


SoC Architecture and Components

Jerry Zhao

Use Cases

Custom SoC architecture
New blocks + reusing existing blocks

RTL Simulation running test binaries/micro-benchmarks

FPGA-accelerated simulation running full workloads

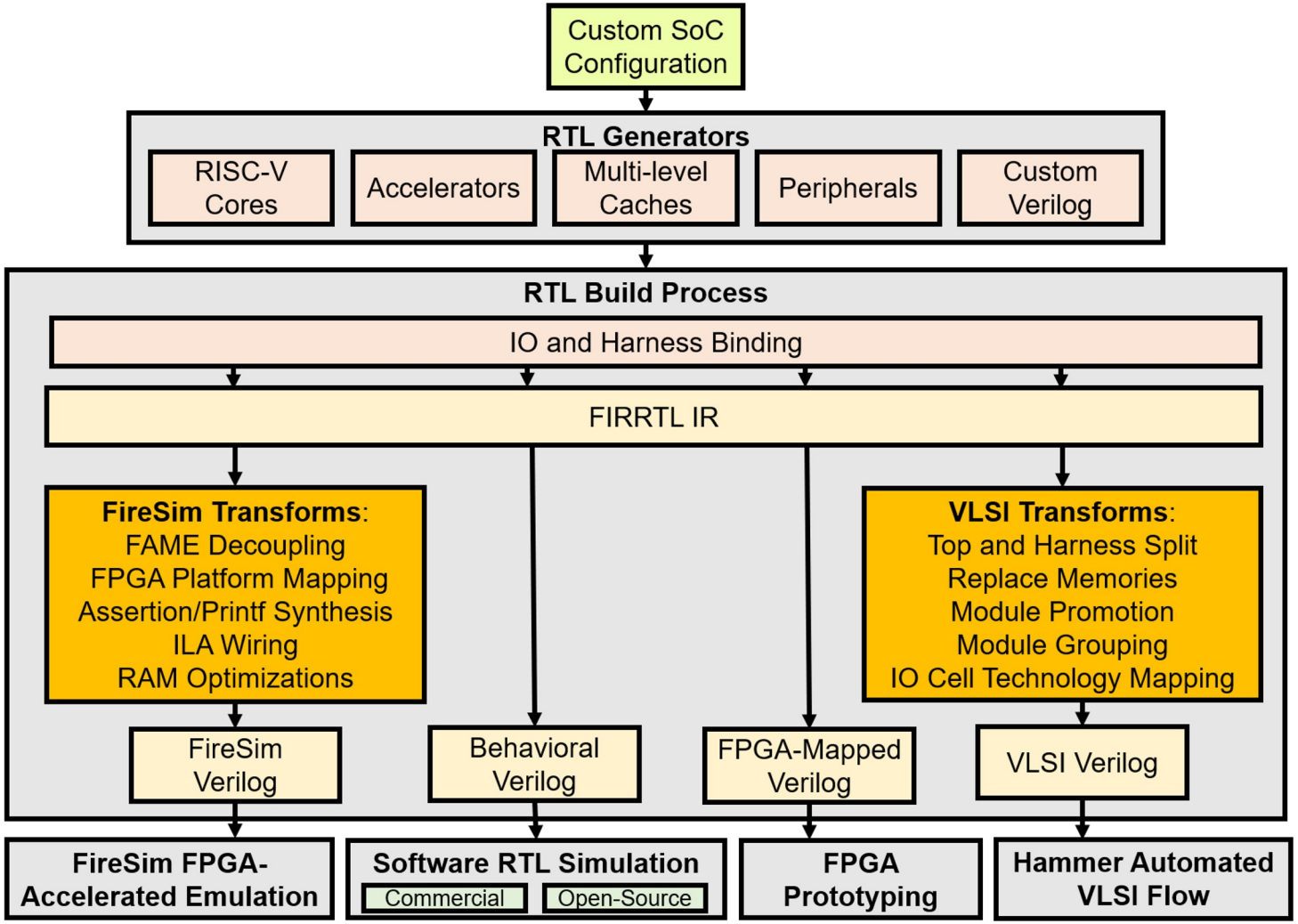
FPGA prototyping for fast cool demos

Tape-out the SoC to get actual silicon results

CHIPYARD Organization

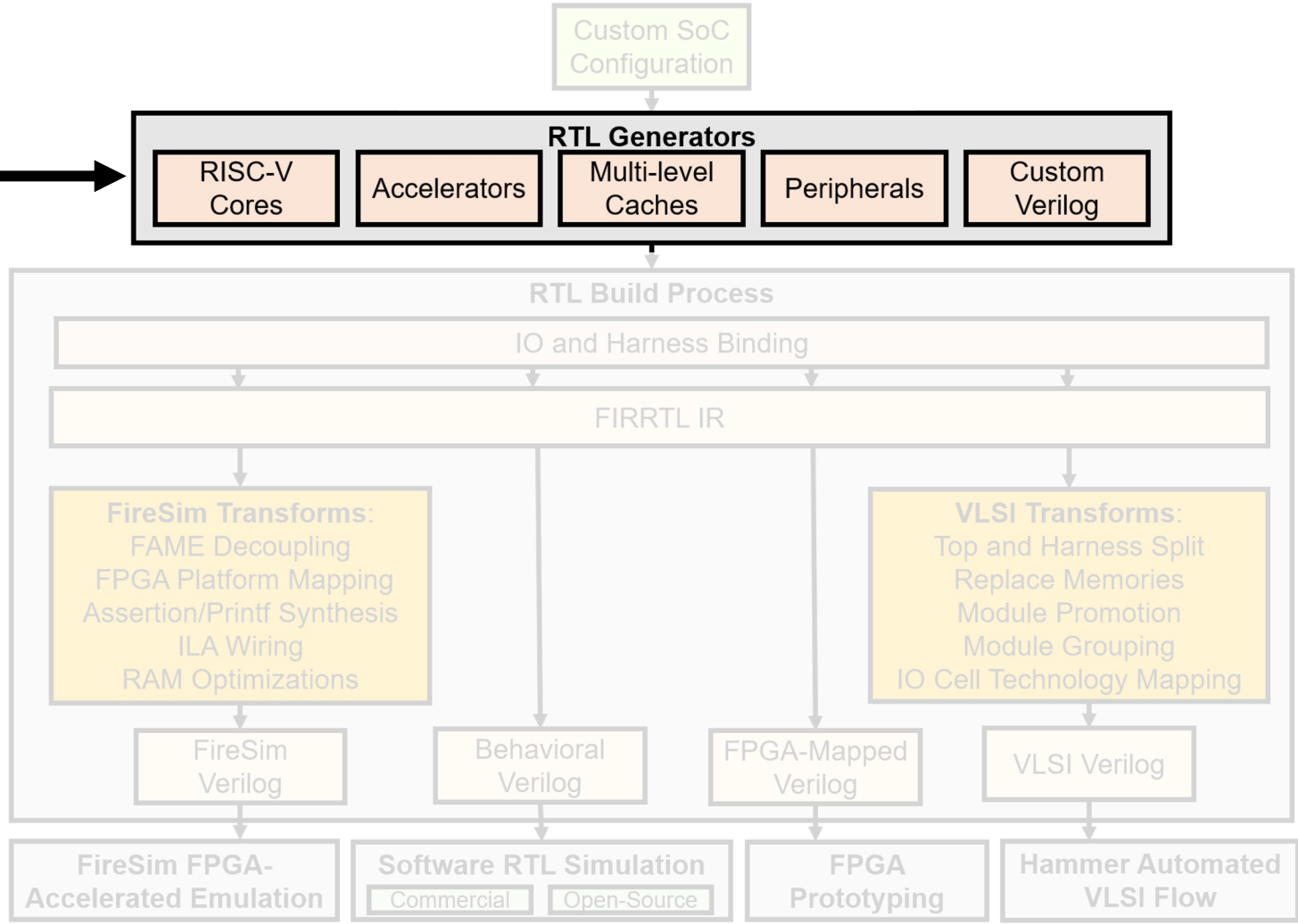
What is Chipyard?

- An organized **framework** for various SoC design tools
- A **curated IP library** of open-source RISC-V SoC components
- A **methodology** for agile SoC architecture design, exploration, and evaluation
- A **tapeout-ready chassis** for custom RISC-V SoCs

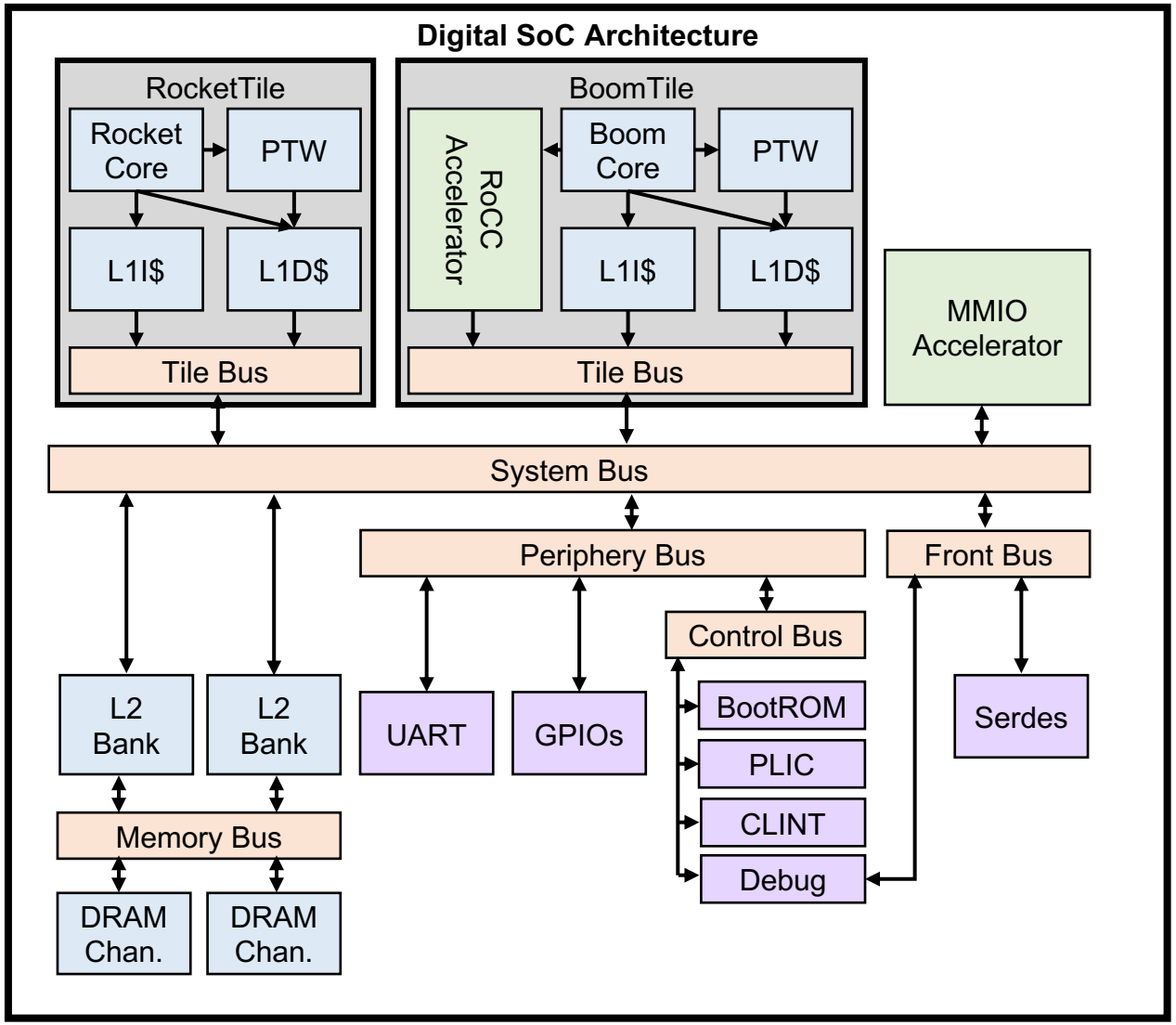


CHIPYARD Organization

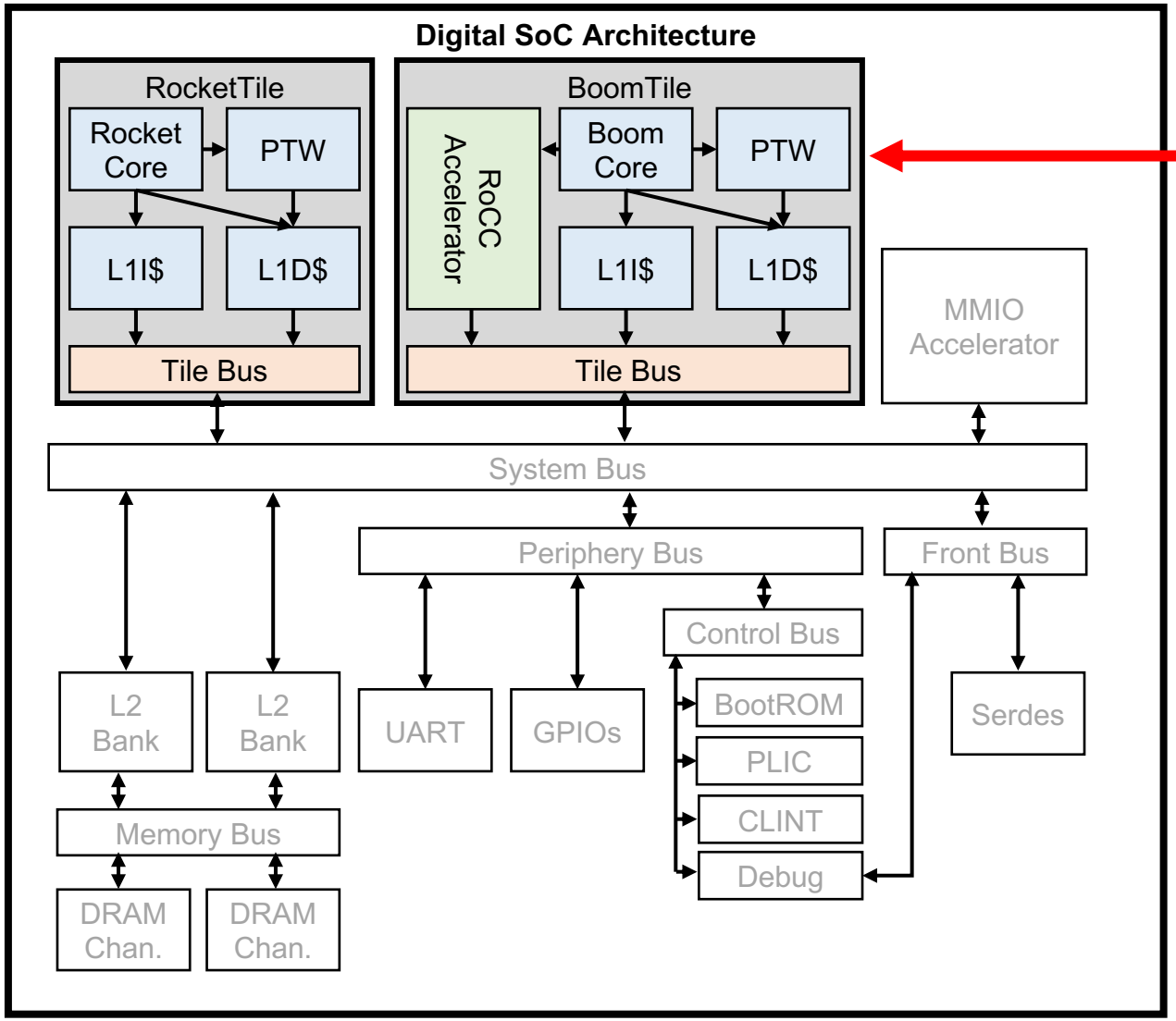
SoC architecture and generators



SoC Architecture



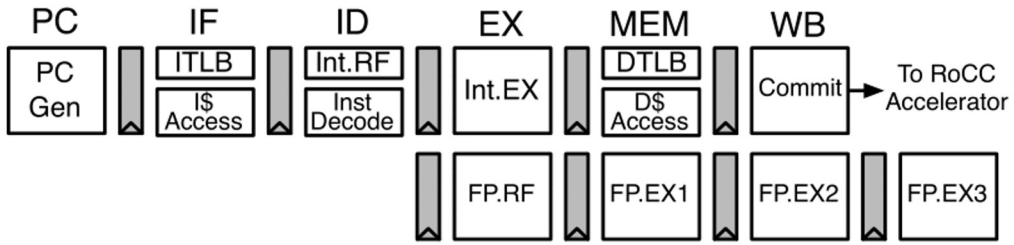
Tiles and Cores



Tiles:

