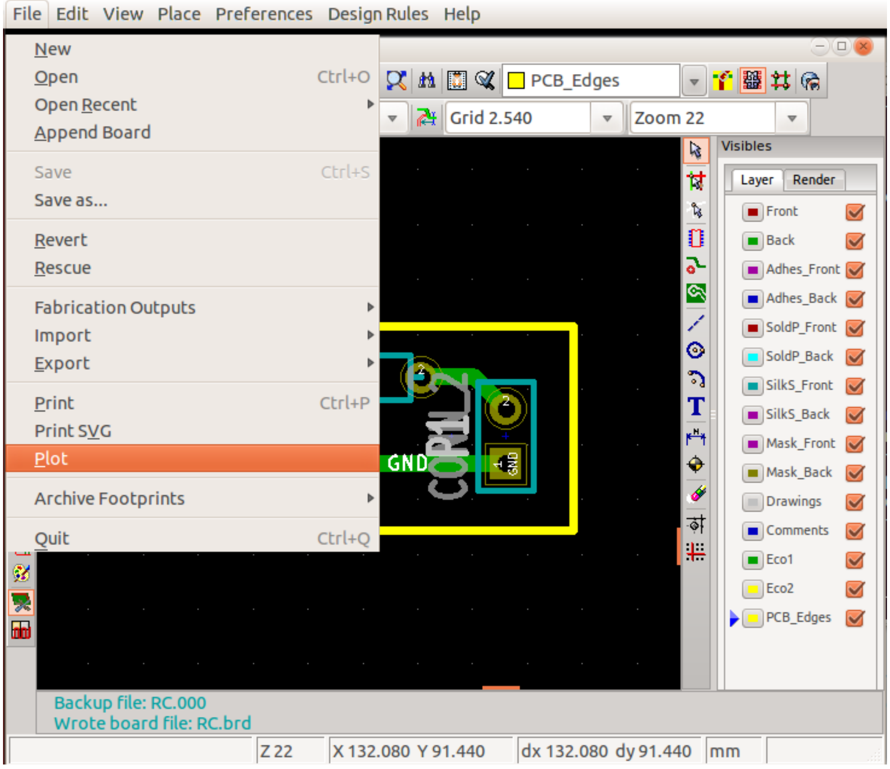
Min via size (mm): 0.8890

Min uVia size (mm): 0.5080

Create Report File ☐

```
Compile ratsnest...
Pad clearances...
Track clearances...
Fill zones...
Test zones...
Unconnected pads...
Finished
```

