SYSTEM REQUIREMENTS:

Start with a Linux system with version: “Ubuntu 22.04.4 LTS” fresh

(Kindly note: Dotted points are commands to be run in the terminal)**

STEP 1: System Updation

It is necessary to update the system, for this, open a terminal in the desktop/home and give command:

- `sudo apt-get update`

STEP 2: SPECIFIC FOLDER CREATION

Now, we will be installing all the required tools in a new folder and to create that folder type:

- `mkdir -p <Installation_directory_name>`

Now, change the directory from desktop to the newly created folder by running the command

- `cd <Installation_directory_name>`

```
(Example: mkdir -p chip_design

          cd chip_design)
```

STEP 3:PRE-REQUISITES

Next step is to install all the necessary supporting libraries for the installation of open source tools for Ubuntu LTS version

```
#pre-requisite
```

- `sudo apt install git`

- sudo apt-get install build-essential clang bison flex
- sudo apt-get install libreadline-dev gawk tcl-dev libffi-dev
- sudo apt-get install git graphviz xdot pkg-config python3
- sudo apt-get install libboost-system-dev libboost-python-dev
- sudo apt-get install libboost-filesystem-dev zlib1g-dev make m4
- sudo apt-get install tcsh csh libx11-dev gperf tcl8.6-dev tk8.6
- sudo apt-get install tk8.6-dev libxmp4 libxpm-dev libxcb1
- sudo apt-get install libcairo2 libxrender-dev libx11-xcb-dev
- sudo apt-get install libxaw7-dev freeglut3-dev automake yosys
- sudo apt-get update
- sudo apt-get -y install libtool

Check whether .local/bin is included in $PATH

Check it is with this command:

- echo $PATH | grep '\.local'

If it’s not there, then add it to your path before starting the install by adding this to your `~/.bashrc` file:

- export PATH="$HOME/.local/bin:$PATH"

For analog chip design: Magic, Xschem, Open pdks, xterm ngspice

STEP 4: INSTALLING XSCHEM TOOL

Once the installation of all the prerequisites is done, we can proceed with installation of individual tools one by one. First, we are going to install xschem. For that, open a terminal in chip_design folder and clone the required repository from github by running the following commands:

- `cd ~/<Installation_directory_name>/`
- `git clone https://github.com/StefanSchippers/xschem.git xschem`
- `cd xschem`
- `./configure`
- `make`
- `sudo make install`

`To run xschem`

- `xschem`

STEP 5: INSTALLING MAGIC TOOL

Next we will be installing magic, for that a terminal is opened in our installation folder and run the commands:

- `cd ~/<Installation_directory_name>/`
- `git clone git://opencircuitdesign.com/magic`
- `cd magic`
- `./configure`
- `make`
- `sudo make install`

STEP 6: INSTALLING NGSPICE TOOL

Next is the installation of ngspice, before installing ngspice, just update the system by : `sudo apt-get update.` Thereafter run the following commands:

- `wget https://sourceforge.net/projects/ngspice/files/ng-spice-rework/43/ngspice-43.tar.gz/download`

(Once the zip file is downloaded using above command extract the ngspice-43 folder)

To extract:

If the unzip command isn't already installed on your system (use which unzip to check), then run:

- **`sudo apt-get install unzip`**

After installing the unzip utility, if you want to extract to a particular destination folder, you can use:

- **`unzip file.zip -d destination_folder`**

If you want to extract to a directory with the same name as the zip in your current working directory, you can simply do:

- **`unzip file.zip`**

and then run the following commands

- `cd ngspice-43`
- `./configure`
- `make`
- `sudo make install`

When successful, check the version using:

- `ngspice -version`

STEP 7: INSTALLING OPEN_PDKS

Next comes the most time consuming part of the installation process, ie., the installation of open_pdks. Once again start by updating the system : `sudo apt-get update.` Thereafter run the following commands:

- `git clone git://opencircuitdesign.com/open_pdks`
- `cd open_pdks`
- `./configure --enable-sky130-pdk`
- `sudo make`
- `sudo make install`

Copy the .magicrc file and paste it in the magic directory

- `Cd ~`
- `cd ~/.xschem`

-
  ```
  cp
  ~/<Installation_directory_name>/open_pdks/sky130/sky130A/libs.tech/xs
  /xschemrc
  ```
- `cd ~`

-
  ```
   cp
  ~/<Installation_directory_name>/open_pdks/sky130/sky130A/libs.tech/ma
  sky130A.magicrc ~/.magicrc
  ```
- `sudo apt -y install vim-gtk3 xterm`

  ```
  Or
  sudo apt-get -y install xterm
  sudo apt-get -y install vim-gtk
  ```

