Chipyard

Berkeley
Architecture
Research

CHIPYARD

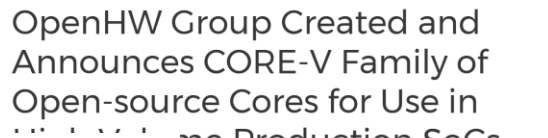
Trends in Open Source Hardware

- Organization/Specifications: RISC-V, CHIPS Alliance, OpenHW
- Community: LowRISC, FOSSi
- Academia: PULP Platform, OpenPiton, ESP
- Government: DARPA POSH
- Industry: WD SWERVE, NVIDIA NVDLA
- Tools: Verilator, Yosys, OpenRoad
- Fabrication: Skywater 130nm

RISC-V

CHIPS ALLIANCE

FOSSi Foundation



OPENHW GROUP
PROVEN PROCESSOR IP

VERILATOR

Building An Open Source RISC-V System

Have you heard of this Free and Open RISC-V thing? It should be so easy to build real systems now

Cool! I want to build an Open-Source custom RISC-V SoC.
What do I need to do?

I think I heard of some stuff from Berkeley (Rocketchip? Chisel?), also OpenPiton, and PULP

Building An Open Source RISC-V System

- Processor core IP
- Supporting system IP (memory system, peripherals, buses, etc.)
- Integrate custom blocks
- Write appropriate software
- Verify using bare-metal simulation
- Validate full-system
- Physical design
- Test environment
- Fabrication

Chipyard – The Real Story

Man… I've been here for 3 years, and I still can't figure out how all these chisel- and rocketchip-based project work together.
Whenever I need to build anything (hardware, software, sim) I just use the firesim infrastructure

I know!!! I just started this year, an d I'm already a BOOM maintainer. We totally depends on interactions and interfaces with a gazillion other repos and projects that are never synced

Chipyard – The Real Story

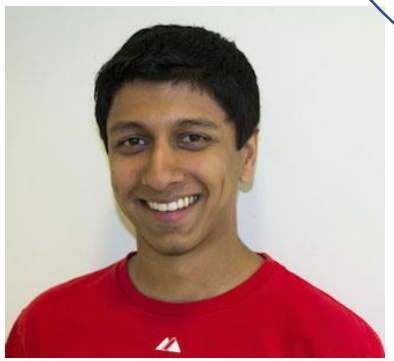
We're doing an ADEPT "offsite" in a place far far away…. The I-House. Everyone should bring topics to talk about

Chipyard – The Real Story

Brainstorming for I-House event!
What should we talk about?

I want to complain about compilers and build systems!

That's nice, but….
You know, we have all these cool projects that we use to build chips, but nobody else knows they work together.
We need some kind of joint landing page or website!



Classic

Chipyard – The Real Story

Totally

Agreed

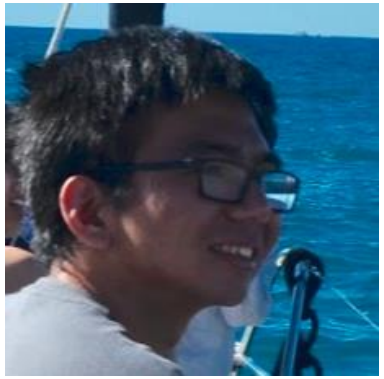
Yup, that makes sense

Maybe we should do something about it?