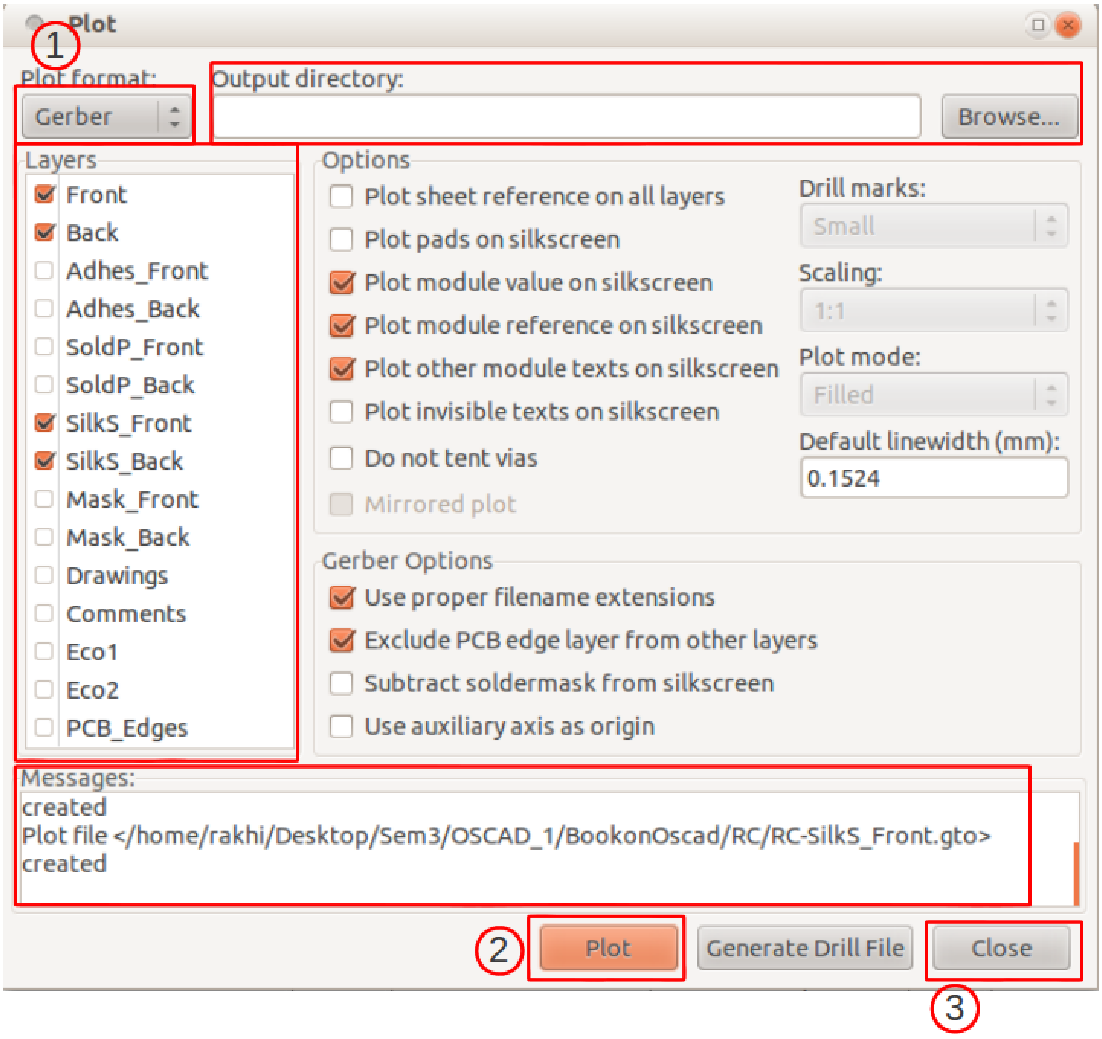
Figure 12.22: Performing design rules check: 1. Click on *Start DRC*, 2. Click on *Ok*



```
Backup file: RC.000
Wrote board file: RC.brd
```

Z 22 | X 132.080 Y 91.440 | dx 132.080 dy 91.440 | mm

Figure 12.23: Choosing *Plot* from the *File* menu



Plot format: Gerber

Output directory:

Layers
- ☑ Front
- ☑ Back
- ☐ Adhes_Front
- ☐ Adhes_Back
- ☐ SoldP_Front
- ☐ SoldP_Back
- ☑ SilkS_Front
- ☑ SilkS_Back
- ☐ Mask_Front
- ☐ Mask_Back
- ☐ Drawings
- ☐ Comments
- ☐ Eco1
- ☐ Eco2
- ☐ PCB_Edges

Options
- ☐ Plot sheet reference on all layers
- ☐ Plot pads on silkscreen
- ☑ Plot module value on silkscreen
- ☑ Plot module reference on silkscreen
- ☑ Plot other module texts on silkscreen
- ☐ Plot invisible texts on silkscreen
- ☐ Do not tent vias
- ☐ Mirrored plot

Drill marks: Small

Scaling: 1:1

Plot mode: Filled

Default linewidth (mm): 0.1524

Gerber Options
- ☑ Use proper filename extensions
- ☑ Exclude PCB edge layer from other layers
- ☐ Subtract soldermask from silkscreen
- ☐ Use auxiliary axis as origin

Messages:

```
created
Plot file </home/rakhi/Desktop/Sem3/OSCAD_1/BookonOscad/RC/RC-SilkS_Front.gto>
created
```

Figure 12.24: Creating Gerber files: 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*