Chipyard - Integrated Design, Simulation, and Implementation of Custom RISC-V SoCs

Alon Amid, Abraham Gonzalez, David Biancolin, Daniel Grubb, Sagar Karandikar, Harrison Liew, Albert Magyar, Howard Mao, Albert Ou, Nathan Pemberton, Colin Schmidt, John Wright, Paul Rigge, Jerry Zhao, Yakun Sophia Shao, Krste Asanovic, Borivoje Nikolic

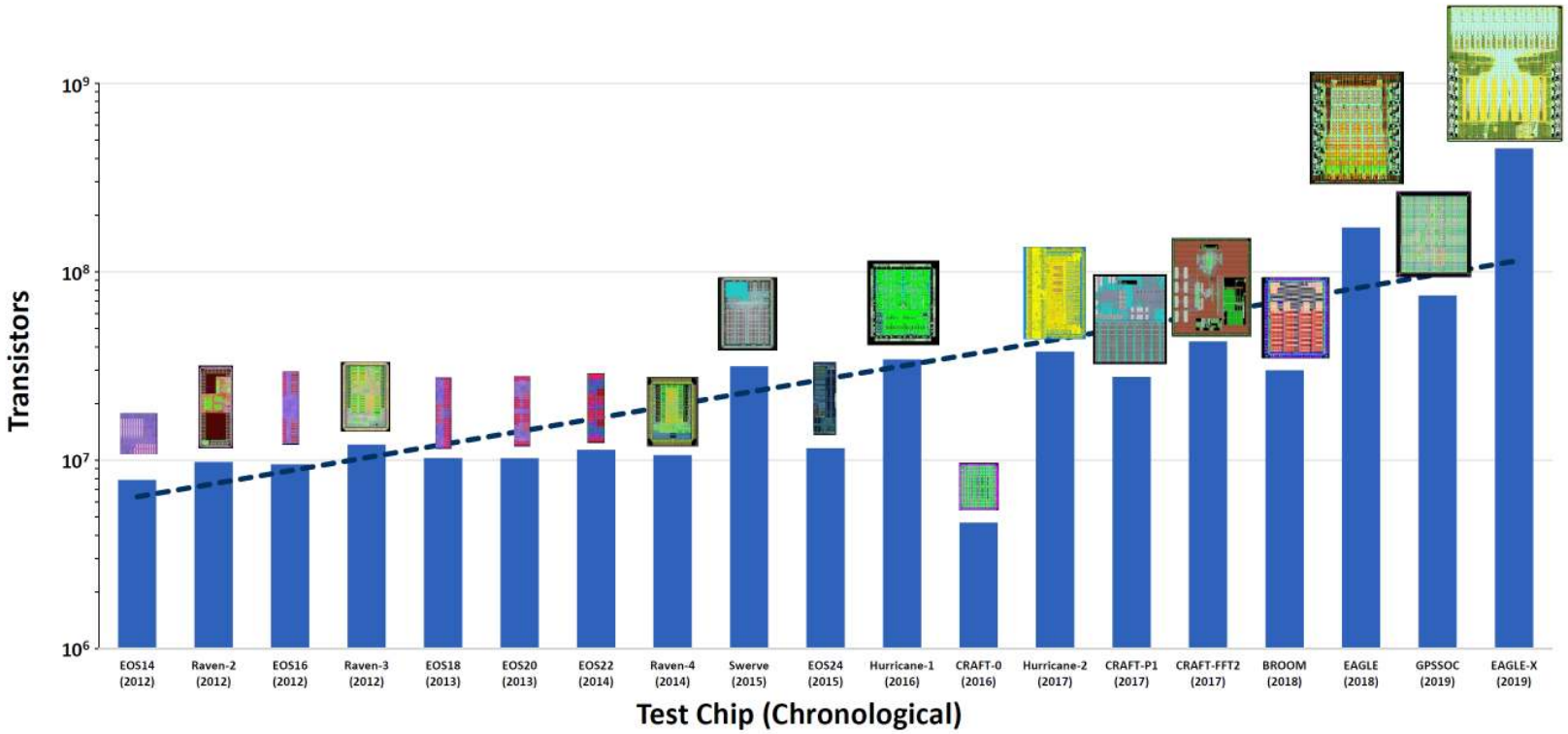
Berkeley Architecture Research

CHIPYARD

We present Chipyard - an open-source integrated SoC design, simulation, and implementation environment for specialized RISC-V compute systems. Continued improvement in computing efficiency requires functional specialization of hardware designs. Chipyard includes configurable, composable, open-source, generator-based IP blocks that can be used across multiple stages of the hardware development flow while maintaining design intent and integration consistency. Through cloud-hosted FPGA simulation and rapid ASIC implementation, this work allows for continuous validation of physically realizable customized systems. We demonstrate the capabilities of the Chipyard framework using the BEAGLE test-chip, a heterogenous SoC composed of an application class open-source Linux-capable BOOM out-of-order core connected to a vector accelerator and an in-order Rocket core connected to a systolic-array machine learning accelerator.