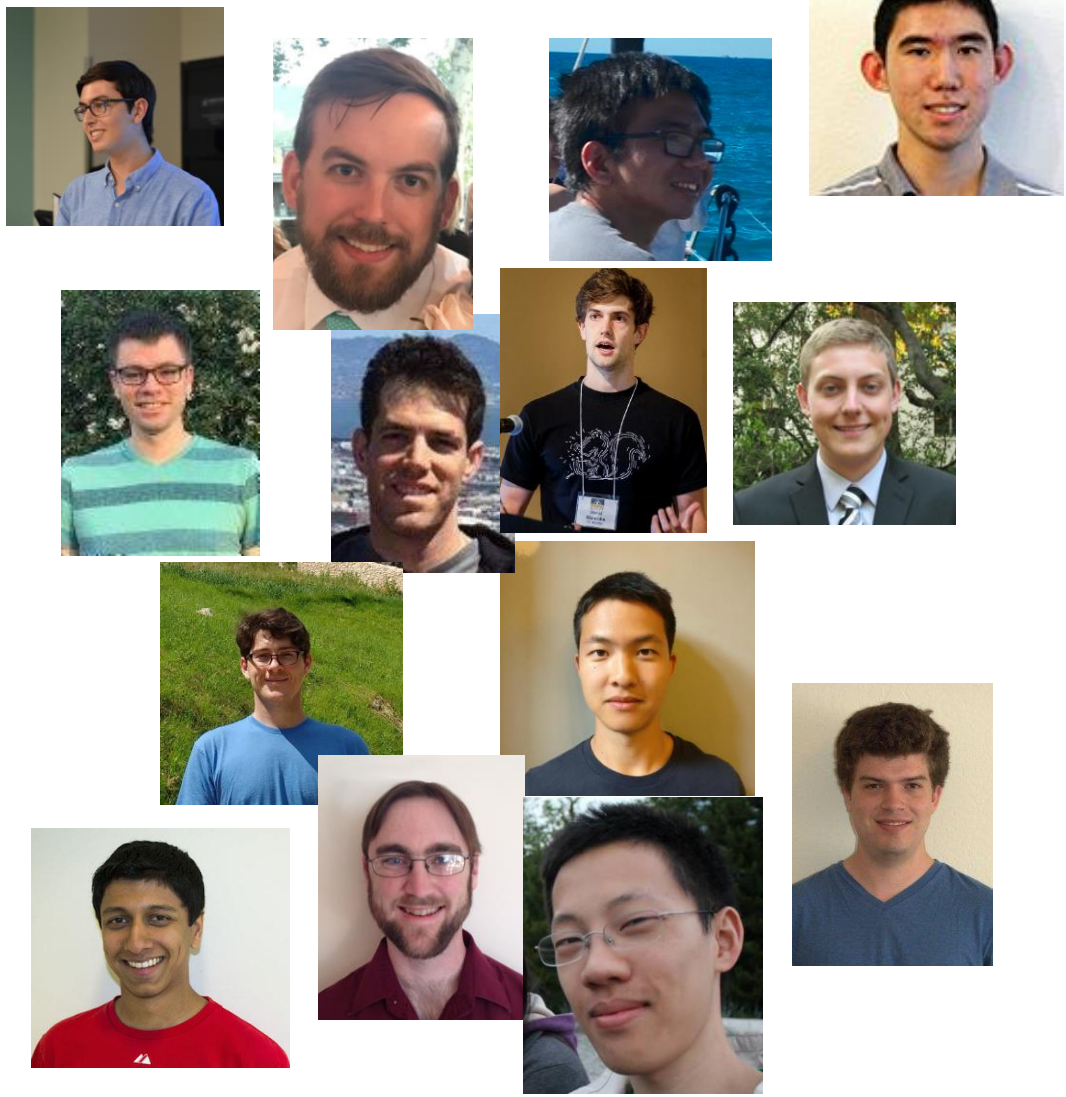






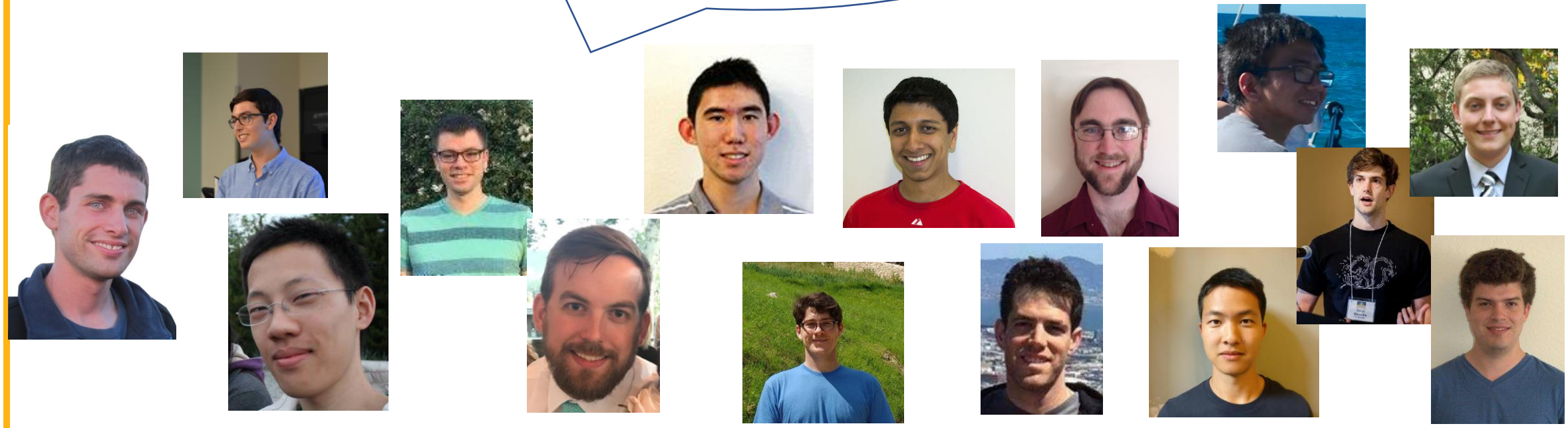
Chipyard – The Real Story

We hereby convene the great REBAR council in Soda 511. Please join!
(Warning – you will be micro-managed to pieces)



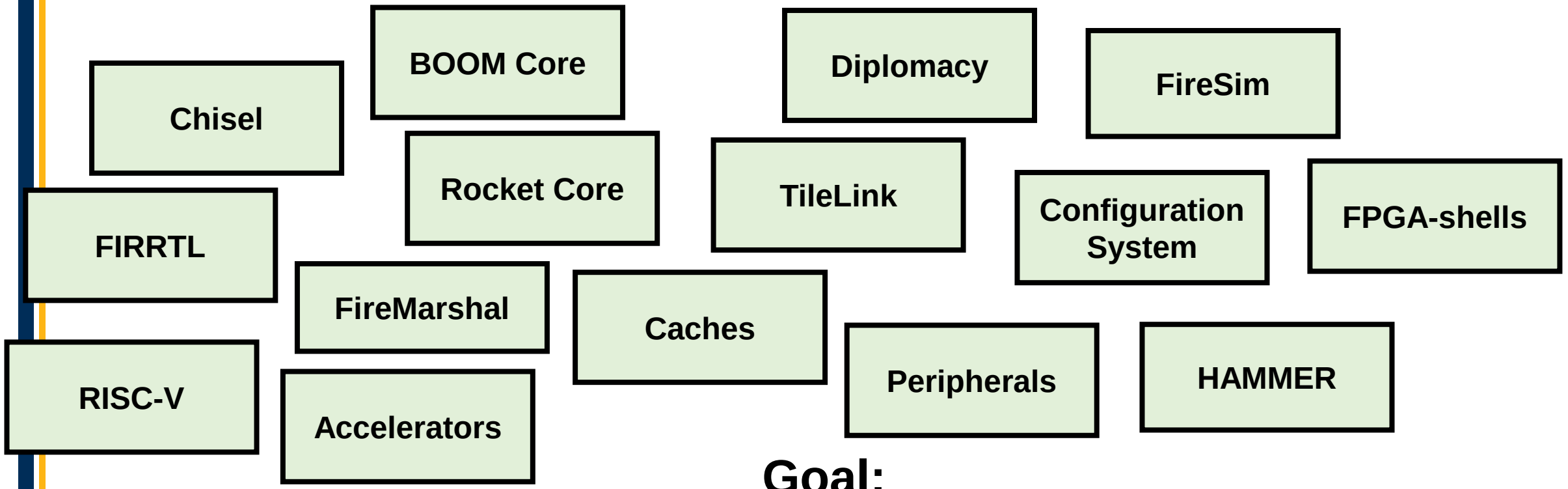
Chipyard – The Real Story

First order of business:
We hate the name REBAR
(but also, mono-repo vs. individual independent modules)




Building An Open Source RISC-V System

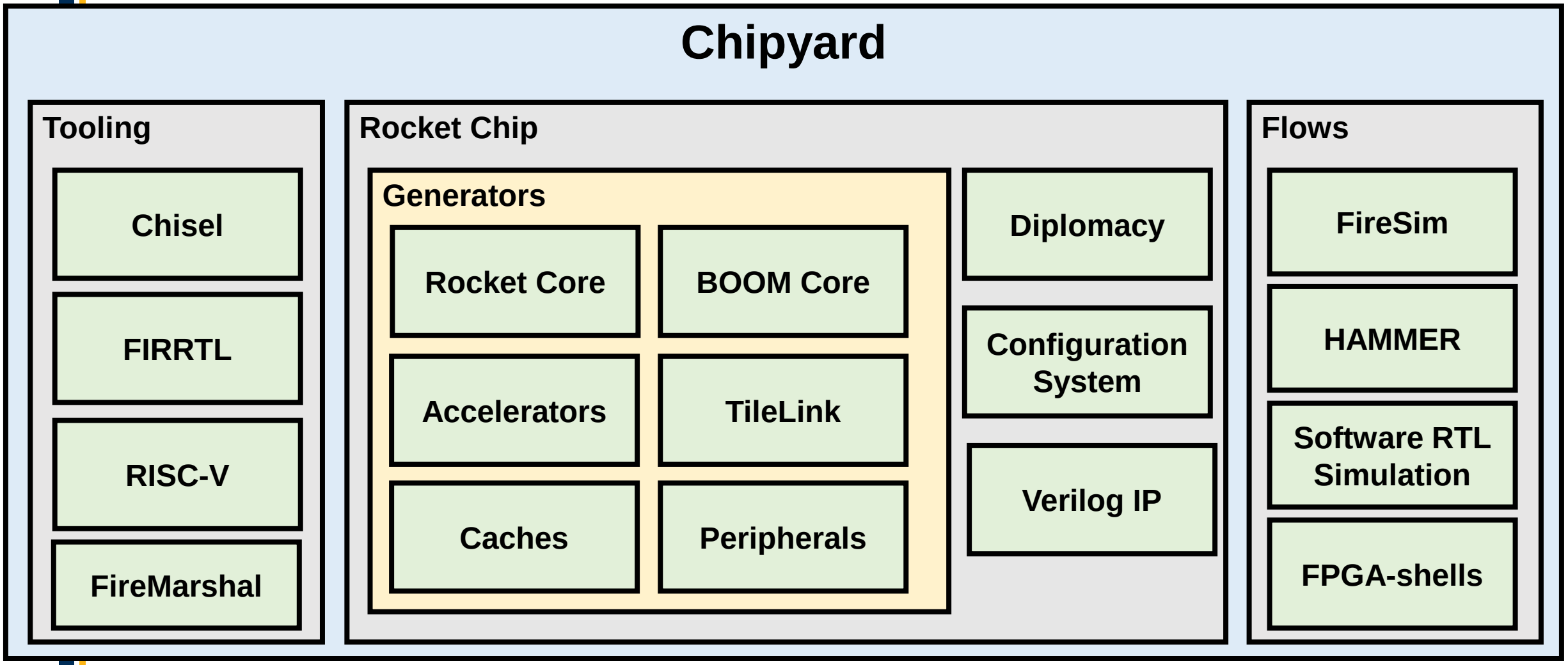
A lot of RISC-V & generator-related open source hardware projects out there