- Each Tile contains a RISC-V core and private caches
- Several varieties of Cores supported
- Interface supports integrating your own RISC-V core implementation

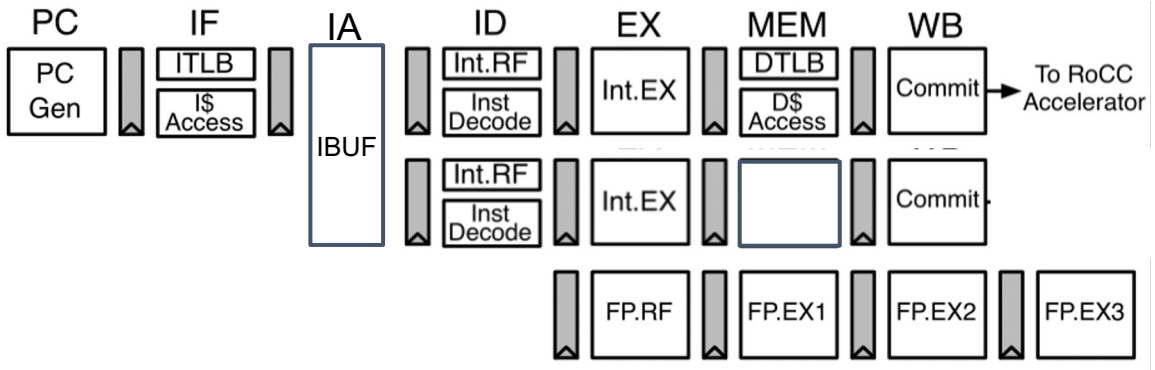
Chisel RISC-V Cores

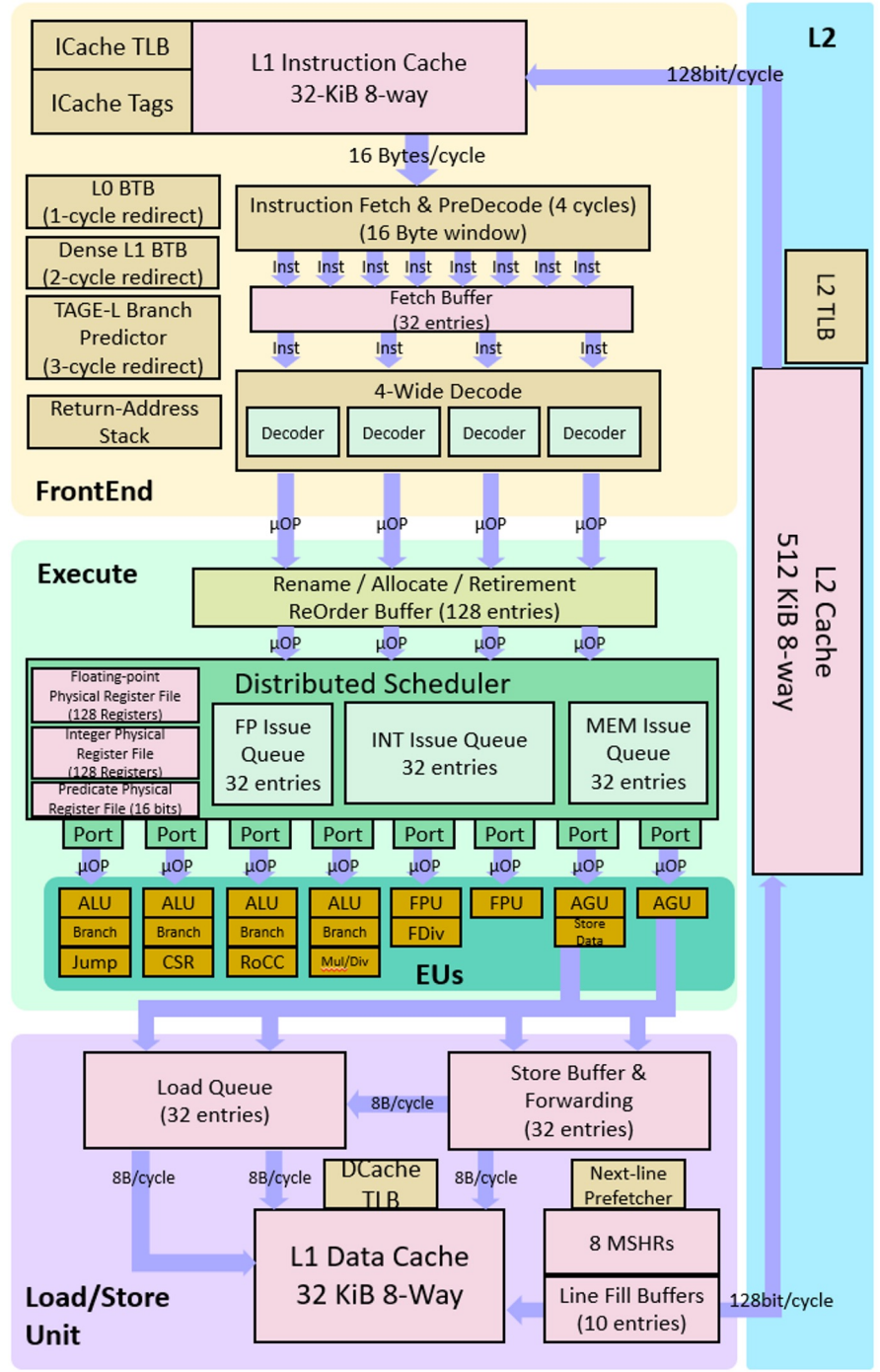


Rocket: 5-stage single-issue in-order

Shuttle: 6-stage superscalar in-order

SonicBOOM: 12-stage superscalar out-of-order

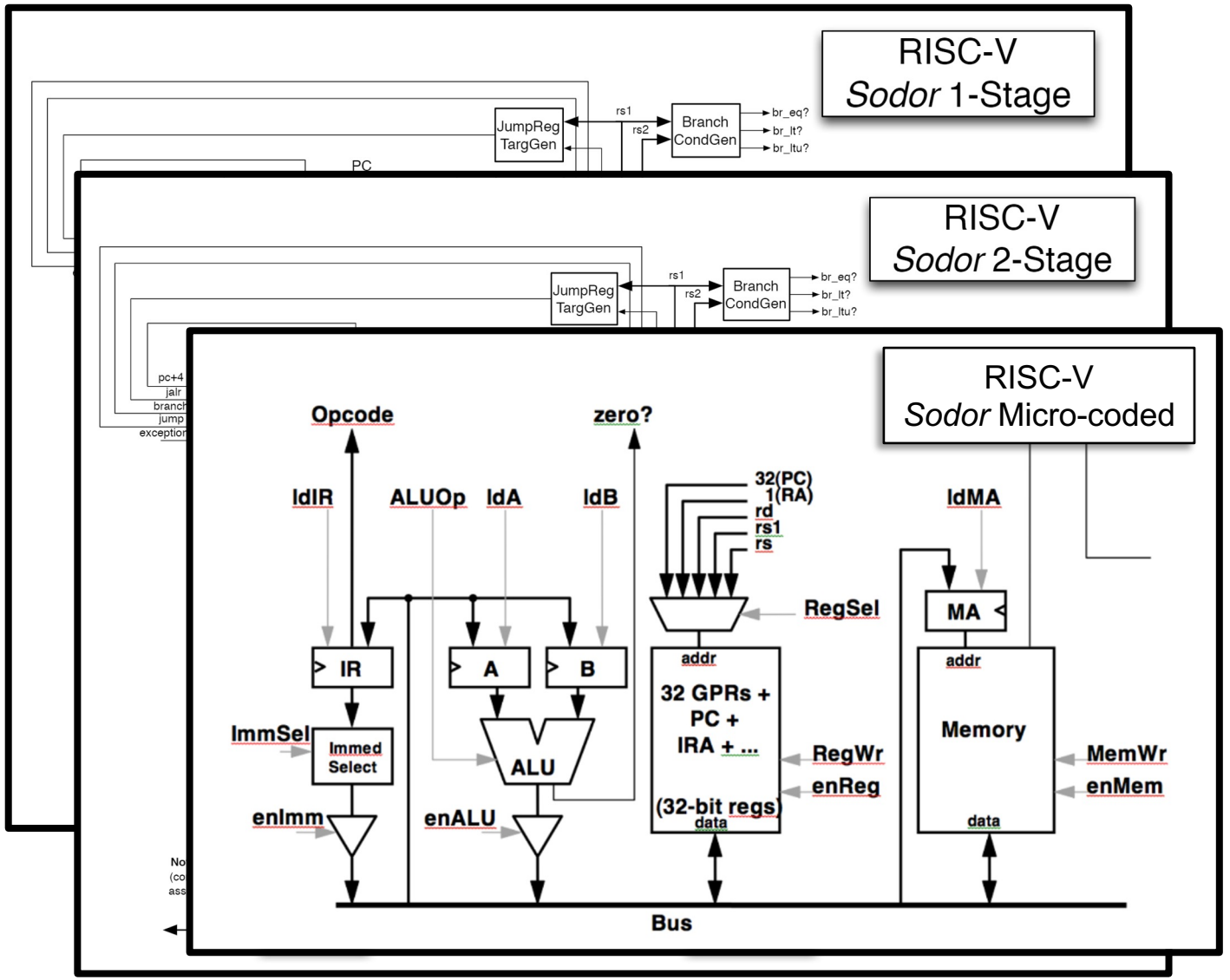




Sodor Education Cores

Sodor Core Collection

- Collection of RV32IM cores for teaching and education
- 1-stage, 2-stage, 3-stage, 5-stage implementations
- Micro-coded “bus-based” implementation
- Used in introductory computer architecture courses at Berkeley



Non-Chisel Cores

CVA6:

- RV64GC 6-stage single-issue in-order core
- SystemVerilog

Ibex:

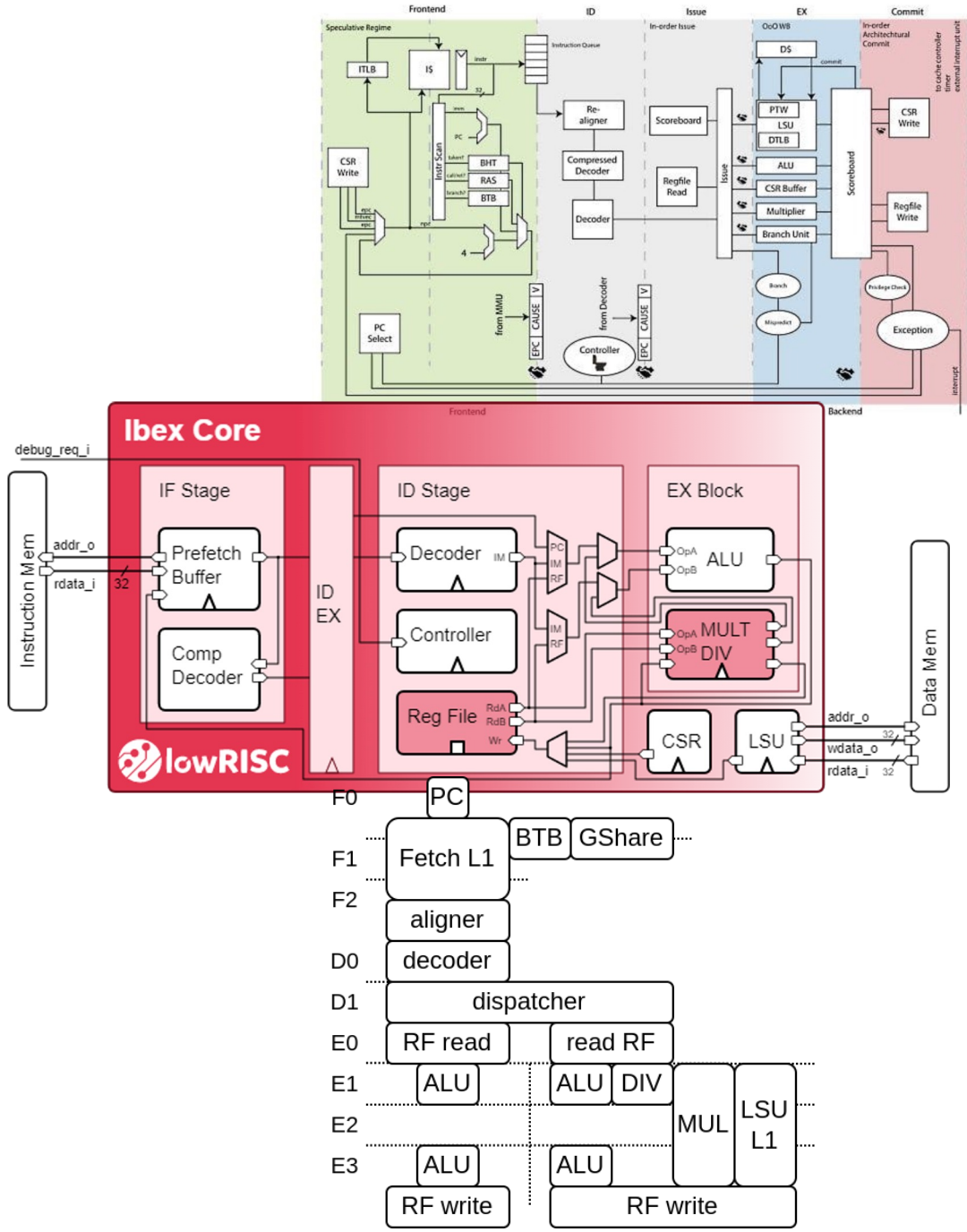
- RV64IMC 2-stage single-issue in-order core
- SystemVerilog

VexiiRiscv:

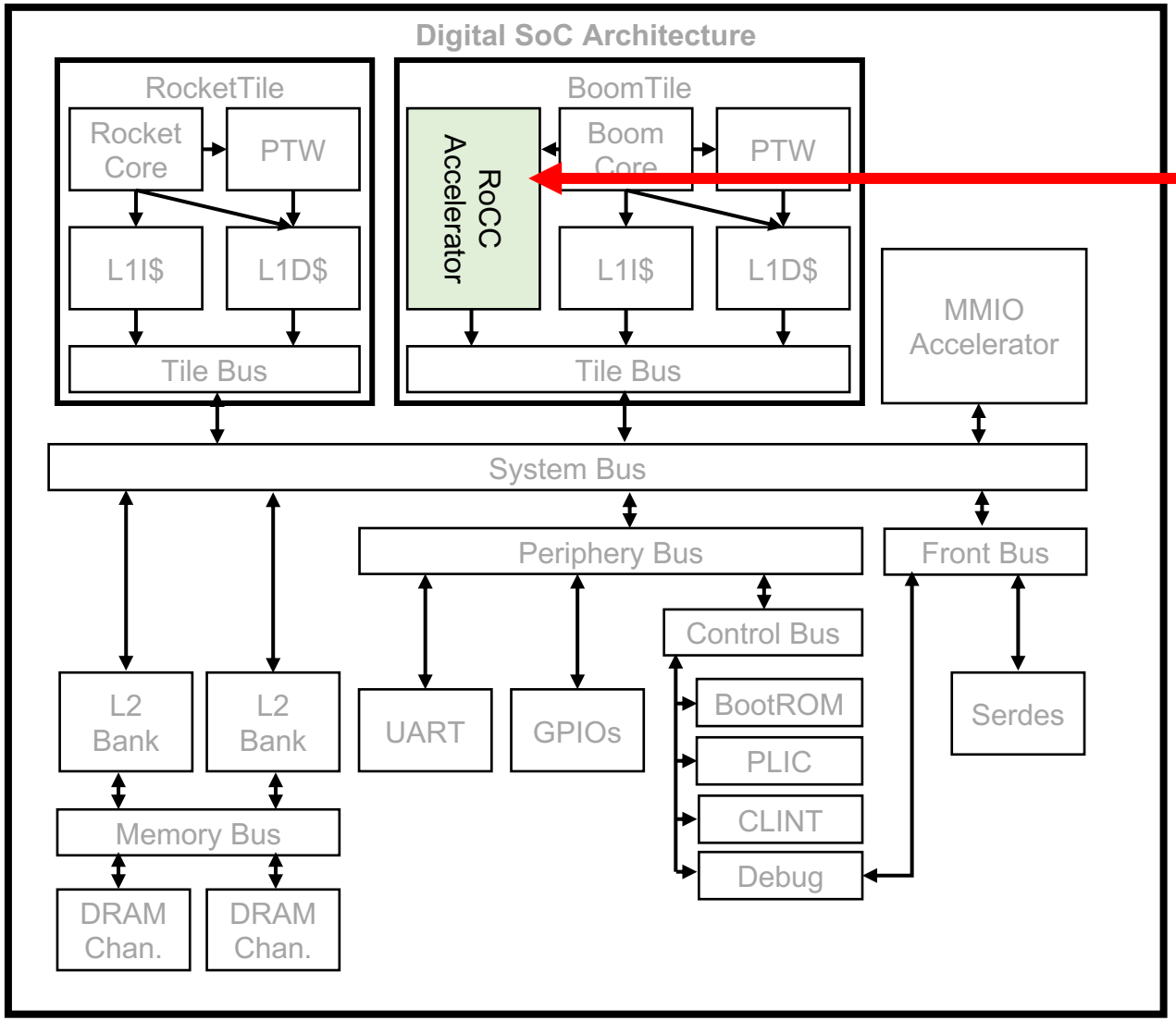
- RV64IMAFDBC 6-stage dual-issue in-order core
- SpinalHDL

SpikeTile:

- RV64IMAFDCBV core ISS model
- C++ with DPI interface to Verilog/Chisel

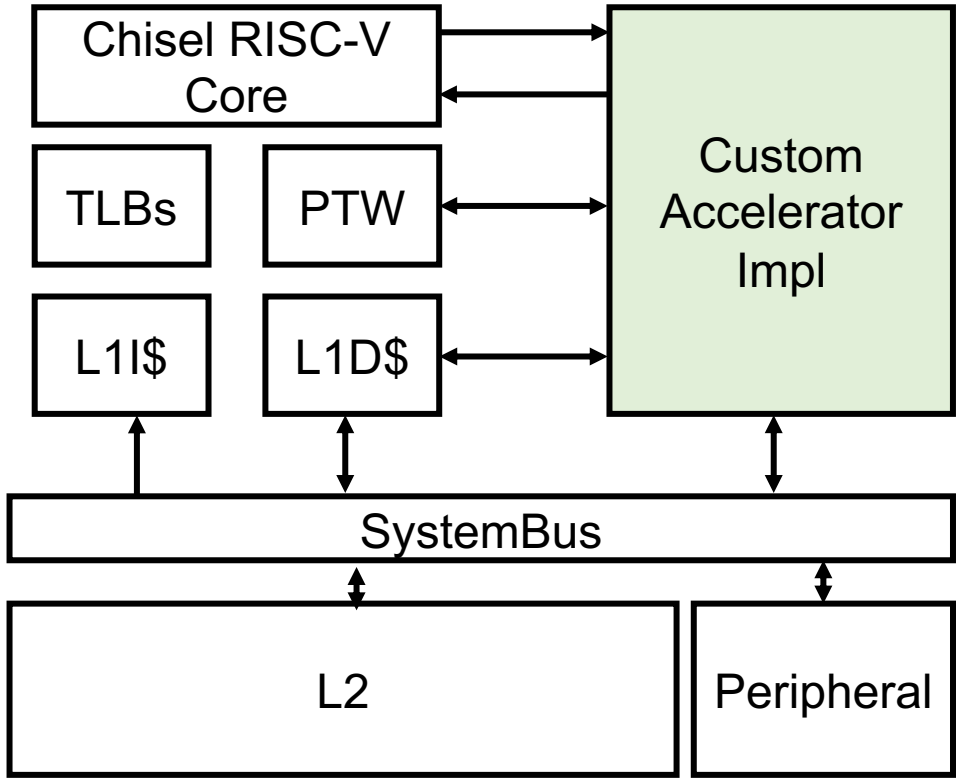


RoCC Accelerators

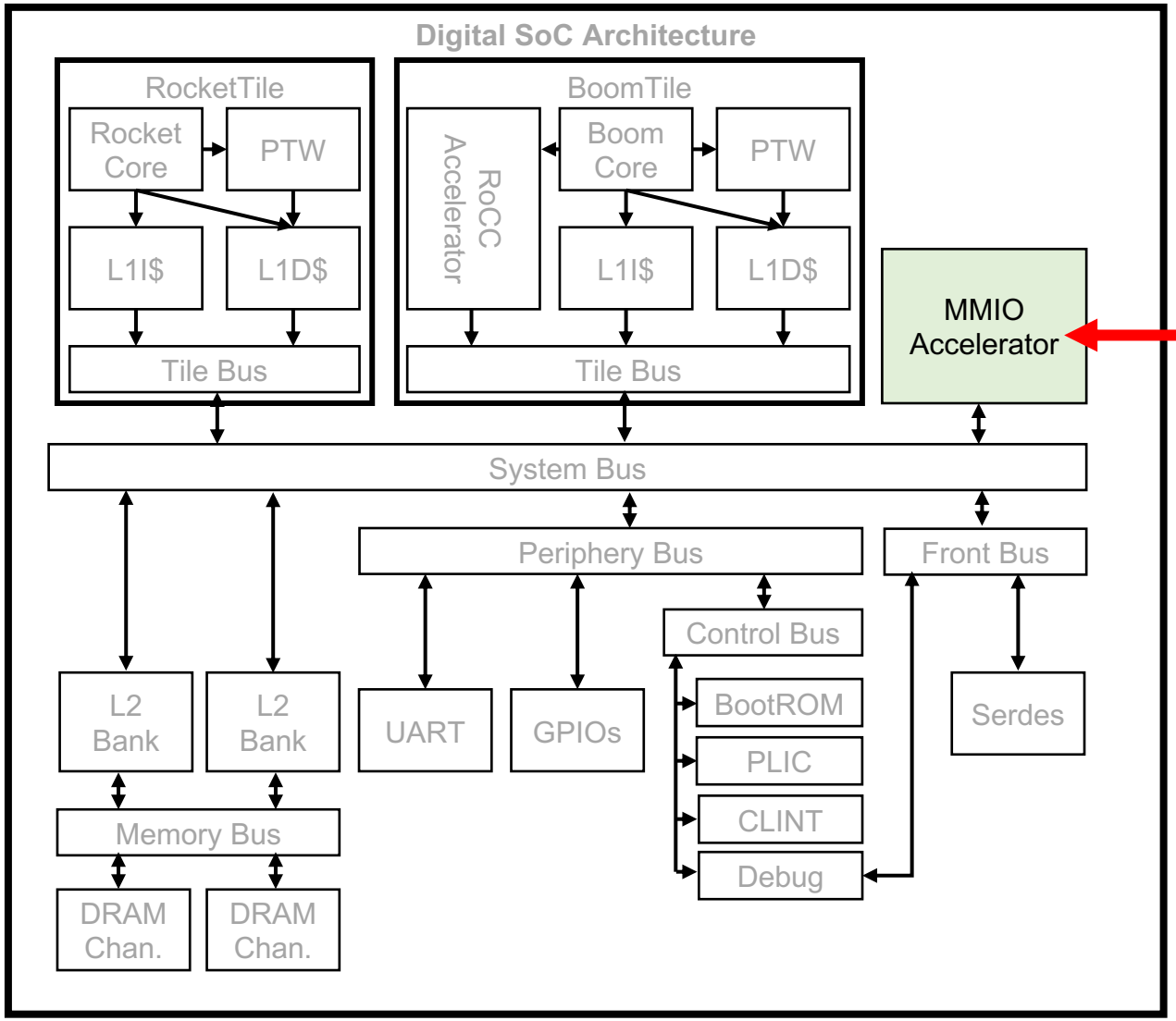


RoCC Accelerators:

- Tightly-coupled accelerator interface
- Attach custom accelerators to Rocket or BOOM cores
- Included example implementations



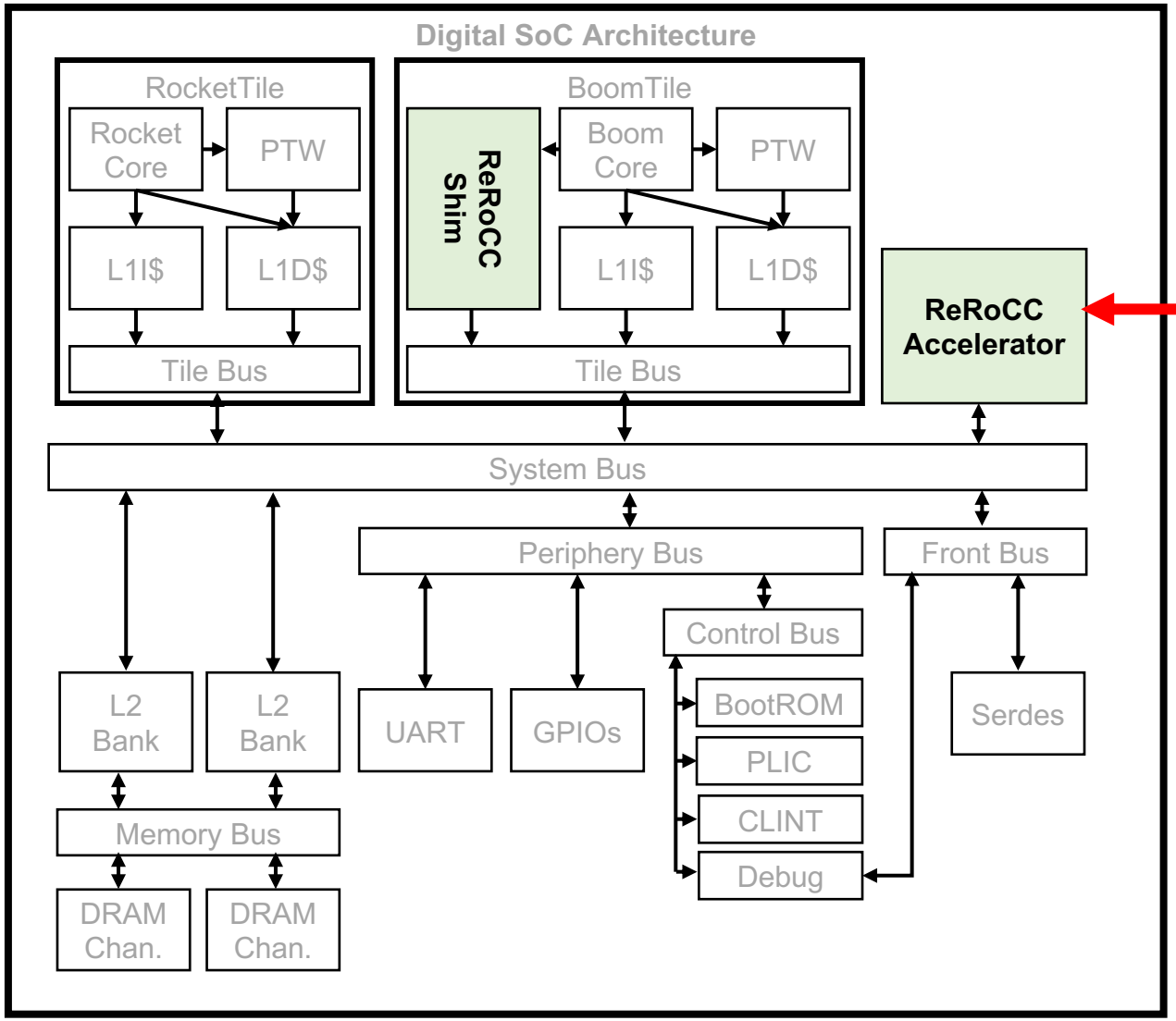
MMIO Accelerators



MMIO Accelerators:

- Controlled by MMIO-mapped registers
- Supports DMA to memory system
- Examples:
  - Nvidia NVDLA accelerator
  - FFT accelerator generator

New: Remote RoCC Accelerators



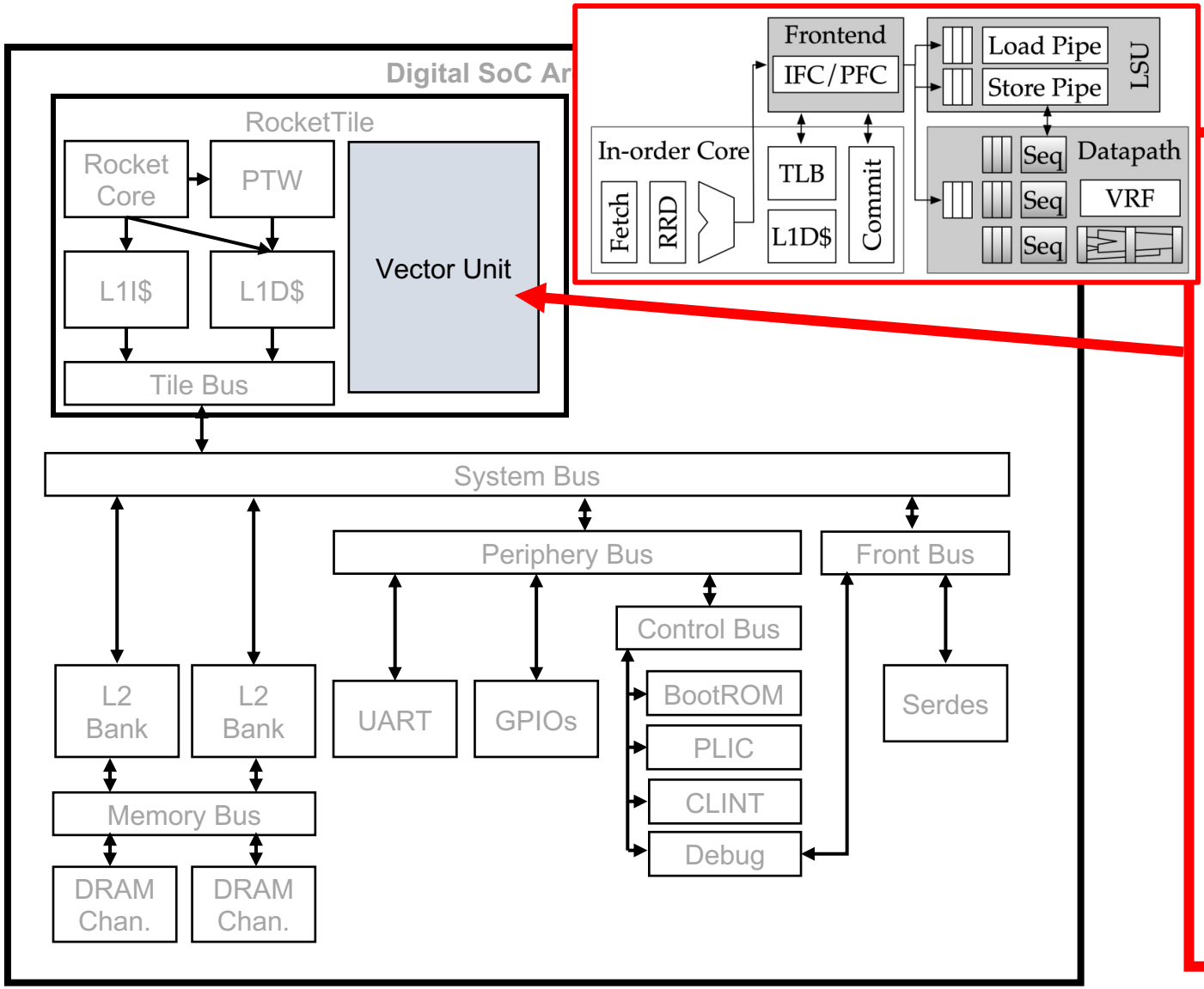
Remote RoCC Accelerators:

