Esperanto's Verification Methodology for a RISC-V Machine Learning SoC

Raymond Tang & Shankar Jayaratnam

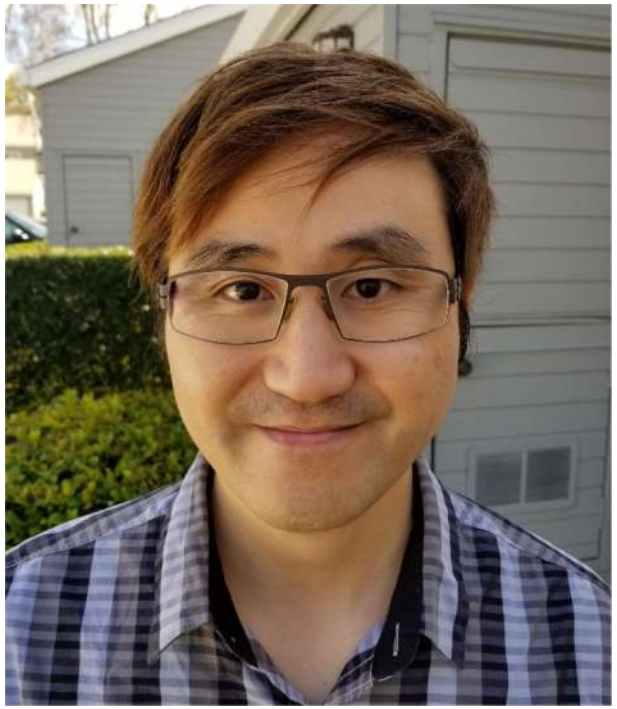
RISC-V Days Tokyo

November 6, 2020

Topics

- Our Speakers
- Esperanto's ET-SoC-1 Supercomputer on Chip
- Challenges to Verification
- Our Methodology
- Results
- Conclusions
- Q&A

Our Speakers



Raymond Tang
Director, Verification, Esperanto Technologies

- Formerly with Intel via Soft Machines
- SUN/Oracle
- Fujitsu, NEC

Shankar Jayaratnam
Senior Tech Lead, Esperanto Technologies

- Formerly with Intel
- Next-Gen big core pathfinding
- Many-core architecture (Knights family)

Esperanto's ET-SoC-1 Supercomputer on Chip

- Over 1,000 RISC-V cores, a mix of two types
  - ET-Maxion™ for Linux and application software
  - ET-Minion™ for machine-learning computation
- Cores and cache blocks are organized into a multi-level hierarchy using a mesh-style network on chip (NoC)
- Blocks connected to the NoC are called Shires
  - The ET-Minion Shire is our basic block of compute cores
  - Other Shires include Maxion cores, DRAM controllers, PCI Express controllers, a service processor, and so on
- The chip is built in TSMC's 7nm process

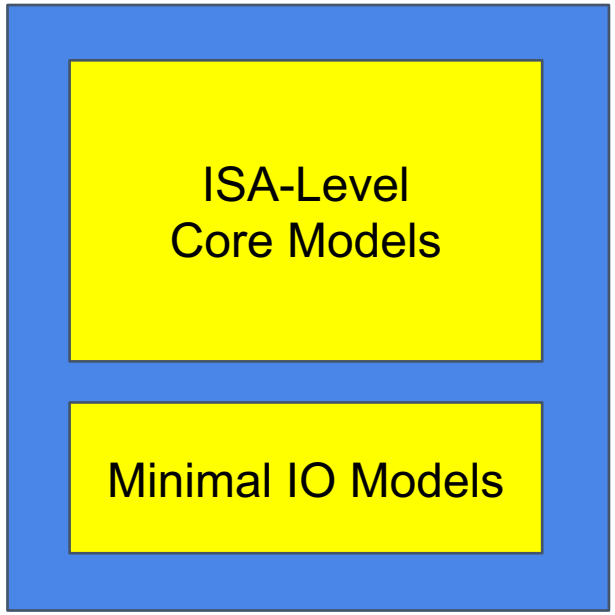
Challenges to ET-SoC-1 Verification

- As a new company, every aspect of the project was new to us
  - Cores, caches, interconnect, process, toolchain etc.
- The scale of the design was exceptionally large
- Architecture/hardware/software codesign saved time but made specifications a moving target

