TORTURE Initial Testing

***Note: This document is meant to be in continuation of the "chipyard setup.pdf" document. It is assumed that you have followed the steps or have at least gone through the mentioned document to know what is going on here.**

Torture is a random RISC-V instruction generator used for testing any core/SOC based on the RISC-V ISA. The generated instructions are fed to:

- Spike RISC-V simulator
- The core/device under test

After the instructions are run through both of the above, their internal registers are dumped to their respective memory from which we can extract and compare them to each other. The ones from the spike simulator are considered to be a reference for checking the device under test.

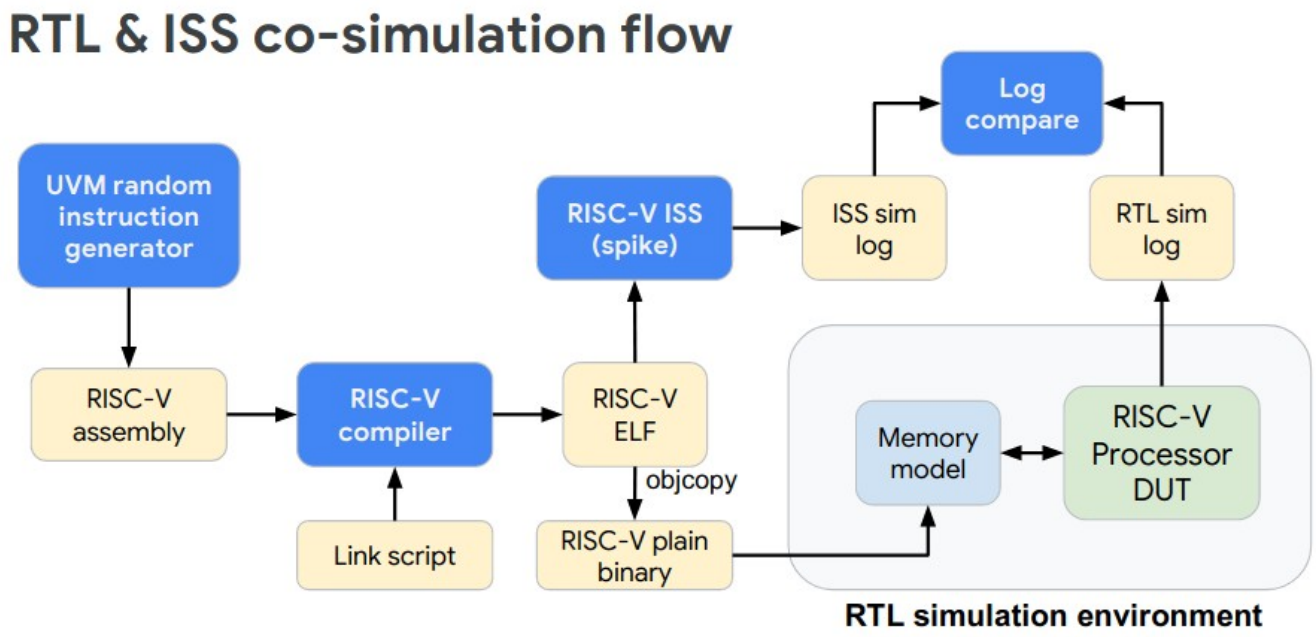

Illustration 1: Torture testing flow

At the end of the chipyard setup document, we did some testing on default configured rocket core and boom core. This testing resulted in the generation of two executable files in the path */home/your_username/chipyard/sims/verilator*:

- simulator-chipyard-RocketConfig
- simulator-chipyard-SmallBoomConfig

These executables are actually based off on the generated verilog code of the respective cores. These act as simulators that exhibit the same behavior as the core would do on an FPGA. We will use these in running torture along with the spike simulator.

First, go to the directory */home/your_username/chipyard/tools/torture* and open terminal. Type:

```
make igentest
```

You should get a success message. After that, type:

```
cd output/

spike +signature=spike_signature.txt test

./../../../sims/verilator/simulator-chipyard-RocketConfig
+signature=rocket_signature.txt test

./../../../sims/verilator/simulator-chipyard-SmallBoomConfig
+signature=boom_signature.txt test

diff spike_signature.txt rocket_signature.txt

diff spike_signature.txt boom_signature.txt
```

If there is no difference in the signature files (indicating no error), then the “diff” command will not show any output.

References:

1. Torture README file.

2. Chipyard Documentation - 8.3.4. Torture tests: https://chipyard.readthedocs.io/en/latest/Advanced-Concepts/Debugging-RTL.html

3. Illustration 1 taken from: https://content.riscv.org/wp-content/uploads/2018/12/14.25-Tao-Liu-Richard-Ho-UVM-based-RISC-V-Processor-Verification-Platform.pdf