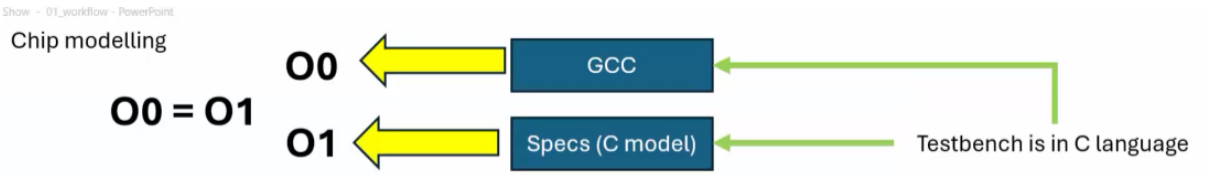
WEEK-0

- "Silicon proven" refers to a design or technology that has been physically tested and verified on silicon hardware

All instructions are for processors →

Step#1

- first we will not directly jump in designing the RTL code for the chip. we will write the specifications for the chip in C kind off environment.
- then we will check the specifications. model the specs in C.
- test it using the architecture based Gcc like arm gcc or risc-v gcc. purpose is o0=o1.

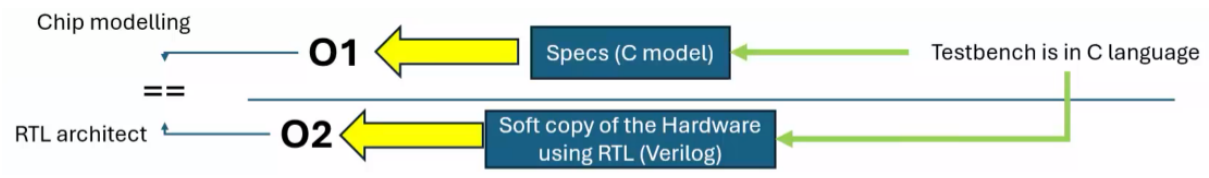


Step#2

- soft copy of the hardware using abtract level languages that lies somewhere between the c and verilog. these languages are **chisel and bluespec**

Step#3

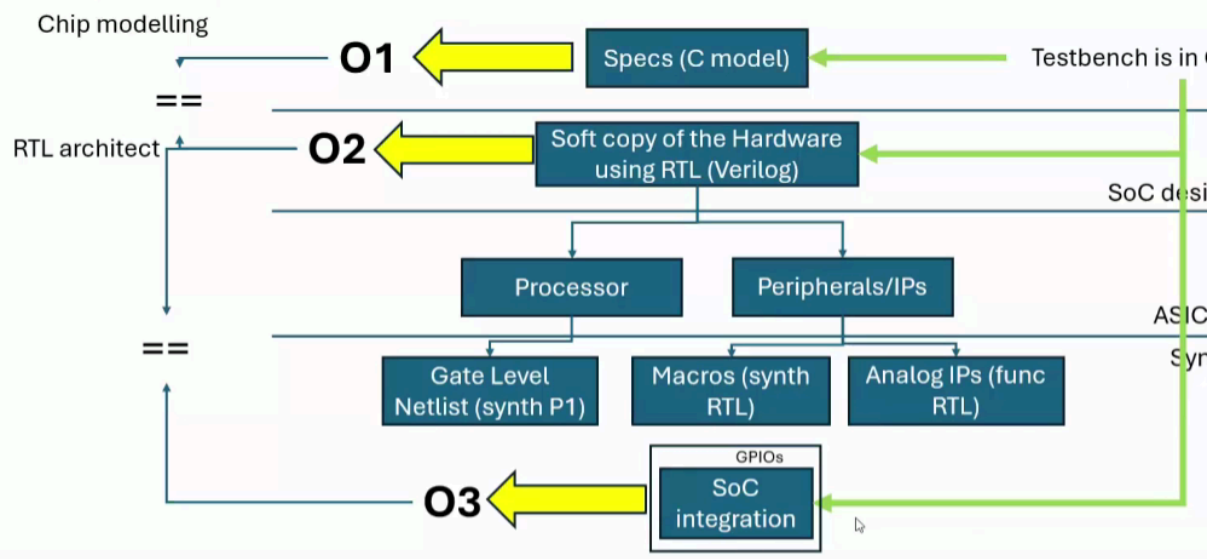
- Application(step#1 specifications) runs on the hardware and measure the output o2.
- Make sure that o2==o1