- Disaggregate standard RoCC accelerators over a system-wide interconnect
- Enables software-managed accelerator sharing of RoCC accels.
- Reuse existing RoCC accelerators + software

AuRORA

- Demonstrates ReRoCC for multi-tenant ML workloads on a multi-accelerator system
- IEEE MICRO Top Picks
- Hands-on tutorial later

NEW: RISC-V Vector Units



Standard RVV vector units:

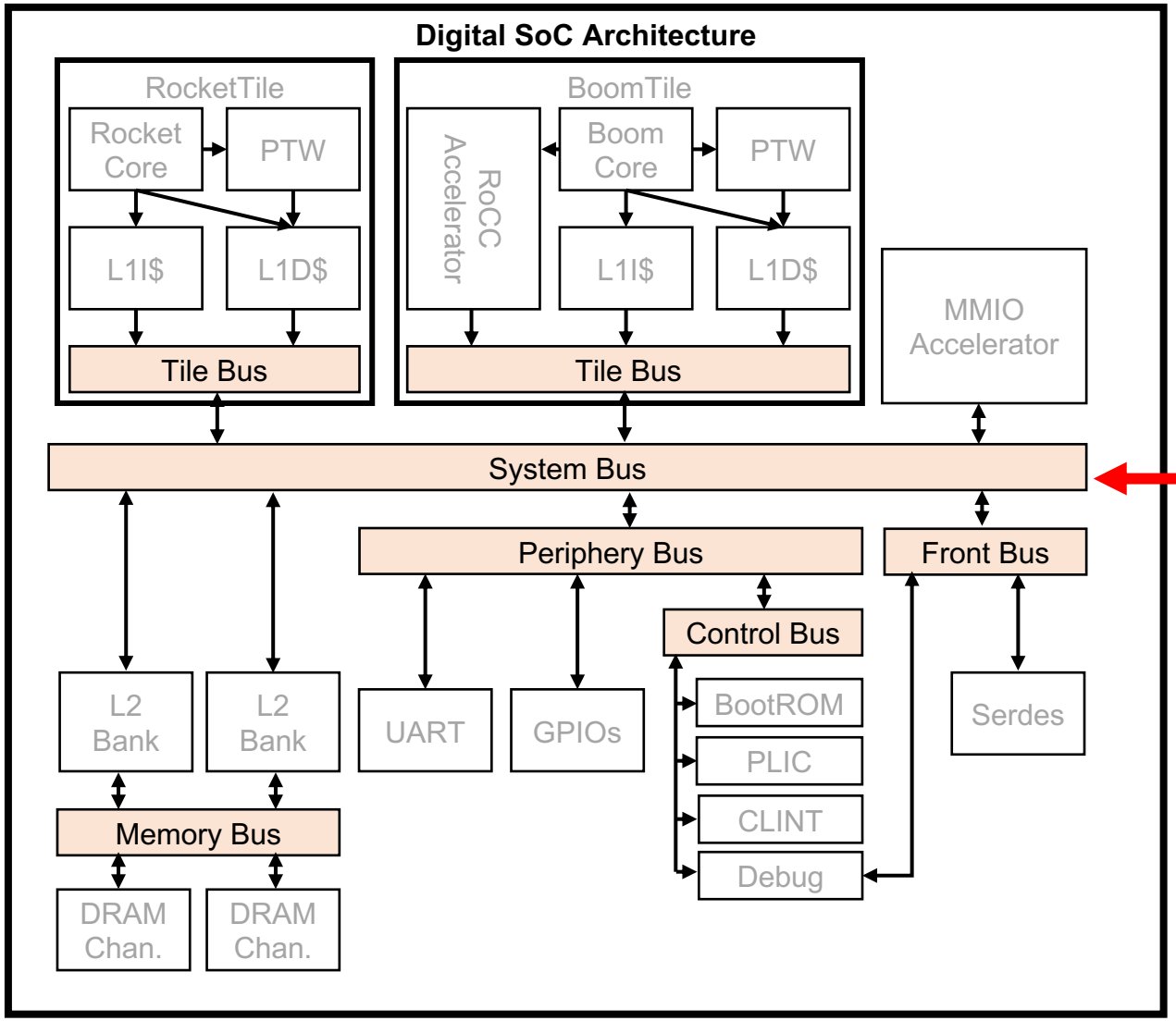
Saturn – Newly developed RVV 1.0

- Integrates with Rocket + Shuttle cores
- Supports all RVV 1.0 instructions
- Supports Zvfh half-prec
- Supports Zvbb vector bit-manip
- Supports virtual memory + precise traps
- github.com/ucb-bar/saturn-vectors

Ara – SystemVerilog vector unit

- SystemVerilog vector unit
- Integrates with Rocket + Shuttle cores

Coherent Interconnect



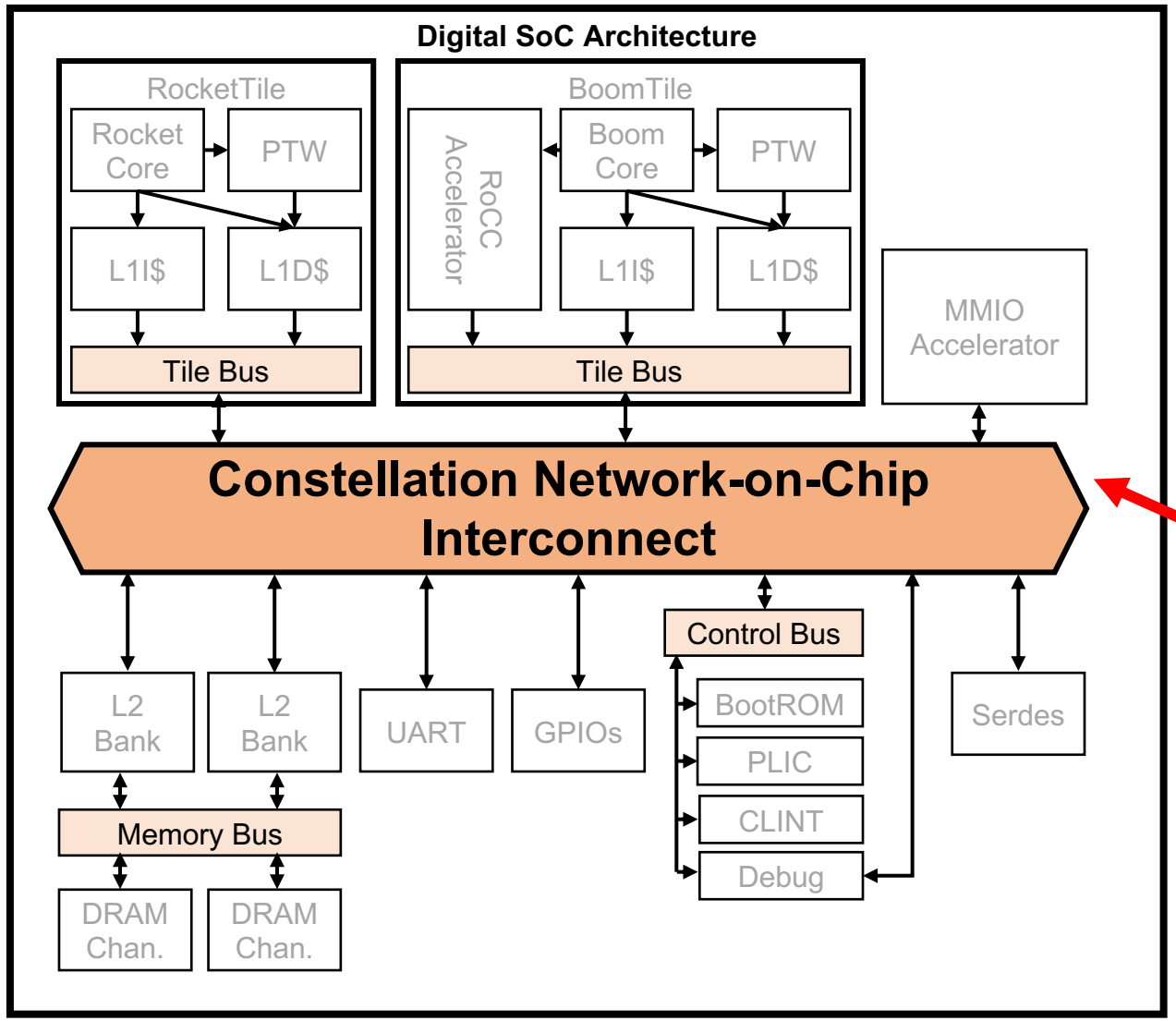
TileLink Standard:

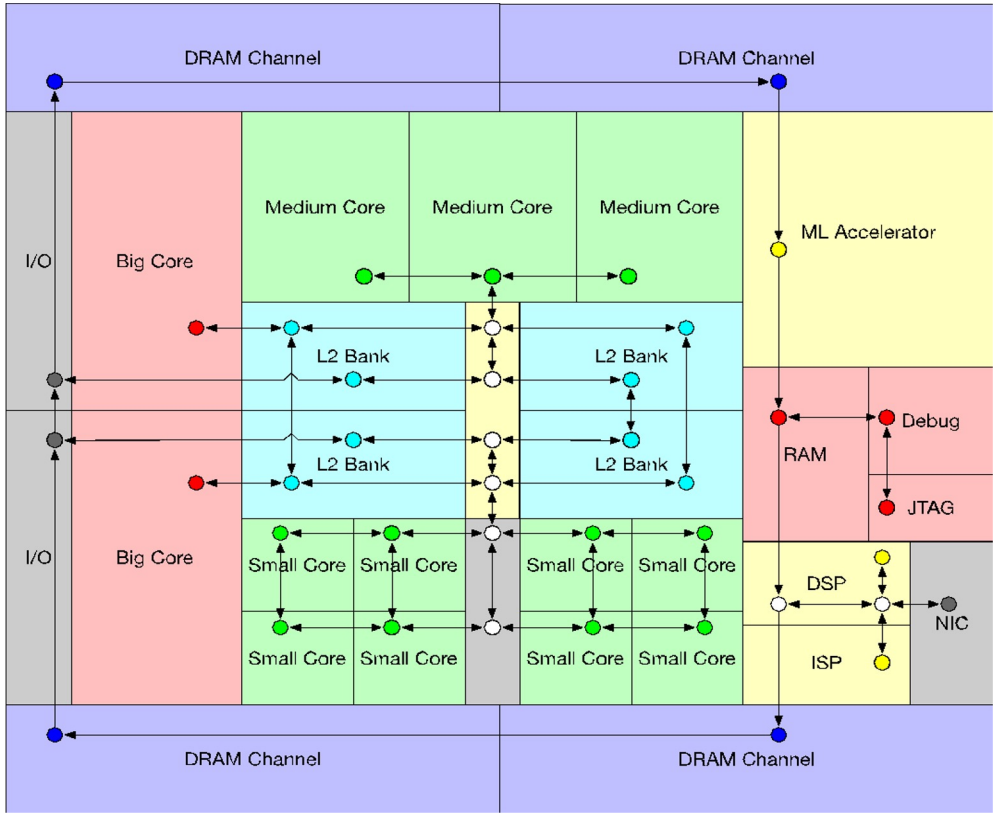
- TileLink is open-source chip-scale interconnect standard
- Comparable to AXI/ACE
- Supports multi-core, accelerators, peripherals, DMA, etc

Interconnect IP:

- Library of TileLink RTL generators provided in RocketChip
- RTL generators for crossbar-based buses
- Width-adapters, clock-crossings, etc.
- Adapters to AXI4, APB

NoC Interconnect

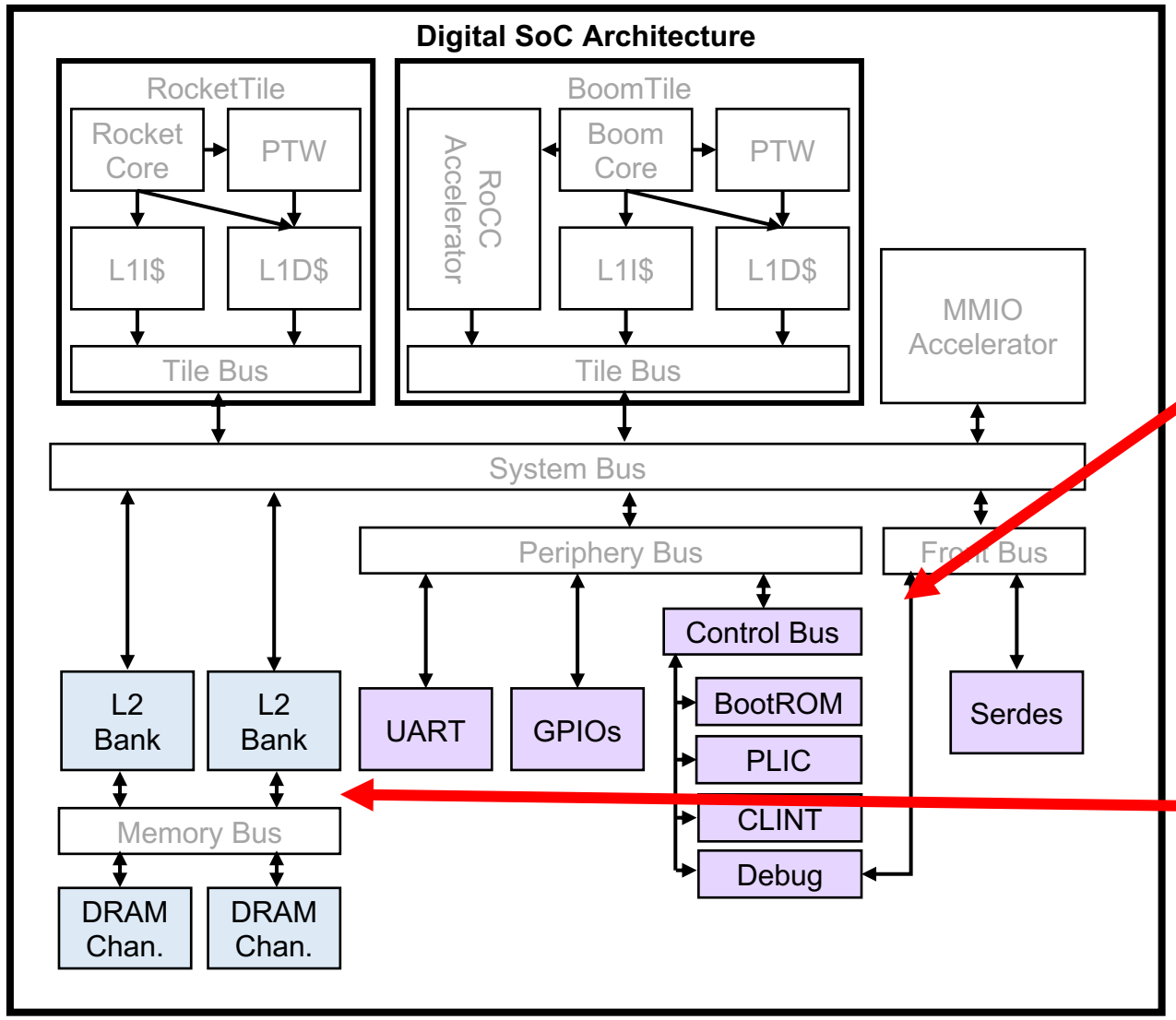




Constellation

- Drop-in replacement for TileLink crossbar buses
- Build multi-NoC, irregular topology interconnects
- Deadlock-free wormhole routing with virtual channels
- Tapeout-proven

