**RISC-V REFERENCE SOC TAPEOUT PROGRAM VSD**

TOOLS INSTALLATION

System Requirements

* 6 GB RAM
* 50 GB HDD
* Ubuntu 20.04 or higher
* 4 vCPU

Installing Ubuntu and preparing for Development in VirtualBox

When you install Ubuntu (especially in a virtual machine like VirtualBox), you’re setting up a Linux environment. This is often used for software development, chip design flows, penetration testing, or learning Linux systems.

After installation, there are some basic steps needed to prepare your environment. That’s what your commands are doing. Let’s walk through them one by one.

STEPS:

$ sudo apt update

$ sudo apt install build-essential dkms linux-headers-$(uname -r)

$ cd /media/spatha/VBox\_GAs\_7.1.8/

$ ./autorun.sh

**YOSYS**

* It reads Verilog RTL (Register Transfer Level) source code.
* Performs Synthesis, transforming behavioral/RTL code into gate-level netlists.
* Supports technology mapping for different standing cell libraries (e.g., sky130, ASAP7,etc.).
* Used in OpenLane, OpenROAD,and open-source tapeouts (e.g., through SkyWater PDK).

$ sudo apt-get update

$ git clone https://github.com/YosysHQ/yosys.git

$ cd yosys

$ sudo apt install make (If make is not installed please install it)

$ sudo apt-get install build-essential clang bison flex \ libreadline-dev gawk tcl-dev libffi-dev git \ graphviz xdot pkg-config python3 libboost-system-dev \ libboost-python-dev libboost-filesystem-dev zlib1g-dev

$ make config-gcc

$ make

$ sudo make install

**Iverilog**

Icarus Verilog (Iverilog) is a Verilog simulator and compiler used to test, simulate, and verify digital logic written in Verilog HDL (Hardware Description Language).

It’s often used in combination with:

GTKWave-for waveform viewing

Yosys-for synthesis

OpenLane/ASIC flow tools-for chip design

$ sudo apt-get update

$ sudo apt-get install iverilog

**gtkwave**

You're installing **GTKWave**, which is a **waveform viewer** used in digital design and simulation workflows, especially after running Verilog simulations with tools like **Icarus Verilog (iverilog)**.

**GTKWave** is a GUI tool that allows engineers to **view simulation waveforms** from digital designs.

**It is used to:**

* **Visualize signal changes over time** (like a digital oscilloscope).
* Analyze **VCD (Value Change Dump)** files generated by simulators (e.g., iverilog, ModelSim, Verilator).
* Debug logic bugs, verify timing, and check behavior of RTL designs.

$ sudo apt-get update(Refresh APT package index)

$ sudo apt install gtkwave(Install GTKWaveform viewer)

**ngspice**

powerful open-source **circuit simulator** based on SPICE (Simulation Program with Integrated Circuit Emphasis). It's used for analog, digital, and mixed-signal simulations.

**Ngspice** is an open-source SPICE simulator for electronic circuit simulation.  
It can simulate **analog**, **digital**, and **mixed-signal** systems using **netlists** or schematic inputs.

**Use Cases:**

* Simulating **RC, RLC, transistor, op-amp circuits**
* Analog IP (e.g., ADC, DAC, PLL)
* **Custom circuits** before hardware implementation
* Used in tools like **KiCad**, **LTspice-compatible flows**, etc.

$ tar -zxvf ngspice-37.tar.gz

$ cd ngspice-37

$ mkdir release

$ cd release

$ ../configure --with-x --with-readline=yes --disable-debug

$ make

$ sudo make install

**magic**

**Magic** is an open-source **VLSI layout editor and tool** used for:

* Drawing physical layouts of integrated circuits (ICs)
* Running DRC (Design Rule Check)
* Running LVS (Layout vs Schematic)
* Exporting GDSII files for tapeout
* Used in modern open-source ASIC flows like OpenLane

$ sudo apt-get install m4

$ sudo apt-get install tcsh

$ sudo apt-get install csh

$ sudo apt-get install libx11-dev

$ sudo apt-get install tcl-dev tk-dev

$ sudo apt-get install libcairo2-dev

$ sudo apt-get install mesa-common-dev libglu1-mesa-dev

$ sudo apt-get install libncurses-dev

**OpenLANE**

**OpenLane** is an **open-source digital ASIC design flow** that automates the process of turning Verilog RTL code into a **GDSII layout** — the format used for chip fabrication.

It is part of the **OpenROAD project** and widely used in projects like:

* **Google’s Skywater MPW tapeouts**
* **TinyTapeout**
* **Efabless shuttle program**

OpenLane automates the **RTL-to-GDSII** flow by integrating multiple open-source tools into one unified environment.

INSTALLATION:  
You can install OpenLane in **two main ways**:

1. **Using Docker (Recommended & Portable)**
2. **Native installation (Advanced users)**

1.clone OpenLane

git clone https://github.com/The-OpenROAD-Project/OpenLane.git

cd OpenLane

2.Install Required system packages

sudo apt update

sudo apt install -y git make python3 python3-pip build-essential \

libx11-dev x11-utils tcl-dev tk-dev libcairo2-dev \

libglu1-mesa-dev libncurses-dev wget

3.Install Docker(Optional but recommended)

sudo apt install docker.io

sudo systemctl enable docker

sudo systemctl start docker

sudo usermod -aG docker $USER

4.Install PDK(e.g., Sky130)

make pdk

5.Build OpenLane Tools(Optional if not using Docker)

make openlane

6.Run OpenLane in Docker

make mount

This command:

* Starts the Docker container
* Mounts your local directory inside Docker
* Launches a bash shell in the preconfigured environment