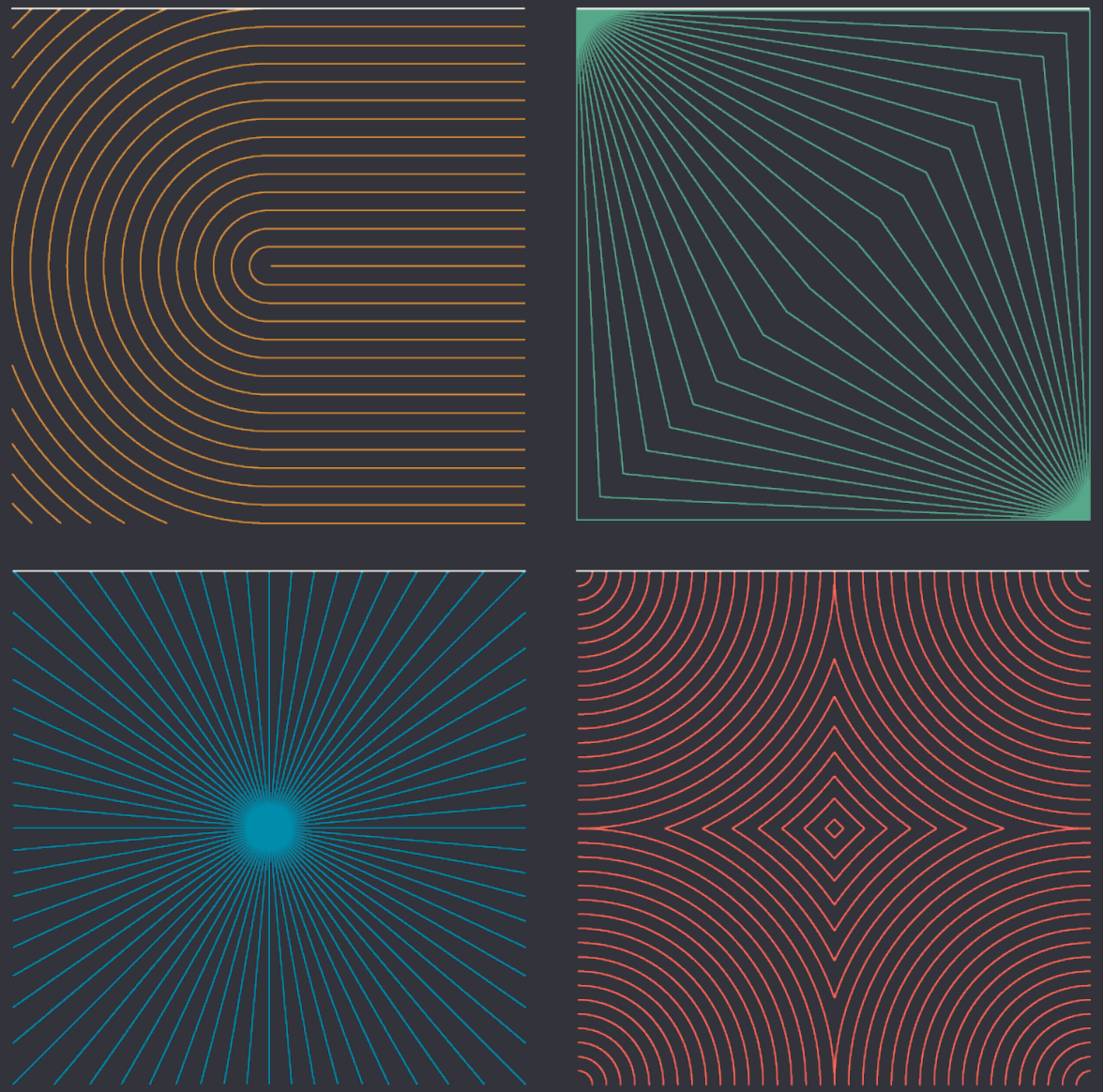
# Integrating a high performance ISS with FireSim to cosimulate OS boots

*Jiahan Zhang, Varun Koyyalagunta, Joe Rahmeh, Divyang Agrawal*



tenstorrent

# Agenda

- Goals
- Methodology
- Challenges
- Results
- Future Work

# Goals

- Reduce bringup/debug time of new RTL by using emulation (FPGA)
- Stages of bringup: unit tests, core bare metal tests, OS, apps
- OS and apps impractical on an RTL simulator: use emulation
- Reduce bringup/debug time by real-time checking of emulation results
  - Stop run on first mismatch
  - Generate snapshot of architectural state a number of instruction before mismatch
  - Use snapshot to reproduce bug on RTL simulator