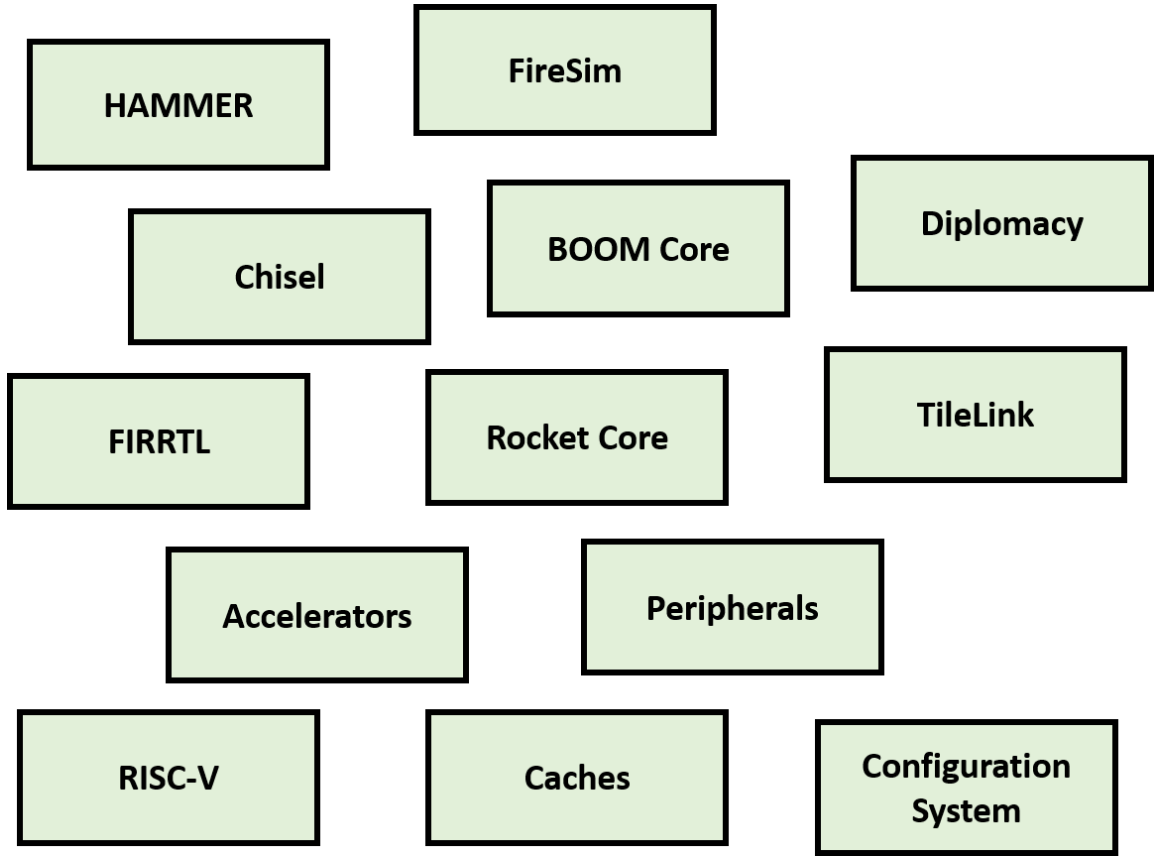
RISC-V Generator-based Chip Design

- RISC-V test chip development at Berkeley
- Evolving agile design methodologies and tooling
- Challenges of scaling test chip complexity
  - Verification and validation
  - Process technology transition
  - Work distribution and tool expertise



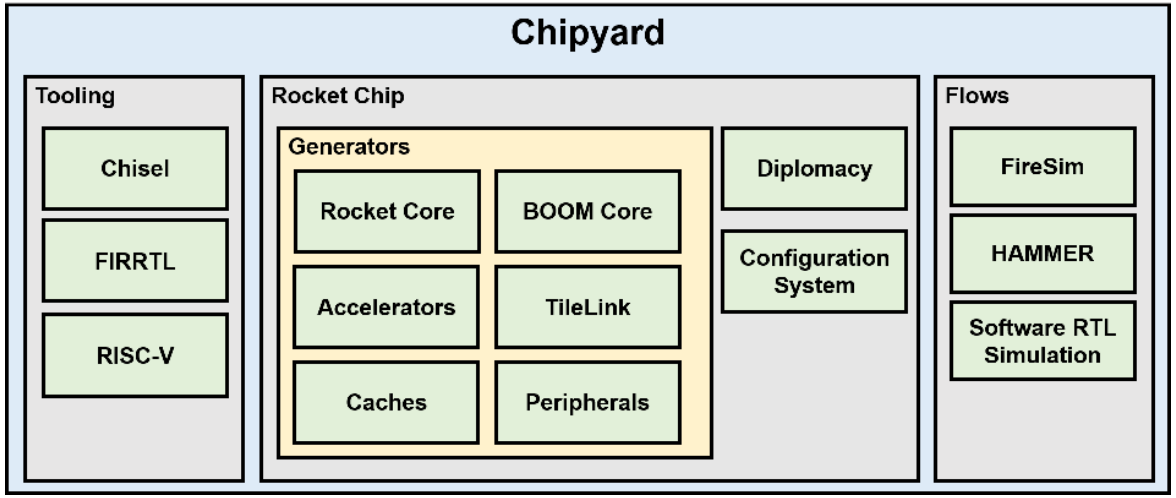
RISC-V Generator-based Chip Design

- Open-source hardware development as a driver
- Challenges of open-source SoC design
  - Organization, discovery, and compatibility
  - Accessibility and design expertise
  - Integration and composability
- Open source IP is not enough
  - Need tools to verify, validate and integrate