- We saw that verification was going to be a great challenge
- We invested heavily in our verification team and tools
  - With early adoption of Synopsys ZeBu® Emulation

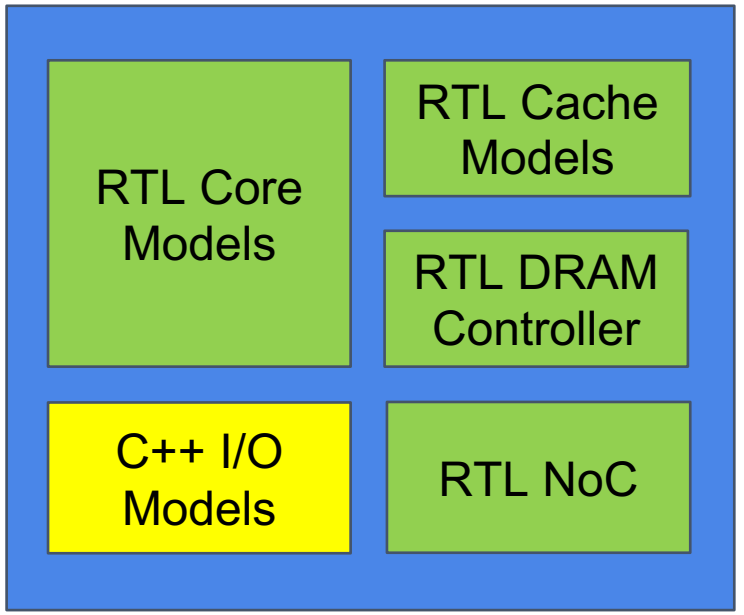
Design and Verification Platforms

C++ Virtual Platform



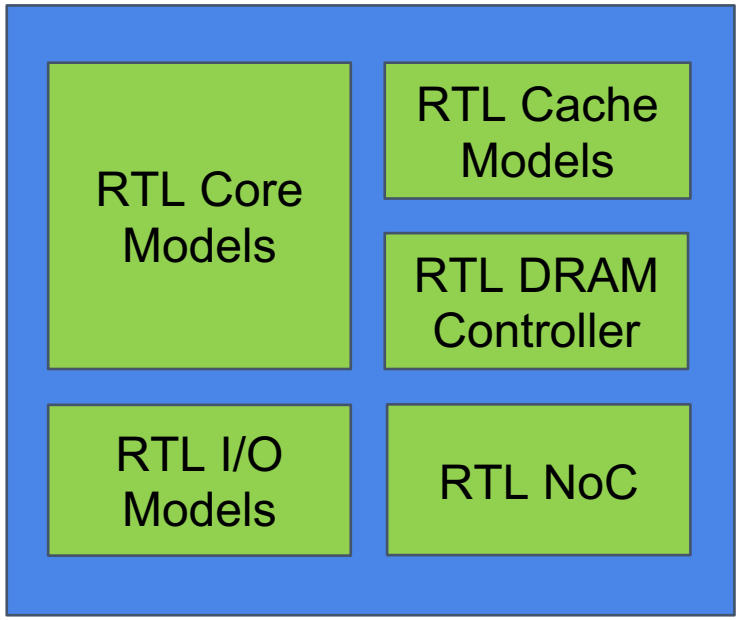
Architectural model for rapid exploration of design space and early software bring-up

Synopsys VCS® Simulation



Fully configurable to run fast simulation for unit and interface tests to find most RTL bugs