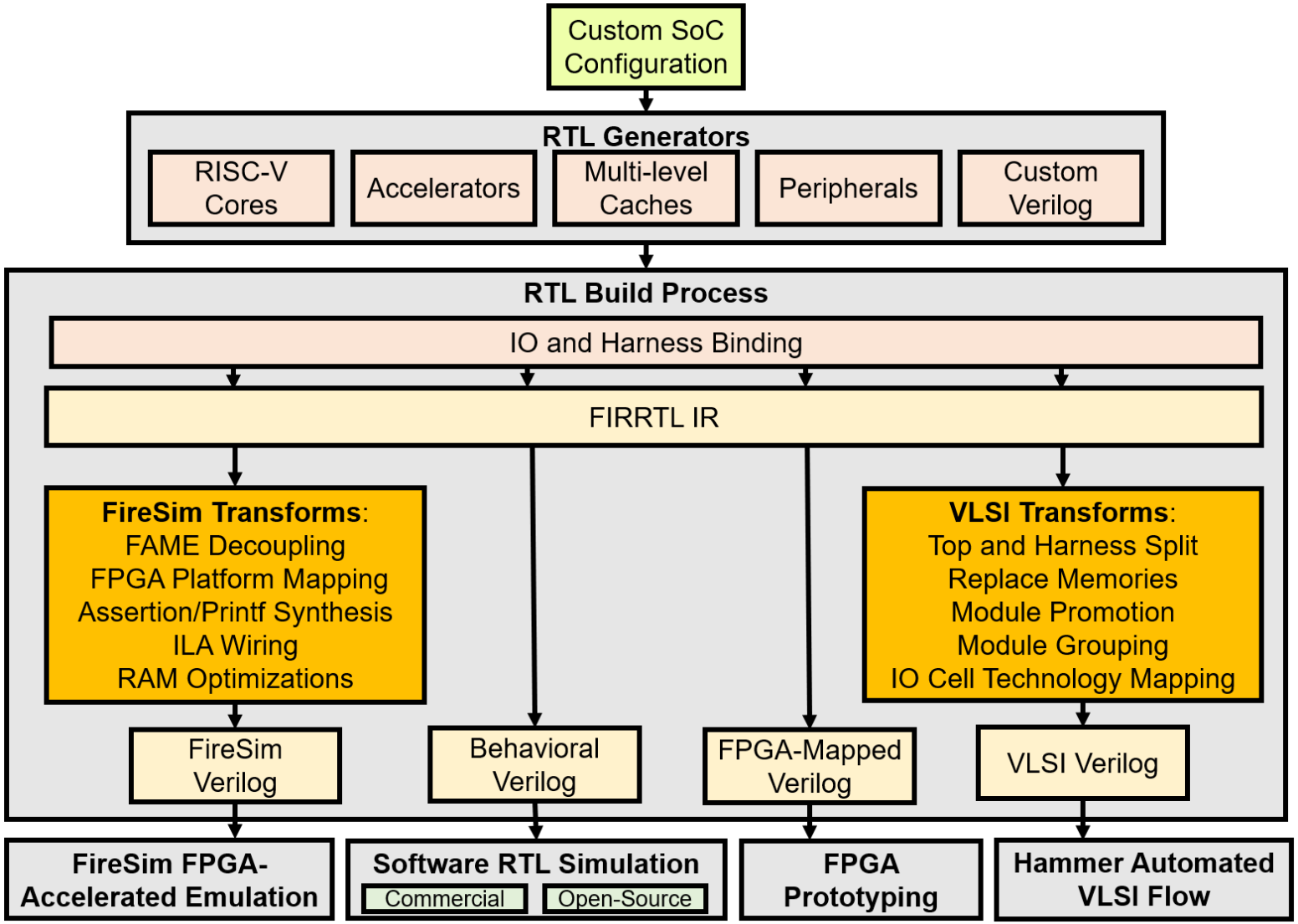
Goal:

Make it easy for small teams to
design, integrate, simulate, and **tape-out** a custom SoC

Chipyard

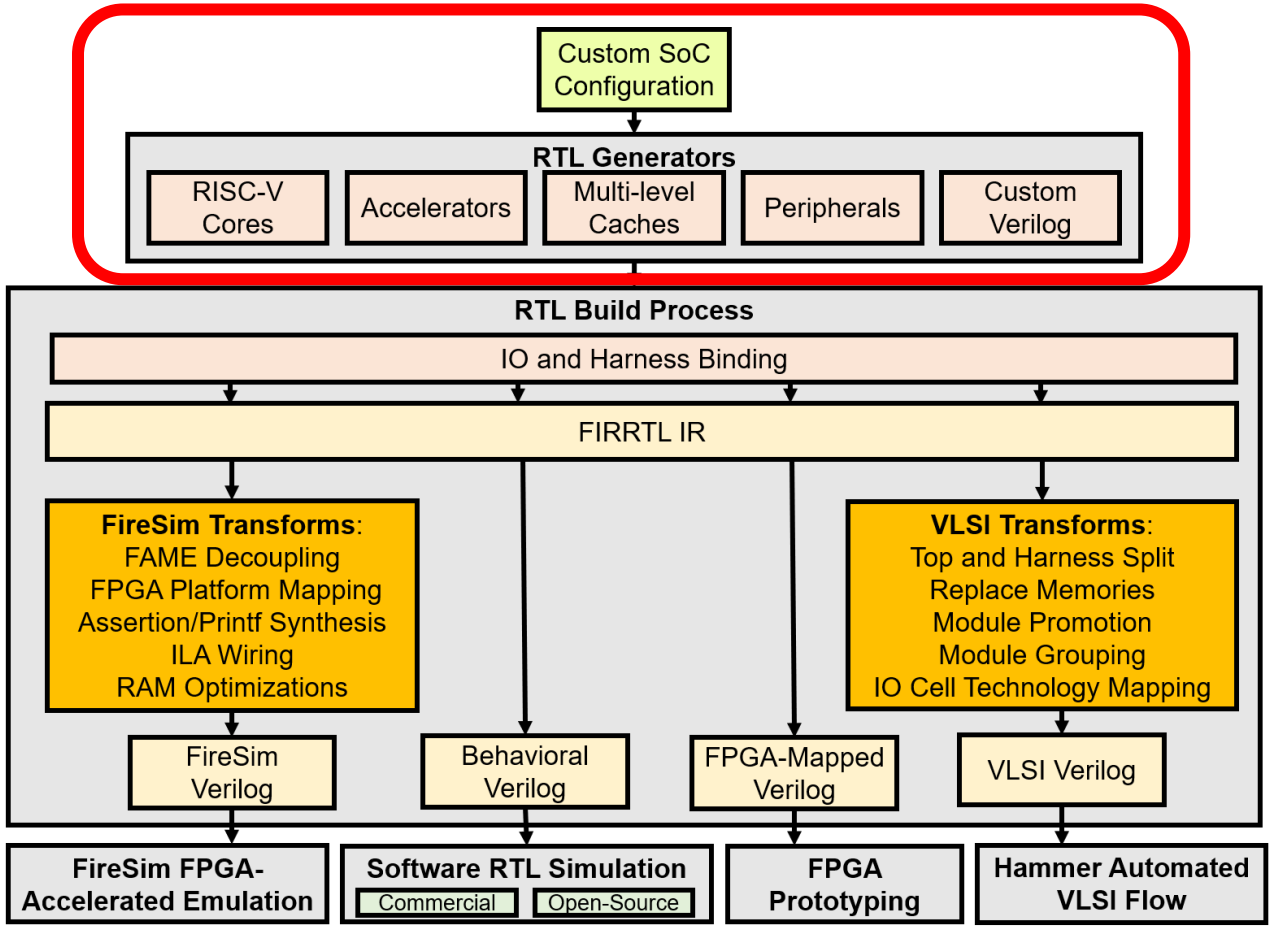

Chipyard
Tooling
Chisel
FIRRTL
RISC-V
FireMarshal
Rocket Chip
Generators
Rocket Core
BOOM Core
Accelerators
TileLink
Caches
Peripherals
Diplomacy
Configuration System
Verilog IP
Flows
FireSim
HAMMER
Software RTL Simulation
FPGA-shells

How is this integrated? Generators!