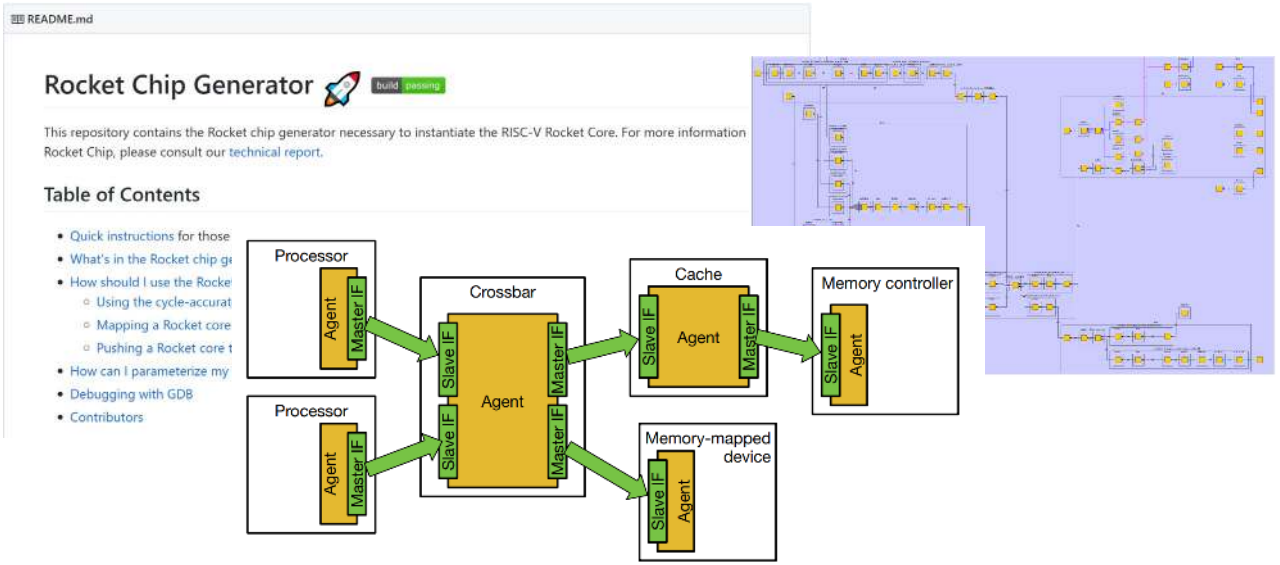
Chipyard

- Chipyard – Integrated SoC Design Simulation and Implementation Environment
- Integrate open-source hardware development tools
  - Single source of hardware description
- Organize and integrate open-source Chisel-based RTL projects
- Enable continuous integration
- Synchronous versioning

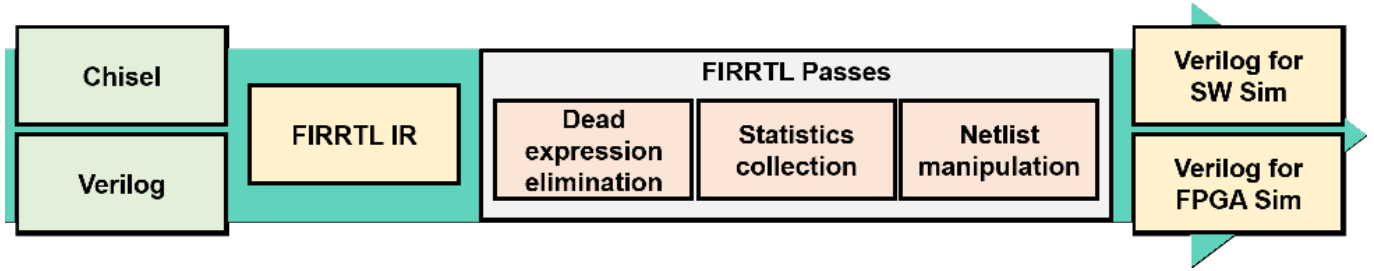


Foundation Components

- Rocket Chip SoC generators
  - Parameter system
  - Component composability
  - SoC system integration
  - TileLink/Diplomacy interconnect

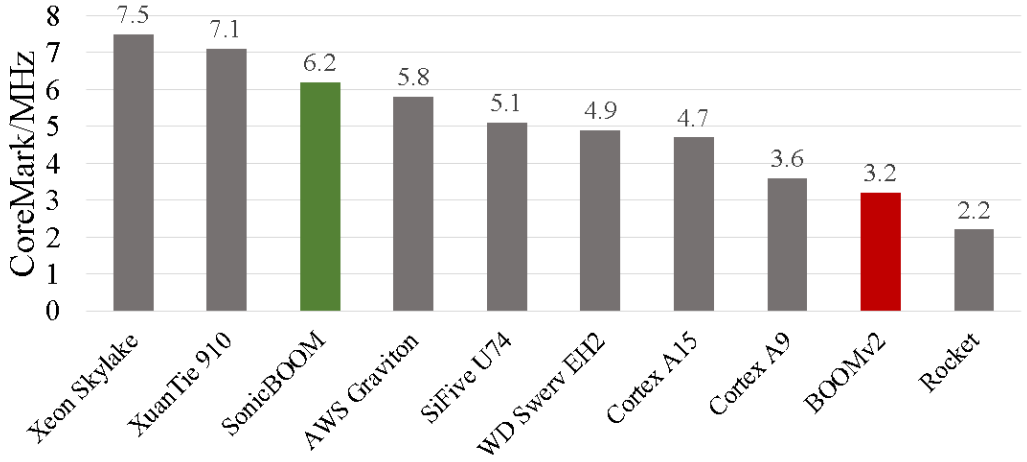


- Chisel / FIRRTL
  - Productivity through object-oriented/functional meta-programming for HDLs
  - IR transforms to enable single-source multi-platform