Synopsys ZeBu® Emulation

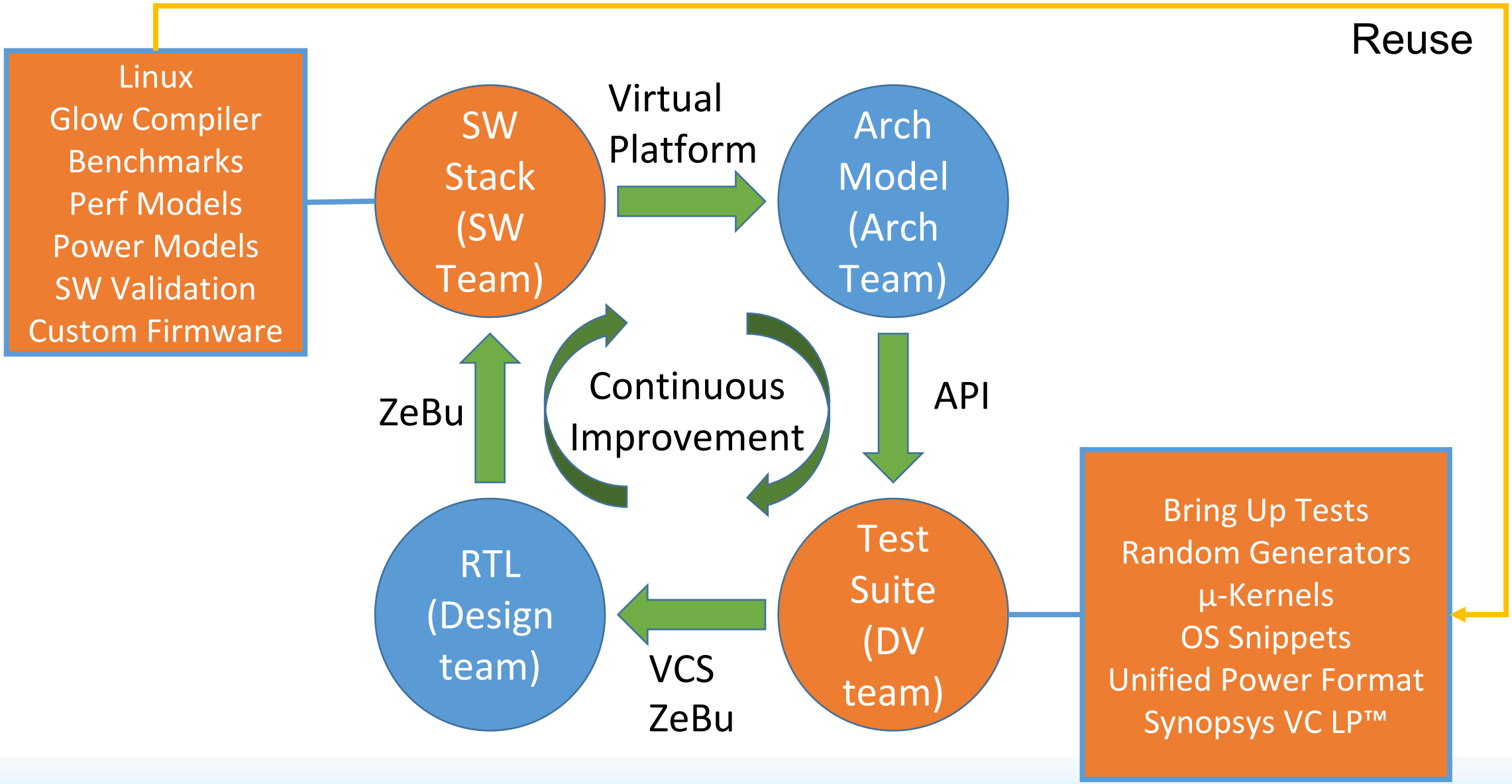


FPGA-based emulation for multi-core and whole-chip testing to find scale-out issues and tune SW perf

