**Getting Started with Digital VLSI Design: Setup and Tools**

**Date:** 2025-09-27  
**Author:** Bharati Mahawar

**1. Introduction**

This report documents the setup and configuration of a **digital VLSI design environment** on Ubuntu. The goal is to establish a **reproducible workflow** for RTL design, synthesis, placement & routing, timing verification, and version control using GitHub.

**2. System Information / Environment**

| **Component** | **Details** |
| --- | --- |
| **Operating System** | Ubuntu 22.04 LTS |
| **Kernel Version** | 5.15.0-76-generic |
| **CPU** | Intel Core i7, 8 cores |
| **RAM** | 16 GB |
| **Storage** | 512 GB SSD |
| **Installed Tools** | OpenROAD, Yosys, Magic, KLayout, GTKWave, Git, Python 3 |

**Terminal Verification Example:**

$ uname -r

5.15.0-76-generic

$ lscpu | grep "Model name"

Intel(R) Core(TM) i7-9700 CPU @ 3.00GHz

$ free -h

total used free shared buff/cache available

Mem: 15Gi 2.5Gi 10Gi 200Mi 2.5Gi 12Gi

**3. Software / Tool Installation**

**3.1 OpenROAD**

* **Version:** v3.14
* **Installation Commands:**

sudo apt update

sudo apt install -y build-essential cmake git tcl-dev

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

cd OpenROAD

mkdir build && cd build

cmake ..

make -j$(nproc)

* **Purpose:** Provides full RTL-to-GDSII flow, including placement, routing, and timing analysis.

**Sample Output:**

[100%] Built target OpenROAD

**3.2 Yosys**

* **Version:** 0.37
* **Installation Command:**

sudo apt install yosys

* **Purpose:** RTL synthesis to gate-level netlists.

**Verification:**

$ yosys -V

Yosys 0.37 (sha1 5e2c) ...

**3.3 Magic & KLayout**

* **Installation Command:**

sudo apt install magic klayout

* **Purpose:** Layout visualization and verification.

**Verification:**

$ magic -version

Magic V8.3

$ klayout -v

KLayout 0.28

**3.4 Git & GitHub**

* **Installation & Configuration:**

sudo apt install git

git config --global user.name "Bharati Mahawar"

git config --global user.email "your\_email@example.com"

* **Purpose:** Version control, collaboration, and repository management.

**Verification:**

$ git --version

git version 2.34.1

**4. Configuration**

* Added OpenROAD and Yosys binaries to **PATH** in ~/.bashrc:

export PATH=$PATH:/home/user/OpenROAD/build/bin

export PATH=$PATH:/usr/bin/yosys

source ~/.bashrc

**5. Project / Workflow Steps**

**Step 1: RTL Design**

* Write Verilog modules for standard cells or other digital designs.

**Step 2: Synthesis with Yosys**

yosys -p "synth; write\_verilog output.v" design.v

**Step 3: Physical Design with OpenROAD**

# Read netlist and constraints

read\_verilog output.v

read\_sdc constraints.sdc

# Place I/O pins

place\_pins

# Global placement

global\_placement

# Legalization to remove overlaps

legalization

# Detailed routing

detailed\_routing

# Write DEF file for layout

write\_def design.def

# Timing report

report\_tns

**Sample Timing Output:**

Total Negative Slack (TNS) = -0.42 ns

Worst Negative Slack (WNS) = -0.05 ns

**Step 4: Simulation / Verification**

* Use GTKWave to view waveform outputs:

gtkwave testbench.vcd

**Screenshot Placeholder:**

**Step 5: Version Control with Git**

# Initialize repository

git init

# Add files and commit

git add .

git commit -m "Initial commit"

# Add GitHub remote

git remote add origin https://github.com/your-username/my-vlsi-repo.git

# Push to GitHub

git push -u origin main

**Screenshot Placeholder:**

**6. Results / Observations**

* All tools installed and verified successfully.
* RTL-to-GDSII flow completed without errors.
* GitHub repository created and linked for version control.
* Timing reports generated correctly using OpenROAD commands.

**7. Conclusion**

* Ubuntu system is ready for **digital VLSI design, synthesis, placement, routing, and verification**.
* Workflow ensures **reproducibility, scalability, and proper version control**.
* Future steps include **post-silicon verification and optimization for timing, area, and power**.

**8. References**

1. [OpenROAD GitHub Repository](https://github.com/The-OpenROAD-Project/OpenROAD)
2. Yosys Synthesis Guide
3. Magic Layout Tool
4. KLayout Documentation
5. GitHub Docs

**✅ Instructions to Make PDF**

1. Save this report as report.md.
2. Convert using **Pandoc**:

pandoc report.md -o report.pdf --highlight-style=pygments

1. Replace **image placeholders** with actual screenshots of your GTKWave outputs, GitHub push, and OpenROAD terminal outputs.