# FireSim – Components Used

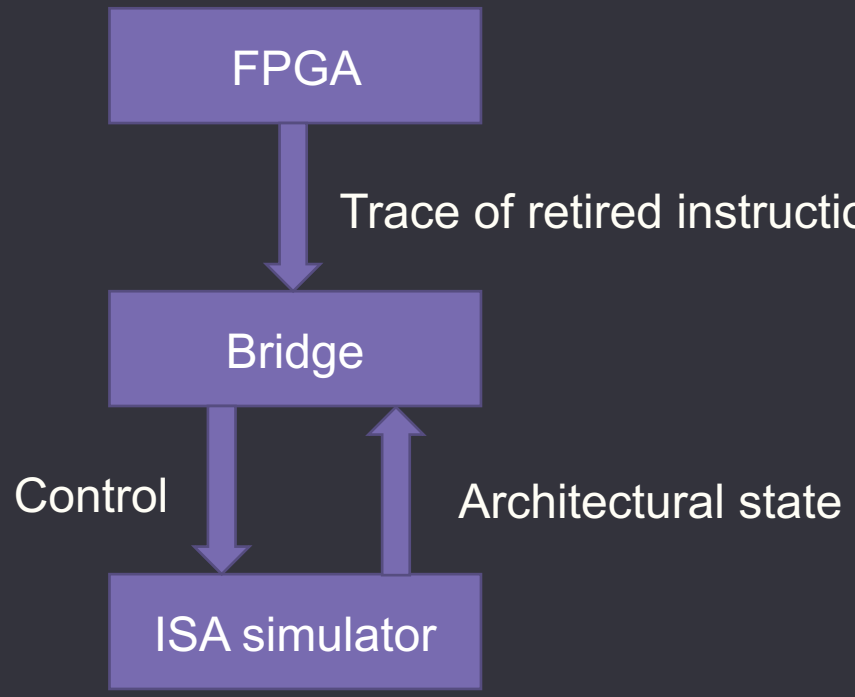
- Chipyard
  - SoC with devices and hw/sw bridges to boot linux
  - Easy to drop-in a custom core
- RISC-V core
  - SmallBoomConfig, LargeBoomConfig, etc
    - Rocket
- FireMarshal Linux image
- AWS FPGAs

# Methodology

- Want to bring up Whisper (ISS) co-simulation for architectural checks

- Client-Server model
  - BOOM already provides architectural state on instruction retires
    - trace packet sent to software driver through DMA (much like existing Dromajo cosim)
  - Bridge communicates to Whisper as a separate process
    - capable of many things
      - instruction stepping - match RTL
      - peek/poke architectural state - device memory regions, reservation sets, performance counters
      - VA translation
      - instruction decode

# System Diagram




- FPGA to bridge: DMA
- Bridge to ISA simulator: socket or shared memory
- FPGA sends 192 bits per retired instruction

# Methodology

1. Run app on ISS standalone
2. Run app on FPGA
3. In case of failure, run in co-simulation (locksetp) mode, stop on 1st divergence
4. Run app on ISS standalone, produce a snapshot of memory and architectural state before the failure point
5. Run app starting from snapshot on FPGA
6. If failure is reproduced, run app starting from snapshot on Verilog simulator

# Challenges

- Cosim should not keep up with FireSim
  - added shared memory IPC
    - latency sensitive
    - socket too slow
  - PC checking only
    - often sufficient, can fallback to full comparisons when mismatch is found

- Portability
  - Bridge driver implements E-Trace spec
    - conversion from BOOM trace packet
    - SYNC, TRAP, LOAD, ATOMIC, CSR

- System Verilog support: Conversion to FIRRTL

# Results

- Linux Boot (~1.3B instructions)
- SmallBoomConfig
- AWS EC2 f1.2xlarge
- Runtime comparison
  - whisper 70s
  - firesim (no cosim) 130s
  - firesim (w/ cosim) 1400s

# Future Work

- Multi-core
  - needs cycle-accurate retire from each core
- E-trace for single core?
- Improve simulation speed
  - still a lot to be gained

# References


[1] Whisper, ”chipsalliance/VeeR-ISS,” GitHub. [Online]. Avail- able: https://github.com/chipsalliance/VeeR-ISS. [Accessed: 20- Feb-2023].

[2] Whisper, ”tenstorrent/rv-core-dv-kit,” GitHub. [Online]. Avail- able: https://github.com/tenstorrent/rv-core-dv-kit. [Accessed: 20- Feb-2023].

[3] ”Dromajo co-simulation with boom designs,” Dromajo Co- simulation with BOOM designs - FireSim 1.15.1 documentation. [Online]. Available: https://docs.fires.im/en/stable/Advanced- Usage/Debugging-and-Profiling-on-FPGA/Dromajo.html. [Accessed: 20-Feb-2023].

[4] riscv-trace-spec, ”riscv-non-isa/riscv-trace-spec,” GitHub. [On- line]. Available: https://github.com/riscv-non-isa/riscv-trace-spec. [Accessed: 20-Feb-2023].