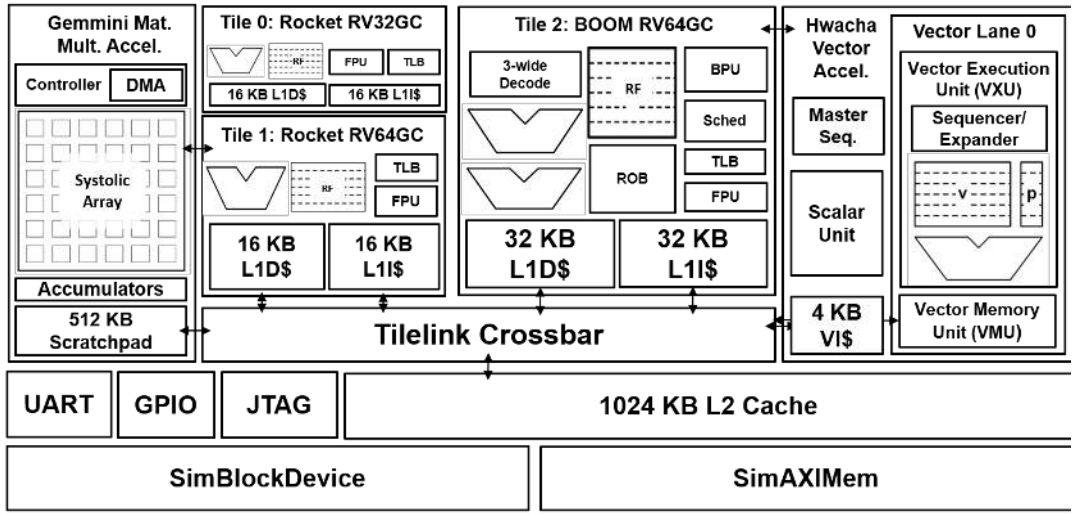


Front End Generators

- Cores
  - Efficiency: Rocket, Ariane
  - Performance: *Sonic*BOOM
    - Fastest Open-Source Core by IPC
    - Reaches 6.2 CoreMark/MHz
- Accelerators
  - Vector (Hwacha)
  - ML (Gemmini, NVDLA)
  - More!
- Peripherals
  - UART, SPI, GPIO, JTAG, More!



