**Compiler Design Assignment**

**Design the custom instruction?**

**I am designing the volume of the sphere.**

Equation -> **4/3(pi\* r^3)**

**Steps: To Follow**

1. Clone the VexRiscv Repository:

git clone **https://github.com/SpinalHDL/VexRiscv**

2. Navigate to plugin directory:

**src/main/scala/vexriscv/plugin/**

3. Create your custom instruction Scala file:

Example: **SphereVolumePlugin.scala**

4. Folder Structure Example:

VexRiscv/

├── build.sbt

├── Makefile

├── README.md

├── src/

│ └── main/

│ └── scala/

│ └── vexriscv/

│ ├── VexRiscv.scala

│ ├── VexRiscvConfigs.scala

│ ├── plugin/

│ │ ├── Decoder.scala

│ │ ├── FetchCachePlugin.scala

│ │ ├── RegFilePlugin.scala

│ │ └── **SphereVolumePlugin.scala** (My plugin)

5. Register your plugin:

Go to: **src/main/scala/vexriscv/demo/GenCustomSimdAdd.scala**

Add your plugin like**: new SphereVolumePlugin(),**

6. Install SBT (Scala Build Tool):

- Download MSI from: **https://www.scala-sbt.org/download/**

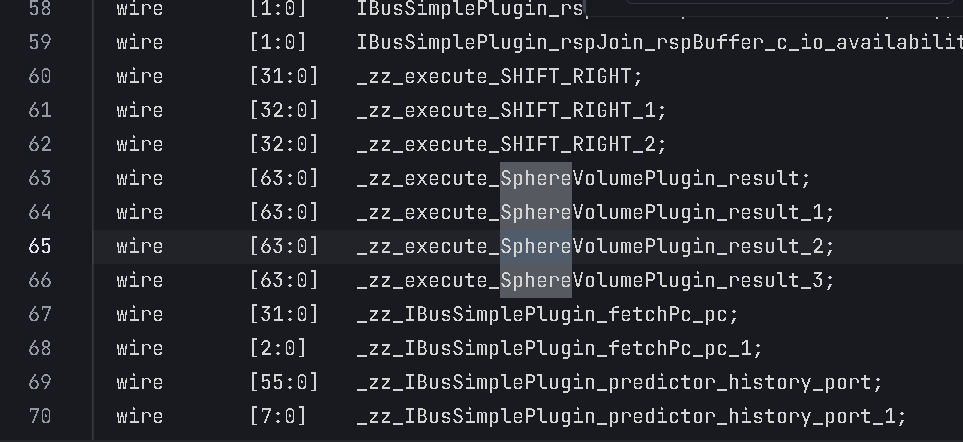
- Install and add SBT's **bin folder to system PATH**

7. Build the project:

sbt compile

=> It will generate **VexRiscv.v file**

It contains the instruction **SphereVolumePlugin\_results.**



8. Write your C++ testbench:

- **Include VVexRiscv.h and main logic to test your custom instruction**

9. Install MSYS2 and Verilator:

- Download **MSYS2 from: https://www.msys2.org**

- Open MSYS2 terminal and run:

**pacman -Syu**

**pacman -S base-devel mingw-w64-x86\_64-toolchain git**

10. Open the MSYS2 MinGW 64 terminal and install Verilator:

**pacman -S mingw-w64-x86\_64-verilator**

Compile the project using:

**verilator -Wall --cc VexRiscv.v --exe main.cpp**

11. Create object directory for the compiler code:

**make -C obj\_dir -f VVexRiscv.mk VVexRiscv -j**

12. Run Simulation:

Navigate to the **obj\_dir**

Run the code:

**./VVexRiscv.exe**

