***Design of a Processor***

**step1:** We write an application using C and then compile using GCC compiler(linux) and measure an output O0

**step2:** Every Processor has got an architecture. We are going to model that architecture(specifications) in C and we are going to run the application on that model and measure an output O1. Architecture defines different instruction format. we need to check O0 == O1

**step3:** we make a soft copy of the hardware using High Level Language and use a converter to convert the high level language to verilog code. Verilog code of an architecture is called Micro Architecture.From step3 measures an output and check O2==O1.

We divide the top verilog code into Processors and Peripherals/IPs. Processor will not run automatically ,Processor needs peripherals to run. We make sure Processor is a sythesizable Processor. Synthesizable means each and every line of the Processor should be able to convert to gates. Peripherals are again divided into Macros(synthesizable RTL) and Analog IPs are built using circuits(NMOS,PMOS)(functional RTL) We do Synthesis, Placement, Routing and Power Planning in this stage and we generate a file called GDSII. After this it will go to some digital level checks(it basically checks for connectivity ) and this goes to foundary and once we get from factory we have to create a board and we pass the same test bench in C language and measures and Output O4 and check