L2/DRAM/Peripherals/IO



Peripherals and IO:

- Interrupt controllers
- JTAG
- UART, GPIOs, SPI, I2C, PWM, etc.
- Clock-management devices
- SerDes
- Scratchpads

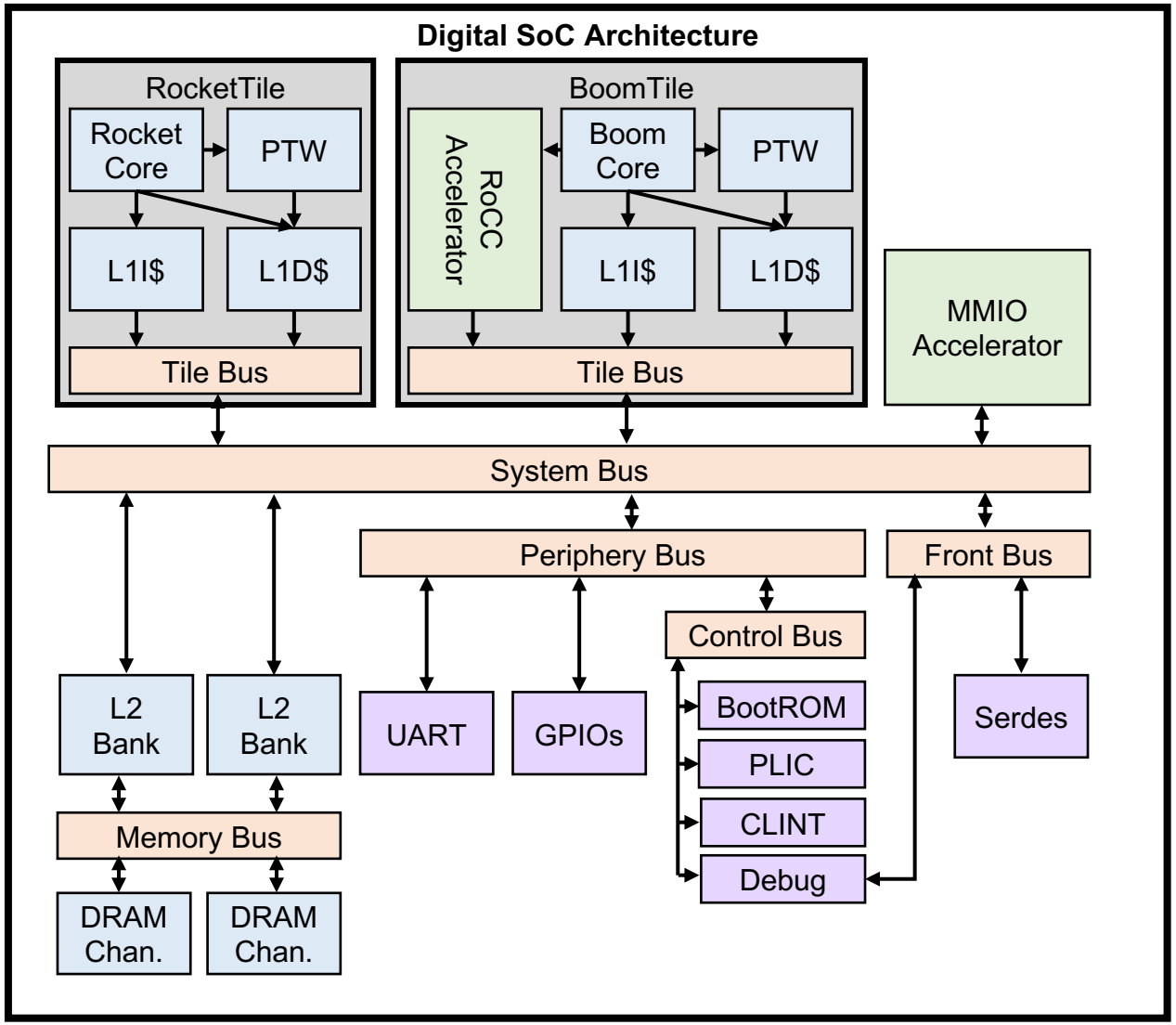
Shared memory:

- Open-source directory-based LLC
- Optional broadcast-based coherence
- Optional incoherent memory systems

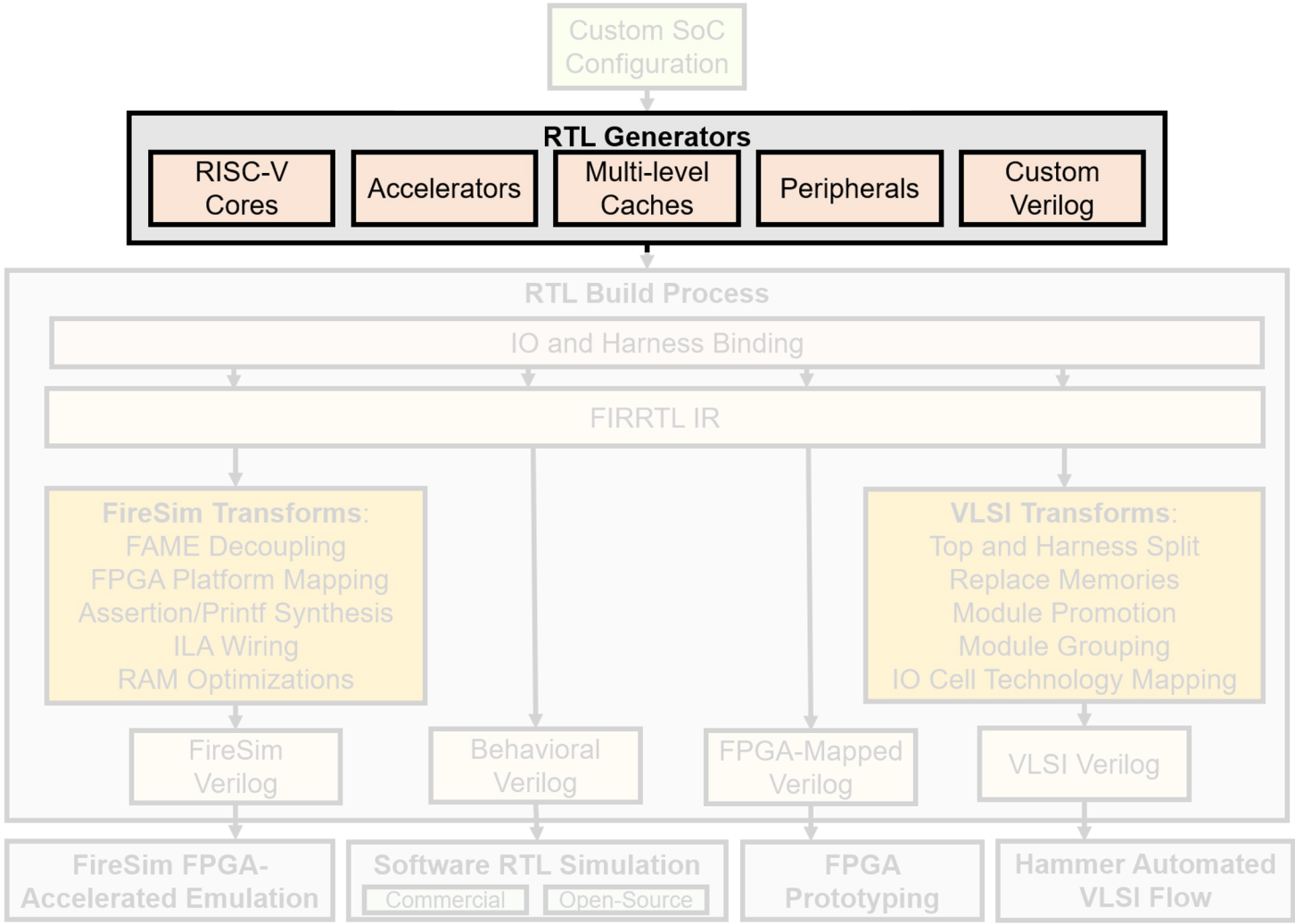
DRAM:

- DRAMSim-backed DRAM model for RTL sim
- FPGA DRAM models in Firesim

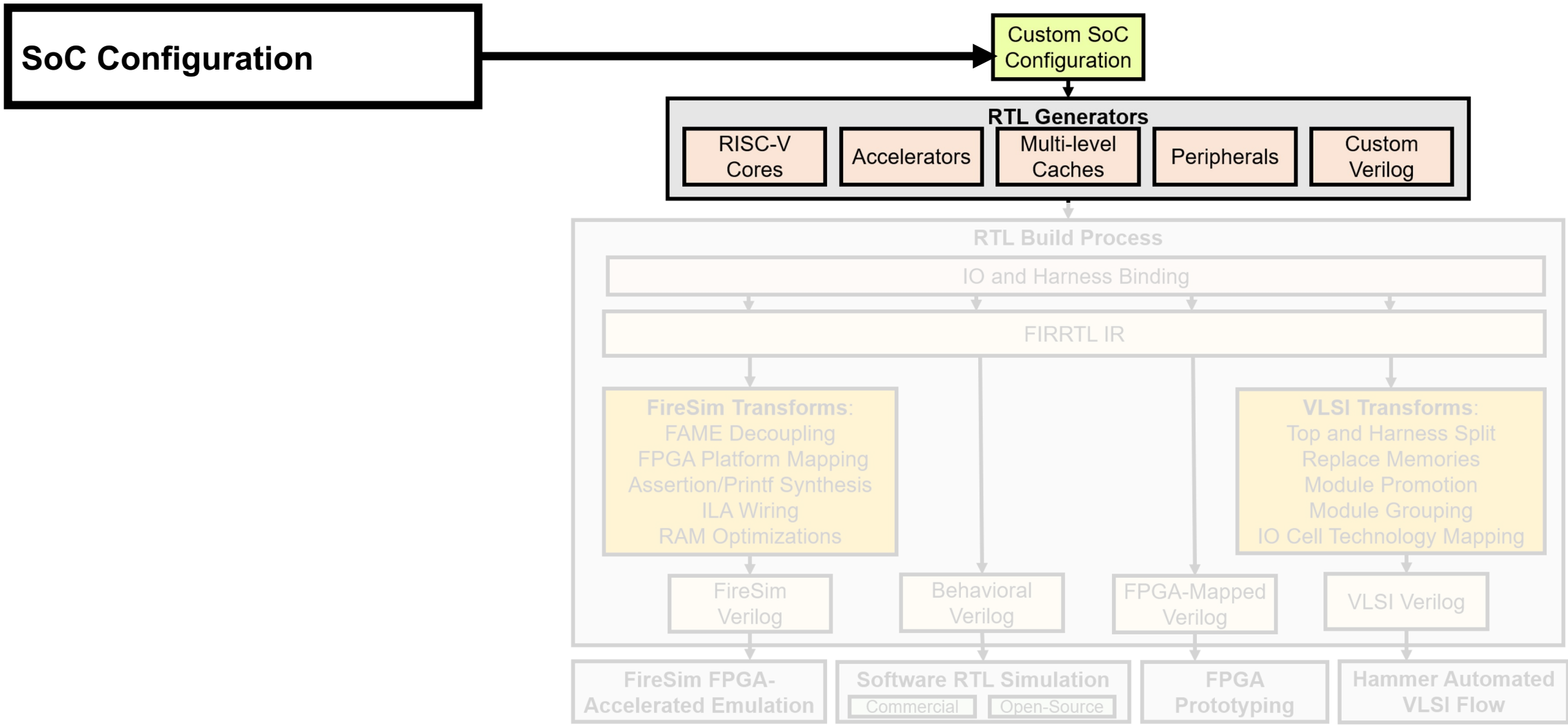
SoC Architecture



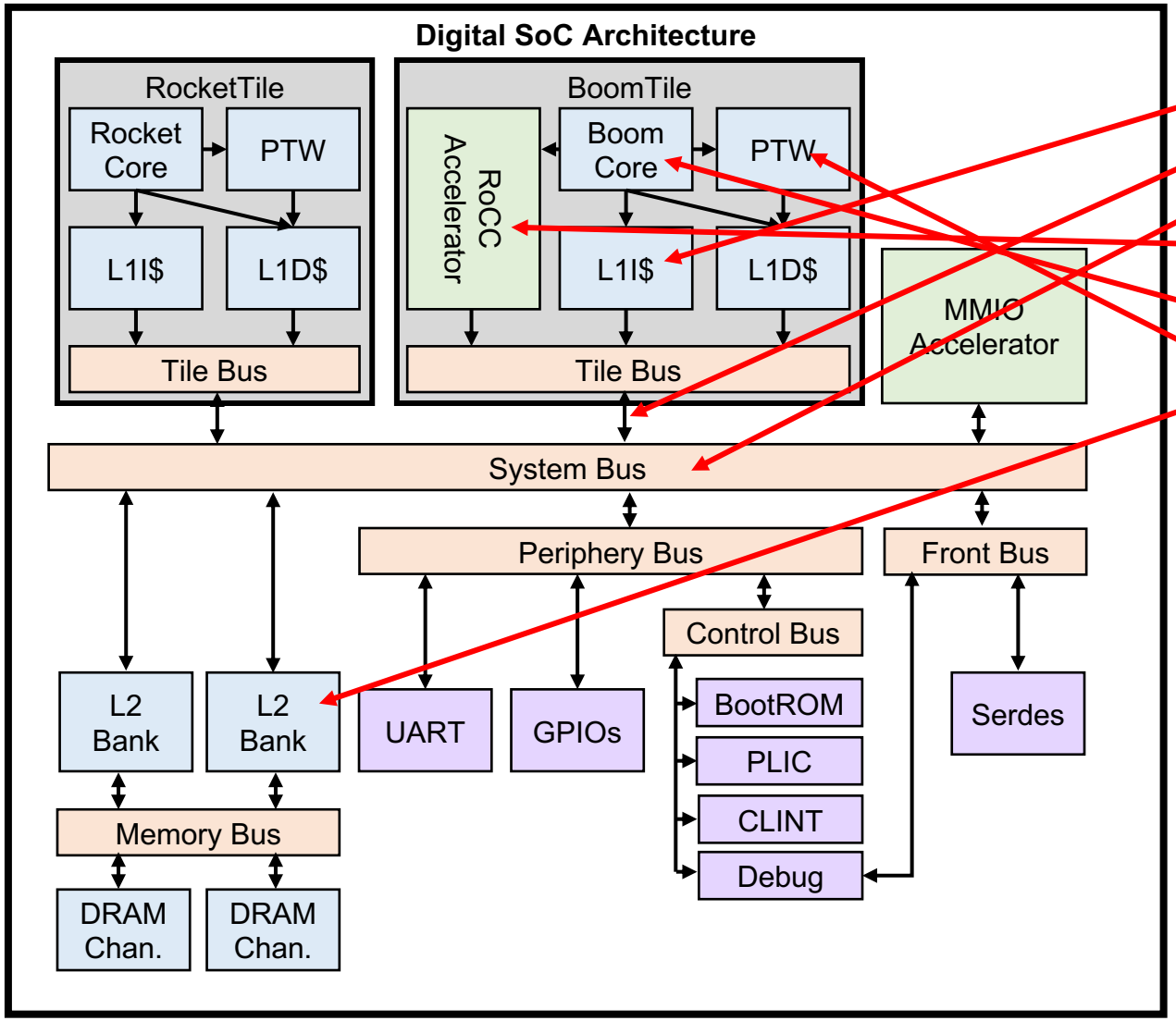

CHIPYARD Organization



CHIPYARD Organization

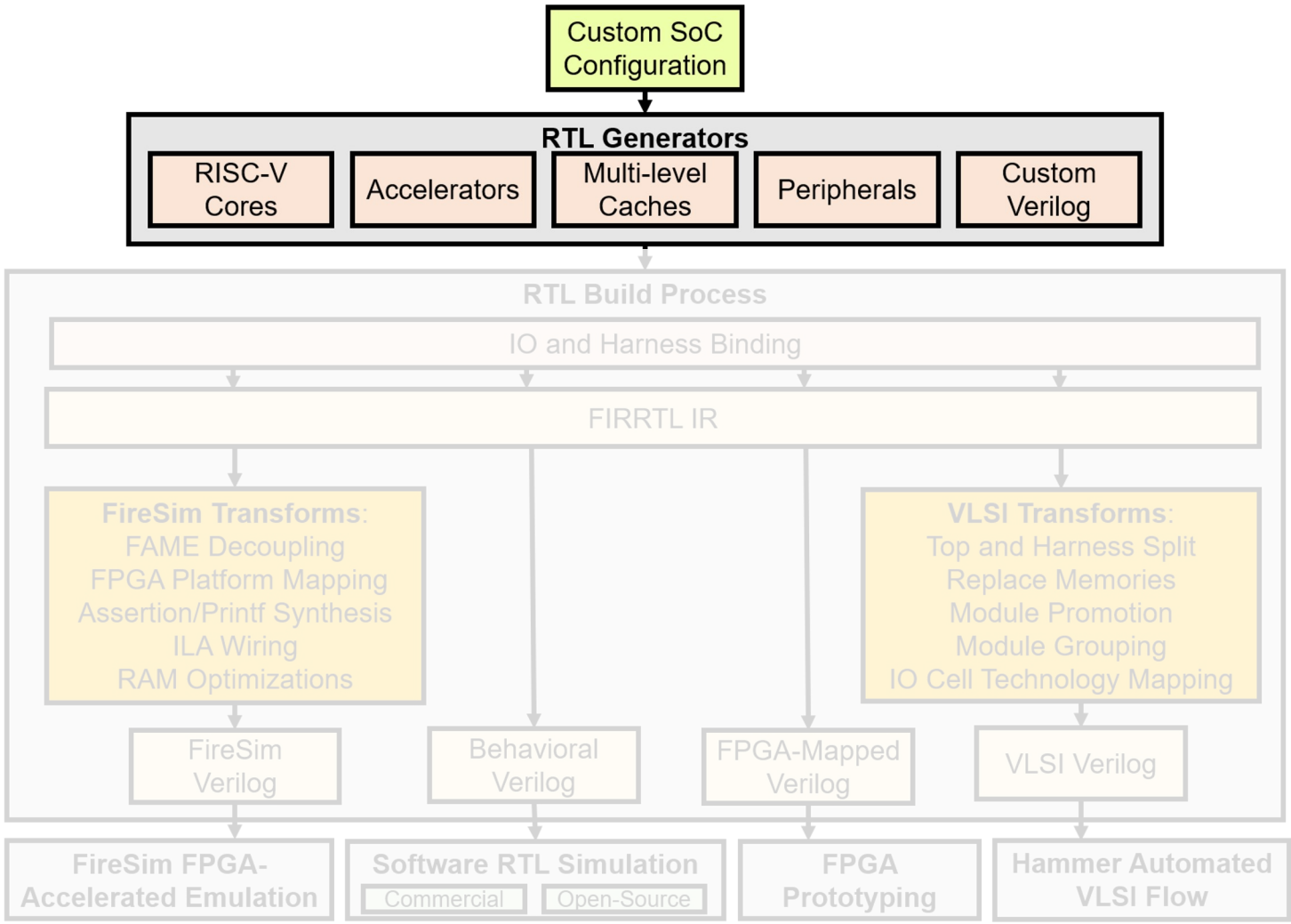


Composable Configurations




```
class CustomConfig extends Config(
  new WithL1CacheWays(4) ++
  new WithAsyncTiles ++
  new WithSystemBusWidth(128) +
  new WithFPGemmini ++
  new With3WideBooms ++
  new WithL2TLBs(512) ++
  new WithL2Sets(1024) ++

  new WithDefaultGemmini ++
  new WithNRocketCores(1) ++
  new WithNBoomCores(1) ++
  new WithBootROM ++
  new WithUART ++
  new WithJtagDTM ++
  new WithGPIOs ++
  new WithInclusiveCache(512) ++
)
```


CHIPYARD Organization



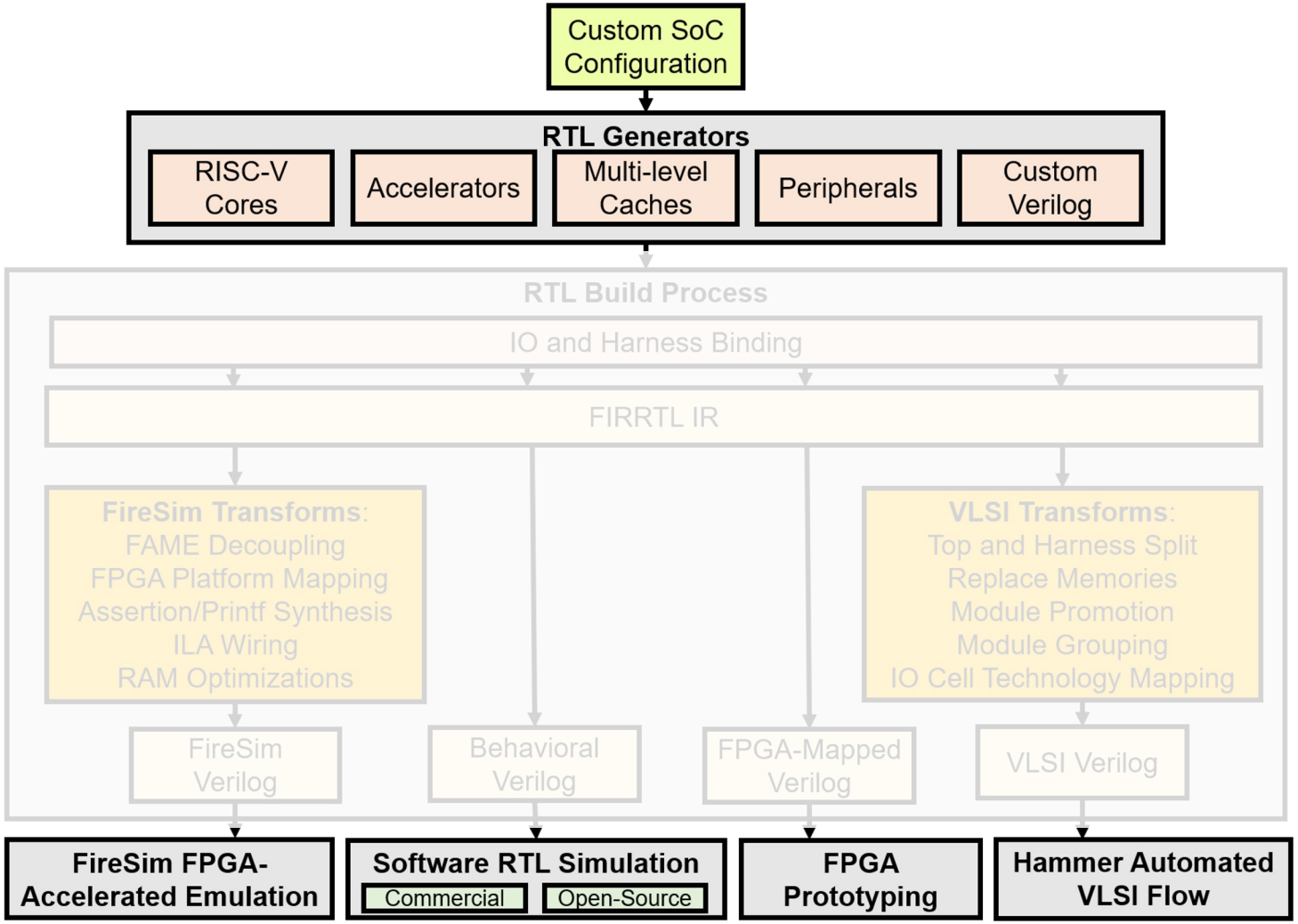
CHIPYARD Organization

SW RTL Simulation:
- RTL-level simulation with Verilator or VCS
- Hands-on tutorial next

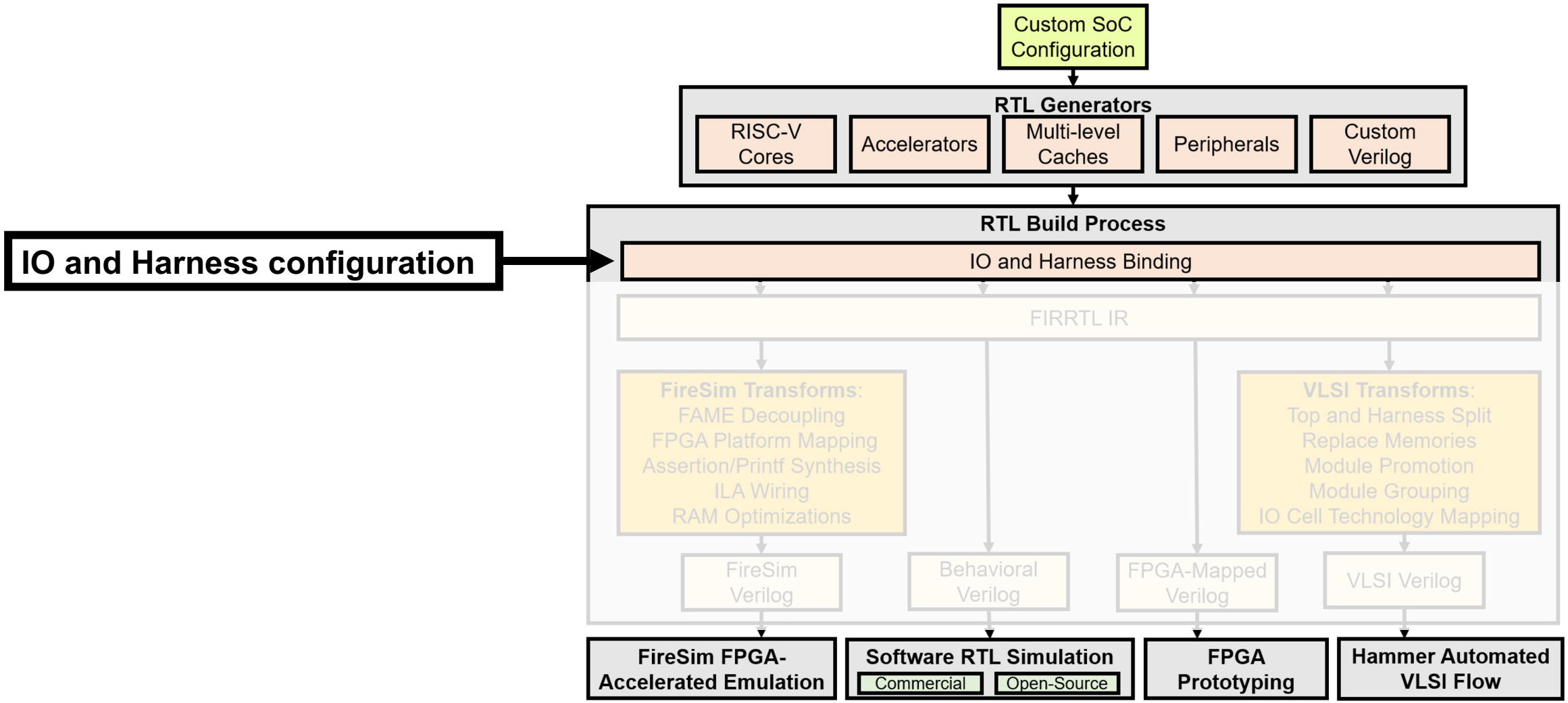
FPGA prototyping:
- Fast, non-deterministic prototypes
- Bringup platform for taped-out chips

Hammer VLSI flow:
- Tapeout a custom config in some process technology
- Overview of flow later

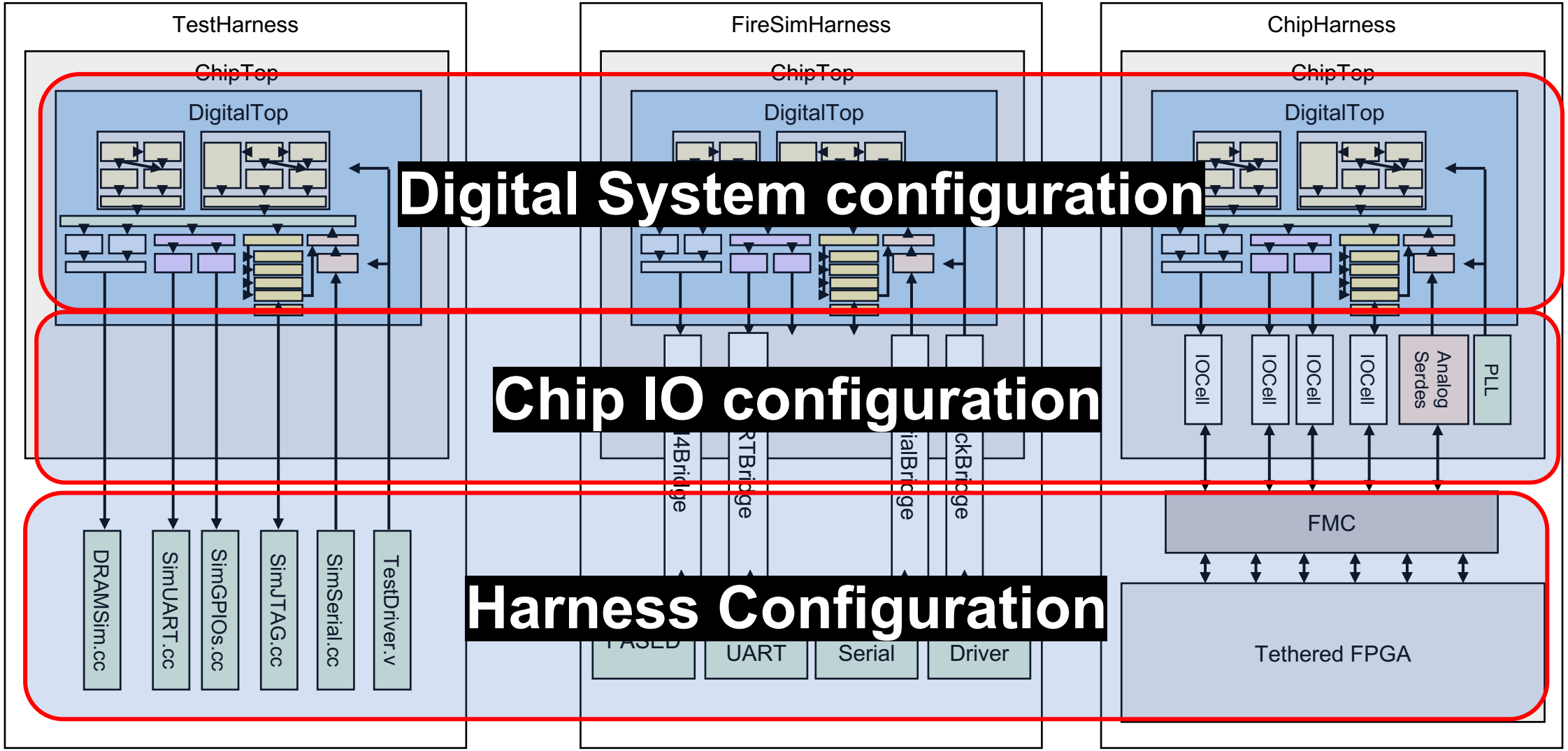
FireSim:
- Fast, accurate FPGA-accelerated simulations
- Hands-on tutorial later