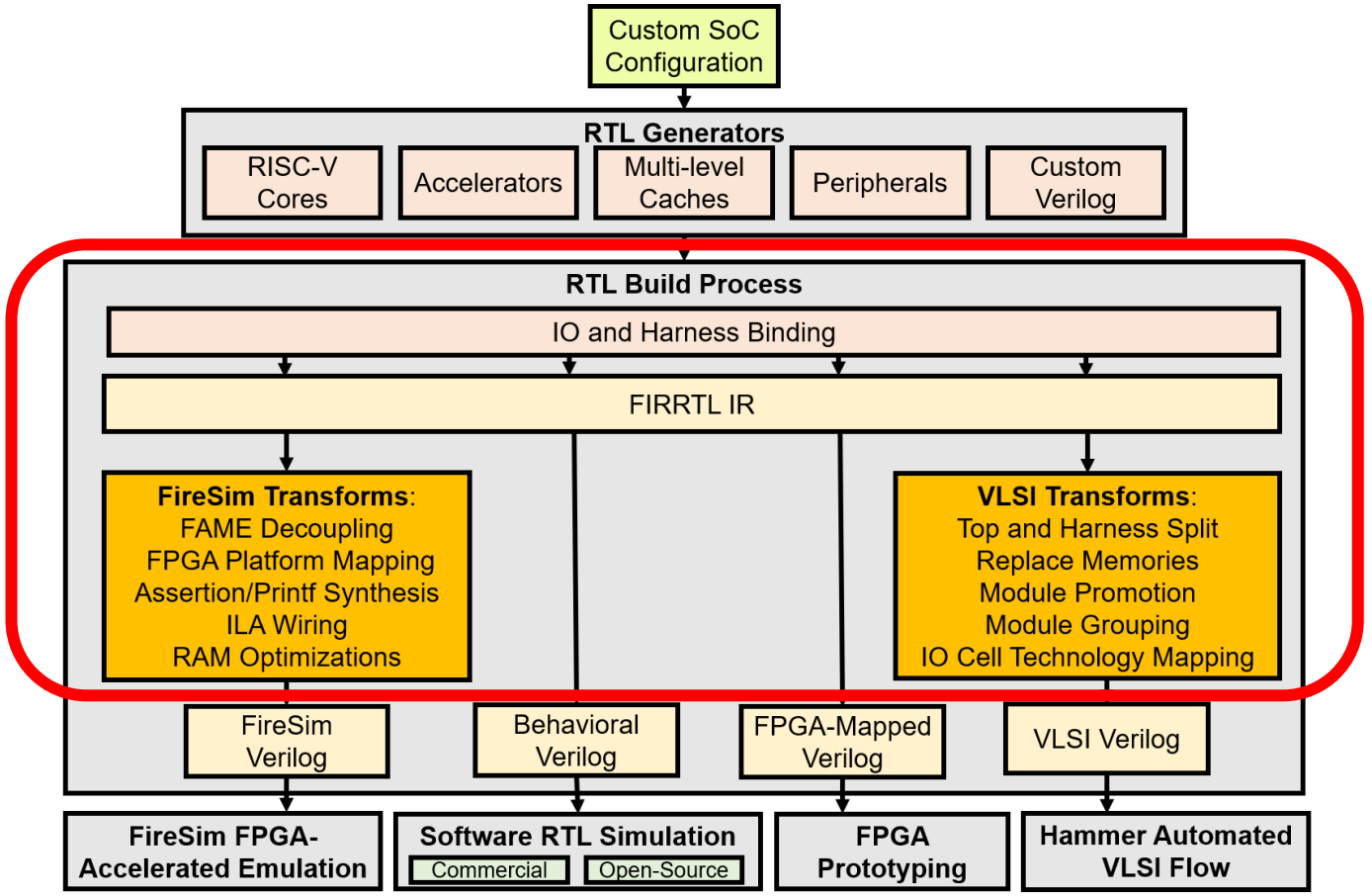
How is this integrated? Generators!

- Everything starts from a generator configuration
- Generators written in Chisel
- Generator SoC basic component libraries (enable integration)
  - Rocket Chip
  - Diplomacy
- Higher level generator libraries: BOOM, Inclusive Cache, SiFive Blocks, Accel.
- Generators can integrate third-party Verilog instance IP
- Generators lead from IP to design flows



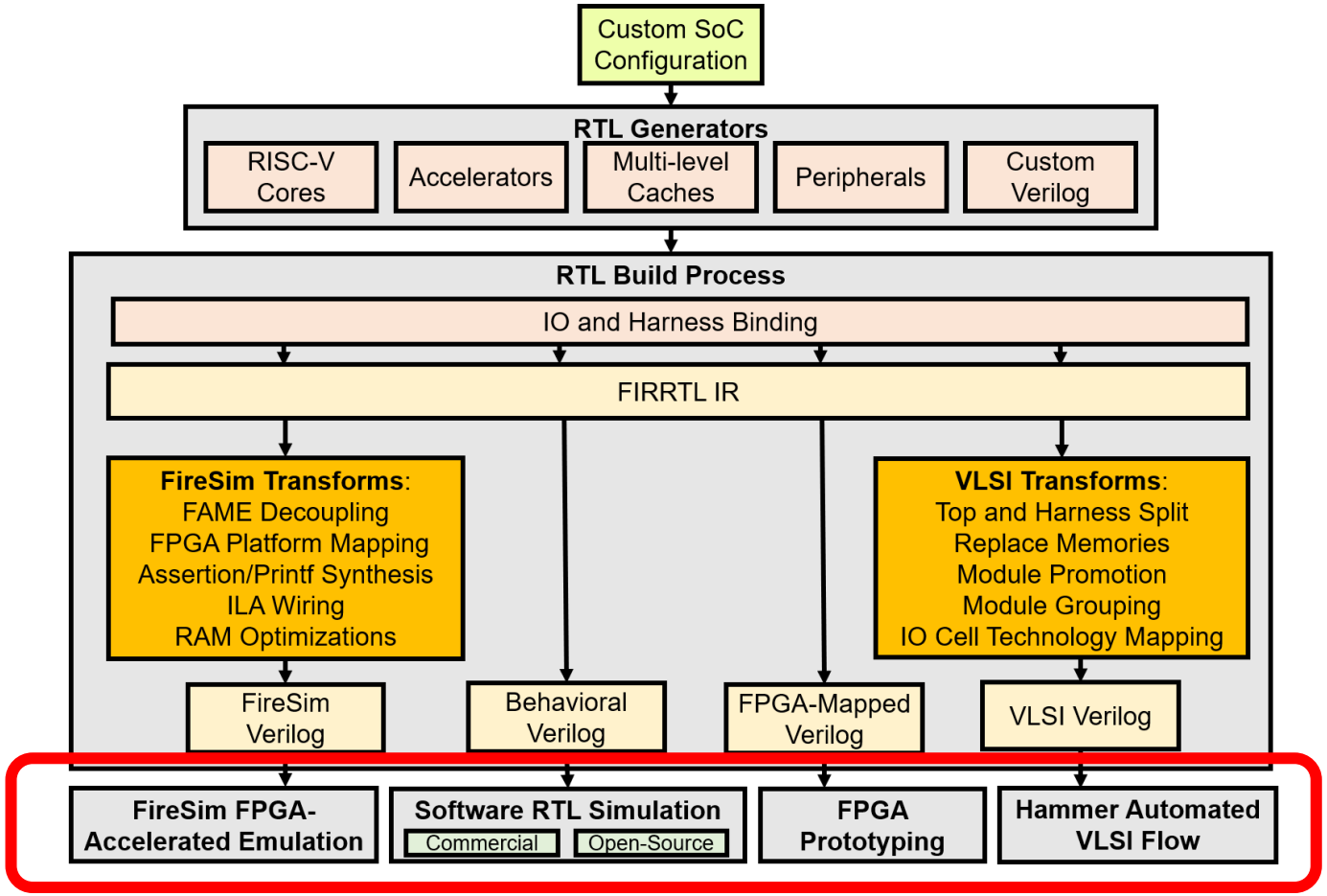
How is this integrated? Generators!