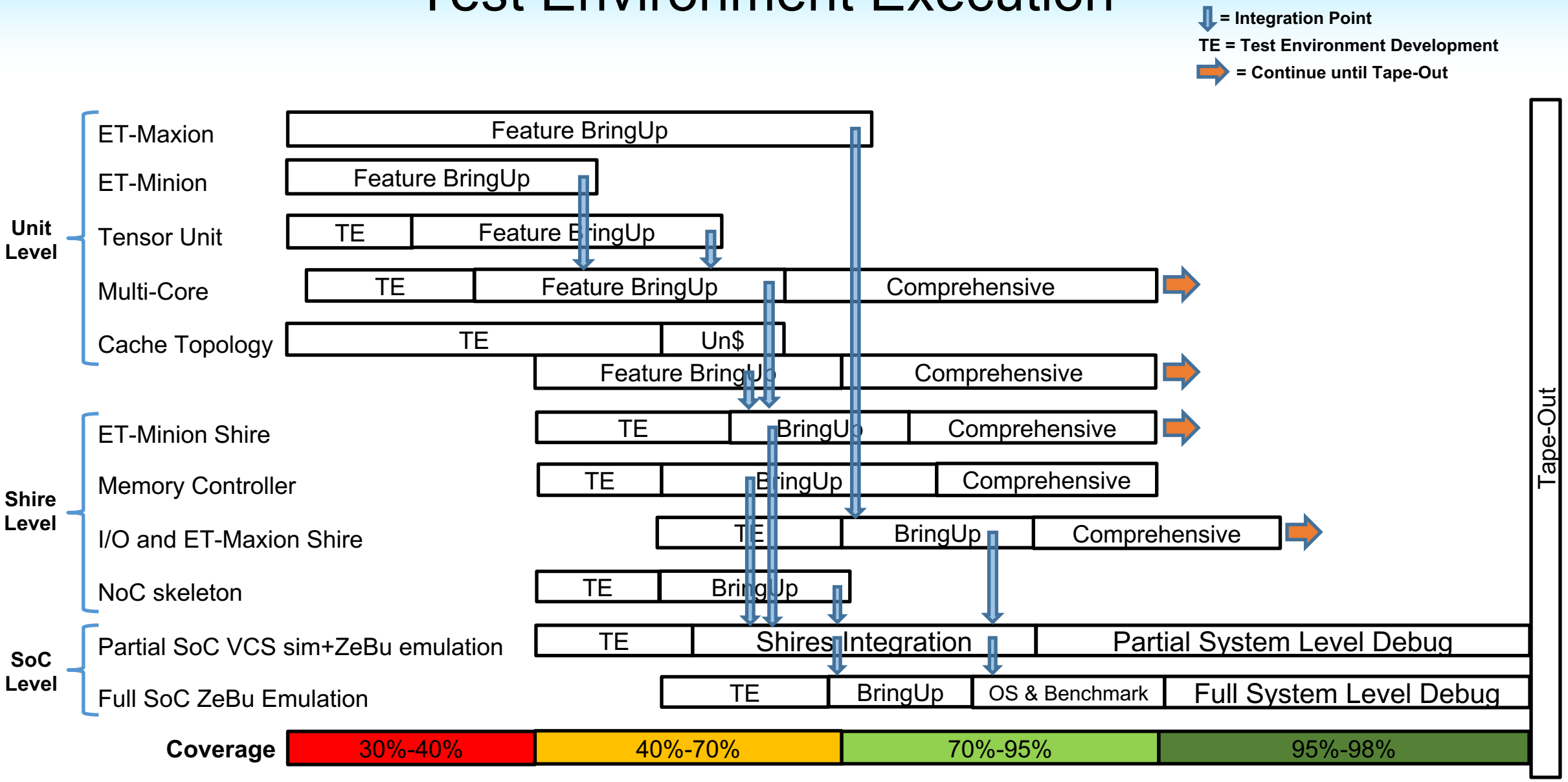
Our Experience With Three-Platform Verification

- Substantial verification IP reuse and minimal porting effort for checkers, assertions, coverage objects, stimulus and random generators
- Weekly VCS coverage-based profiling and 24/7 ZeBu operation drove rapid improvements in DV coverage
- Verification components and Bus functional models (BFM) helped maintain high throughput
- Architectural state checker worked with both VCS and ZeBu without compromising speed
- DV environments could run real software without modification
  - Ensured our SW and HW teams moved forward together