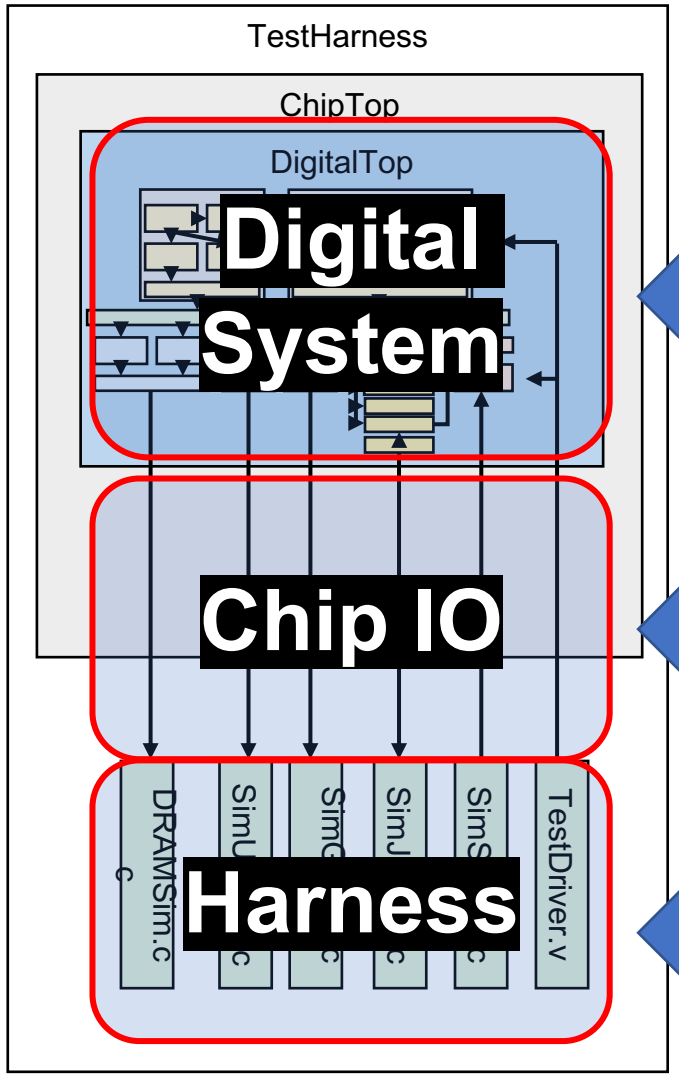

CHIPYARD Organization



Multipurpose



Configuring IO + Harness



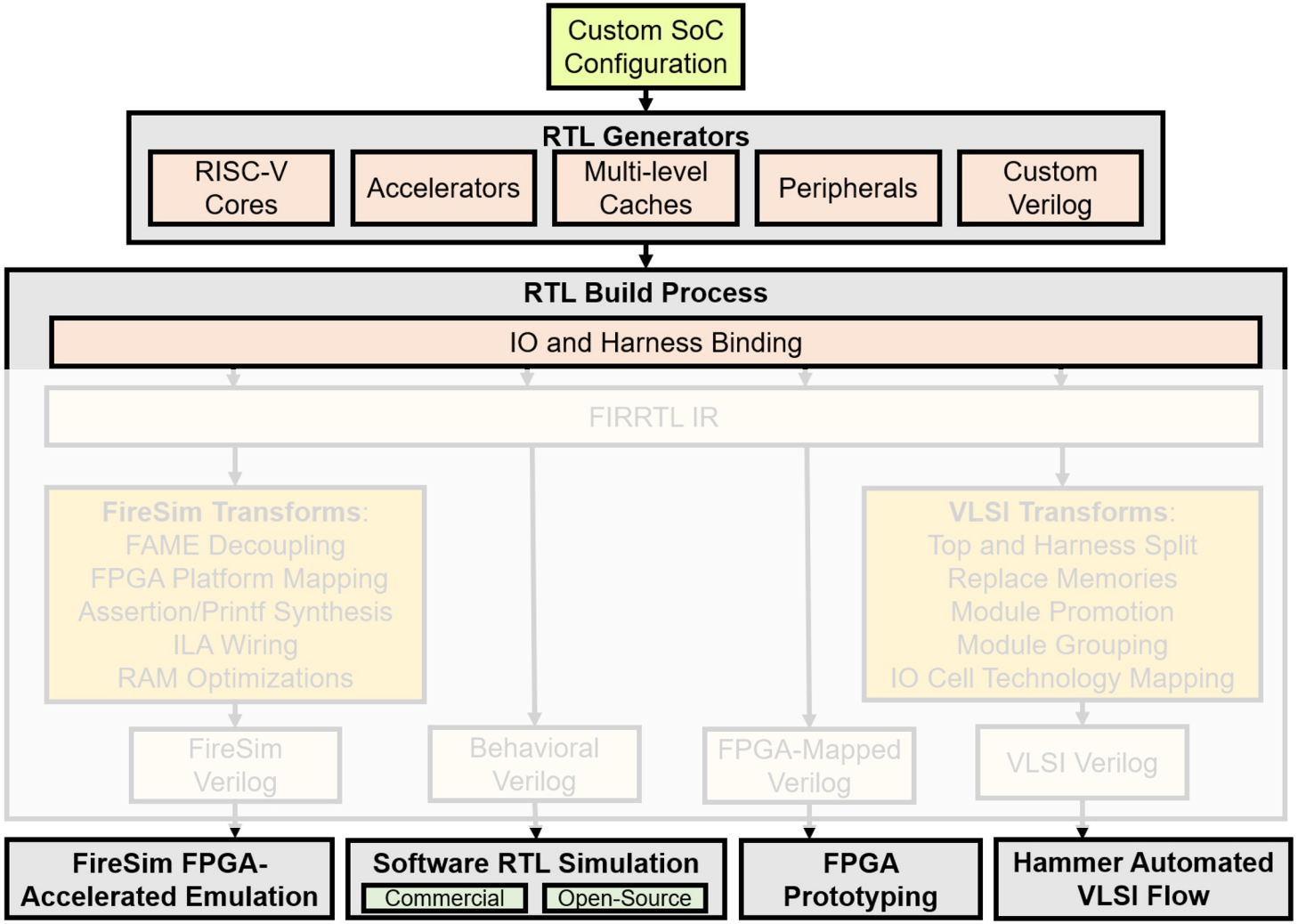
Digital Config

IO Binders

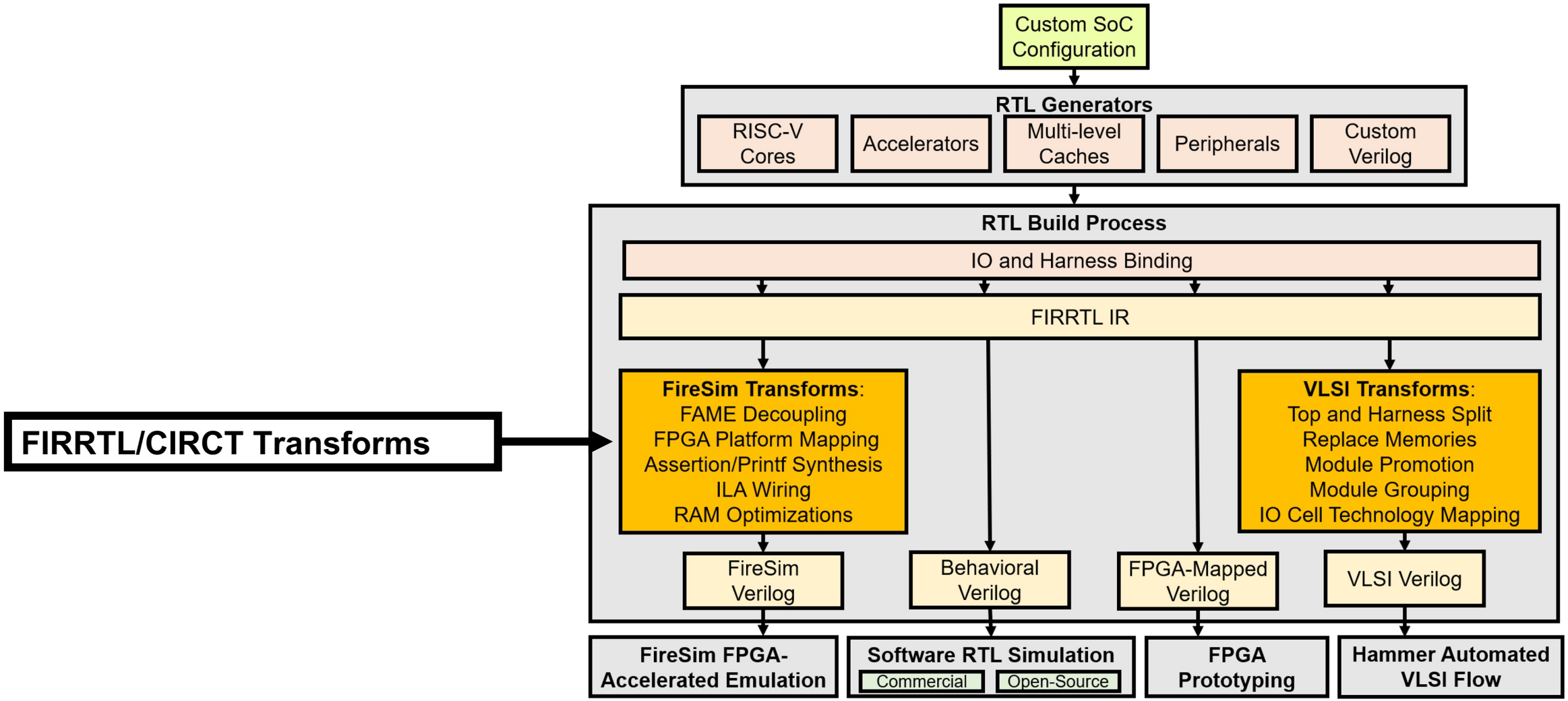
Harness Binders

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

  new WithIOCellModels ++


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

)
```

CHIPYARD Organization

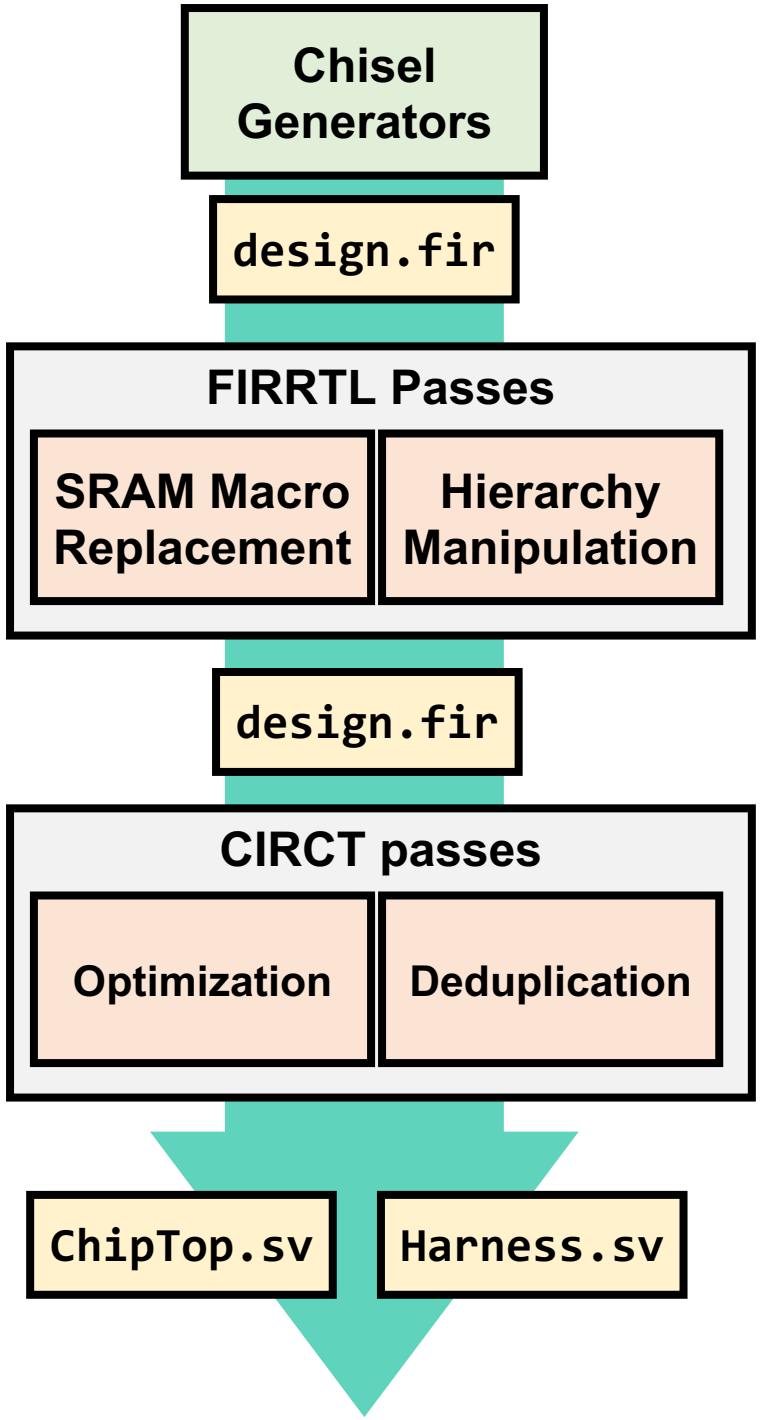


CHIPYARD Organization




Elaboration Flow

- **Chisel** programs generate **FIRRTL** representations of hardware
- **FIRRTL passes** transform the target netlist
  - FireSim's AutoCounter/AutoILA/Printf use FIRRTL passes
  - VLSI flows can use FIRRTL passes to adjust the module hierarchy
- **CIRCT Backend**
  - CIRCT generates tool-friendly synthesizable Verilog from FIRRTL
  - Very fast/powerful FIRRTL compiler



CHIPYARD Organization

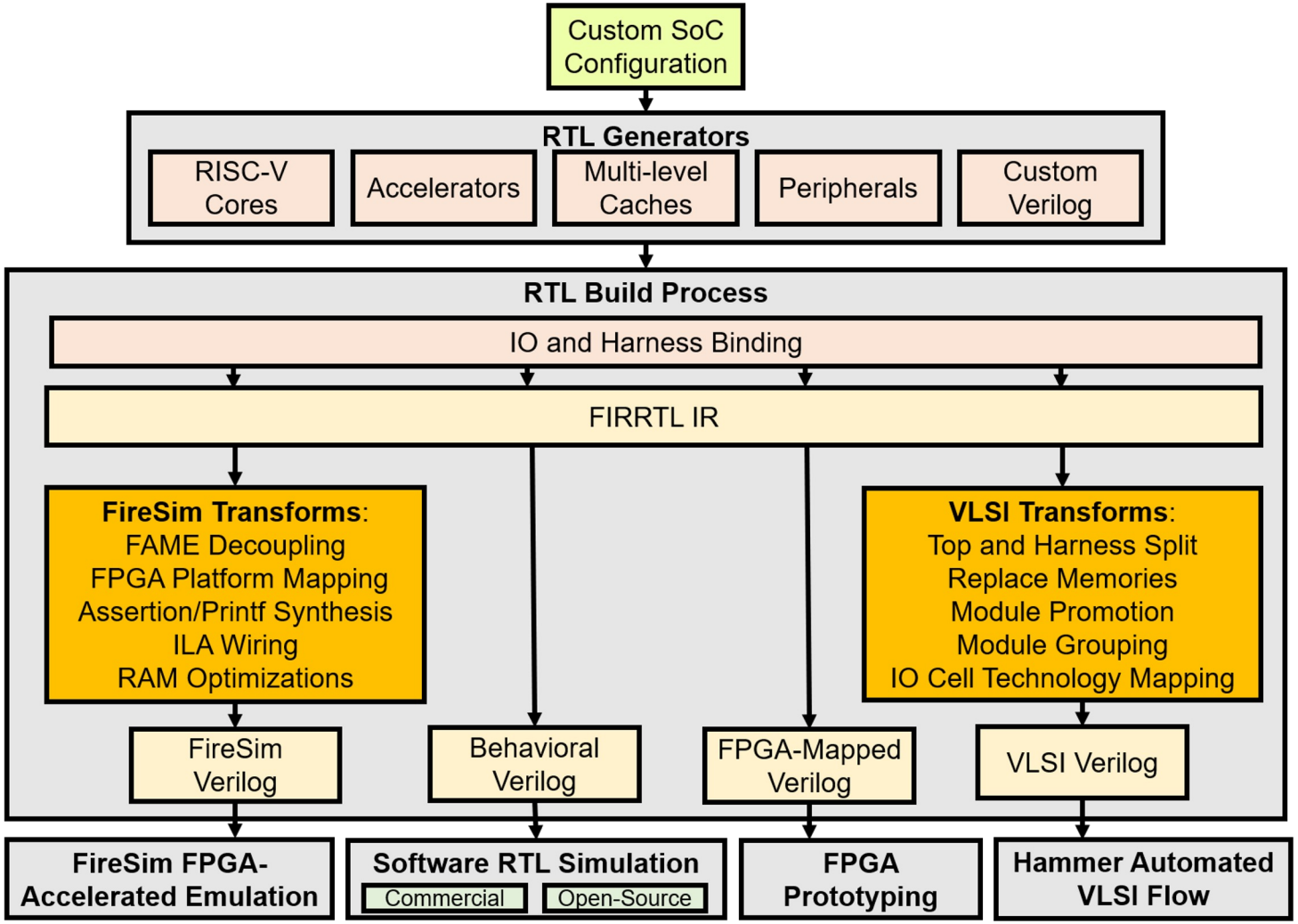
Configs: Describe parameterization of a multi-generator SoC

Generators: Flexible, reusable library of open-source Chisel generators (and Verilog too)

IOBinders/HarnessBinders: Enable configuring IO strategy and Harness features

FIRRTL/CIRCT Passes: Structured mechanism for supporting multiple flows

Target flows: Different use-cases for different types of users

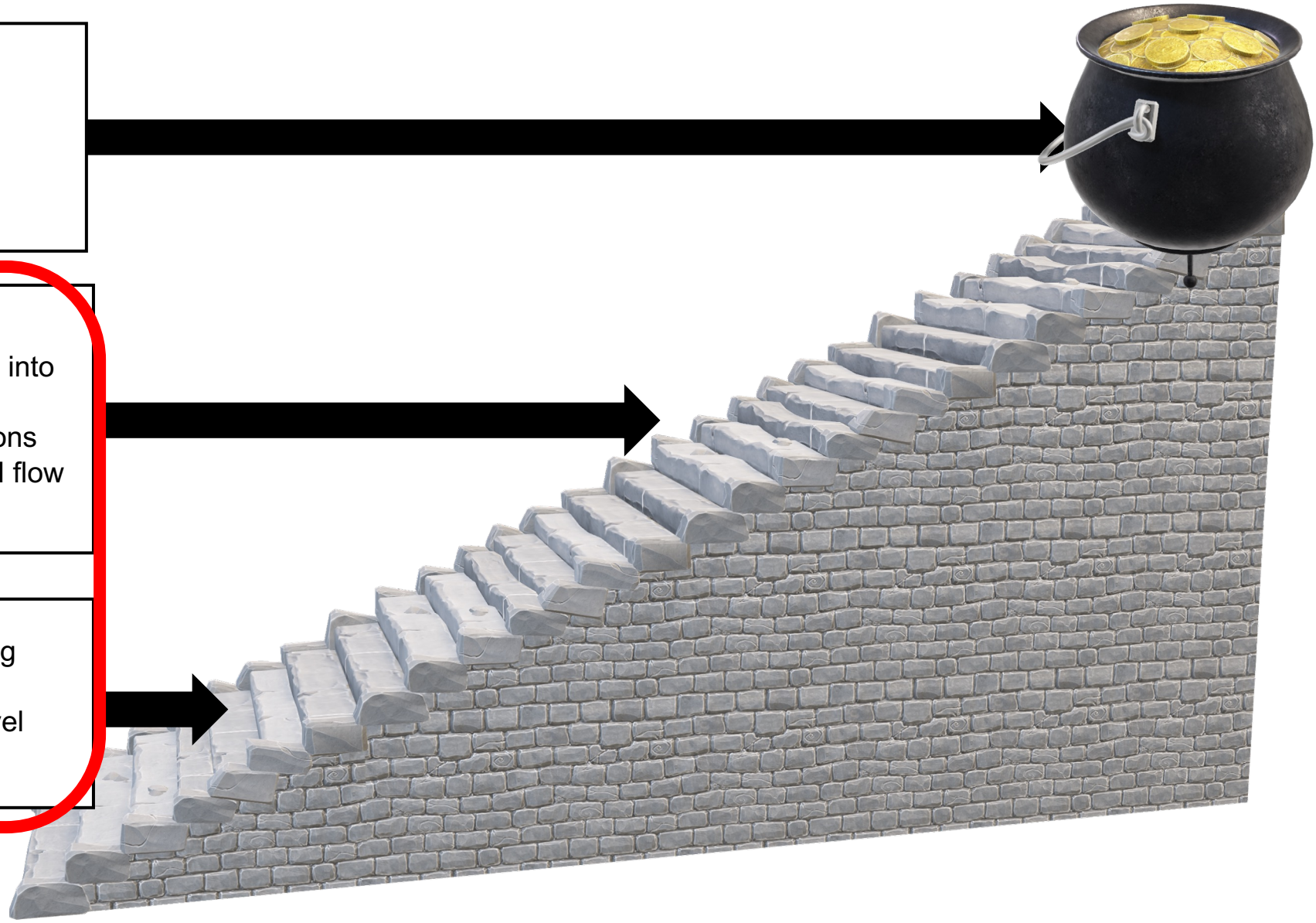


CHIPYARD Learning Curve

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



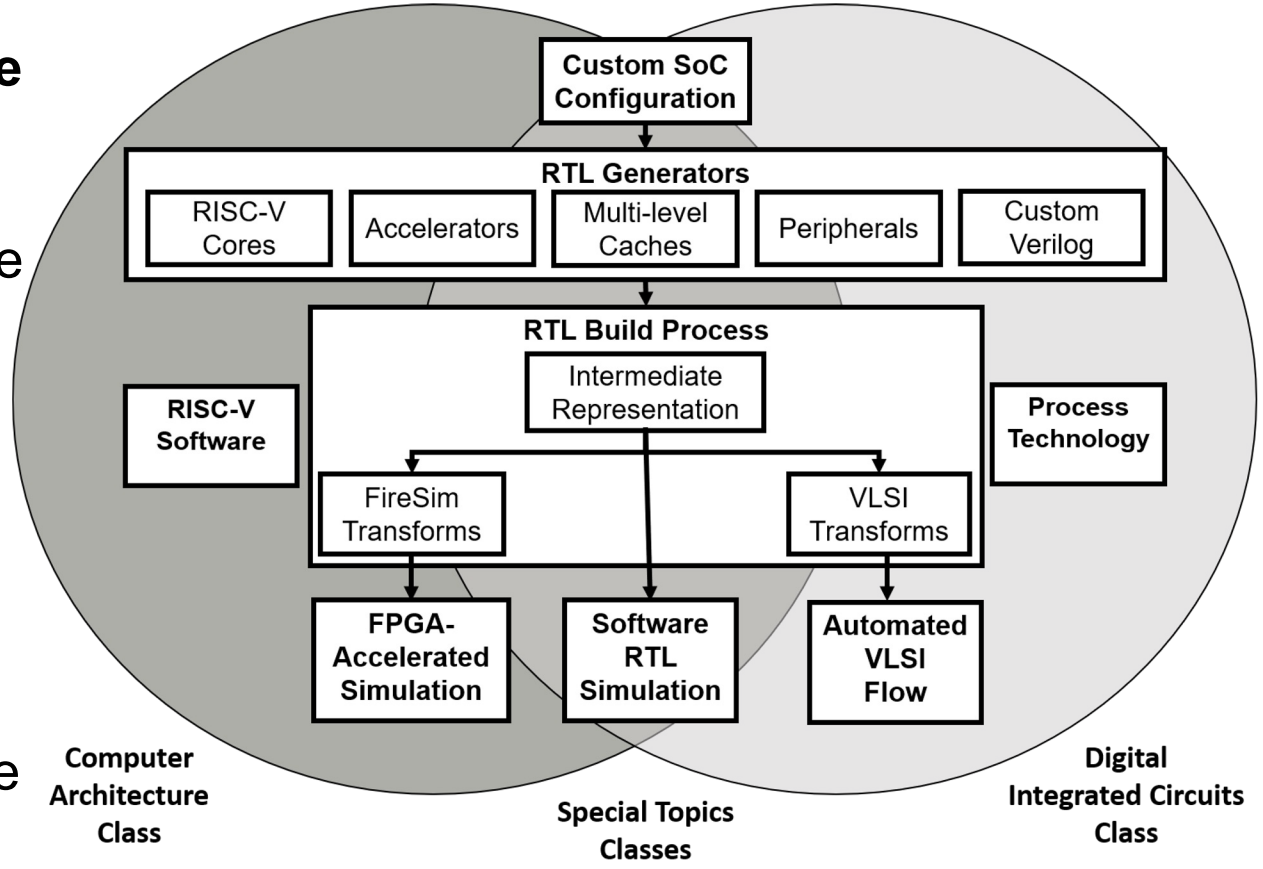
CHIPYARD For Education