- Elaboration and Transformation
- Internals: FIRRTL – IR enables automated manipulation of the hardware description
- Externals: I/O and Harness Binders – pluggable interface functions enable automated targeting of different external interface requirements



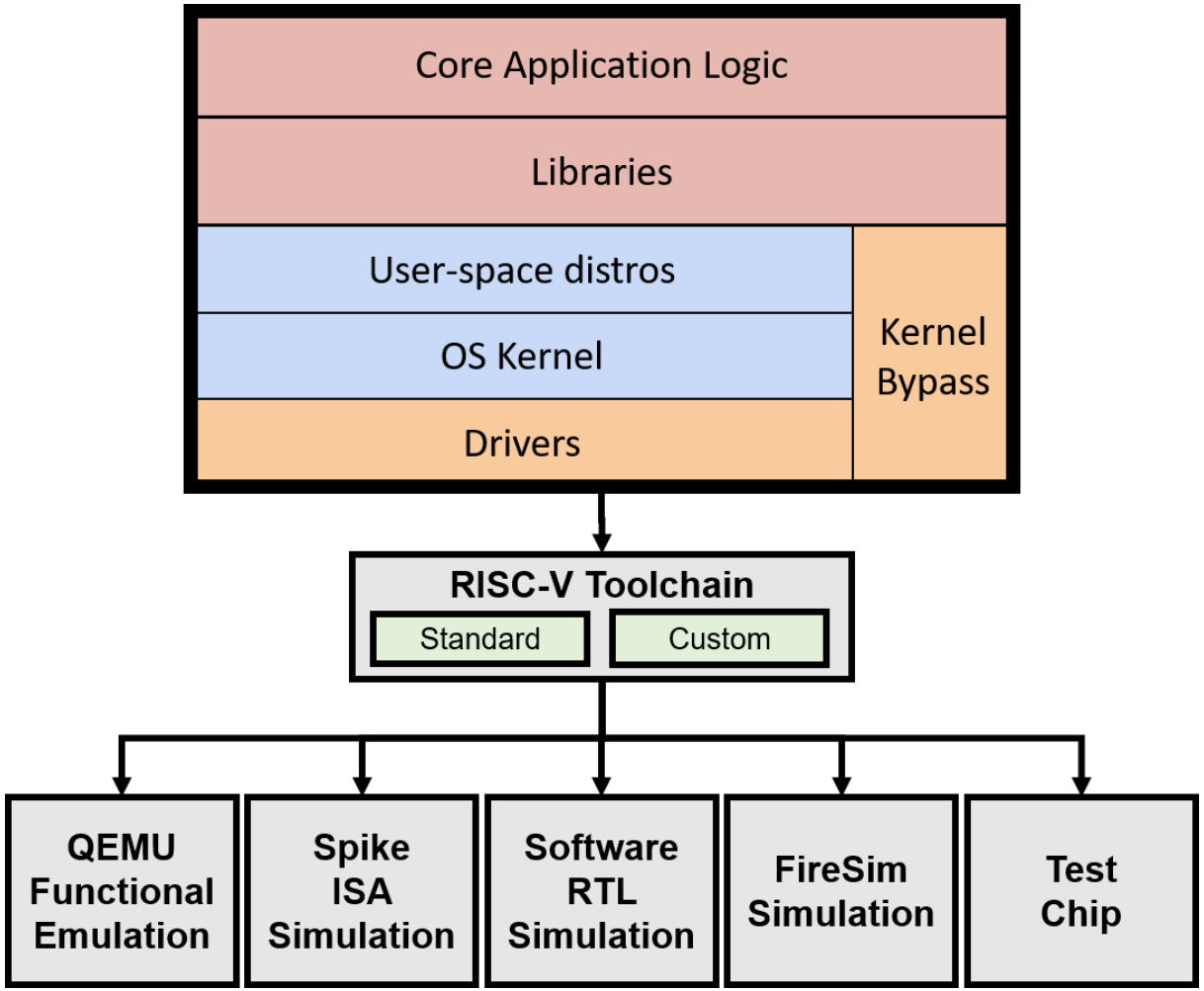
How is this integrated? Generators!

- Design flows
  - Software RTL Simulation
  - FPGA-Accelerated Emulation
  - FPGA Prototyping
  - VLSI Implementation
- Makefile based automation of transition between design flows
- Flow-specific collateral generation (harnesses, drivers, configuration and constraint files, etc.)