Co-Development Flow



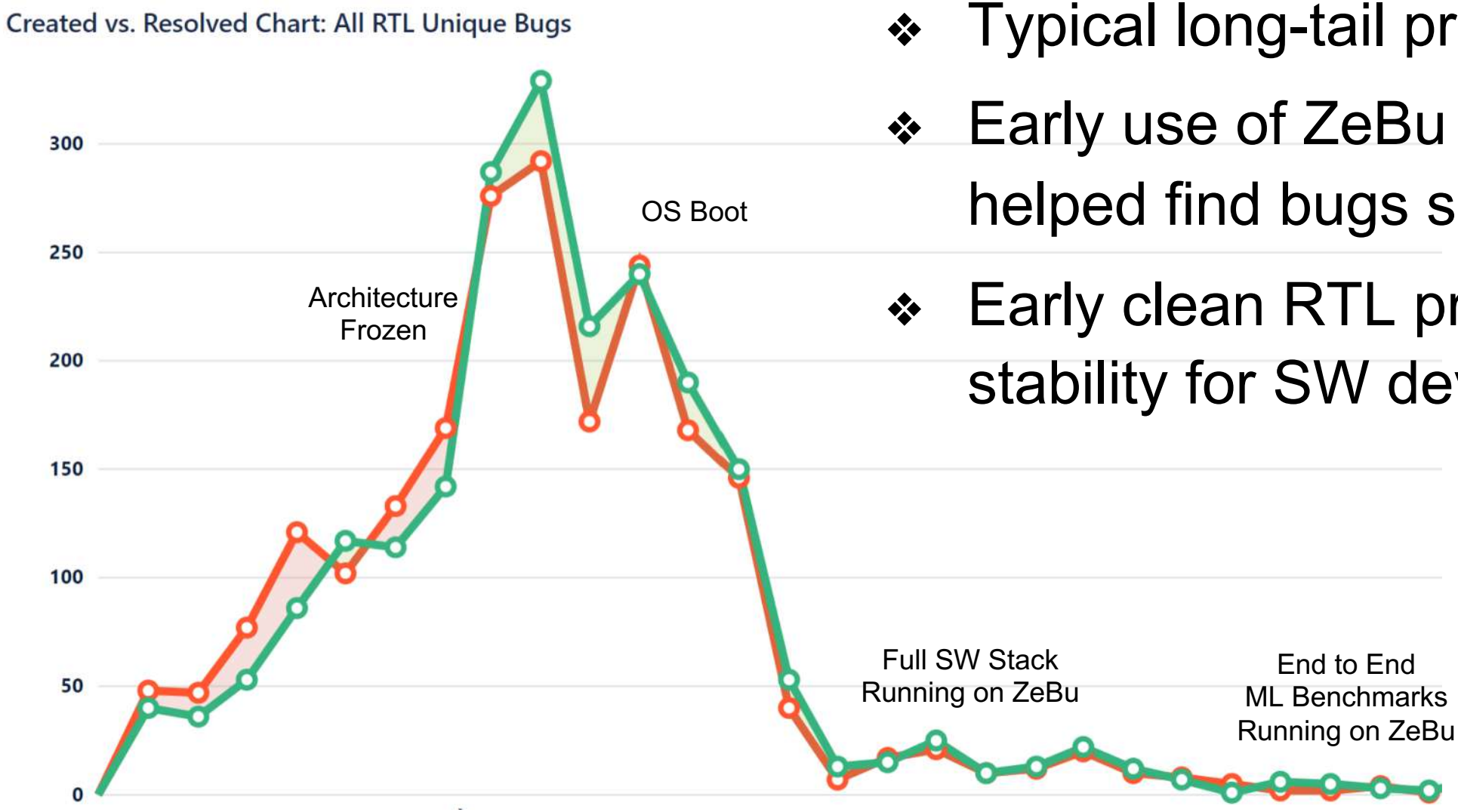
Test Environment Execution



ZeBu Scalability

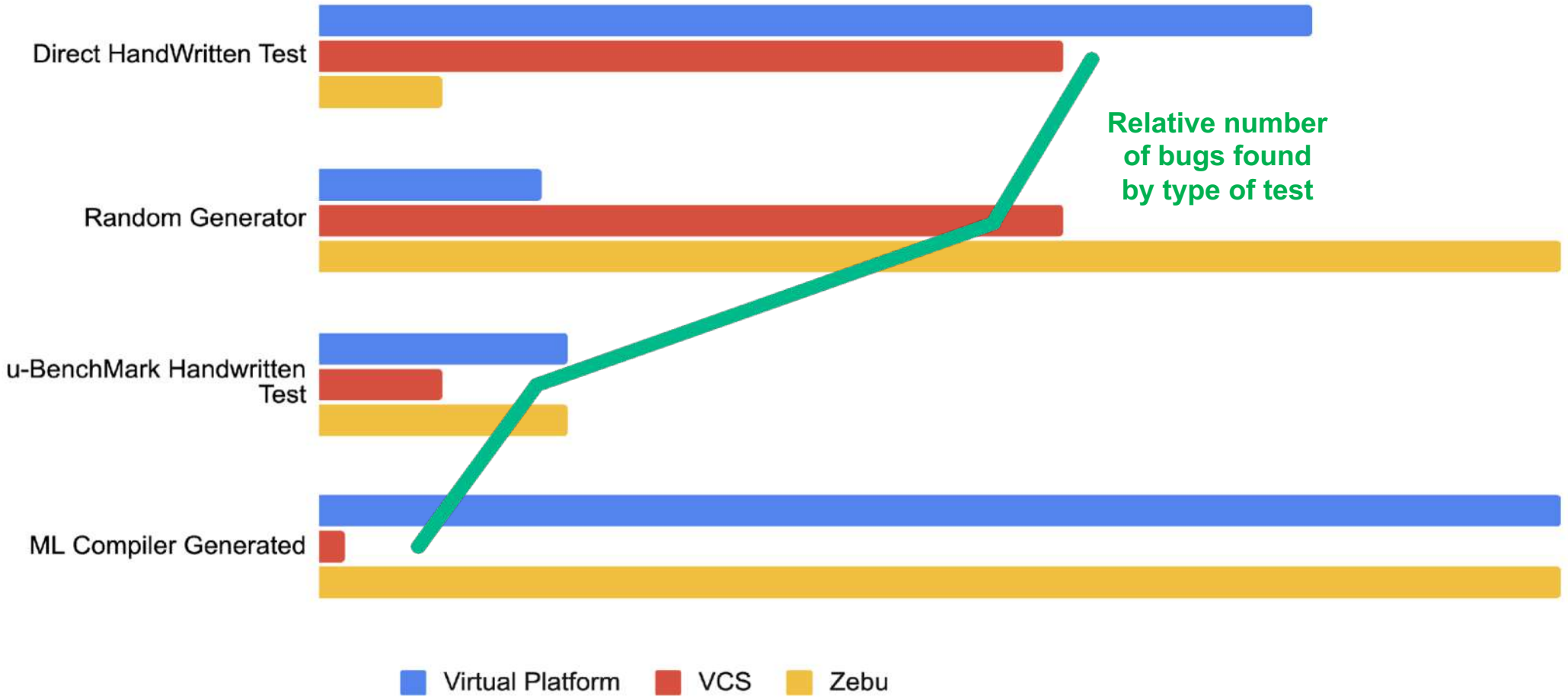
- Initially we used one ZeBu Server 4 (ZS4) to perform testing across pairs of Shires with stub models for memory and I/O
- Once we had confidence in Shire-to-Shire communication over the NoC, we began adding more Shires and RTL IP blocks
  - DRAM and PCIe controllers (both also from Synopsys)
  - On-die Service Processor (an independent ET-Minion core that manages boot and runtime service)
  - UltraSoC debug interfaces
- We scaled smoothly up to 8 ZS4s to emulate the full chip from RTL at MHz-level speed
- Cloud based executor to dispatch multiple jobs parallely

Finding and Fixing Bugs



- Typical long-tail process
- Early use of ZeBu platform helped find bugs sooner
- Early clean RTL provided stability for SW development

Test Cycle Count by Type and Platform



75% of bugs found at Shire level or below. 20% found with inter-Shire tests. 5% found at SoC level.

Benefits of Our Methodology

- Easy scaling from unit-level testing to full-chip testing
- Power analysis and performance validation guided improvements in both architecture and RTL
- ZeBu allowed our software team to achieve an amazing 31x speedup on benchmarks in one year—without silicon!