STEP 8: INSTALLING LVS

Next install LVS tool to perform Layout versus schematic check using the command:

- `sudo apt-get install netgen-lvs`

STEP 8: INSTALLING KLAYOUT

Next install the tool Klayout to have a look at the GDS file using command:

- `sudo apt-get install klayout`

ALTERNATIVE:

If not installed properly using the above try the alternative way as:

Download the latest Klayout from: [Klayout]

KLayout system requirements are:

a) **C++**: clang (tested on 3.8.0 and many more) or gcc (tested on 4.6.3 and many more).
b) **Qt**: Qt4 >= 4.8.5 (Qt 5 preferred, Qt 6 supported).
c) **Ruby**: >= 1.9.x (Ruby 2 or 3 preferred).
d) **Python**: >= 3.x.

Install using:

First install pre-requisites

(# Kindly note: Depending upon system the missing packages may appear, so installing them could help in installing klayout)

- `sudo apt install gcc ruby-full`
- `sudo apt-get install libqt5opengl5 libhttp-parser2.9 libmbedcrypto7 libmbedtls14 libmbedx509-1 libssh2-1 libgit2-1.1`
- `sudo dpkg -i klayout_0.29.5-1_amd64.deb`

#(If it still indicates that a package is missing, install the required packages.)

#(If it pops message as "apt --fix-broken install")

Give command :

- `sudo apt --fix-broken install`

and rerun

- `sudo dpkg -i klayout_0.29.5-1_amd64.deb`

Check Klayout using command

- `klayout`

(For digital chip design: Openlane, Iverilog, gtkwave, klayout)

STEP 1: INSTALLING OPENLANE

In order to generate the GDS file for our digital design, we need to have openlane installed. For the same docker needs to be installed. Use the following commands to install and run docker tests.

- `sudo apt install docker.io`
- `docker run hello-world`

(Note*: After installing docker, run "docker run hello-world" to test if docker runs without root/sudo permissions, if it requires root permissions then you will get error regarding docker.sock in /var/run, you need to change this file permission buy running the following :)

- `cd /var/run`
- `sudo su`
- `chmod 666 docker.sock`
- `exit`

Inside the working directory (chip_design here) clone the following and hence install Openlane

- `git clone https://github.com/The-OpenROAD-Project/OpenLane`
- `cd OpenLane`
- `make`
- `make test`

STEP 2: INSTALLING ICARUS, GTK, KLAYOUT

In order to run different digital designs Icarus Verilog serves as a means. Further in order to cross verify the output gtkwave needs to be installed so as to verify the simulation results.

1) Installation of iverilog

   Iverilog can be installed using the following command:

   - `sudo apt install iverilog`

2) Installation of gtkwave

To install gtkwave first of all, update the system using:

- `sudo apt-get update`

Then in the same terminal, run :

- `sudo apt-get -y install gtkwave`

We can see the gtkwave window by calling it on the respective terminal

3) Installation of klayout

Open a terminal in the main folder and install :

- `sudo apt-get install klayout`

Now, klayout can be opened by calling "klayout".

Sometimes, wayland errors may occur as shown here. At such instances, run the command : XDG_SESSION_TYPE=x11 klayout

Now, try running klayout and see the window appears.

ALTERNATIVE:

If not installed properly using the above try the alternative way as:

Download the latest Klayout from: Klayout

KLayout system requirements are:

e) **C++**: clang (tested on 3.8.0 and many more) or gcc (tested on 4.6.3 and many more).
f) **Qt**: Qt4 >= 4.8.5 (Qt 5 preferred, Qt 6 supported).
g) **Ruby**: >= 1.9.x (Ruby 2 or 3 preferred).
h) **Python**: >= 3.x.

Install using:

First install pre-requisites

(# Kindly note: Depending upon system the missing packages may appear, so installing them could help in installing klayout)

- `sudo apt install gcc ruby-full`

```
sudo apt-get install libqt5opengl5 libhttp-parser2.9 libmbedcrypto7 libmbedtls14 libmbedx509-1 libssh2-1 libgit2-1.1
sudo dpkg -i klayout_0.29.5-1_amd64.deb
```

#(If it still indicates that a package is missing, install the required packages.)

#(If it pops message as “apt --fix-broken install”)

Give command :

```
sudo apt --fix-broken install
```

and rerun

```
sudo dpkg -i klayout_0.29.5-1_amd64.deb
```

Check Klayout using command

```
klayout
```