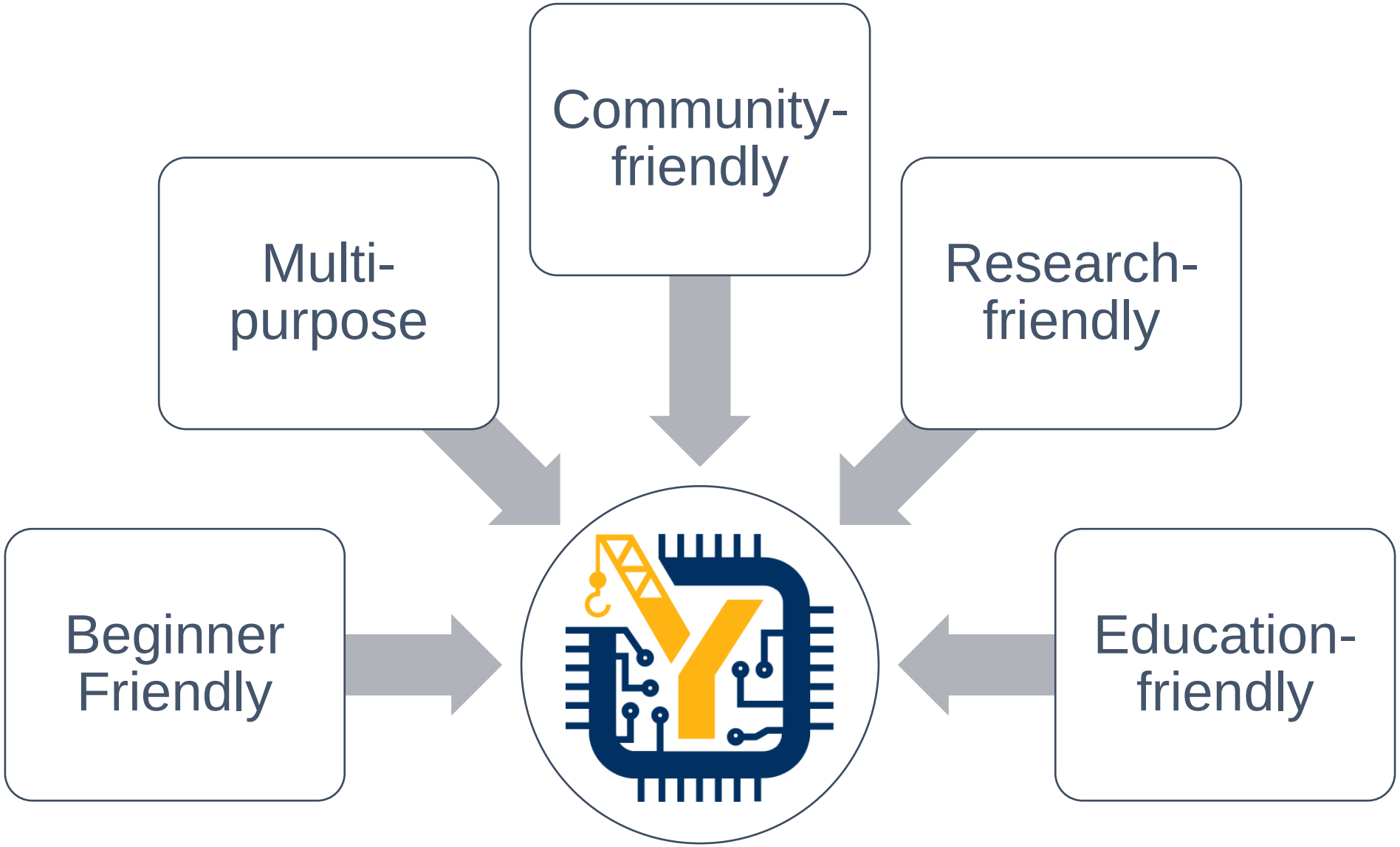


Software

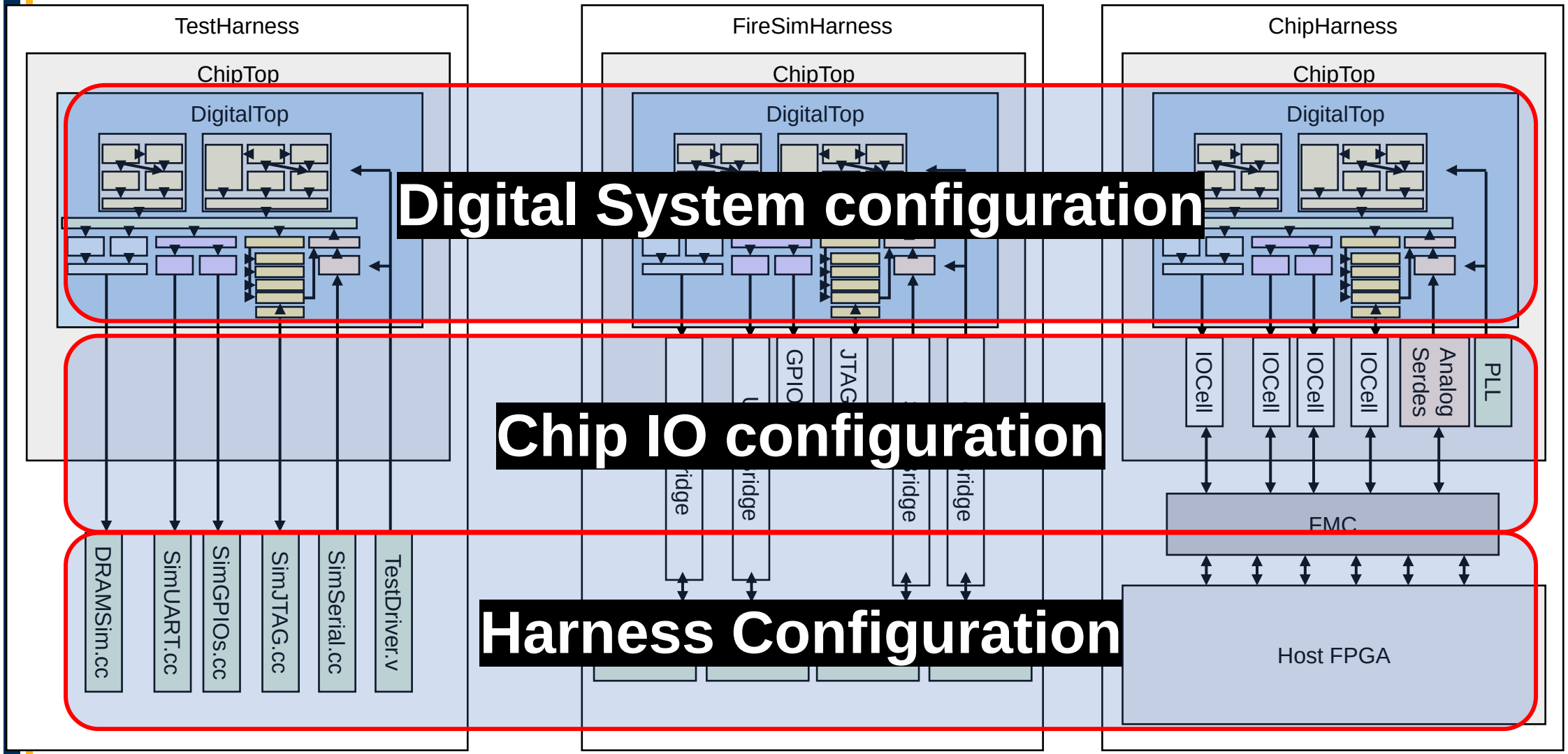
- Hardware alone is not enough
- Custom SoCs require custom software
- Different platforms require different firmware
- Chipyard codifies custom software handling
  - Toolchains
  - Reproducible software generation and management flows using FireMarshal