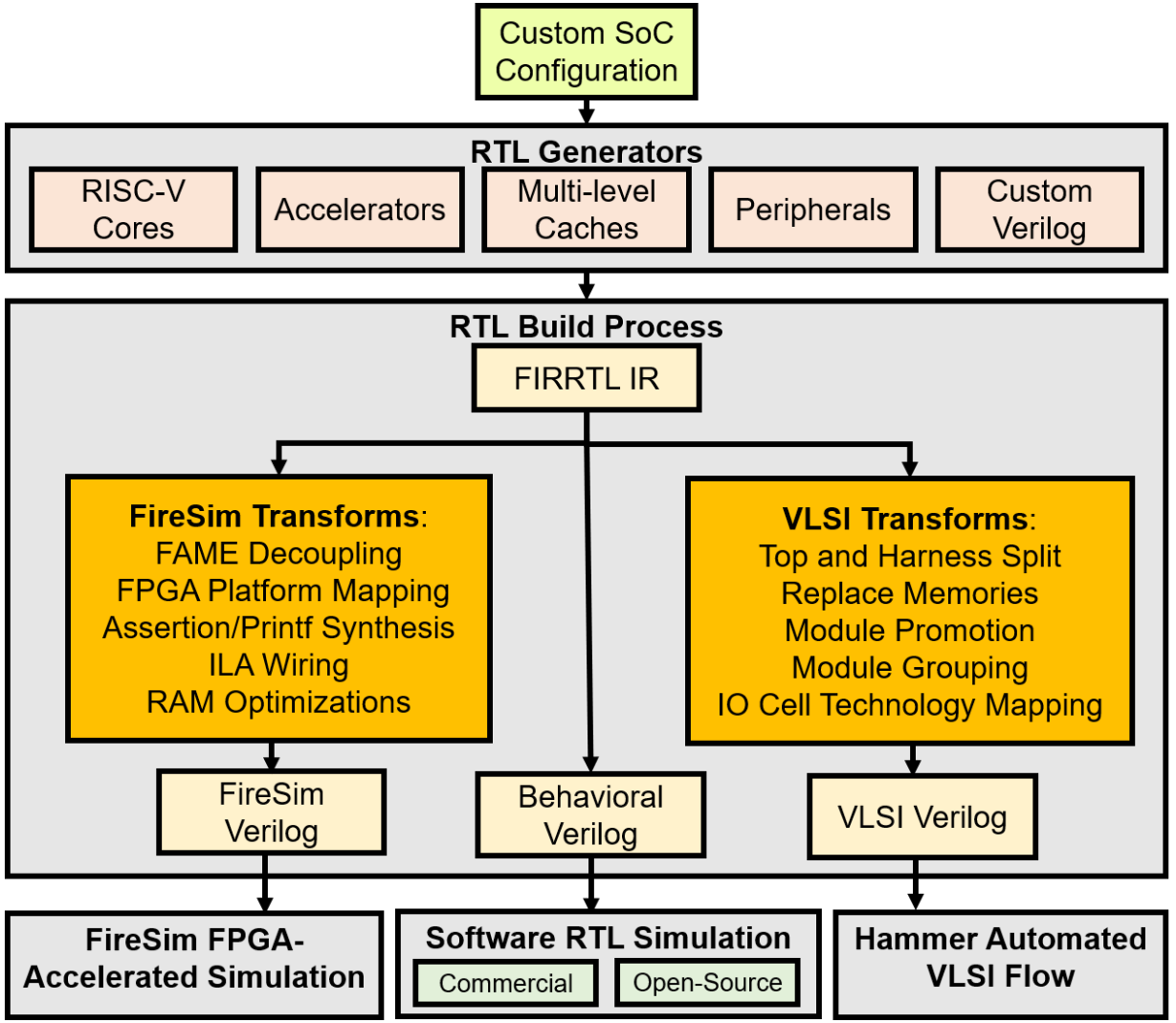
SonicBOOM CoreMark/MHz Comparison



An Example Chipyard Configuration

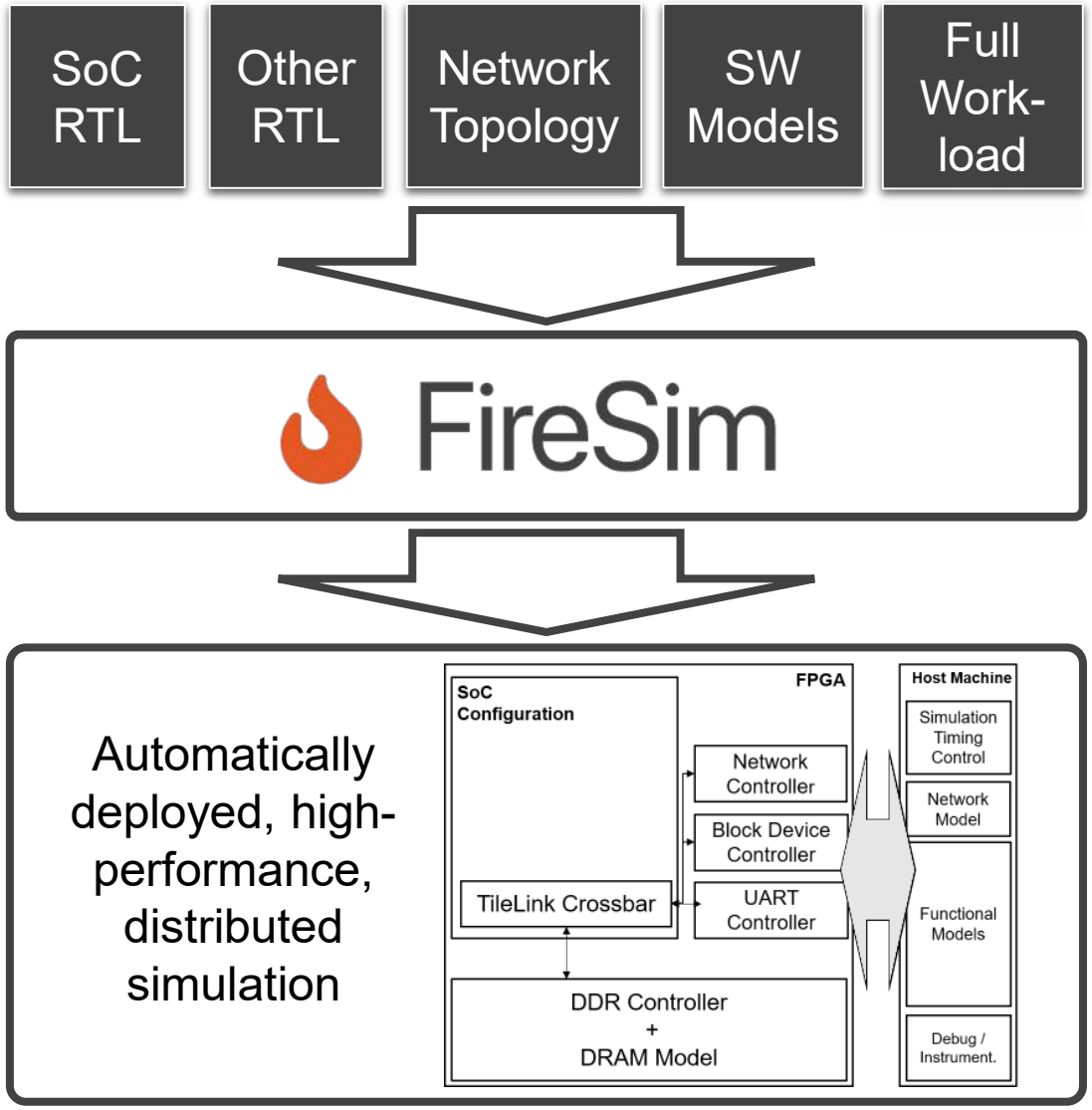
Single-Source Design Flow

- Single source of truth
  - Continuous validation/integration
- Targets for various stages of the design flow
  - Design cycle RTL simulation
  - Verification / Validation
  - VLSI flow
- Different collateral for distinct targets
- High-level constructs lowered to target-specific Verilog through automated transformations
  - Memories, IOs, Instrumentation