Proven in many Berkeley Architecture courses

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

Advantages of common shared HW framework

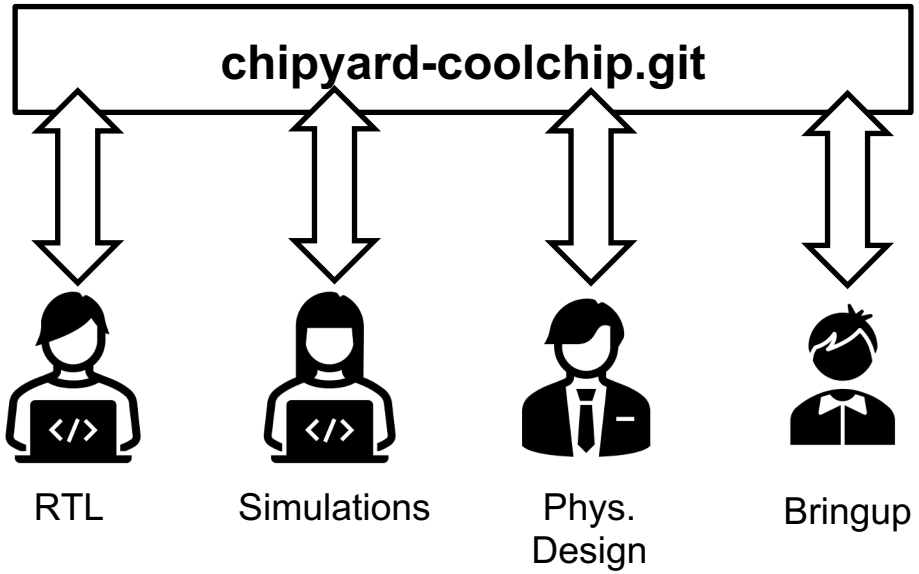
- Reduced ramp-up time for students
- Students learn framework once, reuse it in later courses
- Enables more advanced course projects (tapeout a chip in 1 semester)



CHIPYARD For Tapeouts

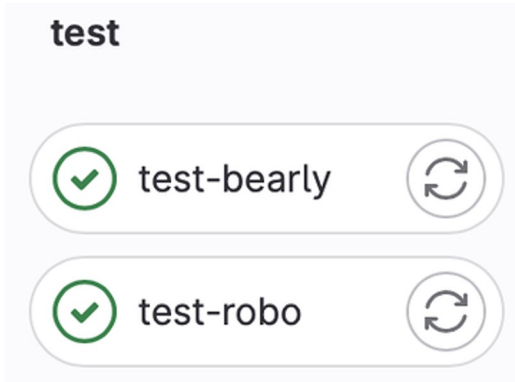
Standard Chipyard "Flow" For Tapeout

1. **RTL Development –** Develop new accelerators/devices and test rapidly
2. **FireSim –** Evaluate your design on real workloads
3. **Hammer -** Reusable/extensible VLSI flow
4. **Bringup –** generate FPGA bringup platforms using Chipyard



Chipyard is a single-source-of-truth for a chip

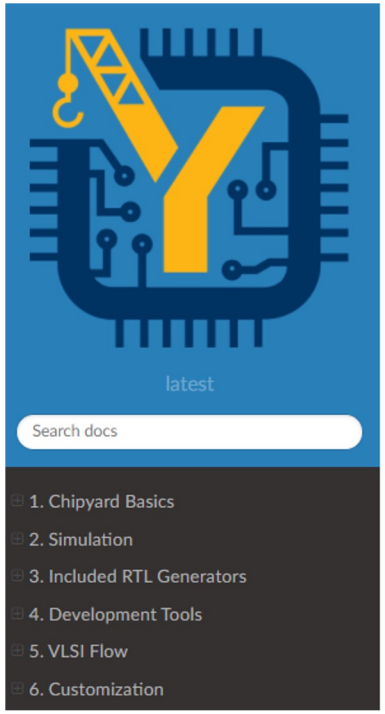
- Enables parallel workflows across different parts of the flow
- Reproducible environments simplify debugging
- Continuous integration for tapeouts



CHIPYARD Community

Documentation:

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

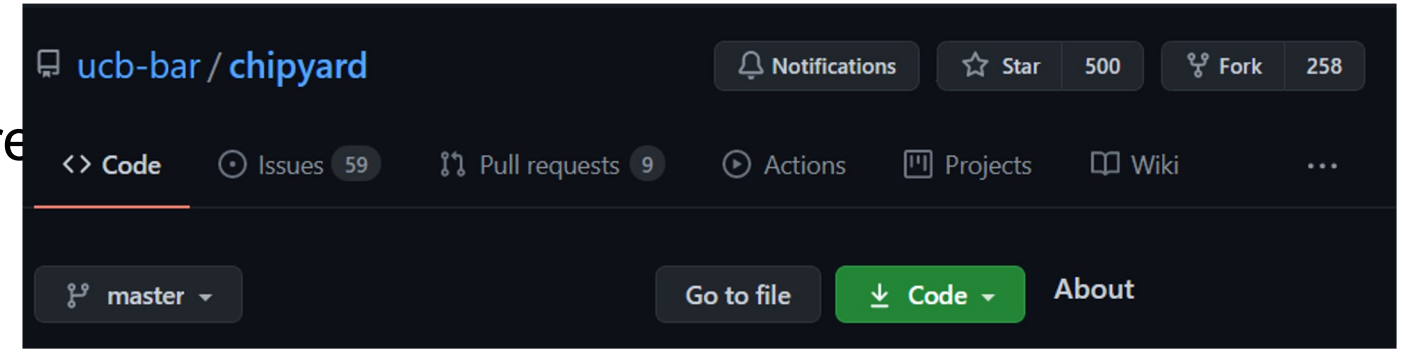


Mailing List:

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

Open-sourced:

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



CHIPYARD

An open, extensible research and design platform for RISC-V SoCs

- Unified framework of parameterized generators
- One-stop-shop for RISC-V SoC design exploration
- Supports variety of flows for multiple use cases
- Open-sourced, community and research-friendly