Step#3 { SOC Design }

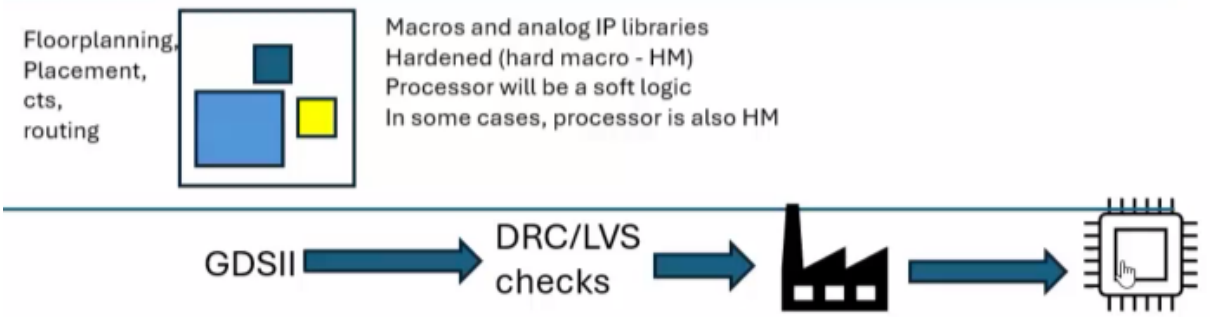
- divide verilog into processor and IP's and the processor verilog code should be synthesizable (make sure that).

- Macros - modules which can be instantiate multiple times should be synthesizable.
- Analog IPs - PLL, ADC, DAC or clock multiplier (functional no need for synthesize).
- then we will integrate all these blcoks using GPIOs the output = o3
- Purpose will be o1=o2=o3.

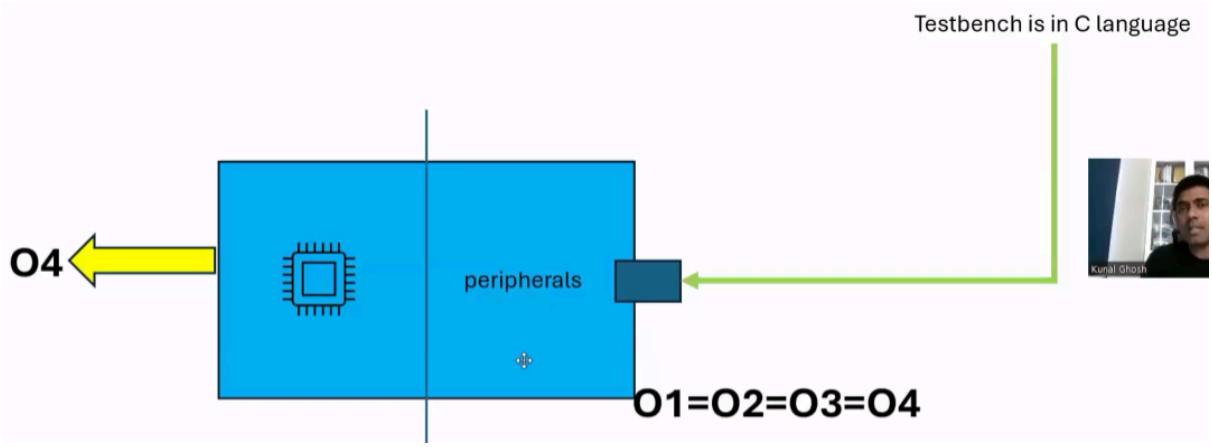


Step#4 { RTL TO GDSII (graphical data stream information interchange) }

- work is focussed on the blue block only.



- sending the chip = tapeout, receiveing = tapein
- run the same application written in thh C language in Step#1, run it on the block get back the output = o4.
- Purpose will be o1=o2=o3=o4 { chip design is done perfectly }



TOOLS SANPSHOTS