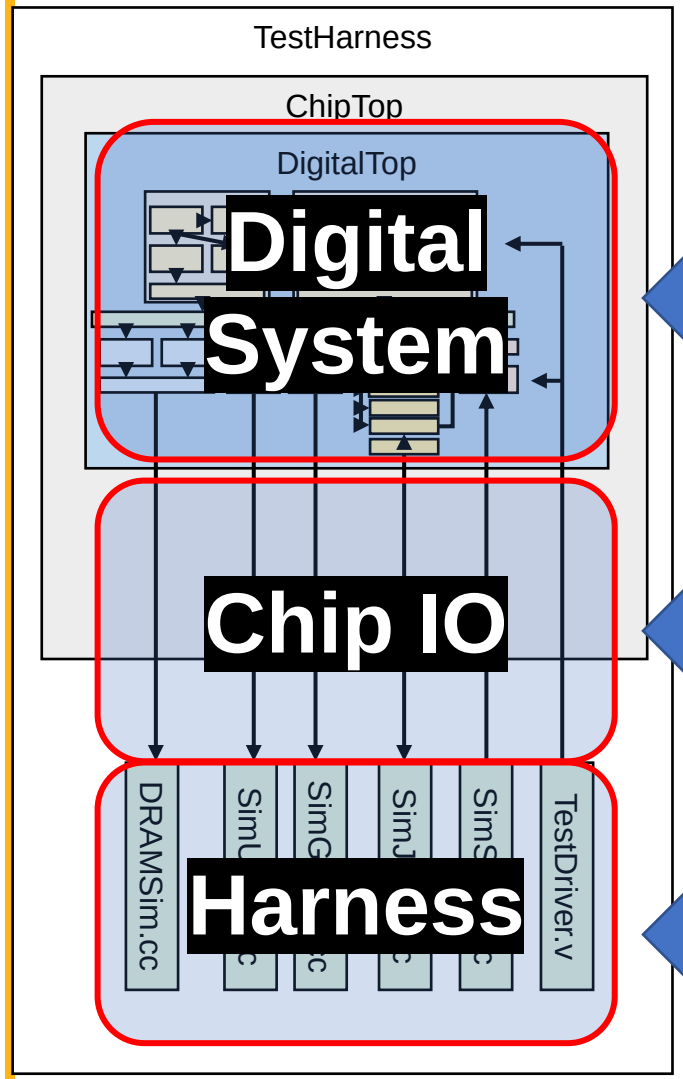
Chipyard Goals



Multipurpose



A Complete Config




```
class CustomConfig extends Config(
  new WithDefaultGemmini ++
  new WithNRocketCores(1) ++
  new WithNBoomCores(1) ++
  new WithBootROM ++
  new WithUART ++
  new WithJtagDTM ++
  new WithGPIOs ++
  new WithInclusiveCache(512) ++

  new WithPassThroughIOs ++


  new WithDRAMSim ++
  new WithSimUART ++
  new WithSimJTAG ++
  new WithSimSerial

)
```

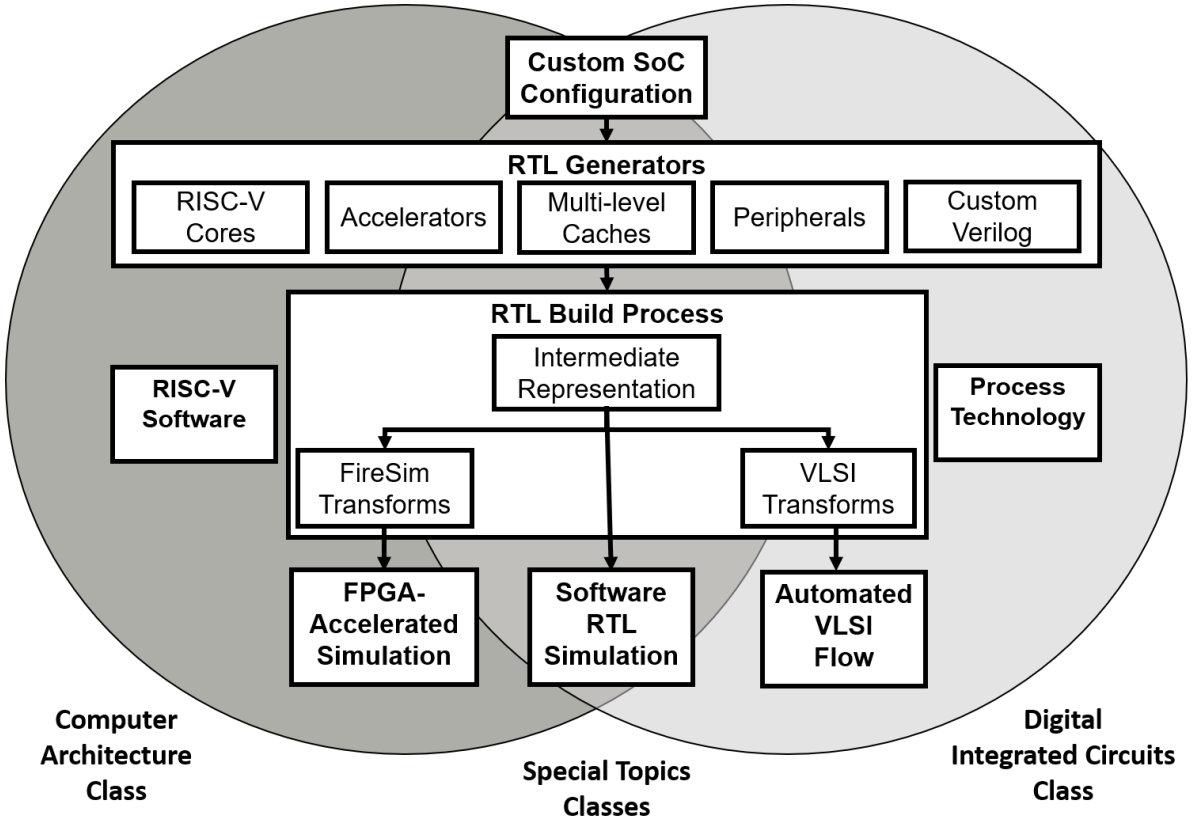
Education

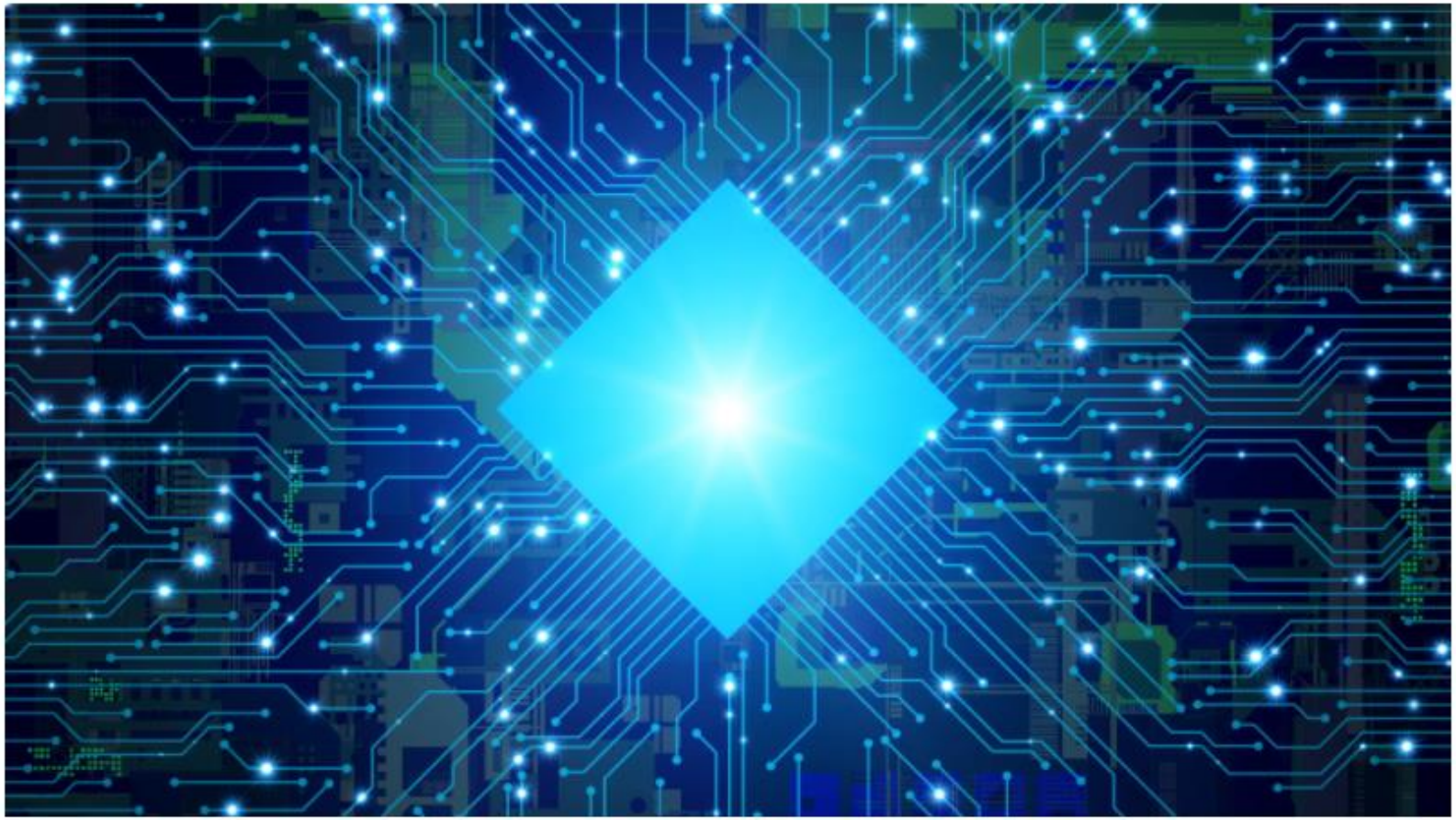
Used in multiple Berkeley grad/ugrad courses

