Week 0 of the INDIA RISC-V Chip Takeout Program (with IIT Gandhinagar) focused on learning the chip design flow, creating a GitHub repo, and setting up tools for future labs.

This week I learned how a chip is designed step by step:

C Model (O0 & O1): First, we write the chip's functions in C and test it with a C testbench. GCC is used to run and check the outputs.

RTL Design (O2): The hardware is described in Verilog (RTL). Same testbench is used to check if RTL behaves like the C model.

Synthesis (O3): RTL is converted into a gate-level netlist. Then SoC is built with processor, peripherals, macros, and IPs. Steps like floorplanning, placement, CTS, routing, and checks (DRC/LVS) are done.

Final Chip (O4): Chip runs at target speed (100–130 MHz). The design is verified again so every stage (C → RTL → Netlist → Final Chip) gives the same result.

Such chips are used in smartwatches, Arduino, TVs, ACs, etc.

Key Idea: Each stage must match the original design to avoid errors before fabrication.

Tasks

Task 1 – GitHub Setup

Created a repository.

Added this summary.

Will store logs, screenshots, and updates there.

Task 2 – Tool Installation (In Progress)

Need to install these on Ubuntu 20.04 (VirtualBox):

- **Yosys**
- **Icarus Verilog (iverilog)**
- **GTKWave**
- **ngspice**

- **Magic**
- **OpenLANE**

The commands below are prepared but not executed yet. Actual installation logs and screenshots will be uploaded once Ubuntu setup is complete.

Installation Commands (Ready to Run after it ready)

- **Yosys**

```
sudo apt-get update

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

cd yosys

sudo apt install make 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
```

- **Icarus Verilog**

```
sudo apt-get update

sudo apt-get install iverilog
```

- **GTKWave**

```
sudo apt-get update

sudo apt install gtkwave
```

- **Ngspice**

```
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**

```
sudo apt-get install 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
make install
```

- **OpenLANE**

```
sudo apt-get update && sudo apt-get upgrade
sudo apt install -y build-essential python3 python3-venv python3-pip make git
sudo apt install apt-transport-https ca-certificates curl software-properties-common
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 update
sudo apt install docker-ce docker-ce-cli containerd.io
sudo docker run hello-world
sudo groupadd docker
sudo usermod -aG docker $USER
sudo reboot
```