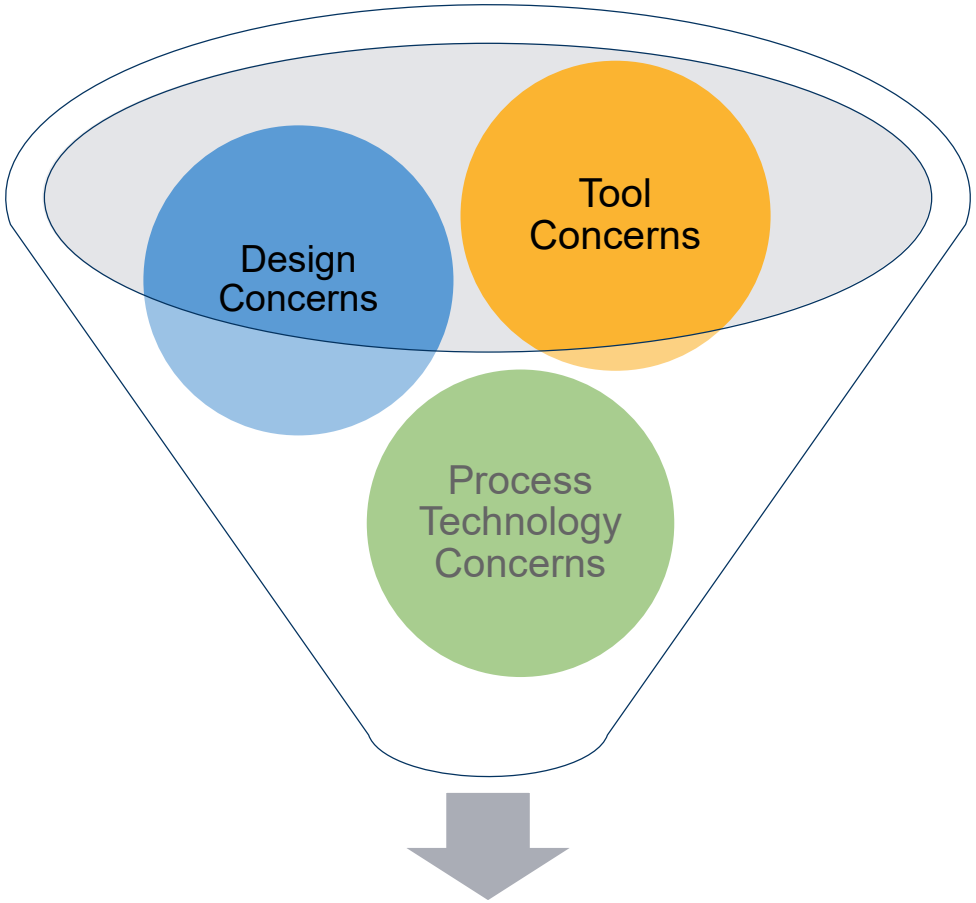
FireSim

- FPGA-accelerated simulation on the Amazon EC2 public cloud
- Originally developed for scale-out cycle-exact simulation
  - Extensive automation
- Not FPGA prototypes, rather FPGA-accelerated simulators
  - Deterministic peripheral modeling
- Cloud-hosted FPGAs
  - Inexpensive, elastic supply of large FPGAs
  - Easy to collaborate