- Hardware for Machine Learning
- Undergraduate Computer Architecture
- Graduate Computer Architecture
- Advanced Digital ICs
- Tapeout HW design course

Common shared HW framework

- Reduced ramp-up time for students
- Students learn framework once, reuse it in later courses
- Enables more advanced course projects (tapeout a chip in 1 semester)





Berkeley Engineering students pull off novel chip design in a single semester. The class shows successful model for expanding entry into field of semiconductor design

Berkeley engineering students pull off novel chip design in a single semester

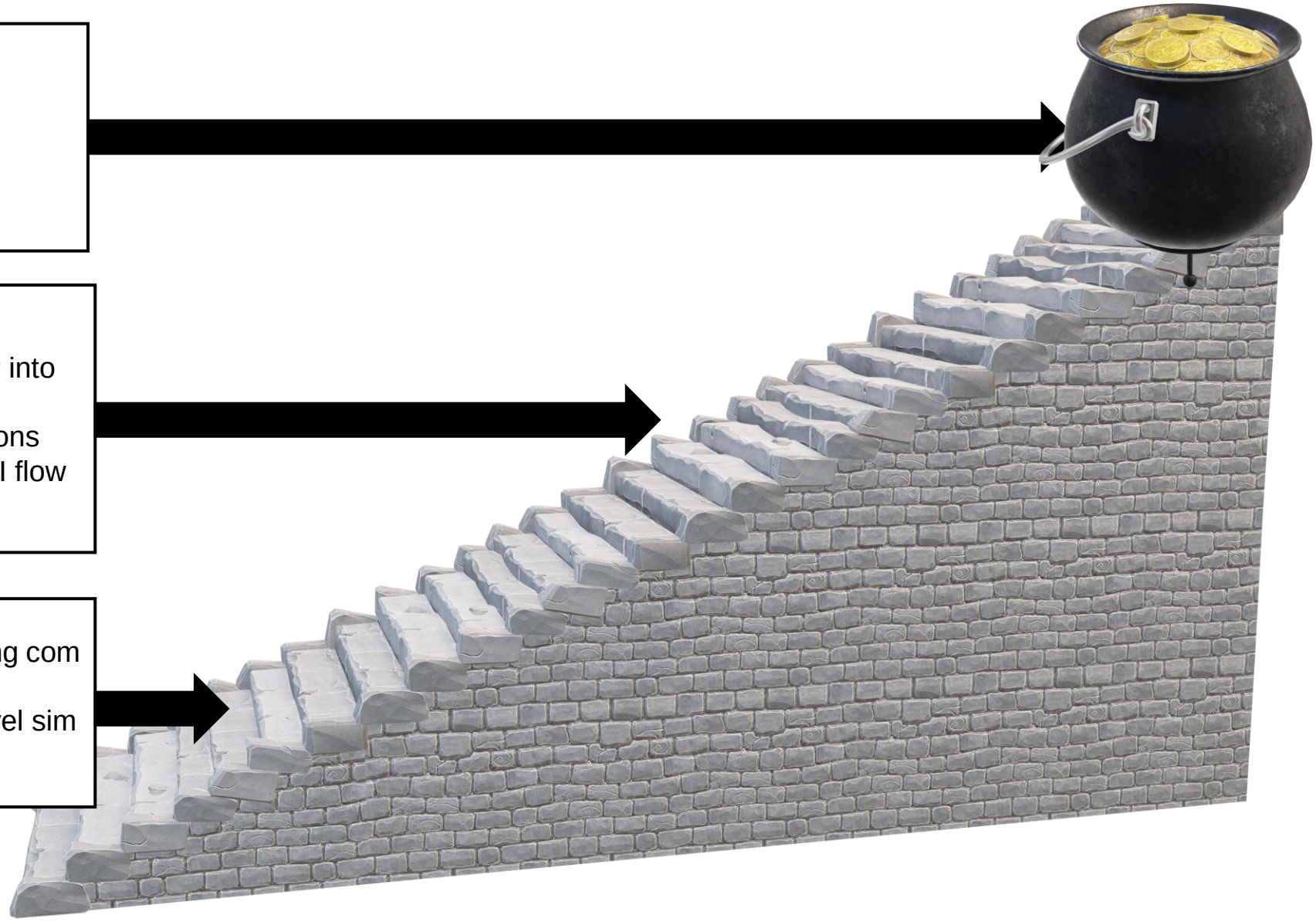
Class shows successful model for expanding entry into field of semiconductor design

Chipyard Learning Curve

Advanced-level
- Configure custom IO/clocking setups
- Develop custom FireSim extensions
- Integrate and tape-out a complete SoC

Evaluation-level
- Integrate or develop custom hardware IP into Chipyard
- Run FireSim FPGA-accelerated simulations
- Push a design through the Hammer VLSI flow
- Build your own system

Exploratory-level
- Configure a custom SoC from pre-existing components
- Generate RTL, and simulate it in RTL level simulation
- Evaluate existing RISC-V designs



Community

Documentation:

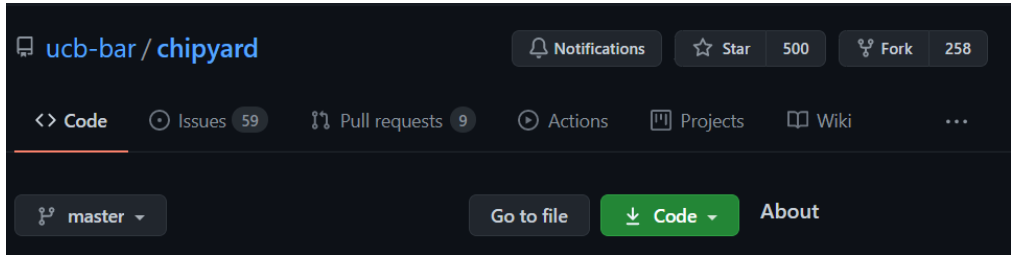
- https://chipyard.readthedocs.io/en/dev/
- 133 pages
- Most of today's tutorial content is covered there

Mailing List:

- google.com/forum/#!forum/chipyard

Open-sourced:

- All code is hosted on GitHub
- Issues, feature-requests, PRs are welcomed



Summary

- Integrated design, simulation and implementation framework
- Converging the various flows
- Feature prioritization based on test chip development experience
- Tighter project integration as lessons from increasingly complex test chips
- Userbase is growing

CHIPYARD