  - Starting with functional reference code and implementing a pre-planned sequence of optimizations that delivered the predicted results on a predictable schedule
- We released Dromajo, our RISC-V RV64GC emulator for RTL co-simulation, as an open-source project through Chips Alliance

Dromajo, a new RISC-V RV64GC Emulator for RTL co-simulation

- Esperanto Technologies emulator for co-simulation
  - Emulator based on F. Bellard's RISCVEMU, bug fixed and enhanced with ISA 2.3/priv 1.11
  - Single core co-simulation with support for exceptions and MMIO
  - Reasonably fast: ~17 MIPS on a 3GHz Intel Xeon Platinum 8124M
  - Apache license
  - https://github.com/chipsalliance/dromajo
- Capacity to create and resume checkpoints reusable across different cores
- Work-in-Progress
  - Integrate with external cores: BOOM, Ariane, black-parrot, …
  - Efficient SPEC2017 checkpoints

Conclusions

- Esperanto used parallel architectural modelling, simulation, and FPGA emulation to design and debug the ET-SoC-1
- This methodology helped find RTL bugs early in the design process, speeding bug fixes and software development
- Full-chip emulation enabled rapid and dramatic software performance gains before tape-out
- Well-integrated Synopsys IP, VCS simulation, and ZeBu emulation gives high confidence in production-ready first silicon

Q&A