**INDIA RISC – V CHIP TAKEOUT PROGRAM (WITH IIT GANDHINAGAR) WEEK 0 :**

Task 1: Chip Design Flow – Summary

This week explained the chip design flow and verification process:

* Chip Modelling (O0 & O1):  
  Model chip functionality in C (C model) and verify with a C testbench compiled using GCC.
* RTL Design (O2):  
  Create hardware description using Verilog RTL, simulate with the same testbench, and get a functional processor + peripherals model.
* Synthesis & Netlist (O3):  
  Convert RTL to a gate-level netlist, integrate processor, peripherals, and IPs, then perform floor planning, placement, CTS, routing, and DRC/LVS checks.
* Final Chip (O4):  
  Test final chip at target frequency (100–130 MHz). Verify that O1 = O2 = O3 = O4.  
  Chips can then be used in devices like smartwatches, Arduino boards, TVs, and AC controllers.

# Task 2: Tool Installation (In Progress)

The installation of required tools is planned but not yet completed. All steps and commands are ready, and screenshots will be uploaded once the setup in Ubuntu is finished.

Tools to be installed:

• Yosys

• Icarus Verilog

• GTKWave

• Ngspice

• Magic

• OpenLane

At the moment, the download of the Ubuntu ISO is still in progress due to slow network speed. Once the operating system installation is complete and all the required tools are installed, the codes will be run properly.

Prepared installation commands (to be run after Ubuntu setup):

The complete set of installation commands for Yosys, Icarus Verilog, GTKWave, Ngspice, Magic, and OpenLane has already been documented. These commands will be executed in sequence once the Ubuntu environment is successfully configured.

* **Yosys :**

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

cd yosys

sudo apt install -y clang bison flex libreadline-dev gawk tcl-dev libffi-dev \

graphviz xdot pkg-config python3 libboost-system-dev libboost-python-dev \

libboost-filesystem-dev zlib1g-dev

make config-gcc

make -j$(nproc)

sudo make install

cd ..

* **Icarus Verilog :**

sudo apt-get install -y iverilog

* **GTKWave :**

sudo apt-get install -y gtkwave

* **Ngspice :**

wget https://sourceforge.net/projects/ngspice/files/ng-spice-rework/37/ngspice-37.tar.gz

tar -zxvf ngspice-37.tar.gz

cd ngspice-37

mkdir release && cd release

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

make -j$(nproc)

sudo make install

cd ../..

* **Magic :**

sudo apt-get install -y m4 tcsh csh libx11-dev tcl-dev tk-dev libcairo2-dev \

mesa-common-dev libglu1-mesa-dev libncurses-dev

git clone https://github.com/RTimothyEdwards/magic

cd magic

./configure

make -j$(nproc)

sudo make install

cd ..

* **OpenLane :**

# Install Docker (first time only)

curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo gpg --dearmor -o \

/usr/share/keyrings/docker-archive-keyring.gpg

echo "deb [arch=amd64 signed-by=/usr/share/keyrings/docker-archive-keyring.gpg] \

https://download.docker.com/linux/ubuntu $(lsb\_release -cs) stable" | \

sudo tee /etc/apt/sources.list.d/docker.list > /dev/null

sudo apt-get update

sudo apt install -y docker-ce docker-ce-cli containerd.io

sudo docker run hello-world

sudo groupadd docker

sudo usermod -aG docker $USER

After reboot

docker run hello-world

Clone and build OpenLane

cd $HOME

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

cd OpenLane

make

make test