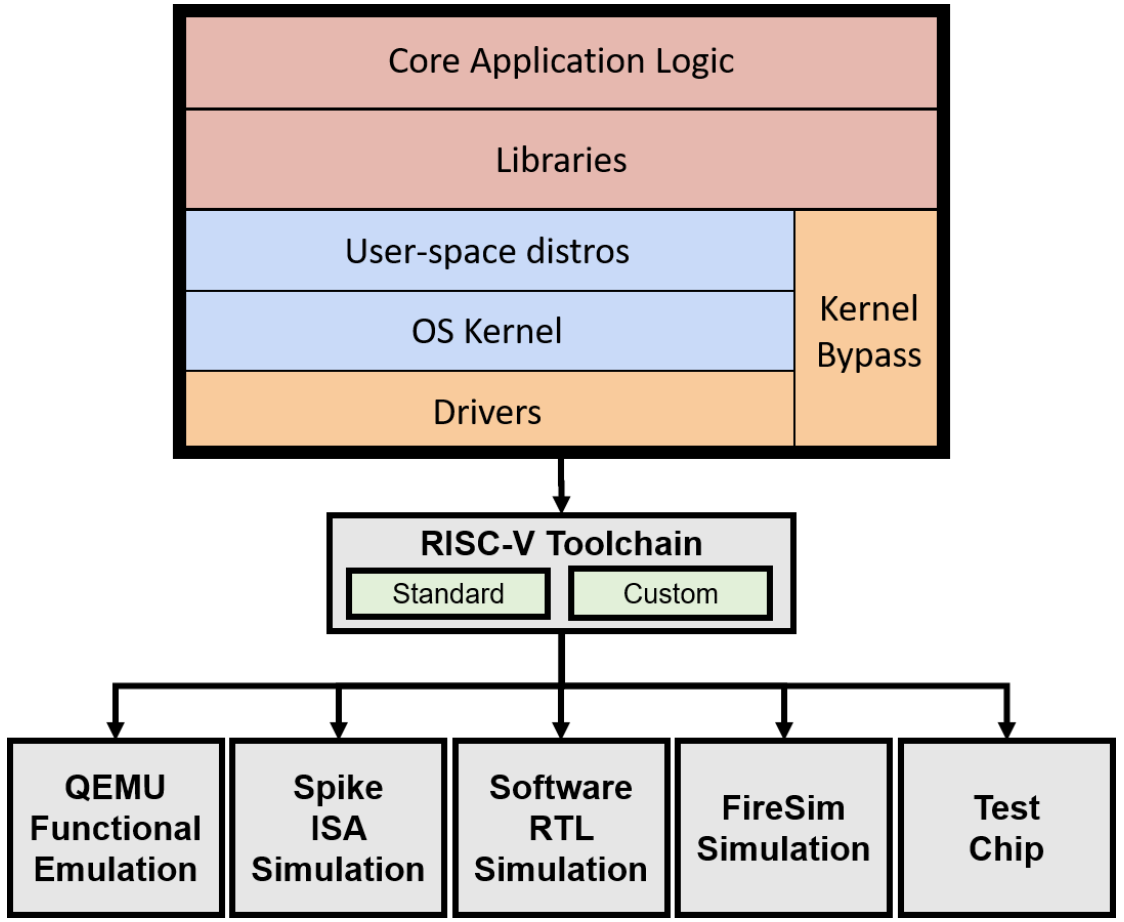
Hammer

- Modular VLSI flow
  - Allow reusability
  - Allow for multiple “small” experts instead of a single “super” expert
  - Build abstractions/APIs on top
  - Improve portability
  - Improve hierarchical partitioning
- Three categories of flow input
  - Design-specific
  - Tool/Vendor-specific
  - Technology-specific



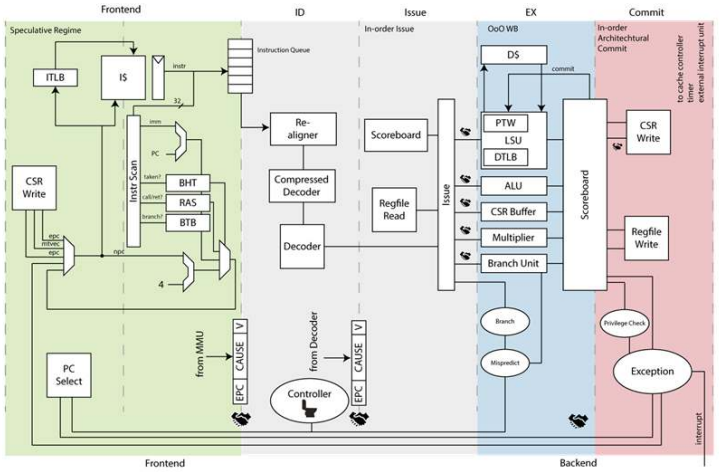
Software

- Standard versioned RISC-V tools
  - GNU toolchain, Spike, QEMU
- Non-standard ESP tools with custom accelerator extensions
  - Interchangeable with RISC-V tools
- Libgloss based bare-metal flow
- FireMarshal workload management
  - Version-controlled software workload descriptions format
  - Shared software development
  - Automated software execution on different simulation targets

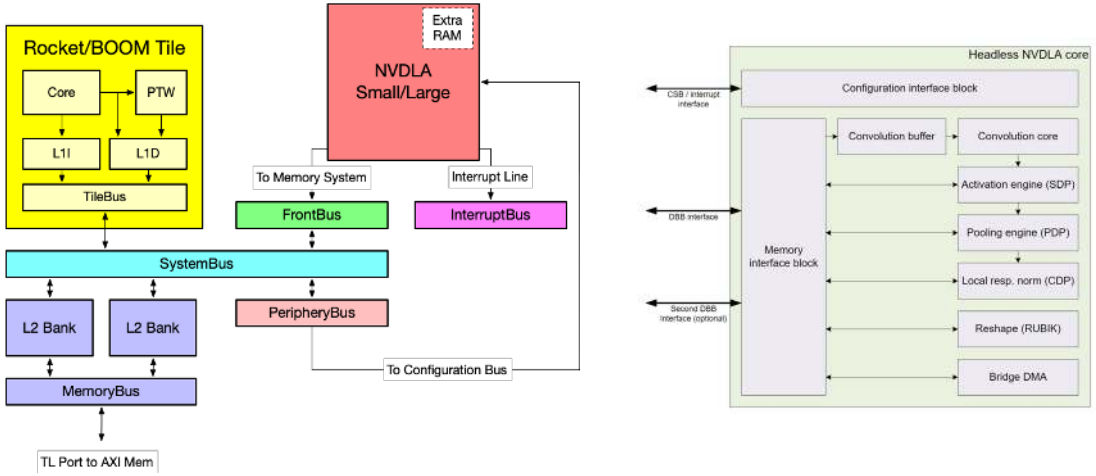


Open Source Verilog IP Integration

- Provide pre-integrated SoC configurations including popular third-party open-source Verilog IP
- Chisel BlackBox primitives
- Ariane Core (PULP/ETH)
  - RV64GC 6-stage in-order processor
  - Plugs-in as a Tile (like Rocket/BOOM)
  - SystemVerilog
- NVDLA (NVIDIA)
  - Open-source deep learning accelerator from NVIDIA
  - Memory-mapped peripheral



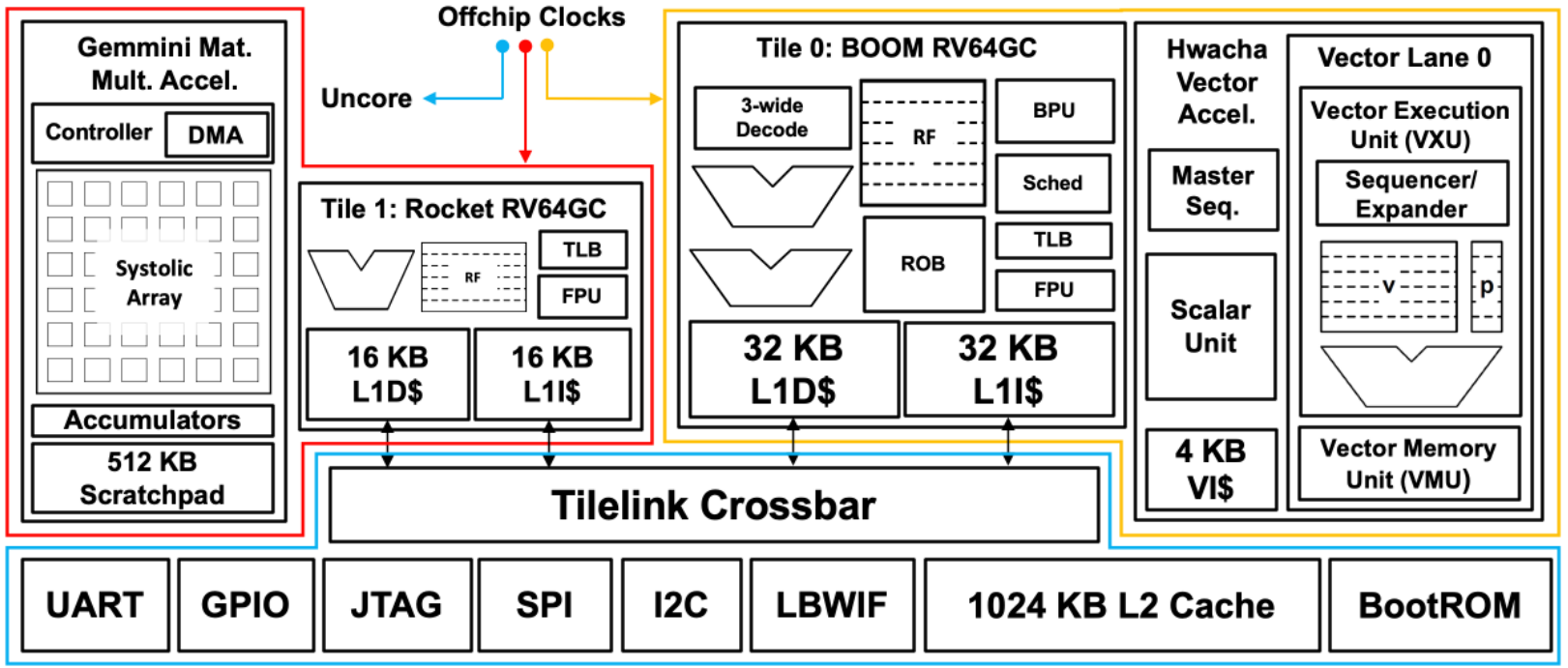
Source: https://github.com/pulp-platform/ariane



Source: http://nvdla.org/primer.html

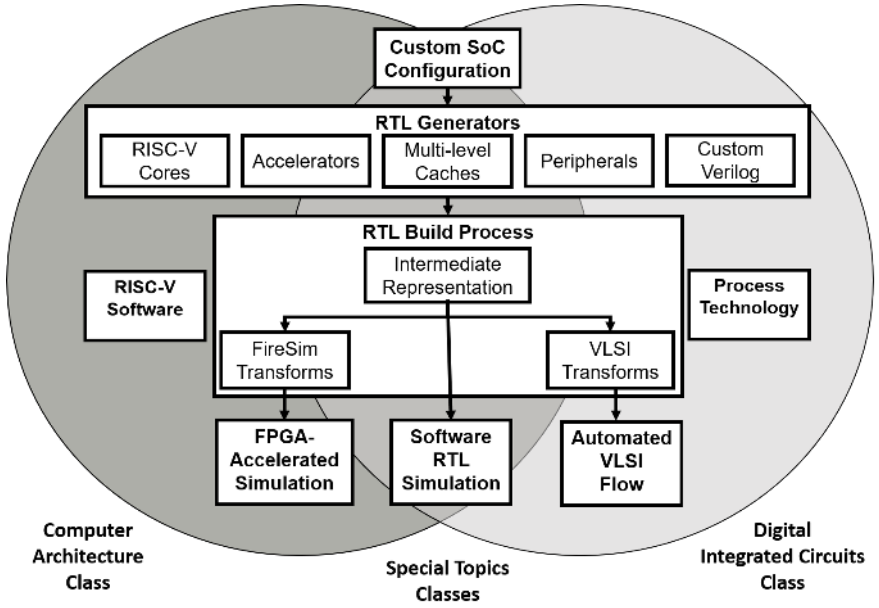
BEAGLE Test Chip

- Heterogeneous Multi-Core Multi-Accelerator Chip in Intel 22FFL
- Collaboration with Intel Labs, designed using Chipyard
- Features
  - BOOMv2 Core with Vector Accelerator
  - Rocket Core with ML Accelerator
  - 1MB Shared L2
  - GPIO, I2C, UART, SPI, JTAG, SerDes
  - Multi-clock (3 domains)
- Currently being tested



Community and Education

- Documentation (over 100 pages)
  - chipyard.readthedocs.io
- Tutorials, Mailing list
- Education - Used in 3 Berkeley classes (concurrently)
  - Computer Architecture and Engineering
  - Advanced Digital Integrated Circuits
  - Hardware for Machine Learning
- Common assignment infrastructure
  - Amortize student ramp-up
  - Connecting thread between EE and CS



Summary

- Open source project:
  - Github: https://github.com/ucb-bar/chipyard/
  - Docs: https://chipyard.readthedocs.io/en/latest/index.html
  - Mailing List: https://groups.google.com/forum/#!forum/chipyard
- Integrated design, simulation and implementation framework
- BEAGLE as an example heterogenous SoC developed using Chipyard